



Hi309a 通用 MPU 芯片

用户指南 03-接口（上）

文档版本 02

发布日期 2025-12-12

版权所有 © 海思技术有限公司2026。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

海思技术有限公司

地址：上海市青浦区虹桥港路2号101室 邮编：201721

网址：<https://www.hisilicon.com/cn/>

客户服务邮箱：support@hisilicon.com



前言

概述

本文档主要介绍Hi309a芯片接口的各项基本功能，常见应用模式的配置方法。

本文档提供Hi309a芯片接口的应用配置方法。

读者对象

本文档主要适用于以下工程师：

- 技术支持工程师
- 单板硬件开发工程师
- 单板软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

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



修订记录

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

修订日期	版本	修订说明
2025-10-30	01	第1次正式版本发布。
2025-12-12	02	第2次正式版本发布。



手册使用约束

须知

应用芯片时，需按照芯片手册注明的功能、性能、配置步骤等信息进行软硬件开发，未按手册注明的信息进行操作，可能会出现不可预知风险。任何不按手册注明的信息进行的应用需得到芯片SE或者FAE书面认可才能进行。切勿依商用器件经验推测，否则后果自行承担，谢谢！



目录

前言.....	i
手册使用约束.....	i
1 DDRC.....	1
1.1 概述.....	1
1.2 特性.....	1
1.3 规格约束.....	4
1.4 关键术语.....	5
1.5 功能描述.....	6
1.5.1 应用框图.....	6
1.5.2 相关模块.....	6
1.5.3 支持的 DDR 容量组合.....	7
1.5.3.1 单 Rank 组合.....	7
1.5.3.2 DIMM 组合（Hi309a 不支持）.....	7
1.5.4 DMC 支持的地址映射方式.....	8
1.5.4.1 地址映射方式.....	8
1.5.4.2 RRBC 映射方式.....	9
1.5.4.3 RRCBC 映射方式.....	10
1.5.4.4 RRRCBC 映射模式.....	11
1.5.4.5 Bank 异或.....	11
1.5.5 DDRC 地址卷绕.....	12
1.5.6 单 Rank mirror 模式.....	13
1.5.7 频率及位宽匹配模式.....	13
1.5.8 Auto refresh（REF）功能.....	13
1.5.9 Self Refresh（SRE）功能.....	14
1.5.10 低功耗模式.....	14
1.5.11 SFC 功能——该功能全部使用 BPG 进行替换.....	14
1.5.12 REC 功能.....	15
1.5.13 ECC write back 功能.....	15
1.5.14 Row hammer 处理.....	16
1.5.15 数据线 CRC 校验功能.....	17
1.5.16 Data Scrambling 功能.....	17
1.5.17 Virtual Lockstep X4 DBDC 功能.....	18



1.5.18 Virtual Lockstep X4 DDDC 功能.....	18
1.5.19 Virtual Lockstep X4 DRDC 功能.....	18
1.5.20 Static Virtual Lockstep X8 DRDC.....	19
1.5.21 Memory Sparing 功能.....	19
1.5.22 Patrol Scrubbing 功能.....	19
1.5.23 Demand Scrubbing 功能.....	19
1.5.24 DDR Memory ECC 保护.....	20
1.6 性能.....	20
1.6.1 带宽目标.....	20
1.6.2 延迟目标.....	21
1.7 调试（ Debug ）.....	22
1.7.1 Memory BIST 测试.....	22
1.7.2 统计.....	23
1.7.3 Debug bus 接口.....	24
1.7.4 PHY 接口异常.....	24
1.7.5 CA parity 校验测试.....	24
1.7.6 数据 CRC 测试.....	25
1.7.7 内部 SRAM ECC 测试.....	25
1.7.8 DDR 颗粒 ECC 测试.....	25
1.7.9 RASC 校验错误注入.....	26
1.7.10 内部状态可观测.....	26
1.7.11 中断.....	26
1.8 工作方式.....	27
1.8.1 初始化.....	27
1.8.2 软复位.....	27
1.8.3 部分配置说明.....	27
1.8.3.1 部分 IO 配置说明.....	27
1.9 硬件约束.....	28
1.10 软件约束.....	28
1.10.1 软件约束列表.....	28
2 Auto-negotiation (AN).....	29
2.1 Main Features.....	29
2.2 Terminology.....	30
2.3 References.....	31
2.4 Auto-Negotiation (AN) Overview.....	31
2.4.1 Introduction.....	31
2.4.2 Connectivity.....	33
2.5 AN Data Structure in IEEE 802.3.....	34
2.5.1 DME page encoding.....	34
2.5.2 DME Base page.....	35
2.5.2.1 Selector Field.....	36
2.5.2.2 Transmitted Nonce Field.....	36



2.5.2.3 Echoed Nonce Field.....	37
2.5.2.4 Technology Ability Field.....	37
2.5.2.5 FEC Capability Field.....	38
2.5.2.6 Pause Ability Field.....	39
2.5.2.7 Remote Fault Field.....	39
2.5.2.8 Acknowledge Field.....	39
2.5.2.9 Next Page Field.....	39
2.5.3 DME Next page.....	39
2.5.3.1 Message Code Field.....	40
2.5.3.2 Unformatted Code Field.....	44
2.5.3.3 Message Page Field.....	44
2.5.3.4 Acknowledge 2 Field.....	44
2.5.3.5 Toggle Field.....	45
2.5.3.6 Acknowledge Field.....	45
2.5.3.7 Next Page Field.....	45
2.6 AN Data Structure for BAM and MSA.....	45
2.6.1 AN Data Structure for BAM.....	46
2.6.2 AN Data Structure for MSA.....	48
2.7 AN Operations.....	48
2.7.1 AN Transmit Operations.....	49
2.7.2 AN Receive Operations.....	50
2.7.3 AN Arbitration Operations.....	52
2.8 AN arbitration rule.....	53
2.8.1 Priority Resolution.....	54
2.8.2 FEC Resolution.....	55
2.8.3 Pause Resolution.....	57
2.9 Control Processor Access.....	59
2.10 Error Handling.....	59
2.11 Initialization.....	59
2.12 Debug Features.....	60
3 LT.....	62
3.1 Link Training (LT).....	62
3.2 Main Features.....	62
3.3 Terminology.....	63
3.4 References.....	64
3.5 Link Training Overview.....	64
3.5.1 Introduction.....	64
3.5.2 Connectivity.....	66
3.6 Training Frame Data Structure (548 Bytes).....	66
3.6.1 DME coder and decoder.....	67
3.6.2 Frame marker field.....	67
3.6.3 Control channel field.....	67



3.6.3.1 Coefficient update field.....	67
3.6.3.2 Status report field.....	69
3.6.4 Training pattern field.....	70
3.7 Hardware Link Training Process.....	72
3.7.1 Frame locking process.....	72
3.7.1.1 Training frame marker detection.....	72
3.7.1.2 Receiver data alignment.....	73
3.7.1.3 Training lock state machine.....	73
3.7.2 Link training process.....	75
3.7.3 Coefficient update process.....	76
3.7.3.1 Controlling SerDes cursor value for initialization or coefficients update command (HiLink H32).....	79
3.7.4 Coefficient searching algorithm.....	82
3.7.4.1 3*3 grid searching algorithm.....	82
3.8 Software Link Training Process.....	88
3.9 Control Processor Access.....	88
3.10 Error Handling.....	88
3.10.1 Exceptions for eye diagram measurement.....	88
3.10.2 Exceptions for received training frame unlocked.....	89
3.10.3 Exceptions for SerDes SPI interface access.....	89
3.11 Initialization.....	89
3.11.1 NRZ mode.....	89
3.12 Debug Features.....	90
4 HiMACV110 XXVGE 初始化.....	92
4.1 XSEC 初始化.....	94
4.1.1 XSEC ENC.....	94
4.1.2 XSEC DEC.....	96
4.2 AN_LT 初始化.....	98
4.2.1 AN_TRAIN 复位/解复位.....	98
4.2.2 AN 复位初始化.....	99
4.2.3 AN 操作流程.....	100
4.2.4 Training 复位初始化.....	101
4.2.5 硬件 Training 流程.....	104
4.2.6 软件 Training 流程.....	105
4.2.7 An_LT 大流程.....	106
4.3 MAG 初始化流程.....	106
4.4 MAC 速率切换流程.....	109
4.4.1 XGE 切换到 GE.....	109
4.4.2 GE 切换到 XGE.....	110
4.4.3 带流切换 XSEC 加解密的 bypass/非 bypass.....	111
4.4.4 带 XSEC 加解密的速率切换流程.....	112
4.5 MAC SDS 环回流程.....	115



4.6 MIB 访问规则说明.....	115
5 MAILBOX.....	116
5.1 概述.....	116
5.2 特性.....	116
5.3 功能描述.....	116
5.3.1 VF 向 PF 发送消息.....	117
5.3.2 PF 向 VF 发送消息.....	119
5.4 软件指南.....	120
5.4.1 典型使用场景初始化.....	120
6 PCIE.....	123
6.1 特性描述.....	123
6.1.1 IP Features.....	123
7 SATA.....	147
7.1 Key Terms.....	147
7.2 Overview.....	148
7.3 Features.....	149
7.4 Compliance.....	150
7.5 Function Description.....	150
7.5.1 Logic Block Diagram.....	151
7.5.2 Device Custom Content.....	152
7.5.2.1 Interrupt Mapping.....	152
7.6 Programming Interface.....	152
7.7 Application Configurations.....	152
7.7.1 SATA Data Structure.....	152
7.8 Initialization Example.....	154
7.8.1 Initialization Configuration Process.....	154
7.8.2 LED light status.....	155
8 SerDes Ctrl.....	156
8.1 概述.....	156
8.2 关键术语.....	156
8.3 功能特性.....	157
8.4 遵循协议.....	157
8.5 功能描述.....	157
8.5.1 相关模块.....	157
8.5.2 Macro.....	157
8.5.3 存储映射和访问.....	158
8.5.3.1 SerDes 基地址.....	158
8.5.3.2 SerDes Macro 的内部地址划分.....	158
8.5.4 Lane 映射.....	160
8.5.5 Lane 映射的注意事项.....	162
8.5.6 外部参考时钟.....	163



8.6 应用配置.....	163
8.6.1 SerDes 频率配置.....	163
8.6.2 Macro 掉电.....	163
8.7 初始化.....	164
9 RGMII.....	166
9.1 概述.....	166
9.2 关键术语.....	167
9.3 功能特性.....	167
9.4 遵循协议.....	168
9.5 功能描述.....	168
9.5.1 数据结构.....	168
9.5.1.1 报文数据格式.....	168
9.5.2 MII 接口信号描述.....	169
9.6 软件指南.....	169
9.6.1 发送报文.....	170
9.6.2 接收报文.....	170
9.6.3 系统内环配置方法.....	170
9.6.4 流控帧过滤.....	170
9.6.5 广播帧过滤.....	171
9.6.6 多播帧过滤.....	171
9.6.7 单播帧过滤.....	172
9.6.8 PAD 剥离.....	172
9.6.9 发送截断.....	173
9.6.10 各种环回介绍.....	174
9.6.11 软复位.....	175
9.6.12 初始化.....	176
9.6.13 RvMII 模式的配置.....	176
9.6.14 RGMII 模式的配置.....	177
9.6.15 RGMII/RvMII/MII 对接 PPE 的 LSW 配置.....	177
9.6.16 MII 的 ddr 访存地址扩展配置.....	177
9.7 性能.....	178
9.7.1 实际测试结果.....	178
9.8 DFX 应用指导.....	178
9.8.1 中断.....	178
10 RMII.....	180
10.1 概述.....	180
10.2 关键术语.....	180
10.3 特性.....	180
10.4 软件指南.....	181
10.4.1 初始化配置.....	181
10.4.2 收帧过程.....	184
10.4.2.1 中断方式收帧.....	185



10.4.2.2 查询方式收帧.....	187
10.4.3 发帧过程.....	189
10.4.3.1 中断方式发帧.....	189
10.4.3.2 查询方式发帧.....	190
10.4.4 MAC 转发配置（可选）.....	191
10.4.5 软复位配置.....	192
10.4.6 内环配置.....	192
10.4.7 中断配置.....	192
10.4.7.1 中断状态寄存器.....	192
10.4.7.2 中断使能寄存器.....	192
10.4.7.3 原始中断状态寄存器.....	192
10.4.8 RMII 接口管教复用配置.....	193
11 LOCAL BUS.....	194
11.1 概述.....	194
11.2 特性.....	194
11.3 关键术语.....	196
11.3.1 应用场景.....	196
11.3.2 接口信号.....	197
11.3.3 工作方式.....	198
11.3.3.1 灵活的时序配置.....	198
11.3.3.2 异步(Ready)模式.....	198
11.3.3.3 字节读写模式.....	199
11.3.3.4 数据字节顺序可配.....	199
11.3.3.5 地址线序和地址偏移可配.....	199
11.3.3.6 NorFLASH 写保护功能.....	200
11.3.3.7 NorFLASH 复位.....	201
11.3.3.8 Local Bus 时序参数配置.....	201
11.3.3.9 Local Bus 典型时序.....	202
11.4 工作流程.....	211
11.4.1 时钟门控.....	211
11.4.1.1 打开时钟.....	211
11.4.1.2 关闭时钟.....	212
11.4.2 软复位.....	212
11.4.3 初始化.....	212
11.4.4 基地址配置.....	212
11.4.5 固定时序模式.....	213
11.4.6 异步等待模式.....	213
11.4.7 Page 读模式.....	214
11.4.8 写保护功能.....	214
11.4.8.1 普通写保护功能.....	214
11.4.8.2 高级写保护功能.....	214
11.5 DFX.....	215



11.5.1 写保护错误查询.....	215
11.5.2 超时错误查询.....	215
11.5.3 注意事项.....	215
11.5.3.1 一致性要求.....	215
11.5.3.2 CPU 数据位宽与 localbus 接口数据位宽.....	215
12 IFC 控制器.....	217
12.1 关键术语.....	217
12.2 SPI FLASH 控制器.....	217
12.2.1 概述.....	217
12.2.2 依赖关系.....	218
12.2.3 操作步骤.....	218
12.2.4 应用说明.....	257
12.2.5 可维可测.....	257
13 NAND Controller.....	258
13.1 微架构概览.....	258
13.2 特性.....	259
13.3 关键术语.....	259
13.4 时钟与复位.....	261
13.4.1 时钟.....	261
13.4.2 复位.....	261
13.5 基本功能.....	261
13.5.1 NAND FLASH 存储结构.....	261
13.5.2 NAND FLASH 的基本操作.....	262
13.5.3 NANDC 的基本操作.....	263
13.5.4 NANDC 操作状态.....	264
13.5.5 其他功能.....	264
13.6 ECC 功能.....	265
13.6.1 ECC 错误信息查询.....	265
13.6.2 ECC0 透传模式.....	265
13.6.3 空白页处理.....	266
13.7 Boot 功能.....	266
13.7.1 坏块识别.....	267
13.8 TIMEOUT 功能.....	267
13.8.1 DQS 超时.....	268
13.8.2 RB 超时.....	268
13.9 手动配置操作功能（NFC_OP）.....	269
13.9.1 可配置操作要素.....	269
13.9.2 相关配置.....	269
13.9.3 多种读 Page 方式.....	270
13.10 自动配置操作功能（NFC_OP_CTRL）.....	271
13.10.1 操作类型.....	271
13.10.2 自动查询器件状态.....	272



13.10.3 多种读 Page 方式.....	272
13.11 Interleave 功能.....	273
13.12 中断处理.....	273
13.13 数据结构.....	274
13.13.1 4bit ECC 模式(纠错 8bit/1KB).....	276
13.13.1.1 2KB Page_size.....	276
13.13.1.2 4KB Page_size.....	276
13.13.2 8bit ECC 模式(纠错 16bit/1KB).....	276
13.13.2.1 2KB Page_size.....	277
13.13.2.2 4KB Page_size.....	277
13.13.3 24bit ECC 模式(纠错 24bit/1KB).....	277
13.13.3.1 2KB Page_size.....	277
13.13.3.2 4KB Page_size.....	278
13.13.3.3 8KB Page_size.....	278
13.13.4 28bit ECC 模式(纠错 28bit/1KB).....	278
13.13.4.1 2KB Page_size.....	278
13.13.4.2 4KB Page_size.....	279
13.13.4.3 8KB Page_size.....	279
13.13.5 40bit ECC 模式(纠错 40bit/1KB).....	279
13.13.5.1 8KB Page_size.....	279
13.13.5.2 16KB Page_size.....	280
13.13.6 64bit ECC 模式(纠错 40bit/1KB).....	280
13.13.6.1 8KB Page_size.....	280
13.13.6.2 16KB Page_size.....	280
13.14 典型应用配置.....	281
13.14.1 NANDC 软复位流程.....	281
13.14.2 初始化.....	281
13.14.3 NANDC 操作结束判断流程.....	282
13.14.4 器件复位流程.....	282
13.14.5 读 ID 流程.....	282
13.14.6 读器件状态流程.....	283
13.14.6.1 软件配置使能读器件状态操作流程.....	283
13.14.6.2 使用 RD_ST 使能读取状态寄存器的操作流程.....	283
13.14.7 接口模式切换流程.....	283
13.14.7.1 异步切换到 NV_DDR.....	284
13.14.7.2 NV_DDR 切换到异步模式.....	284
13.14.8 擦除操作流程.....	285
13.14.8.1 NFC_OP 擦除.....	285
13.14.8.2 NFC_OP_CTRL 擦除/two plane 擦除.....	285
13.14.9 写操作流程.....	285
13.14.9.1 NFC_OP 写.....	285
13.14.9.2 NFC_OP_CTRL 写/two plane 写/FLASH cache 写.....	286



13.14.10 读操作流程.....	287
13.14.10.1 NFC_OP 读.....	287
13.14.10.2 NFC_OP_CTRL 读/two plane 读/FLASH cache 读.....	288
13.14.11 只读 OOB 操作流程.....	288
13.14.11.1 NFC_OP 只读 OOB.....	288
13.14.11.2 NFC_OP_CTRL 只读 OOB/two plane 只读 OOB.....	289
13.14.12 ECC0 透传操作流程.....	289
13.14.12.1 NFC_OP 的 ECC0 透传操作.....	289
13.14.12.2 NFC_OP_CTRL 的 ECC0 透传操作.....	289
13.14.13 Interleave 操作流程.....	290
13.15 性能和功耗调优.....	290
13.16 注意事项.....	291
14 eMMC/SD.....	292
14.1 特性.....	292
14.2 功能描述.....	293
14.2.1 整体结构.....	293
14.2.1.1 原有逻辑结构.....	294
14.2.1.2 修改后的逻辑结构简图.....	294
14.2.2 PHY 模块内部结构.....	295
14.3 SD_EMMC 时序参数.....	296
14.3.1 MMC backward-compatible card interface timing.....	296
14.3.1.1 输入时序参数.....	296
14.3.1.2 输出时序参数.....	297
14.3.2 MMC HS (52M).....	297
14.3.2.1 输入时序参数.....	297
14.3.2.2 输出时序参数.....	297
14.3.3 MMC DDR-8BIT(52M).....	298
14.3.3.1 示意图.....	298
14.3.3.2 输入输出时序，区分 CMD（单沿）和 DATA（双沿）.....	298
14.3.4 MMC HS200.....	300
14.3.4.1 时钟要求.....	300
14.3.4.2 输入时序要求.....	301
14.3.5 输出时序.....	302
14.3.6 MMC HS400.....	303
14.3.6.1 输入方向.....	303
14.3.7 输出方向.....	304
14.3.8 SD 卡 HS（50M）.....	306
14.3.8.1 时序要求.....	306
14.3.9 SD 卡 SDR12,SDR25，SDR50，SDR104.....	307
14.3.9.1 时钟要求.....	307
14.3.9.2 输入时序.....	308
14.3.9.3 输出时序 SDR12 SDR25 SDR50.....	309



14.3.9.4 输出时序 SDR104 最高可支持 208M.....	309
14.3.10 SD 卡 DDR50-4bit.....	310
14.3.10.1 时钟要求.....	310
14.3.10.2 输入输出的时序要求，区分 CMD 和 DATA.....	311
14.4 时钟定义.....	312
14.4.1 SDIOV2P9 内部支持 SDIO/SD/eMMC 三种协议.....	312
14.4.2 各模式下的时序要求.....	312
14.5 典型问题定位流程.....	313
14.6 软件约束.....	313



插图目录

图 1-1 DDRC 的 DDR4 单通道架构图.....	6
图 1-2 DDR4 RRBC 映射示意图.....	9
图 1-3 DDR4 RRCBC 映射示意图.....	10
图 1-4 DDR4 RRCBC 双 RANK 交织映射示意图.....	11
图 1-5 DDR4 RRCBC 映射示意图.....	12
图 1-6 地址卷绕说明.....	12
图 1-7 单 Rank Mirror 映射示意图.....	13
图 1-8 ECC write back 处理流程.....	16
图 1-9 DDR4 CRC 时序图.....	17
图 1-10 恶劣的信号翻转.....	17
图 1-11 DataScrambling 框图.....	18
图 1-12 带宽性能.....	20
图 2-1 AN module with PMA and PMD sub-layer.....	29
图 2-2 Auto-Negotiation Diagram.....	32
图 2-3 AN DME structure.....	34
图 2-4 pseudo-random circuit diagram.....	35
图 2-5 Base page AN information structure.....	35
图 2-6 transmitted nonce circuit diagram.....	37
图 2-7 Message Next Page structure.....	40
图 2-8 Unformatted Next Page structure.....	40
图 2-9 24-bit OUI structure.....	41
图 2-10 32-bit AN Device identifier structure.....	42
图 2-11 The Most Significant 22-bit Data (bit31:bit10) of AN Device Identifier.....	42
图 2-12 Toggle bit usage example.....	45
图 2-13 Auto-negotiation Next Page – OUI extended.....	48
图 2-14 Auto-negotiation OUI Extended Next Page Codes.....	48
图 2-15 AN transmit state diagram.....	49
图 2-16 AN receive state diagram.....	51
图 2-17 AN arbitration state diagram.....	52
图 2-18 MSA Priority Resolution.....	55
图 3-1 Link training module defined in PMD sub-layer.....	62
图 3-2 Link Training Diagram.....	65
图 3-3 Training Frame Data Structure (548 Bytes).....	67



图 3-4 3-tap FIR Filter Structure.....	68
图 3-5 PRBS11 pattern generator.....	70
图 3-6 PRBS pattern generator for polynomial_0.....	71
图 3-7 Training lock diagram.....	74
图 3-8 Training flow diagram (548Bytes).....	75
图 3-9 Coefficient update flow diagram (548Bytes).....	77
图 3-10 Coefficient update flow diagram (4168Bytes).....	78
图 3-11 Initial eye-margin 3*3 grid and measuring path.....	82
图 3-12 Center, edge and corner.....	84
图 4-1 SC_LGE_RESET_REQ Register.....	92
图 4-2 SC_LGE_CFG_RESET_REQ Register.....	92
图 4-3 SC_LGE_CORE_RESET_REQ Register.....	92
图 4-4 LGE 软复位去请求控制寄存器.....	92
图 4-5 GE 时钟使能寄存器.....	93
图 4-6 SC_LGE_CFG_RESET_DREQ Register.....	93
图 4-7 SC_LGE_CORE_RESET_DREQ Register.....	94
图 4-8 SC_PPE_RESET_REQ3 Register.....	94
图 4-9 SC_PPE_RESET_DREQ3 Register.....	95
图 4-10 SC_PPE_RESET_REQ3 Register.....	96
图 4-11 SC_PPE_RESET_DREQ3 Register.....	96
图 4-12 PPE 软复位请求控制寄存器.....	98
图 4-13 PPE 软复位去请求控制寄存器.....	98
图 4-14 PPE 软复位请求控制寄存器.....	98
图 4-15 PPE 软复位去请求控制寄存器.....	98
图 4-16 AN_CTRL_CFG.....	99
图 4-17 AN 完成中断使能寄存器.....	99
图 4-18 自协商使能和完成配置寄存器.....	99
图 4-19 MAC/AN_TRAIN 统计寄存器读清控制.....	104
图 4-20 PPE 时钟使能寄存器 0.....	106
图 4-21 PPE 时钟使能寄存器 1.....	106
图 4-22 PPE 时钟使能寄存器 2.....	107
图 4-23 PPE 时钟使能寄存器 3.....	107
图 4-24 PPE 时钟使能寄存器 4.....	107
图 4-25 PPE 软复位请求控制寄存器 0.....	107
图 4-26 PPE 软复位请求控制寄存器 1.....	107
图 4-27 PPE 软复位请求控制寄存器 2.....	107
图 4-28 PPE 软复位请求控制寄存器 3.....	108
图 4-29 PPE 软复位去请求控制寄存器 0.....	108
图 4-30 PPE 软复位去请求控制寄存器 1.....	108
图 4-31 PPE 软复位去请求控制寄存器 2.....	108
图 4-32 PPE 软复位去请求控制寄存器 3.....	108
图 4-33 MAC 通道配置寄存器.....	109



图 4-34 MAC 通道配置寄存器.....	109
图 4-35 MAC 类型配置寄存器.....	110
图 4-36 MAC 类型配置寄存器.....	111
图 4-37 IGU 和 EGU MAC 通道异常中断状态寄存器.....	111
图 4-38 按通道 bypass xsec 的解密路径.....	111
图 4-39 按通道 bypass xsec 的加密路径.....	112
图 4-40 IGU 和 EGU MAC 通道异常中断状态寄存器.....	112
图 4-41 按通道 bypass xsec 的解密路径.....	112
图 4-42 按通道 bypass xsec 的加密路径.....	112
图 4-43 XSEC_ENC_OADP_PORT_BUF_CFG{0..25}.....	113
图 4-44 XSEC_ENC_IADP_PORT_BUF_CFG{0..25}.....	113
图 4-45 XSEC_ENC_PORT_CREDIT_TH{0..25}.....	113
图 4-46 XSEC_DEC_OADP_PORT_BUF_CFG{0..25}.....	113
图 4-47 XSEC_DEC_IADP_PORT_BUF_CFG{0..25}.....	113
图 4-48 XSEC_DEC_CALENDAR_TIME_SLOT{0..127}.....	114
图 4-49 XSEC_ENC_CALENDAR_TIME_SLOT{0..127}.....	114
图 4-50 XSEC_DEC_IADP_CALENDAR_LOAD_EN.....	114
图 4-51 XSEC_ENC_IADP_CALENDAR_LOAD_EN.....	114
图 4-52 PPF 时钟禁止寄存器.....	115
图 4-53 PPF 时钟使能寄存器.....	115
图 5-1 VF 向 PF 发送消息流程图.....	117
图 5-2 PF 向 VF 发送消息流程图.....	119
图 5-3 两制式下模块示意图.....	120
图 5-4 PF 向 VF 发送消息交互图.....	121
图 5-5 VF 向 PF 发送消息交互图.....	122
图 7-1 The System Architecture of SATAH.....	149
图 7-2 SATAH Structure Diagram.....	151
图 8-1 Hilink Macro 的内部地址分配.....	159
图 8-2 SerDes Lane 复用关系.....	160
图 8-3 MAC 接口排布.....	162
图 8-4 SerDes 初始化流程图.....	164
图 9-1 MII 连接示意图.....	166
图 9-2 802.3 帧格式.....	169
图 9-3 GMAC 的各种环回.....	174
图 9-4 RGMII 时钟状态寄存器.....	175
图 9-5 RGMII 时钟状态寄存器.....	175
图 9-6 RGMII 复位状态寄存器.....	175
图 9-7 RGMII 复位状态寄存器.....	175
图 9-8 MII CRG 寄存器.....	177
图 9-9 MII CRG 寄存器.....	177
图 9-10 SC_CTRL_RGMII 寄存器.....	177
图 9-11 RGMII_DAW 寄存器.....	178



图 10-1 中断方式收帧流程.....	186
图 10-2 查询方式收帧流程图.....	188
图 10-3 CPU 发帧流程图.....	190
图 11-1 Local Bus 的典型应用框图.....	196
图 11-2 16bit cf_addr_offset=1 时.....	200
图 11-3 8bit 或 16bit (cf_addr_offset=0) 时.....	200
图 11-4 Local Bus 不复用无外部 Ready 单次读操作.....	202
图 11-5 Local Bus 不复用无外部 Ready 单次写操作.....	203
图 11-6 Local Bus 不复用无外部 Ready 连续读操作.....	203
图 11-7 Local Bus 不复用无外部 Ready 连续写操作.....	203
图 11-8 Local Bus 不复用无外部 Ready 读写切换操作.....	203
图 11-9 Local Bus 不复用无外部 Ready 写读切换操作.....	204
图 11-10 Local Bus 不复用无外部 Ready Page 读操作.....	204
图 11-11 Local Bus 不复用有外部 Ready 单次读操作.....	204
图 11-12 Local Bus 不复用有外部 Ready 单次写操作.....	205
图 11-13 Local Bus 不复用有外部 Ready 连续读操作.....	205
图 11-14 Local Bus 不复用有外部 Ready 连续写操作.....	205
图 11-15 Local Bus 不复用有外部 Ready 读写操作.....	206
图 11-16 Local Bus 不复用有外部 Ready 写读切换操作.....	206
图 11-17 Local Bus 不复用有外部 Ready 外部 TA 超时.....	206
图 11-18 Local Bus 复用无外部 Ready 单次读操作.....	207
图 11-19 Local Bus 复用无外部 Ready 单次写操作.....	207
图 11-20 Local Bus 复用无外部 Ready 连续读操作.....	207
图 11-21 Local Bus 复用无外部 Ready 连续写操作.....	208
图 11-22 Local Bus 复用无外部 Ready 读写切换操作.....	208
图 11-23 Local Bus 复用无外部 Ready 写读切换操作.....	208
图 11-24 Local Bus 复用无外部 Ready Page 读操作.....	209
图 11-25 Local Bus 复用有外部 Ready 单次读操作.....	209
图 11-26 Local Bus 复用有外部 Ready 单次写操作.....	209
图 11-27 Local Bus 复用有外部 Ready 连续读操作.....	210
图 11-28 Local Bus 复用有外部 Ready 连续写操作.....	210
图 11-29 Local Bus 复用有外部 Ready 读写切换操作.....	210
图 11-30 Local Bus 复用有外部 Ready 写读切换操作.....	211
图 11-31 Local Bus 复用有外部 Ready 外部 Ready 超时.....	211
图 12-1 总线访问写保护.....	255
图 12-2 软件访问写保护.....	256
图 12-3 黑名单保护.....	256
图 13-1 NANDC 连接示意图.....	258
图 13-2 NAND FLASH 存储结构.....	262
图 13-3 NAND FLASH 的写操作示意图.....	263
图 13-4 NAND FLASH 的读操作示意图.....	263
图 13-5 几种典型操作的分解图.....	264



图 13-6 Boot 空间映射关系图.....	266
图 13-7 多点启动 boot 空间映射关系图.....	267
图 13-8 NFC_OP 的操作要素.....	269
图 13-9 Interleave 调度模式与普通模式对比.....	273
图 13-10 4bit ECC 模式 2K Page_size 的数据结构.....	276
图 13-11 4bit ECC 模式 4K Page_size 的数据结构.....	276
图 13-12 8bit ECC 模式 2K Page_size 的数据结构.....	277
图 13-13 8bit ECC 模式 4K Page_size 的数据结构.....	277
图 13-14 24bit ECC 模式 2K Page_size 的数据结构.....	277
图 13-15 24bit ECC 模式 4K Page_size 的数据结构.....	278
图 13-16 24bit ECC 模式 8K Page_size 的数据结构.....	278
图 13-17 28bit ECC 模式 2K Page_size 的数据结构.....	278
图 13-18 28bit ECC 模式 4K Page_size 的数据结构.....	279
图 13-19 28bit ECC 模式 8K Page_size 的数据结构.....	279
图 13-20 40bit ECC 模式 8K Page_size 的数据结构.....	279
图 13-21 40bit ECC 模式 16K Page_size 的数据结构.....	280
图 13-22 64bit ECC 模式 8K Page_size 的数据结构.....	280
图 13-23 64bit ECC 模式 16K Page_size 的数据结构.....	281
图 14-1 Block Diagram.....	294
图 14-2 逻辑结构简图.....	294
图 14-3 PHY 模块内部结构图.....	295
图 14-4 输入时序参数.....	296
图 14-5 输出时序参数.....	297
图 14-6 输入时序参数.....	297
图 14-7 输出时序参数.....	297
图 14-8 时序图.....	298
图 14-9 HS200 时钟信号时序.....	300
图 14-10 HS200 设备输入时序.....	301
图 14-11 HS200 Device output timing.....	302
图 14-12 HS400 Device Data input timing.....	303
图 14-13 HS400 Device output timing.....	304
图 14-14 Clock Signal Timing.....	307
图 14-15 Card Input Timing.....	308
图 14-16 Output Timing of Fixed Data Window.....	309
图 14-17 Output Timing of Variable Data Window.....	309
图 14-18 Clock Signal Timing.....	310
图 14-19 Timing Diagram DAT Inputs/Outputs Referenced to CLK in DDR50 Mode.....	311



表格目录

表 1-1 约束规格列表.....	4
表 1-2 MDDRC 关键术语描述.....	5
表 1-3 DDRC 支持的 DDR4 器件组合.....	7
表 1-4 DDR4 RDIMM 单 rank 组合.....	7
表 1-5 row-hit 下的静态延时值.....	21
表 1-6 row-miss 下的静态延时值.....	21
表 1-7 软件约束列表.....	28
表 2-1 AN Terminology.....	30
表 2-2 Selector Field Encoding.....	36
表 2-3 Technology Ability Field encoding.....	37
表 2-4 Message Code Field Encoding.....	41
表 2-5 Message Code Field Encoding.....	43
表 2-6 Link priority for 25G & 50G Ethernet Consortium.....	46
表 2-7 BAM Page Format.....	46
表 2-8 BAM MP-5 Format.....	46
表 2-9 BAM UP-1 Format.....	47
表 2-10 Priority Resolution.....	54
表 2-11 10G PHY FEC bit F0 setting based on Priority Resolution.....	55
表 2-12 10G PHY FEC Resolution.....	56
表 2-13 PAUSE Resolution.....	58
表 3-1 Link Training Terminology.....	63
表 3-2 Coefficient update field.....	68
表 3-3 Status report field.....	69
表 3-4 Different PRBS parameters for 4 different physical lane.....	71
表 3-5 Frame marker detection setting for NRZ code.....	72
表 3-6 Manchester decode Setting for NRZ code.....	72
表 3-7 Pre-cursor, post-cursor and cur-cursor changing for every position in 3*3 grid.....	83
表 3-8 Move direction based on D metric.....	86
表 6-1 PCIE IP Feature List.....	123
表 7-1 Description of SATA key terms	147
表 7-2 SPI Interrupt Mapping Table.....	152
表 7-3 Generic Host Control Offset 00h: CAP – HBA Capabilities.....	153
表 7-4 SATA controllers and corresponding system CRG registers.....	154



表 8-1 关键术语描述.....	156
表 8-2 SerDes 访问地址空间.....	158
表 8-3 SerDes subctrl 访问地址空间.....	158
表 9-1 关键术语解释.....	167
表 9-2 MII 接口信号描述.....	169
表 9-3 发送方向截断报文及统计计数相关说明（1000M，MAX_FRM_SIZE 配置值为 N）.....	173
表 9-4 发送方向截断报文及统计计数相关说明（10/100M，MAX_FRM_SIZE 配置值为 N）.....	173
表 9-5 中断信息.....	178
表 10-1 关键术语解释.....	180
表 10-2 PERI Subctrl 寄存器描述中 SC_RMII_CFG（0xF200_0000+0x2410）.....	181
表 10-3 MAC CPU 接收帧描述子数据结构.....	185
表 10-4 RMII CPU 发送帧描述子数据结构.....	189
表 11-1 Local Bus 关键术语描述.....	196
表 11-2 Local Bus 接口信号描述.....	197
表 11-3 数据位宽 16bit 时的数据值.....	199
表 11-4 外接器件的地址产生.....	199
表 11-5 cf_addr_offset 不同配置.....	199
表 11-6 Local Bus 的可配置时序参数表.....	201
表 12-1 关键术语解释.....	217
表 12-2 SFC_PROT_ADDR0~3 寄存器.....	251
表 12-3 SFC_PROT_ADDR4~15 寄存器.....	252
表 12-4 SFC_PROT_ADDR4~15 寄存器.....	252
表 12-5 SFC_PROT_SPCMD0~1 寄存器.....	253
表 12-6 SFC_PROT_CMD_LOCK 寄存器.....	254
表 12-7 SFC_PROT_ADDR_LOCK 寄存器.....	255
表 13-1 NANDC 模块关键术语描述.....	260
表 13-2 只读 OOB 的页内偏移地址.....	270
表 13-3 NFC_OP_CTRL 支持的七种操作.....	271
表 13-4 中断源类型.....	273
表 13-5 数据结构参数.....	275
表 14-1 自研 eMMC/SD SPEED MODE.....	292
表 14-2 SD_EMMC 时序参数.....	296
表 14-3 双数据速率接口时序.....	298
表 14-4 HS200 时钟信号时序.....	300
表 14-5 HS200 设备输入时序.....	301
表 14-6 191—Output timing.....	302
表 14-7 HS400 Device input timing.....	303
表 14-8 HS400 Device Output timing.....	304
表 14-9 HS400 Capacitance.....	305
表 14-10 Clock Signal Timing.....	307
表 14-11 SDR104 Input Timing.....	308
表 14-12 SDR50 Input Timing.....	309



表 14-13 Output Timing of Fixed Data Window..... 309

表 14-14 Output Timing of Variable Data Window..... 310

表 14-15 Clock Signal Timing..... 310

表 14-16 Bus Timings - Parameters Values (DDR50 mode)..... 311

表 14-17 各模式下的时序要求..... 312



1 DDRC

1.1 概述

DDRC（DDR4/5 DDR SDRAM Controller）实现对DDR4/5 DRAM的存取控制，是整个芯片的数据存储中心。DMC负责将系统内部总线的操作转换成适用于DDR的时序，并根据系统内部总线上的QoS信息来进行流量管理。

1.2 特性

DDRC具有以下特性：

协议/器件规格

- 支持DDR4。
- DDR4支持接口数据速率为：1600/1866/2133/2400/2666/2933Mbps。
- 在DRAM颗粒方面，DDR4支持2Gb, 4Gb, 8Gb, 16Gb；不支持DIMM。
- 支持x4, x8, x16类型的器件颗粒，但不支持颗粒混用。
- DDR4 DRAM最高频率：
 - 2933Mbps（芯片支持最高频点，2Rank根据SI评估支持最高频点为2666Mbps）。
- 对于DDR4，支持DDR Burst8和Burst Chop 4（仅写操作）。
- 支持on-the-fly 传输，其中，DDR4支持BC4 OTF。
- 对于DDR4，X4器件不支持DBI；X8和器件，支持读/写DBI，但写DBI和Data mask不能同时使用。
- 对于DDR4，支持读/写前导。
- 支持DDR自动刷新，刷新周期可配置，DDR5支持normal和FGR两种模式。
- 自动刷新支持Postponed模式，可推迟刷新下发，等待空闲时连续发送。
- 支持DDR器件自刷新功能。
- 支持空闲状态下，自动进入PD低功耗模式。
- 支持DDR器件的ZQ自校准。



- 解决DDR4器件的Row hammer问题。
- 支持ODT功能，DDR4支持DQ ODT。
- 支持PDA功能。
- 支持DDR4的各种training模式，包含Write Leveling，Vref training和Read Preamble Training。
- 支持Write data eye/read data eye training。
- 支持DDR4命令线时序，可优化命令传输时序；
- 支持DDR4/hPPR功能；

系统规格

- 提供APB3.0总线接口，进行寄存器管理。其中APB支持总线异步模式。
- 支持JTAG访问寄存器接口，可不通过APB总线访问寄存器。
- 支持DMC私有总线协议，兼容DFI 4.0协议，不支持DFI LP功能。对于DFI接口还需要支持读数据优化方式以优化时序。
- 支持DMC与DRAM频率比为1:2。
- DMC处理数据位宽支持模式。DRAM数据位宽支持64bit/32bit/16bit位宽等模式。具体DMC数据位宽比DRAM数据位宽支持bit[256:64]、bit[256:32]、bit[256:16]。
- 支持单Rank场景下的mirror功能。
- 保证非掉电复位的状态下，DRAM内的数据不丢失。即DDR器件在DDRC复位前进入自刷新状态，在DDRC复位后仍旧保证器件处在自刷新状态，DDR器件不会被复位。
- 支持安全访问，包含寄存器的安全访问及调试接口的安全方案。包含MCU的访问。
- 支持DMC带宽利用率统计功能，反馈HHA DMC带宽使用情况进行开环流控，提供四档（可配置率用例）。

性能规格

- 在12nm工艺的默认工作电压ssgnp0.72v（-40,+125）条件下，DMC控制器频率可达到0.9GHz。
- 支持最多个DMC命令调度队列，可以通过代码进行配置。
- 支持RRBC，RRCBC，RRRCBC地址映射方式，可根据不同的应用场景进行选择，通过Rank交织，Bank交织等方式提高访问效率。
- 支持BANKGRUOP交织访问，对于读操作支持两BG交织访问，对于写操作支持无限BG交织方式。
- 支持Bank异或功能，增加Bank轮转的概率。
- 支持匹配项目组的QoS策略，确保高优先级的命令优先发送，低优先级的命令不能饿死；对于低优先级命令，支持Timeout和优先级自适应等功能，防止访问饿死。
- 支持wr/rd>active>precharge的调度策略。
- 支持命令的读写分组功能，减小切换开销。
- 支持Active/RW命令的bypass功能，以减小latency。
- 支持Mid 优先级push功能，以响应系统优先级的提升。



- 支持Close Page、Time-limited Open Page、Self-adapt Close Page策略。
- 不支持WRAP读数据调序功能，直接按照DDR返回数据顺序。
- DDR子系统在异步模式下，DDR4接口速率为2666时，读操作的Read Latency（从HA接口发出读命令开始（包含命令的异步桥），到DDRC接口返回第一个读数据的延迟），在page hit情况下不大于46.875ns，page miss 情况下（非Bank冲突）为60.375ns。
- DDR4 Lmbench访问效率，在满压力下、2Rank、2666Mbps配置下，FRD子项效率不低于86%，RDWR子项效率不低于82%。在32core、1Rank、2933Mbps、X16、2倍刷新率配置下，FRD子项效率不低于70%。
- 2Rank，x8器件3200Mbps频点下，在读写比率2.6:1，其中读访问都是64Byte和128Byte粒度（64Byte访问比128Byte访问比例为2:1）访问；写访问中8Byte访问和16Byte访问各5%的场景下，128Byte访问60%，64Byte访问30%，混合访问效率为75%。

RAS 规格

- 支持DDR4的写CRC功能，可对写入的数据线进行CRC保护。
- 支持DDR4颗粒的CA Parity功能，对命令/地址线进行奇偶校验保护。
- 支持Data Scrambling，对写入DRAM的数据进行加扰，读出数据时解扰还原数据。
- 支持SEC/DED ECC校验，能实现1bit纠错，2bit检错功能。如果使能ECC校验，如外接64bit DDR每个DDR接口需要扩展8 bit ECC数据Memory；如外接32bit DDR但仍需要保证单沿1bit就错误能力，DDR接口需要配置7bit ECC数据Memory。
- 支持poison功能，能传输poison信息，当使能poison时，SEC/DED ECC校验算法只支持单bit纠错，不支持2bit检错（由于上游HHA不支持，当前poison建议不使用）。
- 支持 ECC Write back机制，当读数据出现可纠正错误（ECC算法的单bit错误，SDDC算法的单颗粒错误）时，会重读一次，如果可纠正，则会将纠正的数据再写入DDR，防止错误累积，重读后数据如果不可纠正则上报中断。
- 对于DDR4 的TDM架构，不支持Patrol Scrubbing，在可配置的时间间隔内，对DRAM进行读操作扫描，利用Write back机制，防止错误累积。

测试/调试/DFX

- 支持BPG功能。软件可通过配置相关寄存器对DDR器件下发MRS、PREA、ACT、AREF、SRE/SRX、ZQCL/ZQCS、WRITE/READ、PDA、WRITE PATTERN，PDE/PDX、MPC、VREFCS、VREFCA、NOP以及DDRPHY的DFIUPD等DDR4协议要求的命令。
- BPG增加多种测试和调试手段，主要包含：
 - 发送write leveling strobe。
 - 自动产生写读循环命令，支持随机pattern，支持数据自动比对，可以通过BPG满压力测试。
 - 基于bit、nibble、Byte、DQS上下沿、Rank级别的错误检测，一旦发生错误，错误状态保持。
 - Rank切换测试。
 - 发送序列需要增加loop次数可控。loop完毕的错误状态需要累积（按per-bit粒度）。



