



曳影 1520 存储器接口 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

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

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.....	III
术语与缩略语.....	IX
1 片上 SRAM.....	1
2 LPDDR4.....	2
2.1 概述.....	2
2.2 主要特性.....	2
2.3 接口.....	3
2.4 功能说明.....	7
2.5 使用.....	11
3 eMMC/SD.....	13
3.1 概述.....	13
3.2 主要特性.....	13
3.3 接口.....	14
3.4 功能描述.....	16
3.5 使用.....	23
3.6 寄存器.....	57
4 QSPI.....	211
4.1 概述.....	211
4.2 主要特性.....	211
4.3 接口.....	212
4.4 功能描述.....	213
4.5 使用.....	224
4.6 寄存器.....	225

图表目录

图表 2-1 DDR 子系统框图	2
图表 2-2 64 位连接示意图	4
图表 2-3 32 位连接示意图	5
图表 2-4 管脚说明	5
图表 2-5 uMCTL2 示意图	8
图表 2-6 PHY 示意图	8
图表 2-7 双通道架构	9
图表 2-8 地址映射说明	9
图表 2-9 温度电压参数	11
图表 2-10 复位时序图	11
图表 2-11 DDR 初始化流程	12
图表 3-1 管脚说明表	14
图表 3-2 系统级框图	16
图表 3-3 DWC_mshc 中的卡读写数据流	19
图表 3-4 调优	20
图表 3-5 SD/eMMC 模式下的错误类型和类别	23
图表 3-6 不同卡的初始化序列	24
图表 3-7 不同卡的命令序列	24
图表 3-8 卡检测	25
图表 3-9 SD 主机控制器设置序列	26
图表 3-10 eMMC 接口主机控制器设置序列	27
图表 3-11 主机控制器时钟设置序列	28
图表 3-12 卡时钟提供和停止序列	28
图表 3-13 SD 时钟频率更改序列	29
图表 3-14 SD 卡接口检测序列	30
图表 3-15 eMMC 卡设置序列	31
图表 3-16 调优序列	32
图表 3-17 模式 1 重调流程序列	33
图表 3-18 自动调优序列	34
图表 3-19 初始化和识别 SD 卡	35
图表 3-20 初始化和识别 SD 卡（续）	36
图表 3-21 改变总线宽度序列	37

图表 3-22 SD 总线电源控制序列.....	38
图表 3-23 SD 命令发出与完成.....	39
图表 3-24 有数据传输的事务控制（不使用 DMA）	40
图表 3-25 有数据传输的事务控制（不使用 DMA，续）	41
图表 3-26 有数据传输的事务控制（使用 ADMA2）	42
图表 3-27 有数据传输的事务控制（使用 ADMA2，续）	43
图表 3-28 有数据传输的事务控制（使用 ADMA3）	44
图表 3-29 信号电压开关操作步骤.....	45
图表 3-30 改变 SD 总线速度模式.....	46
图表 3-31 SDIO 卡中断序列	47
图表 3-32 卡初始化和识别编程序列.....	48
图表 3-33 在 eMMC 设备中切换各种速度模式的编程序列	49
图表 3-34 修改 eMMC 设备数据总线宽度的编程序列	50
图表 3-35 初始化命令队列引擎	51
图表 3-36 提交和完成任务	52
图表 3-37 准备启动的编程序列	53
图表 3-38 非 DMA 模式下强制启动的编程序列	54
图表 3-39 PHY 上电和复位编程序列	55
图表 3-40 PAD 配置序列.....	56
图表 3-41 SD PHY 在 1.8V 模式下的 PAD 推荐设置.....	56
图表 3-42 可能的读写行为	57
图表 3-43 内存访问示例.....	58
图表 3-44 DMA_SA_R 寄存器字段说明	58
图表 3-45 BLOCK_SIZE_R 寄存器字段说明.....	59
图表 3-46 BLOCK_COUNT_R 寄存器字段说明	61
图表 3-47 ARGUMENT_R 寄存器字段说明	61
图表 3-48 XFER_MODE_R 寄存器字段说明	62
图表 3-49 CMD_R 寄存器字段说明	65
图表 3-50 RESP01_R 寄存器字段说明	68
图表 3-51 RESP23_R 寄存器字段说明	68
图表 3-52 RESP45_R 寄存器字段说明	69
图表 3-53 寄存器 RESP67_R 字段说明.....	69
图表 3-54 BUF_DATA_R 寄存器字段说明	70
图表 3-55 PSTATE_REG 寄存器字段说明	70
图表 3-56 HOST_CTRL1_R 寄存器字段说明	77

图表 3-57 PWR_CTRL_R 寄存器字段说明	80
图表 3-58 BGAP_CTRL_R 寄存器字段说明	82
图表 3-59 WUP_CTRL_R 寄存器字段说明	84
图表 3-60 CLK_CTRL_R 寄存器字段说明	85
图表 3-61 TOUT_CTRL_R 寄存器字段说明	88
图表 3-62 SW_RST_R 寄存器字段说明	89
图表 3-63 NORMAL_INT_STAT_R 寄存器字段说明	91
图表 3-64 ERROR_INT_STAT_R 寄存器字段说明	96
图表 3-65 NORMAL_INT_STAT_EN_R 寄存器字段说明	102
图表 3-66 ERROR_INT_STAT_EN_R 寄存器字段说明	105
图表 3-67 NORMAL_INT_SIGNAL_EN_R 寄存器字段说明	109
图表 3-68 ERROR_INT_SIGNAL_EN_R 寄存器字段说明	112
图表 3-69 AUTO_CMD_STAT_R 寄存器字段说明	116
图表 3-70 HOST_CTRL2_R 寄存器字段说明	119
图表 3-71 CAPABILITIES1_R 寄存器字段说明	124
图表 3-72 CAPABILITIES2_R 寄存器字段说明	130
图表 3-73 CURR_CAPABILITIES1_R 寄存器字段说明	134
图表 3-74 CURR_CAPABILITIES2_R 寄存器字段说明	135
图表 3-75 FORCE_AUTO_CMD_STAT_R 寄存器字段说明	136
图表 3-76 FORCE_ERROR_INT_STAT_R 寄存器字段说明	138
图表 3-77 ADMA_ERR_STAT_R 寄存器字段说明	141
图表 3-78 ADMA_SA_LOW_R 寄存器字段说明	143
图表 3-79 ADMA_SA_HIGH_R 寄存器字段说明	143
图表 3-80 PRESET_INIT_R 寄存器字段说明	144
图表 3-81 PRESET_DS_R 寄存器字段说明	145
图表 3-82 PRESET_HS_R 寄存器字段说明	146
图表 3-83 PRESET_SDR12_R 寄存器字段说明	147
图表 3-84 PRESET_SDR25_R 寄存器字段说明	148
图表 3-85 PRESET_SDR50_R 寄存器字段说明	149
图表 3-86 PRESET_SDR104_R 寄存器字段说明	150
图表 3-87 PRESET_DDR50_R 寄存器字段说明	151
图表 3-88 ADMA_ID_LOW_R 寄存器字段说明	152
图表 3-89 ADMA_ID_HIGH_R 寄存器字段说明	153
图表 3-90 SLOT_INTR_STATUS_R 寄存器字段说明	153
图表 3-91 HOST_CNTRL_VERS_R 寄存器字段说明	154

图表 3-92 EMBEDDED_CTRL_R 寄存器字段说明.....	155
图表 3-93 CQVER 寄存器字段说明	158
图表 3-94 CQCAP 寄存器字段说明	159
图表 3-95 CQCFG 寄存器字段说明	160
图表 3-96 CQCTL 寄存器字段说明	162
图表 3-97 CQIS 寄存器字段说明	163
图表 3-98 CQISE 寄存器字段说明	165
图表 3-99 CQISG 寄存器 E 字段说明	166
图表 3-100 CQIC 寄存器字段说明	168
图表 3-101 CQTDLBA 寄存器字段说明	171
图表 3-102 寄存器 CQTDLBAU 字段说明	171
图表 3-103 CQTDBR 寄存器字段说明	172
图表 3-104 CQTCN 寄存器字段说明	173
图表 3-105 CQDQS 寄存器字段说明	173
图表 3-106 CQDPT 寄存器字段说明	174
图表 3-107 CQTCLR 寄存器字段说明	174
图表 3-108 CQSSC1 寄存器字段说明	175
图表 3-109 CQSSC2 寄存器字段说明	176
图表 3-110 CQCRDCT 寄存器字段说明	177
图表 3-111 CQRMEM 寄存器字段说明	177
图表 3-112 CQTERRI 寄存器字段说明	178
图表 3-113 CQCRI 寄存器字段说明	180
图表 3-114 CQCRA 寄存器字段说明	180
图表 3-115 PHY_CNFG 寄存器字段说明	181
图表 3-116 CMDPAD_CNFG 寄存器字段说明	181
图表 3-117 DATPAD_CNFG 寄存器字段说明	182
图表 3-118 CLKPAD_CNFG 寄存器字段说明	183
图表 3-119 STBPAD_CNFG 寄存器字段说明	184
图表 3-120 RSTNPAD_CNFG 寄存器字段说明	185
图表 3-121 PADTEST_CNFG 寄存器字段说明	186
图表 3-122 PADTEST_OUT 寄存器字段说明	187
图表 3-123 PADTEST_IN 寄存器字段说明	187
图表 3-124 PRBS_CNFG 寄存器字段说明	187
图表 3-125 PHYLPBK_CNFG 寄存器字段说明	188
图表 3-126 COMMDL_CNFG 寄存器字段说明	189

图表 3-127 SDCLKDL_CNFG 寄存器字段说明	189
图表 3-128 SDCLKDL_DC 寄存器字段说明	190
图表 3-129 SMPLDL_CNFG 寄存器字段说明	191
图表 3-130 ATDL_CNFG 寄存器字段说明	192
图表 3-131 DLL_CTRL 寄存器字段说明	193
图表 3-132 DLL_CNFG1 寄存器字段说明	194
图表 3-133 DLL_CNFG2 寄存器字段说明	194
图表 3-134 DLLDL_CNFG 寄存器字段说明	195
图表 3-135 DLL_OFFST 寄存器字段说明	195
图表 3-136 DLLMST_TSTDC 寄存器字段说明	196
图表 3-137 DLLLBT_CNFG 寄存器字段说明	196
图表 3-138 DLL_STATUS 寄存器字段说明	197
图表 3-139 DLLDBG_MLKDC 寄存器字段说明	197
图表 3-140 DLLDBG_SLKDC 寄存器字段说明	198
图表 3-141 MSHC_VER_ID_R 寄存器字段说明	198
图表 3-142 MSHC_VER_TYPE_R 寄存器字段说明	199
图表 3-143 MSHC_CTRL_R 寄存器字段说明	199
图表 3-144 MBIU_CTRL_R 寄存器字段说明	201
图表 3-145 EMMC_CTRL_R 寄存器字段说明	202
图表 3-146 BOOT_CTRL_R 寄存器字段说明	204
图表 3-147 AT_CTRL_R 寄存器字段说明	206
图表 3-148 AT_STAT_R 寄存器字段说明	210
图表 4-1 从选择	213
图表 4-2 发送 FIFO 阈值 (TFT) 解码值	214
图表 4-3 接收 FIFO 阈值 (TFT) 解码值	214
图表 4-4 发送和接收 SPI 传输的 FIFO 状态	217
图表 4-5 仅发送 SPI 传输的 FIFO 状态	217
图表 4-6 仅接收 SPI 传输的 FIFO 状态	218
图表 4-7 EEPROM 读传输模式的 FIFO 状态	219
图表 4-8 往返路由延迟对 sclk_out 信号的影响	220
图表 4-9 DMA 发送数据级别 (DMATDL) 解码值	221
图表 4-10 DMA 接收数据级别 (DMARDL) 解码值	222
图表 4-11 DW_apb_ssi 主 SPI 传输流程	225
图表 4-12 可能的读写行为	225
图表 4-13 内存访问示例	226

图表 4-14 CTRLR0 寄存器字段说明	227
图表 4-15 CTRLR1 寄存器字段说明	233
图表 4-16 SSIENR 寄存器字段说明	234
图表 4-17 SER 寄存器字段说明	235
图表 4-18 BAUDR 寄存器字段说明	236
图表 4-19 TXFTLR 寄存器字段说明	236
图表 4-20 RXFTLR 寄存器字段说明	237
图表 4-21 TXFLR 寄存器字段说明	238
图表 4-22 RXFLR 寄存器字段说明	238
图表 4-23 SR 寄存器字段说明	239
图表 4-24 IMR 寄存器字段说明	242
图表 4-25 ISR 寄存器字段说明	243
图表 4-26 RISR 寄存器字段说明	245
图表 4-27 TXOICR 寄存器字段说明	247
图表 4-28 RXOICR 寄存器字段说明	248
图表 4-29 RXUICR 寄存器字段说明	248
图表 4-30 MSTICR 寄存器字段说明	249
图表 4-31 ICR 寄存器字段说明	249
图表 4-32 DMACR 寄存器字段说明	250
图表 4-33 DMATDLR 寄存器字段说明	251
图表 4-34 DMARDLR 寄存器字段说明	252
图表 4-35 IDR 寄存器字段说明	252
图表 4-36 SSI_VERSION_ID 寄存器字段说明	253
图表 4-37 DRx (for x = 0; x <= 35) 寄存器字段说明	254
图表 4-38 RX_SAMPLE_DLY 寄存器字段说明	254
图表 4-39 SPI_CTRLR0 寄存器字段说明	255

术语与缩略语

缩略语	英文全名	中文解释
ADMA	Advanced Direct Memory Access	先进的直接内存访问
CAM	Content Addressable Memory	内容可寻址存储器
DDR	Double Data Rate SDRAM	双倍速率同步动态随机存储器
DMA	Direct Memory Access	直接内存访问
eMMC	Embedded Multi Media Card	嵌入式多媒体卡
MSHC	Mobile Storage Host Controller	移动存储主控
QoS	Quality of Service	服务质量
SD Card	Secure Digital Card	安全数字卡
SDIO	Secure Digital Input and Output	安全数字输入输出

1 片上 SRAM

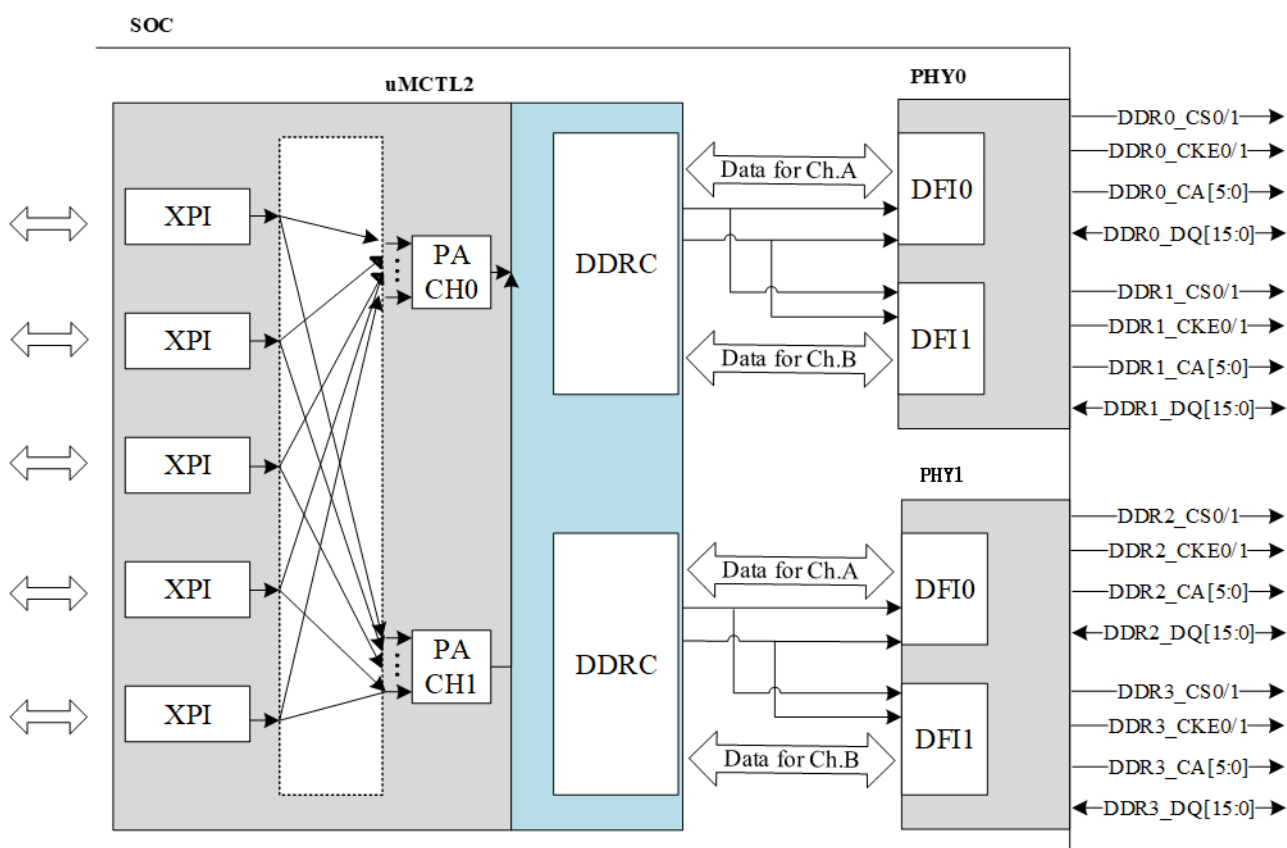
1.1 概述

默认情况下芯片提供 1.5MB+0.5MB 片上 SRAM 可供软件来扩展使用,其中后 0.5MB 地址空间仅当 DSP 复位释放且未工作时可被使用。

2 LPDDR4

2.1 概述

本文档用于说明用户在使用 DDR 子系统时的初始化步骤及注意事项。DDR 子系统结构如图表 2-1 所示，由控制器（DDRC）和两个独立的 PHY 组成，其中 DDRC 负责仲裁、QoS、地址映射等逻辑层的配置工作。PHY 完成 DFI 逻辑信号转换成 LPDDR4 的存储控制信号，且内含 PMU，可独立完成外部颗粒训练及初始化工作。



图表 2-1 DDR 子系统框图

DDRC 接口，可以接收 5 路 AXI 输入，支持标准的 AMBA AXI4 协议。

PHY 接口，最高支持 64 位 LPDDR4 或者 LPDDR4x 颗粒，最高速率 4267Mbps。

2.2 主要特性

内置 DDRC 控制器 (uMCTL2) 和 2 个独立的 PHY，实现了 SoC 系统中的 CPU 等主设备对外存 LPDDR4 SDRAM 的数据访问。主要支持如下特性：

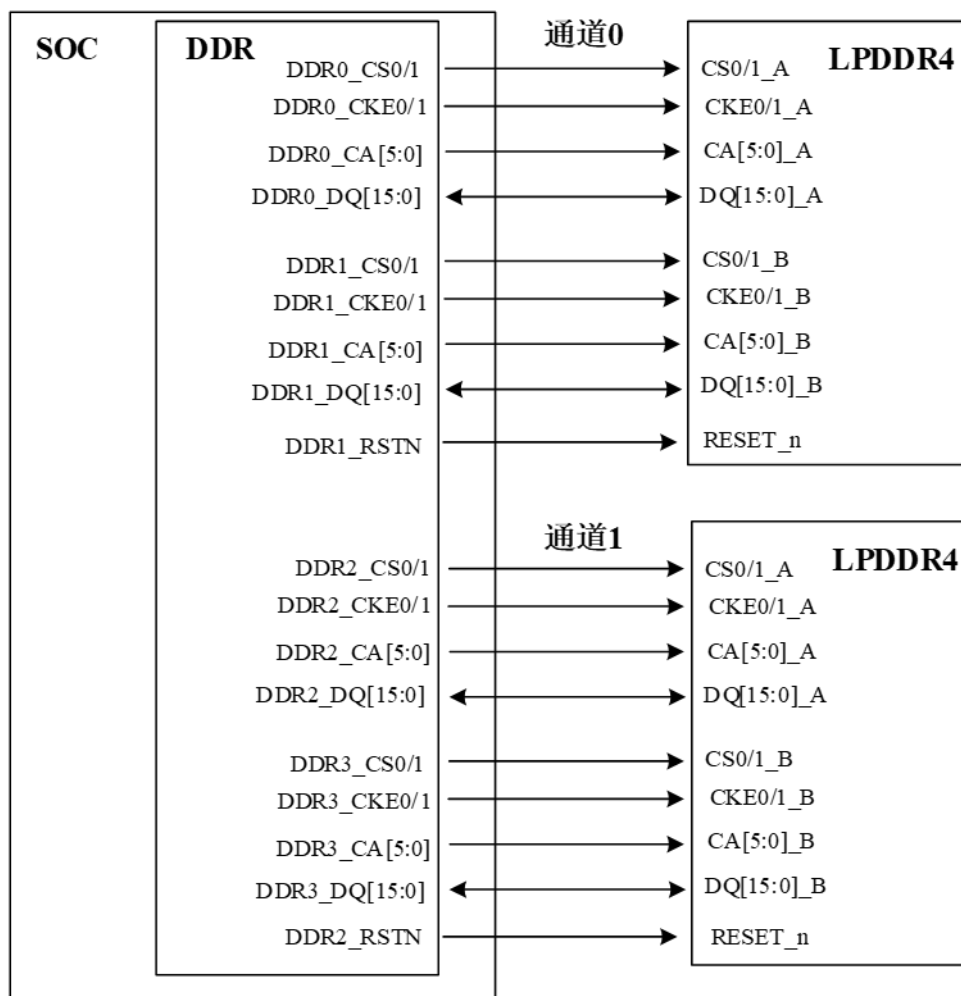
- 支持符合标准的 JEDE LPDDR4/LPDDR4x，最高速率 4267Mbps。

- 支持软件触发 ZQ 校准，也支持硬件固定间隔做 ZQ 校准。
- 支持双通道交织 64 位模式。
- 支持单通道 32 位模式。
- 支持软件配置 DQ/CA bit 交换。
- 支持周期性延迟单元训练，用于补偿电压温度带来的变化。
- PHY 内置独立的训练程序。
 - 针对 LPDDR4，PHY Master 接口支持周期性的 Retraining
 - CA 总线训练
 - 写入/读取训练
 - 读写的参考电压训练
- 可测试特性：
 - 地址数据通道支持内部回环自测
 - 延迟单元的自测试
 - PLL 锁定测试
 - ZQ 校准测试

2.3 接口

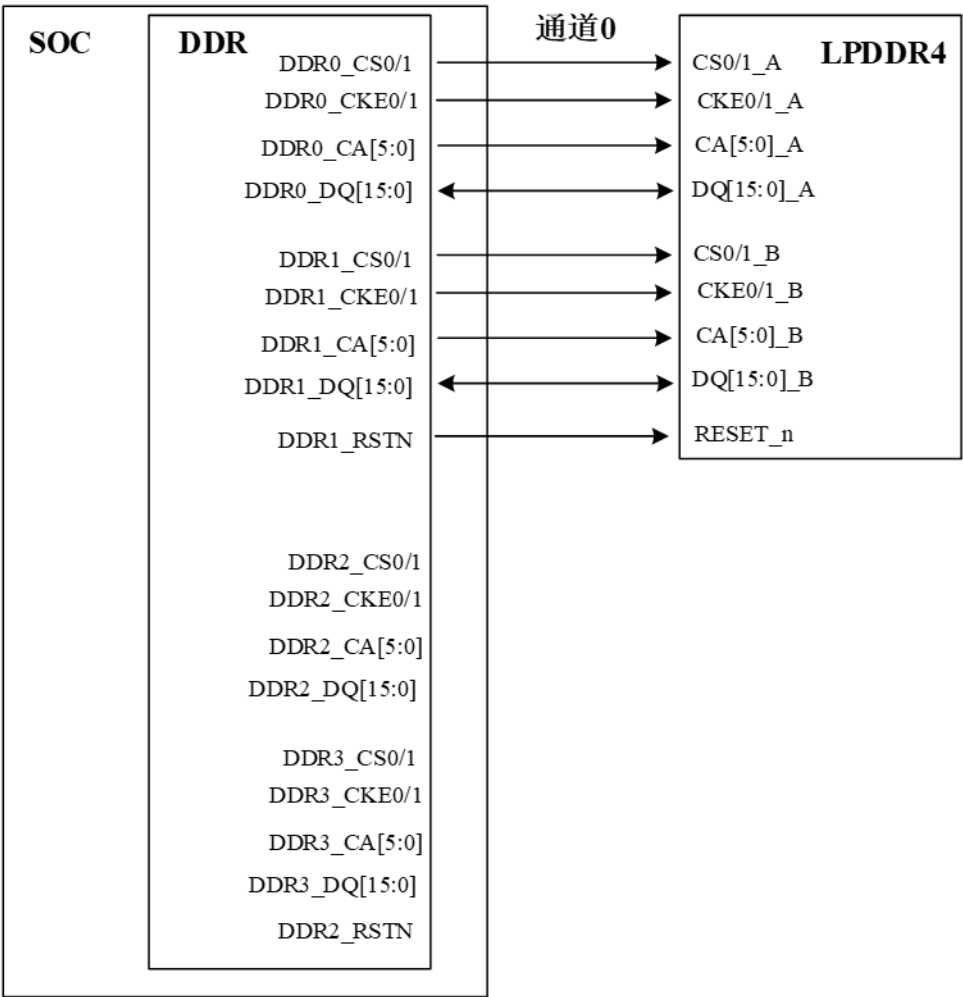
DDR 子系统支持标准的 JEDEC 的 LPDDR4 协议，同时兼容 LPDDR4x。可支持双通道 64 位，或者单通道 32 位，具体模式如下：

- 64 位 LPDDR4/LPDDR4x, 容量 4GB/8GB



图表 2-2 64 位连接示意图

- 32 位 LPDDR4/LPDDR4x, 容量 2GB/4GB



图表 2-3 32 位连接示意图

2.3.1 管脚说明

图表 2-4 管脚说明

管脚名称	方向	宽度	说明
DDR0_CKT DDR0_CKC	O	2	通道 0 低 16 位差分时钟
DDR0_CA0~5	O	6	通道 0 低 16 位的命令地址总线
DDR0_CS0~1	O	2	通道 0 低 16 位的片选信号
DDR0_CKE0~CKE1	O	2	通道 0 低 16 位的时钟使能信号
DDR0_DQSC0 DDR0_DQST0	IO	2	通道 0 的 0~7 位的双向差分数据选择信号

管脚名称	方向	宽度	说明
DDR0_DQSC1 DDR0_DQST1	IO	2	通道 0 的 8~15 位的双向差分数据选择信号
DDR0_DQ0~15	IO	16	通道 0 低 16 位的数据通道
DDR0_DMI0~1	IO	2	通道 0 低 16 位的数据屏蔽/取反信号
DDR1_CKT DDR1_CKC	O	2	通道 0 高 16 位的差分时钟
DDR1_CA0~5	O	6	通道 0 高 16 位的命令地址总线
DDR1_CS0~1	O	2	通道 0 高 16 位的片选信号
DDR1_CKE0~CKE1	O	2	通道 0 高 16 位的时钟使能信号
DDR1_DQSC0 DDR1_DQST0	IO	2	通道 0 的 16~23 位的双向差分数据选择信号
DDR1_DQSC1 DDR1_DQST1	IO	2	通道 0 的 24~31 位的双向差分数据选择信号
DDR1_DQ0~15	IO	16	通道 0 高 16 位的数据通道
DDR1_DMI0~1	IO	2	通道 0 高 16 位的数据屏蔽/取反信号
DDR1_RSTN	O	1	通道 0 的复位信号，低有效
DDR1_VREF	O	1	通道 0 的参考电压信号，可用于监控内部信号
DDR1_ZN	O	1	通道 0 的参考电阻，外接 120 欧姆电阻，精度+/-1%
DDR2_CKT DDR2_CKC	O	2	通道 1 低 16 位差分时钟
DDR2_CA0~5	O	6	通道 1 低 16 位的命令地址总线
DDR2_CS0~1	O	2	通道 1 低 16 位的片选信号
DDR2_CKE0~CKE1	O	2	通道 1 低 16 位的时钟使能信号
DDR2_DQSC0 DDR2_DQST0	IO	2	通道 1 的 0~7 位的双向差分数据选择信号
DDR2_DQSC1 DDR2_DQST1	IO	2	通道 1 的 8~15 位的双向差分数据选择信号
DDR2_DQ0~15	IO	16	通道 1 低 16 位的数据通道

管脚名称	方向	宽度	说明
DDR2_DMIO~1	IO	2	通道 1 低 16 位的数据屏蔽/取反信号
DDR3_CKT DDR3_CKC	O	2	通道 1 高 16 位的差分时钟
DDR3_CA0~5	O	6	通道 1 高 16 位的命令地址总线
DDR3_CS0~1	O	2	通道 1 高 16 位的片选信号
DDR3_CKE0~CKE1	O	2	通道 1 高 16 位的时钟使能信号
DDR3_DQSC0 DDR3_DQST0	IO	2	通道 1 的 16~23 位的双向差分数据选择信号
DDR3_DQSC1 DDR3_DQST1	IO	2	通道 1 的 24~31 位的双向差分数据选择信号
DDR3_DQ0~15	IO	16	通道 1 高 16 位的数据通道
DDR3_DMIO~1	IO	2	通道 1 高 16 位的数据屏蔽/取反信号
DDR2_RSTN	O	1	通道 1 的复位信号，低有效
DDR2_VREF	O	1	通道 1 的参考电压信号，可用于监控内部信号
DDR2_ZN	O	1	通道 1 的参考电阻，外接 120 欧姆电阻，精度+/-1%

2.3.2 电气参数

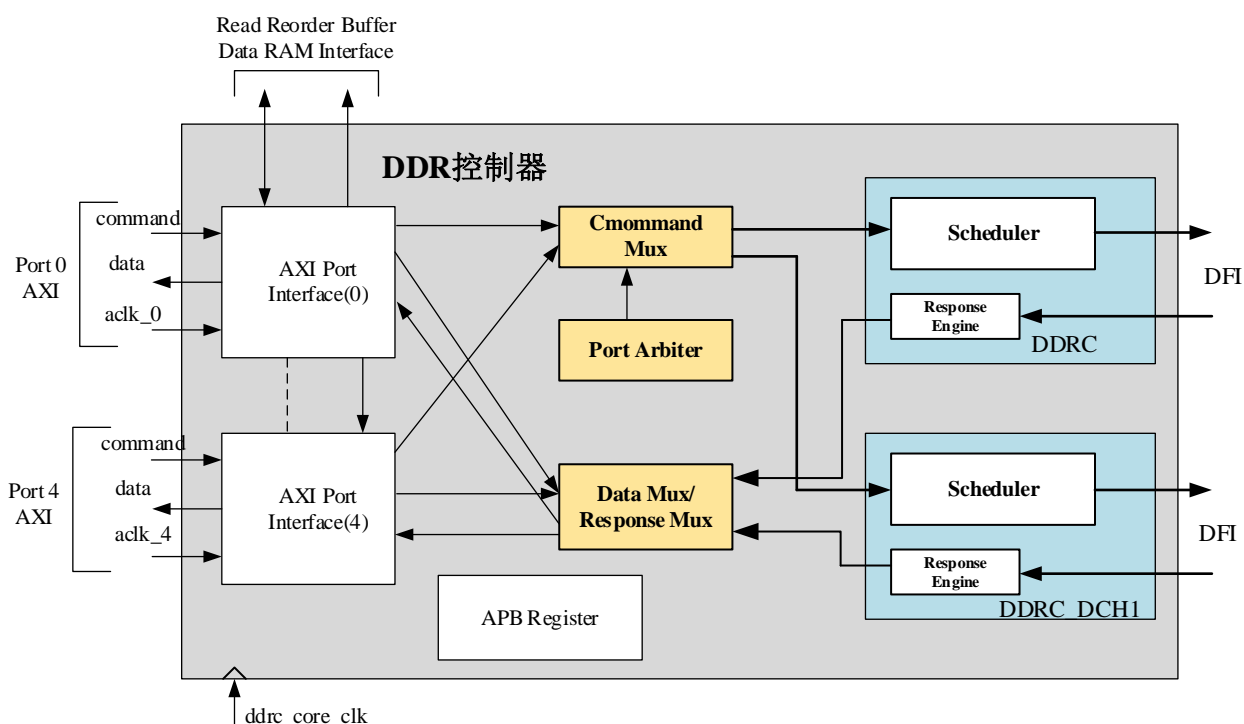
参考 JEDEC 协议说明。

2.4 功能说明

2.4.1 基本功能概述

2.4.1.1 控制器简述

控制器负责将 AXI 协议转换成 DFI 协议，内置交织器和两个 DDRC，内部框图参考图表 2-5。



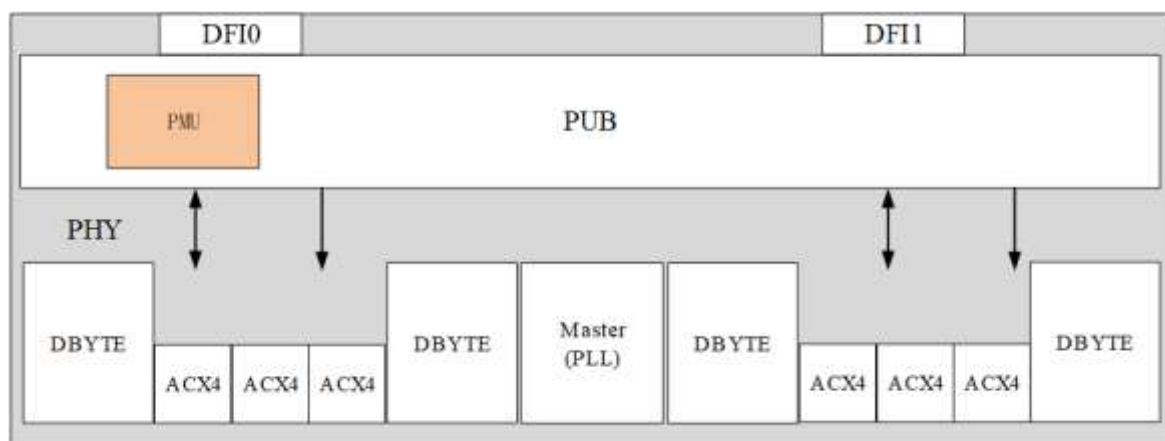
图表 2-5 uMCTL2 示意图

uMCTL2 内置如下主要模块：

- XPI 模块：该模块提供了标准的 AXI 接口。负责 AXI 数据收发、地址处理以及读数据的重排序。
- 端口仲裁（PA）模块：该模块提供各个端口的仲裁机制。
- DDR 控制器：内置内容可寻址存储器（Content Addressable Memory, CAM），用来存储收到的命令，会根据行、列、bank 地址信息来进行调度，转换成 DFI 协议送给 PHY。
- APB 寄存器模块：用于给 DDRC 配置寄存器。

2.4.1.2 PHY 简述

PHY 由软核 PUB 和硬化模块组成，硬化模块包括 4 个 DBYTE、6 个 ACX4 以及 1 个 Master 组成，具体简示结构如图表 2-6 所示。



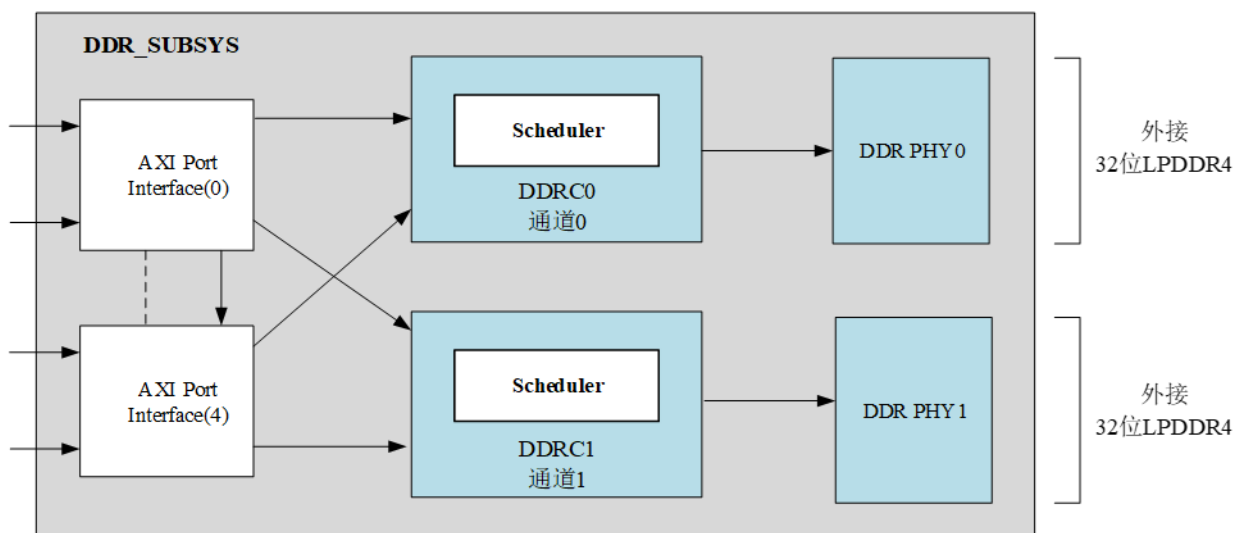
图表 2-6 PHY 示意图

- PUB: 内置 PMU, 负责完成 DDR 训练及初始化。
- DBYTE: 负责对接 SDRAM 1 个 Byte 的数据接口。
- ACX4: 负责对接 SDRAM 的 4 路地址/命令接口。
- Master: 内置 PLL, 提供时钟源。

2.4.2 地址映射说明

2.4.2.1 通道交织说明

为提升调度效率, 提升系统带宽性能, DDR 子系统采用了双通道交织的架构。用户可以根据实际需求实现不同颗粒度的交织, 当前系统默认配置为系统地址的 bit9 作为交织位, 即每隔 512B, 交织访问不同通道。同时也支持在 32 位单通道模式, 用户可以通过配置地址映射寄存器, 只访问通道 0, 外部只接 32 位 DDR 的设计方案。



图表 2-7 双通道架构

2.4.2.2 地址映射配置

通过配置寄存器 ADDRMAP0~11 来把实现不同的映射方式, 推荐的几组配置如下:

图表 2-8 地址映射说明

模式	行地址 宽度	列地址 宽度	Rank 地址	行 地址	通道 地址	Bank 地址	列地址	
	16	10	1	[15:0]	1	[2:0]	[9:7]	[6:0]
8GB 64 位模式	16	10	[32]	[31:16]	[9]	[15:13]	[12:10]	[8:2]

模式	行地址 宽度	列地址 宽度	Rank 地址	行 地址	通道 地址	Bank 地址	列地址	
	16	10	1	[15:0]	1	[2:0]	[9:7]	[6:0]
4GB 32 位模式	16	10	[31]	[30:15]	-	[14:12]	[11:9]	[8:2]
4GB 64 位模式	16	10	-	[31:16]	[9]	[15:13]	[12:10]	[8:2]
2GB 32 位模式	16	10	-	[30:15]	-	[14:12]	[11:9]	[8:2]

2.4.3 自动刷新模式

当配置 RFSHCTL3.dis_auto_refresh 为非 0 值时，DDRC 会自动周期产生刷新命令。另外 DDRC 支持预测刷新。当刷新指令被 hold1 次以上，控制器可以进行预测性刷新。即当 CAM 没有对应 rank/bank 的 transaction 时，控制器会自动插入对应刷新指令。RFSHCTL0.refresh_to_x1_x32 寄存器决定对应 transaction 不在 CAM 多长时间才会考虑插入预测性刷新，每执行一次预测刷新，tREFI 消耗 count 减一次，因此会给 critical 刷新争取时间。同时也会确保预测刷新不会超过应该的刷新次数。

设置 RFSHCTL0.per_bank_refresh=1 后，会使能 per-bank 刷新来取代 all-bank 刷新。开启 per-bank 刷新后，需要设置对应的 tREFIpb 和 tRFCpb 到对应的寄存器 RFSHTMG.t_rfc_nom_x1_x32 和 RFSHTMG.t_rfc_min。在这个模式下，控制器会根据对应当前刷新的 bank 来调度 transaction 到其他 bank 上去，以此来提升潜在的性能。为提升调度的效率，需要设置 RFSHTMG.t_trfc_nom_x1_sel=1。LPDDR4，per-bank 刷新可以按任意 bank 顺序执行。当 CAM 内没有 transaction 或者当前 bank 已经被 precharged，控制器会有优先开始执行当前 bank 的 per-bank 刷新。相关寄存器说明请参考 DWC_ddr_umctl2_databook.pdf。

2.4.4 低功耗管理

DDRC 支持多种的低功耗设计，当配置 PWRCTL.dfi_dram_clk_disable=1，会在如下场景下自动关闭 IO 时钟节省功耗。

- 自刷新
- 工作模式下，无访问请求

当配置 PWCTL.selfref_en=1，系统处于空闲状态时，即 DDRC 总线接口一定时间内无读写内存访问，DDRC 会触发进入自刷新，当系统检测到有访问请求时，DDRC 也立刻会自动退出自刷新，并打开 IO 时钟。

2.4.5 周期性训练说明

随着电压温度的变化，LPDDR4 SDRAM 参数 tDQSCK 和 tDQS2DQ 会受影响，DDR Subsys 会去周期性执行 re-training，用来补偿温度电压变化带来的影响，相关原理可以参考 [Dram Drift Compensation dwc_ddrn_phy_training_firmware_application_note.a.2020.06.pdf](#) 章节 13。具体参数的 spec，请参考各自厂商标准，如下表为美光 LPDDR4 的规格。

图表 2-9 温度电压参数

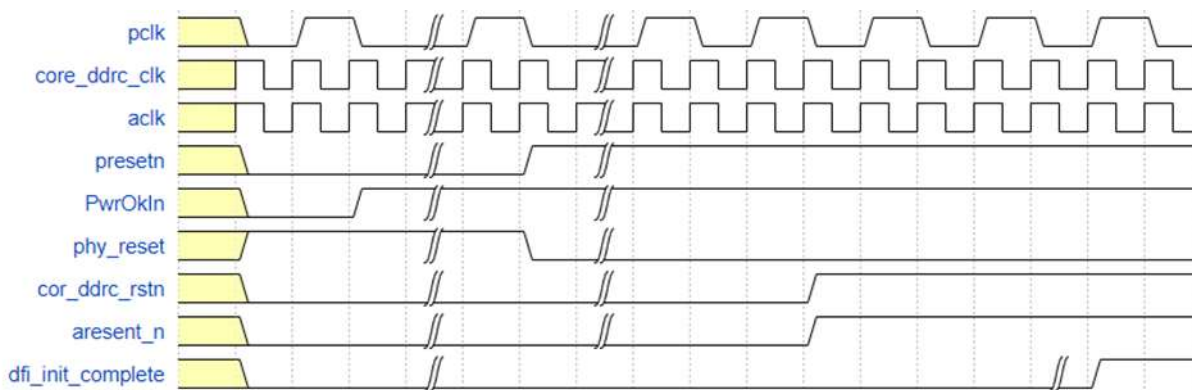
参数	名称	最小值	最大值	单位
Read DQS output access from CK_t/CK_c	tDQSCK	1500	3500	ps
tDQSCK voltage variation	tDQSCK_VOLT	-	7	ps/mv
tDQSCK temperature variation	tDQSCK_TEMP	-	4	ps/°C
DQ-to-DQS offset	tDQS2DQ	200	800	ps
tDQS2DQ voltage variation	tDQS2DQ_VOLT	-	33	ps/50mv
tDQS2DQ temperature variation	tDQS2DQ_TEMP	-	0.6	ps/°C

补偿间隔由 PHY 的寄存器 PhyMstrTrainInterval 来配置，当前默认配置为 0.7s 左右。该寄存器说明请参考 [dwc_lpddr4x_multiphy_pub_databook_2.41a.pdf](#)。

2.5 使用

2.5.1 复位时序

在执行初始化时，时钟和复位关系必须满足如图表 2-10 所示。



图表 2-10 复位时序图

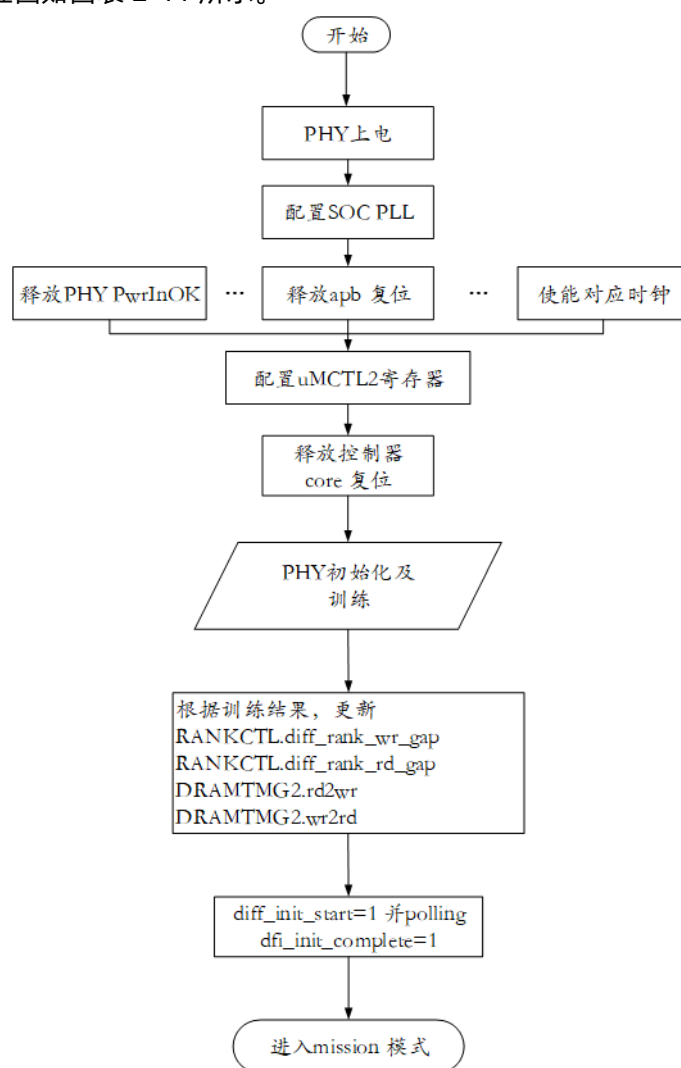
DDR 子系统内的复位均为寄存器可控。

1. 释放 PwrOKIn, ddr_sysreg.DDR_CFG0[6]=1。
2. 延迟 1us。

3. 释放 APB 复位 (ddr_sysreg.DDR_CFG0[4]=1) 和 PHY 的复位 (ddr_sysreg.DDR_CFG0[7])。
4. 配置控制器寄存器。
5. 释放控制器 core 复位 (ddr_sysreg.DDR_CFG0[5]=1) 和 AXI 端口复位 (ddr_sysreg.DDR_CFG0[13:8]=0x1f)。
6. 执行 PHY 的初始化。
7. 执行 DFI handshake, 控制器进入 mission 模式。

2.5.2 初始化流程

DDR 初始化的总流程图如图表 2-11 所示。



图表 2-11 DDR 初始化流程

3 eMMC/SD

3.1 概述

DesignWare Cores DWC_mshc 是一个高度可配置、可编程、高性能的移动存储主控制器，以 AXI 作为数据传输的总线接口。

DWC_mshc 提供了一个灵活的总线接口，使您能够将 DWC_mshc 集成到用于片上系统（SoC）的嵌入式应用程序中。DWC_mshc 有一个支持 64 位地址和数据总线的 AXI 主接口。除了支持非 DMA 模式，DWC_mshc 还支持各种 DMA 选项，如 SDMA、ADMA2 和 ADMA3。具体可以参照 SD 主机控制器标准中的说明。

DWC_mshc 主要针对主机控制器和读卡器应用程序。在这种情况下，DWC_mshc 向其发送或接收数据的设备存储卡通常是用于 FLASH 大容量存储的设备。DWC_mshc 针对以下应用程序和系统进行了优化：

- 便携式电子设备
- 高速存储应用

具体来说，DWC_mshc 主要针对这些设备：

- 手机和笔记本电脑
- 数码相机和摄像机
- 打印机设备
- 嵌入式应用程序

3.2 主要特性

DWC_mshc 的主要特性如下：

- 支持 SD 内存和 SD 输入/输出（SDIO）数字接口协议，并且符合 SD HCI 规范。
- 为 eMMC 传输使用相同的 SD-HCI 寄存器集。
- 支持 eMMC 协议，包括 eMMC 5.1
- 支持 SD-HCI 主机版本 4 或更低版本。
- 支持 SD 和 eMMC 模式下以下类型的数据传输：
 - CPU
 - SDMA
 - ADMA2
 - ADMA3
- 支持独立控制器时钟、从接口时钟和主接口时钟。
- 如果主机控制器不工作，支持控制器基础时钟门控。
- 支持上下文感知功能时钟门控。

- 如果主机控制器不工作，应用程序可以门控从接口时钟。
- 数据缓冲
 - 256 + 32 缓冲深度
 - 自动打包/解包数据，以适应缓冲区的宽度。
- 中断输出
 - 组合中断输出
 - 支持中断使能和屏蔽。
- 支持命令队列引擎（CQE），兼容 eMMC CQ HCI。
 - 可编程调度器的任务执行算法选择
 - 支持数据预取，用于连续写操作。
- 支持调优。
 - 使用 CMD19（SD）或 CMD21（eMMC）进行 SD/eMMC 调优。
 - 模式 1 重新调优-主机驱动维护重新调优定时器
 - 完全软件驱动的调优/重新调优操作
 - 自动调优或模式 3 重新调优

3.3 接口

在 SoC 系统中，有 3 个 DWC_mshc 的实例，eMMC/SDIO0/SDIO1。eMMC 的数据宽度为 8 位，支持 HS400。SDIO0 和 SDIO1 的数据宽度为 4bit，支持 SDR104。eMMC/SDIO0/SDIO1 的 PADs 如图表 3-1 所示。

图表 3-1 管脚说明表

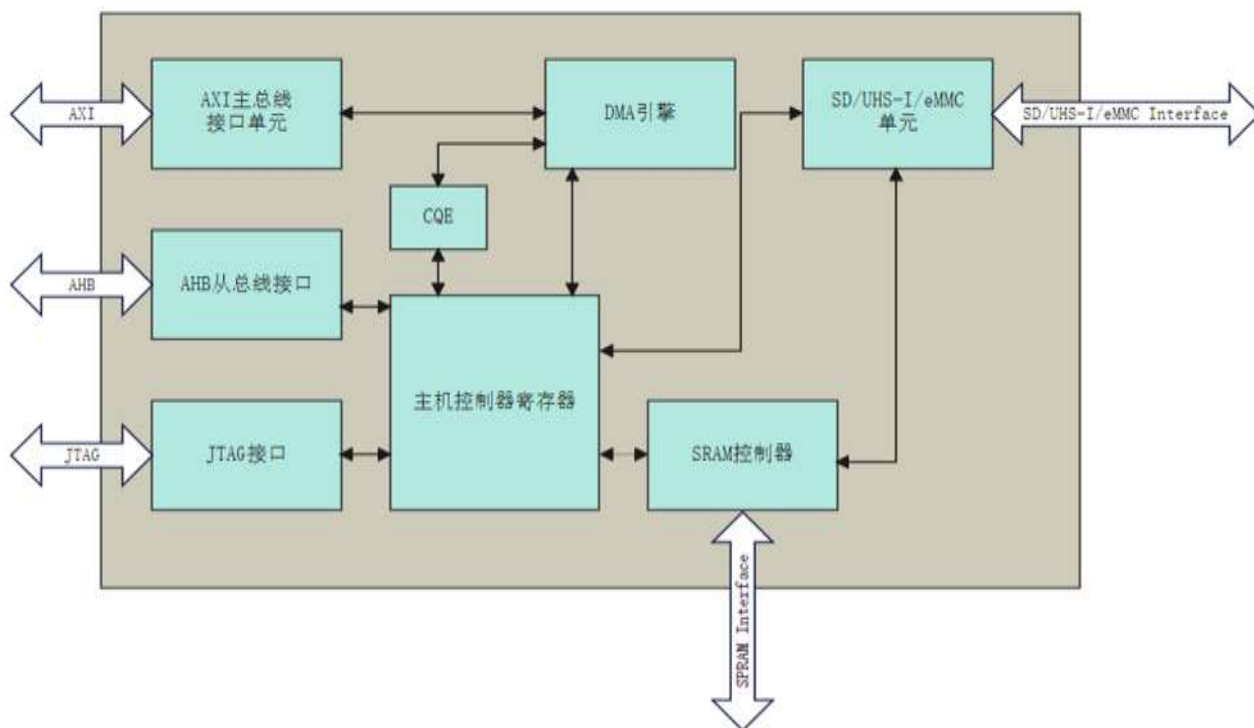
管脚名称	方向	宽度	说明
EMMC_CLK	O	1	eMMC 时钟信号，1.8V 电压域
EMMC_CMD	IO	1	eMMC 命令输入/输出信号，1.8V 电压域
EMMC_DAT0	IO	1	eMMC DAT0 输入/输出信号，1.8V 电压域
EMMC_DAT1	IO	1	eMMC DAT1 输入/输出信号，1.8V 电压域
EMMC_DAT2	IO	1	eMMC DAT2 输入/输出信号，1.8V 电压域
EMMC_DAT3	IO	1	eMMC DAT3 输入/输出信号，1.8V 电压域
EMMC_DAT4	IO	1	eMMC DAT4 输入/输出信号，1.8V 电压域
EMMC_DAT5	IO	1	eMMC DAT5 输入/输出信号，1.8V 电压域
EMMC_DAT6	IO	1	eMMC DAT6 输入/输出信号，1.8V 电压域

管脚名称	方向	宽度	说明
EMMC_DAT7	IO	1	eMMC DAT7 输入/输出信号, 1.8V 电压域
EMMC_DS	I	1	该信号由设备产生, 仅在 HS400 模式下使用。该信号的频率跟随 EMMC_CLK 的频率, 1.8V 电压域。
EMMC_RSTN	O	1	eMMC 卡复位信号输出, 1.8V 电压域
SDIO0_CLK	O	1	SDIO0 时钟信号, 1.8V/3.3V 电压域
SDIO0_CMD	IO	1	SDIO0 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO0_DAT0	IO	1	SDIO0 DAT0 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO0_DAT1	IO	1	SDIO0 DAT1 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO0_DAT2	IO	1	SDIO0 DAT2 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO0_DAT3	IO	1	SDIO0 DAT3 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO0_DET_N	I	1	卡检测信号 该信号为 0 时, 表示卡已连接。 如果启用了中断, 该信号从 0 变为 1 时, 卡插入中断产生。
SDIO0_WPRTN	I	1	SD 卡写保护信号 该信号为 0 时, 表示卡处于写保护状态。
SDIO1_CLK	O	1	SDIO1 时钟信号, 1.8V/3.3V 电压域
SDIO1_CMD	IO	1	SDIO1 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO1_DAT0	IO	1	SDIO1 DAT0 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO1_DAT1	IO	1	SDIO1 DAT1 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO1_DAT2	IO	1	SDIO1 DAT2 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO1_DAT3	IO	1	SDIO1 DAT3 命令输出/输出信号, 1.8V/3.3V 电压域
SDIO1_DET_N	I	1	卡检测信号 该信号为 0 时, 表示卡已连接。 如果启用了中断, 该信号从 0 变为 1 时, 卡插入中断产生。
SDIO1_WPRTN	I	1	SD 卡写保护信号 该信号为 0 时, 表示卡处于写保护状态。

3.4 功能描述

3.4.1 架构概述

本节说明 DWC_mshc 控制器的主要功能构建块，如图表 3-2 所示。



图表 3-2 系统级框图

以下是移动存储主机控制器的各个模块：

- 主总线接口单元（MBIU）

这个 MBIU 实现在 AMBA 可扩展接口总线（AXI）上传输数据的逻辑。AXI 接口通过 AXI 主总线接口在系统内存中传输数据。

- AHB 从总线接口（SBI）模块

AHB SBI 模块通过使用外部 AMBA 高性能（AHB）总线实现主要访问 DWC_mshc 寄存器的逻辑。此模块仅支持用于寄存器访问的小字节序方案。

- JTAG 接口

JTAG 接口未配置使用。

- DMA 引擎

DMA 引擎单元处理 DWC_mshc 和系统内存之间的数据传输。

- 主机控制器寄存器

主机控制器寄存器单元由《SD 规格 A2 部分 SD 主机控制器标准规格 4.20a 版本》规定的标准 SD 主机控制器寄存器组成，它包括了符合 JEDEC eMMC 5.1 HCI 规范的命令队列寄存器。

这些寄存器在三个时钟域中实现，即 AXI 主总线接口时钟 (aclk)、AHB 从总线接口时钟 (hclk) 和控制器基本时钟 (bclk)。非传输相关寄存器在 AHB 从时钟域中实现，从而实现了控制器基本时钟的节电门控。

- SRAM 控制器（数据包缓冲接口）

SRAM 控制器接口主机和事务控制器单元（SD/UHS-I/eMMC 单元）的数据包缓冲区。

- SD/UHS-I/eMMC 单元

SD/UHS-I/eMMC 单元管理 SD/eMMC 接口协议。

- 命令队列引擎 (CQE)

该模块实现命令排队，包括：

- 具有可按优先级执行任务能力的任务调度程序
- 描述符获取的控制逻辑
- 任务提交和执行的控制和排序
- 状态轮询
- CQE 操作专用的定时器和计数器

3.4.2 设备接口和卡接口

3.4.2.1 SD 卡接口

SD 单元负责管理 SD 接口协议。

本单元的主要特点是：

- 生成默认速率 (Default Speed, DS) / 高速速率 (High Speed, HS) / UHS-I 命令和数据包。
- 生成 CRC 并对命令和数据包进行检查。
- 处理数据包超时。
- 支持 1 位和 4 位 DAT 模式。
- 处理 SDIO 卡中断。
- 支持 DS、HS、SDR12、SDR25、SDR50 和 SDR104 速度模式。

3.4.2.2 eMMC 卡接口

eMMC 单元负责管理 eMMC 接口协议。

本单元的主要特点如下：

- 生成 eMMC 总线命令和数据包。
- 生成 CRC 并对命令和数据包进行检查。
- 处理数据包超时。
- 支持 1 位、4 位和 8 位 DAT 模式。
- 支持 legacy、HS SDR、HS DDR、HS200 和 HS400 模式。

3.4.3 主总线接口

DWC_mshc 利用 MBIU 来启动与系统内存的数据通信。系统内存读/写传输请求是由 DWC_mshc 的 DMA 通过本机主接口发起的，本机主接口是通用主（GM）接口。不管选择的系统总线协议是什么，通用接口都用于在 DMA 和 MBIU 之间进行交互。DMA 传输利用读描述符从系统内存传输读写数据。

通用主接口操作数据包级或描述符级的传输。也就是说，每个请求代表一个描述符或数据包大小的传输。相反，AXI 衬垫根据接口的需要将 GM 请求分解为多个 AXI 传输。

等待周期可能出现在主读写数据路径上。如果 RAM 控制器没有足够的带宽来同服务事务层单元（TLU）和 DMA，就会发生这种情况。在 CQE 期间，当 DMA FIFOs 也被用于消息传递时也会发生这种情况。

AXI 主接口是本地主接口（GM）和 AXI 总线之间的一个衬垫。DMA 读写访问由 DWC_mshc 控制器通过 AXI 主接口发起，读取描述符和读写传输数据。读写通道是独立的，可以同时处于活动状态。但是，如果读访问依赖于写访问完成，则控制器收到来自 AXI 的 DMA 写响应后才会发出 DMA 读。

