



# 曳影 1520 外围接口 用户手册

文档版本	1.0.0
保密等级	保密
发布日期	2023-08-26

**Copyright © 2022 T-HEAD (Shanghai) Semiconductor Co., Ltd. All rights reserved.**

This document is the property of T-HEAD (Shanghai) Semiconductor Co., Ltd. This document may only be distributed to: (i) a T-HEAD party having a legitimate business need for the information contained herein, or (ii) a non-T-HEAD party having a legitimate business need for the information contained herein. No license, expressed or implied, under any patent, copyright or trade secret right is granted or implied by the conveyance of this document. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of T-HEAD (Shanghai) Semiconductor Co., Ltd.

**Trademarks and Permissions**

The T-HEAD Logo and all other trademarks indicated as such herein are trademarks of T-HEAD (Shanghai) Semiconductor Co., Ltd. All other products or service names are the property of their respective owners.

**Notice**

The purchased products, services and features are stipulated by the contract made between T-HEAD and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

**平头哥（上海）半导体技术有限公司 T-HEAD (Shanghai) Semiconductor Co., LTD**

Address: 5th Floor Number 2 Chuan He Road 55, Number 366 Shang Ke Road, Shanghai free trade area, China  
Website: [www.t-head.cn](http://www.t-head.cn)

**Copyright © 2022 平头哥（上海）半导体技术有限公司，保留所有权利。**

本文档的所有权及知识产权归属于平头哥（上海）半导体技术有限公司及其关联公司(下称“平头哥”)。本文档仅能分派给：(i) 拥有合法雇佣关系，并需要本文档的信息的平头哥员工，或(ii)非平头哥组织但拥有合法合作关系，并且其需要本文档的信息的合作方。对于本文档，未经平头哥（上海）半导体技术有限公司明示同意，则不能使用该文档。在未经平头哥（上海）半导体技术有限公司的书面许可的情形下，不得复制本文档的任何部分，传播、转录、储存在检索系统中或翻译成任何语言或计算机语言。

**商标申明**

平头哥的 LOGO 和其它所有商标归平头哥（上海）半导体技术有限公司及其关联公司所有，未经平头哥（上海）半导体技术有限公司的书面同意，任何法律实体不得使用平头哥的商标或者商业标识。

**注意**

您购买的产品、服务或特性等应受平头哥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，平头哥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。平头哥（上海）半导体技术有限公司不对任何第三方使用本文档产生的损失承担任何法律责任。

**平头哥（上海）半导体技术有限公司 T-HEAD (Shanghai) Semiconductor Co., LTD**

地址：中国（上海）自由贸易试验区上科路 366 号、川和路 55 弄 2 号 5 层  
网址： [www.t-head.cn](http://www.t-head.cn)

## 版本历史

版本	说明	作者	日期
V1.0.0	初始版本	平头哥	2023-08-26

# 目录

版本历史.....	I
目录.....	II
图表目录.....	IV
术语与缩略语.....	X
1 GMAC .....	1
1.1 概述 .....	1
1.2 主要特性 .....	1
1.3 接口 .....	1
1.4 功能描述 .....	3
1.5 使用 .....	4
1.6 寄存器描述.....	5
2 USB .....	120
2.1 概述 .....	120
2.2 主要特性 .....	120
2.3 接口 .....	120
2.4 功能描述 .....	122
2.5 使用 .....	122
2.6 寄存器描述.....	123
3 MPJTAG .....	148
3.1 概述 .....	148
3.2 主要特性 .....	148
3.3 接口 .....	149
3.4 功能描述 .....	149
3.5 寄存器描述.....	151
3.6 使用 .....	158
4 ADC.....	159
4.1 概述 .....	159
4.2 主要特性 .....	159
4.3 接口 .....	159
4.4 功能描述 .....	160
4.5 使用 .....	161
4.6 寄存器描述.....	163

5 I2S.....	186
5.1 概述 .....	186
5.2 主要特性 .....	186
5.3 接口 .....	187
5.4 功能描述 .....	187
5.5 使用 .....	195
5.6 寄存器描述.....	195
6 PWM.....	220
6.1 概述 .....	220
6.2 主要特性 .....	220
6.3 接口 .....	220
6.4 功能描述 .....	221
6.5 使用 .....	222
6.6 寄存器描述.....	223
7 I2C.....	240
7.1 概述 .....	240
7.2 主要特性 .....	240
7.3 接口 .....	241
7.4 使用 .....	243
7.5 寄存器描述.....	244
8 UART.....	302
8.1 概述 .....	302
8.2 主要特性 .....	302
8.3 接口 .....	302
8.4 功能描述 .....	303
8.5 使用 .....	304
8.6 寄存器描述.....	305
9 GPIO.....	349
9.1 概述 .....	349
9.2 主要特性 .....	349
9.3 接口 .....	349
9.4 功能描述 .....	350
9.5 使用 .....	350
9.6 寄存器描述.....	350

## 图表目录

图表 1-1 GMAC 顶视图 .....	1
图表 1-2 GMAC 接口 .....	2
图表 1-3 GMAC 与 PHY 连接 .....	2
图表 1-4 RGMII 时序 .....	3
图表 1-5 RGMII 发送和接收时序 .....	3
图表 1-6 GMAC APB 寄存器内存映射 .....	5
图表 1-7 GMAC_CLK_EN .....	6
图表 1-8 GMAC_RXCLK_DELAY_CTRL .....	8
图表 1-9 GMAC_TXCLK_DELAY_CTRL .....	9
图表 1-10 GMAC_PLLCLK_DIV .....	9
图表 1-11 GMAC_EPHY_DIV .....	10
图表 1-12 GMAC_PTP_DIV .....	10
图表 1-13 GMAC_GTXCLK_SEL .....	11
图表 1-14 GMAC_PHY_INTF_SEL .....	11
图表 1-15 GMAC_TXCLK_OEN .....	11
图表 1-16 寄存器 0（总线模式寄存器） .....	12
图表 1-17 有效 PBL 范围 .....	16
图表 1-18 寄存器 1（发送轮询需求寄存器） .....	18
图表 1-19 寄存器 2（接收轮询需求寄存器） .....	18
图表 1-20 寄存器 3（接收描述符列表地址寄存器） .....	19
图表 1-21 寄存器 4（发送描述符列表地址寄存器） .....	19
图表 1-22 寄存器 5（状态寄存器） .....	20
图表 1-23 寄存器 6（操作模式寄存器） .....	26
图表 1-24 寄存器 7（中断使能寄存器） .....	32
图表 1-25 寄存器 8（丢失帧和缓冲区溢出计数器寄存器） .....	35
图表 1-26 寄存器 9（接收中断看门狗定时器寄存器） .....	36
图表 1-27 寄存器 10（AXI 总线模式寄存器） .....	36
图表 1-28 寄存器 11（AHB 或 AXI 状态寄存器） .....	39
图表 1-29 寄存器 18（当前主机发送描述符寄存器） .....	40
图表 1-30 寄存器 19（当前主机接收描述符寄存器） .....	41
图表 1-31 寄存器 20（当前主机发送缓冲区地址寄存器） .....	41
图表 1-32 寄存器 21（当前主机接收缓冲区地址寄存器） .....	42

图表 1-33 寄存器 22 (HW 特性寄存器) .....	42
图表 1-34 寄存器 76 (通道 1 插槽功能控制和状态寄存器) .....	44
图表 1-35 寄存器 88 (通道 1 CBS 控制寄存器) .....	45
图表 1-36 寄存器 89 (通道 1 CBS 状态寄存器) .....	46
图表 1-37 寄存器 90 (通道 1 idleSlopeCredit 寄存器) .....	46
图表 1-38 寄存器 91 (通道 1 sendSlopeCredit 寄存器) (续) .....	47
图表 1-39 寄存器 92 (通道 1 hiCredit 寄存器) .....	47
图表 1-40 寄存器 93 (通道 1 loCredit 寄存器) .....	48
图表 1-41 寄存器 0 (MAC 配置寄存器) .....	48
图表 1-42 基于第 27 位和第 20 位的巨帧状态.....	56
图表 1-43 寄存器 1 (MAC 帧过滤器) .....	56
图表 1-44 寄存器 2 (散列表高位寄存器) .....	60
图表 1-45 寄存器 3 (散列表低位寄存器) .....	61
图表 1-46 寄存器 4 (GMII 地址寄存器) .....	61
图表 1-47 寄存器 5 (GMII 数据寄存器) .....	63
图表 1-48 寄存器 6 (流控制寄存器) .....	64
图表 1-49 寄存器 7 (VLAN Tag 寄存器) .....	66
图表 1-50 寄存器 8 (版本寄存器) .....	67
图表 1-51 寄存器 9 (调试寄存器) .....	68
图表 1-52 寄存器 12 (LPI 控制和状态寄存器) .....	71
图表 1-53 寄存器 13 (LPI 定时器控制寄存器) .....	73
图表 1-54 寄存器 14 (中断状态寄存器) .....	74
图表 1-55 寄存器 15 (中断屏蔽寄存器) .....	76
图表 1-56 寄存器 16 (MAC 地址 0 高位寄存器) .....	78
图表 1-57 寄存器 17 (MAC 地址 0 低位寄存器) .....	78
图表 1-58 寄存器 18 (MAC 地址 1 高位寄存器) .....	79
图表 1-59 寄存器 19 (MAC 地址 1 低位寄存器) .....	80
图表 1-60 寄存器 544 (MAC 地址 32 高位寄存器) .....	81
图表 1-61 寄存器 48 (AN 控制寄存器) .....	81
图表 1-62 寄存器 49 (AN 状态寄存器) .....	83
图表 1-63 寄存器 50 (自动协商广告寄存器) .....	84
图表 1-64 寄存器 51 (自动协商对端能力寄存器) .....	85
图表 1-65 寄存器 52 (自动协商扩展寄存器) .....	86
图表 1-66 寄存器 53 (TBI 扩展状态寄存器) .....	87
图表 1-67 寄存器 54 (SGMII/RGMII/SMII 控制和状态寄存器) .....	87

图表 1-68 寄存器 55（看门狗超时寄存器） .....	89
图表 1-69 寄存器 56（通用 IO 寄存器） .....	89
图表 1-70 寄存器 256（第 3 层和第 4 层控制寄存器 0） .....	91
图表 1-71 寄存器 257（第 4 层地址寄存器 0） .....	94
图表 1-72 寄存器 260（第 3 层地址 0 寄存器 0） .....	95
图表 1-73 寄存器 261（第 3 层地址 1 寄存器 0） .....	95
图表 1-74 寄存器 262（第 3 层地址 2 寄存器 0） .....	96
图表 1-75 寄存器 263（第 3 层地址 3 寄存器 0） .....	96
图表 1-76 寄存器 320（散列表寄存器 0） .....	98
图表 1-77 寄存器 353（VLAN Tag 包含或替换寄存器） .....	98
图表 1-78 寄存器 354（VLAN 散列表寄存器） .....	100
图表 1-79 寄存器 448（时间戳控制寄存器） .....	100
图表 1-80 间戳快照对寄存器位的依赖性 .....	104
图表 1-81 寄存器 449（亚秒级增量寄存器） .....	104
图表 1-82 寄存器 450（系统时间-秒寄存器） .....	105
图表 1-83 寄存器 451（系统时间-亚秒寄存器） .....	105
图表 1-84 寄存器 452（系统时间-秒更新寄存器） .....	106
图表 1-85 寄存器 453（系统时间-纳秒更新寄存器） .....	106
图表 1-86 寄存器 454（时间戳加数寄存器） .....	107
图表 1-87 寄存器 455（目标时间秒寄存器） .....	107
图表 1-88 寄存器 456（目标时间纳秒寄存器） .....	107
图表 1-89 寄存器 457（系统时间-高字秒寄存器） .....	108
图表 1-90 寄存器 458（时间戳状态寄存器） .....	108
图表 1-91 寄存器 459（PPS 控制寄存器） .....	111
图表 1-92 寄存器 460（辅助时间戳-纳秒寄存器） .....	114
图表 1-93 寄存器 461（辅助时间戳-秒寄存器） .....	115
图表 1-94 寄存器 462（AV MAC 控制寄存器） .....	116
图表 1-95 寄存器 472（PPS0 间隔寄存器） .....	117
图表 1-96 寄存器 473（PPS0 宽度寄存器） .....	117
图表 1-97 寄存器 480（PPS1 目标时间秒寄存器） .....	118
图表 1-98 寄存器 481（PPS1 目标时间纳秒寄存器） .....	118
图表 2-1 引脚描述 .....	120
图表 2-2 功能逻辑图 .....	122
图表 3-1 MPJTAG 功能框图 .....	148
图表 3-2 管脚描述表 .....	149



图表 3-3 MPJTAG 时序图.....	149
图表 3-4 MPJTAG 时序特性 .....	149
图表 3-5 系统描述符寄存器映射 .....	150
图表 3-6 系统描述符读取流程图 .....	158
图表 4-1 ADC 管脚描述 .....	159
图表 4-2 功能逻辑图 .....	161
图表 4-3 ADC 采样数据格式图.....	162
图表 5-1 I2S 功能框图.....	186
图表 5-2 管脚描述.....	187
图表 5-3 I2S 的简单系统配置 .....	188
图表 5-4 I2S 标准立体帧格 .....	189
图表 5-5 左对齐的 I2S 立体帧格式.....	189
图表 5-6 右对齐的 I2S 立体帧格式.....	189
图表 5-7 I2S 时钟发生器电路图 .....	190
图表 5-8 时钟配置.....	190
图表 5-9 系统中的 I2S.....	192
图表 5-10 发送 FIFO 和接收 IFFO DMA 接口工作流程.....	193
图表 5-11 DMA 接口时序 .....	193
图表 5-12 中断&错误寄存器之间的关系.....	195
图表 5-13 I2S 内存映射.....	195
图表 5-14 IISEN 字段说明 .....	197
图表 5-15 FUNCMODE 字段说明 .....	198
图表 5-16 IISCNF_IN 字段说明 .....	198
图表 5-17 标准立体声帧格式 @RALOLRC=1 .....	200
图表 5-18 FSSTA 字段说明.....	200
图表 5-19 16 位输入/输出和 FIFO 以 16 位存储数据.....	202
图表 5-20 16 位输入/输出和 FIFO 以 24 位存储数据.....	202
图表 5-21 16 位输入/输出和 FIFO 以 32 位存储数据.....	202
图表 5-22 24 位输入/输出和 FIFO 以 16 位存储数据.....	203
图表 5-23 24 位输入/输出和 FIFO 以 24 位存储数据.....	203
图表 5-24 24 位输入/输出和 FIFO 以 32 位存储数据.....	203
图表 5-25 32 位输入/输出和 FIFO 以 16 位存储数据.....	204
图表 5-26 32 位输入/输出和 FIFO 以 24 位存储数据.....	204
图表 5-27 32 位输入/输出和 FIFO 以 32 位存储数据.....	204
图表 5-28 实际输入采样频率.....	205

图表 5-29 IISCNF_OUT 字段说明.....	205
图表 5-30 FADTLR 字段说明 .....	206
图表 5-31 SCCR 字段说明 .....	207
图表 5-32 数据压缩@T/RVCCR = 3, DATAWTH[1:0] = 00 .....	208
图表 5-33 数据压缩@T/RVCCR = 15, DATAWTH[1:0] = 01 .....	209
图表 5-34 数据压缩@T/RVCCR=5, DATAWTH[1:0]=1? .....	209
图表 5-35 数据压缩@ =1, DATAWTH[1:0]=00@立体声源 .....	209
图表 5-36 数据压缩@SSRCR=1, DATAWTH[1:0]=00@单声道源.....	209
图表 5-37 数据压缩@SSRCR=2, DATAWTH[1:0]=00@立体声源.....	209
图表 5-38 数据压缩@SSRCR=2, DATAWTH[1:0]=00@单声道源.....	210
图表 5-39 数据压缩@SSRCR=2, DATAWTH[1:0]=01@立体声源.....	210
图表 5-40 数据压缩@SSRCR=2, DATAWTH[1:0]=01@单声道源.....	210
图表 5-41 TXFTLR 字段说明 .....	210
图表 5-42 RXFTLR 字段说明.....	211
图表 5-43 TXFLR 字段说明 .....	211
图表 5-44 RXFLR 字段说明.....	212
图表 5-45 SR 字段说明.....	212
图表 5-46 IMR 字段说明.....	213
图表 5-47 ISR 字段说明.....	214
图表 5-48 RISR 字段说明 .....	215
图表 5-49 ICR 字段说明.....	217
图表 5-50 DMACR 字段说明.....	218
图表 5-51 DMATDLR 字段说明.....	218
图表 5-52 DMARDLR 字段说明.....	219
图表 5-53 DIV0_LEVEL 字段说明.....	219
图表 5-54 DIV3_LEVEL 字段说明.....	219
图表 6-1 PWM 模块功能图 .....	220
图表 6-2 PWM 接口时序图 .....	220
图表 6-3 引脚描述.....	221
图表 6-4 单次模式图 .....	221
图表 6-5 连续模式图 .....	222
图表 7-1 管脚描述.....	241
图表 7-2 I2C 标准模式和快速模式时序特性.....	241
图表 7-3 I2C 标准模式和快速模式时序特性.....	242
图表 7-4 I2C 高速模式时序特性 .....	242

图表 7-5 I2C 高速模式时序特性 .....	243
图表 8-1 管脚描述.....	302
图表 8-2 UART 串行数据格式 .....	303
图表 9-1 管脚描述.....	349

## 术语与缩略语

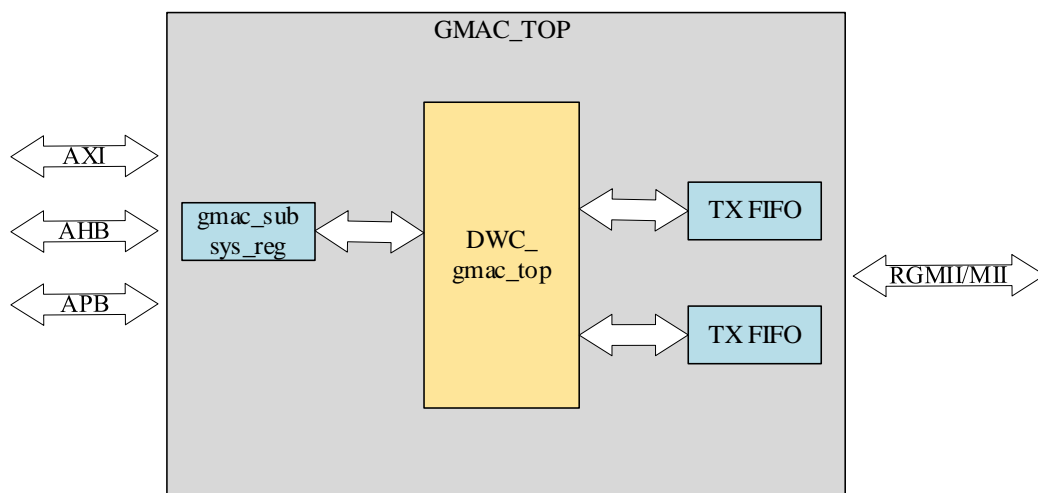
缩略语	英文全名	中文解释
ADC	Analog-to-Digital Converter	模数转换器
DMA	Direct Memory Access	直接存储器访问
DRD	Dual Role Device	双角色设备
FIFO	First Input First Output	先入先出
GMAC	Gigabit Media Access Control	千兆媒体接入控制
GPIO	General Purpose Input Output	通用输入/输出口
I2S	Inter-IC Sound	集成电路内置音频总线
MPJTAG	Multi-Processor JTAG	多 CPU 核 JTAG 调试接口
PIPE	The PHY Interface for the PCI Express, SATA, and USB Architectures	PCIe/SATA/USB 的物理层接口
PWM	Pulse Width Modulation	脉冲宽度调制
RX FIFO	Receiver FIFO	接收 FIFO
sclk	serial clock	串行时钟
sd	serial data	串行数据
SPI	Serial Peripheral Interface	串行外设接口
TX FIFO	Transmitter FIFO	发送 FIFO
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发器
UTMI	USB2.0 Transceiver Macrocell Interface	USB2.0 物理层接口
ws	word select	字段（声道）选择
xHCI	eXtensible Host Controller Interface	扩展主控制器接口

# 1 GMAC

## 1.1 概述

Ethernet IP Core 能够以 10/100/1000 Mbps 的速率运行，用于以太网和快速以太网应用。完整的以太网解决方案需要一个外部 PHY。

Ethernet IP Core 在半双工或全双工模式下工作。在半双工模式下，控制器支持 IEEE802.3 带有冲突检测的载波侦听多路存取(CSMA/CD)协议。在全双工模式下，支持 IEEE802.3 MAC 控制层，包括流量控制的暂停操作。



图表 1-1 GMAC 顶视图

## 1.2 主要特性

- 符合 IEEE802.3 规范
- IEEE 1588-2008 精密网络时钟同步标准
- 支持 10/100/1000Mbps 数据传输速率
- 支持 RGMII/MII 接口
- 在发送路径中插入前导码和帧开始界定符 (SFD)
- 在接收路径中删除前导码和帧开始符 (SFD)
- 接收帧的自动 CRC 和 pad 生成选项
- 用于 PHY 设备配置和管理的 MDIO 主接口

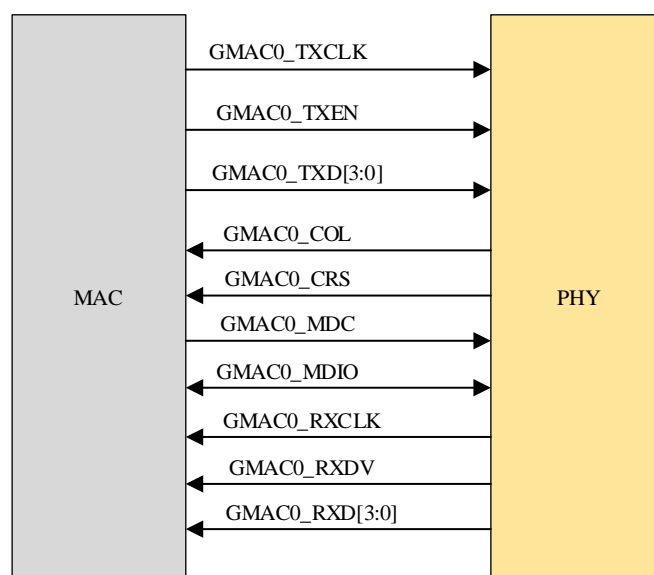
## 1.3 接口

GMAC 接口如[图表 1-2](#) 所示。

图表 1-2 GMAC 接口

引脚	方向	宽度	说明
GMAC0_TXCLK	O	1	用于输出的 TX 时钟
GMAC0_TXD	O	4	PHY 发送数据
GMAC0_TXEN	O	1	PHY 发送使能
GMAC0_COL	I	1	该信号仅在 MII 模式下有效，当在介质上检测到冲突时由 PHY 启用。不用于电影 1520 项目。
GMAC0_CRS	I	1	该信号仅在 MII 模式下有效，在发送或接收介质未空闲时由 PHY 启用。不用于电影 1520 项目。
GMAC0_RXCLK	I	1	用于输出的 RX 时钟
GMAC0_RXD	I	4	PHY 接收数据，在电影 1520 项目中使用 0~3bit。
GMAC0_RXDV	I	1	PHY 接收数据有效
GMAC0_MDC	O	1	管理数据时钟
GMAC0_MDIO	IO	1	管理数据

GMAC 接口与 PHY 连接如图表 1-3 所示。

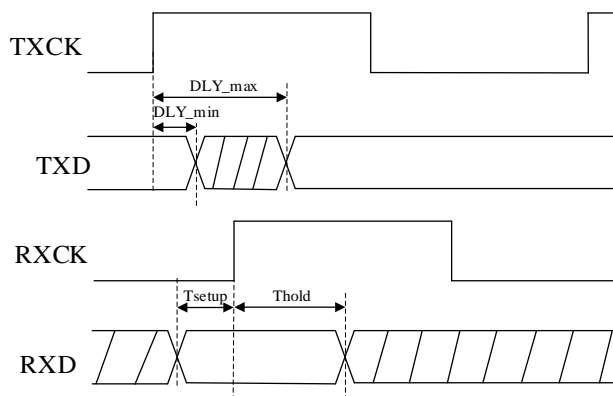


图表 1-3 GMAC 与 PHY 连接

图表 1-4 显示了 RGMII 模式的时序。

图表 1-4 RGMII 时序

信号	最小值	典型值	最大值	单位	备注
TXCK	-	125	125	MHz	发送频率
RXCK	-	125	-	MHz	接收频率
Duty	45%	50%	55%	%	TCK 占空比
tDLY	-0.3	-	1.3	ns	输出延迟
tSETUP	1.0	-	-	ns	输入设置时间
tHOLD	0.6	-	-	ns	输入保持时间



图表 1-5 RGMII 发送和接收时序

## 1.4 功能描述

MAC 支持多种面向 PHY 芯片的接口，如 RGMII 和 MII。复位后只能选择一次 PHY 接口。MAC 通过 MAC 发送接口（MTI）、MAC 接收接口（MRI）和 MAC 控制接口（MCI）与应用端进行通信。

### 1.4.1 发送

总线接口模块（TBU）：通过 FIFO 接口将 MAC 的发送路径与外部帧连接起来。

帧控制器模块（TFC）：由两个寄存器组成，用于保存数据、启用字节和控制从 TBU 接收到的最后一个数据。寄存器在应用程序和发送协议引擎之间提供一个缓冲区，用于调节数据流，并将输入数据转换为 8 位总线送往 TPE。

协议引擎模块（TPE）：由一个控制 Ethernet 帧传输操作的传输状态机组成。

- 生成前导码和 SFD
- 生成发送帧状态

- 包含支持 IEEE 1588 的时间戳快照逻辑

## 1.4.2 接收

协议引擎模块（RPE0）：RPE 由接收状态机组成，它剥离接收到的 Ethernet 帧的前导码、SFD 和载波扩展。

CRC 模块（CRX）：接收 CRC（CRX）连接 RPE 模块，检查接收帧是否存在 CRC 错误。

帧控制器模块（RFC）：接收帧控制器（RFC）从 RPE 模块接收以太网帧数据和状态。RFC 模块由一个参数化深度的 FIFO 和两个用于写和读 FIFO 的状态机组成。FIFO 保存接收到的以太网帧数据和字节使能，以及用于指示最后数据的控制位。状态机管理 FIFO，并为接收来自 RPE 模块的以太网帧提供帧缓冲。RFC 模块的主要功能包括：

- 数据路径转换，将 8 位数据转换为 32 位数据到 RBU 模块
- 帧过滤
- 附加从 IPC 输入的计算 IP 校验和
- 更新接收状态并转发给 RBU

## 1.5 使用

### 1.5.1 初始化时钟/复位/PAD

初始化 sys 寄存器和 ioctrl 寄存器，用于配置 txclk 方向和 txclk 频率。

- MII 接口，txclk 从 pad 输入；RGMII 接口，txclk 输出到 pad。
  - 1000Mbps/100Mbps，配置 PLL 分频寄存器，用于 txclk 125MHz/25MHz。
  - 请确保 gmac\_prst\_n/gmac\_hrst\_n/gmac\_grst\_n/gmac\_arst\_n 处于释放模式。
1. 配置 PHY 接口类型，MII 接口或 RGMII 接口。
  2. 配置时钟频率，1000Mbps 速率下使用 125MHz 的时钟频率，100Mbps 速率下使用 25MHz 的时钟频率。
  3. 配置时钟启用。
  4. 配置 tx-clock 方向：  
如果是 MII 接口，从 PHY 中选择 tx-clk；否则，从 PLL 中选择 tx-clk。  
如果是 MII 接口，则配置 tx-clk 为输入方向，否则，配置 tx-clk 为输出方向。

### 1.5.2 初始化 DMA

1. 提供软件复位。这将复位所有 GMAC 内部寄存器和逻辑。
2. 等待复位进程完成，复位操作完成后，复位才被清除。
3. 初始化总线模式寄存器。



4. 对 AXI 接口选项进行编程。
5. 创建一个合适的描述符链用于发送和接收。当使用 OSF 模式时，至少需要两个描述符。
6. 在重新使用任何一个描述符之前，确保软件在链中创建三个或更多不同的发送或接收描述符。
7. 使用发送和接收描述符的基址初始化发送和接收描述符列表地址。
8. 清除中断请求。
9. 启用中断。
10. 确认之前的所有 AHB 或 AXI 事务都已完成。
11. 启动接收和发送 DMA。

### 1.5.3 初始化 MAC

下面的 MAC 初始化操作可以在 DMA 初始化之后进行。如果 MAC 初始化是在 DMA 设置之前完成的，那么在 DMA 激活之后再启用 MAC 接收器（下面的最后一步）。否则，接收到的帧将填满 Rx FIFO 并溢出。

1. 控制外部 PHY 的管理周期。
2. 为传入的帧设置适当的过滤器。
3. 设置适当的流量控制。
4. 根据需要对中断屏蔽寄存器位进行编程。如果适用，针对您的配置进行编程。
5. 正常收发操作。
6. 停止和启动发送。

## 1.6 寄存器描述

GMAC 寄存器由两部分组成：APB 寄存器和 AHB 寄存器。

APB 寄存器用于配置时钟频率/时钟启用/时钟方向/PHY 接口类型。

AHB 寄存器用于配置 GMAC 核（DesignWare 核部分）。

### 1.6.1 寄存器内存映射

GMAC AHB 寄存器，如 1.6.3 章所述。

GMAC APB 寄存器，GMAC 寄存器的详细描述如图表 1-6 所示。

图表 1-6 GMAC APB 寄存器内存映射

寄存器名	偏移量	宽度	类型	说明
GMAC_CLK_EN	0x0	32bit	R/W	时钟启用寄存器 复位值：0x8
GMAC_RXCLK_DELAY_CTRL	0x4	32bit	R/W	rx-clk 延迟控制寄存器 复位值：0x8000

寄存器名	偏移量	宽度	类型	说明
GMAC_TXCLK_DELAY_CTRL	0x8	32bit	R/W	tx-clk 延迟控制寄存器 复位值: 0x8000
GMAC_PLLCLK_DIV	0xc	32bit	R/W	PLL 分频器数量寄存器 复位值: 0x4
GMAC_EPHY_DIV	0x10	32bit	R/W	ephy-clk 分频器数量寄存器 复位值: 0x14
GMAC_PTPCLK_DIV	0x14	32bit	R/W	ptp-clk 分频器数量寄存器 复位值: 0x2
GMAC_GTXCLK_SEL	0x18	32bit	R/W	gtx-clk 选择寄存器 复位值: 0x1
GMAC_INTF_CTRL	0x1c	32bit	R/W	GMAC PHY 接口类型寄存器 复位值: 0x0
GMAC_TXCLK_OEN	0x20	32bit	R/W	txclk 输出使能寄存器 复位值: 0x1

## 1.6.2 寄存器和字段描述

GMAC AHB 寄存器, 请参阅 *DesignWare Cores Ethernet GMAC Universal Databook* 第 6.3 章。

GMAC APB 寄存器字段描述如下:

### 1.6.2.1 GMAC\_CLK\_EN

- 寄存器名: GMAC\_CLK\_EN 寄存器
- 偏移量: 0x00
- 默认值: 0x0

图表 1-7 GMAC\_CLK\_EN

位段	名称	类型	说明
31:8	保留		
7	gmac_mdc_clk_inv	R/W	Mdc 时钟反相 0: mdc 时钟 1: 反相 mdc 时钟

位段	名称	类型	说明
			默认值: 0x0
6	gmac_ephy_ref_clk_enable	R/W	GMAC EPHY 参考时钟启用 0: 禁用 1: 启用 默认值: 0x0
5	gmac_rx_clk_n_enable	R/W	gmac rx-clk_n 启用 0: 禁用 1: 启用 默认值: 0x0
4	gmac_rx_clk_enable	R/W	gmac rx-clk 启用 0: 禁用 1: 启用 默认值: 0x0
3	gmac_tx_clk_out_enable	R/W	gmac tx-clk_out (用于 PHY) 启用 0: 禁用 1: 启用 默认值: 0x1
2	gmac_tx_clk_n_out_enable	R/W	gmac tx-clk_n 启用 0: 禁用 1: 启用 默认值: 0x0
1	gmac_tx_clk_enable	R/W	gmac tx-clk (用于 GMAC) 启用 0: 禁用 1: 启用 默认值: 0x0
0	gmac_tx_clk_in_inv	R/W	gmac tx-clk 输入反相 0: tx-clk 1: 反相 tx-clk 默认值: 0x0

### 1.6.2.2 GMAC\_RXCLK\_DELAY\_CTRL

- 寄存器名：GMAC\_RXCLK\_DELAY\_CTRL 寄存器
- 偏移量：0x04
- 默认值：0x8000

图表 1-8 GMAC\_RXCLK\_DELAY\_CTRL

位段	名称	类型	说明
31:16	保留		
ASIC 模式			
15	bypass	R/W	旁路时钟 默认值：0x1
14	invert	R/W	反相时钟 默认值：0x0
13:5	保留		
4:0	delay_ctrl	R/W	delay_ctrl 默认值：0x0
FPGA 模式			
15:14	保留		
13	mmcm reset	R/W	Mmcm 复位，低电平有效。 默认值：0x0
12	ps_start	R/W	开始相移。Ps_start 由低到高，MMCM 开始相移。 默认值：0x0
11:0	ps_cfg	R/W	相移值。将 rx_clk（移到 GMAC），相移值：ps_cfg*18ps。 默认值：0x0

### 1.6.2.3 GMAC\_TXCLK\_DELAY\_CTRL

- 寄存器名：GMAC\_TXCLK\_DELAY\_CTRL 寄存器
- 偏移量：0x08
- 默认值：0x8000

图表 1-9 GMAC\_TXCLK\_DELAY\_CTRL

位段	名称	类型	说明
31:16	保留		
ASIC 模式			
15	bypass	R/W	旁路时钟 默认值: 0x1
14	invert	R/W	反相时钟 默认值: 0x0
13:5	保留		
4:0	delay_ctrl	R/W	delay_ctrl 默认值: 0x0
FPGA 模式			
15:14	保留		
13	mmcm reset	R/W	mmcm 复位, 低电平有效。 默认值: 0x0
12	ps_start	R/W	开始相移。Ps_start 由低到高, mmcm 开始相移。 默认值: 0x0
11:0	ps_cfg	R/W	相移值。将 gmac_phy_clk ( 移到 PHY ) , 相移值: ps_cfg*18ps。 默认值: 0x0

### 1.6.2.4 GMAC\_PLLCLK\_DIV

- 寄存器名: GMAC\_PLLCLK\_DIV 寄存器
- 偏移量: 0x0c
- 默认值: 0x4

图表 1-10 GMAC\_PLLCLK\_DIV

位段	名称	类型	说明
31	gmac_pll_clk_div_en	R/W	PLL 时钟分频器启用 默认值: 0x0 软件必须配置为低电平, 然后配置为高电平。

位段	名称	类型	说明
30:8	Reserved	-	-
7:0	gmac_pll_clk_div_num	R/W	PLL 时钟分频器数量 默认值: 0x4

### 1.6.2.5 GMAC\_EPHY\_DIV

- 寄存器名: GMAC\_EPHY\_DIV 寄存器
- 偏移量: 0x010
- 默认值: 0x14

图表 1-11 GMAC\_EPHY\_DIV

位段	名称	类型	说明
31	gmac_ephy_clk_div_en	R/W	Ephy 时钟分频器启用 默认值: 0x0 软件必须配置为低电平, 然后配置为高电平。
30:8	Reserved	-	-
7:0	gmac_ephy_clk_div_num	R/W	ephy 时钟分频器数量 默认值: 0x14

### 1.6.2.6 GMAC\_PTP\_DIV

- 寄存器名: GMAC\_PTP\_DIV 寄存器
- 偏移量: 0x014
- 默认值: 0x2

图表 1-12 GMAC\_PTP\_DIV

位段	名称	类型	说明
31	gmac_ptp_clk_div_en	R/W	ptp 时钟分频器启用 默认值: 0x0 软件必须配置为低电平, 然后配置为高电平。
30:8	Reserved	-	-
3:0	gmac_ptp_clk_div_num	R/W	ptp 时钟分频器数量 默认值: 0x2

### 1.6.2.7 GMAC\_GTXCLK\_SEL

- 寄存器名: MAC\_GTXCLK\_SEL 寄存器
- 偏移量: 0x18
- 默认值: 0x1

图表 1-13 GMAC\_GTXCLK\_SEL

位段	名称	类型	说明
31:1	保留		
0	gmac_gtxclk_sel	R/W	Gtx-clk 选择 默认值: 0x1 1: clk_after_pll_div 0: tx_clk_in_after_inv_mu

### 1.6.2.8 GMAC\_PHY\_INTF\_SEL

- 寄存器名: GMAC\_INTF\_CTRL 寄存器
- 偏移量: 0x1c
- 默认值: 0x0

图表 1-14 GMAC\_PHY\_INTF\_SEL

位段	名称	类型	说明
31:1	保留		
0	gmac_phy_intf_sel	R/W	GMAC PHY 接口选择 默认值: 0x0 1: RGMII 0: GMII/MII

### 1.6.2.9 GMAC\_TXCLK\_OEN

- 寄存器名: GMAC\_TXCLK\_OEN 寄存器
- 偏移量: 0x20
- 默认值: 0x1

图表 1-15 GMAC\_TXCLK\_OEN

位段	名称	类型	说明
31:1	保留		

位段	名称	类型	说明
0	gmac_txclk_oen	R/W	Tx-clk 输出启用 默认值：0x1 1: tx-clk 作为输入 0: tx_clk 作为输出

## 1.6.3 GMAC 核寄存器描述

GMAC 核寄存器由 DMA 寄存器和 GMAC 寄存器组成。

下面是 DMA 寄存器。

### 1.6.3.1 寄存器 0（总线模式寄存器）

总线模式寄存器为 DMA 建立总线操作模式。

图表 1-16 寄存器 0（总线模式寄存器）

位段	名称	类型	说明
29:28	PRWG	R_W	信道优先级权重 该字段在系统总线的 DMA 通道之间的轮询仲裁期间设置通道 0 的优先级权重。 <ul style="list-style-type: none"> <li>00: 优先级权重为 1。</li> <li>01: 优先级权重为 2。</li> <li>10: 优先级权重为 3。</li> <li>11: 优先级权重为 4。</li> </ul> 当您选择 AV 功能时，除 gmac-AXI 外，所有 DWC_gmac 配置中都存在该字段。否则，该字段为保留字段，并且是只读的（RO）。 复位值：00
27	TXPR	R_W	发送优先级 当设置此位时，表示在系统端总线仲裁期间，发送 DMA 比接收 DMA 具有更高的优先级。在 GMAC-AXI 配置中，该位为保留位，并且是只读的。 复位值：0



位段	名称	类型	说明
26	MB	R_W	<p>混合突发</p> <p>当该位设置为高，而 FB 位设置为低时，AHB 主接口启动所有长度大于 16 的 INCR 突发（未定义突发），而当突发长度小于等于 16 时恢复到固定突发传输（INCRx 和 SINGLE）。该位仅在 GMAC-AHB 配置中有效，在所有其他配置中，该位为保留位。</p> <p>复位值：0</p>
25	AAL	R_W	<p>地址对齐节拍</p> <p>当该位设置高且 FB 位等于 1 时，AHB 或 AXI 接口生成所有突发与起始地址 LS 位对齐。如果 FB 位等于 0，则第一次突发（访问数据缓冲区的起始地址）不对齐，但后续突发都与该地址对齐。</p> <p>该位仅在 GMAC-AHB 和 GMAC-AXI 配置中有效，在其他配置中，该位为保留位（只读，默认值为 0）。</p> <p>复位值：0</p>
24	PBLx8	R_W	<p>PBLx8 模式</p> <p>当设置为高时，该位是编程的 PBL 值（bit[22:17]和 bit[13:8]）的 8 倍。因此，DMA 根据 PBL 的值以 8、16、32、64、128 和 256 拍的方式传输数据。</p> <p>注意：此位函数不向后兼容。在发布 3.50a 之前，该位是 4xPBL。</p> <p>复位值：0</p>
23	USP	R_W	<p>使用单独的 PBL</p> <p>当设置为高时，该位配置 Rx DMA 使用 bit[22:17]中配置的值作为 PBL。bit[13:8]中的 PBL 值只适用于 Tx DMA 操作。</p> <p>当复位为低时，PBL 值在 bit[13:8]中适用于两个 DMA 引擎。</p> <p>复位值：0</p>
22:17	RPBL	R_W	<p>Rx DMA PBL</p> <p>该字段表示在一个 Rx DMA 事务中传输的最大节拍数。这是在单个块读或写中使用的最大值。</p> <p>每次在主机总线上启动突发传输时，Rx DMA 总是尝试按 RPBL 位中指定的值进行突发传输。可以使用值 1、2、4、8、16 和 32 对 RPBL 进行编程。任何其他值都会导致未定义的</p>

位段	名称	类型	说明
			行为。 只有当 USP 设置为高时，该字段才有效。 复位值：01H
16	FB	R_W	固定突发 该位控制 AHB 或 AXI 主接口是否执行固定突发传输。当设置时, AHB 接口在正常突发传输开始时只使用 SINGLE、INCR4、INCR8 或 INCR16。复位值时, AHB 或 AXI 接口使用 SINGLE 和 INCR 突发传输操作。 有关更多信息, 请参阅 GMAC-AXI 配置中 AXI 总线模式寄存器的 bit0 (UNDEF)。 复位值: 0
15:14	PR	R_W	优先级比 这些位控制 Rx DMA 和 Tx DMA 之间加权轮询仲裁的优先级比率。这些位只有在 bit1 (DA) 复位时才有效。优先级比率是 Rx:Tx 或 Tx:Rx, 取决于 bit27 (TXPR) 是否复位或设置。 <ul style="list-style-type: none"> <li>■ 00: 优先级比为 1:1。</li> <li>■ 01: 优先级比为 2:1。</li> <li>■ 10: 优先级比为 3:1。</li> <li>■ 11: 优先级比是 4:1。</li> </ul> 在 GMAC-AXI 配置中, 这些位是保留位, 并且是只读的。 复位值: 00
13:8	PBL	R_W	可编程突发长度 这些位表示在一个 DMA 事务中传输的最大节拍数。这是在单个块读或写中使用的最大值。当 DMA 在主机总线上启动突发传输时, 它总是尝试按 PBL 中指定的值进行突发传输。PBL 可编程为允许值 1、2、4、8、16 和 32。任何其他值都会导致未定义的行为。当 USP 设置高时, 该 PBL 值仅适用于 Tx DMA 事务。 如果需要传输的节拍数大于 32, 请执行以下步骤: 设置 PBLx8 模式 设置 PBL 例如, 如果传输的最大节拍数是 64, 那么首先将 PBLx8 设置为 1, 然后将 PBL 设置为 8。PBL 值有以下限制: 最大可能

位段	名称	类型	说明
			的节拍数 (PBL) 受 MTL 层的 Tx FIFO 和 Rx FIFO 的大小以及 DMA 上的数据总线宽度的限制。FIFO 有一个约束, 即除非另外指定, 支持的最大拍数是 FIFO 深度的一半。 复位值: 01H
7	ATDS	R_W	其他描述符大小 设置后, 备用描述符的大小将增加到 32 字节 (8 DWORDS)。当在接收端启用高级时间戳特性或 IPC 完全校验和卸载引擎 (类型 2) 时, 这是必需的。如果没有启用高级时间戳和 IPC 完全校验和卸载引擎 (类型 2) 特性, 则不需要增强型描述符。在这种情况下, 您可以使用 16 字节的描述符来节省 4 字节内存。 只有在核配置中选择备用描述符特性和以下任一特性时, 该位才会出现: <ul style="list-style-type: none"> <li>■ 高级时间戳特性</li> <li>■ IPC 全校验和卸载引擎 (类型 2) 特性</li> </ul> 否则, 该位是保留位, 并且是只读的。 复位时, 描述符大小恢复到 4 DWORDs (16 字节)。 这个位保留了描述符大小的向后兼容性。在 3.50a 之前的版本中, 普通描述符和增强描述符的描述符大小都是 16 字节。在 3.50a 版本中, 由于高级时间戳特性和 IPC 全校验和卸载引擎 (类型 2) 特性, 描述符大小增加到 32 字节。 复位值: 0
6:2	DSL	R_W	描述符跳过长度 该位指定要在两个未链接的描述符之间跳过的 Word、Dword 或 Lword 的数量 (取决于 32 位、64 位或 128 位总线)。地址跳过从当前描述符的结束开始到下一个描述符的开始。当 DSL 值等于 0 时, DMA 在环模式下将描述符视为连续的。 复位值: 00H
1	DA	R_W	DMA 仲裁方案 该位指定通道 0 的发送和接收路径之间的仲裁方案。 <ul style="list-style-type: none"> <li>■ 0: 与 Rx:Tx 或 Tx:Rx 加权轮询</li> </ul> 路径之间的优先级根据 bit[15:14] (PR) 中指定的优先级和第 27 位 (TXPR) 中指定的优先级权重来确定。

位段	名称	类型	说明
			<p>■ 1: 固定优先级</p> <p>当设置 bit27 (TXPR) 时, 发送路径优先于接收路径。否则, 接收路径优先于发送路径。</p> <p>在 GMAC-AXI 配置中, 这些位是保留位, 并且是只读的。</p> <p>复位值: 0</p>
0	SWR	R_WS_SC	<p>软件复位</p> <p>当设置该位时, MAC DMA 控制器将复位 MAC 的逻辑和所有内部寄存器。在所有 DWC_gmac 时钟域中, 复位操作完成后将自动清除该位。在对 DWC_gmac 的任何寄存器重新编程之前, 该位应该为 0 值。</p> <p>注意:</p> <ul style="list-style-type: none"> <li>■ 软件复位功能仅由该位驱动。寄存器 64 (通道 1 总线模式寄存器) 或寄存器 128 (通道 2 总线模式寄存器) 的 bit0 对软件复位功能没有影响。</li> <li>■ 只有当所有活动时钟域的复位操作都被解除后, 复位操作才算完成。因此, 所有 PHY 输入时钟 (适用于所选的 PHY 接口) 都存在对于软件复位完成很重要。完成软件复位操作的时间取决于最慢的活动时钟的频率。</li> </ul> <p>复位值: 1</p>

图表 1-17 为不同的数据总线宽度和 FIFO 大小提供有效的 PBL 范围(包括 x8 模式)。

#### 说明

- 如果 PBL 对于发送和接收 DMA 都是通用的, 则必须考虑最小的 Rx FIFO 和 Tx FIFO 深度。
- 在图表 1-17 中, 半双工模式下的有效 PBL 范围仅对 Tx FIFO 有效。对于 1000Mbps 模式, 如果图表 1-17 中没有指定 PBL, 则表示支持小于或等于深度的一半的 PBL。

图表 1-17 有效 PBL 范围

数据总线宽度	FIFO 深度	全双工模式下的有效 PBL 范围	半双工模式下的有效 PBL 范围 (仅适用于 Tx FIFO)
32	128Bytes	16 或更低	8 或更低 (仅限 10/100Mbps 模式)
	256Bytes	32 或更低	32 或更低 (仅限 10/100Mbps 模式)
	512Bytes	64 或更低	64 或更低 (仅限 10/100Mbps 模式)
	1KB	128 或更低	<ul style="list-style-type: none"> <li>■ 10/100Mbps 模式下, 128 或更低</li> <li>■ 1000Mbps 模式下, 64 或更低</li> </ul>

数据总线宽度	FIFO 深度	全双工模式下的有效 PBL 范围	半双工模式下的有效 PBL 范围（仅适用于 Tx FIFO）
	2KB 和更高	所有	所有
64	128Bytes	8 或更低	4 或更低（仅限 10/100Mbps 模式）
	256Bytes	16 或更低	16 或更低（仅限 10/100Mbps 模式）
	512Bytes	32 或更低	32 或更低（仅限 10/100Mbps 模式）
	1KB	64 或更低	<ul style="list-style-type: none"> <li>■ 10/100Mbps 模式下，64 或更低</li> <li>■ 1000Mbps 模式下，32 或更低</li> </ul>
	2KB	128 或更低	128 或更低
	4KB 和更高	所有	所有
128	128Bytes	4 或更低	2 或更低（仅限 10/100Mbps 模式）
	256Bytes	8 或更低	8 或更低（仅限 10/100Mbps 模式）
	512Bytes	16 或更低	16 或更低（仅限 10/100Mbps 模式）
	1KB	32 或更低	<ul style="list-style-type: none"> <li>■ 仅在 10/100Mbps 模式下，32 或更低</li> <li>■ 仅在 1000Mbps 模式下，16 或更低</li> </ul>
	2KB	64 或更低	64 或更低
	4KB	128 或更低	128 或更低
	8KB 和更高	所有	所有

### 1.6.3.2 寄存器 1（发送轮询需求寄存器）

发送轮询需求寄存器使 Tx DMA 能够检查 DMA 是否拥有当前描述符。如果 Tx DMA 处于暂停模式，该寄存器发送轮询需求命令来唤醒它。Tx DMA 之所以进入暂停模式是由于传输帧中的下溢错误或其拥有的描述符不可用。您可以在任何时候给出这个命令，当 Tx DMA 再次开始从主机内存中获取当前描述符时，它会重置这个命令。当该寄存器被读取时，它总是返回 0。

图表 1-18 寄存器 1（发送轮询需求寄存器）

位段	名称	类型	说明
31:0	TPD	RO_WT	发送轮询需求 当用任何值写入这些位时，DMA 读取寄存器 18（当前主机发送描述符寄存器）指向的当前描述符。如果该描述符不可用（由主机拥有），发送返回到暂停状态，并断言寄存器 5（状态寄存器）的 bit2 (TU)。如果描述符可用，则恢复发送。 复位值：0000_0000H

### 1.6.3.3 寄存器 2（接收轮询需求寄存器）

接收轮询需求寄存器使 Rx DMA 能够检查新的描述符。这个命令用于从暂停状态唤醒 Rx DMA。Rx DMA 之所以会进入暂停状态，只是因为它所拥有的描述符不可用。当这个寄存器被读取时，它总是返回 0。

图表 1-19 寄存器 2（接收轮询需求寄存器）

位段	名称	类型	说明
31:0	RPD	RO_WT	接收轮询需求 当用任何值写入这些位时，DMA 读取寄存器 19(当前主机接收描述符寄存器)指向的当前描述符。如果描述符不可用(由主机所有)，接收返回到暂停状态，寄存器 5(状态寄存器)的第 7 位(RU)被断言。如果描述符可用，Rx DMA 返回活动状态。 复位值：0000_0000H

### 1.6.3.4 寄存器 3（接收描述符列表地址寄存器）

接收描述符列表地址寄存器指向接收描述符列表的起点。描述符列表驻留在主机的物理内存空间中，并且必须是 Word、Dword 或 Lword 对齐的(用于 32 位、64 位或 128 位数据总线)。DMA 通过使相应的 LS 位低，内部转换它为总线宽度对齐的地址。只有当接收停止时，才允许向这个寄存器写入。该寄存器停止时，必须在发出接收启动命令之前写入该寄存器。

只有当 Rx DMA 停止时，也就是说，寄存器 6(操作模式寄存器)中的第 1 位(SR)设置为 0 时，才可以写入该寄存器。当该寄存器停止时，可以用一个新的描述符列表地址写入该寄存器。

当您将 SR 位设置为 1 时，DMA 接受新编程的描述符基址。

如果当 SR 位设置为 0 时该寄存器没有更改，那么 DMA 将使用之前停止的描述符地址。

图表 1-20 寄存器 3（接收描述符列表地址寄存器）

位段	名称	类型	说明
31:0	RDESLA	R_W	接收列表起始位 该字段包含“接收描述符”列表中第一个描述符的基址。32 位、64 位或 128 位总线宽度的 LSB 位(1:0、2:0 或 3:0)被忽略，并在内部被 DMA 视为全 0。 因此，这些 LSB 位是只读的(RO)。 复位值：0000_0000H

### 1.6.3.5 寄存器 4（发送描述符列表地址寄存器）

发送描述符列表地址寄存器指向发送描述符列表的起点。描述符列表驻留在主机的物理内存空间中，并且必须是 Word、Dword 或 Lword 对齐的(用于 32 位、64 位或 128 位数据总线)。DMA 通过使相应的 LS 位低，内部转换它为总线宽度对齐的地址。

只有当 Tx DMA 停止时，也就是说，在寄存器 6(操作模式寄存器)中的第 13 位(ST)被设为 0 时，才可以写入该寄存器。当该寄存器停止时，可以用一个新的描述符列表地址写入该寄存器。当您将 ST 位设置为 1 时，DMA 接受新编程的描述符基址。

如果在 ST 位置 0 时该寄存器没有发生更改，则 DMA 将使用之前停止的描述符地址。

图表 1-21 寄存器 4（发送描述符列表地址寄存器）

位段	名称	类型	说明
31:0	TDESLA	R_W	传输列表起始位 此字段包含“发送描述符”列表中第一个描述符的基址。32 位、64 位或 128 位总线宽度的 LSB 位(1:0、2:0、3:0)被忽略，并在内部被 DMA 视为全 0。 因此，这些 LSB 位是只读的(RO)。 复位值：0000_0000H

### 1.6.3.6 寄存器 5（状态寄存器）

状态寄存器包含 DMA 报告给主机的所有状态位。软件驱动程序在中断服务程序或轮询期间读取这个寄存器。该寄存器中的大部分字段会导致主机被中断。该寄存器的位在读取时不被清除。写入 1'b1 到该寄存器的位[16:0] (非保留位)将清除这些位；写入 1'b0 则无影响。各个字段(位[16:0])可以通过屏蔽寄存器 7(中断使能寄存器)中的适当位来屏蔽。

图表 1-22 寄存器 5（状态寄存器）

位段	名称	类型	说明
31	-	RO	保留 复位值：0
30	GLPII 或 GTMSI	RO	<p>GLPII: GMAC LPI 中断(用于通道 0)</p> <p>该位表示在 MAC 的 LPI 逻辑中发生了中断事件。为了将该位复位为 1'b0, 软件必须读取 DWC_gmac 中相应的寄存器, 以获得确切的中断原因并清除其来源。</p> <p>注意:</p> <p>GLPII 状态仅在通道 0 DMA 寄存器中给出, 且仅在节能以太网 (Energy Efficient Ethernet) 特性启用时适用。否则, 该位是保留位。当该位为高时, 来自 MAC (sbd_intr_o)的中断信号为高。</p> <p>GTMSI: GMAC TMS 中断(用于通道 1 和通道 2)</p> <p>该位指示 DWC_gmac 的流量管理器和调度器逻辑中的中断事件。要复位该位, 软件必须读取相应的寄存器(通道状态寄存器), 以获得中断的确切原因并清除其来源。</p> <p>注意:</p> <p>GTMSI 状态只在通道 1 和通道 2 DMA 寄存器中给出, 且仅当 AV 特性被启用和相应的附加传输通道存在时适用。否则, 该位是保留位。当该位为高时, 来自 MAC (sbd_intr_o)的中断信号为高。</p> <p>复位值：0</p>
29	TTI	RO	<p>时间戳触发器中断</p> <p>该位表示在 DWC_gmac 的时间戳生成器 (Timestamp Generator) 块中有中断事件。软件必须读取 DWC_gmac 中相应的寄存器以获得中断的确切原因, 并清除其源以将该位复位为 1'b0。当该位高时, 来自 DWC_gmac 子系统 (sbd_intr_o)的中断信号高。</p> <p>只有 IEEE 1588 时间戳特性启用时, 该位才适用。否则, 该位是保留位。</p> <p>复位值：0</p>
28	GPI	RO	<p>GMAC PMT 中断</p> <p>该位表示在 DWC_gmac 的 PMT 模块中发生中断事件。软件必须读取 MAC 中的 PMT 控制和状态寄存器以获得中断的确</p>



位段	名称	类型	说明
			<p>切原因，并清除其源以将该位复位为 1'b0。当该位高时，来自 DWC_gmac 子系统(sbd_intr_o)的中断信号高。</p> <p>只有启用电源管理特性时，该位才有效。否则，该位是保留位。</p> <p>注意：GPI 和 pmt_intr_o 中断在不同的时钟域生成。</p> <p>复位值：0</p>
27	GMI	RO	<p>GMAC MMC 中断</p> <p>该位反映了 DWC_gmac 的 MMC 模块中有中断事件。软件必须读取 DWC_gmac 中相应的寄存器，以获得中断的确切原因，并清除中断源，使该位为 1'b0。当该位高时，来自 DWC_gmac 子系统(sbd_intr_o)的中断信号高。</p> <p>该位仅在 MAC Management Counters (MAC 管理计数器，简称 MMC)启用的情况下适用。否则，该位是保留位。</p> <p>复位值：0</p>
26	GLI	RO	<p>GMAC 线路接口中断</p> <p>当设置此位时，它反映了 DWC_gmac 接口中的任何以下中断事件(如果存在并在您的配置中启用)：</p> <ul style="list-style-type: none"> <li>■ PCS (TBI, RTBI, or SGMII): 链路更改或自动协商完成事件</li> <li>■ SMII or RGMII: 链接变更事件</li> <li>■ 通用输入状态(GPIS): gpi_i 输入端口上的所有 LL 或 LH 事件</li> </ul> <p>为了确定中断的确切原因,软件必须首先读取寄存器 14(中断状态寄存器)的第 11 位和位[2:0], 然后清除中断源(也清除 GLI 中断), 读取以下任何一个相应的寄存器：</p> <ul style="list-style-type: none"> <li>■ PCS (TBI, RTBI 或 SGMII): 寄存器 49 (AN Status Register)</li> <li>■ SMII 或 RGMII: 寄存器 54 (SGMII/RGMII/SMII 控制与状态寄存器)</li> <li>■ 通用输入(GPI): 寄存器 56 (通用 IO 寄存器)</li> </ul> <p>当该位高时,来自 DWC_gmac 子系统(sbd_intr_o)的中断信号高。</p> <p>复位值：0</p>
25:23	EB	RO	错误位

位段	名称	类型	说明
			<p>该字段指示导致总线错误的错误类型。例如，AHB 或 AXI 接口上的错误响应。只有设置了 Bit 13 (FBI) 时，该字段才有效。该字段不生成中断。</p> <ul style="list-style-type: none"> <li>0 0 0: Rx DMA 写数据传输时出错</li> <li>0 1 1: Tx DMA 读数据传输时出错</li> <li>1 0 0: Rx DMA 描述符写访问时出错</li> <li>1 0 1: Tx DMA 描述符写访问时出错</li> <li>1 1 0: Rx DMA 描述符读取访问时出错</li> <li>1 1 1: Tx DMA 描述符读取访问时出错</li> </ul> <p>注意：001 和 010 是保留的。</p> <p>复位值：000</p>
22:20	TS	RO	<p>发送流程状态</p> <p>该字段表示“发送 DMA FSM”状态。该字段不生成中断。</p> <ul style="list-style-type: none"> <li>3'b000: 停止；发出复位或停止发送命令</li> <li>3'b001: 运行；获取发送传输描述符</li> <li>3'b010: 运行；等待状态</li> <li>3'b011: 运行；从主机内存缓冲区读取数据并将其列队至发送缓冲区(Tx FIFO)</li> <li>3'b100: TIME_STAMP 写状态</li> <li>3'b101: 保留至将来使用</li> <li>3'b110: 暂停；发送描述符不可用或发送缓冲区下溢</li> <li>3'b111: 运行；关闭发送描述符</li> </ul> <p>复位值：000</p>
19:17	RS	RO	<p>接收流程状态</p> <p>该字段表示“接收 DMA FSM”状态。该字段不生成中断。</p> <ul style="list-style-type: none"> <li>3'b000: 停止；发出复位或停止接收命令</li> <li>3'b001: 运行；获取接收传输描述符</li> <li>3'b010: 保留至将来使用</li> <li>3'b011: 运行；等待接收数据包</li> <li>3'b100: 暂停；接收描述符不可用</li> <li>3'b101: 运行；关闭接收描述符</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>3'b110: TIME_STAMP 写状态</li> <li>3'b111: 运行: 将接收数据包的数据从接收缓冲区传输到主机内存</li> </ul> 复位值: 000
16	NIS	R_SS_WC	所有正常中断 当寄存器 7(中断使能寄存器)中对应的中断位使能时, 所有正常中断位值是以下位的逻辑或: <ul style="list-style-type: none"> <li>寄存器 5[0]: 发送中断</li> <li>寄存器 5[2]: 发送缓冲区不可用</li> <li>寄存器 5[6]: 接收中断</li> <li>寄存器 5[14]: 早期接收中断</li> </ul> 只有未屏蔽位(在寄存器 7 中设置了中断使能的中断)影响所有正常中断位。 这是一个粘结位, 必须在每次清除对应的位(导致设置 NIS)时清除(通过向该位写入 1)。 复位值: 0
15	AIS	R_SS_WC	所有异常中断 在寄存器 7(中断使能寄存器)中对应的中断位使能时, 所有异常中断位值是下列位的逻辑或: <ul style="list-style-type: none"> <li>寄存器 5[1]: 发送进程停止</li> <li>寄存器 5[3]: 发送 Jabber (超限) 超时</li> <li>寄存器 5[4]: 接收 FIFO 溢出</li> <li>寄存器 5[5]: 发送下溢</li> <li>寄存器 5[7]: 收缓冲区不可用</li> <li>寄存器 5[8]: 接收进程停止</li> <li>寄存器 5[9]: 接收看门狗超时</li> <li>寄存器 5[10]: 早期发送中断</li> <li>寄存器 5[13]: 致命总线错误</li> </ul> 只有未屏蔽位影响所有异常中断位。 是一个粘结位, 必须在每次清除对应的位(导致设置 AIS)时清除(通过向该位写入 1)。 复位值: 0

位段	名称	类型	说明
14	ERI	R_SS_WC	早期接收中断 该位表示 DMA 占满数据包的第一个数据缓冲区。当软件将 1 写入该位或设置该寄存器的第 6 位 (RI)时(以较早发生的为准), 该位被清除。 复位值: 0
13	FBI	R_SS_WC	致命总线错误中断 该位表示发生了总线错误, 如位[25:23]所述。当设置了该位后, 相应的 DMA 引擎将禁用其所有的总线访问。 复位值: 0
12:11	-	RO	保留 复位值: 00
10	ETI	R_SS_WC	早期发送中断 该位表示要发送的帧被完全传输到 MTL 发送 FIFO。 复位值: 0
9	RWT	R_SS_WC	接收看门狗超时 当设置该位时, 表示接收到当前帧时, 接收看门狗计时器过期。看门狗超时后, 当前帧被截位。 复位值: 0
8	RPS	R_SS_WC	接收进程停止 该位在接收进程进入停止状态时断言。 复位值: 0
7	RU	R_SS_WC	接收缓冲区不可用 该位表示主机拥有接收列表中的下一个描述符, DMA 无法获得它。接收进程挂起。要恢复接收描述符的进程, 主机应该更改描述符的所有权, 并发出接收轮询需求命令。如果没有发出接收轮询请求, 接收进程将在接收到下一个可识别的传入帧时恢复。只有在 DMA 拥有前一个接收描述符时才设置此位。 复位值: 0
6	RI	R_SS_WC	接收中断 该位表示帧接收完成。当接收完成时, RDES1 的第 31 位 (在完成时禁用中断)在最后一个描述符中复位, 特定的帧状态信息在描述符中更新。

位段	名称	类型	说明
			接收保持在运行状态。 复位值：0
5	UNF	R_SS_W C	发送下溢 该位表示在帧发送期间发送缓冲区有下溢。发送暂停，下溢错误 TDES0[1]被设置。 复位值：0
4	OVF	R_SS_W C	接收溢出 该位表示在接收帧时接收缓冲区有溢出。如果将部分帧传输给应用程序，则在 RDES0[11]中设置溢出状态。 复位值：0
3	TJT	R_SS_W C	发送 Jabber 超时 该位表示发送 Jabber 计时器过期，当帧大小超过 2048(当 Jumbo 帧启用时为 10,240 字节)时发生。当 Jabber 超时发生时，发送进程被中止并置于停止状态。这将导致发送 Jabber 超时 TDES0[14]标志断言。 复位值：0
2	TU	R_SS_W C	发送缓冲区不可用 该位表示主机拥有发送列表中的下一个描述符，DMA 无法获得它。发送被挂起，位[22:20]解释发送进程状态。 要恢复发送描述符进程，主机应该通过设置 TDES0[31]来更改描述符的所有权，然后发出一个发送轮询需求命令。 复位值：0
1	TPS	R_SS_W C	发送过程中停止 该位在发送停止时设置。 复位值：0
0	TI	R_SS_W C	发送中断 该位表示帧发送完成。发送完成后，复位 TDES0 的第 31 位 (OWN)，描述符中具体的帧状态信息被更新。 复位值：0

### 1.6.3.7 寄存器 6（操作模式寄存器）

操作模式寄存器建立发送和接收操作模式和命令。这个寄存器应该是作为 DMA 初始化的一部分写入的最后一个 CSR。这个寄存器也存在于 GMAC-MTL 配置中，其未使用和保留位为 24、13、2 和 1。

图表 1-23 寄存器 6（操作模式寄存器）

位段	名称	类型	说明
31:27	-	RO	保留 复位值：0H
26	DT	R_W	禁用删除 TCP/IP 校验和错误帧  当设置此位时，MAC 不会丢弃只有接收校验和卸载引擎检测到错误的帧。这样的帧在 MAC 接收到的以太网帧中没有任何错误(包括 FCS 错误)，而只有在封装的负载中有错误。当这个位被复位时，如果 FEF 位被复位，则所有的错误帧都被丢弃。  如果 IPC 全校验和卸载引擎(类型 2)被禁用，该位是保留位(只读，值为 1'b0)。 复位值：0
25	RSF	R_W	接收、存储和转发  当设置该位时，MTL 只在将完整的帧写入到 Rx FIFO 后才从 Rx FIFO 读取帧，忽略 RTC 位。当该位被复位时，Rx FIFO 以直通模式工作，受 RTC 位指定的阈值影响。 复位值：0
24	DFF	R_W	禁用刷新接收帧  当该位被设置时，由于接收描述符或缓冲区不可用，Rx DMA 不会刷新任何帧，而在该位被复位时，它通常会刷新帧。(请参阅第 83 页的《接收进程挂起》)  这个位在 GMAC-MTL 配置中是保留位(只读)。 复位值：0
23	RFA_2	R_W	启动流量控制的阈值 MSB  如果将 DWC_gmac 配置为 Rx FIFO 大于等于 8 KB，这个位(设置后)提供额外的阈值水平，用于在半双工和全双工模式下激活流量控制。该位(最高有效位)和 RFA(位[10:9])给出了以下激活流量控制的阈值：  ■ 100: Full minus 5 KB，即 FULL – 5 KB

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>101: Full minus 6 KB, 即 FULL – 6 KB</li> <li>110: Full minus 7 KB, 即 FULL – 7 KB</li> <li>111: 保留</li> </ul> 如果 Rx FIFO 深度小于或等于 4 KB, 这个位是保留位(只读)。 复位值: 0
22	RFD_2	R_W	停止流量控制的阈值 MSB  如果将 DWC_gmac 配置为 Rx FIFO 大于等于 8 KB, 这个位(设置后)提供额外的阈值水平, 用于在半双工和全双工模式下停止流量控制。该位(作为最高有效位)和 RFD (位[12:11])给出了以下停止流量控制的阈值: <ul style="list-style-type: none"> <li>100: Full minus 5 KB, 即 FULL – 5 KB</li> <li>101: Full minus 6 KB, 即 FULL – 6 KB</li> <li>110: Full minus 7 KB, 即 FULL – 7 KB</li> <li>111: 保留</li> </ul> 如果 Rx FIFO 深度小于或等于 4 KB, 该位是保留位(只读)。 复位值: 0
21	TSF	R_W	发送、存储和转发  设置此位时, 当 MTL 发送 FIFO 中驻留一个完整的帧时, 发送才开始。设置该位后, 将忽略位[16:14]中指定的 TTC 值。只有当发送停止时才应更改此位。  复位值: 0
20	FTF	R_WS_SC	刷新发送 FIFO  当设置该位时, 发送 FIFO 控制器逻辑复位为默认值, 因此 Tx FIFO 中的所有数据都丢失或被刷新。当刷新操作完成时, 该位被内部清除。在清除此位之前, 不应该写入操作模式寄存器。已经被 MAC 发送器接受的数据不会被刷新。这些数据被安排传输, 并且会导致下溢和 runt 帧传输。  注意:  只有当发送 FIFO 的内容被清空, 并且所有待处理的传输帧的传输状态被主机接受时, 刷新操作才会完成。为了完成这个刷新操作, 需要激活 PHY 传输时钟(clk_tx_i)。  复位值: 0
19:17	-	RO	保留

位段	名称	类型	说明
			复位值：000
16:14	TTC	R_W	<p>发送阈值控制</p> <p>这些位控制 MTL 发送 FIFO 的阈值水平。当 MTL 发送 FIFO 内的帧尺寸大于阈值时，发送开始。此外，小于阈值的全帧也被发送。这些位只在第 21 位 (TSF)被复位时使用。</p> <ul style="list-style-type: none"> <li>■ 000: 64</li> <li>■ 001: 128</li> <li>■ 010: 192</li> <li>■ 011: 256</li> <li>■ 100: 40</li> <li>■ 101: 32</li> <li>■ 110: 24</li> <li>■ 111: 16</li> </ul> <p>复位值：000</p>
13	ST	R_W	<p>启动或停止发送命令</p> <p>当设置该位时，发送被放置在运行状态，DMA 在当前位置检查发送列表，寻找要发送的帧。描述符会从列表中的当前位置，即寄存器 4(发送描述符列表地址寄存器)设置的发送列表基址尝试获取，或者从发送之前停止时保留的位置尝试获取。如果 DMA 不拥有当前的描述符，发送进入暂停状态，并设置寄存器 5(状态寄存器)的第 2 位(发送缓冲区不可用)。只有在发送停止时，启动发送命令才有效。如果命令是在设置寄存器 4(发送描述符列表地址寄存器)之前发出的，那么 DMA 行为是不可预测的。</p> <p>复位该位后，当前帧发送完成后，发送进程处于停止状态。发送列表中的下一个描述符位置被保存，当重新启动发送时，该位置变为当前位置。要更改列表地址，您需要在寄存器 4(发送描述符列表地址寄存器)复位此位时使用新值编程。当再次设置该位时，将考虑新值。只有当前帧发送完成或发送处于暂停状态时，停止发送命令才会生效。</p> <p>注意：有关如何暂停发送，请参阅第 715 页的《停止和启动发送》。</p> <p>复位值：0</p>
12:11	RFD	R_W	停止流量控制的阈值(半双工和全双工模式)



位段	名称	类型	说明
			<p>这些位控制阈值(Rx FIFO 的占满级), 在该阈值处流量控制激活后被去断言。</p> <ul style="list-style-type: none"> <li>■ 00: Full minus 1 KB, 即 FULL – 1 KB</li> <li>■ 01: Full minus 2 KB, 即 FULL – 2 KB</li> <li>■ 10: Full minus 3 KB, 即 FULL – 3 KB</li> <li>■ 11: Full minus 4 KB, 即 FULL – 4 KB</li> </ul> <p>只有在断言流量控制后, 取消断言才有效。如果 Rx FIFO 大于等于 8 KB, 一个额外的位(RFD_2)被用于更多的阈值水平, 如第 22 位中所述。当 Rx FIFO 深度小于 4 KB 时, 这些位是保留位且只读。</p> <p>注意: 为了进行适当的流量控制, “RFD_2, RFD”字段的编程值应大于或等于“RFA_2, RFA”字段的编程值。</p> <p>复位值: 00</p>
10:9	RFA	R_W	<p>激活流量控制的阈值(半双工和全双工模式)</p> <p>这些位控制阈值(Rx FIFO 的占满级), 在该阈值处流量控制被激活。</p> <ul style="list-style-type: none"> <li>■ 00: Full minus 1 KB, 即 FULL–1KB</li> <li>■ 01: Full minus 2 KB, 即 FULL–2KB</li> <li>■ 10: Full minus 3 KB, 即 FULL–3KB</li> <li>■ 11: Full minus 4 KB, 即 FULL–4KB</li> </ul> <p>这些值只适用于 4 KB 或以上的 Rx FIFOs, 并且当第 8 位 (EFC)设置为高时。如果 Rx FIFO 大于或等于 8 KB, 一个额外的位(RFA_2)被用于更多的阈值水平, 如第 23 位中所述。当 Rx FIFO 深度小于 4 KB 时, 这些位是保留位且只读。</p> <p>注意: 当 FIFO 恰好是 4 KB 时, 尽管 DWC_gmac 允许编程这些位的值为 11, 软件不应该编程这些位为 2'b11。值 2'b11 表示 FIFO 空状态时流量控制。</p> <p>复位值: 00</p>
8	EFC	R_W	<p>使能 HW 流量控制</p> <p>当设置此位时, 基于 Rx FIFO 占满级的流量控制信号操作被启用。复位时, 流量控制操作被禁用。当 Rx FIFO 小于 4 KB 时, 该位不使用(保留并总是复位)。</p> <p>复位值: 0</p>

位段	名称	类型	说明
7	FEF	R_W	<p>转发错误帧</p> <p>该位被复位时, Rx FIFO 丢弃带有错误状态的帧(CRC 错误、碰撞错误、GMII_ER、巨帧、监控超时或溢出)。然而, 如果帧的开始字节(写)指针已经转移到读控制器端(在阈值模式), 那么帧不会被丢弃。</p> <p>在 GMAC-MTL 配置中, 帧长 FIFO 在核心配置期间也被启用。如果帧的开始字节没有在 ARI 总线上传输(输出), 则 Rx FIFO 丢弃该错误帧。</p> <p>当 FEF 位被设置时, 除 runt 错误帧外的所有帧都被转发到 DMA。如果设置了第 25 位 (RSF), 并且当写入部分帧时 Rx FIFO 溢出, 那么不管 FEF 位设置如何, 该帧都会被丢弃。然而, 如果第 25 位 (RSF)被复位并且当写入部分帧时 Rx FIFO 溢出, 那么部分帧可能被转发到 DMA。</p> <p>注意: 当 FEF 位被复位时, 如果以下配置中 Rx 状态(表 8-6 或表 8-23)给出了巨帧状态, 则巨帧被丢弃:</p> <ul style="list-style-type: none"> <li>■ 不选择 IP 校验和引擎(类型 1)和全校验和卸载引擎(类型 2)。</li> <li>■ 不选择高级时间戳特性, 但选择扩展状态。扩展状态适用于以下特性:</li> </ul> <p>在 GMAC-CORE 或 GMAC-MTL 配置的 L3-L4 过滤器</p> <p>在 GMAC-DMA、GMAC-AHB 或 GMAC-AXI 配置中, 完整校验和卸载引擎(类型 2), 增强描述符格式。</p> <p>复位值: 0</p>
6	FUF	R_W	<p>转发过小的良好帧</p> <p>当设置时, Rx FIFO 转发过小帧(即没有错误但长度小于 64 字节的帧), 包括填充字节和 CRC。</p> <p>当复位时, Rx FIFO 丢弃所有小于 64 字节的帧, 除非因为接收阈值较低, 帧已经被传输, 例如 RTC = 01。</p> <p>复位值: 0</p>
5	DGF	R_W	<p>丢弃巨帧</p> <p>设置后, MAC 丢弃在 Rx FIFO 中接收到的巨帧, 即大于计算出的巨帧限制的帧。当复位时, MAC 不会丢弃 Rx FIFO 中的巨帧。</p> <p>注意: 该位在以下配置中可用, 在 Rx 状态中没有提供巨帧状</p>

位段	名称	类型	说明
			态，并且默认情况下巨帧没有被丢弃 <ul style="list-style-type: none"> <li>■ 在 Rx 中选择 IP 校验和卸载(类型 1)的配置</li> <li>■ 在 Rx 中使用正常描述符格式选择 IPC 完全校验和卸载引擎(类型 2)的配置</li> <li>■ 选择高级时间戳特性的配置</li> </ul> 在所有其他配置中，该位不使用(保留并总是复位)。 复位值：0
4:3	RTC	R_W	接收阈值控制 这两个位控制 MTL 接收 FIFO 的阈值水平。当 MTL 接收 FIFO 内的帧大于阈值时，开始传输(请求)到 DMA。此外，长度小于阈值的全帧被自动传输。 当配置的“Receive FIFO size”为 128 字节时，11 的值不适用。只有当 RSF 位为 0 时，这些位才有效。当 RSF 位为 1 时，这些位将被忽略。 <ul style="list-style-type: none"> <li>■ 00: 64</li> <li>■ 01: 32</li> <li>■ 10: 96</li> <li>■ 11: 128</li> </ul> 复位值：00
2	OSF	R_W	操作第二帧 当设置此位时，它指示 DMA 处理发送数据的第二帧，甚至在获得第一帧的状态之前。 复位值：0
1	SR	R_W	启动或停止接收 设置此位后，接收进程将处于运行状态。DMA 尝试从接收列表中获取描述符并处理传入的帧。从列表中的当前位置尝试获取描述符，即从寄存器 3(接收描述符列表地址寄存器)设置的地址或从接收进程之前停止时保留的位置尝试获取描述符。如果 DMA 不拥有描述符，接收将暂停，并设置寄存器 5(状态寄存器)的第 7 位(接收缓冲区不可用)。只有当接收停止时，启动接收命令才有效。如果该命令在设置寄存器 3(接收描述符列表地址寄存器)之前发出，DMA 的行为是不可预测的。 当该位被清除时，Rx DMA 操作在传输当前帧后停止。接收列

位段	名称	类型	说明
			表中的下一个描述符位置将被保存，并在接收进程重新启动后变为当前位置。只有当接收进程处于“运行中”(等待接收数据包)或“挂起”状态时，停止接收命令才有效。 复位值：0
0	-	RO	保留 复位值：0

### 1.6.3.8 寄存器 7（中断使能寄存器）

中断使能寄存器启用寄存器 5(状态寄存器)报告的中断。将位设置为 1'b1 就可以启动相应的中断。硬件复位或软件复位后，所有中断被禁用。

图表 1-24 寄存器 7（中断使能寄存器）

位段	名称	类型	说明
31:17	-	RO	保留 复位值：0000H
16	NIE	R_W	所有正常中断使能 当设置此位时，所有正常中断被启用。当该位被复位时，所有正常中断被禁用。这个位在寄存器 5(状态寄存器)中启用以下中断： <ul style="list-style-type: none"> <li>■ 寄存器 5[0]：发送中断</li> <li>■ 寄存器 5[2]：发送缓冲区不可用</li> <li>■ 寄存器 5[6]：接收中断</li> <li>■ 寄存器 5[14]：早期接收中断</li> </ul> 复位值：0
15	AIE	R_W	所有异常中断使能 当设置此位时，将启用所有异常中断。当该位被复位时，所有异常中断被禁用。这个位在寄存器 5(状态寄存器)中启用以下中断： <ul style="list-style-type: none"> <li>■ 寄存器 5[1]：发送进程停止</li> <li>■ 寄存器 5[3]：发送 Jabber 超时</li> <li>■ 寄存器 5[4]：接收溢出</li> <li>■ 寄存器 5[5]：发送下溢</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>■ 寄存器 5[7]: 接收缓冲区不可用</li> <li>■ 寄存器 5[8]: 接收进程停止</li> <li>■ 寄存器 5[9]: 接收看门狗超时</li> <li>■ 寄存器 5[10]: 早期发送中断</li> <li>■ 寄存器 5[13]: 致命总线错误</li> </ul> 复位值: 0
14	ERE	R_W	早期接收中断使能 当该位与所有正常中断使能(第 16 位)同时设置时, 早期接收中断被启用。当这个位被复位时, 早期接收中断被禁用。 复位值: 0
13	FBE	R_W	致命总线错误使能 当该位与所有异常中断使能(第 15 位)同时设置时, 致命总线错误中断被启用。当该位被复位时, 致命总线错误使能中断被禁用。 复位值: 0
12:11	-	RO	保留 复位值: 00
10	ETE	R_W	早期发送中断使能 当该位与所有异常中断使能(第 15 位)同时设置时, 早期发送中断被启用。当该位被复位时, 早期发送中断被禁用。 复位值: 0
9	RWE	R_W	接收看门狗超时使能 当该位与所有异常中断使能(第 15 位)同时设置时, 接收看门狗超时中断被启用。当该位被复位时, 接收看门狗超时中断被禁用。 复位值: 0
8	RSE	R_W	接收停止使能 当该位与所有异常中断使能(第 15 位)同时设置时, 接收停止中断被启用。当该位被复位时, 接收停止中断被禁用。 复位值: 0
7	RUE	R_W	接收缓冲区不可用使能

位段	名称	类型	说明
			当该位与所有异常中断使能(第 15 位)同时设置时, 接收缓冲区不可用中断被启用。当该位被复位时, 接收缓冲区不可用中断被禁用。 复位值: 0
6	RIE	R_W	接收中断使能 当该位与所有正常中断使能(第 16 位)同时设置时, 接收中断被启用。当这个位被复位时, 接收中断被禁用。 复位值: 0
5	UNE	R_W	下溢中断使能 当该位与所有异常中断使能(第 15 位)同时设置时, 发送下溢中断被启用。当该位被复位时, 下溢中断被禁用。 复位值: 0
4	OVE	R_W	溢出中断使能 当该位与所有异常中断使能(第 15 位)同时设置时, 接收溢出中断被启用。当该位被复位时, 溢出中断被禁用。 复位值: 0
3	TJE	R_W	发送 Jabber 超时使能 当该位与所有异常中断使能(第 15 位)同时设置时, 发送 Jabber 超时中断被启用。当该位被复位时, 发送 Jabber 超时中断被禁用。 复位值: 0
2	TUE	R_W	发送缓冲区不可用使能 当该位与所有正常中断使能(第 16 位)同时设置时, 发送缓冲区不可用中断被启用。当这个位被复位时, 发送缓冲区不可用中断被禁用。 复位值: 0
1	TSE	R_W	发送停止使能 当该位与所有异常中断使能(第 15 位)同时设置时, 发送停止中断被启用。当该位被复位时, 发送停止中断被禁用。 复位值: 0
0	TIE	R_W	发送中断使能 当该位与所有正常中断使能(第 16 位)同时设置时, 发送中断

位段	名称	类型	说明
			被启用。当这个位被复位时，发送中断被禁用。 复位值：0

#### 说明

每个通道生成的中断通过使用“或功能”进行组合，并作为单个位输出给出。因此，在 AV 模式下，软件必须读取所有通道的 DMA 中断状态寄存器才能获得中断源。

### 1.6.3.9 寄存器 8（丢失帧和缓冲区溢出计数器寄存器）

DMA 维护两个计数器来跟踪接收期间丢失的帧数。该寄存器报告计数器的当前值。计数器用于诊断目的。位[15:0]指示由于主机缓冲区不可用而丢失的帧。位[27:17]指示由于缓冲区溢出条件(MTL 和 MAC)而丢失的帧以及 MTL 丢弃的 runt 帧(小于 64 字节的良好帧)。