- 支持对同一段地址单写多读。
- BPG支持多种SI测试手段，包含：
 - 支持只读、只写和读写混合3种测试模式。支持Burst读写交替测试，读/写操作长度可控，读写间隔可控。
 - 支持控制每个DX IO的数据可Bypass。
 - 支持硬件可编程的数据序列产生模块（WDB），可生成PRBS-7/PRBS-15/PRBS-31/Resonance/Clock/SSN-Even/SSN-Odd序列，用于波形测试和数据margin测试。
 - 支持数据序列产生模块（Command Address Data Buffer，CADB），生成预存储的恶化地址信号ISI效应序列，用于地址margin测试。
 - 支持误码率的统计。
- BPG支持除命令列表外除了带读写数据外的其他任何命令下发。
- 支持ECC接口旁路模式。此模式下写操作和读操作可不经ECC处理。ECC旁路模式下可通过BPG功能直接访问ECC颗粒。
- 支持DDR EXMBIST（版本）测试，可自动完成DDR颗粒遍历测试。支持多种测试算法（March C-、Checkerboard等）以及用户自设定测试算法。
- 支持基于个22bit ID（可使用ID Mask映射任意几位ID）进行读写命令流量的统计。
- 支持基于个22bit ID（可使用ID Mask映射任意几位ID）进行读写命令latency统计，包含最大latency和最小latency和平均latency。
- 支持基于个22bit ID（可使用ID Mask映射任意几位ID）进行DMC缓存的命令/数据个数统计。
- 支持对DRAM读/写命令个数、active/precharge命令个数以及Bank/Rank读写切换次数进行统计。
- DDRC能够检测到PHY接口异常，并上报错误状态。
- 支持DDR4的CA parity错误注入，用于测试CA parity功能。
- 支持对DDR4的读写数据CRC错误注入，用于测试写CRC功能。
- 支持对所有颗粒的任意位进行注错，用于测试RAS及ECC相关纠错功能。
- 支持对内部所有的SRAM进行或2bit错误注入，用于测试内部所有SRAM的ECC校验功能。
- DDRC内部的所有SRAM都支持ECC校验，并支持记录SRAM ECC错误的地址。
- DRAM ECC错误支持记录第一次和最后一次的出错信息（错误地址，错误数据和期望数据）。
- 没有独立ecc颗粒的架构下，要支持控制器读取颗粒on die ecc的出错次数信息。

1.3 规格约束

表 1-1 约束规格列表

分类	编号	描述	-
协议类	1	不支持24Gbit颗粒； 不支持64Gbit颗粒（A13未用，未验证）。	未验证



分类	编号	描述	-
	2	不支持32bit+4bit的RDIMM。	未验证
	3	不支持自动自刷新功能。	未开发
	4	hPPR功能，只支持BPG发送，SFC不能发送。	未开发
系统类	1	数据位宽验证过256bit:64bit；256bit:32bit；128bit:16bit，64bit:16bit其他未验证。	未验证
	2	Apb总线不支持error传输处理。	未开发
	3	支持压缩功能，必须要支持poison功能，但无需使用poison数据。	约束要求
	4	mask只支持64bit粒度。	约束要求
	5	对于需要dmc_rdata_begin的项目，不支持虚拟lockstep。	未开发
	6	目前最大收敛频率，12FFC（0.8v）900MHz，N7+（0.75）1066MHz。	-
	7	不支持在DDR初始化结束前，发起异常热复位请求csysreq。如果发起，热复位握手会失败，但CRG要有timeout保护机制，保证热复位触发。	约束要求
RAS	1	BC4/BC8的读不支持解扰，建议对于DDR读不支持otf。	未验证
	2	SDDC算法，DDR4 TDM模式不支持。	未开发

1.4 关键术语

表1-2中的术语将在本章节表示特殊的含义。

表 1-2 MDDRC 关键术语描述

术语	描述
SEC/DED	Single error correct/Double error detect，1bit纠错，2bit检错。
PQ	Priority Queuing，绝对优先级调度。
QoS	Quantity of service，服务质量。
ODT	On die Terminate，片上终端匹配
Device	在本文中是指1个DDR颗粒，有X4，X8和X16三种颗粒。
REFI	Average periodic refresh interval，刷新命令之间的时间间隔。
RFC	Refresh cycle，刷新操作周期。



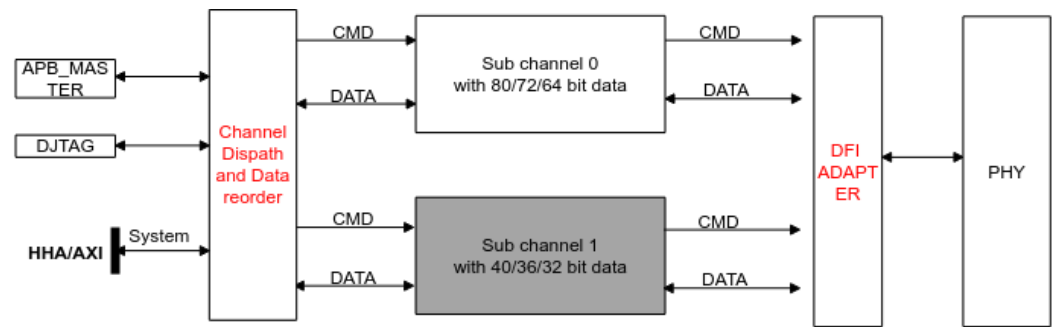
术语	描述
REF	Auto Refresh，自动刷新。
SRE	Self-refresh，自刷新。
ECC	Error checking and correction，差错校验纠正。
ECC WB	ECC write back，ECC数据回写。
CRC	Cyclic redundancy check，循环冗余校验。
CA parity	Command/Address parity，命令/地址奇偶校验。
MRS	Mode register set，模式寄存器设置。
PPR	Post package repair，接线封装修复。
TRR	Target row refresh，特定行刷新。
SFC	Soft ware config，软件配置。
SDDC	Single Device Data Correction，单颗粒错误纠正功能。

1.5 功能描述

1.5.1 应用框图

图1-1为DDRC的DDR4单通道架构图，其中DDR4是单通道架构，DDR4的贴片场景可根据性能和成本考虑选择单通道。

图 1-1 DDRC 的 DDR4 单通道架构图



1.5.2 相关模块

APB_Master：APB总线连接APB MASTER，通过APB总线进行DMC寄存器访问。

DJTAG：提供一套DJTAG接口，也可以对DMC寄存器进行访问。当不能通过APB总线访问时，提供另一套接口访问DMC寄存器。



HHA/AXI: DDRC通过HA（Home Agent）得到写命令和写数据访问DDR内存；同时通过HA的读命令读取DDR内存数据返回给HA。

PHY: dmc操作，满足DFI接口协议传送给DDR PHY。对于命令/地址及写数据相关处理，DDR PHY按照DRAM要求的时序转换成最终的DRAM信号，通过IO最终送到DRAM上。对于读操作，接收来自DRAM的返回信息，进行正确的数据采样，按照DFI协议要求送到DMC中。

1.5.3 支持的 DDR 容量组合

1.5.3.1 单 Rank 组合

DDRC支持外接X4、X8、X16器件，接口位宽支持64bit，单通道最大支持8个Rank。表1-3为DDR4各种情况下的容量搭配。

- DDR4单颗粒容量支持[256MB,2GB]区间内的所有的整数次方容量，DDR5单颗粒支持的容量为[1GB,2GB,3GB,4GB,8GB]。
- X4，X8和三种颗粒支持的单个颗粒容量类型一致，例如X4支持512MB容量，那么X8和X16也支持512MB容量。
- 支持ECC时需要额外添加颗粒，如用X4颗粒时需增加2个，用X8或者X16时需增加1个。
- 不支持颗粒混搭使用。

表 1-3 DDRC 支持的 DDR4 器件组合

DDR总线位宽	颗粒类型	颗粒数量	Rank容量 min	Rank容量 max
64 Bits	X16	4	256MB*4=1GB	2GB*4=8GB
	X8	8	256MB*8=2GB	2GB*8=16GB
	X4	16	256MB*16=4GB	2GB*16=32GB

1.5.3.2 DIMM 组合（Hi309a 不支持）

单通道最多支持2条DIMM，支持的DIMM类型有RDIMM/UDIMM。其中RDIMM最大支持2个RANK。

如表1-4所示，是数据位宽为64bit或者64bit+8bit ECC的RDIMM/UDIMM各容量组合情况。其他DIMM类似分析。

表 1-4 DDR4 RDIMM 单 rank 组合

片选数	颗粒类型	颗粒数	位宽	颗粒容量	DIMM容量
1 Rank	X8	8+1	72	256MB	2GB



片选数	颗粒类型	颗粒数	位宽	颗粒容量	DIMM容量
	X8	8+1	72	512MB	4GB
	X8	8+1	72	1GB	8GB
	X8	8+1	72	2GB	16GB
	X4	16+2	72	256MB	4GB
	X4	16+2	72	512MB	8GB
	X4	16+2	72	1GB	16GB
	X4	16+2	72	2GB	32GB

1.5.4 DMC 支持的地址映射方式

1.5.4.1 地址映射方式

上层系统传给DMC的地址会依据不同的映射规则被分解为Rank地址、Row地址、Bank地址和Column地址，每部分地址的位宽取决于所选器件的容量。由于每个DMC仅连接一个通道，因此不会出现通道地址。

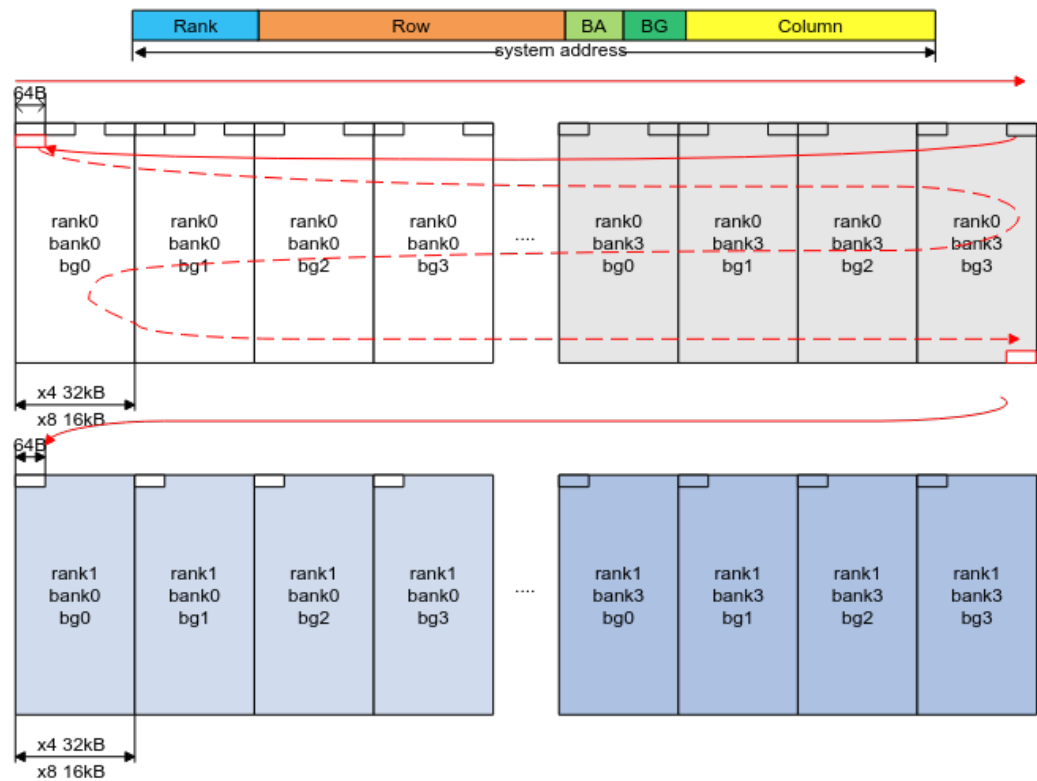
支持如下映射方式：

- Rank-Row-Bank-Col：无交织映射方式，简称RRBC。
- Rank-Row-Col-Bank-Col：Bank交织映射方式，简称RRCBC。
- Rank-Row-Rank-Col-Bank-Col：Rank交织映射方式，简称RRRCBC。



1.5.4.2 RRBC 映射方式

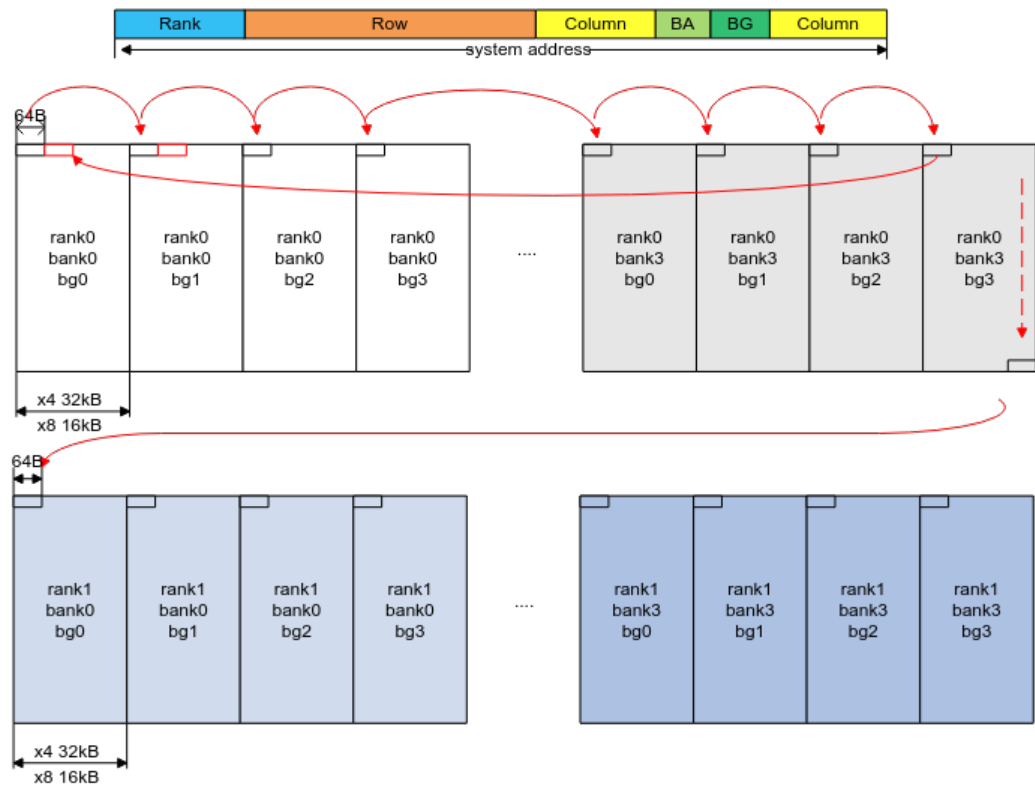
图 1-2 DDR4 RRBC 映射示意图





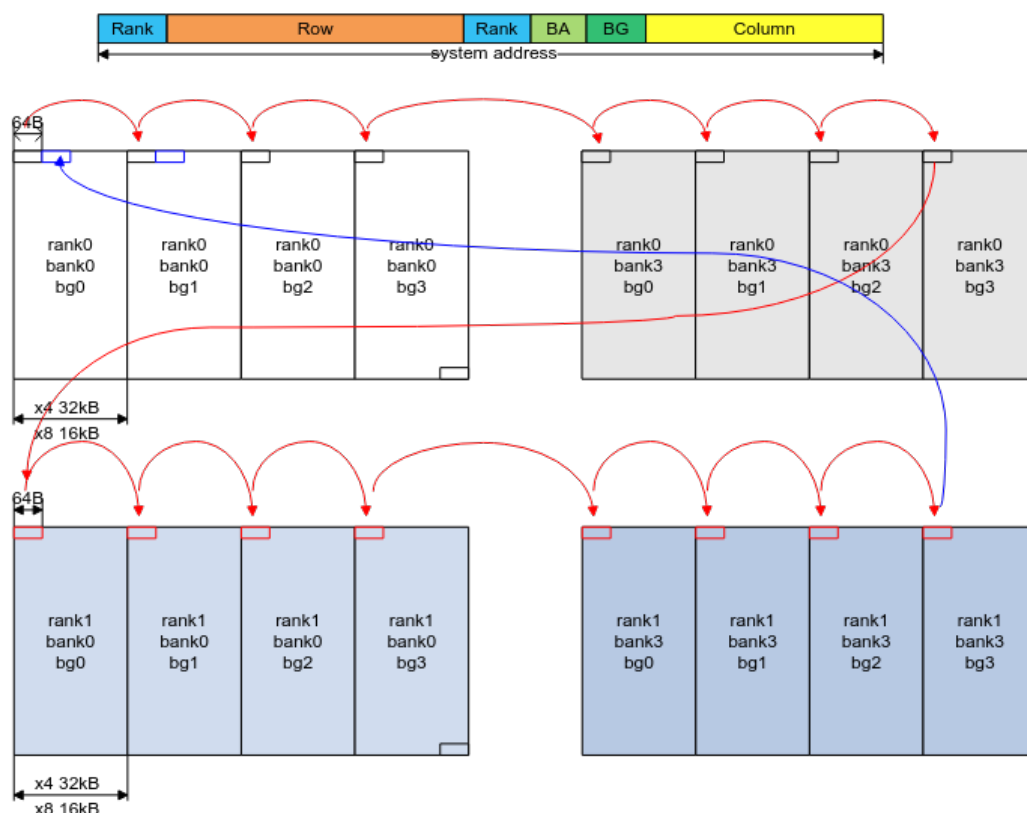
1.5.4.3 RRCBC 映射方式

图 1-3 DDR4 RRCBC 映射示意图



1.5.4.4 RRCBC 映射模式

图 1-4 DDR4 RRCBC 双 RANK 交织映射示意图



须知

RRCBC的映射方式适用于访问DDR的地址空间是连续变化的，但是程序运行起来很难保证访问DDR的地址空间是连续变化的，容易出现较高的Bank打开和关闭的概率，造成访问DDR的效率较低；

RRCBC的映射方式通过将Bank地址映射到更低位（col地址之间），这样程序运行起来的时候可以同时打开更多的Bank，减小了Bank打开和关闭的概率，也就提高了DDR的访问效率；

RRRCBC的映射方式适用于DDR rank数大于1个rank的应用，与RRCBC映射方式原理类似，将rank地址映射到低位，可以同时打开更多的Bank，减小Bank打开和关闭的概率，提高DDR访问效率。

因此，单rank的时候推荐适用RRCBC映射方式，多rank的时候推荐适用RRRCBC映射方式。

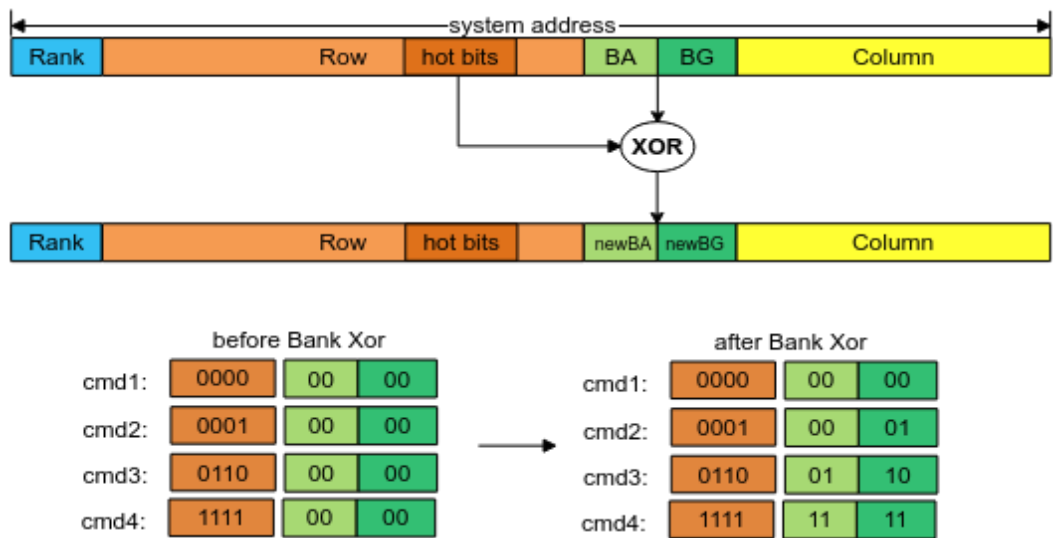
1.5.4.5 Bank 异或

如果连续的一段访问都是同Bank但不同行的，每一个访问都需要进行一次Precharge操作，之后再再来一个ACT才能发送读写。这些操作之间需要等待很长的时序，显然这样的访问的效率是很低的。

假设系统向DDR提供的访问地址中，低位部分通常是相同的，仅高位的若干bit（hot bits）跳变。通常地址映射BA&BG处于较低位，而高位地址一般都是Row。这种访问的地址经过映射就会形成同Bank不同行的访问从而降低DDR总线效率。为了提升效率，希望通过一种方法把这些命令转换成不同Bank的命令，这样可以通过Bank轮转来提高整体性能。

Bank异或就是将BA&BG地址和经常跳变的高位地址进行异或操作，产生新的BA & BG，如图1-5所示。原本4个相同的Bank地址不同hot bits的命令，经过Bank异或之后变成4个不同Bank的命令。

图 1-5 DDR4 RRCRBC 映射示意图



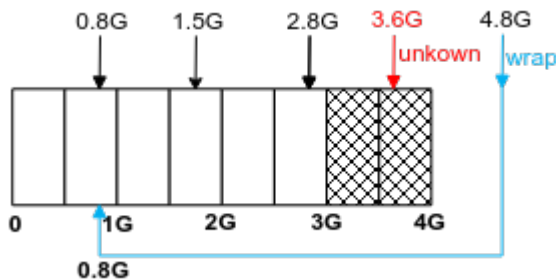
1.5.5 DDRC 地址卷绕

DDRC的地址按照 2^N 空间进行卷绕，如按照1G空间卷绕时，访问1+0.n G地址效果等同于访问0.n G地址。

如外接的DDR器件容量非 2^N ，卷绕空间仍旧按照 2^N 进行卷绕。如外接3G空间DDR，则地址仍旧会从4G物理地址开始卷绕，而非从3G地址空间进行卷绕。3~4G空间会不规则的映射到DDRC的0~3G空间，所以在软件应用时需要避免使用此段地址作为DDR的地址空间。

DDR的地址空间的起始地址必须依照 2^N 空间地址粒度进行分配。

图 1-6 地址卷绕说明

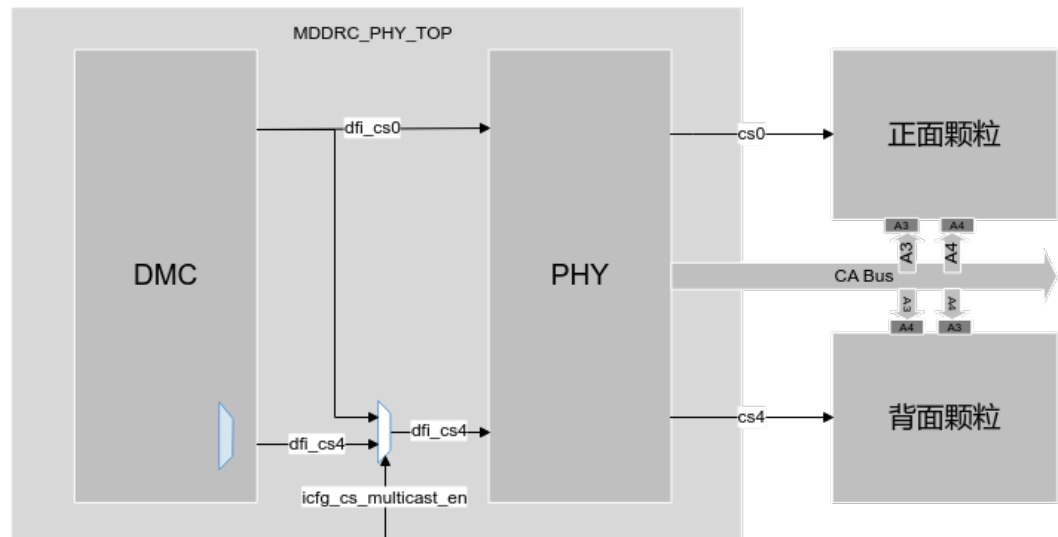


1.5.6 单 Rank mirror 模式

DDRC支持单Rank内的mirror功能：A3与A4，A5与A6，A7与A8，A11与A13，bg0与bg1，ba0与ba1在板级走线上允许交叉。

DDRC在单rank mirror功能使能或不使能的情况下都能正确初始化所有颗粒，并正确访问所有颗粒。

图 1-7 单 Rank Mirror 映射示意图



1.5.7 频率及位宽匹配模式

支持DMC和DDR频率比（clk_ratio）支持1:2模式。当clk_ratio=1:2，DMC频率为400MHz，那么DDR频率为800MHz，即数据率1600Mps。

支持64bit、128bit和256bit三种DMC数据位宽，支持64bit、32bit、16bit三种DDR数据位宽。

支持带宽匹配模式（op_mode）。当单位时间DMC传输的数据等于单位时间DDR传输的数据时，称为带宽匹配模式op_mode=0。当单位时间DMC传输的数据2倍于单位时间DDR传输的数据时，称为带宽不匹配模式op_mode=1。

1.5.8 Auto refresh（REF）功能

DDRC支持灵活的REF策略，在遵循DDR协议的情况下，兼顾效率。REF的功能如下：

- 支持三种REF策略：

刷新推迟模式：在满足协议的情况下，DDRC尽量在空闲时发送REF操作。

- 每次刷新的连续发送刷新的命令次数（1，2，4，8）可配置。
- 若延迟的REF次数超过设置的阈值，则DDRC强制发送REF。

普通模式：DDRC按隔固定的时间间隔发送REF操作。间隔时间可配置。

- 每隔tREFI下发1次REF操作。

刷新优化模式：当推迟的刷新个数到某一阈值，且命令队列中命令个数少于某一阈值时，发送刷新命令。



- 异常处理：当DDRC出现异常挂死，而无法保障REF操作的正常执行时，DDRC进入中断模式（该中断软件可配置屏蔽）。
- 支持多rank场景下的优化策略：
 - 串行模式：DDRC每次只对一个rank下发REF操作，操作完成后立即下发下一个rank的REF操作。
 - 并行模式：DDRC每次对两个rank同时下发REF操作。相对于串行模式，并行模式不用等待第一个rank的REF执行完成即可开始第二个rank的REF操作，在多rank场景下REF效率更高，但需要多个rank同时进行REF操作可能导致的电流过大问题。对于多于4个物理RANK的配置，建议启用该模式。

1.5.9 Self Refresh（SRE）功能

器件在SRE状态下，内部自动进行刷新操作以保存数据，同时减少器件的功耗。功能特性如下：

- 支持软件配置寄存器的方式，使器件进入SRE状态。
- 支持系统通过CSYSREQ信号发出请求，使器件进入SRE状态之后，等待器件进入自刷新状态后，通过CSYSACK信号返回响应给系统。
- 在SRE模式下，可以关闭器件时钟以降低功耗。
- 在SRE模式下接口有访问命令报错并上报中断。

1.5.10 低功耗模式

Rank连续空闲时间达到配置的PD触发等待时间，DDRC使器件进入PD模式。PD模式下器件仍然需要REF以保持数据，因此PD模式会由于REF操作而退出。

1.5.11 SFC 功能——该功能全部使用 BPG 进行替换

DDRC提供SFC（Soft ware config）功能，软件通过配置相关寄存器选择命令类型、准备数据。该功能允许对DRAM进行数据读写、寄存器读写等操作。SFC的主要功能如下：

1. 支持多种DRAM基本命令：例如MRS、PREA、ACT、AREF、SRE/SRX、ZQCL/ZQCS、WRITE/READ、PDA、PDE/PDX、MPC、NOP以及带写DQS功能的NOP等各种命令，此外还支持DDRPHY的PHYUPD等DDR4。
2. 支持深度为16的命令队列，软件可以预先配置多个命令至命令队列，并启动命令队列的执行。软件可在命令队列中使用基本命令组合出其他操作或流程，如：MPR read操作、初始化流程、PHY training流程、PPR流程等。
3. 支持两种执行命令队列的方式：一次执行模式和循环执行模式。
4. 一次执行模式下，DDRC依次将命令队列的所有命令执行一遍。
5. 循环模式下，DDRC循环执行命令序列中所有命令，直至软件设置停止。
6. 支持缓存一次读写操作所需数据。
7. 支持在ECC旁路模式下通过SFC方式直接访问ECC颗粒。

在以上SFC规格的基础上，新增以下SFC规格，用于training和SI测试。

1. 支持write leveling strobe信号的发送，选通脉冲宽度及间隔可配置。
2. 为了加快training速度，支持自动产生写读循环命令，支持随机pattern，支持数据自动比对，支持满压力相同类型命令连续执行。
3. 当数据比对出错的时候，支持自动停止pattern发送。



4. 支持基于nibble、Byte、DQS上下沿、rank级别的错误检测，一旦发生错误，错误状态会保留。
5. 支持RANK间切换pattern发送。
6. 发送序列需要增加loop次数可控。loop完毕的错误状态需要累积（按per-bit粒度）。
7. 支持选择某个bit/某个沿、以及所有沿的比对。
8. 每次SFC序列发送完成之后，支持相同配置可以通过配置start重新启动。
9. 支持对同一段地址单写多读。
10. 支持控制每个DX IO的数据可Bypass，如Bypass可设定处于某固定状态。
11. 支持丰富的随机码型，包括PRBS-7/PRBS-15/PRBS-31/Galois/Resonance/Clock/SSN-Even/SSN-Odd序列，用于眼图测试。
12. 支持数据序列产生模块（Command Address Data Buffer,CADB），生成预存储的恶化地址信号ISI效应序列，用于地址margin测试。
13. 支持单根bit线发送的码型和别的bit发送的码型不同，且可配置。
14. 在误码率计算的时候，mask掉对别的bit的比较，只比较关注的受害线上的误码率。

1.5.12 REC 功能

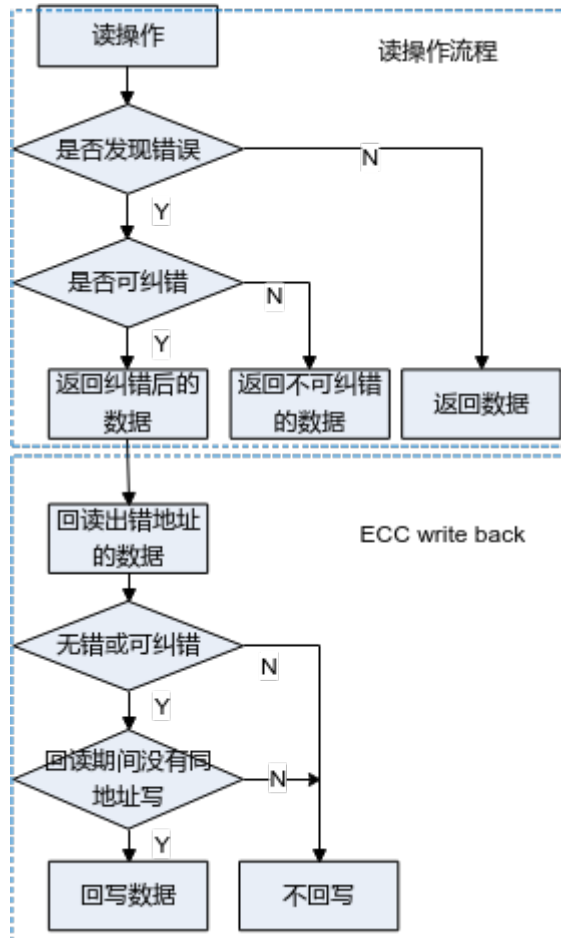
当发生CRC校验或者CA Parity校验失败发生时，DDRC可以重传错误窗口内的操作。此功能特性如下：

1. REC恢复的错误有：
 - a. CA parity错误，包含读写命令、precharge命令、acitvie命令、auto-refresh命令和进退PowerDown及ZQ命令。
 - b. 写数据CRC校验失败。
 - c. 读数据CRC校验失败。
2. 如果发现CA parity错误，在错误命令窗口如果发现不需要rec恢复的命令（除RD/WR/ACT/REF/PDE/PDX/DES/NOP/REFsb/REFab/PREsb/PREab/ZQ），则直接上报中断，不做处理。
3. 支持CRC错误和CA Parity错误的恢复：
 - a. 支持CRC错误重传。发生CRC校验错误时，DDRC恢复从CRC错误产生到最终检测到CRC错误这段时间内的读写操作（仅恢复未返回数据的读命令）。
 - b. 支持CA Parity错误重传。发生CA Parity错误时，DDRC除了恢复读写命令之外，还要恢复ACT、AREF、ZQ等命令。
4. 在恢复过程中出现再次alert_n保持时，停止恢复动作，上报错误中断。

1.5.13 ECC write back 功能

ECC write back功能用于正常的读操作中，若读操作中发现读数据出现错误，ECC write back功能会对出错的地址进行一次回读后再将数据回写到原地址。这种机制用于防止软失效引起的错误累计而导致不可纠错。其处理流程如图1-8所示。

图 1-8 ECC write back 处理流程



ECC write back的操作流程图说明：

1. ECC write back处理都是在读操作的数据返回之后进行。且只针对出现ECC可纠正错误的地址进行回读处理。
2. DDRC回写过程中需要保证数据一致性：从回读至回写的这段时间中出现了目标地址的写操作之后，DDRC不再进行回写操作。
3. ECC write back的状态信息可通过寄存器查询。

1.5.14 Row hammer 处理

随着DDR器件的制造工艺的提升，器件内部cell之间的物理间隔越来越小，row字线电平的多重重复翻转，将会加快其物理空间上邻近row的电荷泄露速度。在某个row的两次激活间隔内，若其邻近row的字线电平出现多次翻转，则可能导致这个row的cell数据出错。这种现象称为row hammer。

说明

与row hammer相关的参数： $tMAW=2048 \times tREFI$ ， $tRRMAC=300K$ （不同器件的具体值不同，需要参考器件手册）。在邻近row的两次刷新间隔中，单个row在时间窗口（ $tMAW$ ）内可以承受的最大激活次数（ $tRRMAC$ ）。也就是说某个row在 $tMAW$ 时间内，若其激活次数超过了 $tRRMAC$ ，而此段时间内其邻近的row没有进行过刷新操作，那么邻近的row就会出现数据错误。

1.5.15 数据线 CRC 校验功能

DDR4支持写数据CRC校验功能。控制器在常规写数据后附加CRC校验码，颗粒根据校验码对写数据进行校验，如果校验失败颗粒将拉低Alert_n信号通知控制器进行错误处理。

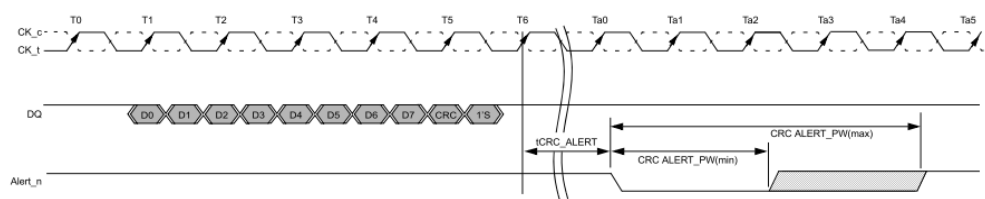
CRC（Cyclic Redundancy Check）即循环冗余校验码：是数据通信领域中最常用的一种查错校验码。通过多项式计算并将结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

颗粒使用ATM-8 HEC标准的CRC算法，生成多项式为 $X^8+X^2+X^1+1$ ，对于DDR4由72bit数据（数据+DM）生成8bit冗余校验码。如图1-9所示。对单比特错误、双比特错误、奇数比特错误、单列多比特错误都有100%的检错能力。注意，100%检错单列多比特错误时必须不包含DBI位。

DDR颗粒的CRC数据图见协议描述。

- 与CA Parity不同的是，当出现CRC错误后颗粒仍将继续执行后续命令而不是丢弃数据。错误数据是否写入取决于当前DM功能是否关闭，这主要是考虑到数据一致性的问题。
- CRC的波形如下图1-9所示，当颗粒检测到写数据存在CRC错误之后tCRC_ALERT时间内会发送一个Alert_n的脉冲，脉冲的宽度CRC_ALERT_PW最小为2个周期，最大5个周期。

图 1-9 DDR4 CRC 时序图



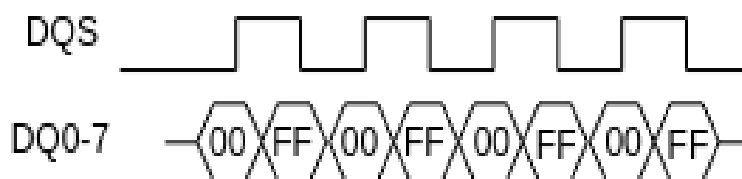
须知

由于CA parity和CRC都使用同一个alert_n管脚，而CA Parity的脉冲长度在10个周期以上，而CRC出错，alert_n脉冲比较短，所以控制器可通过脉冲长度来区分错误类型。但是当两个CRC连续发生时，可能会导致两个脉冲拼接在一起，所以脉冲宽度有可能长到10个周期以上。所以短脉冲一定是CRC，长脉冲则两种都有可能。

1.5.16 Data Scrambling 功能

如果DDR接口IO的数据从全0跳变到全1，这时可能会造成很大的瞬时电流从而增大电源噪声降低信号质量，恶劣的信号翻转如图1-10所示。

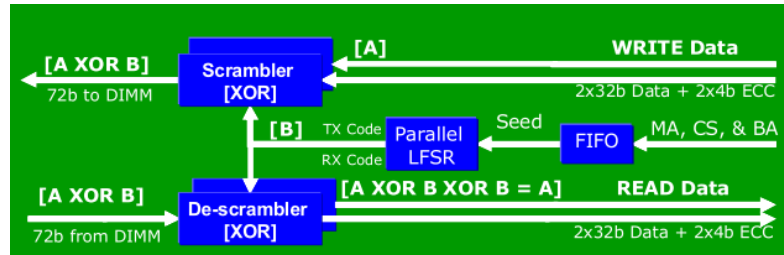
图 1-10 恶劣的信号翻转



为减少出现这种全0跳到全1的数据翻转，控制器Data Scramble支持对写数据进行加扰再进行传输，对读数据进行解扰再返回总线。

数据对应的地址经过LFSR产生加扰码，然后再对原数据进行异或操作产生加扰后的数据写入DDR。读数据时根据地址产生相同加扰码，然后对读出的数据进行异或操作恢复原本数据。

图 1-11 DataScrambling 框图



1.5.17 Virtual Lockstep X4 DBDC 功能

Virtual Lockstep模式仅需一个通道即可实现，分为静态虚拟Lockstep和动态虚拟Lockstep两种模式。在动态虚拟Lockstep模式下，未出现硬失效之前可以工作在正常模式，带宽和效率不受影响。动态虚拟Lockstep模式包括DBDC和DRDC两个功能，可分别独立使能。静态虚拟Lockstep主要用于实现X8的SDDC。

DBDC（Double Bank Data Correction）模式是基于硬失效错误只发生在了某个Device的具体1个Bank（dirty bank）内，此时会寻找另一个没有错误的Bank（buddy bank）来进行数据重分配。

经过两个Bank数据重新分配之后，原先存储在一个cache word中的数据将被拆分至两个Bank中，从而腾出一个设备用于存储失效Device的数据，并保持1个Device的纠错能力。

1.5.18 Virtual Lockstep X4 DDDC 功能

DDDC（Double Device Data Correction）模式是基于硬失效错误只发生在了某个Device内（相比DBDC错误范围扩大），然后找到另一个没有错误的Rank进行数据重分配。

经过两个rank数据重新分配之后，原本存储在一个cache word中的数据会被拆分至两个rank中，从而释放出一个Device来存放失效Device的数据，并且仍然保留1个Device的纠错能力。拆分后的两个半cache word数据会被分别送入不同的rank中，但它们在各自rank内的地址是相同的。

1.5.19 Virtual Lockstep X4 DRDC 功能

DRDC（Double Region Data Correction），指第二个出错区域处理，当进行过第一个错误区域处理后，如果发现另一个错误区域再次出错，通过DRDC处理，还具有一个Device的纠错能力。

第二个区域出错的区域分不同情况，不同情况下的处理方法不同。



1.5.20 Static Virtual Lockstep X8 DRDC

静态虚拟Lockstep模式是为了实现单通道下X8颗粒也可以就有1个Device的纠错能力，在初始化时就进行两个Rank的内存重分配。每次访问就都会被拆分到两个Rank内，访问效率会有所降低。

1.5.21 Memory Sparing 功能

内存备份功能分为两个级别，Rank Sparing和Device Sparing。这两个级别都可以将DDR某一个（Source）Rank或Device的数据内容拷贝到另外一个（target）Rank或Device中去。

Rank Sparing功能特性如下：

1. 支持Rank Sparing时需要配备多一个备份Rank，在启动Rank Sparing之前备份Rank是不可用。
2. 当某个Rank数据出错达到阈值时，启动Rank Sparing把所有数据拷贝的备份Rank中，并且后续对该Rank的所有访问都将被重定向至备份Rank。此外，也可以通过软件手动启动Rank Sparing。
3. 在Rank Sparing过程中，如果读到可纠正错误则纠正后写入备份Rank，如果读到不可纠正错误则重读一次。如仍读到不可纠正错误，则将错误数据写入备份Rank，并上报中断。
4. 在Rank Sparing过程中，系统对DDR的访问命令仍能够被执行，但latency可能会增大，在Rank Sparing完成之后恢复正常。

Device Sparing功能和Rank Sparing相似，区别在于当系统检测到某个Device的出错次数达到阈值时，除了可以将出错Device的数据复制到备份Device外，还可以选择复制到Parity Device。

1.5.22 Patrol Scrubbing 功能

内存巡检功能会在一定时间间隔内向DDR发送连续地址的读命令，保证24h内完成一次全空间的内存扫描。PS（Patrol Scrubbing）功能特性如下：

1. PS能够提早发现内存的问题，防止软失效错误的累积。
2. 当读取数据出现错误时，系统会尝试一次重读；若无法纠正错误，则报告中断；若数据读取正确或仅存在可纠正错误，则将数据写回到DDR（此功能为Demand Scrubbing/ECCWB）。
3. 由PS触发的读取操作所返回的数据最终会被丢弃。

PS功能并不依赖于校验算法，使能PS就开始内存巡检，只是使用的算法是ECC或者SDDC算法。

1.5.23 Demand Scrubbing 功能

当读命令的数据校验出现错误时，系统会记录该读地址，并尝试修复该地址上的数据，以减少后续访问时再次发生错误的概率。DS（Demand Scrubbing）功能特性如下：

1. DS能够消除偶然软失效产生的可纠正错误，从而避免错误一直存在DDR中，导致每次读都需要进行纠错，同时防止错误累积到无法纠正的地步，DS对于硬失效没有作用。
2. 无论读操作数据出错是否可纠正，都会将出错地址记录下来作进一步处理。



- 3. DS从记录下来的地址中选取一个，首先进行Retry读操作，如果Retry返回数据有不可纠正错误，则上报中断；如果Retry返回数据正确或有可纠正错误，并把正确数据写入Write Back 到DDR。
- 4. DS下发的读操作，如果发现为不可纠正错误，需要上报中断。

1.5.24 DDR Memory ECC 保护

DDRC支持64/8bit的SEC/DED ECC校验功能，单比特数据可纠错，双比特数据可检错。

在ECC保护使能的模式下，若检测到1BITS ECC数据出错，DDRC会将纠正的数据返回给HHA，若同时配置了ecc writeback使能，会将正确数据回写入DDR SDRAM。对于2bits数据出错的情况，DDRC没有纠错能力，会将错误数据直接返回给L3 Cache并不会对DDR SDRAM进行数据回写，此时需要系统读ECC错误地址寄存器，对DDR SDRAM进行数据回写。

当发生1bits ECC数据错误时，DDRC寄存器会记录ECC单bit出错的地址，错误数据以及期待数据，并统计单BIT出错的次数，同时上报单BIT错误中断。

当发生2bits ECC数据错误时，DDRC寄存器会记录ECC两bit出错的地址，错误数据，并统计两BIT出错的次数，同时上报两BIT错误中断。

1.6 性能

1.6.1 带宽目标

图 1-12 带宽性能

Pattern描述				硬件使用场景描述							带宽性能		
Pattern类型	命令长度	命令数量	读写占比	lowhit	DDR版本	频率	刷新率	控制器架构	Rank数量	颗粒位宽	利用率目标	带宽（GBps）	
随机读写	64Byte	64	67%:33%	0%	DDR4	2666	3.9us	Single channel	1Rank	X16	33%	6.87	
										X8	47%	9.79	
										X4	82%	17.08	
										X16	63%	13.12	
								dual channel with AC_TDM	2Rank	X8	74%	15.41	
										X4	75%	15.62	
										1Rank	X16	66%	13.75
											X8	83%	17.29
随机读写	128Byte	64	67%:33%	0%	DDR4	2666	3.9us	Single channel	1Rank	X4	85%	17.7	
										X16	66%	13.75	
										X8	83%	17.29	
										X4	88%	18.33	
								dual channel with AC_TDM	2Rank	X16	83%	17.29	
										X8	85%	17.7	
										1Rank	X4	85%	17.7
											X16	85%	17.7
随机读写	64Byte	64	67%:33%	0%	DDR5	3200	3.9us	Single channel	1Rank	X16	40%	10	
										X8	40%	10	
										X4	40%	10	
								dual channel	1Rank	X16	75%	18.75	
										X8	85%	21.25	
										X4	85%	21.25	
									2Rank	X16	85%	21.25	
										X8	85%	21.25	