3.4.4 AHB 从总线接口

AHB 从接口支持以下特性：

- 完全兼容 AMBA 2.0 AHB slave
- 单次突发传输
- 32 位写或读传输到 DWC_mshc 寄存器
- 传输响应：OKAY、ERROR 和 RETRY 响应

3.4.5 DMA 引擎

DMA 引擎单元处理 DWC_mshc 和系统内存之间的数据传输。本单元的主要特点如下：

- 支持 SDMA/ADMA2/ADMA3 模式。
- 获取描述符和数据。使用同一个 DMA 引擎将数据传输和描述符获取交错起来。当 DMA 在任务执行期间移动数据（对于 CMD46 和 CMD47），这允许新的任务描述符获取（对于 CMD44 和 CMD45）。
- AXI 事务 ID 0 用于移动数据，AXI 事务 ID 1 用于获取任务描述符。
- eMMC 连续写命令预取数据
- 将收到的数据包写回系统内存。

3.4.6 SRAM 控制器

SRAM 控制器连接主机和事务控制器单元（SD/UHS-I/eMMC）的数据包缓冲区。缓冲区深度配置为 256 + 32，其中每个地址的大小等于 AXI 数据宽度（64 位）。

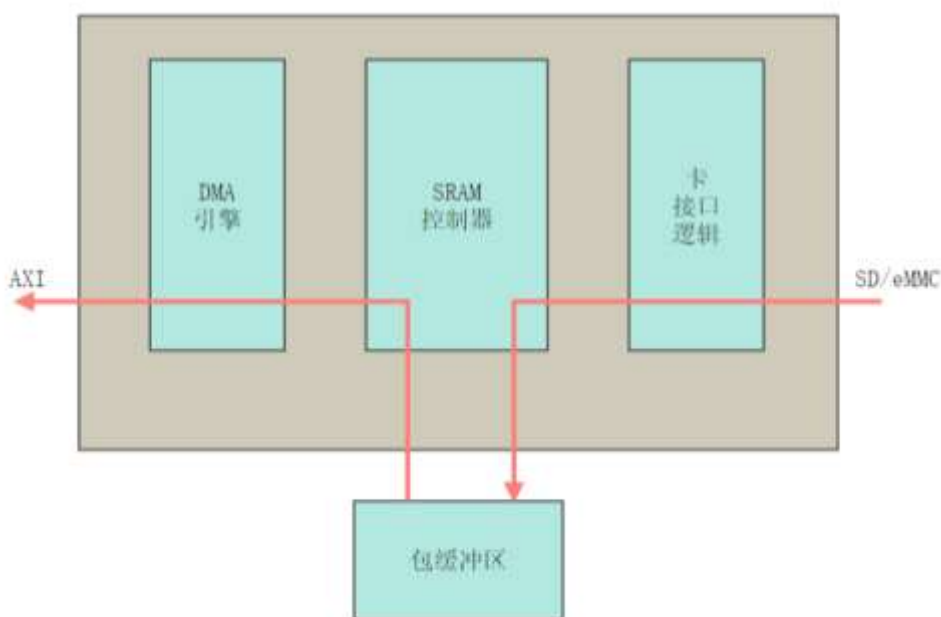
由于对卡的写和读传输不是同时发生的，因此用一个读写操作的共享缓冲区来节省空间。在数据传输命令握手期间，对命令寄存器的读/写位进行采样并存储。这个内部位定义了 DWC_mshc 处于读模式还是写模式。

数据包缓冲区使用与控制器基本时钟（bclk）同步的单一时钟单端口 RAM。

3.4.7 DWC_mshc 中的卡读写数据流

图表 3-3 解释了数据流如何从卡接口（SD/SDIO/eMMC）通过用于卡读传输的数据包缓冲区流向 AXI 接口。从卡接口接收到的数据被写入数据包缓冲区。当接收到一个数据块（SD/SDIO/eMMC）时，DMA 开始从数据包缓冲区读取数据，并将数据传输给系统。对于卡写传输，数据流是反向流动的。DMA 将数据写入数据包缓冲区，随后由卡接口逻辑读取。

DMA 和卡接口逻辑可以同时工作，因为读取和写入到数据包缓冲区可以交错。对于卡读取，在卡接口逻辑接收当前块的同时，DMA 可以发送之前的块。而对于卡写入，在卡接口逻辑发送之前块的同时，DMA 可以将当前块写入数据包缓冲区。



图表 3-3 DWC_mshc 中的卡读写数据流

3.4.8 在 DWC_mshc 中调优

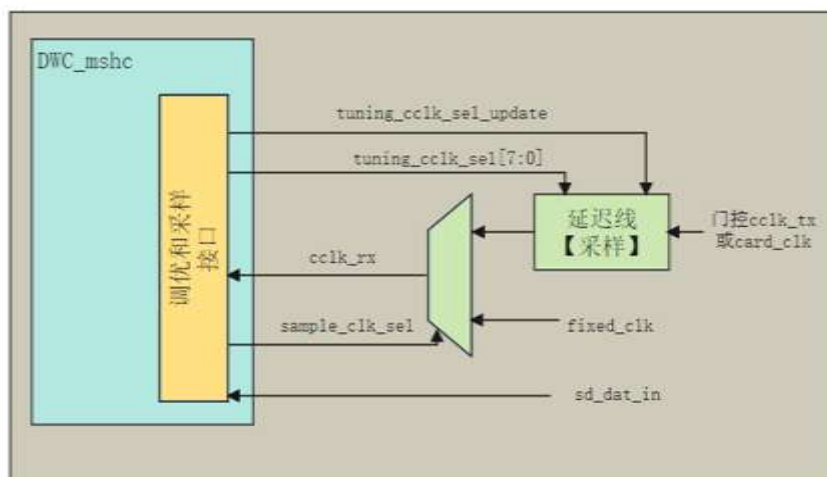
WC_mshc 兼容 SD 4.2 和 eMMC 5.1 协议。这两种协议都有速度模式，在这种模式下，输入数据必须用可编程采样在时钟上进行采样。因此，需要一种称为调优的方法来识别数据可以鲁棒采样的正确相位。

以下是 DWC_mshc 支持的调优和重调优模式：

- 基于 CMD21（eMMC）或 CMD19（SD）的调优流程
- 模式 1（软件辅助）重新调优流程
- 模式 3（硬件管理）重新调优（即自动调优）

- 软件驱动调优流程

如图表 3-4 所示，DWC_mshc 信号 `tuning_cclk_sel` 和 `tuning_cclk_sel_update` 的输出由内部调优引擎驱动，必须用于 DelayLine 的相位控制。这些输出被调优引擎用来改变 `cclk_rx` 时钟的传入相位，以便它可以找到 `sd_dat_in` 端口上的传入数据可以鲁棒采样的相位。



图表 3-4 调优

需要调优的高速 SDR 模式有 eMMC HS200 和 SD SDR104。SD 或 eMMC 调优序列是一种软件辅助序列，由硬件控制 DelayLine 并计算相位以实现鲁棒采样，并由软件确定调优和启动调优序列的必要性。

3.4.8.1 模式 1 重新调优

DWC_mshc 按照 SD HCI 规范实现 Mode 1 的重新调优，甚至可以在 eMMC 操作模式下使用。模式 1 的重新调优类似于调优序列，可以使用 `HOST_CTRL2_R.SAMPLE_CLK_SEL` 寄存器字段值来区分这两者。SD HCI 规范建议软件必须为传输的每 4MB 数据调用重新调优序列，或者根据软件维护的重新调优计时器调用重新调优序列。

在重新调优期间，调优引擎使用在初始调优序列中收集的信息测试采样窗口的有效性。如果窗口未改变，重新调优序列的时间很短（总 `CMD21/CMD19` 迭代等于 `Pass-steps` 的数量+1）。在任何漂移的情况下，重新调可能导致更多的 `CMD21/CMD19` 迭代（等于 DelayLine 的总步长）。

以下是模式 1 重新调优的可选调优方案：

- 基于阈值的选择调优模式

此模式允许调优引擎选择满足阈值标准的第一个完整的采样窗口。在这种模式下，完整采样窗口优于部分采样窗口。当完整的采样窗口满足阈值条件时，调优将停止。当所有抽头都被解析时，调优也会停止。

- 向后兼容调优方案

这种模式允许调优引擎像在 1.50a 和 DWC_mshc 的早期版本中那样工作。在这种模式下，完整采样窗口优于部分采样窗口。当找到一个完整的采样窗口或当所有抽头都被解析时，调优将停止。采样窗口宽度没有最小阈值。

- 最大采样窗口调优方案

这种模式允许调优算法通过延迟线的所有抽头来识别最大可用的采样窗口。在此模式下，只有当所有抽头都被解析时，调优才会停止。该算法选取最大的采样窗口。

3.4.8.2 软件优化

DWC_mshc 实现了一个完全由软件控制的调优序列，称为软件调优。本软件可以设置 AT_CTRL_R.SW_TUNE_EN 位，用于控制 tuning_cclk_sel、tuning_cclk_sel_update 和 sample_clk_sel 信号，这些信号构成调优和数据采样接口。当 AT_CTRL_R.SW_TUNE_EN 设为 1 时，软件可以写入 AT_STAT_R.CENTER_PH_CODE 位，用于更改 tuning_cclk_sel 的值。当使用软件调优时，软件必须计算采样时钟的中心相位和周期性重调。

3.4.8.3 自动调优或模式 3 重调优

自动调优是一种硬件管理的重新调优特性，它遵循 SD HCI Mode3 重新调优过程。自动调优不需要主机软件为每 4MB 数据传输（SD HCI）重调采样时钟。建议在 SD 和 eMMC 模式下使用调优采样时钟，以确保在高 SDR 速度下（如 SDR104 和 HS200）能够轻松地进行定时关闭和稳健运行。

以下是自动调优特性的一些功能：

- 初始化时需要基于 CMD19（SD）或 CMD21（eMMC）的调优

一旦初始化，自动调优引擎可以检测漂移，并在数据传输过程中计算修正。

- 不依赖 SD/eMMC 的调优命令进行漂移检测或修正

这是必要的，因为自动调优发生在定期数据/命令传输的后台。

- 检测控制器所有活动数据线的漂移。

- 提供软件可选的修正间隔。

修正可以应用于块边界或应用于数据传输之间。

- 处理漂移

漂移监控窗口是硬件发起的。如果需要，软件可以对 AT_CTRL_R 寄存器进行编程，以改变初始化的值。如果自动调优未能修正检测到的漂移，则设置调优错误中断。

以下是自动调优操作的过程：

1. 软件必须基于 CMD19（适用于 SD）或 CMD21（适用于 eMMC）发起调优序列来初始化自动调优引擎。
2. 初始化之后，自动调优引擎默认被激活用于所有数据的读取传输。
3. 自动调优引擎使用 DM DelayLine 来生成 drift_cclk_rx。
4. drift_cclk_rx 与 cclk_rx 使用相同的源代码生成；，但是它的相位由自动调优引擎的 autotune_cclk_sel 输出控制。
5. 自动调优引擎使用 drift_cclk_rx 上采样的数据来检查数据线上的漂移。如果发生以下数据错误，则检测到漂移：

- 数据 CRC 错误
- 启动位错误
- 结束位错误

这些漂移检查只有在 HS200/SDR104 模式下进行读传输时才有效。

6. 在漂移检测中，自动调整引擎扫描 DM DelayLine 相位，直到它找到一个没有漂移的采样窗口。
7. 新的采样窗口范围允许自动调整引擎计算采样相位的修正。
8. 在软件选择的修正间隔内，使用 tuning_cclk_sel 输出将计算出的修正应用于采样 DelayLine。

根据 SD 规范，定时器负载值是一种能力，即配置值，而不是可编程值。这个值必须由软件维护，而不是由硬件维护。软件必须维护重新调优计时器，必须在数据传输结束时启动该计时器，并在新的数据传输开始时复位它。当计时器到期时，软件必须调用基于命令的调优过程。

3.4.8.4 中断

在 DWC_mshc 中，中断是基于各种事件生成的。

DWC_mshc 提供了两种中断输出：

- intr
- wakeup_intr

在主动模式下，中断信号必须用作不同事件的中断。在备用模式下，wakeup_intr 必须用于识别任何唤醒事件，如卡移除或插入或 SDIO 卡中断。中断是电平型，也就是说，中断保持生效（高电平），直到被主机或软件清除。

中断状态寄存器指示导致中断产生的事件。通过设置相应的掩码位，可以防止各个事件在中断信号上生效中断。

只有在设置了相应的中断状态使能并观察到中断事件时，才在中断状态寄存器中设置此位。只有在中断信号使能（interrupt signal enable）已设置无关位的情况下，中断状态寄存器中设置的这个位才能使中断失效。

主机驱动程序负责在主机系统进入备用模式时使能唤醒信号和禁用中断信号，并在主机系统进入主动模式时禁用唤醒信号和启用中断信号。主机驱动程序不能同时启用两者。使用中断信号使能（interrupt signal enable）启用中断信号，使用唤醒事件使能（wakeup event enable）启用唤醒信号。

3.4.8.5 错误检测

图表 3-5 列出了 SD/eMMC 模式下的错误类型和类别。

图表 3-5 SD/eMMC 模式下的错误类型和类别

错误类型	类别
命令错误	<ul style="list-style-type: none">■ 命令超时错误■ 命令 CRC 错误■ 命令结束位错误■ 命令索引错误■ 命令冲突错误■ 响应错误
自动命令错误	<ul style="list-style-type: none">■ 命令不是由 Auto CMD12 发出的错误■ 自动命令超时错误■ 自动命令 CRC 错误■ 自动命令结束位错误■ 自动命令索引错误■ 自动命令冲突错误■ 自动命令响应错误
数据错误	<ul style="list-style-type: none">■ 数据超时错误■ 数据 CRC 错误■ 数据结束位错误■ ADMA 错误■ 调优错误

3.5 使用

以下是使用该卡开始数据传输之前所需的编程序列：

1. 开始向 DWC_mshc 提供 hclk 并应用它的复位 hresetn。这个时钟用于卡检测逻辑和访问 DWC_mshc 中的大多数寄存器。
2. 通过读取当前状态寄存器（PSTATE_REG）检查卡是否已经插入。此步骤仅适用于可拆卸卡。如果卡已经插入（eMMC），请执行步骤 4。

3. 等待卡插入中断。此步骤仅适用于可拆卸卡。卡片检测序列在“卡检测”一节中有说明。
4. 按照“主机控制器设置序列”一节中的说明，为 DWC_mshc 进行基本设置。它涉及的设置，如卡的总线供电电压等级，超時計数器的值，设置时钟生成的参数。提供了 SD 卡（“SD 接口”）和 eMMC 卡（“eMMC 设备”）不同的设置序列。该步骤也可以在步骤 1 完成后立即执行。
5. 启用 DWC_mshc 中可用的输入时钟（除了 hclk）。启用内部时钟（aclk、bclk、tmclk 和 cqetmclk），然后启用卡时钟（cclk_tx/cclk_rx），具体可参见“主机控制器时钟设置序列”一节。在这个阶段，DWC_mshc 已经准备好与卡通信。
6. 根据“SD 卡接口检测”一节的编程序列，确定连接到 DWC_mshc 的卡类型（SD 卡或 eMMC 卡）。如果 DWC_mshc 连接到 eMMC 卡，则必须执行“eMMC 卡接口设置”一节中说明的编程序列。这是一个针对 eMMC 卡的简化序列，因为该卡类型总是已知。两种编程序列都为卡提供电源和时钟。
7. 卡上电后，对卡进行初始化。根据所连接卡的类型，必须执行如图表 3-6 所示的相应序列。
8. 主机控制器和卡都初始化以后，主机可以开始向卡发送命令来执行数据传输。控制命令和数据命令是可以向卡发出的两种命令类型。控制命令用于读或写卡中的任一寄存器。数据命令用于向存储卡写入或读取数据。图表 3-7 列出了 SD 卡和 eMMC 卡的控制命令和数据命令序列。

图表 3-6 不同卡的初始化序列

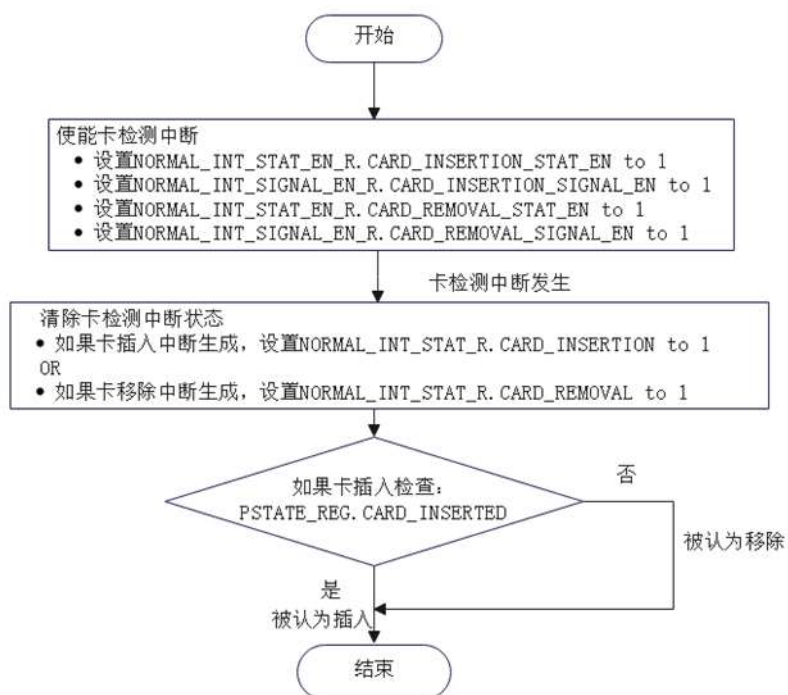
卡类型	卡初始化序列
SD	初始化和识别 SD（“初始化和识别 SD 卡”）
eMMC	初始化和识别 eMMC 卡（“初始化和识别 eMMC 设备”）

图表 3-7 不同卡的命令序列

卡类型	命令序列	
SD/eMMC	控制命令	发出没有数据传输的命令（“发出无数据传输的命令”）
	数据命令	发出有数据传输的命令（“发出有数据传输的命令”）

3.5.1 卡检测

图表 3-8 显示了卡的检测序列。图表 3-8 中概述的过程适用于 SD 卡和 SDIO 卡。eMMC 设备不需要这个编程序列，因为它是不可移动的。

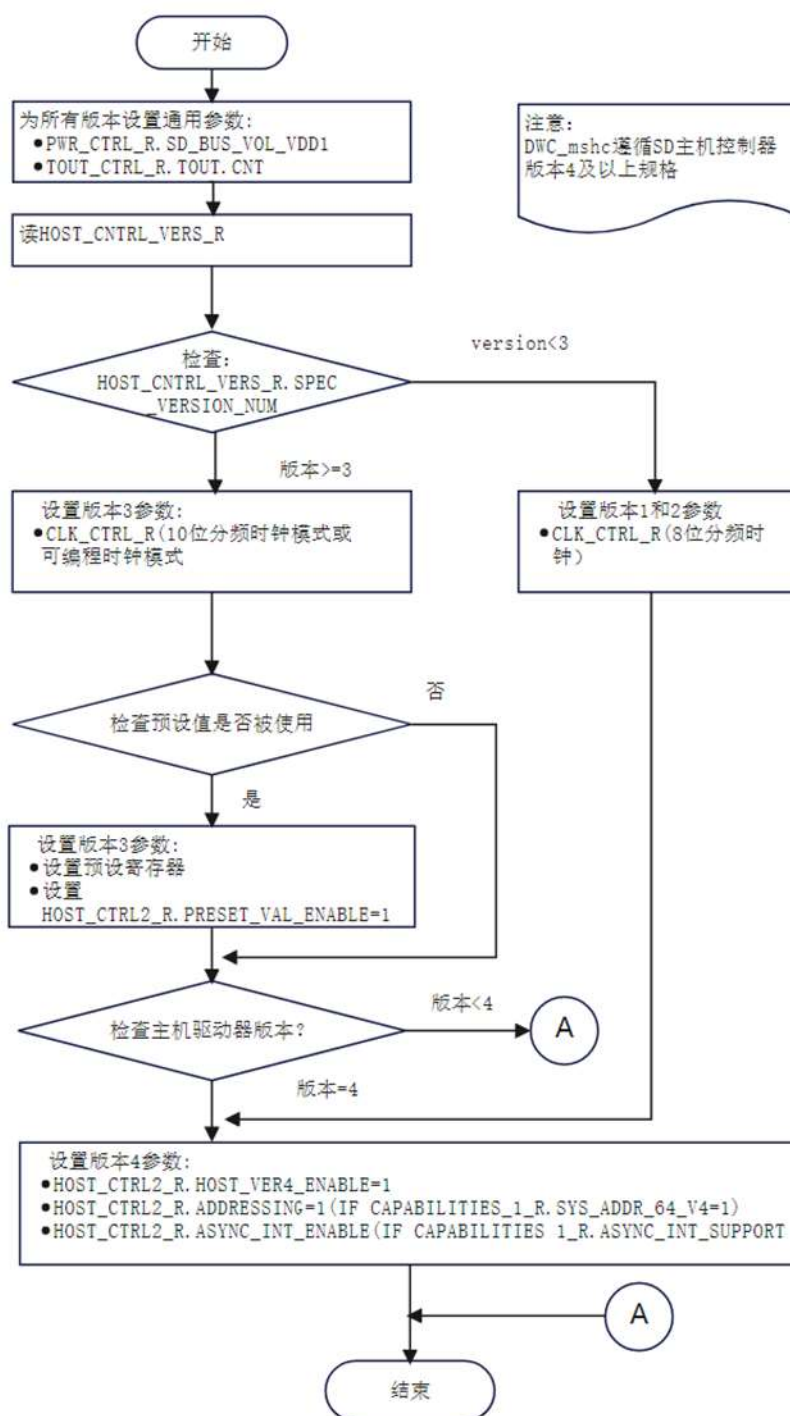


图表 3-8 卡检测

3.5.2 主机控制器设置序列

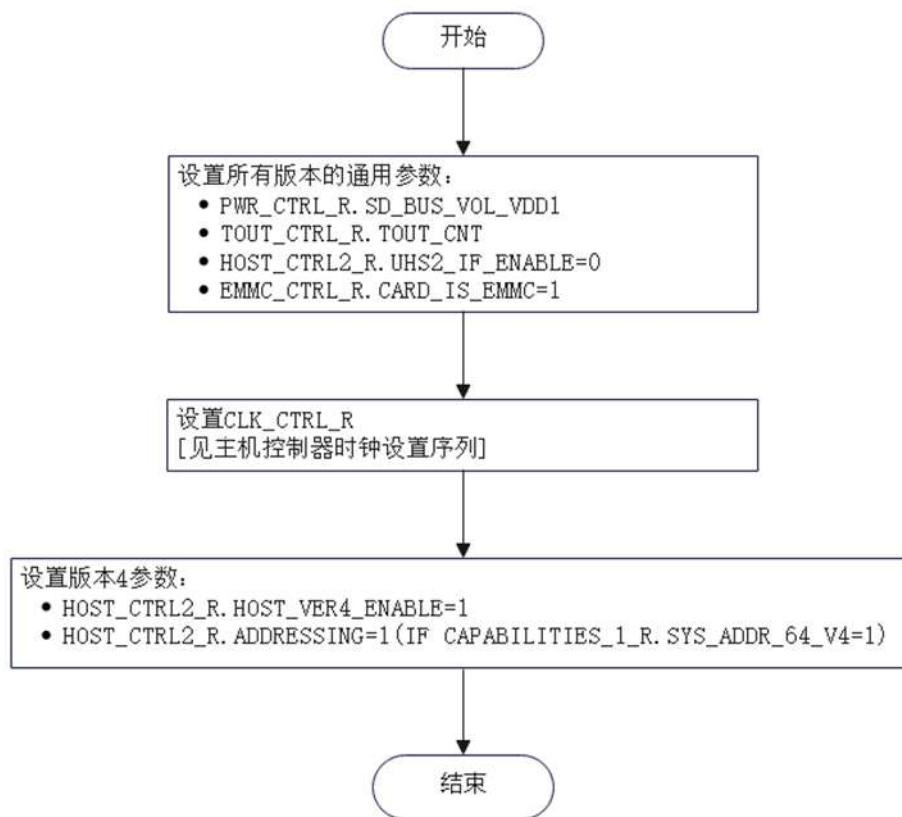
本节讨论 SD 接口和 eMMC 设备的主机控制器设置序列。

3.5.2.1 SD 接口



图表 3-9 SD 主机控制器设置序列

3.5.2.2 eMMC 设备



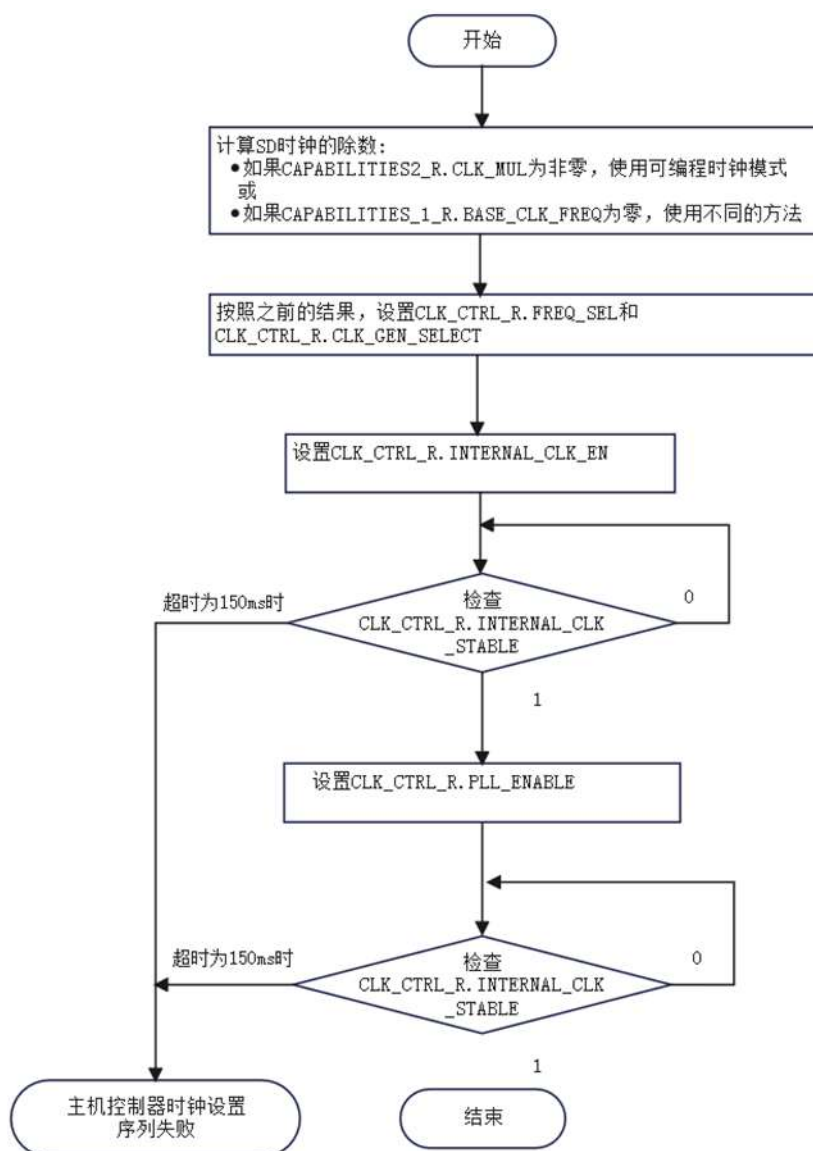
图表 3-10 eMMC 接口主机控制器设置序列

3.5.3 时钟控制

本节讨论设置提供给 DWC_mshc 的内部时钟和卡时钟、提供和设置提供给卡的时钟以及更改卡时钟频率的编程序列。

3.5.3.1 主机控制器时钟设置序列

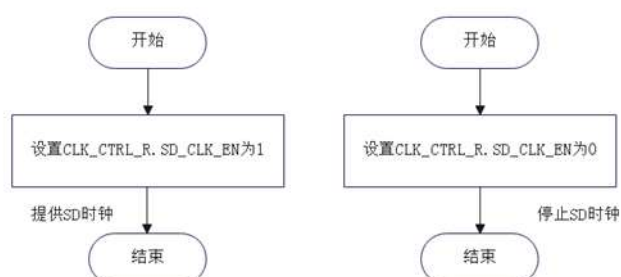
图表 3-11 显示了为 DWC_mshc 设置时钟的序列。DWC_mshc 需要来自系统的一些时钟信号。所有的输入时钟彼此是异步的。根据启用 DWC_mshc 所需时钟的不同控件，它们被分为两种类型，即内部时钟和卡时钟。



图表 3-11 主机控制器时钟设置序列

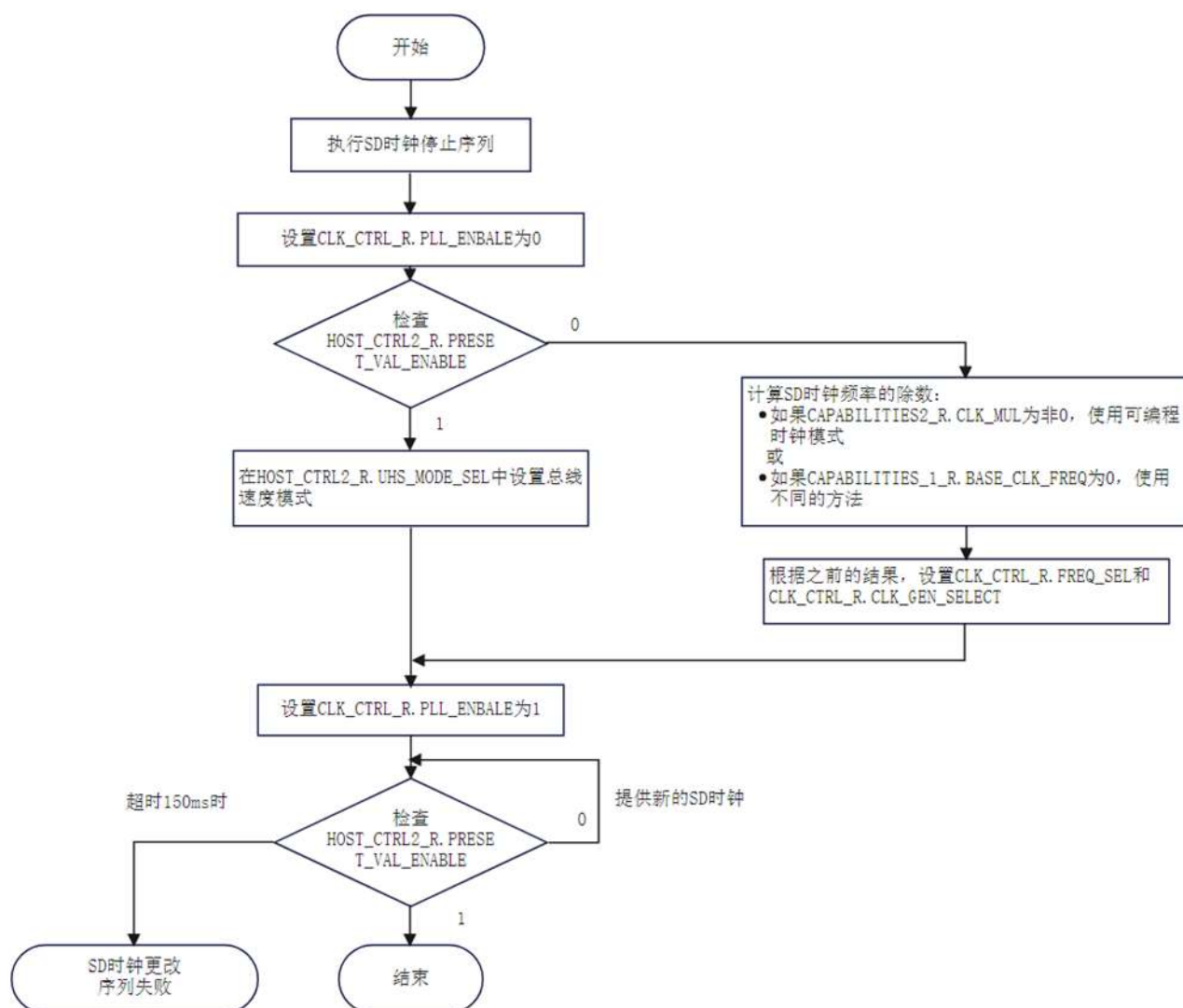
3.5.3.2 卡时钟提供和停止序列

图表 3-12 显示了将卡上时钟停止的流程图。



图表 3-12 卡时钟提供和停止序列

3.5.3.3 SD 时钟频率更改序列



图表 3-13 SD 时钟频率更改序列

3.5.4 卡接口设置序列

本节讨论设置 SD 和 eMMC 卡接口的编程序列。

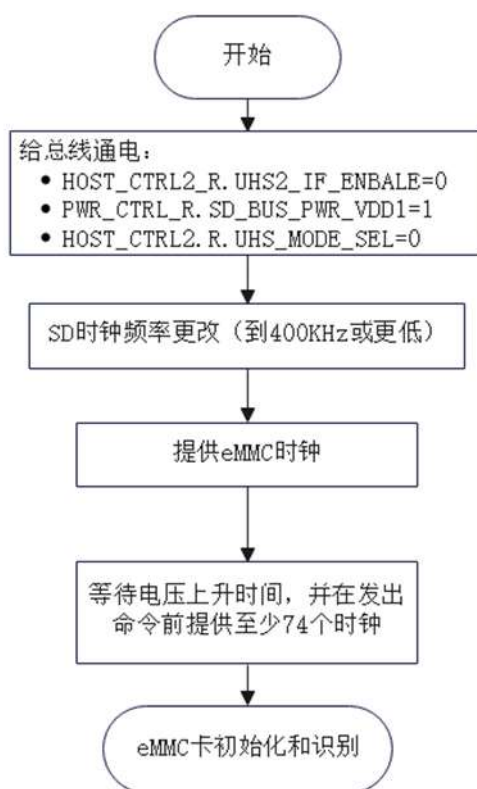
3.5.4.1 SD 卡接口检测



图表 3-14 SD 卡接口检测序列

3.5.4.2 eMMC 卡接口设置

图表 3-15 显示了设置 eMMC 设备的编程序列。



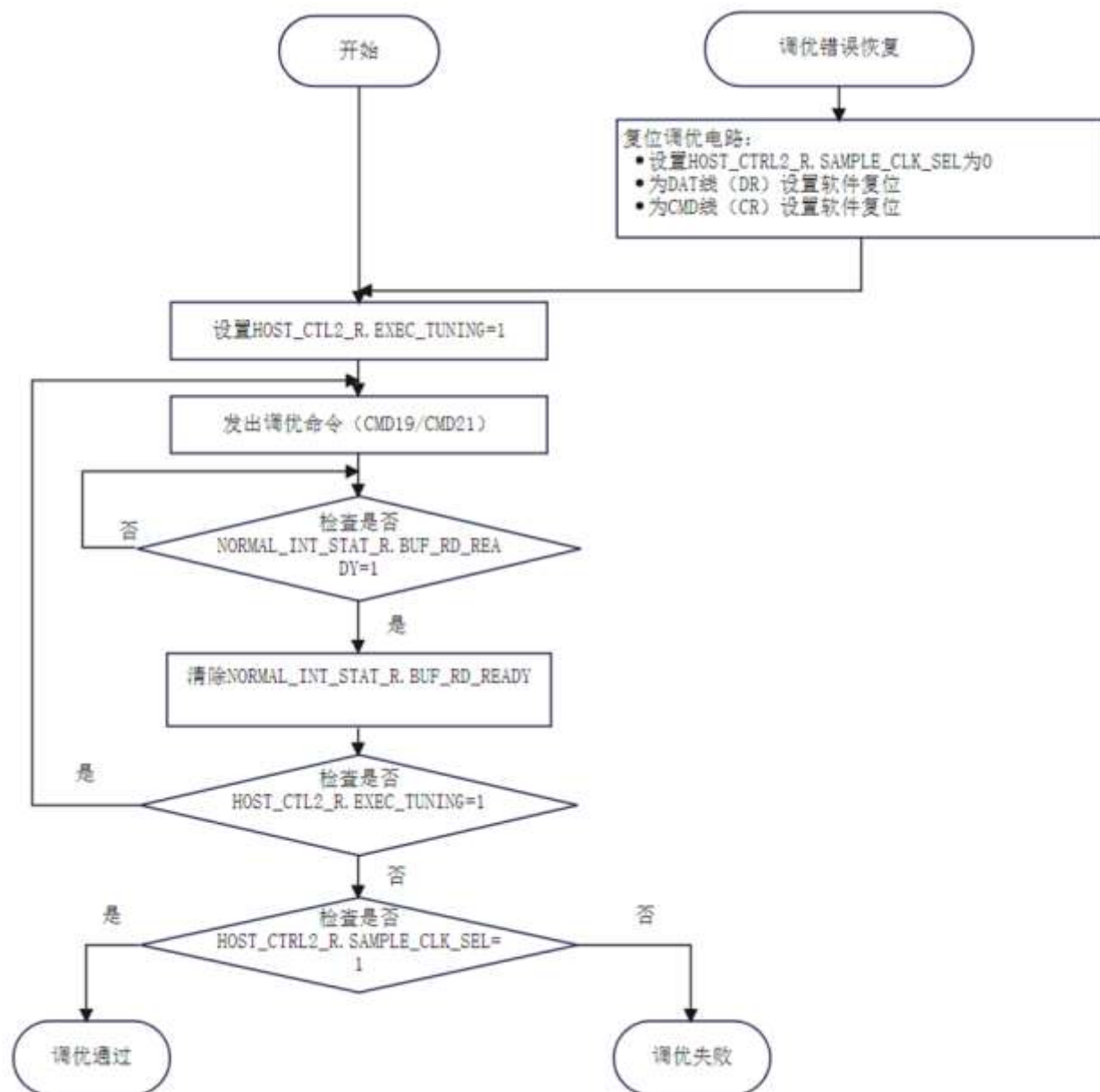
图表 3-15 eMMC 卡设置序列

3.5.5 调优序列

本节讨论 SD 和 eMMC 模式下软件调优的序列。

3.5.5.1 调优流程序列

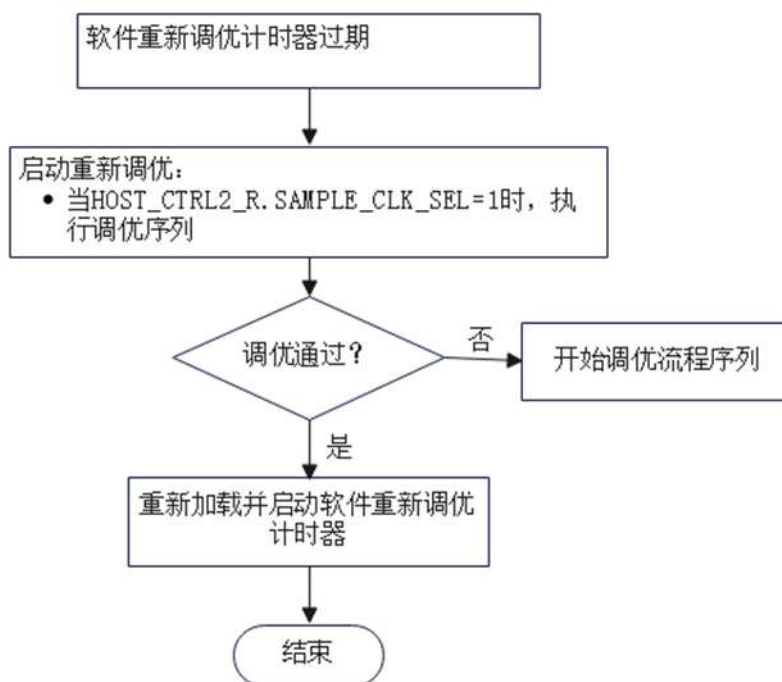
图表 3-16 显示了如何执行模式 1 重新调优, 即 HOST_CTRL2_R.SAMPLE_CLK_SEL 寄存器位设置为 1 时开始调优的过程。在开始新的调优之前使用调优复位, 使用专用的 HOST_CTRL2_R write 来复位调优, 然后再用一个 HOST_CTRL2_R write 来开始调优。CMD19/CMD21 用于启动调优过程。CMD19/CMD21 类似于其他读命令。因此, 在将 CMD19/CMD21 写入 CMD_R 之前, 必须正确设置 BLOCKSIZE_R、BLOCKCOUNT_R 和 XFERMODE_R 寄存器。



图表 3-16 调优序列

3.5.5.2 模式 1 重调流程

图表 3-17 显示了模式 1 重新调优的编程序列。

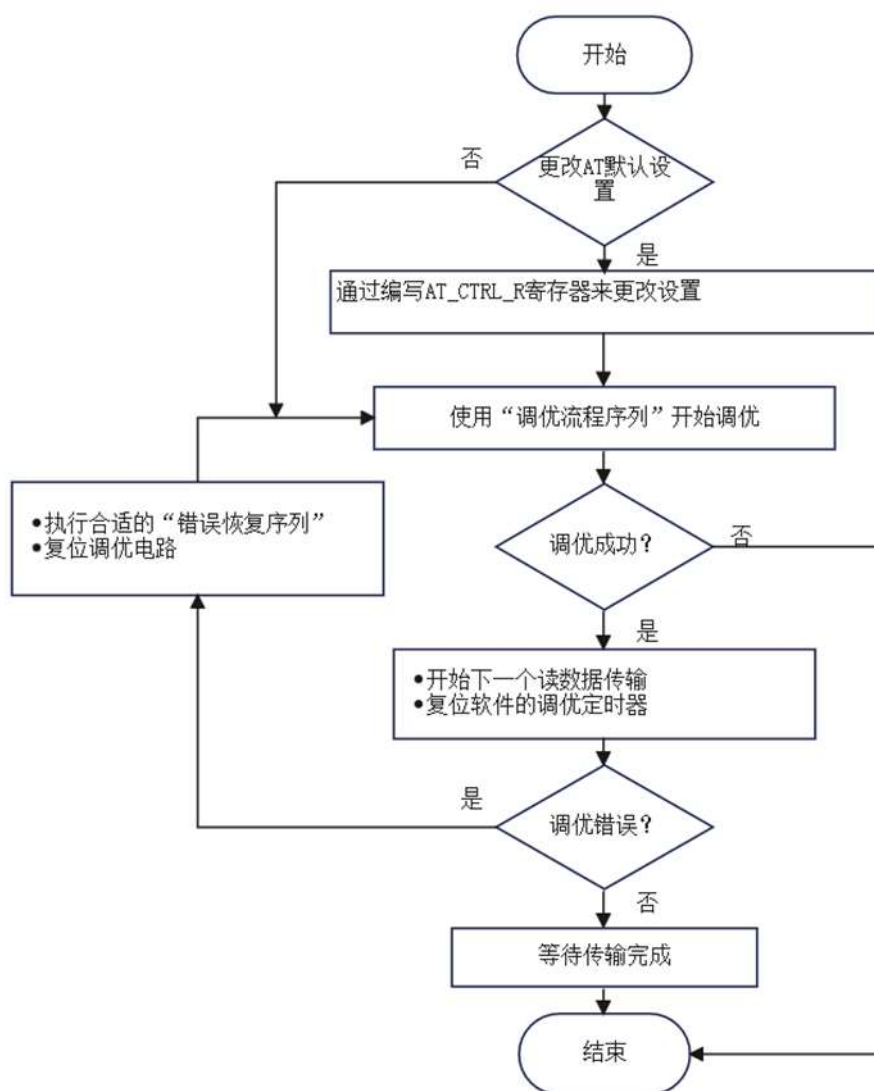


图表 3-17 模式 1 重调流程序列

3.5.5.3 模式 3 重调/自调流程序列

当速度模式切换到 SDR104 或 HS200 时, 自动调优特性在 DWC_mshc 中默认启用。如果需要更改任何参数或禁用该特性, 可以使用 AT_CTRL_R 寄存器启用或禁用该特性。只有当 sample_cclk_sel 设置为 0 时, 即在开始调优之前, 才能对 AT_CTRL_R 进行编程。

在启用自动调优 (模式 3) 时, 不能使用模式 1 重新调优。

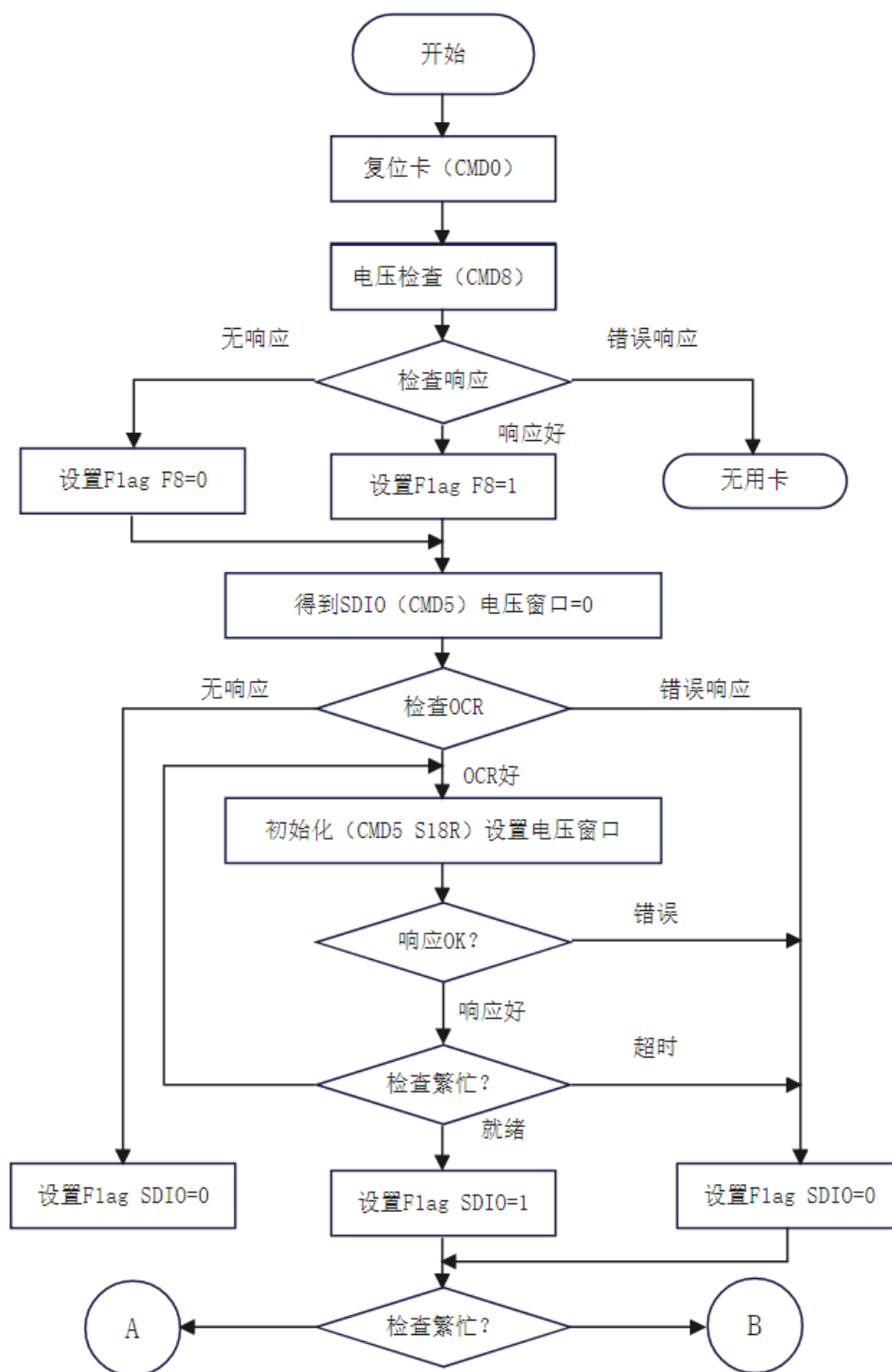


图表 3-18 自动调优序列

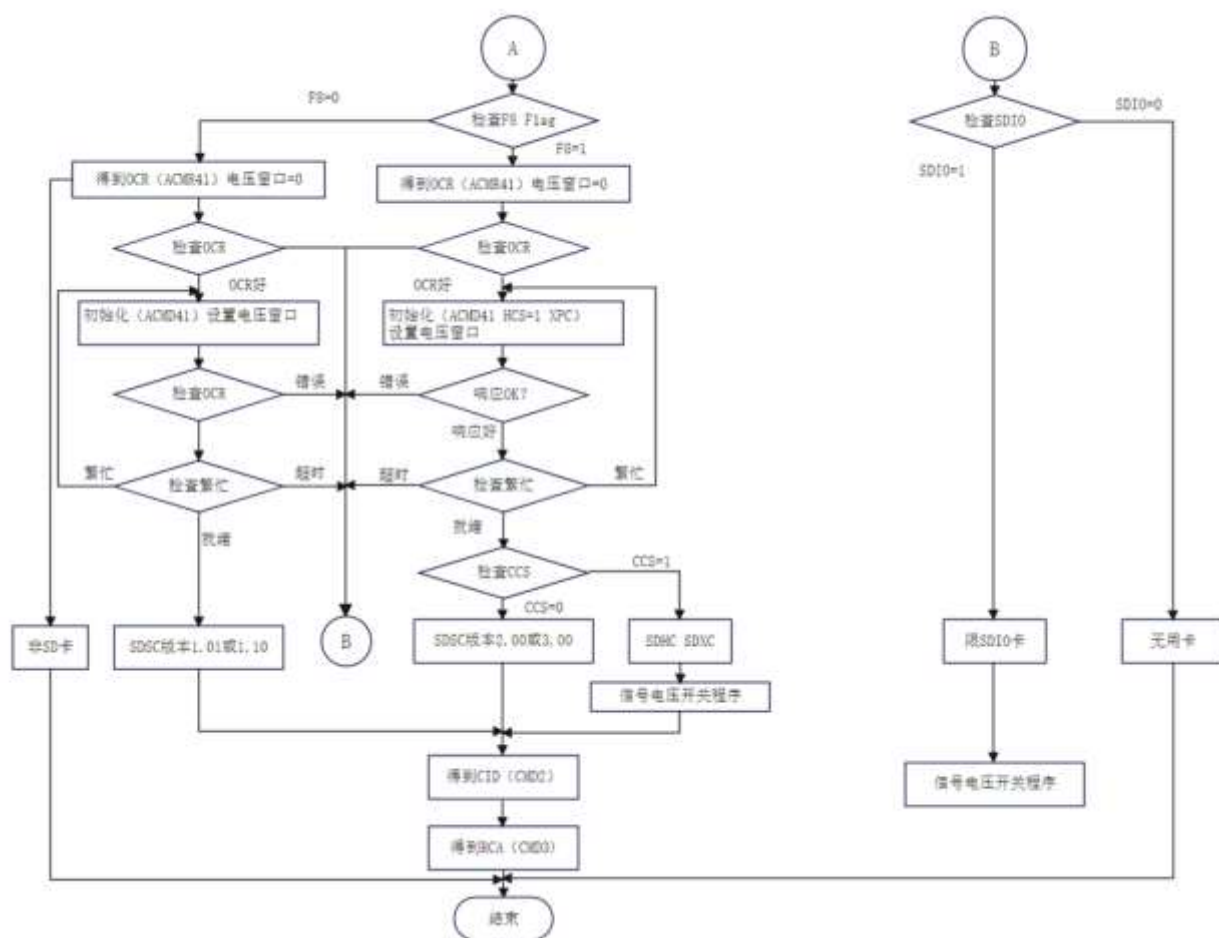
3.5.6 SD/SDIO 事务模式

本节包括生成和控制不同类型 SD/SDIO 事务的序列，这些事务可以分为不使用 DAT 线进行数据传输的事务和使用 DAT 线进行数据传输的事务。本节还讨论了调优流程、控制总线电源、改变总线宽度和速度模式、设置 DAT 线超时、初始化和识别 SD 卡以及终止序列的序列。

3.5.6.1 初始化和识别 SD 卡



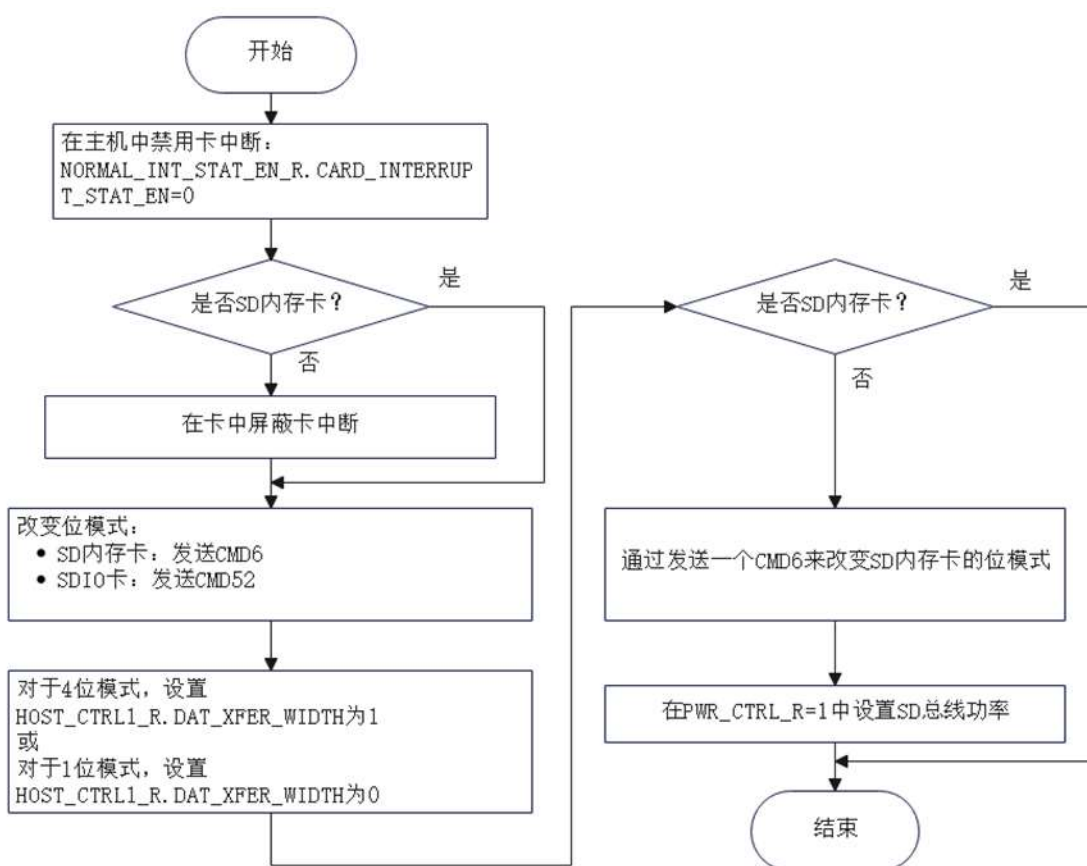
图表 3-19 初始化和识别 SD 卡



图表 3-20 初始化和识别 SD 卡（续）

3.5.6.2 改变 SD 总线宽度

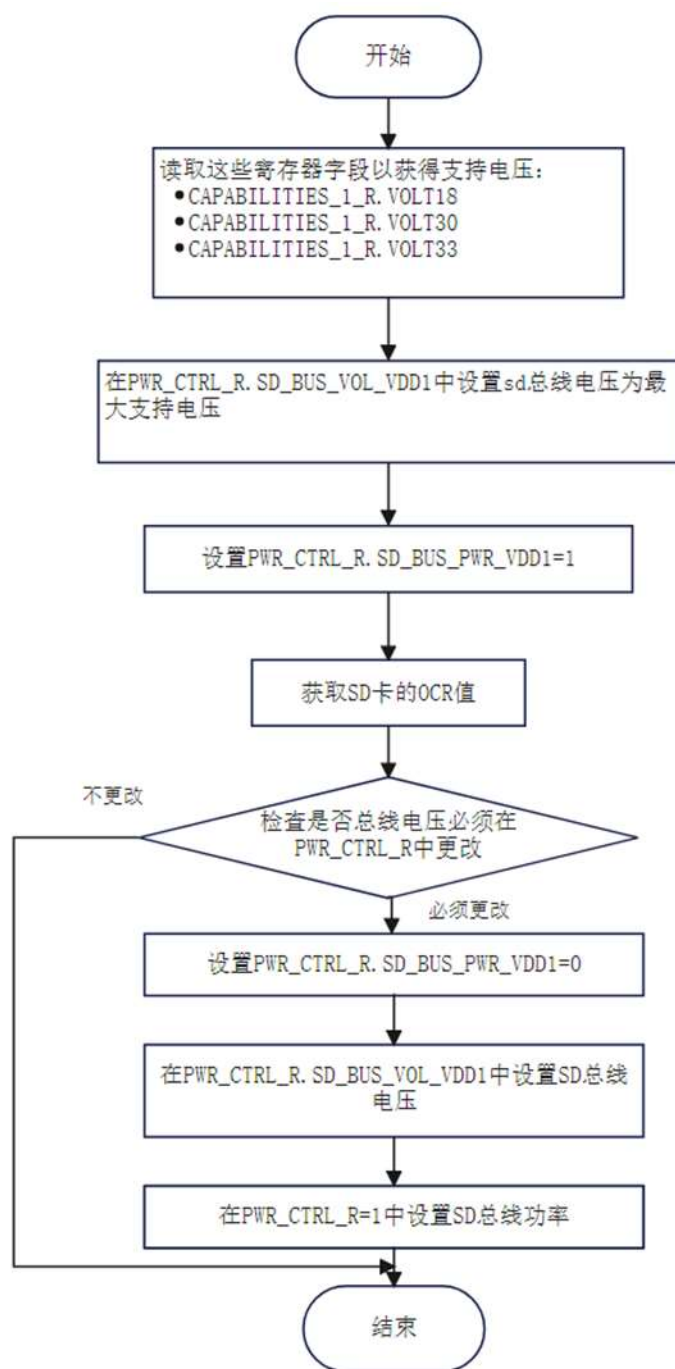
图表 3-21 显示了在 SD 总线上改变位模式的序列。



图表 3-21 改变总线宽度序列

3.5.6.3 SD 总线电源控制

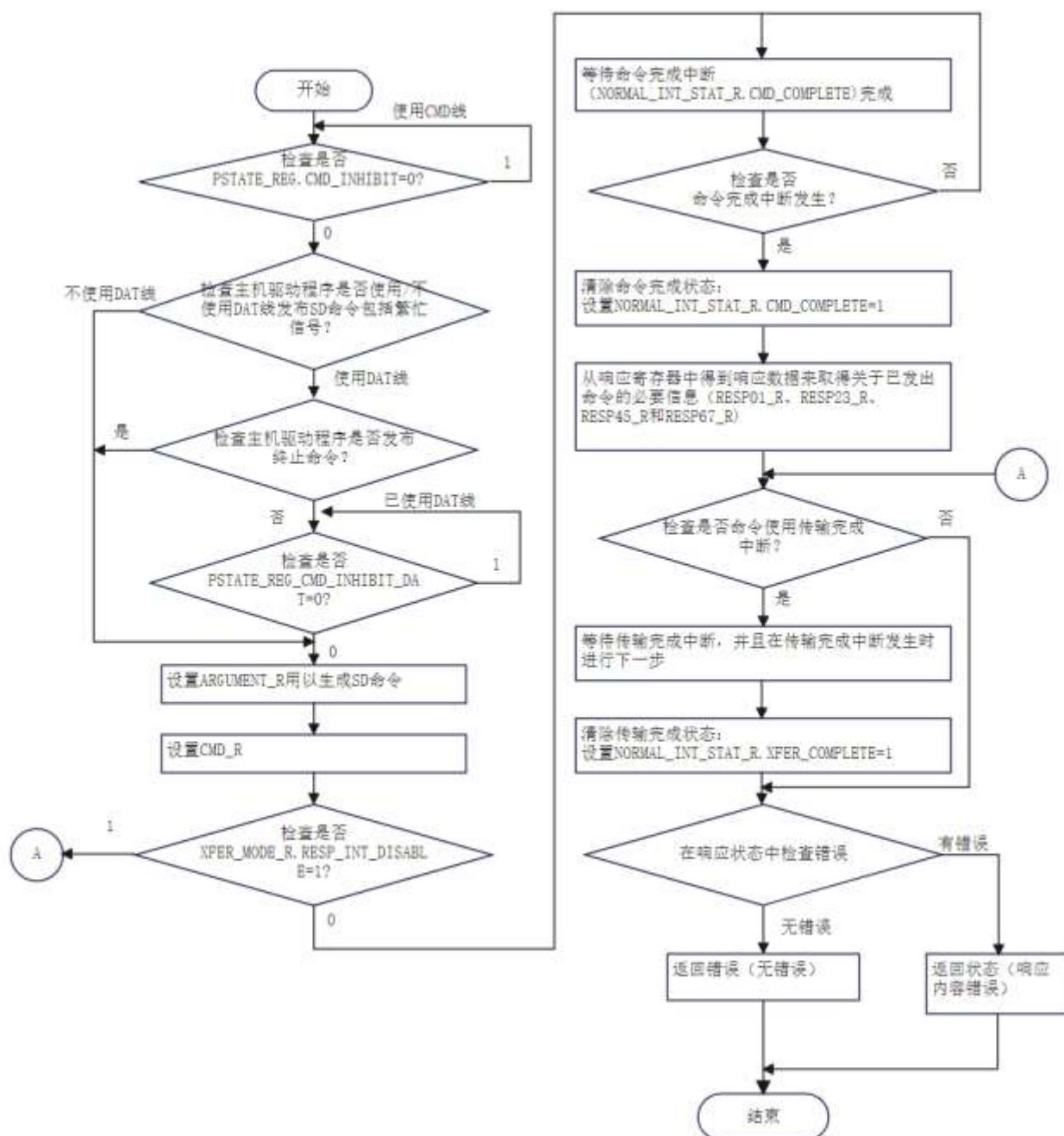
图表 3-22 显示了 SD 总线电源的控制序列。



图表 3-22 SD 总线电源控制序列

3.5.6.4 发出无数据传输的命令

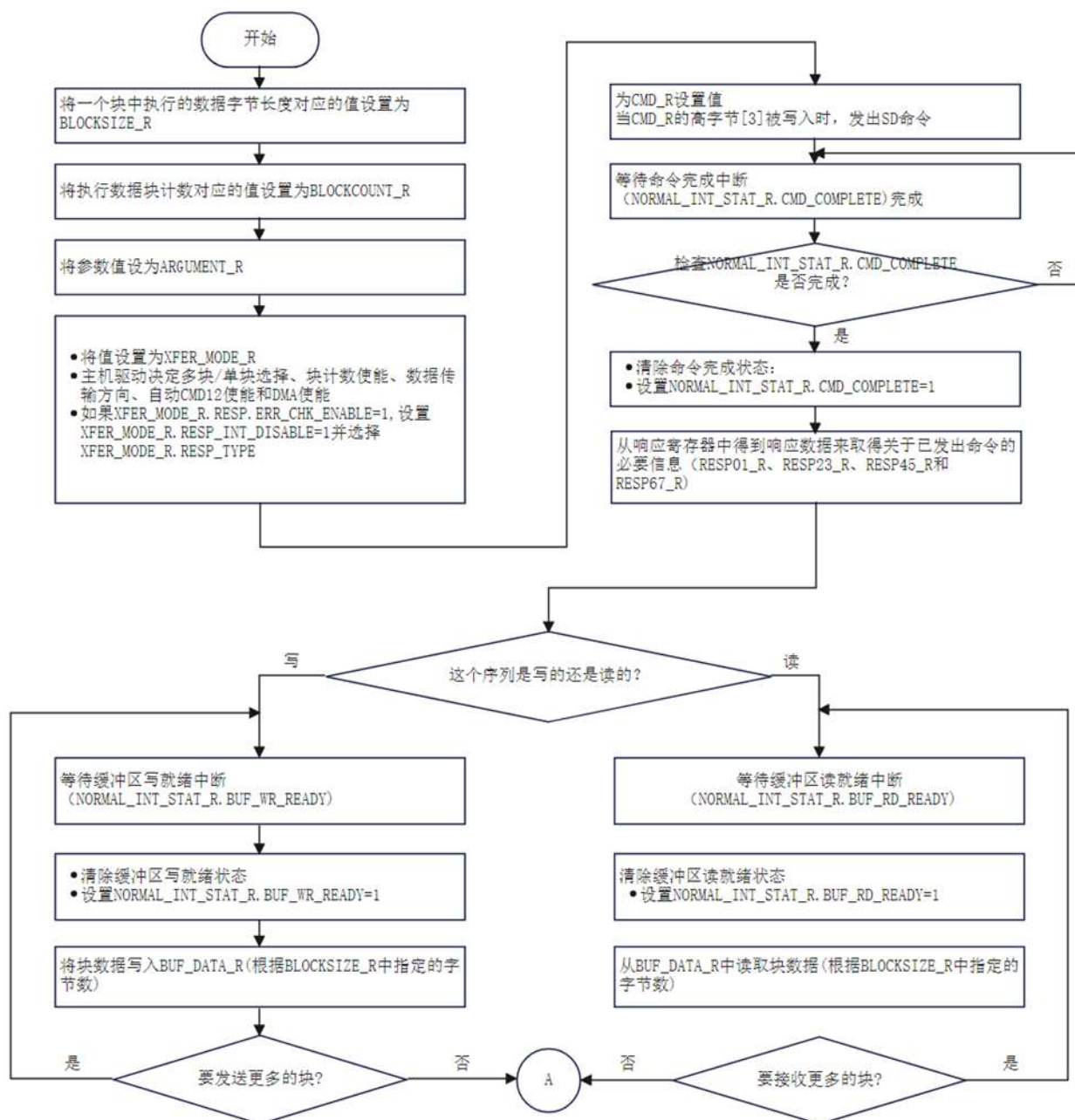
图表 3-23 显示了如何发出和完成 SD 命令的序列。SD 命令可能使用或不使用数据线（DAT）（用于繁忙信令）。图表 3-23 显示了针对这两种场景发出命令的过程。



