乐升半导体
LEVETOP

LT268B/C/D

LT269

LT268C/D/269 内部 MCU 核

Cortex M4 Micro Controller

MCU 内 核 规 格 书

V1.0

www.levetop.cn
Levetop Semiconductor Co., Ltd.

目　　录

图表目录

表格目录

1 总章

1.1 概述

本规格书是描述用于 LT268B/C/D、LT269 串口屏芯片内部的 MCU 核微控制器，供需要进行串口屏二次开发的使用者参考，LT268B/C/D、LT269 串口屏芯片并未将所有 MCU 核控制的硬件管脚引出，因此规格书内对于管脚的控制不全然有效，使用者必须参考及对照 LT268B/C/D、LT269 的规格书。

表格 1-1：M4 核组成表

产 品 组 成	名 称	注 释
处理器系统	Cortex-M4	典型工作频率 120~150MHz
	计时器（PIT*2）/ 看门狗（WDT / TC） / 32.768KHz 实时时钟模块（RTC）	
	DMAC*2 / EDMAC*2 / 中断控制器（NVIC）	
	时钟和复位	时钟源支持内部 120MHz OSC 和 8MHz OSC 系统时钟支持 2/3/4/5 等奇偶分频
存储器	ROM	24KByte
	RAM	256KByte
	FLASH	Main Block：512KByte（用户可用到 508KB） Info Block：512Byte 页大小：512Byte 片擦除时间：20ms 快速页擦除：2ms（典型值） 字编程时间：100us（典型值） 页编程时间：2ms（典型值） 数据最大可擦写次数：100K 数据保存时间：10 年
支持接口	SPI*3	支持主从模式/IO 可复用为 GPIO 端口最高速率 20Mbps
	USB2.0 OTG	支持 HS/FS 模式/支持 8 个端点/支持外接晶振和无晶振模式（无晶振模式不支持 HS）
	EPORT*5	40 个 GPIO 支持边沿检测和电平检测
	PWM	4 路可独立配置管脚
	TSI	支持 2 通道输入 支持触发模式和扫描模式 支持低功耗休眠和唤醒

产品组成	名称	注释
	ADC	
	DAC	
	SSI*2	仅支持主模式 SSI1 支持 XIP 模式
	ISO7816*2	支持 ISO/IEC 7816-3 协议 可配置为卡和读卡模式 IO 支持 5/3.3/1.8V 电压（ISO7816-1 不支持 5V） IO 可复用为 GPIO
	UART*3	时钟源可以选择内部 120MHz OSC IO 可复用为 GPIO 支持 16byte 的 FIFO
	I2C*3	支持主从模式/IO 可复用为 GPIO 可使用软件选项在标准/快速模式和高速模式之间切换。

1.2 特性

Cortex-M4 处理器：

- 处理性能强劲，配合快速中断响应
- 增强的系统调试功能，扩展的断点和追踪能力
- 有效率的处理器核心，系统和存储器
- 内置的睡眠模式实现极低功耗
- 内置的存储保护单元（MPU）实现平台安全性
- 兼容 IEEE754 的单精度 FPU

内核配置模块（CCM）：

- 反映启动引导设备
- Master 模式
- Single Chip 模式
- 选择引导设备
- 配置频率检测的配置值
- 选择总线监控器配置：用于异常情况下产生异常信号给 CPU
- 配置 USBPHY 的一些参数
- 配置某些 PAD 的上拉控制

高速缓存模块（Cache）：

- 8KB 大小指令数据共用 Cache
- 16 字节 Cache 行
- 支持直写和写回模式
- 支持 Cache 组命令和行命令
- 支持页清除命令

时钟电源管理模块（CPM）：

- 两个系统时钟源
- 内部高速振荡器，频率为 120MHz
- 内部低速振荡器，频率为 8MHz
- 支持低功耗模式
- 独立的时钟分频设置
- 独立的模块时钟开关

USB2.0 控制器（USBC）：

- 所有事务调度硬件执行
- 可作为主机/从机与其他 USB 设备进行点对点通信，或作为 USB 设备的功能控制器
- 同步 FIFORAM 接口
- 符合 USB 2.0 高速（480 Mbps）功能标准和 OTG 补充协议标准
- 支持与一个 USB 设备进行高速/全速/低速点对点通信
- 支持会话请求协议（SRP）和主机协商协议（HNP）
- 支持挂起和恢复
- 可配置最多 15 个附加传输端点和最多 15 个附加接收端点
- 可配置的 FIFO 深度，支持动态 FIFO 大小
- 支持对 FIFO 的 DMA 访问
- 可以软件控制 USB 连接/断开
- 作为从机使用时，不支持 VBUS 选择

内部闪存模块（EFLASH）：

- 存储器包含 512KB 主存储区（用户可用到 508KB）和 512B 信息存储区
- 按字节（8 位）、半字（16 位）和字（32 位）读取
- 编程和擦除自动化操作
- 带 ECC 校验，并产生 ECC 错误标志
- 可配置产生中断当命令完成后
- 数据保存时间：10 年
- 0.99~1.21 伏/1.5~1.98 伏双电源供电

串行接口模块（SPI）：

- 主模式和从模式
- 从选择输出
- 模式错误标志有 CPU 中断功能
- Doze 模式可以进行 SPI 操作
- 低功耗下可降低驱动
- Freescale SPI 以及 Texas 串行接口可用的可编程的接口操作
- 收发独立的 FIFO，均为 8 位宽以及 8 深度
- 4-16 位可编程数据页
- 在诊断和调试测试中，有内部可循环的测试操作
- 标准的机于 FIFO 的中断以及机于传输结束的中断
- 用 DMA 可以进行有效率的传输
- 调试时有可视的 TX 以及 RX FIFO
- 传输时序调整可用高速模式

通用异步收发器（UART）：

- 支持全双工操作
- 支持 NRZ（非归 0）通信格式
- 13 位波特率选择
- 可编程 8 位、9 位的数据字长度
- 独立使能的发送器和接收器
- 独立的发送器和接收器中断请求
- 发送器输出极性可编程
- 2 种接收器唤醒方式
- 空闲线唤醒
- 地址标记唤醒
- 8 种中断方式
- 发送器空
- 发生完成
- 接收器满
- 接收器空闲输入
- 接收器溢出
- 噪声错误
- 帧错误
- 奇偶校验错误
- 接收器帧错误侦测
- 硬件奇偶校验检查
- 1/16 位时间噪声检查
- 支持通用输入输出功能
- 支持低速串行 IR 接口功能，兼容 IrDA(最高可达 115.2Kbit/s)
- 独立的 16x9 发送和接收 FIFO，以减少 CPU 中断服务的调用
- FIFO 触发级别为 1/8、1/4、1/2、3/4 和 7/8
- 支持 DMA 传输
- 支持硬件流控功能

脉冲宽度调制模块（PWM）：

- 可编程的周期
- 可编程占空比
- 2 个死区发生器
- 捕捉功能
- 可以配置为 GPIO

I2C 总线（I2C）：

- 支持 7 位寻址和 10 位寻址。
- 支持三种模式：标准模式、快速模式和高速模式。
- 可使用软件选择在标准/快速/高速模式之间切换。
- 与 2.1 版本的 I2C 总线标准模式和快速模式兼容。
- 支持多主机操作。
- 可编程选择 64 种不同串行频率时钟。
- 软件可选应答位
- 基于中断的驱动方式，逐字节地传输数据。
- 传输完成并读取配置的中断。
- 自动从主机模式切换到从机模式的仲裁丢失中断。
- 生成/检测 START 和 STOP 信号。
- 生成重复 START 信号。
- 生成/检测应答信号。
- 总线忙状态检测。
- 当系统时钟处于停止模式时，可选从机地址接收使能。
- 支持 SCL 或 SDA 的 GPIO 功能。

模数转换器模块（ADC）：

- 高性能
- 可配置 12 位、10 位、8 位或 6 位分辨率
- ADC 转换时间：12 位分辨率（1MHz）时为 1.0 微秒，10 位分辨率转换时间为 0.88 微秒，通过降低分辨率可以获得更快的转换时间
- 可编程采样时间
- 数据对齐以保持内置数据一致性
- 支持 DMA
- 低功耗
- 可以在低功耗运行时降低 PCLK 的频率来保持最佳的性能。比如可以在不论 PCLK 的频率情况下，保持 ADC 的 1.0 微秒的转换时间
- 等待模式：使用低频的 PCLK 来防止 ADC 在应用中溢出
- 自动关闭模式：除了主动转换阶段外，ADC 会自动关闭来降低功耗
- 模拟输入通道
- 8 个外部模拟输入通道
- 1 个内部源检测通道
- 初始化转换
- 软件
- 可配置极性的外部硬件触发器
- 转换模式
- 可以转换单一通道或者扫描一系列通道
- 单通道模式每触发一次转换选定的输入
- 连续模式可连续转换选定的输入
- 不连续模式
- 在采样结束、转换结束、序列转换结束以及发生模拟看门狗或溢出事件时产生中断
- 模拟看门狗
- 单端和差分输入配置
- 转换器可选择使用内部参考或外部参考
- 与 DMA 兼容的数据收集功能

数字模拟转化器（DAC）：

- 数据左对齐或右对齐
- 支持 DMA 功能
- 外部转换触发器
- 可编程内部缓冲区
- 输入电压参考 Vrefh
- 基于 FIFO 的操作

复位控制器模块（RESET）：

- 复位的触发源
- 上电复位
- 软件
- 看门狗模块复位
- TC 计时器复位
- 7816 ISORST 复位
- 高、低电压检测复位
- 芯片完成复位状态后，产生 RSTOUT 信号
- 可让软件检查上次复位原因的状态标志位

实时时钟模块（RTC）：

- 能够向天数、小时、分钟、秒寄存器中导入时间数据，并将数据读出。
- 支持设置闹钟。
- 中断源：天数、时、分、秒中断，可编程闹钟中断，1KHz/32KHz 周期中断。

32 位可编程中断计时器模块（PIT32）：

- 32 位计时器，在最少处理器干预的情况下提供精确的定时中断。
- 可以从模数锁存器内写入的值开始递减，也可以是一个自由运行的降值计数器。

中断向量嵌套控制器（NVIC）：

- 64 个可屏蔽中断通道（不包括 16 条带 FPU 的 Cotex-M4 的中断线）；
- 8 个可编程优先级（使用 3 位中断优先级）；
- 低延迟异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现；

注意： NVIC 和处理器核心接口是紧密耦合的，这使得低延时中断处理和有效处理后到达的中断成为可能。所有的中断包括内核异常都由 NVIC 管理。

EDMAC 控制器（EDMAC）：

- 支持双通道
- 可编程传输数据数量
- 可编程读缓存地址和写缓存地址
- 多外设选择
- 支持读、写、写后读传输

直接内存存取控制器模块（DMA）：

- DMA1 有 4 个独立的可编程通道，DMA2 有 2 个独立的可编程通道
- 支持 8/16/32 位数据传输
- 支持单次传输，连续 4/8/16 次传输
- 支持链表传输
- 遵循一个固定的优先级
- 支持通道暂停操作
- 支持外设传输

同步串行接口（SSI）:

- 串行主設備操作。
- DMA 控制器接口–使 SSI 使用握手接口处理传输请求，通过总线与 DMA 控制器接口。
- 支持扩展 SPI 传输中的时钟延长。
- FIFO 深度–在正常传输模式下为 8 字。在芯片内执行（XIP）传输模式下为 32 字。
- FIFO 宽度固定为 32 位。
- 支持扩展 SPI。
- 支持芯片内执行（XIP）模式。
-

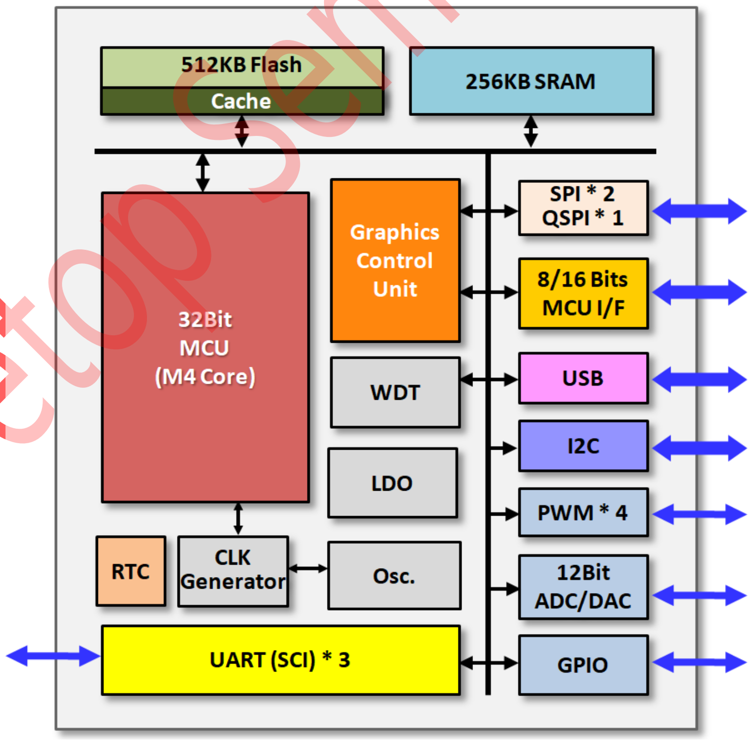
看门狗模块（WDT）:

- 自减计数器，它会产生下溢复位。为了防止复位，软件必须周期性地维护看门狗模块重新设置计数器。
- 16 位计时器，帮助软件从失控程序中恢复正常运行。

1.3 系统框图

图表 1-1 是 LT268C/D 串口屏芯片内部的 M4 核微控制器系统框图。

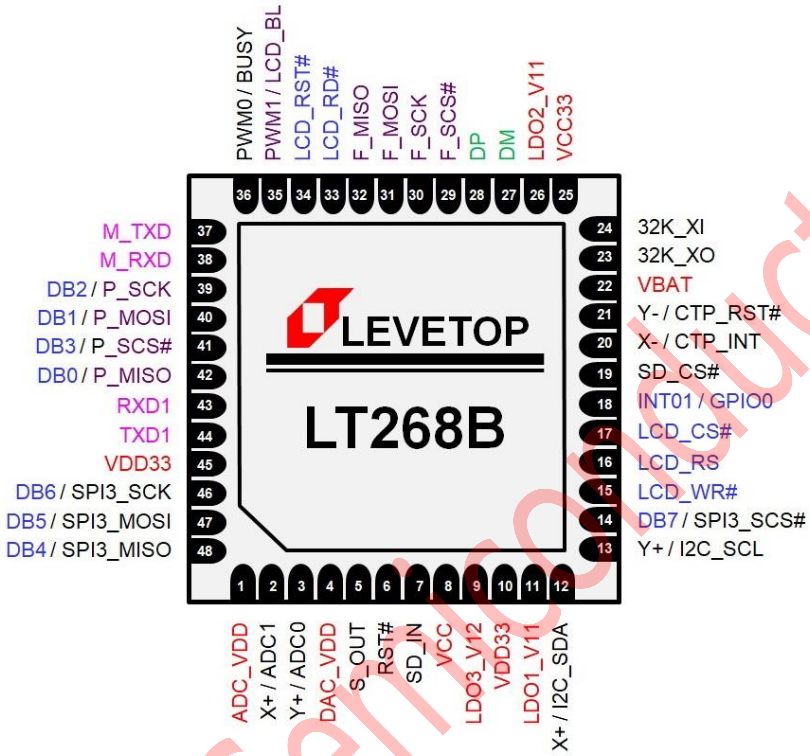
图表 1-1：系统框图



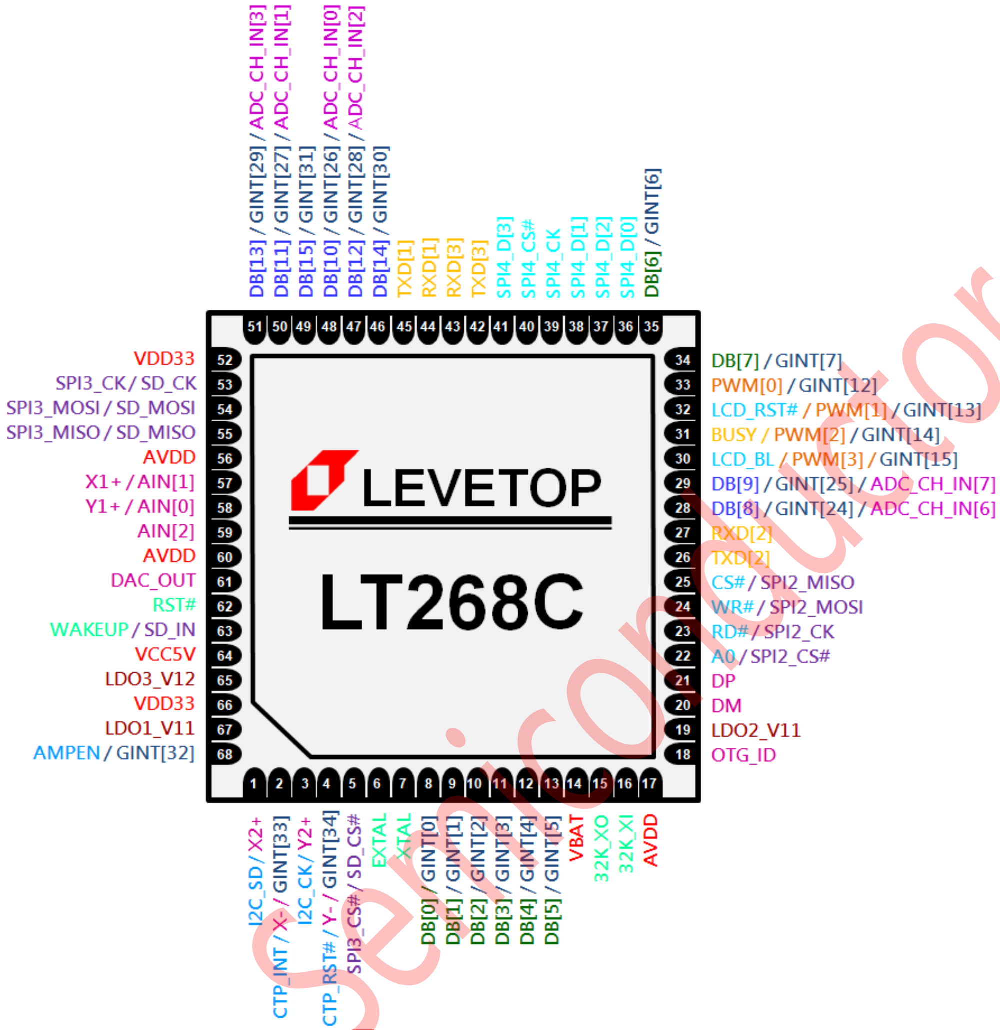
2 管脚描述

2.1 封装管脚汇总

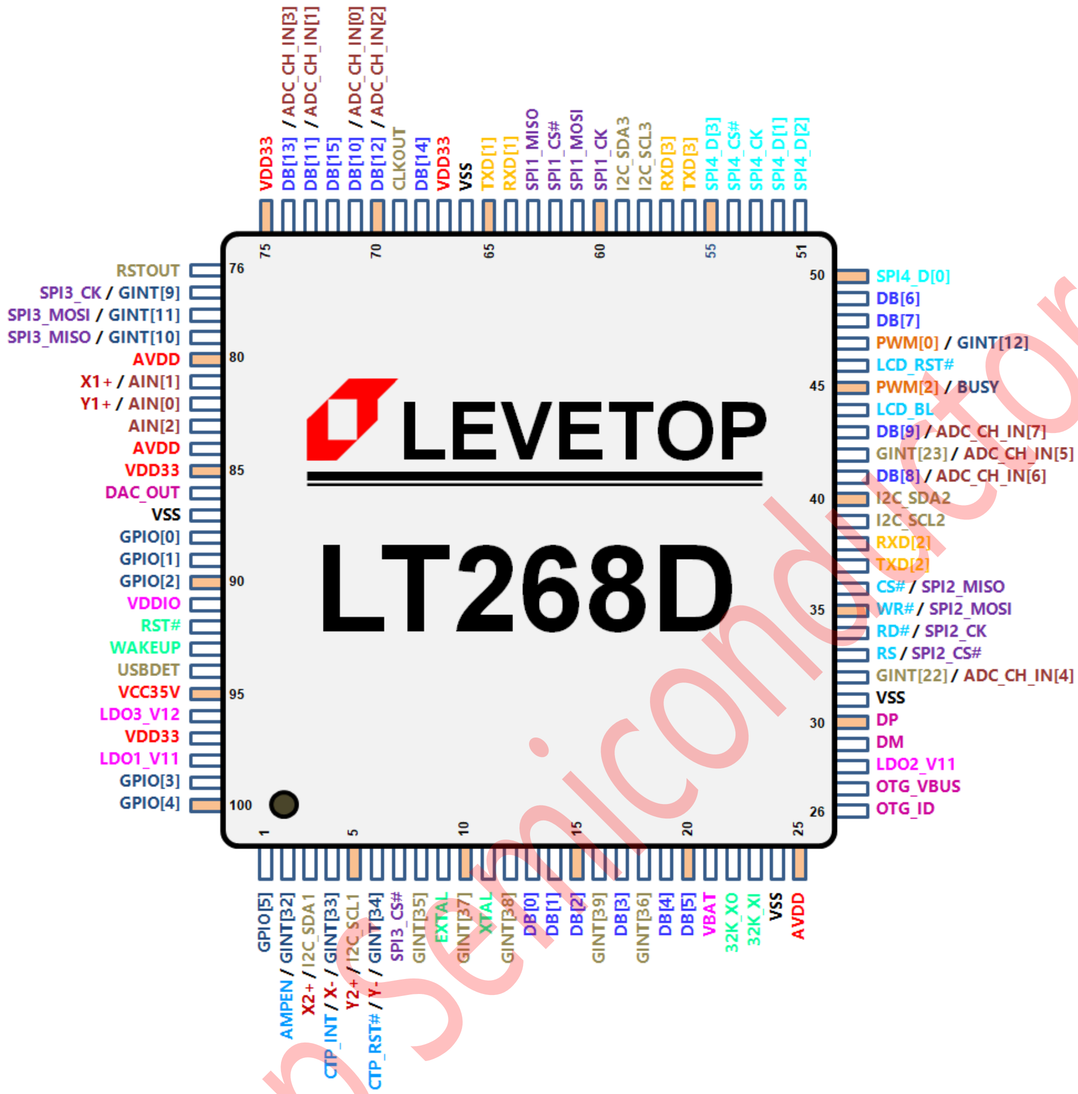
图表 2-1：LT268B 管脚(QFN-48)



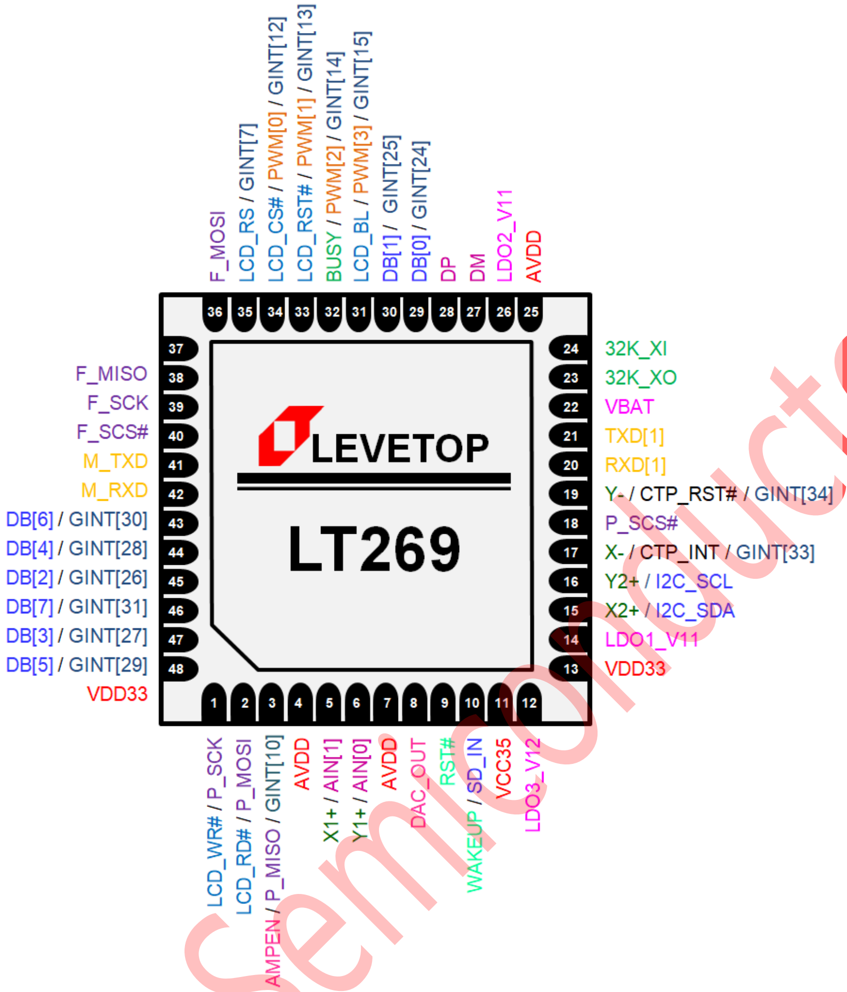
图表 2-2：LT268C 管脚(QFN-68)



图表 2-3：LT268D 管脚(LQFP-100)



图表 2-4：LT269 管脚(QFN-48)



2.2 LT268B 管脚信号说明

2.2.1 Uart 串口信号

表 6-1：LT268B Uart 串口信号

Pin#	引脚名称	I/O	功 能 说 明
38	M_RXD	I	**串口通信(Uart) 接收数据输入** 这是 Uart 串行通信接口的数据输入口，用来接收主控端 MCU 的 TX 送来指令信号。
37	M_TXD	O	**串口通信(Uart) 发送数据输出** 这是 Uart 串行通信接口的数据输出口，用来传送指令信号到主控端 MCU 的 RX。
36	BUSY	O	**忙状态输出** 此脚为串口屏的忙状态输出，主控端可以由此信号了解串口屏是否为忙状态。 此脚也可以做作为 PWM[2] 输出或是仿真 ARM 调试模式的串行数据输入输出数据信号线 SWIO。

2.2.2 MCU 屏接口信号

表 6-2：LT268B MCU 屏接口信号

Pin#	引脚名称	I/O	功 能 说 明
41	P_SCS#	O	**MCU 屏上的 SPI 芯片选择信号** 此信号为 LT268B 对外部 SPI 屏的的片选输出。此信号与 DB[3]信号共享。
42	P_MISO	I	**MCU 屏上的 SPI 数据输入信号** 此信号为 LT268B 由外部 SPI 屏的读取数据输入。此信号与 DB[0]信号共享。
40	P_MOSI	O	**MCU 屏上输出到 SPI 的数据信号** 此信号为 LT268B 输出数据到外部 SPI 屏。此信号与 DB[1]信号共享。
39	P_SCK	O	**MCU 屏上的 SPI 串行时钟信号** 此信号是提供 SPI 屏的时钟信号输出。此信号与 DB[2]信号共享。
16	LCD_RS	O	**输出到 MCU 屏上的的 RS/A0 信号** 使用 SPI 屏时，为 LT268B 输出到 SPI 屏的 A0 信号。

14, 46, 47, 48, 41, 39, 40, 42	DB[7:0]	IO	**8Bit 的 MCU 屏数据信号** 这些信号为 LT268B 对外部 8Bit MCU 屏的数据传输信号。当使用 SPI 屏的接口时 DB[3:0]这些信号将作为 SPI 屏控制信号；DB[7:4]这些信号将作为另一组 SPI 接口控制信号。
17	LCD_CS#	O	**8Bit 的 MCU 屏芯片选择信号** 这些信号为 LT268B 对外部 8Bit MCU 屏的数据传输信号。
15	LCD_WR#	O	**8Bit 的 MCU 屏写入控制信号** 这些信号为 LT268B 对外部 8Bit MCU 屏的数据传输信号。
33	LCD_RD#	O	**8Bit 的 MCU 屏读取控制信号** 这些信号为 LT268B 对外部 8Bit MCU 屏的数据传输信号。
34	LCD_RST#	O	**复位输出信号** 此引脚是提供 SPI 屏或是 8Bit MCU 屏的复位信号，当 LCD_RST# = 0 时，将对 TFT 屏产生复位动作。

2.2.3 SPI Flash 接口信号

表 6-3：LT268B SPI Flash 接口信号

Pin#	引脚名称	I/O	功 能 说 明
29	F_SCS#	O	**SPI Flash 的芯片选择信号** 此信号为 LT268B 对外部 SPI Flash 的片选输出。
32	F_MISO	I	**接收 SPI Flash 的数据输入信号** 此信号为 LT268B 由外部 SPI Flash 的读取数据输入。
31	F_MOSI	O	**输出到 SPI Flash 的数据信号** 此信号为 LT268B 输出数据到外部 SPI Flash。
30	F_SCK	O	**SPI 串行时钟信号** 此引脚是提供到外部 SPI Flash 的时钟信号输出。

2.2.4 触控屏信号

表 6-4：LT268B 触控屏信号

Pin#	引脚名称	I/O	功 能 说 明
21	Y-	I	电阻屏 Y- 信号
20	X-	I	电阻屏 X- 信号
3, 13	Y+	I	电阻屏 Y+ 信号，使用时此 2 根信号必须连接仔在一起。

2, 12	X+	I	电阻屏 X+ 信号，使用时此 2 根信号必须连接仔在一起。
13	I2C_SCL	O	使用电容屏时为电容屏上的 I2C 时钟信号。此信号与 Y+的其中一根信号共享。
12	I2C_SDA	IO	使用电容屏时为电容屏上的 I2C 数据信号。此信号与 X+的其中一根信号共享。
21	CTP_RST#	O	使用电容屏时为电容屏上的复位信号。此信号与 Y-信号共享。
20	CTP_INT	I	使用电容屏时为电容屏上的 I2C 中断信号。此信号与 X-信号共享。

2.2.5 USB 控制信号

表 6-5：LT268B USB 控制信号

Pin#	引脚名称	I/O	功 能 说 明
28	DP	IO	**USB 数据端（Positive）** 此为 USB 数据端 DP 的信号。LT268B 可以透过 USB 接口更新内部 MCU 程序及外接 SPI Flash 内的数据。
27	DM	IO	**USB 数据端（Negative）** 此为 USB 数据端 DM 的信号。

2.2.6 SCI/SPI 扩展信号

表 6-6：LT268B SCI/SPI 扩展信号

Pin#	引脚名称	I/O	功 能 说 明
43	RXD1	I	**串口通信(Uart) #1 接收数据输入** 这是第 2 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。
44	TXD1	O	**串口通信(Uart) #1 发送数据输出** 这是第 2 组 Uart 串行通信接口的数据输出口，用来传送信号到其他元件的 RX。
14	SPI3_SCS#	O	**SPI #3 芯片选择信号** 此信号为第 3 组 SPI 的片选输出。与 DB[7] 共享。
48	SPI3_MISO	I	**SPI #3 数据输入信号** 此信号为第 3 组 SPI 的读取数据输入。与 DB[4]信号共享。
47	SPI3_MOSI	O	**SPI #3 的数据信号** 此信号为第 3 组 SPI 输出数据。与 DB[5]信号共享。
46	SPI3_SCK	O	**SPI #3 串行时钟信号** 此信号为第 3 组 SPI 的时钟信号输出。与 DB[6]信号共享。

2.2.7 其他控制信号

表 6-7：LT268B 其他控制信号

Pin#	引脚名称	I/O	功 能 说 明
19	SD_CS#	O	**SD 卡（SPI 模式）片选信号** 此信号用在 SPI 模式的 SD 卡，作为片选信号。
7	SD_IN	I	**SD 卡（SPI 模式）侦测信号**
39	SD_CLK	O	**SD 卡（SPI 模式）时钟信号** 此信号与 MCU 屏上的 SPI 串行时钟信号 P_SCK 及并口 DB[2]信号共享。
40	SD_MOSI	O	**SD 卡（SPI 模式）数据输出信号** 此信号与 MCU 屏上输出到 SPI 的数据信号 P_MOSI 及并口 DB[1]信号共享。
42	SD_MISO	I	**SD 卡（SPI 模式）数据输入信号** 此信号与 MCU 屏上的 SPI 数据输入信号 P_MISO 及并口 DB[0]信号共享。
6	RST#	I	**复位输入信号** 当 RST# = 0 时，将对内部 MCU 产生复位动作，除了少数由 POR 才能复位的寄存器外，大多数由 MCU 控制的寄存器将回复到默认值。
36	PWM0, BUSY	I	**PWM 0 输出** 此为 LT268B 内部 PWM 输出。 PWM[0] 可以做作为串口屏的 BUSY 忙状态输出。
35	PWM1, LCD_BL	I	**PWM 1 输出** 此为 LT268B 内部 PWM 输出。 PWM[1] 可以做作为串口屏的背光控制信号。
2, 3	ADC[0:1]	I	**模拟输入** 此为 LT268B 仿真信号输入，可预留做其他控制信号使用。
5	S_OUT	O	**模拟输出** 此为 LT268B 仿真信号输出，可作为音频输出或预留做其他控制信号使用。
18	INT01 GPIO0	IO	**中断输入 / 通用输出入** 此信号可以做为中断输入，或是当成普通的输出输入 IO 口。

2.2.8 晶振与电源信号

表 6-8：LT268B 晶振与电源信号

Pin#	引脚名称	I/O	功 能 说 明
24	32K_XI	I	**External RTC Osc Input**
23	32K_XO	O	**External RTC Osc Output**
25	VCC33	PWR	**3.3V 电源输入（USB)**
8	VCC	PWR	**3.3V 电源输入（System)**
1, 4	AVDD	PWR	**3.3V ADC/DAC 电源输入**
22	VBAT	PWR	**2.2V~3.3V RTC 电池电源输入**
10, 45	VDD33	PWR	**3.3V 内核电源输出（I/O)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
11	LDO1_V11	PWR	**1.1V 内核电源输出#1 (Flash)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
26	LDO2_V11	PWR	**1.1V 内核电源输出#2 (USB PHY)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
9	LDO3_V12	PWR	**1.2V 内核电源输出（Core)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
-	Thermal Pad	-	**GND 接地散热焊盘** IC 的背部散热焊盘直接接地(VSS/GND)。

2.3 LT268C 管脚信号说明

2.3.1 MCU 屏接口信号

表格 2-1：LT268C MCU 屏接口信号

Pin#	管脚名称	I/O	功 能 说 明
34, 35, 13, 12, 11, 10, 9, 8	DB[7:0] GINT[7:0]	IO	**8/16Bit 并口的 MCU 屏数据信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268C 对外部 8/16Bit MCU 屏的数据传输信号。 当使用 SPI 的 MCU 屏接口时，这些信号可以作为 GPIO 或是中断输入信号 GINT[7:0] 使用。
49, 46, 51, 47, 50, 48, 29, 28	DB[15:8] GINT[31:24]	IO	**16Bit 并口的 MCU 屏数据信号** 当使用8/16Bit并口的MCU屏时,这些信号为LT268C对外部16Bit MCU 屏的数据传输信号。 当使用 8Bit MCU 屏或是 SPI 的 MCU 屏接口时,这些信号可以作为 GPIO 或是中断输入信号 GINT[31:24] 或是 ADC 通道输入 ADC_CH_IN 使用。
25	CS# SPI2_MISO	IO	**8/16Bit 并口的 MCU 屏芯片选择信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268C 对外部 8/16Bit MCU 屏的片选控制信号。 **串口的 MCU 屏上的 SPI 数据输入信号** 当使用 SPI 的 MCU 屏接口时，此信号为 LT268C 由外部 MCU 屏的读取数据输入。
24	WR# SPI2_MOSI	O	**8/16Bit 并口的 MCU 屏写入控制信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268C 对外部 8/16Bit MCU 屏的数据写入控制信号。 **串口的 MCU 屏上输出到 SPI 的数据信号** 当使用 SPI 的 MCU 屏接口时，此信号为 LT268C 输出数据到 MCU 屏。
23	RD# SPI2_CK	O	**8/16Bit 并口的 MCU 屏读取控制信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268C 对外部 8/16Bit MCU 屏的数据读取控制信号。 **串口的 MCU 屏上的 SPI 串行时钟信号** 当使用 SPI 的 MCU 屏接口时，此信号是提供 SPI 屏的时钟信号输出。
22	A0 SPI2_CS#	O	**8/16Bit 并口的 MCU 屏上的 RS/A0 信号** 当使用 8/16Bit 并口的 MCU 屏时，使用 SPI 屏时，为 LT268C 输出到 SPI 屏的 A0 信号。 **串口的 MCU 屏上的 SPI 芯片选择信号** 当使用 SPI 的 MCU 屏接口时，此信号是提供 SPI 屏的片选信号输出。
30	LCD_RST# PWM[1] GINT[13]	O	**MCU 屏上的复位输出信号** 此管脚是提供 8/16Bit MCU 屏或是 SPI 屏的复位信号，当 LCD_RST# = 0 时，将对 TFT LCD 屏产生复位动作。 此信号与 PWM[1]、GINT[13] 信号共享。

2.3.2 SPI Flash 接口信号

表格 2-2：LT268C SPI Flash 接口信号

Pin#	管脚名称	I/O	功 能 说 明
53	SPI3_CK SD_CK	O	**SPI 组件的时钟信号** SPI3 主要是接到外部 SPI 组件。此管脚是提供到外部 SPI 组件的时钟信号输出。 **SD 卡（SPI 模式）时钟信号** 此信号也可以作为 SD 卡（SPI 模式）时钟信号输出
5	SPI3_CS# SD_CS#	O	**SPI 组件的芯片选择信号** 此信号为 LT268C 对外部 SPI 组件的片选输出。 **SD 卡（SPI 模式）片选信号** 此信号也可以作为 SD 卡（SPI 模式）的片选输出
55	SPI3_MISO SD_MISO	I	**接收 SPI 组件的数据输入信号** 此信号为 LT268C 由外部 SPI 组件的读取数据输入。 **SD 卡（SPI 模式）数据输入信号** 此信号也可以作为 SD 卡（SPI 模式）数据输入信号。
54	SPI3_MOSI SD_MOSI	O	**输出到 SPI 组件的数据信号** 此信号为 LT268C 输出数据到外部 SPI 组件。 **SD 卡（SPI 模式）数据输出信号** 此信号也可以作为 SD 卡（SPI 模式）数据输出信号。
39	SPI4_CK	O	**QSPI Flash 串行时钟信号** SPI4 主要是接到外部 QSPI Flash，用来读取储存在 QSPI Flash 的图片、动画、字库等信息。此管脚是 QSPI Falsh 的时钟信号输出。
40	SPI4_CS#	O	**QSPI Flash 的芯片选择信号** 此信号为 LT268C 对外部 QSPI 组件的片选输出。
41, 37, 38, 36	SPI4_D[3:0]	I/O	**QSPI Flash 的数据输出入信号** 这些信号为 LT268C 对外部 QSPI 组件读取或写入的数据信号。

2.3.3 Uart 串口与 Uart（SCI）扩展信号

表格 2-3：LT268C Uart（SCI）扩展信号

Pin#	管脚名称	I/O	功 能 说 明
44	RXD[1]	I	**串口通信(Uart) #1 接收数据输入** 这是第 1 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。
45	TXD[1]	O	**串口通信(Uart) #1 发送数据输出** 这是第 1 组 Uart 串行通信接口的数据输出口，用来传送信号到其他组件的 RX。
27	RXD[2]	I	**串口通信(Uart) #2 接收数据输入** 这是第 2 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。
26	TXD[2]	O	**串口通信(Uart) #2 发送数据输出** 这是第 2 组 Uart 串行通信接口的数据输出口，用来传送信号到其他组件的 RX。
43	RXD[3]	I	**串口通信(Uart) #3 接收数据输入** 这是第 3 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。 使用串口屏时，这是 Uart 串行通信的数据输入口，用来接收主控端 MCU 的 TX 送来指令信号。
42	TXD[3]	O	**串口通信(Uart) #3 发送数据输出** 这是第 3 组 Uart 串行通信接口的数据输出口，用来传送信号到其他组件的 RX。 使用串口屏时，这是 Uart 串行通信的数据输出口，用来传送指令信号到主控端 MCU 的 RX。

2.3.4 触控屏信号

表格 2-4：LT268C 触控屏信号

Pin#	管脚名称	I/O	功 能 说 明
57	X1+ AIN[1]	I	**电阻屏 X+ 信号** 使用电阻屏时，此为电阻屏 X+ 信号，使用时 X1+ 和 X2+ 这 2 根信号必须同时接到电阻屏的 X+ 信号。 不使用电阻屏时此信号也可以作为模拟输入信号 AIN[1]。
58	Y1+ AIN[0]	I	**电阻屏 Y+ 信号** 使用电阻屏时，此为电阻屏 Y+ 信号，使用时 Y1+ 和 Y2+ 这 2 根信号必须同时接到电阻屏的 Y+ 信号。 不使用电阻屏时此信号也可以作为模拟输入信号 AIN[0]。

Pin#	管脚名称	I/O	功 能 说 明
1	X2+ I2C_SD	IO	**电阻屏 X+ 信号** 使用电阻屏时，此为电阻屏 X+ 信号，使用时 X1+ 和 X2+ 这 2 根信号必须同时接到电阻屏的 X+ 信号。 **电容屏的 I2C 数据信号** 使用电容屏时为电容屏上的 I2C 数据信号。
3	Y2+ I2C_CK	IO	**电阻屏 Y+ 信号** 使用电阻屏时，此为电阻屏 Y+ 信号，使用时 Y1+ 和 Y2+ 这 2 根信号必须同时接到电阻屏的 Y+ 信号。 **电容屏的 I2C 时钟信号** 使用电容屏时为电容屏上的 I2C 时钟信号。
4	Y- CTP_RST# GINT[34]	IO	**电阻屏 Y- 信号** 使用电阻屏时，此为电阻屏 Y- 信号。 **电容屏上的复位信号** 使用电容屏时为电容屏上的复位信号。 此信号也可以作为 GPIO 或是中断输入信号 GINT[34] 使用。
2	X- CTP_INT GINT[33]	IO	**电阻屏 X- 信号** 使用电阻屏时，此为电阻屏 X- 信号。 **电容屏的中断信号** 使用电容屏时为电容屏上的 I2C 中断信号。 此信号也可以作为 GPIO 或是中断输入信号 GINT[33] 使用。

2.3.5 USB 控制信号

表格 2-5：LT268C USB 控制信号

Pin#	管脚名称	I/O	功 能 说 明
21	DP	IO	**USB 数据端 (Positive)** 此为 USB 数据端 DP 的信号。LT268C 可以透过 USB 接口更新内部 MCU 程序及外接 SPI Flash 内的数据。
20	DM	IO	**USB 数据端 (Negative)** 此为 USB 数据端 DM 的信号。
18	OTG_ID	I	**USB 检测** 此 ID 信号用来指示 Mini USB 插头的 ID 状态，从而判断为主设备或从设备。

2.3.6 ADC 与 DAC 信号

表格 2-6：LT268C ADC 与 DAC 信号

Pin#	管脚名称	I/O	功 能 说 明
59, 57, 58	AIN[2:0]	I	**ADC 模拟输入信号** 此内部 ADC 的模拟信号输入管脚。 AIN[0] 信号与使用电阻屏时的 Y1+信号共享。 AIN[1] 信号与使用电阻屏时的 X1+信号共享。
48	ADC_CH_IN[0] GINT[26] DB[10]	IO	**ADC 模拟输入信号** 当只使用 8Bit MCU 屏接口时，此信号可以作为 ADC 输入 ADC_CH_IN[0] 使用。此信号也与 GINT[26] 或是 MCU 屏数据信号 DB[10] 共享。
50	ADC_CH_IN[1] GINT[27] DB[11]	IO	**ADC 模拟输入信号** 当只使用 8Bit MCU 屏接口时，此信号可以作为 ADC 输入 ADC_CH_IN[1] 使用。此信号也与 GINT[27] 或是 MCU 屏数据信号 DB[11] 共享。
47	ADC_CH_IN[2] GINT[28] DB[12]	IO	**ADC 模拟输入信号** 当只使用 8Bit MCU 屏接口时，此信号可以作为 ADC 输入 ADC_CH_IN[2] 使用。此信号也与 GINT[28] 或是 MCU 屏数据信号 DB[12] 共享。
51	ADC_CH_IN[3] GINT[29] DB[13]	IO	**ADC 模拟输入信号** 当只使用 8Bit MCU 屏接口时，此信号可以作为 ADC 输入 ADC_CH_IN[3] 使用。此信号也与 GINT[29] 或是 MCU 屏数据信号 DB[13] 共享。
28	ADC_CH_IN[6] GINT[24] DB[8]	IO	**ADC 模拟输入信号** 当只使用 8Bit MCU 屏接口时，此信号可以作为 ADC 输入 ADC_CH_IN[6] 使用。此信号也与 GINT[24] 或是 MCU 屏数据信号 DB[8] 共享。
29	ADC_CH_IN[7] GINT[25] DB[9]	IO	**ADC 模拟输入信号** 当只使用 8Bit MCU 屏接口时，此信号可以作为 ADC 输入 ADC_CH_IN[7] 使用。此信号也与 GINT[25] 或是 MCU 屏数据信号 DB[9] 共享。
61	DAC_OUT	O	**模拟输出信号** 此为 DAC 信号输出，可作为音频输出或预留做其他控制信号使用。

2.3.7 其他控制信号

表格 2-7：LT268C 其他控制信号

Pin#	管脚名称	I/O	功 能 说 明
33	PWM[0] GINT[12]	IO	**PWM 0 输出** 此为 LT268C 内部 PWM 0 输出。 也可以作为 GPIO 或是中断输入信号 GINT[12] 使用。
32	PWM[1] LCD_RST# GINT[13]	IO	**PWM 1 输出** 此为 LT268C 内部 PWM 1 输出。 此信号与 MCU 屏上的复位信号 LCD_RST#共享。也可以作为 GPIO 或是中断输入信号 GINT[13] 使用。
31	PWM[2] BUSY GINT[14]	IO	**PWM 2 输出** 此为 LT268C 内部 PWM 2 输出。 **忙状态输出** 此脚可以作为串口屏的忙状态 BUSY 输出，主控端可以由此信号了解串口屏是否为忙状态。 也可以作为 GPIO 或是中断输入信号 GINT[14] 使用。
30	PWM[3] LCD_BL GINT[15]	IO	**PWM 3 输出** 此为 LT268C 内部 PWM 3 输出。 **背光控制信号输出** 此信号与串口屏的背光控制信号 LCD_BL 共享。也可以作为 GPIO 或是中断输入信号 GINT[15] 使用。
68	AMPEN GINT[32]	IO	**音频功放控制输出** 此信号为音频功放的使能控制输出。 此信号也可以作为 GPIO 或是中断输入信号 GINT[32] 使用。
63	WAKEUP SD_IN	I	**唤醒输入信号** 此为 LT268C 的唤醒输入。 **SD 卡（SPI 模式）侦测信号** 使用 SD 卡（SPI 模式）时，此信号可以作为 SD 卡（SPI 模式）侦测信号。
62	RST#	I	**复位输入信号** 当 RST# = 0 时，将对内部 MCU 产生复位动作，除了少数由 POR 才能复位的寄存器外，大多数由 MCU 控制的寄存器将回复到默认值。

2.3.8 晶振与电源信号

表格 2-8：LT268C 晶振与电源信号

Pin#	管脚名称	I/O	功 能 说 明
16	32K_XI	I	**RTC 晶振输入** 此管脚连接至外部 32.768Khz 晶振。
15	32K_XO	O	**RTC 晶振输出** 此管脚连接至外部 32.768Khz 晶振。
7	XTAL	I	**USB 时钟信号** 此管脚连接至外部 12Mhz 晶振。
6	EXTAL	O	**USB 时钟信号** 此管脚连接至外部 12Mhz 晶振。
64	VCC5	PWR	**3.3V~5V 电源输入（System)**
17, 56, 60	AVDD	PWR	**3.3V ADC/DAC 电源输入**
14	VBAT	PWR	**2.2V~3.3V RTC 电池电源输入**
52, 66	VDD33	PWR	**3.3V 电源输入**
67	LDO1_V11	PWR	**1.1V 内核电源输出#1 (Flash)** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
19	LDO2_V11	PWR	**1.1V 内核电源输出#2 (USB PHY)** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
65	LDO3_V12	PWR	**1.2V 内核电源输出（Core)** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
0	VSS Thermal Pad	-	**GND 接地散热焊盘** 此为 IC 的背部散热焊盘与 VSS，必须直接接地（GND)。

2.4 LT268D 管脚信号说明

2.4.1 MCU 屏接口信号

表格 2-9：LT268D MCU 屏接口信号

Pin#	管脚名称	I/O	功 能 说 明
48, 49, 20, 19, 17, 15, 14, 13	DB[7:0]	IO	**8/16Bit 并口的 MCU 屏数据信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268D 对外部 8/16Bit MCU 屏的数据 Bit7~Bit0 传输信号。
72, 68, 74, 70, 73, 71, 43, 41	DB[15:8]	IO	**16Bit 并口的 MCU 屏数据信号** 当使用 16Bit 并口的 MCU 屏时，这些信号为 LT268D 对外部 16Bit MCU 屏的数据 Bit15~Bit8 传输信号。
36	CS# SPI2_MISO	IO	**8/16Bit 并口的 MCU 屏芯片选择信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268D 对外部 8/16Bit MCU 屏的片选控制信号。 **串口的 MCU 屏上的 SPI 数据输入信号** 当使用 SPI 的 MCU 屏接口时，此信号为 LT268D 由外部 MCU 屏的读取数据输入。
35	WR# SPI2_MOSI	O	**8/16Bit 并口的 MCU 屏写入控制信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268D 对外部 8/16Bit MCU 屏的数据写入控制信号。 **串口的 MCU 屏上输出到 SPI 的数据信号** 当使用 SPI 的 MCU 屏接口时，此信号为 LT268D 输出数据到 MCU 屏。
34	RD# SPI2_CK	O	**8/16Bit 并口的 MCU 屏读取控制信号** 当使用 8/16Bit 并口的 MCU 屏时，这些信号为 LT268D 对外部 8/16Bit MCU 屏的数据读取控制信号。 **串口的 MCU 屏上的 SPI 串行时钟信号** 当使用 SPI 的 MCU 屏接口时，此信号是提供 SPI 屏的时钟信号输出。
33	RS SPI2_CS#	O	**8/16Bit 并口的 MCU 屏上的 RS 或 A0 信号** 当使用 8/16Bit 并口的 MCU 屏时, 使用 SPI 屏时, 为 LT268D 输出到 SPI 屏的 RS 信号。 **串口的 MCU 屏上的 SPI 芯片选择信号** 当使用 SPI 的 MCU 屏接口时，此信号是提供 SPI 屏的片选信号输出。
46	LCD_RST#	O	**MCU 屏上的复位输出信号** 此管脚是提供 8/16Bit MCU 屏或是 SPI 屏的复位信号, 当 LCD_RST# = 0 时，将对 TFT LCD 屏产生复位动作。

2.4.2 SPI Flash 接口信号

表格 2-10：LT268D SPI Flash 接口信号

Pin#	管脚名称	I/O	功 能 说 明
60	SPI1_CK	O	**SPI#1 元件的时钟信号** SPI1 主要是接到外部 SPI 元件。此管脚是提供到外部 SPI 元件的时钟信号输出。
62	SPI1_CS#	O	**SPI#1 元件的芯片选择信号** 此信号为 LT268D 对外部 SPI 元件的片选输出。
63	SPI1_MISO	I	**接收 SPI#1 元件的数据输入信号** 此信号为 LT268D 由外部 SPI 元件的读取数据输入。
61	SPI1_MOSI	O	**输出到 SPI#1 元件的数据信号** 此信号为 LT268D 输出数据到外部 SPI 元件。
77	SPI3_CK SD_CK	O	**SPI#3 元件的时钟信号** SPI3 主要是接到外部 SPI 元件。此管脚是提供到外部 SPI 元件的时钟信号输出。 **SD 卡（SPI 模式）时钟信号** 此信号也可以作为 SD 卡（SPI 模式）时钟信号输出
7	SPI3_CS# SD_CS#	O	**SPI 组件的芯片选择信号** 此信号为 LT268D 对外部 SPI 组件的片选输出。 **SD 卡（SPI 模式）片选信号** 此信号也可以作为 SD 卡（SPI 模式）的片选输出
79	SPI3_MISO SD_MISO	I	**接收 SPI 组件的数据输入信号** 此信号为 LT268D 由外部 SPI 组件的读取数据输入。 **SD 卡（SPI 模式）数据输入信号** 此信号也可以作为 SD 卡（SPI 模式）数据输入信号。
78	SPI3_MOSI SD_MOSI	O	**输出到 SPI 组件的数据信号** 此信号为 LT268D 输出数据到外部 SPI 组件。 **SD 卡（SPI 模式）数据输出信号** 此信号也可以作为 SD 卡（SPI 模式）数据输出信号。
53	SPI4_CK	O	**QSPI Flash 串行时钟信号** SPI4 主要是接到外部 QSPI Flash，用来读取储存在 QSPI Flash 的图片、动画、字库等信息。此管脚是 QSPI Falsh 的时钟信号输出。
54	SPI4_CS#	O	**QSPI Flash 的芯片选择信号** 此信号为 LT268D 对外部 QSPI 组件的片选输出。
55, 51, 52, 50	SPI4_D[3:0]	I/O	**QSPI Flash 的数据输出入信号** 这些信号为 LT268D 对外部 QSPI 组件读取或写入的数据信号。

2.4.3 Uart 串口与 Uart（SCI）扩展信号

表格 2-11：LT268D Uart（SCI）扩展信号

Pin#	管脚名称	I/O	功 能 说 明
64	RXD[1]	I	**串口通信(Uart) #1 接收数据输入** 这是第 1 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。
65	TXD[1]	O	**串口通信(Uart) #1 发送数据输出** 这是第 1 组 Uart 串行通信接口的数据输出口，用来传送信号到其他组件的 RX。
38	RXD[2]	I	**串口通信(Uart) #2 接收数据输入** 这是第 2 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。
37	TXD[2]	O	**串口通信(Uart) #2 发送数据输出** 这是第 2 组 Uart 串行通信接口的数据输出口，用来传送信号到其他组件的 RX。
57	RXD[3]	I	**串口通信(Uart) #3 接收数据输入** 这是第 3 组 Uart 串行通信接口的数据输入口，用来接收其他组件的 TX 送来信号。 使用串口屏时，这是 Uart 串行通信的数据输入口，用来接收主控端 MCU 的 TX 送来指令信号。
56	TXD[3]	O	**串口通信(Uart) #3 发送数据输出** 这是第 3 组 Uart 串行通信接口的数据输出口，用来传送信号到其他组件的 RX。 使用串口屏时，这是 Uart 串行通信的数据输出口，用来传送指令信号到主控端 MCU 的 RX。

2.4.4 触控屏信号

表格 2-12：LT268D 触控屏信号

Pin#	管脚名称	I/O	功 能 说 明
81	X1+ AIN[1]	I	**电阻屏 X+ 信号** 使用电阻屏时，此为电阻屏 X+ 信号，使用时 X1+ 和 X2+ 这 2 根信号必须同时接到电阻屏的 X+ 信号。 不使用电阻屏时此信号也可以作为模拟输入信号 AIN[1]。
82	Y1+ AIN[0]	I	**电阻屏 Y+ 信号** 使用电阻屏时，此为电阻屏 Y+ 信号，使用时 Y1+ 和 Y2+ 这 2 根信号必须同时接到电阻屏的 Y+ 信号。 不使用电阻屏时此信号也可以作为模拟输入信号 AIN[0]。

Pin#	管脚名称	I/O	功 能 说 明
3	X2+ I2C_SDA1	IO	**电阻屏 X+ 信号** 使用电阻屏时，此为电阻屏 X+ 信号，使用时 X1+ 和 X2+ 这 2 根信号必须同时接到电阻屏的 X+ 信号。 **电容屏的 I2C 数据信号** 使用电容屏时为电容屏上的 I2C 数据信号。
5	Y2+ I2C_SCL1	IO	**电阻屏 Y+ 信号** 使用电阻屏时，此为电阻屏 Y+ 信号，使用时 Y1+ 和 Y2+ 这 2 根信号必须同时接到电阻屏的 Y+ 信号。 **电容屏的 I2C 时钟信号** 使用电容屏时为电容屏上的 I2C 时钟信号。
6	Y- CTP_RST# GINT[34]	IO	**电阻屏 Y- 信号** 使用电阻屏时，此为电阻屏 Y- 信号。 **电容屏上的复位信号** 使用电容屏时为电容屏上的复位信号。 此信号也可以作为 GPIO 或是中断输入信号 GINT[34] 使用。
4	X- CTP_INT GINT[33]	IO	**电阻屏 X- 信号** 使用电阻屏时，此为电阻屏 X- 信号。 **电容屏的中断信号** 使用电容屏时为电容屏上的 I2C 中断信号。 此信号也可以作为 GPIO 或是中断输入信号 GINT[33] 使用。

2.4.5 USB 控制信号

表格 2-13：LT268D USB 控制信号

Pin#	管脚名称	I/O	功 能 说 明
30	DP	IO	**USB 数据端（Positive）** 此为 USB 数据端 DP 的信号。LT268D 可以透过 USB 接口更新内部 MCU 程序及外接 SPI Flash 内的数据。
29	DM	IO	**USB 数据端（Negative）** 此为 USB 数据端 DM 的信号。
26	OTG_ID	I	**USB OTG 检测** 此 ID 信号用来指示 Mini USB 插头的 ID 状态，从而判断为主设备或从设备。当检测到 USB_ID 信号为低时，作为 Host（主机）用，当检测到 USB_ID 信号为高时，表示作为 Slave（从机）用。
27	OTG_VBUS	I	**USB VBUS 检测信号** 作为 Host（主机）用时接到 VDD，作为 Slave（从机）用时接到 GND。

2.4.6 I2C 控制信号

表格 2-14：LT268D I2C 控制信号

Pin#	管脚名称	I/O	功 能 说 明
3	I2C_SDA1 X2+	IO	**第 1 组 I2C 数据信号** 使用电容屏时为电容屏上的 I2C 数据信号。使用电阻屏时，此为电阻屏 X2+ 信号。
5	I2C_SCL1 Y2+	IO	**第 1 组 I2C 时钟信号** 使用电容屏时为电容屏上的 I2C 时钟信号。使用电阻屏时，此为电阻屏 X2+ 信号。
40	I2C_SDA2	IO	**第 2 组 I2C 数据信号** I2C 数据信号，也可以作为 GPIO 信号使用。
39	I2C_SCL2	IO	**第 2 组 I2C 时钟信号** I2C 时钟信号，也可以作为 GPIO 信号使用。
59	I2C_SDA3	IO	**第 3 组 I2C 数据信号** I2C 数据信号，也可以作为 GPIO 信号使用。
58	I2C_SCL3	IO	**第 3 组 I2C 时钟信号** I2C 时钟信号，也可以作为 GPIO 信号使用。

2.4.7 GPIO 与 INT 中断信号

表格 2-15：LT268D GPIO 与 INT 中断输入信号

Pin#	管脚名称	I/O	功 能 说 明
77	GINT[9] SPI3_CK	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 SPI3 的时钟信号输出共享。
79	GINT[10] SPI3_MISO	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 SPI3 的读取数据输入信号共享。
78	GINT[11] SPI3_MOSI	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 SPI3 的数据输出信号共享。
47	GINT[12] PWM[0]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 PWM[0]输出信号共享。
32	GINT[22] ADC_CH_IN[4]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号共享。
42	GINT[23] ADC_CH_IN[5]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号共享。

Pin#	管脚名称	I/O	功 能 说 明
41	GINT[24] ADC_CH_IN[6] DB[8]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号、16Bit 并口的 MCU 屏数据信号共享。
43	GINT[25] ADC_CH_IN[7] DB[9]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号、16Bit 并口的 MCU 屏数据信号共享。
71	GINT[26] ADC_CH_IN[0] DB[10]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号、16Bit 并口的 MCU 屏数据信号共享。
73	GINT[27] ADC_CH_IN[1] DB[11]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号、16Bit 并口的 MCU 屏数据信号共享。
70	GINT[28] ADC_CH_IN[2] DB[12]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号、16Bit 并口的 MCU 屏数据信号共享。
74	GINT[29] ADC_CH_IN[3] DB[13]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与 ADC 输入信号、16Bit 并口的 MCU 屏数据信号共享。
2	GINT[32] AMPEN	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与音频功放的使能控制输出信号共享。
4	GINT[33] CTP_INT X-	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与电容屏上的 I2C 中断输入信号或是电阻屏 X- 信号共享。
6	GINT[34] CTP_RST# Y-	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。此管脚与电容屏上的复位信号或是电阻屏 Y- 信号共享。
8	GINT[35]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。
18	GINT[36]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。
10	GINT[37]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。
12	GINT[38]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。
16	GINT[39]	IO	**GPIO 或是中断输入信号** 此为 GPIO 或是作为中断输入信号使用。
1, 100, 99, 90, 89, 88	GPIO[5:0]	I	**GPIO 输出入信号** 这些信号只能作为 GPIO 输出入信号使用。

2.4.8 ADC / DAC 信号

表格 2-16：LT268D ADC / DAC 信号

Pin#	管脚名称	I/O	功 能 说 明
83, 81, 82	AIN[2:0]	I	**ADC 模拟输入信号** 此内部 ADC 的模拟信号输入管脚。 AIN[0] 信号与使用电阻屏时的 Y1+信号共享。 AIN[1] 信号与使用电阻屏时的 X1+信号共享。
71	ADC_CH_IN[0] GINT[26] DB[10]	I	**选择性 ADC 模拟输入信号 [0]** 透过寄存器设置成 ADC 通道 [0] 模拟输入端口 此信号与16Bit并口的MCU 屏数据信号 DB[10] 及GINT[26]共享。
73	ADC_CH_IN[1] GINT[27] DB[11]	I	**选择性 ADC 模拟输入信号 [1]** 透过寄存器设置成 ADC 通道 [1] 模拟输入端口 此信号与16Bit并口的MCU 屏数据信号 DB[11] 及GINT[27]共享。
70	ADC_CH_IN[2] GINT[28] DB[12]	I	**选择性 ADC 模拟输入信号 [2]** 透过寄存器设置成 ADC 通道 [2] 模拟输入端口 此信号与16Bit并口的MCU 屏数据信号 DB[12] 及GINT[28]共享。
74	ADC_CH_IN[3] GINT[29] DB[13]	I	**选择性 ADC 模拟输入信号 [3]** 透过寄存器设置成 ADC 通道 [3] 模拟输入端口 此信号与16Bit并口的MCU 屏数据信号 DB[13] 及GINT[29]共享。
32	ADC_CH_IN[4] GINT[22]	I	**选择性 ADC 模拟输入信号 [4]** 透过寄存器设置成 ADC 通道 [4] 模拟输入端口 此信号与 16Bit 并口的 MCU 屏数据信号 GINT[22] 共享。
42	ADC_CH_IN[5] GINT[23]	I	**选择性 ADC 模拟输入信号 [5]** 透过寄存器设置成 ADC 通道 [5] 模拟输入端口 此信号与 16Bit 并口的 MCU 屏数据信号 GINT[23] 共享。
41	ADC_CH_IN[6] GINT[24] DB[8]	I	**选择性 ADC 模拟输入信号 [6]** 透过寄存器设置成 ADC 通道 [6] 模拟输入端口 此信号与 16Bit 并口的 MCU 屏数据信号 DB[8] 及 GINT[24]共享。
43	ADC_CH_IN[7] GINT[25] DB[9]	I	**选择性 ADC 模拟输入信号 [7]** 透过寄存器设置成 ADC 通道 [7] 模拟输入端口 此信号与 16Bit 并口的 MCU 屏数据信号 DB[9] 及 GINT[25]共享。
86	DAC_OUT	O	**模拟输出信号** 此为 DAC 信号输出，可作为音频输出或预留做其他控制信号使用。

2.4.9 其他控制信号

表格 2-17：LT268D 其他控制信号

Pin#	管脚名称	I/O	功 能 说 明
47	PWM[0] GINT[12]	IO	**PWM 0 输出** 此为 LT268D 内部 PWM 0 输出。 也可以作为 GPIO 或是中断输入信号 GINT[12] 使用。
45	PWM[2] BUSY GINT[14]	IO	**PWM 2 输出** 此为 LT268D 内部 PWM 2 输出。 **忙状态输出** 此脚可以作为串口屏的忙状态 BUSY 输出，主控端可以由此信号了解串口屏是否为忙状态。
2	AMPEN GINT[32]	IO	**音频功放控制输出** 此信号为音频功放的使能控制输出。 此信号也可以作为 GPIO 或是中断输入信号 GINT[32] 使用。
93	WAKEUP SD_IN	I	**唤醒输入信号** 此为 LT268D 的唤醒输入。 **SD 卡（SPI 模式）侦测信号** 使用 SD 卡（SPI 模式）时，此信号可以作为 SD 卡（SPI 模式）侦测信号。
92	RST#	I	**复位输入信号** 当 RST# = 0 时，将对内部 MCU 产生复位动作，除了少数由 POR 才能复位的寄存器外，大多数由 MCU 控制的寄存器将回复到默认值。
69	CLKOUT	O	**时钟输出信号** 此为 LT268D 的时钟信号输出，可提供外部组件使用。
76	RSTOUT	O	**复位输出信号** 此为 LT268D 的复位信号输出，可作为外部组件同步复位使用。
44	LCD_BL	O	**串口屏的背光控制信号**
94	USB_DET	I	**USB DET 唤醒输入**

2.4.10 晶振与电源信号

表格 2-18：LT268D 晶振与电源信号

Pin#	管脚名称	I/O	功 能 说 明
23	32K_XI	I	**RTC 晶振输入** 此管脚连接至外部 32.768Khz 晶振。
22	32K_XO	O	**RTC 晶振输出** 此管脚连接至外部 32.768Khz 晶振。

Pin#	管脚名称	I/O	功 能 说 明
11	XTAL	I	**USB 时钟信号** 此管脚连接至外部 12Mhz 晶振。
9	EXTAL	O	**USB 时钟信号** 此管脚连接至外部 12Mhz 晶振。
95	VCC35	PWR	**3.3V~5V 电源输入（System)**
25, 80, 84	AVDD	PWR	**3.3V ADC/DAC 电源输入**
21	VBAT	PWR	**2.2V~3.3V RTC 电池电源输入**
67, 75, 85, 97	VDD33	PWR	**3.3V 电源输入**
91	VDDIO	PWR	**IO 电源输出** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
98	LDO1_V11	PWR	**1.1V 内核电源输出#1 (Flash)** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
28	LDO2_V11	PWR	**1.1V 内核电源输出#2 (USB PHY)** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
96	LDO3_V12	PWR	**1.2V 内核电源输出（Core)** 此管脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
24, 31, 66, 87	VSS	PWR	**GND 接地**

2.5 LT269 管脚信号说明

2.5.1 Uart 串口信号

表格 2-19：LT269 Uart 串口信号

Pin#	引脚名称	I/O	功 能 说 明
41	M_RXD	I	**串口通信(Uart) 接收数据输入** 这是 Uart 串行通信接口的数据输入口，用来接收主控端 MCU 的 TX 送来指令信号。实际使用到内部的 SCI#3 的串口通信模块。
40	M_TXD	O	**串口通信(Uart) 发送数据输出** 这是 Uart 串行通信接口的数据输出口，用来传送指令信号到主控端 MCU 的 RX。实际使用到内部的 SCI#3 的串口通信模块。
32	BUSY PWM[2]	O	**忙状态输出** 此脚为串口屏的忙状态输出，主控端可以由此信号了解串口屏是否为忙状态。 此脚也可以做作为 PWM[2] 输出。

2.5.2 MCU 屏接口信号

表格 2-20：LT269 MCU 屏接口信号

Pin#	引脚名称	I/O	功 能 说 明
45, 42, 47, 43, 46, 44, 30, 29	DB[7:0]	IO	**8Bit 的 MCU 屏数据信号** 这些信号为 LT269 对外部 8Bit MCU 屏的数据传输信号。
35	LCD_RS		**8Bit MCU 屏上的 RS/A0 信号** 使用 8Bit MCU 屏时，为 LT269 输出到 TFT 屏的 RS/A0 信号。
34	LCD_CS#		**8Bit MCU 屏的芯片选择信号** 使用 8Bit MCU 屏时，为 LT269 输出到 TFT 屏的片选信号。
33	LCD_RST#	O	**复位输出信号** 此引脚是提供 SPI 屏或是 8Bit MCU 屏的复位信号，当 LCD_RST# = 0 时，将对 TFT 屏产生复位动作。

Pin#	引脚名称	I/O	功 能 说 明
2	LCD_RD# P_MOSI	O	**8Bit 的 MCU 屏读取控制信号** 这些信号为 LT269 对外部 8Bit MCU 屏的数据读取信号。 **输出到 SPI 屏的数据信号** 使用 SPI 屏时，此信号是 LT269 输出到 TFT 屏的数据信号。对 SPI 屏的串口通讯（P_MOSI、P_MISO、P_SCK、P_SCS#）实际使用到的是内部 SPI#3 的串口模块。
1	LCD_WR# P_SCK	O	**8Bit 的 MCU 屏写入控制信号** 这些信号为 LT269 对外部 8Bit MCU 屏的数据写入信号。 **SPI 屏上的串行时钟信号** 使用 SPI 屏时，此信号是 SPI 的时钟信号输出。
18	P_SCS#	O	**输出到 SPI 屏的芯片选择信号** 此信号为 LT269 对外部 SPI 屏的的片选输出。
3	P_MISO	I	**SPI 屏的数据信号输入信号** 此信号为 LT269 由外部 SPI 屏的读取数据输入。
31	LCD_BL PWM[3]	O	**串口屏的背光控制信号** 此信号与 PWM[3] 输出信号共享。

2.5.3 SPI Flash 接口信号

表格 2-21：LT269 SPI Flash 接口信号

Pin#	引脚名称	I/O	功 能 说 明
39	F_SCS#	O	**SPI Flash 的芯片选择信号** 此信号为 LT269 对外部 SPI Flash 的片选输出。这组 SPI Flash 接口（F_SCS#、F_MOSI、F_MISO、F_SCK）实际是使用到内部的 SPI#4 的串口模块。
36	F_MOSI	I	**SPI Flash 的 D0 数据信号** 此信号为 LT269 对外部 SPI Flash 数据写入信号。
37	F_MISO	O	**SPI Flash 的 D1 数据信号** 此信号为 LT269 对外部 SPI Flash 数据读取信号，用来读取存在 Flash 内的图片、动画等信息。
38	F_SCK	O	**SPI 串行时钟信号** 此引脚是提供到外部 SPI Flash 的时钟信号输出。

2.5.4 触控屏信号

表格 2-22：LT269 触控屏信号

Pin#	引脚名称	I/O	功 能 说 明
19	Y-	I	**电阻屏触控 Y- 信号**
17	X-	I	**电阻屏触控 X- 信号**
6, 16	Y1+ Y2+	I	**电阻屏触控 Y+ 信号** 使用电阻屏时此 2 根信号必须连接在一起。
5, 15	X1+ X2+	I	**电阻屏触控 X+ 信号** 使用时此 2 根信号必须连接在一起。
16	I2C_SCL	O	**I2C 时钟信号** 使用触控电容屏时为电容屏上的 I2C 时钟信号。此信号与电阻屏 Y2+的信号共享。
15	I2C_SDA	IO	**I2C 数据信号** 使用触控电容屏时为电容屏上的 I2C 数据信号。此信号与电阻屏 X2+的信号共享。
19	CTP_RST#	O	**触控电容屏上的复位信号** 使用触控电容屏时为电容屏上的复位信号。此信号与电阻屏 Y-信号共享。
17	CTP_INT	I	**触控电容屏上的中断信号** 使用触控电容屏时为电容屏上的 I2C 中断信号。此信号与电阻屏 X-信号共享。

2.5.5 USB 控制信号

表格 2-23：LT269 USB 控制信号

Pin#	引脚名称	I/O	功 能 说 明
28	DP	IO	**USB 数据端 (Positive)** 此为 USB 数据端 DP 的信号。LT269 可以透过 USB 接口更新内部 MCU 程序及外接 SPI Flash 内的数据。
27	DM	IO	**USB 数据端 (Negative)** 此为 USB 数据端 DM 的信号。

2.5.6 SCI / SPI 扩展信号

表格 2-24：LT269 SCI / SPI 扩展信号

Pin#	引脚名称	I/O	功 能 说 明
20	RXD[1]	I	**串口通信(Uart) #1 接收数据输入** 这是串行通信接口的数据输入口，用来接收其他串口组件的 TX 送来信号。
21	TXD[1]	O	**串口通信(Uart) #1 发送数据输出** 这是串行通信接口的数据输出口，用来传送信号到其他串口组件的 RX 接口。
18	SD_SCS# P_SCS#	O	**SD 卡（SPI 模式）片选信号** 此信号用在 SPI 模式的 SD 卡，作为片选信号。
1	SD_SCK P_SCK	O	**SD 卡（SPI 模式）时钟信号** 此信号与输出到 MCU 屏上 SPI 的串行时钟信号 P_SCK 共享。
2	SD_MOSI P_MOSI	O	**SD 卡（SPI 模式）数据输出信号** 此信号与输出到 MCU 屏上 SPI 的数据信号 P_MOSI 共享。
3	SD_MISO P_MISO GINT[10]	I	**SD 卡（SPI 模式）数据输入信号** 此信号与 MCU 屏上的 SPI 数据输入信号 P_MISO、GINT[10] 共享。
10	SD_IN WAKEUP	I	**SD 卡（SPI 模式）侦测信号** 此信号与唤醒信号 WAKEUP 共享。

2.5.7 其他控制信号

表格 2-25：LT269 其他控制信号

Pin#	引脚名称	I/O	功 能 说 明
9	RST#	I	**复位输入信号** 当 RST# = 0 时，将对内部 MCU 产生复位动作，除了少数由 POR 才能复位的寄存器外，大多数由 MCU 控制的寄存器将回复到默认值。
32	PWM[2] GINT[14] BUSY	O	**PWM 2 输出** LT269 内部 PWM[2] 输出。 此信号与串口屏的 BUSY 忙状态输出、GINT[14] 共享。

Pin#	引脚名称	I/O	功 能 说 明
31	PWM[3] GINT[15] LCD_BL	O	**PWM 3 输出** LT269 内部 PWM[3] 输出。 此信号与串口屏的背光控制 LCD_BL 信号、GINT[15] 共享。
5, 6	ADC[1:0]	I	**模拟输入** 此为仿真信号输入，当使用电阻触控屏时为 X1+及 Y1+信号。
8	DAC_OUT	O	**模拟输出** 此为仿真信号输出，可作为音频输出或预留做其他控制信号使用。
45, 42, 47, 43, 46, 44, 30, 29	GINT[31:24]	IO	**通用输出输入接口** 当不使用 8Bit MCU 屏时，这些可做为通用输出输入，或是中断输入接口。
35	GINT[7] LCD_RS	IO	**通用输出输入接口** 当不使用 8Bit MCU 屏时，可做为通用输出输入，或是中断输入接口
3	AMPEN GINT[10] SD_MISO P_MISO	IO	**音频功放控制输出** 当不使用 SPI 屏时，此信号为音频功放的使能控制输出。此信号也可以作为 GPIO 或是中断输入信号 GINT[32] 使用。
34	GINT[12] PWM[0] LCD_CS#	IO	**通用输出输入接口 / PWM[0]输出** 当不使用 8Bit MCU 屏时，可做为 PWM[0]、通用输出输入，或是中断输入接口
33	GINT[13] PWM[1] LCD_RST#	IO	**通用输出输入接口 / PWM[1]输出** 当不使用 8Bit MCU 屏时，可做为 PWM[1]、通用输出输入，或是中断输入接口
17	GINT[33] X- CTP_INT	IO	**通用输出输入接口** 当不使用触控屏时，这根脚可做为通用输出输入，或是中断输入接口。
19	GINT[34] Y- CTP_RST#	IO	**通用输出输入接口** 当不使用触控屏时，这根脚可做为通用输出输入，或是中断输入接口 GINT[34] 使用。

2.5.8 晶振与电源信号

表格 2-26：LT269 晶振与电源信号

Pin#	引脚名称	I/O	功 能 说 明
24	32K_XI	I	**RTC 晶振输入** 此管脚连接至外部 32.768Khz 晶振。
23	32K_XO	O	**RTC 晶振输出** 此管脚连接至外部 32.768Khz 晶振。
11	VCC35	PWR	**3.3V/5V 电源输入（System)**
4, 7, 25	AVDD	PWR	**3.3V ADC/DAC/USB 电源输入**
22	VBAT	PWR	**2.7V~3.6V 电池电源输入**
13, 48	VDD33	PWR	**3.3V 电源输入**
14	LDO1_V11	PWR	**1.1V 内核电源输出#1 (Flash)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
26	LDO2_V11	PWR	**1.1V 内核电源输出#2 (USB PHY)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
12	LDO3_V12	PWR	**1.2V 内核电源输出（Core)** 此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
-	Thermal Pad	-	**GND 接地散热焊盘** IC 的背部散热焊盘直接接地(VSS/GND)。

2.6 管脚属性汇总

表格 2-27：管脚属性表注释

项 目	缩 写	定 义
管脚类型	S	电源管脚
	I/O	输入/输出管脚
	I	仅输入管脚
	O	仅输出管脚
	ANA	模拟管脚
输出方式	ST	标准 CMOS
	OD	开漏(Open-Drain)
默认状态	I	输入
	O	输出
	PU	上拉
	PD	下拉
	HIZ	高阻
复用/默认功能	GPIO	通过外设模块寄存器设置
	EPORT	通过 EPORT 模块寄存器设置，支持输入中断功能

2.6.1 LT268B 管脚属性

表格 2-28：管脚属性表

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN48)
ADC_VDD	-	S	-	ADC 电压输入	I	1
X+ / ADC1	-	A	-	ADC 模拟通道 2	I	2
Y+ / ADC0	-	A	-	ADC 模拟通道 1	I	3
DAC_VDD	-	S	-	DAC 参考电压	I	4
S_OUT	-	A	-	DAC 模拟输出	O	5
RST#	-	I	-	POR 复位	I PU	6
SD_IN	-	I	-	低功耗唤醒	I PD	7
VCC	-	S	-	供电电压输入	I	8
LDO3_V12	-	S	-	核心电压输出	O	9
VDD33	-	S	-	IO/模拟电压输出	I	10
LDO1_V11	-	S	-	外部 FLASH 电压输出	O	11
X+ / I2C_SDA	-	IO	ST/OD	作 IO 口 / I2C 数据	I PU	12
Y+ / I2C_SCL	-	IO	ST/OD	作 IO 口 / I2C 时钟	I PU	13

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN48)
DB7 / SPI3_SCS#	-	IO	ST/OD	作 8 位并口总线 IO 口/ SPI3 片选		14
LCD_WR#	gint[38]	I/O	ST	EPORT 功能	I PU	15
LCD_RS	gint[0] / rxd(swap)	IO	ST/OD	EPORT 功能	I PU	16
LCD_CS#	gint[1] / txd3(swap)	IO	ST/OD	EPORT 功能	I PU	17
INT01 / GPIO0	gint[2] / rxd3(swap)	IO	ST/OD	EPORT 功能	I PU	18
SD_CS#	gint[3] / rxd2(swap)	IO	ST/OD	EPORT 功能	I PU	19
X- / CTP_INT	gint[4] / txd(swap)	IO	ST/OD	EPORT 功能	I PU	20
Y- / CTP_RST#	gint[5] / txd2(swap)	IO	ST/OD	EPORT 功能	I PU	21
VBAT	-	S	-	RTC 电压输入	I	22
32K_XO	-	A	-	32K OSC 输出	O	23
32K_XI	-	A	-	32K OSC 输入	I	24
VCC33	-	S	-	USBPHY 供电电压输入	I	25
LDO2_V11	-	S	-	USBPHY 核心电压输出	O	26
DM	-	A	-	USB D 信号端口		27
DP	-	A	-	USB D+信号端口		28
F_SCS#	gint[8]	I/O	ST/OD	SPI2 片选	I PU	29
F_SCK	-	I/O	ST/OD	SPI2 时钟	I PU	30
F_MOSI	-	I/O	ST/OD	SPI2 主出从入数据	I PU	31
F_MISO	-	I/O	ST/OD	SPI2 主入从出数据	I PU	32
LCD_RD#	gint[5]/(swap)	I/O	ST/OD	UART2_TX 当作 IO 口	I PU	33
LCD_RST#	gint[3]/(swap)	I/O	ST/OD	UART2_RX 当作 IO 口	I PU	34
PWM1 / LCD_BL	gint[15]/pwm[3]/sci1_cts	I/O	ST	PWM 输出	I PU	35
PWM0 / BUSY	gint[14]/pwm[2]/sci1_rts	I/O	ST	EPORT 功能	I PU	36
M_TXD	gint[1] / (swap)	I/O	ST/OD	UART3 发送数据	I PU	37
M_RXD	gint[2] / (swap)	I/O	ST/OD	UART3 接收数据	I PU	38
DB2 / P_SCK	sck_spi5	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI1 时钟	I PU	39
DB1 / P_MOSI	spi5[0]	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI1 主出从入数据	I PU	40
DB3 / P_SCS#	ss_spi5	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI1 片选	I PU	41
DB0 / P_MISO	spi5[1]	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI1 主入从出数据	I PU	42
RXD1	gint[0] / (swap)	I/O	ST/OD	UART1 接收数据	I PU	43

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN48)
TXD1	gint[4] / (swap)	I/O	ST/OD	UART1 发送数据	I PU	44
VDD33	-	S	-	IO/模拟电压输出	O	45
DB6 / SPI3_SCK	gint[9]	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI3 时钟	I PU	46
DB5 / SPI3_MOSI	gint[11]	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI3 主出从入数据	I PU	47
DB4 / SPI3_MISO	gint[10]	I/O	ST/OD	作 8 位并口总线 IO 口 / SPI3 主入从出数据	I PU	48

2.6.2 LT268C 管脚属性

表格 2-29：LT268C 管脚属性表

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN68)
I2C_SD	X2+	IO	ST/OD	I2C 数据，或 RTP 触控屏 X+管脚，或 GPIO 口	I PU	1
CPT_INT	X-，GINT[33]	IO	ST/OD	CTP 触控屏中断输入，或 RTP 触控屏 X-管脚，或作中断 GPIO 口	I PU	2
I2C_CK	Y2+	IO	ST/OD	I2C 时钟，或 RTP 触控屏 Y+管脚，或 GPIO 口	I PU	3
CPT_RST#	Y-，GINT[34]	IO	ST	CTP 触控屏复位输出，或 RTP 触控屏 Y-管脚，或作中断 GPIO 口	I PU	4
SPI3_CS#	SD_CS#	IO	ST/OD	SPI3 片选，或 GPIO 口	-	5
EXTAL	-	A	-	USB 时钟信号(12MHz)	O	6
XTAL	-	A	-	USB 时钟信号	I	7
DB[0]	GINT[0]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	8
DB[1]	GINT[1]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	9
DB[2]	GINT[2]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	10
DB[3]	GINT[3]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	11
DB[4]	GINT[4]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	12
DB[5]	GINT[5]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	13
VBAT	-	S	-	RTC 电压输入	I	14
32K_XO	-	A	-	32K OSC 输出	O	15
32K_XI	-	A	-	32K OSC 输入	I	16
AVDD	-	S	-	USBPHY 供电电压输入	I	17
OTG_ID	-	I	-	USB 检测	I	18
LDO2_V11	-	S	-	USBPHY 核心电压输出	O	19
DM	-	A	-	USB D 信号端口	-	20

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN68)
DP	-	A	-	USB D+信号端口	-	21
A0	SPI2_CS#	I/O	ST/OD	8/16Bit 并口的 MCU 屏上的RS/A0信号,或SPI2片选	I PU	22
RD#	SPI2_CK	I/O	ST/OD	8/16Bit 并口的 MCU 屏读取控制，或 SPI2 时钟	I PU	23
WR#	SPI2__MOSI	I/O	ST/OD	8/16Bit 并口的 MCU 屏写入控制，或 SPI2 主出从入数据	I PU	24
CS#	SPI2__MISO	I/O	ST/OD	8/16Bit 并口的 MCU 屏芯片选择，或 SPI2 主入从出数据	I PU	25
TXD[2]	GINT[5]	I/O	ST/OD	UART2_TX， 或作中断GPIO 口	I PU	26
RXD[2]	GINT[3]	I/O	ST/OD	UART2_RX， 或作中断GPIO 口	I PU	27
DB[8]	GINT[24], ADC_CH_IN[6]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作ADC 输入	I PU	28
DB[9]	GINT[25] ADC_CH_IN[7]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作ADC 输入	I PU	29
LCD_BL	PWM[3]，GINT[15]	I/O	ST	LCD 背光控制，或 PWM 输出，或作中断 GPIO 口	I PU	30
BUSY	PWM[2]，GINT[14]	I/O	ST	BUSY 输入，或 PWM 输出，或作中断 GPIO 口	I PU	31
LCD_RST#	PWM[1]，GINT[13]	I/O	ST	LCD 复位输出，或 PWM 输出，或作中断 GPIO 口	I PU	32
PWM0	GINT[12]	I/O	ST	PWM 输出，或作中断GPIO 口	I PU	33
DB[7]	GINT[7]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	34
DB[6]	GINT[6]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	35
SPI4_D[0]	GINT[18]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断GPIO 口	I PU	36

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN68)
SPI4_D[2]	GINT[16]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	37
SPI4_D[1]	GINT[19]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	38
SPI4_CK	GINT[20]	O	ST/OD	QSPI(SPI4) Flash 的时钟输出信号，或作中断 GPIO 口	I PU	39
SPI4_CS#	GINT[21]	O	ST/OD	QSPI(SPI4) Flash 的片选输出信号，或作中断 GPIO 口	I PU	40
SPI4_D[3]	GINT[17]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	41
TXD[3]	GINT[1]	I/O	ST/OD	UART3 发送数据，或作中断 GPIO 口	I PU	42
RXD[3]	GINT[2]	I/O	ST/OD	UART3 接收数据，或作中断 GPIO 口	I PU	43
RXD[1]	GINT[0]	I/O	ST/OD	UART1 发送数据，或作中断 GPIO 口	I PU	44
TXD[1]	GINT[4]	I/O	ST/OD	UART1 接收数据，或作中断 GPIO 口	I PU	45
DB[14]	GINT[30]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	46
DB[12]	GINT[28] ADC_CH_IN[2]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	47
DB[10]	GINT[26] ADC_CH_IN[0]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	48
DB[15]	GINT[31]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	49
DB[11]	GINT[27] ADC_CH_IN[1]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	50

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN68)
DB[13]	GINT[29] ADC_CH_IN[3]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	51
VDD33	-	S	-	IO/模拟电压输入	I	52
SPI3_CK	GINT[9]	I/O	ST/OD	SPI3 时钟，或作中断 GPIO 口	I PU	53
SPI3_MOSI	GINT[11]	I/O	ST/OD	SPI3 主出从入数据，或作中断 GPIO 口	I PU	54
SPI3_MISO	GINT[10]	I/O	ST/OD	SPI3 主入从出数据，或作中断 GPIO 口	I PU	55
AVDD	-	S	-	ADC 电压输入	I	56
ADC1	X+	A	-	ADC 模拟通道 1，或 RTP 触控屏 X+管脚	I	57
ADC0	Y+	A	-	ADC 模拟通道 0，RTP 触控屏 Y+管脚	I	58
ADC2	-	A	-	ADC 模拟通道 2	I	59
AVDD	-	S	-	ADC/DAC 电压	I	60
DAC_OUT	-	A	-	DAC 模拟输出	O	61
RST#	-	I	-	POR 复位	I PU	62
WAKEUP	SD_IN	I	-	低功耗唤醒，或 SD 卡（SPI 模式）侦测信号	I PD	63
VCC5V	-	S	-	3-5V 供电电压输入	I	64
LDO3_V12	-	S	-	核心电压输出	O	65
VDD33	-	S	-	IO/模拟电压输入	I	66
LDO1_V11	-	S	-	外部 FLASH 电压输出	O	67
AMPEN	GINT[32]	IO	ST/OD	音频功放控制输出，或作中断 GPIO 口	I PU	68

2.6.3 LT268D 管脚属性

表格 2-30：LT268D 管脚属性表

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (LQFP100)
GPIO[5]	-	IO	ST/OD	GPIO 口	I PU	1
AMPEN	GINT[32]	IO	ST/OD	音频功放控制输出，或作中断 GPIO 口	I PU	2
I2C_SDA1	X2+	IO	ST/OD	I2C 数据，或 RTP 触控屏 X+管脚，或 GPIO 口	I PU	3
CPT_INT	X-，GINT[33]	IO	ST/OD	CTP 触控屏中断输入，或 RTP 触控屏 X-管脚，或作中断 GPIO 口	I PU	4
I2C_SCL1	Y2+	IO	ST/OD	I2C 时钟，或 RTP 触控屏 Y+管脚，或 GPIO 口	I PU	5
CPT_RST#	Y-，GINT[34]	IO	ST	CTP 触控屏复位输出，或 RTP 触控屏 Y-管脚，或作中断 GPIO 口	I PU	6
SPI3_CS#	SD_CS#	IO	ST/OD	SPI3 片选，或 GPIO 口	-	7
GINT[35]	-	IO	ST	EPORT	I PU	8
EXTAL	-	A	-	USB 时钟信号(12MHz)	O	9
GINT[37]	-	IO	ST	EPORT	I PU	10
XTAL	-	A	-	USB 时钟信号	I	11
GINT[38]	-	IO	ST	EPORT	I PU	12
DB[0]	GINT[0]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	13
DB[1]	GINT[1]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	14
DB[2]	GINT[2]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	15
GINT[39]	-	IO	ST	EPORT	I PU	16
DB[3]	GINT[3]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	17
GINT[36]	-	IO	ST	EPORT	I PU	18
DB[4]	GINT[4]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	19

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (LQFP100)
DB[5]	GINT[5]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	20
VBAT	-	S	-	RTC 电压输入	I	21
32K_XO	-	A	-	32K OSC 输出	O	22
32K_XI	-	A	-	32K OSC 输入	I	23
VSS	-	S	-	接地	I	24
AVDD	-	S	-	USBPHY 供电电压输入	I	25
OTG_ID	-	I	-	USB 检测	I	26
OTG_VBUS	-	A	-	USB 的 VBUS 端口	I	27
LDO2_V11	-	S	-	USBPHY 核心电压输出	O	28
DM	-	A	-	USB D 信号端口	-	29
DP	-	A	-	USB D 信号端口	-	30
VSS	-	S	-	接地	I	31
GINT[22]	ADC_CH_IN[4]	IO	ST	EPORT	I PU	32
RS/A0	SPI2_CS#	I/O	ST/OD	8/16Bit 并口的 MCU 屏上的RS/A0信号,或SPI2片选	I PU	33
RD#	SPI2_CK	I/O	ST/OD	8/16Bit 并口的 MCU 屏读取控制，或 SPI2 时钟	I PU	34
WR#	SPI2__MOSI	I/O	ST/OD	8/16Bit 并口的 MCU 屏写入控制，或 SPI2 主出从入数据	I PU	35
CS#	SPI2__MISO	I/O	ST/OD	8/16Bit 并口的 MCU 屏芯片选择，或 SPI2 主入从出数据	I PU	36
TXD[2]	GINT[5]	I/O	ST/OD	UART2_TX，或作中断 GPIO 口	I PU	37
RXD[2]	GINT[3]	I/O	ST/OD	UART2_RX，或作中断 GPIO 口	I PU	38
I2C_SCL2	-	IO	ST/OD	I2C 时钟	I PU	39
I2C_SDA2	-	IO	ST/OD	I2C 数据	I PU	40
DB[8]	GINT[24], ADC_CH_IN[6]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	41
GINT[23]	ADC_CH_IN[5]	IO	ST	EPORT	I PU	42
DB[9]	GINT[25] ADC_CH_IN[7]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	43

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (LQFP100)
LCD_BL	PWM[3]，GINT[15]	I/O	ST	LCD 背光控制，或 PWM 输出，或作中断 GPIO 口	I PU	44
BUSY	PWM[2]，GINT[14]	I/O	ST	BUSY 输入，或 PWM 输出，或作中断 GPIO 口	I PU	45
LCD_RST#	PWM[1]，GINT[13]	I/O	ST	LCD 复位输出，或 PWM 输出，或作中断 GPIO 口	I PU	46
PWM0	GINT[12]	I/O	ST	PWM 输出，或作中断 GPIO 口	I PU	47
DB[7]	GINT[7]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	48
DB[6]	GINT[6]	IO	ST/OD	8/16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	49
SPI4_D[0]	GINT[18]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	50
SPI4_D[2]	GINT[16]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	51
SPI4_D[1]	GINT[19]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	52
SPI4_CK	GINT[20]	O	ST/OD	QSPI(SPI4) Flash 的时钟输出信号，或作中断 GPIO 口	I PU	53
SPI4_CS#	GINT[21]	O	ST/OD	QSPI(SPI4) Flash 的片选输出信号，或作中断 GPIO 口	I PU	54
SPI4_D[3]	GINT[17]	I/O	ST/OD	QSPI(SPI4) Flash 的数据输出入信号，或作中断 GPIO 口	I PU	55
TXD[3]	GINT[1]	I/O	ST/OD	UART3 发送数据，或作中断 GPIO 口	I PU	56
RXD[3]	GINT[2]	I/O	ST/OD	UART3 接收数据，或作中断 GPIO 口	I PU	57
I2C_SCL3	-	IO	ST/OD	I2C 时钟	I PU	58
I2C_SDA3	-	IO	ST/OD	I2C 数据	I PU	59
SPI1_CK	-	I/O	ST/OD	SPI1 时钟	I PU	60
SPI1_MOSI	-	I/O	ST/OD	SPI1 主出从入数据	I PU	61
SPI1_CS#	-	I/O	ST/OD	SPI1 片选	I PU	62

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (LQFP100)
SPI1_MISO	-	I/O	ST/OD	SPI1 主入从出数据	I PU	63
RXD[1]	GINT[0]	I/O	ST/OD	UART1 发送数据，或作中断 GPIO 口	I PU	64
TXD[1]	GINT[4]	I/O	ST/OD	UART1 接收数据，或作中断 GPIO 口	I PU	65
VSS	-	S	-	接地	I	66
VDD33	-	S	-	IO/模拟电压输入	I	67
DB[14]	GINT[30]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	68
CLKOUT	-	O	ST	时钟输出	O	69
DB[12]	GINT[28] ADC_CH_IN[2]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	70
DB[10]	GINT[26] ADC_CH_IN[0]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	71
DB[15]	GINT[31]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口	I PU	72
DB[11]	GINT[27] ADC_CH_IN[1]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	73
DB[13]	GINT[29] ADC_CH_IN[3]	IO	ST/OD	16Bit 并口的 MCU 屏数据信号，EPORT 功能可作中断 GPIO 口，或作 ADC 输入	I PU	74
VDD33	-	S	-	IO/模拟电压输入	I	75
RSTOUT	-	O	ST	复位输出	I PU	76
SPI3_CK	GINT[9]	I/O	ST/OD	SPI3 时钟，或作中断 GPIO 口	I PU	77
SPI3_MOSI	GINT[11]	I/O	ST/OD	SPI3 主出从入数据，或作中断 GPIO 口	I PU	78
SPI3_MISO	GINT[10]	I/O	ST/OD	SPI3 主入从出数据，或作中断 GPIO 口	I PU	79
AVDD	-	S	-	ADC 电压输入	I	80
AIN[1]	X1+	A	-	ADC 模拟通道 1，或 RTP 触控屏 X1+管脚	I	81

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (LQFP100)
AIN[0]	Y1+	A	-	ADC 模拟通道 0，RTP 触控屏 Y1+管脚	I	82
AIN[2]	-	A	-	ADC 模拟通道 2	I	83
AVDD	-	S	-	ADC/DAC 电压	I	84
VDD33	-	S	-	IO/模拟电压输入	I	85
DAC_OUT	-	A	-	DAC 模拟输出	O	86
VSS	-	S	-	接地	I	87
GPIO[0]	ISODAT2	IO	ST/OD	GPIO 口	I PU	88
GPIO[1]	ISOCLK2	IO	ST/OD	GPIO 口	I PU	89
GPIO[2]	ISORST2	IO	ST/OD	GPIO 口	I PU	90
VDDIO	-	S	-	IO 电压输出	O	91
RST#	-	I	-	POR 复位	I PU	92
WAKEUP	SD_IN	I	-	低功耗唤醒，或 SD 卡（SPI 模式）侦测信号	I PD	93
USBDET	-	I	-	USB 唤醒	I PD	94
VCC35V	-	S	-	3-5V 供电电压输入	I	95
LDO3_V12	-	S	-	核心电压输出	O	96
VDD33	-	S	-	IO/模拟电压输入	I	97
LDO1_V11	-	S	-	外部 FLASH 电压输出	O	98
GPIO[3]	ISODAT1	IO	ST/OD	GPIO 口	I PU	99
GPIO[4]	ISOCLK1	IO	ST/OD	GPIO 口	I PU	100

2.6.4 LT269 管脚属性

表格 2-31：LT269 管脚属性表

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN48)
LCD_WR#	P_SCK, SD_SCK	I/O	ST/OD	8Bit MCU 屏上的写入信号 / SPI 屏上的串行时钟信号（SPI3）	I PU	1
LCD_RD#	P_MOSI, SD_MOSI	I/O	ST/OD	8Bit MCU 屏上的读取信号 / 输出到 SPI 屏的数据信号（SPI3）	I PU	2
P_MISO	AMPEN, GINT[10], SD_MISO	I/O	ST/OD	SPI 屏的数据信输入信号（SPI3）/音频功放控制输出 / EPORT 功能	I PU	3
AVDD	-	S	-	ADC 电压输入	I	4
X1+	AIN[1]	A	-	电阻屏触控 X+ 信号 / ADC 模拟通道 1	I	5
Y1+	AIN[0]	A	-	电阻屏触控 Y+ 信号 / ADC 模拟通道 0	I	6
AVDD	-	S	-	ADC 参考电压	I	7
DAC_OUT	-	A	-	DAC 模拟（音频）输出	O	8
RST#	-	I	-	POR 复位	I PU	9
SD_IN	WAKEUP	I	-	SD 卡 (SPI 模式) 侦测信号 / 低功耗唤醒	I PD	10
VCC35	-	S	-	3.3V/5V 供电电压输入	I	11
LDO3_V12	-	S	-	核心电压输出	O	12
VDD33	-	S	-	IO/模拟电压输出	O	13
LDO1_V11	-	S	-	外部 FLASH 电压输出	O	14
I2C_SDA	X2+	IO	ST/OD	电容屏上的 I2C 数据信号 / 电阻屏触控 X+ 信号	I PU	15
I2C_SCL	Y2+	IO	ST/OD	电容屏上的 I2C 时钟信号 / 电阻屏触控 Y+ 信号	I PU	16
CTP_INT	X-, GINT[33]	IO	ST/OD	电容屏上的 I2C 中断信号 / 电阻屏触控 X- 信号 / EPORT 功能	I PU	17

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN48)
P_SCS#	SD_SCS#	IO	ST/OD	SPI 屏的芯片选择信号 (SPI3) / SD 卡 (SPI 模式) 片选信号	I PU	18
CTP_RST#	Y-, GINT[34]	IO	ST/OD	电容屏上的复位信号 / 电阻屏触控 Y- 信号 / EPORT 功能	I PU	19
RXD1	-	IO	ST/OD	串口通信 (Uart) #1 接收数据输入	I PU	20
TXD1	-	IO	ST/OD	串口通信 (Uart) #1 发送数据输出	I PU	21
VBAT	-	S	-	RTC 电压输入	I	22
32K_XO	-	A	-	32MOSC 输出	O	23
32K_XI	-	A	-	32MOSC 输入	I	24
AVDD	-	S	-	USBPHY 供电电压输入	I	25
LDO2_V11	-	S	-	USBPHY 核心电压输出	O	26
DM	-	A	-	USB D- 信号端口		27
DP	-	A	-	USB D+ 信号端口		28
DB[0]	GINT[24]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	29
DB[1]	GINT[25]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	30
LCD_BL	PWM[3] , GINT[15]	I/O	ST/OD	背光控制信号 / PWM[3] 输出 / EPORT 功能	I PU	31
BUSY	PWM[2] , GINT[14]	I/O	ST/OD	忙状态输出 / PWM[2] 输出 / EPORT 功能	I PU	32
LCD_RST#	PWM[1] , GINT[13]	I/O	ST/OD	8Bit MCU 屏上的复位信号 / PWM[1] 输出 / EPORT 功能	I PU	33
LCD_CS#	PWM[0], GINT[12]	I/O	ST/OD	8Bit MCU 屏的芯片选择信号 / PWM[0] 输出 / EPORT 功能	I PU	34
LCD_RS	GINT[7]	I/O	ST/OD	8Bit MCU 屏上的 RS (A0) 信号 / EPORT 功能	I PU	35

管脚名称	复用功能	管脚类型	输出方式	管脚描述	默认状态	管脚编号 (QFN48)
F_MOSI	-	I/O	ST/OD	SPI Flash 主出从入数据	I PU	36
F_MISO	-	I/O	ST/OD	SPI Flash 主入从出数据	I PU	37
F_SCK	-	I/O	ST/OD	SPI Flash 时钟信号	I PU	38
F_SCS#	-	I/O	ST/OD	SPI Flash 片选信号	I PU	39
M_TXD	-	I/O	ST/OD	串口通信发送数据输出发送数据（UART3）	I PU	40
M_RXD	-	I/O	ST/OD	串口通信接收数据输入接收数据（UART3）	I PU	41
DB[6]	GINT[30]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	42
DB[4]	GINT[28]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	43
DB[2]	GINT[26]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	44
DB[7]	GINT[31]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	45
DB[3]	GINT[27]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	46
DB[5]	GINT[29]	I/O	ST/OD	作 8 位并口总线 IO 口 / EPORT 功能	I PU	47
VDD33	-	S	-	IO/模拟电压输出	O	48

3 系统存储器映射

3.1 概述

表格 3-2：是内存映射，它包括以下：

24KB 内部只读存储器（ROM）
256KB 内部静态随机存取存储器（SRAM）
512K 内部 FLASH（用户可用到 508KB）
内部存储器映射寄存器

3.2 内存映射

表格 3-1：启动模式

Mode[1]	Mode[0]	启 动 模 式	说 明
0	0	ROM	ROOM 作为启动空间
0	1	EFLASH	EFLASH 作为启动空间
1	0	保留	保留
1	1	SPI FLASH	SPI FLASH 作为启动空间

表格 3-2：存储器映像 VS 启动模式/物理重新映像

地 址	在 ROM 中 启动/重新映射	在 EFLASH 中 启动/重新映射	在 SPI FLASH 中启动/重新映射
0xA4000000~ 0xA7FFFFFF	保留	保留	保留
0x80000000~ 0x87FFFFFF	保留	保留	保留
0x90000000~ 0x90FFFFFF	保留	保留	保留
0x94000000~ 0x97FFFFFF	保留	保留	保留
0x60000000~ 0x60FFFFFF	SPI FLASH	SPI FLASH	SPI FLASH
0x20030000~ 0x20037FFF	SRAM3	SRAM3	SRAM3
0x20020000~ 0x2002FFFF	SRAM2	SRAM2	SRAM2

地 址	在 ROM 中启动/重新映射	在 EFLASH 中启动/重新映射	在 SPI FLASH 中启动/重新映射
0x20010000~0x2001FFFF	SRAM1	SRAM1	SRAM1
0x20000000~0x2000FFFF	SRAM0	SRAM0	SRAM0
0x1FFF8000~0x1FFFFFFF	TCM(SRAMD)	TCM(SRAMD)	TCM(SRAMD)
0x10000000~0x10FFFFFF	SPIM1	SPIM1	SPIM1
0x08000000~0x0807FFFF	EFLASH	EFLASH	EFLASH
0x04000000~0x04005FFF	ROM	ROM	ROM
0x00000000~0x00FFFFFF	ROM(24K)	EFLASH(512K)	SPIM(16M)

表格 3-3：寄存器地址位置映像

0x4000_0000	4KB	AHB-IPS1	保留
0x4000_1000	4KB		CCM
0x4000_2000	4KB		RESET
0x4000_3800	4KB		EFM
0x4000_4000	4KB		CPM
0x4000_5000	4KB		WDT
0x4000_6000	4KB		保留
0x4000_7000	4KB		PIT32_0
0x4000_8000	4KB		PIT32_1
0x4000_9000	4KB		保留
0x4000_a000	4KB		EDMAC1
0x4000_b000	4KB		保留
0x4001_0000	4KB		SPI1
0x4001_1000	4KB		SPI2
0x4001_2000	4KB		SPI3
0x4001_3000	4KB		SCI1
0x4001_4000	4KB		SCI2
0x4001_5000	4KB		保留
0x4001_6000	4KB		保留
0x4001_7000	4KB		I2C

0x4001_8000	4KB		PWM
0x4001_9000	4KB		EPORT
0x4001_a000	4KB		EPORT1
0x4001_b000	4KB		I2C2
0x4001_c000	4KB		I2C3
0x4001_d000	4KB		SCI3
0x4001_e000	4KB		保留
0x4002_0000	4KB		ADC
0x4002_1000	4KB		DAC
0x4002_2000	4KB		保留
0x4002_3000	4KB		保留
0x4002_4000	4KB		EPORT2
0x4002_5000	4KB		EPORT3
0x4002_6000	4KB		EPORT4
0x4003_0000	4KB		保留
0x4003_1000	4KB		保留
0x4003_2000	4KB		保留
0x4003_3000	4KB		保留
0x4003_4000	4KB		保留
0x4003_5000	4KB		PMU_RTC
0x4003_6000	4KB	AHB_IPS2	保留
0x4003_7000	4KB		保留
0x4003_8000	4KB		保留
0x4003_9000	4KB		EDMAC0
0x4003_a000	4KB		保留
0x4004_0000	4KB	AHB1(AHB_CLB)	保留
0x4004_1000	4KB		保留
0x4004_2000	4KB		保留
0x4004_3000	4KB		保留
0x4004_4000	4KB	AHB2	保留
0x4004_5000	4KB		保留
0x4004_6000	4KB		DMAC1
0x4004_7000	4KB		DMAC2
0x4004_8000	4KB		保留
0x4004_9000	4KB		保留
0x4004_a000	4KB		保留
0x4004_b000	4KB	AHB3	保留
0x4004_c000	4KB		USBC

0x4005_0000	4KB	AHB-APB	保留
0x4005_1000	4KB		Cache_Config
0x4005_2000	4KB		保留
0x4005_3000	4KB		保留
0x4005_4000	4KB		保留
0x4005_5000	4KB		Cache2_Config
0x6000_0000	16MB	AHB_SPIM1	保留
保留	4KB	保留	保留
保留	4KB	保留	保留
0xe000_0000	4KB	M4	保留

4 内核配置模块（CCM）

4.1 概述

内核配置模块用于配置芯片初始信息和工作模式。

4.2 特性

内核配置模块执行以下操作：

- 反映启动引导设备
- Master 模式
- Single Chip 模式
- 选择引导设备
- 选择总线监控器配置：用于异常情况下产生异常信号给 CPU
- 配置频率检测的配置值
- 配置 USBPHY 的一些参数
- 配置某些 PAD 的上拉控制

4.3 工作模式

内核配置模块有以下工作模式：

- Master 模式
- Single Chip 模式

工作模式是在复位时确定，此后不能被改变。

4.3.1 Master 模式

即 Master 引导模式，启动存储在内部 FLASH 的引导程序。

4.3.2 Single Chip 模式

即 Single Chip 引导模式，启动存储在内部 ROM 的引导程序。

4.4 外部管脚

表格 4-1 显示了内核配置模块信号的概述。

表格 4-1：内核配置模块信号描述

名 称	功 能	复 位 状 态
Mode[1:0]	模式选择	低电平（内部弱下拉）
Test	测试模式选择	低电平（内部弱下拉）

4.5 内存映射和寄存器

本节介绍了内核配置模块的内存地址映像和寄存器。

4.5.1 编程模型

内核配置模块的编程模型由以下寄存器组成：

CCR（芯片配置寄存器）：控制芯片的配置。详细内容参考芯片配置寄存器(CCR)。
CIR（芯片识别寄存器）：包含了一个唯一序列号。详细内容参考芯片配置寄存器(CCR)。

一些控制寄存器位被实现为一次写入位。这些位始终是可读的，但是一旦写入了该位，除了调试和测试操作期间，其他写操作均无效。

在调试模式或测试模式下，可以读写某些一次写入位和测试位。退出调试或测试模式后，芯片配置模块将根据当前寄存器值恢复操作。如果在调试或测试模式下首次写入一次写入寄存器位，则退出调试或测试模式后该寄存器位仍可写。表格 4-2 显示了一次写入位的可访问性。

表格 4-2：一次性写入位读/写可访问性

配 置	读 写 权 限
所有配置	Read-always
Debug 操作(在所有模式下)	Write-always
测试操作（在所有模式下）	Write-always
主机模式	Write-once

4.5.2 内存映射

内核配置模块的基地址是：0x63F04000

表格 4-3：内核配置模块的内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	芯片配置寄存器(CCR)	保留	S
0x0004	PHY 参数配置寄存器（PHYPA）	芯片识别寄存器（CIR）	S
0x0008	芯片测试寄存器（CTR）	保留	S
0x000C	保留		---
0x0010	PMU1/2 配置寄存器（PCFG12）		S
0x0014	PMU3 配置寄存器（PCFG3）		S
0x0018	PMU_RTC1/2 配置寄存器（RTCCFG12）		S
0x001C	PMU_RTC3 配置寄存器（RTCCFG3）		S
0x0020	PMU_RTC 状态寄存器（RTCSR）		S
0x0028	PMU_RTC 脉冲寄存器（RTC_PR）		S

注意：

S = 仅限管理员访问。用户模式仅访问主管的地址位置无效，并导致周期终止传输错误。

保留地址写入无效，读数返回零。

4.5.3 寄存器描述

寄存器读写属性缩写格式说明如下：

表格 4-4：寄存器读写属性缩写格式

读写属性与缩写	说 明
read/write（rw）	此位可读可写。
read-only（ro）	此位只可读，不可写。
write-only（wo）	此位只可写，不可读。
read-clear（rc）	此位只可读，读的同时自动清除该位。
read/write_1_clear（r/w1c）	此位可读可写，写 1 时清除此位，写 0 无影响。
read/write_0_clear（r/w0c）	此位可读可写，写 0 时清除此位，写 1 无影响。
read/write_1_only（r/w1o）	此位可读可写，只可写 1，写 0 无影响。
read/write_0_only（r/w0o）	此位可读可写，只可写 0，写 1 无影响。

4.5.3.1 芯片配置寄存器

偏移地址：0x0002~0x0003 复位值：0x0808

15	14	13	12	11	10	9	8
SWAPDIS	TESTDIS	BTLDDIS	JTAGDIS	CLKMODE_PDE	MODE2	MODE1	MODE0
ro	ro	ro	ro	rw	ro	ro	ro

7	6	5	4	3	2	1	0
Reserved	PERIPH_BRIDGE_PAE	PERIPH_BRIDGE_RAE	SHINT	BME	BMD	BMT	
ro	rw	rw	rw	rw	rw	rw	

图表 4-1：芯片配置寄存器（CCR）

比特位	名称	复位值	读写属性	功能说明
[15]	SWAPDIS	0x0	RO	该位显示交换模式是否使能。 0 = 使能交换模式；1 = 禁止交换模式
[14]	TESTDIS	0x0	RO	该位显示测试模式是否使能。 0 = 使能测试模式；1 = 禁止测试模式
[13]	BTLDDIS	0x0	RO	该位显示 boot 模式是否使能。 0 = 使能 Bootloader 模式 1 = 禁止 Bootloader 模式
[12]	JTAGDIS	0x0	RO	该位显示 Jtag 模式是否使能。 0 = 使能 Jtag 模式；1 = 禁止 Jtag 模式

比特位	名称	复位值	读写属性	功能说明
[11]	CLKMODE_PDE	0x1	RW	clkmode 管脚下拉使能。 如果设置了 clkmode，则选择内部振荡器。如果 clkmode 清除，则选择外部振荡器。 0 = 管脚下拉不使能；1 = 管脚下拉使能
[10:8]	MODE	0x0	RO	MODE[2:0]：芯片配置模式。 000 = Rom 引导模式 001 = Flash 引导模式 010 = ROM 引导切换模式 011 = Flash 引导切换模式 1xx = 芯片测试模式
[7]	保留	0x0	RO	---
[6]	PERIPH_BRIDGE_PAE	0x0	RW	早期时隙缺陷错误使能(早期总线地址错误使能)。 0 = 未使能；1 = 使能
[5]	PERIPH_BRIDGE_RAE	0x0	RW	下电时隙错误以及从机传输错误使能(下电总线地址错误使能）。 0 = 未使能；1 = 使能
[4]	SHINT	0x0	RW	SHINT 位使得有效中断请求具有可视性。 如果 SHINT 位设定,那么 RSTOUT 管脚就是快速和正常中断信号的逻辑“或”的结果。 0 = 正常 RSTOUT 管脚功能 1 = 内部请求反映在 RSTOUT 管脚上 **注意：**与 SHINT 功能相比，复位控制器上的 FRCRSTOUT 功能具有优先权。
[3]	BME	0x1	RW	BME 位控制着总线监控器的使能。 BME 位使总线监控器能够运行。总线监控器可以监控内部总线的运行。一旦内部总线超时（根据 BMT）有响应，总线监控器就会终止总线运行。 0 = 总线监控器关闭；1 = 总线监控器使能 **注意：**这里的总线指 mlb 总线，功能用于检测 mlb 总线超时。
[2]	BMD	0x0	RW	BMD 位控制着总线监控器在调试模式时的行为。 0 = 总线监控器在调试模式时关闭 1 = 总线监控器在调试模式时使能
[1:0]	BMT	0x0	RW	BMT 位设置总线监控器的时间参数。 00 = 256；01 = 128；10 = 64 ；11 = 16

4.5.3.2 PHY 参数配置寄存器

偏移地址：0x0006~0x0007 复位值：0x8000

15	14	13	12	11	10	9	8
USB_REG_ENDIAN	PHY_RESUME_SEL	PHY_ID_PULLUP	PHY_ID_DIR	PHY_VALID_DIR	PHY_VBUSVALID	PHY_AVALID	PHY_SESSENDVALID
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USBPHY_REG_EN_SEL	USBPHY_12M_EN_SEL	USBPHY_DIR	USBPHY_TERMSEL	USBPHY_TERMSEL_DIR	USBPHY_SUSPENDM	USBPHY_REG_EN	USBPHY_PLL_EN
rw	rw	rw	rw	rw	rw	rw	rw

图表 4-2：PHY 参数配置寄存器（PHYPA）

比特位	名称	复位值	读写属性	功能说明
[15]	USB_REG_ENDIAN	0x1	RW	USB 寄存器大小端选择位 0 = USB 寄存器小端 1 = USB 寄存器大端
[14]	PHY_RESUME_SEL	0x0	RW	USB 恢复信号选择 0 = 通过 LineState 生成 USB 恢复 1 = 通过LineState和Power State生成USB Resume
[13]	PHY_ID_PULLUP	0x0	RW	USBPHY ID 上拉配置，需要与 PHYPA[12]一起配置 0 = 禁止 USBPHY ID 上拉 1 = 启用 USBPHY ID 上拉
[12]	PHY_ID_DIR	0x0	RW	USBPHY ID 上拉由 PHYPA[13]控制选择位 0 = 选择 USBPHY ID 上拉由 USBC 控制器控制的 1 = 选择 USBPHY ID 上拉由 PHYPA 第 13 位控制
[11]	PHY_VALID_DIR	0x0	RW	选择 USBPHY VBUSVALID（PHYPA[10]），AVALID（PHYPA[9]），SESSENDVALID（PHYPA[8]）由寄存器控制 0 = 选择 USBPHY VBUSVALID AVALID SESSENDVALID 控制者 USBC 控制器 1 = 选择 USBPHY VBUSVALID AVALID SESSENDVALID 控制者寄存器

比特位	名称	复位值	读写属性	功能说明
[10]	PHY_VBUSVALID	0x0	RW	USBPHY VBUSVALID 配置，需要使用 PHYPA[11]配置 0 = USBPHY VBUSVALID 禁用 1 = 使能 USBPHY VBUSVALID
[9]	PHY_AVALID	0x0	RW	USBPHY AVALID 配置,需要配置 PHYPA[11] 0 = 禁用 USBPHY 1 = 使能 USBPHY 有效
[8]	PHY_SESSENDVALID	0x0	RW	USBPHY SESSENDVALID 配置，需要配置 PHYPA[11] 0 = 禁用 USBPHY SESSENDVALID 1 = 使能 USBPHY SESSENDVALID
[7]	USBPHY_REG_EN_SEL	0x0	RW	选择由 PHYPA 控制的 USBPHY REG_EN[1] 0 = 选择由 USBC 控制器控制的 USBPHY REG_EN 1 = 选择由 PHYPA[1]控制的 USBPHY REG_EN
[6]	USBPHY_12M_EN_SEL	0x0	RW	控制 CLK_12M_EN 输出 0 = 禁用 USBPHY CLK_12M_EN 输出 1 = 使能 USBPHY CLK_12M_EN 输出
[5]	USBPHY_DIR	0x0	RW	选择 USBPHY PLL_EN（PHYPA[0]），SUSPENDM（PHYPA[2]），由寄存器控制 0 = 选择 USBPHY USBPHY PLL_EN 和 SUSPENDM 由控制 USBC 控制器 1 = 选择 USBPHY USBPHY PLL_EN 和 SUSPENDM 由控制寄存器
[4]	USBPHY_TERMSEL	0x0	RW	USBPHY TERMSELECT 配置，需要配置 PHYPA[3] 0 = 禁用 USBPHY TERMSELECT 1 = 启用 USBPHY TERMSELECT
[3]	USBPHY_TERMSEL_DIR	0x0	RW	选择 USBPHY TERMSELECT（PHYPA[4]） 0 = 选择 USBPHY USBPHY TERMSELECT 由 USBC 控制器控制 1 = 选择 USBPHY USBPHY TERMSELECT 由寄存器控制
[2]	USBPHY_SUSPENDM	0x0	RW	USBPHY 挂起配置，需要配置 PHYPA[5] 0 = 禁用 USBPHY 挂起 1 = 使能 USBPHY 挂起

比特位	名称	复位值	读写属性	功能说明
[1]	USBPHY_REG_EN	0x0	RW	USBPHY REG_EN 配置，需要配置 PHYPA[7] 0 = 禁用 USBPHY REG_EN 1 = 使能 USBPHY REG_EN
[0]	USBPHY_PLL_EN	0x0	RW	USBPHY PLL_EN 配置，需要使用 PHYPA[5] 配置 0 = 禁用 USBPHY PLL_EN 1 = 使能 USBPHY PLL_EN

4.5.3.3 芯片识别寄存器

CIR 寄存器是只读寄存器。写入 CIR 无效。芯片 ID 和修订号可以在该寄存器中读取。

偏移地址：0x0004~0x0005　　复位值：0x3403

15	14	13	12	11	10	9	8
PIN							
ro							

7	6	5	4	3	2	1	0
PRN							
ro							

图表 4-3：芯片识别寄存器（CIR）

比特位	名称	复位值	读写属性	功能说明
[15:8]	PIN	0x34	RO	芯片只读认证码区域。 该只读字段包含芯片的唯一版本标识号。
[7:0]	PRN	0x03	RO	芯片只读版本码区域。 此只读字段包含 full-mask 修订版本号。这个号码对于此部分的每个新的 full-mask 集，其值增加一个。修订数字按时间顺序分配。

4.5.3.4 芯片测试寄存器

芯片测试寄存器保留给出厂测试。

偏移地址：0x000A~0x000B　　　　复位值：0x0000

15	14	13	12	11	10	9	8
保留							TEST
ro	ro	ro	ro	ro	ro	ro	ro

7	6	5	4	3	2	1	0
保留							
ro	ro	ro	ro	ro	ro	ro	ro

图表 4-4：芯片测试寄存器（CTR）

比特位	名称	复位值	读写属性	功能说明
[15:9]	保留	0x0	RO	---
[8]	TEST	0x0	RO	ips_test_access 选择位
[7:0]	保留	0x0	RO	---

注意：为防止意外激活测试逻辑，请写入 0x0000 以锁定测试功能。将 CTR 寄存器中的任何位置 1 可能会导致不可预测的结果。

4.5.3.5 PMU1/2 配置寄存器

偏移地址：0x0010~0x0013　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							
ro							

图表 4-5：PMU1/2 配置寄存器（PCFG12）

比特位	名称	复位值	读写属性	功能说明
[31:0]	保留	0x0	RO	---

4.5.3.6 PMU3 配置寄存器

偏移地址：0x0014~0x0017 复位值：0xFF013C00

31	30	29	28	27	26	25	24
GINT0_IE	GINT1_IE	GINT2_IE	GINT3_IE	GINT4_IE	GINT5_IE	GINT6_IE	GINT7_IE
ro	ro	ro	ro	ro	ro	ro	ro

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留	GINT37_PS	SS2_SWAP	保留	保留	TDO_PUE	RSTOUT_PUE	CLKOUT_PUE
ro	rw	rw	ro	ro	rw	rw	rw

7	6	5	4	3	2	1	0
保留					PWM2_EN	PWM1_EN	PWM0_EN
ro					rw	rw	rw

图表 4-6：PMU2 配置寄存器（PCFG3）

比特位	名称	复位值	读写属性	功能说明
[31:24]	GINTx_IE	0xFF	RO	GINTx_IE—管脚 GINTx IE 选择位 0 = 管脚 GINTx 输入禁用 1 = 管脚 GINTx 输入使能
[23:16]	保留	0x01	RO	---
[15]	保留	0x0	RO	---
[14]	GINT37_PS	0x0	RW	GINT[37]管脚上下拉选择位 0 = GINT[37]下拉使能 1 = GINT[37]上拉使能
[13]	SS2_SWAP	0x1	RW	SS2/GINT 切换使能 0 = 禁止切换（GINT 功能） 1 = 使能切换（SPI 功能）
[12:11]	保留	0x11	RO	--
[10]	TDO_PUE	0x1	RW	tdo 管脚上下拉使能位 0 = tdo 上下来禁止 1 = tdo 上下来使能
[9]	RSTOUT_PUE	0x0	RW	rstout 管脚上下拉使能位。 0 = rstout 上下拉禁止

比特位	名称	复位值	读写属性	功能说明
				1 = rstout 上下拉使能
[8]	CLKOUT_PUE	0x0	RW	clkout 管脚上下拉使能位。 0 = clkout 上下拉禁止 1 = clkout 上下拉使能
[7:3]	保留	0x0	RO	---
[2]	PWM2_EN	0x0	RW	PWM2 输入、输出使能位
[1]	PWM1_EN	0x0	RW	PWM1 输入、输出使能位
[0]	PWM0_EN	0x0	RW	PWM0 输入、输出使能位

4.5.3.7 RTC1/2 配置寄存器

偏移地址：0x0018~0x001B　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
second pulse inv	minute pulse inv	hour pulse inv	day pulse inv	second_int enable	minute_int enable	hour_int enable	day_int enable
rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
rtc_en_interface[1]	保留						
rw	ro						

7	6	5	4	3	2	1	0
保留				bias_trim_value[2]			
ro				rw			

图表 4-7：RTC1/2 配置寄存器（RTCCFG12）

注意：

1. 该寄存器从 eflash 加载，并且读写该寄存器必须通过配置 CTR 进入 ips 测试模式寄存器
2. 该寄存器从 eflash 加载，并且读写该寄存器必须通过配置 CTR 进入 ips 测试模式寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23]	second pulse inv	0x0	RW	PMU_RTC 秒脉冲反转
[22]	minute pulse inv	0x0	RW	PMU_RTC 分钟脉冲反转

比特位	名称	复位值	读写属性	功能说明
[21]	hour pulse inv	0x0	RW	PMU_RTC 小时脉冲反转
[20]	day pulse inv	0x0	RW	PMU_RTC 天脉冲反转
[19]	second_int enable	0x0	RW	PMU_RTC 秒中断使能
[18]	minute_int enable	0x0	RW	PMU_RTC 分钟中断使能
[17]	hour_int enable	0x0	RW	PMU_RTC 小时中断使能
[16]	day_int enable	0x0	RW	PMU_RTC 天中断使能
[15]	rtc_en_interface	0x0	RW	RTC 接口使能位 0 = RTC 接口禁止 1 = RTC 接口使能
[14:4]	保留	0x0	RO	--
[3:0]	bias_trim_value	0x0	RW	配置 osc 偏置调整值

4.5.3.8 RTC3 配置寄存器

偏移地址：0x001C~0x001F　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留				cap_trim_value[1]			
ro				rw			

23	22	21	20	19	18	17	16
保留						RTC_TM	保留
ro[2]						rw[3]	ro[3]

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							
ro							

图表 4-8：RTC3 配置寄存器（RTCCFG3）

注意：

1. 该寄存器从 eflash 加载，并且读写该寄存器必须通过配置 CTR 进入 ips 测试模式寄存器
2. 读写该寄存器必须通过配置 CTR 进入 ips 测试模式寄存器