须知

LMBENCH中的RDWR测试带宽期望和“随机读写128Byte”项等效，2933频率效率预期比2666低1%~2%。

1.6.2 延迟目标

表1-5为row-hit下的静态延时值。

表 1-5 row-hit 下的静态延时值

模块	延时项	DDR4 2666 best case	DDR4 2666 worst case
异步桥	CMD	4.5	4.5
	DATA	0	0
DMC	CMD	8	8
	DATA	6	6
PHY	CMD	6	6
	DATA	13	13
PCB+IO	CMD	4	5
	DATA	1	1
DIMM	CMD	5	5
	DATA	0	0
DDR	tcl	19	20
	tpl	0	0
	tdqsck	0	0
总计（ck）	-	66.5	68.5
总计（ns）	-	49.875	51.375

表1-6为row-miss下的静态延时值。

表 1-6 row-miss 下的静态延时值

模块	延时项	DDR4 2666 best case	DDR4 2666 worst case
异步桥	CMD	4.5	4.5
	DATA	0	0



模块	延时项	DDR4 2666 best case	DDR4 2666 worst case
DMC	CMD	8	8
	DATA	6	6
PHY	CMD	6	6
	DATA	13	13
PCB+IO	CMD	4	5
	DATA	1	1
DIMM	CMD	5.5	5.5
	DATA	0	0
DDR	tcl	19	20
	tpl	0	0
	tdqsck	0	0
TRCD	trcd	20	20
总计（ck）	-	86.5	88.5
总计（ns）	-	64.775	66.375

1.7 调试（Debug）

1.7.1 Memory BIST 测试

DDRC支持对DDR颗粒进行扫描测试，测试完毕后会通过寄存器报告DDR颗粒是否通过测试。

1. Memory BIST支持按rank依次扫描测试，每个rank扫描前，需配置好Rank、Bank、Row、Col信息，用于生成扫描所需的地址。
2. Memory BIST支持ECC接口旁路模式，可以扫描ECC颗粒和DATA颗粒的数据，普通模式下则只能扫描器件DATA颗粒的数据。
3. Memory BIST支持off-line-MBIST扫描模式，DDRC反压正常操作的命令，只接收扫描操作命令。
4. 在启用EXMBIST模式进行扫描测试之前，需要对DMC、PHY、DDR DRAM完成初始化，否则不能进行正常工作。

EXMBIST提供了与控制器互连的接口，是一个可重用的IP，所以需要在设计的前期就考虑怎样将EXMBIST整合到设计中，除了上述功能，EXMBIST还支持以下功能：

- a. 支持高速测试频率下的扫描测试，减少了扫描测试的时间。
- b. 支持March C-、Checkerboard、MOVI、March LR、Butterfly、March-b/y/g/l/u、GALPAT测试算法。



- c. 支持用户通过寄存器配置操作指令设计算法，具体配置算法的以下几个要素：
 - 地址生成的方式：线性递增、线性递减或者随机产生。
 - 操作的类型和序列：整个扫描测试的算法只有读和写两种操作类型，通过配置读写操作的顺序可以配置不同的扫描测试算法。
- d. EXMBIST扫描测试报告存储在FIFO中，通过发送测试报告读取指令读取，测试报告内容包括：错误数据，错误数据地址，扫描到错误经过的扫描算法步骤；FIFO满了之后，将不再存储新的扫描测试报告，也不会反压扫描命令的执行。
- e. EXMBIST支持如下错误检测：
 - 固定逻辑故障SAF - Stuck-at-Fault
 - 状态转换故障TF - Transition Fault
 - 桥接逻辑故障BF - Bridging Fault
 - 耦合逻辑错误CF - Coupling Fault
 - 地址译码故障AF - Address-Decoder Fault
 - 多种动态转换错误Various dynamic faults
 - SSI/SSO on interface（PRBS pattern）
 - Pattern sensitive fault
- f. 支持通过API接口与软件交互操作：
 - 软件根据获得的错误报告信息可以分析出故障产生在哪个部位：控制器故障、memory的故障以及控制器和memory之间的接口的故障。
 - 通过和软件交互可以对EXMBIST模块本身进行自检。
 - 可以协助进行板级和ASIC级的调试。
 - 配置循环扫描测试的次数。
- g. 支持全复位和半复位之后两种复位方式：
 - 全复位：复位之后，所有的EXMBIST配置寄存器清零，停止扫描测试的执行，EXMBIST处于ready状态。
 - 半复位：复位之后，停止扫描测试的执行，EXMBIST处于ready状态。

1.7.2 统计

DDRC支持对性能、负载、命令个数以及Bank/rank读写切换次数进行统计，为性能分析及优化提供有效数据。信息统计具体的功能特性如下：

- 1. DDRC支持对如下信息的统计：
 - a. 性能统计，包含对读写命令有效传输周期的累计计数、对读写命令的latency（包含最大latency、最小latency以及积累latency）以及读写命令的流量（读写命令传输的数据位数）的统计，系统会将统计结果自动按ID号反馈给各请求端口，包含以下几种情况：



- 对所有ID的读命令的性能统计。
- 对所有ID的写命令的性能统计。
- 对指定ID的读命令的性能统计。
- 对指定ID的写命令的性能统计。
- b. 负载统计：即对当前DMC暂存的命令个数和数据个数按周期累积。
- c. 命令特性统计：对所有ID或者指定ID的命令，进行DDR读/写命令个数、active/precharge个数、Bank/rank/读写切换次数的统计。
- d. DDR带宽利用率信息的统计：对一定时间内的所有ID的读写命令个数之和以及DMC周期数进行统计。

DDR的带宽利用率的计算方法如下：

$$\frac{\text{一定时间内所有ID读命令流量统计值} + \text{一定时间内所有ID写命令流量统计值}}{(\text{memory位宽}) * (\text{一定时间内DDR的数据传输次数})}$$

2. DDRC对统计信息支持单次触发、连续触发以及基于阈值触发三种统计方式：
- a. 单次触发方式：寄存器配置开始统计，经过设定周期数后停止统计。该统计方式支持对性能、负载以及命令特性的统计。
 - b. 连续触发方式：性能统计相关计数器连续计数，寄存器配置开始统计，直到配置寄存器停止统计，统计结果溢出后清0重新计数。该统计方式支持对性能、负载以及命令特性的统计。
 - c. 基于阈值触发方式：设置统计周期以及最大最小阈值，当统计周期到达后，如果在规定的阈值范围内则不上报中断同时重新启动一次统计，直到新一次的统计超出阈值范围后停止统计，该统计方式仅支持负载统计信息的统计。

1.7.3 Debug bus 接口

通过Debug bus接口可以访问DDRC内部寄存器。

- 当总线和Debug bus接口同时访问寄存器时，优先保证寄存器总线的访问。
- 当总线访问寄存器出现异常时，可以通过Debug bus接口继续访问寄存器。

1.7.4 PHY 接口异常

DDRC支持针对PHY接口出现的异常情况响应处理，这些异常情况具体包括读数据超时和命令线奇偶校验错误，具体的情况如下。

- 读数据超时：读命令发出后，长时间（寄存器可配置）不返回读数据。
- 命令线奇偶校验错误：检测到命令线奇偶校验错误。

1.7.5 CA parity 校验测试

DDR4支持CA parity错误校验，具体的测试特性如下：

- DDRC支持对DDR4的所有地址线和部分命令线注错。
 - 错误注入的方式：一次触发单个或多个命令注错。
 - 支持对选择对任意命令注错。



- 支持在重传命令，可以过滤掉注错功能。
- 支持上报CA parity错误中断。

1.7.6 数据 CRC 测试

支持DDR4的CRC功能，具体的测试特性如下。

- DDRC支持对DDR4的写数据错误注入。
 - 错误注入的方式：可对单个或多个数据及CRC码注错。
 - 错误注入的位置可配置，且支持在多个bytelane同时注错。
 - 在重传写操作过程中，过滤掉写数据注错功能。
- 错误注入的方式：可对单个或多个数据及CRC码注错。
- 错误注入的位置可配置，且支持在多个bytelane同时注错。
- 在重传读操作过程中，过滤掉读数据注错功能。
- 在重传读操作过程中，统计在错误窗口中没有写数据错误的读操作次数。

1.7.7 内部 SRAM ECC 测试

DDRC支持对内部SRAM进行ECC数据校验。具体的测试特性如下。

- 支持对单比特和多比特两种ECC错误数据的注入。
- 支持对在数据读回之后进行注错。
- 支持通过寄存器配置错误数据注入的位置。
- 支持连续注入的方式：寄存器每配置一次注入操作，一直进行错误注入，直到寄存器配置停止。
- 支持上报单/多比特内部SRAM错误中断。
- 支持记录出错的sram地址信息。

对于长期存储数据的SRAM而言，其出错地址的回写功能是有用的，而在DDRC中，WSRAM在读取时会进行校验，若发现错误则该数据无效（除非支持forward），因此无需支持回写功能；同样地，RPRAM和SBRAM中的数据在读出后即失效，也不需要支持回写功能。此外，REC中的RAM在取出后也会失效，因此SRAM出错地址的回写功能在此类情况下也无需支持。

1.7.8 DDR 颗粒 ECC 测试

DDRC支持对DDR颗粒进行ECC数据校验。具体的测试特性如下。

- DDRC支持对单比特或多比特两种ECC错误数据的注入。
- DDRC支持对在数据读回之后进行注错。
- DDRC支持通过寄存器配置DDR颗粒的错误数据注入的位置。
- 支持单次注入和连续注入两种注入方式。
 - 单次注入：寄存器每配置一次，进行一次错误注入。
 - 连续注入：寄存器每配置一次，一直进行错误注入，直到寄存器配置操作停止或者达到设定的阈值。
- DDRC支持通过软件配置ECC数据不写入内存，这个功能用于在数据写入过程中注入ECC错误。



- DDRC支持对第一次及最后一次ECC错误信息可查询：产生单/多比特错误时的数据ID、错误数据、期望的正确数据、地址、ECC校验码。
- 支持上报DDR颗粒单/多比特错误中断。

1.7.9 RASC 校验错误注入

DDRC支持针对DDR颗粒的RASC错误注入，具体的错误注入功能如下。

- DDRC支持在数据写入之前或读回之后进行错误注入。
- 支持单次注入和连续注入两种注入方式。
 - 单次注入：寄存器每配置一次，进行一次错误注入，一次注入可注入多个地址。
 - 连续注入：寄存器每配置一次，一直进行错误注入，直到寄存器配置操作停止或者达到设定的阈值。
- DDRC支持通过寄存器配置错误注入的器件地址。
- 支持跨地址（ 2^N ）错误注入操作。
- 支持对任意颗粒的任意一位或者多位进行注错。

1.7.10 内部状态可观测

DDRC支持对DMC、PHY、MBIST中的配置寄存器和状态寄存器的数据可观测，寄存器中的数据包含中断状态信息、状态机状态信息、内部功能模块的计数器值、内部指针的值、关键的时序参数信息、内部功能模块执行结果信息。

1.7.11 中断

DDRC支持多种中断。具体的中断源如下。

- 校验错误中断
 - 内部sram出现单比特/多比特ECC错误。
 - DDR颗粒出现单比特/多比特ECC错误。
 - 操作命令/地址出现奇偶校验错误：
 - REC异常中断，发生嵌套错误的次数超过阈值或者判定为CRC错误但是错误窗口中出错的写命令和写命令个数不符。
 - REC重传操作完成。
- 统计信息中断
 - 当DMC中缓存的数据个数超过上溢值的时候触发该中断，DMC中缓存的数据个数低于下溢值的时候触发该中断。
 - 当统计DMC中缓存数据个数的统计周期完毕，触发对应统计完成中断。
 - 当DMC中缓存的命令个数超过上溢值的时候会触发中断，当缓存的命令个数低于下溢值的时候也会触发中断。
 - 当达到统计DMC中缓存命令个数的统计周期时，触发对应统计完成中断。
 - 读写命令的流量统计周期完毕时触发中断，触发对应统计完成中断。
- 刷新模式中断
 - 等待发送的AREF操作个数达到配置的AREF操作个数阈值。
 - 自刷新（非切频场景）模式下有正常访问命令。



1.8 工作方式

1.8.1 初始化

每次上电复位跟随系统一起解复位。

软复位解复位和开启时钟之后，必须读取CRG状态寄存器确认模块已经解复位和开时钟，软复位可通过寄存器SC_DDR_RESET_REQ（0xFA0A5A70）的读取状态，时钟门控可通过SC_DDRC_CLK_EN（0xFA0A5360）读取状态，然后启动DDR模块功能。

须知

DDR初始化过程中，如果针对DDR的配置（包括DDRC、DDRPHY、DDR颗粒的配置）有变更，需要按照上电初始化流程执行DDR初始化。

部分特殊场景需要动态更新的除外，比如根据温度动态更新刷新率。

1.8.2 软复位

DDR各个模块的软复位在系统CRG测进行控制，在复位过程中需确保各模块处于复位状态，随后对MCU子系统的多个小IP进行解复位，具体如下所述。

MCU（UCE）子系统有多个小IP，如I2C，I3C，UART等，默认情况下这些IP的门控是关闭的，且处于复位态；在使用这些小IP之前，需要打开APB配置时钟和工作时钟的门控，并撤离复位，才能正常运行（需要注意的是，对于RISC-V默认打开门控和解复位）；

如果一次性将所有的IP打开门控和解复位，可配置如下：

- 配置UCE_CRG_GT_EN = 0xFFF；打开所有IP时钟门控。
- 配置UCE_CRG_RST_DIS = 0x7FF，对所有IP进行解复位。

如果只使用部分IP，为省功耗，也可以针对部分IP进行控制，下面以I3C和UART为例进行说明：

- 配置UCE_CRG_GT_EN.gt_clk_i3c_en=0x1；打开的工作时钟门控。
- 配置UCE_CRG_GT_EN.gt_pclk_i3c_en=0x1；打开的APB门控。
- 配置UCE_CRG_RST_DIS.rst_i3c_req_dis = 0x1，解复位。
- 配置UCE_CRG_GT_EN.gt_clk_uart_en=0x1；打开UART的工作时钟门控。
- 配置UCE_CRG_GT_EN.gt_pclk_uart_en=0x1；打开UART的APB门控。
- 配置UCE_CRG_RST_DIS.rst_uart_req_en = 0x1，解复位UART。

1.8.3 部分配置说明

1.8.3.1 部分 IO 配置说明

- CLK无接收场景，为编码方便建议ODT与AC保持一致；RX CTLE配置无效，建议配置为0。



- FB配置规则：FB的RON、Slewrate、Boost、RX CTLE建议保持与DQS一致；ODT配置为高阻。

1.9 硬件约束

无。

1.10 软件约束

1.10.1 软件约束列表

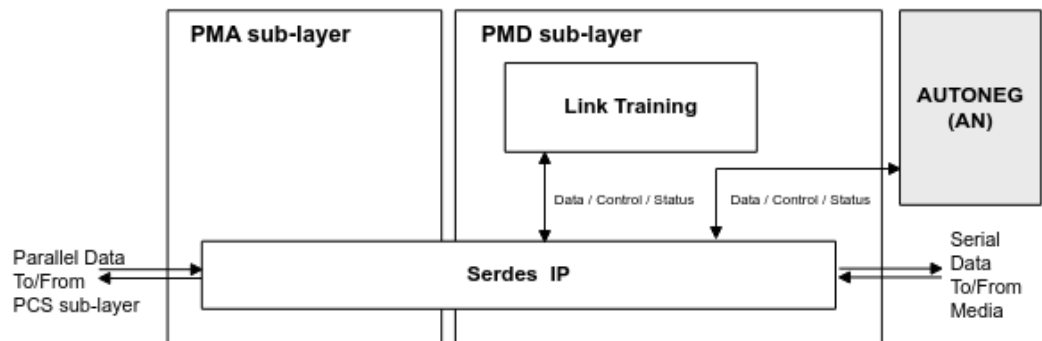
表 1-7 软件约束列表

约束序号	问题描述
1	dmc0和dmc1的clk on (DDRC_CFG_MBIST[15:14], 0x37028) 上电复位值是0，在BIOS的第一步要配置时钟打开。
2	DMC_SBR功能，不能打开icfg_sbr_sbpre_en (DDRC_CFG_PBR[9], 0x34058, 0x3C058)，固定配置为0。
3	在网络双通道场景，两个DMC位宽不一样，DMC0和DMC1分别为256-32和128-32，byte_en (DDRC_CFG_WORKMODE1, 0x34104, 0x3C104) 配置也要对应使用。
4	Hi309a晶振tcxo有3种，晶振tcxo为25/50/66.7MHz，分别进行1/2/4分频后，得到free_clk为25/25/16.67MHz，66.7MHz在4分频后为16.67MHz，在进行软件参数计算的时候要考虑。
5	DDR模式，不支持DFX的PD CNT功能。

2 Auto-negotiation (AN)

Some Ethernet protocols such as IEEE 802.3 standard and 25G & 50G Ethernet Consortium have defined their Ethernet PMA and PMD sub-layer to meet system conformance and communicate with different physical media. In a typical Ethernet implementation high speed SerDes is very common used in Ethernet PMA and PMD layer. An auto-negotiation module is defined to accomplish the local SerDes speed negotiation with the link partner. However, the SerDes speed can also be decided by software configuration directly and don't need any automatic speed negotiation process.

图 2-1 AN module with PMA and PMD sub-layer



2.1 Main Features

The main features of the AN Block are as follows:

- Support the following public Ethernet standards that have defined auto-negotiation function :
 - IEEE std 802.3tm 2015standard (See clause 73).
 - IEEE Std802.3 by-2016.
 - IEEE Std 802.3 cb draft 3.1, 2017.
 - IEEE Std802.3 bs tm-2017.
 - IEEE std 802.3cd draft 3.0(See clause 73).



- 25G & 50G Ethernet Consortium standard (See Chapter 3.2.5.1 Auto-negotiation).
- Support some private protocol that have defined auto-negotiation function:
Support Broadcom private HigiG and HigiG2 protocol.
- Support AN page transmission and reception.
- Support AN next page transmission and reception.
AN next page content can be configurable so that it can support different protocols.
- Support multi-lane PHY auto-negotiation.
Page transmission and reception is done on SerDes lane 0 of multi-lane PHY.
- Support SerDes oversampling rate at 1.25Gbps, 3.125Gbps, 6.25Gbps, 10.3125Gbps, 10.9375Gbps, 25.78125G and 26.5625G.
- Support SerDes bus width at 10bits, 16bits, 20bits, 32bits, 40bits and 64bits.
10bits works at 1.25Gbps, 3.125G and 6.25G SerDes speed.
16bits works at 10.3125G and 10.9375G SerDes speed.
20bits works at 6.25G, 10.3125G and 10.9375G SerDes speed.
32bits and 40bits works at 10.3125G, 10.9375G, 25.78125G and 26.5625G SerDes speed.
64bits works at 25.78125G and 26.56125G SerDes speed.
- Support AN parallel detection function for XAUI (10GBASE-KX4), 1GBASE-KX or 2.5GBASE-KX.
Valid XAUI code groups on Lane0~Lane3 in the absence of AN pages when XAUI operation is selected (legacy support).
- AN timeout counter is Programmable.
AN Timeout counter can has 1ms granularity.
AN Timeout counter can be defined up to 32 seconds.

2.2 Terminology

The following terms have specific meanings in this section of the document.

表 2-1 AN Terminology

Term	Description
AHB	Advanced High Performance Bus
AN	Auto-Negotiation
BAM	Broadcom Auto-Negotiation Mode
CPU	Center Processing Unit
DME	Differential Manchester Encoding
HCD	highest common denominator



Term	Description
IEEE	Institute of Electrical and Electronics Engineers
LT	Link Training
MAC	Media Access Control
PCS	Physical Coding Sublayer
PMD	Physical Medium Dependent
PRBS	Pseudo-Random Binary Sequence
MSA	Ethernet Consortium
-	-

2.3 References

The following documents provide useful information related to this section of the document.

- IEEE std 802.3tm 2015 (Revision of IEEE Std 802.3-2012), Sept 3 2015
- IEEE Std 802.3 cd draft3.0, Nov 13 2017.
- IEEE Std 802.3 cb draft 3.1, Aug 18 2017.
- IEEE Std802.3 by-2016.
- IEEE Std802.3 bstm-2017.
- MLD HiGig2 Design Specification Using TDM SerDes V2p1, Broadcom.
- 25G & 50G Specification, Aug 18, 2015.

2.4 Auto-Negotiation (AN) Overview

2.4.1 Introduction

The objective of the Auto-Negotiation function is to provide the means to exchange information between two devices that share a link across a backplane or copper cable channel. The function can also automatically configure both devices to take maximum advantage of their various abilities. It has the additional objective of supporting a digital signal detect to ensure that the device is attached to a link partner rather than detecting signal due to crosstalk.

This function allows the devices at both ends of a link segment to advertise abilities, acknowledge receipt and discover the common modes of operation that both devices can support, and to reject the use of operational modes that are not shared by both devices.

There are 4 possible conditions:

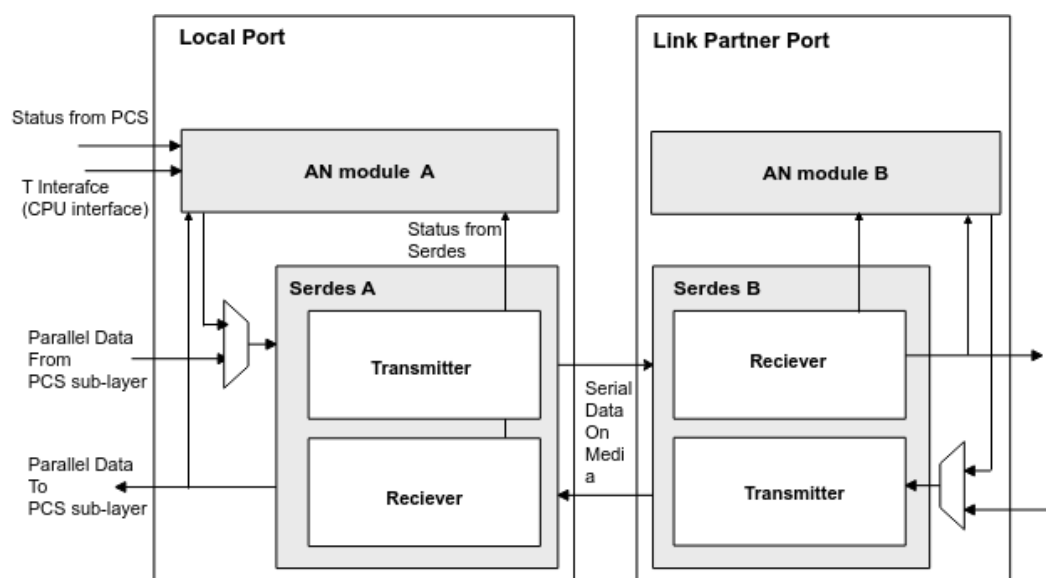
- Only one common mode is found between the two devices that both support and enable Auto-negotiation. Then the only one available mode is selected.
- More than one common mode exists between the two devices that both support and enable Auto-negotiation. A mechanism is provided to allow the devices to resolve to a single mode of operation using a predetermined priority resolution function.
- A device that enabled Auto-Negotiation function and a link partner that disabled Auto-Negotiation function can't use normal auto-negotiation process. A parallel detection function is provided to detect the link partner. This condition is only applicable for low speed Ethernet port 1000BASE-KX, 2.5GBASE-KX or 10GBASE-KX4(XAUI) speed auto-negotiation.
- A device that enabled Auto-Negotiation function and the link partner is a legacy device that does not support auto-negotiation protocol at all. The parallel detection function can also support to detect the link partner. This condition is only applicable for low speed Ethernet port 1000BASE-KX, 2.5GBASE-KX or 10GBASE-KX4(XAUI) speed auto-negotiation.

The Auto-Negotiation function is used for all kinds of Ethernet device PHY, including 1000BASE-KX, 2.5GBASE-KX, 5GBASE-KR, 10GBASE-KX4(XAUI), 10GBASE-KR, 25GBASE-KR-S, 25GBASE-KR, 25GBASE-CR-S, 25GBASE-CR, 40GBASE-KR4, 40GBASE-CR4, 50GBASE-KR2, 50GBASE-CR2, 50GBASE-KR, 50GBASE-CR, 100GBASE-CR10, 100GBASE-CR4, 100GBASE-KP4, 100GBASE-KR4, 100GBASE-KR2, 100GBASE-CR2, 200GBASE-KR4 and 200GBASE-CR4.

Notice : 100GBASE-KP4 AN function is special because whether the chip can support 100GBASE-KP4 also depends on whether SerDes support PAM4 and LT (Link Training) function can support KP4 training function. For HiLink H30 or H32 SerDes, PAM is not supported.

Auto-negotiation (AN) module mainly implements with local complete the negotiation function SerDes IP (mainly See IEEE 802.3 2015 clause 73). Please see Auto-negotiation diagram in 图2-2:

图 2-2 Auto-Negotiation Diagram





In this diagram we only use a local SerDes lane and a link partner SerDes lane as an example to explain how AN function work. Every SerDes lane has a receiver part and a transmitter part. In fact the auto-negotiation operation could be started independently if every SerDes lane can work as an individual Ethernet MAC port.

Auto-Negotiation is performed using differential Manchester encoding (DME) pages. DME provides a DC-balanced signal. DME page does not add packet or upper layer overhead to the network devices but contains the information for negotiation. For more information, please refer to “ [DME page encoding](#) ”.

AN function consists of 3 main operations:

- Transmit operation. This operation constructs link code page as per DME, and transmit link code pages to PMA.
- Receive operation. Receive operation mainly receive data from PMA, lock link code pages boundary, then parse page information and send them to arbitration function.
- Arbitration operation. Arbitration operation use priority resolution function to determine the highest common denominator (HCD). Then the appropriate PHY is enabled to work by software or hardware. If next page function is supported by the link partner, arbitration operation will support next page information arbitration.

The AN module also provides a special parallel detection function to allow Ethernet devices to interoperate with another Ethernet device that either have no Auto-Negotiation or Auto-Negotiation function disabled. Once PCS detect special AN codeword and reach link up status , a link up signal will be sent to notify AN module . There are 2 conditions :

- When link partner's auto-negotiation function is disabled or not supported and local device's auto-negotiation function is enabled, AN arbitration FSM will detect link up signal from PCS at the same time during normal negotiation. If one only link up signal from PCS is detected, AN arbitration process will return AN good result directly. Otherwise it will follow normal negotiation flow.
- When link partner's auto-negotiation function is disabled or not supported and local device's auto-negotiation function is disabled, the parallel SerDes data bus will be connected to PCS directly.

The Auto-Negotiation function can also support multi-lane application. For the PHY port that has multiple SerDes lanes, for instance, 100GBASE-CR10 with 10 SerDes lanes, SerDes lane 0 will be selected for Auto-Negotiation with other PHY.

说明

HiGig2 Auto-Negotiation leverages IEEE Clause73 protocol but uses its own Broadcom Auto-Negotiation Mode (BAM) to convey between link partners regarding its speed and ability. The 25G/50G consortium uses the same mechanisms as BAM to convey abilities. The differences between IEEE802.3, Higig2 and 25G/50G consortium page information format will be described in later chapter.

2.4.2 Connectivity

The AN module communicates with other blocks several different ways:

- AN will share SerDes parallel transmit and receive data bus with PCS layer and LT module. When auto-negotiation is enabled, AN will control the transmit data bus completely. AN control priority will be higher than LT module.
- AN will provide T interface for CPU accessing. T interface can be integrated by higher CPU accessing circuits so that CPU software can control AN module.
- Some status signals from SerDes should be connected to AN module. For example, signal detect signal from SerDes receive side should be connected to AN module directly.
- Some status signals from PCS layer should be connected to AN module. For example, PCS link up status signal should be connected to AN module.

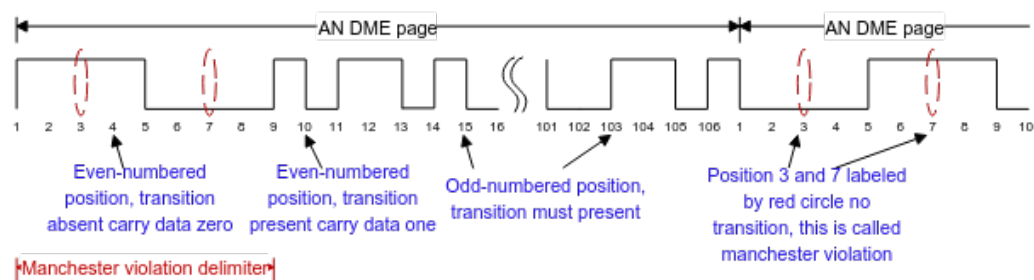
2.5 AN Data Structure in IEEE 802.3

2.5.1 DME page encoding

Auto-Negotiation's method of communication builds upon the encoding mechanism known as differential Manchester encoding (DME). In DME rule, all odd-numbered position represents clock information, so each odd-numbered position shall contain a transition. For even-numbered position, it carry data information, a transition present represents a data one, a transition absent represents a data zero.

A DME page for AN consists of 106 evenly spaced transition positions, shown as follows.

图 2-3 AN DME structure



The first eight positions construct Manchester violation delimiter, which marks the beginning of each DME page and allows receiver to obtain the page synchronization. The Manchester violation delimiter carry 4-bit all zero data as per DME rule, but position 3 and 7 no transition (violation). That results Manchester violation delimiter contains a transition at position 1 and position 5 and no transitions at the remaining positions. The Manchester violation delimiter is the only place where four intervals occur between transitions.

Position 9~104 carry 48-bit data, which define AN page information. Position 105~106 carry 1-bit pseudo-random data. There is not Manchester violation for all odd-numbered positions from 9 to 106 inclusively.



The pseudo-random data bit shall be derived from a pseudo-random generator, as follows:

$$x^7 + x^3 + 1 \text{ or } x^7 + x^6 + 1$$

The circuits diagram is shown here, initial SEED is fixed to 0x40. The S6 value will be output to be taken as the generated pseudo-random data bit. The shift register chains should be clocked only once per DME page.

图 2-4 pseudo-random circuit diagram



The purpose of pseudo-random data bit to remove the spectral peaks that would otherwise occur when sending the same AN page repeatedly. Randomly choosing between 0 or 1 for one of the DME bits results in randomly inverting or not inverting the encoded page so that repetitions of the same page no longer produce a periodic signal.

The AN DME pages are transmitted continuously without any idle or gap. The minimum number of transitions present in a page is 51, and the maximum number is 100.

2.5.2 DME Base page

The base link codeword (Base Page) is one of AN DME page. The 48bit AN information data conveyed in base page is defined as follows:

图 2-5 Base page AN information structure

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
S0	S1	S2	S3	S4	E0	E1	E2	E3	E4	C0	C1	C2	RF	Ack	NP

D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31
T0	T1	T2	T3	T4	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10

D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42	D43	D44	D45	D46	D47
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	F2	F3	F0	F1

The 48-bit data is labeled as D0, D1, D2... till D47 in order. D0 is firstly transmitted. These fields are :

D[4:0] contains the Selector Field S[4:0], S[0] is refers to D0.

D[9:5] contains the Echoed Nonce field.

D[12:10] contains capability bits to advertise capabilities not related to the PHY.

D[15:13] contains the RF, Ack, and NP bits.



- D[20:16] contains the Transmitted Nonce field.
- D[43:21] contains the Technology Ability Field.
- D[47:44] contains FEC capability.

2.5.2.1 Selector Field

The Selector Field, S[4:0] in the link codeword, shall be used to identify the type of message being sent by Auto-Negotiation. The following table identifies the types of messages that may be sent. The Selector Field uses a 5-bit binary encoding, which allows 32 messages to be defined. All unspecified combinations are reserved. Reserved combinations shall not be transmitted.

表 2-2 Selector Field Encoding

S4	S3	S2	S1	S0	Description
0	0	0	0	1	IEEE Std 802.3
0	0	0	1	0	IEEE Std 802.9a-1995 (withdrawn)
0	0	0	1	1	IEEE Std 802.5v-2001 (withdrawn)
0	0	1	0	0	IEEE Std 1394
0	0	1	0	1	INCITS
Other combinations are reserved.					

For AN module, Selector Field should be fixed to S[4:0]=0x1.It is configurable and will be checked during auto-negotiation.

2.5.2.2 Transmitted Nonce Field

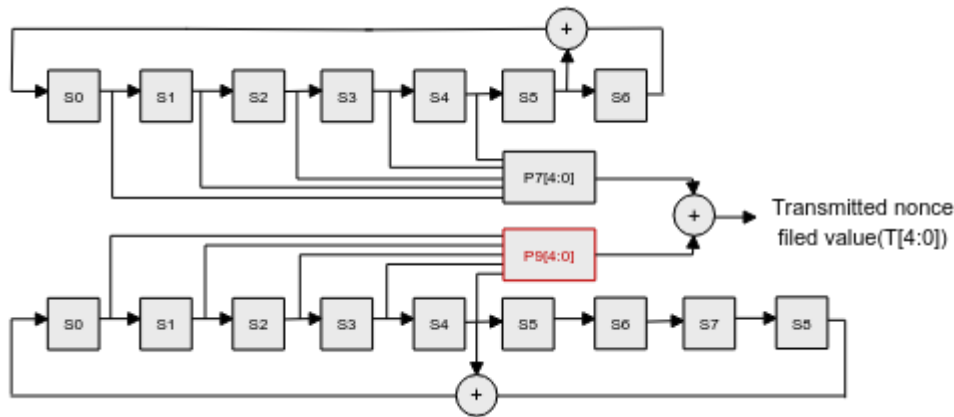
Transmitted Nonce Field (T[4:0]) is a 5-bit wide field containing a random or pseudo-random number. Transmitted Nonce Field is used to distinguish ability information exchange of each time.

The transmitted nonce should have a uniform distribution in the range from 0 to 31. The transmitted nonce could be generated as follows circuit diagram. Two pseudo-random generators are introduced, $x^7 + x^6 + 1$ and $x^9 + x^5 + 1$. Seed is fixed to 0x40 and 0x100 respectively.

It is suggested that these two generators work at different clock domains, and just take a little more asynchronous process. The circuits could be clocked to shift at each cycle, even no new nonce value be requested by transmit operation.



图 2-6 transmitted nonce circuit diagram



2.5.2.3 Echoed Nonce Field

Echoed Nonce Field (E[4:0]) is a 5-bit wide field that is used to convey Transmitted Nonce Field value from the link partner back to link partner. So it is just a copy of Transmitted Nonce Field of received Base Page.

2.5.2.4 Technology Ability Field

Technology Ability Field (A[22:0]) is a 23-bit wide field containing information indicating supported technologies specific to the selector field value when used with the Auto-Negotiation for Backplane or Copper Ethernet. These bits are mapped to individual technologies such that abilities are advertised in parallel for a single selector field value.

The Technology Ability Field encoding for the selector field taking IEEE 802.3, with Auto-Negotiation for Backplane Ethernet is described below. The fields A[22:16] are reserved for future use. Reserved fields shall be sent as zero and ignored on receive.

表 2-3 Technology Ability Field encoding

Bit	Technology
A0	1000BASE-KX
A1	10GBASE-KX4
A2	10GBASE-KR
A3	40GBASE-KR4
A4	40GBASE-CR4
A5	100GBASE-CR10
A6	100GBASE-KP4
A7	100GBASE-KR4



Bit	Technology
A8	100GBASE-CR4
A9	25GBASE-KR-S or 25GBASE-CR-S
A10	25GBASE-KR or 25GBASE-CR
A11	2.5GBASE-KX
A12	5GBASE-KR
A13	50GBASE-KR or 50GBASE-CR
A14	100GBASE-KR2 or 100GBASE-CR2
A15	200GBASE-KR4 or 200GBASE-CR4
A16 through A22	Reserved for future technology

A PHY device must support the data service ability for a technology it advertises. Multiple technologies may be advertised in the link codeword simultaneously.

For 25 Gbps operation the same bits are used to advertise backplane and copper cable assembly operation. For other speeds, a PHY for operation over an electrical backplane (e.g., 100GBASE-KX, 10GBASE-KX4, 10GBASE-KR, 40GBASE-KR4, 100GBASE-KP4, 100GBASE-KR4) shall not be advertised simultaneously with a PHY for operation over a copper cable assembly (e.g., 40GBASE-CR4, 100GBASE-CR10, 100GBASE-CR4), as their physical interfaces are different. For instance, once a device declares 40GBASE-KR4 ability, it should not declare 100GBASE-CR10 ability.

25GBASE-KR-S abilities are a subset of 25GBASE-KR abilities, and likewise 25GBASE-CR-S abilities are a subset of 25GBASE-CR abilities. To allow interoperability between 25GBASE-KR-S and 25GBASE-KR PHY types, and between 25GBASE-CR-S and 25GBASE-CR PHY types, a device that supports 25GBASE-KR or 25GBASE-CR should advertise both A9 and A10 ability bits simultaneously during auto-negotiation.

2.5.2.5 FEC Capability Field

FEC (F2:F3:F0:F1) is encoded in bits D44:D47 of the base link codeword. The four FEC bits are used as follows:

1. F0 is 10 Gb/s per lane FEC ability
2. F1 is 10 Gb/s per lane FEC requested
3. F2 is 25G RS-FEC requested
4. F3 is 25G BASE-R FEC requested

Bits F2 and F3 are used for resolving FEC operation for 25G PHYs while bits F0 and F1 are used for 10 Gb/s per lane operation. Bits F0 and F1 are not used for 25G PHYs.



2.5.2.6 Pause Ability Field

Pause (C0:C1) is encoded in bits D11:D10 of the base link codeword. The two-bit Pause is encoded as follows:

- a) C0 is PAUSE bit.
- b) C1 is ASM_DIR bit.

When PAUSE=0 and ASM_DIR = 0, the capability is "No PAUSE".

When PAUSE=0 and ASM_DIR = 1, the capability is "Asymmetric PAUSE toward link partner".

When PAUSE=1 and ASM_DIR=0, the capability is "Symmetric PAUSE".

When PAUSE = 1 and ASM_DIR = 1, the capability is "Support Asymmetric PAUSE and Symmetric PAUSE toward local device".

2.5.2.7 Remote Fault Field

Remote Fault (RF) is encoded in bit D13 of the base link codeword. The Remote Fault bit provides a standard transport mechanism for the transmission of simple fault information. When constructing base page in transmit, RF field value is directly filled with configuration register AN_BP_ABILITY0_CFG.REMOTE_FAULT.

2.5.2.8 Acknowledge Field

The Acknowledge (Ack) Bit is encoded in bit D14 of link codeword. Acknowledge is used by the Auto-Negotiation function to indicate that a device has successfully received its link partner's link codeword.

The Acknowledge in transmitted link codeword will be set to logical one after the reception of at least three consecutive and consistent DME pages (ignoring the Acknowledge bit value) from link partner.

2.5.2.9 Next Page Field

Next Page (NP) is encoded in bit D15 of link codeword to indicate whether second kind of DME page, Next Page, will be sent. If a device wishes to engage in DME Next Page exchange, it shall set the NP bit to logical one. If the device does not have any DME Next Pages to send, the NP bit shall be set to logical zero, no matter link partner has Next Page to send or not.

2.5.3 DME Next page

The DME Next Page allows the Auto-Negotiation function to exchange more information between local PHY device and link partner.

The DME Next Page is a second kind of AN DME page. The 48bit AN information data conveyed in Next Page is defined as follows, there are two kinds of Next Page, Message Next Page and Unformatted Next Page:



图 2-7 Message Next Page structure

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	T	Ack2	MP	Ack	NP

D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31
U0	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15

D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42	D43	D44	D45	D46	D47
U16	U17	U18	U19	U20	U21	U22	U23	U24	U25	U26	U27	U28	U29	U30	U31

图 2-8 Unformatted Next Page structure

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
U0	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	T	Ack2	MP	Ack	NP

D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31
U11	U12	U13	U14	U15	U16	U17	U18	U19	U20	U21	U22	U23	U24	U25	U26

D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42	D43	D44	D45	D46	D47
U27	U28	U29	U30	U31	U32	U33	U34	U35	U36	U37	U38	U39	U40	U41	U42

The 48-bit data is labeled as D0, D1, D2... till D47 in order. D0 is firstly transmitted.

D[15:11] contains NP, Ack,MP,Ack2,T bits. Both message next page and unformatted next page have the same definition on D[15:11].

For message next page, D[10:0] is message code field M [10:0], D[47:16] is 32-bit unformatted code field U[31:0].

For unformatted next page, D[10:0] and D[47:16] construct 43-bit unformatted code field U[42:0].Unformatted Next Pages can't exist independently. Each series of Unformatted Next Pages (the page number of different content page may be any) is always preceded by a Message Next Page containing the Message code field that defines how the following Unformatted Next Pages shall be interpreted.

2.5.3.1 Message Code Field

Message code field (M[10:0]) only exists in message next page. It is an eleven bit field, encoding 2048 possible message. The following table identifies the types of messages that may be sent.

All unspecified combinations are reserved. Reserved combinations shall not be transmitted.



表 2-4 Message Code Field Encoding

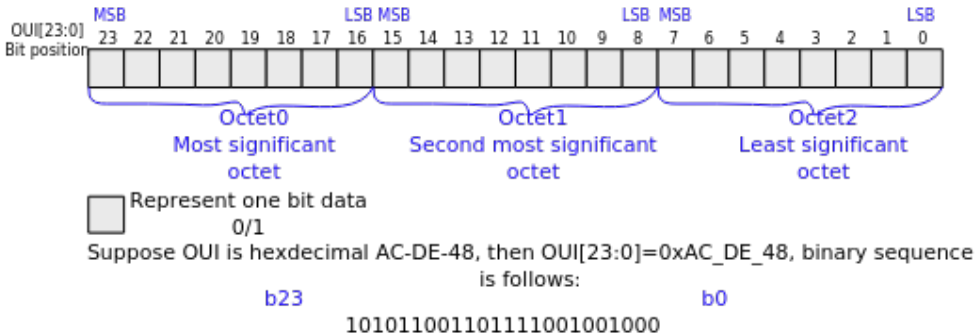
M 10	M 9	M 8	M 7	M 6	M 5	M 4	M 3	M 2	M 1	M 0	Description
0	0	0	0	0	0	0	0	0	0	1	Null Message
0	0	0	0	0	0	0	0	1	0	1	Organizationally Unique Identifier Tagged Message
0	0	0	0	0	0	0	0	1	1	0	AN device Identifier Tag Code
0	0	0	0	0	0	0	1	0	1	0	EEE Technology Message Code. EEE capability is advertised using unformatted message code field in the Message Next Page.
Other combinations are reserved.											

- Null Message
The Null Message is consisted of only Message Next Page, and set message code field to 0x1, set unformatted code field to all zero.
The Null Message shall be transmitted by a device during Next Page exchange when the Device has no further messages to transmit and its Link Partner is still transmitting valid Next Pages.
- Organizationally Unique Identifier (OUI) tag code message
The OUI tag code message is consisted of a Message Next Page with the message code field 0x5 followed by one Unformatted Next Page. The combination of Message Next Page and Unformatted Next Page is used to exchange 24-bit OUI⁸⁹ and 20-bit user-defined code information.

说明

89:OUI is allocated by IEEE, it is 24-bit wide value.

图 2-9 24-bit OUI structure



The unformatted code field of Message Next Page shall contain the most significant 11 bits of the OUI (bits 23:13) in D26:D16 (bits U10 to U0) with the

most significant OUI bit in D26 (bit U10) of the unformatted code field, the next 11 most significant bits of the OUI (bits 12:2) in D42:D32 (bit U26 to U16) with the most significant bit in D42 (bit U26).

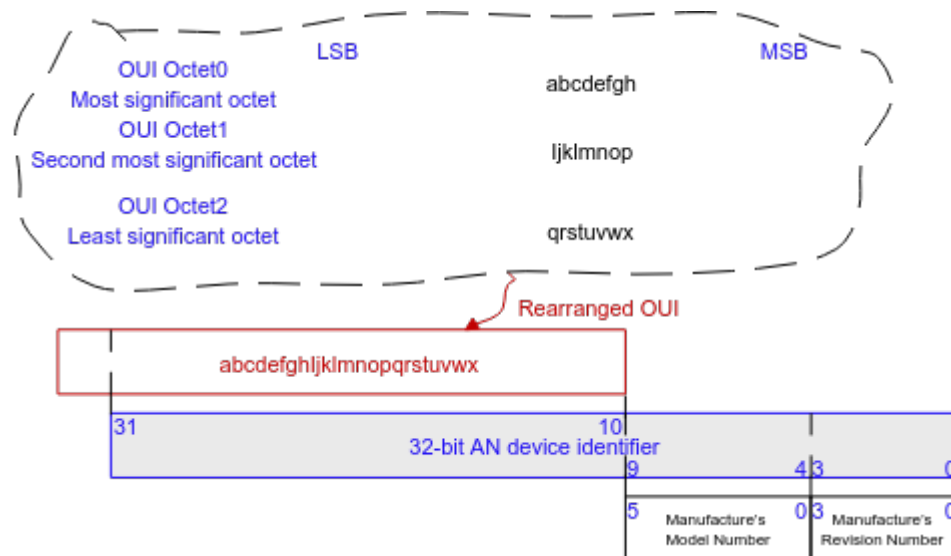
The unformatted code field of the Unformatted Next Page shall contain the remaining least significant 2 bits of the OUI (bits 1:0) in D10:D9 (U10 and U9) with OUI bit 1 in D10 (bit U10), with the D8:D0, D26:D16 (bits U8 to U0, U21 to U11) as a user-defined user code value that is specific to the OUI transmitted.

The remaining unformatted code field bits in the Message Next Page and the Unformatted Next Page shall be sent as zero and ignored on receipt.