图表 1-25 寄存器 8（丢失帧和缓冲区溢出计数器寄存器）

位段	名称	类型	说明
31:29	-	RO	保留 复位值：000
28	OVFCNTOVF	R_SS_RC	FIFO 溢出计数器的溢出位 每当溢出帧计数器(位[27:17])溢出时，即 Rx FIFO 由于帧计数器达到最大值而溢出时，设置该位。在这种情况下，溢出帧计数器被复位为全 0，且该位指示发生翻转。 复位值：0
27:17	OVFFRMCNT	R_SS_RC	溢出帧计数器 此字段指示应用程序丢失的帧数。每当 MTL FIFO 溢出时，这个计数器就会增加计数。当用 mci_be_i[2]在 1'b1 读取该寄存器时，计数器被清除。 复位值：000H
16	MISCNTOVF	R_SS_RC	丢失帧计数器的溢出位 每当丢失帧计数器(位[15:0])溢出时，该位被设置，即丢由于失帧计数器达到最大值，主机接收缓冲区不可用，DMA 丢弃进入的帧。在这种情况下，丢失帧计数器被复位为全 0，且该位指示发生了翻转。 复位值：0
15:0	MISFRMCNT	R_SS_RC	失帧计数器 此字段指示由于主机接收缓冲区不可用而被控制器丢失的帧

位段	名称	类型	说明
			数。每当 DMA 丢弃一个传入的帧时，这个计数器就会增加计数。当用 mci_be_i[0]在 1'b1 读取该寄存器时，计数器被清除。 复位值：0000H

### 1.6.3.10 寄存器 9（接收中断看门狗定时器寄存器）

当写入非零值时，该寄存器启用寄存器 5(状态寄存器)的接收中断(位 6)看门狗定时器。

图表 1-26 寄存器 9(接收中断看门狗定时器寄存器)

位段	名称	类型	说明
31:8	-	RO	保留 复位值：000000H
7:0	RIWT	R_W	RI 看门狗定时器计数 该位指示看门狗定时器设置为系统时钟周期数乘以 256。看门狗定时器在 Rx DMA 完成一个帧的传输后根据编程值被触发。因为相应的描述符 RDES1[31]中的设置，该帧的 RI 状态位没有被设置。当看门狗定时器时间耗尽时，设置 RI 位并停止定时器。根据接收到的任何帧的 RDES1[31]自动设置 RI，当 RI 位设置高时，看门狗定时器复位。 复位值：00H

### 1.6.3.11 寄存器 10（AXI 总线模式寄存器）

AXI 总线模式寄存器控制 AXI 主机的行为。主要用于控制突发分裂和未处理请求的数量。该寄存器只有在 GMAC-AXI 配置中才有效。此外，当 AV 模式中存在多个通道时，此寄存器仅在通道 0 DMA 中有效。

图表 1-27 寄存器 10（AXI 总线模式寄存器）

位段	名称	类型	说明
31	EN_LPI	R_W	使能低电量接口 (LPI) 当设置为 1 时，该位启用 GMAC-AXI 配置支持的 LPI 模式，并接受来自 AXI 系统时钟控制器的 LPI 请求。 当设置为 0 时，该位禁用 LPI 模式，并且总是拒绝来自 AXI 系统时钟控制器的 LPI 请求。 复位值：0
30	LPI_XIT_FRM	R_W	解锁魔法包或远程唤醒帧



位段	名称	类型	说明
			设置为 1 时，仅当接收到魔法包或远程唤醒帧时，该位使 GMAC-AXI 从 LPI 模式退出。 设置为 0 时，当接收到任何帧时，该位使 GMAC-AXI 从 LPI 模式中退出。 复位值：0
29:24	-	RO	保留 复位值：00H
23:20	WR_OSR_LMT	R_W	AXI 最大写未完成请求限制 这个值限制了 AXI 写接口上的最大未完成请求。最大未完成请求 = WR_OSR_LMT+1 注意： <ul style="list-style-type: none"> <li>如果 AXI_GM_MAX_WR_REQUESTS = 4，则保留第 22 位。</li> <li>如果 AXI_GM_MAX_WR_REQUESTS != 16，则保留第 23 位。</li> </ul> 复位值：'h1
19:16	RD_OSR_LMT	R_W	AXI 最大读未完成请求限制 这个值限制了 AXI 读接口上的最大未完成请求。最大未完成请求 = RD_OSR_LMT+1 注意： <ul style="list-style-type: none"> <li>如果 AXI_GM_MAX_RD_REQUESTS = 4，则保留第 18 位。</li> <li>如果 AXI_GM_MAX_RD_REQUESTS != 16，则保留第 19 位。</li> </ul> 复位值：'h1
15:14	-	RO	保留 复位值：00
13	ONEKBBE	R_W	对 GMAC-AXI 主机使能 1 KB 边界跨越 设置后，GMAC-AXI 主机执行不跨越 1 KB 边界的突发传输。 复位时，GMAC-AXI 主机执行不跨越 4 KB 边界的突发传输。 复位值：0
12	AXI_AAL	RO	地址对齐节拍

位段	名称	类型	说明
			该位是只读位，反映寄存器 0(总线模式寄存器)的第 25 位 (AAL)。 当该位设置为 1 时，GMAC-AXI 在读写通道上执行地址对齐的突发传输。 复位值：0
11:8	-	RO	保留 复位值：0H
7	BLLEN256	R_W	AXI 突发长度 256 当该位设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 256 的突发长度。 只有当配置参数 AXI_BL 设置为 256 时，该位存在。否则，该位是保留位，并且是只读(RO)。 复位值：0
6	BLLEN128	R_W	AXI 突发长度 128 当该位设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 128 的突发长度。 只有当配置参数 AXI_BL 设置为 128 或更高时，该位存在。否则，该位是保留位，并且是只读(RO)。 复位值：0
5	BLLEN64	R_W	AXI 突发长度 64 当该位设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 64 的突发长度。 只有当配置参数 AXI_BL 设置为 64 或更高时，该位存在。否则，该位是保留位，并且是只读(RO)。 复位值：0
4	BLLEN32	R_W	AXI 突发长度 32 当该位设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 32 的突发长度。 只有当配置参数 AXI_BL 设置为 32 或更高时，该位存在。否则，该位是保留位，并且是只读(RO)。 复位值：0
3	BLLEN16	R_W	AXI 突发长度 16

位段	名称	类型	说明
			当该位设置为 1 或 UNDEF 设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 16 的突发长度。 复位值：0
2	BLN8	R_W	AXI 突发长度 8 当该位设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 8 的突发长度。 当 UNDEF 设置为 1 时，设置此位无效。 复位值：0
1	BLN4	R_W	AXI 突发长度 4 当该位设置为 1 时，允许 GMAC-AXI 在 AXI 主接口上选择 4 的突发长度。 当 UNDEF 设置为 1 时，设置此位无效。 复位值：0
0	UNDEF	RO	AXI 未定义的突发长度 该位是只读位，表示寄存器 0(总线模式寄存器)中第 16 位(FB)的补码(反码)值。 <ul style="list-style-type: none"> <li>当该位设置为 1 时，允许 GMAC-AXI 执行任何等于或低于位[7:3]编程允许的最大突发长度。</li> <li>当该位设置为 0 时，GMAC-AXI 只允许执行固定的突发长度，如 BLN256、BLN128、BLN64、BLN32、BLN16、BLN8、BLN4 所示，或突发长度 1。</li> </ul> 如果设置了 UNDEF，并且没有设置 BLN 位，则允许 GMAC-AXI 执行 16 的突发长度。 复位值：1

### 1.6.3.12 寄存器 11（AHB 或 AXI 状态寄存器）

该寄存器提供了 AHB 主接口或 AXI 接口的读写通道的活动状态。该寄存器只在 GMAC-AHB 和 GMAC-AXI 配置中存在并有效。该寄存器可用于调试目的。此外，当 AV 模式中存在多个通道时，此寄存器仅在通道 0 DMA 中有效。

图表 1-28 寄存器 11（AHB 或 AXI 状态寄存器）

位段	名称	类型	说明
31:2	-	RO	保留

位段	名称	类型	说明
			复位值：0000_0000H
1	AXIRDSTS	RO	AXI 主读通道状态 当该位为高时，表示 AXI 主机的读通道处于活动状态，正在传输数据。 复位值：0
0	AXWHSTS	RO	AXI 主写通道或 AHB 主状态 当该位为高时，表示在 GMAC-AXI 配置中 AXI 主机的写通道处于活动状态，正在传输数据。在 GMAC-AHB 配置中，表示 AHB 主接口 FSMs 处于非空闲状态。 复位值：0

### 1.6.3.13 寄存器 18（当前主机发送描述符寄存器）

当前主机发送描述符寄存器指向 DMA 读取的当前发送描述符的起始地址。

图表 1-29 寄存器 18（当前主机发送描述符寄存器）

位段	名称	类型	说明
31	AE	R_W	地址使能 设置该位后，地址过滤模块使用第二个 MAC 地址进行完美过滤。 当该位被复位时，地址过滤模块忽略要过滤的地址。 复位值：0
30	SA	R_W	源地址 设置该位后，MAC Address1[47:0]用来与接收帧的 SA 字段进行比较。 当复位该位时，使用 MAC Address1[47:0]与接收帧的 DA 字段进行比较。 复位值：0
29:24	MBC	R_W	屏蔽字节控制 这些位是掩码控制位，用于比较各个 MAC 地址字节。当设置为高时，MAC 不会将接收到的 DA 或 SA 相应字节与 MAC Address1 寄存器的内容进行比较。各个位按如下方式控制字节的屏蔽： ■ 第 29 位：寄存器 18[15:8]

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>第 28 位：寄存器 18[7:0]</li> <li>第 27 位：寄存器 19[31:24]</li> <li>...</li> <li>第 24 位：寄存器 19[7:0]</li> </ul> 您可以通过屏蔽地址的一个或多个字节来过滤一组地址(称为组地址过滤)。 复位值：000000
23:16	-	RO	保留 复位值：00H
15:0	ADDRHI	R_W	MAC Address1 [47:32] 该字段包含第二个 6 字节 MAC 地址的高 16 位(47:32)。 复位值：FFFFH

### 1.6.3.14 寄存器 9（当前主机接收描述符寄存器）

当前主机接收描述符寄存器指向 DMA 读取的当前接收描述符的起始地址。

图表 1-30 寄存器 19（当前主机接收描述符寄存器）

位段	名称	类型	说明
31:0	CURRDESAPTR	RO	主机接收描述符地址指针 复位清除。在操作期间由 DMA 更新的指针。 复位值：0000_0000H

### 1.6.3.15 寄存器 20（当前主机发送缓冲区地址寄存器）

当前主机发送缓冲区地址寄存器指向 DMA 读取的当前发送缓冲区地址。

图表 1-31 寄存器 20（当前主机发送缓冲区地址寄存器）

位段	名称	类型	说明
31:0	CURTBUFAPTR	RO	主机发送缓冲区地址指针 复位清除。在操作期间由 DMA 更新的指针。 复位值：0000_0000H

### 1.6.3.16 寄存器 21（当前主机接收缓冲区地址寄存器）

当前主机接收缓冲区地址寄存器指向 DMA 正在读取的当前接收缓冲区地址。

图表 1-32 寄存器 21（当前主机接收缓冲区地址寄存器）

位段	名称	类型	说明
31:0	CURRBUFAPTR	RO	主机接收缓冲区地址指针 复位清除。在操作期间由 DMA 更新的指针。 复位值：0000_0000H

### 1.6.3.17 寄存器 22（HW 特性寄存器）

该寄存器指示 DWC\_gmac 的可选特性或功能的存在。软件驱动程序可以使用该寄存器来动态地启用或禁用与可选块相关的程序。

图表 1-33 寄存器 22（HW 特性寄存器）

位段	名称	类型	说明
31	-	RO	保留
30:28	ACTPHYIF	RO	激活或选择 PHY 接口 当您的配置中有多个 PHY 接口时，此字段表示在复位去断言期间 phy_intf_sel_i 的采样值。 <ul style="list-style-type: none"> <li>■ 000: GMII 或 MII</li> <li>■ 001: RGMII</li> <li>■ 010: SGMII</li> <li>■ 011: TBI</li> <li>■ 100: RMII</li> <li>■ 101: RTBI</li> <li>■ 110: SMII</li> <li>■ 111: RevMII</li> <li>■ 所有其他：保留</li> </ul>
27	SAVLANINS	RO	源地址或 VLAN 插入
26	FLEXIPSEN	RO	灵活每秒脉冲输出
25	INTTSEN	RO	带有内部系统时间的时间戳
24	ENHDESSEL	RO	备用(增强描述符)

位段	名称	类型	说明
23:22	TXCHCNT	RO	额外 Tx 通道的数量
21:20	RXCHCNT	RO	额外 Rx 通道的数量
19	RXFIFOSIZE	RO	Rx FIFO > 2,048 字节
18	RXTYP2COE	RO	Rx 中的 IP 校验和卸载(类型 2)
17	RXTYP1COE	RO	Rx 中的 IP 校验和卸载(类型 1) 注 意 : 如 果 IPCHKSUM_EN = Enabled 且 IPC_FULL_OFFLOAD = Enabled, 那么 RXTYP1COE = 0 且 RXTYP2COE = 1。
16	TXCOESEL	RO	Tx 中的校验和卸载
15	AVSEL	RO	AV 特性
14	EESEL	RO	节能以太网
13	TSVER2SEL	RO	IEEE 1588-2008 高级时间戳
12	TSVER1SEL	RO	仅 IEEE 1588-2002 时间戳
11	MMCSEL	RO	RMON 模块
10	MGKSEL	RO	PMT 魔法包
9	RWKSEL	RO	PMT 远程唤醒帧
8	SMASEL	RO	SMA (MDIO)接口
7	L3L4FLTREN	RO	Layer 3 和 Layer 4 特性
6	PCSSEL	RO	PCS 寄存器(TBI, SGMII, or RTBI PHY 接口)
5	ADDMACADRSEL	RO	多 MAC 地址寄存器
4	HASHSEL	RO	HASH 过滤器
3	EXTHASHEN	RO	扩展的 DA Hash 过滤器
2	HDSEL	RO	支持半双工
1	GMIISEL	RO	支持 1000 Mbps
0	MIISEL	RO	支持 10 或 100 Mbps

### 1.6.3.18 寄存器 76（通道 1 插槽功能控制和状态寄存器）

该寄存器控制槽比较特性，通道 1 发送 DMA 使用该特性从系统内存中提取缓冲区数据。

图表 1-34 寄存器 76（通道 1 插槽功能控制和状态寄存器）

位段	名称	类型	说明
31:20	-	RO	保留 复位值：00H
19:16	RSN	RO	引用槽位号 该字段给出了 DMA 中用于比较检查的参引用槽位号的当前值。 复位值：0H
15:2	-	RO	保留 复位值：000H
1	ASC	R_W	推进插槽检查 设置此位时，当发送描述符中编程的槽位号(SLOTNUM)为下列两种情况时，该位使 DMA 能够从缓冲区提取数据： <ul style="list-style-type: none"> <li>■ 等于位[19:16]中给出的引用槽位号-或-</li> <li>■ 在引用槽位号前面最多两个槽位。该位仅在设置了位 0 (ESC)时适用。</li> </ul> 复位值：0
0	ESC	R_W	插槽比较使能 当设置时，该位允许使用位[19:16]中给出的当前引用检查发送描述符中编程的槽位号。当槽位号等于引用槽位号或比引用槽位号提前一个槽位时，DMA 才从相应的缓冲区提取数据。 复位时，此位禁用槽位号的检查。DMA 在描述符被处理后立即获取数据。 复位值：0

### 1.6.3.19 寄存器 88（通道 1 CBS 控制寄存器）

该寄存器在流量管理器中控制信用整形调度算法来调度发送的帧。只有当您在 AV 模式中选择发送通道 1 时，该寄存器才会出现。



图表 1-35 寄存器 88 (通道 1 CBS 控制寄存器)

位段	名称	类型	说明
31:18	-	RO	保留 复位值: 00H
17	ABPSSIE	R_W	每槽平均位中断使能  设置该位时, 当通道 1 的每槽状态的平均位更新(寄存器 89 的位 17 (ABSU)时, MAC 断言一个中断(sbd_intr_o 或 mci_intr_o)。当该位被清除时, 就不会为这样的事件断言中断。  复位值: 0
16:7	-	RO	保留 复位值: 00H
6:4	SLC	R_W	插槽数量  软件可以编程插槽的数量(持续时间 125 微秒)。当启用信用整形调度算法时, 通道 1 需要计算每个插槽的平均发送位(在 CBS 状态寄存器中提供)。编码方式如下:  <ul style="list-style-type: none"> <li>■ 3'b000: 1 Slot</li> <li>■ 3'b001: 2 Slots</li> <li>■ 3'b010: 4 Slots</li> <li>■ 3'b011: 8 Slots</li> <li>■ 3'b100: 16 Slots</li> <li>■ 3'b101-3'b111: 保留</li> </ul> 复位值: 00
3:2	-	RO	保留 复位值: 00
1	CC	R_W	信用控制  复位时, 当通道 1 中信用为正数且没有帧发送时, 信用整形调度算法逻辑中的累积信用参数设为零。当通道 1 中没有等待帧, 而其他通道正在发送时, 不积累信用。  设置时, 当通道 1 中信用为正数且没有帧发送时, 信用整形调度算法逻辑中的累积信用参数不复位为零。即使在通道 1 中没有等待帧, 而另一个信道正在发送时, 信用也会累积。  复位值: 0

位段	名称	类型	说明
0	CBSD	R_W	信用整形调度禁用 设置该值时，MAC 将对通道 1 的流量禁用信用整形调度算法，并使流量管理算法对通道 1 的优先级严格高于通道 0。 复位后，信用整形调度算法对通道 1 中的发送流量进行调度。 复位值：0

### 1.6.3.20 寄存器 89（通道 1 CBS 状态寄存器）

该寄存器提供通道 1 发送的平均流量。只有在 AV 模式中选择发送通道 1 时，该寄存器才会出现。

图表 1-36 寄存器 89（通道 1 CBS 状态寄存器）

位段	名称	类型	说明
31:18	-	RO	保留 复位值：000H
17	ABSU	R_SS_SC	ABS 更新 当设置该值时，表示 MAC 更新了 ABS 值。当应用程序读取 ABS 值时，该位被清除。 复位值：0
16:0	ABS	RO	每个插槽平均位数 该字段包含每个槽位的平均发送位。该字段是根据通道 1 通信量的编程槽位数(CBS 控制寄存器中的 SLC 位)计算的。100 Mbps 的最大值为 0x30D4，1000 Mbps 的最大值为 0x1E848。 复位值：000H

### 1.6.3.21 寄存器 90（通道 1 idleSlopeCredit 寄存器）

该寄存器为通道 1 上的 AV 流量提供分配的带宽。只有当 AV 模式中选择发送通道 1 时，该寄存器才会出现。

图表 1-37 寄存器 90（通道 1 idleSlopeCredit 寄存器）

位段	名称	类型	说明
31:14	-	RO	保留 复位值：000H

位段	名称	类型	说明
13:0	ISC	R_W	idleSlopeCredit 该字段包含通道 1 信用整形调度算法所需的 idleSlopeCredit 值。这是当信用增加时，每个周期的信用变化率(100 Mbps 和 1000 Mbps 分别为 40ns 和 8ns)。该软件应编程该字段为计算出的每周期位的信用缩放 1024。最大值为 portTransmitRate，即 1000 Mbps 模式下 0x2000，100Mbps 模式下 0x1000。 复位值：000H

### 1.6.3.22 寄存器 91（通道 1 sendSlopeCredit 寄存器）

该寄存器为其他通道上的 AV 流量提供可用的带宽。只有在 AV 模式中选择发送通道 1 时，该寄存器才会出现。

图表 1-38 寄存器 91（通道 1 sendSlopeCredit 寄存器）（续）

位段	名称	类型	说明
13:0	SSC	R_W	sendSlopeCredit 该字段包含通道 1 信用整形调度算法所需的 sendSlopeCredit 值。这是信用下降时，每个周期的信用变化率(100 Mbps 和 1000 Mbps 分别为 40ns 和 8ns)。该软件应编程该字段为计算出的每周期位的信用缩放 1024。最大值为 portTransmitRate，即 1000 Mbps 模式下 0x2000，100 Mbps 模式下 0x1000。这个字段应该用绝对的 sendSlopeCredit 值来编程。当选择用通道 1 发送时，信用整形调度逻辑从累积的信用中减去该值。 复位值：000H

### 1.6.3.23 寄存器 92（通道 1 hiCredit 寄存器）

该寄存器提供了信用整形调度算法的信用参数中通道 1 可积累的最大值。只有在 AV 模式中选择发送通道 1 时，该寄存器才会出现。

图表 1-39 寄存器 92（通道 1 hiCredit 寄存器）

位段	名称	类型	说明
31:29	-	RO	保留 复位值：0H

位段	名称	类型	说明
28:0	HC	R_W	hiCredit 此字段包含通道 1 的信用整形调度算法所需的 hiCredit 值。这是信用参数中可以累积的最大值。这是按 1024 缩放的位指定的。最大值为 maxInterferenceSize，最优最大帧的尺寸为 16,384 字节或 131,072 比特。取值为 $131,072 * 1,024 = 134,217,728$ 或 0x0800_0000。 复位值：0000000H

### 1.6.3.24 寄存器 93（通道 1 loCredit 寄存器）

该寄存器提供了信用整形调度算法的信用参数中通道 1 可积累的最小值。只有在 AV 模式中选择发送通道 1 时，该寄存器才会出现。

图表 1-40 寄存器 93（通道 1 loCredit 寄存器）

位段	名称	类型	说明
31:29	-	RO	保留 复位值：0x7
28:0	LC	R_W	loCredit 该字段包含通道 1 信用整形调度算法所需的 loCredit 值。这是信用参数中可以累积的最小值。这是按 1024 缩放的位指定的。最大值为 maxInterferenceSize，即最优最大帧的尺寸为 16,384 字节或 131,072 位。取值为 $131,072 * 1,024 = 134,217,728$ 或 0x0800_0000。编程值为 2 的补码(负数)，即 0xF800_0000。 复位值：0x1FFF_FFFF

### 1.6.3.25 寄存器 0（MAC 配置寄存器）

MAC 配置寄存器建立接收和发送操作模式。

图表 1-41 寄存器 0（MAC 配置寄存器）

位段	名称	类型	说明
31	-		0 复位值：RO
30:28	SARC	R_W	源地址插入或替换控件 该字段控制所有传输帧的源地址插入或替换。位 30 根据位

位段	名称	类型	说明
			<p>[29:28] 的值指定用于源地址插入或替换的 MAC 地址寄存器 (0 或 1):</p> <p>2'b0x: 输入信号 mti_sa_ctrl_i 和 ati_sa_ctrl_i 控制 SA 字段的生成。</p> <p>2'b10:</p> <p>如果位 30 为 0, 则 MAC 会在所有传输帧的 SA 字段中插入 MAC 地址 0 寄存器(寄存器 16 和 17)的内容。</p> <p>如果位 30 配置为 1, 并且在核心配置时勾选“启用 MAC 地址寄存器 1 (Enable MAC Address Register 1)”选项, 则 MAC 会在所有传输帧的 SA 字段中插入 MAC Address 1 寄存器(寄存器 18 和 19)的内容。</p> <p>2'b11:</p> <p>如果位 30 为 0, 则 MAC 替换所有传输帧的 SA 字段中 MAC 地址为 0 的寄存器(即寄存器 16 和 17)的内容。</p> <p>如果位 30 设置为 1, 并且在核心配置时勾选“使能 MAC 地址寄存器 1”选项, 则 MAC 替换所有传输帧的 SA 字段中 MAC 地址 1 寄存器(寄存器 18 和 19)的内容。</p> <p>对该字段的更改仅在帧开始时生效。如果在传输帧时写入该寄存器字段, 则只有后续帧可以使用更新后的值, 即当前帧不使用更新后的值。</p> <p>在核心配置时, 如果没有选择 TX 特性的 Enable SA, VLAN 和 CRC Insertion, 这些位是保留位, 且只读。</p> <p>复位值: 0</p>
27	TWOKPE	R_W	<p>IEEE 802.3as 支持 2K 报文</p> <p>当设置此参数时, MAC 将所有长度不超过 2000 字节的帧视为正常报文。</p> <p>当不设置位 20 (JE)时, MAC 将接收到的所有超过 2K 字节的帧视为巨帧。当该位被复位并且位 20 (JE)没有被设置时, MAC 将所有接收到的大于 1518 字节(对于 tagged 数据帧是 1522 字节)的帧都视为巨帧。当设置位 20 时, 设置该位对巨帧状态没有影响。</p> <p>该位和位 20 的设置对巨帧状态的影响请参见表 6-31。</p> <p>复位值: 00</p>
26	SFTERR	R_W	<p>SMII 强制发送错误</p> <p>当设置此位时, 该位指示 PHY 在发送中的 SMII 帧中有强制</p>

位段	名称	类型	说明
			发送错误。如果在核心配置时没有选择 SMII PHY 端口，则保留此位。 复位值：0
25	CST	R_W	类型帧的 CRC 去除 当设置此位时，所有以太类型的帧(长度/类别域于等于 1,536)的最后 4 个字节(FCS)在转发给应用程序之前被去除和丢弃。当 MAC 接收端开启 IP 校验引擎(类型 1)时，该功能无效。当启用类型 2 校验和卸载引擎时，此功能有效。 注意：关于位 7 (ACS)和该位的设置对帧长度的影响，请参见表 6-39（原编号为表 6-32，缺失）。 复位值：0
24	TC	R_W	RGMII, SGMII 或 SMII 的传输配置 当设置此位时，该位使能信息驱动到 PHY，包括 RGMII、SMII 或 SGMII 端口的传输双工模式、链路速度和链路断开信息。当该位被复位时，这些信息不被驱动到 PHY。如果核心配置时没有选择 RGMII / SMII / SGMII PHY 端口，该位为保留位（只读）。 这个特性的细节在以下章节中详述： <ul style="list-style-type: none"> <li>■ 《降低千兆介质独立接口》</li> <li>■ 《串行介质独立接口》</li> <li>■ 《串行千兆介质独立接口》</li> </ul> 复位值：0
23	WD	R_W	看门狗禁用 设置该位后，MAC 关闭接收端的看门狗定时器。MAC 最多可以接收 16,383 字节的帧。 当该位被复位时，MAC 不允许接收超过 2,048 字节(如果 JE 设置高，则为 10,240 字节)的帧或超过寄存器 55(看门狗超时寄存器)中编程值的帧。 看门狗限制字节数后，MAC 将切断所有接收到的字节。 复位值：0
22	JD	R_W	Jabber 禁用 设置该位后，MAC 在发送器上禁用 jabber 定时器。MAC 最多可以传输 16,383 字节的帧。

位段	名称	类型	说明
			<p>当该位被复位时，如果应用程序在传输期间发送超过 2,048 字节的数据(如果 JE 设置高，则为 10,240 字节)，MAC 将切断发送器。</p> <p>复位值：0</p>
21	BE	R_W	<p>帧突发使能</p> <p>当设置该位时，MAC 允许 GMII 半双工模式传输时的帧突发。纯 10/ 100 Mbps 或纯全双工配置时，该位是保留位，且只读。</p> <p>复位值：0</p>
20	JE	R_W	<p>Jumbo 帧使能</p> <p>设置该位后，MAC 允许接收 9,018 字节的 Jumbo 帧(对于 VLAN tagged 帧为 9,022 字节)，而不在接收帧状态中报告巨帧错误。</p> <p>复位值：0</p>
19:17	IFG	R_W	<p>帧间距 (IFG)</p> <p>这些位控制传输过程中帧间的最小帧间距。</p> <ul style="list-style-type: none"> <li>■ 000: 96 位时间</li> <li>■ 001: 88 位时间</li> <li>■ 010: 80 位时间</li> <li>■ ...</li> <li>■ 111: 40 位时间</li> </ul> <p>在半双工模式下，只能配置最小 IFG 为 64 位时间(IFG = 100)，不考虑更低的值。1000-Mbps 模式下，GMAC-CORE 配置支持的最小 IFG 为 64 位时间（及以上），其他配置支持的最小 IFG 为 80 位时间（及以上）。</p> <p>当由于背压激活而传输 JAM 模式时，MAC 不考虑最小帧间距。</p> <p>复位值：000</p>
16	DCRS	R_W	<p>在传输过程中禁用载波侦听</p> <p>当设置高时，在半双工模式下，该位使 MAC 发送器在帧传输过程中忽略(G)MII CRS 信号。此请求不会导致传输过程中由于载波丢失或无载波而产生的错误。当该位设置为低时，由于载波侦听，MAC 发送器产生错误，甚至可能中止传输。</p>

位段	名称	类型	说明
			这个位在全双工配置中是保留位，且只读。 复位值：0
15	PS	R_W	端口选择 该位用于选择以太网线路速度。 <ul style="list-style-type: none"> <li>0: 1000 Mbps 操作</li> <li>1: 10 或 100 Mbps 操作</li> </ul> 在 10 Mbps 或 100 Mbps 操作中，该位与 FES 位一起选择确切的线路速度。在纯 10/ 100 Mbps (始终为 1)或纯 1000 Mbps (始终为 0)配置中，该位具有适当值，且只读。在默认的 10/100/ 1000mbps 配置中，该位为 R_W。mac_portselect_o 或 mac_speed_o[1]信号反映该位的值。 复位值：0
14	FES	R_W	速度 该位选择 MII、RMII、SMII、RGMII、SGMII 或 RevMII 接口的速度： <ul style="list-style-type: none"> <li>0: 10 Mbps</li> <li>1: 100 Mbps</li> </ul> 该位置默认为保留且只读，只有在设置参数 SPEED_SELECT = Enabled 时才会启用。当 RGMII、SMII 或 SGMII 模式下位 24 (TC)被设置时，该位产生链路速度编码。该位对于 RGMII、SGMII、SMII 或 RevMII 接口总是启用的。 在配置 RGMII、SGMII、SMII 或 RevMII 接口时，该位被驱动为输出信号(mac_speed_o[0])，以反映 mac_speed_o 信号中该位的值。在配置 RMII，MII 或 GMII 接口时，可以选择性驱动该位作为输出信号(mac_speed_o[0])，以反映其在 mac_speed_o 信号中的值。 复位值：0
13	DO	R_W	禁用接收自身帧 设置此位，在半双工模式下当 phy_tx_o 被断言时，MAC 禁用接收帧。 当该位被复位时，MAC 接收 PHY 在发送时给出的所有数据包。 如果 MAC 在全双工模式下工作，则此位不适用。如果 MAC 配置为纯全双工操作，则保留该位(只读，默认值)。



位段	名称	类型	说明
			复位值：0
12	LM	R_W	<p>环回模式</p> <p>当设置该位时,MAC在 GMII 或 MII 上以环回模式运行。(G)MII 接收时钟输入(clk_rx_i)是环回正常工作所必需的, 因为传输时钟没有内部环回。</p> <p>复位值：0</p>
11	DM	R_W	<p>双工模式</p> <p>设置该位后, MAC 在全双工模式下工作, 可以同时发送和接收。该位为只读, 在纯全双工模式下默认值为 1'b1。</p> <p>复位值：0</p>
10	IPC	R_W	<p>校验和卸载</p> <p>当设置该位时, MAC 计算收到的所有以太网帧负载的补码和的 16 位补码。它还检查接收帧的 IPv4 头部校验和(假设为接收以太网帧的字节 25-26 或 29-30 (VLANtagged))是否正确, 并在接收状态字中给出状态。MAC 还为 IP 头部数据报负载(字节在 IPv4 报头之后)附加计算出的 16 位校验和, 并将其附加至传输到应用程序的以太网帧(当取消选择 Type 2 COE 时)。</p> <p>当该位被复位时, 该功能被关闭。</p> <p>当选择 Type 2 COE 时, 此位设置后, 将启用 IPv4 头部校验和检查以及 IPv4 或 IPv6 TCP、UDP 或 ICMP 负载校验和检查。复位该位时, 接收端的 COE 功能被关闭, 对应的 PCE 和 IP HCE 状态位(138 页表 3-10 所示 (表 3-10 缺失))总是被清除。</p> <p>如果在核心配置时没有开启 IP 校验和卸载特性, 则保留该位(只读, 默认值)。</p> <p>复位值：0</p>
9	DR	R_W	<p>禁用重试</p> <p>当设置该位时, MAC 只尝试一次传输。当 GMII 或 MII 接口发生碰撞时, MAC 会忽略当前帧的传输, 并在传输帧状态报告因过度碰撞错误中止帧。</p> <p>当该位被复位时, MAC 将根据 BL 字段(位[6:5])的设置进行重试。该位只在半双工模式下有效, 纯全双工模式下为保留(只读, 默认值)。</p>

位段	名称	类型	说明
			复位值：0
8	LUD	R_W	<p>连接正常或断开</p> <p>该位表示在 RGMII、SGMII 或 SMII 接口传输配置时链路是正常还是断开：</p> <ul style="list-style-type: none"> <li>■ 0：链路断开</li> <li>■ 1：连接正常</li> </ul> <p>该位为保留(只读，默认值)。在核心配置时，当 RGMII、SGMII 或 SMII 接口使能时，该位被启用。</p> <p>复位值：0</p>
7	ACS	R_W	<p>自动 Pad 或 CRC 去除</p> <p>设置该位后，只有当长度字段小于 1,536 字节时，MAC 才会去除传入帧的 Pad 或 FCS 字段。所有接收到的长度大于或等于 1,536 字节的帧被传递给应用程序，而不去除 Pad 或 FCS 字段。</p> <p>当该位被复位时，MAC 将所有传入的帧不加修改地传递给主机。</p> <p>注意：关于位 25 (CST)和该位的设置对帧长度的影响，请参见表 6-39（原编号为表 6-32，缺该表）。</p> <p>复位值：0</p>
6:5	BL	R_W	<p>后退限制</p> <p>后退限制决定了时隙延迟的随机整数(r)(1000 Mbps 为 4,096 位时间，10/100 Mbps 为 512 位时间)，MAC 在冲突后重试时，在尝试重新调度传输之前需等待上述时隙延迟时间。该位只在半双工模式下使用，纯全双工模式下为保留(只读)。</p> <ul style="list-style-type: none"> <li>■ 00: <math>k = \min(n, 10)</math></li> <li>■ 01: <math>k = \min(n, 8)</math></li> <li>■ 10: <math>k = \min(n, 4)</math></li> <li>■ 11: <math>k = \min(n, 1)</math></li> </ul> <p>where <math>n</math> = 重传尝试。随机整数 <math>r</math> 取值范围为 <math>0 \leq r &lt; 2^k</math></p> <p>复位值：00</p>
4	DC	R_W	<p>延迟检查</p> <p>当设置该位时，在 MAC 上开启延迟检查功能。当发送状态机</p>

位段	名称	类型	说明
			<p>在 10 Mbps 或 100 Mbps 模式下延迟超过 24,288 位时间的时候，MAC 会发出一个帧中止状态，并且过度延迟错误位在传输帧状态中被设置。</p> <p>当 MAC 配置为 1000 Mbps 操作，或在 10 Mbps 或 100 Mbps 模式下使能 Jumbo 帧模式时，延迟阈值为 155,680 位时间。延迟开始于发射机准备发射时，但由于 GMII 或 MII 上的有源载波检测信号（CRS），延迟被阻止。</p> <p>延迟时间不累加。例如，如果发送器因为 CRS 信号活跃而延迟 10,000 位时间，接着 CRS 信号变得不活跃，发送器发送，碰撞发生。由于碰撞，发送器需要重试，然后重试完成后再次延迟。在这种情况下，延迟计时器被复位为 0 并重新启动。</p> <p>当该位被复位时，延迟检查功能被关闭，MAC 延迟直到 CRS 信号失效。该位只在半双工模式下适用，纯全双工模式下为保留(只读)。</p> <p>复位值：0</p>
3	TE	R_W	<p>发送器使能</p> <p>当设置该位时，MAC 的传输状态机在 GMII 或 MII 上启用传输。当复位该位时，当前帧传输完成后，MAC 传输状态机被禁用，不再传输任何帧。</p> <p>复位值：0</p>
2	RE	R_W	<p>接收器使能</p> <p>当设置该位时，MAC 的接收状态机接收来自 GMII 或 MII 的帧。当该位被复位时，MAC 接收状态机在接收完当前帧后被禁用，不再从 GMII 或 MII 接收任何新的帧。</p> <p>复位值：0</p>
1:0	PRELEN	R_W	<p>传输帧的前导码长度</p> <p>这些位控制添加到每个传输帧开头的前导字节的数量。只有当 MAC 在全双工模式下工作时，才会发生前导码减少。</p> <ul style="list-style-type: none"> <li>■ 2'b00: 前导码的 7 字节</li> <li>■ 2'b01: 前导码的 5 字节</li> <li>■ 2'b10: 前导码的 3 字节</li> <li>■ 2'b11: 保留</li> </ul> <p>复位值：00</p>

寄存器 0 (MAC 配置寄存器)的第 27 位和第 20 位设置对巨帧状态的影响如图表 1-42 所示。

图表 1-42 基于第 27 位和第 20 位的巨帧状态

长度/类别域	接收到的帧长度	位 27 (TWOKPE)	位 20 (JE)	巨帧状态
不带标签的数据包	> 1,518	0	0	1
	> 2,000	1	0	1
	> 9,018	x	1	1
带 VLAN 标签的数据包	> 1,522	0	0	1
	> 2,000	1	0	1
	> 9,022	x	1	1
注意:对于所有其他组合,巨型帧状态为 0。				

寄存器 0 (MAC 配置寄存器)的第 7 位和第 25 位设置对帧长是否包含 CRC 长度的影响如表 6 39 (缺该表) 所示。

### 1.6.3.26 寄存器 1 (MAC 帧过滤器)

MAC 帧过滤寄存器包含接收帧的过滤控件。该寄存器的一些控件转到 MAC 的地址检查块, 执行第一级地址过滤。第二级过滤是在传入帧上执行的, 基于其他控件, 如传递坏帧和传递控制帧。

图表 1-43 寄存器 1 (MAC 帧过滤器)

位段	名称	类型	说明
31	RA	R_W	接收所有 设置此位时, MAC 接收器模块将所有接收到的帧传递给应用程序, 而不管它们是否通过地址过滤器。SA 或 DA 过滤的结果在接收状态字的相应位中更新 (通过或失败)。 当该位被复位时, 接收器模块只将通过 SA 或 DA 地址过滤器的帧传递给应用程序。 复位值: 0
30:22	-	RO	保留 复位值: 00H
21	DNTU	R_W	丢弃 IP 帧上的非 TCP 或 UDP 当设置此位时, 允许 MAC 丢弃 IP 帧上的非 TCP 或 UDP 帧。 MAC 只转发那些经过第 4 层过滤器处理的帧。 当复位时, 该位使 MAC 能够转发 IP 帧上所有的非 TCP 或

位段	名称	类型	说明
			UDP。 如果核心配置时没有选择第 3 层、第 4 层过滤特性，则保留该位(只读，默认值)。 复位值：0
20	IPFE	R_W	第 3 层、第 4 层过滤器使能 当设置此位时，允许 MAC 丢弃与已启用的第 3 层和第四层过滤器不匹配的帧。如果没有启用第 3 层或第 4 层过滤器进行匹配，则此位没有任何作用。 当复位时，MAC 转发所有帧，而不考虑第 3 层和第 4 层字段的匹配状态。 如果核心配置时没有选择第 3 层、第 4 层过滤特性，则保留该位(只读，默认值)。 复位值：0
19:17	-	RO	保留 复位值：000
16	VTFE	R_W	VLAN Tag 过滤器使能 设置该值时，使 MAC 丢弃与 VLAN Tag 比较不匹配的 VLAN tagged 帧。MAC 复位时，不管与 VLAN Tag 的匹配状态如何，都转发所有帧。 复位值：0
15:11	-	R_W	保留 复位值：00
10	HPF	R_W	Hash 或完美过滤器 设置此位时，如果地址过滤器与 HMC 或 HUC 位设置的完美过滤或 Hash 过滤相匹配，则将其配置为通过一个帧。 当该位设置为低且设置了 HUC 或 HMC 位时，仅当该帧与 hash 过滤器匹配时才传递该帧。如果在核心配置过程中没有选择 Hash 过滤器，这个位是保留位(只读)。 复位值：0
9	SAF	R_W	源地址过滤器启用 设置该位后，MAC 将接收到的帧的 SA 字段与使能的 SA 寄存器中的编程值进行比较。如果比较失败，则 MAC 丢弃帧。

位段	名称	类型	说明
			<p>当该位复位后，MAC 根据 SA 地址比较的 Rx 状态的 SAF 更新位将接收到的帧转发给应用程序。</p> <p>注意：根据 IEEE 规范，SA 的第 47 位是保留位，并设置为 0。然而，在 DWC_gmac 中，MAC 比较所有的 48 位。软件驱动程序在为 SA 编程 MAC 地址寄存器时应该考虑到这一点。</p> <p>复位值：0</p>
8	SAIF	R_W	<p>SA 反滤波</p> <p>设置此位时，地址检查块以反向过滤模式运行，以进行 SA 地址比较。SA 与 SA 寄存器匹配的帧被标记为 SA 地址筛选器失败。</p> <p>复位此位时，SA 与 SA 寄存器不匹配的帧将标记为 SA 地址筛选器失败。</p> <p>复位值：0</p>
7:6	PCF	R_W	<p>传递控制帧</p> <p>这些位控制所有控制帧(包括单播和多播暂停帧)的转发。</p> <ul style="list-style-type: none"> <li>00: MAC 过滤所有控制帧，使其无法到达应用程序。</li> <li>01: MAC 将所有控制帧（暂停帧除外）转发给应用程序，即使它们未通过地址过滤器。</li> <li>10: MAC 将所有的控制帧转发给应用程序，即使它们未通过地址过滤器。</li> <li>11: MAC 转发通过地址过滤器的控制帧。</li> </ul> <p>暂停帧处理应满足以下条件：</p> <ul style="list-style-type: none"> <li>条件 1：MAC 处于全双工模式，并将寄存器 6(流量控制寄存器)的位 2 (RFE)设置为 1 以启用流量控制。</li> <li>条件 2：当寄存器 6(流量控制寄存器)的位 3 (UP)被设置时，接收帧的目标地址(DA)匹配特定组播地址或匹配 MAC 地址 0。</li> <li>条件 3：接收帧的类别域是 0x8808, OPCODE 字段是 0x0001。</li> </ul> <p>注意：只有当条件 1 为 true 时，即 MAC 编程为全双工模式工作并使能 RFE 位时，该字段才应该设置为 01。否则，可能会出现暂停帧过滤不一致的情况。当条件 1 为 false 时，暂停帧被认为是通用控制帧。因此，要在未启用全双工模式和流</p>

位段	名称	类型	说明
			量控制时传递所有控制帧（包括暂停帧），应将 PCF 字段设置为 10 或 11（根据应用程序的要求）。 复位值：00
5	DBF	R_W	禁用广播帧 当设置该位时，AFM 模块阻塞所有传入的广播帧。此外，它覆盖所有其他过滤器设置。 当复位该位时，AFM 模块通过所有接收到的广播帧。 复位值：0
4	PM	R_W	通过所有多播 设置时，该位表示所有接收到的具有多播目标地址的帧（目标地址字段中的第一位为“1”）都被传递。 复位时，组播帧的过滤依赖于 HMC 位。 复位值：0
3	DAIF	R_W	DA 反滤波 当设置该位时，地址检查块以反过滤方式对单播和组播帧进行 DA 地址比较。 当复位时，正常的帧过滤被执行。 复位值：0
2	HMC	R_W	散列多播 设置后，MAC 根据散列表对接收到的组播帧进行目标地址过滤。 复位时，MAC 会对组播帧进行完美目标地址过滤，即将 DA 字段与 DA 寄存器中的编程值进行比较。 如果在核心配置时没有选择 Hash Filter，则该位是保留位（只读）。 复位值：0
1	HUC	R_W	散列单播 设置后，MAC 根据散列表对单播帧进行目标地址过滤。 当复位时，MAC 对单播帧进行完美目标地址过滤，即将 DA 字段与 DA 寄存器中的编程值进行比较。 如果在核心配置时没有选择 Hash Filter，则该位是保留位（只读）。

位段	名称	类型	说明
			复位值：0
0	PR	R_W	混合模式 设置此位时，地址过滤器模块将传递所有传入帧，而不管目标地址或源地址如何。设置 PR 时，接收状态字的 SA 或 DA 过滤器故障状态位始终被清除。 复位值：0

### 1.6.3.27 寄存器 2（散列表高位寄存器）

64 位散列表用于组地址过滤。对于 hash 过滤，传入帧的目标地址内容通过 CRC 逻辑传递，CRC 寄存器的高 6 位用于索引散列表的内容。最高位决定使用哪个寄存器(散列表高位寄存器或低位寄存器)，其他 5 位决定寄存器中的哪个位。hash 值 5b'00000 选择所选寄存器的第 0 位，hash 值 5b'11111 选择所选寄存器的第 31 位。

目标地址的 hash 值是这样计算的：

1. 为 DA 计算 32 位的 CRC(关于计算 CRC32 的步骤，请参见 IEEE 802.3，章节 3.2.8)。
2. 对步骤 1 中获取的值进行位反转。
3. 取步骤 2 中值的高 6 位。

例如，如果接收到传入帧的 DA 为 0x1F52419CB6AF (0x1F 是 GMII 接口接收到的第一个字节)，那么内部计算的 6 位 Hash 值为 0x2C，并检查散列表高位寄存器的第 12 位用于过滤。如果接收到传入帧的 DA 为 0xA00A98000045，则计算出的 6 位 Hash 值为 0x07，并检查散列表低位寄存器的第 7 位用于过滤。

#### 说明

为了帮助您编写散列表，工作空间的/sample\_codes/目录中包含一个生成 DA 的 6 位 hash 的示例 C 例程。

如果寄存器对应的位值是 1'b1，则该帧被接受。否则，将被拒绝。如果在寄存器 1 中设置了 PM(通过所有多播)位，那么不管多播散列值是多少，所有多播帧都被接受。

如果配置散列表寄存器对(G)MII 时钟域进行双同步，则只有在写入散列表高位寄存器或低位寄存器的位[31:24](小端模式)或位[7:0](大端模式)时才会触发同步。当启用双同步时，应该在目标时钟域经过至少 4 个时钟周期后，再对这些寄存器执行连续写操作。

散列表高位寄存器包含散列表的高的 32 位。

图表 1-44 寄存器 2（散列表高位寄存器）

位段	名称	类型	说明
31:0	HTH	R_W	散列表高位 该字段包含散列表的高 32 位。 复位值：0000_0000H



### 1.6.3.28 寄存器 3（散列表低位寄存器）

散列表低位寄存器包含散列表的低 32 位。如果关闭 Hash Filter 功能，或者在核心配置时选择 128 位或 256 位散列表，则寄存器 2 和寄存器 3 都将被保留。

图表 1-45 寄存器 3（散列表低位寄存器）

位段	名称	类型	说明
31:0	HTL	R_W	散列表低位 该字段包含散列表的低 32 位。 复位值：0000_0000H

### 1.6.3.29 寄存器 4（GMII 地址寄存器）

GMII 地址寄存器通过管理接口控制外部 PHY 的管理周期。

图表 1-46 寄存器 4（GMII 地址寄存器）

位段	名称	类型	说明
31:16	-	RO	保留 复位值：0000H
15:11	PA	R_W	物理层地址 该字段表示 32 个可能的 PHY 设备中的哪一个正在被访问。 对于 RevMII，该字段给出了 RevMII 模块的 PHY 地址。 复位值：00H
10:6	GR	R_W	GMII 寄存器 这些位选择所选 PHY 设备中所需的 GMII 寄存器。 对于 RevMII，这些位在 RevMII 寄存器集中选择所需的 CSR 寄存器。 复位值：00H
5:2	CR	R_W	CSR 时钟范围 CSR 时钟范围选择根据设计使用的 CSR 时钟频率来确定 MDC 时钟的频率。不同 GMAC 配置对应的 CSR 时钟如 564 页的表 9-2 所示。 每个取值(位[5] = 0 时)的 CSR 时钟频率建议范围确保 MDC 时钟在 1.0 MHz ~ 2.5 MHz 之间。 ■ 0000: CSR 时钟频率为 60-100 MHz，MDC 时钟频率为

位段	名称	类型	说明
			<p>CSR 时钟/42。</p> <ul style="list-style-type: none"> <li>0001: CSR 时钟频率为 100-150 MHz, MDC 时钟频率为 CSR 时钟/62。</li> <li>0010: CSR 时钟频率为 20-35 MHz, MDC 时钟频率为 CSR 时钟/16。</li> <li>0011: CSR 时钟频率为 35-60 MHz, MDC 时钟频率为 CSR 时钟/26。</li> <li>0100: CSR 时钟频率为 150-250MHz, MDC 时钟频率为 CSR 时钟/102。</li> <li>0101: CSR 时钟频率为 250-300 MHz, MDC 时钟频率为 CSR 时钟/124。</li> <li>0110, 0111: 保留</li> </ul> <p>当设置位 5 时, 可以实现比 IEEE Std 802.3 规定的 2.5 MHz 的频率限制更高的 MDC 时钟频率, 并编程一个更低值的时钟分频器。例如, 当 CSR 时钟的频率为 100 MHz 时, 您将这些位设置为 1010, 那么得到的 MDC 时钟的频率为 12.5 MHz, 这超出了 IEEE 802.3 规定的范围。如果接口芯片支持更快的 MDC 时钟, 请编写以下值。</p> <ul style="list-style-type: none"> <li>1000: CSR 时钟/4</li> <li>1001: CSR 时钟/6</li> <li>1010: CSR 时钟/8</li> <li>1011: CSR 时钟/10</li> <li>1100: CSR 时钟/12</li> <li>1101: CSR 时钟/14</li> <li>1110: CSR 时钟/16</li> <li>1111: CSR 时钟/18</li> </ul> <p>这些位不用于访问 RevMII。如果 RevMII 接口被选择为单个 PHY 接口, 这些位是只读的。</p> <p>复位值: 0000</p>
1	GW	R_W	<p>GMII 写</p> <p>当设置时, 该位向 PHY 或 RevMII 表明这是一个使用 GMII 数据寄存器的写操作。如果没有设置该位, 则表示这是一个读操作, 即将数据放在 GMII data 寄存器中。</p> <p>复位值: 0</p>

位段	名称	类型	说明
0	GB	R_WS_ SC	<p>GMII 繁忙</p> <p>这个位应该在写入寄存器 4 和寄存器 5 之前读取逻辑 0。在 PHY 或 RevMII 寄存器访问期间，软件将该位设置为 1'b1，表示正在进行读或写访问。</p> <p>寄存器 5 在 MAC 清除该位之前是无效的，因此，寄存器 5 (GMII 数据)应该一直保持有效，直到 MAC 在进行 PHY 写操作时清除该位。类似地，对于读取操作，寄存器 5 的内容在清除该位之前是无效的。</p> <p>后续的读写操作应在前一个操作完成后再进行。因为在一个读或写操作完成后，没有从 PHY 到 MAC 的确认，即使在 PHY 不存在的情况下，该位的功能也没有改变。</p> <p>复位值：0</p>

### 1.6.3.30 寄存器 5（GMII 数据寄存器）

GMII 数据寄存器存储要写入寄存器 4 (GMII 地址寄存器)中指定地址的 PHY 寄存器的写数据。该寄存器还存储来自位于寄存器 4 指定地址的 PHY 寄存器的读数据。

图表 1-47 寄存器 5（GMII 数据寄存器）

位段	名称	类型	说明
31:16	-	RO	<p>保留</p> <p>复位值：0000H</p>
15:0	GD	R_W	<p>GMII 数据</p> <p>此字段包含管理读取操作后从 PHY 或 RevMII 读取的 16 位数据值，或管理写入操作前要写入 PHY 或 RevMII 的 16 位数据值。</p> <p>复位值：0000H</p>

### 1.6.3.31 寄存器 6（流控制寄存器）

流控制寄存器控制 MAC 的流控制模块的控制帧(暂停命令)的生成和接收。当写入寄存器的 Busy 位设置为“1”时，会触发流控制块生成暂停帧。控制帧的字段按照 802.3x 规范的规定进行选择，且从该寄存器中获得的暂停时间值用于控制帧的暂停时间字段。Busy 位保持设置，直到控制帧被传输到电缆上。主机必须确保在写入寄存器之前清除 Busy 位。

图表 1-48 寄存器 6（流控制寄存器）

位段	名称	类型	说明
31:16	PT	R_W	<p>暂停时间</p> <p>该字段保存在传输控制帧的暂停时间字段中使用的值。如果将暂停时间位配置为双同步到(G)MII 时钟域，那么应该在目标时钟域经过至少 4 个时钟周期之后，再对该寄存器执行连续写操作。</p> <p>复位值：0000H</p>
15:8	-	RO	<p>保留</p> <p>复位值：00H</p>
7	DZPQ	R_W	<p>禁用零量程暂停</p> <p>当设置该位时，它将禁用 FIFO 层(MTL 或外部边带流量控制信号 sbd_flowctrl_i/mti_flowctrl_i)流量控制信号去断言时自动生成零量程暂停帧。</p> <p>当该位被复位时，正常操作与自动零量程暂停帧生成被启用。</p> <p>复位值：0</p>
6	-	RO	<p>保留</p> <p>复位值：0</p>
5:4	PLT	R_W	<p>暂停低阈值</p> <p>该字段配置暂停定时器的阈值，在暂停定时器中检查输入流量控制信号 mti_flowctrl_i(或 sbd_flowctrl_i)是否自动重传暂停帧。</p> <p>该阈值应该总是小于在位[31:16]中配置的暂停时间的值。例如，如果 PT = 100H(256 时隙)，PLT = 01，那么在第一个暂停帧发送后 mti_flowctrl_i 信号在 228(256 - 28)时隙被断言时，则会自动发送第二个暂停帧。</p> <p>下面的列表提供了不同值的阈值：</p> <ul style="list-style-type: none"> <li>■ 00：阈值为暂停时间 - 4 时隙(PT - 4 时隙)。</li> <li>■ 01：阈值为暂停时间 - 28 时隙(PT - 28 时隙)。</li> <li>■ 10：阈值为暂停时间 - 144 时隙(PT - 144 时隙)。</li> <li>■ 11：阈值为暂停时间 - 256 时隙(PT - 256 时隙)。</li> </ul> <p>时隙定义为在 GMII 或 MII 接口上传输 512 位(64 字节)所需的时间。</p>

位段	名称	类型	说明
			复位值：00
3	UP	R_W	<p>单播暂停帧检测</p> <p>当暂停帧具有 IEEE Std 802.3 中指定的唯一组播地址时，将对其进行处理。当设置该位时，MAC 还可以检测到带有该站单播地址的暂停帧。这个单播地址应该在 MAC 地址 0 高位寄存器和 MAC 地址 0 低位寄存器中指定。</p> <p>当复位该位时，MAC 只检测具有唯一组播地址的暂停帧。</p> <p>注意：如果接收到的帧的组播地址与唯一组播地址不一致，则 MAC 不会处理该暂停帧。</p> <p>复位值：0</p>
2	RFE	R_W	<p>接收流量控制</p> <p>当设置此位时，MAC 解码接收到的暂停帧，并在指定的(暂停)时间内禁用它的发送器。当该位被复位时，暂停帧的解码功能被禁用。</p> <p>复位值：0</p>
1	TFE	R_W	<p>发送流量控制使能</p> <p>在全双工模式下，当设置该位时，MAC 允许流控操作发送暂停帧。当该位被复位时，MAC 的流控操作被关闭，MAC 不再发送暂停帧。</p> <p>在半双工模式下，当设置该位时，MAC 使能反压操作。当该位被复位时，反压特性被禁用。</p> <p>复位值：0</p>
0	FCB_ BPA	FCB 为 R_WS_SC  BPA 为 R_W	<p>流量控制繁忙或反压激活</p> <p>该位在全双工模式下启动暂停帧。如果设置了 TFE 位，则在半双工模式下激活反压功能。</p> <p>在全双工模式下，在写入流量控制寄存器之前，这个位应该被读为 1'b0。要启动暂停帧，应用程序必须将该位设置为 1'b1。在控制帧传输期间，该位继续被设置以表示帧传输正在进行中。在完成暂停帧传输后，MAC 将该位复位为 1'b0。在清除该位之前，不应该写入流量控制寄存器。</p> <p>在半双工模式下，当设置了该位(同时设置了 TFE)时，MAC 会产生反压。在反压期间，当 MAC 接收到新帧时，发送端开始发送 JAM 模式，导致碰撞。这个控制寄存器位与反压函数的 mti_flowctrl_i 输入信号是逻辑或的关系。当为全双工模式配</p>

位段	名称	类型	说明
			置 MAC 时，自动禁用 BPA。 复位值：0

### 1.6.3.32 寄存器 7（VLAN Tag 寄存器）

VLAN Tag 寄存器包含 IEEE 802.1Q VLAN Tag，用于标识 VLAN 帧。MAC 将接收帧(长度/类型)的第 13 和 14 个字节与 16'h81000 进行比较，后面两个字节与 VLAN tag 进行比较。如果匹配，则 MAC 在接收帧状态中设置 VLAN 位。帧的合法长度从 1,518 字节增加到 1,522 字节。

如果将 VLAN Tag 寄存器配置为双同步到(G)MII 时钟域，则应该在目标时钟域经过至少 4 个时钟周期后，再对该寄存器执行连续写操作。

图表 1-49 寄存器 7（VLAN Tag 寄存器）

位段	名称	类型	说明
31:20	-	RO	保留 复位值：000H
19	VTHM	R_W	VLAN Tag 散列表匹配使能 当设置时,VLAN tag 的 CRC 的最高 4 位用于索引寄存器 354 (VLAN 散列表寄存器)的内容。VLAN 散列表寄存器中的索引值为 1 表示该帧匹配了 VLAN 散列表。 当位 16 (ETV)被设置时，使用 12 位 VLAN 标识符(VID)的 CRC 进行比较。当 ETV 被复位时，使用 16 位 VLAN tag 的 CRC 进行比较。 复位时，不再执行 VLAN Hash Match 操作。如果在核心配置时没有使能 VLAN Hash 特性，则该位是保留位(只读，默认值)。 复位值：0
18	ESVL	R_W	使能 S-VLAN 当设置该位时，MAC 发送端和接收端也会认为 S-VLAN (Type = 0x88A8)帧是带 VLAN 标签的有效帧。 复位值：0
17	VTIM	R_W	VLAN Tag 逆匹配使能 设置此位时，启用 VLAN Tag 逆匹配。VLAN Tag 不匹配的帧被标记为匹配。 复位时，该位使 VLAN Tag 完美匹配。VLAN Tag 匹配的帧

位段	名称	类型	说明
			被标记为匹配。 复位值：0
16	ETV	R_W	使能 12 位 VLAN Tag 比较功能  设置该位时，使用 12 位的 VLAN 标识符进行比较和过滤，而不使用完整的 16 位 VLAN 标记。将 VLAN tag 的位[11:0]与接收到的 VLAN-tagged 帧中的相应字段进行比较。同样，当启用时，接收帧中只有 12 位 VLAN tag 用于基于 hash 算法的 VLAN 过滤。  复位该位后，将接收到的 VLAN 帧的第 15 字节和第 16 字节中的所有 16 位进行比较和 VLAN hash 过滤。  复位值：0
15:0	VL	R_W	接收帧的 VLAN Tag 标识符  此字段包含用于标识 VLAN 帧的 802.1Q VLAN 标记，并与 VLAN 帧接收的帧的第 15 和第 16 字节进行比较。下表描述了此字段的位： <ul style="list-style-type: none"> <li>■ 位[15:13]：用户优先级</li> <li>■ 位 12：标准格式指示器(CFI)或丢弃合格指示器(DEI)</li> <li>■ 位[11:0]：VLAN tag 的 VLAN 标识符(VID)字段</li> </ul> 当设置 ETV 位时，只使用 VID (位[11:0])进行比较。  如果 VL (如设置 ETV，则为 VL[11:0])为全 0，则 MAC 不再检查第 15 和第 16 字节进行 VLAN tag 比较，而是将 Type 字段值为 0x8100 或 0x88a8 的帧声明为 VLAN 帧。  复位值：0000H

### 1.6.3.33 寄存器 8（版本寄存器）

版本寄存器标识了 DWC\_gmac 的版本。该寄存器包含两个字节：一个用于标识核心发行版本号，另一个用于在核心配置期间设置。

图表 1-50 寄存器 8（版本寄存器）

位段	名称	类型	说明
31:16	-	RO	保留  复位值：0000H
15:8	USERVER	RO	自定义版本(使用 coreConsultant 配置)

位段	名称	类型	说明
			复位值: xxH
7:0	SNPSVER	RO	Synopsys 定义版本(3.7) 复位值: 37H

### 1.6.3.34 寄存器 9（调试寄存器）

调试寄存器给出了所有发送和接收数据路径的主要模块和 FIFO 的状态。全零状态表示 MAC 处于空闲状态(FIFO 为空), 数据路径中没有任何活动。

#### 说明

调试寄存器的复位值只有在复位操作期间存在以下时钟时才有效:

- clk\_csr\_i、clk\_app\_i、hclk\_i、or aclk\_i
- clk\_tx\_i
- clk\_rx\_i

图表 1-51 寄存器 9（调试寄存器）

位段	名称	类型	说明
31:26	-	RO	保留 复位值: 00H
25	TXSTSFSTS	RO	MTL TxStatus FIFO 满状态 当该字段值为高时, 表示 MTL TxStatus FIFO 已满。因此, MTL 不能接受更多的帧进行传输。这个位在 GMAC-AHB 和 GMAC-DMA 配置中是保留位。 复位值: 0
24	TXFSTS	RO	MTL Tx FIFO 不空状态 当该字段值为高时, 表示 MTL Tx FIFO 不为空, 留下一些数据供传输。 复位值: 0
23	-	RO	保留 复位值: 0
22	TWCSTS	RO	MTL Tx FIFO 写控制器状态 当该字段值为高时, 表示 MTL Tx FIFO 写控制器处于活动状态, 正在向 Tx FIFO 传输数据。 复位值: 0



位段	名称	类型	说明
21:20	TRCSTS	RO	MTL Tx FIFO 读控制器状态 该字段表示 Tx FIFO 读取控制器的状态： <ul style="list-style-type: none"> <li>■ 00: 空闲状态</li> <li>■ 01: 读取状态(向 MAC 发送器传输数据)</li> <li>■ 10: 等待来自 MAC 发送器的 TxStatus</li> <li>■ 11: 写入接收到的 TxStatus 或刷新 Tx FIFO</li> </ul> 复位值: 00
19	TXPAUSED	RO	MAC 发送器暂停 当该字段值为高时, 表示 MAC 发送端处于暂停状态(纯全双工模式), 因此不调度任何帧进行传输。 复位值: 0
18:17	TFCSTS	RO	MAC 传输帧控制器状态 该字段表示 MAC 传输帧控制器模块的状态。 <ul style="list-style-type: none"> <li>■ 00: 空闲状态</li> <li>■ 01: 等待前一帧或 IFG 状态或等待后退期结束</li> <li>■ 10: 生成和传输暂停帧(全双工模式下)</li> <li>■ 11: 传输输入帧</li> </ul> 复位值: 00
16	TPESTS	RO	MAC GMII 或 MII 传输协议引擎状态 当该字段值为高时, 表示 MAC GMII 或 MII 传输协议引擎正在主动传输数据, 而不是处于空闲状态。 复位值: 0
15:10	-	RO	保留 复位值: 0H
9:8	RXFSTS	RO	MTL Rx FIFO 充满率状态 该字段给出了 Rx FIFO 的充满率状态： <ul style="list-style-type: none"> <li>■ 00: Rx FIFO 空</li> <li>■ 01: Rx FIFO 充满率低于流量控制停用阈值</li> <li>■ 10: Rx FIFO 充满率高于流量控制激活阈值</li> <li>■ 11: Rx FIFO 满</li> </ul>

位段	名称	类型	说明
			复位值：00
7	-	RO	保留 复位值：0
6:5	RRCSTS	RO	MTL RxFIFO 读控制器状态 该字段给出了 Rx FIFO 读控制器的状态： <ul style="list-style-type: none"> <li>■ 00: 空闲状态</li> <li>■ 01: 读取帧数据</li> <li>■ 10: 读取帧状态(或时间戳)</li> <li>■ 11: 刷新帧数据和状态</li> </ul> 复位值：00
4	RWCSTS	RO	MTL Rx FIFO 写控制器活动状态 当该字段值为高时，表示 MTL Rx FIFO 写控制器处于活动状态，正在向 FIFO 传输接收到的帧。 复位值：0
3	-	RO	保留 复位值：0
2:1	RFCFCSTS	RO	MAC 接收帧 FIFO 控制器状态 当该字段值为高时，表示 MAC 接收帧控制模块的小 FIFO 读写控制器的活动状态。 <ul style="list-style-type: none"> <li>■ RFCFCSTS[1]表示小 FIFO 读控制器的状态。</li> <li>■ RFCFCSTS[0]表示小 FIFO 写控制器的状态。</li> </ul> 复位值：00
0	RPESTS	RO	MAC GMII 或 MII 接收协议引擎状态 当该字段为高时，表示 MAC GMII 或 MII 接收协议引擎正在主动接收数据，而不是处于空闲状态。 复位值：0

### 1.6.3.35 寄存器 12（LPI 控制和状态寄存器）

LPI 控制和状态寄存器控制 LPI 功能，并提供 LPI 中断状态。当读取这个寄存器时，状态位被清除。只有在核心配置期间选择节能以太网特性时，才会出现此寄存器。

图表 1-52 寄存器 12 (LPI 控制和状态寄存器)

位段	名称	类型	说明
31:20	-	RO	保留 复位值: 00000H
19	LPITXA	R_W	<p>LPI TX 自动化</p> <p>该位控制 MAC 在发送端进入或退出 LPI 模式时的行为。该位在 GMACCORE 配置中不起作用。在该配置中, 发送时钟门控在 LPI 模式下完成。</p> <p>如果 LPITXA 和 LPIEN 位设置为 1, 则只有在所有未完成的帧(在核心中)和未决的帧(在应用程序接口中)传输完毕后, MAC 才进入 LPI 模式。当应用程序发送任何帧来传输或应用程序发出 TX FIFO Flush 命令时, MAC 从 LPI 模式退出。MAC 在退出 LPI 状态时, 会自动清除 LPIEN 位。如果在寄存器 6(操作模式寄存器)的第 20 位设置 TX FIFO Flush, 当 MAC 处于 LPI 模式时, MAC 将退出 LPI 模式。</p> <p>当该位为 0 时, LPIEN 位直接控制 MAC 进入或退出 LPI 模式的行为。</p> <p>复位值: 0</p>
18	PLSEN	R_W	<p>PHY 链路状态使能</p> <p>该位使 RGMII、SGMII 或 SMII 接收路径上收到的链路状态用于激活 LPI LS TIMER。</p> <p>当设置时, MAC 使用寄存器 54(SGMII/RGMII/SMII 控制和状态寄存器)的链路状态位和位 17 (PLS)用于 LPI LS 定时器的触发器。当清除时, MAC 忽略寄存器 54 的链路状态位, 只取 PLS 位。</p> <p>如果没有选择 RGMII、SGMII 或 SMII PHY 接口, 该位为只读, 保留。</p> <p>复位值: 0</p>
17	PLS	R_W	<p>PHY 链路状态</p> <p>该位表示 PHY 的链路状态。只有当链路状态 up (好)时, MAC 发送器才会断言 LPI 模式, 至少在 LPI LS TIMER 指定的时间内是这样。</p> <p>当设置时, 认为链路为 up, 当复位时, 认为链路为 down。</p> <p>复位值: 0</p>

位段	名称	类型	说明
16	LPIEN	R_W_SC	LPI 启用 当设置时, 该位指示 MAC 发送器进入 LPI 状态。复位时, 该位指示 MAC 退出 LPI 状态, 恢复正常传输。 当设置 LPITXA 位后, 由于有新的报文发送, MAC 退出 LPI 状态, 该位将被清除。 复位值: 0
15:10	-	RO	保留 复位值: 00H
9	RLPIST	RO	接收 LPI 状态 当设置此位时, 表示 MAC 正在 GMII 或 MII 接口上接收 LPI 模式。 复位值: 0
8	TLPIST	RO	发送 LPI 状态 当设置此位时, 表示 MAC 正在 GMII 或 MII 接口上发送 LPI 模式。 复位值: 0
7:4	-	RO	保留 复位值: 0H
3	RLPIEX	R_SS_RC	接收 LPI 退出 当设置为该值时, 表示 MAC 接收端已经在 GMII 或 MII 接口上停止接收 LPI 模式, 退出 LPI 状态, 恢复正常接收。该位通过读入该寄存器来清除。 注意: 如果 MAC 在很短的时间内停止接收 LPI 模式, 例如小于 CSR 时钟的 3 个时钟周期, 该位可能不会被设置。 复位值: 0
2	RLPIEN	R_SS_RC	接收 LPI 进入 当设置此位时, 表示 MAC 接收器接收到一个 LPI 模式并进入 LPI 状态。该位通过读入该寄存器来清除。 注意: 如果 MAC 在很短的时间内停止接收 LPI 模式, 例如小于 CSR 时钟的 3 个时钟周期, 该位可能不会被设置。 复位值: 0

位段	名称	类型	说明
1	TLPIEX	R_SS_RC	发送 LPI 退出 该值表示当用户清除 LPIEN 位且 LPITW 定时器超时后, MAC 发送端已经退出 LPI 状态。 该位通过读入该寄存器来清除。 复位值: 0
0	TLPIEN	R_SS_RC	发送 LPI 进入 当设置该值时, 表示由于 LPIEN 位的设置, MAC 发送器已经进入 LPI 状态。该位通过读入该寄存器来清除。 复位值: 0

### 1.6.3.36 寄存器 13 (LPI 定时器控制寄存器)

LPI 定时器控制寄存器控制 LPI 状态的超时值。它指定 MAC 传输 LPI 模式所需的时间, 以及指定 MAC 恢复正常传输所需的等待时间。只有在核心配置期间选择节能以太网特性时, 才会出现此寄存器。

图表 1-53 寄存器 13 (LPI 定时器控制寄存器)

位段	名称	类型	说明
31:26	-	RO	保留 复位值: 00H
25:16	LST	R_W	LPI LS 定时器 该字段指定了在 LPI 模式被传输到 PHY 之前, PHY 链路状态成为 up (OKAY) 的最小时间(以毫秒为单位)。即使设置了 LPIEN 位, MAC 也不会发送 LPI 模式, 除非 LPI LS 定时器达到设定的终端计数。LPI LS 定时器的缺省值为 IEEE 标准定义的 1000(1 秒)。 复位值: 0x3E8
15:0	TWT	R_W	LPI TW 定时器 该字段指定 MAC 从停止向 PHY 发送 LPI 模式到恢复正常传输之间等待的最小时间(以微秒为单位)。TLPIEX 状态位在该定时器到期后设置。 复位值: 0

### 1.6.3.37 寄存器 14（中断状态寄存器）

中断状态寄存器标识 MAC 中可能产生中断的事件。只有在核心配置期间选择了相应的可选特性并在运行期间启用时，才会产生所有中断事件。因此，当核心中不存在相应的特征时，这些位是保留位。

图表 1-54 寄存器 14（中断状态寄存器）

位段	名称	类型	说明
31:12	-	RO	保留 复位值：000000H
11	GPIIS	RO	GPI 中断状态 当 GPIO 特性启用时, 当任何活动事件(LL 或 LH)在寄存器 56 (通用 IO 寄存器)的 GPIS 字段(位[3:0])发生且对应的 GPIE 位使能时, 该位被设置。该位在寄存器 56(通用 IO 寄存器)的读取通道 0 (GPIS)上被清除。当 GPIO 特性未启用时, 该位是保留位。 复位值：0
10	LPIIS	RO	LPI 中断状态 当节能以太网特性启用时, 该位用于 MAC 发送器或接收器的任何 LPI 状态的进入或退出。读取寄存器 12 (LPI 控制和状态寄存器)的 0 位时该位被清除。在所有其他模式中, 此位是保留位。 复位值：0
9	TSIS	RO/ R_SS_RC	时间戳中断状态 高级时间戳特性启用时, 当以下任一条件为真时, 该位被设置: <ul style="list-style-type: none"> <li>■ 系统时间值等于或超过目标时间高位和低位寄存器中指定的值。</li> <li>■ 秒寄存器溢出。</li> <li>■ 辅助快照触发器被断言。</li> </ul> 该位在读取寄存器 458(时间戳状态寄存器)的第 0 位时被清除。 如果启用默认时间戳, 当设置此位时, 表示系统时间值等于或超过目标时间寄存器中指定的值。在该模式下, 该位在被读取完成后被清除。在所有其他模式中, 此位是保留位。 复位值：0

位段	名称	类型	说明
8	-	RO	保留 复位值：000H
7	MMCRXIPIS	RO	MMC 接收校验和卸载中断状态 当在 MMC 接收校验和卸载中断寄存器中产生中断时，该位被设置为高。当这个中断寄存器中的所有位都被清除时，这个位也被清除。 只有在核心配置期间选择可选的 MMC 模块和校验和卸载引擎(类型 2)时，该位才有效。 复位值：0
6	MMCTXIS	RO	MMC 发送中断状态 当在 MMC 发送中断寄存器中产生中断时，该位被设置为高。当这个中断寄存器中的所有位都被清除时，这个位也被清除。 该位仅在核心配置时选择可选 MMC 模块时有效。 复位值：0
5	MMCRXIS	RO	MMC 接收中断状态 当在 MMC 接收中断寄存器中产生中断时，该位被设置为高。当这个中断寄存器中的所有位都被清除时，这个位也被清除。 该位仅在核心配置时选择可选 MMC 模块时有效。 复位值：0
4	MMCIS	RO	MMC 中断状态 当位[7:5]中的任何位被设置为高时，该位被设置为高，只有当所有位都设置为低时，该位才被清除。 该位仅在核心配置时选择可选 MMC 模块时有效。 复位值：0
3	PMTIS	RO	PMT 中断状态 当在断电模式下接收到一个魔法包或远程唤醒帧时，该位被设置(参见 PMT 控制和状态寄存器中的第 5 位和第 6 位)。当两个位[6:5]都因为对 PMT 控制和状态寄存器的读操作被清除时，该位被清除。 只有在核心配置期间选择可选的 PMT 模块时，该位才有效。 复位值：0
2	PCSANCIS	RO	PCS 自动协商完成

位段	名称	类型	说明
			该位在 TBI, RTBI 或 SGMII PHY 接口(寄存器 49(AN 状态寄存器)中的第 5 位)完成自动协商时设置。当对 AN 状态寄存器执行读操作时, 该位将被清除。  只有在核心配置和运行时选择了可选的 TBI / RTBI / SGMII PHY 接口时, 该位才有效。  复位值: 0
1	PCSLCHGIS	RO	PCS 链路状态改变  该位是由于 TBI、RTBI 或 SGMII PHY 接口(寄存器 49(状态寄存器)中的第 2 位)中链路状态值的任何变化而设置的。当对 AN Status 寄存器执行读操作时, 该位将被清除。  只有在核心配置和运行时选择了可选的 TBI / RTBI / SGMII PHY 接口时, 该位才有效。  复位值: 0
0	RGSMIIIS	RO	RGMII 或 SMII 中断状态  该位是由于 RGMII 或 SMII 接口(寄存器 54 (SGMII/RGMII/SMII 控制和状态寄存器)中的第 3 位)中链路状态值的任何变化而设置的。当对 SGMII/RGMII/SMII 控制和状态寄存器执行读取操作时, 该位将被清除。  只有在核心配置和运行时选择了可选的 RGMII 或 SMII PHY 接口时, 该位才有效。  复位值: 0

### 1.6.3.38 寄存器 15 (中断屏蔽寄存器)

中断屏蔽寄存器位使您能够因为在中断状态寄存器中对应的事件而屏蔽中断信号。中断信号在 GMAC-AHB、GMAC-AXI 和 GMAC-DMA 配置中是 sbd\_intr\_o, 在 GMAC-MTL 和 GMAC-CORE 配置中是 mci\_intr\_o。

图表 1-55 寄存器 15 (中断屏蔽寄存器)

位段	名称	类型	说明
31:11	-	RO	保留  复位值: 00000H
10	LPIIM	R_W	LPI 中断屏蔽  当设置该位时, 由于寄存器 14(中断状态寄存器)中 LPI 中断状态位的设置, 该位禁用中断信号的断言。



位段	名称	类型	说明
			只有在核心配置时选择节能以太网特性时，该位才有效。在所有其他模式中，此位是保留位。 复位值：0
9	TSIM	R_W	时间戳中断屏蔽 当设置时，由于寄存器 14(中断状态寄存器)中时间戳中断状态位的设置，该位禁用中断信号的断言。 只有启用 IEEE1588 时间戳时，该位才有效。在所有其他模式中，此位是保留位。 复位值：0
8:4	-	RO	保留 复位值：00H
3	PMTIM	R_W	PMT 中断屏蔽 当设置时，由于寄存器 14(中断状态寄存器)中 PMT 中断状态位的设置，该位禁用中断信号的断言。 复位值：0
2	PCSANCIM	R_W	PCS AN 完成中断屏蔽 当设置时，由于在寄存器 14(中断状态寄存器)中 PCS 自协商完成位的设置，该位禁用中断信号的断言。 复位值：0
1	PCSLCHGIM	R_W	PCS 链路状态中断屏蔽 当设置时，由于寄存器 14(中断状态寄存器)中 PCS 链路状态改变位的设置，该位禁用中断信号的断言。 复位值：0
0	RGSMIIIM	R_W	RGMII 或 SMII 中断屏蔽 当设置该位时，由于寄存器 14(中断状态寄存器)中 RGMII 或 SMII 中断状态位的设置，该位禁用中断信号的断言。 复位值：0

### 1.6.3.39 寄存器 16 (MAC 地址 0 高位寄存器)

MAC 地址 0 高位寄存器保存站的第一个 6 字节 MAC 地址的高 16 位。在(G)MII 接口接收到的第一个 DA 字节与 MAC 地址低位寄存器的 LS 字节(位[7:0])对应。例如，如果在(G)MII 上收到 0x112233445566 (0x11 在第一列的通道 0)作为目的地址，那么 MAC 地址 0 寄存器[47:0]将与 0x665544332211 进行比较。

如果将 MAC 地址寄存器配置为双同步到(G)MII 时钟域, 则只有当写入 MAC 地址 0 低位寄存器的位[31:24] (小端模式)或位[7:0] (大端模式)时才会触发同步。为了进行正确的同步更新, 应该在目标时钟域至少四个时钟周期之后, 对这个地址低位寄存器进行连续的写操作。

图表 1-56 寄存器 16 (MAC 地址 0 高位寄存器)

位段	名称	类型	说明
31	AE	RO	地址启用 该位总是设为 1。 复位值: 1
30:16	-	RO	保留 复位值: 0000H
15:0	ADDRHI	R_W	MAC 地址 0 [47:32] 该字段包含 MAC 地址前 6 字节的高 16 位(47:32)。MAC 使用该字段对接收到的帧进行过滤, 并在发送流量控制(暂停)帧中插入 MAC 地址。 复位值: FFFFH

#### 1.6.3.40 寄存器 17 (MAC 地址 0 低位寄存器)

MAC 地址 0 低位寄存器保存站的第一个 6 字节 MAC 地址的低 32 位。

图表 1-57 寄存器 17 (MAC 地址 0 低位寄存器)

位段	名称	类型	说明
31:0	ADDRLO	R_W	MAC 地址 0 [31:0] 该字段包含第一个 6 字节 MAC 地址的低 32 位。MAC 用来过滤接收到的帧, 并在发送流量控制(暂停)帧中插入 MAC 地址。 复位值: FFFF_FFFFH

#### 1.6.3.41 寄存器 18 (MAC 地址 1 高位寄存器)

MAC 地址 1 高位寄存器保存站的第二个 6 字节 MAC 地址的高 16 位。

如果配置 MAC 地址寄存器双同步到(G)MII 时钟域, 则只有当写入 MAC 地址 1 低位寄存器的位[31:24] (小端模式)或位[7:0] (大端模式)时才会触发同步。为了进行正确的同步更新, 应该在目标时钟域至少四个时钟周期之后, 对这个地址低位寄存器进行连续的写操作。

图表 1-58 寄存器 18 (MAC 地址 1 高位寄存器)

位段	名称	类型	说明
31	AE	R_W	地址启用 设置该位后, 地址过滤器模块使用第二个 MAC 地址进行完美过滤。 当该位被复位时, 地址过滤器模块忽略要过滤的地址。 复位值: 0
30	SA	R_W	源地址 设置该位后, MAC 地址 1[47:0]用来与接收帧的 SA 字段进行比较。 当复位该位时, 使用 MAC 地址 1[47:0]与接收帧的 DA 字段进行比较。 复位值: 0
29:24	MBC	R_W	屏蔽字节控制 这些位是屏蔽控制位, 用于比较各个 MAC 地址字节。当设置为高时, MAC 不会将接收到的 DA 或 SA 对应的字节与 MAC 地址 1 寄存器的内容进行比较。各个位按如下方式控制字节的屏蔽: <ul style="list-style-type: none"> <li>■ 第 29 位: 寄存器 18[15:8]</li> <li>■ 第 28 位: 寄存器 18[7:0]</li> <li>■ 第 27 位: 寄存器 19[31:24]</li> <li>■ ...</li> <li>■ 第 24 位: 寄存器 19[7:0]</li> </ul> 您可以通过屏蔽地址的一个或多个字节来过滤一组地址(称为组地址过滤)。 复位值: 000000
23:16	-	RO	保留 复位值: 00H
15:0	ADDRHI	R_W	MAC 地址 1 [47:32] 该字段包含第二个 6 字节 MAC 地址的高 16 位(47:32)。 复位值: FFFFH

### 1.6.3.42 寄存器 19 (MAC 地址 1 低位寄存器)

MAC 地址 1 低位寄存器保存站的第二个 6 字节 MAC 地址的低 32 位。

图表 1-59 寄存器 19 (MAC 地址 1 低位寄存器)

位段	名称	类型	说明
31:0	ADDRLO	R_W	MAC 地址 1 [31:0]  该字段包含第二个 6 字节 MAC 地址的低 32 位。这个字段的内容是未定义的, 直到应用程序在初始化过程之后加载出来。  复位值: FFFF_FFFFH