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:24]	cap_trim_value	0x0	RW	配置 OSC C1/C2 调整值 **注意：**在配置 bias_trim_value 和 cap_trim_value 之前，请先设置位 rtc_en_interface = 1，然后等待 2us，设置 bias_trim_value 和 cap_trim_value。
[23:18]	保留	0x0	RO	---
[17]	RTC_TM	0x0	RW	RTC 测试模式控制位 0 = 禁用 rtc 测试模式 1 = 启用 rtc 测试模式 **注意：**在设置此位之前，应先设置 ips_test_access
[16:0]	保留	0x0	RO	---

4.5.3.9 PMU_RTC 状态寄存器

偏移地址：0x0020~0x0023　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留				lvdt5v_src	lvdt1p8v_src	osc_128k_clk_src	osc_2k_clk_src
ro				rw	rw	rw	rw

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							
ro							

图表 4-9：PMU_RTC 状态寄存器（RTCSR）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27]	lvdt5v_src	0x0	RW	在 vccq 测试模式下的 vdd5v 标志 0 = 不工作；1 = vdd5v 工作

比特位	名称	复位值	读写属性	功能说明
[26]	Lvdt1p8v_src	0x0	RW	在 vccq 测试模式下的 vdd1.8v 标志 0 = 不工作；1 = vdd1.8v 工作
[25]	osc_128k_clk_src	0x0	RW	在测试模式下，vdd3p3v 128k clk 源状态 0 = 时钟源 0；1 = 时钟源 1
[24]	osc_2k_clk_src	0x0	RW	在测试模式下，vdd3p3v 2k clk 源状态 0 = 时钟源 0；1 = 时钟源 1
[23:0]	保留	0x0	RO	---

4.5.3.10 PMU_RTC 脉冲寄存器

偏移地址：0x0028~0x002B 复位值：0x00000000

31	30	29	28	27	26	25	24
day pulse	hour pulse	minute pulse	second pulse	保留			
ro				ro			

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							
ro							

图表 4-10：PMU_RTC 脉冲寄存器（RTCPR）

比特位	名称	复位值	读写属性	功能说明
[31]	day pulse	0x0	RO	每天响应并持续一秒钟
[30]	hour pulse	0x0	RO	每小时响应并持续一秒钟
[29]	minute pulse	0x0	RO	每分钟响应并持续一秒钟
[28]	second pulse	0x0	RO	每秒响应并持续 16 微秒
[27:0]	保留	0x0	RO	---

5 高速缓存模块（Cache）

5.1 概述

高速缓存模块（Cache）为处理器提供了紧密耦合的本地处理器存储器和到所有从存储器空间的总线路径。

高速缓存是包含地址信息的高速存储器位置的块（通常称为标签）和相关数据。目的是减少内存访问的平均时间。Cache 根据两个本地性原则操作：

空间位置-访问一个位置之后很可能会进行访问从相邻位置（例如，顺序执行指令或使用数据结构）。
时间局部性-可能会在很短的时间内重复访问内存区域（例如，执行代码循环）。

为了最大程度地减少控制信息的存储量，空间局部性用于将同一标签下的多个位置分组在一起。这个逻辑块通常称为 Cache 行。将数据加载到 Cache 中时，后续加载和存储的访问时间减少，从而带来整体性能优势。信息获取时已经在 Cache 中的被称为 Cache 命中，其他访问称为 Cache 不命中。

通常，Cache 是自我管理的，更新会自动发生。每当处理器要访问可缓存位置时，Cache 就开始检查。如果访问是 Cache 命中，则立即进行访问。否则，Cache 会分配一个行位置，并从内存中加载到 Cache 行。可能会有不同的 Cache 拓扑和访问策略。

然而，它们必须符合基础存储的一致性模型结构。Cache 会带来许多潜在的问题，主要是由于：
内存访问发生在程序员不期望的时间
存在可以保存数据项的多个物理位置

Cache 控制器旨在与任何 32 位 AMBA_AHB 总线应用一起使用，该应用是需要缓存功能的。此缓存功能可以通过提供对最近使用的代码或数据的快速访问来提高性能。本地内存控制器支持三种操作模式：

1. 直写式-使用此缓存模式访问地址空间是可缓存的。

输入总线上的直写读取未命中导致输出总线上包含所需地址的 16 字节对齐的内存地址的行读取。这个不命中数据加载到 Cache 中，并标记为有效且未修改。
对有效缓存位置的直写读取命中从 Cache 返回数据且没有输出总线访问。
直写写入未命中会绕过 Cache 并写入输出总线（没有为直写模式空间下写丢失策略的访问）。
•直写写入命中会更新 Cache 命中数据并写入输出总线。

2. 写回-使用此缓存模式访问地址空间是可缓存的。

输入总线上的写回读取未命中将导致输出总线上包含所需地址的 16 字节对齐的内存地址的行读取。这个未命中数据被加载到 Cache 中并标记为有效且未修改。
对有效 Cache 位置的回写读命中将从 Cache 返回数据且没有输出总线访问。
写回写未命中将执行“读-写”（为回写模式空间下写丢失策略的访问）。在输出总线上读取包含所需写入地址的 16 字节对齐的行地址。这个未命中的数据加载到 Cache 中并标记为有效并已修改；然后写入数据将并

更新适当的 Cache 数据位置。

3. 不可缓存-使用这种缓存模式无法访问地址空间可缓存的。这些访问绕过 Cache 并访问输出总线。

Cache 模块具有两个 AMBA_AHB 总线接口，一个主设备和一个从设备接口。主接口具有解码逻辑，可决定对有效的地址段缓存模式的访问。然后，主访问将访问或绕过缓存，这取决于其缓存模式。从接口用于缓存未命中以及绕过缓存的访问。

Cache 模块具有 2 路组相联结构。缓存具有 32 位宽的地址和数据路径以及 16 字节的行大小。缓存标签和数据存储使用单端口同步 RAM。Cache 模块控制器有一个各种读写数据缓冲区以提高性能。Cache 未命中和行推送会使用重要字优先访问生成 4 节拍 32 位回滚突发访问以获得最佳性能。

Cache 模块使用其单时钟输入的上升沿。单输入时钟必须与主总线时钟匹配。主总线和从总线与平台总线的频率相同。

Cache 模块具有 AMBA_AHB 总线接口以访问其程序员的模型。程序员模型提供对缓存控制功能和资源以及所有缓存标记和数据存储。

5.2 特性

8KB 大小指令数据共享 Cache

16 字节 Cache 行

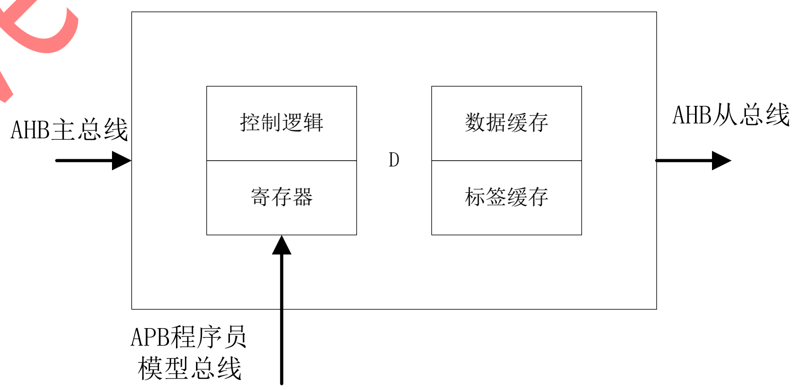
支持直写和写回模式

支持 Cache 组命令和行命令

支持页清除命令

5.3 框图

这些缓存模块提供对 RAM 的零等待状态访问并且可缓存地址空间。



图表 5-1：Cache 模块框图

5.4 内存映射和寄存器

5.4.1 内存映射

表格 5-1：内存映射/寄存器

偏 移 地 址	位 31-16	位 15-0	访 问 权 限
0x0000	Cache 控制寄存器（CCR）		S/U
0x0004	Cache 行控制寄存器（CLCR）		S/U
0x0008	Cache 查询地址寄存器（CSAR）		S/U
0x000C	Cache 读/写数值寄存器（CCVR）		S/U
0x0020	Cache 主地址段存取权限寄存器（CACR）		S/U
0x0024	Cache 子地址段存取权限寄存器（CSACR）		S/U
0x0040	Cache 段 6 子区域 0 地址上限寄存器（CR6S0HA）		S/U
0x0044	Cache 段 6 子区域 1 地址上限寄存器（CR6S1HA）		S/U
0x0048	Cache 段 6 子区域 2 地址上限寄存器（CR6S2HA）		S/U
0x004C	Cache 段 6 子区域 3 地址上限寄存器（CR6S3HA）		S/U
0x0050	Cache 段 6 子区域 0 地址下限寄存器（CR6S0LA）		S/U
0x0054	Cache 段 6 子区域 1 地址下限寄存器（CR6S1LA）		S/U
0x0058	Cache 段 6 子区域 2 地址下限寄存器（CR6S2LA）		S/U
0x005C	Cache 段 6 子区域 3 地址下限寄存器（CR6S3LA）		S/U
0x0060	Cache 段 7 子区域 0 地址上限寄存器（CR7S0HA）		S/U
0x0064	Cache 段 7 子区域 1 地址上限寄存器（CR7S1HA）		S/U
0x0068	Cache 段 7 子区域 2 地址上限寄存器（CR7S2HA）		S/U
0x006C	Cache 段 7 子区域 3 地址上限寄存器（CR7S3HA）		S/U
0x0070	Cache 段 7 子区域 0 地址下限寄存器（CR7S0LA）		S/U
0x0074	Cache 段 7 子区域 1 地址下限寄存器（CR7S1LA）		S/U
0x0078	Cache 段 7 子区域 2 地址下限寄存器（CR7S2LA）		S/U
0x007C	Cache 段 7 子区域 3 地址下限寄存器（CR7S3LA）		S/U
0x0080	Cache 段 2 子区域 0 地址上限寄存器（CR2S0HA）		S/U
0x0084	Cache 段 2 子区域 1 地址上限寄存器（CR2S1HA）		S/U
0x0088	Cache 段 2 子区域 2 地址上限寄存器（CR2S2HA）		S/U
0x008C	Cache 段 2 子区域 3 地址上限寄存器（CR2S3HA）		S/U
0x0090	Cache 段 2 子区域 0 地址下限寄存器（CR2S0LA）		S/U
0x0094	Cache 段 2 子区域 1 地址下限寄存器（CR2S1LA）		S/U
0x0098	Cache 段 2 子区域 2 地址下限寄存器（CR2S2LA）		S/U
0x009C	Cache 段 2 子区域 3 地址下限寄存器（CR2S3LA）		S/U
0x0180	Cache 页清除地址寄存器（CPEA）		S/U
0x0184	Cache 页清除大小寄存器（CPES）		S/U
0x0188	Cache 时钟门控寄存器（CCG）		S/U

注意:

1. S = 仅 CPU 管理员模式访问，U = 仅 CPU 用户模式访问
2. 只能访问主管地址的用户模式无效，并且导致循环终止传输错误

5.4.2 寄存器描述

本小节提供寄存器描述。

5.4.2.1 Cache 控制寄存器(CCR)

偏移地址：0x0000~0x0003 复位值：0x00000030

31	30	29	28	27	26	25	24
GO	保留			PUSH W1	INVW 1	PUSH W0	INVW0
rw	ro			rw	rw	rw	rw

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				保留			ENCache
ro				ro			rw

图表 5-2：Cache 控制寄存器(CCR)

比特位	名称	复位值	读写属性	功能说明
[31]	GO	0x0	RW	启动 Cache 命令 将该位置 1 将启动 Cache 命令，命令由位 27-24 指示。读取该位指示命令是否处于活动状态。这一点保持设置直到命令完成。写零无效。 0 = 写：无效。读取：无缓存命令处于活动状态 1 = 写入：启动由位 27-24 指示的命令。读取：缓存命令活动性
[30:28]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[27]	PUSHW1	0x0	RW	推送 WAY 1 0 = 无操作 1 = 将 GO 位置 1 时，将所有 WAY 1 已修改的行推送
[26]	INVW1	0x0	RW	使 WAY 1 无效 如果 PUSHW1 和 INVW1 位置 1，则在设置 GO 位后，将所有修改的行在 WAY 1 中推送，并使 WAY 1 中所有行无效（清除 WAY 1）。 0 = 无操作 1 = 将 GO 位置 1 时，使 WAY 1 的所有行无效
[25]	PUSHW0	0x0	RW	推送 WAY 0 0 = 无操作 1 = 将 GO 位置 1 时，将所有 WAY 0 已修改的行推送
[24]	INVW0	0x0	RW	使 WAY 0 无效 如果 PUSHW0 和 INVW0 位置 1，则在设置 GO 位后，将所有修改的行在 WAY 0 中推送，并使 WAY 0 中所有行无效（清除 WAY 0）。 0 = 无操作 1 = 将 GO 位置 1 时，使 WAY 0 的所有行无效
[23:4]	保留	0x00003	RO	---
[3:1]	保留	0x0	RO	---
[0]	ENCache	0	RW	启用缓存 0 = 禁用缓存；1 = 启用缓存

5.4.2.2 Cache 行命令控制寄存器（CLCR）

偏移地址：0x0004~0x0007 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				LACC	LADSEL	LCMD[1:0]	
ro				rw	rw	rw	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留	LCR[2:0]			保留			TDSEL
ro	rw			ro			rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留	WSEL	保留		Cache_ADDRESS[9:0]			
ro	rw	ro		rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Cache_ADDRESS[9:0]						保留	LGO
rw						ro	rw

图表 5-3：Cache 行命令控制寄存器（CLCR）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27]	LACC	0x0	RW	行访问类型 0 = 读；1 = 写
[26]	LADSEL	0x0	RW	行地址选择 使用缓存地址时，WAY 0 或者 WAY 1 必须也可以在 CLCR [WSEL]中指定。使用物理地址时，两个 WAY 都将被搜索并仅在命中时执行命令。 0 = 缓存地址；1 = 物理地址
[25:24]	LCMD[1:0]	0x0	RW	行命令 00 = 搜索和读取或写入 01 = 无效 10 = 推送 11 = 清除
[23]	保留	0x0	RW	---
[22:20]	LCR[3:0]	0x0	RW	命令对应行初始状态信息 参考章节 “5.5.7 行命令结果”。
[19:17]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[16]	TDSEL	0x0	RW	标签/数据选择 选择标签或数据进行搜索和读取或写入命令。 0 = 数据；1 = 标签
[15]	保留	0x0	RO	---
[14]	WSEL	0x0	RW	方式选择 0 = WAY 0；1 = WAY 1
[13:12]	保留	0x0	RO	---
[11:2]	Cache_ADDRESS[9:0]	0x0	RW	缓存地址 CLCR[11:4]位用于访问标签阵列; CLCR [11:2]位用于访问数据阵列。
[1]	保留	0x0	RO	---
[0]	LGO	0x0	RW	启动缓存行命令 将该位置 1 将启动 Cache 行命令，命令由位 27-24 指示。读取该位表示是否有行命令活动性。该位将保持设置状态，直到命令完成。写零没有效果。该位与 CSAR [LGO]共享。 0 = 写：无效。读取：无行命令有效 1 = 写：启动由位 27-24 指示的行命令。读取：行命令活动性

5.4.2.3 Cache 查询地址寄存器（CSAR）

CSAR 寄存器用于定义基于行命令的显式缓存地址或物理缓存，该方式取决于 CLCR [LADSEL]位。

偏移地址：0x0008~0x000B　　复位值：0x00000000

31	30	29	28	27	26	25	24
PHYSICAL_ADDRESS[29:0]							
rw							

23	22	21	20	19	18	17	16
PHYSICAL_ADDRESS[29:0]							
rw							

15	14	13	12	11	10	9	8
PHYSICAL_ADDRESS[29:0]							
rw							

7	6	5	4	3	2	1	0
PHYSICAL_ADDRESS[29:0]						保留	LGO
rw						ro	rw

图表 5-4：Cache 查询地址寄存器（CSAR）

比特位	名称	复位值	读写属性	功能说明
[31:2]	PHYSICAL_ADDRESS[29:0]	0x0	RW	物理地址 表示系统地址的位[31:2]。 CSAR [31:12]位用于标签比较 CSAR [11:4]位用于访问标签阵列 CSAR [11:2]位用于访问数据阵列。
[1]	保留	0x0	RO	---
[0]	LGO	0x0	RW	启动缓存行命令 将该位置 1 将启动 Cache 由位 27-24 指示的行命令。读取该位表示是否有行命令活动。该位将保持设置状态，直到命令完成。写入零无效。 该位与 CLCR [LGO]共享。 0 = 写：无效。读取：无行命令有效 1 = 写：启动由位 27-24 指示的行命令。读取：行命令活动性

5.4.2.4 Cache 读/写数值寄存器（CCVR）

CCVR 寄存器用于为写数据提供源数据或返回读数据。

偏移地址：0x000C~0x000F　　复位值：0x00000000

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

图表 5-5：Cache 读/写数值寄存器（CCVR）

比特位	名称	复位值	读写属性	功能说明
[31:0]	DATA[31:0]	0x0	RW	缓存读/写数据 对于标签搜索，请读取或写入： CCVR [31:12]位用于标签阵列的读/写值 CCVR [11:4]位用于读取时的标签集地址；在写时未使用 CCVR [3:2]位保留 对于数据搜索，读取或写入： CCVR [31:0]位用于数据阵列读/写值

5.4.2.5 Cache 主地址段存取寄存器（CACR）

偏移地址：0x0020~0x0023 复位值：0x000FFFFF

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留				保留		R8_CacheABLE	R8WT_WB
ro				ro		rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留							
ro							

图表 5-6：Cache 主地址段存取寄存器

该寄存器用于控制几个不同存储区域的属性，例如直写，回写或不可缓存。

区域 2：0x08000000∽0x0807FFFF

区域 6：0x10000000∽0x10FFFFFF

区域 7：0x00000000∽0x00FFFFFF

区域 8：0x04000000∽0x04005FFF

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:18]	保留	0xFF	RO	---
[17]	R8_CacheABLE	0x1	RW	区域 8 是可缓存的 0 = 不可缓存；1 = 可缓存
[16]	R8_WT_WB	0x1	RW	如果可以缓存，则区域 8 是直写的 0 = 直写；1 = 回写
[15:0]	保留	0xF	RO	---

5.4.2.6 Cache 子地址段存取寄存器（CSACR）

该寄存器用于控制几个不同主存储区域再划分后的属性，例如直写，回写或不可缓存。

偏移地址：0x0024~0x0027　　复位值：0xFFFFFFAA

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R7_3_CacheABLE	R7_3_WT_WB	R7_2_CacheABLE	R7_2_WT_WB	R7_1_CacheABLE	R7_1_WT_WB	R7_0_CacheABLE	R7_0_WT_WB
rw	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R6_3_CacheABLE	R6_3_WT_WB	R6_2_CacheABLE	R6_2_WT_WB	R6_1_CacheABLE	R6_1_WT_WB	R6_0_CacheABLE	R6_0_WT_WB
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R2_3_CacheABLE	R2_3_WT_WB	R2_2_CacheABLE	R2_2_WT_WB	R2_1_CacheABLE	R2_1_WT_WB	R2_0_CacheABLE	R2_0_WT_WB
rw	rw	rw	rw	rw	rw	rw	rw

图表 5-7：Cache 子地址段存取寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	AUTO_HIBER	0xFF	RW	---
[23]	R7_3_CacheABLE	0x1	RW	区域 7 的子段 3 是可缓存的 0 = 不可缓存；1 = 可缓存
[22]	R7_3_WT_WB	0x1	RW	区域 7 的子段 3 是回写的 0 = 直写；1 = 回写
[21]	R7_2_CacheABLE	0x1	RW	区域 7 的子段 2 是可缓存的 0 = 不可缓存；1 = 可缓存
[20]	R7_2_WT_WB	0x1	RW	区域 7 的子段 2 是回写的 0 = 直写；1 = 回写
[19]	R7_1_CacheABLE	0x1	RW	区域 7 的子段 1 是可缓存的 0 = 不可缓存；1 = 可缓存
[18]	R7_1_WT_WB	0x1	RW	区域 7 的子段 1 是回写的 0 = 直写；1 = 回写
[17]	R7_0_CacheABLE	0x1	RW	区域 7 的子段 0 是可缓存的 0 = 不可缓存；1 = 可缓存
[16]	R7_0_WT_WB	0x1	RW	区域 7 的子段 0 是回写的 0 = 直写；1 = 回写

比特位	名称	复位值	读写属性	功能说明
[15]	R6_3_CacheABLE	0x1	RW	区域 6 的子段 3 是可缓存的 0 = 不可缓存; 1 = 可缓存
[14]	R6_3_WT_WB	0x1	RW	区域 6 的子段 3 是回写的 0 = 直写; 1 = 回写
[13]	R6_2_CacheABLE	0x1	RW	区域 6 的子段 2 是可缓存的 0 = 不可缓存; 1 = 可缓存
[12]	R6_2_WT_WB	0x1	RW	区域 6 的子段 2 是回写的 0 = 直写; 1 = 回写
[11]	R6_1_CacheABLE	0x1	RW	区域 6 的子段 1 是可缓存的 0 = 不可缓存; 1 = 可缓存
[10]	R6_1_WT_WB	0x1	RW	区域 6 的子段 1 是回写的 0 = 直写; 1 = 回写
[9]	R6_0_CacheABLE	0x1	RW	区域 6 的子段 0 是可缓存的 0 = 不可缓存; 1 = 可缓存
[8]	R6_0_WT_WB	0x1	RW	区域 6 的子段 0 是回写的 0 = 直写; 1 = 回写
[7]	R2_3_CacheABLE	0x1	RW	区域 2 的子段 3 是可缓存的 0 = 不可缓存; 1 = 可缓存
[6]	R2_3_WT_WB	0x0	RW	区域 2 的子段 3 是回写的 0 = 直写; 1 = 回写
[5]	R2_2_CacheABLE	0x1	RW	区域 2 的子段 2 是可缓存的 0 = 不可缓存; 1 = 可缓存
[4]	R2_2_WT_WB	0x0	RW	区域 2 的子段 2 是回写的 0 = 直写; 1 = 回写
[3]	R2_1_CacheABLE	0x1	RW	区域 2 的子段 1 是可缓存的 0 = 不可缓存; 1 = 可缓存
[2]	R2_1_WT_WB	0x0	RW	区域 2 的子段 1 是回写的 0 = 直写; 1 = 回写
[1]	R2_0_CacheABLE	0x1	RW	区域 2 的子段 0 是可缓存的 0 = 不可缓存; 1 = 可缓存
[0]	R2_0_WT_WB	0x0	RW	区域 2 的子段 0 是回写的 0 = 直写; 1 = 回写

5.4.2.7 Cache 段 6 地址上限寄存器（CR6SXHA）

Cache 段 6 地址上限寄存器 CR6SXHA 用于定义 R6 的子地址段 X 的上限地址。
X = 0，1，2，3。

偏移地址：0x0040~0x004C 复位值：0x80*(X+1)-0x1，X = 0，1，2，3

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R6_SX_HIGH_ADDRESS[8:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R6_SX_HIGH_ADDRESS[8:0]	保留						
rw	ro						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留							
ro							

图表 5-8：Cache 段 6 地址上限寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23:15]	R6_SX_HIGH_ADDRESS[8:0]	0x0	RW	物理地址偏移量 重置值为 0x80*(X+1)-0x1，X = 0，1，2，3。 R6 中对应地址位小于此值的地址将作为可缓存。
[14:0]	保留	0x0	RO	---

5.4.2.8 Cache 段 6 地址下限寄存器（CR6SXLA）

Cache 段 6 地址下限寄存器 CR6SXLA 用于定义 R6 的子地址段 X 的下限地址。
X = 0，1，2，3。

偏移地址：0x0050~0x005C　　　　复位值：0x80*X，X = 0，1，2，3

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
R6_SX_LOW_ADDRESS[8:0]							
rw							

15	14	13	12	11	10	9	8
R6_SX_LOW_ADDRESS[8:0]	保留						
rw	ro						

7	6	5	4	3	2	1	0
保留							
ro							

图表 5-9：Cache 段 6 地址下限寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23:15]	R6_SX_LOW_ADDRESS[8:0]	0x0	RW	物理地址偏移量 重置值为 0x80*X，X = 0，1，2，3。 R6 中对应地址位大于或等于此值的地址将作为可缓存。
[14:0]	保留	0x0	RO	---

5.4.2.9 Cache 段 7 地址上限寄存器（CR7SXHA）

Cache 段 7 地址上限寄存器 CR7SXHA 用于定义 R7 的子地址段 X 的上限地址。
X = 0，1，2，3。

偏移地址：0x0060~0x006C　　　　复位值：0x80*(X+1)-0x1，X = 0，1，2，3

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
R7_SX_HIGH_ADDRESS[8:0]							
rw							

15	14	13	12	11	10	9	8
R7_SX_HIGH_ADDRESS[8:0]	保留						
rw	ro						

7	6	5	4	3	2	1	0
保留							
ro							

图表 5-10：Cache 段 7 地址上限寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23:15]	R7_SX_HIGH_ADDRESS[8:0]	0x0	RW	物理地址偏移量 重置值为 0x80*(X+1)-0x1，X = 0，1，2，3。 R7 中对应地址位小于此值的地址将作为可缓存。
[14:0]	保留	0x0	RO	---

5.4.2.10 Cache 段 7 地址下限寄存器（CR7SXLA）

Cache 段 7 地址下限寄存器 CR7SXLA 用于定义 R7 的子地址段 X 的下限地址。
X = 0，1，2，3。

偏移地址：0x0070~0x007C 复位值：0x80*X，X = 0，1，2，3

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
R7_SX_LOW_ADDRESS[8:0]							
rw							

15	14	13	12	11	10	9	8
R7_SX_LOW_ADDRESS[8:0]	保留						
rw	ro						

7	6	5	4	3	2	1	0
保留							
ro							

图表 5-11：Cache 段 7 地址下限寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23:15]	R7_SX_LOW_ADDRESS[8:0]	0x0	RW	物理地址偏移量 重置值为 0x80*X，X = 0，1，2，3。 R7 中对应地址位大于或等于此值的地址将作为可缓存。
[14:0]	保留	0x0	RO	---

5.4.2.11 Cache 段 2 地址上限寄存器（CR2SXHA）

Cache 段 2 地址上限寄存器 CR2SXHA 用于定义 R2 的子地址段 X 的上限地址。
X = 0，1，2，3。

偏移地址：0x0080~0x008C 复位值：0x4*(X+1)-1，X = 0，1，2，3

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留				R2_SX_HIGH_ADDRESS[4:0]			
ro				rw			

15	14	13	12	11	10	9	8
R2_SX_HIGH_ADDRESS[4:0]	保留						
rw	ro						

7	6	5	4	3	2	1	0
保留							
ro							

图表 5-12：Cache 段 2 地址上限寄存器

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:15]	R2_SX_HIGH_ADDRESS[4:0]	0x0	RW	物理地址偏移量 重置值为 0x4*(X+1)-1，X = 0，1，2，3。 R2 中对应地址位小于此值的地址将作为可缓存。
[14:0]	保留	0x0	RO	---

5.4.2.12 Cache 段 2 地址下限寄存器（CR2SXLA）

Cache 段 2 地址下限寄存器 CR2SXLA 用于定义 R2 的子地址段 X 下限地址。X = 0，1，2，3。

偏移地址：0x0090~0x009C 复位值：0x4*X，X = 0，1，2，3

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留				R2_SX_LOW_ADDRESS[4:0]			
ro				rw			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R2_SX_LOW_ADDRESS[4:0]	保留						
rw	ro						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留							
ro							

图表 5-13：Cache 段 2 地址下限寄存器

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:15]	R2_SX_LOW_ADDRESS[4:0]	0x0	RW	物理地址偏移量 重置值为 0x4*X，X = 0，1，2，3。 R2 中对应地址位大于或等于此值的地址将作为可缓存。
[14:0]	保留	0x0	RO	---

5.4.2.13 Cache 页清除地址寄存器（CPEA）

Cache 页清除地址寄存器用于设定所需清除页的基地址。在设定基地址和页大小并使能后，Cache 将根据基地址将所有对应 Cache 行施以 Cache 行的清除命令。

偏移地址：0x0180~0x0183 复位值：0x00000000

31	30	29	28	27	26	25	24
INVAL_BADDR[27:0]							
rw							

23	22	21	20	19	18	17	16
INVAL_BADDR[27:0]							
rw							

15	14	13	12	11	10	9	8
INVAL_BADDR[27:0]							
rw							

7	6	5	4	3	2	1	0
INVAL_BADDR[27:0]				保留			START_INVAL
rw				ro			rw

图表 5-14：Cache 页清除地址寄存器

比特位	名称	复位值	读写属性	功能说明
[31:4]	INVAL_BADDR[27:0]	0x0	RW	所需清除页的基地址的高 28 位 Cache 以行为单位，所以基地址无需关心低 4 位。
[3:1]	保留	0x0	RO	---
[0]	START_INVAL	0x0	RW	清除页开始 该位与 Cache 页清除大小寄存器的 START_INVAL 共享。可以先配置页大小或者页基地址，但需要在 START_INVAL 设置之前配置好页大小或者基地址。 0 = 无操作。读取 0 表示页清除操作完成 1 = 开始。读取 1 表示页清除操作是否进行中

5.4.2.14 Cache 页清除大小寄存器（CPES）

Cache 页清除大小寄存器用于设定所需清除页的空间大小。在设定基地址和页大小并使能后，Cache 将根据基地址将所有对应 Cache 行施以 Cache 行的清除命令。

偏移地址：0x0184~0x0187　　复位值：0x00000000

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

图表 5-15：Cache 页清除大小寄存器

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:4]	PAGE_SIZE[11:0]	0x0	RW	所需清除页的大小，12 位空间大小表示要清除的行数 Cache 以行为单位，所以页空间大小也无需关心低 4 位。
[3:1]	保留	0x0	RO	---
[0]	START_INVAL	0x0	RW	清除页开始 该位与 Cache 页清除页基地址寄存器的 START_INVAL 共享。可以先配置页大小或者页基地址，但需要在 START_INVAL 设置之前配置好页大小或者基地址。 0 = 无操作。读取 0 表示页清除操作完成 1 = 开始。读取 1 表示页清除操作是否进行中

5.4.2.15 Cache 时钟门控寄存器（CCG）

Cache 时钟门控寄存器用于控制 Cache 主逻辑时钟是否禁用。

偏移地址：0x0188~0x018B　　　　复位值：0x00000001

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							CLK_ENABLE
ro							rw

图表 5-16：Cache 时钟门控寄存器

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	CLK_ENABLE	0x0	RW	Cache 主逻辑时钟使能。没有 Cache 应用场景下，可作降低功耗使用 Cache 以行为单位，所以页空间大小也无需关心低 4 位。 0 = Cache 主逻辑时钟禁用 1 = Cache 主逻辑时钟使能

5.5 功能描述

5.5.1 缓存功能

该设备上的缓存的结构如下。数据和标签两个缓存都有 2 路组相联缓存结构，总大小为 8KB。缓存有 32 位地址和数据通路以及 16 字节行大小。缓存标签和数据存储使用外部单端口同步 RAM。

对于这些 8KB 缓存，缓存标签功能使用两个 256x 22 位 RAM 阵列，缓存数据功能使用两个 1024 x 32 位 RAM 阵列。Cache 标签条目存储 20 位高位地址以及对应于每行的已修改和有效的位。Cache 数据条目存储四个字节的代码或数据。所有常规缓存访问均使用物理地址。这导致以下缓存地址使用：

Cache-8KB 大小 = （256 组）x（16 字节行）x（2 路组关联）

标签：

标签中用于比较（命中）逻辑的地址[31:12]

地址[11:4]用于选择 256 组中的 1 组

地址[3:0]未使用

数据

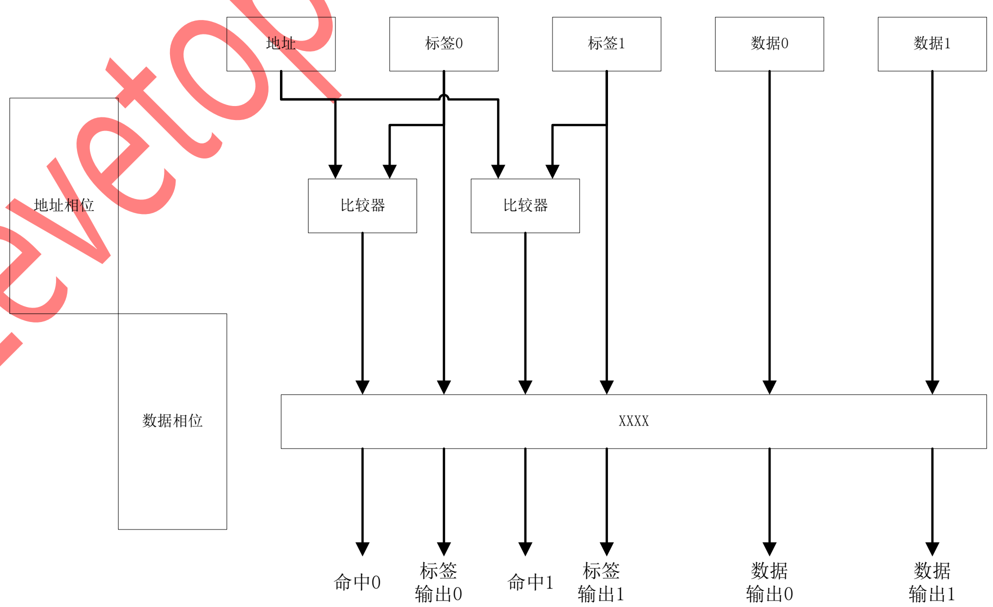
地址[31:12]未使用

地址[11:4]用于选择 256 组之一

address[3:2]用于选择行中的四个 32 位字之一

address[1:0]用于选择 32 位字中的字节

Cache 使用单周期并行标签和数据访问结构。这个结构在两级 AMBA_AHB 总线流水线中从地址相位的时钟边沿到数据阶段的时钟边沿对齐：



图表 5-17：Cache 标签和数据存取结构

在正常的可缓存操作中，完整地址在地址相位阶段寄存于 Cache 控制逻辑中，地址位[11:4]寄存在标签数组，地址位[11：2]寄存在数据数组中。在寻址阶段到数据阶段的完整周期中，标签和数据数组被访问并分析缓存是否命中。对于命中，所需的读取数据将在数据阶段返回，写操作从数据阶段周期开始就完成。对于未命中，数据根据需要维持住，同时 Cache 使用其从总线通过一个 Cache 行大小的数据传输向交叉总线开关访问所需的 Cache 行。

5.5.2 缓存控制

缓存在重置时被禁用。缓存标签和数据数组复位时不清除。因此，要启用缓存，缓存命令必须清除并初始化所需的标签数组位并进行配置和启用缓存。

5.5.3 缓存设置命令

Cache 设置命令可以在以下位置操作：

所有 WAY 0，
所有 WAY 1，或
所有这两个 WAY（完整的缓存）。

Cache 设置命令是使用 CCR 寄存器的高位启动的。Cache 组命令独立于缓存执行其操作使能位 CCR [ENCache]。

通过将 CCR [GO]位置 1 可启动 Cache 组命令。该位还充当组命令繁忙位。当组命令处于活动状态它保持设置状态，并且在组命令完成后由硬件清除。支持的缓存组命令在表格 5-2：中给出。组命令的工作方式为如下：

无效-无条件清除所存条目的有效和修改位。
推送-推送缓存条目（如果有效且已修改），然后清除修改位。如果条目无效或未修改，保持不变。
清除-如果缓存条目有效且已修改，则将其推送，然后清除有效的和修改位。如果条目无效或未修改，清除有效位。

表格 5-2：Cache 组命令

CCR[27:24]				Command
推送 WAY1	无效化 WAY1	推送 WAY0	无效化 WAY1	
0	0	0	0	无操作
0	0	0	1	无效化所有 WAY 0
0	0	1	0	推送所有 WAY 0
0	0	1	1	清除所有 WAY 0
0	1	0	0	无效化所有 WAY 1
0	1	0	1	无效化所有 WAY 1 无效化所有 WAY 0

CCR[27:24]				Command
推送 WAY1	无效化 WAY1	推送 WAY0	无效化 WAY1	
0	1	1	0	无效化所有 WAY 1 推送所有 WAY 0
0	1	1	1	无效化所有 WAY 1 清除所有 WAY 0
1	0	0	0	推送所有 WAY 1
1	0	0	1	推送所有 WAY 1 无效化所有 WAY 0
1	0	1	0	推送所有 WAY 1 推送所有 WAY 0
1	0	1	1	推送所有 WAY 1 清除所有 WAY 0
1	1	0	0	清除所有 WAY 1
1	1	0	1	清除所有 WAY 1 无效化所有 WAY 0
1	1	1	0	清除所有 WAY 1 推送所有 WAY 0
1	1	1	1	清除所有 WAY 1 清除所有 WAY 0

在复位后，使用 Cache 之前要先完成一个无效化 Cache 的命令。

5.5.4 Cache 行命令

Cache 行命令一次基于 Cache 中的一个行上运行。Cache 行命令可以使用物理地址或缓存地址执行。

Cache 地址由组地址和 WAY 选择组成。行命令作用于指定的 Cache 行。

具有物理地址的 Cache 行命令首先通过物理地址的位[11:4]搜索两个 WAY。如果命中，则命令按命中的方式执行。

Cache 行命令是使用 CLCR 寄存器的高位指定的。Cache 行命令独立于 Cache 启用位（CCR [ENCache]）在 Cache 上执行。使用 Cache 地址，命令可以是完全使用 CLCR 寄存器指定。使用实际物理地址，该命令还必须使用 CSAR 寄存器来指定物理地址。

Cache 行命令通过命令进行位（CLCR [LGO]或 CSAR [LGO]）执行。该位对于行命令也表示忙。它保持设置的话该命令处于活动状态，并且当该命令完成后被硬件清除。CLCR [27:24]位按以下方式选择行命令：

表格 5-3：Cache 行命令

CLCR[27:24]			Command
LACC	LADSEL	LCMD	
0	0	00	使用 Cache 地址和 WAY 号查询
0	0	01	使用 Cache 地址和 WAY 号无效化
0	0	10	使用 Cache 地址和 WAY 号推送
0	0	11	使用 Cache 地址和 WAY 号清除
0	1	00	使用物理地址和 WAY 号查询
0	1	01	使用物理地址和 WAY 号无效化
0	1	10	使用物理地址和 WAY 号推送
0	1	11	使用物理地址和 WAY 号清除
1	0	00	使用 Cache 地址和 WAY 号写
1	0	01	保留，无操作
1	0	10	保留，无操作
1	0	11	保留，无操作
1	1	xx	保留，无操作

5.5.5 使用 Cache 地址执行一系列的行命令

一系列带有增量缓存地址的行命令可以通过以下方式只写 CLCR 执行：

在 CLCR[27:24]中设置命令，
根据需要设置控制 WAY 号（CLCR [WSEL]）和标签/数据（CLCR [TDSEL]），
在 CLCR [CacheADDR]放入 Cache 地址，然后
设置行命令执行位（CLCR [LGO]）。
当一个行命令完成时，请遵循以下步骤启动下一条命令：
增加缓存地址（从第 2 位开始逐步访问数据，或从第 4 位开始逐步访问行），并且
设置行命令执行位（CLCR [LGO]）。

5.5.6 使用物理地址执行一系列行命令

执行一系列带有递增物理地址的行命令使用以下步骤：

写入 CLCR。
将命令放在 CLCR[27:24]中
设置标签/数据（CLCR [TDSEL]）控制
将物理地址放在 CSAR [PHYADDR]中，并将行命令设置为 go 位（CSAR [LGO]）。

当一行命令完成时，请遵循以下步骤启动下一条命令：

递增物理地址（在第 2 位以步进数据或在第 4 位以访问行），并且
设置行命令执行位（CSAR [LGO]）。

行命令执行位在 CLCR 和 CSAR 寄存器之间共享，因此以上步骤可通过一次写入 CSAR 寄存器来完成。

5.5.7 行命令结果

在完成一条行命令后，CLCR 的 LCR 位寄存器包含命令对应行初始状态信息。对于行地址命令，该信息在行命令从目标行执行前被读取。对于具有物理地址的行命令，这在从命令行执行行命令操作之前，在命中时信息在行命令执行之前读取，如果命令未命中，则讲初始有效位清除。一般来说，如果有效指示位（LCR [LCIVB]已清除，目标行在行命令的开始是无效的并且不会执行任何行操作。

表格 5-4：Cache 行命令结果

LCR[2:0]			对于 Cache 地址命令	对于物理地址命令
LCWAY	LCIMB	LCIVB		
0	0	0	WAY 0 行无效	未命中
0	0	1	WAY 0 有效，未修改	WAY 0 有效，未修改
0	1	0	WAY 0 行无效	未命中
0	1	1	WAY 0 有效，有修改	WAY 0 有效，有修改
1	0	0	WAY 1 行无效	未命中
1	0	1	WAY 1 有效，未修改	WAY 1 有效，未修改
1	1	0	WAY 1 行无效	未命中
1	1	1	WAY 1 有效，有修改	WAY 1 有效，有修改

在完成除写以外的其他命令后，CCVR（缓存读/写值寄存器）包含目标命令对应有关行标签的初始状态的信息。对于行命令，CLCR [TDSEL]在标签和数据之间选择。如果行命令使用的是物理地址但未命中，则该数据无关紧要。对于写命令，CCVR 保留写数据。

6 时钟电源管理模块（CPM）

6.1 概述

时钟电源管理模块包含：

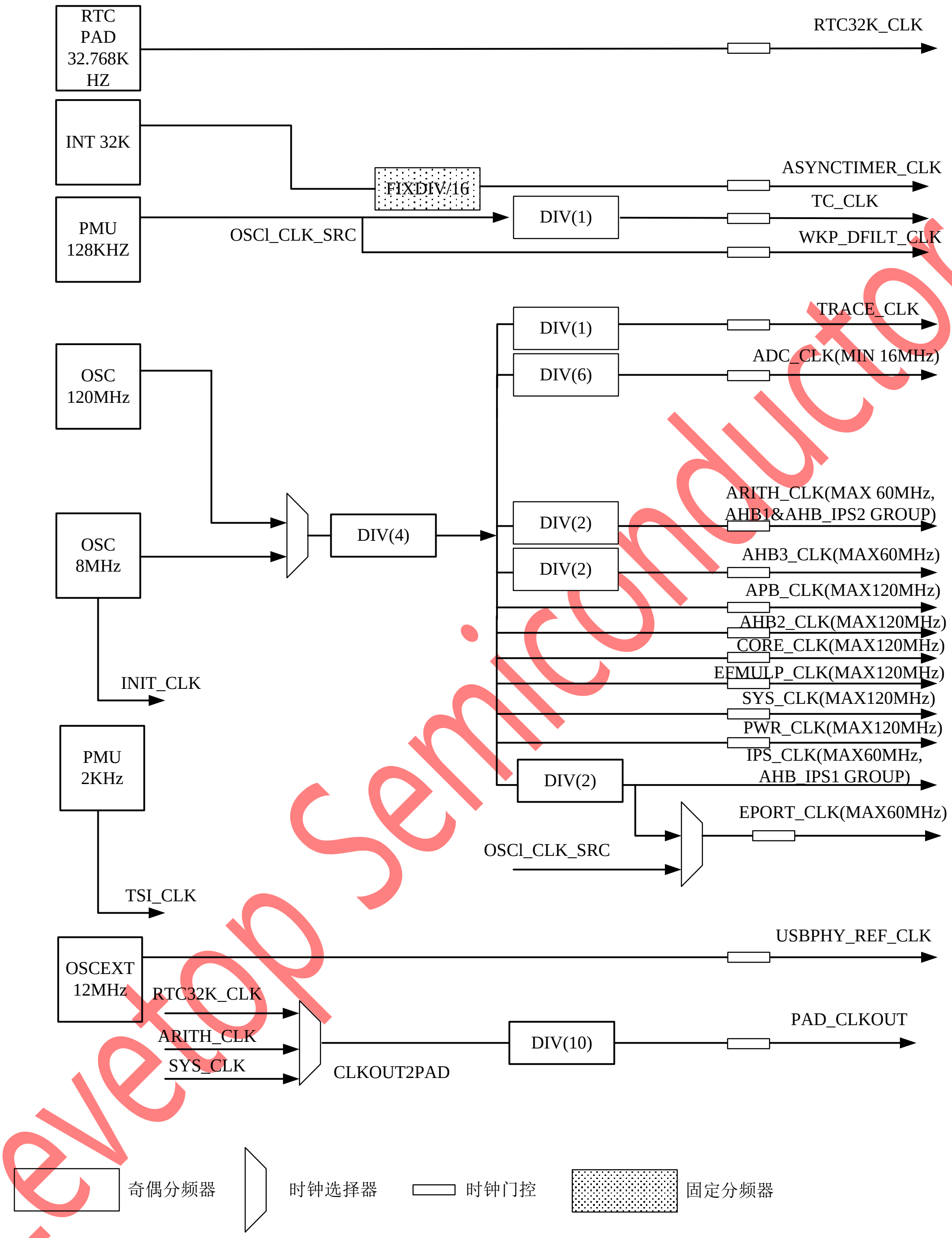
- 内部高速振荡器，频率为 120MHz
- 内部低速振荡器，频率为 8MHz
- 外部晶振，频率为 12MHz
- 状态和控制寄存器
- 时钟和电源控制逻辑

6.2 特性

时钟和电源管理模块特性包括：

- 两个系统时钟源
- 内部高速振荡器，频率为 120MHz
- 内部低速振荡器，频率为 8MHz
- 支持低功耗模式
- 独立的时钟分频设置
- 独立的模块时钟开关

6.3 框图



图表 6-1：时钟控制结构图

6.4 运行模式

6.4.1 低功耗模式操作

6.4.1.1 等待（wait）和瞌睡（doze）操作

通过等待和瞌睡操作，外设及嵌入式 flash 的时钟是使能的，而 CPU 的时钟是被禁止的。

6.4.1.2 停止（Stop）和睡眠（sleep）操作

通过 stop 和 sleep 操作，系统会根据 SLPCFGR 及 SLPCFGR2 寄存器的配置进入特定的功耗状态。

6.4.2 详细的功耗模式和唤醒源

本芯片有四种功耗模式：等待（wait）&瞌睡（doze）模式，低功耗（lowpower）模式，保持（retention）模式，休眠（hibernation/power_off2）模式。详细请参考下表。

表格 6-1：CPM 电源模式和唤醒源

主系统功耗模式	时 钟	唤 醒 源	电 源（注 1）
RUN	所有时钟工作 可配置门控	无	支持 voltage scaling （MPW 1.1V/0.9V） （工程批 1.2V / 1.15V / 1.1V / 1.05V / 0.9V）
WAIT	M4 时钟停止 其他时钟工作	所有中断 WK PAD（高电平） USBDET（高电平） POR（复位） Async Timer 中断	AO 区域 ON WK 区域 ON PD 区域 ON USB 区域可配 Async Timer ON
LOWPOWER	所有时钟停止 （唤醒需要的时钟，以及 RTC32K 除外）	SPI（SS3 PAD，低电平） USI1 I2C1-3 EPORT0-4（高低电平，支持边沿） USB RESUME （USB 区域配置为 ON） TSI TOUCH SDDC RTC 定时唤醒（支持脉冲&闹钟） WK PAD（高电平）	AO 区域 ON WK 区域 ON PD 区域 ON USB 区域可配 Async Timer ON

主系统功耗模式	时 钟	唤 醒 源	电 源（注 1）
		USBDET（高电平） POR（复位，低电平） Async Timer 中断	
RETENTION (POWEROFF1.0)	所有时钟停止 (唤醒需要的时钟， 以及 RTC32K 除外)	SPI（SS3 PAD，低电平） USI1 I2C1 EPORT0 (高低电平，支持边沿) USB RESUME (USB 区域配置为 ON) TSI TOUCH SDDC RTC 定时唤醒（仅支持天、小时、分钟、秒脉冲，工程批） WK PAD（高电平） USBDET（高电平） POR（复位，低电平） Async Timer 中断	AO 区域 ON WK 区域 ON PD 区域 OFF USB 区域可配 Async Timer ON
DEEP SLEEP (POWEROFF1.5)	所有时钟停止 (唤醒需要的时钟，以及 RTC32K 除外)	SPI（SS3 PAD，低电平） USI1 I2C1 EPORT0 (高低电平，支持边沿) USB RESUME (USB 区域配置为 ON) TSI TOUCH RTC 定时唤醒（仅支持天、小时、分钟、秒脉冲，工程批） WK PAD（高电平） USBDET（高电平） POR（复位，低电平） Async Timer 中断	AO 区域 ON WK 区域 OFF PD 区域 OFF USB 区域可配 Async Timer ON
HIBER (POWEROFF2.0)	所有时钟停止 (RTC32K 除外)	WK PAD（高电平） USBDET（高电平） POR（复位，低电平）	AO 区域 OFF WK 区域 OFF PD 区域 OFF USB 区域 OFF Async Timer ON
Async Timer	所有时钟停止 (RTC32K 除外)	无	AO 区域 OFF WK 区域 OFF

主系统功耗模式	时 钟	唤 醒 源	电 源（注 1）
			PD 区域 OFF USB 区域 OFF Async Timer ON

注 1：

AO 区域模块包括：CPM，CCM&RESET，IO_CTRL，IOMUX_AO，EPORT1，USI1，I2C1，TSI；

WK 区域模块包括：SRAMD （32KB，特指 MEMORY IP，不包括控制器），KEY_CTRL，SDDC；

USB 区域模块包括：USB；

Async Timer 区域模块包括 Async Timer，NVRAM；

PD 区域包括：M4 + 其他模块。

区域电源为 ON，表示低功耗唤醒后，区域内寄存器状态为进入低功耗之前的状态；

区域电源为 OFF，表示低功耗唤醒后，区域内寄存器状态为复位状态；

特别的，PD 区域模块中，M4 使用 RETENTION 技术，在 RETENTION 模式唤醒后，寄存器状态为进入低功耗之前的状态，其他 PD 区域模块在 RETENTION 模式唤醒后，寄存器状态为复位状态。另外，在 DEEP SLEEP 和 HIBER 模式唤醒后，M4 的状态为复位状态。

6.5 内存映射和寄存器

6.5.1 内存映射

表格 6-2：CPM 内存映射

偏移地址	位 31:16	位 15:0	访问权限
0x0000	睡眠配置寄存器（SLPCFGR）		S
0x0004	睡眠控制寄存器（SLPCR）		S
0x0008	系统时钟分频寄存器（SCDIVR）		S
0x000c	外设时钟分频寄存器 1（PCDIVR1）		S
0x0010	外设时钟分频寄存器 2（PCDIVR2）		S
0x0014	外设时钟分频寄存器 3（PCDIVR3）		S
0x0018	时钟分频更新寄存器（CDIVUPDR）		S
0x001C	时钟分频使能寄存器（CDIVENR）		S
0x0020	晶体振荡器控制及状态寄存器（OCSR）		S
0x0024	时钟切换配置寄存器（CSWCFGR）		S
0x0028	核计时寄存器(CTICKR)		S
0x002C	芯片配置寄存器(CHIPCFGR)		S
0x0030	电源控制寄存器（PWRCR）		S
0x0034	睡眠计数寄存器（SLPCNTR）		S
0x0038	唤醒计数寄存器（WKPCNTR）		S
0x003C	多时钟门控制寄存器（MULTICGTCR）		S
0x0040	系统时钟门控制寄存器（SYSCGTCR）		S
0x0044	AHB3 时钟门控制寄存器（AHB3CGTCR）		S
0x0048	算法时钟门控制寄存器（ARITHCGTCR）		S
0x004C	IPS 时钟门控制寄存器（IPSCGTCR）		S
0x0050	VCC 通用校准寄存器（VCCGTRIMR）		S
0x0054	VCC 低压阈值检测校准寄存器（VCCLTRIMR）		S
0x0058	VCC 参考电压校准寄存器（VCCVTRIMR）		S
0x005C	VCC 核测试模式寄存器（VCCCTMR）		S
0x0060	OSC8M 校准寄存器（O8MTRIMR）		S
0x0064	保留		S
0x0068	OSC120M 校准寄存器（O120MTRIMR）		S
0x006C	CARDLDO 校准寄存器（CARDTRIMR）		S
0x0070	OSCL 稳定时间寄存器（OSCLSTIMER）		S
0x0074	OSCH 稳定时间寄存器（OSCHSTIMER）		S
0x0078	OSCE 稳定时间寄存器（OSCESTIMER）		S
0x007C	电源状态寄存器（PWRSR）		S
0x0080	保留		S

偏移地址	位 31:16	位 15:0	访问权限
0x0084	保留		S
0x0088	保留		S
0x008C	RTC 校准寄存器（RTCTRIMR）		S
0x0090	管脚唤醒中断控制寄存器(PADWKPINTCR)		S
0x0094	唤醒滤波计数寄存器(WKPFILTCNTR)		S
0x0098	CARD 上电计数寄存器(CARDPOCR)		S
0x009C	RTC32K 稳定时间寄存器（RTCSTIMER）		S
0x00A0	存储器掉电睡眠控制寄存器(MPDSLPCR)		S
0x00A4	保留		S
0x00A8	保留		S
0x00AC	多复位控制寄存器（MULTIRSTCR）		S
0x00B0	系统复位控制寄存器（SYSRSTCR）		S
0x00B4	AHB3 复位控制寄存器（AHB3RSTCR）		S
0x00B8	算法复位控制寄存器（ARITHRSTCR）		S
0x00BC	IPS 复位控制寄存器（IPSRSTCR）		S
0x00C0	睡眠配置寄存器 2（SLPCFGR2）		S
0x00C4	保留		S
0x00C8	保留		S
0x00CC	保留		S
0x00D0	掉电计数寄存器(PDNCNTR)		S
0x00D4	上电计数寄存器(PONCNTR)		S
0x00D8	管脚 SS3 控制寄存器(PADSS3CR)		S
0x00DC	唤醒源控制寄存器（WKPSCR）		S

注意：

S = 超级用户访问。普通户用访问超级用户地址是无效的，且会产生一个时钟周期的传输错误。

6.5.2 寄存器描述

6.5.2.1 睡眠配置寄存器（SLPCFGR）

偏移地址：0x0000 ~ 0x0003　　　　复位值：0x1D6F8879

31	30	29	28	27	26	25	24
SLEEP_MODE[1:0]		保留	EPORT4_SLPEN	EPORT3_SLPEN	EPORT2_SLPEN	保留	
rw		ro	rw	rw	rw	ro	

23	22	21	20	19	18	17	16
保留	EPORT1_SLPEN	EPORT_SLPEN	OSCEXT_SLPEN	PMU128K_SLPEN	V33_HP_LPEN	HP_FLASH_LPEN	RTC32K_SLPEN
ro	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
CARD0PD	保留	CARD0PO		保留			
rw	ro	rw		ro			

7	6	5	4	3	2	1	0
VRFlashPD	保留		FLASH_IPSLP	保留			
rw	ro		rw	ro			

图表 6-2：睡眠配置寄存器（SLPCFGR）

比特位	名称	复位值	读写属性	功能说明
[31:30]	SLEEP_MODE[1:0]	0x0	RW	当停止时系统进入睡眠模式 00 = 进入低耗模式（当 SLPCFGR2 寄存器的 VDD_WK_SWOFF = 0 且 VDD_PD_RET = 0） 01 = 进入保持模式（当 SLPCFGR2 寄存器的 VDD_WK_SWOFF = 0 且 VDD_PD_RET = 1） 01 = 进入深度睡眠模式（当 SLPCFGR2 寄存器的 VDD_WK_SWOFF = 1 且 VDD_PD_RET = 0） 1x = 进入休眠模式（当 SLPCFGR2 寄存器的 VDD_WK_SWOFF = 0 且 VDD_PD_RET = 0）
[29]	保留	0x0	RO	---
[28]	EPORT4_SLPEN	0x1	RW	当停止时 EPORT4 模块时钟睡眠使能 配置在系统睡眠模式下 EPORT4 模块时钟是否使能且自动切换到 OSCL 时钟 0 = 禁止； 1 = 使能

比特位	名称	复位值	读写属性	功能说明
[27]	EPORT3_SLPEN	0x1	RW	当停止时 EPORT3 模块时钟睡眠使能 配置在系统睡眠模式下 EPORT3 模块时钟是否使能且自动切换到 OSCL 时钟 0 = 禁止；1 = 使能
[26]	EPORT2_SLPEN	0x1	RW	当停止时 EPORT2 模块时钟睡眠使能 配置在系统睡眠模式下 EPORT2 模块时钟是否使能且自动切换到 OSCL 时钟 0 = 禁止；1 = 使能
[25:23]	保留	0x2	RO	---
[22]	EPORT1_SLPEN	0x1	RW	当停止时 EPORT1 模块时钟睡眠使能 配置在系统睡眠模式下 EPORT1 模块时钟是否使能且自动切换到 OSCL 时钟 0 = 禁止；1 = 使能
[21]	EPORT_SLPEN	0x1	RW	当停止时 EPORT 模块时钟睡眠使能 配置在系统睡眠模式下 EPORT 模块时钟是否使能且自动切换到 OSCL 时钟 0 = 禁止；1 = 使能
[20]	OSCEXT_SLPEN	0x0	RW	外部晶振时钟睡眠使能 配置在系统睡眠模式下外部晶振时钟是否使能。 0 = 禁止；1 = 使能
[19]	PMU128K_SLPEN	0x1	RW	PMU128K 时钟睡眠使能 配置在系统睡眠模式下 PMU128K 时钟是否使能。 0 = 禁止；1 = 使能
[18]	V33_HP_LPEN	0x1	RW	当停止时 VDD33 或 VDD33_FLASH LDO 进入低耗状态 配置进入系统睡眠模式下 VDD33 或 VDD33_FLASH LDO 是否进入低耗状态。 0 = 禁止 VRFlash 3.3V LDO 进入低耗 1 = 使能 VRFlash 3.3V LDO 进入低耗
[17]	HP_FLASH_LPEN	0x1	RW	当停止时内部 FLASH HP LDO 掉电 配置进入系统睡眠模式下 FLASH HP LDO 是否进入低耗状态。 0 = 禁止 FLASH HP LDO 进入低耗 1 = 使能 FLASH HP LDO 进入低耗

比特位	名称	复位值	读写属性	功能说明
[16]	RTC32K_SLPEN	0x1	RW	RTC32K 时钟睡眠使能 配置在系统睡眠模式下 RTC32K 时钟是否使能。 0 = 禁止；1 = 使能
[15]	CARD0PD	0x1	RW	CADR0 LDO 断电配置 配置 CARD0 LDO 是否断电。 0 = 上电；1 = 断电
[14]	保留	0x0	RO	---
[13:12]	CARD0PO	0x0	RW	CARD0 LDO 电压输出配置 配置 CARD0 LDO 输出电压 00 = 1.8V；01 = 3.0V 10 = 3.3V；11 = 5V
[11:8]	保留	0x8	RO	---
[7]	VRFlashPD	0x0	RW	当停止时 VDD33 或 VDD33_FLASH LDO 断电配置 配置在系统睡眠模式下 VDD33 或 VDD33_FLASH LDO 是否断电。 0 = 上电；1 = 断电
[6:5]	保留	0x3	RO	---
[4]	FLASH_IPSLP	0x1	RW	FLASH IP 睡眠配置 配置在系统睡眠模式下 FLASH IP 是否立即进入睡眠模式。 0 = 不进入睡眠模式；1 = 进入睡眠模式
[3:0]	保留	0x9	RO	---

6.5.2.2 睡眠控制寄存器（SLPCR）

偏移地址：0x0004 ~ 0x0007 复位值：0x00000000

31	30	29	28	27	26	25	24
SLP_CFG_KEY[1:0]		SLP_CFG_MODE	保留				
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							
ro							

图表 6-3：睡眠控制寄存器（SLPCR）

比特位	名称	复位值	读写属性	功能说明
[31:30]	SLP_CFG_KEY[1:0]	0x0	RW	睡眠配置模式密钥 如果 VCCVTRIMR 寄存器的 SLP_PROTECT_EN 比特位设置为 1，SLPCFGR 寄存器就不能被改写，除非按正确的顺序写这两个比特位。正确的顺序就是 2’b01->2’b10->2’b11。按这个顺序写这两个比特位，其值就是 2’b11，那么 SLPCFGR 寄存器就可以被改写。当这个两个比特位为 2’b11 时，只有写 2’b00 才可以清除。写其他值没有效果且还保持 2’b11。
[29]	SLP_CFG_MODE	0x0	RO	睡眠配置模式 SLP_CFG_KEY 按上述顺序写之后，其值变为 1。
[28:0]	保留	0x0	RO	---

6.5.2.3 系统时钟分频寄存器（SCDIVR）

偏移地址：0x0008 ~ 0x000B　　　　复位值：0x00090003

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLKOUT_DIV[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRACE_DIV[7:0]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SYS_DIV[7:0]							
rw							

图表 6-4：系统时钟分频寄存器（SCDIVR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23:16]	CLKOUT_DIV[7:0]	0x9	RW	CLKOUT 分频，分频系数 = CLKOUT_DIV +1 CDIVUPDR 寄存器 SYSDIV_UPDATE 比特位写 1 就会更新 SCDIVR 寄存器的分频系数到 DIVIDER。读永远返回 0。
[15:8]	TRACE_DIV[7:0]	0x0	RW	TRACE 时钟分频，分频系数 = TRACE_DIV + 1 CDIVUPDR 寄存器 SYSDIV_UPDATE 比特位写 1 就会更新 SCDIVR 寄存器的分频系数到 DIVIDER。读永远返回 0。
[7:0]	SYS_DIV[7:0]	0x3	RW	系统时钟分频，分频系数 = SYS_DIV+1 CDIVUPDR 寄存器 SYSDIV_UPDATE 比特位写 1 就会更新 SCDIVR 寄存器的分频系数到 DIVIDER。读永远返回 0。

注意：

如果寄存器被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 11（OVERWR_SCDIV_TRIM）置 1。

6.5.2.4 外设时钟分频寄存器 1（PCDIVR1）

偏移地址：0x000C ~ 0x000F　　　　复位值：0x50E21111

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ARITH_DIV[3:0]				AHB3_DIV[3:0]			
rw				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留				IPS_DIV[3:0]			
ro				rw			

图表 6-5：外设时钟分频寄存器 1（PCDIVR1）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x50E2	RO	---
[15:12]	ARITH_DIV[3:0]	0x1	RW	算法时钟分频，分频系数 = ARITH_DIV+1 CDIVUPDR 寄存器 PERDIV_UPDATE 比特位写 1 就会更新 PCDIVR1，PCDIVR2，PCDIVR3 寄存器的分频系数到 DIVIDER。读永远返回 0。
[11:8]	AHB3_DIV[3:0]	0x1	RW	AHB3 时钟分频，分频系数 = AHB3_DIV+1 CDIVUPDR 寄存器 PERDIV_UPDATE 比特位写 1 就会更新 PCDIVR1，PCDIVR2，PCDIVR3 寄存器的分频系数到 DIVIDER。读永远返回 0。
[7:4]	保留	0x1	RO	---
[3:0]	IPS_DIV[3:0]	0x1	RW	IPS 时钟分频，分频系数 = IPS_DIV+1 CDIVUPDR 寄存器 PERDIV_UPDATE 比特位写 1 就会更新 PCDIVR1，PCDIVR2，PCDIVR3 寄存器的分频系数到 DIVIDER。读永远返回 0。

注意：

如果寄存器被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 10（OVERWR_PCDIV_TRIM）置 1。

6.5.2.5 外设时钟分频寄存器 2（PCDIVR2）

偏移地址：0x0010 ~ 0x0013　　复位值：0x0003E12B

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

图表 6-6：外设时钟分频寄存器 2（PCDIVR2）

比特位	名称	复位值	读写属性	功能说明
[31:28]	TC_DIV[3:0]	0x0	RW	TC 时钟分频，分频系数 = TC_DIV+1 CDIVUPDR 寄存器 PERDIV_UPDATE 比特位写 1 就会更新 PCDIVR1，PCDIVR2，PCDIVR3 寄存器的分频系数到 DIVIDER。读永远返回 0。
[27:24]	保留	0x0	RO	---
[23:0]	保留	0x3E12B	RO	---

6.5.2.6 外设时钟分频寄存器 3（PCDIVR3）

偏移地址：0x0014 ~ 0x0017 复位值：0x02180EBE

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

图表 6-7：外设时钟分频寄存器 3（PCDIVR3）

比特位	名称	复位值	读写属性	功能说明
[31:0]	保留	0x02180EBE	RO	---

6.5.2.7 时钟分频更新寄存器（CDIVUPDR）

偏移地址：0x0018 ~ 0x001B　　复位值：0x00000000

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

图表 6-8：时钟分频更新寄存器（CDIVUPDR）

比特位	名称	复位值	读写属性	功能说明
[31:2]	保留	0x0	RO	---
[1]	SYSDIV_UPD	0x0	RW	系统时钟分频系数更新 SYSDIV_UPD 比特位写 1，配置在 SCDIVR 和 CDIVENR 的时钟分频系数和分频使能才会有效。读返回 0。
[0]	PERDIV_UPD	0x0	RW	PERDIV_UPD 比特位写 1，配置在 PCDIVR1，PCDIVR2，PCDIVR3 和 CDIVENR 的时钟分频系数和分频使能才会有效。读返回 0。

6.5.2.8 时钟分频使能寄存器（CDIVENR）

偏移地址：0x001C ~ 0x001F　　　　复位值：0x001fffff

31	30	29	28	27	26	25	24
保留							
ro							
23	22	21	20	19	18	17	16
保留							
ro							
15	14	13	12	11	10	9	8
CLKOUT_DIVEN	TRACE_DIVEN	TC_DIVEN	保留	保留	ADC_DIVEN	保留	保留
rw	rw	rw	ro	ro	rw	ro	ro
7	6	5	4	3	2	1	0
保留				ARITH_DIVEN	AHB3_DIVEN	保留	IPS_DIVEN
ro				rw	rw	ro	rw

图表 6-9：时钟分频使能寄存器（CDIVENR）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x1f	RO	---
[15]	CLKOUT_DIVEN	0x1	RW	CLKOUT 分频时能 配置 CLKOUT 分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频
[14]	TRACE_DIVEN	0x1	RW	TRACE 时钟分频时能 配置 TRACE 时钟分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频
[13]	TC_DIVEN	0x1	RW	TC 时钟分频时能 配置 TC 时钟分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频
[12]	保留	0x1	RO	---

比特位	名称	复位值	读写属性	功能说明
[11]	保留	0x1	RO	---
[10]	ADC_DIVEN	0x1	RW	ADC 时钟分频时能 配置 ADC 时钟分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频
[9]	保留	0x1	RO	---
[8]	保留	0x1	RO	---
[7:4]	保留	0xF	RO	---
[3]	ARITH_DIVEN	0x1	RW	算法时钟分频时能 配置算法时钟分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频
[2]	AHB3_DIVEN	0x1	RW	AHB3 时钟分频时能 配置 AHB3 时钟分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频
[1]	保留	0x1	RO	---
[0]	IPS_DIVEN	0x1	RW	IPS 时钟分频时能 配置 IPS 时钟分频是否使能。当分频不使能时，没有时钟从分频器输出。CDIVUPDR 寄存器的对应 UPDATA 比特位写 1，就会更新配置的使能位到 DIVIDER。 0 = 禁止时钟分频；1 = 使能时钟分频

6.5.2.9 晶体振荡器控制和状态寄存器（OCSR）

偏移地址：0x0020 ~ 0x0023　　　　复位值：0x00000B6B/00006B6B

31	30	29	28	27	26	25	24
保留				TRNG_OSCEN[3:0]			
ro				rw			

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留	PMU2K_VALID	RTC32K_STABLE	OSCEXT_STABLE	OSC120M_STABLE	USBPHY240M_STABLE	PMU128K_STABLE	OSC8M_STABLE
ro	ro	ro	ro	ro	ro	ro	ro

7	6	5	4	3	2	1	0
保留	PMU2K_EN	RTC32K_EN	OSCEXT_EN	OSC120M_EN	USBPHY240M_EN	PMU128K_EN	OSC8M_EN
ro	rw	rw	rw	rw	rw	rw	rw

图表 6-10：晶体振荡器控制和状态寄存器（OCSR）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:24]	TRNG_OSCEN[3:0]	0x0	RW	时钟源使能 0 = 时钟源禁止；1 = 时钟源使能
[23:15]	保留	0x0	RO	---
[14]	PMU2K_VALID	0x0/1	RO	时钟源稳定 0 = 时钟源不稳定；1 = 时钟源稳定
[13:8]	*_STABLE	0x2B/0B	RO	时钟源稳定 0 = 时钟源不稳定；1 = 时钟源稳定
[7]	保留	0x0	RO	---
[6:0]	*_EN	0x6b	RW	时钟源使能 0 = 时钟源禁止；1 = 时钟源使能

注意：

如果寄存器 TRNG_OSCEN 和*_EN 比特位被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 9（OVERWR_OSCR_TRIM）置 1。

6.5.2.10 时钟切换配置寄存器（CSWCFGR）

偏移地址：0x0024 ~ 0x0027 复位值：0x10515201

31	30	29	28	27	26	25	24
CLKOUT_SEL_ST				保留		CLKOUT_SEL[1:0]	
ro				ro		rw	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留						SYS_SEL_ST[1:0]	
ro						ro	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留							SYS_SEL
ro							rw

图表 6-11：时钟切换配置寄存器（CSWCFGR）

比特位	名称	复位值	读写属性	功能说明
[31:28]	CLKOUT_SEL_ST	0x1	RO	CLKOUT 时钟选择状态位 这些比特位反映了时钟切换时哪个时钟源被选择。 0000 = 选择系统时钟 0001 = 选择算法时钟 0010 = 保留 0011 = 选择 RTC32K 时钟 0100~1111 = 保留
[27:26]	保留	0x0	RO	---
[25:24]	CLKOUT_SEL[1:0]	0x0	RW	CLKOUT 时钟输出选择 00 = 选择系统时钟 01 = 选择算法时钟 10 = 保留 11 = 选择 RTC32K 时钟
[23:10]	保留	0x1454	RO	---
[9:8]	SYS_SEL_ST[1:0]	0x2	RO	系统时钟选择状态位 这些比特位反映了时钟切换时哪个时钟源被选择。 00 = 保留；01 = 选择 OSC8M 10 = 选择 OSC120M；11 = 保留
[7:1]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[0]	SYS_SEL	0x1	RW	系统时钟源选择 当软件想要切换系统时钟源时，必须确保当前选择的时钟源和要选择的时钟源都有效 0 = 选择 OSC8M 作为系统时钟源 1 = 选择 OSC120M 作为系统时钟源

注意：

如果寄存器 CLKOUT_SEL 和 SYS_SEL 比特位被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 29（OVERWR_CSWCFG_TRIM）置 1。

6.5.2.11 核计时寄存器（CTICKR）

偏移地址：0x0028 ~ 0x002B　　复位值：0x01000147

31	30	29	28	27	26	25	24
保留						NO_REF	SKEW
ro						rw	rw

23	22	21	20	19	18	17	16
CALIB[23:16]							
rw							

15	14	13	12	11	10	9	8
CALIB[15:8]							
rw							

7	6	5	4	3	2	1	0
CALIB[7:0]							
rw							

图表 6-12：核计时寄存器（CTICKR）

比特位	名称	复位值	读写属性	功能说明
[31:26]	保留	0x0	RO	---
[25]	NO_REF	0x0	RW	没有备用参考比特 表示没有集成备用参考时钟源 0 = RTC32K 被选择作为参考时钟源 1 = 核时钟被选择作为参考时钟源
[24]	SKEW	0x1	RW	偏移比特 0 = 保证 10ms 的精确倍数 1 = 不保证 10ms 的精确倍数
[23:0]	CALIB[23:0]	0x147	RW	标定位 提供一个整数值来计算 10ms 的延迟

6.5.2.12 芯片控制寄存器（CHIPCFGR）

偏移地址：0x002C ~ 0x002F 复位值：0x3e8dfbf7

31	30	29	28	27	26	25	24
USBPHY_OSC_MODE[1:0]		USBPHY_CFG_SRM	USBPHY_PLL_SRM	保留	USBPHY_O_ISOEN	USBPHY_RSTMASK	USBPHY_PSWEN
rw		rw	rw	ro	rw	rw	rw

23	22	21	20	19	18	17	16
USBPHY_IP_SRM	M2S_INT	S2M_INT	S2M_INT_CLR	S2M_O_ISOEN	S2M_PST_ISOEN	ATIMER_H2L_ISOSEL	ATIMER_H2L_ISOEN
rw	rw	ro	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
保留	RTC1S_CLK_GTE	RTC1K_CLK_GTE	RTC32K_CLK_GTE	RTC32K_ISO_GTE	S2M_INT_EN	保留	
ro	rw	rw	rw	rw	rw	ro	

7	6	5	4	3	2	1	0
保留							
ro							

图表 6-13：芯片控制寄存器（CHIPCFGR）

比特位	名称	复位值	读写属性	功能说明
[31:30]	USBPHY_OSC_MODE[1:0]	0x0	RW	选择 USBPHY 振荡器模式 00 = 自动检波振荡器 01 = 自动检波振荡器（仅为了快速仿真） 10 = 选择内部振荡器 11 = 选择外部振荡器
[29]	USBPHY_CFG_SRM	0x1	RW	USBPHY_CFG 软件复位掩码 设置 USBPHY_CFG_SRM 比特位到复位对应的电路
[28]	USBPHY_PLL_SRM	0x1	RW	USBPHY_PLL 软件复位掩码 设置 USBPHY_PLL_SRM 比特位到复位对应的电路
[27]	保留	0x1	RO	---
[26]	USBPHY_O_ISOEN	0x1	RW	USBPHY 输出隔离使能 0 = 禁止；1 = 使能
[25]	USBPHY_RSTMASK	0x1	RW	USBPHY 复位信号掩码 0 = 复位没有掩码；1 = 复位掩码申明
[24]	USBPHY_PSWEN	0x0	RW	USBPHY 电源切换使能 0 = 禁止；1 = 使能

比特位	名称	复位值	读写属性	功能说明
[23]	USBPHY_IP_SRM	0x1	RW	USBPHY_IP 软件复位掩码 设置 USBPHY_IP_SRM 比特位到复位对应的电路
[22]	保留	0x0	RO	---
[21]	保留	0x0	RO	---
[20]	保留	0x0	RO	---
[19]	保留	0x0	RO	---
[18]	保留	0x0	RO	---
[17]	ATIMER_H2L_ISOSEL	0x0	RW	ATIMER H2L 隔离选择 这个比特表明为 ATIMER H2L 隔离选择哪一个隔离位 0 = 选择 ATIMER 模块的控制位 1 = 选择 CPM 模块的 ATIMER_H2L_ISOEN 控制位
[16]	ATIMER_H2L_ISOEN	0x1	RW	ATIMER H2L 隔离使能 0 = 禁止；1 = 使能
[15]	保留	0x1	RO	---
[14]	RTC1S_CLK_GTE	0x1	RW	RTC1S 时钟使能 0 = 禁止；1 = 使能
[13]	RTC1K_CLK_GTE	0x1	RW	RTC1K 时钟使能 0 = 禁止；1 = 使能
[12]	RTC32K_CLK_GTE	0x1	RW	RTC32K 时钟使能 0 = 禁止；1 = 使能
[11]	RTC32K_ISO_GTE	0x1	RW	RTC32K 隔离使能 0 = 禁止；1 = 使能
[10]	保留	0x0	RO	---
[9:0]	保留	0x3f7	RW	---

6.5.2.13 电源控制寄存器（PWRCR）

偏移地址：0x0030 ~ 0x0033 复位值：0x2600008F

31	30	29	28	27	26	25	24
VCC_CLR_LATCH	VCC_SET_LATCH	VCC_RE_LVDT5	VCC_RE_LVDT18	保留	VCARD0_ISOEN	保留	
rw	rw	rw	rw	ro	rw	ro	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
LVD5V_PD_CHIP	保留						
rw	ro						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VCC_IE_LVDT5	VCC_IE_LVDT18	保留	CARD0_IE_LVD	保留	CARD0_IE_EN_FAIL	保留	CARD0_RE_LVD
ro	rw	ro	rw	ro	rw	ro	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VCC_OE_LVDT5	VCC_OE_LVDT18	保留	CARD0_OE_LVD	VCC_EN_LVDT5	VCC_EN_LVDT18	保留	CARD0_LVD_EN
rw	rw	ro	rw	rw	rw	ro	rw

图表 6-14：电源控制寄存器（PWRCR）

比特位	名称	复位值	读写属性	功能说明
[31]	VCC_CLR_LATCH	0x0	RW	VCC 清除 IO 锁存 写 1 清除 IO 锁存
[30]	VCC_SET_LATCH	0x0	RW	VCC 设置 IO 锁存 写 1 设置 IO 锁存
[29]	VCC_RE_LVDT5	0x1	RW	VCC LVDT5V 复位使能 0 = 禁止；1 = 使能
[28]	VCC_RE_LVDT18	0x0	RW	VCC LVDT1.8V 复位使能 0 = 禁止；1 = 使能
[27]	保留	0x0	RO	---
[26]	VCARD0_ISOEN	0x1	RW	VCARD0 接口隔离使能 0 = 禁止；1 = 使能
[25:24]	保留	0x2	RO	---
[23]	LVD5V_PD_CHIP	0x0	RW	芯片 LVD5V 掉电 0 = 当 LVD5V 被 PMU 检测到，保持当前芯片状态 1 = 当 LVD5V 被 PMU 检测到，进入休眠模式
[22:16]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[15]	VCC_IE_LVDT5	0x0	RW	VCC LVDT5V 中断使能 0 = 禁止； 1 = 使能
[14]	VCC_IE_LVDT18	0x0	RW	VCC LVDT1.8V 中断使能 0 = 禁止； 1 = 使能
[13]	保留	0x0	RO	---
[12]	CARD0_IE_LVD	0x0	RW	LVD CARD0 中断使能 0 = 禁止； 1 = 使能
[11]	保留	0x0	RO	---
[10]	CARD0_IE_EN_FAIL	0x0	RW	CARD0 失败中断使能 0 = 禁止； 1 = 使能
[9]	保留	0x0	RO	---
[8]	CARD0_RE_LVD	0x0	RW	LVD CARD0 复位使能 0 = 禁止； 1 = 使能
[7]	VCC_OE_LVDT5	0x0	RW	VCC LVDT5V 输出使能 0 = 禁止； 1 = 使能
[6]	VCC_OE_LVDT18	0x0	RW	VCC LVDT1.8V 输出使能 0 = 禁止； 1 = 使能
[5]	保留	0x0	RO	---
[4]	CARD0_OE_LVD	0x0	RW	LVD CARD0 输出使能 0 = 禁止； 1 = 使能
[3]	VCC_EN_LVDT5	0x0	RW	VCC LVDT5V 使能 0 = 禁止； 1 = 使能
[2]	VCC_EN_LVDT18	0x0	RW	VCC LVDT1.8V 使能 0 = 禁止； 1 = 使能
[1]	保留	0x0	RO	---
[0]	CARD0_EN_LVD	0x0	RW	LVD CARD0 使能 0 = 禁止； 1 = 使能

6.5.2.14 睡眠计数寄存器（SLPCNTR）

偏移地址：0x0034~ 0x0037　　　　复位值：0xffffffff

31	30	29	28	27	26	25	24
SLPCNT0[7:0]							
rw							

23	22	21	20	19	18	17	16
SLPCNT1[7:0]							
rw							

15	14	13	12	11	10	9	8
SLPCNT2[7:0]							
rw							

7	6	5	4	3	2	1	0
SLPCNT3[7:0]							
rw							

图表 6-15：睡眠计数寄存器（SLPCNTR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	SLPCNT0[7:0]	0xFF	RW	睡眠计数 0 配置延迟时间当系统进入睡眠模式
[23:16]	SLPCNT1[7:0]	0xFF	RW	睡眠计数 1 配置延迟时间当系统进入睡眠模式
[15:8]	SLPCNT2[7:0]	0xFF	RW	睡眠计数 2 配置延迟时间当系统进入睡眠模式
[7:0]	SLPCNT3[7:0]	0xFF	RW	睡眠计数 3 配置延迟时间当系统进入睡眠模式

6.5.2.15 唤醒计数寄存器（WKPCNTR）

偏移地址：0x0038 ~ 0x003B 复位值：0xffffffff

31	30	29	28	27	26	25	24
WKPCNT0[7:0]							
rw							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WKPCNT1[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WKPCNT2[7:0]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WKPCNT3[7:0]							
rw							

图表 6-16：唤醒计数寄存器（WKPCNTR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	WKPCNT0[7:0]	0xff	RW	唤醒计数 0 配置延迟时间当系统从睡眠模式唤醒
[23:16]	WKPCNT1[7:0]	0xff	RW	睡眠计数 1 配置延迟时间当系统从睡眠模式唤醒
[15:8]	WKPCNT2[7:0]	0xff	RW	睡眠计数 2 配置延迟时间当系统从睡眠模式唤醒
[7:0]	WKPCNT3[7:0]	0xff	RW	睡眠计数 3 配置延迟时间当系统从睡眠模式唤醒

6.5.2.16 多时钟门控寄存器（MULTICGTCR）

多时钟门控寄存器，各个模块的时钟门控使能

0 = 模块时钟禁止

1 = 模块时钟使能

偏移地址：0x003C ~ 0x003F　　　　复位值：0xFFFFFFFF

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

图表 6-17：时钟门控寄存器（MULTICGTCR）

比特位	名称	复位值	读写属性	功能说明
[31:27]	保留	0x1F	RO	---
[26]	时钟使能	0x1	RW	TRACE
[25]	保留	0x1	RO	---
[24]	时钟使能	0x1	RW	EPORT4
[23]	时钟使能	0x1	RW	EPORT3
[22]	时钟使能	0x1	RW	EPORT2
[21]	时钟使能	0x1	RW	EPORT1
[20]	时钟使能	0x1	RW	EPORT
[19]	时钟使能	0x1	RW	CPM_IPS
[18]	时钟使能	0x1	RW	EFM_IPS
[17]	保留	0x1	RO	---
[16]	时钟使能	0x1	RW	KEY_CTRL
[15]	时钟使能	0x1	RW	CLKOUT
[14:11]	保留	0xF	RO	---
[10]	时钟使能	0x1	RW	TC
[9]	保留	0x1	RW	---
[8]	保留	0x1	RO	---

比特位	名称	复位值	读写属性	功能说明
[7]	时钟使能	0x1	RO	---
[6]	时钟使能	0x1	RO	---
[5]	时钟使能	0x1	RO	---
[4:2]	保留	0x7	RO	---
[1]	时钟使能	0x1	RW	EFM_BUS
[0]	保留	0x1	RO	---

注意：

如果寄存器 CLKOUT 比特位被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 13（OVERWR_ARITHCGT_TRIM）置 1。

6.5.2.17 系统时钟门控寄存器（SYSCGTCR）

系统时钟门控寄存器，各个模块的时钟门控使能

0 = 模块时钟禁止

1 = 模块时钟使能

偏移地址：0x0040 ~ 0x0043　　　　复位值：0xFFFFFFFF

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

图表 6-18：系统时钟门控寄存器（SYSCGTCR）

比特位	名称	复位值	读写属性	功能说明
[31:23]	保留	0x1FF	RO	---
[22]	时钟使能	0x1	RW	M2S_BUS_M
[21:20]	保留	0x3	RO	---
[19]	时钟使能	0x1	RW	ROM
[18]	时钟使能	0x1	RW	SSI5

比特位	名称	复位值	读写属性	功能说明
[17]	时钟使能	0x1	RW	SSI4
[16]	时钟使能	0x1	RW	SRAM3
[15]	时钟使能	0x1	RW	SRAM2
[14]	时钟使能	0x1	RW	SRAM1
[13]	时钟使能	0x1	RW	SRAM0
[12]	时钟使能	0x1	RW	SRAMD
[11]	时钟使能	0x1	RW	AHB2_MUX
[10:6]	保留	0x1F	RO	---
[5]	时钟使能	0x1	RW	CRC1
[4]	时钟使能	0x1	RW	CRC0
[3]	保留	0x1	RO	---
[2]	时钟使能	0x1	RW	DMAC2
[1]	时钟使能	0x1	RW	DMAC1
[0]	保留	0x1	RO	---

6.5.2.18 AHB3 时钟门控寄存器（AHB3CGTCR）

AHB3 时钟门控寄存器，各个模块的时钟门控使能

0 = 模块时钟禁止

1 = 模块时钟使能

偏移地址：0x0044 ~ 0x0047 复位值：0xFFFFFFFF

31	30	29	28	27	26	25	24
AHB3CLK_GTE[31:24]							
rw							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
AHB3CLK_GTE[23:16]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
AHB3CLK_GTE[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AHB3CLK_GTE[7:0]							
rw							

图表 6-19：AHB3 时钟门控寄存器（AHB3CGTCR）

比特位	名称	复位值	读写属性	功能说明
[31:6]	保留	0x3FFFFFF	RO	---
[5]	时钟使能	0x1	RW	AHB3_MUX
[4]	保留	0x1	RO	---
[3]	时钟使能	0x1	RW	USBC
[2:0]	保留	0x7	RO	---

6.5.2.19 算法时钟门控寄存器（ARITHCGTCR）

算法时钟门控寄存器，各个模块的时钟门控使能

0 = 模块时钟禁止

1 = 模块时钟使能

偏移地址：0x0048 ~ 0x004B 复位值：0xFFFFFFFF

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

图表 6-20：算法时钟门控寄存器（ARITHCGTCR）

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0xFFFFF	RO	---
[11]	时钟使能	0x1	RW	AHB2IPS2
[10]	时钟使能	0x1	RW	AHB2MLB
[9]	保留	0x1	RW	---
[8]	时钟使能	0x1	RW	DES
[7]	时钟使能	0x1	RW	EDMAC0
[6]	时钟使能	0x1	RW	SHA
[5]	保留	0x1	RW	---
[4]	时钟使能	0x1	RW	RF for AES
[3]	保留	0x1	RO	---
[2]	保留	0x1	RW	---
[1]	时钟使能	0x1	RW	AES
[0]	保留	0x1	RO	---

注意：

如果寄存器值被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 13（OVERWR_ARITHCGT_TRIM）置 1。

6.5.2.20 IPS 时钟门控寄存器（IPSCGTCR）

IPS 时钟门控寄存器，各个模块的时钟门控使能

0 = 模块时钟禁止

1 = 模块时钟使能

偏移地址：0x004C ~ 0x004F　　　　复位值：0xFFFFFFFF

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

图表 6-21：IPS 时钟门控寄存器（IPSCGTCR）

比特位	名称	复位值	读写属性	功能说明
[31]	时钟使能	0x1	RW	CCM&RESET
[30]	时钟使能	0x1	RW	AHB2IPS
[29]	时钟使能	0x1	RW	PMURTC
[28]	时钟使能	0x1	RW	Async Timer
[27]	时钟使能	0x1	RW	SEC_DET
[26]	保留	0x1	RO	---
[25]	时钟使能	0x1	RW	TRNG
[24]	保留	0x1	RO	---
[23]	时钟使能	0x1	RW	TSI
[22]	时钟使能	0x1	RO	---
[21]	时钟使能	0x1	RW	DAC
[20]	时钟使能	0x1	RW	QADC
[19]	保留	0x1	RO	---
[18]	时钟使能	0x1	RW	SCI3
[17]	时钟使能	0x1	RW	I2C3
[16]	时钟使能	0x1	RW	I2C2
[15]	时钟使能	0x1	RW	PWM0

比特位	名称	复位值	读写属性	功能说明
[14]	时钟使能	0x1	RW	I2C1
[13]	保留	0x1	RO	---
[12]	时钟使能	0x1	RW	USI2
[11]	时钟使能	0x1	RW	SCI2
[10]	时钟使能	0x1	RW	SCI1
[9]	时钟使能	0x1	RW	SPI3
[8]	时钟使能	0x1	RW	SPI2
[7]	时钟使能	0x1	RW	SPI1
[6]	时钟使能	0x1	RW	EDMAC1
[5]	时钟使能	0x1	RW	USI1
[4]	时钟使能	0x1	RW	PIT2
[3]	时钟使能	0x1	RW	PIT1
[2]	时钟使能	0x1	RW	RTC
[1]	时钟使能	0x1	RW	WDT
[0]	时钟使能	0x1	RW	IO_CTRL

6.5.2.21 VCC 通用校准寄存器（VCCGTRIMR）

通用校准寄存器，校准比特位针对于 PMC VCC

偏移地址：0x0050 ~ 0x0053 复位值：0x8E8E0331

31	30	29	28	27	26	25	24
V33_SW_ENB	DISCHARGE_EN	保留	TRIM_128KHZ[4:0]				
rw	rw	ro	rw				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
2KHZ_GTE	保留		TRIM_2KHZ[4:0]				
rw	ro		rw				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ST_1V	保留	VCC_LATCH_AUTO_SET	VCC_LATCH_AUTO_CLR	保留		SAMPLE_DIV[1:0]	
rw	ro	rw	rw	ro		rw	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEST_BIAS	BIAS1_RES_TRIM[2:0]			保留		BIAS2_RES_TRIM[1:0]	
rw	rw			ro		rw	

图表 6-22：VCC 通用校准寄存器（VCCGTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31]	V33_SW_ENB	0x1	RW	V33 切换使能 当 vdd5v<3.6V 时，本比特位清 0 会将 vd33/vd33_flash = vdd5v。
[30]	DISCHARGE_EN	0x0	RW	放电使能 本比特位被置 1 时，芯片切换到休眠模式（poff2）时 vd33 会放电
[29]	保留	0x0	RO	---
[28:24]	TRIM_128KHZ[4:0]	0x0	RW	用于校准 OSC128K
[23]	2KHZ_GTE	0x1	RW	2KHz 时钟门控使能 0 = 禁止；1 = 使能
[22:21]	保留	0x0	RO	---
[20:16]	TRIM_2KHZ[4:0]	0xE	RW	用于校准 OSC2K
[15]	ST_1V	0x0	RW	当这个比特置 1，芯片电压是 0.9V
[14]	保留	0x0	RO	---
[13]	VCC_LATCH_AUTO_SET	0x0	RW	芯片在进入睡眠模式时 IO 锁存会被自动置位。 写 1 使能 IO 锁存自动设置

比特位	名称	复位值	读写属性	功能说明
[12]	VCC_LATCH_AUTO_CLR	0x0	RW	芯片在退出睡眠模式时 IO 锁存会被自动清除。 写 1 使能 IO 锁存自动清除
[11:10]	保留	0x0	RO	---
[9:8]	SAMPLE_DIV[1:0]	0x3	RW	校准唤醒周期
[7]	TEST_BIAS	0	RW	测试偏执电流使能比特位
[6:4]	BIAS1_RES_TRIM[2:0]	0x3	RW	偏执 1 电流的校准位
[3:2]	保留	0x0	RO	---
[1:0]	BIAS2_RES_TRIM[1:0]	0x1	RW	芯片电压标定功能 00 = 1.05V；01 = 1.1V 10 = 1.15V；11 = 1.21V

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 23（OVERWR_VCC_TRIM）置 1。

6.5.2.22 VCC Lvdt 校准寄存器（VCCLTRIMR）

VCC Lvdt 校准寄存器，校准比特位针对于 PMC VCC。

偏移地址：0x0054 ~ 0x0057 复位值：0xC1001414

31	30	29	28	27	26	25	24
LVDT5V_TRIM_MSB[1:0]		保留					LVDT_COARSE_EN
rw		ro					rw

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留		LVDT5V_HYS_TRIM[1:0]		保留	LVDT5V_TRIM[2:0]		
ro		rw		ro	rw		

7	6	5	4	3	2	1	0
保留		LVDT18_HYS_TRIM[1:0]		保留	LVDT18_TRIM[2:0]		
ro		rw		ro	rw		

图表 6-23：VCC Lvdt 校准寄存器（VCCLTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31:30]	LVDT5V_TRIM_MSB[1:0]	0x3	RW	Lvdt5V 的校准位 00 = 4.3V；01 = 2.7V 10 = 2.0V；11 = 1.6V
[29:25]	保留	0x0	RO	---
[24]	LVDT_COARSE_EN	0x1	RW	Lvdt 粗调使能 0 = 禁止；1 = 使能
[23:14]	保留	0x0	RO	---
[13:12]	LVDT5V_HYS_TRIM[1:0]	0x1	RW	Lvdt5V 的迟滞校准位 20mV~40mV；10mV 一步进
[11]	保留	0x0	RO	---
[10:8]	LVDT5V_TRIM[2:0]	0x4	RW	Lvdt5V 校准位 （受 LVDT5V_TRIM_MSB[1:0]影响） 它的阈值参照表格 6-3。
[7:6]	保留	0x0	RO	---
[5:4]	LVDT18_HYS_TRIM[1:0]	0x1	RW	Lvdt18 的迟滞校准位 20mV~40mV；10mV 一步进
[3]	保留	0x0	RO	---
[2:0]	LVDT18_TRIM	0x4	RW	Lvdt18 校准位

比特位	名称	复位值	读写属性	功能说明
	[2:0]			它的阈值参照表格 6-4。

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 22（OVERWR_LVD_TRIM）置 1。

表格 6-3：lvdt5v 阈值（单位：V）

	LVDT5V_TRIM_ MSB = 2'b00	LVDT5V_TRIM_ MSB = 2'b01	LVDT5V_TRIM_ MSB = 2'b10	LVDT5V_TRIM_ MSB = 2'b11
LVDT5V_TRIM = 3'b000	3.22	2.34	1.84	1.38
LVDT5V_TRIM = 3'b001	3.30	2.43	1.87	1.41
LVDT5V_TRIM = 3'b010	3.39	2.52	1.89	1.45
LVDT5V_TRIM = 3'b011	3.48	2.57	1.92	1.48
LVDT5V_TRIM = 3'b100	3.58	2.63	1.95	1.51
LVDT5V_TRIM = 3'b101	3.68	2.68	2.04	1.53
LVDT5V_TRIM = 3'b110	4.02	2.74	2.15	1.55
LVDT5V_TRIM = 3'b111	4.15	2.80	2.26	1.57

表格 6-4：lvdt18 阈值（单位：V）

校准位	阈值
LVDT18_TRIM = 3'b000	1.34
LVDT18_TRIM = 3'b001	1.37
LVDT18_TRIM = 3'b010	1.40
LVDT18_TRIM = 3'b011	1.43
LVDT18_TRIM = 3'b100	1.46
LVDT18_TRIM = 3'b101	1.49
LVDT18_TRIM = 3'b110	1.52
LVDT18_TRIM = 3'b111	1.55

6.5.2.23 VCC 参考电压校准寄存器（VCCVTRIMR）

VCC 参考电压校准寄存器，校准比特位针对于 PMC VCC

偏移地址：0x0058 ~ 0x005B 复位值：0x00070008

31	30	29	28	27	26	25	24
SLP_PROTECT_EN	保留						
rw	ro						

23	22	21	20	19	18	17	16
保留					WK_EN[2:0]		
ro					rw		

15	14	13	12	11	10	9	8
保留		S2M_O_ISOEN_SRAM_MASK	S2M_O_ISOEN_VDD_MASK	VREF_FORCE_ST	VREF_TRIM_EN	VREF_TRIM_LOAD	VREF_STORE_EN
ro		rw	rw	rw	rw	rw	rw

7	6	5	4	3	2	1	0
保留				VREF_TRIM_VAL			
ro				rw			

图表 6-24：VCC 参考电压校准寄存器（VCCVTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31]	SLP_PROTECT_EN	0x0	RW	睡眠配置寄存器保护使能位 0 = 禁止；1 = 使能
[30:19]	保留	0x0	RO	---
[18:16]	WK_EN[2:0]	0x7	RW	poff2 模式下分别对应于 PAD POR，WAKE 和 USBDET 位的唤醒使能，需要将 VREF_TRIM_LOAD 写 1 来确认这些比特位的改变。
[15:14]	保留	0x0	RO	---
[13]	保留	0x0	RO	---
[12]	保留	0x0	RO	---
[11]	VREF_FORCE_ST	0x0	RW	当遇到 pesd 时让 vref 保持稳定
[10]	VREF_TRIM_EN	0x0	RW	用于校准 vref 电压 0 = 禁止；1 = 使能
[9]	VREF_TRIM_LOAD	0x0	RW	用于载入 vref 校准值和唤醒使能比特
[8]	VREF_STORE_EN	0x0	RW	存储 vref 电压使能 0 = 禁止；1 = 使能

比特位	名称	复位值	读写属性	功能说明
[7:4]	保留	0x0	RO	---
[3:0]	VREF_TRIM_VAL	0x8	RW	vref 的校准值。改变该校准值后需要将 VREF_TRM_LOAD 置 1 才能使其生效。

注意：

1. 如果寄存器 SLP_PROTECT_EN 被 flash info load 值改变，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 21（OVERWR_VREF_TRIM）置 1。

2. 如果用户想要重写 SLP_PROTECT_EN / WK_EN/VREF_TRIM_EN / VREF_STORE_EN / VREF_TRIM_VAL，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 21（OVERWR_VREF_TRIM）置 1。

6.5.2.24 VCC 核测试模式寄存器（VCCCTMR）

偏移地址：0x005C ~ 0x005F　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
CORE_TEST_KEY[1:0]		OVERWR_CSWCFG_TRIM	OVERWR_RTC_TRIM	保留	OVERWR_RTC_STABLE_TRIM	保留	OVERWR_CARDLDO_TRIM
rw		rw	rw	ro	rw	ro	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OVERWR_VCC_TRIM	OVERWR_LVD_TRIM	OVERWR_VREF_TRIM	OVERWR_OSC8M_TRIM	OVERWR_OSC120M_TRIM	OVERWR_OSCL_STABLE_TRIM	OVERWR_OSCH_STABLE_TRIM	OVERWR_OSCE_STABLE_TRIM
rw	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留		OVERWR_ARITHCGT_TRIM	保留	OVERWR_SCDIV_TRIM	OVERWR_PCDIV_TRIM	OVERWR_OCSR_TRIM	OFF_MODE_WK
ro		rw	ro	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEST_MODE_EN	保留			SOFT_POR	OFF_MODE2	OFF_MODE	EN_LP
rw	ro			rw	rw	rw	rw

图表 6-25：VCC 核测试模式寄存器（VCCCTMR）

比特位	名称	复位值	读写属性	功能说明
[31:30]	CORE_TEST_KEY[1:0]	0x0	RW	核测试模式钥匙 VCCQCTMR 寄存器的置不能被直接改写，必须要按照正确的顺序写这两个比特位。正确的顺序

比特位	名称	复位值	读写属性	功能说明
				是：2’b01->2’b10->2’b11。写完这些后，这两个比特位的置是 2’b11，这个时候 VCCQCTMR 寄存器其他比特位的值可以被改写。当这两个比特位的值为 2’b11 时，只有写 2’b00 才能清除这两个比特位，写其他值对这两个比特位没有影响且返回值为 2’b11。
[29]	OVERWR_CSWCFG_TRIM	0x0	RW	使能 CPU 重写 CSWCFGR 的校准值。**注意：**只支持将 SYSCLK_SEL[1:0]校准到 2’b00 0 = 禁止；1 = 使能
[28]	OVERWR_RTC_TRIM	0x0	RW	使能 CPU 重写 RTCTRIMR 的校准值。 0 = 禁止；1 = 使能
[27]	保留	0x0	RO	---
[26]	OVERWR_RTC_STABLE_TRIM	0x0	RW	使能 CPU 重写 RTCSTIMER 的校准值。 0 = 禁止；1 = 使能
[25]	保留	0x0	RO	---
[24]	OVERWR_CARDLDO_TRIM	0x0	RW	使能 CPU 重写 CARD0/1 LDO 和模拟唤醒滤波的校准值。 0 = 禁止；1 = 使能
[23]	OVERWR_VCC_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 VCC 校准值。 0 = 禁止；1 = 使能
[22]	OVERWR_LVD_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 LVD 校准值。 0 = 禁止；1 = 使能
[21]	OVERWR_VREF_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 VREF 校准值。 0 = 禁止；1 = 使能
[20]	OVERWR_OSC8M_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 OSC8M 校准值。 0 = 禁止；1 = 使能
[19]	OVERWR_OSC120M_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 OSC120M 校准值。 0 = 禁止；1 = 使能
[18]	OVERWR_OSCL_STABLE_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 OSC8M 稳定时间校准值。 0 = 禁止；1 = 使能
[17]	OVERWR_OSCH_STABLE_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的 OSC120M 稳定时间校准值。 0 = 禁止；1 = 使能
[16]	OVERWR_OSCE_STABLE_TRIM	0x0	RW	使能 CPU 重写来自 FLASH 的外部 OSC 稳定时间校准值。 0 = 禁止；1 = 使能

比特位	名称	复位值	读写属性	功能说明
[15:14]	保留	0x0	RO	---
[13]	OVERWR_ARITHCGT_TRIM	0x0	RW	使能 CPU 重写 ARITHCGTR 的校准值。 0 = 禁止；1 = 使能
[12]	保留	0x0	RO	---
[11]	OVERWR_SCDIV_TRIM	0x0	RW	使能 CPU 重写 SCDIVR 的校准值。 0 = 禁止；1 = 使能
[10]	OVERWR_PCDIV_TRIM	0x0	RW	使能 CPU 重写 PCDIVR 的校准值。 0 = 禁止；1 = 使能
[9]	OVERWR_OCSR_TRIM	0x0	RW	使能 CPU 重写 OCSR 的校准值。 0 = 禁止；1 = 使能
[8]	OFF_MODE_WK	0x0	RW	在核测试模式下是 PMC VCC 的直接控制比特位，只用于测试，用户不应配置。
[7]	TEST_MODE_EN	0x0	RW	使能 PMC VCC 的核测试模式 0 = 禁止；1 = 使能
[6:4]	保留	0x0	RO	---
[3]	SOFT_POR	0x0	RW	设置这个比特来产生一个软件复位事件到主系统 0 = 没有软件复位；1 = 软件复位
[2]	OFF_MODE2	0x0	RW	同 bit8
[1]	OFF_MODE	0x0	RW	同 bit8
[0]	EN_LP	0x0	RW	同 bit8

6.5.2.25 OSC8M 校准寄存器（O8MTRIMR）

偏移地址：0x0060 ~ 0x0063　　　　复位值：0x00018779

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							OSC8M_TRIM[16]
ro							rw

15	14	13	12	11	10	9	8
OSC8M_TRIM[15:8]							
rw							

7	6	5	4	3	2	1	0
OSC8M_TRIM[7:0]							
rw							

图表 6-26：OSC8M 校准寄存器（O8MTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31:17]	保留	0x0	RO	---
[16:0]	OSC8M_TRIM [16:0]	0x18779	RW	OSC8MHz 校准值

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特位 20（OVERWR_OSC8M_TRIM）置 1。

6.5.2.26 OSC120M 校准寄存器（O120MTRIMR）

偏移地址：0x0068 ~ 0x006B 复位值：0x0001877A

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							OSC120M_TRIM[16]
ro							rw

15	14	13	12	11	10	9	8
OSC120M_TRIM[15:8]							
rw							

7	6	5	4	3	2	1	0
OSC120M_TRIM[7:0]							
rw							

图表 6-27：OSC120M 校准寄存器（O120MTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31:17]	保留	0x0	RO	---
[16:0]	OSC120M_TRIM[16:0]	0x1877A	RW	OSC120MHz 校准值

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特位 19（OVERWR_OSC120M_TRIM）置 1。

6.5.2.27 CARDLDO 校准寄存器（CARDTRIMR）

偏移地址：0x006C ~ 0x006F 复位值：0x10700688

31	30	29	28	27	26	25	24
保留	WKP_DFILT_EN	WKP_DFILT_BYPASS	WKP_DFILT_GTE	保留			
ro	rw	rw	rw	ro			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
EFM_RWSC_INIT[3:0]				保留			
rw				ro			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WKP_AFILT_BYPASS	保留		CARD0_REDUCE	WKP_AFILT_TRIM[3:0]			
rw	ro		rw	rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留							
ro							

图表 6-28：CARDLDO 校准寄存器（CARDTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31]	保留	0x0	RO	---
[30]	WKP_DFILT_EN	0x0	RW	唤醒数字过滤器使能
[29]	WKP_DFILT_BYPASS	0x0	RW	唤醒数字过滤器旁路
[28]	WKP_DFILT_GTE	0x1	RW	唤醒数字过滤器门控使能
[27:24]	保留	0x0	RO	---
[23:20]	EFM_RWSC_INIT[3:0]	0x7	RW	EFM RWSC 初始校准值
[19:16]	保留	0x0	RO	---
[15]	WKP_AFILT_BYPASS	0x0	RW	唤醒模拟过滤器旁路
[14:13]	保留	0x0	RO	---
[12]	CARD0_REDUCE	0x0	RW	当为 1 时会减少电流传感器的 10%电阻
[11:8]	WKP_AFILT_TRIM[3:0]	0x6	RW	唤醒模拟过滤器校准值
[7:0]	保留	0x88	RO	---

注意：

如果用户想要重写 EFM_RWSC_INIT / WKP_AFILT_BYPASS / CARD0_REDUCE / WKP_AFILT_TRIM，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特位 24（OVERWR_CARDLDO_TRIM）置 1。

6.5.2.28 OSCL 稳定时间寄存器（OSCLSTIMER）

偏移地址：0x0070 ~ 0x0073 复位值：0x000301FF

31	30	29	28	27	26	25	24
保留				PMU128K_STIME[11:8]			
ro				rw			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PMU 128K_STIME[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				OSC8M_STIME[11:8]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OSC8M_STIME[7:0]							
rw							

图表 6-29：OSCL 稳定时间寄存器（OSCLSTIMER）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	PMU128K_STIME[11:0]	0x3	RW	时钟源稳定时间（由 PMU128KHz 计数） 指的是 OCSR 内时钟源的使能比特位被置 1 后需要的稳定时间。
[15:12]	保留	0x0	RO	---
[11:0]	OSC8M_STIME[11:0]	0x1FF	RW	时钟源稳定时间（由 OSC8MHz 计数） 指的是 OCSR 内时钟源的使能比特位被置 1 后需要的稳定时间

注意：

如果用户想要重写 OSC8M_STIME 值，用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特位 18（OVERWR_OSCL_STABLE_TRIM）置 1。

6.5.2.29 OSCH 稳定时间寄存器（OSCHSTIMER）

偏移地址：0x0074 ~ 0x0077　　　　复位值：0x003f01ff

31	30	29	28	27	26	25	24
保留				OSC120M_STIME[11:8]			
ro				rw			

23	22	21	20	19	18	17	16
OSC120M_STIME[7:0]							
rw							

15	14	13	12	11	10	9	8
保留				USBPHY240M_STIME[11:8]			
ro				rw			

7	6	5	4	3	2	1	0
USBPHY240M_STIME[7:0]							
rw							

图表 6-30：OSCH 稳定时间寄存器（OSCHSTIMER）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	OSC120M_STIME[11:0]	0x3F	RW	时钟源稳定时间（由 OSC120MHz 计数） 指的是 OCSR 内时钟源的使能比特位被置 1 后需要的稳定时间。
[15:12]	保留	0x0	RO	---
[11:0]	USBPHY240M_STIME[11:0]	0x1FF	RW	时钟源稳定时间（由 USBPHY240MHz 计数） 指的是 OCSR 内时钟源的使能比特位被置 1 后需要的稳定时间

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 17（OVERWR_OSCH_STABLE_TRIM）置 1。

6.5.2.30 OSCE 稳定时间寄存器（OSCESTIMER）

偏移地址：0x0078 ~ 0x007B 复位值：0x000FFFFF

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留				OSCEXT_STIME[19:16]			
ro				rw			

15	14	13	12	11	10	9	8
OSCEXT_STIME[15:8]							
rw							

7	6	5	4	3	2	1	0
OSCEXT_STIME[7:0]							
rw							

图表 6-31：OSCE 稳定时间寄存器（OSCESTIMER）

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:0]	OSCEXT_STIME[19:0]	0xFFFFF	RW	时钟源稳定时间（由 OSC8MHz 计数） 指的是 OCSR 内时钟源的使能比特位被置 1 后需要的稳定时间。

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 16（OVERWR_OSCE_STABLE_TRIM）置 1。

6.5.2.31 电源状态寄存器（PWRSR）

偏移地址：0x007C ~ 0x007F　　　　复位值：0x0600000D

31	30	29	28	27	26	25	24
保留						VCARD0_ISOEN_O	保留
ro						ro	ro

23	22	21	20	19	18	17	16
VCC_LVDT5_F	VCC_LVDT18_F	保留	CARD0_LVD_F	VCC_LVDT5_RT	VCC_LVDT18_RT	保留	CARD0_LVD_RT
r/w1c	r/w1c	ro	r/w1c	ro	ro	ro	ro

15	14	13	12	11	10	9	8
保留	CARD0_EN_FAIL_F	保留	保留				保留
ro	r/w1c	ro	ro				ro

7	6	5	4	3	2	1	0
保留			保留	VCC_HP_READY	保留	VCC_CARD0_READY	保留
ro			ro	ro	ro	ro	ro

图表 6-32：电源状态寄存器（PWRSR）

比特位	名称	复位值	读写属性	功能说明
[31:26]	保留	0x1	RO	---
[25]	VCARD0_ISOEN_O	0x1	RO	VCARD0 输出隔离使能状态 0 = 已禁止；1 = 已使能
[24]	保留	0x0	RO	---
[23]	VCC_LVDT5V_F	0x0	R/W1C	lvdt5v 标志，检测主电源 VDD5V 0 = LVDT5V 事件未发生 1 = LVDT5V 事件发生
[22]	VCC_LVDT18_F	0x0	R/W1C	lvdt18 标志，检测主电源 VDD1P8V 0 = LVDT18 事件未发生 1 = LVDT18 事件发生
[21]	保留	0x0	RO	---
[20]	CARD0_LVD_F	0x0	R/W1C	CARD0_LVD 标志，检测 CARD0 LVD 0 = CARD0_LVD 事件未发生 1 = CARD0_LVD 事件发生
[19]	VCC_LVDT5V_RT	0x0	RO	lvdt5v 事件检测的实时状态， 检测主电源 VDD5V 0 = LVDT5V 事件没有正在发

比特位	名称	复位值	读写属性	功能说明
				1 = LVDT5V 事件正在发生
[18]	VCC_LVDT18_RT	0x0	RO	lvdt18 事件检测的实时状态， 检测 flash 电源 VDD1P8V 0 = LVDT18 事件没有正在发生 1 = LVDT18 事件正在发生
[17]	保留	0x0	RO	---
[16]	CARD0_LVD_RT	0x0	RO	CARD0_LVD 事件检测的实时状态， 检测 CARD0 LVD 0 = CARD0_LVD 事件没有正在发生 1 = CARD0_LVD 事件正在发生
[15]	保留	0x0	RO	---
[14]	CARD0_EN_FAIL_F	0x0	R/W1C	CARD0_LVD 失败标志 0 = CARD0_LVD 失败事件未发生 1 = CARD0_LVD 失败事件发生
[13]	保留	0x0	RO	---
[12:9]	保留	0x0	RO	---
[8]	保留	0x0	RO	---
[7:5]	保留	0x0	RO	---
[4]	保留	0x0	RO	---
[3]	VCC_HP_READY	0x1	RO	仅用于 PMC VCC 的测试
[2]	保留	0x0	RO	---
[1]	VCC_CARD0_READY	0x0	RO	仅用于 PMC VCC 的测试
[0]	保留	0x0	RO	---

6.5.2.32 RTC 校准寄存器（RTCTRIMR）

偏移地址：0x008C ~ 0x008F　　复位值：0x00000000

31	30	29	28	27	26	25	24
RTC_TRIM_EN	RTC_TRIM_LOAD	保留					
rw	rw	ro					

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
RTC_TRIM_VAL[7:0]							
rw							

图表 6-33：RTC 校准寄存器（RTCTRIMR）

比特位	名称	复位值	读写属性	功能说明
[31]	RTC_TRIM_EN	0x0	RW	RTC 校准使能 0 = 禁止；1 = 使能
[30]	RTC_TRIM_LOAD	0x0	RW	RTC 校准加载使能 0 = 禁止；1 = 使能
[29:8]	保留	0x0	RO	---
[7:0]	RTC_TRIM_VAL[7:0]	0x0	RW	RTC 校准值

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 28（OVERWR_RTC_TRIM）置 1。

6.5.2.33 管脚唤醒中断控制寄存器（PADWKPINTCR）

偏移地址：0x0090 ~ 0x0093　　　　复位值：0x00FF0000

31	30	29	28	27	26	25	24
S2M_INT_RT	保留		PAD_SS3_RT	ATIMER_RT	保留	WAKE_RT	USBDET_RT
ro	ro		ro	ro	ro	ro	ro
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
S2M_INT_EN	保留		PAD_SS3_EN	ATIMER_EN	保留	WAKE_EN	USBDET_EN
rw	ro		rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
S2M_INT_INTM	保留		PAD_SS3_INTM	ATIMER_INTM	保留	WAKE_INTM	USBDET_INTM
rw	ro		rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
S2M_INT_STAT	保留		PAD_SS3_STAT	ATIMER_STAT	保留	WAKE_STAT	USBDET_STAT
ro	ro		ro	ro	ro	ro	ro

图表 6-34：管脚唤醒中断控制寄存器（PADWKPINTCR）

比特位	名称	复位值	读写属性	功能说明
[31]	保留	0x0	RO	---
[30:29]	保留	0x0	RO	---
[28]	PAD_SS3_RT	0x0	RO	PAD_SS3 唤醒源实时值 0 = 唤醒源无效；1 = 唤醒源有效
[27]	ATIMER_RT	0x0	RO	Async TIMER 唤醒源实时值 0 = 唤醒源无效；1 = 唤醒源有效
[26]	保留	0x0	RO	---
[25]	WAKE_RT	0x0	RO	WAKE 唤醒源实时值 0 = 唤醒源无效；1 = 唤醒源有效
[24]	USBDET_RT	0x0	RO	USBDET 唤醒源实时值 0 = 唤醒源无效；1 = 唤醒源有效
[23]	保留	0x0	RO	---
[22:21]	保留	0x3	RO	---
[20]	PAD_SS3_EN	0x1	RW	PAD_SS3 唤醒源使能 0 = 禁止；1 = 使能
[19]	ATIMER_EN	0x1	RW	Async TIMER 唤醒源使能 0 = 禁止；1 = 使能
[18]	保留	0x1	RO	---

比特位	名称	复位值	读写属性	功能说明
[17]	WAKE_EN	0x1	RW	WAKE 唤醒源使能 0 = 禁止；1 = 使能
[16]	USBDET_EN	0x1	RW	USBDET 唤醒源使能 0 = 禁止；1 = 使能
[15]	保留	0x0	RO	---
[14:13]	保留	0x0	RO	---
[12]	PAD_SS3_INTM	0x0	RW	PAD_SS3 唤醒源中断使能作为唤醒中断请求 0 = 禁止；1 = 使能
[11]	ATIMER_INTM	0x0	RW	Async TIMER 唤醒源中断使能作为唤醒中断请求 0 = 禁止；1 = 使能
[10]	保留	0x0	RO	---
[9]	WAKE_INTM	0x0	RW	WAKE 唤醒源中断使能作为唤醒中断请求 0 = 禁止；1 = 使能
[8]	USBDET_INTM	0x0	RW	USBDET 唤醒源中断使能作为唤醒中断请求 0 = 禁止；1 = 使能
[7]	保留	0x0	RO	---
[6:5]	保留	0x0	RO	---
[4]	PAD_SS3_STAT	0x0	RO	PAD_SS3 唤醒源中断状态 0 = 唤醒源中断无效；1 = 唤醒源中断有效
[3]	ATIMER_STAT	0x0	RO	Async TIMER 唤醒源中断状态 0 = 唤醒源中断无效；1 = 唤醒源中断有效
[2]	保留	0x0	RO	---
[1]	WAKE_STAT	0x0	RO	WAKE 唤醒源中断状态 0 = 唤醒源中断无效；1 = 唤醒源中断有效
[0]	USBDET_STAT	0x0	RO	USBDET 唤醒源中断状态 0 = 唤醒源中断无效；1 = 唤醒源中断有效

6.5.2.34 唤醒滤波计数寄存器（WKPFILTCNTR）

偏移地址：0x0094 ~ 0x0097　　复位值：0xFF0000F0

31	30	29	28	27	26	25	24
WAKEUP_FILTCNT0[7:0]							
rw							

23	22	21	20	19	18	17	16
WAKEUP_FILTCNT1[7:0]							
rw							

15	14	13	12	11	10	9	8
WAKEUP_FILTCNT2[7:0]							
rw							

7	6	5	4	3	2	1	0
WAKEUP_FILTCNT3[7:0]							
rw							

图表 6-35：唤醒滤波计数寄存器（WKPFILTCNTR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	WAKEUP_FILTCNT0[7:0]	0xFF	RW	唤醒滤波计数 0 配置唤醒滤波计数值
[23:16]	WAKEUP_FILTCNT1[7:0]	0x0	RW	唤醒滤波计数 1 配置唤醒滤波计数值
[15:8]	WAKEUP_FILTCNT2[7:0]	0x0	RW	唤醒滤波计数 2 配置唤醒滤波计数值
[7:0]	WAKEUP_FILTCNT3[7:0]	0xF0	RW	唤醒滤波计数 3 配置唤醒滤波计数值

6.5.2.35 CARD 上电计数寄存器（CARDPOCR）

偏移地址：0x0098 ~ 0x009B　　复位值：0x000C0000

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

图表 6-36：CARD 上电计数寄存器（CARDPOCR）

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:0]	POWAITCNT[19:0]	0xC0000	RW	CARD 上电等待计数 配置 CARD 上电等待一个滤波计数值

6.5.2.36 RTC32K 稳定时间寄存器（RTCSTIMER）

偏移地址：0x009C ~ 0x009F 复位值：0x00000190

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留				RTCSTIME[19:16]			
ro				rw			

15	14	13	12	11	10	9	8
RTCSTIME[15:8]							
rw							

7	6	5	4	3	2	1	0
RTCSTIME[7:0]							
rw							

图表 6-37：RTC32K 稳定时间寄存器（RTCSTIMER）

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:0]	RTCSTIME[19:0]	0x190	RW	RTC32K 参考时钟稳定等待计数

注意：

用户需要在 CPU 重写旧值之前将 VCCCTMR 寄存器的比特 26（OVERWR_RTC_STABLE_TRIM）置 1。

6.5.2.37 存储器掉电睡眠控制寄存器（MPDSLPCR）

存储器掉电睡眠控制寄存器，配置各个存储器在睡眠模式下是否掉电。LS 代表浅睡眠，DS 代表深度睡眠，SD 代表关闭。在 LS 和 DS 模式下，可以保留存储器中的内容；而在 SD 模式下，内容丢失。

0 = 睡眠模式下存储器不掉电

1 = 睡眠模式下存储器掉电

偏移地址：0x00A0 ~ 0x00A3 复位值：0x00000000

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

图表 6-38：存储器掉电睡眠控制寄存器（MPDSLPCR）

比特位	名称	复位值	读写属性	功能说明
[31:29]	保留	0x0	RO	---
[28]	对应的存储器	0x0	RW	USBC_SRAM_DS
[27]	保留	0x0	RO	---
[26]	保留	0x0	RO	---
[25]	对应的存储器	0x0	RW	ARITH_RF_SD
[24:22]	保留	0x0	RO	---
[21]	保留	0x0	RO	---
[20]	保留	0x0	RO	---
[19]	对应的存储器	0x0	RW	ARITH_RF_DS
[18]	对应的存储器	0x0	RW	ARITH_RF_LS
[17:16]	保留	0x0	RO	---
[15]	对应的存储器	0x0	RW	CODE_ROM_LS
[14:13]	保留	0x0	RO	---
[12]	对应的存储器	0x0	RW	SRAMD_DS
[11]	对应的存储器	0x0	RW	SRAM3_DS

比特位	名称	复位值	读写属性	功能说明
[10]	对应的存储器	0x0	RW	SRAM2_DS
[9]	对应的存储器	0x0	RW	SRAM1_DS
[8]	对应的存储器	0x0	RW	SRAM0_DS
[7:5]	保留	0x0	RO	---
[4]	对应的存储器	0x0	RW	DCache_DS
[3:1]	保留	0x0	RO	---
[0]	对应的存储器	0x0	RW	ICache_DS

6.5.2.38 Multiple 复位控制寄存器（MULTIRSTCR）

Multi 复位屏蔽控制寄存器：配置各个模块的复位屏蔽值。

0 = 模块复位不声明

1 = 模块复位声明

偏移地址：0x00AC ~ 0x00AF　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
MULTIRST_MSK[31:24]							
rw							

23	22	21	20	19	18	17	16
MULTIRST_MSK[23:16]							
rw							

15	14	13	12	11	10	9	8
MULTIRST_MSK[15:8]							
rw							

7	6	5	4	3	2	1	0
MULTIRST_MSK[7:0]							
rw							

图表 6-39：Multiple 复位控制寄存器（MULTIRSTCR）

比特位	名称	复位值	读写属性	功能说明
[31:25]	保留	0x0	RO	---
[24]	复位屏蔽值	0x0	RW	EPORT4
[23]	复位屏蔽值	0x0	RW	EPORT3
[22]	复位屏蔽值	0x0	RW	EPORT2
[21]	复位屏蔽值	0x0	RW	EPORT1
[20]	复位屏蔽值	0x0	RW	EPORT
[19]	复位屏蔽值	0x0	RW	CPM_IPS

比特位	名称	复位值	读写属性	功能说明
[18]	复位屏蔽值	0x0	RW	EFM_IPS
[17]	复位屏蔽值	0x0	RW	CCM&RESET
[16]	复位屏蔽值	0x0	RW	KEY_CTRL
[15:7]	保留	0x0	RO	---
[6]	保留	0x0	RW	---
[5]	保留	0x0	RW	---
[4]	复位屏蔽值	0x0	RW	M2S_BUS_S
[3:2]	保留	0x0	RO	---
[1]	复位屏蔽值	0x0	RW	EFM_BUS
[0]	保留	0x0	RO	---

6.5.2.39 系统复位控制寄存器（SYSRSTCR）

系统复位控制寄存器：配置各个模块的复位屏蔽值。

0 = 模块复位不声明

1 = 模块复位声明

偏移地址：0x00B0 ~ 0x00B3 复位值：0x00000000

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

图表 6-40：系统复位控制寄存器（SYSRSTCR）

比特位	名称	复位值	读写属性	功能说明
[31:23]	保留	0x0	RO	---
[22]	复位屏蔽值	0x0	RW	M2S_BUS_M
[21:20]	保留	0x0	RO	---
[19]	复位屏蔽值	0x0	RW	ROM

比特位	名称	复位值	读写属性	功能说明
[18]	复位屏蔽值	0x0	RW	SSI5
[17]	复位屏蔽值	0x0	RW	SSI4
[16]	复位屏蔽值	0x0	RW	SRAM3
[15]	复位屏蔽值	0x0	RW	SRAM2
[14]	复位屏蔽值	0x0	RW	SRAM1
[13]	复位屏蔽值	0x0	RW	SRAM0
[12]	复位屏蔽值	0x0	RW	SRAMD
[11]	复位屏蔽值	0x0	RW	AHB2_MUX
[10:6]	保留	0x0	RO	---
[5]	复位屏蔽值	0x0	RW	CRC1
[4]	复位屏蔽值	0x0	RW	CRC0
[3]	保留	0x0	RO	---
[2]	复位屏蔽值	0x0	RW	DMAC2
[1]	复位屏蔽值	0x0	RW	DMAC1
[0]	保留	0x0	RO	---

6.5.2.40 AHB3 复位控制寄存器（AHB3RSTCR）

AHB3 复位控制寄存器：配置各个模块的复位屏蔽值。

0 = 模块复位不声明

1 = 模块复位声明

偏移地址：0x00B4 ~ 0x00B7 复位值：0x00000000

31	30	29	28	27	26	25	24
AHB3RST_MSK[31:24]							
rw							
23	22	21	20	19	18	17	16
AHB3RST_MSK[23:16]							
rw							
15	14	13	12	11	10	9	8
AHB3RST_MSK[15:8]							
rw							
7	6	5	4	3	2	1	0
AHB3RST_MSK[7:0]							
rw							

图表 6-41：AHB3 复位控制寄存器（AHB3RSTCR）

比特位	名称	复位值	读写属性	功能说明
[31:6]	保留	0x0	RO	---
[5]	复位屏蔽值	0x0	RW	AHB3_MUX
[4]	保留	0x0	RO	---
[3]	复位屏蔽值	0x0	RW	USBC
[2:0]	保留	0x0	RO	---

6.5.2.41 算法复位控制寄存器（ARITHRSTCR）

算法复位控制寄存器：配置各个模块的复位屏蔽值。

0 = 模块复位不声明

1 = 模块复位声明

偏移地址：0x00B8 ~ 0x00BB　　复位值：0x00000000

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

图表 6-42：算法复位控制寄存器（ARITHRSTCR）

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11]	复位屏蔽值	0x0	RW	AHB2IPS2
[10]	复位屏蔽值	0x0	RW	AHB2MLB
[9]	保留	0x0	RW	---
[8]	复位屏蔽值	0x0	RW	DES
[7]	复位屏蔽值	0x0	RW	EDMAC0
[6]	复位屏蔽值	0x0	RW	SHA
[5]	保留	0x0	RW	---
[4]	复位屏蔽值	0x0	RW	RF for AES
[3]	保留	0x0	RO	---
[2]	保留	0x0	RW	---
[1]	复位屏蔽值	0x0	RW	AES
[0]	保留	0x0	RO	---

