Hi3861V100/Hi3861LV100/Hi3881V100 WiFi 芯片

用户指南

文档版本 04

发布日期 2020-07-29

商标声明

 、HISILICON、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

上海海思技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：http://www.hisilicon.com/cn/

客户服务邮箱：support@hisilicon.com

目 录

插图目录

表格目录

目 录

前　言

概述

本文档主要介绍 Hi3861V100、Hi3861LV100、Hi3881V100 芯片的各项基本功能，常见应用模式的配置方法。

本文档提供 Hi3861V100、Hi3861LV100、Hi3881V100 芯片的应用配置方法。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3861芯片	V100
Hi3861L 芯片	V100
Hi3881 芯片	V100

读者对象

本文档主要适用于以下工程师：

- 技术支持工程师
- 客户开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危

符号	说明
	害。
警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

寄存器访问类型约定

类型	说明	类型	说明
RO	只读，不可写。	RW	可读可写。
RC	读清零。	WC	可读，写 1 清零，写 0 保持不变。

寄存器复位值约定

在寄存器定义表格中：

- 如果某一个比特的复位值“Reset”（即“Reset”行）为“？”，表示复位值不确定。
- 如果某一个或者多个比特的复位值“Reset”为“？”，则整个寄存器的复位值“Total Reset Value”为“-”，表示复位值不确定。

数值单位约定

数据容量、频率、数据速率等的表达方式说明如下。

类别	符号	对应的数值
数据容量（如 RAM 容量）	1K	1024
	1M	1,048,576

类别	符号	对应的数值
	1G	1,073,741,824
频率、数据速率等	1k	1000
	1M	1,000,000
	1G	1,000,000,000

地址、数据的表达方式说明如下。

符号	举例	说明
0x	0xFE04、0x18	用 16 进制表示的数据值、地址值。
0b	0b000、0b00 00000000	表示 2 进制的数据值以及 2 进制序列（寄存器描述中除外）。
X	00X、1XX	在数据的表达方式中，X 表示 0 或 1。 例如：00X 表示 000 或 001； 1XX 表示 100、101、110 或 111。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 04 (2020-07-29)

删除“6.10.2 功能描述”中芯片进行增量修改的内容和须知内容。

删除“6.10.3 睡眠唤醒控制”小节。

删除“6.10.4 地址重映射”小节。

删除“6.10.5 公共访问寄存器”小节。

文档版本 03 (2020-06-28)

更新“6.10.1 概述”中 SDIO 符合的工业标准规格。

更新“6.10.2 功能描述”中 SDIO 支持的工业标准规格。

文档版本 02 (2020-05-30)

在“6 外围设备”章节的“6.10 SDIO”小节的功能描述中新增关于对接 Host 芯片的约束说明。

文档版本 01 (2020-04-30)

第一次正式版本发布。

1 产品概述

1.1 概述

Hi3861/Hi3881 是一款高度集成的 2.4GHz WiFi 芯片，集成 IEEE 802.11b/g/n 基带和 RF（Radio Frequency）电路，包括功率放大器 PA（Power Amplifier）、低噪声放大器 LNA（Low Noise Amplifier）、RF balun、天线开关以及电源管理模块等。

Hi3861/Hi3881 WiFi 基带实现正交频分复用（OFDM）技术，并向下兼容直接序列扩频（DSSS）、补码键控（CCK）技术，支持 IEEE 802.11b/g/n 协议。支持 20MHz 标准带宽和 5MHz/10MHz 窄带宽，提供最大 72.2Mbit/s 物理层速率。

Hi3861/Hi3881 芯片集成高性能 32bit 微处理器；提供 SPI（Synchronous Peripheral Interface）、UART（Universal Asynchronous Receiver & Transmitter）、I2C（The Inter-Integrated Circuit）、I2S（Inter-IC Sound）、PWM（Pulse-Width Modulation）、GPIO（General Purpose Input/Output）以及多路 ADC（Analog to Digital Converter）模拟输入等丰富的外设接口，同时支持 SDIO2.0（Secure Digital Input/Output）接口，时钟最高支持 50MHz；支持 Huawei LiteOS 和第三方组件，并配套提供开放、易用的开发和调试环境。

Hi3861 系列的产品型号包括 Hi3861V100、Hi3861LV100。芯片内置 SRAM（Static Random Access Memory）和 Flash，可独立运行，并支持在 Flash 上运行程序。

Hi3881 系列产品型号包括 Hi3881V100。芯片作为从机，通过 SDIO 接口连接到主芯片运行。

1.2 功能描述

1.2.1 功能特性

Hi3861/Hi3881 的功能特点如下：

- 通用规格
  - 1×1 2.4GHz 频段（ch1～ch14）
  - PHY（Physical Layer）支持 IEEE 802.11b/g/n
  - MAC（Media Access Control）支持 IEEE802.11 d/e/h/i/k/v/w

- 内置 PA 和 LNA，集成 TX/RX Switch、Balun 等
- 支持 STA（Station）和 AP 形态，作为 AP 时最大支持 6 个 STA
- 支持 WFA WPA/WPA2 personal、WPS2.0
- 支持与 BT（Bluetooth）/BLE（Bluetooth Low Energy）芯片共存的 2/3/4 线 PTA（Packet Traffic Arbitration）方案
- 支持 RF 自校准方案
- 电源电压输入范围：2.3V～3.6V
- IO 电源电压支持 1.8V 和 3.3V
- Hi3861LV100 功耗：

  在环境温度 25℃条件下测试：

  Ultra Deep Sleep 模式：3μA@3.3V

  在环境温度 25℃、接收 RX 时间长度 1ms、芯片 BUCK 供电、屏蔽环境的条件下测试：

  DTIM1：0.97mA@3.6V

  DTIM3：0.36mA@3.6V

  DTIM10：0.15mA@3.6V
- Hi3861V100 功耗：

  在环境温度 25℃条件下测试：

  Ultra Deep Sleep 模式：3μA@3.3V

  在环境温度 25℃、接收 RX 时间长度 1ms、芯片 BUCK 供电、屏蔽环境的条件下测试：

  DTIM1：1.27mA@3.6V

  DTIM3：0.523mA@3.6V

  DTIM10：0.233mA@3.6V

- PHY 特性
  - 支持 IEEE802.11b/g/n 单天线所有的数据速率
  - 支持最大速率：72.2Mbps@HT20 MCS7
  - 支持标准 20MHz 带宽和 5M/10M 窄带宽
  - 支持 STBC（Space-Time Block Coding） RX
  - 支持 Short-GI（Guard Interval）
- MAC 特性
  - 支持 A-MPDU（Aggregate MAC Protocol Data Unit）
  - 支持 Blk-ACK（Acknowledgement）
  - 支持 QoS（Quality of Service），满足不同业务服务质量需求
- CPU
  - 高性能自研 32bit CPU，最大工作频率 160MHz
  - 内置 SRAM 352KB、ROM（Read-Only Memory） 288KB
  - 内嵌 2MB Flash（仅 Hi3861 支持）
  - 外围接口（通过复用实现）

Hi3861：1 个 SDIO Slave 接口、2 个 SPI 接口、2 个 I2C 接口、3 个 UART 接口、15 个 GPIO 接口、7 路 ADC 输入、6 路 PWM、1 个 I2S 接口、外接 32K 时钟（仅 Hi3861LV100 支持）

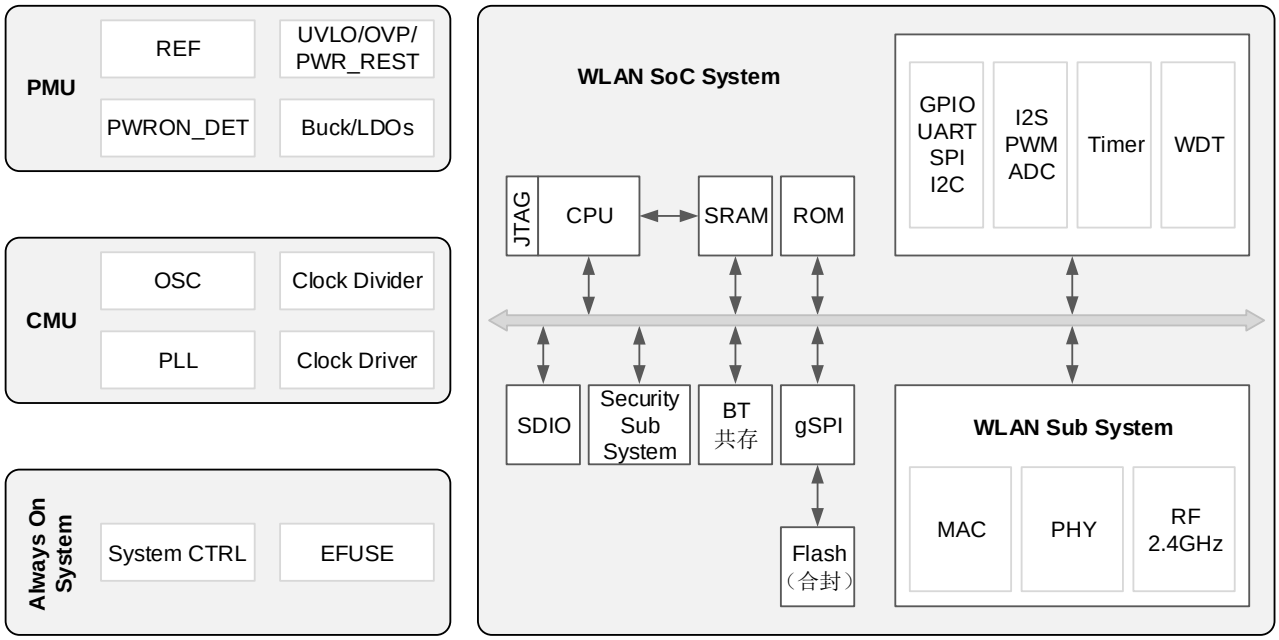
Hi3881：1 个 SDIO Slave 接口、1 个 UART 接口和 5 个 GPIO 接口

- 其他信息
  - 封装信息：QFN-32（Quad Flat Non-leaded package），5mm×5mm
  - 工作温度：-40℃～+85℃

1.3 逻辑框图

Hi3861 的逻辑框图如图 1-1 所示。

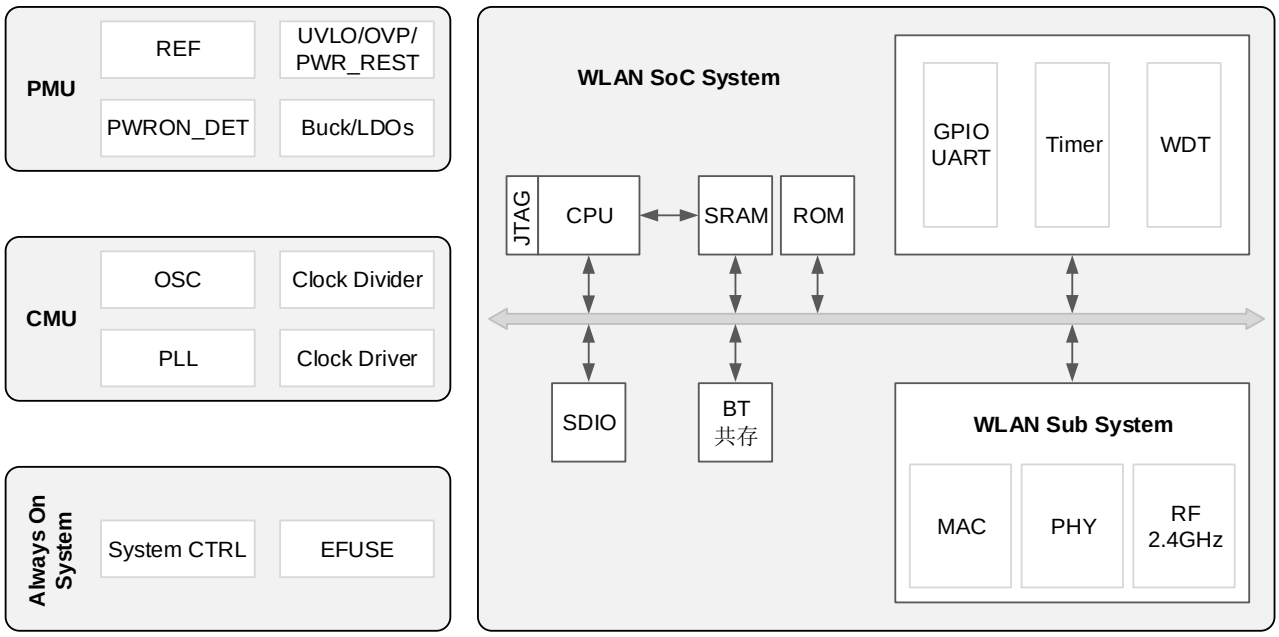
图1-1 Hi3861 逻辑框图



注：PMU（Power Management Unit），REF（Reference Indicator），UVLO（Under Voltage Lock Out），OVP（Overvoltage Protection），LDO（Low Dropout Regulator），CMU（Clock Managing Unit），OSC（Oscillator），PLL（Phase-Locked Loop），EFUSE（Electrical FUSE），JTAG（Joint Test Action Group），WDT（Watch Dog Timer）。

Hi3881 的逻辑框图如图 1-2 所示。

图1-2 Hi3881 逻辑框图



逻辑框图中各模块功能介绍如表 1-1 所示。

表1-1 逻辑框图中各模块列表

模块名	功能
PMU	电源管理单元。
CMU	时钟管理单元。
GPIO	通用 IO。
UART	通用异步串行接口控制器。
SPI	串行外设接口。
I2C	双向二线制同步串行总线。
I2S	音频总线。
ADC	模数转换器。
PWM	脉冲信号宽度调制。
Timer	定时器。
WDT	看门狗单元。
Security Sub System	安全子系统。
SRAM	静态随机存取存储器。
EFUSE	加解密和芯片 ID 存储。

模块名	功能
MAC	MAC 层业务处理。
PHY	信道调制、解调。
RF	射频模块。

1.4 典型应用场景

Hi3861V100 芯片适应于智能家电等物联网智能终端领域。

Hi3861LV100 芯片适应于智能家电、智能门锁、低功耗 Camera、BUTTON 等物联网低功耗智能产品领域。

Hi3881V100 芯片适应于 IP Camera、OTT（Over The Top）、STB、TV 等领域。

2 系统

2.1 复位

2.1.1 概述

复位模块根据输入控制信号产生各模块的复位信号，支持整个芯片全局复位和各个模块的单独复位。

复位信号异步生效，同步撤离。

2.1.2 复位控制

复位控制的输入信号分为：

- 全局复位控制
- 单独模块复位控制

芯片可以使用的复位控制方式如表 2-1 所示。

表2-1 复位控制

复位方式	来源	复位时间	复位范围	复位后芯片模式
POR11（Power On Reset）复位	1.1V POR 模块输出	200μs	全芯片，并锁存硬件控制字及 JTAG 使能信号	WORK
Watch Dog 复位控制	Watch Dog 模块输出，软件可以屏蔽	-	模块	WORK
全局软复位控制	软件配置	-	全芯片	WORK
子系统软复位控制	软件配置	-	CPU 对应子系统	WORK

复位方式	来源	复位时间	复位范围	复位后芯片模式
模块软复位控制	CRG（Clock and Reset Generator）寄存器，软件配置	软件控制	模块	保持
模块硬复位控制	芯片模块，软件可以屏蔽	-	模块	保持

2.2 时钟

2.2.1 概述

时钟管理模块对芯片时钟输入、生成、控制进行统一的管理，其功能包括：

- 时钟输入的管理和控制
- 时钟分频和控制
- 各模块工作时钟的生成

2.2.2 时钟分配

表2-2 正常工作时主要时钟分配

模块名称	时钟频率（MHz）	模块名称	时钟频率（MHz）
自研 CPU	160	SSS（Security Sub System）	160
CPU BUS	160	GPIO	晶体/0.032
TOP BUS	80	SFC（Serial Peripheral Interface Flash Controller）	96（根据外接器件可调）
Timer	晶体分频	UART	160/80
RTC（Real-Time Clock）	晶体/0.032	I2C	80
WatchDog	80	SPI 控制器	160
PWM	160/晶体	EFUSE	晶体
MAC	80	PMU OSC	0.032
PHY	160/80/40/20	晶体	40/24
ADC	80	DMA（Direct Memory Access）	80

模块名称	时钟频率（MHz）	模块名称	时钟频率（MHz）
DAC（Digital Analog Converter）	160	SDIO	50
AUDIO_AIAO	12.288	-	-

2.2.3 时钟控制

时钟管理模块主体包括：

- PLL 频率控制
- 时钟分频和时钟源选择控制
- 时钟门控管理

2.3 电源管理与低功耗模式控制

2.3.1 概述

芯片的低功耗模式用于有效减少芯片的功耗，芯片提供多种低功耗的控制来动态降低芯片的功耗：

- 系统工作模式控制

  除了 Work 模式之外，各种模式对功耗都有一定的减少作用，可以根据实际的功耗要求和功能要求选择不同的工作模式。

- 时钟门控和时钟频率调整

  提供时钟关断功能，可以关闭没有必要的时钟，减少芯片的功耗。

  系统工作的时钟频率可以进行调整，在满足功能的情况下可以调节时钟频率，动态降低芯片功耗。

- 模块级低功耗控制

  提供模块级的低功耗控制，可以在某模块不工作的情况下，关断该模块或使模块处于低功耗状态，以减少芯片的功耗。

须知

- Hi3861V100 支持所有模式，无 32K 外灌时钟。
- Hi3861LV100 支持所有模式，有 32K 外灌时钟。
- Hi3881V100 无深睡模式 Deep Sleep 及超深睡模式 Ultra Deep Sleep，无 32K 外灌时钟。

2.3.2 系统工作模式

系统工作模式分为以下五种模式：

- Work

  正常工作状态，所有电源供电均打开，完成正常的 WiFi 收发等业务。
- Light Sleep

  浅睡模式为可快速恢复业务收发的睡眠模式，关闭收发时钟以降低功耗。此时 CPU 配置为 WFI（Wait For Interrupt）模式，等待中断唤醒后恢复收发，Flash、IO 保持供电（VDDIO），SDIO 由软件控制是否掉电，可通过 SDIO 唤醒系统（SDIO 中断）；SoC 系统可选晶体或 PLL 时钟，时钟频率和外设的通信速率有关。
- Deep Sleep

  深睡模式会关闭大部分业务使用电源以降低功耗。保留 GPIO、RTC、外置 TSF（Timing Synchronization Function）模块，可从深睡中唤醒，系统重新上电，等待 CPU 启动恢复业务收发。此时 Flash 关闭供电，IO 保持供电。该模式下 SDIO 可软件配置保留电源，可通过 SDIO 唤醒系统（SDIO 配置为非 Sleep 模式），RAM 可配置为 RETENTION。
- Ultra Deep Sleep

  超深睡模式为最低功耗模式。此时芯片关闭了大部分电源以降低功耗，RAM（Random Access Memory）均掉电。支持芯片管脚 GPIO3/5/7/14 唤醒系统，通过唤醒管脚可唤醒系统。
- Shutdown

  关机模式，芯片通过 PMU_PWRON 管脚拉低之后进入 Shutdown 模式。当 PMU_PWRON 管脚拉高后退出 Shutdowm 模式，进入到 Work 模式。

2.4 处理器子系统

系统提供一个自研处理器作为主控 CPU，完成各种系统任务和控制工作。处理器带有 32KB 指令 Cache、4KB 数据 Cache。

该芯片 CPU 具有以下功能特点：

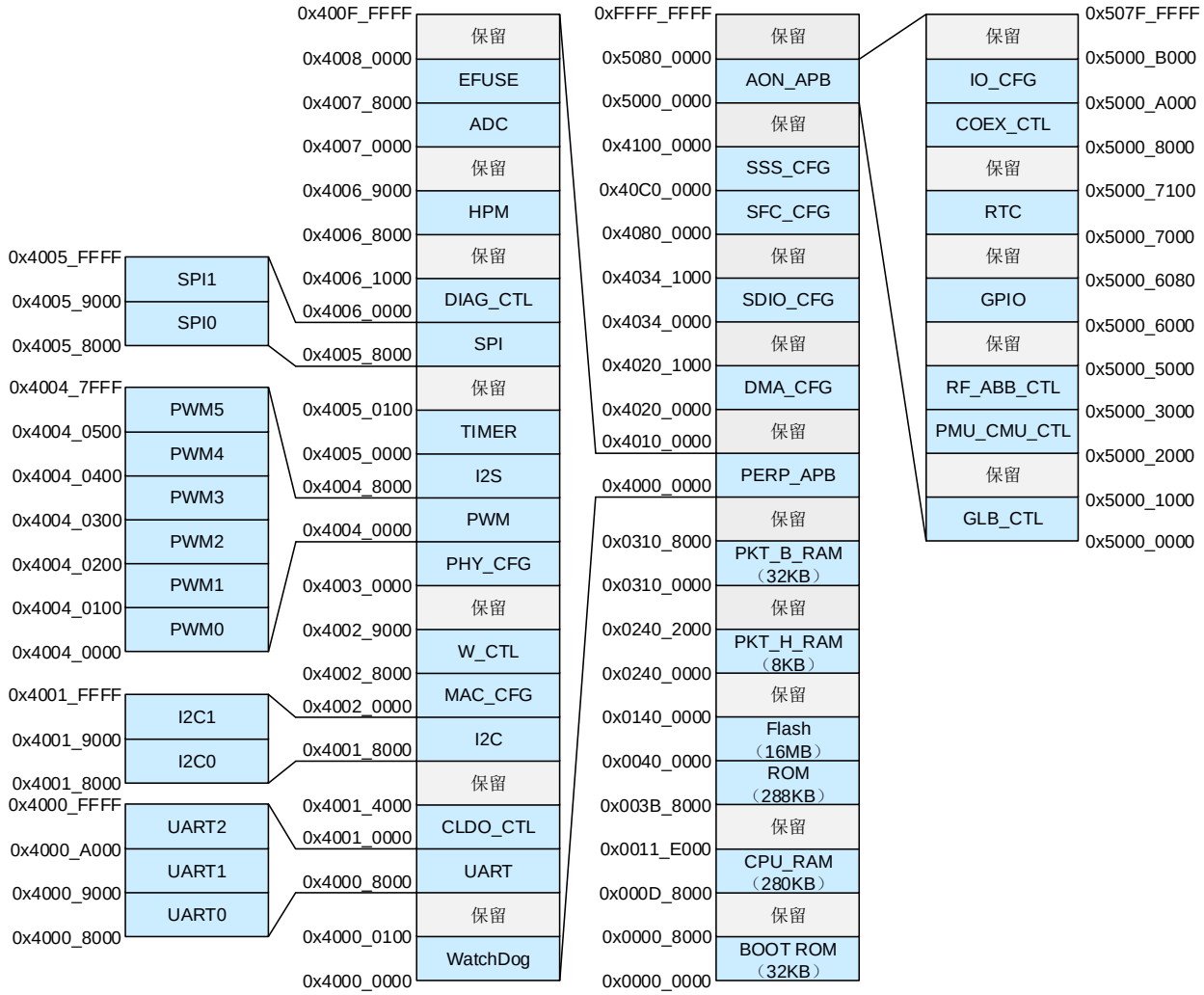
- 处理器的工作频率最高可达 160MHz。
- 支持直接模式和向量模式的中断方式，支持 35 个非标准外部中断。
- 支持边沿和电平两种中断触发方式。
- 支持 Flash Patch 功能，支持 254 个指令比较器和 2 个地址比较器。
- 支持 PMP（Physical Memory Protection）功能。

- 支持 JTAG 和 SWD（Serial Wire Debug）调试接口。

2.5 存储器空间映射

重映射时地址映射关系如图 2-1 所示。

图2-1 重映射时地址映射关系图



注：HPM（Hardware Performance Monitor）。

2.6 SYSTEM TICK

2.6.1 概述

SYSTEM TICK 为系统提供计数功能，采用不同频率的时钟，计数得到的时间值不同。

2.6.2 功能描述

SYSTEM TICK 具有以下功能特点：

- 64bit 的 SYSTEM_TICK 默认使用 POR 产生的 32K 时钟正向计数，用于记录系统时间。
- 软件可清除，软件可以锁定读取当前的计时值，也可以实时读取当前的计时值。
- 仅有 POR 上电复位，芯片的全局复位以及软件复位可以清零。

2.7 中断系统

2.7.1 中断分配

Hi3861、Hi3861L、Hi3881 芯片支持向量模式和直接模式的中断方式，支持边沿和电平两种中断触发方式。支持优先级可编程，优先级配置寄存器（共 3bit）可配置 8 级的优先级。

中断系统包括：

- CPU 的内部标准中断：中断编号 0～25。
- CPU 外部的非标准中断：所支持的非标准中断编号如表 2-3 所示。

表2-3 非标准中断编号列表

中断编号	中断描述	中断编号	中断描述
26	Timer0 中断	44	SPI1 中断
27	Timer1 中断	45	GPIO 中断
28	Timer2 中断	46	Tsensor 中断
29	RTC0 中断	47	wlan_sleep 中断
30	RTC1 中断	48	wlan_wakeup 中断
31	RTC2 中断	49	over_temp 中断
32	RTC3 中断	50	pmu_cmu_err 中断
33	WDT 中断	51	软中断 0
34	MAC 中断	52	软中断 1
35	DMA 中断	53	软中断 2
36	SFC 中断	54	软中断 3
37	SDIO 中断	55	PEK 中断

中断编号	中断描述	中断编号	中断描述
38	UART0 中断	56	AES（Advanced Encryption Standard）/SHA256（Secure Hash Algorithm）/KDF（Key Derivation Function）中断
39	UART1 中断	57	TRNG（True Random Number Generator）中断
40	UART2 中断	58	LSADC（Low-Speed Analog-to-Digital Converter）中断
41	I2C0 中断	59	I2S0 中断
42	I2C1 中断	60	udsleep 中断
43	SPI0 中断	-	-

注：CPU 的 Machine timer 中断使用 Timer3 的中断源，Timer3 不可被用作其他用途。

2.7.2 中断结构

说明

- Hi3861、Hi3861L、Hi3881 芯片使用 CPU 内部集成的中断控制器，所有的外设或寄存器触发的非标准中断均直接连至 CPU 内部进行处理。
- 本节主要介绍非 IP 中断，各个 IP 的中断说明请参见各 IP 对应章节。

2.7.2.1 NMI 中断

NMI（Non-Maskable Interrupt）中断用于控制软件和逻辑对 EFUSE 特定区域的烧写，由寄存器触发，通过写 TEE_NMI_INT[tee_nmi_int]为 1 触发 NMI 中断。

须知

TEE_NMI_INT[tee_nmi_int]不会自动清零，在触发中断后需要将该值写回 0。在触发 NMI 中断后 CPU 会告知 EFUSE 控制逻辑此时 CPU 处于 NMI 中断模式，仅在此状态下才可以烧写 EFUSE 的特定区域，使用方法请参见“5.7 EFUSE”。

2.7.2.2 软中断

软中断为通过配置寄存器触发中断的中断触发方式，包含 4 个可以通过写寄存器触发的中断（软中断 0～软中断 3）。

软中断 0～软中断 3 的触发方式相同，以使用软中断 0 为例，配置步骤如下：

1. 写 SOFT_INT_EN[soft_int0_en]为 1，打开软中断 0 的使能。
2. 写 SOFT_INT_SET[soft_int0_set]为 1，将软中断 0 置位。

   此寄存器为自清零寄存器，写 1 后自动回零，写 0 无效。

3. 软件进入软中断 0 处理程序，此时可读取 SOFT_INT_STS[soft_int0_sts]查询中断状态：
   - 0：无中断。
   - 1：有中断。
4. 写 SOFT_INT_CLR[soft_int0_clr]为 1，清除软中断 0。

   此寄存器为自清零寄存器，写 1 后自动回零，写 0 无效。

----结束

2.7.3 寄存器概览

中断系统寄存器概览如表 2-4 所示。

表2-4 中断系统寄存器概览

绝对地址	名称	描述	页码
0x40010258	TEE_NMI_INT	NMI 中断控制寄存器	2-8
0x50000280	SOFT_INT_EN	软中断使能寄存器	2-8
0x50000284	SOFT_INT_SET	软中断置位寄存器	2-9
0x50000288	SOFT_INT_CLR	软中断清零寄存器	2-10
0x5000028C	SOFT_INT_STS	软中断状态查询寄存器	2-10

2.7.4 寄存器描述

TEE_NMI_INT

TEE_NMI_INT 为 NMI 中断控制寄存器。

Absolute Address: 0x40010258　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	RO	reserved	reserved	0x0000
[0]	RW	tee_nmi_int	NMI 中断控制。 0：CPU NMI 中断拉低； 1：CPU NMI 中断拉高。	0x0

SOFT_INT_EN

SOFT_INT_EN 为软中断使能寄存器。

Absolute Address: 0x50000280 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	保留。	0x000
[3]	RW	soft_int3_en	软中断 3 使能。 0：禁止； 1：使能。	0x0
[2]	RW	soft_int2_en	软中断 2 使能。 0：禁止； 1：使能。	0x0
[1]	RW	soft_int1_en	软中断 1 使能。 0：禁止； 1：使能。	0x0
[0]	RW	soft_int0_en	软中断 0 使能。 0：禁止； 1：使能。	0x0

SOFT_INT_SET

SOFT_INT_SET 为软中断置位寄存器。

Absolute Address: 0x50000284 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	保留。	0x000
[3]	W1_PULSE	soft_int3_set	软中断 3 置位。 写 0：无效； 写 1：置中断。	0x0
[2]	W1_PULSE	soft_int2_set	软中断 2 置位。 写 0：无效； 写 1：置中断。	0x0
[1]	W1_PULSE	soft_int1_set	软中断 1 置位。 写 0：无效； 写 1：置中断。	0x0
[0]	W1_PULSE	soft_int0_set	软中断 0 置位。 写 0：无效； 写 1：置中断。	0x0

SOFT_INT_CLR

SOFT_INT_CLR 为软中断清零寄存器。

Absolute Address: 0x50000288　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	保留。	0x000
[3]	W1_PULSE	soft_int3_clr	软中断 3 清零。 写 0：无效； 写 1：清中断。	0x0
[2]	W1_PULSE	soft_int2_clr	软中断 2 清零。 写 0：无效； 写 1：清中断。	0x0
[1]	W1_PULSE	soft_int1_clr	软中断 1 清零。 写 0：无效； 写 1：清中断。	0x0
[0]	W1_PULSE	soft_int0_clr	软中断 0 清零。 写 0：无效； 写 1：清中断。	0x0

SOFT_INT_STS

SOFT_INT_STS 为软中断状态查询寄存器。

Absolute Address: 0x5000028C　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	保留。	0x000
[3]	RO	soft_int3_sts	软中断 3 状态查询。 0：无中断； 1：有中断。	0x0
[2]	RO	soft_int2_sts	软中断 2 状态查询。 0：无中断； 1：有中断。	0x0
[1]	RO	soft_int1_sts	软中断 1 状态查询。 0：无中断；	0x0

			1：有中断。	
[0]	RO	soft_int0_sts	软中断 0 状态查询。 0：无中断； 1：有中断。	0x0

2.8 RTC

2.8.1 概述

RTC 的功能是从预置的数据开始递减直至 0，产生中断。RTC 的初始数据、工作模式、中断方式均可以通过寄存器配置，且 RTC 可以独立进行时钟使能，以及灵活指定时钟间隔。芯片中有 4 个相同且可独立配置的 RTC。

2.8.2 功能描述

RTC 具有以下功能特点：

- 4 个 32bit 可独立配置的 RTC 单元。
- 支持自由模式和周期模式。
- 每个 RTC 单元可以独立进行使能。
- 每个 RTC 均配置有独立的中断。
- 支持锁定当前的计数值。

2.8.3 工作方式

RTC 工作时钟频率可选择 32KHz 或晶体时钟（40/24MHz）。

RTC 支持自由模式和周期模式，每个 RTC 均可单独配置：

- 在自由模式下，定时器从 0xFFFF_FFFF 递减计数到 0。
- 在周期模式下，计数器从寄存器 TIMERx_LOADCOUNT（x=1～4）的值计数到 0。

以 RTC3、RTC4 为例说明 RTC 的工作流程，具体步骤如下：

1. 通过配置 TIMER3_CONTROLREG 和 TIMER4_CONTROLREG 初始化 RTC。
   - 写 TIMER3_CONTROLREG[timer3_en]、TIMER4_CONTROLREG[timer4_en]为 0，关闭 RTC。

     注意：为了避免不同步问题，在设置 TIMER3_LOADCOUNT、TIMER4_LOADCOUNT 之前必须将 RTC 关闭。
   - 写 TIMER3_CONTROLREG[timer3_mode]、TIMER4_CONTROLREG[timer4_mode]为 0，配置 RTC 为自由模式；或写 1，配置 RTC 为周期模式。

注意：在配置 RTC 模式为自由模式工作之前，必须设置 TIMER3_LOADCOUNT、TIMER4_LOADCOUNT 的所有有效位都为 1。

- 写 TIMER3_CONTROLREG[timer3_int_mask]、TIMER4_CONTROLREG[timer4_int_mask]为 1，设置中断屏蔽为屏蔽方式；或写 0，设置中断屏蔽为不屏蔽方式。

2. 向 TIMER3_LOADCOUNT、TIMER4_LOADCOUNT 写入 RTC 的初始值（配置模式为周期模式）。
3. 写 TIMER3_CONTROLREG[timer3_en]、TIMER4_CONTROLREG[timer4_en]为 1，使能 RTC，开始倒计时。

----结束

2.8.4 寄存器概览

RTC 寄存器概览如表 2-5 所示。

表2-5 RTC 寄存器概览（基址是 0x5000_7000）

偏移地址	名称	描述	页码
0x000	TIMER1_LOADCOUNT	定时器 1 的初始值寄存器	2-13
0x004	TIMER1_CURRENTVALUE	定时器 1 的当前值寄存器	2-13
0x008	TIMER1_CONTROLREG	定时器 1 的控制寄存器	2-13
0x00C	TIMER1_EOI	定时器 1 的清中断寄存器	2-14
0x010	TIMER1_INTSTATUS	定时器 1 的中断状态寄存器	2-14
0x014	TIMER2_LOADCOUNT	定时器 2 的初始值寄存器	2-15
0x018	TIMER2_CURRENTVALUE	定时器 2 的当前值寄存器	2-15
0x01C	TIMER2_CONTROLREG	定时器 2 的控制寄存器	2-15
0x020	TIMER2_EOI	定时器 2 的清中断寄存器	2-16
0x024	TIMER2_INTSTATUS	定时器 2 的中断状态寄存器	2-16
0x028	TIMER3_LOADCOUNT	定时器 3 的初始值寄存器	2-16
0x02C	TIMER3_CURRENTVALUE	定时器 3 的当前值寄存器	2-16
0x030	TIMER3_CONTROLREG	定时器 3 的控制寄存器	2-17
0x034	TIMER3_EOI	定时器 3 的清中断寄存器	2-17
0x038	TIMER3_INTSTATUS	定时器 3 的中断状态寄存器	2-18
0x03C	TIMER4_LOADCOUNT	定时器 4 的初始值寄存器	2-18
0x040	TIMER4_CURRENTVALUE	定时器 4 的当前值寄存器	2-18

偏移地址	名称	描述	页码
0x044	TIMER4_CONTROLREG	定时器 4 的控制寄存器	2-18
0x048	TIMER4_EOI	定时器 4 的清中断寄存器	2-19
0x04C	TIMER4_INTSTATUS	定时器 4 的中断状态寄存器	2-19
0x0A0	TIMERS_INTSTATUS	定时器中断状态寄存器	2-19
0x0A4	TIMERS_EOI	定时器清中断寄存器	2-20
0x0A8	TIMERS_RAWINTSTATUS	原始中断状态寄存器	2-20

2.8.5 寄存器描述

TIMER1_LOADCOUNT

TIMER1_LOADCOUNT 为定时器 1 的初始值寄存器。

Offset Address: 0x000　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer1_loadcount	定时器 1 初始值。	0x00000000

TIMER1_CURRENTVALUE

TIMER1_CURRENTVALUE 为定时器 1 的当前值寄存器。

Offset Address: 0x004　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer1_currentvalue	定时器 1 当前值。	0x00000000

TIMER1_CONTROLREG

TIMER1_CONTROLREG 为定时器 1 的控制寄存器。

Offset Address: 0x008　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	timer1_lock	定时器 1 的锁定控制。 0：不锁定；	0x0

			1：将定时器 1 的当前值锁定到 TIMER1_CURRENTVALUE 中。 当完成定时器的锁定操作后，该寄存器自动清零。	
[2]	RW	timer1_int_mask	定时器 1 中断屏蔽位。 0：不屏蔽； 1：屏蔽。	0x0
[1]	RW	timer1_mode	定时器 1 模式控制位。 0：自由模式； 1：周期模式。	0x0
[0]	RW	timer1_en	定时器 1 的使能位。 0：禁止； 1：使能。	0x0

TIMER1_EOI

TIMER1_EOI 为定时器 1 的清中断寄存器。

Offset Address: 0x00C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer1_eoi	读该寄存器清除定时器 1 的中断。	0x0

TIMER1_INTSTATUS

TIMER1_INTSTATUS 为定时器 1 的中断状态寄存器。

Offset Address: 0x010　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer1_int_status	定时器 1 经过屏蔽之后的中断状态。 0：中断无效； 1：中断有效。	0x0

TIMER2_LOADCOUNT

TIMER2_LOADCOUNT 为定时器 2 的初始值寄存器。

Offset Address: 0x014　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer2_loadcount	定时器 2 初始值。	0x00000000

TIMER2_CURRENTVALUE

TIMER2_CURRENTVALUE 为定时器 2 的当前值寄存器。

Offset Address: 0x018　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer2_currentvalue	定时器 2 当前值。	0x00000000

TIMER2_CONTROLREG

TIMER2_CONTROLREG 为定时器 2 的控制寄存器。

Offset Address: 0x01C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	timer2_lock	定时器 2 的锁定控制。 0：不锁定； 1：将定时器 2 的当前值锁定到 TIMER2_CURRENTVALUE 中。 当完成定时器的锁定操作后，该寄存器自动清零。	0x0
[2]	RW	timer2_int_mask	定时器 2 中断屏蔽位。 0：不屏蔽； 1：屏蔽。	0x0
[1]	RW	timer2_mode	定时器 2 模式控制位。 0：自由模式； 1：周期模式。	0x0
[0]	RW	timer2_en	定时器 2 的使能位。 0：禁止；	0x0

			1：使能。	

TIMER2_EOI

TIMER2_EOI 为定时器 2 的清中断寄存器。

Offset Address: 0x020　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer2_eoi	读该寄存器清除定时器 2 的中断。	0x0

TIMER2_INTSTATUS

TIMER2_INTSTATUS 为定时器 2 的中断状态寄存器。

Offset Address: 0x024　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer2_int_status	定时器 2 经过屏蔽之后的中断状态。 0：中断无效； 1：中断有效。	0x0

TIMER3_LOADCOUNT

TIMER3_LOADCOUNT 为定时器 3 的初始值寄存器。

Offset Address: 0x028　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer3_loadcount	定时器 3 初始值。	0x00000000

TIMER3_CURRENTVALUE

TIMER3_CURRENTVALUE 为定时器 3 的当前值寄存器。

Offset Address: 0x02C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer3_currentvalue	定时器 3 当前值。	0x00000000

TIMER3_CONTROLREG

TIMER3_CONTROLREG 为定时器 3 的控制寄存器。

Offset Address: 0x030　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	timer3_lock	定时器 3 的锁定控制。 0：不锁定； 1：将定时器 3 的当前值锁定到 TIMER3_CURRENTVALUE 中。 当完成定时器的锁定操作后，该寄存器自动清零。	0x0
[2]	RW	timer3_int_mask	定时器 3 中断屏蔽位。 0：不屏蔽； 1：屏蔽。	0x0
[1]	RW	timer3_mode	定时器 3 模式控制位。 0：自由模式； 1：周期模式。	0x0
[0]	RW	timer3_en	定时器 3 的使能位。 0：禁止； 1：使能。	0x0

TIMER3_EOI

TIMER3_EOI 为定时器 3 的清中断寄存器。

Offset Address: 0x034　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer3_eoi	读该寄存器清除定时器 3 的中断。	0x0

TIMER3_INTSTATUS

TIMER3_INTSTATUS 为定时器 3 的中断状态寄存器。

Offset Address: 0x038　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer3_int_status	定时器 3 经过屏蔽之后的中断状态。 0：中断无效； 1：中断有效。	0x0

TIMER4_LOADCOUNT

TIMER4_LOADCOUNT 为定时器 4 的初始值寄存器。

Offset Address: 0x03C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer4_loadcount	定时器 4 初始值。	0x00000000

TIMER4_CURRENTVALUE

TIMER4_CURRENTVALUE 为定时器 4 的当前值寄存器。

Offset Address: 0x040　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer4_currentvalue	定时器 4 当前值。	0x00000000

TIMER4_CONTROLREG

TIMER4_CONTROLREG 为定时器 4 的控制寄存器。

Offset Address: 0x044　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	timer4_lock	定时器 4 的锁定控制。 0：不锁定； 1：将定时器 4 的当前值锁定到	0x0

			TIMER4_CURRENTVALUE 中。 当完成定时器的锁定操作后，该寄存器自动清零。	
[2]	RW	timer4_int_mask	定时器 4 中断屏蔽位。 0：不屏蔽； 1：屏蔽。	0x0
[1]	RW	timer4_mode	定时器 4 模式控制位。 0：自由模式； 1：周期模式。	0x0
[0]	RW	timer4_en	定时器 4 的使能位。 0：禁止； 1：使能。	0x0

TIMER4_EOI

TIMER4_EOI 为定时器 4 的清中断寄存器。

Offset Address: 0x048　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer4_eoi	读该寄存器清除定时器 4 的中断。	0x0

TIMER4_INTSTATUS

TIMER4_INTSTATUS 为定时器 4 的中断状态寄存器。

Offset Address: 0x04C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer4_int_status	定时器 4 经过屏蔽之后的中断状态。 0：中断无效； 1：中断有效。	0x0

TIMERS_INTSTATUS

TIMERS_INTSTATUS 为定时器中断状态寄存器。

Offset Address: 0x0A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	timer4_int_status_sum	定时器 4 经过屏蔽之后的中断状态位。 0：中断无效； 1：中断有效。	0x0
[2]	RO	timer3_int_status_sum	定时器 3 经过屏蔽之后的中断状态位。 0：中断无效； 1：中断有效。	0x0
[1]	RO	timer2_int_status_sum	定时器 2 经过屏蔽之后的中断状态位。 0：中断无效； 1：中断有效。	0x0
[0]	RO	timer1_int_status_sum	定时器 1 经过屏蔽之后的中断状态位。 0：中断无效； 1：中断有效。	0x0

TIMERS_EOI

TIMERS_EOI 为定时器清中断寄存器。

Offset Address: 0x0A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timer_seoi	读该寄存器可以清除定时器的中断。	0x0

TIMERS_RAWINTSTATUS

TIMERS_RAWINTSTATUS 为原始中断状态寄存器。

Offset Address: 0x0A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	timer4_raw_int_status	定时器 4 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0

[2]	RO	timer3_raw_int_status	定时器 3 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0
[1]	RO	timer2_raw_int_status	定时器 2 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0
[0]	RO	timer1_raw_int_status	定时器 1 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0

2.9 定时器

2.9.1 概述

芯片中有 4 个相同且可独立配置的定时器（Timer），4 个 Timer 可以独立进行时钟使能，可以灵活地指定时钟间隔。4 个 Timer 分别从预置的数据开始递减直至 0，产生中断。Timer 的初始数据、工作模式、中断方式都可以通过寄存器配置。

2.9.2 功能描述

Timer 具有以下功能特点：

- 4 个 32bit 的可独立配置的 Timer 单元。
- 支持自由模式和周期模式。
- 每个 Timer 单元可以独立进行使能。

2.9.3 工作方式

Timer0～Timer3 工作时钟频率为 40MHz/24MHz，支持分频配置。

Timer 支持自由模式和周期模式，每个 Timer 均可单独配置：

- 在自由模式下，Timer 从 0xFFFF_FFFF 递减计数到 0。
- 在周期模式下，计数器从 TIMERx_LOADCOUNT（x=0～3）的值计数到 0。

以 Timer1 和 Timer2 为例说明 Timer 的工作流程，具体步骤如下：

1. 配置 TIMER1_CONTROLREG 和 TIMER2_CONTROLREG 初始化 Timer：
   - 写 TIMER1_CONTROLREG[timer1_en]、TIMER2_CONTROLREG[timer2_en]为 0，关闭 Timer。