#### 说明

- 寄存器 20、22、24、26、28、30、32、34、36、38、40、42、44 和 46 (MAC 地址 2 高位寄存器到 MAC 地址 15 高位寄存器)的描述和寄存器 18 (MAC 地址 1 高位寄存器)的描述相同。
- 寄存器 21、23、25、27、29、31、33、35、37、38、41、43、45 和 47 (MAC 地址 2 低位寄存器到 MAC 地址 15 低位寄存器)的描述和寄存器 19 (MAC 地址 1 低位寄存器)的描述相同。
- 寄存器 512、514、516、518、520、522、524、526、528、530、532、534、536、538、540 和 542 (MAC 地址 16 高位寄存器到 MAC 地址 31 高位寄存器)的描述和寄存器 18 (MAC 地址 1 高位寄存器)的描述相同。
- 寄存器 513、515、517、519、521、523、525、527、529、531、533、535、537、539、541 和 543 (MAC 地址 16 低位寄存器到 MAC 地址 31 低位寄存器)的描述与寄存器 19 (MAC 地址 1 低位寄存器)的描述相同。
- 寄存器 546、548、550、552、554、556、558、560、562、564、566、568、570、572、574、576、578、580、582、584、586、588、590、592、594、596、598、600、602、604、606、608、610、612、614、616、618、620、622、624、626、628、630、632、634、636、638、640、642、644、646、648、650、652、654、656、658、660、662、664、666、668、670、672、674、676、678、680、682、684、686、688、690、692、694、696、698、700、702、704、706、708、710、712、714、716、718、720、722、724、726、728、730、732 和 734 (MAC 地址 33 高位寄存器到 MAC 地址 127 高位寄存器)的描述与寄存器 544 (MAC 地址 32 高位寄存器)的描述相同。
- 描述寄存器 545、547、549、551、553、555、557、559、561、563、565、567、569、571、573、575、577、579、581、583、585、587、589、591、593、595、597、599、601、603、605、607、609、611、613、615、617、619、621、623、625、627、629、631、633、635、637、639、641、643、645、647、649、651、653、655、657、659、661、663、665、667、669、671、673、675、677、679、681、683、685、687、689、691、693、695、697、699、701、703、705、707、709、711、713、715、717、719、721、723、725、727、729、731、733 和 735 (MAC 地址 32 低位寄存器到 MAC 地址 127 低位寄存器)与寄存器 19 (MAC 地址 1 低位寄存器)的描述相同。

### 1.6.3.43 寄存器 544 (MAC 地址 32 高位寄存器)

MAC 地址 32 高位寄存器保存站的第 33 个 6 字节 MAC 地址的高 16 位。

通过在 coreConsultant 中选择“同步 CSR MAC 地址到 Tx/Rx 时钟域”选项, 可以配置需要双同步的 MAC 地址寄存器。如果配置 MAC 地址寄存器双同步到(G)MII 时钟域, 那么只有当 MAC 地址低位寄存器(寄

寄存器 545)的位[31:24](小端模式)或位[7:0](大端模式)被写入时才会触发同步。为了进行正确的同步更新,应该在目标时钟域至少四个时钟周期之后,对 MAC 地址低位寄存器(寄存器 545)进行连续的写操作。

图表 1-60 寄存器 544 (MAC 地址 32 高位寄存器)

位段	名称	类型	说明
31	AE	R_W	地址启用 设置该位后,地址过滤器模块使用第 33 个 MAC 地址进行完美过滤。当复位时,地址过滤器模块忽略要过滤的地址。 复位值: 0
30:16	-	RO	保留 复位值: 0000H
15:0	ADDRHI	R_W	MAC 地址 32 [47:32] 该字段包含第 33 个 6 字节 MAC 地址的高 16 位(47:32)。 复位值: FFFFH

### 1.6.3.44 寄存器 48 (AN 控制寄存器)

AN 控制寄存器启用和/或重新启动自动协商。它也启用 PCS 环回。该寄存器是可选的,只有在为 TBI、SGMII 或 RTBI PHY 接口配置 MAC 时才会出现。

图表 1-61 寄存器 48 (AN 控制寄存器)

位段	名称	类型	说明
31:19	-	RO	保留 复位值: 0000H
18	SGMRAL	R_W	SGMII RAL 控制 当设置时,该位强制 SGMII RAL 块以在 MAC 配置寄存器里速度和端口选择位中配置的速度运行。当 SGMII 接口用于 MAC 到 MAC 的直接连接时(没有 PHY),并且任何 MAC 必须重新配置速度时,这是很有用的。 复位时,SGMII RAL 块根据在 SGMII 上(从 PHY)接收到的链路速度状态进行操作。 如果配置核心时没有选择 SGMII PHY 接口,该位为保留位(只读)。 复位值: 0
17	LR	R_W	锁定到参考

位段	名称	类型	说明
			当设置时，该位使 PHY 锁定其锁相环（PLL）到 125 MHz 参考时钟。该位控制 TBI、RTBI 或 SGMII 接口上的 pcs_lck_ref_o 信号。 复位值：0
16	ECD	R_W	使能逗点检测 当设置此位时，将使 PHY 用于逗点检测和字重同步。该位控制 TBI、RTBI 或 SGMII 接口上的 pcs_en_cdet_o 信号。 复位值：0
15	-	RO	保留 复位值：0
14	ELE	R_W	使能外部环回 当设置此位时，使能 PHY 将传输数据环回到接收路径中。当设置该位时，pcs_ewrap_o 信号被断言为高。 复位值：0
13	-	RO	保留 复位值：0
12	ANE	R_W	自动协商使能 当设置此位时，使 MAC 能够与对端进行自动协商。清除此位将禁用自动协商。 复位值：0
11:10	-	RO	保留 复位值：00
9	RAN	R_WS_SC	重启自动协商 当该位被设置时，如果位 12 (ANE) 被设置，该位会导致自动协商重启。在自动协商开始后，该位会自动清除的。该位应清除以供正常操作。 复位值：0
8:0	-	RO	保留 复位值：00H

### 1.6.3.45 寄存器 49 (AN 状态寄存器)

AN 状态寄存器指示链路和自协商状态。该寄存器是可选的，只有在为 TBI、RTBI 或 SGMII PHY 接口配置 MAC 时才会出现。

图表 1-62 寄存器 49 (AN 状态寄存器)

位段	名称	类型	说明
31:9	-	RO	保留 复位值: 00_0000H
8	ES	RO	扩展状态 如果在核心配置时选择了 TBI 或 RTBI 接口, 表明 MAC 支持寄存器 53 (TBI 扩展状态寄存器) 中的扩展状态信息, 则该位为高。如果在核心配置期间选择了 SGMII 接口, 而没有选择 TBI 或 RTBI 接口, 表明寄存器 53 不存在, 则该位为低。 复位值: 1: TBI 或 RTBI 接口 0: SGMII 接口 没有 TBI or RTBI 接口
7:6	-	RO	保留 复位值: 00
5	ANC	RO	自动协商完成 当设置此位时, 表示自动协商过程已经完成。 当重新启动自动协商时, 此位将被清除。 复位值: 0
4	-	RO	保留 复位值: 0
3	ANA	RO	自动协商能力 该位总是为高, 因为 MAC 支持自动协商。 复位值: 1
2	LS	R_SS_SC_LL 0	链路状态 该位表示数据通道(链路)是 up 还是 down。对于 TBI、RTBI 或 SGMII 接口, 如果自协商正在进行, 数据无法通过链路传输, 因此链路被认为是 down。

位段	名称	类型	说明
			复位值：0
1:0	-	RO	保留 复位值：00

### 1.6.3.46 寄存器 50（自动协商广告寄存器）

自动协商广告寄存器指链路和自动协商状态。该寄存器是可选的，只有在为 TBI 或 RTBI PHY 接口配置 MAC 时才会出现。

图表 1-63 寄存器 50（自动协商广告寄存器）

位段	名称	类型	说明
31:16	-	RO	保留 复位值：0000H
15	NP	RO	下一页支持 该位总是为低，因为 MAC 不支持下一页。 复位值：0
14	-	RO	保留 复位值：0
13:12	RFE	R_W	远程故障编码 这些位提供远程错误编码，向对端指示发生了故障或错误。这些位的编码请参见 IEEE 802.3z 第 37.2.1.5 节。 复位值：00
11:9	-	RO	保留 复位值：000
8:7	PSE	R_W	暂停编码 这些位提供了对暂停位的编码，表明 MAC 能够按照 IEEE 802.3 3x 中定义的那样配置暂停功能。这些位的编码请参见 IEEE 802.3z 第 37.2.1.4 节。 复位值：11
6	HD	R_W	半双工 当该值设置为高时，表示 MAC 支持半双工模式。当 MAC 配置为纯全双工模式时，该位一直为低（只读）。



位段	名称	类型	说明
			复位值：1
5	FD	R_W	全双工 当该值设置为高时，表示 MAC 支持全双工模式。 复位值：1
4:0	-	RO	保留 复位值：00000

### 1.6.3.47 寄存器 51（自动协商对端能力寄存器）

自动协商对端能力寄存器包含对端的宣传能力。这个寄存器是可选的，只有在为 TBI 或 RTBI PHY 接口配置 MAC 时才会出现。

图表 1-64 寄存器 51（自动协商对端能力寄存器）

位段	名称	类型	说明
31:16	-	RO	保留 复位值：0000H
15	NO	RO	下一页支持 设置后，该位表示有更多下一页信息可用。当清除此位时，表示不希望交换下一页信息。 复位值：0
14	ACK	RO	确认 设置时，自协商功能使用该位表明对端已经成功接收到 MAC 的基页。清除该值时，表示对端未成功接收 MAC 的基页。 复位值：0
13:12	RFE	RO	远程故障编码 这些位提供远程错误编码，指示对端发生故障或错误。这些位的编码请参见 IEEE 802.3z 第 37.2.1.5 节。 复位值：00
11:9	-	RO	保留 复位值：000
8:7	PSE	RO	暂停编码 这些位为暂停位提供编码，表示对端按照 IEEE 802.3x 规范

位段	名称	类型	说明
			配置暂停功能的能力。这些位的编码请参见 IEEE 802.3z 第 37.2.1.4 节。 复位值：00
6	HD	RO	半双工 当设置此位时，表示对端具有在半双工模式下工作的能力。当该位被清除时，表示对端不具备在半双工模式下工作的能力。 复位值：0
5	FD	RO	全双工 当设置此位时，表示对端具有在全双工模式下工作的能力。当清除该位时，表示对端没有能力在全双工模式下工作。 复位值：0
4:0	-	RO	保留 复位值：00000

### 1.6.3.48 寄存器 52（自动协商扩展寄存器）

自动协商扩展寄存器表示 MAC 是否从对端接收到了新的基页。这个寄存器是可选的，只有在为 TBI 或 RTBI PHY 接口配置 MAC 时才会出现。

图表 1-65 寄存器 52（自动协商扩展寄存器）

位段	名称	类型	说明
31:3	-	RO	保留 复位值：0000_0000H
2	NPA	RO	下一页能力 因为 MAC 不支持下一页功能，所以这个位总是低的。 复位值：0
1	NPR	RO	接收到新页面 当设置此位时，表示 MAC 接收到了一个新页。该位在读取时被清除。 复位值：0
0	-	RO	保留 复位值：0

### 1.6.3.49 寄存器 53 (TBI 扩展状态寄存器)

TBI 扩展状态寄存器指示 MAC 的所有操作模式。该寄存器是可选的，只有在为 TBI 或 RTBI PHY 接口配置 MAC 时才会出现。

图表 1-66 寄存器 53 (TBI 扩展状态寄存器)

位段	名称	类型	说明
31:16	-	RO	保留 复位值: 0000H
15	GFD	RO	1000BASE-X 全双工能力 该位表示 MAC 能够执行全双工和 1000BASE-X 操作。 复位值: 1
14	GHD	RO	1000BASE-X 半双工能力 该位表示 MAC 能够执行半双工和 1000BASE-X 操作。在核心配置期间, 当 MAC 配置为纯全双工操作时, 该位总是为低。 复位值: 1
13:0	-	RO	保留 复位值: 0000H

### 1.6.3.50 寄存器 54 (SGMII/RGMII/SMII 控制和状态寄存器)

SGMII/RGMII/SMII 控制和状态寄存器表示 SGMII、RGMII 或 SMII 接口(复位时选定)从 PHY 接收到的状态信号。这个寄存器是可选的，只有在为 SGMII、RGMII 或 SMII PHY 接口配置 MAC 时才会出现。

图表 1-67 寄存器 54 (SGMII/RGMII/SMII 控制和状态寄存器)

位段	名称	类型	说明
31:17	-	RO	保留 复位值: 000H
16	SMIDRXS	R_W	延迟 SMII RX 数据采样相对于 SMII 同步信号 当设置时, SMII RX 数据的第一个位在 SMII 同步信号之后采样一个周期。复位时, 将 SMII RX 数据的第一个位与 SMII 同步信号一起采样。 如果配置核心时选择了源同步模式的 SMII PHY 接口, 该位为保留位(只读, 默认值)。 复位值: 0

位段	名称	类型	说明
15:6	-	RO	保留 复位值：000H
5	FALSCARDET	RO	检测到假载波 该位表示 SMII PHY 是否检测到假载波(1'b1)。在为 SGMII 或 RGMII PHY 接口配置 MAC 时，该位是保留位。 复位值：0
4	JABTO	RO	Jabber 超时 该位表示接收帧中是否有 jabber 超时错误(1'b1)。在为 SGMII 或 RGMII PHY 接口配置 MAC 时，该位为保留位。 复位值：0
3	LNKSTS	RO	链路状态 该位表示本地PHY和远端PHY之间的链路是 up 还是 down。该位给出了 MAC 的 SGMII 和本地 PHY 的 SGMII 之间的链路状态。在 SGMII 链路上的 MAC 和 PHY 自协商期间，状态位从本地 PHY 接收。 复位值：0
2:1	LNKSPEED	RO	链接速度 该位表示链路的当前速度： <ul style="list-style-type: none"> <li>00: 2.5 MHz</li> <li>01: 25 MHz</li> <li>10: 125 MHz</li> </ul> 为 SMII PHY 接口配置 MAC 时，预留位 2。 复位值： 对于 SGMII 为 10 对于 RGMII 和 SMII 为 00
0	LNKMOD	RO	链接模式 该位表示链路当前的工作模式： <ul style="list-style-type: none"> <li>1'b0: 半双工模式</li> <li>1'b1: 全双工模式</li> </ul> 复位值：0

### 1.6.3.51 寄存器 55（看门狗超时寄存器）

该寄存器控制接收帧的看门狗超时。

图表 1-68 寄存器 55（看门狗超时寄存器）

位段	名称	类型	说明
31:17		RO	保留 复位值：0000H
16	PWE	R_W	可编程看门狗使能 当设置该位并复位寄存器 0(MAC 配置寄存器)的位 23 (WD) 时, WTO 字段(位[13:0])作为接收帧的看门狗超时时间。当该位被清除时, 接收帧的看门狗超时时间由寄存器 0(MAC 配置寄存器)的位 23 (WD)和位 20 (JE)的设置来控制。 复位值：0
15:14		RO	保留 复位值：00
13:0	WTO	R_W	看门狗超时 当位 16 (PWE)和寄存器 0 (MAC 配置寄存器)的位 23 (WD) 被复位时, 该字段被用作接收帧的看门狗超时时间。如果接收帧长度超过该值, 则终止该帧并声明为错误帧。 注意：当设置了位 16 (PWE)时, 该字段的值应该大于 1,522 (0x05F2)。否则, IEEE Std 802.3 指定的有效标记帧将被声明为错误帧并被丢弃。 复位值：0000H

### 1.6.3.52 寄存器 56（通用 IO 寄存器）

该寄存器提供了多达 4 位的输出端口(GPO) 的驱动控制和多达 4 个输入端口(GPIS)的状态。它还提供了在 gpi\_i 引脚上发生事件时生成中断的控制。

图表 1-69 寄存器 56（通用 IO 寄存器）

位段	名称	类型	说明
31:28		RO	保留 复位值：0H
27:24	GPIT	R_W	GPI 类型 当设置此位时, 表示对应的 GPIS 为锁存低(LL)类型。复位时,

位段	名称	类型	说明
			该位表示对应的 GPIS 为锁存高(LH)类型。 该字段中可用的位数取决于 GP 输入信号宽度选项。其他位不使用(保留并总是复位)。 复位值: 0H
23:20		RO	保留 复位值: 0H
19:16	GPIE	R_W	GPI 中断使能 当该位被设置, 已程序事件(LL 或 LH)发生在相应的 GPIS 位上时, 寄存器 14(中断状态寄存器)的位 11 (GPIIS)被设置。相应地, 在 mci_intr_o 或 sbd_intr_o 上生成中断。当主机读取该寄存器的位[7:0]时, GPIIS 位被清除。 复位时, 当对应的 GPIS 位发生任何事件时, 寄存器 14(中断状态寄存器)的第 11 位(GPIIS)不被设置。 该字段中可用的位数取决于 GP 输入信号宽度选项。其他位不使用(保留并总是复位)。 复位值: 0H
15:12		RO	保留 复位值: 0H
11:8	GPO	R_W	通用输出 设置该位后, 它直接驱动 gpo_o 输出端口。当该位被复位时, 它不直接驱动 gpo_o 输出端口。 该字段中可用的位数取决于 GP 输出信号宽度选项。其他位不使用(保留并总是复位)。 复位值: 0H
7:4		RO	保留 复位值: 0H
3:0	GPIS	LL, LH	通用输入状态 这个字段给出连接到 gpi_i 输入端口的信号的状态。根据该寄存器对应的 GPIT 字段的设置, 该字段有以下几种类型: <ul style="list-style-type: none"> <li>■ 锁存低(LL): 当对应的 gpi_i 输入变为 low 时, 该字段被清除。该字段保持低位, 直到主机读取该字段。在此之后, 该字段反映 gpi_i 输入的当前值。</li> </ul>

位段	名称	类型	说明
			<p>■ 锁存高(LH): 当对应的 gpi_i 输入变高时, 设置该字段。该字段保持高位, 直到主机读取该字段。在此之后, 该字段反映 gpi_i 输入的当前值。</p> <p>该字段中可用的位数取决于 GP 输入信号宽度选项。其他位不使用(保留并总是复位)。</p> <p>复位值: 0H</p>

### 1.6.3.53 寄存器 256 (第 3 层和第 4 层控制寄存器 0)

这个寄存器控制第 3 层和第 4 层的过滤器 0 的操作。如果在核心配置期间没有选择第 3 层和第 4 层过滤特性, 则保留此寄存器。

图表 1-70 寄存器 256 (第 3 层和第 4 层控制寄存器 0)

位段	名称	类型	说明
31:22	-	RO	保留 复位值: 000H
21	L4DPIM0	R_W	<p>第 4 层目的端口逆匹配使能</p> <p>当设置此位时, 表示启用第 4 层目的端口数域进行逆匹配。复位时, 该位表示启用 4 层目的端口数域进行完美匹配。</p> <p>该位仅在位 20 (L4DPM0)设置为高时有效。</p> <p>复位值: 0</p>
20	L4DPM0	R_W	<p>第 4 层目的端口匹配使能</p> <p>当设置此位时, 表示启用第 4 层目的端口数域进行匹配。复位时, MAC 忽略进行匹配的第 4 层目的端口数域。</p> <p>复位值: 0</p>
19	L4SPIM0	R_W	<p>第 4 层源端口逆匹配使能</p> <p>当设置此位时, 表示启用第 4 层源端口数域进行逆匹配。复位时, 该位表示启用第 4 层源端口数域进行完美匹配。</p> <p>只有当位 18 (L4SPM0)设置为高时, 该位才有效。</p> <p>复位值: 0</p>
18	L4SPM0	R_W	<p>第 4 层源端口匹配使能</p> <p>当设置此位时, 表示启用第 4 层源端口数域进行匹配。复位时, MAC 忽略进行匹配的第 4 层源端口数域。</p>

位段	名称	类型	说明
			复位值：0
17	-	RO	保留 复位值：0
16	L4PEN0	R_W	第 4 层协议启用 当设置此位时，表示使用 UDP 帧的源端口数域和目的端口数域进行匹配。复位时，该位表示使用 TCP 帧的源端口数域和目的端口数域进行匹配。 只有当 L4SPM0 或 L4DPM0 位设置为高时才进行第 4 层匹配。 复位值：0
15:11	L3HDBM0	R_W	第 3 层 IP DA 高位匹配 IPv4 帧： 该字段包含 IPv4 帧中匹配的 IP 目标地址高位数。下面列出了该字段值的说明： <ul style="list-style-type: none"> <li>■ 0：不屏蔽任何位。</li> <li>■ 1：LSb[0]被屏蔽。</li> <li>■ 2：两个 LSbs [1:0]被屏蔽。</li> <li>■ ...</li> <li>■ 31：除 MSb 外的所有位都被屏蔽。</li> </ul> IPv6 帧： 该字段的位[12:11]对应 L3HSBM0 的位[6:5]，表示 IPv6 帧中被屏蔽的 IP 源地址或目标地址的低位数。下面的列表描述了 L3HDBM0[1:0]和 L3HSBM0 位的连接值： <ul style="list-style-type: none"> <li>■ 0：不屏蔽任何位。</li> <li>■ 1：LSb[0]被屏蔽。</li> <li>■ 2：两个 LSbs [1:0]被屏蔽。</li> <li>■ ...</li> <li>■ 127：除 MSb 外的所有位都被屏蔽。</li> </ul> 当 L3DAM0 或 L3SAM0 设置为高时，该字段才有效。 复位值：00H
10:6	L3HSBM0	R_W	第 3 层 IP SA 高位匹配 IPv4 帧： 该字段包含 IPv4 帧中被屏蔽匹配的 IP 源地址的低位数。下面列出了该字段值的说明：



位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0: 不屏蔽任何位。</li> <li>1: LSB[0]被屏蔽。</li> <li>2: 两个 LSBs [1:0]被屏蔽。</li> <li>...</li> <li>31: 除 MSb 外的所有位都被屏蔽。</li> </ul> IPv6 帧: 该字段包含表示 IPv6 帧中匹配的 IP 源地址或目标地址高位数的位[4:0]。 当 L3DAM0 或 L3SAM0 设置为高时, 该字段才有效。 复位值: 00H
5	L3DAIM0	R_W	第 3 层 IP DA 逆匹配使能 当设置此位时, 表示启用第 3 层 IP 目标地址字段进行反向匹配。复位时, 该位表示启用第 3 层 IP 目标地址字段进行完美匹配。 只有当位 4 (L3DAM0)设置为高时, 该位才有效。 复位值: 0
4	L3DAM0	R_W	第 3 层 IP DA 匹配使能 当设置时, 表示启用第 3 层 IP 目标地址字段进行匹配。复位时, MAC 忽略进行匹配的第 3 层目标地址字段。 注意: 当位 0 (L3PEN0)设置时, 应该设置该位或位 2 (L3SAM0), 因为 IPv6 DA 或 SA 都可以检查过滤。 复位值: 0
3	L3SAIM0	R_W	第 3 层 IP SA 逆匹配使能 当设置此位时, 表示启用第 3 层 IP 源地址字段进行逆匹配。复位时, 该位表示启用第 3 层 IP 源地址字段进行完美匹配。 只有当位 2 (L3SAM0)设置为高时, 该位才有效。 复位值: 0
2	L3SAM0	R_W	第 3 层 IP SA 匹配使能 当设置此位时, 表示启用第 3 层 IP 源地址字段进行匹配。复位时, MAC 忽略进行匹配的第 3 层 IP 源地址字段。 注意: 当位 0 (L3PEN0)设置时, 应该设置该位或位 4 (L3DAM0), 因为 IPv6 SA 或 DA 都可以检查过滤。

位段	名称	类型	说明
			复位值：0
1	-	RO	保留 复位值：0
0	L3PEN0	R_W	第 3 层协议启用 设置该位表示对 IPv6 帧启用第 3 层 IP 源地址或目标地址匹配。复位时，该位表示对 IPv4 帧启用第 3 层 IP 源地址或目标地址匹配。 只有当 L3SAM0 或 L3DAM0 位设置为高时，才进行第 3 层匹配。 复位值：0

### 1.6.3.54 寄存器 257（第 4 层地址寄存器 0）

通过在 coreConsultant 中选择同步第 3 层和第 4 层地址寄存器到 Rx 时钟域选项，可以配置第 3 层和第 4 层地址寄存器双同步。如果第 3 层和第 4 层地址寄存器被配置为双同步到 Rx 时钟域，那么只有当第 3 层和第 4 层地址寄存器的位[31:24](小端模式)或位[7:0](大端模式)被写入时才会触发同步。为了进行正确的同步更新，应该在目标时钟域经过至少 4 个时钟周期后，再对相同的第 3 层和第 4 层地址寄存器执行连续的写操作。

如果在核心配置时没有选择“第 3 层过滤”和“第 4 层过滤”，则保留该寄存器和寄存器 260 ~ 299（只读，默认值）。

图表 1-71 寄存器 257（第 4 层地址寄存器 0）

位段	名称	类型	说明
31:16	L4DP0	R_W	第 4 层目的端口数域 当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中复位位 16 (L4PEN0)，并设置位 20 (L4DPM0)时，该字段包含与 IPv4 或 IPv6 帧中 TCP 目的端口数域匹配的值。 当寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置 16 (L4PEN0)和位 20 (L4DPM0)时，该字段包含与 IPv4 或 IPv6 帧中 UDP 目的端口数域匹配的值。 复位值：0000H
15:0	L4SP0	R_W	第 4 层源端口数域 当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中复位位 16 (L4PEN0)和位 20 (L4DPM0)时，该字段包含与 IPv4 或 IPv6

位段	名称	类型	说明
			帧中 TCP 源端口数域匹配的值。  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 16 (L4PEN0)和位 20 (L4DPM0)时, 该字段包含与 IPv4 或 IPv6 帧中 UDP 源端口数域匹配的值。  复位值: 0000H

### 1.6.3.55 寄存器 260 (第 3 层地址 0 寄存器 0)

对于 IPv4 帧, 第 3 层地址 0 寄存器 0 包含 32 位的 IP 源地址字段。对于 IPv6 帧, 它包含 128 位 IP 源地址或目标地址字段的位[31:0]。

图表 1-72 寄存器 260 (第 3 层地址 0 寄存器 0)

位段	名称	类型	说明
31:0	L3A00	R_W	第 3 层地址 0 字段  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 2 (L3SAM0)时, 该字段包含与 IPv6 帧中 IP 源地址字段的位[31:0]相匹配的值。  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 4 (L3DAM0)时, 该字段包含与 IPv6 帧的 IP 目标地址字段的位[31:0]相匹配的值。  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中复位位 0 (L3PEN0), 并设置位 2 (L3SAM0)时, 该字段包含与 IPv4 帧中 IP 源地址字段匹配的值。  复位值: 00000000H

### 1.6.3.56 寄存器 261 (第 3 层地址 1 寄存器 0)

对于 IPv4 帧, 第 3 层地址 1 寄存器 0 包含 32 位的 IP 目标地址字段。对于 IPv6 帧, 它包含 128 位的 IP 源地址或目标地址字段的位[63:32]。

图表 1-73 寄存器 261 (第 3 层地址 1 寄存器 0)

位段	名称	类型	说明
31:0	L3A10	R_W	第 3 层地址 1 字段  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 2 (L3SAM0)时, 该字段包含与 IPv6 帧中 IP 源地址字段的位[63:32]相匹配的值。

位段	名称	类型	说明
			当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 4 (L3DAM0)时, 该字段包含与 IPv6 帧 IP 目标地址字段的位[63:32]相匹配的值。  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中复位位 0 (L3PEN0), 并设置位 4 (L3DAM0)时, 该字段包含与 IPv4 帧中与 IP 目标地址字段匹配的值。  复位值: 00000000H

### 1.6.3.57 寄存器 262 (第 3 层地址 2 寄存器 0)

对于 IPv4 帧, 第 3 层地址 2 寄存器 0 是保留的。对于 IPv6 帧, 它包含 128 位的 IP 源地址或目标地址字段的位[95:64]。

图表 1-74 寄存器 262 (第 3 层地址 2 寄存器 0)

位段	名称	类型	说明
31:0	L3A20	R_W	第 3 层地址 2 字段  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 2 (L3SAM0)时, 该字段包含与 IPv6 帧中 IP 源地址字段的位[95:64]相匹配的值。  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 4 (L3DAM0)时, 该字段包含与 IPv6 帧 IP 目标地址字段的位[95:64]相匹配的值。  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中复位位 0 (L3PEN0)时, 该寄存器不被使用。  复位值: 00000000H

### 1.6.3.58 寄存器 263 (第 3 层地址 3 寄存器 0)

对于 IPv4 帧, 第 3 层地址 3 寄存器 0 是保留的。对于 IPv6 帧, 它包含 128 位的 IP 源地址或目标地址字段的位[127:96]。

图表 1-75 寄存器 263 (第 3 层地址 3 寄存器 0)

位段	名称	类型	说明
31:0	L3A30	R_W	第 3 层地址 3 字段  当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 2 (L3SAM0)时, 该字段包含与 IPv6 帧中 IP 源

位段	名称	类型	说明
			地址字段的位[127:96]相匹配的值。 当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中设置位 0 (L3PEN0)和位 4 (L3DAM0)时, 该字段包含与 IPv6 帧 IP 目标地址字段的位[127:96]相匹配的值。 当在寄存器 256(第 3 层和第 4 层控制寄存器 0)中复位位 0 (L3PEN0)时, 该寄存器不被使用。 复位值: 00000000H

#### 说明

- 寄存器 268、280 和 292 类似寄存器 256(第 3 层和第 4 层控制寄存器 0)。
- 寄存器 269、281 和 293 类似寄存器 257(第 4 层地址寄存器 0)。
- 寄存器 272、284 和 296 类似寄存器 260(第 3 层地址 0 寄存器 0)。
- 寄存器 273、285 和 297 类似寄存器 261(第 3 层地址 1 寄存器 0)。
- 寄存器 274、286 和 298 类似寄存器 262(第 3 层地址 2 寄存器 0)。
- 寄存器 275、287 和 299 类似寄存器 263(第 3 层地址 3 寄存器 0)。
- 当在 coreConsultant 中选择一个以上的第 3 层和第 4 层过滤器时, 会出现寄存器 268 到 275。
- 当在 coreConsultant 中选择两个以上的第 3 层和第 4 层过滤器时, 会出现寄存器 280 到 287。
- 当在 coreConsultant 中选择四个第 3 层和第 4 层过滤器时, 会出现寄存器 292 到 299。

### 1.6.3.59 寄存器 320 (散列表寄存器 0)

当散列表的宽度为 128 位或 256 位时, 该寄存器包含散列表的前 32 位。您可以使用 coreConsultant 中的散列表大小选项指定散列表的宽度。

组地址过滤使用 128 位或 256 位散列表。对于 hash 过滤, 传入帧的目标地址的内容通过 CRC 逻辑传递, CRC 寄存器的高 7 位(256 位散列表为高 8 位)用于索引散列表的内容。最高位决定了要使用的寄存器(散列表寄存器 X), 最低 5 位决定了寄存器内的位。例如, 散列值 7b'1100000(128 位散列表中)选择散列表寄存器 3 的第 0 位, 散列值 8b'10111111(256 位散列表中)选择散列表寄存器 5 的第 31 位。

目标地址的散列值是这样计算的:

1. 为 DA 计算 32 位的 CRC(关于计算 CRC32 的步骤, 请参见 IEEE 802.3, 章节 3.2.8)。
2. 对步骤 1 中获取的值进行位反转。
3. 取步骤 2 中值的高 7(或 8)位。

如果寄存器对应位的值是 1'b1, 则该帧被接受。否则, 该帧被拒绝。如果在寄存器 1 (MAC 帧过滤器)中设置了位 1 (通过所有组播), 则不管组播散列值是多少, 所有组播帧都被接受。

如果配置散列表寄存器双同步到(G)MII 时钟域, 则只有当写入散列表寄存器 X 的位[31:24](小端模式)或位[7:0](大端模式)时才会触发同步。

#### 说明

如果启用了双同步，则应该在目标时钟域经过至少 4 个时钟周期后，再对这个寄存器执行连续的写操作。

图表 1-76 寄存器 320（散列表寄存器 0）

位段	名称	类型	说明
31:0	HT31T0	R_W	散列表的前 32 位 该字段包含散列表的前 32 位(31:0)。 复位值：0000_0000H

#### 说明

- 寄存器 321 到 327 类似寄存器 320(散列表寄存器 0)。
- Note 只有在核心配置时选择 256 位散列表时，才会出现寄存器 324 到 327。

### 1.6.3.60 寄存器 353（VLAN Tag 包含或替换寄存器）

VLAN Tag 包含或替换寄存器包含用于在传输帧中插入或替换的 VLAN Tag。只有在核心配置时选择了 Enable SA、VLAN 和 CRC Insertion on TX 选项时，该寄存器才会出现。

图表 1-77 寄存器 353（VLAN Tag 包含或替换寄存器）

位段	名称	类型	说明
31:20	-	RO	保留 复位值：000H
19	CSVL	R_W	C-VLAN 或 S-VLAN 设置该位后，在传输帧的第 13 字节和第 14 字节插入或替换 S-VLAN 类型(0x88A8)。复位该位时，在传输帧中插入或替换 C-VLAN 类型(0x8100)。 复位值：0
18	VLP	R_W	VLAN 优先级控制 设置该位后，控制位[17:16]用于 VLAN 的删除、插入和替换。当复位该位时，使用 mti_vlan_ctrl_i 控制输入，且位[17:16]被忽略。 复位值：0
17:16	VLC	R_W	传输帧中的 VLAN Tag 控制 <ul style="list-style-type: none"> <li>■ 2'b00: 不删除、插入或更换 VLAN 标签</li> <li>■ 2'b01: 删除 VLAN 标签</li> </ul> MAC 删除所有带有 VLAN tag 的传输帧的 VLAN 类型(13 和 14 字节)和 VLAN tag(15 和 16 字节)。

位段	名称	类型	说明
			<p>■ 2'b10: VLAN 标签插入</p> <p>MAC 在 13 和 14 字节插入类型值(0x8100/0x88a8)后, 在 15 和 16 字节插入 VLT。该操作将在所有传输帧上执行, 不管它们是否已经有 VLAN tag。</p> <p>■ 2'b11: 更换 VLAN 标签</p> <p>MAC 在所有 VLAN 类型传输帧 (13 和 14 字节为 0x8100/0x88a8)的 15 和 16 字节替换 VLT。</p> <p>注意: 对该字段的更改只在帧开始时生效。如果在传输帧时写入该寄存器字段, 则只有后续帧可以使用更新后的值, 即当前帧不使用更新后的值。</p> <p>复位值: 00</p>
15:0	VLT	R_W	<p>传输帧的 VLAN Tag</p> <p>该字段包含要插入或替换的 VLAN tag 的值。该值只能在传输线路不活动或初始化阶段更改。位[15:13]为用户优先级, 位 12 为 CFI/DEI, 位[11:0]为 VLAN tag 的 VID 字段。</p> <p>复位值: 0000H</p>

### 1.6.3.61 寄存器 354 (VLAN 散列表寄存器)

当在寄存器 7 (VLAN tag 寄存器)中设置位 19 (VTHM)时, 使用 16 位散列表进行基于 VLAN tag 的组地址过滤。Hash 过滤是对传入帧的 16 位 VLAN tag 或 12 位 VLAN ID(基于 VLAN tag 寄存器的第 16 位 ETV)的内容进行 CRC 逻辑处理, 计算出的 CRC 的高 4 位用于索引 VLAN 散列表的内容。例如, 散列值 4b'1000 选择 VLAN 散列表的第 8 位。

目标地址的散列值是这样计算的:

1. 计算 VLAN tag 或 ID 的 32 位 CRC(关于计算 CRC32 的步骤, 请参见 IEEE 802.3, 章节 3.2.8)。
2. 对步骤 1 中获取的值进行位反转。
3. 取步骤 2 中值的高 4 位。

如果寄存器对应位的值是 1'b1, 则该帧被接受。否则, 该帧被拒绝。如果配置散列表寄存器双同步到(G)MII 时钟域, 则只有当该寄存器的位[15:8] (小端模式)或位[7:0] (大端模式)被写入时才会触发同步。

#### 说明

- 如果启用了双同步, 则应该在目标时钟域经过至少 4 个时钟周期后, 再对此寄存器执行连续写操作。
- 为了帮助您编写散列表, 工作空间的/sample\_codes/目录中包含一个生成 4 位 hash 的示例 C 例程。

只有在核心配置期间启用 VLAN Hash 特性时, 该寄存器才有效和存在。

图表 1-78 寄存器 354 (VLAN 散列表寄存器)

位段	名称	类型	说明
31:16	-	RO	保留 复位值: 0000H
15:0	VLHT	R_W	VLAN 散列表 该字段包含 16 位 VLAN 散列表。 复位值: 0000H

### 1.6.3.62 寄存器 448 (时间戳控制寄存器)

该寄存器控制系统时间生成器的操作，以及控制接收端对 PTP 数据包进行时间戳的处理。

图表 1-79 寄存器 448 (时间戳控制寄存器)

位段	名称	类型	说明
31:29	-	RO	保留 复位值: 000
28	ATSEN3	R_W	辅助快照 3 启用 该字段控制捕获辅助快照触发器 3。设置此位后，启用 ptp_aux_trig_i[3] 输入上事件的辅助快照。当该位被复位时，该输入上的事件将被忽略。 当核心配置时没有选择“添加 IEEE 1588 辅助快照”选项或“IEEE 1588 辅助快照输入数量”选项中选择数字小于 4 时，该位为保留位。 复位值: 0
27	ATSEN2	R_W	辅助快照 2 启用 该字段控制捕获辅助快照触发器 2。设置此位后，启用 ptp_aux_trig_i[2] 输入上事件的辅助快照。当该位被复位时，该输入上的事件将被忽略。 当核心配置时没有选择“添加 IEEE 1588 辅助快照”选项或“IEEE 1588 辅助快照输入数量”选项中选择数字小于 3 时，该位为保留位。 复位值: 0
26	ATSEN1	R_W	辅助快照 1 启用 该字段控制捕获辅助快照触发器 1。设置此位后，启用 ptp_aux_trig_i[1] 输入上事件的辅助快照。当该位被复位时，



位段	名称	类型	说明
			该输入上的事件将被忽略。 当核心配置时没有选择“添加 IEEE 1588 辅助快照”选项或“IEEE 1588 辅助快照输入数量”选项中选择数字小于 2 时，该位为保留位。 复位值：0
25	ATSENO	R_W	辅助快照 0 启用 该字段控制捕获辅助快照触发器 0。设置此位后，启用 ptp_aux_trig_i[0] 输入上事件的辅助快照。当该位被复位时，该输入上的事件将被忽略。 当在核心配置期间没有选择“添加 IEEE 1588 辅助快照”选项时，该位为保留位。 复位值：0
24	ATSFC	R_WS_SC	辅助快照 FIFO 清除 当设置时，它复位辅助快照 FIFO 的指针。当指针复位及 FIFO 为空时，该位被清除。当该位设置为高时，辅助快照存储在 FIFO 中。当在核心配置期间没有选择“添加 IEEE 1588 辅助快照”选项时，该位为保留位。 复位值：0
23:19	-	RO	保留 复位值：0
18	TSENMADDR	R_W	使能 PTP 帧过滤的 MAC 地址 设置后，当 PTP 直接通过以太网发送时，DA MAC 地址（与任何 MAC 地址寄存器匹配）用于过滤 PTP 帧。 复位值：0
17:16	SNAPTYPSEL	R_W	选择用于拍摄快照的 PTP 数据包 这些位与位 15 和位 14 一起决定需要拍摄快照的 PTP 数据包类型集。第 462 页表 6-70 给出了编码。 复位值：00
15	TSMSTRENA	R_W	为主服务器相关的消息启用快照 设置后，仅为与主节点相关的消息拍摄快照。否则，将为与从节点相关的消息拍摄快照。 复位值：0

位段	名称	类型	说明
14	TSEVENTENA	R_W	<p>为事件消息启用时间戳快照</p> <p>设置后，仅为事件消息（SYNC、Delay_Req、Pdelay_Req 或 Pdelay_Resp）拍摄时间戳快照。复位时，将为除公告、管理和信令之外的所有消息拍摄快照。有关时间戳快照的更多信息，请参阅第 462 页的表 6-70。</p> <p>复位值：0</p>
13	TSIPV4ENA	R_W	<p>启用对通过 IPv4-UDP 发送的 PTP 帧的处理</p> <p>设置后，MAC 接收器处理封装在 UDP-IPv4 数据包中的 PTP 数据包。如果清除此位，MAC 将忽略通过 UDP-IPv4 数据包传输的 PTP 数据包。此位为默认设置。</p> <p>复位值：1</p>
12	TSIPV6ENA	R_W	<p>启用对通过 IPv6-UDP 发送的 PTP 帧的处理</p> <p>设置后，MAC 接收器处理封装在 UDP-IPv6 数据包中的 PTP 数据包。如果清除此位，MAC 将忽略通过 UDP-IPv6 数据包传输的 PTP 数据包。</p> <p>复位值：0</p>
11	TSIPENA	R_W	<p>使能通过以太网帧处理 PTP</p> <p>设置后，MAC 接收器处理直接封装在以太网帧中的 PTP 数据包。当该位清除时，MAC 忽略以太网 PTP 数据包。</p> <p>复位值：0</p>
10	TSVER2ENA	R_W	<p>启用版本 2 格式的 PTP 数据包处理</p> <p>设置后，使用 1588 版本 2 格式处理 PTP 数据包。否则，将使用版本 1 格式处理 PTP 数据包。关于 IEEE 1588 版本 1 和版本 2 格式，请参见第 155 页的《PTP 处理和控制》。</p> <p>复位值：0</p>
9	TSCTRLSSR	R_W	<p>时间戳数字或二进制翻转控制</p> <p>当设置时，时间戳低位寄存器在 0x3B9A_C9FF 值(即 1 纳秒精度)之后翻转，并增加时间戳(高位)的秒。复位时，亚秒寄存器的翻转值为 0x7FFF_FFFF。亚秒增量必须根据 PTP 参考时钟频率和该位的值进行正确编程。</p> <p>复位值：0</p>
8	TSENALL	R_W	<p>为所有帧启用时间戳</p>

位段	名称	类型	说明
			设置时，MAC 接收到的所有帧都启用时间戳快照。 复位值：0
7:6	-	RO	保留 复位值：00
5	TSADDREG	R_WS_SC	加数寄存器更新 当设置时，时间戳加数寄存器的内容在 PTP 块中更新以进行精细校正。更新完成时将清除该位。该寄存器位在设置之前应该为 0。 复位值：0
4	TSTRIG	R_WS_SC	使能时间戳中断触发器 当设置时，当系统时间大于写入目标时间寄存器的值时，将生成时间戳中断。 该位在时间戳触发器中断生成后复位。 复位值：0
3	TSUPDT	R_WS_SC	更新时间戳 设置后，系统时间将以寄存器 452(系统时间-秒更新寄存器)和寄存器 453(系统时间-纳秒更新寄存器)中指定的值更新(加或减)。 在更新该位之前，该位应该为 0。当硬件更新完成时，该位被复位。时间戳高位字寄存器(如果在核心配置期间启用)不被更新。 复位值：0
2	TSINIT	R_WS_SC	时间戳初始化 设置该位时，系统时间初始化(覆盖)为寄存器 452(系统时间-秒更新寄存器)和寄存器 453(系统时间-纳秒更新寄存器)中指定的值。 在更新该位之前，该位应该为 0。当初始化完成时，该位被复位。时间戳高位字寄存器(如果在核心配置期间启用)只能初始化。 复位值：0
1	TSCFUPDT	R_W	时间戳精更新或粗更新 当设置此位时，表示应该使用精更新方式完成系统时间更新。

位段	名称	类型	说明
			复位时，表示系统时间戳更新应该使用粗更新方式进行。 复位值：0
0	TSENA	R_W	时间戳启用 设置后，发送帧和接收帧将添加时间戳。禁用时，发送帧和接收帧不会添加时间戳，时间戳生成器也会挂起。启用该模式后，需要初始化时间戳(系统时间)。 只有设置了该位，在接收端，MAC 才会处理 1588 帧。 复位值：0

图表 1-80 表示根据寄存器 448(时间戳控制寄存器)中位[17:14] (SNAPTYPSEL)获取的 PTP 消息的快照。

图表 1-80 间戳快照对寄存器位的依赖性

SNAPTYPSEL (Bits 17:16)	TSMSTRENA (Bit 15)	TSEVNTENA (Bit 14)	PTP Messages
00	X	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	X	0	SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up
01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	X	X	SYNC, Delay_Req
11	X	X	Pdelay_Req, Pdelay_Resp

### 1.6.3.63 寄存器 449（亚秒级增量寄存器）

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时，才会出现此寄存器。在粗更新模式下(寄存器 448 中的 TSCFUPDT 位)，在 clk\_ptp\_ref\_i 的每个时钟周期，该寄存器中的值被添加到系统时间。在精更新模式中，当累加器溢出时，此寄存器中的值将添加到系统时间。

图表 1-81 寄存器 449（亚秒级增量寄存器）

位段	名称	类型	说明
31:8	-	RO	保留 复位值：000000H

位段	名称	类型	说明
7:0	SSINC	R_W	亚秒级增量值  在该字段中编程的值在每个时钟周期(clk_ptp_i)和亚秒寄存器的内容中累积。例如，当 PTP 时钟为 50 MHz(周期为 20 ns)时，当系统时间-纳秒寄存器的精度为 1 ns[在寄存器 448(时间戳控制寄存器)中设置位 9 (TSCTRLSSR)]时，您应该编程 20 (0x14)。当 TSCTRLSSR 被清除时，纳秒寄存器的分辨率为 ~0.465ns。在这种情况下，应该编写一个由 20ns/0.465 导出的值 43 (0x2B)。  复位值：00H

### 1.6.3.64 寄存器 450（系统时间-秒寄存器）

系统时间-秒寄存器和系统时间-纳秒寄存器是 MAC 维护的系统时间的当前值。虽然它是连续更新的，但由于时钟域传输延迟(从 clk\_ptp\_ref\_i 到 CSR 时钟)，该值与实际时间有一定的延迟。

这些寄存器(450 和 451)只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时才会出现。

图表 1-82 寄存器 450（系统时间-秒寄存器）

位段	名称	类型	说明
31:0	TSS	RO	时间戳秒  该字段表示 MAC 维护的系统时间的当前值，单位为秒。  复位值：00000000H

### 1.6.3.65 寄存器 451（系统时间-亚秒寄存器）

图表 1-83 寄存器 451（系统时间-亚秒寄存器）

位段	名称	类型	说明
31	-	RO	保留  复位值：0
30:0	TSSS	RO	时间戳亚秒  该字段的值表示以亚秒为单位时间，精度为 0.46 ns。在寄存器 448 (时间戳控制寄存器)中设置第 9 位(TSCTRLSSR)时，每一位表示 1ns，最大值为 0x3B9A_C9FF，之后归零。  复位值：00000000H

### 1.6.3.66 寄存器 452（系统时间-秒更新寄存器）

系统时间-秒更新寄存器和系统时间-纳秒更新寄存器初始化或更新 MAC 维护的系统时间。在时间戳控制寄存器中设置 TSINIT 或 TSUPDT 位之前，必须写入这两个寄存器。只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时，才会出现此寄存器。

图表 1-84 寄存器 452（系统时间-秒更新寄存器）

位段	名称	类型	说明
31:0	TSS	R_W	时间戳秒 该字段的值表示初始化或添加到系统时间中的时间，单位为秒 复位值：00000000H

### 1.6.3.67 寄存器 453（系统时间-纳秒更新寄存器）

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时，才会出现此寄存器。

图表 1-85 寄存器 453（系统时间-纳秒更新寄存器）

位段	名称	类型	说明
31	ADDSUB	R_W	增加或减少时间 设置此位时，时间值与更新寄存器的内容相减。复位此位时，时间值与更新寄存器的内容相加。 复位值：0
30:0	TSSS	R_W	时间戳亚秒 该字段的值表示以亚秒为单位的时间，精度为 0.46 ns。在寄存器 448 (时间戳控制寄存器)中设置位 9 (TSCTRLSSR)时，每个位代表 1ns，编程值不应超过 0x3B9A_C9FF。 复位值：00000000H

### 1.6.3.68 寄存器 454（时间戳加数寄存器）

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时，才会出现此寄存器。该寄存器的值仅在将系统时间配置为精更新模式(寄存器 448 中的 TSCFUPDT 位)时使用。在每个时钟周期(clk\_ptp\_ref\_i)，该寄存器内容被添加到一个 32 位的累加器中。每当累加器溢出时，系统时间就会更新。

图表 1-86 寄存器 454（时间戳加数寄存器）

位段	名称	类型	说明
31:0	TSAR	R_W	时间戳加数寄存器 该字段表示在累加器寄存器中加入 32 位时间值, 以实现时间同步。 复位值: 00000000H

### 1.6.3.69 寄存器 455（目标时间秒寄存器）

当系统时间超过这些寄存器中的编程值时, 目标时间秒寄存器和目标时间纳秒寄存器用于调度中断事件 (当启用高级时间戳时, 为寄存器 458[1]; 否则, 为寄存器 14[9]中的 TS 中断位)。

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时, 才会出现此寄存器。

图表 1-87 寄存器 455（目标时间秒寄存器）

位段	名称	类型	说明
31:0	TSTR	R_W	目标时间秒寄存器 该寄存器以秒为单位存储时间。当时间戳值同时匹配或超过这两个目标时间戳寄存器时, 基于寄存器 459(PPS 控制寄存器)的位[6:5], MAC 启动或停止 PPS 信号输出并产生中断 (如果启用)。 复位值: 00000000H

### 1.6.3.70 寄存器 456（目标时间纳秒寄存器）

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳特性时, 才会出现此寄存器。

图表 1-88 寄存器 456（目标时间纳秒寄存器）

位段	名称	类型	说明
31	TRGTBUSY	R_WS_SC	目标时间寄存器忙 当寄存器 459 (PPS 控制寄存器) 中的 PPSCMD 字段 (位[3:0]) 被编程为 010 或 011 时, MAC 设置此位。 编程 PPSCMD 字段到 010 或 011, 指示 MAC 同步目标时间寄存器到 PTP 时钟域。 MAC 在同步目标时间寄存器到 PTP 时钟域后清除该位。当读取该位为 1 时, 应用程序不能更新目标时间寄存器。否则, 之前编程时间的同步会被破坏。当没有选择“启用灵活每秒脉冲输出 (Enable Flexible Pulse-Per-Second Output)”特

位段	名称	类型	说明
			性时，该位是保留位。 复位值：0
30:0	TTSLO	R_W	目标时间戳低位寄存器 该寄存器以(带符号的)纳秒来存储时间。当时间戳的值与两个目标时间戳寄存器相匹配时，MAC 根据寄存器 459 (PPS 控制寄存器)中的 TRGTMODSEL0 字段(位[6:5])启动或停止 PPS 信号输出并产生中断(如果启用)。 当寄存器 448(时间戳控制寄存器)中设置位 9(TSCTRLSSR)时，该值不应超过 0x3B9A_C9FF。PPS 信号输出的实际启动或停止时间可具有最大一个单位亚秒增量值的误差。 复位值：00000000H

### 1.6.3.71 寄存器 457（系统时间-高字秒寄存器）

只有在没有外部时间戳输入的情况下选择 IEEE 1588 高级时间戳特性时，才会出现此寄存器。

图表 1-89 寄存器 457（系统时间-高字秒寄存器）

位段	名称	类型	说明
31:16	-	RO	保留 复位值：0000H
15:0	TSHWR	R_W_SU	时间戳高字寄存器 此字段包含时间戳秒值的最高有效 16 位。该寄存器是可选的，可在核心配置期间使用启用 IEEE 1588 高位字寄存器选项进行选择。该寄存器被直接写入以初始化值。当系统时间-秒寄存器的 32 位溢出时，该寄存器将递增。 复位值：0000H

### 1.6.3.72 寄存器 458（时间戳状态寄存器）

只有选择高级 IEEE 1588 时间戳特性时，才会出现此寄存器。当主机读取该寄存器时，除了位[27:25]以外的所有位都被清除。

图表 1-90 寄存器 458（时间戳状态寄存器）

位段	名称	类型	说明
31:30	-	RO	保留



位段	名称	类型	说明
			复位值：00
29:25	ATSNS	RO	辅助时间戳快照个数  该字段表示 FIFO 中可用的快照数量。与选择的 FIFO 深度(4、8 或 16)相同的值表示辅助快照 FIFO 已满。当设置辅助快照 FIFO 清除位被设置时，这些位被清除(到 00000)。只有在核心配置时选择了“添加 IEEE 1588 辅助快照 (Add IEEE 1588 Auxiliary Snapshot)”选项时，该位才有效。  复位值：00000
24	ATSSTM		缺少辅助时间戳快照触发器  当辅助时间戳快照 FIFO 满并设置外部触发器时设置该位。该位表明最新的快照不存储在 FIFO 中。只有在核心配置时选择了“添加 IEEE 1588 辅助快照”选项时，该位才有效。
23:20	-	RO	保留  复位值：0H
19:16	ATSSTN	R_SS_RC	辅助时间戳快照触发器标识符  这些位标识辅助快照寄存器中可用的时间戳适用的辅助触发器输入。当同时设置多位时，表示在同一时钟采样相应的辅助触发器。这些位只适用于辅助快照的数量多于一个的情况。每个触发器分配了一个位，如下表所示： <ul style="list-style-type: none"> <li>■ 位 16：辅助触发器 0</li> <li>■ 位 17：辅助触发器 1</li> <li>■ 位 18：辅助触发器 2</li> <li>■ 位 19：辅助触发器 3</li> </ul> 软件可以读取该寄存器，以找到在获取时间戳时设置的触发器。  复位值：0000
15:10	-	RO	保留  复位值：00H
9	TSTRGTERR3	R_SS_RC	时间戳目标时间错误  该位是在寄存器 496 和寄存器 497 中编程的目标时间已经过去时设置的。该位在被应用程序读取时被清除。  复位值：0

位段	名称	类型	说明
8	TSTARGET3	R_SS_RC	时间戳目标时间到达目标时间 PPS3  当设置此位时，表示系统时间的值大于或等于寄存器 496 (PPS3 目标时间高位寄存器)和寄存器 497 (PPS3 目标时间低位寄存器)中指定的值。  复位值：0
7	TSTRGTERR2	R_SS_RC	时间戳目标时间错误  该位是在寄存器 488 和寄存器 489 中编程的目标时间已经过去时设置的。该位在被应用程序读取时被清除。  复位值：0
6	TSTARGET2	R_SS_RC	时间戳目标时间到达目标时间 PPS2  该位表示系统时间的值大于或等于寄存器 488 (PPS2 目标时间高位寄存器)和寄存器 489(PPS2 目标时间低位寄存器)中指定的值。  复位值：0
5	TSTRGTERR1	R_SS_RC	时间戳目标时间错误  该位是在寄存器 480 和寄存器 481 中编程的目标时间已经过去时设置的。该位在被应用程序读取时被清除。  复位值：0
4	TSTARGET1	R_SS_RC	时间戳目标时间到达目标时间 PPS1  该位表示系统时间的值大于或等于寄存器 480 (PPS1 目标时间高位寄存器)和寄存器 481(PPS1 目标时间低位寄存器)中指定的值。  复位值：0
3	TSTRGTERR	R_SS_RC	时间戳目标时间错误  该位是在寄存器 455 和寄存器 456 中编程的目标时间已经过去时设置的。该位在被应用程序读取时被清除。  复位值：0
2	AUXTSTRIG	R_SS_RC	辅助时间戳触发器快照  当将辅助快照写入 FIFO 时，该位被设置为高。只有勾选“启用 IEEE 1588 辅助快照”特性时，该位才有效。  复位值：0
1	TSTARGET	R_SS_RC	时间戳目标时间到达

位段	名称	类型	说明
			当设置此位时，表示系统时间的值大于或等于寄存器 455(目标时间秒寄存器)和寄存器 456(目标时间纳秒寄存器)中指定的值。 复位值：0
0	TSSOVF	R_SS_RC	时间戳秒溢出 当设置此位时，表示时间戳的秒值(当支持版本 2 格式时)已经溢出超过 32'hFFFF_FFFF。 复位值：0

### 1.6.3.73 寄存器 459 (PPS 控制寄存器)

只有在选择高级时间戳特性且未启用外部时间戳时，才会出现此寄存器。

#### 说明

- 位[30:24] 仅当选择四个 Flexible PPS 输出时有效。
- 位[22:16] 仅当选择三个或更多 Flexible PPS 输出时有效。
- 位[14:8] 仅当选择两个或多个 Flexible PPS 输出时有效。
- 位[6:4] 仅当选择 Flexible PPS 特性时有效。

图表 1-91 寄存器 459 (PPS 控制寄存器)

位段	名称	类型	说明
26:24	PPSCMD3	R_WS_SC	Flexible PPS3 输出控制 该字段控制 Flexible PPS3 输出(ptp_pps_o[3])信号。该字段在功能上类似于 PPSCMD0[2:0]。 复位值：000
23	-	RO	保留 复位值：0
22:21	TRGTMODSEL2	R_W	PPS2 输出的目标时间寄存器模式 该字段表示 PPS2 输出信号的目标时间寄存器(寄存器 488 和 489)模式。 该字段类似于 TRGTMODSEL0 字段。 复位值：00
20:19	-	RO	保留 复位值：00

位段	名称	类型	说明
18:16	PPSCMD2	R_WS_SC	Flexible PPS2 输出控制 该字段控制 Flexible PPS2 输出(ptp_pps_o[2])信号。该字段在功能上类似于 PPSCMD0[2:0]。 复位值: 000
15	-	RO	保留 复位值: 0
14:13	TRGTMODSEL1	R_W	PPS1 输出的目标时间寄存器模式 该字段表示 PPS1 输出信号的目标时间寄存器(寄存器 480 和 481)模式。 该字段类似于 TRGTMODSELO 字段。 复位值: 00
12:11	-	RO	保留 复位值: 00
10:8	PPSCMD1	R_WS_SC	Flexible PPS1 输出控制 该字段控制 Flexible PPS1 输出(ptp_pps_o[1])信号。该字段在功能上类似于 PPSCMD0[2:0]。 复位值: 00
7	-	RO	保留 复位值: 0
6:5	TRGTMODSELO	R_W	PPS0 输出的目标时间寄存器模式 该字段表示 PPS0 输出信号的目标时间寄存器(寄存器 455 和 456)模式: <ul style="list-style-type: none"> <li>00: 指示目标时间寄存器仅为产生中断事件而编程。</li> <li>01: 保留</li> <li>10: 指示目标时间寄存器为产生中断事件和启动或停止 PPS0 输出信号的生成而编程。</li> <li>11: 表示目标时间寄存器仅用于启动或停止 PPS0 输出信号的生成。不断言中断。</li> </ul> 复位值: 00
4	PPSENO	R_W	Flexible PPS 输出模式启用 当设置为低时, 位[3:0]用作 PPSCTRL(向后兼容)。

位段	名称	类型	说明
			当设置高时, 位[3:0]用作 PPSCMD。 复位值: 0
3:0	PPSCTRL0	R_W	<p>PPSCTRL0: PPS0 输出频率控制</p> <p>该字段控制 PPS0 输出(ptp_pps_o[0])信号的频率。PPSCTRL 默认值为 0000,PPS 输出为每秒 1 次脉冲(宽度为 clk_ptp_i)。对于 PPSCTRL 的其他值, PPS 输出变成以下频率的生成时钟:</p> <ul style="list-style-type: none"> <li>■ 0001: 二进制翻转为 2 Hz, 数字翻转为 1 Hz</li> <li>■ 0010: 二进制翻转为 4 Hz, 数字翻转为 2 Hz</li> <li>■ 0011: 二进制翻转为 8 Hz, 数字翻转为 4 Hz</li> <li>■ 0100: 二进制翻转为 16 Hz, 数字翻转为 8 Hz</li> <li>■ ...</li> <li>■ 1111: 二进制翻转为 32.768 KHz, 数字翻转为 16.384 KHz</li> </ul> <p>注意:</p> <p>在二进制翻转模式中, PPS 输出(ptp_pps_o)在这些频率中占空比为 50%。</p> <p>在数字翻转模式中, PPS 输出频率为平均值。实际的时钟有不同的频率, 每秒同步一次。例如:</p> <ul style="list-style-type: none"> <li>■ 当 PPSCTRL = 0001 时, PPS (1 Hz)的低周期为 537 ms, 高周期为 463 ms</li> <li>■ 当 PPSCTRL = 0010 时, PPS (2 Hz)为以下序列: 一个 50%占空比和 537 ms 周期的时钟 第二个时钟为 463 ms 周期(低电平为 268 ms,高电平为 195 ms)</li> <li>■ 当 PPSCTRL = 0011 时, PPS (4 Hz) 为以下序列: 三个占空比 50%和 268 ms 周期的时钟 第四个时钟为 195 ms 周期 (低电平为 134 ms,高电平为 61 ms)。这种行为是由于寄存器 451(系统时间-纳秒寄存器)数字翻转模式下位的非线性切换造成的。</li> </ul> <p>复位值: 0H</p>

### 1.6.3.74 寄存器 460（辅助时间戳-纳秒寄存器）

这个寄存器和寄存器 461(辅助时间戳-秒寄存器)一起给出了存储为辅助快照的 64 位时间戳。这两个寄存器一起构成 FIFO 的读端口，宽度为 64 位，深度为 4、8 或 16。这是在核心配置期间选择的。该 FIFO 中可以存储多个快照。时间戳状态寄存器中的 ATSNS 位表示该 FIFO 的充满率。只有当寄存器 461(辅助时间戳-秒寄存器)的最后一个字节被读取时，FIFO 的顶部才被移除。在小端模式中，这意味着读取位[31:24]。在大端模式下，它对应于读取寄存器 461(辅助时间戳-秒寄存器)的位[7:0]。

只有在核心配置期间选择辅助快照启用特性时，才会出现此寄存器和寄存器 461(辅助时间戳-秒寄存器)。

图表 1-92 寄存器 460（辅助时间戳-纳秒寄存器）

位段	名称	类型	说明
31:26	-	RO	保留 复位值：00H
25:24	PTPCH	R_W	PTP 包排队通道 该字段指定了通过以太网负载(而不是通过 IPv4 或 IPv6)发送的未标记 PTP 包排队的通道。 <ul style="list-style-type: none"> <li>00: 通道 0</li> <li>01: 通道 1</li> <li>10: 通道 2</li> <li>11: 保留</li> </ul> 如果通道 1 或通道 2 的接收路径未启用，这些位是保留位。 复位值：00
23	-	RO	保留 复位值：0
22:21	AVCH	R_W	AV 控制包排队通道 该字段指定接收到的未标记 AV 控制包排队的通道。 <ul style="list-style-type: none"> <li>00: 通道 0</li> <li>01: 通道 1</li> <li>10: 通道 2</li> <li>11: 保留</li> </ul> 如果通道 1 或通道 2 的接收路径未启用，这些位是保留位。 复位值：00
20	AVCD	R_W	AV 通道禁用

位段	名称	类型	说明
			设置该位后，MAC 将所有数据包转发到缺省的 Channel 0，并忽略 AVP、AVCH 和 PTPCH 中编程的值。 如果在核心配置时没有选择通道 1 或通道 2 的接收路径，该位是保留位，并且是为只读。 复位值：0
19	VQE	R_W	VLAN Tagged 非 AV 数据包排队使能 设置该位后，MAC 根据 AVP 位的值，将带 VLAN 标签的非 AV 数据包排队到可用通道中。 如果在核心配置时没有选择通道 1 和通道 2 的接收路径，该位是保留位，并且为只读。 复位值：0
18:16	AVP	R_W	AV 排队优先级 在这些位中编程的值控制接收通道(0、1 或 2)，其中的 AV 数据包按照给定优先级排队。 <ul style="list-style-type: none"> <li>如果只启用通道 1 的接收路径，则优先级大于或等于编程值的 AV 数据包在通道 1 排队，其他数据包在通道 0 排队。</li> <li>如果同时开启了通道 2 接收路径，则优先级大于或等于编程值的 AV 数据包将在通道 2 排队，小于编程值的 AV 数据包将在通道 1 排队。其他所有数据包在通道 0 排队。</li> </ul> 只有在 AV 模式中选择了至少一个额外的接收通道时，这些位才适用。 复位值：000
15:0	AVT	R_W	AV 以太网类型值 该字段与传入的(tagged 或 untagged)以太网帧的以太网类型字段进行比较，用于检测 AV 数据包。 复位值：0000H

### 1.6.3.75 寄存器 461（辅助时间戳-秒寄存器）

图表 1-93 寄存器 461（辅助时间戳-秒寄存器）

位段	名称	类型	说明
31:0	AUXTSHI	RO	包含辅助时间戳 Seconds 字段的低 32 位。

位段	名称	类型	说明
			复位值：00000000H

### 1.6.3.76 寄存器 462 (AV MAC 控制寄存器)

该寄存器通过识别 AV 流量并将其列队到适当的通道来控制 AV 流量。只有在核心配置时选择 AV 特性时才会出现此寄存器。

图表 1-94 寄存器 462 (AV MAC 控制寄存器)

位段	名称	类型	说明
22:21	AVCH	R_W	AV 控制包排队通道 该字段指定接收到的 untagged AV 控制包排队的通道。 <ul style="list-style-type: none"> <li>00: 通道 0</li> <li>01: 通道 1</li> <li>10: 通道 2</li> <li>11: 保留</li> </ul> 如果通道 1 或通道 2 的接收路径未启用，这些位是保留位。 复位值：00
20	AVCD	R_W	AV 通道禁用 设置该位后，MAC 将所有数据包转发到缺省的 Channel 0，并忽略 AVP、AVCH 和 PTPCH 中编程的值。 如果在核心配置时没有选择通道 1 或通道 2 的接收路径，该位是保留位，并且是只读的。 复位值：0
19	VQE	R_W	VLAN Tagged 非 AV 数据包排队使能 设置该位后，MAC 根据 AVP 位的值，会将带 VLAN 标签的非 AV 数据包排队到可用通道中。 如果在核心配置时没有选择通道 1 和通道 2 的接收路径，该位是保留位，并且是只读的。 复位值：0
18:16	AVP	R_W	AV 排队优先级 在这些位中编程的值控制接收通道(0、1 或 2)，其中的 AV 数据包必须按照给定优先级排队进入该接收通道。 <ul style="list-style-type: none"> <li>如果只启用通道 1 的接收路径，则优先级大于或等于编程</li> </ul>



位段	名称	类型	说明
			值的 AV 数据包在通道 1 排队，其他数据包在通道 0 排队。 ■ 如果同时开启了通道 2 接收路径，则优先级大于或等于编程值的 AV 数据包将在通道 2 排队，小于编程值的 AV 数据包将在通道 1 排队。其他所有数据包在通道 0 排队。 只有在 AV 模式中选择了至少一个额外的接收通道时，这些位才适用。 复位值：000
15:0	AVT	R_W	AV 以太网类型值 该字段与传入的(tagged 或 untagged)以太网帧的以太网类型字段进行比较，用于检测 AV 数据包。 复位值：0000H

### 1.6.3.77 寄存器 472（PPS0 间隔寄存器）

PPS0 间隔寄存器包含 PPS0 信号输出(otp\_pps\_o[0])的上升沿之间亚秒增量值的单位数。

图表 1-95 寄存器 472（PPS0 间隔寄存器）

位段	名称	类型	说明
31:0	PPSINT	R_W	PPS0 输出信号间隔 这些位以亚秒增量值单位存储 PPS0 信号输出上升沿之间的间隔。 需要编写一个比所需间隔小的值。例如，如果 PTP 参考时钟是 50mhz(周期为 20ns)，PPS0 信号输出上升沿之间的期望间隔为 100ns(即 5 个单位的亚秒增量值)，那么应该在这个寄存器中编程值 4 (5 - 1)。 复位值：00000000H

### 1.6.3.78 寄存器 473（PPS0 宽度寄存器）

PPS0 宽度寄存器包含 PPS0 信号输出(otp\_pps\_o[0])上升沿和相应下降沿之间亚秒增量值的单位数。

图表 1-96 寄存器 473（PPS0 宽度寄存器）

位段	名称	类型	说明
31:0	PPSWIDTH	R_W	PPS0 输出信号宽度 这些位以亚秒增量值单位存储 PPS0 信号输出的上升沿和相

位段	名称	类型	说明
			<p>应下降沿之间的宽度。</p> <p>需要编程一个小于所需间隔的值。例如，如果 PTP 参考时钟为 50 MHz（周期为 20ns），并且 PPS0 信号输出的上升沿和相应下降沿之间的所需宽度为 80ns（即四个单位的亚秒增量值），则应在该寄存器中编程值 3（4 - 1）。</p> <p>注意：该寄存器中编程的值必须小于寄存器 472（PPS0 间隔寄存器）中编程的值。</p> <p>复位值：00000000H</p>

### 1.6.3.79 寄存器 480（PPS1 目标时间秒寄存器）

当系统时间超过这些寄存器中编程的值时，PPS1 目标时间秒寄存器以及 PPS1 目标时间纳秒寄存器用于调度中断事件（寄存器 458（时间戳状态寄存器）中的位 1（TSTARGT））。

只有在核心配置期间选择了多个 Flexible PPS 输出时，才存在该寄存器。

图表 1-97 寄存器 480（PPS1 目标时间秒寄存器）

位段	名称	类型	说明
31:0	TSTRH1	R_W	<p>PPS1 目标时间秒寄存器</p> <p>此寄存器以秒为单位存储时间。当时间戳值匹配或超过两个目标时间戳寄存器时，则基于寄存器 459（PPS 控制寄存器）的位[14:13]和 TRGTMODSEL1，MAC 启动或停止 PPS 信号输出并生成中断（如果启用）。</p> <p>复位值：00000000H</p>

### 1.6.3.80 寄存器 481（PPS1 目标时间纳秒寄存器）

只有在核心配置期间选择了多个 Flexible PPS 输出时，才存在该寄存器。

图表 1-98 寄存器 481（PPS1 目标时间纳秒寄存器）

位段	名称	类型	说明
31	TRGTBUSY1	R_WS_SC	<p>PPS1 目标时间寄存器忙</p> <p>当寄存器 459（PPS 控制寄存器）中的 PPSCMD1 字段（位[10:8]）编程为 010 或 011 时，MAC 设置该位。将 PPSCMD1 字段编程为 010 或 011 指示 MAC 将目标时间寄存器同步到 PTP 时钟域。</p> <p>MAC 在将目标时间寄存器同步到 PTP 时钟域后清除该位。当该位读取为 1 时，应用程序不得更新目标时间寄存器。否则，</p>

位段	名称	类型	说明
			先前编程时间的同步会损坏。 复位值：0
30:0	TTSL1	R_W	PPS1 寄存器的目标时间低位 此寄存器以（有符号的）纳秒为单位存储时间。当时间戳的值与两个目标时间戳寄存器匹配时，则基于寄存器 459（PPS 控制寄存器）中的 TRGTMODSEL1 字段（位[14:13]），MAC 启动或停止 PPS 信号输出并生成中断（如果启用）。 当寄存器 448（时间戳控制寄存器）中设置位 9（TSCTRLSSR）时，该值不应超过 0x3B9A_C9FF。PPS 信号输出的实际开始或停止时间可能具有最大一个单位亚秒增量值的误差。 复位值：00000000H

## 2 USB

### 2.1 概述

该芯片设计了 USB3 DRD 模块。它支持 DRD 模式，支持设备或主机单独操作，而不是同时操作。USB3 DRD 模块支持 USB 3.0 超高速（5Gbps）协议和数据速率，并向后兼容 USB 2.0 高速（480Mbps）、全速（12Mbps）和低速（1.5Mbps）协议及数据速率。USB3 DRD 模块包括 USB3 DRD 控制器和物理层。

USB3 DRD 模块支持 xHCI1.1，可以使用标准或开源 xHCI 和类驱动程序。

它可以用于便携式电子设备，如高带宽大容量存储磁盘和高带宽应用。

### 2.2 主要特性

- USB3 DRD 模块支持的主要特性如下：
- 支持 USB3.0 标准协议，可配置为设备或主机模式
- 支持设备或主机单独操作，而不是同时操作
- 支持超高速（5Gbps 输入和 5Gbps 输出）、高速（480Mbps）、全速（12Mbps）、低速（1.5Mbps，仅限主机模式）
- 支持 DMA 访问（内部 DMA 控制器）
- 支持控制传输、中断传输、批量传输、同步传输
- 省电功能（时钟门控）
- 描述符缓存和数据预取，用于满足高延迟系统中的系统性能
- 超高速（SS）、高速（HS）、全速（FS）和低速（LS）编程模型相同
- 兼容 xHCI1.1
- 标准或开源 xHCI 和类驱动程序

### 2.3 接口

USB 的 PAD 如下图表 2-1 所述：

图表 2-1 引脚描述

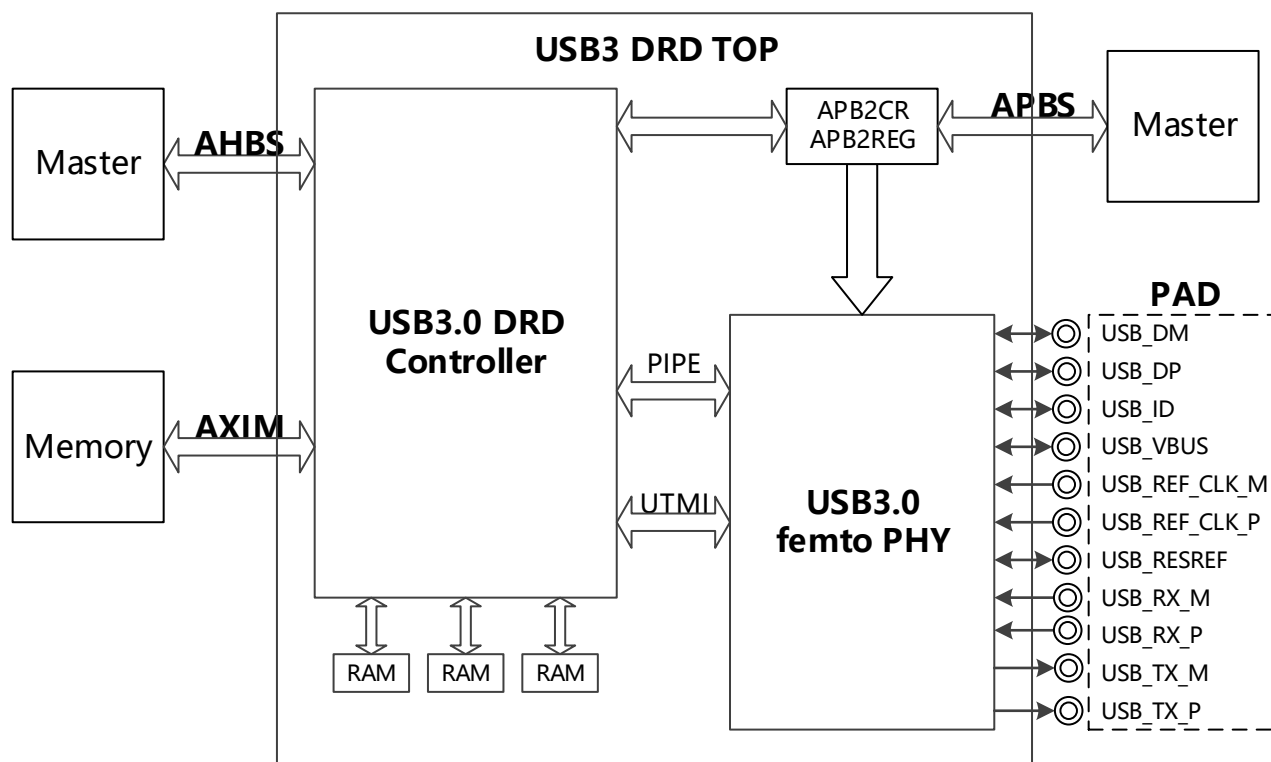
引脚名称	方向	宽度	说明
USB_DRD_DM	IO	1	USB D-信号 该双向引脚用于 USB2.0 数据传输。 在 HS 操作中，该引脚接收/发送的最大标称值为 800mV 或 400mV。在 FS 或 LS 操作中，该引脚接收/发送 3.3V 标称电

引脚名称	方向	宽度	说明
			压。
USB_DRD_DP	IO	1	<p>USB D+信号</p> <p>该双向引脚用于 USB2.0 数据传输。</p> <p>在 HS 操作中，该引脚接收/发送的最大标称值为 800mV 或 400mV。在 FS 或 LS 操作中，该引脚接收/发送 3.3V 标称电压。</p>
USB_DRD_ID	IO	1	<p>USB 3.0 直流点探头的 USB 迷你插座标识符和测试点</p> <p>区分迷 mini-A 和 mini-B 插头。仅当 IDPULLUP0 信号为高时，才会对 ID&lt;#&gt;线路进行采样。对 ID&lt;#&gt;线路采样后，IDDIG0 指示是否连接了 mini-A 或 mini-B 线。</p> <p>如果不使用该信号，当 IDPULLUP0 为高时，内部电阻会将该信号的电压电平上拉到模拟 I/O 电源电平。如果 IDPULLUP0 为低，ID&lt;#&gt;会被下拉到 PHY 地。</p> <p>ID&lt;#&gt;引脚还可用作输入/输出模拟测试信号。该节点上的电压由测试接口模式和 VATESTENB 信号决定。如果不使用 ID&lt;#&gt;，则将其保持悬空，并将 VATESTENB 拉低。</p>
USB_DRD_VBUS	IO	1	<p>USB 5V 电源引脚</p> <p>USB 电源引脚。芯片外部的电荷泵必须为该引脚供电。</p> <p>PHY VBUS&lt;#&gt;引脚不得直接连接到 USB 3.0 链路上的 5V VBUS 电压；相反，此引脚必须由外部电阻（REXT1）隔离，以便 PHY VBUS&lt;#&gt;引脚提供较低的电压。</p>
USB_DRD_RCLKN	I	1	USB 3.0 低摆差分输入时钟对。
USB_DRD_RCLKP	I	1	USB 3.0 低摆差分输入时钟对。
USB_DRD_RES	IO	1	<p>参考电阻连接</p> <p>将一个精密电阻连接到板上的地，规格如下：</p> <ul style="list-style-type: none"> <li>■ 电阻：200 欧姆</li> <li>■ 温度：系数+/-100ppm/摄氏度</li> <li>■ 容差：+/-1%</li> </ul>
USB_DRD_RXM	I	1	<p>USB 3.0 高速差分接收对。</p> <p>接收差分对。</p>
USB_DRD_RXP	I	1	<p>USB 3.0 高速差分接收对。</p> <p>接收差分对。</p>

引脚名称	方向	宽度	说明
USB_DRD_TXM	O	1	USB 3.0 高速差分发送对。 发送差分对。
USB_DRD_TXP	O	1	USB 3.0 高速差分发送对。 发送差分对。

## 2.4 功能描述

USB3 DRD 的功能逻辑图如图表 2-2 所示。USB3 DRD 包括一个 USB3 DRD 控制器、一个 USB3.0 femtoPHY 和三个 SRAM。控制器和 PHY 通过 PIPE/UTMI 接口连接。



图表 2-2 功能逻辑图

USB3 DRD 的外设接口包括用于控制器寄存器配置的高速 AHB 从总线接口、用于数据传输的高速 AXI 主总线接口、用于在初始化或 ATE 测试中控制控制器和 PHY 某些信号的低速 APB 总线接口，以及与 USB 插座连接的 PAD。

## 2.5 使用

### 2.5.1 时钟与复位

这些时钟和复位初始化过程应在 USB3 DRD 控制器初始化之前配置：

1. 将 1'b1 写入时钟寄存器 MISCSYS\_USB\_CLK\_CTRL[0xFF\_EC02\_C104] 的 usb3\_drd\_clk\_en/usb3\_drd\_ctrl\_ref\_clk\_en/usb3\_drd\_phy\_ref\_clk\_en 字段，以启用 AHB/AXI/APB 配置时钟、控制器参考时钟、PHY 参考时钟和挂起时钟。注意，默认情况下启用该寄存器。
2. 将 1'b1 写入系统寄存器 REF\_SSP\_EN[0xFF\_EC03\_F034] 的 REF\_SSP\_EN 字段，以启用 PHY 内部参考时钟。
3. 在使用 USB 之前，请启用 USB 的 IOPMP 模块。USB\_IOPMP 的基址是 0xFF\_FC02\_E000。有关详细信息，请参阅 IOPMP 手册。
4. 完成上述步骤 10us 后，将 1 写入复位寄存器 USB3\_DRD\_SWRST[0xFF\_EC02\_C014] 的 sw\_usb3\_drd\_vccrst\_n/sw\_usb3\_drd\_phyrst\_n/sw\_usb3\_drd\_prst\_n 以释放 USB DRD 复位。
5. 初始化 USB 3.0 控制器寄存器以开始转换。

## 2.5.2 切换主机/设备模式

在 DRD 配置中切换模式的顺序如下：

- 从设备切换到主机：
  1. 使用 GCTL[11] (CoreSoftReset) 复位控制器。
  2. 将 GCTL[13:12] (PrtCapDir) 设置为 2'b01 (主机模式)。
  3. 使用 USBCMD.HCRESET 复位主机。
  4. 按照《编程指南》中“初始化主机寄存器”部分的步骤操作。
- 从主机切换到设备：
  1. 使用 GCTL[11] (CoreSoftReset) 复位控制器。
  2. 将 GCTL[13:12] (PrtCapDir) 设置为 2'b10 (设备模式)。
  3. 通过设置 DCTL[30] (CSftRst) 复位设备。
  4. 按照《编程指南》中“初始化寄存器”部分的步骤操作。