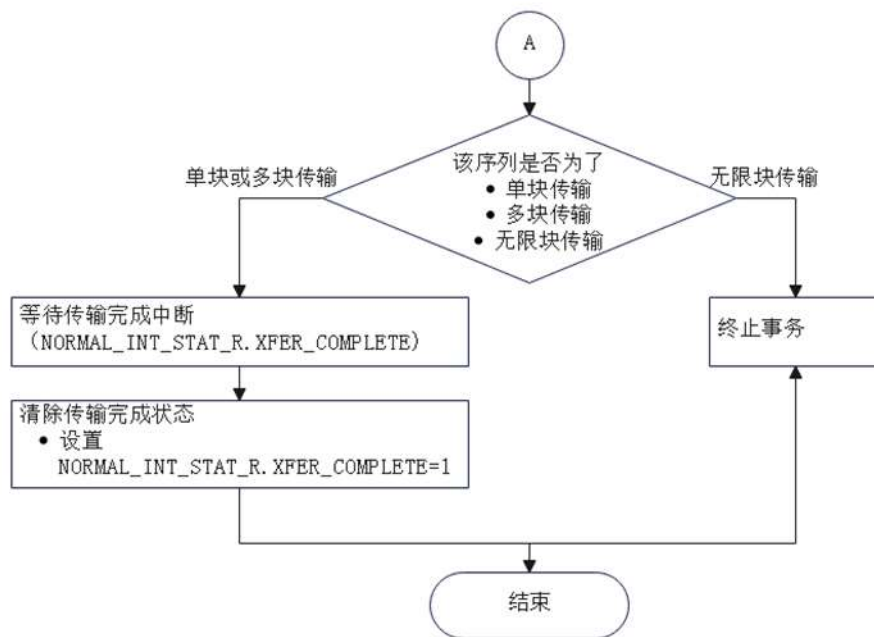
图表 3-23 SD 命令发出与完成

3.5.6.5 发出有数据传输的命令

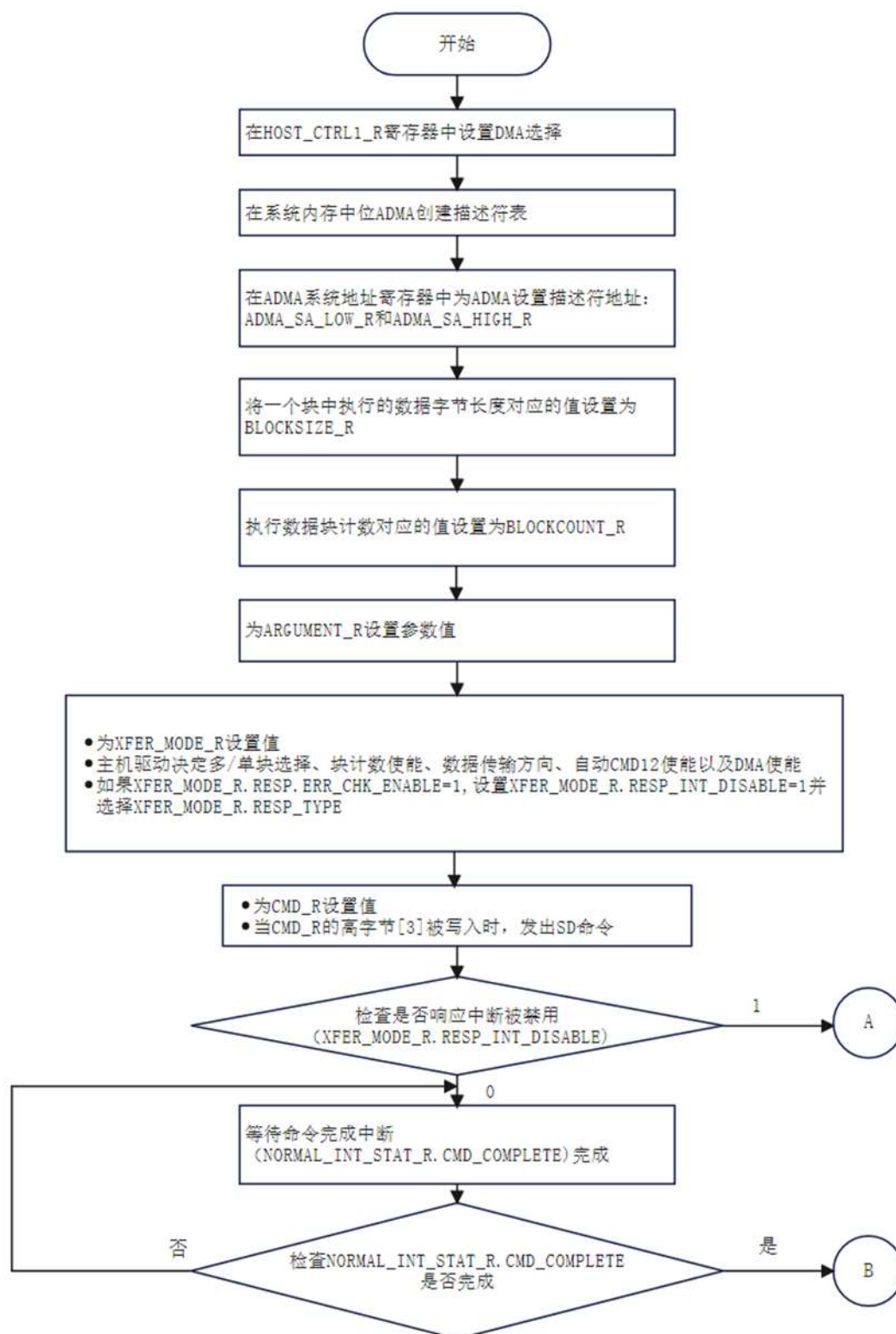
本节包括在不使用 DMA，而使用 SDMA、ADMA2 和 ADMA3 模式的数据传输过程中发出命令的序列。



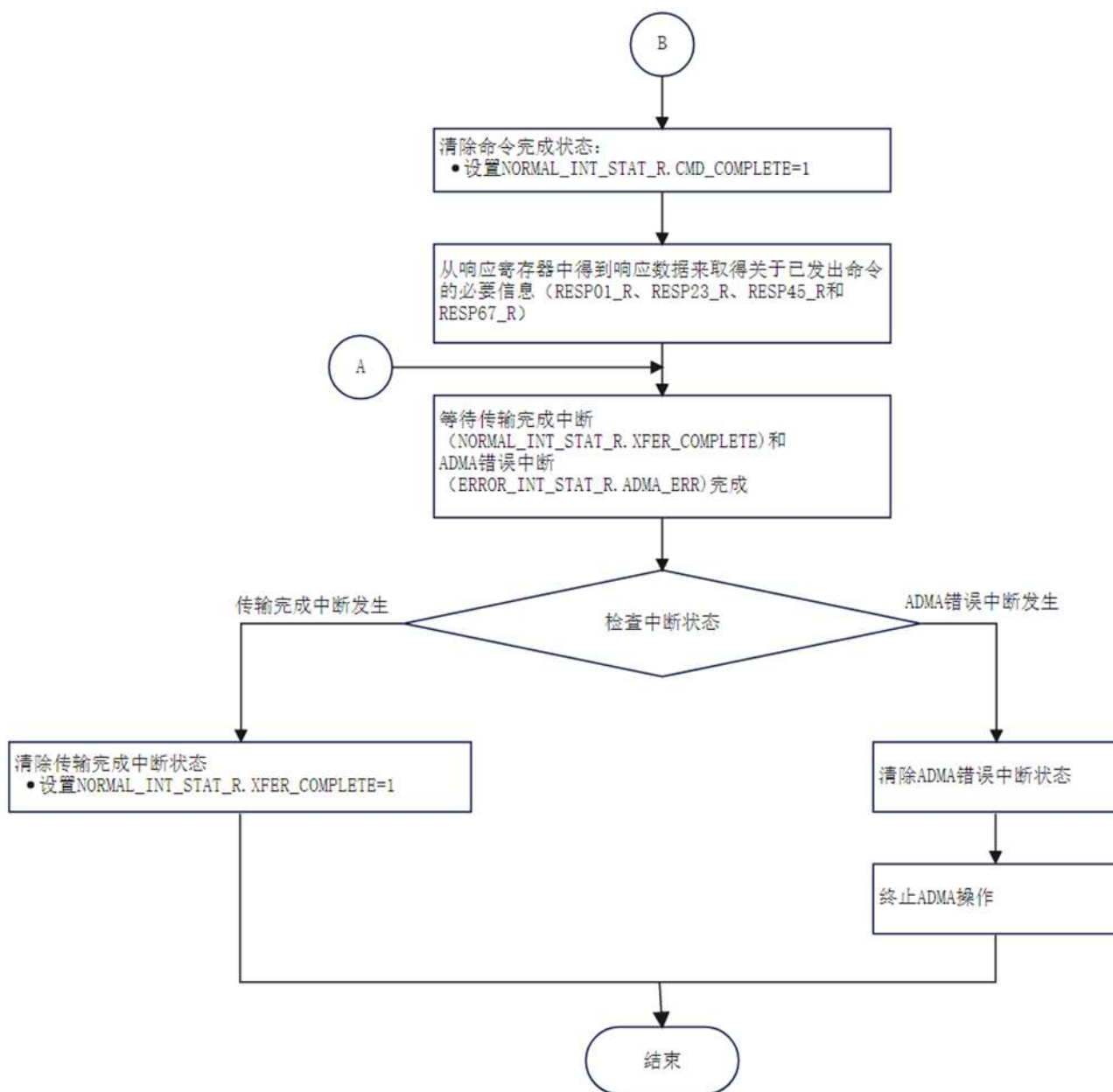
图表 3-24 有数据传输的事务控制（不使用 DMA）



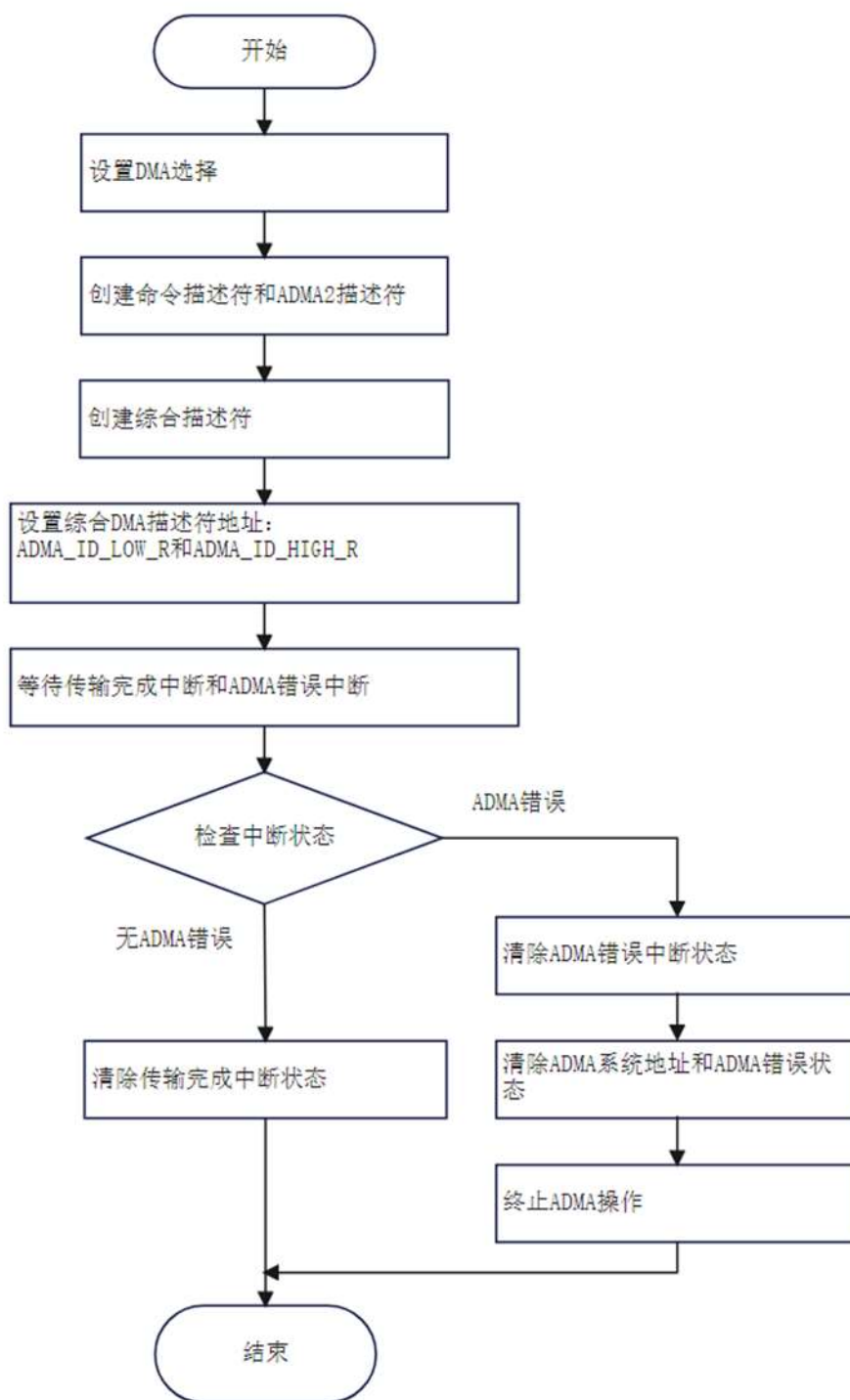
图表 3-25 有数据传输的事务控制（不使用 DMA，续）



图表 3-26 有数据传输的事务控制（使用 ADMA2）

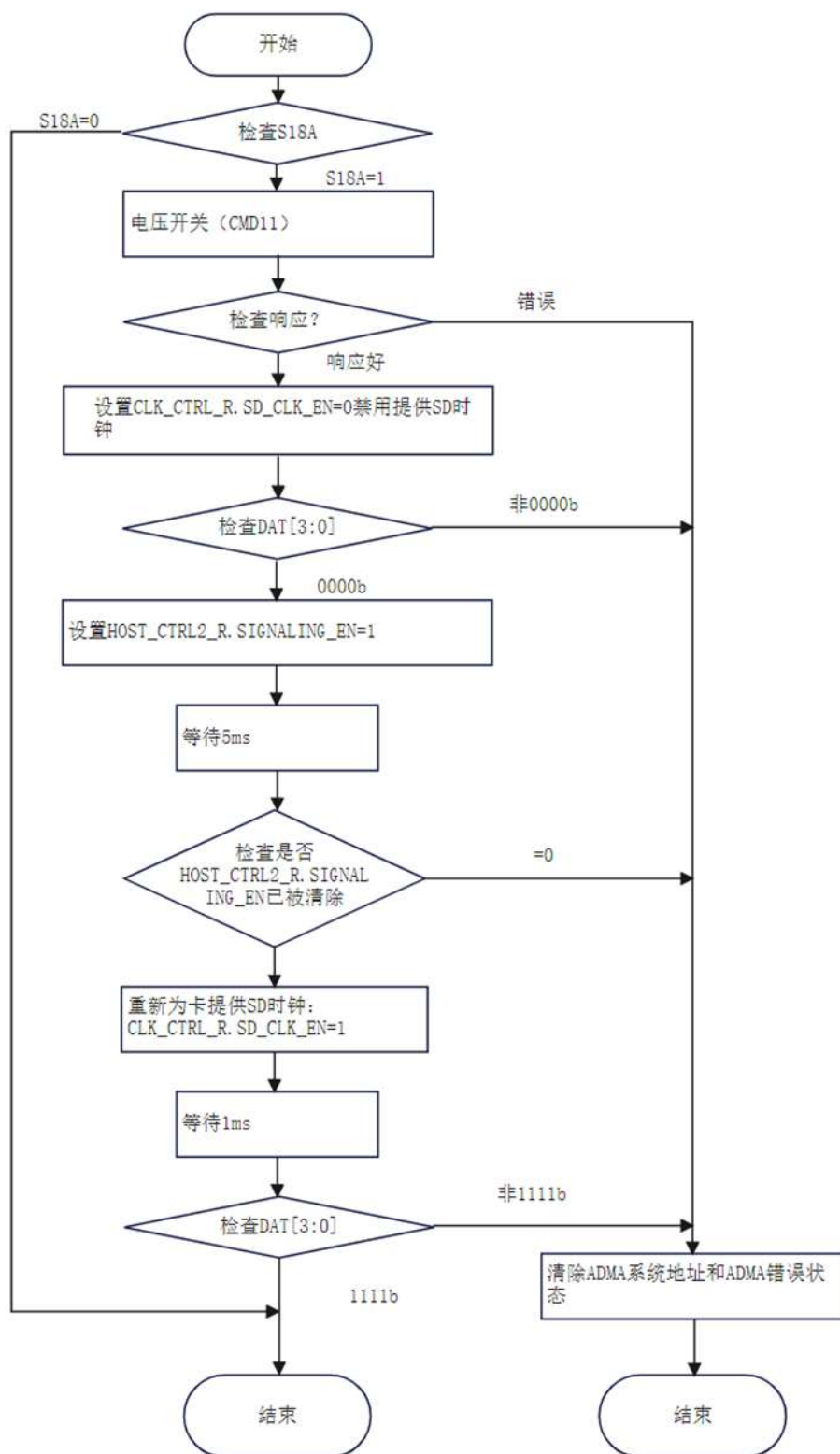


图表 3-27 有数据传输的事务控制（使用 ADMA2，续）



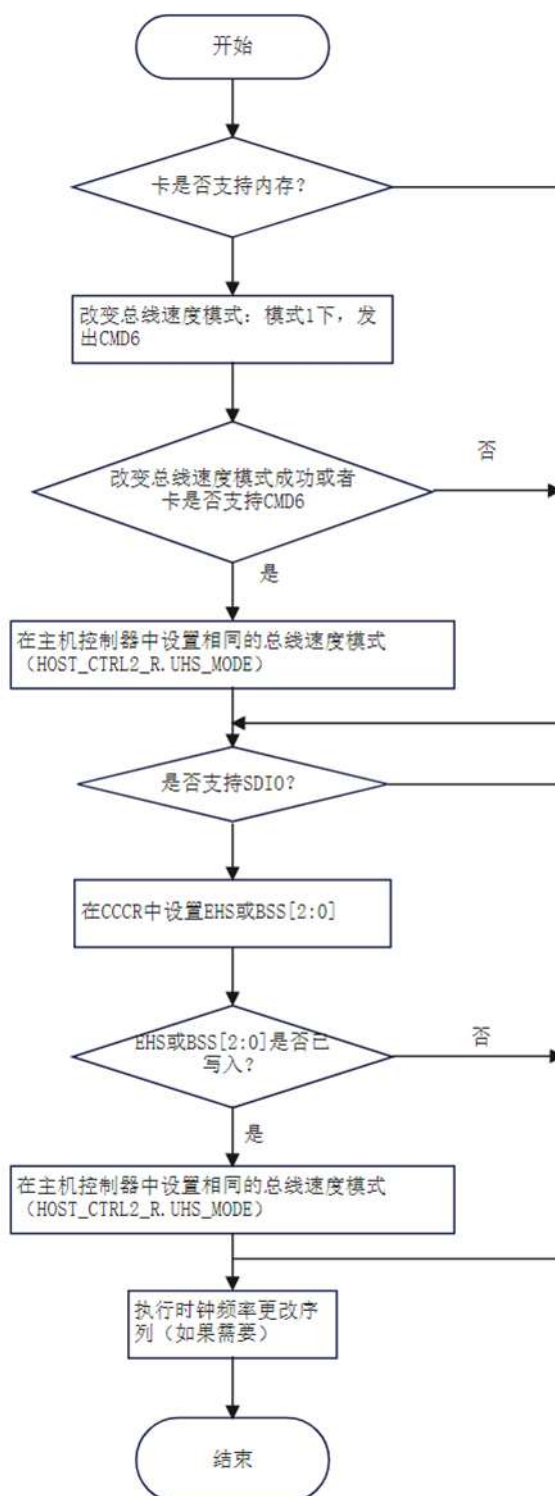
图表 3-28 有数据传输的事务控制（使用 ADMA3）

3.5.6.6 UHS-I 信号电压开关操作步骤



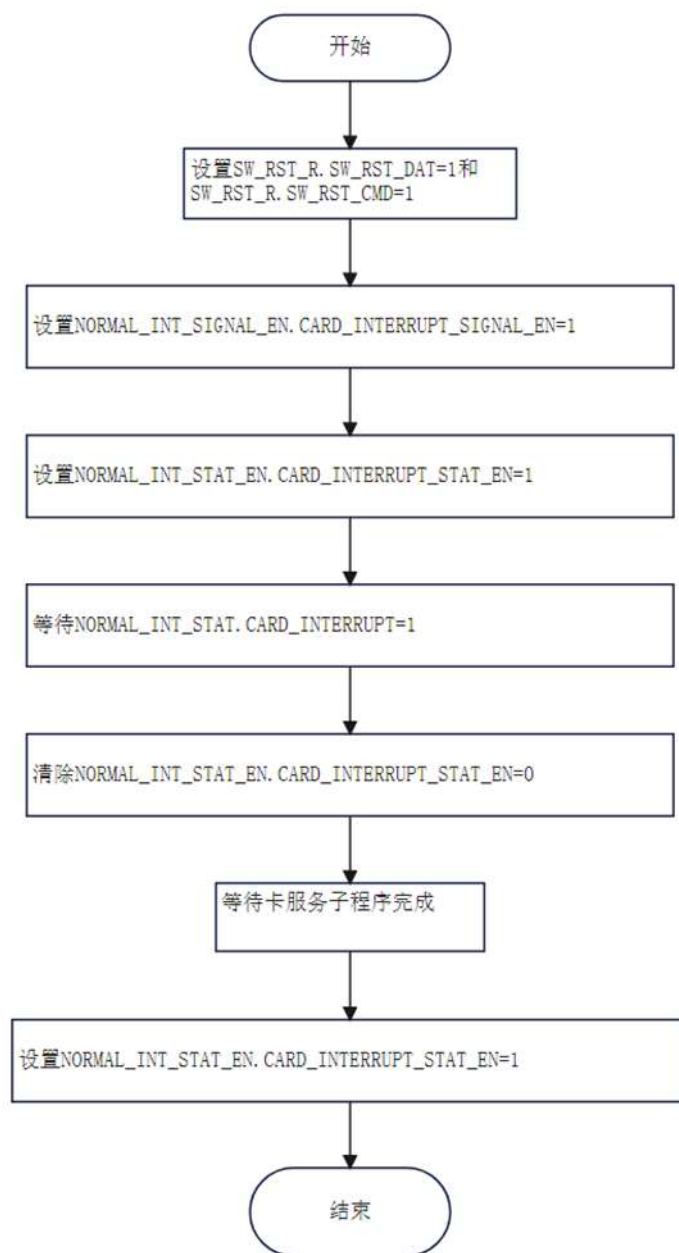
图表 3-29 信号电压开关操作步骤

3.5.6.7 改变 SD 总线速度模式



图表 3-30 改变 SD 总线速度模式

3.5.6.8 SDIO 卡中断

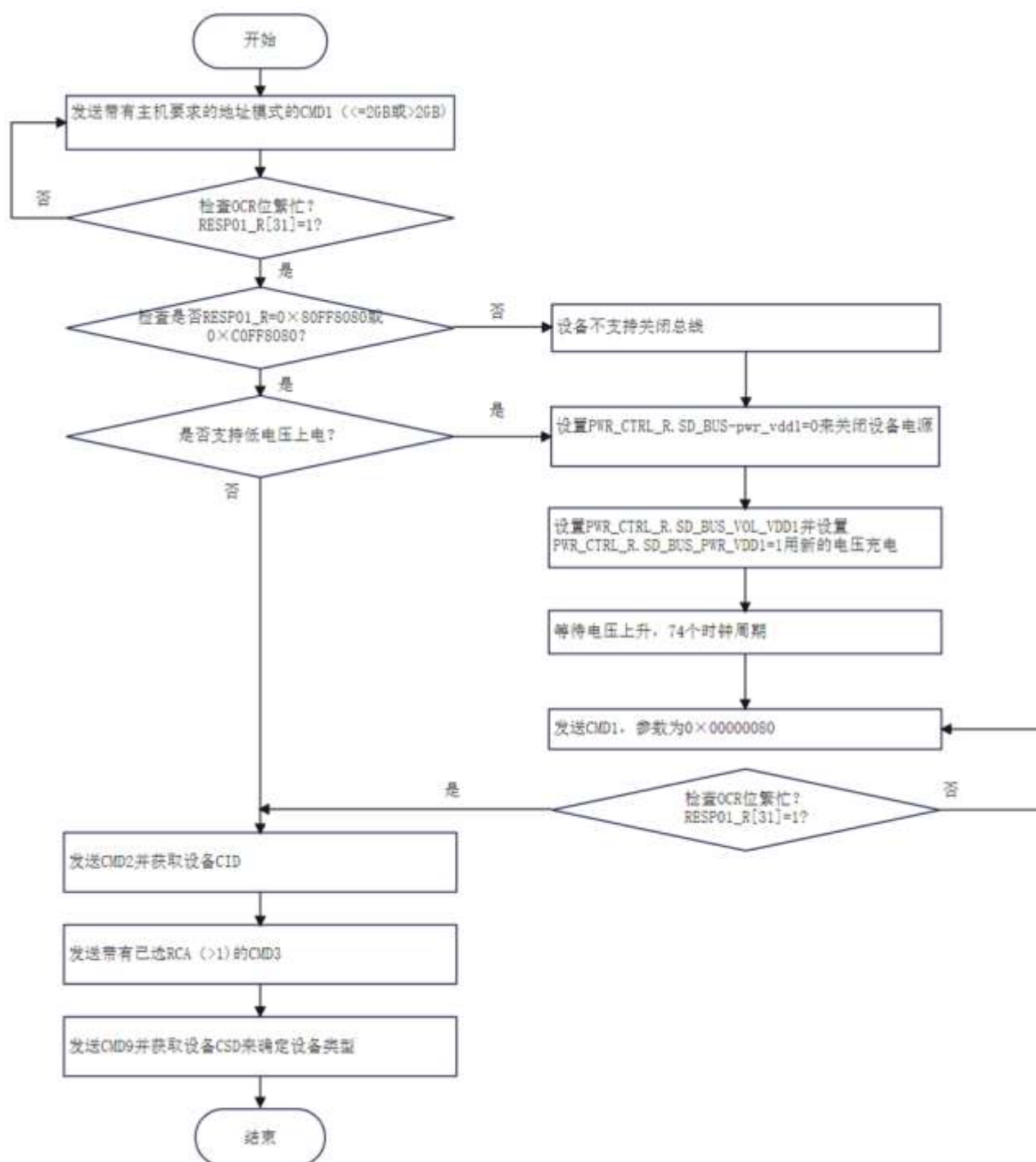


图表 3-31 SDIO 卡中断序列

3.5.7 eMMC 事务模式

本节讨论设置 eMMC 接口主机控制器的编程序列、上电时序、初始化和识别卡、发出使用/不使用 DAT 线进行/不进行数据传输的命令、切换到各种速度模式和改变数据总线宽度。本节还将讨论如何对不同的启动速度模式进行编程、启动和终止。

3.5.7.1 初始化和识别 eMMC 设备



图表 3-32 卡初始化和识别编程序列

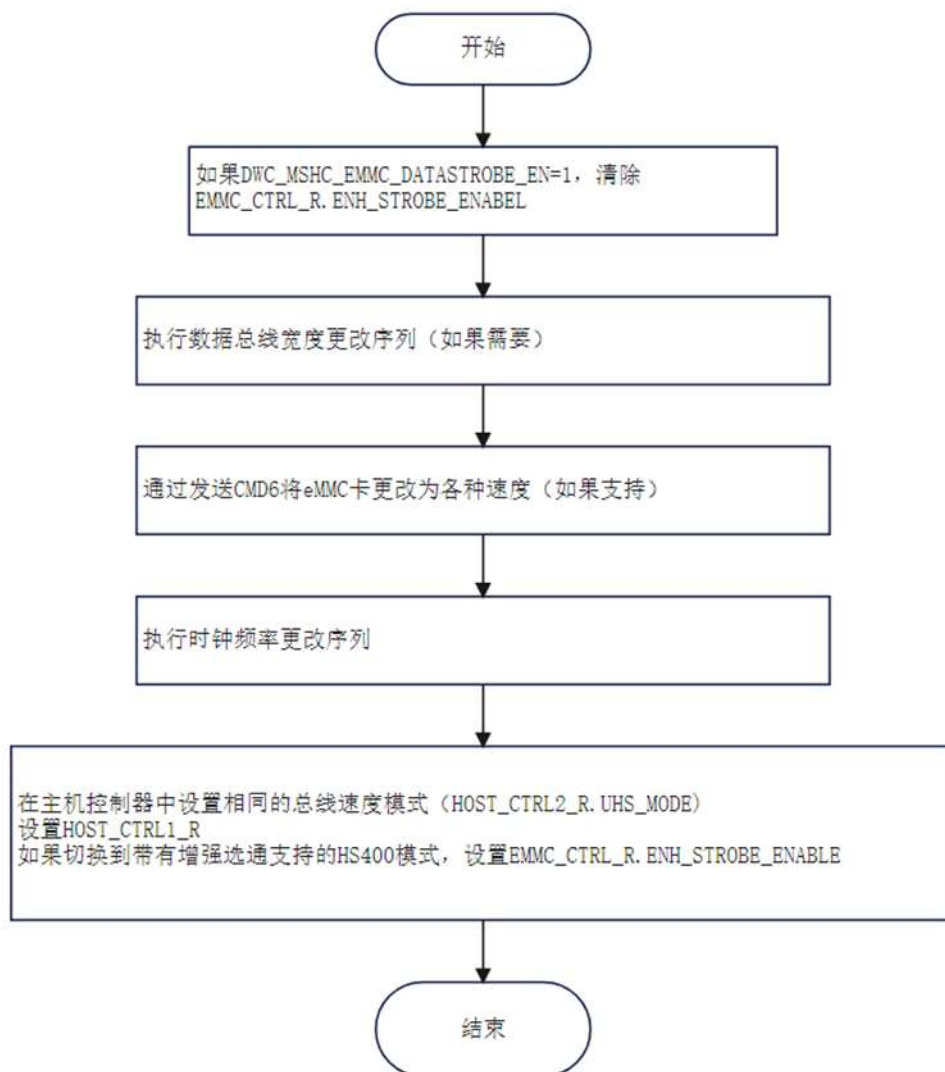
3.5.7.2 对 eMMC 设备发出无数据传输的命令

对 eMMC 设备发出无数据传输的命令的编程序列类似于在“发出无数据传输的命令”一节中提到的序列。

3.5.7.3 对 eMMC 设备发出有数据传输的命令

对 eMMC 设备发出有数据传输的命令的编程序列类似于在“发出有数据传输的命令”一节中提到的序列。数据传输可以使用或不使用 DMA。支持的 DMA 模式有 SDMA、ADMA2、ADMA3。

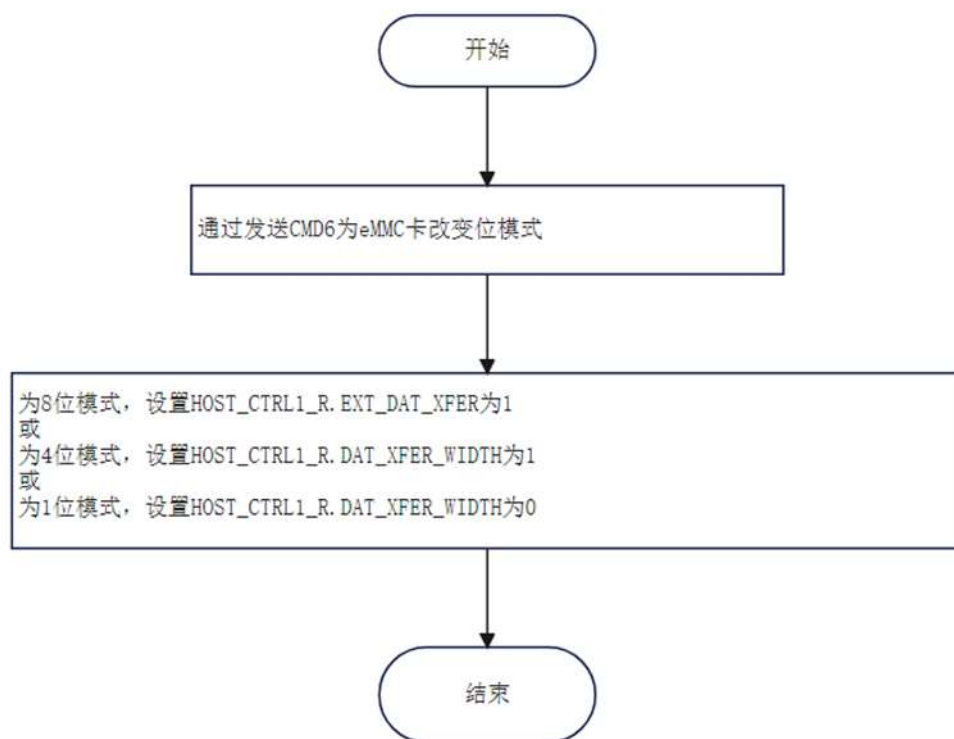
3.5.7.4 切换 eMMC 设备的各种速度模式



图表 3-33 在 eMMC 设备中切换各种速度模式的编程序列

3.5.7.5 修改 eMMC 设备的数据总线宽度

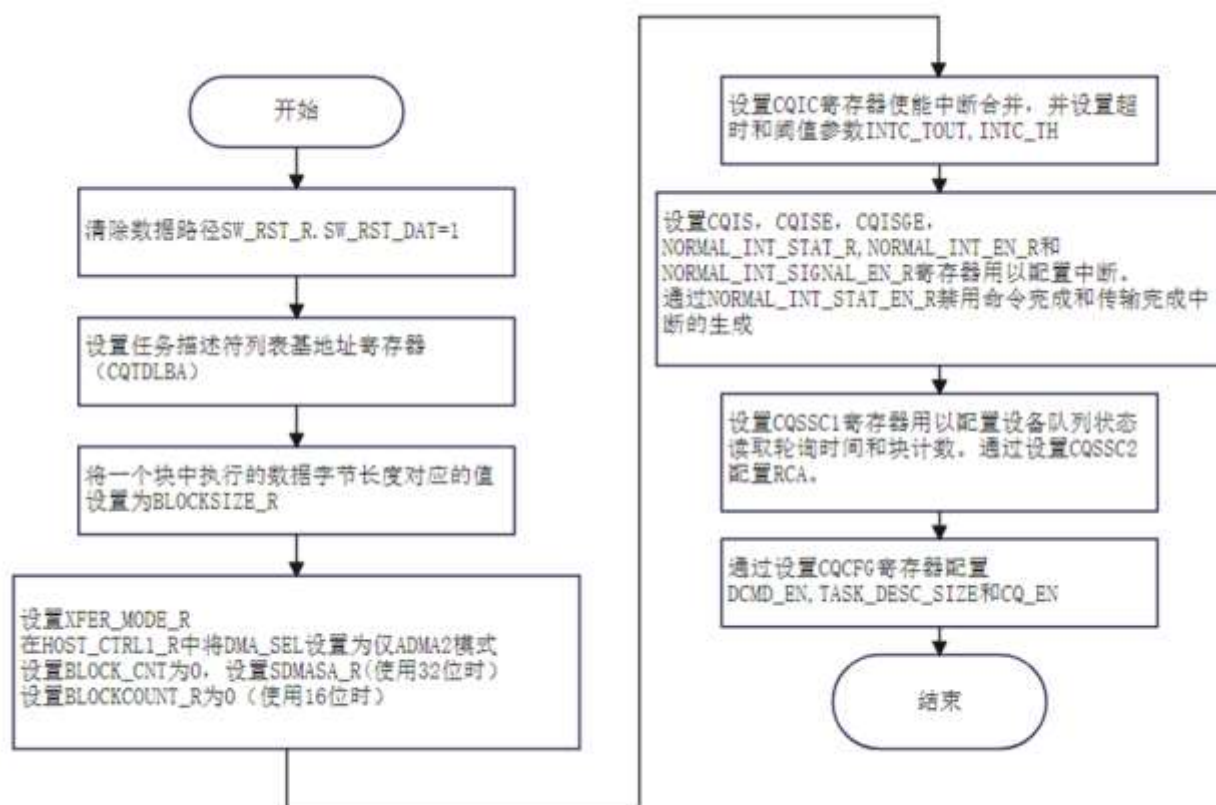
图表 3-34 显示了修改 eMMC 设备的数据总线宽度的编程序列。在图表 3-34 中，注意 CMD6 仅在传输状态下有效。



图表 3-34 修改 eMMC 设备数据总线宽度的编程序列

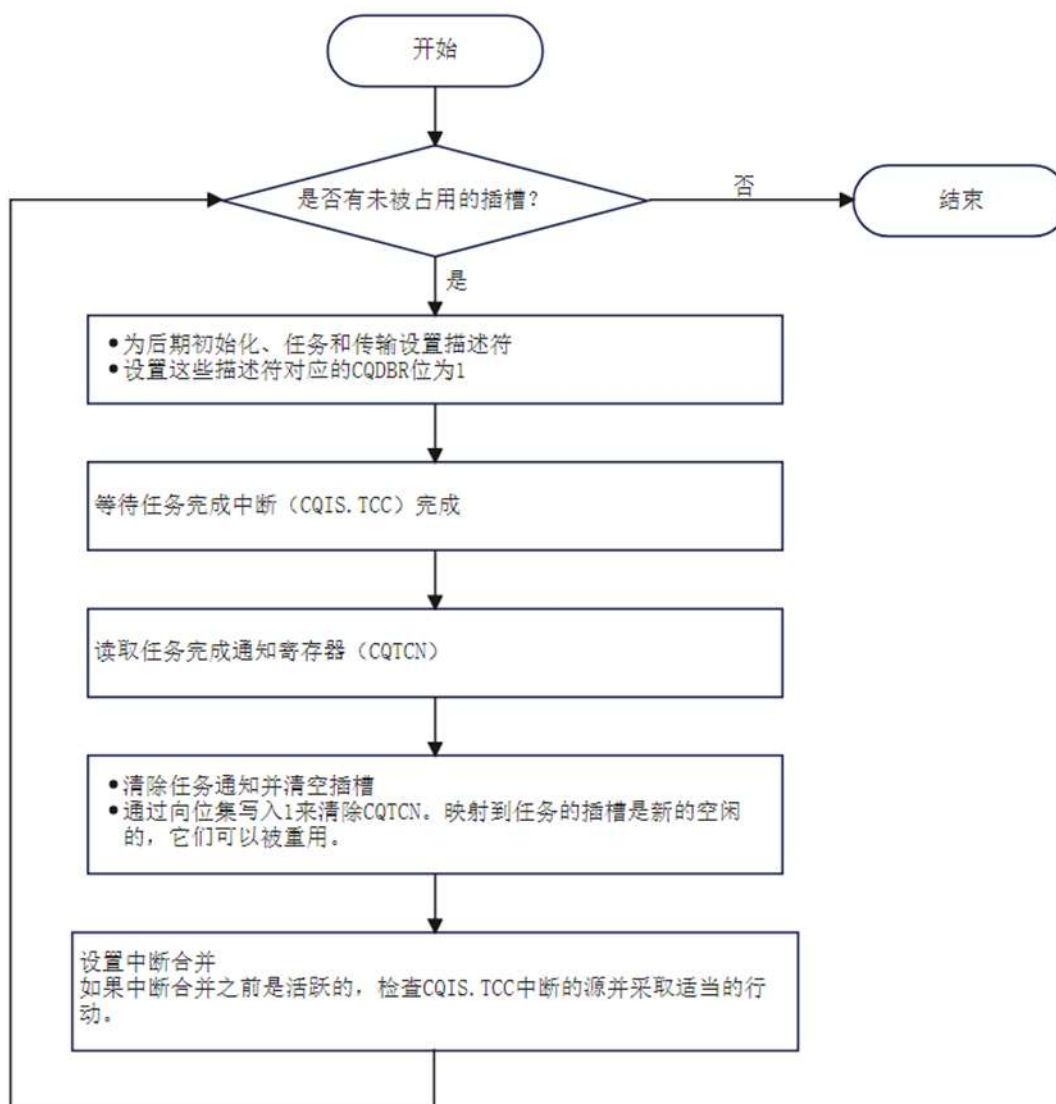
3.5.7.6 命令队列

图表 3-35 显示了初始化命令队列引擎的编程序列。



图表 3-35 初始化命令队列引擎

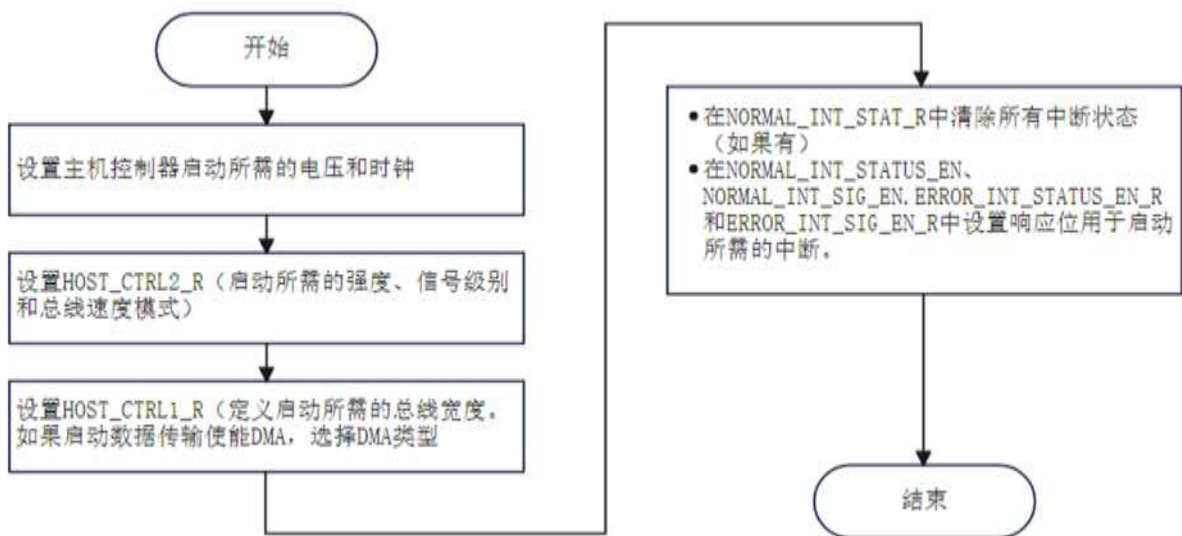
图表 3-36 显示了在命令队列引擎中提交和完成任务的序列。



图表 3-36 提交和完成任务

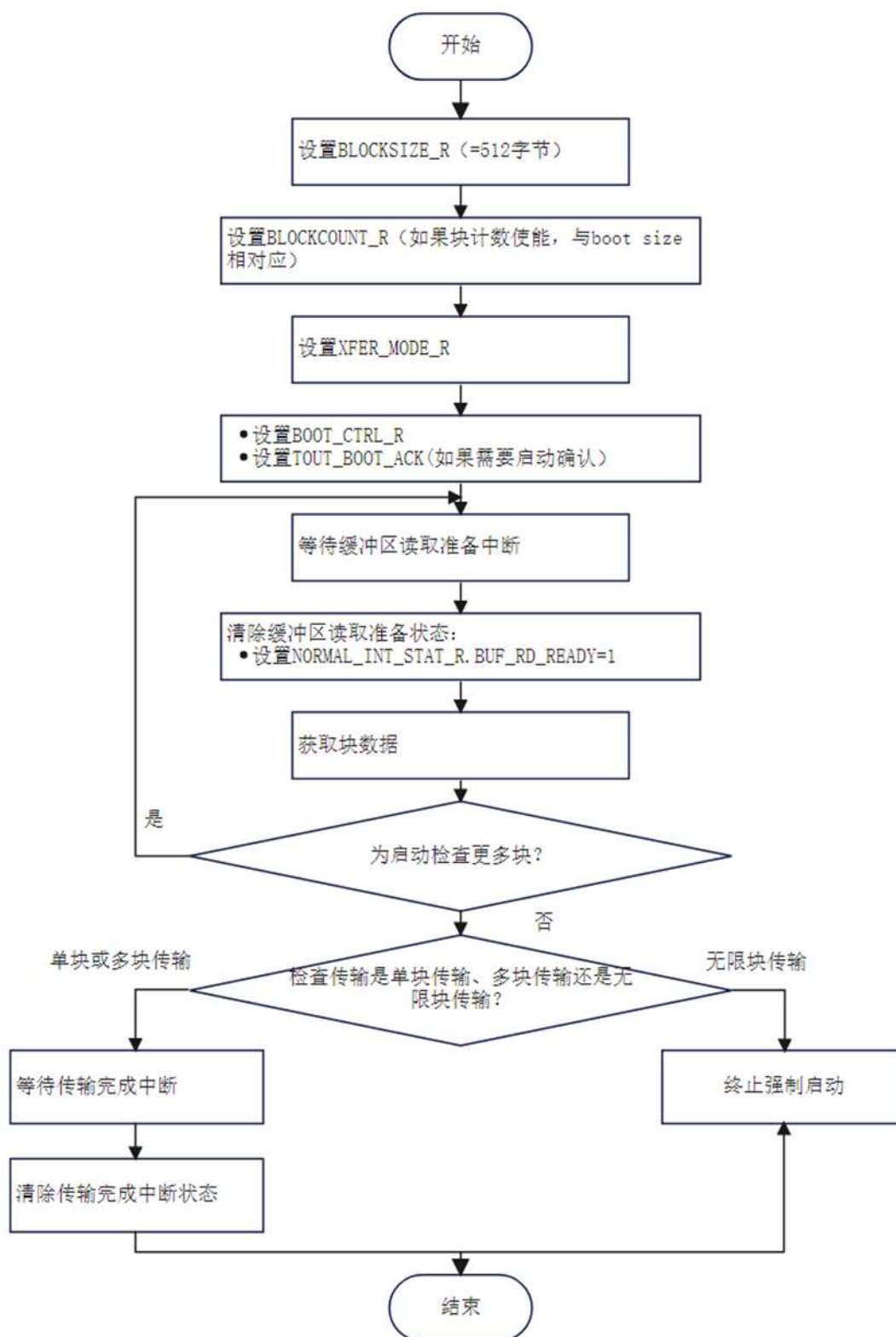
3.5.7.7 启动编程序列

图表 3-37 显示了准备启动的编程序列。这种序列在强制模式和备用模式中都很常见。



图表 3-37 准备启动的编程序列

图表 3-38 显示了在非 DMA 模式下发起强制启动的编程流程。



图表 3-38 非 DMA 模式下强制启动的编程序列

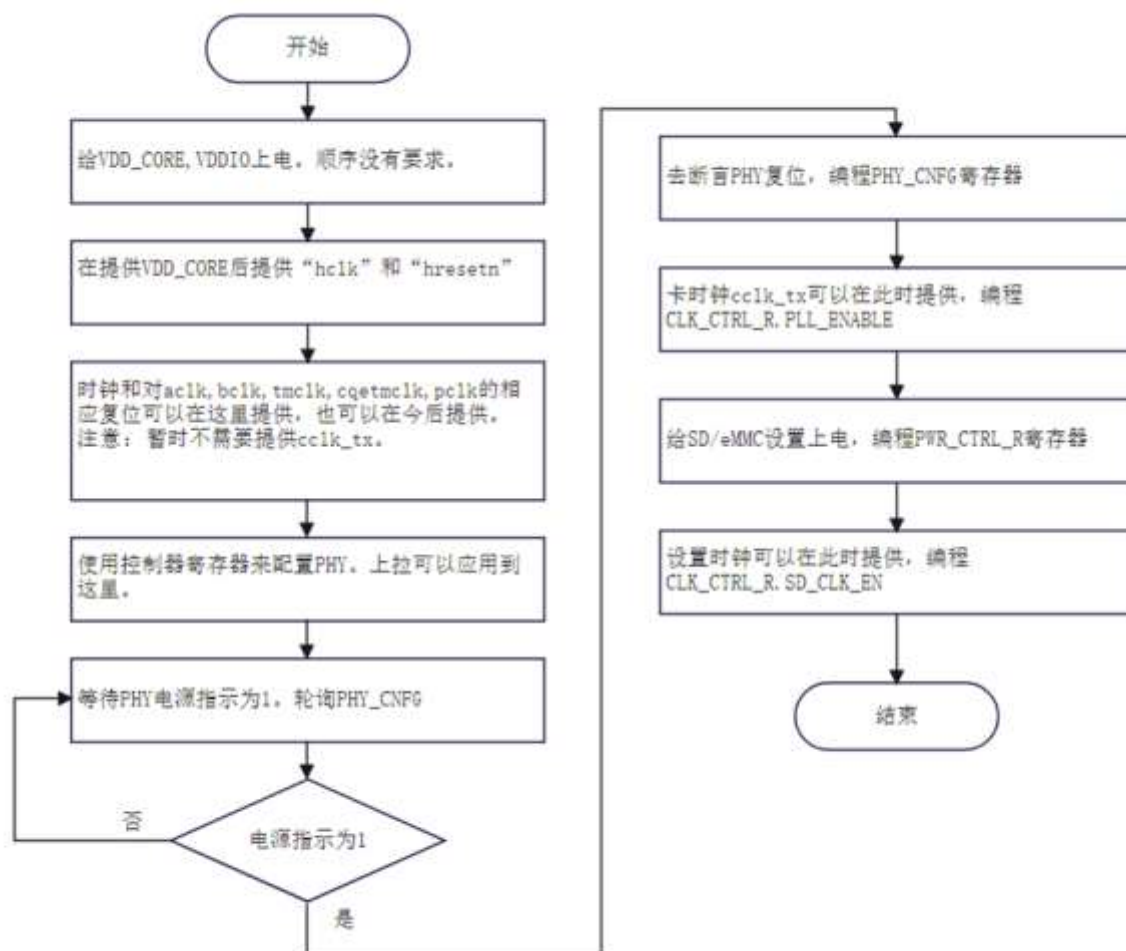
备用启动是基于命令的启动操作，可以通过 0xFFFFFFA 参数发出 CMD0 来启动。关于开启备用启动的编程序列，请参考“对 eMMC 设备发出有数据传输的命令”，但有以下例外：

- 如果预期引导确认，在设置 CMD_R 寄存器之前应编程 BOOT_CTRL_R 用以设置引导确认超时计数器。
- 由于 CMD0 无响应，无法执行响应检查。

在启动时，设备状态被假定为预启动状态。在发出 CMD0 之前，电源稳定需要 74 个时钟周期。

3.5.8 SD/eMMC PHY 配置

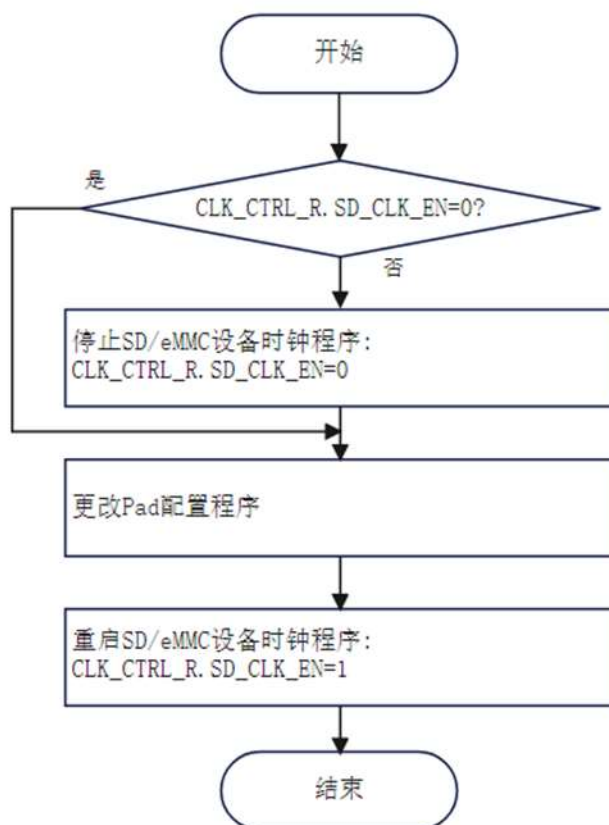
3.5.8.1 PHY 首次上电和复位



图表 3-39 PHY 上电和复位编程序列

3.5.8.2 PAD 配置序列

切换速度模式时，请按照推荐的 PAD 配置序列进行配置。



图表 3-40 PAD 配置序列

图表 3-41 SD PHY 在 1.8V 模式下的 PAD 推荐设置

3.3V/1.8V PHY 1.8V Mode	TXSLEW_CTRL_N*	TXSLEW_CTRL_P*	WEAKPULL_EN	RXSEL	PAD_SN*	PAD_SP*
LEGACY	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS SDR	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS DDR	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS 200	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9

3.3V/1.8V PHY 1.8V Mode	TXSLEW_CTRL_N*	TXSLEW_CTRL_P*	WEAKPULL_EN	RXSEL	PAD_SN*	PAD_SP*
HS 400	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0 STB: 0x2	0x1	0x8	0x9
HS400 ENHANCED	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0 STB: 0x2	0x1	0x8	0x9