     注意：为了避免不同步问题，在设置 TIMER1_LOADCOUNT 和 TIMER2_LOADCOUNT 之前必须将 Timer 关闭。

- 写 TIMER1_CONTROLREG[timer1_mode]、TIMER2_CONTROLREG[timer2_mode]为 0，配置 Timer 为自由模式；或写 1，配置 Timer 为周期模式。

  注意：在配置 Timer 模式为自由模式工作之前，必须设置 TIMER1_LOADCOUNT 和 TIMER2_LOADCOUNT 的所有有效位都为 1，避免 Timer 定时出现异常。

- 写 TIMER1_CONTROLREG[timer1_int_mask]、TIMER2_CONTROLREG[timer2_int_mask]为 1，设置中断屏蔽为屏蔽方式；或写 0，设置中断屏蔽为不屏蔽方式。

2. 向 TIMER1_LOADCOUNT 和 TIMER2_LOADCOUNT 写入 Timer 的初始值。
3. 写 TIMER1_CONTROLREG[timer1_en]、TIMER2_CONTROLREG[timer2_en]为 1，使能 Timer，开始计时。

----结束

2.9.4 寄存器概览

Timer 寄存器概览如表 2-6 所示。

表2-6 Timer 寄存器概览（基址是 0x4005_0000）

偏移地址	名称	描述	页码
0x000＋0x14×n	TIMERn_LOADCOUNT	定时器 n 的初始值寄存器	2-23
0x004＋0x14×n	TIMERn_CURRENTVALUE	定时器 n 的当前值寄存器	2-23
0x008＋0x14×n	TIMERn_CONTROLREG	定时器 n 的控制寄存器	2-23
0x00C＋0x14×n	TIMERn_EOI	定时器 n 的清中断寄存器	2-24
0x010＋0x14×n	TIMERn_INTSTATUS	定时器 n 的中断状态寄存器	2-24
0x0A0	TIMERS_INTSTATUS	定时器中断状态寄存器	2-19
0x0A4	TIMERS_EOI	定时器清中断寄存器	2-20
0x0A8	TIMERS_RAWINTSTATUS	原始中断状态寄存器	2-20

Timer 寄存器偏移地址中变量的取值范围和含义如表 2-7 所示。

表2-7 Timer 寄存器偏移地址变量表

变量名称	取值范围	描述
n	0～3	Timer 编号。

2.9.5 寄存器描述

TIMERn_LOADCOUNT

TIMERn_LOADCOUNT 为定时器 n 的初始值寄存器。

Offset Address: 0x000＋0x14×n　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timern_loadcount	定时器 n 初始值。	0x00000000

TIMERn_CURRENTVALUE

TIMERn_CURRENTVALUE 为定时器 n 的当前值寄存器。

Offset Address: 0x004＋0x14×n　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timern_currentvalue	定时器 n 当前值。	0x00000000

TIMERn_CONTROLREG

TIMERn_CONTROLREG 为定时器 n 的控制寄存器。

Offset Address: 0x008＋0x14×n　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	timern_lock	定时器 n 的锁定控制。 0：不锁定； 1：将定时器 n 的当前值锁定到 TIMERn_CURRENTVALUE 中。 **注意：当完成定时器的锁定操作后，该寄存器自动清零。**	0x0
[2]	RW	timern_int_mask	定时器 n 中断屏蔽位。 0：不屏蔽； 1：屏蔽。	0x0
[1]	RW	timern_mode	定时器 n 模式控制位。 0：自由模式； 1：周期模式。	0x0

[0]	RW	timern_en	定时器 n 的使能位。 0：禁止； 1：使能。	0x0

TIMERn_EOI

TIMERn_EOI 为定时器 n 的清中断寄存器。

Offset Address: 0x00C＋0x14×n　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timern_eoi	读该寄存器清除定时器 n 的中断。	0x0

TIMERn_INTSTATUS

TIMERn_INTSTATUS 为定时器 n 的中断状态寄存器。

Offset Address: 0x010＋0x14×n　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timern_int_status	定时器 n 经过屏蔽后的中断状态。 0：中断无效； 1：中断有效。	0x0

TIMERS_INTSTATUS

TIMERS_INTSTATUS 为定时器中断状态寄存器。

Offset Address: 0x0A0　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	timer3_int_status_sum	定时器 3 经过屏蔽后的中断状态位。 0：中断无效； 1：中断有效。	0x0
[2]	RO	timer2_int_status_sum	定时器 2 经过屏蔽后的中断状态位。	0x0

			0：中断无效； 1：中断有效。	
[1]	RO	timer1_int_status_sum	定时器 1 经过屏蔽后的中断状态位。 0：中断无效； 1：中断有效。	0x0
[0]	RO	timer0_int_status_sum	定时器 0 经过屏蔽后的中断状态位。 0：中断无效； 1：中断有效。	0x0

TIMERS_EOI

TIMERS_EOI 为定时器清中断寄存器。

Offset Address: 0x0A4　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	timers_eoi_sum	读该寄存器清除所有定时器的中断。	0x0

TIMERS_RAWINTSTATUS

TIMERS_RAWINTSTATUS 为原始中断状态寄存器。

Offset Address: 0x0A8　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	timer3_raw_int_status	定时器 3 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0
[2]	RO	timer2_raw_int_status	定时器 2 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0
[1]	RO	timer1_raw_int_status	定时器 1 未经屏蔽的中断状态。 0：中断无效； 1：中断有效。	0x0
[0]	RO	timer0_raw_int_status	定时器 0 未经屏蔽的中断状态。	0x0

			0：中断无效； 1：中断有效。	

2.10 看门狗

2.10.1 概述

WatchDog 用于系统异常恢复，如果未得到更新则隔一定时间（可编程）产生一个系统复位信号，当 WatchDog 在此之前关闭工作时钟或更新计数器，复位信号不会产生。

2.10.2 功能描述

WatchDog 内置 1 个可编程 32bit 计数器，具有以下功能特点：

- 内置计数器进行递减计数，当由预设值递减到 0 时产生超时。
- WatchDog 具有 2 种工作方式：
  - 方式一：如果超时，则只产生系统复位。
  - 方式二：第一次超时，WatchDog 产生中断；第二次超时，如果中断未被清除，WatchDog 产生系统复位。
- 可配置超时间隔。

2.10.3 工作方式

WatchDog 计数器复位值是 0xFFFF，第一次计数都是从 0xFFFF 开始递减。踢狗或在第二种工作模式下，计数器首次从 0xFFFF 递减到 0 后，才会从配置的超时间隔开始计数。如果需 WatchDog 直接从配置的超时间隔开始计数，必须在使能 WatchDog 后立即进行踢狗。

在 WatchDog 打开后只能通过写 SOFT_RESET[soft_rst_wdt_n]为 0 或全局复位，停止 WatchDog 计数时钟，从而暂停 WatchDog。

WarchDog 启动的步骤如下：

1. 写 CLKEN[wdt_clken]为 1，打开 WatchDog 计数时钟。
2. 写 WDT_TORR，设置 WatchDog 超时间隔（单位：时钟周期）。
3. 写 WDT_CR[rmod]，设置 WatchDog 工作方式。
4. 写 WDT_CR[wdt_en]为 1，启动 WatchDog。

----结束

2.10.4 寄存器概览

WDT 寄存器概览如表 2-8 所示。

表2-8 WDT 寄存器概览（基址是 0x4000_0000）

偏移地址	名称	描述	页码
0x000	WDT_CR	看门狗控制寄存器	2-27
0x004	WDT_TORR	超时间隔寄存器	2-27
0x008	WDT_CCVR	计数器当前值寄存器	2-28
0x00C	WDT_CRR	计数器重新启动寄存器	2-28
0x010	WDT_STAT	中断状态寄存器	2-28
0x014	WDT_EOI	清除中断寄存器	2-29

2.10.5 寄存器描述

WDT_CR

WDT_CR 为看门狗控制寄存器。

Offset Address: 0x000　　Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	cr_bit5	无意义。	0x0
[4:2]	RW	reserved	保留。	0x2
[1]	RW	rmod	复位模式选择。 0：如果超时，则只产生复位信号； 1：第一次超时产生中断；第二次超时时如果中断还未被清除，则产生复位信号。	0x0
[0]	RWS	wdt_en	WatchDog 使能控制位。 0：禁止； 1：使能。 **注意：一旦该位被置为 1，只能由系统复位或软复位清除。**	0x0

WDT_TORR

WDT_TORR 为超时间隔寄存器。

Offset Address: 0x004　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RW	top	设置计时时间间隔，单位：时钟周期。 计算方法：Timeoutperiod=0xFFFF×2^top = 2^(top+16) **注意：该 WatchDog 复位后是从 0xFFFF 开始计数而不是配置的值，如要从配置的值开始计数必须使能 WatchDog 后进行踢狗。**	0x0

WDT_CCVR

WDT_CCVR 为计数器当前值寄存器。

Offset Address: 0x008　　Total Reset Value: 0x0000_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RO	ccvr	计数器当前值。	0x0000FFFF

WDT_CRR

WDT_CRR 为计数器重新启动寄存器。

Offset Address: 0x00C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	WO	crr	向该寄存器写入 0x76 可以重新启动计数器计数。	0x00

WDT_STAT

WDT_STAT 为中断状态寄存器。

Offset Address: 0x010　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	isr	经过屏蔽后的中断状态位。 0：中断无效；	0x0

			1：中断有效。	

WDT_EOI

WDT_EOI 为清除中断寄存器。

Offset Address: 0x014　　Total Reset Value: 0x0000_0000

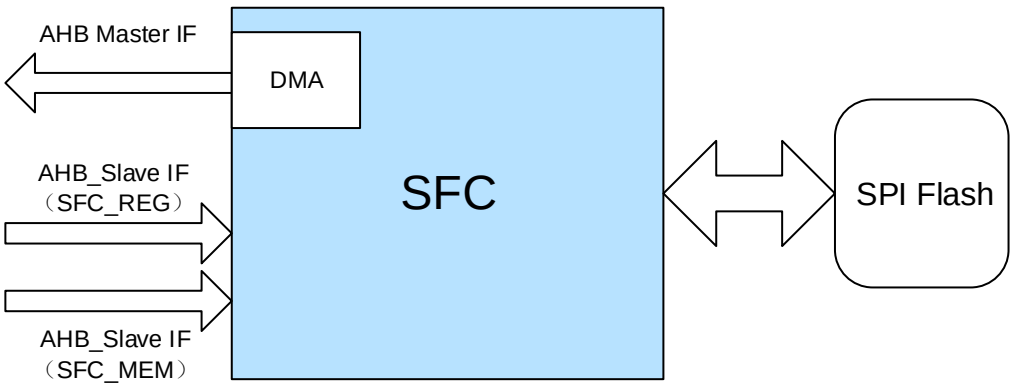
Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	eoi	中断清除，通过读该寄存器来清除中断。	0x0

3 SPI Flash 控制器

3.1 概述

SFC 是一个 SPI Flash 控制器。业务侧提供一个 AHB（Advanced High Performance Bus） Slave 接口，主要完成 AHB 通道对 SPI Flash 的访问控制功能；提供一个 AHB Master 接口，用于 DMA 方式读写 Flash。

图3-1 SFC 应用框图



注：IF（Interface）。

3.2 功能描述

AHB Slave 接口

AHB Slave 接口具有以下特点：

- 提供一个 AHB Slave 接口，可以根据不同的选择信号访问内部配置寄存器或直接访问 SPI Flash Memory。
- 支持 AMBA2.0 协议。
- 仅支持小端（Little-Endian）。

AHB Master 接口

AHB Master 接口具有以下特点：

- 提供一个 AHB Master 接口，用于 DMA 方式在内存和 Flash 之间搬运数据。
- 支持 AMBA2.0 协议。
- 只支持小端。
- 只有 Single、INCR4、INCR8、INCR16 传输类型。
- 不支持 Early Termination。
- 支持总线 Lock。

存储器接口

存储器接口具有以下特点：

- 片选的存储空间最大支持到 16Mbit（3byte 地址模式）。片选映射基地址可配置。只支持片选 1，不支持片选 0。
- 支持地址 Alias，可通过地址 Alias 实现上电后映射 0 地址到 Flash，芯片从 Flash 启动。地址 Alias 固定为 Flash 片选 1。
- 支持 Standard SPI、Dual-Output/Dual-Input SPI、Quad-Output/Quad-Input SPI、Dual-I/O SPI、Quad-I/O SPI 五种接口类型。上电后默认支持 Standard SPI 接口类型，可通过寄存器配置切换接口类型。
- 不支持 XIP（Executed In Place）。
- SPI Flash 读写操作支持总线直接读写、寄存器编程读写、DMA 读写三种方式。
- 支持多种写保护操作。
- SFC 模块支持 SPI Mode0 和 Mode3，按协议要求，支持 SPI Mode0 和 Mode3 的 SPI Flash 器件在时钟的上升沿采样数据，在时钟的下降沿输出数据。

Flash 数据加解扰

Flash 数据加解扰具有以下特点：

- 支持对 Flash 写入数据进行加扰，对 Flash 读出数据进行解扰。
- 支持单独对 Flash 某一段指定地址区间的读出数据不解扰。

说明

控制器对 Flash 数据加解扰使能、加解扰参数由 EFUSE 控制，为了实现 Flash 数据读写一致，必须保证：在整个 Flash 读写过程中，EFUSE 输出的 Flash 控制器加解扰使能和加解扰参数保持不变。

3.3 工作方式

3.3.1 读写 Flash

有三种方式读写 Flash：

- 通过寄存器配置方式发送 SPI FLASH Program、Read 等命令来读写 Flash。例如：对寄存器 CMD_CONFIG 写 0x0000_7F8B，表示读 64byte 的操作。

  此方式直接控制需要发送的 Flash 命令。
- 通过 AHB Slave 接口以类似读写普通 Memory 的方式读写 Flash。

  SFC 模块会自动将 AHB 总线的读写操作时序映射为 SPI Flash 读写命令。
- 通过 DMA 方式在 Flash 和外部 Memory 之间 Copy 数据。

3.3.2 其他操作

对 Flash 的其他操作如 Erase、进入 Deep Power Down、读 Device ID 等必须通过寄存器访问来实现。需要配置 CMD_INS[REG_INS]为相应的命令，具体请参见 Flash 器件手册。

例如：对寄存器 CMD_INS 写 0x0000_009F，表示读器件 ID 的操作。

3.3.3 初始化流程

须知

- 寄存器操作寄存器无需初始化，每次操作都需要重新配置。
- 注意以下初始化流程仅做参考，请根据器件差异进行调整。

初始化流程如下：

1. （如果需要调整 Timing 参数）配置 TIMING 寄存器。
2. 配置总线操作方式寄存器：
   - 根据实际 Flash 大小配置 BUS_FLASH_SIZE[flash_size_cs0]（直接获知器件大小或可通过发 Read ID 命令给 Flash 查询获得）。
   - 依据 Flash 映射到系统地址空间情况配置 BUS_BASE_ADDR_CS0、BUS_BASE_ADDR_CS1、BUS_ALIAS_CS。映射空间应在系统总线分配给 SFC_MEM 的地址空间范围内。

     通常 BUS_ALIAS_CS 对应从 Flash 启动时映射的地址，只默认值有意义，所以一般固定不变。
   - 有些器件要求进入非 Standard SPI 读写时序，需要预先以特殊命令配置 Flash。根据器件需要，对寄存器 CMD_INS 进行写操作，发特定命令配置 Flash。
   - 通过 BUS_CONFIG1/BUS_CONFIG2 配置总线读写操作指令和参数。

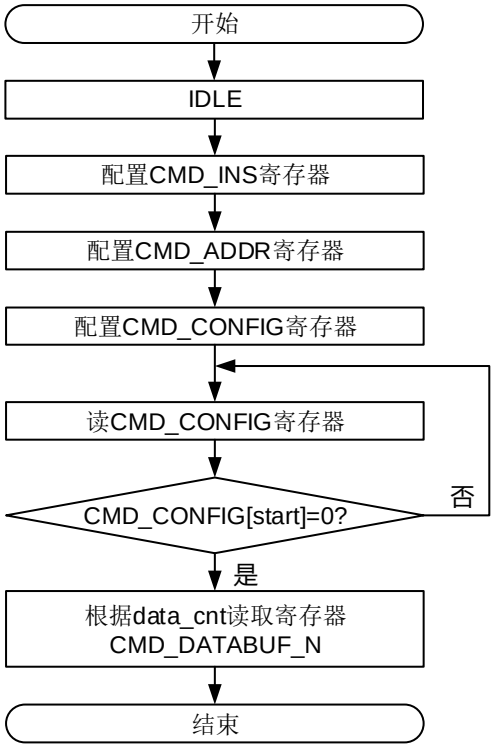
     例如：对寄存器 BUS_CONFIG1 写 0xCC85_EB1E 表示配置的参数为写指令 32，写方式为 Quad-Input SPI，读指令 EB、读方式为 Quad I/O SPI。
   - 如果需要开启总线写操作，配置 BUS_CONFIG1[wr_enable]为 1，使能总线写。默认关闭总线写功能。

----结束

3.3.4 通过寄存器方式读 Flash 操作流程

通过寄存器读取 Flash 的操作流程（查询方式）如图 3-2 所示。

图3-2 通过寄存器读取 Flash 的操作流程（查询方式）



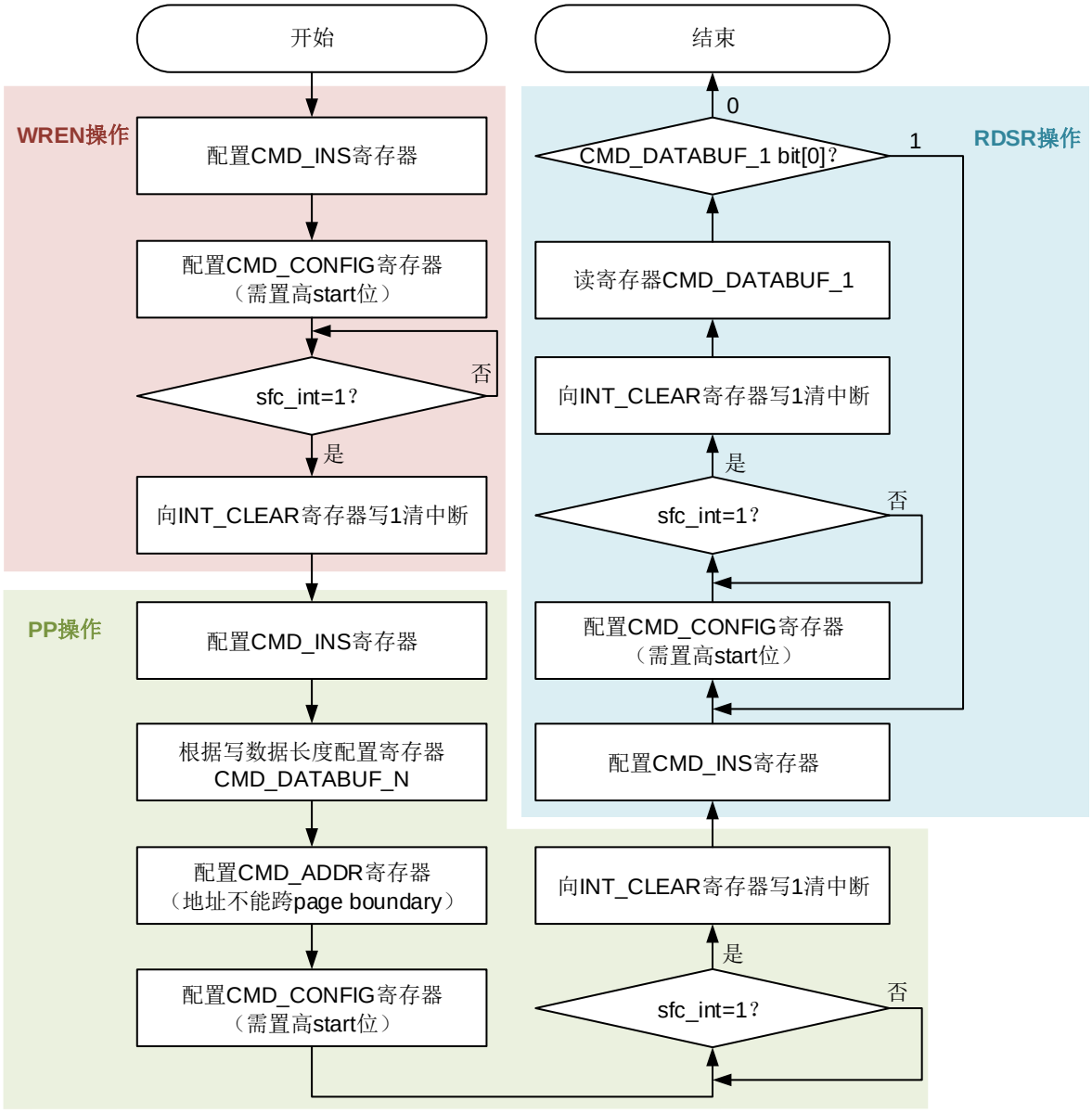
3.3.5 通过寄存器方式写 Flash 操作流程

须知

- 通过寄存器方式写 Flash 数据时，总线和 DMA 不得访问 Flash。
- 单次写 Flash 不能跨越 Page 边界（寄存器写方式没有跨越 Page 边界保护，需要软件保证，如果跨越 256byte 边界，将会 Wrap 到该 Page 的起始地址，覆盖原来的内容）。

通过寄存器写 Flash 的操作流程（中断方式）如图 3-3 所示。

图3-3 通过寄存器写 Flash 的操作流程（中断方式）

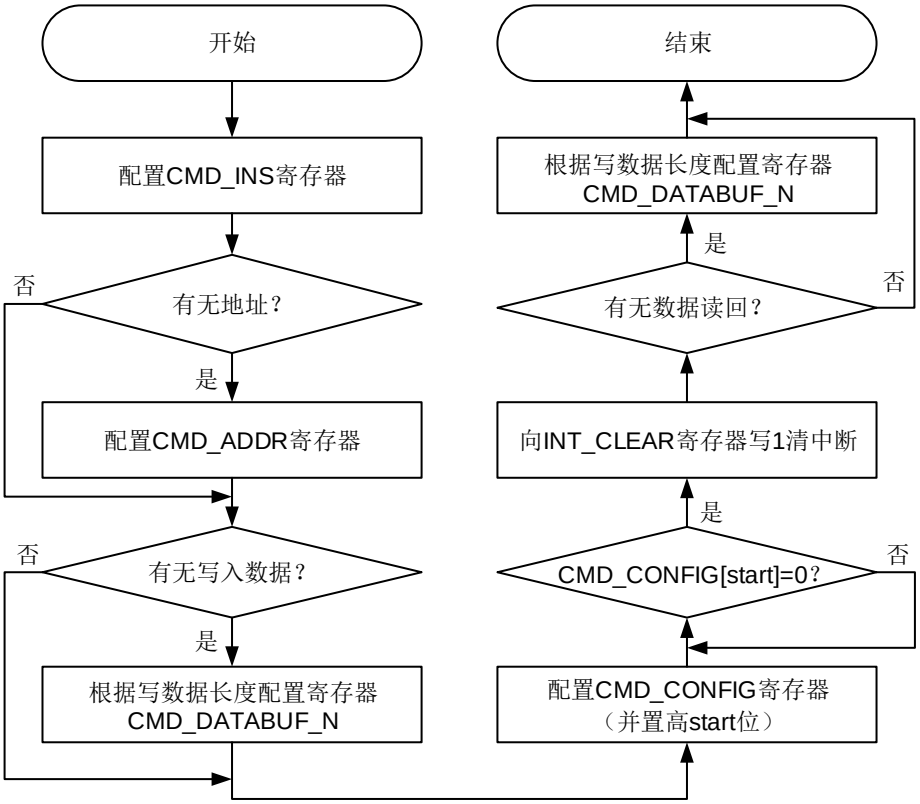


注：WREN（Write Read Enable），PP（Page Program），RDSR（Read Status Register）。

3.3.6 通过寄存器方式其他操作流程

通过寄存器方式其他操作流程如图 3-4 所示。

图3-4 通过寄存器方式其他操作流程



说明

SFC 控制器不支持发出“OPCODE（1byte） + DUMMY（3byte 全 0）”组合 SPI 时序，某些 Flash 指令需要这种组合时序时，可以采用“OPCODE（1byte） + ADDR（3byte 全 0）”组合代替。

3.3.7 通过 AHB Slave 直接读写 Flash 操作流程

上电复位后，默认配置为 Standard SPI 时序模式。不需要额外配置，可直接读 Flash。

默认通过 AHB Slave 写 Flash 是禁止的。需要配置 BUS_CONFIG1[wr_enable]为 1，使能总线写操作。

如果需要调整默认配置，请参见“3.3.3 初始化流程”。

3.3.8 通过 DMA 方式读写 Flash 操作流程

DMA 操作流程如下：

1. 如需调整总线操作方式时序配置，请参考“3.3.3 初始化流程”。
2. 写 BUS_DMA_MEM_SADDR，配置 DMA 操作的内存端起始地址；写 BUS_DMA_FLASH_SADDR，配置 Flash 端起始地址（Flash 偏移地址）；写 BUS_DMA_LEN，配置数据长度。
3. 写 BUS_DMA_CTRL，配置读写方向，操作 Flash 片选（Flash0 或 Flash1）。

4. 写 BUS_DMA_CTRL[start]为 1，使能 DMA 操作。
5. 等待 dma_done 中断触发（中断方式）或轮询 DMA 操作完成（BUS_DMA_CTRL[start]变为 0）。

说明

- DMA 操作时可以同时 Flash 寄存器读命令操作。
- DMA 操作时可以同时通过 AHB Slave 直接访问 Flash，但需保证中间不修改总线操作相关配置。
- DMA 操作时需要保证首地址 4byte 对齐。

----结束

3.3.9 软件擦除 Flash 后回读数据操作流程

如果当前控制器加解扰使能开启，软件擦除 Flash 后，需要回读擦除地址数据是否为 0xFF 来判断擦除是否成功（这种回读的数据不能解扰），具体操作流程如下：

1. 写 SFC_NOSCM_ADDR_START[sfc_noscm_addr_start]、SFC_NOSCM_ADDR_END[sfc_noscm_addr_end]，配置软件回读 Flash 的起始地址、截止地址。

   注意：此处配置的起始地址、截止地址为 Flash 相对地址。

2. 写 SFC_NOSCM_EN[sfc_noscm_en]为 1，实现对步骤 1 中配置地址区间的读出数据不解扰。
3. 软件回读 Flash 数据进行判断。

----结束

3.4 寄存器概览

SFC 寄存器概览如表 3-1 所示。

表3-1 SFC 寄存器概览（基址是 0x4080_0000）

偏移地址	名称	描述	页码
0x0100	GLOBAL_CONFIG	全局配置寄存器	3-9
0x0110	TIMING	Timing 配置寄存器	3-9
0x0120	INT_RAW_STATUS	中断原始状态寄存器	3-10
0x0124	INT_STATUS	经过屏蔽处理的中断状态寄存器	3-10
0x0128	INT_MASK	中断屏蔽寄存器	3-11
0x012C	INT_CLEAR	中断清除寄存器	3-11
0x0150	SFC_NOSCM_ADDR_START	读数据不解扰起始地址寄存器	3-12

偏移地址	名称	描述	页码
0x0154	SFC_NOSCM_ADDR_END	读数据不解扰截止地址寄存器	3-12
0x0158	SFC_NOSCM_EN	读数据不解扰使能寄存器	3-12
0x01F8	VERSION	版本寄存器	3-13
0x01FC	VERSION_SEL	版本选择寄存器	3-13
0x0200	BUS_CONFIG1	总线操作方式配置 1 寄存器	3-13
0x0204	BUS_CONFIG2	总线操作方式配置 2 寄存器	3-15
0x0210	BUS_FLASH_SIZE	总线操作方式映射尺寸寄存器	3-15
0x0214	BUS_BASE_ADDR_CS0	总线操作方式片选 0 映射基地址寄存器	3-16
0x0218	BUS_BASE_ADDR_CS1	总线操作方式片选 1 映射基地址寄存器	3-17
0x021C	BUS_ALIAS_ADDR	总线操作方式 Alias 映射基地址寄存器	3-17
0x0220	BUS_ALIAS_CS	总线操作方式 Alias 片选指示寄存器	3-17
0x0240	BUS_DMA_CTRL	DMA 操作控制寄存器	3-18
0x0244	BUS_DMA_MEM_SADDR	DMA 操作 DDR 起始地址寄存器	3-18
0x0248	BUS_DMA_FLASH_SADDR	DMA 操作 Flash 起始地址寄存器	3-18
0x024C	BUS_DMA_LEN	DMA 操作搬运数据长度寄存器	3-19
0x0250	BUS_DMA_AHB_CTRL	DMA 操作 AHB 时 burst 操作方式选择控制寄存器	3-19
0x0300	CMD_CONFIG	命令操作方式配置寄存器	3-19
0x0308	CMD_INS	命令操作方式指令寄存器	3-21
0x030C	CMD_ADDR	命令操作方式地址寄存器	3-21
0x0400＋4×n	CMD_DATABUF_N	命令操作方式数据 Buffer 寄存器	3-21

SFC 寄存器偏移地址中变量的取值范围和含义如表 3-2 所示。

表3-2 SFC 寄存器偏移地址变量表

变量名称	取值范围	描述
n	0～15	数据 Buffer。

3.5 寄存器描述

GLOBAL_CONFIG

GLOBAL_CONFIG 为全局配置寄存器。

Offset Address: 0x0100　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:3]	RW	rd_delay	SPI 读出数据延迟周期个数。 000：0.5～1 个时钟周期(默认值)； 001：1～1.5 个时钟周期； 010：1.5～2 个时钟周期； 011：2～2.5 个时钟周期； 100：2.5～3 个时钟周期； 101：3～3.5 个时钟周期； 110：3.5～4 个时钟周期； 111：不支持(按照“110”含义处理)。	0x0
[2]	RW	flash_addr_mode	SPI 地址模式。 0：3byte 寻址模式(默认值)； 1：4byte 寻址模式。 **注意：CMD_CONFIG[start]为 1 时写无效。**	0x0
[1]	RW	wp_en	硬件写保护使能(写保护管脚)。 0：禁止； 1：使能。	0x0
[0]	RW	mode	SPI 模式设置。 0：支持 Mode0； 1：支持 Mode3。	0x0

TIMING

TIMING 为 Timing 配置寄存器。

Offset Address: 0x0110　　Total Reset Value: 0x0000_660F

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	tcsh	片选保持时间。 0x0～0x7：(n＋1)个时钟周期。 例如：0x6 表示 7 个时钟周期。	0x6
[11]	-	reserved	保留。	0x0
[10:8]	RW	tcss	片选建立时间。 0x0～0x7：(n＋1)个时钟周期。 例如：0x6 表示 7 个时钟周期。	0x6
[7:4]	-	reserved	保留。	0x0
[3:0]	RW	tshsl	设置 2 次 Flash 操作之间的时间间隔。 0x0～0xF：(n＋2)个时钟周期。 例如：0xF 表示 17 个时钟周期。	0xF

INT_RAW_STATUS

INT_RAW_STATUS 为中断原始状态寄存器。

Offset Address: 0x0120　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RO	dma_done_int_raw_status	DMA 操作完成中断原始状态(未经过屏蔽)。 0：未完成； 1：已完成。	0x0
[0]	RO	cmd_op_end_raw_status	指令操作结束原始中断状态(未经过屏蔽)。 0：未完成； 1：已完成。	0x0

INT_STATUS

INT_STATUS 为经过屏蔽处理的中断状态寄存器。

Offset Address: 0x0124　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RO	dma_done_int_status	DMA 操作完成中断状态(经过屏蔽)。 0：未完成； 1：已完成。	0x0
[0]	RO	cmd_op_end_status	指令操作结束中断状态(经过屏蔽)。 0：未完成； 1：已完成。	0x0

INT_MASK

INT_MASK 为中断屏蔽寄存器。

Offset Address: 0x0128　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	dma_done_int_mask	DMA 操作完成中断屏蔽位。 0：屏蔽； 1：不屏蔽。	0x0
[0]	RW	cmd_op_end_int_mask	指令操作结束中断屏蔽位。 0：屏蔽； 1：不屏蔽。	0x0

INT_CLEAR

INT_CLEAR 为中断清除寄存器。

Offset Address: 0x012C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	WO	dma_done_int_clr	DMA 操作完成中断清除位，向该位写 1 将清除 INT_STATUS[dma_done_int_status]和 INT_RAW_STATUS[dma_done_int_raw_status]。 0：不清除； 1：清除。	0x0

			注意：清除操作完成后该位自动返回 0。	
[0]	WO	cmd_op_end_int_clr	指令操作结束中断清除位，向该位写 1 将清除 INT_STATUS[cmd_op_end_status]和 INT_RAW_STATUS[cmd_op_end_raw_status]。 0：不清除； 1：清除。 **注意：清除操作完成后该位自动返回 0。**	0x0

SFC_NOSCM_ADDR_START

SFC_NOSCM_ADDR_START 为读数据不解扰起始地址寄存器。

Offset Address: 0x0150　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	sfc_noscm_addr_start	读数据不解扰起始地址。 软件擦除后，如果需要回读验证擦除是否正确，需要配置回读起始和截止地址、读数据不解扰使能。	0x000000

SFC_NOSCM_ADDR_END

SFC_NOSCM_ADDR_END 为读数据不解扰截止地址寄存器。

Offset Address: 0x0154　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	sfc_noscm_addr_end	读数据不解扰截止地址。 软件擦除后，如果需要回读验证擦除是否正确，需要配置回读起始和截止地址、读数据不解扰使能。	0x000000

SFC_NOSCM_EN

SFC_NOSCM_EN 为读数据不解扰使能寄存器。

Offset Address: 0x0158　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	sfc_noscm_en	读数据不解扰使能。 0：读数据正常解扰； 1：读数据不解扰。 软件擦除后，如果需要回读验证擦除是否正确，需要配置回读起始和截止地址、读数据不解扰使能。	0x0

VERSION

VERSION 为版本寄存器。

Offset Address: 0x01F8　　Total Reset Value: 0x0000_0350

Bits	Access	Name	Description	Reset
[31:0]	RO	version	SFC 版本 V350。	0x00000350

VERSION_SEL

VERSION_SEL 为版本选择寄存器。

Offset Address: 0x01FC　　Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	version_sel	新旧寄存器组指示信号。 0：旧版寄存器组； 1：新版寄存器组。 **注意：固定为 1，不可配置。**	0x1

BUS_CONFIG1

BUS_CONFIG1 为总线操作方式配置 1 寄存器。

Offset Address: 0x0200　　Total Reset Value: 0x8080_0300

Bits	Access	Name	Description	Reset
[31]	RW	rd_enable	总线读使能。	0x1

			0：禁止； 1：使能。	
[30]	RW	wr_enable	总线写使能。 0：禁止； 1：使能。	0x0
[29:22]	RW	wr_ins	写指令。	0x02
[21:19]	RW	wr_dummy_bytes	总线写操作 DummyByte。 000：没有 DummyByte； 001：1byte； 010：2byte； …… 111：7byte。	0x0
[18:16]	RW	wr_mem_if_type	总线写操作指定连接的 SPI FLASH 接口类型。 000：Standard SPI 接口类型； 001：Dual-Input/Dual-Output SPI； 010：Dual-I/O SPI； 011：Full DIO SPI； 100：保留； 101：Quad-Input/Dual-Output SPI； 110：Quad-I/O SPI； 111：Full QIO SPI。	0x0
[15:8]	RW	rd_ins	读指令。	0x03
[7:6]	RW	rd_prefetch_cnt	总线访问 Flash 方式(非定长读)预取周期。 00：不预取(默认值)； 01：预取 1 个时钟周期数据； 10：预取 2 个时钟周期数据； 11：预取 3 个时钟周期数据。	0x0
[5:3]	RW	rd_dummy_bytes	总线读操作 DummyByte。 00：没有 DummyByte； 001：1byte 010：2byte； …… 111：7byte。	0x0
[2:0]	RW	rd_mem_if_type	总线读操作指定连接的 SPI FLASH 接口类型。	0x0

			000：Standard SPI 接口类型； 001：Dual-Input/Dual-Output SPI； 010：Dual-I/O SPI； 101：Quad-Input/Dual-Output SPI； 110：Quad-I/O SPI； 其他：保留。	

BUS_CONFIG2

BUS_CONFIG2 为总线操作方式配置 2 寄存器。

Offset Address: 0x0204　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	wip_locate	WIP(Write In Progress)位于 Flash 状态寄存器的位置。 000：WIP 位于 Flash 状态寄存器的 bit[0](默认值)； 001：WIP 位于 Flash 状态寄存器的 bit[1]； 010：WIP 位于 Flash 状态寄存器的 bit[2]； 011：WIP 位于 Flash 状态寄存器的 bit[3]； 100：WIP 位于 Flash 状态寄存器的 bit[4]； 101：WIP 位于 Flash 状态寄存器的 bit[5]； 110：WIP 位于 Flash 状态寄存器的 bit[6]； 111：WIP 位于 Flash 状态寄存器的 bit[7]。	0x0

BUS_FLASH_SIZE

BUS_FLASH_SIZE 为总线操作方式映射尺寸寄存器。

Offset Address: 0x0210　　Total Reset Value: 0x0000_0909

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:8]	RW	flash_size_cs1	指定片选 1 连接的 SPI FLASH 容量。 0x0：没有连接 SPI FLASH； 0x1：512Kbit； 0x2：1Mbit；	0x9

			0x3：2Mbit； 0x4：4Mbit； 0x5：8Mbit； 0x6：16Mbit； 0x7：32Mbit； 0x8：64Mbit； 0x9：128Mbit(默认值)； 0xA：256Mbit； 0xB：512Mbit； 0xC：1Gbit； 0xD：2Gbit； 0xE：4Gbit； 0xF：8Gbit。	
[7:4]	-	reserved	保留。	0x0
[3:0]	RW	flash_size_cs0	指定片选 0 连接的 SPI Flash 容量。 0x0：没有连接 SPI FLASH； 0x1：512Kbit； 0x2：1Mbit； 0x3：2Mbit； 0x4：4Mbit； 0x5：8Mbit； 0x6：16Mbit； 0x7：32Mbit； 0x8：64Mbit； 0x9：128Mbit(默认值)； 0xA：256Mbit； 0xB：512Mbit； 0xC：1Gbit； 0xD：2Gbit； 0xE：4Gbit； 0xF：8Gbit。	0x9

BUS_BASE_ADDR_CS0

BUS_BASE_ADDR_CS0 为总线操作方式片选 0 映射基地址寄存器。

Offset Address: 0x0214　　Total Reset Value: 0x3200_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	bus_base_addr_high_cs0	CS0 Flash 映射到系统空间基地址。	0x3200
[15:0]	-	reserved	保留。	0x0000

BUS_BASE_ADDR_CS1

BUS_BASE_ADDR_CS1 为总线操作方式片选 1 映射基地址寄存器。

Offset Address: 0x0218　　Total Reset Value: 0x0040_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	bus_base_addr_high_cs1	CS1 Flash 映射到系统空间基地址。	0x0040
[15:0]	-	reserved	保留。	0x0000

BUS_ALIAS_ADDR

BUS_ALIAS_ADDR 为总线操作方式 Alias 映射基地址寄存器。

Offset Address: 0x021C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	flash_alias_addr	Flash 映射到系统空间第 2 个基地址。	0x0000
[15:0]	-	reserved	保留。	0x0000

BUS_ALIAS_CS

BUS_ALIAS_CS 为总线操作方式 Alias 片选指示寄存器。

Offset Address: 0x0220　　Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	alias_cs	Alias 片选选择指示位，外部管脚控制，复位后保持。 0：Alias 片选为 0； 1：Alias 片选为 1。	0x1

BUS_DMA_CTRL

BUS_DMA_CTRL 为 DMA 操作控制寄存器。

Offset Address: 0x0240　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	dma_sel_cs	DMA 操作指定片选。 0：片选 0； 1：片选 1。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	dma_rw	DMA 读写指示。 0：写操作； 1：读操作。	0x0
[0]	RW	dma_start	DMA 传输使能控制。 0：无操作； 1：开始 DMA 操作。 **注意：DMA 传输完成自动回 0。**	0x0

BUS_DMA_MEM_SADDR

BUS_DMA_MEM_SADDR 为 DMA 操作 DDR 起始地址寄存器。

Offset Address: 0x0244　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr	DMA 操作 memory 起始地址。 配置值应在 0x0200_0000～0x0203_FFFF 之间。	0x00000000

BUS_DMA_FLASH_SADDR

BUS_DMA_FLASH_SADDR 为 DMA 操作 Flash 起始地址寄存器。

Offset Address: 0x0248　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_flash_saddr	DMA 操作 Flash 起始地址。	0x00000000

BUS_DMA_LEN

BUS_DMA_LEN 为 DMA 操作搬运数据长度寄存器。

Offset Address: 0x024C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:0]	RW	dma_len	DMA 操作数据搬运长度(n+1)，单位：byte。 例如：6 表示长度为 7byte。	0x00000000

BUS_DMA_AHB_CTRL

BUS_DMA_AHB_CTRL 为 DMA 操作 AHB 时 burst 操作方式选择控制寄存器。

Offset Address: 0x0250　　Total Reset Value: 0x0000_0007

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	incr16_en	INC16 burst 类型使能。 0：禁止； 1：使能。	0x1
[1]	RW	incr8_en	INC8 burst 类型使能。 0：禁止； 1：使能。	0x1
[0]	RW	incr4_en	INC4 burst 类型使能。 0：禁止； 1：使能。	0x1

CMD_CONFIG

CMD_CONFIG 为命令操作方式配置寄存器。

Offset Address: 0x0300　　Total Reset Value: 0x0000_7E00

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19:17]	RW	mem_if_type	指定寄存器命令操作方式连接的 SPI FLASH 接口类型。	0x0

			000：Standard SPI 接口类型； 001：Dual-Input/Dual-Output SPI； 010：Dual-I/O SPI； 101：Quad-Input/Dual-Output SPI； 110：Quad-I/O SPI； 其他：保留。	
[16:15]	-	reserved	保留。	0x0
[14:9]	RW	data_cnt	读写数据长度(单位：byte)。 0x00～0x3F：(n+1)byte。 例如：0x3F 表示 64byte。	0x3F
[8]	RW	rw	标识此次操作数据读写，需[data_en]为 1。 0：写，有发送数据； 1：读，有返回数据。	0x0
[7]	RW	data_en	标识此次操作是否有数据。 0：无数据； 1：有数据。	0x0
[6:4]	RW	dummy_byte_cnt	寄存器命令操作方式 DummyByte。 000：没有 DummyByte； 001：1byte； 010：2byte； …… 111：7byte。	0x0
[3]	RW	addr_en	此次操作是否有地址。 0：无地址； 1：有地址。	0x0
[2]	-	reserved	保留。	0x0
[1]	RW	sel_cs	片选选择操作。 0：选择片选 0 进行操作； 1：选择片选 1 进行操作。	0x0
[0]	RW	start	标识指令操作开始。 0：结束； 1：开始。 **注意：此次操作完成后该位自动回 0。**	0x0

CMD_INS

CMD_INS 为命令操作方式指令寄存器。

Offset Address: 0x0308　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	reg_ins	寄存器访问 Flash 方式下的指令码。	0x00

CMD_ADDR

CMD_ADDR 为命令操作方式地址寄存器。

Offset Address: 0x030C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:0]	RW	cmd_addr	寄存器访问 Flash 方式下的操作地址。	0x00000000

CMD_DATABUF_N

CMD_DATABUF_N 为命令操作方式数据 Buffer 寄存器。

Offset Address: 0x0400＋4×n　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cmd_databuf_n	寄存器访问 Flash 方式下第 n 数据 Buffer(n：0～15)。	0x00000000

4 WiFi

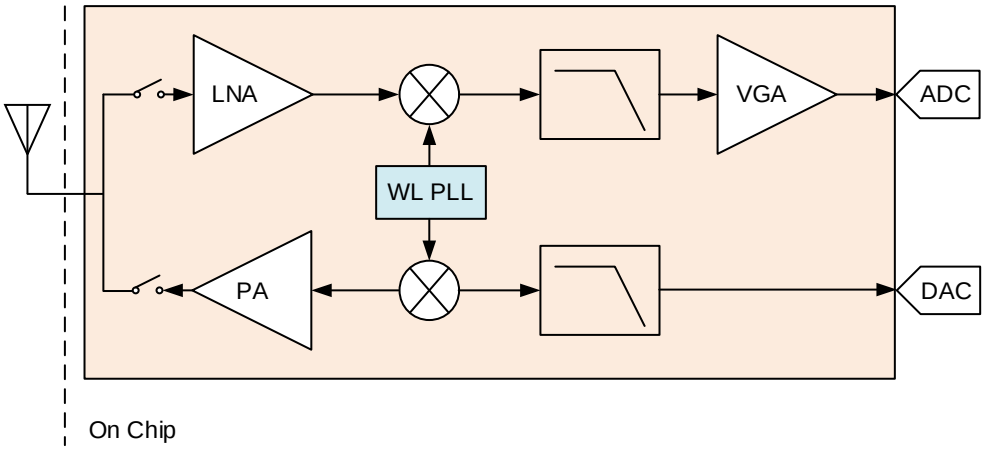
4.1 WiFi RF

4.1.1 概述

Hi3861V100、Hi3861LV100、Hi3881V100 的 RF 部分包含 2.4G RX、TX、PLL 三个模块。支持 IEEE 802.11b/g/n 20M 模式。RF 的电路功能主要包含：