## 2.6 寄存器描述

USB 寄存器包括四个部分：

- 控制器寄存器
- xHCI 寄存器
- PHY 寄存器
- 系统控制状态寄存器

四部分寄存器的基址如下：

- USB 的 AHB 接口（基址：0xFF\_E704\_0000）可以访问控制器寄存器和 xHCI 通用寄存器。控制器寄存器的基址为 0xFF\_E704\_0000。

xHCI 寄存器的基址为 0xFF\_E704\_0020。

- USB 的 APB 接口（基址：0xFF\_EC03\_0000）可以访问 PHY 寄存器和系统控制状态寄存器。

PHY 寄存器的基址为 0xFF\_EC03\_0000，偏移量为寄存器描述偏移量\*2。

系统控制状态寄存器的基址为 0xFF\_EC03\_F000。

系统控制状态寄存器的详细说明如下：

## 2.6.1 寄存器内存映射

寄存器名	偏移量	说明	章节/页码
USB_CLK_GATE_STS	0x0	时钟门控状态寄存器	<a href="#">2.6.2.1/125</a>
USB_LOGIC_ANALYZER_TRACE_STS0	0x4	逻辑分析仪轨迹寄存器 0	<a href="#">2.6.2.2/125</a>
USB_LOGIC_ANALYZER_TRACE_STS1	0x8	逻辑分析仪轨迹寄存器 1	<a href="#">2.6.2.3/125</a>
USB_GPIO	0xc	GPIO 寄存器	<a href="#">2.6.2.4/126</a>
USB_DEBUG_STS0	0x10	调试寄存器 0	<a href="#">2.6.2.5/126</a>
USB_DEBUG_STS1	0x14	调试寄存器 1	<a href="#">2.6.2.6/126</a>
USB_DEBUG_STS2	0x18	调试寄存器 2	<a href="#">2.6.2.7/127</a>
USBCTL_CLK_CTRL0	0x1c	HS 抖动调整寄存器	<a href="#">2.6.2.8/127</a>
USBPHY_CLK_CTRL1	0x20	PHY 时钟控制寄存器	<a href="#">2.6.2.9/127</a>
USBPHY_TEST_CTRL0	0x24	PHY 测试控制寄存器 0	<a href="#">2.6.2.10/128</a>
USBPHY_TEST_CTRL1	0x28	PHY 测试控制寄存器 1	<a href="#">2.6.2.11/131</a>
USBPHY_TEST_CTRL2	0x2c	PHY 测试控制寄存器 2	<a href="#">2.6.2.12/134</a>
USBPHY_TEST_CTRL3	0x30	PHY 测试控制寄存器 3	<a href="#">2.6.2.13/135</a>
REF_SSP_EN	0x34	参考时钟使能寄存器	<a href="#">2.6.2.14/135</a>
USB_HADDR_SEL	0x38	AHB 地址选择寄存器	<a href="#">2.6.2.15/136</a>
USB_SYS	0x3c	USB 系统控制寄存器	<a href="#">2.6.2.16/136</a>
USB_HOST_STATUS	0x40	主机状态寄存器	<a href="#">2.6.2.17/138</a>
USB_HOST_CTRL	0x44	主机控制寄存器	<a href="#">2.6.2.18/139</a>
USBPHY_HOST_CTRL	0x48	USB PHY 主机控制寄存器	<a href="#">2.6.2.19/143</a>



寄存器名	偏移量	说明	章节/页码
USBPHY_HOST_STATUS	0x4c	USB PHY 主机状态寄存器	<a href="#">2.6.2.20/145</a>
USB_TEST_REG0	0x50	USB 测试寄存器 0	<a href="#">2.6.2.21/146</a>
USB_TEST_REG1	0x54	USB 测试寄存器 1	<a href="#">2.6.2.22/146</a>
USB_TEST_REG2	0x58	USB 测试寄存器 2	<a href="#">2.6.2.23/146</a>
USB_TEST_REG3	0x5c	USB 测试寄存器 3	<a href="#">2.6.2.24/146</a>

## 2.6.2 寄存器和字段描述

### 2.6.2.1 USB\_CLK\_GATE\_STS

- 描述：时钟门控状态寄存器
- 偏移量：0x0
- 默认值：0x0

位段	名称	类型	说明
[31:3]	RESERVED_1	-	
[2:0]	CLK_GATE_CTRL	RO	时钟门控状态 复位值：0x0

### 2.6.2.2 USB\_LOGIC\_ANALYZER\_TRACE\_STS0

- 描述：逻辑分析仪轨迹寄存器 0
- 偏移量：0x4
- 默认值：0x0

位段	名称	类型	说明
[31:0]	LOGIC_ANALYZER_TRACE0	RO	逻辑分析仪轨迹 0 复位值：0x0

### 2.6.2.3 USB\_LOGIC\_ANALYZER\_TRACE\_STS1

- 描述：逻辑分析仪轨迹寄存器 1
- 偏移量：0x8
- 默认值：0x0

位段	名称	类型	说明
[31:0]	LOGIC_ANALYZER_TRACE1	RO	逻辑分析仪轨迹 1 复位值: 0x0

## 2.6.2.4 USB\_GPIO

- 描述: GPIO 寄存器
- 偏移量: 0xc
- 默认值: 0x0

位段	名称	类型	说明
[31:16]	GP_OUT	RO	GPIO 输出 复位值: 0x0
[15:0]	GP_IN	RW	GPIO 输入 复位值: 0x0

## 2.6.2.5 USB\_DEBUG\_STS0

- 描述: 调试寄存器 0
- 偏移量: 0x10
- 默认值: 0x40

位段	名称	类型	说明
[31:0]	DEBUG0	RO	USB 调试 0 复位值: 0x40

## 2.6.2.6 USB\_DEBUG\_STS1

- 描述: 调试寄存器 1
- 偏移量: 0x14
- 默认值: 0x3c400

位段	名称	类型	说明
[31:0]	DEBUG1	RO	USB 调试 1 复位值: 0x3C400

### 2.6.2.7 USB\_DEBUG\_STS2

- 描述：调试寄存器 2
- 偏移量：0x18
- 默认值：0x0

位段	名称	类型	说明
[31:3]	RESERVED_1	-	
[2:0]	DEBUG2	RO	USB 调试 2 复位值：0x00

### 2.6.2.8 USBCTL\_CLK\_CTRL0

- 描述：HS 抖动调整寄存器
- 偏移量：0x1c
- 默认值：0x20

位段	名称	类型	说明
[31:6]	RESERVED_1	-	
[5:0]	FLADJ_30MHZ_REG	RW	HS 抖动调整。指示为测量 125 的持续时间而适应 mac3 时钟和 utmi 时钟抖动所需的校正。 复位值：0x20

### 2.6.2.9 USBPHY\_CLK\_CTRL1

- 描述：PHY 时钟控制寄存器
- 偏移量：0x20
- 默认值：0x2a00

位段	名称	类型	说明
[31:29]	RESERVED_3	-	
[28:20]	PHY_SSC_REF_CLK_SEL	RW	扩频参考时钟移位 使非标准振荡器频率产生目标 MPLL 输出速率。输入对应频率合成系数。 ■ ssc_ref_clk_sel[8:6] = modulus - 1 ■ ssc_ref_clk_sel[5:0] = 2 的补码推送量 复位值：0x0

位段	名称	类型	说明
[19]	RESERVED_2	-	
[18:16]	PHY_SSC_RANGE	RW	扩频时钟范围 当 ssc_en 有效且 PHY 正在扩频高速发送时钟时，选择扩频调制的范围。 对相位累加器应用固定偏移量。 复位值：0x0
[15]	REF_CLK_SEL	RW	选择连接到 ref_pad_clk_{p,m} 的参考时钟。 启用时，选择外部 ref_pad_clk_{p,m} 输入作为参考时钟源。禁用 ref_use_pad 时，选择 ref_alt_clk_{p,m} 作为片内参考时钟源。 复位值：0x0
[14]	PHY_REF_CLKDIV2	RW	输入参考时钟分频器控制 如果输入参考时钟频率大于 100MHz，则必须启用该信号。然后将参考时钟频率除以 2，使其保持在 MPLL 要求的范围内。 复位值：0x0
[13:8]	PHY_FSEL	RW	频率选择 选择用于 SS 和 HS 操作的参考时钟频率。 复位值：0x2A
[7]	RESERVED_1	-	
[6:0]	PHY_MPLL_MULTIPLIER	RW	MPLL 倍频控制 将参考时钟倍频至适合预期运行速度的频率。 复位值：0x0

## 2.6.2.10 USBPHY\_TEST\_CTRL0

- 描述：PHY 测试控制寄存器 0
- 偏移量：0x24
- 默认值：0x95182

位段	名称	类型	说明
[31:29]	RESERVED_4	-	

位段	名称	类型	说明
[28:24]	LANE0_TX_TERM_OFFSET	RW	发射机端偏移 保留。设置为 5'b00000。 复位值：0x0
[23:21]	RESERVED_3	-	
[20:16]	LOS_LEVEL	RW	信号丢失检测器灵敏度电平控制 保留。将该总线设置为 0x09。 复位值：0x9
[15]	RESERVED_2	-	
[14:12]	LOS_BIAS	RW	信号丢失检测器阈值电平控制 设置 LOS 检测阈值电平。要在板级进行调整，可将该总线连接到寄存器。 <ul style="list-style-type: none"> <li>111: 135mV</li> <li>110: 120mV</li> <li>101: 105mV</li> <li>100: 90mV</li> <li>011: 75mV</li> <li>010: 60mV</li> <li>001: 45mV</li> <li>000: 无效</li> </ul> 默认值：3'b101 复位值：0x5
[11]	RTUNE_ACK	RO	电阻调谐确认 启用该位时，表示电阻调谐仍在进行中。 复位值：0x0
[10]	RTUNE_REQ	RW	电阻调谐请求 启用该信号触发电阻调谐请求（如果尚未进行）。启用该信号后，rtune_ack 变为高，直到终端阻抗的校准完成。 复位值：0x0
[9:8]	VDATREFTUNE0	RW	数据检测电压调整 调整充电器类型检测期间用于检测数据的阈值电压电平（Vdat_ref）。

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>■ 11: -20%</li> <li>■ 10: -10%</li> <li>■ 01: 0 (默认)</li> <li>■ 00: +10%</li> </ul> 复位值: 0x1
[7:4]	TXVREFTUNE0	RW	HS 直流电压电平调整 调整高速直流电平电压。若要在板级启用调整, 请将该总线连接到寄存器。 <ul style="list-style-type: none"> <li>■ 1111: +8.75%</li> <li>■ 1110: +7.5%</li> <li>■ 1101: +6.25%</li> <li>■ 1100: +5%</li> <li>■ 1011: +3.75%</li> <li>■ 1010: +2.5%</li> <li>■ 1001: +1.25%</li> <li>■ 1000: 0 (默认)</li> <li>■ 0111: -1.25%</li> <li>■ 0110: -2.5%</li> <li>■ 0101: -3.75%</li> <li>■ 0100: -5%</li> <li>■ 0011: -6.25%</li> <li>■ 0010: -7.5%</li> <li>■ 0001: -8.75%</li> <li>■ 0000: -10%</li> </ul> 复位值: 0x8
[3:2]	RESERVED_1	-	
[1:0]	TXRISETUNE0	RW	HS 发射机上升/下降时间调整 调整高速波形的上升/下降时间。要在板级启用调整, 请将该位连接到寄存器。 <ul style="list-style-type: none"> <li>■ 11: -10%</li> <li>■ 10: 0 (默认)</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>01: +15%</li> <li>00: +20%</li> </ul> 复位值: 0x2

## 2.6.2.11 USBPHY\_TEST\_CTRL1

- 描述: PHY 测试控制寄存器 1
- 偏移量: 0x28
- 默认值: 0x10303344

位段	名称	类型	说明
[31:30]	RESERVED_7	-	
[29:28]	TXRESTUNE0	RW	USB 电源阻抗调整 一些应用程序需要在 USB 上添加额外的设备, 例如串联开关, 这可能会增加显著的串联电阻。 该总线调整驱动器源阻抗 (LS/FS 模式下的上拉/下拉阻抗和 HS 模式下的下拉阻抗), 以补偿 USB 上增加的串联电阻。 <ul style="list-style-type: none"> <li>11: -40hm</li> <li>10: -20hm</li> <li>01: 0 (默认)</li> <li>00: +1.50hm</li> </ul> 复位值: 0x1
[27:26]	RESERVED_6	-	
[25:24]	TXPREEMPAMPTUNE0	RW	HS 发送器预加重电流控制 该信号控制在 J 到 K 或 K 到 J 转换后源向 DP<#>和 DM<#>的电流。HS 发送器预加重电流以单位量定义。一个单位量约为 600uA, 定义为 1X 预加重电流。 <ul style="list-style-type: none"> <li>11: 3x 预加重电流</li> <li>10: 2x 预加重电流</li> <li>01: 1x 预加重电流</li> <li>00: 禁用 (默认)</li> </ul> 如果未使用这些信号, 请将其设置为 2'b00。 复位值: 0x0
[23:22]	RESERVED_5	-	

位段	名称	类型	说明
[21:20]	TXHSXVTUNE0	RW	发射器高速交叉调整 该总线调整在 HS 模式下传输时 DP<#> 和 DM<#> 信号交叉的电压。 <ul style="list-style-type: none"> <li>11: 0 (默认)</li> <li>10: +15mV</li> <li>01: -15mV</li> <li>00: 保留</li> </ul> 复位值: 0x3
[19:17]	RESERVED_4	-	
[16]	TXPREEMPULSETUNE0	RW	HS 发射器预加重持续时间控制 该信号控制 HS 预加重电流源输入 DP<#>或 DM<#>的持续时间。HS 发射器预加重持续时间以单位数量定义。 预加重持续时间的一个单位约为 580ps, 定义为 1X 预加重持续时间。仅当 TXPREEMPAMPTUNE<#>[1] 或 TXPREEMPAMPTUNE<#>[0]设置为 1'b1 时, 该信号才有效。 <ul style="list-style-type: none"> <li>1: 短</li> <li>0: 长 (默认)</li> </ul> 如果未使用 TXPREEMPULSETUNE<#>, 请将其设置为 1'b0。 复位值: 0x0
[15:12]	TXFSLSTUNE0	RW	FS/LS 源阻抗调整 调整高电平驱动时的低速和全速单端源阻抗。要在板级启用调整, 请将该总线连接到寄存器。 <ul style="list-style-type: none"> <li>1111: -5%</li> <li>0111: -2.5%</li> <li>0011: 0 (默认)</li> <li>0001: +2.5%</li> <li>0000: +5%</li> </ul> 该参数控件以温度计代码编码。保留不支持的任何非温度计代码设置 (即 1001)。 复位值: 0x3
[11]	RESERVED_3	-	
[10:8]	SQRXTUNE0	RW	静噪阈值调整



位段	名称	类型	说明
			调整用于检测有效高速数据的阈值的电压电平。要在板级启用调整，请将该总线连接到寄存器。 <ul style="list-style-type: none"> <li>■ 111: -20%</li> <li>■ 110: -15%</li> <li>■ 101: -10%</li> <li>■ 100: -5%</li> <li>■ 011: 0 (默认)</li> <li>■ 010: +5%</li> <li>■ 001: +10%</li> <li>■ 000: +15%</li> </ul> 复位值: 0x3
[7]	RESERVED_2	-	
[6:4]	OTGTUNE0	RW	VBUS 有效阈值调整 该总线用于调整 VBUS<#>有效阈值的电压电平。要在板级启用调整，请将该总线连接到寄存器。 <ul style="list-style-type: none"> <li>■ 111: +9%</li> <li>■ 110: +6%</li> <li>■ 101: +3%</li> <li>■ 100: 0 (默认)</li> <li>■ 011: -3%</li> <li>■ 010: -6%</li> <li>■ 001: -9%</li> <li>■ 000: -12%</li> </ul> 默认值: 0x4
[3]	RESERVED_1	-	
[2:0]	COMPDISTUNE0	RW	断开阈值调整 调整用于检测主机上断开事件的阈值的电压电平。要在板级启用调整，请将此总线连接到寄存器。 <ul style="list-style-type: none"> <li>■ 111: +4.5%</li> <li>■ 110: +3%</li> <li>■ 101: +1.5%</li> <li>■ 100: 0 (默认)</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>011: -1.5%</li> <li>010: -3%</li> <li>001: -4.5%</li> <li>000: -6%</li> </ul> 复位值: 0x4

## 2.6.2.12 USBPHY\_TEST\_CTRL2

- 描述: PHY 测试控制寄存器 2
- 偏移量: 0x2c
- 默认值: 0x1c1c0f0

位段	名称	类型	说明
[31:26]	RESERVED_3	-	
[25:20]	PCS_TX_DEEMPH_6DB	RW	6dB 时的 TX 去加重值 当 pipeP_tx_deemph[1:0] 设置为 2'b10 (根据 PIPE3 规范) 时, 该静态值设置 TX 驱动器去加重值。 提供该总线的目的是使其完整, 并作为第二个潜在的发射振幅。 默认值: 28 复位值: 0x1C
[19:18]	RESERVED_2	-	
[17:12]	PCS_TX_DEEMPH_3P5DB	RW	3.5dB 时的 TX 去加重值 当 pipeX_tx_deemph[1:0] 设置为 2'b01 (根据 PIPE3 规范) 时, 该静态值设置 TX 驱动器去加重值。要在板级启用调整以符合 RX eye 规范, 请将信号连接到寄存器。 默认值: 28 复位值: 0x1C
[11:10]	RESERVED_1	-	
[9:0]	PCS_RX_LOS_MASK_VAL	RW	可配置的信号丢失屏蔽宽度 设置在 U3 和 U2 状态下屏蔽传入 LFPS 的参考时钟周期数。 屏蔽传入 LFPS 的参考时钟周期数等于 pcs_rx_los_mask_val[9:0] 的值。此控件过滤掉由不兼容主机发送的短的、不兼容的 LFPS 故障。

位段	名称	类型	说明
			复位值：0xF0

### 2.6.2.13 USBPHY\_TEST\_CTRL3

- 描述：PHY 测试控制寄存器 3
- 偏移量：0x30
- 默认值：0x47f

位段	名称	类型	说明
[31:12]	RESERVED_2	-	
[11]	PHY_LOOPBACKENB0	RW	环回测试使能 该控制器信号将 USB 3.0 femtoPHY 置于 HS 环回模式，同时启用 HS 接收和发送逻辑。 <ul style="list-style-type: none"> <li>■ 1：HS 数据传输过程中，启用 HS 接收逻辑。</li> <li>■ 0：HS 数据传输过程中，禁用 HS 接收逻辑。</li> </ul> 复位值：0x0
[10:8]	TX_VBOOST_LVL	RW	发射升压电平 设置发射升压幅度（mVppd）。要在板级启用调整，请将该总线连接到寄存器。 默认值：3'b100 复位值：0x4
[7]	RESERVED_1	-	
[6:0]	PCS_TX_SWING_FULL	RW	发射振幅（全摆幅模式） 此静态值设置发射机的发射振幅。要在板级启用调整以符合 RX eye 标准，请将该信号连接到寄存器。 默认值：127 (7b'11111111) 复位值：0x7F

### 2.6.2.14 REF\_SSP\_EN

- 描述：参考时钟使能寄存器
- 偏移量：0x34
- 默认值：0x0

位段	名称	类型	说明
[31:1]	RESERVED_1	-	
[0]	REF_SSP_EN	RW	SS 功能的参考时钟启用 启用预分频器的参考时钟。ref_ssp_en 信号必须保持为无效，直到参考时钟以适当的频率运行，此时可以启用 ref_ssp_en。 对于低功耗状态，还可以禁用 ref_ssp_en。 复位值：0x0

### 2.6.2.15 USB\_HADDR\_SEL

- 描述：AHB 地址选择寄存器
- 偏移量：0x38
- 默认值：0x0

位段	名称	类型	说明
[31:4]	RESERVED_1	-	
[3:0]	HADDR_SEL	RW	用于选择[19:16]: 0x0 到 0x3: 正常的 AHB 总线的寄存器访问（仅 0x0 有用） 0x4 到 0x7: 内部 RAM0 - 调试访问（256KB） 0x8 到 0xb: 内部 RAM1 - 调试访问（256KB） 0xc 到 0xf: 内部 RAM2 - 调试访问（256KB） 复位值：0x0

### 2.6.2.16 USB\_SYS

- 描述：USB 系统控制寄存器
- 偏移量：0x3c
- 默认值：0x0

位段	名称	类型	说明
[31:9]	RESERVED_1	-	
[8]	ATERESET	RW	来自自动测试设备的复位输入 USB 3.0 femtoPHY 上电后（未处于挂起或睡眠模式），自动测试设备可使用此测试信号禁用 PHYCLOCK<#> 和 FREECLK，然后以对齐的相位重新启用它们。 ■ 1: 禁用 PHYCLOCK<#>和 FREECLK 输出。

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0: 禁用 ATERESET 后的一段特定时间内, PHYCLOCK&lt;#&gt; 和 FREECLK 输出可用。</li> </ul> 复位值: 0x0
[7:4]	BUS_FILTER_BYPASS	RW	总线过滤器旁路。禁任由 DWC_USB3_EN_BUS_FILTERS coreConsultant 参数启用的内部总线过滤器。仅当 DWC_USB3_EN_BUS_FILTERS 为 1 时, 该静态信号才存在。预计该信号在上电复位时设置或复位, 并且在控制器正常运行期间不会改变。每个位的功能是: <ul style="list-style-type: none"> <li>bus_filter_bypass[3]: 保留</li> <li>bus_filter_bypass[2]: 旁路 utmisrp_bvalid 过滤器</li> <li>bus_filter_bypass[1]: 旁路 pipe3_PowerPresent all U3 端口过滤器</li> <li>bus_filter_bypass[0]: 旁路 utmiotg_vbusvalid all U2 端口过滤器</li> </ul> 在仅主机模式下, 不需要内部总线过滤器。因此, bus_filter_bypass[2: 0]必须连接到逻辑高值 (3'b111)。保留位可以设置为 0 或 1, 但不应悬空。           值: <ul style="list-style-type: none"> <li>1'b0: 启用总线过滤器。</li> <li>1'b1: 禁用总线过滤器。(旁路)</li> </ul> 复位值: 0x0
[3]	VATESTENB	RW	模拟测试引脚选择 允许将模拟测试电压置于 ID<#>引脚上。 <ul style="list-style-type: none"> <li>1: 可以在 ID&lt;#&gt;上查看或应用模拟测试电压。</li> <li>0: 不能在 ID&lt;#&gt;上查看或应用模拟测试电压。</li> </ul> 复位值: 0x0
[2]	TEST_POWERDOWN_SSP	RW	SS 功能电路关断控制 关闭 PHY 中的所有 SS 功能电路以进行 IDDQ 测试。 在启用该信号之前, 请确保将 phy_reset 设置为 1'b1。 复位值: 0x0
[1]	TEST_POWERDOWN_HSP	RW	HS 功能电路关断控制 关闭 PHY 中的所有 HS 功能电路以进行 IDDQ 测试。

位段	名称	类型	说明
			在启用该信号之前，请确保将 VDATSRCEENB<#>、VDATDETENB<#>、DCDENB<#>、BYPASSSEL<#> 和 test_burnin 设置为 1'b0。 复位值：0x0
[0]	COMMONONN	RW	通用块掉电控制 当 USB 3.0 femtoPHY 处于挂起或睡眠模式时，控制 PLL 块中的掉电信号。 <ul style="list-style-type: none"> <li>1：在挂起或睡眠模式下，PLL 块掉电。</li> <li>0：在挂起或睡眠模式下，PLL 块保持通电并继续消耗电流。</li> </ul> 当 PHY 既不处于挂起模式也不处于睡眠状态时，可以动态控制 COMMONONN。 复位值：0x0

## 2.6.2.17 USB\_HOST\_STATUS

- 描述：主机状态寄存器
- 偏移量：0x40
- 默认值：0x0

位段	名称	类型	说明
[31:22]	RESERVED_3	-	
[21]	HOST_SYSTEM_ERR	RO	主机系统错误 该信号表示发生了主机系统错误，如 USBSTS.HSE 字段所示。只有当 USB_CMD.HSEE 字段设置为 1 时，才会启用该信号。当主机控制器在 AHB、AXI 或本机主总线中遇到“错误”响应时，可能会发生这种情况。当软件清除 USBSTS.HSE 字段时，除非主机继续启用其总线错误输出，否则禁用该信号。对 HSE 的典型软件响应是复位控制器。有关更多详细信息，请参阅 xHCI 1.0 规范的第 4.10.2.6 节。 复位值：0x0
[20]	HOST_LEGACY_SMI_INTERRUPT	RO	SMI 中断，高电平有效。 PCIe 接口需要传递该 SMI 中断输出。 复位值：0x0
[19:18]	RESERVED_2	-	

位段	名称	类型	说明
[17:16]	OPERATIONAL_MODE	RO	端口能力方向 该信号定义了控制器的操作模式。它直接反映了 GCTL 寄存器的 bit[13:12]中编程的值。 PRTCAPDIR: 端口能力方向 (PrtCapDir) <ul style="list-style-type: none"> <li>2'b01: 用于主机配置</li> <li>2'b10: 用于设备配置</li> </ul> 复位值: 0x0
[15:14]	RESERVED_1	-	
[13:12]	HUB_VBUS_CTRL	RO	每个下游端口的端口电源控制 <ul style="list-style-type: none"> <li>0: VBUS 关闭。</li> <li>1: VBUS 打开。</li> </ul> 低位用于 USB 2.0 端口, 高位用于 USB 3.0 SS 端口: hub_vbus_ctrl[N-1:0] = { hub_vbus_ctrl['DWC_USB3_NUM_U3_ROOT_PORTS-1:0], hub_vbus_ctrl['DWC_USB3_NUM_U2_ROOT_PORTS-1:0]} 复位值: 0x0
[11:0]	HOST_CURRENT_BELT	RO	当前 BELT 值 该信号指示所有接收到的 BELT 值的最小值以及由“设置 LTV”命令设置的 BELT。该信号仅在主机模式下有效。 复位值: 0x0

## 2.6.2.18 USB\_HOST\_CTRL

- 描述: 主机控制寄存器
- 偏移量: 0x44
- 默认值: 0x1101

位段	名称	类型	说明
[31:26]	RESERVED_4	-	
[25]	PIPE3_POWERPRESENT	RW	未使用, 保留 复位值: 0x0
[24]	UTMIOTG_VBUSVALID	RW	Vbus 有效。指示电压 Vbus 是否对主机和 A 设备操作有效。

位段	名称	类型	说明
			比较器阈值为： <ul style="list-style-type: none"> <li>■ 1'b0: Vbus &lt; 4.4V</li> <li>■ 1'b1: Vbus &gt; 4.75V</li> </ul> 如果选择 DWC_USB3_EN_BUS_FILTERS 为 0, 则此信号必须是滤波信号。它必须与 pipe3_PowerPresent 相同。如果选择 DWC_USB3_HSPHY_INTERFACE 作为 UTMI, 则必须在所有非外设（主机）模式下驱动此输入。 在仅主机模式下, 此信号必须连接到逻辑高值 1'b1。 复位值: 0x0
[23:22]	RESERVED_3	-	
[21:20]	HUB_PORT_OVERCURRENT	RW	这是根集线器端口的每个端口过流指示： <ul style="list-style-type: none"> <li>■ 0: 无过流</li> <li>■ 1: 过流</li> </ul> 低位用于 USB 2.0 端口, 高位用于 USB 3.0 SS 端口： hub_port_overcurrent[N-1:0] = { hub_port_overcurrent['DWC_USB3_NUM_U3_ROOT_PORTS-1:0], hub_port_overcurrent['DWC_USB3_NUM_U2_ROOT_PORTS-1:0]} 所需的最小过流持续时间为 3 个挂起时钟周期。 复位值: 0x0
[19:18]	RESERVED_2	-	
[17:16]	HUB_PORT_PERM_ATTACH	RW	指示连接到下游端口的设备是否永久连接。 <ul style="list-style-type: none"> <li>■ 0: 未永久连接</li> <li>■ 1: 永久连接</li> </ul> 低位用于 USB 2.0 端口, 高位用于 USB 3.0 SS 端口： hub_port_perm_attach[N-1:0] = { hub_port_perm_attach['DWC_USB3_NUM_U3_ROOT_PORTS-1:0], [hub_port_perm_attach['DWC_USB3_NUM_U2_ROOT_PORTS-1:0]} 复位值: 0x0
[15:12]	HOST_NUM_U2_PORT	RW	USB 2.0 端口数



位段	名称	类型	说明
			<p>该信号覆盖 DWC_USB3_NUM_U2_ROOT_PORTS 合成配置参数。</p> <p>例如，这允许您开发带有 4 端口 USB 2.0 主机的芯片，并将其封装为板上的两个 USB 2.0 端口。上面的两个端口未启用。HCSPARAMS1 寄存器的“端口数”字段由该端口控制。“端口数”表示“host_num_u3_ports + host_num_u2_port”值。</p> <p>如果不需要覆盖功能，请按如下方式赋值此端口：</p> <p>赋值 host_num_u2_port[3:0] = 'DWC_USB3_NUM_U2_ROOT_PORTS;</p> <p>复位值：0x1</p>
uz[11:8]	HOST_NUM_U3_PORT	RW	<p>USB 3.0 SS 端口数</p> <p>该信号覆盖 DWC_USB3_NUM_U3_ROOT_PORTS 合成配置参数。</p> <p>例如，这允许您开发带有 4 端口 USB 3.0 SS 主机的芯片，并将其封装为板上的两个 USB 3.0 SS 端口。上面的两个端口未启用。HCSPARAMS1 寄存器的“端口数”字段由该端口控制。“端口数”表示“host_num_u3_ports + host_num_u2_port”值。</p> <p>如果不需要覆盖功能，请按如下方式赋值此端口：</p> <p>赋值 host_num_u3_port[3:0] = 'DWC_USB3_NUM_U3_ROOT_PORTS;</p> <p>复位值：0x1</p>
[7]	RESERVED_1	-	
[6]	HOST_U2_PORT_DISABLE	RW	<p>USB 2.0 端口禁用控制</p> <ul style="list-style-type: none"> <li>0：启用端口。</li> <li>1：禁用端口。</li> </ul> <p>该信号为 1 时，将停止报告端口的连接/断开事件，并使端口处于禁用状态。出于安全考虑，硬件可以禁用端口，而不管 xHCI 驱动程序是否启用端口。</p> <p>HCSPARAMS1 寄存器的“端口数”字段不受该信号影响。</p> <p>该信号应该是静态的（在运行过程中不应该变化），或者在运行过程中只从 0 到 1 变化一次，之后保持为 1。</p> <p>注意：如果任何其他 U2 端口正常运行，不得禁用端口-0（第</p>

位段	名称	类型	说明
			一个 U2 端口)。 复位值: 0x0
[5]	HOST_U3_PORT_DISABLE	RW	USB 3.0 SS 端口禁用控制 <ul style="list-style-type: none"> <li>0: 启用端口。</li> <li>1: 禁用端口。</li> </ul> 该信号为 1 时, 将停止报告端口的连接/断开事件, 并使端口处于禁用状态。出于安全考虑, 硬件可以禁用端口, 而不管 xHCI 驱动程序是否启用端口。 HCSPARAMS1 寄存器的“端口数”字段不受该信号影响。 该信号应该是静态的(在运行过程中不应该变化), 或者在运行过程中只从 0 到 1 变化一次, 之后保持为 1。 注意: 如果任何其他 U2 端口处于工作状态, 不得禁用端口-0 (第一个 U2 端口)。 注意: <ul style="list-style-type: none"> <li>该信号在仅 USB 2.0 模式下不存在。</li> <li>如果其他超高速端口正常运行, 不得禁用超高速端口 0 (第一个 SS 端口)。</li> </ul> 复位值: 0x0
[4]	HOST_PORT_POWER_CONTROL_PRESENT	RW	该端口定义能力参数 (HCCPARAMS) 的 bit[3]。通过引脚端口电源控制 (PPC) 更改 PPC 值。这表示主控制器实现是否包括端口电源控制。 <ul style="list-style-type: none"> <li>0: 表示端口没有端口电源开关。</li> <li>1: 表示端口有端口电源开关。</li> </ul> 复位值: 0x0
[3]	HOST_MSI_ENABLE	RW	这将启用脉冲型中断信号(一个总线时钟周期)“中断”端口, 而不是电平敏感中断。当连接到 PCIe 时, 可以轻松地将“中断”映射到 PCIe 控制器中的 MSI。 只能在主机模式下启用 MSI。设备模式下还不支持 MSI。因此, 如果控制器配置为 DRD 模式, 则只能使用作为电平信号的线中断。 复位值: 0x0
[2]	HOST_LEGACY_SMI_PCI_CMD_REG_WR	RW	PCI 指令寄存器写, 一个时钟脉冲。 PCIe 接口需要在 PCIe 指令寄存器写期间生成一个时钟脉冲。

位段	名称	类型	说明
			如果您不使用 Legacy 支持，请拉低该信号。 复位值：0x0
[1]	HOST_LEGACY_SMI_BAR_WR	RW	PCI 基址寄存器（BAR）写，一个时钟脉冲。 PCIe 接口需要在 PCIe 基址寄存器写期间生成一个时钟脉冲。 如果您不使用 Legacy 支持，请拉低该信号。 复位值：0x0
[0]	XHC_BME	RW	该信号用于禁用 xHC 的总线主控功能。在 PCI 系统中，它来自 PCI 配置寄存器空间中设备控制寄存器的总线主使能（BME）位。 <ul style="list-style-type: none"> <li>1'b0：禁用总线主控功能。主机控制器无法使用总线主接口。</li> <li>1'b1：启用总线主控功能。</li> </ul> 注意： 在主机模式下： <ul style="list-style-type: none"> <li>对于非 PCI 系统，xhc_bme 始终设置为 1'b1，以便控制器主工作。</li> <li>对于 PCI 系统，将 PCI 的 BME 寄存器位连接到 xhc_bme 输入端。</li> </ul> 在设备模式下，xhc_bme 可以是任何值，但建议将该信号设置为 1。 复位值：0x1

## 2.6.2.19 USBPHY\_HOST\_CTRL

- 描述：USB PHY 主机控制寄存器
- 偏移量：0x48
- 默认值：0x18

位段	名称	类型	说明
[31:6]	RESERVED_1	-	
[5]	VBUSVLDEXT0	RW	外部 VBUS 有效选择 选择 VBUSVLDEXT<#>输入或内部会话有效比较器，以指示 USB 线上的 VBUS<#>信号何时有效。 <ul style="list-style-type: none"> <li>1：使用 VBUSVLDEXT&lt;#&gt;输入。</li> </ul>

位段	名称	类型	说明
			<p>■ 0: 使用内部会话有效比较器。</p> <p>该信号是一个捆绑选项, 必须在上电复位之前设置, 并在正常工作期间保持静态。</p> <p>捆绑选项对于 STA 来说并不重要, 引脚的任何其他时序或负载限制在产品交付中包含的.lib 时序模型中指定。</p> <p>复位值: 0x0</p>
[4]	RX0LOSLFPSSEN	RW	<p>RX LOS LFPS 滤波器启用 (0)</p> <p>启用 RX LOS LFPS 滤波器。在 USB 3.0 操作期间, 设置为 1'b1。</p> <p>■ 1: 启用 RX LOS LFPS 滤波器。</p> <p>■ 0: 禁用 RX LOS LFPS 滤波器。</p> <p>仅在复位时更改此引脚设置。</p> <p>复位值: 0x1</p>
[3]	OTGDISABLE0	RW	<p>OTG 块禁用</p> <p>该控制器信号关断 OTG 模块, 从而禁用 VBUS 有效比较器和会话结束比较器。</p> <p>无论 OTGDISABLE0 的状态如何, 会话有效比较器 (其输出用于在设备模式下使能 DP&lt;#&gt;上的上拉电阻) 始终开启。</p> <p>如果应用不使用 OTG 功能, 则可以将此输入设置为高以节省功耗。</p> <p>■ 1: OTG 块断电。</p> <p>■ 0: OTG 块上电。</p> <p>复位值: 0x1</p>
[2]	VBUSVLDEXTSELO	RW	<p>复位值: 0x0</p>
[1]	DRVVBUS0	RW	<p>驱动 VBUS</p> <p>该控制器信号控制 VBUS 有效比较器。该信号通过外部电荷泵驱动 VBUS 上的 5V 电压。</p> <p>当 OTGDISABLE0 设置为 1'b0 并且 DRVVBUS0 有效时, 即使在挂起或睡眠模式下, 带隙电路和 VBUS 有效比较器也会通电。</p> <p>■ 1: 启用 VBUS 有效比较器。</p> <p>■ 0: 禁用 VBUS 有效比较器。</p> <p>复位值: 0x0</p>

位段	名称	类型	说明
[0]	IDPULLUP0	RW	模拟 ID 输入采样启用 该控制器信号控制 ID<#>线路采样。 <ul style="list-style-type: none"> <li>1: 启用 ID&lt;#&gt;引脚采样, IDDIG0 输出有效。</li> <li>0: 禁用 ID&lt;#&gt;引脚采样, IDDIG0 输出无效。</li> </ul> 复位值: 0x0

## 2.6.2.20 USBPHY\_HOST\_STATUS

- 描述: USB PHY 主机状态寄存器
- 偏移量: 0x4c
- 默认值: 0x0

位段	名称	类型	说明
[31:3]	RESERVED_1	-	
[2]	VBUSVALID0	RO	VBUS 有效指示 该控制器信号由 USB 3.0 femtoPHY 的 VBUS 有效比较器输出, 指示 VBUS0 输出是否处于有效电平。 <ul style="list-style-type: none"> <li>1: VBUS&lt;#&gt;输出有效。</li> <li>0: VBUS&lt;#&gt;输出无效。</li> </ul> 复位值: 0x0
[1]	IDDIG0	RO	Mini-A/B 插头指示 此控制器信号指示连接的插头是 mini-A 插头还是 mini-B 插头。 该信号必须在 IDPULLUP0 信号设置为 1'b1 后的 20ms 内有效。 <ul style="list-style-type: none"> <li>1: 连接的插头是 mini-B 插头。</li> <li>0: 连接的插头是 mini-A 插头。</li> </ul> 复位值: 0x0
[0]	OTGSESSVLD0	RO	OTG 设备会话有效指示 (低电压) 该控制器信号从 USB 3.0 femtoPHY 的会话有效比较器输出, 指示 VBUS 上的电压是否低于 OTG 设备会话有效阈值。 <ul style="list-style-type: none"> <li>1: VBUS 上的电压高于 OTG 设备会话有效阈值。</li> <li>0: VBUS 上的电压低于 OTG 设备会话有效阈值。</li> </ul> 复位值: 0x0

### 2.6.2.21 USB\_TEST\_REG0

- 描述：USB 测试寄存器 0
- 偏移量：0x50
- 默认值：0x0

位段	名称	类型	说明
[31:0]	USB_TEST_REG0	RW	USB 测试寄存器 0 复位值：0x0

### 2.6.2.22 USB\_TEST\_REG1

- 描述：USB 测试寄存器 1
- 偏移量：0x54
- 默认值：0x0

位段	名称	类型	说明
[31:0]	USB_TEST_REG1	RW	USB 测试寄存器 1 复位值：0x0

### 2.6.2.23 USB\_TEST\_REG2

- 描述：USB 测试寄存器 2
- 偏移量：0x58
- 默认值：0xffffffff

位段	名称	类型	说明
[31:0]	USB_TEST_REG2	RW	USB 测试寄存器 2 复位值：0xFFFFFFFF

### 2.6.2.24 USB\_TEST\_REG3

- 描述：USB 测试寄存器 3
- 偏移量：0x5c
- 默认值：0xffffffff

位段	名称	类型	说明
[31:0]	USB_TEST_REG3	RW	USB 测试寄存器 3

位段	名称	类型	说明
			复位值：0xFFFFFFFF

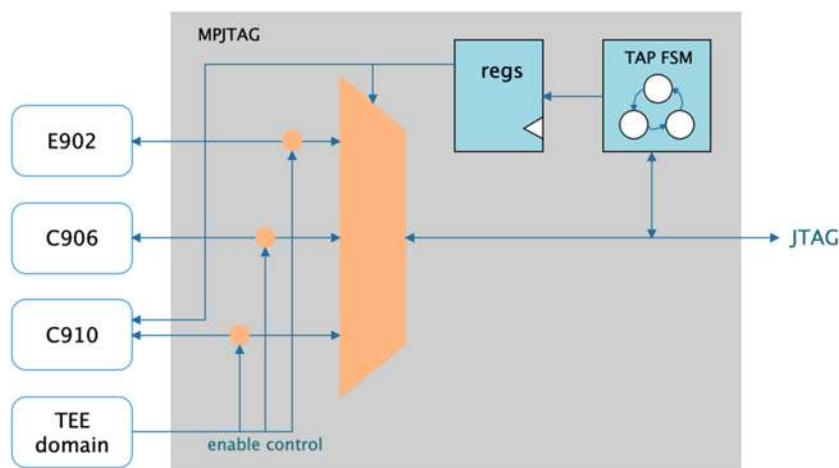
## 3 MPJTAG

### 3.1 概述

Multi-Processor JTAG (MPJTAG) 主要为支持片内多 CPU 核调试而设计，将多个 CPU 核的 JTAG 调试接口通过 MUX 聚合到一个 JTAG 接口，外部使用一个调试器同时调试片内多个 CPU，在一个时刻只能调试一个 CPU 核。

MPJTAG 模块中的 TAP 控制器中有一个标准的 JTAG 状态机，用于接受 JTAG 端口命令，更新内部寄存器；这些寄存器包括 MUX 选择寄存器、一些控制和状态寄存器、系统描述符寄存器，其中系统描述符中描述了当前 SoC 中的 CPU 核心信息，如：CPU 个数、每个 CPU 内的核心数、调试接口状态等信息。

MPJTAG 模块内部结构如图表 3-1 所示：



图表 3-1 MPJTAG 功能框图

### 3.2 主要特性

MPJTAG 用于 CPU 调试，主要有以下特性：

- 支持 E902、C906、C910 同时调试
- 支持 C910 TEE 核和 C910 REE 核分开单独调试或并行调试
- 支持在 CPU 掉电情况下，让调试器获取 SoC 内 CPU 型号参数

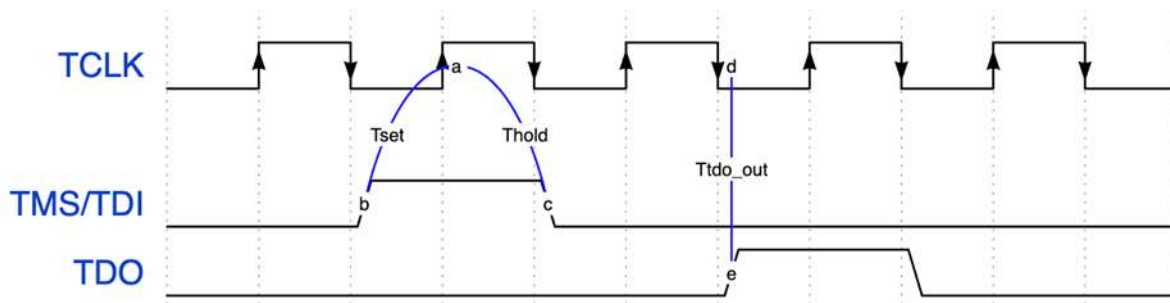


### 3.3 接口

图表 3-2 管脚描述表

管脚名称	方向	宽度	描述
CPU_JTG_TCLK	I	1	JTAG 测试时钟输入
CPU_JTG_TMS	I	1	JTAG 测试模式选择
CPU_JTG_TDI	I	1	JTAG 测试数据输入
CPU_JTG_TDO	O	1	JTAG 测试数据输出
CPU_JTG_TRST	I	1	JTAG 测试复位输入，低电平有效

MPJTAG 接口中，CPU\_JTG\_TCLK 是由外部调试器产生的时钟信号；CPU\_JTG\_TMS 和 CPU\_JTG\_TDI 作为 MPJTAG 的输入接口，在 CPU\_JTG\_TCLK 上升沿采样；CPU\_JTG\_TDO 作为输出接口，在 CPU\_JTG\_TCLK 下降沿设置。MPJTAG 接口时序特性如下：



图表 3-3 MPJTAG 时序图

图表 3-4 MPJTAG 时序特性

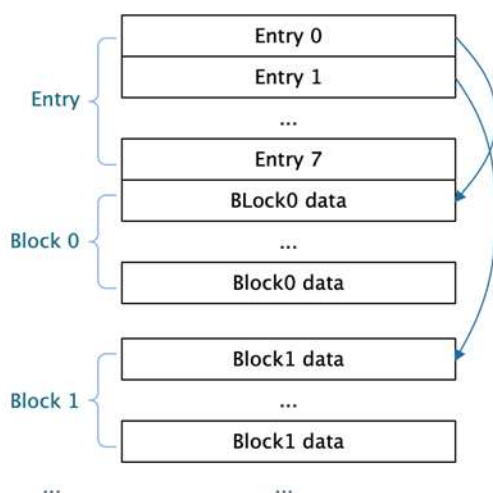
符号	参数	最小值	最大值	单位
fTCLK	TCLK 时钟频率	1	24000	kHz
Tset	TCK 上拉前的 TDI、TMS 设置时间	21	--	ns
Thold	TCK 上拉后的 TDI、TMS 保持时间	21	--	ns
Ttdo_out	从 TCK 下拉到 TDO 输出的延时	1	16	ns

### 3.4 功能描述

MPJTAG 内部主要功能分为 2 部分：SoC 信息记录、CPU 核调试通路选择。

### 3.4.1 SoC 信息

在 MPJTAG 内部使用了一组系统描述符只读寄存器描述芯片内的各个 CPU 核信息，这组寄存器不是直接映射在 MPJTAG 寄存器列表中，而是使用了 SD\_ADDR 和 SD\_DATA 寄存器间接访问。系统描述符内部针对每个 CPU 核都有一组寄存器描述 CPU 的型号、CPUID、CPUJTAG 是否使能等信息；每组寄存器使用二级索引的方式组织的，如图表 3-5 所示。



图表 3-5 系统描述符寄存器映射

Entry 存放区域需要连续不中断；每一个 Entry 为 32bit，其中高 28bit 为指向 block 的地址，bit[0]为 Exists，表示 block 是否存在。Entry 区域以一个全 0 的值作为结束。Entry 格式具体如下所示：

31:4	3:1	0
INDEX	RESERVED	EXISTS

Block 寄存器组由 Header 和 Contents 组成，其中：Header 为 4 字节，存放 Magic、Vendor ID、SD ID 和 Version。

- Magic 为固定值，用于确保该描述模块的可信度。
- Vendor ID 指明使用系统描述模块的供应商。
- SD ID 指明该模块的内容。
- Version 指的是指定 SD ID 的版本信息，即由 SD ID 加上 Version 来指明 contents 中的保存的信息内容和格式。

31:28	27:16	15:4	3:0
Magic (0x9)	Vendor(0x5B7)	SD_ID (0x0)	Version (0x0)

Contents 内容和 SD ID 有关，不同的 ID 对应不同的 Contents。目前只使用到 ID 为 0，表示 CPU 和信息。

31:27	26:23	22:19	18:15	14:10	9	8	7:0
CPI_INDXX	CORE_ID	GROUP	CLASTER	POWER_DOMAIN	TEE	JTAG_EN	RESERVED
MISA[31:0]							
MISA[63:32]							
CPUID 0							
CPUID 1							
CPUID 2							
CPUID 3							
CPUID 4							
CPUID 5							
CPUID 6							

### 3.4.2 CPU 核选择

MPJTAG 管理 SoC 内的 3 个 CPU 核与外部 JTAG 接口直接的连通和断开, 同时 MPJTAG 内部的寄存器也是通过 JTAG 接口更新的。为了支持 3 个 CPU 核和 MPJTAG 内部寄存器与 JTAG 接口的通信, MPJTAG 内部设计有一个 CPUSEL 寄存器; CPUSEL 中每个 bit 对应选通一个 CPU 核, 当 CPUSEL 为 0 时, 表示没有选通 CPU 核, 此时外部调试器可以操作 MPJTAG 内部寄存器, 包括 CPUSEL 寄存器。

31:3	2	1	0
RESERVED	C910_SEL	C906_SEL	E902_SEL

## 3.5 寄存器描述

### 3.5.1 寄存器内存映射

寄存器名	JTAG 地址	CPU 访问地址	位宽	JTAG RW	CPU RW	默认值	说明	章节/页码
HACR	-	-	16	R/W	-	0x8200	调试控制器命令寄存器	<a href="#">3.5.2.1/152</a>
DBGID	5'b00010	-	32	RO	-	0x1000301	调试控制器 ID 寄存器	<a href="#">3.5.2.2/152</a>
CPUSEL	5'b11010	-	32	R/W	-	0x0	多核 CPU 选择寄存器	<a href="#">3.5.2.3/153</a>

寄存器名	JTAG 地址	CPU 访问地址	位宽	JTAG RW	CPU RW	默认值	说明	章节/页码
CPUST	5'b11011	-	32	RO	-	0x0	多核 CPU 状态寄存器	
SD_ADDR	5'b11100	-	32	RW	-	0x0	系统描述符地址选择寄存器	<a href="#">3.5.2.4/153</a>
SD_DATA	5'b11101	-	32	RO	-	0x0	系统描述符数据读出寄存器	<a href="#">3.5.2.5/154</a>
HCR	5'b01101	0xFFFFF019180	32	RW	RW	0x0	MPJTAG 控制寄存器	<a href="#">3.5.2.6/154</a>
EHSR	5'b01111	0xFFFFF019184	32	RO	RO	0x0	MPJTAG 状态寄存器	<a href="#">3.5.2.7/154</a>

## 3.5.2 寄存器和字段描述

### 3.5.2.1 HACR

- 描述: 该寄存器相当于 JTAG 中的 IR 寄存器, 在 TAP 状态机的 Shift\_IR、Update\_IR 状态下被更新。
- 偏移量: 0x0
- 默认值: 0x8200

位段	名称	类型	说明
[7:0]	RESERVED_1	-	
[12:8]	RS	RW	RS[4:0]: 寄存器选择, 指向本次 JTAG 操作的寄存器
[14:13]	RESERVED_2	-	
[15]	R/W	RW	R/W: 表示本次 JTAG 操作是读还是写。 0: 写操作 1: 读操作
[31:16]	RESERVED_3	-	

### 3.5.2.2 DBGID

- 描述: MPJTAG ID 寄存器, 用于外部调试器识别, CPU 不可访问。
- 偏移量: 0x2
- 默认值: 0x10c0322

位段	名称	类型	说明
[3:0]	ID_VER	RO	本 DBGID 寄存器位域定义的版本
[7:4]	ISA_VER	RO	SoC 中的 CPU 指令集类型
[12:8]	CPU_CNT	RO	SoC 中 CPU 核数量
[17:13]	RESERVED_1	-	
[18]	HACR16	RO	MPJTAG 中的 JTAG IR 寄存器, 即 HACR 寄存器的宽度 0x0: HACR 为 8bit 宽度。 0x1: HACR 为 16bit 宽度。
[19]	HAS_SD	RO	默认为 1, 表示 MPJTAG 中存在系统描述符寄存器。
[23:20]	RESERVED_2	-	
[25:24]	MARK	RO	MPJTAG 标志位, 表示本 ID 寄存器是 CPU 中的 ID 寄存器还是 MPJTAG 中的 ID 寄存器。 0x0: CPU 核中的 HAD ID 寄存器 0x1: MPJTAG 中的 DBGID 寄存器
[27:26]	RESERVED_3	-	
[31:28]	CDI_TYPE	RO	调试接口类型, 默认为 0, 表示为 JTAG 接口

### 3.5.2.3 CPUSEL

- 描述: CPU 通路选择, one-hot 编码, 每一 bit 对应一个 CPU 核, CPU 不可访问。
- 偏移量: 0x1A
- 默认值: 0x0

位段	名称	类型	说明
[0]	C910_SEL	RW	C910 CPU JTAG 调试通路选择
[1]	C906_SEL	RW	C906 CPU JTAG 调试通路选择
[2]	E902_SEL	RW	E902 CPU JTAG 调试通路选择
[31:3]	RESERVED_1	-	

### 3.5.2.4 SD\_ADDR

- 描述: 系统描述符寄存器地址选择, CPU 不可访问。
- 偏移量: 0x1C

- 默认值：0x0

位段	名称	类型	说明
[31:0]	SD_ADDR	RW	系统描述符寄存器地址选择，读 SD_DATA 后自动累加

### 3.5.2.5 SD\_DATA

- 描述：系统描述符寄存器数据，CPU 不可访问。
- 偏移量：0x1D
- 默认值：0x0

位段	名称	类型	说明
[31:0]	SD_DATA	RO	系统描述符寄存器数据

### 3.5.2.6 HCR

- 描述：MPJTAG 控制寄存器，CPU 可读写。
- 偏移量：JTAG 为 0x0D，CPU 为 0xFFFF019180
- 默认值：0x0

位段	名称	类型	说明
[17:0]	RESERVED_1	-	
[18]	JW_INT_EN	RW	外部调试器写数据中断使能
[19]	JR_INT_EN	RW	外部调试器读数据中断使能
[31:20]	RESERVED_2	-	

### 3.5.2.7 EHSR

- 描述：MPJTAG 状态寄存器，CPU 可读。
- 偏移量：JTAG 为 0x0F，CPU 为 0xFFFF019184
- 默认值：0x0

位段	名称	类型	说明
[15:0]	RESERVED_1	-	
[16]	J2C_DATA_VLD	RO	JTAG2CPU 寄存器中数据有效。
[17]	C2J_DATA_VLD	RO	CPU2JTAG 寄存器中数据有效。
[18]	JW_INT_ST	RO	外部调试器写数据中断状态有效。

位段	名称	类型	说明
[19]	JR_INT_ST	RO	外部调试器读数据状态状态有效。
[31:20]	RESERVED_2	-	

### 3.5.2.8 系统描述符

地址	条目/块	寄存器名	默认值	说明
0x0	条目	ENTRY0	0x00000081	Block 0 首地址，并且 Block 0 数据有效
0x1		ENTRY1	0x00000131	Block 1 首地址，并且 Block 1 数据有效
0x2		ENTRY2	0x000001E1	Block 2 首地址，并且 Block 2 数据有效
0x3		ENTRY3	0x00000291	Block 3 首地址，并且 Block 3 数据有效
0x4		ENTRY4	0x00000341	Block 4 首地址，并且 Block 4 数据有效
0x5		ENTRY5	0x000003F1	Block 5 首地址，并且 Block 5 数据有效
0x6		ENTRY6	0x00000000	Block 6 首地址，并且 Block 6 数据有效
0x7		RESERVED	0x00000000	-
0x8	系统描述符 块 0	BLOCK_HEADER	0x95B70000	Block header
0x9		CPU_ST	0x00000500	CPU 及 CPU 核心编号及状态信息
0xA		MISA[31:0]	0x40101014	CPU 机器模式处理器指令集特性低 32bit
0xB		MISA[63:32]	0x00000000	CPU 机器模式处理器指令集特性高 32bit
0xC		CPUID0	0x0804000D	CPU ID 0
0xD		CPUID1	0x12080000	CPU ID 1
0xE		CPUID2	0x2420F038	CPU ID 2
0xF		CPUID3	0x00000000	CPU ID 3
0x10		CPUID4	0x00000000	CPU ID 4
0x11		CPUID5	0x00000000	CPU ID 5
0x12		CPUID6	0x00000000	CPU ID 6
0x13	系统描述符 块 1	BLOCK_HEADER	0x95B70000	Block header
0x14		CPU_ST	0x08000900	CPU 及 CPU 核心编号及状态信息
0x15		MISA[31:0]	0x00B4112D	CPU 机器模式处理器指令集特性低 32bit

地址	条目/块	寄存器名	默认值	说明
0x16		MISA[63:32]	0x80000000	CPU 机器模式处理器指令集特性高 32bit
0x17		CPUID0	0x0910090D	CPU ID 0
0x18		CPUID1	0x12006000	CPU ID 1
0x19		CPUID2	0x260C0001	CPU ID 2
0x1A		CPUID3	0x30030066	CPU ID 3
0x1B		CPUID4	0x42180000	CPU ID 4
0x1C		CPUID5	0x50000000	CPU ID 5
0x1D		CPUID6	0x60000853	CPU ID 6
0x1E	系统描述符块 2	BLOCK_HEADER	0x95B70000	Block header
0x1F		CPU_ST	0x10080F00	CPU 及 CPU 核心编号及状态信息
0x20		MISA[31:0]	0x00B4112D	CPU 机器模式处理器指令集特性低 32bit
0x21		MISA[63:32]	0x80000000	CPU 机器模式处理器指令集特性高 32bit
0x22		CPUID0	0x090C090D	CPU ID 0
0x23		CPUID1	0x110C9000	CPU ID 1
0x24		CPUID2	0x260C0001	CPU ID 2
0x25		CPUID3	0x30530077	CPU ID 3
0x26		CPUID4	0x42080407	CPU ID 4
0x27		CPUID5	0x50000003	CPU ID 5
0x28		CPUID6	0x60000A53	CPU ID 6
0x29	系统描述符块 3	BLOCK_HEADER	0x95B70000	Block header
0x2A		CPU_ST	0x10880F00	CPU 及 CPU 核心编号及状态信息
0x2B		MISA[31:0]	0x00B4112D	CPU 机器模式处理器指令集特性低 32bit
0x2C		MISA[63:32]	0x80000000	CPU 机器模式处理器指令集特性高 32bit
0x2D		CPUID0	0x090C090D	CPU ID 0
0x2E		CPUID1	0x110C9000	CPU ID 1
0x2F		CPUID2	0x260C0001	CPU ID 2
0x30		CPUID3	0x30530077	CPU ID 3
0x31		CPUID4	0x42080407	CPU ID 4

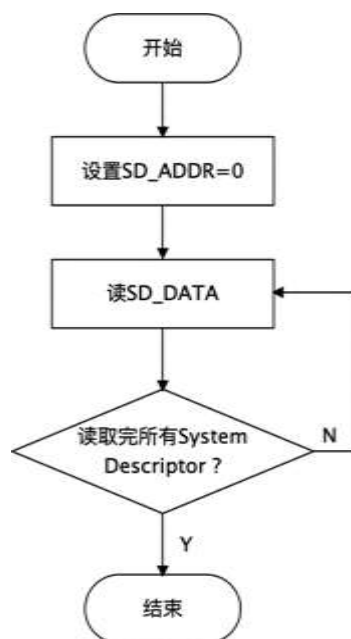


地址	条目/块	寄存器名	默认值	说明
0x32	系统描述符 块 4	CPUID5	0x50000003	CPU ID 5
0x33		CPUID6	0x60000A53	CPU ID 6
0x34		BLOCK_HEADER	0x95B70000	Block header
0x35		CPU_ST	0x11080F00	CPU 及 CPU 核心编号及状态信息
0x36		MISA[31:0]	0x00B4112D	CPU 机器模式处理器指令集特性低 32bit
0x37		MISA[63:32]	0x80000000	CPU 机器模式处理器指令集特性高 32bit
0x38		CPUID0	0x090C090D	CPU ID 0
0x39		CPUID1	0x110C9000	CPU ID 1
0x3A		CPUID2	0x260C0001	CPU ID 2
0x3B		CPUID3	0x30530077	CPU ID 3
0x3C		CPUID4	0x42080407	CPU ID 4
0x3D		CPUID5	0x50000003	CPU ID 5
0x3E		CPUID6	0x60000A53	CPU ID 6
0x3F	系统描述符 块 5	BLOCK_HEADER	0x95B70000	Block header
0x40		CPU_ST	0x11880F00	CPU 及 CPU 核心编号及状态信息
0x41		MISA[31:0]	0x00B4112D	CPU 机器模式处理器指令集特性低 32bit
0x42		MISA[63:32]	0x80000000	CPU 机器模式处理器指令集特性高 32bit
0x43		CPUID0	0x090C090D	CPU ID 0
0x44		CPUID1	0x110C9000	CPU ID 1
0x45		CPUID2	0x260C0001	CPU ID 2
0x46		CPUID3	0x30530077	CPU ID 3
0x47		CPUID4	0x42080407	CPU ID 4
0x48		CPUID5	0x50000003	CPU ID 5
0x49		CPUID6	0x60000A53	CPU ID 6

## 3.6 使用

### 3.6.1 读取系统描述符寄存器

系统描述符寄存器不可以直接读取，需要通过 SD\_ADDR 和 SD\_DATA 寄存器间接读取：首先设置 SD\_ADDR 寄存器，指向需要读取的系统描述符寄存器地址，然后读取 SD\_DATA 寄存器获取对应系统描述符寄存器的值，并且 SD\_ADDR 寄存器自动累加，此时如果再继续读取 SD\_DATA 寄存器将会读取到下一个系统描述符的值。如想遍历所有系统描述符寄存器，按照以下流程读取：



图表 3-6 系统描述符读取流程图

## 4 ADC

### 4.1 概述

ADC 可将外模模拟信号转换成一定比例的数字值，从而实现对模拟信号的测量，可用于电压测量、按键检测等。ADC 模块包括 ADC 控制器和 PHY。控制器支持一个 APB 从总线接口来配置寄存器。ADC 模块是一种功率和面积优化的逐次逼近型 ADC，分辨率可在 12/10/8 和 6 位之间选择。ADC 支持 8 个独立的单端通道。

### 4.2 主要特性

ADC 模块支持的主要特性如下：

- 支持可选的 12/10/8/6 位分辨率
- 支持最大 2.63MSPS 转换速率（12 位，主时钟：73.728MHz）
- 支持 8 个独立的单端输入
- 1.8V 模拟电源，0.8V 数字电源，0~1.8V 输入满量程电压
- 典型 12 位信噪比和失真比（SINAD）62.5dB（待定）
- 支持内部带隙和参考电压缓冲器
- 支持单次模式和连续模式
- 支持异常采样值的中断上报

### 4.3 接口

ADC 的管脚如下图表 4-1 所述：

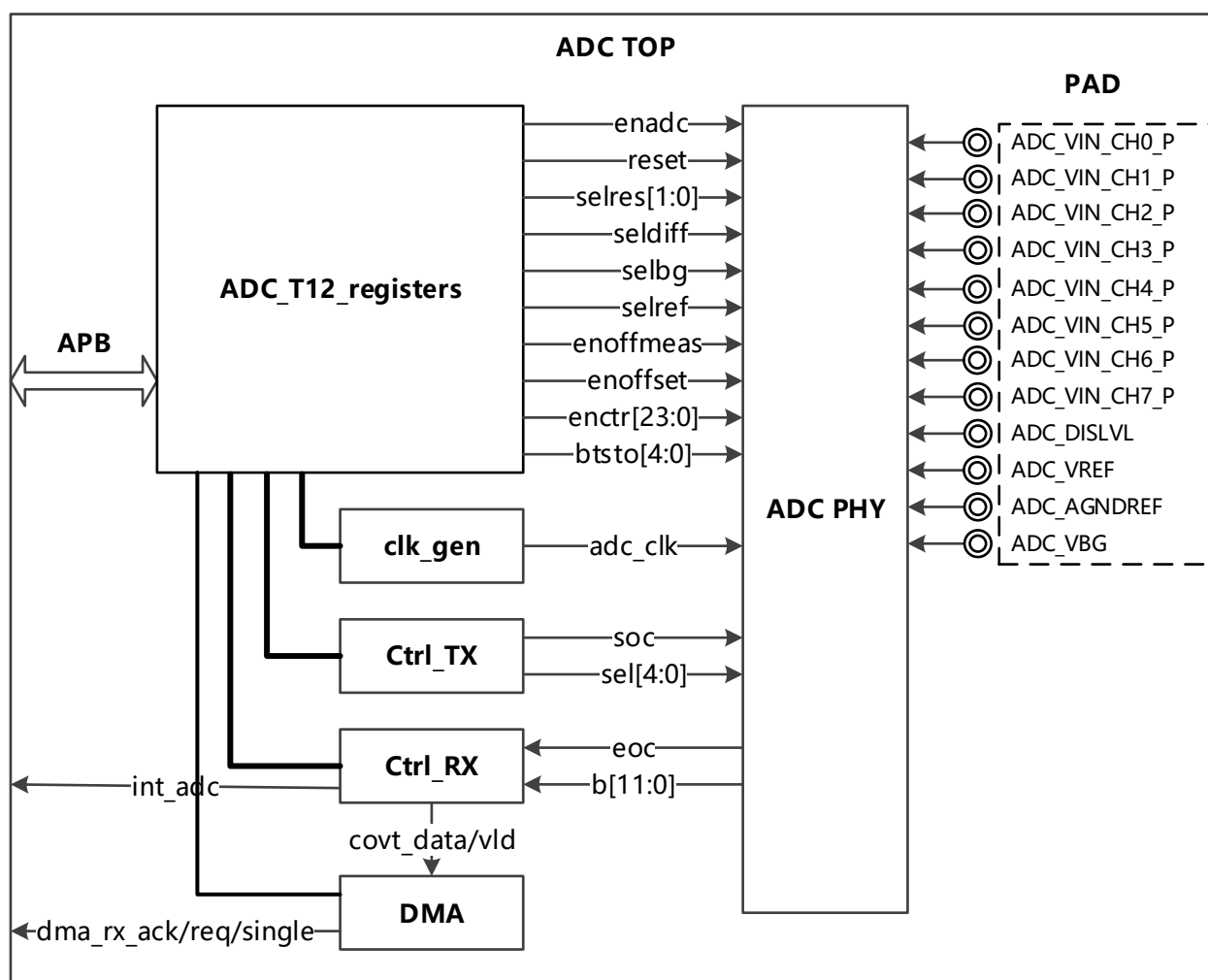
图表 4-1 ADC 管脚描述

管脚名称	方向	宽度	说明
ADC_DISLVL	IO	1	没有数字电源。 当 avddhv 存在但 dvdd 不在指定范围内时（例如在系统启动期间），该信号必须设置为高电平。这会将内部模块之间的接口设置为已知状态，从而防止模拟电源（avddhv）上的电流消耗。 要将模块设置为深度掉电模式，该信号应置为高电平（使用 enadc 和 selbg = 1'b0）。 注：该输入信号必须具有 1.8V 电平（avddhv）。
ADC_VREF	IO	1	当 selref = 1'b0，正参考电压。

管脚名称	方向	宽度	说明
ADC_AGNDREF	IO	1	负参考电压 必须连接到模拟地。
ADC_VIN_CH0_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH0_P 是信号输入。
ADC_VIN_CH1_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH1_P 是信号输入。
ADC_VIN_CH2_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH2_P 是信号输入。
ADC_VIN_CH3_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH3_P 是信号输入。
ADC_VIN_CH4_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH4_P 是信号输入。
ADC_VIN_CH5_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH5_P 是信号输入。
ADC_VIN_CH6_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH6_P 是信号输入。
ADC_VIN_CH7_P	IO	1	在单端输入模式下 (seldiff = 1'b0), ADC_VIN_CH7_P 是信号输入。
ADC_VBG	IO	1	当 selbg = 1'b0 和 selref = 1'b1 时, 内置参考缓冲器的外部参考电压。

## 4.4 功能描述

ADC 的功能逻辑图如图表 4-2 所示。ADC TOP 包括一个 ADC\_T12\_register、一个 CLK\_GEN、一个 CTRL\_TX、一个 CTRL\_RX 和一个 ADC PHY。



图表 4-2 功能逻辑图

## 4.5 使用

### 4.5.1 时钟和复位

这些时钟和复位初始化过程应在 ADC 控制器初始化之前配置：

1. 配置 ADC 时钟寄存器，打开 `apb_pclk`。
2. 配置 ADC 复位寄存器，释放 `apb_presetn`。
3. 初始化 ADC 寄存器，开始转换。

### 4.5.2 初始化和操作

软件通过控制以下寄存器来初始化 ADC：

- `ADC_PHY_CFG/ADC_PHY_TEST`：用于控制 ADC macro 的工作模式。

注：曳影 1520 项目的 ADC 只支持单端输入。

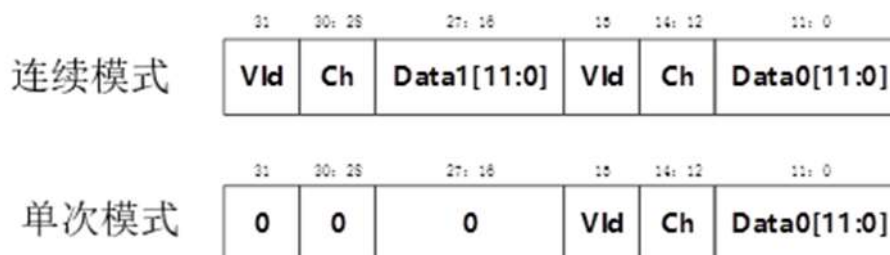
adc\_phy\_enctr 需要更改为 0x8E0 用于 ADC 的 SPEC。

- ADC\_OP\_CTRL/ADC\_OP\_SINGLE\_START/ADC\_FCLK\_CTRL/ADC\_START\_TIME/ADC\_SAMPLE\_TIME：用于控制操作模式、启用通道，启用 DMA 和 FCLK 的频率和相位以及采样率。

注：请参照寄存器描述。

- ADC\_SAMPLE\_DATA：用于访问来自 ADC 的采样数据。

注：采样数据遵循以下数据格式：



图表 4-3 ADC 采样数据格式图

- ADC\_INT\*\_MASK/ADC\_INT\*\_STATUS/ADC\_INT\*\_VALUE\*/ADC\_DFX\_SOC\_CNT\*：用于控制产生中断和 DFX。

注：ADC 支持两种类型的中断：

- 实际中断：当同一通道的 adc\_int\_actual\_mask\_vect[i] 为 unmask (1'b0)，采样数据的值小于 adc\_int\_actual\_lvalue 或大于 adc\_int\_actual\_hvalue，该通道会产生一个中断 (adc\_int\_actual\_st)。
- delta 中断：当同一通道的 adc\_int\_delta\_mask\_vect[i] 为 unmask (1'b0)，采样数据的最后一次测量值与本次测量值之差大于 adc\_int\_delta\_value，该通道会产生一个中断 (adc\_int\_delta\_st)。

### 4.5.3 偏移测量

ADC 没有偏移补偿机制。但是，它允许测量其偏移量。偏移量可以在单端模式的底部刻度处测量。在偏移量测量期间，转换以与常规操作相同的方式发生，仅影响采样。支持连续和单次转换模式。

- 要进入 ADC 的偏移量测量模式，需要设置 ADC\_PHY\_CTRL.ADC\_PHY\_ENOFFMEAS 为 1'b1。
- 设置 ADC\_OP\_CTRL.ADC\_CH\_EN\_VECT 为 8'h1。
- 设置 ADC\_PHY\_CTRL.ADC\_PHY\_ENOFFSET 为 1'b1 (对于 12 位分辨率，预期添加的偏移量为 256 LSB)。
- 偏移值从 ADC\_SMP\_DATA 的 ADC 转换数据读取。

注

理想的偏移代码 (ADC\_PHY\_ENOFFSET) 是底部刻度 (12'h256)。如果偏移值小于 12'h256，偏移量可以为负。在这种情况下，ADC\_PHY\_CTRL.ADC\_PHY\_ENOFFSET 在正常操作期间必须保持 1'b1。由于噪声问题，偏移测量需要几个样本的平均值。建议至少使用 8 个。

## 4.6 寄存器描述

### 4.6.1 寄存器内存映射

寄存器	偏移量	说明	章节/页码
ADC_PHY_CFG	0x0	selbg 和 selref 控制信号在启动序列后必须保持不变。必须为这些控制信号的任何变化制定新的启动序列。	<a href="#">4.6.2.1/165</a>
ADC_PHY_CTRL	0x4	用于控制 ADC 的启用和复位。	<a href="#">4.6.2.2/166</a>
ADC_PHY_TEST	0x8	用于控制 ADC 的测试，仅供 Synopsys 内部使用。	<a href="#">4.6.2.3/167</a>
ADC_OP_CTRL	0xC	用于控制 ADC 的操作模式、启用通道、启用 DMA。	<a href="#">4.6.2.4/167</a>
ADC_OP_SINGLE_START	0x10	单次模式下，用于启动转换。	<a href="#">4.6.2.5/168</a>
ADC_FCLK_CTRL	0x14	用于控制 FCLK 的频率和相位。	<a href="#">4.6.2.6/169</a>
ADC_START_TIME	0x18	用于控制 ADC 的初始化时间，用于 ADC PHY 的 SPEC。	<a href="#">4.6.2.7/169</a>
ADC_SAMPLE_TIME	0x1C	用于设置 ADC 采样率。	<a href="#">4.6.2.8/170</a>
ADC_SAMPLE_DATA	0x20	用于获取 ADC 采样数据。	<a href="#">4.6.2.9/170</a>
ADC_INT_ACTUAL_MASK	0x50	用于设置实际中断屏蔽。 1'b1：屏蔽实际中断。 1'b0：不屏蔽实际中断。 实际中断：当同一个通道的 adc_int_actual_mask_vect[i] 为 unmask ( 1'b0 ) 时，采样数据的值小于 adc_int_actual_lvalue 或大于 adc_int_actual_hvalue，该通道会产生一个中断 ( adc_int_actual_st )。	<a href="#">4.6.2.10/171</a>
ADC_INT_DELTA_MASK	0x54	用于设置 delta 中断屏蔽。 1'b1：屏蔽 delta 中断。 1'b0：不屏蔽 delta 中断。 delta 中断：当同一通道的	<a href="#">4.6.2.11/171</a>

寄存器	偏移量	说明	章节/页码
		adc_int_delta_mask_vect[i] 为 unmask (1'b0), 采样数据的最后一次测量值与本次测量值之差大于 adc_int_delta_value, 该通道会产生一个中断 (adc_int_delta_st)。	
ADC_INT_ACTUAL_STATUS	0x58	实际中断状态 1'b1: 实际中断有效。 1'b0: 实际中断无效。	<a href="#">4.6.2.12/172</a>
ADC_INT_DELTA_STATUS	0x5C	delta 中断状态 1'b1: delta 中断有效。 1'b0: delta 中断无效。	<a href="#">4.6.2.13/173</a>
ADC_INT_ACTUAL_VALUE_CH0	0x60	通道 0 的实际中断的高低值	<a href="#">4.6.2.14/173</a>
ADC_INT_ACTUAL_VALUE_CH1	0x64	通道 1 的实际中断的高低值	<a href="#">4.6.2.15/174</a>
ADC_INT_ACTUAL_VALUE_CH2	0x68	通道 2 的实际中断的高低值	<a href="#">4.6.2.16/174</a>
ADC_INT_ACTUAL_VALUE_CH3	0x6C	通道 3 的实际中断的高低值	<a href="#">4.6.2.17/175</a>
ADC_INT_ACTUAL_VALUE_CH4	0x70	通道 4 的实际中断的高低值	<a href="#">4.6.2.18/175</a>
ADC_INT_ACTUAL_VALUE_CH5	0x74	通道 5 的实际中断的高低值	<a href="#">4.6.2.19/176</a>
ADC_INT_ACTUAL_VALUE_CH6	0x78	通道 6 的实际中断的高低值	<a href="#">4.6.2.20/176</a>
ADC_INT_ACTUAL_VALUE_CH7	0x7C	通道 7 的实际中断的高低值	<a href="#">4.6.2.21/176</a>
ADC_INT_DELTA_VALUE_CH0	0x90	通道 0 的 delta 中断的高低值	<a href="#">4.6.2.22/177</a>
ADC_INT_DELTA_VALUE_CH1	0x94	通道 1 的 delta 中断的高低值	<a href="#">4.6.2.23/177</a>
ADC_INT_DELTA_VALUE_CH2	0x98	通道 2 的 delta 中断的高低值	<a href="#">4.6.2.24/178</a>
ADC_INT_DELTA_VALUE_CH3	0x9C	通道 3 的 delta 中断的高低值	<a href="#">4.6.2.25/178</a>
ADC_INT_DELTA_VALUE_CH4	0xA0	通道 4 的 delta 中断的高低值	<a href="#">4.6.2.26/178</a>
ADC_INT_DELTA_VALUE_CH5	0xA4	通道 5 的 delta 中断的高低值	<a href="#">4.6.2.27/179</a>
ADC_INT_DELTA_VALUE_CH6	0xA8	通道 6 的 delta 中断的高低值	<a href="#">4.6.2.28/179</a>
ADC_INT_DELTA_VALUE_CH7	0xAC	通道 7 的 delta 中断的高低值	<a href="#">4.6.2.29/180</a>
ADC_DFX_SOC_CNT_CH0	0xC0	通道 0 的 SOC 计数器值	<a href="#">4.6.2.30/180</a>



寄存器	偏移量	说明	章节/页码
ADC_DFX_EOC_CNT_CH0	0xC4	通道 0 的 EOC 计数器值	<a href="#">4.6.2.31/180</a>
ADC_DFX_SOC_CNT_CH1	0xC8	通道 1 的 SOC 计数器值	<a href="#">4.6.2.32/181</a>
ADC_DFX_EOC_CNT_CH1	0xCC	通道 1 的 EOC 计数器值	<a href="#">4.6.2.33/181</a>
ADC_DFX_SOC_CNT_CH2	0xD0	通道 2 的 SOC 计数器值	<a href="#">4.6.2.34/181</a>
ADC_DFX_EOC_CNT_CH2	0xD4	通道 2 的 EOC 计数器值	<a href="#">4.6.2.35/182</a>
ADC_DFX_SOC_CNT_CH3	0xD8	通道 3 的 SOC 计数器值	<a href="#">4.6.2.36/182</a>
ADC_DFX_EOC_CNT_CH3	0xDC	通道 3 的 EOC 计数器值	<a href="#">4.6.2.37/182</a>
ADC_DFX_SOC_CNT_CH4	0xE0	通道 4 的 SOC 计数器值	<a href="#">4.6.2.38/182</a>
ADC_DFX_EOC_CNT_CH4	0xE4	通道 4 的 EOC 计数器值	<a href="#">4.6.2.39/183</a>
ADC_DFX_SOC_CNT_CH5	0xE8	通道 5 的 SOC 计数器值	<a href="#">4.6.2.40/183</a>
ADC_DFX_EOC_CNT_CH5	0xEC	通道 5 的 EOC 计数器值	<a href="#">4.6.2.41/183</a>
ADC_DFX_SOC_CNT_CH6	0xF0	通道 6 的 SOC 计数器值	<a href="#">4.6.2.42/184</a>
ADC_DFX_EOC_CNT_CH6	0xF4	通道 6 的 EOC 计数器值	<a href="#">4.6.2.43/184</a>
ADC_DFX_SOC_CNT_CH7	0xF8	通道 7 的 SOC 计数器值	<a href="#">4.6.2.44/184</a>
ADC_DFX_EOC_CNT_CH7	0xFC	通道 7 的 EOC 计数器值	<a href="#">4.6.2.45/185</a>

## 4.6.2 寄存器和字段描述

### 4.6.2.1 ADC\_PHY\_CFG

- 描述：selbg 和 selref 控制信号在启动序列后必须保持不变。必须为这些控制信号的任何变化制定新的启动序列。
- 偏移量：0x0
- 默认值：0x1102

位段	名称	类型	说明
[31:13]	RESERVED_4	-	
[12]	ADC_PHY_SELREF	RW	选择转换器的满量程输入范围 1'b1：使用内部参考发生器。（默认） 1'b0：使用施加到 vref 的电压。

位段	名称	类型	说明
			复位值: 0x1
[11:9]	RESERVED_3	-	
[8]	ADC_PHY_SELBG	RW	启用内部带隙参考电压发生器 1'b1: 内部带隙有效。ADC_VBG 引脚处于高阻抗。(默认) 1'b0: 内部带隙处于掉电状态。如果要使用内置参考缓冲器 (ADC_PHY_SELREF = 1'b1), 应将外部 1.2V 电压连接到 vbg 引脚。 复位值: 0x1
[7:5]	RESERVED_2	-	
[4]	ADC_PHY_SELDIFF	RW	选择 ADC 输入模式。 1'b0: 单端输入 (默认) 1'b1: 差分输入 复位值: 0x0
[3:2]	RESERVED_1	-	
[1:0]	ADC_PHY_SELRES	RW	选择 ADC 分辨率。 2'h0: 6 位模式 2'h1: 8 位模式 2'h2: 10 位模式 (默认) 2'h3: 12 位模式 复位值: 0x2

#### 4.6.2.2 ADC\_PHY\_CTRL

- 描述: 用于控制 ADC 的启用和复位。
- 偏移量: 0x4
- 默认值: 0x0

位段	名称	类型	说明
[31:13]	RESERVED_4	-	
[12]	ADC_PHY_ENOFFSET	RW	添加模拟正偏移量, 默认值 1'b0。 1'b1: 在模拟输入信号上添加正偏移。

位段	名称	类型	说明
			1'b0: 不在模拟输入信号上添加正偏移。 复位值: 0x0
[11:9]	RESERVED_3	-	
[8]	ADC_PHY_ENOFFMEAS	RW	启用偏移测量。 1'b1: 采样零输入 1'b0: 采样选择输入 (默认) 复位值: 0x0
[7:5]	RESERVED_2	-	
[4]	ADC_PHY_RESET	RW	终止转换。激活该信号会将 ADC 设置为采样阶段。 复位值: 0x0
[3:1]	RESERVED_1	-	
[0]	ADC_PHY_ENADC	RW	1'b0: ADC 处于掉电/待机模式。 1'b1: ADC 正常工作。 复位值: 0x0

#### 4.6.2.3 ADC\_PHY\_TEST

- 描述: 用于控制 ADC 的测试, 仅供 Synopsys 内部使用。
- 偏移量: 0x8
- 默认值: 0x0

位段	名称	类型	说明
[31:29]	RESERVED_1	-	
[28:24]	ADC_PHY_BTSTO	RO	测试接口输入, 仅供 Synopsys 内部使用。 在正常操作中, 这些信号应保持低电平。 复位值: 0x0
[23:0]	ADC_PHY_ENCTR	RW	测试接口输出, 仅供 Synopsys 内部使用。 复位值: 0x0

#### 4.6.2.4 ADC\_OP\_CTRL

- 描述: 用于控制 ADC 的操作模式、启用通道、启用 DMA。
- 偏移量: 0xC

- 默认值：0x10

位段	名称	类型	说明
[31:20]	RESERVED_3	-	
[19:12]	ADC_CH_EN_VECT	RW	bit19: 通道 7 启用 bit18: 通道 6 启用 bit17: 通道 5 启用 bit16: 通道 4 启用 bit15: 通道 3 启用 bit14: 通道 2 启用 bit13: 通道 1 启用 bit12: 通道 0 启用 复位值：0x0
[11:5]	RESERVED_2	-	
[4]	ADC_DMA_ENABLE	RW	adc_dma_enable, 用于启用 ADC DMA ctrl。 复位值：0x1
[3:1]	RESERVED_1	-	
[0]	ADC_OP_MODE	RW	1: 单次模式 0: 连续模式 复位值：0x0

#### 4.6.2.5 ADC\_OP\_SINGLE\_START

- 描述：单次模式下，用于启动转换。
- 偏移量：0x10
- 默认值：0x0

位段	名称	类型	说明
[31:1]	RESERVED_1	-	
[0]	ADC_OP_SINGLE_START	W1S	单次模式下，用于启动由 adc_ch_en_vect 启用的每个通道的转换。 复位值：0x0

### 4.6.2.6 ADC\_FCLK\_CTRL

- 描述：用于控制 FCLK 的频率和相位。
- 偏移量：0x14
- 默认值：0x10004

位段	名称	类型	说明
[31:25]	RESERVED_3	-	
[24]	ADC_FCLK_INIT_VALUE	RW	FCLK 控制寄存器的初始值，用于调试。 复位值：0x0
[23:22]	RESERVED_2	-	
[21:16]	ADC_FCLK_PHASE_SEL	RW	FCLK 控制寄存器的相位选择，用于调试。 复位值：0x1
[15:7]	RESERVED_1	-	
[6:0]	ADC_FCLK_DIV	RW	Fclk = (0.7~70MHz)，分频为 0.7MHz，使用核时钟 73.728MHz = 105 = 0x69 典型 FCLK = coreclk/4 = 73.728/4 = 18.432MHz 不得使用（0、1、3、5、7、9），这是时钟占空比 （45%~55%）的无效配置。 复位值：0x4