6.5.2.42 IPS 复位控制寄存器（IPSRSTCR）

IPS 复位控制寄存器：配置各个模块的复位屏蔽值。

0 = 模块复位不声明

1 = 模块复位声明

偏移地址：0x00BC ~ 0x00BF　　　　复位值：0x00000000

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

图表 6-43：IPS 复位控制寄存器（IPSRSTCR）

比特位	名称	复位值	读写属性	功能说明
[31]	保留	0x0	RO	---
[30]	复位屏蔽值	0x0	RW	AHB2IPS
[29]	复位屏蔽值	0x0	RW	PMURTC
[28]	复位屏蔽值	0x0	RW	Async Timer
[27]	复位屏蔽值	0x0	RW	SEC_DET
[26]	保留	0x0	RO	---
[25]	复位屏蔽值	0x0	RW	TRNG
[24]	保留	0x0	RO	---
[23]	复位屏蔽值	0x0	RW	TSI
[22]	保留	0x0	RW	---
[21]	复位屏蔽值	0x0	RW	DAC
[20]	复位屏蔽值	0x0	RW	QADC
[19]	保留	0x0	RO	---
[18]	复位屏蔽值	0x0	RW	SCI3
[17]	复位屏蔽值	0x0	RW	I2C3
[16]	复位屏蔽值	0x0	RW	I2C2

比特位	名称	复位值	读写属性	功能说明
[15]	复位屏蔽值	0x0	RW	PWM0
[14]	复位屏蔽值	0x0	RW	I2C1
[13]	保留	0x0	RO	---
[12]	复位屏蔽值	0x0	RW	USI2
[11]	复位屏蔽值	0x0	RW	SCI2
[10]	复位屏蔽值	0x0	RW	SCI1
[9]	复位屏蔽值	0x0	RW	SPI3
[8]	复位屏蔽值	0x0	RW	SPI2
[7]	复位屏蔽值	0x0	RW	SPI1
[6]	复位屏蔽值	0x0	RW	EDMAC1
[5]	复位屏蔽值	0x0	RW	USI1
[4]	复位屏蔽值	0x0	RW	PIT2
[3]	复位屏蔽值	0x0	RW	PIT1
[2]	复位屏蔽值	0x0	RW	RTC
[1]	复位屏蔽值	0x0	RW	WDT
[0]	复位屏蔽值	0x0	RW	IO_CTRL

6.5.2.43 睡眠控制寄存器 2（SLPCFGR2）

偏移地址：0x00C0 ~ 0x00C3　　　　复位值：0x0000F70A

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">保留</td></tr>
<tr><td colspan="8">ro</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>S2M_INTM_SGL</td><td colspan="2">保留</td><td>PAD_SS3_INTM_SGL</td><td>ATIMER_INTM_SGL</td><td>保留</td><td>WAKE_INTM_SGL</td><td>USBDET_INTM_SGL</td></tr>
<tr><td>rw</td><td colspan="2">ro</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">保留</td><td>CORE_F_SLPEN</td><td>CLKOUT_SLPEN</td><td>CPM_IPS_SLPEN</td><td>TC_SLPEN</td></tr>
<tr><td colspan="4">ro</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">保留</td><td>VDD_WK_SWOFF</td><td colspan="4">保留</td><td>VDD_PD_RET</td></tr>
<tr><td colspan="2">ro</td><td>rw</td><td colspan="4">ro</td><td>rw</td></tr>
</table>

图表 6-44：睡眠控制寄存器 2（SLPCFGR2）

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23]	保留	0x0	RO	---
[22:21]	保留	0x0	RO	---
[20]	PAD_SS3_INTM_SGL	0x0	RW	PAD_SS3 唤醒源中断使能作为每个信号的中断请求 0 = 禁止; 1 = 使能
[19]	ATIMER_INTM_SGL	0x0	RW	Async TIMER 唤醒源中断使能作为每个信号的中断请求 0 = 禁止; 1 = 使能
[18]	保留	0x0	RW	---
[17]	WAKE_INTM_SGL	0x0	RW	WAKE 唤醒源中断使能作为每个信号的中断请求 0 = 禁止; 1 = 使能
[16]	USBDET_INTM_SGL	0x0	RW	USBDET 唤醒源中断使能作为每个信号的中断请求 0 = 禁止; 1 = 使能
[15:12]	保留	0xF	RO	---
[11]	CORE_F_SLPEN	0x0	RW	当停止时核时钟睡眠使能 配置在系统睡眠模式下核时钟是否使能 0 = 禁止; 1 = 使能
[10]	CLKOUT_SLPEN	0x1	RW	当停止时 CLKOUT 时钟睡眠使能 配置在系统睡眠模式下 CLKOUT 时钟是否使能 0 = 禁止; 1 = 使能
[9]	CPM_IPS_SLPEN	0x1	RW	当停止时 CPM_IPS 时钟睡眠使能 配置在系统睡眠模式下CPM_IPS时钟是否使能 **注意：** 当系统进入保持或深度睡眠模式，CPM_IPS_SLPEN 应该被配置为 0 0 = 禁止; 1 = 使能
[8]	TC_SLPEN	0x1	RW	当停止时 TC 时钟睡眠使能 配置在系统睡眠模式下 TC 时钟是否使能 0 = 禁止; 1 = 使能
[7:6]	保留	0x0	RO	---
[5]	VDD_WK_SWOFF	0x0	RW	和 SLPCFGR 寄存器的 SLEEP_MODE 一起配置来进入深度睡眠模式 0 = VDD_WK 电源域打开 1 = VDD_WK 电源域关掉
[4:1]	保留	0x5	RO	---
[0]	VDD_PD_RET	0x0	RW	和 SLPCFGR 寄存器的 SLEEP_MODE 一起配置

比特位	名称	复位值	读写属性	功能说明
				来进入保持模式 0 = VDD_PD 电源域保持关闭（仅限 M4 核） 1 = VDD_PD 电源域保持打开（仅限 M4 核）

6.5.2.44 掉电计数寄存器（PDNCNTR）

偏移地址：0x00D0 ~ 0x00D3　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
PDNCNT0[7:0]							
rw							

23	22	21	20	19	18	17	16
PDNCNT1[7:0]							
rw							

15	14	13	12	11	10	9	8
PDNCNT2[7:0]							
rw							

7	6	5	4	3	2	1	0
PDNCNT3[7:0]							
rw							

图表 6-45：掉电计数寄存器（PDNCNTR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	PDNCNT0[7:0]	0x0	RW	掉电计数 0 配置系统进入睡眠模式的延迟时间
[23:16]	PDNCNT1[7:0]	0x0	RW	掉电计数 1 配置系统进入睡眠模式的延迟时间
[15:8]	PDNCNT2[7:0]	0x0	RW	掉电计数 2 配置系统进入睡眠模式的延迟时间
[7:0]	PDNCNT3[7:0]	0x0	RW	掉电计数 3 配置系统进入睡眠模式的延迟时间

6.5.2.45 上电计数寄存器（PONCNTR）

偏移地址：0x00D4 ~ 0x00D7　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
PONCNT0[7:0]							
rw							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PONCNT1[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PONCNT2[7:0]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PONCNT3[7:0]							
rw							

图表 6-46：上电计数寄存器（PONCNTR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	PONCNT0[7:0]	0x0	RW	上电计数 0 配置系统退出睡眠模式的延迟时间
[23:16]	PONCNT1[7:0]	0x0	RW	上电计数 1 配置系统退出睡眠模式的延迟时间
[15:8]	PONCNT2[7:0]	0x0	RW	上电计数 2 配置系统退出睡眠模式的延迟时间
[7:0]	PONCNT3[7:0]	0x0	RW	上电计数 3 配置系统退出睡眠模式的延迟时间

6.5.2.46 PAD SS3 控制寄存器（PADSS3CR）

偏移地址：0x00D8 ~ 0x00DB　　　　　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
SS3_CTR_EN	保留			SS3_WKPSRC_MSK	SS3_DOUT	SS3_OBE	SS3_PUE
rw	ro			rw	rw	rw	rw

图表 6-47：PAD SS3 控制寄存器（PADSS3CR）

比特位	名称	复位值	读写属性	功能说明
[31:8]	保留	0x0	RO	---
[7]	SS3_CTR_EN	0x0	RW	管脚 SS3 控制使能 设置这个比特来覆盖 PADSS3CR 寄存器的管脚 SS3 控制 0 = 不覆盖；1 = 覆盖
[6:4]	保留	0x0	RO	---
[3]	SS3_WKPSRC_MSK	0x0	RW	管脚 SS3 唤醒源屏蔽 0 = 不屏蔽；1 = 屏蔽
[2]	SS3_DOUT	0x0	RW	管脚 SS3 输出数据 0 = 低；1 = 高
[1]	SS3_OBE	0x0	RW	管脚 SS3 输出使能 0 = 禁止；1 = 使能
[0]	SS3_PUE	0x0	RW	管脚 SS3 上下拉使能 0 = 禁止；1 = 使能

6.5.2.47 唤醒源控制寄存器（WKPSCR）

唤醒源控制寄存器：配置各个模块的唤醒源是否使能。

0 = 模块唤醒源禁止

1 = 模块唤醒源使能

偏移地址：0x00DC ~ 0x00DF　　复位值：0x0000FFFF

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

图表 6-48：唤醒源控制寄存器（WKPSCR）

比特位	名称	复位值	读写属性	功能说明
[31:15]	保留	0x0	RO	---
[14]	唤醒源使能	0x1	RW	PMURTC_PULSE
[13]	唤醒源使能	0x1	RW	USBC
[12]	保留	0x1	RO	---
[11]	保留	0x0	RO	---
[10]	唤醒源使能	0x1	RW	ASYNC_TIMER
[9]	唤醒源使能	0x1	RW	TSI
[8]	唤醒源使能	0x1	RW	I2C
[7]	唤醒源使能	0x1	RW	USI1_RE
[6]	唤醒源使能	0x1	RW	USI1_ATR
[5]	唤醒源使能	0x1	RW	EPORT[5]
[4]	唤醒源使能	0x1	RW	EPORT[4]
[3]	唤醒源使能	0x1	RW	EPORT[3]
[2]	唤醒源使能	0x1	RW	EPORT[2]
[1]	唤醒源使能	0x1	RW	EPORT[1]
[0]	唤醒源使能	0x1	RW	EPORT[0]

6.6 功能描述

6.6.1 时钟源的选择

系统时钟源可以是内部高速晶体振荡器（120MHz）或内部低速晶体振荡器（8MHz）。时钟源选择由本模块的 CSWCFGR 寄存器控制。改变时钟源，需要配置 CSWCFGR，并将 CSWUPDR 寄存器内的 CSWUPD 比特位置 1 且需要等待 CSWSSR 寄存器的切换状态为置 1。

6.6.2 系统时钟源切换配置

步骤 1：确保检测到两个需要切换的时钟源的使能位*EN 及稳定状态位*STABLE 比均为 1，以 OSC120MHz 时钟源切换到 OSC8MHz 时钟源为例。

步骤 2：配置 CSWCFGR 寄存器内的 SYS_SEL = 0，选中 OSC8MHz 时钟源。

6.6.3 系统时钟分频配置

在配置*_DIV 后，必须向 CDIVUPDR[DIVUPD]写“1”才能使配置生效。

6.6.4 时钟源的校准

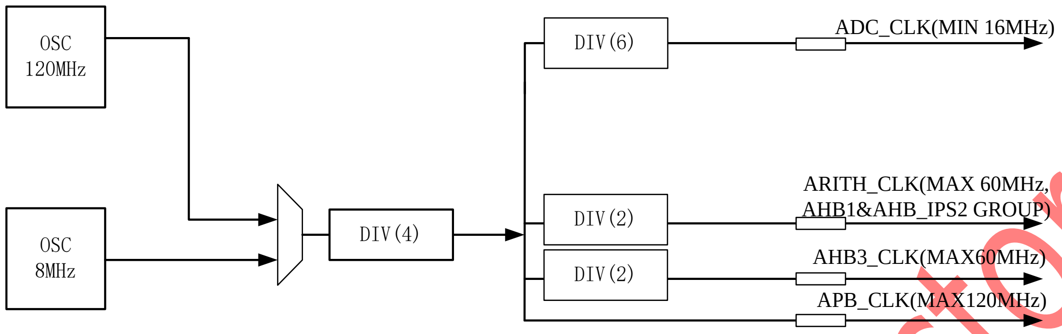
OSC8/120M 的校准，OSC8M 时钟源在没有被 info trim 时，可通过直接写 O8/120MTRIMR 寄存器的值来校准时钟。若在出厂时已经被 info trim 过了，改写 OSC8/120M 时钟源的校准值需要先将 WKPCSR 寄存器内的 FLASH_TRIM_MASK 比特位置 1 后 CPU 才能改写 O8/120MTRIMR 寄存器，trim 值才能生效。

6.6.5 时钟源稳定时间配置

PMU128K、OSC8M、OSC120M、OSCEXT 时钟源的稳定时间若在出厂时没有被 info trim 时，可通过直接写 OSCLSTIMER、OSCHSTIMER、OSCESTIMER 寄存器的值来改变相应的稳定时间。若在出厂时已经被 info trim 过了，改写 PMU128K、OSC8M、OSC120M、OSCEXT 时钟源的稳定时间值需要先将 WKPCSR 寄存器内的 FLASH_TRIM_MASK 比特位置 1 后 CPU 才能改写寄存器 OSCLSTIMER、OSCHSTIMER、OSCESTIMER 的值。

6.6.6 时钟配置示例

时钟的配置通过下例 1 的流程来配置，例如需要 ADC_CLK 来自 OSC120MHz/OSC8MHz 的分频时钟



图表 6-49：时钟配置示意图

具体配置如下：

步骤 1：在确保 OSC120M_EN = 1，SYS_SEL = 1 的前提下，需要先配置 ADC_CLK 的分频系数 ADC_DIV 和系统时钟分频 SYS_DIV，然后向 CDIVUPDR[DIVUPD]写"1"使分频系数生效；

步骤 2：再配置 ADC_DIVEN，然后向 CDIVUPDR[CSWUPD]写"1"使分频使能有效；

步骤 3：最后配置时钟使能 CLK_GTE[7] = 1，在检测到 OSC128M 时钟的 stable 状态位 OSC128M_STABLE = 1 后可得到预期的 ADC_CLK 稳定输出。

6.7 中断描述

6.7.1 VCC_LVDT5V 中断

产生该中断的条件是中断使能位 VCC_IE_LVDT5V = 1 且相应的 VCC_LVDT5V 状态位（VCC_LVDT5V_F 或 VCC_LVDT5V_RT）为 1。**注意：** LVDT5V 的使能位 VCC_EN_LVDT5V 及 LVDT5V 的输出使能位 VCC_OE_LVDT5V 必须配置为 1。

6.7.2 VCC_LVDT18V 中断

产生该中断的条件是中断使能位 VCC_IE_LVDT18 = 1 且相应的 VCC_LVDT18 状态位（VCC_LVDT18_F 或 VCC_LVDT18_RT)为 1。**注意：**LVDT18 的使能位 VCC_EN_LVDT18 及 LVDT18 的输出使能位 VCC_OE_LVDT18 必须配置为 1。

6.7.3 唤醒中断

WKPSCR 寄存器内只要至少一个比特位的值为 1 且 PADWKPINTCR 寄存器内的*_EN 和*_INTM 为 1 则会产生唤醒中断 pad_wkp_int。

7 USB2.0 控制器（USB）

7.1 概述

USB 2.0 高速双向控制器用于 USB 通信相关的管理，可作为主机或从机与另一个 USB 设备进行高速/全速/低速点对点通信，或用作高速/全速 USB 设备的功能控制器。

该模块符合 USB 2.0 高速和全速功能标准，以及 USB2.0-OTG 补充协议标准。

7.2 特性

USB 2.0 高速双向控制器拥有以下特性：

- 所有事务调度硬件执行
- 可作为主机/从机与其他 USB 设备进行点对点通信，或作为 USB 设备的功能控制器
- 同步 FIFORAM 接口
- 符合 USB 2.0 高速（480 Mbps）功能标准和 OTG 补充协议标准
- 支持与一个 USB 设备进行高速/全速/低速点对点通信
- 支持会话请求协议（SRP）和主机协商协议（HNP）
- 支持挂起和恢复
- 可配置最多 15 个附加传输端点和最多 15 个附加接收端点
- 可配置的 FIFO 深度，支持动态 FIFO 大小
- 支持对 FIFO 的 DMA 访问
- 可以软件控制 USB 连接/断开
- 作为从机使用时，不支持 VBUS 选择

7.3 系统时钟

USB 控制器设计为从 AHB 总线时钟获取其系统时钟 CLK，避免 USB 控制器和 AHB 之间的异步问题，并允许对 USB 控制器的寄存器和 FIFO 进行单周期访问。

USB 控制器（以及 AHB 总线）的最小时钟频率取决于所选 UTMI 宽度。如果使用 8 位 UTMI（强烈建议），接近 30MHz 即可。但如果使用 16 位 UTMI，需要将最小频率增加到 48MHz 以上，以确保满足总线换向时间。

7.4 复位

7.4.1 从机模式（Peripheral Mode）

当 USB 控制器作为从机使用，并在 USB 上检测到复位条件时，设备将执行以下操作：
将 FAddr 寄存器设置为 0。
将 Index 寄存器设置为 0。
冲刷所有端点的 FIFO。
复位所有控制/状态寄存器。
打开所有端点中断使能。
产生一个复位中断信号。

如果 UCSR 寄存器的 HS_Enab 位(bit5)被设置为 1，USB 控制器会尝试协商进入高速模式。UCSR 寄存器的 HS_Mode 位(bit4)将指示是否进入高速模式。

如果驱动 USB 控制器的应用软件接收到复位中断，它应该关闭所有打开的管道，并等待总线枚举。

7.4.2 主机模式（Host Mode）

如果在 USB 控制器处于主机模式时设置了 UCSR 寄存器中的 Reset 位，USB 控制器将在总线上生成复位信号。如果 UCSR 寄存器的 HS_Enab 位(bit5)被设置为 1，USB 控制器会在复位时尝试协商进入高速模式。

CPU 应将 Reset 位置 1 保持至少 20 毫秒，以确保目标设备能正确复位。CPU 清零 Reset 位后，USB 控制器将启动帧计数器和事务调度程序。UCSR 寄存器的 HS_Mode(bit4)位将指示是否进入高速模式。

7.5 内部映射和寄存器

USBC 寄存器可以分为以下章节：
USBC 通用寄存器(Common Registers)(偏移地址 0x00-0x0F 和 0x60)，这些寄存器为 USBC 模块提供控制和状态描述，见表 11-1。

USBC 索引寄存器(Indexed Registers)(偏移地址 0x10-0x1F 和 0x62-0x67)，这些寄存器为每个端点提供控制和状态描述。这些寄存器对应的端点由索引寄存器(Index)来选定。在主机模式和从机模式下寄存器具有不同的意义，见表 11-2 和表 11-3。

注意：选中端点 0 时寄存器 CSR0(0x12)，Count0(0x18)，NAKLimit0(0x1A)生效；选中其他端点时 TxCSR(0x12)，RxCount(0x18)，TxInterval(0x1A)寄存器生效。

FIFO 寄存器(FIFO Registers)(偏移地址 0x20-0x5F)，这些寄存器提供 CPU 访问端点 FIFO 的地址，见表

11-4。

DMA 寄存器(DMA Registers)(偏移地址 0x200-0x27F)，这些寄存器为 DMA 操作提供控制和状态描述，见表 11-5。

7.5.1 内存映射

表格 7-1：通用 USB 寄存器

偏移地址	位 15-8	位 7-0	访问权限
0x0000	USCR	Faddr	S/U
0x0002	IntrTx		S/U
0x0004	IntrRx		S/U
0x0006	IntrTxE		S/U
0x0008	IntrRxE		S/U
0x000A	IntrUSBE	IntrUSB	S/U
0x000C	Frame		S/U
0x000E	Testmode	Index	S/U
0x0060	DevCtl		S/U

注意：S = 超级用户访问；U = 普通用户访问

表格 7-2：主机模式下的索引寄存器

偏移地址	位 15-8	位 7-0	访问权限
0x0010	TxMaxP		S/U
0x0012	CSR0		S/U
	TxCSR		S/U
0x0014	RxMaxP		S/U
0x0016	RxCSR		S/U
0x0018	Count0		S/U
	RxCount		S/U
0x001A	NAKLimit0	TxType	S/U
	TxInterval		S/U
0x001C	RxInterval	RxType	S/U
0x0062	RxFIFOsz	TxFIFOsz	S/U
0x0064	TxFIFOadd		S/U
0x0066	RxFIFOadd		S/U

注意：S = 超级用户访问；U = 普通用户访问

表格 7-3：从机模式下的索引寄存器

偏移地址	位 15-8	位 7-0	访问权限
0x0010	TxMaxP		S/U
0x0012	CSR0		S/U
	TxCSR		S/U
0x0014	RxMaxP		S/U
0x0016	RxCSR		S/U
0x0018	Count0		S/U
	RxCount		S/U
0x0062	RxFIFOsz	TxFIFOsz	S/U
0x0064	TxFIFOadd		S/U
0x0066	RxFIFOadd		S/U

注意：S = 超级用户访问；U = 普通用户访问

表格 7-4：FIFO 寄存器

偏移地址	位 31-16	位 15-0	访问权限
0x0020	FIFO0		S/U
0x0024	FIFO1		S/U
0x0028	FIFO2		S/U
…	…		…
0x005B	FIFO15		S/U

注意：S = 超级用户访问；U = 普通用户访问

DMA 控制器有一个中断寄存器，用于指示哪些信道有一个挂起的中断，DMA 一共有 8 个信道，每个信道都有自己的三个控制寄存器。

表格 7-5：DMA 寄存器

偏移地址	位 31-16	位 15-0	访问权限
0x0200	INTR		S/U
0x0204	CNTL(1)		S/U
0x0208	ADDR(1)		S/U
0x020c	COUNT(1)		S/U
0x0210	保留		---
0x0214	CNTL(2)		S/U
0x0218	ADDR(2)		S/U
0x021c	COUNT(2)		S/U
0x0210	保留		---

偏移地址	位 31-16	位 15-0	访问权限
…	…		…
0x0274	CNTL(8)		S/U
0x0278	ADDR(8)		S/U
0x027c	COUNT(8)		S/U

注意：S = 超级用户访问；U = 普通用户访问

7.5.2 USBC 通用寄存器（Common Registers）

7.5.2.1 功能地址寄存器（FAddr）

FAddr 是一个 8 位寄存器，当 USB 控制器作为从机使用时，该寄存器由事务写入其 7 位地址。当 USB 控制器作为主机使用时，该寄存器应设置为在设备枚举期间 SET_ADDRESS 命令发送的值，作为从机设备的地址。

当 USB 控制器作为从机使用时，该寄存器应写入通过 SET_ADDRESS 命令接收到的地址，然后该命令将用于解码后续令牌包中的功能地址。

偏移地址：0x0000　　　　复位值：0x00

7	6	5	4	3	2	1	0
保留	Func Addr[6:0]						
ro	rw						

图表 7-1：功能地址寄存器（FAddr）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6:0]	Func Addr[6:0]	0x0	RW	功能地址： 功能地址应该与通过 SET_ADDRESS 命令获得的地址保持一致。

7.5.2.2 USB 控制和状态寄存器（UCSR）

UCSR 是一个 8 位寄存器，用于控制挂起和恢复信号，以及 USB 控制器的一些基本操作。

偏移地址：0x0001 复位值：0x20

7	6	5	4	3	2	1	0
ISO Update	Soft Conn	HS Enab	HS Mode	Reset	Resume	Suspend Mode	Enable SuspendM
rw	rw	rw	ro	ro	rw	ro	rw

图表 7-2：控制和状态寄存器（UCSR）

比特位	名称	复位值	读写属性	功能说明
[7]	ISO Update	0x0	RW	ISO Update： 0 = 正常 1 = USB 控制器等到从 TxPktRdy 信号置 1 后的 SOF 令牌后，再发送数据包。如果在 SOF 令牌之前接收到 IN 令牌，则发送零长度数据包。
[6]	Soft Conn	0x0	RW	软件控制连接/断开： 启用软件控制连接/断开功能时此位有效。 0 = 令 USB D+/D-成为高阻态。 1 = 打开 USB D+/D-功能。
[5]	HS Enab	0x1	RW	高速模式使能： 0 = 当设备被 USB 主机复位时，USBC 只工作在全速状态。 1 = 当设备被 USB 主机复位时，USBC 会协商进入高速状态。
[4]	HS Mode	0x0	RO	高速模式状态： 在复位状态时，该位监测设备是否进入高速模式，并且在复位结束后（产生复位中断）一直保持为 1。 0 = 设备不工作在高速状态。 1 = 设备工作在高速状态。
[3]	Reset	0x0	RO	复位状态： 当总线上检测到复位信号时，该位置 1。 0 = 没有复位信号 1 = 检测到复位信号
[2]	Resume	0x0	RW	恢复信号位： 当设备工作在挂起模式时，通过 CPU 置 1，产生恢复信号。在 10ms 后（最多为 15ms）CPU 应该清除该位，关闭恢复信号。

比特位	名称	复位值	读写属性	功能说明
				0 = 关闭恢复信号 1 = 产生恢复信号
[1]	Suspend Mode	0x0	RO	挂起模式： 当设备进入挂起模式时，该位置 1。当 CPU 读取 SUSPEND 中断寄存器或设置恢复位时，该位清零。 0 = USBC 没有进入挂起模式 1 = USBC 进入挂起模式
[0]	Enable SuspendM	0x0	RW	使能挂起功能： 该位通过 CPU 置 1 来使能 SUSPENDM 信号。 0 = SUSPENDM 输出禁用。 1 = SUSPENDM 输出使能。

7.5.2.3 发送中断寄存器（IntrTx）

IntrTx 是一个 16 位只读寄存器，读后清零，用于指示当前端点 0 和发送端点 1–15 的中断状态。

偏移地址：0x0002　　复位值：0x0000

15	14	13	12	11	10	9	8
EP15 Tx	EP14 Tx	EP13 Tx	EP12 Tx	EP11 Tx	EP10 Tx	EP9 Tx	EP8 Tx
rc	rc	rc	rc	rc	rc	rc	rc
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EP7 Tx	EP6 Tx	EP5 Tx	EP4 Tx	EP3 Tx	EP2 Tx	EP1 Tx	EP0
rc	rc	rc	rc	rc	rc	rc	rc

图表 7-3：发送中断寄存器（IntrTx）

比特位	名称	复位值	读写属性	功能说明
[15:1]	EPx Tx	0x0	RC	EPx(x 为 1-15）Tx：产生端点 x（x 为 1-15）发送中断 0 = 无对应端点发送中断 1 = 产生对应端点发送中断
[0]	EP0	0x0	RC	产生 EP0 中断： 0 = 没有产生 EP0 中断 1 = 产生 EP0 中断

7.5.2.4 接收中断寄存器（IntrRx）

IntrRx 是一个 16 位只读寄存器，读后清零，用于指示当前接收端点 1–15 的中断状态。

偏移地址：0x0004 复位值：0x0000

15	14	13	12	11	10	9	8
EP15 Rx	EP14 Rx	EP13 Rx	EP12 Rx	EP11 Rx	EP10 Rx	EP9 Rx	EP8 Rx
rc	rc	rc	rc	rc	rc	rc	rc
7	6	5	4	3	2	1	0
EP7 Rx	EP6 Rx	EP5 Rx	EP4 Rx	EP3 Rx	EP2 Rx	EP1 Rx	保留
rc	rc	rc	rc	rc	rc	rc	ro

图表 7-4：接收中断寄存器（IntrRx）

比特位	名称	复位值	读写属性	功能说明
[15:1]	EPx Rx	0x0	RC	EPx(x 为 1-15) Rx：产生端点 x（x 为 1-15）接收中断 0 = 无对应端点接收中断 1 = 产生对应端点接收中断
[0]	保留	---	RO	---

7.5.2.5 发送中断使能寄存器（IntrTxE）

IntrTxE 是一个 16 位寄存器，为发送中断提供中断使能位。当中断使能位为 1，IntrTx 寄存器中的中断置 1 时会产生 MC_NINT 信号。当中断使能位为 0，IntrTx 中的中断置 1 时不会产生 MC_NINT。复位时，端点 0 和设计中包含的发送端点对应的位置 1，其余位置 0。

注意：未配置的端点对应的位始终为 0。

偏移地址：0x0006 复位值：0xFFFF

15	14	13	12	11	10	9	8
EP15 TxE	EP14 TxE	EP13 TxE	EP12 TxE	EP11 TxE	EP10 TxE	EP9 TxE	EP8 TxE
rw	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
EP7 TxE	EP6 TxE	EP5 TxE	EP4 TxE	EP3 TxE	EP2 TxE	EP1 TxE	EP0E
rw	rw	rw	rw	rw	rw	rw	rw

图表 7-5：发送中断使能寄存器（IntrTxE）

比特位	名称	复位值	读写属性	功能说明
[15:1]	EPx TxE	0x7FFF	RW	EPx(x 为 1-15) TxE： 使能端点 x（x 为 1-15）发送中断 0 = 禁用对应端点发送中断 1 = 使能对应端点发送中断
[0]	EP0E	0x1	RW	EP0 中断使能： 0 = 禁用 EP0 中断 1 = 使能 EP0 中断

7.5.2.6 接收中断使能寄存器（IntrRxE）

IntrRxE 是一个 16 位寄存器，为接收中断提供中断使能位。当中断使能位为 1，IntrRx 寄存器中的中断置 1 时会产生 MC_NINT 信号。当中断使能位为 0，IntrRx 中的中断置 1 时不会产生 MC_NINT。复位时，设计中包含的接收端点对应的位置 1，其余位置 0。

注意：未配置的端点对应的位始终为 0。

偏移地址：0x0008　　　　复位值：0xFFFE

15	14	13	12	11	10	9	8
EP15 RxE	EP14 RxE	EP13 RxE	EP12 RxE	EP11 RxE	EP10 RxE	EP9 RxE	EP8 RxE
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EP7 RxE	EP6 RxE	EP5 RxE	EP4 RxE	EP3 RxE	EP2 RxE	EP1 RxE	保留
rw	rw	rw	rw	rw	rw	rw	ro

图表 7-6：接收中断使能寄存器（IntrRxE）

比特位	名称	复位值	读写属性	功能说明
[15:1]	EPx RxE	0x7FFF	RW	EPx(x 为 1-15) RxE： 使能端点 x（x 为 1-15）接收中断 0 = 禁用对应端点发送中断 1 = 使能对应端点发送中断
[0]	保留	0x0	RO	---

7.5.2.7 USB 中断寄存器（IntrUSB）

IntrUSB 是一个 8 位只读寄存器，用于指示哪些 USB 中断当前处于激活状态。
注意：读取此寄存器时，所有激活的中断都将被清除。

偏移地址：0x000A　　复位值：0x00

7	6	5	4	3	2	1	0
VBus Error	Sess Req	Discon	Conn	SOF	Reset/ Babble	Resume	Suspend
rc	rc	rc	rc	rc	rc	rc	rc

图表 7-7：USB 中断寄存器（IntrUSB）

比特位	名称	复位值	读写属性	功能说明
[7]	VBus Error	0x0	RC	VBus 错误标识： 0 = Vbus 电压不低于 VBus 有效阈值 1 = Vbus 电压低于 VBus 有效阈值 **注意：**仅当 USB 控制器是“A”设备时有效。
[6]	Sess Req	0x0	RC	会话请求标识： 0 = 未检测到会话请求 1 = 检测到会话请求 **注意：**仅当 USB 控制器是“A”设备时有效。
[5]	Discon	0x0	RC	USB 设备产生断开中断： 0 = 无断开中断产生 1 = 断开中断产生
[4]	Conn	0x0	RC	检测到设备连接标识： 0 = 未检测到设备连接 1 = 检测到设备连接 **注意：**仅在主机模式下有效。在所有速度模式下都有效。
[3]	SOF	0x0	RC	产生帧开始中断： 0 = 没有检测到帧开始令牌包 1 = 检测到帧开始令牌包
[2]	Reset/ Bubble	0x0	RC	产生 USB 复位中断：（从机模式） 0 = 没有检测到 USB 复位信号 1 = 检测到 USB 复位信号 产生 Bubble 中断：（主机模式） 0 = 没有检测到 Bubble 信号 1 = 检测到 Bubble 信号

比特位	名称	复位值	读写属性	功能说明
[1]	Resume	0x0	RC	产生 USB 恢复中断： 0 = 没有检测到恢复信号 1 = 当 USB 处于挂起模式时，检测到恢复信号
[0]	Suspend	0x0	RC	产生挂起中断： 0 = 没有检测到挂起信号 1 = 检测到挂起信号

7.5.2.8 USB 中断使能寄存器（IntrUSBE）

IntrUSBE 是一个 8 位寄存器，为 IntrUSB 中的每个中断提供中断使能位。

偏移地址：0x000B　　复位值：0x06

7	6	5	4	3	2	1	0
VBus ErrorE	Sess ReqE	DisconE	ConnE	SOFE	ResetE/ BabbleE	ResumeE	SuspendE
rw	rw	rw	rw	rw	rw	rw	rw

图表 7-8：USB 中断使能寄存器（IntrUSBE）

比特位	名称	复位值	读写属性	功能说明
[7]	VBus ErrorE	0x0	RW	VBus Error 中断使能： 0 = VBus Error 中断禁用 1 = VBus Error 中断使能
[6]	Sess ReqE	0x0	RW	Sess Req 中断使能： 0 = Sess Req 中断禁用 1 = Sess Req 中断使能
[5]	DisconE	0x0	RW	USB 设备断开中断使能： 0 = 断开中断禁用；1 = 断开中断使能
[4]	ConnE	0x0	RW	Conn 中断使能： 0 = Conn 中断禁用；1 = Conn 中断使能
[3]	SOFE	0x0	RW	帧开始中断使能： 0 = 帧开始中断禁用；1 = 帧开始中断使能
[2]	ResetE/ BubbleE	0x1	RW	USB 复位中断使能：（从机模式） 0 = USB 复位中断禁用 1 = USB 复位中断使能 Bubble 中断使能：（主机模式） 0 = Bubble 中断禁用 1 = Bubble 中断使能
[1]	ResumeE	0x1	RW	USB 恢复中断使能：

比特位	名称	复位值	读写属性	功能说明
				0 = USB 恢复中断禁用；1 = USB 恢复中断使能
[0]	SuspendE	0x0	RW	挂起中断使能： 0 = 挂起中断禁用；1 = 挂起中断使能

7.5.2.9 帧数寄存器（Frame）

Frame 是一个 16 位只读寄存器，用来保存上次接收到的帧数。

偏移地址：0x000C　　复位值：0x0000

15	14	13	12	11	10	9	8
保留					Frame Number[10:8]		
ro					ro		

7	6	5	4	3	2	1	0
Frame Number[7:0]							
ro							

图表 7-9：帧数寄存器（Frame）

比特位	名称	复位值	读写属性	功能说明
[15:11]	保留	0x0	RO	---
[10:0]	Frame Number[10:0]	0x0	RO	帧数计数： 值为上次接收到的帧数。

7.5.2.10 端点索引寄存器（Index）

Index 是一个 4 位寄存器，用于确定访问哪个端点控制/状态寄存器。每个 Tx 端点和每个 Rx 端点都有自己的一组控制/状态寄存器，地址 0x0100-0x01FF。此外还有一组 Tx 控制/状态寄存器和一组 Rx 控制/状态寄存器，地址 0x0010–0x0019。

偏移地址：0x000E 复位值：0x00

7	6	5	4	3	2	1	0
保留				Index[3:0]			
ro				rw			

图表 7-10：端点索引寄存器（Index）

比特位	名称	复位值	读写属性	功能说明
[7:4]	保留	0x0	RO	---
[3:0]	Index[3:0]	0x0	RW	端点索引： 寄存器的值表示当前选择的端点号。

7.5.2.11 测试模式寄存器（Testmode）

Testmode 是一个 8 位寄存器，主要用于将 USB 控制器配置到 USB 2.0 规范中描述的四种高速操作测试模式中的一种，以响应 SET FEATURE:TESTMODE 命令。正常运行时不使用。

偏移地址：0x000F 复位值：0x00

7	6	5	4	3	2	1	0
Force_Host	FIFO_Access	Force_FS	Force_HS	Test_Packet	Test_K	Test_J	Test_SE0_NAK
rw	rw	rw	rw	rw	rw	rw	rw

图表 7-11：测试模式寄存器（Testmode）

比特位	名称	复位值	读写属性	功能说明
[7]	Force_Host	0x0	RW	主机测试模式： 当该位置 1 时，不管是否连接任何外围设备，模块强制进入主机模式。此状态下 CID 输入、主机断开连接和线路状态信号都将被忽略。设备保持锁定在主机模式，直到该位清零。如果设备断开连接但不清零此位，下次启动时会重新进入主机模式。在此模式下，可以从 DevCtl 寄存器的第 7 位读取来自 PHY 的 HOSTDISCON 信号的状态。速度模式由 Force_HS 和 Force_FS 位确定，如下

比特位	名称	复位值	读写属性	功能说明
				所示： Force_HS Force_FS 速度模式 0 0 低速 0 1 全速 1 0 高速 1 1 未定义 0 = 正常模式 1 = 强制 USBC 进入到主机测试模式
[6]	FIFO_Acess	0x0	RW	FIFO_Access 测试： 当该位置 1 时，测试数据包从端点 0 的发送 FIFO 传输到端点 0 接收 FIFO 中，该位自动清零。 0 = 正常模式 1 = 进行 FIFO_Access 测试
[5]	Force_FS	0x0	RW	全速(Full Speed)模式测试模式： 当该位置 1 时，当接收到 USB 复位信号，USBC 进入到全速模式 0 = 正常模式 1 = 强制 USBC 进入到全速模式
[4]	Force_HS	0x0	RW	高速(High Speed)模式测试模式： 当该位置 1 时，当接收到 USB 复位信号，USBC 进入到高速模式 0 = 正常模式 1 = 强制 USBC 进入到高速模式
[3]	Test_Packet	0x0	RW	数据包测试模式： 在高速模式下，当该位置 1 时，USB 进入到数据包测试模式。USBC 在总线上重复发送一个 53 字节的测试数据包。该测试数据包具有固定格式，必须在进入测试模式前，测试数据包加载到 EP0 FIFO 中。 0 = 正常模式 1 = 强制 USBC 进入到数据包测试模式
[2]	Test_K	0x0	RW	K 状态测试模式： 在高速模式下，当该位置 1 时，USB 进入到 K 状态测试模式。在该模式下，USBC 在总线上传输连续的 K。 0 = 正常模式 1 = 强制 USBC 进入到 K 状态测试模式
[1]	Test_J	0x0	RW	J 状态测试模式： 在高速模式下，当该位置 1 时，USB 进入到 J 状

比特位	名称	复位值	读写属性	功能说明
				态测试模式。在该模式下，USBC 在总线上传输连续的 J。 0 = 正常模式 1 = 强制 USBC 进入到 J 状态测试模式
[0]	Test_SE0_NAK	0x0	RW	SE0_NAK 状态测试模式： 在高速模式下，当该位置 1 时，USB 进入到 SE0_NAK 状态测试模式。在该模式下 USBC 工作于高速模式，对任何的有效的 IN 包都回应 NAK 信号。 0 = 正常模式 1 = 强制 USBC 进入到 SE0_NAK 状态测试模式

7.5.2.12 设备控制寄存器（DevCtl）

DevCtl 是一个 8 位寄存器，用于选择 USB 控制器是在主机模式还是从机模式下运行，以及控制和监视 USB 的 VBus 线路。

偏移地址：0x0060　　复位值：0x80

7	6	5	4	3	2	1	0
B-Device	FSDev	LSDev	VBus[1:0]		Host Mode	Host Req	Session
ro	ro	ro	ro		rw	rw	rw

图表 7-12：设备控制寄存器（DevCtl）

比特位	名称	复位值	读写属性	功能说明
[7]	B-Device	0x1	RO	设备 A-B 指示： 此位指示 USB 控制器是作为 A 还是 B 设备运行。 0 = “A”设备 1 = “B”设备 **注意：**仅在会话进行时有效。
[6]	FSDev	0x0	RO	高速/全速设备指示： 0 = 未检测到全速或高速设备连接到端口 1 = 检测到全速或高速设备连接到端口 （当设备复位时，通过检测高速调频信号来区分高速设备和全速设备。） **注意：**仅在主机模式下有效。
[5]	LSDev	0x0	RO	低速设备指示：

比特位	名称	复位值	读写属性	功能说明
				0 = 未检测到全速或高速设备连接到端口 1 = 检测到低速设备连接到端口 **注意：**仅在主机模式下有效。
[4:3]	VBus[1:0]	0x0	RO	Vbus 电压指示： 00 = 低于 SessionEnd 电压 01 = 高于 SessionEnd 电压，低于 AValid 电压 10 = 高于 Avalid 电压，低于 VBusValid 电压 11 = 高于 VbusValid 电压
[2]	Host Mode	0x0	RO	主机模式指示： 0 = 当前 USB 控制器未处于主机模式 1 = 当前 USB 控制器处于主机模式
[1]	Host Req	0x0	RW	主机请求： 0 = 不请求成为主机 1 = USB 控制器将在进入挂起模式时协商成为主机（HNP），当主机协商完成时此位自动清零。 **注意：**仅限当前是“B”设备有效。
[0]	Session	0x0	RW	会话控制： 当作为“A”设备运行时，该位由 CPU 置 1/置 0 以启动/结束会话。当作为“B”设备运行时，该位在会话开始/结束时由 USB 控制器置 1/置 0。CPU 还可以置 1 以启动会话请求协议（SRP），或在挂起模式下置 0 以执行软件断开连接操作。 0 = 结束会话 1 = 开启会话

7.5.3 USBC 索引寄存器（Indexed Registers）

7.5.3.1 端点 0 控制和状态寄存器（CSR0）

CSR0 是一个 16 位寄存器，为端点 0 提供控制位和状态位。寄存器的解释取决于 USB 控制器是作为从机还是作为主机。用户还应注意，当读取寄存器时返回的值反映所获得的状态，例如写入寄存器的结果。

注意：该寄存器仅在索引寄存器为 0 时可访问，且在主机模式和从机模式下具有不同功能。

7.5.3.1.1 从机模式

偏移地址：0x0012（索引寄存器设置为 0 时）　　　　复位值：0x0000

15	14	13	12	11	10	9	8
保留							FlushFIFO
ro							wo

7	6	5	4	3	2	1	0
ServicedSetupEnd	ServicedRxPktRdy	SendStall	SetupEnd	DataEnd	SendStall	TxPktRdy	RxPktRdy
wo	wo	wo	ro	r/w1o	rc	r/w1o	ro

图表 7-13：端点 0 控制和状态寄存器（CSR0）从机模式

比特位	名称	复位值	读写属性	功能说明
[15:9]	保留	0x0	RO	---
[8]	FlushFIFO	0x0	WO	清空 FIFO： 写 1 = 清除下一个即将从端点 0 的 FIFO 发送/接收的数据包，FIFO 的指针复位，TxPktRdy 位和 RxPktRdy 位清零。
[7]	ServicedSetupEnd	0x0	WO	清除 SetupEnd 位： 写 1 = 清除 SetupEnd 位。
[6]	ServicedRxPktRdy	0x0	WO	清除 RxPktRdy 位： 写 1 = 清除 RxPktRdy 位。
[5]	SendStall	0x0	WO	发送 Stall： 写 1 = 终止当前传输，并发送 Stall 握手包。
[4]	SetupEnd	0x0	RO	建立终止： 如果一个控制传输在 DataEnd 位被置 1 前就终止了，该位会被置 1。同时产生中断，并清空 FIFO。该位通过 ServicedSetupEnd 位写 1 来清除。 0 = 正常 1 = 控制传输在 DataEnd 位被置 1 前就终止

比特位	名称	复位值	读写属性	功能说明
[3]	DataEnd	0x0	R/W1O	数据传输终止： CPU 应在如下情况置 1 该位，该位自动清零： 为最后一个数据包设置 TxPktRdy 时。 在读取最后一个数据包后清除 RxPktRdy 时。 为零长度数据包设置 TxPktRdy 时。 0 = 正常 1 = 终止控制传输的数据阶段
[2]	SentStall	0x0	RC	Stall 握手包发送标志位： 当 Stall 握手包发送后，该位被置 1，读后清零。 0 = 正常 1 = Stall 握手包已发送
[1]	TxPktRdy	0x0	R/W1O	发送完成标志位： 当数据包写入到 FIFO 后，CPU 应将该位置 1。 当包发送完成后，该位自动清零，同时产生中断（若中断已使能）。 0 = FIFO 中数据包已发送完成 1 = FIFO 中数据包未发送完成
[0]	RxPktRdy	0x0	RO	接收完成标志位： 当一个数据包被接收后该位被自动置 1，同时产生中断（若中断已使能）。对 ServicedRxPktRdy 位写 1 可以清零此位。 0 = 未接收到数据包 1 = 已接收到数据包

7.5.3.1.2 主机模式

偏移地址：0x0012（索引寄存器设置为 0 时） 复位值：0x0000

15	14	13	12	11	10	9	8
保留				Dis Ping	保留		FlushFIFO
ro				rw	ro		wo

7	6	5	4	3	2	1	0
NAK Timeout	StatusPkt	ReqPkt	Error	SetupPkt	RxStall	TxPktRdy	RxPktRdy
r/w0c	r/w1o	r/w1o	r/w0c	rw	r/w0c	r/w1c	r/w0c

图表 7-14：端点 0 控制和状态寄存器（CSR0）主机模式

比特位	名称	复位值	读写属性	功能说明
[15:12]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[11]	Dis Ping	0x0	RW	禁止 Ping: 该位为 1 后，CPU 在高速控制传输的数据和状态阶段不发出 PING 令牌（用于不响应 PING 的设备）。 0 = 正常 1 = 禁止发送 PING 令牌
[10:9]	保留	0x0	RO	---
[8]	FlushFIFO	0x0	WO	清空 FIFO: 写 1 = 清除下一个即将从端点 0 的 FIFO 发送/接收的数据包，FIFO 的指针复位，TxPktRdy 位和 RxPktRdy 位清零。
[7]	NAK Timeout	0x0	R/W0C	NAK 超时: 当接收到 NAK 响应后，端点 0 停止的时间超过 NAKLimit0 寄存器设置为 NAK 限制的时间，此位将被置 1。CPU 应清零此位，以允许端点继续传输。 0 = 正常 1 = 超过 NAK 后 NAKLimit0 限制时间
[6]	StatusPkt	0x0	R/W1O	状态阶段包: CPU 在将 TxPktRdy 或 ReqPkt 位置 1 的同时置 1 该位，用以执行状态阶段的事务。设置此位可确保将数据切换设置为 1，以将 DATA1 数据包用于状态阶段事务。 0 = 不在状态阶段 1 = 在状态阶段
[5]	ReqPkt	0x0	R/W1O	请求包: CPU 向该位写 1 以请求一个 IN 事务。RxPktRdy 位置 1 时此位自动清除。
[4]	Error	0x0	R/W0C	错误: 当三次事务尝试而未收到从机的握手数据包时，此位被置 1，并生成中断。CPU 应该写 0 清零此位。 0 = 正常 1 = 未接收到握手包
[3]	SetupPkt	0x0	RW	建立包: CPU 在 TxPktRdy 位置 1 的同时置 1 该位，以发送事务的 SETUP 令牌代替 OUT 令牌。 0 = 正常 1 = 发送 SETUP 令牌代替 OUT 令牌

比特位	名称	复位值	读写属性	功能说明
[2]	RxStall	0x0	R/W0C	接收到 STALL 握手包： 当接收到 STALL 握手包后，此位被置 1。CPU 应该写 0 清零此位。 0 = 正常 1 = 接收到 STALL 握手包。
[1]	TxPktRdy	0x0	R/W1C	发送完成位： 当数据包写入到 FIFO 后，CPU 应将该位置 1。当数据包传输完成后，该位自动清零，同时产生中断（若中断已使能）。 0 = 无数据包加载到 FIFO 中 1 = 数据包已加载到 FIFO 中
[0]	RxPktRdy	0x0	R/W0C	接收完成位： 当数据包已接收后，该位置 1。该位置 1 时，同时产生中断（若中断使能）。ServicedRxPktRdy 位置 1 后，该位清零。 1 = 数据包已接收 0 = 无数据包接收

7.5.3.2 端点 0 计数寄存器（Count0）

Count0 是一个 7 位只读寄存器，指示端点 0 FIFO 中接收的数据字节数。返回的值随着 FIFO 内容的更改而更改，并且仅在 RxPktRdy 为 1 时有效。

注意：该寄存器仅在索引寄存器为 0 时可访问。

偏移地址：0x0018（索引寄存器设置为 0 时） 复位值：0x00

7	6	5	4	3	2	1	0
保留	EP0 Rx Counter[6:0]						
ro	ro						

图表 7-15：端点 0 计数寄存器（Count0）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6:0]	EP0 Rx Counter[6:0]	0x0	RO	端点 0 接收到的字节数： 该寄存器值随着 FIFO 的内容改变而改变，值为端点 0 FIFO 接收到的字节数，仅在 RxPktRdy 为 1 时有效。

7.5.3.3 端点 0 超时寄存器（NAKLimit0）

NAKLimit0 是一个 5 位寄存器，决定端点 0 接收到连续 NAK 响应时，触发超时所需的帧/微帧（高速传输）的数量。设置的帧/微帧数为 2*(m-1)（其中 m 是寄存器中设置的值，有效值为 2–16）。如果主机从目标接收的 NAK 响应的帧数超过此寄存器中设置的帧数限制，则端点传输将被中止。

注意：该寄存器仅在索引寄存器为 0 时可访问，仅限主机模式下有效，寄存器值为 0 或 1 时功能不启用。

偏移地址：0x001B（索引寄存器设置为 0 时） 复位值：0x00

7	6	5	4	3	2	1	0
保留			NAK LIMIT0[4:0]				
ro			rw				

图表 7-16：端点 0 超时寄存器（NAKLimit0）

比特位	名称	复位值	读写属性	功能说明
[7:5]	保留	0x0	RO	---
[4:0]	NAKLIMIT0[4:0]	0x0	RW	NAK 帧超时限制： 设置的帧/微帧数为 2*(m-1)，m 位寄存器值，如果 NAK 响应的帧数超过此数，则端点传输将被中止。 有效值为 2-16。

7.5.3.4 端点 0 超时寄存器（NAKLimit0）

TxMaxP 是一个 16 位寄存器，定义在单个操作中可以通过选定的 Tx 端点传输的最大数据量。每个 Tx 端点（端点 0 除外）都有一个 TxMaxP 寄存器。

偏移地址：0x0010 复位值：0x0000

15	14	13	12	11	10	9	8
保留					TXMAXP[10:8]		
ro					rw		

7	6	5	4	3	2	1	0
TXMAXP[7:0]							
rw							

图表 7-17：端点 0 超时寄存器（NAKLimit0）

比特位	名称	复位值	读写属性	功能说明
[15:11]	保留	0x0	RO	---
[10:0]	TXMAXP[10:0]	0x0	RW	发送包最大尺寸：

比特位	名称	复位值	读写属性	功能说明
				该寄存器值决定一次传输的最大字节数。设置的值可以高达 1024 字节，但受 USB 规范对全速和高速操作中的批量、中断和同步传输数据包大小的限制。

7.5.3.5 发送控制和状态寄存器（TxCSR）

TxCSR 是一个 16 位寄存器，为当前选定的 Tx 端点的控制位和状态位。每个 Tx 端点都有一个独立的 TxCSR 寄存器，通过索引寄存器切换选择。

注意：该寄存器在主机模式和从机模式下具有不同功能。

7.5.3.5.1 从机模式

偏移地址：0x0012　　复位值：0x0000

15	14	13	12	11	10	9	8
AutoSet	ISO	Mode	DMAReqEnab	FrcDataTog	DMAReqMode	保留	
rw	rw	rw	rw	rw	rw	ro	

7	6	5	4	3	2	1	0
IncompTx	ClrDataTog	SentStall	SendStall	FlushFIFO	UnderRun	FIFOnotEmpty	TxPktRdy
r/w0c	wo	r/w0c	rw	r/w1o	r/w0c	ro	r/w1o

图表 7-18：发送控制和状态寄存器（TxCSR）从机模式

比特位	名称	复位值	读写属性	功能说明
[15]	AutoSet	0x0	RW	自动设置 TxPktRdy： 该位为 1 时，当加载到 FIFO 的数据达到发送包最大尺寸（TxMaxP）后，TxPktRdy 会自动置 1。如果加载的数据小于 TxMaxP，TxPktRdy 需手动置 1。 0 = 正常 1 = TxPktRdy 自动置 1
[14]	ISO	0x0	RW	设置同步传输： 0 = 设置此端点进行批量或中断传输 1 = 设置此端点进行同步传输
[13]	Mode	0x0	RW	设置方向： 0 = 设置此端点方向为接收 1 = 设置此端点方向为发送

比特位	名称	复位值	读写属性	功能说明
[12]	DMAReqEnab	0x0	RW	DMA 请求使能: 0 = 禁止此端点的 DMA 请求 1 = 使能此端点的 DMA 请求
[11]	FrcDataTog	0x0	RW	强制数据 DATA0/DATA1 切换: 0 = 正常 1 = 强制端点数据 DATA0 和 DATA1 切换，并清除 FIFO 中的数据包，不关心是否接收到 ACK。可以在中断端点时使用，用于同步端点时的通信速率反馈。
[10]	DMAReqMode	0x0	RW	DMA 请求模式: 0 = 选择 DMA 模式 0 1 = 选择 DMA 模式 1
[9:8]	保留	0x0	RO	---
[7]	IncompTx	0x0	R/W0C	发送包不匹配: 当端点被用于高带宽的同步/中断传输时，如果大数据包被分成 2 或 3 个数据包进行传输，但是接收到的 IN 令牌数量不足，无法发送所有数据，此位被置 1。向此位写 0 可清零此位。 0 = 发送包正常 1 = 发送包所需 IN 令牌数不匹配
[6]	ClrDataTog	0x0	WO	重置数据切换: 写 1 将端点数据切换重置为 DATA0。
[5]	SentStall	0x0	R/W0C	已发送 STALL 握手包: 在 STALL 握手包被发送后此位会被置 1，同时 FIFO 被清空，TxPktRdy 位被清零。 CPU 应该写 0 以清零此位。
[4]	SendStall	0x0	RW	发送 STALL 握手包: 0 = 正常 1 = 保持对接受到的 IN 令牌发送 STALL 握手包
[3]	FlushFIFO	0x0	R/W1O	清除 FIFO: 当该位置 1 时，清除即将从端点 FIFO 发送的数据包。FIFO 的指针复位，清零 TxPktRdy 位。该位只在 TxPktRdy 位为 1 时有效，中止当前包装载进 FIFO。 0 = 正常 1 = 清除即将从端点 FIFO 发送的数据包

比特位	名称	复位值	读写属性	功能说明
[2]	UnderRun	0x0	R/W0C	欠载标志位： 如果在 TxPktRdy 位为 0 时接收到 IN 令牌，此位会被置 1。CPU 应该写 0 清除此位。 0 = 正常 1 = TxPktRdy 位为 0 时接收到 IN 令牌
[1]	FIFONotEmpty	0x0	RO	FIFO 非空标志位： 0 = FIFO 中无数据 1 = FIFO 中有数据。
[0]	TxPktRdy	0x0	R/W1O	发送完成位： 当数据包写入到 FIFO 后，CPU 应将该位置 1。当数据包传输完成后，该位自动清零，同时产生中断（若中断已使能）。 0 = 无数据包加载到 FIFO 中 1 = 数据包已加载到 FIFO 中 在将第二个数据包加载到双缓冲 FIFO 之前，TxPktRdy 也会被自动清除。

7.5.3.5.2 主机模式

偏移地址：0x0012　　　　复位值：0x0000

15	14	13	12	11	10	9	8
AutoSet	保留	Mode	DMAReqEnab	FrcDataTog	DMAReqMode	保留	
rw	ro	rw	rw	rw	rw	ro	

7	6	5	4	3	2	1	0
Nak Timeout/ IncompTx	ClrDataTog	RxStall	保留	FlushFIFO	Error	FIFONotEmpty	TxPktRdy
r/w0c	wo	r/w0c	ro	r/w1o	r/w0c	ro	r/w1o

图表 7-19：发送控制和状态寄存器（TxCSR）主机模式

比特位	名称	复位值	读写属性	功能说明
[15]	AutoSet	0x0	RW	自动设置 TxPktRdy： 该位为 1 时，当加载到 FIFO 的数据达到发送包最大尺寸（TxMaxP）后，TxPktRdy 会自动置 1。如果加载的数据小于 TxMaxP，TxPktRdy 需手动置 1。 0 = 正常 1 = TxPktRdy 自动置 1

比特位	名称	复位值	读写属性	功能说明
[14]	保留	0x0	RO	---
[13]	Mode	0x0	RW	设置方向： 0 = 设置此端点方向为接收 1 = 设置此端点方向为发送
[12]	DMAReqEnab	0x0	RW	DMA 请求使能： 0 = 禁止此端点的 DMA 请求 1 = 使能此端点的 DMA 请求
[11]	FrcDataTog	0x0	RW	强制数据 DATA0/DATA1 切换： 0 = 正常 1 = 强制端点数据 DATA0 和 DATA1 切换，并清除 FIFO 中的数据包，不关心是否接收到 ACK。可以在中断端点时使用，用于同步端点时的通信速率反馈。
[10]	DMAReqMode	0x0	RW	DMA 请求模式： 0 = 选择 DMA 模式 0 1 = 选择 DMA 模式 1
[9:8]	保留	0x0	RO	---
[7]	Nak Timeout/ IncompTx	0x0	R/W0C	**注意：**此位在不同事务下有不同意义： Nak Timeout：（仅限批量事务）（R/W0C） 当接收到 NAK 响应后，Tx 端点停止的时间超过 TxInterval 寄存器设置为 NAK 限制的时间，此位将被置 1。CPU 应清零此位，以允许端点继续传输。 0 = 正常 1 = 超过 NAK 后 TxInterval 限制时间 IncompTx：（仅限高带宽中断事务）（RO） 如果发送数据包后未从设备接收到响应，此位将被置 1。 0 = 正常 1 = 设备无响应
[6]	ClrDataTog	0x0	WO	重置数据切换： 写 1 将端点数据切换重置为 DATA0。
[5]	RxStall	0x0	R/W0C	接收到 STALL 握手包： 当接收到 STALL 握手包后，此位被置 1，停止任何正在进行的 DMA 请求，FIFO 被清空，TxPktRdy 将被清零。CPU 应该写 0 清零此位。 0 = 正常 1 = 接收到 STALL 握手包。
[4]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[3]	FlushFIFO	0x0	R/W1O	清除 FIFO： 当该位置 1 时，清除即将从端点 FIFO 发送的数据包。FIFO 的指针复位，清零 TxPktRdy 位。该位只在 TxPktRdy 位为 1 时有效，中止当前包装载进 FIFO。 0 = 正常 1 = 清除即将从端点 FIFO 发送的数据包
[2]	Error	0x0	R/W0C	错误位： 当三次尝试发送数据包而未收到握手数据包时，此位被置 1，并生成中断，清零 TxPktRdy 位，清空 FIFO。CPU 应该写 0 清零此位。该位仅当端点工作在批量或中断传输下有效。 0 = 正常；1 = 未接收到握手包
[1]	FIFONotEmpty	0x0	RO	FIFO 非空标志位： 0 = FIFO 中无数据；1 = FIFO 中有数据。
[0]	TxPktRdy	0x0	R/W1O	发送完成位： 当数据包写入到 FIFO 后，CPU 应将该位置 1。当数据包传输完成后，该位自动清零，同时产生中断（若中断已使能）。 0 = 无数据包加载到 FIFO 中 1 = 数据包已加载到 FIFO 中 在将第二个数据包加载到双缓冲 FIFO 之前，TxPktRdy 也会被自动清除。

7.5.3.6 接收包最大尺寸寄存器（RxMaxP）

RxMaxP 是一个 16 位寄存器，定义在单个操作中可以通过选定的 Rx 端点传输的最大数据量。每个 Rx 端点（端点 0 除外）都有一个 RxMaxP 寄存器。

偏移地址：0x0014 复位值：0x0000

15	14	13	12	11	10	9	8
保留					RXMAXP[10:8]		
ro					rw		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXMAXP[7:0]							
rw							

图表 7-20：接收包最大尺寸寄存器（RxMaxP）

比特位	名称	复位值	读写属性	功能说明
[15:11]	保留	0x0	RO	---
[10:0]	RXMAXP[10:0]	0x0	RW	接收包最大尺寸： 该寄存器值决定一次传输的最大字节数。设置的值可以高达 1024 字节，但受 USB 规范对全速和高速操作中的批量、中断和同步传输数据包大小的限制。

7.5.3.7 接收控制和状态寄存器（RxCSR）

RxCSR 是一个 16 位寄存器，为当前选定的 Rx 端点的控制位和状态位。每个 Rx 端点都有一个独立的 RxCSR 寄存器，通过索引寄存器切换选择。

该寄存器在主机模式和从机模式下具有不同功能。

7.5.3.7.1 从机模式

偏移地址：0x0016　　　　复位值：0x0000

15	14	13	12	11	10	9	8
AutoClear	ISO	DMAReqEnab	DisNyet/PID_Error	DMAReqMode	保留		IncompRx
rw	rw	rw	rw	rw	ro		ro
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ClrDataTog	SentStall	SendStall	FlushFIFO	DataError	OverRun	FIFOFull	RxPktRdy
wo	r/w0c	rw	r/w1o	ro	r/w0c	ro	r/w0c

图表 7-21：接收控制和状态寄存器（RxCSR）从机模式

比特位	名称	复位值	读写属性	功能说明
[15]	AutoClear	0x0	RW	自动清零 RxPktRdy： 该位为 1 时，当有接收包最大尺寸(RxMaxP)大小的数据包从 FIFO 卸载后，RxPktRdy 位会自动置 0。如果卸载的数据小于 RxMaxP，RxPktRdy 需手动置 0。 0 = 正常；1 = RxPktRdy 自动置 0
[14]	ISO	0x0	RW	设置同步传输： 0 = 设置此端点进行批量或中断传输 1 = 设置此端点进行同步传输
[13]	DMAReqEnab	0x0	RW	DMA 请求使能： 0 = 禁止此端点的 DMA 请求 1 = 使能此端点的 DMA 请求

比特位	名称	复位值	读写属性	功能说明
[12]	DisNyet/PID_Error	0x0	RW	**注意：**此位在不同事务下有不同意义： DisNyet：(仅限批量/中断事务)（RW） 0 = 正常发送 NYET 握手包。 1 = 禁用 NYET 握手包的发送。所有成功接收到的 Rx 数据包都回复 ACK，包括在 FIFO 已满的时候。 PID Error：（仅限同步事务）（RO） 0 = 接收的数据包 PID 无错误。 1 = 接收的数据包 PID 有错误。
[11]	DMAReqMode	0x0	RW	DMA 请求模式： 0 = 选择 DMA 模式 0； 1 = 选择 DMA 模式 1
[10:9]	保留	0x0	RO	---
[8]	IncompRx	0x0	RO	接收包不匹配： 当端点被用于高带宽的同步/中断传输时，如果 Rx FIFO 中的包由于部分数据未被接收而不完整，该位被置 1。当 RxPktRdy 被清零时，此位清零。 0 = 接收包正常； 1 = 接收包数据不完整
[7]	ClrDataTog	0x0	WO	重置数据切换： 写 1 将端点数据切换重置为 DATA0。
[6]	SentStall	0x0	R/W0C	已发送 STALL 握手包： 在 STALL 握手包被发送后此位会被置 1，同时 FIFO 被清空，TxPktRdy 位被清零。 CPU 应该写 0 以清零此位。
[5]	SendStall	0x0	RW	发送 STALL 握手包： 0 = 正常 1 = 保持对接受到的 IN 令牌发送 STALL 握手包
[4]	FlushFIFO	0x0	R/W1O	清除 FIFO： 当该位置 1 时，清除即将从端点 FIFO 发送的数据包。FIFO 的指针复位，RxPktRdy 位清零。该位只在 RxPktRdy 位为 1 时有效，中止当前包装载进 FIFO。 0 = 正常； 1 = 清除即将从端点 FIFO 发送的数据包

比特位	名称	复位值	读写属性	功能说明
[3]	DataError	0x0	RO	数据错误： 如果数据包有 CRC 错误或位填充错误，则在 RxPktRdy 置 1 时此位置 1。当 RxPktRdy 被清零时，此位清零。 0 = 数据无错误；1 = 数据有错误
[2]	OverRun	0x0	R/W0C	过载标志位： 如果 OUT 数据包无法加载到 Rx FIFO 中，此位会被置 1。CPU 应该写 0 清除此位。 0 = 正常；1 = OUT 数据包无法加载到 Rx FIFO 中
[1]	FIFOFull	0x0	RO	FIFO 满标志位： 0 = FIFO 未满；1 = FIFO 已满
[0]	RxPktRdy	0x0	R/W0C	接收完成位： 此位在接收到数据包时被置 1，并产生一个接收中断（若中断已使能）。当从 Rx FIFO 卸载数据包时，CPU 应将该位置 0。， 0 = 接收数据包完成；1 = 接收到数据包

7.5.3.7.2 主机模式

偏移地址：0x0016　　复位值：0x0000

15	14	13	12	11	10	9	8
AutoClear	AutoReq	DMAReqEnab	PID Error	DMAReqMode	保留		IncompRx
rw	rw	rw	ro	rw	ro		ro

7	6	5	4	3	2	1	0
ClrDataTog	RxStall	ReqPkt	FlushFIFO	DataError/NAK Timeout	Error	FIFOFull	RxPktRdy
wo	r/woc	w1o	rw	r/w0c	r/w0c	ro	r/w0c

图表 7-22：接收控制和状态寄存器（RxCSR）主机模式

比特位	名称	复位值	读写属性	功能说明
[15]	AutoClear	0x0	RW	自动清零 RxPktRdy： 该位为 1 时，当有接收包最大尺寸(RxMaxP)大小的数据包从 FIFO 卸载后，RxPktRdy 位会自动置 0。如果卸载的数据小于 RxMaxP，RxPktRdy 需手动置 0。 0 = 正常；1 = RxPktRdy 自动置 0

比特位	名称	复位值	读写属性	功能说明
[14]	AutoReq	0x0	RW	自动请求： 0 = 正常;1 = 清零 RxPktRdy 会自动将 ReqPkt 位置 1。
[13]	DMAReqEnab	0x0	RW	DMA 请求使能： 0 = 禁止此端点的 DMA 请求 1 = 使能此端点的 DMA 请求
[12]	PID_Error	0x0	RO	PID Error： 0 = 接收的数据包 PID 无错误。 1 = 接收的数据包 PID 有错误。 **注意：**仅限仅限同步事务有效，批量和中断事务下此位无效。
[11]	DMAReqMode	0x0	RW	DMA 请求模式： 0 = 选择 DMA 模式 0 1 = 选择 DMA 模式 1
[10:9]	保留	0x0	RO	---
[8]	IncompRx	0x0	RO	接收包不匹配： 当端点被用于高带宽的同步/中断传输时，如果 Rx FIFO 中的包由于部分数据未被接收而不完整，该位被置 1。当 RxPktRdy 被清零时，此位清零。 0 = 正常；1 = 接收包数据不完整
[7]	ClrDataTog	0x0	WO	重置数据切换： 写 1 将端点数据切换重置为 DATA0。
[6]	RxStall	0x0	R/W0C	接收到 STALL 握手包： 当接收到 STALL 握手包时，该位被置 1，并生成中断。CPU 应该写 0 清除此位。 0 = 正常；1 = 接收到 STALL 握手包。
[5]	ReqPkt	0x0	W1O	请求包： CPU 向该位写 1 以请求一个 IN 事务。RxPktRdy 位置 1 时此位自动清除。
[4]	FlushFIFO	0x0	R/W1O	清除 FIFO： 当该位置 1 时，清除即将从端点 FIFO 发送的数据包。FIFO 的指针复位，RxPktRdy 位清零。该位只在 RxPktRdy 位为 1 时有效，中止当前包装载进 FIFO。 0 = 正常 1 = 清除即将从端点 FIFO 发送的数据包
[3]	DataError/ NAK Timeout	0x0	R/W0C	**注意：**此位在不同事务下有不同意义： DataError：(仅限同步事务)（RO）

比特位	名称	复位值	读写属性	功能说明
				如果数据包有 CRC 错误或位填充错误，在 RxPktRdy 置 1 时该位被置 1，并在清除 RxPktRdy 时清除该位。 0 = 正常；1 = 数据错误 Nak Timeout：（仅限批量事务）（R/W0C） 当接收到 NAK 响应后，Rx 端点停止的时间超过 RxInterval 寄存器设置为 NAK 限制的时间，此位将被置 1。CPU 应清零此位，以允许端点继续传输。 0 = 正常；1 = 超过 NAK 后 RxInterval 限制时间
[2]	Error	0x0	R/W0C	错误： 当三次尝试发送数据包而未收到握手数据包时，此位被置 1，并生成中断，清零 RxPktRdy 位，清空 FIFO。CPU 应该写 0 清零此位。 0 = 正常；1 = 未接收到握手包 **注意：**该位仅当端点工作在批量或中断传输下有效。
[1]	FIFOFull	0x0	RO	FIFO 满标志位： 0 = FIFO 未满；1 = FIFO 已满
[0]	RxPktRdy	0x0	R/W0C	接收完成位： 此位在接收到数据包时被置 1，并产生一个接收中断（若中断已使能）。当从 Rx FIFO 卸载数据包时，CPU 应将该位置 0。 0 = 接收数据包完成；1 = 接收到数据包

7.5.3.8 接收计数寄存器（RxCount）

RxCount 是一个 13 位只读寄存器，显示端点 FIFO 中已接收到的数据字节数。该寄存器的值随着 FIFO 内的数据变化，且只有在 RxPktRdy = 1 时有效。

偏移地址：0x0018 复位值：0x0000

15	14	13	12	11	10	9	8
保留			RxCount[12:8]				
ro			ro				

7	6	5	4	3	2	1	0
RxCount[7:0]							
ro							

图表 7-23：接收计数寄存器（RxCount）

比特位	名称	复位值	读写属性	功能说明
[15:13]	保留	0x0	RO	---
[12:0]	RxCount[12:0]	0x0	RO	端点接收到的字节数： 该寄存器值随着 FIFO 的内容改变而改变，且只有在 RxPktRdy 位为 1 时有效。该寄存器值显示的是端点 FIFO 接收到的字节数。

7.5.3.9 发送类型寄存器（TxType）

TxType 是一个 6 位寄存器，低 4 位用于设置目标端点的编号，高 2 位用于设置当前选定 Tx 端点的事务协议。

注意：该寄存器仅限主机模式有效。

偏移地址：0x001A 复位值：0x00

7	6	5	4	3	2	1	0
保留		Protocol[1:0]		Target Endpoint Number[3:0]			
ro		rw		rw			

图表 7-24：发送类型寄存器（TxType）

比特位	名称	复位值	读写属性	功能说明
[7:6]	保留	0x0	RO	---
[5:4]	Protocol[1:0]	0x0	RW	**协议：** CPU 在此设置为 Tx 端点选择所需的协议： 00 = 非法 01 = 同步传输

比特位	名称	复位值	读写属性	功能说明
				10 = 批量传输 11 = 中断传输
[3:0]	Target Endpoint Number[3:0]	0x0	RW	**目标端点编号：** CPU 应将此值设置为设备枚举期间返回给 USB 控制器的 Tx 端点描述符中包含的端点号。

7.5.3.10 发送间隔寄存器（TxInterval）

TxInterval 是一个 8 位寄存器。中断和同步传输时，用于定义当前选定 Tx 端点的轮询间隔。批量传输时，用于设置端点在连续接收 NAK 响应时判定超时的帧/微帧数。每个已配置的 Tx 端点（端点 0 除外）都有一个 TxInterval 寄存器。

注意：该寄存器仅限主机模式有效。

偏移地址：0x001B　　复位值：0x00

7	6	5	4	3	2	1	0
TxIntv[7:0]							
rw							

图表 7-25：发送间隔寄存器（TxInterval）

比特位	名称	复位值	读写属性	功能说明
[7:0]	TxIntv[7:0]	0x0	RW	**中断传输：** 低速或全速时，寄存器有效值 1-255，轮询帧间隔数 = 寄存器值。 高速时，有效值 1-16，轮询帧间隔数 = 2^(寄存器值-1) **同步传输：** 全速或高速时，寄存器有效值 1-16，轮询帧间隔数 = 2^(寄存器值-1) **批量传输：** 全速或高速时，寄存器有效值 2-16，NAK 帧计数 = 2^(寄存器值-1)

7.5.3.11 接收类型寄存器（RxType）

RxType 是一个 6 位寄存器，低 4 位用于设置目标端点的编号，高 2 位用于设置当前选定 Rx 端点的事务协议。

注意：该寄存器仅限主机模式有效。

偏移地址：0x001C 复位值：0x00

7	6	5	4	3	2	1	0
保留		Protocol[1:0]		Target Endpoint Number[3:0]			
ro		rw		rw			

图表 7-26：接收类型寄存器（RxType）

比特位	名称	复位值	读写属性	功能说明
[7:6]	保留	---	RO	---
[5:4]	Protocol[1:0]	0x0	RW	**协议：** CPU 在此设置为 Rx 端点选择所需的协议： 00 = 非法 01 = 同步传输 10 = 批量传输 11 = 中断传输
[3:0]	Target Endpoint Number[3:0]	0x0	RW	**目标端点编号：** CPU 应将此值设置为设备枚举期间返回给 USB 控制器的 Rx 端点描述符中包含的端点号。

7.5.3.12 接收间隔寄存器（RxInterval）

RxInterval 是一个 8 位寄存器。中断和同步传输时，用于定义当前选定 Rx 端点的轮询间隔。批量传输时，用于设置端点在连续接收 NAK 响应时判定超时的帧/微帧数。每个已配置的 Rx 端点（端点 0 除外）都有一个 RxInterval 寄存器。

注意：该寄存器仅限主机模式有效。

偏移地址：0x001D　　　　　　　　　　　　复位值：0x00

7	6	5	4	3	2	1	0
RxIntv[7:0]							
rw							

图表 7-27：接收间隔寄存器（RxInterval）

比特位	名称	复位值	读写属性	功能说明
[7:0]	RxIntv[7:0]	0x0	RW	**中断传输：** 低速或全速时，寄存器有效值 1-255，轮询帧间隔数 = 寄存器值。 高速时，有效值 1-16，轮询帧间隔数 = 2^(寄存器值-1) **同步传输：** 全速或高速时，寄存器有效值 1-16，轮询帧间隔数 = 2^(寄存器值-1) **批量传输：** 全速或高速时，寄存器有效值 2-16，NAK 帧计数 = 2^(寄存器值-1)

动态 FIFO 大小

如果需要，可以将 USB 控制器配置为具有 128、256、512、1K...64K 字节的单个整体 FIFO 大小，然后当 USB 控制器初始化时，可以将其区域分配给不同的端点。(不过，强烈建议用户仅在 USB 控制器用于在不同环境下实际需要不同 FIFO 大小的设备时才使用此功能，因为此功能在额外的门数和固件复杂性方面会增加成本。)

将 FIFO 空间分配给不同的端点需要为每个 Tx 和 Rx 端点指定：

RAM 块内 FIFO 的起始地址

支持的最大数据包大小

是否需要双缓冲

(后两项一起定义了需要分配给 FIFO 的空间量。)

7.5.3.13 发送 FIFO 大小寄存器（TxFIFOsz）

发送 FIFO 大小寄存器控制所选端点发送 FIFO 的大小。

偏移地址：0x0062　　　　复位值：0x00

7	6	5	4	3	2	1	0
保留			DPB	SZ[3:0]			
ro			rw	rw			

图表 7-28：发送 FIFO 大小寄存器（TxFIFOsz）

比特位	名称	复位值	读写属性	功能说明
[7:5]	保留	0x0	RO	---
[4]	DPB	0x0	RW	双数据包缓冲： DPB 位决定是否使用双数据包缓冲。 0 = 单数据包缓冲 1 = 双数据包缓冲
[3:0]	SZ[3:0]	0x0	RW	数据包大小选择： SZ[3:0]定义了所允许的最大包尺寸(byte)。 SZ[3:0] Packet Size(Bytes) 0000: 8 0001: 16 0010: 32 0011: 64 0100: 128 0101: 256 0110: 512 0111: 1024 1000: 2048 1001: 4096 如果 DPB = 0，FIFO 也是这个大小； 如果 DPB = 1，FIFO 将是这个大小的两倍。

7.5.3.14 接收 FIFO 大小寄存器（RxFIFOsz）

接收 FIFO 大小寄存器控制所选端点接收 FIFO 的大小。

偏移地址：0x0063 复位值：0x00

7	6	5	4	3	2	1	0
保留			DPB	SZ[3:0]			
ro			rw	rw			

图表 7-29：接收 FIFO 大小寄存器（RxFIFOsz）

比特位	名称	复位值	读写属性	功能说明
[7:5]	保留	0x0	RO	---
[4]	DPB	0x0	RW	双数据包缓冲： DPB 位决定是否使用双数据包缓冲。 0 = 单数据包缓冲；1 = 双数据包缓冲
[3:0]	SZ[3:0]	0x0	RW	数据包大小选择： SZ[3:0]定义了所允许的最大包尺寸(byte)。 SZ[3:0] Packet Size(Bytes) 0000: 8 0001: 16 0010: 32 0011: 64 0100: 128 0101: 256 0110: 512 0111: 1024 1000: 2048 1001: 4096 如果 DPB = 0，FIFO 也是这个大小； 如果 DPB = 1，FIFO 将是这个大小的两倍。

7.5.3.15 发送 FIFO 地址偏移寄存器（TxFIFOadd）

发送 FIFO 地址偏移寄存器控制着所选端点发送 FIFO 的起始地址。

偏移地址：0x0064 复位值：0x0000

15	14	13	12	11	10	9	8
保留			AD[12:8]				
ro			rw				

7	6	5	4	3	2	1	0
AD[7:0]							
rw							

图表 7-30：发送 FIFO 地址偏移寄存器（TxFIFOadd）

比特位	名称	复位值	读写属性	功能说明
[15:13]	保留	0x0	RO	---
[12:0]	AD[12:0]	0x0	RW	所选端点发送 FIFO 的起始地址： 起始地址以 8 字节为单位进行计算。 AD[12:0] 起始地址 0x0000: 0x0000 0x0001: 0x0008 0x0002: 0x0010 0x0003: 0x0018 .... 0x1FFF: 0xFFF8

7.5.3.16 接收 FIFO 地址偏移寄存器（RxFIFOadd）

接收 FIFO 地址偏移寄存器控制着所选端点接收 FIFO 的起始地址。

偏移地址：0x0066 复位值：0x0000

15	14	13	12	11	10	9	8
保留			AD[12:8]				
ro			rw				

7	6	5	4	3	2	1	0
AD[7:0]							
rw							

图表 7-31：接收 FIFO 地址偏移寄存器（RxFIFOadd）

比特位	名称	复位值	读写属性	功能说明
[15:13]	保留	0x0	RO	---
[12:0]	AD[12:0]	0x0	RW	所选端点发送 FIFO 的起始地址： 起始地址以 8 字节为单位进行计算。 AD[12:0] 起始地址 0x0000: 0x0000 0x0001: 0x0008 0x0002: 0x0010 0x0003: 0x0018 .... 0x1FFF: 0xFFF8

7.5.4 FIFO 寄存器（FIFO Registers）

这个地址范围为每个端点提供 16 个可供 CPU 访问的 FIFO 的地址。写这些地址会将数据写入到相应端点的 TxFIFO 中。读取这些地址将从 RxFIFO 读出对应端点的数据。

FIFO 寄存器地址范围是 0x20~0x5F，FIFO 边界对齐 32 位双字（端点 0 地址 0x20，端点 1 地址 0x24...，端点 15 地址 0x5C）。

7.5.5 DMA 寄存器（DMA Registers）

DMA 控制器有 8 个通道，DMA 控制器有一个中断寄存器，用于指示哪些通道有一个挂起的中断，每个配置的信道有一组三个控制寄存器。

7.5.5.1 DMA 中断寄存器（INTR）

INTR 是一个 8 位寄存器，指示 DMA 中断状态，每一位表示一个 DMA 通道。0 位对应 DMA 信道 0 中断，1 位对应 DMA 信道 1 中断，以此类推。寄存器读后清零。

偏移地址：0x0200 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
DMA7	DMA6	DMA5	DMA4	DMA3	DMA2	DMA1	DMA0
ro	ro	ro	ro	ro	ro	ro	ro

图表 7-32：DMA 中断寄存器（INTR）

比特位	名称	复位值	读写属性	功能说明
[31:8]	保留	0x0	RO	---
[7:0]	DMAx	0x0	RO	DMA 中断： DMAx(x 为 0-7)：产生 DMA 通道 x(x 为 0-7)中断。 读取后清零。 0 = 对应 DMA 通道无中断 1 = 对应 DMA 通道产生中断

7.5.5.2 DMA 控制寄存器（CNTL）

偏移地址：0x0204 + (0x10*x) 复位值：0x00000000

(x 为 DMA 通道号，范围 0-7)

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留					Burst Mode[1:0]		Bus Error
ro					rw		ro

7	6	5	4	3	2	1	0
Endpoint number[3:0]				Interrupt Enable	DMA Mode	Direction	Enable DMA
rw				rw	rw	rw	rw

图表 7-33：DMA 控制寄存器（CNTL）

比特位	名称	复位值	读写属性	功能说明
[31:11]	保留	0x0	RO	---
[10:9]	Burst Mode[1:0]	0x0	RW	Burst 传输模式： 00 = Burst 模式 0，未指定长度 01 = Burst 模式 1，INCR4 或未指定长度 10 = Burst 模式 2，INCR8、INCR4 或未指定长度 11 = Burst 模式 3，INCR16、INCR8、INCR4 或未指定长度
[8]	Bus Error	0x0	RO	总线错误： 0 = 正常；1 = 总线上出现错误
[7:4]	Endpoint number[3:0]	0x0	RW	端点编号： 寄存器值 = 当前 DMA 操作对应的端点编号
[3]	Interrupt Enable	0x0	RW	中断使能： 0 = 不使能 DMA 中断；1 = 使能 DMA 中断
[2]	DMA Mode	0x0	RW	DMA 模式： 0 = DMA 模式 0；1 = DMA 模式 1
[1]	Direction	0x0	RW	传输方向： 0 = DMA 写（Rx 端点）；1 = DMA 读（Tx 端点）
[0]	Enable DMA	0x0	RW	DMA 使能： 0 = 不使能 DMA；1 = 使能 DMA

7.5.5.3 DMA 地址寄存器（ADDR）

偏移地址：0x0208 + (0x10*x) 复位值：0x00000000
(x 为 DMA 通道号，范围 0-7)

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

图表 7-34：DMA 地址寄存器（ADDR）

比特位	名称	复位值	读写属性	功能说明
[31:0]	DMA Addr	0x0	RW	DMA 地址：DMA 信道的 AHB 地址。

7.5.5.4 DMA 计数寄存器（COUNT）

偏移地址：0x020C + (0x10*x) 复位值：0x00000000
(x 为 DMA 通道号，范围 0-7)

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

图表 7-35：DMA 计数寄存器（COUNT）

比特位	名称	复位值	读写属性	功能说明
[31:0]	DMA Count	0x0	RW	DMA 计数：DMA 通道的字节计数器。

DMA 总线错误

如果在 DMA 控制器访问 AHB 上的存储器时发生总线错误，DMA 控制器将立即终止 DMA 传输并用 CNTL 寄存器的 Bus Error 位中断处理器。**注意：**该中断的产生不受 DMA CNTL 寄存器的中断使能位的影响，即使关闭仍会产生中断。

传输数据包

使用内置的 DMA 控制器访问 FIFO 需要对 DMA 控制器和 USBC 端点进行适当的编程，方法不唯一。以下各节详细介绍了用于传输单个数据包和多个数据包的基本操作的标准设置。

单个数据包：Rx 端点，单个数据包的传输通常使用 DMA 模式 0 进行。USBC Rx 端点应编程如下：

IntrRxE 寄存器中的对应端点中断使能位置 1。

对应端点·RxCSR 寄存器的 DMAReqEnab 位置 0。（**注意：**不需要设置 USBC 来为此操作生成 DMA 请求。）

当 USBC 接收到数据包时，会产生对应端点的中断。然后处理器应按如下方式对 DMA 控制器的选定信道进行编程：

ADDR：存储数据包的内存地址

COUNT：包的大小（通过读取 USBC RxCount 寄存器确定）

CNTL：DMA 使能(Bit0) = 1；方向(Bit1) = 0；DMA 模式(bit2) = 0；中断使能(bit3) = 1；以及所需的 Burst Mode(Bit10-9)。

然后 DMA 控制器将请求总线控制权并将数据包传输到存储器。当传输完成时，会生成一个 DMA 中断（DMA_NINT 变低位）。之后处理器应清除 RxCSR 寄存器中的 RxPktRdy 位。

单个数据包：Tx 端点，要使用 DMA 模式 0 执行此操作，USBC Tx 端点应编程如下：

IntrTxE 寄存器中的对应端点中断使能位设置为 1。

对应端点 TxCSR 寄存器的 DMAReqEnab 位设置为 0。（**注意：**不需要设置 USBC 来为此操作生成 DMA 请求）。

当 USBC 中的 FIFO 可用时，USBC 将使用适当的 Tx 端点中断来中断处理器。然后，处理器应按如下方式对 DMA 控制器进行编程：

ADDR：要发送的数据包的内存地址

COUNT：要发送的数据包大小

CNTL：DMA 使能(Bit0) = 1；方向(Bit1) = 0；DMA 模式(bit2) = 0；中断使能(bit3) = 1；以及所需的 Burst Mode(Bit10-9)。

然后 DMA 控制器将请求总线控制权并将数据包传输到 USBC FIFO。当传输完成时，会产生一个 DMA 中断。之后处理器应把 USBC TxCSR 寄存器中的 TxPktRdy 位置 1。

7.6 功能描述

7.6.1 复位（Reset）

当在总线上检测到复位条件，如果 USB 控制和状态寄存器（UCSR）中的 HS Enable 位被置 1，则 USBC 将尝试协商进入高速模式，然后设备将执行以下操作：

将功能地址寄存器（FAddr）清零
将端点索引寄存器（Index）清零
刷新所有端点的 FIFO
复位所有控制和状态寄存器
开启所有端点中断使能
产生一个复位中断

7.6.2 软连接（Soft Connect）

USBC 允许通过软件控制与 USB 总线进行连接。与 USBC 一起使用的 PHY 可以通过设置或清除 UCSR 寄存器的软连接位(bit6)来实现普通模式和非驱动模式的切换。当该位置 1 时，PHY 处于正常模式，USB 总线的数据线已使能。同时，USBC 处于通电状态，不会响应除了 USB 复位的任何 USB 信号。当该位置 0 时，PHY 进入无驱动模式，数据线处于高阻态，在 USB 主机看来，USBC 已经断开。

7.6.3 帧起始包（SOF）

当 USBC 处于全速模式（默认）时，USBC 应每 1 毫秒从主机接收一次帧起始包。

如果 SOF 中断使能已打开，当收到 SOF 包时，将产生 SOF 中断。包中包含的帧数被写入帧数寄存器(Frame)。如果 1.00385ms 没收到 SOF 包，会假设该包已丢失。USBC 将继续每 1 毫秒产生一次 SOF 中断，但帧数寄存器不会更新。

7.6.4 挂起和恢复（Suspend/Resume）

当 USB 总线上 3ms 没有活动时，USBC 将进入挂起模式并产生挂起中断。当检测到恢复信号时，USBC 时钟会重新启动，并且 USBC 将退出挂起模式，同时产生一个恢复中断。

USBC 也支持远程唤醒。CPU 可以设置 UCSR 寄存器的 Resume 位，使 USBC 脱离挂起模式并驱动总线产生恢复信号。CPU 在大约 10ms 后清除这个恢复信号，这种方式不会产生恢复中断。

7.6.5 IN 事务处理（In-Transaction Handling）

IN 事务通过 USBC 的 Tx FIFO 进行处理。

数据包的最大尺寸是由端点的 TxMaxP 寄存器决定，且该值可配。任何端点的最大数据包尺寸都不能超过 FIFO 的大小。当 FIFO 中有数据时，不能对 TxMaxP 寄存器进行写操作，否则会产生不可预知的错误。

7.6.6 发送包缓存（Transmit Packet Buffering）

在 FIFO 中只能缓存一个数据包。当要发送的数据包加载到发送 FIFO 中后，TxCSR 的 TxPktRdy 位需置 1。如果 TxCSR 中的 AutoSet 位为 1，当最大包大小的数据加载到 FIFO 中后，TxPktRdy 位会自动置 1。如果小于最大包数的数据加载到 FIFO 中，TxPktRdy 需要手动置 1。

TxPktRdy 位不管是自动还是手动置 1，TxCSR 中的 FIFONotEmpty 位都会置 1，然后数据准备发送。当数据包成功发送后，TxPktRdy 和 FIFONotEmpty 位会被清除，且相应的端点会产生发送中断。下一个数据包可以开始加载到 FIFO 中。

7.6.7 OUT 事务处理（Out-Transaction Handling）

OUT 事务通过 USBC 的 Rx FIFO 进行处理。数据包的最大尺寸是由端点的 RxMaxP 寄存器决定，且该值可配。任何端点的最大数据包尺寸都不能超过 FIFO 的大小。

7.6.8 接收包缓存（Receive Packet Buffering）

在 FIFO 中只能缓存一个数据包。当数据包被接收到 FIFO 中后，RxCSR 的 RxPktRdy 位和 FIFOFull 位被置 1，相应的端点产生接收中断（如果中断已使能），表示数据可以从 FIFO 中读出。

当数据包从 FIFO 中读出后，RxPktRdy 位需要清除，使得下一个数据能准确接收。如果 RxCSR 的 AutoClear 位为 1，且最大包尺寸的数据从 FIFO 中读出后，RxPktRdy 位会被自动清除，FIFOFull 位也会被清除。对于小于最大包尺寸的数据，RxPktRdy 位需要手动清除。

7.6.9 外部参考时钟检测器（External clock reference detector）

释放 RSTN 后，USBC 将在 20 毫秒内自动检测 CLK12_REF。如果有来自 CLK12_REF 的时钟，USB 将使用它并关闭内部 RC 和时钟恢复单元，否则 USB 将使用内部 RC 输出作为参考时钟，并从 DP/DM 数据帧恢复 12MHz 时钟。

7.7 传输操作

USBC 支持三种基本传输类型，见下文。

7.7.1 控制传输（Control Transfer）

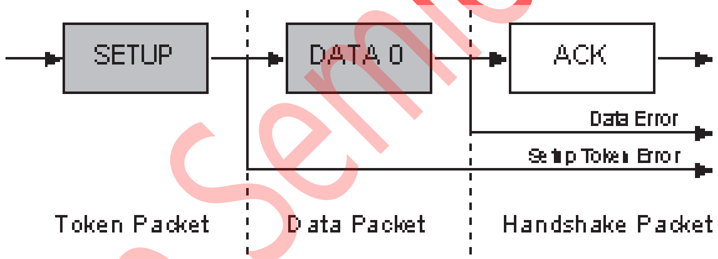
控制传输通常用于命令和状态操作。控制传输对 USB 设备建立至关重要，所有枚举功能都是使用控制传输来执行的。在 USBC 中，端点 0 支持这种类型的传输。

控制传输最多可以有三个阶段：

1. 建立（Setup）阶段用来发送请求。此阶段包括三个包。首先发送包含地址和端点编号的建立（SETUP）令牌包。

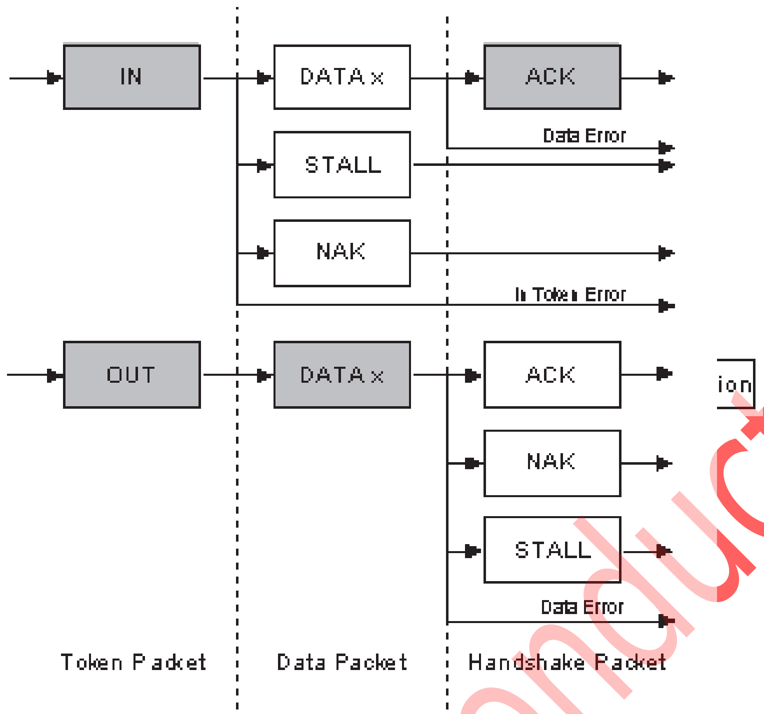
2. 接下来发送数据包，数据包的 PID 类型始终为 Data0，并包含详细说明请求类型的 setup 数据。稍后将详细介绍 setup 数据。

3. 最后一个包是握手包，用于确认接收成功或指示错误。如果成功地接收到 setup 数据（有效的 CRC 和 PID 等）回应 ACK，否则忽略不发送握手包。USBC 不能对 setup 回应 NAK 和 STALL。



图表 7-36：建立阶段格式

可选的数据阶段由一个或多个输入或输出传输组成。设置请求指定此阶段要传输的数据量。如果超过最大数据包大小，数据将以多个传输方式发送，每个传输都是最大数据包长度，最后一个数据包除外。根据数据传输的方向，数据阶段有两种不同的场景（见图 1-3）。

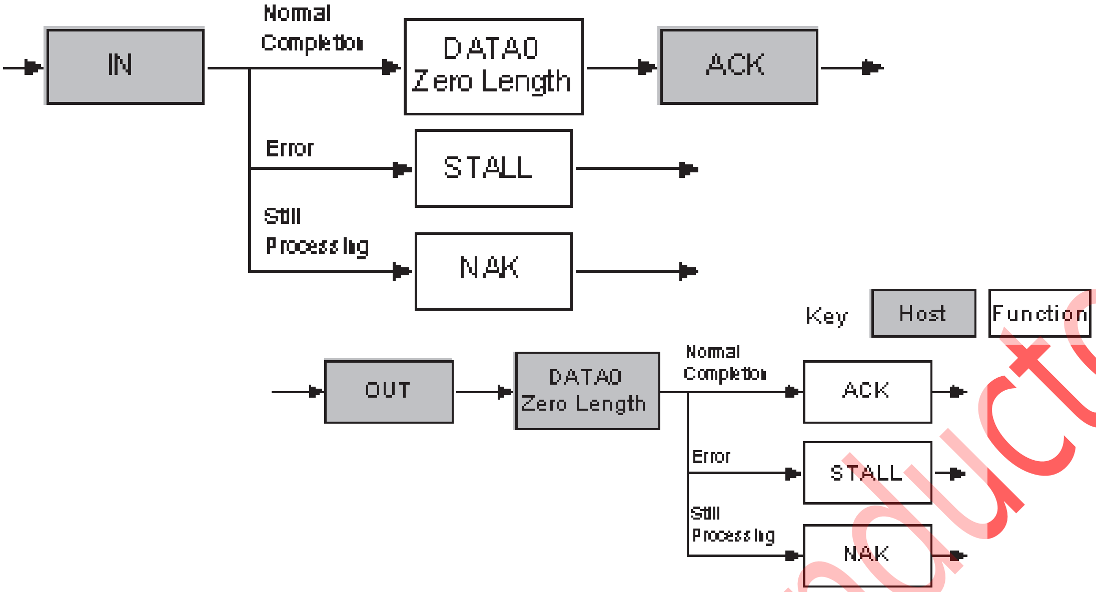


图表 7-37：数据阶段格式

IN：当主机准备好接收控制数据时会发出 IN 令牌。如果设备接收到带有错误的 IN 令牌，例如 PID 与取反 PID 位不匹配，则忽略该数据包。如果令牌接收正确，则设备可以使用包含要发送的控制数据的数据包，指示端点发生错误的 STALL 握手包，指示端点在忙暂时无法发送数据的 NAK 握手包，其中之一对主机进行应答。

OUT：当主机需要向设备发送一个控制数据包时会发出 OUT 令牌，后跟一个包含控制数据的数据包作为荷载。如果 OUT 令牌或数据包的任何部分损坏，设备将忽略该数据包。如果设备的端点缓冲区为空，并且已将数据保存到端点缓冲区中，则回复 ACK 握手包，通知主机它已成功接收数据。如果由于前一个数据包的处理，端点缓冲区不为空，设备回复 NAK 握手包。如果端点自身有错误并且其停止(halt)位已置 1，则回复 STALL 握手包。

状态(Status)阶段报告整个请求的状态，并且每次传输方向改变时，都会汇报一次。状态报告始终由设备执行。



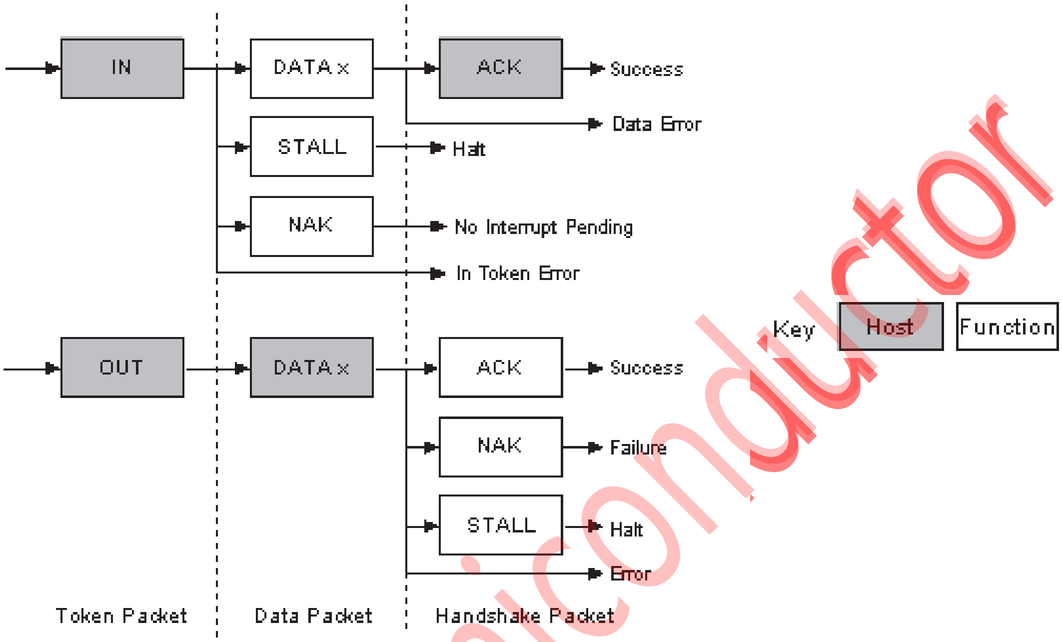
图表 7-38：状态阶段格式

IN：如果主机在数据阶段发送 IN 令牌来接收数据，则必须在接收数据后告知从机数据已成功接收。这种握手通过主机发送一个 OUT 令牌，后跟一个零长度的数据包来实现。设备可以在发送握手包时报告其状态。ACK 表示从机已完成命令，现在可以接受另一个命令。STALL 表示从机在处理此命令期间发生错误。NAK 表示从机在忙，需要主机稍后重复状态阶段。

OUT：如果主机在数据阶段发送 OUT 令牌来传输数据，从机可以通过发送零长度的数据包响应 IN 令牌来确认数据的成功接收。如果从机发生错误发送 STALL 来响应，如果从机在忙需要主机稍后重复状态阶段用 NAK 来响应。

7.7.2 中断传输（Interrupt Transfer）

中断传输过程从机排成队列，由主机轮询 USB 从机请求数据。下图显示了中断 IN 和中断 OUT 事务的格式。在 USBC 中，端点 1 支持这种类型的传输。



图表 7-39：中断传输格式

IN：主机将定期轮询中断传输的端点。轮询速率由端点描述符指定，稍后介绍。每次轮询主机都会发送 IN 令牌。

如果中断队列的从机被轮询到，则当从机接收到 IN 令牌时，将发送包含与中断相关数据的数据包。主机成功接收后回复 ACK。但是如果数据已损坏，主机不回复任何握手包。

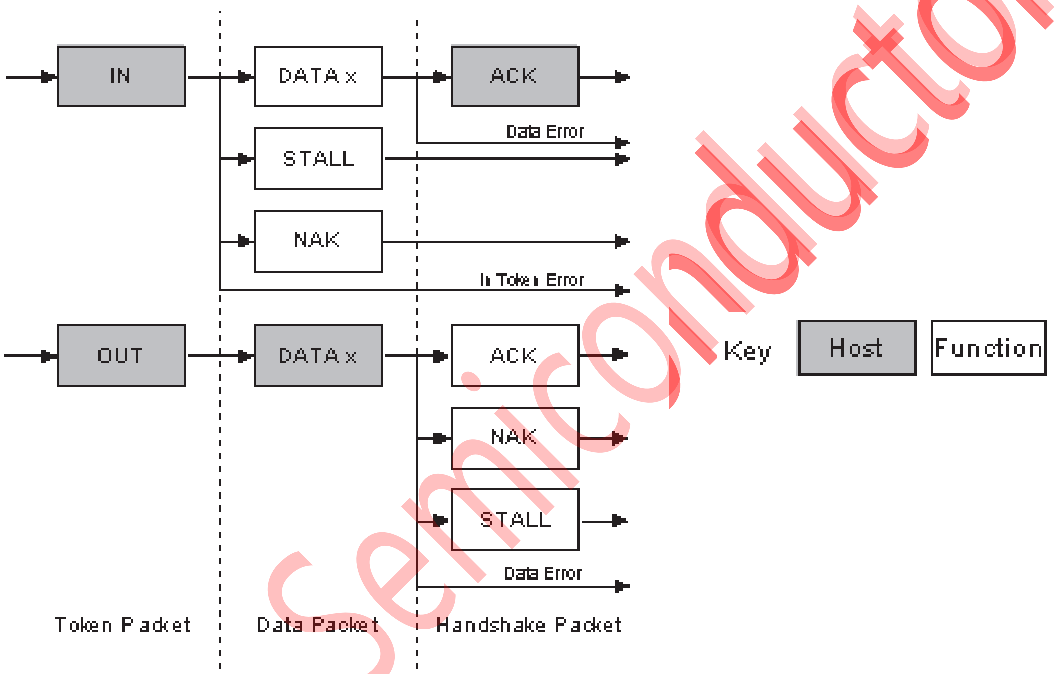
如果主机使用 IN 令牌轮询中断端点时，从机未满足中断条件，则从机回复 NAK 告知主机。如果端点上发生错误，回复 STALL。

OUT：当主机想要向从机发送中断数据时，会发出一个 OUT 令牌，后跟一个包含中断数据的数据包。设备将根据不同的条件向主机发出响应。

7.7.3 批量传输（Bulk Transfer）

批量传输可用于传输大量数据，例如发送到打印机的打印作业。批量传输以 CRC16 域的形式对数据内容和错误检测/重新传输机制进行错误校验，以确保数据收发无误。在 USBC 中，端点 1 支持这种类型的传输。

全速批量传输的最大数据包大小为 64 字节。数据长度低于最大数据包大小不需要用零填充。当批量传输已传输了请求的数据量、传输了小于最大端点大小的数据包、传输了零长度的数据包三者其中之一时，认为该批量传输已完成。下图展示了批量 IN 和批量 OUT 事务的格式。



图表 7-40：中断传输格式

IN：当主机准备好接收批量传输数据时，发出 IN 令牌。如果从机接收到的 IN 令牌有错误，忽略该令牌包。如果令牌接收正确，则从机可以使用包含要发送的数据的数据包、表示端点出错的 STALL 握手包、表示端点正忙且暂时无法发送数据的 NAK 握手包，三者其中之一进行应答。

OUT：当主机想向从机发送批量传输数据包时，发出 OUT 令牌，后跟一个包含批量数据的数据包。如果 OUT 令牌或数据包的任何部分损坏，从机将忽略该数据包。如果设备的端点缓冲区为空，并且已将数据保存到端点缓冲区中，则回复 ACK 握手包，通知主机它已成功接收数据。如果由于前一个数据包的处理，端点缓冲区不为空，设备回复 NAK 握手包。如果端点自身有错误并且其停止(halt)位已置 1，则回复 STALL 握手包。

7.7.4 控制事务（Control Transactions）

端点 0 是 USBC 的控制端点。软件应通过端点 0 收发处理所有的标准设备请求。这些设备请求的协议涉及每一种传输(Transfer)，以及传输中的数目不一的各类事务(Transaction)。

USB 从机的标准设备请求分为三类：零数据请求，写请求和读请求。

零数据请求：所有的信息已包含在命令中。
写请求：命令后包含额外的数据。
读请求：从机要求发送数据给主机。

7.7.4.1 零数据请求

零数据请求的所有信息都包含在 8 字节的命令中，不需要传输额外的数据。例如：SET_ADDRESS，SET_CONFIGURATION，SET_INTERFACE 和 SET_FEATURE。

CSR0 寄存器是 USBC 控制传输状态寄存器。软件应根据不同的条件设置相应的位来处理控制传输。当 USBC 接收到一个端点 0 中断时，请求事件开始。RxPktRdy 位被置 1，从端点 0 FIFO 读取 8 字节的命令进行解码，并采取相应的操作。例如：如果命令是 SET_ADDRESS，则命令中包含 7 位地址值，随后将该值写入 FADDR 寄存器。

CSR0 寄存器应将 ServicedRxPktRdy 位置 1(表示命令已从 FIFO 中读取)，以及将 DataEnd 位置 1(表示此请求不需要进一步的数据)。

当主机进入请求状态阶段时，生成第二个端点 0 中断指示请求已完成。此中断只用于确认请求成功完成，软件无需进一步操作。

如果该命令无法识别，或者由于其他原因不能执行，CSR0 寄存器应将 ServicedRxPktRdy 位和 SendStall 位为 1。当主机进入请求状态阶段时，USBC 将发送 STALL 握手包，告诉主机请求没有被执行，并产生第二个端点 0 中断，将 SentStall 位置 1。

如果主机在 DataEnd 置 1 后发送了更多的数据，USBC 将发送 STALL 握手包，产生端点 0 中断，将 SentStall 位置 1。

7.7.4.2 写请求

写请求在主机发送 8 字节命令后，发送一个(或多个)数据包。例如：SET_DESCRIPTOR。CSR0 寄存器是 USBC 控制传输状态寄存器。软件应根据不同的条件设置相应的位来处理控制传输。

当 USBC 接收到一个端点 0 中断时，请求事件开始。RxPktRdy 位被置 1，从端点 0 FIFO 读取 8 字节的命令进行解码。将 CSR0 寄存器的 ServicedRxPktRdy 位置 1（命令已从 FIFO 中读取），但是不应设置 DataEnd 位(表示还需更多的数据)。

当收到第二个端点 0 中断时，应读取 CSR0 寄存器，检查端点状态。RxPktRdy 位为 1 表示数据包已收到，读取 Count0 寄存器确认这个数据包的大小，然后从端点 0 FIFO 中读取数据包。如果与请求数据的长度(在命令中由 wLength 字段表示)大于端点 0 的最大数据包大小，将继续发送数据包。在这种情况下，应该将 CSR0 的 ServicedRxPktRdy 位置 1，但是不应设置 DataEnd 位。

当所有期望的数据包都已收到时，CSR0 寄存器的 ServicedRxPktRdy 位和 DataEnd 位应置 1(表示不需要更多数据)。当主机进入请求状态阶段时，生成另一个端点 0 中断，用以指示请求已完成。此中断只用于确认请求成功完成，软件无需进一步操作。

如果该命令无法识别，或者由于其他原因不能执行，CSR0 寄存器应将 ServicedRxPktRdy 位和 SendStall 位为 1。当主机进入请求状态阶段时，USBC 将发送 STALL 握手包，告诉主机请求没有被执行，并产生第二个端点 0 中断，将 SentStall 位置 1。

如果主机在 DataEnd 置 1 后发送了更多的数据，USBC 将发送 STALL 握手包，产生端点 0 中断，将 SentStall 位置 1。

7.7.4.3 读请求

读请求在主机发送 8 字节命令后，从机向主机发送一个(或多个)数据包。例如：GET_CONFIGURATION，GET_STATUS，GET_INTERFACE 和 GET_DESCRIPTOR。

当 USBC 接收到一个端点 0 中断时，请求事件开始。RxPktRdy 位被置 1，从端点 0 FIFO 读取 8 字节的命令进行解码。将 CSR0 寄存器的 ServicedRxPktRdy 位置 1（命令已从 FIFO 中读取）。

从机要发送到主机的数据应写入端点 0 FIFO。如果要发送的数据的长度大于端点 0 的最大数据包大小，应只将最大数据包大小的数据写入 FIFO。然后将 CSR0 寄存器的 TxPktRdy 位置 1（指示 FIFO 中有要发送的数据包）。当数据包发送到主机时，将产生另一个端点 0 中断，下一个数据包可以写入 FIFO。当最后一个数据包被写入 FIFO 时，应该将 CSR0 寄存器 TxPktRdy 位和 DataEnd 位置 1（表示该数据包之后没有更多的数据）。

当主机进入请求状态阶段时，生成另一个端点 0 中断，用以指示请求已完成。此中断只用于确认请求成功完成，软件无需进一步操作。

如果该命令无法识别，或者由于其他原因不能执行，CSR0 寄存器应将 ServicedRxPktRdy 位和 SendStall 位为 1。当主机进入请求状态阶段时，USBC 将发送 STALL 握手包，告诉主机请求没有被执行，并产生第二个端点 0 中断，将 SentStall 位置 1。

如果主机在 DataEnd 置 1 后发送了更多的数据，USBC 将发送 STALL 握手包，产生端点 0 中断，将 SentStall 位置 1。

7.7.4.4 端点 0 的状态

端点 0 控制有三种模式:空闲(IDLE)、发送(TX)和接收(RX)，对应控制传输的不同阶段下端点 0 进入的状态。开机或重置时的默认模式应为空闲。

当端点 0 处于空闲状态时，RxPktRdy 位被置 1 表示有新的从机请求。从 FIFO 中读取从机请求，USBC 对描述符进行解码，确认是否存在数据阶段。如果存在，确认控制传输的数据阶段的传输方向（以便设置 FIFO 方向）。

根据数据阶段的传输方向，端点 0 进入 TX 状态或 RX 状态。如果没有数据阶段，端点 0 将保持空闲状态，等待接受下一个从机请求。

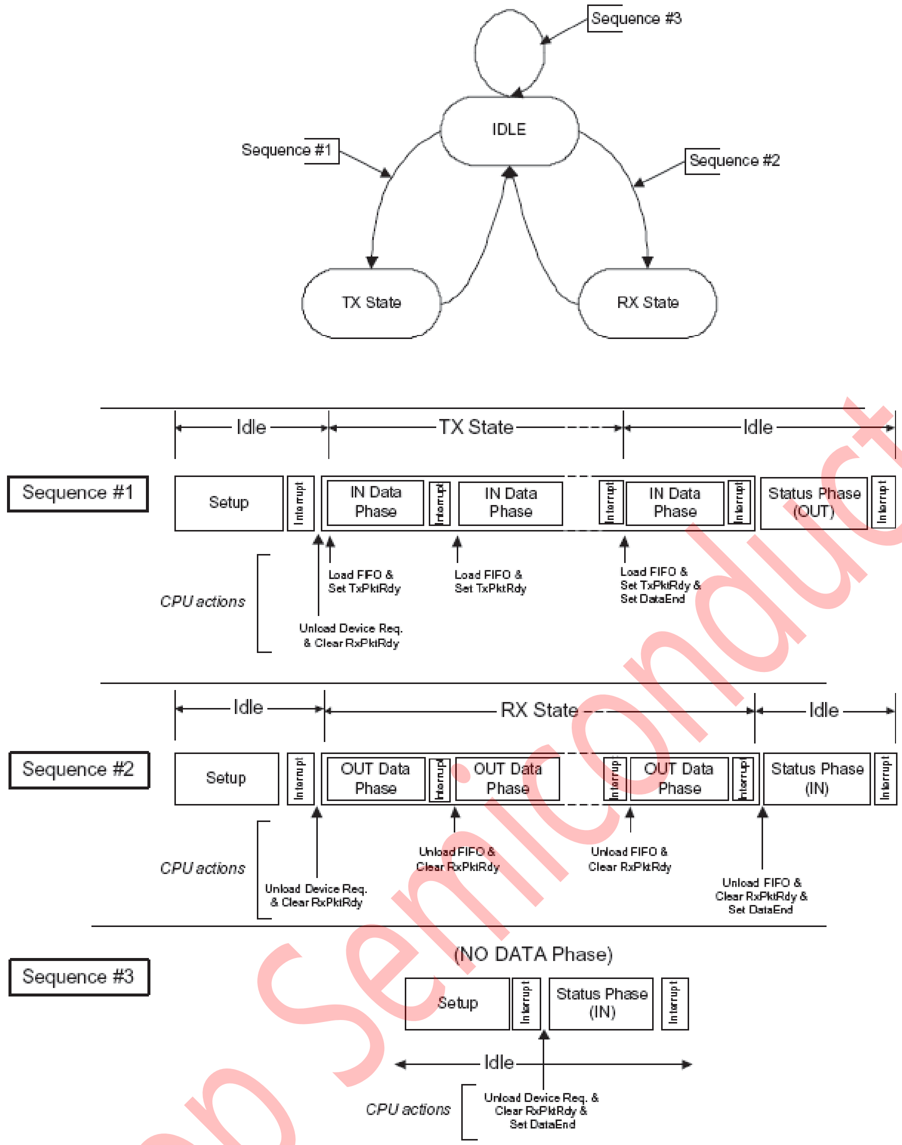
CPU 需要在可能的传输的不同阶段（例如加载 FIFO、设置 TxPktRdy），采取响应的操作（如下面的图 1-7 所示）。USBC 独立于 CPU 根据数据阶段的方向来改变 FIFO 的方向。

7.7.4.5 端点 0 的中断服务程序（ISR）

端点 0 中断将在以下情况产生:

当 USBC 收到一个有效的令牌并将数据写入到 FIFO 中，RxPktRdy 位(CSR0.bit0)被置 1。
当 FIFO 中的数据包已成功发送到主机，清零 TxPktRdy 位(CSR0.bit1)。
当控制事务由于违反协议而终止时，SentStall 位(CSR0.bit2)被置 1。
当控制传输在 DataEnd 位(CSR0.bit3)置 1 之前结束，SetupEnd 位(CSR0.bit4)被置 1。

当进入端点 0 中断服务程序时，软件必须首先检查控制传输是否因为 STALL 或过早结束而中断。如果控制传输因为 STALL 中断，SentStall 位会被置 1。如果控制传输由于过早结束而中断，SetupEnd 位会被置 1。不管哪种情况，软件应该停止处理当前的控制传输并回到空闲状态。



图表 7-41：端点 0 传输场景

如果软件确定中断不是由总线非法状态产生，下一步动作取决于端点 0 状态。如果端点 0 处于空闲状态，中断的唯一原因是从 USB 总线接收数据。中断服务程序必须检查 RxPktRdy 位。如果该位为 1，说明收到一个 SETUP 包，下一步须从 FIFO 读出数据，译码来决定下一步的操作。

根据 SETUP 包中包含的命令，端点 0 将进入下面三种状态之一：

如果该命令是单个数据包事务（SET_ADDRESS，SET_INTERFACE 等)，没有任何数据阶段，端点将保持在空闲状态。

如果该命令有一个 OUT 数据阶段(SET_DESCRIPTOR 等)，端点将进入 RX 状态。

如果该命令有一个 IN 数据阶段(GET_DESCRIPTOR 等)，端点将进入 TX 状态。

如果端点处于 TX 状态，中断表明 USB 控制器已接收到 IN 令牌且 FIFO 中的数据已经发送。软件必须对此

作出响应，如果还有数据需要传送，将更多数据加载到 FIFO 中；如果数据已全部传送完毕，将 DataEnd 位置 1 表明数据阶段完成，同时端点 0 应返回到空闲状态，等待下一个控制事务。

如果端点处于 RX 状态，中断表明接收到数据包。软件必须从 FIFO 中读取接收到的数据并作出响应，确定数据是否全部接受完毕。如果接受完毕，将 DataEnd 位置 1 表明数据阶段完成，同时端点 0 应返回到空闲状态；如果还需要更多数据，软件应将 ServicedRxPktRdy 位(CSR0.bit6)置 1，表明 FIFO 中的数据已被读取，并让端点处于 RX 状态。

7.7.5 批量输入事务（Bulk IN Transactions）

批量输入事务用于从机传输大规模数据到主机。

TxCSR 寄存器是 USBC 发送控制状态寄存器。软件应该根据不同的条件设置相应的位来处理到主机的传输。

当 AutoSet 位使能时，如果有 TxMaxP 字节的数据被加载到 FIFO 中，TxPktRdy 位将被自动置 1。

7.7.5.1 建立

首先，端点的 TxMaxP 寄存器必须配置为数据包大小的最大值。这个值应该与标准端点描述符的 wMaxPacketSize 字段一致。如果此端点需要中断，应将 IntrTxE 寄存器中相关的中断使能位设置为 1。TxCSR 寄存器应设置如下:

Bit15：如果将 AutoSet 置 1，则启用 AutoSet 功能，否则 TxPktRdy 位需要手动设置。

Bit13：Mode 设置为 1，确保 FIFO 作为 Tx 使用。

Bit11：FrcDataTog 设置为 0，允许正常的数据切换操作。

配置端点的时候，首先应将 TxCSR 的 ClrDataTog 位该置 1，确保数据切换状态正确；如果 FIFO 中还有数据，应把 FlushFIFO 位置 1 以清空 FIFO。

7.7.5.2 运行

批量输入数据，需要将数据包加载到 FIFO 中，并将 TxCSR 寄存器 TxPktRdy 位置 1。数据包发送后，USBC 会自动清除 TxPktRdy 位，并产生中断，以便下一个数据包可以加载到 FIFO 中。

数据包大小不能超过 TxMaxP 寄存器低 11 位指定的大小，此数值定义为单次数据包的有效载荷。根据 USB 协议规定全速传输的有效载荷可以配置为 8、16、32、64 字节。如果要传输的数据量超过此数量，则需要将其作为多个数据包发送，这些数据包的大小都应为有效载荷的值，除了最后一个可能填充不满的数据包。

主机通过比较实际传输的数据量与预期的数据量来确定传输的所有数据是否都已发送。

另一种方案是，如果主机接收到小于有效载荷值的包，则推断所有数据已经被发送。这种情况下，如果数据的总大小是该有效载荷的整倍数，则在发送所有数据之后，从机需要发送空数据包。此操作可以通过在接收

到下一个中断时将 TxPktRdy 置 1 来完成，无需将任何数据加载到 FIFO 中。

7.7.5.3 错误处理

软件可以通过将 TxCSR 的 SendStall 位置 1 来关闭批量输入管道，如果后续 USBC 接收到 IN 令牌，会向主机发送一个 STALL 握手包，随后 SentStall 位被置 1 并产生中断。当软件接收到来自 SentStall 的中断后清除 SentStall 位，但是保留 SendStall 位，直到准备好重新启用批量输入管道。重启管道时，应将 ClrDataTog 位置 1 重置数据切换序列。

7.7.6 批量输出事务（Bulk OUT Transactions）

批量输入事务用于主机传输大规模数据到从机。

RxCSR 寄存器是 USBC 的接收控制状态寄存器。软件应该根据不同的条件设置相应的位来处理到主机的传输。

当 AutoSet 位使能时，如果有 RxMaxP 字节的数据从 FIFO 中卸载，RxPktRdy 位将被自动清零。

7.7.6.1 建立

首先，端点的 RxMaxP 寄存器必须配置为数据包大小的最大值。这个值应该与标准端点描述符的 wMaxPacketSize 字段一致。如果此端点需要中断，应将 IntrRxE 寄存器中相关的中断使能位设置为 1。RxCSR 寄存器应设置如下:

Bit15：如果将 AutoClear 置 1，则启用 AutoClear 功能，否则 RxPktRdy 位需要手动清零。

Bit12：DisNyet 模式置 0，允许正常的 PING 流控制。

配置端点的时候，首先应将 RxCSR 的 ClrDataTog 位该置 1，确保数据切换状态正确；如果 FIFO 中还有数据，应把 FlushFIFO 位置 1 以清空 FIFO。

7.7.6.2 运行

当批量输出的端点接收到一个数据包，RxPktRdy 位被置 1，并产生中断。软件应该读取端点的 RxCount 寄存器来确定数据包的大小，然后从 FIFO 中读取数据包，并将 RxPktRdy 位自动或手动清除。

发送的数据包大小不能超过 RxMaxP 寄存器低 11 位指定的大小，此数值定义为单次数据包的有效载荷。根据 USB 协议规定全速传输的有效载荷可以配置为 8、16、32、64 字节。如果要接收的数据量超过此数量，则需要将其作为多个数据包接收，这些数据包的大小都应为有效载荷的值，除了最后一个可能填充不满的数据包。

软件可以用特定程序来确定包的总大小，从而推算出何时接收最后一个包。另一种方案是，当接收到的包小于有效载荷或接收到空数据包时，表示当前的传输已完成。

7.7.6.3 错误处理

软件可以通过将 RxCSR 的 SendStall 位置 1 来关闭批量输出管道，如果后续 USBC 接收到下一个包，会向主机发送一个 STALL 握手包，随后 SentStall 位被置 1 并产生中断。当软件接收到来自 SentStall 的中断后清除 SentStall 位，但是保留 SendStall 位，直到准备好重新启用批量输出管道。重启管道时，应将 ClrDataTog 位置 1 重置数据切换序列。

7.7.7 中断事务（Interrupt Transactions）

中断输入事务与批量输入事务使用相同的协议，可以用同样方式使用。相似的，中断输出事务与批量输出事务使用几乎相同的协议，可以以同样方式使用。

USB 协议定义了中断事务需要支持数据切换位的连续切换，批量传输则不需要。这一特点通过设置 TxCSR 的 FrcDataTog 来实现。当此位置 1，USBC 在成功发送数据包后即切换数据位，不再关心是否从主机接收到 ACK。

中断端点不支持 PING 流控制。这意味着 USBC 永远不应该用 NYET 握手包来响应，只能使用 ACK/NAK/STALL。为了确保这一点，RxCSR 寄存器的 DisNyet 位应置 1，以禁用 NYET 握手包。

8 内部闪存模块（EFLASH）

8.1 概述

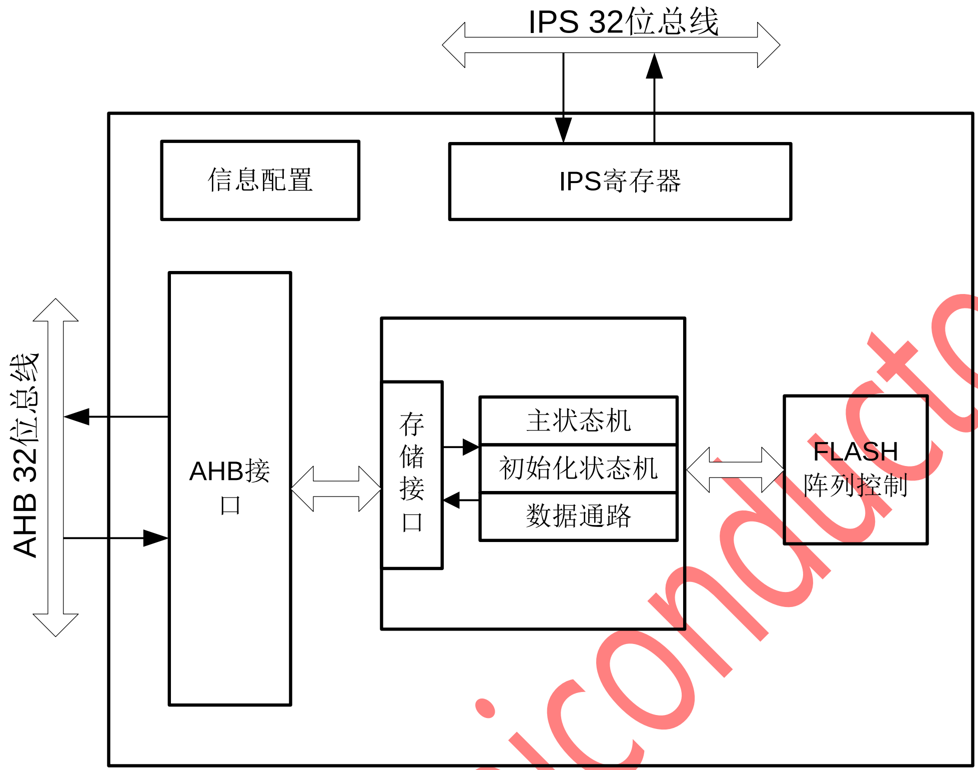
FLASH（EFM）是一款 CMOS 工艺的、按页（512 字节）擦除、按字（38 位）编程的片内闪存存储器，38 位的字由 32 位数据和 6 位 ECC 校验位组成。存储空间划分为两个存储区，一个是主存储区，另一个是信息存储区。主存储区由 131072 个 38 位的字组成（512K 字节）。信息存储区由 128 个 38 位的字组成（512 字节）。信息存储区可以用来存放芯片的一些特有信息。页擦除操作可以擦除一页内所有的字节。无论是主存储区还是信息存储区的页都是由两个相邻的行组成。FLASH 的擦除和编程的供电都是由 VDIO（1.5V~1.98V）来提供的。