3.6 寄存器

图表 3-42 可能的读写行为

读（或写）行为	说明
RC	读操作清除该寄存器字段。
RS	读操作设置该寄存器字段。
RM	读操作修改该寄存器字段的内容。
Wo	只能写入该寄存器一次。
W1C	写 1 会清除该寄存器字段。
W1S	写 1 会设置该寄存器字段。
W1T	写入值为 1 将切换该寄存器字段。
W0C	写入值为 0 时清除该寄存器字段。
W0S	写 0 设置该寄存器字段。
W0T	写入值为 0 将切换该寄存器字段。
WC	任何写操作都会清除该寄存器字段
WS	任何写操作都会设置该寄存器字段。
WM	任何写入都将切换该寄存器字段。
no Read Behavior attribute	无法读取该寄存器。它是只写的。
no Write Behavior attribute	无法写入该寄存器。它是只读的。

图表 3-43 内存访问示例

内存访问	说明
R	只读寄存器字段
W	只写寄存器字段
R/W	读/写寄存器字段
R/W1C	您可以读取该寄存器字段。写 1 会清除它。
RC/W1C	读取该寄存器字段将清除它。写 1 会清除它。
R/Wo	可以读取该寄存器字段。你只能写入一次。

3.6.1 DWC_mshc_map/DWC_mshc_block 寄存器

3.6.1.1 SDMASA_R

- 寄存器名：SDMA 系统地址寄存器
- 描述：该寄存器用于配置 32 位块计数或在主机控制 2 寄存器中基于 Host Version 4 Enable 位配置 SDMA 系统地址。该寄存器适用于 SD 和 eMMC 模式。
- 大小：32 位
- 偏移量：0x0

图表 3-44 DMASA_R 寄存器字段说明

位段	名称	类型	说明
31:0	BLOCKCNT_SDMA	R/W	<p>32 位块计数（SDMA 系统地址）</p> <ul style="list-style-type: none"> ■ SDMA 系统地址（Host Version 4 Enable = 0）：这个寄存器包含 32 位寻址模式下 SDMA 传输的系统内存地址。当主控制器停止 SDMA 传输时，这个寄存器指向下一个相邻数据位置的系统地址。该寄存器只有在没有事务执行时才能访问。在数据传输期间读取此寄存器可能会返回无效值。 ■ 32 位块计数（Host Version 4 Enable = 1）：根据主机控制器版本 4.10 规范，这个寄存器被重新定义为 32 位块计数。主机控制器在每次块传输时减少该寄存器的块计数，当计数为零时数据传输停止。只有在没有事务执行时，才能访问该寄存器。在数据传输期间读取此寄存器可能会返回无效值。 <p>以下是 BLOCKCNT_SDMA 的值：</p>

			<ul style="list-style-type: none"> ■ 0xFFFF_FFFF: 4G - 1 块 ■ ... ■ 0x0000_0002: 2 块 ■ 0x0000_0001: 1 块 ■ 0x0000_0000: 停止计数 N <p>备注:</p> <ul style="list-style-type: none"> ■ 对于 Host Version 4 Enable = 0, 主机驱动程序在 ADMA 模式下时, 对这个寄存器中的系统地址不进行编程。系统地址必须在 ADMA 系统地址寄存器中编程。 ■ 对于 Host Version 4 Enable = 0, 当 Auto CMD23 为 non-DMA 和 ADMA 模式启用时, 主机驱动程序在这个寄存器中编程一个非零 32 位块计数值。Auto CMD23 不能与 SDMA 一起使用。 ■ 如果使用 32 位块计数寄存器而不是 16 位块计数寄存器, 则该寄存器必须使用非零值进行编程以进行数据传输。 <p>复位值: 0x0</p> <p>存在: 一直</p> <p>易失: 是</p>
--	--	--	---

3.6.1.2 BLOCKSIZE_R

- 寄存器名: 块大小寄存器
- 描述: 该寄存器用于配置一个 SDMA 缓冲区边界和数据块的字节数。该寄存器适用于 SD 和 eMMC 模式。
- 大小: 16 位
- 偏移量: 0x4

图表 3-45 BLOCKSIZE_R 寄存器字段说明

位段	名称	类型	说明
15	RSVD_BLOCKSIZE15	R	BLOCKSIZE_R 寄存器的该位是保留的。它总是返回 0。 复位值: 0x0 存在: 一直
14:12	SDMA_BUF_BDARY	R/W	SDMA 缓冲区边界 这些位规定了系统内存中相邻缓冲区的大小。SDMA 传输在由这些字段指定的每个边界处等待, 主机控制器生成 DMA 中断, 请求主机驱动程序更新 SDMA 系统地址寄存器。

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> ■ 0x0 (BYTES_4K): 4K bytes SDMA 缓冲区边界 ■ 0x1 (BYTES_8K): 8K bytes SDMA 缓冲区边界 ■ 0x2 (BYTES_16K): 16K bytes SDMA 缓冲区边界 ■ 0x3 (BYTES_32K): 32K bytes SDMA 缓冲区边界 ■ 0x4 (BYTES_64K): 64K bytes SDMA 缓冲区边界 ■ 0x5 (BYTES_128K): 128K bytes SDMA 缓冲区边界 ■ 0x6 (BYTES_256K): 256K bytes SDMA 缓冲区边界 ■ 0x7 (BYTES_512K): 512K bytes SDMA 缓冲区边界 复位值：0x0 存在：一直
11:0	XFER_BLOCK_SIZE	R/W	传输块大小 这些位规定了数据传输的块大小。内存设置为 512 字节。只有在没有事务在执行时才能访问它。传输过程中的读操作可能返回无效值，而写操作将被忽略。 以下是 XFER_BLOCK_SIZE 的值： <ul style="list-style-type: none"> ■ 0x1: 1 byte ■ 0x2: 2 bytes ■ 0x3: 3 bytes ■ ■ 0x1FF: 511 bytes ■ 0x200: 512 bytes ■ ■ 0x800: 2048 bytes 备注：为了数据传输，这个寄存器必须用非零值进行编程。 复位值：0x0 存在：一直

3.6.1.3 BLOCKCOUNT_R

- 寄存器名：16 位块计数寄存器
- 描述：这个寄存器用于配置数据块的数量。该寄存器适用于 SD 和 eMMC 模式。
- 大小：16 位

- 偏移量：0x6

图表 3-46 BLOCKCOUNT_R 寄存器字段说明

位段	名称	类型	说明
15:0	BLOCK_CNT	R/W	<p>16 位块计数</p> <ul style="list-style-type: none"> ■ 如果 Host Version 4 Enable 位设置为 0 或 16-bit Block Count 寄存器设置为非零，则 16-bit Block Count 寄存器被选中。 ■ 如果 Host Version 4 Enable 位设置为 1, 且 16-bit Block Count 寄存器设置为 0，则 32-bit Block Count 寄存器被选中。 <p>以下是 BLOCK_CNT 的值：</p> <ul style="list-style-type: none"> ■ 0x0：停止计数 ■ 0x1：1 块 ■ 0x2：2 块 ■ ... - ... ■ 0xFFFF：65535 块 <p>备注：对于 Host Version 4 Enable = 0，当 Auto CMD23 为 non-DMA 和 ADMA 模式启用时，在编程 32 位块计数寄存器之前，必须将该寄存器设置为 0000h。</p> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>

3.6.1.4 ARGUMENT_R

- 寄存器名：参数寄存器
- 描述：该寄存器用于配置 SD/eMMC 命令参数。
- 大小：32 位
- 偏移量：0x8

图表 3-47 ARGUMENT_R 寄存器字段说明

位段	名称	类型	说明
31:0	ARGUMENT	R/W	<p>命令参数</p> <p>这些位规定了 Command 格式的 39-8 位指定的 SD/eMMC 命令参数。</p>

位段	名称	类型	说明
			复位值：0x0 存在：一直

3.6.1.5 XFER_MODE_R

- 寄存器名：传输模式寄存器
- 描述：这个寄存器用于控制 SD/eMMC 模式下的数据传输操作。主机驱动程序在发出传输数据的命令之前设置这个寄存器。
- 大小：16 位
- 偏移量：0xc

图表 3-48 XFER_MODE_R 寄存器字段说明

位段	名称	类型	说明
15:9	RSVD	R	XFER_MODE_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
8	RESP_INT_DISABLE	R/W	<p>响应中断禁用</p> <p>主机控制器支持响应检查功能，以避免主机驱动程序的响应错误检查负担。控制器只能检查 R1 和 R5 类型的响应。如果主机驱动检查响应错误，设置此位为 0，等待命令完成中断（Command Complete Interrupt），然后检查响应寄存器。</p> <p>如果主机控制器检测响应错误，设置此位为 1，并且设置 Response Error Check Enable 位为 1。无论 Command Complete Signal Enable 的设置如何，Command Complete Interrupt（命令完成中断）都被这个位禁用。</p> <p>备注：在调优期间(当主机控制 2 寄存器中的执行调优位被设置时)，无论 Response Interrupt Disable 的设置如何，Command Complete Interrupt（命令完成中断）都不会生成。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (ENABLED): 响应中断使能。 ■ 0x1 (DISABLED): 响应中断禁用。 <p>复位值：0x0 存在：一直</p>

位段	名称	类型	说明
7	RESP_ERR_CHK_ENABLE	R/W	<p>响应错误检查启用</p> <p>主机控制器支持响应检查功能，以避免主机驱动程序响应错误检查的负担。控制器只能检查 R1 和 R5 类型的响应。如果主机控制器检查响应错误，将该位设置为 1，并将 Response Interrupt Disable 设置为 1。如果检测到错误，则在错误中断状态寄存器中生成响应错误中断。</p> <p>备注：</p> <ul style="list-style-type: none"> ■ 除 R1 和 R5 以外的任何响应类型都不能启用响应错误检查。 ■ 调优命令不能启用响应检查。 <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (DISABLED)：响应错误检查禁用。 ■ 0x1 (ENABLED)：响应错误检查启用。 <p>复位值：0x0</p> <p>存在：一直</p>
6	RESP_TYPE	R/W	<p>响应类型 R1/R5</p> <p>当选择响应错误检查时，该位选择 R1 或 R5 作为响应类型。</p> <p>在 R1 中检查的错误状态：</p> <ul style="list-style-type: none"> ■ OUT_OF_RANGE ■ ADDRESS_ERROR ■ BLOCK_LEN_ERROR ■ WP_VIOLATION ■ CARD_IS_LOCKED ■ COM_CRC_ERROR ■ CARD_ECC_FAILED ■ CC_ERROR ■ ERROR <p>在 R5 中检查的响应 flag：</p> <ul style="list-style-type: none"> ■ COM_CRC_ERROR ■ ERROR ■ FUNCTION_NUMBER ■ OUT_OF_RANGE <p>值：</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (RESP_R1): R1 (内存) ■ 0x1 (RESP_R5): R5 (SDIO) 复位值: 0x0
5	MULTI_BLK_SEL	R/W	多/单块选择 该位是在使用 DAT 线发出多块传输命令时设置的。如果该位设置为 0, 则不需要设置块计数寄存器。 值: <ul style="list-style-type: none"> ■ 0x0 (SINGLE): 单个块 ■ 0x1 (MULTI): 多个块 复位值: 0x0 存在: 一直
4	DATA_XFER_DIR	R/W	数据传输方向选择 这个位定义了 DAT 线数据传输的方向。该位由主机驱动程序设置为 1, 用于将数据从 SD/eMMC 卡传输到主机控制器, 其他所有命令都设置为 0。 值: <ul style="list-style-type: none"> ■ 0x1 (READ): 读 (卡到主机) ■ 0x0 (WRITE): 写 (主机到卡) 复位值: 0x0 存在: 一直
3:2	AUTO_CMD_ENABLE	R/W	自动命令启用 这个字段决定使用自动命令功能。 备注: 在 SDIO 中, 该字段必须设置为 00b (自动命令禁用)。 值: <ul style="list-style-type: none"> ■ 0x0 (AUTO_CMD_DISABLED): 自动命令禁用 ■ 0x1 (AUTO_CMD12_ENABLED): 自动命令 12 启用 ■ 0x2 (AUTO_CMD23_ENABLED): 自动命令 23 启用 ■ 0x3 (AUTO_CMD_AUTO_SEL): 自动命令自动选择 复位值: 0x0 存在: 一直
1	BLOCK_COUNT_ENABLE	R/W	块计数启用

位段	名称	类型	说明
			这个位用于启用块计数寄存器，该寄存器与多个块传输相关。如果该位设置为 0，则禁用块计数寄存器，这在执行无限传输时非常有用。使用 ADMA 时，主机驱动程序必须将该位设置为 0。 值： <ul style="list-style-type: none"> 0x1 (ENABLED): 启用 0x0 (DISABLED): 禁用 复位值: 0x0 存在: 一直
0	DMA_ENABLE	R/W	DMA 启用 这个位启用 DMA 功能。如果该位设置为 1，则当主机驱动程序写入命令寄存器时，DMA 操作开始。您可以通过使用主机控制 1 寄存器中的 DMA Select 选择 DMA 的其中一种模式。 值： <ul style="list-style-type: none"> 0x1 (ENABLED): DMA 数据传输 0x0 (DISABLED): 无数据传输或 Non-DMA 数据传输 复位值: 0x0 存在: 一直

3.6.1.6 CMD_R

- 寄存器名: 命令寄存器
- 描述: 该寄存器用于提供与命令和响应数据包相关的信息。这个寄存器适用于 SD/eMMC 模式。
- 大小: 16 位
- 偏移量: 0xe

图表 3-49 CMD_R 寄存器字段说明

位段	名称	类型	说明
15:14	RSVD	R	CMD_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
13:8	CMD_INDEX	R/W	命令索引 这些位被设置为命令编号，见命令格式的第 45-40 位规定。 复位值: 0x0

位段	名称	类型	说明
			存在：一直
7:6	CMD_TYPE	R/W	命令类型 这些位表示命令类型。 备注：当使用 CMD12/CMD52 发出 Abort CMD 或使用 CMD0/CMD52 复位 CMD 时, CMD_TYPE 字段应设置为 0x3。 值： <ul style="list-style-type: none"> ■ 0x3 (ABORT_CMD): 终止 ■ 0x2 (RESUME_CMD): 恢复 ■ 0x1 (SUSPEND_CMD): 暂停 ■ 0x0 (NORMAL_CMD): 正常 复位值：0x0 存在：一直
5	DATA_PRESENT_SEL	R/W	数据存在选择 该位设置为 1，表示数据存在，并且使用 DAT 线传输数据。 在以下情况下，该位设置为 0： <ul style="list-style-type: none"> ■ 使用 CMD 线命令 ■ 没有数据传输但在 DAT[0] line 上使用繁忙信号的命令 ■ 恢复命令 值： <ul style="list-style-type: none"> ■ 0x0 (NO_DATA): 无数据存在。 ■ 0x1 (DATA): 数据存在。 复位值：0x0 存在：一直
4	CMD_IDX_CHK_ENABLE	R/W	命令索引检查开启 该位使主机控制器能够检查响应中的索引字段，以验证它是否与命令索引具有相同的值。如果值不相同，则报告为命令索引错误。 备注： <ul style="list-style-type: none"> ■ 无响应、R2 响应、R3 响应、R4 响应命令的“Index Check enable”必须设置为 0。 ■ 对于调优命令，必须始终设置此位以启用索引检查。 值：

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用 ■ 0x1 (ENABLED): 开启 复位值: 0x0 存在: 一直
3	CMD_CRC_CHK_ENABLE	R/W	命令 CRC 校验使能 该位使主机控制器能够检查响应中的 CRC 字段。如果检测到错误, 则报告为命令 CRC 错误。 备注: <ul style="list-style-type: none"> ■ 对于无响应、R3 响应和 R4 响应命令, CRC Check enable 必须设置为 0。 ■ 对于调优命令, 该位必须始终设置为 1 以启用 CRC 检查。 值: <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用 ■ 0x1 (ENABLED): 启用 复位值: 0x0 存在: 一直
2	SUB_CMD_FLAG	R/W	子命令标志 这个位区分主命令和子命令。 值: <ul style="list-style-type: none"> ■ 0x0 (MAIN): 主命令 ■ 0x1 (SUB): 子命令 复位值: 0x0 存在: 一直
1:0	RESP_TYPE_SELECT	R/W	响应类型选择 这一位表示期望从卡中得到的响应类型。 值: <ul style="list-style-type: none"> ■ 0x0 (NO_RESP): 无响应 ■ 0x1 (RESP_LEN_136): 响应长度 136 ■ 0x2 (RESP_LEN_48): 响应长度 48 ■ 0x3 (RESP_LEN_48B): 响应长度 48; 响应后检查繁忙 (Check Busy)。

位段	名称	类型	说明
			复位值：0x0 存在：一直

3.6.1.7 RESP01_R

- 寄存器名：响应寄存器 0/1
- 描述：该寄存器存储 SD/eMMC 模式的响应字段的 39-08 位。在 UHS-II 模式下，这个寄存器存储 TRANS_ABORT CCMD 的响应。SD/eMMC 命令的最大响应长度为 128 位。这 128 位被分隔成 4 个 32 位寄存器：RESP01_R、RESP23_R、RESP45_R 和 RESP67_R。
- 大小：32 位
- 偏移量：0x10

图表 3-50 RESP01_R 寄存器字段说明

位段	名称	类型	说明
31:0	RESP01	R	命令响应 这些位反映 SD/eMMC 响应字段的 39-8 位。在 UHS-II 模式下，它存储 TRANS_ABORT CCMD 的响应。 备注：对于 Auto CMD，32 位响应（响应字段的 39-8 位）在 RESP67_R 寄存器中更新。 复位值：0x0 存在：一直 易失：是

3.6.1.8 RESP23_R

- 寄存器名：响应寄存器 2/3
- 描述：该寄存器存储 SD/eMMC 模式响应字段的 71-40 位。这个寄存器用于存储卡片的响应。响应最大长度为 128 位。
这 128 位被分隔成 4 个 32 位寄存器：RESP01_R、RESP23_R、RESP45_R 和 RESP67_R。在 UHS-II 模式下，该寄存器被保留。
- 大小：32 位
- 偏移量：0x14

图表 3-51 RESP23_R 寄存器字段说明

位段	名称	类型	说明
31:0	RESP23	R	命令响应

位段	名称	类型	说明
			这些位反映了 SD/eMMC 响应字段的 71-40 位。在 UHS-II 模式下，该寄存器被保留。 复位值：0x0 存在：一直 易失：是

3.6.1.9 RESP45_R

- 寄存器名：响应寄存器 4/5
- 描述：这个寄存器存储 SD/eMMC 模式下响应字段的 103-72 位。在 UHS-II 模式下，该寄存器用于存储低 4 字节 CMD12 响应。SD/eMMC 命令的响应长度最大为 128 位。这 128 位被分隔成 4 个 32 位寄存器：RESP01_R、RESP23_R、RESP45_R 和 RESP67_R。
- 大小：32 位
- 偏移量：0x18

图表 3-52 RESP45_R 寄存器字段说明

位段	名称	类型	说明
31:0	RESP45	R	命令响应 这些位反映响应字段的 103-72 位。在 UHSII 模式下，它存储低 4 字节 CMD12 响应。 复位值：0x0 存在：一直 易失：是

3.6.1.10 RESP67_R

- 寄存器名：响应寄存器 6/7
- 描述：这个寄存器存储 SD/eMMC 模式下响应字段的 135-104 位。在 UHSII 模式下，该寄存器存储高 4 字节的 CMD12 响应。SD/eMMC 响应最大支持 128 位。这 128 位被分隔成 4 个 32 位寄存器：RESP01_R、RESP 高 23_R、RESP45_R 和 RESP67_R。
- 大小：32 位
- 偏移量：0x1c

图表 3-53 寄存器 RESP67_R 字段说明

位段	名称	类型	说明
31:0	RESP67	R	命令响应

位段	名称	类型	说明
			这些位反映 SD/EMMC 响应字段的 135-104 位。UHS-II 模式下，存储高 4 字节的 CMD12 响应。 备注：对于自动 CMD，该寄存器也反映 32 位响应（响应字段的 39-8 位）。 复位值：0x0 存在：一直 易失：是

3.6.1.11 BUF_DATA_R

- 寄存器名：缓冲区数据端口寄存器
- 描述：这个寄存器用于访问数据包缓冲区。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：32 位
- 偏移量：0x20

图表 3-54 BUF_DATA_R 寄存器字段说明

位段	名称	类型	说明
31:0	BUF_DATA	R/W	缓冲区数据 这些位使能访问主机控制器数据包缓冲区。 复位值：0x0 存在：一直 易失：是

3.6.1.12 PSTATE_REG

- 寄存器名：当前状态寄存器
- 描述：这个寄存器指示主机控制器的当前状态。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：32 位
- 偏移量：0x24

图表 3-55 PSTATE_REG 寄存器字段说明

位段	名称	类型	说明
31	UHS2_IF_DETECT	R	UHS-II 接口检测 该位指示卡是否支持 UHS-II 接口。对于 SD/eMMC 模式，该位总是返回 0。

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 未检测到 HS-II 接口。 ■ 0x1 (TRUE): 检测到 HS-II 接口。 复位值: 0x0 存在: 一直 易失: 是
30	LANE_SYNC	R	通道同步 该位指示通道是否在 UHSII 模式下保持同步。对于 SD/eMMC 模式, 该位总是返回 0。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): UHS-II PHY 未初始化。 ■ 0x1 (TRUE): UHS-II PHY 初始化。 复位值: 0x0 存在: 一直 易失: 是
29	IN_DORMANT_ST	R	休眠状态 该位指示 UHS-II 通道是否在 UHS-II 模式下进入休眠状态。对于 SD/eMMC 模式, 该位总是返回 0。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 未处于休眠状态。 ■ 0x1 (TRUE): 处于休眠状态。 复位值: =mshc_present_state_in_dormant_reset_val 存在: 一直 易失: 是
28	SUB_CMD_STAT	R	子命令状态 该位用于区分主命令和子命令状态。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 主命令状态 ■ 0x1 (TRUE): 子命令状态 复位值: 0x0

位段	名称	类型	说明
			存在：一直 易失：是
27	CMD_ISSUE_ERR	R	由于错误未发出命令 如果由于除 Auto CMD12 错误外的错误而在设置命令寄存器后不能发出命令，则设置此位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：发出命令时没有错误。 ■ 0x1 (TRUE)：命令不能发出。 复位值：0x0 存在：一直 易失：是
26	RSVD_26	R	PRESENT_ST_R 寄存器的这个位是保留位。它总是返回 0。 复位值：0x0 存在：一直 易失：是
25	HOST_REG_VOL	R	主机稳压电压稳定 该位通过开关 UHS-I 模式电压，检查主机稳压电压是否稳定。 该位反映了 host_reg_vol_stable 信号的同步值。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：主机稳压电压不稳定。 ■ 0x1 (TRUE)：主机稳压电压稳定。 复位值：0x0 存在：一直 易失：是
24	CMD_LINE_LVL	R	命令行信号电平 该位用于检查 CMD 线电平，以便从错误中恢复并调试。 这些位反映了 sd_cmd_in 信号的值。UHS-II 模式下，该位是无关位，并且总是返回 0。 复位值：0x0 存在：一直 易失：是

位段	名称	类型	说明
23:20	DAT_3_0	R	<p>DAT[3:0]线路信号电平</p> <p>该位用于检查 DAT 线电平，以便从错误中恢复并进行调试。这些位反映了 sd_dat_in（低四位）信号的值。UHS-II 模式下，该位是无关位，并且总是返回 0。</p> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
19	WR_PROTECT_SW_LVL	R	<p>写保护开关管脚电平</p> <p>该位只支持内存和组合卡。这个位反映了 card_write_prot 信号的同步值。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 写保护 ■ 0x1 (TRUE): 写使能 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
18	CARD_DETECT_PIN_LEVEL	R	<p>卡检测管脚电平</p> <p>这个位反映了 card_detect_n 信号的反向同步值。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无卡 ■ 0x1 (TRUE): 有卡 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
17	CARD_STABLE	R	<p>卡稳定</p> <p>此位指示卡检测管脚电平的稳定性。如果该位设置为 1 并且 CARD_INSERTED 位的值为 0，则卡未检测到。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 复位或去抖 ■ 0x1 (TRUE): 无卡或插入 <p>复位值：0x0</p>

位段	名称	类型	说明
			存在：一直 易失：是
16	CARD_INSERTED	R	卡插入 此位表示是否已插入卡片。主机控制器对这个信号去抖，因此主机驱动程序不需要等待它稳定下来。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 复位，去抖或无卡 ■ 0x1 (TRUE): 卡插入 复位值：0x0 存在：一直 易失：是
15:12	RSVD_15_12	R	RESENT_STAT_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：总是 易失：是
11	BUF_RD_ENABLE	R	缓冲区读使能 该位用于 non-DMA 传输。如果主机缓冲区中存在有效数据，则设置此位。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用读。 ■ 0x1 (ENABLED): 使能读。 复位值：0x0 存在：一直 易失：是
10	BUF_WR_ENABLE	R	缓冲区写使能 该位用于 non-DMA 传输。如果有写入数据的空间，则设置此位。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用写。 ■ 0x1 (ENABLED): 使能写。

位段	名称	类型	说明
			复位值: 0x0 存在: 一直 易失: 是
9	RD_XFER_ACTIVE	R	读传输激活 该位表示 SD/eMMC 模式的读传输是否激活。UHS-II 模式下, 该位是无关位, 并始终返回 0。 值: <ul style="list-style-type: none"> ■ 0x0 (INACTIVE): 无有效数据 ■ 0x1 (ACTIVE): 传输数据 复位值: 0x0 存在: 一直 易失: 是
8	WR_XFER_ACTIVE	R	写传输激活 该状态表示 SD/eMMC 模式下写传输是否激活。UHS-II 模式下, 该位是无关位, 总是返回 0。 值: <ul style="list-style-type: none"> ■ 0x0 (INACTIVE): 无有效数据 ■ 0x1 (ACTIVE): 传输数据 复位值: 0x0 存在: 一直 易失: 是
7:4	DAT_7_4	R	DAT[7:4]线信号电平 该位用于检查 DAT 线电平, 以便从错误中恢复和调试。这些位反映了 sd_dat_in (高四位) 信号的值。UHS-II 模式下, 该位是无关位, 并且总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
3	RE_TUNE_REQ	R	重调请求 DWC_mshc 不生成返回请求。软件必须维护返回定时器。 复位值: 0x0

位段	名称	类型	说明
			存在：一直 易失：是
2	DAT_LINE_ACTIVE	R	<p>DAT 线激活（仅限 SD/eMMC 模式）</p> <p>该位表示 SD/eMMC 总线上的 DAT 线是否在使用中。UHS-II 模式下，该位是无关位。</p> <p>在读事务的情况下，该位表示是否在 SD/eMMC 总线上执行读传输。</p> <p>在写事务的情况下，该位表示是否在 SD/eMMC 总线上执行写传输。</p> <p>对于繁忙（busy）的命令，该状态表示繁忙（busy）的命令是在 SD 总线上还是 eMMC 总线上执行。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (INACTIVE): DAT 线未激活。 ■ 0x1 (ACTIVE): DAT 线已激活。 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
1	CMD_INHIBIT_DAT	R	<p>禁止命令（DAT）</p> <p>该位适用于 SD/eMMC 模式，当 DAT 线活跃或 Read transfer active 设置为 1 时产生。该位设置为 0，表示主机控制器可以发出后续的 SD/eMMC 命令。UHS-II 模式下，该位是无关位，并且总是返回 0。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (READY): 能发出使用 DAT 线的命令。 ■ 0x1 (NOT_READY): 不能发出使用 DAT 线的命令。 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
0	CMD_INHIBIT	R	<p>禁止命令（CMD）</p> <p>该位表示：</p> <ul style="list-style-type: none"> ■ SD/eMMC 模式：该位设置为 0，表示 CMD 线未被使用，主机控制器可以通过 CMD 线发出 SD/eMMC 命令。该位在写入命令寄存器时设置。当收到命令响应时，该位被清

位段	名称	类型	说明
			<p>除。auto CMD12/23 的响应不会清除该位，但是读/写命令的响应会清除该位。</p> <ul style="list-style-type: none"> ■ UHS-II 模式：该位为 0 时，表示命令数据包可以由主机控制器发出。 <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (READY)：主机控制器准备好发出命令。 ■ 0x1 (NOT_READY)：主机控制器未准备好发出命令。 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>

3.6.1.13 HOST_CTRL1_R

- 寄存器名：主机控制 1 寄存器
- 描述：这个寄存器用于控制主机控制器的操作。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：8 位
- 偏移量：0x28

图表 3-56 HOST_CTRL1_R 寄存器字段说明

位段	名称	类型	说明
7	CARD_DETECT_SIG_SEL	R/W	<p>卡检测信号选择</p> <p>该位为卡片检测选择源。当卡检测的源被切换时，在切换期间中断必须禁用。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (CARD_DT_TEST_LEVEL)：选择卡检测测试电平。(用于测试目的) ■ 0x0 (SDCD_PIN)：SDCD# (card_detect_n signal) 被选中。(正常使用) <p>复位值：0x0</p> <p>存在：一直</p>
6	CARD_DETECT_TEST_LVL	R/W	<p>卡片检测测试电平</p> <p>当 Card Detect Signal Selection 设置为 1 时，该位被启用，并且它显示是否有卡插入。</p> <p>值：</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x1 (CARD_INSERTED): 卡已插入。 ■ 0x0 (No_CARD): 无卡。 复位值: 0x0 存在: 一直
5	EXT_DAT_XFER	R/W	扩展数据传输宽度 该位控制嵌入式设备的 8 位总线宽度模式。此位不适用于 UHS-II 模式。 值: <ul style="list-style-type: none"> ■ 0x1 (EIGHT_BIT): 8 位总线宽度 ■ 0x0 (DEFAULT): 总线宽度由数据传输宽度选定 复位值: 0x0 存在: 一直
4:3	DMA_SEL	R/W	DMA 选择 此字段用于选择 DMA 类型。 当主机控制 2 寄存器中的 Host Version 4 Enable 是 1: <ul style="list-style-type: none"> ■ 0x0: SDMA 被选中。 ■ 0x1: 保留。 ■ 0x2: ADMA2 被选中。 ■ 0x3: ADMA2 or ADMA3 被选中。 当主机控制 2 寄存器中的 Host Version 4 Enable 是 0: <ul style="list-style-type: none"> ■ 0x0: SDMA 被选中。 ■ 0x1: 保留。 ■ 0x2: 32 位地址 ADMA2 被选中。 ■ 0x3: 64 位地址 ADMA2 被选中。 值: <ul style="list-style-type: none"> ■ 0x0 (SDMA): SDMA 被选中。 ■ 0x1 (RSVD_BIT): 保留。 ■ 0x2 (ADMA2): ADMA2 被选中。 ■ 0x3 (ADMA2_3): ADMA2 或 ADMA3 被选中。 复位值: 0x0 存在: 一直

位段	名称	类型	说明
2	HIGH_SPEED_EN	R/W	<p>高速使能(仅支持 SD/eMMC 模式)</p> <p>在 SD/eMMC 模式下, 该位用于确定高速模式预置值的选定。在设置此位之前, 主机驱动程序检查 Capabilities 寄存器中的高速支持。</p> <p>备注: 无论该位设置如何, DWC_MSHC 总是在 cclk_tx 时钟的上升边缘输出 sd_cmd_out 和 sd_dat_out 行。请参考移动存储主机控制器用户指南中关于 SD/eMMC 卡时钟要求的“链接时钟输入/输出接口”章节。</p> <p>UHS-II 模式下, 该位是无关位。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x1 (HIGH_SPEED): 高速模式 ■ 0x0 (NORMAL_SPEED): 正常速度模式 <p>复位值: 0x0</p> <p>存在: 一直</p>
1	DAT_XFER_WIDTH	R/W	<p>数据传输的宽度</p> <p>对于 SD/eMMC 模式, 该位选择主机控制器的数据传输宽度。主机驱动程序将其设置为与 SD/eMMC 卡的数据宽度相匹配。UHS-II 模式下, 该位是无关位。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x1 (FOUR_BIT): 4 位模式 ■ 0x0 (ONE_BIT): 1 位模式 <p>复位值: 0x0</p> <p>存在: 一直</p>
0	LED_CTRL	R/W	<p>LED 控制</p> <p>此位用于警告用户在 SD 卡被访问时不要拔出卡。这个值反映在 led_control 信号上。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (OFF): LED 关。 ■ 0x1 (ON): LED 开。 <p>复位值: 0x0</p> <p>存在: 一直</p>

3.6.1.14 PWR_CTRL_R

- 寄存器名：电源控制寄存器
- 描述：这个寄存器用于控制卡的总线电源。该寄存器适用于 SD、eMMC 和 UHS-II 模式。
- 大小：8 位
- 偏移量：0x29

图表 3-57 PWR_CTRL_R 寄存器字段说明

位段	名称	类型	说明
7:5	SD_BUS_VOL_VDD2	R/W	<p>为 VDD2 选择 SD 总线电压。</p> <p>该位决定了 UHS-II 卡 VDD2 的供电电压范围。如果 Capabilities 寄存器中的 1.8V VDD2 支持设置为 1，该位可以设置为 0x5。对于 SD/eMMC 卡，该位是无关位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x7 (NOT_USED7): 未使用。 ■ 0x6 (NOT_USED6): 未使用。 ■ 0x5 (V_1_8): 1.8V ■ 0x4 (V_1_2): 预留给 1.2V。 ■ 0x3 (RSVD3): 预留 ■ 0x2 (RSVD2): 预留 ■ 0x1 (RSVD1): 预留 ■ 0x0 (NO_VDD2): VDD2 不支持。 <p>复位值：0x0</p> <p>存在：一直</p>
4	SD_BUS_PWR_VDD2	R/W	<p>用于 VDD2 的 SD 总线电源。</p> <p>该位开启 UHS-II 卡的 VDD2 电源。这个设置在 DWC_mshc 的 sd_vdd2_on 输出上可用，以便它用来控制卡的 VDD2 电源。</p> <p>对于 SD/eMMC 卡，该位是无关位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (OFF): 电源关。 ■ 0x1 (ON): 电源开。 <p>复位值：0x0</p> <p>存在：一直</p>

位段	名称	类型	说明
3:1	SD_BUS_VOL_VDD1	R/W	<p>VDD1 的 SD 总线电压选择/VDD 的 eMMC 总线电压选择</p> <p>这些位使主机驱动程序能够选择 SD/eMMC 卡的电压等级。在设置这个寄存器之前,主机驱动程序检查 Capabilitie 寄存器中的电压支持位。如果选择不支持的电压,主机系统不提供 SD 总线电压。该字段中设置的值在 DWC_mshc 输出信号 (sd_vdd1_sel) 上可用, 该信号被电压开关电路使用。</p> <p>SD 总线电压选择选项:</p> <ul style="list-style-type: none"> ■ 0x7: 3.3V (典型) ■ 0x6: 3.0V (典型) ■ 0x5: 1.8V 对于嵌入式是 (典型) ■ 0x4:0x0: 保留 <p>eMMC 总线电压选择选项:</p> <ul style="list-style-type: none"> ■ 0x7: 3.3V (典型) ■ 0x6: 1.8V (典型) ■ 0x5: 1.2V (典型) ■ 0x4:0x0: 保留 <p>值:</p> <ul style="list-style-type: none"> ■ 0x7 (V_3_3): 3.3V (典型) ■ 0x6 (V_3_0): 3.0V (典型) ■ 0x5 (V_1_8): 1.8V 对于嵌入式是 (典型) ■ 0x4 (RSVD4): 保留 ■ 0x3 (RSVD3): 保留 ■ 0x2 (RSVD2): 保留 ■ 0x1 (RSVD1): 保留 ■ 0x0 (RSVD0): 保留 <p>复位值: 0x0</p> <p>存在: 一直</p>
0	SD_BUS_PWR_VDD1	R/W	<p>VDD1 的 SD 总线电源</p> <p>该位使卡的 VDD1 电源生效。这个设置在 DWC_mshc 的 sd_vdd1_on 输出上可用, 以便它控制卡的 VDD1 电源。在设置该位之前, SD 主机驱动会设置 SD Bus Voltage Select 位。如果主机控制器检测到无卡状态, 则该位被清除。</p>

位段	名称	类型	说明
			<p>在 SD 模式下, 如果该位被清除, 主机控制器通过清除 CLK_CTRL_R 寄存器的 SD_CLK_IN 位来停止 SD 时钟。</p> <p>UHS-II 模式下, 清除该位之前, 主机驱动会清除 CLK_CTRL_R 寄存器的 SD_CLK_IN 位。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (OFF): 电源关。 ■ 0x1 (ON): 电源开。 <p>复位值: 0x0</p> <p>存在: 一直</p>

3.6.1.15 BGAP_CTRL_R

- 寄存器名: 块间隙控制寄存器
- 描述: 主机驱动程序使用这个寄存器来控制任何与块间隙相关的操作。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小: 8 位
- 偏移量: 0x2a

图表 3-58 BGAP_CTRL_R 寄存器字段说明

位段	名称	类型	说明
7:4	RSVD_7_4	R	<p>块间隙控制寄存器的这些位是保留的。它们总是返回 0。</p> <p>复位值: 0x0</p> <p>存在: 一直</p>
3	INT_AT_BGAP	R/W	<p>块间隙的中断</p> <p>该位仅在 SDIO 卡的 4 位模式下有效, 用于在中断周期中选择采样点。设置为 1, 可以在块间隙处对多个块传输进行中断检测。在 UHS-II 模式下, 该位被禁用。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (DISABLE): 禁用 ■ 0x1 (ENABLE): 开启 <p>复位值: 0x0</p> <p>存在: 一直</p>
2	RD_WAIT_CTRL	R/W	<p>读等控制</p> <p>如果卡支持读等待, 该位使能读等待协议来终止使用 DAT[2]</p>

位段	名称	类型	说明
			<p>线读数据。否则，主控制器必须停止卡时钟来保存读取的数据。UHS-II 模式下，“读等待”是关闭的。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (DISABLE): 禁用读等待控制。 ■ 0x1 (ENABLE): 开启读等待控制。 <p>复位值：0x0</p> <p>存在：一直</p>
1	CONTINUE_REQ	R/W	<p>继续请求</p> <p>该位用于重新启动那些使用 Stop At Block Gap Request 停止的事务。当事务重新启动时，主机控制器自动清除此位。如果 block gap request 设置为 1，对该位的任何写入都被忽略。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (NO_AFFECT): 无影响 ■ 0x1 (RESTART): 重启 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
0	STOP_BG_REQ	R/W	<p>在块间隙请求处停止</p> <p>对于 non-DMA、SDMA 和 ADMA 传输，该位用于在下一个块间隙处停止执行读写事务。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (XFER): 传输 ■ 0x1 (STOP): 停止 <p>复位值：0x0</p> <p>存在：一直</p>

3.6.1.16 WUP_CTRL_R

- 寄存器名：唤醒控制寄存器
- 描述：这个寄存器对于主机控制器是强制性的，但是唤醒功能依赖于主机控制器系统的硬件和软件。主机驱动通过在电源控制寄存器中设置 SD 总线电源（SD Bus Power）为 1 来维持 SD 总线上的电压，同时需要通过卡中断来唤醒事件。

- 大小：8 位
- 偏移量：0x2b

图表 3-59 WUP_CTRL_R 寄存器字段说明

位段	名称	类型	说明
7:3	RSVD_7_3	R	唤醒控制寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
2	CARD_REMOVAL	R/W	SD 卡移除的唤醒事件使能 该位通过正常中断状态（Normal Interrupt Status）寄存器中的卡移除断言启用唤醒事件。对于 SDIO 卡，卡信息结构（CIS）寄存器中的唤醒支持(FN_WUS)不影响该位。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用 ■ 0x1 (ENABLED): 使能 复位值：0x0 存在：一直
1	CARD_INSERT	R/W	SD 卡插入的唤醒事件使能 该位通过正常中断状态（Normal Interrupt Status）寄存器中的卡插入断言启用唤醒事件。卡信息结构（CIS）中的 FN_WUS(唤醒支持)不影响该位。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用 ■ 0x1 (ENABLED): 使能 复位值：0x0 存在：一直
0	CARD_INT	R/W	卡中断的启用唤醒事件 该位通过正常中断状态（Normal Interrupt Status）寄存器中的卡中断断言启用唤醒事件。如果卡信息结构（CIS）中的 FN_WUS(唤醒支持)设置为 1，该位也可以设置为 1。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用 ■ 0x1 (ENABLED): 使能 复位值：0x0

位段	名称	类型	说明
			存在：一直

3.6.1.17 CLK_CTRL_R

- 寄存器名：时钟控制寄存器
- 描述：该寄存器以 SD/eMMC 模式控制 SDCLK(卡钟)，以 UHS-II 模式控制 RCLK。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x2c

图表 3-60 CLK_CTRL_R 寄存器字段说明

位段	名称	类型	说明
15:8	FREQ_SEL Divided clock	R/W	<p>SDCLK/RCLK 频率选择</p> <p>这些位用于选择 SDCLK 信号的频率。这些位依赖于主机控制 2 寄存器中的预置值启用 (Preset Value Enable) 的设置。当 Preset Value Enable = 0 时，这些位由主机驱动设置。如果 Preset Value Enable = 1，这些位自动设置为预置值寄存器中指定的值。该值反映在 card_clk_freq_sel 信号的低 8 位。</p> <p>10 位分频时钟模式：</p> <ul style="list-style-type: none"> ■ 0x3FF: 1/2046 分频时钟 ■ ■ N: 1/2N 分频时钟 ■ ■ 0x002: 1/4 分频时钟 ■ 0x001: 1/2 分频时钟 ■ 0x000: 基础时钟 (10MHz - 255MHz) <p>可编程时钟模式:使主机系统选择一个细粒度的 SD 时钟频率：</p> <ul style="list-style-type: none"> ■ 0x3FF: 基础时钟*M/1024 ■ ■ N-1: 基础时钟*M/N ■ ■ 0x002: 基础时钟*M/3 ■ 0x001: 基础时钟*M/2

位段	名称	类型	说明
			<p>■ 0x000: 基础时钟*M</p> <p>复位值: 0x0</p> <p>存在: 一直</p> <p>易失: 是</p>
7:6	UPPER_FREQ_SEL	R/W	<p>这些位指定 10 位 SDCLK/RCLK 频率选择控制的高 2 位。该值反映在 card_clk_freq_sel 信号的高 2 位上。</p> <p>复位值: 0x0</p> <p>存在: 一直</p> <p>易失: 是</p>
5	CLK_GEN_SELECT	R/W	<p>时钟发生器选择</p> <p>此位是用来选择时钟发生器模式的</p> <p>当 Preset Value Enable = 0 时, 该位由主机驱动设置。如果 Preset Value Enable = 1, 该位自动设置为预置值寄存器中指定的值。该值反映在 card_clk_gen_sel 信号上。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 分频时钟模式 ■ 0x1 (TRUE): 可编程时钟模式 <p>复位值: 0x0</p> <p>存在: 一直</p> <p>易失: 是</p>
4	RSVD_4	R	<p>CLK_CTRL_R 寄存器的这一部分是保留的。它总是返回 0。</p> <p>复位值: 0x0</p> <p>存在: 一直</p>
3	PLL_ENABLE	R/W	<p>PLL 使能</p> <p>该位用于激活锁相环(适用于 Host Version 4 Enable = 1)。当 Host Version 4 Enable = 0 时, INTERNAL_CLK_EN 位可以用来激活 PLL。该值反映在 card_clk_en 信号上。</p> <p>注意: 当 Host Version 4 Enable = 1 时, 如果不使用该位来激活 PLL, 建议将该位设置为“1”。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): PLL 处于低功耗模式 ■ 0x1 (TRUE): PLL 使能

位段	名称	类型	说明
			复位值: 0x0 存在: 一直
2	SD_CLK_EN	R/W	SD/eMMC 时钟启用 当设置为 0 时, 该位停止 SDCLK 或 RCLK。当该位设置为 0 时, SDCLK/RCLK 频率选择位可以更改。该值反映在 clk2card_on 管脚上。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 禁用提供 SDCLK/RCLK。 ■ 0x1 (TRUE): 启用提供 SDCLK/RCLK。 复位值: 0x0 存在: 一直
1	INTERNAL_CLK_STABLE	R	内部时钟稳定 该位使主机驱动程序能够在设置了内部时钟使能位和 PLL 使能位之后检查两次时钟稳定性。该位反映了内部时钟使能位为 1 后 intclk_stable 信号的同步值, 也反映了 PLL 使能位为 1 后 card_clk_stable 信号的同步值。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 未就绪 ■ 0x1 (TRUE): 就绪 复位值: 0x0 存在: 一直 易失: 是
0	INTERNAL_CLK_EN	R/W	内部时钟使能 当主机驱动程序不使用主机控制器或主机控制器等待唤醒中断时, 该位设置为 0。主控制器必须停止其内部时钟以进入一个非常低的功耗状态。但是, 寄存器仍然可以读取和写入。该值反映在 intclk_en 信号上。 注意: 如果该位不用于控制内部时钟 (基本时钟和主时钟), 建议设置为 “1” 。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 停止 ■ 0x1 (TRUE): 震荡

位段	名称	类型	说明
			复位值：0x0 存在：一直

3.6.1.18 TOUT_CTRL_R

- 寄存器名：超时控制寄存器
- 描述：该寄存器用于根据 Capabilities 寄存器定义的计时器时钟为 SD/eMMC 模式设置数据超时计数器值，同时初始化主机控制器。
- 大小：8 位
- 偏移量：0x2e

图表 3-61 TOUT_CTRL_R 寄存器字段说明

位段	名称	类型	说明
7:4	RSVD_7_4	R	超时控制寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
3:0	TOUT_CNT	R/W	数据超时计数器值 这个值决定了检测 DAT 线超时的时间间隔。超时 (Timeout) 时钟频率由基本时钟 TMCLK 值除以该值产生。在设置这个寄存器时，通过清除 Data timeout Error Status Enable(在 Error Interrupt Status Enable 寄存器中)来防止意外超时事件。这些位的值是： <ul style="list-style-type: none"> ■ 0xF：保留 ■ 0x：TMCLK x 2²⁷ ■ ■ 0x1：TMCLK x 2¹⁴ ■ 0x0：TMCLK x 2¹³ 注意：在 eMMC 模式下启动期间，应用程序必须在此位配置启动数据超时值（大约 1 秒）。 复位值：0x0 存在：一直

3.6.1.19 SW_RST_R

- 寄存器名：软件复位寄存器

- 描述：这个寄存器通过向寄存器的每一位写入 1 来产生一个复位脉冲。复位完成后，主机控制器清除每个位。由于需要一些时间来完成软件复位，主机驱动程序确认这些位是 0。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：8 位
- 偏移量：0x2f

图表 3-62 SW_RST_R 寄存器字段说明

位段	名称	类型	说明
7:3	RSVD_7_3	R	SW_RST_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直 易失：是
2	SW_RST_DAT	R/W	DAT 线的软件复位 该位用于 SD/eMMC 模式，它只复位部分数据电路。DMA 电路同样被复位。以下寄存器和位被该位清除： <ul style="list-style-type: none"> ■ 缓冲区数据端口寄存器 <ul style="list-style-type: none"> – 清除缓冲区并初始化。 ■ 当前状态寄存器 <ul style="list-style-type: none"> – 启用缓冲区读功能 – 启用缓冲区写功能 – 读传输活跃 – 写传输活跃 – DAT 线活跃 – 禁止命令 (DAT) ■ 块间隙控制寄存器 <ul style="list-style-type: none"> – 继续请求 – 在块间隙处停止请求 ■ 正常中断状态寄存器 <ul style="list-style-type: none"> – 缓冲区读就绪 – 缓冲区写就绪 – DMA 中断 – 块间隙事件 – 传输完成 UHS-II 模式下，该位应设置为 0 值：

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (FALSE): 工作 ■ 0x1 (TRUE): 复位 复位值: 0x0 存在: 一直 易失: 是
1	SW_RST_CMD	R/W	CMD 线的软件复位 该位只复位命令电路的一部分, 以便能够发出命令。它也用于初始化 UHS-II 命令电路。此复位只对命令发出电路有效 (包括与禁止命令 (CMD) 控制相关的响应错误状态), 不影响数据传输电路。在处理子命令响应错误时, 主机控制器即使执行了复位也可以继续数据传输。 以下寄存器和位被该位清除: <ul style="list-style-type: none"> ■ 当前状态寄存器: 禁止命令 (CMD) 位 ■ 正常中断状态寄存器: 命令完成位 ■ 错误中断状态: 与禁止命令 (CMD) 位错误中断状态相关的响应错误状态 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 工作 ■ 0x1 (TRUE): 复位 复位值: 0x0 存在: 一直 易失: 是
0	SW_RST_ALL	R/W	全部的软件复位 这种复位会影响除卡检测电路之外的整个主控制器。在初始化期间, 主机驱动程序将该位设置为 1 以复位主机控制器。除 Capabilities 寄存器外, 所有寄存器都被复位。如果该位设置为 1, 主机驱动程序必须发出复位命令并重新初始化卡。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 工作 ■ 0x1 (TRUE): 复位 复位值: 0x0 存在: 一直

位段	名称	类型	说明
			易失：是

3.6.1.20 NORMAL_INT_STAT_R

- 寄存器名：正常中断状态寄存器
- 描述：这个寄存器反映了正常中断的状态。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x30

图表 3-63 NORMAL_INT_STAT_R 寄存器字段说明

位段	名称	类型	说明
15	ERR_INTERRUPT	R	错误中断 如果错误中断状态寄存器中的任何位被设置，那么这个位被设置。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：无错误 ■ 0x1 (TRUE)：有错误 复位值：0x0 存在：一直 易失：是
14	CQE_EVENT	R/W1C	命令队列事件 如果在 eMMC/SD 模式下命令队列/加密相关事件发生，则设置此状态。详细信息请阅读 CQHCI 的 CQIS/CRNQIS 寄存器相关内容。在 UHS-II 模式下，该位为无关位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：无事件 ■ 0x1 (TRUE)：检测到命令队列事件 复位值：0x0 存在：一直 易失：是
13	FX_EVENT	R	FX 事件 当响应寄存器的 R[14] 为 1，并且在传输模式寄存器中响应类型 R1/R5 为 0 时，设置该状态。这个中断与响应检查功能一起使用。

位段	名称	类型	说明
			值： ■ 0x0 (FALSE): 无事件 ■ 0x1 (TRUE): 检测到 FX 事件 复位值: 0x0 存在: 一直 易失: 是
12	RE_TUNE_EVENT	R	重调事件 如果重新调优请求 (Re-Tuning Request) 从 0 更改为 1, 则设置此位。不支持重新调优请求。 复位值: 0x0 存在: 一直 易失: 是
11	INT_C	R	INT_C (嵌入式) 如果 INT_C 启用, 并且 INT_C#管脚是低电平, 则设置此位。不支持 INT_C#管脚。 复位值: 0x0 存在: 一直 易失: 是
10	INT_B	R	INT_B (嵌入式) 如果 INT_B 被启用并且 INT_B#管脚处于低电平, 则设置该位。不支持 INT_B#管脚。 复位值: 0x0 存在: 一直 易失: 是
9	INT_A	R	INT_A (嵌入式) 如果 INT_A 被启用并且 INT_A#管脚处于低电平, 则该位被设置。不支持 INT_A#管脚。 复位值: 0x0 存在: 一直 易失: 是
8	CARD_INTERRUPT	R	卡中断

位段	名称	类型	说明
			这个位反映了同步值： <ul style="list-style-type: none"> ■ SD 模式的 DAT[1]中断输入 ■ UHS-II 模式的 DAT[2]中断输入 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 没有卡中断。 ■ 0x1 (TRUE): 生成卡中断。 复位值：0x0 存在：一直 易失：是
7	CARD_REMOVAL	R/W1C	卡移除 如果卡插入 (Card Inserted) 在当前状态寄存器中从 1 变化为 0, 则设置此位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 卡状态稳定或去抖 ■ 0x1 (TRUE): 卡被移除 复位值：0x0 存在：一直 易失：是
6	CARD_INSERTION	R/W1C	卡插入 如果当前状态寄存器中的卡插入 (Card Inserted) 从 0 变为 1, 则设置此位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 卡状态稳定或去抖 ■ 0x1 (TRUE): 卡被插入 复位值：0x0 存在：一直 易失：是
5	BUF_RD_READY	R/W1C	缓冲区读就绪 如果缓冲区读使能 (Buffer Read Enable) 从 0 变为 1, 则设置此位。 值：

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (FALSE): 缓冲区读未就绪。 ■ 0x1 (TRUE): 缓冲区读就绪。 复位值: 0x0 存在: 一直 易失: 是
4	BUF_WR_READY	R/W1C	缓冲区写就绪 当缓冲区写使能 (Buffer Write Enable) 从 0 变为 1 时, 该位被设置。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 缓冲区写未就绪。 ■ 0x1 (TRUE): 缓冲区写就绪。 复位值: 0x0 存在: 一直 易失: 是
3	DMA_INTERRUPT	R/W1C	DMA 中断 如果主机控制器在传输过程中检测到 SDMA 缓冲区边界, 则设置此位。在 ADMA 的情况下, 通过在描述符表中设置 Int 字段, 主机控制器生成这个中断。传输完成后不会产生此中断。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无 DMA 中断。 ■ 0x1 (TRUE): DMA 中断产生。 复位值: 0x0 存在: 一直 易失: 是
2	BGAP_EVENT	R/W1C	块间隙事件 当读/写事务都因块间隙请求而在块间隙处停止时, 设置该位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无块间隙事件 ■ 0x1 (TRUE): 事务在 block gap 处停止 复位值: 0x0

位段	名称	类型	说明
			存在：一直 易失：是
1	XFER_COMPLETE	R/W1C	传输完成 当一个读/写传输和一个状态为的命令完成时设置该位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 未完成 ■ 0x1 (TRUE): 命令执行完成 复位值：0x0 存在：一直 易失：是
0	CMD_COMPLETE	R/W1C	命令完成 在 SD/eMMC 模式下，该位在除 Auto CMD12 和 Auto CMD23 以外的响应结束位时设置。在 UHSII 模式下，当收到响应数据包时设置该位。 当传输模式寄存器中的响应中断禁用（Response Interrupt Disable）设置为 1 时，不会产生此中断。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无命令完成 ■ 0x1 (TRUE): 命令完成 复位值：0x0 存在：一直 易失：是

3.6.1.21 ERROR_INT_STAT_R

- 寄存器名：错误中断状态寄存器
- 描述：当错误中断状态使能（Error Interrupt Status Enable）并且至少一个状态被设置为 1 时，该寄存器启用中断。写入 1 时清除该位，写入 0 时保留该位不变。在这个寄存器中定义的信号可以通过错误中断状态使能（Error Interrupt Status Enable）寄存器使能，但不能通过错误中断信号使能（Error Interrupt Signal Enable）寄存器使能。一个寄存器写操作可以清除多个状态。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x32

图表 3-64 ERROR_INT_STAT_R 寄存器字段说明

位段	名称	类型	说明
15	VENDOR_ERR3	R/W1C	ERROR_INT_STAT_R 寄存器的这个位(VENDOR_ERR3)是保留的。它总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
14	VENDOR_ERR2	R/W1C	ERROR_INT_STAT_R 寄存器的这个位(VENDOR_ERR2)是保留的。它总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
13	VENDOR_ERR1	R/W1C	ERROR_INT_STAT_R 寄存器的这个位(VENDOR_ERR1)是保留的。它总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
12	BOOT_ACK_ERR	R/W1C	引导确认错误 该位在引导确认超时或检测到引导确认状态的值不是 010 时设置。这只适用于在 eMMC 模式下需要引导确认的情况。 UHS-II 模式下, 该位是无关位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 复位值: 0x0 存在: 一直 易失: 是
11	RESP_ERR	R/W1C	响应错误 主机控制器版本 4.00 支持响应错误检查功能, 避免 DMA 执行期间主机驱动程序响应错误检查的负担。如果在传输模式寄存器中响应错误检查使能 (Response Error Check Enable) 设置为 1, 主机控制器检查 R1 或 R5 响应。如果在响应中检测到错误, 则该位设置为 1。这适用于 SD/eMMC 模

位段	名称	类型	说明
			<p>式。</p> <p>UHS-II 模式下, 该位是无关位。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 <p>复位值: 0x0</p> <p>存在: 一直</p> <p>易失: 是</p>
10	TUNING_ERR	R/W1C	<p>调优错误</p> <p>当调优电路中检测到不可恢复的错误时设置该位, 但在调优过程中除外(在调优过程中发生的错误由主机控制 2 寄存器中的采样时钟选择表示)。通过检测调优错误, 主机驱动程序需要终止执行命令并执行调优。要复位调优电路, 在执行调优程序之前, 采样时钟选择 (Sampling Clock Select) 被设置为 0。调优错误优先级高于数据传输过程中产生的其他错误中断。通过检测调优错误, 主机驱动程序必须丢弃当前读/写命令传输的数据, 并在主机控制器从调优电路错误中检索后重试数据传输。适用于 SD/eMMC 模式。</p> <p>UHS-II 模式下, 该位是无关位。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 <p>复位值: 0x0</p> <p>存在: 一直</p> <p>易失: 是</p>
9	ADMA_ERR	R/W1C	<p>ADMA 错误</p> <p>基于 ADMA 的数据传输过程中, 当主机控制器检测到错误时, 设置此位。该错误可能是由于以下原因:</p> <ul style="list-style-type: none"> ■ 从系统总线(Master I/F)收到错误响应 ■ 无效的 ADMA3, ADMA2 描述符 ■ CQE 任务或传输描述符无效 <p>当发生错误时, ADMA 的状态被保存在 ADMA 错误状态寄存器中。</p>

位段	名称	类型	说明
			<p>在 eMMC CQE 模式下：</p> <p>当主机控制器在 ST_FDS 状态下检测到无效的描述符数据 (Valid=0) 时，产生这个中断。ADMA Error Status 中的 ADMA Error State 表示在 ST_FDS 状态下发生了错误。主机驱动程序可能会发现错误描述符处没有设置有效位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE)：无错误 ■ 0x1 (TRUE)：有错误 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
8	AUTO_CMD_ERR	R/W1C	<p>自动 CMD 错误</p> <p>此错误状态被 Auto CMD12 和 Auto CMD23 在 SD/eMMC 模式下使用。当检测到自动 CMD 错误状态寄存器中 D00 到 D05 的任何位从 0 更改为 1 时，该位被设置。D07 对 Auto CMD12 有效。自动 CMD 错误状态寄存器在此位设置为 1 时有效，可以通过清除此位来清除。</p> <p>UHS-II 模式下，该位是无关位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE)：无错误 ■ 0x1 (TRUE)：有错误 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
7	CUR_LMT_ERR	R/W1C	<p>电流限制错误</p> <p>通过在电源控制寄存器中设置 SD 总线电源位，要求主控制器为 SD 总线供电。如果主机控制器支持电流限制功能，可以通过切断卡的电源来保护它不受非法卡的影响，在这种情况下，该位显示故障状态。该位的读数为 1 时表示由于某些故障，主机控制器没有给 SD 卡供电。该位的读数为 0 表示主机控制器正在供电，没有发生错误。主控制器可能需要一些采样时间来检测电流限制。DWC_mshc 主机控制器不支持此功能，该位总是设置为 0。</p> <p>值：</p>