### 4.6.2.7 ADC\_START\_TIME

- 描述：用于控制 ADC PHY 参数规格的 ADC 初始化时间。
- 偏移量：0x18
- 默认值：0x160

位段	名称	类型	说明
[31:9]	RESERVED_1	-	
[8:0]	ADC_START_TIME	RW	Fclk = (0.7~70MHz), cnt 5us@70MHz = 350 = 0x15E 1. 当 ADC 从掉电模式进入使用内存参考的正常操作时，需要等待定义的上电时间（tpup = 5us），然后开始转换周期。 2. 当 ADC 从待机模式进入使用内部参考的正常操作时，需要等待定义的唤醒时间（twup = 1.5us），然后开始转换周期。

位段	名称	类型	说明
			3. 当 ADC 从掉电模式进入使用外部参考的正常操作时，需要等待定义的 ADC 启用时间（ $t_{enadc} = 5$ 个时钟周期），然后开始转换周期。 复位值：0x160

#### 4.6.2.8 ADC\_SAMPLE\_TIME

- 描述：用于设置 ADC 采样率。
- 偏移量：0x1C
- 默认值：0x10

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ADC_SAMPLE_TIME	RW	用于设置 ADC 采样率。 $ADC \text{ 采样率} = F_{clk} / (1 + adc\_sample\_time)$ 复位值：0x10

#### 4.6.2.9 ADC\_SAMPLE\_DATA

- 描述：用于获取 ADC 采样数据。
- 偏移量：0x20
- 默认值：0x0

位段	名称	类型	说明
[31:0]	ADC_SMP_DATA	RO	单次模式下： bit[31:16]：保留 bit[15]：数据有效 bit[14:12]：通道号 bit[11:0]：ADC 转换数据 连续模式下： bit[31]：数据有效 bit[30:28]：通道号 bit[27:16]：ADC 转换数据 bit[15]：数据有效 bit[14:12]：通道号

位段	名称	类型	说明
			bit[11:0]: ADC 转换数据 复位值: 0x0

#### 4.6.2.10 ADC\_INT\_ACTUAL\_MASK

- 描述：用于设置实际中断屏蔽。1'b1：屏蔽实际中断。1'b0：不屏蔽实际中断。实际中断：当同一个通道的 `adc_int_actual_mask_vect[i]` 为 `unmask (1'b0)` 时，采样数据的值小于 `adc_int_actual_lvalue` 或大于 `adc_int_actual_hvalue`，该通道会产生一个中断 (`adc_int_actual_st`)。
- 偏移量：0x50
- 默认值：0x0

位段	名称	类型	说明
[31:8]	RESERVED_1	-	
[7:0]	ADC_INT_ACTUAL_MASK_VECT	RW	bit7: 通道 7 实际中断屏蔽 bit6: 通道 6 实际中断屏蔽 bit5: 通道 5 实际中断屏蔽 bit4: 通道 4 实际中断屏蔽 bit3: 通道 3 实际中断屏蔽 bit2: 通道 2 实际中断屏蔽 bit1: 通道 1 实际中断屏蔽 bit0: 通道 0 实际中断屏蔽 复位值: 0x0

#### 4.6.2.11 ADC\_INT\_DELTA\_MASK

- 描述：用于设置 delta 中断屏蔽。1'b1：屏蔽 delta 中断。1'b0：不屏蔽 delta 中断。Delta 中断：当同一通道的 `adc_int_delta_mask_vect[i]` 为 `unmask (1'b0)`，采样数据的最后一次测量值与本次测量值之差大于 `adc_int_delta_value`，该通道会产生一个中断 (`adc_int_delta_st`)。
- 偏移量：0x54
- 默认值：0x0

位段	名称	类型	说明
[31:8]	RESERVED_1	-	
[7:0]	ADC_INT_DELTA_MASK_VECT	RW	bit7: 通道 7 delta 中断屏蔽

位段	名称	类型	说明
			bit6: 通道 6 delta 中断屏蔽 bit5: 通道 5 delta 中断屏蔽 bit4: 通道 4 delta 中断屏蔽 bit3: 通道 3 delta 中断屏蔽 bit2: 通道 2 delta 中断屏蔽 bit1: 通道 1 delta 中断屏蔽 bit0: 通道 0 delta 中断屏蔽 复位值: 0x0

#### 4.6.2.12 ADC\_INT\_ACTUAL\_STATUS

- 描述：实际中断状态。1'b1：实际中断有效。1'b0：实际中断无效。
- 偏移量：0x58
- 默认值：0x0

位段	名称	类型	说明
[31:8]	RESERVED_1	-	
[7]	ADC_INT_ACTUAL_ST_CH7	RC	通道 7 实际中断状态 复位值: 0x0
[6]	ADC_INT_ACTUAL_ST_CH6	RC	通道 6 实际中断状态 复位值: 0x0
[5]	ADC_INT_ACTUAL_ST_CH5	RC	通道 5 实际中断状态 复位值: 0x0
[4]	ADC_INT_ACTUAL_ST_CH4	RC	通道 4 实际中断状态 复位值: 0x0
[3]	ADC_INT_ACTUAL_ST_CH3	RC	通道 3 实际中断状态 复位值: 0x0
[2]	ADC_INT_ACTUAL_ST_CH2	RC	通道 2 实际中断状态 复位值: 0x0
[1]	ADC_INT_ACTUAL_ST_CH1	RC	通道 1 实际中断状态 复位值: 0x0



位段	名称	类型	说明
[0]	ADC_INT_ACTUAL_ST_CH0	RC	通道 0 实际中断状态 复位值：0x0

#### 4.6.2.13 ADC\_INT\_DELTA\_STATUS

- 描述：delta 中断状态。1'b1：delta 中断有效。1'b0：delta 中断无效。
- 偏移量：0x5C
- 默认值：0x0

位段	名称	类型	说明
[31:8]	RESERVED_1	-	
[7]	ADC_INT_DELTA_ST_CH7	RC	通道 7 delta 中断状态 复位值：0x0
[6]	ADC_INT_DELTA_ST_CH6	RC	通道 6 delta 中断状态 复位值：0x0
[5]	ADC_INT_DELTA_ST_CH5	RC	通道 5 delta 中断状态 复位值：0x0
[4]	ADC_INT_DELTA_ST_CH4	RC	通道 4 delta 中断状态 复位值：0x0
[3]	ADC_INT_DELTA_ST_CH3	RC	通道 3 delta 中断状态 复位值：0x0
[2]	ADC_INT_DELTA_ST_CH2	RC	通道 2 delta 中断状态 复位值：0x0
[1]	ADC_INT_DELTA_ST_CH1	RC	通道 1 delta 中断状态 复位值：0x0
[0]	ADC_INT_DELTA_ST_CH0	RC	通道 0 delta 中断状态 复位值：0x0

#### 4.6.2.14 ADC\_INT\_ACTUAL\_VALUE\_CH0

- 描述：通道 0 的实际中断的高低值
- 偏移量：0x60

- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH0	RW	通道 0 的实际中断的高值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH0	RW	通道 0 的实际中断的低值 复位值：0x0

#### 4.6.2.15 ADC\_INT\_ACTUAL\_VALUE\_CH1

- 描述：通道 1 的实际中断的高低值
- 偏移量：0x64
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH1	RW	通道 1 的实际中断的高值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH1	RW	通道 1 的实际中断的低值 复位值：0x0

#### 4.6.2.16 ADC\_INT\_ACTUAL\_VALUE\_CH2

- 描述：通道 2 的实际中断的高低值
- 偏移量：0x68
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH2	RW	通道 2 的实际中断的高值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH2	RW	通道 2 的实际中断的低值 复位值: 0x0

#### 4.6.2.17 ADC\_INT\_ACTUAL\_VALUE\_CH3

- 描述: 通道 3 的实际中断的高低值
- 偏移量: 0x6C
- 默认值: 0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH3	RW	通道 3 的实际中断的高值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH3	RW	通道 3 的实际中断的低值 复位值: 0x0

#### 4.6.2.18 ADC\_INT\_ACTUAL\_VALUE\_CH4

- 描述: 通道 4 的实际中断的高低值
- 偏移量: 0x70
- 默认值: 0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH4	RW	通道 4 的实际中断的高值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH4	RW	通道 4 的实际中断的低值

位段	名称	类型	说明
			复位值：0x0

#### 4.6.2.19 ADC\_INT\_ACTUAL\_VALUE\_CH5

- 描述：通道 5 的实际中断的高低值
- 偏移量：0x74
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH5	RW	通道 5 的实际中断的高值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH5	RW	通道 5 的实际中断的低值 复位值：0x0

#### 4.6.2.20 ADC\_INT\_ACTUAL\_VALUE\_CH6

- 描述：通道 6 的实际中断的高低值
- 偏移量：0x78
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH6	RW	通道 6 的实际中断的高值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH6	RW	通道 6 的实际中断的低值 复位值：0x0

#### 4.6.2.21 ADC\_INT\_ACTUAL\_VALUE\_CH7

- 描述：通道 7 的实际中断的高低值
- 偏移量：0x7C
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_ACTUAL_HVALUE_CH7	RW	通道 7 的实际中断的高值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_ACTUAL_LVALUE_CH7	RW	通道 7 的实际中断的低值 复位值: 0x0

#### 4.6.2.22 ADC\_INT\_DELTA\_VALUE\_CH0

- 描述: 通道 0 的 delta 中断的高低值
- 偏移量: 0x90
- 默认值: 0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH0	RW	通道 0 的 delta 中断的初始值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH0	RW	通道 0 的 delta 中断的 delta 值 复位值: 0x0

#### 4.6.2.23 ADC\_INT\_DELTA\_VALUE\_CH1

- 描述: 通道 1 的 delta 中断的高低值
- 偏移量: 0x94
- 默认值: 0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH1	RW	通道 1 的 delta 中断的初始值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH1	RW	通道 1 的 delta 中断的 delta 值

位段	名称	类型	说明
			复位值：0x0

#### 4.6.2.24 ADC\_INT\_DELTA\_VALUE\_CH2

- 描述：通道 2 的 delta 中断的高低值
- 偏移量：0x98
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH2	RW	通道 2 的 delta 中断的初始值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH2	RW	通道 2 的 delta 中断的 delta 值 复位值：0x0

#### 4.6.2.25 ADC\_INT\_DELTA\_VALUE\_CH3

- 描述：通道 3 的 delta 中断的高低值
- 偏移量：0x9C
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH3	RW	通道 3 的 delta 中断的初始值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH3	RW	通道 3 的 delta 中断的 delta 值 复位值：0x0

#### 4.6.2.26 ADC\_INT\_DELTA\_VALUE\_CH4

- 描述：通道 4 的 delta 中断的高低值
- 偏移量：0xA0
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH4	RW	通道 4 的 delta 中断的初始值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH4	RW	通道 4 的 delta 中断的 delta 值 复位值: 0x0

#### 4.6.2.27 ADC\_INT\_DELTA\_VALUE\_CH5

- 描述: 通道 5 的 delta 中断的高低值
- 偏移量: 0xA4
- 默认值: 0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH5	RW	通道 5 的 delta 中断的初始值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH5	RW	通道 5 的 delta 中断的 delta 值 复位值: 0x0

#### 4.6.2.28 ADC\_INT\_DELTA\_VALUE\_CH6

- 描述: 通道 6 的 delta 中断的高低值
- 偏移量: 0xA8
- 默认值: 0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH6	RW	通道 6 的 delta 中断的初始值 复位值: 0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH6	RW	通道 6 的 delta 中断的 delta 值

位段	名称	类型	说明
			复位值：0x0

#### 4.6.2.29 ADC\_INT\_DELTA\_VALUE\_CH7

- 描述：通道 7 的 delta 中断的高低值
- 偏移量：0xAC
- 默认值：0x0

位段	名称	类型	说明
[31:28]	RESERVED_2	-	
[27:16]	ADC_INT_DELTA_INIT_CH7	RW	通道 7 的 delta 中断的初始值 复位值：0x0
[15:12]	RESERVED_1	-	
[11:0]	ADC_INT_DELTA_VALUE_CH7	RW	通道 7 的 delta 中断的 delta 值 复位值：0x0

#### 4.6.2.30 ADC\_DFX\_SOC\_CNT\_CH0

- 描述：通道 0 的 SOC 计数器值
- 偏移量：0xC0
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH0	RC	通道 0 的 SOC 计数器值 复位值：0x0

#### 4.6.2.31 ADC\_DFX\_EOC\_CNT\_CH0

- 描述：通道 0 的 EOC 计数器值
- 偏移量：0xC4
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	



位段	名称	类型	说明
[23:0]	ADC_DFX_EOC_CNT_CH0	RC	通道 0 的 EOC 计数器值 复位值: 0x0

#### 4.6.2.32 ADC\_DFX\_SOC\_CNT\_CH1

- 描述: 通道 1 的 SOC 计数器值
- 偏移量: 0xC8
- 默认值: 0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH1	RC	通道 1 的 SOC 计数器值 复位值: 0x0

#### 4.6.2.33 ADC\_DFX\_EOC\_CNT\_CH1

- 描述: 通道 1 的 EOC 计数器值
- 偏移量: 0xCC
- 默认值: 0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH1	RC	通道 1 的 EOC 计数器值 复位值: 0x0

#### 4.6.2.34 ADC\_DFX\_SOC\_CNT\_CH2

- 描述: 通道 2 的 SOC 计数器值
- 偏移量: 0xD0
- 默认值: 0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH2	RC	通道 2 的 SOC 计数器值 复位值: 0x0

#### 4.6.2.35 ADC\_DFX\_EOC\_CNT\_CH2

- 描述：通道 2 的 EOC 计数器值
- 偏移量：0xD4
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH2	RC	通道 2 的 EOC 计数器值 复位值：0x0

#### 4.6.2.36 ADC\_DFX\_SOC\_CNT\_CH3

- 描述：通道 3 的 SOC 计数器值
- 偏移量：0xD8
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH3	RC	通道 3 的 SOC 计数器值 复位值：0x0

#### 4.6.2.37 ADC\_DFX\_EOC\_CNT\_CH3

- 描述：通道 3 的 EOC 计数器值
- 偏移量：0xDC
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH3	RC	通道 3 的 EOC 计数器值 复位值：0x0

#### 4.6.2.38 ADC\_DFX\_SOC\_CNT\_CH4

- 描述：通道 4 的 SOC 计数器值
- 偏移量：0xE0
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH4	RC	通道 4 的 SOC 计数器值 复位值: 0x0

#### 4.6.2.39 ADC\_DFX\_EOC\_CNT\_CH4

- 描述: 通道 4 的 EOC 计数器值
- 偏移量: 0xE4
- 默认值: 0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH4	RC	通道 4 的 EOC 计数器值 复位值: 0x0

#### 4.6.2.40 ADC\_DFX\_SOC\_CNT\_CH5

- 描述: 通道 5 的 SOC 计数器值
- 偏移量: 0xE8
- 默认值: 0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH5	RC	通道 5 的 SOC 计数器值 复位值: 0x0

#### 4.6.2.41 ADC\_DFX\_EOC\_CNT\_CH5

- 描述: 通道 5 的 EOC 计数器值
- 偏移量: 0xEC
- 默认值: 0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH5	RC	通道 5 的 EOC 计数器值

位段	名称	类型	说明
			复位值：0x0

#### 4.6.2.42 ADC\_DFX\_SOC\_CNT\_CH6

- 描述：通道 6 的 SOC 计数器值
- 偏移量：0xF0
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH6	RC	通道 6 的 SOC 计数器值 复位值：0x0

#### 4.6.2.43 ADC\_DFX\_EOC\_CNT\_CH6

- 描述：通道 6 的 EOC 计数器值
- 偏移量：0xF4
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH6	RC	通道 6 的 EOC 计数器值 复位值：0x0

#### 4.6.2.44 ADC\_DFX\_SOC\_CNT\_CH7

- 描述：通道 7 的 SOC 计数器值
- 偏移量：0xF8
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_SOC_CNT_CH7	RC	通道 7 的 SOC 计数器值 复位值：0x0

#### 4.6.2.45 ADC\_DFX\_EOC\_CNT\_CH7

- 描述：通道 7 的 EOC 计数器值
- 偏移量：0xFC
- 默认值：0x0

位段	名称	类型	说明
[31:24]	RESERVED_1	-	
[23:0]	ADC_DFX_EOC_CNT_CH7	RC	通道 7 的 EOC 计数器值 复位值：0x0



- 基于飞利浦 I2S 串行协议的 I2S 发送器和接收器
- 串行主和串行从操作
- FIFO-发送和接收 FIFO 的深度为 32，宽度为 32 位。
- 可编程 FIFO 阈值
- 音频数据的分辨率为 16、24 或 32 位。I2S 可以将数据从 16 位转换为 24 位或 32 位，反之亦然。当 I2S 工作在从模式时，它可以自动检测传输的数据是 16 位、24 位还是 32 位。
- DMA 控制器接口：使 I2S 能够通过 APB 总线与 DMA 控制器连接，使用握手接口传输请求。
- 中断和错误的单独屏蔽-所有单个中断和错误都可以单独屏蔽。
- 从 I2S 到中断控制器的一条组合中断线
- 从 I2S 到错误控制器的一条组合错误线
- 兼容三种串行音频格式：左对齐、I2S、右对齐。
- 多块同步的 I2S 使能信号
- 支持音频样本压缩。

### 5.3 接口

I2S 的 PAD 如下图表 5-2 所述：

图表 5-2 管脚描述

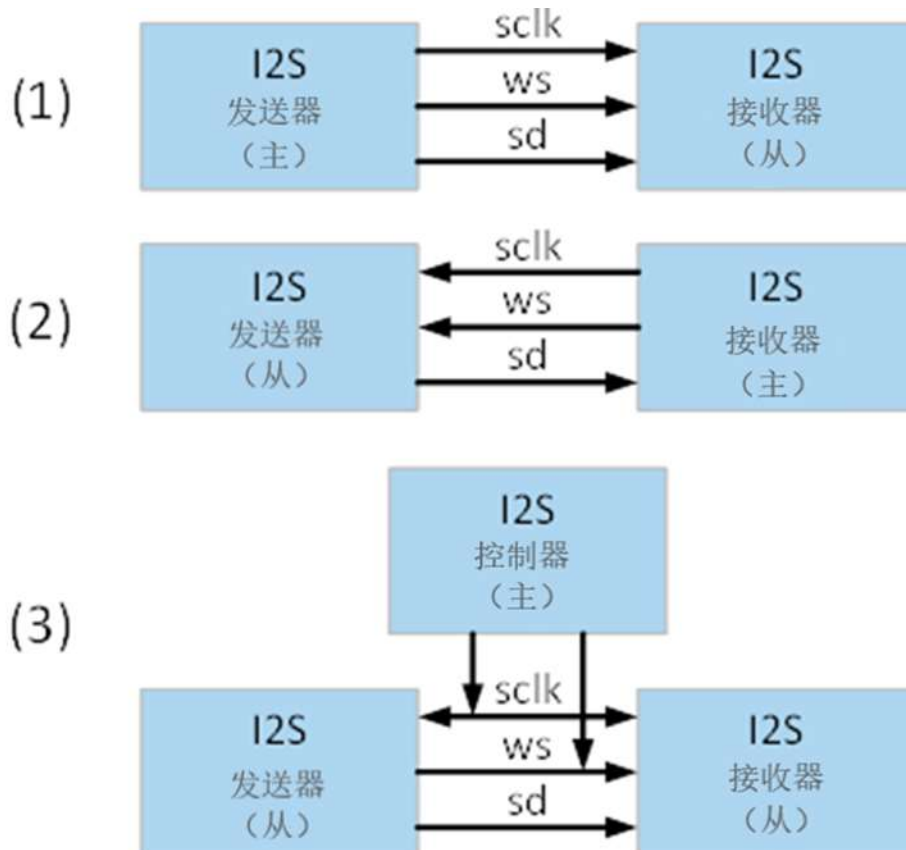
管脚名称	方向	宽度	说明
I2S_SDA0	IO	1	串行数据 0
I2S_SDA1	IO	1	串行数据 1
I2S_SDA2	IO	1	串行数据 2
I2S_SDA3	IO	1	串行数据 3
I2S_MCLK	O	1	I2S 作为主时的串行时钟
I2S_SCLK	I	1	I2S 作为从时的串行时钟
I2S_WS	IO	1	字段选择信号

### 5.4 功能描述

该总线由串行数据线（sd）、字段选择线（ws）和串行时钟（sclk）组成。串行时钟是时分复用的，以允许传输两个数据流（例如，左右立体声数据）。

图表 5-3 说明了 I2S 组件的三个简单系统配置。请注意，这些示例显示了 I2S 组件的第二个实例化，它充当了配置为从或主的接收器。

图中的示例 1 和示例 2 表明发送器或接收器都可以充当总线主。主负责生成共享的 sclk 和 ws 时钟信号。在可能有多个发送器和接收器的复杂系统中，可以使用单独的系统主。如图中的示例 3 所示，系统主也可以与系统中的一个发射器或接收器组合。本例中“控制器”的启用和禁用是通过将组件配置为主，并对时钟使能寄存器和时钟配置寄存器进行编程来实现的。

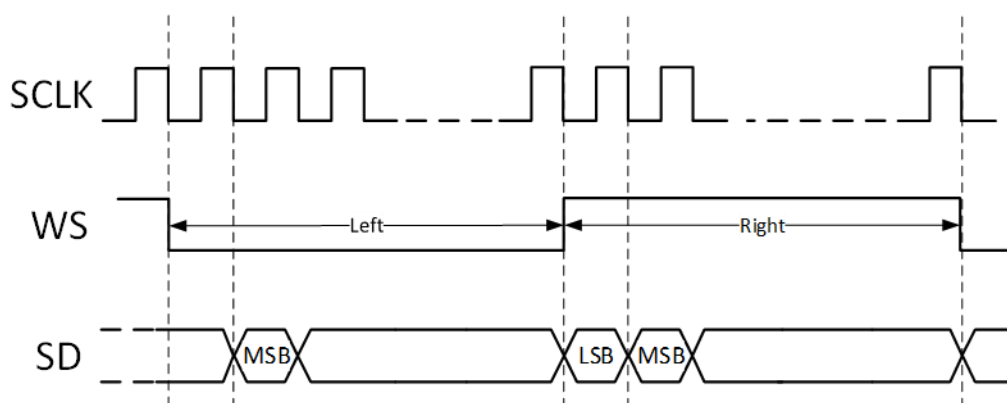


图表 5-3 I2S 的简单系统配置

串行数据以二进制补码格式传输，最高有效位（MSB）在前。这意味着发送器和接收器可以有不同的字长，发送器和接收器不需要知道对方可以处理什么大小的字。如果正在传输的字对于接收器来说太大，将截断最低有效位（LSB）。同样，如果字长小于接收器可以处理的大小，数据将被填充为零。

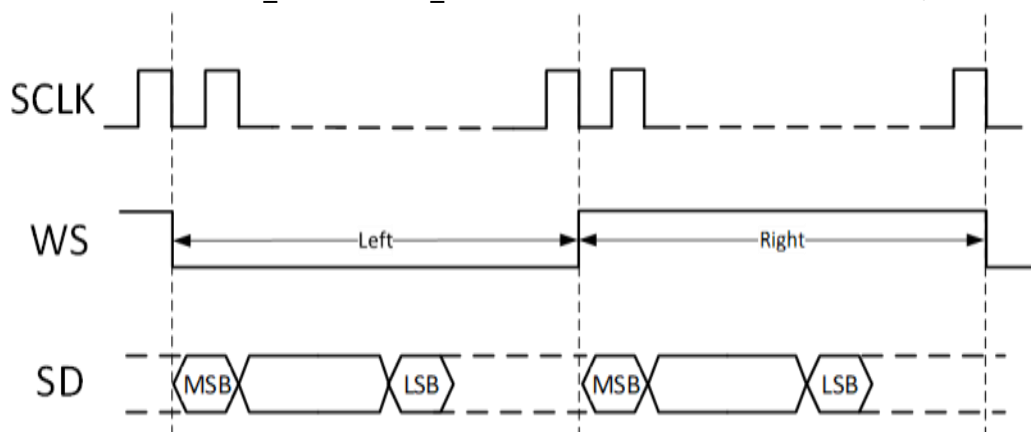
字段选择线用于控制时分复用的数据流。例如，当 ws 为低时，传输的字是左立体声数据；当 ws 为高时，传输的数据是右立体声数据。它可以通过 IISCNF\_IN 和 IISCNF\_OUT 寄存器中的 RALOLRC 和 LALOLRC 进行配置。这种格式如图表 5-4 所示。对于标准 I2S 格式，字的 MSB 在 ws 更改后一个 sclk 周期发送。发送器发送的串行数据可以与 sclk 信号的下降沿或上升沿同步。但是，接收器必须在 sclk 的上升沿锁存串行数据。



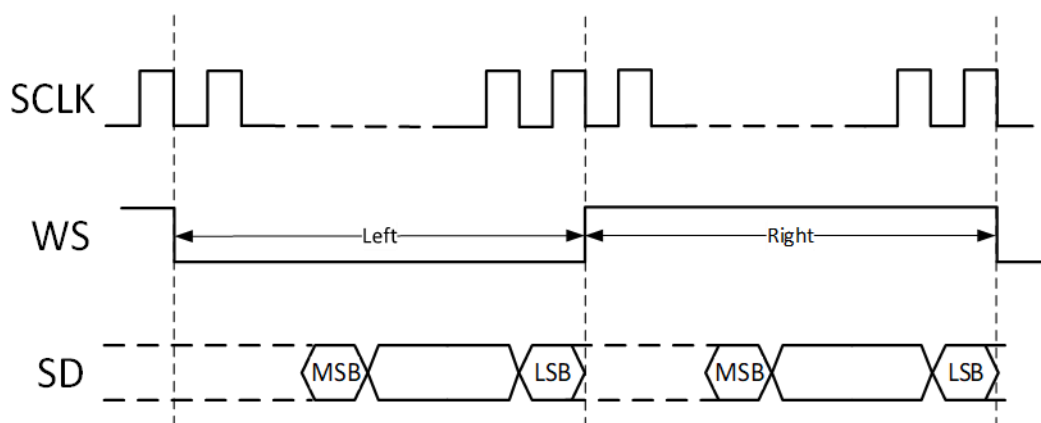


图表 5-4 I2S 标准立体帧格

同时，该 I2S 模块可以接收和发送左对齐格式（如图表 5-5 所示）和右对齐格式（如图表 5-6 所示）的音频数据，帧格式可以通过 IISCNF\_IN 和 IISCNF\_OUT 寄存器中的 RSAFS 和 TSAFS 配置。



图表 5-5 左对齐的 I2S 立体帧格式



图表 5-6 右对齐的 I2S 立体帧格式

### 5.4.1 复位

I2S 模块中只有一个复位，APB 复位 rst\_b 低电平有效。复位后，所有寄存器、FSM、计数器、DMA 接口、中断接口等都为默认值，两个 FIFO 都为初始值，所有内部产生的时钟都关闭。

## 5.4.2 启用 I2S

用户必须先启用 I2S 组件，然后才能在 FIFO 中接收和发送数据。要启用 I2S，请将 I2S 启用 (I2SEN) 位 (偏移量: 0x00, I2SEN[0]) 设置为 1。要禁用 I2S，请将此位设置为 0。

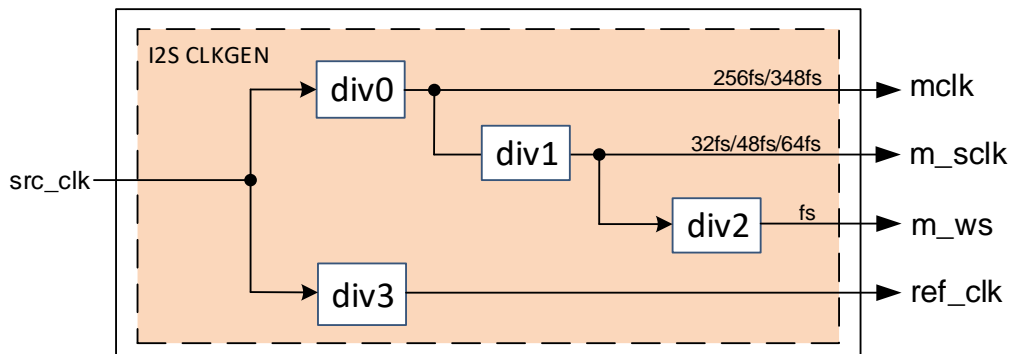
禁用后，会发生以下事件：

- 发送和接收 FIFO 被清除，读/写指针复位。
- 发送和接收过程中的数据丢失。
- 组件中的所有其他可编程使能信号（例如发送器/接收器模块使能信号和单独的 TX/RX 通道使能）被覆盖。

## 5.4.3 时钟

### 5.4.3.1 电路图

I2S 支持 16 位、24 位和 32 位数据，并且有多种采样频率 (fs)。字段选择信号 (ws) 的频率是采样频率。常用的采样频率为 8kHz、11.025kHz、12kHz、16kHz、22.05kHz、24kHz、32kHz、44.1kHz、48kHz、64kHz、88.2kHz、96kHz 和 192kHz。为了方便用户，这里 I2S 只需要用户提供有一个源时钟 (src\_clk)，然后 I2S 时钟发生器模块会通过三个分频器产生采样频率。I2S 时钟发生器电路图如图表 5-7 所示：



图表 5-7 I2S 时钟发生器电路图

### 5.4.3.2 分析信号

src\_clk：源时钟，来自系统时钟。这里我们需要一个 src\_clk 如图表 5-8 所示：

图表 5-8 时钟配置

采样频率 (kHz)		8	11.025	16	22.5	24	32	44.1	48	64	88.2	96	192
Mclk	256*fs	2.048	2.8224	4.096	5.76	6.144	8.192	11.2896	12.288	16.384	22.5792	24.576	49.152
(kHz)	384*fs	3.072	4.2336	6.144	8.64	9.216	12.288	16.9344	18.432	24.576	33.8688	36.864	73.728

采样频率 (kHz)	8	11.025	16	22.5	24	32	44.1	48	64	88.2	96	192
Src_clk	@ 主模式: $\text{src\_clk} = \text{mclk} * n$ ( $n = \text{整数}, n > 0$ ), $\text{Src\_clk} > \text{sclk} * 10/3$ @从模式: $\text{Src\_clk} > \text{sclk} * 20/3$											

- mclk: 主时钟, 来自 DIV0 分频器对 src\_clk 的分频, 连接到输出接口。用户需要配置寄存器 FSSTA[16] 来选择你会得到哪个 mclk,  $256 * fs$  或  $384 * fs$ , 并配置寄存器 DIV0\_LEVEL[7:0] 来分频 src\_clk。确保 mclk 频率为  $256 * fs$  或  $384 * fs$ 。

$$\text{mclk} = \text{src\_clk} / \text{div0} = 256 \text{ fs 或 } 384 \text{ fs}$$

- m\_sclk: 串行时钟, 用户需要配置寄存器 FSSTA[13:12] 来选择你需要的 m\_sclk。

$$\text{m\_sclk} = \text{m\_ws} * 2 * \text{data\_bit\_width} = 32 * fs \text{ 或 } 48 * fs \text{ 或 } 64 * fs$$

- m\_ws: 采样频率, 即 fs。它必须固定为常用的采样频率。
- ref\_clk: 参考时钟, 用于从检测 I2S 的输入采样频率。

$$\text{ref\_clk} = \text{src\_sclk} / ((\text{DIV3}[7:0] + 1) * 2)$$

### 5.4.3.3 I2S 时钟发生器中的可配置寄存器

为了通过分频 src\_sclk 生成常用的采样频率, 用户需要配置这些寄存器: DIV0\_LEVEL、DIV1\_LEVEL、DIV2\_LEVEL 和 DIV3\_LEVEL。

- $\text{mclk} = \text{src\_clk}$                        $\text{DIV0}[7:0] = 0$   
 $\text{mclk} = \text{src\_clk} / \text{DIV0}[7:0]$        $\text{DIV0}[7:0] \geq 1$   
 例如:
  - $\text{DIV0}[7:0] = 0x0a \rightarrow \text{mclk} = \text{src\_clk} / 10$
  - $\text{DIV0}[7:0] = 0x19 \rightarrow \text{mclk} = \text{src\_clk} / 25$
- $\text{ref\_clk} = \text{src\_sclk} / ((\text{DIV3}[7:0] + 1) * 2)$ 
  - 此参考时钟用于从模式以检测 I2S 输入采样频率。
  - 对于所有的采样频率, 用户应该在这个频率下通过配置 DIV3\_LEVEL 寄存器中的 DIV3[7:0] 来生成 ref\_clk:

$$3.072 \text{ MHz} \leq \text{ref\_clk} \leq 4.032 \text{ MHz}$$

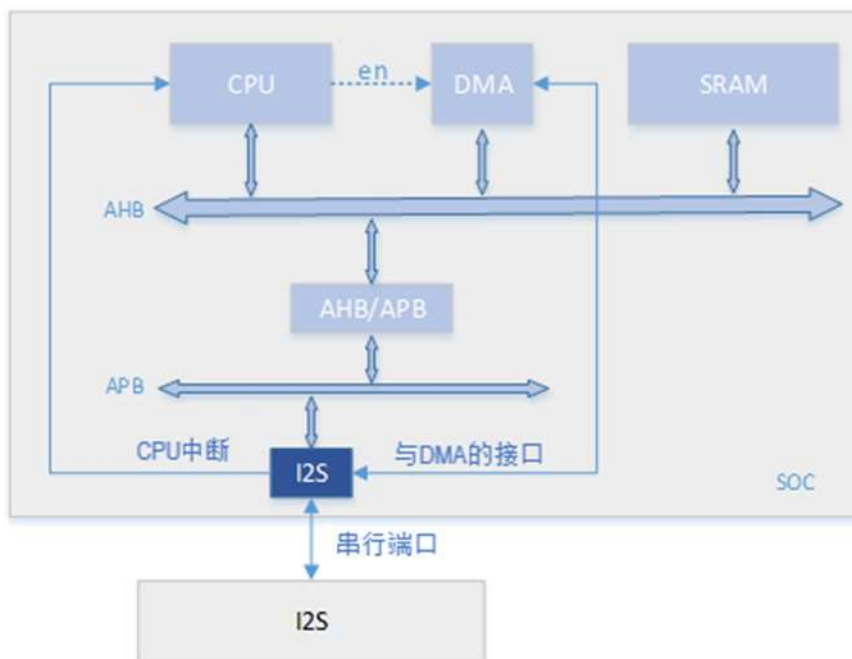
- 然后用户可以在寄存器 FADTLR 和 FSSTA 中查看输入采样频率的计算结果。

### 5.4.4 DMA 接口

发送和接收 FIFO 在 PCLK 时钟域中都有 DMA 接口 (dma\*\_ack、dma\*\_req 和 dma\*\_single), 用于减少 CPU 过载。

- dma\*\_ack: 该信号在当前事务中最后一次 APB 传输的数据阶段之后生效。
- dma\*\_req: 该信号在 I2S 模块请求突发事务时生效。

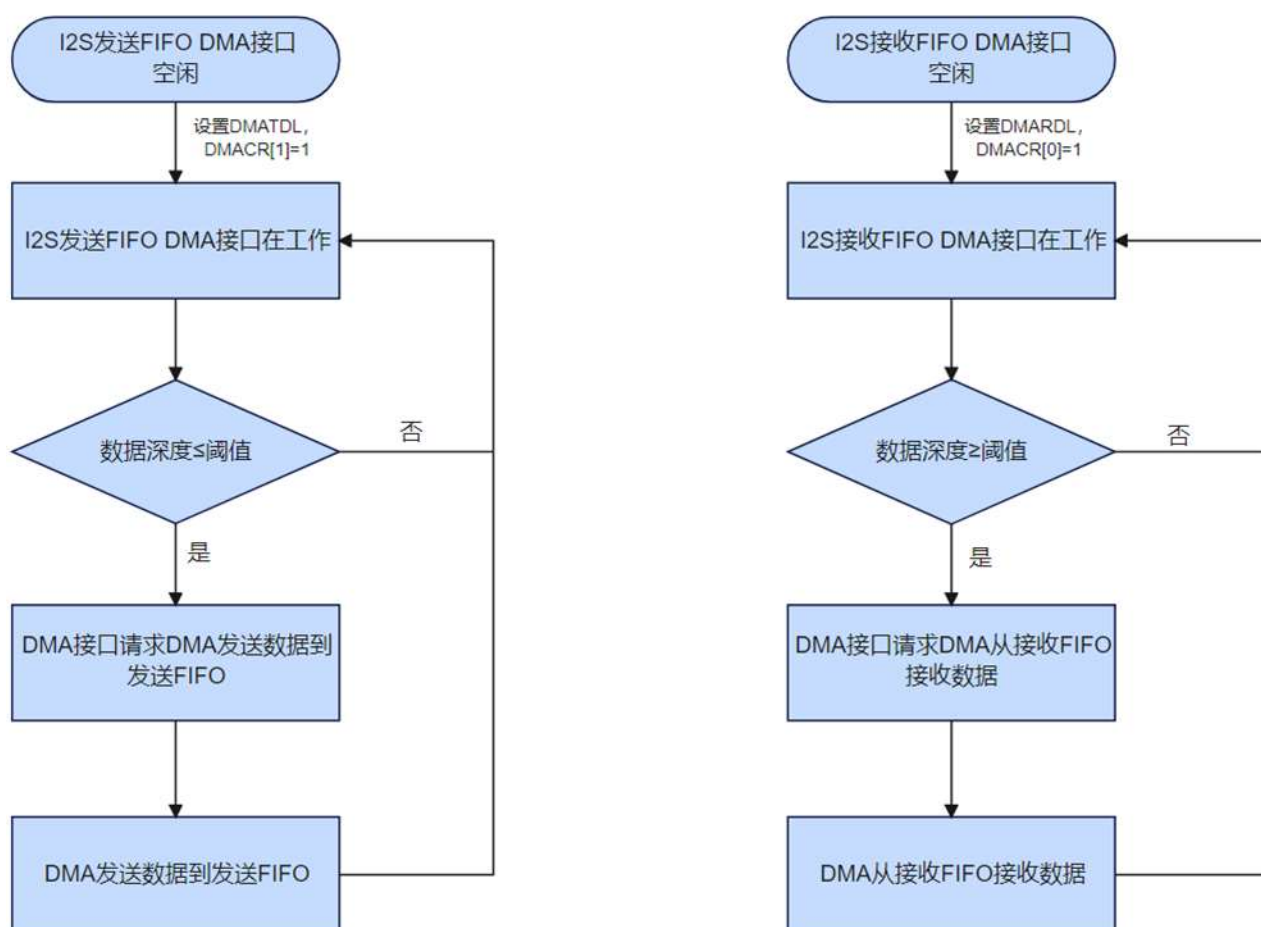
- dma\*\_single: 这是一个状态信号，当 I2S 模块可以传输至少一个数据时，此信号生效。该信号仅适用于支持该功能的 DMA。



图表 5-9 系统中的 I2S

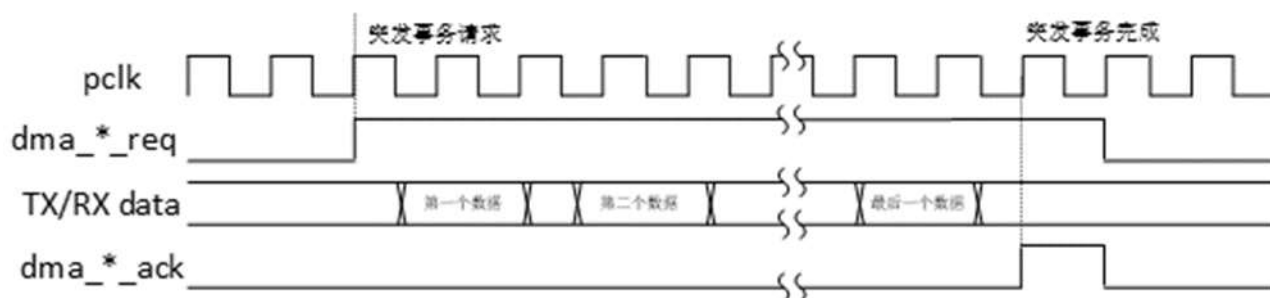
图表 5-9 显示了 I2S 在系统中的位置和连接。当用户想要使用 I2S DMA 接口时，必须按以下流程配置 DMA 和 I2S。

1. CPU 配置 I2S 控制寄存器，然后使能 I2S。
2. CPU 配置 DMA 控制寄存器，然后使能 I2S DMA 接口。
3. I2S 通过数据通道 APB 和控制通道 DMA 接口与 DMA 传输数据。
4. 如果 CPU 要访问 I2S，必须等到 DMA 与 I2S 数据传输完成。CPU 可以查看 I2S 寄存器或中断的状态。



图表 5-10 发送 FIFO 和接收 IFIFO DMA 接口工作流程

dma\*\_req 是在满足阈值触发条件时由事件触发的。通过 DMA 接口完成 DMA 突发事务后，DMA 将向 DMA 接口断言 dma\*\_ack 反馈，然后 dma\*\_req 将被清除。此后，仍然满足阈值触发条件，重新设置 dma\_req\_\*。DMA 接口时序如图表 5-11 所示。



图表 5-11 DMA 接口时序

为了使 FIFO 不溢出，用户必须根据 DMA 接口阈值配置 DMA 事务大小。有关详细信息，请参阅 DMA 的参考文档。

## 5.4.5 中断和错误

I2S 模块中的中断和错误主要分为三种：一种是关于 FIFO 水平的（共 2 个独立中断和 4 个独立错误），一种是关于传输模式（共 3 个独立中断），另一种是关于寄存器地址（共有 1 个单独的错误）。复位后默认屏蔽所有的中断和错误。这些单独的中断和错误被异或在一起以形成在外部看到的组合中断或错误信号。中断或错误为高电平有效。

### 5.4.5.1 I2S FIFO 水平中断

- 发送 FIFO 阈值空中断 (i2s\_txe\_intr): 当发送 FIFO 等于或低于其阈值并需要服务以防止欠载时设置。通过软件可编程寄存器设置的阈值决定了产生中断的发送 FIFO 条目的水平。
- 接收 FIFO 阈值满中断 (i2s\_rxf\_intr): 当接收 FIFO 等于或高于其阈值加 1 并且需要服务以防止溢出时设置。通过软件可编程寄存器设置的阈值确定产生中断的接收 FIFO 条目的水平。

### 5.4.5.2 I2S FIFO 水平错误

- 发送 FIFO 溢出错误 (i2s\_txo\_err): 当 APB 访问在发送 FIFO 完全填满后尝试向其写入时设置。设置后，从 APB 写入的数据将被丢弃。
- 发送 FIFO 下溢错误 (i2s\_txu\_err): 当 APB 访问在接收 FIFO 为空时尝试从其读取时设置。设置后，从发送 FIFO 中读回零。
- 接收 FIFO 溢出错误 (i2s\_rxo\_err): 当接收逻辑在接收 FIFO 完全填满后尝试将数据放入接收 FIFO 时设置。设置后，新接收的数据将被丢弃。
- 接收 FIFO 下溢错误 (i2s\_rxu\_err): 当 APB 访问在接收 FIFO 为空时尝试从其读取时设置。设置后，从接收 FIFO 中读回零。

### 5.4.5.3 I2S 传输模式中断

- I2S RX 忙标志变化 (i2s\_rxbfc\_intr): 当 I2S RX 状态从空闲变为忙或从忙变为空闲时设置。
- I2S TX 忙标志变化 ((i2s\_txbfc\_intr): 当 I2S TX 状态从空闲变为忙或从忙变为空闲时设置。
- 输入采样频率变化 (in\_fsc\_intr): 当输入采样音频频率变化时设置。

### 5.4.5.4 I2S 寄存器地址错误

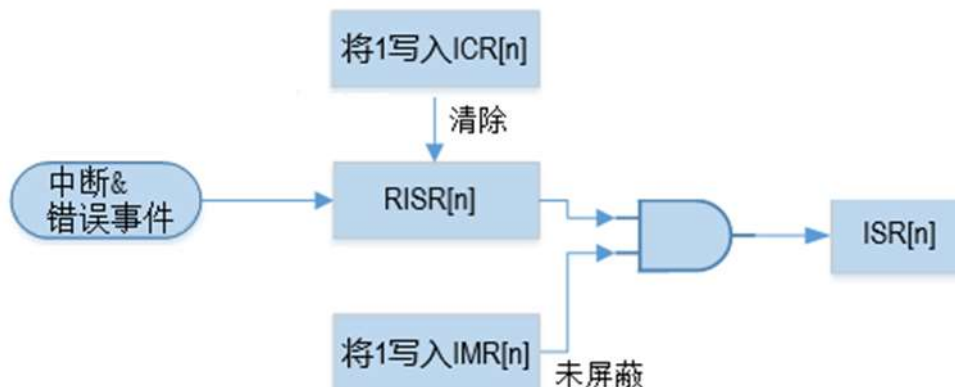
- I2S 错误寄存器地址错误 (i2s\_waddr\_err): 当 APB 访问尝试读取或写入错误地址中的寄存器时设置。

### 5.4.5.5 中断&错误寄存器之间的关系

每个中断或错误由控制寄存器 ICR 和 IMR 中的控制位控制, 每个中断或错误将在状态寄存器 RISR 和 ISR 中产生两个状态位。

如果发生中断或错误事件, RISR 将被设置, 状态值保持为 1, 直到向 ICR 寄存器写入 1 将其清除。如果 IMR 为 0, 中断或错误将被屏蔽。

这些寄存器之间的关系如图表 5-12 所示:



图表 5-12 中断&错误寄存器之间的关系

## 5.5 使用

外围子系统的 I2S 模块有两个用途:

- 通用音频接口的输入和输出
- HDMI 音频输出

有关如何配置 I2S 的详细信息, 请参阅章节。

如何配置 I2S 软件, 请参考功能描述和寄存器描述部分。

## 5.6 寄存器描述

### 5.6.1 寄存器内存映射

图表 5-13 I2S 内存映射

寄存器名	偏移量	说明	章节/页码
IISEN	0x00	IIS_IO 使能寄存器 复位值: 0x00	<a href="#">5.6.2.1/197</a>
FUNCMODE	0x04	IIS_IO 功能模式 复位值: 0x00	<a href="#">5.6.2.2/197</a>

寄存器名	偏移量	说明	章节/页码
IISCNF_IN	0x08	RX 侧的 I2S 接口配置 复位值: 0x00	<a href="#">5.6.2.3/198</a>
FSSTA	0x0C	I2S ATX 音频输入控制/状态寄存器 复位值: 0xf0	<a href="#">5.6.2.4/200</a>
IISCNF_OUT	0x10	TX 侧的 I2S 接口配置 复位值: 0x00	<a href="#">5.6.2.5/205</a>
FADTLR	0x14	I2S 采样频率自动检测阈值水平寄存器 复位值: 0x00	<a href="#">5.6.2.6/206</a>
SCCR	0x18	样本压缩控制寄存器 复位值: 0x00	<a href="#">5.6.2.7/207</a>
TXFTLR	0x1C	发送 FIFO 阈值水平 复位值: 0x10	<a href="#">5.6.2.8/210</a>
RXFTLR	0x20	接收 FIFO 阈值水平 复位值: 0x10	<a href="#">5.6.2.9/211</a>
TXFLR	0x24	发送 FIFO 水平寄存器 复位值: 0x00	<a href="#">5.6.2.10/211</a>
RXFLR	0x28	接收 FIFO 水平寄存器 复位值: 0x00	<a href="#">5.6.2.11/211</a>
SR	0x2C	状态寄存器 复位值: 0x0C	<a href="#">5.6.2.12/212</a>
IMR	0x30	中断屏蔽寄存器 复位值: 0x7F	<a href="#">5.6.2.13/213</a>
ISR	0x34	中断状态寄存器 复位值: 0x20	<a href="#">5.6.2.14/214</a>
RISR	0x38	原始中断状态寄存器 复位值: 0x20	<a href="#">5.6.2.15/215</a>
ICR	0x3C	中断清除寄存器 复位值: 0x00	<a href="#">5.6.2.16/216</a>



寄存器名	偏移量	说明	章节/页码
DMACR	0x40	DMA 控制寄存器 复位值: 0x00	<a href="#">5.6.2.17/217</a>
DMATDLR	0x44	DMA 发送数据水平 复位值: 0x10	<a href="#">5.6.2.18/218</a>
DMARDLR	0x48	DMA 接收数据水平 复位值: 0x00	<a href="#">5.6.2.19/218</a>
DIV0_LEVEL	0x50	分频源时钟, 得到 mclk 复位值: 0x00	<a href="#">5.6.2.20/219</a>
DIV3_LEVEL	0x54	分频源时钟, 得到参考时钟 复位值: 0x00	<a href="#">5.6.2.21/219</a>

## 5.6.2 寄存器和字段描述

### 5.6.2.1 I2SEN

- 寄存器名: I2S 使能寄存器
- 描述: 该寄存器控制 I2S 启用, 通过写 I2SEN 位启用和禁用 I2S。
- 偏移量: 0x00

图表 5-14 I2SEN 字段说明

位段	名称	类型	说明
31:1	保留, 读为零。		
0	I2SEN	R/W	I2S 使能位 1: 使能 0: 禁用 (默认)

### 5.6.2.2 FUNCMODE

- 寄存器名: 功能模式寄存器
- 描述: 该寄存器控制功能模式, I2S 使能时无法写入该寄存器。只有当相应的 MODE\_wen 同时生效时才能写入 MODE 位。对 MODE 位或 MODE\_wen 位的一次写入是无用的, 将被忽略。读 MODE\_wen 位将始终返回 0。
- 偏移量: 0x04

图表 5-15 FUNCMODE 字段说明

位段	名称	类型	说明
31:12	保留，读为零。		
11:8	CH_SEL	R/W	I2S 通道选择 CH_SEL[n]: 通道[n] 使能 0: 通道[n]关闭 1: 通道[n]打开
7:5	保留，读为零。		
4	I2S_RMODE	R/W	使能 RX 模式 0: 模块处于接收模式。（默认） 1: 模块关闭接收模式。
3:1	保留，读为零。		
0	I2S_TMODE	R/W	使能 TX 模式 0: 模块处于发送模式。（默认） 1: 模块关闭发送模式。

### 5.6.2.3 IISCNF\_IN

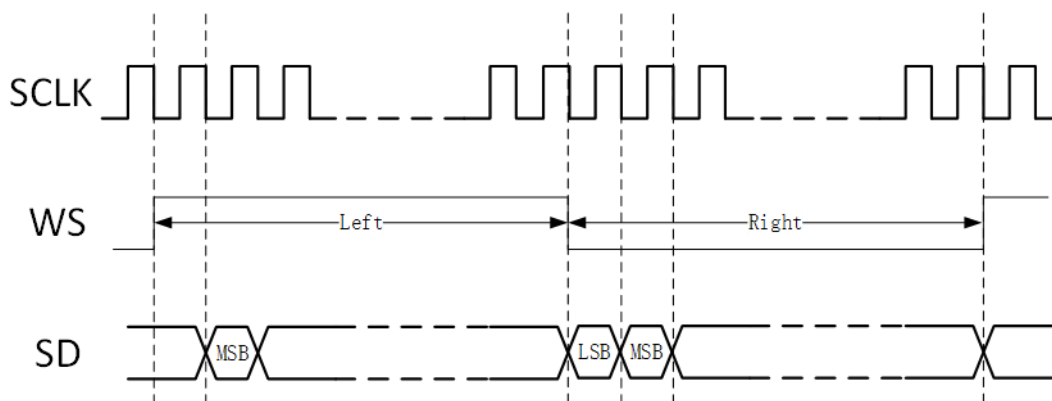
- 寄存器名：I2S 接收器输入接口模式
- 描述：该寄存器控制 I2S 输入接口格式，I2S 使能时无法写入该寄存器。
- 偏移量：0x08

图表 5-16 IISCNF\_IN 字段说明

位段	名称	类型	说明
31:14	保留，读为零。		
13:12	RDELAY1	R/W	I2S 接收器 s_sclk 和 s_ws 延迟水平 00: 无延迟（默认） 01: 添加 1 个 src_clk 周期延迟。 10: 添加 2 个 src_clk 周期延迟。 11: 添加 3 个 src_clk 周期延迟。
11:10	保留，读为零。		

位段	名称	类型	说明
9	RX_CLK_SEL	R/W	I2S 从设备模式下接收时钟源选择 0: src_clk (默认) 1: mclk_i
8	I2S_RXMODE	R/W	I2S 接收器操作模式选择 0: 从模式 (默认) 1: 主模式
7:6	保留, 读为零。		
5	RX_CH_SEL	R/W	I2S 单通道模式下采样通道选择 0: 右声道 (默认) 1: 左声道
4	RX_VOICE_EN	R/W	I2S 采样源类型选择 0: 源为立体声, 具有不同的左右声道信号。(默认) 1: 源为单声道, 左右声道数据相同, 只接收并存储其中一个。
3	-	N/A	保留
2	RALOLRC	R/W	左/右声道的有效电平 0: 左声道为低电平。(默认) 1: 左声道为高电平。(右声道为低电平)
1:0	RSAFS	R/W	串行音频格式选择 0x0: I2S (默认) 0x1: 右对齐 0x2: 左对齐
注:			
1: 在 I2S 接收模式下, SD 信号由其他设备提供。它会在 PAD 输入信号的过程中造成延迟。			

左声道音频数据总是在前, 图表 5-17 显示了 RALOLRC=1 时的 I2S 总线格式。



图表 5-17 标准立体声帧格式 @RALOLRC=1

### 5.6.2.4 FSSTA

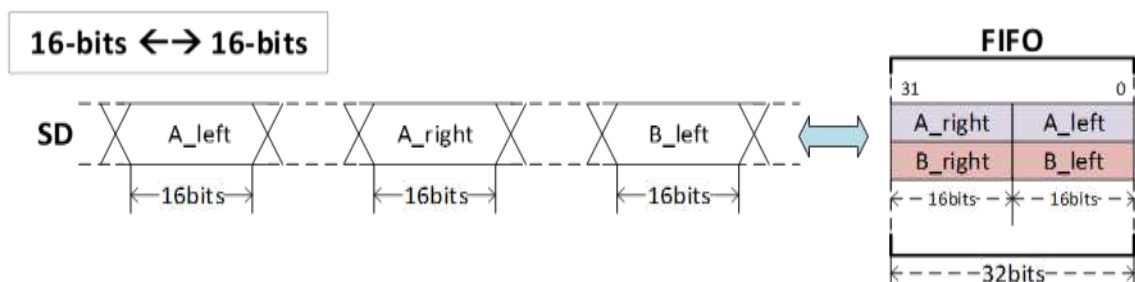
- 寄存器名：I2S 串行音频输入控制寄存器
- 描述：该寄存器控制 I2S 的数据宽度模式和从时钟的自动检测。I2S 使能时无法写入该寄存器。
- 偏移量：0x0C

图表 5-18 FSSTA 字段说明

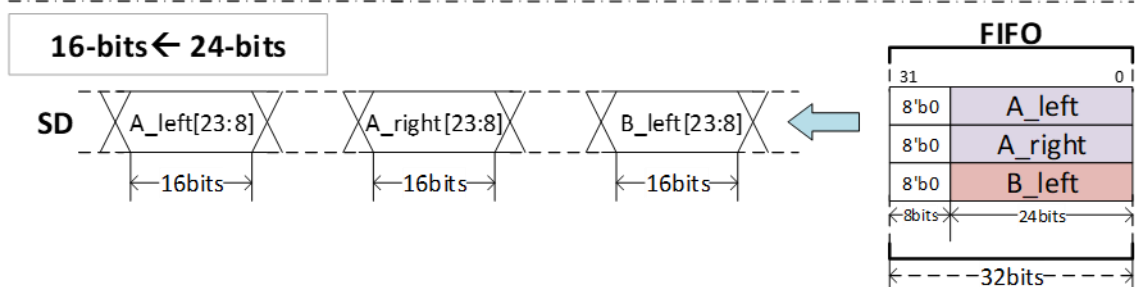
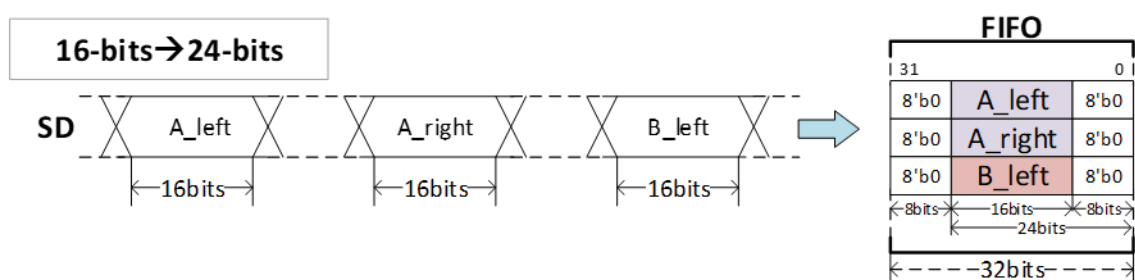
位段	名称	类型	说明
31:17	保留，读为零。		
16	MCLK_SEL	R/W	mclk 频率选择 0: mclk = 256*fs (默认) 1: mclk = 384*fs
15:14	保留，读为零。		
13:12	SCLK_SEL	R/W	sclk 频率选择 00: sclk = 32*fs (默认) 01: sclk = 48*fs 1?: sclk = 64*fs
11:8	DATAWTH	R/W	I2S 数据宽度模式 0000: 16 位输入/输出 (左右声道) 和 FIFO 以 16 位存储数据。(默认) 0001: 16 位输入/输出 (左右声道) 和 FIFO 以 24 位存储数据。 001?: 16 位输入/输出 (左右声道) 和 FIFO 以 32 位存储数据。 0100: 24 位输入/输出 (左右声道) 和 FIFO 以 16 位存储

位段	名称	类型	说明
			数据。 0101: 24 位输入/输出 (左右声道) 和 FIFO 以 24 位存储数据。 011?: 24 位输入/输出 (左右声道) 和 FIFO 以 32 位存储数据。 1000: 32 位输入/输出 (左右声道) 和 FIFO 以 16 位存储数据。 1001: 32 位输入/输出 (左右声道) 和 FIFO 以 24 位存储数据。 1?1?: 32 位输入/输出 (左右声道) 和 FIFO 以 32 位存储数据。
7:6	ARS	R	音频速率比例因子 (仅限 RX 模式) 00: 1 (默认) 01: 0.5 10: 0.25 11: 0.125 详见表 4-6。
5:4	AFR	R	输入音频采样频率基本速率 (仅限 RX 模式) 00: 88.2KHz (默认) 01: 96KHz 10: 64KHz 11: 192KHz 详见表 4-6。
3:1	保留, 读为零。		
0	AIRAD	R/W	音频输入速率自动检测位 (仅限 RX 模式) 0: 不检测。(默认) 1: 音频输入速率由硬件自动检测。(仅限 RX 模式)

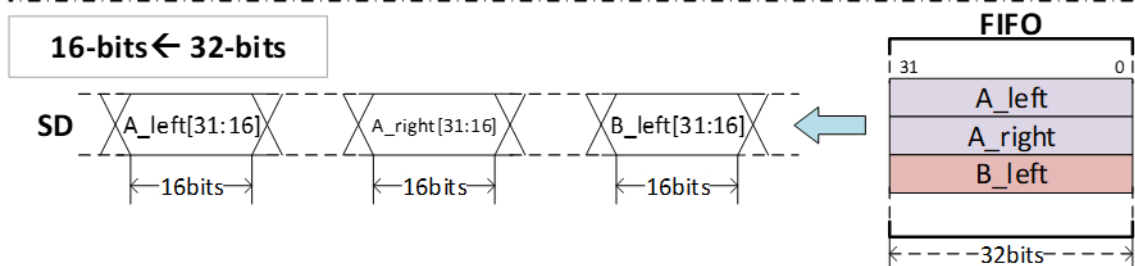
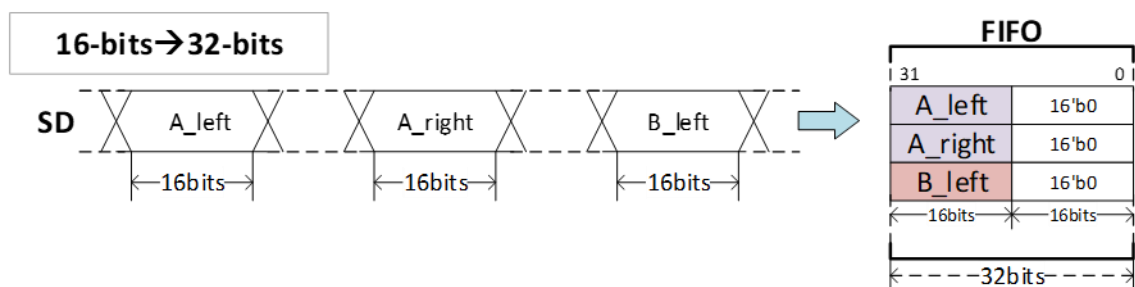
FIFO 数据宽度为 32 位, FIFO 以 16 位存储数据意味着一次可以存储两个样本。FIFO 以 24 位和 32 位存储数据意味着它可以一次存储一个样本。数据宽度变化如 [图表 5-19](#) 到 [图表 5-27](#):



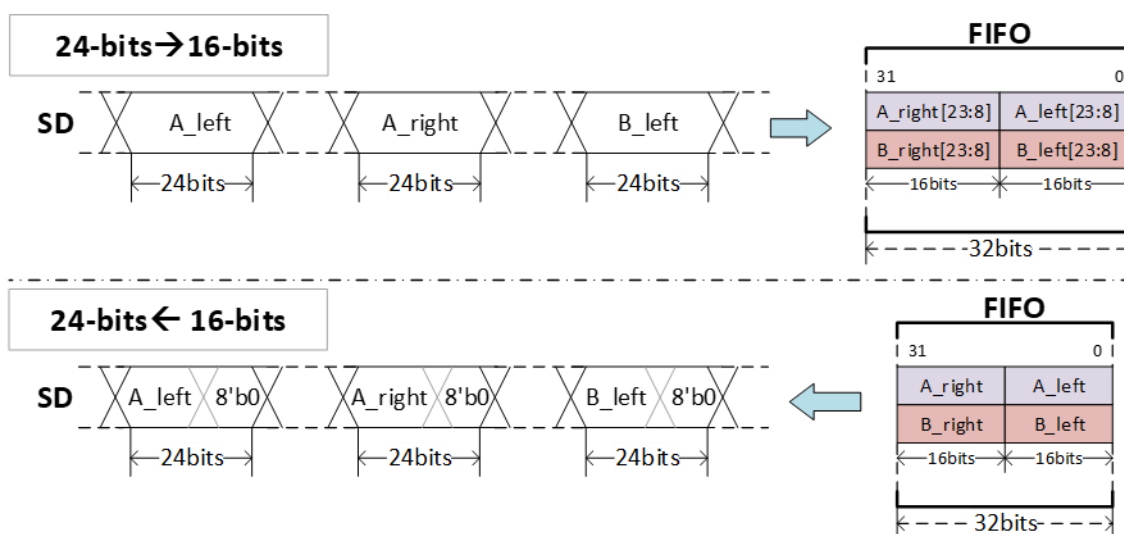
图表 5-19 16 位输入/输出和 FIFO 以 16 位存储数据



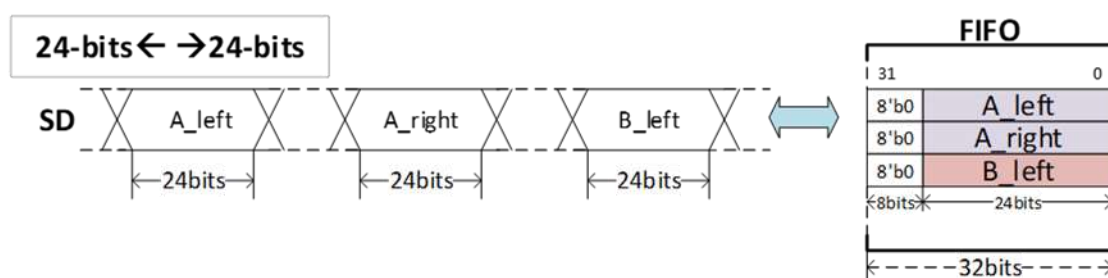
图表 5-20 16 位输入/输出和 FIFO 以 24 位存储数据



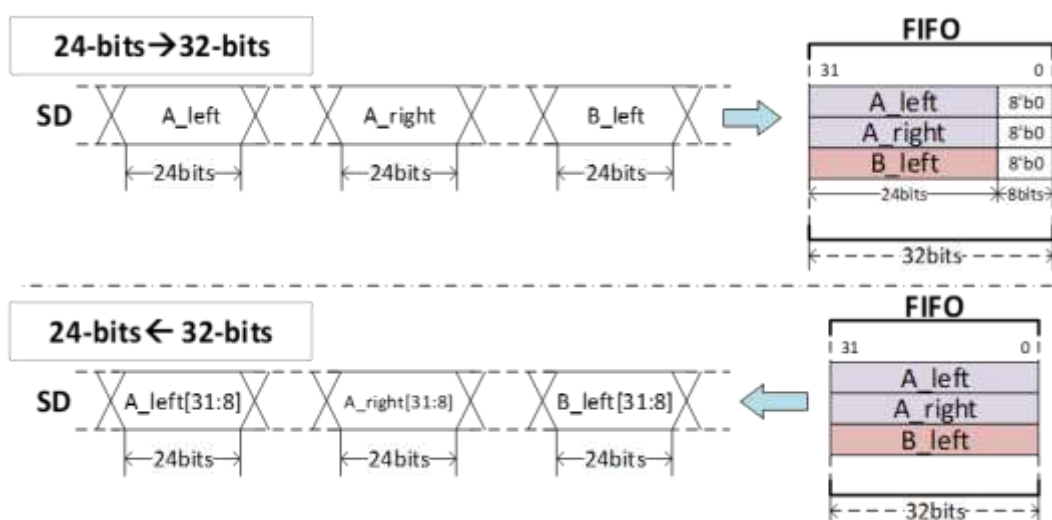
图表 5-21 16 位输入/输出和 FIFO 以 32 位存储数据



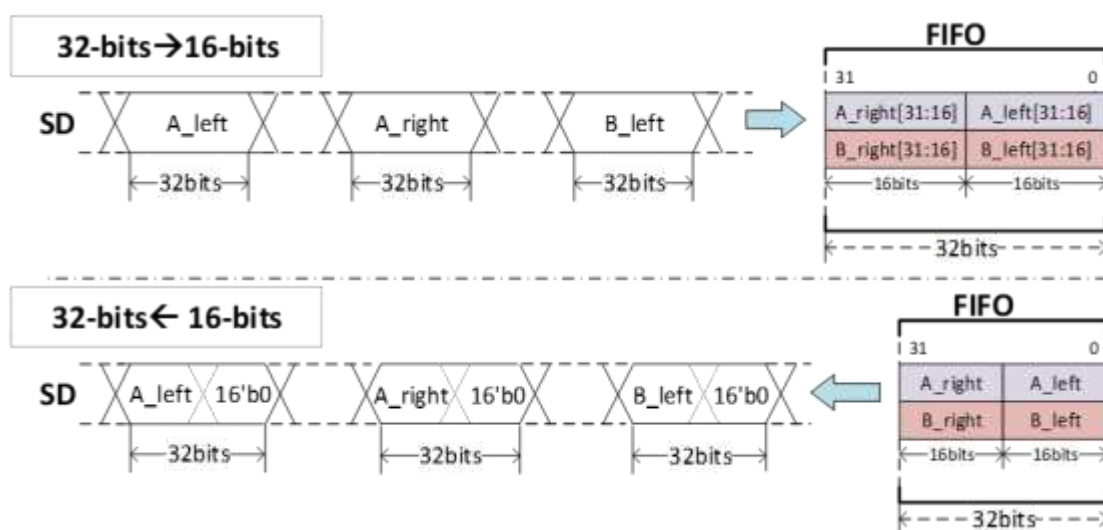
图表 5-22 24 位输入/输出和 FIFO 以 16 位存储数据



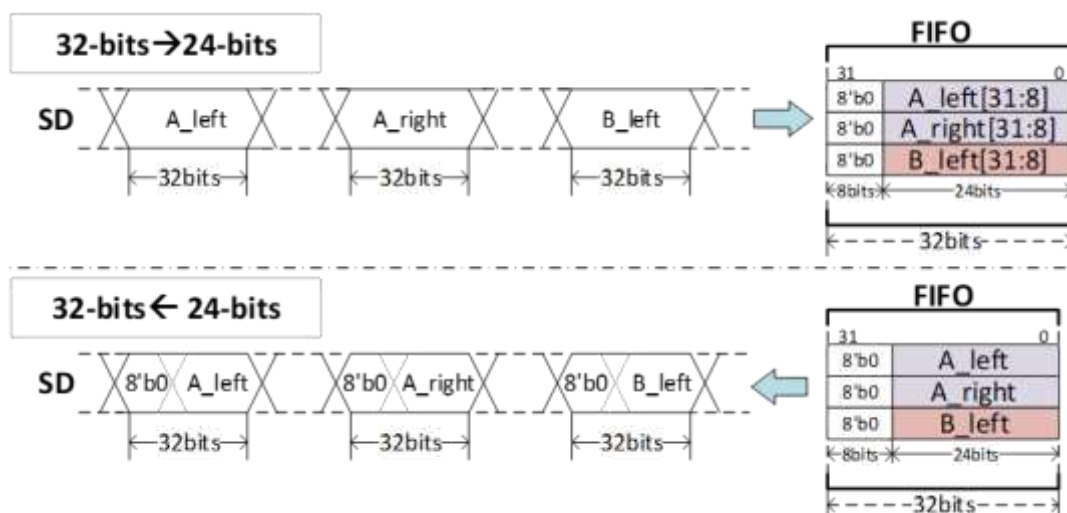
图表 5-23 24 位输入/输出和 FIFO 以 24 位存储数据



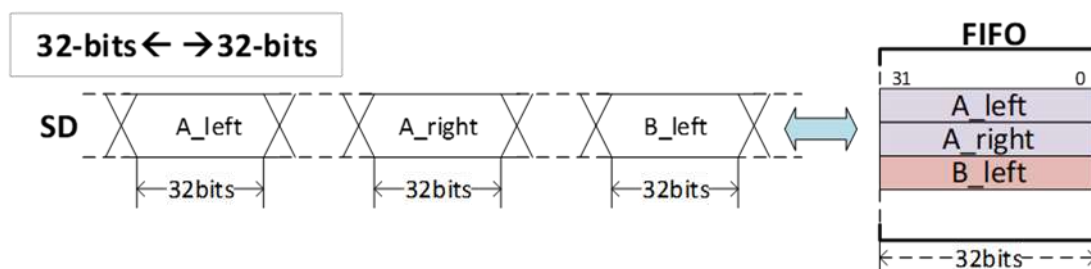
图表 5-24 24 位输入/输出和 FIFO 以 32 位存储数据



图表 5-25 32 位输入/输出和 FIFO 以 16 位存储数据



图表 5-26 32 位输入/输出和 FIFO 以 24 位存储数据



图表 5-27 32 位输入/输出和 FIFO 以 32 位存储数据

输入采样频率的实际值由 AFR 和 ARS 得出：



图表 5-28 实际输入采样频率

AFR	ARS	FS (kHz)
2'b00	2'b00	88.2
	2'b01	44.1
	2'b10	22.05
	2'b11	11.025
2'b01	2'b00	96
	2'b01	48
	2'b10	24
	2'b11	12
2'b10	2'b00	64
	2'b01	32
	2'b10	16
	2'b11	8
2'b11	2'b00	192
	2'b10	error
	2'b11	default

### 5.6.2.5 IISCNF\_OUT

- 寄存器名：I2S 发送器接口格式寄存器
- 描述：该寄存器控制 I2S 输出接口格式。I2S 使能时无法写入该寄存器。
- 偏移量：0x10

图表 5-29 IISCNF\_OUT 字段说明

位段	名称	类型	说明
31:6	保留，读为零。		
5	TX_CLK_SEL	R/W	I2S 从设备模式下发送源时钟选择 0: src_clk (默认) 1: mclk_i
4	I2S_TXMODE	R/W	TX 工作模式选择信号 0: 芯片为主。(默认) 1: 芯片为从。
3	TX_VOICE_EN	R/W	采样源类型选择 0: 源为立体声，具有不同的左右声道信号。(默认) 1: 源为单声道，左右声道数据相同，从发送 FIFO 中取一个数据扩展为立体声。

位段	名称	类型	说明
2	TALOLRC	R/W	左/右声道的有效电平 0: 左声道为低电平。(默认) 1: 左声道为高电平。
1:0	TSAFS	R/W	串行音频格式选择 0x0: I2S (默认) 0x1: 右对齐 0x2: 左对齐

### 5.6.2.6 FADTLR

- 寄存器名: I2S FS 自动检测阈值水平
- 描述: 该寄存器反映了 I2S FS 自动检测阈值水平, 它控制在接收模式下对输入 I2S 音频采样频率的判断。3.072MHz 和 4.032MHz 之间的快速参考时钟用于计算输入 I2S 音频  $f_s$  ( $s_{ws}$ ) 的周期。通过计数值可以判断输入的 I2S 音频采样频率。使能 I2S 时不能写入该寄存器。
- 偏移量: 0x14

图表 5-30 FADTLR 字段说明

位段	名称	类型	说明
31:29	保留, 读为零。		
28:24	192FTR	R/W	192KHz 采样频率阈值寄存器 这些位设置 192kHz 采样频率的中心计数。 $192FTR = ref\_clk / 192k$ 如果参考频率时钟频率为 3.072MHz, 因为 $192K * 16 = 3.072M$ , 通常这个寄存器应该设置为 0x10。当计数值在[14, 18]范围内时, 输入 I2S 音频采样频率被视为 192kHz。
23:22	保留, 读为零。		
21:16	64FTR	R/W	64K 采样频率阈值寄存器 这些位设置 64kHz、32kHz、16kHz 和 8kHz 采样频率的中心计数。 $64FTR = ref\_clk / 64k$ 如果参考频率时钟频率为 3.072MHz, 因为 $64K * 48 = 3.072M$ , 通常这个寄存器应该设置为 0x30。当计数值在[46, 50]范围内时, 输入 I2S 音频采样频率被视为 $(64 * ARS)kHz$ 。

位段	名称	类型	说明
15:14	保留，读为零。		
13:8	88FTR	R/W	88.2K 采样频率阈值寄存器 这些位设置 88.2kHz、44.1kHz、22.05kHz 和 11.025kHz 采样频率的中心计数。 $88FTR = \text{ref\_clk}/88.2k$ 如果参考频率时钟频率为 3.072MHz，因为 $88.2K \times 35 = 3.072M$ ，通常这个寄存器应该设置为 0x23。当计数值在[33, 37]范围内时，输入 I2S 音频采样频率被视为 $(88.2 \times ARS)kHz$ 。
7:6	保留，读为零。		
5:0	96FTR	R/W	96K 采样频率阈值寄存器 这些位设置 96kHz、48kHz、24kHz 和 12kHz 采样频率的中心计数。 $96FTR = \text{ref\_clk}/96k$ 如果参考频率时钟频率为 3.072MHz，因为 $96K \times 32 = 3.072M$ ，通常这个寄存器应该设置为 0x20。当计数值在[30, 34]范围内时，输入 I2S 音频采样频率被视为 $(96 \times ARS)kHz$ 。

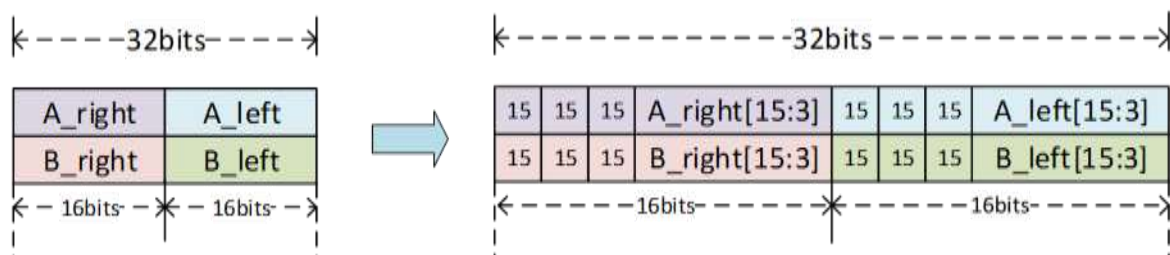
### 5.6.2.7 SCCR

- 寄存器名：I2S 采样压缩控制寄存器
- 描述：该寄存器控制音频采样数据的压缩。
- 偏移量：0x18

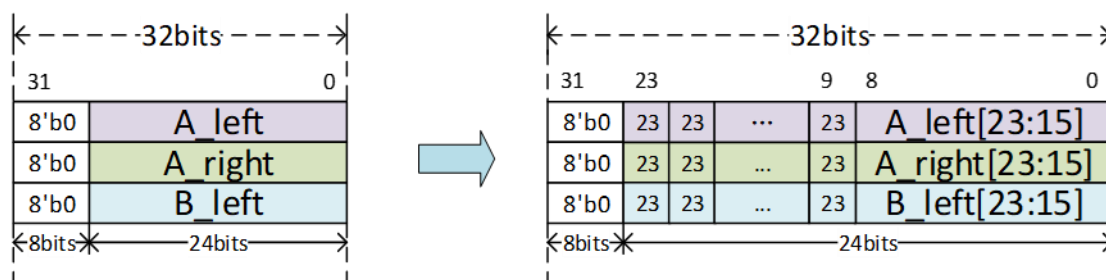
图表 5-31 SCCR 字段说明

位段	名称	类型	说明
31:13	保留，读为零。		
12:8	TVCCR	R/W	TX 音量压缩控制寄存器 根据这个寄存器值，将输出的采样数据右移，左边的空白位用原来的 MSB 填充，也就是 bit15 或 bit23 位或 bit31，这取决于数据宽度（16/24/32 位）。 0：不压缩 1：将样本长度从 16/24/32 位移位到 15/23/31 位。 2：将样本长度从 16/24/32 位移位到 14/22/30 位。 ...

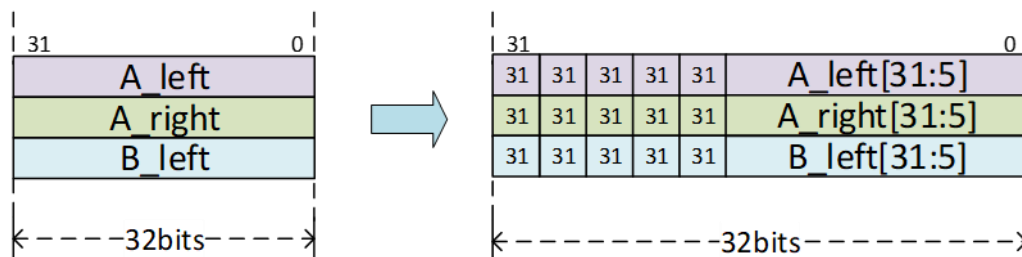
位段	名称	类型	说明
			15: 将样本长度从 16/24/32 位移位到 1/9/17 位。 其他: 将样本长度从 16/24/32 位移位到 0/8/16 位。 见图表 5-32、图表 5-33、图表 5-34。
7	保留, 读为零。		
6:5	SSRCR	R/W	RX 下采样率压缩控制寄存器 根据该寄存器值, 压缩输入的采样数据。 0: 不压缩 1: 压缩采样数据一次, 即丢弃每两个样本中的一个, 接收两个样本中的第一个数据。 2: 压缩采样数据两次, 即丢弃每三个样本中的两个, 接收三个样本中的第一个数据。 3: 不压缩 见图表 5-35、图表 5-36、图表 5-37、图表 5-38、图表 5-39、图表 5-40。
4:0	RVCCR	R/W	RX 音量压缩控制寄存器 根据这个寄存器值, 将输入的采样数据右移, 左边的空白位用原来的 MSB 填充, 也就是 bit15 位或 bit23 位或 bit31, 这取决于数据宽度 (16/24/32 位)。 0: 不压缩 1: 将样本数据从 16/24/32 位移位到 15/23/31 位。 2: 将样本数据从 16/24/32 位移位到 14/22/30 位。 ... 15: 将样本数据从 16/24/32 位移位到 1/9/17 位。 其他: 将样本数据从 16/24/32 位移位到 0/8/16 位。 见图表 5-32、图表 5-33、图表 5-34。



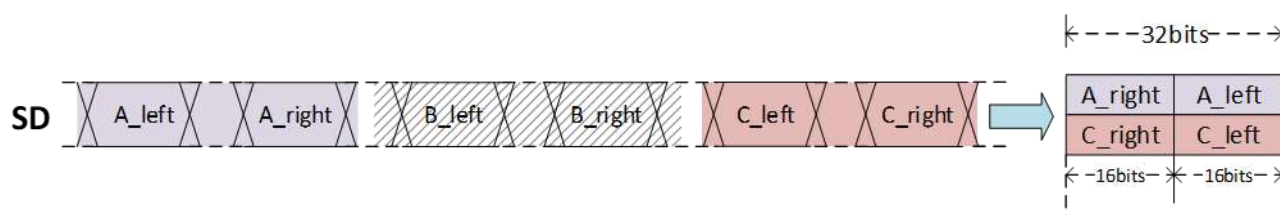
图表 5-32 数据压缩@T/RVCCR = 3, DATAWTH[1:0] = 00



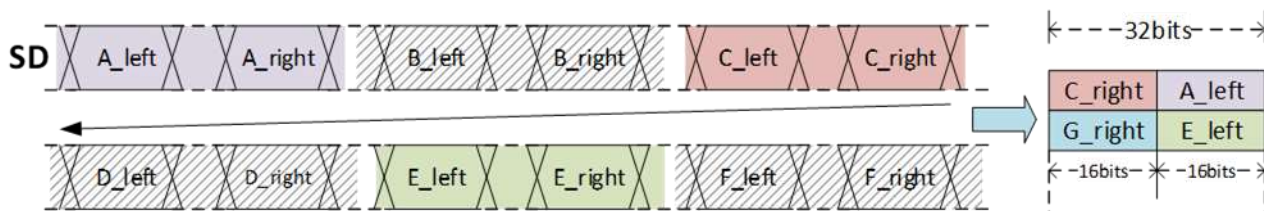
图表 5-33 数据压缩@T/RVCCR = 15, DATAWTH[1:0] = 01



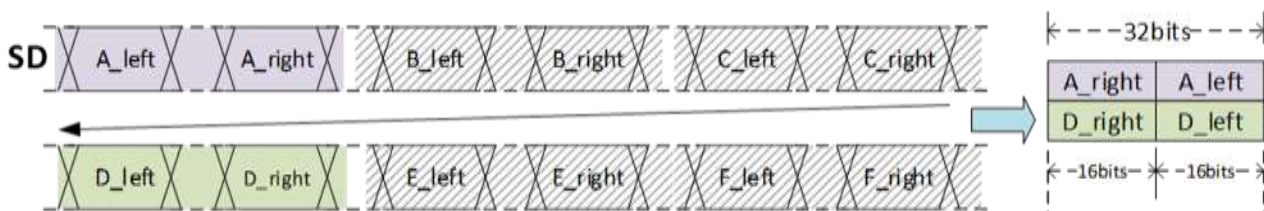
图表 5-34 数据压缩@T/RVCCR=5, DATAWTH[1:0]=1?