8.2 特性

EFM 的特性如下：

存储器包含 512KB（主存储区）和 512B（信息存储区）
按字节（8 位）、半字（16 位）和字（32 位）读取
编程和擦除自动化操作
带 ECC 校验，并产生 ECC 错误标志
可配置产生中断当命令完成后
数据保存时间：10 年
0.99~1.21 伏/1.5~1.98 伏双电源供电

8.3 框图



图表 8-1：EFM 框图

如图 10-1 所示，EFM 有两个接口，一个是 32 位的 IPS 总线接口，一个是 32 位的 AHB 总线接口。IPS 总线接口用于访问 EFM 的寄存器，通过寄存器可以配置访问 FLASH 的时序以及对 FLASH 进行编程和擦除。AHB 总线用于访问 FLASH 的存储内容，无论是主存储区的指令或数据还是信息存储区的信息，都是通过 AHB 总线接口进行传输的。

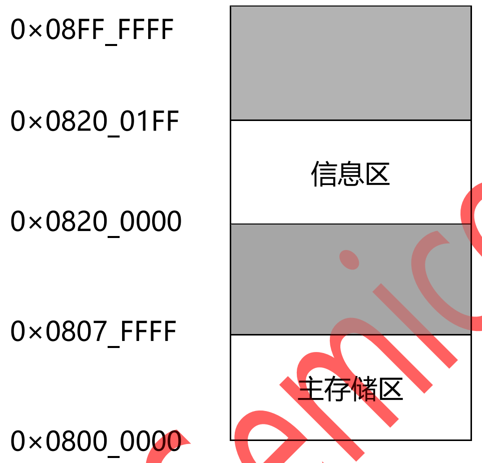
8.4 工作模式

FLASH 用户模式——在这种模式下，EFM 用于非易失的程序和数据存储器。FLASH 编程以及擦除操作由用户软件控制。

8.5 内存映射和寄存器

8.5.1 内存映射

FLASH 的地址在 COS 模式下的地址 0×0040_0000 上开始被映射的。下图是 FLASH 主存储区和信息区的地址映像。



图表 8-2：EFM 内存地址映像

EFM 模块也包括一套控制和状态寄存器。下表展示了这些寄存器的内存映射。EFM 寄存器基地址为 0x63f80000。

表格 8-1：寄存器内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	EFM 配置寄存器(EFCR)		S/U
0x0004	EFM 访问保护寄存器(EFAPR)		S
0x0008	EFM 状态寄存器(EFSTAT)		S
0x000c	EFM 中断屏蔽寄存器(EFINTM)		S
0x0010	EFM 命令配置寄存器(EFCMD)		S
0x0014	保留		S
0x0018	EFM 编程擦除单位时间配置寄存器(ETIMBASE)		S
0x001c	EFM 编程擦除时序配置寄存器(ETIMCFG)		S
0x0020	EFM 编程擦除写等待超时寄存器(EFPETIMER)		S
0x0024	EFM 智能写时序配置寄存器(SMWOP0)		S
0x0028-0x00FF	保留		S

注意：

S = 只限于超级用户使用。

普通用户对只限于超级用户使用的地址位置访问无效，且会产生一个周期终止迁移错误。

对保留地址位置不能进行写，否则会导致不确定的失败。

8.5.2 寄存器描述

8.5.2.1 EFM 配置寄存器（EFCR）

偏移地址：0x0000~0x0003 复位值：0x00000007

31	30	29	28	27	26	25	24
ECC_DIS	INFO_SWITCH	AHB_RES_CFG		保留	ULTRA_HIGH_SPEED	保留	保留
rw	rw	rw	rw	ro	rw	ro	

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留					RWSC		
ro					rw		

图表 8-3：EFM 配置寄存器（EFCR）

比特位	名称	复位值	读写属性	功能说明
[31]	ECC_DIS	0x0	RW	关闭 ECC 校验配置 配置在对 FLASH 访问时是否需要进行 ECC 校验。 0 = 使能 FLASH 数据的 ECC 校验功能 1 = 关闭 FLASH 数据的 ECC 校验功能
[30]	INFO_SWITCH	0x0	RW	在访问 FLASH 的信息区时，来切换访问的是正常用户信息区还是测试信息区。正常模式必须是 0。 0 = 访问信息区空间会切换到测试信息区 1 = 访问信息区空间时正常访问用户信息区
[29:28]	AHB_RES_CFG[1:0]	0x0	RW	当编程或擦擦 FLASH 时，发生读 FLASH 操作后配置 AHB 总线的响应。 00 = 拉住 AHB 总线即停住 CPU 访问，直到编程或擦除完成后再响应读操作 01 = 在发生写时读的情况，AHB 总线返回 OK 应答，产生中断标志位 1x = 在发生写时读的情况，AHB 总线返回 ERROR 应答
[27]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[26]	ULTRA_HIGH_SPEED	0	RW	超高速模式，当控制器时钟频率超过 200MHz 时，必须将此位设为 1。
[25:3]	保留	0x0	RO	---
[2:0]	RWSC[2:0]	0x7	RW	读等待周期配置位，配置在多少个周期内返回读 FLASH 的数据。 000 = 等待 1 个 AHB 总线周期 001 = 等待 2 个 AHB 总线周期 010 = 等待 3 个 AHB 总线周期 011 = 等待 4 个 AHB 总线周期 100 = 等待 5 个 AHB 总线周期 101 = 等待 6 个 AHB 总线周期 110 = 等待 7 个 AHB 总线周期 111 = 等待 8 个 AHB 总线周期

8.5.2.2 EFM 访问控制寄存器（EFAPR）

偏移地址：0x0004~0x0007　　　　复位值：0x00000001

31	30	29	28	27	26	25	24
WPASSWD[31:24]							
w							

23	22	21	20	19	18	17	16
WPASSWD[23:16]							
w							

15	14	13	12	11	10	9	8
WPASSWD[15:8]							
w							

7	6	5	4	3	2	1	0
保留		I0AP		保留		M0AP	
ro		rw		ro		rw	

图表 8-4：EFM 访问控制寄存器（EFAPR）

比特位	名称	复位值	读写属性	功能说明
[31:8]	WPASSWD[31:8]	0x0	W	设置 I0AP 和 M0AP 值得密码。M0AP[1:0]必须在 WPASSWD[31:8]等于 24’ h9786AC 时才能通过 IPS_WDATA[1:0]更改。 I0AP[1:0]必须在 WPASSWD[31:8]等于 24’ hD8D899 时才能通过 IPS_WDATA[5:4]更改。
[7：6]	保留	0x0	RO	---
[5:4]	I0AP[1:0]	0x0	RW	用于信息页的访问控制。 00 = 信息页不可读写 01 = 信息页只可读 10 = 信息页只可写 11 = 信息页可读可写
[3:2]	保留	0x0	RW	---
[1:0]	M0AP[1:0]	0x1	RW	用于主存储块的访问控制。 00 = 主存储块不可读写 01 = 主存储块只可读 10 = 主存储块只可写 11 = 主存储块可读可写

8.5.2.3 EFM 状态寄存器（EFSTAT）

EFM 状态寄存器（EFMSTAT）用来显示 FLASH 状态机命令状态以及访问错误的寄存器。

偏移地址：0x0008~0x000b　　　　复位值：0xc0008000

31	30	29	28	27	26	25	24
PWRRDY	GRD_READ_VLD	FLASH_ERROR	SLOW_ACC_VLD	保留			
ro	ro	ro	ro	ro			

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
EFM_IDLE	保留						
ro	ro						

7	6	5	4	3	2	1	0
保留	FLASH_SLEEP	ECC_ERROR	RDWW	DONE	PEGOOD	SMWERR	FLASH_BUSY
ro	ro	w1c	w1c	w1c	w1c	w1c	ro

图表 8-5：EFM 状态寄存器（EFSTAT）

比特位	名称	复位值	读写属性	功能说明
[31]	PWRRDY	0x1	RO	FLASH 模块的供电是否准备好的标志位。 0 = FLASH 模块上电没有完成 1 = FLASH 上电完成
[30]	GRD_READ_VLD	0x1	RO	表示 FLASH 良片记录是否有效。 0 = FLASH 良片记录错误 1 = FLASH 良片记录错误
[29]	FLASH_ERROR	0x0	RO	表示上电后 Flash 初始化加载是否有错。 0 = FLASH 初始化加载正确 1 = FLASH 初始化加载出错
[28]	SLOW_ACC_VLD	0x1	RO	表示访问 FLASH 的频率采用低频模式。 0 = 访问 FLASH 采用高频模式 1 = 访问 FLASH 采用低频模式
[27:16]	保留	0x0	RO	---
[15]	EFM_IDLE	0x1	RW	表示 FLASH 控制器是否处于空闲。 0 = FLASH 控制器处于忙的状态 1 = FLASH 控制器处于空闲状态
[14:7]	保留	0x0	RO	---
[6]	FLASH_SLEEP	0x0	RO	表示 FLASH 处于 SLEEP 状态的标志位。 0 = FLASH 不在 SLEEP 状态

比特位	名称	复位值	读写属性	功能说明
				1 = FLASH 处于 SLEEP 状态
[5]	ECC_ERROR	0x0	W1C	读 FLASH 时发生 ECC 校验错误标志位。 0 = 读 FLASH 时 ECC 校验正确 1 = 读 FLASH 时出现 ECC 错误
[4]	RDWW	0x0	W1C	当 FLASH 处于编程或擦除状态时发生了读 FLASH 操作的标志位。 0 = 在 FLASH 处于编程或擦除状态时未发生过读 FLASH 操作 1 = 在 FLASH 处于编程或擦除状态时发生过读 FLASH 操作
[3]	DONE	0x0	W1C	FLASH 命令完成标志。新的命令发生时会自动清 0。 0 = 操作 FLASH 的命令正在进行中 1 = 操作 FLASH 的命令已完成
[2]	PEGOOD	0x0	W1C	编程或擦除命令正常结束。新的擦除或编程命令发生时会自动清 0，此标志位只有在 DONE 为 1 时去判断才有效。 0 = FLASH 的编程或擦除命令非正常结束 1 = FLASH 的编程或擦除命令正常结束
[1]	SMWERR	0x0	W1C	表示智能写 FLASH 时是否发生错误的标志位。新的智能写命令会自动将此标志位清 0。 0 = 智能写 FLASH 的命令执行成功 1 = 智能写 FLASH 的命令执行错误
[0]	FLASH_BUSY	0x0	RO	FLASH 处于忙的状态标志位 0 = FLASH 处于空闲状态 1 = FLASH 处于忙的状态

8.5.2.4 EFM 中断屏蔽寄存器（EFINTM）

偏移地址：0x000c~0x000f　　复位值：0x00000038

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留		ECC_ERR_MASK	RDWW_MASK	DONE_MASK	保留		
ro		rw	rw	rw	ro		

图表 8-6：EFM 中断屏蔽寄存器（EFINTM）

比特位	名称	复位值	读写属性	功能说明
[31:6]	保留	0x0	RO	---
[5]	ECC_ERR_MASK	0x1	RW	使能或屏蔽 ECC_ERROR 标志位的中断。 0 = 当产生 ECC 错误时使能 ECC_ERROR 中断 1 = 当产生 ECC 错误时屏蔽 ECC_ERROR 中断
[4]	RDWW_MASK	0x1	RW	使能或屏蔽 RDWW 标志位的中断。 0 = 当在写 FLASH 时发生读的操作时使能 RDWW 中断 1 = 当在写 FLASH 时发生读的操作时屏蔽 RDWW 中断
[3]	DONE_MASK	0x1	RW	使能或屏蔽 DONE 标志位的中断。 0 = 当操作 FLASH 的命令执行完成时使能 DONE 中断 1 = 当操作 FLASH 的命令执行完成时屏蔽 DONE 中断
[2:0]	保留	0x0	RO	---

8.5.2.5 EFM 命令配置寄存器（EFCMD）

偏移地址：0x0010~0x0013　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
CMD_VALID	保留	PRGW_NUM[5:0]					
rw	ro	rw					

7	6	5	4	3	2	1	0
保留			CMD[4:0]				
ro			rw				

图表 8-7：EFM 命令配置寄存器（EFCMD）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15]	CMD_VALID	0x0	RW	用于发起操作 FLASH 的命令。向这一位写 1 会将 CMD[4:0]发送给 FLASH 控制器，这一位会自动清 0。
[14]	保留	0x0	RO	---
[13:8]	PRGW_NUM[5:0]	0x0	RW	用于配置一次智能编程命令需要编程的字的个数。当 CMD_VALID 有效时且 CMD[4:0]为智能编程命令，告诉控制器有 PRGW_NUM[5:0]+1 个字需要写入 FLASH。
[7:5]	保留	0x0	RO	---
[4:0]	CMD[4:0]	0x0	RW	发送给 FLASH 的操作命令。 00000 = 使 FLASH 空闲或恢复到用户模式 00100 = 智能编程命令 00110 = 智能全片擦除命令 00111 = 智能页擦除命令 其他值 = 保留

8.5.2.6 EFM 编程擦除单位时间配置寄存器（ETIMBASE）

EFM 时序参考寄存器（ETIMBASE）用来配置时序参数，参数用于编程或擦除操作时的需要的时间。

偏移地址：0x0018~0x001b　　　　复位值：0x000003C8

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

图表 8-8：EFM 编程擦除单位时间配置寄存器（EFTIMBASE）

比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x	RO	---
[19:16]	20NS_BASE_OF_CLK[3:0]	0x3	RW	用于配置 20 纳秒所需要的时钟（AHB 总线时钟）的个数。
[15:0]	1US_BASE_OF_CLK[15:0]	0xC8	RW	用于配置 1 微秒所需要的时钟（AHB 总线时钟）的个数。

8.5.2.7 EFM 编程擦除时序配置寄存器（ETIMCFG）

偏移地址：0x001c~0x001f 复位值：0x1C9EF877

31	30	29	28	27	26	25	24
保留		CFG_TNVSE[3:0]				CFG_TPWS[3:2]	
ro		rw				rw	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CFG_TPWS[1:0]		CFG_TRE[4:0]					CFG_TME[4]
rw		rw					rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CFG_TME[3:0]				CFG_TPROG[3:0]			
rw				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFG_TNVH[3:0]				CFG_TNVSP[3:0]			
rw				rw			

图表 8-9：EFM 编程擦除时序配置寄存器（ETIMCFG）

比特位	名称	复位值	读写属性	功能说明
[31:30]	保留	0x0	RO	---
[29:26]	CFG_TNVSE[3:0]	0x7	RW	用于配置 Tnvse 参数，单位为 1 微妙。配置得到的 Tnvse 的值必须在 5~10 微秒范围之内。
[25:22]	CFG_TPWS[3:0]	0x2	RW	用于配置 Tpws 参数,单位为 AHB 总线时钟周期。配置得到的 Tpws 的值必须大于 10 纳秒。
[21:17]	CFG_TRE[4:0]	0xF	RW	用于配置 Tre 参数，单位为 1 毫妙。配置得到的 Tre 的值必须在 10~20 毫秒范围之内。
[16:12]	CFG_TME[4:0]	0xF	RW	用于配置 Tme 参数，单位为 1 毫妙。配置得到的 Tme 的值必须在 10~20 毫秒范围之内。
[11:8]	CFG_TPROG[3:0]	0x8	RW	用于配置 Tprog 参数，单位为 1 微妙。配置得到的 Tprog 的值必须在 6~10 微秒范围之内。
[7:4]	CFG_TNVH[3:0]	0x7	RW	用于配置 Tnvh 参数，单位为 1 微妙。配置得到的 Tnvh 的值必须在 5~10 微秒范围之内。
[3:0]	CFG_TNVSP[3:0]	0x07	RW	用于配置 Tnvsp 参数，单位为 1 微妙。配置得到的 Tnvsp 的值必须在 5~10 微秒范围之内。

8.5.2.8 EFM 编程擦除写等待超时寄存器（EFPETIMER）

偏移地址：0x0020~0x0023 复位值：0x00002000

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

图表 8-10：EFM 编程擦除写等待超时寄存器（EFPETIMER）

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x	RO	---
[23:0]	PE_WAIT_CYCLE[23:0]	0x2000	RW	写等待超时配置，单位是 AHB 总线时钟周期。在编程或擦除命令执行时，FLASH 控制器在 PE_WAIT_CYCLE 个周期内还没有等到需要的写操作，PE_GOOD 会变成 0，同时将 DONE 置起。

8.5.2.9 EFM 智能写时序配置寄存器（SMWOP0）

偏移地址：0x0024~0x0027 复位值：0x015E14C8

31	30	29	28	27	26	25	24
保留							WMV[2]
ro							rw

23	22	21	20	19	18	17	16
WMV[1:0]		WHV[3:0]				WIPGM[1:0]	
rw		rw				rw	

15	14	13	12	11	10	9	8
保留		SLOW_CLK	200NS_CLK_NUM				
ro		rw	rw				

7	6	5	4	3	2	1	0
1US_CLK_NUM[7:0]							
rw							

图表 8-11：EFM 智能写时序配置寄存器（SMWOP0）

比特位	名称	复位值	读写属性	功能说明
[31:25]	保留	0x	RO	---
[24:22]	WMV[2:0]	0x5	RW	当使用智能写时，配置写的中间电压值。
[21:18]	WHV[3:0]	0x7	RW	当使用智能写时，配置写的高电压值。
[17:16]	WIPGM[1:0]	0x2	RW	当使用智能写时，配置写的电流值。在编程时，WIPGM 选择两种电流里的一种：当需要写的数据和 FLASH 里的数据比较后，发现不一样的位数大于等于 19 时，WIPGM = 2’ b10；反之，WIPGM = 2’ b11。
[15]	保留	0x0	RO	---
[14:9]	200NS_CLK_NUM[5:0]	0x14	RW	定义写校验时的读周期个数。校验读的时钟周期个数为 2*（200NS_CLK_NUM+1）个。最小可设 0，必须保证 Tclock*(2*200NS_CLK_NUM+0.5)大于 200 纳秒。 **注意：**Tclock 表示单个 AHB 时钟周期的时间。
[8:0]	1US_CLK_NUM[8:0]	0xC8	RW	定义产生大于等于 1 微秒脉冲的 AHB 总线时钟的个数。

注意：此寄存器不建议用户更改。

8.6 功能描述

8.6.1 编程和擦除操作

在发起编程或擦除命令之前需要检查 FLASH 状态，确保 EFM_IDLE 为高，然后才可以发命令给 FLASH 控制器，否则命令会被忽略掉。

8.6.2 FLASH 擦除流程说明

此 FLASH 控制器支持智能全片擦除和智能页擦除命令。步骤如下：

1. 设置 EFAPR 寄存器，将需要擦除的 FLASH 设为可写。
2. 将执行擦除的命令写入 EFCMD 寄存器，并将其命令有效 CMD_VALID 置起。
3. 通过 AHB 接口写一个数据，地址为需要擦除的地址，数据任意。
4. 查询 EFSTAT 寄存器，等待 EFM_IDLE 为 1。
5. 设置 EFAPR 寄存器，将需要擦除的 FLASH 设为只读。

8.6.3 FLASH 编程流程说明

此 FLASH 控制器支持智能编程命令。步骤如下：

1. 设置 EFAPR 寄存器，将需要编程的 FLASH 设为可写。
2. 将执行编程的命令写入 EFCMD 寄存器，同时设置好需要写入的 word 个数（PRGW_NUM+1 个字将会写入 flash)，当 PRGW_NUM 大于零时，这些需要写的数据的地址必须是在同一个 Page 里。然后再将 CMD_VALID 设为有效。
3. 通过 AHB 接口写入需要编程的数据，数据的个数需要与设置 EFCMD 寄存器的 PRG_NUM 的值对应。
4. 查询 EFSTAT 寄存器，等待 EFM_IDLE 为 1
5. 设置 EFAPR 寄存器，将需要编程的 FLASH 设为只读。

8.7 中断描述

EFM 中断事件有三种：

1. 编程或擦除命令结束。
2. 在 FLASH 编程或擦除时发生读 FLASH 操作。
3. 发生 ECC 校验错误。

9 同步串行接口（SSI）

9.1 概述

SSI 是可编程的同步串行接口（SSI）外设。是符合 AMBA 2.0 的 AHB 组件。主处理器通过 AHB 接口访问 SSI 上的数据，控制和状态信息。SSI 还可以使用 DMA 信号集与 DMA 控制器接口。

9.2 特性

SSI 特性包括：

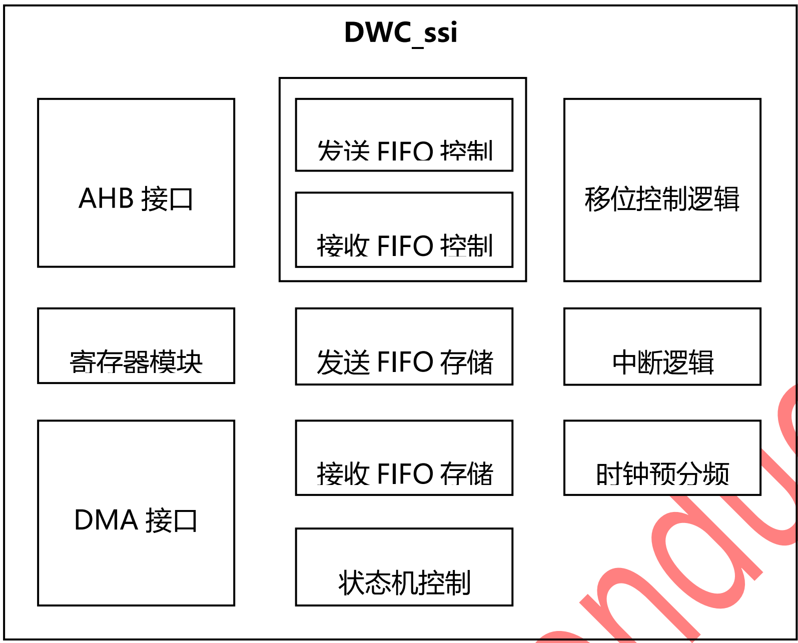
串行主设备操作
DMA 控制器接口–使 SSI 使用握手接口处理传输请求，通过总线与 DMA 控制器接口
扩展 SPI 传输中的时钟延长支持
FIFO 深度–在正常传输模式下为 8 字。在芯片内执行（XIP）传输模式下为 32 字。FIFO 宽度固定为 32 位
扩展 SPI 支持
芯片内执行（XIP）模式支持

9.3 操作模式

SSI 在以下三种模式下运行：

1. 运行模式—运行模式是正常的操作模式。
2. 瞌睡模式—瞌睡模式是可配置的低功耗模式。
3. 停止模式—SSI 在停止模式下不活动。

9.4 框图



图表 9-1：SSI 框图

9.5 存映射和寄存器

9.5.1 内存映射

表格 9-1：EDMAC 内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	SSI 控制寄存器 0(SSICTRLR0)		S/U
0x0004	SSI 控制寄存器 1(SSICTRLR1)		S/U
0x0008	SSI 使能寄存器(SSIENR)		S/U
0x000C	SSI Microwire 控制寄存器(MWCR)		S/U
0x0010	SSI 从选择寄存器(SSISER)		S/U
0x0014	SSI 波特率选择寄存器(BAUDR)		S/U
0x0018	SSI 发送 FIFO 阈值寄存器(SSITXFTLR)		S/U
0x001C	SSI 接收 FIFO 阈值寄存器(SSIRXFTLR)		S/U
0x0020	SSI 发送 FIFO 水平寄存器(SSITXFLR)		S/U
0x0024	SSI 接收 FIFO 水平寄存器(SSIRXFLR)		S/U
0x0028	SSI 状态寄存器(SSISR)		S/U
0x002C	SSI 中断屏蔽寄存器(SSIIMR)		S/U
0x0030	SSI 中断状态寄存器(SSIISR)		S/U
0x0034	SSI 原始中断状态寄存器(SSIRISR)		S/U

偏移地址	位 31-16	位 15-0	访问权限
0x0038	SSI 发送 FIFO 溢出中断清除寄存器(SSITXOICR)		S/U
0x003C	SSI 接收 FIFO 溢出中断清除寄存器(SSIRXOICR)		S/U
0x0040	SSI 接收 FIFO 下溢出中断清除寄存器(SSIRXUICR)		S/U
0x0048	SSI 中断清除寄存器(SSIICR)		S/U
0x004C	SSI DMA 控制寄存器(SSIDMACR)		S/U
0x0050	SSI DMA 发送 FIFO 数据水平寄存器 (SSIDMATDLR)		S/U
0x0054	SSI DMA 接收 FIFO 数据水平寄存器 (SSIDMARDLR)		S/U
0x0060+i*0x4	SSI 数据寄存器(SSIDRx)		S/U
0x00F0	SSI 采样延时寄存器(SSIRXSDR)		S/U
0x00F4	SSI SPI 控制寄存器(SPICTRLR0)		S/U
0x00FC	SSI XIP 模式位寄存器(SSIXIPMBR)		S/U
0x0100	SSI XIP 递增命令寄存器(SSIIIR)		S/U
0x0104	SSI XIP 回环命令寄存器(SSIWIR)		S/U
0x0108	SSI XIP 控制寄存器(SSIXIPCR)		S/U
0x010C	SSI XIP 从使能寄存器(SSIXIPSER)		S/U
0x0110	SSI XIP 接收 FIFO 溢出中断清除寄存器 (SSIXRXIOCR)		S/U

9.5.2 寄存器描述

9.5.2.1 SSI 控制寄存器 0(SSICTRLR0)

偏移地址：0x0000~0x0003 复位值：0x0080481F

31	30	29	28	27	26	25	24
保留							
ro							
23	22	21	20	19	18	17	16
SPI_FRF		保留		CFS			
rw		ro		rw			
15	14	13	12	11	10	9	8
保留	SSTE	SRL	保留	TMOD		SCPOL	SCPH
ro	rw	rw	ro	rw		rw	rw
7	6	5	4	3	2	1	0
FRF		保留	DFS				
rw		ro	rw				

图表 9-2：SSI 控制寄存器 0(SSICTRLR0)

该寄存器控制串行数据传输。使能 SSI 后，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23:22]	SPI_FRF	0x2	RW	SPI 帧格式 选择用于发送/接收数据的数据帧格式。 0x0 = 标准 SPI 格式 0x1 = 双线 SPI 格式 0x2 = 四线 SPI 格式 0x3 = 保留
[21:20]	保留	0x0	RO	---
[19:16]	CFS	0x0	RW	CFS—控制框架大小 选择 Microwire 帧格式的控制字的长度。 控制字的长度为（配置值+1）。
[15]	保留	0x0	RO	---
[14]	SSTE	0x1	RW	片选切换。 在时钟相位（SCPH）设置为 SPI 模式下工作时，该寄存器控制数据帧之间片选（ss）的行为。 0 = ss 将保持低电平，并且在传输期间 sclk 将连续运行 1 = ss 将在连续的数据帧之间置起，并且在 ss 为高电平时将串行时钟（sclk）保持为其默认值

比特位	名称	复位值	读写属性	功能说明
[13]	SRL	0x0	RW	移位寄存器循环。 仅用于测试目的。有效时，将发送移位寄存器输出连接到接收移位寄存器输入。 0 = 正常模式操作 1 = 测试模式操作
[12]	保留	0x0	RO	---
[11:10]	TMOD	0x2	RW	选择串行通信的传输方式。 0x0 = 发送和接收；不适用于扩展 SPI 操作模式 0x1 = 仅发送模式，扩展 SPI 操作模式写入 0x2 = 仅接收模式，扩展 SPI 操作模式读取 0x3 = 保留
[9]	SCOPL	0x0	RW	串行时钟极性。 当帧格式（FRF）设置为 Motorola SPI 时有效。 用于选择无效串行时钟的极性，当 SSI 主设备未在串行总线上主动传送数据时，该极性保持无效。 0 = 串行时钟的无效状态为低电平 1 = 串行时钟的无效状态为高电平 0 = 串行时钟的无效状态为低电平 1 = 串行时钟的无效状态为高电平
[8]	SCPH	0x0	RW	串行时钟相位。 当帧格式（FRF）设置为 Motorola SPI 时有效。 串行时钟相位选择串行时钟与从机选择信号的关系。当 SCPH = 0 时，数据在串行时钟的第一个边沿捕获。当 SCPH = 1 时，片选拉低后，串行时钟切换一个周期，并在串行时钟的第二个边沿捕获数据。 0 = 串行时钟在第一位的中间切换 1 = 串行时钟在第一位开始时切换
[7:6]	FRF	0x0	RW	帧格式。 选择哪个串行协议传输数据。 0x0 = 摩托罗拉 SPI 帧格式 0x1 = 德州仪器（TI）SSP 帧格式 0x2 = 美国国家半导体的 Microwire 帧格式 0x3 = 保留
[5]	保留	0x0	RO	---
[4:0]	DFS	0x1F	RW	选择数据帧长度。 当数据帧大小少于 32 位，接收数据会自动由接收器逻辑右对齐，高位补零。 在写入发送 FIFO 之前，您必须对发送数据右对齐，

比特位	名称	复位值	读写属性	功能说明
				传输逻辑在传输数据时会忽略未使用的高位。 数据帧的长度为（配置值+1），数据帧的长度最小为 4 位。

9.5.2.2 SSI 控制寄存器 1(SSICTRLR1)

偏移地址：0x0004~0x0007 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
NDF							
rw							

7	6	5	4	3	2	1	0
NDF							
rw							

图表 9-3：SSI 控制寄存器 1(SSICTRLR1)

在仅接收模式和仅传输模式下，控制寄存器 1 控制串行传输的结束长度。使能 SSI 后，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	NDF	0x0	RW	数据帧数。 当 TMOD = 01 或 TMOD = 10 时，此寄存器字段设置要由 SSI 连续接收或发送的数据帧数。SSI 持续接收或发送串行数据，直到数据帧数等于该寄存器值加 1 为止。当 TMOD = 01 时，必须设置 SPI_CTRLR0 中的 CLK_STRETCH_EN。

9.5.2.3 SSI 使能寄存器(SSIENR)

偏移地址：0x0008~0x000B 复位值：0x00000001

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

图表 9-4：SSI 使能寄存器(SSIENR)

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	SSIC_EN	0x1	RW	SSI 使能。 使能和禁止所有 SSI 操作。禁止后，所有串行传输将立即停止，将清除发送和接收 FIFO 缓存。使能时，不能对某些 SSI 控制寄存器进行写入。 0 = 禁止 SSI 1 = 使能 SSI

9.5.2.4 SSI Microwire 控制寄存器(MWCR)

偏移地址：0x000C~0x000F 复位值：0x00000000

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

图表 9-5：SSI Microwire 控制寄存器(MWCR)

该寄存器控制半双工 Microwire 串行协议的数据字方向。使能 SSI 后，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:3]	保留	0x0	RO	---
[2]	MHS	0x0	RW	Microwire 握手。 用于使能和禁止 Microwire 协议的忙/就绪握手接口。使能后，SSI 会在传输完最后一个数 RW 据/控制位之后，在清除 SR 寄存器中的 BUSY 状态之前，检查从机的就绪状态。 0 = 禁止握手 1 = 使能握手
[1]	MDD	0x0	RW	Microwire 控制。 使用 Microwire 串行协议时，定义数据字的方向。当该位设置为 0 时，SSI 从外部串行设备接收数据字。当此位设置为 1 时，数据字从 SSI 传输到外部串行设备。 0 = SSI 接收数据 1 = SSI 传输数据

比特位	名称	复位值	读写属性	功能说明
[0]	MWMOD	0x0	RW	Microwire 传输模式。 定义 Microwire 传输是序列传输还是非序列 传输。使用序列模式时，只需要一个控制字就可以发送或接收数据字块。使用非序列模式时，每个发送或接收的数据字都必须有一个控制字。 0 = 非序列传输 1 = 序列传输

9.5.2.5 SSI 从选择寄存器(SSISER)

偏移地址：0x0010~0x0013　　　　复位值：0x00000001

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							SER
ro							rw

图表 9-6：SSI 从选择寄存器(SSISER)

该寄存器使能 SSI 主设备的各个从设备片选。当 SSI 繁忙且 SSIC_EN = 1 时，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	SER	0x1	RW	片选使能标志。 0 = 禁止 1 = 使能

9.5.2.6 SSI 波特率选择寄存器(BAUDR)

偏移地址：0x0014~0x0017 复位值：0x00000002

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

图表 9-7：SSI 波特率选择寄存器(BAUDR)

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	SCKDV	0x2	RW	SSI 时钟分频器。 该字段的第 0 位始终保持为 0，并且不受写操作的影响，这确保了该寄存器中的值保持偶数。如果值为 0，则禁止串行输出时钟（sclk_out）。sclk_out 的频率从以下公式得出： **Fsclk_out = Fssi_clk / SCKDV** 其中 SCKDV 是 2 到 65534 之间的任意偶数。例如：对于 Fssi_clk = 3.6864MHz 和 SCKDV= 2，Fsclk_out = 3.6864 / 2 = 1.8432MHz。

9.5.2.7 SSI 发送 FIFO 阈值寄存器(SSITXFTLR)

偏移地址：0x0018~0x001B 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留			TXFTHR				
ro			rw				

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留			TFT				
ro			rw				

图表 9-8：SSI 发送 FIFO 阈值寄存器(SSITXFTLR)

该寄存器控制发送 FIFO 的阈值。

比特位	名称	复位值	读写属性	功能说明
[31:21]	保留	0x0	RO	---
[20:16]	TXFTHR	0x0	RW	传输开始 FIFO 水平。 用于控制发送 FIFO 中数据量水平，高于该水平将在串行线上开始传输。在开始对串行线的写操作之前，该寄存器可用于确保发送 FIFO 中存在足够的数据。
[15:5]	保留	0x0	RO	--
[4:0]	TFT	0x0	RW	发送 FIFO 阈值。 控制发送 FIFO 控制器触发中断的数据水平。 如果尝试将此值写为大于或等于 FIFO 的深度，则不会写入该字段，并保留其当前值。当发送 FIFO 水平小于或等于该值时，将触发发送 FIFO 空中断。

9.5.2.8 SSI 接收 FIFO 阈值寄存器(SSIRXFTLR)

偏移地址：0x001C~0x001F 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留			RFT				
ro			rw				

图表 9-9：SSI 接收 FIFO 阈值寄存器(SSIRXFTLR)

该寄存器控制接收 FIFO 的阈值。

比特位	名称	复位值	读写属性	功能说明
[31:5]	保留	0x0	RO	---
[4:0]	TFT	0x0	RW	接收 FIFO 阈值。 控制接收 FIFO 控制器触发中断的数据水平。如果尝试将此值设置为大于 FIFO 的深度，则不会写入该字段，并保留其当前值。当接收 FIFO 条目数大于或等于该值+1 时，将触发接收 FIFO 满中断。该值应设置为非零值，否则 SSI 将在接收 FIFO 完全满后持续发出满中 断。

9.5.2.9 SSI 发送 FIFO 水平寄存器(SSITXFLR)

偏移地址：0x0020~0x0023 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留		RFT					
ro		rw					

图表 9-10：SSI 发送 FIFO 水平寄存器(SSITXFLR)

比特位	名称	复位值	读写属性	功能说明
[31:6]	保留	0x0	RO	---
[5:0]	TFT	0x0	RW	发送 FIFO 水平 包含发送 FIFO 中有效数据条目的数量。

9.5.2.10 SSI 接收 FIFO 水平寄存器(SSIRXFLR)

偏移地址：0x0024~0x0027　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留		RXFLR					
ro		rw					

图表 9-11：SSI 接收 FIFO 水平寄存器(SSIRXFLR)

比特位	名称	复位值	读写属性	功能说明
[31:6]	保留	0x0	RO	---
[5:0]	TFT	0x0	RW	接收 FIFO 水平 包含接收 FIFO 中有效数据条目的数量。

9.5.2.11 SSI 状态寄存器(SSISR)

偏移地址：0x0028~0x002B　　复位值：0x00000006

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留			RFF	RFNE	TFE	TFNF	BUSY
ro			ro	ro	ro	ro	ro

图表 9-12：SSI 状态寄存器(SSISR)

比特位	名称	复位值	读写属性	功能说明
[31:5]	保留	0x0	RO	---
[4]	RFF	0x0	RO	接收 FIFO 已满。 当接收 FIFO 完全满时，该位置 1。当接收 FIFO 包含一个或多个空位置时，该位被清除。 0 = 接收 FIFO 未满 1 = 接收 FIFO 已满
[3]	RFNE	0x0	RO	接收 FIFO 不为空。 当接收 FIFO 包含一个或多个数据时置 1，并在接收 FIFO 为空时清除。该位可以由软件查询，以完全清空接收 FIFO。 0 = 接收 FIFO 为空 1 = 接收 FIFO 不为空
[2]	TFE	0x0	RO	发送 FIFO 为空。 当发送 FIFO 完全为空时，该位置 1。当发送 FIFO 包含一个或多个有效数据时，该位被清除。该位字段不发中断请求。 0 = 发送 FIFO 不为空 1 = 发送 FIFO 为空
[1]	TFNF	0x0	RO	发送 FIFO 未满。 当发送 FIFO 包含一个或多个空位置时置 1，并在 FIFO 满时将其清除。 0 = 发送 FIFO 已满

比特位	名称	复位值	读写属性	功能说明
				1 = 发送 FIFO 未满
[0]	BUSY	0x0	RO	SSI 忙标志。 置位时表示正在进行串行传输；清除时表示 SSI 空闲或禁用。 0 = SSI 空闲或禁用 1 = SSI 正在主动传输数据

9.5.2.12 SSI 中断屏蔽寄存器(SSIMR)

偏移地址：0x002C~0x002F 复位值：0x0000007F

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留	XRXOIM	保留	RXFIM	RXOIM	RXUIM	TXOIM	TXEIM
ro	rw	ro	rw	rw	rw	rw	rw

图表 9-13：SSI 中断屏蔽寄存器(SSIMR)

比特位	名称	复位值	读写属性	功能说明
[31:7]	保留	0x0	RO	---
[6]	XRXOIM	0x1	RW	XIP 接收 FIFO 溢出中断屏蔽 0 = ssi_xrxo_intr 中断被屏蔽 1 = ssi_xrxo_intr 中断未屏蔽
[5]	保留	0x1	RO	---
[4]	RXFIM	0x1	RW	接收 FIFO 满中断屏蔽 0 = ssi_rxf_intr 中断被屏蔽 1 = ssi_rxf_intr 中断未屏蔽
[3]	RXOIM	0x1	RW	接收 FIFO 溢出中断屏蔽 0 = ssi_rxo_intr 中断被屏蔽 1 = ssi_rxo_intr 中断未屏蔽
[2]	RXUIM	0x1	RW	接收 FIFO 下溢中断屏蔽 0 = ssi_rxu_intr 中断被屏蔽 1 = ssi_rxu_intr 中断未屏蔽
[1]	TXOIM	0x1	RW	发送 FIFO 溢出中断屏蔽 0 = ssi_txo_intr 中断被屏蔽 1 = ssi_txo_intr 中断未屏蔽
[0]	TXEIM	0x1	RW	发送 FIFO 空中断屏蔽 0 = ssi_txe_intr 中断被屏蔽 1 = ssi_txe_intr 中断未屏蔽

9.5.2.13 SSI 中断状态寄存器(SSIISR)

偏移地址：0x0030~0x0033 复位值：0x00000001

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留	XRXOIS	保留	RXFIS	RXOIS	RXUIS	TXOIS	TXEIS
ro	ro	ro	ro	ro	ro	ro	ro

图表 9-14：SSI 中断状态寄存器(SSIISR)

该寄存器报告 SSI 屏蔽后中断的状态。

比特位	名称	复位值	读写属性	功能说明
[31:7]	保留	0x0	RO	---
[6]	XRXOIS	0x0	RO	XIP 接收 FIFO 溢出中断状态 0 = 屏蔽后 ssi_xrxo_intr 中断无效 1 = 屏蔽后 ssi_xrxo_intr 中断有效
[5]	保留	0x0	RO	---
[4]	RXFIS	0x0	RO	接收 FIFO 满中断状态 0 = 屏蔽后 ssi_rxf_intr 中断无效 1 = 屏蔽后 ssi_rxf_intr 中断有效
[3]	RXOIS	0x0	RO	接收 FIFO 溢出中断状态 0 = 屏蔽后 ssi_rxo_intr 中断无效 1 = 屏蔽后 ssi_rxo_intr 中断有效
[2]	RXUIS	0x0	RO	接收 FIFO 下溢中断状态 0 = 屏蔽后 ssi_rxu_intr 中断无效 1 = 屏蔽后 ssi_rxu_intr 中断有效
[1]	TXOIS	0x0	RO	发送 FIFO 溢出中断状态 0 = 屏蔽后 ssi_txo_intr 中断无效 1 = 屏蔽后 ssi_txo_intr 中断有效
[0]	TXEIS	0x1	RO	发送 FIFO 空中断状态 0 = 屏蔽后 ssi_txe_intr 中断无效 1 = 屏蔽后 ssi_txe_intr 中断有效

9.5.2.14 SSI 原始中断状态寄存器(SSIRISR)

偏移地址：0x0034~0x0037　　　　复位值：0x00000001

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留	XRXOIR	保留	RXFIR	RXOIR	RXUIR	TXOIR	TXEIR
ro	ro	ro	ro	ro	ro	ro	ro

图表 9-15：SSI 原始中断状态寄存器(SSIRISR)

该寄存器报告 SSI 屏蔽前中断的状态。

比特位	名称	复位值	读写属性	功能说明
[31:7]	保留	0x0	RO	---
[6]	XRXOIR	0x0	RO	XIP 接收 FIFO 溢出中断原始状态 0 = 屏蔽前 ssi_xrxo_intr 中断无效 1 = 屏蔽前 ssi_xrxo_intr 中断有效
[5]	保留	0x0	RO	---
[4]	RXFIR	0x0	RO	接收 FIFO 满中断原始状态 0 = 屏蔽前 ssi_rxf_intr 中断无效 1 = 屏蔽前 ssi_rxf_intr 中断有效
[3]	RXOIR	0x0	RO	接收 FIFO 溢出中断原始状态 0 = 屏蔽前 ssi_rxo_intr 中断无效 1 = 屏蔽前 ssi_rxo_intr 中断有效
[2]	RXUIR	0x0	RO	接收 FIFO 下溢中断原始状态 0 = 屏蔽前 ssi_rxu_intr 中断无效 1 = 屏蔽前 ssi_rxu_intr 中断有效
[1]	TXOIR	0x0	RO	发送 FIFO 溢出中断原始状态 0 = 屏蔽前 ssi_txo_intr 中断无效 1 = 屏蔽前 ssi_txo_intr 中断有效
[0]	TXEIR	0x1	RO	发送 FIFO 空中断原始状态 0 = 屏蔽前 ssi_txe_intr 中断无效 1 = 屏蔽前 ssi_txe_intr 中断有效

9.5.2.15 SSI 发送 FIFO 溢出中断清除寄存器(SSITXOICR)

偏移地址：0x0038~0x003B 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							TXOICR
ro							rc

图表 9-16：SSI 发送 FIFO 溢出中断清除寄存器(SSITXOICR)

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	TXOICR	0x0	RC	清除发送 FIFO 溢出中断。 该寄存器反映了中断的状态。对该寄存器的读操作将清除 ssi_txo_intr 中断，写无效。

9.5.2.16 SSI 接收 FIFO 溢出中断清除寄存器(SSIRXOICR)

偏移地址：0x003C~0x003F　　复位值：0x00000000

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

图表 9-17：SSI 接收 FIFO 溢出中断清除寄存器(SSIRXOICR)

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	RXOICR	0x0	RC	清除接收 FIFO 溢出中断。 该寄存器反映了中断的状态。对该寄存器的读操作将清除 ssi_rxo_intr 中断，写无效。

9.5.2.17 SSI 接收 FIFO 下溢出中断清除寄存器(SSIRXUICR)

偏移地址：0x0040~0x0043 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							RXUICR
ro							rc

图表 9-18：SSI 接收 FIFO 下溢出中断清除寄存器(SSIRXUICR)

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	RXUICR	0x0	RC	清除接收 FIFO 下溢出中断。 该寄存器反映了中断的状态。对该寄存器的读操作将清除 ssi_rxu_intr 中断，写无效。

9.5.2.18 SSI 中断清除寄存器(SSIICR)

偏移地址：0x0048~0x004B　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							ICR
ro							rw

图表 9-19：SSI 中断清除寄存器(SSIICR)

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	ICR	0x0	RC	清除中断。 如果以下任何中断有效，则该寄存器被置 1。 读取将清除 ssi_txo_intr，ssi_rxu_intr，ssi_rxo_intr 中断。写入该寄存器无效。

9.5.2.19 SSI DMA 控制寄存器(SSIDMACR)

偏移地址：0x004C~0x004F　　复位值：0x00000000

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

图表 9-20：SSI DMA 控制寄存器(SSIDMACR)

该寄存器用于启用 DMA 控制器接口操作。

比特位	名称	复位值	读写属性	功能说明
[31:2]	保留	0x0	RO	---
[1]	TDMAE	0x0	RW	发送 DMA 使能 该位使能/禁止发送 FIFO DMA 通道。 0 = 禁止发送 DMA 1 = 使能发送 DMA
[0]	RDMAE	0x0	RW	接收 DMA 使能 该位使能/禁止接收 FIFO DMA 通道。 0 = 禁止接收 DMA 1 = 使能接收 DMA

9.5.2.20 SSI DMA 发送 FIFO 数据水平寄存器(SSIDMATDLR)

偏移地址：0x0050~0x0053　　复位值：0x00000000

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

图表 9-21：SSI DMA 发送 FIFO 数据水平寄存器(SSIDMATDLR)

比特位	名称	复位值	读写属性	功能说明
[31:5]	保留	0x0	RO	---
[4:0]	DMATDL	0x0	RW	DMA 发送数据水平。 该字段控制发送逻辑发出 DMA 请求的水平。 它等于水位，也就是说，当发送 FIFO 中的有效数据数等于或小于此字段值且 TDMAE = 1 时，将生成 dma_tx_req 信号。

9.5.2.21 SSI DMA 接收 FIFO 数据水平寄存器(SSIDMARDLR)

偏移地址：0x0054~0x0057 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留			DMARDL				
ro			rw				

图表 9-22：SSI DMA 接收 FIFO 数据水平寄存器(SSIDMARDLR)

比特位	名称	复位值	读写属性	功能说明
[31:5]	保留	0x0	RO	---
[4:0]	DMARDL	0x0	RW	DMA 接收数据水平。 该字段控制接收逻辑发出 DMA 请求的水平。水位 = DMARDL + 1; 也就是说，当接收 FIFO 中的有效数据条目数等于或大于此字段值 +1，并且 RDMAE = 1 时，将生成 dma_rx_req。

9.5.2.22 SSI 数据寄存器(SSIDRx)

偏移地址：0x0060+i*0x4　　复位值：0x00000000

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

图表 9-23：SSI 数据寄存器(SSIDRx)

SSI 数据寄存器用于发送/接收 FIFO 的 32 位读/写缓存。读取寄存器后，将访问接收 FIFO 缓存中的数据。写入时，数据被移入发送 FIFO 缓存。仅当 SSIC_EN = 1 时才会发生写操作。当 SSIC_EN = 0 时，FIFO 被复位。

比特位	名称	复位值	读写属性	功能说明
[31:0]	DR	0x0	RW	数据寄存器。 写入该寄存器时，必须对数据进行右对齐。读取的数据会自动右对齐。 读 = 接收 FIFO 缓存 写 = 发送 FIFO 缓存。

9.5.2.23 SSI 采样延时寄存器(SSIRXSDR)

偏移地址：0x00F0~0x00F4 复位值：0x00000000

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

图表 9-24：SSI 采样延时寄存器(SSIRXSDR)

比特位	名称	复位值	读写属性	功能说明
[31:17]	保留	0x0	RO	---
[16]	SE	0x0	RW	SE—接收数据（rxd）采样边沿。 0 = ssi_clk 的上升沿将用于采样传入数据 1 = ssi_clk 的下降沿将用于采样传入数据
[15:8]	保留	0x0	RO	---
[7:0]	TFT	0x0	RW	RSD-接收数据（rxd）采样延迟。 该寄存器用于延迟 rxd 输入端口的采样。每个值代表 rxd 样本上的单个 ssi_clk 延迟

9.5.2.24 SSI SPI 控制寄存器(SPICTRLR0)

偏移地址：0x00FC~0x00FF 复位值：0x00000218

31	30	29	28	27	26	25	24
保留	CLK_STRETCH	保留					
ro	rw	ro					

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
WAIT_CYCLES					保留	INST_L	
rw					ro	rw	

7	6	5	4	3	2	1	0
保留		ADDR_L				TRANS_TYPE	
ro		rw				rw	

图表 9-25：SSI SPI 控制寄存器(SPICTRLR0)

比特位	名称	复位值	读写属性	功能说明
[31]	保留	0x0	RO	---
[30]	CLK_STRETCH	0x0	RW	在 SPI 传输中启用时钟扩展功能。 在写入的情况下，如果 FIFO 变空，SSI 将延长时钟，直到 FIFO 具有足够的数据来继续传输。在读取的情况下，如果接收 FIFO 变满，SSI 将停止时钟，直到从 FIFO 读取数据为止。 •注意：建议始终设置此位 0 = 禁止时钟扩展 1 = 使能时钟扩展
[29:16]	保留	0x0	RO	---
[15:11]	WAIT_CYCLES	0x0	RW	控制帧发送和数据接收之间的双线/四线模式等待 SPI 时钟周期数。
[10]	保留	0x0	RO	---
[9:8]	INST_L	0x2	RW	双线/四线模式指令长度 0x0 = 无指令 0x1 = 4 位指令长度 0x2 = 8 位指令长度 0x3 = 16 位指令长度
[7:6]	保留	0x0	RO	---
[5:2]	ADDR_L	0x6	RW	地址长度 地址长度 = 配置数*4

比特位	名称	复位值	读写属性	功能说明
[1:0]	TRANS_TYPE	0x0	RW	地址和指令传输格式。 选择 SSI 将以标准 SPI 模式还是在 CTRLR0.SPI_FRF 字段中选择的 SPI 模式发送指令/地址。 0x0 = 指令和地址将以标准 SPI 模式发送。 0x1 = 指令将以标准 SPI 模式发送，地址将以 CTRLR0.SPI_FRF 指定的模式发送。 0x2 = 指令和地址都将以 SPI_FRF 指定的模式发送。 0x3 = 保留

9.5.2.25 SSI XIP 模式位寄存器(SSIXIPMBR)

偏移地址：0x00FC~0x00FF　　　　复位值：0x00000000

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

图表 9-26：SSI XIP 模式位寄存器(SSIXIPMBR)

在该寄存器包含在地址阶段之后以 XIP 工作模式发送的模式位，仅当 SSIENR 寄存器设置为 0 时才能写入此寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	XIP_MD_BITS	0x0	RW	XIP 传输的地址阶段之后要发送的 XIP 模式位。

9.5.2.26 SSI XIP 递增命令寄存器(SSIIIR)

偏移地址：0x0100~0x0103　　　　复位值：0x0000006B

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

图表 9-27：SSI XIP 递增命令寄存器(SSIIIR)

当在 AHB 接口上请求 INCR 时，此寄存器用于存储将在 INCR 请求中使用的指令操作码。启用 SSI 后，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	INCR_INST	0x6B	RW	XIP INCR 传输操作码。 当 SPI_CTRLR0.XIP_INST_EN 位设置为 1 时，SSI 发送 XIP 传输的指令，该寄存器字段存储当在 AHB 总线上请求 INCR 类型传输时要发送的指令操作码。在指令阶段要发送的位数由 SPI_CTRL0.INST_L 字段确定。

9.5.2.27 SSI XIP 回环命令寄存器(SSIWIR)

偏移地址：0x0104~0x0107 复位值：0x0000006B

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

图表 9-28：SSI XIP 回环命令寄存器(SSIWIR)

当在 AHB 接口上请求 WRAP 时，此寄存器用于存储将在 WRAP 请求中使用的指令操作码。启用 SSI 后，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	WRAP_INST	0x6B	RW	XIP WRAP 传输操作码。 当 SPI_CTRLR0.XIP_INST_EN 位设置为 1 时，SSI 发送 XIP 传输的指令，该寄存器字段存储当在 AHB 总线上请求 WRAP 类型传输时要发送的指令操作码。在指令阶段要发送的位数由 SPI_CTRL0.INST_L 字段确定。

9.5.2.28 SSI XIP 控制寄存器(SSIXIPCR)

偏移地址：0x0108~0x010B 复位值：0x28c10462

31	30	29	28	27	26	25	24
保留		XIP_PREFETCH_EN	保留	XIP_MBL		保留	
ro		rw	ro	rw		ro	

23	22	21	20	19	18	17	16
CONT_XFER_EN	INST_EN	保留			DFS_HC	WAIT_CYCLES	
rw	rw	ro			rw	rw	

15	14	13	12	11	10	9	8
WAIT_CYCLES			MD_BOTS_EN	保留	INST_L		保留
rw			rw	ro	rw		ro

7	6	5	4	3	2	1	0
ADDR_L				TRANS_TYPE		FRF	
rw				rw		rw	

图表 9-29：SSI XIP 控制寄存器(SSIXIPCR)

比特位	名称	复位值	读写属性	功能说明
[31:30]	保留	0x0	RO	---
[29]	XIP_PREFETCH_EN	0x1	RW	XIP 预取使能位 0 = 禁止 SSI 中的 XIP 预取功能。 1 = 使能 SSI 中的 XIP 预取功能。
[28]	保留	0x0	RO	---
[27:26]	XIP_MBL	0x2	RW	XIP 模式位的长度。 设置 XIP 操作模式下模式位的长度。 0x0 = 模式位长度等于 2 0x1 = 模式位长度等于 4 0x2 = 模式位长度等于 8 0x3 = 模式位长度等于 16
[25:24]	保留	0x0	RO	---
[23]	CONT_XFER_EN	0x1	RW	XIP 连续传输使能位 0 = 在 XIP 模式下禁止连续传输。 1 = 在 XIP 模式下使能连续传输。
[22]	INST_EN	0x1	RW	XIP 指令使能位 0 = XIP 传输将没有指令阶段。 1 = XIP 传输将具有指令阶段。
[21:19]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[18]	DFS_HC	0x0	RW	固定用于 XIP 传输的 DFS 0 = 数据帧大小将由 HSIZE 和 HBURST 信号确定 1 = XIP 传输的数据帧大小将固定为 CTRLR0.DFS 中的编程值。此位固定配置为 0
[17:13]	WAIT_CYCLES	0x8	RW	等待周期 控制帧发送和数据接收之间的双线/四线模式等待 SPI 时钟周期数。
[12]	MD_BOTS_EN	0x0	RW	XIP 模式位使能。 0 = 地址阶段之后没有模式位。 1 = 地址阶段之后插入模式位。
[11]	保留	0x0	RO	---
[10:9]	INST_L	0x2	RW	双线/四线模式指令长度。 0x0 = 无指令 0x1 = 4 位指令长度 0x2 = 8 位指令长度 0x3 = 16 位指令长度
[8]	保留	0x0	RO	---
[7:4]	ADDR_L	0x6	RW	XIP 地址长度 地址长度 = 配置数*4
[3:2]	TRANS_TYPE	0x0	RW	地址和指令传输格式。 选择 SSI 将以标准 SPI 模式还是在 CTRLR0.SPI_FRF 字段中选择的 SPI 模式发送指令/地址。 0x0 = 指令和地址将以标准 SPI 模式发送。 0x1 = 指令将以标准 SPI 模式发送，地址将以 XIP_CTRL.SPI_FRF 指定的模式发送。 0x2 = 指令和地址都将以 XIP_CTRL.SPI_FRF 指定的模式发送。 0x3 = 保留。
[1:0]	FRF	0x2	RW	SPI 帧格式 选择用于发送/接收数据的数据帧格式。 0x0 = 保留 0x1 = 双线 SPI 格式 0x2 = 四线 SPI 格式 0x3 = 保留

9.5.2.29 SSI XIP 从使能寄存器(SSIXIPSER)

偏移地址：0x010C~0x010F　　　　复位值：0x00000001

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

图表 9-30：SSI XIP 从使能寄存器(SSIXIPSER)

该寄存器使能 SSI 主设备的各个从设备片选，以进行 XIP 操作模式。当 SSI 繁忙且 SSIC_EN = 1 时，无法写入该寄存器。

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	SER	0x1	RW	片选使能标志。 0 = 禁止 1 = 使能

9.5.2.30 SSI XIP 接收 FIFO 溢出中断清除寄存器(SSIXRXIOCR)

偏移地址：0x0040~0x0043 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							XRXOICR
ro							rc

图表 9-31：SSI XIP 接收 FIFO 溢出中断清除寄存器(SSIXRXIOCR)

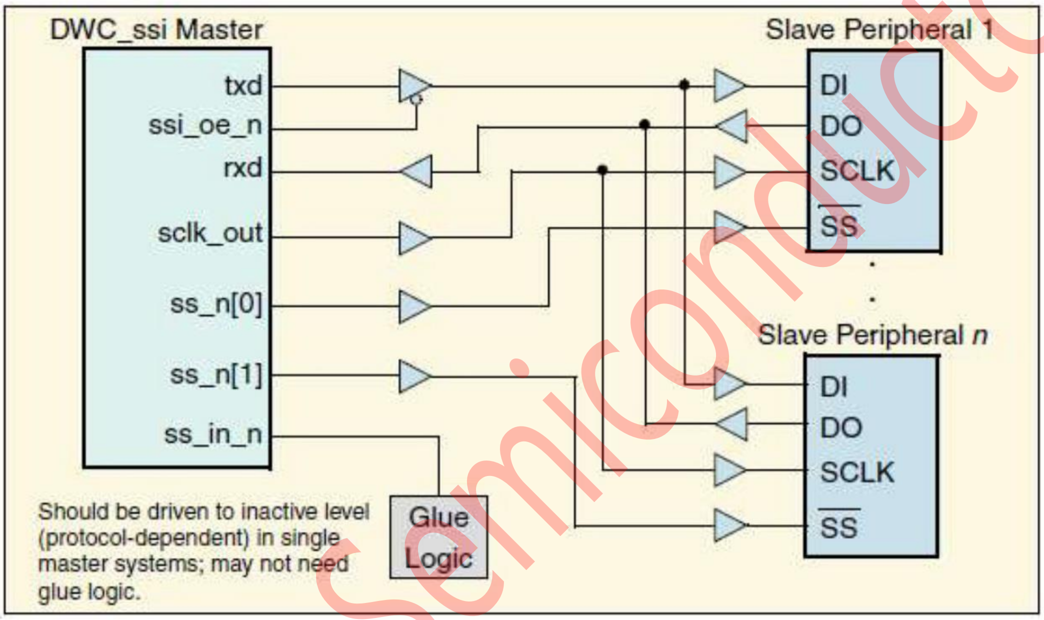
比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	XRXOICR	0x0	RC	清除 XIP 接收 FIFO 溢出中断。 该寄存器反映了中断的状态。读该寄存器将清除 ssi_xrxo_intr（_n）中断，写无效。

9.6 功能描述

9.6.1 主模式

此模式与串行从属外围设备进行串行通信。当配置为串行主设备时，SSI 会启动并控制所有串行传输。图表 9-32 显示了将 SSI 配置为串行主设备的示例，并将串行总线上的所有其他设备配置为串行从设备。

SSI 生成和控制的串行比特率时钟在 sclk_out 线上被输出。当禁止 SSI（SSIC_EN = 0）时，将不会发生串行传输，并且 sclk_out 会按照其操作所依据的串行协议的定义保持在无效状态。



图表 9-32：SSI 配置为主设备

9.6.2 时钟比率

SSI 采用过采样架构。对于主操作模式，外设时钟（sclk_out）周期是内部时钟（ssi_clk）的倍数。将 SSI 配置为主设备时，位速率时钟（sclk_out）的最大频率是 ssi_clk 频率的一半。这是为了允许移位控制逻辑在 sclk_out 的一个时钟沿上捕获数据，并在相反的沿上传播数据。

sclk_out 的频率可以从以下公式得出。

Fsclk_out = Fssi_clk / SCKDV

SCKDV 是一个可编程寄存器，可配置 0-65534 范围内的任何偶数。
如果 SCKDV = 0，则禁止 sclk_out。

9.6.3 接收和发送 FIFO 缓存

SSI 使用的 FIFO 缓存是内部 D 型触发器，深度配置为 8。由于串行规范，发送和接收 FIFO 缓存的宽度都固定为 32 位，这表明串行传输（数据帧）的长度可以为 4 到 32 位。小于 32 位的数据帧在写入发送 FIFO 缓存时必须右对齐。移位控制逻辑会自动右对齐接收 FIFO 缓存中的接收数据。

9.6.3.1 发送 FIFO

通过 AHB 写命令将发送 FIFO 加载到 SSI 数据寄存器（DR）。数据通过移位控制逻辑从发送 FIFO 中移出到发送移位寄存器中。当 FIFO 中的数据量小于或等于 FIFO 阈值时，发送 FIFO 会生成 FIFO 空中断请求（ssi_txe_intr）。通过可编程寄存器 TXFTLR 设置的阈值确定产生中断的 FIFO 水平。该阈值允许您向处理器提供早期指示，即发送 FIFO 几乎为空。如果您尝试将数据写入一个已经满的发送 FIFO，则会产生一个发送 FIFO 溢出中断

9.6.3.2 接收 FIFO

通过 AHB 读命令将数据从接收 FIFO 移出到 SSI 数据寄存器（DR）。接收 FIFO 由移位控制逻辑从接收移位寄存器加载。当 FIFO 中的数据量大于或等于 FIFO 阈值加 1 时，接收 FIFO 会生成 FIFO 满中断请求（ssi_rxf_intr）。通过可编程寄存器 RXFTLR 设置的阈值确定 FIFO 水平产生中断。

该阈值允许您向处理器提供早期指示，即接收 FIFO 即将满。当接收移位逻辑试图将数据加载到完全满的接收 FIFO 中时，将生成接收 FIFO 溢出中断（ssi_rxo_intr）。但是，该新接收的数据丢失。如果您尝试从空的接收 FIFO 中读取，则会生成接收 FIFO 下溢中断（ssi_rxu_intr）。这会警告处理器读取的数据无效。

9.6.4 DMA 操作

SSI 具有 DMA 功能。它具有与 DMA 控制器的握手接口，以请求和控制传输。AHB 总线用于执行与 DMA 之间的数据传输。SSI DMA 操作是通过通用方式设计的，以尽可能轻松地适合任何 DMA 控制器。要在 SSI 上使能 DMA 控制器接口，您必须写入 DMA 控制寄存器(DMACR)。将 1 写入 DMACR 寄存器的 TDMAE 位字段将使能 SSI 发送握手接口。将 1 写入 DMACR 寄存器的 RDMAE 位字段将使能 SSI 接收握手接口。

9.6.5 扩展 SPI 模式

SSI 使用 SSIC_SPI_MODE 配置参数支持 SPI 的双线模式，四线模式。当为此参数选择双线模式，四线模式时，txd，rxd 和 ssi_oe_n 信号的宽度分别更改为 2、4。因此，数据在多条线上移出/移入，从而提高了整体吞吐量。双信道 SPI，四信道 SPI 模式的功能类似，除了 txd，rxd 和 ssi_oe_n 信号的宽度不同。可以使用 CTRLR0.TMOD 字段选择操作模式（写/读）。

9.6.6 芯片内执行（XIP）模式

SSI 提供了直接用 AHB 总线进行读取操作的功能。这称为芯片内执行模式，其中 SSI 充当 SPI 存储器的存储器映射接口。通过选择配置参数 SSIC_XIP_EN，可以在 SSI 中使能 XIP 模式。这包括 AHB 接口上的额外信号 xip_en。该信号电平决定 AHB 传输是寄存器读写还是 XIP 读取。在 XIP 操作期间仅支持 AHB 读。如果 xip_en 信号为 1，则 SSI 希望在 AHB 接口上发出读取请求。该请求被转换为在串行接口上读取的 SPI。一旦接收到数据，它将返回到 AHB 接口。haddr 用于导出要在 SPI 接口上发送的地址。某些设备希望指令阶段在 XIP 传输期间存在。SSI 支持在 XIP 操作模式期间包含一些固定的指令集。

9.6.7 XIP 中的连续传输模式

SSI 收到 XIP 请求时，来自 AHB 接口的地址将直接传输到 SPI 接口。AHB 接口上的每个新传输（XIP 读取）都以相同的方式处理。因此，对于每个请求，必须将新地址发送到设备，从而导致系统延迟。

如果存储设备允许在 XIP 读取传输之间扩展片选信号，则可以将 SSI 编程为连续 XIP 模式以获得更高的性能。在这种模式下，主机通过确保不重新发送命令和地址，并且主机控制器无需等待这些突发之间的多余周期，将两个或多个 AHB 突发请求融合为一个 SPI 命令。

使能此功能后，一旦收到第一个 XIP 命令，SSI 便会以连续 XIP 模式运行。对于第一次 XIP 传输，地址在 SPI 接口上发送。接收到请求的数据后，SSI 继续保持从机处于选中状态，并且时钟（sclk_out）保持默认状态。对于 AHB 接口上的后续 XIP 传输，SSI 恢复时钟（sclk_out），命令和地址都不会传输到 SPI 接口上，并且不会立即从设备中获取数据（无多余周期）。

在连续读取模式下，不支持未定义的 INCR（hburst = 001）突发。

在连续传输期间，由于始终选择从设备，因此从设备上会消耗大量功率。为了避免这种情况，SSI 提供了一个配置选项，以使看门狗计时器在计数器用完后取消片选。

在以下情况下，SSI 可以取消选择从站：

在 XIP 接口上接收到非 XIP 命令（任何将 xip_en 驱动为 0 的 AHB 事务）。
当 AHB 事务处理到非连续地址时，将拉高片选，然后 SSI 发起新的 XIP 请求。
SSI 在 XIP_CNT_TIME_OUT 寄存器中指定的时间段内未检测到 AHB 接口上的任何 XIP 传输。

9.6.8 XIP 操作中的数据预取

使用 SSI 中的数据预取功能，控制器在当前 XIP 操作期间为连续突发预取数据。如果对连续地址发出下一个请求，则可以直接从 RXFIFO 读取数据，而不必等待新的地址和数据发送到设备。这样可以提高系统的整体性能。

要预取的数据量应等于最后一个 AHB 请求的突发长度或 FIFO 深度（以较低者为准）。例如，如果 AHB 定

义从地址 0x00 开始的突发长度为 16，则 SSI 提取 16 个数据以完成当前传输，然后再次提取 16 个以上的数据并将其保存在数据寄存器中。如果 AHB 总线再次请求从结束地址开始进行最后一次传输的数据，则其余数据将从 RX FIFO 本身发送到设备。同时，SSI 再次开始 XIP 传输，以预取下一个数据块。如果 AHB 主设备发出非连续地址，则将当前数据从 FIFO 中清除，然后控制器开始新的操作。

SSI 完成当前数据串并为下一个数据串预取数据时，可以接收新的 XIP 请求。在这种情况下，SSI 可以终止当前传输，或者根据地址将数字增加到要提取的数据长度。

如果使能了 XIP 预取，则不允许 AHB 请求增量传输未定义的长度（hburst = 3'b001）。

9.7 中断描述

9.7.1 发送 FIFO 空中断(ssi_txe_intr)

当发送 FIFO 等于或小于其阈值且需软件防止下溢出时置位。通过软件可编程寄存器设置的阈值确定产生中断的发送 FIFO 水平。当将数据写入发送 FIFO 缓存，使其超过阈值水平时。

9.7.2 发送 FIFO 溢出中断(ssi_txo_intr)

AHB 试图写入完全填满的发送 FIFO 时置位。置位时，将从 AHB 写入的数据丢弃。该中断将保持置位状态，直到读取发送 FIFO 溢出中断清除寄存器（TXOICR）。

9.7.3 接收 FIFO 完全中断(ssi_rxf_intr)

当接收 FIFO 等于或大于其阈值加 1 且需软件防止溢出时置位。通过软件可编程寄存器设置的阈值确定产生中断的接收 FIFO 水平。从接收 FIFO 缓存中读取数据，使其低于阈值水平时，硬件将清除此中断。

9.7.4 接收 FIFO 溢出中断(ssi_rxo_intr)

当接收逻辑在数据完全填满后试图将数据放入接收 FIFO 中时置位。置位时，新接收的数据将被丢弃。该中断将保持设置状态，直到读取接收 FIFO 溢出中断清除寄存器（RXOICR）。

9.7.5 接收 FIFO 下溢中断(ssi_rxu_intr)

AHB 访问空接收 FIFO 读取时置位。置位时，将从接收 FIFO 读回零。该中断将保持置位状态，直到读取接收 FIFO 下溢中断清除寄存器（RXUICR）。

9.7.6 组合中断请求(ssi_intr)

上述所有中断请求的屏蔽后的或结果。要屏蔽此中断信号，必须屏蔽所有其他 SSI 中断请求。

10 串行接口模块（SPI）

10.1 概述

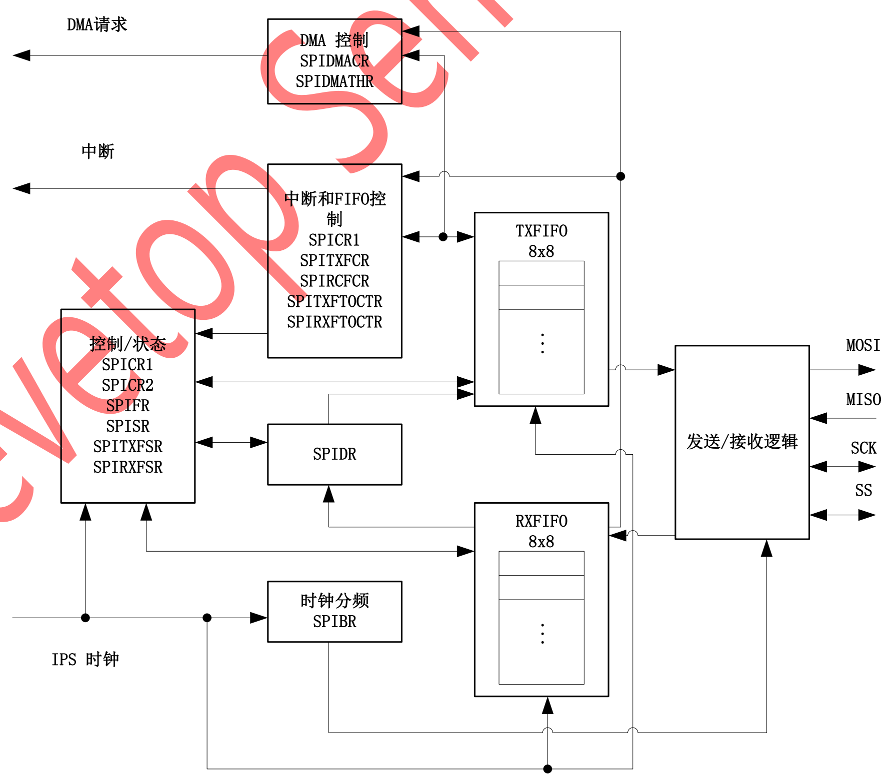
串行接口模块允许 MCU 与外设之间进行全双工，同步，连续的通信。软件可以拉起 SPI 标志位，或者 SPI 可以用中断来驱动。

10.2 特性

- 主模式和从模式
- 从选择输出
- 模式错误标志有 CPU 中断功能
- Doze 模式可以进行 SPI 操作
- 低功耗下可降低驱动
- Freescale SPI 以及 Texas 串行接口可用的可编程的接口操作
- 收发独立的 FIFO，均为 8 位宽以及 8 深度
- 4-16 位可编程数据页
- 在诊断和调试测试中，有内部可循环的测试操作
- 标准的机于 FIFO 的中断以及机于传输结束的中断
- 用 DMA 可以进行有效率的传输
- 调试时有可视的 TX 以及 RX FIFO
- 传输时序调整可用高速模式

10.3 框图

SPI 框架如下图所示：



图表 10-1：SPI 框图

10.4 工作模式

在下列三种工作模式下，SPI 的功能：

1．运行模式：运行模式是正常的工作模式。
2．瞌睡模式：瞌睡模式是一个低功耗可配置模式。
3．停止模式：在停止模式下，SPI 停止运行。

10.5 外部管脚

本章节是对信号的概述如下表所示。

表格 10-1：信号属性

名称	端口	功能	复位状态
MISO	SPIPORT1[0]	主数据输入/从数据输出	0
MOSI	SPIPORT1[1]	主数据输出/从数据输入	0
SCK	SPIPORT1[2]	串行时钟	0
SS	SPIPORT1[3]	从机选择脚	0

注意：禁用 SPI（SPE = 0）时，特定的 SPI 端口（MISO，MOSI，SCK，SS）是 GP I/O 端口。

10.5.1 MISO（主机输入/从机输出）

MISO 是两个 SPI 数据管脚之一。
在主机模式下，MISO 是数据输入。
在从机模式下，MISO 是高阻态数据输出，主机会将 SS 拉低。
在双向模式下，从机的 MISO 管脚是 SISO（从机输入/输出）。
在一个多主机系统中，所有的 MISO 连接在一起。

10.5.2 MOSI（主机输出/从机输入）

MOSI 是两个 SPI 数据管脚之一。
在主机模式下，MOSI 是数据输出。
在从机模式下，MOSI 是数据输入。
在双向模式下，主机的 MOSI 管脚是 MOMI 管脚（主机输出/主机输入）。
在一个多主机系统中，所有的 MOSI 连接在一起。

10.5.3 SCK（串行时钟）

SCK 管脚是一个串行时钟，用于主设备和从设备之间的同步发送。

在主机模式下，SCK 是一个输出管脚。

在从机模式下，SCK 是一个输入管脚。

在多主机系统中，所有的 SCK 连接在一起。

10.5.4 SS（从机选择）

在主机模式下，SS 管脚可以是：

模式错误输入

通用输入

通用输出

从机选择脚输出

在从机模式下，SS 管脚是从机选择脚输入。

10.6 内存映射及和寄存器

10.6.1 内存映射

表格 10-2：SPI 内存映射

偏移地址	位 7-0	访问权限
0x0000	SPI 波特率寄存器（SPIBR）	S/U
0x0001	SPI 帧寄存器（SPIFR）	S/U
0x0002	SPI 控制寄存器 1（SPICR1）	S/U
0x0003	SPI 控制寄存器 2（SPICR2）	S/U
0x0004	SPI RXFIFO 超时计数器寄存器（SPIRXFTOCTR）	S/U
0x0005	SPI TXFIFO 超时计数器寄存器（SPITXFTOCTR）	S/U
0x0006	SPI RXFIFO 控制寄存器（SPIRXFCR）	S/U
0x0007	SPI TXFIFO 控制寄存器（SPITXFCR）	S/U
0x0008	SPI SCK 后延迟寄存器（PSIASCDR）	S/U
0x0009	SPI SCK 前延迟寄存器（PSIBSCDR）	S/U
0x000A	SPI 端口数据方向寄存器（SPIDDR）	S/U
0x000B	SPI 上拉和低驱动寄存器（SPIPURD）	S/U
0x000C-D，F	SPI 传输计数器寄存器（SPITCNT）	S/U
0x000E	SPI 端口数据寄存器（SPIPORT）	S/U
0x0010	SS 端口中断寄存器（IRSP）	S/U
0x0011	保留	S/U
0x0012-13	SPI 数据寄存器（SPIDR）	S/U
0x0014	SPI RX FIFO 状态寄存器（SPIRXFSR）	S/U
0x0015	SPI TX FIFO 状态寄存器（SPITXFSR）	S/U
0x0016-17	SPI 状态寄存器（SPISR）	S/U
0x0018	SPI FIFO 调试控制寄存器（SPIFDCR）	S/U
0x0019	SPI 中断控制寄存器（SPIICR）	S/U
0x001A	SPI DMA 控制寄存器（SPIDMACR）	S/U
0x001B	SPI DMA 阈值寄存器（SPIDMATHR）	S/U
0x001C	SPI TX FIFO 调试寄存器（SPITXFDBGR）	S/U
0x001E	SPI RX FIFO 调试寄存器（SPIRXFDBGR）	S/U
0x0020-21	SPI 配置数据寄存器（SPICFGDATAR）	S/U

10.6.2 寄存器描述

10.6.2.1 SPI 波特率寄存器

偏移地址：0x0000 复位值：0x00

7	6	5	4	3	2	1	0
EOTFIE	SPPR[6:4]			FIRM	SPR[2:0]		
ro	rw			ro	rw		

图表 10-2：SPI 波特率寄存器（SPIBR）

比特位	名称	复位值	读写属性	功能说明
[7]	EOTFIE	0x0	RW	SPI EOTF 中断使能位 EOTFIE 位使能 EOTF 标志以生成中断请求。复位清除 SPIE。 0 = 禁止 EOTF 中断请求 1 = 使能 EOTF 中断请求
[6:4]	SPPR	0x0	RW	SPPR[6:4]以及 SPR[2:0]位选择 SPI 时钟分频，见表格 10-3：
[3]	FIRM	0x0	RW	SPI FIRM 数据发送使能位 0 = 禁止 FIRM 数据传输 1 = 使能 FIRM 数据传输 在从模式下将 FIRM 位使能时，SPI 从 SPICFGDATA 寄存器发送数据，SPIDR 寄存器数据无效。
[2:0]	SPR	0x0	RW	SPPR[6:4]以及 SPR[2:0]位选择 SPI 时钟分频，见表格 10-3：

表格 10-3：SPI 波特率选择（10MHz 模块时钟）

SPPR[6:4]	SPPR[2:0]	时钟分频器	波特率	SPPR[6:4]	SPPR[2:0]	时钟分频器	波特率
000	000	2	5MHz	100	000	10	1MHz
000	001	4	2.5MHz	100	001	20	0.5MHz
000	010	8	1.25MHz	100	010	40	0.25MHz
000	011	16	0.625MHz	100	011	80	125KHz
000	100	32	0.31MHz	100	100	160	62.5KHz
000	101	64	156.25KHz	100	101	320	31.25KHz
000	110	128	78.125KHz	100	110	640	15.625KHz
000	111	256	39.06KHz	100	111	1280	7.81KHz
001	000	4	2.5MHz	101	000	12	833.33KHz

SPPR[6:4]	SPPR[2:0]	时钟分频器	波特率	SPPR[6:4]	SPPR[2:0]	时钟分频器	波特率
001	001	8	1.25MHz	101	001	24	416.67KHz
001	010	16	0.625MHz	101	010	48	208.33KHz
001	011	32	0.31MHz	101	011	96	104.17KHz
001	100	64	156.25KHz	101	100	192	52.08KHz
001	101	128	78.125KHz	101	101	384	26.04KHz
001	110	256	39.06KHz	101	110	768	13.02KHz
001	111	512	19.53KHz	101	111	1536	6.51KHz
010	000	6	1.67MHz	110	000	14	714.29KHz
010	001	12	0.83MHz	110	001	28	357.14KHz
010	010	24	0.42MHz	110	010	56	178.57KHz
010	011	48	208.33KHz	110	011	112	89.29KHz
010	100	96	104.17KHz	110	100	224	44.64KHz
010	101	192	52.08KHz	110	101	448	22.32KHz
010	110	384	26.04KHz	110	110	896	11.16KHz
010	111	768	13.02KHz	110	111	1792	5.58KHz
011	000	8	1.25MHz	111	000	16	0.625MHz
011	001	16	0.625MHz	111	001	32	0.31MHz
011	010	32	0.31MHz	111	010	64	156.25KHz
011	011	64	156.25KHz	111	011	128	78.125KHz
011	100	128	78.125KHz	111	100	256	39.06KHz
011	101	256	39.06KHz	111	101	512	19.53KHz
011	110	512	19.53KHz	111	110	1024	9.77KHz
011	111	1024	9.77KHz	111	111	2048	4.88KHz

10.6.2.2 SPI 帧寄存器

偏移地址：0x0001 复位值：0x47

7	6	5	4	3	2	1	0
CONT	GTE	LBM	FFSEL	FMSZ[3:0]			
rw	rw	rw	rw	rw			

图表 10-3：SPI 帧寄存器（SPIFR）

比特位	名称	复位值	读写属性	功能说明
[7]	CONT	0x0	RW	连续的外设新跑选择使能 0 = 在传输中返回外设芯片选择信号高 1 = 在传输中保持外设芯片选择信号低知道 EOTF 置起
[6]	GTE	0x1	RW	保护时间使能 0 = 保护时间不使能；1 = 保护时间使能
[5]	LBM	0x0	RW	回环模式 0 = 普通模式；1 = 回环模式
[4]	FFSEL	0x0	RW	帧格式选择 0 = FREESCALE 帧格式选择 1 = TI 帧格式选择
[3:0]	FMSZ	0x7	RW	帧大小 FMSZ[3:0]控制帧数据长度从 4 位到 16 位。0x3 设置帧长度为 4 位，0xf 设置帧长度为 16 位。同时，0x0-0x2 将自动设置帧长度为 4 位。

10.6.2.3 SPI 控制寄存器 1

偏移地址：0x0002 复位值：0x04

7	6	5	4	3	2	1	0
SPIE	SPE	SWOM	MSTR	CPOL	CPHA	SSOE	LSBFE
rw	rw	rw	rw	rw	rw	rw	rw

图表 10-4：SPI 控制寄存器 1（SPICR1）

比特位	名称	复位值	读写属性	功能说明

比特位	名称	复位值	读写属性	功能说明
[7]	SPIE	0x0	RW	SPI 中断使能位 SPIE 位使能 SPIF 和 MODF 标志产生中断请求。复位会清除 SPIE。 0 = 不使能 SPIF 和 MODF 中断请求 1 = 使能 SPIF 和 MODF 中断请求
[6]	SPE	0x0	RW	SPI 系统使能位 SPE 位使能 SPI，SPI 端口管脚[3:0]用于 SPI 功能。当 SPE 清零时，SPI 系统被初始化，处于低能耗关闭状态。复位会清除 SPE。 0 = 不使能 SPI；1 = 使能 SPI
[5]	SWOM	0x0	RW	线或模式位 SWOM 位配置 SPI 端口管脚[3:0]的输出缓冲器作为开漏(Open-Drain)输出。SWOM 控制 SPI 端 口管脚[3:0]是 SPI 输出还是通用输出。复位会清除 SWOM。 0 = SPI 端口管脚[3:0]输出缓冲器是 CMOS 驱动 1 = SPI 端口管脚[3:0]输出缓冲器是开漏(Open-Drain)输出 SWOM 位在本章节中无影响。
[4]	MSTR	0x0	RW	主机模式位 MSTR 位选择 SPI 为主机模式或从机模式。 0 = 从机模式；1 = 主机模式
[3]	CPOL	0x0	RW	时钟极性选择位 CPOL 位选择反相或非反相 SPI 时钟。为了在 SPI 模块间传输数据，主从 SPI 模块必须配置为完全相同的 CPOL 值。复位会清除 CPOL。 0 = 高态有效时钟；空闲时 SCK 为低 1 = 低态有效时钟；空闲时 SCK 为高

比特位	名称	复位值	读写属性	功能说明
[2]	CPHA	0x1	RW	时钟相位位 CPHA 位延迟 SCK 时钟的第一个边沿。复位会清除 CPHA。 0 = 传输开始后 SCK 第一个边沿的 1/2 周期后为移位边沿 1 = 传输开始后 SCK 的第一个边沿为移位边沿 传输过程中(当 SS 为低时)改变 CPOL 或 CPHA 的值会导致错误结果。请在传输开始前（当 SS 为高时）改变 CPOL 和 CPHA 的值。
[1]	SSOE	0x0	RW	从机选择脚输出使能位 SSOE 位和 DDRSP3 位配置 SS 管脚作为通用输入或从机选择脚输出。复位会清除 SSOE。见 表格 10-4: 设置 SSOE 位会关闭模式错误检测功能。
[0]	LSBFE	0x0	RW	最低有效位优先使能位 LSBFE 使能 LSB 数据优先被传输。复位会清除 LSBFE。 0 = 数据 MSB 优先被传输 1 = 数据 LSB 优先被传输

表格 10-4：SS 管脚 I/O 配置

DDRSP3	SSOE	主机模式	从机模式
0	0	模式错误	从机选择脚输入
0	1	通用输入	从机选择脚输入
1	0	通用输出	从机选择脚输入
1	1	从机选择脚输出	从机选择脚输入

10.6.2.4 SPI 控制寄存器 2

偏移地址：0x0003 复位值：0x00

7	6	5	4	3	2	1	0
GT[5:0]						SPISDOZ	SPC0
rw						rw	rw

图表 10-5：SPI 控制寄存器 2（SPICR2）

比特位	名称	复位值	读写属性	功能说明
[7:2]	GT	0x0	RW	保护时间位 Guard_Time = $(GT[5:3]+1) * 2^{(GT[2:0]+1)}$
[1]	SPISDOZ	0x0	RW	睡眠模式下 SPI 停止位 当 CPU 处于睡眠模式时，DOZE 位会停止 SPI 时钟。复位会清除 SPISDOZ。 0 = 在睡眠模式下 SPI 运行 1 = 在睡眠模式下 SPI 停止
[0]	SPC0	0x0	RW	串行管脚控制位 0 0 = 普通管脚模式 1 = 双向管脚模式 SPC0 位使能双向管脚配置，见 表格 10-5:

表格 10-5：双向管脚配置

	管脚模式	SPC0	MSTR	MISO 管脚 [1]	MOSI 管脚 [1]	SCK 管脚 [1]	SS 管脚 [4]
A	普通	0	0	从数据输出	从数据输入	SCK 输入	从机选择脚输入
B			1	主数据输入	主数据输出	SCK 输出	MODF/GP 输入（DDRSP3 = 0）或 GP 输出（DDRSP3 = 1）
C	双向	0	0	从数据 I/O	GP[5] I/O	SCK 输入	从机选择脚输入
D			1	GP I/O	主数据 I/O	SCK 输出	MODF/GP 输入（DDRSP3 = 0）或 GP 输出（DDRSP3 = 1）

注意:

若 SPIDDR 位 0 = 1，SS = 0，MSTR = 0（A，C），从输出使能。

若 SPIDDR 位 1 = 1，MSTR = 1（B，D），主输出使能。

若 SPIDDR 位 2 = 1，MSTR = 1（B，D），SCK 输出使能。

若 SPIDDR 位 3 = 1，SPICR1 位（SSOE） = 1，MSTR = 1（B，D），SS 输出使能。若 SPIDDR 位 3 = 0，SSOE = 0，GP 输入使能。

GP = 通用

10.6.2.5 SPI RXFIFO 超时计数器寄存器

偏移地址：0x0004 复位值：0x40

7	6	5	4	3	2	1	0
RXFTOIE	RXFTOE	RXFTOCNT[5:0]					
rw	rw	rw					

图表 10-6：SPI RXFIFO 超时计数器寄存器（SPIRXFTOCTR）

比特位	名称	复位值	读写属性	功能说明
[7]	RXFTOIE	0x0	RW	超时中断使能 0 = RX FIFO 超时中断不使能 1 = RX FIFO 超时中断使能
[6]	RXFTOE	0x0	RW	超时功能使能 0 = RX FIFO 超时功能不使能 1 = RX FIFO 超时功能使能
[5:0]	RXFTOCNT	0x20	RW	设置 SPI RXFIFO 超时计数器数值。一旦 RXFIFO 不空，计数器开始工作。如果在计数器数到 0 前 RXFIFO 没有操作，RXF_TIMEOUT 将会置起

10.6.2.6 SPI TXFIFO 超时计数器寄存器

偏移地址：0x0005 复位值：0x40

7	6	5	4	3	2	1	0
TXFTOIE	TXFTOE	TXFTOCNT[5:0]					
rw	rw	rw					

图表 10-7：SPI TXFIFO 超时计数器寄存器（SPITXFTOCRT）

比特位	名称	复位值	读写属性	功能说明
[7]	TXFTOIE	0x0	RW	超时中断使能 0 = TX FIFO 超时中断不使能 1 = TX FIFO 超时中断使能
[6]	TXFTOE	0x0	RW	超时功能使能 0 = TX FIFO 超时功能不使能 1 = TX FIFO 超时功能使能

比特位	名称	复位值	读写属性	功能说明
[5:0]	TXFTOCNT	0x20	RW	设置 SPI TXFIFO 超时计数器数值。一旦 TXFIFO 不空，计数器开始工作。如果在计数器数到 0 前 TXFIFO 没有操作，TXF_TIMEOUT 将会置起

10.6.2.7 SPI RXFIFO 控制寄存器

偏移地址：0x0006　　复位值：0x00

7	6	5	4	3	2	1	0
RXFCLR	RXFOVIE	RXFUDIE	RXFSTHIE	保留	RXFSTH[2:0]		
rw	rw	rw	rw	ro	rw		

图表 10-8：SPI RXFIFO 控制寄存器（SPIRXFCR）

比特位	名称	复位值	读写属性	功能说明
[7]	RXFCLR	0x0	RW	RX FIFO 清除 该位写 1 复位 RXFIFO
[6]	RXFOVIE	0x0	RW	RX FIFO 溢出中断使能 0 = RX FIFO 溢出中断不使能 1 = RX FIFO 溢出中断使能
[5]	RXFUDIE	0x0	RW	RX FIFO 下溢出中断使能 0 = RX FIFO 下溢出中断不使能 1 = RX FIFO 下溢出中断使能
[4]	RXFSTHIE	0x0	RW	RX FIFO 服务阈值中断使能 0 = RX FIFO 服务阈值中断不使能 1 = RX FIFO 服务阈值中断使能
[3]	保留	0x0	RO	---
[2:0]	RXFSTH	0x0	RW	RX FIFO 服务阈值 一旦有效的传输数据数量大于或者等于这个阈值设置，RX FIFO 服务中断标志会置起。 数据数量 = RXFSTH[2:0]+1

10.6.2.8 SPI TXFIFO 控制寄存器

偏移地址：0x0007 复位值：0x07

7	6	5	4	3	2	1	0
TXFCLR	TXFOVIE	TXFUDIE	TXFSTHIE	保留	TXFSTH[2:0]		
rw	rw	rw	rw	ro	rw		

图表 10-9：SPI TXFIFO 控制寄存器（SPITXFCR）

比特位	名称	复位值	读写属性	功能说明
[7]	TXFCLR	0x0	RW	TX FIFO 清除 该位写 1 复位 TXFIFO
[6]	TXFOVIE	0x0	RW	TX FIFO 溢出中断使能 0 = TX FIFO 溢出中断不使能 1 = TX FIFO 溢出中断使能
[5]	TXFUDIE	0x0	RW	TX FIFO 下溢出中断使能 0 = TX FIFO 下溢出中断不使能 1 = TX FIFO 下溢出中断使能
[4]	TXFSTHIE	0x0	RW	TX FIFO 服务阈值中断使能 0 = TX FIFO 服务阈值中断不使能 1 = TX FIFO 服务阈值中断使能
[3]	保留	0x0	RO	---
[2:0]	TXFSTH	0x7	RW	TX FIFO 服务阈值 一旦有效的传输数据数量小于或者等于这个阈值设置，TX FIFO 服务中断标志会置起。 数据数量 = TXFSTH[2:0]+1

10.6.2.9 SPI SCK 后延迟寄存器

偏移地址：0x0008 复位值：0x00

7	6	5	4	3	2	1	0
ASCDE	PASCD[6:4]			保留	ASCD[2:0]		
rw	rw			ro	rw		

图表 10-10：SPI SCK 后延迟寄存器（SPIASCDR）

比特位	名称	复位值	读写属性	功能说明
[7]	ASCDE	0x0	RW	SCK 后延迟使能 0 = SCK 后延迟不使能 1 = SCK 后延迟使能

比特位	名称	复位值	读写属性	功能说明
[6:4]	PASCD	0x0	RW	SPI SCK 后延迟预选择位 PASCD[6:4]以及 ASCD[2:0]位选择 SPI SCK 后延迟分频。
[3]	保留	0x0	RO	---
[2:0]	ASCD	0x0	RW	SPI SCK 后延迟位 ASCD = （PASCD[6:4]+1）*$2^{(ASCD[2:0]+1)}$； t_T = 0.5*SCK+ASCD

10.6.2.10 SPI SCK 前延迟寄存器

偏移地址：0x0009　　　　复位值：0x00

7	6	5	4	3	2	1	0
BSCDE	PBSCD[6:4]			保留	BSCD[2:0]		
rw	rw			ro	rw		

图表 10-11：SPI SCK 前延迟寄存器（PSIBSCDR）

比特位	名称	复位值	读写属性	功能说明
[7]	BSCDE	0x0	RW	SCK 前延迟使能 0 = SCK 前延迟不使能 1 = SCK 前延迟使能
[6:4]	PBSCD	0x0	RW	SPI SCK 前延迟预选择位 PBSCD[6:4]以及 BSCD[2:0]位选择 SPI SCK 前延迟分频。
[3]	保留	0x0	RO	---
[2:0]	BSCD	0x0	RW	SPI SCK 前延迟位 BSCD = （PBSCD[6:4]+1）*$2^{(BSCD[2:0]+1)}$； t_L = 0.5*SCK+BSCD

10.6.2.11 SPI 端口数据方向寄存器

偏移地址：0x000A 复位值：0x00

7	6	5	4	3	2	1	0
PPS[3:0]				DDRSP[3:0]			
rw				rw			

图表 10-12：SPI 端口数据方向寄存器（SPIDDR）

比特位	名称	复位值	读写属性	功能说明
[7:4]	PPS	0x0	RW	端口优先级选择位 PPS［3:0］选择 SPIPORT 端口的功能的优先级。 0 = 普通模式优先于 GPIO 模式 1 = GPIO 功能优先于普通模式
[3:0]	DDRSP	0x0	RW	数据输入输出定向位 DDRSP［3:0］位控制 SPIPORT 管脚的数据输入输出方向。复位会清除 DDRSP［3:0］。 0 = 对应的管脚配置作为输入 1 = 对应的管脚配置作为输出 在从机模式下，DDRSP3 无意义或不起作用。 在主机模式下，DDRSP3 和 SSOE 位决定 SPI 端口管脚 3 是模式错误输入，通用输入输出或是一个从选择脚输出。 **注意：**当 SPI 使能时，（SPE = 1），MISO，MOSI 和 SCK 管脚： ●当 SPI 需要其为输入时，其输入功能跟对应的 DDRSP 位的状态无关。 ●当 SPI 需要其为输出时，其输出功能必须有对应的 DDRSP 位置位。 PPS［3:0］和 DDRSP［3:0］对应管脚[SS，SCK，MOSI/MOMi，MISO/SISO]

10.6.2.12 SPI 上拉和低驱动寄存器

偏移地址：0x000B　　　　复位值：0x01

7	6	5	4	3	2	1	0
HS	PSW	DBLRXDR	ENDING	MSPD[1:0]		保留	PUPSP
rw	rw	rw	rw	rw		ro	rw

图表 10-13：SPI 上拉和低驱动寄存器（SPIPURD）

比特位	名称	复位值	读写属性	功能说明
[7]	HS	0x0	RW	从机高速模式使能 当 HS 置位，SPI 从机将在采样沿移出数据，而采样时机和普通模式一致。 0 = 从机高速模式不使能 1 = 从机高速模式使能
[6]	PSW	0x0	RW	位转换 转换 MOSI 到 MISO，以及 MISO 到 MOSI。 0 = 转换不使能；1 = 转换使能
[5]	DBLTXDR	0x0	RW	双重 TX 数据寄存模式使能位 在从模式中，该位设置双重 TX 数据寄存模式，用于减小两帧数据间的延时。在这种模式下，至少需要在 TX FIFO 中预存一组数据。
[4]	ENDING	0x0	RW	大小端选择位 0 = 小端；1 = 大端
[3:2]	MSPD	0x0	RW	SPI 主机采样延时 指定采样时能延迟的系统时钟边沿数量。
[1]	保留	0x0	RO	---
[0]	PUPSP	0x1	RW	SPI 端口上拉使能位 0 = SPIPORT 位 [3:0] 对应的管脚上拉关闭 1 = SPIPORT 位 [3:0] 对应的管脚上拉开启

10.6.2.13 SPI 传输计数器寄存器

偏移地址：0x000C~0x000D，0x000F　　　　　　复位值：0x000000

31	30	29	28	27	26	25	24
SPITCNT[31:24]							
rw							
23	22	21	20	19	18	17	16
SPITCNT[23:16]							
rw							
15	14	13	12	11	10	9	8
SPITCNT[15:8]							
rw							

图表 10-14：SPI 传输计数器寄存器（SPITCNT）

比特位	名称	复位值	读写属性	功能说明
[31:8]	SPITCNT	0x0	RW	SPI 传输计数器寄存器 在主机模式中，帧传送时，如果 SPIFR 中 CONT 位已配置，该计数器向下减计数到 1。一旦计数器减到 0，并且两外一帧数据已发送，SS 端口将为高，计数器值将重新加载。

10.6.2.14 SPI 端口数据寄存器（SPIPORT）

偏移地址：0x000E 复位值：0x00

7	6	5	4	3	2	1	0
保留				PORTSP[3:0]			
ro				rw			

图表 10-15：SPI 端口数据寄存器（SPIPORT）

比特位	名称	复位值	读写属性	功能说明
[7:4]	保留	0x0	RO	---
[3:0]	PORTSP	0x0	RW	SPI 端口数据位 被写入 SPIPORT 的数据只有在被作为通用输出时才会驱动管脚。 读一个输入（DDRSP 位清除）返回管脚的电平值；读一个输出（DDRSP 位置位）返回管脚输出驱动的输入电平。 当管脚配置作为 SPI 输出时，对 PORTSP [3:0] 任何管脚写入都不会改变管脚状态。 SPIPORT I/O 功能取决于 SPICR1 的 SPE 位状态和 SPIDDR 的 DDRSP 位状态。 PORTSP［3:0］对应管脚[SS，SCK，MOSI/MOMi，MISO/SISO]

10.6.2.15 SS 端口中断寄存器（IRSP）

偏移地址：0x0010 复位值：0x00

7	6	5	4	3	2	1	0
SSIE	保留		SSF	SSLPR	SSD	SSPA	
rw	ro		ro	rw	ro	ro	

图表 10-16：SS 端口中断寄存器（IRSP）

比特位	名称	复位值	读写属性	功能说明
[7]	SSIE	0x0	RW	SS 端口中断使能信号 可读写的 SSIE 位使能中断请求。如果 SSIE 已配置，在以下情况，SPI 将产生一个中断请求： 对应位的 SS 标志寄存器（SSF）已设置或者将要设置 对应位的端口信号是低并且端口配置成电平敏感操作 清除 SSIE 会清除对应 SS 端口的中断请求。复位清除 SSIE 位 0 = 禁止来自 SS 管脚的中断请求。 1 = 使能来自 SS 管脚的中断请求
[6:5]	保留	0x0	RO	---
[4]	SSF	0x0	RO	SS 位标志位 当 SS 端口配置成边沿触发时，SSFR 位表示该边沿已被侦测到。复位清除 SSF 位。 0 = 被选择的边沿未被侦测到 1 = 被选择的边沿被侦测到
[3]	SSLPR	0x0	RW	SS 端口电平极性位 如果 SS 配置成电平敏感，SSLPR 中配置 1 表示高电平有效。清除 SSLPR 表示低电平有效。复位清除 SSLPR。 0 = 对应的 SS 位低电平有效 1 = 对应的 SS 位高电平有效
[2]	SSD	0x0	RO	SS 端口数据位 该位只读。只有 SSD 可以反映 SS 端口目前的状况。写 SSD 位无用。写操作正常结束。复位不影响该位。

比特位	名称	复位值	读写属性	功能说明
[1:0]	SSPA	0x0	RW	可读写的 SSPA 区域可以配置 SS 端口的电平检测，上升沿以及下降沿检测。 电平敏感配置可以决定外部端口逻辑 0 或者逻辑 1 表现出来的有效的中断请求。电平敏感中断输入中断不会独占。来保证电平敏感中断请求是知道的，中断源必须保持信号知道软件得到回应。如果需要用 SS 端口中断唤醒 stop 模式，电平敏感选项必须选择。 如果选择边沿触发，则不需要维持中断信号。一个端口配置成边沿触发总是会被检测到，无论配置成输入或者输出。 2'b00 = 管脚 SS 电平敏感 2'b01 = 管脚 SS 上升沿触发 2'b10 = 管脚 SS 下降沿触发 2'b11 = 管脚 SS 上升下降沿均触发

10.6.2.16 SPI 数据寄存器

偏移地址：0x0013　　复位值：0x00

15	14	13	12	11	10	9	8
SPIDR[15:8]							
rw							

偏移地址：0x0012　　复位值：0x00

7	6	5	4	3	2	1	0
SPIDR[7:0]							
rw							

图表 10-17：SPI 数据寄存器

比特位	名称	复位值	读写属性	功能说明
[15:0]	SPIDR	0x0	RW	SPIDR 既是 SPI 数据的输入寄存器也是其输出寄存器。写入 SPIDR 将填充 TX FIFO，而从 SPIDR 读取将减少 RX FIFO。

10.6.2.17 SPIRX FIFO 状态寄存器

偏移地址：0x0014 复位值：0x00

7	6	5	4	3	2	1	0
保留	RXNXTP[2:0]			RXFFCT[3:0]			
ro	ro			ro			

图表 10-18：SPI RX FIFO 状态寄存器（SPIRXFSR）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6:4]	RXNXTP	0x0	RO	RX 下一个指针 该位表示指向 RX FIFO 中下一个将要传送的数据的指针
[3:0]	RXFFCT	0x0	RO	RX FIFO 计数器 该位表示 RX FIFO 的数据计数器

10.6.2.18 SPITX FIFO 状态寄存器

偏移地址：0x0015 复位值：0x00

7	6	5	4	3	2	1	0
保留	TXNXTP[2:0]			TXFFCT[3:0]			
ro	ro			ro			

图表 10-19：SPI TX FIFO 状态寄存器（SPITXFSR）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6:4]	TXNXTP	0x0	RO	TX 下一个指针 该位表示指向 TX FIFO 中下一个将要传送的数据的指针
[3:0]	TXFFCT	0x0	RO	TX FIFO 计数器 该位表示 TX FIFO 的数据计数器

10.6.2.19 SPI 状态寄存器

偏移地址：0x0017 复位值：0x00

15	14	13	12	11	10	9	8
TXFTO	TXFOVF	TXFUDF	TXFSER	RXFTO	RXFOVF	RXFUDF	RXFSER
r/w1c	r/w1c	r/w1c	ro	r/w1c	r/w1c	r/w1c	ro

偏移地址：0x0016 复位值：0x05

7	6	5	4	3	2	1	0
SPIF	FLOAT	EOTF	MODF	TXFFULL	TXFEMP	RXFFULL	RXFEMP
ro	r/w1c	r/w1c	ro	ro	ro	ro	ro

图表 10-20：SPI 状态寄存器（SPISR）

比特位	名称	复位值	读写属性	功能说明
[15]	TXFTO	0x0	R/W1C	TX FIFO 超时 0 = TX FIFO 没有超时；1 = TX FIFO 有超时 该位写 1 清 0，或者写 SPIDR 也可以清除该位。
[14]	TXFOVF	0x0	R/W1C	TX FIFO 溢出标志 0 = TX FIFO 未溢出；1 = TX FIFO 溢出
[13]	TXFUDF	0x0	R/W1C	TX FIFO 下溢出标志 0 = TX FIFO 没有下溢出 1 = TX FIFO 有下溢出
[12]	TXFSER	0x0	RO	TX FIFO 服务标志 0 = TX FIFO 中数据多余 TXFSTH 1 = TX FIFO 中数据少于或者等于 TXFSTH
[11]	RXFTO	0x0	R/W1C	RX FIFO 超时 0 = RX FIFO 没有超时；1 = RX FIFO 有超时 该位写 1 清 0，或者写 SPIDR 也可以清除该位。
[10]	RXOVF	0x0	R/W1C	RX FIFO 溢出标志 0 = RX FIFO 未溢出；1 = RX FIFO 溢出
[9]	RXFUDF	0x0	R/W1C	RX FIFO 下溢出标志 0 = RX FIFO 没有下溢出 1 = RX FIFO 有下溢出
[8]	RXFSER	0x0	RO	RX FIFO 服务标志 0 = RX FIFO 中数据多余 RXFSTH 1 = RX FIFO 中数据少于或者等于 RXFSTH

比特位	名称	复位值	读写属性	功能说明
[7]	SPIF	0x0	RO	SPI 完成标志 每次单次传输后都会设置 SPIF 标志。通过读取 SPIF，然后访问 SPIDR 清除 SPIF。 0 = 单次传输尚未完成或没有传输 1 = 单次传输已完成
[6]	FLOST	0x0	R/W1C	帧丢失 当 SPI 在从模式是，并且 TX FIFO 中没有有效数据，如果主机开始一个传输，SPI 会返回最后收到的数据给主机并且 FLOST 会置起。如果 FLOSTIE 也配置了，FLOST 将会产生一个中断请求。 0 = 没有帧丢失；1 = 帧丢失
[5]	EOTF	0x0	R/W1C	传输结束标志 当所有 TX FIFO 中的数据传输完成时，EOTF 会置起。 0 = 传输没有结束或者没有传输 1 = 传输结束 该位写 1 清 0，或者写 SPIDR 也会清除该位。
[4]	MODF	0x0	RO	模式错误标志 当 SPI 为主机模式，SS 管脚是低电平，且 SS 管脚被配置作为模式错误输入时，MODF 标志会被置位。若 SPIE 也被设置，MODF 会产生一个中断请求。模式错误会清除 SPE，MSTR 和 DDRSP[2:0]位。读取 SPISR 后对 SPICR1 进行写操作将会清除 MODF。复位会清除 MODF。 0 = 无模式错误；1 = 模式错误
[3]	TXFFULL	0x0	RO	TX FIFO 满标志 0 = TX FIFO 不满；1 = TX FIFO 已满
[2]	TXFEMP	0x1	RO	TX FIFO 空标志 0 = TX FIFO 不空；1 = TX FIFO 空
[1]	RXFFULL	0x0	RO	RX FIFO 满标志 0 = RX FIFO 不满；1 = RX FIFO 已满
[0]	RXFEMP	0x1	RO	RX FIFO 空标志 0 = RX FIFO 不空；1 = RX FIFO 空

10.6.2.20 SPI FIFO 调试控制寄存器

偏移地址：0x0018 复位值：0x00

7	6	5	4	3	2	1	0
保留	TXFIDX[6:4]			保留	RXFIDX[2:0]		
ro	rw			ro	rw		

图表 10-21：SPI FIFO 调试控制寄存器（SPIFDCR）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6:4]	TXFIDX	0x0	RW	TX FIFO 索引 该位制定 TX FIFO 到 SPITXFDBGR 的数据索引
[3]	保留	0x0	RO	---
[2:0]	RXFIDX	0x0	RW	RX FIFO 索引 该位制定 RX FIFO 到 SPIRXFDBGR 的数据索引

10.6.2.21 SPI 中断控制寄存器

偏移地址：0x0019 复位值：0x00

7	6	5	4	3	2	1	0
保留	FLOSTIE	保留	MODFIE	保留			
ro	rw	ro	rw	ro			

图表 10-22：SPI 中断控制寄存器（SPIICR）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6]	FLOSTIE	0x0	RW	FLOST 中断使能 0 = FLOST 中断不使能 1 = FLOST 中断使能
[5]	保留	0x0	RO	---
[4]	MODFIE	0x0	RW	FLOST 中断使能 0 = FLOST 中断不使能 1 = FLOST 中断使能
[3:0]	保留	0x0	RO	---

10.6.2.22 SPI DMA 控制寄存器

偏移地址：0x001A 复位值：0x00

7	6	5	4	3	2	1	0
保留						TXDMAE	RXDMAE
ro						rw	rw

图表 10-23：SPI DMA 控制（SPIDMACR）

比特位	名称	复位值	读写属性	功能说明
[7:2]	保留	0x0	RO	---
[1]	TXDMAE	0x0	RW	TX FIFO DMA 请求使能 0 = TX DMA 请求不使能 1 = TX DMA 请求使能
[0]	RXDMAE	0x0	RW	RX FIFO DMA 请求使能 0 = RX DMA 请求不使能 1 = RX DMA 请求使能

10.6.2.23 SPI DMA 阈值寄存器

偏移地址：0x001B 复位值：0x00

7	6	5	4	3	2	1	0
保留	TXDMATH[6:4]			保留	RXDMATH[2:0]		
ro	rw			ro	rw		

图表 10-24：SPI DMA 阈值寄存器（SPIDMATHR）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6:4]	TXDMATH	0x0	RW	TX DMA 阈值 该位指定 TX FIFO 的数据数量阈值。一旦 TX FIFO 中的数据数量低于这个阈值，并且 SPIDMACR 中 TXDMAE 位置起，SPI 会送一个 TX DMA 的请求给 DMA。 数据数量 = TXDMATH[2:0]
[3]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[2:0]	RXDMATH	0x0	RW	RX DMA 阈值 该位指定 RX FIFO 的数据数量阈值。一旦 RX FIFO 中的数据数量低于这个阈值，并且 SPIDMACR 中 RXDMAE 位置起，SPI 会送一个 RX DMA 的请求给 DMA。 数据数量 = RXDMATH[2:0]+1

10.6.2.24 SPI TX FIFO 调试寄存器

偏移地址：0x001C　　　　复位值：0x00

7	6	5	4	3	2	1	0
SPITXFGBG[7:0]							
ro							

图表 10-25：SPI TX FIFO 调试寄存器（SPITXFDBGR）

比特位	名称	复位值	读写属性	功能说明
[7:0]	SPITXFGBG	0x0	RO	SPITXFGBGR 在 dubug 中提供 TX FIFO 一个可视性。改寄存器只读，并且不能修改。读该位不影响 TX FIFO 的状态。

10.6.2.25 SPI RX FIFO 调试寄存器

偏移地址：0x001E　　　　复位值：0x00

7	6	5	4	3	2	1	0
SPIRXFGBG[7:0]							
ro							

图表 10-26：SPI RX FIFO 调试寄存器（SPIRXFDBGR）

比特位	名称	复位值	读写属性	功能说明
[7:0]	SPIRXFGBG	0x0	RO	SPIRXFGBGR 在 dubug 中提供 RX FIFO 一个可视性。改寄存器只读，并且不能修改。读该位不影响 RX FIFO 的状态。

10.6.2.26 SPI 配置数据寄存器

偏移地址：0x0021　　复位值：0x00

15	14	13	12	11	10	9	8
SPICFGDATA[15:8]							
rw							

偏移地址：0x0020　　复位值：0x00

7	6	5	4	3	2	1	0
SPICFGDATA[7:0]							
rw							

图表 10-27：SPI 配置数据寄存器（SPICFGDATAR）

比特位	名称	复位值	读写属性	功能说明
[15:0]	SPICFGDATA	0x0	RW	SPICFGDATA 对于 SPI 配置数据是输入也是输出寄存器

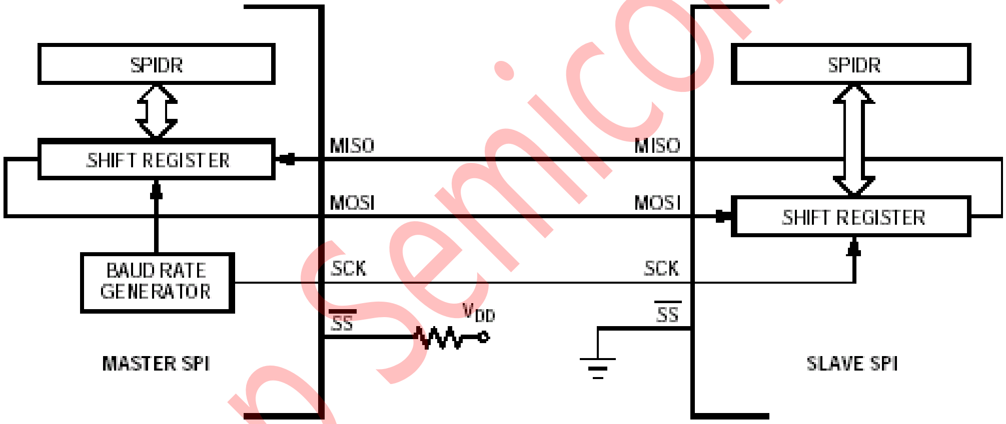
10.7 功能描述

SPI 模块支持串行外围设备与微型控制器（MPU）之间全双工、同步和串行的数据传输。软件能够轮询 SPI 状态标志，也可以基于中断完成 SPI 操作。

设置 SPICR1 寄存器中的 SPE 位可以使能 SPI 功能，SPI 的 4 个管脚为：

从机选择脚（SS）
串行时钟（SCK）
主机出/从机进（MOSI）
主机进/从机出（MISO）

当清除 SPE 位，可以通过 SPIDDR 寄存器控制 SS，SCK，MOSI 和 MISO 管脚作为通用输入输出管脚。MOSI 和 MISO 管脚将主设 SPI 的移位寄存器和从设的移位寄存器相连。相连的移位寄存器形成一个分布式寄存器。在一个 SPI 传输过程中，会基于主机的 SCK 时钟将主机和从机寄存器的数据交换。写入主设 SPIDR 寄存器的数据是从设 SPIDR 寄存器的输出数据。交换后，从主设 SPIDR 寄存器读取的数据是来自从设的数据。



图表 10-28：全双工操作

10.7.1 主机模式

对 SPICR1 寄存器的 MSTR 位进行设置，可使 SPI 进入主机模式。只有主 SPI 才能发起传输操作，对主 SPIDR 写入会开始一个传输。在移位寄存器是空的情况下，数据在会向移位寄存器传输，在主 SCK 时钟的控制下转移出 MOSI 管脚。写入 SPIDR 后，SCK 时钟在 1.5 个 SCK 周期后开始。

SPIBR 内的 SPR[2:0]和 SPR[6:4]位控制波特率发生器，决定了移位寄存器的速度。SCK 管脚是 SPI 时钟输出管脚，通过 SCK 管脚，主机的波特率发生器控制从机的移位寄存器。

SPICR1 寄存器的 MSTR 位，SPICR2 寄存器的 SPC0 位控制着 MOSI 和 MISO 两个数据管脚功能。一般情况下，SS 管脚是非高电平状态的输入管脚。对 SPIDDR 寄存器的 DDRSP3 位进行设置，可以将 SS 作为一个

输出管脚。SPICR1 寄存器的 DDRSP3 位和 SSOE 位能配置 SS 作为一个通用输入输出、默认管脚模式、或从机选择脚。

SS 输出在每次发送期间变低，而在 SPI 处于空闲状态时变高。将主 SS 输入驱动为低电平会将 SPISR 中的 MODF 标志置 1，指示模式错误。可能有多个主试图同时驱动 MOSI 和 SCK 线路。模式错误会清除 MISO，MOSI（或 MOMI）和 SCK 管脚的数据方向位，使其成为输入。模式错误还会清除 SPICR1 中的 SPE 和 MSTR 位。如果还将 SPIE 位置 1，则 MODF 标志将产生一个中断请求。

10.7.2 从机模式

将 SPICR1 中的 MSTR 位清零会将 SPI 置于从机模式。SCK 管脚是来自主机的 SPI 时钟输入，而 SS 管脚是从机选择输入。为了进行传输，必须将 SS 管脚驱动为低电平并保持低电平直到传输完成。

SPICR2 中的 MSTR 位和 SPC0 位控制数据管脚 MOSI 和 MISO 的功能。SS 输入还控制 MISO 管脚。如果 SS 为低电平，则移位寄存器中的 MSB 在 MISO 管脚上移出。如果 SS 为高电平，则 MISO 管脚处于高阻抗状态，而从机忽略 SCK 输入。

注意：使用具有全双工功能的外围设备时，请勿同时启用两个驱动同一 MISO 输出线的接收器。只要只有一个从机驱动主输入线，则多个从机可能会同时接收相同的传输。

如果 SPICR1 中的 CPHA 位清零，则 SCK 输入上的奇数沿将锁存 MOSI 管脚上的数据。偶数沿将数据移至 SPI 移位寄存器的 LSB 位置，并将 MSB 移至 MISO 管脚。

如果 CPHA 位置 1，则 SCK 输入上的偶数沿将锁存 MOSI 管脚上的数据。奇数沿将数据移入 SPI 移位寄存器的 LSB 位置，并将 MSB 移至 MISO 管脚。

第八次移位后传输完成。接收到的数据传输到 SPIDR，并在 SPISR 中设置 SPIF 标志。

10.7.3 FIFO 操作

当选择小于 16 位或 8 位的数据大小时，用户必须右对齐写入发送 FIFO 的数据。发送逻辑忽略未使用的位。小于 16 或 8 位的接收数据会在接收缓冲区中自动右对齐。

10.7.3.1 发送 FIFO

通用发送 FIFO 是一个 8 位宽，8 深度的先进先出存储缓存。CPU 通过写入 SPI 数据（SPIDR）寄存器将数据写入 FIFO，并将数据存储在 FIFO 中，直到被传输逻辑读出为止。

当配置为主机或从机时，并行数据在串行转换之前分别写入发送 FIFO，并通过 SPI Tx 管脚分别传输到连接的从机或主机。在从机模式下，SPI 在每次主机发出请求传输数据。如果发送 FIFO 为空，并且主机启动，则从机将传输最后一次接收到的数据。应注意确保根据需要将有效数据存储在 FIFO 中。当 FIFO 数据数小于设置阈值时，可以将 SPI 配置为生成中断或 DMA 请求。

10.7.3.2 接收 FIFO

通用接收 FIFO 是一个 8 位宽，8 深度的先进先出存储缓存，来自串行接口的接收数据被存储在缓存中，直到被 CPU 读取为止，CPU 通过读取 SPIDR 寄存器读取 FIFO。

10.7.4 传输格式

SPICR1 寄存器内的 CPHA 和 CPOl 位将会对串行时钟方向和相位 4 种组合进行四选一。主 SPI 设备和通信从设备的时钟相位和极性必须相同。

10.7.4.1 当 CPHA = 1 时的传输格式

某些外围设备要求，在第一个 SCK 边沿出现之后，从设才能传输数据的 MSB 位。当 CPHA 位被置后，SPI 主设备会进行 SCK 半个时钟周期的同步延迟。在传输开始之前，先产生第一个 SCK 边缘。第一个时钟边缘会让从机将其 MSB 向主机的 MISO 管脚发送。第二个边缘和以后的偶数边缘会锁存数据。第三个边缘和以后的奇数边缘将锁定的数据由从机转移到主机移位寄存器，并将数据输出至主机 MOSI 管脚上。

在第 16 个和最后一个 SCK 边缘后：

原来在主机 SPIDR 寄存器内的数据保存在从机 SPIDR 内。

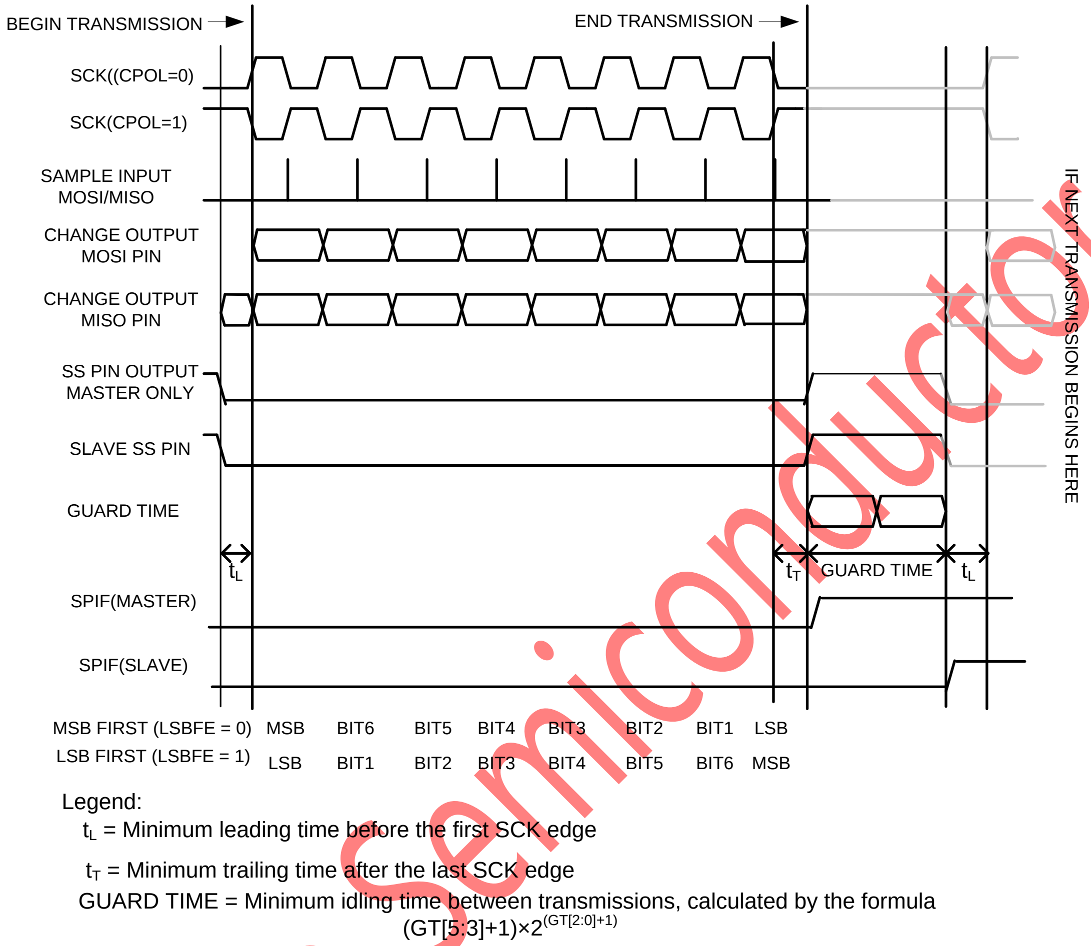
原来在从机 SPIDR 寄存器内的数据现保存在主机 SPIDR 内。

SCK 时钟停止，SPISR 内的 SPIF 标志被置，传输已完成。若 SPCR1 内的 SPIE 位被设置，SPIF 会产生一个中断请求。

下图显示了 CPHA 位置 1 时的发送时序。主器件的 SS 管脚必须为高电平或配置为不影响 SPI 的通用输出。当 CPHA = 1 时，从 SS 线可以在字节之间保持低电平。这种格式非常适合具有单个主机和单个从机驱动 MISO

数据线的系统。

SPIF 中断请求在每次单次传输结束时发出。



图表 10-29：SPI 时序格式 1（CPHA = 1）

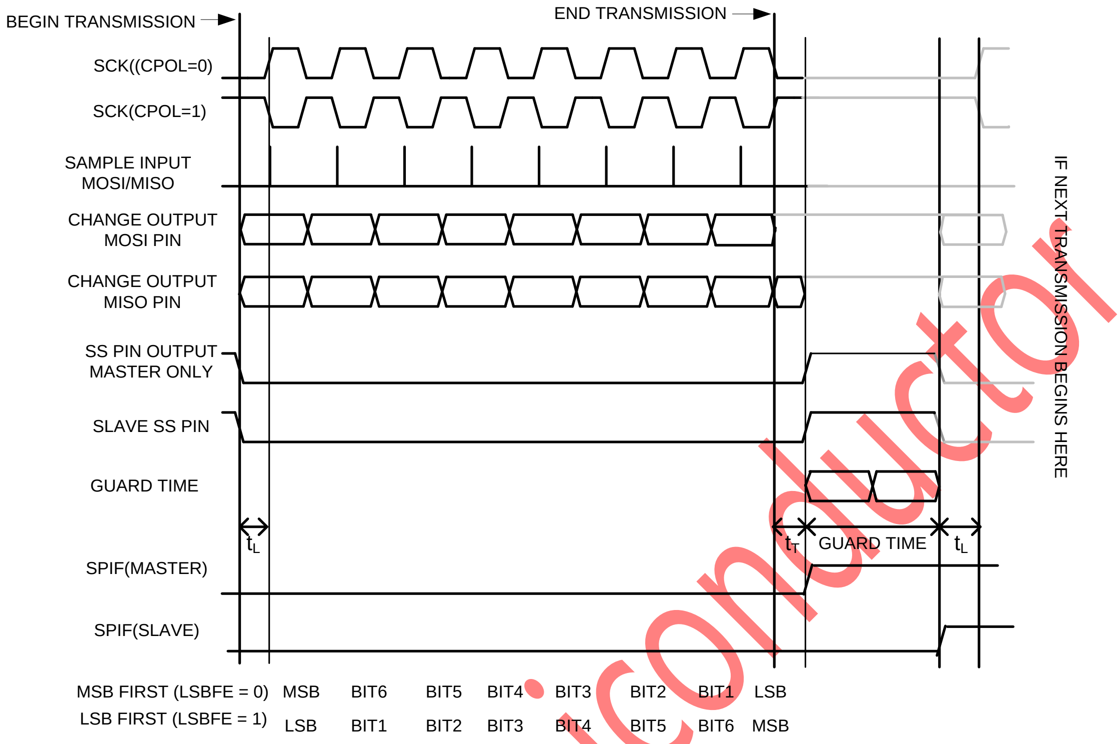
10.7.4.2 当 CPHA = 0 时的发送方式

在某些外围设备内，从机被选之后，从机数据的 MSB 位也同时有效。当 CPHA 位为“0”，在传输开始后，SPI 主设会延迟其第一个 SCK 边缘半个 SCK 周期。第一个时钟边沿和以后的所有奇数时钟边沿会锁存从机数据。SCK 的偶数边沿会将从机数据转移到主机移位寄存器，将主机数据输出至主机 MOSI 管脚上。