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 断电 复位值: 0x0 存在: 一直 易失: 是
6	DATA_END_BIT_ERR	R/W1C	数据结束位错误 该错误在 SD/eMMC 模式中发生, 它可能发生在使用 DAT 线读取数据时结束位位置检测到 0 时, 也可能发生在 CRC 状态的结束位位置检测到 0。 UHS-II 模式下, 该位是无关位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 复位值: 0x0 存在: 一直 易失: 是
5	DATA_CRC_ERR	R/W1C	数据 CRC 错误 该错误发生在 SD/eMMC 模式下, 可能在以下三种情况检测到该错误。使用 DAT 线传输读取数据检测到 CRC 错误, 检测写 CRC (Write CRC) 状态的值不是 010 或在写 CRC 状态超时。 UHS-II 模式下, 该位是无关位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 复位值: 0x0 存在: 一直 易失: 是
4	DATA_TOUT_ERR	R/W1C	数据超时错误 当检测到以下超时条件之一时, 该位在 SD/eMMC 模式下设置: <ul style="list-style-type: none"> ■ R1b、R5b 类型繁忙超时

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 写 CRC 状态后繁忙超时 ■ 写 CRC 状态超时 ■ 读取数据超时 UHS-II 模式下, 该位是无关位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 超时 复位值: 0x0 存在: 一直 易失: 是
3	CMD_IDX_ERR	R/W1C	命令索引错误 如果在 SD/eMMC 模式下的命令响应中出现命令索引错误, 则设置该位。 UHS-II 模式下, 该位是无关位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 复位值: 0x0 存在: 一直 易失: 是
2	CMD_END_BIT_ERR	R/W1C	命令结束位错误 在 SD/eMMC 模式下, 当检测到命令响应结束位为 0 时设置。 UHS-II 模式下, 该位是无关位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 结束位错误产生 复位值: 0x0 存在: 一直 易失: 是
1	CMD_CRC_ERR	R/W1C	命令 CRC 错误 在 SD/eMMC 模式下, 有两种情况会产生命令 CRC 错误。

位段	名称	类型	说明
			<ul style="list-style-type: none"> 如果响应返回且命令超时错误设置为 0（表示无超时），则在命令响应中检测到 CRC 错误时，设置该位为 1。 当命令发出时，主机控制器通过监控 CMD 线来检测 CMD 线冲突。如果主机控制器将 CMD 线驱动到级别 1，但是在下一个 SD 时钟边缘检测到 CMD 线上的级别为 0，则主机控制器终止命令(停止驱动 CMD 线)并将该位设置为 1。命令超时错误也设置为 1，以区分 CMD 线冲突。 <p>UHS-II 模式下，该位是无关位。</p> <p>值：</p> <ul style="list-style-type: none"> 0x0 (FALSE): 无错误 0x1 (TRUE): CRC 错误产生 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
0	CMD_TOUT_ERR	R/W1C	<p>命令超时错误</p> <p>在 SD/eMMC 模式下，只有从命令结束位算起的 64 个 SD 时钟周期内没有响应时才需要设置该位。如果主机控制器检测到 CMD 线冲突和命令 CRC 错误位，则该位设置为 1，无需等待 64 个 SD/eMMC 卡时钟周期。UHS-II 模式下，该位是无关位。</p> <p>值：</p> <ul style="list-style-type: none"> 0x0 (FALSE): 无错误 0x1 (TRUE): 超时 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>

3.6.1.22 NORMAL_INT_STAT_EN_R

- 寄存器名：正常中断状态使能寄存器
- 描述：当 NORMAL_INT_STAT_R 设置为 1 时，该寄存器使能正常中断状态寄存器（NORMAL_INT_STAT_R）的中断状态。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x34

图表 3-65 NORMAL_INT_STAT_EN_R 寄存器字段说明

位段	名称	类型	说明
15	RSVD_15	R	NORMAL_INT_STAT_EN_R 寄存器的这一部分是保留的。它总是返回 0。 复位值: 0x0 存在: 总是
14	CQE_EVENT_STAT_EN	R/W	CQE 事件状态使能 值: <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
13	FX_EVENT_STAT_EN	R/W	FX 事件状态启用 从版本 4.10 开始增加该位。 值: <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
12	RE_TUNE_EVENT_STAT_EN	R/W	重调事件 (仅限 UHS-I) 状态使能 值: <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
11	INT_C_STAT_EN	R/W	INT_C(嵌入式)状态启用 如果该位设置为 0, 则主机控制器清除对系统的中断请求。主机驱动程序可以在处理 INT_C 之前清除该位, 也可以在所有对 INT_C 管脚的中断请求被清除后再次设置该位, 以防止意外中断。 值: <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽

位段	名称	类型	说明
			■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
10	INT_B_STAT_EN	R/W	如果该位设置为 0, 则主机控制器清除对系统的中断请求。主机驱动程序可以在服务 INT_B 之前清除该位, 也可以在所有对 INT_B 管脚的中断请求被清除后再次设置该位, 以防止意外中断。 值: ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
9	INT_A_STAT_EN	R/W	INT_A (嵌入式) 状态启用 如果该位设置为 0, 则主机控制器清除对系统的中断请求。主机驱动程序可以在处理 INT_A 之前清除该位, 也可以在所有对 INT_A 管脚的中断请求被清除后再次设置该位, 以防止意外中断。 值: ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
8	CARD_INTERRUPT_STAT_EN	R/W	卡中断状态使能 如果该位设置为 0, 则主机控制器清除对系统的中断请求。当清除该位时卡中断检测停止, 当该位设置为 1 时重新启动。主机驱动程序可能会在处理卡中断之前清除卡中断状态使能 (Card Interrupt Status Enable), 并且在清除所有来自卡的中断请求后再次设置该位, 以防止意外中断。 通过设置该位为 0, 中断输入必须被实现屏蔽, 以使中断输入不受任何状态(例如, 浮动)的外部信号的影响。 值: ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能

位段	名称	类型	说明
			复位值: 0x0 存在: 一直
7	CARD_REMOVAL_STAT_EN	R/W	卡移除状态使能 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
6	CARD_INSERTION_STAT_EN	R/W	卡插入状态使能 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 总是
5	BUF_RD_READY_STAT_EN	R/W	缓冲区读准备状态使能 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
4	BUF_WR_READY_STAT_EN	R/W	缓冲区读写准备状态使能 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
3	DMA_INTERRUPT_STAT_EN	R/W	DMA 中断状态使能 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能

位段	名称	类型	说明
			复位值：0x0 存在：一直
2	BGAP_EVENT_STAT_EN	R/W	块间隙事件状态使能 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
1	XFER_COMPLETE_STAT_EN	R/W	传输完成状态使能 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
0	CMD_COMPLETE_STAT_EN	R/W	命令完成状态使能 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直

3.6.1.23 ERROR_INT_STAT_EN_R

- 寄存器名：错误中断状态使能寄存器
- 描述：当 ERROR_INT_STAT_EN_R 设置为 1 时，该寄存器为错误中断状态寄存器（ERROR_INT_STAT_R）设置中断状态。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x36

图表 3-66 ERROR_INT_STAT_EN_R 寄存器字段说明

位段	名称	类型	说明
15	VENDOR_ERR_STAT_EN3	R/W	错误中断状态使能寄存器的第 15 位被保留。

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
14	VENDOR_ERR_STAT_EN2	R/W	错误中断状态使能寄存器的第 14 位被保留。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
13	VENDOR_ERR_STAT_EN1	R/W	错误中断状态使能寄存器的第 13 位被保留。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
12	BOOT_ACK_ERR_STAT_EN	R/W	引导确认错误 (仅限 eMMC 模式) 设置此位为 1 启动错误中断状态寄存器 (ERROR_INT_STAT_R) 中的引导确认错误 (Boot Acknowledgment Error) 设置。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
11	RESP_ERR_STAT_EN	R/W	响应错误状态使能 (仅限 SD 模式) 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0

位段	名称	类型	说明
			存在：一直
10	TUNING_ERR_STAT_EN	R/W	调优错误状态使能（仅限 UHS-I 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
9	ADMA_ERR_STAT_EN	R/W	ADMA 错误状态使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
8	AUTO_CMD_ERR_STAT_EN	R/W	自动 CMD 错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
7	CUR_LMT_ERR_STAT_EN	R/W	电流限制错误状态使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
6	DATA_END_BIT_ERR_STAT_EN	R/W	数据结束位错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0

位段	名称	类型	说明
			存在：一直
5	DATA_CRC_ERR_STAT_EN	R/W	数据 CRC 错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
4	DATA_TOUT_ERR_STAT_EN	R/W	数据超时错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
3	CMD_IDX_ERR_STAT_EN	R/W	命令指示索引错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
2	CMD_END_BIT_ERR_STAT_EN	R/W	命令结束位错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
1	CMD_CRC_ERR_STAT_EN	R/W	命令 CRC 错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0

位段	名称	类型	说明
			存在：一直
0	CMD_TOUT_ERR_STAT_EN	R/W	命令超时错误状态使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽 0x1 (TRUE): 使能 复位值：0x0 存在：一直

3.6.1.24 NORMAL_INT_SIGNAL_EN_R

- 寄存器名：正常中断信号使能寄存器
- 描述：这个寄存器用于选择指示主机系统作为中断的中断状态。所有这些状态位共享同一个 1 位中断线。将这些位中的任何一位设置为 1，即可启用中断生成。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x38

图表 3-67 NORMAL_INT_SIGNAL_EN_R 寄存器字段说明

位段	名称	类型	说明
15	RSVD_15	R	NORMAL_INT_STAT_R 寄存器的这一位是保留的。它总是返回 0。 复位值：0x0 存在：一直
14	CQE_EVENT_SIGNAL_EN	R/W	命令队列引擎事件信号使能 值： <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽 0x1 (TRUE): 使能 复位值：0x0 存在：一直
13	FX_EVENT_SIGNAL_EN	R/W	FX 事件信号使能 值： <ul style="list-style-type: none"> 0x0 (FALSE): 屏蔽 0x1 (TRUE): 使能

位段	名称	类型	说明
			复位值：0x0 存在：一直
12	RE_TUNE_EVENT_SIGNAL_EN	R/W	重调事件（仅限 UHS-I）信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
11	INT_C_SIGNAL_EN	R/W	INT_C（嵌入式）信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
10	INT_B_SIGNAL_EN	R/W	INT_B（嵌入式）信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
9	INT_A_SIGNAL_EN	R/W	INT_A（嵌入式）信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
8	CARD_INTERRUPT_SIGNAL_EN	R/W	卡中断信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能

位段	名称	类型	说明
			复位值：0x0 存在：一直
7	CARD_REMOVAL_SIGNAL_EN	R/W	卡移除信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
6	CARD_INSERTION_SIGNAL_EN	R/W	卡插入信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
5	BUF_RD_READY_SIGNAL_EN	R/W	缓冲区读就绪信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
4	BUF_WR_READY_SIGNAL_EN	R/W	缓冲区写就绪信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
3	DMA_INTERRUPT_SIGNAL_EN	R/W	DMA 中断信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能

位段	名称	类型	说明
			复位值：0x0 存在：一直
2	BGAP_EVENT_SIGNAL_EN	R/W	块间隙事件信号使能 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
1	XFER_COMPLETE_SIGNAL_EN	R/W	传输完成信号使能 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
0	CMD_COMPLETE_SIGNAL_EN	R/W	命令完成信号使能 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直

3.6.1.25 ERROR_INT_SIGNAL_EN_R

- 寄存器名：错误中断信号使能寄存器
- 描述：这个寄存器用于选择指示给主机系统作为中断的中断状态。所有这些状态位共享同一个 1 位中断线。将这些位中的任何一位设置为 1，就可以产生中断。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x3a

图表 3-68 ERROR_INT_SIGNAL_EN_R 寄存器字段说明

位段	名称	类型	说明
15	VENDOR_ERR_SIGNAL_EN3	R/W	错误中断信号使能的第 16 位被保留。

位段	名称	类型	说明
			值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
14	VENDOR_ERR_SIGNAL_EN2	R/W	错误中断信号使能的第 15 位被保留。 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
13	VENDOR_ERR_SIGNAL_EN1	R/W	错误中断信号使能的第 14 位被保留。 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
12	BOOT_ACK_ERR_SIGNAL_EN	R/W	引导确认错误 (仅限 eMMC 模式) 当错误中断状态 (Error Interrupt Status) 寄存器中设置了引导确认错误 (Boot Acknowledgement Error) 时, 设置该位为 1, 可以产生中断信号。 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直
11	RESP_ERR_SIGNAL_EN	R/W	响应错误信号使能 (仅限 SD 模式) 值： ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值: 0x0

位段	名称	类型	说明
			存在：一直
10	TUNING_ERR_SIGNAL_EN	R/W	调优错误信号使能（仅限 UHS-I 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
9	ADMA_ERR_SIGNAL_EN	R/W	ADMA 错误信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
8	AUTO_CMD_ERR_SIGNAL_EN	R/W	自动 CMD 错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
7	CUR_LMT_ERR_SIGNAL_EN	R/W	电流限制错误信号使能 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
6	DATA_END_BIT_ERR_SIGNAL_EN	R/W	数据结束位错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0

位段	名称	类型	说明
			存在：一直
5	DATA_CRC_ERR_SIGNAL_EN	R/W	数据 CRC 错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
4	DATA_TOUT_ERR_SIGNAL_EN	R/W	数据超时错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
3	CMD_IDX_ERR_SIGNAL_EN	R/W	命令索引错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 无错误 ■ 0x1 (TRUE): 有错误 复位值：0x0 存在：一直
2	CMD_END_BIT_ERR_SIGNAL_EN	R/W	命令结束位错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直
1	CMD_CRC_ERR_SIGNAL_EN	R/W	命令 CRC 错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0

位段	名称	类型	说明
			存在：一直
0	CMD_TOUT_ERR_SIGNAL_EN	R/W	命令超时错误信号使能（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 屏蔽 ■ 0x1 (TRUE): 使能 复位值：0x0 存在：一直

3.6.1.26 AUTO_CMD_STAT_R

- 寄存器名：自动 CMD 状态寄存器
- 描述：该寄存器用于指示自动 CMD12 的 CMD12 响应错误和自动 CMD23 的 CMD23 响应错误。主机驱动程序可以确定在这个寄存器中可能发生的自动 CMD12/CMD23 错误类型。自动 CMD23 错误显示在 04-01 位。这个寄存器只有在设置了自动 CMD 错误（Auto CMD Error）时才有效。这个寄存器适用于 SD/eMMC 模式。
- 大小：16 位
- 偏移量：0x3c

图表 3-69 AUTO_CMD_STAT_R 寄存器字段说明

位段	名称	类型	说明
15:8	RSVD_15_8	R	AUTO_CMD_STAT_R 寄存器的这些位是保留位。它们总是返回 0。 复位值：0x0 存在：一直 易失：是
7	CMD_NOT_ISSUED_AUTO_CMD12	R	命令不被自动 CMD12 发出错误 如果该位设置为 1, CMD_wo_DAT 将不会执行，因为在该寄存器中有自动 CMD12 错误（D04-D01）。 当自动 CMD23 生成自动 CMD 错误，该位设置为 0。 值： <ul style="list-style-type: none"> ■ 0x1 (TRUE): 没有发出 ■ 0x0 (FALSE): 没有错误 复位值：0x0

位段	名称	类型	说明
			存在：一直 易失：是
6	RSVD_6	R	AUTO_CMD_STAR_R 寄存器的这一部分是保留的。它总是返回 0。 复位值：0x0 存在：一直 易失：是
5	AUTO_CMD_RESP_ERR	R	自动 CMD 响应错误 传输模式寄存器中的响应错误检查使能（Response Error Check Enable）设置为 1 时，设置该位。在 R1 响应中检测到 Auto CMD12 或 CMD13 错误。如果 D00 到 D04 之间的任何位设置为 1，则该状态被忽略。 值： <ul style="list-style-type: none"> ■ 0x1 (TRUE): 错误 ■ 0x0 (FALSE): 无错误 复位值：0x0 存在：一直 易失：是
4	AUTO_CMD_IDX_ERR	R	自动 CMD 索引错误 如果在对命令的响应中发生命令索引错误，则设置此位。 值： <ul style="list-style-type: none"> ■ 0x1 (TRUE): 有错误 ■ 0x0 (FALSE): 无错误 复位值：0x0 存在：一直 易失：是
3	AUTO_CMD_EBIT_ERR	R	自动 CMD 结束位错误 当检测到命令响应的结束位为 0 时，设置该位。 值： <ul style="list-style-type: none"> ■ 0x1 (TRUE): 结束位错误产生 ■ 0x0 (FALSE): 无错误

位段	名称	类型	说明
			复位值: 0x0 存在: 一直 易失: 是
2	AUTO_CMD_CRC_ERR	R	自动 CMD CRC 错误 该位在命令响应中检测到 CRC 错误时设置。 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): CRC 错误产生 ■ 0x0 (FALSE): 无错误 复位值: 0x0 存在: 一直 易失: 是
1	AUTO_CMD_TOUT_ERR	R	自动 CMD 超时错误 如果从命令的结束位算起 64 个 SDCLK 周期没有返回响应, 则设置此位。 如果该位设置为 1, 则错误状态位 (D04-D01) 无意义 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 超时 ■ 0x0 (FALSE): 无错误 复位值: 0x0 存在: 一直 易失: 是
0	AUTO_CMD12_NOT_EXEC	R	自动 CMD12 未执行 如果由于命令错误导致多个内存块数据传输没有开始, 则不设置此位, 因为没有必要发出自动 CMD12。将该位设置为 1 意味着由于某些错误, 主机控制器不能发出自动 CMD12 用以停止多个内存块数据传输。如果设置为 1, 则错误状态位 (D04-D01) 无意义。 当自动 CMD23 生成自动 CMD 错误 (Auto CMD Error) 时, 该位设置为 0。 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 未执行 ■ 0x0 (FALSE): 执行

位段	名称	类型	说明
			复位值: 0x0 存在: 一直 易失: 是

3.6.1.27 HOST_CTRL2_R

- 寄存器名: 主机控制 2 寄存器
- 描述: 这个寄存器用来控制主控制器的操作。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小: 16 位
- 偏移量: 0x3e

图表 3-70 HOST_CTRL2_R 寄存器字段说明

位段	名称	类型	说明
15	PRESET_VAL_ENABLE	R/W	预设值启用 该位允许 SDCLK 频率和驱动器强度预置值寄存器的自动选择。当设置预置值使能 (Preset Value Enable) 时, 控制器会进行 SDCLK 频率生成 (频率选择和时钟生成器选择) 和驱动器强度选择。这些值是根据所选的速度模式从预设值寄存器中选择的。 注意: 有关更多信息, 请参阅 DWC_mshc 数据手册中关于预设寄存器的常见问题。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): SDCLK 和驱动器强度由主机驱动程序选择 ■ 0x1 (TRUE): 预设值自动选择启用 复位值: 0x0 存在: 一直
14	ASYNC_INT_ENABLE	R/W	异步中断使能 如果一张卡支持异步中断, 并且在能力寄存器中异步中断支持被设置为 1, 则设置此位。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 禁用 ■ 0x1 (TRUE): 使能 复位值: 0x0 存在: 一直

位段	名称	类型	说明
13	ADDRESSING	R/W	<p>64 位寻址</p> <p>当主机版本 4 启用（Host Version 4 Enable）设置为 1 时，此位有效。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 32 位寻址 ■ 0x1 (TRUE): 64 位寻址 <p>复位值：0x0</p> <p>存在：一直</p>
12	HOST_VER4_ENABLE	R/W	<p>主机版本 4 启用</p> <p>该位选择版本 3.00 兼容模式或版本 4 模式。</p> <p>在主机版本 4 模式下，修改了以下字段的功能：</p> <ul style="list-style-type: none"> ■ DMA 地址：SDMA 使用 ADMA 系统地址（05Fh-058h）代替 SDMA 系统地址寄存器（003h-000h） ■ ADMA2/ADMA3 选择：在主机控制 1 寄存器中通过 DMA 选择来选中 ADMA3 ■ 64 位 ADMA 描述符大小：当 64 位寻址设置为 1 时，使用 128 位描述符，而不是 96 位描述符 ■ 2 位/64 位系统寻址选择：32 位或 64 位系统寻址由该寄存器中的 64 位寻址选择 ■ 32 位块计数：SDMA 系统地址寄存器（003h-000h）被修改为 32 位块计数寄存器 <p>注意：建议在主机版本低于 4 模式下（Host Version 4 Enable = 0）运行时，不要对 ADMA3 综合描述符地址寄存器、UHS-II 寄存器和命令队列寄存器（如果适用）进行编程。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (FALSE): 版本 3.00 兼容模式 ■ 0x1 (TRUE): 版本 4 模式 <p>复位值：0x0</p> <p>存在：一直</p>
11	CMD23_ENABLE	R/W	<p>CMD23 使能</p> <p>如果卡支持 CMD23，该位设置为 1。该位用于选择自动 CMD23 或自动 CMD12 进行 ADMA3 数据传输。</p>

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 自动 CMD23 被禁用。 ■ 0x1 (TRUE): 自动 CMD23 被启用。 复位值：0x0 存在：一直
10	ADMA2_LEN_MODE	R/W	ADMA2 长度模式 该位选择 ADMA2 长度模式为 16 位或 26 位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 16 位数据长度模式 ■ 0x1 (TRUE): 26 位数据长度模式 复位值：0x0 存在：一直
9	RSVD_9	R	HOST_CTRL2_R 寄存器的这一位是保留的。它总是返回 0。 复位值：0x0 存在：一直
8	UHS2_IF_ENABLE	R/W	UHS-II 接口启用 该位用于启用 UHS-II 接口。该值反映在 uhs2_if_en 管脚上。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): SD/eMMC 接口启用 ■ 0x1 (TRUE): UHS-II 接口启用 复位值：0x0 存在：一直
7	SAMPLE_CLK_SEL	R/W	采样时钟选择 该位被主控制器用来在 SD/eMMC 模式下选择采样时钟来接收 CMD 和 DAT。该位由调优过程设置，在调优完成后（执行调优“Execute Tuning”被清除时）有效。将该位设置为 1 表示调优成功完成；将该位设置为 0 表示调优失败。该值反映在 sample_cclk_sel 管脚上。在 UHS-II 模式下，该位为无关位。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 固定时钟用于采样数据

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x1 (TRUE): 调优时钟用于采样数据 复位值: 0x0 存在: 一直 易失: 是
6	EXEC_TUNING	R/W	执行调优 该位设置为 1, 表示在 UHSI/eMMC 速度模式下启动调优过程, 调优过程完成后, 该位自动清除。在 UHS-II 模式下, 该位为无关位。 值: <ul style="list-style-type: none"> 0x0 (FALSE): 未调优或已调优 0x1 (TRUE): 执行调优 复位值: 0x0 存在: 总是 易失: 是
5:4	DRV_STRENGTH_SEL	R/W	驱动器强度选择 该位用于选择 1.8V 信令 UHS-I/eMMC 速率模式下的主控制器输出驱动。该位取决于预设值使能 (Preset Value Enable) 的设置。该值反映在 uhs1_drv_sth 管脚上。在 UHS-II 模式下, 该位为无关位。 值: <ul style="list-style-type: none"> 0x0 (TYPEB): 驱动器类型 B 被选择 0x1 (TYPEA): 驱动器类型 A 被选择 0x2 (YPEC): 驱动器类型 C 被选择 0x3 (TYPED): 驱动器类型 D 被选择 复位值: 0x0 存在: 一直 易失: 是
3	SIGNALING_EN	R/W	1.8V 信号使能 该位控制 UHS-I/eMMC 速度模式的 I/O 单元电压调节器。将该位从 0 设置为 1, 开始将信号电压从 3.3V 改变为 1.8V。如果切换到 1.8 信令失败, 主机控制器清除该位。该值反映在 uhs1_svwolt_en 管脚上。UHS-II 模式下, 该位应被设置

位段	名称	类型	说明
			为 0 注意：所有 UHS-I 速度模式 (SDR12/SDR25/SDR50/SDR104/DDR50) 都必须设置此位。 值： <ul style="list-style-type: none"> ■ 0x0 (V_3_3): 3.3V 信令 ■ 0x1 (V_1_8): 1.8V 信令 复位值：0x0 存在：一直 易失：是
2:0	UHS_MODE_SEL	R/W	UHS 模式/eMMC 速度模式选择 这些位用于在 SD 操作模式下选择 UHS 模式。在 eMMC 模式下，这些位用于选择 eMMC 速度模式 UHS 模式（仅限 SD/UHS-II 模式） <ul style="list-style-type: none"> ■ 0x0: SDR12 ■ 0x1: SDR25 ■ 0x2: SDR50 ■ 0x3: SDR104 ■ 0x4: DDR50 ■ 0x5: 保留 ■ 0x6: 保留 ■ 0x7: UHS-II eMMC 速度模式（仅限 eMMC 模式） <ul style="list-style-type: none"> ■ 0x0: 旧有 ■ 0x1: 高速 SDR ■ 0x2: 保留 ■ 0x3: HS200 ■ 0x4: 高速 DDR ■ 0x5: 保留 ■ 0x6: 保留 ■ 0x7: HS400 值：

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x0 (SDR12): SDR12/旧有 0x1 (SDR25): SDR25/高速 SDR 0x2 (SDR50): SDR50 0x3 (SDR104): SDR104/HS200 0x4 (DDR50): DDR50/高速 DDR 0x5 (RSVD5): 保留 0x6 (RSVD6): 保留 0x7 (UHS2): UHS-II/HS400 复位值: 0x0 存在: 一直

3.6.1.28 CAPABILITIES1_R

- 寄存器名: 能力 1 寄存器- 0 到 31
- 描述: 这个寄存器为主机驱动程序提供专门用于主机控制器实现的信息。在上电初始化期间, 主控制器可以将这些值作为固定值或从闪存加载这些值。能力寄存器被分隔成两个 32 位寄存器: CAPABILITIES1_R 和 CAPABILITIES2_R。CAPABILITIES1_R 寄存器是能力寄存器的中下部分。
- 大小: 32 位
- 偏移量: 0x40

图表 3-71 CAPABILITIES1_R 寄存器字段说明

位段	名称	类型	说明
31:30	SLOT_TYPE_R	R	槽型 这些位表示特定主机系统对插槽的使用情况。 值: <ul style="list-style-type: none"> 0x0 (REMOVABLE_SLOT): 可拆卸卡槽 0x1 (EMBEDDED_SLOT): 一个设备的内嵌槽位 0x2 (SHARED_SLOT): 共享总线槽位 (SD 模式) 0x3 (UHS2_EMBEDDED_SLOT): UHS-II 多个嵌入式设备 复位值: 0x0 存在: 一直
29	ASYNC_INT_SUPPORT	R	异步中断支持 (仅限 SD 模式) 值:

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持异步中断。 ■ 0x1 (TRUE): 支持异步中断。 复位值: 0x0 存在: 一直
28	SYS_ADDR_64_V3	R	V3 的 64 位系统地址支持 该位设置主机控制器以支持 V3 模式的 64 位系统寻址。 SDMA 不能在版本 3 模式的 64 位寻址中使用。 设置为 1 时, 通过设置主机版本 4 使能 (HOST_VER4_ENABLE = 0)和 DMA 选择 (DMA_SEL = 11b) 可以启用使用 96 位 SDMA 描述符的 64 位 ADMA2。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 V3 的 64 位系统地址。 ■ 0x1 (TRUE): 支持 V3 的 64 位系统地址。 复位值: 0x0 存在: 一直
27	SYS_ADDR_64_V4	R	V4 的 64 位系统地址支持 该位设置主机控制器以支持 V4 模式的 64 位系统寻址。当该位设置为 1 时, 必须使用全部或部分 64 位地址来解码主机控制器寄存器, 以便主机控制器寄存器可以放置在系统内存区域之上。无论在主机控制 2 中设置为 64 位地址与否, 主机控制器寄存器的 64 位地址解码都是有效的。 如果该位设置为 1, 则版本 4 的 64 位 DMA 寻址是通过设置主机版本 4 使能 (HOST_VER4_ENABLE = 1)和在主机控制 2 寄存器中设置 64 位寻址(ADDRESSING = 1)来启用的。SDMA 可以被使用, ADMA2 使用 128 位描述符。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 V4 的 64 位系统地址。 ■ 0x1 (TRUE): 支持 V4 的 64 位系统地址。 复位值: 0x0 存在: 一直
26	VOLT_18	R	支持 1.8V 电压 值:

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 1.8V。 ■ 0x1 (TRUE): 支持 1.8V。 复位值: DWC_MSHC_VOLT18_VDD1_SUPPORT 存在: 一直
25	VOLT_30	R	支持 SD 3.0V 或嵌入式 1.2V 电压 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 SD 3.0V 或嵌入式 1.2V 电压 ■ 0x1 (TRUE): 支持 SD 3.0V 或嵌入式电压 复位值: DWC_MSHC_VOLT30_VDD1_SUPPORT 存在: 一直
24	VOLT_33	R	支持 3.3V 电压 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 3.3V 电压 ■ 0x1 (TRUE): 支持 3.3V 电压 复位值: DWC_MSHC_VOLT33_VDD1_SUPPORT 存在: 一直
23	SUS_RES_SUPPORT	R	支持暂停/恢复 该位表示主机控制器是否支持暂停/恢复功能。如果该位为 0, 则主机驱动程序不会发出暂停或恢复命令, 因为不支持暂停和恢复机制。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 ■ 0x1 (TRUE): 支持 复位值: 0x0 存在: 一直
22	SDMA_SUPPORT	R	支持 SDMA 该位表示主机控制器是否能够使用 SDMA 在系统内存和主机控制器之间直接传输数据。

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> 0x0 (FALSE): 不支持 SDMA。 0x1 (TRUE): 支持 SDMA。 复位值: 0x1 存在: 一直
21	HIGH_SPEED_SUPPORT	R	支持高速 该位表示主机控制器和主机系统是否支持高速模式，以及是否可以提供 25MHz 到 50MHz 的 SD 时钟频率。 值： <ul style="list-style-type: none"> 0x0 (FALSE): 不支持高速 0x1 (TRUE): 支持高速 复位值: 0x1 存在: 一直
20	RSVD_20	R	CAPABILITIES1_R 的这一位是保留的。它总是返回 0。 复位值: 0x0 存在: 一直
19	ADMA2_SUPPORT	R	支持 ADMA2 该位表示主控制器是否能够使用 ADMA2。 值： <ul style="list-style-type: none"> 0x0 (FALSE): 不支持 ADMA2 0x1 (TRUE): 支持 ADMA2 复位值: 0x1 存在: 一直
18	Embedded_8_BIT	R	嵌入式设备的 8 位支持 该位表示主控制器是否能够使用 8 位总线宽度模式。当插槽类型设置为 10b 时，该位无效。 值： <ul style="list-style-type: none"> 0x0 (FALSE): 不支持 8 位总线宽度 0x1 (TRUE): 支持 8 位总线宽度 复位值: 0x1 存在: 一直

位段	名称	类型	说明
17:16	MAX_BLK_LEN	R	<p>最大的块长度</p> <p>该位表示主机驱动程序可以对主机控制器中的缓冲区进行读写的最大块的大小。缓冲区传输这个大小的块无需等待周期。无论这个位的值是什么，SD 内存传输块的长度总是 512 字节。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (ZERO): 512 字节 ■ 0x1 (ONE): 1024 字节 ■ 0x2 (TWO): 2048 字节 ■ 0x3 (THREE): 保留 <p>复位值：0x1</p> <p>存在：一直</p>
15:8	BASE_CLK_FREQ	R	<ul style="list-style-type: none"> ■ SD 时钟的 e Clock 频率 <p>这些位表示 SD 时钟的基础（最大）时钟频率。这些位的定义取决于主机控制器版本。</p> <p>6 位基础时钟频率：主机控制器版本 1.00 和 2.00 支持该模式。高两位是无效的，总是 0。单位为 1MHz。支持的时钟频率范围为 10MHz 到 63MHz。</p> <p>0x00: 通过其他方法获取信息</p> <p>0x01: 1MHz</p> <p>0x02: 2MHz</p> <p>.....</p> <p>0x3F: 63MHz</p> <p>0x40-0xFF: 不支持</p> <ul style="list-style-type: none"> ■ 8 位基础时钟频率：主机控制器版本 3.00 支持该模式。单位为 1MHz。支持的时钟频率范围为 10MHz 到 255MHz。 <p>0x00: 通过其他方法获取信息</p> <p>0x01: 1MHz</p> <p>0x02: 2MHz</p> <p>.....</p> <p>0xFF: 255MHz</p> <p>频率为 16.5MHz 时，由于主机驱动使用此值计算时钟分压器</p>

位段	名称	类型	说明
			<p>值，且它不超过 SD 时钟频率上限，因此设置较大的值为 0001001b (17MHz)。如果这些位都是 0，主机系统必须使用不同的方法获取信息。</p> <p>复位值：0xC8</p> <p>存在：一直</p>
7	TOUT_CLK_UNIT	R	<p>超时时钟单元</p> <p>该位显示用于检测数据超时错误的基本时钟频率的单位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (KHZ): KHz ■ 0x1 (MHZ): MHz <p>复位值：0x1</p> <p>存在：一直</p>
6	RSVD_6	R	<p>CAPABILITIES1_R 寄存器的这一位是保留的。它总是返回 0。</p> <p>复位值：0x0</p> <p>存在：一直</p>
5:0	TOUT_CLK_FREQ	R	<p>超时时钟频率</p> <p>该位显示用于检测数据超时错误的基础时钟频率。超时时钟单位定义了超时时钟频率的单位。可以是 KHz 或 MHz。</p> <ul style="list-style-type: none"> ■ 0x00: 通过其他方法获取信息 ■ 0x01: 1KHz/1MHz ■ 0x02: 2KHz/2MHz ■ 0x03: 3KHz/3MHz ■ ... ■ 0x3F: 63KHz/63MHz <p>复位值：0x1</p> <p>存在：一直</p>

3.6.1.29 CAPABILITIES2_R

- 寄存器名：能力寄存器 - 32 到 63
- 描述：这个寄存器为主机驱动程序提供了专门用于主机控制器实现的信息。在上电初始化期间，主控制器可以将这些值作为固定值或从闪存中加载的值来实现。能力寄存器被分隔成两个 32 位寄存

器，即 CAPABILITIES1_R 和 CAPABILITIES2_R。CAPABILITIES2_R 寄存器位于 Capabilities 寄存器的上部。

- 大小：32 位
- 偏移量：0x44

图表 3-72 CAPABILITIES2_R 寄存器字段说明

位段	名称	类型	说明
31:30	RSVD_62_63	R	CAPABILITIES2_R 寄存器的这些位(RSVD_62_63)是保留位。它们总是返回 0。 复位值：0x0 存在：一直
29	RSVD_61	R	CAPABILITIES2_R 寄存器的这个位(RSVD_61)是保留的。它总是返回 0。 复位值：0x0 存在：一直
28	VDD2_18V_SUPPORT	R	支持 1.8V VDD2 该位表示对主机系统 VDD2 的支持。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 1.8V VDD2。 ■ 0x1 (TRUE): 支持 1.8V VDD2。 复位值：0x0 存在：一直
27	ADMA3_SUPPORT	R	支持 ADMA3 该位表示主控制器是否能够使用 ADMA3。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持 ADMA3。 ■ 0x1 (TRUE): 支持 ADMA3。 复位值：0x1 存在：一直
26:24	RSVD_56_58	R	CAPABILITIES2_R 寄存器的这些位(RSVD_56_58)是保留的。它们总是返回 0。 复位值：0x0 存在：一直

位段	名称	类型	说明
23:16	CLK_MUL	R	时钟倍频器 这些位表示可编程时钟发生器的时钟倍频器。设置这些位为 0 意味着主机控制器不支持可编程时钟发生器。 <ul style="list-style-type: none"> ■ 0x0: 支持时钟倍频器 ■ 0x1: 时钟倍频器 M = 2 ■ 0x2: 时钟倍频器 M = 3 ■ ■ 0xFF: Clock Multiplier M = 256 复位值: 0x0 存在: 一直
15:14	RE_TUNING_MODES	R	重调模式 (仅限 UHS-I) 这些位选择重调的方法并限制最大数据长度。 值: <ul style="list-style-type: none"> ■ 0x0 (MODE1): 计时器 ■ 0x1 (MODE2): 定时器和重调请求 (不支持) ■ 0x2 (MODE3): 自动重调 (用于传输) ■ 0x3 (RSVD_MODE): 保留 复位值: 0x2 存在: 一直
13	USE_TUNING_SDR50	R	使用 SDR50 调优 (仅限 UHS-I) 值: <ul style="list-style-type: none"> ■ 0x0 (ZERO): SDR50 不需要调优。 ■ 0x1 (ONE): SDR50 需要调优。 复位值: 0x0 存在: 一直
12	RSVD_44	R	CAPABILITIES2_R 寄存器的这个位(RSVD_44)是保留的。它总是返回 0。 复位值: 0x0 存在: 一直
11:8	RETUNE_CNT	R	重调的计时器计数 (仅限 UHS-I) <ul style="list-style-type: none"> ■ 0x0: 重调计时器禁用

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x1: 1 秒 ■ 0x2: 2 秒 ■ 0x3: 4 秒 ■ ... ■ 0xB: 1024 秒 ■ 0xC: 保留 ■ 0xD: 保留 ■ 0xE: 保留 ■ 0xF: 从其他来源获取信息 复位值: 0x1 存在: 一直
7	RSVD_39	R	CAPABILITIES2_R 寄存器的这个位(RSVD_39)是保留的。它总是返回 0。 复位值: 0x0 存在: 一直
6	DRV_TYPED	R	支持驱动器类型 D (仅限 UHS-I) 该位表示支持 1.8 信令的驱动器类型 D。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持驱动器类型 D。 ■ 0x1 (TRUE): 支持驱动器类型 D。 复位值: 0x1 存在: 一直
5	DRV_TYPEC	R	支持驱动器类型 C (仅限 UHS-I) 该位表示支持 1.8 信令的驱动器类型 C。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 不支持驱动器类型 C。 ■ 0x1 (TRUE): 支持驱动器类型 C。 复位值: 0x1 存在: 一直
4	DRV_TYPEA	R	支持驱动器类型 A (仅限 UHS-I) 该位表示支持 1.8 信令的驱动器类型 A。

位段	名称	类型	说明
			值： ■ 0x0 (FALSE): 不支持驱动器类型 A。 ■ 0x1 (TRUE): 支持驱动器类型 A。 复位值: 0x1 存在: 一直
3	UHS2_SUPPORT	R	支持 UHS-II (仅限 UHS-II) 该位表示主机控制器是否支持 UHS-II。 值： ■ 0x0 (FALSE): 不支持 UHS-II。 ■ 0x1 (TRUE): 支持 UHS-II。 复位值: 0x0 存在: 一直
2	DDR50_SUPPORT	R	支持 DDR50 (仅限 UHS-I) 值： ■ 0x0 (FALSE): 不支持 DDR50。 ■ 0x1 (TRUE): 支持 DDR50。 复位值: 0x1 存在: 一直
1	SDR104_SUPPORT	R	支持 SDR104 (仅限 UHS-I) 这一位提示 SDR104 需要调优。 值： ■ 0x0 (FALSE): 不支持 SDR104。 ■ 0x1 (TRUE): 支持 SDR104。 复位值: 0x1 存在: 一直
0	SDR50_SUPPORT	R	支持 SDR50 (仅限 UHS-I) 该位表示支持 SDR50。13 位(USE_TUNING_SDR50)表示 SDR50 是否需要调优。 值： ■ 0x0 (FALSE): 不支持 SDR50。 ■ 0x1 (TRUE): 支持 SDR50。

位段	名称	类型	说明
			复位值：0x1 存在：一直

3.6.1.30 CURR_CAPABILITIES1_R

- 寄存器名：最大电流容量寄存器--0 到 31
- 描述：这个寄存器表示对于 VDD1，每个电压的最大电流容量。如果电压支持是在能力寄存器中设置的，这个值是有意义的。如果主机系统通过另一种方法提供此信息，则所有最大电流容量寄存器都设置为 0。
- 大小：32 位
- 偏移量：0x48

图表 3-73 CURR_CAPABILITIES1_R 寄存器字段说明

位段	名称	类型	说明
31:24	RSVD_31_24	R	CURR_CAPABILITIES1_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
23:16	MAX_CUR_18V	R	最大电流为 1.8V 该位指定卡的 1.8V VDD1 电源的最大电流。 <ul style="list-style-type: none"> ■ 0：通过其他方法获取信息 ■ 1：4mA ■ 2：8mA ■ 3：13mA ■ ... ■ 255：1020mA 复位值：0x19 存在：一直
15:8	MAX_CUR_30V	R	最大电流为 3.0V 该位指定卡的 3.0V VDD1 电源的最大电流。 <ul style="list-style-type: none"> ■ 0：通过其他方法获取信息 ■ 1：4mA ■ 2：8mA

位段	名称	类型	说明
			<ul style="list-style-type: none"> 3: 13mA ... 255: 1020mA 复位值: 0x19 存在: 总是
7:0	MAX_CUR_33V	R	最大电流为 3.3V 该位指定卡的 3.3V VDD1 电源的最大电流。 <ul style="list-style-type: none"> 0: 通过其他方法获取 1: 4mA 2: 8mA 3: 13mA ... 255: 1020mA 复位值: 0x19 存在: 一直

3.6.1.31 CURR_CAPABILITIES2_R

- 寄存器名: 最大电流容量寄存器 - 32 到 63
- 描述: 这个寄存器表示每个电压(对于 VDD2)的最大电流容量。如果在能力寄存器中设置了电压支持, 则该值有意义。如果主机系统通过另一种方法提供此信息, 则所有 Maximum Current Capabilities 寄存器都设置为 0。
- 大小: 32 位
- 偏移量: 0x4c

图表 3-74 CURR_CAPABILITIES2_R 寄存器字段说明

位段	名称	类型	说明
31:8	RSVD_63_40	R	CURR_CAPABILITIES2_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
7:0	MAX_CUR_VDD2_18V	R	最大电流为 1.8V VDD2 该位指定 UHS-II 卡 1.8V VDD2 电源的最大电流。

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0: 通过其他方法获取信息 1: 4mA 2: 8mA 3: 13mA ... 255: 1020mA 复位值: 0x0 存在: 一直

3.6.1.32 FORCE_AUTO_CMD_STAT_R

- 寄存器名: 自动 CMD 错误状态寄存器的强制事件寄存器
- 描述: 该寄存器不是物理实现的, 而是一个地址, 在该地址上可以写入自动 CMD 错误状态 (Auto CMD Error Status) 寄存器。这个寄存器适用于 SD/eMMC 模式。
 - 1: 设置自动 CMD 错误状态 (Auto CMD Error Status) 寄存器的每一位
 - 0: 无影响
- 大小: 16 位
- 偏移量: 0x50

图表 3-75 FORCE_AUTO_CMD_STAT_R 寄存器字段说明

位段	名称	类型	说明
15:8	RSVD_15_8	R	FORCE_AUTO_CMD_STAT_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
7	FORCE_CMD_NOT_ISSUED_AUTO_CMD12	W	命令不由自动 CMD12 发出错误的强制事件 值: <ul style="list-style-type: none"> 0x1 (TRUE): 命令不由自动 CMD12 发出错误状态设置 0x0 (FALSE): 无影响 复位值: 0x0 存在: 一直
6	RSVD_6	R	FORCE_AUTO_CMD_STAT_R 寄存器的这一部分是保留的。它们总是返回 0。

位段	名称	类型	说明
			复位值: 0x0 存在: 一直
5	FORCE_AUTO_CMD_RESP_ERR	W	自动 CMD 响应错误的强制事件 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 设置自动 CMD 响应错误状态 ■ 0x0 (FALSE): 无影响 复位值: 0x0 存在: 一直
4	FORCE_AUTO_CMD_IDX_ERR	W	自动 CMD 索引错误的强制事件 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 设置自动 CMD 索引错误状态 ■ 0x0 (FALSE): 无影响 复位值: 0x0 存在: 一直
3	FORCE_AUTO_CMD_EBIT_ERR	W	自动 CMD 结束位错误的强制事件 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 设置自动 CMD 结束位错误状态 ■ 0x0 (FALSE): 无影响 复位值: 0x0 存在: 一直
2	FORCE_AUTO_CMD_CRC_ERR	W	自动 CMD CRC 错误的强制事件 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 设置自动 CMD CRC 错误状态 ■ 0x0 (FALSE): 无影响 复位值: 0x0 存在: 一直
1	FORCE_AUTO_CMD_TOUT_ERR	W	自动 CMD 超时错误的强制事件 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 设置自动 CMD 超时错误状态 ■ 0x0 (FALSE): 无影响

位段	名称	类型	说明
			复位值：0x0 存在：一直
0	FORCE_AUTO_CMD12_NOT_EX EC	W	自动 CMD12 不执行的强制事件 值： <ul style="list-style-type: none"> 0x1 (TRUE): 设置自动 CMD12 不执行状态 0x0 (FALSE): 无影响 复位值：0x0 存在：一直

3.6.1.33 FORCE_ERROR_INT_STAT_R

- 寄存器名：错误中断状态的强制事件寄存器
- 描述：这个寄存器不是物理实现的，而是一个可以写入错误中断状态寄存器的地址。如果错误中断状态使能寄存器的相应位被设置，对这个地址的写的效果会反映在错误中断状态寄存器中。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：16 位
- 偏移量：0x52

图表 3-76 FORCE_ERROR_INT_STAT_R 寄存器字段说明

位段	名称	类型	说明
15	FORCE_VENDOR_ERR3	W	FORCE_ERROR_INT_STAT_R 寄存器的这个位 (FORCE_VENDOR_ERR3) 是保留的。它总是返回 0。 复位值：0x0 存在：一直
14	FORCE_VENDOR_ERR2	W	FORCE_ERROR_INT_STAT_R 寄存器的这个位 (FORCE_VENDOR_ERR2) 是保留的。它总是返回 0。 复位值：0x0 存在：一直
13	FORCE_VENDOR_ERR1	W	FORCE_ERROR_INT_STAT_R 寄存器的这个位 (FORCE_VENDOR_ERR1) 是保留的。它总是返回 0。 复位值：0x0 存在：一直
12	FORCE_BOOT_ACK_ERR	W	Force Event for Boot Ack error 引导确认错误的强制事件

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置引导确认错误状态 复位值: 0x0 存在: 一直
11	FORCE_RESP_ERR	W	响应错误强制事件（仅限 SD 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置响应错误状态 复位值: 0x0 存在: 一直
10	FORCE_TUNING_ERR	W	调优错误的强制事件（仅限 UHS-I 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置调优错误状态 复位值: 0x0 存在: 一直
9	FORCE_ADMA_ERR	W	ADMA 错误的强制事件 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置 ADMA 错误状态 复位值: 0x0 存在: 一直
8	FORCE_AUTO_CMD_ERR	W	自动 CMD 错误的强制事件为（限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置自动 CMD 错误状态 复位值: 0x0 存在: 一直
7	FORCE_CUR_LMT_ERR	W	电流限制错误的强制事件

位段	名称	类型	说明
			值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置电流限制错误状态 复位值: 0x0 存在: 一直
6	FORCE_DATA_END_BIT_ERR	W	数据结束位错误的强制事件（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置状态据结束位错误 复位值: 0x0 存在: 一直
5	FORCE_DATA_CRC_ERR	W	数据 CRC 错误的强制事件（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置数据 CRC 错误状态 复位值: 0x0 存在: 一直
4	FORCE_DATA_TOUT_ERR	W	数据超时错误的强制事件（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置数据超时错误状态 复位值: 0x0 存在: 一直
3	FORCE_CMD_IDX_ERR	W	命令索引错误强制事件（仅限 SD/eMMC 模式） 值： <ul style="list-style-type: none"> 0x0 (FALSE): 无影响 0x1 (TRUE): 设置命令索引错误状态 复位值: 0x0 存在: 一直
2	FORCE_CMD_END_BIT_ERR	W	命令结束位错误的强制事件（仅限 SD/eMMC 模式）

位段	名称	类型	说明
			值： ■ 0x0 (FALSE): 无影响 ■ 0x1 (TRUE): 设置命令结束位错误状态 复位值: 0x0 存在: 一直
1	FORCE_CMD_CRC_ERR	W	命令 CRC 错误的强制事件 (仅限 SD/eMMC 模式) 值： ■ 0x0 (FALSE): 无影响 ■ 0x1 (TRUE): 设置命令 CRC 错误状态 复位值: 0x0 存在: 一直
0	FORCE_CMD_TOUT_ERR	W	命令超时错误的强制事件 (仅限 SD/eMMC 模式) 值： ■ 0x0 (FALSE): 无影响 ■ 0x1 (TRUE): 设置命令超时错误状态 复位值: 0x0 存在: 一直

3.6.1.34 ADMA_ERR_STAT_R

- 寄存器名: ADMA 错误状态寄存器
- 描述: 这个寄存器在发生 ADMA 错误时存储 ADMA 状态。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小: 8 位
- 偏移量: 0x54

图表 3-77 ADMA_ERR_STAT_R 寄存器字段说明

位段	名称	类型	说明
7:3	RSVD_7_3	R	ADMA_ERR_STAT_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直 易失: 是

位段	名称	类型	说明
2	ADMA_LEN_ERR	R	ADMA 长度失配错误状态 以下情况下会发生此错误： <ul style="list-style-type: none"> ■ 当设置块计数使能（Block Count Enable）时，描述符表指定的总数据长度与块计数和块长度指定的数据长度不一致 ■ 当数据总长度不能除以数据块长度时 值： <ul style="list-style-type: none"> ■ 0x0 (NO_ERR): 无错误 ■ 0x1 (ERROR): 有错误 复位值：0x0 存在：一直 易失：是
1:0	ADMA_ERR_STATES	R	ADMA 错误状态 当 ADMA 数据传输过程中发生错误时, 这些位表示 ADMA 的状态。 值： <ul style="list-style-type: none"> ■ 0x0 (ST_STOP): 停止 DMA - SYS_ADR 寄存器指向错误描述符旁边的位置 ■ 0x1 (ST_FDS): 获取描述符- SYS_ADR 寄存器指向错误描述符 ■ 0x2 (UNUSED): 不要设置此状态 ■ 0x3 (ST_TFR): 传输数据 - SYS_ADR 寄存器指向错误描述符旁边的位置 复位值：0x0 存在：一直 易失：是

3.6.1.35 ADMA_SA_LOW_R

- 寄存器名：ADMA 系统地址寄存器 - 低
- 描述：这个寄存器保存用于 DMA 传输的低 32 位系统地址。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：32 位
- 偏移量：0x58

图表 3-78 ADMA_SA_LOW_R 寄存器字段说明

位段	名称	类型	说明
31:0	ADMA_SA_LOW	R/W	ADMA 系统地址 这些位表示 ADMA 系统地址的低 32 位。 <ul style="list-style-type: none"> ■ SDMA: 如果主机版本 4 使能 (Host Version 4 Enable) 设置为 1, 这个寄存器存储数据位置的系统地址 ■ ADMA2: 这个寄存器存储描述符表中执行命令的字节地址 ■ ADMA3: 该寄存器由 ADMA3 设置。每次获取一个描述符行, ADMA2 就增加指向下一行的寄存器地址。 复位值: 0x0 存在: 一直 易失: 是

3.6.1.36 ADMA_SA_HIGH_R

- 寄存器名: ADMA 系统地址寄存器 - 高
- 描述: 这个寄存器保存着用于 DMA 传输的高 32 位系统地址。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小: 32 位
- 偏移量: 0x5c

图表 3-79 ADMA_SA_HIGH_R 寄存器字段说明

位段	名称	类型	说明
31:0	ADMA_SA_HIGH	R/W	ADMA 系统地址 这些位表示 ADMA 系统地址的高 32 位 复位值: 0x0 存在: 一直 易失: 是

3.6.1.37 PRESET_INIT_R

- 寄存器名: 初始化的预设值
- 描述: 这个寄存器定义了 SD/eMMC 模式下初始化的预设值。
- 大小: 16 位
- 偏移量: 0x60

图表 3-80 PRESET_INIT_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示驱动器的强度由 1.8V 信令总线速度模式支持。这些位对于 3.3V 信号是没有意义的。 值： <ul style="list-style-type: none"> ■ 0x0 (TYPEB): 选择驱动类型 B ■ 0x1 (TYPEA): 选择驱动类型 A ■ 0x2 (TYPEC): 选择驱动类型 C ■ 0x3 (TYPED): 选择驱动类型 D 复位值: 0x0 存在: 一直
13:11	RSVD_13_11	R	PRESET_INIT_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主控制器支持可编程时钟发生器时, 该位是有效的。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE): 主机控制器版本 2.0 兼容时钟生成器 ■ 0x1 (PROG): 可编程时钟发生器 复位值: 0x0 存在: 一直
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置的 10 位预设值。 复位值: 250 存在: 一直

3.6.1.38 PRESET_DS_R

- 寄存器名: 默认速度的预设值
- 描述: 这个寄存器为 SD 模式下的默认速度模式定义了预设值。
- 大小: 16 位

- 偏移量：0x62

图表 3-81 PRESET_DS_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 1.8V 信令总线速度模式支持的驱动器强度值。对于默认速度模式,该字段没有意义,因为该模式使用的是 3.3V 信令。 值: <ul style="list-style-type: none"> ■ 0x0 (TYPEB): 选择驱动类型 B ■ 0x1 (TYPEA): 选择驱动类型 A ■ 0x2 (TYPEC): 选择驱动类型 C ■ 0x3 (TYPED): 选择驱动类型 D 复位值: 0x0 存在: 一直
13:11	RSVD_13_11	R	PRESET_DS_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时, 该位有效。 值: <ul style="list-style-type: none"> ■ 0x0 (FALSE): 主机控制器版本 2.0 兼容时钟生成器 ■ 0x1 (PROG): 可编程时钟发生器 复位值: 0 存在: 总是
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置的 10 位预设值。 复位值: 4 存在: 一直

3.6.1.39 PRESET_HS_R

- 寄存器名: 高速的预设值

- 描述：这个寄存器为 SD 模式下的高速模式定义了预设值。
- 大小：16 位
- 偏移量：0x64

图表 3-82 PRESET_HS_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 1.8V 信令总线速度模式支持的驱动器强度值。该字段在高速模式下没有意义，因为它使用 3.3V 信令。 值： <ul style="list-style-type: none"> ■ 0x0 (TYPEB)：选择驱动类型 B ■ 0x1 (YPEA)：选择驱动类型 A ■ 0x2 (YPEC)：选择驱动类型 C ■ 0x3 (YPED)：选择驱动类型 D 复位值：0x0 存在：一直
13:11	RSVD_13_11	R	PRESET_HS_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时，该位有效。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：主机控制器版本 2.0 兼容时钟生成器 ■ 0x1 (PROG)：可编程时钟发生器 复位值：0 存在：一直
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置的 10 位预设值。 复位值：2 存在：一直

3.6.1.40 PRESET_SDR12_R

- 寄存器名：SDR12 的预设值
- 描述：该寄存器分别为 SD 和 eMMC 模式下的 SDR12 和 Legacy 速度模式定义了预设值。
- 大小：16 位
- 偏移量：0x66

图表 3-83 PRESET_SDR12_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 SDR12 总线速度模式支持的驱动器强度值。这些位对于 3.3V 信号是没有意义的。 值： <ul style="list-style-type: none"> ■ 0x0 (TYPEB)：选择驱动类型 B ■ 0x1 (TYPEA)：选择驱动类型 A ■ 0x2 (TYPEC)：选择驱动类型 C ■ 0x3 (TYPED)：选择驱动类型 D 复位值：0 存在：一直
13:11	RSVD_13_11	R	PRESET_SDR12_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时，该位有效 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：主机控制器版本 2.0 兼容时钟生成器 ■ 0x1 (PROG)：可编程时钟发生器 复位值：0 存在：一直
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置的 10 位预设值。 复位值：4

位段	名称	类型	说明
			存在：一直

3.6.1.41 PRESET_SDR25_R

- 寄存器名：SDR25 的预设值
- 描述：该寄存器分别为 SD 和 eMMC 模式下的 SDR25 和 High Speed SDR 速度模式定义了预设值。
- 大小：16 位
- 偏移量：0x68

图表 3-84 PRESET_SDR25_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 SDR25 总线速度模式支持的驱动器强度值。这些位对于 3.3V 信号是没有意义的。 值： <ul style="list-style-type: none"> ■ 0x0 (TYPEB)：选择驱动类型 B ■ 0x1 (TYPEA)：选择驱动类型 A ■ 0x2 (YPEC)：选择驱动类型 C ■ 0x3 (TYPED)：选择驱动类型 D 复位值：0 存在：一直
13:11	RSVD_13_11	R	PRESET_SDR25_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时，该位有效。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：主机控制器版本 2.0 兼容时钟生成器 ■ 0x1 (PROG)：可编程时钟发生器 复位值：0 存在：一直
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值

			在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置的 10 位预设值。 复位值：2 存在：一直
--	--	--	---

3.6.1.42 PRESET_SDR50_R

- 寄存器名：SDR50 的预设值
- 描述：这个寄存器为 SD 模式下的 SDR50 速度模式定义了预设值。
- 大小：16 位
- 偏移量：0x6a

图表 3-85 PRESET_SDR50_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 SDR50 总线速度模式支持的驱动器强度值。这些位对于 3.3V 信号是没有意义的。 值： <ul style="list-style-type: none"> ■ 0x0 (TYPEB)：选择驱动类型 B ■ 0x1 (TYPEA)：选择驱动类型 A ■ 0x2 (YPEC)：选择驱动类型 C ■ 0x3 (TYPED)：选择驱动类型 D 复位值：0 存在：一直
13:11	RSVD_13_11	R	PRESET_SDR50_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时，该位有效。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：主机控制器版本 2.0 兼容时钟生成器 ■ 0x1 (PROG)：可编程时钟发生器 复位值：0 存在：一直

位段	名称	类型	说明
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置的 10 位预设值。 复位值：1 存在：一直

3.6.1.43 PRESET_SDR104_R

- 寄存器名：SDR104 的预设值
- 描述：该寄存器分别为 SD 和 eMMC 模式下的 SDR104 和 HS200 速度模式定义了预设值。
- 大小：16 位
- 偏移量：0x6c

图表 3-86 PRESET_SDR104_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 SDR104 总线速度模式支持的驱动器强度值。这些位对于 3.3V 信号是没有意义的。 值： <ul style="list-style-type: none"> ■ 0x0 (TYPEB)：选择驱动类型 B ■ 0x1 (TYPEA)：选择驱动类型 A ■ 0x2 (TYPEC)：选择驱动类型 C ■ 0x3 (TYPED)：选择驱动类型 D 复位值：0 存在：一直
13:11	RSVD_13_11	R	PRESET_SDR104_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时，该位有效。 值： <ul style="list-style-type: none"> ■ 0x0 (FALSE)：主机控制器版本 2.0 兼容时钟生成器

位段	名称	类型	说明
			■ 0x1 (PROG): 可编程时钟发生器 复位值: 0 存在: 一直
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 这些位指定了一个 10 位的预设值, 该预设值必须在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置。 复位值: 0 存在: 一直

3.6.1.44 PRESET_DDR50_R

- 寄存器名: DDR50 的预设值
- 描述: 这个寄存器分别定义了 SD 和 eMMC 模式下的 DDR50 和高速 DDR 速度模式的预设值。
- 大小: 16 位
- 偏移量: 0x6e

图表 3-87 PRESET_DDR50_R 寄存器字段说明

位段	名称	类型	说明
15:14	DRV_SEL_VAL	R	驱动强度选择值 这些位表示 DDR50 总线速度模式支持的驱动器强度值。这些位对于 3.3V 信号是没有意义的。 值: ■ 0x0 (TYPEB): 选择驱动类型 B ■ 0x1 (TYPEA): 选择驱动类型 A ■ 0x2 (TYPEC): 选择驱动类型 C ■ 0x3 (TYPED): 选择驱动类型 D 复位值: 0 存在: 一直
13:11	RSVD_13_11	R	PRESET_DDR50_R 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直

位段	名称	类型	说明
10	CLK_GEN_SEL_VAL	R	时钟发生器选择值 当主机控制器支持可编程时钟发生器时，该位有效。 值： <ul style="list-style-type: none"> 0x0 (FALSE): 主机控制器版本 2.0 兼容时钟生成器 0x1 (PROG): 可编程时钟发生器 复位值：0 存在：一直
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK 频率选择值 这些位指定了一个 10 位的预设值，该预设值必须在主机系统描述的时钟控制寄存器的 SDCLK/RCLK 频率选择字段中设置。 复位值：2 存在：一直

3.6.1.45 ADMA_ID_LOW_R

- 寄存器名：ADMA3 综合描述符地址寄存器 - 低
- 描述：这个寄存器保存低 32 位综合描述符地址。该寄存器适用于 SD/eMMC/UHS-II 模式。
- 大小：32 位
- 偏移量：0x78

图表 3-88 ADMA_ID_LOW_R 寄存器字段说明

位段	名称	类型	说明
31:0	ADMA_ID_LOW	R/W	ADMA 综合描述符地址 这些位指示 ADMA 综合描述符地址的低 32 位。综合描述符的起始地址设置为这些寄存器位。ADMA3 获取一个描述符地址并增加这些位以指示下一个描述符地址。 复位值：0x0 存在：一直 易失：是

3.6.1.46 ADMA_ID_HIGH_R

- 寄存器名：ADMA3 综合描述符地址寄存器--高
- 描述：这个寄存器保存着高 32 位综合描述符地址。该寄存器适用于 SD/eMMC/UHS-II 模式。

- 大小：32 位
- 偏移量：0x7c

图表 3-89 ADMA_ID_HIGH_R 寄存器字段说明

位段	名称	类型	说明
31:0	ADMA_ID_HIGH	R/W	ADMA 综合描述符地址 这些位表示 ADMA 综合描述符地址的高 32 位。 复位值：0x0 存在：一直 易失：是

3.6.1.47 SLOT_INTR_STATUS_R

- 寄存器名：插槽中断状态寄存器
- 描述：这个寄存器指示每个插槽的中断状态。
- 大小：16 位
- 偏移量：0xfc

图表 3-90 SLOT_INTR_STATUS_R 寄存器字段说明

位段	名称	类型	说明
15:8	RESERVED_15_8	R	SLOT_INTR_STATUS_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直 易失：是
7:0	INTR_SLOT	R	每个插槽的中断信号 这些状态位表示每个插槽的中断信号和唤醒信号的逻辑顺序。最多可以定义 8 个插槽。如果一个中断信号与多个插槽相关联，主机驱动程序可以通过读取这些位识别产生的中断。通过电源复位或设置软件为所有位复位（Software Reset For All），中断信号失效，这个状态读取 00h。 <ul style="list-style-type: none"> ■ Bit 00：插槽 1 ■ Bit 01：插槽 2 ■ Bit 02：插槽 3 ■ ■

位段	名称	类型	说明
			<ul style="list-style-type: none"> Bit 07: 插槽 8 注意: MSHC 主机控制器支持单卡槽。这个寄存器总是返回 0。 复位值: 0x0 存在: 一直 易失: 是

3.6.1.48 HOST_CNTRL_VERS_R

- 寄存器名: 主机控制器版本
- 描述: 这个寄存器用于指示主机控制器版本号。
- 大小: 16 位
- 偏移量: 0xfe

图表 3-91 HOST_CNTRL_VERS_R 寄存器字段说明

位段	名称	类型	说明
15:8	VENDOR_VERSION_NUM	R	厂商的版本号 此字段为厂商版本号保留。主机驱动程序不能使用此状态。 复位值: 0 存在: 一直
7:0	SPEC_VERSION_NUM	R	规范版本号 这些位表示主机控制器规范版本。上四位和下四位分别表示版本号。0x06-0xFF 为预留值。 值: <ul style="list-style-type: none"> 0x0 (VER_1_00): SD 主机控制器规范版本 1.00 0x1 (VER_2_00): SD 主机控制器规范版本 2.00 0x2 (VER_3_00): SD 主机控制器规范版本 3.00 0x3 (VER_4_00): SD 主机控制器规范版本 4.00 0x4 (VER_4_10): SD 主机控制器规范版本 4.10 0x5 (VER_4_20): SD 主机控制器规范版本 4.20 复位值: 5 存在: 一直

3.6.2 DWC_mshc_map/DWC_mshc_embedded_control_block 寄存器

3.6.2.1 EMBEDDED_CTRL_R

- 寄存器名：嵌入式控制寄存器
- 描述：这个寄存器控制嵌入式设备。当主机控制器连接到可移动设备时，这个寄存器不被使用。
- 大小：32 位
- 偏移量：0xf6c

图表 3-92 EMBEDDED_CTRL_R 寄存器字段说明

位段	名称	类型	说明
31	RSVD_31	R	EMBEDDED_CTRL_R 寄存器的这个位(RSVD_31)是保留的。 它总是返回 0。 复位值：0x0 存在：一直
30:24	BACK_END_PWR_CTRL	R/W	后端电源控制(SD 模式) 这个字段的每个位控制嵌入式设备的后端电源。 <ul style="list-style-type: none"> ■ 0：后端电源关闭 ■ 1：后端供电 D24：设备 1 后端电源 D25：设备 2 后端电源 D26：设备 3 后端电源 D27：设备 4 后端电源 D28：设备 5 后端电源 D29：设备 6 后端电源 D30：设备 7 后端电源 复位值：0x0 存在：一直
23	RSVD_23	R	EMBEDDED_CTRL_R 寄存器的这个位(RSVD_23)是保留的。 它总是返回 0。 复位值：0x0 存在：一直
22:20	INT_PIN_SEL	R/W	中断管脚选择

位段	名称	类型	说明
			这些位启用中断管脚输入。 <ul style="list-style-type: none"> ■ 000: 禁用中断 (INT_A、INT_B、INT_C)。 ■ xx1: 启用 INT_A。 ■ x1x: 启用 INT_B。 ■ 1xx: 启用 INT_C。 复位值: 0x0 存在: 一直
19	RSVD_19	R	EMBEDDED_CTRL_R 寄存器的这个位(RSVD_19)是保留的。 它总是返回 0。 复位值: 0x0 存在: 一直
18:16	CLK_PIN_SEL	R/W	时钟管脚选择 (SD 模式) 该位由其中一个时钟管脚输出选择。 <ul style="list-style-type: none"> ■ 0x0: 关闭时钟管脚 ■ 0x1: 选择 CLK [1] ■ 0x2: 选择 CLK [2] ■ ... ■ ... ■ ... ■ 0x7: 选择 CLK [7] 复位值: 0x0 存在: 一直
15	RSVD_15	R	EMBEDDED_CTRL_R 寄存器的这个位(RSVD_15)是保留的。 它总是返回 0。 复位值: 0x0 存在: 一直
14:8	BUS_WIDTH_PRESET	R	总线宽度预设(SD 模式) 该字段的各个位指定各个嵌入式设备的总线宽度。共享总线支持 4 位和 8 位总线宽度设备的混合。 <ul style="list-style-type: none"> ■ D08: 设备 1 的总线宽度预置 ■ D09: 设备 2 的总线宽度预置

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ D10: 设备 3 的总线宽度预置 ■ D11: 设备 4 的总线宽度预置 ■ D12: 设备 5 的总线宽度预置 ■ D13: 设备 6 的总线宽度预置 ■ D14: 设备 7 的总线宽度预置 每个位的函数定义如下: <ul style="list-style-type: none"> ■ 0: 4 位总线宽度模式 ■ 1: 8 位总线宽度模式 复位值: 0x0 存在: 一直
7:6	RSVD_7_6	R	EMBEDDED_CTRL_R 寄存器的这些位(RSVD_7_6)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
5:4	NUM_INT_PIN	R	中断输入管脚数 该字段表示对嵌入式系统中中断输入管脚的支持。 复位值: 0x0 存在: 一直
3	RSVD_3	R	EMBEDDED_CTRL_R 寄存器的这个位(RSVD_3)是保留的。它总是返回 0。 复位值: 0x0 存在: 一直
2:0	NUM_CLK_PIN	R	时钟管脚数 (SD 模式) 这个字段表示支持时钟管脚为共享总线系统选择一个设备。最多可支持 7 个时钟管脚。 <ul style="list-style-type: none"> ■ 0x0: 不支持共享总线 ■ 0x1: 支持 1 SDCLK ■ 0x2: 支持 2 SDCLK ■ ... ■ ... ■ ...