- 集成 TX/RX Switch。
- RX 通路包含 LNA、Mixer、LPF（Low Pass Filter）、VGA（Variable Gain Amplifier）。
- TX 通路包含 LPF、UPC（UP Converter）、PA（Power Amplifier）。
- 集成 PLL/LO（Local Oscillator）通路，为信号通路提供 LO。

图4-1 RF 电路模块架构



4.1.2 功能描述

WiFi RF 具有以下功能特点：

- RF 电路提供稳定的 LO 信号，支持收发信号的上下变频功能。

- 支持校准功能，包含：RX DC（Direct Current）校准、TX LO Leakage 校准、TX Power 校准、TRX IQ 校准等。

4.1.3 工作方式

WiFi RF 工作模式有上电校准模式、TX 模式、RX 模式、睡眠模式。上电校准为软件控制，对 RF 的电路性能进行补偿。业务模式下，RF 的工作模式由 PHY 控制其 TX、RX 或睡眠模式。

4.1.4 RF 性能

说明

芯片集成 2.4G WiFi 收发机。如无特殊说明，性能描述基于芯片射频输入输出接口 WL_RF_RFIO_2G 的测试结果。

WLAN 2.4GHz 接收和发射性能如表 4-1、表 4-2、表 4-3 所示。

表4-1 WLAN 2.4GHz 接收性能-1

参数	速率	性能			单位
		最小值	典型值	最大值	
工作频段	-	2400	-	2480	MHz
RX 灵敏度 11b（8% PER for 1024 octet PSDU）	1Mbit/s	-	-100	-	dBm
	2Mbit/s	-	-96.5	-	dBm
	5.5Mbit/s	-	-95	-	dBm
	11Mbit/s	-	-92	-	dBm
RX 灵敏度 11g（10% PER for 1024 octet PSDU）	6Mbit/s	-	-96	-	dBm
	9Mbit/s	-	-94	-	dBm
	12Mbit/s	-	-93	-	dBm
	18Mbit/s	-	-91	-	dBm
	24Mbit/s	-	-88	-	dBm
	36Mbit/s	-	-85	-	dBm
	48Mbit/s	-	-81	-	dBm
	54Mbit/s	-	-79	-	dBm
RX 灵敏度 11n 20MHz HT-MF, 800ns GI（10% PER for 1024 octet PSDU）：non-STBC	HT20 MCS0	-	-95	-	dBm
	HT20 MCS1	-	-92	-	dBm
	HT20 MCS2	-	-90	-	dBm
	HT20 MCS3	-	-87	-	dBm
	HT20 MCS4	-	-84	-	dBm

参数	速率	性能			单位
		最小值	典型值	最大值	
	HT20 MCS5	-	-80	-	dBm
	HT20 MCS6	-	-78	-	dBm
	HT20 MCS7	-	-74	-	dBm
最大输入功率	（11b 1M/2M）<8% PER	-	10	-	dBm
	（11b 5.5M/11M）<8% PER	-	10	-	dBm
	（11g 6M～54M）<10% PER	-	TBD	-	dBm
	（11n mcs 0～7）<10% PER	-	TBD	-	dBm

表4-2 WLAN 2.4GHz 接收性能-2

参数	速率	有用信号强度（dBm）	性能			单位
			最小值	典型值	最大值	
ACI 抑制-11b 干扰信号与有用信号比	1Mbit/s	-74	-	51	-	dB
	2Mbit/s	-74	-	51	-	dB
	5.5Mbit/s	-70	-	43	-	dB
	11Mbit/s	-70	-	42	-	dB
ACI 抑制-11g 干扰信号与有用信号比	6Mbit/s	-79	-	44	-	dB
	9Mbit/s	-78	-	42	-	dB
	12Mbit/s	-76	-	41	-	dB
	18Mbit/s	-74	-	40	-	dB
	24Mbit/s	-71	-	37	-	dB
	36Mbit/s	-67	-	35	-	dB
	48Mbit/s	-63	-	31	-	dB
	54Mbit/s	-62	-	29	-	dB
ACI 抑制-11n 干扰信号与有用	HT20 MCS0	-79	-	42	-	dB
	HT20 MCS1	-76	-	40	-	dB

参数	速率	有用信号强度（dBm）	性能			单位
			最小值	典型值	最大值	
信号比	HT20 MCS2	-74	-	39	-	dB
	HT20 MCS3	-71	-	37	-	dB
	HT20 MCS4	-67	-	34	-	dB
	HT20 MCS5	-63	-	30	-	dB
	HT20 MCS6	-62	-	29	-	dB
	HT20 MCS7	-61	-	27	-	dB

表4-3 WLAN 2.4GHz 发射性能

参数	速率	参数			单位
		最小值	典型值	最大值	
TX 最高功率，VDD=3.3V，满足 mask 和 EVM 协议规范	11b（1M～11Mbit/s）	-	22	-	dBm
	11g（6Mbit/s）	-	21	-	dBm
	11g（54Mbit/s）	-	20	-	dBm
	OFDM，BPSK（Binary Phase Shift Keying） MCS0	-	19	-	dBm
	OFDM，QPSK（Quadrature Phase Shift Keying） MCS1～2	-	19	-	dBm
	OFDM，16QAM（Quadrature Amplitude Modulation） MCS3～4	-	19	-	dBm
	OFDM，64QAM MCS6	-	19	-	dBm
	OFDM，64QAM MCS7	-	19	-	dBm
输出功率精度	VSWR（Voltage Standing Wave Ratio）≤2:1	-	TBD	-	dB

参数	速率	参数			单位
		最小值	典型值	最大值	
输出功率分辨率	-	-	TBD	-	dB

注：以上数据测试条件为 VBAT=3.3V；VBAT=2.3V 测试数据 TBD。

4.2 WiFi ABB

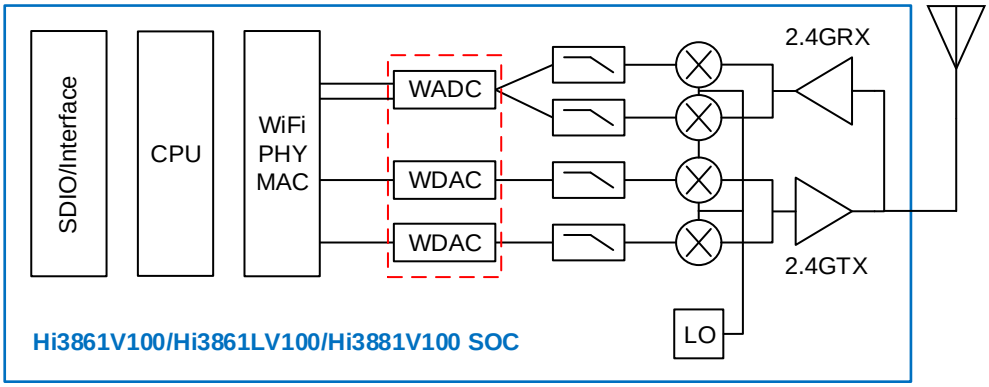
4.2.1 概述

Hi3861/Hi3861L/Hi3881 ABB IP 用于 Hi3861/Hi3861L/Hi3881 Connectivity SoC 芯片，是支持 WiFi 802.11b/g/n（2.4G mode）系统的模拟数字接口模块，根据功能分为以下 2 个功能模块：

- WiFi IQ-ADC
- WiFi 双通道 DAC

完成发送时的数模转换及接收时的模数转换功能。

图4-2 ABB IP 在 SoC 中应用示意图



注：WADC（WiFi Analog Digital Converter），WDAC（WiFi digital analog converter）。

4.2.2 功能描述

WiTP ABB IP 具有以下功能特点：

- 提供 1 路 WiFi IQ ADC、1 对 WiFi IQ DAC
- 支持 WiFi 802.11b/g/n（2.4G mode）

4.2.3 工作方式

Hi3861/Hi3861L/Hi3881 的业务模式寄存器配置为固定一次性配置，业务工作期间无需重复配置，仅在逻辑电源掉电重新上电时需要重新配置；校正算法在温度、电压漂移

下受影响，如果温度、电压变化较大，需要重新运行算法并刷新寄存器，除此情况之外无需重复配置。

校准包括：

- WLAN（Wireless Local Area Network）的 ADC 比较器校准
- WLAN DC Offset 校准
- WLAN 电容校准

校准顺序：

1. 比较器校准。
2. 电容校准。
3. DC Offset 校准。

----结束

4.3 WiFi PHY

4.3.1 概述

WLAN PHY 实现 802.11 协议定义的物理层功能，包括：

- 802.11b 协议定义的 DSSS、CCK 调制解调。
- 802.11n、802.11g 协议定义的 OFDM 调制解调包括发送的加扰、编码（卷积编码）、交织、OFDM 调制等处理；接收方向 Viterbi 译码处理；同时实现 AGC（Automatic Gain Control）、CCA（Clear Channel Assessment）功能。
- 实现 RF/ABB 测试及校准功能。

4.3.2 功能描述

WiFi PHY 具有以下功能特点：

- 支持 IEEE802.11b/g/n 无线局域网络通信协议。
- 支持 2.4G Band、5MHz/10MHz/20MHz 带宽，最大支持 1 流、1 天线。
- 支持 DPD（Digital Pre-Distortion）和校准功能。

4.4 WiFi MAC

4.4.1 概述

DBB（Digital Baseband） MAC 主要完成 WiFi MAC 层的硬件处理，包括信道接入、组解帧、数据收发、加解密、节能控制等功能。

4.4.2 功能描述

WiFi MAC 具有以下功能特点：

- 支持 IEEE802.11b/g/n 无线局域网络通信协议。
- 支持 STA 模式和 AP 模式。
- 支持 2.4G Band、20M/10M/5M 带宽，最大支持 1 流、1 天线。
- 支持 WPA、WPA2、AES 加解密，支持中国 WAPI 标准。
- 支持 WPS2.0。
- 支持协议低功耗：PSM（Power Saving Mode）、UAPSD（Unscheduled Automatic Power Save Delivery）、P2P（Peer-to-Peer） Power Save。

4.4.3 工作方式

芯片支持 3 种基本模式：

- AP 模式
- STA 模式
- AP/STA 共存模式

说明
下文介绍的各种特性均基于这 3 种基本模式衍生而来。

4.4.3.1 AP 模式

在一个基础 BSS（Basic Service Set）网络中提供所有接入点的基本功能，包括：

- 发送 Beacon 帧声明 BSS 的存在和能力。
- 为 BSS 中的客户端提供无线关联和认证服务。
- 管理 BSS 网络中与之关联的客户端。

芯片支持 1 个 AP。

4.4.3.2 STA 模式

在一个基础 BSS 网络中提供扫描发现网络、加入网络并管理与 AP 的连接以提供数据收发服务的功能。

芯片支持 2 个 STA。

4.4.3.3 Monitor 模式

芯片进入 Monitor 模式，实现抓包网卡的功能，MAC 将接收到的所有帧上报软件。

4.4.3.4 AP 与 STA 共存

Hi3861、Hi3861L、Hi3881 均是具有单 MAC、单 PHY、双 RF 的芯片，无法实现 DBDC（Dual Band Dual Concurrent），但可以通过在频带间的快速切换（即分时复用）实现 DBAC（Dual Band Adaptive Concurrent）特性，即实现多个设备共存。

芯片支持 1 个 AP 和 1 个 STA 同时工作。

芯片支持 2.4G 下 AP/STA 在相同或不同信道的并发，分别对应同频共存和异频共存。

约束：STA 上电后会进行信道扫描，导致信道切换，因此开启 AP/STA 动态共存时，需要先创建 STA，再创建 AP，否则将会影响 AP 的工作信道。

4.4.3.5 CSI 模式

CSI（Channel State Information）模式支持将 PHY 上报的信道状态信息（CSI）过滤后上报软件：

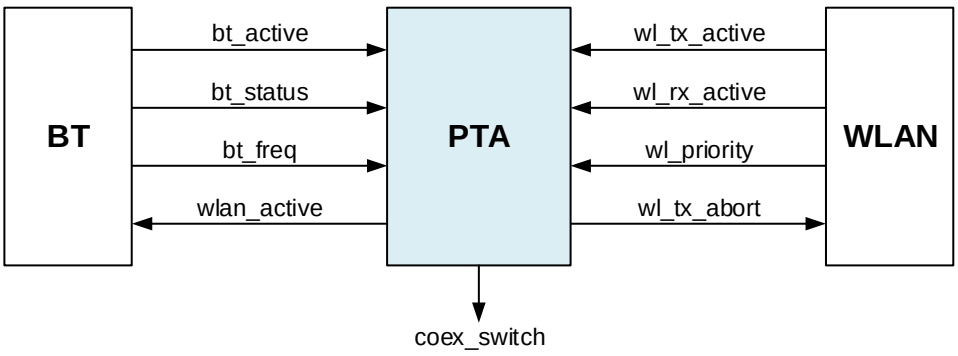
- 支持 11g/11n 的 CSI 信息上报，不支持 11b。
- 支持对将提取 CSI 的帧进行源地址过滤，源地址过滤列表（白名单）共 6 个（关联设备使用 LUT（Lookup Table）中的地址内容）。
- 支持 6 个 CSI 采样周期，CSI 采样周期与白名单绑定，一个白名单对应一个采集周期。
- 支持白名单、采样周期、特定帧类型等匹配条件，满足匹配条件才上报 CSI 信息。
- 支持 H 矩阵以 I/Q 格式上报，12bit 精度。
- 支持带宽（20MHz）、帧格式（NON-HT、HT-MF）、RSSI（Received Signal Strength Indicator）、SNR（Signal Noise Ratio）随 CSI 信息上报（不支持 STBC 帧上报）。上报 L-LTF H 矩阵数据。不支持 FTM（Fine Timing Measurement）。

4.5 通信包仲裁（PTA）

4.5.1 概述

PTA 是 BT 和 WLAN 通信包仲裁模块。提供一组 BT 请求接口和一组 WLAN 请求接口，用于识别请求的收发状态和优先级，并返回仲裁结果；BT 和 WLAN 共天线场景下，根据仲裁结果输出 coex_switch 信号，指示天线在 BT 和 WLAN 之间切换。

图4-3 PTA 应用框图



4.5.2 功能描述

BT 请求接口

BT 请求接口具有以下特点：

- 提供一组 BT 请求接口，用于识别 BT 收发请求和优先级，并返回仲裁结果。
- 支持 4 种共存模式：COEX_MODE_2A、COEX_MODE_2B、COEX_MODE_3、COEX_MODE_4。
- 支持 2 种仲裁模式：独立天线、共天线。
- 支持 BT 请求接口输入输出使能和有效电平配置。
- 仲裁结果 wlan_active 支持硬件控制或软件配置（软件配置优先级更高）。

WLAN 请求接口

WLAN 请求接口具有以下特点：

- 提供一组 WLAN 请求接口，用于识别 WLAN 收发请求和优先级，并返回仲裁结果。
- 支持 4 种共存模式：COEX_MODE_2A、COEX_MODE_2B、COEX_MODE_3、COEX_MODE_4。
- 支持 2 种仲裁模式：独立天线、共天线。
- 仲裁结果 wl_tx_abort 采用硬件控制，可支持软件屏蔽。

COEX_SWITCH 接口

COEX_SWITCH 接口具有以下特点：

- 输出 coex_switch 信号，指示天线在 BT 和 WLAN 之间切换。
- 支持 4 种共存模式：COEX_MODE_2A、COEX_MODE_2B、COEX_MODE_3、COEX_MODE_4。
- 仅在共天线场景有效，独立天线场景 coex_switch 信号无效。
- 天线仲裁结果 coex_switch 信号支持硬件控制或软件配置（软件配置优先级更高）。

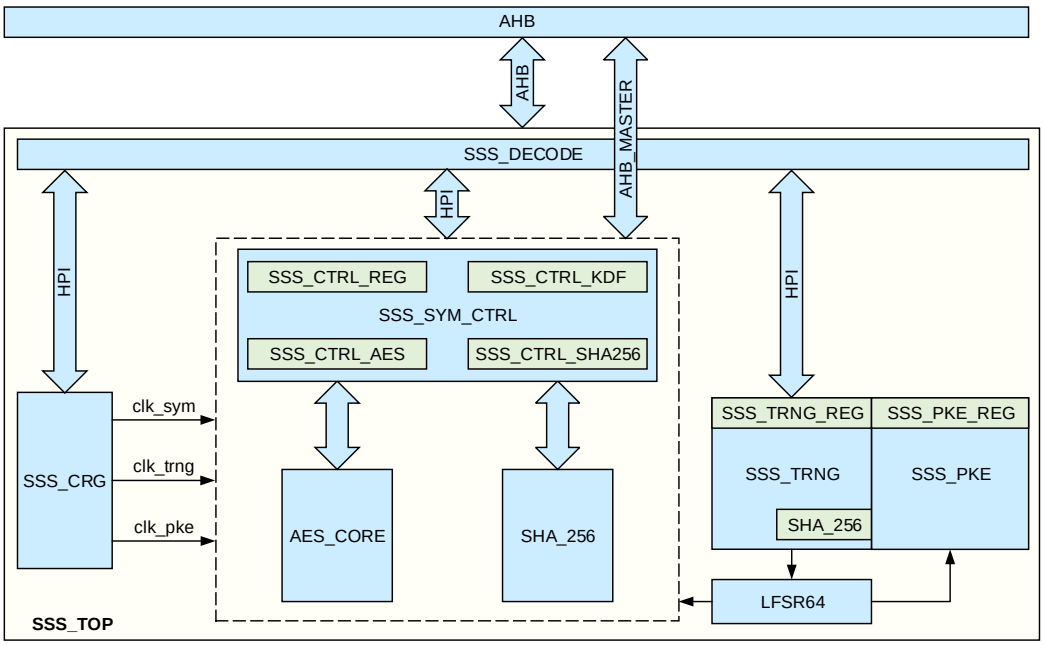
5 安全系统

5.1 安全子系统

5.1.1 概述

SSS 是一个安全子系统，包括 AES、HASH、KDF、PKE（Public-Key-Engine）、TRNG 子算法模块，通过 AHB 总线接口连接。SSS 逻辑框图如图 5-1 所示，CPU 通过 AHB 总线可以调用 SSS 中的 AES、HASH、KDF、PKE、TRNG 等算子实现相应功能的运算，并且可以通过对应的时钟开关寄存器控制算子时钟有无，从而达到降低功耗的作用。

图5-1 SSS 逻辑框图



注：HPI（Hardware Platform Interface）。

5.1.2 功能描述

SSS 具有以下功能特点：

- 支持 AES 算法。
- 支持 HASH 算法。
- 支持 KDF 算法。
- 支持 PKE 算法。
- 支持 TRNG 算法。

5.2 AES

5.2.1 概述

AES 是一个实现 AES 算法的模块，主要用于对数据的加解密。软件可以配置密钥、初始值、工作模式、加解密、密钥长度、待处理数据的长度、待处理数据的存放地址、处理结果是否回写、回写地址等参数，然后启动 AES 计算。

AES 算法的实现符合 FIPS197 标准。

5.2.2 功能描述

AES 模块具有以下功能特点：

- 支持 AES 加密和解密。
- 支持 AES 的 ECB（Electronic Code Book）、CBC（Cipher Block Chaining）、CTR（Counter）这 3 种工作模式。
- 支持 128bit、192bit、256bit 这 3 种密钥长度。
- 支持 DMA 方式读取待处理数据。
- 配置的数据长度必须按照块对齐，即 16byte 的倍数，支持的配置的最大数据长度为 4096byte。
- 支持软件配置和 KDF 派生这 2 种密钥来源，通过软件配置进行选择。
- 支持工作模式为 CBC 模式时，软件通过配置寄存器控制加解密结果是否通过 DMA 方式回写，不回写时软件可以通过读取逻辑寄存器得到最后的结果。
- 支持随机延时启动计算。

5.3 HASH

5.3.1 概述

HASH 是一个实现 SHA256 算法的模块，主要用于数据完整性认证。软件可以配置为初始值更新模式（需要配置初始值）或非初始值更新模式，然后启动 SHA256 算法计算消息摘要。

SHA256 算法的实现符合 FIPS180-2 标准。

5.3.2 功能描述

HASH 模块具有以下功能特点：

- 支持算法 SHA256。
- 支持 DMA 方式读取输入数据。
- 输入数据（软件填充之后）的总长度必须按照块对齐，即 64byte 的倍数，由软件配置，支持最长为 4096byte。
- SHA256：配置的数据长度为填充（需要软件完成）之后的消息长度（单位：byte）。
- SHA256 支持初始值可配置。

5.4 KDF

5.4.1 概述

KDF 是一个实现密钥派生的模块，根据输入派生不同的密钥。

KDF 实现符合 SP800-132 PBKDF2 标准。

5.4.2 功能描述

KDF 模块具有以下功能特点：

- 支持调用 HMAC-SHA256 实现 KDF。
- 支持满足 SP800-132 PBKDF2 标准。
- 支持 KDF 调用 HMAC-SHA256 的迭代次数可配置。
- 支持设备密钥和存储密钥派生。
- 支持 EFUSE 关闭 CPU 的读取存储密钥的权限。

5.5 PKE

5.5.1 概述

PKE 作为一个算法集合，PKE 可以实现非对称密码算法 RSA（Rivest-Shamir-Adleman）、ECC（Elliptic Curve Cryptography）的功能。PKE 通过 AHB 总线接口与外部控制模块进行通信，实现数据的交互。外部控制器也可以通过总线访问 PKE 模块的内部寄存器，即可以配置计算参数和读取计算结果或查询 PKE 模块的工作状态，从而控制 PKE 模块的工作。

5.5.2 功能描述

PKE 具有以下功能特点：

- 支持 RSA、ECC 两种非对称算法。
- RSA 支持 2048bit、3072bit、4096bit 三种密钥位宽；ECC 支持 256bit 一种密钥位宽。
- PKE 引擎支持基本算子功能（包括：模逆、模乘、基本模、大数乘法、模加、模减）。

5.6 TRNG

5.6.1 概述

TRNG 是一个能够产生真随机数的模块。

5.6.2 功能描述

TRNG 模块具有以下功能特点：

- 支持真随机数产生。
- 支持 8 个 RO（Ring Oscillator）环随机源选择。
- 支持 DRBG（Deterministic Random Bit Generator）符合 SP800-90A 标准要求。
- 支持 TRNG 产生的随机数性能＞4Mbit/s，RO 环采样时钟为 24M，系统时钟 160M。

5.7 EFUSE

5.7.1 概述

EFUSE IP 是一种可编程的存储单元，由于其仅可编程一次的特征，多用于芯片保存 Chip ID、密钥或其他一次性存储数据。

5.7.2 功能描述

EFUSE 模块具有以下功能特点：

- 存储容量为 2048bit。按照软件读写权限分为 4 个区域：
  - 软件可读可写区域
  - 软件不可读取的存储区域
  - 软件在 CPU 中断满足情况下可以读取的存储区域
  - 软件在 CPU 中断满足情况下可以烧写的存储区域

  EFUSE 区域划分如表 5-1 所示。

表5-1 EFUSE 区域划分

bit 域	位宽（bit）	软件可读	软件可写
bit[751:496]	256	否	是
bit[1671:1656]	16	是	CPU 中断控制
bit[1719:1672]	48	是	
bit[1847:1720]	128	CPU 中断控制	
bit[2036]	1	是	
bit[2037]	1	是	
bit[2038]	1	是	
其他	-	是	是

- bit[2047:0]区域分为 1 个锁定区域和 41 个子区域：
  - bit[2047:2012]：依次为区域 35～0 的锁定位。
  - bit[239:235]：依次为区域 40～36 的锁定位。

  如果区域 31 的锁定位烧写为 1，则锁定功能即刻生效，软件不能再烧写对应区域；如果为其他区域的锁定位烧写为 1，则需重新上电后，锁定功能才生效，软件不可再烧写对应区域。

  EFUSE 的具体排布表如表 5-2 所示。

表5-2 EFUSE 区域划分

字段	定义	bit 域	bit 数	锁定位	写入源	DFT 可读	DFT 可写	软件可读	软件可写
die_id	DIE ID	bit[199:8]	192	PG1	ATE	1	1	1	1
flash_encpt_cnt[7:6]	固件加解密计数 4	bit[221:220]	2	PG36	SW	1	1	1	1
flash_encpt_cnt[9:8]	固件加解密计数 5	bit[223:222]	2	PG37	SW	1	1	1	1
flash_encpt_cnt[11:10]	固件加解密计数 6	bit[225:224]	2	PG38	SW	1	1	1	1
rsvd0	-	bit[233:226]	8	PG39	SW	1	1	1	1
PG36	锁定位	bit[235]	1	-	SW	1	1	1	1
PG37	锁定位	bit[236]	1	-	SW	1	1	1	1
PG38	锁定位	bit[237]	1	-	SW	1	1	1	1
PG39	锁定位	bit[238]	1	-	SW	1	1	1	1

字段	定义	bit 域	bit 数	锁定位	写入源	DFT 可读	DFT 可写	软件可读	软件可写
PG40	锁定位	bit[239]	1	-	SW	1	1	1	1
root_pubkey	根公钥 HASH 值	bit[495:240]	256	PG3	SW	0	1	1	1
root_key	根秘钥	bit[751:496]	256	PG4	SW	0	1	0	1
uart_pubkey	UART 校验公钥 HASH 值	bit[1007:752]	256	PG5	SW	0	1	1	1
subkey_cat	二级密钥类别	bit[1039:1008]	32	PG6	SW	1	1	1	1
ivn	二级密钥吊销标识	bit[1071:1040]	32	PG7	SW	1	1	1	1
jtm	屏蔽 JTAG 接口	bit[1073]	1	PG8	SW	1	1	1	1
utm0	屏蔽 UART0 接口	bit[1074]	1	PG9	SW	1	1	1	1
utm1	屏蔽 UART1 接口	bit[1075]	1	PG10	SW	1	1	1	1
utm2	屏蔽 UART2 接口	bit[1076]	1	PG11	SW	1	1	1	1
sdc	安全 debug 控制位	bit[1077]	1	PG12	SW	1	1	1	1
kdf2ecc_huk_disable	HUK 参与的 KDF 密钥输出回读关闭使能信号	bit[1079]	1	PG35	ATE	1	1	1	1
boot_uart_sel	BOOT 启动时的 DBG UART 口选择	bit[1081:1080]	2	PG14	SW	1	1	1	1
uart_halt_interval	BOOT 启动时 DBG UART 中断等待时间	bit[1083:1082]	2	PG15	SW	1	1	1	1
ts_trim	Tsensor 温度校准码	bit[1087:1084]	4	PG2	ATE	1	1	1	1
abb_trim	ABB 内部寄存器调整值 1088bit： 0：非 SSS corner； 1：SSS corner	bit[1095:1088]	8	PG16	ATE	1	1	1	1
ipv4_mac_addr	IPV4 MAC 地址	bit[1143:1096]	48	PG17	SW	1	1	1	1
ipv6_mac_addr	IPV6 MAC 地址	bit[1271:1144]	128	PG17	SW	1	1	1	1
pa2gccka0_trim0	802.11b CORE0，功率校准系数，第 1 轮	bit[1303:1272]	32	PG18	SW	1	1	1	1
pa2gccka1_trim0	802.11b CORE0，功率校准系数，第 1 轮	bit[1335:1304]	32	PG18	SW	1	1	1	1

字段	定义	bit 域	bit 数	锁定位	写入源	DFT 可读	DFT 可写	软件可读	软件可写
nvram_pa2ga0_trim0	802.11g 20M，功率校准系数，第 1 轮	bit[1367:1336]	32	PG19	SW	1	1	1	1
nvram_pa2ga1_trim0	802.11g 20M，功率校准系数，第 1 轮	bit[1399:1368]	32	PG19	SW	1	1	1	1
pa2gccka0_trim1	802.11b CORE0，功率校准系数，第 2 轮	bit[1431:1400]	32	PG20	SW	1	1	1	1
pa2gccka1_trim1	802.11b CORE0，功率校准系数，第 2 轮	bit[1463:1432]	32	PG20	SW	1	1	1	1
nvram_pa2ga0_trim1	802.11g 20M，功率校准系数，第 2 轮	bit[1495:1464]	32	PG21	SW	1	1	1	1
nvram_pa2ga1_trim1	802.11g 20M，功率校准系数，第 2 轮	bit[1527:1496]	32	PG21	SW	1	1	1	1
pa2gccka0_trim2	802.11b CORE0，功率校准系数，第 3 轮	bit[1559:1528]	32	PG22	SW	1	1	1	1
pa2gccka1_trim2	802.11b CORE0，功率校准系数，第 3 轮	bit[1591:1560]	32	PG22	SW	1	1	1	1
nvram_pa2ga0_trim2	802.11g 20M，功率校准系数，第 3 轮	bit[1623:1592]	32	PG23	SW	1	1	1	1
nvram_pa2ga1_trim2	802.11g 20M，功率校准系数，第 3 轮	bit[1655:1624]	32	PG23	SW	1	1	1	1
tee_boot_ver	TEE BOOT 版本	bit[1671:1656]	16	PG24	SW	1	0	1	1*
tee_firmware_ver	TEE FIRMWARE 版本	bit[1719:1672]	48	PG25	SW	1	0	1	1*
tee_salt	TEE 盐值	bit[1847:1720]	128	PG26	SW	0	0	1*	1*
flash_encpt_cnt[1:0]	固件加解密计数 1	bit[1849:1848]	2	PG27	SW	1	1	1	1
flash_encpt_cnt[3:2]	固件加解密计数 2	bit[1851:1850]	2	PG28	SW	1	1	1	1
flash_encpt_cnt[5:4]	固件加解密计数 3	bit[1853:1852]	2	PG29	SW	1	1	1	1
flash_encpt_cfg	固件加解密控制位	bit[1854]	1	PG30	SW	1	1	1	1
flash_scramble_en	FLASH 数据加扰使能	bit[1855]	1	PG31	SW	1	1	1	1

字段	定义	bit 域	bit 数	锁定位	写入源	DFT 可读	DFT 可写	软件可读	软件可写
user_flash_ind	FLASH 数据加扰盐值	bit[1865:1856]	10	PG31	SW	1	1	1	1
rf_pdbuffer_gcal	RF 校准系数	bit[1883:1866]	18	PG32	ATE	1	1	1	1
customer_rsvd0	用户预留	bit[1947:1884]	64	PG33	SW	1	1	1	1
customer_rsvd1	用户预留	bit[2011:1948]	64	PG34	SW	1	1	1	1
PG0	锁定位	bit[2012]	1	-	ATE	1	1	1	1
PG1	锁定位	bit[2013]	1	-	ATE	1	1	1	1
PG2	锁定位	bit[2014]	1	-	ATE	1	1	1	1
PG3	锁定位	bit[2015]	1	-	ATE	1	1	1	1
PG4	锁定位	bit[2016]	1	-	SW	1	1	1	1
PG5	锁定位	bit[2017]	1	-	SW	1	1	1	1
PG6	锁定位	bit[2018]	1	-	SW	1	1	1	1
PG7	锁定位	bit[2019]	1	-	SW	1	1	1	1
PG8	锁定位	bit[2020]	1	-	SW	1	1	1	1
PG9	锁定位	bit[2021]	1	-	SW	1	1	1	1
PG10	锁定位	bit[2022]	1	-	SW	1	1	1	1
PG11	锁定位	bit[2023]	1	-	SW	1	1	1	1
PG12	锁定位	bit[2024]	1	-	SW	1	1	1	1
PG13	锁定位	bit[2025]	1	-	SW	1	1	1	1
PG14	锁定位	bit[2026]	1	-	SW	1	1	1	1
PG15	锁定位	bit[2027]	1	-	SW	1	1	1	1
PG16	锁定位	bit[2028]	1	-	SW	1	1	1	1
PG17	锁定位	bit[2029]	1	-	SW	1	1	1	1
PG18	锁定位	bit[2030]	1	-	SW	1	1	1	1
PG19	锁定位	bit[2031]	1	-	SW	1	1	1	1
PG20	锁定位	bit[2032]	1	-	SW	1	1	1	1
PG21	锁定位	bit[2033]	1	-	SW	1	1	1	1

字段	定义	bit 域	bit 数	锁定位	写入源	DFT 可读	DFT 可写	软件可读	软件可写
PG22	锁定位	bit[2034]	1	-	SW	1	1	1	1
PG23	锁定位	bit[2035]	1	-	SW	1	1	1	1
PG24	锁定位	bit[2036]	1	-	SW	1	0	1	1*
PG25	锁定位	bit[2037]	1	-	SW	1	0	1	1*
PG26	锁定位	bit[2038]	1	-	SW	1	0	1	1*
PG27	锁定位	bit[2039]	1	-	SW	1	1	1	1
PG28	锁定位	bit[2040]	1	-	SW	1	1	1	1
PG29	锁定位	bit[2041]	1	-	SW	1	1	1	1
PG30	锁定位	bit[2042]	1	-	SW	1	1	1	1
PG31	锁定位	bit[2043]	1	-	SW	1	1	1	1
PG32	锁定位	bit[2044]	1	-	SW	1	1	1	1
PG33	锁定位	bit[2045]	1	-	SW	1	1	1	1
PG34	锁定位	bit[2046]	1	-	SW	1	1	1	1
PG35	锁定位	bit[2047]	1	-	SW	1	1	1	1

6 外围设备

6.1 IO MUX

6.1.1 概述

芯片数字管脚数量有限，通过 IO 复用的方式丰富管脚功能。

6.1.2 软件复用管脚描述

须知

- RTC：Hi3861、Hi3861L、Hi3881 三款芯片的 RTC 方案如表 6-1 所示 Hi3861、Hi3881 芯片无外置 RTC 时钟，GPIO_00、GPIO_01 作为 GPIO 功能使用，可进行 IO 复用。Hi3861L 芯片有外置 RTC 时钟，当选用晶体时钟方案时，GPIO_00 为 RTC32K_XOUT 管脚，GPIO_01 为 RTC32K_XIN 管脚；当选用晶振时钟方案时，GPIO_00 为 RTC_OSC_32K 管脚，GPIO_01 作为 GPIO 功能使用，可进行 IO 复用）。
- ADC 管脚：LSADC 通道与 GPIO 功能只支持其中 1 种功能，ADC 通道管脚与 GPIO 管脚的对应关系如表 6-2 所示。

表6-1 RTC 时钟方案

Pad 信息	Hi3861 芯片	Hi3861L 芯片		Hi3881 芯片
		晶体	晶振	
GPIO_00	GPIO 功能，可 IO 复用	RTC32K_XOUT	RTC_OSC_32K	GPIO 功能，可 IO 复用
GPIO_01	GPIO 功能，可 IO 复用	RTC32K_XIN	GPIO 功能，可 IO 复用	GPIO 功，可 IO 复用

表6-2 ADC 通道管脚与复用管脚对应关系

复用管脚名称	ADC 管脚
GPIO_04	ADC1
GPIO_05	ADC2
GPIO_07	ADC3
GPIO_09	ADC4
GPIO_11	ADC5
GPIO_12	ADC0
GPIO_13	ADC6

软件复用管脚如表 6-3 所示。

表6-3 软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 0	复用信号 1	复用信号 2	复用信号 3	复用信号 4	复用信号 5	复用信号 6	复用信号 7
2	GPIO_00	GPIO_00_SEL	GPIO[0]	HW_ID[0]	UART1_TXD	SPI1_CK	JTAG_TDO	PWM3_OUT	I2C1_SDA	-
3	GPIO_01	GPIO_01_SEL	GPIO[1]	HW_ID[1]	UART1_RXD	SPI1_RXD	JTAG_TCK	PWM4_OUT	I2C1_SCL	BT_FREQ
4	GPIO_02	GPIO_02_SEL	GPIO[2]	REFCLK_FREQ_STATUS	UART1_RTS_N	SPI1_TXD	JTAG_TRSTN	PWM2_OUT	DIAG[0]	SSI_CLK
5	GPIO_03	GPIO_03_SEL	GPIO[3]	UART0_TXD	UART1_CTS_N	SPI1_CSN	JTAG_TDI	PWM5_OUT	I2C1_SDA	SSI_DATA
6	GPIO_04	GPIO_04_SEL	GPIO[4]	HW_ID[3]	UART0_RXD	JTAG_TMS	PWM1_OUT	I2C1_SCL	DIAG[7]	-
17	GPIO_05	GPIO_05_SEL	GPIO[5]	HW_ID[4]	UART1_RXD	SPI0_CSN	DIAG[1]	PWM2_OUT	I2S0_MCLK	BT_STATUS
18	GPIO_06	GPIO_06_SEL	GPIO[6]	JTAG_MODE	UART1_TXD	SPI0_CK	DIAG[2]	PWM3_OUT	I2S0_TX	COEX_SWITCH
19	GPIO_07	GPIO_07_SEL	GPIO[7]	HW_ID[5]	UART1_CTS_N	SPI0_RXD	DIAG[3]	PWM0_OUT	I2S0_BCLK	BT_ACTIVE
20	GPIO_08	GPIO_08_SEL	GPIO[8]	JTAG_ENABLE	UART1_RTS_N	SPI0_TXD	DIAG[4]	PWM1_OUT	I2S0_WS	WLAN_ACTIVE
27	GPIO_09	GPIO_09_SEL	GPIO[9]	I2C0_SCL	UART2_RTS_N	SDIO_D2	SPI0_TXD	PWM0_OUT	DIAG[5]	I2S0_MCLK
28	GPIO_10	GPIO_10_SEL	GPIO[10]	I2C0_SDA	UART2_CTS_N	SDIO_D3	SPI0_CK	PWM1_OUT	DIAG[6]	I2S0_TX

Pin	Pad 信号	复用控制寄存器	复用信号 0	复用信号 1	复用信号 2	复用信号 3	复用信号 4	复用信号 5	复用信号 6	复用信号 7
29	GPIO_11	GPIO_11_SEL	GPIO[11]	HW_ID[6]	UART2_TXD	SDIO_CMD	SPI0_RXD	PWM2_OUT	RF_TX_EN_EXT	I2S0_RX
30	GPIO_12	GPIO_12_SEL	GPIO[12]	HW_ID[7]	UART2_RXD	SDIO_CLK	SPI0_CSN	PWM3_OUT	RF_RX_EN_EXT	I2S0_BCLK
31	GPIO_13	GPIO_13_SEL	SSI_DATA	UART0_TXD	UART2_RTS_N	SDIO_D0	GPIO[13]	PWM4_OUT	I2C0_SDA	I2S0_WS
32	GPIO_14	GPIO_14_SEL	SSI_CLK	UART0_RXD	UART2_CTS_N	SDIO_D1	GPIO[14]	PWM5_OUT	I2C0_SCL	HW_ID[2]
27	SFC_CSN	SFC_CSN_SEL	SFC_CSN	SDIO_D2	GPIO[9]	DIAG[5]	SPI0_TXD	-	-	-
28	SFC_IO1	SFC_IO1_SEL	SFC_DO	SDIO_D3	GPIO[10]	DIAG[6]	SPI0_CK	-	-	-
29	SFC_IO2	SFC_IO2_SEL	SFC_WPN	SDIO_CMD	GPIO[11]	RF_TX_EN_EXT	SPI0_RXD	-	-	-
30	SFC_IO0	SFC_IO0_SEL	SFC_DI	SDIO_CLK	GPIO[12]	RF_RX_EN_EXT	SPI0_CSN	-	-	-
31	SFC_CLK	SFC_CLK_SEL	SFC_CLK	SDIO_D0	GPIO[13]	SSI_DATA	-	-	-	-
32	SFC_IO3	SFC_IO3_SEL	SFC_HOLDN	SDIO_D1	GPIO[14]	SSI_CLK	-	-	-	-

注：标红管脚表示不同芯片封装功能有差异，差异点请参见以下注意说明。

须知

- Hi3881 芯片 SFC_CSN、SFC_IO1、SFC_IO2、SFC_IO0、SFC_CLK、SFC_IO3 作为 27～32 号管脚，GPIO_09～GPIO_14 不作为管脚输出。
- Hi3881 芯片没有 UART2、I2C0、HW_ID2、HW_ID6、HW_ID7 功能。
- Hi3881 芯片请勿将 GPIO_09～GPIO_14 复用功能与 SFC_CSN、SFC_IO1、SFC_IO2、SFC_IO0、SFC_CLK、SFC_IO3 复用功能配置为相同，以防止 SFC 管脚被复用为非 SFC 功能时该管脚不可用，如果 GPIO_09_SEL 配置为 3 即 SDIO 功能时，SFC_CSN_SEL 配置为 1 也为 SDIO 功能，将导致 SFC_CSN 管脚无法使用 SDIO 功能。为防止上述情况，建议将 GPIO_09_SEL～GPIO_14_SEL 配置为 0x2 或 0x7。
- Hi3861、Hi3861L 芯片 GPIO_09～GPIO_14 作为 27～32 号管脚，SFC_CSN、SFC_IO1、SFC_IO2、SFC_IO0、SFC_CLK、SFC_IO3 不作为管脚输出。
- Hi3861、Hi3861L 芯片请将 SFC_CSN、SFC_IO1、SFC_IO2、SFC_IO0、SFC_CLK、SFC_IO3 管脚配置为 SFC 功能，否则将导致内置 Flash 不可用。

GPIO 的软件复用管脚说明如表 6-4 所示。

表6-4 GPIO 的软件复用管脚说明

信号名	方向	说明
BT_ACTIVE	I	BT 业务请求信号。
BT_FREQ	I	BT 信道状态信号。
BT_STATUS	I	BT 业务状态信号。
COEX_SWITCH	O	天线切换指示信号。
DIAG[0]	O	维测信号。
DIAG[1]	O	维测信号。
DIAG[2]	O	维测信号。
DIAG[3]	O	维测信号。
DIAG[4]	O	维测信号。
DIAG[5]	O	维测信号。
DIAG[6]	O	维测信号。
DIAG[7]	O	维测信号。
GPIO[0]	I/O	GPIO 管脚信号。
GPIO[1]	I/O	GPIO 管脚信号。

信号名	方向	说明
GPIO[10]	I/O	GPIO 管脚信号。
GPIO[11]	I/O	GPIO 管脚信号。
GPIO[12]	I/O	GPIO 管脚信号。
GPIO[13]	I/O	GPIO 管脚信号。
GPIO[14]	I/O	GPIO 管脚信号。
GPIO[2]	I/O	GPIO 管脚信号。
GPIO[3]	I/O	GPIO 管脚信号。
GPIO[4]	I/O	GPIO 管脚信号。
GPIO[5]	I/O	GPIO 管脚信号。
GPIO[6]	I/O	GPIO 管脚信号。
GPIO[7]	I/O	GPIO 管脚信号。
GPIO[8]	I/O	GPIO 管脚信号。
GPIO[9]	I/O	GPIO 管脚信号。
HW_ID[0]	I	上电硬件控制字。
HW_ID[1]	I	上电硬件控制字。
HW_ID[2]	I	上电硬件控制字。
HW_ID[3]	I	上电硬件控制字。
HW_ID[4]	I	上电硬件控制字。
HW_ID[5]	I	上电硬件控制字。
HW_ID[6]	I	上电硬件控制字。
HW_ID[7]	I	上电硬件控制字。
I2C0_SCL	I/O	I2C 时钟。
I2C0_SDA	I/O	I2C 数据/nSET。
I2C1_SCL	I/O	I2C 时钟。
I2C1_SDA	I/O	I2C 数据/nSET。
I2S0_BCLK	O	I2S 工作钟信号。
I2S0_MCLK	O	I2S 主时钟信号。
I2S0_RX	I	I2S 数据接收信号。
I2S0_TX	O	I2S 数据发送信号。