在第 16 个和最后一个 SCK 边缘之后：
主 SPIDR 中的数据位于从 SPIDR 中。从 SPIDR 中的数据位于主 SPIDR 中。

SCK 时钟停止并且 SPISR 中的 SPIF 标志被置位，指示发送完成。如果 SPCR1 中的 SPIE 位置 1，则 SPIF 会生成一个中断请求。

下图显示了 CPHA 位清零时的发送时序。主器件的 SS 管脚必须为高电平或配置为不影响 SPI 的通用输出。当 CPHA = 0 时，从 SS 管脚必须取反并在字节之间重新置位。



Legend:

t_L = Minimum leading time before the first SCK edge

t_T = Minimum trailing time after the last SCK edge

GUARD TIME = Minimum idling time between transmissions, calculated by the formula $(GT[5:3]+1)\times 2^{(GT[2:0]+1)}$

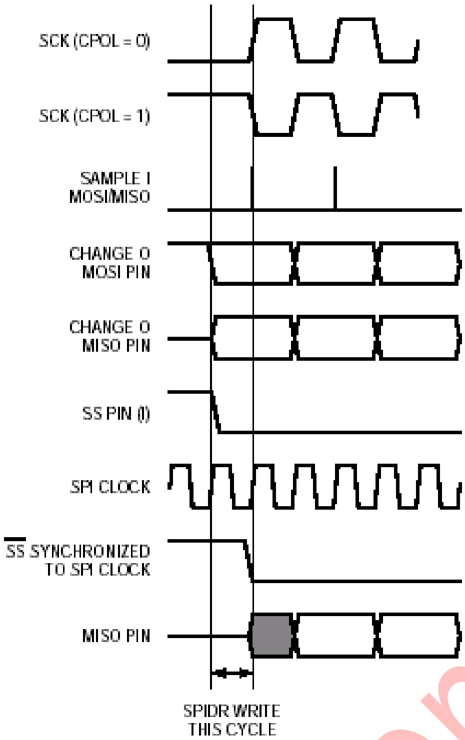
图表 10-30：SPI 时序格式 1（CPHA = 1）

注意：主设和从设之间的时钟偏差可能会导致数据丢失

当：CPHA = 0，并且，波特率是 SPI 时钟 2 分频，并且主 SCK 频率是从 SPI 时钟频率的一半，并且在同步的 SS 信号刚变低之前，软件将数据写入从 SPIDR。

同步的 SS 信号与 SPI 时钟同步。下图给出了一个示例，其中同步的 SS 信号几乎要延迟一个完整的 SPI 时钟周期。

当从机的同步 SS 为高电平时，即使 SS 管脚已经为低电平，也允许写入。当主机对 MISO 线进行采样时，写操作可以更改 MISO 管脚。当主机采样传输的第一位时，它可能不稳定，因此发送到主机的字节可能已损坏。



图表 10-31：主/从时钟偏差导致传输错误

同样，如果从机产生延迟写入，其状态机可能没有时间复位，从而导致其错误地从主机接收了一个字节。

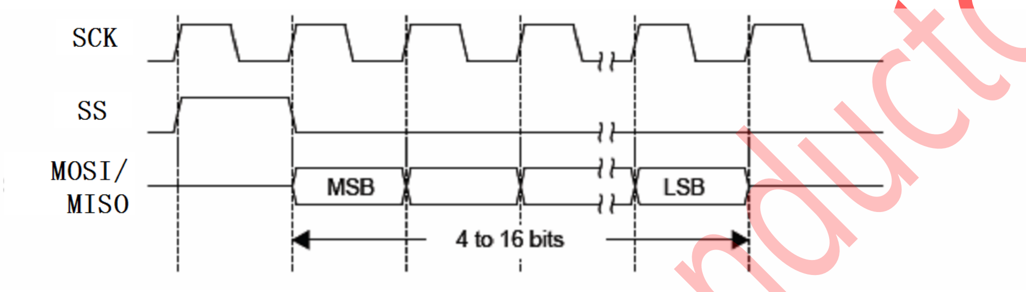
当 SCK 频率为从 SPI 时钟频率的一半时，很可能出现此错误。在其他波特率下，SCK 偏斜不超过一个 SPI 时钟，并且在同步 SS 信号和第一个 SCK 沿之间有更多时间。例如，当 SCK 频率为从 SPI 时钟频率的四分之一时，在 SS 的下降沿和 SCK 沿之间有两个 SPI 时钟。

只要不发生另一次较晚的 SPIDR 写操作，就可以正确地传输与从设备之间的后续字节。

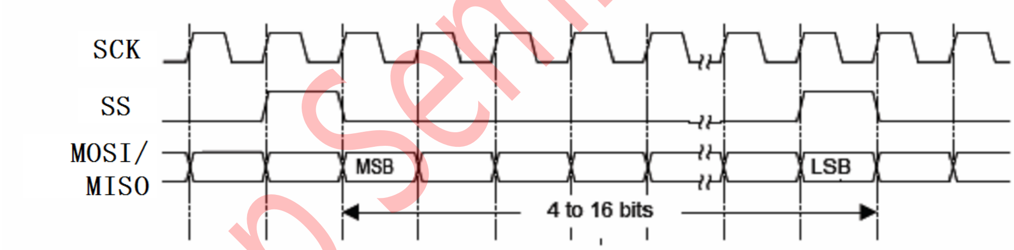
10.7.4.3 德州仪器（TI）同步串行帧格式

在这种模式下，只要 SSI 空闲，SCK 和 SS 就会被强制为低电平，并且发送数据线 SPI Tx 处于三态。一旦发送 FIFO 的底部包含数据，SS 就会在一个 SCK 周期内被脉冲为高电平。要发送的值也从发送 FIFO 传输到发送逻辑的串行移位寄存器。在 SSIClk 的下一个上升沿，4 至 16 位数据帧的 MSB 在 SSITx 管脚上移出。同样，片外串行从器件将接收到的数据的 MSB 移到 SSIRx 管脚上。

SPI 和片外串行从器件都在 SCK 的每个下降沿将每个数据位输入其串行移位器。锁存 LSB 之后，在 SCK 的第一个上升沿将接收到的数据从串行移位器传输到接收 FIFO。



图表 10-32：TI 单数据传输



图表 10-33：TI 连续传输

10.7.5 SPI 波特率

波特率发生器将 SPI 时钟分频以产生 SPI 波特时钟。SPIBR 中的 SPPR[6：4]和 SPR[2：0]位选择 SPI 时钟除数：

$$\text{SPI clock divisor} = (\text{SPPR} + 1) \times 2^{(\text{SPR} + 1)}$$

其中：

SPPR = 写入 SPPR [6:4]位的值

SPR = 写入 SPR [2:0]位的值

波特率发生器仅在 SPI 处于主机模式并正在发送时才处于工作状态。否则，分频器将无效以减少 IDD 电流。

10.7.6 从机选择（SS）输出

从机选择输出功能在发送期间自动将 SS 管脚驱动为低电平以选择外部设备，在空闲期间自动将其驱动为高电平以取消选择外部设备。选择 SS 输出时，SS 输出管脚连接到外部设备的 SS 输入管脚。

仅在主机模式下，将 SPICR1 中的 SSOE 位和 SPIDDR 中的 DDRSP[3]位置 1 会将 SS 管脚配置为从机选择输出。将 SSOE 位置 1 将禁用模式错误功能。

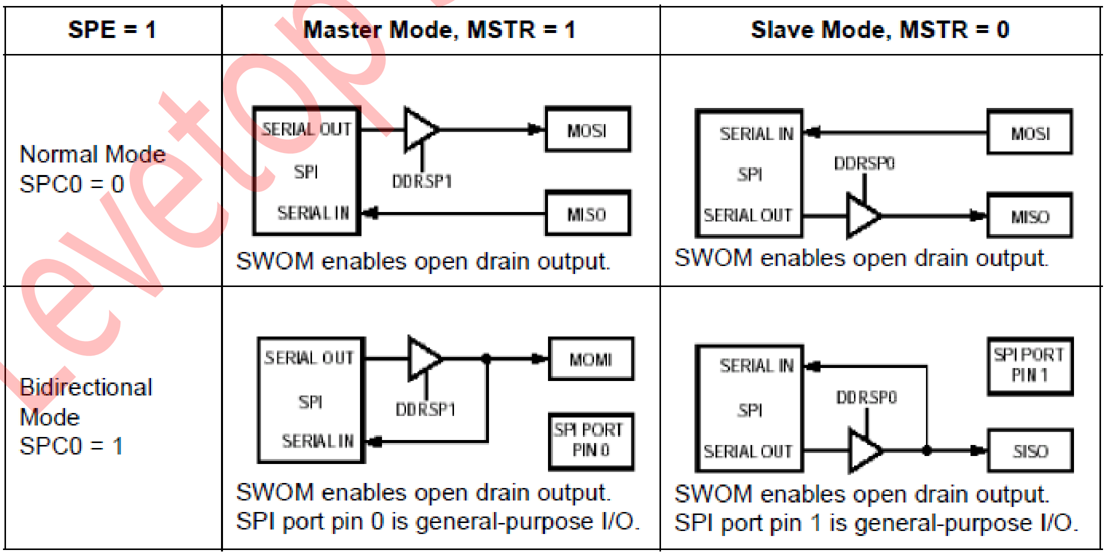
注意：在多主系统中使用从选择输出功能时，请当心。模式故障功能不适用于检测主设之间的系统错误。

10.7.7 双向模式

将 SPICR1 中的 SPC0 位置 1 选择双向模式。SPI 仅使用一个数据管脚作为与外部设备的接口。MSTR 位决定使用哪个管脚。在主机模式下，MOSI 管脚为主机输出/主机输入管脚 MOMI。在从模式下，MISO 管脚是从输出/从输入管脚 SISO。主机模式下的 MISO 管脚和从机模式下的 MOSI 管脚是通用 I/O 管脚。

每个数据 I/O 管脚的方向取决于其数据方向寄存器位。配置为输出的管脚是移位寄存器的输出。配置为输入的管脚是移位寄存器的输入，并且将从移位寄存器输出的数据丢弃。

SCK 管脚在主机模式下为输出，在从机模式下为输入。
在主模式下，SS 管脚可以是输入或输出，而在从模式下，它始终是输入。
在双向模式下，模式错误不会清除 DDRSP0（SISO 管脚的数据方向位）。



图表 10-34：普通模式和双向模式

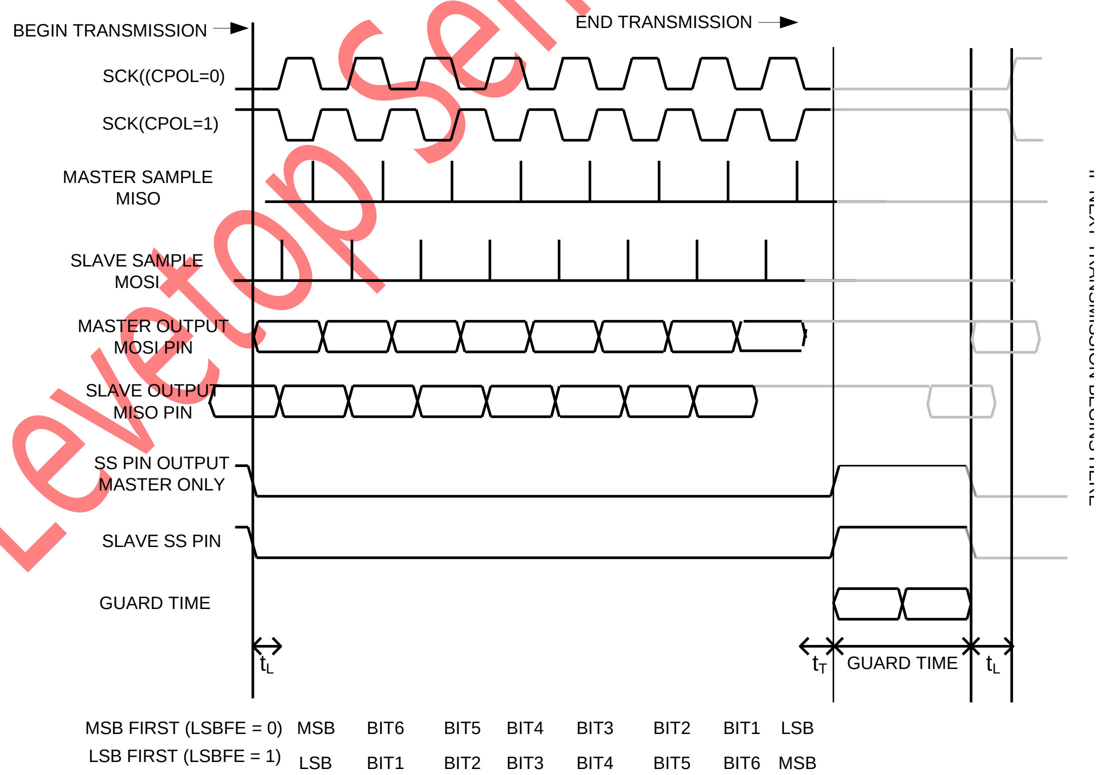
10.7.8 DMA 操作

SPI 外设通过单独的通道为 DMA 控制器提供接口，用于发送和接收。SPI 的 DMA 操作通过 SPI DMA 控制（SPIDMACR）寄存器启用。使能 DMA 操作时，当相关的 FIFO 满足传输要求时，SPI 在接收或发送通道上发出 DMA 请求。对于接收通道，只要接收 FIFO 中的数据数大于或等于 SPIDMATHR 设置的阈值或 SPIRXFTOCTR 设置的 RXF 超时计数器递减计数到零，就会发出 DMA 请求。对于发送通道，只要发送 FIFO 中的数据数量小于或等于 SPIDMATHR 设置的阈值或 SPITXFTOCTR 设置为 TXF 超时计数器的计数降低到零，就发送单个传输请求。DMA 控制器根据 DMA 信道的配置自动处理请求。为了使能接收通道的 DMA 操作，应将 DMA 控制（SPIDMACR）寄存器的 RXDMAE 位置 1。为了使能发送通道的 DMA 操作，应将 SPIDMACR 的 TXDMAE 位置 1。

10.7.9 高速模式

在高速模式下，与在正常 SPI 模式下相比，在 SCK 周期内，主机采样和从机都更早地移出数据，以允许器件焊盘和电路板走线的延迟。随着 SCK 周期随着波特率的增加而减小，这些延迟将成为 SCK 周期的重要部分。

对于主机，当配置为主机时，高速模式通过 SPIURD 寄存器中的 HS 位启用，而采样点延迟由 SPIPURD 寄存器中的 MSPD[1：0]设置。对于从机，当配置为从机时，通过 SPIURD 寄存器中的 HS 位启用高速模式。如果设置了 HS，则 SPI 从机将在移入数据沿将数据移出，而移入数据时序与正常模式相同。为了使高速模式正确运行，您必须彻底分析 SPI 链路时序预算。



图表 10-35：高速模式（CPHA = 0）

10.7.10 低功耗模式选项

这部分是对低功耗备选模式的介绍。

10.7.10.1 工作模式

清除 SPICR1 内的 SPE 位，会使得 SPI 关闭，处于低功耗状态。SPI 寄存器可供访问，但 SPI 时钟关闭。

10.7.10.2 瞌睡模式

SPI 在瞌睡模式的运行取决于 SPISDOZE 在 SPICR2 内的设置。

若 SPISDOZE 清除，SPI 在瞌睡模式下正常运行

若 SPISDOZ 被设置，SPI 时钟会停止运行，SPI 会进入瞌睡模式下的低功耗状态。

进入瞌睡模式时，正在处理的任何主机传输都会停止，退出瞌睡模式时恢复。

若主机寄存驱动从机 SCK 管脚，正在处理的任何主机传输继续。从机会和主机 SCK 时钟保持同步。

注意：尽管从移位寄存器可以接收 MOSI 数据，但是它不能将数据传输到 SPIDR 或将 SPIF 标志设置为瞌睡或停止模式。如果从机在空闲状态下进入瞌睡模式，而在空闲状态下退出瞌睡模式，则 SPIF 保持清零状态，并且不会发生向 SPIDR 的传输。

10.7.10.3 停止模式

SPIDOZ 位设置时，SPI 在停止模式下的运行和其在瞌睡模式下的运行一样。

10.7.11 复位

复位会使 SPI 寄存器初始化为初始状态，在复位后对 SPIDR 寄存器写入前，来自于从机的传输要么是不确定的，要么是在复位前，从主机最后接收的字节。复位后读 SPIDR 会得到 0。

10.8 中断描述

SPI 有以下几种中断方式:

表格 10-6：SPI 中断请求源

中断源	标志	使能位
模式错误	MODF	MODFIE
传输完成	EOTF	SPIE
帧丢失	FLOST	FLOSTIE
TXFIFO 超时	TXFTO	TXFTOIE
TXFIFO 溢出	TXFOVF	TXFOVIE
TXFIFO 下溢	TXFUDF	TXFUDIE
TXFIFO 服务	TXFSER	TXFSTHIE
RXFIFO 超时	RXFTO	RXFTOIE
RXFIFO 溢出	RXFOVF	RXFOVIE
RXFIFO 下溢	RXFUDF	RXFUDIE
RXFIFO 服务	RXFSER	RXFSTHIE

10.8.1 模式错误（MODF）中断

当主机 SPI 的 SS 管脚驱动为低电平且 SS 管脚配置作为模式错误输入管脚时，MODF 被置位。如果 SPIE 也被置位了。MODF 会产生中断请求。模式错误会清除 SPE，MSTR，和 DDRSP [2:0] 位。读 SPISR 后对 SPICR1 写会清除 MODF。复位会清除 MODF。

10.8.2 EOT 中断（EOTF）

发送 TX FIFO 中的所有数据时，将设置 EOTF。如果还将 SPIE 位置 1，则 EOTF 会生成一个中断请求。通过向该位写入 1 清除 EOTF 或由 CPU 或 DMA 填充 TX FIFO。复位清除 EOTF。

10.8.3 帧丢失中断（FLOST）

当 SPI 在从模式下，并且 TXFIFO 中没有有效数据，如果主机开始一个传输，SPI 会返回给主机一个冗余数据，并且将 FLOST 置起。如果 FLOSTIE 是置起的，FLOST 将产生一个中断请求。清楚中断在该位写 1，reset 将清除 FLOST。

10.8.4 TXFIFO 超时中断（TXFTO）

一旦 TXFIFO 非空，计数器会开始计数，如果计数器数到 0 时对 TXFIFO 没有操作，TXFTO 将置起。计数器是按照 SCK 计数的，TXFTO 写 1 清 0。

10.8.5 TXFIFO 溢出中断（TXFOVF）

任何引起 TXFIFO 数据数量大于 8 的写操作会引起 TXFOVF 中断。TXFOVF 写 1 清 0。

10.8.6 TXFIFO 下溢中断（TXFUDF）

任何引起 TXFIFO 数据数量小于 0 的写操作将会引起 TXFUDF 中断。TXFUDF 写 1 清 0。

10.8.7 TXFIFO 服务中断标志（TXFSER）

当 TXFIFO 中的数据数量低于或等于 SPITXFCR 规定的数量时将会引起 TXFSER 中断。

10.8.8 TXFIFO 超时中断

一旦 TXFIFO 非空，计数器会开始计数，如果计数器数到 0 时对 TXFIFO 没有操作，TXFTO 将置起。计数器是按照 SCK 计数的，TXFTO 写 1 清 0。

10.8.9 RXFIFO 超时中断（RXFTO）

一旦 RXFIFO 非空，计数器会开始计数，如果计数器数到 0 时对 RXFIFO 没有操作，RXFTO 将置起。计数器是按照 SCK 计数的，RXFTO 写 1 清 0。

10.8.10 RXFIFO 溢出中断（RXFOVF）

任何引起 RXFIFO 数据数量大于 8 的写操作会引起 RXFOVF 中断。RXFOVF 写 1 清 0。

10.8.11 RXFIFO 下溢中断（RXFUDF）

任何引起 RXFIFO 数据数量小于 0 的写操作将会引起 RXFUDF 中断。RXFUDF 写 1 清 0。

10.8.12 RXFIFO 服务中断标志（RXFSER）

当 RXFIFO 中的数据数量低于或等于 SPIRXFCR 规定的数量时将会引起 RXFSER 中断。

11 通用异步收发器（UART）

11.1 概述

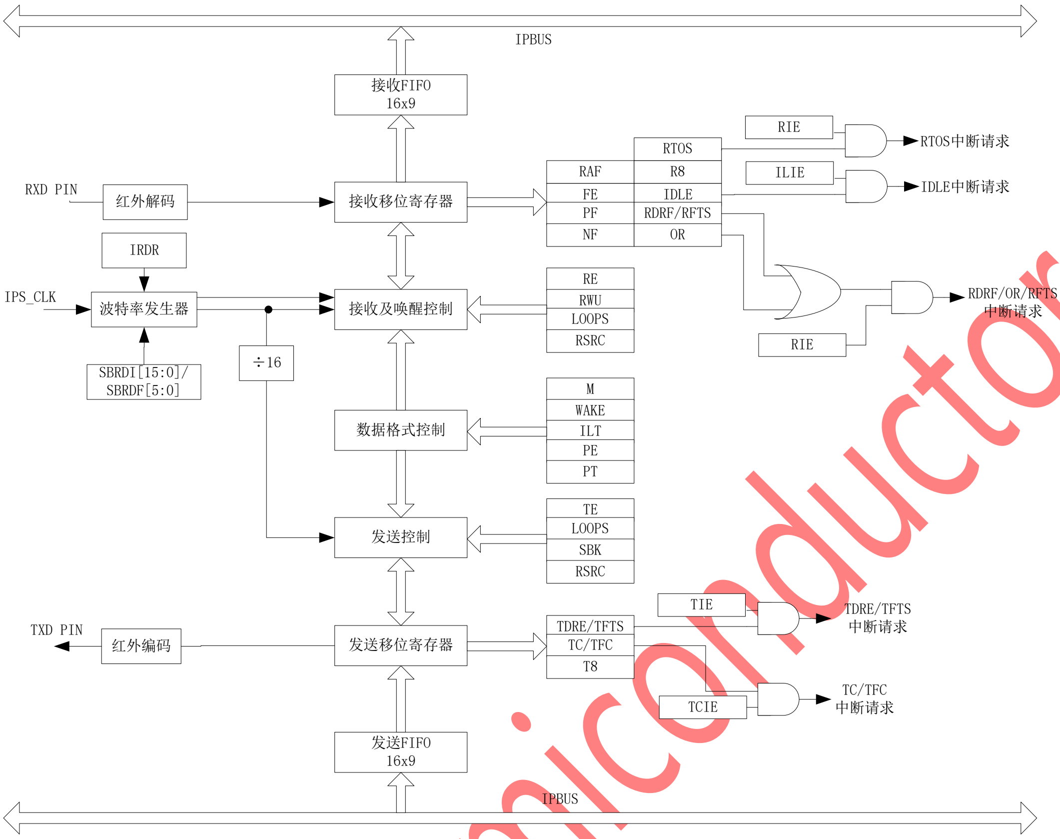
通用异步收发器（UART）允许与外围设备或其他微控制器单元(MCU)进行异步串行通信。

11.2 特性

UART 的特性包括：

- 支持全双工操作
- 支持 NRZ（非归 0）通信格式
- 13 位波特率选择
- 可编程 8 位、9 位的数据字长度
- 独立使能的发送器和接收器
- 独立的发送器和接收器中断请求
- 发送器输出极性可编程
- 2 种接收器唤醒方式
- 空闲线唤醒
- 地址标记唤醒
- 8 种中断方式
- 发送器空
- 发生完成
- 接收器满
- 接收器空闲输入
- 接收器溢出
- 噪声错误
- 帧错误
- 奇偶校验错误
- 接收器帧错误侦测
- 硬件奇偶校验检查
- 1/16 位时间噪声检查
- 支持通用输入输出功能
- 支持低速串行 IR 接口功能，兼容 IrDA(最高可达 115.2Kbit/s)
- 独立的 16x9 发送和接收 FIFO，以减少 CPU 中断服务的调用
- FIFO 触发级别为 1/8、1/4、1/2、3/4 和 7/8
- 支持 DMA 传输
- 支持硬件流控功能

11.3 框图



图表 11-1：UART 框图

11.4 工作模式

UART 模块在正常模式、特殊模式以及仿真模式下是完全一样的。UART 有两个低功耗工作模式，瞌睡和停止。

注意：工作模式是运行的正常模式，等待指令不会影响 UART 的运行。

11.4.1 瞌睡模式

当 UART 上拉和驱动控制寄存器（UARTPURD）内的 UARTSDOZ 位被设置后，瞌睡指令就会停止 UART 时钟，使得 UART 处于低功耗状态。瞌睡指令不会影响 UART 寄存器状态。当一个内部或外部中断请求使得 CPU 退出瞌睡模式时，任何此前停止在该瞌睡模式下的发送或接收操作将被恢复。若是由复位退出瞌睡模式，则 UART 被复位，原来在进行的发送和接收被中止，不再恢复。参阅 UART 上拉和驱动控制寄存器（UARTPURD）。

当 UARTSDOZ 位被清除时，瞌睡指令的执行对 UART 不会有影响。正常模式的运行继续，任何 UART 中断会使 CPU 退出瞌睡模式。

停止模式，停止指令会停止 UART 时钟，使得 UART 进入低功耗模。停止指令不会影响 UART 寄存器状态。当一个外部中断请求使得 CPU 退出停止模式时，任何此前停止在该停止模式下的发送或接收操作将被恢复若是由复位退出停止模式，则 UART 被复位，原来在进行的发送和接收被中止，不再恢复。

11.5 外部管脚

表格 11-1 为信号概述。

表格 11-1：信号属性

名称	基本功能	接口	初始状态	默认上拉状态
RXD	数据接收管脚	UARTPORT0	0	禁止
TXD	数据发送管脚	UARTPORT1	0	禁止
RTSN	请求发送管脚	UARTRTS	0	禁止
CTSN	清除发送管脚	UARTCTS	0	禁止

11.5.1 RXD

该信号是 UART 接收器管脚，当没有配置为接收时也可被用作 GPIO。

11.5.2 TXD

该信号是 UART 发送器管脚，当没有配置为发送时也可被用作 GPIO。

11.5.3 RTSN

该信号低有效时表示 UART 告知外设已经准备好可以接收数据。

11.5.4 CTSN

该信号低有效时表示外设请求 UART 停止发送数据。

11.6 内存映射和寄存器

11.6.1 内存映射

表格 11-2 是 UART 模块寄存器内存映射表。

表格 11-2：通用串行接口模块内存映射

偏移地址	位 7-0	访问权限
0x0001	UART 波特率寄存器高位(UARTBDRH)	S/U
0x0000	UART 波特率寄存器低位(UARTBDRL)	S/U
0x0003	UART 控制寄存器 1(UARTCR1)	S/U
0x0002	UART 控制寄存器 2(UARTCR2)	S/U
0x0005	UART 状态寄存器 1(UARTSR1)	S/U
0x0004	UART 状态寄存器 2(UARTSR2)	S/U
0x0007	UART 数据寄存器高位(UARTDRH)	S/U
0x0006	UART 数据寄存器低位(UARTDRL)	S/U
0x0009	UART 上拉和驱动控制寄存器(UARTPURD)	S/U
0x0008	UART 端口数据寄存器(UARTPORT)	S/U
0x000b	UART 数据方向寄存器(UARTDDR)	S/U
0x000a	UART 小数波特率寄存器(UARTBRDF)	S/U
0x000d	UART 测试寄存器(UARTTR)	S/U
0x000c	UART 红外控制寄存器(UARTIRCR)	S/U
0x000f	UART 红外分频寄存器(UARTIRDR)	S/U
0x000e	UART FIFO 控制寄存器(UARTFCR)	S/U
0x0011	UART FIFO 状态寄存器(UARTFSR)	S/U
0x0010	UART DMA 控制寄存器(UARTDCR)	S/U
0x0013	UART FIFO 控制寄存器 2(UARTFCR2)	S/U
0x0012	UART 接收 FIFO 超时寄存器(UARTRXTOCTR)	S/U
0x0015	UART FIFO 状态寄存器 2(UARTFSR2)	S/U
0x0014	UART 流控控制寄存器(UARTFCTRL)	S/U

11.6.2 寄存器描述

11.6.2.1 UART 波特率寄存器

偏移地址：0x0001 复位值：0x00

7	6	5	4	3	2	1	0
SBRDI15	SBRDI14	SBRDI13	SBRDI12	SBRDI11	SBRDI10	SBRDI9	SBRDI8
rw	rw	rw	rw	rw	rw	rw	rw

图表 11-2：UART 波特率寄存器高(UARTBDRH)

比特位	名称	复位值	读写属性	功能说明
[7:0]	SBRDI	0x0	RW	UART 整数部分波特率分频系数 15~8 位

偏移地址：0x0000 复位值：0x04

7	6	5	4	3	2	1	0
SBRDI17	SBRDI6	SBRDI5	SBRDI4	SBRDI3	SBRDI2	SBRDI1	SBRDI0
rw	rw	rw	rw	rw	rw	rw	rw

图表 11-3：UART 波特率寄存器低(UARTBDRL)

比特位	名称	复位值	读写属性	功能说明
[7:0]	SBRDI	0x4	RW	UART 整数部分波特率分频系数 7~0 位

偏移地址：0x000a 复位值：0x00

7	6	5	4	3	2	1	0
保留		SBRDF5	SBRDF4	SBRDF3	SBRDF2	SBRDF1	SBRDF0
ro		rw	rw	rw	rw	rw	rw

图表 11-4：UART 小数波特率寄存器(UARTBRDF)

比特位	名称	复位值	读写属性	功能说明
[7:6]	保留	---	RO	---
[5:0]	SBRDF	0x0	RW	UART 小数部分波特率分频系数 5~0 位

BRD = BRDI + BRDF = Fsys / (16 * UART baudrate)

SBRDI = 取整（BRD）；

SBRDIF = 取整（BRDF * 64 + 0.5）；

这些读/写位控制 UART 波特率：

$$UART\ baudrate = Fsys / (16 * SBRD)$$

其中：

1≤SBRDI≤65535；0≤SBRDF≤63

$$SBRD = SBRDI + (SBRDF / 64)$$

注意：波特率发生器被禁用，直到在 UARTCR 2 中的 TE 位或 RE 位在复位后第一次被设置。当 SBRDI[15:0] = 0 和 SBRDF[5:0] = 0 时，波特率发生器也会被禁用。

写入 UARTBRDH 和 UARTBRDF 没有任何效果，除非也写入 UARTBRDL。写入 UARTBRDH 和 UARTBRDF 的值将会被存在临时缓存中，直到 UARTBRDL 也被写入，才会真正生效。

11.6.2.2 UART 控制寄存器 1

偏移地址：0x0003 复位值：0x00

7	6	5	4	3	2	1	0
LOOPS	WOMS	RSRC	M	WAKE	ILT	PE	PT
rw	rw	rw	rw	rw	rw	rw	rw

图表 11-5：UART 控制寄存器 1(UARTCR1)

比特位	名称	复位值	读写属性	功能说明
[7]	LOOPS	0x0	RW	环路选择位 0 = SCI 以正常方式操作 1 = SCI 以环路方式操作
[6]	WOMS	0x0	RW	线或方式选择位 0 = TxD 和 RxD 引脚作为输出时为 CMOS 驱动 1 = TxD 和 RxD 引脚作为输出时开漏(Open-Drain)
[5]	RSRC	0x0	RW	接收器输入源选择位 0 = 在 LOOPS 为 1 的条件下，接收器输入连到发送器输出 1 = 在 LOOPS 为 1 的条件下，接收器输入连到 RxD 引脚
[4]	M	0x0	RW	数据格式的方式选择位 0 = 帧长为 10bit，1 个起始位，8 个数据位，1 个停止位

比特位	名称	复位值	读写属性	功能说明
				0 = 帧长为 11bit，1 个起始位，9 个数据位，1 个停止位
[3]	WAKE	0x0	RW	唤醒方式位 0 = 空闲线为唤醒接收器的条件 1 = 地址标记为唤醒接收器的条件
[2]	ILT	0x0	RW	空闲线类型位 0 = 接收器从起始位之后开始对空闲帧的位数计算 1 = 接收器从停止位之后开始对空闲帧的位数计算
[1]	PE	0x0	RW	奇偶校验使能位 0 = 奇偶校验功能被禁止 1 = 奇偶校验功能被使能
[0]	PT	0x0	RW	校验类型位 0 = 在 PE 为 1 的条件下采用偶校验 1 = 在 PE 为 1 的条件下采用奇校验

注意：当 LOOPS = 0，TE = RE = 1 时，不管 DDRSC1（TXD）和 DDRSC0（RXD）的状态，RXD 管脚都为输入，TXD 管脚都为输出。

表格 11-3：UART 正常，环路及单线模式管脚配置

<table>
<tr><th>LOOPS</th><th>RSRC</th><th>UART 模式</th><th>接收器输入</th><th>RXD 管脚功能</th><th>DDRSC0</th><th>发送器输出</th><th>TXD 管脚功能</th></tr>
<tr><td>0</td><td>x</td><td>正常</td><td>接 RXD 输入缓存</td><td>接收管脚</td><td>x</td><td>接 TXD 输出驱动</td><td>发送管脚</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td rowspan="2">环路</td><td rowspan="2">接发送器输出</td><td rowspan="4">通用 I/O</td><td>0</td><td>只接接收器输入</td><td>空闲为高电平</td></tr>
<tr><td>1</td><td>接接收器输入和 TXD 输出驱动</td><td>发送管脚</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">单线</td><td rowspan="2">接 TXD</td><td>0</td><td>不接</td><td>接收管脚</td></tr>
<tr><td>1</td><td>接 TXD 输出驱动</td><td>发送管脚</td></tr>
</table>

11.6.2.3 UART 控制寄存器 2

偏移地址：0x0002 复位值：0x00

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
rw	rw	rw	rw	rw	rw	rw	rw

图表 11-6：UART 控制寄存器 2(UARTCR2)

比特位	名称	复位值	读写属性	功能说明
[7]	TIE	0x0	RW	发送器中断使能位 该读/写位配置允许 TDRE/TFTS 产生中断请求。复位可清除 TIE。 0 = TDRE/TFTS 中断禁止 1 = TDRE/TFTS 中断使能
[6]	TCIE	0x0	RW	发送完成中断使能位 该读/写位配置允许 TC/FTC 位产生中断请求。复位可清除 TCIE。 0 = TC/FTC 中断禁止 1 = TC/FTC 中断使能
[5]	RIE	0x0	RW	接收器中断使能位 该读/写位配置允许 RDRF/RFTS 和 OR/FOR 及 RTOS 产生中断请求。复位可清除 RIE。 0 = RDRF/RFTS 和 OR/FOR 及 RTOS 中断禁止 1 = RDRF/RFTS 和 OR/FOR 及 RTOS 中断使能
[4]	ILIE	0x0	RW	线上空闲中断使能位 该读/写位配置允许 ILDE 位产生中断。复位可清除 ILIE。 0 = ILDE 中断禁止；1 = ILDE 中断使能
[3]	TE	0x0	RW	发送器使能位 该读/写位使能发送器并将 TXD 管脚配置为发送器输出。将 TE 由低置高会引发一空闲帧排上队列中。复位可清除 TE。 0 = 发送器禁止；1 = 发送器使能
[2]	RE	0x0	RW	接收器使能位 该读/写位使能接收器。复位可清除 RE。 0 = 接收器禁止；1 = 接收器使能
[1]	RWU	0x0	RW	接收器唤醒位 此读/写位将接收器置于待机状态，禁止接收器

比特位	名称	复位值	读写属性	功能说明
				中断请求。WAKE 位决定是空闲线输入还是地址标记来唤醒接收器并清除 RWU。复位可清除 RWU。 0 = 接受唤醒；1 = 不接受唤醒
[0]	SBK	0x0	RW	发送断点 此读/写位将配置 UART 发出 10-bit(M = 0)或 11-bit(M = 1)逻辑 0 的中止帧。当只发送一个中止帧时，需要在该发送完成前就清除掉 SBK。如果 SBK 一直置高，则发送器就会不停地发出中止帧。 0 = 发送器不发送中止帧 1 = 发送器发送中止帧

注意：当 LOOPS = 0，TE = RE = 1 时，不管 DDRSC1（TXD）和 DDRSC0（RXD）的状态，RXD 管脚都为输入，TXD 管脚都为输出。

11.6.2.4 UART 状态寄存器 1

偏移地址：0x0005　　　　复位值：0xC0

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	ILIE	OR	NF	FE	PF
ro	ro	ro	ro	ro	ro	ro	ro

图表 11-7：UART 状态寄存器 1(UARTSR1)

比特位	名称	复位值	读写属性	功能说明
[7]	TDRE	0x1	RO	发送数据寄存器空标志位 在单字模式下，当发送移位寄存器从 UART 发送数据寄存器接收一个字时，TDRE 标志会被设置。表明此时 UARTTDR 是空的，能接收新的数据。如果 UARTCR2 中的 TIE 位被设置，TDRE 将会产生一个中断请求。可以通过先读取 UARTSR1 后再写 UARTDRL 来清除 TDRE。在 FIFO 模式下，当发送 FIFO 为空时，TDRE 标志会被设置。可以通过往发送 FIFO 中写入数据来清除 TDRE。 0 = 发送数据寄存器非空 1 = 发送数据寄存器空
[6]	TC	0x1	RO	发送完成标志 在单字模式下，TC 标志在 TDRE = 1 时并且没

比特位	名称	复位值	读写属性	功能说明
				有数据，导引符或中止帧被发送时被设置，它表明没有正在进行的发送操作。如果 UARTCR2 中的 TCIE 位被设置，TC 会产生一个中断请求。当 TC = 1 时，TXD 管脚上将保持空闲态（逻辑 1）。当有数据，导引符或中止帧被排入队列中。TC 将会被自动清除掉。可以通过先读取 UARTSR1 后再写 UARTDRL 来清除 TC。当有传输正在进行时，TC 不可以被清除。 在 FIFO 模式下，TC 行为同单字模式一致 0 = 正在发送 1 = 无正在发送的数据
[5]	RDRF	0x0	RO	接收数据寄存器满标志位 在单字模式下，当接收移位寄存器内的数据被发送到 UARTRDRH 和 UARTRDRL 时，RDRF 标志被设置。它表明接收的数据对 MCU 是可用的。如果 UARTCR2 寄存器中 RIE 位被置，RDRF 会产生一个中断请求。可以通过先读 UARTSR1 后再读 UARTDRL 来清除 RDRF。 在 FIFO 模式下，当有数据在接收移位寄存器内，但同时接收 FIFO 为满时，RDRF 标志被设置。可以通过读 UARTDRL 并等待下笔数据到来清除 RDRF。 0 = 接收的数据在 UARTRDRH 和 UARTRDRL 内不可用 1 = 接收的数据在 UARTRDRH 和 UARTRDRL 内可用
[4]	IDLE	0x0	RO	线上空闲标志位 当 10-bit(M = 0)或 11-bit(M = 1)连续逻辑 1 出现在接收器输入上时，IDLE 标志被设置。如果设置了 UARTCR2 中的 ILIE 位，则 IDLE 将产生中断请求。一旦 IDLE 被清除，一个有效帧必须再次设置 RDRF，只有这样接下来的空闲条件才可以设置 IDLE 标志。可以通过先读 UARTSR1 后再读 UARTDRL 来清除 IDLE。复位可清除 IDLE。 0 = 接收器未空闲或者空闲由于复位或者上次 IDLE 标志的清除 1 = 接收器空闲
[3]	OR	0x0	RO	溢出标志

比特位	名称	复位值	读写属性	功能说明
				在单字模式下，在接收移位寄存器接收到下一个停止位之前，如果数据不是从 UARTDRL 读取的话，那么 OR 标志就会被设置，这就是接收器溢出的条件。如果 UARTCR2 寄存器中的 RIE 位被置，OR 会产生一个中断请求。发生溢出时，移位寄存器中的数据会被丢失，但是已近在 UARTDRH 和 UARTDRL 中的数据不受影响。可以通过先读 UARTSR1 后再读 UARTDRL 来清除 OR。 在 FIFO 模式下，可以通过读 UARTDRL 来清除 OR。复位可清除 OR。 0 = 无溢出；1 = 溢出
[2]	NF	0x0	RO	噪音标志 在单字模式下，当 UART 在接收器输入上检测到噪音时，NF 标志就会被置。NF 在与 RDRF 标志相同的周期内设置，但当发生溢出时不会被设置。可以通过先读 UARTSR1 后再读 UARTDRL 来清除 NF。 在 FIFO 模式下，该位无效。复位可清除 NF。 0 = 无噪音；1 = 有噪音
[1]	FE	0x0	RO	字段错误位 在单字模式下，当接收到停止位为逻辑 0 时，FE 标志就会被置。FE 在与 RDRF 标志相同的周期内设置，但当发生溢出时不会被设置。FE 会禁止接下来的数据接收，直到被清除。可以通过先读 UARTSR1 后再读 UARTDRL 来清除 FE。 在 FIFO 模式下，该位无效。复位可清除 FE。 0 = 无帧错误；1 = 有帧错误
[0]	PE	0x0	RO	奇偶校验错误位 在单字模式下，当 PE = 1 并且接收到的校验位与硬件产生的校验位不一致时，PF 标志就会被设置。可以通过先读 UARTSR1 后再读 UARTDRL 来清除 PE。 在 FIFO 模式下，该位无效。复位可清除 PF。 0 = 无奇偶校验错；1 = 奇偶校验错

11.6.2.5 UART 状态寄存器 2

偏移地址：0x0004 复位值：0x00

7	6	5	4	3	2	1	0
保留							RAF
ro							r/w1c

图表 11-8：UART 状态寄存器 2(UARTSR2)

比特位	名称	复位值	读写属性	功能说明
[7:1]	保留	0x0	RO	---
[0]	RAF	0x0	R/W1C	接收器工作标志位 当处于搜索起始位的 RT1 时间点，接收器检测到逻辑 0 时，RAF 标志就会被设置。当接收器检测到空闲线时，RAF 会被自动清除掉。复位可清除 RAF。 0 = 接收器不处于接收状态 1 = 接收器处于接收状态

11.6.2.6 UART 数据寄存器

偏移地址：0x0007 复位值：0x00

7	6	5	4	3	2	1	0
R8	T8	保留					
ro	rw	ro					

图表 11-9：UART 数据寄存器高位(UARTDRH)

比特位	名称	复位值	读写属性	功能说明
[7]	R8	0x0	RO	接收数据 bit 8 R8 位是第九位接收到的数据，当采用 9-bits 数据格式（M = 1），复位可清除 R8。
[6]	T8	0x0	RW	发送数据 bit 8 T8 位是第九位将发送的数据，当采用 9-bits 数据格式（M = 1），复位可清除 T8。
[5:0]	保留	0x0	RO	---

注意：如果 T8 的值和之前传输配置的一样，就无须再写。将会发出同样的值直到 T8 被写入新的值。当采用 8-bit 数据格式时，只需访问 UARTDRL。当使用 8-bit 写指令来发送 9-bits 数据时，必须先写 UARTDRH，再写 UARTDRL。

偏移地址：0x0006 复位值：0x00

7	6	5	4	3	2	1	0
R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
rw	rw	rw	rw	rw	rw	rw	rw

图表 11-10：UART 数据寄存器低位(UARTDRL)

比特位	名称	复位值	读写属性	功能说明
[7]	R7/T8	0x0	RW	接收数据位[7]/发送数据位[7] 接收数据位[7]，当采用 9-bit 或 8-bit 格式时。复位可清除 R[7]。 发送数据位[7]，当采用 9-bit 或 8-bit 格式时。复位可清除 T[7]。
[6]	R6/T6	0x0	RW	接收数据位[6]/发送数据位[6] 接收数据位[6]，当采用 9-bit 或 8-bit 格式时。复位可清除 R[6]。 发送数据位[6]，当采用 9-bit 或 8-bit 格式时。复位可清除 T[6]。
[5]	R5/T5	0x0	RW	接收数据位[5]/发送数据位[5]

比特位	名称	复位值	读写属性	功能说明
				接收数据位[5]，当采用 9-bit 或 8-bit 格式时。 复位可清除 R[5]。 发送数据位[5]，当采用 9-bit 或 8-bit 格式时。 复位可清除 T[5]。
[4]	R4/T4	0x0	RW	接收数据位[4]/发送数据位[4] 接收数据位[4]，当采用 9-bit 或 8-bit 格式时。 复位可清除 R[4]。 发送数据位[4]，当采用 9-bit 或 8-bit 格式时。 复位可清除 T[4]。
[3]	R3/T3	0x0	RW	接收数据位[3]/发送数据位[3] 接收数据位[3]，当采用 9-bit 或 8-bit 格式时。 复位可清除 R[3]。 发送数据位[3]，当采用 9-bit 或 8-bit 格式时。 复位可清除 T[3]。
[2]	R2/T2	0x0	RW	接收数据位[2]/发送数据位[2] 接收数据位[2]，当采用 9-bit 或 8-bit 格式时。 复位可清除 R[2]。 发送数据位[2]，当采用 9-bit 或 8-bit 格式时。 复位可清除 T[2]。
[1]	R1/T1	0x0	RW	接收数据位[1]/发送数据位[1] 接收数据位[1]，当采用 9-bit 或 8-bit 格式时。 复位可清除 R[1]。 发送数据位[1]，当采用 9-bit 或 8-bit 格式时。 复位可清除 T[1]。
[0]	R0/T0	0x0	RW	接收数据位[0]/发送数据位[0] 接收数据位[0]，当采用 9-bit 或 8-bit 格式时。 复位可清除 R[0]。 发送数据位[0]，当采用 9-bit 或 8-bit 格式时。 复位可清除 T[0]。

11.6.2.7 UART 上拉和驱动寄存器

偏移地址：0x0009 复位值：0x01

7	6	5	4	3	2	1	0
UARTDOZ	保留	RSVD[2:1]		保留		RSVD[0]	PUPUART
rw	ro	rw		ro		rw	rw

图表 11-11：UART 上拉和驱动寄存器(UARTPURD)

比特位	名称	复位值	读写属性	功能说明
[7]	UARTDOZ	0x0	RW	瞌睡模式下 UART 停止位 0 = 在瞌睡模式下，使能 UART 1 = 在瞌睡模式下，禁止 UART
[6]	保留	0x0	RO	---
[5:4]	RSVD	0x0	RW	保留位，写这些位会更新值但无实际功能意义
[3:2]	保留	0x0	RO	---
[1]	RSVD	0x0	RW	保留位，写这些位会更新值但无实际功能意义
[0]	PUPUART	0x1	RW	UART 端口上拉使能位 此读/写位使能 TXD 和 RXD 管脚上拉。如果管脚被配置为输出，上拉被禁止。 0 = TXD 和 RXD 管脚上拉禁止 1 = TXD 和 RXD 管脚上拉使能

注意：PUPUART 在该设计中无效，上拉一直保持使能。

11.6.2.8 UART 端口数据寄存器

偏移地址：0x0008　　　　复位值：0x03

7	6	5	4	3	2	1	0
RSVD[5:0]						PORTSC1	PORTSC0
rw						rw	rw

图表 11-12：UART 端口数据寄存器(UARTPORT)

比特位	名称	复位值	读写属性	功能说明
[7:2]	RSVD	0x0	RW	写这些读/写位会更新它们的值，但对其功能不会有影响。
[1]	PORTSC1	0x1	RW	UART TXD 端口数据位 这些读/写位对应了 UART 相应的管脚。当 DDRSCx = 0 时，对应的管脚被配置为输入，此时读取 PORTSCx 将返回该管脚上的电平；当 DDRSCx = 1 时，对应的管脚被配置为输出，此时读取 PORTSCx 将返回该管脚上驱动的电平。如果 DDRSCx = 1，往 PORTSCx 写入的值，将被内部锁存住并去驱动对应的管脚输出。
[0]	PORTSC0	0x1	RW	UART RXD 端口数据位 这些读/写位对应了 UART 相应的管脚。当 DDRSCx = 0 时，对应的管脚被配置为输入，此时读取 PORTSCx 将返回该管脚上的电平；当 DDRSCx = 1 时，对应的管脚被配置为输出，此时读取 PORTSCx 将返回该管脚上驱动的电平。如果 DDRSCx = 1，往 PORTSCx 写入的值，将被内部锁存住并去驱动对应的管脚输出。

注意：当管脚作为 UART 输入使用时，对 PORTSP[1:0]的任何写入值都不会改变管脚的状态。为了确保能够正确地从 UARTPORT 读取 UART 管脚电平，在写 UARTDDR 之后到读 UARTPORT 之前，至少有一个时钟周期的等待。

11.6.2.9 UART 数据方向寄存器

偏移地址：0x000B　　复位值：0x00

7	6	5	4	3	2	1	0
RSVD[5:0]						DDRSC1	DDRSC0
rw						rw	rw

图表 11-13：UART 数据方向寄存器(UARTDDR)

比特位	名称	复位值	读写属性	功能说明
[7:2]	RSVD	0x0	RW	写这些读/写位会更新它们的值，但对其功能不会有影响。
[1]	DDRSC1	0x0	RW	UARTPORT TXD 数据方向位 DDRSP[1:0]位控制 UARTPORT 管脚的数据输入输出方向。复位会清除 DDRSC[1:0]。 0 = 对应的管脚配置作为输入 1 = 对应的管脚配置作为输出
[0]	DDRSC0	0x0	RW	UARTPORT RXD 数据方向位 DDRSP[1:0]位控制 UARTPORT 管脚的数据输入输出方向。复位会清除 DDRSC[1:0]。 0 = 对应的管脚配置作为输入 1 = 对应的管脚配置作为输出

注意：当 LOOPS = 0，TE = RE = 1 时，RXD 为输入管脚，TXD 为输出管脚，不管 DDRSC1(TXD)和 DDRSC0(RXD)的状态。

11.6.2.10 UART 测试寄存器

偏移地址：0x000D　　复位值：0x00

7	6	5	4	3	2	1	0
RSVD[1:0]		保留					
rw		ro					

图表 11-14：UART 测试寄存器(UARTTR)

比特位	名称	复位值	读写属性	功能说明
[7:6]	RSVD	0x0	RW	写这些读/写位会更新它们的值，但对其功能不会有影响。保留用于工厂测试。
[5:0]	保留	0x0	RO	---

11.6.2.11 UART 红外控制寄存器

偏移地址：0x000C　　　　　　　　　　　　复位值：0x90

7	6	5	4	3	2	1	0
TNUM		RNUM		TINV	RINV	IRSC	IREN
rw		rw		rw	rw	rw	rw

图表 11-15：UART 红外控制寄存器(UARTIRCR)

比特位	名称	复位值	读写属性	功能说明
[7:6]	TNUM	0x2	RW	发送时钟次数 这些读/写控制位发送时钟数目来产生一个脉冲。复位值为 2’b10。 00 = 1/16 baudrate 01 = 2/16 baudrate 10 = 3/16 baudrate 11 = 保留
[5:4]	RNUM	0x1	RW	接收采样次数 这些读/写控制位确定投票逻辑的接收采样的次数。复位值为 2’b01。 00 = 1 次；01 = 2 次 10 = 3 次；11 = 4 次
[3]	TINV	0x0	RW	红外发送位极性取反 该读/写控制位决定发送时逻辑的电平。复位可以清除 TINV。 0 = 低电平有效发送。红外逻辑期望一个高电平或者一个正相的 IR 3/16 脉冲为 0，期望一个低电平为 1 1 = 高电平有效发送。红外逻辑期望一个低电平或者一个负相的 IR 3/16 脉冲为 0，期望一个高电平为 1
[2]	RINV	0x0	RW	红外接收位极性取反 该读/写控制位决定接收检测时逻辑的电平。复位可以清除 RINV。 0 = 低电平有效检测。红外逻辑期望一个低电平或者一个负相的 IR 3/16 脉冲为 0，期望一个高电平为 1 1 = 高电平有效检测。红外逻辑期望一个高电平或者一个正相的 IR 3/16 脉冲为 0，期望一个低电平为 1
[1]	IRSC	0x0	RW	红外接口采样时钟选择位

比特位	名称	复位值	读写属性	功能说明
				该读/写控制位选择红外接口的采样时钟。复位可以清除 IRSC。 0 = 红外接口采用波特率采样时钟（16 x baudrate） 1 = 红外接口采用系统时钟的分频时钟（根据 UARTIRDR）
[0]	IREN	0x0	RW	红外接口使能位 该读/写控制位使能/关闭红外接口。复位可以清除 IREN。 0 = 红外接口关闭 1 = 红外接口使能

注意：IR 采样时钟由 IRSC 决定。当设置 RNUM[1：0] = 2’b11 时，用户应设置 IRSC = 1，因为根据 IrDA 规范要求 1.41us<MPD<3/16 位时间。如果 IRSC = 0，选择 BS_CLOCK 去取样脉冲 4 次，这是不可能的。

应用于 IRSC 位的值基于两个参数：波特率和收发器的最小脉冲持续时间(MPD)。根据 IrDA 标准，SIR(串行 IR）波特率为 2.4kbit/s 到 115.2kbit/s。该标准的脉冲持续时间等于位持续时间的 3/16(以选定的波特率)。但是，对于所有波特率，还指定了最小脉冲持续时间。根据 IrDA 标准，零由光脉冲表示，因此 IrDA 收发器不能发出比 MPD 更短的光脉冲。对于 SIR，MPD 是恒定的，等于 1.41μs。这意味着用户在任何时候都必须确保 IR 采样时钟的频率足够高，以便能测量脉冲。

在正常操作时，IRSC = 0，脉冲必须持续至少 1 个波特率采样时钟周期(IR 采样时钟周期数由 RNUM 决定)。如果此条件不满足，则 IRSC 必须设置为 1。

11.6.2.12 UART 红外分频系数寄存器

偏移地址：0x000F　　　　复位值：0x00

7	6	5	4	3	2	1	0
IRDR[7:0]							
rw							

图表 11-16：UART 红外分频系数寄存器(UARTIRDR)

比特位	名称	复位值	读写属性	功能说明
[7:0]	IRDR	0x0	RW	红外分频系数 Irclk = fsys/IRDR

注意：当 IrDA 收发器发出比 UART 波特率采样时钟更短的光脉冲时，用户必须确保 IR 采样时钟的频率足够高，以便能测量脉冲。用户应将 IRSC 设置为 1，并设置 IRDR 得到高频红外采样时钟。

举两个例子，最小脉冲持续时间等于 IrDA 规范中的 MPD(SIR)。

例 1. 波特率采样时钟周期的计算(BS_CLOCK 周期<1.41us)。

用户希望以 115.2kbit/s 的速度接收 IrDA 数据。UARTBRDI 和 UARTBRDF 寄存器设置是为了创建波特率采样时钟(BS_CLOCK)，等于 16*波特率的频率 = 16*115.2 = 1.843MHz。但同时，为了能正确检测脉冲，用户必须确保 N*BS_CLOCK 周期低于 1.41us。(N 由 RNUM 决定)让我们做如下检查：

BS_clock period = 1/1843000 = 542ns

因此 2*BS_clock period = 1.09us < 1.41us. 满足要求。N 可以为 1 或 2。即 RNUM 可以被配置为 2’b00 或 2’ b01。

例 2. 波特率采样时钟周期的计算(BS_CLOCK 周期<1.41us)。

本次用户希望以 19.2kbit/s 的速度接收 IrDA 数据。因此 BS_CLOCK 频率应等于 16*19200 = 307.2KHz。让我们做检查 N* BS_clock period < 1.4us：

BS_clock period = 1/307200 = 3.25us。

所以 BS_clock period > 1.41us。不满足工作条件。因此在这种情况下，BS_clock 不能正确测量脉冲持续时间。用户必须配置 IRSC = 1 来生成 IR 采样时钟。当系统时钟频率为 32MHz，

SYS_clock period = 1/32000000 = 0.03125us = IR_clock period/IRDR;

要满足 N*IR_clock period < 1.41us（NN< = 4），就得要求 IRDR<1.41/(0.03125*N)
当 N = 1，IRDR 配置为 8’ h2d；IR_clock period = 1.406us<1.41us；（N 只能为 1）
当 N = 2，IRDR 配置为 8’ h16；2*IR_clock period = 1.375us<1.41us；（N 只能为 1 或 2）
当 N = 3，IRDR 配置为 8’ h0f；3*IR_clock period = 1.406us<1.41us；（N 只能为 1 或 2 或 3）
当 N = 4，IRDR 配置为 8’ h0b；4*IR_clock period = 1.375us<1.41us；（N 只能为 1 或 2 或 3 或 4）

通常我们设置 RNUM = 2’ b01（N = 2）去采样脉冲。

11.6.2.13 UART FIFO 控制寄存器

偏移地址：0x000E 复位值：0x12

7	6	5	4	3	2	1	0
RFEN	TFEN	RXFLSEL			TXFLSEL		
rw	rw	rw			rw		

图表 11-17：UART FIFO 控制寄存器(UARTFCR)

比特位	名称	复位值	读写属性	功能说明
[7]	RFEN	0x0	RW	UART 接收 FIFO 使能 0 = 接收 FIFO 禁止（单字模式） 1 = 接收 FIFO 使能（FIFO 模式）
[6]	TFEN	0x0	RW	UART 发送 FIFO 使能 0 = 发送 FIFO 禁止（单字模式） 1 = 发送 FIFO 使能（FIFO 模式）
[5:3]	RXFLSEL	0x2	RW	UART 接收 FIFO 深度选择 接收 FIFO 的深度配置如下: 000 = RX FIFO > = 1/8 full 001 = RX FIFO > = 1/4 full 010 = RX FIFO > = 1/2 full 011 = RX FIFO > = 3/4 full 100 = RX FIFO > = 7/8 full 101 = 保留 110 = 保留 111 = 保留
[2:0]	TXFLSEL	0x2	RW	UART 发送 FIFO 深度选择 发送 FIFO 的深度配置如下: 000 = TX FIFO < = 7/8 empty 001 = TX FIFO < = 3/4 empty 010 = TX FIFO < = 1/2 empty 011 = TX FIFO < = 1/4 empty 100 = TX FIFO < = 1/8 empty 101 = 保留 110 = 保留 111 = 保留

注意: UARTFCR 寄存器是 FIFO 深度选择寄存器。您可以使用此寄存器来定义 UARTFSR 中的 TFT 和 RFTS 的触发点。当传输的数据量大于设定的基础数据量，中断是会产生。举例，若设定的接收数据量为实际 FIFO 深度（16）的一半，那么当接收到第 8 个数据字时，中断即被触发。在复位后，TXFLSEL 和 RXFLSEL 都被设置为一半深度的等级触发。

11.6.2.14 UART FIFO 状态寄存器

偏移地址：0x0011 复位值：0xC5

7	6	5	4	3	2	1	0
TFTS	FTC	RFTS	RTOS	T_FULL	T_EMPTY	R_FULL	R_EMPTY
ro	ro	ro	ro	ro	ro	ro	ro

图表 11-18：UART FIFO 状态寄存器(UARTFSR)

比特位	名称	复位值	读写属性	功能说明
[7]	TFTS	0x1	RO	UART 发送 FIFO 触发状态 表明当前发送 FIFO 中缓存的数据量已超过 UARTFCR 中定义的阈值。如果 UARTCR2 中的 TIE 位也被设置，则 TFTS 会生成中断请求。 0 = UART 发送 FIFO 中缓存数据量未超过设定的阈值 1 = UART 发送 FIFO 中缓存数据量已超过 FIFO 设定的深度或者少于设定的阈值
[6]	FTC	0x1	RO	UART FIFO 模式下传输完成标志 该位行为定义和 UARTSR1 中的 TC 一样。 0 = 传输正在进行中 1 = 无传输在进行
[5]	RFTS	0x0	RO	UART 接收 FIFO 触发状态 表明当前接收 FIFO 中缓存的数据量已超过 UARTFCR 中定义的阈值。如果 UARTCR2 中的 RIE 位也被设置，则 RFTS 会生成中断请求。 0 = UART 接收 FIFO 中缓存数据量未超过设定的阈值 1 = UART 接收 FIFO 中缓存数据量已超过 FIFO 设定的深度或者多于设定的阈值
[4]	RTOS	0x0	RO	UART 接收超时标志 只有在接收 FIFO 不空的情况下 RTOS 才有意义。 0 = 没有发生接收超时 1 = 发生了一次接收超时
[3]	T_FULL	0x0	RO	UART 发送 FIFO 满标志 该位是否存在意义要基于 FEN 的状态。 0 = 发送 FIFO 不满 1 = 若 FEN = 0，该位无效；若 FEN = 1，则表明发送 FIFO 满
[2]	T_EMPTY	0x0	RO	UART 发送 FIFO 空标志

比特位	名称	复位值	读写属性	功能说明
				该位是否存在意义要基于 FEN 的状态。 0 = 发送 FIFO 没有数据要发送 1 = 若 FEN = 0，该位无效；若 FEN = 1，则表明发送 FIFO 空
[1]	R_FULL	0x0	RO	UART 接收 FIFO 满标志 该位是否存在意义要基于 FEN 的状态。 0 = 接收 FIFO 可以接收新的数据 1 = 若 FEN = 0，该位无效；若 FEN = 1，则表明接收 FIFO 满
[0]	R_EMPTY	0x0	RO	UART 接收 FIFO 空标志 该位是否存在意义要基于 FEN 的状态。 0 = 接收 FIFO 没有数据要发送 1 = 若 FEN = 0，该位无效；若 FEN = 1，则表明接收 FIFO 空

11.6.2.15 UART DMA 控制寄存器

偏移地址：0x0010　　　　复位值：0x00

7	6	5	4	3	2	1	0
保留						TXDMAE	RXDMAE
ro						rw	rw

图表 11-19：UART DMA 控制寄存器(UARTDCR)

比特位	名称	复位值	读写属性	功能说明
[7:2]	保留	0x0	RO	---
[1]	TXDMAE	0x0	RW	UART 发送 DMA 通道使能 0 = 禁止发送 FIFO 的 DMA 通道 1 = 使能发送 FIFO 的 DMA 通道
[0]	RXDMAE	0x0	RW	UART 接收 DMA 通道使能 0 = 禁止接收 FIFO 的 DMA 通道 1 = 使能接收 FIFO 的 DMA 通道

11.6.2.16 UART FIFO 控制寄存器

偏移地址：0x0013 复位值：0x04

7	6	5	4	3	2	1	0
TXFIE	TXFCIE	RXFIE	RXORIE	RXFTOIE	RXFTOE	TXFCLR	RXFCLR
rw	rw	rw	rw	rw	rw	rw	rw

图表 11-20：UART FIFO 控制寄存器 2(UARTFCR2)

比特位	名称	复位值	读写属性	功能说明
[7]	TXFIE	0x0	RW	发送 FIFO 中断使能位 该读/写位配置 TFTS 标志位产生中断请求，复位可以清除 TXFIE 0 = 关闭 TFTS 中断请求 1 = 使能 TFTS 中断请求
[6]	TXFCIE	0x0	RW	发送 FIFO 完成中断使能位 该读/写位配置 FTC 标志位产生中断请求,复位可以清除 TXFCIE 0 = 关闭 FTC 中断请求 1 = 使能 FTC 中断请求
[5]	RXFIE	0x0	RW	接收 FIFO 中断使能位 该读/写位配置 RFTS 标志位产生中断请求，复位可以清除 RXFIE 0 = 关闭 RFTS 中断请求 1 = 使能 RFTS 中断请求
[4]	RXORIE	0x0	RW	接收 FIFO 溢出中断使能位 该读/写位配置 RFOS 标志位产生中断请求，复位可以清除 RXFTOIE 0 = 关闭 RFOS 中断请求 1 = 使能 RFOS 中断请求
[3]	RXFTOIE	0x0	RW	接收 FIFO 超时功能使能位 0 = 关闭接收 FIFO 超时功能 1 = 使能接收 FIFO 超时功能
[2]	RXFTOE	0x1	RW	接收 FIFO 超时功能使能位 0 = 关闭接收 FIFO 超时功能 1 = 使能接收 FIFO 超时功能
[1]	TXFCLR	0x0	RW	清空发送 FIFO 往该位写 1 将复位清空发送 FIFO，读该位总返回 0

比特位	名称	复位值	读写属性	功能说明
[0]	RXFCLR	0x0	RW	清空接收 FIFO 往该位写 1 将复位清空接收 FIFO，读该位总返回 0

11.6.2.17 UART 接收 FIFO 超时计数寄存器

偏移地址：0x0012　　　　复位值：0x28

7	6	5	4	3	2	1	0
UARTRFTOCRT							
rw							

图表 11-21：UART 接收 FIFO 超时计数寄存器 2(UARTRXFTOCTR)

比特位	名称	复位值	读写属性	功能说明
[7:0]	UARTRFTOCRT	0x28	RW	接收 FIFO 超时计数阈值设置 一旦 RX FIFO 不为空，内部的计数器将按一个位时间递减一次。如果在计数器倒计时到 0 之前没有对 RX FIFO 的操作，则 UARTFSR1 中的 RTOS 标志将被设置。

11.6.2.18 UART FIFO 状态寄存器 2

偏移地址：0x0015　　　　复位值：0x00

7	6	5	4	3	2	1	0
保留				FXOR	FXNF	FXFE	FXPF
ro				r/w1o	r/w1o	r/w1o	r/w1o

图表 11-22：UART FIFO 状态寄存器 2(UARTFSR2)

比特位	名称	复位值	读写属性	功能说明
[7:4]	保留	0x0	RO	---
[3]	FXOR	0x0	R/W1O	FIFO 模式下溢出标志 在 FIFO 模式下，当有数据在接收移位寄存器中，但同时接收 FIFO 为满时，FXOR 标志被置起。可以通过读取 UARTDRL 然后往该位写入 1 来清除 FXOR。如果 UARTFCR2 中的 RXORIE 位也被设置，则 FXOR 生成一个中断请求。 0 = FIFO 模式下未发生溢出

比特位	名称	复位值	读写属性	功能说明
				1 = FIFO 模式下发生溢出
[2]	FXNF	0x0	R/W1O	FIFO 模式下噪声标志 在 FIFO 模式下，在从接收 FIFO 读数据过程中，若 UART 在接收器输入上检测到噪声，则 FXNF 标志被置起。可以通过读取 UARTDRL 然后往该位写入 1 来清除 FXNF。 0 = FIFO 模式下未检测到噪声 1 = FIFO 模式下检测到噪声
[1]	FXFE	0x0	R/W1O	FIFO 模式下帧错误标志 在 FIFO 模式下，在从接收 FIFO 读数据过程中，若 UART 检测到停止位为逻辑 0，则 FXFE 标志被置起。可以通过读取 UARTDRL 然后往该位写入 1 来清除 FXFE。 0 = FIFO 模式下未检测到帧错误 1 = FIFO 模式下检测到帧错误
[0]	FXPF	0x0	R/W1O	FIFO 模式下奇偶校验错误标志 在 FIFO 模式下，在从接收 FIFO 读数据过程中，若奇偶校验打开 PE = 1，UART 接收到的校验位与硬件产生的校验位不一致时，则 FXPF 标志被置起。可以通过读取 UARTDRL 然后往该位写入 1 来清除 FXPF。 0 = FIFO 模式下未检测到奇偶校验错误 1 = FIFO 模式下检测到奇偶校验错误

11.6.2.19 UART 流控控制寄存器

偏移地址：0x0014　　复位值：0x00

7	6	5	4	3	2	1	0
保留				RTSE	CTSE	CTSIE	CTSIS
ro				rw	rw	rw	r/w0c

图表 11-23：UART 流控控制寄存器(UARTFCTRL)

比特位	名称	复位值	读写属性	功能说明
[7:4]	保留	0x0	RO	---
[3]	RTSE	0x0	RW	UART 流控 RTS 功能使能 0 = RTS 功能使能；1 = RTS 功能禁止
[2]	CTSE	0x0	RW	UART 流控 CTS 功能使能

比特位	名称	复位值	读写属性	功能说明
				0 = CTS 功能使能；1 = CTS 功能禁止
[1]	CTSIE	0x0	RW	UART 流控 CTS 中断使能 0 = CTS 中断使能；1 = CTS 中断禁止
[0]	CTSIS	0x0	R/W0C	UART 流控 CTS 触发状态 该位表示 CTS 触发状态，写入 0 可清除该位。 0 = CTS 无效，未被触发 1 = CTS 有效，被触发

11.7 功能描述

通用异步收发器（UART）支持全双工、异步、非归 0（NRZ）串行通信方式，可以与 MCU 或其他外部设备进行数字通信。UART 的发送器和接收器是相互独立的，但共享同样的波特率发生器。CPU 监控 UART 的状态，将待写入的数据发送出去，并处理接收到的数据。

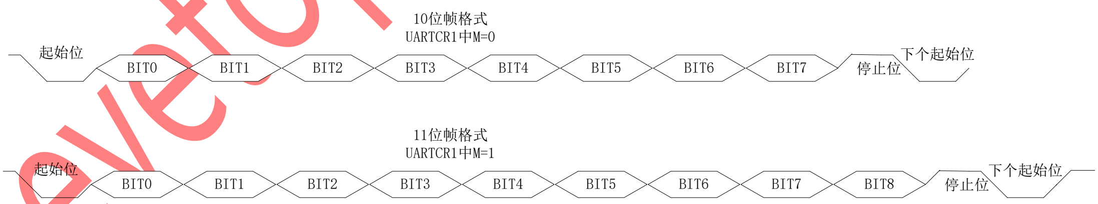
UART 还包括串行 IR(SIR)编码器/解码器模块，可以连接到红外收发机来实现 IrDA SIR 的物理层。

11.7.1 数据格式

UART 使用标准 NRZ 标记/空格数据格式。如下图 10-24 所示。

每个数据段包括一个启动位，8 或 9 位数据，1 或 2 位停止位。清除 SCCR1 寄存器中 M 位可以配置 10-bit 的 UART 数据帧格式。设置 M 位将设置 11-bit 格式的数据帧。

当 UART 配置为 9 位数据时，第九个数据位是 UART 数据寄存器 UARTDRH 中的 T8 位。若传输后 T8 保持不变，在没有重写的情况下可以反复使用。具有 9 个数据比特的帧总共有 11 位。



图表 11-24：UART 数据格式

11.7.2 串行红外（SIR）

UART 内含 IrDA 串行 IR(SIR)编码器/解码器。IrDA SIR 模块提供在异步数据流和半双工串行 SIR 接口之间的转换的功能。片上不执行模拟处理。SIR 的作用是为 UART 提供数字编码输出和解码输入。当红外功能启用时(IREN = 1)，SIR 块将 Tx 和 Rx 管脚用于 SIR 协议。这些信号应该连接到红外收发器以实现 IrDA SIR 物理层链路。SIR 块可以接收和发送，但是它只是半双工的，所以它不能同时进行。在可以接收数据之前发送必须停止 (IrDA SIR 物理层规定发送和接收之间必须有最小 10ms 的延迟)。红外接口兼容 IrDA 串行红外物理层规范。在本规范中，“0”标示正脉冲，“1”表示没有脉冲(线上保持低电平)。

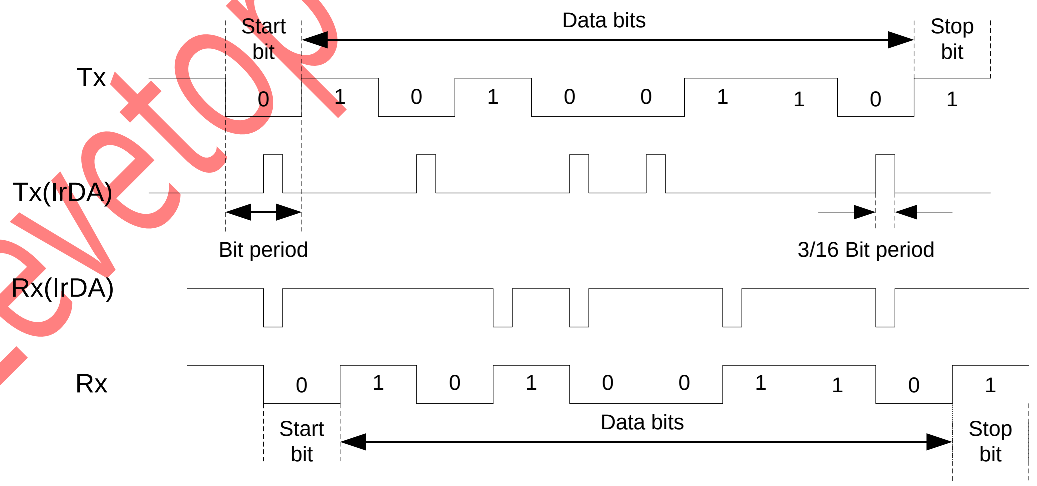
在发送时：对于每个要发送的“零”，产生一个窄的正脉冲，宽度为 1/16~3/16 的位时间(依据 TNUM)。对于要发送的每个“1”，没有脉冲会产生(输出低)。必须提供外部电路来驱动红外线 LED。

在接收时：当接收时，每个“0”都预期有一个窄的负脉冲。每个“1”都预期没有脉冲(输入为高电平)。

TINV 和 RINV 的值取决于连接在 UART 的 TXD 和 RXD 管脚上的 IrDA 收发器。如果此收发器未在 TX 和 Rx 上反转，则 0 由正脉冲表示，1 由无脉冲（线上保持为低）表示。在这种情况下，TINV 必须设置为 0，RINV 必须设置为 1(因为 Rx IR 需要反转的信号)。

反之，如果收发器的 TX 和 RX 都是反相的，则用户必须设置 TINV = 1 和 RINV = 0。也就是说，0 表示为负脉冲，1 表示无脉冲(线保持高电平)。若收发器仅有 Tx 或 Rx 是反相的，在这种情况下，TINV 和 RINV 都必须一致设置成 1 或 0，具体取决于那个管脚需要反相。

TINV = 0，RINV = 0 的情况，如下图 10-25 所示。
如果 RINV = 1，则在接收时，每个“0”预期一个窄的正脉冲。每个“1”预期没有脉冲(输入为低电平)。



图表 11-25：IrDA 数据调制

11.7.3 FIFO 操作

UART 有两个 16 深度的 FIFO；一个用于发送，另一个用于接收。两个 FIFO 通过 UARTDR 寄存器访问。读 UARTDR 寄存器返回 12 位值，该值由 9 个数据位和 3 个错误标志组成。而写入操作则是将 9 位数据放入发送 FIFO 中。

发送 FIFO 通过配置 FCR 中的 TFEN 来使能。
接收 FIFO 通过配置 FCR 中的 RFEN 来使能。

FIFO 的状态可以通过 FSR 寄存器来监管。硬件检测 FIFO 的空，满及溢出的条件。FSR 寄存器包括空和满标志（T_EMPTY，T_FULL，R_EMPTY 和 R_FULL），UARTSR2 寄存器包括溢出标志 FOR。FSR 寄存器包括 TFT，RFTS 和 RTOS 标志。

FIFO 生成中断的触发点通过 FCR 进行控制。两个 FIFO 都可以单独配置为不同的触发中断的阈值。可用的阈值配置包括 1/8，1/4，1/2，3/4 和 7/8。举例，如果为接收 FIFO 设置了 1/4 的配置，则 UART 在接收到 4 个数据字节之后会生成接收中断。在复位后，两个 FIFO 都配置为触发中断在 1/2 深度。

当接收器中至少有一个字符时，并且在四个串行字符的传输时间内未收到任何数据，而 CPU 在该时间内也没有读 FIFO，这种情况下将发生 FIFO 超时（RTOS）。一个串行字符的传输时间周期为：

1/(baud rate) x (# start bits + word length + # parity bits + # stop bits)

举例，一个串行字符包括 8 位数据，一个奇偶校验位，两个停止位。波特率为 56K，则该字符的传输时间计算如下：

1/(56000) * (1+8+1+2) = 214.3us

即在 857.1us 后将发生超时。

11.7.4 波特率计算

波特率发生器中的 13 位计数器同时提供波特率给接收器和发生器。写入 0 到 8191 的值到 UARTBDH 和 UARTBDL，来确定系统时钟的分频。波特率时钟和总线时钟同步并驱动接收器。波特率时钟再除以 16 分频后来驱动发送器。接收器的采样率为每一 bit 时间内采样 16 次。

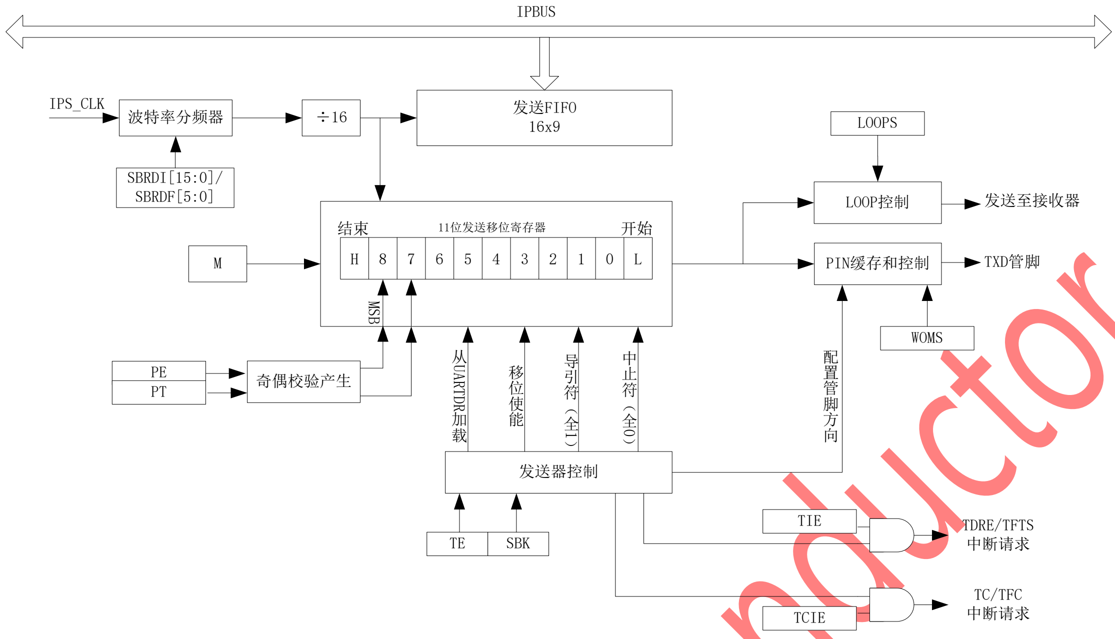
波特率生成受两个误差源的影响：

1. 模块时钟的整数部分分频可能不会得到准确的目标频率
2. 与总线时钟同步可能会导致相移

表格 11-4：波特率配置举例（System Clock = 31MHz）

SBRD[15:0]	SBRDF[5:0]	接收器频率(Hz)	发生器频率(Hz)	目标波特率	误差百分比
0x44CD	0x29	1,760.00	110	110	0.00002%
0x193A	0x15	4,800.00	300.0002	300	0.00008%
0x0C9D	0x0B	9,599.98	599.999	600	0.00016%
0x064E	0x25	19,200.06	1,200.004	1200	0.00032%
0x0327	0x13	38,399.75	2,399.985	2400	0.00065%
0x0193	0x29	76,800.99	4,800.062	4800	0.00129%
0x00C9	0x35	153,596.04	9,599.752	9,600	0.00258%
0x0086	0x23	230,402.97	14,400.19	14,400	0.00129%
0x0064	0x3A	307,215.86	19,200.99	19,200	0.00516%
0x0032	0x1D	614,431.71	38,401.98	38,400	0.00516%
0x0022	0x26	896,115.63	56,007.23	56,000	0.01290%
0x0021	0x29	921,504.88	57,594.05	57,600	0.01032%
0x0010	0x34	1,843,866.17	115,241.6	115,200	0.03614%
0x000F	0x09	2,047,471.62	127,967	128,000	0.02580%
0x0008	0x1A	3,687,732.34	230,483.3	230,400	0.03614%
0x0004	0x0D	7,375,464.68	460,966.5	460,800	0.03614%
0x0001	0x3C	16,000,000.00	1,000,000	1,000,000	0.00000%

11.7.5 发送器



图表 11-26：UART 发送器框图

11.7.5.1 帧长度

发送器可以生成 10 位或 11 位帧。在 UARTCR1 中的 M 位选择帧长度，PE 位使能奇偶校验功能。附加一个数据位可以是地址标记或额外的停止位。所有帧都以起始位开始并以一个或两个停止位结束。当发送 9 位数据时，UART 数据寄存器 UARTDRH 中的位 T8 为第九位数据。

表格 11-5：10-bit 和 11-bit 帧格式

M 位	帧长度	起始位数	数据位数	校验位数	地址标记	停止位数
0	10 bits	1	8	No	No	1
		1	7	No	No	2
		1	7	No	Yes	1
		1	7	Yes	No	1
1	11 bits	1	9	No	No	1
		1	8	No	No	2
		1	8	No	Yes	1
		1	8	Yes	No	1
		1	7	No	Yes	2
		1	7	Yes	No	1

注意：当使用 UART 实现多点网络时，地址标记位用于将后续数据帧指定为网络地址，而不是设备数据。

11.7.5.2 发送帧数据

开始一次 UART 发送，步骤如下：

1. 配置 UART:
   a. 配置波特率寄存器 UARTBDH 和 UARTBDL
   b. 配置 UARTCR1 寄存器
      i. 使能或关闭环路模式并选择接收器回环通路
      ii. 选择开漏(Open-Drain)或线或 UART 输出
      iii. 选择 10-bit 或 11-bit 帧格式
      iv. 选择接收器唤醒条件：地址标记或空闲线
      v. 选择空闲线探测类别
      vi. 选择使能或关闭奇偶校验功能，以及选择奇校验还是偶校验
   c. 配置 UARTCR2:
      i. 使能或关闭 TDRE，TC，RDRD 和 IDLE 中断请求
      ii. 使能发送器并发送中止帧
      iii. 使能或关闭接收器
      iv. 配置接收器进入待机状态如果需要

2. 发送一个字节数据
   a. 清除 TDRE 标志，通过先读 UARTSR1，如果需要发送 9-bit 数据，往 SCDRH 写入第九位数据
   b. 写 UARTDRL（9-bit 模式下低 8 位数据），开始数据发送

3. 重复步骤 2，进行随后的数据发送。

将 TE 位从 0 置成 1 会加载一个 10-bit（M = 0）或者 11-bit（M = 1）逻辑 1 的导引符到发送移位寄存器中去。当导引符移出去后，UARTDRH 和 UARTDRL 中的数据将会加载到发送移位寄存器中。发送移位寄存器会在数据前面加上 0 开始位，并在数据后面追加 1 停止位，并开始将该帧数据移出去。

UART 在每次将数据从 UARTDRH 和 UARTDRL 加载到发送移位寄存器后，将 TDRE 标志置起。TDRE 拉高表示 UARTDRH 和 UARTDRL 可以接受新的数据。如果设置了 TIE 位，则 TDRE 会生成中断请求。

注意： UARTDRH 和 UARTDRL 加载数据至发送移位寄存器并拉高 TDRE，这些动作将在上一帧停止位开始移出后的 9/16 的位时间开始进行。

硬件支持奇偶校验功能。当该功能打开时，最高有效数据位即为校验位。当发送移位寄存器没有发送数据时，TXD 管脚进入空闲态，逻辑 1。在发送器空闲时清除 TE 位后将由 UART 数据方向(UARTDDR)和 UART 端口(UARTPORT)这两个寄存器来控制 TXD 管脚。

如果在传输过程中清除 TE 位(当 TC = 0 时)，则当前仍在发送移位寄存器中的帧数据将继续向外移位。之后 TXD 管脚将恢复为通用 I/O 管脚，即使仍有数据缓存在 UART 数据寄存器中。为避免意外切断消息，请等到

TDRE 拉高之后再清除 TE。

为了区分多个信息数据，可以通过插入导引符，该导引符实际为最短时间的空闲线。具体步骤如下：

1. 将第一笔信息数据的最后一字节写入 UARTDRH 和 UARTDRL。
2. 等到 TDRE 标志置起，表明数据成功加载到了发送移位寄存器中。
3. 通过清除后再置起 TE 位，将导引符排入发送队列中。
4. 将第二笔信息数据的第一字节写入 UARTDRH 和 UARTDRL。

当 UART 放弃 TXD 管脚时，将由 UARTPORT 和 UARTDDR 寄存器控制 TXD 管脚。

要在关闭发射器后强制拉高 TXD，请设置 UART 端口寄存器(UARTPORT)的 bit1 以及 UART 数据方向寄存器(UARTDDR)的 bit1。TXD 管脚。一旦 UART 放弃对它的控制，TXD 管脚就会拉高。参见图表 11-12：和图表 11-13:。

11.7.5.3 中止帧

通过设置 UARTCR2 中的 SBK 位，将加载一个中止帧到发送移位寄存器中。中止帧全部为逻辑 0，并且没有开始位、停止位或奇偶校验位。中止帧长度取决于 UARTCR1 寄存器中的 M 位。只要设置了 SBK，UART 就会连续将中止帧加载到发送移位寄存器中。在 SBK 清除后，发送移位寄存器完成最后一个中止帧的发送，然后再传输至少一个逻辑 1。中止帧末尾的自动逻辑 1 用于保证能识别到下一个起始位。

当 UART 识别到这样的中止帧：起始位后面跟 8-bit 或 9-bit 全 0 的数据位时，停止位也为 0。接收该中止帧会对 UART 寄存器产生以下影响：

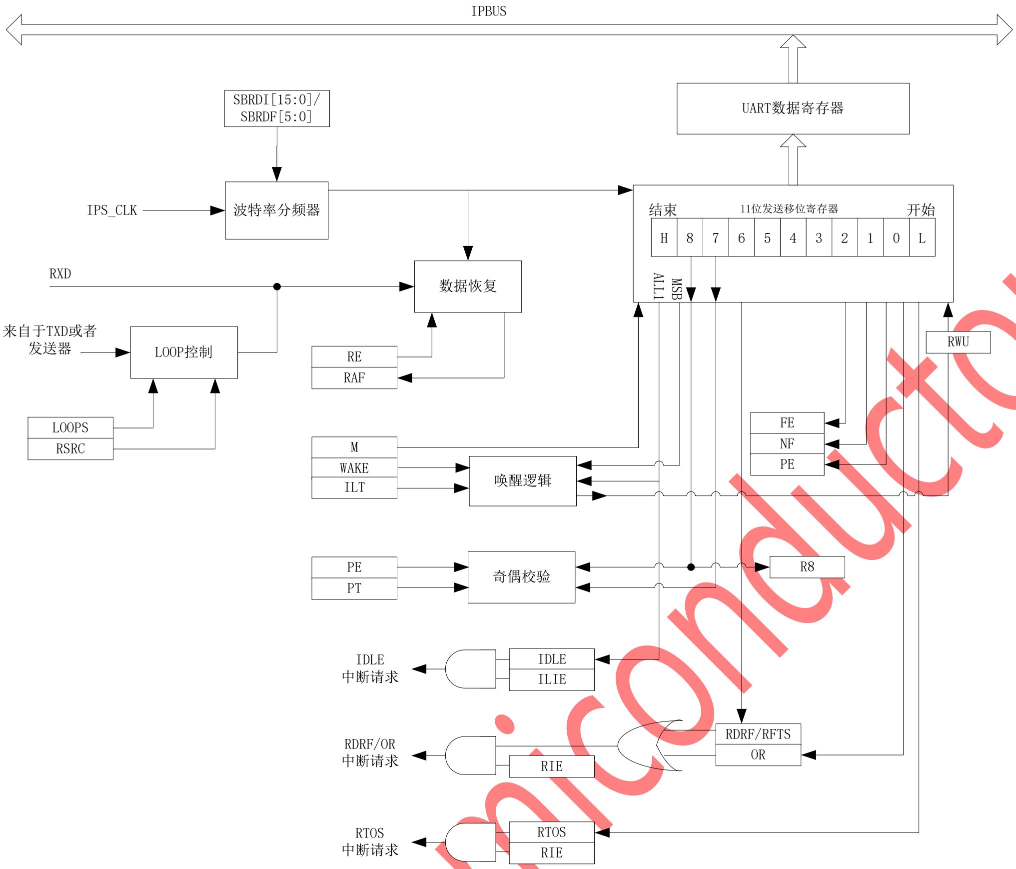
FE 标志置起
RDRF 标志置起
清除 UARTDRH 和 UARTDRL
OR，NF，PE，RAF 标志也可能置起

11.7.5.4 空闲帧

空闲帧全部为逻辑 1，没有起始、停止或奇偶校验位。空闲帧长度取决于 UARTCR1 寄存器中的 M 位。导引符是在将 TE 位从 0 置成 1 之后开始第一次传输的同步空闲帧。如果 TE 位在传输期间被清零，则 TXD 管脚在传输完成后变为空闲。在传输期间清除后再设置 TE 位将在当前正在传输的帧之后插入空闲帧。

注意：在当前帧的停止位从 TXD 移出之前将 TE 由低置高，会插入一个空闲帧至队列中在当前帧的停止位从 TXD 移出之后将 TE 由低置高，会导致先前写入到 UARTDRH 和 UARTDRL 的数据丢失。切换 TE 将空闲帧插入队列中，此时 TDRE 标志会置起在 UARTDRH 和 UARTDRL 被写入新数据之前。

11.7.6 接收器



图表 11-27：UART 接收器框图

11.7.6.1 帧长度

接收器可以处理 8 位或 9 位数据。UARTCR1 中 M 位用来选择帧长度。当接收 9 位数据时，UARTDRH 中的 R8 位是接收数据的第九位(位 8)。

11.7.6.2 接收帧数据

当 UART 接收到帧时，接收移位寄存器将帧从 RXD 管脚移入。当接收器可以处理 8 位或 9 位数据。UARTCR1 中 M 位用来选择帧长度。当接收 9 位数据时，UARTDRH 中的 R8 位是接收数据的第九位(位 8)。

在整个帧移位到接收移位寄存器后，帧的数据部分传输到 UARTDRH 和 UARTDRL。RDRF 标志置起，表示可以读取接收到的数据。如果 RIE 位也被设置，则 RDRF 生成中断请求。

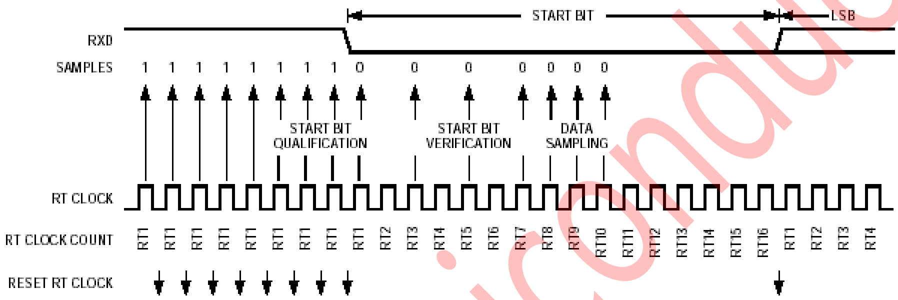
11.7.6.3 数据采样

接收器以 RT 时钟速率对 RXD 管脚进行采样。RT 时钟是频率为波特率 16 倍的内部信号。为调整波特率不匹配，RT 时钟重新同步：

在每个起始位

接收器检测到数据位从逻辑 1 变为逻辑 0 之后（在 RT8、RT9 和 RT10 的大多数数据位样本返回有效逻辑 1，并且接下来的 RT8、RT9 和 RT10 样本的大部分返回有效逻辑 0 之后）

为了定位起始位，数据恢复逻辑对前面有三个 1 后面紧接着一个 0 进行异步搜索。当可能的起始位出现下降沿时，RT 时钟便开始计数到 16。



图表 11-28：接收数据的采样

为了验证起始位和检测噪声，数据恢复逻辑在 RT3、RT5 和 RT7 进行采样。

表格 11-6：起始位验证

RT3，RT5 和 RT7 采样值	起始位验证	噪声标志
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

如果起始位验证不成功，则重置 RT 时钟，并开始新的起始位搜索。为了确定数据位的值并检测噪声，恢复逻辑在 RT8、RT9 和 RT10 进行采样。

表格 11-7：数据位恢复

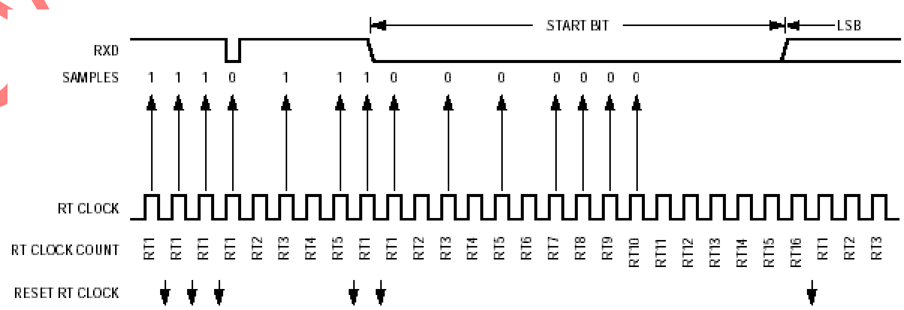
RT8，RT9 和 RT10 采样值	数据位断定	噪声标志
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

注意：RT8、RT9 和 RT10 数据样本不影响起始位验证。如果在起始位验证成功之后，RT8、RT9 和 RT10 样本中的任何一个或全部为逻辑 1，则 NF 标志置高，并且接收器将该位解释为起始位(逻辑 0)。RT8、RT9 和 RT10 样本还验证停止位。

表格 11-8：停止位恢复

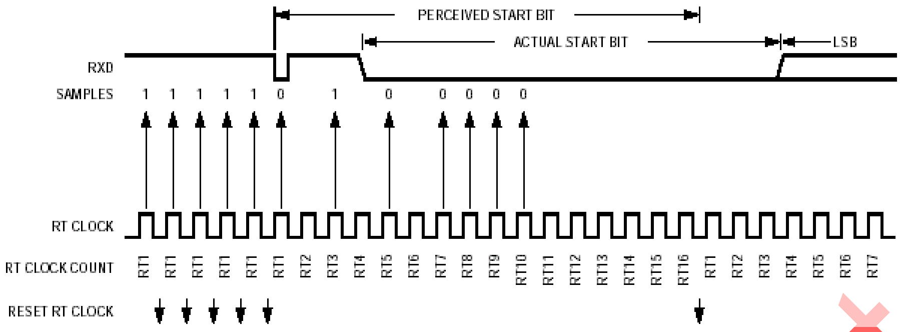
RT8，RT9 和 RT10 采样值	帧错误标志	噪声标志
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

在图表 11-29：中，验证样本 RT3 和 RT5 确定检测到的第一个低电平是噪声，而不是起始位的开始。重置 RT 时钟，并再次开始搜索起始位。NF 标志未置位，因为噪声发生在起始位验证成功之前。



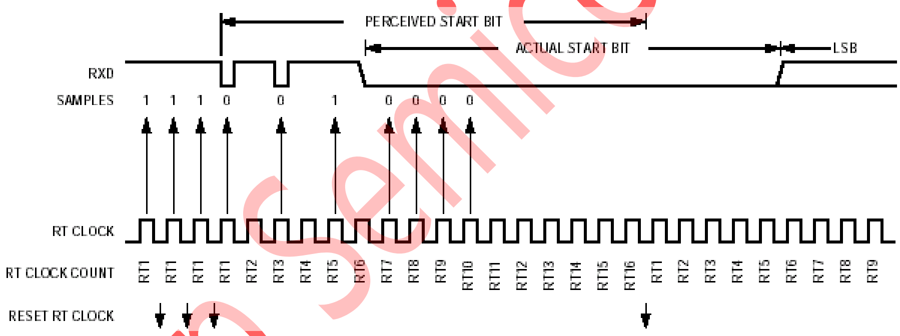
图表 11-29：起始位搜索示例 1

在图表 11-30：中，虽然 RT3 样本是高电平，但噪声被认为是起始位的开始。RT3 采样使噪声标志置起。虽然检测点位时间不标准，但 RT8、RT9 和 RT10 数据采样是在同一个位时间内的，因此数据恢复成功。



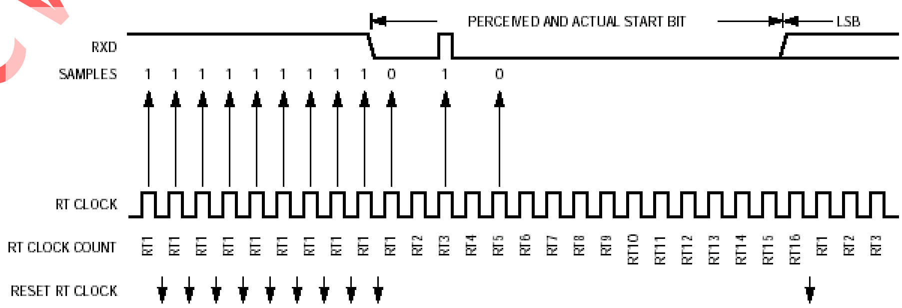
图表 11-30：起始位搜索示例 2

在图表 11-31：中，一串多个噪声突发被认为是起始位的开始，尽管 RT5 采样为高电平。RT5 采样将噪声标志置起。尽管这是位时间不标准最坏情况的情景，但 RT8、RT9 和 RT10 数据采样是在同一个位时间内的，因此数据恢复成功。



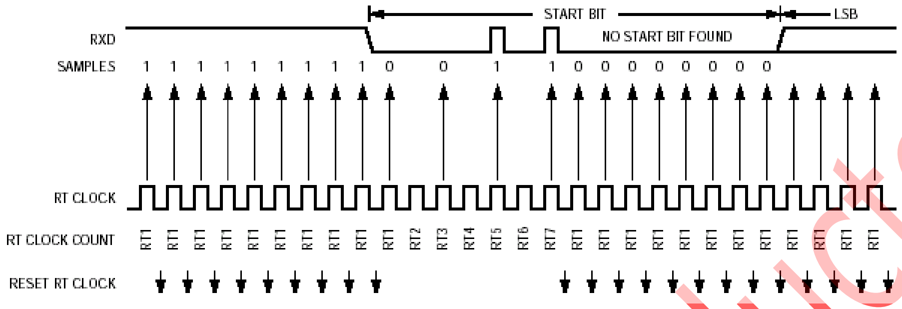
图表 11-31：起始位搜索示例 3

图表 11-32：显示了在起始位时间内早期出现的噪声产生的影响。虽然该噪声不影响与起始位时间的正确同步，但它确实导致了噪声标志置高。



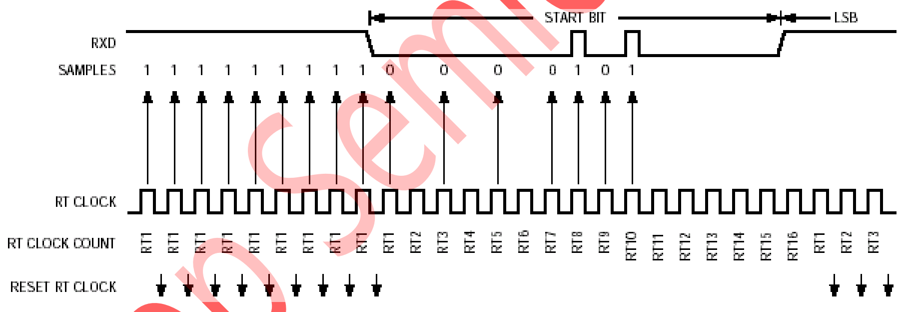
图表 11-32：起始位搜索示例 4

图表 11-33：显示了起始位开始处附近的突发噪声将 RT 时钟复位了。复位后的虽然采样为低电平，但前面没有三个高电平，不能作为正确的下降沿。根据起始位搜索的时序和数据，该帧可能会被整个丢失掉，或者也可能将帧错误标志置高。



图表 11-33：起始位搜索示例 5

在图表 11-34：中，噪声突发使 RT8、RT9 和 RT10 的大多数数据样本为高电平。这会导致噪声标志置高，但不会复位 RT 时钟。仅在起始位中，RT8、RT9 和 RT10 数据样本被忽略。



图表 11-34：起始位搜索示例 6

11.7.6.4 帧错误

如果数据恢复逻辑未在输入的帧中检测到停止位为 1，则会将 UARTSR1 中的 FE 标志置高。中止帧也会将 FE 标志置高，因为中止帧没有停止位。FE 标志和 RDRF 标志同时置高。

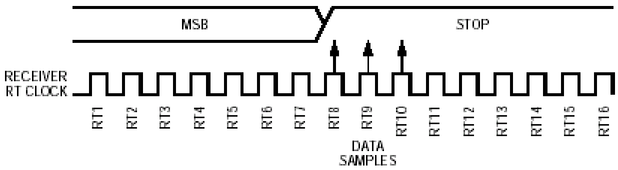
11.7.6.5 波尔率容错

发送设备可以在低于或高于接收器波特率的波特率下工作。累积的未对齐位时间可能会导致 RT8、RT9 和 RT10 停止位采样点落在了该停止位之外。如果采样不都是相同的值，则会发生噪声错误。如果多个样本在停止位之外，则会发生成帧错误。在大多数应用中，波特率容发生的概率远远大于错位可能发生的概率。

当接收器对输入帧进行采样时，它会在帧内的任何有效下降沿重新同步 RT 时钟。帧内的重新同步纠正发送器位时间和接收器位时间之间的未对齐。

11.7.6.5.1 慢速数据容错

图表 11-35：显示了在不导致噪声错误或帧错误的情况下，被接收的帧最慢可以慢到什么程度。慢停止位开始于 RT8，而不是 RT1，但还是及时赶上了接收器对停止位的 3 个数据采样点 RT8、RT9 和 RT10。



图表 11-35：慢速数据

对于 8-bit 数据，接收器完成停止位采样所花费的时间为：

9 bits times x 16 RT cycles + 10 RT cycles = 154 RT cycles

对于图表 11-35：中所示，当接收器计数 154 RT 时，发送设备对自己的 RT 计数值为：

9 bits times x 16 RT cycles + 3 RT cycles = 147 RT cycles

可以推出，在不出现错误的情况下接收器和发送器之间波特率相差的最大百分比为：

(154 – 147) / 154 * 100% = 4.54%

对于 9-bit 数据，接收器完成停止位采样所花费的时间为：

10 bits times x 16 RT cycles + 10 RT cycles = 170 RT cycles

对于图表 11-35：中所示，当接收器计数 170 RT 时，发送设备对自己的 RT 计数值为：

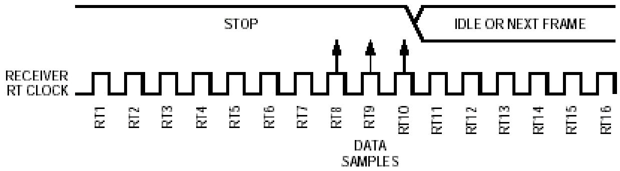
10 bits times x 16 RT cycles + 3 RT cycles = 163 RT cycles

可以推出，在不出现错误的情况下接收器和发送器之间波特率相差的最大百分比为：

(170 – 163) / 170 * 100% = 4.12%

11.7.6.5.2 快速数据容错

图表 11-36：显示了在不导致噪声错误或帧错误的情况下，被接收的帧最快可以快到什么程度。快停止位结束于 RT10，而不是 RT16，但是接收器对它的 3 个采样点（RT8、RT9 和 RT10）还能落在停止位之内。



图表 11-36：快速数据

对于 8-bit 数据，接收器完成停止位采样所花费的时间为：

9 bits times x 16 RT cycles + 10 RT cycles = 154 RT cycles

对于图表 11-36：中所示，当接收器计数 154 RT 时，发送设备对自己的 RT 计数值为：

10 bits times x 16 RT cycles = 160 RT cycles

可以推出，在不出现错误的情况下接收器和发送器之间波特率相差的最大百分比为：

(154 - 160) / 154 * 100% = 3.90%

对于 9-bit 数据，接收器完成停止位采样所花费的时间为：

10 bits times x 16 RT cycles + 10 RT cycles = 170 RT cycles

对于图表 11-36：中所示，当接收器计数 170 RT 时，发送设备对自己的 RT 计数值为：

11 bits times x 16 RT cycles = 176 RT cycles

可以推出，在不出现错误的情况下接收器和发送器之间波特率相差的最大百分比为：

(170 - 176) / 170 * 100% = 3.53%

11.7.6.6 接收器唤醒

为了使 UART 能够忽略仅针对多接收器系统中的其他设备的传输，接收器可以被置于待机状态。通过设置 UARTCR2 中的 RWU 位会使接收器进入待机状态，在此期间接收器中断被禁用。

发送设备可以通过在一个或多个初始帧中添加寻址信息来定位需要通讯的接收器。UARTCR1 中的 WAKE 位确定如何使 UART 脱离待机状态以处理输入的消息。唤醒位使能空闲线唤醒或地址标记的唤醒功能。

11.7.6.6.1 空闲线唤醒（WAKE = 0）

当 WAKE = 0 时，RXD 管脚上的空闲条件会清除 RWU 位并唤醒接收器。每个消息的一个或多个初始帧包含寻址信息。所有接收器评估寻址信息，被成功寻址的接收器将会处理随后的帧。任何未被寻址的接收器都可以设置其 RWU 位并返回到待机状态，直到 RXD 管脚上出现另一个空闲帧。

唤醒接收器的空闲帧不会设置 IDLE 标志或 RDRF 标志。UARTCR1 中的 ILT 位决定接收器是在开始位之后还是在停止位之后开始计数逻辑 1 作为空闲帧位。

11.7.6.6.2 地址标记唤醒（WAKE = 1）

当 WAKE = 1 时，地址标志清除 RWU 位并唤醒接收器。地址标记在最高有效数据位位置为 1。接收器将数据解释为地址数据。当使用地址标志唤醒时，所有非地址数据的 MSB 必须为 0。用户程序必须将地址数据与接收方的地址进行比较，如果地址匹配，则该接收方将处理随后的帧。如果地址不匹配，用户程序必须通过设置 RWU 位使接收器重新进入睡眠状态。RWU 位保持为高，接收器保持待机状态，直到 RXD 管脚上出现另一个地址帧。

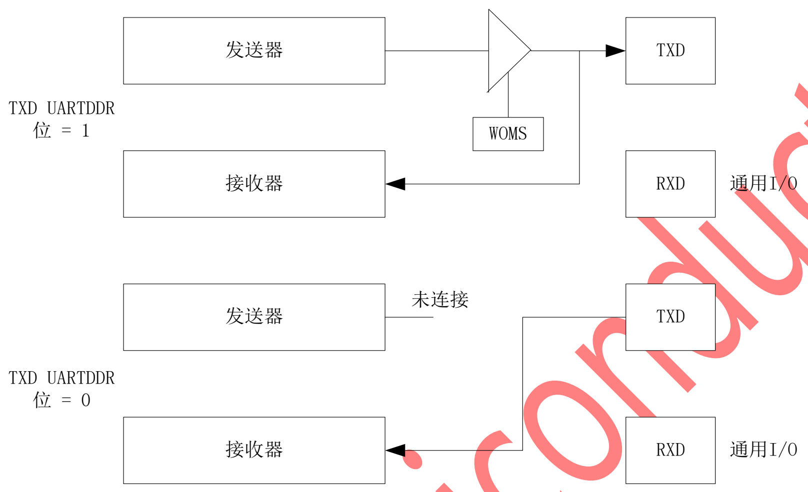
地址标志在接收停止位之前清除 RWU 位，并设置 RDRF 标志。地址标记唤醒允许信息包含空闲帧，但要求为地址数据保留最高有效字节(MSB)。

注意：当 WAKE 位清 0 时，在 RXD 管脚空闲后设置 RWU 位可将接收器立即唤醒。

11.7.7 单线操作

通常情况下，UART 使用 TXD 管脚进行发送，使用 RXD 管脚进行接收(LOOPS = 0，RSRC = X)。在单线模式下，RXD 管脚与 UART 断开，可用作通用 I/O 管脚。UART 使用 TXD 管脚进行接收和发送。

在单线模式下(LOOPS = 1，RXRC = 1)，通过设置 TXD 管脚的数据方向位可将 TXD 配置为输出。清除数据方向位可将 TXD 配置为输入。



图表 11-37：单线操作（LOOPS = 1，RSRC = 1）

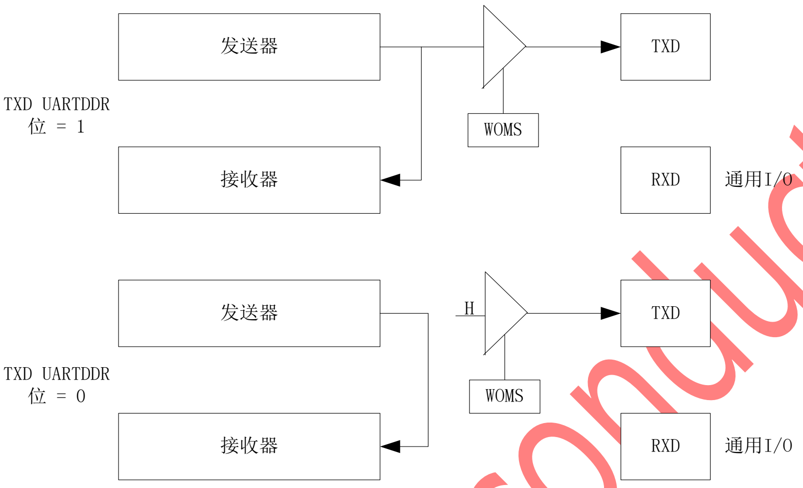
通过设置 UARTCR1 中的 LOOPS 位和 RSRC 位使能单线操作。设置 LOOPS 位来切断 RXD 管脚到接收器的通路。设置 RSRC 位来把接收器的输入连接到 TXD 管脚驱动器的输出。发送器和接收器都必须使能(TE = 1 和 RE = 1)。

UARTCR1 寄存器中的 WOMS 位配置 TXD 管脚为全 CMOS 驱动或开漏(Open-Drain)驱动。在正常操作和单线操作中，WOMS 都可以控制 TXD 管脚。当 WOMS = 1 时，发射器无需清除 TXD 管脚的 DDR 位即可接收数据。

11.7.8 环路操作

在环路模式(LOOPS = 1，RSRC = 0)下，发送器输出进入接收器输入。RXD 管脚与 UART 断开，可作为通用 I/O 管脚使用。

设置 TXD 管脚的 DDR 位可将发送器输出连接到 TXD 管脚。清除数据方向位会断开发送器输出与 TXD 管脚的连接。



图表 11-38：环路操作（LOOPS = 1，RSRC = 0）

通过设置 UARTCR1 中的 LOOPS 位和 RSRC 位使能环路操作。设置 LOOPS 位来切断 RXD 管脚到接收器的通路。清除 RSRC 位来把发送器的输出接到接收器的输入。发送器和接收器都必须使能(TE = 1 和 RE = 1)。

UARTCR1 寄存器中的 WOMS 位配置 TXD 管脚为全 CMOS 驱动或开漏(Open-Drain)驱动。在正常操作和环路操作中，WOMS 都可以控制 TXD 管脚。

11.7.9 硬件流控控制

硬件流控就是通过输入信号 CTSN 和输出信号 RTSN 来控制数据流在两个设备之间的收发通信。RTS 和 CTS 可以独立使能，通过配置 UARTFCTRL 中的 RTSE 和 CTSE。

11.7.9.1 RTS 流控

在 RTS 功能使能（RTSE = 1）时，当 UART 接收器准备好可以接收新的数据时，RTSN 拉低变为有效。当接收数据寄存器满（例如单字模式下 RDRF 置起，FIFO 模式下 RFTS 置起），RTSN 置高无效，表明此时接收器将满，在接收完当前帧数据后停止通信。

11.7.9.2 CTS 流控

在 CTS 功能使能（CTSE = 1）时，发送器会在发送下一帧数据前去检查 CTSN 的状态。如果 CTSN 拉低有效，将开始发送数据，反之则不会发送数据。当在传输过程中 CTSN 拉高无效，发送器将在完成当前传输后停止通信。

当 CTSE = 1 时，当输入信号 CTSN 被触发时，CTSIS 状态位将被硬件自动置起。表明外部接收器已经准备好可以接收数据。当 CTSIE 也被配置，将会产生中断。

11.7.10 I/O 端口

UARTPORT 寄存器与两个管脚相关联：
TXD 接到 UARTPORT1
RXD 接到 UARTPORT0

UART 数据方向寄存器 UARTDDR 配置管脚为输入或输出（参见图表 11-13:）。

在环路模式(LOOPS = 1，RSRC = 0)下，发送器输出进入接收器输入。RXD 管脚与 UART 断开，可作为通用 I/O 管脚使用。

11.7.11 复位

复位将 UART 寄存器初始化为已知的启动状态，如 16.7 存储器映射和寄存器中所述。

11.8 中断描述

列出五种 UART 相关中断请求。

表格 11-9：UART 中断请求源

中断源	中断标志位	中断使能
发送器	TDRE	TIE
	TC	TCIE
	TFTS	TIE/TXFIE
	FTC	TCIE/TXFCIE
接收器	RDRF	RIE
	OR	RIE
	RFTS	RIE/RXFIE
	FOR	RIE/RXORIE
	IDLE	ILIE
	RTOS	RIE/RXFTOIE

12 I2C 总线（I2C）

12.1 概述

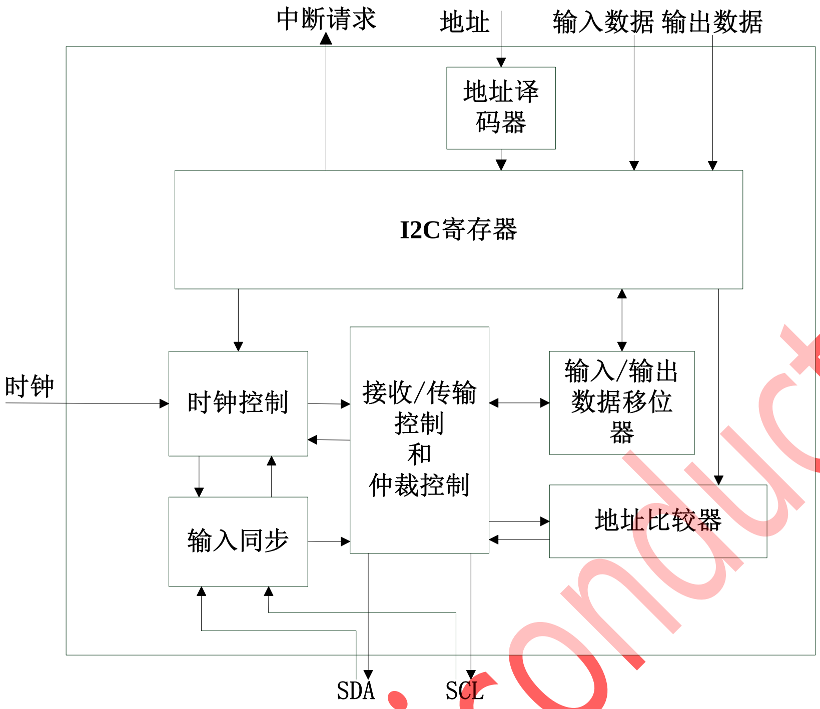
I2C 是双向二线制总线，它提供了一种简单有效的数据方法，最大程度地减少了设备之间的互连。该总线适用于需要在许多设备之间进行短距离偶尔通信的应用。灵活的 I2C 允许将其他设备连接到总线，以进行扩展和系统开发。

12.2 特征

模块特征包括：

支持 7 位寻址和 10 位寻址。
支持三种模式：标准模式、快速模式和高速模式。
可使用软件选择在标准/快速模式和高速模式之间切换。
与 2.1 版本的 I2C 总线标准的标准模式和快速模式兼容。
支持多主机操作。
可编程选择 64 种不同串行频率时钟。
软件可选应答位
基于中断的驱动方式，逐字节地传输数据。
自动从主机模式切换到从机模式的仲裁丢失中断。
传输完成并读取配置的中断。
生成/检测 START 和 STOP 信号。
生成重复 START 信号。
生成/检测应答信号。
总线忙状态检测。
当系统时钟处于停止模式时，可选从机地址接收使能。
支持 SCL 或 SDA 的 GPIO 功能。

12.3 框架图



图表 12-1：I2C 框图

12.4 工作模式

12.4.1 低功耗模式

在特定低功耗模式下，I2C 可以通过中断唤醒芯片。

12.5 外部管脚

12.5.1 SCL

该信号用作 I2C 时钟，也可被用作 GPIO。

12.5.2 SDA

该信号用作 I2C 数据，也可被用作 GPIO。

12.6 内存映射和寄存器

12.6.1 内存映射

在 I2C 内存映射中有 14 个寄存器，如表格 12-1 所示：

表格 12-1：I2C 偏移地址映像

偏移地址	位 7-0	访问权限
0x0000	I2C 从地址高位寄存器（I2CSAH）	S/U
0x0001	I2C 从地址低位寄存器（I2CSAL）	S/U
0x0002	I2C 控制寄存器（I2CC）	S/U
0x0003	I2C 时钟预分频寄存器（I2CP）	S/U
0x0004	I2C 状态寄存器（I2CS）	S/U
0x0005	I2C 数据寄存器（I2CD）	S/U
0x0006	I2C 从机 SDA 保持时间寄存器（I2CSHT）	S/U
0x0007	I2C 从机高速指示器寄存器（I2CSHIR）	S/U
0x0008	I2C 端口控制寄存器（I2CPCR）	S/U
0x0009	I2C 端口数据寄存器（I2CPDR）	S/U
0x000A	I2C 端口方向寄存器（I2CDDR）	S/U
0x000B	I2C 滤波器和电流源测试寄存器（I2CFCTR）	S/U
0x000C	I2C 10ns 滤波器调整值寄存器（I2C10NSFTVR）	S/U
0x000D	I2C 50ns 滤波器调整值寄存器（I2C50NSFTVR）	S/U

注意： S = 超级用户访问；U = 普通用户访问

12.6.2 寄存器描述

12.6.2.1 I2C 从地址高位寄存器（I2CSAH）

偏移地址：0x0000 复位值：0x40

7	6	5	4	3	2	1	0
ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	保留
rw	rw	rw	rw	rw	rw	rw	ro

图表 12-2：I2C 从地址高位寄存器（I2CSAH）

比特位	名称	复位值	读写属性	功能说明
[7:1]	ADDR[7:1]	0x20	RW	**高位地址位：** 当 I2C 作为从机响应主机发送的地址时，I2CSAH 存储地址的高 7 位。当 I2C 作为从机支持 10 位地址时，I2CSAH 为 11110XX。第一个字节的前 7 位是 11110XX，其中后两位（XX）是两位最高位-10 位地址的有效位（MSBs）。第一个字节的第八位是一个 R/W 位，它确定信息传递的方向
[0]	保留	0x0	RO	---

12.6.2.2 I2C 从地址低位寄存器（I2CSAL）

偏移地址：0x0001 复位值：0x00

7	6	5	4	3	2	1	0
ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-3：I2C 从地址低位寄存器（I2CSAL）

比特位	名称	复位值	读写属性	功能说明
[7:0]	ADDR[7:0]	0x40	RW	**低位地址位：** 当 I2C 作为从机支持 10 位地址时，I2CSAL 存储地址的低 8 位。在 7 位地址寻址模式下，I2CSAL 无效。

12.6.2.3 I2C 控制寄存器（I2CC）

偏移地址：0x0002 复位值：0x08

7	6	5	4	3	2	1	0
SLV_HSIE	HMS_EN	AMIE	REPSTA	ACKEN	MSMOD	IEN	EN
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-4：I2C 控制寄存器（I2CC）

比特位	名称	复位值	读写属性	功能说明
[7]	SLV_HSIE	0x0	RW	从机高速模式中断使能位： 选择当 I2C 中断使能控制（IEN）为 1 时，从机高速模式状态是否产生中断请求。 0 = 禁止从机高速模式中断请求 1 = 使能从机高速模式中断请求
[6]	HMS_EN	0x0	RW	**高速模式使能位：** 在主机模式下为 I2C 模块选择高速模式或快速/标准模式操作。将 HSM_EN 位置 1 以选择高速模式操作。在 HS_I2C 不会放弃仲裁且以 F/S 模式发送主代码后，HSM_EN 需要通过软件设置，如果准备清除 MSMOD 以传输 STOP 信号，在清除 MSMOD 之前，应先通过软件将 HSM_EN 清除。 0 = 快速/标准模式选项（默认） 1 = 高速模式选项
[5]	AMIE	0x0	RW	**地址匹配中断使能位：** 选择当 I2C 中断使能控制（IEN）为 1 时，从机地址匹配时是否会产生中断请求。进入停止模式之前，应先设置 AMIE，以便在从机地址匹配时唤醒系统，而此位在正常工作模式下必须清除。 0 = 禁用地址匹配中断请求 1 = 使能地址匹配中断请求
[4]	REPSTA	0x0	RW	重复开始位： 在这些情况下：1）在发送从机地址或数据字节后主接收器未应答，2）无论是否应答，主机发送器均已发送地址或数据字节。主机可以产生一个重复 START 信号，然后发送新的从机地址。在 REPSTA 位置位后配置从机地址（通过写 I2CD）时，将发送重复 START 位信号。 0 = 不重复开始；1 = 发生重复开始

比特位	名称	复位值	读写属性	功能说明
[3]	ACKEN	0x1	RW	应答使能控制： 指定主接收器和从接收器在应答周期内驱动到 SDA 的值。注意，仅当 I2C 总线接收到数据字节时，ACKEN 位才生效。在接收地址期间，如果接收到的地址与从机地址匹配，无论 ACKEN 位状态，都将发送应答信号。 0 = 接收一个字节的数据后，第九个时钟位没有应答信号发送到 SDA 1 = 接收一个字节的数据后，第九个时钟位将应答信号发送到 SDA。
[2]	MSMOD	0x0	RW	I2C 主/从模式选择控制： 将 MSMOD 从 1 更改为 0 会在总线上产生一个 STOP 并选择从机模式。将 MSMOD 从 0 更改为 1 会在总线上产生一个 START 并选择主机模式。 0 = 从机模式；1 = 主机模式
[1]	IEN	0x0	RW	I2C 中断使能控制位： 当 IEN 置位时，I2C 中断使能。 0 = I2C 中断禁用；1 = I2C 中断使能
[0]	EN	0x0	RW	I2C 模块使能控制位： 它使能/禁用模块。还能控制整个 I2C 模块的软件复位。设置该位会生成模块的内部复位，该复位将在设置该位 2 个时钟周期后置位，并保持 3 个时钟周期。因此，在设置 EN 位的 5 个时钟周期后，复位无效。 如果在字节传输过程中使能了模块，则从机模式将忽略当前的 I2C 总线传输，并在检测到下一个 START 条件时开始运行。主机模式不能检测总线忙。因此开始启动周期可能会破坏当前的总线周期，最终导致当前的主机或 I2C 模块失去仲裁，此后总线操作将恢复正常。 0 = I2C 模块禁用；1 = I2C 模块使能

12.6.2.4 I2C 时钟预分频寄存器（I2CP）

偏移地址：0x0003 复位值：0x00

7	6	5	4	3	2	1	0
保留	TEST	PRE5	PRE4	PRE3	PRE2	PRE1	PRE0
ro	rw	rw	rw	rw	rw	rw	rw

图表 12-5：I2C 时钟预分频寄存器（I2CP）

比特位	名称	复位值	读写属性	功能说明
[7]	保留	0x0	RO	---
[6]	TEST	0x0	RW	时钟测试使能**位**： TEST 位可使能 I2C 时钟的测试模式。在正常模式下，频率为 fsys/（396×（PRE[5:0]+1）+1）。在测试模式下，频率为 fsys/（8×（PRE[5:0]+1）+1）。 0 = 正常模式； 1 = 使能时钟测试模式
[5:0]	PRE[5:0]	0x0	RW	预分频器的分频值： I2CP 是一个预分频器，用于为数据收发器生成一个比特率时钟。由于 SCL 和 SDA 的上升和下降时间可能很慢，因此会以预分频器频率对总线信号进行采样。串行位时钟频率等于系统时钟除以分频器。

12.6.2.5 I2C 状态寄存器（I2CS）

偏移地址：0x0004　　复位值：0x00

7	6	5	4	3	2	1	0
AACK	DACK	RXTX	ARBL	BBUSY	AASLV	RC	TF
ro	ro	ro	ro	ro	ro	ro	ro

图表 12-6：I2C 状态寄存器（I2CS）

比特位	名称	复位值	读写属性	功能说明
[7]	AACK	0x0	RO	地址应答错误 指示主机在地址阶段是否检测到应答位。它由地址位的第九个时钟的上升沿设置，并通过将 MSMOD 从 1 更改为 0 或重复 START 来清除。 0 = 没有地址应答错误；1 = 地址应答错误
[6]	DACK	0x0	RO	收到数据应答 指示在地址或数据传输期间是否检测到应答信号。在第九个时钟的上升沿生效。 0 = 没有收到应答信号；1 = 收到应答信号
[5]	RXTX	0x0	RO	接收发送位 指示 I2C 模块充当接收器还是发送器。它在第八个时钟的下降沿生效。 0 = 接收器，接收数据；1 = 发送器，发送数据
[4]	ARBL	0x0	RO	失去仲裁 显示总线的仲裁状态。在 SCL 为高期间，它将在以下情况下设置： 主机在 START 条件，地址周期，数据发送周期或 STOP 条件下驱动为高电平时，SDA 采样为低电平。 当主机在数据接收周期的确认位期间驱动为高电平时，SDA 采样为低电平。 总线繁忙时尝试 START 周期。 必须通过软件读 I2CS 寄存器来清除 ARBL。 0 = 没有失去仲裁；1 = 失去仲裁
[3]	BBUSY	0x0	RO	I2C 总线忙 显示总线状态。 0 = 总线空闲，由 STOP 清除 1 = 总线忙，由 START 置位
[2]	AASLV	0x0	RO	作为从机寻址 如果将 I2C 模块作为从机寻址，并且其自己的从机地址与 SDL 上接收到的调用地址相匹配，则由第八