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x7: 支持 7 SDCLK 复位值: 0x0 存在: 一直

3.6.3 DWC_mshc_map/DWC_mshc_vendor2_block 寄存器

3.6.3.1 CQVER

- 寄存器名: 命令队列版本寄存器
- 描述: 该寄存器提供有关 eMMC 命令队列标准的版本信息, 该标准由命令队列引擎以 BCD 格式实现。
- 大小: 32 位
- 偏移量: 0x180

图表 3-93 CQVER 寄存器字段说明

位段	名称	类型	说明
31:12	EMMMC_VER_RSVD	R	CQVER 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
11:8	EMMC_VER_MAJOR	R	该位表示 BCD 格式的 eMMC 主版本(小数点后左边第 1 位)。 复位值: 0x5 存在: 一直
7:4	EMMC_VER_MINOR	R	该位表示 BCD 格式的 eMMC 小版本号(小数点右边第 1 位)。 复位值: 0x1 存在: 一直
3:0	EMMC_VER_SUFFIX	R	该位表示 BCD 格式的 eMMC 版本后缀(小数点右边第二位)。 复位值: 0x0 存在: 一直

3.6.3.2 CQCAP

- 寄存器名: 命令队列能力寄存器
- 描述: 这个寄存器表示命令队列引擎的功能。
- 大小: 32 位

- 偏移量：0x180 + 0x4

图表 3-94 CQCAP 寄存器字段说明

位段	名称	类型	说明
31:29	CQCCAP_RSVD3	R	CQCAP 寄存器的这些位[31:29]是保留的。它们总是返回 0。 复位值：0x0 存在：一直
28	CRYPTO_SUPPORT	R	支持加密 该位表示主机控制器是否支持加密操作。 值： ■ 0x0 (FALSE)：不支持加密。 ■ 0x1 (TRUE)：支持加密。 复位值：0 存在：一直
27:16	CQCCAP_RSVD2	R	CQCAP 寄存器的这些位[27:16]是保留的。它们总是返回 0。 复位值：0x0 存在：一直
15:12	ITCFMUL	R	内部定时器时钟倍频器(ITCFMUL) 此字段指示用于中断合并计时器和确定 SQS 轮询周期的时钟频率。参见 ITCFVAL 定义的详细信息。0x5 ~ 0xF 为保留值。 值： ■ 0x0 (CLK_1KHz)：1KHz 时钟 ■ 0x1 (CLK_10KHz)：10KHz 时钟 ■ 0x2 (CLK_100KHz)：100KHz 时钟 ■ 0x3 (CLK_1MHz)：1MHz 时钟 ■ 0x4 (CLK_10MHz)：10MHz 时钟 复位值：3 存在：一直
11:10	CQCCAP_RSVD1	R	CQCAP 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
9:0	ITCFVAL	R	内部定时器时钟频率(ITCFVAL)

位段	名称	类型	说明
			该字段标度 scales 由 ITCFMUL 提供的定时器时钟的频率。 实际计时器时钟的最终时钟频率计算为 ITCFVAL * ITCFMUL。 复位值：0 存在：一直

3.6.3.3 CQCFG

- 寄存器名：命令队列配置寄存器
- 描述：这个寄存器控制影响命令队列引擎一般操作的 CQE 行为。
- 大小：32 位
- 偏移量：0x180 + 0x8

图表 3-95 CQCFG 寄存器字段说明

位段	名称	类型	说明
31:13	CQCCFG_RSVD3	R	CQCFG 寄存器的这些位(CQCCFG_RSVD3)是保留的。它们总是返回 0。 复位值：0x0 存在：一直
12	DCMD_EN	R/W	该位指示硬件 TDL 插槽 31 号中的任务描述符是数据传输描述符或直接命令描述符。当任务在插槽 31 号发出时，CQE 使用该位来确定如何解码任务描述符。 值： <ul style="list-style-type: none"> ■ 0x1 (SLOT31_DCMD_ENABLE)：在插槽 31 号中的任务描述符是一个 DCMD 任务描述符。 ■ 0x0 (SLOT31_DCMD_DISABLE)：在插槽 31 号中的任务描述符是一个数据传输任务描述符。 复位值：0x0 存在：一直
11:9	CQCCFG_RSVD2	R	CQCFG 寄存器的这些位(CQCCFG_RSVD2)是保留的。它们总是返回 0。 复位值：0x0 存在：一直
8	TASK_DESC_SIZE	R/W	位值描述 该位表示主机内存中使用的任务描述符的大小。只有当命令

位段	名称	类型	说明
			队列使能 (Command Queuing Enable) 为 0 (命令队列未使能) 时, 才能配置该位。 值: <ul style="list-style-type: none"> ■ 0x1 (TASK_DESC_128b): 任务描述符大小为 128 位。 ■ 0x0 (TASK_DESC_64b): 任务描述符大小为 64 位。 复位值: 0x0 存在: 一直
7:2	CQCCFG_RSVD1	R	CQCCFG 寄存器的这些位(CQCCFG_RSVD1)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
1	Reserved	R	
0	CQ_EN	R/W	启用命令队列引擎(CQE)。 当 CQE 关闭时, 软件使用地址 0x000 到 0x1FF 之间的寄存器来控制 eMMC 总线。在软件写入该位之前, 软件会检查 eMMC 主机控制器是否处于空闲状态, 是否有正在执行的命令或数据传输。当软件想要退出命令队列模式时, 它将该位设置为 0 之前清除所有之前的任务(如果有)。 值: <ul style="list-style-type: none"> ■ 0x1 (CQE_ENABLE): 使能命令队列。 ■ 0x0 (CQE_DISABLE): 关闭命令队列。 复位值: 0x0 存在: 一直 测试: 恢复

3.6.3.4 CQCTL

- 寄存器名: 命令队列控制寄存器
- 描述: 该寄存器控制影响命令队列模块一般操作或多个任务同时操作的 CQE 行为。
- 大小: 32 位
- 偏移量: 0x180 + 0xc

图表 3-96 CQCTL 寄存器字段说明

位段	名称	类型	说明
31:9	CQCTL_RSVD2	R	这些位(CQCTL_RSVD2)的 CQCTL 寄存器是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
8	CLR_ALL_TASKS	R/W	清除所有任务 该位只能在控制器停止时写入。此位不清除设备中的任务。软件需要使用 CMDQ_TASK_MGMT 命令清除设备队列。 值: <ul style="list-style-type: none"> 0x1 (CLEAR_ALL_TASKS): 清除控制器上的所有任务。 0x0 (NO_EFFECT): 编程 0 没有效果。 复位值: 0x0 存在: 一直 易失: 是
7:1	CQCTL_RSVD1	R	CQCTL 寄存器的这些位(CQCTL_RSVD1)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
0	HALT	R/W	暂停请求和恢复 值: <ul style="list-style-type: none"> 0x1 (HALT_CQE): 当软件想要获得对 eMMC 总线的软件控制并禁止 CQE 在总线上发出命令时, 就向该位写入 1。例如, 发出一个丢弃任务 (Discard Task) 命令 (CMDQ_TASK_MGMT)。当软件写入 1 时, CQE 完成正在进行的任务(如果有)。任务完成后, CQE 处于空闲状态时, CQE 不再发出新的命令, 而是通过将该位置 1 来指示软件。软件可以对该位进行轮询, 直到该位被设置为 1, 然后才在 eMMC 总线上发送命令。 0x0 (RESUME_CQE): 软件写 0 到该位, 以退出暂停状态并恢复 CQE 活动。

位段	名称	类型	说明
			复位值: 0x0 存在: 一直 易失: 是

3.6.3.5 CQIS

- 寄存器名: 命令队列中断状态寄存器
- 描述: 这个寄存器指示需要服务的暂挂中断。只有在 CQISE 寄存器中设置了相应位后, 该寄存器中的每一个位才会响应特定的事件。
- 大小: 32 位
- 偏移量: 0x180 + 0x10

图表 3-97 CQIS 寄存器字段说明

位段	名称	类型	说明
31:6	CQIS_RSVD1	R	CQIS 寄存器的这些位是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直 易失: 是
5	Reserved	R	
4	Reserved	R	
3	TCL	R/W1C	任务清除中断 当 CQE 完成任务清除操作时, 这个状态位生效 (如果 CQISE.TCL_STE = 1)。已完成的任务清除操作可以是单个任务清除 (通过写入 CQTCLR) 或清除所有任务 (通过写入 CQCTL)。 值为 1 时清除此状态位。 值: <ul style="list-style-type: none"> ■ 0x1 (SET): 设置了 TCL 中断。 ■ 0x0 (NOTSET): 没有设置 TCL 中断。 复位值: 0x0 存在: 一直 易失: 是
2	RED	R/W1C	检测到中断的响应错误

位段	名称	类型	说明
			<p>当收到一个在设备状态字段中设置的错误位的响应时，这个状态位生效(如果 CQISE.RED_STE = 1)。配置 CQRMEM 寄存器以识别可能触发中断和被屏蔽的设备状态位域。值为 1 时清除此状态位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (SET): RED 中断已设置。 ■ 0x0 (NOTSET): RED 中断未设置。 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
1	TCC	R/W1C	<p>任务完成中断</p> <p>当满足以下至少一个条件时，这个状态位生效(如果 CQISE.TCC_STE = 1):</p> <ul style="list-style-type: none"> ■ 一个任务完成，INT 位在其任务描述符中被设置 ■ 中断合并逻辑超时导致的中断 ■ 中断合并逻辑达到配置的阈值 ■ 值为 1 时清除该状态位。 <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (SET): TCC 中断已设置。 ■ 0x0 (NOTSET): TCC 中断未设置。 <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>
0	HAC	R/W1C	<p>停止完成中断</p> <p>当 CQCTL 寄存器中的 halt 位从 0 转换到 1 (只有当 CQISE.HAC_STE=1)时，该状态位才会生效，表示主控制器已完成当前正在进行的任务并进入 halt 状态。值为 1 时清除此状态位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (SET): HAC 中断已设置。 ■ 0x0 (NOTSET): HAC 中断未设置。 <p>复位值：0x0</p>

位段	名称	类型	说明
			存在：一直 易失：是

3.6.3.6 CQISE

- 寄存器名：命令队列中断状态使能寄存器
- 描述：这个寄存器启用和禁用在 CQIS 寄存器中相应中断向主机软件的报告。当一个位被设置为(1)并且相应的中断条件是激活的，那么 CQIS 中的位生效。被禁用的中断源(当'0'时)在 CQIS 寄存器中没有指示。这个寄存器与 CQIS 寄存器是位索引匹配的。
- 大小：32 位
- 偏移量：0x180 + 0x14

图表 3-98 CQISE 寄存器字段说明

位段	名称	类型	说明
31:6	CQISTE_RSVD1	R	CQISE 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
5	Reserved	R	
4	Reserved	R	
3	TCL_STE	R/W	任务清除中断状态使能 值： <ul style="list-style-type: none"> ■ 0x1 (INT_STS_ENABLE): CQIS.TCL 在其中断条件处于激活状态时设置。 ■ 0x0 (INT_STS_DISABLE): CQIS.TCL 被禁用。 复位值：0x0 存在：一直
2	RED_STE	R/W	响应错误检测中断状态使能 值： <ul style="list-style-type: none"> ■ 0x1 (INT_STS_ENABLE): CQIS.RED 在其中断条件处于激活状态时设置。 ■ 0x0 (INT_STS_DISABLE): CQIS.RED 被禁用。 复位值：0x0 存在：一直

位段	名称	类型	说明
1	TCC_STE	R/W	任务完成中断状态使能 值： ■ 0x1 (INT_STS_ENABLE): CQIS.TCC 在其中断条件处于激活状态时设置。 ■ 0x0 (INT_STS_DISABLE): CQIS.TCC 被禁用。 复位值：0x0 存在：一直
0	HAC_STE	R/W	停止完成中断状态使能 值： ■ 0x1 (INT_STS_ENABLE): CQIS.HAC 在其中断条件处于激活状态时设置。 ■ 0x0 (INT_STS_DISABLE): CQIS.HAC 被禁用。 复位值：0x0 存在：一直

3.6.3.7 CQISGE

- 寄存器名：命令队列中断信号使能寄存器
- 描述：这个寄存器启用和禁用对主机软件中断的产生。当一个位被设置并且在 CQIS 中相应的位也被设置时，就会产生一个中断。被禁用的中断源仍然在 CQIS 寄存器中显示。这个寄存器与 CQIS 寄存器是位索引匹配的。
- 大小：32 位
- 偏移量：0x180 + 0x18

图表 3-99 CQISG 寄存器 E 字段说明

位段	名称	类型	说明
31:6	CQISGE_RSVD1	R	CQISGE 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
5	Reserved	R	
4	Reserved	R	
3	TCL_SGE	R/W	任务清除中断信号使能 值：

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x1 (INT_SIG_ENABLE): CQIS.TCL 中断信号生成处于激活状态。 ■ 0x0 (INT_SIG_DISABLE): CQIS.TCL 中断信号生成被禁用。 复位值: 0x0 存在: 一直
2	RED_SGE	R/W	响应错误检测中断信号使能 值: <ul style="list-style-type: none"> ■ 0x1 (INT_SIG_ENABLE): CQIS.RED 中断信号生成处于激活状态。 ■ 0x0 (INT_SIG_DISABLE): CQIS.RED 中断信号生成被禁用。 复位值: 0x0 存在: 一直
1	TCC_SGE	R/W	任务完成中断信号使能 值: <ul style="list-style-type: none"> ■ 0x1 (INT_SIG_ENABLE): CQIS.TCC 中断信号生成处于激活状态。 ■ 0x0 (INT_SIG_DISABLE): CQIS.TCC 中断信号生成被禁用。 复位值: 0x0 存在: 一直
0	HAC_SGE	R/W	停止完成中断信号使能 值: <ul style="list-style-type: none"> ■ 0x1 (INT_SIG_ENABLE): CQIS.HAC 中断信号生成处于激活状态。 ■ 0x0 (INT_SIG_DISABLE): CQIS.HAC 中断信号生成被禁用。 复位值: 0x0 存在: 一直

3.6.3.8 CQIC

- 寄存器名：命令队列中断合并寄存器
- 描述：这个寄存器控制和配置中断合并功能。
- 大小：32 位
- 偏移量：0x180 + 0x1c

图表 3-100 CQIC 寄存器字段说明

位段	名称	类型	说明
31	INTC_EN	R/W	中断合并使能位 值： <ul style="list-style-type: none"> ■ 0x1 (ENABLE_INT_COALESCING)：中断合并机制激活。对中断进行计数和计时，并生成合并中断。 ■ 0x0 (DISABLE_INT_COALESCING)：断合并机制被禁用(默认)。 复位值：0x0 存在：一直
30:21	CQIC_RSVD3	R	CQIC 寄存器的这些位(CQIC_RSVD3)是保留的。它们总是返回 0。 复位值：0x0 存在：一直
20	INTC_STAT	R	中断合并状态位 该位指示软件是否有任务(INT = 0)已经完成并计入中断合并(当且仅当 INTC 计数器 > 0 时设置该值)。 值： <ul style="list-style-type: none"> ■ 0x1 (INTC_ATLEAST1_COMP)：至少有一个 INT0 任务完成计数 (INTC 计数器 > 0)。 ■ 0x0 (INTC_NO_TASK_COMP)：自上次计数器复位以来 (INTC 计数器 == 0)，INT0 任务完成未发生。 复位值：0x0 存在：一直
19:17	CQIC_RSVD2	R	CQIC 寄存器的这些位(CQIC_RSVD2)是保留的。它们总是返回 0。 复位值：0x0 存在：一直

位段	名称	类型	说明
16	INTC_RST	W	<p>计数器和定时器复位</p> <p>当主机驱动写入 1 时，中断合并定时器和计数器被复位。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (ASSERT_INTC_RESET): 中断合并定时器和计数器被复位 ■ 0x0 (NO_EFFECT): 无影响 <p>复位值：0x0</p> <p>存在：一直</p>
15	INTC_TH_WEN	W	<p>中断合并计数器阈值写使能</p> <p>当软件写入 1 到该位时，值 INTC_TH 会随着相同周期内写入的内容而更新。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (WEN_SET): 设置 INTC_TH_WEN。 ■ 0x0 (WEN_CLR): 清除 INTC_TH_WEN。 <p>复位值：0x0</p> <p>存在：一直</p>
14:13	CQIC_RSVD1	R	<p>CQIC 寄存器的这些位(CQIC_RSVD1)是保留的。它们总是返回 0。</p> <p>复位值：0x0</p> <p>存在：一直</p>
12:8	INTC_TH	W	<p>中断合并计数器阈值字段</p> <p>软件使用这个字段来配置任务完成的数量(仅任务描述符中 INT = 0 的任务)，这是生成中断所必需的。</p> <p>计数器操作：当 INT=0 的数据传输任务完成时，由 CQE 计数。计数器在中断服务程序期间由软件复位。当计数器达到 INTC_TH 中配置的值时停止计数，并产生中断。</p> <ul style="list-style-type: none"> ■ 0x0: 中断合并功能被禁用 ■ 0x1: 当 INT=0 完成时，1 个任务后产生的中断合并中断 ■ 0x2: 当 INT=0 完成时，2 个任务后产生的中断合并中断 ■ ■ 0x1f: 当 INT=0 完成时，31 个任务后产生的中断合并中断

位段	名称	类型	说明
			要写入该字段, 必须在相同的写操作期间设置 INTC_TH_WEN 位。 复位值: 0x0 存在: 一直 测试: 不可测试
7	TOUT_VAL_WEN	W	当软件写入 1 到该位时, TOUT_VAL 值会随着同一周期写入的内容而更新。 值: <ul style="list-style-type: none"> 0x1 (WEN_SET): 设置 TOUT_VAL_WEN。 0x0 (WEN_CLR): 清除 TOUT_VAL_WEN。 复位值: 0x0 存在: 一直
6:0	TOUT_VAL	R/W	中断合并超时值 软件使用此字段配置在总线上完成任务和产生中断之间允许的最大时间。 定时器操作: 在中断服务程序期间, 由软件复位定时器。当第一个 INT=0 的数据传输任务完成时, 定时器被复位, 它开始运行。当定时器到达 ICTOVAL 字段中配置的值时, 产生一个中断并停止。 定时器的单位等于时钟的 1024 个时钟周期, 其频率在内部定时器时钟频率字段 CQCAP 寄存器中指定。 <ul style="list-style-type: none"> 0x0: 关闭定时器。不会产生基于超时的中断 0x1: 定时器时钟频率 01x1024 周期的超时 0x2: 定时器时钟频率 02x1024 周期的超时 0x7f: 定时器时钟频率为 127x1024 个周期的超时 为了写入该字段, TOUT_VAL_WEN 位必须设置为相同的写操作。 复位值: 0x0 存在: 一直 测试: 不可测试

3.6.3.9 CQDLBA

- 寄存器名：命令队列任务描述符列表基址寄存器
- 描述：这个寄存器用于配置主机内存中任务描述符列表头的字节地址的低 32 位。
- 大小：32 位
- 偏移量：0x180 + 0x20

图表 3-101 CQDLBA 寄存器字段说明

位段	名称	类型	说明
31:0	TDLBA	R/W	<p>这个寄存器存储系统内存中任务描述符列表头字节地址的 LSB 位(31:0)。</p> <p>根据主机驱动配置，任务描述符列表的大小为 32*(任务描述符大小+传输描述符大小)。这个地址设置在 1KB 的边界上。该寄存器的低 10 位被软件设置为 0，并被 CQE 忽略。</p> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>

3.6.3.10 CQDLBAU

- 寄存器名：命令队列任务描述符列表基址高位寄存器
- 描述：这个寄存器用于配置主机内存中任务描述符列表头字节地址的高 32 位。
- 大小：32 位
- 偏移量：0x180 + 0x24

图表 3-102 寄存器 CQDLBAU 字段说明

位段	名称	类型	说明
31:0	TDLBAU	R/W	<p>寄存器存储系统内存中任务描述符列表头字节地址的 MSB 位 (63:32)。根据主机驱动的配置，任务描述符列表的大小为 32*(任务描述符大小+传输描述符大小)。这个地址设置在 1KB 的边界上。该寄存器的下 10 位被软件设置为 0，并被 CQE 忽略。当使用 32 位寻址模式时，这个寄存器是保留的。</p> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>

3.6.3.11 CQTDBR

- 寄存器名：命令队列门铃寄存器
- 描述：使用这个寄存器，软件触发 CQE 来处理一个新任务。
- 大小：32 位
- 偏移量：0x180 + 0x28

图表 3-103 CQTDBR 寄存器字段说明

位段	名称	类型	说明
31:0	DBR	R/W	<p>在使用该寄存器之前，软件配置 TDLBA 和 TDLBAU，并在 CQCFG 中使能 CQE。</p> <p>将 1 写入该寄存器的第 n 位触发 CQE，以开始处理编码在 TDL 的 n 号槽中的任务。软件写入 0 对硬件没有任何影响，也不会改变寄存器位的值。</p> <p>CQE 总是根据 CQTDBR 写事务提交给列表的顺序来处理任务。CQE 通过读取任务描述符来处理数据传输任务，并发送 QUEUED_TASK_PARAMS (CMD44) 和 QUEUED_TASK_ADDRESS (CMD45) 命令到该设备。CQE 通过读取任务描述符来处理 DCMD 任务（启用时位于 slot #31），并生成由其索引和参数编码的命令。</p> <p>当发生以下事件之一时，CQE 会将相应位清除为 0：</p> <ul style="list-style-type: none"> ■ 任务执行完成(成功或错误)。 ■ 使用 CQTCLR 寄存器清除任务。 ■ 使用 CQCTL 寄存器清除所有任务。 ■ 使用 CQCFG 寄存器禁用 CQE。 <p>软件可以通过在同一个事务中对这个寄存器的多个位写入 1 来同时发起多个任务(批量提交)。在批量提交的情况下，CQE 按照任务索引的顺序处理任务，从最低索引开始。如果批处理中的一个或多个任务被标记为 QBR，则执行的顺序将基于已述处理顺序。</p> <p>复位值：0x0</p> <p>存在：一直</p> <p>易失：是</p>

3.6.3.12 CQTCN

- 寄存器名：命令队列任务清除通知寄存器
- 描述：这个寄存器被 CQE 用来通知软件完成的任务。

- 大小：32 位
- 偏移量：0x180 + 0x2c

图表 3-104 CQTCN 寄存器字段说明

位段	名称	类型	说明
31:0	TCN	R/W1C	任务完成通知 32 位中的各个位被位映射到 32 个任务上。 <ul style="list-style-type: none"> ■ Bit-N(1): Task-N 已完成执行（成功或错误）。 ■ Bit-N(0): Task-N 还没有完成，可能是或挂起或未提交。 在任务完成时，软件可以读取这个寄存器来获知已经完成的 任务。在读取这个寄存器后，软件可以通过向相应位写入 1 来 清除相关位域。 复位值：0x0 存在：一直 易失：是

3.6.3.13 CQDQS

- 寄存器名：驱动队列状态寄存器
- 描述：这个寄存器存储设备队列状态的最新值。
- 大小：32 位
- 偏移量：0x180 + 0x30

图表 3-105 CQDQS 寄存器字段说明

位段	名称	类型	说明
31:0	DQS	R	设备队列状态 32 位中的各个位被位映射到 32 个任务上。 <ul style="list-style-type: none"> ■ Bit-N (1): 设备已经将任务 N 标记为准备执行 ■ Bit-N (0): 任务 N 还没有准备好执行。此任务可能在设备 中挂起或未提交。 主机控制器用设备队列状态命令的响应更新这个寄存器。 复位值：0x0 存在：一直

3.6.3.14 CQDPT

- 寄存器名：设备挂起任务寄存器

- 描述：该寄存器维护列队进入设备并等待执行完成的任务列表。
- 大小：32 位
- 偏移量：0x180 + 0x34

图表 3-106 CQDPT 寄存器字段说明

位段	名称	类型	说明
31:0	DPT	R	<p>设备挂起任务</p> <p>32 位中的各个位被映射到 32 个任务上。</p> <ul style="list-style-type: none"> ■ Bit-N (1): 任务 N 已经成功进入设备队列, 正在等待执行 ■ Bit-N (0): 任务 N 不在队列 <p>当 且 仅 当 UEUED_TASK_PARAMS (CMD44) 和 QUEUED_TASK_ADDRESS (CMD45) 为这个特定的任务被发送, 但任务没有被执行时, 设置该寄存器的第 n 位。控制器在收到 CMD45 的成功响应后设置此位。CQE 在任务完成执行后清除此位。</p> <p>软件在任务丢弃过程中读取这个寄存器, 以确定该任务是否在设备中排队。</p> <p>复位值: 0x0</p> <p>存在: 一直</p>

3.6.3.15 CQTCLR

- 寄存器名：命令队列门铃寄存器
- 描述：这个寄存器用于删除 CQE 中未完成任务。只有当 CQE 处于停止状态时, 才必须使用该寄存器。
- 大小：32 位
- 偏移量：0x180 + 0x38

图表 3-107 CQTCLR 寄存器字段说明

位段	名称	类型	说明
31:0	TCLR	R/W	<p>写入 1 到该寄存器的第 n 位命令 CQE 清除软件先前发出的任务。</p> <p>该位只能在 CQE 处于停止状态时写入, 如 CQCFG 寄存器停止位所示。当软件在这个寄存器中写入 1 时, CQE 将该值更新为 1, 并开始清除与任务相关的数据结构。一旦清除操作完成, CQE 将清除 CQTCLR 和 CQTDDBR 中的位字段(设置值为 0)。软件必须轮询 CQTCLR, 直到它被清除, 以验证清除操作</p>

位段	名称	类型	说明
			已完成。 复位值：0x0 存在：一直 易失：是

3.6.3.16 CQSSC1

- 寄存器名：CQ 发送状态配置 1 寄存器
- 描述：这个寄存器用于删除 CQE 中未完成任务。寄存器控制何时发送 SEND_QUEUE_STATUS 命令。
- 大小：32 位
- 偏移量：0x180 + 0x40

图表 3-108 CQSSC1 寄存器字段说明

位段	名称	类型	说明
31:20	RSVD_20_31	R	CQSSC1 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
19:16	SQSCMD_BLK_CNT	R/W	此字段表示在数据传输过程中何时发送 SQS CMD。 值为“n”表示在传输数据块 BLOCK_CNTn 时，CQE 在 CMD 线上发送的状态命令，在数据行，BLOCK_CNT 是当前事务的块数。 <ul style="list-style-type: none"> ■ 0x0: SEND_QUEUE_STATUS (CMD13)命令在事务期间未发送。相反，它只在数据线空闲时发送。 ■ 0x1: SEND_QUEUE_STATUS 命令将在事务的最后一个块期间发送。 ■ 0x2: 当最后两个块挂起时，执行 SEND_QUEUE_STATUS 命令。 ■ 0x3: 当最后 3 个块挂起时，执行 SEND_QUEUE_STATUS 命令。 ■ ■ 0xf: 当最后 15 个块挂起时，执行 SEND_QUEUE_STATUS 命令。 仅当 CQCFG.CQ_EN 是“0”时，进行编程。

位段	名称	类型	说明
			复位值: 0x0 存在: 一直
15:0	SQSCMD_IDLE_TMR	R/W	<p>该字段配置在使用周期性 SEND_QUEUE_STATUS (CMD13) 轮询时使用的轮询周期。</p> <p>当设备中有任务等待处理, 但没有数据传输正在进行时, 使用周期性轮询。当一个 SEND_QUEUE_STATUS 响应表示没有任务准备好执行, CQE 计算配置的时间, 直到它发出下一个 SEND_QUEUE_STATUS。</p> <p>计时器单位是时钟的时钟周期, 其频率在内部定时器时钟频率字段 CQCAP 寄存器中指定。最小值为 0001h(1 个时钟周期), 最大值为 FFFFh(65535 个时钟周期)。</p> <p>例如, CQCAP 字段值 0 表示 19.2 MHz 时钟频率(周期= 52.08 ns)。如果 CQSSC1.CIT 中的设置为 1000h, 计算出的轮询周期为 4096*52.08 ns= 213.33 ns。</p> <p>仅当 CQCFG.CQ_EN 是 “0” 时, 进行编程</p> <p>复位值: 0x1000 存在: 一直</p>

3.6.3.17 CQSSC2

- 寄存器名: CQ 发送状态配置 2 寄存器
- 描述: 该寄存器用于在 SEND_QUEUE_STATUS 命令参数中配置 RCA 字段。
- 大小: 32 位
- 偏移量: 0x180 + 0x44

图表 3-109 CQSSC2 寄存器字段说明

位段	名称	类型	说明
31:16	RSVD_16_31	R	<p>CQSSC2 寄存器的这些位是保留的。它们总是返回 0。</p> <p>复位值: 0x0 存在: 一直</p>
15:0	SQSCMD_RCA	R/W	<p>该字段向 CQE 提供 SEND_QUEUE_STATUS (CMD13) 命令参数中 16 位 RCA 字段的内容。</p> <p>当发送 SEND_QUEUE_STATUS (CMD13) 命令时, CQE 将该字段复制到参数的 31:16 位。</p> <p>复位值: 0x0</p>

位段	名称	类型	说明
			存在：一直

3.6.3.18 CQCRDCT

- 寄存器名：直接命令的命令响应寄存器
- 描述：这个寄存器存储上次执行的 DCMD 的响应。
- 大小：32 位
- 偏移量：0x180 + 0x48

图表 3-110 CQCRDCT 寄存器字段说明

位段	名称	类型	说明
31:0	DCMD_RESP	R	该寄存器包含由上次发送的直接命令(DCMD)任务生成的命令的响应。 只有在控制器清除 CQTDBR 寄存器的第 31 位后，该寄存器的内容才有效。 复位值：0x0 存在：一直

3.6.3.19 CQRMEM

- 寄存器名：命令响应模式错误屏蔽寄存器
- 描述：这个寄存器控制响应错误检测(RED)中断的产生。只有在这里启用的位才能为 RED 服务。
- 大小：32 位
- 偏移量：0x180 + 0x50

图表 3-111 CQRMEM 寄存器字段说明

位段	名称	类型	说明
31:0	RESP_ERR_MASK	R/W	该字段的位被映射到设备响应。该位在 R1/R1b 响应中被用作设备状态字段的的中断掩码。 <ul style="list-style-type: none"> ■ 1：当接收到 R1/R1b 响应时，在设备状态设置一个位 i，产生一个响应错误检测（RED）中断。 ■ 0：当收到 R1/R1b 响应时，忽略设备状态中的位 i。 这个寄存器的复位值被设置为在设备状态的所有“错误”类型位上触发中断。 注意：对 CMD13（SQS）的响应编码了 QSR，因此它们会被这个逻辑忽略。

位段	名称	类型	说明
			复位值: 0xfdf9a080 存在: 一直

3.6.3.20 CQTERRI

- 寄存器名: CQ 任务错误信息寄存器
- 描述: 当与任务活动相关的数据或命令发生错误时, CQE 会更新该寄存器。当 CQE 检测到或 eMMC 控制器指出错误时, CQE 在 CQTERRI 寄存器中存储以下信息: 错误发生时在命令行和数据行上执行的任务 ID 和命令索引。软件必须在错误恢复过程中使用此信息。
- 大小: 32 位
- 偏移量: 0x180 + 0x54

图表 3-112 CQTERRI 寄存器字段说明

位段	名称	类型	说明
31	TRANS_ERR_FIELDS_VALID	R	当数据传输事务正在进行时检测到错误时, 该位将被更新。 值: <ul style="list-style-type: none"> ■ 0x1 (SET): 检测到数据传输相关的错误。检查 TRANS_ERR_TASKID 和 TRANS_ERR_CMD_INDX 字段的内容 ■ 0x0 (NOT_SET): 忽略 TRANS_ERR_TASKID 和 TRANS_ERR_CMD_INDX 的内容。 复位值: 0x0 存在: 一直
30:29	RSVD_30_29	R	CQTERRI 寄存器的这些位(RSVD_30_29)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
28:24	TRANS_ERR_TASKID	R	此字段捕获已执行且其数据传输有错误的任务 ID。 复位值: 0x0 存在: 一直
23:22	RSVD_23_22	R	CQTERRI 寄存器的这些位(RSVD_23_22)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直

位段	名称	类型	说明
21:16	TRANS_ERR_CMD_INDX	R	该字段捕获已执行且其数据传输有错误的命令索引。 复位值：0x0 存在：一直
15	RESP_ERR_FIELDS_VALID	R	当执行命令事务检测到错误时，该位将被更新。 值： <ul style="list-style-type: none"> ■ 0x1 (SET)：检测到响应相关的错误。检查 RESP_ERR_TASKID 和 RESP_ERR_CMD_INDX 字段的内容。 ■ 0x0 (NOT_SET)：忽略 RESP_ERR_TASKID 和 RESP_ERR_CMD_INDX 的内容。 复位值：0x0 存在：一直
14:13	RSVD_13_14	R	CQTERRI 寄存器的这些位(RSVD_13_14)是保留的。它们总是返回 0。 复位值：0x0 存在：一直
12:8	RESP_ERR_TASKID	R	此字段捕获错误发生时在命令行上执行的任务 ID。 复位值：0x0 存在：一直
7:6	RSVD_6_7	R	CQTERRI 寄存器的这些位(RSVD_6_7)是保留的。它们总是返回 0。 复位值：0x0 存在：一直
5:0	RESP_ERR_CMD_INDX	R	此字段捕获发生错误时在命令行上执行的命令索引。 复位值：0x0 存在：一直

3.6.3.21 CQCRI

- 寄存器名：CQ 命令响应索引
- 描述：这个寄存器存储最后接收到的命令响应的索引。
- 大小：32 位

- 偏移量：0x180 + 0x58

图表 3-113 CQCRI 寄存器字段说明

位段	名称	类型	说明
31:6	RSVD_31_6	R	CQCRI 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
5:0	CMD_RESP_INDX	R	最后一个命令响应的索引 该字段存储上次接收到的命令响应的索引。控制器在每次收到命令响应时更新该值。 复位值：0x0 存在：一直

3.6.3.22 CQCRA

- 寄存器名：CQ 命令响应参数寄存器
- 描述：这个寄存器存储最后接收到的命令响应的参数。
- 大小：32 位
- 偏移量：0x180 + 0x5c

图表 3-114 CQCRA 寄存器字段说明

位段	名称	类型	说明
31:0	CMD_RESP_ARG	R	最后一个命令响应的参数 该字段存储最后接收到的命令响应的参数。控制器在每次收到命令响应时更新该值。 复位值：0x0 存在：一直

3.6.4 DWC_mshc_map/DWC_mshc_phy_block 寄存器

3.6.4.1 PHY_CNFG

- 寄存器名：SD/eMMC PHY 通用配置
- 描述：SD/eMMC PHY 通用配置寄存器
- 大小：32 位
- 偏移量：0x300 + 0x0

图表 3-115 PHY_CNFG 寄存器字段说明

位段	名称	类型	说明
31:24			保留字段：是
23:20	PAD_SN	R/W	NMOS TX 驱动强度控制。所有 SD/eMMC pad 的通用配置。 复位值：0x0 存在：一直
19:16	PAD_SP	R/W	PMOS TX 驱动强度控制。所有 SD/eMMC pad 的通用配置。 复位值：0x0 存在：一直
15:2			保留字段：是
1	PHY_PWRGOOD	R	Phy 的电源指示状态在此捕获。在 state 事务之前确保这个值是“1”。 复位值：0x0 存在：一直 复位屏蔽：0x0
0	PHY_RSTN	R/W	PHY 的低电平有效复位控制，写“0”复位 PHY，写“1”使复位失效。 复位值：0x0 存在：一直

3.6.4.2 CMDPAD_CNFG

- 寄存器名：SD/eMMC PHY CMD/RESP PAD 设置
- 描述：SD/eMMC PHY 的命令/响应 PAD 设置在这里被控制
- 大小：16 位
- 偏移量：0x300 + 0x4

图表 3-116 CMDPAD_CNFG 寄存器字段说明

位段	名称	类型	说明
15:13			保留字段：是
12:9	TXSLEW_CTRL_N	R/W	N 型 CMD Pad 的 TX 回转控制 复位值：0x2 存在：一直

位段	名称	类型	说明
8:5	TXSLEW_CTRL_P	R/W	P 型 CMD Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
4:3	WEAKPULL_EN	R/W	CMD PAD 的上拉/下拉启用控制 值: <ul style="list-style-type: none"> 0x0 (DISABLED): 禁用上拉和下拉功能 0x1 (PULLUP): 启用弱上拉 0x2 (PULLDOWN): 启用弱下拉 0x3 (ILLEGAL): 不应该使用 复位值: 0x0 存在: 一直
2:0	RXSEL	R/W	PAD 接收器类型选择。控制 SD/eMMC PHY CMD PAD 的 RXSEL 值 复位值: 0x0 存在: 一直

3.6.4.3 DATPAD_CNFG

- 寄存器名: SD/eMMC PHY 数据 PAD 设置
- 描述: SD/eMMC PHY 的数据 PAD 设置控制在这里。所有数据 pads 的通用设置。
- 大小: 16 位
- 偏移量: 0x300 + 0x6

图表 3-117 DATPAD_CNFG 寄存器字段说明

位段	名称	类型	说明
15:13			保留字段: 是
12:9	TXSLEW_CTRL_N	R/W	N 型 DATA Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
8:5	TXSLEW_CTRL_P	R/W	P 型 DATA Pad 的 TX 回转控制 复位值: 0x2 存在: 一直

位段	名称	类型	说明
4:3	WEAKPULL_EN	R/W	DATA PADS 的上拉/下拉启用控件 值： <ul style="list-style-type: none"> 0x0 (DISABLED): 禁用上拉和下拉功能 0x1 (PULLUP): 启用弱上拉 0x2 (PULLDOWN): 启用弱下拉 0x3 (ILLEGAL): 不应该使用 复位值: 0x0 存在: 一直
2:0	RXSEL	R/W	PAD 接收器类型选择。控制 SD/eMMC PHY DATA PADS 的 RXSEL 值 复位值: 0x0 存在: 一直

3.6.4.4 CLKPAD_CNFG

- 寄存器名: SD/eMMC PHY Clock PAD 设置
- 描述: SD/eMMC PHY 的 CLK PAD 设置在这里被控制。
- 大小: 16 位
- 偏移量: 0x300 + 0x8

图表 3-118 CLKPAD_CNFG 寄存器字段说明

位段	名称	类型	说明
15:13			保留字段: 是
12:9	TXSLEW_CTRL_N	R/W	N 型 CLK Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
8:5	TXSLEW_CTRL_P	R/W	P 型 CLK Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
4:3	WEAKPULL_EN	R/W	CLK PAD 的上拉/下拉启用控件 值： <ul style="list-style-type: none"> 0x0 (DISABLED): 禁用上拉和下拉功能

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x1 (PULLUP): 启用弱上拉 0x2 (PULLDOWN): 启用弱下拉 0x3 (ILLEGAL): 不应该使用 复位值: 0x0 存在: 一直
2:0	RXSEL	R/W	PAD 接收器类型选择。控制 SD/eMMC PHY CLK PAD 的 RXSEL 值 复位值: 0x0 存在: 一直

3.6.4.5 STBPAD_CNFG

- 寄存器名: SD/eMMC PHY 选通 PAD 设置
- 描述: SD/eMMC PHY 的选通 PAD 设置控制在这里。
- 大小: 16 位
- 偏移量: 0x300 + 0xA

图表 3-119 STBPAD_CNFG 寄存器字段说明

位段	名称	类型	说明
15:13			保留字段: 是
12:9	TXSLEW_CTRL_N	R/W	N 型 Strobe Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
8:5	TXSLEW_CTRL_P	R/W	P 型 Strobe Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
4:3	WEAKPULL_EN	R/W	选通 PAD 的上拉/下拉启用控件 值: <ul style="list-style-type: none"> 0x0 (DISABLED): 禁用上拉和下拉功能 0x1 (PULLUP): 启用弱上拉 0x2 (PULLDOWN): 启用弱下拉 0x3 (ILLEGAL): 不应该使用

位段	名称	类型	说明
			复位值: 0x0 存在: 一直
2:0	RXSEL	R/W	PAD 接收器类型选择。控制 SD/eMMC PHY STROBE PAD 的 RXSEL 值 复位值: 0x0 存在: 一直

3.6.4.6 STNPAD_CNFG

- 寄存器名: SD/eMMC PHY RSTN PAD 设置
- 描述: SD/eMMC PHY 的 RSTN PAD 设置控制在这里。
- 大小: 16 位
- 偏移量: 0x300 + 0xC

图表 3-120 RSTNPAD_CNFG 寄存器字段说明

位段	名称	类型	说明
15:13			保留字段: 是
12:9	TXSLEW_CTRL_N	R/W	N 型 RST_N Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
8:5	TXSLEW_CTRL_P	R/W	P 型 RST_N Pad 的 TX 回转控制 复位值: 0x2 存在: 一直
4:3	WEAKPULL_EN	R/W	RST_N PAD(s)的上拉/下拉启用控件 值: <ul style="list-style-type: none"> ■ 0x0 (DISABLED): 禁用上拉和下拉功能 ■ 0x1 (PULLUP): 启用弱上拉 ■ 0x2 (PULLDOWN): 启用弱下拉 ■ 0x3 (ILLEGAL): 不应该使用 复位值: 0x0 存在: 一直
2:0	RXSEL	R/W	PAD 接收器类型选择。控制 SD/eMMC PHY RST_N PAD(s)

位段	名称	类型	说明
			的 RXSEL 值 复位值: 0x0 存在: 一直

3.6.4.7 PADTEST_CNFG

- 寄存器名: SD/eMMC PHY PAD TEST 接口设置
- 描述: PAD TEST 路径和方向控制
- 大小: 16 位
- 偏移量: 0x300 + 0xE

图表 3-121 PADTEST_CNFG 寄存器字段说明

位段	名称	类型	说明
15:10			保留字段: 是
9:4	TEST_OE	R/W	测试接口 OE 控件。驱动器的 PHY 的 itest_oe 输入。 复位值: 0x0 存在: 一直
3:1	RSVD_1	R	RSVD1 字段是保留的。 复位值: 0x0 存在: 一直
0	TESTMODE_EN	R/W	为所有 PADS 开启测试模式接口。功能接口禁用。 值: <ul style="list-style-type: none"> ■ 0x0 (PAD_FUNCMODE): PAD 的功能模式 I/F 激活 ■ 0x1 (PAD_TESTMODE): PAD 的测试模式接口激活 复位值: 0x0 存在: 一直

3.6.4.8 PADTEST_OUT

- 寄存器名: SD/eMMC PHY PAD TEST Data 输出值
- 描述: PAD TEST Path Data 输出, 驱动 SD/eMMC PHY itest_a 输入
- 大小: 16 位
- 偏移量: 0x300 + 0x10

图表 3-122 PADTEST_OUT 寄存器字段说明

位段	名称	类型	说明
15:6			保留字段：是
5:0	TESTDATA_OUT	R/W	这里写入的数据反映在相应的 itest_a 上。每个位被映射到 PHY 对应的 itest_a 输入。 复位值：0x0 存在：一直

3.6.4.9 PADTEST_IN

- 寄存器名：SD/eMMC PHY PAD TEST Data 输入值
- 描述：PAD TEST Path Data 输入，反映 SD/eMMC PHY 的 otest_y 输出的值。
- 大小：16 位
- 偏移量：0x300 + 0x12

图表 3-123 PADTEST_IN 寄存器字段说明

位段	名称	类型	说明
15:6			保留字段：是
5:0	TESTDATA_IN	R	这里的单个位捕获对应 otest_y 输出上的可用数据。应用于直流测试和低频率数据模式。 复位值：0x0 存在：一直

3.6.4.10 PRBS_CNFG

- 寄存器名：Controller PRBS 配置寄存器
- 描述：注册以配置 PRBS 引擎
- 大小：16 位
- 偏移量：0x300 + 0x18

图表 3-124 PRBS_CNFG 寄存器字段说明

位段	名称	类型	说明
15:0	INIT_SEED	R/W	这里编程的值用作 PRBS 引擎的 SEED 复位值：0xffff 存在：一直

3.6.4.11 PHYLPBK_CNFG

- 寄存器名：环回配置寄存器
- 描述：设置环回模式的寄存器
- 大小：8 位
- 偏移量：0x300 + 0x1A

图表 3-125 PHYLPBK_CNFG 寄存器字段说明

位段	名称	类型	说明
7:2			保留字段：是
1	OUT_EN_PHYLPBK_MODE	R/W	PHY 环回模式下使能 CMD/DATA 输出 值： <ul style="list-style-type: none"> ■ 0x1 (OUTPUT_EN): CMD/DATA 输出使能 PHY 环回模式 ■ 0x0 (OUTPUT_DIS): CMD/DATA 输出关闭 PHY 环回模式 复位值：0x0 存在：一直
0	PHYLPBK_EN	R/W	PHY 本地环回模式使能 值： <ul style="list-style-type: none"> ■ 0x1 (PHYLPBK_MODE_EN): 控制器当前处于 PHY 环回模式。 ■ 0x0 (PHYLPBK_MODE_DIS): 控制器未处于 PHY 环回模式。 复位值：0x0 存在：一直

3.6.4.12 COMMDL_CNFG

- 寄存器名：通用 DelayLine 配置设置寄存器
- 描述：配置寄存器，将 PHY 中使用的所有 DelayLines 设为通用设置
- 大小：8 位
- 偏移量：0x300 + 0x1C

图表 3-126 COMMDL_CNFG 寄存器字段说明

位段	名称	类型	说明
7:2			保留字段：是
1	DLOUT_EN	R/W	当'1' DL 输出可以在 PADS 上采样时。在所有 PADS 驱动 idlout_en。 值： ■ 0x1 (DL_OUTEN): PAD 启用 DelayLine 输出。 ■ 0x0 (DL_OUTDIS): PAD 禁用 DelayLine 输出。 复位值：0x0 存在：一直
0	DLSTEP_SEL	R/W	DelayLine 的每一步延迟选择，驱动 PHY 的 idl_step 输入 复位值：0x0 存在：一直

3.6.4.13 SDCLKDL_CNFG

- 寄存器名：SD/eMMC DelayLine 设置
- 描述：SD/eMMC CLK DelayLine 的设置
- 大小：8 位
- 偏移量：0x300 + 0x1D

图表 3-127 SDCLKDL_CNFG 寄存器字段说明

位段	名称	类型	说明
7:5			保留字段：是
4	UPDATE_DC	R/W	“1” 时准备 DelayLine 的代码更新。在写 SDCLKDL_DC 时，建议该位为 1。确保在不更新代码时为 “0”。 注意： 在该字段进行编程之前，使用 CLK_CTRL_R.SD_CLK_EN 关闭卡时钟。 值： ■ 0x1 (BYPASSMODE): DelayLine 的输出是 DelayLine 输出激活 ■ 0x0 (DLMODE): DelayLine 输出使能 复位值：0x0

位段	名称	类型	说明
			存在：一直
3:2	INPSEL_CNFG	R/W	驱动 SD/eMMC CLK DelayLine 的配置输入。值选择 DelayLine 的输入源 复位值：0x0 存在：一直
1	BYPASS_EN	R/W	驱动 SD/eMMC CLK DelayLine 的旁路输入 值： ■ 0x1 (BYPASSMODE): DelayLine 为旁路模式 ■ 0x0 (DLMODE): DelayLine 激活模式 复位值：0x0 存在：一直
0	EXTDLY_EN	R/W	驱动 SD/eMMC CLK DelayLine 的 extdlyen 输入 值： ■ 0x1 (EXTDL_MODE): DelayLine 可扩展延迟范围设置 ■ 0x0 (DEF_MODE): DelayLine 默认范围设置 复位值：0x0 存在：一直

3.6.4.14 SDCLKDL_DC

- 寄存器名：SD/eMMC DelayLine Delay 代码设置
- 描述：SD/eMMC CLK DelayLine Delay Code 值
- 大小：8 位
- 偏移量：0x300 + 0x1E

图表 3-128 SDCLKDL_DC 寄存器字段说明

位段	名称	类型	说明
7			保留字段：是
6:0	CCKDL_DC	R/W	驱动 SD/eMMC CLK DelayLine 的延迟码入。此处值选择卡片时钟延迟线中活跃阶段的数量。 注意：在对该字段进行编程之前，使用 CLK_CTRL_R.SD_CLK_EN 关闭卡时钟。 复位值：0x0

位段	名称	类型	说明
			存在：总一直

3.6.4.15 SMPLDL_CNFG

- 寄存器名：SD/eMMC cclk_rx DelayLine 设置
- 描述：SD/eMMC cclk_rx DelayLine 配置设置
- 大小：8 位
- 偏移量：0x300 + 0x20

图表 3-129 SMPLDL_CNFG 寄存器字段说明

位段	名称	类型	说明
7:5			保留字段：是
4	INPSEL_OVERRIDE	R/W	PHY 的采样 delay line 配置由控制器使用 sample_cclk_sel 控制，该信号覆盖 sample_cclk_sel，这样 INPSEL_CFG 字段直接控制 PHY 的配置输入。 ■ 0x0：控制器逻辑驱动器采样 delay line 配置。 ■ 0x1：SMPLDL_CNFG.INPSEL_CNFG 驱动采样 delay line 配置 复位值：0x0 存在：一直
3:2	INPSEL_CNFG	R/W	驱动 CCLK_RX DelayLine 的配置输入。这里的值选择 DelayLine 的输入源 复位值：0x3 存在：一直
1	BYPASS_EN	R/W	驱动 CCLK_RX DelayLine 的旁路输入 值： ■ 0x1 (BYPASSMODE)：DelayLine 为旁路模式 ■ 0x0 (DLMODE)：Delay line 激活模式 复位值：0x1 存在：一直
0	EXTDLY_EN	R/W	驱动 CCLK_RX DelayLine 的 extdlyen 输入 值： ■ 0x1 (EXTDL_MODE)：DelayLine 可扩展延迟范围设置

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x0 (DEF_MODE): Delay line 默认范围设置 复位值: 0x1 存在: 一直

3.6.4.16 ATDL_CNFG

- 寄存器名: SD/eMMC drift_cclk_rx DelayLine 配置设置
- 描述: SD/eMMC drift_cclk_rx DelayLine 配置设置
- 大小: 8 位
- 偏移量: 0x300 + 0x21

图表 3-130 ATDL_CNFG 寄存器字段说明

位段	名称	类型	说明
7:4			保留字段: 是
3:2	INPSEL_CNFG	R/W	驱动 drift_cclk_rx DelayLine 的配置输入。这里的值选择 DelayLine 的输入源 复位值: 0x0 存在: 一直
1	BYPASS_EN	R/W	驱动 drift_cclk_rx DelayLine 的旁路输入 值: <ul style="list-style-type: none"> 0x1 (BYPASSMODE): DelayLine 为旁路模式 0x0 (DLMODE): Delay line 激活模式 复位值: 0x0 存在: 一直
0	EXTDLY_EN	R/W	驱动 drift_cclk_rx DelayLine 的 extdlyen 输入 值: <ul style="list-style-type: none"> 0x1 (EXTDL_MODE): DelayLine 适用于扩展延迟范围设置 0x0 (DEF_MODE): Delay line 默认范围设置 复位值: 0x0 存在: 一直