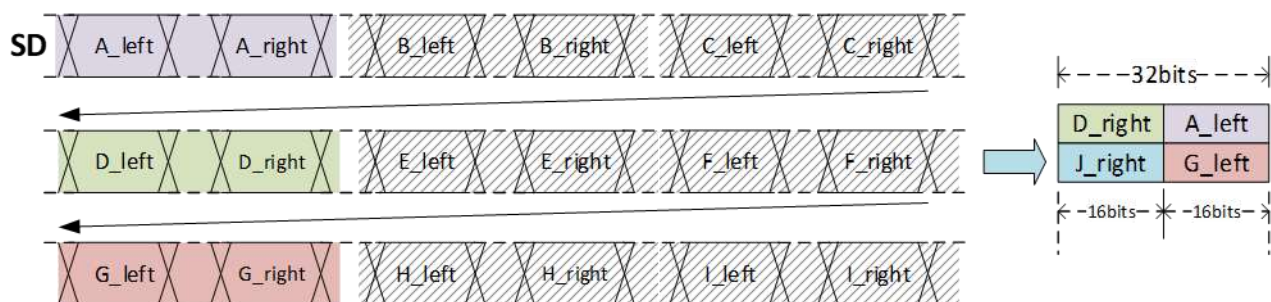
图表 5-35 数据压缩@ = 1, DATAWTH[1:0]=00@立体声源



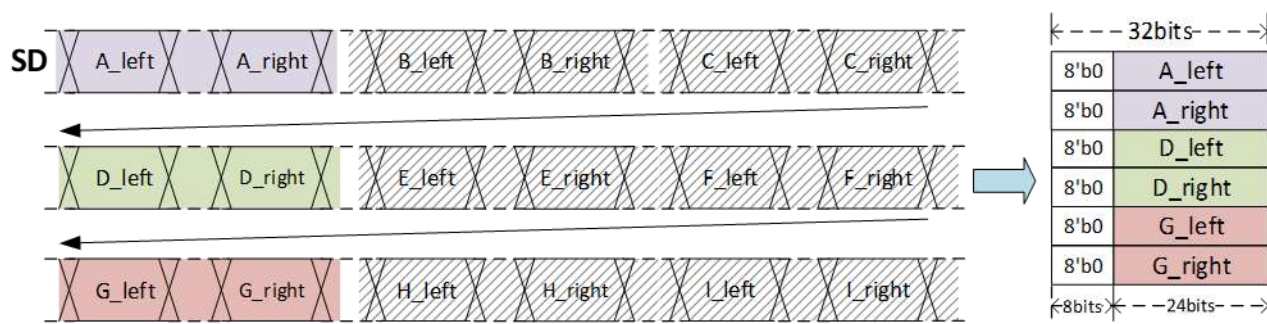
图表 5-36 数据压缩@SSR=1, DATAWTH[1:0]=00@单声道源



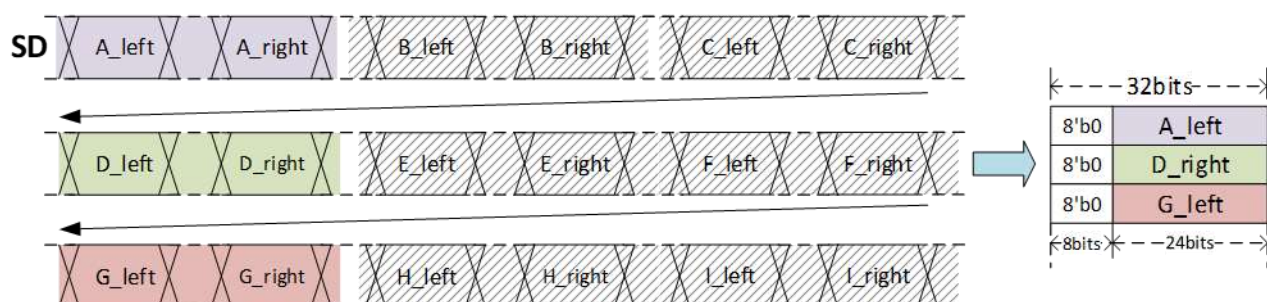
图表 5-37 数据压缩@SSR=2, DATAWTH[1:0]=00@立体声源



图表 5-38 数据压缩@SSRCR=2, DATAWTH[1:0]=00@单声道源



图表 5-39 数据压缩@SSRCR=2, DATAWTH[1:0]=01@立体声源



图表 5-40 数据压缩@SSRCR=2, DATAWTH[1:0]=01@单声道源

## 5.6.2.8 TXFTLR

- 寄存器名：I2S 发送 FIFO 阈值寄存器
- 描述：该寄存器控制发送 FIFO 的阈值。使能 I2S（或 SPDIF）时不能写入该寄存器。
- 偏移量：0x1C

图表 5-41 TXFTLR 字段说明

位段	名称	类型	说明
31:5	保留，只读。		

位段	名称	类型	说明
4:0	TFT	R/W	发送 FIFO 阈值 控制发送 FIFO 控制器触发中断的条目水平（或更低）。 FIFO 阈值可在 0-31 范围内配置。默认值为 16。

### 5.6.2.9 RXFTLR

- 寄存器名：I2S 接收 FIFO 阈值寄存器
- 描述：该寄存器控制接收 FIFO 的阈值。使能 I2S（或 SPDIF）时不能写入该寄存器。
- 偏移量：0x20

图表 5-42 RXFTLR 字段说明

位段	名称	类型	说明
31:5	保留，读为零。		
4:0	RFT	R/W	接收 FIFO 阈值 控制接收 FIFO 控制器触发中断的条目水平（或更高）。 FIFO 阈值可在 1-31 范围内配置。设置为 0 时，阈值为 32。默认值为 16。

### 5.6.2.10 TXFLR

- 寄存器名：I2S 发送 FIFO 水平寄存器
- 描述：该寄存器包含发送 FIFO 中有效数据条目的数量。
- 偏移量：0x24

图表 5-43 TXFLR 字段说明

位段	名称	类型	说明
31:6	保留，读为零。		
5:0	TXTFL	R	发送 FIFO 水平 包含发送 FIFO 中当前有效数据条目的数量。

### 5.6.2.11 RXFLR

- 寄存器名：I2S 接收 FIFO 水平寄存器
- 描述：该寄存器包含接收 FIFO 中有效数据条目的数量。
- 偏移量：0x28

图表 5-44 RXFLR 字段说明

位段	名称	类型	说明
31:6	保留，读为零。		
5:0	RXTFL	R	接收 FIFO 水平 包含接收 FIFO 中当前有效数据条目的数量。

## 5.6.2.12 SR

- 寄存器名：I2S 状态寄存器
- 描述：这是一个只读寄存器，用于指示当前传输状态、FIFO 状态以及发生的任何发送/接收错误。
- 偏移量：0x2C

图表 5-45 SR 字段说明

位段	名称	类型	说明
31:6	保留，读为零。		
5	RFF	R	接收 FIFO 满 当接收 FIFO 完全填满时，该位被设置。当接收 FIFO 包含一个或多个空位置时，该位被清除。 0：接收 FIFO 未滿。 1：接收 FIFO 已滿。
4	RFNE	R	接收 FIFO 非空 当接收 FIFO 中包含一个或多个条目时设置，当接收 FIFO 为空时清除。 0：接收 FIFO 为空。 1：接收 FIFO 不为空。
3	TFE	R	发送 FIFO 空 当发送 FIFO 完全为空时，该位被设置。当发送 FIFO 中包含一个或多个有效条目时，该位被清除。 0：发送 FIFO 不为空。 1：发送 FIFO 为空。



位段	名称	类型	说明
2	TFNF	R	发送 FIFO 未 满 当发送 FIFO 包含一个或多个空条目时设置，当发送 FIFO 满时清除。 0：发送 FIFO 已满。 1：发送 FIFO 未。
1	TX_BUSY	R	I2S TX 忙标志 0：I2S TX 空闲或禁用。 1：I2S TX 正在传输数据。
0	RX_BUSY	R	I2S RX 忙标志 0：I2S RX 空闲或禁用。 1：I2S RX 正在接收数据。

### 5.6.2.13 IMR

- 寄存器名：I2S 中断屏蔽寄存器
- 描述：该读/写寄存器屏蔽或启用由 I2S 生成的所有中断。
- 偏移量：0x30

图表 5-46 IMR 字段说明

位段	名称	类型	说明
31	IFSCM	R/W	输入采样频率变化屏蔽。 0：in_fsc_intr 中断被屏蔽。（在自动检测模式下）（默认） 1：in_fsc_intr 中断未被屏蔽。（在自动检测模式下）
30:7	保留，读为零。		
6	RXFIM	R/W	接收 FIFO 阈值满中断屏蔽 0：i2s_rxf_intr 中断被屏蔽。 1：i2s_rxf_intr 中断未被屏蔽。（默认）
5	TXEIM	R/W	发送 FIFO 阈值空中断屏蔽 0：i2s_txe_intr 中断被屏蔽。 1：i2s_txe_intr 中断未被屏蔽。（默认）

位段	名称	类型	说明
4	RXOIM	R/W	接收 FIFO 溢出错误屏蔽 0: i2s_rxo_err 错误被屏蔽。 1: i2s_rxo_err 错误未被屏蔽。（默认）
3	RXUIM	R/W	接收 FIFO 下溢错误屏蔽 0: i2s_rxu_err 错误被屏蔽。 1: i2s_rxu_err 错误未被屏蔽。（默认）
2	TXOIM	R/W	发送 FIFO 溢出错误屏蔽 0: i2s_txo_err 错误被屏蔽。 1: i2s_txo_err 错误未被屏蔽。（默认）
1	TXUIM	R/W	发送 FIFO 下溢错误屏蔽 0: i2s_txu_err 错误被屏蔽。 1: i2s_txu_err 错误未被屏蔽。（默认）
0	WADEM	R/W	I2S 错误地址错误屏蔽 0: i2s_waddr_err 错误被屏蔽。 1: i2s_waddr_err 错误未被屏蔽。（默认）

### 5.6.2.14 ISR

- 寄存器名: I2S 中断状态寄存器
- 描述: 该寄存器指示 I2S 中断被屏蔽后的状态。
- 偏移量: 0x34

图表 5-47 ISR 字段说明

位段	名称	类型	说明
31	IFSCS	R	屏蔽后的输入采样频率变化中断状态 0: in_fsc_intr 中断已激活。（在自动检测模式下） 1: in_fsc_intr 中断未激活。（在自动检测模式下）
30:7	保留，读为零。		
6	RXFIS	R	屏蔽后的接收 FIFO 阈值满中断状态 0: i2s_rxf_intr 中断未激活。 1: i2s_rxf_intr 中断已激活。

位段	名称	类型	说明
5	TXEIS	R	屏蔽后的发送 FIFO 阈值空中断状态 0: i2s_txe_intr 中断未激活。 1: i2s_txe_intr 中断已激活。
4	RXOIS	R	屏蔽后的接收 FIFO 溢出错误状态 0: i2s_rxo_err 错误未激活。 1: i2s_rxo_err 错误已激活。
3	RXUIS	R	屏蔽后的接收 FIFO 下溢错误状态 0: i2s_rxu_err 错误未激活。 1: i2s_rxu_err 错误已激活。
2	TXOIS	R	屏蔽后的发送 FIFO 溢出错误状态 0: i2s_txo_err 错误未激活。 1: i2s_txo_err 错误已激活。
1	TXUIRS	R	屏蔽后的发送 FIFO 下溢错误状态 0: i2s_txu_err 错误未激活。 1: i2s_txu_err 错误已激活。
0	WADES	R	屏蔽后的 I2S 错误地址错误状态 0: i2s_waddr_err 错误未激活。 1: i2s_waddr_err 错误已激活。

### 5.6.2.15 RISR

- 寄存器名: I2S 原始中断状态寄存器
- 描述: 该只读寄存器报告 I2S 中断在屏蔽前的状态。
- 偏移量: 0x38

图表 5-48 RISR 字段说明

位段	名称	类型	说明
31	RIFSCS	R	屏蔽前的输入采样频率变化原始中断状态 0: in_fsc_intr 中断已激活。(在自动检测模式下) 1: in_fsc_intr 中断未激活。(在自动检测模式下)
30:7	保留, 读为零。		

位段	名称	类型	说明
6	RXFIR	R	屏蔽前的接收 FIFO 阈值满原始中断状态 0: i2s_rxf_intr 中断未激活。 1: i2s_rxf_intr 中断已激活。
5	TXEIR	R	屏蔽前的发送 FIFO 阈值空原始中断状态 0: i2s_txe_intr 中断未激活。 1: i2s_txe_intr 中断已激活。
4	RXOIR	R	屏蔽前的接收 FIFO 溢出原始错误状态 0: i2s_rxo_err 错误未激活。 1: i2s_rxo_err 错误已激活。
3	RXUIR	R	屏蔽前的接收 FIFO 下溢原始错误状态 0: i2s_rxu_err 错误未激活。 1: i2s_rxu_err 错误已激活。
2	TXOIR	R	屏蔽前的发送 FIFO 溢出原始错误状态 0: i2s_txo_err 错误未激活。 1: i2s_txo_err 错误已激活。
1	TXUIR	R	屏蔽前的发送 FIFO 下溢原始错误状态 0: i2s_txu_err 错误未激活。 1: i2s_txu_err 错误已激活。
0	RWADES	R	屏蔽前的 I2S 错误地址原始错误状态 0: i2s_waddr_err 错误未激活。 1: i2s_waddr_err 错误已激活。

### 5.6.2.16 ICR

- 寄存器名: I2S 中断清除寄存器
- 描述: 该只写寄存器用于清除 I2S 中断。
- 偏移量: 0x3C

图表 5-49 ICR 字段说明

位段	名称	类型	说明
31	CRIFSC	W	清除输入采样频率变化中断 0: 不清除 in_fsc_intr 中断。 1: 清除 in_fsc_intr 中断。
30:7	保留，读为零。		
6	RXFIC	W	清除接收 FIFO 阈值满中断 0: 不清除 i2s_rxf_intr 中断。 1: 清除 i2s_rxf_intr 中断。
5	TXEIC	W	清除发送 FIFO 阈值空中断 0: 不清除 i2s_txe_intr 中断。 1: 清除 i2s_txe_intr 中断。
4	RXOIC	W	清除接收 FIFO 溢出错误 0: 不清除 i2s_rxo_err 错误。 1: 清除 i2s_rxo_err 错误。
3	RXUIC	W	清除接收 FIFO 下溢错误 0: 不清除 i2s_rxu_err 中断。 1: 清除 i2s_rxu_err 中断。
2	TXOIC	W	清除发送 FIFO 溢出错误状态 0: 不清除 i2s_txo_err 中断。 1: 清除 i2s_txo_err 中断。
1	TXUIC	W	清除发送 FIFO 下溢错误 0: 不清除 i2s_txu_err 中断。 1: 清除 i2s_txu_err 中断。
0	CWADEC	W	清除 I2S 错误地址错误 0: 不清除 i2s_waddr_err 中断。 1: 清除 i2s_waddr_err 中断。

### 5.6.2.17 DMACR

- 寄存器名：I2S DMA 控制寄存器
- 描述：该寄存器用于使能 DMA 控制器接口操作。

- 偏移量：0x40

图表 5-50 DMACR 字段说明

位段	名称	类型	说明
31:2	保留，读为零。		
1	TDMAE	R/W	发送 DMA 使能 该位使能/禁用发送 FIFO DMA 通道。 0：禁用发送 DMA。（默认） 1：使能发送 DMA
0	RDMAE	R/W	接收 DMA 使能 该位使能/禁用接收 FIFO DMA 通道。 0：禁用接收 DMA。（默认） 1：使能接收 DMA

### 5.6.2.18 DMATDLR

- 寄存器名：I2S DMA 发送数据水平寄存器
- 描述：该寄存器控制 I2S DMA 发送数据水平。
- 偏移量：0x44

图表 5-51 DMATDLR 字段说明

位段	名称	类型	说明
31:5	保留，读为零。		
4:0	DMATDL	R/W	发送数据水平 该位段控制发送逻辑发出 DMA 请求的水平。水线水平 =DMATDL；即当发送 FIFO 中有效数据条目数等于或小于该字段值，并且 TDMAE=1 时，产生 dma_tx_req 信号。 默认值为 16，可在 0-31 范围内配置。当设置为 0 时，阈值 =32。

### 5.6.2.19 DMARDLR

- 寄存器名：I2S DMA 接收数据水平寄存器
- 描述：该寄存器控制 I2S DMA 接收数据水平。
- 偏移量：0x48

图表 5-52 DMARDLR 字段说明

位段	名称	类型	说明
31:5	保留，读为零。		
4:0	DMARDL	R/W	DMA 接收数据水平 该位段控制接收逻辑发出 DMA 请求的水平。水线水平 =DMARDL；即当接收 FIFO 中有效数据条目数等于或大于该字段值，并且 RDMAE=1 时，产生 dma_rx_req 信号。 默认值为 0，可在 1-31 范围内配置。当设置为 0 时，阈值 =32。

### 5.6.2.20 DIV0\_LEVEL

- 名称：I2S 分频器 0 控制寄存器
- 描述：作为分频器写入该寄存器，分频 src\_clk，然后得到 mclk。
- 偏移量：0x50

图表 5-53 DIV0\_LEVEL 字段说明

位段	名称	类型	说明
31:8	保留，读为零。		
7:0	DIV0	R/W	用于从 src_clk 获取 mclk 的分频器。 0：不分频（默认） 否则：divide = DIV0

### 5.6.2.21 DIV3\_LEVEL

- 名称：I2S 分频器 3 控制寄存器
- 描述：作为分频器写入该寄存器，分频 src\_clk，然后得到 ref\_clk。
- 偏移量：0x54

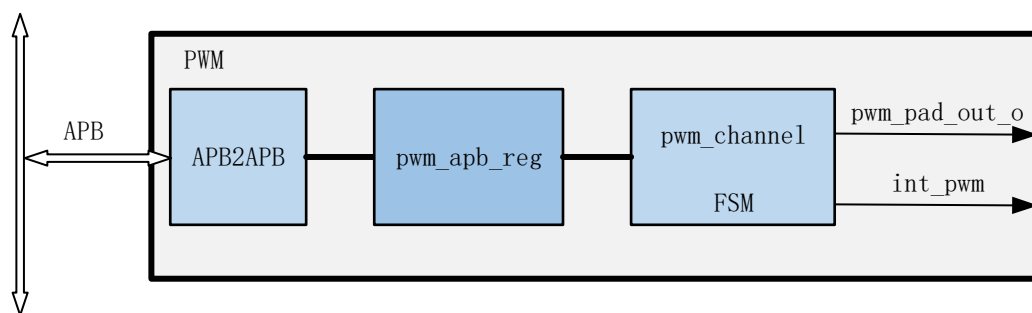
图表 5-54 DIV3\_LEVEL 字段说明

位段	名称	类型	说明
31:8	保留，读为零。		
7:0	DIV3	R/W	用于从 src_clk 获取 ref_clk 的分频器。 $divide = (DIV3 + 1) * 2$

## 6 PWM

### 6.1 概述

脉冲宽度调制（PWM）可以通过产生周期性的脉冲波形来控制电机和其他设备。脉冲宽度和周期数可通过寄存器进行配置。



图表 6-1 PWM 模块功能图

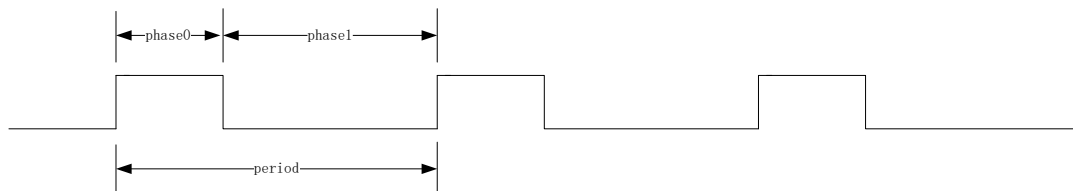
### 6.2 主要特性

PWM 模块由 6 个 PWM 发生器组成，每个发生器与 1 个 IO 引脚相关，每个发生器具有如下特性：

- 周期频率可配置，最大支持 12MHz
- 占空比可配置
- 支持单次模式和连续模式
- 支持单次模式下的事件触发模式

### 6.3 接口

PWM 输出时序图如图表 6-2 所示，可以配置周期和高电平相位长度。



图表 6-2 PWM 接口时序图



PWM 的 PAD 如下图表 6-3 所述：

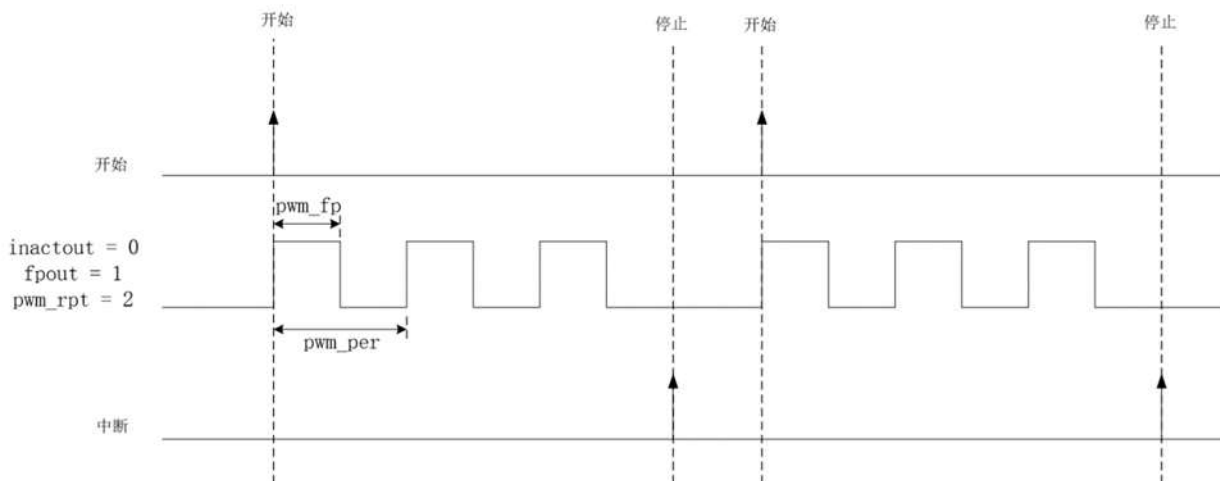
图表 6-3 引脚描述

引脚名称	方向	宽度	说明
PWM0	O	1	PWM 通道 0 输出的脉冲波形数据
PWM1	O	1	PWM 通道 1 输出的脉冲波形数据
PWM2	O	1	PWM 通道 2 输出的脉冲波形数据
PWM3	O	1	PWM 通道 3 输出的脉冲波形数据
PWM4	O	1	PWM 通道 4 输出的脉冲波形数据
PWM5	O	1	PWM 通道 5 输出的脉冲波形数据

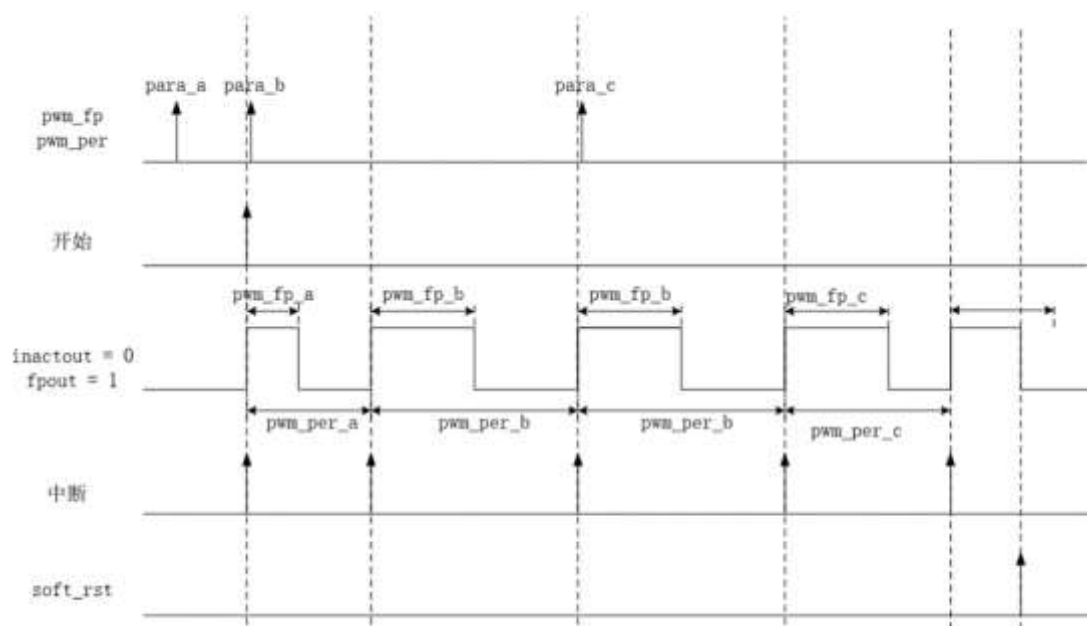
## 6.4 功能描述

### 6.4.1 数据和控制流程

PWM 脉冲生成模式分为单次模式和连续模式。单次模式的脉冲周期数可以通过寄存器 PWM\_RPT 进行配置。连续模式可以在流程中更新配置信息。



图表 6-4 单次模式图



图表 6-5 连续模式图

## 6.4.2 PWM 中断

PWM 提供一个中断信号 `int_pwm`。详见芯片的中断向量表。

## 6.5 使用

### 6.5.1 单次模式

单次模式配置如下：

1. 配置寄存器 `PWM_CTRL` 的相关信息，但 `bit0[start]` 和 `bit2[cfg_update]` 除外。
2. 配置寄存器 `PWM_RPT`、`PWM_PER` 和 `PWM_FP`。
3. 配置寄存器 `PWM_CTRL` 的 `bit2`，写入 1。
4. 配置寄存器 `PWM_CTRL` 的 `bit0`，写入 1。
5. 查看 `PWM_STATUS` 和 `PWM_INTR_STA`。

### 6.5.2 连续模式

连续模式配置如下：

1. 配置寄存器 `PWM_CTR` 的相关信息，但 `bit0[start]` 和 `bit2[cfg_update]` 除外。
2. 配置寄存器 `PWM_RPT`、`PWM_PER` 和 `PWM_FP`。
3. 配置寄存器 `PWM_CTRL` 的 `bit2`，写入 1。
4. 配置寄存器 `PWM_CTRL` 的 `bit0`，写入 1。
5. 查看 `PWM_STATUS` 和 `PWM_INTR_STA`。

## 6.6 寄存器描述

### 6.6.1 寄存器内存映射

寄存器	偏移量	说明	章节/页码
PWM_CTRL_0	0x00	PWM 通道 0 控制寄存器	<a href="#">6.6.2.1/224</a>
PWM_RPT_0	0x04	PWM 通道 0 单次控制寄存器	<a href="#">6.6.2.2/225</a>
PWM_PER_0	0x08	PWM 通道 0 周期控制寄存器	<a href="#">6.6.2.3/225</a>
PWM_FP_0	0x0c	PWM 通道 0 第一相位控制寄存器	<a href="#">6.6.2.4/226</a>
PWM_STATUS_0	0x10	PWM 通道 0 状态寄存器	<a href="#">6.6.2.5/226</a>
PWM_CTRL_1	0x20	PWM 通道 1 控制寄存器	<a href="#">6.6.2.6/227</a>
PWM_RPT_1	0x24	PWM 通道 1 单次控制寄存器	<a href="#">6.6.2.7/228</a>
PWM_PER_1	0x28	PWM 通道 1 周期控制寄存器	<a href="#">6.6.2.8/228</a>
PWM_FP_1	0x2c	PWM 通道 1 第一相位控制寄存器	<a href="#">6.6.2.9/228</a>
PWM_STATUS_1	0x30	PWM 通道 1 状态寄存器	<a href="#">6.6.2.10/229</a>
PWM_CTRL_2	0x40	PWM 通道 2 控制寄存器	<a href="#">6.6.2.11/229</a>
PWM_RPT_2	0x44	PWM 通道 2 单次控制寄存器	<a href="#">6.6.2.12/230</a>
PWM_PER_2	0x48	PWM 通道 2 周期控制寄存器	<a href="#">6.6.2.13/230</a>
PWM_FP_2	0x4c	PWM 通道 2 第一相位控制寄存器	<a href="#">6.6.2.14/231</a>
PWM_STATUS_2	0x50	PWM 通道 2 状态寄存器	<a href="#">6.6.2.15/231</a>
PWM_CTRL_3	0x60	PWM 通道 3 控制寄存器	<a href="#">6.6.2.16/232</a>
PWM_RPT_3	0x64	PWM 通道 3 单次控制寄存器	<a href="#">6.6.2.17/233</a>
PWM_PER_3	0x68	PWM 通道 3 周期控制寄存器	<a href="#">6.6.2.18/233</a>
PWM_FP_3	0x6c	PWM 通道 3 第一相位控制寄存器	<a href="#">6.6.2.19/233</a>
PWM_STATUS_3	0x70	PWM 通道 3 状态寄存器	<a href="#">6.6.2.20/233</a>
PWM_CTRL_4	0x80	PWM 通道 4 控制寄存器	<a href="#">6.6.2.21/234</a>
PWM_RPT_4	0x84	PWM 通道 4 单次控制寄存器	<a href="#">6.6.2.22/235</a>

寄存器	偏移量	说明	章节/页码
PWM_PER_4	0x88	PWM 通道 4 周期控制寄存器	<a href="#">6.6.2.23/235</a>
PWM_FP_4	0x8c	PWM 通道 4 第一相位控制寄存器	<a href="#">6.6.2.24/236</a>
PWM_STATUS_4	0x90	PWM 通道 4 状态寄存器	<a href="#">6.6.2.25/236</a>
PWM_CTRL_5	0xa0	PWM 通道 5 控制寄存器	<a href="#">6.6.2.26/236</a>
PWM_RPT_5	0xa4	PWM 通道 5 单次控制寄存器	<a href="#">6.6.2.27/238</a>
PWM_PER_5	0xa8	PWM 通道 5 周期控制寄存器	<a href="#">6.6.2.28/238</a>
PWM_FP_5	0xac	PWM 通道 5 第一相位控制寄存器	<a href="#">6.6.2.29/238</a>
PWM_STATUS_5	0xb0	PWM 通道 5 状态寄存器	<a href="#">6.6.2.30/238</a>
PWM_INTR_STA	0xff0	PWM 中断状态寄存器	<a href="#">6.6.2.31/239</a>

## 6.6.2 寄存器和字段描述

### 6.6.2.1 PWM\_CTRL\_0

- 描述：PWM 通道 0 控制寄存器
- 偏移量：0x00
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	INACTOUT	RW	未启用时的 PWM 输出电平 复位值：0x0
[8]	FPOUT	RW	一个周期内第一相位的 PWM 输出电平 复位值：0x0
[7:6]	EVTRIG	RW	单次模式下的事件触发模式 1：上升沿触发 2：下降沿触发 0&3：通用模式，即 PWM 信号的输出不基于外部信号。 复位值：0x0
[5:4]	MODE	RW	PWM 模式

位段	名称	类型	说明
			1: 单次模式 2: 连续模式 0、3: 保留 复位值: 0x0
[3]	INTEN	RW	中断使能 0: 禁用 1: 使能 复位值: 0x0
[2]	CFG_UPDATE	RW	在连续模式下, PWM_PER 和 PWM_FP 寄存器在中间更新。 当寄存器的值发生变化时, cfg_update 将从 0 配置为 1 (表示两个寄存器值同时更新)。 复位值: 0x0
[1]	SOFT_RST	RW	软件复位使能用于随时关闭 PWM_cycle (PWM 状态[7:0])。 复位值: 0x0
[0]	START	RW	PWM 启动编码使能, 上升沿有效 (即从 0 到 1)。 复位值: 0x0

### 6.6.2.2 PWM\_RPT\_0

- 描述: PWM 通道 0 单次控制寄存器
- 偏移量: 0x04
- 默认值: 0x0

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ONE_SHOT_RPT	RW	单次传输中重复的周期数 复位值: 0x0

### 6.6.2.3 PWM\_PER\_0

- 描述: PWM 通道 0 周期控制寄存器
- 偏移量: 0x08
- 默认值: 0x0

位段	名称	类型	说明
[31:0]	PWM_PER	RW	周期波形持续的系统时钟拍数。如果 WM_per 为 0，不输出 PWM 有效信号。 复位值：0x0

#### 6.6.2.4 PWM\_FP\_0

- 描述：PWM 通道 0 第一相位控制寄存器
- 偏移量：0x0c
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_FP	RW	周期波形第一相位的拍数。如果 PWM_FP 大于 PWM_per，所有有效信号均为 FPOUT。如果 PWM FP 为 0，所有有效信号均为! FPOUT。 复位值：0x0

#### 6.6.2.5 PWM\_STATUS\_0

- 描述：PWM 通道 0 状态寄存器
- 偏移量：0x10
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	PWM_OUT	RW	波形的 PWM 电流值 复位值：0x0
[8]	PWM_BUSY	RW	状态机的 PWM 状态 0：空闲 1：忙 复位值：0x0
[7:0]	PWM_CYCLE	RW	在连续模式下，同一组 PWM_per 和 PWM_FP 参数输出的周期数 复位值：0x0

## 6.6.2.6 PWM\_CTRL\_1

- 描述：PWM 通道 1 控制寄存器
- 偏移量：0x20
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	INACTOUT	RW	未启用时的 PWM 输出电平 复位值：0x0
[8]	FPOUT	RW	一个周期内第一相位的 PWM 输出电平 复位值：0x0
[7:6]	EVTRIG	RW	单次模式下的事件触发模式 1：上升沿触发 2：下降沿触发 0&3：通用模式，即 PWM 信号的输出不基于外部信号。 复位值：0x0
[5:4]	MODE	RW	PWM 模式 1：单次模式 2：连续模式 0、3：保留 复位值：0x0
[3]	INTEN	RW	中断使能 0：禁用 1：使能 复位值：0x0
[2]	CFG_UPDATE	RW	在连续模式下，PWM_PER 和 PWM_FP 寄存器在中间更新。 当寄存器的值发生变化时，cfg_update 将从 0 配置为 1（表示两个寄存器值同时更新）。 复位值：0x0
[1]	SOFT_RST	RW	软件复位使能用于随时关闭 PWM_cycle（PWM 状态[7:0]）。 复位值：0x0

位段	名称	类型	说明
[0]	START	RW	PWM 启动编码使能，上升沿有效（即从 0 到 1）。 复位值：0x0

### 6.6.2.7 PWM\_RPT\_1

- 描述：PWM 通道 1 单次控制寄存器
- 偏移量：0x24
- 默认值：0x0

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ONE_SHOT_RPT	RW	单次传输中重复的周期数 复位值：0x0

### 6.6.2.8 PWM\_PER\_1

- 描述：PWM 通道 1 周期控制寄存器
- 偏移量：0x28
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_PER	RW	周期波形持续的系统时钟拍数。如果 WM_per 为 0，不输出 PWM 有效信号。 复位值：0x0

### 6.6.2.9 PWM\_FP\_1

- 描述：PWM 通道 1 第一相位控制寄存器
- 偏移量：0x2c
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_FP	RW	周期波形第一相位的拍数。如果 PWM_FP 大于 PWM_per，所有有效信号均为 FPOUT。如果 PWM FP 为 0，所有有效信号均为! FPOUT。 复位值：0x0



### 6.6.2.10 PWM\_STATUS\_1

- 描述：PWM 通道 1 状态寄存器
- 偏移量：0x30

默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	PWM_OUT	RW	波形的 PWM 电流值 复位值：0x0
[8]	PWM_BUSY	RW	状态机的 PWM 状态 0：空闲 1：忙 复位值：0x0
[7:0]	PWM_CYCLE	RW	在连续模式下，同一组 PWM_per 和 PWM_FP 参数输出的周期数 复位值：0x0

### 6.6.2.11 PWM\_CTRL\_2

- 描述：PWM 通道 2 控制寄存器
- 偏移量：0x40
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	INACTOUT	RW	未启用时的 PWM 输出电平 复位值：0x0
[8]	FPOUT	RW	一个周期内第一相位的 PWM 输出电平 复位值：0x0
[7:6]	EVTRIG	RW	单次模式下的事件触发模式 1：上升沿触发 2：下降沿触发 0&3：通用模式，即 PWM 信号的输出不基于外部信号。 复位值：0x0

位段	名称	类型	说明
[5:4]	MODE	RW	PWM 模式 1: 单次模式 2: 连续模式 0、3: 保留 复位值: 0x0
[3]	INTEN	RW	中断使能 0: 禁用 1: 使能 复位值: 0x0
[2]	CFG_UPDATE	RW	在连续模式下, PWM_PER 和 PWM_FP 寄存器在中间更新。 当寄存器的值发生变化时, cfg_update 将从 0 配置为 1 (表示两个寄存器值同时更新)。 复位值: 0x0
[1]	SOFT_RST	RW	软件复位使能用于随时关闭 PWM_cycle (PWM 状态[7:0])。 复位值: 0x0
[0]	START	RW	PWM 启动编码使能, 上升沿有效 (即从 0 到 1)。 复位值: 0x0

### 6.6.2.12 PWM\_RPT\_2

- 描述: PWM 通道 2 单次控制寄存器
- 偏移量: 0x44
- 默认值: 0x0

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ONE_SHOT_RPT	RW	单次传输中重复的周期数 复位值: 0x0

### 6.6.2.13 PWM\_PER\_2

- 描述: PWM 通道 2 周期控制寄存器
- 偏移量: 0x48

- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_PER	RW	周期波形持续的系统时钟拍数。如果 WM_per 为 0，不输出 PWM 有效信号。 复位值：0x0

### 6.6.2.14 PWM\_FP\_2

- 描述：PWM 通道 2 第一相位控制寄存器
- 偏移量：0x4c
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_FP	RW	周期波形第一相位的拍数。如果 PWM_FP 大于 PWM_per，所有有效信号均为 FPOUT。如果 PWM FP 为 0，所有有效信号均为! FPOUT。 复位值：0x0

### 6.6.2.15 PWM\_STATUS\_2

- 描述：PWM 通道 2 状态寄存器
- 偏移量：0x50
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	PWM_OUT	RW	波形的 PWM 电流值 复位值：0x0
[8]	PWM_BUSY	RW	状态机的 PWM 状态 0：空闲 1：忙 复位值：0x0
[7:0]	PWM_CYCLE	RW	在连续模式下，同一组 PWM_per 和 PWM_FP 参数输出的周期数 复位值：0x0

## 6.6.2.16 PWM\_CTRL\_3

- 描述：PWM 通道 3 控制寄存器
- 偏移量：0x60
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	INACTOUT	RW	未启用时的 PWM 输出电平 复位值：0x0
[8]	FPOUT	RW	一个周期内第一相位的 PWM 输出电平 复位值：0x0
[7:6]	EVTRIG	RW	单次模式下的事件触发模式 1：上升沿触发 2：下降沿触发 0&3：通用模式，即 PWM 信号的输出不基于外部信号。 复位值：0x0
[5:4]	MODE	RW	PWM 模式 1：单次模式 2：连续模式 0、3：保留 复位值：0x0
[3]	INTEN	RW	中断使能 0：禁用 1：使能 复位值：0x0
[2]	CFG_UPDATE	RW	在连续模式下，PWM_PER 和 PWM_FP 寄存器在中间更新。 当寄存器的值发生变化时，cfg_update 将从 0 配置为 1（表示两个寄存器值同时更新）。 复位值：0x0
[1]	SOFT_RST	RW	软件复位使能用于随时关闭 PWM_cycle（PWM 状态[7:0]）。
[0]	START	RW	PWM 启动编码使能，上升沿有效（即从 0 到 1）。

### 6.6.2.17 PWM\_RPT\_3

- 描述：PWM 通道 3 单次控制寄存器
- 偏移量：0x64
- 默认值：0x0

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ONE_SHOT_RPT	RW	单次传输中重复的周期数 复位值：0x0

### 6.6.2.18 PWM\_PER\_3

- 描述：PWM 通道 3 周期控制寄存器
- 偏移量：0x68
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_PER	RW	周期波形持续的系统时钟拍数。如果 WM_per 为 0，不输出 PWM 有效信号。 复位值：0x0

### 6.6.2.19 PWM\_FP\_3

- 描述：PWM 通道 3 第一相位控制寄存器
- 偏移量：0x6c
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_FP	RW	周期波形第一相位的拍数。如果 PWM_FP 大于 PWM_per，所有有效信号均为 FPOUT。如果 PWM FP 为 0，所有有效信号均为! FPOUT。 复位值：0x0

### 6.6.2.20 PWM\_STATUS\_3

- 描述：PWM 通道 3 状态寄存器
- 偏移量：0x70
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	PWM_OUT	RW	波形的 PWM 电流值 复位值: 0x0
[8]	PWM_BUSY	RW	状态机的 PWM 状态 0: 空闲 1: 忙 复位值: 0x0
[7:0]	PWM_CYCLE	RW	在连续模式下, 同一组 PWM_per 和 PWM_FP 参数输出的周期数 复位值: 0x0

### 6.6.2.21 PWM\_CTRL\_4

- 描述: PWM 通道 4 控制寄存器
- 偏移量: 0x80
- 默认值: 0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	INACTOUT	RW	未启用时的 PWM 输出电平 复位值: 0x0
[8]	FPOUT	RW	一个周期内第一相位的 PWM 输出电平 复位值: 0x0
[7:6]	EVTRIG	RW	单次模式下的事件触发模式 1: 上升沿触发 2: 下降沿触发 0&3: 通用模式, 即 PWM 信号的输出不基于外部信号。 复位值: 0x0
[5:4]	MODE	RW	PWM 模式 1: 单次模式 2: 连续模式 0、3: 保留

位段	名称	类型	说明
			复位值: 0x0
[3]	INTEN	RW	中断使能 0: 禁用 1: 使能 复位值: 0x0
[2]	CFG_UPDATE	RW	在连续模式下, PWM_PER 和 PWM_FP 寄存器在中间更新。 当寄存器的值发生变化时, cfg_update 将从 0 配置为 1 (表示两个寄存器值同时更新)。 复位值: 0x0
[1]	SOFT_RST	RW	软件复位使能用于随时关闭 PWM_cycle (PWM 状态[7:0])。 复位值: 0x0
[0]	START	RW	PWM 启动编码使能, 上升沿有效 (即从 0 到 1)。 复位值: 0x0

### 6.6.2.22 PWM\_RPT\_4

- 描述: PWM 通道 4 单次控制寄存器
- 偏移量: 0x84
- 默认值: 0x0

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ONE_SHOT_RPT	RW	单次传输中重复的周期数 复位值: 0x0

### 6.6.2.23 PWM\_PER\_4

- 描述: PWM 通道 4 周期控制寄存器
- 偏移量: 0x88
- 默认值: 0x0

位段	名称	类型	说明
[31:0]	PWM_PER	RW	周期波形持续的系统时钟拍数。如果 WM_per 为 0, 不输出 PWM 有效信号。

位段	名称	类型	说明
			复位值：0x0

### 6.6.2.24 PWM\_FP\_4

- 描述：PWM 通道 4 第一相位控制寄存器
- 偏移量：0x8c
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_FP	RW	周期波形第一相位的拍数。如果 PWM_FP 大于 PWM_per，所有有效信号均为 FPOUT。如果 PWM_FP 为 0，所有有效信号均为! FPOUT。 复位值：0x0

### 6.6.2.25 PWM\_STATUS\_4

- 描述：PWM 通道 4 状态寄存器
- 偏移量：0x90
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	PWM_OUT	RW	波形的 PWM 电流值 复位值：0x0
[8]	PWM_BUSY	RW	状态机的 PWM 状态 0：空闲 1：忙 复位值：0x0
[7:0]	PWM_CYCLE	RW	在连续模式下，同一组 PWM_per 和 PWM_FP 参数输出的周期数 复位值：0x0

### 6.6.2.26 PWM\_CTRL\_5

- 描述：PWM 通道 5 控制寄存器
- 偏移量：0xa0



- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	INACTOUT	RW	未启用时的 PWM 输出电平 复位值：0x0
[8]	FPOUT	RW	一个周期内第一相位的 PWM 输出电平 复位值：0x0
[7:6]	EVTRIG	RW	单次模式下的事件触发模式 1：上升沿触发 2：下降沿触发 0&3：通用模式，即 PWM 信号的输出不基于外部信号。 复位值：0x0
[5:4]	MODE	RW	PWM 模式 1：单次模式 2：连续模式 0、3：保留 复位值：0x0
[3]	INTEN	RW	中断使能 0：禁用 1：使能 复位值：0x0
[2]	CFG_UPDATE	RW	在连续模式下，PWM_PER 和 PWM_FP 寄存器在中间更新。 当寄存器的值发生变化时，cfg_update 将从 0 配置为 1（表示两个寄存器值同时更新）。 复位值：0x0
[1]	SOFT_RST	RW	软件复位使能用于随时关闭 PWM_cycle（PWM 状态[7:0]）。 复位值：0x0
[0]	START	RW	PWM 启动编码使能，上升沿有效（即从 0 到 1）。 复位值：0x0

### 6.6.2.27 PWM\_RPT\_5

- 描述：PWM 通道 5 单次控制寄存器
- 偏移量：0xa4
- 默认值：0x0

位段	名称	类型	说明
[31:16]	RESERVED_1	-	
[15:0]	ONE_SHOT_RPT	RW	单次传输中重复的周期数 复位值：0x0

### 6.6.2.28 PWM\_PER\_5

- 描述：PWM 通道 5 周期控制寄存器
- 偏移量：0xa8
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_PER	RW	周期波形持续的系统时钟拍数。如果 WM_per 为 0，不输出 PWM 有效信号。 复位值：0x0

### 6.6.2.29 PWM\_FP\_5

- 描述：PWM 通道 5 第一相位控制寄存器
- 偏移量：0xac
- 默认值：0x0

位段	名称	类型	说明
[31:0]	PWM_FP	RW	周期波形第一相位的拍数。如果 PWM_FP 大于 PWM_per，所有有效信号均为 FPOUT。如果 PWM FP 为 0，所有有效信号均为! FPOUT。 复位值：0x0

### 6.6.2.30 PWM\_STATUS\_5

- 描述：PWM 通道 5 状态寄存器
- 偏移量：0xb0
- 默认值：0x0

位段	名称	类型	说明
[31:10]	RESERVED_1	-	
[9]	PWM_OUT	RW	波形的 PWM 电流值 复位值: 0x0
[8]	PWM_BUSY	RW	状态机的 PWM 状态 0: 空闲 1: 忙 复位值: 0x0
[7:0]	PWM_CYCLE	RW	在连续模式下, 同一组 PWM_per 和 PWM_FP 参数输出的周期数 复位值: 0x0

### 6.6.2.31 PWM\_INTR\_STA

- 描述: PWM 中断状态寄存器
- 偏移量: 0xff0
- 默认值: 0x0

位段	名称	类型	说明
[31:6]	RESERVED_1	-	
[5]	PWM5_INTR	RO	5 输出的 PWM 中断状态 复位值: 0x0
[4]	PWM4_INTR	RO	输出的 PWM4 中断状态 复位值: 0x0
[3]	PWM3_INTR	RO	输出的 PWM3 中断状态 复位值: 0x0
[2]	PWM2_INTR	RO	输出的 PWM2 中断状态 复位值: 0x0
[1]	PWM1_INTR	RO	输出的 PWM1 中断状态 复位值: 0x0
[0]	PWM0_INTR	RO	输出的 PWM0 中断状态 复位值: 0x0

## 7 I2C

---

### 7.1 概述

I2C 总线为系统中集成电路之间的通信链路提供支持。它是一种简单的两线总线，具有软件定义的系统控制协议，用于温度传感器、到 EEPROM 的电压电平转换器、通用 I/O、A/D 和 D/A 转换器、CODEC 和许多类型的微处理器。

芯片内有 9 个 I2C 实例，1 个在常开电源域，2 个在音频子系统，6 个在掉电电源域。

### 7.2 主要特性

I2C 具有以下特性：

- 两线 I2C 串行接口-由串行数据线（SDA）和串行时钟（SCL）组成
- 三种速度：
  - 标准模式（0 到 100Kb/s）
  - 快速模式（ $\leq 400\text{Kb/s}$ ）或快速模式+（ $\leq 1000\text{ Kb/s}$ ）
  - 高速模式（ $\leq 3.4\text{Mb/s}$ ）
- 时钟同步
- 主或从 I2C 操作
- 7 位或 10 位寻址
- 7 位或 10 位组合格式传输
- 批量发送模式
- 发送和接收缓冲区
- 中断和轮询模式操作
- 处理所有总线速度下的比特和字节等待
- DMA 握手接口
- 可编程的 SDA 保持时间（tHD；DAT）
- 总线清除特性
- 设备 ID 特性

## 7.3 接口

图表 7-1 管脚描述

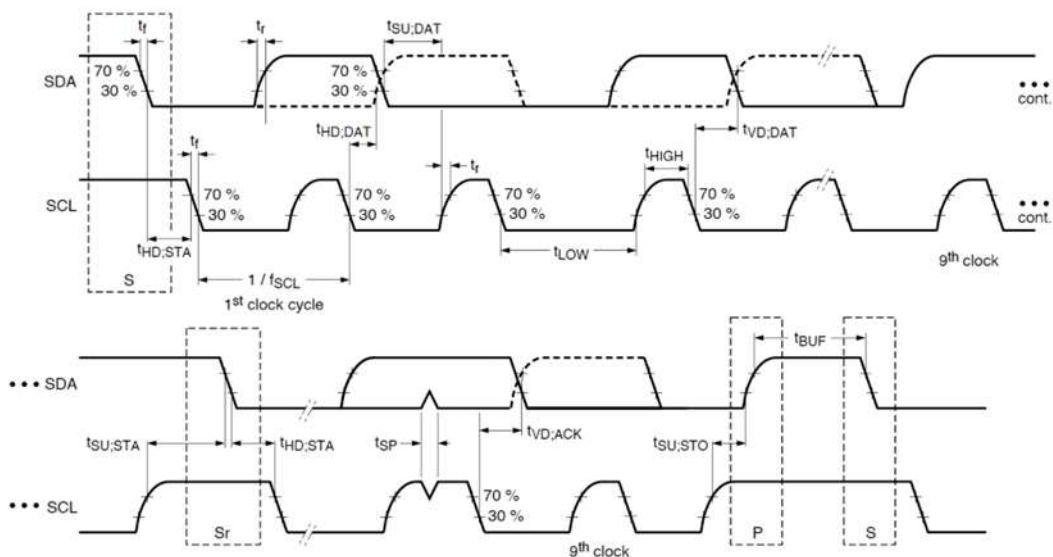
管脚名称	方向	宽度	说明
I2C_AON_SCL	I/O	1	I2C 时钟信号，常开电源域中的 I2C
I2C_AON_SDA	I/O	1	I2C 数据信号，常开电源域中的 I2C
I2C[i]_SCL, i=0~5	I/O	1	I2C 时钟信号，掉电电源域中的 I2C
I2C[i]_SDA, i=0~5	I/O	1	I2C 数据信号，掉电电源域中的 I2C
AUDIO_PA6	I/O	1	音频子系统中的 I2C0, I2C0_SCL
AUDIO_PA7	I/O	1	音频子系统中的 I2C0, I2C0_SDA
AOGPIO_9	I/O	1	音频子系统中的 I2C1, I2C1_SCL
AOGPIO_10	I/O	1	音频子系统中的 I2C1, I2C1_SDA

I2C 支持标准模式和快速模式。其时序参数如图表 7-2 所示。

图表 7-2 I2C 标准模式和快速模式时序特性

符号	参数	条件	标准模式		快速模式		单位
			最小值	最大值	最小值	最大值	
$t_{SP}$	必须由输入滤波器抑制的尖峰脉冲宽度		-	-	0	50	ns
$f_{SCL}$	SCL 时钟频率		0	100	0	400	kHz
$t_{HD;STA}$	重复启动条件的保持时间	在这段时间之后，产生第一个时钟脉冲	4.0	-	0.6	-	us
$t_{LOW}$	SCL 时钟的低电平周期		4.7	-	1.3	-	us
$t_{HIGH}$	SCL 时钟的高电平周期		4.0	-	0.6	-	us
$t_{SU;STA}$	重复启动条件的设置时间		4.7	-	0.6	-	us
$t_{HD;DAT}$	数据保持时间		5.0	-	-	-	us
$t_{SU;DAT}$	数据设置时间		250	-	100	-	ns
$t_r$	SDA 和 SCL 的上升时间		-	1000	20	300	ns
$t_f$	SDA 和 SCL 的下降时间		-	300	-	300	ns

符号	参数	条件	标准模式		快速模式		单位
			最小值	最大值	最小值	最大值	
$t_{SU;STO}$	停止条件的设置时间		4.0	-	0.6	-	us
$t_{BUF}$	停止和启动条件之间的总线空闲时间		4.7	-	1.3	-	us
$C_b$	每条总线的容性负载		-	400	-	400	pF
$t_{VD;DAT}$	数据有效时间		-	3.45	-	0.9	us
$t_{VD;ACK}$	数据有效确认时间		-	3.45	-	0.9	us

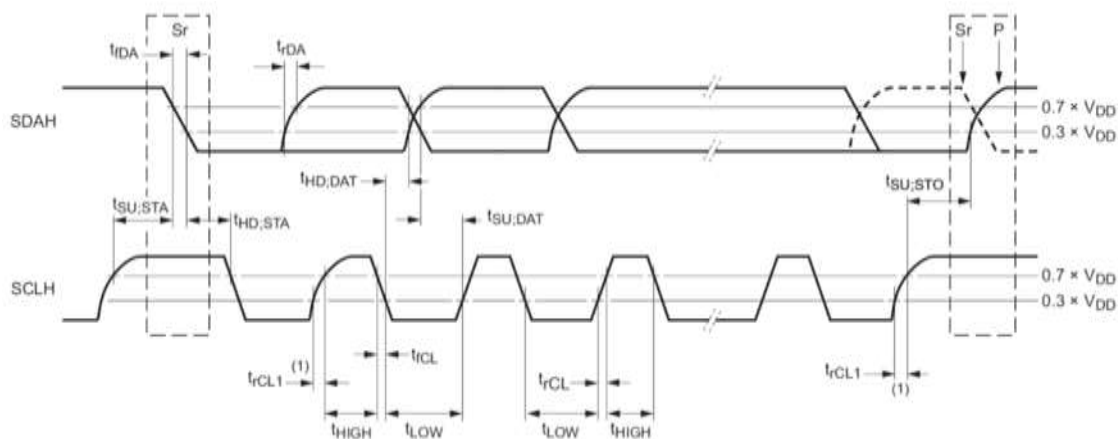


图表 7-3 I2C 标准模式和快速模式时序特性

图表 7-4 I2C 高速模式时序特性

符号	参数	条件	高速模式		单位
			最小值	最大值	
$t_{SP}$	必须由输入滤波器抑制的尖峰脉冲宽度		0	10	ns
$f_{SCL}$	SCL 时钟频率		0	3.4	MHz
$t_{SU;STA}$	重复启动条件的设置时间		160	-	ns
$t_{HD;STA}$	重复启动条件的保持时间		160	-	ns
$t_{LOW}$	SCL 时钟的低电平周期		160	-	ns

符号	参数	条件	高速模式		单位
			最小值	最大值	
$t_{\text{HIGH}}$	SCL 时钟的高电平周期		60	-	ns
$t_{\text{SU;DAT}}$	数据设置时间		10	-	ns
$t_{\text{HD;DAT}}$	数据保持时间		0	70	ns
$t_{\text{rCL}}$	SCL 的上升时间		10	40	ns
$t_{\text{rCL1}}$	在重复启动条件和确认位之后 SCL 的上升时间		10	80	ns
$t_{\text{fCL}}$	SCL 的下降时间		10	40	ns
$t_{\text{rDA}}$	SDA 的上升时间		10	80	ns
$t_{\text{fDA}}$	SDA 的下降时间		10	80	ns
$t_{\text{SU;STO}}$	停止条件的设置时间		160	-	ns
$C_b$	每条总线的容性负载	SDA 和 SCL 线	-	100	pF



图表 7-5 I2C 高速模式时序特性

## 7.4 使用

要将 I2C 用作从，请执行如下步骤：

1. 将 0 写入 IC\_ENABLE 寄存器的 bit0，禁用 I2C。
2. 写入 IC\_SAR 寄存器（bit[9:0]）设置从地址。这是 I2C 响应的地址。
3. 写入 IC\_CON 寄存器以指定支持的寻址类型（通过设置 bit3 选择 7 位或 10 位）。将 0 写入 bit6（IC\_SLAVE\_DISABLE）并将 0 写入 bit0（MASTER\_MODE），在仅从模式下启用 I2C。

4. 将 1 写入寄存器 IC\_ENABLE 的 bit0，启用 I2C。

在 I2C\_DYNAMIC\_TAR\_UPDATE 配置时使用 I2C 作为主：

1. 将 0 写入 IC\_ENABLE 寄存器，禁用 I2C。
2. 写入 IC\_CON 寄存器以设置支持的最大速度模式（bit[2:1]）和所需的 I2C 主启动传输的速度，7 位或 10 位寻址（bit4）。确 bit6（IC\_SLAVE\_DISABLE）写入 1，bit0（MASTER\_MODE）写入 1。
3. 将要寻址的 I2C 设备的地址（bit[9:0]）写入寄存器 IC\_TAR。这个寄存器还指示 I2C 是否将执行通用广播或起始字节命令。
4. 仅适用于高速模式传输。将所需的 I2C 主代码写入 IC\_HS\_MADDR 寄存器。主代码是程序员定义的。
5. 将 1 写入 IC\_ENABLE 寄存器的 bit0，启用 I2C。
6. 现在将传输方向和要发送的数据写入 IC\_DATA\_CMD 寄存器。如果在 I2C 启用之前写入 IC\_DATA\_CMD 寄存器，数据和命令将会丢失，因为当 I2C 禁用时缓冲区会保持清除状态。该步骤在 I2C 上生成启动条件和地址字节。一旦 I2C 启用且 TX FIFO 中有数据，I2C 就会开始读取数据。

## 7.5 寄存器描述

### 7.5.1 寄存器内存映射

寄存器	偏移量	说明	章节/页码
IC_CON	0x0	I2C 控制寄存器	<a href="#">7.5.2.1/246</a>
IC_TAR	0x4	I2C 目标地址寄存器	<a href="#">7.5.2.2/250</a>
IC_SAR	0x8	I2C 从地址寄存器	<a href="#">7.5.2.3/252</a>
IC_HS_MADDR	0xc	I2C 高速主模式代码地址寄存器	<a href="#">7.5.2.4/253</a>
IC_DATA_CMD	0x10	I2C 发送/接收数据缓冲和命令寄存器	<a href="#">7.5.2.5/253</a>
IC_SS_SCL_HCNT	0x14	标准速度 I2C 时钟 SCL 高计数寄存器	<a href="#">7.5.2.6/255</a>
IC_SS_SCL_LCNT	0x18	标准速度 I2C 时钟 SCL 低计数寄存器	<a href="#">7.5.2.7/255</a>
IC_FS_SCL_HCNT	0x1c	快速模式或快速模式+I2C 时钟 SCL 高计数寄存器	<a href="#">7.5.2.8/256</a>
IC_FS_SCL_LCNT	0x20	快速模式或快速模式+I2C 时钟 SCL 低计数寄存器	<a href="#">7.5.2.9/257</a>
IC_HS_SCL_HCNT	0x24	高速 I2C 时钟 SCL 高计数寄存器	<a href="#">7.5.2.10/258</a>
IC_HS_SCL_LCNT	0x28	高速 I2C 时钟 SCL 低计数寄存器	<a href="#">7.5.2.11/258</a>
IC_INTR_STAT	0x2c	I2C 中断状态寄存器	<a href="#">7.5.2.12/259</a>



寄存器	偏移量	说明	章节/页码
IC_INTR_MASK	0x30	I2C 中断屏蔽寄存器	<a href="#">7.5.2.13/263</a>
IC_RAW_INTR_STAT	0x34	I2C 原始中断状态寄存器	<a href="#">7.5.2.14/266</a>
IC_RX_TL	0x38	I2C 接收 FIFO 阈值寄存器	<a href="#">7.5.2.15/267</a>
IC_TX_TL	0x3c	I2C 发送 FIFO 阈值寄存器	<a href="#">7.5.2.16/268</a>
IC_CLR_INTR	0x40	清除组合和单个中断寄存器	<a href="#">7.5.2.17/268</a>
IC_CLR_RX_UNDER	0x44	清除 RX_UNDER 中断寄存器	<a href="#">7.5.2.18/269</a>
IC_CLR_RX_OVER	0x48	清除 RX_OVER 中断寄存器	<a href="#">7.5.2.19/269</a>
IC_CLR_TX_OVER	0x4c	清除 TX_OVER 中断寄存器	<a href="#">7.5.2.20/270</a>
IC_CLR_RD_REQ	0x50	清除 RD_REQ 中断寄存器	<a href="#">7.5.2.21/270</a>
IC_CLR_TX_ABRT	0x54	清除 TX_ABRT 中断寄存器	<a href="#">7.5.2.22/271</a>
IC_CLR_RX_DONE	0x58	清除 RX_DONE 中断寄存器	<a href="#">7.5.2.23/271</a>
IC_CLR_ACTIVITY	0x5c	清除 ACTIVITY 中断寄存器	<a href="#">7.5.2.24/272</a>
IC_CLR_STOP_DET	0x60	清除 STOP_DET 中断寄存器	<a href="#">7.5.2.25/272</a>
IC_CLR_START_DET	0x64	清除 START_DET 中断寄存器	<a href="#">7.5.2.26/273</a>
IC_CLR_GEN_CALL	0x68	清除 GEN_CALL 中断寄存器	<a href="#">7.5.2.27/273</a>
IC_ENABLE	0x6c	I2C 使能寄存器	<a href="#">7.5.2.28/274</a>
IC_STATUS	0x70	I2C 状态寄存器	<a href="#">7.5.2.29/276</a>
IC_TXFLR	0x74	I2C 发送 FIFO 水平寄存器	<a href="#">7.5.2.30/279</a>
IC_RXFLR	0x78	I2C 接收 FIFO 水平寄存器	<a href="#">7.5.2.31/280</a>
IC_SDA_HOLD	0x7c	I2C SDA 保持时间长度寄存器	<a href="#">7.5.2.32/281</a>
IC_TX_ABRT_SOURCE	0x80	I2C 发送中止源寄存器	<a href="#">7.5.2.33/282</a>
IC_DMA_CR	0x88	DMA 控制寄存器	<a href="#">7.5.2.34/288</a>
IC_DMA_TDLR	0x8c	DMA 发送数据水平寄存器	<a href="#">7.5.2.35/289</a>
IC_DMA_RDLR	0x90	I2C 接收数据水平寄存器	<a href="#">7.5.2.36/290</a>
IC_SDA_SETUP	0x94	I2C SDA 设置寄存器	<a href="#">7.5.2.37/290</a>
IC_ACK_GENERAL_CALL	0x98	I2C ACK 通用广播寄存器	<a href="#">7.5.2.38/291</a>

寄存器	偏移量	说明	章节/页码
IC_ENABLE_STATUS	0x9c	I2C 使能状态寄存器	<a href="#">7.5.2.39/292</a>
IC_FS_SPKLEN	0xa0	I2C SS、FS 或 FM+尖峰抑制限制	<a href="#">7.5.2.40/294</a>
IC_HS_SPKLEN	0xa4	I2C 高速尖峰抑制限制寄存器	<a href="#">7.5.2.41/295</a>
IC_SCL_STUCK_AT_LOW_TIMEOUT	0xac	I2C SCL 卡在低超时寄存器	<a href="#">7.5.2.42/295</a>
IC_SDA_STUCK_AT_LOW_TIMEOUT	0xb0	I2C SDA 卡在低超时寄存器	<a href="#">7.5.2.43/296</a>
IC_CLR_SCL_STUCK_DET	0xb4	清除 SCL 卡在低检测中断寄存器	<a href="#">7.5.2.44/296</a>
REG_TIMEOUT_RST	0xf0	寄存器超时计数器复位值	<a href="#">7.5.2.45/297</a>
IC_COMP_PARAM_1	0xf4	组件参数寄存器	<a href="#">7.5.2.46/297</a>
IC_COMP_VERSION	0xf8	I2C 组件版本寄存器	<a href="#">7.5.2.47/300</a>
IC_COMP_TYPE	0xfc	I2C 组件类型寄存器	<a href="#">7.5.2.48/300</a>

## 7.5.2 寄存器和字段描述

### 7.5.2.1 IC\_CON

- 寄存器名：I2C 控制寄存器
- 描述：I2C 控制寄存器。该寄存器只能在 DW\_apb\_i2c 禁用时写入，对应于 IC\_ENABLE[0]寄存器设置为 0。其他时间写入无效。
- 大小：32 bits
- 偏移量：0x0
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_CON_2	R	IC_CON_2 保留位 - 只读 存在：一直
15:12	RSVD_IC_CON_1	R	IC_CON_1 保留位 - 只读 存在：一直
11	BUS_CLEAR_FEATURE_CTRL	R/W	在主模式下： <ul style="list-style-type: none"> <li>■ 1'b1：启用总线清除功能。</li> <li>■ 1'b0：禁用总线清除功能。</li> </ul>

位段	名称	类型	说明
			<p>在从模式下，该寄存器位不适用。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用总线清除功能。</li> <li>■ 0x0 (DISABLED): 禁用总线清除功能。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>
10	STOP_DET_IF_MASTER_ACTIVE	RO	<p>在主模式下：</p> <ul style="list-style-type: none"> <li>■ 1'b1: 主仅处于激活状态时才发出 STOP_DET 中断。</li> <li>■ 1'b0: 无论是否处于激活状态，主都会发出 STOP_DET 中断。</li> </ul> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 主仅处于激活状态时发出 STOP_DET 中断。</li> <li>■ 0x0 (DISABLED): 无论是否处于激活状态，主都会发出 STOP_DET 中断。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>
9	RX_FIFO_FULL_HLD_CTRL	RO	<p>该位控制 DW_apb_i2c 是否应在 Rx FIFO 物理满至 RX_BUFFER_DEPTH 时保持总线。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): RX_FIFO 满时保持总线。</li> <li>■ 0x0 (DISABLED): RX_FIFO 满时溢出。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>
8	TX_EMPTY_CTRL	R/W	<p>该位控制 TX_EMPTY 中断的产生，如 IC_RAW_INTR_STAT 寄存器中所述。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 控制 TX_EMPTY 中断的生成。</li> <li>■ 0x0 (DISABLED): TX_EMPTY 中断的默认行为。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>

位段	名称	类型	说明
7	STOP_DET_IFADDRESSED	R/W	<p>在从模式下：</p> <ul style="list-style-type: none"> <li>1'b1：仅在被寻址时才发出 STOP_DET 中断。</li> <li>0'b0：无论是否被寻址，都发出 STOP_DET 中断。</li> </ul> <p>注：在通用广播地址期间，如果 STOP_DET_IF_ADDRESSED = 1'b1，即使从通过产生 ACK 响应通用调用广播地址，也不会发出 STOP_DET 中断。只有当发送的地址与从地址（SAR）相匹配时，才会产生 STOP_DET 中断。</p> <p>值：</p> <ul style="list-style-type: none"> <li>0x1 (ENABLED)：从仅在被寻址时发出 STOP_DET 中断。</li> <li>0x0 (DISABLED)：从总是发出 STOP_DET 中断。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>
6	IC_SLAVE_DISABLE	R/W	<p>该位控制 I2C 是否禁用其从，这意味着一旦应用了 presetn 信号，该位将采用配置参数 IC_SLAVE_DISABLE 的值。您可以选择在应用复位后启用或禁用从，这意味着软件不必配置从。默认情况下，从始终处于启用状态（复位状态下也是如此）。如果您需要在复位后禁用它，请将该位设置为 1。</p> <p>如果该位被设置（从被禁用），DW_apb_i2c 仅作为主运行，不执行任何需要从的操作。</p> <p>注：软件应确保如果该位写入 0，bit0 也应写入 0。</p> <p>值：</p> <ul style="list-style-type: none"> <li>0x1 (SLAVE_DISABLED)：禁用从模式。</li> <li>0x0 (SLAVE_ENABLED)：启用从模式。</li> </ul> <p>复位值：IC_SLAVE_DISABLE</p> <p>存在：一直</p>
5	IC_RESTART_EN	R/W	<p>确定作为主时是否可以发送重启条件。一些较旧的从不支持处理重启条件；但是，在几个 DW_apb_i2c 操作中使用了重启条件。禁用重启时，禁止主执行以下功能：</p> <ul style="list-style-type: none"> <li>发送起始字节</li> <li>执行任何高速模式操作</li> <li>高速模式操作</li> <li>在组合格式模式下执行方向更改</li> <li>通过 10 位地址执行读操作</li> </ul>

位段	名称	类型	说明
			通过将重启条件替换为停止和随后的启动条件，拆分操作被分解为多个 DW_apb_i2c 传输。如果执行上述操作，将导致设置 IC_RAW_INTR_STAT 寄存器的 bit 6 (TX_ABRT)。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用主重启。</li> <li>■ 0x0 (DISABLED): 禁用主重启。</li> </ul> 复位值: IC_RESTART_EN 存在：一直
4	RSVD_IC_CON_1	R	IC_CON_1 保留位 - 只读 存在：一直
3	IC_10BITADDR_SLAVE	R/W	当作为从时，该位控制 DW_apb_i2c 是响应 7 位还是 10 寻址。 <ul style="list-style-type: none"> <li>■ 0: 7 位寻址。DW_apb_i2c 忽略涉及 10 位寻址的事务；对于 7 位寻址，仅比较 IC_SAR 寄存器的低 7 位。</li> <li>■ 1: 10 位寻址。DW_apb_i2c 仅响应与 IC_SAR 寄存器的全部 10 位匹配的 10 位寻址传输。</li> </ul> 值： <ul style="list-style-type: none"> <li>■ 0x1 (ADDR_10BITS): 从 10 位寻址</li> <li>■ 0x0 (ADDR_7BITS): 从 7 位寻址</li> </ul> 复位值: IC_10BITADDR_SLAVE 存在：一直
2:1	SPEED	R/W	该位控制 DW_apb_i2c 的运行速度；仅在主模式下运行 DW_apb_i2c 时，其设置才相关。硬件防止软件编程的非法值。 这些位也必须针对从模式进行适当编程，因为它用于根据速度模式捕获正确的尖峰滤波器值。 该寄存器只能使用 1 到 IC_MAX_SPEED_MODE 范围内的值进行编程；否则，硬件会使用 IC_MAX_SPEED_MODE 的值更新寄存器。 1: 标准模式 (100kbit/s) 2: 快速模式 (<= 400kbit/s) 或快速模式+ (<= 1000Kbit/s) 3: 高速模式 (3.4Mbit/s) 注：当 IC_ULTRA_FAST_MODE = 1 时，该字段不适用。

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> <li>■ 0x1 (STANDARD): 标准速度运行模式</li> <li>■ 0x2 (FAST): 快速或快速+运行模式</li> <li>■ 0x3 (HIGH): 高速运行模式</li> </ul> 复位值: IC_MAX_SPEED_MODE 存在: 一直
0	MASTER_MODE	R/W	该位控制是否启用 DW_apb_i2c 主。 注: 软件应该确保如果该位写入 1, bit6 也应写入 1。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用主模式。</li> <li>■ 0x0 (DISABLED): 禁用主模式。</li> </ul> 复位值: IC_MASTER_MODE 存在: 一直

### 7.5.2.2 IC\_TAR

- 寄存器名: I2C 目标地址寄存器
- 描述: I2C 目标地址寄存器

如果配置参数 I2C\_DYNAMIC\_TAR\_UPDATE 设置为“否”(0), 该寄存器为 12 位宽, 并且 bits 31:12 被保留。只有当 IC\_ENABLE[0] 设置为 0 时, 才能写入该寄存器。但是, 如果 I2C\_DYNAMIC\_TAR\_UPDATE = 1, 寄存器变为 13 位宽。在这种情况下, 当以下条件之一为真时, 写入 IC\_TAR 会成功:

- DW\_apb\_i2c 未启用 (IC\_ENABLE[0] 设置为 0); 或
- 启用 DW\_apb\_i2c (IC\_ENABLE[0] = 1); 并且 DW\_apb\_i2c 没有参与任何主 (TX, RX) 操作 (IC\_STATUS[5] = 0); 并且 DW\_apb\_i2c 被启用以主模式运行 (IC\_CON[0] = 1); 并且 TX FIFO 中没有条目 (IC\_STATUS[2] = 1)。

只有满足以下条件, 才能动态更改目标地址而不会失去总线。

- 启用 DW\_apb\_i2c (IC\_ENABLE[0] = 1); IC\_EMPTYFIFO\_HOLD\_MASTER\_EN 配置参数设置为 1; DW\_apb\_i2c 被启用以主模式运行 (IC\_CON[0] = 1); TX FIFO 中没有条目并且主处于保持状态 (IC\_INTR\_STAT[13] = 1)。

注: 如果软件或应用程序知道 DW\_apb\_i2c 没有使用 TX FIFO 中的挂起命令的目标地址, 那么即使 TX FIFO 有条目 (IC\_STATUS[2] = 0), 也可以更新目标地址。

- 如果 DW\_apb\_i2c 仅作为 I2C 从启用, 无需对该寄存器执行任何写操作。

- 大小：32 bits
- 偏移量：0x4
- 存在：一直

位段	名称	类型	说明
31:13	RSVD_IC_TAR	R	IC_TAR_2 保留位 - 只读 存在：一直
12	IC_10BITADDR_MASTER	R/W	该位控制 DW_apb_i2c 在作为主时是以 7 位还是 10 位寻址模式开始传输。 <ul style="list-style-type: none"> <li>■ 0: 7 位寻址</li> <li>■ 1: 10 位寻址</li> </ul> 值: <ul style="list-style-type: none"> <li>■ 0x1 (ADDR_10BITS): 10 位地址传输格式</li> <li>■ 0x0 (ADDR_7BITS): 7 位地址传输格式</li> </ul> 复位值: IC_10BITADDR_MASTER 存在：一直
11	SPECIAL	R/W	该位指示软件是否执行 Device-ID、General Call 或 START BYTE 命令。 <ul style="list-style-type: none"> <li>■ 0: 忽略 bit10 GC_OR_START, 正常使用 IC_TAR。</li> <li>■ 1: 执行 Device_ID 或 GC_OR_START 位指定的特殊 I2C 命令。</li> </ul> 值: <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用 GENERAL_CALL 或 START_BYTE 传输的编程。</li> <li>■ 0x0 (DISABLED): 禁用 GENERAL_CALL 或 START_BYTE 传输的编程。</li> </ul> 复位值: 0x0 存在：一直
10	GC_OR_START	R/W	如果 bit 11 (SPECIAL) 设置为 1, bit 13 (Device-ID) 设置为 0, 该位指示 DW_apb_i2c 是否要执行通用广播或起始字节命令。 <ul style="list-style-type: none"> <li>■ 0: 通用广播地址-发出通用广播后, 只能执行写操作。尝试发出读命令会导致设置 IC_RAW_INTR_STAT 寄存器的 bit 6 (TX_ABRT)。DW_apb_i2c 保持在通用广播模式,</li> </ul>

位段	名称	类型	说明
			直到 SPECIAL 位 (bit 11) 值被清除。 ■ 1: 起始字节 值: ■ 0x1 (START_BYTE): 起始字节传输 ■ 0x0 (GENERAL_CALL): 通用广播字节传输 复位值: 0x0 存在: 一直
9:0	IC_TAR	R/W	这是任何主事务的目标地址。发送通用广播时, 这些位被忽略。要生成一个起始字节, CPU 只需向这些位写一次。 如果 IC_TAR 和 IC_SAR 相同, 则存在环回, 但 FIFO 在主和从之间共享, 因此完全环回是不可行的。仅支持单向环回模式 (单工), 不支持双工。主无法发送给自己; 它只能发送给从。 复位值: IC_DEFAULT_TAR_SLAVE_ADDR 存在: 一直

### 7.5.2.3 IC\_SAR

- 寄存器名: I2C 从地址寄存器
- 描述: I2C 从地址寄存器
- 大小: 32 bits
- 偏移量: 0x8
- 存在: 一直

位段	名称	类型	说明
31:10	RSVD_IC_SAR	R	IC_SAR 保留位 - 只读 存在: 一直
9:0	IC_SAR	R/W	当 I2C 作为从运行时, IC_SAR 保持从地址。对于 7 位寻址, 仅使用 IC_SAR[6:0]。 该寄存器只能在 I2C 接口禁用时写入, 对应于 IC_ENABLE[0] 寄存器设置为 0。其他时间写入无效。 注: 默认值不能是任何保留的地址位置: 即 0x00 到 0x07, 或 0x78 到 0x7f。如果将 IC_SAR 或 IC_TAR 编程为保留值, 则不能保证设备的正确运行。有关这些保留值的完整列表, 请参阅表“第一个字节中的位的 I2C/SMBus 定义”。



位段	名称	类型	说明
			复位值: IC_DEFAULT_SLAVE_ADDR 存在: 一直

### 7.5.2.4 IC\_HS\_MADDR

- 寄存器名: I2C 高速主模式代码地址寄存器
- 描述: I2C 高速主模式代码地址寄存器
- 大小: 32 bits
- 偏移量: 0xc
- 存在: 一直

位段	名称	类型	说明
31:3	RSVD_IC_HS_MAR	R	IC_HS_MAR 保留位 - 只读 存在: 一直
2:0	IC_HS_MAR	R/W	该位段保持 I2C 高速模式主代码的值。高速模式主代码是保留的 8 位代码 (00001xxx), 不用于从寻址或其他目的。每个主都有其唯一的主代码; 在同一个 I2C 总线系统上最多可以有 8 个高速模式主。有效值为 0 到 7。如果 IC_MAX_SPEED_MODE 配置参数设置为标准(1)或快速(2), 该寄存器就会消失, 变成只读返回 0。  该寄存器只能在 I2C 接口禁用时写入, 对应于 IC_ENABLE[0] 寄存器设置为 0。其他时间写入无效。  复位值: IC_HS_MASTER_CODE 存在: 一直

### 7.5.2.5 IC\_DATA\_CMD

- 寄存器名: I2C 接收/发送数据缓冲和命令寄存器
- 描述: I2C 接收/发送数据缓冲和命令寄存器; 这是 CPU 在填充发送 FIFO 时写入的寄存器, CPU 在从接收 FIFO 检索字节时读取的寄存器。

寄存器大小变化如下:

写:

- 当 IC\_EMPTYFIFO\_HOLD\_MASTER\_EN = 1 时为 11 位。
- 当 IC\_EMPTYFIFO\_HOLD\_MASTER\_EN = 0 时为 9 位。

读:

- IC\_FIRST\_DATA\_BYTE\_STATUS = 1 时为 12 位。

– 当 IC\_FIRST\_DATA\_BYTE\_STATUS = 0 为位。

注：为了让 DW\_apb\_i2c 继续确认读取，应该为每个要接收的字节写入一个读命令；否则，DW\_apb\_i2c 将停止确认。

- 大小：32 bits
- 偏移量：0x10
- 存在：一直

位段	名称	类型	说明
31:9	RSVD_IC_DATA_CMD	R	IC_DATA_CMD 保留位 - 只读 存在：一直 易失：是
8	CMD	W	该位控制是执行读取还是写入。当 DW_apb_i2c 作为从时，该位不控制方向。当它作为主时，它只控制方向。  当命令输入到发送 FIFO 时，该位区分写和读命令。在从接收模式下，该位是无关的，因为不需要写入该寄存器。在从发送模式下，0 表示要发送 IC_DATA_CMD 中的数据。  编程该位时，您需要记住以下几点：  在发送通用广播命令后尝试执行读操作会导致 TX_ABRT 中断（IC_RAW_INTR_STAT 寄存器的 bit 6），除非 IC_TAR 寄存器的 bit 11（SPECIAL）被清除。如果在收到 RD_REQ 中断后向该位写入 1，发生 TX_ABRT 中断。  值： <ul style="list-style-type: none"> <li>■ 0x1 (READ): 主读命令</li> <li>■ 0x0 (WRITE): 主写命令</li> </ul> 复位值：0x0 存在：一直 易失：是
7:0	DAT	R/W	该寄存器包含要在 I2C 总线上发送或接收的数据。如果您正在写入该寄存器并想要执行读取，DW_apb_i2c 将忽略 bits 7:0（DAT）。但是，当您读取该寄存器时，这些位返回在 DW_apb_i2c 接口上接收到的数据值。  复位值：0x0 存在：一直 易失：是

### 7.5.2.6 IC\_SS\_SCL\_HCNT

- 寄存器名：标准速度 I2C 时钟 SCL 高计数寄存器
- 描述：标准速度 I2C 时钟 SCL 高计数寄存器
- 大小：32 bits
- 偏移量：0x14
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_SS_SCL_HIGH_COUNT	R	IC_SS_SCL_HCNT 保留位 - 只读 存在：一直
15:0	IC_SS_SCL_HCNT	R/W	<p>该寄存器必须在任何 I2C 总线事务发生之前设置，以确保正确的 I/O 时序。该寄存器设置标准速度的 SCL 时钟高周期计数。有关详细信息，请参阅“IC_CLK 频率配置”。</p> <p>该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。</p> <p>最小有效值为 6；硬件会阻止写入小于此值的值，如果尝试会导致设置 6。对于 APB_DATA_WIDTH = 8 的设计，编程顺序对于确保 DW_apb_i2c 的正确运行很重要。必须首先对低字节进行编程。然后对高字节进行编程。</p> <p>当配置参数 IC_HC_COUNT_VALUES 设置为 1 时，该寄存器为只读。</p> <p>注：该寄存器的值不得编程高于 65525；因为当该计数器达到 IC_SS_SCL_HCNT + 10 的值时，DW_apb_i2c 使用 16 位计数器来标记 I2C 总线空闲条件。</p> <p>复位值：IC_SS_SCL_HIGH_COUNT</p> <p>存在：一直</p>