信号名	方向	说明
I2S0_WS	O	I2S 声道选择信号。
JTAG_ENABLE	I	JTAG 使能。 0：普通 IO； 1：JTAG 使能。
JTAG_MODE	I	JTAG 模式选择信号。 0：正常功能模式； 1：DFT（Design For Testability）测试模式。
JTAG_TCK	I	JTAG 时钟输入。
JTAG_TDI	I	JTAG 数据输入。
JTAG_TDO	I/O	JTAG 数据输出。
JTAG_TMS	I/O	JTAG 模式选择输入。
JTAG_TRSTN	I	JTAG 复位输入，低电平有效。默认状态为复位。
PWM0_OUT	O	PWM0 输出信号。
PWM1_OUT	O	PWM1 输出信号。
PWM2_OUT	O	PWM2 输出信号。
PWM3_OUT	O	PWM3 输出信号。
PWM4_OUT	O	PWM4 输出信号。
PWM5_OUT	O	PWM5 输出信号。
REFCLK_FREQ_STATUS	I	晶体时钟频率的指示信号。 0：40M； 1：24M。
RF_RX_EN_EXT	I	外部输入的 RF RX 使能信号。
RF_TX_EN_EXT	I	外部输入的 RF TX 使能信号。
SDIO_CLK	I	SDIO 时钟信号。
SDIO_CMD	I/O	SDIO 命令信号。
SDIO_D0	I/O	SDIO 数据信号 0。
SDIO_D1	I/O	SDIO 数据信号 1。
SDIO_D2	I/O	SDIO 数据信号 2。
SDIO_D3	I/O	SDIO 数据信号 3。

信号名	方向	说明
SPI0_CK	I/O	SPI0 时钟信号。
SPI0_CSN	I/O	SPI0 片选信号。
SPI0_RXD	I	SPI0 数据接收信号。
SPI0_TXD	I/O	SPI0 数据发送信号。
SPI1_CK	I/O	SPI1 时钟信号。
SPI1_CSN	I/O	SPI1 片选信号。
SPI1_RXD	I	SPI1 数据接收信号。
SPI1_TXD	I/O	SPI1 数据发送信号。
SSI_CLK	I	SSI 时钟信号。
SSI_DATA	I/O	SSI 数据信号。
UART0_RXD	I	主板通信 UART RX。
UART0_TXD	O	主板通信 UART TX。
UART1_CTS_N	I	UART1 流控信号。
UART1_RTS_N	O	UART1 流控信号。
UART1_RXD	I	调试 UART1 RX。
UART1_TXD	O	调试 UART1 TX。
UART2_CTS_N	I	UART2 流控信号。
UART2_RTS_N	O	UART2 流控信号。
UART2_RXD	I	调试 UART2 RX。
UART2_TXD	O	调试 UART2 TX。
WLAN_ACTIVE	O	WLAN 业务状态指示信号。
SFC_CLK	O	Flash 控制信号。 Flash 时钟范围：CMU 中 PLL 产生 96M、80M、60M、48M 的时钟。 上电使用晶体时钟：20M 或 12M。
SFC_CSN	O	Flash 片选信号。
SFC_DI	I/O	Flash 数据信号 0。
SFC_DO	I/O	Flash 数据信号 1。
SFC_HOLDN	I/O	Flash 数据信号 3。

信号名	方向	说明
SFC_WPN	I/O	Flash 数据信号 2。

6.1.3 复用寄存器概览

复用寄存器概览如表 6-1 所示。

表6-5 复用寄存器概览（基址是 0x5000_A000）

偏移地址	名称	描述	页码
0x604	GPIO_00_SEL	GPIO_00 管脚的复用控制寄存器	6-10
0x608	GPIO_01_SEL	GPIO_01 管脚的复用控制寄存器	6-10
0x60C	GPIO_02_SEL	GPIO_02 管脚的复用控制寄存器	6-11
0x610	GPIO_03_SEL	GPIO_03 管脚的复用控制寄存器	6-11
0x614	GPIO_04_SEL	GPIO_04 管脚的复用控制寄存器	6-11
0x618	GPIO_05_SEL	GPIO_05 管脚的复用控制寄存器	6-12
0x61C	GPIO_06_SEL	GPIO_06 管脚的复用控制寄存器	6-12
0x620	GPIO_07_SEL	GPIO_07 管脚的复用控制寄存器	6-13
0x624	GPIO_08_SEL	GPIO_08 管脚的复用控制寄存器	6-13
0x628	GPIO_09_SEL	GPIO_09 管脚的复用控制寄存器	6-14
0x62C	GPIO_10_SEL	GPIO_10 管脚的复用控制寄存器	6-14
0x630	GPIO_11_SEL	GPIO_11 管脚的复用控制寄存器	6-15
0x634	GPIO_12_SEL	GPIO_12 管脚的复用控制寄存器	6-15
0x638	GPIO_13_SEL	GPIO_13 管脚的复用控制寄存器	6-16
0x63C	GPIO_14_SEL	GPIO_14 管脚的复用控制寄存器	6-16
0x640	SFC_CSN_SEL	SFC_CSN 管脚的复用控制寄存器	6-17
0x644	SFC_IO1_SEL	SFC_IO1 管脚的复用控制寄存器	6-17
0x648	SFC_IO2_SEL	SFC_IO2 管脚的复用控制寄存器	6-18
0x64C	SFC_IO0_SEL	SFC_IO0 管脚的复用控制寄存器	6-18
0x650	SFC_CLK_SEL	SFC_CLK 管脚的复用控制寄存器	6-18
0x654	SFC_IO3_SEL	SFC_IO3 管脚的复用控制寄存器	6-19

6.1.4 复用寄存器描述

GPIO_00_SEL

GPIO_00_SEL 为 GPIO_00 管脚复用控制寄存器。

Offset Address: 0x604　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_00_sel	GPIO_00 管脚的具体复用情况。 000：GPIO[0]； 001：HW_ID[0]； 010：UART1_TXD； 011：SPI1_CK； 100：JTAG_TDO； 101：PWM3_OUT； 110：I2C1_SDA； 其他：保留。	0x0

GPIO_01_SEL

GPIO_01_SEL 为 GPIO_01 管脚复用控制寄存器。

Offset Address: 0x608　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_01_sel	GPIO_01 管脚的具体复用情况。 000：GPIO[1]； 001：HW_ID[1]； 010：UART1_RXD； 011：SPI1_RXD； 100：JTAG_TCK； 101：PWM4_OUT； 110：I2C1_SCL； 111：BT_FREQ。	0x0

GPIO_02_SEL

GPIO_02_SEL 为 GPIO_02 管脚复用控制寄存器。

Offset Address: 0x60C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_02_sel	GPIO_02 管脚的具体复用情况。 000：GPIO[2]； 001：REFCLK_FREQ_STATUS； 010：UART1_RTS_N； 011：SPI1_TXD； 100：JTAG_TRSTN； 101：PWM2_OUT； 110：DIAG[0]； 111：SSI_CLK。	0x0

GPIO_03_SEL

GPIO_03_SEL 为 GPIO_03 管脚复用控制寄存器。

Offset Address: 0x610　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_03_sel	GPIO_03 管脚的具体复用情况。 000：GPIO[3]； 001：UART0_TXD； 010：UART1_CTS_N； 011：SPI1_CSN； 100：JTAG_TDI； 101：PWM5_OUT； 110：I2C1_SDA； 111：SSI_DATA。	0x0

GPIO_04_SEL

GPIO_04_SEL 为 GPIO_04 管脚复用控制寄存器。

Offset Address: 0x614 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_04_sel	GPIO_04 管脚的具体复用情况。 000：GPIO[4]； 001：HW_ID[3]； 010：UART0_RXD； 100：JTAG_TMS； 101：PWM1_OUT； 110：I2C1_SCL； 111：DIAG[7]； 其他：保留。	0x0

GPIO_05_SEL

GPIO_05_SEL 为

Offset Address: 0x618 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_05_sel	GPIO_05 管脚的具体复用情况。 000：GPIO[5]； 001：HW_ID[4]； 010：UART1_RXD； 011：SPI0_CSN； 100：DIAG[1]； 101：PWM2_OUT； 110：I2S0_MCLK； 111：BT_STATUS。	0x0

GPIO_06_SEL

GPIO_06_SEL 为 GPIO_06 管脚复用控制寄存器。

Offset Address: 0x61C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_06_sel	GPIO_06 管脚的具体复用情况。 000：GPIO[6]； 001：JTAG_MODE； 010：UART1_TXD； 011：SPI0_CK； 100：DIAG[2]； 101：PWM3_OUT； 110：I2S0_TX； 111：COEX_SWITCH。	0x0

GPIO_07_SEL

GPIO_07_SEL 为 GPIO_07 管脚复用控制寄存器。

Offset Address: 0x620　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_07_sel	GPIO_07 管脚的具体复用情况。 000：GPIO[7]； 001：HW_ID[5]； 010：UART1_CTS_N； 011：SPI0_RXD； 100：DIAG[3]； 101：PWM0_OUT； 110：I2S0_BCLK； 111：BT_ACTIVE。	0x0

GPIO_08_SEL

GPIO_08_SEL 为 GPIO_08 管脚复用控制寄存器。

Offset Address: 0x　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000

[2:0]	RW	gpio_08_sel	GPIO_08 管脚的具体复用情况。 000：GPIO[8]； 001：JTAG_ENABLE； 010：UART1_RTS_N； 011：SPI0_TXD； 100：DIAG[4]； 101：PWM1_OUT； 110：I2S0_WS； 111：WLAN_ACTIVE。	0x0

GPIO_09_SEL

GPIO_09_SEL 为 GPIO_09 管脚复用控制寄存器。

Offset Address: 0x628　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_09_sel	GPIO_09 管脚的具体复用情况。 000：GPIO[9]； 001：I2C0_SCL； 010：UART2_RTS_N； 011：SDIO_D2； 100：SPI0_TXD； 101：PWM0_OUT； 110：DIAG[5]； 111：I2S0_MCLK。	0x0

GPIO_10_SEL

GPIO_10_SEL 为 GPIO_10 管脚复用控制寄存器。

Offset Address: 0x62C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_10_sel	GPIO_10 管脚的具体复用情况。 000：GPIO[10]；	0x0

			001：I2C0_SDA； 010：UART2_CTS_N； 011：SDIO_D3； 100：SPI0_CK； 101：PWM1_OUT； 110：DIAG[6]； 111：I2S0_TX。	

GPIO_11_SEL

GPIO_11_SEL 为 GPIO_11 管脚复用控制寄存器。

Offset Address: 0x630　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_11_sel	GPIO_11 管脚的具体复用情况。 000：GPIO[11]； 001：HW_ID[6]； 010：UART2_TXD； 011：SDIO_CMD； 100：SPI0_RXD； 101：PWM2_OUT； 110：RF_TX_EN_EXT； 111：I2S0_RX。	0x0

GPIO_12_SEL

GPIO_12_SEL 为 GPIO_12 管脚复用控制寄存器。

Offset Address: 0x634　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_12_sel	GPIO_12 管脚的具体复用情况。 000：GPIO[12]； 001：HW_ID[7]； 010：UART2_RXD；	0x0

			011：SDIO_CLK； 100：SPI0_CSN； 101：PWM3_OUT； 110：RF_RX_EN_EXT； 111：I2S0_BCLK。	

GPIO_13_SEL

GPIO_13_SEL 为 GPIO_13 管脚复用控制寄存器。

Offset Address: 0x638　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_13_sel	GPIO_13 管脚的具体复用情况。 000：SSI_DATA； 001：UART0_TXD； 010：UART2_RTS_N； 011：SDIO_D0； 100：GPIO[13]； 101：PWM4_OUT； 110：I2C0_SDA； 111：I2S0_WS。	0x0

GPIO_14_SEL

GPIO_14_SEL 为 GPIO_14 管脚复用控制寄存器。

Offset Address: 0x63C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	gpio_14_sel	GPIO_14 管脚的具体复用情况。 000：SSI_CLK； 001：UART0_RXD； 010：UART2_CTS_N； 011：SDIO_D1； 100：GPIO[14]；	0x0

			101：PWM5_OUT； 110：I2C0_SCL； 111：HW_ID[2]。	

SFC_CSN_SEL

SFC_CSN_SEL 为

Offset Address: 0x640　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	sfc_csn_sel	SFC_CSN 管脚的具体复用情况。 000：SFC_CSN； 001：SDIO_D2； 010：GPIO[9]； 011：DIAG[5]； 100：SPI0_TXD； 其他：保留。	0x0

SFC_IO1_SEL

SFC_IO1_SEL 为 SFC_IO1 管脚复用控制寄存器。

Offset Address: 0x644　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	sfc_io1_sel	SFC_IO1 管脚的具体复用情况。 000：SFC_DO； 001：SDIO_D3； 010：GPIO[10]； 011：DIAG[6]； 100：SPI0_CK； 其他：保留。	0x0

SFC_IO2_SEL

SFC_IO2_SEL 为 SFC_IO2 管脚复用控制寄存器。

Offset Address: 0x648　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	sfc_io2_sel	SFC_IO2 管脚的具体复用情况。 000：SFC_WPN； 001：SDIO_CMD； 010：GPIO[11]； 011：RF_TX_EN_EXT； 100：SPI0_RXD； 其他：保留。	0x0

SFC_IO0_SEL

SFC_IO0_SEL 为 SFC_IO0 管脚复用控制寄存器。

Offset Address: 0x64C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	sfc_io0_sel	SFC_IO0 管脚的具体复用情况。 000：SFC_DI； 001：SDIO_CLK； 010：GPIO[12]； 011：RF_RX_EN_EXT； 100：SPI0_CSN； 其他：保留。	0x0

SFC_CLK_SEL

SFC_CLK_SEL 为 SFC_CLK 管脚复用控制寄存器。

Offset Address: 0x650　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000

[2:0]	RW	sfc_clk_sel	SFC_CLK 管脚的具体复用情况。 000：SFC_CLK； 001：SDIO_D0； 010：GPIO[13]； 100：SSI_DATA； 其他：保留。	0x0

SFC_IO3_SEL

SFC_IO3_SEL 为 SFC_IO3 管脚复用控制寄存器。

Offset Address: 0x654　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	sfc_io3_sel	SFC_IO3 管脚的具体复用情况。 000：SFC_HOLDN； 001：SDIO_D1； 010：GPIO[14]； 100：SSI_CLK； 其他：保留。	0x0

6.1.5 控制寄存器概览

可以通过配置 DS0（Drive Strength）、DS1、DS2 的值改变芯片管脚的驱动电流的强度，驱动强度与 DS0、DS1、DS2 的对应关系如表 6-6、表 6-7 所示，控制寄存器名称与芯片管脚名称一一对应。

表6-6 GPIO_00～GPIO_11、GPIO_13、GPIO_14 驱动强度

DS1	DS0	驱动强度（mA）
0	0	8
0	1	6
1	0	4
1	1	2

表6-7 GPIO_12、SFC 管脚驱动强度

DS2	DS1	DS0	驱动强度（mA）
0	0	0	16
0	0	1	14
0	1	0	12
0	1	1	10
1	0	0	8
1	0	1	6
1	1	0	4
1	1	1	2

控制寄存器概览如表 6-8 所示。

表6-8 控制寄存器概览（基址是 0x5000_A000）

偏移地址	名称	描述	页码
0x904	pad_gpio_00_ctrl	GPIO_00 功能管脚控制寄存器	6-21
0x908	pad_gpio_01_ctrl	GPIO_01 功能管脚控制寄存器	6-22
0x90C	pad_gpio_02_ctrl	GPIO_02 功能管脚控制寄存器	6-23
0x910	pad_gpio_03_ctrl	GPIO_03 功能管脚控制寄存器	6-24
0x914	pad_gpio_04_ctrl	GPIO_04 功能管脚控制寄存器	6-25
0x918	pad_gpio_05_ctrl	GPIO_05 功能管脚控制寄存器	6-26
0x91C	pad_gpio_06_ctrl	GPIO_06 功能管脚控制寄存器	6-26
0x920	pad_gpio_07_ctrl	GPIO_07 功能管脚控制寄存器	6-27
0x924	pad_gpio_08_ctrl	GPIO_08 功能管脚控制寄存器	6-28
0x928	pad_gpio_09_ctrl	GPIO_09 功能管脚控制寄存器	6-29
0x92C	pad_gpio_10_ctrl	GPIO_10 功能管脚控制寄存器	6-30
0x930	pad_gpio_11_ctrl	GPIO_11 功能管脚控制寄存器	6-31
0x934	pad_gpio_12_ctrl	GPIO_12 功能管脚控制寄存器	6-31
0x938	pad_gpio_13_ctrl	GPIO_13 功能管脚控制寄存器	6-32
0x93C	pad_gpio_14_ctrl	GPIO_14 功能管脚控制寄存器	6-33
0x940	pad_sfc_csn_ctrl	SFC_CSN 功能管脚控制寄存器	6-34

偏移地址	名称	描述	页码
0x944	pad_sfc_io1_ctrl	SFC_IO1 功能管脚控制寄存器	6-35
0x948	pad_sfc_io2_ctrl	SFC_IO2 功能管脚控制寄存器	6-35
0x94C	pad_sfc_io0_ctrl	SFC_IO0 功能管脚控制寄存器	6-36
0x950	pad_sfc_clk_ctrl	SFC_CLK 功能管脚控制寄存器	6-37
0x954	pad_sfc_io3_ctrl	SFC_IO3 功能管脚控制寄存器	6-38

6.1.6 控制寄存器描述

pad_gpio_00_ctrl

pad_gpio_00_ctrl 为 GPIO_00 功能管脚控制寄存器。

Offset Address: 0x904　　Total Reset Value: 0x0E0B_44B0

Bits	Access	Name	Description	Reset
[31:28]	RW	reseverd	保留。	0x0
[27]	RW	pad_gpio_00_ctrl_osc_ds2	XTAL（Quartz Crystal Unit）模式下的驱动电流控制。 [ds2,ds1,ds0]： 000：0.6μA； 001：0.7μA； 010：0.8μA； 011：0.9μA； 100：1.0μA； 101：1.2μA； 110：1.4μA； 111：1.6μA。	0x1
[26]	RW	pad_gpio_00_ctrl_osc_ds1		0x1
[25]	RW	pad_gpio_00_ctrl_osc_ds0		0x1
[24]	RW	reseverd	保留。	0x0
[23]	RW	pad_gpio_00_ctrl_osc_ctl0	OSC 控制信号。 0：外接晶振； 1：外接晶体。	0x0
[22]	RW	pad_gpio_00_ctrl_osc_e	RTC 使能信号。 0：禁止； 1：使能。	0x0
[21:11]	RW	reserved	保留。	0x168

[10]	RW	pad_gpio_00_ctrl_ie1	GPIO_00 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_00_ctrl_pd1	GPIO_00 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_00_ctrl_pu1	GPIO_00 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_00_ctrl_sr1	GPIO_00 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_00_ctrl_ds11	GPIO_00 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_00_ctrl_ds01	GPIO_00 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_00_ctrl_se1	GPIO_00 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_01_ctrl

pad_gpio_01_ctrl 为 GPIO_01 功能管脚控制寄存器。

Offset Address: 0x908　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_01_ctrl_ie	GPIO_01 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_01_ctrl_pd	GPIO_01 输入下拉使能。 0：禁止； 1：使能。	0x0

[8]	RW	pad_gpio_01_ctrl_pu	GPIO_01 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_01_ctrl_sr	GPIO_01 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_01_ctrl_ds1	GPIO_01 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_01_ctrl_ds0	GPIO_01 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_01_ctrl_se	GPIO_01 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_02_ctrl

pad_gpio_02_ctrl 为 GPIO_02 功能管脚控制寄存器。

Offset Address: 0x90C　　Total Reset Value: 0x0000_06B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_02_ctrl_ie	GPIO_02 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_02_ctrl_pd	GPIO_02 输入下拉使能。 0：禁止； 1：使能。	0x1
[8]	RW	pad_gpio_02_ctrl_pu	GPIO_02 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_02_ctrl_sr	GPIO_02 Slew Rate。 0：快沿； 1：慢沿。	0x1

[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_02_ctrl_ds1	GPIO_02 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_02_ctrl_ds0	GPIO_02 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_02_ctrl_se	GPIO_02 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_03_ctrl

pad_gpio_03_ctrl 为 GPIO_03 功能管脚控制寄存器。

Offset Address: 0x910　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_03_ctrl_ie	GPIO_03 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_03_ctrl_pd	GPIO_03 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_03_ctrl_pu	GPIO_03 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_03_ctrl_sr	GPIO_03 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_03_ctrl_ds1	GPIO_03 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_03_ctrl_ds0	GPIO_03 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_03_ctrl_se	GPIO_03 施密特触发器使能控制。	0x0

			0：No Schmitt； 1：Schmitt Enable。	
[2:0]	RW	reserved	保留。	0x0

pad_gpio_04_ctrl

pad_gpio_04_ctrl 为 GPIO_04 功能管脚控制寄存器。

Offset Address: 0x914　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_04_ctrl_ie	GPIO_04 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_04_ctrl_pd	GPIO_04 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_04_ctrl_pu	GPIO_04 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_04_ctrl_sr	GPIO_04 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_04_ctrl_ds1	GPIO_04 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_04_ctrl_ds0	GPIO_04 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_04_ctrl_se	GPIO_04 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_05_ctrl

pad_gpio_05_ctrl 为 GPIO_05 功能管脚控制寄存器。

Offset Address: 0x918　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_05_ctrl_ie	GPIO_05 输入信号能使。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_05_ctrl_pd	GPIO_05 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_05_ctrl_pu	GPIO_05 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_05_ctrl_sr	GPIO_05 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_05_ctrl_ds1	GPIO_05 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_05_ctrl_ds0	GPIO_05 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_05_ctrl_se	GPIO_05 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_06_ctrl

pad_gpio_06_ctrl 为 GPIO_06 功能管脚控制寄存器。

Offset Address: 0x91C　　Total Reset Value: 0x0000_06B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000

[10]	RW	pad_gpio_06_ctrl_ie	GPIO_06 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_06_ctrl_pd	GPIO_06 输入下拉使能。 0：禁止； 1：使能。	0x1
[8]	RW	pad_gpio_06_ctrl_pu	GPIO_06 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_06_ctrl_sr	GPIO_06 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_06_ctrl_ds1	GPIO_06 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_06_ctrl_ds0	GPIO_06 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_06_ctrl_se	GPIO_06 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_07_ctrl

pad_gpio_07_ctrl 为 GPIO_07 功能管脚控制寄存器。

Offset Address: 0x920　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_07_ctrl_ie	GPIO_07 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_07_ctrl_pd	GPIO_07 输入下拉使能。 0：禁止； 1：使能。	0x0

[8]	RW	pad_gpio_07_ctrl_pu	GPIO_07 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_07_ctrl_sr	GPIO_07 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_07_ctrl_ds1	GPIO_07 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_07_ctrl_ds0	GPIO_07 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_07_ctrl_se	GPIO_07 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_08_ctrl

pad_gpio_08_ctrl 为 GPIO_08 功能管脚控制寄存器。

Offset Address: 0x924　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_08_ctrl_ie	GPIO_08 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_08_ctrl_pd	GPIO_08 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_08_ctrl_pu	GPIO_08 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_08_ctrl_sr	GPIO_08 Slew Rate。 0：快沿； 1：慢沿。	0x1

[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_08_ctrl_ds1	GPIO_08 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_08_ctrl_ds0	GPIO_08 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_08_ctrl_se	GPIO_08 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_09_ctrl

pad_gpio_09_ctrl 为 GPIO_09 功能管脚控制寄存器。

Offset Address: 0x928　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_09_ctrl_ie	GPIO_09 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_09_ctrl_pd	GPIO_09 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_09_ctrl_pu	GPIO_09 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_09_ctrl_sr	GPIO_09 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_09_ctrl_ds1	GPIO_09 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_09_ctrl_ds0	GPIO_09 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_09_ctrl_se	GPIO_09 施密特触发器使能控制。	0x0

			0：No Schmitt； 1：Schmitt Enable。	
[2:0]	RW	reserved	保留。	0x0

pad_gpio_10_ctrl

pad_gpio_10_ctrl 为 GPIO_10 功能管脚控制寄存器。

Offset Address: 0x92C　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_10_ctrl_ie	GPIO_10 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_10_ctrl_pd	GPIO_10 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_10_ctrl_pu	GPIO_10 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_10_ctrl_sr	GPIO_10 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_10_ctrl_ds1	GPIO_10 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_10_ctrl_ds0	GPIO_10 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_10_ctrl_se	GPIO_10 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_11_ctrl

pad_gpio_11_ctrl 为 GPIO_11 功能管脚控制寄存器。

Offset Address: 0x930　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_11_ctrl_ie	GPIO_11 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_11_ctrl_pd	GPIO_11 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_11_ctrl_pu	GPIO_11 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_11_ctrl_sr	GPIO_11 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_11_ctrl_ds1	GPIO_11 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_11_ctrl_ds0	GPIO_11 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_11_ctrl_se	GPIO_11 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_12_ctrl

pad_gpio_12_ctrl 为 GPIO_12 功能管脚控制寄存器。

Offset Address: 0x934　　Total Reset Value: 0x0000_04F0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000

[10]	RW	pad_gpio_12_ctrl_ie	GPIO_12 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_12_ctrl_pd	GPIO_12 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_12_ctrl_pu	GPIO_12 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_12_ctrl_sr	GPIO_12 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	pad_gpio_12_ctrl_ds2	GPIO_12 驱动控制 DS2。	0x1
[5]	RW	pad_gpio_12_ctrl_ds1	GPIO_12 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_12_ctrl_ds0	GPIO_12 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_12_ctrl_se	GPIO_12 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_13_ctrl

pad_gpio_13_ctrl 为 GPIO_13 功能管脚控制寄存器。

Offset Address: 0x938　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_13_ctrl_ie	GPIO_13 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_13_ctrl_pd	GPIO_13 输入下拉使能。 0：禁止； 1：使能。	0x0

[8]	RW	pad_gpio_13_ctrl_pu	GPIO_13 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_13_ctrl_sr	GPIO_13 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_13_ctrl_ds1	GPIO_13 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_13_ctrl_ds0	GPIO_13 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_13_ctrl_se	GPIO_13 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_gpio_14_ctrl

pad_gpio_14_ctrl 为 GPIO_14 功能管脚控制寄存器。

Offset Address: 0x93C　　Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_gpio_14_ctrl_ie	GPIO_14 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_gpio_14_ctrl_pd	GPIO_14 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_gpio_14_ctrl_pu	GPIO_14 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_gpio_14_ctrl_sr	GPIO_14 Slew Rate。 0：快沿； 1：慢沿。	0x1

[6]	RW	reserved	保留。	0x0
[5]	RW	pad_gpio_14_ctrl_ds1	GPIO_14 驱动控制 DS1。	0x1
[4]	RW	pad_gpio_14_ctrl_ds0	GPIO_14 驱动控制 DS0。	0x1
[3]	RW	pad_gpio_14_ctrl_se	GPIO_14 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_sfc_csn_ctrl

pad_sfc_csn_ctrl 为 SFC_CSN 功能管脚控制寄存器。

Offset Address: 0x940　　Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_sfc_csn_ctrl_ie	SFC_CSN 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_sfc_csn_ctrl_pd	SFC_CSN 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_sfc_csn_ctrl_pu	SFC_CSN 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_sfc_csn_ctrl_sr	SFC_CSN Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	pad_sfc_csn_ctrl_ds2	SFC_CSN 驱动控制 DS2。	0x1
[5]	RW	pad_sfc_csn_ctrl_ds1	SFC_CSN 驱动控制 DS1。	0x1
[4]	RW	pad_sfc_csn_ctrl_ds0	SFC_CSN 驱动控制 DS0。	0x0
[3]	RW	pad_sfc_csn_ctrl_se	SFC_CSN 施密特触发器使能控制。	0x0

			0：No Schmitt； 1：Schmitt Enable。	
[2:0]	RW	reserved	保留。	0x0

pad_sfc_io1_ctrl

pad_sfc_io1_ctrl 为 SFC_IO1 功能管脚控制寄存器。

Offset Address: 0x944　　Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_sfc_io1_ctrl_ie	SFC_IO1 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_sfc_io1_ctrl_pd	SFC_IO1 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_sfc_io1_ctrl_pu	SFC_IO1 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_sfc_io1_ctrl_sr	SFC_IO1 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	pad_sfc_io1_ctrl_ds2	SFC_IO1 驱动控制 DS2。	0x1
[5]	RW	pad_sfc_io1_ctrl_ds1	SFC_IO1 驱动控制 DS1。	0x1
[4]	RW	pad_sfc_io1_ctrl_ds0	SFC_IO1 驱动控制 DS0。	0x0
[3]	RW	pad_sfc_io1_ctrl_se	SFC_IO1 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_sfc_io2_ctrl

pad_sfc_io2_ctrl 为 SFC_IO2 功能管脚控制寄存器。

Offset Address: 0x948 Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_sfc_io2_ctrl_ie	SFC_IO2 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_sfc_io2_ctrl_pd	SFC_IO2 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_sfc_io2_ctrl_pu	SFC_IO2 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_sfc_io2_ctrl_sr	SFC_IO2 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	pad_sfc_io2_ctrl_ds2	SFC_IO2 驱动控制 DS2。	0x1
[5]	RW	pad_sfc_io2_ctrl_ds1	SFC_IO2 驱动控制 DS1。	0x1
[4]	RW	pad_sfc_io2_ctrl_ds0	SFC_IO2 驱动控制 DS0。	0x0
[3]	RW	pad_sfc_io2_ctrl_se	SFC_IO2 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_sfc_io0_ctrl

pad_sfc_io0_ctrl 为 SFC_IO0 功能管脚控制寄存器。

Offset Address: 0x94C Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_sfc_io0_ctrl_ie	SFC_IO0 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_sfc_io0_ctrl_pd	SFC_IO0 输入下拉使能。	0x0

			0：禁止； 1：使能。	
[8]	RW	pad_sfc_io0_ctrl_pu	SFC_IO0 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_sfc_io0_ctrl_sr	SFC_IO0 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	pad_sfc_io0_ctrl_ds2	SFC_IO0 驱动控制 DS2。	0x1
[5]	RW	pad_sfc_io0_ctrl_ds1	SFC_IO0 驱动控制 DS1。	0x1
[4]	RW	pad_sfc_io0_ctrl_ds0	SFC_IO0 驱动控制 DS0。	0x0
[3]	RW	pad_sfc_io0_ctrl_se	SFC_IO0 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_sfc_clk_ctrl

pad_sfc_clk_ctrl 为 SFC_CLK 功能管脚控制寄存器。

Offset Address: 0x950　　Total Reset Value: 0x0000_04D0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_sfc_clk_ctrl_ie	SFC_CLK 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_sfc_clk_ctrl_pd	SFC_CLK 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_sfc_clk_ctrl_pu	SFC_CLK 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_sfc_clk_ctrl_sr	SFC_CLK Slew Rate。 0：快沿；	0x1

			1：慢沿。	
[6]	RW	pad_sfc_clk_ctrl_ds2	SFC_CLK 驱动控制 DS2。	0x1
[5]	RW	pad_sfc_clk_ctrl_ds1	SFC_CLK 驱动控制 DS1。	0x0
[4]	RW	pad_sfc_clk_ctrl_ds0	SFC_CLK 驱动控制 DS0。	0x1
[3]	RW	pad_sfc_clk_ctrl_se	SFC_CLK 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0
[2:0]	RW	reserved	保留。	0x0

pad_sfc_io3_ctrl

pad_sfc_io3_ctrl 为 SFC_IO3 功能管脚控制寄存器。

Offset Address: 0x954　　Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	保留。	0x000000
[10]	RW	pad_sfc_io3_ctrl_ie	SFC_IO3 输入信号使能。 0：禁止； 1：使能。	0x1
[9]	RW	pad_sfc_io3_ctrl_pd	SFC_IO3 输入下拉使能。 0：禁止； 1：使能。	0x0
[8]	RW	pad_sfc_io3_ctrl_pu	SFC_IO3 输入上拉使能。 0：禁止； 1：使能。	0x0
[7]	RW	pad_sfc_io3_ctrl_sr	SFC_IO3 Slew Rate。 0：快沿； 1：慢沿。	0x1
[6]	RW	pad_sfc_io3_ctrl_ds2	SFC_IO3 驱动控制 DS2。	0x1
[5]	RW	pad_sfc_io3_ctrl_ds1	SFC_IO3 驱动控制 DS1。	0x1
[4]	RW	pad_sfc_io3_ctrl_ds0	SFC_IO3 驱动控制 DS0。	0x0
[3]	RW	pad_sfc_io3_ctrl_se	SFC_IO3 施密特触发器使能控制。 0：No Schmitt； 1：Schmitt Enable。	0x0

[2:0]	RW	reserved	保留。	0x0

6.2 GPIO

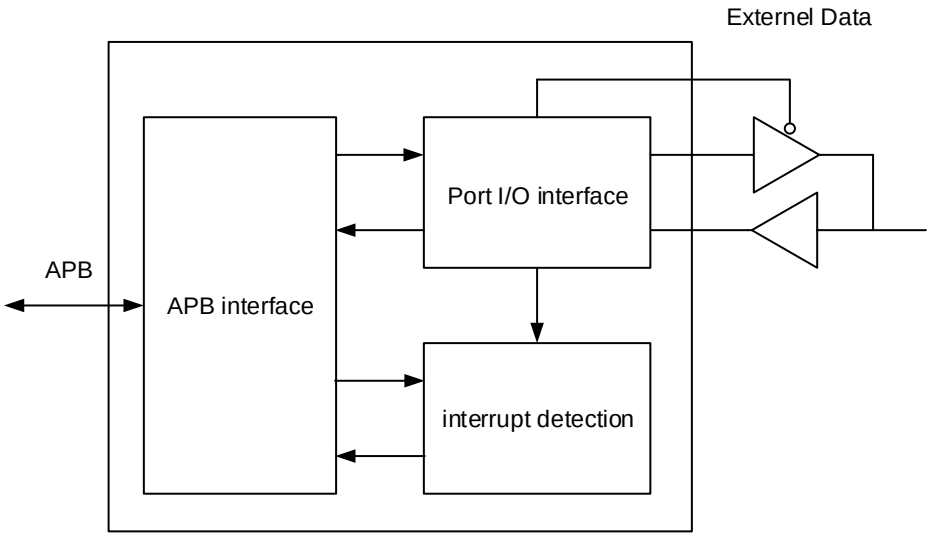
6.2.1 概述

GPIO 是可编程的通用输入/输出接口，用于生成和采集特定应用的输入或输出信号，实现系统和外设之间的通信，方便系统对外设的控制。Hi3861、Hi3861L、Hi3881 芯片 GPIO 符合 AMBA2.0 的 APB 协议。

如图 6-1 所示，GPIO 模块主要接口如下：

- APB 接口
- I/O pad 的外部数据接口
- 中断信号接口

图6-1 GPIO 模块原理图



6.2.2 功能描述

GPIO 接口具有以下功能特点：

- 时钟源可选择：工作模式晶体时钟 24M/40M、低功耗模式 32K 时钟。
- 1 组 GPIO，共 15 个独立的可配置管脚。
- 每个 GPIO 管脚都可单独控制传输方向。
- 每个 GPIO 可以单独被配置为外部中断源。
- GPIO 用作中断时有 4 种中断触发方式，中断时触发方式可配：

- 上升沿触发
- 下降沿触发
- 高电平触发
- 低电平触发

- GPIO 上报一个中断，CPU 查询上报的 GPIO 编号。
- 每个中断支持独立屏蔽的功能，脉冲中断支持可清除功能。

6.2.3 工作方式

GPIO 可以配置为输入或输出方式，通过 GPIO_SWPORT_DDR 配置。其中：

- 当配置为输入方式时，不但可以通过管脚输入到输入端口寄存器 GPIO_EXT_PORT，还可以作为外部中断源，以及外部的睡眠唤醒信号。
- 当配置为输出方式时，配置 GPIO_SWPORT_DR 可以将配置数据输入到管脚上。

中断模式可选择电平触发或边沿触发，通过 GPIO_INTTYPE_LEVEL 配置。

电平触发可选择高电平触发或低电平触发，通过 GPIO_INT_PLOARITY 配置。

单沿触发可选择上升沿触发或下降沿触发，通过 GPIO_INT_PLOARITY 配置。

每个中断支持独立的使能，通过 GPIO_INTEN 配置。

每个中断支持独立的屏蔽，通过 GPIO_INTMASK 配置。

每个中断的状态支持可查询：

- 屏蔽前的原始中断状态，通过 GPIO_RAWINTSTATUS 查询。
- 屏蔽后的最终中断状态，通过 GPIO_INTSTATUS 查询。

脉冲中断支持可清除，通过 GPIO_PORT_EOI 配置。

须知

为避免多余中断、多余唤醒信号的产生，使用 GPIO 作为中断源时应该按位操作，只配置需要作为中断的 GPIO 管脚。

6.2.4 寄存器概览

GPIO 寄存器概览如表 6-9 所示。

表6-9 GPIO 寄存器概览（基址是 0x5000_6000）

偏移地址	名称	描述	页码
0x00	GPIO_SWPORT_DR	端口的数据输出寄存器	6-41
0x04	GPIO_SWPORT_DDR	端口的数据传输方向寄存器	6-41
0x30	GPIO_INTEN	端口的中断使能寄存器	6-42

偏移地址	名称	描述	页码
0x34	GPIO_INTMASK	端口的中断屏蔽寄存器	6-42
0x38	GPIO_INTTYPE_LEVEL	端口的中断类型寄存器	6-42
0x3C	GPIO_INT_PLOARITY	端口的中断极性寄存器	6-43
0x40	GPIO_INTSTATUS	端口的中断状态寄存器	6-43
0x44	GPIO_RAWINTSTATUS	端口的原始中断状态寄存器	6-43
0x4C	GPIO_PORT_EOI	端口的清除中断寄存器	6-44
0x50	GPIO_EXT_PORT	端口的数据输入接口寄存器	6-44

6.2.5 寄存器描述

GPIO_SWPORT_DR

GPIO_SWPORT_DR 为端口的数据输出寄存器。

Offset Address: 0x00　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_swport_dr	输出数据。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15。	0x0000

GPIO_SWPORT_DDR

GPIO_SWPORT_DDR 为端口的数据传输方向寄存器。

Offset Address: 0x04　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_swport_ddr	控制数据传输的方向，决定每 bit 是输入还是输出。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：输入(默认值)； 1：输出。	0x0000

GPIO_INTEN

GPIO_INTEN 为端口的中断使能寄存器。

Offset Address: 0x30　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_inten	中断使能，配置端口信号的每 bit 是中断模式还是正常模式。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：正常模式(默认值)； 1：中断模式。	0x0000

GPIO_INTMASK

GPIO_INTMASK 为端口的中断屏蔽寄存器。

Offset Address: 0x34　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_intmask	中断屏蔽，控制端口的对应 bit 是否被中断屏蔽。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：不屏蔽(默认值)； 1：屏蔽。	0x0000

GPIO_INTTYPE_LEVEL

GPIO_INTTYPE_LEVEL 为端口的中断类型寄存器。

Offset Address: 0x38　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_inttype_level	中断触发类型，控制中断的类型。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：电平触发(默认值)； 1：边沿触发。	0x0000

GPIO_INT_PLOARITY

GPIO_INT_PLOARITY 为端口的中断极性寄存器。

Offset Address: 0x3C　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_int_ploarity	中断极性。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：下降沿或低电平触发(默认值)； 1：上升沿或高电平触发。	0x0000

GPIO_INTSTATUS

GPIO_INTSTATUS 为端口的中断状态寄存器。GPIO 配置为中断方式时，该寄存器用于指示 GPIO 输入信号的中断状态。

Offset Address: 0x40　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RO	gpio_intstatus	中断状态，指示对应 bit 中断是否产生。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：无中断； 1：有中断。	0x0000

GPIO_RAWINTSTATUS

GPIO_RAWINTSTATUS 为端口的原始中断状态寄存器。GPIO 配置为中断方式时，该寄存器用于指示 GPIO 输入信号的原始中断状态。

Offset Address: 0x44　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RO	gpio_rawintstatus	原始中断状态，指示屏蔽中断前对应 bit 中断是否产生。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：无中断； 1：有中断。	0x0000

GPIO_PORT_EOI

GPIO_PORT_EOI 为端口的清除中断寄存器。该寄存器用于清除 GPIO 输入中断。

Offset Address: 0x4C　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	WO	gpio_port_eoi	中断清除，控制清除端口的中断。不自动清零。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15，每 bit 含义如下： 0：不清除； 1：清除。	0x0000

GPIO_EXT_PORT

GPIO_EXT_PORT 为端口的数据输入接口寄存器。

Offset Address: 0x50　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RO	gpio_ext_port	当端口配置为输入时，存放端口的输入数据。 bit[0]～bit[15]分别对应 GPIO_0～GPIO_15。	0x0000

6.3 UART

6.3.1 概述

通用异步收发器 UART 是一个异步串行的通信接口，UART 主要用于和外部芯片的 UART 进行对接，实现两芯片间的通信。

芯片提供 3 个 UART 单元，支持 2 线模式。

6.3.2 功能描述

UART 具有以下功能特点：

- 支持 64×8bit 的发送 FIFO 和 64×12bit 的接收 FIFO（First In First Out）。
- 支持数据位和停止位的位宽可编程：
  - 数据位可通过编程设定为 5/6/7/8 bit。
  - 停止位可通过编程设定为 1/2 bit。

- 支持奇、偶校验方式或无校验。
- 支持传输速率可编程、支持整数小数分频。
- 支持接收 FIFO 中断、发送 FIFO 中断、接收超时中断、错误中断。
- 支持初始中断状态查询和屏蔽后中断状态查询。
- 支持通过编程禁止 UART 模块或 UART 发送/接收功能以降低功耗。
- UART0 不支持硬件流控；UART1、UART2 支持硬件流控。

6.3.3 工作方式

接口信号

UART 接口信号描述如表 6-10 所示。

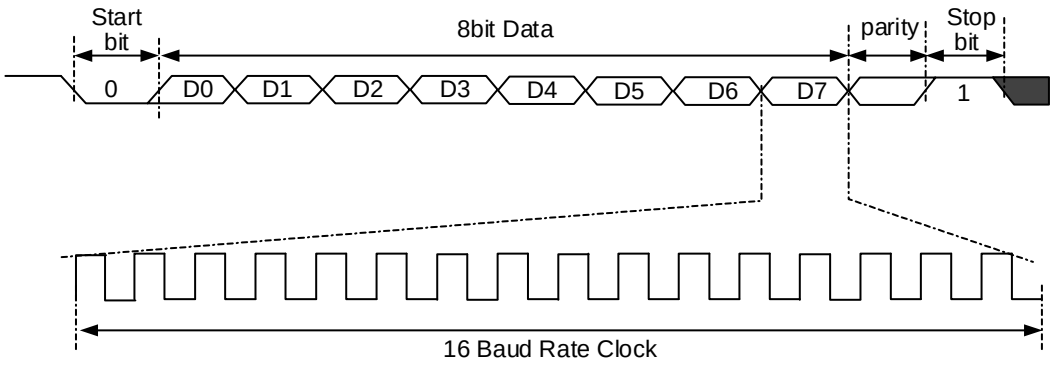
表6-10 UART 接口信号描述

信号名	宽度（bit）	方向	功能描述
RXD	1	I	输入数据。
TXD	1	O	输出数据。
CTS	1	I	清除发送信号，用于硬件流控，低有效。
RTS	1	O	请求发送信号，用于硬件流控，低有效。

UART 数据格式

UART 的数据帧格式如图 6-2 所示。其中数据帧长度、停止位位数和奇偶检验可配置。

图6-2 UART 数据帧格式



UART 初始化配置

UART 初始化配置流程如下：

1. 读 UART_FR[busy]，等待 UART 状态为空闲，对 UART 模块进行软复位。
2. 写 UART_CR[uarten]为 0，禁止 UART。
3. 写 UART_LCR_H 为 0，清空 UART 的配置。
4. 根据波特率配置 UART_IBRD 和 UART_FBRD，设置波特率分频值。
5. 根据需求配置 UART_LCR_H，设置奇偶校验位、数据长度、FIFO 使能、停止位个数。
6. 写 UART_IFLS，设置发送 FIFO 水线、接收 FIFO 水线、流控水线。
7. 如果需要进行 DMA 搬移，写 UART_DMACR[txdmae]或[rxdmae]为 1，使能 UART 的 DMA。
8. 如果需要使能 UART 的硬件流控功能，写 UART_CR[cts_en]、[rts_en]为 1，启动硬件流控功能。
9. 写 UART_CR[rxe]为 1，配置 UART 接收使能；或写 UART_CR[txe]为 1，配置 UART 发送使能；写 UART_CR[uarten]为 1，配置 UART 使能。

----结束

在初始化寄存器之后便可以开始数据的传输。

须知

- 整数波特率寄存器和小数波特率寄存器的值必须等到当前数据发送和接收完毕才更新。
- 最小的分频值为 1，最大的分频值为 65535（0xFFFF）。即 UART_IBRD=0 无效，此时 UART_FBRD 将被忽略；UART_IBRD=65535（0xFFFF），此时 UART_FBRD 必须为 0，否则会导致发送和接收失败。