比特位	名称	复位值	读写属性	功能说明
				个时钟的下降沿设置此位。通过检测到的 START 或 STOP 位将其清除。 0 = 不作为从机； 1 = 作为从机寻址
[1]	RC	0x0	RO	配置为接收器位 表示配置为接收器。对于主接收器，无论是否检测到应答位，它都是通过第九个时钟的下降沿来设置的。对于从接收器，它由接收到的地址或数据字节的第九个时钟的下降沿设置，并且必须接收到确认位。如果 I2CC 中的 IEN 位置 1，将产生一个中断。通过读 RC 已置位的 I2CS 且写 I2CC 来清除 RC。 0 = 无意义； 1 = 表示配置为接收器
[0]	TF	0x0	RO	传输完成标志位 表示发送或接收数据。对于接收器，无论是否检测到应答位，均由接收到的数据（非地址）字节的第九个时钟的下降沿设置此位。对于主发送器，无论是否检测到应答位，它都是通过发送的数据或地址字节的第九个时钟的下降沿来设置的。对于从发送器，它由地址或数据字节传输的第九个时钟的下降沿设置，必须检测到应答位。如果 I2CC 中的 IEN 位置 1，则会产生一个中断。通过读取设置了 TF 的 I2CS 清除 TF，然后访问 I2CD 或写 I2CC 的主发送模式。 0 = 无意义； 1 = 数据或地址传输完成

I2CS 显示了 I2C 模块的状态。

12.6.2.6 I2C 数据寄存器（I2CD）

偏移地址：0x0005 复位值：0x00

7	6	5	4	3	2	1	0
R/T[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-7：I2C 数据寄存器（I2CD）

比特位	名称	复位值	读写属性	功能说明
[7:0]	R/T[7:0]	0x0	RW	接收/发送位 I2CD 寄存器保存要发送的数据（下一个字节）或接收的数据。在主机模式下，它还保存要发送的从机地址和传输方向。[7:1]位形成从站地址，[0]位是传输方向（R/W 位）。在主接收器模式下，读 I2CD 将允许进行读取并启动下一个字节数据的接收。在主发送模式下，写 I2CD 将存储下一次发送的字节。在从机模式下，寻址后可以使用相同的功能。

12.6.2.7 I2C 从机 SDA 保持时间寄存器（I2CSHT）

偏移地址：0x0006 复位值：0x09

7	6	5	4	3	2	1	0
SCL_FILTER_EN	SDA_FILTER_EN	SLVHT					
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-8：从机 SDA 保持时间寄存器（I2CSHT）

比特位	名称	复位值	读写属性	功能说明
[7]	SCL_FILTER_EN	0x0	RW	SCL 滤波器使能 如果使能了 SCL 滤波器，则在进行高速模式转换时，将滤除 SCL 线上发生的 10ns 脉冲。如果正在进行快速/普通模式转换，则 SCL 线上发生的 50ns 脉冲将被滤波。
[6]	SDA_FLITER_EN	0x0	RW	SDA 滤波器使能 如果使能了 SDA 滤波器，则在进行高速模式转换时，将滤除 SDA 线上发生的 10ns 脉冲。如果正在进行快速/普通模式转换，则 SDA 线上

比特位	名称	复位值	读写属性	功能说明
				发生的 50ns 脉冲将被滤波。
[5:0]	SLVHT	0x09	RW	从机 SDA 线保持时间配置 当 I2C 作为从机输出模式时，数据将在 SCL 下降沿且内部 SDA 保持寄存器的值等于 SLVHT 之后改变

12.6.2.8 I2C 从机高速模式指示寄存器（I2CSHIR）

偏移地址：0x0007　　复位值：0x00

7	6	5	4	3	2	1	0
保留							SLV_HS
ro	ro	ro	ro	ro	ro	ro	r/w1c

图表 12-9：从机高速模式指示寄存器（I2CSHIR）

比特位	名称	复位值	读写属性	功能说明
[7:1]	保留	0x0	RO	---
[0]	SLV_HS	0x0	R/W1C	从机高速模式**位** 当作为从机时，该位指示 I2C 模块用于高速模式还是快速/标准模式数据传输。该位在主代码的第九个 SCL 上升沿且接收到未应答位后被置位。该位必须通过向其写 1 来清除，否则 SCL 线将被强制变为 LOW 状态 0 = I2C 模块选择进行快速/标准数据传输（默认） 1 = I2C 模块进行高速模式数据传输

12.6.2.9 I2C 端口控制寄存器（I2CPCR）

偏移地址：0x0008　　复位值：0x03

7	6	5	4	3	2	1	0
SDAPA	SCLPA	WOMI2C[1:0]		PDI2C[1:0]		PUI2C[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-10：I2C 端口控制寄存器（I2CPCR）

比特位	名称	复位值	读写属性	功能说明
[7]	SDAPA	0x0	RW	SDA 端口分配 读/写此位选择 SDA 的功能模式。 0 = 管脚配置为功能管脚 1 = 管脚配置为 GPIO
[6]	SCLPA	0x0	RW	SCL 端口分配： 读/写此位选择 SCL 的功能模式。 0 = 管脚配置为功能管脚 1 = 管脚配置为 GPIO
[5:4]	WOMI2C[1:0]	0x0	RW	线或模式 读/写此位将相应的 I2C 管脚设置为漏极开路驱动模式。WOMI2C[1] 用于 SDA 管脚，WOMI2C[0]用于 SCL 管脚。这些位仅在 GPIO 模式下可用。 0 = 输出时 CMOS 驱动；1 = 输出时开漏(Open-Drain)
[3:2]	PDI2C[1:0]	0x0	RW	下拉使能位 读/写此位使能相应的 I2C 管脚的下拉。PDI2C[1]用于 SDA 管脚，PDI2C[0]用于 SCL 管脚。这些位仅在 GPIO 模式下可用。 0 = 下拉禁用；1 = 下拉使能
[1:0]	PUI2C[1:0]	0x3	RW	上拉使能位 读/写此位使能相应的 I2C 管脚的上拉。PUI2C[1]用于 SDA 管脚，PUI2C[0]用于 SCL 管脚。这些位仅在 GPIO 模式下可用。 0 = 上拉禁用；1 = 上拉使能

12.6.2.10 I2C 端口数据寄存器（I2CPDR）

偏移地址：0x0009 复位值：0x00

7	6	5	4	3	2	1	0
保留						PORTI2C[1:0]	
ro	ro	ro	ro	ro	ro	rw	rw

图表 12-11：I2C 端口数据寄存器（I2CPDR）

比特位	名称	复位值	读写属性	功能说明
[7:2]	保留	0x0	RO	---
[1:0]	PORTI2C[1:0]	0x0	RW	I2C 端口数据 写操作将设置被配置为 GPIO 的相对应的 I2C 管脚输出数据。读操作将返回 I2C 管脚的电平。

12.6.2.11 I2C 端口方向寄存器（I2CDDR）

偏移地址：0x000A 复位值：0x00

7	6	5	4	3	2	1	0
保留						DDRI2C[1:0]	
ro	ro	ro	ro	ro	ro	rw	rw

图表 12-12：I2C 端口方向寄存器（I2CDDR）

比特位	名称	复位值	读写属性	功能说明
[7:2]	保留	0x0	RO	---
[1:0]	DDRI2C[1:0]	0x0	RW	I2C 端口方向 读/写此位控制 I2C 管脚的数据方向。这些位仅在 GPIO 模式下可用。 0 = 对应管脚配置为输入 1 = 对应管脚配置为输出

12.6.2.12 I2C 滤波器和电流源测试寄存器（I2CFCTR）

偏移地址：0x000B　　　　复位值：0x00

7	6	5	4	3	2	1	0
保留				I2CCSTE	I2CFTE	SCL10NSTE	SDA10NSTE
ro	ro	ro	ro	rw	rw	rw	rw

图表 12-13：I2C 滤波器和电流源测试寄存器（I2CFCTR）

比特位	名称	复位值	读写属性	功能说明
[7:4]	保留	0x0	RO	---
[3]	I2CCSTE	0x0	RW	I2C 电流源测试使能位 0 = I2C 电流源测试禁用 1 = I2C 电流源测试使能
[2]	I2CFTE	0x0	RW	I2C 滤波器测试使能位 0 = I2C 滤波器测试禁用 1 = I2C 滤波器测试使能
[1]	SCL10NSTE	0x0	RW	I2CSCL10ns 滤波器测试使能位 0 = 如果 I2CFTE 置位，I2C 模块 SCL 50ns 滤波器测试使能 1 = 如果 I2CFTE 置位，I2C 模块 SCL 10ns 滤波器测试使能
[0]	SDA10NSTE	0x0	RW	I2CSDA10ns 滤波器测试使能 0 = 如果 I2CFTE 置位，I2C 模块 SDA 50ns 滤波器测试使能 1 = 如果 I2CFTE 置位，I2C 模块 SDA 10ns 滤波器测试使能

12.6.2.13 I2C 10ns 滤波器调整值寄存器（I2C10NSFTVR）

偏移地址：0x000C　　复位值：0x88

7	6	5	4	3	2	1	0
SCL10F[3:0]				SDA10F[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-14：I2C 10ns 滤波器调整值寄存器（I2C10NSFTVR）

比特位	名称	复位值	读写属性	功能说明
[7:4]	SCL10F[3:0]	0x8	RW	I2C SCL 线 10ns 滤波器调整值
[3:0]	SDA10F[3:0]	0x8	RW	I2C SDA 线 10ns 滤波器调整值

如果闪存信息区域中的值与键值不匹配，则 CPU 可以写入该寄存器的值。如果键值匹配，则该值将被闪存信息区域中的值覆盖。

12.6.2.14 I2C 50ns 滤波器调整值寄存器（I2C50NSFTVR）

偏移地址：0x000D　　复位值：0x66

7	6	5	4	3	2	1	0
SCL50F[3:0]				SDA50F[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw

图表 12-15：I2C 50ns 滤波器调整值寄存器（I2C50NSFTVR）

比特位	名称	复位值	读写属性	功能说明
[7:4]	SCL50F[3:0]	0x6	RW	I2C SCL 线 50ns 滤波器调整值
[3:0]	SDA50F[3:0]	0x6	RW	I2C SDA 线 50ns 滤波器调整值

如果闪存信息区域中的值与键值不匹配，则 CPU 可以写入该寄存器的值。如果键值匹配，则该值将被闪存信息区域中的值覆盖。

12.7 功能描述

12.7.1 主机模式

当总线空闲（I2CS 的 BBUSY 位清零）时，如果作为主机，I2C 模块可以初始化传输。将 MSMOD 位从 0 更改为 1 会在总线上产生 START 信号，并选择主机模式。主机可控制传输方向，即 R/W 位。主机发送传输的第一个字节是从机地址，后一个字节是数据。如果在每个第九个时钟周期后未收到应答，则 MSMOD 位从 1 更改为 0，以在总线上产生 STOP 信号。I2CP 中的 PRE[5:0]位控制 I2C 总线的比特率时钟。

通过将 REPSTA 位置 1 之后配置从机地址，主机可以重复 START 信号，而不用发出 STOP 信号。

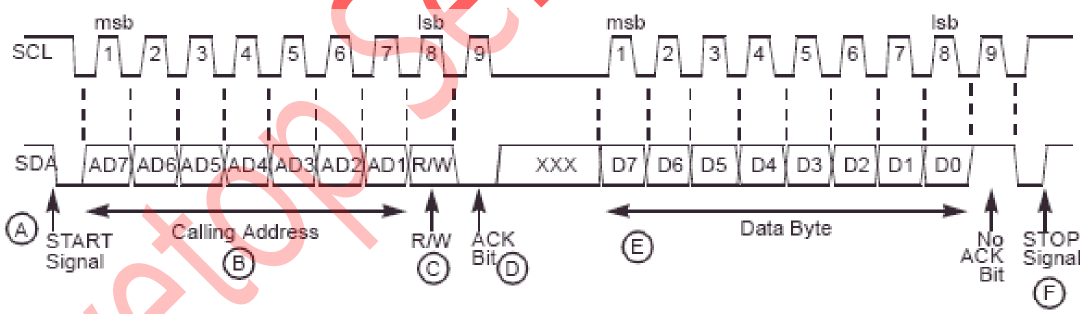
12.7.2 从机模式

如果 MSMOD 位被清除，则该 I2C 模块作为从机，可以被其他主机寻址。当仲裁胜出主机尝试寻址它时，它将释放 SDA 线并立即切换到其主机的从机模式。

注意：I2C 不能同时在从机模式和主机模式下工作。

12.7.3 协议

I2C 通信协议包含六个部分：START 信号，数据源/接收器，数据方向，从机应答，数据，数据应答和 STOP 信号。如图 21-2 中所示：



图表 12-16：I2C 通讯协议

START 信号：当没有其他设备成为总线主设备时(SCL 和 SDA 线均处于逻辑高电平)，设备可以通过发送 START 信号来启动通信（参考图 21-2 中的 A）。START 信号定义为当 SCL 为高时，SDA 由高到低的跳变。该信号表示数据传输的开始（每次数据传输可以是几个字节长），并唤醒所有从机。

从机地址发送：主机在 START 信号（B）之后的第一个字节中发送从机地址。在 7 位寻址地址之后，它将发送 R/W 位（C），告诉从机数据传输方向。

每个从机必须具有唯一的地址。I2C 主机不能与自己作为从机时地址相同的地址进行传输，因为它不能同时是主机和从机。地址与主机发送的地址匹配的从机在第九个时钟（D）将 SDA 拉低，以返回一个应答信号。

数据传输：成功完成从机寻址后，数据可以按调用主机发送的 R/W 位指定的方向逐字节进行传输（E）。

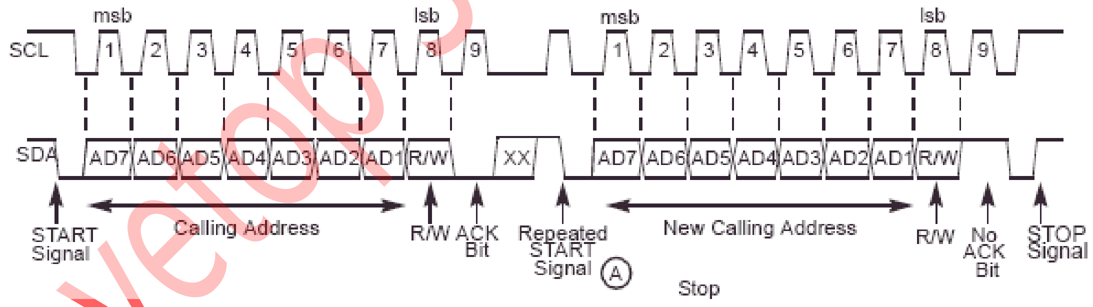
如图 21-2 所示，只有在 SCL 为低时才能更改数据，并且在 SCL 为高时必须保持稳定。SCL 在每个数据位脉冲一次，首先发送 MSB 位。每个字节接收设备必须在第九个时钟周期将 SDA 拉低来应答。因此，数据字节传输需要九个时钟脉冲。

如果不应答主机，则从机接收器必须将 SDA 拉高。然后，主机可以产生一个 STOP 信号来中止数据传输，或者产生一个 START 信号（重复 START 信号，如图 21-3 所示）来开始一个新的寻址序列。

如果主机接收器在字节传输后没有应答从机发送器，则意味着从机的数据传输结束。从机释放 SDA 给主机，以产生 STOP 或重复 START 信号。

STOP 信号：主机可以通过生成 STOP 信号以释放总线来终止通信。STOP 信号定义为 SCL 为逻辑高电平（F）时 SDA 由低到高的跳变。请注意，即使从机做出了应答，主机也可以产生 STOP 信号，此时从机必须释放总线。

主机可以重复发出 START 信号，然后再发出调用命令（图 21-3 中的 A），而不是产生 STOP 信号。当没有先生成 STOP 信号结束通信而是生成 START 信号时，就会发生重复开始操作。主机使用重复的 START 信号来与另一个从机或以不同模式（发送/接收模式）与同一从机通信，而无需释放总线。



图表 12-17：I2C 协议中的重复 START 位

12.7.4 仲裁程序

如果多个设备同时请求总线，总线时钟则需要同步过程确定，其中慢周期等于设备中最长的时钟慢周期，而快周期等于最短周期。数据仲裁程序确定竞争设备的相对优先级。如果一台设备发送逻辑高电平，而另一台设备发送逻辑低电平，则它将失去仲裁。它会立即切换到从机接收模式并停止驱动 SDA。

失去仲裁的主机可以生成时钟脉冲，直到丢失仲裁的字节的末尾为止，因为获得仲裁的主机可能正在寻址它。此时丢失仲裁的主机必须立即切换到其从机模式

在这种情况下，从主机模式到从机模式的转换不会产生 STOP 信号。同时，硬件将 I2CSR 的 ARBL 位置 1 以标志仲裁丢失。

12.7.5 时钟同步

由于使用了线与逻辑，因此 SCL 上从高到低的转换会影响连接到总线的所有设备。当主机将 SCL 驱动为低时，设备开始计数其低电平周期。当设备时钟变为低电平时，它将 SCL 保持为低电平，直到时钟变成高电平状态为止。但是，如果另一个设备时钟仍处于其低电平周期，则该设备时钟的从低到高变化可能不会更改 SCL 的状态。

所以，具有最长低电平周期的设备会将同步时钟 SCL 保持为低电平。短周期的设备在此期间进入高等待状态（请参见图 21-4）。当涉及的所有设备都计数完低电平周期后，同步时钟 SCL 就会释放并拉高。

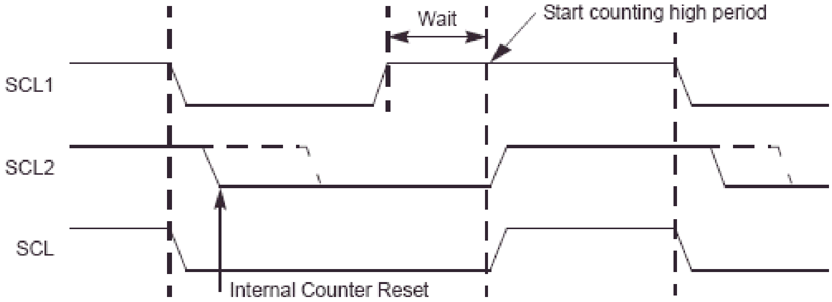
这样，设备时钟和 SCL 状态之间就没有区别，因此所有设备都开始计数其高电平周期。第一个完成其高电平周期的器件将 SCL 再次拉低

12.7.6 握手操作

时钟同步机制可以用作数据传输中的握手。从机可以在完成一个字节传输（9 位）后将 SCL 保持为低电平。在这种情况下，时钟机制将停止总线时钟，并迫使主时钟进入等待状态，直到从设备释放 SCL。

12.7.7 时钟延展

从机可以使用时钟同步机制来降低传输比特率。主机将 SCL 驱动为低电平后，从机可以在所需的时间内将 SCL 驱动为低电平，然后释放它。如果从机 SCL 低电平周期长于主机 SCL 低电平周期，则延长所得到的 SCL 总线信号低电平周期。



图表 12-18：SCL 同步

12.7.8 高速模式操作

I2C 模块可以在高速模式下以高达 3.4Mbits/s 的比特率传输数据，在混合传输的总线系统中，可以与进行双向通信的快速/标准模式（F/S 模式）设备向下兼容。除了在 HS 模式传输期间不执行仲裁和时钟同步外，其他时候与 F/S 模式系统保持相同的串行总线协议和数据格式。

HS 模式下的串行数据传输格式符合标准模式 I2C 总线规范。只有满足以下条件（所有条件均在 F/S 模式下发生）后，HS 模式才能开始：

START 信号（S）
8 位主机码（00001XXX）
不应答位（A）

(图 21-5 HS 模式下的数据传输格式）和（图 21-6 一次完整的 HS 模式传输）对此进行了更详细的说明。HS 主机码具有两个主要功能：

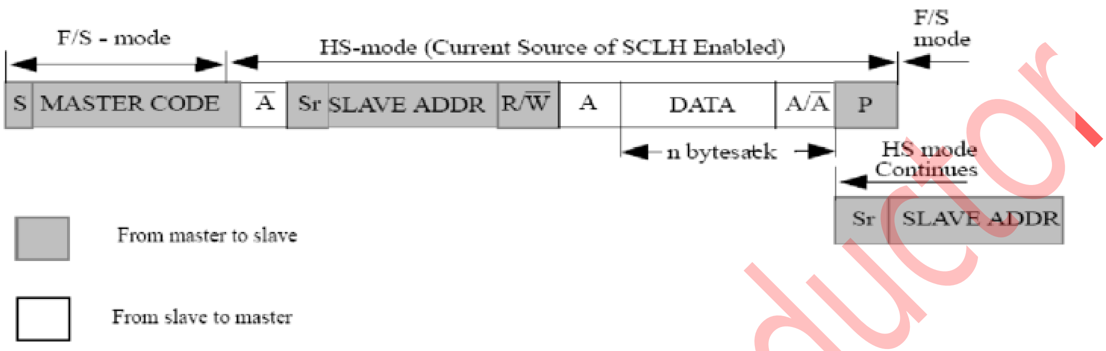
它允许在竞争主机之间以 F/S 模式速度进行仲裁和同步，从而产生一个获胜主机。
它指示 HS 模式传输的开始

HS 模式主机码是保留的 8 位代码，不用于从机寻址或其他目的。此外，由于每个主机都有自己独特的主机码，所以在一个 I2C 总线系统上最多可以存在八个 HS 模式主机（但是应保留主机码 0000 1000 进行测试和诊断）。I2C 模块的主机码是软件可编程的。仲裁和时钟同步仅在发送主机码和不应答位（A）的过程中发生，此后，获胜的主机保持活动状态。主机码向其他设备指示 HS 模式传输将开始，并且连接的设备必须符合 HS 模式规范。由于不允许任何设备应答主机码，因此主机码后跟了一个不应答位（A）。在不应答位（A）且 SCL 线已被拉高为高电平后，活跃主机切换到 HS 模式并启用（在时间 tH 处，参考（图 21-6 一次完整的 HS 模式传输））SCL 信号的电流源上拉电路。由于其他设备可以通过延长 SCL 信号的慢周期来延迟 tH 之前的串行传输，因此当所有设备都释放了 SCL 线并且 SCL 信号达到高电平时，有源主机将启用其电流源上拉电路，从而加快了 SCL 信号上升时间的最后一部分。然后，活跃主机发送重复 START 信号（Sr），后跟一个具有 R/W 位的 7 位从机地址，并从所选从机接收一个应答位（A）。

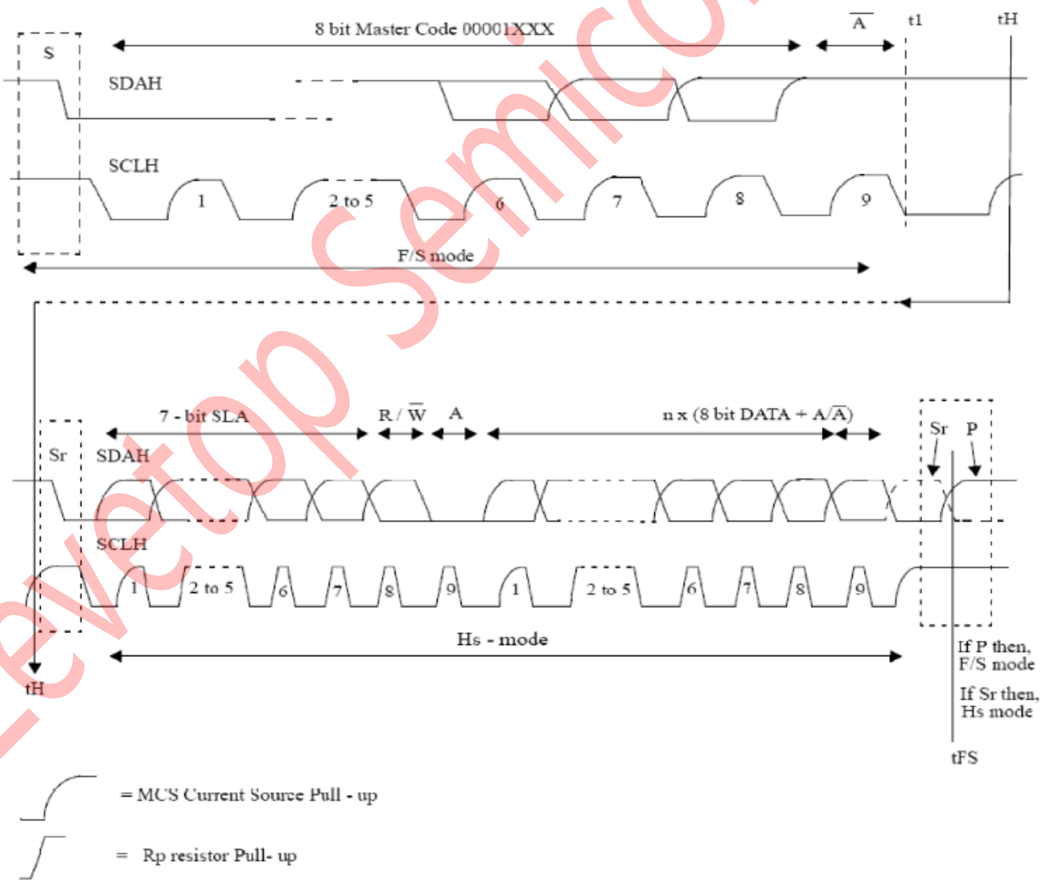
在重复 START 操作之后，以及在每个应答位（A）或不应答位（A）之后，活跃主机禁用其电流源上拉电

路。这使得其他设备可以通过延长 SCL 信号的低周期来延迟串行传输。当所有器件都已释放并且 SCL 信号达到高电平时，有源主控器再次重新启用其电流源上拉电路，从而加快了 SCL 信号上升时间的最后一部分。

在下一次重复的 START 信号（Sr）之后，数据传输以 HS 模式继续，并且仅在 STOP 信号（P）之后才切换回 F/S 模式。为了更效率的利用主机码，主设备可能会链接多个 HS 模式传输，并由重复的 START 条件（Sr）分隔。



图表 12-19：HS 模式下的数据传输格式



图表 12-20：一次完整的 HS 模式传输

12.7.9 10 位寻址

10 位从机地址由 START 条件（S）或重复 START 条件（Sr）下的前两个字节组成。第一个字节的前 7 位是组合 11110XX，其中后两位（XX）是 10 位地址的两个最高有效位（MSB）；第一个字节的第八位是读/写位，它确定消息的方向。第一个字节的最低有效位置为“0”表示主机将信息写入所选的从机；为“1”则表示主机将从从机读取信息。如果 R/W 位为“0”，则第二个字节将包含 10 位地址的其余 8 位（XXXXXXXX）。如果 R/W 位为“1”，则第二个字节包含从机向主机发送的数据。

在包含 10 位寻址的传输中，读/写格式的各种组合都是可能的。可能的数据传输格式为：

主发送器发送 10 位从机地址到从接收器

传输方向不变（见图 21-7）。当 10 位地址遵循 START 条件时，每个从机将从机地址的第一个字节的前七个位（11110XX）与自己的地址进行比较，并测试第八个位（读/写方向位）是否为 0。一个以上的设备可能会找到一个匹配项并生成一个应答信号（A1）。找到匹配项的所有从机都会将从机地址第二个字节的八位（XXXXXXXX）与自己的地址进行比较，但是只有一个从机会找到匹配项并产生应答（A2）。匹配的从机将一直被主机寻址，直到收到 STOP 条件（P）或重复 START 条件（Sr），然后是主机发送另一个从机地址。

主接收器从从发送器读取 10 位从机地址

在第二个读/写位之后更改传输方向（图 21-8），直到包含应答位 A2。该过程与主发送器寻址从接收器的过程相同。在重复 START 条件（Sr）之后，匹配的从站会记住之前已对其寻址的主机。然后，该从机检查 Sr 之后的从机地址第一个字节的前 7 位是否与 START 条件（S）之后的从机地址相同，并测试第八位（读/写）是否为 1。如果匹配，则从机认为已将其寻址为发送器并生成应答信号 A3。从机发送器保持寻址状态，直到它接收到一个 STOP 条件（P），或者直到它接收到另一个重复 START 条件（Sr），后跟一个不同的从机地址。在重复 START 条件（Sr）之后，所有其他从机也将从机地址（11110XX）的第一个字节的前 7 位与自己的地址进行比较，并测试第 8 位（读/写位）。但是，由于读/写位为 1（对于 10 位设备）或 11110XX 从机地址（对于 7 位设备）不匹配，因此不会寻址它们。

组合模式

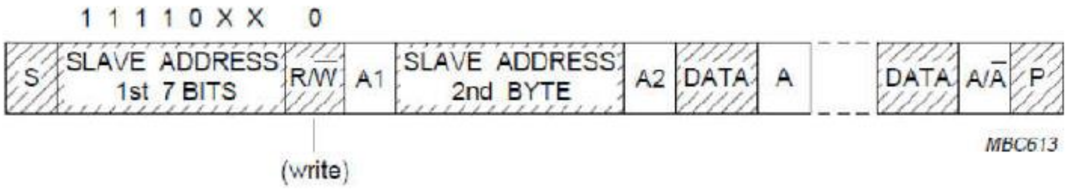
主机将数据发送到从机，然后从同一从机读取数据（图 21-9）。相同的主机始终占用总线。在第二个读/写位之后更改传输方向。

组合模式

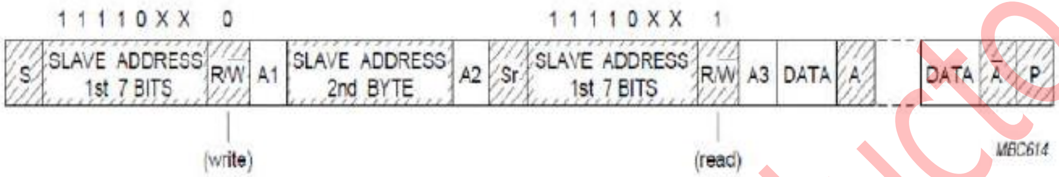
主机将数据传输到一个从机，然后再将数据传输到另一个从机（图 21-10）。同一主机始终占用总线。

组合模式

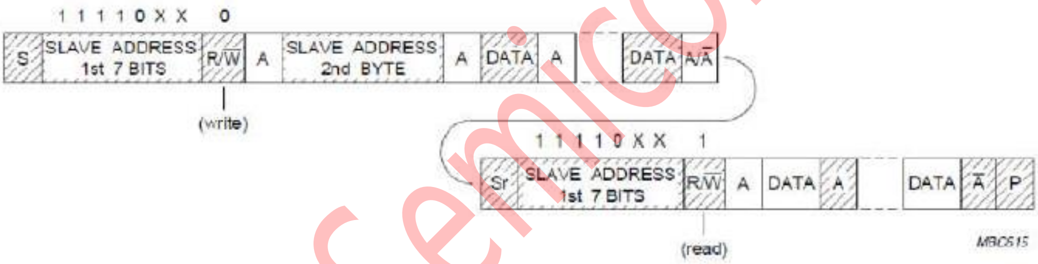
通过一次串行传输将 10 位和 7 位寻址组合在一起（图 21-11）。在每个 START 条件（S）或每个重复 START 条件（Sr）之后，可以发送 10 位或 7 位从机地址。图 21-11 显示了主机如何将数据传输到具有 7 位地址的从机，然后再将数据传输至具有 10 位地址的第二个从机。同一主机始终占用总线。



图表 12-21：主发送器使用 10 位地址寻址从接收器

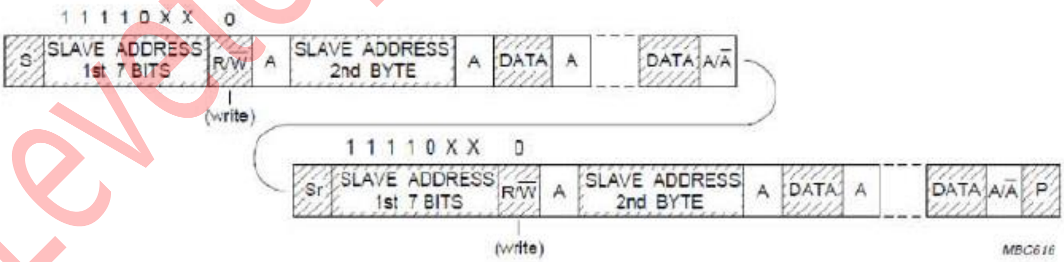


图表 12-22：主接收器使用 10 位地址寻址从发送器



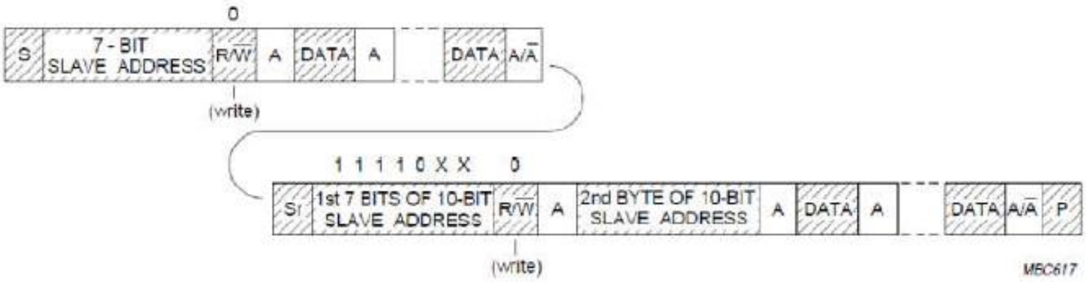
图表 12-23：组合模式

主机用 10 位地址寻址从机，然后将数据传输到该从机并从该从机读取数据



图表 12-24：组合模式

主机向两个从机发送数据，两个从机均具有 10 位地址

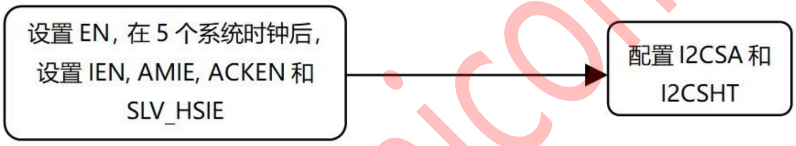


图表 12-25：组合模式

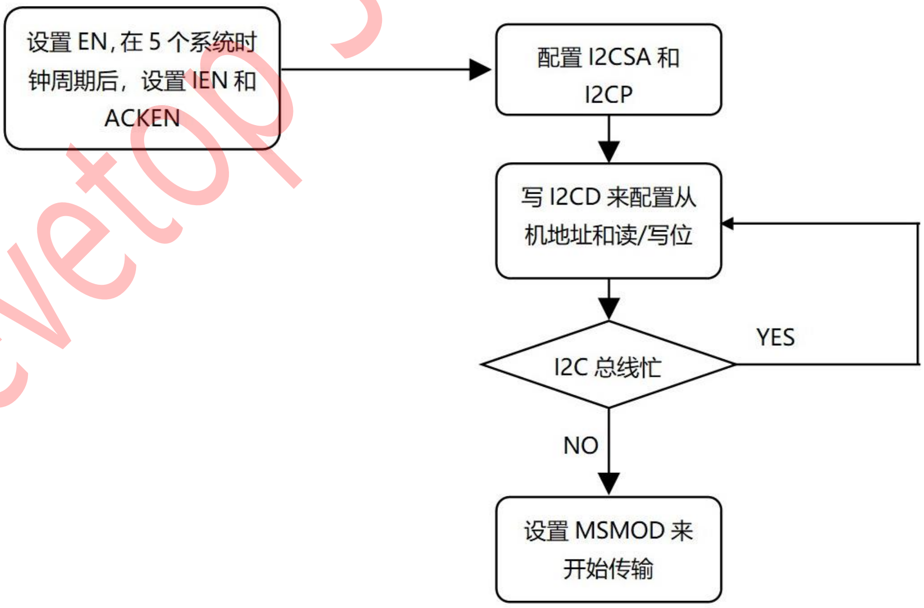
主机将数据传输到两个从机，第一个地址为 7 位，第二个地址为 10 位

12.7.10 软件工作流程图

初始化

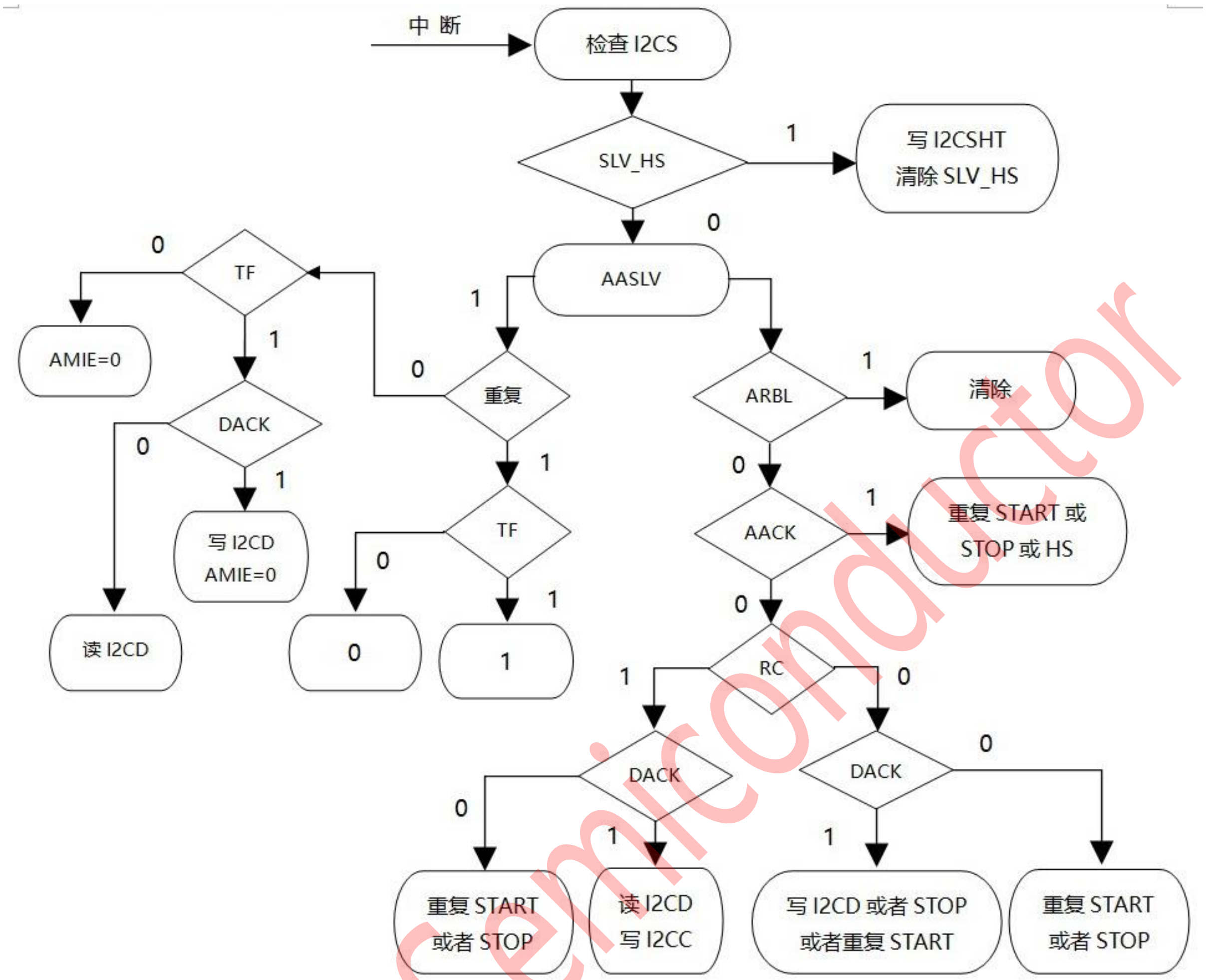


图表 12-26：从机模式初始化



图表 12-27：主机模式初始化

中断流程



图表 12-28：中断流程

如果 I2C 发生中断，首先检查 I2CS 以确认 I2C 处于主机模式还是从机模式。检查 SLV_HS 状态，如果设置了 SLV_HS，则写 I2CSHT 以配置高速计时器，并向 I2CSHIR 写 1 以清除 SLV_HS。

在从机模式下，如果已设置 RC，则表示 I2C 处于从机接收器模式，然后检查 TF，如果还设置了 TF，则表示已接收数据（非地址），然后读取 I2CD 并写 I2CC 以清除 RC。如果未设置 TF，则表示仅接收到从机地址，然后写 I2CC 清除 RC 和 AMIE，接收下一个字节数据。

在从机模式下，如果未设置 RC，则检查 TF，如果已设置 TF，则检查 DACK，如果还设置了 DACK，则表示 I2C 处于从机发送模式，然后写 I2CD 清除 TF，写 I2CC 以清除 AMIE 并发送下一个字节数据。如果尚未设置 DACK，则读 I2CD 以获取最后一个接收的字节以清除 TF。在从机模式下，如果未同时设置 TF 和 RC，则写 I2CC 以清除 AMIE。

在主机模式下，如果 ARBL 置位，则表示仲裁已经发生，然后写 I2CC 以清除 MSMOD。如果未置位 ARBL，则检查 AACK，如果 AACK 置位，则表示地址应答错误，向 I2CC 写重复 START 或 STOP。

如果尚未设置 ARBL 和 AACK，则检查 RC，如果已设置 RC，则表示 I2C 处于天线接收器模式，然后检查 DACK，如果还设置了 DACK，则读 I2CD 清除 RC 并写 I2CC 以选择是否确认数据。如果尚未设置 DACK，则写 I2CC 以重复 START 或 STOP。

12.8 中断描述

下表列出了和 I2C 模块有关的中断请求。

表格 12-2：I2C 中断请求源

中断源	标志	使能位
传输完成	TF	IEN
接收器	RC	IEN
从机地址匹配	AMI	AMIE
从机高速模式	SLV_HS	SLV_HSIE

12.8.1 传输完成（TF）中断

在 IEN 为 1 时，I2C 模块完成数据的传送或者接收发生中断。接收时，在接收到数据字节的第九个时钟下降沿时置位 TF。主机发送时，在发送的数据或者地址字节的第九个时钟下降沿时置位 TF。从机发送时，在受到确认位后，发送的数据或者地址字节的第九个时钟下降沿时置位 TF。读 I2CS 清 0。

12.8.2 接收器（RC）中断

在 IEN 为 1 时，I2C 被配置为接收器时产生中断。主机接收时，RC 在接收数据或地址字节的第九个时钟下降沿置位。从机接收时，在检测到确认位后，RC 在接收数据或地址字节的第九个时钟下降沿置位。读 I2CS 清 0。

12.8.3 从机地址匹配（AMI）中断

当 IEN 为 1 时，在进入 STOP 模式前应置位 AMIE 以便在从机地址匹配时唤醒 I2C 模块。在正常工作模式下 AMIE 应清 0。

12.8.4 从机高速模式（SLV_HS）中断

在 IEN 为 1 时，若置位 SLV_HSIE，则从机在高速模式下会产生一个中断。接收到不匹配位后，SLV_HS 在主机模式的第九个 SCL 上升沿置位。写 1 清 0。

13 复位控制器模块（RESET）

13.1 概述

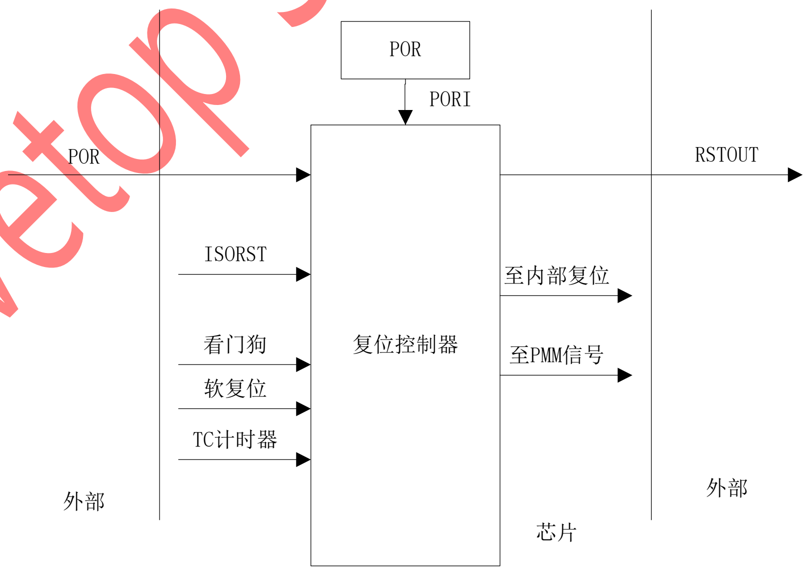
复位控制器模块用来判断复位原因，向系统判断提示精确的复位信号，并将复位原因进行保存。芯片上次复位的状态位保存在复位模块里。

13.2 特性

模块特性包括：

复位的触发源
上电复位
软件
看门狗模块复位
TC 计时器复位
7816 ISORST 复位
高、低频率检测复位
金属屏蔽网复位
芯片完成复位状态后，产生 RSTOUT 信号
可让软件检查上次复位原因的状态标志位

13.3 框图



图表 13-1：复位控制块框图

13.4 外部管脚

下表是复位控制器信号属性的总结。下面的段落是对信号的描述：

表格 13-1：复位控制器信号属性

名称	描述	输入滞后	输入同步
POR 管脚	I	Y	N
RSTOUT 管脚	O	—	—

13.4.1 POR

设置该信号为低电平将导致系统立刻复位，为了有效复位芯片，推荐低电平维持至少 1ms 以上。

13.4.2 RSTOUT

当内部复位控制模块复位芯片时，该输出信号显示低电平。

13.5 内存映射和寄存器

13.5.1 内存映射

基地址是 0x63F08000 复位控制器偏移地址映像如下表所示：

表格 13-2：复位控制器偏移地址映像

偏移地址	位 7-0	访问权限
0x0000	复位控制寄存器(RCR)	S/U
0x0001		
0x0002		
0x0003		
0x0004	保留	S/U
0x0005	保留	S/U
0x0006	复位测试寄存器(RTR)	S/U
0x0007	复位状态寄存器(RSR)	S/U

注意： S = 超级用户访问；U = 普通用户访问

13.5.2 寄存器描述

13.5.2.1 复位控制寄存器

复位控制寄存器（RCR）使得软件控制能够请求复位，能够独立判断外部 RSTOUT 管脚，以及 SIM 卡复位启用功能。

偏移地址：0x0000~0x0003　　复位值：0x00000000

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

图表 13-2：复位控制寄存器（RCR）

注意：只有上电复位能够重置 CRE，CRWE 位。

比特位	名称	复位值	读写属性	功能说明
[31]	SOFTRST	0x0	RW	软件复位请求位 设置该位会引起复位，同时也会清消除该位 0 = 无软复位；1 = 软复位
[30]	FRCRSTOUT	0x0	RW	外部 FORCE RSTOUT 管脚有效使能位 0 = RSTOUT 管脚无效 1 = RSTOUT 管脚有效
[29:24]	保留	0x0	RO	---
[23]	CRE	0x0	RW	SIM 卡复位功能使能控制位 注意该位只对 USI1 管脚有效 0 = 禁止；1 = 使能
[22]	CRWE	0x0	RW	SIM 卡唤醒功能使能控制位 注意该位只对 USI1 管脚有效 0 = 禁止；1 = 使能
[21:12]	保留	0x0	RO	---
[11]	保留	0x0	RO	---
[10]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[9]	保留	0x0	RO	---
[8]	保留	0x0	RO	---
[7:4]	保留	0x0	RO	---
[3]	保留	0x0	RO	---
[2]	保留	0x0	RO	---
[1]	保留	0x0	RO	---
[0]	保留	0x0	RO	---

13.5.2.2 复位测试寄存器

复位测试寄存器（RTR）只适用于工厂检测，在非测试模式下可读。

偏移地址：0x0006　　复位值：0x00

7	6	5	4	3	2	1	0
FPOR	保留						保留
rw	ro						ro

图表 13-3：复位测试寄存器（RTR）

比特位	名称	复位值	读写属性	功能说明
[7]	FPOR	0x0	RW	系统复位请求位 设置该位会引起复位，同时也会清消除该位。 0 = 无系统复位；1 = 系统复位
[6:1]	保留	0x0	RO	---
[0]	保留	0x0	RO	---

13.5.2.3 复位状态寄存器

复位状态寄存器（RSR）包含了每一个复位源对应的状态位。当复位完成后，产生复位的复位源会被锁存，并未引发复位的其他复位源对应的状态位值为 0。

偏移地址：0x0007　　　复位值：0x00

7	6	5	4	3	2	1	0
TCR	保留	SOFT	WDR	POR	保留	CR	保留
ro	ro	ro	ro	ro	ro	ro	ro

图表 13-4：复位状态寄存器（RSR）

比特位	名称	复位值	读写属性	功能说明
[7]	TCR	0x0	RO	TC 计时器复位检测指示位 标识前一次系统复位是否由 TC 计时器触发引起 0 = 不是由 TC 计时器引起 1 = 由 TC 计时器引起
[6]	保留	0x0	RO	---
[5]	SOFT	0x0	RO	软复位检测指示位 标识前一次系统复位是否由软复位引起 0 = 不是由软复位引起 1 = 由软复位引起
[4]	WDR	0x0	RO	看门狗复位检测指示位 标识前一次系统复位是否由看门狗触发引起 0 = 不是由看门狗引起 1 = 由看门狗引起
[3]	POR	0x0	RO	外部上电复位检测指示位 标识前一次系统复位是否由外部上电复位引起 0 = 不是由外部上电复位引起 1 = 由外部上电复位引起
[2]	保留	0x0	RO	---
[1]	CR	0x0	RO	SIM 卡复位检测指示位 标识前一次系统复位是否由 SIM 卡复位引起 0 = 不是由 SIM 卡复位引起 1 = 由 SIM 卡复位引起
[0]	保留	0x0	RO	---

13.6 功能描述

13.6.1 复位源

表格 13-3：复位源概述

复位源	类型
POR 管脚	异步
ISORST 管脚	异步
看门狗模块	同步
软件	同步
TC 计时器	异步

为了保护数据完整性，由复位控制逻辑控制的同步复位一直到当前总线周期结束时，才会开始运行。该周期结束后，复位由系统时钟的下一次的上升沿决定。任何时候复位控制逻辑都必须到总线周期结束后执行，无论芯片配置模块 CCR 寄存器中的 BME 位状态如何，内部总线监控器都会自动使能。如果当前总线周期没有正常结束，总线监控器会根据时间长度进行关闭。这个时间长度可在 CCR 寄存器的 BMT 位进行编程。

当出现同步复位时，正在执行的内部字节/半字/字写入必须完成，且要保证数据不被破坏。外部写入操作，包括对 16 位端口的字写入，也要保证能够完成。异步复位源通常表明严重的错误，因此，复位控制逻辑不会等当前的总线周期完成，复位会立即生效。

13.6.1.1 上电复位

上电后，复位控制器将控制 RSTOUT 输出低电平，直到 POR 达到合适的电平强度，该电平强度的到达时间可以通过 POR 管脚外接阻容电路进行调整。上电后约 1ms，RSTOUT 就会输出高电平，系统开始正常运行。持续拉低或拉高 POR 管脚不会损坏器件。

13.6.1.2 SIM 卡复位

ISORST 管脚输入信号引起 SIM 卡复位后，复位信号被识别和锁定。总线监控器使能，当前总线忙周期完成。如果 LRESET 管脚无效，复位控制器将控制 RSTOUT 大约 10240 个周期，然后系统开始正常运行。

13.6.1.3 看门狗模块复位

看门狗模块引起复位后，复位信号被识别和锁定。总线监控器使能，当前总线忙周期完成。如果 LRESET 管脚无效，复位控制器将控制 RSTOUT 大约 10240 个周期，然后系统开始正常运行。

13.6.1.4 软件复位

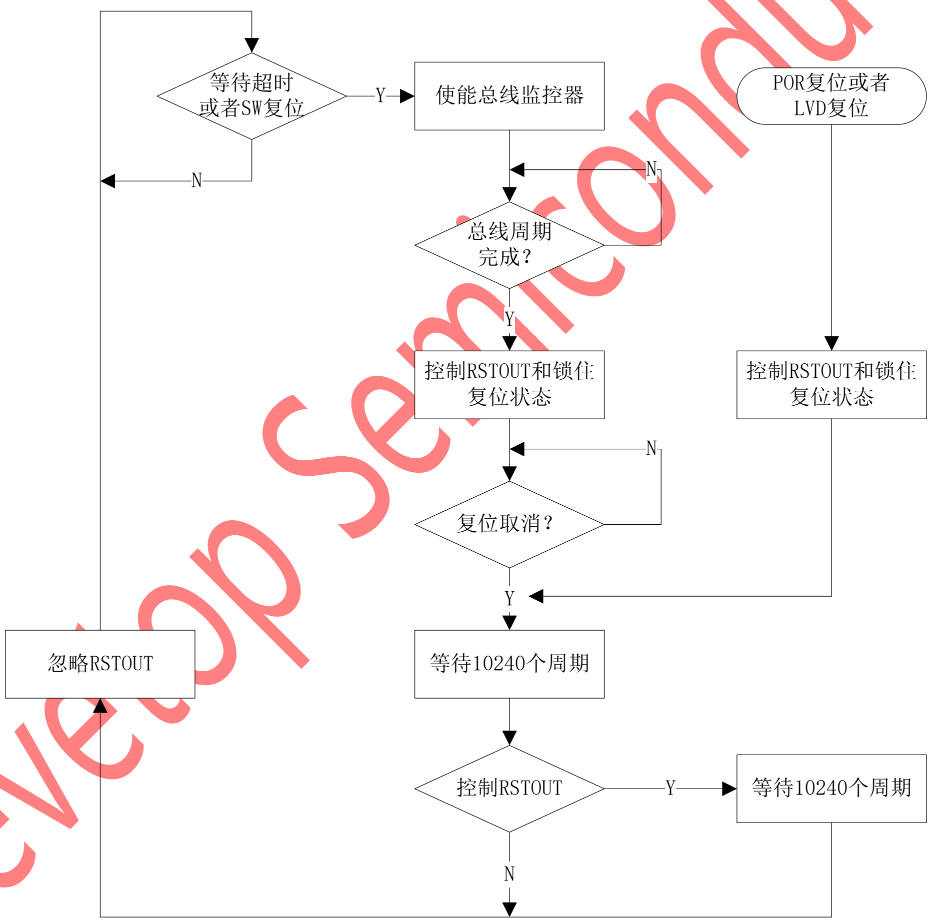
当 SOFTRST 位设置时，会出现软件复位。复位控制器将控制 RSTOUT 大约 8192 个周期，然后系统开始正常运行。

13.6.1.5 TC 计时器复位

当 TC 计时器引发复位，复位控制器将控制 RSTOUT 大约 10240 个周期，然后系统开始正常运行。

13.6.2 复位控制流程图

复位逻辑控制流程如下图所示，其中，给出的所有循环计数值均为近似值。



图表 13-5：复位控制流程图

14 模数转换器（ADC）

14.1 概述

12 位 ADC 是一个逐次逼近的模数转换器。它具有多达 9 个通道，可测量来自 8 个外部源和 1 个内部源的信号。这些通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 12 位*8 深度的 FIFO 中。

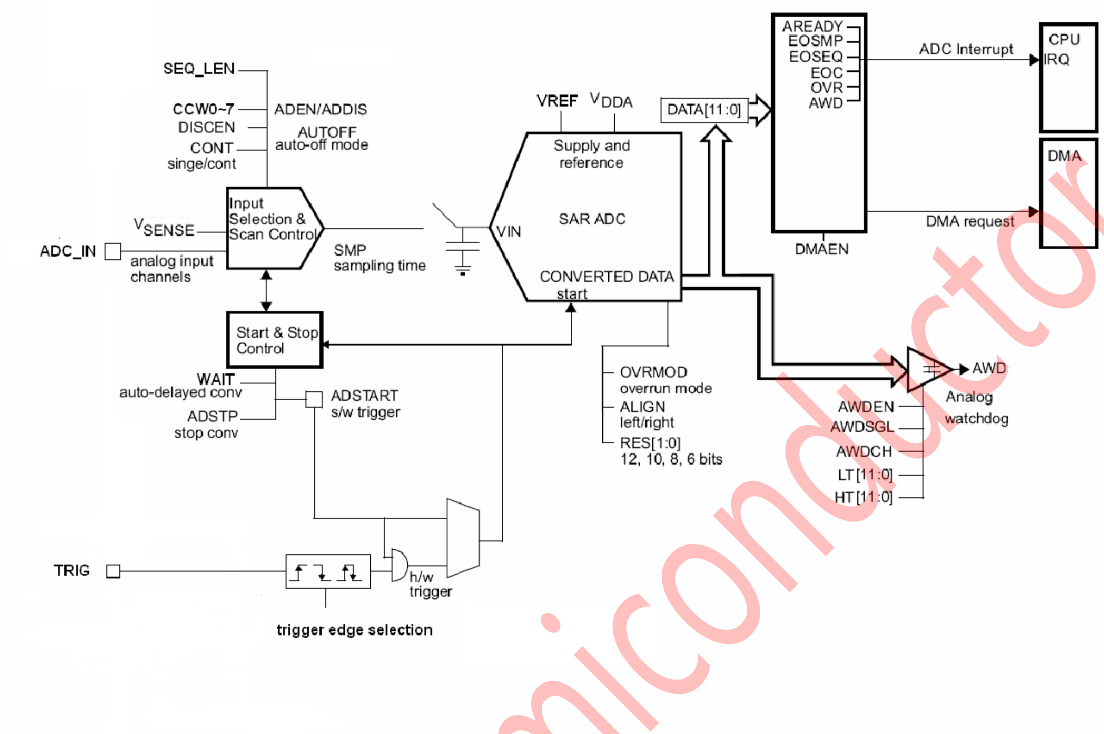
ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。ADC 进入低功耗模式后，可以在低频情况下工作以降低功耗。

14.2 特征

高性能
可配置 12 位、10 位、8 位或 6 位分辨率
ADC 转换时间：12 位分辨率（1MHz）时为 1.0 微秒，10 位分辨率转换时间为 0.88 微秒，通过降低分辨率可以获得更快的转换时间
可编程采样时间
数据对齐以保持内置数据一致性
支持 DMA
低功耗
可以在低功耗运行时降低 PCLK 的频率来保持最佳的性能。比如可以在不论 PCLK 的频率情况下，保持 ADC 的 1.0 微秒的转换时间
等待模式：使用低频的 PCLK 来防止 ADC 在应用中溢出
自动关闭模式：除了主动转换阶段外，ADC 会自动关闭来降低功耗
模拟输入通道
8 个外部模拟输入通道
1 个内部源检测通道
初始化转换
软件
可配置极性的外部硬件触发器
转换模式
可以转换单一信道或者扫描一系列信道
单信道模式每触发一次转换选定的输入
连续模式可连续转换选定的输入
不连续模式
在采样结束、转换结束、序列转换结束以及发生模拟看门狗或溢出事件时产生中断
模拟看门狗
单端和差分输入配置
转换器可选择使用内部参考或外部参考
与 DMA 兼容的数据收集功能

14.3 ADC 功能描述

图 28-1 展示了 QADC 框图。



图表 14-1：ADC 框图

14.3.1 ADC 开关控制（ADEN，ADDIS，ADRDY）

当 MCU 上电时，ADC 被禁用并进入掉电模式（ADEN = 0）。如图表 14-2：所示，ADC 开始准确转换之前需要稳定时间 tSTAB（~2.0μs）。

2 个控制位用来使能和禁用 ADC：

置 ADEN = 1 来使能 ADC。标志位 ADRDY 在 ADC 准备就绪后置 1。

置 ADDIS = 1 来禁用 ADC 同时设置 ADC 为掉电模式。ADC 完全禁用后，硬件将自动清除 ADEN 和 ADDIS 位。

然后就可以通过设置 ADSTART = 1 或在启用了触发器情况下发生外部触发事件时开始转换。

按以下步骤使能 ADC：

设置 ADC_CR 寄存器里的 ADEN = 1。

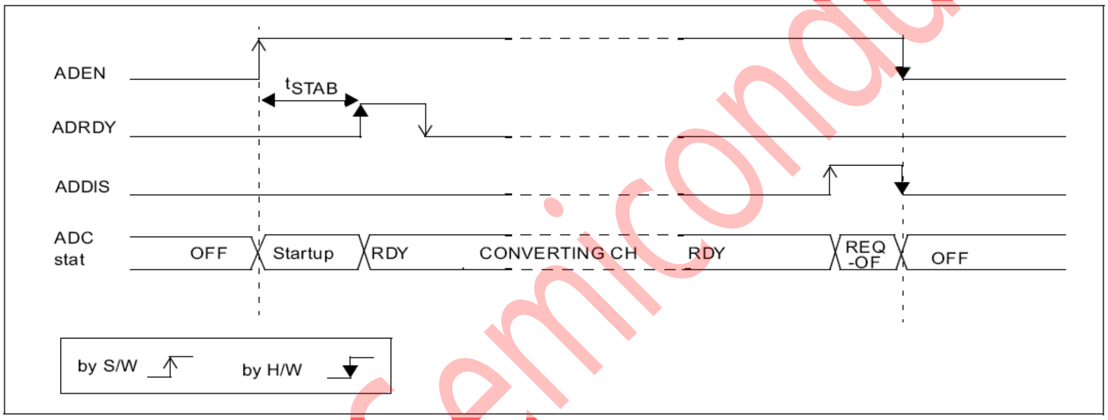
等待直到 ADC_ISR 寄存器中的 ADRDY = 1（ADRDY 将在 ADC 启动时间后置位）。在中可以通过设置 ADC_IER 里的 ADRDYIE 位使能中断来处理。

按以下步骤禁用 ADC：

首先检查 ADC_CR 中的 ADSTART = 0 来确认此时没有转换在进行。如果需要，可以通过向 ADC_CR 里的 ADSTP 位写 1 并等待直到该位读为 0 来结束当前进行的转换。

设置 ADC_CR 里的 ADDIS = 1。

如果应用需要，等待直到 ADC_CR 的 ADEN = 0，此时表明 ADC 已完全禁用（当 ADEN = 0 时 ADDIS 将自动复位）。

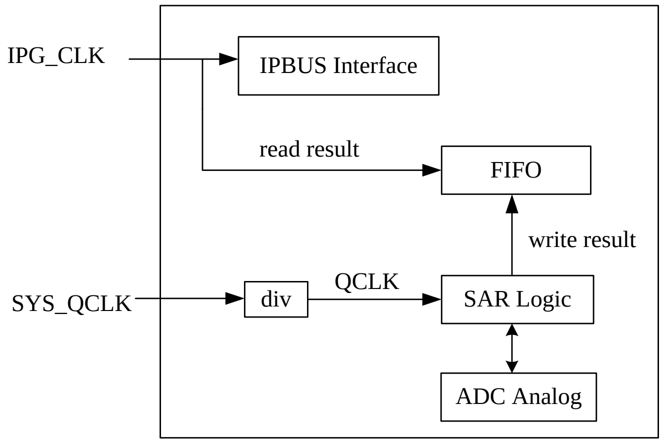


图表 14-2：使能/禁用 ADC

注意：在自动关闭模式下（AUTOFF = 1），开/关机阶段由硬件自动执行且不清楚 ADRDY 标志位。

14.3.2 ADC 时钟

ADC 有双时钟域架构，如图 28-3 所示，无论是否选择 IPG 时钟方案，都有能达到最大 ADC 时钟频率的优势。



图表 14-3：ADC 时钟方案

14.3.3 配置 ADC

当 ADC 被禁用时（ADEN 位必须为 0），软件只能向 ADC_CR 里写 ADEN 位。当 ADC 使能且没有未处理的禁用请求时（ADEN = 1 和 ADDIS = 0），软件只能写 ADC_CR 中的 ADSTART 位和 ADDIS 位。

对于在 ADC_IER，ADC_CFGRi，ADC_SMPR，ADC_TR，ADC_CHSELRi 和 ADC_WDG 中的其他控制位，软件只能在没有转换进行时才能写这些控制位。

当 ADC 使能（且有可能在转换）同时没有未处理的禁用请求时，软件只能写 ADC_CR 中的 ADSTP 位。

注意：没有硬件保护措施来防止软件在上述规则中的违规写操作。如果发生了非法写访问，ADC 可能会进入不确定状态。此时为了恢复正确操作，必须禁用 ADC（ADDIS = 1）。

14.3.4 通道选择（CCWi）

ADC 中有多达 17 个多路通道：

16 个来自管脚（ADC_IN0...ADC_IN15）模拟输入
1 个内部模拟输入（温度传感器）

ADC 可以转换单个通道或者自动扫描一系列信道。序列通道可以从 CCW[0]开始，然后是 CCW[1]，...，CCW[7]。CCWi 可在 ADC_CHESELRi 中选择控制。序列长度在 ADC_CFGR1 中的 SEQ_LEN[2:0]选择控制。比如，如果序列长度设置为 3，则序列由 CCW[0]开始，然后是 CCW[1]，然后是 CCW[2]。通道码如表 28-1 所示。

表格 14-1：通道码

CCWi[4:0]	通道选择
5’b00000	ADC_IN0
5’b00001	ADC_IN1
5’b00010	ADC_IN2
5’b00011	ADC_IN3
5’b00100	ADC_IN4
5’b00101	ADC_IN5
5’b00110	ADC_IN6
5’b00111	ADC_IN7
5’b01000	ADC_IN8
5’b01001	ADC_IN9
5’b01010	ADC_IN10
5’b01011	ADC_IN11
5’b01100	ADC_IN12
5’b01101	ADC_IN13
5’b01110	ADC_IN14
5’b01111	ADC_IN15
5’b10000	VBATTERY DETECT

14.3.5 可编程采样时间（SMP）

在开始转换之前，ADC 需要在测量电压源和 ADC 的嵌入式采样电容器之间建立一个直连。采样时间必须满足输入电压源对样品充电并将电容器保持在输入电压水平。

可编程的采样时间允许根据输入电压源的输入电阻来调整转换速度。可以使用 ADC_SMPR 中的 SMP[3:0] 位修改 ADC 对多个 ADC 时钟周期的输入电压进行采样。可编程采样时间适用于所有通道。ADC 通过设置 EOSMP 标志位来指示采样阶段结束。

14.3.6 单次转换模式（CONT = 0）

在单次转换模式中，ADC 仅转换序列一次。当 AD_CFGR1 中的 CONT = 0 时选择此模式。转换将开始于：

设置 ADC_CR 里的 ADSTART 位

硬件触发事件

在序列转换过程中，每次转换完成后：

转换数据被储存于 FIFO 中
EOC（转换结束）标志位置位
如果 EOCIE 位被设置，将产生中断

在整个序列转换结束后：

EOSEQ（序列结束）标志位置位
如果 EOSEQIE 位被设置，将产生中断

之后 ADC 将停止运行直到一个新的外部触发事件发生或者 ADSTART 位被重新设置。
注意：为了单次转换一个通道，编辑序列长度为 1。

14.3.7 连续转换模式（CONT = 1）

在连续转换模式下，当发生一次硬件或者软件触发事件时，ADC 执行一个序列的转换，完成一个转换后将自动重新开始连续转换一个相同的序列。当 ADC_CFGR1 中 CONT = 1 时选择此模式。转换开始于：

设置 ADC_CR 中的 ADSTART 位
硬件触发事件

在序列转换过程中，每次转成完成后：

转换数据被储存于 FIFO 中
EOC（转换结束）标志位置位
如果 EOCIE 位被设置，将产生中断

在序列转换结束后：

EOSEQ（序列结束）标志位置位
如果 EOSEQIE 位被设置，将产生中断

在转换完成后 ADC 立即启动一个新的重复的连续转换。
注意：连续转换模式和不连续模式不能同时使能：即不能同时设置 DISCEN = 1 和 CONT = 1。

14.3.8 开始转换（ADSTART）

软件通过设置 ADSTART = 1 来开始 ADC 转换。当 ADSTART 被设置，转换将：

如果 TRGIMODE = 0x0（软件触发），转换立即开始
如果 TRGIMODE≠0x0，在所选硬件触发器的下一个有效沿开始
ADSTART 位同时可指示 ADC 是否在运行操作。ADSTART = 0 时，ADC 为空闲状态，此时可重新配置 ADC。
ADSTART 位被硬件清除：
在单次转换模式中软件触发
在任意序列转换结束后（EOSEQ = 1）
在不连续模式中软件触发
任意转换结束后
在所有情况下
在执行由软件调用的 ADSTP 程序之后

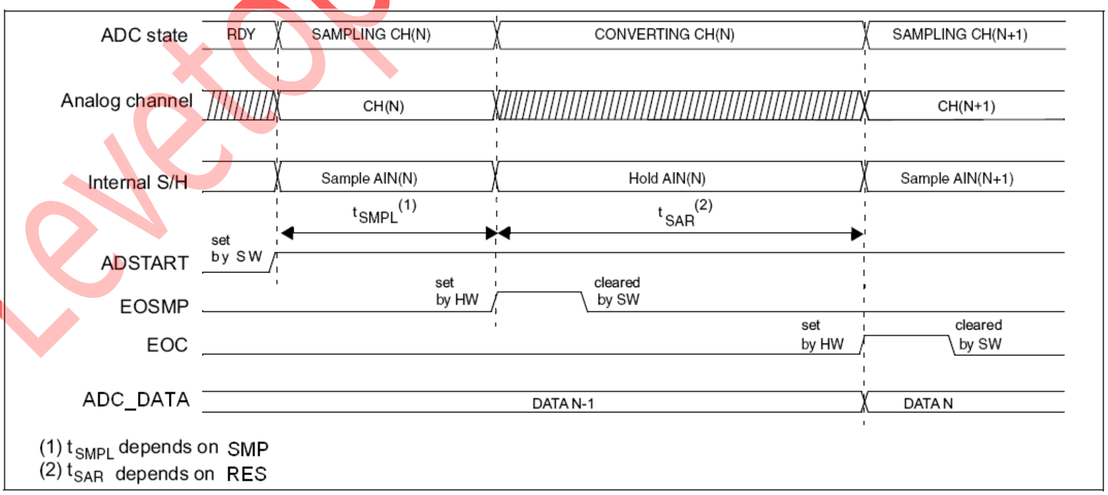
注意： 在连续模式中（CONT = 1），当 EOSEQ 标志位置位时，ADSTART 位不被硬件清除，此时序列会自动重新开始。

在单次转换模式中选择硬件触发时，如果 EOSEQ 标志位置位，ADSTART 不被硬件清除。这样就避免了需要再次软件设置 ADSTART 位同时确保了不丢失下一个触发事件。

14.3.9 时序图

从转换开始到转换结束之间经过的时间是配置的采样时间加上逐次逼近时间的总和，具体取决于数据分辨率：

tADC = tSMPL+tSAR = [4|min+12|12bit]*tQCLK = 1μs|min(for fQCLK = 16MHz)



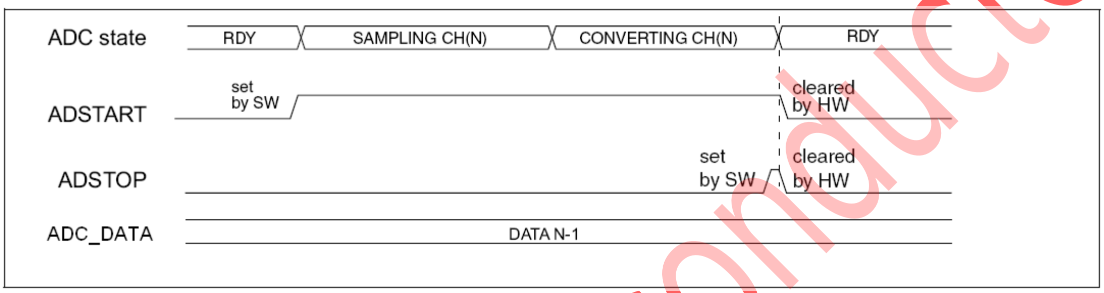
图表 14-4：ADC 转换时序

14.3.10 停止正在进行的转换（ADSTP）

软件可以通过在 ADC_CR 寄存器中设置 ADSTP = 1 来决定停止任何正在进行的转换。这将重置 ADC 操作，并且使 ADC 处于空闲状态，准备进行新操作。

当 ADSTP 位被软件置位时，任何正在进行的转换都将中止，结果都将被丢弃（FIFO 不会使用当前转换来进行更新）。扫描序列也将中止并重置（即重新启动 ADC 时将重启新的序列）。

完成此过程后，硬件将清除 ADSTP 和 ADSTART 位，软件必须等待 ADSTART = 0 后才能开始新的转换。
注意：QADC_ISR 中的标志位不会被 STOP 命令清除，FIFO 中的数据也不会丢失。



图表 14-5：停止正在进行的转换

14.4 外部触发转换和触发极性

除了通过软件，也可通过外部事件触发转换或序列。如果 TRIGMODE 控制位不为 0，外部事件能够以所选极性触发转换。设置 ADSTART = 1 时触发选项将生效。转换进行时发生的硬件触发将被忽略。

当 ADSTART = 0 时，发生任何硬件触发都被忽略。表格 13-2 提供了 TRIGMODE 值和触发极性之间的对应关系。

表格 14-2：配置触发极性

TRIGMODE[2:0]	SOURCE
3’ b000	禁止触发检测，软件触发
3’ b001	检测上升沿
3’ b010	检测下降沿
3’ b011	同时检测上升沿和下降沿
3’ b100	检测高电平
3’ b101	检测低电平
3’ b110	检测单次触发
3’ b111	保留

注意：外部触发极性只能在 ADC 没有进行转换（ADSTART = 0）时更改。

14.4.1 不连续采样模式（DISCEN）

通过设置 ADC_CFGR1 中的 DISCEN 位使能不连续采样模式。此模式下（DISCEN = 1），需要硬件或软件触发事件才能启动序列中定义的每个转换。相反，如果 DISCEN = 0，一个硬件或软件触发事件将连续启动序列中定义的所有转换。示例：

DISCEN = 1，要转换的通道 = 0，3，7，10
第 1 次触发：转换通道 0，生成 EOC 事件
第 2 次触发：转换通道 3，生成 EOC 事件
第 3 次触发：转换通道 7，生成 EOC 事件
第 4 次触发：转换通道 10，同时生成 EOC 和 EOSEQ 事件
第 5 次触发：转换通道 0，生成 EOC 事件
第 6 次触发：转换通道 3，生成 EOC 事件
...
DISCEN = 0，要转换的通道 = 0，3，7，10
第 1 次触发：转换整个序列：通道 0，3，7 最后是 10。每次转换生成 EOC 事件同时最后一次转换额外生成 EOSEQ 事件。
任意后续触发事件将重新开始完整序列转换。

14.4.2 可编程分辨率（RES）—快速转换模式

通过降低 ADC 分辨率，可以获得更快的转换时间（tSAR）。通过编程 ADC_CFGR1 中的 RES[1:0]位分辨率可以配置为 12 位，10 位，8 位或者 6 位。在数据精度要求不高的应用中，更低的分辨率将有更快的转换时间。

注意： RES[1:0]只有在 ADEN 复位时改变。转换的结果始终为 13 位宽，任何未使用的 LSB 均读为 0。较低的分辨率减少了逐次逼近步骤所需的转换时间。

14.4.3 转换结束，采样阶段结束（EOC，EOSMP 标志位）

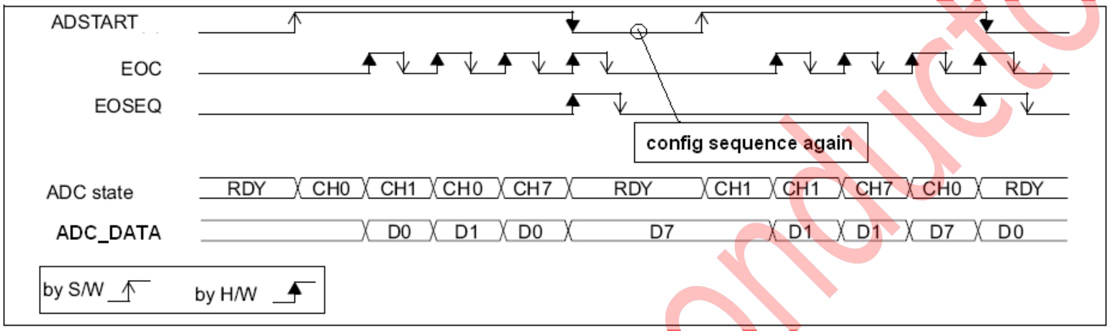
ADC 会标出每次转换结束（EOC）事件。只要有新的数据转换结果，ADC 就会设置 ADC_ISR 里的 ECO 标志位。当 ADC_IER 中的 EOCIE 被设置时，将会产生中断。EOC 标志位可以通过软件向此位写 1 或者读 FIFO 清除。

ADC 同样会通过设置 ADC_ISR 中 EOSMP 位标出采样阶段结束。EOSMP 可以通过软件写 1 清除。当 ADC_IER 中 EOSMPIE 被设置时，可以产生中断。

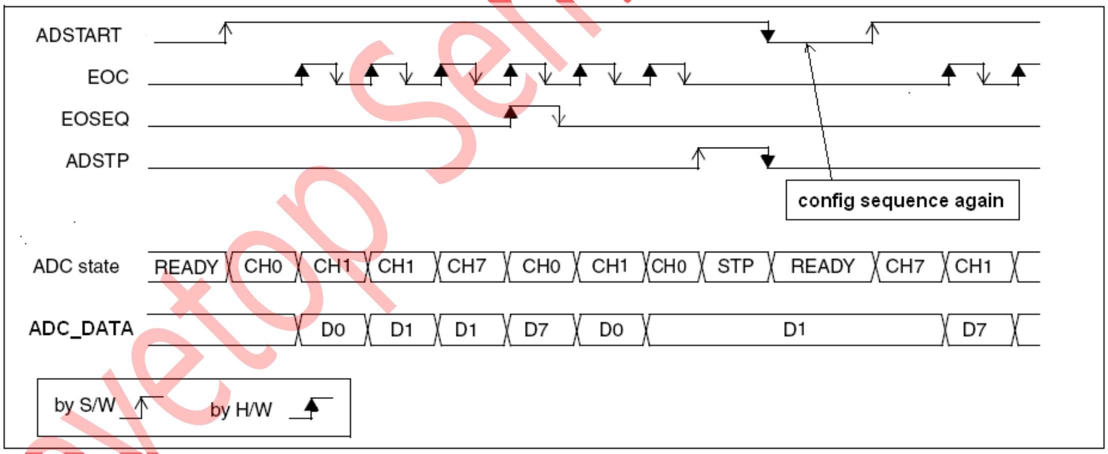
14.4.4 转换序列结束（EOSEQ 标志位）

ADC 会通知应用每个序列结束（EOSEQ）事件。只要在 FIFO 中获得了转换序列最后的数据结果，ADC 就将设置 ADC_ISR 中的 EOSEQ 标志位。ADC_IER 中 EOSEQIE 置位时，将产生中断。通过软件向此位写 1 清除 EOSEQ 位。

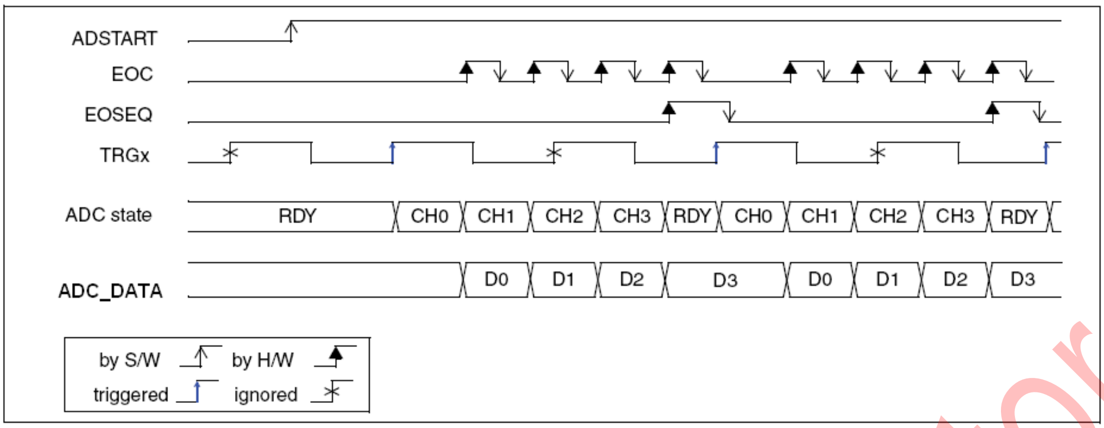
14.4.5 时序图示例（单次/连续模式硬件/软件触发）



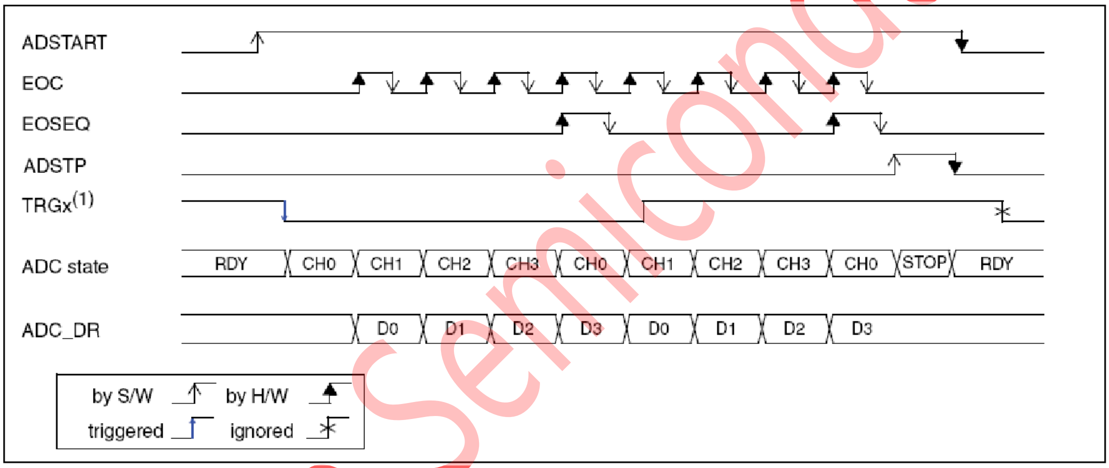
图表 14-6：单次序列转换，软件触发



图表 14-7：连续序列转换，软件触发



图表 14-8：单次序列转换，硬件触发



图表 14-9：连续序列转换，硬件触发

14.5 数据管理

14.5.1 数据 FIFO 和数据对齐（ADC_FIFO，ALIGN）

在每次转换结束（EOC 事件）后，数据结果存储在 13 位宽*8 深度的 ADC_FIFO 中。
读取数据的格式取决于配置的数据对齐格式和分辨率。

ADC_CFGR1 中 ALIGN 位选择数据转换后储存的对齐格式。数据可以如图 28-10 选择右对齐(ALIGN = 0)或者左对齐（ALIGN = 1）

ALIGN	RES[1:0]	31	30	…	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0x0			…					data[11:0]											
	0x1			…							data[9:0]									
	0x2			…									data[7:0]							
	0x3			…											data[5:0]					
1	0x0			…	data[11:0]															
	0x1			…	data[9:0]															
	0x2			…									data[7:0]							
	0x3			…									data[5:0]							

图表 14-10：数据对齐和分辨率

FIFO 支持字节，半字和字读，但是偏移地址应始终为 0x4c。对于不同的数据格式和分辨率，使用者需注意：

如果按字读，数据格式按图中 data[31:0]所示

如果按半字读，数据格式如果如图中 data[15:0]所示

如果按字节读且数据比 8 位长时，高位字节先读出，再读低位字节。

如果按字节读且数据比 8 位短时，数据格式如图中 data[7:0]所示。

14.5.2 ADC 溢出（OVR，OVRMOD）

溢出标志位（OVR）表示在 FIFO 满之前 CPU 或者 DMA 未及时读取转换后的数据时的数据溢出事件。

当一个新的转换完成时如果 FULL 标志位仍为‘1’，ADC_ISR 中的 OVR 标志位置位。如果同时 ADC_IER 中 OVRIE 位被设置，则产生中断。

当溢出发生时，除非软件通过设置 ADC_CR 中 ADSTP 位来决定停止并复位程序，否则 ADC 会继续工作并且继续转换。

OVR 位通过软件向此位写 1 清除。通过编程 ADC_CFGR1 中 OVRMOD 位可以配置在发生溢出时保留还是覆盖数据：

OVRMOD = 0

溢出事件防止数据寄存器被覆盖：此时保留旧数据，丢弃新的转换数据。如果 OVR 仍为 1，可以继续转换，但结果数据将被丢弃。

OVRMOD = 1

数据寄存器被最新的转换结果覆盖。如果 OVR 仍为 1，转换继续同时 FIFO 总是保存最新的转换数据。

14.5.3 管理不使用 DMA 的转换数据序列

如果转换足够慢，则转换序列可以由软件来处理。在这种情况下，软件可以使用 EOC 标志位和与其相关的中断来处理每个数据结果。每次转换完成时，ADC_ISR 中 EOC 位置位同时 FIFO 寄存器可读。

软件也可以使用 FIFO EMPTY 标志位来处理每个数据结果。如果 EMPTY 不为 0，即意味着 FIFO 中有新的数据。ADC_CFGR1 中 OVRMOD 位应配置为 0，来对溢出事件进行错误管理。

14.5.4 管理不使用 DMA 不溢出的转换数据

让 ADC 在转换一个或多个通道后读取数据而不是每次转换后读取数据就提高效率。在这种情况下，软件必须配置 OVRMOD 位为 1，忽略 OVR 标志位。当 OVRMOD = 1，溢出事件不会阻止 ADC 继续转换，FIFO 始终包含最新的转换数据。

14.5.5 管理使用 DMA 的转换数据

一旦 FIFO 中数据数量不为 0 且 DMAEN 位置位，QADC 将向 DMA 发送请求。这将允许将转后后的数据从 FIFO 传输到软件选择的目标位置。

尽管如此，如果因为 DMA 无法及时满足 DMA 传输请求而发生溢出时，ADC 停止生成 DMA 请求，且 DMA 将不会传输与新转换对应的数据。这意味着所有传输到 RAM 的数据都可以视为有效数据。通过配置 OVRMOD 位，可以决定数据是否保留或者覆盖。DMA 传输请求被阻止直到软件清除 OVR 位。

14.6 低功耗特性

14.6.1 等待模式转换

等待模式转换可以用于简化软件同时优化可能存在 ADC 溢出风险的低频时钟应用。当 ADC_CFGR1 中 WAIT 位被设置为 1，新的转换只能在 FIFO 未满时开始。这是一种使 ADC 速度自动适应读取数据的系统速度的方法。

注意：在转换进行中和读取访问之前的等待时间内发生的硬件触发将被忽略。

14.6.2 自动关闭模式（AUTOFF）

ADC 具有自动电源管理功能，称为自动关闭模式，通过设置 ADC_CFGR1 中 AUTOFF = 1 来使能。当 AUTOFF = 1 时，ADC 总会在没有转换时关闭，在转换开始时（硬件或软件触发）自动唤醒。在开始传输的触发事件和 ADC 采样时间自动插入一个启动时间。ADC 在一次序列采样完成后自动关闭。

自动关闭模式可以大幅减少仅需较少转换的应用的功耗，或者降低当转换请求的间隔时间足够长（比如低频硬件触发）因此需要额外电源和额外时间来开启和关闭 ADC 的应用的功耗。

对于低频时钟应用，自动关闭模式可以和等待模式转换（WAIT = 1）结合使用。如果在等待阶段自动关闭 ADC 电源，在应用读取 FIFO 后立即重启 ADC，可以节省大量功率。

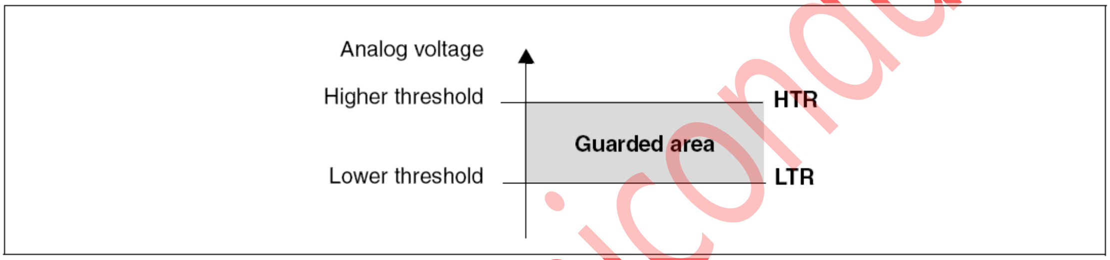
14.7 模拟看门狗（AWDEN，AWDSGL，AWDCH，...，AWD）

AWD 模拟看门狗特性通过设置 ADC_WDG 中 AWDEN 位使能。它用于监视一个选定的通道或所有启用的信道是否保持在配置的电压范围（窗口）内，如图 28-11 所示。

AWD 模拟看门狗的状态位将在模拟电压低于最低阈值和高于最高阈值时设置。这些阈值的值由 ADC_TR 设置。如果 ADC_IER 中 AWDIE 置位，此时可以使能中断。AWD 标志位通过软件向此位写 1 清除。

当转换数据的分辨率小于 12 位（通过 RES[1:0]位设置）时，必须清除已编程阈值的 LSB，因为内部比较总是对完整 12 位原始转换数据执行的（左对齐）。

表 28-3 展示了如何配置 ADC_WDG 的 AWDSGL 和 AWDEN 位来使能单个或多个通道的模拟看门狗。



图表 14-11：模拟看门狗监视范围

表格 14-3：模拟看门狗通道选择

模拟看门狗保护通道	AWDSGL 位	AWDEN 位
无	x	0
所有通道	0	1
单通道 (1)	1	1

(1)： 通过 AWDCH 选择

14.8 温度传感器

温度传感器内部连接到 TS 输入通道，用于将传感器输出电压转换为数字值。

14.9 数据采集

数据采集功能用于自动收集和合并 QADC 结果和 FIFO 的数据。为了使用这个特性，需要：

配置 ADC_DGATR 中 DGAT_WRTH[3:0]指定写入数据缓冲器[95:0]的结果数据数量。每个结果数据的位宽由 ADC_CFGR1 中 RES[1:0]指定。

配置 ADC_DGATR 中 DGAT_RDTH[3:0]指定读数据缓冲器[95:0]的采集数据数量。每个采集数据的位宽由 ADC_DGATR 的 DGAT_DSIZ[1:0]确定。

设置 ADC_CFGR1 中的 DMAEN 来在数据缓冲器[95:0]和 DMA 间建立连接。或者保持 DMAEN 清除，在 ADC_ISR2 中 DGAT_DBRDY 置位时，读 ADC_DBUFR 来访问数据缓冲器。

设置 ADC_DGATR 中的 DGAT_EN 位来使能这个功能和开始转换。

14.10 ADC 中断

中断可以产生于以下事件：

ADC 上电，ADC 准备好时（ADRDY 标志位）

任意转换结束（EOC 标志位）

序列转换结束（EOSEQ 标志位）

模拟看门狗检测发生（AWD 标志位）

采样阶段结束（EOSMP 标志位）

发生数据溢出（OVR 标志位）单独的使能中断位提供了更多的灵活性

结果 FIFO 超时发生（FIFO_TOF 标志位）

数据缓冲器超时发生（DBUF_TOF 标志位）

数据缓冲器或者数据采集功能准备完成（DGAT_DBRDY 标志位）

14.11 内存映射和寄存器

本节描述内存映射和寄存器结构。

14.11.1 内存映射

QADC 内存映像如表格 13-4 所示，**注意：**S = 超级用户访问。普通用户对只限于超级用户使用的地址位置访问无效，且会产生一个周期终止传输错误。

表格 14-4：QADC 内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	ADC 中断和状态寄存器（ADC_ISR）		S
0x0004	ADC 中断使能寄存器（ADC_IER）		S
0x0008	ADC 控制寄存器（ADC_CR）		S
0x000C	ADC 配置寄存器 1（ADC_CFGR1）		S
0x0010	ADC 配置寄存器 2（ADC_CFGR2）		S
0x0014	ADC 采样时间寄存器（ADC_SMPR）		S
0x0018	ADC 看门狗寄存器（ADC_WDG）		S
0x001C	ADC 看门狗阈值寄存器（ADC_TR）		S
0x0020~28	保留		S
0x002C	ADC 通道选择寄存器 1（ADC_CHSELR1）		S
0x0030	ADC 通道选择寄存器 2（ADC_CHSELR2）		S
0x0034~48	保留		S
0x004C	ADC 访问 FIFO 寄存器（ADC_FIFO）		S
0x0050	保留		S
0x0054	ADC 中断和状态寄存器 2（ADC_ISR2）		S
0x0058	ADC 数据采集寄存器（ADC_DGATR）		S
0x005C	ADC 数据缓冲寄存器（ADC_DBUFR）		S
0x0060	ADC FIFO 超时寄存器（ADC_FIFOTOR）		S
0x0064~7C	保留		S
0x0080	ADC 测试数据 3（ADC_DFT3）		S
0x0084	ADC 测试数据 2（ADC_DFT2）		S
0x0088	ADC 测试数据 1（ADC_DFT1）		S
0x008C	ADC 测试数据 0（ADC_DFT0）		S
0x0090	ADC 测试数据 7（ADC_DFT7）		S
0x0094	ADC 测试数据 6（ADC_DFT6）		S
0x0098	ADC 测试数据 5（ADC_DFT5）		S
0x009C	ADC 测试数据 4（ADC_DFT4）		S
0x00A0	ADC 测试数据 8（ADC_DFT8）		S
0x00A4	ADC 通道选择寄存器 3（ADC_CHSELR3）		S

14.11.2 寄存器

14.11.2.1 ADC 中断和状态寄存器（ADC_ISR）

偏移地址：0x0000 复位值：0x00000040

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AWD	EMPTY	FULL	OVR	EOSEQ	EOC	EOSMP	ADRDY
r/w1c	ro	ro	r/w1c	r/w1c	r/w1c	r/w1c	r/w1c

图表 14-12：ADC 中断和状态寄存器（ADC_ISR）

读：任何时间

写：任何时间

比特位	名称	复位值	读写属性	功能说明
[31:8]	保留	0x0	RO	---
[7]	AWD	0x0	R/W1C	模拟看门狗标志位： 当转换后电压超过了 ADC_TR 中设定的值时，此位由硬件置位。通过软件向此位写 1 清除。 0 = 没有模拟看门狗事件发生（或者标志事件已被软件确认并清除） 1 = 模拟看门狗事件发生
[6]	EMPTY	0x1	RO	FIFO 空状态位： 当 FIFO 空时此位由硬件置位。当 FIFO 不空时由硬件清除。 0 = FIFO 不空； 1 = FIFO 空
[5]	FULL	0x0	RO	FIFO 满状态位： 当 FIFO 满时此位由硬件置位。当 FIFO 不满时由硬件清除。 0 = FIFO 不满； 1 = FIFO 满
[4]	OVR	0x0	R/W1C	ADC 溢出： 当发生溢出，即 FULL 标志位已经置位且完成了一个新的转换时，此位由硬件置位。通过软件向此位写 1 清除。

比特位	名称	复位值	读写属性	功能说明
				0 = 没有溢出发生（或者标志事件已被软件确认并清除） 1 = 溢出发生
[3]	EOSEQ	0x0	R/W1C	序列结束标志位： 在转换序列结束时由硬件置位，通过软件向此位写 1 清除。 0 = 转换序列未完成（或者标志事件已被软件确认并清除） 1 = 转换序列完成
[2]	EOC	0x0	R/W1C	转换结束标志位： 每一个通道完成转换，即有新的数据结果写入 ADC_FIFO 里时，此位由硬件置位。通过软件向此位写 1 或者读 ADC_FIFO 清除此位。 0 = 通道转换未完成（或者标志事件已被软件确认并清除） 1 = 通道转换完成
[1]	EOSMP	0x0	R/W1C	采样结束标志位： 在转换过程中采样阶段结束时由硬件置位。 0 = 采样阶段未结束（或者标志事件已被软件确认并清除） 1 = 采样阶段已结束
[0]	ADRDY	0x0	R/W1C	ADC 准备就绪： 在 ADC 已使能（ADEN = 1）且 ADC 已准备就绪接受转换请求的状态时由硬件置位。通过软件向此位写 1 清除。 0 = ADC 还没准备好开始转换（或者标志事件已被软件确认并清除） 1 = ADC 已准备好开始转换

14.11.2.2 ADC 中断使能寄存器（ADC_IER）

偏移地址：0x0004　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
AWDIE	保留		OVRIE	EOSEQIE	EOCIE	EOSMPIE	ADRDYIE
rw	ro		rw	rw	rw	rw	rw

图表 14-13：ADC 中断使能寄存器（ADC_IER）

读：任何时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31:8]	保留	0x0	RO	---
[7]	AWDIE	0x0	RW	模拟看门狗中断使能： 通过软件可以设置或清除此位来使能/禁用模拟看门狗中断。 0 = 模拟看门狗中断禁用 1 = 模拟看门狗中断使能
[6:5]	保留	0x0	RO	---
[4]	OVRIE	0x0	RW	溢出中断使能： 通过软件可以设置或清除此位来使能/禁用溢出中断。 0 = 溢出中断禁用 1 = 溢出中断使能。当 OVR 置位时产生中断
[3]	EOSEQIE	0x0	RW	序列结束中断使能： 通过软件可以设置或清除此位来使能/禁用序列结束中断。 0 = EOSEQ 中断禁用 1 = EOSEQ 中断使能。当 EOSEQ 置位时产生中断
[2]	EOCIE	0x0	RW	转换结束中断使能： 通过软件可以设置或清除此位来使能/禁用转换结束中断。

比特位	名称	复位值	读写属性	功能说明
				0 = EOC 中断禁用 1 = EOC 中断使能。当 EOC 置位时产生中断
[1]	EOSMPIE	0x0	RW	采样结束标志位： 通过软件可以设置或清除此位来使能/禁用采样阶段结束中断。 0 = EOSMP 中断禁用 1 = EOSMP 中断使能。当 EOSMP 置位时产生中断
[0]	ADRDYIE	0x0	RW	ADC 准备就绪： 通过软件可以设置或清除此位来使能/禁用 ADC 就绪中断。 0 = ADRDY 中断禁用 1 = ADRDY 中断使能。当 ADRDY 置位时产生中断

14.11.2.3 ADC 控制寄存器（ADC_CR）

偏移地址：0x0008　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				ADSTP	ADSTART	ADDIS	ADEN
ro				r/w1	r/w1	r/w1	r/w1

图表 14-14：ADC 控制寄存器（ADC_CR）

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0x0	RO	---
[3]	ADSTP	0x0	R/W1	ADC 停止转换指令： 此位由软件设置以停止并放弃正在进行的转换（ADSTP 指令）。在转换已完成放弃操作且 ADC 准备接受新的开始转换指令时由硬件清除。

比特位	名称	复位值	读写属性	功能说明
				0 = 没有 ADC 停止指令进行 1 = 写 1 停止 ADC。读此位为 1 时意味着 ADC 停止指令在进行中 **注意：** 软件只能在 ADSTART = 1 和 ADIS = 0（ADC 使能且可能在转换中且没有未处理的 ADC 禁用请求）时设置 ADSTP
[2]	ADSTART	0x0	R/W1	ADC 开始转换指令： 此位由软件设置以开始 ADC 转换。受 TRIGMODE 位配置控制，转换可以直接开始（软件触发配置），或者一次硬件触发开始（硬件触发配置）。 此位由硬件清除： 单次转换模式中选择软件触发时：在声明序列转换结束（EOSEQ）标志位时 不连续模式中选择软件触发时：在声明转换结束（EOC）标志位时 在所有情况下：在执行完 ADSTP 指令后。同时硬件会清除 ADSTP 位。 0 = 没有 ADC 转换在进行 1 = 写 1 启动 ADC。读此位为 1 意味着 ADC 在运行中且可能在进行转换。 **注意：** 软件只能在 ADEN = 1 和 ADIS = 0 (ADC 使能且没有未处理的 ADC 禁用请求）时设置 ADSTART
[1]	ADDIS	0x0	R/W1	ADC 禁用指令： 此位由软件设置以禁用 ADC（ADDIS 指令）同时进入掉电状态（OFF 状态）。当 ADC 禁用生效时由硬件清除（同时清除 ADEN 位）。 0 = 没有 ADDIS 指令进行 1 = 写 1 禁用 ADC。读 1 时指示 ADDIS 指令进行中。 **注意：** 软件只能在 ADEN = 1 和 ADSTART = 0（此时没有转换进行）时设置 ADDIS
[0]	ADEN	0x0	R/W1	ADC 使能指令： 此位由软件设置以使能 ADC。当 ADRDY 标志位置位时，ADC 可开始工作。当执行 ADDIS 指令后 ADC 禁用时此位由硬件清除。 0 = ADC 禁用（OFF 状态）；1 = 写 1 使能 ADC **注意：**软件只能在 ADC_CR 中所有位为 0(ADSTP

比特位	名称	复位值	读写属性	功能说明
				= 0，ADSTART = 0，ADDIS = 0 和 ADEN = 0）时置位 ADEN。

14.11.2.4 ADC 配置寄存器 1（ADC_CFGR1）

偏移地址：0x000C　　复位值：0x07008000

31	30	29	28	27	26	25	24
DIFF	OVRMOD	QCLK_DIS	SMP_OBE	保留	SEQ_LEN[2:0]		
rw	rw	Rw	rw	ro	rw		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DISCEN	AUTOFF	WAIT	CONT	保留			
rw	rw	Rw	rw	ro			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEL_EXT_VREF	保留	TRIGMODE[2:0]			ALIGN	RES[1:0]	
rw	ro	rw			rw	rw	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留	DMATH[2:0]			保留			DMAEN
ro	rw			ro			rw

图表 14-15：ADC 配置寄存器（ADC_CFGR1）

读：任何时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31]	DIFF	0x0	RW	选择差分输入： 此位决定输入是单端输入还是差分输入。 0 = 模拟输入为单端采样 1 = 模拟输入为差分采样
[30]	OVRMOD	0x0	RW	溢出管理模式： 此位由软件设置和清除，配置数据溢出的管理方式。 0 = 当检测到溢出时，ADC_CR 保留旧数据 1 = 当检测到溢出时，ADC_CR 覆盖新的转换结果
[29]	QCLK_DIS	0x0	RW	QADC 时钟禁用： 指定是否禁用 QADC 时钟 0 = 不禁用；1 = 禁用
[28]	SMP_OBE	0x0	RW	采样信号输出缓冲器使能： 此位决定采样信号输出缓冲器是否使能。**注意：**

比特位	名称	复位值	读写属性	功能说明
				此位功能在本项目中未实现，因此此位为保留位且写数据无效。 0 = 禁用；1 = 使能
[27]	保留	0x0	RO	---
[26:24]	SEQ_LEN[2:0]	0x7	RW	序列长度： 此位决定了序列的长度。其中序列长度 = SEQ_LEN+1。即当 SEQ_LEN = 7 时，序列长为 8；SEQ_LEN = 0 时，序列长为 1。
[23]	DISCEN	0x0	RW	不连续模式： 此位通过软件设置或清除以使能/禁用 ADC 的不连续模式。 0 = 不连续模式禁用；1 = 不连续模式使能
[22]	AUTOFF	0x0	RW	自动关闭模式： 此位由软件设置或清除以使能/禁用自动关闭模式。 0 = 自动关闭模式禁用 1 = 自动关闭模式使能
[21]	WAIT	0x0	RW	等待转换模式： 此位由软件设置或清除以使能/禁用等待转换模式。 0 = 等待转换模式关闭 1 = 等待转换模式开启
[20]	CONT	0x0	RW	单次/连续转换模式： 此位由软件设置或清除，如果置位，则转换将连续进行直到清除此位。 0 = 单次转换模式；1 = 连续转换模式
[19:16]	保留	0x0	RO	---
[15]	SEL_EXT_VREF	0x1	RW	选择外部 VREF： 此位由软件设置或清除以选择 ADC 的模拟参考电压。 0 = 选择内部参考电压 1 = 选择外部参考电压
[14]	保留	0x0	RO	---
[13:11]	TRIGMODE[2:0]	0x0	RW	触发模式选择： 这些位用于选择软件触发模式或是外部触发极性。 详细参考**表格 14-2:**。
[10]	ALIGN	0x0	RW	数据对齐： 此位由软件设置或清除以选择数据右对齐或左

比特位	名称	复位值	读写属性	功能说明
				对齐。详细参考**图表 14-10:**。 0 = 右对齐; 1 = 左对齐
[9:8]	RES[1:0]	0x0	RW	数据分辨率: 此位由软件设置或清除以选择转换的分辨率。 详细请参考**图表 14-10:**。
[7]	保留	0x0	RW	---
[6:4]	DMATH[2:0]	0x0	RW	直接内存访问阈值: 指定 FIFO 的数据数量阈值,一旦超过这个阈值,则 DMAEN 置位,QADC 发送传输请求给 DMA。
[3:1]	保留	0x0	RO	---
[0]	DMAEN	0x0	RW	直接内存访问使能: 此位由软件设置或清除以使能产生 DMA 请求。这将允许使用 DMA 控制器自动管理转换的数据。 0 = DMA 禁用; 1 = DMA 使能

14.11.2.5 ADC 配置寄存器 2（ADC_CFGR2）

偏移地址：0x0010　　　　　　　　　　复位值：0x00000220

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
ADC_BYPASS	保留			QPR[3:0]			
rw	ro			rw			

7	6	5	4	3	2	1	0
STCNT[7:0]							
rw							

图表 14-16：ADC 配置寄存器 2（ADC_CFRG2）

读：任意时间

写：ADC 使能前

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15]	ADC_BYPASS	0x0	RW	ADC 模拟输入支路： 0 = 正常 ADC 输入 1 = ADC 模拟输入支路，数据从 ADC_DFT* 输入作为测试
[14:12]	保留	0x0	RO	---
[11:8]	QPR[3:0]	0x2	RW	预分频器时钟分频位： 这些位选择预分频器时钟分频值以产生 QADC 时钟： $F_{QCLK} = F_{SYS\_QCLK}/(QPR[3:0]+1)$ 其中： 0< = QPR[3:0]< = 15
[7:0]	STCNT[7:0]	0x20	RW	ADC 启动计数器位： ADC 需要 tSTAB（~2μs）的稳定时间才能开始准确转换。这个时间是通过计数 QCLK 周期，直到计数器达到 STCNT[7:0]值。所以用户应该在 ADC 使能前设置这些位。如果 QCLK = 16MHz，则 STCNT[7:0] = 2000(/1000/16) = 32

14.11.2.6 ADC 采样时间寄存器（ADC_SMPR）

偏移地址：0x0014　　复位值：0x00000002

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

图表 14-17：ADC 采样时间寄存器（ADC_SMPR）

读：任何时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31:8]	保留	0x0	RO	---
[7:0]	SMP[7:0]	0x02	RW	采样时间选择： 这些位由软件写入以选择应用于所有通道的采样时间。采样时间为（SMP[7:0]+2）QCLKs 示例：SMP[7:0] = 0x2 即意味着采样时间为 4QCLKs。

14.11.2.7 ADC 看门狗寄存器（ADC_WDG）

偏移地址：0x0018 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AWDEN	AWDSGL	保留	AWDCH[4:0]				
rw	rw	ro	rw				

图表 14-18：ADC 看门狗寄存器（ADC_WDG）

读：任何时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31:8]	保留	0x0	RO	---
[7]	AWDEN	0x0	RW	模拟看门狗使能位： 此位由软件设置或清除。 0 = 模拟看门狗禁用；1 = 模拟看门狗使能
[6]	AWDSGL	0x0	RW	单信道或者全信道使能看门狗： 此位由软件设置或清除，可选择看门狗在AWDCH[4:0]中定义的通道单独使能或者在全通道使能。 0 = 模拟看门狗全通道使能 1 = 模拟看门狗在特定通道使能
[5]	保留	0x0	RO	---
[4:0]	AWDCH[4:0]	0x0	RW	模拟看门狗通道选择： 这些位由软件设置或清除，选择由模拟看门狗保护的输入通道： 00000：AWD 监视 ADC 模拟输入通道 0 00001：AWD 监视 ADC 模拟输入通道 1 …… 01111：AWD 监视 ADC 模拟输入通道 15 10000：AWD 监视电池检测 其他值：保留且未被使用

14.11.2.8 ADC 看门狗阈值寄存器（ADC_TR）

偏移地址：0x001C 复位值：0x1FFF0000

31	30	29	28	27	26	25	24
保留			NOTE	HT[11:8]			
ro			rw	rw			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HT[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留			NOTE	LT[11:8]			
ro			rw	rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LT[7:0]							
rw							

图表 14-19：ADC 看门狗阈值寄存器（ADC_TR）

读：任何时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31:29]	保留	0x0	RO	---
[28]	NOTE	0x1	RW	配置 HT[11:0]时清除此位
[27:16]	HT[11:0]	0xFFF	RW	模拟看门狗上阈值： 这些位由软件设置或清除，定义模拟看门狗的上阈值。
[15:13]	保留	0x0	RO	---
[12]	NOTE	0x0	RW	配置 LT[11:0]时清除此位
[11:0]	LT[11:0]	0x0	RW	模拟看门狗下阈值： 这些位由软件设置或清除，定义模拟看门狗的下阈值。

14.11.2.9 ADC 通道选择寄存器 i（ADC_CHSELR1，ADC_CHSELR2）

偏移地址：0x002C 复位值：0x00000000

31	30	29	28	27	26	25	24
保留			CCW3[4:0]				
ro			rw				
23	22	21	20	19	18	17	16
保留			CCW2[4:0]				
ro			rw				
15	14	13	12	11	10	9	8
保留			CCW1[4:0]				
ro			rw				
7	6	5	4	3	2	1	0
保留			CCW0[4:0]				
ro			rw				

图表 14-20：ADC 通道选择寄存器 1（ADC_CHSELR1）

读：任意时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31:29]	保留	0x0	RO	---
[28:24]	CCW3[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**
[23:21]	保留	0x0	RO	---
[20:16]	CCW2[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**
[15:13]	保留	0x0	RO	---
[12:8]	CCW1[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**
[7:5]	保留	0x0	RO	---
[4:0]	CCW0[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**

偏移地址：0x0030　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留			CCW7[4:0]				
ro			rw				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留			CCW6[4:0]				
ro			rw				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留			CCW5[4:0]				
ro			rw				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留			CCW4[4:0]				
ro			rw				

图表 14-21：ADC 通道选择寄存器 2（ADC_CHSELR2）

读：任意时间

写：ADC 开始前

比特位	名称	复位值	读写属性	功能说明
[31:29]	保留	0x0	RO	---
[28:24]	CCW7[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**
[23:21]	保留	0x0	RO	---
[20:16]	CCW6[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**
[15:13]	保留	0x0	RO	---
[12:8]	CCW5[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**
[7:5]	保留	0x0	RO	---
[4:0]	CCW4[4:0]	0x0	RW	**选择转换信道，信道码见表 28-1。**

14.11.2.10 ADC FIFO 访问寄存器（ADC_FIFO）

偏移地址：0x004C 复位值：0x00000000

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

图表 14-22：ADC FIFO 访问寄存器（ADC_FIFO）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	DATA[15:0]	0x0	RO	转换的数据: 详细参考图表 14-10：

14.11.2.11 ADC 中断和状态寄存器 2（ADC_ISR2）

偏移地址：0x0054 复位值：0x00000000

31	30	29	28	27	26	25	24
FIFO_TOF	保留			FIFO_CNT[3:0]			
r/w1c	ro			ro			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留				AWD_ERRCH[3:0]			
ro				ro			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DBUF_TOF	保留		DGAT_DBRDY	DGAT_RDCNT[3:0]			
r/w1c	ro		ro	ro			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DGAT_WRCNT[3:0]				保留			
ro				ro			

图表 14-23：ADC 中断和状态寄存器 2（ADC_ISR2）

比特位	名称	复位值	读写属性	功能说明
[31]	FIFO_TOF	0x0	R/W1C	FIFO 超时标志位： 当结果 FIFO 超时事件发生时，此位由硬件置位。通过软件向此位写 1 可清除 0 = FIFO 超时事件未发生 1 = FIFO 超时事件发生
[30:28]	保留	0x0	RO	---
[27:24]	FIFO_CNT[3:0]	0x0	RO	FIFO 计数器： 计数 FIFO 中数据数。
[23:20]	保留	0x0	RO	---
[19:16]	AWD_ERRCH[3:0]	0x0	RO	模拟看门狗错误通道 指示最后一个发生看门狗的错误通道。
[15]	DBUF_TOF	0x0	R/W1C	数据缓冲器超时标志位： 当数据缓冲超时时此位硬件置位。可由软件写 1 清除。 0 = 数据缓冲超时未发生 1 = 数据缓冲超时发生
[14:13]	保留	0x0	RO	---
[12]	DGAT_DBRDY	0x0	RO	数据采集缓冲器就绪： 当数据采集功能的数据缓冲器就绪时，此位由硬件置位。可由软件写 1 清除。

比特位	名称	复位值	读写属性	功能说明
				0 = 数据缓冲器未就绪 1 = 数据缓冲器就绪
[11:8]	DGAT_RDCNT[3:0]	0x0	RO	数据采集读计数： 指示数据采集功能中读操作计数。当计数值达到 ADC_DGATR 中 DGAT_RDTH[3:0]值，发生对 ADC_DBUFR 新的读访问时，DGAT_DBRDY 位被清除。
[7:4]	DGAT_WRCNT[3:0]	0x0	RO	数据采集写计数： 指示数据采集功能中写操作计数。当计数值达到 ADC_DGATR 中 DGAT_WRTH[3:0]值，发生对 ADC_DBUFR 新的写访问时，DGAT_DBRDY 位被清除。
[3:0]	保留	0x0	RO	---

14.11.2.12 ADC 数据采集寄存器（ADC_DGATR）

偏移地址：0x0058　　复位值：0x00000000

31	30	29	28	27	26	25	24
DBUF_TOVAL[15:8]							
rw							

23	22	21	20	19	18	17	16
DBUF_TOVAL[7:0]							
rw							

15	14	13	12	11	10	9	8
DBUF_TOIE	DBUF_TOEN	保留	DGAT_DBRDYIE	DGAT_RDTH[3:0]			
rw	rw	ro	rw	rw			

7	6	5	4	3	2	1	0
DGAT_WRTH[3:0]				DGAT_DSIZ[1:0]		保留	DGAT_EN
rw				rw		ro	rw

图表 14-24：ADC 数据采集寄存器（ADC_DGATR）

比特位	名称	复位值	读写属性	功能说明
[31:16]	DBUF_TOVAL[15:0]	0x0	RW	数据缓冲器超时值： 指定数据缓冲超时触发的值。当数据缓冲器不为空且 DBUF_TOEN 置位时，数据缓冲器超时计数器打开。如果直到达到触发值对数据缓冲器都

比特位	名称	复位值	读写属性	功能说明
				没有操作，ADC_ISR2 中的 DBUF_TOF 被设置。
[15]	DBUF_TOIE	0x0	RW	数据缓冲器超时中断使能： 此位由软件设置或清除以使能/禁用数据缓冲器超时中断。 0 = 中断禁用；1 = 中断使能
[14]	DBUF_TOEN	0x0	RW	数据缓冲器超时功能使能： 此位由软件设置或清除以使能/禁用数据缓冲器超时功能。 0 = 功能禁用；1 = 功能使能
[13]	保留	0x0	RO	---
[12]	DGAT_DBRDYIE	0x0	RW	数据采集缓冲器就绪中断使能： 此位由软件设置或清除以使能/禁用数据采集缓冲器就绪中断。 0 = 中断禁用；1 = 中断使能
[11:8]	DGAT_RDTH[3:0]	0x0	RW	数据采集读阈值： 指定数据采集功能读操作阈值。当 ADC_ISR2 中 DGAT_RDCNT[3:0]值达到阈值，发生对 ADC_DBUFR 新的读访问时，DGAT_DBRDY 位被清除。
[7:4]	DGAT_WRTH[3:0]	0x0	RW	数据采集写阈值： 指定数据采集功能写操作阈值。当 ADC_ISR2 中 DGAT_WRCNT[3:0]值达到阈值，发生对 ADC_DBUFR 新的写访问时，DGAT_DBRDY 位被清除。
[3:2]	DGAT_DSIZ[1:0]	0x0	RW	数据采集读操作数据长度： 指定数据采集功能中通过软件操作对数据缓冲器进行读访问的数据长度： 00 = 字 01 = 半字 10 = 字节 11 = 保留
[1]	保留	0x0	RO	---
[0]	DGAT_EN	0x0	RW	数据采集功能使能位： 此位由软件设置或清除以使能/禁用数据采集功能。 0 = 功能禁用；1 = 功能使能

14.11.2.13 ADC 数据缓冲寄存器（ADC_DBUFR）

偏移地址：0x005C 复位值：0x00000000

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

图表 14-25：ADC 数据缓冲寄存器（ADC_DBUFR）

比特位	名称	复位值	读写属性	功能说明
[31:0]	DBUF_DATA[31:0]	0x0	RO	数据缓冲器数据： 当 DGAT_EN 置位时，读此寄存器将返回采集的数据。注意，软件的读访问类型应与 ADC_DGATR 中 DGAT_DSIZ[1:0]设置的相同。

14.11.2.14 ADC FIFO 超时寄存器（ADC_FIFOTOR）

偏移地址：0x0060 复位值：0x00000000

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

图表 14-26：ADC FIFO 超时寄存器（ADC_FIFOTOR）

比特位	名称	复位值	读写属性	功能说明
[31:16]	FIFO_TOVAL[15:0]	0x0	RW	结果 FIFO 超时值： 指定结果 FIFO 的超时值。当 FIFO 不为空且 FIFO_TOEN 置位时，FIFO 超时计数器打开。直到计数器达到触发值对 FIFO 都没有操作时，ADC_ISR2 中 FIFO_TOF 位置位。
[15]	FIFO_TOIE	0x0	RW	结果 FIFO 超时中断使能： 此位由软件设置或清除以使能/禁用 FIFO 超时中断。 0 = 中断禁用；1 = 中断使能
[14]	FIFO_TOEN	0x0	RW	结果 FIFO 超时功能使能： 此位由软件设置或清除以使能/禁用 FIFO 超时功能。 0 = 功能禁用；1 = 功能使能
[13:0]	保留	0x0	RO	---

14.11.2.15 ADC 测试数据寄存器 3（ADC_DFT3）

偏移地址：0x0080 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL7[11:8]			
ro				rw			
23	22	21	20	19	18	17	16
CHANEL7[7:0]							
rw							
15	14	13	12	11	10	9	8
保留				CHANEL6[11:8]			
ro				rw			
7	6	5	4	3	2	1	0
CHANEL6[7:0]							
rw							

图表 14-27：ADC 测试数据寄存器 3（ADC_DFT3）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL7[11:0]	0x0	RW	ADC 通道 7 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL6[11:0]	0x0	RW	ADC 通道 6 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.16 ADC 测试数据寄存器 2（ADC_DFT2）

偏移地址：0x0084 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL5[11:8]			
ro				rw			

23	22	21	20	19	18	17	16
CHANEL5[7:0]							
rw							

15	14	13	12	11	10	9	8
保留				CHANEL4[11:8]			
ro				rw			

7	6	5	4	3	2	1	0
CHANEL4[7:0]							
rw							

图表 14-28：ADC 测试数据寄存器 2（ADC_DFT2）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL5[11:0]	0x0	RW	ADC 通道 5 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL4[11:0]	0x0	RW	ADC 通道 4 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.17 ADC 测试数据寄存器 1（ADC_DFT1）

偏移地址：0x0088 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL3[11:8]			
ro				rw			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CHANEL3[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				CHANEL2[11:8]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANEL2[7:0]							
rw							

图表 14-29：ADC 测试数据寄存器 1（ADC_DFT1）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL3[11:0]	0x0	RW	ADC 通道 3 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL2[11:0]	0x0	RW	ADC 通道 2 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.18 ADC 测试数据寄存器 0（ADC_DFT0）

偏移地址：0x008C 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL1[11:8]			
ro				rw			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CHANEL1[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				CHANEL0[11:8]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANEL0[7:0]							
rw							

图表 14-30：ADC 测试数据寄存器 0（ADC_DFT0）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL1[11:0]	0x0	RW	ADC 通道 1 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL0[11:0]	0x0	RW	ADC 通道 0 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.19 ADC 测试数据寄存器 7（ADC_DFT7）

偏移地址：0x0090 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL15[11:8]			
ro				rw			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CHANEL15[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				CHANEL14[11:8]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANEL14[7:0]							
rw							

图表 14-31：ADC 测试数据寄存器 7（ADC_DFT7）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL15[11:0]	0x0	RW	ADC 通道 15 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL14[11:0]	0x0	RW	ADC 通道 14 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.20 ADC 测试数据寄存器 6（ADC_DFT6）

偏移地址：0x0094 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL13[11:8]			
ro				rw			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CHANEL13[7:0]							
rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				CHANEL12[11:8]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANEL12[7:0]							
rw							

图表 14-32：ADC 测试数据寄存器 6（ADC_DFT6）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL13[11:0]	0x0	RW	ADC 通道 13 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL12[11:0]	0x0	RW	ADC 通道 12 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.21 ADC 测试数据寄存器 5（ADC_DFT5）

偏移地址：0x0098 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL11[11:8]			
ro				rw			
23	22	21	20	19	18	17	16
CHANEL11[7:0]							
rw							
15	14	13	12	11	10	9	8
保留				CHANEL10[11:8]			
ro				rw			
7	6	5	4	3	2	1	0
CHANEL10[7:0]							
rw							

图表 14-33：ADC 测试数据寄存器 5（ADC_DFT5）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL11[11:0]	0x0	RW	ADC 通道 11 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL10[11:0]	0x0	RW	ADC 通道 10 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.22 ADC 测试数据寄存器 4（ADC_DFT4）

偏移地址：0x009c 复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CHANEL9[11:8]			
ro				rw			
23	22	21	20	19	18	17	16
CHANEL9[7:0]							
rw							
15	14	13	12	11	10	9	8
保留				CHANEL8[11:8]			
ro				rw			
7	6	5	4	3	2	1	0
CHANEL8[7:0]							
rw							

图表 14-34：ADC 测试数据寄存器 4（ADC_DFT4）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27:16]	CHANEL9[11:0]	0x0	RW	ADC 通道 9 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效
[15:12]	保留	0x0	RO	---
[11:0]	CHANEL8[11:0]	0x0	RW	ADC 通道 8 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.23 ADC 测试数据寄存器 8（ADC_DFT8）

偏移地址：0x00A0 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				CHANEL16[11:8]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANEL16[7:0]							
rw							

图表 14-35：ADC 测试数据寄存器 8（ADC_DFT8）

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:0]	CHANEL16[11:0]	0x0	RW	ADC 通道 16 输入数据以进行测试 仅当 ADC_CFGR2[ADC_BYPASS] = 1 时生效

14.11.2.24 ADC 通道选择寄存器 3（ADC_CHSELR3）

偏移地址：0x00A4 复位值：0x00000000

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

图表 14-36：ADC 通道选择寄存器 3（ADC_CHSELR3）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	CHSEL[15:0]	0x0	RW	ADC 通道选择： 这些位用于端口复用。默认下，ADC 通道[15:12]和通道[7:4]的端口不用作 ADC 输入端口。需要复用时，CHSEL16[15:0]中与 CCWi 中选择的通道对应的位需要被设置。例如，如果 CCW1 中为 0x1A3C，即代表通道 1，10，3，12 被用作 ADC 输入通道，所以 ADC_CHESELR3 中的位 1，10，3，12 需要被置位。**表格 14-5**：显示了具体的端口复用。 **注意**：读 ADC_CHSELR3 将返回 0，不建议一位操作，建议对 ADC_CHSELR3 进行字写操作。

表格 14-5：ADC 端口复用

bits	value	pad reuse
CHSEL[0]	0	adc_in[0]
	1	
CHSEL[1]	0	adc_in[2]
	1	
CHSEL[2]	0	---
	1	

bits	value	pad reuse
CHSEL[3]	0	---
	1	
CHSEL[4]	0	gint[26]
	1	adc_ch_in[0]
CHSEL[5]	0	gint[28]
	1	adc_ch_in[2]
CHSEL[6]	0	gint[22]
	1	adc_ch_in[4]
CHSEL[7]	0	gint[24]
	1	adc_ch_in[6]
CHSEL[8]	0	adc_in[1]
	1	
CHSEL[9]	0	vref1v(inside input)
	1	
CHSEL[10]	0	---
	1	
CHSEL[11]	0	dac_out(inside input from dac)
	1	
CHSEL[12]	0	gint[27]
	1	adc_ch_in[1]
CHSEL[13]	0	gint[29]
	1	adc_ch_in[3]
CHSEL[14]	0	gint[23]
	1	adc_ch_in[5]
CHSEL[15]	0	gint[25]
	1	adc_ch_in[7]