- AN device identifier tag code

The AN device ID tag code message is used to exchange AN device identifier information. The AN device identifier is 32-bit unique identifier for an Auto-Negotiation, and be defined as follows:

图 2-10 32-bit AN Device identifier structure



The AN Device identifier is composed of 22-bit content taken from Organizationally Unique Identifier (OUI) assigned to the PHY manufacturer by the IEEE, plus a six-bit manufacturer's model number, plus a four-bit manufacturer's revision number. The figure details the process how to take OUI content to construct the most significant 22-bit data (bit31:bit10) of AN Device identifier. For instance, giving OUI AC-DE-48, then the most significant 22-bit data (bit31:bit10) of AN Device identifier is shown below:

图 2-11 The Most Significant 22-bit Data (bit31:bit10) of AN Device Identifier

b31 b10
1101010111101100010010

The AN device ID tag code message is consisted of a Message Next Page with the message code field 0x6 followed by one Unformatted Next Page.



The unformatted code field of Message Next Page shall contain the most significant 11 bits of the AN device identifier in D26:D16 (bits U10 to U0) with the most significant AN device identifier bit in D26 (bit U10) of the unformatted code field, and the next 11 most significant bits of the AN device identifier in D42:D32 (bit U26 to U16) with the most significant bit in D42 (bit U26) of the unformatted code field.

The unformatted code field of the Unformatted Next Page shall contain the remaining least significant 10 bits of the AN device identifier in bits D10:D1 (bit U10 to U1) with the most significant bit in D10 (bit U10), with D0, D26:D16 (bits U0, U21 to U11) as a user-defined user code value that is specific to the OUI transmitted.

The remaining unformatted code field bits in the Message Next Page and the Unformatted Next Page shall be sent as zero and ignored on receipt.

- **EEE technology message code**
The EEE technology code message is consisted of only a Message Next Page. The message code field set to 0xA, the contents of the unformatted code field bits U31:U0 (D47:D16) shall be defined as follows:

表 2-5 Message Code Field Encoding

Bit position of the unformatted code field	Usage Description
U14(D30)	Whether to advertise the that 25GBASE-R PHY has EEE deep sleep capability. 1=yes, 0=no ⁹⁰
U13(D29)	Whether to advertise that the 100GBASE-CR4 PHY has EEE deep sleep capability. 1=yes, 0=no
U12(D28)	Whether to advertise that the 100GBASE-KR4 PHY has EEE deep sleep capability. 1=yes, 0=no
U11(D27)	Whether to advertise that the 100GBASE-KP4 PHY has EEE deep sleep capability. 1=yes, 0=no
U10(D26)	Whether to advertise that the 100GBASE-CR10 PHY has EEE deep sleep capability. 1=yes, 0=no
U9(D25)	Reserved with 0
U8(D24)	Whether to advertise that the 40GBASE-CR4 PHY has EEE deep sleep capability. 1=yes, 0=no



Bit position of the unformatted code field	Usage Description
U7(D23)	Whether to advertise that the 40GBASE-KR4 PHY has EEE deep sleep capability. 1=yes, 0=no
U6(D22)	Whether to advertise that the 10GBASE-KR PHY has EEE capability. 1=yes, 0=no
U5(D21)	Whether to advertise that the 10GBASE-KX4 PHY has EEE capability. 1=yes, 0=no
U4(D20)	Whether to advertise that the 1000BASE-KX PHY has EEE capability. 1=yes, 0=no
U0~U3, U15~U31	Reserved with 0

说明

90:If set to no, either doesn't have this function, or this function is disabled.

2.5.3.2 Unformatted Code Field

The Unformatted Code Field(U0~U31 or U0~U42) is really used to carry device information in Next page. Unformatted Code Field is interpreted by preceded Message Code Field.

2.5.3.3 Message Page Field

Message Page (MP) Field is used to differentiate a Message Next Page from an Unformatted Next Page. MP shall be set as follows:

- logic zero = Unformatted Next Page.
- logic one = Message Next Page.

2.5.3.4 Acknowledge 2 Field

Acknowledge 2 (Ack2) is used to indicate that a device has the ability to comply with the message of Next Page received from link partner. Ack2 shall be set as follows:

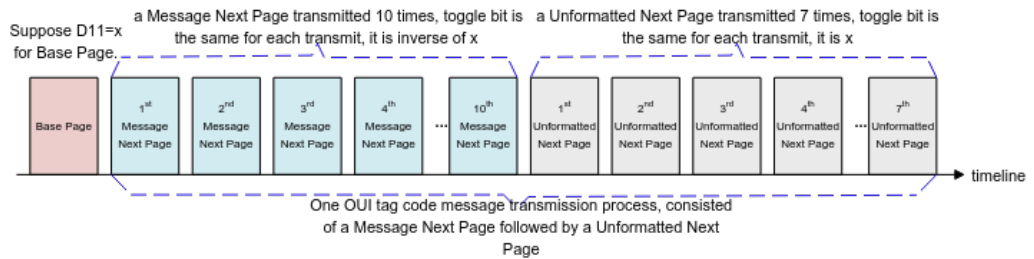
- logic zero = cannot comply with message.
- logic one = will comply with message.

The 'comply with' means that a device could act the information (or perform the task) defined in the Next Page.

2.5.3.5 Toggle Field

Toggle (T) is used to ensure synchronization with the Link Partner during Next Page exchange. This bit shall always take the opposite value of the Toggle bit in the previously exchanged Next Page. The value of the Toggle bit in the first Next Page transmitted is the inverse of D11 in the Base Page.

图 2-12 Toggle bit usage example



2.5.3.6 Acknowledge Field

The Acknowledge Bit is encoded in bit D14 of link codeword. Acknowledge is used by the Auto-Negotiation function to indicate that a device has successfully received its link partner's link codeword. The Acknowledge in transmitted link codeword will be set to logical one after the reception of at least three consecutive and consistent DME pages (ignoring the Acknowledge bit value) from link partner.

2.5.3.7 Next Page Field

Next Page (NP) is encoded in bit D15 of link codeword to indicate whether second kind of DME page, Next Page, will be send. If a device wishes to engage in DME Next Page exchange, it shall set the NP bit to logical one. If the device does not have any DME Next Pages to send, the NP bit shall be set to logical zero, no matter link partner has Next Page to send or not.

2.6 AN Data Structure for BAM and MSA

For special AN function mode , the Auto-negotiation base pages are exchanged between the two ends of the backplane or copper cable channel, with the exchange taking place on physical lane 0. After the exchange of the base page, the link partners exchange an OUI tagged formatted Next Page. IEEE Message Page is used to indicate the OUI and BAM mode. Specific OUIs are transmitted to define whether BAM (Broadcom), MSA (Consortium), or some other format information is being provided.

For 25G & 50G Ethernet Consortium, the link priority is in the following table.

25G & 50G Ethernet Consortium Link Priority List



表 2-6 Link priority for 25G & 50G Ethernet Consortium

Priority	Technology	Capability	Note
1	50GBASE-R2	50Gb/s, 2 lanes	Consortium mode
2	25GBASE-R1	25Gb/s, 1 lane	Consortium mode

2.6.1 AN Data Structure for BAM

In Broadcom auto-negotiation mode (BAM), the MP-5 carries the Broadcom OUI as listed in the following tables. The UP-1 page contains a 9-bit BAM Code and the BAM abilities. BAM requires a BAM code value of 3, which indicates that extended abilities are specified directly in the BAM abilities fields. The BAM abilities fields are mapped into bits 16-26 and 32-44 of the UP1.

表 2-7 BAM Page Format

Page	D0-D8	D9-D10	D11	D12	D13	D14	D15	D16-D26	D27-D31	D32-D44	D45-D47
MP-5	5		T	ACK2	MP	ACK	NP	OUI[13:23]	Reserved	OUI[2:12]	reserved
UP-1	BC[0:8]	OUI[0:1]	T	ACK2	MP	ACK	NP	BA[0:10]	Reserved	BA[11:23]	reserved

表 2-8 BAM MP-5 Format

Field Name	Message Page Position
MESSAGE_OUI (set to 11'd5 for MP-5)	10:0
TOGGLE	11
ACK2	12
PAGE_TYPE (Set to 0b1 for MP-5)	13
ACK	14
Next Page indication (set to 0b1 for MP-5)	15
OUI[13:23] (BROADCASTCOM_OUI in case of BAM, CONSORTIUM_OUI in case of Consortium)	[26:16]
Reserved	[31:27]



Field Name	Message Page Position
OUI[12:2] (BROADCOM_OUI in case of BAM, CONSORTIUM_OUI in case of Consortium)	[42:32]
Reserved	[47:43]

表 2-9 BAM UP-1 Format

Field Name	UNFORMATTED NEX PAGE (UP Bits, aka U47:0)
CL73_BAM_CODE[8:0]	8:0
OUI_BROADCOM[1:0]	10:9
TOGGLE	11
ACK2	12
PAGE_TYPE(0b0 for UP1)	13
ACK	14
Next Page indication (set to 0b0 for UP1)	15
SPEED_20G_KR2	16
SPEED_20G_CR2	17
SPEED_20G_KR1	18
SPEED_20G_CR1	19
SPEED_25G_KR1	20
SPEED_25G_CR1	21
SPEED_40G_KR2	22
SPEED_40G_CR2	23
SPEED_50G_KR2	24
SPEED_50G_CR2	25
Reserved	31:26
SPEED_50G_KR4	32
SPEED_50G_CR4	33
Reserved	39:34
CL91_EN (for 25G, bits 20 and 21, and 50G, bits 24 and 25)	40



Field Name	UNFORMATTED NEX PAGE (UP Bits, aka U47:0)
CL74_EN (for 25G, bits 20 and 21, and 50G, bits 24 and 25)	41
HG2 Enable	42
CL91_REQ_EN	43
CL74_REQ_EN	44
Reserved	47:45

2.6.2 AN Data Structure for MSA

The Organization Unique Identifier (OUI) tagged formatted Next Page, with the OUI set to the 25G Ethernet Consortium CID of 0x6A737D(h) and a message code of 5 is shown in the figure Below:

图 2-13 Auto-negotiation Next Page – OUI extended

Next Page	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31
	D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42	D43	D44	D45	D46	D47
MP5 (D0 - D15)	1	0	1	0	0	0	0	0	0	0	0	0	T	0	1	ACK
(D16 - D31)	1	1	0	0	1	0	1	0	1	1	0					Reserved
(D32 - D47)	1	1	1	1	1	0	1	1	0	0	1					Reserved

message code #5

OUI [23:2]

The OUI tagged unformatted Next Page, with the OUI set to the 25G Ethernet Consortium CID of 0x6A737D(h), a code for extended technology abilities, and the appropriate code bits for 25 GE CR1/KR1 and 50 GE CR2/KR2 operation, as well as four FEC control bits, is shown in the figure below:

图 2-14 Auto-negotiation OUI Extended Next Page Codes

Next Page	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31
	D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42	D43	D44	D45	D46	D47
UP-1 (D0 - D15)	1	1	0	0	0	0	0	0	0	1	0	T	0	0	ACK	NP
(D16 - D31)	0	0	0	0	1	1	0	0	1	1	0					Reserved (set to 0)
(D32 - D47)										F1	F2	F3	F4			

Code for extended technology abilities

OUI [1:0]

25GBASE-KR1

50GBASE-KR2

25GBASE-CR1

FEC Control

50GBASE-CR2

2.7 AN Operations

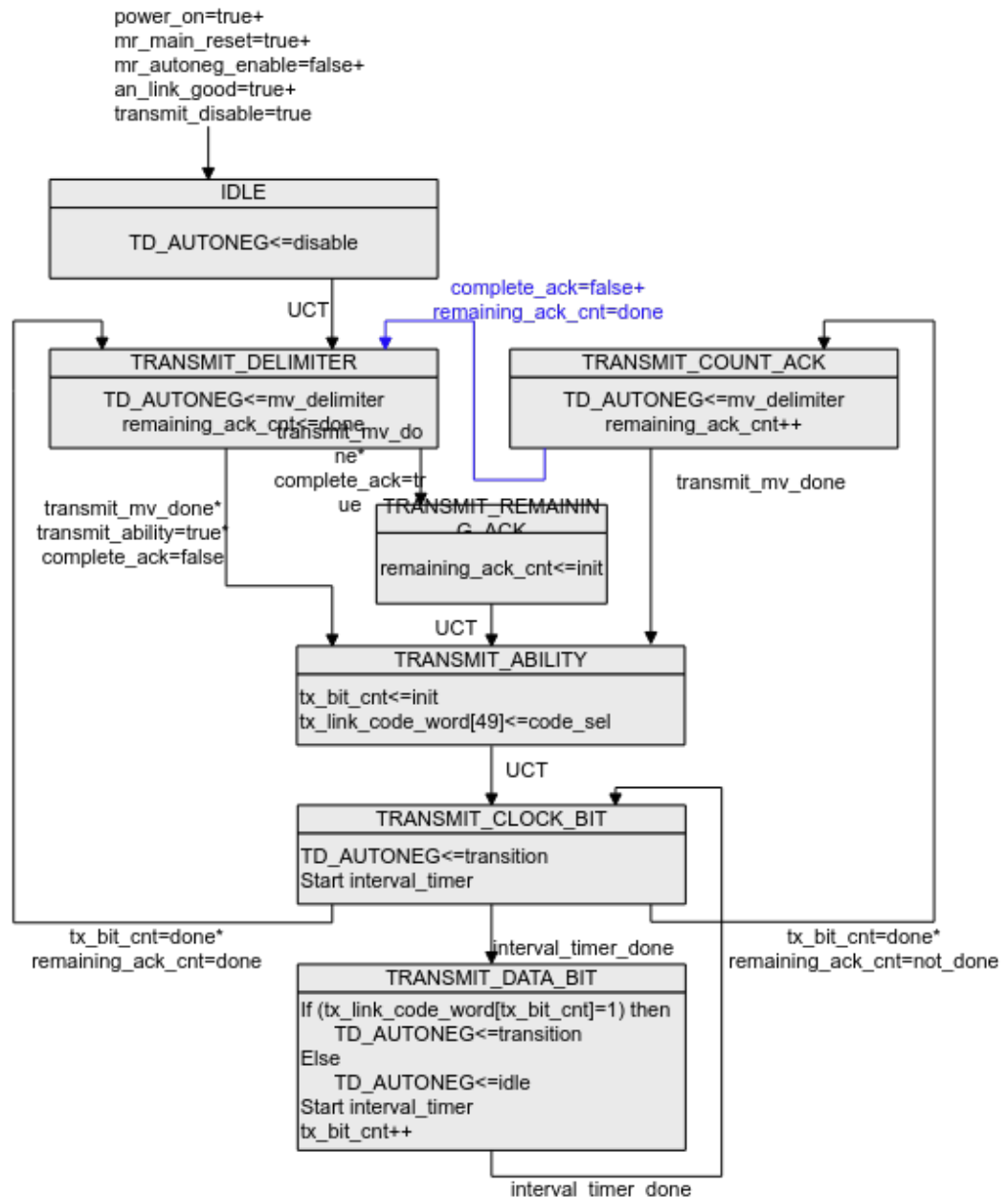


2.7.1 AN Transmit Operations

The transmit function provides the ability to transmit DME pages. There are two types of pages, base page and next page. Base page provides basic link code information, which is firstly transmitted once power-on, reset removal, and restart, until finishing exchange between local device and remote device. Next page provides more abundant information to exchange.

The state diagram for AN transmission is as the following paragraph:

图 2-15 AN transmit state diagram



Here are some explanations for these states in the diagram:

IDLE state: Initial state for AN transmit operation.



TRANSMIT_DELIMITER state: Begin to transmit Manchester violation delimiter when jump to this state. If transmitted link codewords with the Acknowledge bit set are to be counted, the next state will be TRANSMIT_REMAINING_ACKNOWLEDGE, otherwise will jump to TRANSMIT_ABILITY state directly.

TRANSMIT_REMAINING_ACKNOWLEDGE state: Just set remaining acknowledge counter to initial value 0, then jump to TRANSMIT_ABILITY state without any condition.

TRANSMIT_ABILITY state: Prepare AN code word, then change state from TRANSMIT_CLOCK_BIT and TRANSMIT_DATA_BIT state in a Ping-Pong mode to transfer all AN page information bit by bit. Transmit bit counter will be initialized to 1 and 1-bit pseudo-random data is also initialized at this state.

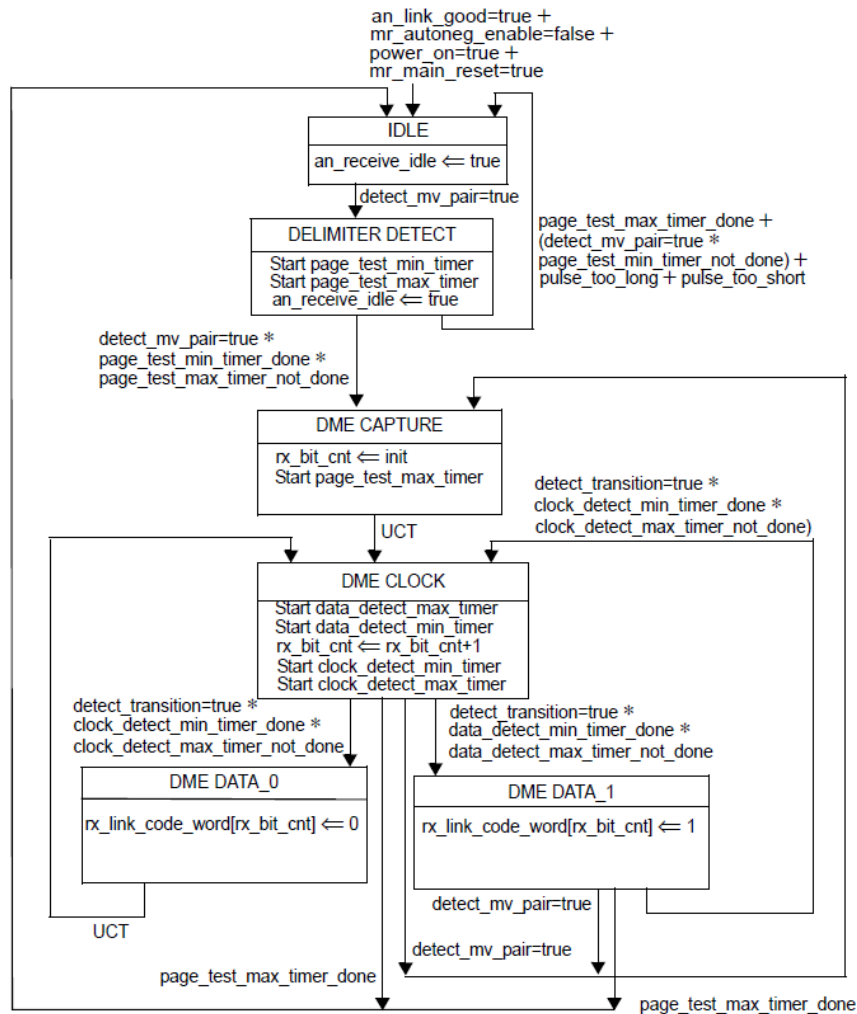
TRANSMIT_CLOCK_BIT and TRANSMIT_DATA_BIT state: Change between the two states and transmit all AN page information after an internal timer is done. This internal timer is to ensure the correct transmit baud rate. When transmit bit count is counted to 50, that means all information has been transmitted. If remaining acknowledge counter is enough (value 6 to 8 is enough), the next state will be TRANSMIT_DELIMITER so that state machine can start a next new page information transmit process, otherwise it will be TRANSMIT_COUNT_ACK to transmit enough acknowledge information to remote link partner.

TRANSMIT_COUNT_ACK state: Remaining acknowledge counter increase 1 and check whether it is enough. The next state will be TRANSMIT_DELIMITER and the AN page information will be transmitted again.

2.7.2 AN Receive Operations

The state diagram for AN receive is as the following paragraph:

图 2-16 AN receive state diagram



Here are some explanations for these states in the diagram:

IDLE state: It's the initial or link good state for AN receive state diagram. Variable `an_receive_idle` is set to 1 to show the state machine is under IDLE or DELIMITER_DETECT state. Once a sequence of three consecutive transitions with 12.8ns +/- 200ps is detected, it will jump to DELIMITER_DETECT state for further process.

DELIMITER_DETECT state: Start page test process, measure page time by starting up `page_test_min_timer` and `page_test_max_timer` and measure pulse time. If a sequence of three consecutive transitions are measured and the page time is between `page_test_min_timer` and `page_test_max_timer`, jump to DME_CAPTURE state to capture page information. If page test time exceeds `page_test_max_timer`, or a sequence of three consecutive transitions are measured less than `page_test_min_timer`, or pulse too long, or pulse too short, jump back to IDLE state to detect the new a sequence of three consecutive transitions.

DME_CAPTURE state: Start `rx_bit_cnt` counter to check the bit number in a DME page. Start `page_test_max_timer` counter to check the total time. Then jump to DME_CLOCK state directly.



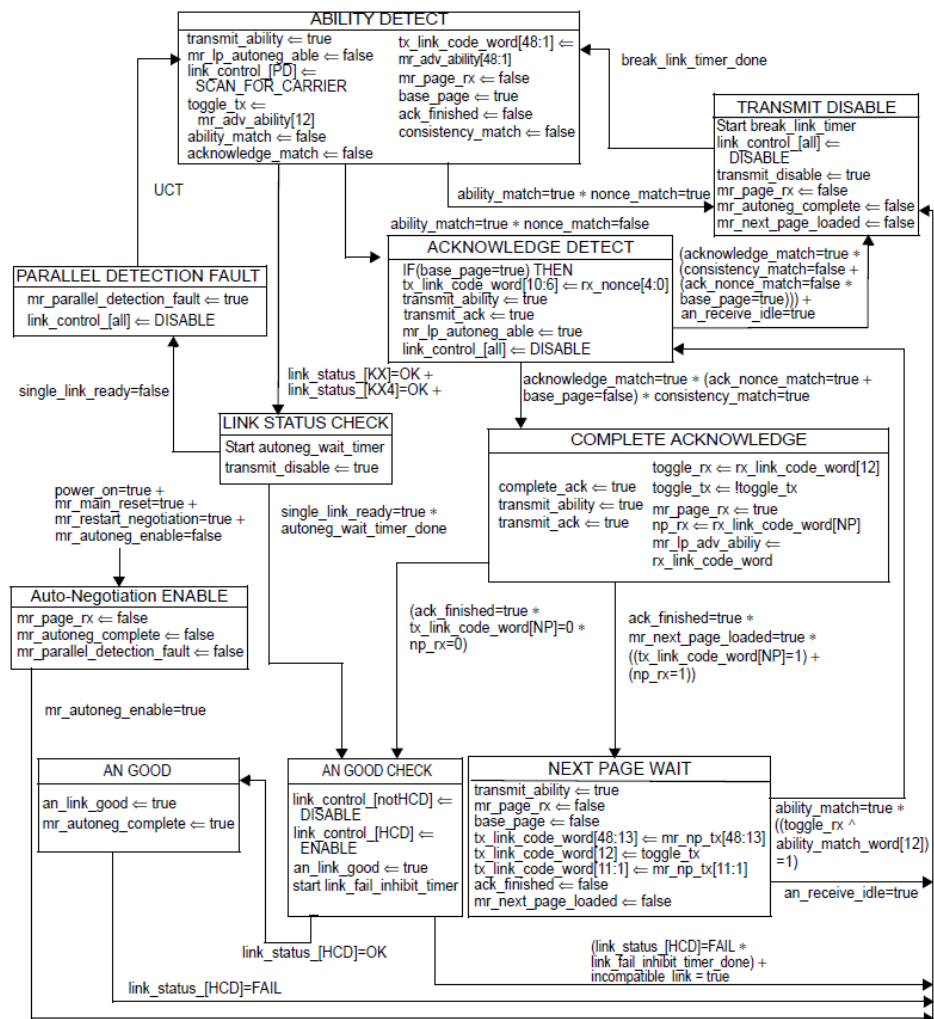
DME_CLOCK state: Start data detect time checker (minimum and maximum) and clock detect timer checker(minimum and maximum). Rx_bit_cnt plus 1. If a transition is detected and clock detect timer is between clock_detect_min_timer and clock_detect_max_timer, that means data 0 is detected and it will jump to DME_DATA_0. If a transition is detected and data detect timer is between data_detect_min_timer and data_detect_max_timer, that means data 0 is detected and it will jump to DME_DATA_1. If page_test_max_timer is done , that means the whole receive process is failed and jump back to IDLE state. If a sequence of three consecutive transitions are measured, that means a whole DME capture is finished and jump back to DME_CAPTURE for a new DME page capture process.

DME_DATA_0 and DME_DATA_1 state: These two states are just fill 0 or 1 to received link code word to related bit according to rx_bit_cnt counter. Normally the next state will be DME_CLOCK for the next round clock and data bit receiving. The only special thing is that DME_DATA_1 will check whether there is a new AN DME violation, if yes it will jump back to DME_CAPTURE for a new DME capture process.

2.7.3 AN Arbitration Operations

The state diagram for AN arbitration is as the following paragraph:

图 2-17 AN arbitration state diagram





Here are some explanations for these states in the diagram:

ABILITY_DETECT state: This is the initial state for AN arbitration state machine. There are two variable are checked. One variable is ability_match which means 3 same consecutive link codewords are received ignoring the Acknowledge bit. Another variable is nonce_match which means remote nonce field match the local nonce field. When both ability_match and nonce_match are asserted , jump to TRANSMIT_DISABLE to disable transmit make the link partner link failed state. When ability_match is asserted and nonce_match is deasserted, jump to ACKNOWLEDGE_DETECT to confirm the Acknowledge bit set. If parallel detection function is enabled and PCS link status for 1GBASE-KX, 2.5GBASE-KX or 10GBASE-KX4 is asserted, jump to LINK_STATUS_CHECK status to do further link status check.

TRANSMIT_DISABLE state: When enter this state, tx_link_code_word[48:1] transmission is halted and wait for some time that is decided by a timer named break_link_timer . The timer shall expire 60 ms to 75 ms after being started. After the timer is timed out , jump back to ABILITY_DETECT state for a new ability detection.

ACKNOWLEDGE_DETECT state: When enter this state, a variable named acknowledge_match which means three consecutive link codewords match and have the Acknowledge bit set is being checked. When these following conditions are met , jump to COMPLETE_ACKNOWLEDGE state, otherwise jump to TRANSMIT_DISABLE state:

1. Once acknowledge_match is asserted;
2. The link codeword that caused acknowledge_match valid same as the link codeword that caused ability_match valid except Acknowledge bit;
3. The echoed nonce received from the link partner matches the transmitted nonce field sent by the local device, or a page other than base link codeword is being transmitted;

COMPLETE_ACKNOWLEDGE state: Set the acknowledge bit and count the transmitted link codewords until 6 to 8 link codewords are transmitted. If don't need to exchange next page information with link partner, the next state will be AN_GOOD_CHECK state , or the next state will be NEXT_PAGE_WAIT state.

LINK_STATUS_CHECK state: When link status from 1000BASE-KX or 10GBASE-KX4 or 2.5GBASE-KX is OK , the state will jump from ABILITY_DETECT state to LINK_STATUS_CHECK state. The autoneg_wait_timer will be started and count when enter the LINK_STATUS_CHECK state. When the autoneg_wait_timer is done , single_link_ready signal will be checked . If single_link_ready is true, the next state will be AN_GOOD_CHECK ; If it is false, the next state will be PARALLEL_DETECTION_FAULT.

2.8 AN arbitration rule

This chapter is to introduce the arbitration rules.



2.8.1 Priority Resolution

The mechanism used by AN module is a Priority Resolution function that predefines the hierarchy of supported technologies. The priority resolution is defined in the following table:

表 2-10 Priority Resolution

Priority (1 has the highest priority,16 has the lowest)	Technology
1	200GBASE-KR4 or 200GBASE-CR4
2	100GBASE-KR2 or 100GBASE-CR2
3	100GBASE-CR4
4	100GBASE-KR4
5	100GBASE-KP4
6	100GBASE-CR10
7	50GBASE-KR or 50GBASE-CR
8	40GBASE-CR4
9	40GBASE-KR4
10	25GBASE-KR or 25GBASE-CR
11	25GBASE-KR-S or 25GBASE-CR-S
12	10GBASE-KR
13	10GBASE-KX4
14	5GBASE-KR
15	2.5GBASE-KX
16	1000BASE-KX

After negotiation, technology resolution will be shown in register AN_LINK_CONTROL_STATUS so that software can set the port to work at this technology.

Specially for MSA((Consortium) application, please refer to the following figure:



图 2-18 MSA Priority Resolution

Priority	Technology	Capability	Note
1	100GBASE-R4	100 Gb/s, 4 lane	IEEE mode
2	50GBASE-R2	50 Gb/s, 2 lanes	Consortium mode
3	40GBASE-R4	40 Gb/s, 4 lanes	IEEE mode
4	25GBASE-R1	25 Gb/s, 1 lane	IEEE mode
5	25GBASE-R1	25 Gb/s, 1 lane	Consortium mode
6	10GBASE-R	10 Gb/s, 1 lane	IEEE mode

2.8.2 FEC Resolution

During negotiation, F0 bit shall be enabled and advertise FEC ability to the link partner. See the following table to FEC bit ability setting:

表 2-11 10G PHY FEC bit F0 setting based on Priority Resolution

Priority	Technology	FEC function setting
1	200GBASE-KR4 or 200GBASE-CR4	-
2	100GBASE-KR2 or 100GBASE-CR2	-
3	100GBASE-CR4	-
4	100GBASE-KR4	-
5	100GBASE-KP4	-
6	100GBASE-CR10	F0 bit shall be enabled
7	50GBASE-KR or 50GBASE-CR	-
8	40GBASE-CR4	F0 bit shall be enabled
9	40GBASE-KR4	F0 bit shall be enabled
10	25GBASE-KR or 25GBASE-CR	-
11	25GBASE-KR-S or 25GBASE-CR-S	-
12	10GBASE-KR	F0 bit shall be enabled
13	10GBASE-KX4	-
14	5GBASE-KR	-
15	2.5GBASE-KX	-



Priority	Technology	FEC function setting
16	1000BASE-KX	-

Only when both devices advertise FEC ability on the F0 bit and at least one device request FEC on the F1 bit, the FEC should be enabled after negotiation, otherwise FEC should be disabled.

After negotiation, 10G PHY FEC resolution will be shown in register AN_STATUS.FEC_NEGOTIATED so that software can enable or disable FEC function.

表 2-12 10G PHY FEC Resolution

Local Device F0	Link Partner F0	Local Device F1	Link Partner F1	10G PHY FEC resolution
0	Don't care	Don't care	Don't care	Disabled
1	0	Don't care	Don't care	Disabled
1	1	1	Don't care	Enabled
1	1	0	1	Enabled

For 25G PHY, the FEC resolution is:

For 25G PHYs if neither PHY requests FEC operation in bits F2 or F3 then FEC is not enabled.

For 25GBASE-KR and 25GBASE-CR PHYs if either PHY requests RS-FEC then RS-FEC operation is enabled, otherwise if either PHY requests BASE-R FEC then BASE-R operation is enabled.

For 25GBASE-KR-S and 25GBASE-CR-S PHYs if either PHY requests RS-FEC or BASE-R FEC then BASE-R operation is enabled. This is because 25GBASE-KR-S and 25GBASE-CR-S PHYs do not support RS-FEC operation.

After negotiation, 25G PHY FEC resolution will be shown in register AN_STATUS.RSFEC_NEGOTIATED so that software can enable or disable FEC function.

Specially for MSA((Consortium) application,

In addition to existing Clause 73 Base Page FEC control bits, four bits are added to the UP-1

(Unformatted Next Page 1) to determine FEC usage. These are F1, F2, F3, and F4. These

bits only apply if the port speed selected by the AN process is either 25G or 50G.

Note the following references:



BP.F0 Refers to the Clause 73 Base Page (BP) F0 (10 Gb/s per lane FEC ability) bit
BP.F1 Refers to the Clause 73 Base Page (BP) F1 (10 Gb/s per lane FEC requested) bit

F1 advertises Clause 91 FEC ability

F2 advertises Clause 74 FEC ability

F3 requests Clause 91 FEC

F4 requests Clause 74 FEC

LD = local device

RD = remote device

Algorithmically, the meaning of these FEC control bits can be expressed as follows:

If the speed Highest Common Denominator is either 25 Gb/s or 50 Gb/s {

If ((LD.F3 | RD.F3) & LD.F1 & RD.F1) { // If either link partner requests the

Clause 91 FEC, and BOTH support it.

Use Clause 91 FEC ;

}

Else if ((LD.F4 | RD.F4) & LD.F2 & RD.F2) { // If either link partner requests the
Clause 74 FEC, and BOTH support it.

Use Clause 74 FEC; // Regardless of the Clause 73 Base
Page FEC bits

}

Else if ((LD.BP.F1 | RD.BP.F1) & LD.BP.F0 & RD.BP.F0) { // Use Clause 73 10G Base
Page FEC control bits to determine whether CL74 FEC is used

Use Clause 74 FEC;

}

Else NO FEC;

}

须知

To guarantee proper FEC operation, it is recommended to program the UP-1 FEC bits and the Clause 73 Base Page FEC bits in a consistent manner.

2.8.3 Pause Resolution

Here is a table to show the pause resolution.



表 2-13 PAUSE Resolution

Local Device		Link partner		Local device resolution	Link partner resolution
C0(PAUSE)	C1(ASM_DIR)	C0(PAUSE)	C1(ASM_DIR)		
0	0	Don't care	Don't care	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
0	1	0	Don't care	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
0	1	1	0	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
0	1	1	1	Enable PAUSE transmit Disable PAUSE receive	Enable PAUSE receive Disable PAUSE transmit
1	0	0	Don't care	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
1	Don't care	1	Don't care	Enable PAUSE Transmit and Receive	Enable PAUSE Transmit and Receive
1	1	0	0	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive



Local Device		Link partner		Local device resolution	Link partner resolution
C0(PAUSE)	C1(ASM_DIR)	C0(PAUSE)	C1(ASM_DIR)		
1	1	0	1	Enable PAUSE receive Disable PAUSE transmit	Enable PAUSE transmit Disable PAUSE receive

After negotiation, PAUSE resolution will be shown in register AN_STATUS.PAUSE_NEGOTIATED so that software can set corresponding local MAC PAUSE transmit and receive function.

2.9 Control Processor Access

It's same as HIMACV310 common Control Processor Access. Please see “HIMACV310 400GE Combo FS” chapter 1.1.28 Control Processor Access for more information.

2.10 Error Handling

If the interrupt is AN_UPDATE and register AN_STATUS.RENEW_PAGE is logic 1, that means local SerDes received some error data and local state machine enter TRANSMIT_DISABLE state. When enter this state, software need to configure the first two next page information into AN_XNP_ABILITY0_CFG , AN_XNP_ABILITY1_CFG , AN_XNP_ABILITY2_CFG and AN_XNP_ABILITY3_CFG again in break_link_timer(60ms to 75ms). After that, state machine will jump back to ABILITY_DETECT state for a new ability detection.

2.11 Initialization

AN initialization process is :

Assert RST_AN_CORE_N , RST_AN_RS_N , RST_AN_CFG_N , RST_AN_RX_SDS_N and RST_AN_TX_SDS_N to logic 0 at least 100ns so that all circuits will be successfully reset.

Wait until the local SerDes has been successfully ramped up. Then assert RST_AN_RS_N , RST_AN_CFG_N to logic 1 to enable register configuration.

Config register AN_MODE_CFG to set AN mode configurations.

Config register AN_TIMER0_CFG, AN_TIMER1_CFG, AN_TIMER2_CFG and AN_TIMER_UNIT_CFG to set related timers and timer settings.

Config AN_BP_ABILITY0_CFG and AN_BP_ABILITY1_CFG to set local base page information.



If next page function is supported, config AN_XNP_ABILITY0_CFG, AN_XNP_ABILITY1_CFG, AN_XNP_ABILITY2_CFG, and AN_XNP_ABILITY3_CFG to set local next page information.

Config An_RX_SAMPLE_TOLERANCE_CFG to set sample position deviation.

Assert RST_AN_CORE_N, RST_AN_RX_SDS_N and RST_AN_TX_SDS_N to logic 1.

Set AN_CTRL_CFG.AUTONEG_ENABLE to logic 1 to enable AN function and start AN progress.

Wait until some interrupt is asserted:

- If the interrupt is AN_UPDATE and register AN_STATUS.RENEW_PAGE is logic 0, config AN_XNP_ABILITY0_CFG, AN_XNP_ABILITY1_CFG, AN_XNP_ABILITY2_CFG and AN_XNP_ABILITY3_CFG will be loaded new next page contents then set AN_CTRL_CFG.NEXT_PAGE_LOADED to logic 1 to wait another interrupt;
- If the interrupt is AN_SUCCESS, the AN progress is successfully done and link training progress can be started;

If the interrupt is AN_FAIL, the AN progress is failed.

If the interrupt is AN_NO_HCD, the AN progress is successfully done but no HCD is found.

2.12 Debug Features

Some interrupts are defined for debugging. They are defined in register AN_INT_STATUS/AN_INT_ENABLE/AN_INT_SET.

Interrupt AN_NO_HCD: AN not find HCD. It means the AN progress is successfully done but not HCD is found.

Interrupt AN_UPDATE: AN next page update interrupt. It means software need to fill new next page information to related register.

Interrupt AN_FAIL: AN fail interrupt.

Interrupt AN_SUCCESS: AN success interrupt.

Interrupt IERR_C: Correctable integrity error interrupt.

Interrupt IERR_U: Uncorrectable integrity error interrupt.

Some counters are defined for debugging.

Counter MORE_TRANS_ERR(See register AN_RX_DME_STAT_CNT): Count the number of more-transition errors during DME detection.

Counter TWO_TRANS_ERR(See register AN_RX_DME_STAT_CNT): Count the number of two-transition with abnormal distance during DME detection.

Counter IERR_C_CNT(See register AN_IERR_C_CNT): Count the IERR_C interrupt numbers.

Counter IERR_U_CNT(See register AN_IERR_U_CNT): Count the IERR_U interrupt numbers.



Counter AN_GOOD_CHECK_CNT(See register AN_DBG_TIMER): Count the time from AN_GOOD_CHECK state to MAC link up status state.

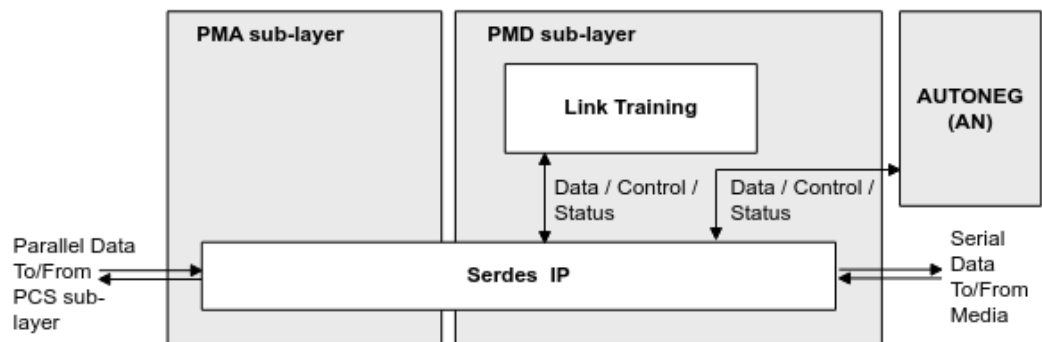
Some debug information registers are defined in AN_DBG_INFO . For more information , please refer to the register XML document.

3_{LT}

3.1 Link Training (LT)

Some Ethernet protocols such as IEEE 802.3 standard and 25G & 50G Ethernet Consortium have defined their Ethernet PMA and PMD sub-layers to meet system conformance and communicate with different physical media. Most high speed SerDes IP such as HiLink IP designed by Hisilicon or other commercial SerDes IP have implemented most function of transmitter and receiver in PMD sub-layer. However, the link training function is not implemented in HiLink SerDes and many other commercial SerDes. An additional Link Training module is defined in PMD sub-layer to complete the link training function missed by SerDes IP.

图 3-1 Link training module defined in PMD sub-layer



3.2 Main Features

The main features of the Link Training Block are as follows:

- This Link Training block is only applicable for HiLink H32 SerDes.
- Support the following Ethernet standards that define link training function:
Supports IEEE 802.3[™] 2018 (See clause 72.6.10 , clause 72.7 and clause 92)



Support 25G & 50G Ethernet Consortium standard (See Chapter 3.2.5.3 Training).

- Supported SerDes Speed from 10.3125G to 30Gbps.
- Support both hardware training mode and software training mode.
- Support 548 Bytes training frame structure sending and receiving during training.

Support Differential Manchester Encoding (DME) algorithm is used to encode and decode control channel information in training frame structure.

Support coefficient update field to preset, initialize and update 3 transmit equalizer taps.

Support sending report receiver status and coefficient status to the link partner.

Support configurable PRBS training pattern. User can select fixed PRBS11mode or configurable polynomial identifier and related binary seeds.

- Support hardware training algorithm, it is:
For NRZ hardware training algorithm, the size of eye diagram is the metric that measure signal quality.

Configurable training step size.

Training time counter all the training time cost.

- Support parallel data bus to exchange training data with SerDes.
Supports 16/20/32/40 bits data bus width.
16/20/32/40 bits work for 10.3125G~30Gbps SerDes .
Support bit reverse function to conform to different SerDes device.
- Support SPI master interface to exchange control or status information with SerDes.

SPI master interface is applicable for HiLink H32 SerDes, etc.

Can read eye information then set pre-cursor, post-cursor and cur-cursor value.

- Support T interface for CPU command and status register arrays accessing:
Normal CPU register read and write access.
- Maximum training time is configurable.
Default value is 500ms and the maximum value can reach 50s.
Maximum training time can also be disabled.

3.3 Terminology

The following terms have specific meanings in this section of the document.

表 3-1 Link Training Terminology

Term	Description
AHB	Advanced High Performance Bus



Term	Description
CPU	Center Processing Unit
DME	Differential Manchester Encoding
FIR	Finite Impulse Response
IEEE	Institute of Electrical and Electronics Engineers
LT	Link Training
MAC	Media Access Control
PCS	Physical Coding Sub-layer
PMD	Physical Medium Dependent
PRBS	Pseudo-Random Binary Sequence
SPI	Serial Peripheral Interface

3.4 References

The following documents provide useful information related to this section of the document.

- IEEE std 802.3™ 2018
- 25G & 50G Specification, Aug 18, 2015

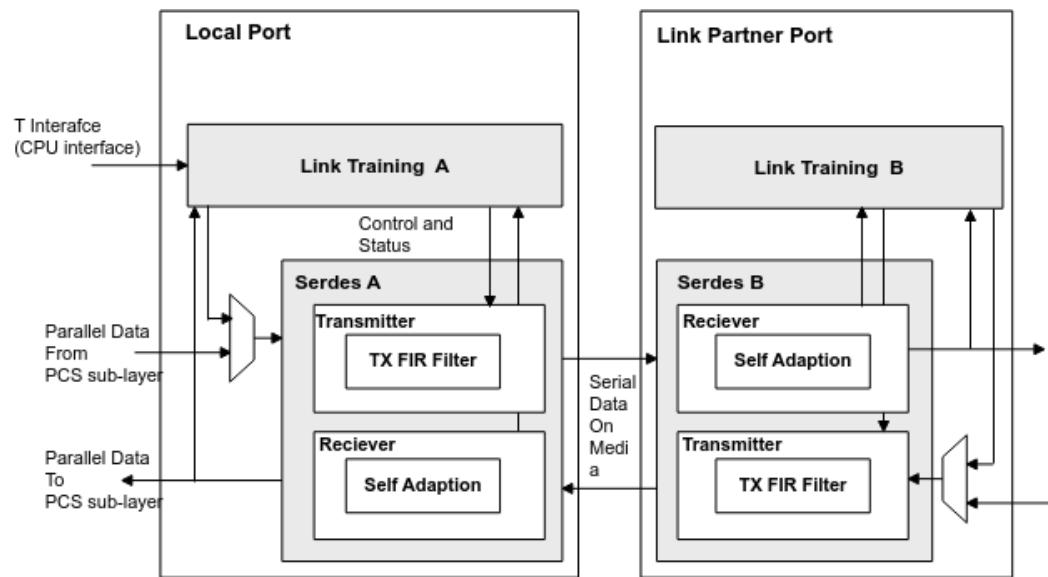
3.5 Link Training Overview

3.5.1 Introduction

Link training module mainly works with local SerDes IP and implements the complete the link training control function (See IEEE 802.3 2018 clause 72.6.10 and 72.7) .

Please see link training diagram in [图3-2](#):

图 3-2 Link Training Diagram



In this diagram we only draw a SerDes lane and a link partner SerDes lane as an example to explain how link training module work. Every SerDes lane has a receiver part and a transmitter part. In fact the training operation could be started independently for every SerDes lane.

Every SerDes slice has implemented a 3-tap FIR (Finite Impulse Response) equalization circuits on transmit side and also has implemented a self-adaption circuits on receive side. When SerDes starts up, it will power up and run transmit side and receive side calibration processing first. After that SerDes will set all FIR coefficients at an initial value then run receiver adaption. Once receiver adaption process has finished, the SerDes can stay in ready state for normal data transferring.

However, SerDes is probably not very stable in ready state because of various reasons. One of these reasons maybe the initial tap coefficients are not good enough in bad environment and we need to explore and adjust all tap coefficients setting step by step so that SerDes can transmit a better pre-emphasized signal. We call it as “link training” process.