说明

波特率分频值计算方法：

- 计算理论分频系数 n，UART 的波特率=内部总线频率/(16×分频系数)。
- 分频值的整数部分 UART_IBRD 的配置值为 integer(n)。
- 分频值的小数部分 UART_FBRD 的值为 integer((n-integer (n))×64+0.5)，integer()为向下取整函数。

6.3.4 寄存器概览

UART 寄存器概览如表 6-11 所示。

表6-11 UART 寄存器概览（UART0 基址是 0x4000_8000、UART1 基址是 0x4000_9000、UATR2 基址是 0x4000_A000）

偏移地址	名称	描述	页码
0x000	UART_DR	UART 数据寄存器	6-47

偏移地址	名称	描述	页码
0x004	UART_RSR	接收状态/错误清除寄存器	6-48
0x018	UART_FR	UART 标志寄存器	6-49
0x024	UART_IBRD	整数波特率寄存器	6-50
0x028	UART_FBRD	小数波特率寄存器	6-50
0x02C	UART_LCR_H	传输模式控制寄存器	6-51
0x030	UART_CR	UART 控制寄存器	6-52
0x034	UART_IFLS	中断 FIFO 阈值选择寄存器	6-53
0x038	UART_IMSC	中断屏蔽寄存器	6-54
0x03C	UART_RIS	原始中断状态寄存器	6-55
0x040	UART_MIS	屏蔽后中断状态寄存器	6-56
0x044	UART_ICR	中断清除寄存器	6-57
0x048	UART_DMACR	DMA 控制寄存器	6-58

6.3.5 寄存器描述

UART_DR

UART_DR 为 UART 数据寄存器。存放接收数据和发送数据，同时可以从该寄存器中读出接收状态。

Offset Address: 0x000　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:12]	-	reserved	保留。	0x0
[11]	RO	oe	溢出错误状态位。 0：无溢出错误； 1：有溢出错误(即接收 FIFO 满且接收了一个数据)。	0x0
[10]	RO	be	Break 错误状态位。 0：无 Break 错误； 1：有 Break 错误。 Break 的条件：接收数据的输入保持低电平的时间比一个全字传输(包括：start、data、parity、stop bit)还要长。	0x0

[9]	RO	pe	校验错误状态位。 0：无校验错误； 1：有校验错误。	0x0
[8]	RO	fe	帧错误状态位。 0：无帧错误； 1：有帧错误。	0x0
[7:0]	RW	data	接收数据和发送数据。	0x00

UART_RSR

须知

对 UART_RSR 的任何写操作都会对 UART_RSR 进行复位。

UART_RSR 为接收状态/错误清除寄存器。读时作为接收状态寄存器；写时作为错误清除寄存器。

接收状态也可以从 UART_DR 中读出。从 UART_DR 中读出的 break、frame、parity 的状态信息比从 UART_RSR 读出的信息优先级高（即 UART_DR 中的状态变化比 UART_RSR 更快）。

Offset Address: 0x004　　Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:4]	-	reserved	保留。	0x0
[3]	RW	oe	溢出错误状态和清除位。 0：无溢出错误； 1：溢出错误。 当 FIFO 满时，FIFO 中的内容保持有效(因为不会有下一个数据写到 FIFO 中，只是移位寄存器会溢出)。CPU 必须立刻读数据以腾空 FIFO。	0x0
[2]	RW	be	Break 错误状态和清除位。 0：无 Break 错误； 1：有 Break 错误。 Break 的条件：接收数据的输入保持低电平的时间比一个全字传输(包括：start、data、parity、stop bit)还要长。	0x0
[1]	RW	pe	校验错误状态和清除位。	0x0

			0：无校验错误； 1：接收数据的校验错误。	
[0]	RW	fe	帧错误状态和清除位。 0：无帧错误； 1：接收到的数据的停止位错误(有效的停止位为高电平)。	0x0

UART_FR

UART_FR 为 UART 标志寄存器。

Offset Address: 0x018　　Total Reset Value: 0x0197

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x01
[7]	RO	txfe	发送 FIFO 空状态位。 当 UART_LCR_H[fen]为 0 时： 0：发送 Holding Register 非空； 1：发送 Holding Register 空。 当 UART_LCR_H[fen]为 1 时： 0：发送 FIFO 为非空； 1：发送 FIFO 为空。	0x1
[6]	RO	rxff	接收 FIFO 满状态位。 当 UART_LCR_H[fen]为 0 时： 0：接收 Holding Register 非满； 1：接收 Holding Register 满。 当 UART_LCR_H[fen]为 1 时： 0：接收 FIFO 为非满； 1：接收 FIFO 为满。	0x0
[5]	RO	txff	发送 FIFO 满状态位。 当 UART_LCR_H[fen]为 0 时： 0：发送 Holding Register 非满； 1：发送 Holding Register 满。 当 UART_LCR_H[fen]为 1 时： 0：发送 FIFO 为非满； 1：发送 FIFO 为满。	0x0
[4]	RO	rxfe	接收 FIFO 空状态位。	0x1

			当 UART_LCR_H[fen]为 0 时： 0：接收 Holding Register 非空； 1：接收 Holding Register 空。 当 UART_LCR_H[fen]为 1 时： 0：接收 FIFO 为非空； 1：接收 FIFO 为空。	
[3]	RO	busy	UART 忙闲状态位。 0：空闲或完成发送数据； 1：正忙于发送数据。 该 bit 一旦置位，该状态一直保持到整个字节(包括所有的停止位)完全从移位寄存器中发送出去。发送 FIFO 非空，该 bit 置 1。	0x0
[2:0]	-	reserved	保留。	0x7

UART_IBRD

UART_IBRD 为整数波特率寄存器。

Offset Address: 0x024　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	bauddivint	整数波特率分频值。 **注意：复位时全部清零。**	0x0000

UART_FBRD

UART_FBRD 为小数波特率寄存器。

UART_IBRD 和 UART_FBRD 的值必须等到当前数据发送和接收完成才更新。

最小的分频值为 1，最大的分频值为 65535（2^16－1）。UART_IBRD=0 是无效的，此时 UART_FBRD 将被忽略；如果 UART_IBRD＝65535（0xFFFF），则 UART_FBRD 必须为 0，否则会导致发送和接收的失败。波特率分频值计算方法：

计算理论分频系数 n，UART 的波特率=内部总线频率/(16×分频系数)；

分频值的整数部分 UART_IBRD 的配置值为 integer(n)；

分频值的小数部分 UART_FBRD 的值为 integer((n-integer(n))×64+0.5)，integer()为向下取整函数。

Offset Address: 0x028　　Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:6]	-	reserved	保留。	0x0
[5:0]	RW	banddivfrac	小数波特率分频值。 **注意：复位时全部清零。**	0x00

UART_LCR_H

UART_LCR_H 为传输模式控制寄存器。

说明

如果更新 UART_IBRD 和 UART_FBRD 的内容，必须同时更新 UART_LCR_H。

Offset Address: 0x02C　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x00
[7]	RW	sps	校验位设置位。 0：Stick Parity 禁止； 1：当[pen]、[eps]被置 1 时，奇偶校验位发送和校验按 0 校验；当[pen]被置 1 且[eps]为 0 时，奇偶校验位发送和校验按 1 校验。	0x0
[6:5]	RW	wlen	指示发送和接收一帧数据的数据位数。 00：5bit； 01：6bit； 10：7bit； 11：8bit。	0x0
[4]	RW	fen	发送和接收 FIFO 使能控制。 0：禁止； 1：使能。	0x0
[3]	RW	stp2	发送帧尾 2bit 停止位判断。 0：无 2bit 停止位； 1：有 2bit 停止位。 **注意：接收逻辑在接收时不检查 2bit 的停止位。**	0x0
[2]	RW	eps	发送和接收过程中的奇偶校验选择。 0：生成或检查奇校验； 1：生成或检查偶校验。	0x0

			注意：当[pen]为 0 时，该位不起作用。	
[1]	RW	pen	校验选择位。 0：不作校验； 1：发送方向产生校验，接收方向进行校验检查。	0x0
[0]	RW	brk	发送 Break 选择位。 0：无效； 1：在完成当前数据的发送后，UTXD 连续输出低电平。 **注意：须正确执行 Break 命令，软件将该 bit 置 1 的时间必须超过 2 个完整帧；在正常使用中，该 bit 必须清零。**	0x0

UART_CR

UART_CR 为 UART 控制寄存器。

Offset Address: 0x030　　Total Reset Value: 0x0300

Bits	Access	Name	Description	Reset
[15]	RW	ctsen	CTS 硬件流控使能。 0：禁止； 1：使能(仅当 nUARTCTS 信号有效时才发送数据)。	0x0
[14]	RW	rtsen	RTS 硬件流控使能。 0：禁止； 1：使能(仅当接收 FIFO 有空间时才请求接收数据)。	0x0
[13:12]	-	reserved	保留。	0x0
[11]	RW	rts	请求发送设置位。 该 bit 为 UART Modem 状态输出信号 nUARTRTS 的取反。 0：输出信号不变； 1：输出信号为 0。	0x0
[10]	RW	dtr	数据发送准备设置位。 该 bit 为 UART Modem 状态输出信号 nUARTDTR 的取反。 0：输出信号不变；	0x0

			1：输出信号为 0。	
[9]	RW	rxe	UART 接收使能。 0：禁止； 1：使能。 **注意：在接收的过程中如果 UART 被禁止，则当前数据的接收会在正常停止之前结束。**	0x1
[8]	RW	txe	UART 发送使能。 0：禁止； 1：使能。 **注意：在发送的过程中如果 UART 被禁止，则当前数据的发送会在正常停止之前结束。**	0x1
[7]	RW	lbe	环回使能。 0：禁止； 1：UARTTXD 输出环回到 UARTRXD。	0x0
[6:1]	-	reserved	保留。	0x00
[0]	RW	uarten	UART 使能。 0：禁止； 1：使能。 **注意：如果在发送和接收过程中将 UART 禁止，则会在正常停止之前结束当前数据的传送。**	0x0

UART_IFLS

UART_IFLS 为中断 FIFO 阈值选择寄存器。用于设置 FIFO 的中断（UART_TXINTR 或 UART_RXINTR）触发线。

Offset Address: 0x034　　Total Reset Value: 0x0092

Bits	Access	Name	Description	Reset
[15:9]	-	reserved	保留。	0x00
[8:6]	RW	rtsflsel	硬件流控 rts_n 的触发条件选择(触发点如下)。 000：接收 FIFO≥1/8full； 001：接收 FIFO≥1/4full； 010：接收 FIFO≥1/2full；	0x2

			011：接收 FIFO≥3/4full； 100：接收 FIFO≥7/8full； 101～111：保留。	
[5:3]	RW	rxiflsel	接收中断 FIFO 的阈值选择(接收中断的触发点如下)。 000：接收 FIFO≥1/8full； 001：接收 FIFO≥1/4full； 010：接收 FIFO≥1/2full； 011：接收 FIFO≥3/4full； 100：接收 FIFO≥7/8full； 101～111：保留。	0x2
[2:0]	RW	txiflsel	发送中断 FIFO 的阈值选择(发送中断的触发点如下)。 000：发送 FIFO≤1/8full； 001：发送 FIFO≤1/4full； 011：发送 FIFO≤3/4full； 010：发送 FIFO≤1/2full； 100：发送 FIFO≤7/8full； 101～111：保留。	0x2

UART_IMSC

UART_IMSC 为中断屏蔽寄存器。

Offset Address: 0x038　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	保留。	0x00
[10]	RW	oeim	溢出错误中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0
[9]	RW	beim	Break 错误中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0
[8]	RW	peim	校验中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0

[7]	RW	feim	帧错误中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0
[6]	RW	rtim	接收超时中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0
[5]	RW	txim	发送中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0
[4]	RW	rxim	接收中断的屏蔽状态。 0：屏蔽； 1：不屏蔽。	0x0
[3:0]	-	reserved	保留。	0x0

UART_RIS

UART_RIS 为原始中断状态寄存器。

说明

其内容不受中断屏蔽寄存器的影响。

Offset Address: 0x03C　　Total Reset Value: 0x000F

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	保留。	0x00
[10]	RO	oeris	原始的溢出错误中断状态。 0：未产生中断； 1：已产生中断。	0x0
[9]	RO	beris	原始的 Break 错误中断状态。 0：未产生中断； 1：已产生中断。	0x0
[8]	RO	peris	原始的校验中断状态。 0：未产生中断； 1：已产生中断。	0x0
[7]	RO	feris	原始的错误中断状态。 0：未产生中断； 1：已产生中断。	0x0

[6]	RO	rtris	原始的接收超时中断状态。 0：未产生中断； 1：已产生中断。	0x0
[5]	RO	txris	原始的发送中断状态。 0：未产生中断； 1：已产生中断。	0x0
[4]	RO	rxris	原始的接收中断状态。 0：未产生中断； 1：已产生中断。	0x0
[3:0]	-	reserved	保留。	0xF

UART_MIS

UART_MIS 为屏蔽后中断状态寄存器。

说明

其内容为原始中断状态和中断屏蔽进行“与”操作后的结果。

Offset Address: 0x040　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	保留。	0x00
[10]	RO	oemis	屏蔽后的溢出错误中断状态。 0：未产生中断； 1：已产生中断。	0x0
[9]	RO	bemis	屏蔽后的 Break 错误中断状态。 0：未产生中断； 1：已产生中断。	0x0
[8]	RO	pemis	屏蔽后的校验中断状态。 0：未产生中断； 1：已产生中断。	0x0
[7]	RO	femis	屏蔽后的错误中断状态。 0：未产生中断； 1：已产生中断。	0x0
[6]	RO	rtmis	屏蔽后的接收超时中断状态。 0：未产生中断； 1：已产生中断。	0x0

[5]	RO	txmis	屏蔽后的发送中断状态。 0：未产生中断； 1：已产生中断。	0x0
[4]	RO	rxmis	屏蔽后的接收中断状态。 0：未产生中断； 1：已产生中断。	0x0
[3:0]	-	reserved	保留。	0x0

UART_ICR

UART_ICR 为中断清除寄存器。

说明

写 1 时，相应的中断被清除；写 0 时，则不起作用。

Offset Address: 0x044　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	保留。	0x00
[10]	WO	oeic	清除溢出错误中断。 0：无效； 1：清除中断。	0x0
[9]	WO	beic	清除 Break 错误中断。 0：无效； 1：清除中断。	0x0
[8]	WO	peic	清除校验中断。 0：无效； 1：清除中断。	0x0
[7]	WO	feic	清除错误中断。 0：无效； 1：清除中断。	0x0
[6]	WO	rtic	清除接收超时中断。 0：无效； 1：清除中断。	0x0
[5]	WO	txic	清除发送中断。 0：无效； 1：清除中断。	0x0

[4]	WO	rxic	清除接收中断。 0：无效； 1：清除中断。	0x0
[3:0]	-	reserved	保留。	0x0

UART_DMACR

UART_DMACR 为 DMA 控制寄存器。用于配置发送 FIFO 和接收 FIFO 的 DMA 使能。

Offset Address: 0x048　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:3]	-	reserved	保留。	0x0000
[2]	RW	dmaonerr	UART 错误中断(UARTEINTR)有效时，接收通道 DMA 使能控制。 0：请求输出(UARTRXDMASREQ 或 UARRTXDMABREQ)有效； 1：请求输出(UARTRXDMASREQ 或 UARRTXDMABREQ)无效。	0x0
[1]	RW	txdmae	发送 FIFO 的 DMA 使能控制。 0：禁止； 1：使能。	0x0
[0]	RW	rxdmae	接收 FIFO 的 DMA 使能控制。 0：禁止； 1：使能。	0x0

6.4 I2C

6.4.1 概述

I2C 模块是 APB 总线上的从设备，是 I2C 总线上的主设备。I2C 模块的作用是完成 CPU 对 I2C 总线上从设备的数据读写，CPU 可以连续配置多个发送的数据和接收多个数据。I2C 总线上可挂载多个从设备。

6.4.2 功能描述

I2C 具有以下功能特点：

- 2.0 版本的 I2C 总线协议，只支持 Master 模式。
- I2C 模块在 APB 总线上执行 APB Slave 的功能，在 I2C 总线上作为 Master，支持多主设备时的总线仲裁。
- I2C 主机可以向从机写入数据，也可以接收从机发来的数据。
- 支持 Clock synchronization 和 Bit and Byte waiting。
- 支持中断或轮询操作。
- I2C 模块支持标准地址（7bit）和扩展地址（10bit）。
- 可以工作在两种速度模式下：标准模式（100Kbit/s）、快速模式（400Kbit/s）。
- I2C 模块支持 General Call 和 Start Byte 功能。
- I2C 总线上不支持 CBUS 器件。
- 对接收到的 SDA（Serial Data and Address）和 SCL（Serial Clock Line）信号进行滤波。
- 内部包含 1 个 32×8bit 的发送 FIFO 和 1 个 32×8bit 的接收 FIFO。
- 支持硬件检测 FIFO 数据深度并发出相应中断。
- 兼容不使用 FIFO 和使用 FIFO 两种工作方式。

6.4.3 工作方式

I2C 包含以下两种工作场景：

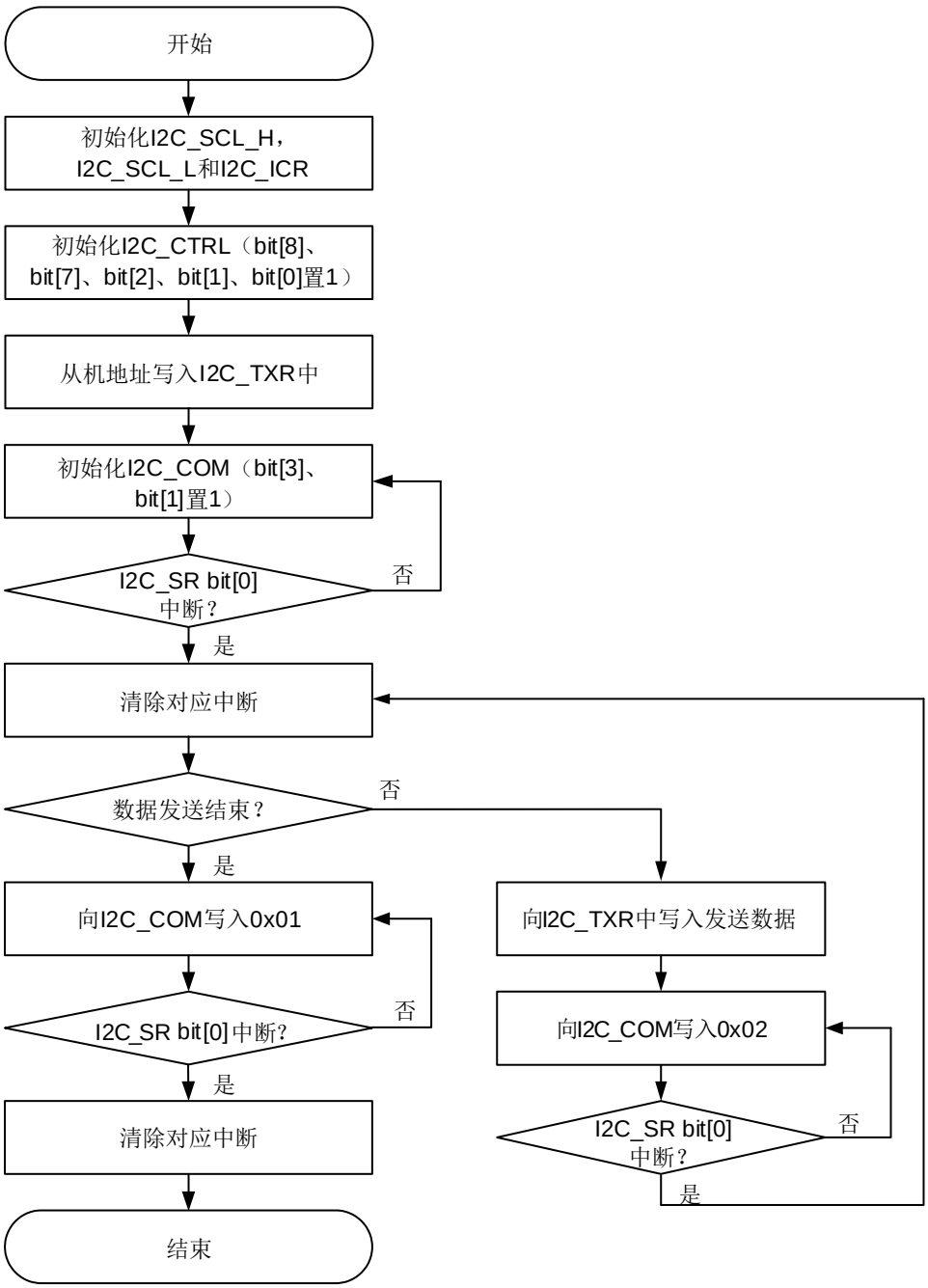
- 主机仅对单个数据发送和接收（不使用 FIFO）。
- 主机连续发送多个数据、连续接收多个数据（使用 FIFO）。

6.4.3.1 不使用 FIFO

I2C 主机发送数据流程

I2C 主机发送数据流程如图 6-3 所示。

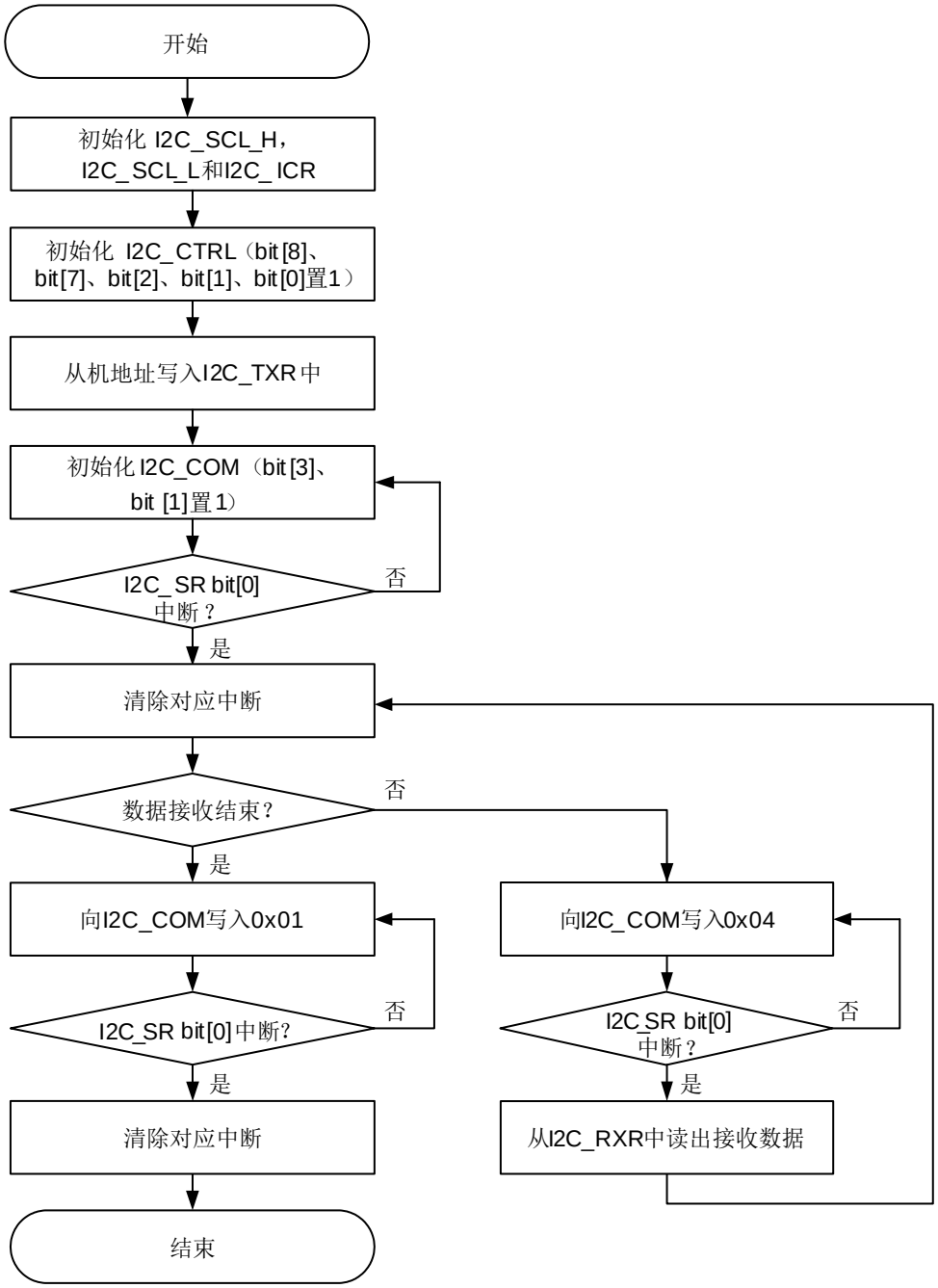
图6-3 I2C 主机发送数据（不使用 FIFO）流程图



I2C 主机接收数据流程

I2C 主机接收数据流程如图 6-4 所示。

图6-4 I2C 主机接收数据（不使用 FIFO）流程图

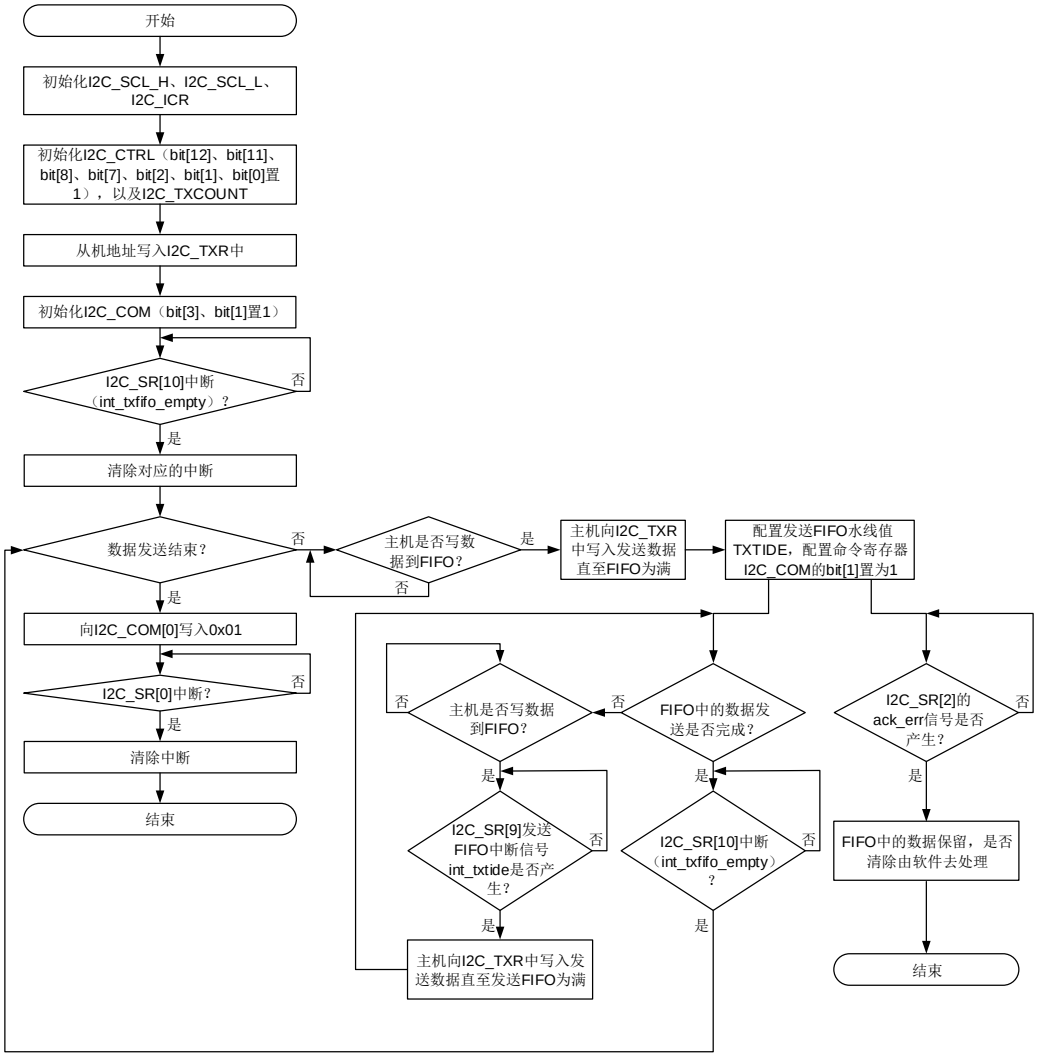


6.4.3.2 使用 FIFO

I2C 主机发送数据流程

I2C 主机发送数据流程图 6-5 所示。

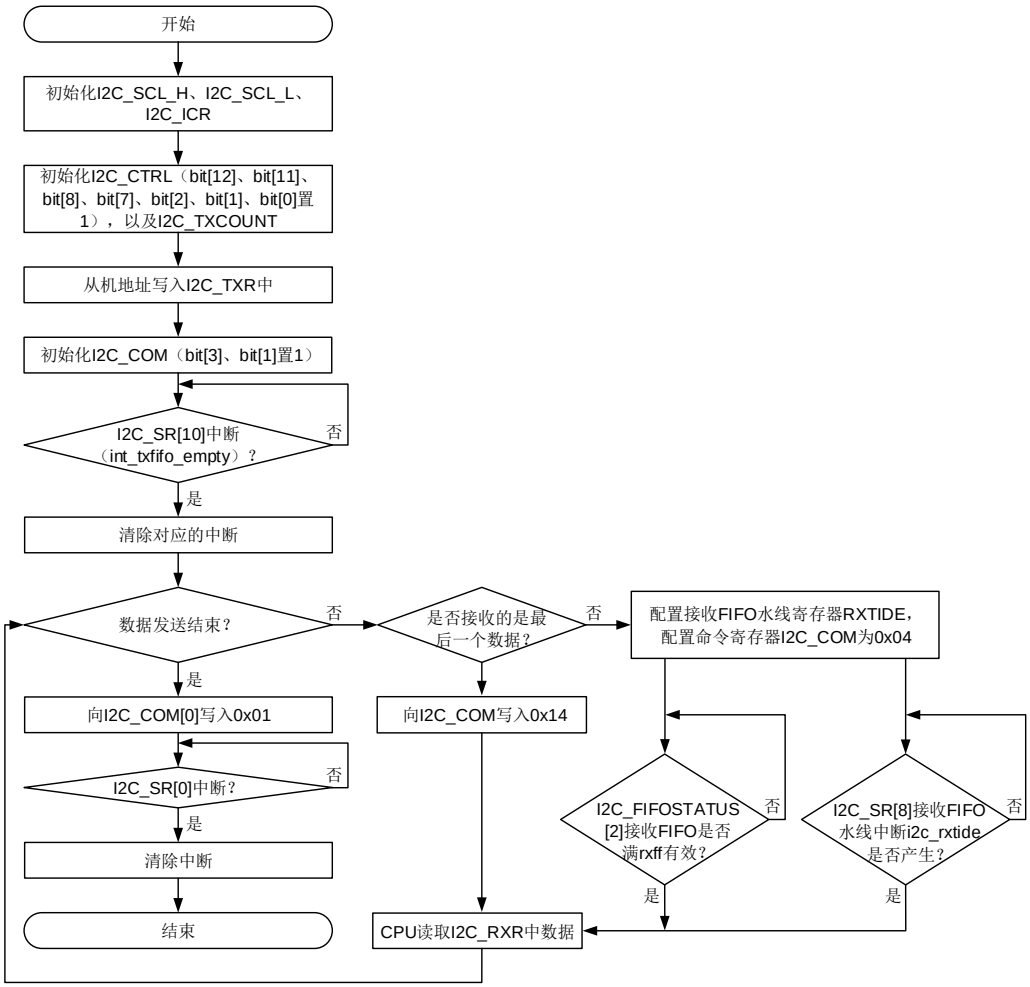
图6-5 I2C 主机发送数据（使用 FIFO）流程图



I2C 主机接收数据流程

I2C 主机接收数据流程如图 6-6 所示。

图6-6 I2C 主机接收数据（使用 FIFO）流程图



6.4.4 寄存器概览

I2C 寄存器概览如表 6-12 所示。

表6-12 I2C 寄存器概览（I2C0 基址是 0x4001_8000、I2C1 基址是 0x4001_9000）

偏移地址	名称	描述	页码
0x00	I2C_CTRL	I2C 控制寄存器	6-64
0x04	I2C_COM	I2C 模块的命令寄存器	6-65
0x08	I2C_ICR	I2C 模块的中断清除寄存器	6-66
0x0C	I2C_SR	I2C 模块状态寄存器	6-67
0x10	I2C_SCL_H	I2C 总线 SCL 信号高电平周期数寄存器	6-69
0x14	I2C_SCL_L	I2C 总线 SCL 信号低电平周期数寄存器	6-69

偏移地址	名称	描述	页码
0x18	I2C_TXR	I2C 发送数据寄存器	6-69
0x1C	I2C_RXR	I2C 接收数据寄存器	6-70
0x20	I2C_FIFOSTATUS	FIFO 状态寄存器	6-70
0x24	I2C_TXCOUNT	发送 FIFO 数据个数寄存器	6-71
0x28	I2C_RXCOUNT	接收 FIFO 数据个数寄存器	6-71
0x2C	I2C_RXTIDE	接收 FIFO 的溢出阈值寄存器	6-71
0x30	I2C_TXTIDE	发送 FIFO 的溢出阈值寄存器	6-72

6.4.5 寄存器描述

I2C_CTRL

I2C_CTRL 为 I2C 控制寄存器。用于配置 I2C 使能和中断屏蔽。

Offset Address: 0x00　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	int_txfifo_over_mask	发送 FIFO 数据发送完成中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[11]	RW	mode_ctrl	I2C 工作模式选择。 0：不使用 FIFO 传输模式； 1：使用 FIFO 传输模式；	0x0
[10]	RW	int_txtide_mask	发送 FIFO 溢出中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[9]	RW	int_rxtide_mask	接收 FIFO 溢出中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[8]	RW	i2c_en	I2C 使能。 0：不使能； 1：使能。	0x0
[7]	RW	int_mask	I2C 中断总屏蔽。	0x0

			0：屏蔽； 1：不屏蔽。	
[6]	RW	int_start_mask	主机开始条件发送结束中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[5]	RW	int_stop_mask	主机停止条件发送结束中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[4]	RW	int_tx_mask	主机发送中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[3]	RW	int_rx_mask	主机接收中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[2]	RW	int_ack_err_mask	从机 ACK 错误中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[1]	RW	int_arb_loss_mask	总线仲裁失败中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[0]	RW	int_done_mask	总线传输完成中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0

I2C_COM

须知

在系统初始化时配置或配置前，需要清除对应中断标志。I2C_COM bit[3:0]在操作结后将自动清 0。

I2C_COM 为 I2C 模块的命令寄存器。用于配置 I2C 模块工作时命令。

Offset Address: 0x04　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	op_ack	主机作为接收器是否发送 ACK。 0：发送； 1：不发送。	0x0
[3]	RW	op_start	产生开始条件操作。 0：操作结束； 1：操作有效。	0x0
[2]	RW	op_rd	产生读操作。 0：操作结束； 1：操作有效。	0x0
[1]	RW	op_we	产生写操作。 0：操作结束； 1：操作有效。	0x0
[0]	RW	op_stop	产生停止条件操作。 0：操作结束； 1：操作有效。	0x0

I2C_ICR

须知

新中断到来时，I2C 模块会自动将 I2C_ICR 相应位清 0。

I2C_ICR 为 I2C 模块的中断清除寄存器。

Offset Address: 0x08　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9]	WC	clr_int_txfifo_over	发送 FIFO 数据发送完成中断标志清除。 0：不清除； 1：清除。	0x0
[8]	WC	clr_int_txtide	发送 FIFO 溢出中断标志清除。 0：不清除； 1：清除。	0x0

[7]	WC	clr_int_rxtide	接收 FIFO 溢出中断标志清除。 0：不清除； 1：清除。	0x0
[6]	WC	clr_int_start	主机开始条件发送结束中断标志清除。 0：不清除； 1：清除。	0x0
[5]	WC	clr_int_stop	主机停止条件发送结束中断标志清除。 0：不清除； 1：清除。	0x0
[4]	WC	clr_int_tx	主机发送中断标志清除。 0：不清除； 1：清除。	0x0
[3]	WC	clr_int_rx	主机接收中断标志清除。 0：不清除； 1：清除。	0x0
[2]	WC	clr_int_ack_err	从机 ACK 错误中断标志清除。 0：不清除； 1：清除。	0x0
[1]	WC	clr_int_arb_loss	总线仲裁失败中断标志清除。 0：不清除； 1：清除。	0x0
[0]	WC	clr_int_done	总线传输完成中断标志清除。 0：不清除； 1：清除。	0x0

I2C_SR

须知

I2C_SR bit[1]表示 I2C 总线仲裁失败。当 I2C_SR bit[1]有效时，当前操作失败。在清 I2C_SR bit[1]之前，需要清除其他中断标志，然后清除 I2C_COM 或向 I2C_COM 写入新的操作命令，最后清除 I2C_SR bit[1]。

I2C_SR 为 I2C 模块状态寄存器。用于读取 I2C 模块工作状态。

Offset Address: 0x0C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RO	int_txfifo_over	发送 FIFO 数据发送完成中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[9]	RO	int_txtide	发送 FIFO 溢出中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[8]	RO	int_rxtide	接收 FIFO 溢出中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[7]	RO	bus_busy	总线忙。 0：空闲； 1：忙。	0x0
[6]	RO	int_start	主机开始条件发送结束中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[5]	RO	int_stop	主机停止条件发送结束中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[4]	RO	int_tx	主机发送中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[3]	RO	int_rx	主机接收中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[2]	RO	int_ack_err	从机 ACK 错误中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[1]	RO	int_arb_loss	总线仲裁失败中断标志。 0：无中断标志产生； 1：中断标志产生。	0x0
[0]	RO	int_done	总线传输完成中断标志。 0：无中断标志产生；	0x0

			1：中断标志产生。	

I2C_SCL_H

须知

在系统初始化时配置或配置前使 I2C_CTRL bit[7]=0。

I2C_SCL_H 为 I2C 总线 SCL 信号高电平周期数寄存器。用于配置 I2C 模块工作时 SCL 高电平周期数。

Offset Address: 0x10　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	scl_h	配置数值乘 2 等于 SCL 高电平周期数。	0x0000

I2C_SCL_L

须知

在系统初始化时配置或配置前使 I2C_CTRL bit[7]=0。

I2C_SCL_L 为 I2C 总线 SCL 信号低电平周期数寄存器。用于配置 I2C 模块工作时 SCL 低电平周期数。

Offset Address: 0x14　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	scl_l	配置数值乘 2 等于 SCL 低电平周期数。	0x0000

I2C_TXR

须知

不使用 FIFO 模式下，发送结束后，I2C 模块不会修改 I2C_TXR 内容；使用 FIFO 模式下，写入的数据会自动载入到发送 FIFO 中保存直到该数据发送结束。

I2C_TXR 为 I2C 发送数据寄存器。用于配置 I2C 模块工作时发送数据。

Offset Address: 0x18　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	i2c_txr	主机发送数据。	0x00

I2C_RXR

须知

不使用 FIFO 模式下，I2C_RXR 数据在 I2C_SR bit[3]=1 时，数据有效。同时数据将保持到下一个读操作前。使用 FIFO 模式下，读取 I2C_RXR 会直接从接收 FIFO 中取数据。

I2C_RXR 为 I2C 接收数据寄存器。用于主机接收从机数据。

Offset Address: 0x1C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	i2c_rxr	主机接收数据。	0x00

I2C_FIFOSTATUS

I2C_FIFOSTATUS 为 FIFO 状态寄存器。

Offset Address: 0x20　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	rxfe	接收 FIFO 空状态。 0：非空； 1：空。	0x0

[2]	RO	rxff	接收 FIFO 满状态。 0：未满； 1：满。	0x0
[1]	RO	txfe	发送 FIFO 空状态。 0：非空； 1：空。	0x0
[0]	RO	txff	发送 FIFO 满状态。 0：未满； 1：满。	0x0

I2C_TXCOUNT

I2C_TXCOUNT 为发送 FIFO 数据个数寄存器。

Offset Address: 0x24　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	WC	txcount	读该寄存器返回发送 FIFO 中的字符数，写该寄存器(任意值)将清空发送 FIFO。	0x00

I2C_RXCOUNT

I2C_RXCOUNT 为接收 FIFO 数据个数寄存器。

Offset Address: 0x28　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	WC	rxcount	读该寄存器返回接收 FIFO 中的字符数，写该寄存器(任意值)将清空接收 FIFO。	0x00

I2C_RXTIDE

I2C_RXTIDE 为接收 FIFO 的溢出阈值寄存器。

Offset Address: 0x2C　　Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RW	rxtide	设置 int_rxtide 中断的触发值。 RXFIFO 中的字符个数≥ I2C_RXTIDE[rxtide]时会触发接收 FIFO 溢出中断。	0x01

I2C_TXTIDE

I2C_TXTIDE 为发送 FIFO 的溢出阈值寄存器。

说明
TXFIFO 中的字符只有在成功发送后才会被移除。

Offset Address: 0x30　　Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RW	txtide	设置 int_txtide 中断的触发值。 TXFIFO 中的字符个数≤ I2C_TXTIDE[txtide]时会触发发送 FIFO 溢出中断。	0x01

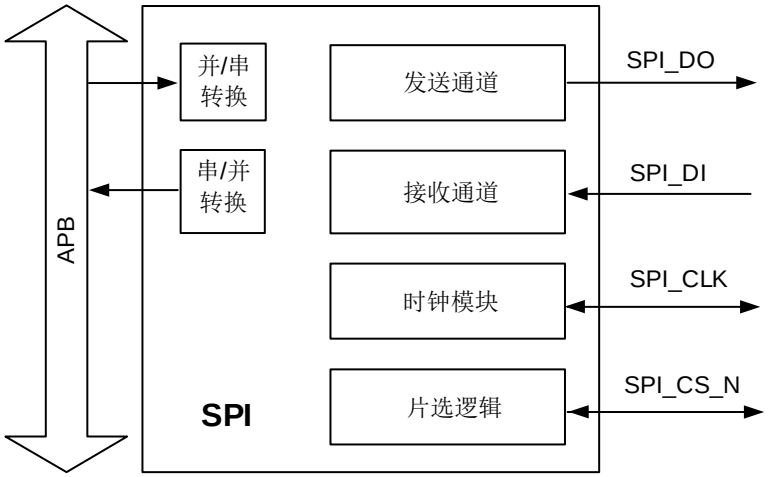
6.5 SPI

6.5.1 概述

SPI 实现数据的串并、并串转换，可以作为 Master 或 Slave 与外部设备进行同步串行通信（外围设备必须支持 SPI 帧格式）。

芯片的 SPI 工作参考时钟为 160MHz。SPI 功能框图如图 6-7 所示。

图6-7 SPI 功能框图



6.5.2 功能描述

SPI 具有以下功能特点：

- 支持接口时钟频率可编程。
  - 作为主设备：最大支持接口时钟频率的 4 分频。
  - 作为从设备：最大支持接口时钟频率的 8 分频。
- SPI0 与 SPI1 的 FIFO 规格不同。
  - SPI0：支持收/发分开的字宽为 256×16bit 的 FIFO（发送 FIFO 和接收 FIFO 各 1 个）。
  - SPI1：支持收/发分开的字宽为 64×16bit 的 FIFO（发送 FIFO 和接收 FIFO 各 1 个）。
- 支持 SPI 帧格式，分为以下三种：
  - Motorola 帧格式
  - TI（Texas Instruments）帧格式
  - National Microwire 帧格式
- 串行数据帧长度可编程：4bit～16bit。
- 支持发送 FIFO 中断、接收 FIFO 中断、接收超时中断和接收 FIFO 溢出中断独立屏蔽。
- 内部提供环回测试模式。
- 支持 DMA 操作，但不支持作为 DMA 的流控设备。

6.5.3 工作方式

6.5.3.1 接口信号

Master 模式

SPI 接口信号描述如表 6-13 所示。

表6-13 SPI Master 接口信号描述

信号名	宽度（bit）	方向	功能描述
SPI_CLK	1	O	SPI 串行时钟信号，由 Master 产生并控制。
SPI_CS_N	1	O	SPI 片选信号，低电平有效。
SPI_DI	1	I	输入数据。
SPI_DO	1	O	输出数据。