15 脉冲宽度调制模块（PWM）

15.1 概述

PWM 模块有 4 个 PWM 计时器。这 4 个 PWM 计时器有 2 个预分频，2 个时钟分频，4 个时钟选择，4 个 16 位比较器，2 个死区发生器。每个 PWM 可以作为一个计时器，并且独立产生中断。

每 2 个 PWM 计时器共享一个相同的预分频。时钟分频给每个计时器 5 个时钟源（1，1/2，1/4，1/8，1/16）。16 位的计数器从时钟选择器得到一个时钟。16 位的比较器比较计数器中的值与寄存器中的阈值，用来产生 PWM 的规定的周期。从时钟分配器来的时钟信号称为 PWM 时钟。死区产生器利用 PWM 时钟作为时钟源。一旦死区产生器使能，PWM 计时器输出会被限制。2 个输出端口都会被死区产生器用来作为输出信号，来控制 off_chip 的动力控制器件。比较器的值用于脉冲宽度调制。当向下计数的计数器匹配比较寄存器时，计数器控制逻辑改变输出的信号电平。

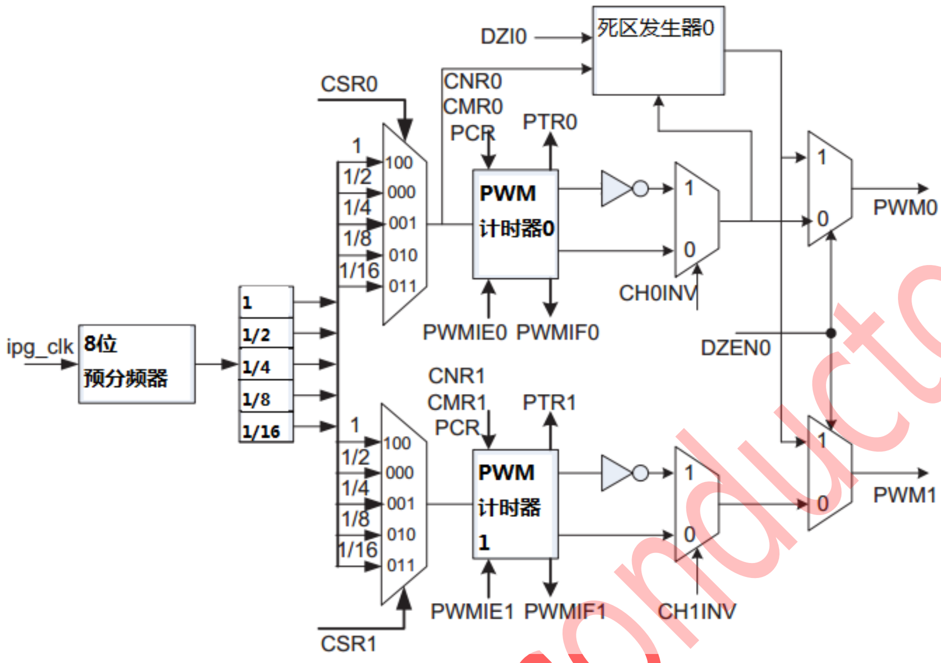
每个 PWM 计时器包含一个捕捉通道。捕捉通道 0 和 PWM0 共享一个计时器。捕捉通道 1 和 PWM1 共享另外一个计时器，以此类推。因此，用户在打开捕捉特性时，必须设置 PWM 计时器。使能捕捉特性后，，当输入通道有上升沿时，捕捉器会锁存 PWM 计数器到 CRLR；当输入通道出现下降沿时，捕捉器会 PWM 计数器到 CFLR 寄存器。捕捉信道 0 可以配置为上升沿匹配中断或者下降沿匹配中断。其他通道也可以这样。无论通道 0/1/2/3 产生中断，PWM 计数器 0/1/2/3 都会重新装载。最大的捕捉频率取决于中断处理时间。如果中断处理时间是 T0，捕捉通道输入信号不能再 T0 时间内改变，最大捕捉频率就是 1/T0。

PWM 有 4 个中断，PWM0 和捕捉通道 0 共享一个中断，PWM1 和捕捉通道 1 共享一个中断，以此类推。因此，PWM 功能和捕捉功能不能同时使用。

15.2 特性

- 可编程的周期
- 可编程占空比
- 2 个死区发生器
- 捕捉功能
- 可以配置为 GPIO

15.3 框图



图表 15-1：PWM 框图

15.4 信号描述

表格 15-1：PWM 信号描述

信号名	I/O	宽度	复位状态	描述
PWM0	I/O	1	0	PWM0 pin
PWM1	I/O	1	0	PWM1 pin
PWM2	I/O	1	0	PWM2 pin
PWM3	I/O	1	0	PWM3 pin

PWMx 可以用于 GPIO 输入/输出，也可以用于 PWM 输出和输入捕捉。在默认状态，用于 GPIO 输入。

注意： PWMx_EN 在内核配置模块中设置。

15.5 内存映射和寄存器

本节描述 PWM 模块的内存映射和寄存器结构。

15.5.1 内存映射

表格 15-2：PWM 模块内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	PWM 预分频寄存器(PPR)		S/U
0x0004	PWM 时钟选择寄存器(PCSR)		S/U
0x0008	PWM 控制寄存器(PCR)		S/U
0x000C	PWM 计数寄存器 0(PCNR0)		S/U
0x0010	PWM 比较寄存器 0(PCMR0)		S/U
0x0014	PWM 计时寄存器 0(PTR0)		S/U
0x0018	PWM 计数寄存器 1(PCNR1)		S/U
0x001C	PWM 比较寄存器 1(PCMR1)		S/U
0x0020	PWM 计时寄存器 1(PTR1)		S/U
0x0024	PWM 计数寄存器 2(PCNR2)		S/U
0x0028	PWM 比较寄存器 2(PCMR2)		S/U
0x002C	PWM 计时寄存器 2(PTR3)		S/U
0x0030	PWM 计数寄存器 3(PCNR3)		S/U
0x0034	PWM 比较寄存器 3(PCMR3)		S/U
0x0038	PWM 计时寄存器 3(PTR3)		S/U
0x003C	PWM 中断使能寄存器(PIER)		S/U
0x0040	PWM 中断标志寄存器(PIFR)		S/U
0x0044	PWM 捕捉控制寄存器 0(PCCR0)		S/U
0x0048	PWM 捕捉控制寄存器 1(PCCR1)		S/U
0x004C	PWM 捕捉上升沿锁存寄存器 0(PCRLR0)		S/U
0x0050	PWM 下降沿锁存寄存器 0(PCFLR0)		S/U
0x0054	PWM 捕捉上升沿锁存寄存器 1(PCRLR1)		S/U
0x0058	PWM 下降沿锁存寄存器 1(PCFLR1)		S/U
0x005C	PWM 捕捉上升沿锁存寄存器 2(PCRLR2)		S/U
0x0060	PWM 下降沿锁存寄存器 2(PCFLR2)		S/U
0x0064	PWM 捕捉上升沿锁存寄存器 3(PCRLR3)		S/U
0x0068	PWM 下降沿锁存寄存器 3(PCFLR3)		S/U
0x006C	PWM 端口控制寄存器(PPCR)		S/U

注意： S = 超级用户访问；U = 普通用户访问

15.5.2 寄存器描述

15.5.2.1 PWM 预分频寄存器

本寄存器用于设置预分频以及设置死区值。

偏移地址：0x0000~0x0003 复位值：0x00000000

31	30	29	28	27	26	25	24
DZI1[7:0]							
rw							
23	22	21	20	19	18	17	16
DZI0[7:0]							
rw							
15	14	13	12	11	10	9	8
CP1[7:0]							
rw							
7	6	5	4	3	2	1	0
CP0[7:0]							
rw							

图表 15-2：PWM 预分频寄存器（PPR）

比特位	名称	复位值	读写属性	功能说明
[31:24]	DZI1[7:0]	0x0	RW	**死区间隔时间寄存器 1（PWM2 和 PWM3）** 这个 8 位值决定了死区的长度。每个单位时间长度是从时钟选择器 1 中来。
[23:16]	DZI0[7:0]	0x0	RW	**死区间隔时间寄存器 0（PWM0 和 PWM1）** 这个 8 位值决定了死区的长度。每个单位时间长度是从时钟选择器 0 中来。
[15:8]	CP1[7:0]	0x0	RW	**时钟预分频配置寄存器 1(PWM2 和 PWM3)** PWM 输入时钟被（CP1+1）分频，然后输入计时器 2 和 3。如果 CP1 = 0，预分频器 1 输出时钟会停止。
[7:0]	CP0[7:0]	0x0	RW	**时钟预分频配置寄存器 0(PWM0 和 PWM1)** PWM 输入时钟被（CP1+1）分频，然后输入计时器 0 和 1。如果 CP1 = 0，预分频器 0 输出时钟会停止。

15.5.2.2 PWM 时钟选择寄存器

时钟分频提供给每个计时器 5 个时钟源（1，1/2，1/4，1/8，1/16）。每个计时器时钟来源于各自的时钟分频器。

偏移地址：0x0004~0x0007　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留	CSR3			保留	CSR2		
ro	rw			ro	rw		

7	6	5	4	3	2	1	0
保留	CSR1			保留	CSR0		
ro	rw			ro	rw		

图表 15-3：PWM 时钟选择寄存器（PCSR）

比特位	名称	复位值	读写属性	功能说明
[31:15]	保留	0x0	RO	---
[14:12]	CSR3[3:0]	0x0	RW	**计时器 3 时钟源选择** 选择计时器 3 的时钟输入 000 = 2 分频 001 = 4 分频 010 = 8 分频 011 = 16 分频 其他 = 1 分频
[23:16]	CSR2[3:0]	0x0	RW	**计时器 2 时钟源选择** 选择计时器 2 的时钟输入 000 = 2 分频 001 = 4 分频 010 = 8 分频 011 = 16 分频 其他 = 1 分频

比特位	名称	复位值	读写属性	功能说明
[15:8]	CSR1[3:0]	0x0	RW	**计时器 1 时钟源选择** 选择计时器 1 的时钟输入 000 = 2 分频 001 = 4 分频 010 = 8 分频 011 = 16 分频 其他：1 分频
[7:0]	CSR0[3:0]	0x0	RW	**计时器 0 时钟源选择** 选择计时器 0 的时钟输入 000 = 2 分频 001 = 4 分频 010 = 8 分频 011 = 16 分频 其他 = 1 分频

15.5.2.3 PWM 控制寄存器

时钟分频提供给每个计时器 5 个时钟源（1，1/2，1/4，1/8，1/16）。每个计时器时钟来源于各自的时钟分频器。

偏移地址：0x0008~0x000B　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留				CH3MOD	CH3INV	保留	CH3EN
ro				rw	rw	ro	rw

23	22	21	20	19	18	17	16
保留				CH2MOD	CH2INV	保留	CH2EN
ro				rw	rw	ro	rw

15	14	13	12	11	10	9	8
保留				CH1MOD	CH1INV	保留	CH1EN
ro				rw	rw	ro	rw

7	6	5	4	3	2	1	0
保留		DZEN1	DZEN0	CH0MOD	CH0INV	保留	CH0EN
ro		rw	rw	rw	rw	ro	rw

图表 15-4：PWM 时钟选择寄存器（PCSR）

比特位	名称	复位值	读写属性	功能说明
[31:28]	保留	0x0	RO	---
[27]	CH3MOD	0x0	RW	**计时器 3 自动加载/单次性模式** 0 = 自动加载模式；1 = 单性模式 **注意：**如果此位有上升沿或者下降沿，会导致 CNR3 及 CMR3 清除。
[26]	CH3INV	0x0	RW	**计时器 3 反向器打开/关闭** 0 = 反向器打开；1 = 反向器关闭
[25]	保留	0x0	RO	保留
[24]	CH3EN	0x0	RW	**计时器 3 使能位** 0 = 不使能；1 = 使能
[23:20]	保留	0x0	RO	---
[19]	CH2MOD	0x0	RW	**计时器 2 自动加载/单性模式** 0 = 自动加载模式；1 = 单性模式 **注意：**如果此位有上升沿或者下降沿，会导致 CNR2 及 CMR2 清除。
[18]	CH2INV	0x0	RW	**计时器 2 反向器打开/关闭** 0 = 反向器打开；1 = 反向器关闭
[17]	保留	0x0	RO	---
[16]	CH2EN	0x0	RW	**计时器 2 使能位** 0 = 不使能；1 = 使能
[15:12]	保留	0x0	RO	---
[11]	CH1MOD	0x0	RW	**计时器 1 自动加载/单性模式** 0 = 自动加载模式； 1 = 单性模式 **注意：**如果此位有上升沿或者下降沿，会导致 CNR1 及 CMR1 清除。
[10]	CH1INV	0x0	RW	**计时器 1 反向器打开/关闭** 0 = 反向器打开；1 = 反向器关闭
[9]	保留	0x0	RO	---
[8]	CH1EN	0x0	RW	**计时器 1 使能位** 0 = 不使能；1 = 使能
[7:6]	保留	0x0	RO	---
[5]	DZEN1	0x0	RW	**死区发生器 1 使能** 0 = 不使能；1 = 使能
[4]	DZEN0	0x0	RW	**死区发生器 0 使能** 0 = 不使能；1 = 使能
[3]	CH0MOD	0x0	RW	**计时器 0 自动加载/单性模式** 0 = 自动加载模式；1 = 单性模式 **注意：**如果此位有上升沿或者下降沿，会导致

比特位	名称	复位值	读写属性	功能说明
				CNR0 及 CMR0 清除。
[2]	CH0INV	0x0	RW	**计时器 0 反向器打开/关闭** 0 = 反向器打开；1 = 反向器关闭
[1]	保留	0x0	RO	---
[0]	CH0EN	0x0	RW	**计时器 0 使能位** 0 = 不使能；1 = 使能

15.5.2.4 PWM 计数寄存器（PCNR0/1/2/3）

本寄存器控制 PWM 的周期。

偏移地址：0x000C~0x000F/0x0018~0x001B/0x0024~0x0027/0x0030~0x0033

复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
CNR0/1/2/3[15:8]							
rw							

7	6	5	4	3	2	1	0
CNR0/1/2/3[7:0]							
rw							

图表 15-5：PWM 计数寄存器（PCNR）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	CNR0/1/2/3[7:0]	0x0	RW	**计时器 0/1/2/3 计数器加载值** 设置值的范围：65536~0（单位：PWM 时钟周期） 注意 1：一个 PWM 周期宽度 = CNR+1，如果 CNR 为 0，PWM 计数器/计时器将会停止。 注意 2：可以在任意时刻写入 CNR 值，该值会在下一个 PWM 计数周期生效。

15.5.2.5 PWM 比较寄存器（PCMR0/1/2/3）

本寄存器控制 PWM 的脉冲的宽度。当计数器值与本寄存器的值匹配时，输出会复位。

偏移地址：0x0010~0x0013/0x001C~0x001F/0x0028~0x002B/0x0034~0x0037　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	22	21	20	19	18	17	16
保留							
ro							
15	14	13	12	11	10	9	8
CMR0/1/2/3[15:8]							
rw							
7	6	5	4	3	2	1	0
CMR0/1/2/3[7:0]							
rw							

图表 15-6：PWM 比较寄存器（PCMR0/1/2/3）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	CMR0/1/2/3[7:0]	0x0	RW	**PWM 比较寄存器** 设置值的范围：65536~0（单位：1 个 PWM 时钟周期） CMR 用于决定输出波形的占空比。 假设：PWM 输出初始：高 CMR>= CNR：PWM 输出一直为高 CMR<CNR：PWM 输出高 = （CMR+1）个单位 CMR = 0：PWM 输出高 = 1 个单位 注意 1：PWM duty = CMR+1，如果 CMR = 0，则 PWM duty = 1 注意 2：可以在任意时刻写入 CMR 值，该值会在下一个 PWM 计数周期生效。

15.5.2.6 PWM 计时寄存器（PTR0/1/2/3）

本寄存器只读，表示了目前的计数器的计数值。

偏移地址：0x0014~0x0017/0x0020~0x0023/0x002C~0x002F/0x0038~0x003B 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
PTR0/1/2/3[15:8]							
ro							

7	6	5	4	3	2	1	0
PTR0/1/2/3[7:0]							
ro							

图表 15-7：PWM 计时寄存器（PTR0/1/2/3）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	PTR0/1/2/3[7:0]	0x0	RO	**PWM 计时寄存器** 本位只读，表示了目前的内部 16 位减计数器的计数值。

15.5.2.7 PWM 中断使能寄存器（PIER）

本寄存器用于使能 PWM 计时器中断。

偏移地址：0x003C~0x003F　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				PIER3	PIER2	PIER1	PIER0
ro				rw	rw	rw	rw

图表 15-8：PWM 中断使能寄存器（PIER）

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0x0	RO	---
[3]	PIER3	0x0	RW	**PWM 计时器 3 中断使能** 0 = 不使能 1 = 使能
[2]	PIER2	0x0	RW	**PWM 计时器 2 中断使能** 0 = 不使能 1 = 使能
[1]	PIER1	0x0	RW	**PWM 计时器 1 中断使能** 0 = 不使能 1 = 使能
[0]	PIER0	0x0	RW	**PWM 计时器 0 中断使能** 0 = 不使能 1 = 使能

15.5.2.8　PWM 中断标志寄存器（PIFR）

本寄存器用于使能 PWM 计时器中断。

偏移地址：0x0040~0x0043　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				PIFR3	PIFR2	PIFR1	PIFR0
ro				rw	rw	rw	rw

图表 15-9：PWM 中断标志寄存器（PIFR）

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0x0	RO	---
[3]	PIFR3	0x0	RW	**PWM 计时器 3 中断标志** 当计时器 3 的计数值为 0 时，并且 PIER3 为 1，则 PIFR3 将被设置为 1，该位写 1 清 0。 0 = 无中断标志 1 = 有中断标志
[2]	PIFR2	0x0	RW	**PWM 计时器 2 中断标志** 当计时器 2 的计数值为 0 时，并且 PIER2 为 1，则 PIFR2 将被设置为 1，该位写 1 清 0。 0 = 无中断标志；1 = 有中断标志
[1]	PIER1	0x0	RW	**PWM 计时器 1 中断标志** 当计时器 1 的计数值为 0 时，并且 PIER1 为 1，则 PIFR1 将被设置为 1，该位写 1 清 0。 0 = 无中断标志；1 = 有中断标志
[0]	PIER0	0x0	RW	**PWM 计时器 0 中断标志** 当计时器 0 的计数值为 0 时，并且 PIER0 为 1，则 PIFR0 将被设置为 1，该位写 1 清 0。 0 = 无中断标志；1 = 有中断标志

15.5.2.9 PWM 捕捉控制寄存器（PCCR0/1）

本寄存器用于控制捕捉功能。

偏移地址：0x0044~0x0047　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	22	21	20	19	18	17	16
CFLRD1	CRLRD1	保留	CAPIF1	CAPCH1EN	FL_IE1	RL_IE1	INV1
rw	rw	ro	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8
保留							
ro							
7	6	5	4	3	2	1	0
CFLRD0	CRLRD0	保留	CAPIF0	CAPCH0EN	FL_IE0	RL_IE0	INV0
ro	rw	ro	rw	rw	rw	rw	rw

偏移地址：0x0048~0x004B　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	22	21	20	19	18	17	16
CFLRD3	CRLRD3	保留	CAPIF3	CAPCH3EN	FL_IE3	RL_IE3	INV3
ro	rw	ro	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8
保留							
ro							
7	6	5	4	3	2	1	0
CFLRD2	CRLRD2	保留	CAPIF2	CAPCH2EN	FL_IE2	RL_IE2	INV2
rw	rw	ro	rw	rw	rw	rw	rw

图表 15-10：PWM 捕捉控制寄存器（PCCR0/1）

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23] / [7]	CFLRDx	0x0	RW	**捕捉下降沿锁存寄存器加载标志** 0 = 输入通道 x 无下降沿 1 = 输入通道 x 有下降沿时，该位为 1，该位写 1 清 0
[22] / [6]	CRLRDx	0x0	RW	**捕捉上升沿锁存寄存器加载标志** 0 = 输入通道 x 无上升沿 1 = 输入通道 x 有上升沿时，该位为 1，该位写 1 清 0
[20] / [4]	CAPIFx	0x0	RW	**捕捉通道 x 中断标志** 当计时器 1 的计数值为 0 时，并且 PIER1 为 1，则 PIFR1 将被设置为 1，该位写 1 清 0。 0 = 捕捉通道无中断 1 = 当输入通道有下降沿时，FL_IEx 位使能，本标志位置 1。当输入通道有上升沿，并且 RL_IEx 使能，则本标志位置 1。写 1 清 0。
[19] / [3]	CAPCHxEN	0x0	RW	**捕捉通道 x 使能位** 0 = 不使能；1 = 使能 当使能时，捕捉器锁存 PWM 计数器值并且保存到 CRLR（上升沿锁存）和 CFLR（下降沿锁存）寄存器中。当不使能时，捕捉器不会更新 CRLR 和 CFLR 值，并且清除通道 x 的中断。
[18] / [2]	FL_IEx	0x0	RW	**通道 x 下降沿中断使能位** 0 = 不使能；1 = 使能 当使能时，如果捕捉器侦测到通道 x 有下降沿，捕捉器将会置中断标志
[17] / [1]	RL_IEx	0x0	RW	**通道 x 下降沿中断使能位** 0 = 不使能；1 = 使能 当使能时，如果捕捉器侦测到通道 x 有下降沿，捕捉器将会置中断标志。
[16] / [0]	INVx	0x0	RW	**通道 x 信号翻转** 0 = 不翻转；1 = 翻转 当使能时，如果捕捉器侦测到通道 x 有下降沿，捕捉器将会置中断标志。

15.5.2.10 PWM 捕捉上升沿锁存寄存器（PCRLR0/1/2/3）

本寄存器用于当捕捉到输入上升沿时，锁存 PWM 的计数器值。

偏移地址：0x004C~0x004F/0x0054~0x0057/0x005C~0x005F/0x0064~0x0067　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRLR0/1/2/3[15:8]							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRLR0/1/2/3[7:0]							
ro							

图表 15-11：PWM 捕捉上升沿锁存寄存器（PCRLR0/1/2/3）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	CRLR0/1/2/3 [7:0]	0x0	RO	**捕捉上升沿锁存寄存器** 当通道 x 有上升沿时，锁存计数器中的值。

15.5.2.11 PWM 捕捉下降沿锁存寄存器（PCFLR0/1/2/3）

本寄存器用于当捕捉到输入下降沿时，锁存 PWM 的计数器值。

偏移地址：0x0050~0x0053/0x0058~0x005B/0x0060~0x0063/0x0068~0x006B 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
CFLR0/1/2/3[15:8]							
ro							

7	6	5	4	3	2	1	0
CFLR0/1/2/3[7:0]							
ro							

图表 15-12：PWM 下降沿锁存寄存器（PCFLR0/1/2/3）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	CFLR0/1/2/3[7:0]	0x0	RO	**捕捉下降沿锁存寄存器** 当通道 x 有下降沿时，锁存计数器中的值。

15.5.2.12 PWM 端口控制寄存器（PPCR）

本寄存器用于控制 PWMx 端口方向和状态。

偏移地址：0x006C~0x006F　　　　复位值：0x00000F00

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留				PDDR[3:0]			
ro				rw			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留				PULLUP_EN[3:0]			
ro				rw			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留				PDR[3:0]			
ro				rw			

图表 15-13：PWM 下降沿锁存寄存器（PCFLR0/1/2/3）

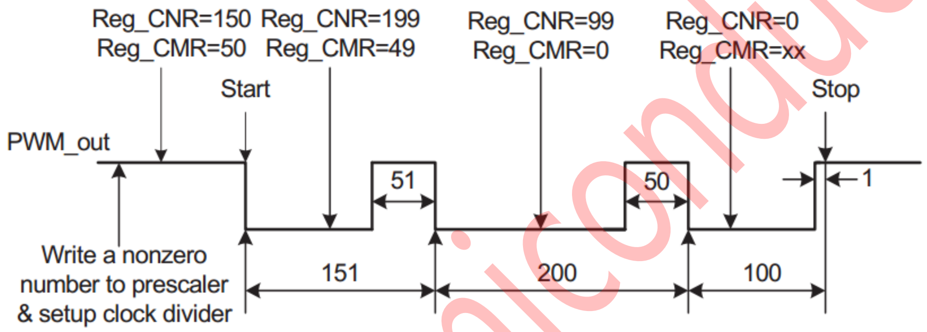
比特位	名称	复位值	读写属性	功能说明
[31:20]	保留	0x0	RO	---
[19:16]	PDDR[3:0]	0x0	RW	**端口数据方向** 控制 PWM 端的方向。复位会清除本值。 0 = 对应端口为输入 1 = 对应端口为输出
[15:12]	保留	0x0	RO	保留
[11:8]	PULLUP_EN[3:0]	0xF	RW	**端口上拉使能** 0 = 上拉不使能 1 = 上拉使能
[7:4]	保留	0x0	RO	保留
[3:0]	PDR[3:0]	0x0	RW	**端口数据寄存器** 当端口为 GPIO 状态时，数据写入本寄存器，会驱动端口。读本寄存器会返回端口状态。

15.6 功能描述

本节描述 PWM 的功能操作。

15.6.1 PWM 双缓存和自动装载

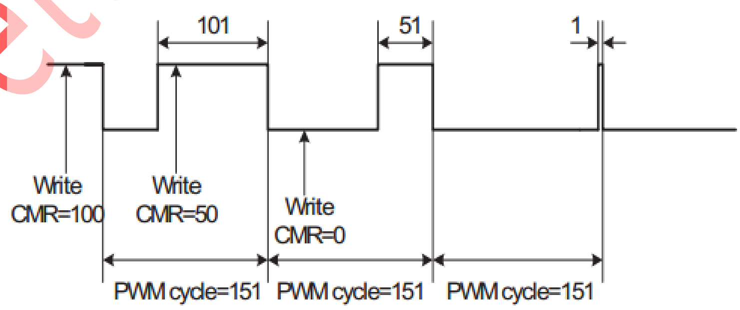
PWM 计时器有一个双缓存功能，重载功能使能后，重载值改变，在下次计数周期的时候自动加载，不需要停止目前的计时器操作。尽管新的计时器值被设置，目前的计时器操作仍然会正常操作。计数器值会写进 CNR0-3 并且目前的计数器值可以从 PTR0-3 里读出来。当减计数器计到 0 时，自动重载功能会从 CNR0-3 取值加载到减计数器。如果 CNR0-3 值设置为 0，减计数器计到 0 时，计数器会暂停。如果自动重载值设置为 0，计数器会马上停止。



图表 15-14：PWM 双缓存机制

15.6.2 调制占空比

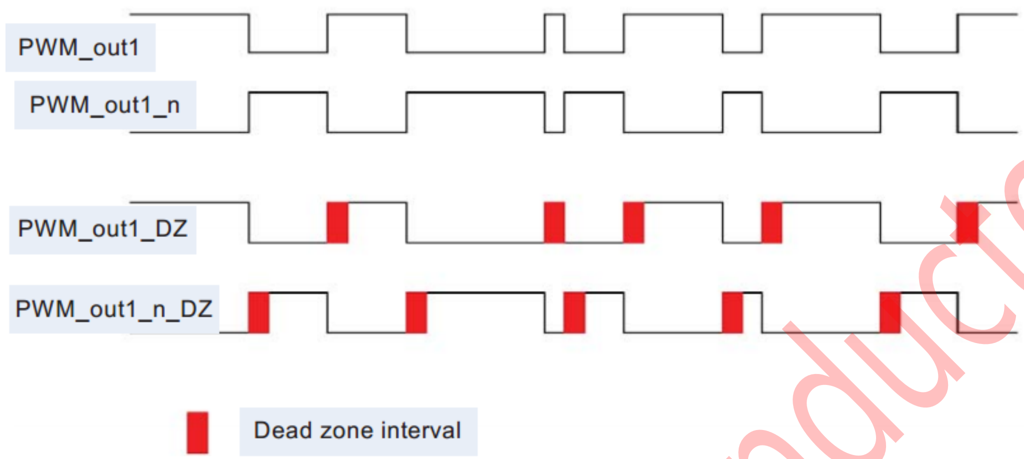
双缓存功能允许 CMR 在任何时间写入。加载值会在下一个周期生效。



图表 15-15：PWM 控制输出占空比

15.6.3 死区产生器

PWM 可以有死区产生的应用。这是为了动力设备保护而设立的。这个功能在 PWM 的上升沿前，产生了一个可编程的间隙，用户可以偏差 PPR[31:24]和 PPR[23:16]来决定 2 个死区的间隙。



图表 15-16：死区产生操作

15.6.4 PWM 计时器开始操作流程

设置时钟选择（CSR）

设置预分频及死区间隔时间（PPR）

设置反向器打开还是关闭，死区是否产生，自动重载模式还是单次模式以及 PWM 计时器关闭。

设置比较寄存器（CMR）

设置计数器寄存器（CNR）

设置中断使能寄存器（PIER）

设置 PWMx 端口为输出

使能 PWM 计时器（PCR）

15.6.5 PWM 计时器停止操作流程

方法 1：设置 16 位减计数器寄存器（CNR）为 0，并且监控 PTR。当 PTR 到达 0 时，关闭 PWM 计时器（PCR）（推荐）

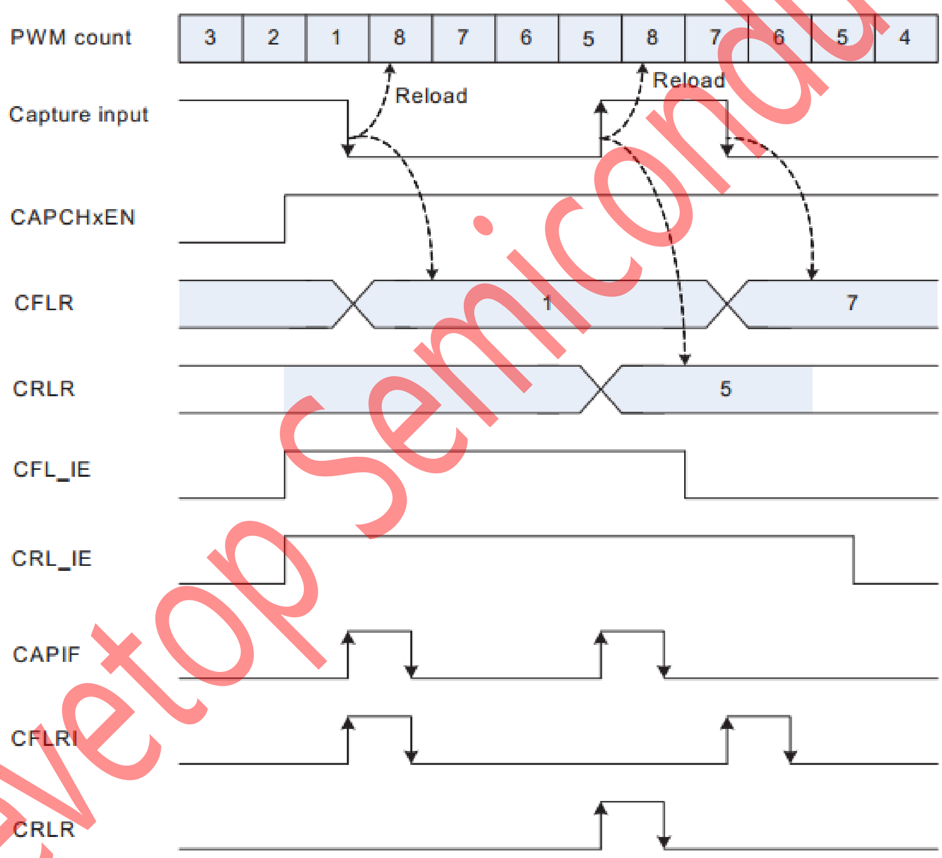
方法 2：设置 16 位减计数器寄存器（CNR）为 0，当中断到来时，关闭 PWM 计时器（PCR）（推荐）

方法 3：直接关闭 PWM 计时器（PCR）（不推荐）

15.6.6 捕捉开始流程

设置时钟选择（CSR）
设置预分频（PPR）
设置反向器打开还是关闭，死区是否产生，自动重载模式还是单次模式以及 PWM 计时器关闭。
设置计数器寄存器（CNR）
设置捕捉寄存器（CCR）
设置 PWMx 端口为输入（PPCR）
使能 PWM 计时器（PCR）

15.6.7 捕捉流程基本时序操作



图表 15-17：捕捉基本时序操作

在这个情况中，CNR 是 8：
当 CAPIFx 设置 1，PWM 计数器 CNRx 会重新装载。
通道低电平宽度为（CNR+1-CRLR）
通道高电平宽度为（CNR+1-CFLR）

16 实时时钟模块（RTC）

16.1 概述

该模块是一个能够对带有实时时钟（RTC）功能的电源管理单元（PMU）进行控制的控制器，能够将时间信息导入到 RTC 中、设置闹钟，并产生定时/闹钟中断。

16.2 特性

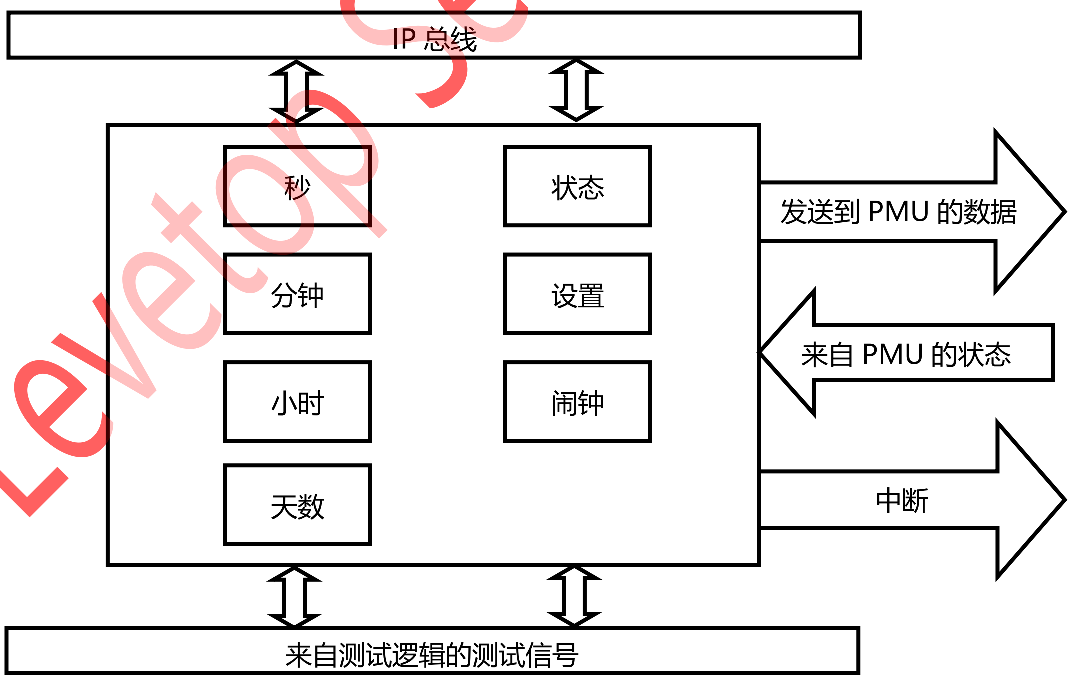
模块的主要特性：

能够向天数、小时、分钟、秒寄存器中导入时间数据，并将数据读出。
支持设置闹钟。
中断源：天数、时、分、秒中断，可编程闹钟中断，1KHz/32KHz 周期中断。

16.3 测试模式

在测试模式中，可以通过 cpu 软件配置或者芯片外部管脚对 RTC 进行配置，并且可以通过外部管脚检测 RTC 状态和时钟信号。

16.4 框图



图表 16-1：RTC 框图

16.5 外部管脚

没有片外信号。

16.6 内存映射和寄存器

本节描述 RTC 模块的内存映射和寄存器结构。

16.6.1 内存映射

引用表格 16-1 来描述 RTC 模块的内存映射。

表格 16-1：可编程中断计时器模块内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	RTC 计时器 1 寄存器（PRT1R）		S/U
0x0004	RTC 计时器 2 寄存器（PRT2R）		S/U
0x0008	RTC 闹钟 1 寄存器（PRA1R）		S/U
0x000c	RTC 闹钟 2 寄存器（PRA2R）		S/U
0x0010	RTC 时间计数器寄存器（PRTCR）		S/U
0x0014	RTC 控制和状态寄存器（PRCSR）		S/U
0x0018	RTC 使能寄存器（PRENR）		S/U
0x001c	RTC 密钥寄存器（PRKEYR）		S/U

注意： S = 超级用户访问；U = 普通用户访问

16.6.2 寄存器描述

RTC 模块由以下寄存器组成：

RTC 计时器 1 寄存器（PRT1R）能够配置和读取天数计数器中的预载入数值。
RTC 计时器 2 寄存器（PRT2R）能够配置和读取时、分、秒计数器中的预载入数值。
RTC 闹钟 1 寄存器（PRA1R）能够配置闹钟的天数计数器中的数值。
RTC 闹钟 2 寄存器（PRA2R）能够配置闹钟的时、分、秒计数器中的数值。
RTC 时间计数器寄存器（PRTCR）能够反映天数、时、分、秒计数器中的数值。
RTC 控制和状态寄存器（PRCSR）能够控制 RTC 操作并提示中断标识。
RTC 使能寄存器（PRENR）能够控制 RTC 模块使能和不使能。
RTC 密钥寄存器（PRKEYR）能够防止电压不稳定时 RTC 时钟改变。

16.6.2.1 RTC 计时器 1 寄存器

RTC 计时器 1 寄存器（PRT1R）能够配置和读取天数计数器中的预载入数值。

偏移地址：0x0000~0x0003 复位值：0x0000XXXX

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

图表 16-2：RTC 计时器 1 寄存器（PRT1R）

比特位	名称	复位值	读写属性	功能说明
[31:15]	保留	0x0	RO	---
[14:0]	Days[14:0]	---	RW	RTC 天数计数器 当向 PRT1R 寄存器写入数据时，天数数据将会被载入到天数计数器中。当读 PRT1R 寄存器时，读到的数据将反映天数计数器的数值。

16.6.2.2 RTC 计时器 2 寄存器

RTC 计时器 2 寄存器（PRT2R）能够配置和读取时、分、秒计数器中的预载入数值。

偏移地址：0x0004~0x0007 复位值：0x00XXXXXX

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留			Hours[4:0]				
ro			rw				

15	14	13	12	11	10	9	8
保留		Minutes[5:0]					
ro		rw					

7	6	5	4	3	2	1	0
保留		Seconds[5:0]					
ro		rw					

图表 16-3：RTC 计时器 2 寄存器（PRT2R）

比特位	名称	复位值	读写属性	功能说明
[31:21]	保留	0x0	RO	---
[20:16]	Hours[4:0]	---	RW	RTC 小时计数器 当向 PRT2R 寄存器 Hours[4:0]位写入数据时，小时数据将会被载入到小时计数器中。当读 PRT2R 寄存器 Hours[4:0]位时，读到的数据将反映小时计数器的数值。设置的数值应当在 0~23 之间。
[15:14]	保留	0x0	RO	---
[13:8]	Minutes[5:0]	---	RW	RTC 分钟计数器 当向 PRT2R 寄存器 Minutes[5:0]位写入数据时，分钟数据将会被载入到分钟计数器中。当读 PRT2R 寄存器 Minutes[5:0]位时，读到的数据将反映分钟计数器的数值。设置的数值应当在 0~59 之间。
[7:6]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[5:0]	Seconds[5:0]	---	RW	RTC 秒计数器 当向 PRT2R 寄存器 Seconds[5:0]位写入数据时，秒数据将会被载入到秒计数器中。当读 PRT2R 寄存器 Seconds[5:0]位时，读到的数据将反映秒计数器的数值。设置的数值应当在 0~59 之间。

16.6.2.3 RTC 闹钟 1 寄存器

RTC 闹钟 1 寄存器（PRA1R）能够配置闹钟的天数计数器中的数值。

偏移地址：0x0008~0x000b 复位值：0x00000000

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

图表 16-4：RTC 闹钟 1 寄存器（PRA1R）

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15]	AlaD_en	0x0	RW	天数闹钟使能位 1 = 使能天数闹钟功能 0 = 关闭天数闹钟功能
[14:0]	Alarm_days[14:0]	0x0	RW	RTC 天数闹钟时间设置 Alarm_days 数值配置天数闹钟对比时间。

16.6.2.4 RTC 闹钟 2 寄存器

RTC 闹钟 2 寄存器（PRA2R）能够配置闹钟的时、分、秒计数器中的数值。

偏移地址：0x000c~0x000f 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
AlaH_en	保留		Alarm_Hours[4:0]				
rw	ro		rw				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
AlaM_en	保留	Alarm_Minutes[5:0]					
rw	ro	rw					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AlaS_en	保留	Alarm_Seconds[5:0]					
rw	ro	rw					

图表 16-5：RTC 闹钟 2 寄存器（PRA2R）

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23]	AlaH_en	0x0	RW	小时闹钟使能位 1 = 使能小时闹钟功能 0 = 关闭小时闹钟功能
[22:21]	保留	0x0	RO	---
[20:16]	Alarm_Hours[4:0]	---	RW	RTC 小时闹钟时间设置 Alarm_days 数值配置小时闹钟对比时间。
[15]	AlaM_en	0x0	RW	分钟闹钟使能位 1 = 使能分钟闹钟功能 0 = 关闭分钟闹钟功能
[14]	保留	0x0	RO	---
[13:8]	Alarm_Minutes[5:0]	---	RW	RTC 分钟闹钟时间设置 Alarm_days 数值配置分钟闹钟对比时间。
[7]	AlaS_en	0x0	RW	秒闹钟使能位 1 = 使能秒闹钟功能 0 = 关闭秒闹钟功能
[6]	保留	0x0	RO	---
[5:0]	Alarm_Seconds[5:0]	---	RW	RTC 秒闹钟时间设置 Alarm_days 数值配置秒闹钟对比时间。

16.6.2.5 RTC 时间计数器寄存器

RTC 时间计数器寄存器（PRTCR）能够反映天数、时、分、秒计数器中的数值。

偏移地址：0x0010~0x0013 复位值：0xXXXXXXXX

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

图表 16-6：RTC 时间计数器寄存器（PRTCR）

16.6.2.6 RTC 控制和状态寄存器

RTC 控制和状态寄存器（PRCSR）能够控制 RTC 操作并提示中断标识。

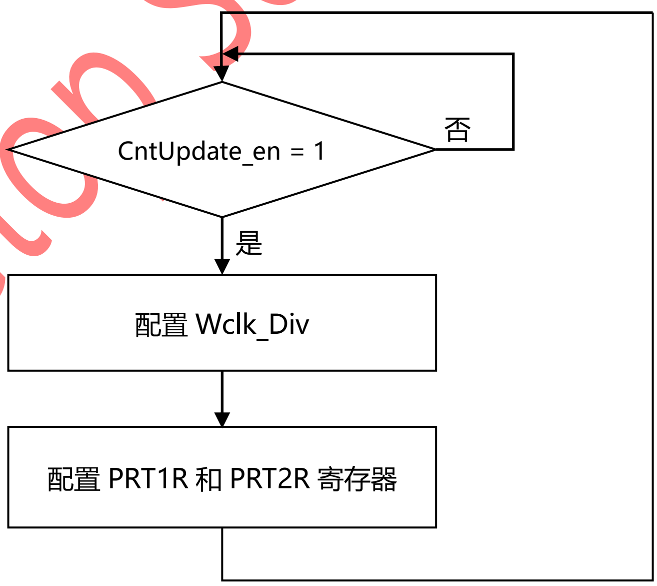
偏移地址：0x0014~0x0017 复位值：0x00000004

31	30	29	28	27	26	25	24
保留	Day_IEN	Hou_IEN	Min_IEN	Sec_IEN	Ala_IEN	1KHz_IEN	32KHz_IEN
ro	rw	rw	rw	rw	rw	rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留	Day_intf	Hou_intf	Min_intf	Sec_intf	Ala_intf	1KHz_intf	32KHz_intf
ro	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留						int_type[1:0]	
ro						rw	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Wclk_Div[4:0]					CntUpdate_en	RCNT_WEN	Dir
rw					ro	rw	rw

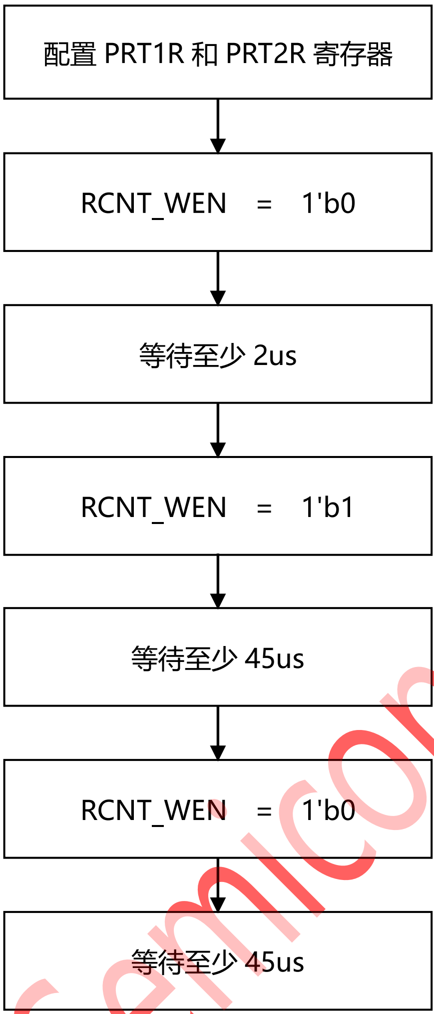
图表 16-7：RTC 控制和状态寄存器（PRCSR）

比特位	名称	复位值	读写属性	功能说明
[31]	保留	0x0	RO	---
[30]	Day_IEN	0x0	RW	RTC 天数中断使能位 0 = 关闭；1 = 使能
[29]	Hou_IEN	0x0	RW	RTC 小时中断使能位 0 = 关闭；1 = 使能
[28]	Min_IEN	0x0	RW	RTC 分钟中断使能位 0 = 关闭；1 = 使能
[27]	Sec_IEN	0x0	RW	RTC 秒中断使能位 0 = 关闭；1 = 使能
[26]	Ala_IEN	0x0	RW	RTC 闹钟中断使能位 0 = 关闭；1 = 使能
[25]	1KHz_IEN	0x0	RW	1KHz 定时中断使能位 0 = 关闭；1 = 使能
[24]	32KHz_IEN	0x0	RW	32KHz 定时中断使能位 0 = 关闭；1 = 使能
[23]	保留	0x0	RO	---
[22]	Day_intf	0x0	RW	RTC 天数中断标识
[21]	Hou_intf	0x0	RW	RTC 小时中断标识
[20]	Min_intf	0x0	RW	RTC 分钟中断标识
[19]	Sec_intf	0x0	RW	RTC 秒中断标识
[18]	Ala_intf	0x0	RW	RTC 闹钟中断标识
[17]	1KHz_intf	0x0	RW	1KHz 定时中断标识
[16]	32KHz_intf	0x0	RW	32KHz 定时中断标识
[15:10]	保留	0x0	RO	---
[9:8]	int_type[1:0]	0x0	RW	中断和标识产生类型 00：低电平产生中断 01：上升沿产生中断 10：下降沿产生中断 11：上升沿和下降沿都产生中断
[7:3]	Wclk_Div[4:0]	0x0	RW	将系统时钟分成低频（<10MHz），用于将 PRT1/2R 寄存器中的时间数据自动载入到 RTC 计数器中。 分频后的时钟频率 = 系统时钟频率 / (Wclk_Div+1) 例如：当系统时钟是 50MHz 时，wclk_div = 3’b100，此时分频后的 RTC 计数器时钟是 10MHz。
[2]	CntUpdate_en	0x1	RO	RTC 计数器数值载入就绪标识 0 = 未就绪；1 = 就绪

比特位	名称	复位值	读写属性	功能说明
[1]	RCNT_WEN	0x0	RW	使能将 PRT1/2R 寄存器中的时间数据载入到 RTC 计数器中（当 Dir 位 = 1 时） 0 = 关闭；1 = 使能
[0]	Dir	0x0	RW	直接控制使能位 0 = RCNT_WEN 不能直接控制将 PRT1/2R 寄存器中的时间数据载入到 RTC 计数器中，当写 PRT1/2R 寄存器时，时间数据会被自动载入到 RTC 计数器，只需要去检查 CntUpdate_en 位状态即可（详见图表 16-8:） 1 = RCNT_WEN 能够直接控制将 PRT1/2R 寄存器中的时间数据载入到 RTC 计数器中（详见图表 16-9:） **注意 1：** 当 Dir = 0 时，载入时间数据到 RTC 计数器中，PRT1R 和 PRT2R 两个寄存器都要进行配置。 **注意 2：** 当芯片进入停止模式时，必须确保 RTC 计数器数值载入就绪（CntUpdate_en = 1）。 **注意 3：** 当 Dir = 1 时，载入时间数据到 RTC 计数器中，必须在设置 RTC_EN 位之前，将 rtc_en_interface 位（在 RTCCFG12 寄存器中）设置为 1。



图表 16-8：RTC 计数器更新流程（Dir = 0）



图表 16-9：RTC 计数器更新流程（Dir = 1）

16.6.2.7 RTC 使能寄存器

RTC 使能寄存器（PRENR）能够控制 RTC 模块使能和不使能。

偏移地址：0x0018~0x001b　　　　　　复位值：0x00000401

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

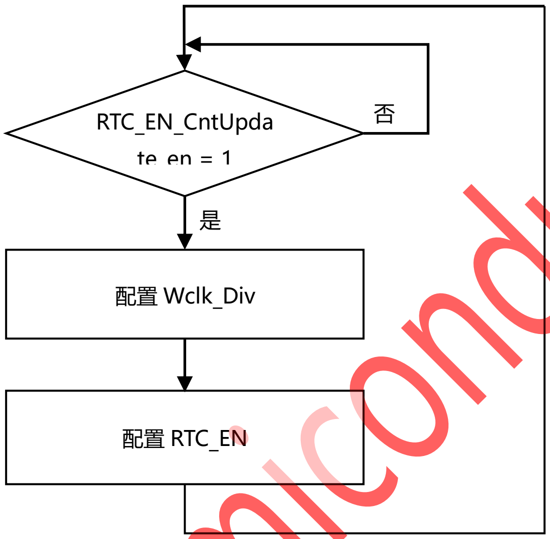
15	14	13	12	11	10	9	8
保留					RTC_EN_CntUpdate_en	RTC_EN_RCNT_WEN	RTC_EN_Dir
ro					ro	rw	rw

7	6	5	4	3	2	1	0
保留							RTC_EN
ro							rw

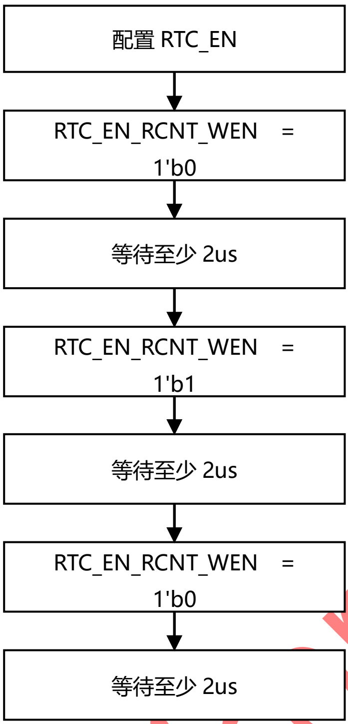
图表 16-10：RTC 使能寄存器（PRENR）

比特位	名称	复位值	读写属性	功能说明
[31:11]	保留	0x0	RO	---
[10]	RTC_EN_CntUpdate_en	0x1	RO	RTC_EN 数值载入就绪标识 0 = 未就绪；1 = 就绪
[9]	RTC_EN_RCNT_WEN	0x0	RW	使能将 RTC_EN 位的数值载入到 RTC_EN 中（当 RTC_EN_Dir 位 = 1 时） 0 = 关闭；1 = 使能
[8]	RTC_EN_Dir	0x0	RW	RTC_EN 直接控制使能位 0 = RTC_EN_RCNT_WEN 不能直接控制将 RTC_EN 位的数值载入到 RTC 计数器中，当写 RTC_EN 寄存器时，数值会被自动载入到 RTC 中，只需要去检查 RTC_EN_CntUpdate_en 位状态即可（详见图表 16-11:） 1 = RTC_EN_RCNT_WEN 能够直接控制将 RTC_EN 位的数值载入到 RTC 中（详见图表 16-12:） **注意 1：** wclk_div 和 PRCSR 中相同。 **注意 2：**当芯片进入停止模式时，必须确保 RTC_EN 数值载入就绪（RTC_EN_CntUpdate_en = 1）。 **注意 3：** 当 RTC_EN_Dir = 1 时，必须在设置

比特位	名称	复位值	读写属性	功能说明
				RTC_EN 位之前，将 rtc_en_interface 位（在 RTCCFG12 寄存器中）设置为 1。
[7:1]	保留	0x0	RO	---
[0]	RTC_EN	0x1	RW	RTC 使能位 0 = 关闭；1 = 使能



图表 16-11：RTC_EN 更新流程（RTC_EN_Dir = 0）



图表 16-12：RTC_EN 更新流程（RTC_EN_Dir = 1）

16.6.2.8 RTC 密钥寄存器

RTC 密钥寄存器（PRKEYR）能够防止电压不稳定时 RTC 时钟改变。

偏移地址：0x001c~0x001f　　　　复位值：0x00000000

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

图表 16-13：RTC 密钥寄存器（PRKEYR）

注意：rtc_key 应当被设置为 32’ h5AA55AA5

17 32 位可编程中断计时器模块（PIT32）

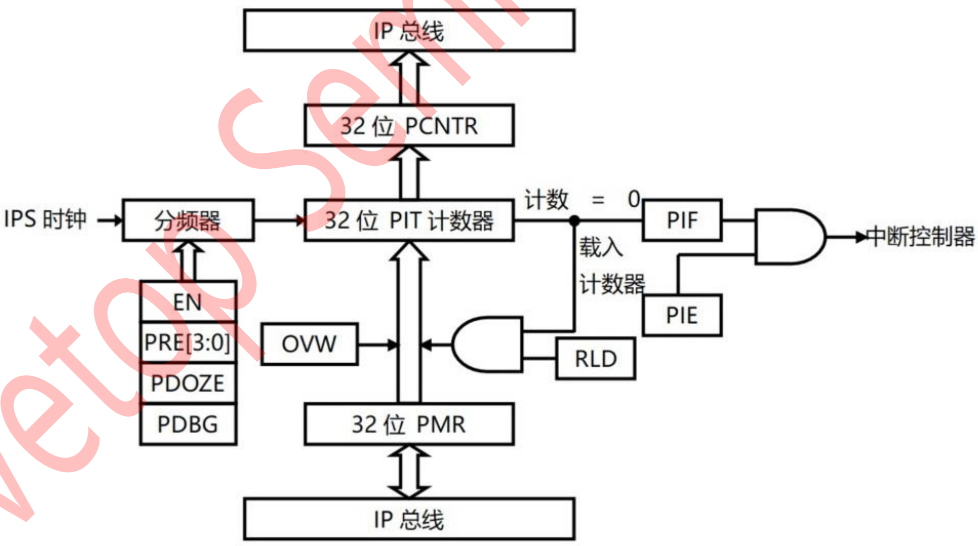
17.1 概述

32 位可编程中断计时器模块（PIT32）是一个 32 位计时器，在最少处理器干预的情况下提供精确的定时中断。计时器可以从模数锁存器内写入的值开始递减，也可以是一个自由运行的降值计数器。

17.2 特性

32 位计时器，在最少处理器干预的情况下提供精确的定时中断。
可以从模数锁存器内写入的值开始递减，也可以是一个自由运行的降值计数器。

17.3 框图



图表 17-1：PIT32 框图

17.4 工作模式

本节描述 PIT32 模块在三种低功耗模式和调试模式下的运行情况

。

17.4.1 等待模式

在等待模式下，PIT32 模块会继续正常运行，并且能够被配置为通过产生一个中断请求，使芯片退出低功耗模式。

17.4.2 瞌睡模式

在瞌睡模式下，当 PIT32 控制和状态寄存器（PCSR）中的 PDOZE 位设置为 1，PIT32 模块运行停止。在瞌睡模式下，当 PDOZE 位清零，瞌睡模式不会影响 PIT32 的运行。当退出瞌睡模式时，PIT32 继续从其在进入瞌睡模式前的状态运行。

17.4.3 停止模式

在停止模式下，由于系统时钟缺失，PIT32 模块停止运行。

17.4.4 调试模式

在调试模式下，当 PIT32 控制和状态寄存器（PCSR）中的 PDBG 位设置为 1，PIT32 模块运行停止。在调试模式下，当 PDBG 位清零，调试模式不会影响 PIT32 的运行。当退出调试模式时，PIT32 继续从其在进入调试模式前的状态运行，但是调试模式下所做的任何更改都将保持。

17.5 外部管脚

没有片外信号。

17.6 内存映射和寄存器

本节描述 PIT32 模块的内存映射和寄存器结构。

17.6.1 内存映射

引用表格 17-1 来描述 PIT32 模块的内存映射。

表格 17-1：可编程中断计时器模块内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	保留	PIT32 控制和状态寄存器（PCSR）	S
0x0004	PIT32 模数寄存器（PMR）		S
0x0008	PIT32 计数寄存器（PCNTR）		S/U
0x000C	保留		---

注意：S = 超级用户访问；U = 普通用户访问

17.6.2 寄存器描述

PIT32 可编程模块包括 3 个寄存器：

PIT32 控制和状态寄存器（PCSR）配置计时器运行。参看 PIT32 控制和状态寄存器。

PIT32 模数寄存器（PMR）决定计时器重新装载的数值。参看 PIT32 模数寄存器。

PIT32 计数寄存器（PCNTR）反映计数器中的数值。参看 PIT32 计数寄存器。

17.6.2.1 PIT32 控制和状态寄存器

偏移地址：0x0000~0x0001 复位值：0x0000

15	14	13	12	11	10	9	8
保留				PRE3	PRE2	PRE1	PRE0
ro				rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留	PDOZE	PDBG	OVW	PIE	PIF	RLD	EN
ro	rw	rw	rw	rw	rw	rw	rw

图表 17-2：PIT32 控制和状态寄存器（PCSR）

比特位	名称	复位值	读写属性	功能说明
[15:12]	保留	0x0	RO	---
[11:8]	PRE[3:0]	0x0	RW	分频器位 可读/写的 PRE [3:0] 位用于选择由系统时钟分频 产生 PIT32 时钟的分频系数，如 表格 17-2 所示。为了能够准确地预知下一次计数的时间，修改 PRE [3:0] 位的数值只能在使能位（EN）清零的状态下进行。修改 PRE [3:0] 位将会重置分频计数器，系统复位或者重新载入计数器数值也会重置分频计数器。对 EN 位的设置和对 PRE [3:0] 位的写入可以在同一个时钟周期内完成。EN 位清零将会停止分频计数器。
[7]	保留	0x0	RO	---
[6]	PDOZE	0x0	RW	休眠时钟睡眠使能 配置在系统睡眠模式下休眠时钟是否使能。 0 = 禁止；1 = 使能

比特位	名称	复位值	读写属性	功能说明
[5]	PDBG	0x0	RW	调试模式位 可读/写的 PDBG 位控制 PIT 在调试模式下的功能。复位会将 PDBG 位清零。 0 = 在调试模式下，PIT 功能未受影响 1 = 在调试模式下，PIT 功能停止 在调试模式中，寄存器的读写访问功能正常。当退出调试模式时，计时器继续从其在进入调试模式前的状态运行，但是调试模式下所做的任何更改都将保持。 **注意：**在调试模式中将 PDBG 位由 1 更改为 0 将会启动 PIT 计时器，同样地，在调试模式中将 PDBG 位由 0 更改为 1 将停止 PIT 计时器。
[4]	OVW	0x0	RW	可读/写的 OVW 位用于使能将 PMR 寄存器中的数值立即写入 PIT 计数器并将计数器内原数值覆盖。 0 = 当 PIT 计数器到达 0x0000 时，PIT 计数器中的数值会被替换为 PMR 的数值。 1 = 写 PMW 时 PIT 计数器中的数值将会被立刻替换为 PMW 的数值。
[3]	PIE	0x0	RW	PIT 中断使能位 可读/写的 PIE 位使能 PIF 标识来产生中断请求。 0 = PIF 中断请求关闭 1 = PIF 中断请求开启
[2]	PIF	0x0	RW	PIT 中断标识 当 PIT 计数器到达 0x0000 时，可读/写的 PIF 标识被置 1。通过对 PIF 写入 1 或者写 PMR 寄存器能够清除 PIF 标识。写入 0 无效。复位会清将 PIF 清零。 0 = PIT 计数器未达到 0x0000 1 = PIT 计数器达到 0x0000
[1]	RLD	0x0	RW	重装位 可读/写的 RLD 位用于使能当计数器到达 0x0000 时，将 PMR 寄存器中的数值重新载入到 PIT 计数器内。 0 = 计数到 0x0000 时，计时器重置为 0xFFFF 1 = 计数到 0x0000 时，将 PMR 的数值重新载入到计数器中

比特位	名称	复位值	读写属性	功能说明
[0]	EN	0x0	RW	PIT 使能位 可读/写的 EN 位使能 PIT 模块运行。当 PIT 关闭时，计时器和分频器会保持在停止状态。 0 = PIT 关闭；1 = PIT 开启

表格 17-2：分频器选择编码

PRE[3：0]	系统时钟分频系数
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1024
1011	2048
1100	4,096
1101	8,192
1110	16,384
1111	32,768

17.6.2.2 PIT32 模数寄存器

当 PIT32 计数器达到 0x0000，并且 RLD 位被置 1 时，32 位可读/写的 PIT32 模数寄存器（PMR）中的计时器模数值，将会被载入到 PIT32 计数器中。

当 OVW 位被置 1 时，PMR 是透明的，写入 PMR 的值会立刻被载入到 PIT32 计时器中。当一个新的数值被载入到 PIT32 计数器中，或者在芯片复位过程中，分频计数器都会被重置。读 PMR 寄存器会返回写入的模数锁存器中的数值。PMR 寄存器复位后的数值是 0xFFFFFFFF。

偏移地址：0x0004~0x0007　　　　复位值：0xFFFFFFFF

31	30	29	28	27	26	25	24
PM31	PM30	PM29	PM28	PM27	PM26	PM25	PM24

rw	rw	rw	rw	rw	rw	rw	rw
23	22	21	20	19	18	17	16
PM23	PM22	PM21	PM20	PM19	PM18	PM17	PM16
rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8
PM15	PM14	PM13	PM12	PM11	PM10	PM9	PM8
rw	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
rw	rw	rw	rw	rw	rw	rw	rw

图表 17-3：PIT32 模数寄存器（PMR）

17.6.2.3 PIT32 计数寄存器

32 位只读的 PIT32 计数寄存器（PCNTR）反映计数器的数值。通过连续读取高低两个 16 位数值来组成这个 32 位计数器数值的方法不能保证数值是连贯的。对 PCNTR 写入无效，并且写周期将正常终止。

偏移地址：0x0008~0x000B　　　　复位值：0xFFFFFFFF

31	30	29	28	27	26	25	24
PC31	PC30	PC29	PC28	PC27	PC26	PC25	PC24
ro	ro	ro	ro	ro	ro	ro	ro
23	22	21	20	19	18	17	16
PC23	PC22	PC21	PC20	PC19	PC18	PC17	PC16
ro	ro	ro	ro	ro	ro	ro	ro
15	14	13	12	11	10	9	8
PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8
ro	ro	ro	ro	ro	ro	ro	ro
7	6	5	4	3	2	1	0
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
ro	ro	ro	ro	ro	ro	ro	ro

图表 17-4：PIT32 计数寄存器（PCNTR）

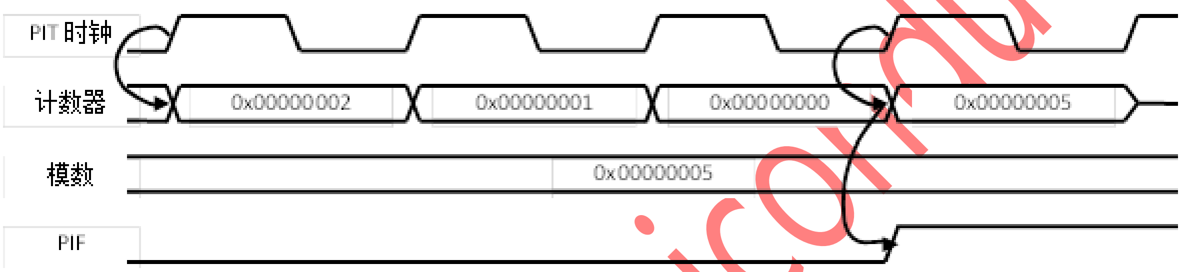
17.7 功能描述

本节描述 PIT32 模块的功能操作。

17.7.1 一次性设置计时器操作

当 PCSR 内的 RLD 位被置 1 时，此种工作模式就被选定了。当 PIT32 计时器数到 0x0000 时，PCSR 寄存器中的 PIF 标识被置 1。模数锁存器内的数值会被载入到计数器中，并且计数器数值开始递减直到 0x0000。如果 PCSR 寄存器中的 PIE 标识被置 1，那么此时 PIF 标识会向 CPU 发出一个中断请求。

当 PCSR 寄存器中的 OVW 位被置 1 时，初始化计数器的数值可以通过直接写 PMR 寄存器完成，而不需要等到计数器到达 0x0000。

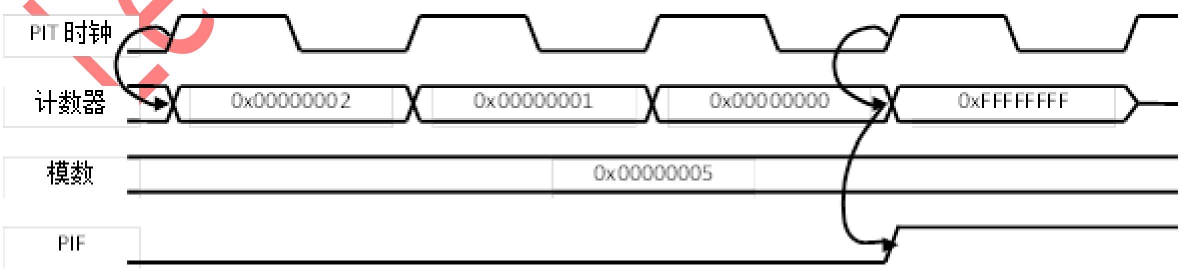


图表 17-5：计数器数值从模数锁存器中重新载入

17.7.2 自由运行的计时器操作

当 PCSR 内的 RLD 位被清零时，此种工作模式就被选定了。在此种模式下，PIT32 计数器数到 0x0000 后，不会载入模数锁存器中的数据，而是将计数器的数值滚动到 0xFFFFFFFF，然后继续递减。

如果 PCSR 寄存器中的 PIE 标识被置 1，那么此时 PIF 标识会向 CPU 发出一个中断请求。当 PCSR 寄存器中的 OVW 位被置 1 时，初始化计数器的数值可以通过直接写 PMR 寄存器完成，而不需要等到计数器到达 0x0000。



图表 17-6：自由工作模式下的计数器

17.7.3 定时说明

32 位 PIT32 计数器和分频器支持不同的定时时长。分频器按照 PCSR 寄存器中的 PRE［3:0］位选定的分频系数将系统时钟进行分频。PMR 寄存器的 PM［31:0］位选定了定时时长。

$$定时时长 = 2^{PRE[3:0]} \times (PM[31:0]+1) \times 系统时钟周期$$

17.8 中断描述

表格 17-3 列出了 PIT32 产生的中断请求。

表格 17-3：PIT32 中断请求

中断请求	标识	使能位
定时	PIF	PIE

当 PIT32 计时器数到 0x0000 时，PIF 标识被置 1。PIE 位使能 PIF 标识产生中断请求。通过向 PIF 写入 1 或者写 PMR 寄存器可以清除 PIF 标识。

18 中断向量嵌套控制器（NVIC）

18.1 概述

本章节描述中断向量嵌套控制器。

18.2 特性

NVIC 的主要特性如下：

- 64 个可屏蔽中断通道（不包括 16 条带 FPU 的 Cotex-M4 的中断线）；
- 8 个可编程优先级（使用 3 位中断优先级）；
- 低延迟异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现；NVIC 和处理器核心接口是紧密耦合的，这使得低延时中断处理和有效处理后到达的中断成为可能。所有的中断包括内核异常都由 NVIC 管理。

18.3 中断和异常向量

Cortex-M4 处理器提供了一个功能强大的异常处理架构，它支持大量的系统异常和外部中断。对异常进行了编号，编号 1 到 15 为系统异常，编号 16 及以上为中断输入（输入到处理器，但不一定需要在封装的 I/O 管脚上访问）。大多数的异常和所有的中断具有可编程优先级，只有一小部分的系统异常有固定的优先级。

表格 18-1：向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000_0000
	-3	固定	Reset	复位	0x0000_0004
	-2	固定	NMI	不可屏蔽中断	0x0000_0008
	-1	固定	硬件错误 (HardFault)	各类故障	0x0000_000C
	-	可设置	存储管理 (MemManage)	存储器管理	0x0000_0010
	-	可设置	总线错误 (BusFault)	预取指错误，存储器访问错误	0x0000_0014
	-	可设置	应用错误 (UsageFault)	未定义指令或非法状态	0x0000_0018
	-	-	-	保留	0x0000_001C-0x0000_002B

位置	优先级	优先级类型	名称	说明	地址
	-	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C
	-	-	-	保留	0x0000_0030
	-	-		保留	0x0000_0034
	-	可设置	PendSV	系统服务的挂起请求	0x0000_0038
	-	可设置	SysTick	系统滴答定时器	0x0000_003C
0	-	可设置	EFM	EFM 闪存中断	0x0000_0040
1	-	可设置	PMU	vcc_lvdt5v 中断 vcc_lvdt1p8v 中断 card0/1_lvd 中断 card0/1_en_fail 中断	0x0000_0044
2	-	可设置	TC	计时器中断	0x0000_0048
3	-	可设置	PIT1	Pit1 中断	0x0000_004C
4	-	可设置	PIT2	Pit2 中断	0x0000_0050
5	-	可设置	EDMAC1	EDMAC1 中断	0x0000_0054
6	-	可设置	EDMAC2	EDMAC2 中断	0x0000_0058
7	-	可设置	DMAC1	DMAC1 中断	0x0000_005C
8	-	可设置	DMAC2	DMAC2 中断	0x0000_0060
9	-	可设置	保留	保留	0x0000_0064
10	-	可设置	TRNG	TRNG 中断	0x0000_0068
11	-	可设置	保留	保留	0x0000_006C
12	-	可设置	保留	保留	0x0000_0070
13	-	可设置	Async Timer	wkp_timer0_ov_flag Async timer 中断	0x0000_0074
14	-	可设置	保留	保留	0x0000_0078
15	-	可设置	PMU_RTC	rims_day_int rims_hour_int rims_minute_int rims_second_int Day_int Hour_int Minute_intt Second_int Alarm_intt 1khz_int / 32khz_int	0x0000_007C
16	-	可设置	保留	保留	0x0000_0080
17	-	可设置	SHA	SHA 完成中断	0x0000_0084
18	-	可设置	保留	保留	0x0000_0088
19	-	可设置	AES	AES 中断	0x0000_008C

位置	优先级	优先级类型	名称	说明	地址
20	-	可设置	QADC	adrdy_int eosmp_int eoc_int eoseq_int ovr_int ewd_int dbrdy_int dgatto_int fifoto_int	0x0000_0090
21	-	可设置	DAC	DAC 中断	0x0000_0094
22	-	可设置	保留	保留	0x0000_0098
23	-	可设置	TSI	TSI 中断	0x0000_009C
24	-	可设置	USBC	wkp_USBDET_IN wkp_ipi_int_usbc Usbc_int Otg_dma_int	0x0000_00A0
25	-	可设置	保留	保留	0x0000_00A4
26	-	可设置	SPI1	SPIF 中断 MODF 中断 SSF 中断	0x0000_00A8
27	-	可设置	SPI2	SPIF 中断 MODF 中断 SSF 中断	0x0000_00AC
28	-	可设置	SPI3	wkp_ipp_ind_pl2_ss3 SPIF 中断 MODF 中断 SSF 中断	0x0000_00B0
29	-	可设置	SPIM1	SS1 中断	0x0000_00B4
30	-	可设置	SPIM2	SS2 中断	0x0000_00B8
31	-	可设置	保留	保留	0x0000_00BC
32	-	可设置	SCI1	TDRE / TC/ RDRF/TIMOUT/OR /IDLE 中断	0x0000_00C0
33	-	可设置	SCI2	TDRE / TC/ RDRF/TIMOUT/OR /IDLE 中断	0x0000_00C4
34	-	可设置	USI2	ATRI/ ERI/PEI/TRI 中断	0x0000_00C8
35	-	可设置	保留	保留	0x0000_00CC
36	-	可设置	I2C1	I2C1 中断	0x0000_00D0
37	-	可设置	PWM	PWM[0]中断	0x0000_00D4

位置	优先级	优先级类型	名称	说明	地址
38	-	可设置	PWM	PWM[1]中断	0x0000_00D8
39	-	可设置	PWM	PWM[2]中断	0x0000_00DC
40	-	可设置	PWM	PWM[3]中断	0x0000_00E0
41	-	可设置	EPORT1/EPORT3	EPF0 /EPF16 中断	0x0000_00E4
42	-	可设置	EPORT1/EPORT3	EPF1 /EPF17 中断	0x0000_00E8
43	-	可设置	EPORT1/EPORT3	EPF2 /EPF18 中断	0x0000_00EC
44	-	可设置	EPORT1/EPORT3	EPF3 /EPF19 中断	0x0000_00F0
45	-	可设置	EPORT1/EPORT3	EPF4 /EPF20 中断	0x0000_00F4
46	-	可设置	EPORT1/EPORT3	EPF5 /EPF21 中断	0x0000_00F8
47	-	可设置	EPORT1/EPORT3	EPF6 /EPF22 中断	0x0000_00FC
48	-	可设置	EPORT1/EPORT3	EPF7 /EPF23 中断	0x0000_0100
49	-	可设置	EPORT2/EPORT4 /EPORT5	EPF8 /EPF24/EPF32 中断	0x0000_0104
50	-	可设置	EPORT2/EPORT4 /EPORT5	EPF9 /EPF25/EPF33 中断	0x0000_0108
51	-	可设置	EPORT2/EPORT4 /EPORT5	EPF10 /EPF26/EPF34 中断	0x0000_010C
52		可设置	EPORT2/EPORT4 /EPORT5	EPF11 /EPF27/EPF35 中断	0x0000_0110
53		可设置	EPORT2/EPORT4 /EPORT5	EPF12 /EPF28/EPF36 中断	0x0000_0114
54		可设置	EPORT2/EPORT4 /EPORT5	EPF13 /EPF29/EPF37 中断	0x0000_0118
55		可设置	EPORT2/EPORT4 /EPORT5	EPF14 /EPF30/EPF38 中断	0x0000_011C
56		可设置	EPORT2/EPORT4 /EPORT5	EPF15 /EPF31/EPF39 中断	0x0000_0120
57		可设置	保留	保留	0x0000_0124
58		可设置	保留	保留	0x0000_0128
59		可设置	I2C2	I2C2 中断	0x0000_012C
60		可设置	I2C3	I2C3 中断	0x0000_0130
61		可设置	SCI3	TDRE / TC/ RDRF/TIMOUT/OR /IDLE 中断	0x0000_0134
62		可设置	保留	保留	0x0000_0138
63	-	可设置	USI1	ATRI/ ERI/PEI/TRI 中断	0x0000_0013C

18.4 寄存器

18.4.1 中断相关系统控制模块（SCB）寄存器

基地址：0xE000ED00

18.4.1.1 中断控制及状态寄存器（ICSR）

偏移地址：0x0004~0x0007　　复位值：0x00000000

31	30	29	28	27	26	25	24
NMIPENDSET	保留		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	保留
rw	ro	ro	rw	wo	rw	wo	ro
23	22	21	20	19	18	17	16
ISRPRE-EMPT	ISRPENDING	VECTPENDING					
ro	ro	ro	ro	ro	ro	ro	ro
15	14	13	12	11	10	9	8
VECTPENDING				RETTOBASE	保留	VECTACTIVE	
ro	ro	ro	ro	ro	ro	ro	ro
7	6	5	4	3	2	1	0
VECTACTIVE							
ro	ro	ro	ro	ro	ro	ro	ro

图表 18-1：中断控制及状态寄存器（ICSR）

比特位	名称	复位值	读写属性	功能说明
[31]	NMIPENDSET	0x0	RW	NMI 挂起设置，读取返回 NMI 挂起状态。 0 = 写 0 无影响；1 = 写 1 挂起 NMI
[30:29]	保留	0x0	RO	---
[28]	PENDSVSET	0x0	RW	系统调用挂起设置，读取返回 SV 挂起状态。 0 = 写 0 无影响；1 = 写 1 挂起 SV
[27]	PENDSVCLR	0x0	WO	系统调用挂起清除。 0 = 写 0 无影响；1 = 写 1 清除 SV 挂起状态
[26]	PENDSTSET	0x0	RW	SYSTICK 挂起设置，读取返回 SYSTICK 挂起的状态。 0 = 写 0 无影响；1 = 写 1 挂起 SYSTICK
[25]	PENDSTCLR	0x0	WO	SYSTICK 挂起设置，读取返回 SYSTICK 挂起的状态。 0 = 写 0 无影响；1 = 写 1 挂起 SYSTICK
[24]	保留	0x0	RO	---

比特位	名称	复位值	读写属性	功能说明
[23]	ISRPRE-EMPT	0x0	RO	表明一个挂起的中断将在下一步时进入活动状态（用于单步执行时的调试目的）。 0 = 没有挂起的中断将在下一步时进入活动状态 1 = 一个挂起的中断将在下一步时进入活动状态
[22]	ISRPENDING	0x0	RO	是否外部中断正在挂起（不包括系统异常，例如 NMI）。 0 = 没有外部中断被挂起 1 = 外部中断正在被挂起
[21:12]	VECTPENDING [9:0]	0x0	RO	挂起的 ISR 编号。
[11]	RETTOBASE	0x0	RO	返回线程。 0 = 处理器没有运行异常处理程序 1 = 设置为 1，当处理器运行一个异常处理程序时，如果中断返回且没有其他异常等待，则返回线程级别。
[10]	保留	0x0	RO	---
[9:0]	VECTACTIVE[9:0]	0x0	RO	当前运行的 ISR 编号。

18.4.1.2 向量表地址偏移寄存器（VTOR）

偏移地址：0x0008~0x000B　　　　复位值：0x00000000

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

图表 18-2：向量表地址偏移寄存器（VTOR）

18.4.1.3 应用程序中断和复位控制寄存器（AIRCR）

偏移地址：0x000C~0x000F 复位值：0x00000000

31	30	29	28	27	26	25	24
VECTKEY[15:8]							
rw	rw	rw	rw	rw	rw	rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VECTKEY[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ENDIANESS	保留				PRIGROUP[2:0]		
ro	ro	ro	ro	ro	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留					SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
ro	ro	ro	ro	ro	rw	rw	rw

图表 18-3：应用程序中断和复位控制寄存器（AIRCR）

比特位	名称	复位值	读写属性	功能说明
[31:16]	VECTKEY[15:0]	0x0	RW	访问钥匙：任何对该寄存器的写操作，都必须同时把 0x05FA 写入此段，否则写操作被忽略。若读取此半字，则返回 0xFA05。
[15]	ENDIANESS	0x0	RO	表明数据的大小端模式。只能在复位后改变。 0 = 小端；1 = 大端
[14:11]	保留	0x0	RO	---
[10:8]	PRIGROUP[2:0]	0x0	RW	优先级分组。该字段表示将 IPR 中 PRI_n 字段分割为单独的抢占优先级和响应优先级字段的二进制位置。
[7:3]	保留	0x0	RO	---
[2]	SYS RESET REQ	0x0	RW	系统复位请求。这将强制除调试之外的所有组件进行复位。读返回 0。 0 = 没有系统复位请求 1 = 产生一个信号给系统请求复位
[1]	VECT CLR ACTIVE	0x0	RW	清除所有异常活动的状态信息。通常只在调试时用，或者在 OS 从错误中恢复时用。
[0]	VECT RESET	0x0	RW	复位 Cortex-M3/M4 处理器内核（调试逻辑除外）。但是此复位不影响处理器外部的电路。用于调试操作，不要与 SYS RESET REQ 同时使用。

18.4.1.4 系统控制寄存器（SCR）

偏移地址：0x0010~0x0013 复位值：0x00000000

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

图表 18-4：系统控制寄存器（SCR）

比特位	名称	复位值	读写属性	功能说明
[31:5]	保留	0x0	RW	---
[4]	SEVONPEND	0x0	RW	挂起发送事件位。当一个事件或者中断进入挂起状态，事件信号将处理器从 WFE 指令处唤醒。如果处理器不是在等待一个事件，这个事件将影响下一个 WFE 指令。处理器在执行 SEV 指令或发生外部事件时也会唤醒。 0 = 只允许使能的中断或者事件能够唤醒处理器，未使能的中断不可以 1 = 允许事件和所有中断（包括未使能中断）能够唤醒处理器
[3]	保留	0x0	RW	---
[2]	SLEEPDEEP	0x0	RW	控制处理在低功耗模式下使用 sleep 或 deep sleep。 0 = Sleep；1 = Deep sleep
[1]	SLEEPONEXIT	0x0	RW	从 Handler 模式返回线程模式时配置 sleep-on-exit。该位置 1 可以使中断驱动应用程序避免返回到空的应用程序。 0 = 当返回线程模式时不进入 sleep 1 = 从中断服务例程中返回时进入 sleep 或者 deep sleep
[0]	保留	0x0	RW	---

18.4.1.5 配置与控制寄存器（CCR）

偏移地址：0x0014~0x0017 复位值：0x00000000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">保留</td></tr>
<tr><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">保留</td></tr>
<tr><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">保留</td><td>STKALIGN</td><td>BFHFNMIGN</td></tr>
<tr><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">保留</td><td>DIV_0_TRP</td><td>UNALIGN_TRP</td><td>保留</td><td>USERSETPEND</td><td>NONBASETHRDENA</td></tr>
<tr><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
</table>

图表 18-5：配置与控制寄存器（CCR）

比特位	名称	复位值	读写属性	功能说明
[31:10]	保留	0x0	RW	---
[9]	STKALIGN	0x0	RW	配置异常进入时的堆栈对齐方式。在进入异常时，处理器使用堆栈 PSR 的第 9 位来来指示堆栈对齐方式。当从异常返回，处理器用该堆栈位恢复正确的堆栈对齐。 0 = 4-byte 对齐；1 = 8-byte 对齐
[8]	BFHFNMIGN	0x0	RW	允许在 HardFalut 与 NMI 服务例程中忽略由于加载和保存指令导致的数据总线错误。只有当服务例程及其数据处于绝对安全的内存中时，才将此位设置为 1。通常使用此位来探测系统设备和桥接器，以检测和修复控制路径问题。 0 = 加载和保存指令引起的数据总线错误导致锁定。 1 = HardFalut 与 NMI 服务例程运行时忽略由于加载和保存指令导致的数据总线错误。
[7:5]	保留	0x0	RW	---
[4]	DIV_0_TRP	0x0	RW	当处理器执行 SDIV 或者 UDIC 指令除零时允许产生错误或者停止。 0 = 不陷入除零错误；1 = 陷入除零错误

比特位	名称	复位值	读写属性	功能说明
[3]	UNALIGN_TRP	0x0	RW	访问未对齐时允许陷入用法错误。如果该位设置为 1，访问未对齐产生一个用法错误。未对齐的 LDM，STM，LDRD 和 STRD 指令总是错误的，不管 UNALIGN_TRP 是不是设置为 1。 0 = 不陷入半字或字访问未对齐错误 1 = 陷入半字或字访问未对齐错误
[2]	保留	0x0	RW	---
[1]	USERSETPEND	0x0	RW	允许用户软件访问 STIR。 0 = 不允许；1 = 允许
[0]	NONBASETHRDENA	0x0	RW	配置处理器如何进入线程模式。 0 = 只有当没有异常活动时，处理器可以进入线程模式 1 = 通过控制 EXC_RETURN 的值，处理器可以从任何级别进入线程模式

18.4.1.6 系统异常优先级寄存器（SHP）

偏移地址：0x0018+i(i = 0~11)　　　　复位值：0x00

31	30	29	28	27	26	25	24
SHP[i][7:5]			0	0	0	0	0
rw	rw	rw	ro	ro	ro	ro	ro

图表 18-6：系统异常优先级寄存器（SHP）

比特位	名称	复位值	读写属性	功能说明
[31:39]	SHP[i][7:5]	0x0	RW	3 个位用于设置系统异常优先级。 SHP[0]：MemManage Fault 优先级 SHP[1]：Bus Falut 优先级 SHP[2]：Usage Fault 优先级 SHP[7]：SVC 优先级 SHP[10]：PendSV 优先级 SHP[11]：SysTick 优先级 SHP[3]/[4]/[5]/[6]/[8]/[9]未实现。
[28:24]	保留	0x0	RO	---

18.4.1.7 系统处理过程控制及状态寄存器（SHCSR）

偏移地址：0x0024~0x0027 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro	ro	ro	ro	ro	ro	ro	ro

23	22	21	20	19	18	17	16
保留					USGFAULTENA	BUSFAULTENA	MEMFAULTENA
ro	ro	ro	ro	ro	rw	rw	rw

15	14	13	12	11	10	9	8
SVCALLPENDED	BUSFAULTPENDED	MEMFAULTPENDED	USGFAULTPENDED	SYSTICKACT	PENDSVACT	保留	
rw	rw	rw	rw	rw	rw	ro	ro

7	6	5	4	3	2	1	0
SVCALLACT	保留			USGFAULTACT	保留	BUSFAULTACT	MEMFAULTACT
rw	ro	ro	ro	rw	ro	rw	rw

图表 18-7：系统处理过程控制及状态寄存器（SHCSR）

比特位	名称	复位值	读写属性	功能说明
[31:19]	保留	0x0	RO	---
[18]	USGFAULTENA	0x0	RW	用法错误服务例程使能位。 0 = 不使能；1 = 使能
[17]	BUSFAULTENA	0x0	RW	总线错误服务例程使能位。 0 = 不使能；1 = 使能
[16]	MEMFAULTENA	0x0	RW	存储器管理服务例程使能位。 0 = 不使能；1 = 使能
[15]	SVCALLPENDED	0x0	RW	SVCall 挂起中。SVCall 已经开始，但被更高优先级的异常取代。 0 = 未挂起；1 = 挂起
[14]	BUSFAULTPENDED	0x0	RW	总线错误挂起中。总线错误已经开始，但被更高优先级的异常取代。 0 = 未挂起；1 = 挂起
[13]	MEMFAULTPENDED	0x0	RW	存储器管理错误挂起中。存储器错误已经开始，但被更高优先级的异常取代。 0 = 未挂起；1 = 挂起

比特位	名称	复位值	读写属性	功能说明
[12]	USGFAULTPENDED	0x0	RW	用法错误挂起中。用法错误已经开始，但被更高优先级的异常取代。 0 = 未挂起；1 = 挂起
[11]	SYSTICKACT	0x0	RW	SYTICK 异常活动中。 0 = 未活动；1 = 活动中
[10]	PENDSVACT	0x0	RW	PendSV 异常活动中。 0 = 未活动；1 = 活动中
[9:8]	保留	0x0	RO	---
[7]	SVCALLACT	0x0	RW	SVCall 异常活动中。 0 = 未活动；1 = 活动中
[6:4]	保留	0x0	RO	---
[3]	USGFAULTACT	0x0	RW	用法错误异常活动中。 0 = 未活动；1 = 活动中
[2]	保留	0x0	RO	---
[1]	BUSFAULTACT	0x0	RW	总线错误异常活动中。 0 = 未活动；1 = 活动中
[0]	MEMFAULTACT	0x0	RW	存储器管理错误异常活动中。 0 = 未活动；1 = 活动中

18.4.2 NVIC 寄存器

基地址：0xE000E000

18.4.2.1 中断设置使能寄存器（ISER）

偏移地址：0x0100+0x4*i(i = 0~7)　　复位值：0x00000000

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

图表 18-8：中断设置使能寄存器（ISER）

比特位	名称	复位值	读写属性	功能说明
[31:0]	ISE[31:0]	0x0	RW	中断设置使能。 0 = 写 0 无效；1 = 写 1 设置中断使能

18.4.2.2 中断清除使能寄存器（ICER）

偏移地址：0x0180+0x4*i(i = 0~7)　　　　　　　　复位值：0x00000000

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

图表 18-9：中断清除使能寄存器（ISER）

比特位	名称	复位值	读写属性	功能说明
[31:0]	ICE[31:0]	0x0	RW	中断清除使能。 0 = 写 0 无效；1 = 写 1 清除中断使能

18.4.2.3 中断设置挂起寄存器（ISPR）

偏移地址：0x0200+0x4*i(i = 0~1)　　　　复位值：0x00000000

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

图表 18-10：中断清除使能寄存器（ISER）

比特位	名称	复位值	读写属性	功能说明
[31:0]	ISP[31:0]	0x0	RW	中断设置挂起状态。 0 = 写 0 无效 1 = 写 1 设置中断挂起状态

18.4.2.4 中断清除挂起寄存器（ICPR）

偏移地址：0x0280+0x4*i(i = 0~1)　　　　复位值：0x00000000

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

图表 18-11：中断清除使能寄存器（ISER）

比特位	名称	复位值	读写属性	功能说明
[31:0]	ICP[31:0]	0x0	RW	中断清除挂起状态。 0 = 写 0 无效；1 = 写 1 清除中断挂起状态

18.4.2.5 中断活动位寄存器（IABR）

偏移地址：0x0300+0x4*i(i = 0~1) 复位值：0x00000000

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

图表 18-12：中断活动位寄存器（IABR）

比特位	名称	复位值	读写属性	功能说明
[31:0]	IAB[31:0]	0x0	RO	中断活跃状态位。只读寄存器。 0 = 中断活动状态未置位 1 = 中断活动状态置位

18.4.2.6 中断优先级寄存器（IPR）

偏移地址：0x0400+i(i = 0~239)　　　　复位值：0x0000

31	30	29	28	27	26	25	24
IP[i][7:5]			保留				
rw	rw	rw	ro	ro	ro	ro	ro

图表 18-13：中断优先级寄存器（IPR）

比特位	名称	复位值	读写属性	功能说明
[31:29]	IP[i][7:5]	0x0	RW	每个中断的中断优先级（8 位宽度）。
[28:24]	保留	0x0	RO	---

表格 18-2：优先级组

	中断优先级 IP[7:5]		数量	
PRIGROUP[2:0]	抢占优先级位	响应优先级位	抢占优先级	响应优先级
0b111	无	[7:5]	无	8
0b110	[7]	[6:5]	2	4
0b101	[7:6]	[5]	4	2
其他	[7:5]	无	8	无

18.4.2.7 软件触发中断寄存器（STIR）

偏移地址：0x0F00~0x0F03　　复位值：0x00000000

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

图表 18-14：软件触发中断寄存器（STIR）

比特位	名称	复位值	读写属性	功能说明
[31:0]	STI[31:0]	0x0	RW	软件触发中断设置。写入中断号设置该中断为挂起状态。 0 = 写 0 无效；1 = 写 1 设置中断挂起状态

19 管脚控制模块（IO_CTRL）

19.1 概述

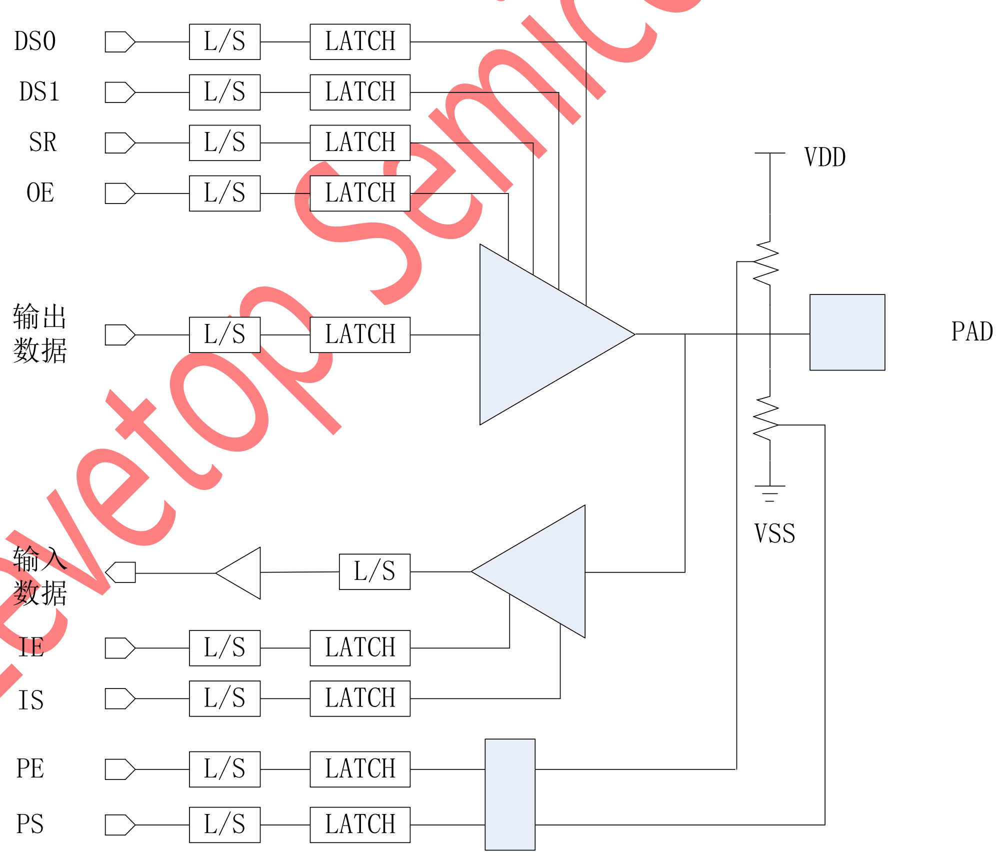
管脚控制模块用于配置芯片管脚的工作状态及功能。

19.2 特性

管脚控制模块的特性包括：

- 配置管脚方向
- 配置管脚的上下拉状态
- 配置管脚驱动能力
- 配置管脚的复用功能

19.3 框图



图表 19-1：IO_CTRL 框图

19.4 内存映射和寄存器

19.4.1 内存映射

管脚控制模块包含的寄存器如下表所示：

表格 19-1：管脚控制器偏移地址映像

偏移地址	位 31-16	位 15-0	访问
0x0000	SPI 管脚控制寄存器（SPICR）		S/U
0x0004	USI 管脚控制寄存器（USICR）		S/U
0x0008	I2C 管脚控制寄存器（I2CCR）		S/U
0x000C	SCI 管脚控制寄存器（SCICR）		S/U
0x0010	GPIOL 管脚控制寄存器（GINTLCR）		S/U
0x0014	GPIOH 管脚控制寄存器（GINTHCR）		S/U
0x0018	保留		----
0x001C	管脚功能控制寄存器（SWAPCR）		S/U
0x0020	SPIM1 管脚控制寄存器（SPIM1CR）		S/U
0x0024	SPIM2 管脚控制寄存器（SPIM2CR）		S/U
0x0028	保留		----
0x002C	GINT[31:30]管脚控制寄存器（GINT31:30CR）		S/U
0x0030~0x0034	保留		----
0x0038	GINT[39:32]管脚控制寄存器（GINT39:32CR）		S/U
0x003C	GINT[29:22]管脚控制寄存器（GINT29:22CR）		S/U
0x0040	保留		----
0x0044	CLKOUT/RSTOUT 管脚功能控制寄存器（CLKRSTCR）		S/U
0x0048~0x0050	保留		----

注意： S = 超级用户访问 ; U = 普通用户访问

19.4.2 寄存器描述

19.4.2.1 SPI 管脚控制寄存器

SPI 管脚控制寄存器的功能包括控制 SPI 管脚的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0000~0x0003　　　　复位值：0xFFFFFF05

31	30	29	28	27	26	25	24
MOSI3_IE	MISO3_IE	SCK3_IE	SS3_IE	MOSI3_PS	MISO3_PS	SCK3_PS	SS3_PS
rw	rw	rw	rw	rw	rw	rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MOSI2_IE	MISO2_IE	SCK2_IE	SS2_IE	MOSI2_PS	MISO2_PS	SCK2_PS	SS2_PS
rw	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MOSI1_IE	MISO1_IE	SCK1_IE	SS1_IE	MOSI1_PS	MISO1_PS	SCK1_PS	SS1_PS
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留				IS	SR	DS[1:0]	
ro				rw	rw	ro	

图表 19-2：SPI 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	MOSI3_IE	0x1	RW	SPI3 管脚 MOSI3 的输入功能是否开启。 0 = MOSI3 输入功能被禁止 1 = MOSI3 输入功能开启
[30]	MISO3_IE	0x1	RW	SPI3 管脚 MISO3 的输入功能是否开启。 0 = MISO3 输入功能被禁止 1 = MISO3 输入功能开启
[29]	SCK3_IE	0x1	RW	SPI3 管脚 SCK3 的输入功能是否开启。 0 = SCK3 输入功能被禁止 1 = SCK3 输入功能开启
[28]	SS3_IE	0x1	RW	SPI3 管脚 SS3 的输入功能是否开启。 0 = SS3 输入功能被禁止 1 = SS3 输入功能开启

比特位	名称	复位值	读写属性	功能说明
[27]	MOSI3_PS	0x1	RW	SPI3 管脚 MOSI3 的输入上下拉选择。 0 = MOSI3 选择输入下拉 1 = MOSI3 选择输入上拉
[26]	MISO3_PS	0x1	RW	SPI3 管脚 MISO3 的输入上下拉选择。 0 = MISO3 选择输入下拉 1 = MISO3 选择输入上拉
[25]	SCK3_PS	0x1	RW	SPI3 管脚 SCK3 的输入上下拉选择。 0 = SCK3 选择输入下拉 1 = SCK3 选择输入上拉
[24]	SS3_PS	0x1	RW	SPI3 管脚 SS3 的输入上下拉选择。 0 = SS3 选择输入下拉 1 = SS3 选择输入上拉
[23]	MOSI2_IE	0x1	RW	SPI2 管脚 MOSI2 的输入功能是否开启。 0 = MOSI2 输入功能被禁止 1 = MOSI2 输入功能开启
[22]	MISO2_IE	0x1	RW	SPI2 管脚 MISO2 的输入功能是否开启。 0 = MISO2 输入功能被禁止 1 = MISO2 输入功能开启
[21]	SCK2_IE	0x1	RW	SPI2 管脚 SCK2 的输入功能是否开启。 0 = SCK2 输入功能被禁止 1 = SCK2 输入功能开启
[20]	SS2_IE	0x1	RW	SPI2 管脚 SS2 的输入功能是否开启。 0 = SS2 输入功能被禁止 1 = SS2 输入功能开启
[19]	MOSI2_PS	0x1	RW	SPI2 管脚 MOSI2 的输入上下拉选择。 0 = MOSI2 选择输入下拉 1 = MOSI2 选择输入上拉
[18]	MISO2_PS	0x1	RW	SPI2 管脚 MISO2 的输入上下拉选择。 0 = MISO2 选择输入下拉 1 = MISO2 选择输入上拉
[17]	SCK2_PS	0x1	RW	SPI2 管脚 SCK2 的输入上下拉选择。 0 = SCK2 选择输入下拉 1 = SCK2 选择输入上拉
[16]	SS2_PS	0x1	RW	SPI2 管脚 SS2 的输入上下拉选择。 0 = SS2 选择输入下拉 1 = SS2 选择输入上拉
[15]	MOSI1_IE	0x1	RW	SPI1 管脚 MOSI1 的输入功能是否开启。 0 = MOSI1 输入功能被禁止

比特位	名称	复位值	读写属性	功能说明
				1 = MOSI1 输入功能开启
[14]	MISO1_IE	0x1	RW	SPI1 管脚 MISO1 的输入功能是否开启。 0 = MISO1 输入功能被禁止 1 = MISO1 输入功能开启
[13]	SCK1_IE	0x1	RW	SPI1 管脚 SCK1 的输入功能是否开启。 0 = SCK1 输入功能被禁止 1 = SCK1 输入功能开启
[12]	SS1_IE	0x1	RW	SPI1 管脚 SS1 的输入功能是否开启。 0 = SS1 输入功能被禁止 1 = SS1 输入功能开启
[11]	MOSI1_PS	0x1	RW	SPI1 管脚 MOSI1 的输入上下拉选择。 0 = MOSI1 选择输入下拉 1 = MOSI1 选择输入上拉
[10]	MISO2_PS	0x1	RW	SPI1 管脚 MISO1 的输入上下拉选择。 0 = MISO1 选择输入下拉 1 = MISO1 选择输入上拉
[9]	SCK2_PS	0x1	RW	SPI1 管脚 SCK1 的输入上下拉选择。 0 = SCK1 选择输入下拉 1 = SCK1 选择输入上拉
[8]	SS2_PS	0x1	RW	SPI1 管脚 SS1 的输入上下拉选择。 0 = SS1 选择输入下拉 1 = SS1 选择输入上拉
[7:4]	保留	0x0	RO	----
[3]	IS	0x0	RW	SPI1、SPI2 和 SPI3 管脚的输入类型。 0 = SPI 管脚为 CMOS 输入 1 = SPI 管脚为施密特触发器输入
[2]	SR	0x1	RW	SPI1、SPI2 和 SPI3 管脚电压转换率。 0 = SPI 管脚电压转换率为高速状态 1 = SPI 管脚电压转换率为低速状态
[1:0]	DS	0x1	RW	SPI1、SPI2 和 SPI3 管脚驱动能力管脚驱动能力如下: 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.2 USI 管脚控制寄存器

USI 管脚控制寄存器的功能包括控制 USI 管脚的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0004~0x0007　　　　复位值：0x707007C5

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留	ISOCLK2_DIEN	ISODAT2_DIEN	ISORST2_DIEN	保留	ISOCLK2_PUEN	ISODAT2_PUEN	ISORST2_PUEN
ro	rw	rw	rw	ro	rw	rw	rw

15	14	13	12	11	10	9	8
保留					ISOCLK1_PS	ISODAT1_PS	ISORST1_PS
ro					rw	rw	rw

7	6	5	4	3	2	1	0
保留	USI2_DREN	保留		IS	SR	DS[1:0]	
ro	rw	ro		rw	rw	ro	

图表 19-3：USI 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:23]	保留	0x0E0	RO	---
[22]	ISOCLK2_DIEN	0x1	RW	USI 管脚 ISOCLK2 的输入功能是否开启。 0 = ISOCLK2 输入功能被禁止 1 = ISOCLK2 输入功能开启
[21]	ISODAT2_DIEN	0x1	RW	USI 管脚 ISODAT2 的输入功能是否开启。 0 = ISODAT2 输入功能被禁止 1 = ISODAT2 输入功能开启
[20]	ISORST2_DIEN	0x1	RW	USI 管脚 ISORST2 的输入功能是否开启。 0 = ISORST2 输入功能被禁止 1 = ISORST3 输入功能开启
[19]	保留	0x0	RO	---
[18]	ISOCLK2_PUEN	0x0	RW	USI 管脚 ISOCLK2 的输入上拉功能是否开启。 0 = ISOCLK2 输入上拉功能被禁止 1 = ISOCLK2 输入上拉功能开启
[17]	ISODAT2_PUEN	0x0	RW	USI 管脚 ISODAT2 的输入上拉功能是否开启。

比特位	名称	复位值	读写属性	功能说明
				0 = ISODAT2 输入上拉功能被禁止 1 = ISODAT2 输入上拉功能开启
[16]	ISORST2_PUEN	0x0	RW	USI 管脚 ISORST2 的输入上拉功能是否开启。 0 = ISORST2 输入上拉功能被禁止 1 = ISORST2 输入上拉功能开启
[15:11]	保留	0x0	RO	---
[10]	ISOCLK1_PS	0x1	RW	USI 管脚 ISOCLK1 的输入上下拉选择。 0 = ISOCLK1 选择输入下拉 1 = ISOCLK1 选择输入上拉
[9]	ISODAT1_PS	0x1	RW	USI 管脚 ISODAT1 的输入上下拉选择。 0 = ISODAT2 选择输入下拉 1 = ISODAT2 选择输入上拉
[8]	ISORST1_PS	0x1	RW	USI 管脚 ISORST2 的输入上下拉选择。 0 = ISORST2 选择输入下拉 1 = ISORST2 选择输入上拉
[7]	保留	0x1	RO	---
[6]	USI2_DREN	0x1	RW	USI2 管脚驱动能力是否增强。 0 = 管脚驱动能力不增强 1 = 管脚驱动能力增强
[5:4]	保留	0x0	RO	---
[3]	IS	0x0	RW	USI 管脚的输入类型。 0 = USI 管脚为 CMOS 输入 1 = USI 管脚为施密特触发器输入
[2]	SR	0x1	RW	USI 管脚电压转换率。 0 = USI 管脚电压转换率为高速状态 1 = USI 管脚电压转换率为低速状态
[1:0]	DS	0x1	RW	USI 管脚驱动能力 管脚驱动能力如下： 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.3 I2C 管脚控制寄存器

I2C 管脚控制寄存器的功能包括控制 I2C 管脚的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0008~0x000B　　复位值：0x33333305

31	30	29	28	27	26	25	24
保留		SDA3_IE	SCL3_IE	保留		SDA3_PS	SCL3_PS
ro		rw	rw	ro		rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		SDA2_IE	SCL2_IE	保留		SDA2_PS	SCL2_PS
ro		rw	rw	ro		rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留		SDA_IE	SCL_IE	保留		SDA_PS	SCL_PS
ro		rw	rw	ro		rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-4：I2C 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:30]	保留	0x0	RO	----
[29]	SDA3_IE	0x1	RW	I2C3 管脚 SDA3 的输入功能是否开启。 0 = SDA3 输入功能被禁止 1 = SDA3 输入功能开启
[28]	SCL3_IE	0x1	RW	I2C3 管脚 SCL3 的输入功能是否开启。 0 = SCL3 输入功能被禁止 1 = SCL3 输入功能开启
[27:26]	保留	0x0	RO	----
[25]	SDA3_PS	0x1	RW	I2C3 管脚 SDA3 的输入上下拉选择。 0 = SDA3 选择输入下拉 1 = SDA3 选择输入上拉
[24]	SCL3_PS	0x1	RW	I2C3 管脚 SCL3 的输入上下拉选择。 0 = SCL3 选择输入下拉 1 = SCL3 选择输入上拉
[23:22]	保留	0x0	RO	----
[21]	SDA2_IE	0x1	RW	I2C2 管脚 SDA2 的输入功能是否开启。 0 = SDA2 输入功能被禁止 1 = SDA2 输入功能开启

比特位	名称	复位值	读写属性	功能说明
[20]	SCL2_IE	0x1	RW	I2C3 管脚 SCL3 的输入功能是否开启。 0 = SCL3 输入功能被禁止 1 = SCL3 输入功能开启
[19:18]	保留	0x0	RO	----
[17]	SDA2_PS	0x1	RW	I2C2 管脚 SDA2 的输入上下拉选择。 0 = SDA2 选择输入下拉 1 = SDA2 选择输入上拉
[16]	SCL2_PS	0x1	RW	I2C2 管脚 SCL2 的输入上下拉选择。 0 = SCL2 选择输入下拉 1 = SCL2 选择输入上拉
[15:14]	保留	0x0	RO	----
[13]	SDA2_IE	0x1	RW	I2C 管脚 SDA 的输入功能是否开启。 0 = SDA 输入功能被禁止 1 = SDA 输入功能开启
[12]	SCL_IE	0x1	RW	I2C 管脚 SCL 的输入功能是否开启。 0 = SCL 输入功能被禁止 1 = SCL 输入功能开启
[11:10]	保留	0x0	RO	---
[9]	SDA_PS	0x1	RW	I2C 管脚 SDA 的输入上下拉选择。 0 = SDA 选择输入下拉 1 = SDA 选择输入上拉
[8]	SCL_PS	0x1	RW	I2C 管脚 SCL 的输入上下拉选择。 0 = SCL 选择输入下拉 1 = SCL 选择输入上拉
[7:4]	保留	0x0	RO	---
[3]	IS	0x0	RW	I2C 管脚的输入类型。 0 = I2C 管脚为 CMOS 输入 1 = I2C 管脚为施密特触发器输入
[2]	SR	0x1	RW	I2C 管脚电压转换率。 0 = I2C 管脚电压转换率为高速状态 1 = I2C 管脚电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	I2C 管脚驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.4 SCI 管脚控制寄存器

SCI 管脚控制寄存器的功能包括控制 SCI 管脚的上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x000C~0x000F 复位值：0x803BBB05

31	30	29	28	27	26	25	24
SCI_GINT_SWAP_LOAD_EN	保留	SCI_GINT_SWAP[5:0]					
rw	ro	rw					

23	22	21	20	19	18	17	16
保留				SCI2_CTS_PUE	保留	RXD2_PS	TXD2_PS
ro				rw	ro	rw	rw

15	14	13	12	11	10	9	8
SCI3_CTS_PUE	保留	RXD3_PS	TXD3_PS	SCI1_CTS_PUE	保留	RXD_PS	TXD_PS
rw	ro	rw	rw	rw	ro	rw	rw

7	6	5	4	3	2	1	0
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-5：SCI 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	SCI_GINT_SWAP_LOAD_EN	0x1	RW	SCI 和 GINT 复用控制是否通过 flash load 0 = 复用关系不受 flash load 影响 1 = 复用关系由 flash load 控制
[30]	保留	0x0	RO	----
[29:24]	SCI_GINT_SWAP[5:0]	0x0	RW	SCI 和 GINT 复用使能信号 0 = 不复用（GINT 功能生效） 1 = 复用（SCI 功能生效） 六个 bit 对应 3 路串口： bit0: sci1 rxd bit1: sci3 txd bit2: sci3 rxd bit3: sci2 rxd bit4: sci1 txd bit5: sci2 txd

比特位	名称	复位值	读写属性	功能说明
[23:20]	保留	0x3	RO	----
[19]	SCI2_CTS_PUE	0x1	RW	SCI2 流控信号下拉使能 0 = 下拉无效 1 = 下拉有效
[18]	保留	0x0	RO	----
[17]	RXD2_PS	0x1	RW	SCI2 管脚 RXD2 的输入上下拉选择。 0 = RXD2 选择输入下拉 1 = RXD2 选择输入上拉
[16]	TXD2_PS	0x1	RW	SCI2 管脚 TXD2 的输入上下拉选择。 0 = TXD2 选择输入下拉 1 = TXD2 选择输入上拉
[15]	SCI3_CTS_PUE	0x1	RW	SCI3 流控信号下拉使能 0 = 下拉无效 1 = 下拉有效
[14]	保留	0x0	RO	----
[13]	RXD3_PS	0x1	RW	SCI3 管脚 RXD3 的输入上下拉选择。 0 = RXD3 选择输入下拉 1 = RXD3 选择输入上拉
[12]	TXD3_PS	0x1	RW	SCI3 管脚 TXD3 的输入上下拉选择。 0 = TXD3 选择输入下拉 1 = TXD3 选择输入上拉
[11]	SCI1_CTS_PUE	0x1	RW	SCI1 流控信号下拉使能 0 = 下拉无效 1 = 下拉有效
[10]	保留	0x0	RO	----
[9]	RXD_PS	0x1	RW	SCI 管脚 RXD 的输入上下拉选择。 0 = RXD 选择输入下拉 1 = RXD 选择输入上拉
[8]	TXD_PS	0x1	RW	SCI 管脚 TXD 的输入上下拉选择。 0 = TXD 选择输入下拉 1 = TXD 选择输入上拉
[7:4]	保留	0x0	RO	---
[3]	ISSR	0x0	RW	SCI 管脚的输入类型。 0 = SCI 管脚为 CMOS 输入 1 = SCI 管脚为施密特触发器输入
[2]	ISSR	0x1	RW	SCI 管脚电压转换率。 0 = SCI 管脚电压转换率为高速状态 1 = SCI 管脚电压转换率为低速状态

比特位	名称	复位值	读写属性	功能说明
[1:0]	DS[1:0]	0x1	RW	SCI 管脚驱动能力 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.5 GPIOL 管脚控制寄存器

GPIO1 管脚控制寄存器的功能包括控制第一组 GPIO 管脚 GINT0~GINT7 的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0010~0x0013　　复位值：0xFFFF0005

31	30	29	28	27	26	25	24
GINT7_PS	GINT6_PS	GINT5_PS	GINT4_PS	GINT3_PS	GINT2_PS	GINT1_PS	GINT0_PS
rw	rw	rw	rw	rw	rw	rw	rw
23	22	21	20	19	18	17	16
GINT7_IE	GINT6_IE	GINT5_IE	GINT4_IE	GINT3_IE	GINT2_IE	GINT1_IE	GINT0_IE
rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8
保留							
ro							
7	6	5	4	3	2	1	0
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-6：GPIOL 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	GINT7_PS	0x1	RW	GPIOL 管脚 GINT7 的输入上下拉选择。 0 = GINT7 选择输入下拉 1 = GINT7 选择输入上拉
[30]	GINT6_PS	0x1	RW	GPIOL 管脚 GINT6 的输入上下拉选择。 0 = GINT6 选择输入下拉 1 = GINT6 选择输入上拉
[29]	GINT5_PS	0x1	RW	GPIOL 管脚 GINT5 的输入上下拉选择。 0 = GINT5 选择输入下拉 1 = GINT5 选择输入上拉
[28]	GINT4_PS	0x1	RW	GPIOL 管脚 GINT4 的输入上下拉选择。 0 = GINT4 选择输入下拉 1 = GINT4 选择输入上拉
[27]	GINT3_PS	0x1	RW	GPIOL 管脚 GINT3 的输入上下拉选择。 0 = GINT3 选择输入下拉 1 = GINT3 选择输入上拉
[26]	GINT2_PS	0x1	RW	GPIOL 管脚 GINT2 的输入上下拉选择。 0 = GINT2 选择输入下拉 1 = GINT2 选择输入上拉

比特位	名称	复位值	读写属性	功能说明
[25]	GINT1_PS	0x1	RW	GPIOL 管脚 GINT1 的输入上下拉选择。 0 = GINT1 选择输入下拉 1 = GINT1 选择输入上拉
[24]	GINT0_PS	0x1	RW	GPIO 管 L 脚 GINT0 的输入上下拉选择。 0 = GINT0 选择输入下拉 1 = GINT0 选择输入上拉
[23]	GINT7_IE	0x1	RW	GPIOL 管脚 GINT7 的输入功能是否开启。 0 = GINT7 输入功能被禁止 1 = GINT7 输入功能开启
[22]	GINT6_IE	0x1	RW	GPIOL 管脚 GINT6 的输入功能是否开启。 0 = GINT6 输入功能被禁止 1 = GINT6 输入功能开启
[21]	GINT5_IE	0x1	RW	GPIOL 管脚 GINT5 的输入功能是否开启。 0 = GINT5 输入功能被禁止 1 = GINT5 输入功能开启
[20]	GINT4_IE	0x1	RW	GPIOL 管脚 GINT4 的输入功能是否开启。 0 = GINT4 输入功能被禁止 1 = GINT4 输入功能开启
[19]	GINT3_IE	0x1	RW	GPIOL 管脚 GINT3 的输入功能是否开启。 0 = GINT3 输入功能被禁止 1 = GINT3 输入功能开启
[18]	GINT2_IE	0x1	RW	GPIOL 管脚 GINT2 的输入功能是否开启。 0 = GINT2 输入功能被禁止 1 = GINT2 输入功能开启
[17]	GINT1_IE	0x1	RW	GPIOL 管脚 GINT1 的输入功能是否开启。 0 = GINT1 输入功能被禁止 1 = GINT1 输入功能开启
[16]	GINT0_IE	0x1	RW	GPIOL 管脚 GINT0 的输入功能是否开启。 0 = GINT0 输入功能被禁止 1 = GINT0 输入功能开启
[15:4]	保留	0x0	RO	---
[3]	IS	0x0	RW	GPIO 管脚 GINT0~GINT7 的输入类型。 0 = 管脚 GINT0~GINT7 为 CMOS 输入 1 = 管脚 GINT0~GINT7 为施密特触发器输入

比特位	名称	复位值	读写属性	功能说明
[2]	SR	0x1	RW	管脚 GINT0~GINT7 电压转换率。 0 = 管脚 GINT0~GINT7 电压转换率为高速状态 1 = 管脚 GINT0~GINT7 电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	管脚 GINT0~GINT7 驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.6 GPIOH 管脚控制寄存器

GPIO2 管脚控制寄存器的功能包括控制第二组 GPIO 管脚 GINT8~GINT15 的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0014~0x0017　　　　复位值：0xFFFF0005

31	30	29	28	27	26	25	24
GINT15_PS	GINT14_PS	GINT13_PS	GINT12_PS	GINT11_PS	GINT10_PS	GINT9_PS	GINT8_PS
rw	rw	rw	rw	rw	rw	rw	rw

23	22	21	20	19	18	17	16
GINT15_IE	GINT14_IE	GINT13_IE	GINT12_IE	GINT11_IE	GINT10_IE	GINT9_IE	GINT8_IE
rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-7：GPIOH 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	GINT15_PS	0x1	RW	GPIOH 管脚 GINT15 的输入上下拉选择。 0 = GINT15 选择输入下拉 1 = GINT15 选择输入上拉

比特位	名称	复位值	读写属性	功能说明
[30]	GINT14_PS	0x1	RW	GPIOH 管脚 GINT14 的输入上下拉选择。 0 = GINT14 选择输入下拉 1 = GINT14 选择输入上拉
[29]	GINT13_PS	0x1	RW	GPIOH 管脚 GINT13 的输入上下拉选择。 0 = GINT13 选择输入下拉 1 = GINT13 选择输入上拉
[28]	GINT12_PS	0x1	RW	GPIOH 管脚 GINT12 的输入上下拉选择。 0 = GINT12 选择输入下拉 1 = GINT12 选择输入上拉
[27]	GINT11_PS	0x1	RW	GPIOH 管脚 GINT11 的输入上下拉选择。 0 = GINT11 选择输入下拉 1 = GINT11 选择输入上拉
[26]	GINT10_PS	0x1	RW	GPIOH 管脚 GINT10 的输入上下拉选择。 0 = GINT10 选择输入下拉 1 = GINT10 选择输入上拉
[25]	GINT9_PS	0x1	RW	GPIOH 管脚 GINT9 的输入上下拉选择。 0 = GINT9 选择输入下拉 1 = GINT9 选择输入上拉
[24]	GINT8_PS	0x1	RW	GPIOH 管脚 GINT8 的输入上下拉选择。 0 = GINT8 选择输入下拉 1 = GINT8 选择输入上拉
[23]	GINT15_IE	0x1	RW	GPIOH 管脚 GINT15 的输入功能是否开启。 0 = GINT15 输入功能被禁止 1 = GINT15 输入功能开启
[22]	GINT14_IE	0x1	RW	GPIOH 管脚 GINT14 的输入功能是否开启。 0 = GINT14 输入功能被禁止 1 = GINT14 输入功能开启
[21]	GINT13_IE	0x1	RW	GPIOH 管脚 GINT13 的输入功能是否开启。 0 = GINT13 输入功能被禁止 1 = GINT13 输入功能开启
[20]	GINT12_IE	0x1	RW	GPIOH 管脚 GINT12 的输入功能是否开启。 0 = GINT12 输入功能被禁止 1 = GINT12 输入功能开启
[19]	GINT11_IE	0x1	RW	GPIOH 管脚 GINT11 的输入功能是否开启。 0 = GINT11 输入功能被禁止 1 = GINT11 输入功能开启

比特位	名称	复位值	读写属性	功能说明
[18]	GINT10_IE	0x1	RW	GPIOH 管脚 GINT10 的输入功能是否开启。 0 = GINT10 输入功能被禁止 1 = GINT10 输入功能开启
[17]	GINT9_IE	0x1	RW	GPIOH 管脚 GINT9 的输入功能是否开启。 0 = GINT9 输入功能被禁止 1 = GINT9 输入功能开启
[16]	GINT8_IE	0x1	RW	GPIOH 管脚 GINT8 的输入功能是否开启。 0 = GINT8 输入功能被禁止 1 = GINT8 输入功能开启
[15:4]	保留	0x0	RO	---
[3]	IS	0x0	RW	GPIO 管脚 GINT8~GINT15 的输入类型。 0 = 管脚 GINT8~GINT15 为 CMOS 输入 1 = 管脚 GINT8~GINT15 为施密特触发器输入
[2]	SR	0x1	RW	管脚 GINT8~GINT15 电压转换率。 0 = 管脚 GINT8~GINT15 电压转换率为高速状态 1 = 管脚 GINT8~GINT15 电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	管脚 GINT8~GINT15 驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.7 管脚功能控制寄存器

管脚功能控制寄存器用来控制特定管脚的功能是否开启。具体控制信息如下：

偏移地址：0x001C~0x001F 复位值：0x00000002

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

图表 19-8：管脚功能控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:24]	保留	0x0	RO	---
[23]	gint[7]/sci3_cts	0x0	RW	gint[7]和 sci3_cts 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（sci3 功能有效）
[22]	gint[6]/sci3_rts	0x0	RW	gint[6]和 sci3_rts 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（sci3 功能有效）
[21]	gint[13]/sci2_cts	0x0	RW	gint[13]和 sci2_cts 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（sci2 功能有效）
[20]	gint[12]/sci2_rts	0x0	RW	gint[12]和 sci2_rts 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（sci2 功能有效）
[19]	gint[15]/sci1_cts	0x0	RW	gint[15]和 sci1_cts 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（sci1 功能有效）
[18]	gint[14]/sci1_rts	0x0	RW	gint[14]和 sci1_rts 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（sci1 功能有效）

比特位	名称	复位值	读写属性	功能说明
[17]	gint[13]和 trace	0x0	RW	gint[13]和 trace 是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（trace 功能有效）
[16]	flash test pin	0x0	RW	flash 测试管脚是否使能。 0 = 不使能 1 = 使能
[15]	spi4[3:2]/gint[17:16]	0x0	RW	spi4[3:2]和 gint[17:16]是否复用。 0 = 不复用（spi4 功能有效） 1 = 复用（gint[17:16]功能有效）
[14]	ss_spi4,sck_spi4,spi4[1:0]/gint[21:18]	0x0	RW	ss_spi4，sck_spi4，spi4[1:0]和 gint[21:18]是否复用。 0 = 不复用（spi4 功能有效） 1 = 复用（gint[21:18]功能有效）
[13]	gint[25:22]	0x0	RW	gint[25:22]是否使能。 0 = 不使能 1 = 使能
[12]	gint[31:26]	0x0	RW	gint[31:26]是否使能。 0 = 不使能 1 = 使能
[11]	miso2	0x0	RW	miso2 是否使能。 0 = 使能（spi2 功能有效） 1 = 不使能
[10]	mosi2	0x0	RW	mosi2 是否使能。 0 = 使能（spi2 功能有效） 1 = 不使能
[9]	sck2	0x0	RW	sck2 是否使能。 0 = 使能（spi2 功能有效） 1 = 不使能
[8]	ss2	0x0	RW	ss2 是否使能。 0 = 使能（spi2 功能有效） 1 = 不使能
[7]	gint[37:32]	0x0	RW	gint[37:32]是否使能。 0 = 不使能 1 = 使能
[6]	usb DTO/sck_spi4	0x0	RW	usb DTO 和 sck_spi4 否复用。 0 = 不复用（spi4 功能有效） 1 = 复用（usb 测试功能有效）

比特位	名称	复位值	读写属性	功能说明
[5]	usb DRVVBUS/ss_spi4	0x0	RW	usb DRVVBUS 和 ss_spi4 否复用。 0 = 不复用（spi4 功能有效） 1 = 复用（usb 功能有效）
[4]	gint[7:6]/spi5[2],spi5[3]	0x0	RW	gint[7:6]和 spi5[2]，spi5[3]是否复用。 0 = 不复用（gint 功能有效） 1 = 复用（spi5 功能有效）
[3]	ss1,sck1,mosi1,miso1/ss_spi5,sck_spi5,spi5[0],spi5[1]	0x0	RW	ss1，sck1，mosi1，miso1 和 ss_spi5，sck_spi5，spi5[0]，spi5[1]是否复用。 0 = 不复用（spi1 功能有效） 1 = 复用（spi5 功能有效）
[2]	gint[39:38]	0x0	RW	gint[39:38]是否使能。 0 = 不使能 1 = 使能
[1]	gint[15:14]	0x0	RW	gint[15:14]是否使能。 0 = 不使能 1 = 使能
[0]	sck3,miso3,mosi3/gint[11:9]	0x0	RW	sck3，miso3，mosi3 和 gint[11:9]是否复用。 0 = 不复用（spi3 功能有效） 1 = 复用（gint[11:9]功能有效）

19.4.2.8 SPIM1 管脚控制寄存器

SPIM1 管脚控制寄存器的功能包括控制 SSI1 管脚的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0020~0x0023　　复位值：0x3F3F0015