An additional link training block and a serial of related processes are defined for this training process. This training process can be started up by software or hardware on either side (local or link partner side). A complete training flow is defined to explore all possible 3-tap FIR coefficients. Once the optimized FIR coefficients are found, link training process is successfully done and SerDes will be stable enough under data ready state.

Training support both hardware training and software training. But the procedures of both are nearly the same.

A training procedure would follow these steps (Use software training as an example):

- 步骤1** Software sets related training option registers then sends training enable command to link training module. The data bus from/to SerDes PCS layer will be



bypassed and training data bus from/to link training module will be selected. That means SerDes data bus is controlled by link training module.

- 步骤2** Local link training module will send training frame to remote link partner. In the training frame a coefficient update field is set and all transmit taps coefficients can be called for “updated”. Link partner does the same thing at the same time.
- 步骤3** Training frame is analyzed by the link partner and updates tap coefficients as training frame requested. Once it is updated, link partner will send coefficient status acknowledgement to local link training module. Local link training module also does the same thing at the same time.
- 步骤4** Local LT module sends training frame and calls for keeping all transmit tap coefficients “hold”. During this time, local LT module can gather all statistics information and calculate new training information for the next update. Link partner does the same thing at the same time.
- 步骤5** If the LT module determines that remote TX FIR has been finished training and a local receive ready signal is asserted then continue step 6 and wait for remote link partner assert receiver ready signal, otherwise it will generate new training frame calculated by specific exploring algorithm and continue Step 2.
- 步骤6** Once remote link training is also done and a remote receive ready signal is asserted. LT module will wait for some time then finish the whole training process.
- 步骤7** Finally software can release the control right on data bus. SerDes PCS layer can send and receive data normally.

----结束

3.5.2 Connectivity

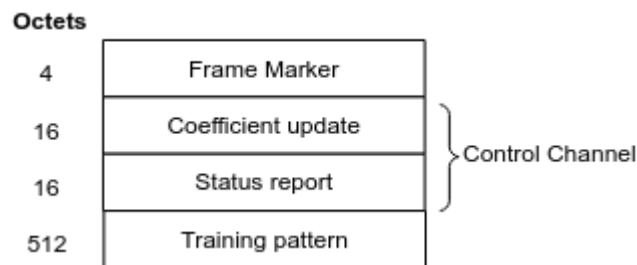
The LT module communicates with other blocks several different ways:

- LT will share SerDes parallel transmit and receive data bus with PCS layer. When hardware or software link training mode is enabled, LT will control the transmit data bus completely.
- LT will provide T interface for CPU accessing. T interface can be integrated by higher CPU accessing circuits so that CPU so that software can control LT module.
- LT will communicate with SerDes register control bus so that LT can send control information or command and check needed SerDes status. There are two different ways:
SPI interface. It is supported by HiLink H32 SerDes, etc.
- Some independent status signals from SerDes should be connected to LT module.

3.6 Training Frame Data Structure (548 Bytes)

The training frame is a fixed length structure that is sent continuously during training. The training frame is shown in the following figure.

图 3-3 Training Frame Data Structure (548 Bytes)



3.6.1 DME coder and decoder

The DME encoding rules are as the followings:

- There are two possible data transitions for every bit information transmission. One is clock transition time, the other is data transition time. That means the data transfer efficiency is 50%.
- At clock transition time, 1 should be set to 0 and 0 should be set to 1.
- At data transition time, if we want to signal a logic one, 1 should be set to 0 and 0 should be set to 1.
- At data transition time, if we want to signal a logic zero, just keep current value and any transitions.

说明

If a coding violation is detected within the bounds of the control channel in a given training frame, the contents of the control channel for that frame shall be ignored.

3.6.2 Frame marker field

Frame marker field is a fixed 32-bit pattern and the content is hexadecimal FFFF0000 (ones transmitted first). This pattern is detected by remote link partner to delimit the training frame header. It is a unique indicator of the start of a training frame and it does not appear in the control channel or the training pattern field.

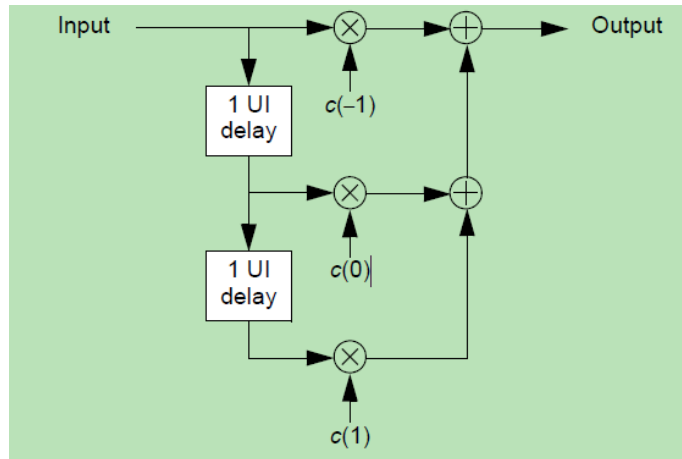
3.6.3 Control channel field

Control channel field contains the remote FIR coefficients update command and related parameters, local link status and local FIR coefficients update status. All these fields are using DME coder/decoder.

3.6.3.1 Coefficient update field

Before introduce coefficient update field, we need to know some background information about A typical SerDes transmit equalizer implementation. The SerDes transmitter includes programmable equalization to compensate for frequency-dependent loss in the backplane channel and facilitate data recovery at the receiver. It is accomplished with a 3 tap finite impulse response (FIR) structure as shown in the following Figure.

图 3-4 3-tap FIR Filter Structure



That show there are 3 coefficients $c(0)$, $c(+1)$ and $c(-1)$. $c(+1)$ and $c(-1)$ only have negative values. $c(0)$ only have positive value.

The coefficient update field carries correction information from the local receiver to the link partner transmits equalizer. The field consists of preset controls, initialization controls, and coefficient updates for three transmit equalizer taps. The format of the coefficient update field shall be as shown in the following table. Cell 15 of the coefficient update field sent shall be transmitted first.

表 3-2 Coefficient update field

Cell(s)	Name	Description
15:14	Reserved	Transmitted as 0, ignored on reception
13	Preset	When set to 1, preset command is issued. 1=Preset coefficients 0=Normal operation
12	Initialize	When set to 1, initialize command is issued. 1=Initialize coefficients 0=Normal operation
11:8	Reserved	Transmitted as 0, ignored on reception
7:6	Reserved	Reserved
5:4	Coefficient $c(+1)$ update	5 4 1 1 =reserved 0 1 = increment 1 0 = decrement 0 0 = hold



Cell(s)	Name	Description
3:2	Coefficient c(0) update	3 2 1 1 =reserved 0 1 = increment 1 0 = decrement 0 0 = hold
1:0	Coefficient c(-1) update	1 0 1 1 =reserved 0 1 = increment 1 0 = decrement 0 0 = hold

3.6.3.2 Status report field

The status report field is used to signal state information from the local PMD to the link partner. The format of the status report field is shown in the following table. Cell 15 of the status report field shall be transmitted first.

表 3-3 Status report field

Cell(s)	Name	Description
15	Receive ready	This bit is used to signal the local receiver state to the link partner. When asserted, the receiver ready bit indicates that the local receiver has concluded training and is prepared to receive data. When de-asserted, the receiver ready bit indicates that the local receiver is requesting that training continue. 1 = The local receiver has determined that training is complete and is prepared to receive data. 0 = The local receiver is requesting that training continue
14:8	Reserved	Transmitted as 0, ignored on reception
7:6	Reserved	Reserved



Cell(s)	Name	Description
5:4	Coefficient(+1) status	5 4 1 1 = maximum 1 0 = minimum 0 1 = updated 0 0 = not_updated
3:2	Coefficient(0) status	3 2 1 1 = maximum 1 0 = minimum 0 1 = updated 0 0 = not_updated
1:0	Coefficient(-1) status	1 0 1 1 = maximum 1 0 = minimum 0 1 = updated 0 0 = not_updated

3.6.4 Training pattern field

The training pattern shall be a 512 octet pattern consisting of 4094 bits from the output of a pseudo-random bit sequence generator followed by two zeros.

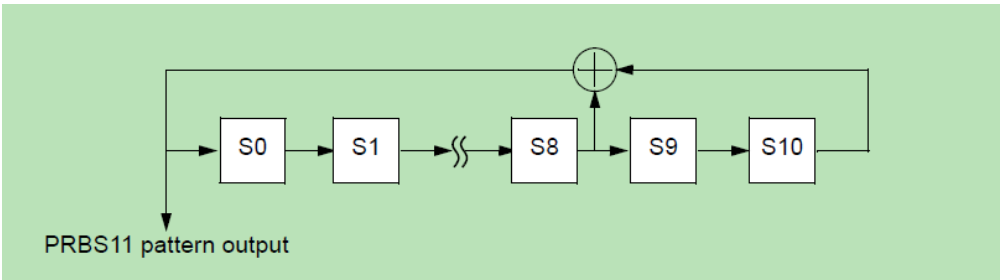
Two types of pattern generators are supported in LT module :

The first one implements PRBS11 pattern generator. The generator polynomial is:

$$G(x) = 1 + x^9 + x^{11}$$

The following figure is PRBS11 pattern generator .This pseudo-random generator shall have a random seed at the start of the training pattern. In IEEE standard 802.3bj 2014, it is defined that each bit of the training pattern is transmitted as a single 10.3125Gbps symbol.

图 3-5 PRBS11 pattern generator



The second one implements 4 different generator polynomials to support 4 different SerDes lane.

To minimize the correlation between physical lanes, the PRBS generator for each lane shall implement each of the four generator polynomials (polynomial_n where n goes from 0 to 3) given in the following table. It is different from the first one that it is not a random seed but a fixed seed listed in the following table. The first 32 bits of the training pattern for each polynomial is also provided in the following table.

表 3-4 Different PRBS parameters for 4 different physical lane

Lane Number	Polynomial_n,G(x)	Default seed bits (S0 is the left most bit)	Initial output
0	$1 + x^5 + x^6 + x^{10} + x^{11}$	10101111110	FBF1CB3E
1	$1 + x^5 + x^6 + x^9 + x^{11}$	11001000101	FBB1E665
2	$1 + x^4 + x^6 + x^8 + x^{11}$	11100101101	F3FDAE46
3	$1 + x^4 + x^6 + x^7 + x^{11}$	11110110110	F2FFA46B

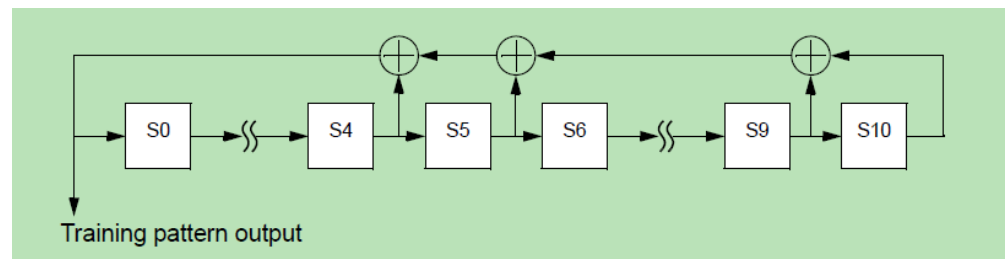
Table Note: The initial output value is presented in a hexadecimal representation where the hex symbols are transmitted from left to right and the most significant bit of each hex symbol is transmitted first.

For lane0, the generator polynomial is:

$$G(x) = 1 + x^5 + x^6 + x^{10} + x^{11}$$

The implementation of the PRBS generator for lane0 with default settings is given in the following figure:

图 3-6 PRBS pattern generator for polynomial_0



LT_PRBS_CFG.CFG_PRBS_SEED_MODE register has defined 3 fields to support various PRBS configuration scenarios:Random mode, multiple fixed mode and custom mode.

- Random mode: It is used for PRBS11 mode and random seed is internally generated.



- Multiple fixed mode: It is used for 4 different PRBS lanes for 4 different physical lane. Which lane is supported is decided by LT_PRBS_CFG.CFG_PRBS_SEED_FIXED. For more information, please see register description.
- Custom mode: It is used for customized seed for PRBS11.

3.7 Hardware Link Training Process

3.7.1 Frame locking process

3.7.1.1 Training frame marker detection

For 548 Bytes training frame, it is delimited by training frame marker and the 32-bit pattern value is hexadecimal 0xFFFF0000.

The MARKER_DEC control field (register LT_CONTROL_CFG1.CFG_MARKER_DEC) is defined to control the mode of the detection of Frame Markers.

The MAN_DEC control field (register LT_CONTROL_CFG1.CFG_MAN_DEC) is defined to control the mode of the decoder of Manchester bits.

Decode is performed as shown in the following Figures:

表 3-5 Frame marker detection setting for NRZ code

MARKER_DEC Setting	Frame Marker
00	1111_1111_1111_1111_0000_0000_0000_0000
01	XX11_1111_1111_1111_0000_0000_0000_00XX
10	XXX1_1111_1111_1111_0000_0000_0000_0XXX
11	XXXX_1111_1111_1111_0000_0000_0000_XXXX

表 3-6 Manchester decode Setting for NRZ code

MAN_DEC Setting	Manchester logic 0	Manchester logic 1
00	0000_0000 or 1111_1111	1111_0000 or 0000_1111
01	X000_000X or X111_111X	X11X_X00X or X00X_X11X
10	XX00_00XX or XX11_11XX	X11X_X00X or X00X_X11X

Please note that frame marker and Manchester logic 0 or 1 detection should be implemented independently.



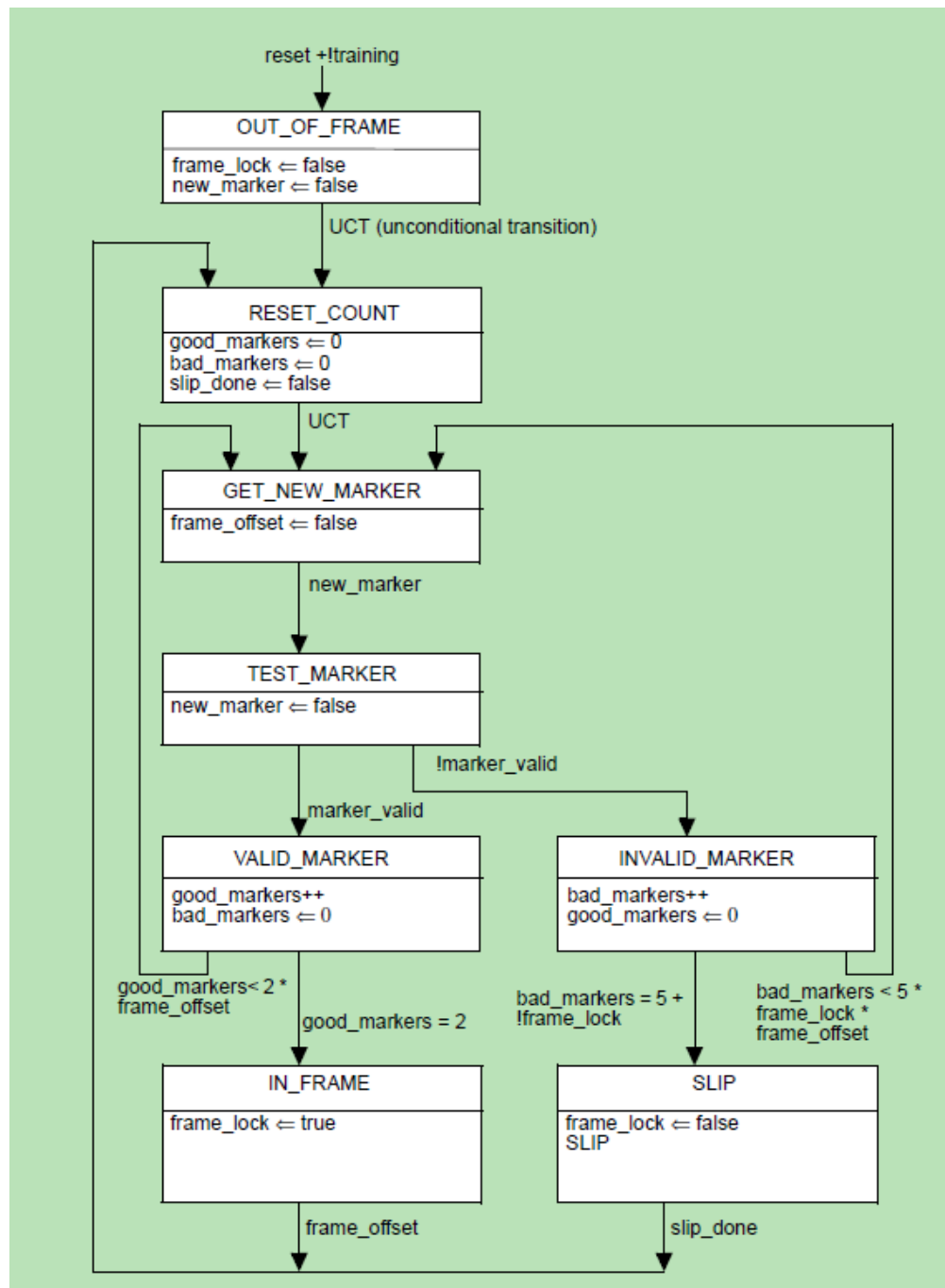
3.7.1.2 Receiver data alignment

The data received from SerDes bit order may not follow the transmitter's. It is necessary to adjust the bit order to get the correct bit order. And abstract the taps status from the transmitted frame. And the control channel different Manchester encode rule is also checked if the check is failed, this frame of data would be ignored

3.7.1.3 Training lock state machine

Here's the training lock state machine:

图 3-7 Training lock diagram



The LT module shall implement the training lock diagram. The frame lock diagram determines when the PMD control function has detected the frame boundaries in the received data stream.

Here are some explanations for these state in the diagram:

OUT_OF_FRAME: It's the initial state after reset. Another condition is that the whole training process has been finished then the state machine will stay in this state.

RESET_COUNT: This state is only for Prepare for marker sampling. Clear good marker counter and bad marker counter. Clear slip_done flag so that next step can sample and detect frame marker.

GET_NEW_MARKER: This state begins to sample frame marker data.

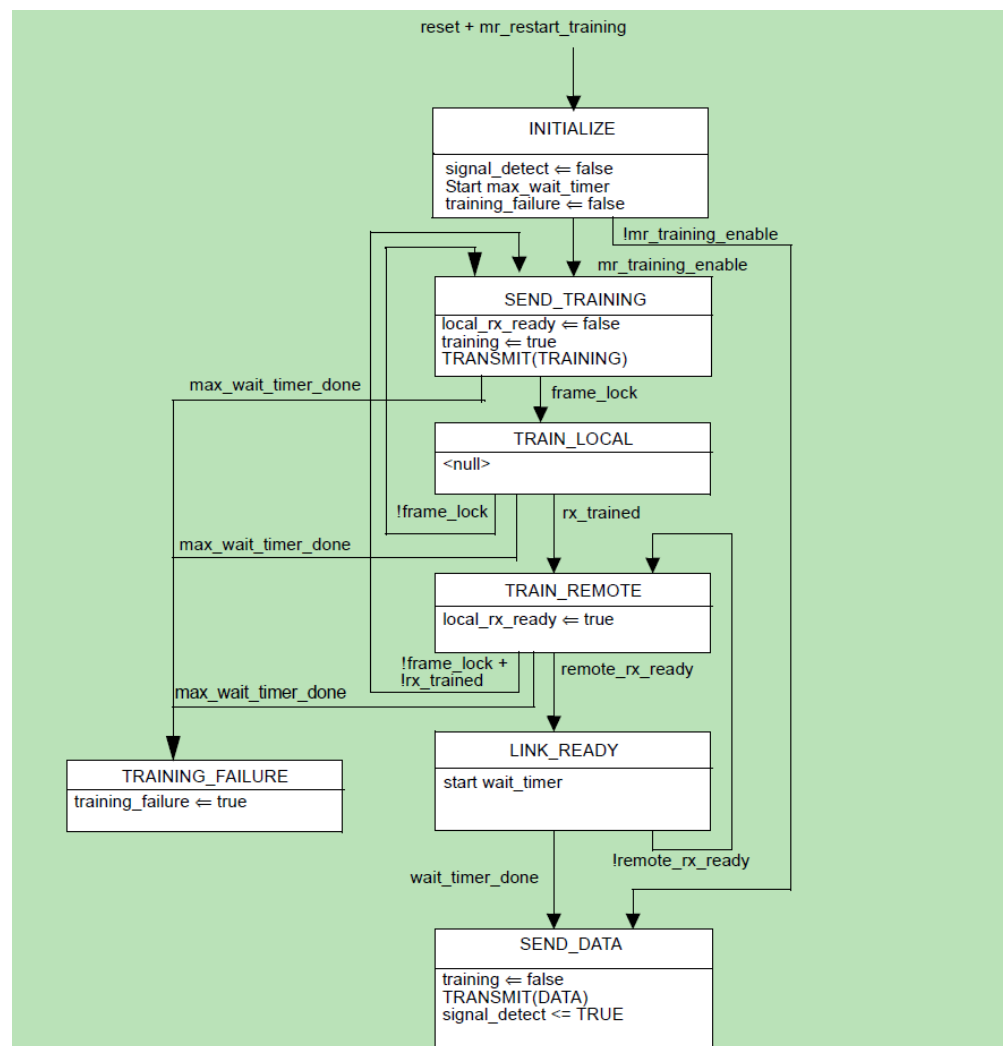
TEST_MARKER: This state checks if a sample frame marker is a valid marker.

If the valid frame marker is detected more than two times successively and the differential Manchester encodes is valid the tap updates command and tap state from the training frame is collect. The tap update command is used for local transmitter tap weight updating and the tap status is used for to explore the optimal eye margin.

3.7.2 Link training process

Link training process for 548 Bytes can be shown in the following figure:

图 3-8 Training flow diagram (548Bytes)



Here are some explanations for these states in the diagram:



INITIALIZE state: Initial state for training main FSM. If training function is disabled, then jump to SEND_DATA state directly. If training function is enabled, then jump to SEND_TRAINING state to start a new link training process. Please note that maximum wait timer is started at this state.

SEND_TRAINING state: Send training frame and receive response frame from remote link partner. If received response frame is locked, then it will jump to TRAIN_LOCAL state. If received response frame is not locked and the maximum wait timer is timed out, it will jump to TRAINING_FAILURE state.

TRAIN_LOCAL state: Start local training process and wait until this process is finished. Once local training is finished, it will jump to TRAIN_REMOTE state. If local training is not finished and the maximum wait timer is timed out, it will jump to TRAINING_FAILURE state. Once frame is not locked, it will jump back to SEND_TRAINING state again.

TRAIN_REMOTE state: Start remote training process and wait until this process is finished. Once remote training is finished and remote_rx_ready signal is asserted, it will jump to LINK_READY state. If remote training is not finished and the maximum wait timer is timed out, it will jump to TRAINING_FAILURE state. Once frame is not locked or local training state is not OK, it will jump back to SEND_TRAINING state again.

LINK_READY state: Wait until link wait timer is timed out then confirm the training process is done. It will jump to SEND_DATA state. Once remote_rx_ready signal is not valid and REMOTE_RX_READY_CHECK_EN (LT_CONTROL_CFG2.REMOTE_RX_READY_CHECK_EN in XML) is also set to 1, then it will jump back to TRAIN_REMOTE and start remote training process again.

SEND_DATA state: The whole training process is done and signal_detect to PCS layer is asserted valid.

Some comments for the timers in the diagram :

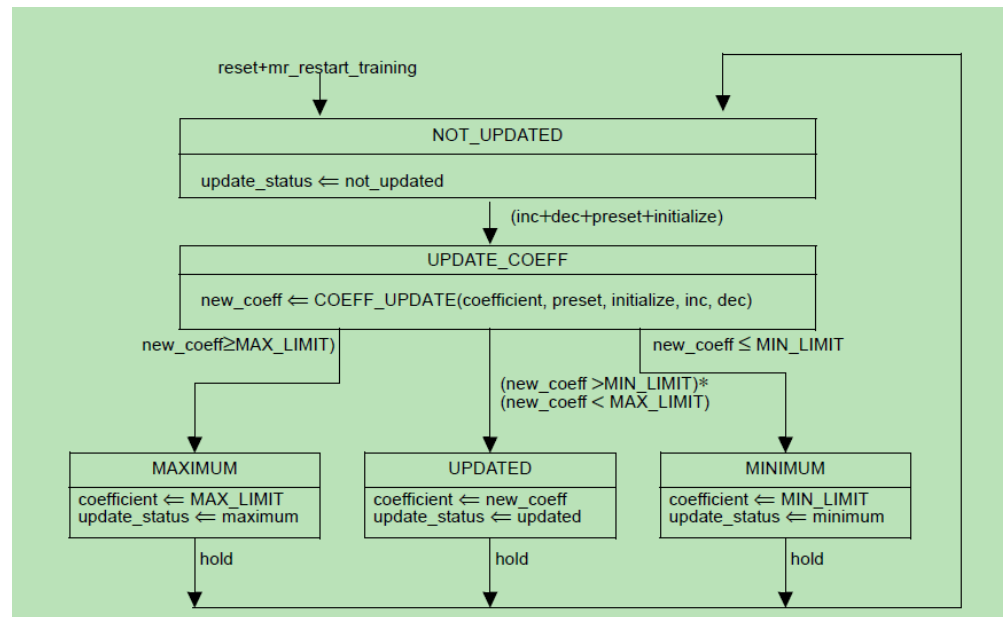
max_wait_timer: the terminal count of max_wait_timer is 500ms±1%.

wait_timer: the wait_timer shall be between 100 and 300 training frames.

3.7.3 Coefficient update process

The Coefficient update process for 548 Bytes can be shown in the following figure:

图 3-9 Coefficient update flow diagram (548Bytes)



Here are some explanations for these states in the diagram:

NOT_UPDATE state: If reset is asserted or training is restarted, state machine will be initialized to this state; update status is set to not_updated. Another condition is that coefficient has been updated and the next training frame contains hold information is received, state machine will jump back to this state.

UPDATE_COEFF state: If a command is received from remote partner that has increase, decrease, preset or initialize request, state machine will jump from NOT_UPDATE to this state.

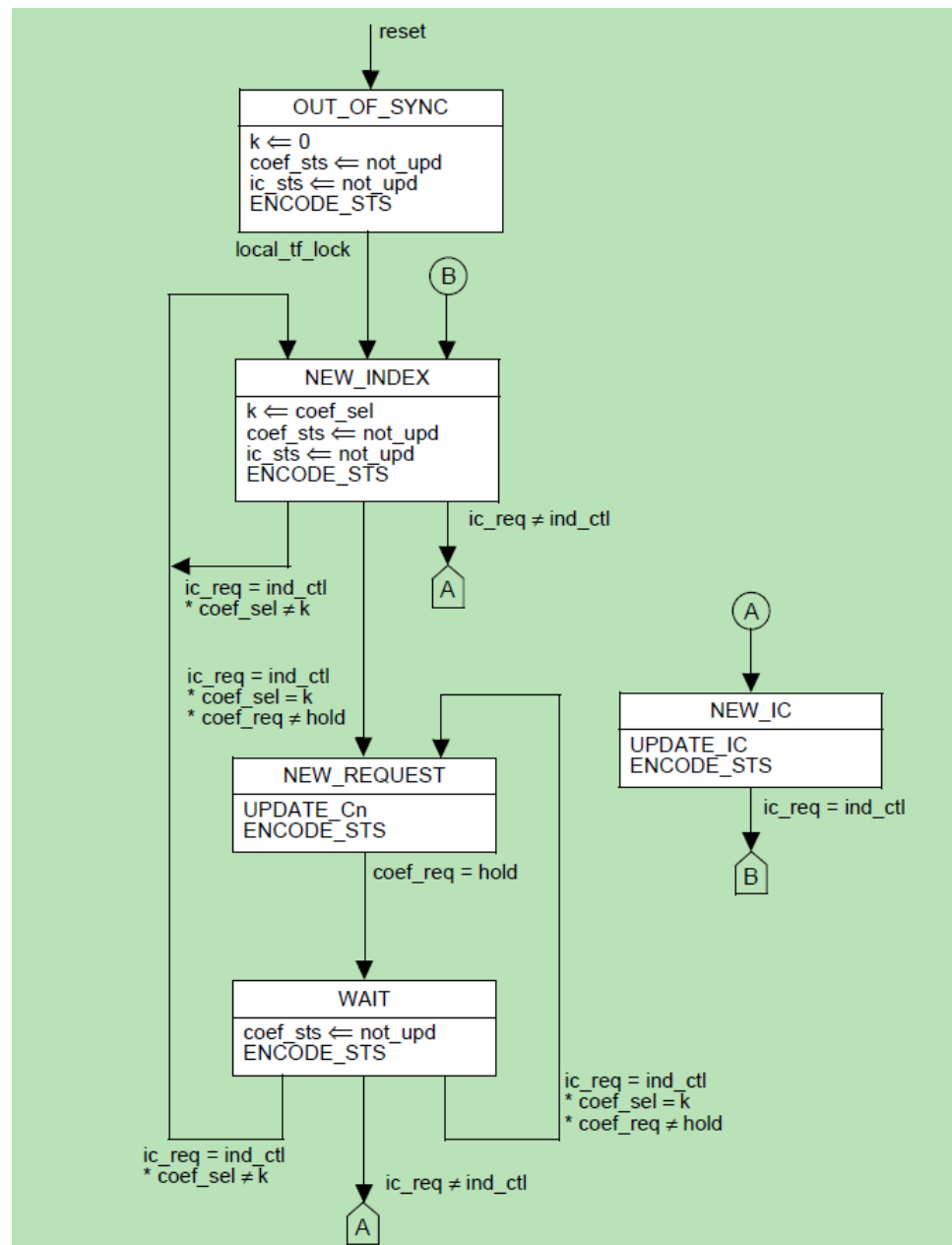
MAXIMUM state: If the new coefficient is even bigger or equal to maximum value, then the new coefficient will be updated to maximum value and the coefficient value from remote partner will be ignored. After a whole training frame is received and the coefficient update field of that frame for this coefficient is hold, then jump back to NOT_UPDATE state.

MINIMUM state: If the new coefficient is even smaller or equal to minimum value, then the new coefficient will be updated to minimum value and the coefficient value from remote partner will be ignored. After a whole training frame is received and the coefficient update field of that frame for this coefficient is hold, then jump back to NOT_UPDATE state.

UPDATED state: If the new coefficient is in normal value, then the new coefficient will be updated to the coefficient value from remote partner. After a whole training frame is received and the coefficient update field of that frame for this coefficient is hold, then jump back to NOT_UPDATE state.

The Coefficient update process for 4168 Bytes can be shown in the following figure:

图 3-10 Coefficient update flow diagram (4168Bytes)



Here are some explanations for these states in the diagram:

OUT_OF_SYNC state: If reset is asserted or training is restarted, state machine will be initialized to this state. When state machine is at this state, coefficient index is set to 0, coefficient status is set to “not updated”, initial condition request is set to “not updated”. Execute ENCODE_STS function at the same time. ENCODE_STS function will encode and update the status field of transmitted training frames. Jump to NEW_INDEX state when local training frame is locked.

NEW_INDEX state: At this state, coefficient index is updated to “coefficient select” field of received training frames so that a new coefficient index is loaded. Both coefficient status and initial condition request are set to “not updated”



and fill “status field” in training frame at the same time. The next state generation rule is:

- “Initial condition request” == “Individual coefficient control” and “coefficient select” == current coefficient index and “coefficient request” != hold, the next state is NEW_REQUEST;
- “Initial condition request” != “Individual coefficient control”, the next state is NEW_IC;
- “Initial condition request” == “Individual coefficient control” and “coefficient select” != current coefficient index, just stay at NEW_INDEX state;

NEW_REQUEST state: At this state, new coefficient will be updated and fill the whole “status field” in training frame. Please refer to chapter 1.1.6.4.6 Coefficient status for the coefficient updating algorithm. If remote link partner has sent hold value in “coefficient request” field, jump to WAIT state, otherwise just stay at NEW_REQUEST state.

WAIT state: This state mainly waits for new request, new coefficient index or new initial condition request for next action. At this state, coefficient status is set to “not updated” and fill the whole “status field” in training frame. The next state generation rule is as same as NEW_INDEX state.

NEW_IC state: This state mainly execute initial condition request. After that the whole “status field” in training frame will be updated thoroughly. If “Initial condition request” is changed to “Individual coefficient control”, jump to NEW_INDEX state, otherwise just stay at NEW_IC state.

3.7.3.1 Controlling SerDes cursor value for initialization or coefficients update command (HiLink H32)

The initial, preset, maximum value and minimum value of Main-cursor, pre-cursor and post cursor can be set by various registers.

- Main-cursor can be set by LT_MAIN_CURSOR_CFG.CFG_TRAIN_INIT_MAIN after initialization or remote link partner sends an INITIALIZE command. Pre-cursor and post cursor have similar configuration register. Also see LT_PRE_CURSOR_CFG.CFG_TRAIN_INIT_PRE and LT_POST_CURSOR_CFG.CFG_TRAIN_INIT_POST for pre-cursor and post-cursor initialization or INITIALIZE command setting progress.
- Main-cursor can be set by LT_MAIN_CURSOR_CFG.CFG_TRAIN_PRESET_MAIN after remote link partner sends a PRESET command. Pre-cursor and post cursor have similar configuration register. Also see register LT_PRE_CURSOR_CFG.CFG_TRAIN_PRESET_PRE and LT_POST_CURSOR_CFG.CFG_TRAIN_PRESET_POST for pre-cursor and post-cursor PRESET command setting progress.
- Main-cursor maximum value and minimum value will be checked and the reference maximum or minimum value can be set by LT_MAIN_CURSOR_CFG.CFG_TRAIN_MAX_MAIN and LT_MAIN_CURSOR_CFG.CFG_TRAIN_MIN_MAIN. Also see register LT_PRE_CURSOR_CFG.CFG_TRAIN_MAX_PRE, LT_PRE_CURSOR_CFG.CFG_TRAIN_MIN_PRE, LT_POST_CURSOR_CFG.CFG_TRAIN_MAX_POST,



LT_POST_CURSOR_CFG.CFG_TRAIN_MIN_POST.

Specially, there is a configuration register LT_CONTROL_CFG1.CFG_AP_MAIN_MODE to control local current cursor update strategy.

- If LT_CONTROL_CFG1.CFG_AP_MAIN_MODE = 1, any main-cursor increase or decrease command from remote link partner will not go into effect. Main-cursor value will be only affected by pre-cursor or post-cursor update commands from remote link partner. The update strategy is:
Pre-cursor increase command received, main-cursor will increase relatively;
Pre-cursor decrease command received, main-cursor will decrease relatively;
Post-cursor increase command received, main-cursor will increase relatively;
Post-cursor decrease command received, main-cursor will decrease relatively.
Main-cursor increase command received, MAXIMUM status will be sent to remote link partner.
Main-cursor decrease command received, MINIMUM status will be sent to remote link partner.
- If LT_CONTROL_CFG1.CFG_AP_MAIN_MODE = 0 and LT_CONTROL_CFG1.CFG_AP_VPK_MODE = 1, main-cursor will be a fixed value. Also see register definition LT_CONTROL_CFG1.CFG_AP_VPK_MODE. The update strategy is:
Any pre-cursor or post-cursor update command received, main-cursor will not be changed;
Main-cursor increase command received, MAXIMUM status will be sent to remote link partner.
Main-cursor decrease command received, MINIMUM status will be sent to remote link partner.
- If LT_CONTROL_CFG1.CFG_AP_MAIN_MODE = 0 and LT_CONTROL_CFG1.CFG_AP_VPK_MODE = 0, main-cursor increase or decrease operation will be mainly decided by main-cursor update command from remote link partner.

To enforce some tap limits, register LT_CONTROL_CFG1.CFG_AP_LMT_MODE is defined to check some limitation and it will only take effect when LT_CONTROL_CFG1.CFG_AP_MAIN_MODE=0. It can work independently with LT_CONTROL_CFG1.CFG_AP_VPK_MODE.

If CFG_AP_LMT_MODE = 1, these limitation will be checked:

- Register LT_CONTROL_CFG2.CFG_LIMIT_1 value will be used as maximum value to limit the sum of pre-cursor, post-cursor and main-cursor absolute value; For initial state, |pre-cursor| + |post-cursor| + |main-cursor| should be set as no more than (LT_CONTROL_CFG2.CFG_LIMIT_1 - 1*step_size). During training phase, if |pre-cursor| + |post-cursor| + |main-cursor| >= LT_CONTROL_CFG2.CFG_LIMIT_1, it means current command is “pre-cursor decrease, post-cursor decrease, main-cursor decrease” or “pre-cursor increase, post-cursor increase, main-cursor increase”. For both of these two cases, |pre-cursor| + |post-cursor| + |main-cursor| only increase 1*step_size. 3 MINIMUM status will be sent to remote link partner as decrease command response. Similarly, 3 MAXIMUM status will be sent as increase command response.



- Register LT_CONTROL_CFG2.CFG_LIMIT_2 value will be used as maximum value to limit the sum of pre-cursor and post-cursor value. For initial state, $|initial\ pre-cursor| + |initial\ post-cursor|$ should be set no more than $(LT_CONTROL_CFG2.CFG_LIMIT_2 - 2 * step_size)$. During training phase, if $|new\ pre-cursor\ value| + |new\ post-cursor\ value| \geq LT_CONTROL_CFG2.CFG_LIMIT_2$, it means current coefficient combinations have reached predefined limitation. The new cursor value will not be accepted and the training algorithm will keep the old value. MINIMUM status will be sent to remote link partner as decrease change failed response and MAXIMUM status will be sent to remote link partner as increase change failed response. In summary, There are some comments for CFG_AP_LMT_MODE:

These two limitations can work at the same time. For more complex cursor command combination application, it is suggested that CFG_AP_LMT_MODE should be set as 0 and disable this function.

CFG_AP_LMT_MODE and CFG_AP_MAIN_MODE should not be used at the same time. When CFG_AP_MAIN_MODE is enabled, CFG_AP_LMT_MODE should be disabled; When CFG_AP_LMT_MODE is enabled, CFG_MAIN_MODE should be disabled.

When CFG_AP_LMT_MODE is enabled, the response will be based on one command. If any limitation is detected, increase command will get MAXIMUM status and decrease command will get MINIMUM status.

When CFG_AP_LMT_MODE is enabled and remote link partner send commands to change pre-cursor, post-cursor and main-cursor at the same time. Pre-cursor command will be processed first, then post-cursor command and main-cursor command.

When LT_CONTROL_CFG2.CFG_LIMIT_1 is checked, $|pre-cursor| + |post-cursor| + |main-cursor|$ should not be changed more than $1 * step_size$ in one command from the link partner. There are 2 examples to explain this exception: When both pre-cursor command and post-cursor command are increase commands while main-cursor command is decrease or hold command, the responses of post-cursor and main-cursor maybe incorrect ;

When both pre-cursor command and post-cursor command are decrease commands while main-cursor command is increase or hold command, the responses of post-cursor and main-cursor maybe incorrect.

When increase or decrease request is processed upon SerDes transmitter output waveform amplitude. Three configuration registers are defined to invert related requests:

LT_CONTROL_CFG1.CFG_TRAIN_INV_POST = 1, increase request will increase the related waveform amplitude while decrease request will decrease the related waveform amplitude;

LT_CONTROL_CFG1.CFG_TRAIN_INV_POST = 0, increase request will decrease the related waveform amplitude while decrease request will increase the related waveform amplitude.

Also see similar register configuration LT_CONTROL_CFG1.CFG_TRAIN_INV_PRE and LT_CONTROL_CFG1.CFG_TRAIN_INV_MAIN.

When LT_CONTROL_CFG1.CFG_AP_MAIN_MODE = 1 or
LT_CONTROL_CFG1.CFG_AP_LMT_MODE = 1,
LT_CONTROL_CFG1.CFG_TRAIN_INV_POST should be set as 0.

When LT_CONTROL_CFG1.CFG_AP_MAIN_MODE = 1 or
LT_CONTROL_CFG1.CFG_AP_LMT_MODE = 1,
LT_CONTROL_CFG1.CFG_TRAIN_INV_PRE should be set as 0.

When LT_CONTROL_CFG1.CFG_AP_MAIN_MODE = 1 or
LT_CONTROL_CFG1.CFG_AP_LMT_MODE = 1,
LT_CONTROL_CFG1.CFG_TRAIN_INV_MAIN should be set as 1.

Cursor tap increase or decrease steps for every command is configurable. It can be 1, 2, 3 and 4 steps by configuring LT_CONTROL_CFG2.CFG_INC_STEP.

3.7.4 Coefficient searching algorithm

When we update coefficients of 3 or 4 Taps FIR filter in the remote link partner, we can measure and get different signal quality at local SerDes receiver. The size of eye margin and any other possible metrics can be used to evaluate the signal quality. That means we can measure all eye margins or any other possible metrics then use one digital value to represent the signal quality at every possible coefficients combination value then we will choose the maximum one. Coefficient searching algorithm helps to implement the whole searching process.

Note: How to evaluate the eye margin or any other possible metric and get one value, it mainly depends on the SerDes requirements, please refers to LT AS document or related SerDes document for more detailed information to implement it.

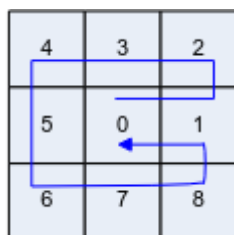
Generally in theory, exploring all possible coefficient areas then we can get the global optimum value finally. However, it's very hard in real world. IEEE 802.3 protocol has defined 500ms limitation for the total link training time. That means we need some special searching algorithm to find a local optimum and may not get the global optimum but the local optimum.

Two coefficient searching algorithms are to be implemented.

3.7.4.1 3*3 grid searching algorithm

An initial 3*3 grid of eye-margin measurements is defined as the following figure:

图 3-11 Initial eye-margin 3*3 grid and measuring path



步骤1 Initial eye-margin 3*3 grid measurement operation.



The eye-margin measuring sequence path for the grid is (Pos0 represents position 0 and it's the initial state):

Pos0→Pos1→Pos2→Pos3→Pos4→Pos5→Pos6→Pos7→Pos8→Pos1→Pos0

When moving on the horizontal direction needs to change pre-cursor value relatively. When move to the right direction, the pre-cursor will increase. When move to the left direction, the pre-cursor will decrease.

When moving on the vertical direction needs to change post-cursor value relatively. When move to the up direction, the post-cursor will increase. When move to the down direction, the post-cursor will decrease.

There are two options for current cursor value changing. One option is holding initial current cursor value during search process. Another option is how to change current cursor coefficients depends on how post-cursor and pre-cursor changes. The rule is shown in the following table:

表 3-7 Pre-cursor, post-cursor and cur-cursor changing for every position in 3*3 grid

Position Number	Option 1: Changing cur-cursor coefficients referring to pre-cursor and post-cursor			Option 2: Holding cur-cursor coefficients		
	Pre-cursor	Post-cursor	Cur-cursor	Pre-cursor	Post-cursor	Cur-cursor
Pos0	initial	initial	initial	initial	initial	initial
Pos1	+1	hold	+1	+1	hold	hold
Pos2	+1	+1	+2	+1	+1	hold
Pos3	hold	+1	+1	hold	+1	hold
Pos4	-1	+1	hold	-1	+1	hold
Pos5	-1	hold	-1	-1	hold	hold
Pos6	-1	-1	-2	-1	-1	hold
Pos7	hold	-1	-1	hold	-1	hold
Pos8	+1	-1	hold	+1	-1	hold

Once the initial grid measurement operation is finished, the pre-cursor, cur-cursor and post- cursor will return to the Pos0 value.

Attention: When return to the Pos0, pre-cursor , post-cursor and cur-cursor value should not be changed . They must be same as the value before the measurement operation.

After measurement operation, we can get a new 3*3 eye-margin value metric and we can know which one in the 9 possible positions has the maximum value. If there are two or more positions have the same maximum value, the smaller position number should be selected as the maximum.



Some information will be recorded same as maximum value:

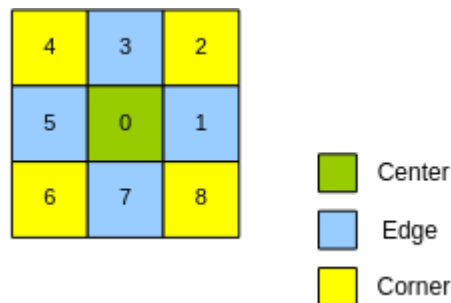
1. Position 0 is recorded to label the whole grid position. For example, initial grid position is recorded as (0, 0). If the new grid position is (1, 1), that means the next moving direction is northeast.
2. Position of the maximum value should be recorded.

步骤2 Estimate the next position to move.

The maximum value can be in 3 possible positions shown in the following figure:

- Center (Position 0)
- Edge (Position 1, position 3, position 5, position 7)
- Corner (Position 2, position 4, position 6, position 8)

图 3-12 Center, edge and corner



Some rules are defined for searching the next search window:

Rule 1: The new search window must include the current position that has the maximum value. That means a bit-logic metric is generated as the followings:

$$\begin{pmatrix} A4 & A3 & A2 \\ A5 & A0 & A1 \\ A6 & A7 & A8 \end{pmatrix} = \begin{pmatrix} \text{northwest is ok for max} & \text{north is ok for max} & \text{northeast is ok for max} \\ \text{west is ok for max} & 0 & \text{east is ok for max} \\ \text{south west is ok for max} & \text{south is ok for max} & \text{southeast is ok for max} \end{pmatrix}$$

1 means the possible direction can be selected based on rule 1.