Slave 模式

SPI 接口信号描述如表 6-14 所示。

表6-14 SPI Slave 接口信号描述

信号名	宽度（bit）	方向	功能描述
SPI_CLK	1	I	SPI 串行时钟信号，由 Master 产生并控制。
SPI_CS_N	1	I	SPI 片选信号，低电平有效。
SPI_DI	1	I	输入数据。
SPI_DO	1	O	输出数据。

6.5.3.2 传输帧格式

SPI 可以与支持 SPI 帧格式的任意串行 Master 或 Slave 器件进行通信。

说明

图 6-8～图 6-15 中的缩略语含义为：MSB（Most Significant Bit），LSB（Least Significant Bit），Q（Q is an undefined signal）。

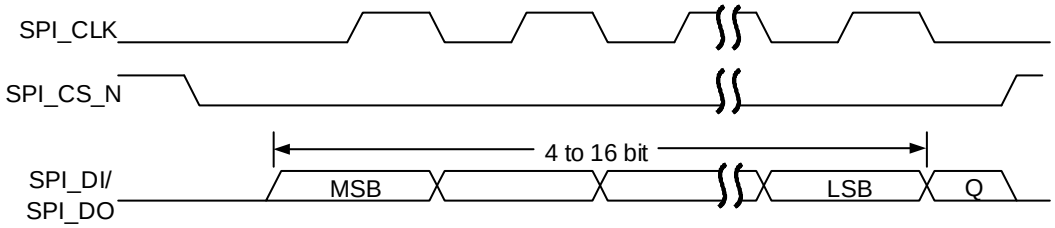
SPO 表示 SPICLKOUT 极性，SPH 表示 SPICLKOUT 相位。它们是寄存器 SPICR0 bit[7:6]。

Motorola 帧格式

【SPO=0、SPH=0】

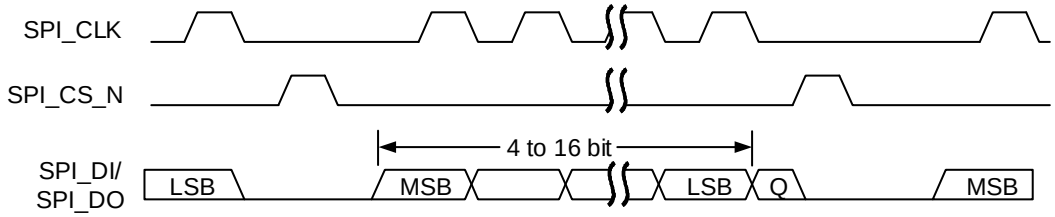
SPI 单帧帧格式如图 6-8 所示。

图6-8 SPI 单帧帧格式（SPO=0、SPH=0）



SPI 连续帧帧格式如图 6-9 所示。

图6-9 SPI 连续帧帧格式（SPO=0、SPH=0）



在该模式下，当 SPI 处于空闲状态时：

- SPI_CLK 信号设置为低电平。
- SPI_CS_N 信号设置为高电平。
- 发送数据线 SPI_DO 强制为低电平。

当 SPI 处于使能状态，而且发送 FIFO 内有有效数据时，设置 SPI_CS_N 信号为低电平表示开始传输数据。此时使能 Slave 的数据放入 Master 的输入 SPI_DI。半个 SPI_CLK 时钟周期之后，有效的 Master 数据传输到 SPI_DO。此时 Master 和 Slave 数据都已经有效，SPI_CLK 管脚在接下来的半个 SPI_CLK 时钟周期之后变为高电平。数据在 SPI_CLK 时钟的上升沿被捕获，在时钟的下降沿被传送。

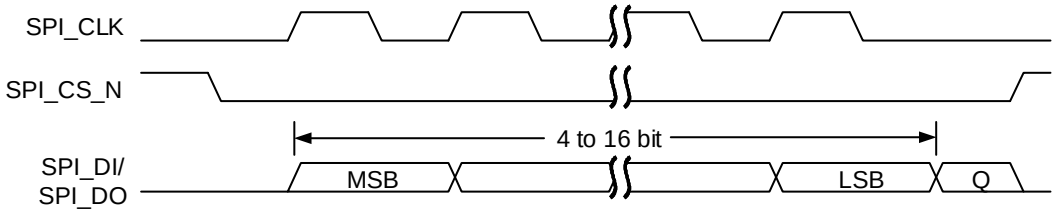
如果传输单个 word（4～16 bit），当捕捉到最后 1bit 数据时，SPI_CS_N 在接下来的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

如果是连续传输，SPI_CS_N 信号在每个 word（4～16 bit）传输之间会拉高一个时钟周期。这是因为 SPH 为 0 时，Slave 选择管脚会固定其内部串行设备寄存器的数据，使它不会变化。因此在连续传输时，主设备必须在每个 word（4～16 bit）传输之间将 SPI_CS_N 信号拉高。连续传输结束时，SPI_CS_N 在捕获到最后 1bit 之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

【SPO=0、SPH=1】

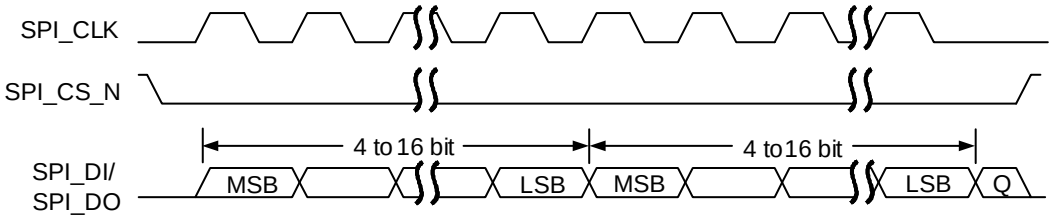
SPI 单帧帧格式如图 6-10 所示。

图6-10 SPI 单帧帧格式（SPO=0、SPH=1）



SPI 连续帧帧格式如图 6-11 所示。

图6-11 SPI 连续帧帧格式（SPO=0、SPH=1）



在该模式下，当 SPI 处于空闲状态时：

- SPI_CLK 信号设置为低电平。
- SPI_CS_N 信号设置为高电平。
- 发送数据线 SPI_DO 强制为低电平。

当 SPI 为使能状态，且发送 FIFO 内有有效数据时，设置 SPI_CS_N 信号为低电平表示开始传输数据。此时 Slave 的数据立刻发送到 Master 的接收数据线 SPI_DI。半个 SPI_CLK 时钟周期之后，Master 的有效数据在自身的传输线上有效，同时 SPI_CLK 产生第一个上升沿。数据在 SPI_CLK 时钟的下降沿被捕获，在时钟的上升沿被传送。

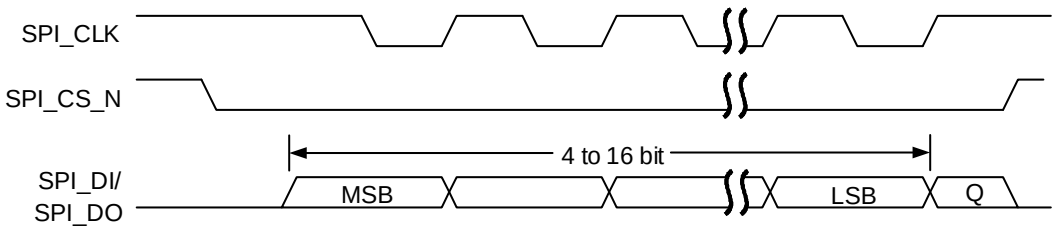
如果传输单个 word（4～16 bit），当捕捉到最后 1bit 数据时，SPI_CS_N 在接下来的 1 个 SPI_CLK 时钟之后恢复为高电平。

当连续传输时，在传输数据 word（4～16 bit）之间 SPI_CS_N 保持为低电平。连续传输结束时，SP_CS_N 在最后 1bit 捕获之后的 1 个 SPI_CLK 时钟之后恢复为高电平。

【SPO=1、SPH=0】

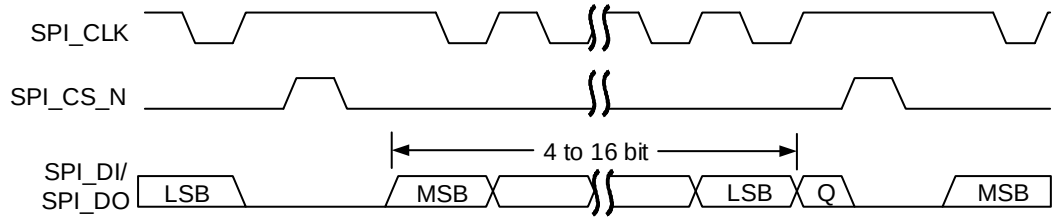
SPI 单帧帧格式如图 6-12 所示。

图6-12 SPI 单帧帧格式（SPO=1、SPH=0）



SPI 连续帧帧格式如图 6-13 所示。

图6-13 SPI 连续帧帧格式（SPO=1、SPH=0）



在该配置下，当 SPI 处于空闲状态时：

- SPI_CLK 信号设置为高电平。
- SPI_CS_N 信号设置为高电平。
- 发送数据线 SPI_DO 强制为低电平。

当 SPI 为使能状态，且发送 FIFO 内有有效数据时，设置 SPI_CS_N 信号为低电平表示开始传输数据。此时 Slave 的数据立刻发送到 Master 的接收数据线 SPI_DI。半个 SPI_CLK 周期之后，Master 的有效数据传送到 SPI_DO。再过半个 SPI_CLK 时钟周期之后，SPI_CLK Master 管脚设置为低电平，这表示数据在 SPI_CLK 时钟的下降沿被捕获，在 SPI_CLK 时钟的上升沿被传送。

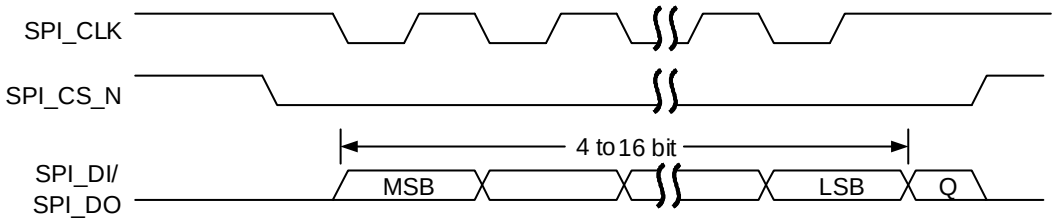
如果传输单个 word（4～16 bit），当捕捉到最后 1bit 数据时，SPI_CS_N 在接下来的 1 个 SPI_CLK 时钟之后恢复为高电平。

如果是连续的传输，SPI_CS_N 信号在每个 word（4～16 bit）传输之间必须拉高。这是因为当 SPH 为 0 时，Salve 选择管脚固定其内部串行设备寄存器的数据，使它不会变化。SPI_CS_N 在捕获到最后 1bit 数据之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

【SPO=1、SPH=1】

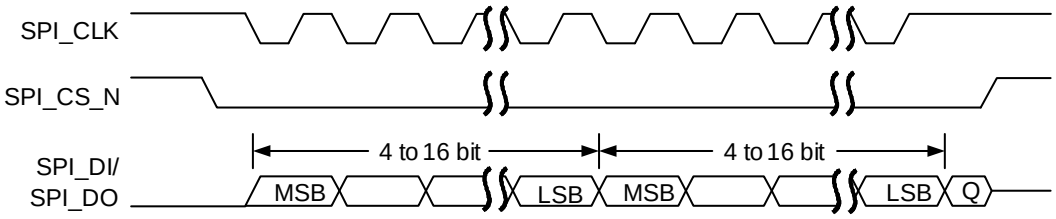
SPI 单帧帧格式如图 6-14 所示。

图6-14 SPI 单帧帧格式（SPO=1、SPH=1）



SPI 连续帧帧格式如图 6-15 所示。

图6-15 SPI 连续帧帧格式（SPO=1、SPH=1）



在该模式下，当 SPI 处于空闲状态时：

- SPI_CLK 信号设置为高电平。
- SPI_CS_N 信号设置为高电平。
- 发送数据线 SPI_DO 强制为低电平。

当 SPI 为使能状态，且发送 FIFO 内有有效数据时，设置 SPI_CS_N Master 信号为低电平表示开始传输数据。此时 Slave 的数据立刻发送到 Master 的接收数据线 SPI_DI。半个 SPI_CLK 时钟周期后，Master 数据在自身的传输线上有效，同时时钟 SPI_CLK 产生第一个下降沿。数据在 SPI_CLK 时钟的上升沿被捕获，在时钟的下降沿被传送。

当传输单个 word（4～16 bit）时，SPI_CS_N 在传输的最后 1bit 捕获之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

如果是连续传输，SPI_CS_N 信号始终保持为低电平。SPI_CS_N 在捕获到最后 1bit 之后的 1 个 SPI_CLK 时钟周期之后恢复到高状态。对于连续传输来说，SPI_CS_N 在传输过程中一直保持为低电平，结束方式与单个传输方式相同。

6.5.3.3 工作模式

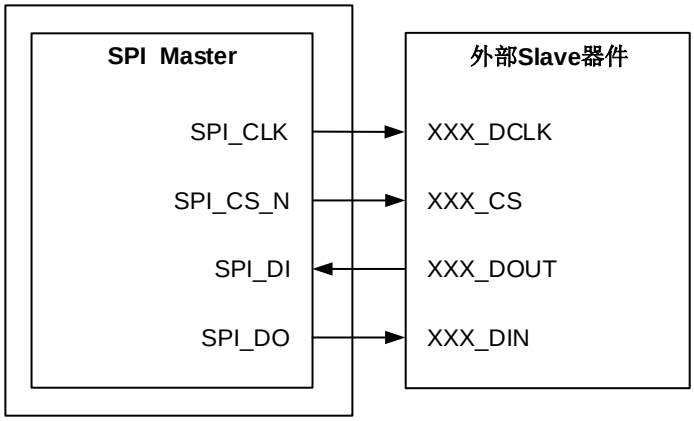
SPI 的工作方式分为中断或查询方式下的数据传输。

SPI 模块支持 Master 以及 Slave 两种基本工作模式。

Master 模式

在 Master 模式下，SPI 模块初始化并直接控制所有的外围设备。配置为 Master 的 SPI 与外围 Slave 器件通信示意图如图 6-16 所示。

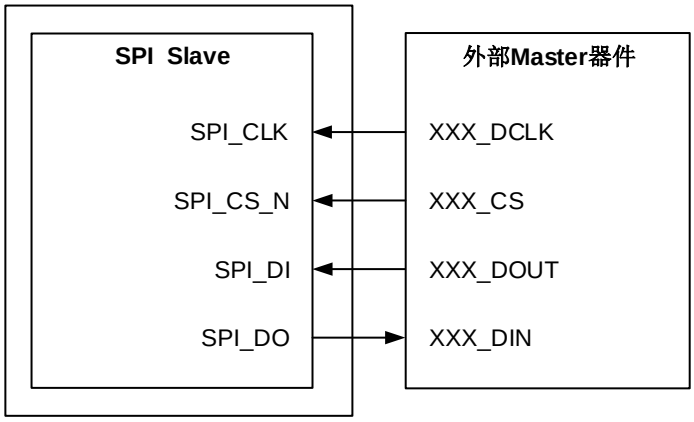
图6-16 SPI 接单 Slave 时的应用



Slave 模式

在 Slave 模式下，SPI 由串行总线上的 Master 进行初始化和控制。配置为 Slave 的 SPI 与外围 Master 器件通信示意图如图 6-17 所示。

图6-17 SPI 接 Master 时的应用



6.5.3.4 中断处理

SPI 有 5 个中断（其中前 4 个是独立中断源，可屏蔽，高电平有效）：

- SPIRXINTR

  接收 FIFO 中断请求。当接收 FIFO 中数据量达到水线或具有更多的有效数据时，该中断置位。

- SPITXINTR

  发送 FIFO 中断请求。当发送 FIFO 中数据量达到水线或具有更少的有效数据时，该中断置位。

- SPIRORINTR

接收 overrun 中断请求。当 FIFO 已满，且又有新的数据需要写入 FIFO 时，会引起 FIFO overrun，该中断置位。此时数据被写入接收移位寄存器，而不是 FIFO。

- SPIRTINTR

  接收 time out 中断请求。当接收 FIFO 非空，且 SPI 处于 idle 态超过一个固定的 32bit 周期，该中断置位。

  此时表明接收 FIFO 中仍有数据需要传输。如果接收 FIFO 被读空或者当有新的数据被接收到 SPIRXD 中，该中断解除置位。

- SPIINTR

  组合中断，为以上 4 个中断经过“或”运算后的结果。如果上述 4 个独立中断中任意一个置位且使能，该中断置位。

6.5.3.5 应用说明

以中断或查询方式下的数据传输为例。

初始化

初始化步骤如下：

1. 写 SPICR1[SSE]为 0，禁止 SPI。
2. 配置 SPICR0，设定帧格式及传输数据位宽、SPICLKOUT 相位和极性等参数。
3. 配置 SPICPSR，设定时钟分频因子。
4. 中断方式下，设置 SPIIMSC，屏蔽相应中断信号；查询方式下，应屏蔽所有产生的中断信号。

----结束

数据发送

发送定长数据（以 4 个数据为例）的步骤如下：

1. 写 SPICR1[sse]为 1，使能 SPI。
2. 读 SPISR，判断 FIFO 空满状态。如果 SPISR[tnf]为 1，说明发送 FIFO 不满，写 4 个数据到 SPIDR；如果 FIFO 满，则等待前面的数据发送，直至 FIFO 空后才可以输入数据。
3. 读 SPISR，判断 FIFO 空满状态，直至完全发送。
4. 写 SPICR1[sse]为 0，禁止 SPI。

----结束

数据接收

接收定长数据（以 4 个数据为例）的步骤如下：

1. 写 SPICR1[sse]为 1，使能 SPI。

2. 读 SPISR，判断接收 FIFO 状态。如果接收 FIFO 不空（SPISR[rne]=1），则读 SPIDR，接收 4 个数据。
3. 查询状态寄存器 SPISR[bsy]，直至不忙。
4. 写 SPICR1[sse]为 0，禁止 SPI。

----结束

6.5.4 寄存器概览

SPI 寄存器概览如表 6-15 所示。

表6-15 SPI 寄存器概览（SPI0 基址是 0x4005_8000、SPI1 基址是 0x4005_9000）

偏移地址	名称	描述	页码
0x000	SPICR0	SPI 控制寄存器 0	6-81
0x004	SPICR1	SPI 控制寄存器 1	6-83
0x008	SPIDR	SPI 发送/接收数据寄存器	6-83
0x00C	SPISR	SPI 状态寄存器	6-84
0x010	SPICPSR	SPI 时钟分频寄存器	6-84
0x014	SPIIMSC	SPI 中断屏蔽寄存器	6-85
0x018	SPIRIS	SPI 原始中断状态寄存器	6-85
0x01C	SPIMIS	SPI 屏蔽后中断状态寄存器	6-86
0x020	SPIICR	SPI 中断清除寄存器	6-86
0x024	SPIDMACR	SPI DMA 功能使能寄存器	6-87
0x028	SPITXFIFOCR	SPI 发送 FIFO 控制寄存器	6-87
0x02C	SPIRXFIFOCR	SPI 接收 FIFO 控制寄存器	6-88

6.5.5 寄存器描述

SPICR0

SPICR0 为 SPI 控制寄存器 0。

Offset Address: 0x000　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	RW	scr	串行时钟率，取值范围 0～255。 SCR 的值用于产生 SPI 发送和接收的比特	0x00

			率，公式为： SSPCLKOUT=FSSPCLK/[CPSDVSR×(1+SCR)]。 其中，CPSDVSR 是一个 2～254 之间的偶数，由寄存器 SPICPSR 配置。 **注意：从设备的分频值[CPSDVSR×(1+SCR)]必须≥8，主设备应保证数据传输速率(SSPCLKOUT)≤从设备的传输速率。**	
[7]	RW	sph	SPICLKOUT 相位。 具体含义请参见《Hi3861V100/Hi3861LV100/Hi3881V100 WiFi 芯片 硬件用户指南》中 SPI 接口的 SPI 帧格式。	0x0
[6]	RW	spo	SPICLKOUT 极性。 具体含义请参见《Hi3861V100/Hi3861LV100/Hi3881V100 WiFi 芯片 硬件用户指南》中 SPI 接口的 SPI 帧格式。	0x0
[5:4]	RW	frf	帧格式选择。 00：Motorola SPI 帧格式； 01：TI 同步串行帧格式； 10：National Microwire 帧格式； 11：保留。	0x0
[3:0]	RW	dss	设置数据位宽。 0x3：4bit； 0x4：5bit； 0x5：6bit； 0x6：7bit； 0x7：8bit； 0x8：9bit； 0x9：10bit； 0xA：11bit； 0xB：12bit； 0xC：13bit； 0xD：14bit； 0xE：15bit； 0xF：16bit； 其他：保留。	0x0

SPICR1

SPICR1 为 SPI 控制寄存器 1。

Offset Address: 0x004　　Total Reset Value: 0x7F00

Bits	Access	Name	Description	Reset
[15:5]	-	reserved	保留。	0x3F8
[4]	RW	bigend	设置数据大小端模式。 0：小端结束； 1：大端结束。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	ms	设置 Master 或 Slave 模式。 0：Master 模式(默认值)； 1：Slave 模式。 **注意：此位只能在 SPI 被禁止时改变。**	0x0
[1]	RW	sse	设置 SPI 使能。 0：禁止； 1：使能。	0x0
[0]	RW	lbm	设置环回模式。 0：正常的串行接口操作使能； 1：发送串行移位寄存器的输出在内部连接到接收串行移位寄存器的输入上。	0x0

SPIDR

SPIDR 为 SPI 发送/接收数据寄存器。

Offset Address: 0x008　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	data	发送/接收 FIFO。 读：接收 FIFO； 写：发送 FIFO。 如果数据少于 16bit 则必须右对齐。发送逻辑将忽略高位未使用的比特位，接收逻辑则自动将数据右对齐。	0x0000

SPISR

SPISR 为 SPI 状态寄存器。

Offset Address: 0x00C　　Total Reset Value: 0x0003

Bits	Access	Name	Description	Reset
[15:5]	-	reserved	保留。	0x000
[4]	RO	bsy	SPI 忙标记。 0：空闲； 1：忙。	0x0
[3]	RO	rff	接收 FIFO 是否已满。 0：未满； 1：已满。	0x0
[2]	RO	rne	接收 FIFO 是否未空。 0：已空； 1：未空。	0x0
[1]	RO	tnf	发送 FIFO 是否未满。 0：已满； 1：未满。	0x1
[0]	RO	tfe	发送 FIFO 是否已空。 0：未空； 1：已空。	0x1

SPICPSR

SPICPSR 为 SPI 时钟分频寄存器。

Offset Address: 0x010　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x00
[7:0]	RW	cpsdvsr	时钟分频因子。 此值必须是 2～254 之间的偶数，取决于输入时钟 SPICLK 的频率。最低位读作“0”。	0x00

SPIIMSC

SPIIMSC 为 SPI 中断屏蔽寄存器。

Offset Address: 0x014　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	保留。	0x000
[3]	RW	txim	发送 FIFO 半空或更少情况下中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[2]	RW	rxim	接收 FIFO 半空或更少情况下中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[1]	RW	rtim	接收超时中断屏蔽。 0：屏蔽； 1：不屏蔽。	0x0
[0]	RW	rorim	接收 FIFO 溢出中断屏蔽。 0：屏蔽； 1：不屏蔽。 值为 1 时，使能硬件流控功能，即接收 FIFO 满后 SPI 停止发送数据。	0x0

SPIRIS

SPIRIS 为 SPI 原始中断状态寄存器。

Offset Address: 0x018　　Total Reset Value: 0x0008

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	保留。	0x000
[3]	RO	txris	发送 FIFO 中断的原始中断状态。 0：无中断； 1：有中断。	0x1
[2]	RO	rxris	接收 FIFO 中断的原始中断状态。 0：无中断； 1：有中断。	0x0
[1]	RO	rtris	接收超时中断的原始中断状态。	0x0

			0：无中断； 1：有中断。	
[0]	RO	rorris	接收溢出中断的原始中断状态。 0：无中断； 1：有中断。	0x0

SPIMIS

SPIMIS 为 SPI 屏蔽后中断状态寄存器。

Offset Address: 0x01C　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	保留。	0x000
[3]	RO	txmis	发送 FIFO 中断屏蔽后的状态。 0：无中断； 1：有中断。	0x0
[2]	RO	rxmis	接收 FIFO 中断屏蔽后的状态。 0：无中断； 1：有中断。	0x0
[1]	RO	rtmis	接收超时中断屏蔽后的状态。 0：无中断； 1：有中断。	0x0
[0]	RO	rormis	接收溢出中断屏蔽后的状态。 0：无中断； 1：有中断。	0x0

SPIICR

SPIICR 为 SPI 中断清除寄存器。

Offset Address: 0x020　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:2]	-	reserved	保留。	0x0000
[1]	WC	rtic	清除接收超时中断。 写 0：无影响；	0x0

			写 1：清除中断。	
[0]	WC	roric	清除接收溢出中断。 写 0：无影响； 写 1：清除中断。	0x0

SPIDMACR

SPIDMACR 为 SPI DMA 功能的使能寄存器。

Offset Address: 0x024　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:2]	-	reserved	保留。	0x0000
[1]	RW	dmatxen	SPI DMA TX 功能使能。 0：禁止； 1：使能。	0x0
[0]	RW	dmarxen	SPI DMA RX 功能使能。 0：禁止； 1：使能。	0x0

SPITXFIFOCR

SPITXFIFOCR 为 SPI 发送 FIFO 控制寄存器。

Offset Address: 0x028　　Total Reset Value: 0x0009

Bits	Access	Name	Description	Reset
[15:6]	-	reserved	保留。	0x000
[5:3]	RW	txintsize	配置发送 FIFO 请求中断的水线。即发送 FIFO 中数据数目≤TXINTSize 所配置的字数时，TXRIS 有效。 SPI0： 000：1； 001：4； 010：8； 011：16； 100：32； 101：64；	0x1

			110：128； 111：192。 SPI1： 000：1； 001：4； 010：8； 011：16； 100：32； 101：48； 110：56； 111：64。	
[2:0]	RW	dmatxbrsizepclk	配置发送 FIFO 的 Burst 请求(根据配置的水线档位，当发送 FIFO 中数据量≤相应的档位时，可以产生 DMA 的 Burst 请求)。 SPI0： 000：水线档位为 255； 001：水线档位为 252； 010：水线档位为 248； 011：水线档位为 240； 100：水线档位为 224； 101：水线档位为 192； 110：水线档位为 128； 111：水线档位为 64。 SPI1： 000：水线档位为 63； 001：水线档位为 60； 010：水线档位为 48； 011：水线档位为 32； 100：水线档位为 16； 101：水线档位为 8； 110：水线档位为 4； 111：水线档位为 1。	0x1

SPIRXFIFOCR

SPIRXFIFOCR 为 SPI 接收 FIFO 控制寄存器。

Offset Address: 0x02C　　Total Reset Value: 0x0009

Bits	Access	Name	Description	Reset
[15:6]	-	reserved	保留。	0x000
[5:3]	RW	rxintsize	配置接收 FIFO 请求中断的水线。即接收 FIFO 中数据数目≥RXINTSize 所配置的字数时，RXRIS 有效，此处字长是 16bit。 SPI0： 000：255； 001：252； 010：248； 011：240； 100：224； 101：192； 110：128； 111：32。 SPI1： 000：65； 001：62； 010：48； 011：32； 100：16； 101：8； 110：4； 111：1。	0x1
[2:0]	RW	dmarxbrsizepclk	配置接收 FIFO 的 Burst 请求(根据配置的水线档位，当接收 FIFO 中数据量≥相应的档位时，可以产生 DMA 的 Burst 请求)。 SPI0： 000：水线档位为 1； 001：水线档位为 4； 010：水线档位为 8； 011：水线档位为 16； 100：水线档位为 32； 101：水线档位为 64； 110：水线档位为 128； 111：水线档位为 192。 SPI1： 000：水线档位为 1；	0x1

			001：水线档位为 4； 010：水线档位为 8； 011：水线档位为 16； 100：水线档位为 32； 101～111：水线档位为 64。	

6.6 PWM

6.6.1 概述

PWM 模块用于生成 PWM 信号，调节灯的亮度等功能。

6.6.2 功能描述

PWM 模式具有以下功能特点：

- 共集成 6 路 PWM。
- 每一路的 PWM 时钟可以单独门控。
- 可对晶体时钟（40/24M）或 160M 时钟进行 1～65535 倍分频。
- PWM 信号占空比可调整范围：1～1/65535。

6.6.3 工作方式

PWM 模块有 2 个时钟可以选择：晶体时钟（40/24M）、160M。

以配置 PWM 信号对 160M 时钟分频、分频倍数为 25 倍、占空比为 1/5 为例，配置步骤如下：

1. 写 CLK_SEL[pwm_clk_sel]为 0，选择 PWM 时钟为 160M。
2. 写 PWM_EN[pwm_en]为 1，使能 PWM 信号输出。
3. 写 PWM_FREQ[pwm_freq]为 0x19，确定对 PWM 时钟的分频倍数为 25。
4. 写 PWM_DUTY[pwm_duty]为 0x5，确定 PWM 信号的占空比为 PWM_DUTY[pwm_duty]/PWM_FREQ[pwm_freq] = 1/5。
5. 写 PWM_START[pwm_start]为 1，使能步骤 2、3、4 的配置。

----结束

6.6.4 寄存器概览

PWM 寄存器概览如表 6-16 所示。

表6-16 PWM 寄存器概览（PWM0 基址是 0x4004_0000、PWM1 基址是 0x4004_0100、PWM2 基址是 0x4004_0200、PWM3 基址是 0x4004_0300、PWM4 基址是 0x4004_0400、PWM5 基址是 0x4004_0500）

偏移地址	名称	描述	页码
0x00	PWM_EN	PWM 使能寄存器	
0x04	PWM_START	PWM 配置生效寄存器	
0x08	PWM_FREQ	PWM 频率控制计数值寄存器	
0x0C	PWM_DUTY	PWM 占空比计数值寄存器	

6.6.5 寄存器描述

PWM_EN

PWM_EN 为 PWM 使能寄存器。

Offset Address: 0x00　　Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	pwm_en	PWM 功能使能。 0：禁止，pwm_out 输出持续为 0； 1：使能。	0x1

PWM_START

PWM_START 为 PWM 配置生效寄存器。

Offset Address: 0x04　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	pwm_start	PWM 配置生效寄存器。 0：无操作； 1：此前赋值的 PWM 相关寄存器生效，逻辑自清零。	0x0

PWM_FREQ

PWM_FREQ 为 PWM 频率控制计数值寄存器。

Offset Address: 0x08　　Total Reset Value: 0x0000_05DC

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	pwm_freq	PWM 频率控制计数值，即对 PWM 时钟的分频倍数，取值范围 1～65535。	0x05DC

PWM_DUTY

PWM_DUTY 为 PWM 占空比计数值寄存器。

Offset Address: 0x0C　　Total Reset Value: 0x0000_02EE

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	pwm_duty	PWM 占空比控制计数值，取值范围 1～65535。 pwm_duty 与 pwm_freq 的比值为占空比。	0x02EE

6.7 HPM

6.7.1 概述

HPM 是一款硬件性能监测模块，通过使用 SVB（Selective Voltage Binning）技术，对芯片进行动态功耗调整，对于不同的 Corner 芯片采用不同的电压，进行动态功耗调整。

6.7.2 功能描述

HPM 模块具有以下功能特点：

- 集成了 3 套工艺（SVT（Standard Voltage Threshold）、HVT（High Voltage Threshold）、LVT（Low Voltage Threshold））的 HPM。
- HPM 时钟分频比可配置。
- 支持循环监控与单次监控。

6.7.3 工作方式

芯片中集成了 3 套工艺（SVT、HVT、LVT）的 HPM，下面以 SVT 工艺的 HPM 模块为例，配置步骤如下：

1. 写 PERI_PMC22[hpm0_div]为 1，配置时钟为 HPM 工作时钟的 2 分频（48MHz）。
2. 写 PERI_PMC22[hpm0_monitor_en]为 1，使能 HPM 模块功能。
3. 读 PERI_PMC23[hpm0_pc_record0]和[hpm0_pc_record1]、PERI_PMC24[hpm0_pc_record2]和[hpm0_pc_record3]。
4. 将步骤 3 中的 4 个读取值做平均，该平均值即最终的原始码字，值越大说明硬件性能（振荡器频率）越好，以此监测芯片性能。

----结束

6.7.4 寄存器概览

HPM 寄存器概览如表 6-17 所示。

表6-17 HPM 寄存器概览（基址是 0x4006_8000）

偏移地址	名称	描述	页码
0x0058	PERI_PMC22	HPM0 相关的时钟及软复位控制寄存器	6-94
0x005C	PERI_PMC23	HPM0 原始码字及报警信息寄存器	6-94
0x0060	PERI_PMC24	HPM0 原始码字及 RCC 码上报寄存器	6-95
0x0064	PERI_PMC25	HPM0 原始码型门限配置寄存器	6-96
0x0068	PERI_PMC26	HPM1 相关的时钟及软复位控制寄存器	6-96
0x006C	PERI_PMC27	HPM1 原始码字及报警信息寄存器	6-97
0x0070	PERI_PMC28	HPM1 原始码字及 RCC 码上报寄存器	6-98
0x0074	PERI_PMC29	HPM1 原始码型门限配置寄存器	6-98
0x0078	PERI_PMC30	HPM2 相关的时钟及软复位控制寄存器	6-98
0x007C	PERI_PMC31	HPM2 原始码字及报警信息寄存器	6-99
0x0080	PERI_PMC32	HPM2 原始码字及 RCC 码上报寄存器	6-100
0x0084	PERI_PMC33	HPM2 原始码型门限配置寄存器	6-100

6.7.5 寄存器描述

PERI_PMC22

PERI_PMC22 为 HPM0 相关的时钟及软复位控制寄存器。

Offset Address: 0x0058　　Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	hpm0_rst_req	HPM0 复位请求使能。 0：禁止； 1：使能。	0x0
[26]	RW	hpm0_monitor_en	循环监控 HPM0 的使能。 0：禁止； 1：使能。	0x0
[25]	RW	hpm0_bypass	HPM0 单次使能信号。 0：HPM0 的启动信号由[hpm0_monitor_en]决定； 1：HPM0 的启动信号由[hpm0_en]决定。	0x0
[24]	RW	hpm0_en	使能一次 HPM0 测量过程。 0：开始一次流程之前，此值需保持为 0； 1：开始一次调频流程。	0x0
[23:22]	RW	reserved	保留。	0x0
[21:12]	RW	hpm0_offset	HPM0 原始码字右移后的偏移量配置值。	0x000
[11:10]	RW	reserved	保留。	0x0
[9:8]	RW	hpm0_shift	HPM0 原始码字右移位数。	0x0
[7:6]	RW	reserved	保留。	0x0
[5:0]	RW	hpm0_div	HPM0 时钟分频比配置，对应([hpm0_div]+1)分频。 支持 2～64 分频。	0x0A

PERI_PMC23

PERI_PMC23 为 HPM0 原始码字及报警信息寄存器。

Offset Address: 0x005C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RO	hpm0_up_warning	HPM0 的原始码字超过上门限的上报标志。 0：未超过上门限。 1：超过上门限。	0x0
[24]	RO	hpm0_low_warning	HPM0 的原始码字低于下门限的上报标志。 0：未低于下门限。 1：低于下门限。	0x0
[23:22]	-	reserved	保留。	0x0
[21:12]	RO	hpm0_pc_record1	HPM0 原始码字上报值(上一次的历史值)。	0x000
[11]	-	reserved	保留。	0x0
[10]	RO	hpm0_pc_valid	HPM0 输出有效指示。 0：对应[hpm0_pc_record0]的上一次读取值； 1：对应[hpm0_pc_record0]的当次读取值。	0x0
[9:0]	RO	hpm0_pc_record0	HPM0 原始码字上报值(当前值)。	0x000

PERI_PMC24

PERI_PMC24 为 HPM0 原始码字及 RCC 码上报寄存器。

Offset Address: 0x0060　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28:24]	RO	hpm0_rcc	HPM0 输出的 RCC 码。	0x00
[23:22]	-	reserved	保留。	0x0
[21:12]	RO	hpm0_pc_record3	HPM0 原始码字上报值(上三次的历史值)。	0x000
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	hpm0_pc_record2	HPM0 原始码字上报值(上两次的历史值)。	0x000

PERI_PMC25

PERI_PMC25 为 HPM0 原始码型门限配置寄存器。

Offset Address: 0x0064　　Total Reset Value: 0x0100_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	hpm0_monitor_period	HPM0 循环监控周期。 如果配置值为 N，则监控的周期 T = N×2048/1000(ms)。 其中：N 最大为 0xFF，监控最大间隔为 522ms，最小间隔为 2ms。	0x01
[23]	RW	reserved	保留。	0x0
[22:12]	RW	hpm0_lowlimit	HPM0 原始码型下限。	0x000
[11:10]	RW	reserved	保留。	0x0
[9:0]	RW	hpm0_uplimit	HPM0 原始码型上限。	0x000

PERI_PMC26

PERI_PMC26 为 HPM1 相关的时钟及软复位控制寄存器。

Offset Address: 0x0068　　Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	hpm1_rst_req	HPM1 复位请求使能。 0：禁止； 1：使能。	0x0
[26]	RW	hpm1_monitor_en	循环监控 HPM1 的使能。 0：禁止； 1：使能。	0x0
[25]	RW	hpm1_bypass	HPM1 单次使能信号。 0：HPM1 的启动信号由[hpm1_monitor_en]决定； 1：HPM1 的启动信号由[hpm1_en]决定。	0x0
[24]	RW	hpm1_en	使能一次 HPM1 测量过程。 0：开始一次流程之前，此值需保持为 0； 1：开始一次调频流程。	0x0

[23:22]	RW	reserved	保留。	0x0
[21:12]	RW	hpm1_offset	HPM1 原始码字右移后的偏移量配置值。	0x000
[11:10]	RW	reserved	保留。	0x0
[9:8]	RW	hpm1_shift	HPM1 原始码字右移位数。	0x0
[7:6]	RW	reserved	保留。	0x0
[5:0]	RW	hpm1_div	HPM1 时钟分频比配置，对应([hpm1_div]+1)分频。 支持 2～64 分频。	0x0A

PERI_PMC27

PERI_PMC27 为 HPM1 原始码字及报警信息寄存器。

Offset Address: 0x006C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RO	hpm1_up_warning	HPM1 的原始码字超过上门限的上报标志。 0：未超过上门限。 1：超过上门限。	0x0
[24]	RO	hpm1_low_warning	HPM1 的原始码字低于下门限的上报标志。 0：未低于下门限。 1：低于下门限。	0x0
[23:22]	-	reserved	保留。	0x0
[21:12]	RO	hpm1_pc_record1	HPM1 原始码字上报值(上一次的历史值)。	0x000
[11]	-	reserved	保留。	0x0
[10]	RO	hpm1_pc_valid	HPM1 输出有效指示。 0：对应[hpm1_pc_record0]的上一次读取值； 1：对应[hpm1_pc_record0]的当次读取值。	0x0
[9:0]	RO	hpm1_pc_record0	HPM1 原始码字上报值(当前值)。	0x000

PERI_PMC28

PERI_PMC28 为 HPM1 原始码字及 RCC 码上报寄存器。

Offset Address: 0x0070　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28:24]	RO	hpm1_rcc	HPM1 输出的 RCC 码。	0x00
[23:22]	-	reserved	保留。	0x0
[21:12]	RO	hpm1_pc_record3	HPM1 原始码字上报值(上三次的历史值)。	0x000
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	hpm1_pc_record2	HPM1 原始码字上报值(上两次的历史值)。	0x000

PERI_PMC29

PERI_PMC29 为 HPM1 原始码型门限配置寄存器。

Offset Address: 0x0074　　Total Reset Value: 0x0100_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	hpm1_monitor_period	HPM1 循环监控周期。 如果配置值为 N，则监控的周期 T = N×2048/1000(ms)。 其中：N 最大为 0xFF，监控最大间隔为 522ms，最小间隔为 2ms。	0x01
[23]	RO	reserved	保留。	0x0
[22:12]	RW	hpm1_lowlimit	HPM1 原始码型下限。	0x000
[11:10]	RW	reserved	保留。	0x0
[9:0]	RW	hpm1_uplimit	HPM1 原始码型上限。	0x000

PERI_PMC30

PERI_PMC30 为 HPM2 相关的时钟及软复位控制寄存器。

Offset Address: 0x0078　　Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0

[27]	RW	hpm2_rst_req	HPM2 复位请求使能。 0：禁止； 1：使能。	0x0
[26]	RW	hpm2_monitor_en	循环监控 HPM2 的使能。 0：禁止； 1：使能。	0x0
[25]	RW	hpm2_bypass	HPM2 单次使能信号。 0：HPM2 的启动信号由[hpm2_monitor_en]决定； 1：HPM2 的启动信号由[hpm2_en]决定。	0x0
[24]	RW	hpm2_en	使能一次 HPM2 测量过程。 0：开始一次流程之前，此值需保持为 0； 1：开始一次调频流程。	0x0
[23:22]	RW	reserved	保留。	0x0
[21:12]	RW	hpm2_offset	HPM2 原始码字右移后的偏移量配置值。	0x000
[11:10]	RW	reserved	保留。	0x0
[9:8]	RW	hpm2_shift	HPM2 原始码字右移位数。	0x0
[7:6]	RW	reserved	保留。	0x0
[5:0]	RW	hpm2_div	HPM2 时钟分频比配置，对应([hpm2_div]+1)分频。 支持 2～64 分频。	0x0A

PERI_PMC31

PERI_PMC31 为 HPM2 原始码字及报警信息寄存器。

Offset Address: 0x007C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RO	hpm2_up_warning	HPM2 的原始码字超过上门限的上报标志。 0：未超过上门限。 1：超过上门限。	0x0
[24]	RO	hpm2_low_warning	HPM2 的原始码字低于下门限的上报标志。	0x0

			0：未低于下门限。 1：低于下门限。	
[23:22]	-	reserved	保留。	0x0
[21:12]	RO	hpm2_pc_record1	HPM2 原始码字上报值(上一次的历史值)。	0x000
[11]	-	reserved	保留。	0x0
[10]	RO	hpm2_pc_valid	HPM2 输出有效指示。 0：对应[hpm2_pc_record0]的上一次读取值； 1：对应[hpm2_pc_record0]的当次读取值。	0x0
[9:0]	RO	hpm2_pc_record0	HPM2 原始码字上报值(当前值)。	0x000

PERI_PMC32

PERI_PMC32 为 HPM2 原始码字及 RCC 码上报寄存器。

Offset Address: 0x0080　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28:24]	RO	hpm2_rcc	HPM2 输出的 RCC 码。	0x00
[23:22]	-	reserved	保留。	0x0
[21:12]	RO	hpm2_pc_record3	HPM2 原始码字上报值(上三次的历史值)。	0x000
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	hpm2_pc_record2	HPM2 原始码字上报值(上两次的历史值)。	0x000

PERI_PMC33

PERI_PMC33 为 HPM2 原始码型门限配置寄存器。

Offset Address: 0x0084　　Total Reset Value: 0x0100_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	hpm2_monitor_period	HPM2 循环监控周期。 如果配置值为 N，则监控的周期 T = N×2048/1000(ms)。 其中：N 最大为 0xFF，监控最大间隔为 522ms，最小间隔为 2ms。	0x01

[23]	RO	reserved	保留。	0x0
[22:12]	RW	hpm2_lowlimit	HPM2 原始码型下限。	0x000
[11:10]	RW	reserved	保留。	0x0
[9:0]	RW	hpm2_uplimit	HPM2 原始码型上限。	0x000

6.8 Tsensor

6.8.1 概述

Tsensor 是模拟温度检测 IP（即 1 个 8bit 并行输出数字温度传感器 IP），检测 DIE 的节温并以二进制形式输出温度信息。分辨率 0.71℃/LSB，8bit 数据 0～255 对应的温度范围为-40～+140 ℃。

IP 精度修调：IP 支持绝对精度线性修调，修调后的精度±3℃以内（修调指 ATE（Automatic Test Equipment）测试的 TRIM 修调，修调精度的进一步提高依赖于测试的精确度）。