31	30	29	28	27	26	25	24
保留		D3_IE	D2_IE	D1_IE	D0_IE	SCK_IE	SS_IE
ro		rw	rw	rw	Rw	rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		D3_PS	D2_PS	D1_PS	D0_PS	SCK_PS	SS_PS
ro		rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留		ODE	PUE	IS	SR	DS[1:0]	
ro		rw	rw	rw	rw	rw	

图表 19-9：SPIM1 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:30]	保留	0x0	RO	----
[29]	D3_IE	0x1	RW	SPIM1 管脚 D3 的输入功能是否开启。 0 = D3 输入功能被禁止 1 = D3 输入功能开启
[28]	D2_IE	0x1	RW	SPIM1 管脚 D2 的输入功能是否开启。 0 = D2 输入功能被禁止 1 = D2 输入功能开启
[27]	D1_IE	0x1	RW	SPIM1 管脚 D1 的输入功能是否开启。 0 = D1 输入功能被禁止 1 = D1 输入功能开启
[26]	D0_IE	0x1	RW	SPIM1 管脚 D0 的输入功能是否开启。 0 = D0 输入功能被禁止 1 = D0 输入功能开启
[25]	SCK_IE	0x1	RW	SPIM2 管脚 SCK 的输入功能是否开启。 0 = SCK 输入功能被禁止 1 = SCK 输入功能开启
[24]	SS_IE	0x1	RW	SPIM1 管脚 SS 的输入功能是否开启。 0 = SS 输入功能被禁止 1 = SS 输入功能开启

比特位	名称	复位值	读写属性	功能说明
[21]	D3_PS	0x1	RW	SPIM1 管脚 D3 的输入上下拉选择。 0 = D3 选择输入下拉 1 = D3 选择输入上拉
[20]	D2_PS	0x1	RW	SPIM1 管脚 D2 的输入上下拉选择。 0 = D2 选择输入下拉 1 = D2 选择输入上拉
[19]	D1_PS	0x1	RW	SPIM1 管脚 D1 的输入上下拉选择。 0 = D1 选择输入下拉 1 = D1 选择输入上拉
[18]	D0_PS	0x1	RW	SPIM1 管脚 D0 的输入上下拉选择。 0 = D0 选择输入下拉 1 = D0 选择输入上拉
[17]	SCK_PS	0x1	RW	SPIM1 管脚 SCK 的输入上下拉选择。 0 = SCK 选择输入下拉 1 = SCK 选择输入上拉
[16]	SS_PS	0x1	RW	SPIM1 管脚 SS 的输入上下拉选择。 0 = SS 选择输入下拉 1 = SS 选择输入上拉
[15:6]	保留	0x0	RO	----
[5]	ODE	0x1	RW	开漏(Open-Drain)输出是否使能。 0 = CMOS 输出使能 1 = Open-drain 输出使能
[4]	PUE	0x0	RO	上下拉是否使能 0 = 不使能 1 = 使能
[3]	IS	0x0	RW	SPIM2 管脚的输入类型。 0 = SPIM1 管脚为 CMOS 输入 1 = SPIM1 管脚为施密特触发器输入
[2]	SR	0x1	RW	SPIM2 管脚电压转换率。 0 = SPIM1 管脚电压转换率为高速状态 1 = SPIM1 管脚电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	SPIM1 管脚驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.9 SPIM2 管脚控制寄存器

SPIM2 管脚控制寄存器的功能包括控制 SSI2 管脚的输入使能，上拉使能，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0024~0x0027　　复位值：0x3F3F0015

31	30	29	28	27	26	25	24
保留		D3_IE	D2_IE	D1_IE	D0_IE	SCK_IE	SS_IE
ro		rw	rw	rw	Rw	rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		D3_PS	D2_PS	D1_PS	D0_PS	SCK_PS	SS_PS
ro		rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留		ODE	PUE	IS	SR	DS[1:0]	
ro		rw	rw	rw	rw	rw	

图表 19-10：SPIM2 管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:30]	保留	0x0	RO	----
[29]	D3_IE	0x1	RW	SPIM2 管脚 D3 的输入功能是否开启。 0 = D3 输入功能被禁止 1 = D3 输入功能开启
[28]	D2_IE	0x1	RW	SPIM2 管脚 D2 的输入功能是否开启。 0 = D2 输入功能被禁止 1 = D2 输入功能开启
[27]	D1_IE	0x1	RW	SPIM2 管脚 D1 的输入功能是否开启。 0 = D1 输入功能被禁止 1 = D1 输入功能开启
[26]	D0_IE	0x1	RW	SPIM2 管脚 D0 的输入功能是否开启。 0 = D0 输入功能被禁止 1 = D0 输入功能开启
[25]	SCK_IE	0x1	RW	SPIM2 管脚 SCK 的输入功能是否开启。 0 = SCK 输入功能被禁止 1 = SCK 输入功能开启
[24]	SS_IE	0x1	RW	SPIM2 管脚 SS 的输入功能是否开启。 0 = SS 输入功能被禁止 1 = SS 输入功能开启

比特位	名称	复位值	读写属性	功能说明
[21]	D3_PS	0x1	RW	SPIM2 管脚 D3 的输入上下拉选择。 0 = D3 选择输入下拉 1 = D3 选择输入上拉
[20]	D2_PS	0x1	RW	SPIM2 管脚 D2 的输入上下拉选择。 0 = D2 选择输入下拉 1 = D2 选择输入上拉
[19]	D1_PS	0x1	RW	SPIM2 管脚 D1 的输入上下拉选择。 0 = D1 选择输入下拉 1 = D1 选择输入上拉
[18]	D0_PS	0x1	RW	SPIM2 管脚 D0 的输入上下拉选择。 0 = D0 选择输入下拉 1 = D0 选择输入上拉
[17]	SCK_PS	0x1	RW	SPIM2 管脚 SCK 的输入上下拉选择。 0 = SCK 选择输入下拉 1 = SCK 选择输入上拉
[16]	SS_PS	0x1	RW	SPIM2 管脚 SS 的输入上下拉选择。 0 = SS 选择输入下拉 1 = SS 选择输入上拉
[15:6]	保留	0x0	RO	----
[5]	ODE	0x1	RW	开漏(Open-Drain)输出是否使能。 0 = CMOS 输出使能 1 = Open-drain 输出使能
[4]	PUE	0x0	RO	上下拉是否使能 0 = 不使能 1 = 使能
[3]	IS	0x0	RW	SPIM2 管脚的输入类型。 0 = SPIM2 管脚为 CMOS 输入 1 = SPIM2 管脚为施密特触发器输入
[2]	SR	0x1	RW	SPIM2 管脚电压转换率。 0 = SPIM2 管脚电压转换率为高速状态 1 = SPIM2 管脚电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	SPIM2 管脚驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.10 GINT[31:30]管脚控制寄存器

GINT[31:30]如下：

偏移地址：0x002C~0x002F 复位值：0x3F3F0012

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro		rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-11：GINT[31:30]管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0xF3F3001	RO	----
[3]	IS	0x0	RW	GINT[31:30]管脚的输入类型。 0 = GINT[31:30]管脚为 CMOS 输入 1 = GINT[31:30]管脚为施密特触发器输入
[2]	SR	0x0	RW	GINT[31:30]管脚电压转换率。 0 = GINT[31:30]管脚电压转换率为高速状态 1 = GINT[31:30]管脚电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	GINT[31:30]管脚驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.11 GINT[39:32]管脚控制寄存器

GINT[39:32]管脚控制寄存器的功能包括控制 GINT[39:32]管脚的输入类型，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x0038~0x003B　　复位值：0xFFFF0312

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro		rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-12：GINT[39:32]管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0xFFFF0 31	RO	----
[3]	IS	0x0	RW	GINT[39:32]管脚的输入类型。 0 = GINT[39:32]管脚为 CMOS 输入 1 = GINT[39:32]管脚为施密特触发器输入
[2]	SR	0x0	RW	GINT[39:32]管脚电压转换率。 0 = GINT[39:32]管脚电压转换率为高速状态 1 = GINT[39:32]管脚电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	GINT[39:32]管脚驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.12 GINT[29:22]管脚控制寄存器

GINT[29:22]管脚控制寄存器的功能包括控制 GINT[29:22]管脚的上下拉选择，输入类型，电压转换速率，驱动能力等。具体控制信息如下：

偏移地址：0x003C~0x003F　　　　复位值：0xFFFFFF01

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro		rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8
PS[7:0]							
rw							

7	6	5	4	3	2	1	0
保留				IS	SR	DS[1:0]	
ro				rw	rw	rw	

图表 19-13：GINT[29:22]管脚控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0xFFFF	RO	----
[15:8]	PS[7:0]	0xFF	RW	分别对应 GINT[29:22]管脚的输入上下拉选择。 0 = 选择输入下拉 1 = 选择输入上拉
[7:4]	保留	0x0	RO	----
[3]	IS	0x0	RW	GINT[29:22]管脚的输入类型。 0 = GINT[29:22]管脚为 CMOS 输入 1 = GINT[29:22]管脚为施密特触发器输入
[2]	SR	0x0	RW	GINT[29:22]管脚电压转换率。 0 = GINT[29:22]管脚电压转换率为高速状态 1 = GINT[29:22]管脚电压转换率为低速状态
[1:0]	DS[1:0]	0x1	RW	GINT[29:22]管脚驱动能力。 00 = 2mA 01 = 8mA 10 = 4mA 11 = 12mA

19.4.2.13 CLKOUT/RSTOUT 管脚功能控制寄存器

CLKOUT/RSTOUT 管脚功能控制寄存器，控制 CLKOUT/RSTOUT 管脚的复用关系。

偏移地址: 0x0044~0x0047　　复位值：0x3F070501

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

图表 19-14：CLKOUT/RSTOUT 管脚功能控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	SWAP[3]	0x0	RW	RSTOUT 是否复用作 GPIO 0 = 不复用 1 = 复用
[30]	SWAP[2]	0x0	RW	CLKOUT 是否复用作 GPIO 0 = 不复用 1 = 复用
[29]	SWAP[1]	0x1	RW	RSTOUT 管脚是否复用 0 = 不复用 1 = 复用
[28]	SWAP[0]	0x1	RW	CLKOUT 管脚是否复用 0 = 不复用 1 = 复用
[27:0]	保留	0x0	RO	----

19.5 功能描述

19.5.1 配置管脚上下拉状态

管脚设置为输入状态且浮空时，配置 PS 位为高，可以使管脚输入稳定的高电平，同样，配置 PS 位为低，可以使管脚输入稳定的低电平，从而减少了漏电。

19.5.2 配置管脚驱动能力

当管脚为输出时，通过配置 DS0 和 DS1，调整管脚的驱动电流，可以增强或减弱管脚的驱动能力。

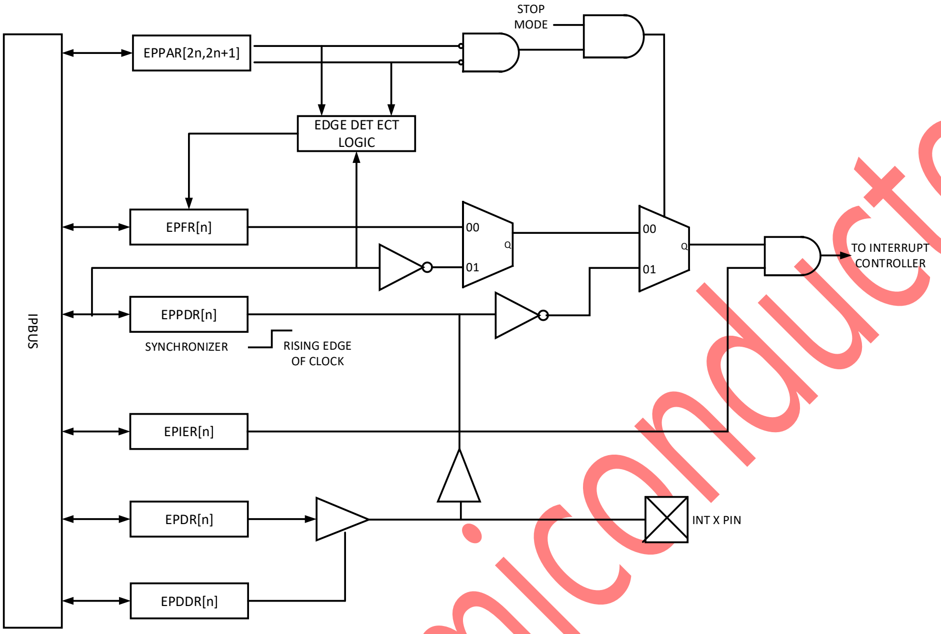
19.5.3 配置管脚的复用功能

通过配置管脚的复用控制位，可以根据需求动态改变管脚的功能，提高芯片使用的灵活性。

20 边沿端口模块（EPORT）

20.1 概述

边沿端口模块（EPORT）有八个外部中断管脚（单个模块）。每个管脚都是能独立配置为支持电平（高电平或低电平）敏感或边沿检测（上升沿，下降沿或两者）的中断管脚，或者配置为通用输入/输出（GPIO）管脚。



图表 20-1：EPORT 模块框图

20.2 低功耗模式

本章节描述了 EPORT 模块在不同的低功耗模式下的工作情况。

20.2.1 等待和瞌睡模式

在等待和瞌睡模式下，ERPOT 模块继续正常运行。并且可配置支持外部电平敏感或边缘检测功能。接收外部信号并产生中断，可使系统退出低功耗模式。

20.2.2 停止和睡眠模式

在停止模式下，没有时钟可用以支持运行边沿检测功能。只有电平检测逻辑是激活的（如果被配置的话）。这时可以通过外部中断管脚上的电平去产生一个中断（如果中断使能的话），从而退出停止模式。

注意：由于停止模式下时钟停止，所以电平逻辑检测时，输入管脚的同步逻辑会被旁路掉。

20.3 功能描述

在复位时，所有的管脚默认配置为通用输入管脚。从 EPORT 管脚数据寄存器（EPPDR）读取的管脚电压值是和 CLKOUT 的上升沿同步的。用于边沿/电平检测的电压值也是和 CLKOUT 的上升沿同步的。并且 EPORT 管脚采用施密特触发的输入缓冲器，当外部信号出现缓慢的上升或下降沿时，内置的迟滞触发设计可以降低产生虚假的边沿触发中断的可能性。

20.4 内存映射和寄存器

20.4.1 内存映射

本章节对内存映射和寄存器结构作了详细描述介绍。

管脚配置寄存器（EPPAR）—控制每个管脚的功能
数据方向寄存器（EPDDR）—控制每个管脚的方向
中断使能寄存器（EPIER）—使能每一个管脚的中断请求
数据寄存器（EPDR）—保存向管脚驱动的数据
管脚数据寄存器（EPPDR）—反映当前管脚的状态
标志寄存器（EPFR）—锁存 EPORT 触发事件
管脚上拉使能寄存器（EPPUE）—分别控制管脚的上拉使能
电平极性寄存器（EPLPR）—在电平敏感状态下，用以控制管脚的电平极性
开漏(Open-Drain)使能寄存器（EPODE）—控制每个管脚的开漏输出使能

表格 20-1：控制器偏移地址映像

偏移地址	位 15-8	位 7-0	访问权限
0x0000	管脚配置寄存器(EPPAR)		S
0x0002	数据方向寄存器(EPDDR)	中断使能寄存器(EPIER)	S
0x0004	数据寄存器(EPDR)	管脚数据寄存器(EPPDR)	S/U
0x0006	标志寄存器(EPFR)	管脚上拉使能寄存器(EPPUE)	S/U
0x0008	电平极性寄存器(EPLPR)	开漏使能寄存器(EPODE)	S

20.4.2 寄存器描述

20.4.2.1 管脚配置寄存器

偏移地址：0x0000~0x0001　　复位值：0x0000

15	14	13	12	11	10	9	8
EPPA7		EPPA6		EPPA5		EPPA4	
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPPA3		EPPA2		EPPA1		EPPA0	
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-2：管脚配置寄存器（EPPAR）

比特位	名称	复位值	读写属性	功能说明
[15:14]	EPPA7	0x0	RW	指定 EPORT 管脚为电平检测或上升沿/下降沿检测 00 = 管脚电平敏感 01 = 管脚上升沿触发 10 = 管脚下降沿触发 11 = 管脚上升沿触发和下降沿触发 **注意：** EPORT 模块的中断请求可以被中断控制模块屏蔽。
[13:12]	EPPA6	0x0	RW	
[11:10]	EPPA5	0x0	RW	
[9:8]	EPPA4	0x0	RW	
[7:6]	EPPA3	0x0	RW	
[5:4]	EPPA2	0x0	RW	
[3:2]	EPPA1	0x0	RW	
[1:0]	EPPA0	0x0	RW	

20.4.2.2 中断使能寄存器

偏移地址：0x0002　　复位值：0x00

7	6	5	4	3	2	1	0
EPIE7	EPIE6	EPIE5	EPIE4	EPIE3	EPIE2	EPIE1	EPIE0
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-3：中断使能寄存器（EPIER）

比特位	名称	复位值	读写属性	功能说明
[7]	EPIE7	0x0	RW	使能 EPORT 的中断请求 0 = 使能管脚中断请求 1 = 禁用管脚中断请求
[6]	EPIE6	0x0	RW	
[5]	EPIE5	0x0	RW	
[4]	EPIE4	0x0	RW	
[3]	EPIE3	0x0	RW	
[2]	EPIE2	0x0	RW	
[1]	EPIE1	0x0	RW	

[0]	EPIE0	0x0	RW	

20.4.2.3 数据方向寄存器

偏移地址：0x0003 复位值：0x00

7	6	5	4	3	2	1	0
EPDD7	EPDD6	EPDD5	EPDD4	EPDD3	EPDD2	EPDD1	EPDD0
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-4：数据方向寄存器（EPDDR）

比特位	名称	复位值	读写属性	功能说明
[7]	EPDD7	0x0	RW	配置 EPORT 的管脚输入输出功能 0x0 = 配置管脚作为输入 0x1 = 配置管脚作为输出 **注意：** 当使用管脚作为外部中断请求源时，EPDDR 内对应的位必须清除。
[6]	EPDD6	0x0	RW	
[5]	EPDD5	0x0	RW	
[4]	EPDD4	0x0	RW	
[3]	EPDD3	0x0	RW	
[2]	EPDD2	0x0	RW	
[1]	EPDD1	0x0	RW	
[0]	EPDD0	0x0	RW	

20.4.2.4 管脚数据寄存器

偏移地址：0x0004 复位值：当前管脚状态

7	6	5	4	3	2	1	0
EPPD7	EPPD6	EPPD5	EPPD4	EPPD3	EPPD2	EPPD1	EPPD0
ro	ro	ro	ro	ro	ro	ro	ro

图表 20-5：数据寄存器（EPPDR）

比特位	名称	复位值	读写属性	功能说明
[7]	EPPD7	P	RO	指示当前管脚状态。 **注意：** P 表示当前管脚状态
[6]	EPPD6	P	RO	
[5]	EPPD5	P	RO	
[4]	EPPD4	P	RO	
[3]	EPPD3	P	RO	
[2]	EPPD2	P	RO	
[1]	EPPD1	P	RO	
[0]	EPPD0	P	RO	

20.4.2.5 数据寄存器

偏移地址：0x0005 复位值：0xFF

7	6	5	4	3	2	1	0
EPD7	EPD6	EPD5	EPD4	EPD3	EPD2	EPD1	EPD0
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-6：数据寄存器（EPDR）

比特位	名称	复位值	读写属性	功能说明
[7]	EPPD7	0x1	RW	当端口的任何一个管脚被作为输出的话，写入的位就会被驱动到管脚上去。 0 = 配置管脚低电平输出 1 = 配置管脚高电平输出
[6]	EPPD6	0x1	RW	
[5]	EPPD5	0x1	RW	
[4]	EPPD4	0x1	RW	
[3]	EPPD3	0x1	RW	
[2]	EPPD2	0x1	RW	
[1]	EPPD1	0x1	RW	
[0]	EPPD0	0x1	RW	

20.4.2.6 管脚上拉使能寄存器

偏移地址：0x0006 复位值：0xFF

7	6	5	4	3	2	1	0
EPPUE7	EPPUE6	EPPUE5	EPPUE4	EPPUE3	EPPUE2	EPPUE1	EPPUE0
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-7：管脚上拉使能寄存器（EPPUE）

比特位	名称	复位值	读写属性	功能说明
[7]	EPPUE7	0x1	RW	配置管脚上下拉功能。 0 = 配置管脚下拉功能 1 = 使能管脚上拉功能
[6]	EPPUE6	0x1	RW	
[5]	EPPUE5	0x1	RW	
[4]	EPPUE4	0x1	RW	
[3]	EPPUE3	0x1	RW	
[2]	EPPUE2	0x1	RW	
[1]	EPPUE1	0x1	RW	
[0]	EPPUE0	0x1	RW	

20.4.2.7 标志寄存器

偏移地址：0x0007 复位值：0x00

7	6	5	4	3	2	1	0
EPFR7	EPFR6	EPFR5	EPFR4	EPFR3	EPFR2	EPFR1	EPFR0
ro	ro	ro	ro	ro	ro	ro	ro

图表 20-8：标志寄存器（EPFR）

比特位	名称	复位值	读写属性	功能说明
[8]	EPFR7	0x0	RO	指示管脚边缘检测状态 0 = 选择边沿未被检测到 1 = 选择边沿已经被检测到 **注意：**当配置为边沿触发时，此寄存器指示边沿触发状态。
[7]	EPFR6	0x0	RO	
[6]	EPFR5	0x0	RO	
[5]	EPFR4	0x0	RO	
[4]	EPFR3	0x0	RO	
[3]	EPFR2	0x0	RO	
[2]	EPFR1	0x0	RO	
[1]	EPFR0	0x0	RO	

20.4.2.8 开漏使能寄存器

偏移地址：0x0008 复位值：0x00

7	6	5	4	3	2	1	0
EPODE7	EPODE6	EPODE5	EPODE4	EPODE3	EPODE2	EPODE1	EPODE0
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-9：开漏使能寄存器（EPODE）

比特位	名称	复位值	读写属性	功能说明
[8]	EPODE7	0x0	RW	配置管脚输出 0 = CMOS 输出 1 = 开漏(Open-Drain)输出
[7]	EPODE6	0x0	RW	
[6]	EPODE5	0x0	RW	
[5]	EPODE4	0x0	RW	
[4]	EPODE3	0x0	RW	
[3]	EPODE2	0x0	RW	
[2]	EPODE1	0x0	RW	
[1]	EPODE0	0x0	RW	

20.4.2.9 电平极性寄存器

偏移地址：0x0009　　　　复位值：0x00

7	6	5	4	3	2	1	0
EPLPR7	EPLPR6	EPLPR5	EPLPR4	EPLPR3	EPLPR2	EPLPR1	EPLPR0
rw	rw	rw	rw	rw	rw	rw	rw

图表 20-10：电平极性寄存器（EPLPR）

比特位	名称	复位值	读写属性	功能说明
[8]	EPODE 7	0x0	RW	配置管脚为高电平敏感或低电平敏感。 0 = 低电平敏感 1 = 高电平敏感
[7]	EPODE 6	0x0	RW	
[6]	EPODE 5	0x0	RW	
[5]	EPODE 4	0x0	RW	
[4]	EPODE 3	0x0	RW	
[3]	EPODE 2	0x0	RW	
[2]	EPODE 1	0x0	RW	
[1]	EPODE 0	0x0	RW	

20.5 中断描述

下表列出了和 EPORT 模块有关的中断请求。

表格 20-2：EPORT 中断请求源

中断源	标志	使能位
端口边沿检测	EPFR[7:0]	EPIE[7:0]

20.5.1 端口边沿检测（EPFR）中断

可读/写的 EPIE［7:0］位使能 EPORT 的中断请求。当 EPORT 标志寄存器（EPFR）内对应的位被置位或后来被置位或对应的管脚是低电平(或高电平)并且管脚配置为低电平敏感（或高电平敏感）时，将产生中断。复位清 0。

21 EDMAC 控制器

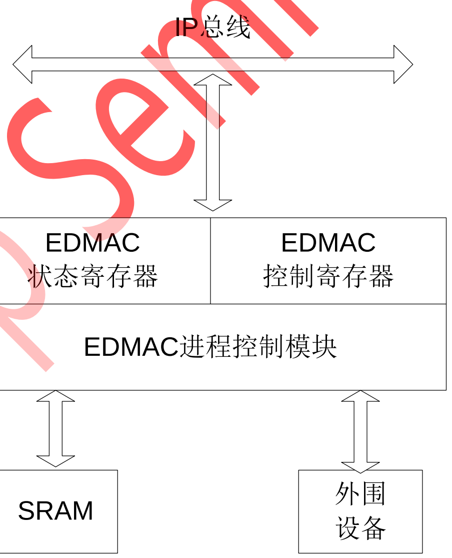
21.1 概述

EDMAC 控制器用于在 RAM 和外围设备之间传输数据。

21.2 特性

EDMAC 控制器特性包括：
- 支持双通道
- 可编程传输数据数量
- 可编程读缓存地址和写缓存地址
- 多外设选择
- 支持读、写、写后读传输

21.3 框图



图表 21-1：EDMAC 框图

21.4 工作模式

EDMAC 可以把数据从 RAM 传到外围设备，也可以把数据从外围设备传到 RAM。外设可以为 SPI1，SPI2，SPI3，USI2，USI3，AES，DES，SCI1，SHA。EDMAC 读写缓存和系统 SRAM 共享。

21.5 内存映射和寄存器

21.5.1 内存映射

表格 21-1：EDMAC 内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	通道 0 控制寄存器（EDMACCR0）		S/U
0x0004	通道 0 状态寄存器 0（EDMACSR0）		S/U
0x0008	信道 0 读缓存地址寄存器（EDMACRBAR0）		S/U
0x000C	信道 0 写缓存地址寄存器（EDMACWBAR0）		S/U
0x0010	通道 0 次要传输总和寄存器（EDMACMINSUMR0）		S/U
0x0014	通道 0 次要传输计数寄存器（EDMACMINCNTR0）		S/U
0x0018	通道 0 主要传输总和寄存器（EDMACMAJCNTR0）		S/U
0x001C	通道 0 主要传输计数寄存器（EDMACMAJCNTR0）		S/U
0x0020	信道 0 特殊外设地址寄存器（EDMACSPAR0）		S/U
0x0024	信道 0 读缓存地址阶跃寄存器（EDMACRBARSTEP0）		S/U
0x0028	信道 0 写缓存地址阶跃寄存器（EDMACWBARSTEP0）		S/U
0x002C	通道 0 最后次要总数寄存器（EDMACLASTMINSUMR0）		S/U
0x0030-0x003F	保留		---
0x0040	通道 1 控制寄存器（EDMACCR1）		S/U
0x0044	通道 1 状态寄存器 1（EDMACSR1）		S/U
0x0048	信道 1 读缓存地址寄存器（EDMACRBAR1）		S/U
0x004C	信道 1 写缓存地址寄存器（EDMACWBAR1）		S/U
0x0050	通道 1 次要传输总和寄存器（EDMACMINSUMR1）		S/U
0x0054	通道 1 次要传输计数寄存器（EDMACMINCNTR1）		S/U
0x0058	通道 1 主要传输总和寄存器（EDMACMAJCNTR1）		S/U
0x005C	通道 1 主要传输计数寄存器（EDMACMAJCNTR1）		S/U
0x0060	信道 1 特殊外设地址寄存器（EDMACSPAR1）		S/U
0x0064	信道 1 读缓存地址阶跃寄存器（EDMACRBARSTEP1）		S/U
0x0068	信道 1 写缓存地址阶跃寄存器（EDMACWBARSTEP1）		S/U
0x006C	通道 1 最后次要总数寄存器（EDMACLASTMINSUMR1）		S/U

21.5.2 寄存器描述

21.5.2.1 通道控制寄存器（EDMACCR0/EDMACCR1）

偏移地址：0x0000~0x0003　　　　　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
PIPELINE	SPI_HW	PRIOR_CFG	PRIOR	SPI_WROPT	INFINITY	SPI_FIFO_PRELOAD[3]	MAJ_DONEIE
rw	rw	rw	rw	rw	rw	rw	rw
23	22	21	20	19	18	17	16
MAJ_CRC_SEL	CRC_CHEN	SEND_CRC_CHEN	WRITE_CRC_CHEN	LINK_SRC_SEL[2:0]			LINK
rw	rw	rw	rw	rw			rw
15	14	13	12	11	10	9	8
COMPARE_EN	COMPARE_SKIP	PRELOAD_DIS	MEM-TRANSCRC	PN[3:0]			
rw	rw	rw	rw	rw			
7	6	5	4	3	2	1	0
STARTIE	TTYPE[1:0]		SPI_FIFO_PRELOAD[2:0]			VALID	MIN_DONEIE
rw	rw		rw			rw	rw

图表 21-2：通道 0 控制寄存器

偏移地址：0x0040~0x0043　　　　　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
PIPELINE	SPI_HW	PRIOR_CFG	PRIOR	SPI_WROPT	INFINITY	SPI_FIFO_PRELOAD[3]	MAJ_DONEIE
rw	rw	rw	rw	rw	rw	rw	rw
23	22	21	20	19	18	17	16
MAJ_CRC_SEL	CRC_CHEN	SEND_CRC_CHEN	WRITE_CRC_CHEN	LINK_SRC_SEL[2:0]			LINK
rw	rw	rw	rw	rw			rw
15	14	13	12	11	10	9	8
COMPARE_EN	COMPARE_SKIP	PRELOAD_DIS	MEM-TRANSCRC	PN[3:0]			
rw	rw	rw	rw	rw			
7	6	5	4	3	2	1	0
STARTIE	TTYPE[1:0]		SPI_FIFO_PRELOAD[2:0]			VALID	MIN_DONEIE
rw	rw		rw			rw	rw

图表 21-3：通道 1 控制寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	PIPELINE	0x0	RW	管道传输模式 0 = 禁止管道传输模式；1 = 使能管道传输模式
[30]	SPI_HW	0x0	RW	SPI 半字操作 在 SPI 半字操作模式下，请将 SPITXFCR 中的 TXFSTH 配置为 6，然后 SPIRXFCR 中的 RXFSTH 配置为 1。有关详细信息，请参见 SPI 部分。 0 = SPI 字节操作；1 = SPI 半字操作
[29]	PRIOR_CFG	0x0	RW	使能优先级更改 0 = 禁止；1 = 使能
[28]	PRIOR	0x0	RW	通道优先级 在配置该位之前配置 PRIOR_CHG。 0 = 通道 0 优先；1 = 通道 1 优先
[27]	SPI_WROPT	0x0	RW	在管道模式下，当 TTYPE 为 2’b11 时，SPI 写优化 0 = 禁止写 SPI 时序优化 1 = 使能写 SPI 时序优化
[26]	INFINITY	0x0	RW	通道无限传输 0 = 有限；1 = 无限 将 INFINITY 置 1 时，MINOR_SUM 保持有效，而 MAJOR_SUM 无效，MAJOR_SUM 不限制数据传输。

比特位	名称	复位值	读写属性	功能说明
[25]	SPI_FIFO_PRELOAD[3]	0x0	RW	将数据预加载到 spi 写 fifo 该字段决定要向 spi 发 FIFO 预发送多少数据。edmac 选择 SPI1/2/3，传输类型为 2’b11 或 2’b10 时，此字段有效。当 SPI_HW 使能，请将 SPI_FIFO_PRELOAD 配置为偶数 SPI_FIFO_PRELOAD[2:0] 位于寄存器比特位 [4:2] SPI_FIFO_PRELOAD[3:0]: 4’b0000：无预加载 4’b0001：1 4’b0010：2 4’b0011：3 4’b0100：4 4’b0101：5 4’b0110：6 4’b0111：7 4’b1000：8 4’b1001：9 4’b1010：10
[24]	MAJ_DONEIE	0x0	RW	主要传输 DONE 中断使能位,使主要传输 DONE 标志产生一个中断请求。 0 = 禁止主要传输 DONE 中断请求 1 = 使能主要传输 DONE 中断请求
[23]	MAJ_CRC_SEL	0x0	RW	发送或写入 crc 码操作时序选择 0 = 次要传输结束时进行操作 1 = 主要传输结束时进行操作
[22]	CRC_CHEN	0x0	RW	CRC 通道使能 0 = edmac 操作禁止 CRC 1 = edmac 操作使能 CRC
[21]	SEND_CRC_CHEN	0x0	RW	使能将 crc 码发送到外围 IP 通道 0 = 不将 crc 码发送到外围 IP 1 = 将 crc 码发送到外围 IP
[20]	WRITE_CRC_CHEN	0x0	RW	使能将 crc 码写入 sram 通道 0 = 不将收到的 crc 码写入 sram 1 = 将收到的 crc 码写入 sram

比特位	名称	复位值	读写属性	功能说明
[19:17]	LINK_SRC_SEL	0x0	RW	链接源选择 LINK_SRC_SEL[2:0]对应的硬件事件： 3' b000：EDMAC0 通道 0 完成 3' b001：EDMAC0 通道 1 完成 3' b010：EDMAC1 通道 0 完成 3' b011：EDMAC1 通道 1 完成 3' b100：保留 3' b101：保留 3' b110：保留 3' b111：保留
[16]	LINK	0x0	RW	使能链接功能。 启用 LINK 功能时，可以通过 LINK_SRC_SEL[2:0]选择的硬件事件来声明通道的 VALID 位。 0 = 禁止链接功能 1 = 使能链接功能
[15]	COMPARE_EN	0x0	RW	数据比较功能使能位（仅用于工厂测试） 0 = 禁止数据比较功能 1 = 使能数据比较功能
[14]	COMPARE_SKIP	0x0	RW	跳过数据比较功能（仅用于工厂测试） 0 = 数据比较失败后设置 fail 位，继续比较操作直到数据全部比较完成 1 = 一旦数据比较失败，跳出比较操作，设置 done 位和 fail 位
[13]	PRELOAD_DIS	0x0	RW	禁止预加载数据到内部缓存功能（仅用于工厂测试） 0 = 正常模式下使能预加载功能，此位为 0 1 = 当需要用 LBA（仅用于工厂测试）产生随机数据时，需设置此位为 1
[12]	MEM-TRANS CRC	0x0	RW	存储间传输 CRC 校验 0 = 传输类型为存储间传输时不进行 CRC 校验 1 = 传输类型为存储间传输时进行 CRC 校验

比特位	名称	复位值	读写属性	功能说明
[11:8]	PN	0x0	RW	通道外围设备选择，控制当前通道选择哪一个外围设备 4’b0000：EDMAC1 SPI1 4’b0001：EDMAC1 SPI2 4’b0010：EDMAC1 USI1 4’b0011：EDMAC1 USI2 4’b0100：保留 4’b0101：EDMAC0 AES 4’b0110：EDMAC0 DES 4’b0111：保留 4’b1000：EDMAC1 SCI1 4’b1001：保留 4’b1010：EDMAC0 SHA 4’b1011：EDMAC1 SCI3 4’b1100：保留 4’b1101：EDMAC1 SCI2 4’b1110：EDMAC1 SPI3 4’b1111：保留
[7]	STARTIE	0x0	RW	START 中断使能位，使 START 标志产生一个中断请求。 0 = 禁止 START 中断请求 1 = 使能 START 中断请求
[6:5]	TTYPE	0x0	RW	通道传输类型选择位 TTYPE[1:0]: 2’b00：从 RAM 到 RAM 2’b01：从外围设备到 RAM 2’b10：从 RAM 到外围设备 2’b11：从 RAM 到外围设备然后从外围设备到 RAM
[4:2]	SPI_FIFO_PRELOAD[2:0]	0x0	RW	将数据预加载到 spi 写 fifo 详见比特位[25]
[1]	VALID	0x0	RW	信道有效位，控制信道可用，当传输开始时，此位清除为 0 0 = 当前通道不可用；1 = 当前通道可用
[0]	MIN_DONEIE	0x0	RW	次要传输 DONE 中断使能位，使主要传输 DONE 标志产生一个中断请求 0 = 禁止次要传输 DONE 中断请求 1 = 使能次要传输 DONE 中断请求

21.5.2.2 EDMAC 状态寄存器（EDMACSR0/ EDMACSR1）

偏移地址：0x0004~0x0007 复位值：0x00000000

31	30	29	28	27	26	25	24
SSF_SPI1	SSF_SPI2	SSF_SPI3	SSF_SPI4	保留			保留
r/w1c	r/w1c	r/w1c	r/w1c	ro			rw
23	22	21	20	19	18	17	16
保留							EDMACEN
ro							rw
15	14	13	12	11	10	9	8
FAIL	保留						
ro	ro						
7	6	5	4	3	2	1	0
SCHNUM	DCHNUM	保留		MAJ_DONE	START	MIN_DONE	BUSY
ro	ro	ro		r/w1c	ro	r/w1c	ro

图表 21-4：通道 0 状态寄存器

偏移地址：0x0004~0x0007 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	22	21	20	19	18	17	16
保留							EDMACEN
ro							rw
15	14	13	12	11	10	9	8
FAIL	保留						
ro	ro						
7	6	5	4	3	2	1	0
SCHNUM	DCHNUM	保留		MAJ_DONE	START	MIN_DONE	BUSY
ro	ro	ro		r/w1c	ro	r/w1c	ro

图表 21-5：通道 1 状态寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	SSF_SPI1	0x0	R/W1C	SPI1 SS 标志位
[30]	SSF_SPI2	0x0	R/W1C	SPI2 SS 标志位
[29]	SSF_SPI3	0x0	R/W1C	SPI3 SS 标志位
[28]	SSF_SPI4	0x0	R/W1C	SPI4 SS 标志位
[27:25]	保留	0x0	RO	---
[24]	保留	0x0	RO	---
[23:17]	保留	0x0	RO	---
[16]	EDMACEN	0x0	RW	EDMAC 使能位 0 = 禁止 EDMAC；1 = 使能 EDMAC 当 COMPARE_SKIP = 1 时，可以自动清除 EDMACEN，然后发生比较失败事件。 使用 EDMAC 控制外设传输时，先禁止 EDMAC，然后再使能 EDMAC。在使能 EDMAC 之前，应首先复位外设（例如，SPI 清除 SPE 然后再次启用它）。
[15]	FAIL	0x0	RO	通道比较数据失败 通道比较数据失败时置位。向 DONE 写 1 清除此位。
[14:8]	保留	0x0	RO	---
[7]	SCHNUM	0x0	RO	上次开始传输的通道号 0 = 上次开始传输的是通道 0 1 = 上次开始传输的是通道 1
[6]	DCHNUM	0x0	RO	上次完成传输的通道号 0 = 上次完成传输的是通道 0 1 = 上次完成传输的是通道 1
[5:4]	保留	0x0	RO	---
[3]	MAJ_DONE	0x0	R/W1C	通道的主要传输已完成 MAJ_DONE 在通道完成主要传输块时置起。如果在 EDMACCR 中设置了 MAJ_DONEIE，则 MAJ_DONE 将产生中断。写入 1 会清除此位。 0 = 通道未完成传输块 1 = 通道完成一个传输块
[2]	START	0x0	RO	通道开始传输标志位 表示通道开始传输数据，如果 STARTIE 为 1 会产生中断请求。写入 1 会清除此位。 0 = 通道未开始传输块 1 = 通道开始一个传输块

比特位	名称	复位值	读写属性	功能说明
[1]	MIN_DONE	0x0	R/W1C	通道的次要传输完成 MIN_DONE 在通道完成一个次要传输的块时置起。如果在 EDMACCR 中设置了 MIN_DONEIE，则 MIN_DONE 将产生中断。写入 1 会清除此位。 0 = 通道未完成次要传输块 1 = 通道完成了次要传输块
[0]	BUSY	0x0	RO	通道正在传输 信道开始传输后此位置 1，信道完成后此位自动清除为 0。 0 = 未处理次要传输块 1 = 正在处理次要传输块

21.5.2.3 信道读缓存地址寄存器（EDMACRBAR0/ EDMACRBAR1）

偏移地址：0x0008~0x000B　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
RNONINC	RDEC	保留					
rw	rw	ro					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		EDMACRBAR[21:16]					
ro		rw					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EDMACRBAR[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EDMACRBAR[7:0]							
rw							

图表 21-6：信道 0 读缓存地址寄存器

偏移地址：0x0048~0x004B　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
RNONINC	RDEC	保留					
rw	rw	ro					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		EDMACRBAR[21:16]					
ro		rw					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EDMACRBAR[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EDMACRBAR[7:0]							
rw							

图表 21-7：信道 1 读缓存地址寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	RNONINC	0x0	RW	禁止读缓存地址增加 0 = 读缓存地址增加功能使能 1 = 读缓存地址增加功能被禁止

比特位	名称	复位值	读写属性	功能说明
[30]	RDEC	0x0	RW	EDMAC 读缓存地址减少 0 = 读缓存地址取决于 RNONINC 1 = 读缓存地址减少功能使能
[29:22]	保留	0x0	RO	---
[21:0]	EDMACRBAR	0x0	RW	设置 EDMAC 读缓存地址 EDMACRBAR 字段选择信道的读缓存地址 当 RNONINC = 1 时，必须将 EDMACRBAR [1:0]设置为 00（字对齐）。 启用 NFRAND 模块时，必须将 EDMACRBAR [1:0]设置为 00（字对齐）。

21.5.2.4 信道写缓存地址寄存器（EDMACWBAR0/ EDMACWBAR1）

偏移地址：0x000C~0x000F　　复位值：0x00000000

31	30	29	28	27	26	25	24
WNONINC	WDEC	保留					
rw	rw	ro					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		EDMACWBAR[21:16]					
ro		rw					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EDMACWBAR[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EDMACWBAR[7:0]							
rw							

图表 21-8：信道 0 写缓存地址寄存器

偏移地址：0x004C~0x004F 复位值：0x00000000

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

图表 21-9：信道 1 写缓存地址寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	WNONINC	0x0	RW	禁止写缓存地址增加 0 = 写缓存地址增加功能使能 1 = 写缓存地址增加功能被禁止
[30]	WDEC	0x0	RW	EDMAC 写缓存地址减少 0 = 写缓存地址取决于 WNONINC 1 = 写缓存地址减少功能使能
[29:22]	保留	0x0	RO	---
[21:0]	EDMACWBAR	0x0	RW	设置 EDMAC 写缓存地址 EDMACWBAR 字段选择信道的写缓存地址 当 WNONINC = 1 时，必须将 EDMACWBAR [1:0]设置为 00（字对齐）。 启用 NFRAND 模块时，必须将 EDMACWBAR [1:0]设置为 00（字对齐）。

21.5.2.5 EDMAC 次要传输总和寄存器（EDMACMINSUMR0/ EDMACMINSUMR1）

偏移地址：0x0010~0x0013 复位值：0x00000000

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

图表 21-10：通道 0 次要传输总和寄存器

偏移地址：0x0050~0x0053 复位值：0x00000000

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

图表 21-11：通道 1 次要传输总和寄存器

比特位	名称	复位值	读写属性	功能说明
[31:22]	保留	0x0	RO	---
[21:0]	MINOR_SUM	0x0	RW	通道次要传输数据总和 MINOR_SUM 字段控制在通道的一个次要传输中发送的数据总和（以字节为单位）。

21.5.2.6 EDMAC 次要传输计数寄存器（EDMACMINCNTR0/ EDMACMINCNTR1）

偏移地址：0x0014~0x0017 复位值：0x00000000

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

图表 21-12：通道 0 次要传输计数寄存器

偏移地址：0x0054~0x0057 复位值：0x00000000

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

图表 21-13：通道 1 次要传输计数寄存器

比特位	名称	复位值	读写属性	功能说明
[31:22]	保留	0x0	RO	---
[21:0]	MINOR_CNT	0x0	RW	EDMAC 次要传输计数器 计数器用于指示在次要循环中已写入 edmac 写缓存的总字节数。

21.5.2.7 EDMAC 主要传输总和寄存器（EDMACMAJSUMR0/ EDMACMAJSUMR1）

偏移地址：0x0018~0x001B 复位值：0x00000000

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

图表 21-14：通道 0 主要传输总和寄存器

偏移地址：0x0058~0x005B 复位值：0x00000000

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

图表 21-15：通道 1 主要传输总和寄存器

比特位	名称	复位值	读写属性	功能说明
[31:22]	保留	0x0	RO	---
[21:0]	MAJOR SUM	0x0	RW	通道主要传输数据总和 MAJOR SUM 字段控制在通道的一个主要块传输中发送的次要块数量。

21.5.2.8 EDMAC 主要传输计数寄存器（EDMACMAJCNTR0/ EDMACMAJCNTR1）

偏移地址：0x001C~0x001F 复位值：0x00000000

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

图表 21-16：通道 0 主要传输计数寄存器

偏移地址：0x001C~0x001F 复位值：0x00000000

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

图表 21-17：通道 1 主要传输计数寄存器

比特位	名称	复位值	读写属性	功能说明
[31:22]	保留	0x0	RO	---
[21:0]	MAJOR CNT	0x0	RW	EDMAC 主要传输计数器 计数器用于指示在主循环中完成的次要块数。

21.5.2.9 EDMAC 特殊外设地址寄存器 0/1（EDMACSPAR0/1）

偏移地址：0x0020~0x0023　　　　复位值：0x00000000

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

图表 21-18：信道 0 特殊外设地址寄存器

偏移地址：0x0060~0x0063　　　　复位值：0x00000000

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

图表 21-19：信道 1 特殊外设地址寄存器

比特位	名称	复位值	读写属性	功能说明
[31:16]	保留	0x0	RO	---
[15:0]	EDMACSPAR	0x0	RW	特殊外设地址 当外设有多个数据寄存器地址时，配置该寄存器以区分。

21.5.2.10 信道读缓存地址阶跃寄存器（EDMACRBARSTEP0/EDMACRBARSTEP 1）

偏移地址：0x0024~0x0027 复位值：0x00000000

31	30	29	28	27	26	25	24
RBAR_STEP_EN	RBAR_STEP_DIR	保留					
rw	rw	ro					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		RBAR_STEP[21:16]					
ro		rw					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RBAR_STEP[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBAR_STEP[7:0]							
rw							

图表 21-20：信道 0 读缓存地址阶跃寄存器

偏移地址：0x0064~0x0067 复位值：0x00000000

31	30	29	28	27	26	25	24
RBAR_STEP_EN	RBAR_STEP_DIR	保留					
rw	rw	ro					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		RBAR_STEP[21:16]					
ro		rw					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RBAR_STEP[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBAR_STEP[7:0]							
rw							

图表 21-21：信道 1 读缓存地址阶跃寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	RBAR_STEP_EN	0x0	RW	EDMAC 读缓存地址阶跃使能 0 = 禁止；1 = 使能
[30]	RBAR_STEP_DIR	0x0	RW	EDMAC 读缓存地址阶跃方向 0 = EDMAC 读缓存地址增加 1 = EDMAC 读缓存地址减少
[29:22]	保留	0x0	RO	---
[21:0]	RBAR_STEP	0x0	RW	EDMAC 读缓存阶跃基地址 当次循环开始时（第一个次循环除外），EDMACRBAR 字段将用 RBAR_STEP 更新。

21.5.2.11 信道写缓存地址阶跃寄存器（EDMACWBARSTEP0/EDMACWBARSTEP 1）

偏移地址：0x0028~0x002B 复位值：0x00000000

31	30	29	28	27	26	25	24
WBAR_STEP_EN	WBAR_STEP_DIR	保留					
rw	rw	ro					

23	22	21	20	19	18	17	16
保留		WBAR_STEP[21:16]					
ro		rw					

15	14	13	12	11	10	9	8
WBAR_STEP[15:8]							
rw							

7	6	5	4	3	2	1	0
WBAR_STEP[7:0]							
rw							

图表 21-22：信道 0 写缓存地址阶跃寄存器

偏移地址：0x0068~0x006B 复位值：0x00000000

31	30	29	28	27	26	25	24
WBAR_STEP_EN	WBAR_STEP_DIR	保留					
rw	rw	ro					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留		WBAR_STEP[21:16]					
ro		rw					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WBAR_STEP[15:8]							
rw							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WBAR_STEP[7:0]							
rw							

图表 21-23：信道 1 写缓存地址阶跃寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	WBAR_STEP_EN	0x0	RW	EDMAC 写缓存地址阶跃使能 0 = 禁止；1 = 使能
[30]	WBAR_STEP_DIR	0x0	RW	EDMAC 写缓存地址阶跃方向 0 = EDMAC 写缓存地址增加 1 = EDMAC 写缓存地址减少
[29:22]	保留	0x0	RO	---
[21:0]	WBAR_STEP	0x0	RW	EDMAC 写缓存阶跃基地址 当次循环开始时（第一个次循环除外），EDMACWBAR 字段将用 WBAR_STEP 更新。

21.5.2.12 通道最后次要总数寄存器（EDMACLASTMINSUMR0/1）

偏移地址：0x002C~0x002F　　复位值：0x00000000

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

图表 21-24：通道 0 最后次要总数寄存器

偏移地址：0x006C~0x006F　　复位值：0x00000000

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

图表 21-25：通道 1 最后次要总数寄存器

比特位	名称	复位值	读写属性	功能说明
[31]	LASTMINSUM_EN	0x0	RW	使能最后次要总数功能 0 = 禁止； 1 = 使能
[30:22]	保留	0x0	RO	---
[21:0]	LAST_MINOR_SUM	0x0	RW	通道最后次要传输总数 当设置 LASTMINSUM_EN 时，LAST_MINOR_SUM 字段控制在通道的最后次要块传输中传输的数据总和（以字节为单位）。

21.6 功能描述

EDMAC 支持 RAM 和外围设备间进行数据传输，包含了信道 0 和信道 1 两个信道，每个信道都支持不同的外围设备和不同的传输类型。优先通道启动块传输后，edmac 将交换优先级。

21.6.1 传输类型

21.6.1.1 从 RAM 到外围设备，然后从外围设备到 RAM

从 SRAM 中读取数据
把数据写到外围设备
查询外围设备是否准备好
从外围设备读取数据
把数据写到 SRAM 中
如果传输没有完成，重复 1～5

21.6.1.2 从 SRAM 读数据写到外围设备

从 SRAM 中读取数据
把数据写到外围设备
查询外围设备是否准备好
如果传输没有完成，重复 1～3

21.6.1.3 从外围设备读数据写到 RAM

查询外围设备是否准备好
从外围设备读取数据
把数据写到 SRAM 中
如果传输没有完成，重复 1～3

21.6.1.4 从 SRAM 读数据写到 SRAM

从 SRAM 中读取数据
把数据写到 SRAM
如果传输没有完成，重复 1～2

21.6.2 双信道配置

在启动某个通道传输前，软件需要对此信道的读缓存地址寄存器，写缓存地址寄存器，传输数据数量，传输类型，外设的选择和 VALID 位进行配置。如果需要产生中断，还需设置相应的中断使能位。当 EDMACEN 位设为 1 且某个通道可用时，开始传输数据。

当系统复位或 EDMACEN 为 0 时，通道 0 具有较高的优先级。信道 0 开始传输后，通道 1 获得较高的优先级。信道 0 完成传输后，如果通道 1 的 VALID 有效则通道 1 开始传输，通道 0 获得较高的优先级。下面介绍几种关于优先级可能出现的情况：

通道 0 具有较高的优先级且两个通道都有效。通道 0 开始数据传输，通道 1 获得较高的优先级。当通道 0 完成传输后通道 1 开始传输，同时通道 0 获得较高的优先级。
通道 0 具有较高的优先级且只有通道 0 有效。通道 0 启动传输，通道 1 获得较高优先级。
通道 0 具有较高优先级且只有通道 1 有效。通道 1 启动传输，通道 0 保持较高优先级。
通道 1 具有较高优先级且两个通道都有效。通道 1 启动传输，同时通道 0 获得较高优先级。通道 1 完成传输后启动通道 0，通道 1 获得较高优先级。
通道 1 具有较高优先级且只有通道 1 有效。通道 1 启动，通道 0 获得较高优先级。
通道 1 具有较高优先级且只有通道 0 有效。通道 0 启动传输，通道 1 保持较高优先级。

21.7 中断描述

表格 21-2：EDMAC4 种中断请求

中断请求	标志	使能位
通道 0 传输完成	DONE0	DONE0IE
通道 1 传输完成	DONE1	DONE1IE
通道 0 传输启动	START0	START0IE
通道 1 传输启动	START1	START1IE

21.7.1 通道 0 传输完成（DONE0）

通道 0 传输完成后产生 DONE0 中断，写入 1 被清除。

21.7.2 通道 1 传输完成（DONE1）

通道 1 传输完成后产生 DONE1 中断，写入 1 被清除。

21.7.3 通道 0 传输启动（START0）

通道 0 启动传输后产生 START0 中断，写入 1 被清除。

21.7.4 通道 1 传输启动（START1）

通道 1 启动传输后产生 START0 中断，写入 1 被清除。

22 直接内存访问控制器模块（DMA）

22.1 概述

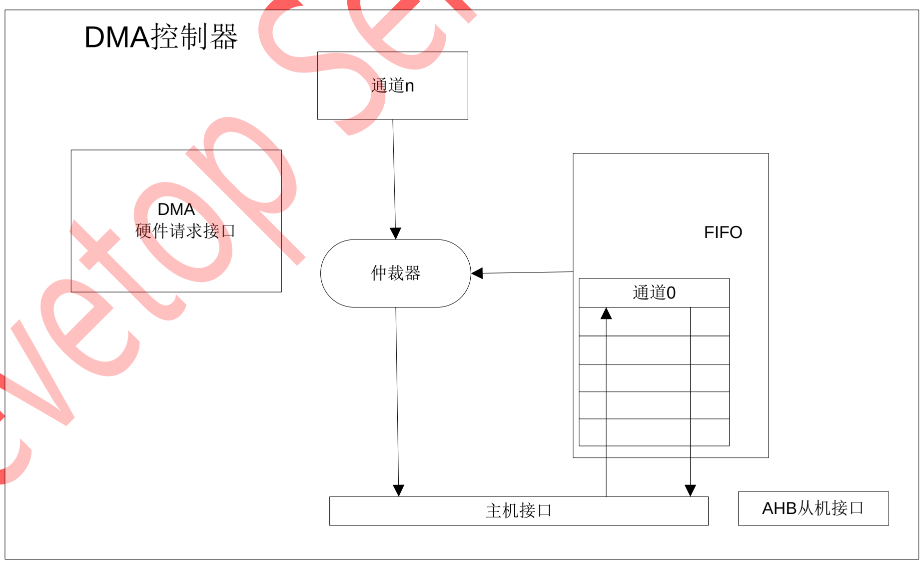
直接内存访问控制器（DMA）模块是一个主机，它可以用最小的 CPU 消耗来传输数据。DMA 模块提供 4 个通道，允许字节，半字和字传输。

22.2 特性

模块特性包括：

- DMA1 有 4 个独立的可编程通道，DMA2 有 2 个独立的可编程通道
- 支持 8/16/32 位数据传输
- 支持单次传输，连续 4/8/16 次传输
- 支持链表传输
- 遵循一个固定的优先级
- 支持通道暂停操作
- 支持外设传输

22.3 框图



图表 22-1：DMA 控制块框图

22.4 工作模式

22.4.1 低功耗模式

DMA 控制器不受低功耗模式控制。CPU 可以通过设置模块时钟来停止 DMAC。

22.5 内存映射和寄存器

22.5.1 内存映射

表格 22-1：显示了 DMAC 寄存器映射

表格 22-1：DMAC 寄存器映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	DMAC 源地址寄存器 0（DMA_SADDR0）		S/U
0x0008	DMAC 目标地址寄存器 0（DMA_DADDR0）		S/U
0x0010	DMAC 链表指针 0（DMA_LLP0）		S/U
0x0018	DMAC 控制寄存器 0（DMA_CTRL0）		S/U
0x001C	DMAC 控制寄存器高位 0（DMA_CTRL_HIGH0）		S/U
0x0040	DMAC 配置寄存器 0（DMA_CFG0）		S/U
0x0044	DMAC 配置寄存器高位 0（DMA_CFC_HIGH0）		S/U
0x0058	DMAC 源地址寄存器 1（DMA_SADDR1）		S/U
0x0060	DMAC 目标地址寄存器 1（DMA_DADDR1）		S/U
0x0068	DMAC 链表指针 1（DMA_LLP1）		S/U
0x0070	DMAC 控制寄存器 1（DMA_CTRL1）		S/U
0x0074	DMAC 控制寄存器高位 1（DMA_CTRL_HIGH1）		S/U
0x0098	DMAC 配置寄存器 1（DMA_CFG1）		S/U
0x009C	DMAC 配置寄存器高位 1（DMA_CFC_HIGH1）		S/U
0x00B0	DMAC 源地址寄存器 2（DMA_SADDR2）		S/U
0x00B8	DMAC 目标地址寄存器 2（DMA_DADDR2）		S/U
0x00C0	DMAC 链表指针 2（DMA_LLP2）		S/U
0x00C8	DMAC 控制寄存器 2（DMA_CTRL2）		S/U
0x00CC	DMAC 控制寄存器高位 2（DMA_CTRL_HIGH2）		S/U
0x00F0	DMAC 配置寄存器 2（DMA_CFG2）		S/U
0x00F4	DMAC 配置寄存器高位 2（DMA_CFC_HIGH2）		S/U
0x0108	DMAC 源地址寄存器 3（DMA_SADDR3）		S/U
0x0110	DMAC 目标地址寄存器 3（DMA_DADDR3）		S/U
0x0118	DMAC 链表指针 3（DMA_LLP3）		S/U
0x0120	DMAC 控制寄存器 3（DMA_CTRL3）		S/U
0x0124	DMAC 控制寄存器高位 3（DMA_CTRL_HIGH3）		S/U

偏移地址	位 31-16	位 15-0	访问权限
0x0148	DMAC 配置寄存器 3（DMA_CFG3）		S/U
0x014C	DMAC 配置寄存器高位 3（DMA_CFC_HIGH3）		S/U
0x02C0	传输中断原始状态寄存器（DMA_RAWTFR）		S/U
0x02C8	块中断原始状态寄存器（DMA_RAWBLOCK）		S/U
0x02D0	源端中断原始状态寄存器（DMA_RAWSRCTRAN）		S/U
0x02D8	目的端中断原始状态寄存器（DMA_RAWDSTTRAN）		S/U
0x02E0	错误中断原始状态寄存器（DMA_RAWERR）		S/U
0x02E8	传输中断状态寄存器（DMA_STATTFR）		S/U
0x02F0	块中断状态寄存器（DMA_STATBLOCK）		S/U
0x02F8	源端中断状态寄存器（DMA_STATSRCTRAN）		S/U
0x0300	目的端中断状态寄存器（DMA_STATDSTTRAN）		S/U
0x0308	错误中断状态寄存器（DMA_STATERR）		S/U
0x0310	传输中断屏蔽寄存器（DMA_MASKTFR）		S/U
0x0318	块中断屏蔽寄存器（DMA_MASKBLOCK）		S/U
0x0320	源端中断屏蔽寄存器（DMA_MASKSRCTRAN）		S/U
0x0328	目的端中断屏蔽寄存器（DMA_MASKDSTTRAN）		S/U
0x0330	错误中断屏蔽寄存器（DMA_MASKERR）		S/U
0x0338	传输中断清除寄存器（DMA_CLRTFR）		S/U
0x0340	块中断清除寄存器（DMA_CLRBLOCK）		S/U
0x0348	源端中断清除寄存器（DMA_CLRSRCTRAN）		S/U
0x0350	目的端中断清除寄存器（DMA_CLRDSTTRAN）		S/U
0x0358	错误中断清除寄存器（DMA_CLRERR）		S/U
0x0360	各中断类型寄存器（STATUSINT）		S/U
0x0368	软件源端传输请求寄存器（DMA_SRCREQ）		S/U
0x0370	软件目的端传输请求寄存器（DMA_DSTREQ）		S/U
0x0378	软件源端单次请求寄存器（DMA_SINGLESRC）		S/U
0x0380	软件目的端单次请求寄存器（DMA_SINGLEDST）		S/U
0x0388	软件源端最后一次请求寄存器（DMA_LASTSRC）		S/U
0x0390	软件目的端最后一次请求寄存器（DMA_LASTEDST）		S/U
0x0398	DMA 配置寄存器（DMA_CONFIG）		S/U
0x03A0	DMA 通道使能寄存器（DMA_CHEN）		S/U

22.5.2 寄存器描述

22.5.2.1 DMAC 源地址寄存器 n（DMA_SADDRn）

偏移地址：0x0000，0x0058，0x00B0，0x0108　　复位值：0x00000000

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

图表 22-2：DMAC 源地址寄存器 n（DMA_SADDRn）

比特位	名称	复位值	读写属性	功能说明
[31:0]	DMA_SADDR [31:0]	0x0	RW	**指向源数据的内存地址**

22.5.2.2 DMAC 目的地址寄存器 n（DMA_DADDRn）

偏移地址：0x0008，0x0060，0x00B8，0x0110　　复位值：0x00000000

31	30	29	28	27	26	25	24
DMA_DADDR[31:24]							
rw							

23	22	21	20	19	18	17	16
DMA_DADDR[23:16]							
rw							

15	14	13	12	11	10	9	8
DMA_DADDR[15:8]							
rw							

7	6	5	4	3	2	1	0
DMA_DADDR[7:0]							
rw							

图表 22-3：DMAC 目的地址寄存器 n（DMA_DADDRn）

比特位	名称	复位值	读写属性	功能说明
[31:0]	DMA_DADDR[31:0]	0x0	RW	**指向目的数据的内存地址**

22.5.2.3 DMAC 链表寄存器 n（DMA_LLPn）

偏移地址：0x0010，0x0068，0x00C0，0x0118　　复位值：0x00000000

31	30	29	28	27	26	25	24
LOC[31:24]							
rw							

23	22	21	20	19	18	17	16
LOC[23:16]							
rw							

15	14	13	12	11	10	9	8
LOC[15:8]							
rw							

7	6	5	4	3	2	1	0
LOC[7:2]						保留	
rw						ro	

图表 22-4：DMAC 链表寄存器 n（DMA_LLPn）

比特位	名称	复位值	读写属性	功能说明
[31:2]	LOC[31:2]	0x0	RW	**指向目的数据的内存地址** 链表中的内存起始地址。**注意：**地址的低 2 位不存储，因为地址是 32 位对齐的。链表地址总是 32 位访问，不能改成其他非 32 位访问。
[1:0]	保留	0x0	RO	---

22.5.2.4 DMAC 控制寄存器 n（DMA_CTRLn）

偏移地址：0x0010，0x0068，0x00C0，0x0118　　复位值：0x00304801

31	30	29	28	27	26	25	24
保留			LLP_SRC_EN	LLP_DST_EN	SMS[1:0]		DMS[1]
ro			rw	rw	rw		rw

23	22	21	20	19	18	17	16
DMS[[0]	TT_FC[2:0]			保留			SRC_MSIZE[2]
rw	rw			ro			rw

15	14	13	12	11	10	9	8
SRC_MSIZE[1:0]		DEST_MSIZE[2:0]			SINC[1：0]		DINC[1]
rw		rw			rw		rw

7	6	5	4	3	2	1	0
DINC[0]	SRC_TR_WIDTH[2:0]			DST_TR_WIDTH[2:0]			INT_EN
rw	rw			rw			rw

图表 22-5：DMAC 控制寄存器 n（DMA_CRTLn）

比特位	名称	复位值	读写属性	功能说明
[31:29]	保留	0x0	RO	---
[28]	LLP_SRC_EN	0x0	RW	**源端链表功能使能** 0 = DMAC 源端不使用链表操作 1 = DMAC 源端使用链表操作
[27]	LLP_DST_EN	0x0	RW	**目的端链表功能使能** 0 = DMAC 目的端不使用链表操作 1 = DMAC 目的端使用链表操作
[26:25]	SMS[1:0]	0x0	RW	**SMS：源端主机选择** 在本 DMAC 中，这 2 个 bit 固定为 2’b00
[24:23]	DMS[1:0]	0x0	RW	**DMS：目的端主机选择** 在本 DMAC 中，这 2 个 bit 固定为 2’b00

比特位	名称	复位值	读写属性	功能说明
[22:20]	TT_FC[2:0]	0x3	RW	**传输类型和流程控制** 000 = 内存到内存，DMAC 控制 001 = 内存到外设，DMAC 控制 010 = 外设到内存，DMAC 控制 011 = 外设到外设，DMAC 控制 100 = 外设到内存，外设控制 101 = 外设到外设，源端外设控制 110 = 内存到外设，外设控制 111 = 外设到外设，目的端外设控制
[19:17]	保留	0x0	RO	---
[16:14]	SRC_MSIZE[2:0]	0x1	RW	**源端连续传输长度** 000 = 1 001 = 4 010 = 8 011 = 16 100 = 32 101 = 64 110 = 128 000 = 256
[13:11]	DEST_MSIZE[2:0]	0x1	RW	**目的端连续传输长度** 000 = 1 001 = 4 010 = 8 011 = 16 100 = 32 101 = 64 110 = 128 000 = 256
[10:9]	SINC[1:0]	0x0	RW	**DMAC 源端地址变化** 00 = 增加 01 = 减少 1x = 不变
[8:7]	DINC[1:0]	0x0	RW	**DMAC 目的端地址增加** 00 = 增加 01 = 减少 1x = 不变

比特位	名称	复位值	读写属性	功能说明
[6:4]	SRC_TR_WIDTH[2:0]	0x0	RW	**DMAC 源端传输宽度** 000 = 8 位 001 = 16 位 010 = 32 位 011 = 64 位 100 = 128 位 101 = 256 位 11x = 256 位
[3:1]	DST_TR_WIDTH[2:0]	0x0	RW	**DMAC 目的端传输宽度** 000 = 8 位 001 = 16 位 010 = 32 位 011 = 64 位 100 = 128 位 101 = 256 位 11x = 256 位
[0]	INT_EN	0x1	RW	**DMAC 中断使能位** 0 = DMAC 中断不使能；1 = DMAC 中断使能

22.5.2.5 DMAC 控制寄存器高位 n（DMA_CTRL_HIGHn）

偏移地址：0x001C，0x0074，0x00CC，0x0124 复位值：0x00000002

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

图表 22-6：DMAC 控制寄存器高位 n（DMA_CRTL_HIGHn）

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:0]	BLOCK_TS[11:0]	0x2	RW	**块传输长度** 当 DMAC 是流程控制者时，用户在使能信道之前配置这个区域，配置块长度。在此区域的数字表明了单次块传输的总数。单次传输取决于 AMBA 的一个 beat，传输宽度取决于 CTRLx 的 SRC_TR_WIDTH 位。 一旦传输开始，这个区域的读返回值是源端已经读到的数据。如果是外设控制流程，这个读返回值可能不准确。

22.5.2.6 DMAC 配置寄存器 n（DMA_CFGn）

偏移地址：0x0040，0x0098，0x00F0，0x0148 复位值：0x00000C00

31	30	29	28	27	26	25	24
保留		MAX_ABRST[9:4]					
ro		rw					

23	22	21	20	19	18	17	16
MAX_ABRST[3:0]				保留			
rw				ro			

15	14	13	12	11	10	9	8
保留				HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP
ro				rw	rw	rw	rw

7	6	5	4	3	2	1	0
CH_PRIOR[2:0]			保留				
rw			ro				

图表 22-7：DMAC 控制寄存器高位 n（DMA_CFGn）

比特位	名称	复位值	读写属性	功能说明
[31:30]	保留	0x0	RO	---
[29:20]	MAX_ABRST[3:0]	0x0	RW	**最大 AMBA 总线连续传输长度** 最大 AMBA 总线连续传输长度用于 DMA 的传输。 0 表示本通道 DMA 不限制 AMBA 连续传输长度
[19:12]	保留	0x0	RO	---
[11]	HS_SEL_SRC	0x1	RW	**源端软件或者硬件握手选择** 这个寄存器选择哪个握手机制-源端使用硬件或者软件握手机制 0 = 使用硬件握手机制。软件触发的源端请求是不被响应的。 1 = 使用软件握手机制。硬件触发的源端请求是不被响应的。如果源端硬件是存储器，这个位可以忽略。
[10]	HS_SEL_DST	0x1	RW	**目的端软件或者硬件握手选择** 这个寄存器选择哪个握手机制-目的端使用硬件或者软件握手机制 0 = 使用硬件握手机制。软件触发的目的端请求是不被响应的。 1 = 使用软件握手机制。硬件触发的目的端请求是不被响应的。如果源端硬件是存储器，这个位可以忽略。

比特位	名称	复位值	读写属性	功能说明
[9]	FIFO_EMPTY	0x1	RW	**用来指示是否有数据在 FIFO 中** 0 = 通道 FIFO 非空；1 = 通道 FIFO 空 **注意：**当 DMA 使能时，该位写入无效。
[8]	CH_SUSP	0x0	RW	**通道暂停** 暂停 DMA 的数据传输，直到该位被清除。不保证目前正在传输的数据是否完成。可以与 CFGx FIFO_EMPTY 位使用，才能保证不丢失数据。 0 = 通道 FIFO 非空；1 = 通道 FIFO 空
[7]	CH_PRIOR	-	RW	**通道优先级** 优先级为 3 代表最高优先级，0 是最低优先级。本区域配置只能为 0-3。信道 0，默认值是 0，通道 1 默认值是 1，以此类推。
[4:0]	保留	0x0	RO	---

22.5.2.7 DMAC 配置寄存器高位 n（DMA_CFG_HIGHn）

偏移地址：0x0040，0x0098，0x00F0，0x0148　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留	DST_PER[3:0]				SRC_PER[3:1]		
ro	rw				rw		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SRC_PER[0]	保留						FC_MODE
rw	ro						rw

图表 22-8：DMAC 配置寄存器高位 n（DMA_CFG_HIGHn）

比特位	名称	复位值	读写属性	功能说明
[31:15]	保留	0x0	RO	---
[14:11]	DST_PER[3:0]	0x0	RW	**配置目的端硬件握手接口** **DMAC1：** 0000 = spi1 发送 0001 = spi2 发送 0010 = ssi2 发送 0011 = spi1 接收 0100 = spi2 接收 0101 = ssi2 接收 0110 = adc 0111 = 保留 1000 = 保留 1001 = 保留 1010 = sci1 发送 1011 = sci1 接收 1100 = sci3 发送 1101 = sci3 接收 DMAC2： 0000 = sci2 接收 0001 = sci2 发送 0010 = ssi1 接收

比特位	名称	复位值	读写属性	功能说明
				0011 = ssi1 发送 0100 = spi3 接收 0101 = spi3 发送 0110 = reserved 0111 = reserved 1000 = dac 1001 = reserved 1010 = reserved 1011 = reserved 1100 = sddc 1101 = reserved
[10:7]	SRC_PER[3:0]	0x0	RW	**配置源端硬件握手接口** 配置信息与 DST_PER 相同
[6:1]	保留	0x0	RO	---
[0]	FC_MODE	0x0	RW	**FC_MODE：流程控制模式** 当目的端是外设是流程控制时，决定什么时候源端请求被响应。 0 = 源端请求会被响应，数据预取是打开的。 1 = 源端请求不会被响应，直到目的端传输请求到来时。在这个模式中，传输数据的量会被限制，因此保证了目的端的优先性，数据预取是关闭的。

22.5.2.8 中断原始状态寄存器（DMA_RAWTFR/ DMA_RAWBLOCK/ DMA_RAWSRC/ DMA_RAWDST/ DMA_RAWERR)

偏移地址：0x02C0，0x02C8，0x02D0，0x02D8，0x02E0　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留				RAW[3:0]			
ro				ro			

图表 22-9：DMAC 中断原始状态寄存器

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0x0	RO	---
[3:0]	RAW[3:0]	0x0	RO	**原始中断状态** 原始的中断存储在这些原始中断寄存器中，这个状态在屏蔽之前。每个原始中断状态寄存器都有一位对应一个通道。例如：DMA_RAWTFR[2]是信道 2 的信道传输完成标志。每位的清除都是在相应清除状态寄存器中对应寄存器位中写 1。

22.5.2.9 中断状态寄存器（DMA_STATTFR/ DMA_STATBLOCK/ DMA_STATSRC/ DMA_STATDST/ DMA_STATERR）

偏移地址:0x02E8,0x02F0,0x02F8,0x0300,0x0308　复位值：0x00000000

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

图表 22-10：DMAC 中断原始状态寄存器

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0x0	RO	---
[3:0]	STATUS[3:0]	0x0	RO	**中断状态** 中断状态存储在这些中断寄存器中，这个状态在屏蔽之后。每个中断状态寄存器都有一位对应一个通道。例如：DMA_STATUSTFR[2]是信道 2 的信道传输完成标志。每位的清除都是在响应清除状态寄存器中对应寄存器位中写 1，这些寄存器中的内容用于产生中断信号。

22.5.2.10 中断屏蔽寄存器（DMA_MASKTFR/ DMA_MASKBLOCK/ DMA_MASKSRC/ DMA_MASKDST/ DMA_MASKERR)

偏移地址：0x0310，0x0318，0x0320，0x0328，0x0330　　复位值：0x00000000

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

图表 22-11：DMAC 中断屏蔽寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	INT_MASK_WE[3:0]	0x0	RW	**中断屏蔽写使能位** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	INT_MASK[3:0]	0x0	RW	**中断屏蔽位** 0 = 屏蔽；1 = 不屏蔽

22.5.2.11 中断清除寄存器（DMA_CLRTFR/ DMA_CLRBLOCK/ DMA_CLRSRC/ DMA_CLRDST/ DMA_CLRERR)

偏移地址：0x0338，0x0340，0x0348，0x0350，0x0358 复位值：0x00000000

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

图表 22-12：DMAC 中断清除寄存器

比特位	名称	复位值	读写属性	功能说明
[31:4]	保留	0x0	RO	---
[3:0]	CLEAR[3:0]	0x0	WO	**清除位** 中断原始状态寄存器以及中断状态寄存器可以在相对应的清除寄存器中写 1 清 0。例如：DMA_CLRTFR[2]是通道 2 传输完成状态位的清除位。写 0 无效。这些寄存器不可读。

22.5.2.12 合并中断状态寄存器（DMA_STATE）

偏移地址：0x0360　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
ro							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
ro							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留			ERR	DSTT	SRCT	BLOCK	TRR
ro			ro	ro	ro	ro	ro

图表 22-13：DMAC 合并中断状态寄存器

比特位	名称	复位值	读写属性	功能说明
[31:5]	保留	0x0	RO	---
[4]	ERR	0x0	RO	**错误状态寄存器位的或运算值** 0 = 无错误；1 = 有错误
[3]	DSTT	0x0	RO	**目的端状态寄存器位的或运算值** 0 = 未完成；1 = 完成
[2]	SRCT	0x0	RO	**源端状态寄存器位的或运算值** 0 = 未完成；1 = 完成
[1]	BLOCK	0x0	RO	**块状态寄存器位的或运算值** 1 = 有错误；0 = 无错误
[0]	TRR	0x0	RO	**状态寄存器位的或运算值** 0 = 未完成；1 = 所有传输都完成

22.5.2.13 软件源端传输请求寄存器（DMA_SRCREQ）

偏移地址：0x0368 复位值：0x00000000

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

图表 22-14：DMAC 软件源端传输请求寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	SRC_REQ_WE [3:0]	0x0	RW	**源端请求写使能** 0 = 写不使能；1 = 写使能
[8:4]	保留	0x0	RO	---
[3:0]	SRC_REQ[3:0]	0x0	RW	**源端请求** SRC_REQ 与 SRC_SGLREQ 或者 DST_REQ 与 DST_SGLREQ 对应位写 1，代表开始一个单独的传输，当传输完成的时候，相对应的位都会被硬件清 0。

22.5.2.14 软件源端传输请求寄存器（DMA_DSTREQ）

偏移地址：0x0370　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留				DST_REQ_WE[3:0]			
ro				rw			

7	6	5	4	3	2	1	0
保留				DST_REQ[3:0]			
ro				rw			

图表 22-15：DMAC 软件目的端传输请求寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	DST_REQ_WE [3:0]	0x0	RW	**目的端请求写使能** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	DST_REQ[3:0]	0x0	RW	**目的端请求** SRC_REQ 与 SRC_SGLREQ 或者 DST_REQ 与 DST_SGLREQ 对应位写 1，代表开始一个单独的传输，当传输完成的时候，相对应的位都会被硬件清 0。

22.5.2.15 软件源端单次传输请求寄存器（DMA_SGLSRCREQ）

偏移地址：0x0378　　　　　　　　　　　　　　复位值：0x00000000

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

图表 22-16：DMAC 软件源端单次传输请求寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	SRC_SGLREQ_WE[3:0]	0x0	RW	**源端请求写使能** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	SRC_SGLREQ[3:0]	0x0	RW	**源端请求** SRC_REQ 与 SRC_SGLREQ 或者 DST_REQ 与 DST_SGLREQ 对应位写 1，代表开始一个单独的传输，当传输完成的时候，相对应的位都会被硬件清 0。

22.5.2.16 软件目的端单次传输请求寄存器（DMA_SGLDSTREQ）

偏移地址：0x0380 复位值：0x00000000

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

图表 22-17：DMAC 软件目的端单次传输请求寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	DST_SGLREQ_WE[3:0]	0x0	RW	**源端请求写使能** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	DST_SGLREQ[3:0]	0x0	RW	**目的端请求** SRC_REQ 与 SRC_SGLREQ 或者 DST_REQ 与 DST_SGLREQ 对应位写 1，代表开始一个单独的传输，当传输完成的时候，相对应的位都会被硬件清 0。

22.5.2.17 软件源端最后一次传输请求寄存器（DMA_LASTSRCREQ）

偏移地址：0x0388　　复位值：0x00000000

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

图表 22-18：DMAC 软件源端最后一次传输请求寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	SRC_LSTREQ_WE[3:0]	0x0	RW	**SRC_LSTREQ_WE：源端最后一次请求写使能** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	SRC_LSTREQ[3:0]	0x0	RW	**SRC_LSTREQ：源端最后一次请求** SRC_REQ 与 SRC_LSTREQ 或者 DST_REQ 与 DST_LSTREQ 对应位写 1，代表开始最后一次传输，当传输完成的时候，相对应的位都会被硬件清 0。

22.5.2.18 软件目的端最后一次传输请求寄存器（DMA_LSTDSTREQ）

偏移地址：0x0390　　复位值：0x00000000

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

图表 22-19：DMAC 软件目的端最后一次传输请求寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	DST_LSTREQ_WE[3:0]	0x0	RW	**源端最后一次请求写使能** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	DST_LSTREQ[3:0]	0x0	RW	**源端最后一次请求** SRC_REQ 与 SRC_LSTREQ 或者 DST_REQ 与 DST_LSTREQ 对应位写 1，代表开始最后一次传输，当传输完成的时候，相对应的位都会被硬件清 0。

22.5.2.19 DMA 配置寄存器（DMACONFIG）

偏移地址：0x0398 复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							DMA_EN
ro							rw

图表 22-20：DMA 配置寄存器

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	---
[0]	DMA_EN	0x0	RW	**DMA_EN：DMA 使能** 0 = 不使能；1 = 使能

22.5.2.20 DMA 通道使能寄存器（DMA_CHEN）

偏移地址：0x03A0 复位值：0x00000000

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

图表 22-21：DMA 通道使能寄存器

比特位	名称	复位值	读写属性	功能说明
[31:12]	保留	0x0	RO	---
[11:8]	CH_EN_WE[3:0]	0x0	RW	**DMA 通道写使能位** 0 = 写不使能；1 = 写使能
[7:4]	保留	0x0	RO	---
[3:0]	CH_EN[3:0]	0x0	RW	**DMA 通道使能位** 0 = 不使能；1 = 使能

22.6 功能描述

22.6.1 单次传输配置流程

读通道使能寄存器来选择一个空闲的通道

中断清除寄存器 DMA_CLRTFR/ DMA_CLRBLOCK/ DMA_CLRSRC/ DMA_CLRDST/ DMA_CLRERR 中写 1 清 0，来清除之前 DMA 传输的一些未处理的中断。读对应的原始中断寄存器以及中断状态寄存器来确认所有中断都被清除。

配置信道寄存器

1. 写信道源地址寄存器
2. 写通道目的地址寄存器
3. 配置 CTRLx 以及 CFCx 寄存器，将 LLPx 配置成 0
4. 写 DMA 通道控制寄存器 CTRLx，举例如下：
   - 配置传输类型（存储器或者非存储器，源端或者目的端）
   - 配置传输特性，譬如：
     a) 源端传输宽度 SRC_TR_WIDTH 区域
     b) 目的端传输宽度 DST_TR_WIDTH 区域
     c) 源端总线层次 SMS 区域
     d) 目的端总线层次 DMS 区域
     e) 源端地址是否增加，减少或者不变，SINC 区域
     f) 目的端地址是否增加，减少或者不变，DINC 区域
5. 配置信道配置寄存器信息 CFGx
   - 配置源端以及目的端外设的握手协议类型，是软件或者硬件握手如果是存储器传输则不需要。这个步骤还需要配置 HS_SEL_SRC/ HS_SEL_DST 位。写 0 使能硬件握手协议，写 1 使能软件握手协议。
   - 如果源端或者目的端的硬件握手协议被使能，给源端或者目的端安排一个握手外设，配置 SRC_PER 或者 DEST_PER。

配置 DMA_EN 为 1。

源端和目的端提出单次或者连续请求 DMA 传输来传输数据（假设是非存储器外设）。DMA 在每次传输完成退出传输。

一旦传输结束，硬件设置中断，关闭通道。这个时候，你可以检查相应的完成或者传输完成中断，或者检查原始中断状态寄存器，对应位被拉高位来侦测传输的结束。注意如果侦测使用之后，软件必须保证通道使能时，传输完成中断时是被清除的。

22.6.2 使用链表传输的多块传输配置流程

读通道使能寄存器来选择一个空闲的通道

在存储器中设置链表选项。将链表地址信息写入链表的 CTRL 寄存器位置。例如，在寄存器中，你可以如下配置:

1. 配置传输类型（存储器或者非存储器，源端或者目的端），以及通过 TT_FC 设置流程控制设备。
2. 设置传输特性，譬如:
   - 源端传输宽度 SRC_TR_WIDTH 区域
   - 目的端传输宽度 DST_TR_WIDTH 区域
   - 源端地址是否增加，减少或者不变，SINC 区域
   - 目的端地址是否增加，减少或者不变，DINC 区域

配置信道配置寄存器信息 CFGx

1. 配置源端以及目的端外设的握手协议类型，是软件或者硬件握手如果是存储器传输则不需要。这个步骤还需要配置 HS_SEL_SRC/ HS_SEL_DST 位。写 0 使能硬件握手协议，写 1 使能软件握手协议。
2. 如果源端或者目的端的硬件握手协议被使能，给源端或者目的端安排一个握手外设，配置 SRC_PER 或者 DEST_PER。

确保链表的 CTLx 区域入口已设置。

确保链表的 LLPx 区域为非 0（除非是最后一次），执行下一次链表的起始地址

确保链表的 SARx 以及 DARx 区域指向源端以及目的端的首地址。

中断清除寄存器 DMA_CLRTFR/ DMA_CLRBLOCK/ DMA_CLRSRC/ DMA_CLRDST/ DMA_CLRERR 中写 1 清 0，来清除之前 DMA 传输的一些未处理的中断。读对应的原始中断寄存器以及中断状态寄存器来确认所有中断都被清除。

配置 CTRLx 以及 CFGx 寄存器。

配置 LLPx 寄存器，指向第一次的链表地址。

配置 DMA_EN 为 1。

DMA 会曲第一次的链表地址的控制数据。

源端和目的端提出单次或者连续请求 DMA 传输来传输数据（假设是非存储器外设）。DMA 在每次传输完成退出传输。

DMA 不需要等待块中断清除，继续取下次的链表的中的信息，然后自动重新覆盖通道 SARx，DARx，LLPx 以及 CTLx 寄存器。直到所有的块传输结束。

22.6.3 取消传输流程

在正常操作中，软件通过在通道使能寄存器 CH_EN 位中写 1，使能了一个通道，硬件会在传输完成后清除 CH_EN 位。

建议软件需要取消通道传输但不丢失数据时，配合使用 CH_SUSP 位和 FIFO_EMPTY 位。

如果软件希望在一个 DMA 传输完成前取消一个 DMA 传输，可以先设置 CFGx .CH_SUSP 位来告诉 DMA 暂停源端所有的传输。因此，源端不会再收到数据。
软件然后侦测 CFGx.FIFO_EMPTY 位直到该位为 1，表明内部 FIFO 中为空，所有的数据已被传输。
如果需要继续该通道传输，清除 CFGx.CH_SUSP 位，DMA 会继续传输至结束。

22.7 中断描述

本 DMA 在外部只有一个合并中断信号。如果出现中断，想要知道何种中断，可以查询合并中断状态寄存器（DMA_STATE）。如果想要明确了解哪个通道产生中断，可以查询中断状态寄存器（DMA_STATTFR/DMA_STATBLOCK/DMA_STATSRC/DMA_STATDST/ DMA_STATERR)。每个中断状态寄存器都有一位对应一个通道。例如：DMA_TFR[2]是信道 2 的信道传输完成标志。每位的清除都是在相应清除状态寄存器中对应寄存器位中写 1，这些寄存器中的内容用于产生中断信号。

23 数字模拟转化器（DAC）

23.1 概述

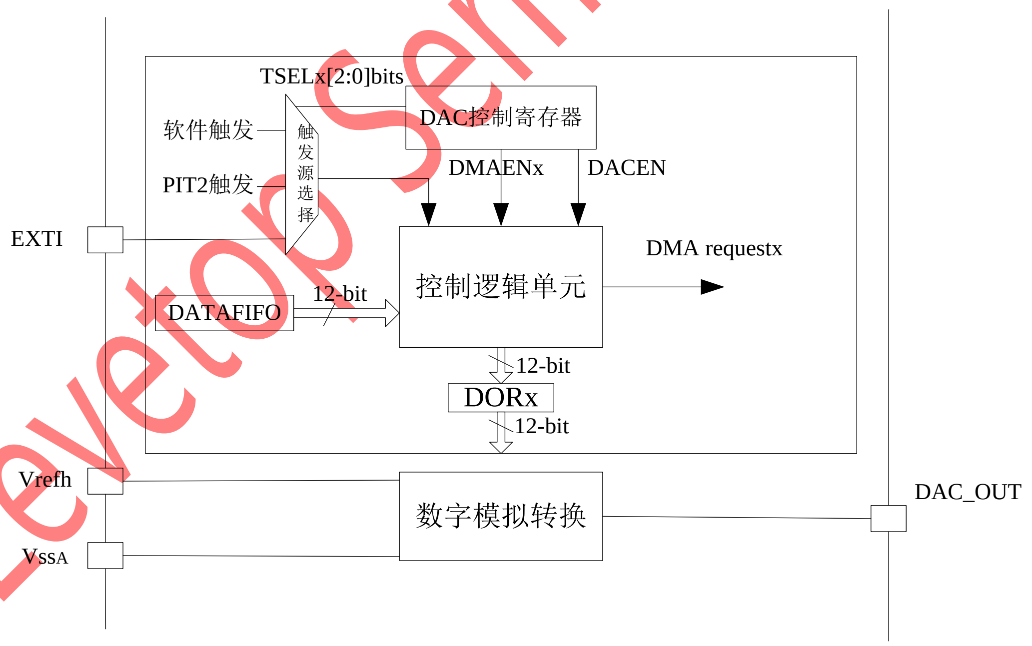
DAC 模块是一个 12 位的电压输出数模转换器。DAC 可配置为 8 位或 12 位模式，并可与 DMA 控制器结合使用。数据可以左对齐或右对齐。输入参考电压 Vrefh 可用。输出可以选择缓冲为更高的电流驱动器。

23.2 DAC 主要特性

数据左对齐或右对齐
支持 DMA 功能
外部转换触发器
可编程内部缓冲区
输入电压参考 Vrefh
基于 FIFO 的操作

23.3 DAC 功能描述

图表 21-1 显示 DAC 的框图



图表 23-1：DAC 框图

23.3.1 DAC 使能

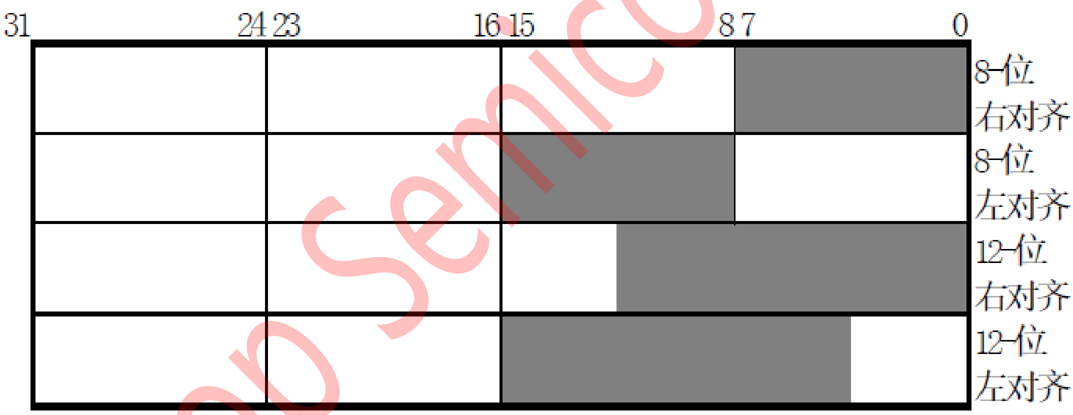
DAC 可以通过在 DAC 控制寄存器中设置 DACEN 位来启动，然后在一个启动时间唤醒后启用 DAC。

23.3.2 DAC 数据格式

DAC 数据格式取决于 DAC 控制寄存器中的 RES 和 ALIGN 位，写入到 DAC 数据寄存器的数据将被安排和存储在 DAC 数据 FIFO 中。

表格 23-1：数据格式

配置	RES	ALIGN
8 位，右对齐	0	0
8 位，左对齐	0	1
12 位，右对齐	1	0
12 位，左对齐	1	1

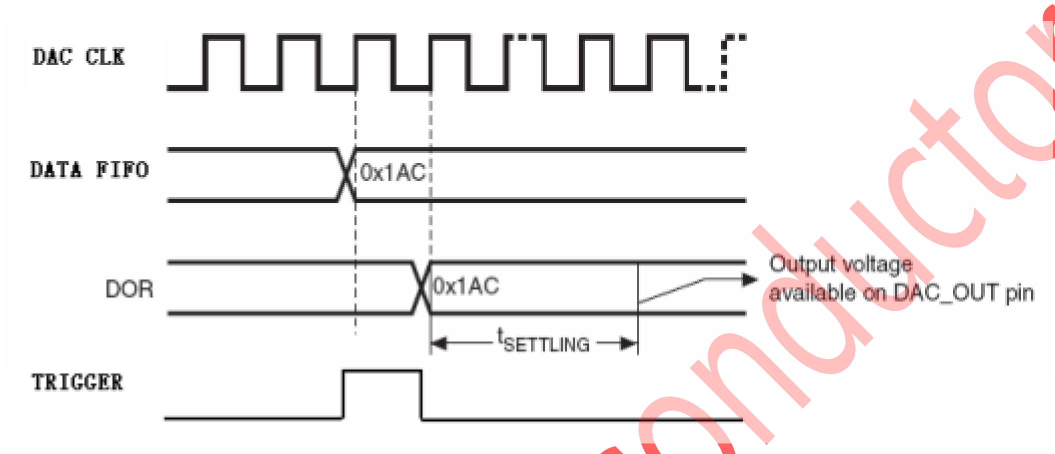


图表 23-2：有效数据存储到 FIFO 中

23.3.3 DAC 转换

DAC_DOR 不能被直接写入数据，执行对 DAC 的任何数据传输都必须通过加载到 DAC 数据 FIFO（写入到 DAC_DR）。

当检测到触发时，DAC 数据 FIFO 寄存器中存储的数据将自动传输到 DAC_DOR 寄存器中。当 DAC_DOR 加载 DAC 数据 FIFO 寄存器内容时，根据电源电压和模拟输出负载进行时间调整后，模拟输出电压就可用了。



图表 23-3：DAC 数据时序图

23.3.4 DAC 输出电压

数字输入在 0V 和 Vrefh 之间的线性转换为输出电压。
每一个 DAC 通道管脚上的模拟输出电压由下面公式确定：

DACoutput = Vrefh * DOR / 4095

23.3.5 DMA 请求

DAC FIFO 中的有效数据数一旦小于或等于 DMAEN 设置时 DMATH 指定的阈值，DAC 就会向 DMA 发送 DMA 请求。

23.4 内存映射和寄存器

本小节描述内存映射和寄存器结构。

23.4.1 内存映射

有关 DAC 内存映射的描述，请参阅表格 21-2。

表格 23-2：DAC 内存映射

偏移地址	位 31-16	位 15-0	访问权限
0x0000	DAC 控制寄存器（DAC_CR）		S/U
0x0004	DAC 数据寄存器（DAC_DR）		S/U
0x0008	DAC 软件触发寄存器(DAC_SWTR)		S/U
0x000C	DAC 数据输出寄存器（DAC_DOR）		S/U
0x0010	DAC FIFO 状态寄存器（DAC_FSR）		S/U
0x0014	DAC 触发寄存器（DAC_TRIMR）		S/U
0x0018	保留		S/U
0x001C	保留		S/U

注意：

S = 只允许 CPU 管理模式访问。

用户模式访问仅管理地址位置没有影响，并导致循环终止传输错误。

23.4.2 寄存器

23.4.2.1 DAC 控制寄存器（DAC_CR）

偏移地址：0x0000~0x0003 复位值：0x00000000

31	30	29	28	27	26	25	24
FCLR	OVFIE	UDFIE	STHIE	保留			STH
rw	rw	rw	rw	ro			rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAEN	保留						DMATH
rw	ro						rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留	EXT_TMOD[2:0]			保留	TSEL[1:0]		保留
ro	rw	rw	rw	ro	rw	rw	ro
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留				RES	ALIGN	BUFDIS	DACEN
ro				rw	rw	rw	rw

图表 23-4：DAC 控制寄存器（DAC_CR）

比特位	名称	复位值	读写属性	功能说明
[31]	FCLR	0x0	RW	DAC FIFO 清除位 写 1 到该位将重置 DAC FIFO
[30]	OVFIE	0x0	RW	DAC 溢出中断使能位 0 = FIFO 溢出中断禁止 1 = FIFO 溢出中断使能
[29]	UDFIE	0x0	RW	DAC 下溢中断使能位 0 = FIFO 下溢中断禁止 1 = FIFO 下溢中断使能
[28]	STHIE	0x0	RW	DACFIFO 服务阈值中断使能位 0 = FIFO 服务阈值中断禁止 1 = FIFO 服务阈值中断使能
[27:25]	STH	0x0	RW	DAC FIFO 服务阈值 当 DAC FIFO 中的有效数据数小于或等于指定的阈值时，将使 DAC_FSR 中 DAC FIFO 服务标志置 1。
[23]	DMAEN	0x0	RW	DMA 使能位 0 = DMA 禁止；1 = DMA 使能
[22:17]	保留	0x0	RO	--

比特位	名称	复位值	读写属性	功能说明
[16]	DMATH	0x0	RW	DMA 阈值 当 DAC FIFO 中的有效数据数小于或等于设置 DMAEN 时的阈值时，DAC 将向 DMA 发送 DMA 请求
[15]	保留	0x0	RO	--
[14:12]	EXT_TMOD[2:0]	0x0	RW	这些位是用来选择外部触发极性： 3’b000 检测上升沿； 3’b001 检测下降沿； 3’b010 检测上升沿和下降沿； 3’b011 检测高电压； 3’b100 检测低电压； 3’b101 保留； 3’b110 保留； 3’b111 保留；
[11]	保留	0x0	RO	--
[10:9]	TSEL[1:0]	0x0	RW	触发源选择： 2’b00 软件触发； 2’b01 硬件触发； 2’b10 Pit 触发； 2’b11 保留； **注意：** 1、外部触发使用 SS2 管脚触发。 2、Pit 触发使用 pit2 触发。
[8:4]	保留	0x0	RO	--
[3]	RES	0x0	RW	DAC 数据分辨率 0 = 8 位；1 = 12 位
[2]	ALIGN	0x0	RW	DAC 数据对齐方式 0 = 右对齐；1 = 左对齐
[1]	BUFDIS	0x0	RW	DAC 输出缓冲区禁用 0 = 缓冲区使能；1 = 缓冲区禁止
[0]	DACEN	0x0	RW	DAC 使能 0 = DAC 禁止；1 = DAC 使能

23.4.2.2 DAC 数据寄存器(DAC_DR)

偏移地址：0x0004~0x0007 复位值：0x00000000

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

图表 23-5：DAC 数据寄存器（DAC_DR）

写入 DAC_DR 将填充 DAC 数据 FIFO 从 DAC_DR 读取返回 0。

23.4.2.3 DAC 软件触发寄存器（DAC_SWTR）

偏移地址：0x0008~0x000B　　　　　　　　复位值：0x00000000

31	30	29	28	27	26	25	24
保留							
ro							

23	22	21	20	19	18	17	16
保留							
ro							

15	14	13	12	11	10	9	8
保留							
ro							

7	6	5	4	3	2	1	0
保留							SWTRIG
ro							rw

图表 23-6：DAC 软件触发寄存器（DAC_SWTR）

比特位	名称	复位值	读写属性	功能说明
[31:1]	保留	0x0	RO	--
[0]	SWTRIG	0x0	RW	设置此位将触发 DAC 转换，从 DAC 数据 FIFO 进行数据加载。

23.4.2.4 DAC 数据输出寄存器（DAC_DOR）

偏移地址：0x000C~0x000F 复位值：0x00000000

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

图表 23-7：DAC 数据输出寄存器（DAC_DOR）

从 DAC_DOR 读取将返回 DAC 转换后的数据。

23.4.2.5 DAC FIFO 状态寄存器（DAC_FSR）

偏移地址：0x0010~0x0013 复位值：0x10000000

31	30	29	28	27	26	25	24
LD_DONE	OVF	UDF	SER	保留			FCNT[1:0]
rw	rw	rw	ro	ro			
23	22	21	20	19	18	17	16
保留							
ro							
15	14	13	12	11	10	9	8
保留							
ro							
7	6	5	4	3	2	1	0
保留							
ro							

图表 23-8：DACFIFO 状态寄存器（DAC_FSR）

比特位	名称	复位值	读写属性	功能说明
[31]	LD_DONE	0x0	RW	加载完成标志 0 = DAC 数据加载未完成 1 = DAC 数据加载完成
[30]	OVF	0x0	RW	FIFO 溢出标志 0 = FIFO 溢出未发生 1 = FIFO 溢出发生
[29]	UDF	0x0	RW	FIFO 下溢标志 0 = FIFO 下溢未发生 1 = FIFO 下溢发生
[28]	SER	0x1	RO	FIFO 服务阈值标志 0 = FIFO 数据数大于阈值 1 = FIFO 数据数小于或等于阈值
[27:26]	保留	0x0	RO	--
[25:24]	FCNT[1:0]	0x0	RO	FIFO 数据计数器 显示 FIFO 数据计数值
[23:0]	保留	0x0	RO	--

23.4.2.6 DAC 触发寄存器（DAC_TRIMR）

偏移地址：0x0014~0x0017 复位值：0x01000410

31	30	29	28	27	26	25	24
保留							EXT_VREF_EN
rw	rw	rw	rw	rw	rw	rw	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							HD_EN
rw	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TEST_BIAS	保留				BIAS_TRIM[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRIM_VOS_EN	保留		TRIM_VOS[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw

图表 23-9：DAC 触发寄存器（DAC_TRIMR）

注意：bit0-15 不对用户开放，只做测试。

比特位	名称	复位值	读写属性	功能说明
[31:25]	保留	0x0	RO	--
[24]	EXT_VREF_EN	0x1	RW	外部参考电压 Vref 使能 0 = 禁止；1 = 使能
[23:17]	保留	0x0	RO	--
[16]	HD_EN	0x0	RW	高驱动使能 0 = 禁止；1 = 使能
[15]	TEST_BIAS	0x0	RW	偏置电流输出使能 0 = 禁止；1 = 使能
[14:11]	保留	0x0	RO	--
[10:8]	BIAS_TRIM[2:0]	0x4	RW	偏置电流触发
[7]	TRIM_VOS_EN	0x0	RW	电压偏移触发使能 0 = 禁止；1 = 使能
[6:5]	保留	0x0	RO	--
[4:0]	TRIM_VOS[4:0]	0x10	RW	电压偏移触发

24 看门狗模块(WDT)

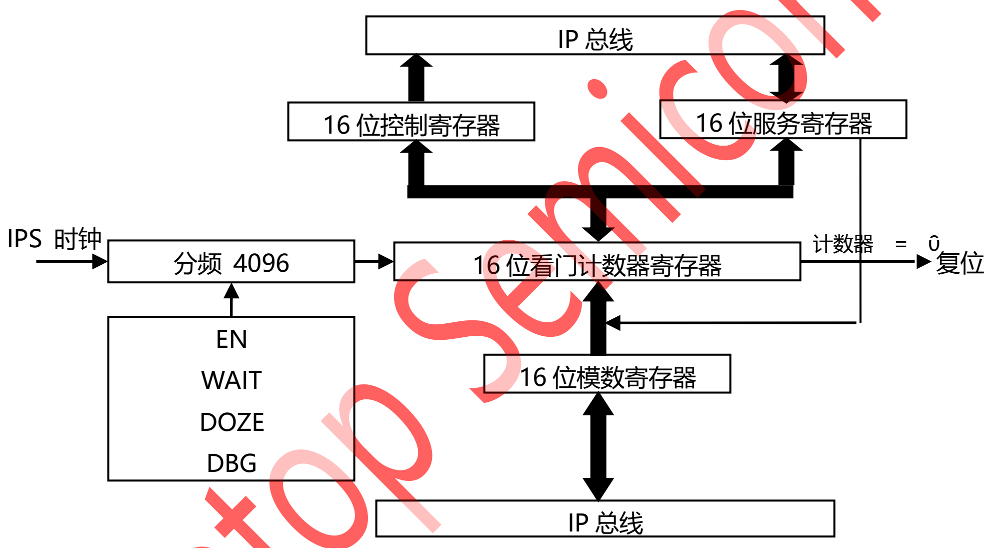
24.1 概述

看门狗模块是一个 16 位计时器，帮助软件从失控程序中恢复正常运行。看门狗模块有一个自减计数器，它会产生下溢复位。为了防止下溢复位，软件必须周期性地维护看门狗模块重新设置计数器。

24.2 特性

16 位计时器，帮助软件从失控程序中恢复正常运行。
自减计数器，它会产生下溢复位。为了防止复位，软件必须周期性地维护看门狗模块重新设置计数器。

24.3 框图



图表 24-1：看门狗模块框图

24.4 工作模式

24.4.1 等待模式

置位 WCR 寄存器的 WAIT 位，计时器进入等待模式。清除 WAIT 位，计时器继续运行。

24.4.2 瞌睡模式

置位 WCR 寄存器的 DOZE 位，计时器进入瞌睡模式。清除 DOZE 位，计时器继续运行。

24.4.3 停止模式

看门狗在停止模式下停止运行，当推出停止模式后看门狗进入停止模式之前的状态继续运行。

24.4.4 调试模式

置位 WCR 寄存器的 DBG 位，计时器进入调试模式。清除 DBG 位，计时器继续运行，在调试模式中所有更改将保留生效。

24.5 外部管脚

没有片外信号。

24.6 内存映射和寄存器

24.6.1 内存映射

看门狗模块基地址 0x40005000，寄存器如下：

表格 24-1：看门狗寄存器

偏移地址	位 15-8	位 7-0	访问权限
0x0000	看门狗控制寄存器（WCR）		S
0x0002	看门狗模数寄存器（WMR）		S
0x0004	看门狗计数寄存器（WCNTR）		S/U
0x0006	看门狗服务寄存器（WSR）		S/U

24.6.2 寄存器描述

24.6.2.1 看门狗控制寄存器(WCR)

偏移地址：0x0000~0x0001 复位值：0x0000

15	14	13	12	11	10	9	8
保留	保留	保留	保留	保留	保留	保留	保留
ro	ro	ro	ro	ro	ro	ro	ro
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留	保留	保留	保留	WAIT	DOZE	DBG	EN
ro	ro	ro	ro	rw	rw	rw	rw

图表 24-2：WCR 寄存器

比特位	名称	复位值	读写属性	功能说明
[15:4]	保留	0x0	RO	---
[3]	WAIT	0x0	RW	等待模式位 可读、一次写入控制看门狗模块处在等待模式中。 非调试模式下，重复写入此位将不会产生效果。 0 = 看门狗模块未处在等待模式下 1 = 看门狗模块处在等待模式下
[2]	DOZE	0x0	RW	**瞌睡模式位** 可读、一次写入控制看门狗模块处在瞌睡模式中。 非调试模式下，重复写入此位将不会产生效果。 0 = 看门狗模块未处在瞌睡模式下 1 = 看门狗模块处在瞌睡模式下

比特位	名称	复位值	读写属性	功能说明
[1]	DBG	0x0	RW	**调试模式** 可读、一次写入控制看门狗模块处在调试模式中。 非调试模式下，重复写入此位将不会产生效果。 在调试模式下，看门狗模块寄存器能被正常读写。 跳出调试后，看门狗模块恢复到进入调试模式之前的状态继续执行，但是在调试模式中所做的更新操作会被保留继续生效。如果一次写入寄存器在调试模式下被写入过，跳出调试模式后，依然可以写入一次。 在调试模式下，DBG 从 1 写成 0 将启动看门狗模块。DBG 从 0 写成 1 将停止看门狗模块。 0 = 看门狗模块未处在调试模式下 1 = 看门狗模块处在调试模式下
[0]	EN	0x0	RW	**看门狗使能位** 可读、一次写入控制看门狗模块使能的寄存器控制位。非调试模式下，重复写入此位将不会产生效果。当看门狗模块没有被使能的情况下，看门狗计数器和预分频寄存器将处在停止模式下。 0 = 看门狗模块关闭 1 = 看门狗模块开启

24.6.2.2 看门狗模数寄存器（WMR）

偏移地址：0x0002~0x0003　　　　　　　　　　复位值：0xFFFF

15	14	13	12	11	10	9	8
WM15	WM14	WM13	WM12	WM11	WM10	WM9	WM8
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WM7	WM6	WM5	WM4	WM3	WM2	WM1	WM0
rw	rw	rw	rw	rw	rw	rw	rw

图表 24-3：WMR 寄存器

比特位	名称	复位值	读写属性	功能说明
[15:0]	WM	0xFFFF	RW	**模数寄存器** 可读，一次性写入模数寄存器，再次写入将不会在对看门狗模块产生影响。写入 WM[15：0]新值后，新值会立即载入看门狗计数器寄存器(WCNTR)，新值也用于下一次和所有后续的重新加载。读 WMR[15:0]寄存器，返回设置的模数值。

24.6.2.3 看门狗计数寄存器（WCNTR）

偏移地址：0x0004~0x0005　　　　　　　　　　复位值：0xFFFF

15	14	13	12	11	10	9	8
WC15	WC14	WC13	WC12	WC11	WC10	WC8	WC8
ro	ro	ro	ro	ro	ro	ro	ro
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WC7	WC6	WC5	WC4	WC3	WC2	WC1	WC0
ro	ro	ro	ro	ro	ro	ro	ro

图表 24-4：WCNTR 寄存器

比特位	名称	复位值	读写属性	功能说明
[15:0]	WC	0xFFFF	RO	**计数器** 可只读寄存器，写无影响，但写操作指令执行正常结束。读取 WC[15：0]计数器寄存器返回当前值；一次读取 16 位 WC[15:0]和两次读取（一次读 WC[15:8]，一次读取 WC[7:0]）的结果不能保证一致。

24.6.2.4 看门狗服务寄存器（WSR）

偏移地址：0x0006~0x0007 复位值：0x0000

15	14	13	12	11	10	9	8
WS15	WS14	WS13	WS12	WS11	WS10	WS8	WS8
rw	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
rw	rw	rw	rw	rw	rw	rw	rw

图表 24-5：WSR 寄存器

比特位	名称	复位值	读写属性	功能说明
[15:0]	WS	0xFFFF	RW	**服务寄存器** 看门狗模块被使能后，在计数器溢出复位之前向 WSR[15：0]顺序写入 0x5555、0xAAAA 后可中断看门狗计数溢出复位。在看门狗复位之前，WSR[15:0]被顺序写入 0x5555 和 0xAAAA 之间可执行任何其它指令，但是如果写入了非 0x5555 或者 0xAAAAA 的其它任何数据，想要避免看门狗溢出引起的复位，需要重新顺序写入 0x5555 后在写入 0xAAAA。

24.7 功能描述

看门狗定时器提供了一个默认打开的硬件定时器，它可以让软件从失控程序中恢复。运行时，WDT 的 16 位减计数器自动运行，减至 0 时会产生一个复位信号，使系统恢复。该计数器软件可维护，定时维护后，计数器不减至 0，则不会产生复位信号。

24.8 中断描述

无中断信号。

25 电气特性

25.1 概述

本章节提供了该微控制器的电气特性参数和限额。

25.2 绝对最大额定值

如果施加在芯片上的载荷超过**表格 25-1** 中列出的绝对最大额定值，则可能导致芯片永久损坏。虽然芯片包含了抵抗高静态电压损坏的电路，但是不要在芯片上施加超过表格中额定的电压。这些数值只是额定值，并不意味着芯片在这些条件下功能正常。

受保证的芯片工作条件，请参考**表格 25-3**，**表格 25-4**。

表格 25-1：绝对最大额定值（工业级）

编 号	项 目	符 号	值	单 位
1	工作温度范围	T_{OPT}	-40 ~ 85	℃
2	存储温度范围	T_{STG}	-40 ~ 125	℃

25.3 静电放电（ESD）保护

表格 25-2：静电放电（ESD）保护特性

项 目	符 号	值	单 位	参 考 标 准
人体模型	HBM	2000	V	ANSI/ESDA/JEDEC JS-001-2014
带电器件模型	CDM	500	V	JEDEC EIA/JESD22-C101F
闩锁效应	LATCH UP	200	毫安（mA）	JEDEC STANDARD NO. 78D NOVEMBER 2011

25.4 静态特性

表格 25-3：IO 静态特性（3.3V）

项 目	符 号	最小值	典型值	最大值	单 位
IO 供电电压	VDD33	2.97	3.3	3.63	V
输入高电平电压	V_{IH}	2	-	VDD33	V
输入低电平电压	V_{IL}	0	-	0.8	V
输出高电平电压	V_{OH}	2.4	-	VDD33	V
输出低电平电压	V_{OL}	0	-	0.4	V
输入漏电流	I_{IN}	-	-	1	uA
输入上拉电阻	RPU	9	-	19.4	KΩ
输入下拉电阻	RPD	6.7	-	16	KΩ

表格 25-4：芯片电压特性

项 目	符 号	最小值	典型值	最大值	单 位
芯片供电电压输入	VCC35	2.97	3.3/5	5.5	V
芯片 IO/模拟电压输出	VDD33	2.97	3.3	3.63	V
芯片 ADC/DAC/USBPHY 电压输入	AVDD	2.97	3.3	3.63	V
芯片核心电压输出	LDO3_V12	1.1	1.2	1.25	V
芯片 EFLASH 电压输出	LDO1_V11	0.99	1.1	1.21	V
USBPHY 核心电压输出	LDO2_V11	0.99	1.1	1.21	V
RTC 电压输入	VBAT	2.97	3.3	3.63	V

表格 25-5：芯片电流特性 (1) (2)

项 目	符 号	最小值	典型值	最大值	单 位
低耗模式电流	I_{LP}	-	60	-	uA
保持模式电流	I_{RET}	-	8	-	uA
深度睡眠模式电流	I_{DS}	-	6	-	uA
休眠模式电流	I_{HIBER}	-	0.3	-	uA
运行模式电流	I_{RUN}	-	15	-	mA
RTC 域电流	I_{RTC}	-	3	-	uA

通过特性分析确定，未经生产测试。

电流测试条件均为常温 25 摄氏度。运行模式电流测试时，M4 工作频率为 120MHz，对应通讯接口动作，内部运行 coremark，没用到的模块时钟关闭，没用到的 IO 配置为输入。

表格 25-6：芯片时间特性 (1) (2)

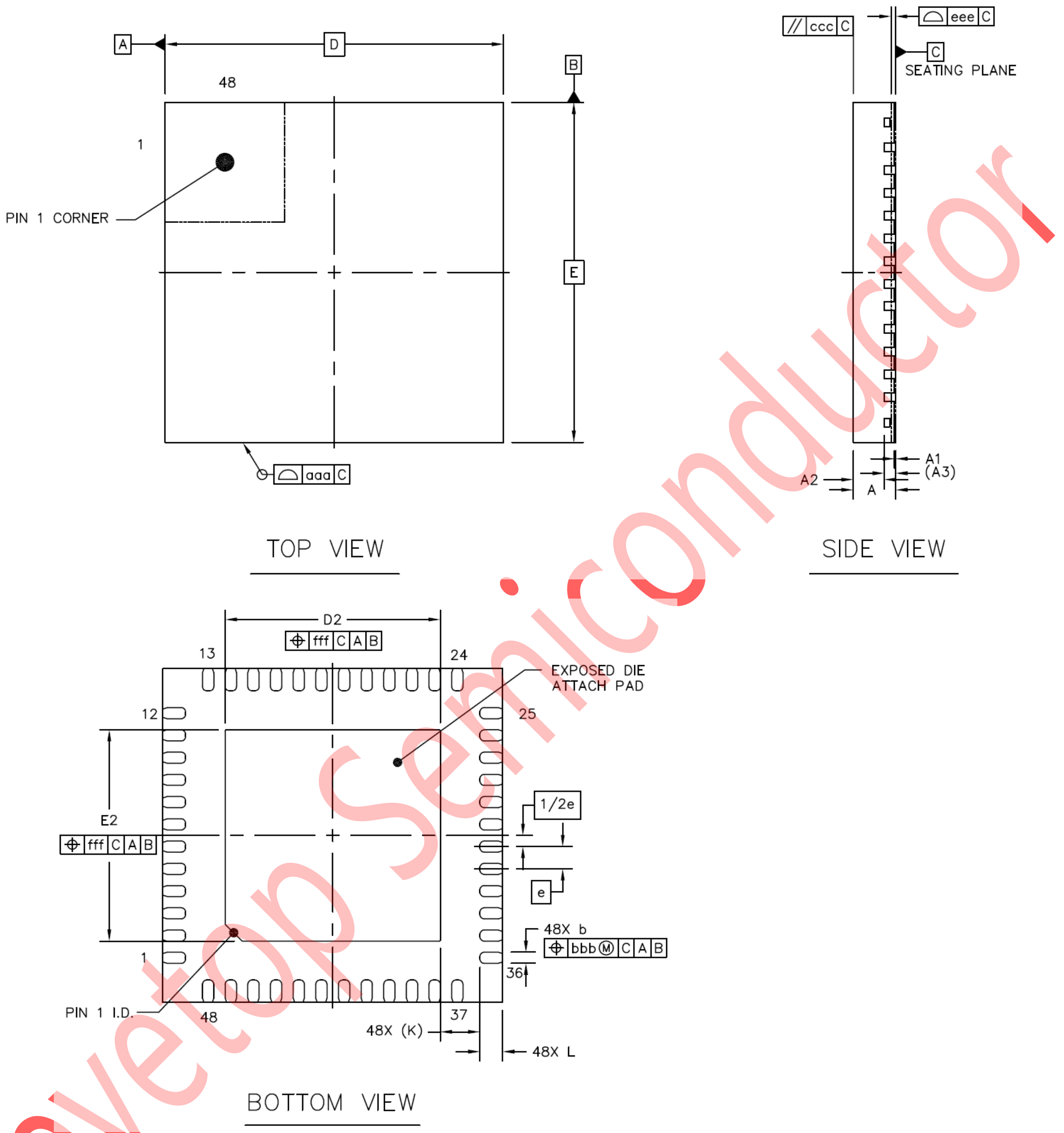
项 目	符 号	最小值	典型值	最大值	单 位
上电复位时间	T_{POR}	-	800	-	uS
低耗模式唤醒时间	T_{LP}	-	66	-	uS
保持模式唤醒时间	T_{RET}	-	100	-	uS
深度睡眠模式唤醒时间	T_{DS}	-	100	-	uS
休眠模式唤醒时间	T_{HIBER}	-	660	-	uS

通过特性分析确定，未经生产测试。

上电复位时间的测量从芯片供电电压 VCC5V 达到 POR 复位释放点开始，到应用程序代码读取第一条指令为止。唤醒时间的测量从触发唤醒事件开始，到应用程序代码读取第一条指令为止。

26 封装信息

26.1 LT268B/269 封装（QFN-48pin）



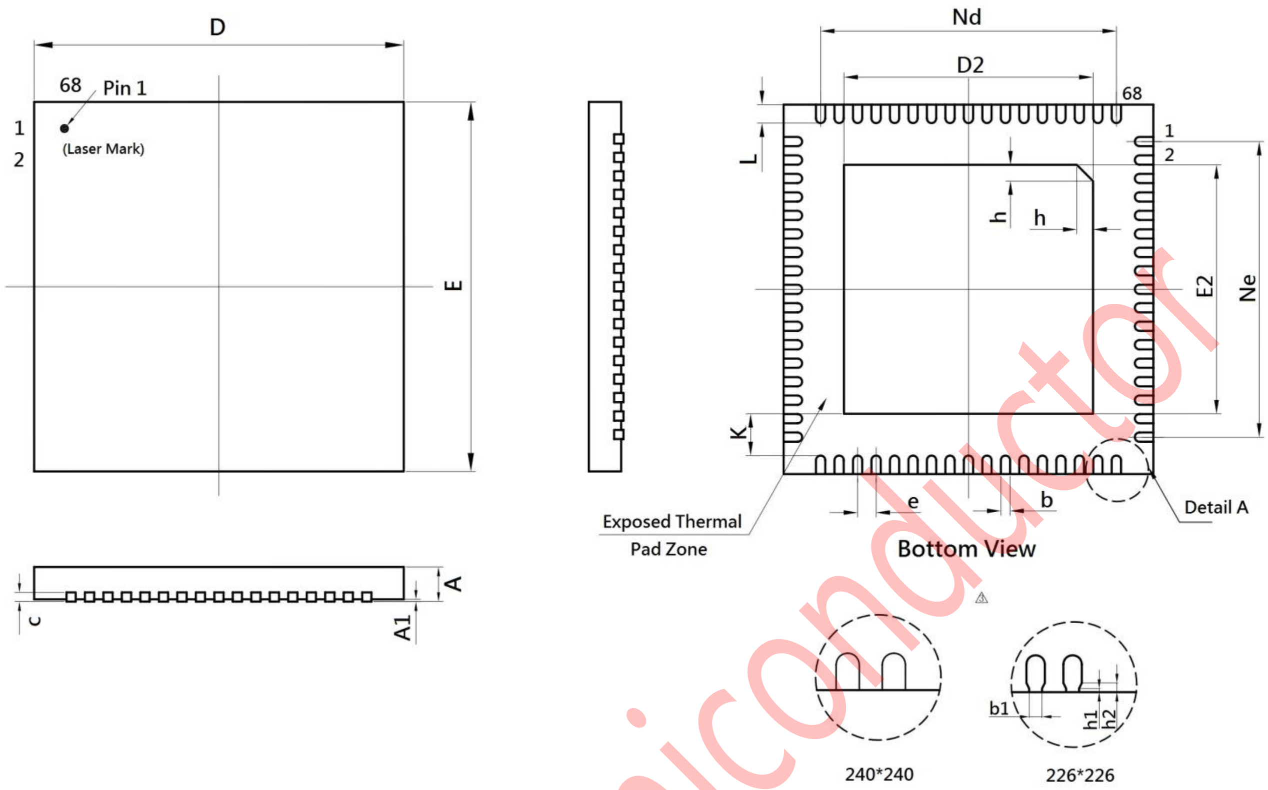
图表 26-1：LT268B/269 外观尺寸图

表格 26-1：LT268B/269 尺寸参数

		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.15	0.2	0.25
BODY SIZE	X	D	6 BSC		
	Y	E	6 BSC		
LEAD PITCH		e	0.4 BSC		
EP SIZE	X	D2	3.7	3.8	3.9
	Y	E2	3.7	3.8	3.9
LEAD LENGTH		L	0.3	0.4	0.5
LEAD TIP TO EXPOSED PAD EDGE		K	0.7 REF		
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		ccc	0.1		
COPLANARITY		eee	0.08		
LEAD OFFSET		bbb	0.07		
EXPOSED PAD OFFSET		fff	0.1		

提示：PCB 布局时，LT268B/269 背部的散热焊盘（Thermal Pad Zone）必须直接接地。

26.2 LT268C 封装（QFN-68pin）



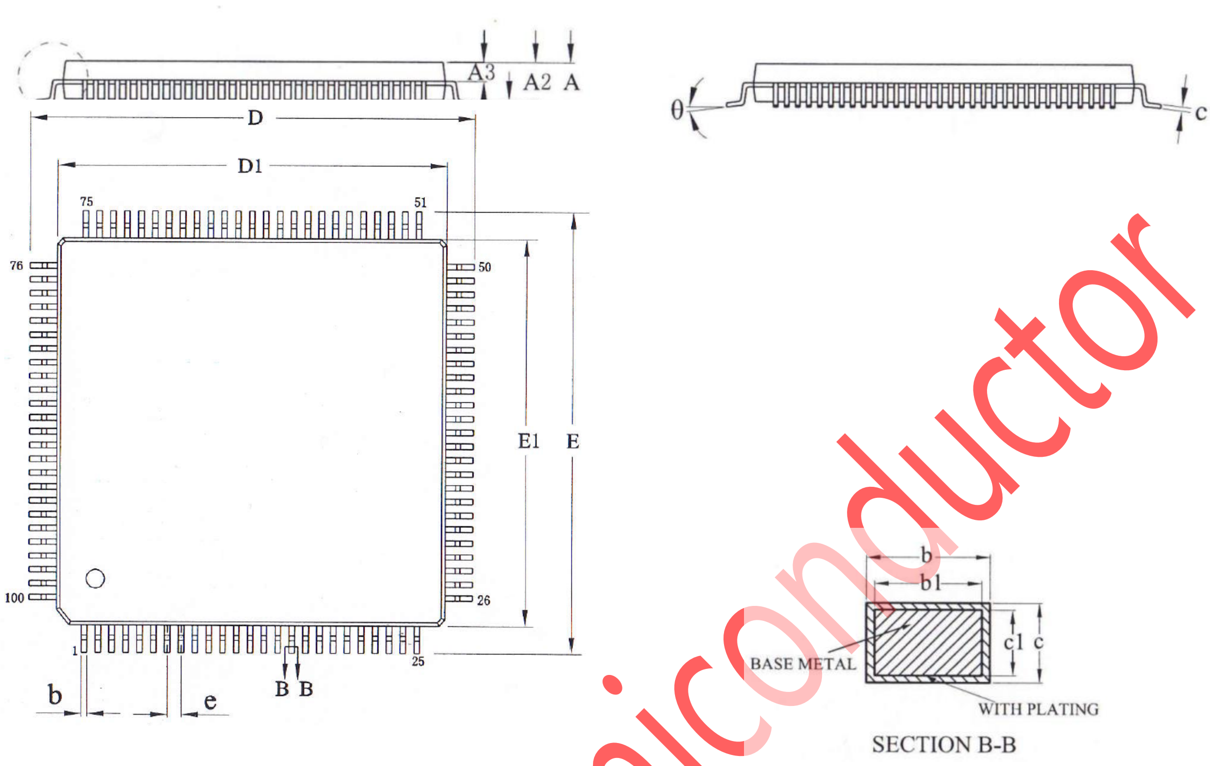
图表 26-2：LT268C 外观尺寸图

提示：PCB 布局时，LT268C 背部的散热焊盘（Thermal Pad Zone）必须直接接地。

表格 26-2：LT268C 尺寸参数

Symbol	Millimeter			Symbol	Millimeter		
	Min.	Nom.	Max		Min.	Nom.	Max
A	0.70	0.75	0.8	Ne	6.40BSC		
A1	-	0.02	0.05	L	0.35	0.40	0.45
b	0.15	0.20	0.25	K	0.20	-	-
b1	0.14REF			h	0.30	0.35	0.40
c	0.18	0.20	0.25	h1	0.04REF		
D	7.90	8.00	8.10	h2	0.10REF		
e	0.40BSC			D2	5.39	5.49	5.59
Nd	6.40BSC			E2	5.39	5.49	5.59
E	7.9	8.0	8.10				

26.3 LT268D 封装（LQFP-100pin）



图表 26-3：LT268D 外观尺寸图

表格 26-3：LT268D 尺寸参数

Symbol	Millimeter			Symbol	Millimeter		
	Min.	Nom.	Max		Min.	Nom.	Max
A	-	-	1.60	D1	13.9	14.0	14.1
A1	0.05	-	0.15	E	15.8	16.0	16.2
A2	1.35	1.40	1.45	E1	13.9	14.0	14.1
A3	0.59	0.64	0.69	eB	15.05	-	15.35
b	0.18	-	0.26	e	0.50BSC		
b1	0.17	0.20	0.23	L	0.45	-	0.75
c	0.13	-	0.17	L1	1.00REF		
c1	0.12	0.13	0.14	θ	0		7
D	15.8	16.00	16.2				

27 版本记录

表格 27-1：规格书版本记录

版 别	发 布 日 期	改 版 说 明
V1.0	2022/02/08	Preliminary Release

28 版权说明