0 means the possible direction is forbidden.

There are 3 examples to explain the bit-logic metric:

Example 1: If current position 0 (Center) has the maximum value, then the bit-logic metric is:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

There are 8 possible directions to move: east, northeast, north, northwest, west, southwest, south and south east.

Example 2: If current position 1 (Edge) has the maximum value, then the bit-logic metric is:



$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

There are 5 possible directions to move: east, northeast, north, south and south east.

Example 3: If current position 2 (Corner) has the maximum value, then the bit-logic metric is:

$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

There are 3 possible directions to move: east, northeast and north.

Rule 2: The new search window should not exceed the maximum and minimum coefficients limitation. That means a bit-logic metric is generated as the followings if cur-cursor keeps holding:

$$\begin{pmatrix} B4 & B3 & B2 \\ B5 & B0 & B1 \\ B6 & B7 & B8 \end{pmatrix} = \begin{pmatrix} \text{premin \& postmax not hit} & \text{postmax not hit} & \text{premax \& postmax not hit} \\ \text{premin not hit} & 0 & \text{premax not hit} \\ \text{premin \& postmin not hit} & \text{postmin not hit} & \text{premax \& postmin not hit} \end{pmatrix}$$

We should also consider cur-cursor maximum and minimum value limitation if we need to change cur-cursor at the same time. Then the metric will be changed to a more complex logic:

$$\begin{pmatrix} \text{premin \& postmax not hit} & \text{postmax \& curmax not hit} & \text{premax \& postmax \& cur max not hit} \\ \text{premin \& curmin not hit} & 0 & \text{premax \& cur max not hit} \\ \text{premin \& postmin \& cur min not hit} & \text{postmin \& curmin not hit} & \text{premax \& postmin not hit} \end{pmatrix}$$

1 means the possible direction can be selected based on rule 2. 0 means the possible direction is forbidden

Rule 3: The possible move direction should be estimated by current 3*3 metric value .

Then a new estimation metric is calculated as the followings:

$$\begin{pmatrix} C4 & C3 & C2 \\ C5 & C0 & C1 \\ C6 & C7 & C8 \end{pmatrix} = \begin{pmatrix} \text{sum(pos1,pos7,pos8)} & \text{sum(pos6,pos7,pos8)} & \text{sum(pos5,pos6,pos7)} \\ \text{sum(Pos1,Pos2,pos8)} & 0 & \text{sum(pos4,pos5,pos6)} \\ \text{sum(pos1,pos2,pos3)} & \text{sun(pos2,pos3,pos4)} & \text{sum(pos3,pos4,pos5)} \end{pmatrix}$$

For example, if sum (pos1,pos7,pos8) is the smallest one of the 8 summing up values. Then northwest direction has the highest priority to select. If both of the bit logic value in rule 1 and the bit logic in rule2 are 1 value.

The final decision metric can be calculated as the followings:

$$\begin{pmatrix} D4 & D3 & D2 \\ D5 & D0 & D1 \\ D6 & D7 & D8 \end{pmatrix} = \begin{pmatrix} (A4\&B4 == 1)?C4:MAX_VALUE & (A3\&B3 == 1)?C3:MAX_VALUE & (A2\&B2 == 1)?C2:MAX_VALUE \\ (A5\&B5 == 1)?C5:MAX_VALUE & MAX_VALUE & (A1\&B1 == 1)?C1:MAX_VALUE \\ (A6\&B6 == 1)?C6:MAX_VALUE & (A7\&B7 == 1)?C7:MAX_VALUE & (A8\&B8 == 1)?C8:MAX_VALUE \end{pmatrix}$$



Then the position of minimum value selected from D1 to D8 is decided as the next moving position. MAX_VALUE means the direction is invalid.

表 3-8 Move direction based on D metric

Minimum Value	Move Direction
D1	East
D2	Northeast
D3	North
D4	Northwest
D5	West
D6	Southwest
D7	South
D8	Southeast

Once the move direction is decided, the position of the new center point is decided.

步骤3 At the new position execute the new measuring operation.

The window moving and new measuring operation will be done at the same time.

If the old window position 1 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 5 (Old Pos 0) -> New Pos 6 -> New Pos 7 -> New Pos 8 (MO) -> New Pos 1 (MO) -> New Pos 2 (MO) -> New Pos 3 -> New Pos 4 -> New Pos 5 -> New Pos 6 -> New Pos 7 -> New Pos 8 -> New Pos 1 -> New Pos 0.

If the old window position 2 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 6 (Old Pos 0) -> New Pos 7 -> New Pos 8 (MO) -> New Pos 1 (MO) -> New Pos 2 (MO) -> New Pos 3 (MO) -> New Pos 4(MO) -> New Pos 5 -> New Pos 6 -> New Pos 7 -> New Pos 8 -> New Pos 1 -> New Pos 0.

If the old window position 3 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 7 (Old Pos 0) -> New Pos 8 -> New Pos 1 -> New Pos 2 (MO) -> New Pos 3 (MO) -> New Pos 4(MO) -> New Pos 5 -> New Pos 6 -> New Pos 7 -> New Pos 8 -> New Pos 1 -> New Pos 0.

If the old window position 4 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 8 (Old Pos 0) -> New Pos 1 -> New Pos 2 (MO) -> New Pos 3 (MO) -> New Pos 4(MO) -> New Pos 5 (MO) -> New Pos 6 (MO) -> New Pos 7 -> New Pos 8 -> New Pos 1 -> New Pos 0.



If the old window position 5 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 1 (Old Pos 0) -> New Pos 2 -> New Pos 3-> New Pos 4(MO) -> New Pos 5 (MO) -> New Pos 6 (MO) -> New Pos 7 -> New Pos 8 -> New Pos 1 -> New Pos 0.

If the old window position 6 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 2 (Old Pos 0) -> New Pos 3-> New Pos 4(MO) -> New Pos 5 (MO) -> New Pos 6 (MO) -> New Pos 7(MO) -> New Pos 8 (MO) -> New Pos 1 -> New Pos 0.

If the old window position 7 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 3 (Old Pos 0) -> New Pos 4 -> New Pos 5 -> New Pos 6 (MO) -> New Pos 7(MO) -> New Pos 8 (MO) -> New Pos 1 -> New Pos 0.

If the old window position 8 is selected as the new window position 0, then the position moving sequence will be (MO means needs measuring operation):

New Pos 4 (Old Pos 0) -> New Pos 5 -> New Pos 6 (MO) -> New Pos 7(MO) -> New Pos 8 (MO) -> New Pos 1(MO) -> New Pos 2 (MO) -> New Pos 3-> New Pos 4 -> New Pos 5 -> New Pos 6 -> New Pos 7 -> New Pos 8 -> New Pos 1 -> New Pos 0.

Most of the work is same as step 1 and step 2 but there is still something different from step 1 and step 2:

- After eye-margin measurement if there are multiple positions have the same maximum value but it is same as the old position, old position must be labeled as new position of maximum value.

步骤4 Decide whether continue or terminate searching process.

There are several conditions that can terminate the searching process:

- The searching time has exceeded limited time. A timer (register LT_HARD_TRAIN_CFG2.TRAIN_TERMINATE_TIMER is defined to measure the whole training time. If it exceeds pre-defined value, next searching operation will be terminated and current maximum value will be the final decision.
- The searching window has explored enough times. If it exceeds pr-defined times next searching operation will be terminated and current maximum value will be the final decision.
- There is no any available position in {D1, D2, D3, D4, D5, D6, D7, D8} .
- The same search window will be searched twice. For the worst case the latest maximum value will explore all 9 positions in the search window. That means it is the biggest value in a 5*5 metric.

If no any terminate condition is met, just jump to step 2 for a new search operation.

Once the terminate condition is met, the point that has the maximum value in the last searching window will be selected. One special condition is that there is no any available position in {D1,D2,D3,D4,D5,D6,D7,D8} but the final maximum point can still be selected in {Pos1,Pos2,Pos3,Pos4,Pos5,Pos6,Pos7,Pos8}.



Once the final maximum point is decided, The way of moving to the final maximum point is :

D1: Pos0 -> Pos1

D2: Pos0-> Pos1 -> Pos2

D3: Pos0-> Pos3

D4: Pos0-> Pos3 -> Pos4

D5: Pos0->Pos5

D6: Pos0->Pos5->Pos6

D7: Pos0-> Pos7

D8: Pos0 -> Pos7-> Pos8

----结束

3.8 Software Link Training Process

When work in software training mode, the hardware will only perform:

- Training frame assembling and transmitting.
- Training frame receiving and field analyzing.

That means software should do all of the rest things to finish the complete training process.

Software will do these things to finish software training mode:

- Initialize weight value of the taps in FIR.
- Fill training frame coefficient update field.
- Read back status report field received from remote link partner.

Firstly, Software can set register bit LT_CONTROL_CFG1.CFG_TRAINING_ENABLE as logic 0 to disable the training enable control bit and set related training mode register bitLT_CONTROL_CFG1.CFG_TRAINING_MODE as logic 1 to enter software training mode.

3.9 Control Processor Access

Please see “HIMACV320 400GE Combo FS” chapter 1.1.28 Control Processor Access.

3.10 Error Handling

3.10.1 Exceptions for eye diagram measurement

Sometimes the eye diagram measurement result received from SerDes is not correct. For example, the measurement result is zero or too big and it doesn't show



the real signal quality. Register LT_HARD_TRAIN_CFG1.TRAIN_EYE_MARGIN_MAX[14:0] is used to define the maximum value of training eye margin that is correct. A value over this field will be looked as invalid value. Once the exception condition happens, some actions will be done:

- Increase eye diagram exception counter. Field MARGIN_ZERO_CNT or MARGIN_MAX_CNT in register LT_DBG_INFO5 will increase one if get zero or too big value over LT_HARD_TRAIN_CFG1.TRAIN_EYE_MARGIN_MAX[14:0].
- Retry the eye diagram measurement. Please note that there is only one retry chance for every eye diagram measurement operation.

3.10.2 Exceptions for received training frame unlocked

Received data errors will not impact the training state machine directly and these errors are checked by bad marker counter. Only enough continuous data errors can increase bad marker counter until frame lock state is changed to unlocked. Once the received training frame is unlocked, then the training state machine will jump to SEND_TRAINING state.

Specially for received training frame unlocked and eye diagram measurement error occur at the same time, eye diagram measurement operation should hold until received training frame is locked again.

3.10.3 Exceptions for SerDes SPI interface access

There are two time-out configurations that can process SerDes SPI interface access exception.

A timer named LT_SPI_CFG.CFG_COEFF_UPD_TIMER is defined to set the maximum polling times to check the tap update response from SerDes. Only when register LT_SPI_CFG.CFG_COEFF_UPD_TIMER_ENB is asserted, the timer is enabled. When the timer has reached the defined value and timed out, the coefficient update access will be terminated. If register LT_SPI_CFG.CFG_COEFF_UPD_TIMER_ENB is not asserted, the coefficient update access will polling until the tap update response from SerDes is received.

A timer named LT_SPI_CFG.CFG_EYE_SCAN_TIMER is defined to set the maximum polling times to check the eye diagram scan response from SerDes. Only when register LT_SPI_CFG.CFG_EYE_SCAN_TIMER_ENB is asserted, the timer is enabled. When the timer has reached the defined value and timed out, the eye diagram scan access will be terminated. If register LT_SPI_CFG.CFG_EYE_SCAN_TIMER_ENB is not asserted, the eye diagram scan access will keep polling until the eye diagram scan response from SerDes is received.

3.11 Initialization

3.11.1 NRZ mode

Here we use hardware training as a simple example here to explain how to finish initialization. Software training is same as hardware training on initialization



process except some registers need to configured in different value, please refer to “software link training process” .

The initialization process is:

Step 1: Wait until system power on reset or any other global hardware / software reset finished. After reset, link training enable function is disabled by default.

Step 2: Assert LT function reset and configure needed registers. For more detailed description, please refer to LT module register document.

Step 3: Wait until local SerDes has been started up. If speed auto-negotiation function is enabled, then execute auto-negotiation. SerDes speed has been adjusted and SerDes receive side begin to receive training frame. Step 4: Set LT_CONTROL_CFG1.CFG_TRAINING_ENABLE and LT_CONTROL_CFG1.TRAIN_RESTART as logic 1 to enabled state, trigger the hardware training progress or start software training progress directly. There is a configuration bit LT_CONTROL_CFG1.CFG_TRAINING_MODE to control it's hardware training mode or software training mode .For default setting, it's hardware training mode.

During hardware training process, there are still special interactions between software:

Software needs to start HiLink CTLE self-adaption after training search state machine has initialized remote link partner's FIR filter by sending preset or initial command. Register bit LT_CONTROL_CFG2.TRAIN_PRESET_INITIAL_SEL is defined to decide preset command or initial command will be used. So training search state machine should report whether the training search state machine initialization is finished or not. LT_INIT_STATUS.TRAIN_INIT_DONE is defined to notify software that the training search state machine initialization has finished.

After that Software need to check related status and start HiLink CTLE self-adaption process. During CTLE self-adaption process, training search state machine do nothing until software write a register LT_CONTROL_CFG2.FROZEN_INI_EN or LT_CONTROL_CFG2.FORZEN_IDLE_EN as logic 0 to notify LT module the CTLE self-adaption process is finished.

During hardware training process, if training enable is not asserted, hardware training state machine will only execute initial or preset command, then the local training state machine will jump to rx_ready state. This is only to ensure that remote link partner has finished initialization process.

3.12 Debug Features

Some registers are defined for debugging:

LT_DME_STATUS: LT training and DME status register.

LT_CUR_EYE_MARGIN_STATUS: LT current eye margin measurement status register.

LT_RMT_STATUS: LT remote link partner status register.

LT_CUR_TAP_WEIGHT_STATUS: LT current tap weight status register.

LT_TXEQ_LIMIT_STATUS: LT TXEQ limitations status register.

LT_STATUS: LT local device status register.



LT_DME_INFO: LT DME debug information register.

LT_FRAME_LOCK_FSM_INFO: LT frame lock FSM state machine history record register.

LT_TX_CMD_INFO: LT transmitted command history record register.

LT_DME_CNT: LT DME error counter.

LT_SM_DURATION0_CNT: LT remote and local training duration counter.

LT_SM_DURATION1_CNT: LT total training timer and send training frame timer.

LT_SM_TRANSFER_CNT: LT state machine transition counter.

LT_DBG_INFO0 ~ LT_DBG_INFO6: LT debug information register No.0 to No.6.



4 HiMACV110 XXVGE 初始化

309a例化8个HiMACV110 XXVGE，前6个为X4版本，后2个为X1版本。一个X4版本对应4条LANE，共用一个core时钟；一个X1版本对应只有1条LANE，使用独立的core时钟。

初始化流程如下描述：

具体配置可以参考《HiMACV110 基础寄存器配置方案》，请联系FAE获取。

1. Make sure SerDes initialization is completed.
2. Make sure XXVGE has the correct external reference clock (i_clk_xxvge_cfg, i_clk_xxvge_ptp, i_clk_xxvge_core, i_clk_xxvge_rx_sds{0..7} and i_clk_xxvge_tx_sds{0..7}).
3. Assert all of the reset (i_rstn_xxvge_t, i_rstn_xxvge_cfg{0..7}, i_rstn_xxvge_core{0..7}).

图 4-1 SC_LGE_RESET_REQ Register

SC_LGE_RESET_REQ	LGE软复位请求控制寄存器	5A30	0x0A30	32		0x0000FFFF	reserved	31:16	-	0x0000	保留。
							rst_req_lge_t	15:8	MSB0_SET_WO_WF	0xFF	软复位请求 0：无效； 1：软复位有效。

图 4-2 SC_LGE_CFG_RESET_REQ Register

SC_LGE_CFG_RESET_REQ	LGE CFG软复位请求控制寄存器	5A38	0x0A38	32		0x03FFFFFF	reserved	31:26	-	0x00	保留。
							rst_req_lge_cfg	25:0	MSB0_SET_WO_WF	0x3FFFFFFF	软复位请求 0：无效； 1：软复位有效。

图 4-3 SC_LGE_CORE_RESET_REQ Register

SC_LGE_CORE_RESET_REQ	LGE CORE软复位请求控制寄存器	5A40	0x0A40	32		0x03FFFFFF	reserved	31:26	-	0x00	保留。
							rst_req_lge_core	25:0	MSB0_SET_WO_WF	0x3FFFFFFF	软复位请求 0：无效； 1：软复位有效。

4. De-assert reset (i_rstn_xxvge_t), and resets make effect at least 1μs(for EDA:256 *UI ns).

图 4-4 LGE 软复位去请求控制寄存器

SC_LGE_RESET_DREQ	LGE软复位去请求控制寄存器		0x0A34	32		0x00000000	reserved	31:16	-	0x0000	保留。
							rst_dreq_lge_t	15:8	MRREG_CLR_WO	0x00	撤销软复位请求 0：无效； 1：撤销软复位请求。



5. Set i_regs_xxvge_clk_gate_en to high.

图 4-5 GE 时钟使能寄存器

寄存器名称	寄存器地址	寄存器大小	寄存器类型	寄存器值	寄存器描述
reg_clk_en	0x00000000	32	write-only	0x00000000	保留
reg_train_clk_gate_en	0x00000004	32	write-only	0x00000000	训练时钟使能寄存器。当该寄存器为1时，训练时钟使能。该寄存器为0时，训练时钟使能关闭。
reg_tx_clk_gate_en	0x00000008	32	write-only	0x00000000	发送时钟使能寄存器。当该寄存器为1时，发送时钟使能。该寄存器为0时，发送时钟使能关闭。
reg_rx_clk_gate_en	0x0000000C	32	write-only	0x00000000	接收时钟使能寄存器。当该寄存器为1时，接收时钟使能。该寄存器为0时，接收时钟使能关闭。
reg_mac_clk_gate_en	0x00000010	32	write-only	0x00000000	MAC时钟使能寄存器。当该寄存器为1时，MAC时钟使能。该寄存器为0时，MAC时钟使能关闭。
reg_mac_clk_gate_en	0x00000014	32	write-only	0x00000000	MAC时钟使能寄存器。当该寄存器为1时，MAC时钟使能。该寄存器为0时，MAC时钟使能关闭。
reg_mac_clk_gate_en	0x00000018	32	write-only	0x00000000	MAC时钟使能寄存器。当该寄存器为1时，MAC时钟使能。该寄存器为0时，MAC时钟使能关闭。
reg_mac_clk_gate_en	0x0000001C	32	write-only	0x00000000	MAC时钟使能寄存器。当该寄存器为1时，MAC时钟使能。该寄存器为0时，MAC时钟使能关闭。

参见《Hi309a寄存器手册》；

6. Set i_regs_xxvge_clk_gate_en to low.
7. De-assert reset (i_rstn_xxvge_cfg{0..7}).
 - a. i_rstn_xxvge_cfg{0..7} must be de-assert for high grade(TOP 8 PORT).
 - b. i_rstn_xxvge_cfg{0..3} must be de-assert for middle grade(TOP 4 PORT).
 - c. i_rstn_xxvge_cfg0 must be de-assert for lite grade(TOP 1 PORT).

图 4-6 SC_LGE_CFG_RESET_DREQ Register

寄存器名称	寄存器地址	寄存器大小	寄存器类型	寄存器值	寄存器描述
SC_LGE_CFG_RESET_DREQ	0x00000000	32	write-only	0x00000000	保留
rst_dreq_cfg	0x00000004	32	write-only	0x00000000	复位请求配置寄存器。当该寄存器为1时，复位请求使能。该寄存器为0时，复位请求使能关闭。

8. XXVGE_CLK_ENABLE register must be all 0(NOT GATE), it could be set to 1 if this module not use after finish initialization.
9. Configure XXVGE_WORK_MODE, XXVGE_PORT_MODE, XXVGE_PORT_SPEED registers for work mode and speed of used port.
10. Configure XXVGE_WORK_FREQ_L, XXVGE_WORK_FREQ_H registers based on the frequency fed to i_clk_xxvge_core.
11. Configure XXVGE_MAC_TX_BASE_FREQ_L, XXVGE_MAC_TX_BASE_FREQ_H registers based on the work mode and speed for different port individually.
12. Configure XXVGE_TX_CALEN_DEPTH for calendar table at transmit side.
 - a. Configure XXVGE_TX_CALEN_DEPTH is 8 for high grade(TOP 8 PORT).
 - b. Configure XXVGE_TX_CALEN_DEPTH is 4 for middle grade(TOP 4 PORT).
 - c. Configure XXVGE_TX_CALEN_DEPTH is 4 for lite grade(TOP 1 PORT).
13. Configure XXVGE_TX_CALENDAR_TABLE for the content of calendar table at transmit side. Refer to the section “MAC Transmit” of FS for the specified value.
14. Configure XXVGE_PMA_CALENDAR_TABLE for the content of calendar table at receive side. Refer to the section “PMA” of FS for the specified value.
15. Configure XXVGE_RXPMA_CONTROL.PARA_DATA_WIDTH for the used port.
16. Configure XXVGE_RXPMA_CONTROL.RX_BIT_REVERSE and XXVGE_TXPMA_CONTROL.TX_BIT_REVERSE to appropriate value.



17. For 100BASE-FX, reconfigure XXVGE_TXPMA_CONTROL. TX_OS_MODE to value 0b11 to adapt 1.25Gbps SerDes.
18. De-assert reset (i_rstn_xxvge_core{0..7}).

图 4-7 SC_LGE_CORE_RESET_DREQ Register

SC_LGE_CORE_RESET_DREQ	LGE CORE软复位去请求控制寄存器	5A44	32		0x00000000	reserved	31:26	-	0x00	保留。
						srst_dreq_lge_core	25:0	MREG_CLR_WO	0x00000000	复位去请求。 0: 无效。 1: 撤销复位去请求。

19. Check link-up status by read XXVGE_INT_STATUS and XXVGE_LINK_STATUS register. If this status can't be achieved for a long time (100ms), try to check all of the XXVGE's configuration and SerDes is correct, and reset RX or TX (mostly RX) direction by XXVGE_RESET register bit or check configure register value is correct.
20. Clear all the interrupts, status and MIB counter caused by XXVGE initialization procedure after XXVGE interface is ready. Note: it is prohibited to do read operation to registers which not include current work port mode. Eg. It is prohibited to do read operation to BRFECE registers if BRFECE function is disabled.
21. Configure TX transmit enable by XXVGE_TXMAC_ENABLE and RX receive enable by XXVGE_RXMAC_ENABLE register, and after 1μs(for EDA:256 *UI ns), then TX/RX can transmit/receive data (This register supports dynamic configurable).
22. XXVGE initialization is done...

说明

the clk_xxvge_tx_sds* or clk_xxvge_rx_sds* clock must not be shutdown, If the project team want to shut down the clk_xxvge_tx_sds* or clk_xxvge_rx_sds* clock, it must have clock for clk_xxvge_tx_sds* or clk_xxvge_rx_sds* , then reset the according port for at least 100ns, then shut down the clk_xxvge_tx_sds* or clk_xxvge_rx_sds* clock. And must set the according port register PMA_COMMON_XXVGE_PMA_RESET.TXPMA/RXPMA to value 1.

4.1 XSEC 初始化

4.1.1 XSEC ENC

The XSEC_ENC's procedure for initialization is shown as below.

1. Make sure XSEC_ENC has the correct external reference clock (CORE_CLK, T_CLK).
2. Assert the reset (XSEC_ENC_RST_N).

图 4-8 SC_PPE_RESET_REQ3 Register

SC_PPE_RESET_REQ3	PPE软复位去请求控制寄存器	5A18	0x0A28	32		0x0000001F		reserved	31:5	-	0x00000000	保留。
							srst_req_xsec_cfg	4	MREG_SET_WO	0x1		软复位去请求。 0: 无效。 1: 软复位去请求。
							srst_req_xsec_dec	3	MREG_SET_WO	0x1		软复位去请求。 0: 无效。 1: 软复位去请求。
							srst_req_xsec_enc	2	MREG_SET_WO	0x1		软复位去请求。 0: 无效。 1: 软复位去请求。
							srst_req_xsec_common	1	MREG_SET_WO	0x1		软复位去请求。 0: 无效。 1: 软复位去请求。
							srst_req_rtc	0	MREG_SET_WO	0x1		软复位去请求。 0: 无效。 1: 软复位去请求。



3. De-assert (XSEC_ENC_RST_N).

图 4-9 SC_PPE_RESET_DREQ3 Register

SC_PPE_RESET_DREQ3	PPE软复位去请求控制寄存器	0x0A2C	32	0x00000000	reserved	31:5	-	0x00000000	保留。
					srst_dreq_xsec_cfg	4	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
					srst_dreq_xsec_dec	3	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
					srst_dreq_xsec_enc	2	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
					srst_dreq_xsec_common	1	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
					srst_dreq_rtc	0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。

4. Wait at least 12 T_CLK cycles.

Step 1~3 please refer to Reset and Initialization section in the host chip FS document for details of clock and reset.

- Start to initial the configurable memories by XSEC_ENC_INIT_START, and wait the procession to be done by inquiring the status register XSEC_ENC_INIT_STATUS. The configurable tables (TX_PROFILE_TBL, TX_SC_CFG_TBL, TX_SA_CFG_TBL, TX_NEXTPN_TBL and TX_SAK_TBL) should be initialized to all zeros. The TX_IADP_FREE_LIST_RAM should be initialized from 0 to maximum address of TX_IADP_DATA_BUF_MEM.
- Configure base profile table XSEC_ENC_BASE_PROFILE{*} for the using base profile ID.
- Configure profile table XSEC_ENC_PROFILE_TABLE_ENTRY{*}_WORD{*} for the using profile ID.
- Configure the FIFO size and backpressure threshold through registers XSEC_ENC_{IADP,OADP}_PORT_BUF_CFG{*}, XSEC_ENC_{IADP,OADP}_TOTAL_BUF_CFG, XSEC_ENC_PORT_CREDIT_CURRENT_CNT{*}.
- Configure the calendar scheduler through registers XSEC_ENC_CALENDAR_SLOT_LENGTH, XSEC_ENC_CALENDAR_TIME_SLOT{*}, XSEC_ENC_IADP_CALENDAR_LOAD_EN and XSEC_ENC_OADP_CALENDAR_LOAD_EN.
- Configure XSEC_ENC_MPLS_LABEL if there is MPLS packet encryption business.
- Configure Transmit SC/SA/NextPN/SAK configuration Table for the using SC and SA. The configuration registers are below:
 - XSEC_ENC_SC_CFG_TABLE_ENTRY{*}_WORD{*}
 - XSEC_ENC_SA_CFG_TABLE_ENTRY{*}_WORD{*}
 - XSEC_ENC_NEXTPN_TABLE_ENTRY{*}_WORD{*}
 - XSEC_ENC_SAK_TABLE_ENTRY{*}_WORD{*} and XSEC_ENC_SAK_ENTRY_VLD{*}
- Clear all the interrupts, status and counter caused by XSEC_ENC initialization procedure.
XSEC_ENC initialization has been done



4.1.2 XSEC DEC

The XSEC_DEC's procedure for initialization is shown as below.

1. Make sure XSEC_DEC has the correct external reference clock (CORE_CLK, T_CLK).
2. Assert the reset (XSEC_DEC_RST_N).

图 4-10 SC_PPE_RESET_REQ3 Register

SC_PPE_RESET_REQ3	PPE 配置位需求控制寄存器	SA18	0x0A2B	32		0x0000001F	reserved	31:5	-	0x00000000	保留。
							srst_req_xsec_cfg	4	MREG_SET_WO	0x1	配置位需求 0: 无效; 1: 配置位有效。
							srst_req_xsec_dec	3	MREG_SET_WO	0x1	配置位需求 0: 无效; 1: 配置位有效。
							srst_req_xsec_enc	2	MREG_SET_WO	0x1	配置位需求 0: 无效; 1: 配置位有效。
							srst_req_xsec_common	1	MREG_SET_WO	0x1	配置位需求 0: 无效; 1: 配置位有效。
							srst_req_dec	0	MREG_SET_WO	0x1	配置位需求 0: 无效; 1: 配置位有效。

3. De-assert (XSEC_DEC_RST_N).

图 4-11 SC_PPE_RESET_DREQ3 Register

SC_PPE_RESET_DREQ3	PPE 配置位需求控制寄存器	0x0A2C	32		0x00000000	reserved	31:5	-	0x00000000	保留。
						srst_dreq_xsec_cfg	4	MREG_CLR_WO	0x0	清除配置位需求 0: 无效; 1: 清除配置位需求。
						srst_dreq_xsec_dec	3	MREG_CLR_WO	0x0	清除配置位需求 0: 无效; 1: 清除配置位需求。
						srst_dreq_xsec_enc	2	MREG_CLR_WO	0x0	清除配置位需求 0: 无效; 1: 清除配置位需求。
						srst_dreq_xsec_common	1	MREG_CLR_WO	0x0	清除配置位需求 0: 无效; 1: 清除配置位需求。
						srst_dreq_dec	0	MREG_CLR_WO	0x0	清除配置位需求 0: 无效; 1: 清除配置位需求。

4. Wait at least 12 T_CLK cycles.

Step 1~3 please refer to Reset and Initialization section in the host chip FS document for details of clock and reset.

5. Start to initial the configurable memories by XSEC_DEC_INIT_START, and wait the procession to be done by inquiring the status register XSEC_DEC_INIT_STATUS. The configurable tables (RX_PROFILE_TBL, RX_SA_CFG_TBL, RX_NEXTPN_TBL, and RX_SAK_TBL) and RX_UPDTPN_TBL(a copy table of RX_NEXTPN_TBL) should be initialized to all zeros. The RX_IADP_FREE_LIST_RAM should be initialized from 0 to maximum address of RX_IADP_DATA_BUF_MEM.
6. Configure the XSEC_DEC_TRUSTABLE_PKT_MASK if the corresponding packet requires to mark a trustable flag.
7. Configure the XSEC_DEC_PORT_CTRL{*}.UNKOWNSCI_BYPASS, set it to 1 if necessary.
8. Configure the XSEC_DEC_PORT_CTRL{*}.DEFAULT_SC if the port only uses one SC and IncludeSCIEnable of the SC is set to 0.
9. Configure base profile table XSEC_DEC_BASE_PROFILE{*} for the using base profile ID.
10. Configure profile table XSEC_DEC_PROFILE_TABLE_ENTRY{*}_WORD{*} for the using profile ID.



11. Configure the FIFO size and backpressure threshold through registers
XSEC_DEC_{IADP,OADP}_PORT_BUF_CFG{*},
XSEC_DEC_{IADP,OADP}_TOTAL_BUF_CFG.
12. Configure the output calendar scheduler through registers
XSEC_DEC_CALENDAR_SLOT_LENGTH, XSEC_DEC_CALENDAR_TIME_SLOT{*},
XSEC_DEC_IADP_CALENDAR_LOAD_EN and
XSEC_DEC_OADP_CALENDAR_LOAD_EN.
13. Configure XSEC_DEC_PORT{*}_CTRL register:
Configure field VALIDATE_EN to decide how to process packets without
SecTAG or receive SC is not found.
14. Configure XSEC_DEC_SCI_VLD{*} and XSEC_DEC_SCI{*}_WORD{*} registres
used for receive SC lookup.
15. Configure Receive SC/SA/NextPN/SAK configuration Table. The detail
configuration is below:
 - a. XSEC_DEC_SC_CFG_ENTRY{*}
 - b. XSEC_DEC_SA_CFG_TABLE_ENTRY{*}_WORD{*}
 - c. XSEC_DEC_NEXTPN_TABLE_ENTRY{*}_WORD{*}
 - d. XSEC_DEC_SAK_TABLE_ENTRY{*}_WORD{*} and
XSEC_DEC_SAK_ENTRY_VLD{*}
16. Clear all the interrupts, status and counter caused by XSEC_DEC initialization
procedure.

XSEC_DEC initialization has been done.

须知

1. 在当前的Hi309a芯片中XSEC的base profile 表项只使用表项0。
2. XSEC在Hi309a芯片中仅采用TDM调度模式，此时XSEC_ENC_OADP_SCH_MODE和XSEC_DEC_OADP_SCH_MODE寄存器可保持其默认设置。
3. 对于超过32bit的XSEC寄存器，必须进行完整寄存器内容的连续读取或写入操作，不支持按word为粒度读或者写。比如XSEC_DEC_BASE_PROFILE寄存器是128bit位宽，需要连续写4个word的数据，或者连续读4个word的数据。不支持写一个word后就读该word对应的值。
4. 超过32bit的XSEC寄存器，数据存放的顺序是NP大端序。比如XSEC_DEC_BASE_PROFILE寄存器是128bit位宽，对应的偏移地址为0x2800，写入0x2800的数据会存放在data[127:96]；写入0x2804的数据会存放在data[95:64]；写入0x2808的数据会存放在data[63:32]；写入0x280c的数据会存放在data[31:0]；对应的，读返回数据也按照这种顺序拼接。
5. 超过32bit的XSEC寄存器的读写操作对数据的拼接转换，在BSP处理。
6. 超过32bit的XSEC寄存器不是随包修改或配置，不影响性能。
7. Hi309a的XSEC最大支持128个时隙，时隙分配计算器中默认使用128个时隙，建议同样使用128个时隙。加密的时隙粒度最小为1G，解密的时隙粒度最小为0.7G。
8. 任意一个port带XSEC了，所有port都需要按照《时隙分配计算器》中来分配。
9. 任意port速率切换时，总的速率不能超过44G。
10. 总的速率不超过44G、总端口数不超过26、总时隙个数不超过128个时，详细组合场景见《时隙分配计算器》中的《解密端口组合》和《加密端口组合》子页，文档请联系FAE获取。



4.2 AN_LT 初始化

4.2.1 AN_TRAIN 复位/解复位

步骤1 确认an_train配置接口复位解复位。详细参见《Hi309a寄存器手册》。

图 4-12 PPE 软复位请求控制寄存器

SC_PPE_RESET_REQ1	PPE软复位请求控制寄存器	SA10	0x0A18	32		0x00000000		srst_req_an_train_command	31:28	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_an_train_cfg	27:24	MREG_SET_WO	0xF	软复位请求 0: 无效;
								srst_req_an_train_spi	23:20	MREG_SET_WO	0xF	软复位请求 0: 无效;
								srst_req_an_rx	19:16	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_an_tx	15:12	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_an_core	11:8	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_train_rx	7:4	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_train_tx	3:0	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。

图 4-13 PPE 软复位去请求控制寄存器

SC_PPE_RESET_REQ1	PPE软复位去请求控制寄存器	SA10	0x0A1C	32		0x00000000		srst_dreq_an_train_command	31:28	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
								srst_dreq_an_train_cfg	27:24	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_train_spi	23:20	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_rx	19:16	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_tx	15:12	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_core	11:8	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_train_rx	7:4	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_train_tx	3:0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;

步骤2 确认an_train SerDes接口复位解复位。

图 4-14 PPE 软复位请求控制寄存器

SC_PPE_RESET_REQ1	PPE软复位请求控制寄存器	SA10	0x0A18	32		0x00000000		srst_req_an_train_command	31:28	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_an_train_cfg	27:24	MREG_SET_WO	0xF	软复位请求 0: 无效;
								srst_req_an_train_spi	23:20	MREG_SET_WO	0xF	软复位请求 0: 无效;
								srst_req_an_rx	19:16	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_an_tx	15:12	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_an_core	11:8	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_train_rx	7:4	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。
								srst_req_train_tx	3:0	MREG_SET_WO	0xF	软复位请求 0: 无效; 1: 软复位有效。

图 4-15 PPE 软复位去请求控制寄存器

SC_PPE_RESET_REQ1	PPE软复位去请求控制寄存器	SA10	0x0A1C	32		0x00000000		srst_dreq_an_train_command	31:28	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
								srst_dreq_an_train_cfg	27:24	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_train_spi	23:20	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_rx	19:16	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_tx	15:12	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_an_core	11:8	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_train_rx	7:4	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;
								srst_dreq_train_tx	3:0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效;

----结束



4.2.2 AN 复位初始化

步骤1 配置AN_CTRL_CFG.ANTONEG_RESET=1复位AN逻辑。

图 4-16 AN_CTRL_CFG

2.2.5 - AN_CTRL_CFG - AN control register

AN control register. User can control AN module by configuring this register.
This register can not be programmed on the fly.

Device	Block Name	Register Name	Start Address	Type	Tags
HiMACV320_AN	AN	AN_CTRL_CFG	0x0000_0009_0010	IOps	1/1

Bits	Field	Access Type	Description
15:7	RESERVED	RO	Reserved.
6:6	TRANSMITTED_NONCE_SEL	RW	Select TRANSMITTED_NONCE source between configuration and logic generation. 1'b0 : From logic generation. 1'b1 : From register AN_XNP_ABILITY0 field TRANSMITTED_NONCE.
5:5	FORCE_NONCE_MISMATCH	RW	Control the result, when comparing Transmitted Nonce Field of received link codeword with local device at ABILITY DETECT state. 1'b0 : the comparison result generated properly. 1'b1 : force the comparison result is mismatch. This is for debug purpose.
4:4	RESERVED	RO	Reserved.
3:3	NEXT_PAGE_LOADED	RWW	Write 1 to indicate a new next page has been loaded into AN XNP Transmit registers(AN_XNP_ABILITY0/1), then cleared by Arbitration state diagram for each entry to NEXT PAFF WAIT state or TRANSMIT DISABLE state. 1'b0 : the loaded page has been transmitted, or need destroy link. 1'b1 : a new next page has been loaded.
2:2	AUTONEG_RESTART	RWW	Write 1 to restart Auto-Negotiation, then cleared by Arbitration state diagram. 1'b0 : auto-Negotiation in process, or disabled. 1'b1 : restart Auto-Negotiation process.
1:1	AUTONEG_RESET	RW	1'b0 : auto-Negotiation normal operation. 1'b1 : auto-Negotiation reset.
0:0	AUTONEG_ENABLE	RW	1'b0 : disable Auto-Negotiation process. 1'b1 : enable Auto-Negotiation process.

步骤2 配置RTC寄存器中对应MAC的AN_INT_MASK.an_complete_int_mask为1。

图 4-17 AN 完成中断使能寄存器

7	AN_INT_MASK	AN完成中断使能	reg	0x0208	0x00000000	reserved	31:3	BY	0x00000000	保留。
6						mac_sync_fail_int_mask	2	BY	0x0	mac 同步失败中断使能
5						mac_sync_ok_int_mask	1	BY	0x0	mac 同步成功中断使能
0						an_complete_int_mask	0	BY	0x0	AN完成中断使能

步骤3 配置AN_INT_ENABLE寄存器。

1. 配置AN_INT_ENABLE.IERR_U=1。
2. 配置AN_INT_ENABLE.AN_UPDATE=1。
3. 配置AN_INT_ENABLE.AN_FAIL=1。
4. 配置AN_INT_ENABLE.AN_SUCCESS=1。

步骤4 配置RTC寄存器中对应MAC的MAC_AN_ENABLE.cfg_an_en为1。

图 4-18 自协商使能和完成配置寄存器

MAC_AN_ENABLE 的中断使能和完成配置 (下列寄存器是每MAC一份, 每个MAC的偏移地址为mac_id*0x1000)										
MAC_AN_ENABLE	自协商使能和完成配置	reg	0x0200	0x00000000	reserved	31:2	BY	0x00000000	保留。	
					cfg_an_en	1	BY	0x0	自协商使能	
					cfg_an_complete	0	BY	0x0	软件通过读取中断得知自协商完成并配置该	

步骤5 配置AN_MODE_CFG寄存器。

1. 配置AN_MODE_CFG.AN_LOSE_THD_ENABLE为0。
2. 从IGU寄存器中读取当前通道的模式MAC_MODE_CFG, 如果为0b011, 则SerDes为10G 模式; 否则为其他模式。如果当前为10G模式, 按照SerDes位宽配置AN_MODE_CFG.PARALLEL_DATA_WIDTH为0x4; 否则配置AN_MODE_CFG.PARALLEL_DATA_WIDTH为0x0。
3. 按照SerDes频点配置AN_MODE_CFG.BIT_DIVISOR, 如果当前为10G模式, 配置AN_MODE_CFG.BIT_DIVISOR为0x042 (默认值), 否则配置AN_MODE_CFG.BIT_DIVISOR为0x8。
4. 配置AN_MODE_CFG.NP_NUMBER=2。
5. 根据SerDes的并行数据大小端配置:
AN_MODE_CFG.RX_BIT_ORDER_REVERSE, AN_MODE_CFG.TX_BIT_ORDER_REVERSE为0x1。



6. 配置AN_CTRL_CFG寄存器,配置AN_CTRL_CFG.AUTONEG_ENABLE=1。
7. 配置AN_TIMER_3_CFG为104。
8. AN_TIMER_UNIT_CFG.UNIT_1MS为0x51615, 与core时钟有关。
9. 读清所有的car寄存器以及写清中断。
10. 根据驱动配置的AN ability配置AN_BP_ABILITY0_CFG和AN_BP_ABILITY1_CFG. (BASE_PAGE)。

步骤6 配置AN_TRAIN_PMA_AN_BP_ABILITY0_CFG.NEXT_PAGE=1（CS 无该配置）。

1. 配置AN_RX_SAMPLE_TOLERANCE_CFG寄存器0x172b。
2. 根据具体需求配置AN_XNP_ABILITY0_CFG寄存器、AN_XNP_ABILITY1_CFG寄存器、AN_XNP_ABILITY2_CFG寄存器和AN_XNP_ABILITY3_CFG寄存器。
3. 配置AN_XNP_ABILITY0_CFG =0xA005。
4. 配置AN_XNP_ABILITY1_CFG =0x4DF0353。
5. 配置AN_XNP_ABILITY2_CFG=0x203。
6. 如果驱动在config_an_ablity命令中配置了50G模式, 配置AN_XNP_ABILITY3_CFG =0xF000330, 否则配置AN_XNP_ABILITY3_CFG =0xF000030。

步骤7 AN_CTRL_CFG.AUTONEG_RESET=0解复位AN。

----结束

4.2.3 AN 操作流程

步骤1 等待AN中断。收到中断进入中断服务程序。

步骤2 如果IERR_U中断, 那么AN出现ECC错误, AN失败, 重启AN; 如果AN_UPDATE中断, 那么继续下一步; 如果有AN_FAIL中断, 那么AN失败, 重启AN; 如果AN_SUCCESS中断, 那么AN成功, 进入到**步骤7**。

步骤3 识别异常跳转路径。

读取AN_STATUS.RENEW_PAGE, 如果值为0, 那么跳到**步骤5**; 否则清除所有软件保存的结果, 继续下一步。

步骤4 发送最初的两页nextpage。

1. 配置最初的两页nextpage到AN_XNP_ABILITY0_CFG、AN_XNP_ABILITY1_CFG、AN_XNP_ABILITY2_CFG和AN_XNP_ABILITY3_CFG。
2. 配置AN_XNP_ABILITY0_CFG=0xA005。
3. 配置AN_XNP_ABILITY1_CFG=0x4DF0353。
4. 配置AN_XNP_ABILITY2_CFG=0x203。
5. 如果驱动在config_an_ablity命令中配置了50G模式, 配置AN_XNP_ABILITY3_CFG =0xF000330, 否则配置AN_XNP_ABILITY3_CFG =0xF000030。
6. 配置AN_CTRL_CFG.NEXT_PAGE_LOADED, 跳到**步骤1**。

步骤5 保存相关信息。

保存对方发送的nextpage信息。读取并保存AN_XNP_ABILITY0_CFG、AN_XNP_ABILITY1_CFG、AN_XNP_ABILITY2_CFG和AN_XNP_ABILITY3_CFG。



步骤6 发送接下来的两页nextpage。

1. 配置接下来的两页nextpage到AN_XNP_ABILITY0_CFG、AN_XNP_ABILITY1_CFG、AN_XNP_ABILITY2_CFG和AN_XNP_ABILITY3_CFG。
2. 配置AN_XNP_ABILITY0_CFG =0x2001。
3. 配置AN_XNP_ABILITY1_CFG =0。
4. 配置AN_XNP_ABILITY2_CFG =0x2001
5. 配置AN_XNP_ABILITY3_CFG =0。
6. 配置AN_CTRL_CFG.NEXT_PAGE_LOADED，跳到**步骤1**。

步骤7 500ms计数器启动。保存相关信息。

1. 读取AN_STATUS.FEC_NEGOTIATED和AN_STATUS.RSFEC_NEGOTIATED（base_page的fec协商结果）。
2. 读取并保存AN_STATUS.PAUSE_NEGOTIATED，得到两端协商出来的pause结果。
3. 读取AN_LINK_CONTROL_STATUS寄存器，确认base_page协商结果。如果协商出来的结果为100G，那么以base page协商出来的速率以及fec启动Training；如果不是这个结果，那么下一步。
4. 读取并保存AN_XNP_ABILITY0_CFG、AN_XNP_ABILITY1_CFG、AN_XNP_ABILITY2_CFG和AN_XNP_ABILITY3_CFG。
5. 判断对方发送的next page是否符合25g/50g联盟的内容；如果符合，那么继续下一步；如果不符合，那么判断硬件是否协商出共同模式，如果硬件有共同模式，根据硬件base page协商出来的速率以及fec完成AN第一步，启动Training；否则AN失败，clock_gate住 AN。
6. 判断联盟的协商出来的速率是不是50g，如果是，那么根据软件算法可以得到速率以及fec能力，完成AN第一步，启动Training；否则，继续下一步。
7. 判断**步骤7.2**读到的协商速率，如果是40g，或者25g，那么以base page协商出来的速率以及fec启动Training；如果不是这个结果，那么下一步。
8. 判断联盟的协商出来的速率是不是25g，如果是，那么根据软件算法可以得到速率以及fec能力，完成AN，启动Training；如果不是，那么下一步。
9. 如果硬件有共同模式，那么以base page协商出来的速率以及fec启动Training；否则AN失败，返回失败，AN时钟门控关闭。