6.8.2 功能描述

Tsensor 模块具有以下功能特点：

- 上报获取的 Tsensor IP 温度值。
  - 辅助算法校准，即计算 16 次温度采样平均值。
  - 支持软件可配是否计算单点 16 次平均值。
  - 利用中断的方式实时上报当前校准后的温度值。
- 支持高温低温门限的使用。
  - 通过中断的方式实时上报当前温度是否超过高温门限或低于低温门限。
  - 支持软件分别可配高温门限和低温门限。
- 支持超高温保护的使用。
  - 通过中断的方式实时上报当前温度是否超过超高温的温度门限。
  - 支持软件可配超高温的温度门限。
  - 支持产生过温保护信号提供给射频模块（RF）使用。
- 支持过温断电的使用。
  - 通过产生 PD（Power Down）信号实现关断逻辑的功能。
  - 支持软件可配过温断电的温度门限。
- 支持周期采样温度上报。
  - 支持软件可配周期采样间隔。
- 支持电源门控的控制。
  - 支持手动控制、自动控制两种方式。

- 支持 ATE 校准温度值。
  - 支持通过配置 TRIM 值的方式实现对 Tsensor IP 的温度校准。

6.8.3 工作方式

Tsensor 模块的工作模式分为以下 4 种：

- 正常检测温度值模式
- 高低温门限中断模式
- 过温保护中断模式
- RF 过温保护中断模式

说明

以上 4 种模式均为在检测温度值基础上进行，检测温度的模式有多种，此处检测温度的模式均为周期采样配合 16 次单点平均计算。

6.8.3.1 正常检测温度模式

正常检测温度模式配置步骤如下：

1. 写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel]为 0x1，写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man]为 0x1，通过手动模式打开 Tsensor 的电源门控。
2. 写 TSENSOR_AUTO_STS[tsensor_auto_clr]为 0x1，清除自动模式下产生温度有效信号。
3. 写 TSENSOR_TEMP_INT_CLR[tsensor_int_clr]为 0x1，清除 Tsensor 中断信号。
4. 写 TSENSOR_TEMP_INT_EN[tsensor_done_int_en]为 0x1，打开 Tsensor 的采集温度完成中断使能。
5. 写 TSENSOR_CTRL[tsensor_mode]为 0x0，选择 16 次平均值的单次上报方式。
6. 写 TSENSOR_CTRL[tsensor_enable]为 0x1，开启 Tsensor 的使能信号。
7. 写 TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period]，设置合适的周期采样的时间间隔。
8. 写 TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable]为 0x1，开启周期采样的使能信号。
9. 读 TSENSOR_TEMP_INT_STS[tsensor_done_int_sts]为 0x1，等待 16 点平均计算温度模式下的中断信号的产生。
10. 读 TSENSOR_AUTO_STS[tsensor_data_auto]，获取 16 点平均单次上报模式下的温度值。

----结束

6.8.3.2 高低温门限中断模式

高低温门限中断模式配置步骤如下：

1. 写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel]为 0x1，写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man]为 0x1，利用手动模式打开 Tsensor 的电源门控。
2. 写 TSENSOR_AUTO_STS[tsensor_auto_clr]为 0x1，清除自动模式下产生的 rdy 信号。
3. 写 TSENSOR_TEMP_INT_CLR[tsensor_int_clr]为 0x1，清除 Tsensor 中断信号。
4. 写 TSENSOR_TEMP_INT_EN[tsensor_out_thresh_int_en]为 0x1，打开 Tsensor 的超门限范围中断使能。
5. 写 TSENSOR_TEMP_LIMIT1[tsensor_temp_high_limit]、TSENSOR_TEMP_LIMIT2[tsensor_temp_low_limit]，设置合适的高低温门限值。
6. 写 TSENSOR_CTRL[tsensor_mode]为 0x0，选择 16 次平均值的单次上报方式。
7. 写 TSENSOR_CTRL[tsensor_enable]为 0x1，开启 Tsensor 的使能信号。
8. 写 TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period]，设置合适的周期采样的时间间隔。
9. 写 TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable]为 0x1，开启周期采样的使能信号。
10. 读 TSENSOR_TEMP_INT_STS[tsensor_out_thresh_int_sts]是否为 1，如果为 1，则当前的温度值高于高温门限或低于低温门限。

----结束

6.8.3.3 过温保护中断模式

过温保护中断模式配置步骤如下：

1. 写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel]为 0x1，写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man]为 0x1，利用手动模式打开 Tsensor 的电源门控。
2. 写 TSENSOR_AUTO_STS[tsensor_auto_clr]为 0x1，清除自动模式下产生的 rdy 信号。
3. 写 TSENSOR_TEMP_INT_CLR[tsensor_int_clr]为 0x1，清除 Tsensor 中断信号。
4. 写 TSENSOR_TEMP_INT_EN[tsensor_overtemp_int_en]为 0x1，打开 Tsensor 的过温中断使能。
5. 写 TSENSOR_OVER_TEMP[tsensor_overtemp_thresh]，设置合适的过温门限值。
6. 写 TSENSOR_OVER_TEMP[tsensor_overtemp_thresh_en]为 0x1，打开过温保护使能信号。
7. 写 TSENSOR_CTRL[tsensor_mode]为 0x0，选择 16 次平均值的单次上报方式。
8. 写 TSENSOR_CTRL[tsensor_enable]为 0x1，开启 Tsensor 的使能信号。
9. 写 TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period]，设置合适的周期采样的时间间隔。

10. 写 TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable]为 0x1，开启周期采样的使能信号。
11. 读 TSENSOR_TEMP_INT_STS[tsensor_overtemp_int_sts]是否为 1，如果为 1，则当前的温度值超过了设置的过温门限值。

----结束

6.8.3.4 RF 过温保护中断模式

RF 过温保护中断模式配置步骤如下：

1. 写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel]为 0x1，写 GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man]为 0x1，利用手动模式打开 Tsensor 的电源门控。
2. 写 TSENSOR_AUTO_STS[tsensor_auto_clr]为 0x1，清除自动模式下产生的 rdy 信号。
3. 写 RF_OVER_TEMP_INT_CLR[rf_over_temp_int_clr]为 0x1，清除 RF 过热有效中断信号。
4. 写 RF_OVER_TEMP_INT_EN[rf_over_temp_int_en]为 0x1，打开 RF 过热中断使能。
5. 写 TSENSOR_OVER_TEMP[tsensor_overtemp_thresh]，设置合适的过温门限值。
6. 写 TSENSOR_OVER_TEMP[tsensor_overtemp_thresh_en]为 0x1，打开过温保护使能信号。
7. 写 TSENSOR_CTRL[tsensor_mode]为 0x0，选择 16 次平均值的单次上报方式。
8. 写 TSENSOR_CTRL[tsensor_enable]为 0x1，开启 Tsensor 的使能信号。
9. 写 TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period]，设置合适的周期采样的时间间隔。
10. 写 TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable]为 0x1，开启周期采样的使能信号。
11. 读 RF_OVER_TEMP_INT_STS[rf_over_temp_int]是否为 1，如果为 1，则当前的温度值超过了设置的过温门限值，且作为一个有效信号传导给 RF 模块。

----结束

6.8.4 寄存器概览

Tsensor 寄存器概览如表 6-18 所示。

表6-18 Tsensor 寄存器概览（基址是 0x4002_8000）

偏移地址	名称	描述	页码
0x0490	RF_TEMP_MODE	RF TEMP 模式配置寄存器	6-105
0x04B4	RF_TEMP_STS	RF TEMP 过温状态寄存器	6-106

偏移地址	名称	描述	页码
0x04C0	RF_OVER_TEMP_INT_EN	RF TEMP 过温中断使能寄存器	6-107
0x04C4	RF_OVER_TEMP_INT_CLR	RF TEMP 过温中断清除寄存器	6-107
0x04C8	RF_OVER_TEMP_INT_STS	RF TEMP 过温中断寄存器	6-107
0x0500	TSENSOR_START	Tsensor 启动寄存器	6-107
0x0504	TSENSOR_CTRL	Tsensor 控制寄存器	6-108
0x0508	TSENSOR_MAN_STS	Tsensor 手动控制状态寄存器	6-108
0x050C	TSENSOR_AUTO_STS	Tsensor 自动控制状态寄存器	6-109
0x0510	TSENSOR_CTRL1	Tsensor 控制寄存器 1	6-110
0x0514	TSENSOR_TEMP_LIMIT1	Tsensor 温度门限上限寄存器	6-110
0x0518	TSENSOR_TEMP_LIMIT2	Tsensor 温度门限下限寄存器	6-111
0x051C	TSENSOR_OVER_TEMP	Tsensor 过温控制寄存器	6-111
0x0520	TSENSOR_TEMP_INT_EN	Tsensor 中断使能寄存器	6-112
0x0524	TSENSOR_TEMP_INT_CLR	RF TEMP 配置寄存器	6-112
0x0528	TSENSOR_TEMP_INT_STS	RF TEMP 配置寄存器	6-113
0x0530	TSENSOR_OVER_TEMP_PD	Tsensor 过温下电控制寄存器	6-113
0x0540	TSENSOR_AUTO_REFRESH_PERIOD	Tsensor 自动检测周期配置寄存器	6-114
0x0544	TSENSOR_AUTO_REFRESH_CFG	Tsensor 自动检测使能控制寄存器	6-114

6.8.5 寄存器描述

RF_TEMP_MODE

RF_TEMP_MODE 为 RF TEMP 模式配置寄存器。

Offset Address: 0x0490　　Total Reset Value: 0xFF44

Bits	Access	Name	Description	Reset
[15:7]	RW	reserved	保留。	0x1FE
[6]	RW	rf_over_temp_prt_hw_src	硬件上报来源选择。	0x1

			0：RF 内部 Tsensor 上报温度保护； 1：Tsensor 上报温度保护。 **注意：仅支持通过 Tsensor 上报的温度保护。**	
[5]	-	reserved	保留。	0x0
[4]	RW	rf_over_temp_prt_polar	过温保护信号极性配置。 0：逻辑产生的过温保护信号为 0 时表示有效，为 1 时表示无效； 1：逻辑产生的过温保护信号为 1 时表示有效，为 0 时表示无效。	0x0
[3]	RW	rf_over_temp_prt_sel	过温保护信号选择。 0：由硬件控制； 1：由手动控制。	0x0
[2]	RW	rf_over_temp_prt_man	过温保护手动控制信号。 0：手动控制过温保护信号有效； 1：手动控制过温保护信号无效。	0x1
[1:0]	-	reserved	保留。	0x0

RF_TEMP_STS

RF_TEMP_STS 为 RF TEMP 过温状态寄存器。

Offset Address: 0x04B4　　Total Reset Value: 0x000E

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	保留。	0x000
[3]	RO	tsensor_overtemp_prt	Tsensor 过温保护控制信号。 0：关断； 1：正常工作。	0x1
[2]	-	reserved	保留	0x1
[1]	RO	rf_over_temp_prt	过温保护控制信号(当检测的温度＞设定门限时)。 0：关断； 1：正常工作。	0x1
[0]	-	reserved	保留。	0x0

RF_OVER_TEMP_INT_EN

RF_OVER_TEMP_INT_EN 为 RF TEMP 过温中断使能寄存器。

Offset Address: 0x04C0　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	-	reserved	保留。	0x0000
[0]	RW	rf_over_temp_int_en	过热中断使能。 0：禁止； 1：使能。	0x0

RF_OVER_TEMP_INT_CLR

RF_OVER_TEMP_INT_CLR 为 RF TEMP 过温中断清除寄存器。

Offset Address: 0x04C4　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	-	reserved	保留。	0x0000
[0]	W1_PULSE	rf_over_temp_int_clr	RF 过温保护中断清除信号。 写 0：无效； 写 1：清除中断。	0x0

RF_OVER_TEMP_INT_STS

RF_OVER_TEMP_INT_STS 为 RF TEMP 过温中断寄存器。

Offset Address: 0x04C8　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	-	reserved	保留。	0x0000
[0]	RO	rf_over_temp_int	RF 过温保护中断状态。 0：中断状态无效； 1：中断状态有效。	0x0

TSENSOR_START

TSENSOR_START 为 Tsensor 启动寄存器。

Offset Address: 0x0500　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	-	reserved	保留。	0x0000
[0]	W1_PULSE	tsensor_start	自动模式刷新温度码信号。 写 0：无效； 写 1：自动模式下，刷新一次温度码，回读TSENSOR_AUTO_STS[tsensor_data_auto]获取当前温度值，当TSENSOR_AUTO_STS[tsensor_rdy_auto]为1 时，表明温度值有效。	0x0

TSENSOR_CTRL

TSENSOR_CTRL 为 Tsensor 控制寄存器。

Offset Address: 0x0504　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	保留。	0x000
[3]	RW	gate_tsensor_vddio_polar	Tsensor 电源门控极性配置。 0：Tsensor IP 工作使能为 0 关闭，为 1 开启； 1：Tsensor IP 工作使能为 1 关闭，为 0 开启。	0x0
[2:1]	RW	tsensor_mode	Tsensor 上报温度模式选择。 00：16 点平均单次上报模式； 01：16 点平均循环上报模式； 10、11：单点循环上报模式(该模式不比较阈值，仅上报温度码)。	0x0
[0]	RW	tsensor_enable	TSENSOR_CTRL 开关。 0：关闭； 1：打开。	0x0

TSENSOR_MAN_STS

TSENSOR_MAN_STS 为 Tsensor 手动控制状态寄存器。

Offset Address: 0x0508　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:10]	-	reserved	保留。	0x00
[9:2]	RO	tsensor_data_man	单点循环上报有效的温度码字。 表示温度值的码字 TEMP_OUT 共有 8 bit，换算为十进制值为 0～255，对应温度范围为-40℃～140℃。温度码输出呈现线性分布，即-40℃对应 0，140℃对应 255。 温度码换算公式：T(℃)= DEC(TEMP_OUT[7:0])×0.705+(-40)℃	0x00
[1]	RO	tsensor_rdy_man	单点循环上报模式下温度有效信号。 0：检测未启动或手动检测中； 1：[tsensor_data_man]值为有效的温度值。	0x0
[0]	W1_PULSE	tsensor_man_clr	清除单点循环上报模式的状态。 写 0：无效； 写 1：清除。	0x0

TSENSOR_AUTO_STS

TSENSOR_AUTO_STS 为 Tsensor 自动控制状态寄存器。

Offset Address: 0x050C　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:10]	-	reserved	保留。	0x00
[9:2]	RO	tsensor_data_auto	16 点平均单次上报模式或 16 点平均循环上报有效的温度码字。 表示温度值的码字 TEMP_OUT 共有 8 bit，换算为十进制值为 0～255，对应温度范围为-40℃～140℃。温度码输出呈现线性分布，即-40℃对应 0，140℃对应 255。 温度码换算公式：T(℃)= DEC(TEMP_OUT[7:0])×0.705+(-40)℃	0x00
[1]	RO	tsensor_rdy_auto	16 点平均单次上报模式或 16 点平均循环上报模式温度有效信号。 0：自动检测未启动或自动检测中； 1：[tsensor_data_auto]值为有效的温度值。	0x0
[0]	W1_PULSE	tsensor_auto_clr	清除 16 点平均单次上报模式或 16 点平均循环上报模式的状态。 写 0：无效；	0x0

			写 1：清除。	

TSENSOR_CTRL1

TSENSOR_CTRL1 为 Tsensor 控制寄存器 1。

Offset Address: 0x0510　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:5]	-	reserved	保留。	0x000
[4]	RW	tsensor_temp_trim_sel	Tsensor IP 的 Trim 修调功能的源头选择。 0：选择 Tsensor IP 的 temp_trim 由 EFUSE 直接加载； 1：选择 Tsensor IP 的 temp_trim 由寄存器配置。	0x0
[3:0]	RW	tsensor_temp_trim	校准 Tsensor IP 温度的 Trim 值。 该值参考 ATE 测试结果提供，配置值对应的温度调整值如下： 0x0：0.000℃； 0x1：1.410℃； 0x2：2.820℃； 0x3：4.230℃； 0x4：5.640℃； 0x5：7.050℃； 0x6：8.460℃； 0x7：9.870℃； 0x8：0.000℃； 0x9：-1.410℃； 0xA：-2.820℃； 0xB：-4.230℃； 0xC：-5.640℃； 0xD：-7.050℃； 0xE：-8.460℃； 0xF：-9.870℃。	0x0

TSENSOR_TEMP_LIMIT1

TSENSOR_TEMP_LIMIT1 为 Tsensor 温度门限上限寄存器。

Offset Address: 0x0514 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x00
[7:0]	RW	tsensor_temp_high_limit	16 点平均单次上报模式或 16 点平均循环上报模式下高温门限阈值。 表示温度值的码字 TEMP_OUT 共有 8 bit，换算为十进制值为 0～255，对应温度范围为-40℃～140℃。温度码输出呈现线性分布，即-40℃对应 0，140℃对应 255。 温度码换算公式：T(℃)= DEC(TEMP_OUT[7:0])×0.705+(-40)℃	0x00

TSENSOR_TEMP_LIMIT2

TSENSOR_TEMP_LIMIT2 为 Tsensor 温度门限下限寄存器。

Offset Address: 0x0518 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x00
[7:0]	RW	tsensor_temp_low_limit	16 点平均单次上报模式或 16 点平均循环上报模式下低温门限阈值。 表示温度值的码字 TEMP_OUT 共有 8 bit，换算为十进制值为 0～255，对应温度范围为-40℃～140℃。温度码输出呈现线性分布，即-40℃对应 0，140℃对应 255。 温度码换算公式：T(℃)= DEC(TEMP_OUT[7:0])×0.705+(-40)℃	0x00

TSENSOR_OVER_TEMP

TSENSOR_OVER_TEMP 为 Tsensor 过温控制寄存器。

Offset Address: 0x051C Total Reset Value: 0x00FF

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	保留。	0x00
[10]	RW	tsensor_overtemp_thresh_en	16 点平均单次上报模式或 16 点平均循环上报模式下过温 PA 保护使能。 0：禁止；	0x0

			1：使能。	
[9:8]	-	reserved	保留。	0x0
[7:0]	RW	tsensor_overtemp_thresh	16 点平均单次上报模式或 16 点平均循环上报模式下过温 PA 保护阈值。 表示温度值的码字 TEMP_OUT 共有 8 bit，换算为十进制值为 0～255，对应温度范围为-40℃～140℃。温度码输出呈现线性分布，即-40℃对应 0，140℃对应 255。 温度码换算公式：T(℃)= DEC(TEMP_OUT[7:0])×0.705+(-40)℃	0xFF

TSENSOR_TEMP_INT_EN

TSENSOR_TEMP_INT_EN 为 Tsensor 中断使能寄存器。

Offset Address: 0x0520　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:3]	-	reserved	保留。	0x0000
[2]	RW	tsensor_overtemp_int_en	Tsensor 温度过温中断使能。 0：禁止； 1：使能。	0x0
[1]	RW	tsensor_out_thresh_int_en	Tsensor 温度超门限范围中断使能。 0：禁止； 1：使能。	0x0
[0]	RW	tsensor_done_int_en	Tsensor 温度采集完毕中断使能。 0：禁止； 1：使能。	0x0

TSENSOR_TEMP_INT_CLR

TSENSOR_TEMP_INT_CLR 为 RF TEMP 配置寄存器。

Offset Address: 0x0524　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	-	reserved	保留。	0x0000
[0]	W1_PULSE	tsensor_int_clr	Tsensor 中断清除。 写 0：无效；	0x0

			写 1：清中断。	

TSENSOR_TEMP_INT_STS

TSENSOR_TEMP_INT_STS 为 RF TEMP 配置寄存器。

Offset Address: 0x0528　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:3]	-	reserved	保留。	0x0000
[2]	RO	tsensor_overtemp_int_sts	Tsensor 温度过温中断状态(备用)。 0：无效； 1：有效。	0x0
[1]	RO	tsensor_out_thresh_int_sts	Tsensor 温度超门限范围中断状态。 0：无效； 1：有效。	0x0
[0]	RO	tsensor_done_int_sts	Tsensor 温度采集完毕中断状态。 0：无效； 1：有效。	0x0

TSENSOR_OVER_TEMP_PD

TSENSOR_OVER_TEMP_PD 为 Tsensor 过温下电控制寄存器。

Offset Address: 0x0530　　Total Reset Value: 0x00FF

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	保留。	0x00
[10]	RW	tsensor_overtemp_pd_en	16 点平均单次上报模式或 16 点平均循环上报模式下过温下电保护使能。 0：禁止； 1：使能。	0x0
[9:8]	-	reserved	保留。	0x0
[7:0]	RW	tsensor_overtemp_pd_thresh	16 点平均单次上报模式或 16 点平均循环上报模式下过温下电保护阈值。 表示温度值的码字 TEMP_OUT 共有 8bit，换算为十进制值为 0～255，对应温度范围为-40℃～140℃。温度码输出呈现线性分布，即-40℃对应 0，140℃对应 255。	0xFF

			温度码换算公式：T(℃)= DEC(TEMP_OUT[7:0])×0.705+(-40)℃	

TSENSOR_AUTO_REFRESH_PERIOD

TSENSOR_AUTO_REFRESH_PERIOD 为 Tsensor 自动检测周期配置寄存器。

Offset Address: 0x0540　　Total Reset Value: 0xFFFF

Bits	Access	Name	Description	Reset
[15:0]	RW	tsensor_auto_refresh_period	Tsensor 自动检测周期(32K 时钟周期数)。	0xFFFF

TSENSOR_AUTO_REFRESH_CFG

TSENSOR_AUTO_REFRESH_CFG 为 Tsensor 自动检测使能控制寄存器。

Offset Address: 0x0544　　Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	RW	reserved	保留。	0x0000
[0]	RW	tsensor_auto_refresh_enable	16 点平均循环上报模式下周期检测使能。 0：禁止； 1：使能。	0x0

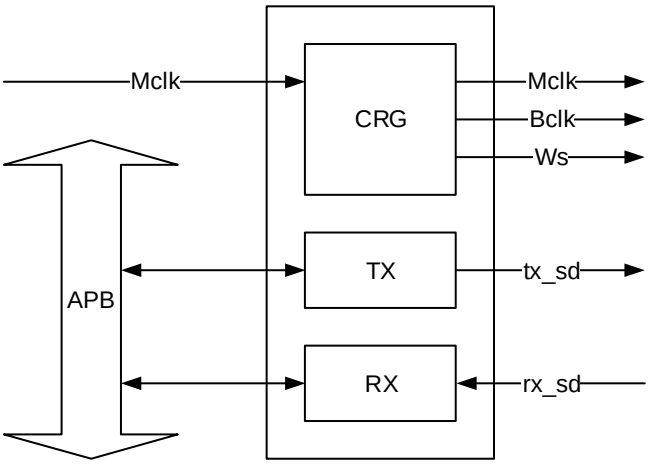
6.9 I2S

6.9.1 概述

I2S 模块是 APB 总线上的从设备、I2S 总线上的主设备。I2S 模块用于完成 CPU 对 I2S 总线上从设备的数据读写。

I2S 主时钟（Mclk）为 12.288MHz。

图6-18 I2S 功能框图



6.9.2 功能描述

I2S 模块具有以下功能特点：

- 采样位数 16bit 或 24bit。
- 采样率支持 8kHz、16kHz、32kHz、48kHz。

6.9.3 工作方式

6.9.3.1 接口信号

I2S 接口信号描述如表 6-13 所示。

表6-19 I2S 接口信号描述

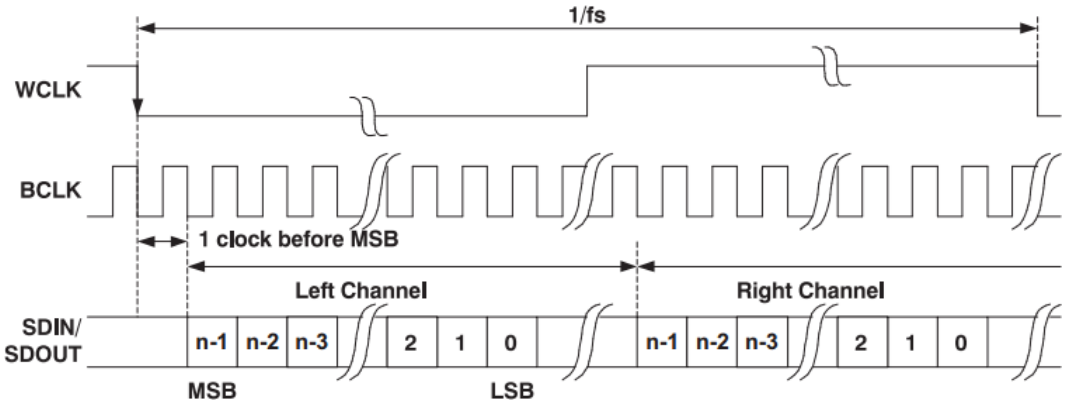
信号名	宽度（bit）	方向	功能描述
AUDIO_AIAO_BCLK	1	O	串行时钟 SCLK，也称位时钟（BCLK），即对应数字音频的每 1bit 数据，SCLK 都有 1 个脉冲。 SCK（Serial Clock singal）的频率=2×采样频率×采样位数。 由 Master 产生并控制。

信号名	宽度（bit）	方向	功能描述
AUDIO_AIAO_WS	1	O	左右声道选择信号 Word Select。 WS 为声道选择信号，即数据发送端所选择的声道： 0：选择左声道； 1：选择右声道。 WS 也称帧时钟，即 LRCLK（Left Right Clock）。WS 频率等于声音的采样率。由 Master 产生并控制。
AUDIO_AIAO_MCLK	1	O	Master Clock。在 I2S 接口的 ADC/DAC 系统中，除 SCK 和 WS 外，CODEC（Coder/Decoder）还需要控制器提供 MCLK（Master Clock）。由 Master 产生并控制，频率为 12.288MHz。
AUDIO_AIAO_TX_SD	1	O	输出数据。
AUDIO_AIAO_RX_SD	1	I	输入数据。

6.9.3.2 操作模式

I2S 操作模式如图 6-19 所示。

图6-19 I2S 操作模式



SD 是串行数据，在 I2S 中以二进制补码的形式在数据线上传输。在 WS 变化后的第 1 个 SCK 脉冲，先传输最高位（MSB）。左声道的数据 MSB 在 WS 下降沿之后第 2 个 SCK/BCLK 上升沿有效，右声道数据的 MSB 在 WS 上升沿之后第 2 个 SCK/BCLK 上升沿有效。

6.9.3.3 中断处理

I2S 有 6 个中断（独立中断源，可禁止，高电平有效）：

- TX_UNFULL_INT
  发送 FIFO 不满中断。当发送 FIFO 的有效数据＜16 个时，该中断置位。
- TX_LEVEL_INT
  发送 FIFO 中断请求。当发送 FIFO 中的有效数据＜配置水线时，该中断置位。
- RX_UNEMPTY_INT
  接收 FIFO 非空中断。当接收 FIFO 的有效数据＞1 个时，该中断置位。
- RX_LEVEL_INT
  接收 FIFO 中断请求。当接收 FIFO 的有效数据＞配置水线时，该中断置位。
- TX_RERROR_INT
  发送 FIFO 下溢出中断。当发送 FIFO 出现下溢出时，该中断置位。
- RX_WERROR_INT
  接收 FIFO 上溢出中断。当接收 FIFO 出现上溢出时，该中断置位。

6.9.3.4 应用说明

以中断或查询方式下的数据传输为例。

初始化

初始化的步骤如下：

1. 写 AUDIO_AIAO_CTRL[rx_en]、AUDIO_AIAO_CTRL[tx_en]为 0，禁止 I2S。
2. 写 AUDIO_AIAO_CTRL，设置采样精度、FIFO 水线、FSCLK（Sampling Frequency Clock）分配倍数、BCLK 分频倍数等参数。
3. 中断方式下，写 AUDIO_AIAO_CTRL，使能相应中断信号；查询方式下，应禁止产生相应中断信号。

   ----结束

数据发送

发送定长数据（以 4 个数据为例）的步骤如下：

1. 写 AUDIO_AIAO_CTRL[tx_en]为 1，使能 I2S。
2. 读 AUDIO_AIAO_INT，查看 FIFO 水线状态。
   - 如果为 1，则发送 FIFO 未达到水线，写 4 个数据到 AUDIO_AIAO_DMA_WR。
   - 如果为 0，则发送 FIFO 达到水线，等待前面的数据发送，直至 FIFO 未达到水线后才可输入。
3. 重复步骤 2，直至完全发送。
4. 写 AUDIO_AIAO_CTRL[tx_en]为 0，禁止 I2S。

----结束

数据接收

接收定长数据的步骤如下：

1. 写 AUDIO_AIAO_CTRL[rx_en]为 1，使能 I2S。
2. 读 AUDIO_AIAO_INT，查看接收 FIFO 水线状态。如果为接收 FIFO 达到水线，读 AUDIO_AIAO_DMA_RD，接收 4 个数据；否则，查看接收 FIFO 非空，如果接收 FIFO 非空，接收 1 个数据。否则，状态继续执行。
3. 重复步骤 2，直至完全接收。
4. 写 AUDIO_AIAO_CTRL[rx_en]为 0，禁止 I2S。

----结束

6.9.3.5 性能分析

采样精度为 16bit

I2S 内部的 FIFO 大小为 16×32bit。采样精度为 16bit 时，FIFO 最大可以缓存 16 个采样点的左右声道数据。

采样率为 8kHz 时，最大可以缓存时间长度为 2ms 的声音数据。

采样率为 16kHz 时，最大可以缓存时间长度为 1ms 的声音数据。

采样率为 32kHz 时，最大可以缓存时间长度为 0.5ms 的声音数据。

采样率为 48kHz 时，最大可以缓存时间长度为 0.33ms 的声音数据。

采样精度为 24bit

I2S 内部的 FIFO 大小为 16×32bit。采样精度为 24bit 时，FIFO 最大可以缓存 8 个采样点的左右声道数据。

采样率为 8kHz 时，最大可以缓存时间长度为 1ms 的声音数据。

采样率为 16kHz 时，最大可以缓存时间长度为 0.5ms 的声音数据。

采样率为 32kHz 时，最大可以缓存时间长度为 0.25ms 的声音数据。

采样率为 48kHz 时，最大可以缓存时间长度为 0.16ms 的声音数据。

6.9.4 寄存器概览

I2S 寄存器概览如表 6-20 所示。

表6-20 I2S 寄存器概览（基址是 0xF8CC_0000）

偏移地址	名称	描述	页码
0x0000	AUDIO_AIAO_CTRL	控制寄存器	6-119
0x0004	AUDIO_AIAO_RX_FIFO_INT_CLR	RX_FIFO 中断清除寄存器	6-122
0x0008	AUDIO_AIAO_TX_FIFO_INT_CLR	ATX_FIFO 中断清除寄存器	6-122
0x000C	AUDIO_AIAO_BCLK_CNT	BCLK_CNT 寄存器	6-122
0x0010	AUDIO_AIAO_INT	中断寄存器	6-122
0x0014	AUDIO_AIAO_DMA_WR	DMA 写地址寄存器	6-124
0x0018	AUDIO_AIAO_DMA_RD	DMA 读地址寄存器	6-124

6.9.5 寄存器描述

AUDIO_AIAO_CTRL

AUDIO_AIAO_CTRL 为控制寄存器。

Offset Address: 0x0000　　Total Reset Value: 0x0070_9000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RW	tx_rerror_int_mask	TX FIFO 下溢出中断使能位。 0：禁止中断； 1：使能中断。	0x0
[24]	RW	rx_werror_int_mask	RX FIFO 上溢出中断使能位。 0：禁止中断； 1：使能中断。	0x0
[23:21]	RW	aiao_fsclk_div	BCLK 与 WS 的频率比。 计算公式：BCLK=2×WS×采样位数。 I2S 支持 16bit 与 24bit 的采样位数，默认 BCLK 与 WS 的频率比例为 64。 WS=MCLK/(FSCLK_DIV×BCLK_DIV)。 000：16； 001：32； 010：48； 011：64；	0x3

			100：128； 101：256； 其他：8。	
[20:17]	RW	aiao_bclk_div	MCLK 与 BCLK 的频率比。 0x0：1； 0x1：3； 0x2：2； 0x3：4； 0x4：6； 0x5：8； 0x6：12； 0x7：16； 0x8：24； 0x9：32； 0xA：48； 0xB：64； 0xC：96； 其他：保留。	0x8
[16:14]	RW	dma_tx_level	TX FIFO 水线。 000：2； 001：4； 010：8； 011：12； 100：14； 其他：保留。	0x2
[13:11]	RW	dma_rx_level	RX FIFO 水线。 000：2； 001：4； 010：8； 011：12； 100：14； 其他：保留。	0x2
[10]	RW	aiao_ck_gt_en	低功耗使能。 0：时钟禁止，AIAO 内部无时钟； 1：时钟使能，AIAO 内部时钟正常产生。	0x0
[9]	RW	tx_level_int_mask	TX 水线中断使能位。	0x0

			0：禁止中断； 1：使能中断。	
[8]	RW	tx_unfull_int_mask	TX 非满中断使能位。 0：禁止中断； 1：使能中断。	0x0
[7]	RW	rx_level_int_mask	RX 水线中断使能位。 0：禁止中断； 1：使能中断。	0x0
[6]	RW	rx_unempty_int_mask	RX 非空中断使能位。 0：禁止中断； 1：使能中断。	0x0
[5]	RW	dma_tx_en	DMA TX 使能。 0：禁止； 1：使能。	0x0
[4]	RW	dma_rx_en	DMA RX 使能。 0：禁止； 1：使能。	0x0
[3]	RW	bclk_inv_en	输出 BCLK 反相使能。 0：输出 BCLK 相位不改变； 1：输出 BCLK 相位取反。 协议规定数据发送既可以同步于 SCK 的上升沿，也可以是下降沿，但接收设备在 SCK 的上升沿采样。 如果外设在 SCK 的下降沿采样接收数据，则需要配置[bclk_inv_en]为 1，使外设的 BCLK 与 I2S 的 BCLK 为反相关系。	0x0
[2]	RW	tx_en	TX 使能。 0：禁止； 1：使能。	0x0
[1]	RW	rx_en	RX 使能。 0：禁止； 1：使能。	0x0
[0]	RW	data_bit	采样精度。 0：16bit； 1：24bit。	0x0

AUDIO_AIAO_RX_FIFO_INT_CLR

AUDIO_AIAO_RX_FIFO_INT_CLR 为 RX_FIFO 中断清除寄存器。

Offset Address: 0x0004　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	rx_fifo_int_clr	RX FIFO 上溢出中断清除位。 0：无作用； 1：清除中断。	0x0

AUDIO_AIAO_TX_FIFO_INT_CLR

AUDIO_AIAO_TX_FIFO_INT_CLR 为 ATX_FIFO 中断清除寄存器。

Offset Address: 0x0008　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	tx_fifo_int_clr	TX FIFO 下溢出中断清除位。 0：无作用； 1：清除中断。	0x0

AUDIO_AIAO_BCLK_CNT

AUDIO_AIAO_BCLK_CNT 为 BCLK_CNT 寄存器。

Offset Address: 0x000C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	bclk_cnt	Debug 寄存器，用于观测 BCLK 时钟是否正常。	0x00000000

AUDIO_AIAO_INT

AUDIO_AIAO_INT 为中断寄存器。

Offset Address: 0x0010　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11]	RO	tx_rerror_int	TX FIFO 下溢出中断源标志位(带 MASK)。 0：无中断； 1：有中断。	0x0
[10]	RO	tx_rerror_int_unmask	TX FIFO 下溢出中断源标志位(未带 MASK)。 0：无中断； 1：有中断。	0x0
[9]	RO	tx_level_int	TX FIFO 水线中断源标志位(带 MASK)。 0：无中断； 1：有中断。	0x0
[8]	RO	tx_level_int_unmask	TX FIFO 水线中断源标志位(未带 MASK)。 0：无中断； 1：有中断。	0x0
[7]	RO	tx_unfull_int	TX FIFO 非满中断源标志位(带 MASK)。 0：无中断； 1：有中断。	0x0
[6]	RO	tx_unfull_int_unmask	TX FIFO 非满中断源标志位(未带 MASK)。 0：无中断； 1：有中断。	0x0
[5]	RO	rx_werror_int	RX FIFO 上溢出中断源标志位(带 MASK)。 0：无中断； 1：有中断。	0x0
[4]	RO	rx_werror_int_unmask	RX FIFO 上溢出中断源标志位(未带 MASK)。 0：无中断； 1：有中断。	0x0
[3]	RO	rx_level_int	RX FIFO 水线中断源标志位(带 MASK)。 0：无中断； 1：有中断。	0x0
[2]	RO	rx_level_int_unmask	RX FIFO 水线中断源标志位(未带	0x0

			MASK)。 0：无中断； 1：有中断。	
[1]	RO	rx_unempty_int	RX FIFO 非空中断源标志位(带 MASK)。 0：无中断； 1：有中断。	0x0
[0]	RO	rx_unempty_int_unmask	RX FIFO 非空中断源标志位(未带 MASK)。 0：无中断； 1：有中断。	0x0

AUDIO_AIAO_DMA_WR

AUDIO_AIAO_DMA_WR 为 DMA 写地址寄存器。

Offset Address: 0x0014　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_wr_addr	DMA 通过该地址往 AIAO 写数据。	0x00000000

AUDIO_AIAO_DMA_RD

AUDIO_AIAO_DMA_RD 为 DMA 读地址寄存器。

Offset Address: 0x0018　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dma_rd_addr	DMA 通过该地址从 AIAO 读数据。	0x00000000

6.10 SDIO

6.10.1 概述

SDIO 从机控制器集成了符合工业标准 SDIO3.0 规格的 SD（Secure Digital）设备接口，并允许主机控制器使用 SDIO 总线协议访问 SoC 设备。主机可直接访问 SDIO 接口的寄存器，并通过使用 DMA 访问芯片内存，无需处理器处理。

6.10.2 功能描述

SDIO 具有以下功能特点：

- 支持 SDIO3.0，仅支持 Slave 模式。
- SDIO 支持访问芯片所有的 Memory 和寄存器。
- 支持 Clock 门控。
- 最高支持 50MHz 时钟，4bit 模式下速率最高支持 200Mbit/s。
- 支持 SDIO 通路上行、下行、回环测试。
- 支持高速模式和 SDR25（Single Date Rate）模式。
- 支持 SDIO 频率动态调整。
- 支持深睡时 Host 侧下电，而 Device 侧 SDIO 不下电。
- SDIO 内部 DMA 支持 SDIO 到内部 RAM 的 DMA 搬移。

6.11 DMA

6.11.1 概述

直接存储器访问（DMA）方式是一种完全由硬件执行 I/O 交换的工作方式。在此方式中，直接存储器访问控制器（DMAC）直接在存储器和外设、外设和外设、存储器和存储器之间进行数据传输，减少处理器的干涉和开销。

DMA 方式一般用于高速传输成组的数据。DMAC 在收到 DMA 传输请求后，根据 CPU 对通道的配置启动总线主控制器，向存储器和外设发出地址和控制信号，对传输数据的个数计数，并且以中断方式向 CPU 报告传输操作结束或错误。

6.11.2 功能描述

DMA 具有以下功能特点：

- 支持 8bit、16bit、32bit 数据位宽方式传输。
- 提供 4 个 DMA 通道，每个通道可配置用于一种单向传输。
- DMA 通道优先级固定，优先级从高到低对应的通道号依次为 0～3。当来自 2 个外设的 DMA 请求同时有效时，优先级高的通道先开始传输。
- DMAC 通道 0～通道 3 中各包含 1 个 4×32bit 的 FIFO。
- 提供 2 个总线宽度为 32bit 的 Master 总线接口用于数据传输。

  注意：芯片仅支持 Master0 用于数据传输。
- 外设可使用单次传输（Single）和连续传输（Burst）2 种 DMA 请求。
- 提供 16 组 DMA 请求输入，可通过配置作为通道的源端请求或目的端请求。
- 支持软件控制的 DMA 请求。
- 支持通过编程决定 DMA Burst 长度。
- 源地址和目的地址可分别配置为在 DMA 传输过程中自动递增或不递增。
- 支持 4 种数据传输方向：

- 存储器至外设
- 存储器至存储器
- 外设至存储器
- 外设至外设

注意：芯片不支持外设至外设的传输场景。

- 支持链表 DMA 传输。
- 支持 DMAC 流控。
- 提供 1 个可屏蔽中断输出，支持 DMA 错误和 DMA 传输完成中断屏蔽前后状态查询，及两者的组合中断状态查询。
- 支持 DMAC 使能禁止，用于功耗控制，支持 DMAC 时钟门控。

6.11.3 工作方式

DMA 初始化配置步骤如下：

1. 读 DMAC_ENBLD_CHNS，获取空闲的通道编号 ch_num。
2. 写 DMAC_Cn_CONFIG 为 0，关闭 ch_num 通道并清空相应的配置。
3. 如果需要通过软请求方式进行 DMA 搬数，则需要根据表 6-21 中的外设编号配置 DMAC_SOFT_BREQ、DMAC_SOFT_SREQ。

   注意：如果通过硬请求方式进行 DMA 搬数或传输方向为存储器到存储器，则忽略此步骤。

表6-21 DMA 接口信号描述

外设编号	外设端口	功能描述
0	Uart0_rx	UART0 的接收信号。
1	Uart0_tx	UART0 的发送信号。
2	Uart1_rx	UART1 的接收信号。
3	Uart1_tx	UART1 的发送信号。
4	Uart2_rx	UART2 的接收信号。
5	Uart2_tx	UART2 的发送信号。
6	Spi0_rx	SPI0 的接收信号。
7	Spi0_tx	SPIT0 的发送信号。
8	Spi1_rx	SPI1 的接收信号。
9	Spi1_tx	SPIT1 的发送信号。
10	I2s_rx	I2S 的接收信号。
11	I2s_tx	I2S 的发送信号。

外设编号	外设端口	功能描述
12～15	-	保留。

4. 根据通道编号 ch_num 配置对应通道的源地址 DMAC_Cn_SRC_ADDR、目的地址 DMAC_Cn_DEST_ADDR。
5. 写 DMAC_Cn_CONTROL，根据具体需求配置 DMA 通道控制信息（例如：传输长度、Burst 长度、传输位宽等）。
6. 如果需要进行链表传输，则配置 DMAC_CnLLI。
7. 根据需求配置 DMAC_Cn_CONFIG：
   - 写[flow_cntrl]，配置流控和传输类型。
   - 写[dest_peripheral]，配置目的设备，配置值为表 6-21 中的外设编号。
   - 写[src_peripheral]，配置源设备，配置值为表 6-21 中的外设编号。
   - 写[itc]，配置中断屏蔽位。
   - 写[ie]，配置中断屏蔽位。
   - 写[e]，配置通道使能位。
8. 如果中断屏蔽位 DMAC_Cn_CONFIG[itc]未进行屏蔽，则当 DMAC_Cn_CONTROL[transfersize]个数据被传输完成后上报完成中断，或轮询读取 DMAC_RAW_INT_TC_STATUS 查询完成状态。

----结束

6.11.4 寄存器概览

DMAC 寄存器概览如表 6-22 所示。

表6-22 DMAC 寄存器概览（基址是 0x4020_0000）

偏移地址	名称	描述	页码
0x000	DMAC_INT_STAT	中断状态寄存器	6-128
0x004	DMAC_INT_TC_STAT	DMAC 传输完成中断状态寄存器	6-129
0x008	DMAC_INT_TC_CLR	传输结束状态清除寄存器	6-129
0x00C	DMAC_INT_ERR_STAT	错误中断状态寄存器	6-130
0x010	DMAC_INT_ERR_CLR	错误中断清除寄存器	6-130
0x014	DMAC_RAW_INT_TC_STATUS	原始传输完成中断状态寄存器	6-130
0x018	DMAC_RAW_INT_ERR_STATUS	原始传输错误中断状态寄存器	6-131

偏移地址	名称	描述	页码
0x01C	DMAC_ENBLD_CHNS	通道使能寄存器	6-131
0x020	DMAC_SOFT_BREQ	软件 Burst 请求寄存器	6-132
0x024	DMAC_SOFT_SREQ	软件 Single 请求寄存器	6-132
0x028	DMAC_SOFT_LBREQ	软件最后一个 Burst 请求寄存器	6-133
0x02C	DMAC_SOFT_LSREQ	软件最后一个 Single 请求寄存器	6-133
0x030	DMAC_CONFIG	配置 DMAC 操作寄存器	6-134
0x034	DMAC_SYNC	同步寄存器	6-134
0x100＋n×0x20	DMAC_Cn_SRC_ADDR	DMA 源地址寄存器	6-134
0x104＋n×0x20	DMAC_Cn_DEST_ADDR	DMA 目的地址寄存器	6-135
0x108＋n×0x20	DMAC_CnLLI	链表寄存器	6-135
0x10C＋n×0x20	DMAC_Cn_CONTROL	通道控制寄存器	6-136
0x110＋n×0x20	DMAC_Cn_CONFIG	通道配置寄存器	6-140