3.6.4.17 DLL_CTRL

- 寄存器名：SD/eMMC PHY DLL 控件设置
- 描述：SD/eMMC PHY 的 DLL 控件设置寄存器
- 大小：8 位
- 偏移量：0x300 + 0x24

图表 3-131 DLL_CTRL 寄存器字段说明

位段	名称	类型	说明
7:3	RSVD_3_7	R	寄存器的这些位是保留的。 复位值：0x0 存在：一直
2	SLV_SWDC_UPDATE	R/W	相应输出驱动 PHY 的 DLL Slave 的 dc 更新输入。当使用 DLL_OFFST 寄存器改变延迟码时，这用于关闭从延迟线 (Slave Delay line) 的输出。 值： <ul style="list-style-type: none"> ■ 0x1 (DL_OUT_OFF): 正在更新。 ■ 0x0 (DL_OUT_ON): 更新完成。 复位值：0x0 存在：一直
1	OFFST_EN	R/W	启用 DLL 时 PHY 的偏移模式启用。当 DLL 被禁用时，这允许直接控制 DLL 的 Slave 产生的延迟 值： <ul style="list-style-type: none"> ■ 0x1 (OFFSTEN): 偏移值有效。 ■ 0x0 (OFFSTDIS): 偏移值无效。 复位值：0x0 存在：一直
0	DLL_EN	R/W	当“1”时启用 DLL 值： <ul style="list-style-type: none"> ■ 0x1 (DLLENABLE): PHY DLL 启用 ■ 0x0 (DLLDISABLE): PHY DLL 禁用 复位值：0x0 存在：一直

3.6.4.18 DLL_CNFG1

- 寄存器名：DLL 配置寄存器 1
- 描述：SD/eMMC PHY DLL 配置寄存器 1
- 大小：8 位
- 偏移量：0x300 + 0x25

图表 3-132 DLL_CNFG1 寄存器字段说明

位段	名称	类型	说明
7:6			保留字段：是
5:4	SLVDLY	R/W	设置 DLL slave 的更新延迟输入值为 islv_update_dly 复位值：0x0 存在：一直
3			保留字段：是
2:0	WAITCYCLE	R/W	设置 DLL 的等待周期输入的值 复位值：0x0 存在：一直

3.6.4.19 DLL_CNFG2

- 寄存器名：DLL 配置寄存器 2
- 描述：SD/eMMC PHY DLL 配置寄存器 2
- 大小：8 位
- 偏移量：0x300 + 0x26

图表 3-133 DLL_CNFG2 寄存器字段说明

位段	名称	类型	说明
7			保留字段：是
6:0	JUMPSTEP	R/W	设置 DLL 的跳步输入的值 复位值：0x0 存在：一直

3.6.4.20 DLLDL_CNFG

- 寄存器名：DLL 配置寄存器 2
- 描述：SD/eMMC PHY DLL MST & Slave DL 配置设置

- 大小：8 位
- 偏移量：0x300 + 0x28

图表 3-134 DLLDL_CNFG 寄存器字段说明

位段	名称	类型	说明
7	SLV_BYPASS	R/W	Slave DL 的旁路启用控制 复位值：0x0 存在：一直
6:5	SLV_INPSEL	R/W	为 Slave DL 选择时钟源 复位值：0x0 存在：一直
4	SLV_EXTDLYEN	R/W	使能 Slave 的扩展延迟模式 复位值：0x0 存在：一直
3	MST_BYPASS	R/W	主 DL 的旁路启用控制 复位值：0x0 存在：一直
2:1	MST_INPSEL	R/W	为主 DL 选择时钟源 复位值：0x0 存在：一直
0	MST_EXTDLYEN	R/W	为 Master 启用扩展延迟模式 复位值：0x0 存在：一直

3.6.4.21 DLL_OFFST

- 寄存器名：DLL Offset 设置寄存器
- 描述：SD/eMMC PHY DLL 偏移值设置
- 大小：8 位
- 偏移量：0x300 + 0x29

图表 3-135 DLL_OFFST 寄存器字段说明

位段	名称	类型	说明
7			保留字段：是

位段	名称	类型	说明
6:0	OFFST	R/W	设置 DLL 的偏移量输入值 复位值：0x0 存在：一直

3.6.4.22 DLLMST_TSTDC

- 寄存器名：DLL 主测试代码设置寄存器
- 描述：SD/eMMC PHY DLL 主测试延时码寄存器
- 大小：8 位
- 偏移量：0x300 + 0x2A

图表 3-136 DLLMST_TSTDC 寄存器字段说明

位段	名称	类型	说明
7			保留字段：是
6:0	MSTTST_DC	R/W	当 DLL 被禁用时，设置 DLL 的主测试代码输入值。 复位值：0x0 存在：一直

3.6.4.23 DLLLBT_CNFG

- 寄存器名：DLL LBT 设置寄存器
- 描述：SD/eMMC PHY DLL 低带宽定时器配置寄存器
- 大小：16 位
- 偏移量：0x300 + 0x2C

图表 3-137 DLLLBT_CNFG 寄存器字段说明

位段	名称	类型	说明
15:0	LBT_LOADVAL	R/W	设置 DLL 的 olbt_loadal 输入值。控制 DLL 在其上运行重新验证周期的 lbt 计时器超时值。 复位值：0x0 存在：一直

3.6.4.24 DLL_STATUS

- 寄存器名：DLL 状态寄存器
- 描述：SD/eMMC PHY DLL 状态寄存器

- 大小：8 位
- 偏移量：0x300 + 0x2E

图表 3-138 DLL_STATUS 寄存器字段说明

位段	名称	类型	说明
7:2			保留字段：是
1	ERROR_STS	R	捕获 DLL 的锁错误状态信息的值。只有设置了 LOCK_STS 参数时，该值才有效。 <ul style="list-style-type: none"> ■ 如果 LOCK_STS = 1 和 ERR_STS = 0，则 DLL 被锁定，不会产生错误 ■ 如果 LOCK_STS = 1 和 ERR_STS = 1，则 DLL 被锁定为默认值并有错误。这个阶段的事务可能失败。 值： <ul style="list-style-type: none"> ■ 0x1 (DLL_ERROR)：DLL 已锁定并准备就绪。 ■ 0x0 (DLL_LOCK_OKAY)：DLL 未被锁定。 复位值：0x0 存在：一直
0	LOCK_STS	R	捕获 DLL 的锁状态信息的值 值： <ul style="list-style-type: none"> ■ 0x1 (DLL_IS_LOCKED)：DLL 已锁定并准备就绪。 ■ 0x0 (DLL_NOT_LOCKED)：DLL 未被锁定。 复位值：0x0 存在：一直

3.6.4.25 DLLDBG_MLKDC

- 寄存器名：DLL 主程序锁定的代码调试寄存器
- 描述：SD/eMMC PHY DLL 的主程序锁定的代码状态
- 大小：8 位
- 偏移量：0x300 + 0x30

图表 3-139 DLLDBG_MLKDC 寄存器字段说明

位段	名称	类型	说明
7			保留字段：是
6:0	MSTLKDC	R	捕获 DLL 的主程序锁定的延迟码值

位段	名称	类型	说明
			复位值：0x0 存在：一直

3.6.4.26 DLLDBG_SLKDC

- 寄存器名：DLL 主从代码调试寄存器
- 描述：SD/eMMC PHY DLL 的 Slave 锁代码状态
- 大小：8 位
- 偏移量：0x300 + 0x32

图表 3-140 DLLDBG_SLKDC 寄存器字段说明

位段	名称	类型	说明
7			保留字段：是
6:0	SLVLKDC	R	捕获 DLL 的 Slave 已锁定的延迟码值 复位值：0x0 存在：一直

3.6.5 DWC_mshc_map/DWC_mshc_vendor1_block 寄存器

3.6.5.1 MSHC_VER_ID_R

- 寄存器名：MSHC version
- 描述：这个寄存器反映了 DWC_mshc 的当前版本号。
- 大小：32 位
- 偏移量：0x500

图表 3-141 MSHC_VER_ID_R 寄存器字段说明

位段	名称	类型	说明
31:0	MSHC_VER_ID	R	当前版本号 该字段表示应用程序读取的 Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite 当前版本号。 一个应用程序与 MSHC_VER_TYPE_R 寄存器一起读取该寄存器，收集当前版本的详细信息。 复位值：SNPS_RSVDPARAM_5 存在：一直

3.6.5.2 MSHC_VER_TYPE_R

- 寄存器名：MSHC 版本类型
- 描述：这个寄存器反映了 DWC_mshc 的当前版本类型
- 大小：32 位
- 偏移量：0x500 + 0x4

图表 3-142 MSHC_VER_TYPE_R 寄存器字段说明

位段	名称	累心	说明
31:0	MSHC_VER_TYPE	R	当前版本类型 该字段表示应用程序读取的 Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite 当前发布类型。 一个应用程序与 MSHC_VER_ID_R 寄存器一起读取该寄存器，收集当前版本的详细信息。 复位值：SNPS_RSVDPARAM_6 存在：一直

3.6.5.3 MSHC_CTRL_R

- 寄存器名：MSHC 控制寄存器
- 描述：这个寄存器用于控制 MSHC 主机控制器的操作。
- 大小：8 位
- 偏移量：0x500 + 0x8

图表 3-143 MSHC_CTRL_R 寄存器字段说明

位段	名称	类型	说明
7	Reserved	R	
6	Reserved	R	
5			保留字段：是
4	SW_CG_DIS	R/W	内部时钟门控禁用控制 当需要时，该位必须用于禁用 IP 的内部时钟门控。当被禁用时，时钟没有门控。在编程时，核心的时钟(除了 hclk)必须停止。 值： <ul style="list-style-type: none"> ■ 0x0 (ENABLE): 内部时钟门是激活的，时钟门控是内部控制的。

位段	名称	类型	说明
			<p>■ 0x1 (DISABLE): 内部时钟门控被禁用, 时钟没有内部门控。</p> <p>复位值: 0x0</p> <p>存在: 一直</p>
3:1	RSVD1	R	<p>MSHC_CTRL_R 寄存器的这些位(RSVD1)是保留的。它们总是返回 0。</p> <p>复位值: 0x0</p> <p>存在: 一直</p>
0	CMD_CONFLICT_CHECK	R/W	<p>命令冲突检查</p> <p>此位启用命令冲突检查。</p> <p>注意: 每当发出命令时, DWC_mshc 控制器监控 CMD 线, 并检查 sd_cmd_out 上驱动的值是否与 cclk_tx 的下一个后续边 sd_cmd_in 上的值匹配, 以确定命令冲突错误。只有反馈延迟 (包括 IO Pad 延迟) 大于 ($t_{card_clk_period} - t_{setup}$) 时, 该位才会被清除, 其中 t_{setup} 是 DWC_mshc 中触发的设置时间。I/O pad 延迟与 CMD 线和 DATA 行一致, 其值的范围为:</p> <p>$(2 * t_{card_clk_period} - t_{setup})$</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x0 (DISABLE_CMD_CONFLICT_CHK): 关闭命令冲突检查 ■ 0x1 (CMD_CONFLICT_CHK_LAT1): 在 1 个卡时钟周期后检查命令冲突 <p>复位值: 0x1</p> <p>存在: 一直</p>

3.6.5.4 MBIU_CTRL_R

- 寄存器名: MBIU 控制寄存器
- 描述: 该寄存器用于选择 AHB 主总线接口可以生成的有效突发类型。当设置超过一个位时, 主机在寄存器中选择已启用的优先的突发。
- 大小: 8 位
- 偏移量: 0x500 + 0x10

图表 3-144 MBIU_CTRL_R 寄存器字段说明

位段	名称	类型	说明
7:4	RSVD	R	保留字段 复位值: 0x0 存在: 一直
3	BURST_INCR16_EN	R/W	INCR16 突发 在主接口上控制 INCR16 传输的生成。 值: <ul style="list-style-type: none"> 0x0 (FALSE): 主 I/F 上不产生 AHB INCR16 突发类型。 0x1 (TRUE): AHB INCR16 突发类型可以在 Master I/F 上产生。 复位值: 1 存在: 一直
2	BURST_INCR8_EN	R/W	INCR8 突发 在主接口上控制 INCR8 传输的生成。 值: <ul style="list-style-type: none"> 0x0 (FALSE): 主 I/F 上不产生 AHB INCR8 突发类型。 0x1 (TRUE): AHB INCR8 发类型可以在 Master I/F 上产生。 复位值: 1 存在: 一直
1	BURST_INCR4_EN	R/W	INCR4 突发 在主接口上控制 INCR4 传输的生成。 值: <ul style="list-style-type: none"> 0x0 (FALSE): 主 I/F 上不产生 AHB INCR4 突发类型。 0x1 (TRUE): AHB INCR4 发类型可以在 Master I/F 上产生。 复位值: 1 存在: 一直
0	UNDEFL_INCR_EN	R/W	未定义的 INCR 突发 控制在主接口上生成未定义长度的 INCR 传输。 值:

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x0 (FALSE): 未定义 INCR 型突发是 AHB 主 I/F 上最首选的突发。 0x1 (TRUE): 未定义 INCR 类型的突发是 AHB 主 I/F 上最首选的突发。 复位值: DWC_MSHC_MBIU_UNDEFLBURSTEN 存在: 一直

3.6.5.5 EMMC_CTRL_R

- 寄存器名: eMMC 控制寄存器
- 描述: 这个寄存器用于控制 eMMC 操作。
- 大小: 16 位
- 偏移量: 0x500 + 0x2c

图表 3-145 EMMC_CTRL_R 寄存器字段说明

位段	名称	类型	说明
15:11	RSVD	R	EMMC_CTRL_R 寄存器的这些位(RSVD)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
10	CQE_PREFETCH_DISABLE	R/W	开启/关闭 CQE 的预取特性 当设置为 1 时, 允许软件关闭 CQE 的数据预取功能。 值: <ul style="list-style-type: none"> 0x0 (PREFETCH_ENABLE): CQE 可以对连续的 WRITE 传输进行预取, 对连续的 READ 传输进行流水线处理 0x1 (PREFETCH_DISABLE): WRITE 预取和 READ 流水线实现的功能被禁用 复位值: 0x0 存在: 一直
9	CQE_ALGO_SEL	R/W	选择执行的排程算法 这个位选择算法, 用于选择要执行的多个就绪任务中的一个。 值: <ul style="list-style-type: none"> 0x0 (PRI_REORDER_PLUS_FCFS): 基于优先级的先到先

位段	名称	类型	说明
			做重新排序，以解决同等优先级的任务。 ■ 0x1 (FCFS_ONLY): 先到先得，按 DBR 环形的顺序。 复位值: 0x0 存在: 一直
8	ENH_STROBE_ENABLE	R/W	增强选通启用 该位指示 DWC_mshc 在 HS400 模式下使用数据选通采样 CMD 线。 值: ■ 0x1 (ENH_STB_FOR_CMD): HS400 模式使用数据选通对 CMD 进行采样。 ■ 0x0 (NO_STB_FOR_CMD): HS400 模式使用 cclk_rx 对 CMD 线进行采样。 复位值: 0x0 存在: 一直
7:4			保留字段: 是
3	EMMC_RST_N_OE	R/W	EMMC 设备复位信号 PAD 控制的输出使能控制。 该字段驱动 DWC_mshc 的 sd_rst_n_oe 输出 值: ■ 0x1 (ENABLE): sd_rst_n_oe 为 1。 ■ 0x0 (DISABLE): sd_rst_n_oe 为 0。 复位值: 0x1 存在: 一直
2	EMMC_RST_N	R/W	EMMC 设备复位信号控制 这个寄存器字段控制 DWC_mshc 值的 sd_rst_n 输出 值: ■ 0x1 (RST_DEASSERT): eMMC 设备的复位失效。 ■ 0x0 (RST_ASSERT): eMMC 设备的复位生效（低电平有效）。 复位值: 0x1 存在: 一直
1	DISABLE_DATA_CRC_CHK	R/W	禁用数据 CRC 检测

位段	名称	类型	说明
			该位控制 eMMC 模式下卡写入 CRC16 错误的屏蔽。这在 eMMC 设备的总线测试(CMD19)中很有用。在总线测试中，eMMC 卡不发送块的 CRC 状态，可能会产生 CRC 错误。在总线测试期间，可以使用此位屏蔽 CRC 错误。 值： <ul style="list-style-type: none"> 0x1 (DISABLE): 禁用 DATA CRC 检测。 0x0 (ENABLE): 启用 DATA CRC 检测。 复位值: 0x0 存在: 总是
0	CARD_IS_EMMC	R/W	eMMC 有卡 此位表示所连接卡的类型。此位的基于一个应用程序将卡连接到 MSHC。 值： <ul style="list-style-type: none"> 0x1 (EMMC_CARD): 连接到 MSHC 的卡是 eMMC 卡。 0x0 (NON_EMMC_CARD): 连接到 MSHC 的 Card 非 eMMC 卡。 复位值: 0x0 存在: 一直

3.6.5.6 BOOT_CTRL_R

- 寄存器名: eMMC 启动控制寄存器
- 描述: 这个寄存器用于控制 eMMC 启动操作。
- 大小: 16 位
- 偏移量: 0x500 + 0x2e

图表 3-146 BOOT_CTRL_R 寄存器字段说明

位段	名称	类型	说明
15:12	BOOT_TOUT_CNT	R/W	引导确认超时计数器值 在启动操作期间, 当预期出现引导确认时, 这个值决定了引导确认超时检测的时间间隔(50 毫秒)。 <ul style="list-style-type: none"> 0xF: 保留 0xE: $TMCLK \times 2^{27}$.. -

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x1: TMCLK x 2¹⁴ ■ 0x0: TMCLK x 2¹³ 复位值: 0x0 存在: 一直
11:9	RSVD_11_9	R	BOOT_CTRL_R 寄存器的这些位(RSVD_11_9)是保留的。它们总是返回 0。 复位值: 0x0 存在: 一直
8	BOOT_ACK_ENABLE	R/W	引导确认使能 设置此位后, DWC_mshc 在启动操作期间, 检查 0-1-0 的引导确认开启模式。此位既适用于强制启动模式, 也适用于备用启动模式。 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 引导确认使能 ■ 0x0 (FALSE): 引导确认禁用 复位值: 0x0 存在: 一直
7	VALIDATE_BOOT	W	验证强制启动使能位 该位用于验证 MAN_BOOT_EN 位。 值: <ul style="list-style-type: none"> ■ 0x1 (TRUE): 验证强制启动使能位 ■ 0x0 (FALSE): 忽略强制启动使能位 复位值: 0x0 存在: 一直
6:1	RSVD_6_1	R	BOOT_CTRL_R 寄存器的这些位(RSVD_6_1)是保留的。它们总是返回 0。 复位值: 0x0 存在: 总是
0	MAN_BOOT_EN	R/W	强制启动使能 该位用于发起强制启动操作。应用程序将该位与 VALIDATE_BOOT 位一起设置。写入 0 将被忽略。DWC_mshc

位段	名称	类型	说明
			在启动传输完成或终止后清除该位。 值： <ul style="list-style-type: none"> 0x1 (MAN_BOOT_EN): 强制启动使能 0x0 (MAN_BOOT_DIS): 强制启动禁用 复位值: 0x0 存在: 一直 测试: 只读

3.6.5.7 AT_CTRL_R

- 寄存器名: 调优和自动调优控制寄存器
- 描述: 这个寄存器控制调优和自动调优特性的某些方面。当 HOST_CTRL2_R.SAMPLE_CLK_SEL=1 时, 不要对这个寄存器进行编程。
- 大小: 32 位
- 偏移量: 0x500 + 0x40

图表 3-147 AT_CTRL_R 寄存器字段说明

位段	名称	类型	说明
30:24	SWIN_TH_VAL	R/W	采样窗口阈值设置 这里可以设置的最大值取决于用于调优的 delayline 的长度。具有 128 抽头的 delayLine 可以使用从 0x0 到 0x7F 的值。 当 SWIN_TH_EN 为 “1” 时, 该字段才有效。仅当 SAMPLE_CLK_SEL 为 “0” 时才可以编程 <ul style="list-style-type: none"> 0x0: 阈值为 0x1, 可选择长度为 1 抽头及以上的窗口作为采样窗口。 0x1: 阈值为 0x2, 可选择长度为 2 抽头及以上的窗口作为采样窗口。 0x2: 阈值为 0x1, 可选择长度为 3 抽头及以上的窗口作为采样窗口。 0x7F: 阈值为 0x1, 可选择长度为 127 抽头及以上的窗口作为采样窗口。 复位值: 3
23:21			保留字段: 是

位段	名称	类型	说明
20:19	POST_CHANGE_DLY	R/W	<p>相位切换和稳定时钟输出所需的时间。</p> <p>指定在 tuning_cclk_sel 或 autotuning_cclk_sel 发生变化后，延迟线切换其输出相位的最大时间(以时钟周期为单位)。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (LATENCY_LT_1): 小于 1 个周期的延迟 ■ 0x1 (LATENCY_LT_2): 小于 2 个周期的延迟 ■ 0x2 (LATENCY_LT_3): 小于 3 个周期的延迟 ■ 0x3 (LATENCY_LT_4): 小于 4 个周期的延迟 <p>复位值: 0x0</p>
18:17	PRE_CHANGE_DLY	R/W	<p>cclk_tx 和 cclk_rx 之间的最大延迟规格。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (LATENCY_LT_1): 小于 1 个周期的延迟 ■ 0x1 (LATENCY_LT_2): 小于 2 个周期的延迟 ■ 0x2 (LATENCY_LT_3): 小于 3 个周期的延迟 ■ 0x3 (LATENCY_LT_4): 小于 4 个周期的延迟 <p>复位值: 0x0</p>
16	TUNE_CLK_STOP_EN	R/W	<p>时钟停止对调优和自动调优电路的控制。</p> <p>当启用时，在 tuning_cclk_sel 和 autotuning_cclk_sel 上改变相位选择代码之前，DWC_mshc (clk2card_on)的时钟门控输出被拉低。这有效地停止了 Device/Card 时钟，cclk_rx 和 drift_cclk_rx。当时钟停止时改变相位代码确保无故障的相位切换。如果 PHY 或延迟线可以保证无故障开关，则将该位设为 0。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (ENABLE_CLK_STOPPING): 相位代码改变时，时钟停止 ■ 0x0 (DISABLE_CLK_STOPPING): 时钟未停止。PHY 确保无故障相位开关。 <p>复位值: 0x0</p>
15:12	RSDV3	R	<p>AT_CTRL_R 寄存器的这些位(RSDV3)是保留的。它们总是返回 0。</p> <p>复位值: 0x0</p>

位段	名称	类型	说明
			存在：一直
11:8	WIN_EDGE_SEL	R/W	<p>此字段设置用于漂移监控的左边缘和右边缘的相位。[左边缘偏移量+右边缘偏移量]必须不小于 delayLine 的总抽头数。</p> <ul style="list-style-type: none"> ■ 0x0: 禁用用户选择。使用优化计算后的边缘。 ■ 0x1: 右边缘相位为中心+ 2 阶段, 左边缘相位为中心- 2 阶段。 ■ 0x2: 右边缘相位为中心+ 3 阶段, 左边缘相位为中心- 3 阶段。 ■ ... ■ 0xF: 右边缘相位为中心+16 阶段, 左边缘相位为中心- 16 阶段。 <p>复位值: 0x0</p> <p>存在：一直</p>
7:5	RSDV2	R	<p>AT_CTRL_R 寄存器的这些位 (RSDV2) 是保留的。它们总是返回 0。</p> <p>复位值: 0x0</p> <p>存在：一直</p>
4	SW_TUNE_EN	R/W	<p>该字段支持软件管理的调优流。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x1 (SW_TUNING_ENABLE): 启用软件管理调优。AT_STAT_R. CENTER_PH_CODE 字段现在是可写的。 ■ 0x0 (SW_TUNING_DISABLE): 禁用软件管理调优。 <p>复位值: 0x0</p> <p>存在：一直</p>
3	RPT_TUNE_ERR	R/W	<p>执行调优时不会生成帧错误。这个调试位允许用户报告这些错误。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x1 (DEBUG_ERRORS): 报告帧错误的调试模式 ■ 0x0 (ERRORS_DISABLED): 默认模式, 每个 SD-HCI 没有错误报告。 <p>复位值: 0x0</p> <p>存在：一直</p>

位段	名称	类型	说明
2	SWIN_TH_EN	R/W	采样窗口阈值启用 选择调优模式 字段只有在 SAMPLE_CLK_SEL 为 “0” 时才可以编程 值： <ul style="list-style-type: none"> 0x1 (THRESHOLD_MODE): 调优引擎选择第一个满足 SWIN_TH_VAL 字段设置的阈值的完整采样窗口 0x0 (LARGEST_WIN_MODE): 调优引擎扫描所有抽头，并在最大的窗口上解决 复位值：1
1	CI_SEL	R/W	当校正的中心相位选择可以在 tuning_cclk_sel 输出上驱动的代码，选择时间间隔。 值： <ul style="list-style-type: none"> 0x0 (WHEN_IN_BLK_GAP): 以块间隙驱动 0x1 (WHEN_IN_IDLE): 传输结束时驱动 复位值：0x0 存在：一直
0	AT_EN	R/W	设置这个位使能自动调整引擎。当 core 配置了 mode3 返回支持时，这个位是默认启用的。当 core 配置了具有 Mode3 重新调优但 SW 希望禁用 Mode3 重新调优时，将此位清除为 0。 只有当 CLK_CTRL_R.SD_CLK_EN 是 0 时，才可以对该字段进行编程。 值： <ul style="list-style-type: none"> 0x1 (AT_ENABLE): 启用自动调优。 0x0 (AT_DISABLE): 禁用自动调优。 复位值：1 存在：一直

3.6.5.8 AT_STAT_R

- 寄存器名：调优和自动调优状态寄存器
- 描述：寄存器读取调优和自动调优引擎使用的中心、左和右代码。中心代码字段也用于软件托管调优。

- 大小：32 位
- 偏移量：0x500 + 0x44

图表 3-148 AT_STAT_R 寄存器字段说明

位段	名称	类型	说明
31:24	RSDV1	R	AT_STAT_R 寄存器的这些位是保留的。它们总是返回 0。 复位值：0x0 存在：一直
23:16	L_EDGE_PH_CODE	R	左边缘相位代码。读取此字段由自动调优引擎返回相位代码 值，用于采样窗口左边缘的数据采样。 复位值：0x0 存在：一直
15:8	R_EDGE_PH_CODE	R	右边缘相位代码。读取此字段由自动调优引擎返回相位代码 值，用于采样窗口右边缘的数据采样。 复位值：0x0 存在：一直
7:0	CENTER_PH_CODE	R/W	中间相位代码。读取此字段将返回 tuning_cclk_sel 输出的 当前值。设置 AT_CTRL_R.SW_TUNE_EN 允许软件写入该字 段，其内容反映在 tuning_cclk_sel 上。 复位值：6 存在：一直 测试：只读

4 QSPI

4.1 概述

DW_apb_ssi 是一个可配置、可合成、可编程的组件，它是一个全双工的主串行接口。主机处理器通过 APB 接口访问 DW_apb_ssi 上的数据、控制和状态信息。DW_apb_ssi 也可以使用一组可选的 DMA 信号与 DMA 控制器连接，该信号集可以在配置时选择。

如后面更详细的描述，DW_apb_ssi 可以配置为一个串行主接口。DW_apb_ssi 可以连接到使用摩托罗拉串行外设接口（SPI）的任何串行从外围设备。

4.2 主要特性

DW_apb_ssi 具有以下特性：

- APB 接口
- APB3 协议支持
- 可扩展的 APB 数据总线宽度-支持 32 位的 APB 数据总线宽度。
- 串行主-使能串行通信串行从外围设备。
- 在主模式下可配置和可编程的 Quad SPI 支持-配置 DW_apb_ssi 以支持 Quad SPI 模式，然后为 Quad SPI 传输编程 DW_apb_ssi。
- 数据屏蔽支持-使 DW_apb_ssi 能够有选择地更新设备中的字节。该特性仅适用于增强型 SPI 模式。
- eXecute-In-Place (XIP) 支持-使 DW_apb_ssi 主能够作为内存映射的 I/O，并根据 APB 读取请求从设备中获取数据。该特性仅适用于增强型 SPI 模式。
- DMA 控制器接口-使 DW_apb_ssi 能够使用传输请求的握手接口与总线上的 DMA 控制器连接。
- 独立屏蔽中断-主冲突、发送 FIFO 溢出、发送 FIFO 溢出、发送 FIFO 空、接收 FIFO 满、接收 FIFO 下溢和接收 FIFO 溢出中断都可以被独立屏蔽。
- 多主竞争检测-通知处理器多个串行主访问串行总线。
- 当配置为主模式时，接收串行数据位（rxdata）的采样时间可编程延迟；实现对路由延迟的可编程控制，从而产生更高的串行数据比特率。
- 从选择输出数据数量-当作为串行主运行时，可以产生 4 个串行从选择输出信号。
- 组合中断线-将 DW_apb_ssi 的一条组合中断线连接到中断控制器。
- 可编程特性：
 - 串行接口操作-选择摩托罗拉 SPI。
 - 时钟比特率-动态控制数据传输的串行比特率；仅用于串行主操作模式。
 - 数据项大小（4~32 位）-每个数据传输项的大小由程序员控制。

- 可配置特性：
 - 中断极性-此配置项选择输出中断线的有效电平。
 - 串行时钟极性-此配置选项在复位后直接选择 SPI 格式的串行时钟极性。
 - 串行时钟相位-此配置选项在复位后直接选择 SPI 格式的串行时钟相位。

4.3 接口

管脚名称	方向	宽度	说明
QSPI0_CSN0	O	1	QSPI0 的从选择输出，有效极性为逻辑 0。
QSPI0_CSN1	O	1	QSPI0 的从选择输出，有效极性为逻辑 0。
QSPI0_D0_MOSI	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。
QSPI0_D1_MISO	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。
QSPI0_D2_WP	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。 写保护（WP）管脚可用于防止设备的状态寄存器被写入。
QSPI0_D3_HOLD	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。 HOLD 管脚允许器件在被主动选择时暂停。
QSPI0_SCLK	O	1	SPI 串行时钟输出管脚为串行输入和输出操作提供时序。
QSPI1_CSN0	O	1	QSPI1 的从选择输出，有效极性为逻辑 0。
QSPI1_D0_MOSI	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。
QSPI1_D1_MISO	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。
QSPI1_D2_WP	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。 写保护管脚可用于防止设备的状态寄存器被写入。
QSPI1_D3_HOLD	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。 HOLD 管脚允许设备在被主动选择时暂停。
QSPI1_SCLK	O	1	SPI 串行时钟输出管脚为串行输出和输出操作提供时序。
SPI_CSN	O	1	SPI 的从选择输出，有效极性为逻辑 0。
SPI_MISO	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。
SPI_MOSI	IO	1	双向 IO 管脚将指令、地址或数据串行写入设备。
SPI_SCLK	O	1	SPI 串行时钟输出管脚为串行输入和输出操作提供时序。

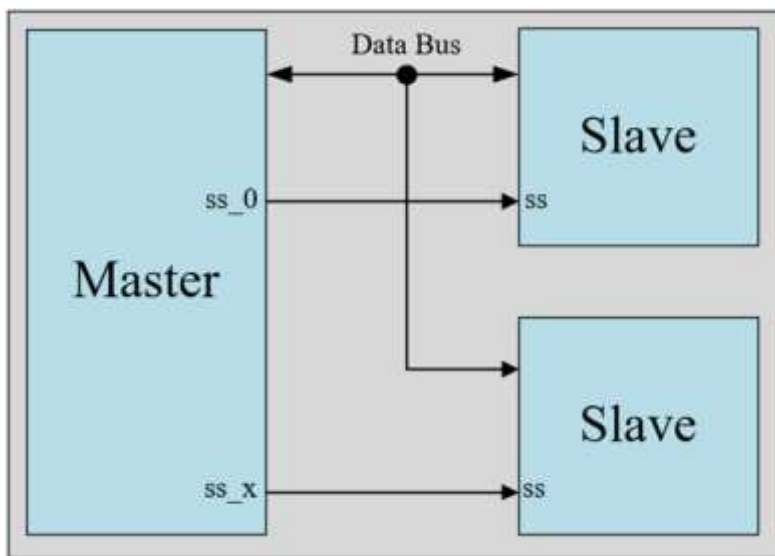
4.4 功能描述

4.4.1 数据流

为了让 DW_apb_ssi 连接到串行从外围设备，外围设备必须具有摩托罗拉 SPI。这是摩托罗拉的四线全双工串行协议。串行时钟相位和极性有四种可能的组合。时钟相位（SCPH）决定串行传输是从从选择信号的下降沿还是从串行时钟的一个沿开始。DW_apb_ssi 空闲或禁用时，从选择线保持高电平。

您可以对控制寄存器 0（CTRLR0）中的帧格式（FRF）位段进行编程，以选择使用哪种协议。

DW_apb_ssi 支持的串行协议允许使用硬件或软件选择或寻址串行从。在硬件中实现时，串行从在专用硬件选择线的控制下进行选择。从串行总生成的选择线数等于总线上的串行从数。串行主设备在数据传输开始之前断言目标串行从的选择线。该架构如图表 4-1 所示。



图表 4-1 从选择

4.4.2 时钟频率

当 DW_apb_ssi 配置为主设备时，比特率时钟（sclk_out）的最大频率是 ssi_clk 频率的一半，这允许移位控制逻辑在 sclk_out 的一个时钟沿捕获数据并在相反沿传播数据。sclk_out 线仅在活动传输正在进行时切换。在所有其他时间，它都处于非活动状态。

sclk_out 的频率可以从以下公式中得出：

$$F_{sclk_out} = F_{ssi_clk} / SCKDV$$

SCKDV 是可编程寄存器 BAUDR 中的一个位段，保存 0~65,534 范围内的任何偶数值。如果 SCKDV 为 0，则禁用 sclk_out。

4.4.3 发送和接收 FIFO 缓冲区

当 DW_apb_ssi 被禁用（SSI_EN = 0）或复位（presetn）时，发送和接收 FIFO 缓冲区被清空。

发送 FIFO 通过 APB 写命令加载到 DW_apb_ssi 数据寄存器 (DR)。移位控制逻辑将数据从发送 FIFO 弹出 (移除) 到发送移位寄存器。当 FIFO 中的条目数小于或等于 FIFO 阈值时, 发送 FIFO 生成 FIFO 空中断请求 (ssi_txe_intr)。通过可编程寄存器 TXFTLR 生成的阈值决定了产生中断的 FIFO 条目数。阈值允许您向处理器提供 FIFO 几乎为空的早期指示。如果你尝试将数据写入已满的发送 FIFO, 则会生成发送 FIFO 溢出中断 (ssi_txo_intr)。

APB 读命令将数据从接收 FIFO 弹出到 DW_apb_ssi 数据寄存器 (DR)。接收 FIFO 由接收控制逻辑从接收移位寄存器加载。当 FIFO 中的条目数大于或等于 FIFO 阈值加 1 时, 接收 FIFO 生成 FIFO 满中断请求 (ssi_rxf_intr)。通过可编程寄存器 RXFTLR 设置的阈值决定产生中断的 FIFO 条目数。阈值允许您向处理器提供接收 FIFO 几乎已满的早期指示。当接收移位逻辑尝试将数据加载到完全满的接收 FIFO 中时, 会生产接收 FIFO 溢出中断 (ssi_rxo_intr)。然而, 这些新收到的数据会丢失。如果您尝试读空的接收 FIFO, 会生成接收 FIFO 下溢中断 (ssi_rxu_intr)。这会提醒处理器读取的数据是无效的。

图表 4-2 发送 FIFO 阈值 (TFT) 解码值

TFT 值 (二进制)	说明
0000_0000	当发送 FIFO 中存在 0 个数据条目时, 产生 ssi_txe_intr。
0000_0001	当发送 FIFO 中存在 1 个或更少数据条目时, 产生 ssi_txe_intr。
0000_0010	当发送 FIFO 中存在 2 个或更少数据条目时, 产生 ssi_txe_intr。
0000_0011	当发送 FIFO 中存在 3 个或更少数据条目时, 产生 ssi_txe_intr。
...	...
...	...
1111_1100	当发送 FIFO 中存在 252 或更少数据条目时, 产生 ssi_txe_intr。
1111_1101	当发送 FIFO 中存在 253 或更少数据条目时, 产生 ssi_txe_intr。
1111_1110	当发送 FIFO 中存在 254 或更少数据条目时, 产生 ssi_txe_intr。
1111_1111	当发送 FIFO 中存在 255 或更少数据条目时, 产生 ssi_txe_intr。

图表 4-3 接收 FIFO 阈值 (RFT) 解码值

RFT 值	说明
0000_0000	当接收 FIFO 中存在 1 个或更多数据条目时, 产生 ssi_rxf_intr。
0000_0001	当接收 FIFO 中存在 2 个或更多数据条目时, 产生 ssi_rxf_intr。
0000_0010	当接收 FIFO 中存在 3 个或更多数据条目时, 产生 ssi_rxf_intr。
0000_0011	当接收 FIFO 中存在 4 个或更多数据条目时, 产生 ssi_rxf_intr。

RFT 值	说明
...	...
...	...
1111_1100	当接收 FIFO 中存在 253 个或更多数据条目时，产生 ssi_rxf_intr。
1111_1101	当接收 FIFO 中存在 254 个或更多数据条目时，产生 ssi_rxf_intr。
1111_1110	当接收 FIFO 中存在 255 个或更多数据条目时，产生 ssi_rxf_intr。
1111_1111	当接收 FIFO 中存在 256 个数据条目时，产生 ssi_rxf_intr。

4.4.4 SSI 中断

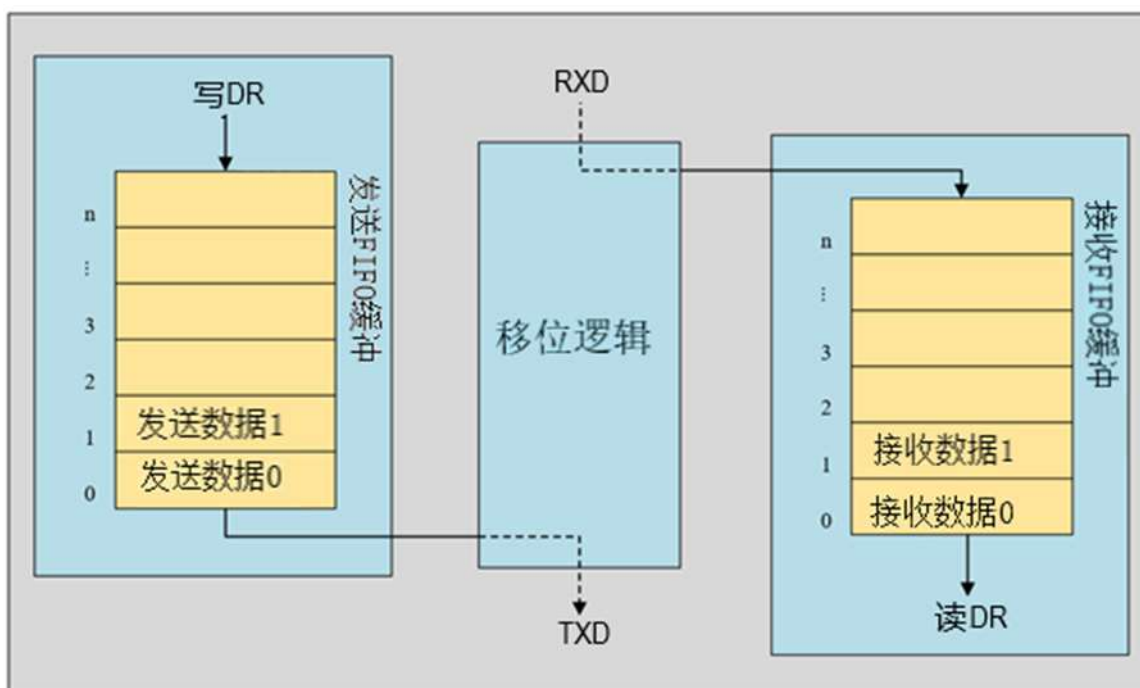
DW_apb_ssi 中断描述如下：

- 发送 FIFO 空中断 (ssi_txe_intr) -当发送 FIFO 等于或小于阈值并需要服务防止下溢时设置。通过软件可编程的寄存器设置的阈值决定产生中断的发送 FIFO 条目数。当数据写入发送 FIFO 缓冲区时，该中断由硬件清除，使其超过阈值水平。
- 发送 FIFO 溢出中断(ssi_txo_intr)-当 APB 访问在发送 FIFO 完全满后尝试写入发送 FIFO 时设置。设置后，从 APB 写入的数据将被丢弃。此中断保持置位，直到读取发送 FIFO 溢出中断清除寄存器 (TXOICR)。
- 接收 FIFO 满中断 (ssi_rxf_intr)-当接收 FIFO 等于或大于其阈值加 1 并需要服务防止溢出时设置。通过软件可编程寄存器设置的阈值决定产生中断的接收 FIFO 条目数。当从接收 FIFO 缓冲区读取数据时，该中断由硬件清除，使其低于阈值水平。
- 接收 FIFO 溢出中断 (ssi_rxo_intr) -当接收逻辑尝试在接收 FIFO 被完全填满后将数据放入接收 FIFO 时设置。设置后，新接收的数据将被丢弃。该中断保持置位，直到读取接收 FIFO 溢出中断清除寄存器 (RXOICR)。
- 接收 FIFO 下溢中断 (ssi_rxu_intr) -当 APB 访问尝试在 FIFO 为空时从其读取时设置。设置后，从接收 FIFO 读回 0。该中断保持置位，直到读取接收 FIFO 下溢中断清除寄存器 (RXUICR)。
- 多主争用中断 (ssi_mst_intr) -仅在 DW_apb_ssi 组件配置为串行主设备时出现。当串行总线上的另一个串行主选择 DW_apb_ssi 主作为串行从设备并正在主动传输数据时，将设置中断。这会通知处理器串行总线上可能的争用。该中断保持置位，直到读取多主中断清除寄存器 (MSTICR)。
- 组合中断请求 (ssi_intr) -上述所有中断请求屏蔽后的 OR'ed 结果。要屏蔽该中断信号，您必须屏蔽所有其他 DW_apb_ssi 中断请求。

4.4.5 传输模式

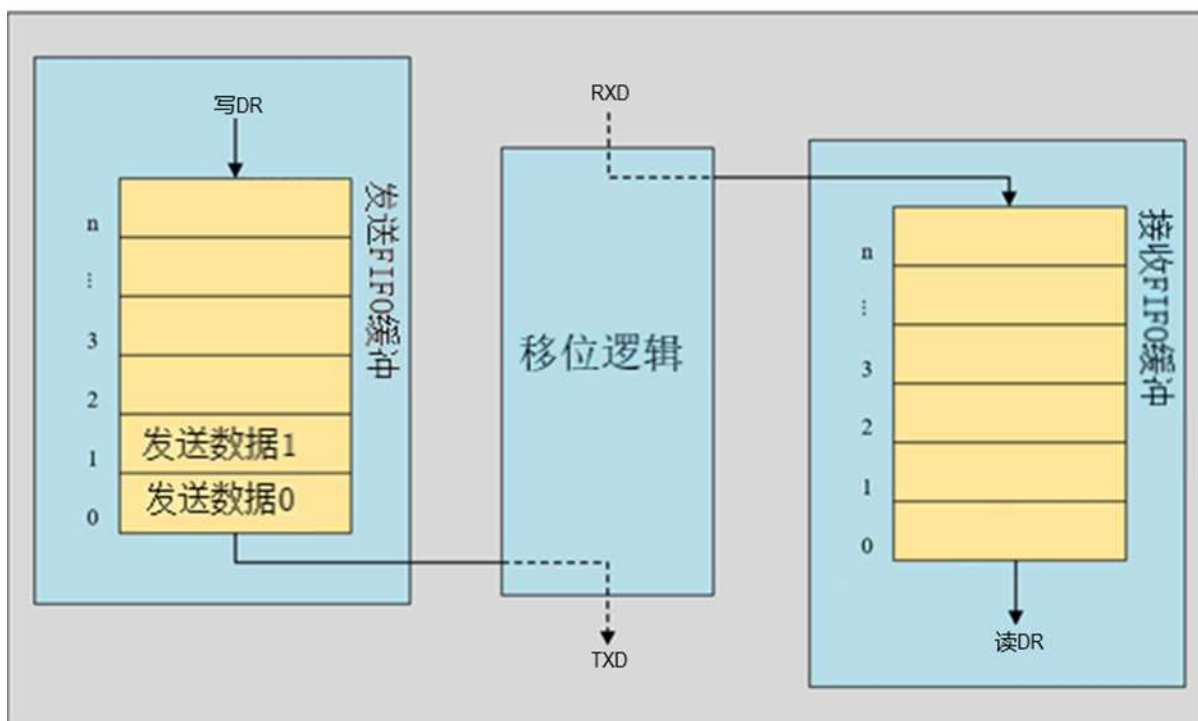
- 当 $TMOD = 2'b00$ 时, 发送和接收逻辑都有效。根据所选的帧格式 (串行协议), 数据传输正常进行。发送数据从发送 FIFO 弹出并通过 TXD 线发送到目标设备, 目标设备在 RXD 线上回复数据。来自目标设备的接收数据在每个数据帧结束时从接收移位寄存器移入接收 FIFO。
- 当 $TMOD = 2'b01$ 时, 接收数据无效, 不应存入接收 FIFO。根据所选的帧格式 (串行协议), 数据传输正常进行。发送数据从发送 FIFO 弹出并通过 TXD 线发送到目标设备, 目标设备在 RXD 线上回复数据。在数据帧结束时, 接收移位寄存器不会将其新接收到的数据加载到接收 FIFO 中。接收移位寄存器中的数据被下一次传输覆盖。进入该模式时, 应该屏蔽源自接收逻辑的中断。
- 当 $TMOD = 2'b10$ 时, 发送数据无效。当配置为从时, 在仅接收模式下, 发送 FIFO 从不弹出。TXD 输出在传输期间保持恒定的逻辑电平。根据所选的帧格式 (串行协议), 数据传输正常进行。来自目标设备的接收数据在每个数据帧结束时从接收移位寄存器移入接收 FIFO。进入该模式时, 应该屏蔽源自发送逻辑的中断。
- 当 $TMOD = 2'b11$ 时, 发送数据用于向 EEPROM 设备发送操作码和/或地址。通常, 这需要三个数据帧 (8 位操作码后跟 8 位高地址和 8 位低地址)。在传输操作码和地址期间, 接收逻辑不捕获任何数据 (只要 DW_apb_ssi 主在其 TXD 线上传输数据, 就会忽略 RXD 线上的数据)。DW_apb_ssi 主继续发送数据直到发送 FIFO 为空。因此, 在发送 FIFO 中只需要有足够的帧来向 EEPROM 提供操作码和地址。如果发送 FIFO 中的数据帧比需要的多, 读取的数据就会丢失。当发送 FIFO 变空 (所有控制信息都已发送), RXD 线上的数据有效, 存入接收 FIFO。TXD 输出保持在一个恒定的逻辑电平。串行传输继续进行, 直到 DW_apb_ssi 主接收到的数据帧数与 CTRLR1 寄存器中 NDF 字段的值加 1 一致。

对于发送和接收传输 (控制寄存器 0 的传输模式字段[9:8]=2'b00), 从 DW_apb_ssi 发送到外部串行设备的数据被写入到发送 FIFO。从外部串行设备接收到 DW_apb_ssi 的数据被推入到接收 FIFO。



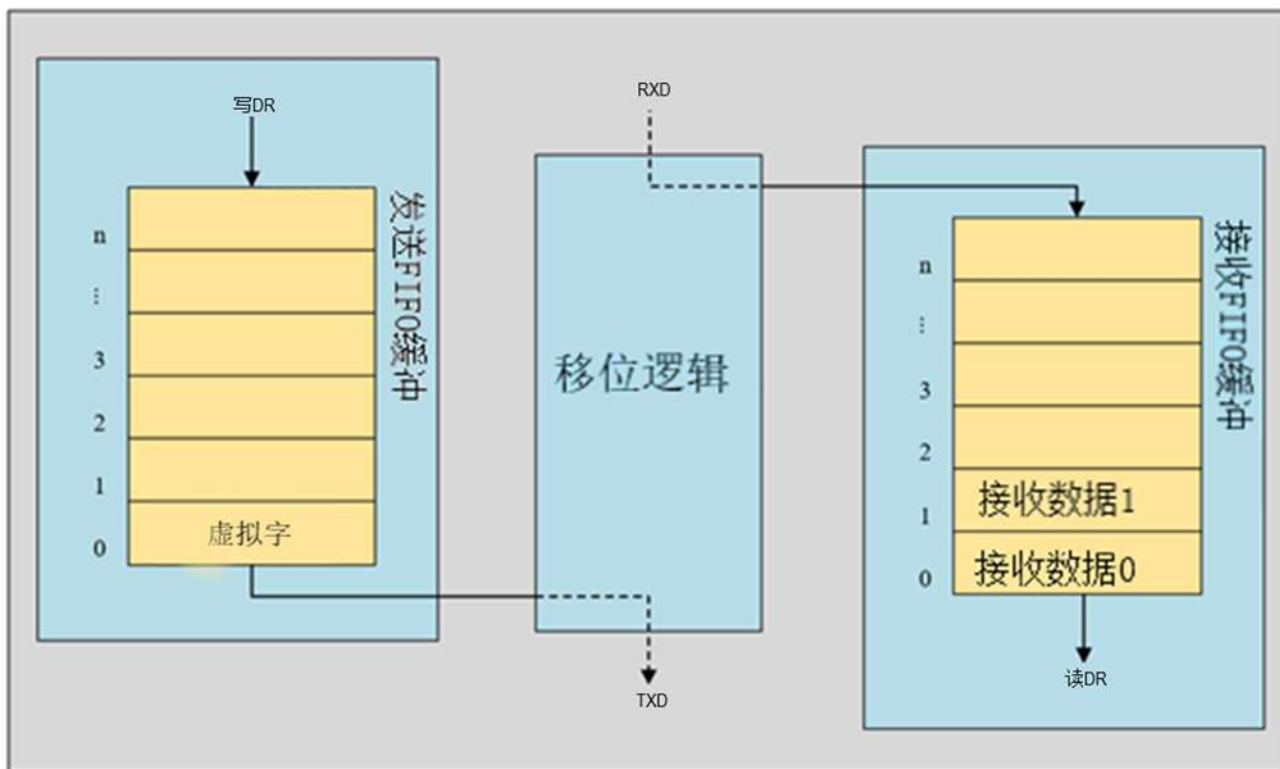
图表 4-4 发送和接收 SPI 传输的 FIFO 状态

对于仅发送传输（控制寄存器 0 的传输模式字段[9:8]=2'b01），从 DW_apb_ssi 发送到外部串行设备的数据被写入发送 FIFO。从外部串行设备接收到的数据被认为是无效的，因此不会存储在 DW_apb_ssi 接收 FIFO 中。



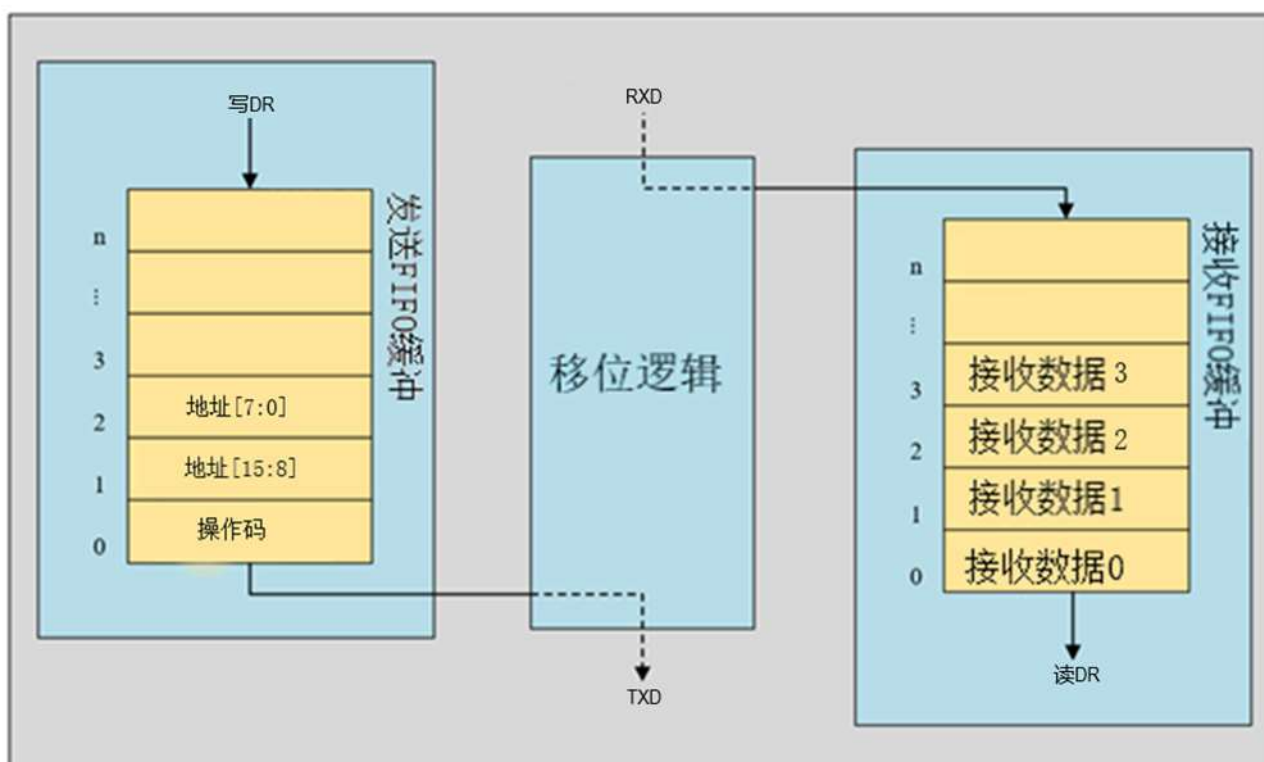
图表 4-5 仅发送 SPI 传输的 FIFO 状态

对于仅接收传输（控制寄存器 0 的传输模式字段[9:8]= 2'b10），从 DW_apb_ssi 传输到外部串行设备的数据无效，因此将单个虚拟字写入发送 FIFO，开始串行传输。DW_apb_ssi 的 TXD 输出在串行传输期间保持在恒定逻辑电平。从外部串行设备接收到 DW_apb_ssi 的数据被推入接收 FIFO。



图表 4-6 仅接收 SPI 传输的 FIFO 状态

对于 EEPROM 读传输（控制寄存器 0 的传输模式字段[9:8]= 2'b11），操作码和/或 EEPROM 地址被写入发送 FIFO。在这些控制帧传输期间，接收到的数据不会被 DW_apb_ssi 主捕获到。发送控制帧后，来自 EEPROM 的接收数据存储在接收 FIFO 中。



图表 4-7 EEPROM 读传输模式的 FIFO 状态

4.4.6 RXD 采样延迟

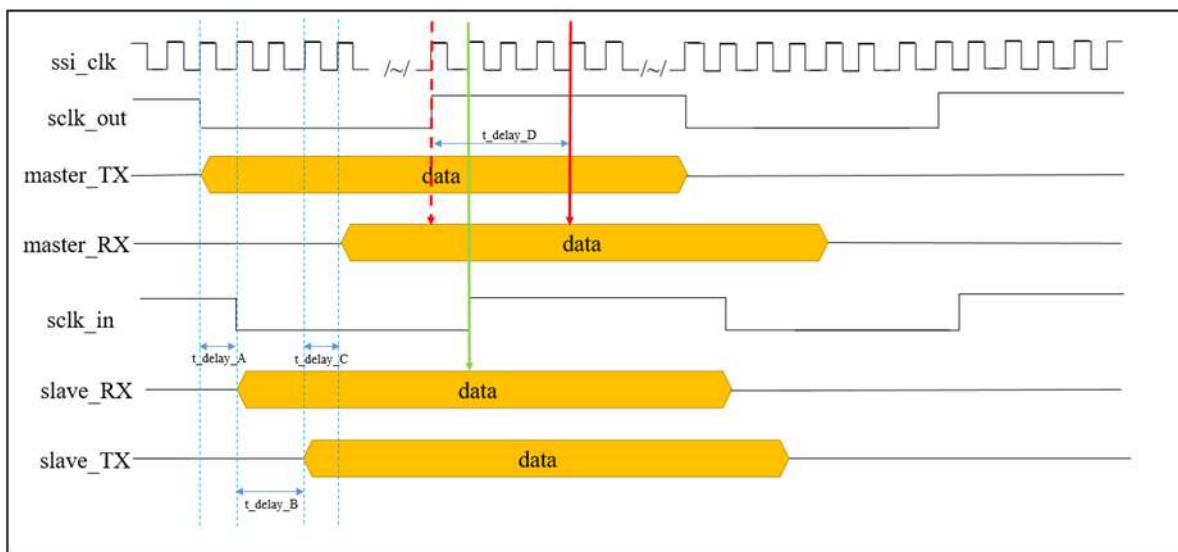
当 DW_apb_ssi 配置为主时，可在设计中包含额外的逻辑，以延迟 RXD 信号的默认采样时间。这个额外的逻辑可以帮助增加串行总线上可达到的最大频率。

来自自主的 sclk_out 信号和来自从的 RXD 信号的往返路由延迟可能意味着 RXD 信号的时序（如主所见）已偏离正常采样时间。图表 4-8 说明了这种情况。

从使用来自自主的 sclk_out 信号作为选通信号，以便将 RXD 信号驱动到串行总线上。从设备对 sclk_out 信号的路由和采样延迟可能意味着在主对 RXD 信号采样之前 RXD 位没有稳定到正确的值。图表 4-8 显示了一个示例，说明 rxd 信号的路由延迟如何会导致在主采样端口的默认时间出现不正确的 rxd 值。

如果没有 RXD 采样延迟逻辑，必须增加传输的波特率，以确保 RXD 信号的建立时间在范围内，这会降低串行接口的频率。

当包括 RXD 采样延迟逻辑时，可以动态编辑延迟值，以便将 RXD 信号的采样时间从默认值移至等于 ssi_clk 周期数。



图表 4-8 往返路由延迟对 sclk_out 信号的影响

4.4.7 数据传输

数据传输由串行主设备启动。当 DW_apb_ssi 使能时 (SSI_EN=1)，发送 FIFO 中至少存在一个有效数据条目，并且选择了串行从设备。当主动传输数据时，状态寄存器 (SR) 中的忙标志 (BUSY) 被置位。在尝试新的串行传输之前，必须等到忙标志被清除。

当传输模式为“发送和接收”或“仅发送”时（分别为 TMOD = 2'b00 或 TMOD = 2'b01），发送 FIFO 为空时，移位控制逻辑终止传输。对于连续数据传输，必须确保在所有数据传输完成之前，发送 FIFO 缓冲区不会变空。发送 FIFO 阈值级别 (TXFTLR) 可用于提前中断 (ssi_txe_intr) 处理器，指示发送 FIFO 缓冲区几乎为空。当 DMA 用于 APB 访问时，发送数据级别 (DMATDLR) 可用于请求 (dma_tx_re) DMA 控制器，指示发送 FIFO 几乎为空。然后可以用数据重新填充 FIFO，继续串行传输。在使能串行从之前，可以将一个数据块（至少两个 FIFO 条目）写入发送 FIFO。这可以确保在发送 FIFO 中出现构成连续传输的数据帧数之前，串行传输不会开始。