### 7.5.2.7 IC\_SS\_SCL\_LCNT

- 寄存器名：标准速度 I2C 时钟 SCL 低计数寄存器
- 描述：标准速度 I2C 时钟 SCL 低计数寄存器
- 大小：32 bits
- 偏移量：0x18
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_SS_SCL_LOW_COUNT	R	RSVD_IC_SS_SCL_LOW_COUNT 保留位 - 只读 存在：一直
15:0	IC_SS_SCL_LCNT	R/W	<p>该寄存器必须在任何 I2C 总线事务发生之前设置，以确保正确的 I/O 时序。该寄存器设置标准速度的 SCL 时钟低周期计数。有关详细信息，请参阅“IC_CLK 频率配置”。</p> <p>该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。</p> <p>最小有效值为 8；硬件会阻止写入小于此值的值，如果尝试会导致设置 8。对于 APB_DATA_WIDTH = 8 的设计，编程顺序对于确保 DW_apb_i2c 的正确运行很重要。必须首先对低字节进行编程。然后对高字节进行编程。</p> <p>当配置参数 IC_HC_COUNT_VALUES 设置为 1 时，该寄存器为只读。</p> <p>复位值：IC_SS_SCL_LOW_COUNT</p> <p>存在：一直</p>

### 7.5.2.8 IC\_FS\_SCL\_HCNT

- 寄存器名：快速模式或快速模式+I2C 时钟 SCL 高计数寄存器
- 描述：快速模式或快速模式+I2C 时钟 SCL 高计数寄存器
- 大小：32 bits
- 偏移量：0x1c
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_FS_SCL_HCNT	R	IC_FS_SCL_HCNT 保留位 - 只读 存在：一直
15:0	IC_FS_SCL_HCNT	R/W	<p>该寄存器必须在任何 I2C 总线事务发生之前设置，以确保正确的 I/O 时序。该寄存器设置快速模式或快速模式+的 SCL 时钟高周期计数。有关详细信息，请参阅“IC_CLK 频率配置”。</p> <p>如果 IC_MAX_SPEED_MODE = standard，该寄存器消失，变为只读返回 0。</p> <p>该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。</p> <p>最小有效值为 6；硬件会阻止写入小于此值的值，如果尝试会</p>

位段	名称	类型	说明
			导致设置 6。对于 APB_DATA_WIDTH = 8 的设计，编程顺序对于确保 DW_apb_i2c 的正确运行很重要。必须首先对低字节进行编程。然后对高字节进行编程。 复位值：IC_FS_SCL_HIGH_COUNT 存在：一直

### 7.5.2.9 IC\_FS\_SCL\_LCNT

- 寄存器名：快速模式或快速模式+I2C 时钟 SCL 低计数寄存器
- 描述：快速模式或快速模式+I2C 时钟 SCL 低计数寄存器
- 大小：32 bits
- 偏移量：0x20
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_FS_SCL_LCNT	R	IC_FS_SCL_LCNT 保留位 - 只读 存在：一直
15:0	IC_FS_SCL_LCNT	R/W	该寄存器必须在任何 I2C 总线事务发生之前设置，以确保正确的 I/O 时序。该寄存器设置快速的 SCL 时钟低周期计数。它用于在高速模式下发送主代码、起始字节和通用广播。有关详细信息，请参阅“IC_CLK 频率配置”。 如果 IC_MAX_SPEED_MODE = standard，该寄存器消失，变为只读返回 0。 该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。 最小有效值为 8；硬件会阻止写入小于此值的值，如果尝试会导致设置 8。对于 APB_DATA_WIDTH = 8 的设计，编程顺序对于确保 DW_apb_i2c 的正确运行很重要。必须首先对低字节进行编程。然后对高字节进行编程。如果数值小于 8，计数值将更改为 8。 当配置参数 IC_HC_COUNT_VALUES 设置为 1 时，该寄存器为只读。 复位值：IC_FS_SCL_LOW_COUNT 存在：一直

### 7.5.2.10 IC\_HS\_SCL\_HCNT

- 寄存器名：高速 I2C 时钟 SCL 高计数寄存器
- 描述：高速 I2C 时钟 SCL 高计数寄存器
- 大小：32 bits
- 偏移量：0x24
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_HS_SCL_HCNT	R	IC_HS_SCL_HCNT 保留位 - 只读 存在：一直
15:0	IC_HS_SCL_HCNT	R/W	<p>该寄存器必须在任何 I2C 总线事务发生之前设置，以确保正确的 I/O 时序。该寄存器设置高速的 SCL 时钟高周期计数。请参阅“IC_CLK 频率配置”。</p> <p>SCL 高电平时间取决于总线的负载。对于 100pF 负载，SCL 高电平时间为 60ns；对于 400pF 负载，SCL 高电平时间为 120ns。如果 IC_MAX_SPEED_MODE != high，该寄存器消失，变为只读返回 0。</p> <p>该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。</p> <p>最小有效值为 6；硬件会阻止写入小于此值的值，如果尝试会导致设置 6。对于 APB_DATA_WIDTH=8 的设计，编程顺序对于确保 DW_apb_i2c 的正确操作很重要。必须首先对低字节进行编程。然后对高字节进行编程。</p> <p>复位值：IC_HS_SCL_HIGH_COUNT</p> <p>存在：一直</p>

### 7.5.2.11 IC\_HS\_SCL\_LCNT

- 寄存器名：高速 I2C 时钟 SCL 低计数寄存器
- 描述：高速 I2C 时钟 SCL 低计数寄存器
- 大小：32 bits
- 偏移量：0x28
- 存在：一直

位段	名称	类型	说明
31:16	RSVD_IC_HS_SCL_LOW_CNT	R	IC_HS_SCL_LCNT 保留位 - 只读

位段	名称	类型	说明
			存在：一直
15:0	IC_HS_SCL_LCNT	R/W	<p>该寄存器必须在任何 I2C 总线事务发生之前设置，以确保正确的 I/O 时序。该寄存器设置高速的 SCL 时钟高电平周期计数。有关详细信息，请参阅“IC_CLK 频率配置”。</p> <p>SCL 低电平时间取决于总线的负载。对于 100pF 负载，SCL 低电平时间为 160ns；对于 400pF 负载，SCL 低电平时间为 320ns。如果 IC_MAX_SPEED_MODE != high，该寄存器消失，变为只读返回 0。</p> <p>该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。</p> <p>最小有效值为 8；硬件会阻止写入小于此值的值，如果尝试会导致设置 8。对于 APB_DATA_WIDTH = 8 的设计，编程顺序对于确保 DW_apb_i2c 的正确运行很重要。必须首先对低字节进行编程。然后对高字节进行编程。如果数值小于 8，计数值将更改为 8。</p> <p>复位值：IC_HS_SCL_LOW_COUNT</p> <p>存在：一直</p>

### 7.5.2.12 IC\_INTR\_STAT

- 寄存器名：I2C 中断状态寄存器

- 描述：I2C 中断状态寄存器

该寄存器的每一位在 IC\_INTR\_MASK 寄存器中都有一个对应的屏蔽位。这些位通过读取匹配的中断清除寄存器来清除。这些位的未屏蔽原始版本在 IC\_RAW\_INTR\_STAT 寄存器中可用。

- 大小：32 bits
- 偏移量：0x2c
- 存在：一直

位段	名称	类型	说明
31:15	RSVD_IC_INTR_STAT	R	<p>IC_INTR_STAT 保留位 - 只读</p> <p>存在：一直</p> <p>易失：是</p>
14	R_SCL_STUCK_AT_LOW	R	<p>有关 R_SCL_STUCK_AT_LOW 位的详细说明，请参见 IC_RAW_INTR_STAT。</p> <p>值：</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_SCL_STUCK_AT_LOW 中断处于激活状态。</li> <li>■ 0x0 (INACTIVE): R_SCL_STUCK_AT_LOW 中断处于非激活状态。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
13	R_MASTER_ON_HOLD	R	有关 R_MASTER_ON_HOLD 位的详细说明, 请参见 IC_RAW_INTR_STAT。 值: <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_MASTER_ON_HOLD 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_MASTER_ON_HOLD 中断未激活。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
12	R_RESTART_DET	R	有关 R_RESTART_DET 位的详细说明, 请参见 IC_RAW_INTR_STAT。 值: <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_RESTART_DET 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_RESTART_DET 中断未激活。</li> </ul> 复位值: 0x0 存在: 总是 易失: 是
11	R_GEN_CALL	R	有关 R_GEN_CALL 位的详细说明, 请参见 IC_RAW_INTR_STAT。 值: <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_GEN_CALL 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_GEN_CALL 中断未激活。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
10	R_START_DET	R	有关 R_START_DET 位的详细说明, 请参见



位段	名称	类型	说明
			IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_START_DET 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_START_DET 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
9	R_STOP_DET	R	有关 R_STOP_DET 位的详细说明，请参见 IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_STOP_DET 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_STOP_DET 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
8	R_ACTIVITY	R	有关 R_ACTIVITY 位的详细说明，请参见 IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_ACTIVITY 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_ACTIVITY 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
7	R_RX_DONE	R	有关 R_RX_DONE 位的详细说明，请参见 IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_RX_DONE 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_RX_DONE 中断未激活。</li> </ul> 复位值：0x0 存在：总是 易失：是

位段	名称	类型	说明
6	R_TX_ABRT	R	有关 R_TX_ABRT 位的详细说明，请参见 IC_RAW_INTR_STAT。  值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_TX_ABRT 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_TX_ABRT 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
5	R_RD_REQ	R	有关 R_RD_REQ 位的详细说明，请参见 IC_RAW_INTR_STAT。  值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_RD_REQ 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_RD_REQ 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
4	R_TX_EMPTY	R	有关 R_TX_EMPTY 位的详细说明，请参见 IC_RAW_INTR_STAT。  值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_TX_EMPTY 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_TX_EMPTY 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
3	R_TX_OVER	R	有关 R_TX_OVER 位的详细说明，请参见 IC_RAW_INTR_STAT。  值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_TX_OVER 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_TX_OVER 中断未激活。</li> </ul> 复位值：0x0 存在：一直

位段	名称	类型	说明
			易失：是
2	R_RX_FULL	R	有关 R_RX_FULL 位的详细说明，请参见 IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_RX_FULL 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_RX_FULL 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
1	R_RX_OVER	R	有关 R_RX_OVER 位的详细说明，请参见 IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): R_RX_OVER 中断已激活。</li> <li>■ 0x0 (INACTIVE): R_RX_OVER 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是
0	R_RX_UNDER	R	有关 R_RX_UNDER 位的详细说明，请参见 IC_RAW_INTR_STAT。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): RX_UNDER 中断已激活。</li> <li>■ 0x0 (INACTIVE): RX_UNDER 中断未激活。</li> </ul> 复位值：0x0 存在：一直 易失：是

### 7.5.2.13 IC\_INTR\_MASK

- 寄存器名：I2C 中断屏蔽寄存器
- 描述：I2C 中断屏蔽寄存器
- 大小：32 bits
- 偏移量：0x30

这些位屏蔽它们相应的中断状态位。该寄存器低电平有效；值 0 屏蔽中断，而值 1 不屏蔽中断。

- 存在：一直

位段	名称	类型	说明
31:15	RSVD_IC_INTR_STAT	R	IC_INTR_STAT 保留位 - 只读 存在：一直
14	M_SCL_STUCK_AT_LOW	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_SCL_STUCK_AT_LOW 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): SCL_STUCK_AT_LOW 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): SCL_STUCK_AT_LOW 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
13:12	RSVD_IC_INTR_STAT_1	R	IC_INTR_STAT 保留位 - 只读 存在：一直
11	M_GEN_CALL	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_GEN_CALL 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): GEN_CALL 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): GEN_CALL 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
10	M_START_DET	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_START_DET 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): START_DET 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): START_DET 中断屏蔽。</li> </ul> 复位值：0x0 存在：一直
9	M_STOP_DET	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_STOP_DET 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): STOP_DET 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): STOP_DET 中断屏蔽。</li> </ul> 复位值：0x0 存在：一直

位段	名称	类型	说明
8	M_ACTIVITY	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_ACTIVITY 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): ACTIVITY 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): ACTIVITY 中断屏蔽。</li> </ul> 复位值：0x0 存在：一直
7	M_RX_DONE	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_RX_DONE 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): RX_DONE 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): RX_DONE 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
6	M_TX_ABRT	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_TX_ABRT 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): TX_ABORT 未屏蔽。</li> <li>■ 0x0 (ENABLED): TX_ABORT 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
5	M_RD_REQ	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_RD_REQ 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): RD_REQ 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): RD_REQ 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
4	M_TX_EMPTY	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_TX_EMPTY 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): TX_EMPTY 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): TX_EMPTY 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直

位段	名称	类型	说明
3	M_TX_OVER	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_TX_OVER 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): TX_OVER 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): TX_OVER 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
2	M_RX_FULL	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_RX_FULL 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): RX_FULL 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): RX_FULL 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
1	M_RX_OVER	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_RX_OVER 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): RX_OVER 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): RX_OVER 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直
0	M_RX_UNDER	R/W	该位屏蔽 IC_INTR_STAT 寄存器中的 R_RX_UNDER 中断。 值： <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): RX_UNDER 中断未屏蔽。</li> <li>■ 0x0 (ENABLED): RX_UNDER 中断屏蔽。</li> </ul> 复位值：0x1 存在：一直

### 7.5.2.14 IC\_RAW\_INTR\_STAT

- 寄存器名：I2C 原始中断状态寄存器
- 描述：I2C 原始中断状态寄存器  
与 IC\_INTR\_STAT 寄存器不同，这些位没有被屏蔽，因此它们始终显示 DW\_apb\_i2c 的真实状态。
- 大小：32 bits
- 偏移量：0x34

- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_ACTIVITY	R	IC_CLR_ACTIVITY 保留位 - 只读 存在：一直 易失：是
0	CLR_ACTIVITY	R	如果 I2C 不再处于激活状态，读取该寄存器会清除 ACTIVITY 中断。如果 I2C 模块在总线上仍然处于激活状态，ACTIVITY 中断位继续被设置。如果模块被禁用并且总线上没有进一步的活动，它会由硬件自动清除。从该寄存器读取值以获取 IC_RAW_INTR_STAT 寄存器的 ACTIVITY 中断（bit 8）的状态。 复位值：0x0 存在：一直 易失：是

### 7.5.2.15 IC\_RX\_TL

- 寄存器名：I2C 接收 FIFO 阈值寄存器
- 描述：I2C 接收 FIFO 阈值寄存器
- 大小：32 bits
- 偏移量：0x38
- 存在：一直

位段	名称	类型	说明
31:8	RSVD_IC_RX_TL	R	IC_RX_TL 保留位 - 只读 存在：一直
7:0	RX_TL	R/W	接收 FIFO 阈值水平 控制触发 RX_FULL 中断（IC_RAW_INTR_STAT 寄存器中的 bit 2）的条目水平（或更高）。有效范围是 0-255，但硬件不允许将此值设置为大于缓冲区深度的值。如果尝试这样做，设置的实际值将是缓冲区的最大深度。值 0 设置 1 个条目的阈值，值 255 设置 256 个条目的阈值。 复位值：0x0 存在：一直

### 7.5.2.16 IC\_TX\_TL

- 寄存器名：I2C 发送 FIFO 阈值寄存器
- 描述：I2C 发送 FIFO 阈值寄存器
- 大小：32 bits
- 偏移量：0x3c
- 存在：一直

位段	名称	类型	说明
31:8	RSVD_IC_TX_TL	R	IC_TX_TL 保留位 -只读 存在：一直
7:0	TX_TL	R/W	发送 FIFO 阈值水平 控制触发 TX_EMPTY 中断(IC_RAW_INTR_STAT 寄存器中的 bit 4) 的条目水平（或更低）。有效范围是 0-255，但硬件不允许将此值设置为大于缓冲区深度的值。如果尝试这样做，设置的实际值将是缓冲区的最大深度。值 0 设置 0 个条目的阈值，值 255 设置 255 个条目的阈值。 复位值：0x0 存在：一直

### 7.5.2.17 IC\_CLR\_INTR

- 寄存器名：清除组合和单个中断寄存器
- 描述：清除组合和单个中断寄存器
- 大小：32 bits
- 偏移量：0x40
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_INTR	R	CLR_INTR 保留位 - 只读 存在：一直 易失：是
0	CLR_INTR	R	读取该寄存器以清除组合中断、所有单个中断和 IC_TX_ABRT_SOURCE 寄存器。该位不会清除硬件可清除中断，而是清除软件可清除中断。有关清除 IC_TX_ABRT_SOURCE 的例外情况，请参阅 IC_TX_ABRT_SOURCE 寄存器的 bit 9。



位段	名称	类型	说明
			复位值：0x0 存在：一直 易失：是

### 7.5.2.18 IC\_CLR\_RX\_UNDER

- 寄存器名：清除 RX\_UNDER 中断寄存器
- 描述：清除 RX\_UNDER 中断寄存器
- 大小：32 bits
- 偏移量：0x44
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_RX_UNDER	R	IC_CLR_RX_UNDER 保留位 - 只读 存在：一直 易失：是
0	CLR_RX_UNDER	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 RX_UNDER 中断 (bit 0)。 复位值：0x0 存在：一直 易失：是

### 7.5.2.19 IC\_CLR\_RX\_OVER

- 寄存器名：清除 RX\_OVER 中断寄存器
- 描述：清除 RX\_OVER 中断寄存器
- 大小：32 bits
- 偏移量：0x48
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_RX_OVER	R	IC_CLR_RX_OVER 保留位 - 只读 存在：一直 易失：是

位段	名称	类型	说明
0	CLR_RX_OVER	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 RX_OVER 中断 (bit 1)。  复位值: 0x0  存在: 一直  易失: 是

### 7.5.2.20 IC\_CLR\_TX\_OVER

- 寄存器名: 清除 TX\_OVER 中断寄存器
- 描述: 清除 TX\_OVER 中断寄存器
- 大小: 32 bits
- 偏移量: 0x4c
- 存在: 一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_TX_OVER	R	IC_CLR_TX_OVER 保留位 - 只读  存在: 一直  易失: 是
0	CLR_TX_OVER	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 TX_OVER 中断 (bit 3)。  复位值: 0x0  存在: 一直  易失: 是

### 7.5.2.21 IC\_CLR\_RD\_REQ

- 寄存器名: 清除 RD\_REQ 中断寄存器
- 描述: 清除 RD\_REQ 中断寄存器
- 大小: 32 bits
- 偏移量: 0x50
- 存在: 一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_RD_REQ	R	IC_CLR_RD_REQ 保留位 - 只读  存在: 一直

位段	名称	类型	说明
			易失：是
0	CLR_RD_REQ	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 RD_REQ 中断（bit 5）。 复位值：0x0 存在：一直 易失：是

### 7.5.2.22 IC\_CLR\_TX\_ABORT

- 寄存器名：清除 TX\_ABORT 中断寄存器
- 描述：清除 TX\_ABORT 中断寄存器
- 大小：32 bits
- 偏移量：0x54
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_TX_ABORT	R	IC_CLR_TX_ABORT 保留位 - 只读 存在：一直 易失：是
0	CLR_TX_ABORT	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 TX_ABORT 中断（bit 6）和 IC_TX_ABORT_SOURCE 寄存器。这也将发送 FIFO 从刷新/复位状态释放，允许对发送 FIFO 进行更多写入。有关清除 IC_TX_ABORT_SOURCE 的例外情况，请参阅 IC_TX_ABORT_SOURCE 寄存器的 bit 9。 复位值：0x0 存在：一直 易失：是

### 7.5.2.23 IC\_CLR\_RX\_DONE

- 寄存器名：清除 RX\_DONE 中断寄存器
- 描述：清除 RX\_DONE 中断寄存器
- 大小：32 bits
- 偏移量：0x58
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_RX_DONE	R	IC_CLR_RX_DONE 保留位 - 只读 存在：一直 易失：是
0	CLR_RX_DONE	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 RX_DONE 中断 (bit 7)。 复位值：0x0 存在：一直 易失：是

### 7.5.2.24 IC\_CLR\_ACTIVITY

- 寄存器名：清除 ACTIVITY 中断寄存器
- 描述：清除 ACTIVITY 中断寄存器
- 大小：32 bits
- 偏移量：0x5c
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_ACTIVITY	R	IC_CLR_ACTIVITY 保留位 - 只读 存在：一直 易失：是
0	CLR_ACTIVITY	R	如果 I2C 不再处于激活状态，读取该寄存器会清除 ACTIVITY 中断。如果 I2C 模块在总线上仍然处于激活状态，ACTIVITY 中断位继续被设置。如果模块被禁用并且总线上没有进一步的活动，它会由硬件自动清除。从该寄存器读取值以获取 IC_RAW_INTR_STAT 寄存器的 ACTIVITY 中断 (bit 8) 的状态。 复位值：0x0 存在：一直 易失：是

### 7.5.2.25 IC\_CLR\_STOP\_DET

- 寄存器名：清除 STOP\_DET 中断寄存器
- 描述：清除 STOP\_DET 中断寄存器

- 大小：32 bits
- 偏移量：0x60
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_STOP_DET	R	IC_CLR_STOP_DET 保留位 - 只读 存在：一直 易失：是
0	CLR_STOP_DET	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 STOP_DET 中断 (bit 9)。 复位值：0x0 存在：一直 易失：是

### 7.5.2.26 IC\_CLR\_START\_DET

- 寄存器名：清除 START\_DET 中断寄存器
- 描述：清除 START\_DET 中断寄存器
- 大小：32 bits
- 偏移量：0x64
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_START_DET	R	IC_CLR_START_DET 保留位 - 只读 存在：一直 易失：是
0	CLR_START_DET	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 START_DET 中断 (bit 10)。 复位值：0x0 存在：一直 易失：是

### 7.5.2.27 IC\_CLR\_GEN\_CALL

- 寄存器名：清除 GEN\_CALL 中断寄存器
- 描述：清除 GEN\_CALL 中断寄存器

- 大小：32 bits
- 偏移量：0x68
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_CLR_GEN_CALL	R	IC_CLR_GEN_CALL 保留位 - 只读 存在：一直 易失：是
0	CLR_GEN_CALL	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 GEN_CALL 中断 (bit 11)。 复位值：0x0 存在：一直 易失：是

### 7.5.2.28 IC\_ENABLE

- 寄存器名：I2C 使能寄存器
- 描述：I2C 使能寄存器
- 大小：32 bits
- 偏移量：0x6c
- 存在：一直

位段	名称	类型	说明
31:4	RSVD_IC_ENABLE_2	R	IC_ENABLE 保留位 - 只读 存在：一直
3	SDA_STUCK_RECOVERY_ENABLE	R/W	如果通过 TX_ABORT 中断 (IC_TX_ABRT_SOURCE[17]) 指示 SDA 卡在低电平，该位被用作控制旋钮以启动 SDA 恢复机制（即最多发送 9 个 SCL 时钟和 STOP 以释放 SDA 线），然后该位自动清除。 值： <ul style="list-style-type: none"> <li>■ 0x1 (SDA_STUCK_RECOVERY_ENABLED): 主启动 SDA 卡在低电平的恢复机制。</li> <li>■ 0x0 (SDA_STUCK_RECOVERY_DISABLED): 主禁用 SDA 卡在低电平的恢复机制。</li> </ul> 复位值：0x0 存在：一直

位段	名称	类型	说明
2	TX_CMD_BLOCK	R/W	<p>在主模式下：</p> <ul style="list-style-type: none"> <li>1'b1：阻止 I2C 总线上的数据传输，即使发送 FIFO 有数据要传输。</li> <li>1'b0：一旦第一个数据在发送 FIFO 中可用，数据就会在 I2C 总线上自动开始传输。</li> </ul> <p>注：要阻止主命令的执行，仅当发送 FIFO 为空（IC_STATUS[2] == 1）且主处于空闲状态（IC_STATUS[5] == 0）时设置 TX_CMD_BLOCK 位。在 TX_CMD_BLOCK 位取消设置之前，不会执行放入发送 FIFO 中的任何其他命令。</p> <p>值：</p> <ul style="list-style-type: none"> <li>0x1 (BLOCKED)：发送命令执行被阻止。</li> <li>0x0 (NOT_BLOCKED)：发送命令执行未被阻止。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>
1	ABORT	R/W	<p>设置后，控制器启动传输中止。</p> <ul style="list-style-type: none"> <li>0：ABORT 未启动或 ABORT 完成。</li> <li>1：ABORT 操作正在进行中。</li> </ul> <p>软件可以通过设置该位中止主模式下的 I2C 传输。只有当 ENABLE 已经设置时，软件才能设置该位；否则，控制器将忽略任何对 ABORT 位的写入。ABORT 位一旦设置，软件就无法清除。为响应 ABORT，控制器在完成当前传输后发出 STOP 并刷新发送 FIFO，然后中止操作后设置 TX_ABORT 中断。ABORT 位在中止操作后自动清零。</p> <p>有关如何中止 I2C 传输的详细说明，请参阅“中止 I2C 传输”。</p> <p>值：</p> <ul style="list-style-type: none"> <li>0x1 (ENABLED)：ABORT 操作正在进行中。</li> <li>0x0 (DISABLE)：ABORT 操作未在进行中。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>
0	ENABLE	R/W	<p>控制是否启用 DW_apb_i2c。</p> <ul style="list-style-type: none"> <li>0：禁用 DW_apb_i2c。（发送和接收 FIFO 保持在擦除状态）</li> <li>1：启用 DW_apb_i2c。</li> </ul>

位段	名称	类型	说明
			<p>软件可以在 DW_apb_i2c 激活时禁用它。</p> <p>但是，务必注意确保正确禁用 DW_apb_i2c。“禁用 DW_apb_i2c”中描述了推荐的过程。</p> <p>当 DW_apb_i2c 被禁用时，会发生以下情况：</p> <ul style="list-style-type: none"> <li>■ 发送 FIFO 和接收 FIFO 被刷新。</li> <li>■ IC_INTR_STAT 寄存器中的状态位仍然有效直到 DW_apb_i2c 进入空闲状态。</li> </ul> <p>如果模块正在发送，它会在当前传输完成后停止并删除发送缓冲区的内容。如果模块正在接收，DW_apb_i2c 在当前字节结束时停止当前传输并且不确认传输。</p> <p>在具有异步 pclk 和 ic_clk 的系统中，当 IC_CLK_TYPE 参数设置为异步(1)时,启用或禁用 DW_apb_i2c 会有两个 ic_clk 延迟。有关如何禁用 DW_apb_i2c 的详细说明，请参阅“禁用 DW_apb_i2c”。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用 I2C。</li> <li>■ 0x0 (DISABLED): 禁用 I2C。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>

## 7.5.2.29 IC\_STATUS

- 寄存器名：I2C 状态寄存器
  - 描述：I2C 状态寄存器
  - 这是一个只读寄存器，用于指示当前传输状态和 FIFO 状态。可以随时读取状态寄存器。该寄存器中的所有位都不会请求中断。
- 当 I2C 通过在 IC\_ENABLE 寄存器的位 0 写入 0 被禁用时：
- bit 1 和 bit 2 设置为 1。
  - bit 3 和 bit 10 设置为 0。
- 当主或从状态机进入空闲且 ic\_en = 0 时：
- bit 5 和 bit 6 设置为 0。
- 大小：32 bits
  - 偏移量：0x70
  - 存在：一直



位段	名称	类型	说明
31:12	RSVD_IC_STATUS_2	R	IC_STATUS 保留位 - 只读 存在：一直 易失：是
11	SDA_STUCK_NOT_RECOVERED	R	该位指示卡在低电平的 SDA 在恢复机制后没有恢复。在从模式下，该寄存器位不适用。 值： <ul style="list-style-type: none"> <li>0x1 (ACTIVE): 在恢复机制后恢复卡在低电平的 SDA。</li> <li>0x0 (INACTIVE): 在恢复机制后不恢复卡在低电平的 SDA。</li> </ul> 复位值：0x0 存在：一直 易失：是
10:7	RSVD_IC_STATUS_1	R	IC_STATUS 保留位 - 只读 存在：一直 易失：是
6	SLV_ACTIVITY	R	从 FSM 活动状态。当从 FSM 不处于空闲状态时，该位被设置。 <ul style="list-style-type: none"> <li>0: 从 FSM 处于空闲状态，因此 DW_apb_i2c 的从部分未激活。</li> <li>1: 从 FSM 不处于空闲状态，因此 DW_apb_i2c 的从部分已激活。</li> </ul> 值： <ul style="list-style-type: none"> <li>0x1 (ACTIVE): 从不空闲。</li> <li>0x0 (IDLE): 从空闲。</li> </ul> 复位值：0x0 存在：一直 易失：是
5	MST_ACTIVITY	R	主 FSM 激活状态。当主 FSM 不处于空闲状态时，该位被设置。 <ul style="list-style-type: none"> <li>0: 主 FSM 处于空闲状态，因此 DW_apb_i2c 的主部分未激活。</li> <li>1: 主不处于空闲状态，因此 DW_apb_i2c 的主部分已激活。</li> </ul>

位段	名称	类型	说明
			<p>活。</p> <p>注：IC_STATUS[0]-即 ACTIVITY 位-是 SLV_ACTIVITY 和 MST_ACTIVITY 位的 OR。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): 主不空闲。</li> <li>■ 0x0 (IDLE): 主空闲。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
4	RFF	R	<p>接收 FIFO 完全满。当接收 FIFO 完全满时，该位被设置。当接收 FIFO 中包含一个或多个空位置时，该位被清零。</p> <ul style="list-style-type: none"> <li>■ 0: 接收 FIFO 未满。</li> <li>■ 1: 接收 FIFO 已满。</li> </ul> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (FULL): 接收 FIFO 已满。</li> <li>■ 0x0 (NOT_FULL): 接收 FIFO 未满。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
3	RFNE	R	<p>接收 FIFO 非空。当接收 FIFO 中包含一个或多个条目时，该位被设置；当接收 FIFO 为空时清除。</p> <ul style="list-style-type: none"> <li>■ 0: 接收 FIFO 为空。</li> <li>■ 1: 接收 FIFO 不为空。</li> </ul> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (NOT_EMPTY): 接收 FIFO 不为空。</li> <li>■ 0x0 (EMPTY): 接收 FIFO 为空。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
2	TFE	R	<p>发送 FIFO 完全空。当发送 FIFO 完全空时，该位被设置。当它包含一个或多个有效条目时，该位被清除。该位段不请求中断。</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0: 发送 FIFO 不为空。</li> <li>1: 发送 FIFO 为空。</li> </ul> 值: <ul style="list-style-type: none"> <li>0x1 (EMPTY): 发送 FIFO 为空。</li> <li>0x0 (NON_EMPTY): 发送 FIFO 不为空。</li> </ul> 复位值: 0x1 存在: 一直 易失: 是
1	TFNF	R	发送 FIFO 未滿。当发送 FIFO 包含一个或多个空位置时设置, 当 FIFO 满时清除。 <ul style="list-style-type: none"> <li>0: 发送 FIFO 已滿。</li> <li>1: 发送 FIFO 未滿。</li> </ul> 值: <ul style="list-style-type: none"> <li>0x1 (NOT_FULL): 发送 FIFO 未滿。</li> <li>0x0 (FULL): 发送 FIFO 已滿。</li> </ul> 复位值: 0x1 存在: 一直 易失: 是
0	ACTIVITY	R	I2C 活动状态。                     值: <ul style="list-style-type: none"> <li>0x1 (ACTIVE): I2 处于活动状态。</li> <li>0x0 (INACTIVE): I2C 空闲。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是

### 7.5.2.30 IC\_TXFLR

- 寄存器名: I2C 发送 FIFO 水平寄存器
- 描述: I2C 发送 FIFO 水平寄存器

该寄存器包含发送 FIFO 缓冲区中有效数据条目的数量。发生如下情况时, 它被清除:

- I2C 被禁用。
- 发送中止-即在 IC\_RAW\_INTR\_STAT 寄存器中设置 TX\_ABRT 位。

- 从批量传输模式被中止。

每当将数据放入发送 FIFO 时，该寄存器就会递增，而当从发送 FIFO 中取出数据时，该寄存器就会递减。

- 大小：32 bits
- 偏移量：0x74
- 存在：一直

位段	名称	类型	说明
31:5	RSVD_TXFLR	R	TXFLR 寄存器字段保留位 - 只读 存在：一直 易失：是
4:0	TXFLR	R	发送 FIFO 水平。包含发送 FIFO 中有效数据条目的数量。 复位值：0x0 存在：一直 易失：是

### 7.5.2.31 IC\_RXFLR

- 寄存器名：I2C 接收 FIFO 水平寄存器
- 描述：I2C 接收 FIFO 水平寄存器

该寄存器包含接收 FIFO 缓冲区中有效数据条目的数量。发生如下情况时，它被清除：

- I2C 被禁用。
- IC\_TX\_ABRT\_SOURCE 中跟踪的任何事件导致发送中止。

每当将数据放入接收 FIFO 时，该寄存器就会递增，而当从接收 FIFO 中取出数据时，该寄存器就会递减。

- 大小：32 bits
- 偏移量：0x78
- 存在：一直

位段	名称	类型	说明
31:5	RSVD_RXFLR	R	RXFLR 保留位- 只读 存在：一直 易失：是
4:0	RXFLR	R	接收 FIFO 水平。包含接收 FIFO 中有效数据条目的数量。 复位值：0x0

位段	名称	类型	说明
			存在：一直 易失：是

### 7.5.2.32 IC\_SDA\_HOLD

- 寄存器名：I2C SDA 保持时间长度寄存器
- 描述：I2C SDA 保持时间长度寄存器

该寄存器的 bit[15:0]用于控制 SDA 在从模式和主模式下发送期间的保持时间（在 SCL 从高电平变为低电平之后）。

无论在主模式还是从模式下，只要 SCL 在接收器中为高电平，该寄存器 bit[23:16]用于扩展 SDA 转换（如果有）。

仅当 IC\_ENABLE[0] = 0 时，写入该寄存器才会成功。

该寄存器中的值以 ic\_clk 周期为单位。IC\_SDA\_TX\_HOLD 中编程的值必须大于每个模式下的最小保持时间，主模式下 1 个周期，从模式下 7 个周期，该值才能被执行。

编程的发送期间 SDA 保持时间（IC\_SDA\_TX\_HOLD）在任何时候都不能超过 SCL 低电平部分的持续时间。因此，编程值不能大于 N\_SCL\_LOW-2，其中 N\_SCL\_LOW 是以 ic\_clk 周期测量的 scl 周期低电平部分的持续时间。

- 大小：32 bits
- 偏移量：0x7c
- 存在：一直

位段	名称	类型	说明
31:24	RSVD_IC_SDA_HOLD	R	IC_SDA_HOLD 保留位 - 只读 存在：一直
23:16	IC_SDA_RX_HOLD	R/W	当 DW_apb_i2c 充当接收器时，以 ic_clk 周期为单位设置所需的 SDA 保持时间。 复位值：0x0 存在：一直
15:0	IC_SDA_TX_HOLD	R/W	当 DW_apb_i2c 充当发送器时，以 ic_clk 周期为单位设置所需的 SDA 保持时间。 复位值：0x1 存在：一直

### 7.5.2.33 IC\_TX\_ABRT\_SOURCE

- 寄存器名：I2C 发送中止源寄存器
- 描述：I2C 发送中止源寄存器

该寄存器有 32 位指示 TX\_ABRT 位的来源。除了 bit 9，只要读取 IC\_CLR\_TX\_ABRT 寄存器或 IC\_CLR\_INTR 寄存器，该寄存器被清除。要清除 bit 9，必须先固定 ABRT\_SBYTE\_NORSTRT 的来源；必须启用 RESTART (IC\_CON[5] = 1)，必须清除 SPECIAL 位 (IC\_TAR[11])，或者必须清除 GC\_OR\_START 位 (IC\_TAR[10])。

一旦 ABRT\_SBYTE\_NORSTRT 的源被固定，那么该位可以以与该寄存器中的其他位相同的方式被清除。如果在尝试清除该位之前未固定 ABRT\_SBYTE\_NORSTRT 的源，bit 9 会清除一个周期，然后被重新设置。

- 大小：32 bits
- 偏移量：0x80
- 存在：一直

位段	名称	类型	说明
31:23	TX_FLUSH_CNT	R	该字段指示由于 TX_ABRT 中断而刷新的发送 FIFO 数据命令的数量。当 I2C 被禁用时，它被清零。 DW_apb_i2c 的作用：主发送器或从发送器 复位值：0x0 存在：一直 易失：是
22:18	RSVD_IC_TX_ABRT_SOURCE	R	IC_TX_ABRT_SOURCE 保留位 - 只读 存在：一直 易失：是
17	ABRT_SDA_STUCK_AT_LOW	R	这是一个仅限主模式的位。主检测到 SDA 在 ic_clks 的 IC_SDA_STUCK_AT_LOW_TIMEOUT 值时卡在低电平。 DW_apb_i2 的作用：主 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE)：由于 SDA 在 ic_clks 的 IC_SDA_STUCK_AT_LOW_TIMEOUT 值时卡在低电平，因此产生该中止。</li> <li>■ 0x0 (INACTIVE)：不产生该中止。</li> </ul> 复位值：0x0 存在：一直

位段	名称	类型	说明
			易失：是
16	ABRT_USER_ABRT	R	<p>这是一个仅限主模式的位。主检测到传输中止 (IC_ENABLE[1])。</p> <p>DW_apb_i2c 的作用：主发送器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_USER_ABRT_GENERATED)：主检测到传输中止。</li> <li>■ 0x0 (ABRT_USER_ABRT_VOID)：主检测到传输中止-场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
15	ABRT_SLVRD_INTX	R	<p>当处理器侧响应从模式请求将数据发送到远程主并且用户在 IC_DATA_CMD 寄存器的 CMD (bit 8) 写入 1。</p> <p>DW_apb_i2c 的作用：从发送器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_SLVRD_INTX_GENERATED)：从尝试以读取模式发送到远程主。</li> <li>■ 0x0 (ABRT_SLVRD_INTX_VOID)：从尝试以读取模式发送到远程主-场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
14	ABRT_SLV_ARBLOST	R	<p>该字段表示从在向远程主发送数据时失去了总线。</p> <p>同时设置 IC_TX_ABRT_SOURCE[12]。</p> <p>注：即使从未“拥有”总线，总线上也可能出现问题。这是一个故障安全检查。例如，在 SCL 从低到高转换的数据传输期间，如果数据总线上的内容不是应该传输的内容，DW_apb_i2c 不再拥有总线。</p> <p>DW_apb_i2c 的作用：从发送器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_SLV_ARBLOST_GENERATED)：从失去对远程主的仲裁。</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0x0 (ABRT_SLV_ARBLOST_VOID): 从失去对远程主的仲裁-场景不存在。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
13	ABRT_SLVFLUSH_TXFIFO	R	该字段指定从已收到读取命令, 并且发送 FIFO 中存在一些数据, 因此从发出 TX_ABRT 中断以刷新发送 FIFO 中的旧数据。 DW_apb_i2c 的作用: 从发送器 值: <ul style="list-style-type: none"> <li>0x1 (ABRT_SLVFLUSH_TXFIFO_GENERATED): 从收到读取命令后刷新发送 FIFO 中的现有数据。</li> <li>0x0 (ABRT_SLVFLUSH_TXFIFO_VOID): 从收到读取命令后刷新发送 FIFO 中的现有数据-场景不存在。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
12	ARB_LOST	R	该字段指定主已失去仲裁, 或者如果 IC_TX_ABRT_SOURCE[14]也被设置, 从发送器已失去仲裁。 DW_apb_i2c 的作用: 主发送器或从发送器 值: <ul style="list-style-type: none"> <li>0x1 (ABRT_LOST_GENERATED): 主或从发送器失去仲裁。</li> <li>0x0 (ABRT_LOST_VOID): 主或从发送器失去仲裁-场景不存在。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
11	ABRT_MASTER_DIS	R	该字段表示用户尝试在禁用主模式的情况下启动主操作。 DW_apb_i2c 的作用: 主发送器或主接收器 值: <ul style="list-style-type: none"> <li>0x1 (ABRT_MASTER_DIS_GENERATED): 主禁用时用户启动主操作。</li> </ul>



位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0x0 (ABRT_MASTER_DIS_VOID): 主禁用时用户启动主操作-场景不存在。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
10	ABRT_10B_RD_NORSTRT	R	该字段表示禁用重启 (IC_RESTART_EN bit (IC_CON[5]) = 0) 并且主在 10 位寻址模式下发送读命令 DW_apb_i2c 的作用: 主接收器 值: <ul style="list-style-type: none"> <li>0x1 (ABRT_10B_RD_GENERATED): 当 RESTART 禁用时, 主尝试以 10 位寻址模式读取。</li> <li>0x0 (ABRT_10B_RD_VOID) 当 RESTART 禁用时, 主未尝试以 10 位寻址模式读取。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
9	ABRT_SBYTE_NORSTRT	R	要清除 bit 9, 必须先固定 ABRT_SBYTE_NORSTRT 的来源; 必须启用重启 (IC_CON[5] = 1), 必须清除 SPECIAL 位 (IC_TAR[11]), 或者必须清除 GC_OR_STAR 位 (IC_TAR[10])。一旦 ABRT_SBYTE_NORSTRT 的源被固定, 该位可以以与该寄存器中的其他位相同的方式被清除。如果在尝试清除该位之前未固定 ABRT_SBYTE_NORSTRT 的源, bit 9 会清除一个周期, 然后被重新设置。当该字段设置为 1 时, 重启被禁用 (IC_RESTART_EN 位 (IC_CON[5]) = 0), 用户正尝试发送起始字节。 DW_apb_i2c 作用: 主 值: <ul style="list-style-type: none"> <li>0x1 (ABRT_SBYTE_NORSTRT_GENERATED): 用户尝试在禁用 RESTART 时发送起始字节。</li> <li>0x0 (ABRT_SBYTE_NORSTRT_VOID): 用户尝试在禁用 RESTART 时发送起始字节-场景不存在。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是

位段	名称	类型	说明
8	ABRT_HS_NORSTRT	R	<p>该字段表示禁用重启（IC_RESTART_EN 位（IC_CON[5]）= 0）并且用户正在尝试使用主在高速模式下传输数据。</p> <p>DW_apb_i2c 的作用：主发送器或主接收器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_HS_NORSTRT_GENERATED): 用户尝试在禁用 RESTART 将主切换到高速模式。</li> <li>■ 0x0 (ABRT_HS_NORSTRT_VOID): 用户尝试在禁用 RESTART 将主切换到高速模式场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
7	ABRT_SBYTE_ACKDET	R	<p>该字段表示主已发送一个起始字节并且起始字节已被确认（错误行为）。</p> <p>DW_apb_i2c 的作用：主</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_SBYTE_ACKDET_GENERATED): 检测到起始字节的 ACK。</li> <li>■ 0x0 (ABRT_SBYTE_ACKDET_VOID): 检测到起始字节的 ACK-场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
6	ABRT_HS_ACKDET	R	<p>该字段表示主处于高速模式并且高速主代码被确认（错误行为）。</p> <p>DW_apb_i2c 的作用：主</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_HS_ACK_GENERATED): 高速模式下的高速主代码被确认。</li> <li>■ 0x0 (ABRT_HS_ACK_VOID): 高速模式下的高速主代码被确认-场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p>

位段	名称	类型	说明
			易失：是
5	ABRT_GCALL_READ	R	<p>该字段表示主模式下的 DW_apb_i2c 已发送了一个通用广播，但是用户将通用广播之后的字节编程为从总线读取（IC_DATA_CMD[9]设置为 1）。</p> <p>DW_apb_i2c 的作用：主发送器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_GCALL_READ_GENERATED): 通用广播之后是从总线读取。</li> <li>■ 0x0 (ABRT_GCALL_READ_VOID): 通用广播之后是从总线读取-场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
4	ABRT_GCALL_NOACK	R	<p>该字段表示主模式下的 DW_apb_i2c 已发送一个通用广播，并且总线上没有从确认该通用广播。</p> <p>DW_apb_i2c 的作用：主发送器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_GCALL_NOACK_GENERATED): GCALL 没有被任何从确认。</li> <li>■ 0x0 (ABRT_GCALL_NOACK_VOID): GCALL 没有被任何从确认-场景不存在。</li> </ul> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
3	ABRT_TXDATA_NOACK	R	<p>在字段表示仅主模式位。当主收到地址确认，但在地址后面发送数据字节时，它没有收到来自远程从的确认。</p> <p>DW_apb_i2c 的作用：主发送器</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (ABRT_TXDATA_NOACK_GENERATED): 发送的数据没有被寻址的从确认。</li> <li>■ 0x0 (ABRT_TXDATA_NOACK_VOID): 发送的数据没有被寻址的从确认-场景不存在。</li> </ul> <p>复位值：0x0</p>

位段	名称	类型	说明
			存在：一直 易失：是
2	ABRT_10ADDR2_NOACK	R	该字段表示主处于 10 位地址模式，并且 10 位地址的第二个地址字节未被任何从确认。 DW_apb_i2c 的作用：主发送器或主接收器 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): 10 位地址的第二个字节未被任何从确认</li> <li>■ 0x0 (INACTIVE): 不产生此中止。</li> </ul> 复位值：0x0 存在：一直 易失：是
1	ABRT_10ADDR1_NOACK	R	该字段表示主处于 10 位地址模式，并且 10 位地址的第一个字节未被任何从确认。 DW_apb_i2c 的作用：主发送器或主接收器 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): 10 位地址的第一个字节未被任何从确认。</li> <li>■ 0x0 (INACTIVE): 不产生此中止。</li> </ul> 复位值：0x0 存在：一直 易失：是
0	ABRT_7B_ADDR_NOACK	R	该字段表示主处于 7 位寻址模式，并且发送的地址未被任何从确认。 DW_apb_i2c 的作用：主发送器或主接收器 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): 因为 7 位地址未被任何从确认，生成此中止。</li> <li>■ 0x0 (INACTIVE): 不生成此中止。</li> </ul> 存在：一直 易失：是

### 7.5.2.34 IC\_DMA\_CR

- 寄存器名：DMA 控制寄存器

- 描述：DMA 控制寄存器

该寄存器仅在 DW\_apb\_i2c 配置有一组 DMA 控制器接口信号 (IC\_HAS\_DMA = 1) 时才有效。当 DW\_apb\_i2c 未配置为 DMA 操作时，该寄存器不存在，写入寄存器地址无效，从该寄存器地址读取将返回零。该寄存器用于启用 DMA 控制器接口操作。发送和接收有一个单独的位。无论 IC\_ENABLE 的状态如何，都可以对其进行编程。

- 大小：32 bits
- 偏移量：0x88
- 存在：一直

位段	名称	类型	说明
31:2	RSVD_IC_DMA_CR_2_31	R	RSVD_IC_DMA_CR_2_31 保留位 - 只读 存在：一直
1	TDMAE	R/W	发送 DMA 启用。该位启用/禁用发送 FIFO DMA 通道。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用发送 FIFO DMA 通道。</li> <li>■ 0x0 (DISABLED): 禁用发送 FIFO DMA 通道。</li> </ul> 复位值：0x0 存在：一直
0	RDMAE	R/W	接收 DMA 启用。该位启用/禁用接收 FIFO DMA 通道。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用接收 FIFO DMA 通道。</li> <li>■ 0x0 (DISABLED): 禁用接收 FIFO DMA 通道。</li> </ul> 复位值：0x0 存在：一直

### 7.5.2.35 IC\_DMA\_TDLR

- 寄存器名：DMA 发送数据水平寄存器
- 描述：DMA 发送数据水平寄存器

该寄存器仅在 DW\_apb\_i2c 配置有一组 DMA 接口信号 (IC\_HAS\_DMA = 1) 时才有效。当 DW\_apb\_i2c 未配置为 DMA 操作时，该寄存器不存在；写入寄存器地址无效，从该寄存器地址读取返回零。

- 大小：32 bits
- 偏移量：0x8c
- 存在：一直

位段	名称	类型	说明
31:4	RSVD_DMA_TDLR	R	DMA_TDLR 保留位 - 只读 存在：一直
3:0	DMATDL	R/W	发送数据水平。该位段控制发送逻辑发出 DMA 请求的水平。等于水线水平；也就是说，当发送 FIFO 中的有效数据条目数等于或低于该字段值，并且 TDMAE = 1 时，会生成 dma_tx_req 信号。 复位值：0x0 存在：一直

### 7.5.2.36 IC\_DMA\_RDLR

- 寄存器名：DMA 接收数据水平寄存器

- 描述：DMA 接收数据水平寄存器

该寄存器仅在 DW\_apb\_i2c 配置有一组 DMA 接口信号(IC\_HAS\_DMA = 1)时有效。当 DW\_apb\_i2c 未配置为 DMA 操作时，该寄存器不存在；写入寄存器地址无效，从该寄存器地址读取返回零。

- 大小：32 bits
- 偏移量：0x90
- 存在：一直

位段	名称	类型	说明
31:4	RSVD_DMA_RDLR	R	DMA_RDLR 保留位 - 只读 存在：一直
3:0	DMARDL	R/W	接收数据水平。该位段控制接收逻辑发出 DMA 请求的水平。水线水平=DMARDL+1；也就是说，当接收 FIFO 中的有效数据条目数等于或大于该字段值+1，并且 RDMAE = 1 时，会生成 dma_rx_req 信号。例如，当 DMARD 为 0 时，当接收 FIFO 中存在一个或多个数据条目时，dma_rx_req 生效。 复位值：0x0 存在：一直

### 7.5.2.37 IC\_SDA\_SETUP

- 寄存器名：I2C SDA 设置寄存器
- 描述：I2C SDA 设置寄存器

当 DW\_apb\_i2c 在从发送操作中为读取请求提供服务时，该寄存器控制在 SCL 的上升沿引入的时间延迟量（以 ic\_clk 时钟周期数表示）- 相对于 SDA 的变化。相关的 I2C 要求是 tSU:DAT（注 4），详见 I2C 总线规范。该寄存器必须编程为等于或大于 2 的值。仅当 IC\_ENABLE[0] = 0 时，写入该寄存器才会成功。

注：设置时间的长度是使用  $[(IC\_SDA\_SETUP - 1) * (ic\_clk\_period)]$  计算的，因此如果用户需要 10 个 ic\_clk 设置时间周期，应该将值编程为 11。IC\_SDA\_SETUP 寄存器仅在 DW\_apb\_i2c 作为从发送器工作时使用。

- 大小：32 bits
- 偏移量：0x94
- 存在：一直

位段	名称	类型	说明
31:8	RSVD_IC_SDA_SETUP	R	IC_SDA_SETUP 保留位 - 只读 存在：一直
7:0	SDA_SETUP	R/W	SDA 设置。建议如果所需延迟为 1000ns，那么对于 10MHz 的 ic_clk 频率，IC_SDA_SETUP 应编程为值 11。 IC_SDA_SETUP 必须使用最小值 2 进行编程。 复位值：0x64 存在：一直

### 7.5.2.38 IC\_ACK\_GENERAL\_CALL

- 寄存器名：I2C ACK 通用广播寄存器
- 描述：I2C ACK 通用广播寄存器  
该寄存器控制 DW\_apb\_i2c 在收到 I2C 通用广播地址时是响应 ACK 还是 NACK。  
该寄存器仅在 DW\_apb\_i2c 处于从模式时适用。
- 大小：32 bits
- 偏移量：0x98
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_IC_ACK_GEN_1_31	R	RSVD_IC_ACK_GEN_1_31 保留位 - 只读 存在：一直
0	ACK_GEN_CALL	R/W	ACK 通用广播。当设置为 1 时，DW_apb_i2c 在收到通用广播时以 ACK 响应（通过使 ic_data_oe 生效）。否则，DW_apb_i2c 以 NACK 响应（通过使 ic_data_oe 无效）。

位段	名称	类型	说明
			值： ■ 0x1 (ENABLED): 为通用广播生成 ACK。 ■ 0x0 (DISABLED): 为通用广播生成 NACK。 复位值: 0x1 存在: 一直

### 7.5.2.39 IC\_ENABLE\_STATUS

- 寄存器名: I2C 使能状态寄存器

- 描述: I2C 使能状态寄存器

该寄存器用于在 IC\_ENABLE[0]寄存器从 1 设置为 0 时上报 DW\_apb\_i2c 硬件状态; 也就是说, 当 DW\_apb\_i2c 被禁用时。

如果 IC\_ENABLE[0]已设置为 1, bits 2:1 被强制为 0, bit 0 被强制为 1。

如果 IC\_ENABLE[0]已设置为 0, bits 2:1 仅在 bit 0 被读取为 0 时才有效。

注: 当 IC\_ENABLE[0]已设置为 0 时, 由于禁用 DW\_apb\_i2c 取决于 I2C 总线活动, 因此会延迟将 bit 0 读为 0。

- 大小: 32 bits
- 偏移量: 0x9c
- 存在: 一直

位段	名称	类型	说明
31:3	RSVD_IC_ENABLE_STATUS	R	IC_ENABLE_STATUS 保留位 - 只读 存在: 一直 易失: 是
2	SLV_RX_DATA_LOST	R	从收到的数据丢弃。该位指示是否由于将 IC_ENABLE 的 bit 0 从 1 设置为 0, 从 I2C 传输接收到至少一个数据字节时从接收器操作已中止。读为 1 时, 认为 DW_apb_i2c 已主动参与了一个中断的 I2C 传输 (地址匹配) 并且进入了 I2 传输的数据阶段, 即使一个数据字节已经用 NACK 响应。 注: 如果远程 I2C 主在 DW_apb_i2c 有机会 NACK 传输之前以 STOP 条件中止传输, 并且 IC_ENABLE[0]已设置为 0, 该位也设置为 1。 读取为 0 时, DW_apb_i2c 被认为已禁用, 而没有主动参与到从接收器传输的数据阶段。 注: 当 IC_EN (bit 0) 读为 0 时, CPU 可以安全读取该位。



位段	名称	类型	说明
			值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): 从收到的数据丢失。</li> <li>■ 0x0 (INACTIVE): 从收到的数据未丢失。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
1	SLV_DISABLED_WHILE_BUSY	R	忙时禁用从 (发送、接收)。该位指示是否由于将 IC_ENABLE 寄存器的位 0 从 1 设置为 0 而终止了潜在的或活动的从操作。当 CPU 将 0 写入 IC_ENABLE 寄存器时, 该位被设置, 同时: <p>DW_apb_i2c is 正从远程主接收从发送器操作的地址字节; 或者,</p> <p>来自远程主的从接收器操作的地址和数据字节。</p> <p>读为 1 时, 无论 I2C 地址是否与 DW_apb_i2c (IC_SAR 寄存器) 中设置的从地址匹配, 或者是否在 IC_ENABLE 设置为 0 但尚未生效之前完成传输, 都认为 DW_apb_i2c 在 I2C 传输的任何部分都强制执行 NACK。</p> <p>注: 如果远程 I2C 主在 DW_apb_i2c 有机会 NACK 传输之前以 STOP 条件中止传输, 并且 IC_ENABLE[0]已设置为 0, 该位也将设置为 1。</p> <p>读为 0 时, 当主处于激活状态或 I2C 总线空闲时, DW_apb_i2c 被视为已禁用。</p> <p>注: 当 IC_EN (bit 0) 读为 0 时, CPU 可以安全地读取该位。</p> 值： <ul style="list-style-type: none"> <li>■ 0x1 (ACTIVE): 从处于活动状态时被禁用。</li> <li>■ 0x0 (INACTIVE): 从空闲时被禁用。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是
0	IC_EN	R	ic_en 状态。该位始终反映在输出端口 ic_en 上驱动的值。 <ul style="list-style-type: none"> <li>■ 当读为 1 时, DW_apb_i2c 被视为处于启用状态。</li> <li>■ 当读为 0 时, DW_apb_i2c 被视为完全不活动。</li> </ul> <p>注: CPU 可以随时安全读取该位。当该位被读为 0 时, CPU</p>

位段	名称	类型	说明
			可以安全地读取 SLV_RX_DATA_LOST ( bit 2 ) 和 SLV_DISABLED_WHILE_BUSY ( bit 1 ) 。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用 I2C。</li> <li>■ 0x0 (DISABLED): 禁用 I2C。</li> </ul> 复位值: 0x0 存在: 一直 易失: 是

### 7.5.2.40 IC\_FS\_SPKLEN

- 寄存器名: I2C SS、FS 或 FM+尖峰抑制限制
- 描述: I2C SS、FS 或 FM+尖峰抑制限制

当组件在 SS、FS 或 FM+模式下运行时, 该寄存器用于存储由尖峰抑制逻辑滤除的最长尖峰的持续时间, 单位为 ic\_clk 周期。相关的 I2C 要求是 tSP (表 4), 详见 I2C 总线规范。该寄存器必须被编程为最小值 1。

- 大小: 32 bits
- 偏移量: 0xa0
- 存在: 一直

位段	名称	类型	说明
31:8	RSVD_IC_FS_SPKLEN	R	IC_FS_SPKLEN 保留位 - 只读 存在: 一直
7:0	IC_FS_SPKLEN	R/W	必须在任何 I2C 总线事务发生之前设置该寄存器以确保稳定运行。该寄存器设置 SCL 或 SD 线路中将被尖峰抑制逻辑滤除的最长尖峰的持续时间, 以 ic_clk 周期为单位。 该寄存器只能在 I2C 接口被禁用时写入, 对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。最小有效值为 1; 硬件会阻止写入小于此值的值, 如果尝试会导致设置为 1。更多信息, 请参阅“尖峰抑制”。 复位值: 0x1 存在: 一直

### 7.5.2.41 IC\_HS\_SPKLEN

- 寄存器名：I2C 高速尖峰抑制限制寄存器
- 描述：I2C 高速尖峰抑制限制寄存器

该寄存器用于存储组件在高速模式下运行时由尖峰抑制逻辑滤除的最长尖峰的持续时间，以 ic\_clk 周期为单位。相关的 I2C 要求是 tSP，详见 I2C 总线规范。该寄存器必须编程为最小值 1，并且仅在组件配置为支持高速模式时实现；也就是说，如果 IC\_MAX\_SPEED\_MODE 参数设置为 3。

- 大小：32 bits
- 偏移量：0xa4
- 存在：一直

位段	名称	类型	说明
31:8	RSVD_IC_HS_SPKLEN	R	IC_HS_SPKLEN 保留位 - 只读 存在：一直
7:0	IC_HS_SPKLEN	R/W	必须在任何 I2C 总线事务发生之前设置该寄存器以确保稳定运行。该寄存器设置 SCL 或 SD 线路中将被尖峰抑制逻辑滤除的最长尖峰的持续时间，以 ic_clk 周期为单位。更多信息，请参阅“尖峰抑制”。  该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。  最小有效值为 1；硬件会阻止写入小于此值的值，如果尝试会导致设置为 1。  复位值：0x1 存在：一直

### 7.5.2.42 IC\_SCL\_STUCK\_AT\_LOW\_TIMEOUT

- 寄存器名：I2C SCL 卡在低超时寄存器
- 描述：I2C SCL 卡在低超时寄存器

该寄存器用于存储持续时间，以 ic\_clk 周期为单位。如果 SCL 在 IC\_SCL\_STUCK\_LOW\_TIMEOUT 持续时间内保持低电平，则用于生成中断（SCL\_STUCK\_AT\_LOW）。

- 大小：32 bits
- 偏移量：0xac
- 存在：一直

位段	名称	类型	说明
31:0	IC_SCL_STUCK_LOW_TIMEOUT	R/W	如果 DW_apb_i2c 在 IC_SCL_STUCK_LOW_TIMEOUT 持续

位段	名称	类型	说明
			时间内（以 ic_clk 周期为单位）检测到 SCL 卡在低电平，则生成中断以指示 SCL 卡在低电平（SCL_STUCK_AT_LOW）。该寄存器只能在 I2C 接口被禁用时写入，对应于 IC_ENABLE[0]寄存器设置为 0。在其他时间写入无效。 复位值：0xFFFFFFFF 存在：一直

### 7.5.2.43 IC\_SDA\_STUCK\_AT\_LOW\_TIMEOUT

- 寄存器名：I2C SDA 卡在低超时寄存器

- 描述：I2C SDA 卡在低超时寄存器

该寄存器用于存储持续时间，以 ic\_clk 周期为单位。如果 SDA 在上述持续时间内保持低电平，则用于发送 SCL 脉冲来恢复 SDA 线。

- 大小：32 bits
- 偏移量：0xb0
- 存在：一直

位段	名称	类型	说明
31:0	IC_SDA_STUCK_LOW_TIMEOUT	R/W	如果 DW_apb_i2c 在 IC_SDA_STUCK_LOW_TIMEOUT 持续时间内（以 ic_clk 周期为单位）检测到 SDA 卡在低电平，则通过启用 DA_STUCK_RECOVERY_EN (IC_ENABLE[3]) 寄存器位来启动 SDA 线的恢复。 复位值：0xFFFFFFFF 存在：一直

### 7.5.2.44 IC\_CLR\_SCL\_STUCK\_DET

- 寄存器名：清除 SCL 卡在低检测中断寄存器

- 描述：清除 SCL 卡在低检测中断寄存器

- 大小：32 bits
- 偏移量：0xb4
- 存在：一直

位段	名称	类型	说明
31:1	RSVD_CLR_SCL_STUCK_DET	R	CLR_SCL_STUCK_DET 保留位 - 只读 存在：一直

位段	名称	类型	说明
			易失：是
0	CLR_SCL_STUCK_DET	R	读取该寄存器以清除 IC_RAW_INTR_STAT 寄存器的 SCL_STUCT_AT_LOW 中断（bit 15）。  复位值：0x0  存在：一直  易失：是

### 7.5.2.45 REG\_TIMEOUT\_RST

- 寄存器名：寄存器超时计数器复位值
- 描述：该寄存器保存寄存器定时器计数器的超时值。寄存器的复位值为 REG\_TIMEOUT\_VALUE。如果 HC\_REG\_TIMEOUT\_VALUE = 0，则可以进一步修改默认复位值。最终编程值（如果未编程，则为默认复位值）确定寄存器超时计数器从哪个值开始倒计时。如果此计数器为零，那么在 PSLVERR 为高的情况下，将中断等待的事务。
- 大小：32 bits
- 偏移量：0xf0
- 存在：一直

位段	名称	类型	说明
31:4	RSVD_REG_TIMEOUT_RST	R	保留位 - 只读  存在：一直  易失：是
3:0	REG_TIMEOUT_RST_ro	R	该字段保存 REG_TIMEOUT 计数器寄存器的复位值。  复位值：REG_TIMEOUT_VALUE  存在：一直  易失：是

### 7.5.2.46 IC\_COMP\_PARAM\_1

- 寄存器名：组件参数寄存器 1
  - 描述：组件参数寄存器 1
- 注：这是一个常量只读寄存器，包含有关组件参数设置的编码信息。复位值取决于 coreConsultant 参数。
- 大小：32 bits

- 偏移量：0xf4
- 存在：一直

位段	名称	类型	说明
31:24	RSVD_IC_COMP_PARAM_1	R	IC_COMP_PARAM_1 保留位 - 只读 存在：一直
23:16	TX_BUFFER_DEPTH	R	该寄存器的值来自 IC_TX_BUFFER_DEPTH coreConsultant 参数。 <ul style="list-style-type: none"> <li>■ 0x00: 保留</li> <li>■ 0x01: 2</li> <li>■ 0x02: 3</li> <li>■ ...</li> <li>■ 0xFF: 256</li> </ul> 复位值：0xF 存在：一直
15:8	RX_BUFFER_DEPTH	R	该寄存器的值来自 IC_RX_BUFFER_DEPTH coreConsultant 参数。 <ul style="list-style-type: none"> <li>■ 0x00: 保留</li> <li>■ 0x01: 2</li> <li>■ 0x02: 3</li> <li>■ ...</li> <li>■ 0xFF: 256</li> </ul> 复位值：0xF 存在：一直
7	ADD_ENCODED_PARAMS	R	该寄存器的值来自 IC_ADD_ENCODED_PARAMS coreConsultant 参数。 该位读 1 表示已包含通过软件读取这些编码参数的功能。 否则, 无论该位中编码的任何其他参数的设置如何, 整个寄存器都为 0。 值: <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用读取编码参数的功能。</li> <li>■ 0x0 (DISBALED): 禁用读取编码参数的功能。</li> </ul> 复位值：0x1

位段	名称	类型	说明
			存在：一直
6	HAS_DMA	R	该寄存器的值来自 IC_HAS_DMA coreConsultant 参数。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 启用 DMA 握手信号。</li> <li>■ 0x0 (DISABLED): 禁用 DMA 握手信号。</li> </ul> 复位值：0x1 存在：一直
5	INTR_IO	R	该寄存器的值来自 IC_INTR_IO coreConsultant 参数。 值： <ul style="list-style-type: none"> <li>■ 0x1 (COMBINED): 组合中断输出</li> <li>■ 0x0 (INDIVIDUAL): 单个中断输出</li> </ul> 复位值：0x1 存在：一直
4	HC_COUNT_VALUES	R	该寄存器的值来自 IC_HC_COUNT_VALUES coreConsultant 参数。 值： <ul style="list-style-type: none"> <li>■ 0x1 (ENABLED): 硬编码每种模式的计数值。</li> <li>■ 0x0 (DISABLED): 每种模式的可编程计数值。</li> </ul> 复位值：0x0 存在：一直
3:2	MAX_SPEED_MODE	R	该寄存器的值来自 IC_MAX_SPEED_MODE coreConsultant 参数。 值： <ul style="list-style-type: none"> <li>■ 0x0: 保留</li> <li>■ 0x1: 标准</li> <li>■ 0x2: 快速</li> <li>■ 0x3: 高速</li> </ul> 值： <ul style="list-style-type: none"> <li>■ 0x1 (STANDARD): IC MAX SPEED 为标准模式。</li> <li>■ 0x2 (FAST): IC MAX SPEED 为快速模式。</li> <li>■ 0x3 (HIGH): IC MAX SPEED 为高速模式。</li> </ul> 复位值：0x3

位段	名称	类型	说明
			存在：一直
1:0	APB_DATA_WIDTH	R	该寄存器的值来自 APB_DATA_WIDTH coreConsultant 参数。 值： <ul style="list-style-type: none"> <li>0x0 (APB_08BITS): APB 数据总线宽度为 8 位。</li> <li>0x1 (APB_16BITS): APB 数据总线宽度为 16 位。</li> <li>0x2 (APB_32BITS): APB 数据总线宽度为 32 位。</li> <li>0x3 (RESERVED): 保留位</li> </ul> 复位值：0x2 存在：一直

### 7.5.2.47 IC\_COMP\_VERSION

- 寄存器名：I2C 组件版本寄存器
- 描述：I2C 组件版本寄存器
- 大小：32 bits
- 偏移量：0xf8
- 存在：一直

位段	名称	类型	说明
31:0	IC_COMP_VERSION	R	DW_apb_i2c 发布说明中的发布表中描述了该寄存器的具体值。 复位值：0x3230322A 存在：一直

### 7.5.2.48 IC\_COMP\_TYPE

- 寄存器名：I2C 组件类型寄存器
- 描述：I2C 组件类型寄存器
- 大小：32 bits
- 偏移量：0xfc
- 存在：一直

位段	名称	类型	说明
31:0	IC_COMP_TYPE	R	Designware 组件类型编号 = 0x44_57_01_40。这个分配的



位段	名称	类型	说明
			唯一十六进制值是恒定的, 并且来自两个 ASCII 字母 “DW”, 后跟一个 16 位无符号数。 复位值: 0x44570140 存在: 一直

## 8 UART

### 8.1 概述

UART 是仿照行业标准 16550 设计的。用于与外部设备进行串行通信。数据通过 APB 总线从主机设备（CPU）写入 UART，转换为串行格式并传输到目标设备。UART 也接收串行数据并将其存储起来以供主机（CPU）回读。

UART 包含控制字符长度、波特率、奇偶校验生成/检查和中断生成的寄存器。有几种类型的 UART 优先级中断可以导致 UART 被设置，进而生成中断。控制寄存器单独启用/禁用每种中断类型。

为了减少 UART 对主机所需的时间，FIFO 可用于缓冲发送和接收数据。主机不必在每次收到一个字节的數據时访问 UART。

UART 可以通过专用通道连接到 DMA 控制器，以指示何时可以读取数据或何时发送 FIFO 为空。

与 16750 兼容的自动流控制模式提高了系统效率并减少了软件负载。当启用 FIFO 和自动流控制时，串行数据流由请求发送输出（rts\_n）和清除发送输入（cts\_n）自动控制。

### 8.2 主要特性

UART 支持以下特性：

- 曳影 1520 芯片有 8 个 UART 接口模块，UART0~5、AUDUART 和 AOUART
- 9 位串行数据支持
- 假起始位检测信号
- 可编程的小数波特率支持
- 基于 16550 行业标准的功能
- 多点 RS485 接口支持
- IrDA 1.0 SIR 模式支持高达 115.2Kbaud 的 UART3 和 AOUART 数据
- 16750 标准中规定的 UART 自动流控制模式

### 8.3 接口

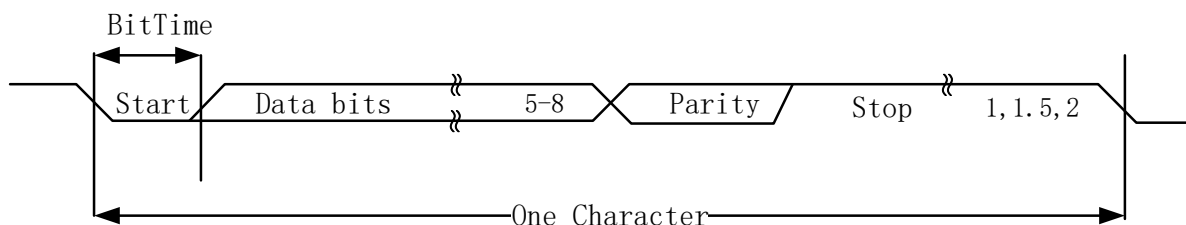
图表 8-1 管脚描述

管脚名称	方向	宽度	说明
AOUART_RXD	IO	1	串行数据接收信号
AOUART_TXD	IO	1	串行数据发送信号

管脚名称	方向	宽度	说明
AOUART_IR_IN	IO	1	SIR 模式下的串行数据接收信号
AOUART_IR_OUT	IO	1	SIR 模式下的串行数据发送信号
UART[0~5]_RXD	IO	1	串行数据接收信号
UART[0~5]_TXD	IO	1	串行数据发送信号
UART3_IR_IN	IO	1	SIR 模式下的串行数据接收信号
UART3_IR_OUT	IO	1	SIR 模式下的串行数据发送信号
UART4_CTSN	IO	1	流控制信号
UART4_RTSN	IO	1	流控制信号

## 8.4 功能描述

由于 UART 和选定设备之间的串行通信是异步的，因此在串行数据中添加了额外的位（开始和停止）以指示开始和结束。利用这些位允许两个设备同步。这种带有开始和停止位的串行数据结构称为字符，如图表 8-2 所示。



图表 8-2 UART 串行数据格式

可以在串行字符中增加一个额外的奇偶校验位。该位出现在字符结构中最后一个数据位之后和停止位之前，以便 UART 能够对接收到的数据执行简单的错误检查。

当中断发生时，主访问 IIR 寄存器。可以使用 IER 寄存器启用以下中断类型：

- 接收错误
- 接收数据可用
- 字符超时（仅在 FIFO 模式下）
- 发送保持寄存器在等于或低于阈值时空（在可编程 THRE 中断模式下）
- 调制解调器状态
- 忙检测指示

## 8.5 使用

RS485 模式下的编程流程

- 全双工模式 (XFER\_MODE = 0)
  1. 对 TCR 寄存器进行编程, 设置 XFER\_MODE(0)、DE\_POL (DE 信号的极性) 和 RE\_POL (RE 信号的极性)。
  2. 在 DET 寄存器中编程 DE 启用和禁用时序。
  3. 对 RE\_EN 和 DE\_EN 寄存器进行编程, 分别启用 RE 信号和 DE 信号。
  4. 执行数据发送和接收。
  5. 对 RE\_EN 进行编程, 禁用 RE 信号。
  6. DE 信号基于发送 FIFO 为空而被禁用。如果不想继续发送, 请将 DE\_EN 编程为 0。
- 软件启用的半双工模式 (XFER\_MODE = 1)
  1. 对 TCR 寄存器进行编程, 设置 XFER\_MODE(1)、DE\_POL (DE 信号的极性) 和 RE\_POL (RE 信号的极性)。
  2. 在 DET 寄存器中编程 DE 启用和禁用时序。
  3. 在 TAT 寄存器中编程周转时间。
  4. 对 RE\_EN 和 DE\_EN 寄存器进行编程, 分别启用 RE 信号和 DE 信号。
  5. 执行数据发送/接收。
  6. 对 RE\_EN 寄存器进行编程, 禁用 RE 信号。
  7. 将 DE\_EN 寄存器编程为 0, 然后将 RE\_EN 编程为 1。
- 硬件启用的半双工模式 (XFER\_MODE = 2)
  1. 对 TCR 寄存器进行编程, 设置 XFER\_MODE(2)、DE\_POL (DE 信号的极性) 和 RE\_POL (RE 信号的极性)。
  2. 在 DET 寄存器中编程 DE 启用和禁用时序。
  3. 在 TAT 寄存器中编程周转时间。
  4. 对 RE\_EN 和 DE\_EN 进行编程, 启用发送和接收通路。
  5. 执行数据发送/接收。
  6. 一旦 RE\_EN 和 DE\_EN 被编程为 1, 默认情况下, RE 信号被启用, DE 信号被禁用。当软件将数据推入发送 FIFO 时, 如果没有正在进行的接收传输, RE 信号被禁用, 然后 DE 信号被启用直到发送 FIFO 中有数据要发送。
  7. RE\_EN 和 DE\_EN 仍然作为软件覆盖, 决定何时关闭发送和接收路径。

## 8.6 寄存器描述

### 8.6.1 寄存器内存映射

寄存器	偏移量	说明	章节/页码
RBR	0x0	接收缓冲寄存器	<a href="#">8.6.2.1/306</a>
DLL	0x0	分频数锁存器（低）	<a href="#">8.6.2.2/307</a>
THR	0x0	发送保持寄存器	<a href="#">8.6.2.3/308</a>
DLH	0x4	分频数锁存器（DLH）高寄存器	<a href="#">8.6.2.4/308</a>
IER	0x4	中断使能寄存器	<a href="#">8.6.2.5/309</a>
FCR	0x8	FIFO 控制寄存器	<a href="#">8.6.2.6/311</a>
IIR	0x8	中断识别寄存器	<a href="#">8.6.2.7/313</a>
LCR	0xc	线路控制寄存器	<a href="#">8.6.2.8/314</a>
MCR	0x10	调制解调器控制寄存器	<a href="#">8.6.2.9/316</a>
LSR	0x14	线路状态寄存器	<a href="#">8.6.2.10/319</a>
MSR	0x18	调制解调器状态寄存器	<a href="#">8.6.2.11/323</a>
SCR	0x1c	暂存器寄存器	<a href="#">8.6.2.12/326</a>
LPDLL	0x20	低功耗分频数锁存器低寄存器	<a href="#">8.6.2.13/326</a>
LPDLH	0x24	低功耗分频数锁存器高寄存器	<a href="#">8.6.2.14/327</a>
SRBRn (for n = 0; n <= 15)	0x30 + n*0x4	影子接收缓冲寄存器	<a href="#">8.6.2.15/328</a>
STHRn (for n = 0; n <= 15)	0x30 + n*0x4	影子发送保持寄存器	<a href="#">8.6.2.16/328</a>
FAR	0x70	FIFO 访问寄存器	<a href="#">8.6.2.17/329</a>
TFR	0x74	发送 FIFO 读	<a href="#">8.6.2.18/329</a>
RFW	0x78	接收 FIFO 写	<a href="#">8.6.2.19/330</a>
USR	0x7c	UART 状态寄存器	<a href="#">8.6.2.20/331</a>
TFL	0x80	发送 FIFO 水平	<a href="#">8.6.2.21/333</a>
RFL	0x84	接收 FIFO 水平	<a href="#">8.6.2.22/333</a>

寄存器	偏移量	说明	章节/页码
SRR	0x88	软件复位寄存器	<a href="#">8.6.2.23/333</a>
SRTS	0x8c	影子发送请求	<a href="#">8.6.2.24/334</a>
SBCR	0x90	影子中断控制寄存器	<a href="#">8.6.2.25/335</a>
SDMAM	0x94	影子 DMA 模式寄存器	<a href="#">8.6.2.26/336</a>
SFE	0x98	影子 FIFO 使能寄存器	<a href="#">8.6.2.27/336</a>
SRT	0x9c	影子 RCVR 触发寄存器	<a href="#">8.6.2.28/337</a>
STET	0xa0	影子发送空触发寄存器	<a href="#">8.6.2.29/337</a>
HTX	0xa4	中止发送	<a href="#">8.6.2.30/338</a>
DMA SA	0xa8	DMA 软件确认寄存器	<a href="#">8.6.2.31/338</a>
TCR	0xac	收发器控制寄存器	<a href="#">8.6.2.32/339</a>
DE_EN	0xb0	驱动输出使能寄存器 (DE_EN)	<a href="#">8.6.2.33/340</a>
RE_EN	0xb4	接收器输出使能寄存器 (RE_EN)	<a href="#">8.6.2.34/341</a>
DET	0xb8	驱动输出使能时间寄存器 (DET)	<a href="#">8.6.2.35/341</a>
TAT	0xbc	周转时间寄存器 (TAT)	<a href="#">8.6.2.36/342</a>
DLF	0xc0	除数锁存器小数寄存器	<a href="#">8.6.2.37/342</a>
RAR	0xc4	接收地址寄存器	<a href="#">8.6.2.38/343</a>
TAR	0xc8	发送地址寄存器	<a href="#">8.6.2.39/343</a>
LCR_EXT	0xcc	线路扩展控制寄存器	<a href="#">8.6.2.40/344</a>
CPR	0xf4	组件参数寄存器	<a href="#">8.6.2.41/345</a>
UCV	0xf8	UART 组件版本	<a href="#">8.6.2.42/348</a>
CTR	0xfc	组件类型寄存器	<a href="#">8.6.2.43/348</a>

## 8.6.2 寄存器和字段描述

### 8.6.2.1 RBR

- 寄存器名：接收缓冲寄存器
- 描述：接收缓冲寄存器。只有当 DLAB 位 (LCR[7]) 被清除时才能访问该寄存器。

- 偏移量：0x0

位段	名称	类型	说明
31:8	RSVD_RBR	R	保留位，读为零。 复位值：0x0
7:0	RBR	R	接收缓冲寄存器 该寄存器包含在 UART 模式下串行输入端口 (sin) 或红外模式下串行红外输入 (sir_in) 接收的数据字节。仅当线路状态寄存器 (LSR) 中的数据就绪 (DR) 位设置时，该寄存器中的数据才有效。 如果在非 FIFO 模式 (FIFO_MODE == NONE) 或 FIFO 被禁用 (FCR[0] 设置为 0)，必须在下一个数据到达之前读取 RBR 中的数据，否则将被覆盖，导致溢出错误。 如果在 FIFO 模式下 (FIFO_MODE != NONE) 并且启用了 FIFO (FCR[0] 设置为 1)，该寄存器访问接收 FIFO 的头部。如果接收 FIFO 已满且在下一个数据字符到达之前未读取该寄存器，FIFO 中的数据将被保留，但任何传入的数据都将丢失并发生溢出错误。 复位值：0x0

### 8.6.2.2 DLL

- 寄存器名：分频数锁存器（低）
- 偏移量：0x0

位段	名称	类型	说明
31:8	RSVD_DLL_31to8	R	保留位，读为零。
7:0	DLL	R/W	分频数锁存器（低） 该寄存器构成 16 位读/写分频数锁存寄存器的低 8 位，该寄存器包含 UART 的波特率分频数。 输出波特率等于串行时钟（如果一个时钟设计为 pclk，如果两个时钟设计为 sclk (CLOCK_MODE == Enabled)）频率除以波特分频数值的 16 倍，如下：波特率 = (串行时钟频率)/(16*分频数)。 注：当分频数锁存寄存器 (DLL 和 DLH) 设置为零时，波特时钟被禁用并且不会发生串行通信。此外，一旦设置了 DLL，在发送或接收数据之前，应允许最慢 UART 时钟的至少 8 个

位段	名称	类型	说明
			时钟周期通过。 复位值：0x0

### 8.6.2.3 THR

- 寄存器名：发送保持寄存器
- 描述：发送保持寄存器。只有当 DLAB 位（LCR[7]）被清除时才能访问该寄存器。
- 偏移量：0x0

位段	名称	类型	说明
32:9	RSVD_THR	R	保留位，读为零。
8:0	THR	W	<p>发送保持寄存器</p> <p>该寄存器包含要在 UART 模式下的串行输出端口（sout）或红外模式下的串行红外输出（sir_out_n）上发送的数据。只有设置了 THR 空（LSR[5]）位时，才应将数据写入 THR。</p> <p>如果在非 FIFO 模式或 FIFO 被禁用（FCR[0]设置为零）并且 THRE 被设置，将单个字符写入 THR 会清除 THRE。在再次设置 THRE 之前对 THR 的任何额外写入都会导致 THR 数据被覆盖。</p> <p>如果在 FIFO 模式下启用 FIFO（FCR[0]设置为 1）并且 THRE 被设置，x 个字符的数据可能会在 FIFO 满之前写入 THR。数字 x（默认 = 16）由配置期间设置的 FIFO 深度值确定。FIFO 已满时任何写入数据的尝试都会导致写入数据丢失。</p> <p>复位值：0x0</p>

### 8.6.2.4 DLH

- 寄存器名：分频数锁存器高
- 描述：分频数锁存器高（DLH）寄存器。如果 UART\_16550\_COMPATIBLE = No，只有在 DLAB 位（LCR[7]）被设置且 UART 空闲时才能访问该寄存器，即 USR[0]为 0；否则只有当 DLAB 位（LCR[7]）被设置时才能访问该寄存器。
- 偏移量：0x4

位段	名称	类型	说明
31:8	RSVD_DLH	R	保留位，读为零。



位段	名称	类型	说明
7:0	dlh	R/W	<p>该寄存器构成 16 位读/写分频数锁存寄存器的高 8 位，该寄存器包含 UART 的波特率分频数。</p> <p>输出波特率等于串行时钟（如果一个时钟设计为 pclk，如果两个时钟设计为 sclk（CLOCK_MODE == Enabled））频率除以波特率分频数值的 16 倍，如下：波特率 = (串行时钟频率)/(16*分频数)。</p> <p>注：当分频数锁存寄存器（DLL 和 DLH）设置为零时，波特时钟被禁用并且不会发生串行通信。此外，一旦设置了 DLH，在发送或接收数据之前，应允许最慢 UART 时钟的至少 8 个时钟周期通过。</p> <p>复位值：0x0</p>