步骤8 跳到硬件training流程或者软件training流程。

步骤9 软件处于等待中断状态。如果接收到AN_FAIL中断，AN失败，配置AN的clock_gate=1；如果在规定时间内完成了link up（或者轮询AN的状态机寄存器AN_STATUS，如果AN_STATUS[30:27]==4'd7），那么AN成功，配置AN的clock_gate=1。

步骤10 AN去使能：

1. 配置RTC寄存器中对应MAC的MAC_AN_ENABLE.cfg_an_en为0。
2. 配置AN_TRAIN_PMA_AN_CTRL.AUTONEG_ENABLE=0。

----结束

4.2.4 Training 复位初始化

步骤1 配置LT_CONTROL_CFG2. RESET_TRAIN_TX =1，LT_CONTROL_CFG2. RESET_TRAIN_RX=1，LT_CONTROL_CFG2. RESET_SPI=1，复位training电路和SPI接口。



步骤2 配置AN_TRAIN_PMA_INT_ENABLE寄存器。

1. 配置LT_INT_ENABLE.TRAIN_INIT_DONE=1。
2. 配置LT_INT_ENABLE.TRAIN_SOFT_FETCH_EYE=1。
3. 配置LT_INT_ENABLE.TRAIN_SOFT_CMD_TRIP_END=1。
4. 配置LT_INT_ENABLE.TRAIN_FAIL=1。
5. 配置LT_INT_ENABLE.TRAIN_SUCCESS=1。

步骤3 配置LT_CONTROL_CFG1寄存器。

1. 配置LT_CONTROL_CFG1.CFG_TX_BIT_ORDER、LT_CONTROL_CFG1.CFG_RX_BIT_ORDER，按SerDes并行数据大小端来给配置值，默认为1。
2. 配置LT_CONTROL_CFG1.CFG_MARKER_DEC=3。
3. 配置LT_CONTROL_CFG1.CFG_MAN_DEC=1。
4. 配置LT_CONTROL_CFG1.CFG_TRAIN_INV_PST、LT_CONTROL_CFG1.CFG_TRAIN_INV_PRE、LT_CONTROL_CFG1.CFG_TRAIN_INV_MAIN，根据SerDes抽头极性决定配置值。默认：PST/PRE为0，MAIN为1。
5. 配置LT_CONTROL_CFG1.CFG_AP_MAIN_MODE，根据对方设备进行配置。
6. 建议HiMACV310之前（不包括HiMACV310）hw老设备，配置为1，否则配为0。
7. 配置LT_CONTROL_CFG1.CFG_VPK_MODE和LT_CONTROL_CFG1.CFG_AP_LMT_MODE，配置均为0。

步骤4 配置LT_CONTROL_CFG2寄存器。

1. 配置LT_CONTROL_CFG2.FROZEN_IDLE_EN，根据具体流程配置（initial流程或者preset流程），如果流程为initial流程，那么其值为1，否则其值为0。注意：此处及以下皆配置为preset流程。
2. 配置LT_CONTROL_CFG2.FROZEN_INI_EN，根据具体流程配置，如果流程为initial流程，那么其值为0，否则其值为1。
3. 配置LT_CONTROL_CFG2.TRAIN_PRESET_INITIAL_SEL，根据具体流程配置，如果流程为initial流程，那么其值为0，否则其值为1。
4. 配置LT_CONTROL_CFG2.CFG_LIMIT_1和LT_CONTROL_CFG2.CFG_LIMIT_2，如果LT_CONTROL_CFG1.CFG_AP_LMT_MODE为0，那么此寄存器不用配置。
5. 配置LT_CONTROL_CFG2.CFG_DATA_WIDTH，根据SerDes位宽配置位0x3。
6. 配置LT_CONTROL_CFG2.REMOTE_RX_READY_CHECK_EN，根据需求配置为0。注：0表示2015年的协议，1表示2007年的协议。默认选择0。
7. 配置LT_CONTROL_CFG2.CFG_INC_STEP=0，根据SerDes TAP值分辨率配置为0。

步骤5 配置LT_SOFT_TRAIN_CFG寄存器。

配置LT_SOFT_TRAIN_CFG.CFG_RECEIVE_READY_CMD=0。

步骤6 配置LT_HARD_TRAIN_CFG1寄存器。

1. 配置LT_HARD_TRAIN_CFG1.INITIAL_OR_PRESET_ONLY，根据具体需求配置。如果只想发送initial/preset后结束training流程，那么配置为1，否则配置为0。
2. 配置LT_HARD_TRAIN_CFG1.TRAIN_CUR_CMD_KEEP=0，配置LT_HARD_TRAIN_CFG1.TRAIN_EYE_MARGIN_MAX，根据hilink返回的最大眼图值配置为128。



步骤7 配置LT_HARD_TRAIN_CFG2寄存器。

1. 配置LT_HARD_TRAIN_CFG2.TRAIN_TERMINATE_EN，根据需求配置，需要控制training时间，配置为1。
2. 配置LT_HARD_TRAIN_CFG2.TRAIN_TERMINATE_TIMER，根据具体情况配置为500ms。

步骤8 配置LT_HARD_TRAIN_CFG3寄存器。

配置LT_HARD_TRAIN_CFG3.TRAIN_ITERATION_EN，配置为0。

步骤9 配置LT_PRE_CURSOR_CFG寄存器。

1. 配置LT_PRE_CURSOR_CFG.CFG_TRAIN_INIT_PRE。
2. 配置LT_PRE_CURSOR_CFG.CFG_TRAIN_PRESET_PRE。
3. 配置LT_PRE_CURSOR_CFG.CFG_TRAIN_MAX_PRE。
4. 配置LT_PRE_CURSOR_CFG.CFG_TRAIN_MIN_PRE。

步骤10 配置LT_POST_CURSOR_CFG寄存器。

1. 配置LT_POST_CURSOR_CFG.CFG_TRAIN_INIT_POST，根据SerDes的建议配置。
2. 配置LT_POST_CURSOR_CFG.CFG_TRAIN_PRESET_POST，根据具体情况配置，建议配置为0x80。
3. 配置LT_POST_CURSOR_CFG.CFG_TRAIN_MAX_POST，根据SerDes的限制配置。
4. 配置LT_POST_CURSOR_CFG.CFG_TRAIN_MIN_POST，根据SerDes的限制配置。

步骤11 配置LT_MAIN_CURSOR_CFG寄存器。

1. 配置LT_MAIN_CURSOR_CFG.CFG_TRAIN_INIT_MAIN，根据SerDes的建议配置。
2. 配置LT_MAIN_CURSOR_CFG.CFG_TRAIN_PRESET_MAIN，根据具体情况配置，建议配置为LT_MAIN_CURSOR_CFG.CFG_TRAIN_MAX_MAIN。
3. 配置LT_MAIN_CURSOR_CFG.CFG_TRAIN_MAX_MAIN，根据SerDes的限制配置。
4. 配置LT_MAIN_CURSOR_CFG.CFG_TRAIN_MIN_MAIN，根据SerDes的限制配置。

步骤12 配置LT_PRBS_CFG寄存器。

1. 配置LT_PRBS_CFG.CFG_PRBS_SEED_MODE，根据具体情况配置，如果是单lane 10G，那么配置为1；如果是单lane 25G，那么配置为2。
2. 配置LT_PRBS_CFG.CFG_PRBS_SEED_FIXED，如果是25G，那么配置为0；如果是4lane 100G，那么lane0配置为0，lane1配置为1，以此类推。

步骤13 配置LT_SPI_CFG寄存器。

配置LT_SPI_CFG.CFG_READ_DELAY为1。

步骤14 配置LT_MS_CNT_CFG寄存器。

配置LT_MS_CNT_CFG.TRAIN_TIMER_1MS_UNIT，根据SerDes频点配置，SerDes速率为25G时，配置为644531，SerDes速率为10G时，配置为515625。

步骤15 配置LT_TIMER_CFG寄存器。

配置LT_TIMER_CFG.CFG_TRAIN_MAX_TIMER_ENB=1和
LT_TIMER_CFG.CFG_TRAIN_MAX_TIMER（寄存器默认值即可）。

步骤16 软件读清TRAINING、SPI或者AHB的CAR寄存器，并写清中断，初始化完成。

----结束

4.2.5 硬件 Training 流程

步骤1 参考Training时钟复位初始化。

步骤2 配置LT_CONTROL_CFG1寄存器。

配置LT_CONTROL_CFG1.CFG_TRAINING_MODE=0选择硬件training模式。

步骤3 打开Hilink RXTXLTRAINEN (DSRX_CSR58 bit[3]配置为1) 功能, 使能TRAIN 配置 Hilink FFE值。

配置Hilink的FFE值（和training的initial值保持一致）。

步骤4 解复位操作。

同时配置LT_CONTROL_CFG2. RESET_TRAIN_TX=0、LT_CONTROL_CFG2. RESET_TRAIN_RX=0和LT_CONTROL_CFG2. RESET_TRAIN_SPI=0，完成解复位，进入下一步。

步骤5 delay一段时间后校准Hilink（5ms）。

步骤6 配置LT CONTROL CFG1寄存器。

同时配置LT_CONTROL_CFG1.CFG_TRAINING_ENABLE=1、LT_CONTROL_CFG1.TRAIN_RESTART=1。

步骤7 如果是initial流程，跳到第9[q(1)步，否则等待中断，收到中断之后进入中断服务程序，进入下一步。

步骤8 如果是TRAIN_INIT_DONE中断，进入下一步；否则如果是TRAIN_SUCCESS中断，那么软件切换通路到PCS(配置RTC寄存器中对应MAC的MAC_AN_ENABLE.cfg_train_en为1)，clock_gate住training，跳回AN操作流程中的第9步（initial_or_preset_only=1）；如果接收TRAIN_FAIL，clock_gate住training和AN（如果有AN）。

步骤9 开启SerDes CTLE自适应，进入下一步。

步骤10 软件等待SerDes CTLE自适应完成。进入下一步。

步骤11 更改外部寄存器，打开读清使能(RTC.MAC_RD_CLR)。读LT_DME_CNT 寄存器。关闭读清使能。

图 4-19 MAC/AN TRAIN 统计寄存器读清控制

MAC_012A	MAC/AA/DAI设计寄存器读控制	0x020	32	0x00000003	reserved	31:2	0	0x00000000	保留=
						0x_train_r_d_clr	1	0	0x_train_r_d_clr 1: CAG训练寄存器清零 0: CAG训练寄存器不清
						lcp_r_d_clr	0	0	0x_lcp_r_d_clr 1: CAG训练寄存器清零 0: CAG训练寄存器不清

步骤12 配置LT CONTROL CFG2寄存器。

如果是initial流程，那么配置LT_CONTROL_CFG2.FROZEN_IDLE_EN=0；否则配置LT_CONTROL_CFG2.FROZEN_INI_EN=0，进入下一步。

步骤13 软件等待上报的Training中断。软件收到中断后，进入中断服务程序。



步骤14 中断服务程序中检查中断类型，如果接收到TRAIN_SUCCESS中断，软件切换通路到PCS，clock_gate住training，跳回AN操作流程中的第15步；如果接收TRAIN_FAIL，clock_gate住training和AN（如果有AN）。

----结束

4.2.6 软件 Training 流程

步骤1 参考Training时钟复位初始化。

步骤2 配置LT_CONTROL_CFG1寄存器。

配置LT_CONTROL_CFG1.CFG_TRAINING_MODE=1选择软件training模式。

步骤3 配置LT_CONTROL_CFG2寄存器。

配置LT_CONTROL_CFG2. FROZEN_IDLE_EN= 1停止硬件training状态机跳转。

步骤4 切速，配置hilink的FFE值。（和training的initial值保持一致）。数据通路切，换至training通路。

步骤5 解复位操作。

同时配置LT_CONTROL_CFG2. RESET_TRAIN_TX = 0、LT_CONTROL_CFG2. RESET_TRAIN_RX = 0，LT_CONTROL_CFG2. RESET_TRAIN_SPI = 0，完成解复位，进入下一步。

步骤6 delay一段时间后校准（5ms）。

步骤7 配置LT_CONTROL_CFG1寄存器。

同时配置LT_CONTROL_CFG1.CFG_TRAINING_ENABLE=1、LT_CONTROL_CFG1.TRAIN_RESTART=1。

步骤8 如果是initial流程，那么跳到步骤**步骤10**；否则配置LT_SOFT_TRAIN_CFG.CFG_PRESET_CMD=1。

步骤9 软件等待中断，如果是TRAIN_FAIL中断，那么training失败，clock_gate住training和AN（如果有AN）；如果是TRAIN_SOFT_CMD_TRIP_END中断，那么进入下一步。

步骤10 软件开启SerDes CTLE自适应。

步骤11 软件等待SerDes CTLE自适应完成，再进入下一步。

步骤12 软件按软件需要的算法，配置LT_SOFT_TRAIN_CFG寄存器，控制Training发送相应的命令，进入下一步。

步骤13 软件等待中断，如果是TRAIN_FAIL中断，那么training失败，clock_gate住training和AN（如果有AN）；如果是TRAIN_SOFT_CMD_TRIP_END中断，进入下一步。

步骤14 软件根据算法要求，决定是否读取眼图，如果不读眼图，那么跳到软件Training流程**步骤17**。

如果需要读取眼图，则配置LT_SOFT_TRAIN_CFG.CFG_START_MARGIN=1，发起读取眼图命令。

步骤15 软件等待中断，如果是TRAIN_FAIL中断，那么training失败，clock_gate住training和AN（如果有AN）；如果是TRAIN_SOFT_FETCH_EYE中断，进入下一步。

步骤16 读取LT_CUR_EYE_MARGIN_STATUS. MARGIN_VALUE。



- 步骤17** 软件判定Training是否完成。
- 1. 如果Training没有完成，那么跳到软件Training流程**步骤12**；
 - 2. 如果认为Training已经完成，配置
LT_SOFT_TRAIN_CFG.CFG_RECEIVE_READY_CMD = 1完成本端Training，进入下一步。
- 步骤18** 等待TRAIN_SUCCESS 或者TRAIN_FAIL中断，收到中断后进入下一步。
- 步骤19** 如果接收到TRAIN_SUCCESS中断，软件切换通路到PCS，clock_gate住training，跳回AN操作流程中的**步骤9**；如果接收TRAIN_FAIL，clock_gate住training和AN（如果有AN）。
- 结束

4.2.7 An_LT 大流程

4.3 MAG 初始化流程

MAG模块初始化配置基本流程说明如下。涉及到具体的IP初始化配置时，请查阅IP相关的FS中关于初始化流程的说明。使用不同的IP时**步骤3~步骤5**的步骤略有不同。

- 步骤1** 确认MAG相关的工作时钟ICG使能开关打开。详细查阅《Hi309a寄存器手册》PPE部分描述。

图 4-20 PPE 时钟使能寄存器 0

3C_PPE_105_P00	PPE时钟使能寄存器	533B	0x031B	32	0x00001FFF	reserved	31:13	-	0x0000	位描述
						icg_en_mac_igpu	8	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_mac_igpu_cfg	7	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_mac_app	6	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_mac_cfg	5	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_vse_cfg	4	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_vsec_dec	3	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_vsec_enc	2	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_vsec_common	1	MREG_SET_WO	0x1	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。

图 4-21 PPE 时钟使能寄存器 1

3C_PPE_105_P01	PPE时钟使能寄存器	533B	0x031B	32	0x00001FFF	icg_en_igpu	31:24	MREG_SET_WO	0x0F	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_igpu_cfg	23:16	MREG_SET_WO	0x0F	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_igpu_dec	15:8	MREG_SET_WO	0x0F	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。
						icg_en_igpu_enc	7:0	MREG_SET_WO	0x0F	工作时钟 使能位，高有效。 0：无效； 1：打开时钟。



图 4-22 PPE 时钟使能寄存器 2

SC_PPE_CLK_EN2	PPE时钟使能寄存器	5328	0x0328	32		0x03FFFFFF	reserved	31:28	-	0x0	保留位
							icg_en_train_comma n	27:24	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。
							icg_en_train_spi	23:20	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。
							icg_en_ahb_tx	19:16	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。
							icg_en_ahb_rx	15:12	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。
							icg_en_ahb_core	11:8	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。
							icg_en_train_rx	7:4	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。
							icg_en_train_tx	3:0	MREG_SET_WO	0x0	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。

图 4-23 PPE 时钟使能寄存器 3

SC_PPE_CLK_EN3	PPE时钟使能寄存器	5328	0x0328	32		0x03FFFFFF	reserved	31:28	-	0x0	保留位
							icg_en_ahb_nb	25:0	MREG_SET_WO	0x3FFFFFF	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。

图 4-24 PPE 时钟使能寄存器 4

SC_PPE_CLK_EN4	PPE时钟使能寄存器	5328	0x0328	32		0x03FFFFFF	reserved	31:28	-	0x0	保留位
							icg_en_ahb_nb	25:0	MREG_SET_WO	0x3FFFFFF	工作时钟 使能位，高有效。 0: 无效； 1: 打开时钟。

步骤2 配置MAG相关的软复位请求寄存器为复位状态（HiMACV110 XXVGE的解复位在STEP5中IP初始化流程中控制）。详细查阅《Hi309a寄存器手册》PPE部分描述。

图 4-25 PPE 软复位请求控制寄存器 0

SC_PPE_RESET_REQ0	PPE软复位请求控制寄存器	540C	0x0A10	32		0x0001FFFF	reserved	31:17	-	0x0000	保留。
							srst_req_mac_regu	1	MREG_SET_WO	0x1	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_mac_regu_cfg	0	MREG_SET_WO	0x1	软复位请求 0: 无效； 1: 软复位有效。

图 4-26 PPE 软复位请求控制寄存器 1

SC_PPE_RESET_REQ1	PPE软复位请求控制寄存器	5A10	0x0A18	32		0x0FFFFFFF	srst_req_ahb_train_comma n	31:28	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_ahb_train_cfg	27:24	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_ahb_train_spi	23:20	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_ahb_tx	19:16	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_ahb_rx	15:12	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_ahb_core	11:8	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_train_rx	7:4	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_train_tx	3:0	MREG_SET_WO	0x0	软复位请求 0: 无效； 1: 软复位有效。

图 4-27 PPE 软复位请求控制寄存器 2

SC_PPE_RESET_REQ2	PPE软复位请求控制寄存器	5A14	0x0A20	32		0x0FFFFFFF	reserved	31:28	-	0x0	保留。
							srst_req_mag_ahb_ige	27:2	MREG_SET_WO	0x3FFFFFF	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_mag_ahb_comma n	1	MREG_SET_WO	0x1	软复位请求 0: 无效； 1: 软复位有效。
							srst_req_mag_cfg	0	MREG_SET_WO	0x1	软复位请求 0: 无效； 1: 软复位有效。



图 4-28 PPE 软复位请求控制寄存器 3

SC_PPE_RESET_REQ3	PPE软复位请求控制寄存器	SA18	0x0A28	32		0x0000001F	reserved	31:5	-	0x00000000	保留。
							srst_req_xsec_cfg	4	MREG_SET_WO	0x1	软复位请求 0: 无效; 1: 软复位有效。
							srst_req_xsec_dec	3	MREG_SET_WO	0x1	软复位请求 0: 无效; 1: 软复位有效。
							srst_req_xsec_enc	2	MREG_SET_WO	0x1	软复位请求 0: 无效; 1: 软复位有效。
							srst_req_xsec_common	1	MREG_SET_WO	0x1	软复位请求 0: 无效; 1: 软复位有效。

步骤3 配置MAG相关的软复位去请求寄存器为解复位状态 (HiMACV110 XXVGE的解复位在 STEP5中IP初始化流程中控制)。《Hi309a寄存器手册》PPE部分描述。

图 4-29 PPE 软复位去请求控制寄存器 0

SC_PPE_RESET_REQ0	PPE软复位去请求控制寄存器	0x0A14	32			0x00000000	reserved	31:17	-	0x0000	保留。
							srst_dreq_mac_iegu	1	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_mac_iegu_cfg	0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。

图 4-30 PPE 软复位去请求控制寄存器 1

SC_PPE_RESET_REQ1	PPE软复位去请求控制寄存器	0x0A1C	32			0x00000000	srst_dreq_an_train_common	31:28	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_an_train_cfg	27:24	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_an_train_xpi	23:20	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_an_rx	19:16	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_an_tx	15:12	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_an_core	11:8	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_train_rx	7:4	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_train_tx	3:0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。

图 4-31 PPE 软复位去请求控制寄存器 2

SC_PPE_RESET_REQ2	PPE软复位去请求控制寄存器	0x0A24	32			0x00000000	reserved	31:28	-	0x0	保留。
							srst_dreq_mag_app_ie	27:2	MREG_CLR_WO	0x00000000	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_mag_app_common	1	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_mag_cfg	0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。

图 4-32 PPE 软复位去请求控制寄存器 3

SC_PPE_RESET_REQ3	PPE软复位去请求控制寄存器	0x0A2C	32			0x00000000	reserved	31:5	-	0x00000000	保留。
							srst_dreq_xsec_cfg	4	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_xsec_dec	3	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_xsec_enc	2	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_xsec_common	1	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。
							srst_dreq_etc	0	MREG_CLR_WO	0x0	撤销软复位请求 0: 无效; 1: 撤销软复位请求。

步骤4 关闭IGU EGU的接收发送使能；详细参见《Hi309a MAG 寄存器描述》中IGU_EGU部分描述。



图 4-33 MAC 通道配置寄存器

MAC_EN_CFG	MAC 通道配置寄存器	rw	0x00000050~0x00000053	32	rw	0x0000000F	reserved	31:4	RD	0x00000000	保留。
							mac_rx_en_inner	3	RD	0x1	芯片内部I/OV MAC接收通道使能状态，I/OV MAC接收通道使能寄存器配置是否生效，需要查看该状态。
							mac_tx_en	2	RW	0x1	I/OV MAC 1) 接收通道使能，在MAC模式切换时，需要关闭接收，再进行模式切换。在模式配置完成后再开启接收。 2) 在时钟门控打开的场景下，配置该寄存器之前，需要关闭时钟门控！
							mac_tx_en_inner	1	RD	0x1	I/OV MAC发送通道使能，需要查看该状态。
							mac_tx_en	0	RW	0x1	I/OV MAC发送通道使能。 1) 在MAC模式切换时，需要关闭发送，再进行模式切换。在模式配置完成后再开启发送。 2) 在时钟门控打开的场景下，配置该寄存器之前，需要关闭时钟门控！

- 步骤5** 如果速率为GE，则配置hilink los信号屏蔽掉hilink的los信号（寄存器为MAG.COMMMON.MAC_SerDes_LOS_MSK.lge_SerDes_los_msk配置为1）否则配置不屏蔽hilink los。根据网口速率模式，初始化配置对应的IP的相关寄存器。具体请查阅IP部分的初始化流程。
- 步骤6** 配置IGU_EGU的接收发送使能有效。

图 4-34 MAC 通道配置寄存器

MAC_EN_CFG	MAC 通道配置寄存器	rw	0x00000050~0x00000053	32	rw	0x0000000F	reserved	31:4	RD	0x00000000	保留。
							mac_rx_en_inner	3	RD	0x1	芯片内部I/OV MAC接收通道使能状态，I/OV MAC接收通道使能寄存器配置是否生效，需要查看该状态。
							mac_tx_en	2	RW	0x1	I/OV MAC 1) 接收通道使能，在MAC模式切换时，需要关闭接收，再进行模式切换。在模式配置完成后再开启接收。 2) 在时钟门控打开的场景下，配置该寄存器之前，需要关闭时钟门控！
							mac_tx_en_inner	1	RD	0x1	I/OV MAC发送通道使能，需要查看该状态。
							mac_tx_en	0	RW	0x1	I/OV MAC发送通道使能。 1) 在MAC模式切换时，需要关闭发送，再进行模式切换。在模式配置完成后再开启发送。 2) 在时钟门控打开的场景下，配置该寄存器之前，需要关闭时钟门控！

----结束

4.4 MAC 速率切换流程

4.4.1 XGE 切换到 GE

- 步骤1** 模式切换前配置IGU-EGU将端口接收使能IGU_EGU.MAC_EN_CFG.mac_rx_en=0。（此步骤的目的是停止上行的数据流）。同时将SSU的对应通道发送使能关闭，配置SSU.SSU_TX_OQ_SCHEDULE_EN[port_id]=0（此步骤的目的是停止下行的数据流）。
- 步骤2** 如果切换之前端口响应pause帧的使能是有效的。则关闭切换前的端口对应的pause帧响应使能。
- 步骤3** 等待1ms。
- 步骤4** 查询IGU_EGU对应通道使能关闭配置是否生效，
IGU_EGU.MAC_EN_INNER.mac_rx_en_inner[0~3]=0。
对应通道EGU的发送Buffer是否为空：
IGU_EGU_TNL_INT_MSIX_STS.tx_buf_sta_dfx[7]=1。



如果以上2个条件都满足则跳到**步骤5**，否则进入**步骤3**。

- 步骤5** 如果**步骤2**中有关闭切换前的端口对应的pause帧响应使能，则恢复切换前的端口对应的pause帧响应使能配置。
- 步骤6** 配置IGU_EGU中的mac_rate_cfg为对应的速率。

图 4-35 MAC 类型配置寄存器

MAC_NODE_CFG	MAC类型配置寄存器	reg	0x0000054+0x400 #MAC_04	32	RV	0x00000000	reserved	31:9	80	0x00000000	保留*
							mac_rate_egu_inner	0:6	80	0x0	EGU 内部生效的MAC类型配置寄存器* 3'8000: 当前MAC类型为1G 3'8001: 当前MAC类型为5G_1 3'8010: 当前MAC类型为5G_1 3'8011: 当前MAC类型为10G1 3'8100: 当前MAC类型为802.11n 其他: 保留*
							mac_rate_egu_inner	5:3	80	0x0	EGU 内部生效的MAC类型配置寄存器* 3'8000: 当前MAC类型为1G 3'8001: 当前MAC类型为5G_1 3'8010: 当前MAC类型为5G_1 3'8011: 当前MAC类型为10G1 3'8100: 当前MAC类型为802.11n 其他: 保留*
							mac_rate_cfg	2:0	8F	0x0	MAC类型配置寄存器* 3'8000: 当前MAC类型为1G 3'8001: 当前MAC类型为5G_1 3'8010: 当前MAC类型为5G_1 3'8011: 当前MAC类型为10G1 3'8100: 当前MAC类型为802.11n 其他: 保留*

- 步骤7** 配置Hilink速率为GE模式。详细配置请查阅Hilink配置指导。
- 步骤8** 按通道号选择复位如下复位信号：srst_xxvge_core_n{0..25}, srst_app_mac_n{0..25}。
- 步骤9** 内部实际例化6个X4和2个X1；前24个端口4个port为一组，一个core复位则整体4个端口全部复位。
- 步骤10** 初始化配置XXVGE端口，选择对应的配置模式。配置流程参见XXVGE 初始化。
- 步骤11** 解复位**步骤7**中的处于复位状态的复位信号。
- 步骤12** 打开SSU对应端口的发送调度使能SSU.SSU_TX_OQ_SCHEDULE_EN=1。
- 步骤13** 打开IGU-EGU对应端口接收使能。（IGU_EGU.MAC_EN_CFG.mac_rx_en=1）。

----结束

4.4.2 GE 切换到 XGE

- 步骤1** 模式切换前配置IGU-EGU将端口接收使能IGU_EGU.MAC_EN_CFG.mac_rx_en=0。
（此步骤的目的是停止上行的数据流）。同时将SSU的对应通道发送使能关闭，配置SSU.SSU_TX_OQ_SCHEDULE_EN[port_id]=0（此步骤的目的是停止下行的数据流）。
- 步骤2** 如果切换之前端口响应pause帧的使能是有效的。则关闭切换前的端口对应的pause帧响应使能。
- 步骤3** 等待1ms。
- 步骤4** 查询IGU_EGU对应通道使能关闭配置是否生效，
IGU_EGU.MAC_EN_INNER.mac_rx_en_inner[0~3]=0。
对应通道EGU的发送Buffer是否为空：
IGU_EGU_TNL_INT_MSIX_STS.tx_buf_sta_dfx[7]=1。
如果以上2个条件都满足则跳到**步骤5**，否则进入**步骤3**。
- 步骤5** 如果**步骤2**中有关闭切换前的端口对应的pause帧响应使能，则恢复切换前的端口对应的pause帧响应使能配置。



步骤6 配置IGU_EGU中的mac_rate_cfg为对应的速率。

图 4-36 MAC 类型配置寄存器

MAC_NODE_CFG	MAC类型配置寄存器	reg	0x0000054+0x00000000	32	RV	0x00000000	reserved	31:9	RD	0x00000000	保留。
							mac_rate_egu_inner	0:6	RD	0x0	EGU 内部生成的MAC类型配置寄存器。 3'0000: 当前MAC类型为1G 3'0001: 当前MAC类型为5G 3'0010: 当前MAC类型为5G+ 3'0011: 当前MAC类型为10G+ 3'0100: 当前MAC类型为X5000II+ 其他: 保留。
							mac_rate_egu_outer	5:3	RD	0x0	EGU 内部生成的MAC类型配置寄存器。 3'0000: 当前MAC类型为1G 3'0001: 当前MAC类型为5G 3'0010: 当前MAC类型为5G+ 3'0011: 当前MAC类型为10G+ 3'0100: 当前MAC类型为X5000II+ 其他: 保留。
							mac_rate_cfg	2:0	RV	0x0	MAC类型配置寄存器。 3'0000: 当前MAC类型为1G 3'0001: 当前MAC类型为5G 3'0010: 当前MAC类型为5G+ 3'0011: 当前MAC类型为10G+ 3'0100: 当前MAC类型为X5000II+ 其他: 保留。

步骤7 配置hilink速率为XGE模式。详细配置请查阅Hilink配置指导。

步骤8 按通道号选择复位如下复位信号：srst_xxvge_core_n{0..7}, srst_xxvge_core_async{0..7}, srst_app_mac_n{0..25}。内部实际例化6个X4和2个X1；前24个端口4个port为一组，一个core复位则整体4个端口全部复位。

步骤9 初始化配置XXVGE端口，选择对应的配置模式。配置流程参见XXVGE 初始化。

步骤10 解复位step7中的处于复位状态的复位信号。

步骤11 打开SSU对应端口的发送调度使能SSU.SSU_TX_OQ_SCHEDULE_EN=1。

步骤12 打开IGU-EGU对应端口接收使能。（IGU_EGU.MAC_EN_CFG.mac_rx_en=1）。

----结束

4.4.3 带流切换 XSEC 加解密的 bypass/非 bypass

步骤1 发送方向将SSU对应的通道发包调度使能关闭，将对应MAC端口响应接收pause帧的使能关闭，接收方向将IGU的接收使能关闭配置；IGU_EGU.MAC_EN_CFG.mac_rx_en[port_num]=0。

步骤2 等待1ms 查询对应通道的EGU BUFFER中的报文缓存是否为空，非空则继续等待1ms后查询，超出10ms的等待，BUFFER还为非空则软件上报异常，注意MAC不能是被复位的状态，否则报文是无法排空的。

图 4-37 IGU 和 EGU MAC 通道异常中断状态寄存器

IGU_EGU_TNL_INT_MSDX_STS	IGU和EGU MAC通道异常中断状态寄存器	reg	0x0035070+0x10000000+mac_n*4	32	RD	0x00010000	reserved	31:27	RD	0x00	保留。
							tx_buf_sta_db	26:18	RD	0x000	EGU内部BUFFER状态。 (0) 空闲状态 (1) 忙状态 (2) 内部缓存有数据个数

步骤3 查询IGU_EGU.MAC_EN_INNER.mac_rx_en_inner=0，非0则继续等待1ms后查询，确认接收方向是否断流成功。

步骤4 完成step1~3发送和接收方向都断流成功后，则可以配置XSEC的bypass寄存器（MAG.XSEC_DPPA_COMMON）进行切换。

图 4-38 按通道 bypass xsec 的解密路径

XSEC_DEC_PORT_BY_PASS	按通道bypass xsec的解密路径	reg	0x240	0x00000000	0	cfg_xsec_dec_port_bypass
-----------------------	---------------------	-----	-------	------------	---	--------------------------



图 4-39 按通道 bypass xsec 的加密路径

XSEC_ENC_PORT_BY PASS	按通道bypass xsec的加密路径	reg	0x344	0x00000000 0	cfg_xsec_enc_port_bypass
--------------------------	---------------------	-----	-------	-----------------	--------------------------

步骤5 恢复MAC的响应接收pause帧配置，将IGU的接收使能开启
IGU_EGU.MAC_EN_CFG.mac_rx_en[port_num]=1；SSU对应的通道发包调度使能开启；

如果需要整体bypass XSEC，则需要将26个通道都进行**步骤1~步骤3**发送和接收方向的断流操作。

----结束

4.4.4 带 XSEC 加解密的速率切换流程

- 步骤1** 发送方向将SSU对应的通道发包调度使能关闭，将对应MAC端口响应接收pause帧的使能关闭，接收方向将IGU的接收使能关闭配置；
IGU_EGU.MAC_EN_CFG.mac_rx_en[port_num]=0。
- 步骤2** 等待1ms 查询对应通道的EGU BUFFER中的报文缓存是否为空，非空则继续等待1ms后查询，超出10ms的等待，BUFFER还为非空则软件上报异常，注意MAC不能是被复位的状态，否则报文是无法排空的。

图 4-40 IGU 和 EGU MAC 通道异常中断状态寄存器

IGU_EGU_TNL_INT_ MSIX_STS	IGU/EGU MAC通道异常中断状态 寄存器	reg	0x2035070H/0x10000Hmac_x4	32	80	0x02010000	reserved	31:27	80	0x00	保留*
							ts_buf_sta_dfs	26:18	80	0x080	EGU内部BUFFER状态， (0)空闲状态 (1)空状态 (05)内部缓存有效数据个数

- 步骤3** 查询IGU_EGU.MAC_EN_INNER.mac_rx_en_inner=0，非0则继续等待1ms后查询，确认接收方向是否断流成功。
- 步骤4** 完成**步骤1~步骤3**发送和接收方向都断流成功后，参照**MAC速率切换流程**，在**步骤9**到**步骤11**之间（**步骤10**为解复位MAC逻辑，**步骤11**为打开SSU端口调度，step12为打开IGU_EGU收发使能），插入对XSEC的修改。
- 步骤5** XSEC的修改内容如下：
- 根据速率切换时port的变化，修改顶层bypass port信息（这个根据业务场景来配置），XSEC_DPPA_COMMON.XSEC_DEC_PORT_BYPASS[port_num]=0，
XSEC_DPPA_COMMON.XSEC_ENC_PORT_BYPASS[port_num]=0：

图 4-41 按通道 bypass xsec 的解密路径

XSEC_DEC_PORT_BY PASS	按通道bypass xsec的解密路径	reg	0x240	0x00000000 0	cfg_xsec_dec_port_bypass
--------------------------	---------------------	-----	-------	-----------------	--------------------------

图 4-42 按通道 bypass xsec 的加密路径

XSEC_ENC_PORT_BY PASS	按通道bypass xsec的加密路径	reg	0x344	0x00000000 0	cfg_xsec_enc_port_bypass
--------------------------	---------------------	-----	-------	-----------------	--------------------------

步骤6 针对新的速率接口，调整反压水线和credit信息：



图 4-43 XSEC_ENC_OADP_PORT_BUF_CFG{0..25}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSECV200B05	XSEC_ENC	XSEC_ENC_OADP_PORT_BUF_CFG{0..25}	0x0000_0000_0998	flops	n/a
Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:27	RESERVED	RO	Reserved.	0	n/a
26:16	XOFF_TH	RW	The backpressure threshold value. Each value represent for one 16B unit. Note: 1. The value for unused port can't be 0x0. 2. User must ensure the configuration value within the valid range from 2*7 to (ADDR_SIZE - maximum overshoot)*7. The register can be configured on the fly.	0x200	n/a
15:9	RESERVED	RO	Reserved.	0	n/a
8:0	ADDR_SIZE	RW	Address size of port fifo. Each value represents a depth of 7*16B RAM. Note: The value for used port can't be smaller than 3, or the logic will force the size to 3. It can not be configured to 0 when the port is used. It can be configured only when the port has no traffic.	0x100	n/a

图 4-44 XSEC_ENC_IADP_PORT_BUF_CFG{0..25}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSECV200B05	XSEC_ENC	XSEC_ENC_IADP_PORT_BUF_CFG{0..25}	0x0000_0000_0A08	flops	n/a
Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:26	RESERVED	RO	Reserved.	0	n/a
25:16	XOFF_TH	RW	The backpressure threshold value. Each value represent for one 64B unit. Note: 1. The value for unused port can't be 0x0. 2. User must ensure the configuration value within the valid range from 2 to (ADDR_SIZE - maximum overshoot). This register can be configured on the fly.	0x20	n/a
15:10	RESERVED	RO	Reserved.	0	n/a
9:0	ADDR_SIZE	RW	Address size of port fifo. Each value represents a depth of 64B RAM. It can be configured only when the port has no traffic.	0x40	n/a

图 4-45 XSEC_ENC_PORT_CREDIT_TH{0..25}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSECV200B05	XSEC_ENC	XSEC_ENC_PORT_CREDIT_TH{0..25}	0x0000_0000_01F0	flops	n/a
Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:6	RESERVED	RO	Reserved.	0	n/a
5:0	VALUE	RW	The credit threshold value.	0x3F	n/a

图 4-46 XSEC_DEC_OADP_PORT_BUF_CFG{0..25}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSECV200B05	XSEC_DEC	XSEC_DEC_OADP_PORT_BUF_CFG{0..25}	0x0000_0000_0868	flops	n/a
Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:26	RESERVED	RO	Reserved.	0	n/a
25:16	XOFF_TH	RW	The backpressure threshold value. Each value represent for one 16B unit. Note: 1. The value for unused port can't be 0x0. 2. User must ensure the configuration value within the valid range from 2*5 to (ADDR_SIZE - maximum overshoot) * 5. The register can be configured on the fly.	0x100	n/a
15:8	RESERVED	RO	Reserved.	0	n/a
7:0	ADDR_SIZE	RW	Address size of port fifo. Each value represents a depth of 5*16B RAM. Note: The value for used port can't be smaller than 3, or the logic will force the size to 3. It can not be configured to 0 when the port is used. It can be configured only when the port has no traffic.	0x80	n/a

图 4-47 XSEC_DEC_IADP_PORT_BUF_CFG{0..25}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSECV200B05	XSEC_DEC	XSEC_DEC_IADP_PORT_BUF_CFG{0..25}	0x0000_0000_08D8	flops	n/a
Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:10	RESERVED	RO	Reserved.	0	n/a
9:0	ADDR_SIZE	RW	Address size of port fifo. Each value represents a depth of 32B (for Hi1230) RAM. It can be configured only when the port has no traffic.	0x100	n/a

步骤7 回收原来port对应的时隙资源。寄存器如下，bit8为高，低5bit的值和port值相等的时隙资源需要被收回：



图 4-48 XSEC_DEC_CALENDAR_TIME_SLOT{0..127}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSEC200B05	XSEC_DEC	XSEC_DEC_CALENDAR_TIME_SLOT{0..127}	0x0000_0000_0254	flops	n/a

Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:9	RESERVED	RO	Reserved.	0	n/a
8:8	VLD	RW	Indicate this time slot is valid or invalid. 0 = invalid 1 = valid	0x0	n/a
7:5	RESERVED	RO	Reserved.	0	n/a
4:0	CAL	RW	configure the port number that should be scheduled out in this calendar time slot. the configured range is 0~31.	0x0	n/a

图 4-49 XSEC_ENC_CALENDAR_TIME_SLOT{0..127}

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSEC200B05	XSEC_ENC	XSEC_ENC_CALENDAR_TIME_SLOT{0..127}	0x0000_0000_0384	flops	n/a

Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:9	RESERVED	RO	Reserved.	0	n/a
8:8	VLD	RW	Indicate this time slot is valid or invalid. 0 = invalid 1 = valid	0x0	n/a
7:5	RESERVED	RO	Reserved.	0	n/a
4:0	CAL	RW	configure the port number that should be scheduled out in this calendar time slot.	0x0	n/a

步骤8 根据新的速率，重新分配时隙资源。bit8为低的所有时隙资源都可以使用，重新分配时，不能修改其他port的时隙资源；重新分配时，尽量做到均匀分配。

步骤9 最后，使能新的时隙分配模型：

图 4-50 XSEC_DEC_IADP_CALENDAR_LOAD_EN

4.3.50 - XSEC_DEC_IADP_CALENDAR_LOAD_EN - IADP Schedule Calendar Time Slot Load Enable Configuration Register

This register can be configured on the fly.

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSEC200B05	XSEC_DEC	XSEC_DEC_IADP_CALENDAR_LOAD_EN	0x0000_0000_0454	flops	n/a

Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:1	RESERVED	RO	Reserved.	0	n/a
0:0	VALUE	RWW	When user modify the CALENDAR_SLOT_LENGTH and CALENDAR_TIME_SLOT registers, the user should write 1 to this IADP_CALENDAR_LOAD_EN register to active the new calendar configuration, after the hardware has finished load all configuration, LOAD_EN will be clear to 0.	0x0	n/a

4.3.51 - XSEC_DEC_OADP_CALENDAR_LOAD_EN - OADP Schedule Calendar Time Slot Load Enable Configuration Register

This register can be configured on the fly.

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSEC200B05	XSEC_DEC	XSEC_DEC_OADP_CALENDAR_LOAD_EN	0x0000_0000_0450	flops	n/a

Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:1	RESERVED	RO	Reserved.	0	n/a
0:0	VALUE	RWW	When user modify the CALENDAR_SLOT_LENGTH and CALENDAR_TIME_SLOT registers, the user should write 1 to this OADP_CALENDAR_LOAD_EN register to active the new calendar configuration, after the hardware has finished load all configuration, LOAD_EN will be clear to 0.	0x0	n/a

图 4-51 XSEC_ENC_IADP_CALENDAR_LOAD_EN

5.3.53 - XSEC_ENC_IADP_CALENDAR_LOAD_EN - IADP Schedule Calendar Time Slot Load Enable Configuration Register

This register can be configured on the fly.

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSEC200B05	XSEC_ENC	XSEC_ENC_IADP_CALENDAR_LOAD_EN	0x0000_0000_0584	flops	n/a

Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:1	RESERVED	RO	Reserved.	0	n/a
0:0	VALUE	RWW	When user modify the CALENDAR_SLOT_LENGTH and CALENDAR_TIME_SLOT registers, the user should write 1 to this IADP_CALENDAR_LOAD_EN register to active the new calendar configuration, after the hardware has finished load all configuration, LOAD_EN will be clear to 0.	0x0	n/a

5.3.54 - XSEC_ENC_OADP_CALENDAR_LOAD_EN - OADP Schedule Calendar Time Slot Load Enable Configuration Register

This register can be configured on the fly.