当传输模式为“仅接收” (TMOD = 2'b10) 时，选择串行从时，通过向发送 FIFO 写入一个“虚拟”数据字开始串行传输。DW_apb_ssi 的 TXD 输出在串行传输期间保持在一个恒定的逻辑电平。发送 FIFO 在开始时仅弹出一次，并且在传输期间可能保持为空。串行传输的结束由控制寄存器 1 (CTRLR1) 中的“数据帧数” (NDF) 字段控制。

例如，如果你想从串行从外设接收 24 个数据帧，应该用值 23 编程 NDF 字段。当接收到的帧数等于 NDF 值+1 时，接收逻辑终止串行传输。这种传输模式增加了 APB 总线的带宽，因为在传输期间不需要服务发送 FIFO。每次接收 FIFO 生成 FIFO 满中断请求时，都应读取接收 FIFO 缓冲区以防止溢出。

当传输模式为“eeprom_read”时 (TMOD = 2'b11)，选择串行从 (EEPROM) 时，通过将操作码和/或地址写入发送 FIFO 开始串行传输。操作码和地址被发送到 EEPROM 设备，之后从 EEPROM 设备接收读取数据并存储在接收 FIFO 中。串行传输的结束由 CTRLR1 中的 NDF 字段控制。

4.4.8 SPI 模式下的 XIP 模式支持

XIP 模式可直接通过 APB 接口传输 SPI 数据，而无须写入 DW_apb_ssi 的数据寄存器。通过在 APB 接口上选择 SSI_XIP_EN 配置参数，可以在 DW_apb_ssi 中使能 XIP 模式。该参数包括一个额外的边带信号 xip_en，该信号指示 APB 传输时寄存器读写还是 XIP 读取。如果 xip_en 信号被驱动为 1，DW_apb_ssi 期望 APB 接口上只有读取请求。该请求被转换为串行接口上的 SPI 读，在收到数据后不久，在同一事务中将数据返回到 APB 接口。

4.4.9 DMA 控制器接口

DW_apb_ssi 具有可选的内置 DMA 功能，可以在配置时选择；它具有与 DMA 控制器的握手接口，以请求和控制传输。APB 总线用于执行与 DMA 之间的数据传输。虽然 DW_apb_ssi DMA 操作以通用方式设计的，以尽可能容易地适应任何 DMA 控制器，但它被设计为与 DesignWare DMA 控制器 DW_axi_dmac 无缝工作，并得到最佳使用。

当 DW_apb_ssi 与 DW_axi_dmac 连接时，DW_axi_dmac 始终是一个流控制器；也就是说，它控制块大小。这必须通过 DW_axi_dmac 中的软件进行编程。

DW_apb_ssi 使用两个 DMA 通道，一个用于发送数据，一个用于接收数据。DW_apb_ssi 具有以下 DMA 寄存器：

- DMACR -使能 DMA 操作的控制寄存器。
- DMATDLR -设置发出 DMA 请求时发送 FIFO 级别的寄存器。
- DMARDLR -设置发出 DMA 请求时接收 FIFO 级别的寄存器

DW_apb_ssi 使用以下握手信号与 DMA 控制器连接：

- dma_tx_req
- dma_tx_single
- dma_tx_ack
- dma_rx_req
- dma_rx_single
- dma_rx_ack

要启用 DW_apb_ssi 上的 DMA 控制器接口，必须写入 DMA 控制寄存器 (DMACR)。将 1 写入 DMACR 寄存器的 TDMAE 位段，使能 DW_apb_ssi 发送握手接收。将 1 写入 DMACR 寄存器的 RDMAE 位段，使能 DW_apb_ssi 接收握手接口。

图表 4-9 DMA 发送数据级别 (DMATDL) 解码值

DMATDL 值	说明
0000_0000	当发送 FIFO 中存在 0 个数据条目时，产生 dma_tx_req。
0000_0001	当发送 FIFO 中存在 1 个或更少数据条目时，产生 dma_tx_req。

DMATDL 值	说明
0000_0010	当发送 FIFO 中存在 2 个或更少数据条目时, 产生 dma_tx_req。
0000_0011	当发送 FIFO 中存在 3 个或更少数据条目时, 产生 dma_tx_req。
...	...
...	...
1111_1100	当发送 FIFO 中存在 252 个或更少数据条目时, 产生 dma_tx_req。
1111_1101	当发送 FIFO 中存在 253 个或更少数据条目时, 产生 dma_tx_req。
1111_1110	当发送 FIFO 中存在 254 个或更少数据条目时, 产生 dma_tx_req。
1111_1111	当发送 FIFO 中存在 255 个或更少数据条目时, 产生 dma_tx_req。

图表 4-10 DMA 接收数据级别 (DMARDL) 解码值

DMARDL 值	说明
0000_0000	当接收 FIFO 中存在 1 个或更多数据条目时, 产生 dma_rx_req。
0000_0001	当接收 FIFO 中存在 2 个或更多数据条目时, 产生 dma_rx_req。
0000_0010	当接收 FIFO 中存在 3 个或更多数据条目时, 产生 dma_rx_req。
0000_0011	当接收 FIFO 中存在 4 个或更多数据条目时, 产生 dma_rx_req。
...	...
...	...
1111_1100	当接收 FIFO 中存在 253 个或更多数据条目时, 产生 dma_rx_req。
1111_1101	当接收 FIFO 中存在 254 个或更多数据条目时, 产生 dma_rx_req。
1111_1110	当接收 FIFO 中存在 255 个或更多数据条目时, 产生 dma_rx_req。
1111_1111	当接收 FIFO 中存在 256 数据条目时, 产生 dma_rx_req。

4.4.9.1 传输水位级别和传输 FIFO 下溢

在 DW_apb_ssi 串行传输期间, 只要发送 FIFO 中的条目数小于或等于 DMA 发送数据级别寄存器 (DMATDLR) 值, 就会向 DW_axi_dmac 发出发送 FIFO 请求, 这称为水位级别。DW_axi_dmac 通过将长度为 CTLx.DEST_MSIZ 的数据突发写入发送 FIFO 缓冲区进行响应。

应该经常从 DMA 获取数据, 以便发送 FIFO 连续执行串行传输, 也就是说, 当 FIFO 开始变空时, 应该触发另一个 DMA 请求, 否则 FIFO 的数据就会耗尽 (下溢)。为防止出现这种情况, 必须正确设置水印级别。

考虑做出假设的示例：

$DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - SSI.DMATDLR$

这里，一个 DMA 突发中要传输的数据项数等于发送 FIFO 中的空白空间。考虑两种不同的水印级别设置。

案例 1: $DMATDLR = 2$

- 发送 FIFO 水位级别 = $SSI.DMATDLR = 2$
- $DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - SSI.DMATDLR = 6$
- SSI transmit $FIFO_DEPTH = 8$
- $DMA.CTLx.BLOCK_TS = 30$

因此，所需的突发事务数等于块大小除以每个突发的数据项数：

$DMA.CTLx.BLOCK_TS / DMA.CTLx.DEST_MSIZE = 30 / 6 = 5$

DMA 块传输的突发事务数为 5。但是水位级别 $SSI.DMATDLR$ 相当低。因此，在 SSI 串行传输线需要传输数据但在发送 FIFO 中没有数据的情况下，发生 SSI 下溢的可能性很高。这是因为 DMA 在发送 FIFO 变空之前没有时间处理 DMA 请求。

案例 2: $DMATDLR = 6$

- 发送 FIFO 水位级别 = $SSI.DMATDLR = 6$
- $DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - SSI.DMATDLR = 2$
- SSI transmit $FIFO_DEPTH = 8$
- $DMA.CTLx.BLOCK_TS = 30$

块中的突发事务数：

$DMA.CTLx.BLOCK_TS / DMA.CTLx.DEST_MSIZE = 30 / 2 = 15$

在块传输中，一个 DMA 块传输中有 15 个目标突发事务。但是水位级别 $SSI.DMATDLR$ 很高。因此，SSI 下溢的可能性很低，因为 DMA 控制器在 SSI 发送 FIFO 变空之前有足够的时间处理目标突发事务请求。

因此，第二种情况下溢的概率较低，代价是每个块有更多的突发事务。与前一种情况相比，这提供了每个块可能更大数量的 AMBA 突发和更差的总线利用率。

因此，选择水位级别的目标是最小化每个块的事务数量，同时将下溢条件的概率保持在可接受的水平。实际上，这是 SSI 传输数据的速率与 DMA 响应目标突发请求的速率之比的函数。

例如，将通道提升到 DMA 中的最高优先级通道，并将 DMA 主接口提升到 AMBA 层中的最高优先级主，提高了 DMA 控制器可以响应突发事务请求的速率，这反过来又允许降低水位级别，从而提高总线利用率，而不会影响发生下溢的可能性。

为实现最佳操作， $DMA.CTLx.DEST_MSIZE$ 应设置为触发发送 DMA 请求的 FIFO 级别，也就是：

$DMA.CTLx.DEST_MSIZE = SSI.FIFO_DEPTH - SSI.DMATDLR$

遵循这个公式可以减少块传输所需的 DMA 突发数量，从而提高 AMBA 总线利用率。

4.4.9.2 接收水位级别和接收 FIFO 溢出

在 DW_apb_ssi 串行传输期间，只要接收 FIFO 中的条目数等于或大于 DMA 接收数据水平寄存器，即 DMARDLR+1，就会向 DW_axi_dmac 发出接收 FIFO 请求，这称为水位级别。DW_axi_dmac 通过从长度为 CTLX.SRC_MSIZ 的接收 FIFO 缓冲区获取突发数据来响应。

DMA 应该经常获取数据，以便接收 FIFO 连续接收串行传输，也就是说，当 FIFO 开始填充时，会请求另一个 DMA 传输。否则，FIFO 会充满数据（溢出）。为防止出现这种情况，应正确设置水位级别。

与前面描述的选择发送水位级别类似，接收水位级别 DMARDLR+1 应设置为最小化溢出概率。这是每个块所需的 DMA 突发事务数与发生溢出的概率之间的权衡。

为了实现最佳操作，DMA.CTLX.SRC_MSIZ 应该设置在水位级别，也就是：

$$\text{DMA.CTLX.SRC_MSIZ} = \text{SSI.DMARDLR} + 1$$

遵循这个公式可以减少块传输中的 DMA 突发数量，而这又可以提高 AMBA 总线的利用率。

4.4.10 复位信号

DW_apb_ssi 包括以下复位信号，每个信号专用于自己的时钟域：

- presetn - 复位 PCLK 时钟域中的逻辑。
- ssi_rst_n - 复位 ssi_clk 时钟域中的逻辑。

复位 DW_apb_ssi 的推荐过程如下：

1. 产生 ssi_rst_n 和 presetn 信号。这两个信号的产生顺序以及它们与 ssi_clk 和 pclk 的时序关系并不重要。
2. 与 ssi_clk 同步释放 ssi_rst_n 信号。
3. 与 pclk 同步释放 presetn 信号。

4.5 使用

图表 4-11 显示了启动 DW_apb_ssi 主 SPI 串行传输的典型软件流程。该图还显示了串行主组件内部的硬件流程。

读/写行为	说明
W1S	写入 1 会设置该寄存器字段。
W1T	写入 1 会切换该寄存器字段。
W0C	写入 0 会清除该寄存器字段。
W0S	写入 0 会设置该寄存器字段。
W0T	写入 0 会切换该寄存器字段。
WC	任何写入都会清除该寄存器字段。
WS	任何写入都会设置该寄存器字段。
WM	任何写入都会切换该寄存器字段。
no Read Behavior attribute	无法读取该寄存器，它是只写的。
no Write Behavior attribute	无法写入该寄存器，它是只读的。

图表 4-13 内存访问示例

内存访问	说明
R	只读寄存器字段
W	只写寄存器字段
R/W	读/写寄存器字段
R/W1C	可以读取该寄存器字段，写 1 清除它。
RC/W1C	读取该寄存器字段将其清除，写 1 清除它。
R/Wo	可以读取该寄存器字段，只能向该字段写入一次。

4.6.1 ssi_memory_map/ssi_address_block 寄存器

4.6.1.1 CTRLR0

- 寄存器名：控制寄存器 0
- 描述：该寄存器控制串行数据传输。当 DW_apb_ssi 使能时，无法写入该寄存器。通过写入 SSIENR 寄存器来使能和禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0x0
- 存在：一直

图表 4-14 CTRLR0 寄存器字段说明

位段	名称	类型	说明
31:25	RSVD_CTRLR0	R	SSTE 保留位 - 只读 存在：一直 复位值：0x0
24	SSTE	R/W	从选择切换使能 当在 SPI 模式下，时钟相位（SCPH）设置为 0 时，该寄存器控制数据帧之间从选择线（ss*_n）的行为。如果该寄存器字段设为 1，ss*_n 线将在连续的数据帧之间切换，ss*_n 为高电平时，串行时钟（sclk）保持默认值。如果该寄存器字段设为 0，ss*_n 将保持低电平，sclk 将在传输期间连续运行。 存在：一直 复位值：1
23	RSVD_CTRLR0_23	R	CTRLR0_23 保留位 - 只读 存在：一直 复位值：0x0
22:21	SPI_FRF	R(SPI); R/W(QSPI)	SPI 帧格式 为发送/接收数据选择数据帧格式。仅当 SSI_SPI_MODE 设为“Dual”、“Quad”或“Octal”模式时才有效。 当 SSI_SPI_MODE 配置为“Dual Mode”时，保留 10/11 组合；当 SSI_SPI_MODE 配置为“Quad Mode”时，保留 11 组合。 值： <ul style="list-style-type: none"> 0x0 (STD_SPI_FRF): 标准 SPI 帧格式 0x1 (DUAL_SPI_FRF): Dual SPI 帧格式 0x2 (QUAD_SPI_FRF): Quad SPI 帧格式 0x3 (OCTAL_SPI_FRF): Octal SPI 帧格式 存在：一直 复位值：0x0
20:16	DFS_32	R/W	32 位传输大小模式下的数据帧大小。用于选择 32 位传输模式下的数据帧大小。这些位仅在 SSI_MAX_XFER_SIZE 配置为 32 时有效。当数据帧大小编程为小于 32 位时，接收逻辑自动右对齐接收数据，接收 FIFO 的高位补零。需要确保在写入发送 FIFO 之前发送数据是右对齐的。发送数据时，发送逻辑

位段	名称	类型	说明
			<p>辑忽略未使用的高位。</p> <p>注:当 SSI_SPI_MODE 设置为 “Dual”、“Quad” 或 “Octal” 模式且 SPI_FRF 未设置为 2'b00 时, 如果 SPI_FRF = 0x01, DFS 值应该是 2 的倍数, 如果 SPI_FRF = 0x10, DFS 值应该是 4 的倍数, 如果 SPI_FRF = 0x11, DFS 值应该是 8 的倍数。</p> <p>值:</p> <ul style="list-style-type: none"> ■ 0x3 (FRAME_04BITS): 4 位串行数据传输 ■ 0x4 (FRAME_05BITS): 5 位串行数据传输 ■ 0x5 (FRAME_06BITS): 6 位串行数据传输 ■ 0x6 (FRAME_07BITS): 7 位串行数据传输 ■ 0x7 (FRAME_08BITS): 8 位串行数据传输 ■ 0x8 (FRAME_09BITS): 9 位串行数据传输 ■ 0x9 (FRAME_10BITS): 10 位串行数据传输 ■ 0xa (FRAME_11BITS): 11 位串行数据传输 ■ 0xb (FRAME_12BITS): 12 位串行数据传输 ■ 0xc (FRAME_13BITS): 13 位串行数据传输 ■ 0xd (FRAME_14BITS): 14 位串行数据传输 ■ 0xe (FRAME_15BITS): 15 位串行数据传输 ■ 0xf (FRAME_16BITS): 16 位串行数据传输 ■ 0x10 (FRAME_17BITS): 17 位串行数据传输 ■ 0x11 (FRAME_18BITS): 18 位串行数据传输 ■ 0x12 (FRAME_19BITS): 19 位串行数据传输 ■ 0x13 (FRAME_20BITS): 20 位串行数据传输 ■ 0x14 (FRAME_21BITS): 21 位串行数据传输 ■ 0x15 (FRAME_22BITS): 22 位串行数据传输 ■ 0x16 (FRAME_23BITS): 23 位串行数据传输 ■ 0x17 (FRAME_24BITS): 24 位串行数据传输 ■ 0x18 (FRAME_25BITS): 25 串行数据传输 ■ 0x19 (FRAME_26BITS): 26 位串行数据传输 ■ 0x1a (FRAME_27BITS): 27 位串行数据传输 ■ 0x1b (FRAME_28BITS): 28 位串行数据传输

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x1c (FRAME_29BITS): 29 位串行数据传输 ■ 0x1d (FRAME_30BITS): 30 位串行数据传输 ■ 0x1e (FRAME_31BITS): 31 位串行数据传输 ■ 0x1f (FRAME_32BITS): 32 位串行数据传输 存在：一直 复位值：0x7
15:12	CFS	R/W	控制帧大小。为 Microwire 帧格式选择控制字的长度。 值： <ul style="list-style-type: none"> ■ 0x0 (SIZE_01_BIT): 1 位控制字 ■ 0x1 (SIZE_02_BIT): 2 位控制字 ■ 0x2 (SIZE_03_BIT): 3 位控制字 ■ 0x3 (SIZE_04_BIT): 4 为控制字 ■ 0x4 (SIZE_05_BIT): 5 位控制字 ■ 0x5 (SIZE_06_BIT): 6 位控制字 ■ 0x6 (SIZE_07_BIT): 7 位控制字 ■ 0x7 (SIZE_08_BIT): 8 位控制字 ■ 0x8 (SIZE_09_BIT): 9 位控制字 ■ 0x9 (SIZE_10_BIT): 10 位控制字 ■ 0xa (SIZE_11_BIT): 11 位控制字 ■ 0xb (SIZE_12_BIT): 12 位控制字 ■ 0xc (SIZE_13_BIT): 13 位控制字 ■ 0xd (SIZE_14_BIT): 14 位控制字 ■ 0xe (SIZE_15_BIT): 15 位控制字 ■ 0xf (SIZE_16_BIT): 16 位控制字 存在：一直 复位值：0x0
11	SRL	R/W	移位寄存器环路 仅用于测试目的。当内部激活时，将发送移位寄存器输出连接到接收移位寄存器输入。 可用于串行从和串行主模式。 当 DW_apb_ssi 在回环模式下配置为从时，ss_in_n 和

位段	名称	类型	说明
			<p>ssi_clk 信号必须由外部源提供。在这种模式下，从无法生成这些信号，因为没有可返回的内容。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (TESTING_MODE): 测试模式，连接发送和接收移位寄存器 ■ 0x0 (NORMAL_MODE): 正常模式操作 <p>存在：一直</p> <p>复位值：0x0</p>
10	SLV_OE	-	<p>从输出使能。仅当 DW_apb_ssi 配置为串行从设备时才相关。当配置为串行主时，该位段没有功能。该位使能或禁用 DW_apb_ssi 串行从的 ssi_oe_n 输出设置。当 SLV_OE = 1 时，ssi_oe_n 输出永远不能激活。当 ssi_oe_n 输出控制从 TXD 输出上的三态缓冲器时，当 SLV_OE = 1 时，从 TXD 输出上始终存在高阻抗状态。</p> <p>这在主以广播模式发送时很有用（主将数据发送到所有从设备）。只有一个从可以在主 RXD 线上响应数据。这个位在复位后被使能，如果你不希望这个设备响应数据，必须通过软件禁用（当使用广播模式时）。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x1 (DISABLED): 禁用从输出。 ■ 0x0 (ENABLED): 使能从输出。 <p>存在：否</p> <p>复位值：0x0</p>
9:8	TMOD	R/W	<p>传输模式</p> <p>选择串行通信的传输模式。该字段不影响传输重复性。仅指示接收或发送数据是否有效。</p> <p>在仅发送模式下，从外部设备接收到的数据无效，并且不存储在接收 FIFO 存储器中；在下次传输时被覆盖。</p> <p>在仅接收模式下，发送的数据无效。在第一次写入发送 FIFO 之后，在传输期间重新发送相同的字。</p> <p>在发送和接收模式下，发送和接收数据都有效。传输将继续，直到发送 FIFO 为空。从外部设备接收到的数据存储在接收 FIFO 存储器中，主机处理器可以在该存储器中访问这些数据。</p> <p>在 EEPROM 读模式下，在传输控制数据时接收数据无效。当</p>

位段	名称	类型	说明
			<p>所有控制数据都发送到 EEPROM 时，接收数据变为有效，发送数据变为无效。在该模式下，发送 FIFO 中的所有数据都被视为控制数据。该传输模式仅在 DW_apb_ssi 配置为主设备时有效。</p> <p>00 - 发送和接收</p> <p>01 - 仅发送</p> <p>10 - 仅接收</p> <p>11 - EEPROM 读</p> <p>当 SSI_SPI_MODE 设置为 “Dual”、“Quad” 或 “Octal” 模式且 SPI_FRF 未设置为 2'b00 时，只有两种有效的组合：</p> <p>10 - 读</p> <p>01 - 写</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (TX_AND_RX)：发送和接收 ■ 0x1 (TX_ONLY)：仅发送模式或写 ■ 0x2 (RX_ONLY)：仅接收模式或读 ■ 0x3 (EEPROM_READ)：EEPROM 读模式 <p>存在：一直</p> <p>复位值：0x0</p>
7	SCPOL	R/W	<p>串行时钟极性</p> <p>当帧格式（FRF）设置为摩托罗拉 SPI 时有效。用于选择非活动串行时钟的极性。当 DW_apb_ssi 主未在串行总线上主动传输数据时，该时钟保持非活动状态。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (SCLK_LOW)：串行时钟的非活动状态为低。 ■ 0x1 (SCLK_HIGH)：串行时钟的非活动状态为高。 <p>存在：一直</p> <p>复位值：0x0</p>
6	SCPH	R/W	<p>串行时钟相位</p> <p>当帧格式（FRF）设置为摩托罗拉 SPI 时有效。该串行时钟相位选择串行时钟与从选择信号的关系。</p> <p>当 SCPH = 0 时，在串行时钟的第一个边沿捕获数据。当 SCPH = 1 时，串行时钟在从选择线被激活后一个周期开始切换，并</p>

位段	名称	类型	说明
			在串行时钟的第二个边沿捕获数据。 值： <ul style="list-style-type: none"> 0x0 (SCPH_MIDDLE): 串行时钟在第一个数据位的中间切换。 0x1 (SCPH_START): 串行时钟在第一个数据位的开始切换。 存在：一直 复位值：0x0
5:4	FRF	R/W	帧格式 选择传输数据的串行协议。 值： <ul style="list-style-type: none"> 0x0 (MOTOROLA_SPI): Motorola SPI 帧格式 0x1 (TEXAS_SSP): TI SSP 帧格式 0x2 (NS_MICROWIRE): National Microwire 帧格式 0x3 (RESERVED): 保留值 存在：一直 复位值：0x0
3:0	DFS	R	数据帧大小 该寄存器字段仅在 SSI_MAX_XFER_SIZE 配置为 16 时有效。如果 SSI_MAX_XFER_SIZE 配置为 32，写入该字段无效。 选择数据帧长度。当数据帧大小被编程为小于 16 位时，接收数据会自动由接收逻辑右对齐，接收 FIFO 的高位补零。 在写入发送 FIFO 之前，必须右对齐发送数据。发送数据时，发送逻辑忽略未使用的高位。 注：当 SSI_SPI_MODE 设置为 “Dual”、“Quad” 或 “Octal” 模式且 SPI_FRF 未设置为 2'b00 时，如果 SPI_FRF = 01，DFS 值应该是 2 的倍数；如果 SPI_FRF = 10，DFS 值应该是 4 的倍数；如果 SPI_FRF = 11，DFS 值应该是 8 的倍数。 值： <ul style="list-style-type: none"> 0x3 (FRAME_04BITS): 4 位串行数据传输 0x4 (FRAME_05BITS): 5 位传输数据传输 0x5 (FRAME_06BITS): 6 位传输数据传输

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x6 (FRAME_07BITS): 7 位串行数据传输 ■ 0x7 (FRAME_08BITS): 8 位串行数据传输 ■ 0x8 (FRAME_09BITS): 9 位传输数据传输 ■ 0x9 (FRAME_10BITS): 10 位串行数据传输 ■ 0xa (FRAME_11BITS): 11 位串行数据传输 ■ 0xb (FRAME_12BITS): 12 位串行数据传输 ■ 0xc (FRAME_13BITS): 13 位串行数据传输 ■ 0xd (FRAME_14BITS): 14 位传输数据传输 ■ 0xe (FRAME_15BITS): 15 位串行数据传输 ■ 0xf (FRAME_16BITS): 16 位串行数据传输 存在：一直 复位值：0x0

4.6.1.2 CTRLR1

- 寄存器名：控制寄存器 1
- 描述：该寄存器仅在 DW_apb_ssi 配置为主设备时存在。当 DW_apb_ssi 配置为串行从时，写入该位置无效；从该位置读取返回 0。控制寄存器 1 在仅接收模式下控制串行传输的结束。当 DW_apb_ssi 使能时，无法写入该寄存器。通过写入 SSIENR 寄存器来使能或禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0x4
- 存在：是

图表 4-15 CTRLR1 寄存器字段说明

位段	名称	类型	说明
31:16	RSVD_CTRLR1	R	CTRLR1 保留位 - 只读 存在：一直 复位值：0x0
15:0	NDF	R/W	数据帧数 当 TMOD = 10 或 TMOD = 11 时，该寄存器字段设置 DW_apb_ssi 连续接收的数据帧数。DW_apb_ssi 继续接收串行数据，直到接收到的数据帧数等于该寄存器值+1，这使你能够在连续传输中接收多达 64KB 的数据。 当 DW_apb_ssi 配置为串行从时，只要选择了从，传输就会

位段	名称	类型	说明
			继续。因此，当 DW_apb_ssi 配置为串行从时，该寄存器没有任何用途且不存在。 存在：一直 复位值：0x0

4.6.1.3 SSIENR

- 寄存器名：SSI 使能寄存器
- 描述：该寄存器使能和禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0x8
- 存在：一直

图表 4-16 SSIENR 寄存器字段说明

位段	名称	类型	说明
31:1	RSVD_SSIENR	R	SSIENR 保留位 - 只读 存在：一直 复位值：0x0
0	SSI_EN	R/W	SSI 使能 使能和禁用所有 DW_apb_ssi 操作。禁用时，所有串行传输都会立即停止。当设备被禁用时，发送和接收 FIFO 缓冲区都被清空。使能时，无法对某些 DW_apb_ssi 控制寄存器进行编程。禁用时，设置 ssi_sleep 输出（延迟后）以通知系统可以安全移除 ssi_clk，从而节省系统功耗。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLE)：禁用串行传输。 ■ 0x1 (ENABLED)：使能串行传输。 存在：一直 复位值：0x0

4.6.1.4 SER

- 寄存器名：从使能寄存器
- 描述：该寄存器仅在 DW_apb_ssi 配置为主设备时有效。当 DW_apb_ssi 配置为串行从时，写入该位置无效；从该位置读取返回 0。该寄存器使能来自 DW_apb_ssi 主的各个从选择输出线。

DW_apb_ssi 主要上提供多达 16 个从选择输出引脚。当 SSI_EN=0 时，可以设置或清除寄存器位。如果 SSI_EN=1，可以设置（以在填充发送 FIFO 时延迟从选择断言）但不能删除寄存器位。

- 大小：32 位
- 偏移量：0x10
- 存在：是

图表 4-17 SER 寄存器字段说明

位段	名称	类型	说明
31:4	RSVD_SER	R	SER 保留位 - 只读 存在：一直 复位值：0x0
3:0	SER	R/W	从选择使能标志 该寄存器中的每个位对应于来自 DW_apb_ssi 主的从选择线 (ss_x_n)。当该寄存器中的某个位被设置为 1 且串行传输开始时，来自主的相应从选择线被激活。应该注意的是，在开始传输之前，设置或清除该寄存器中的位对相应的从选择输出没有影响。在开始传输之前，应该使能该寄存器中对应于主要与之通信的从设备的位。当不在广播模式下运行时，该字段只应设置一位。 值： <ul style="list-style-type: none"> ■ 0x0 (NOT_SELECTED): 未选择从。 ■ 0x1 (SELECTED): 已选择从。 存在：一直 复位值：0x0

4.6.1.5 BAUDR

- 寄存器名：波特率选择
- 描述：该寄存器仅在 DW_apb_ssi 配置为主设备时有效。当 DW_apb_ssi 配置为串行从时，写入该位置无效；从该位置读取返回 0。该寄存器导出调节数据传输的串行时钟频率。该寄存器中的 16 位字段定义 ssi_clk 分频器值。当 DW_apb_ssi 使能时，无法写入该寄存器。通过写入 SSIENR 寄存器使能和禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0x14
- 存在：是

图表 4-18 BAUDR 寄存器字段说明

位段	名称	类型	说明
31:16	RSVD_BAUDR	R	BAUDR 保留位 - 只读 存在：一直 复位值：0x0
15:0	SCKDV	R/W	SSI 时钟分频器 该字段的 LSB 始终设置为 0 且不受写操作的影响，从而确保该寄存器中的数值是偶数。如果值为 0，则禁用串行输出时钟 (sclk_out)。sclk_out 的频率由以下公式得出： $F_{sclk_out} = F_{ssi_clk} / SCKDV$ 其中 SCKDV 是 2~65534 之间的任何偶数值。 例如： 对于 $F_{ssi_clk} = 3.6864\text{MHz}$ 和 $SCKDV = 2$ ， $F_{sclk_out} = 3.6864 / 2 = 1.8432\text{MHz}$ 存在：一直 复位值：0x0

4.6.1.6 TXFTLR

- 寄存器名：发送 FIFO 阈值级别
- 描述：该寄存器控制发送 FIFO 存储器的阈值。通过写入 SSIENR 寄存器来使能或禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0x18
- 存在：一直

图表 4-19 TXFTLR 寄存器字段说明

位段	名称	类型	说明
31:8	RSVD_TXFTLR	R	TXFTLR 保留位 - 只读 存在：一直 复位值：0x0
7:0	TFT	R/W	发送 FIFO 阈值。 控制发送 FIFO 控制器触发中断的条目级别（或更低）。FIFO 深度可在 2~256 范围内配置。该寄存器的大小取决于访问 FIFO 所需的地址位数。如果你尝试将该值设置为大于或等于 FIFO 的深度，该字段不会被写入并保留其当前值。当发送

位段	名称	类型	说明
			<p>FIFO 条目数小于或等于该值时，触发发送 FIFO 空中断。有关发送 FIFO 阈值的信息，请参阅《DW_apb_ssi 数据手册》中的“主 SPI 和 SSP 串行传输”。</p> <p>当发送 FIFO 中存在 TFT 或更少数据条目时，使能 ssi_txe_intr。</p> <p>存在：一直</p> <p>复位值：0x0</p>

4.6.1.7 RXFTLR

- 寄存器名：接收 FIFO 阈值级别
- 描述：该寄存器控制接收 FIFO 存储器的阈值。通过写入 SSIENR 寄存器来使能或禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0x1c
- 存在：一直

图表 4-20 RXFTLR 寄存器字段说明

位段	名称	类型	说明
31:8	RSVD_RXFTLR	R	<p>RXFTLR 保留位 - 只读</p> <p>存在：一直</p> <p>复位值：0x0</p>
7:0	RFT	R/W	<p>接收 FIFO 阈值</p> <p>控制接收 FIFO 控制器触发中断的条目级别（或更高）。FIFO 深度可在 2~256 范围内配置。该寄存器的大小取决于访问 FIFO 所需的地址位数。如果你尝试将该值设置为大于 FIFO 的深度，该字段不会被写入并保留其当前值。当接收 FIFO 条目数大于或等于该值+1 时，触发接收 FIFO 满中断。有关接收 FIFO 阈值的信息，请参阅《DW_apb_ssi 数据手册》中的“主 SPI 和 SSP 串行传输”。</p> <p>当接收 FIFO 中存在 RFT 或更多数据条目时，触发 ssi_rxf_intr。</p> <p>存在：一直</p> <p>复位值：0x0</p>

4.6.1.8 TXFLR

- 寄存器名：发送 FIFO 级别寄存器
- 描述：该寄存器包含发送 FIFO 存储器中有效数据条目的数量。
- 大小：32 位
- 偏移量：0x20
- 存在：一直

图表 4-21 TXFLR 寄存器字段说明

位段	名称	类型	说明
31:9	RSVD_TXFLR	R	TXFLR 保留位 - 只读 存在：一直 易失性：是 复位值：0x0
8:0	TXTFL	R	发送 FIFO 级别 包含发送 FIFO 中有效数据条目的数量。 存在：一直 易失性：是 复位值：0x0

4.6.1.9 RXFLR

- 寄存器名：接收 FIFO 级别寄存器
- 描述：该寄存器包含接收 FIFO 存储器中有效数据条目的数量。这个寄存器可以被随时读取。
- 大小：32 位
- 偏移量：0x24
- 存在：一直

图表 4-22 RXFLR 寄存器字段说明

位段	名称	类型	说明
31:9	RSVD_RXFLR	R	RXFLR 保留位 - 只读 存在：一直 易失性：是 复位值：0x0
8:0	RXTFL	R	接收 FIFO 级别

位段	名称	类型	说明
			包含接收 FIFO 中有效数据条目的数量。 存在：一直 易失性：是 复位值：0x0

4.6.1.10 SR

- 寄存器名：状态寄存器
- 描述：这是一个只读寄存器，用于指示当前的传输状态、FIFO 状态以及任何可能发生的传输/接收错误。状态寄存器可以被随时读取。该寄存器中的所有位都不会请求中断。
- 大小：32 位
- 偏移量：0x28
- 存在：一直

图表 4-23 SR 寄存器字段说明

位段	名称	类型	说明
31:7	RSVD_SR	R	SR 保留位 - 只读 存在：一直 易失性：是 复位值：0x0
6	DCOL	R	数据冲突错误 仅当 DW_apb_ssi 配置为主设备时才相关。如果 ss_in_n 输入被其他主使能，该位将被设置，此时 DW_apb_ssi 处于传输过程中。这会通知处理器最后一次数据传输在完成之前被暂停。该位在读取时被清除。 值： <ul style="list-style-type: none"> ■ 0x0 (NO_ERROR_CONDITION): 无错误 ■ 0x1 (TX_COLLISION_ERROR): 发送数据冲突错误 存在：是 易失性：是 复位值：0x0
5	TXE	R	传输错误 如果传输开始时 FIFO 为空，则设置。仅当 DW_apb_ssi 配置为从设备时，才能设置该位。来自先前传输的数据在 TXD

位段	名称	类型	说明
			<p>线上重新发送。该位在读取时被清除。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (NO_ERROR): 无错误 ■ 0x1 (TX_ERROR): 传输错误 <p>存在：否</p> <p>易失性：是</p> <p>复位值：0x0</p>
4	RFF	R	<p>接收 FIFO 满</p> <p>当接收 FIFO 完全满时，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。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (NOT_FULL): 接收 FIFO 未滿。 ■ 0x1 (FULL): 接收 FIFO 已滿。 <p>存在：一直</p> <p>易失性：是</p> <p>复位值：0x0</p>
3	RFNE	R	<p>接收 FIFO 非空</p> <p>当接收 FIFO 包含一个或多个条目时设置，当接收 FIFO 为空时清除。软件可以轮询该位以完全清空接收 FIFO。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (EMPTY): 接收 FIFO 为空。 ■ 0x1 (NOT_EMPTY): 接收 FIFO 不为空。 <p>存在：一直</p> <p>易失性：是</p> <p>复位值：0x0</p>
2	TFE	R	<p>发送 FIFO 空</p> <p>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。该位段不请求中断。</p> <p>值：</p> <ul style="list-style-type: none"> ■ 0x0 (NOT_EMPTY): 发送 FIFO 不为空。 ■ 0x1 (EMPTY): 发送 FIFO 为空。

位段	名称	类型	说明
			存在：一直 易失性：是 复位值：0x1
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。 值： <ul style="list-style-type: none"> ■ 0x0 (FULL): 发送 FIFO 已滿。 ■ 0x1 (NOT_FULL): 发送 FIFO 未滿。 存在：一直 易失性：是 复位值：0x1
0	BUSY	R	SSI 忙碌标志 设置时，表示正在进行串行传输；清除时表示 DW_apb_ssi 空闲或禁用。 值： <ul style="list-style-type: none"> ■ 0x0 (INACTIVE): DW_apb_ssi 空闲或禁用。 ■ 0x1 (ACTIVE): DW_apb_ssi 正在主动传输数据。 存在：一直 易失性：是 复位值：0x0

4.6.1.11 IMR

- 寄存器名：中断屏蔽寄存器
- 描述：该读/写寄存器屏蔽或使能 DW_apb_ssi 产生的所有中断。当 DW_apb_ssi 配置为从设备时，MSTIM 位段不存在。这会将复位值从串行主配置的 0x3F 更改为串行从配置的 0x1F。
- 大小：32 位
- 偏移量：0x2c
- 存在：一直

图表 4-24 IMR 寄存器字段说明

位段	名称	类型	说明
31:6	RSVD_IMR	R	IMR 保留位 - 只读 存在：一直 复位值：0x0
5	MSTIM	R/W	多主争用中断屏蔽。如果 DW_apb_ssi 配置为串行从设备，该位段不存在。 值： <ul style="list-style-type: none"> 0x0 (MASKED): ssi_mst_intr 中断被屏蔽。 0x1 (UNMASKED): ssi_mst_intr 中断未被屏蔽。 存在：是 复位值：1
4	RXFIM	R/W	接收 FIFO 满中断屏蔽 值： <ul style="list-style-type: none"> 0x0 (MASKED): ssi_rxf_intr 中断被屏蔽。 0x1 (UNMASKED): ssi_rxf_intr 中断未被屏蔽。 存在：总是 复位值：0x1
3	RXOIM	R/W	接收 FIFO 溢出中断屏蔽 值： <ul style="list-style-type: none"> 0x0 (MASKED): ssi_rxo_intr 中断被屏蔽。 0x1 (UNMASKED): ssi_rxo_intr 中断未被屏蔽。 存在：一直
2	RXUIM	R/W	接收 FIFO 下溢中断屏蔽 值： <ul style="list-style-type: none"> 0x0 (MASKED): ssi_rxu_intr 中断被屏蔽。 0x1 (UNMASKED): ssi_rxu_intr 中断未被屏蔽。 存在：一直 复位值：0x1
1	TXOIM	R/W	发送 FIFO 溢出中断屏蔽。 值：

位段	名称	类型	说明
			<ul style="list-style-type: none"> 0x0 (MASKED): ssi_txo_intr 中断被屏蔽。 0x1 (UNMASKED): ssi_txo_intr 中断未被屏蔽。 存在：一直 复位值：0x1
0	TXEIM	R/W	发送 FIFO 空中断屏蔽。 值： <ul style="list-style-type: none"> 0x0 (MASKED): ssi_txe_intr 中断被屏蔽。 0x1 (UNMASKED): ssi_txe_intr 中断未被屏蔽。 存在：一直 复位值：0x1

4.6.1.12 ISR

- 寄存器名：中断状态寄存器
- 描述：该寄存器报告 DW_apb_ssi 中断被屏蔽后的状态。
- 大小：32 位
- 偏移量：0x30
- 存在：一直

图表 4-25 ISR 寄存器字段说明

位段	名称	类型	说明
31:6	RSVD_ISR	R	ISR 保留位- 只读 存在：一直 易失性：是 复位值：0x0
5	MSTIS	R	多主争用中断状态。如果 DW_apb_ssi 配置为串行从设备，该位段不存在。 值： <ul style="list-style-type: none"> 0x0 (INACTIVE): ssi_mst_intr 中断在屏蔽后未激活。 0x1 (ACTIVE): ssi_mst_intr 中断在屏蔽后处于活动状态。 存在：是 易失性：是

位段	名称	类型	说明
			复位值: 0x0
4	RXFIS	R	接收 FIFO 满中断状态 值: <ul style="list-style-type: none"> 0x0 (INACTIVE): ssi_rxf_intr 中断屏蔽后未激活。 0x1 (ACTIVE): ssi_rxf_intr 中断屏蔽后处于活动状态。 存在: 一直 易失性: 是 复位值: 0x0
3	RXOIS	R	接收 FIFO 溢出中断状态 值: <ul style="list-style-type: none"> 0x0 (INACTIVE): ssi_rxo_intr 中断屏蔽后未激活。 0x1 (ACTIVE): ssi_rxo_intr 中断屏蔽后处于活动状态。 存在: 一直 易失性: 是 复位值: 0x0
2	RXUIS	R	接收 FIFO 下溢中断状态 值: <ul style="list-style-type: none"> 0x0 (INACTIVE): ssi_rxu_intr 中断屏蔽后未激活。 0x1 (ACTIVE): ssi_rxu_intr 中断屏蔽后处于活动状态。 存在: 一直 易失性: 是 复位值: 0x0
1	TXOIS	R	发送 FIFO 溢出中断状态 值: <ul style="list-style-type: none"> 0x0 (INACTIVE): ssi_txo_intr 中断屏蔽后未激活。 0x1 (ACTIVE): ssi_txo_intr 中断屏蔽后处于活动状态。 存在: 一直 易失性: 是 复位值: 0x0
0	TXEIS	R	发送 FIFO 空中断状态

位段	名称	类型	说明
			值： ■ 0x0 (INACTIVE): ssi_txe_intr 中断屏蔽后未激活。 ■ 0x1 (ACTIVE): ssi_txe_intr 中断屏蔽后处于活动状态。 存在：一直 易失性：是 复位值：0x0

4.6.1.13 RISR

- 寄存器名：原始中断状态寄存器
- 描述：该只读寄存器报告屏蔽前 DW_apb_ssi 中断的状态。
- 大小：32 位
- 偏移量：0x34
- 存在：一直

图表 4-26 RISR 寄存器字段说明

位段	名称	类型	说明
31:6	RSVD_RISR	R	RISR 保留位- 只读 存在：一直 易失性：是 复位值：0x0
5	MSTIR	R	多主争用原始中断状态。如果 DW_apb_ssi 配置为串行从设备，该位段不存在。 值： ■ 0x0 (INACTIVE): ssi_mst_intr 中断在屏蔽前未激活。 ■ 0x1 (ACTIVE): ssi_mst_intr 中断在屏蔽前处于活动状态。 存在：是 易失性：是 复位值：0x0
4	RXFIR	R	接收 FIFO 满原始中断状态 值： ■ 0x0 (INACTIVE): ssi_rxf_intr 中断在屏蔽前未激活。

位段	名称	类型	说明
			■ 0x1 (ACTIVE): ssi_rxf_intr 中断在屏蔽前处于活动状态。 存在：一直 易失性：是 复位值：0x0
3	RXOIR	R	接收 FIFO 溢出原始中断状态 值： ■ 0x1 (ACTIVE): ssi_rxo_intr 中断在屏蔽前未激活。 ■ 0x0 (INACTIVE): ssi_rxo_intr 中断在屏蔽前处于活动状态。 存在：一直 易失性：是 复位值：0x0
2	RXUIR	R	接收 FIFO 下溢原始中断状态 值： ■ 0x0 (INACTIVE): ssi_rxu_intr 中断在屏蔽前未激活。 ■ 0x1 (ACTIVE): ssi_rxu_intr 中断在屏蔽前处于活动状态。 存在：一直 易失性：是 复位值：0x0
1	TXOIR	R	发送 FIFO 溢出原始中断状态 值： ■ 0x0 (INACTIVE): ssi_txo_intr 中断在屏蔽前未激活。 ■ 0x1 (ACTIVE): ssi_txo_intr 中断在屏蔽前处于活动状态。 存在：一直 易失性：是 复位值：0x0
0	TXEIR	R	发送 FIFO 空原始中断状态 值： ■ 0x0 (INACTIVE): ssi_txe_intr 中断在屏蔽前未激活。 ■ 0x1 (ACTIVE): ssi_txe_intr 中断在屏蔽前处于活动状态。

位段	名称	类型	说明
			存在：一直 易失性：是 复位值：0x0

4.6.1.14 TXOICR

- 寄存器名：发送 FIFO 溢出中断清除寄存器
- 描述：发送 FIFO 溢出中断清除寄存器
- 大小：32 位
- 偏移量：0x38
- 存在：一直

图表 4-27 TXOICR 寄存器字段说明

位段	名称	类型	说明
31:1	RSVD_TXOICR	R	TXOICR 保留位- 只读 存在：一直 易失性：是 复位值：0x0
0	TXOICR	R	清除发送 FIFO 溢出中断 该寄存器反映中断的状态。读取该寄存器会清除 ssi_txo_intr 中断；写则无效。 存在：一直 易失性：是 复位值：0x0

4.6.1.15 RXOICR

- 寄存器名：接收 FIFO 溢出中断清除寄存器
- 描述：接收 FIFO 溢出中断清除寄存器
- 大小：32 位
- 偏移量：0x3c
- 存在：一直

图表 4-28 RXOICR 寄存器字段说明

位段	名称	类型	说明
31:1	RSVD_RXOICR	R	RXOICR 保留位 - 只读 存在：一直 易失性：是 复位值：0x0
0	RXOICR	R	清除接收 FIFO 溢出中断 该寄存器反映中断的状态。读取该寄存器会清除 ssi_rxo_intr 中断；写则无效。 存在：一直 易失性：是 复位值：0x0

4.6.1.16 RXUICR

- 寄存器名：接收 FIFO 下溢中断清除寄存器
- 描述：接收 FIFO 下溢中断清除寄存器
- 大小：32 位
- 偏移量：0x40
- 存在：一直

图表 4-29 RXUICR 寄存器字段说明

位段	名称	类型	说明
31:1	RSVD_RXUICR	R	RXUICR 保留位 - 只读 存在：一直 易失性：是 复位值：0x0
0	RXUICR	R	清除接收 FIFO 下溢中断。 该寄存器反映中断的状态。读取该寄存器会清除 ssi_rxu_intr 中断；写则无效。 存在：一直 易失性：是 复位值：0x0

4.6.1.17 MSTICR

- 寄存器名：多主中断清除寄存器
- 描述：多主中断清除寄存器
- 大小：32 位
- 偏移量：0x44
- 存在：一直

图表 4-30 MSTICR 寄存器字段说明

位段	名称	类型	说明
31:1	RSVD_MSTICR	R	MSTICR 保留位 - 只读 存在：一直 易失性：是 复位值：0x0
0	MSTICR	R	清除多主争用中断 该寄存器反映中断的状态。读取该寄存器会清除 ssi_mst_intr 中断；写则无效。 存在：一直 易失性：是 复位值：0x0

4.6.1.18 ICR

- 寄存器名：中断清除寄存器
- 描述：中断清除寄存器
- 大小：32 位
- 偏移量：0x48
- 存在：一直

图表 4-31 ICR 寄存器字段说明

位段	名称	类型	说明
31:1	RSVD_ICR	R	ICR 保留位- 只读 存在：一直 易失性：是 复位值：0x0
0	ICR	R	清除中断

位段	名称	类型	说明
			如果以下任何中断处于活动状态，将设置该寄存器。读取会清除 ssi_txo_intr、ssi_rxu_intr、ssi_rxo_intr 和 ssi_mst_intr 中断。写入该寄存器无效。 存在：一直 易失性：是 复位值：0x0

4.6.1.19 DMACR

- 寄存器名：DMA 控制寄存器
- 描述：该寄存器仅在 DW_apb_ssi 配置了一组 DMA 控制器接口信号 (SSI_HAS_DMA = 1) 时有效。当 DW_apb_ssi 未配置为 DMA 操作时，该寄存器将不存在，写入该寄存器地址将无效。从该寄存器地址读取将返回零。该寄存器用于使能 DMA 控制器接口操作。
- 大小：32 位
- 偏移量：0x4c
- 存在：是

图表 4-32 DMACR 寄存器字段说明

位段	名称	类型	说明
31:2	RSVD_DMCCR	R	DMACR 保留位 - 只读 存在：一直 复位值：0x0
1	TDMAE	R/W	发送 DMA 使能 该位使能/禁用发送 FIFO DMA 通道。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLE): 禁用发送 DMA。 ■ 0x1 (ENABLED): 使能发送 DMA。 存在：一直 复位值：0x0
0	RDMAE	R/W	接收 DMA 使能。 该位使能/禁用接收 FIFO DMA 通道。 值： <ul style="list-style-type: none"> ■ 0x0 (DISABLE): 禁用接收 DMA。

位段	名称	类型	说明
			■ 0x1 (ENABLED): 使能接收 DMA。 存在: 一直 复位值: 0x0

4.6.1.20 DMATDLR

- 寄存器名: DMA 发送数据级别
- 描述: 该寄存器仅在 DW_apb_ssi 配置了一组 DMA 接口信号 (SSI_HAS_DMA = 1) 时有效。当 DW_apb_ssi 未配置为 DMA 操作时, 该寄存器将不存在, 写入该寄存器地址将无效。从该寄存器地址读取将返回零。
- 大小: 32 位
- 偏移量: 0x50
- 存在: 是

图表 4-33 DMATDLR 寄存器字段说明

位段	名称	类型	说明
31:8	RSVD_DMATDLR	R	DMATDLR 保留位 - 只读 存在: 一直 复位值: 0x0
7:0	DMATDL	R/W	发送数据级别 该位段控制发送逻辑发出 DMA 请求的级别, 等于水位级别, 即: 当发送 FIFO 中的有效数据条目数等于或低于该字段值且 TDMAE = 1 时, 会生成 dma_tx_req 信号。有关 DMATDL 解码值的信息, 请参阅《DW_apb_ssi 数据手册》中的“从 SPI 和 SSP 串行传输”部分。 当发送 FIFO 中存在 DMATDL 或更少数据条目时, 产生 dma_tx_req。 存在: 一直 复位值: 0x0

4.6.1.21 DMARDLR

- 寄存器名: DMA 接收数据级别

- 描述：该寄存器仅在 DW_apb_ssi 配置了一组 DMA 接口信号（SSI_HAS_DMA = 1）时有效。当 DW_apb_ssi 未配置为 DMA 操作时，该寄存器将不存在，写入该寄存器地址将无效。从该寄存器地址读取将返回零。
- 大小：32 位
- 偏移量：0x54
- 存在：是

图表 4-34 DMARDLR 寄存器字段说明

位段	名称	类型	说明
31:8	RSVD_DMARDLR	R	DMARDLR 保留位 - 只读 存在：一直 复位值：0x0
7:0	DMARDL	R/W	接收数据级别 该位段控制接收逻辑发出 DMA 请求的级别。水位级别 DMARDL+1，即：当接收 FIFO 中的有效数据条目数等于或大于该字段值+1 且 RDMAE = 1 时，会生成 dma_rx_req 信号。有关 DMARDL 解码值的信息，请参阅《DW_apb_ssi 数据手册》中的“从 SPI 和 SSP 串行传输”部分。 当接收 FIFO 中存在 DMARDL 或更多数据条目时，产生 dma_rx_req。 存在：一直 复位值：0x0

4.6.1.22 IDR

- 寄存器名：识别寄存器
- 描述：该寄存器包含外设识别码，在配置时使用 coreConsultant 写入该寄存器。
- 大小：32 位
- 偏移量：0x58
- 存在：一直

图表 4-35 IDR 寄存器字段说明

位段	名称	类型	说明
31:0	IDCODE	R	识别码 该寄存器包含外设的识别码，在配置时使用 CoreConsultant 写入该寄存器。

位段	名称	类型	说明
			存在：一直 复位值：0xffff_ffff

4.6.1.23 SSI_VERSION_ID

- 寄存器名：coreKit 版本 ID 寄存器
- 描述：该只读寄存器存储具体的 DW_apb_ssi 组件版本。
- 大小：32 位
- 偏移量：0x5c
- 存在：一直

图表 4-36 SSI_VERSION_ID 寄存器字段说明

位段	名称	类型	说明
31:0	SSI_COMP_VERSION	R	包含 Synopsys 组件版本的十六进制表示，由版本中每个数字的 ASCII 值组成，后跟*。例如：32_30_31_2A 代表版本 2.01*。 存在：一直 复位值：0x3430_322a

4.6.1.24 DRx (for x = 0; x <= 35)

- 寄存器名：数据寄存器 x
- 描述：DW_apb_ssi 数据寄存器是一个 16/32 位（取决于 SSI_MAX_XFER_SIZE）用于发送/接收 FIFO 的读/写缓冲区。如果配置参数 SSI_MAX_XFER_SIZE 设置为 32，所有 32 位都有效，否则只有寄存器的 16 位（[15:0]）有效。读取寄存器时，会访问接收 FIFO 缓冲区中的数据。写入时，数据被移入发送 FIFO 缓冲区；只有当 SSI_EN = 1 时才会发生写操作。当 SSI_EN = 0 时，FIFO 会被复位。注：DW_apb_ssi 中的 DR 寄存器占用内存映射的 36 个 32 位地址位置，以促进 AHB 突发传输。写入任何这些地址位置与将数据从 pwwdata 总线推入发送 FIFO 具有相同的效果。从任何这些位置读取数据与将数据从接收 FIFO 弹出到 prdata 总线具有相同的效果。DW_apb_ssi 上的 FIFO 缓冲区不可寻址。
- 大小：32 位
- 偏移量：0x60
- 存在：一直

图表 4-37 DRx (for x = 0; x <= 35) 寄存器字段说明

位段	名称	类型	说明
31:0	DR	R/W	<p>数据寄存器。写入该寄存器，你必须对数据进行右对齐。读取数据自动右对齐。如果 SSI_MAX_XFER_SIZE 配置参数设置为 32，所有 32 位都有效。否则寄存器只有 16 位 ([15:0]) 有效。读取=接收 FIFO 缓冲区，写入=发送 FIFO 缓冲区。</p> <p>存在：一直</p> <p>易失性：是</p> <p>复位值：0x0</p>

4.6.1.25 RX_SAMPLE_DLY

- 寄存器名：接收采样延迟寄存器
- 描述：该寄存器仅在 DW_apb_ssi 配置有 RXD 采样延迟逻辑 (SSI_HAS_RX_SAMPLE_DELAY==1) 时有效。当 DW_apb_ssi 未配置 RXD 采样延迟逻辑时，该寄存器不存在，写入其地址位置将无效；读取其地址将返回零。该寄存器控制在 RXD 输入的实际采样发生之前延迟（从默认采样时间开始）的 ssi_clk 周期数。当 DW_apb_ssi 使能时，无法写入该寄存器。通过写入 SSIENR 寄存器来使能和禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0xf0
- 存在：是

图表 4-38 RX_SAMPLE_DLY 寄存器字段说明

位段	名称	类型	说明
31:8	RSVD_RX_SAMPLE_DLY	R	<p>SAMPLE_DLY 保留位 - 只读</p> <p>存在：一直</p> <p>复位值：0x0</p>
7:0	RSD	R/W	<p>RXD 采样延迟</p> <p>该寄存器用于延迟 RXD 输入端口的采样。每个值代表 RXD 采样上的单个 ssi_clk 延迟。注：如果该寄存器编程的值超过内部移位寄存器的深度 (SSI_RX_DLY_SR_DEPTH)，零延迟将应用于 RXD 采样。</p> <p>存在：一直</p> <p>复位值：0x0</p>

4.6.1.26 SPI_CTRLR0

- 寄存器名：SPI 控制寄存器
- 描述：该寄存器仅在 SSI_SPI_MODE 设置为 “Dual”、“Quad”或 “Octal”模式时有效。该寄存器用于控制 SPI 操作模式下的串行数据传输。该寄存器仅在 SPI_FRF 设置为 01、10 或 11 时相关。当 DW_apb_ssi 使能（SSI_EN=1）时，无法写入该寄存器。通过写入 SSIENR 寄存器来使能和禁用 DW_apb_ssi。
- 大小：32 位
- 偏移量：0xf4
- 存在：仅 QSPI

图表 4-39 SPI_CTRLR0 寄存器字段说明

位段	名称	类型	说明
31:19	RSVD_SPI_CTRLR0	R	SPI_CTRLR0 保留位 - 只读 存在：一直 复位值：0x0
18	SPI_RXDS_EN	R	读数据选通使能位 一旦该位设置为 1，DW_apb_ssi 将使用读数据选通（rxds）在 DDR 模式下捕获读取数据。 存在：一直 复位值：0x0
17	INST_DDR_EN	R	指令 DDR 使能位 这将为指令阶段使能双数据速率传输。 存在：一直 复位值：0x0
16	SPI_DDR_EN	R	SPI DDR 使能位。 这将使能 SPI 的 Dual/Quad/Octal 帧格式的双数据速率传输。 存在：一直
15:11	WAIT_CYCLES	R/W	等待周期。 Dual/Quad/Octal 模式下控制帧发送和接收数据之间的等待周期数。该值指定为 SPI 时钟周期数。有关 WAIT_CYCLES 解码值的信息，请参阅《DW_apb_ssi 数据手册》中的“增强 SPI 模式中的读操作”部分。

位段	名称	类型	说明
			存在：一直 复位值：0x0
10	RSVD_SPI_CTRLR0_10	R	CTRLR0_10 保留位 - 只读 存在：一直 复位值：0x0
9:8	INST_L	R/W	指令长度 Dual/Quad/Octal 模式指令长度（以位为单位）。 值： <ul style="list-style-type: none"> ■ 0x0 (INST_L_0): 0 位（无指令） ■ 0x1 (INST_L_1): 4 位指令 ■ 0x2 (INST_L_2): 8 位指令 ■ 0x3 (INST_L_3): 16 位指令 存在：一直 复位值：0x2
7:6	RSVD_SPI_CTRLR0_6_7	R	CTRLR0_6_7 保留位 - 只读 存在：一直 复位值：0x0
5:2	ADDR_L	R/W	地址长度 该位定义要发送的地址长度。只有将这么多位编程到 FIFO 后，才能开始传输。有关 ADDR_L 解码值的信息，请参阅《DW_apb_ssi 数据手册》中的“增强 SPI 模式中的读操作”部分。 值： <ul style="list-style-type: none"> ■ 0x0 (ADDR_L_0): 0 位地址宽度 ■ 0x1 (ADDR_L_1): 4 位地址宽度 ■ 0x2 (ADDR_L_2): 8 位地址宽度 ■ 0x3 (ADDR_L_3): 12 位地址宽度 ■ 0x4 (ADDR_L_4): 16 位地址宽度 ■ 0x5 (ADDR_L_5): 20 位地址宽度 ■ 0x6 (ADDR_L_6): 24 位地址宽度 ■ 0x7 (ADDR_L_7): 28 位地址宽度

位段	名称	类型	说明
			<ul style="list-style-type: none"> ■ 0x8 (ADDR_L_8): 32 位地址宽度 ■ 0x9 (ADDR_L_9): 36 位地址宽度 ■ 0xa (ADDR_L_10): 40 位地址宽度 ■ 0xb (ADDR_L_11): 44 位地址宽度 ■ 0xc (ADDR_L_12): 48 位地址宽度 ■ 0xd (ADDR_L_13): 52 位地址宽度 ■ 0xe (ADDR_L_14): 56 位地址宽度 ■ 0xf (ADDR_L_15): 60 位地址宽度 存在：一直 复位值：0x0
1:0	TRANS_TYPE	R/W	地址和指令传输格式 选 择 DW_apb_ssi 是 在 标 准 SPI 模 式 还 是 在 CTRLR0.SPI_FRF 字段中选定的 SPI 模式下发送指令/地址。 00 -指令和地址将在标准 SPI 模式下发送。 01-指令在标准 SPI 模式下发送, 地址以 CTRLR0.SPI_FRF 指定的模式发送。 10 -指令和地址都将以 SPI_FRF 指定的模式发送。 11 - 保留 存在：一直 复位值：0x0