### 8.6.2.5 IER

- 寄存器名：中断使能寄存器
- 描述：中断使能寄存器。只有当 DLAB 位（LCR[7]）被清除时才能访问该寄存器。
- 偏移量：0x4

位段	名称	类型	说明
31:8	RSVD_IER_31to8	R	保留位，读为零。
7	PTIME	RW	<p>可编程 THRE 中断模式使能。仅当 THRE_MODE_USER == Enabled 时可写，始终可读。用于启用/禁用 THRE 中断的生成。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用可编程 THRE 中断模式。</li> <li>■ 0x1 (ENABLED): 启用可编程 THRE 中断模式。</li> </ul> <p>复位值：0x0</p>
6:5	RSVD_IER_6to5	R	<p>保留位，读为零。</p> <p>复位值：0x0</p>
4	ELCOLR	RO	<p>中断使能寄存器：ELCOLR，该位控制清除 LSR 寄存器中状态的方法。这仅适用于溢出错误、奇偶校验错误、帧错误和中断中断状态位。0 = 在读取接收 FIFO（RBR 读）或读取 LSR 寄存器时清除 LSR 状态位。1 = LSR 状态位仅在读取 LSR 寄存器时被清除。仅当 LSR_STATUS_CLEAR == Enabled 时可写，始终可读。</p>

位段	名称	类型	说明
			值： ■ 0x0 (DISABLED): 禁用 ALC。 ■ 0x1 (ENABLED): 启用 ALC。 复位值: 0x0
3	EDSSI	R/W	启用调制解调器状态中断。这用于启用/禁用调制解调器状态中断的生成。 这是第四高优先级的中断。 值： ■ 0x0 (DISABLED): 禁用调制解调器状态中断。 ■ 0x1 (ENABLED): 启用调制解调器状态中断。 复位值: 0x0
2	ELSI	R/W	启用接收器线路状态中断。这用于启用/禁用接收器线路状态中断的生成。这是最高优先级的中断。 值： ■ 0x0 (DISABLED): 禁用接收器线路状态中断。 ■ 0x1 (ENABLED): 启用接收器线路状态中断。 复位值: 0x0
1	ETBEI	R/W	启用发送保持寄存器空中断。这用于启用/禁用发送器保持寄存器空中断的生成。这是第三高优先级的中断。 值： ■ 0x0 (DISABLED): 禁用发送空中断。 ■ 0x1 (ENABLED): 启用发送空中断。 复位值: 0x0
0	ERBFI	R/W	启用接收数据可用中断。这用于启用/禁用接收数据可用中断和字符超时中断（如果在 FIFO 模式下并且启用了 FIFO）的生成。这些是第二高优先级的中断。 值： ■ 0x0 (DISABLED): 禁用接收数据中断。 ■ 0x1 (ENABLED): 启用接收数据中断。 复位值: 0x0

## 8.6.2.6 FCR

- 寄存器名：FIFO 控制寄存器
- 偏移量：0x8

位段	名称	类型	说明
31:8	RSVD_FCR_31to8	R	保留位，读为零。
7:6	RT	W	<p>RCVR 触发（或 RT）。这用于选择接收器 FIFO 中将产生接收数据可用中断的触发水平。在自动流控制模式下，仅当禁用 RTC_FCT 时，它用于确定何时禁用 rts_n 信号。它还确定在某些操作模式下何时启用 dma_rx_req_n 信号。有关 DMA 支持的详细信息，请参阅数据手册的“DMA 支持”部分。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (FIFO_CHAR_1): FIFO 中有 1 个字符</li> <li>■ 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 满</li> <li>■ 0x2 (FIFO_HALF_FULL): FIFO 1/2 满</li> <li>■ 0x3 (FIFO_FULL_2): FIFO 差 2 不满</li> </ul> <p>复位值：0x0</p>
5:4	TET	W	<p>发送空触发（或 TET）。当 THRE_MODE_USER == Disabled 时，写入将无效。这用于选择当模式处于活动状态时将产生 THRE 中断的空阈值水平。它还确定在某些操作模式下何时启用 dma_tx_req_n。有关 DMA 支持的详细信息，请参阅数据手册的“DMA 支持”部分。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (FIFO_EMPTY): FIFO 为空</li> <li>■ 0x1 (FIFO_CHAR_2): FIFO 中有 2 个字符</li> <li>■ 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 满</li> <li>■ 0x3 (FIFO_HALF_FULL): FIFO 1/2 满</li> </ul> <p>复位值：0x0</p>

位段	名称	类型	说明
3	DMAM	W	<p>DMA 模式（或 DMAM）。当未选择额外的 DMA 握手信号时（DMA_EXTRA == NO），这决定了用于 dma_tx_req_n 和 dma_rx_req_n 输出信号的 DMA 信令模式。有关 DMA 支持的详细信息，请参阅数据手册的“DMA 支持”部分。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (MODE0)：模式 0</li> <li>■ 0x1 (MODE1)：模式 1</li> </ul> <p>复位值：0x0</p>
2	XFIFOR	W	<p>XMIT FIFO 复位（或 XFIFOR）。这会复位发送 FIFO 的控制部分并将 FIFO 视为空。当选择了额外的 DMA 握手信号时（DMA_EXTRA == YES），这也将禁用 DMA 发送请求和单个信号。注意，该位是自动清除的，无需清除该位。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (RESET)：发送 FIFO 复位</li> </ul> <p>复位值：0x0</p>
3	DMAM	W	<p>DMA 模式（或 DMAM）。当未选择额外的 DMA 握手信号时（DMA_EXTRA == NO），这决定了用于 dma_tx_req_n 和 dma_rx_req_n 输出信号的 DMA 信令模式。有关 DMA 支持的详细信息，请参阅数据手册的“DMA 支持”部分。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (MODE0)：模式 0</li> <li>■ 0x1 (MODE1)：模式 1</li> </ul> <p>复位值：0x0</p>
2	XFIFOR	W	<p>XMIT FIFO 复位（或 XFIFOR）。这会复位发送 FIFO 的控制部分并将 FIFO 视为空。当选择了额外的 DMA 握手信号时（DMA_EXTRA == YES），这也将禁用 DMA 发送请求和单个信号。注意，该位是自动清除的，无需清除该位。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (RESET)：发送 FIFO 复位</li> </ul> <p>复位值：0x0</p>
1	RFIFOR	W	<p>RCVR FIFO 复位（或 RFIFOR）。这会复位接收 FIFO 的控制部分并将 FIFO 视为空。当选择了额外的 DMA 握手信号时（DMA_EXTRA == YES），这也将禁用 DMA 接收请求和单个</p>

位段	名称	类型	说明
			信号。注意，该位是自动清除的，无需清除该位。 值： ■ 0x1 (RESET): 接收 FIFO 复位 复位值: 0x0
0	FIFOE	W	FIFO 使能(或 FIFOE)。启用/禁用发送(XMIT)或接收(RCVR) FIFO。每当该位的值改变时，FIFO 的发送和接收控制器部分都会被复位。 值： ■ 0x0 (DISABLED): 禁用 FIFO。 ■ 0x1 (ENABLED): 启用 FIFO。 复位值: 0x0

### 8.6.2.7 IIR

- 寄存器名：中断识别寄存器
- 描述：中断识别寄存器
- 偏移量：0x8

位段	名称	类型	说明
31:8	RSVD_IIR_31to8	R	保留位，读为零。
7:6	FIFOSE	R	FIFO 启用（或 FIFOSE）。指示 FIFO 是启用还是禁用。 值： ■ 0x0 (DISABLED): 禁用 FIFO。 ■ 0x3 (ENABLED): 启用 FIFO。 复位值: 0x0
5:4	RSVD_IIR_5to4	R	保留位，读为零。 复位值: 0x0
3:0	IID	R	中断 ID（或 IID）。这表示最高优先级的挂起中断，它可以是值中指定的以下类型之一。 注: 如果 coreConsultant 中的 UART_16550_COMPATIBLE == YES, 永远不会指示 0111 类型的中断（忙检测）。 bit3 表示只有在启用 FIFO 并用于区分字符超时条件中断时才会发生中断。

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> <li>0x0 (MODEM_STATUS): 调制解调器状态</li> <li>0x1 (NO_INTERRUPT_PENDING): 没有挂起的中断</li> <li>0x2 (THR_EMPTY): THR 为空</li> <li>0x4 (RECEIVED_DATA_AVAILABLE): 接收数据可用</li> <li>0x6 (RECEIVER_LINE_STATUS): 接收器线路状态</li> <li>0x7 (BUSY_DETECT): 忙检测</li> <li>0xc (CHARACTER_TIMEOUT): 字符超时</li> </ul> 复位值: 0x1

### 8.6.2.8 LCR

- 寄存器名: 线路控制寄存器
- 描述: 线路控制寄存器
- 偏移量: 0xc

位段	名称	类型	说明
31:8	RSVD_LCR_31to8	R	保留位, 读为零。
7	DLAB	R/W	分频数锁存器访问位  如果 UART_16550_COMPATIBLE == NO, 仅在 UART 空闲时 (USR[0]为 0) 可写, 否则始终可写并始终可读。该位用于启用分频数锁存器寄存器 (DLL、DLH/LPDLH 和 LPDLH) 的读写, 以设置 UART 的波特率。为了访问其他寄存器, 必须在初始波特率设置后清除该位。  值： <ul style="list-style-type: none"> <li>0x0 (DISABLED): 分频数锁存器寄存器仅在 UART 空闲时可写。</li> <li>0x1 (ENABLED): 分频数锁存器寄存器始终可读和可写。</li> </ul> 复位值: 0x0
6	BC	R/W	中断控制位  这用于使中断条件发送到接收设备。如果设置为 1, 串行输出被强制为间隔 (逻辑 0) 状态。当不处于环回模式时, 由 MCR[4] 确定, sout 线被强制为低电平, 直到中断位被清除。如果 SIR_MODE == Enabled 且处于活动状态 (MCR[6]设置为 1),

位段	名称	类型	说明
			<p>sir_out_n 线将连续脉冲。在环回模式下，中断条件在内部环回到接收器，并且 sir_out_n 被强制为低电平。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 释放串行输出以进行数据传输。</li> <li>■ 0x1 (ENABLED): 串行输出被强制为间隔状态。</li> </ul> <p>复位值：0x0</p>
5	SP	R/W	<p>Stick 奇偶校验</p> <p>如果 UART_16550_COMPATIBLE = NO，仅在 UART 空闲时 (USR[0]为 0) 可写；否则始终可写且始终可读。该位用于强制奇偶校验值。当 PEN、EPS 和 Stick 奇偶校验设置为 1 时，发送奇偶校验位并检查为逻辑 0。如果 PEN 和 Stick 奇偶校验设置为 1 且 EPS 为逻辑 0，发送奇偶校验位并检查为逻辑 1。如果该位设置为 0，禁用 Stick 奇偶校验。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 Stick 奇偶校验。</li> <li>■ 0x1 (ENABLED): 启用 Stick 奇偶校验。</li> </ul> <p>复位值：0x0</p>
4	EPS	R/W	<p>奇偶校验选择</p> <p>如果 UART_16550_COMPATIBLE == NO，仅在 UART 空闲时 (USR[0]为 0) 可写，否则始终可写且始终可读。当启用奇偶校验时 (PEN 设置为 1)，用于在偶数奇偶校验和奇数奇偶校验之间进行选择。如果设置为 1，发送或检查偶数个逻辑 1。如果设置为 0，发送或检查奇数个逻辑 1。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (ODD_PARITY): 发送或检查奇校验。</li> <li>■ 0x1 (EVEN_PARITY): 发送或检查偶校验。</li> </ul> <p>复位值：0x0</p>
3	PEN	R/W	<p>奇偶校验使能。</p> <p>如果 UART_16550_COMPATIBLE == NO，仅在 UART 空闲时 (USR[0]为 0) 可写，否则始终可写且始终可读。该位用于分别启用和禁用发送和接收串行字符中的奇偶校验生成和检测。</p> <p>值：</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用奇偶校验。</li> <li>0x1 (ENABLED): 启用奇偶校验。</li> </ul> 复位值: 0x0
2	STOP	R/W	停止位的数量 如果 UART_16550_COMPATIBLE == NO, 仅在 UART 空闲时 (USR[0]为 0) 可写, 否则始终可写且始终可读。用于选择外设将发送和接收的每个字符的停止位数。如果设置为 0, 串行数据中发送 1 个停止位。 如果设置为 1 且数据位设置为 5 (LCR[1:0]设置为 0), 发送 1.5 个停止位。否则, 发送 2 个停止位。注意, 无论选择多少停止位, 接收器只会检查第一个停止位。 值: <ul style="list-style-type: none"> <li>0x0 (STOP_1BIT): 1 个停止位</li> <li>0x1 (STOP_1_5BIT_OR_2BIT): DLS 时 1.5 个停止位</li> <li>(LCR[1:0])为 0, 否则为 2 个停止位</li> </ul> 复位值: 0x0
1:0	DLS	R/W	数据长度选择 (或旧版中使用的 CLS) 如果 UART_16550_COMPATIBLE == NO, 仅在 UART 空闲时 (USR[0]为 0) 可写, 否则始终可写且始终可读。当 LCR_EXT 中的 DLS_E 设置为 0 时, 该寄存器用于选择外设将发送和接收的每个字符的数据位数。 值: <ul style="list-style-type: none"> <li>0x0 (CHAR_5BITS): 每个字符 5 个数据位</li> <li>0x1 (CHAR_6BITS): 每个字符 6 个数据位</li> <li>0x2 (CHAR_7BITS): 每个字符 7 个数据位</li> <li>0x3 (CHAR_8BITS): 每个字符 8 个数据位</li> </ul> 复位值: 0x0

### 8.6.2.9 MCR

- 寄存器名: 调制解调器控制寄存器
- 描述: 调制解调器控制寄存器
- 偏移量: 0x10



位段	名称	类型	说明
31:7	RSVD_MCR_31to7	R	保留位，读为零。
6	SIRE	R/W	<p>SIR 模式使能</p> <p>仅当 SIR_MODE == Enabled 时可写，始终可读。用于启用/禁用 IrDA SIR 模式功能，如数据手册的“IrDA 1.0 SIR 协议”部分所述。</p> <p>注：要启用 SIR 模式，请在写入 LCR 寄存器之前将适当的值写入 MCR 寄存器。有关推荐的编程顺序的详细信息，请参阅数据手册的“编程示例”部分。</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 IrDA SIR 模式。</li> <li>■ 0x1 (ENABLED): 启用 IrDA SIR 模式。</li> </ul> <p>复位值：0x0</p>
5	AFCE	R/W	<p>自动流控制使能</p> <p>仅当 AFCE_MODE == Enabled 时可写，始终可读。当启用 FIFO 且设置自动流控制启用 (AFCE) 位时，将启用自动流控制功能，如数据手册的“自动流控制”部分所述。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用自动流控制模式。</li> <li>■ 0x1 (ENABLED): 启用自动流控制模式。</li> </ul> <p>复位值：0x0</p>
4	LoopBack	R/W	<p>环回位</p> <p>用于将 UART 置于诊断模式以进行测试。如果在 UART 模式下运行 (SIR_MODE != Enabled 或未激活，MCR[6] 设置为 0)，sout 线上的数据保持高电平，而串行数据输出在内部环回到 sin 线。在这种模式下，所有中断都可以正常工作。此外，在环回模式下，调制解调器控制输入 (dsr_n、cts_n、ri_n、dcd_n) 断开，调制解调器控制输出 (dtr_n、rts_n、out1_n、out2_n) 在内部环回到输入。</p> <p>如果在红外模式下运行 (SIR_MODE == Enabled 且激活，MCR[6] 设置为 1)，sir_out_n 线上的数据保持低电平，而串行数据输出被反转并环回到 sir_in 线上。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用环回模式。</li> <li>■ 0x1 (ENABLED): 启用环回模式。</li> </ul>

位段	名称	类型	说明
			复位值: 0x0
3	OUT2	R/W	<p>输出 2</p> <p>这用于直接控制用户指定的输出 2 (out2_n) 输出。写入该位置的值被反转并在 out2_n 上输出。注意, 在环回模式下 (MCR[4]设置为 1), out2_n 输出保持无效高电平, 而该位置的值在内部环回到输入。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (OUT2_0): out2_n 无效 (逻辑 1)。</li> <li>■ 0x1 (OUT2_1): out2_n 有效 (逻辑 0)。</li> </ul> <p>复位值: 0x0</p>
2	OUT1	R/W	<p>输出 1</p> <p>这用于直接控制用户指定的输出 1 (out1_n) 输出。写入该位置的值被反转并在 out1_n 上输出。注意, 在环回模式下 (MCR[4]设置为 1), out1_n 输出保持无效高电平, 而该位置的值在内部环回到输入。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (OUT1_0): out1_n 无效 (逻辑 1)。</li> <li>■ 0x1 (OUT1_1): out1_n 有效 (逻辑 0)。</li> </ul> <p>复位值: 0x0</p>
1	RTS	R/W	<p>请求发送</p> <p>这用于直接控制请求发送 (rts_n) 输出。请求发送 (rts_n) 输出用于控制调制解调器或数据集 UART 已准备好交换数据。当自动 RTS 流控制未启用时 (MCR[5]设置为 0), 通过将 MCR[1] (RTS) 编程为高电平, 将 rts_n 信号设置为低电平。在自动流控制中, AFCE_MODE == Enabled 和激活 (MCR[5]设置为 1) 并启用 FIFO (FCR[0]设置为 1), rts_n 输出以相同方式控制, 也由接收器 FIFO 阈值触发进行 (当高于阈值时 rts_n 为无效高电平)。当 MCR[1]设置为低电平时, rts_n 信号将无效。注意, 在环回模式下 (MCR[4]设置为 1), rts_n 输出保持无效高电平, 而该位置的值在内部环回到输入。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (INACTIVE): 请求发送 rts_n 无效 (逻辑 1)。</li> <li>■ 0x1 (ACTIVE): 请求发送 rts_n 有效 (逻辑 0)。</li> </ul>

位段	名称	类型	说明
			复位值: 0x0
0	DTR	R/W	<p>数据终端就绪</p> <p>这用于直接控制数据终端就绪 (dtr_n) 输出。写入该位置的值被反转并在 dtr_n 上输出。</p> <p>数据终端就绪输出用于通知调制解调器或数据集 UART 已准备好建立通信。注意, 在环回模式下 (MCR[4] 设置为 1), dtr_n 输出保持无效高电平, 而该位置的值在内部环回到输入。</p> <p>值:</p> <ul style="list-style-type: none"> <li>0x0 (INACTIVE): dtr_n 无效 (逻辑 1)。</li> <li>0x1 (ACTIVE): dtr_n 有效 (逻辑 0)。</li> </ul> <p>复位值: 0x0</p>

### 8.6.2.10 LSR

- 寄存器名: 线路状态寄存器
- 描述: 线路状态寄存器
- 偏移量: 0x14

位段	名称	类型	说明
31:9	RSVD_LSR_31to9	R	保留位, 读为零。
8	ADDR_RCVD	R	<p>地址接收位</p> <p>如果启用了 9Bit 数据模式 (LCR_EXT[0] = 1), 该位用于指示接收数据的 bit9 设置为 1。该位还可用于指示传入的字符是地址还是数据。</p> <ul style="list-style-type: none"> <li>1: 表示字符是地址。</li> <li>0: 表示字符是数据。</li> </ul> <p>在 FIFO 模式下, 由于 bit9 与收到的字符相关联, 因此当 bit9 设置为 1 的字符位于 FIFO 的顶部时, 它就会显示出来。</p> <p>读取 LSR 会清除 9BIT。</p> <p>注: 用户需要确保中断在下一个地址字节到达之前被清除 (读取 LSR 寄存器)。如果清除中断有延迟, 软件将无法区分多地址相关中断。</p> <p>复位值: 0x0</p>

位段	名称	类型	说明
7	RFE	R	<p>接收器 FIFO 错误位</p> <p>该位仅在 FIFO_MODE != NONE 且 FIFO 启用 (FCR[0]设置为 1) 时相关。这用于指示 FIFO 中是否存在至少一个奇偶校验错误、帧错误或中断指示。</p> <p>读取 LSR 时, 有错误的字符位于接收器 FIFO 的顶部且 FIFO 中没有后续错误, 该位被清除。</p> <p>值:</p> <ul style="list-style-type: none"> <li>0x0 (NO_RX_FIFO_ERROR): 接收 FIFO 中没有错误。</li> <li>0x1 (RX_FIFO_ERROR): 接收 FIFO 中有错误。</li> </ul> <p>复位值: 0x0</p>
6	TEMT	R	<p>发送器空位</p> <p>如果在 FIFO 模式 (FIFO_MODE != NONE) 且 FIFO 启用时 (FCR[0]设置为 1), 只要发送移位寄存器和 FIFO 都为空, 该位就会设置。如果在非 FIFO 模式或 FIFO 禁用时, 只要发送器保持寄存器和发送移位寄存器都为空, 该位就会设置。</p> <p>值:</p> <ul style="list-style-type: none"> <li>0x0 (DISABLED): 发送器不为空。</li> <li>0x1 (ENABLED): 发送器为空。</li> </ul> <p>复位值: 0x1</p>
5	THRE	R	<p>发送保持寄存器空位</p> <p>如果 THRE_MODE_USER = Disabled 或 THRE 模式禁用 (IER[7]设置为 0), 无论 FIFO 是否被实现/启用, 该位指示 THR 或 TX FIFO 为空。</p> <p>每当数据从 THR 或 TX FIFO 传输到发送移位寄存器且没有新数据写入 THR 或 TX FIFO 时, 该位就会设置。</p> <p>如果启用了 THRE 中断, 这也导致发生 THRE 中断。</p> <p>如果 THRE_MODE_USER == Enabled, FIFO_MODE != NONE 且两种模式都处于活动状态 (IER[7]和 FCR[0]分别设置为 1), 功能切换以指示发送 FIFO 已满, 且不再控制 THRE 中断, 然后由 FCR[5:4]阈值设置控制。可编程 THRE 中断模式操作的详细描述, 请参阅数据手册的“可编程 THRE 中断”部分。</p> <p>值:</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 THRE 中断控制。</li> <li>■ 0x1 (ENABLED): 启用 THRE 中断控制。</li> </ul> 复位值: 0x1
4	BI	R	<p>中断中断位</p> <p>这用于指示检测到串行输入数据上的中断序列。</p> <p>如果在 UART 模式下, 只要串行输入 sin 保持在逻辑 0 状态的时间超过开始时间 + 数据位 + 奇偶校验 + 停止位的总和, 就会设置它。</p> <p>如果在红外模式下, 只要串行输入 sir_in 连续脉冲到逻辑 0 的时间超过开始时间 + 数据位 + 奇偶校验 + 停止位的总和, 就会设置它。串行输入的中断条件会导致 UART 收到一个且只有一个字符, 该字符由全零组成。</p> <p>在 FIFO 模式下, 与中断条件相关的字符通过 FIFO 传送, 并在字符位于 FIFO 顶部时显示。</p> <p>读取 LSR 会清除 BI 位 (如果 LSR_STATUS_CLEAR == 1) 或读取 LSR 或 RBR 清除 BI 位 (如果 LSR_STATUS_CLEAR == 0)。</p> <p>在非 FIFO 模式下, BI 指示立即发生并持续到 LSR 被读取。</p> <p>注: 如果收到中断条件时 FIFO 已满, 会发生 FIFO 溢出。中断条件和与它相关的所有信息-奇偶校验和帧错误——被丢弃。任何收到中断字符的信息都会丢失。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (NO_BREAK): 没有检测到中断序列。</li> <li>■ 0x1 (BREAK): 检测到中断序列。</li> </ul> 复位值: 0x0
3	FE	R	<p>帧错误位</p> <p>这用于指示器中发生帧错误。当接收器在收到的数据中没有检测到有效的停止位时, 就会发生帧错误。</p> <p>在 FIFO 模式下, 由于帧错误与收到的字符相关, 因此当帧错误的字符位于 FIFO 的顶部时, 就会显示帧错误。当发生帧错误时, UART 将尝试重新同步。它通过假设错误是由于下一个字符的起始位来实现这一点, 然后继续收另一个位, 即数据和/或奇偶校验和停止。</p> <p>应该注意的是, 如果发生了中断中断, 帧错误位 (LSR[3]) 将被设置, 如中断中断位 (LSR[4]) 所示。发生这种情况时因为</p>

位段	名称	类型	说明
			<p>中断字符通过将 <code>sin</code> 输入保持为逻辑 0 的时间长于字符的持续时间，从而隐式生成了帧错误。</p> <p>读取 LSR 清除 FE 位（如果 <code>LSR_STATUS_CLEAR == 1</code>）或读取 LSR 或 RBR 清除 FE 位（如果 <code>LSR_STATUS_CLEAR == 0</code>）。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (NO_FRAMING_ERROR): 没有帧错误</li> <li>■ 0x1 (FRAMING_ERROR): 帧错误</li> </ul> <p>复位值：0x0</p>
2	PE	R	<p>奇偶校验错误位</p> <p>如果奇偶校验使能位（<code>LCR[3]</code>）被设置，这用于指示接收器中发生奇偶校验错误。</p> <p>在 FIFO 模式下，由于奇偶校验错误与收到的字符相关联，当有奇偶校验错误的字符到达 FIFO 的顶部时就会显示出来。</p> <p>应该注意的是，如果发生中断中断，奇偶校验错误位（<code>LSR[2]</code>）将被设置，如中断中断位（<code>LSR[4]</code>）所示。在这种情况下，如果启用奇偶校验生成和检测（<code>LCR[3] = 1</code>）且奇偶校验设置为奇数奇偶校验（<code>LCR[4] = 0</code>），奇偶校验错误位被设置。</p> <p>读取 LSR 会清除 PE 位（如果 <code>LSR_STATUS_CLEAR == 1</code>）或读取 LSR 或 RBR 会清除 PE 位（如果 <code>LSR_STATUS_CLEAR == 0</code>）。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (NO_PARITY_ERROR): 没有奇偶校验错误</li> <li>■ 0x1 (PARITY_ERROR): 奇偶校验错误</li> </ul> <p>复位值：0x0</p>
1	OE	R	<p>溢出错误位</p> <p>这用于指示发生溢出错误。如果在读取前一个数据之前收到新数据字符，就会发生这种情况。</p> <p>在非 FIFO 模式下，当新字符在从 RBR 读取前一个字符之前到达接收器时，OE 位被设置。发生这种情况时，RBR 中的数据被覆盖。在 FIFO 模式下，当 FIFO 已满且有新字符到达接收器时，会发生溢出错误。FIFO 中的数据被保留，接收移位寄存器中的数据丢失。</p> <p>读取 LSR 清除 OE 位（如果 <code>LSR_STATUS_CLEAR == 1</code>）或读</p>

位段	名称	类型	说明
			取 LSR 或 RBR 清除 OE 位（如 LSR_STATUS_CLEAR == 0）。 值： <ul style="list-style-type: none"> <li>0x0 (NO_OVER_RUN_ERROR): 没有溢出错误</li> <li>0x1 (OVER_RUN_ERROR): 溢出错误</li> </ul> 复位值：0x0
0	DR	R	数据就绪位 这用于指示接收器在 RBR 或接收器 FIFO 中至少包含一个字符。在非 FIFO 模式下读取 RBR 时，或在 FIFO 模式下，当接收 FIFO 为空时，该位被清除。 值： <ul style="list-style-type: none"> <li>0x0 (NOT_READY): 数据未就绪。</li> <li>0x1 (READY): 数据就绪。</li> </ul> 复位值：0x0

### 8.6.2.11 MSR

- 寄存器名：调制解调器状态寄存器
- 描述：当位 0、1、2 或 3 设置为逻辑 1 时，表示调制解调器控制输入发生变化，如果通过 IER 启用，无论何时发生变化，都会生成调制解调器状态中断。这些位（bit0、1、3）可以在复位后设置-即使它们各自的调制解调器信号处于非活动状态-因为调制解调器信号的同步版本具有复位值 0 并在复位后变为值 1。为防止由于此变化而导致不必要的中断，可以在复位后读取 MSR 寄存器。
- 偏移量：0x18

位段	名称	类型	说明
31:8	RSVD_MSR_31to8	R	保留位，读为零。
7	DCD	R	数据载波检测 这用于指示调制解调器控制线 dcd_n 的当前状态。也就是这个位是补码 dcd_n。当数据载波检测输入（dcd_n）有效时，它表示调制解调器或数据集已检测到载波。 在环回模式下（MCR[4]设置为 1），DCD 与 MCR[3]（输出 2）相同。 值： <ul style="list-style-type: none"> <li>0x0 (DEASSERTED): dcd_n 输入无效（逻辑 1）。</li> <li>0x1 (ASSERTED): dcd_n 输入有效（逻辑 0）。</li> </ul>

位段	名称	类型	说明
			复位值: 0x0
6	RI	R	<p>振铃指示器</p> <p>这用于指示调制解调器控制线 ri_n 的当前状态。也就是这个位是补码 ri_n。当振铃指示器输入 (ri_n) 有效, 它表示调制解调器或数据集已收到电话振铃信号。</p> <p>在环回模式下 (MCR[4]设置为 1), RI 与 MCR[2] (输出 1) 相同。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (DEASSERTED): ri_n 输入无效 (逻辑 1)。</li> <li>■ 0x1 (ASSERTED): ri_n 输入有效 (逻辑 0)。</li> </ul> <p>复位值: 0x0</p>
5	DSR	R	<p>数据集就绪</p> <p>这用于指示调制解调器控制线 dsr_n 的当前状态。也就是这个位是补码 dsr_n。</p> <p>在环回模式下 (MCR[4]设置为 1), DSR 与 MCR[0] (DTR) 相同。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (DEASSERTED): dsr_n 输入无效 (逻辑 1)。</li> <li>■ 0x1 (ASSERTED): dsr_n 输入有效 (逻辑 0)。</li> </ul> <p>复位值: 0x0</p>
4	CTS	R	<p>清除发送</p> <p>这用于指示调制解调器控制线 cts_n 的当前状态。也就是说, 该位是补码 cts_n。当清除发送输入 (cts_n) 有效时, 表示调制解调器或数据集已准备好与 UART 交换数据。</p> <p>在环回模式下 (MCR[4]设置为 1), CTS 与 MCR[1] (RTS) 相同。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (DEASSERTED): cts_n 输入无效 (逻辑 1)。</li> <li>■ 0x1 (ASSERTED): cts_n 输入有效 (逻辑 0)。</li> </ul> <p>复位值: 0x0</p>
3	DDCD	R	<p>Delta 数据载波检测</p> <p>这用于指示调制解调器控制线 dcd_n 自上次读取 MSR 以来</p>



位段	名称	类型	说明
			<p>发生了变化。</p> <p>读取 MSR 会清除 DDCD 位。在环回模式下（MCR[4]设置为 1），DDCD 反映 MCR[3]（输出 2）上的变化。</p> <p>注：如果未设置 DDCD 位，dcd_n 信号有效（低），且发生复位（软件或其他方式），在 dcd_n 信号保持有效的情况下移除复位时 DDCD 位将被设置。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (NO_CHANGE): 自从上次读取 MSR 以来 dcd_n 没有变化。</li> <li>■ 0x1 (CHANGE): 自从上次读取 MSR 以来 dcd_n 发生变化。</li> </ul> <p>复位值：0x0</p>
2	TERI	R	<p>振铃指示器的下降沿</p> <p>这用于指示自上次读取 MSR 以来，输出 ri_n 发生了变化（从有效低电平状态到无效高电平状态）。</p> <p>读取 MSR 会清除 TERI 位。在环回模式下（MCR[4]设置为 1），TERI 反映 MCR[2]（输出 1）何时从高电平变为低电平。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (NO_CHANGE): 自从上次读取 MSR 以来 ri_n 没有变化。</li> <li>■ 0x1 (CHANGE): 自从上次读取 MSR 以来 ri_n 发生变化。</li> </ul> <p>复位值：0x0</p>
1	DDSR	R	<p>delta 数据集就绪</p> <p>这用于指示自上次读取 MSR 以来，调制解调器控制线 dsr_n 发生了变化。</p> <p>读取 MSR 会清除 DDSR 位。在环回模式下（MCR[4]设置为 1），DDSR 反映 MCR[0]（DTR）上的变化。</p> <p>注：如果未设置 DDSR 位，dsr_n 信号有效（低），且发生复位（软件或其他方式），在 dsr_n 信号保持有效的情况下移除复位时 DDSR 位将被设置。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (NO_CHANGE): 自从上次读取 MSR 以来 dsr_n 没有变化。</li> <li>■ 0x1 (CHANGE): 自从上次读取 MSR 以来 dsr_n 发生变</li> </ul>

位段	名称	类型	说明
			化。 复位值：0x0
0	DCTS	R	delta 清除发送 这用于指示自上次读取 MSR 以来，调制解调器控制线 cts_n 发生了变化。 读取 MSR 会清除 DCTS 位。在环回模式下（MCR[4]设置为 1），DCTS 反映 MCR[1]（RTS）上的变化。 注：如果未设置 DCTS 位，cts_n 信号有效（低），且发生复位（软件或其他方式），在 cts_n 信号保持有效的情况下移除复位时 DCTS 位将被设置。 值： <ul style="list-style-type: none"> <li>0x0 (NO_CHANGE): 自从上次读取 MSR 以来 cts_n 没有变化。</li> <li>0x1 (CHANGE): 自从上次读取 MSR 以来 cts_n 发生变化。</li> </ul> 复位值：0x0

### 8.6.2.12 SCR

- 寄存器名：暂存器寄存器
- 描述：暂存器寄存器
- 偏移量：0x1c

位段	名称	类型	说明
31:8	RSVD_SCR_31to8	R	保留位，读为零。
7:0	SCR	R/W	该寄存器供程序员用作临时存储空间。它在 UART 中没有明确的用途。 复位值：0x0

### 8.6.2.13 LPDLL

- 寄存器名：低功耗分频数锁存器低
- 描述：低功耗分频数锁存器低寄存器
- 偏移量：0x20

位段	名称	类型	说明
31:8	RSVD_LPDLL_31to8	R	保留，读为零。
7:0	LPDLL	R/W	<p>该寄存器构成 16 位读/写低功耗分频数锁存器寄存器的低 8 位，该寄存器包含 UART 的波特率分频数，必须提供 115.2K 的波特率。这是接收器的 SIR 低功耗（最小脉冲宽度）检测所必需的。</p> <p>输出低功耗波特率等于串行时钟（sclk）频率除以波特率分频数值的 16 倍，如下：</p> $\text{低功耗波特率} = (\text{串行时钟频率}) / (16 * \text{分频数})$ <p>因此，必须选择分频数以提供 115.2K 的波特率。</p> <p>注：当低功耗分频数锁存器寄存器（LPDLL 和 LPDLH）设置为 0 时，低功耗波特时钟被禁用并且接收器不会发生低功耗脉冲检测（或任何脉冲检测）。此外，一旦设置了 LPDLL，在发送和接收数据之前，应允许最慢 UART 时钟的至少 8 个时钟周期通过。</p> <p>复位值：0x0</p>

### 8.6.2.14 LPDLH

- 寄存器名：低功耗分频数锁存器高
- 描述：低功耗分频数锁存器高寄存器
- 偏移量：0x24

位段	名称	类型	说明
31:8	RSVD_LPDLH_31to8	R	保留，读为零。
7:0	LPDLH	R/W	<p>该寄存器构成 16 位读/写低功耗分频数锁存器寄存器的高 8 位，该寄存器包含 UART 的波特率分频数，必须提供 115.2K 的波特率。这是接收器的 SIR 低功耗（最小脉冲宽度）检测所必需的。</p> <p>输出低功耗波特率等于串行时钟（sclk）频率除以波特率分频数值的 16 倍，如下：</p> $\text{低功耗波特率} = (\text{串行时钟频率}) / (16 * \text{分频数})$ <p>因此，必须选择分频数以提供 115.2K 的波特率。</p> <p>注：当低功耗分频数锁存器寄存器（LPDLL 和 LPDLH）设置为 0 时，低功耗波特时钟被禁用并且接收器不会发生低功耗脉冲检测（或任何脉冲检测）。此外，一旦设置了 LPDLH，</p>

位段	名称	类型	说明
			在发送和接收数据之前，应允许最慢 UART 时钟的至少 8 个时钟周期通过。 复位值：0x0

### 8.6.2.15 SRBRn

- 寄存器名：影子接收缓冲寄存器
- 偏移量：0x30 + n\*0x4

位段	名称	类型	说明
31:9	RSVD_SRBRn	R	保留，读为零。
8:0	SRBRn	R	影子接收缓冲寄存器 n。这是 RBR 的影子寄存器，已分配 16 个 32 位位置，以适应来自主的突发访问。该寄存器包含在 UART 模式下串行输入端口（sin）或红外模式下串行红外输入（sir_in）接收的数据字节。仅当线路状态寄存器（LSR）中的数据就绪（DR）位被设置时，该寄存器中的数据才有效。如果在非 FIFO 模式（FIFO_MODE == NONE）或 FIFO 被禁用（FCR[0]设置为 0），必须在下一个数据到达之前读取 RBR 中的数据，否则将被覆盖，导致溢出错误。 如果在 FIFO 模式（FIFO_MODE != NONE）并且启用了 FIFO（FCR[0]设置为 1），该寄存器访问接收 FIFO 的头部。如果接收 FIFO 已满，并且在下一个数据字符到达之前没有读取该寄存器，FIFO 中已经存在的数据将被保留，但任何传入的数据都将丢失，也会发生溢出错误。 复位值：0x0

### 8.6.2.16 STHRn

- 寄存器名：影子发送保持寄存器
- 偏移量：0x30 + n\*0x4

位段	名称	类型	说明
31:9	RSVD_STHRn	R	保留位，读为零。
8:0	STHRn	W	影子发送保持寄存器 n。这是 THR 的影子寄存器，已分配了 16 个 32 位位置，以适应来自主的突发访问。该寄存器包含要在 UART 模式下的串行输出端口（sout）或红外模式下的串行红外输出（sir_out_n）上传输的数据。只有设置了 THR

位段	名称	类型	说明
			<p>空 (THRE) 位 (LSR[5]) 时, 才应将数据写入 THR。</p> <p>如果在非 FIFO 模式或 FIFO 被禁用 (FCR[0]设置为 0) 并且 THRE 被设置, 将单个字符写入 THR 会清除 THRE。再次设置 THRE 之前对 THR 的任何额外写入都会导致 THR 数据被覆盖。</p> <p>在 FIFO 模式下启用 FIFO (FCR[0]设置为 1) 并且 THRE 被设置, x 个字符的数据可能会在 FIFO 满之前写入 THR。数字 x (默认 = 16) 由在配置期间设置的 FIFO 深度值确定。FIFO 已满时任何写入数据的尝试都会导致写入数据丢失。</p> <p>复位值: 0x0</p>

### 8.6.2.17 FAR

- 寄存器名: FIFO 访问寄存器
- 描述: FIFO 访问寄存器
- 偏移量: 0x70

位段	名称	类型	说明
31:1	RSVD_FAR_31to1	R	保留, 读为零。
0	FAR	R/W	<p>当 FIFO_ACCESS == No, 写入将无效, 始终可读。该寄存器用于启用 FIFO 访问模式进行测试, 以便在实现和启用 FIFO 时, 主可以写入接收 FIFO, 主可以读取发送 FIFO。当 FIFO 未实现或未启用时, 它允许主写入 RBR, 主读取 THR。</p> <p>注意, 当启用/禁用 FIFO 访问模式时, 接收 FIFO 和发送 FIFO 的控制部分被复位并且 FIFO 被视为空。</p> <p>值:</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 FIFO 访问模式。</li> <li>■ 0x1 (ENABLED): 启用 FIFO 访问模式。</li> </ul> <p>复位值: 0x0</p>

### 8.6.2.18 TFR

- 寄存器名: 发送 FIFO 读
- 偏移量: 0x74

位段	名称	类型	说明
31:8	RSVD_TFR_31to8	R	保留位，读为零。
7:0	TFR	R	<p>发送 FIFO 读</p> <p>这些位仅在启用 FIFO 访问模式（FAR[0]设置为 1）时有效。</p> <p>当实现并启用 FIFO 时，读取该寄存器会在发送 FIFO 的顶部提供数据。每次连续读取都会弹出发送 FIFO 并给出当前位于 FIFO 顶部的下一个数据值。</p> <p>当 FIFO 没有实现或没有启用时，读取该寄存器会提供 THR 中的数据。</p> <p>复位值：0x0</p>

### 8.6.2.19 RFW

- 寄存器名：接收 FIFO 写
- 偏移量：0x78

位段	名称	类型	说明
31:10	RSVD_RFW_31to10	R	保留位，读为零。
9	RFFE	W	<p>接收 FIFO 帧错误</p> <p>这些位仅在启用 FIFO 访问模式（FAR[0]设置为 1）时有效。</p> <p>当实现并启用 FIFO 时，该位用于将帧错误检测信息写入接收 FIFO。当 FIFO 未实现或未启用时，该位用于将帧错误检测信息写入 RBR。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用帧错误。</li> <li>■ 0x1 (ENABLED): 启用帧错误。</li> </ul> <p>复位值：0x0</p>
8	RFPE	W	<p>接收 FIFO 奇偶校验错误</p> <p>这些位仅在启用 FIFO 访问模式（FAR[0]设置为 1）时有效。</p> <p>当实现并启用 FIFO 时，该位用于将奇偶校验错误检测信息写入接收 FIFO。当 FIFO 未实现或未启用时，该位用于将奇偶校验错误检测信息写入 RBR。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用奇偶校验错误。</li> <li>■ 0x1 (ENABLED): 启用奇偶校验错误。</li> </ul>

位段	名称	类型	说明
			复位值: 0x0
7:0	RFWD	W	接收 FIFO 写数据  这些位仅在启用 FIFO 访问模式 (FAR[0] 设置为 1) 时有效。 当实现并启用 FIFO 时, 写入 RFWD 的数据将被推到接收 FIFO。每次连续写入都会将新数据推到接收 FIFO 中的下一个写入位置。当 FIFO 未实现或未启用时, 写入 RFWD 的数据被推到 RBR。  复位值: 0x0

### 8.6.2.20 USR

- 寄存器名: UART 状态寄存器
- 描述: UART 状态寄存器
- 偏移量: 0x7c

位段	名称	类型	说明
31:5	RSVD_USR_31to5	R	保留位, 读为零。
4	RFF	R	接收 FIFO 已满  该位仅在 FIFO_STAT == YES 时有效。这用于指示接收 FIFO 已满。即:  当接收 FIFO 不再满时, 该位被清除。  值: <ul style="list-style-type: none"> <li>■ 0x0 (NOT_FULL): 接收 FIFO 未滿。</li> <li>■ 0x1 (FULL): 接收 FIFO 已滿。</li> </ul> 复位值: 0x0
3	RFNE	R	接收 FIFO 非空  该位仅在 FIFO_STAT == YES 时有效。这用于指示接收 FIFO 包含一个或多个条目。当接收 FIFO 为空时, 该位被清除。  值: <ul style="list-style-type: none"> <li>■ 0x0 (EMPTY): 接收 FIFO 为空。</li> <li>■ 0x1 (NOT_EMPTY): 接收 FIFO 不为空。</li> </ul> 复位值: 0x0
2	TFE	R	发送 FIFO 空

位段	名称	类型	说明
			<p>该位仅在 FIFO_STAT == YES 时有效。这用于指示发送 FIFO 完全为空。当发送 FIFO 不再为空时，该位被清除。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (NOT_EMPTY): 发送 FIFO 不为空。</li> <li>■ 0x1 (EMPTY): 发送 FIFO 为空。</li> </ul> <p>复位值：0x1</p>
1	TFNF	R	<p>发送 FIFO 未滿</p> <p>该位仅在 FIFO_STAT == YES 时有效。这用于指示发送 FIFO 未滿。当发送 FIFO 已滿时，该位被清除。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (FULL): 发送 FIFO 已滿。</li> <li>■ 0x1 (NOT_FULL): 发送 FIFO 未滿。</li> </ul> <p>复位值：0x1</p>
0	BUSY	R	<p>UART 忙</p> <p>该位仅在 UART_16550_COMPATIBLE == NO 时有效。这表示正在进行串行传输，清除时表示 UART 空闲或未激活。</p> <p>在以下任何情况下，该位将设置为 1（忙）：</p> <p>串行接口上的发送正在进行</p> <p>THR 中存在发送数据，当不使用 FIFO 访问模式（FAR = 0），且分频数锁存器访问位为 0（LCR.DLAB = 0）时，波特率分频数为非零（{DLH,DLL}不等于 0）。</p> <p>接口上的接收正在进行</p> <p>当不使用 FIFO 访问模式（FAR = 0）时，RBR 中存在接收数据。</p> <p>注：UART 忙位可能会被清除，即使新字符可能已从另一个设备发送。也就是说，如果 UART 在 THR 和 RBR 中没有数据并且没有正在进行的传输并且新字符的起始位刚刚到达 UART。这是因为有效的起始位要在比特周期的中间才会出现，而这个时间取决于已经编程的波特率分频数。如果第二个系统时钟已经实现（CLOCK_MODE == Enabled），该位的生效也会被较慢的时钟的几个周期所延迟。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x0 (IDLE): UART 空闲或未激活。</li> </ul>



位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>0x1 (BUSY): UART 忙（正在传输数据）。</li> </ul> 复位值: 0x0

### 8.6.2.21 TFL

- 寄存器名: 发送 FIFO 水平
- 偏移量: 0x80

位段	名称	类型	说明
31:5	RSVD_TFL_31toADDR_WIDTH	R	保留位, 读为零。
4:0	tfl	R	发送 FIFO 水平 表示发送 FIFO 中的数据条目数。 复位值: 0x0

### 8.6.2.22 RFL

- 寄存器名: 接收 FIFO 水平
- 偏移量: 0x84

位段	名称	类型	说明
31:5	RSVD_RFL_31toADDR_WIDTH	R	保留位, 读为零。
4:0	rfl	R	接收 FIFO 水平 表示接收 FIFO 中的数据条目数。 复位值: 0x0

### 8.6.2.23 SRR

- 寄存器名: 软件复位寄存器
- 偏移量: 0x88

位段	名称	类型	说明
31:3	RSVD_SRR_31to3	R	保留位, 读为零。

位段	名称	类型	说明
2	XFR	W	<p>XMIT FIFO 复位</p> <p>当 FIFO_MODE == NONE 时，写入无效。</p> <p>这是 XMIT FIFO 复位位（FCR[2]）的影子寄存器。这可用于消除软件必须存储先前写入的 FCR 值（它们是相当静态的）来复位发送 FIFO 的负担。这会复位发送 FIFO 的控制部分并将 FIFO 视为空。当选择了额外的 DMA 握手信号（DMA_EXTRA = YES）时，这也将禁用 DMA 发送请求和单个信号。注意，该位是自动清除的，不需要清除该位。</p> <p>复位值：0x0</p>
1	RFR	W	<p>RCVR FIFO 复位</p> <p>当 FIFO_MODE == NONE 时，写入无效。</p> <p>这是 RCVR FIFO 复位位（FCR[1]）的影子寄存器。这可用于消除软件必须存储先前写入的 FCR 值（它们是相当静态的）来复位接收 FIFO 的负担。这会复位接收 FIFO 的控制部分并将 FIFO 视为空。当选择了额外的 DMA 握手信号（DMA_EXTRA = YES）时，这也将禁用 DMA 接收请求和单个信号。注意，该位是自动清除的，不需要清除该位。</p> <p>复位值：0x0</p>
0	UR	W	<p>UART 复位</p> <p>这会异步复位 UART 并同步移除复位启用。对于双时钟实现，pclk 和 sclk 域都将被复位。</p> <p>值：</p> <ul style="list-style-type: none"> <li>0x0 (NO_RESET): 无 UART 复位</li> <li>0x1 (RESET): UART 复位</li> </ul> <p>复位值：0x0</p>

### 8.6.2.24 SRTS

- 寄存器名：影子请求发送
- 偏移量：0x8c

位段	名称	类型	说明
31:1	RSVD_SRTS_31to1	R	保留位，读为零。

位段	名称	类型	说明
0	SRTS	R/W	<p>影子请求发送</p> <p>这是 RTS 位(MCR[1])的影子寄存器,可用于消除必须在 MCR 上执行读修改写的负担。这用于直接控制请求发送 (rst_n) 输出。请求发送 (rst_n) 输出用于通知调制解调器或数据集 UART 已准备好交换数据。</p> <p>当自动 RTS 流控制未启用时 (MCR[5]设置为 0), 通过将 MCR[1] (RTS) 编程为高电平来将 rts_n 信号设置为低电平。</p> <p>在自动流控制中, AFCE_MODE == Enabled 和激活 (MCR[5] 设置为 1) 和 FIFO 启用 (FCR[0]设置为 1), rts_n 输出以相同方式控制, 但仅在禁用 TRC 流触发时, 也由接收器 FIFO 阈值触发 (高于阈值时 rts_n 为无效高电平); 否则, 它由接收器 FIFO 几乎满触发器控制, 其中“几乎满”是指 FIFO 中的两个可用槽 (当高于阈值时, rts_n 为无效高电平)。</p> <p>注: 在环回模式下 (MCR[4]设置为 1), rts_n 输出保持无效高电平, 而该位置的值在内部环回到输入。</p> <p>值:</p> <ul style="list-style-type: none"> <li>0x0 (DEASSERTED): 影子请求发送 uart_rts_n 逻辑 1。</li> <li>0x1 (ASSERTED): 影子请求发送 uart_rts_n 逻辑 0。</li> </ul> <p>复位值: 0x0</p>

### 8.6.2.25 SBCR

- 寄存器名: 影子中断控制寄存器
- 偏移量: 0x90

位段	名称	类型	说明
31:1	RSVD_SBCR_31to1	R	保留位, 读为零。
0	SBCB	R/W	<p>影子中断控制位</p> <p>这是中断位 (LCR[6]) 的影子寄存器, 这用于消除必须对 LCR 执行读修改写的负担。这用于使中断条件发送到接收设备。如果设置为 1, 串行输出被强制为间隔 (逻辑 0) 状态。当不处于环回模式时, 由 MCR[4]确定, sout 线被强制为低电平, 直到 Break 位被清除。</p> <p>如果 SIR_MODE == Enabled 且处于活动状态 (MCR[6]设置为 1), sir_out_n 线将连续脉冲。在环回模式下, 中断条件在内部环回到接收器。</p>

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> <li>■ 0x0 (NO_BREAK): 串行输出上没有间隔</li> <li>■ 0x1 (BREAK): 串行输出强制间隔</li> </ul> 复位值: 0x0

### 8.6.2.26 SDMAM

- 寄存器名: 影子 DMA 模式寄存器
- 偏移量: 0x94

位段	名称	类型	说明
31:1	RSVD_SDMAM_31to1	R	保留位, 读为零。
0	SDMAM	R/W	影子 DMA 模式  这是 DMA 模式位 (FCR[3]) 的影子寄存器。这用于消除必须将先前写入的值存储到内存中的 FCR 以及必须屏蔽该值的负担, 以便仅更新 DMA 模式位。当未选择额外的 DMA 握手信号时 (DMA_EXTRA == NO), 这决定了用于 dma_tx_req_n 和 dma_rx_req_n 输出信号的 DMA 信令模式。有关 DMA 支持的详细信息, 请参见第 54 页的第 5.9 节。  值： <ul style="list-style-type: none"> <li>■ 0x0 (MODE_0): 模式 0</li> <li>■ 0x1 (MODE_1): 模式 1</li> </ul> 复位值: 0x0

### 8.6.2.27 SFE

- 寄存器名: 影子 FIFO 启用寄存器
- 偏移量: 0x98

位段	名称	类型	说明
31:1	RSVD_SFE_31to1	R	保留位, 读为零。
0	SFE	R/W	影子 FIFO 启用  这是 FIFO 启用位 (FCR[0]) 的影子寄存器。这用于消除必须将先前写入的值存储到内存中的 FCR 以及必须屏蔽该值的负担, 以便仅更新 FIFO 启用位。这将启用/禁用发送 (XMIT) 和接收 (RCVR) FIFO。如果该位在启用后设置为 0 (禁用),

位段	名称	类型	说明
			FIFO 中的 XMIT 和 RCVR 控制器部分都将被复位。 值： <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用 FIFO。</li> <li>0x1 (ENABLED): 启用 FIFO。</li> </ul> 复位值：0x0

### 8.6.2.28 SRT

- 寄存器名：影子 RCVR 触发寄存器
- 大小：32 bits
- 偏移量：0x9c

位段	名称	类型	说明
31:2	RSVD_SRT_31to2	R	保留位，读为零。
1:0	SRT	R/W	影子 RCVR 触发器 这是 RCVR 触发位 (FCR[7:6]) 的影子寄存器。这用于消除必须将先前写入的值存储到内存中的 FCR 以及必须屏蔽该值的负担，以便仅更新 RCVR 触发位。 这用于选择接收器 FIFO 中产生接收数据可用中断的触发水平。它还确定当 DMA 模式 (FCR[3]) 设置为 1 时，dma_rx_req_n 信号何时生效。 值： <ul style="list-style-type: none"> <li>0x0 (FIFO_CHAR_1): FIFO 中有 1 个字符</li> <li>0x1 (FIFO_QUARTER_FULL): FIFO 1/4 满</li> <li>0x2 (FIFO_HALF_FULL): FIFO 1/2 满</li> <li>0x3 (FIFO_FULL_2): FIFO 还差 2 个不满</li> </ul> 复位值：0x0

### 8.6.2.29 STET

- 寄存器名：影子发送空触发寄存器
- 偏移量：0xa0

位段	名称	类型	说明
31:2	RSVD_STET_31to2	R	保留位，读为零。

位段	名称	类型	说明
1:0	STET	R/W	<p>影子发送空触发器</p> <p>这是发送空触发位 (FCR[5:4]) 的影子寄存器。这用于消除必须将先前写入的值存储到内存中的 FCR 以及必须屏蔽该值的负担, 以便仅更新发送空触发位。当 THRE_MODE_USER == Disabled 时, 写入无效。这用于选择当模式处于活动状态时产生 THRE 中断的空阈值水平。</p> <p>值:</p> <ul style="list-style-type: none"> <li>0x0 (FIFO_EMPTY): FIFO 为空</li> <li>0x1 (FIFO_CHAR_2): FIFO 中有 2 个字符</li> <li>0x2 (FIFO_QUARTER_FULL): FIFO 1/4 满</li> <li>0x3 (FIFO_HALF_FULL): FIFO 1/2 满</li> </ul> <p>复位值: 0x0</p>

### 8.6.2.30 HTX

- 寄存器名: 中止发送
- 名称: 中止发送
- 偏移量: 0xa4

位段	名称	类型	说明
31:1	RSVD_HTX_31to1	R	保留位, 读为零。
0	HTX	R/W	<p>中止发送</p> <p>当 FIFO_MODE == NONE 时写入无效, 始终可读。该寄存器用于中止发送以进行测试, 以便在实现和启用 FIFO 时主可以填充发送 FIFO。</p> <p>注意, 如果 FIFO 已实现且未启用, 中止发送寄存器的设置将不会影响操作。</p> <p>值:</p> <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用中止发送。</li> <li>0x1 (ENABLED): 启用中止发送。</li> </ul> <p>复位值: 0x0</p>

### 8.6.2.31 DMA\_SA

- 寄存器名: DMA 软件确认寄存器

- 描述：DMA 软件确认寄存器
- 偏移量：0xa8

位段	名称	类型	说明
31:1	RSVD_DMASA_31to1	R	保留位，读为零。
0	DMASA	W	<p>DMA 软件确认</p> <p>当 DMA_EXTRA == No 时，写入无效。如果由于错误条件需要中止传输，该寄存器用于执行 DMA 软件确认。例如，如果 DMA 禁用通道，UART 应该清除它的请求。这将导致发送请求、发送单、接收请求和接收单信号无效。注意，该位是自动清除的，无需清除该位。</p> <p>值：</p> <ul style="list-style-type: none"> <li>■ 0x1 (SOFT_ACK): DMA 软件确认</li> </ul> <p>复位值：0x0</p>

### 8.6.2.32 TCR

- 寄存器名：收发器控制寄存器
- 描述：该寄存器用于启用或禁用 RS485 模式，同时控制 DE 和 RE 信号的极性值。
- 偏移量：0xac

位段	名称	类型	说明
31:5	RSVD_TCR_31to5	R	保留位，读为零。
4:3	XFER_MODE	R/W	<p>传输模式</p> <ul style="list-style-type: none"> <li>■ 0：在这种模式下，发送和接收可以同时发生。用户可以在任何时间点启用 DE_EN 和 RE_EN。在 TAT 寄存器中编程的周转时间不适用于该模式。</li> <li>■ 1：在该模式下，DE 和 RE 互斥。DE 或 RE 中只有一个可以通过编程启用。</li> </ul> <p>从 RE 切换到 DE 或从 DE 切换到 RE 时，硬件将考虑在 TAT 寄存器中编程的周转时间。对于传输，如果硬件正在接收任何传输，它会在开始传输之前等待。</p> <ul style="list-style-type: none"> <li>■ 2：在这种模式下，DE 和 RE 互斥。一旦 DE_EN/RE_EN 被编程-默认情况下，RE 将被启用并且 UART 控制器将准备好接收。如果用户使用数据对发送 FIFO 进行编程，UART 在确保没有接收正在进行后，禁用 RE 信号并启用 DE 信号。</li> </ul>

位段	名称	类型	说明
			一旦发送 FIFO 变空，RE 信号将被启用，DE 信号将被禁用。在这种操作模式下，当从 RE 切换到 DE 或从 DE 切换到 RE 时，硬件将考虑在 TAT 寄存器中编程的周转时间。在这种模式下，DE 和 RE 信号严格互补。 复位值：0x0
2	DE_POL	R/W	驱动器使能极性 ■ 1：DE 信号高电平有效。 ■ 0：DE 信号低电平有效。 复位值：UART_DE_POL
1	RE_POL	R/W	接收器使能极性 ■ 1：RE 信号高电平有效。 ■ 0：RE 信号低电平有效。 复位值：UART_RE_POL
0	RS485_EN	R/W	RS485 传输使能 ■ 0：在这种模式下，传输仍处于 RS232 模式。该寄存器中的所有其他字段均保留。寄存器 DE_EN/RE_EN/TAT 也保留。 ■ 1：在这种模式下，传输将以 RS485 模式进行。该寄存器中的所有其他字段均适用。 复位值：0x0

### 8.6.2.33 DE\_EN

- 寄存器名：驱动器输出使能寄存器
- 描述：驱动器输出使能寄存器（DE\_EN）用于控制 DE 信号的启用和禁用。
- 偏移量：0xb0

位段	名称	类型	说明
31:1	RSVD_DE_EN_31to1	R	保留位，读为零。
0	DE_Enable	R/W	DE 使能控制 DE 使能寄存器位用于控制 DE 信号的启用和禁用。 ■ 0：禁用 DE 信号。 ■ 1：启用 DE 信号。



位段	名称	类型	说明
			复位值：0x0

### 8.6.2.34 RE\_EN

- 寄存器名：接收器输出使能寄存器
- 描述：接收器输出使能寄存器（RE\_EN）用于控制 RE 信号的启用和禁用。
- 偏移量：0xb4

位段	名称	类型	说明
31:1	RSVD_RE_EN_31to1	R	保留位，读为零。
0	RE_Enable	R/W	RE 使能控制 RE 使能寄存器位用于控制 RE 信号的启用和禁用。 ■ 0：禁用 RE 信号。 ■ 1：启用 RE 信号。 复位值：0x0

### 8.6.2.35 DET

- 寄存器名：驱动器输出使能时间寄存器
- 描述：驱动器输出使能时间寄存器（DET）用于控制 DE 信号的启用和禁用时间。
- 偏移量：0xb8

位段	名称	类型	说明
31:24	RSVD_DE_DEAT_31to24	R	保留位，读为零。
23:16	DE_De-assertion_Time	R/W	驱动器使能无效时间 该字段控制从 sout 上的停止位结束到驱动器输出使能信号的下降沿之间的时间量（以串行时钟周期数计）。 复位值：0x0
15:8	RSVD_DE_AT_15to8	R	保留位，读为零。 复位值：0x0
7:0	DE_Assertion_Time	R/W	驱动器使能有效时间 该字段控制驱动器输出使能信号的上升沿有效到串行发送使能之间的时间量（以串行时钟周期数计）。发送缓冲区中的任何数据都将在发送使能后从串行输出（sout）开始。

位段	名称	类型	说明
			复位值：0x0

### 8.6.2.36 TAT

- 寄存器名：周转时间寄存器
- 描述：周转时间寄存器（TAT）用于保持 RE 和 DE 信号切换之间的周转时间。
- 偏移量：0xbc

位段	名称	类型	说明
31:16	RE_to_DE	R/W	接收器使能到驱动使能的周转时间 RE 无效到 DE 有效的周转时间（以串行时钟计） 注： 如果 DET 寄存器中的 DE 有效时间为 0，实际值为编程值+3。 如果 DET 寄存器中的 DE 有效时间为 1，实际值为编程值+2。 如果 DET 寄存器中的 DE 有效时间大于 1，实际值为编程值+1。 复位值：0x0
15:0	DE_to_RE	R/W	驱动使能到接收器使能的周转时间 从 DE 有效到 RE 有效的周转时间（以串行时钟计） 注：实际时间为编程值+1。 复位值：0x0

### 8.6.2.37 DLF

- 寄存器名：分频数锁存器小数寄存器
- 偏移量：0xc0

位段	名称	类型	说明
31:4	RSVD_DLF	R	保留，读为零。
3:0	DLF	R/W	分频数的小数部分 小数值与 DLH、DLL 设置的整数值相加。小数值由分频数小数值/2 <sup>DLF_SIZE</sup> 确定。 复位值：0x0

### 8.6.2.38 RAR

- 寄存器名：接收地址寄存器
- 描述：接收地址寄存器
- 偏移量：0xc4

位段	名称	类型	说明
31:8	RSVD_RAR_31to8	R	保留位，读为零。
7:0	RAR	R/W	<p>这是接收模式期间的地址匹配寄存器。如果在输入字符中设置了 bit9，将根据该寄存器值检查剩余的 8 位。如果匹配发生，将 bit9 设置为 0 的后续字符将被视为数据字节，直到收到下一个地址字节。</p> <p>注：</p> <p>仅当 ADDR_MATCH (LCR_EXT[1]) 和 DLS_E (LCR_EXT[0]) 位设置为 1 时，该寄存器才适用。</p> <p>如果 UART_16550_COMPATIBLE 配置为 0，RAR 应仅在 UART 空闲时进行编程。</p> <p>如果 UART_16550_COMPATIBLE 配置为 0，RAR 可以在任何时间点进行编程。但是，当接收正在进行时，用户不得更改该寄存器值。</p> <p>复位值：0x0</p>

### 8.6.2.39 TAR

- 寄存器名：发送地址寄存器
- 描述：发送地址寄存器
- 偏移量：0xc8

位段	名称	类型	说明
31:8	RSVD_TAR_31to8	R	保留位，读为零。
7:0	TAR	R/W	<p>这是发送模式期间的地址匹配寄存器。如果启用了 DLS_E (LCR_EXT[0]) 位，UART 将发送 9 位字符并将 bit9 设置为 1，并且剩余的 8 位地址将从该寄存器发送，前提是 SEND_ADDR (LCR_EXT[2]) 位设置为 1。</p> <p>注：</p> <ul style="list-style-type: none"> <li>■ 该寄存器仅适用于发送地址。正常的数据应该通过编程 THR 寄存器发送。</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>一旦地址开始在 UART 串行通道上发送, 硬件将自动清除 SEND_ADDR 位。</li> </ul> 复位值: 0x0

### 8.6.2.40 LCR\_EXT

- 寄存器名: 线路扩展控制寄存器
- 描述: 线路扩展控制寄存器
- 大小: 32 bits
- 偏移量: 0xcc

位段	名称	类型	说明
31:4	RSVD_LCR_EXT	R	保留位, 读为零。
3	TRANSMIT_MODE	R/W	发送模式控制寄存器。该位用于控制 9 位数据传输期间的传输模式类型。 <ul style="list-style-type: none"> <li>1: 在这种操作模式下, 发送保持寄存器 (THR) 和影子发送保持寄存器 (STHR) 为 9 位宽。用户需要确保 THR/STHR 寄存器正确写入地址/数据。 地址: bit9 设置为 1, 数据: bit9 设置为 0。 注: 发送地址寄存器 (TAR) 不适用于这种操作模式。</li> <li>0: 在这种操作模式下, 发送保持寄存器 (THR) 和影子发送保持寄存器 (STHR) 为 8 位宽。用户需要将地址编程到发送地址寄存器 (TAR), 将数据编程到 THR/STHR 寄存器。SEND_ADDR 位作为一个控制旋钮, 指示 UART 何时发送地址。</li> </ul> 复位值: 0x0
2	SEND_ADDR	R/W	发送地址控制位。该位用作控制旋钮, 供用户确定在发送期间何时发送地址。 <ul style="list-style-type: none"> <li>1: 9 位字符将被发送, bit9 设置为 1, 其余 8 位将与发送地址寄存器中编程的内容相匹配。</li> <li>0: 9 位字符将被发送, bit9 设置为 0, 其余 8 位将从通过 8 位宽 THR/STHR 寄存器编程的发送 FIFO 中获取。</li> </ul> 注: <ul style="list-style-type: none"> <li>1. 该位在发送地址字符后由硬件自动清除, 用户不应将此位编程为 0。</li> </ul>

位段	名称	类型	说明
			<ul style="list-style-type: none"> <li>2. 该字段仅在 DLS_E 位设置为 1 且 TRANSMIT_MODE 设置为 0 时适用。</li> </ul> 复位值: 0x0
1	ADDR_MATCH	R/W	地址匹配模式。该位用于在接收期间启用地址匹配功能。 <ul style="list-style-type: none"> <li>1: 地址匹配模式; UART 将等到 bit9 设置为 1 的传入字符。并进一步检查地址是否与接收地址匹配寄存器中编程的内容匹配。如果发现匹配, 后续字符将被视为有效数据, UART 开始接收数据。</li> <li>0: 正常模式; UART 将开始接收数据并形成 9 位字符写入接收 FIFO。用户需要负责读取数据并区分地址和数据。</li> </ul> 注: 该字段仅在 DLS_E 设置为 1 时适用。 复位值: 0x0
0	DLS_E	R/W	DLS 的扩展。该位用于启用 9 位数据以进行发送和接收传输。 复位值: 0x0

### 8.6.2.41 CPR

- 寄存器名: 组件参数寄存器
- 描述: 组件参数寄存器
- 偏移量: 0xf4

位段	名称	类型	说明
31:24	RSVD_CPR_31to24	R	保留位, 读为零。
23:16	FIFO_MODE	R	FIFO_MODE 配置参数的编码值: <ul style="list-style-type: none"> <li>0x0 (FIFO_MODE_0): FIFO 模式为 0。</li> <li>0x1 (FIFO_MODE_16): FIFO 模式为 16。</li> <li>0x2 (FIFO_MODE_32): FIFO 模式为 32。</li> <li>0x4 (FIFO_MODE_64): FIFO 模式为 64。</li> <li>0x8 (FIFO_MODE_128): FIFO 模式为 128。</li> <li>0x10 (FIFO_MODE_256): FIFO 模式为 256。</li> <li>0x20 (FIFO_MODE_512): FIFO 模式为 512。</li> <li>0x40 (FIFO_MODE_1024): FIFO 模式为 1024。</li> </ul>

位段	名称	类型	说明
			■ 0x80 (FIFO_MODE_2048): FIFO 模式为 2048。 复位值: UART_ENCODED_FIFO_MODE
15:14	RSVD_CPR_15to14	R	保留位, 读为零。 复位值: 0x0
13	DMA_EXTRA	R	DMA_EXTRA 配置参数值的编码 值: <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 DMA_EXTRA。</li> <li>■ 0x1 (ENABLED): 启用 DMA_EXTRA。</li> </ul> 复位值: DMA_EXTRA
12	UART_ADD_ENCODED_PARAMS	R	UART_ADD_ENCODED_PARAMS 配置参数值的编码 值: <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 UART_ADD_ENCODED_PARAMS。</li> <li>■ 0x1 (ENABLED): 启用 UART_ADD_ENCODED_PARAMS。</li> </ul> 复位值: UART_ADD_ENCODED_PARAMS
11	SHADOW	R	SHADOW 配置参数值的编码 值: <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 SHADOW。</li> <li>■ 0x1 (ENABLED): 启用 SHADOW。</li> </ul> 复位值: SHADOW
10	FIFO_STAT	R	FIFO_STAT 配置参数值的编码 值: <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 FIFO_STAT。</li> <li>■ 0x1 (ENABLED): 启用 FIFO_STAT。</li> </ul> 复位值: FIFO_STAT
9	FIFO_ACCESS	R	FIFO_ACCESS 配置参数值的编码 值: <ul style="list-style-type: none"> <li>■ 0x0 (DISABLED): 禁用 FIFO_ACCESS。</li> <li>■ 0x1 (ENABLED): 启用 FIFO ACCESS。</li> </ul>

位段	名称	类型	说明
			复位值: FIFO_ACCESS
8	ADDITIONAL_FEAT	R	ADDITIONAL_FEATURES 配置参数值的编码值: <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用附加功能。</li> <li>0x1 (ENABLED): 启用附件功能。</li> </ul> 复位值: ADDITIONAL_FEATURES
7	SIR_LP_MODE	R	SIR_LP_MODE 配置参数值的编码值: <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用 SIR_LP 模式。</li> <li>0x1 (ENABLED): 启用 SIR_LP 模式。</li> </ul> 复位值: SIR_LP_MODE
6	SIR_MODE	R	SIR_MODE 配置参数值的编码值: <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用 SIR 模式。</li> <li>0x1 (ENABLED): 启用 SIR 模式。</li> </ul> 复位值: SIR_MODE
5	THRE_MODE	R	THRE_MODE 配置参数值的编码值: <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用 THRE 模式。</li> <li>0x1 (ENABLED): 启用 THRE 模式。</li> </ul> 复位值: THRE_MODE_RST
4	AFCE_MODE	R	AFCE_MODE 配置参数值的编码值: <ul style="list-style-type: none"> <li>0x0 (DISABLED): 禁用 AFCE 模式。</li> <li>0x1 (ENABLED): 启用 AFCE 模式。</li> </ul> 复位值: AFCE_MODE
3:2	RSVD_CPR_3to2	R	保留位, 读为零。 复位值: 0x0
1:0	APB_DATA_WIDTH	R	APB_DATA_WIDTH 配置参数值的编码

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> <li>0x0 (APB_8BITS): APB 数据宽度为 8 位。</li> <li>0x1 (APB_16BITS): APB 数据宽度为 16 位。</li> <li>0x2 (APB_32BITS): APB 数据宽度为 32 位。</li> </ul> 复位值: UART_ENCODED_APB_WIDTH

### 8.6.2.42 UCV

- 寄存器名: UART 组件版本
- 偏移量: 0xf8

位段	名称	类型	说明
31:0	UART_Component_Version	R	版本中每个数字的 ASCII 值, 后跟*。例如, 32_30_31_2A 代表版本 2.01*。 复位值: UART_COMP_VERSION

### 8.6.2.43 CTR

- 寄存器名: 组件类型寄存器
- 偏移量: 0xfc

位段	名称	类型	说明
31:0	Peripheral_ID	R	该寄存器包含外设识别码。



## 9 GPIO

### 9.1 概述

曳影 1520 芯片中有 6 组 GPIO，其中 2 组在 AON 电源域，分别为：AOGPIO 和 GPIO4；4 组在 AP 电源域，分别命名为 GPIO0、GPIO1、GPIO2 和 GPIO3，每组有 32 个多功能 I/O 端口管脚。每个端口都可以通过软件轻松配置，以满足各种系统配置和设计要求。必须在启动主程序之前定义要使用的每个管脚的功能。如果不使用该管脚上的复用功能，可以将该管脚配置为 I/O 端口。在管脚配置之前，应仔细配置初始管脚状态以避免一些问题。

### 9.2 主要特性

GPIO 支持以下特性：

- 曳影 1520 芯片中有 6 组相同的 GPIO
  - 2 组在 AON 电源域，分别命名为 AOGPIO、GPIO4
  - 4 组在 AP 电源域，分别命名为 GPIO0、GPIO1、GPIO2 和 GPIO3
- 每组有 32 个可独立配置的信号
- 每个信号都有单独的数据寄存器 and 数据方向寄存器
- 独立可控的信号位
- 可配置的中断模式并生成单个中断
- GPIO 组件类型寄存器
- GPIO 组件版本寄存器
- 输出信号的可配置复位值

### 9.3 接口

图表 9-1 管脚描述

管脚名称	方向	宽度	说明
GPIO0_[0~31]	IO	1	GPIO0 PAD 管脚名称
GPIO1_[0~31]	IO	1	GPIO1 PAD 管脚名称
GPIO2_[0~31]	IO	1	GPIO2 PAD 管脚名称
GPIO3_[0~31]	IO	1	GPIO3 PAD 管脚名称

## 9.4 功能描述

GPIO 控制外部 I/O PAD 的输出数据和方向。它还可以使用内存映射寄存器读回外部 PAD 上的数据。

可以对 GPIO I/O 端口进行编程，以接收外部信号作为信号任何位的中断源。中断类型是可编程的。可以通过对 gpio\_intmask 寄存器进行编程来屏蔽中断。可以在屏蔽之前（称为原始状态）和屏蔽之后读取中断状态。这些中断也可以组合成一个单独的中断输出信号，它与单个中断具有相同的极性。为了屏蔽组合中断，必须屏蔽所有单个中断。单个组合中断没有自己的屏蔽位。

## 9.5 使用

- 配置 GPIO 输出：
  1. 配置 GPIO.gpio\_swporta\_ctl.gpio\_swporta\_ctl[0] = 1'b1，配置 GPIO[0]使用软件模式。
  2. 配置 GPIO.gpio\_swporta\_ddr.gpio\_swporta\_ddr[0] = 1'b1，配置 GPIO[0]直接输入。
  3. 配置 GPIO.gpio\_swporta\_dr.gpio\_swporta\_dr[0] = 1'b1，将数据 1'b1 写入 GPIO。
  4. 监控 PAD GPIO[0] = 1。
- 配置 GPIO 输入：
  1. 配置 GPIO.gpio\_swporta\_ctl.gpio\_swporta\_ctl[0] = 1'b1，配置 GPIO[0]使用软件模式。
  2. 配置 GPIO.gpio\_swporta\_ddr.gpio\_swporta\_ddr[0] = 1'b0，配置 GPIO[0]直接输入。
  3. 将 PAD GPIO[0]设置为 1。
  4. 读取 GPIO 端口 A 外部端口寄存器，端口 A 外部端口寄存器[0] = 1，检查 GPIO[0] = 1。

## 9.6 寄存器描述

### 9.6.1 寄存器内存映射

名称	偏移量	说明	章节/页码
GPIO_SWPORTA_DR	0x0	端口 A 数据寄存器 复位值：0x0	<a href="#">9.6.2.1/351</a>
GPIO_SWPORTA_DDR	0x4	端口 A 数据方向寄存器 复位值：0x0	<a href="#">9.6.2.2/352</a>
GPIO_INTEN	0x30	中断使能寄存器 复位值：0x0	<a href="#">9.6.2.3/352</a>
GPIO_INTMASK	0x34	中断屏蔽寄存器 复位值：0x0	<a href="#">9.6.2.4/352</a>

名称	偏移量	说明	章节/页码
GPIO_INTTYPE_LEVEL	0x38	中断电平 复位值：0x0	<a href="#">9.6.2.5/352</a>
GPIO_INT_POLARITY	0x3C	中断极性 复位值：0x0	<a href="#">9.6.2.6/353</a>
GPIO_INTSTATUS	0x40	中断状态 复位值：0x0	<a href="#">9.6.2.7/353</a>
GPIO_RAW_INTSTATUS	0x44	原始中断状态 复位值：0x0	<a href="#">9.6.2.8/353</a>
GPIO_PORTA_EOI	0x4C	端口 A 清除中断寄存器 复位值：0x0	<a href="#">9.6.2.9/353</a>
GPIO_EXT_PORTA	0x50	外部端口 A 寄存器 复位值：0x0	<a href="#">9.6.2.10/354</a>
GPIO_LS_SYNC	0x60	同步电平 复位值：0x0	<a href="#">9.6.2.11/354</a>
GPIO_ID_CODE	0x64	GPIO ID 码 复位值：0x0	<a href="#">9.6.2.12/354</a>
GPIO_VER_ID_CODE	0x6C	GPIO 组件版本 复位值：0x0	<a href="#">9.6.2.13/354</a>
GPIO_CONFIG_REG2	0x70	GPIO 配置寄存器 2 复位值：0x0	<a href="#">9.6.2.14/355</a>
GPIO_CONFIG_REG1	0x74	GPIO 配置寄存器 1 复位值：0x0	<a href="#">9.6.2.15/355</a>

## 9.6.2 寄存器和字段描述

### 9.6.2.1 GPIO\_SWPORTA\_DR

- 寄存器名：GPIO\_SWPORTA\_DR
- 偏移量：0x00
- 默认值：0x0

位段	名称	类型	说明
31:0	GPIO_SWPORTA_DR	R/W	端口 A 数据寄存器

### 9.6.2.2 GPIO\_SWPORTA\_DDR

- 寄存器名: GPIO\_SWPORTA\_DDR
- 偏移量: 0x04
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_SWPORTA_DDR	R/W	端口 A 数据方向寄存器

### 9.6.2.3 GPIO\_INTEN

- 寄存器名: GPIO\_INTEN
- 偏移量: 0x30
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_INTEN	R/W	中断使能寄存器

### 9.6.2.4 GPIO\_INTMASK

- 寄存器名: GPIO\_INTMASK
- 偏移量: 0x34
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_INTMASK	R/W	中断屏蔽寄存器

### 9.6.2.5 GPIO\_INTTYPE\_LEVEL

- 寄存器名: GPIO\_INTTYPE\_LEVEL
- 偏移量: 0x38
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_INTTYPE_LEVEL	R/W	中断电平, 控制端口 A 上可能发生的中断类型。每当向该寄存器的某个位写入 0 时, 它将中断类型配置为电平敏感; 否

位段	名称	类型	说明
			则，它是边沿敏感。

### 9.6.2.6 GPIO\_INT\_POLARITY

- 寄存器名：GPIO\_INT\_POLARITY
- 偏移量：0x3C
- 默认值：0x0

位段	名称	类型	说明
31:0	GPIO_INT_POLARITY	R/W	中断电平中断极性。控制端口 A 输入上可能出现的边沿或电平敏感的极性。每当向该寄存器的某个写入 0 时，它将中断类型配置为下降沿或低电平敏感；否则，它是上升沿或高电平敏感。

### 9.6.2.7 GPIO\_INTSTATUS

- 寄存器名：GPIO\_INTSTATUS
- 偏移量：0x40
- 默认值：0x0

位段	名称	类型	说明
31:0	GPIO_INTSTATUS	RO	中断状态

### 9.6.2.8 GPIO\_RAW\_INTSTATUS

- 寄存器名：GPIO\_RAW\_INTSTATUS
- 偏移量：0x44
- 默认值：0x0

位段	名称	类型	说明
31:0	GPIO_RAW_INTSTATUS	RO	原始中断状态

### 9.6.2.9 GPIO\_PORTA\_EOI

- 寄存器名：GPIO\_PORTA\_EOI
- 偏移量：0x4C
- 默认值：0x0

位段	名称	类型	说明
31:0	GPIO_PORTA_EOI	WO	端口 A 清除中断寄存器

### 9.6.2.10 GPIO\_EXT\_PORTA

- 寄存器名: GPIO\_EXT\_PORTA
- 偏移量: 0x50
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_EXT_PORTA	RO	外部端口 A 寄存器

### 9.6.2.11 GPIO\_LS\_SYNC

- 寄存器名: GPIO\_LS\_SYNC
- 偏移量: 0x60
- 默认值: 0x0

位段	名称	类型	说明
31:1	RSVD_GPIO_LS_SYNC		
0	GPIO_LS_SYNC	R/W	向该寄存器写入 1 会导致所有电平敏感中断同步到 pclk_intr。

### 9.6.2.12 GPIO\_ID\_CODE

- 寄存器名: GPIO\_ID\_CODE
- 偏移量: 0x64
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_ID_CODE	RO	GPIO ID 码

### 9.6.2.13 GPIO\_VER\_ID\_CODE

- 寄存器名: GPIO\_VER\_ID\_CODE
- 偏移量: 0x6C
- 默认值: 0x0

位段	名称	类型	说明
31:0	GPIO_VER_ID_CODE	RO	GPIO 组件版本

### 9.6.2.14 GPIO\_CONFIG\_REG2

- 寄存器名: GPIO\_CONFIG\_REG2
- 偏移量: 0x70
- 默认值: 0x0

位段	名称	类型	说明
31:20	RSVD_GPIO_CONFIG_REG2	RO	保留
19:15	ENCODED_ID_PWIDTH_D	RO	保留
14:10	ENCODED_ID_PWIDTH_C	RO	保留
9:5	ENCODED_ID_PWIDTH_B	RO	保留
4:0	ENCODED_ID_PWIDTH_A	RO	该寄存器的值等于 GPIO_PWIDTH_A-1。

### 9.6.2.15 GPIO\_CONFIG\_REG1

- 寄存器名: GPIO\_CONFIG\_REG1
- 偏移量: 0x74
- 默认值: 0x0

位段	名称	类型	说明
31:22	RSVD_GPIO_CONFIG_REG1	RO	保留位，读为零
21	INTERRUPT_BOTH_EDGE_TYPE	RO	该寄存器的值来自 GPIO_INT_BOTH_EDGE 配置参数。
20:16	ENCODED_ID_WIDTH	RO	该寄存器的值来自 GPIO_ID_WIDTH 配置参数。
15	GPIO_ID	RO	该寄存器的值来自 GPIO_ID 配置参数。
14	ADD_ENCODED_PARAMS	RO	该寄存器的值来自 GPIO_ADD_ENCODED_PARAMS 配置参数。
13	DEBOUNCE	RO	该寄存器的值来自 GPIO_DEBOUNCE 配置参数。
12	PORTA_INTR	RO	该寄存器的值来自 GPIO_PORTA_INTR 配置参数。
11	HW_PORTD	RO	保留
10	HW_PORTC	RO	保留

位段	名称	类型	说明
9	HW_PORTB	RO	保留
8	HW_PORTA	RO	该寄存器的值来自 GPIO_HW_PORTA 配置参数。
7	PORTD_SINGLE_CTL	RO	保留
6	PORTC_SINGLE_CTL	RO	保留
5	PORTB_SINGLE_CTL	RO	保留
4	PORTA_SINGLE_CTL	RO	该寄存器的值来自 GPIO_PORTA_SINGLE_CTL 配置参数。
3:2	NUM_PORTS	RO	该寄存器的值来自 GPIO_NUM_PORT 配置参数。
1:0	APB_DATA_WIDTH	RO	该寄存器的值来自 GPIO_APB_DATA_WIDTH 配置参数。