DMAC 寄存器偏移地址中变量的取值范围和含义如表 6-23 所示。

表6-23 DMAC 寄存器偏移地址变量表

变量名称	取值范围	描述
n	0～3	DMA 通道编号。

6.11.5 寄存器描述

DMAC_INT_STAT

DMAC_INT_STAT 为中断状态寄存器。给出经过屏蔽后的中断状态。

说明

如果 DMAC_INT_TC_STAT 和 DMAC_INT_ERR_STAT 的相应位同时被屏蔽，则该寄存器的对应位被屏蔽。

Offset Address: 0x000　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RO	int_stat	DMA 各通道经屏蔽后的中断状态。 bit[0]～bit[3]分别对应 DMAC 的通道 0～3。 每 bit 对应的取值含义如下： 0：未产生中断； 1：已产生中断(该中断请求可能来自该通道的错误中断或传输完成中断)。	0x0

DMAC_INT_TC_STAT

DMAC_INT_TC_STAT 为 DMAC 传输完成中断状态寄存器。给出经过屏蔽后的传输完成中断状态，对应的屏蔽位为 DMAC_Cn_CONFIG[itc]（其中 n 表示通道号 0～3）。

说明

该寄存器必须与 DMAC_INT_STAT 结合在一起使用，根据 DMAC_INT_STAT[int_stat]的状态进行中断源查询。

Offset Address: 0x004　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RO	int_tc_stat	经过屏蔽后的传输完成中断状态。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下： 0：未产生传输完成中断； 1：已产生传输完成中断。	0x0

DMAC_INT_TC_CLR

DMAC_INT_TC_CLR 为传输结束状态清除寄存器。用于清除传输完成中断。

Offset Address: 0x008　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	WO	int_tc_clr	清除传输完成中断。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下：	0x0

			0：不清除该中断； 1：清除该中断。	

DMAC_INT_ERR_STAT

DMAC_INT_ERR_STAT 为错误中断状态寄存器。给出经过屏蔽后的错误中断状态，对应的屏蔽位为 DMAC_Cn_CONFIG[ie]。

Offset Address: 0x00C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RO	int_err_stat	经过屏蔽后的错误中断状态。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下： 0：未产生错误中断； 1：已产生错误中断。	0x0

DMAC_INT_ERR_CLR

DMAC_INT_ERR_CLR 为错误中断清除寄存器。用于清除出错中断。

Offset Address: 0x010　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	WO	int_err_clr	清除出错中断。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下： 0：不清除该中断； 1：清除该中断。	0x0

DMAC_RAW_INT_TC_STATUS

DMAC_RAW_INT_TC_STATUS 为原始传输完成中断状态寄存器。给出各通道屏蔽前的传输完成中断状态。

Offset Address: 0x014　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RO	raw_int_tc_stat	原始传输完成中断状态。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下： 0：未产生传输完成中断； 1：已产生传输完成中断。	0x0

DMAC_RAW_INT_ERR_STATUS

DMAC_RAW_INT_ERR_STATUS 为原始传输错误中断状态寄存器。给出各通道屏蔽前的错误中断状态。

Offset Address: 0x018　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RO	raw_int_err_stat	各通道屏蔽前的错误中断状态。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下： 0：未产生错误中断； 1：已产生错误中断。	0x0

DMAC_ENBLD_CHNS

DMAC_ENBLD_CHNS 为通道使能寄存器。

说明

使能某个通道，由该通道的通道寄存器 DMAC_Cn_CONFIG 的使能位决定。当某个通道的 DMA 传输结束时，DMAC_ENBLD_CHNS 中与该通道对应的位被清零。

Offset Address: 0x01C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RO	enabled_channels	通道使能状态。 bit[0]～bit[3]分别对应 DMAC 通道 0～3。 每 bit 对应的取值含义如下： 0：禁止；	0x0

			1：使能。	

DMAC_SOFT_BREQ

DMAC_SOFT_BREQ 为软件 Burst 请求寄存器。用于供软件控制产生 DMA Burst 请求。

读该寄存器，可得知当前正在请求 DMA Burst 传输的设备。外设和该寄存器都可以产生 1 个 DMA 请求。

说明

建议勿同时使用软件 DMA 请求和硬件 DMA 请求。

Offset Address: 0x020　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	soft_breq	软件控制产生 DMA Burst 传输请求。 bit[0]～bit[15]与接口信号的对应关系请参见“DMA 接口信号描述”。 当写该寄存器时： 0：无影响； 1：产生 DMA Burst 传输请求，当传输结束时该寄存器中的相应位被清零。 当读该寄存器时： 0：与请求线 DMACBREQ bit[15:0]对应的外设未发出 DMA Burst 请求； 1：与请求线 DMACBREQ bit[15:0]对应的外设正在请求 DMA Burst 传输。	0x0000

DMAC_SOFT_SREQ

DMAC_SOFT_SREQ 为软件 Single 请求寄存器。用于供软件控制产生 DMA 单次传输请求。例如：读该寄存器，可得知当前正在请求 DMA 单次传输的设备。通过 DMAC 的 16 个 DMA 请求输入信号和该寄存器都可以产生 1 个 DMA 请求。

说明

建议勿同时使用软件 DMA 请求和硬件 DMA 请求。

Offset Address: 0x024　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000

[15:0]	RW	soft_sreq	软件控制产生 DMA Single 传输请求。 当写该寄存器时： 0：无影响； 1：产生 DMA Single 传输请求，当传输结束时该寄存器中的相应位被清零。 当读该寄存器时： 0：与请求线 DMACBREQ bit[15:0]对应的外设未发出 DMA Single 请求； 1：与请求线 DMACBREQ bit[15:0]对应的外设正在请求 DMA Single 传输。	0x0000

DMAC_SOFT_LBREQ

DMAC_SOFT_LBREQ 为软件最后一个 Burst 请求寄存器。用于供软件控制产生 DMA Last Burst 传输请求。

Offset Address: 0x028　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	WO	soft_lbreq	由软件发起 Last Burst 请求。 0：无影响； 1：产生 DMA Last Burst 传输请求，当传输结束时该寄存器中的相应位被清零。	0x0000

DMAC_SOFT_LSREQ

DMAC_SOFT_LSREQ 为软件最后一个 Single 请求寄存器。用于供软件控制产生 DMA Last Burst 传输请求。

Offset Address: 0x02C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	WO	soft_lsreq	由软件发起 Last Burst 传输请求。 0：无影响； 1：产生一个 DMA Last Burst 传输请求，当传输结束时该寄存器中的相应位被清零。	0x0000

DMAC_CONFIG

DMAC_CONFIG 为配置 DMAC 操作寄存器。通过写 DMAC_CONFIG[m1]和[m2]，可改变 DMAC 的 2 个 Master 接口的 Endianness。复位时，DMAC 的 2 个 Master 接口被设为 Little Endian 模式。

说明

2 个 Master 均采用 Little Endian。

Offset Address: 0x030　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	m2	Master 2 Endianness 配置位。 0：Little Endian 模式； 1：Big Endian 模式。	0x0
[1]	RW	m1	Master 1 Endianness 配置位。 0：Little Endian 模式； 1：Big Endian 模式。	0x0
[0]	RW	e	DMAC 使能。 0：禁止； 1：使能。	0x0

DMAC_SYNC

DMAC_SYNC 为同步寄存器。

Offset Address: 0x034　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	dmac_sync	控制是否需要对请求线进行同步。 0：使能对应外设的 DMA 请求信号同步逻辑； 1：禁止对应外设的 DMA 请求信号同步逻辑。	0x0000

DMAC_Cn_SRC_ADDR

DMAC_Cn_SRC_ADDR 为 DMA 源地址寄存器。给出当前待传数据的源地址（字节排序）。

每个寄存器在对应的通道被启动前都需由软件对其直接编程。当通道被启动后，该寄存器在下列情况下更新：

- 当源地址递增时。
- 当传完一个完整的数据块后，从链表结点中载入时。

当该通道处于活动状态时，读该寄存器无法获得有效信息（由于当软件得到读出的寄存器值时，该寄存器的值已经随着通道传输改变），在通道停止传输时读该寄存器，此时读取值显示 DMAC 读最后一项时的源地址。

源地址和目的地址必须与源设备和目的设备的传输宽度对齐。

Offset Address: 0x100＋n×0x20　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	src_addr	DMA 源地址。	0x00000000

DMAC_Cn_DEST_ADDR

DMAC_Cn_DEST_ADDR 为 DMA 目的地址寄存器。包含了当前待传数据的目的地址（字节排序）。

每个寄存器在对应的通道被启动前都需由软件对其直接编程。当通道被启动后，该寄存器在下列情况下更新：

- 当目的地址递增时。
- 当传完一个完整的数据块后，从链表结点中载入时。

当该通道处于活动状态时，读该寄存器无法获得有效信息（由于当软件得到读出的寄存器值时，该寄存器的值已经随着通道传输改变），在通道停止传输时读该寄存器，此时读取值显示 DMAC 写最后一项时的目的地址。

Offset Address: 0x104＋n×0x20　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dest_addr	DMA 目的地址。	0x00000000

DMAC_CnLLI

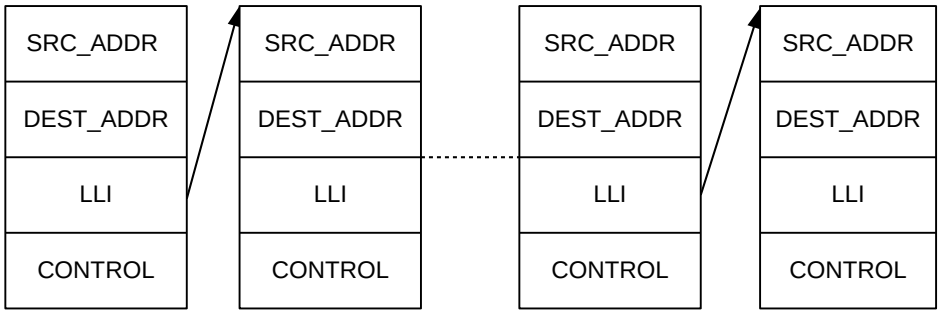
DMAC_CnLLI 为链表寄存器。

DMAC 的链表结点数据结构：

- DMAC_Cn_SRC_ADDR 设置源设备首地址。
- DMAC_Cn_DEST_ADDR 设置目的设备首地址。
- DMAC_CnLLI 设置下一个结点的地址。

- DMAC_Cn_CONTROL 设置访问源/目的设备所采用的 Master、源/目的设备的位宽、Burst Size、地址递增以及 Transfer Size 等参数。

表6-24 DMAC 链表结构示例



须知

DMAC_CnLLI[lli]不可指定 1 个大于 0xFFFF_FFF0 的数；否则，1 个 4 Word 的 Burst 传输将使地址回卷到 0x00000000 处，导致链表结点数据结构不能存储在连续的地址区域中。

如果[lli]值为 0，表示当前结点是链表的链尾，则当本结点对应的数据块全部传完后，该通道会被关闭。

Offset Address: 0x108＋n×0x20　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	RW	lli	下一个链表结点地址 bit[31:2](地址 bit[1:0]为 0)。 **注意：要求链表地址 4byte 对齐。**	0x00000000
[1]	RW	reserved	保留。 **注意：写入时必须写 0，读出时应被屏蔽。**	0x0
[0]	RW	lm	载入下一个链表结点的 Master。 0：Master1； 1：Master2。	0x0

DMAC_Cn_CONTROL

DMAC_Cn_CONTROL 为通道控制寄存器。包含了 DMA 通道控制信息（例如：传输长度、Burst 长度、传输位宽等）。

每个寄存器在对应的通道被启动前都需由软件对其直接编程。当通道被启动后，该寄存器的值在传完 1 个完整的数据块后，从链表结点载入时更新。

当该通道处于活动状态时，读该寄存器无法获得有效信息（由于当软件得到读出的寄存器值时，该寄存器的值已经随着通道传输改变），在通道停止传输时，可进行该寄存器的读操作。

Offset Address: 0x10C＋n×0x20　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	i	当前链表结点触发传输完成中断使能位。 0：不触发； 1：触发。	0x0
[30:28]	RW	prot	Master 发出的访问保护 HPROT bit[2:0]信号。	0x0
[27]	RW	di	目的地址递增。 0：不递增； 1：每传一个数递增一次。 **注意：目的设备为外设时，目的地址不递增；目的设备为存储器时，目的地址递增。**	0x0
[26]	RW	si	源地址递增。 0：不递增； 1：每传一个数递增一次。 **注意：源设备为外设时，源地址不递增；源设备为存储器时，源地址递增。**	0x0
[25]	RW	d	设置访问目的设备的 Master。 0：使用 Master1 访问； 1：使用 Master2 访问。	0x0
[24]	RW	s	设置访问源设备的 Master。 0：使用 Master1 访问； 1：使用 Master2 访问。	0x0
[23:21]	RW	dwidth	目的设备传输位宽。 宽于 Master 位宽的传输位宽是非法的。 目的设备和源设备的位宽可以不同，硬件自动对数据进行 Pack 和 Unpack。 DWidth 的值和具体的位宽对应关系请参见“DWidth 和 SWidth 的值与其对应传输位宽”。	0x0
[20:18]	RW	swidth	源设备传输位宽。 宽于 Master 位宽的传输位宽是非法的。 目的设备和源设备的位宽可以不同，硬件	0x0

			自动对数据进行 Pack 和 Unpack。 SWidth 的值和具体的位宽对应关系请参见“DWidth 和 SWidth 的值与其对应传输位宽”。	
[17:15]	RW	dbsize	目的设备 Burst 长度(1 次目的设备 Burst 传输所需传输的数据个数，即当 DMACCnBREQ 有效时，传输的数据个数)。 该值必须设置为目的设备支持的 Burst 大小，或如果目的设备为存储器时，则设置为到存储地址边界的存储区域大小。 DBSize 的值和具体的传输长度的对应关系请参见“DBSize 及 SBSize 的值与其对应的 burst 长度”。	0x0
[14:12]	RW	sbsize	源设备 Burst 长度(1 次源设备 Burst 传输所需传输的数据个数，即当 DMACCnBREQ 有效时，传输的数据个数)。 该值必须设置为源设备支持的 Burst 大小，或如果源设备为存储器时，则设置为到存储地址边界的存储区域大小。 SBSize 的值和具体的传输长度的对应关系请参见“DBSize 及 SBSize 的值与其对应的 burst 长度”。	0x0
[11:0]	RW	transfersize	写：当 DMAC 是流控制器时，该值表示源设备待传数据的个数。 读：获取与目的设备相连的总线上已传出的数据个数。 当该通道处于活动状态时，读该寄存器无法获得有效信息(由于当软件得到读出的寄存器值时，该寄存器的值已经随着通道传输改变)，因此一般情况下，在通道被启动后再停止传输时，对该寄存器进行读操作。	0x000

DMAC_Cn_CONTROL 寄存器的 DBSize 及 SBSize 的值与其对应的 Burst 长度如表 6-25 所示。

表6-25 DBSize 及 SBSize 的值与其对应的 Burst 长度

DBSize 或 SBSize 的值	Burst 长度
000	1
001	4

DBSize 或 SBSize 的值	Burst 长度
010	8
011	16
100	32
101	64
110	128
111	256

DMAC_Cn_CONTROL 寄存器的 DWidth 和 SWidth 的值与其对应传输位宽如表 6-26 所示。

表6-26 DWidth 和 SWidth 的值与其对应传输位宽

SWidth 或 DWidth 的值	传输位宽
000	Byte（8bit）
001	Halfword（16bit）
010	Word（32bit）
011	reserved
100	reserved
101	reserved
110	reserved
111	reserved

配置寄存器 DMAC_Cn_CONTROL 时需注意：

- 当源设备的传输宽度＜目的设备传输宽度时，源设备的传输宽度与 transfer size 的乘积应为目的设备传输宽度的整数倍，否则 FIFO 中的数据将会滞留并丢失。
- SWidth 和 DWidth 字段不能设置为未定义的位宽。
- transfer size 字段如果被写为 0 且 DMAC 又是流控制器，则 DMAC 将不会发生任何传输动作。编程者应负责关闭此 DMA 通道并对此通道重新编程。
- 不应对 DMAC_Cn_CONTROL 寄存器进行普通的写入/读出测试。由于 transfer size 字段不是一个普通的可写入并读回相同值的寄存器字段。当写入时，该字段如一个控制寄存器，因为其决定了 DMAC 应传输多少个数据；当读回时，该字段则相当于一个状态寄存器，因为其返回剩下的待传输数据个数（以源设备位宽为单位）。

- 当 transfer size 字段的设置值＞源设备或目的设备中的 FIFO 的深度（为外设的 FIFO，非 DMAC 的 FIFO），则 DMAC 的源地址或目的地址必须被设为不递增模式，否则有可能导致外设的 FIFO 溢出。

总线访问信息在传输发生时由 master 接口信号提供给源设备或目的设备。这些访问信息是通过对通道寄存器编程设定的 DMAC_Cn_CONTROL[prot]和 DMAC_Cn_CONFIG[l]。使用 prot 的 3 个保护位的含义如表 6-27 所示。

表6-27 DMAC_Cn_CONTROL 寄存器 prot 段属性及定义

位域	描述	目的
[2]	Cacheable or noncacheable	指明访问是可 cache 还是不可 cache。 0：不可 cache； 1：可 cache。 例如：该位可用于告知 1 个 AMBA 桥：当其发现 8 个数的 burst 读的第 1 个读操作时，该桥可在目标总线上直接发起一个 8 个数的 burst 读，而不用将源总线上的读操作 1 次 1 个的传到目标总线。 该位控制总线信号 HPROT[3]的输出。
[1]	Bufferable or nonbufferable	指明访问是可缓冲还是不可缓冲。 0：不可缓冲； 1：可缓冲。 例如：该位可用于告知一个 AMBA 桥在源端总线上写操作可以以零等待状态完成，而无需等该桥把操作仲裁到目的总线上，也无需等 slave 接收完数据。 该位控制总线信号 HPROT[2]的输出。
[0]	Privileged or User	指明访问是用户模式还是特权模式。 0：用户模式； 1：特权模式。 该位控制总线信号 HPROT[1]的输出。

DMAC_Cn_CONFIG

DMAC_Cn_CONFIG 为通道配置寄存器。

说明

该寄存器在新的链表结点被载入时不会被更新。

Offset Address: 0x110＋n×0x20　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。 **注意：写入时必须写 0，读出时应被屏蔽。**	0x0000
[18]	RW	h	Halt 暂停位。 0：允许 DMA 请求； 1：忽略后续的 DMA 请求，通道 FIFO 中的内容均被传输完。 该 bit 可以与[a]、[e_c]共同用于无数据丢失地关闭一个 DMA 通道。	0x0
[17]	RO	a	Active 有效位。 0：通道 FIFO 中无数据； 1：通道 FIFO 中有数据。 该 bit 可以与[h]、[e_c]共同用于无数据丢失地关闭一个 DMA 通道。	0x0
[16]	RW	l	Lock 锁定位。 0：禁止总线上 Lock 锁定传输； 1：使能总线上 Lock 锁定传输。	0x0
[15]	RW	itc	本通道的传输完成中断屏蔽位。 0：屏蔽； 1：不屏蔽。	0x0
[14]	RW	ie	本通道的错误中断屏蔽位。 0：屏蔽； 1：不屏蔽。	0x0
[13:11]	RW	flow_cntrl	流控及传输类型。 流控制器：可以为 DMAC、源设备和目的设备。 传输类型：可以为存储器到外设、外设到存储器、外设到外设、存储器到存储器。 流控类型和传输类型请参见“流控制器及传输类型位定义”。	0x0
[10]	-	reserved	保留。 **注意：写入时必须写 0，读出时应被屏蔽。**	0x0
[9:6]	RW	dest_peripheral	目的设备。用于选择一个外设请求信号作为本通道的 DMA 目的设备的请求信号。 **注意：如果 DMA 传输的目的设备为存储**	0x0

			器，则忽略该值。	
[5]	-	reserved	保留。 **注意：写入时必须写 0，读出时应被屏蔽。**	0x0
[4:1]	RW	src_peripheral	源设备。用于选择一个外设请求信号作为本通道的 DMA 源设备的请求信号。 **注意：如果 DMA 传输的源设备为存储器，则忽略该值。**	0x0
[0]	RW	e_c	本通道开启使能位。 0：关闭； 1：开启。 将该位清零(即关闭通道)时，当前的总线传输会继续执行直至完成，然后通道关闭，FIFO 中剩余的数据全部丢失；当最后一个 LLI 完成或传输中出现错误时，通道也会被关闭，同时该 bit 被清零；如果需关闭通道且通道 FIFO 中的数据不丢失，则[h]也必须同时被置位，使通道忽略后续的 DMA 请求。然后必须轮询[a]，直至其值变为 0，表明通道 FIFO 中不再留有数据，此时才能清除[e_c]。 **注意：通过置位启动通道必须先重新初始化通道，然后才能再次启动通道；如果通过简单的置位启动通道，会引发不可预测性的后果。**	0x0

须知

当通过写[e_c]关闭一个通道时，必须等到轮询到寄存器 DMAC_ENBLD_CHNS 中的相应 bit 为 0 后，才能将[e_c]重新置位。这是因为通道实际的关闭并没有在将[e_c]清零后立即生效。总线 Burst 的运行延时也必须考虑此情况。

DMAC_Cn_CONFIG 寄存器的 flow_cntrl 字段对应的流控和传输类型如表 6-28 所示。

表6-28 流控制器及传输类型位定义

比特值	传输类型	控制器
000	存储器至存储器	DMAC
001	存储器至外设	DMAC
010	外设至存储器	DMAC
011	源设备至目的设备	DMAC

比特值	传输类型	控制器
100	源设备至目的设备	目的设备
101	存储器至外设	目的设备
110	外设至存储器	源设备
111	源设备至目的设备	源设备

6.12 ADC

6.12.1 概述

LSADC 为一款 SAR（Successive Approximations Register） ADC（逐次逼近型数模转换设备），实现将模拟信号转变成数字信号的功能。

表6-29 ADC 规格

参数	最小值	典型值	最大值	单位	描述
Power supply					
AVDD	1.71	1.8	1.89	V	模拟电源电压
DVDD	0.99	1.1	1.21	V	数字电源电压
LSADC 规范					
Full Scale Intput	2.5	-	3.6	V	ADC 输入范围
DNL	-	±1.5	-	LSB	差分非线性
INL	-	±1.5	-	LSB	积分非线性
Resolution	-	10	-	bit	精度
Power Dissipation	-	0.8	-	mA	工作时功耗
	-	1	-	μA	关机时功耗
Clock 规范					
f_{CLK}	-	-	3	MHz	输入时钟频率
DC	45	50	55	%	占空比
f_S	-	-	157	Ksps	采样率
T_{CONV}	-	15	-	CLK cycle	数据转换时间

6.12.2 功能描述

LSADC 模块具有以下功能特点：

- 输入时钟 3MHz，12bit 采样精度，单通道采样频率<200KHz。
- 共 8 个通道，支持软件配置 0～7 任意通道使能，逻辑按通道编号先低后高发起切换，每个通道采样 1 个数，通道切换时会使 ADC 电路进行 1 次复位处理。
- 支持从解复位到开始 ADC 数据转换时间寄存器可配。
- 支持 128×15bit FIFO 用于数据缓存，数据存储格式：高 3bit 为通道编号，低 12bit 为有效数据。
- 支持对 ADC 采样数据进行平均滤波处理，平均次数支持 1（不进行平均）、2、4、8；多通道时，每个通道接收 N 个数据（平均滤波个数）再切换通道。
- 支持 FIFO 水线中断、满中断上报。

须知

芯片 LSADC 的通道 7 为内部 VBAT 电压检测通道，非管脚输入。

6.12.3 工作方式

软件配置 ADC 的步骤如下：

1. CPU 配置扫描通道号 LSADC_CTRL0[ch_vld]，使能 1 个或多个通道；配置平均滤波模式 LSADC_CTRL0[equ_model_sel]。
2. 写 LSADC_CTRL1[rxintsize]、LSADC_CTRL2[rxim]、LSADC_CTRL2[rorim]，配置 FIFO 的中断和水线。
3. 写 LSADC_CTRL11[power_down]为 0，ADC 上电。
4. 写 LSADC_CTRL7[start]为 1，开始 ADC 采样。
5. 待中断上报，读 LSADC_CTRL9[dr]，获取 ADC 数据。
6. 写 LSADC_CTRL8[stop]为 1，写 LSADC_CTRL11[power_down]为 1，停止采样。
7. 读 LSADC_CTRL9[dr]，获取 ADC 数据，直至 FIFO 空（LSADC_CTRL10[rne]为 0）。

----结束

须知

- 配置过程中，LSADC_CTRL7[start]和 LSADC_CTRL8[stop]的配置必须组合配置（即在开始采样时配置[start]，在结束本次 ADC 采样后须配置[stop]），如果在进行第一次转换结束后未配置[stop]，在第二次 ADC 转化时直接配置[start]将导致数据错误，即不支持 start→start→stop 的操作。
- 请保证本次采样结束（配置完[start]）后将 FIFO 中的数据读空，否则下次采样时将存在上次 ADC 采样的残留数据。
- 请保证软件在第一次转换结束（配置 LSADC_CTRL8[stop]）后到配置第二次开始 AD 转换（配置 LSADC_CTRL8[start]）的时间间隔大于 6μs（大于 15 个 clk_3m 时钟周期）。

6.12.4 寄存器概览

ADC 寄存器概览如表 6-30 所示。

表6-30 ADC 寄存器概览（基址是 0x4007_0000）

偏移地址	名称	描述	页码
0x000	LSADC_CTRL0	ADC 控制寄存器	6-145
0x004	LSADC_CTRL1	ADC FIFO 设置寄存器	6-146
0x008	LSADC_CTRL2	ADC 中断控制寄存器	6-147
0x00C	LSADC_CTRL3	ADC 中断清除寄存器	6-147
0x010	LSADC_CTRL4	ADC FIFO 状态寄存器	6-147
0x014	LSADC_CTRL5	ADC 原始中断状态寄存器	6-148
0x018	LSADC_CTRL6	ADC 屏蔽后中断状态寄存器	6-148
0x01C	LSADC_CTRL7	ADC 起始扫描控制寄存器	6-149
0x020	LSADC_CTRL8	ADC 停止扫描控制寄存器	6-149
0x024	LSADC_CTRL9	ADC FIFO 读数据控制寄存器	6-149
0x028	LSADC_CTRL10	ADC 保留控制寄存器	6-150
0x02C	LSADC_CTRL11	ADC 下电控制寄存器	6-150

6.12.5 寄存器描述

LSADC_CTRL0

LSADC_CTRL0 为 ADC 控制寄存器。

Offset Address: 0x000　　Total Reset Value: 0x0000_F000

Bits	Access	Name	Description	Reset
[31:26]	RW	reserved	保留。	0x00
[25:24]	RW	cur_bais	模拟电源控制。 10：手动控制(AVDD=1.8V)； 11：手动控制(AVDD=3.3V)； 其他：自动识别模式。	0x0
[23:12]	RW	rst_cnt	从 RST 到开始转换的时间计数。 该值应≥15。	0x00F
[11:10]	RW	reserved	保留。	0x0
[9:8]	RW	equ_model_sel	平均算法模式选择。 00：1 次平均(即不进行平均)； 01：2 次平均算法模式； 10：4 次平均算法模式； 11：8 次平均算法模式。	0x0
[7:0]	RW	ch_vld	每个通道是否有效。 bit[0]～bit[7]分别对应通道 A～通道 H。 每 bit 对应的取值含义如下： 0：无效； 1：有效。	0x00

LSADC_CTRL1

LSADC_CTRL1 为 ADC FIFO 设置寄存器。

Offset Address: 0x004　　Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:3]	RW	reserved	保留。	0x00000000
[2:0]	RW	rxintsize	FIFO 水线设置。 当接收 FIFO 中数据个数≥配置的字数时，RXRIS 有效。 000：127； 001：124； 010：64； 011：32； 100：16；	0x2

			101：8； 110：4； 111：1。	

LSADC_CTRL2

LSADC_CTRL2 为 ADC 中断控制寄存器。

Offset Address: 0x008　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	RW	reserved	保留。	0x00000000
[1]	RW	rxim	水线中断屏蔽位。 0：屏蔽； 1：不屏蔽。	0x0
[0]	RW	rorim	FIFO 溢出中断屏蔽位。 0：屏蔽； 1：不屏蔽。	0x0

LSADC_CTRL3

LSADC_CTRL3 为 ADC 中断清除寄存器。

Offset Address: 0x00C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RW	reserved	保留。	0x00000000
[0]	W1_PULSE	roric	FIFO 溢出中断清除位。 写 0：不清除； 写 1：清除。	0x0

LSADC_CTRL4

LSADC_CTRL4 为 ADC FIFO 状态寄存器。

Offset Address: 0x010　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000

[2]	RO	bsy	ADC 忙标记。 0：空闲； 1：忙。	0x0
[1]	RO	rff	FIFO 是否已满。 0：未满； 1：已满。	0x0
[0]	RO	rne	FIFO 是否未空。 0：已空； 1：未空。	0x0

LSADC_CTRL5

LSADC_CTRL5 为 ADC 原始中断状态寄存器。

Offset Address: 0x014　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RO	rxris	接收 FIFO 水线中断的原始中断状态。 0：无中断； 1：有中断。	0x0
[0]	RO	rorris	接收溢出中断的原始中断状态。 0：无中断； 1：有中断。	0x0

LSADC_CTRL6

LSADC_CTRL6 为 ADC 屏蔽后中断状态寄存器。

Offset Address: 0x018　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RO	rxmis	接收 FIFO 水线中断屏蔽后的状态。 0：无中断； 1：有中断。	0x0
[0]	RO	rormis	接收溢出中断屏蔽后的状态。	0x0

			0：无中断； 1：有中断。	

LSADC_CTRL7

LSADC_CTRL7 为 ADC 起始扫描控制寄存器。

Offset Address: 0x01C　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	W1_PULSE	lsadc_start	LSADC 启动信号。 写 0：无效； 写 1：启动 LSADC。	0x0

LSADC_CTRL8

LSADC_CTRL8 为 ADC 停止扫描控制寄存器。

Offset Address: 0x020　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	W1_PULSE	lsadc_stop	停止自动扫描。 写 0：无效； 写 1：停止 LSADC 的扫描功能(需再写 LSADC_CTRL7[lsadc_start]才可以重新启动扫描)。	0x0

LSADC_CTRL9

LSADC_CTRL9 为 ADC FIFO 读数据控制寄存器。

Offset Address: 0x024　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:0]	RO	dr	读取 ADC 数据。 bit[14:12]：当前数据对应的通道编号(通道编号 0x0～0x7 分别代表通道 A～通道 H)；	0x0000

			bit[11:0]：有效数据(其中：bit[1:0]为小数位)。	

LSADC_CTRL10

LSADC_CTRL10 为 ADC 保留控制寄存器。

Offset Address: 0x028　　Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	RW	reserved	保留。	0x0
[27:24]	RW	lsadc_reg_3to0	通用寄存器 4。	0x0
[23:22]	RW	reserved	保留。	0x0
[21:16]	RW	lsadc_reg2_5to0	通用寄存器 3。	0x00
[15:14]	RW	reserved	保留。	0x0
[13:12]	RW	lsadc_reg1_7to6	通用寄存器 2。	0x0
[11:9]	RW	reserved	保留。	0x0
[8]	RW	lsadc_reg1_3	通用寄存器 1。	0x0
[7]	RW	reserved	保留。	0x0
[6:0]	RW	lsadc_reg0_6to0	通用寄存器 0。	0x00

LSADC_CTRL11

LSADC_CTRL11 为 ADC 下电控制寄存器。

Offset Address: 0x02C　　Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	RW	reserved	保留。	0x00000000
[0]	RW	power_down	ADC 下电控制。 0：正常工作； 1：下电。	0x1

7 JTAG

7.1 概述

Hi3861、Hi3861L、Hi3881 芯片内部集成一个自研 CPU，除了支持自带的 JTAG 调试接口外，还在内部集成 Coresight 调试架构，支持基于 Coresight 的 JTAG 和 SWD（Serial Wire Debug）调试接口，通过 Coresight 的 JTAG 或 SWD 调试接口实现调试。

在默认的情况下，使用的调试接口为 CPU 自带调试接口，如果需要使用 Coresight 调试接口，则需要配置调试模式使能寄存器 DBG_PORT_SEL[dbg_port_sel]为 1，切换到 Coresight 调试模式。

Hi3861、Hi3861L、Hi3881 芯片支持 Lauterbach 仿真器和 JLINK 仿真器。在接入仿真器之后需要根据调试模式使能寄存器 DBG_PORT_SEL 选择使用默认 JTAG 模式或 Coresight 的 JTAG 和 SWD 模式，否则无法进行仿真器连接。

7.2 调试接口

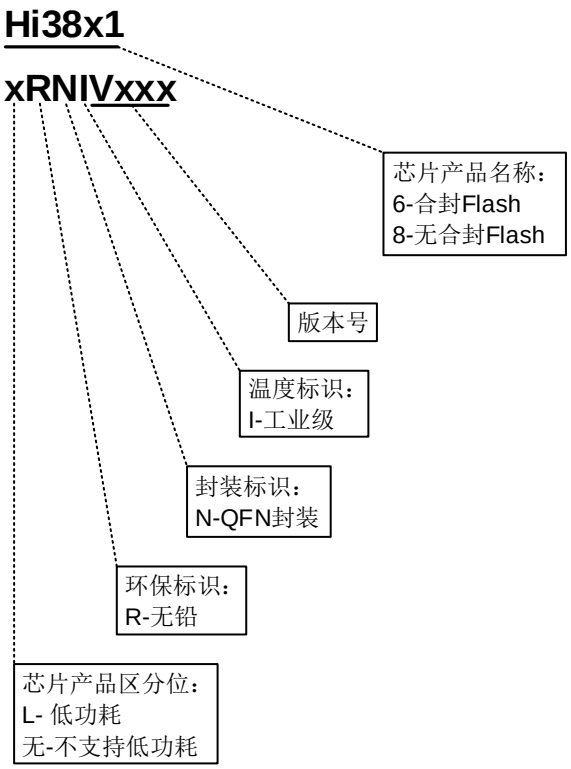
须知

芯片调试接口默认复用为其他功能，如果需要使用调试功能，则管脚 GPIO_08 在上电时置高电平，系统复位解除后，对应管脚即可复位为调试接口，此后 GPIO_08 管脚功能不受影响，可以作为正常管脚使用。

调试管脚与 PAD 名字的对应关系请参见《Hi3861V100/Hi3861LV100/Hi3881V100 WiFi 芯片 硬件用户指南》。

A 订购须知

图A-1 芯片 Mark 命名规则



注：版本号中的“xxx”表示可能会变化的数字，仅供内部使用。

表A-1 Hi3861、Hi3861L、Hi3881 不同应用的配置

Part Number	封装类型	封装尺寸	Pitch
Hi3861RNIV100	QFN 32	5mm×5mm	0.4mm
Hi3861LRNIV100	QFN 32	5mm×5mm	0.4mm
Hi3881RNIV100	QFN 32	5mm×5mm	0.4mm

B 缩略语

A		
ABB	Analog Baseband	模拟基带
ACK	Acknowledgement	通信协议中的肯定应答
ADC	Analog to Digital Converter	模数转换器
AES	Advanced Encryption Standard	高级加密标准
AGC	Automatic Gain Control	自动增益控制
AHB	Advanced High Performance Bus	先进高性能总线
AMBA	Advanced Microcontroller Bus Architecture	高级微控制器总线架构
A-MPDU	Aggregate MAC Protocol Data Unit	MAC 协议数据单元聚合
APB	Advanced Peripheral Bus	高级外设总线
ASIC	Application-Specific Integrated Circuit	专用集成电路
ATE	Automatic Test Equipment	自动测试设备
AVDD	Analog Voltage Device Drain	模拟电源电压
B		
BCC	Binary Convolutional Code	二进制卷积代码
BCLK	Bit Clock	位时钟
BLE	Bluetooth Low Energy	低功耗蓝牙
BPSK	Binary Phase Shift Keying	二进制相移键控
BSS	Basic Service Set	基本服务集
BT	Bluetooth	蓝牙

C

CBB	Common Building Block	通用基础模块
CBC	Cipher Block Chaining	密码分组链接
CBUS	C-BUS	一种 I2C 之前的总线协议
CCA	Clear Channel Assessment	空闲信道评估
CCK	Complementary Code Keying	补码键控
CG	Clock Gating	时钟门控
CMU	Clock Managing Unit	时钟管理单元
CODEC	Coder/Decoder	编解码器
CPU	Central Processing Unit	中央处理器
CRG	Clock and Reset Generator	时钟和复位生成器
CSI	Channel State Information	信道状态信息
CTR	Counter	计算器

D

DAC	Digital Analog Converter	数字模拟转换器
DC	Direct Current	直流电
DC	Duty Cycle	占空比
DBAC	Dual Band Adaptive Concurrent	双频自适应并发
DBB	Digital Baseband	数字基带
DBDC	Dual Band Dual Concurrent	双频双发
DFT	Design For Testability	可测性设计
DMA	Direct Memory Access	直接存储器存取
DMAC	Direct Memory Access Controller	直接存储器存取控制器
DNL	Differential Nonlinearlty	差分非线性
DPD	Digital Pre-Distortion	数字预失真
DRBG	Deterministic Random Bit Generator	确定性随机数产生器
DS	Drive Strength	驱动强度
DSSS	Direct Sequence Spread Spectrum	直接序列扩频
DTIM	Delivery Traffic Indication Map	传输流量指示消息
DVDD	Digital Voltage Device Drain	数字电源电压

E

ECB	Electronic Code Book	电子密码本
ECC	Elliptic Curve Cryptography	椭圆加密算法
EFUSE	Electrical FUSE	一次性的电可编程的 FUSE
EIFS	Extended Inter-Frame Space	扩展帧间间隔
EMI	Electromagnetic Interference	电磁干扰

F

FIFO	First In First Out	先进先出
FIPS	Federal Information Processing Standards	联邦信息处理标准
FOUT	Frequency Output	频率输出
FSCLK	Sampling Frequency Clock	采样时钟
FTM	Fine Timing Measurement	精准测时机制

G

GI	Guard Interval	保护间隔
GPIO	General Purpose Input/Output	通用目的输入输出接口

H

HMAC	Hash-based Message Authentication Code	散列信息认证码
HPI	Hardware Platform Interface	硬件平台接口
HPM	Hardware Performance Monitor	硬件性能监控
HT	High Throughput	高吞吐量
HVT	High Voltage Threshold	高阈值电压

I

I2C	The Inter-Integrated Circuit	一种串行总线协议标准
I2S	Inter-IC Sound	集成电路内置音频总线
ID	Identifier	标识符
IEEE	Institute of Electrical and Electronics Engineers	电气和电子工程师学会[美]

IF	Interface	接口
INL	Integral Nonlinearity	积分非线性
IO	Input Output	输入输出
IP	Intelligent Property	知识资产
ISA	Instruction Set Architecture	指令集架构
IV	Initialization Vector	输入向量
J		
JTAG	Joint Test Action Group	联合测试行为组织，是一种国际标准的测试协议
K		
KDF	Key Derivation Function	密钥导出函数
L		
LDO	Low Dropout Regulator	低压差调节器
LNA	Low Noise Amplifier	低噪声放大器
LO	Local Oscillator	本地振荡器
LPF	Low Pass Filter	低通滤波器
LRCLK	Left Right Clock	左右声道时钟
LSADC	Low-Speed Analog-to-Digital Converter	低速模数转换器
LSB	Least Significant Bit	最低位
LUT	Lookup Table	查找表
LVT	Low Voltage Threshold	低阈值电压
M		
MAC	Media Access Control	媒体接入控制
MCLK	Master Clock	主时钟
MCU	Main Control Unit	主控制器单元
MSB	Most Significant Bit	最高位
MU-MIMO	Multi-User Multiple-Input Multiple-Output	多用户多输入多输出
MUX	Multiplexer	复用器

N

NMI	Non-Maskable Interrupt	不可屏蔽中断

O

OFDM	Orthogonal Frequency Division Multiplex	正交频分复用
OSC	Oscillator	振荡器
OTT	Over The Top	OTT 解决方案
OVP	Overvoltage Protection	过压保护

P

P2P	Peer-to-Peer	点到点
PA	Power Amplifier	功率放大器
PD	Power Down	下电
PHY	Physical Layer	物理层
PKE	Public-Key-Engine	公钥引擎
PLL	Phase-Locked Loop	锁相环
PMP	Physical Memory Protection	物理内存保护
PMU	Power Management Unit	电源管理模块
POR	Power On Reset	上电复位
PP	Page Program	页编程
PPM	Parts Per Million	百万分率
PSM	Power Saving Mode	节电模式
PTA	Packet Traffic Arbitration	包流量仲裁
PWM	Pulse-Width Modulation	脉冲宽度调制

Q

QAM	Quadrature Amplitude Modulation	正交幅度调制
QFN	Quad Flat Non-leaded package	四侧无引脚扁平封装
QoS	Quality of Service	业务质量
QPSK	Quadrature Phase Shift Keying	正交相移键控

R

RAM	Random Access Memory	随机存取存储器
RDSR	Read Status Register	读状态寄存器
REF	Reference Indicator	基准信号指示
RF	Radio Frequency	射频
RIFS	Reduced Interface Space	缩短帧间间隔
RISC	Reduced Instruction Set Computing	精简指令集计算
RO	Ring Oscillator	振荡
ROM	Read-Only Memory	只读存储器
RSA	Rivest-Shamir-Adleman	RSA 加密算法
RSSI	Received Signal Strength Indicator	接收信号强度指示
RTC	Real-Time Clock	实时时钟
RX	Receiver	接收器

S

SAR	Successive Approximations Register	逐次逼近寄存器
SCK	Serial Clock singal	串行时钟信号
SCL	Serial Clock Line	串行时钟线
SCLK	Serial Clock	串行时钟
SCO	Sampling Clock Offset	采样时钟偏移

SD	Serial Data	串行数据
SD	Secure Digital	安全数字
SDA	Serial Data and Address	串行数据地址线
SDIO	Secure Digital Input/Output	安全数字输入输出接口
SDR	Single Date Rate	单倍速率
SFC	Serial Peripheral Interface Flash Controller	SPI Flash 控制器
SHA	Secure Hash Algorithm	安全散列算法
SIFS	Short Interframe Space	短帧间间隔
SNR	Signal Noise Ratio	信噪比
SoC	System On Chip	片上系统
SPI	Synchronous Peripheral Interface	同步串行接口
SPH	SPICLKOUT Phase	SPICLKOUT 相位
SPO	SPICLKOUT Polarity	SPICLKOUT 极性
SRAM	Static Random Access Memory	静态随机存储器
SSC	Spread Spectrum Clocking	扩频时钟
SSCG	Spread Spectrum Clock Generator	展频时钟时钟生成器
SSI	Synchronous Serial Interface	同步串行接口
SSS	Security Sub System	安全子系统
STA	Station	站点
STBC	Space-Time Block Coding	空时分组编码
SVB	Selective Voltage Binning	选择电压分级
SVT	Standard Voltage Threshold	标准阈值电压
SWD	Serial Wire Debug	串行线调试
T		
TI	Texas Instruments	德州仪器

TPC	Transmite Power Control	发射功率控制
TRNG	True Random Number Generator	真随机数发生器
TSF	Timing Synchronization Function	定时同步功能
TX	Transmitter	发送器

U

UAPSD	Unscheduled Automatic Power Save Delivery	非调度自动节能发送
UART	Universal Asynchronous Receiver & Transmitter	通用异步收发器
UPC	UP Converter	升压模块
UVLO	Under Voltage Lock Out	欠压锁定

V

VAP	Virtual Access Point	虚拟接入点
VCO	Voltage Controlled Oscillator	压控振荡器
VGA	Variable Gain Amplifier	可变增益放大器
VHT	Very High Throughput	极高吞吐量
VSWR	Voltage Standing Wave Ratio	电压驻波比

W

WADC	WiFi Analog Digital Converter	无线模拟数字转换器
WAPI	WLAN Authentication and Privacy Infrastructure	无线局域网鉴别和保密基础结构
WDAC	WiFi digital analog converter	无线数字模拟转换器
WDT	Watch Dog Timer	看门狗定时器
WFA	Wi-Fi Alliance	Wi-Fi 联盟
WFI	Wait For Interrupt	等待中断
WiFi	Wireless Fidelity	无线保真

WIP	Write In Progress	处于写状态
WLAN	Wireless Local Area Network	无线局域网
WPA	Wi-Fi Protected Access	Wi-Fi 保护访问
WPS	Wi-Fi Protected Setup	Wi-Fi 保护设置
WREN	Write Read Enable	读写使能
WS	Word Select	字选

X

XIP	Executed In Place	芯片内执行
XTAL	Quartz Crystal Unit	石英晶体谐振器