Device	Block Name	Register Name	Start Address	Type	Tags
HIXSEC200B05	XSEC_ENC	XSEC_ENC_OADP_CALENDAR_LOAD_EN	0x0000_0000_0588	flops	n/a

Bits	Field	Access Type	Description	Reset Val	Behaviour Type
31:1	RESERVED	RO	Reserved.	0	n/a
0:0	VALUE	RWW	When user modify the CALENDAR_SLOT_LENGTH and CALENDAR_TIME_SLOT registers, the user should write 1 to this OADP_CALENDAR_LOAD_EN register to active the new calendar configuration, after the hardware has finished load all configuration, LOAD_EN will be clear to 0.	0x0	n/a

----结束

4.5 MAC SDS 环回流程

步骤1 将对应MAC的clk_xxvge_rx_sds时钟关闭，防止切换过程中的时钟毛刺。

图 4-52 PPF 时钟禁止寄存器

SC_PVR_305_0153	浮点寄存器	0x032C	32	0x00000000	reserved	31:26	-	0x00	保留位
					iq_dis_ige_nb	25:0	MREG_CLR_WO	0x00000000	工程启动 禁止位 0: 无效; 1: 关断时钟。

步骤2 配置/取消SDS串行/并行内环；详细配置请查阅Hilink配置指导。

步骤3 开启对应MAC的clk_xxvge_rx_sds时钟。

图 4-53 PPF 时钟使能寄存器

0x777_00_00	1777777777777777	0x00	0x0128	32	0x01FFFFFF	reserved	31:28	-	0x00	保留位
						icg_en_icg_nb	25:0	MREG_SET_WO	0x0FFFFFFF	工作时钟 使能位，高有效。 工作时钟 使能位 0: 关闭 1: 打开时钟。

说明

HiMACV110 不支持SerDes并口和串口环回，Hilink支持。

----结束

4.6 MIB 访问规则说明

- MIB寄存器是48bit位宽的寄存器，软件读或者写访问需要拆封成两次连续的32bit读或者写访问。
- 软件连续访问两个MIB地址读写操作之间间隔至少需要256个core cycle（400MHz）。



5 MAILBOX

5.1 概述

邮箱（MAILBOX）是用于VF-PF之间的消息传递，硬件提供一套通用的消息传递机制，方便软件实现对应功能。

5.2 特性

MAILBOX具有以下特性：

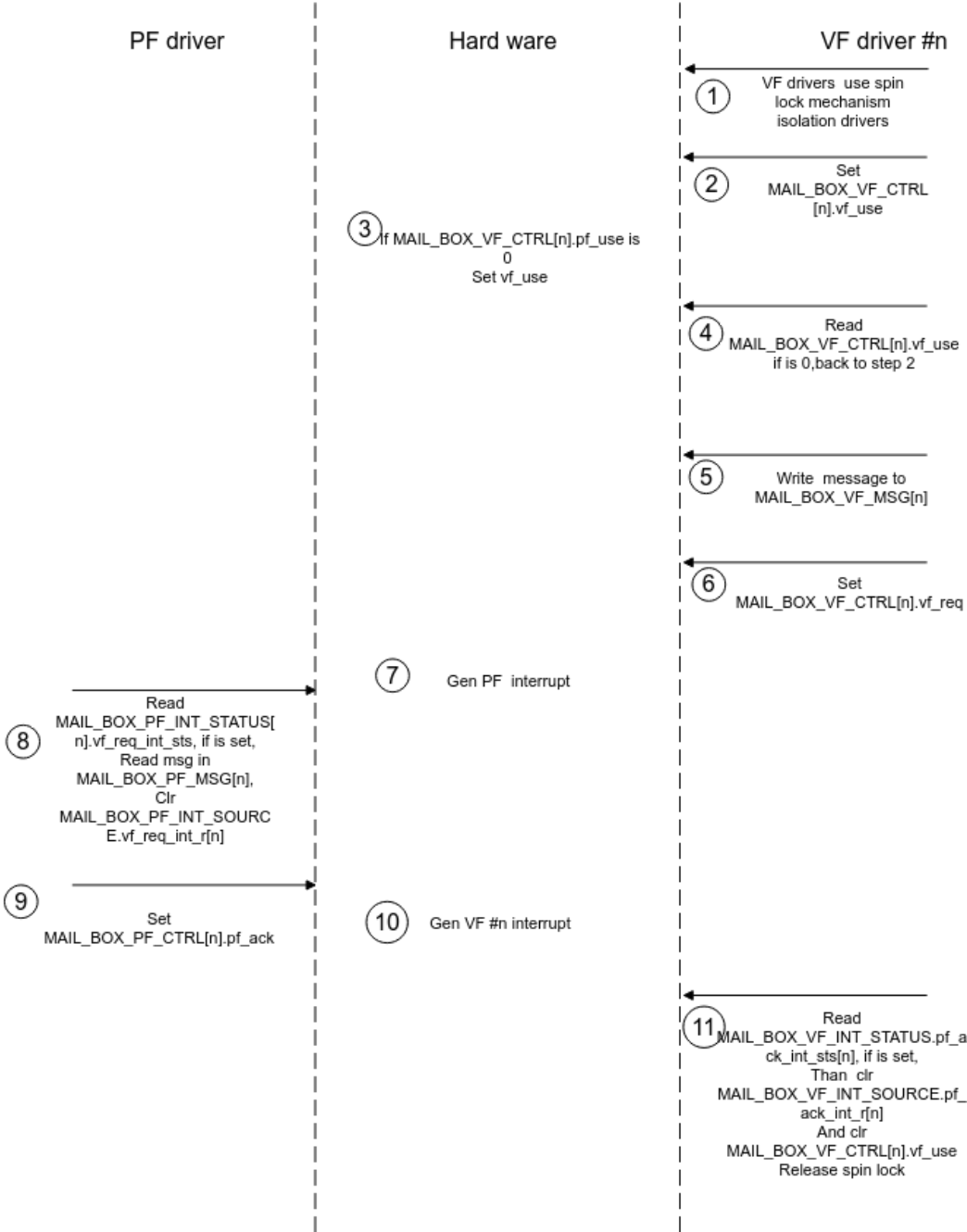
- 支持5个独立的VF和PF通信通道，每个通道能够独立工作，同时支持双向消息传输，即支持VF向PF发送消息，也支持PF向VF发送消息。
- 每个通道可承载的最大消息大小为128bit（该通道的VF和PF共用，不能VF和PF同时写，需要得到竞争权后才写）。
- 每个通道支持VF和PF同时竞争使用权（竞争机制：VF和PF向各自地址的竞争域写1，然后通过返回值判断竞争是否成功）。
- 支持直接通过操作PF的寄存器触发VF的中断，以此告知VF需执行的后续动作；同样地，VF亦可通过操作自身的寄存器直接激活PF上的中断，以提醒PF进行下一步工作。
- 每个VF拥有通道支持多达16个中断点，PF只需指定中断编号即可触发特定的中断事件（每个中断可以直接送到某个模块的VF软件）。
- PF只有1个中断，通过查看中断信息确认是哪个VF触发的中断。

5.3 功能描述



5.3.1 VF 向 PF 发送消息

图 5-1 VF 向 PF 发送消息流程图



具体步骤如下：

步骤1 VF #n驱动进行自旋锁操作，抢到自旋锁的驱动进入**步骤2**，否则停留在**步骤1**继续抢锁（n=0~7，下同）。



须知

如果同一个制式多个模块的驱动需要同时向PF发送消息，需要使用自旋锁，得到锁后再进入**步骤2**，不支持多个驱动不抢锁后直接进入**步骤2**，否则会出现消息异常或不可预期的结果。

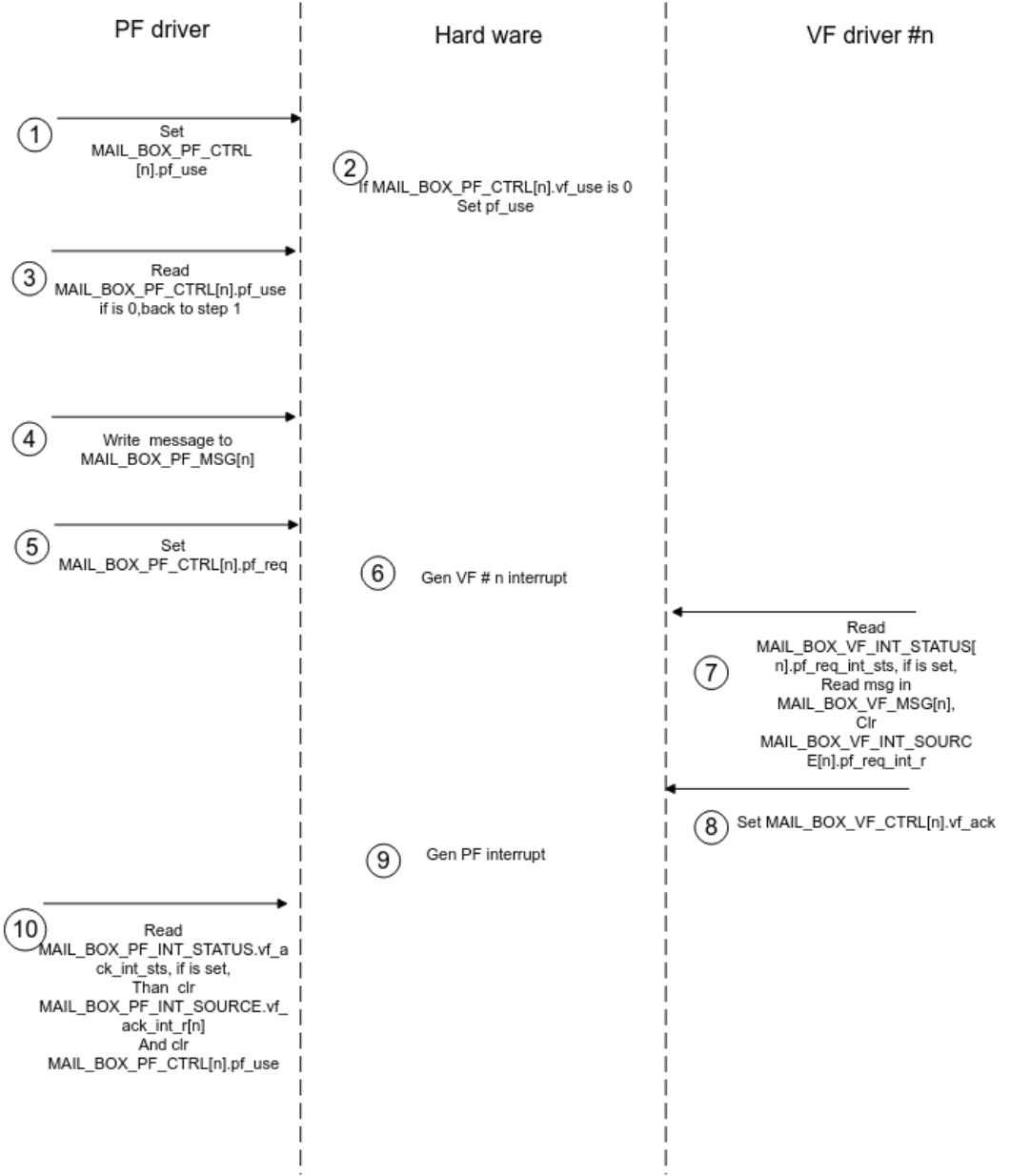
- 步骤2** VF #n驱动试图将MAIL_BOX_VF_CTRL[n].vf_use置位。
- 步骤3** 硬件收到置位操作后，判断该通道的PF是否在使用中，如果PF未使用，完成置位操作，否则忽略操作。
- 步骤4** VF #n驱动读取MAIL_BOX_VF_CTRL[n].vf_use，如果已经置位，则进入**步骤5**，否则返回**步骤2**（PF正在使用这个通道，VF没有获得使用权，返回**步骤2**重新尝试获得使用权）。
- 步骤5** VF #n驱动向MAIL_BOX_VF_MSG[n]写入消息。
- 步骤6** VF #n驱动将MAIL_BOX_VF_CTRL[n].vf_req置位。
- 步骤7** 硬件产生PF中断。
- 步骤8** PF驱动读出MAIL_BOX_PF_INT_STATUS.vf_req_int_sts[n]，如果已经置位，表示收到了VF的消息，从MAIL_BOX_VF_MSG[n]中取出消息处理；清除中断MAIL_BOX_PF_INT_SOURCE.vf_req_int_r[n]。
- 步骤9** PF驱动将MAIL_BOX_PF_CTRLpf_ack[n].置位。
- 步骤10** 硬件产生对于VF #n中断。
- 步骤11** VF #n驱动读出MAIL_BOX_VF_INT_STATUS.pf_ack_int_sts[n]，如果已经置位，表示PF已经收到并处理完消息，再清除中断MAIL_BOX_VF_INT_SOURCE[n].pf_ack_int_r，清除MAIL_BOX_VF_CTRL[n].vf_use，释放消息通道，并释放自旋锁。

----结束



5.3.2 PF 向 VF 发送消息

图 5-2 PF 向 VF 发送消息流程图



具体步骤如下：

- 步骤1** PF驱动试图将MAIL_BOX_PF_CTRL[n].pf_use置位（n=0~7，下同），如果多个PF驱动需要同时发消息，需要参考VF向PF发送消息的**VF向PF发送消息的步骤1**抢到自旋锁后再执行该步骤。
- 步骤2** 硬件收到置位操作后，判断该通道的VF是否在使用中，如果VF未使用，完成置位操作，否则忽略操作。
- 步骤3** PF驱动读取MAIL_BOX_PF_CTRL[n].pf_use，如果已经置位，则进入**步骤4**，否则返回**步骤1**。

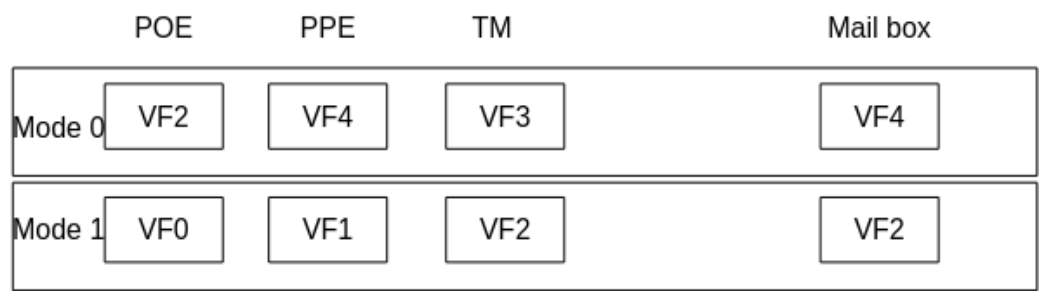


- 步骤4** PF驱动向MAIL_BOX_PF_MSG[n]写入消息。
- 步骤5** PF驱动将MAIL_BOX_PF_CTRL[n].pf_req置位。
- 步骤6** 硬件产生对应的VF中断。
- 步骤7** VF #n驱动读出MAIL_BOX_VF_INT_STATUS[n].pf_req_int_sts，如果已经置位，表示收到了PF的消息，从MAIL_BOX_VF_MSG[n]中取出消息处理；清除中断MAIL_BOX_VF_INT_SOURCE[n].pf_req_int_r。
- 步骤8** VF #n驱动将MAIL_BOX_VF_CTRL[n].vf_ack置位。
- 步骤9** 硬件产生PF中断。
- 步骤10** PF驱动读出MAIL_BOX_PF_INT_STATUS.vf_ack_int_sts[n]，如果已经置位，表示VF已经收到并处理完消息，再清除中断MAIL_BOX_PF_INT_SOURCE.vf_ack_int_r[n]，清除MAIL_BOX_PF_CTRL[n].pf_use，释放消息通道。
- 结束

5.4 软件指南

5.4.1 典型使用场景初始化

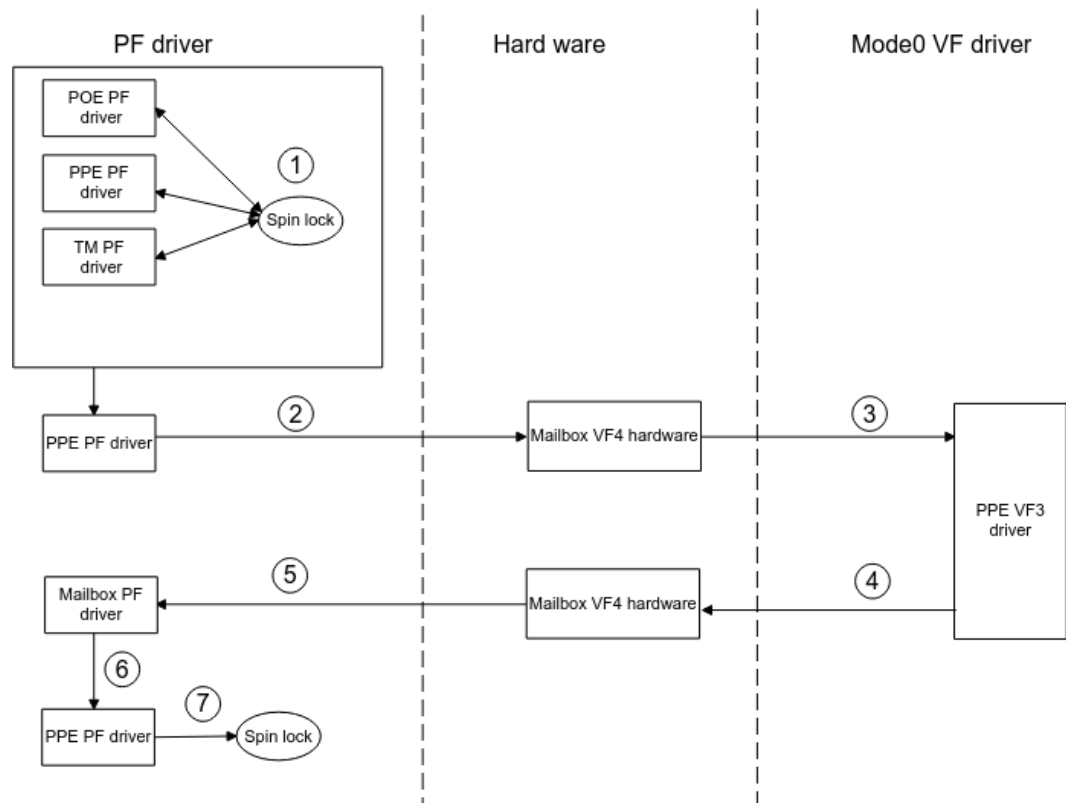
图 5-3 两制式下模块示意图



初始化流程如下：

- 步骤1** 软件加载了2个制式，每个制式下用了POE、PPE模块的1个VF。
- 步骤2** 软件将Mailbox作为一个模块使用，使用了2个VF对应2个制式。
- 步骤3** 加载Mailbox时，将Mailbox的VF空间的页表同时配置给对应制式的其他模块（POE、PPE），这样其他模块驱动的对应该制式可以访问mailbox的对应该制式（例如：将mailbox的VF4的地址映射到POE的VF2、PPE的VF4，这样POE的VF2驱动、PPE的VF4驱动都可以通过访问mailbox的VF4地址向PF发送消息），每个模块的每个VF分配一个MAILBOX VF中断号。
- 步骤4** 每个制式的HAC驱动需要和平台OS通信时，通过使用该制式分配的邮件编号进行通信，如Mode0的POE（VF2）使用邮箱号4（VF4）实现HAC设备驱动与OS底层设备管理驱动之间的通信。
- 步骤5** VF和PF之间的消息内容最大128bit，软件可以自行定义格式。

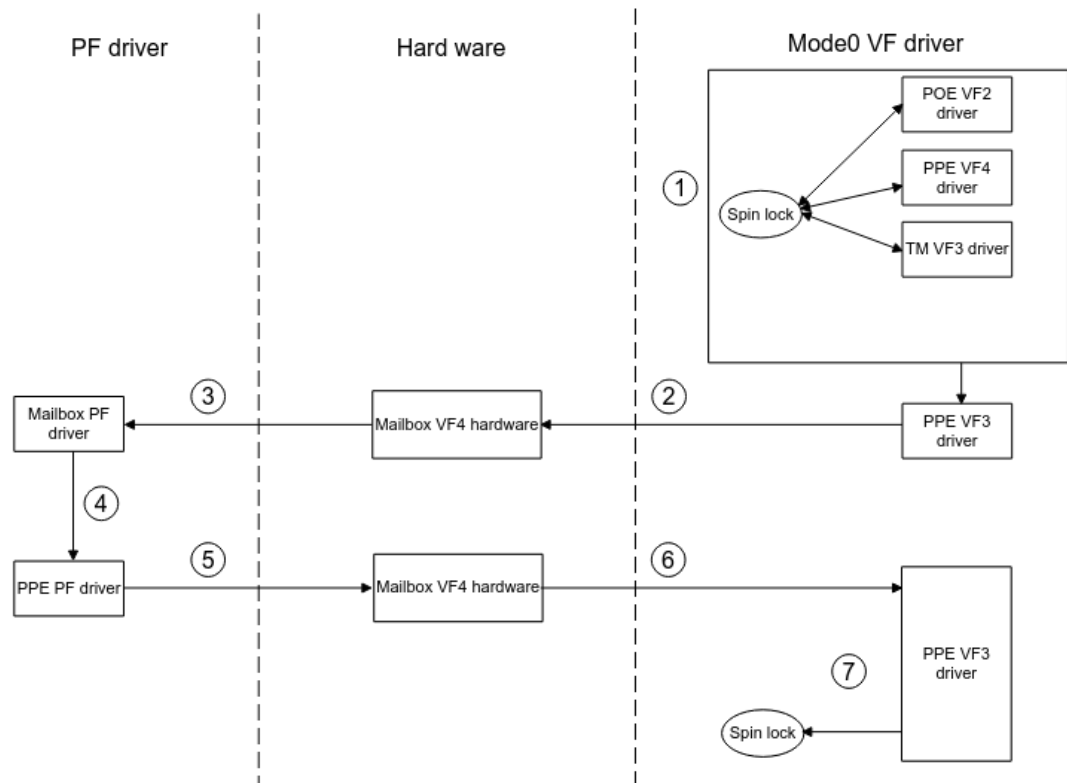
图 5-4 PF 向 VF 发送消息交互图



sRIO的PF需要向它的VF3驱动发送消息：

- 步骤6** POE、PPE的PF驱动同时需要向自己的VF驱动发送消息，使用自旋锁抢锁，PPE的PF驱动得到锁。
- 步骤7** PPE的PF驱动先对硬件MAILBOX VF4进行抢锁（可能此时MAILBOX VF4正在向PF发送消息），抢到锁后，将消息写入MAILBOX VF4中，并触发MAILBOX VF4的编号为PPE的中断。
- 步骤8** PPE 的VF3驱动收到中断，并清除中断。
- 步骤9** PPE 的VF3驱动处理消息并触发MAILBOX的PF中断。
- 步骤10** MAILBOX 的PF驱动收到中断。
- 步骤11** MAILBOX 的PF驱动通知各模块的PF驱动消息已经完成（也可以通过读取MAILBOX的中断信息，查询是VF的哪个hac_id），并清除中断，释放MAILBOX 的VF4的硬件资源（清除pf_use位）。
- 步骤12** PPE的PF驱动释放自旋锁。

图 5-5 VF 向 PF 发送消息交互图



PPE的PF需要向它的VF3驱动发送消息：

- 步骤13** POE、PPE的VF驱动同时需要向自己的PF驱动发送消息，使用自旋锁抢锁，PPE的VF3驱动得到锁。
- 步骤14** PPE的VF3驱动先对硬件MAILBOX VF4进行抢锁（可能此时MAILBOX VF4正在向VF发送消息），抢到锁后，将消息写入MAILBOX VF4中，并触发MAILBOX PF的中断。
- 步骤15** MAILBOX 的PF驱动收到中断。
- 步骤16** MAILBOX 的PF驱动取出消息并中断信息里面的hac id交给PPE 的PF驱动，并清除中断。
- 步骤17** PPE 的PF驱动处理消息并触发MAILBOX VF4的编号为PPE的中断。
- 步骤18** MAILBOX 的VF4驱动收到中断。
- 步骤19** PPE的VF3驱动收到中断，并清除中断，释放MAILBOX 的VF4的硬件资源（清除vf_use位），PPE的VF3驱动释放自旋锁。

----结束



6 PCIE

6.1 特性描述

The following is a list of the features supported by the HiPCIECTRL50 PCIe IP Module. A PCIe Module supports one or more ports that may be configured in one of four modes: Root Port, Endpoint, Switch Upstream Port, and Switch Downstream Port. The features supported by the PCIe Module that are common to all four modes of operation are listed in the first section. Then features for each port type are listed in the following sections.

In all cases, the module is compliant with the PCI Express Base Specification, Revision 5.0.

6.1.1 IP Features

The following table lists the features implemented by the PCIe controller in this project.

表 6-1 PCIe IP Feature List

Feature	Description	Hi309a
IP Version		
PCIe IP Version	Which version is used by project for PCIe IP?	V5.2
SerDes/Phy		
HiLink32T12V101	HiLink 32G Comb PHY。支持多Macro内展频，静态配置。	Y
Link Feature		



Feature	Description	Hi309a
Max Link Speed: GEN5/GEN4/GEN3/GEN2/GEN1 (32GT/s,16GT/s, 8GT/s, 5GT/s, 2.5GT/s)	-	GEN3/GEN2/GEN1
Number of Lanes	total lane number	4x1+1x2+1x4
Number of Port Controllers	That is link number	1, 单port
Bifurcation Options	link mode	4x1+1x2+1x4
Faildown Options(change to another lane when the one fail,and disable the failed one,perfer more bandwidth)	-	x4, x2,x1 and Reversal (some case supported by software)
Port of Core independent	clock an reset 1) port reset is asynchronous reset 2) port reset is synchronous reset	NA
Preset TX Equalization	1) fixed P9; 2)select from P5 or P9	Y (1)
Coefficient TX Equalization	1) fixed P9; 2)select from P5 or P9	Y
lane reverse	not any lane reverse, some restriction; For example 1X4, only support one mode from 0123 reserve to 3210	Y
polarity inversion	-	Y



Feature	Description	Hi309a
Loopback, include: 1) Loopback mode in PCIe spec. (master/slave) 2) internal Loopback mode,	1) PCIe spec Loopback mode 模式下只能发全0的 原始数据 2) internal Loopback mode 内部Loopback模式 可以支持01时钟 码、PRBS7/23/31、 CRPAT、CJPAT、 UDP等码型 具体见 “HiPCIECTRL IP DFx设计 checklist	Y
支持随路参考时钟（common reference clock）	EP feature, EP设备 的参考时钟由RC单 板提供。	Y
支持独立参考时钟（separate clock）	EP/RP feature; 参 考时钟由本地晶振提 供。	Y
支持SRNS（Separate Reference Clocks with No SSC）	(+/- 300ppm) 600ppm	Y
支持SRIS（Separate Reference Clocks with Independent SSC）	(-5000ppm) 5600 ppm More MACRO internal SSC (MMISC)	Y
支持接收Lane Margining（RX lane Margin）	4.0 feature - voltage offset to up/down - timing offset right/left	Y
Support SKP local Data Parity calculate and check at 16GT/s	4.0特性, Register in Physical Layer 16.0 GT/s Extended Capability	NA
Support SKP First retimer Data Parity calculate and check at 16GT/s	4.0特性, Register in Physical Layer 16.0 GT/s Extended Capability	NA



Feature	Description	Hi309a
Support SKP second retimer Data Parity calculate and check at 16GT/s	4.0特性, Register in Physical Layer 16.0 GT/s Extended Capability	NA
Support Jitter Measurement Pattern at 16GT/s	4.0特性	NA
link error process	Link Error Recovery, for example symbol unlock to recovery.	Y
PIPE接口		
PIPE 4.3, 兼容PIPE 4.0	PIPE spec	PIPE5.1
支持8-bits/16-bits/32-bits PIPE位宽静态切换	GEN1/GEN2/GEN4 Fixed 16-bits, GEN3 can change to 8bit/16bit.GEN4 fix 16bit, GEN5 fix 32bit.	Y(Fixed 16-bits)
支持62.5MHz/125MHz/250MHz/500MHz/1GHz PIPE时钟频率动态切换	1) GEN1 16bit 125MHz, 32bit 62.5MHz; 2) GEN2 16bit 250MHz; 3) GEN3 8bit 1GHz,16bit 500MHz 4) GEN4 16bit 1GHz, 5)32bit 500MHz 6) GEN5 32bit 1GHz	1) GEN1 16bit 125MHz; 2) GEN2 16bit 250MHz; 3) GEN3 16bit 500MHz,
Device/Port Type		
支持End Point (EP)	-	N
支持Root Complex (RC)	-	Y
Switch Upstream port	-	N
Switch Downstream port	-	N
Switch routing, whole switch function	-	N
Functional	-	-



Feature	Description	Hi309a
支持Max_Payload_Size为128 Bytes~512 Bytes	协议最大到4096 Bytes	256
支持Max_Request_Size为128 Bytes~4096 Bytes	协议最大到4096 Bytes	512
支持8-bits Tag	-	Y
支持10-bits Tag	REC: support receive and process. SENT: support transmit. MAX 96.	N
Order feature.Relaxed Ordering	RP receive RO command. EP can generate request with RO set.	Y，但是不会传递到系统总线，因为系统总线不支持保序.
Order feature.Strongly Ordering	-	Y，但是系统总线不支持保序，如果需要实现写写保序需要配置AXI Master口的写Outstanding能力为1.
Order feature.ID-Based Ordering（IDO）	-	N
Order feature.Posteds pass Completions/NP	-	Y
Order feature.Completion pass Posteds (violates PCIe)	allow RO completions to pass Posteds in accordance with the spec	Y(only for RO)
Order feature.Reads pass Posteds (violates PCIe)	-	N
支持32-bits/64-bits Memory BAR可配	EP feature,non-prefetchable, prefetchable	N
支持16-bits/32-bits IO BAR	EP feature	N
支持32-bits ROM BAR	EP feature	N
支持INTx中断模式	EP: support send RC: suport INTX interrupt wire	Y（RP）



Feature	Description	Hi309a
支持MSI中断模式	EP支持发送 RP支持接收MSI消息，和MSI Cap变化时本地上报MSI中断	Y（RP）
支持MSI-X中断模式	RP NO EP feature	N
支持DWORD-aligned、QWORD-aligned访问MSI-X Table	EP feature	N
MSI-X Table size for one port	EP feature 1）总Size可配 2）每个function支持的最大Size可配，MAX 1024 Vector。 当无虚拟化时，PF中断向量为1024个； 当存在虚拟化时，所有function（包括PF,VF）平分中断向量，function个数按照2的幂次方计算。	N



Feature	Description	Hi309a
支持本地Pin中断	1) INTA,INTB,INTC,INTD; 2) FHI,ERI, SPI0, SPI1, AP和core的本地错误 NFE/FE/CE/NI 可寄存器配置给 FHI,ERI,SPI0, 或 SPI1; 3) SPI2, SPI3, 两个MCTP中断 (MCTP1_RP, MCPT2_EP0)可寄存器配置给SPI2, SPI3。 4) DMA, VIRTIO and NVME use MSI of IEP. 也可寄存器配置给SPI0 或SPI1。 5) 在RC时, 收到的MSI可以选择上总线, 或配置到SPI0/SPI1。	Y 1) INTA,INTB,INTC,INTD; 2) FHI,ERI, SPI0, SPI1, AP和core的本地错误 NFE/FE/CE/NI 可寄存器配置给 FHI,ERI,SPI0, 或 SPI1; 3) RC时, 收到的MSI选择配置到SPI0/SPI1。
支持2 VC (Virtual Channel)	VC0 and CCIX VC	1
支持8 TC (Traffic Class)	最大可实现8 TC	8
支持Inbound Memory访问	1) Receive TLP , send to axi master interface. 2)Receive TLP, send to native interface. 3) Receive CCIX TLP	Y
支持Outbound Memory访问	1)CHI slave receive data from system, sent to remote PCIE component. 2)DMA, MCTP TLP is sent to remote PCIE component. 3) Native send TLP to remote PCIE component. 4) sent CCIX TLP to remote.	Y



Feature	Description	Hi309a
支持 RX ATU	parameter, max number 34 仅仅在EP模式下使用，将PCIE域地址转换为本地系统地址。	N
支持TX ATU	parameter,max number 16 is ok。 不可参数化。EP 保留12个作为DFX。 RP 4个，通常是PR使用。 TXATU至少需要一个64bit Prefetchable Memory, 32bit Prefetchable memory, 32bit Non-Prefetchable memory, 和一个IO空间，就是4个。 CFG命令的转换是用ECAM。	Y(4) RP, TXATU至少需要一个64bit Memory, 32bit memory和一个IO空间，就是3个，按4个实现便于逻辑实现。每个TX ATU可以独立配置三种类型的操作空间。
支持ECRC产生/校验	默认关闭	Y
ECRC Forwarding	RP feature. When peer2peer, ECRC bit forwarding SWTICH feature support	N
Poison Bit Handling (Forwarding)	forward to upstream, set poison bit in system BUS or drop TLP	N（当前总线都支持不了中毒，所以报文需要在PCIE丢弃，并报NONFATAL错误。但会正常进行总线反馈，保证总线正常。）
Role Based Error Reporting	Base error Reporing in PCIe Spec.	Y
Surprise Down Error Reporting	RP dectect pin. EP linkdown interrupt.	Y
支持Peer-to-Peer传输	all RP peer2peer, maximum support 4 chip peer2peer。	N



Feature	Description	Hi309a
支持ECAM配置访问	RP特性,fixed 256M大小	Y(作为RC访问下面的ECAM地址范围4M)
支持通知式热插拔， standard hotplug	RP feature.	N
支持暴力热插拔,Async Removal	RP feature.逻辑实现自动link error recovery 支持OOB（ Out-Of-Band ）， 不支持IB(In-Band)。	Y (暴力热插拔， 只要保证CPU不挂死就行。暴力热插拔后， port复位后初始化再使用。)
EP feature. standard hotplug,surprise hotplug, only linkdown interrupt	-	N
Inband Surprise Hotplug Removal	RP feature. Only detect the detect pin, depend on 50 Ω	N
DL Link Active Reporting	local interrupt in DL. DL_UP, DL_DOWN	Y
支持CCIX	只支持PCIE模式， 不支持CCIX ESM模式	N
支持协议复位， 包括Hot/Warm/Cold Reset	-	Y
ACK发送机制可配	支持Timer mode和Counter mode	Y
支持Update FC DLLP发送机制可配	A、支持Timer mode， 支持Counter mode B、支持Timer mode， 不支持Counter mode 注： Timer mode ， 寄存器为0时， 根据协议时间更新； 寄存器非0时， 按照寄存器时间更新。 Counter mode: TL提供上下水线， 上水线立即更新， 下水线更新频率降低。	A



Feature	Description	Hi309a
支持Data Link Feature Exchange	4.0特性	N
支持Scaled Flow Control	4.0特性	N
支持协议定义的Completion Timeout机制	completion timeout, (RANGE A/B/C, not supported D)	Y
支持Request Timeout机制	protect local CPU not died out	Y
Atomic Operations	RP receive. EP transmit.	N
Atomic feature. Completer Capability	All EP, RP feature	N
Atomic feature. Routing Capability	1) the support for peer-to-peer routing of AtomicOp Requests and Completions between Root Ports 2) switch support	N
Atomic feature. Requestor Capability	All EP, RP feature	N
Atomic feature. FetchAdd (32b and 64b)	All EP, RP feature	N
Atomic feature. Swap (32b and 64b)	All EP, RP feature	N
Atomic feature. Compare and Swap (32bit, 64bit, and 128bit)	All EP, RP feature	N
Link Speed Management	MAX link speed	Y
Link Bandwidth Changed	one link max lane number,16	N
Dynamic Link Width increased	lane number of one link increased dynamicly.	N
Beacon	SerDes receive beacon and wake up.	N
PF number pre Port	EP feature. max 4 PF in one port.	1



Feature	Description	Hi309a
VF number pre Port	EP feature. max 254 VF in one port EP feature. max 254 VF in one port PF0 supports up to 127 VFs; PF1 supports up to 63 VFs; PF0 + VFs in PF0 + PF1 +VFs in PF1 <=128; PF2 supports up to 127 VFs; PF3 supports up to 63 VFs; PF2 + VFs in PF2 + PF3 +VFs in PF3 <=128;	0
Phantom Functions	EP Feature;To increase the maximum possible number of outstanding Requests.	N
FEC	FEC function (GEN6以上需求)	N
支持CXL	1630特性，HOST特性支持，为了验证也同时支持Device侧功能。	N
Engine function		
DMA feature.support DMA Engine	-	N
DMA feature.maximum channel number configurable	通道个数可配置	N
DMA feature. maximum remote NP outstanding number	更改为“outstanding个数可配置, sent to remote	N
DMA feature.支持pcie port、dma channel绑定关系静态配置(Support relax bound between PCIe Port and the DMA Channel by local register.)	EP feature.可根据需要将dam channel绑定到不同pcie port	N



Feature	Description	Hi309a
DMA feature.support local to remote mode	EP feature,write data to remote.	N
DMA feature.support remote to local mode	EP feature, read data from remote host	N
DMA feature.support local to local mode	RP feature, access local address through Axi master interface	N
DMA feature.support 1Byte~4GByte size transmit	-	N
DMA feature. Support 32/64bits source address and target address	-	N
DMA feature.support DMA channel soft reset independently	Each channel have independent	N
DMA feature. Support whole DMA soft reset	The soft reset can reset whole DMA block.	N
DMA feature. Host configure DMA	EP feature. Remote host can initiate DMA	N
DMA feature. Support DMA iEP	-	N
DMA. Feature. Support Atomic	support transmit in both RP and EP?	N
MCTP feature	Transmit or receive between DDR and PCIe remote	N
MCTP feature. Support MCTP Engine Number	maximum 2	N
MCTP feature. Support MCTP VDM Transmit	MCTP VDM Tx	N
MCTP feature. Support MCTP VDM Receive	MCTP VDM Rx	N
MCTP feature. Support Other VDM Transmit	Any type of vendor defined legal VDM message can be sent out	N



Feature	Description	Hi309a
MCTP feature. Support Other VDM Receive	The VDM can't be filtered will route to the MCTP. Except for Page Request Message/ Invalidate Completion Message.	N
NVME feature.	Just for EP feature	N
NVME feature. Support hardware transfer the SQE	Use private DMA channel transfer SQE	N
NVME feature. Support hardware integrated doorbell register	The register address above 0x1000. These doorbell registers are located in the PCIe IP.	N
NVME feature. Support hardware integrated control register.	The register address from 0~0xFFF. These control registers are located in the PCIe IP.	N
NVME feature. Support maximum queues.	max support PF0, PF1, PF0 support 128 queue, PF0+PF0的VF平分128个队列; PF1 support 128 queue., PF1+PF1的VF平分128个队列; Total 256 queue.	N
NVME feature. Support NVMe iEP	in PCIe the IEP name is SDI	N
Virtio feature.	max support 1024 queues. Support PF0, PF1, PF3,PF4. 每个PF的队列个数只是配置。PF下VF的队列个数平分PF分配的队列个数。	N



Feature	Description	Hi309a
Virtio feature. NVME feature. Support hardware transfer the SQE	Use private DMA channel transfer SQE	N
Virtio feature. Support VirtIO Spec 1.0&0.9x	-	N
Virtio feature. Support maximum queue number	maximum 1024	N
Virtio feature. Support VirtIO iEP	-	N
SDI	Network Through Feature,网卡直通业 务，网卡直接通过 CHISLAVE口发送 TLP到远端。	N
IEP for DFX	给软件开发DFX定位 工具，软件当前需要 一个root权限才能访 问。 硬件增加的IEP，可 以通过APB配置 CFGSPACE的vendid 为全F，让软件枚举 不到。	N
Capability		
PCI_CFG	PCI-Compatible Configuration Registers	Y
PCIE_CFG	-	Y
MSI	EP是否支持该特性 RP支持接收MSI消 息，和MSI Cap变化 时本地上报MSI中断	Y（RC支持）
MSI-X	RP NO EP feature,	N
Single Root I/O Virtualization (SR- IOV)	-	N
Multi-Root I/O Virtualization (MR-IOV)	-	N
PCI Express Advanced Error Reporting (AER) for PF	with one Header Logg	Y



Feature	Description	Hi309a
PCI Express Advanced Error Reporting (AER) for vf	EP feature. VF AER.	N
Subsystem ID and Subsystem Vendor ID (SSID/SSVID)	EP（包括在 PCI_CFG，type0寄存器中） RP：有独立的SSID Capability寄存器	Y
Alternative Routing-ID Interpretation (ARI)	RP：支持处理接收的ARI报文的能力。 EP：支持发送ARI报文。	Y
Address Translating Services (ATS) (only RC function, not capability)	RP feature. Root Complex is able to receive Address Translation Requests and send Address Translation Invalidation Requests	N
Page Request Services (PRS) (only RC function, not capability)	RP feature. Able to receive Page Request Messages so I/O Paging may be supported	N
Page Request Interface (PRI)	EP feature. Able to generate Page Request Messages RP feature: can translation address from Smmu.	N
Address Translation Services (ATS) Extended Capability Structure	EP feature. Endpoint implements an Address Translation Cache (ATC) and Capability Structure	N
Address Translation Cache (only function, not capability)	EP feature. ATC	N
Completion Timeout Ranges	Device Control 2 Register 中的域。 (Range A/B/C, not supported D)	Y



Feature	Description	Hi309a
Configuration Request Retry Status (CRS)	EP response CPL with CRS. RP receive CPL with CRS and retry.	Y (RP支持接收CRS CPL，并处理，具体处理参看AP FS)
Function Level Reset (FLR)	EP Feature.	N
TLP Processing Hints (TPH) Requester Capability	EP Feature	N
TPH Completer Capability	RP Feature.yes, but no support for OS - ST used to determine the L2 cache destination, not L3	N
Secondary PCIE (SEC_PCIE) Capability	-	Y
Device Serial Number (DSN) Capability	-	Y
PASID TLP Prefix receive	RC Receive TLP has TLP Prefix	N
PASID TLP Prefix Extended Capability	EP transmit TLP has TLP Prefix	N
Extended Tag, 10-bits Tag	4.0特性	N
Lightweight Notifications Completer (LNC)	RP feature，收到EP的LNR后，返回LNC。LN 协议，是EP可以获取host中特定cacheline是否更新。	N
LNR Extended Capability	EP feature，EP向RP发出。	N
Readiness Notifications (RN)	Including FRS and DRS。EP主动通知RC是否ready。	N
Function Readiness Status (FRS) Queuing Capability	a part of RN	N
Readiness Time Reporting Capability	a part of RN	N
Vital Product Data (VPD)	EP feature.	N



Feature	Description	Hi309a
PCI Express Power Management (PME)	<ul style="list-style-type: none">- 支持device D-states, D0/D3 (no work)- 支持ASPM的L0s、L1 (No L2)- 支持PCI-PM的L1- 支持PM_PME message	Y
D-States Supported (PCI-PM)	D0, D3hot, D3cold (no wake)	Y
L-States Supported	L0, L0s, L1, L3	Y (L0, L1)
ASPM Supported	硬件自动进行功耗管理	Y
PM-PME Supported	PM_PME Messages, to wake up power state	Y
PCI-PM	需要本地CPU介入进行低功耗管理	Y
Power Budgeting	功耗预算，此寄存器给RP看到EP消耗的功耗。	N
Dynamic Power Allocation (DPA)	动态给EP设备分配功耗	N
Latency Tolerance Reporting (LTR)	定位device容忍的memory读写的延时时间。	Y
L1 Substates (L1SS)	L1 PM Substates Capability	N
Slot Power Limit Messaging	Slot Power Limit Control需要系统软件配合，软件配置RP的slot寄存器，RP发送message给EP，EP捕获功耗值写入Device Capabilities Register，EP工作的功耗不能超过这个值。	N
Emergency Power Reduction	Device Capabilities 2 Register，如果支持，EP设备可以紧急降低功耗。	N



Feature	Description	Hi309a
Optimized Buffer Fill and Flush (OBBF)	RP给EP的降低功耗的时间窗口	N
Precision Time Measurement (PTM)	时间同步机制	N
Access Control Services (ACS)	RP特性, EP 特性	Y
Downstream Port Containment (DPC)	RP特性, SWITCH 特性	N
Enhanced DPC (eDPC) , ECR	RP特性, SWITCH 特性, PCIE5.0 Spec, have included this ECR	N
Data Link Feature Extended (DLF)	4.0特性, 扩展credit位宽。	N
RXMARGIN (Lane Margining at the Reciever Capability)	4.0特性	N
Designated Vendor-Specific Extended Capability (DVSEC)	4.0特性, transport CCIX capability, It is used by CCIX.	N
Physical Layer 16.0 GT/s Extended Capability	4.0特性	N
Physical Layer 32.0 GT/s Extended Capability	5.0 feature,	N
Enhanced Allocation Capability	The Enhanced Allocation (EA) Capability is an optional Capability that allows the allocation of I/O, Memory and Bus Number resources in ways not possible with the BAR and Base/Limit 10 mechanisms in the Type 0 and Type 1 Configuration Headers.	N



Feature	Description	Hi309a
Resizable BAR Extended Capability	allows hardware to communicate resource sizes, and system software, after determining the optimal size, to communicate this optimal size back to the hardware. 定义BAR size。	N
Virtual Channel Capability	CCIX VC and VC0, CCIX VC need Virtual Channel Capability	N
Multi-Function Virtual Channel Capability	Multi-Function Virtual Channel Extended Capability (MFVC), permits enhanced QoS management in a Multi-Function Device	N
Multicast Capability	广播	N
Root Complex Link Declaration Extended Capability	to be implemented by Root Ports, RCiEPs, or RCRBs to declare a Root Complex's internal topology.	N
Root Complex Internal Link Control Extended Capability	controls an internal Root Complex Link between two distinct Root Complex Components	N
Root Complex Event Collector Endpoint Association Extended Capability	is implemented by Root Complex Event Collectors	N
Vendor-Specific Extended Capability	This Capability is used by Virtio.	N



Feature	Description	Hi309a
RCRB Header Extended Capability	The PCI Express RCRB Header Capability is an optional Extended Capability that may be implemented in an RCRB to provide a Vendor ID and Device ID for the RCRB and to permit the management of parameters that affect the behavior of Root Complex functionality associated with the RCRB.	N
Hierarchy ID Extended Capability	The Hierarchy ID Extended Capability provides an optional mechanism for passing a unique identifier to Functions within a Hierarchy.	N
Flattening Portal Bridge (FPB) Capability	from Eric, including the FPB function	N
Native PCIe Enclosure Management Extended Capability (NPEM)	provide management for enclosures containing PCIe SSDs that is consistent with the established capabilities in the storage ecosystem。 From Alex	N
Designated Vendor-Specific Extended Capability (DVSEC)	Gen5.0版本， for CXL function。	N
Designated Vendor-Specific Extended Capability (DVSEC)	预留的capability, dummy 的 Capability	N
Interface		



Feature	Description	Hi309a
支持Native私有接口，及数量可配	TLP接口，用于外部DMA等应用	N
支持APB 3.0 V2.0 Slave接口	AMBA总线接口，本地所有寄存器访问	Y
APB.地址位宽	IP接口位宽 (内部实际使用)	32-bits
APB.数据位宽	-	32-bits
支持CHI Slave接口	AMBA总线接口，用于Outbound Memory访问	N
地址位宽	-	NA
数据位宽	-	NA
max receive outstanding of read/write	96 header entry, 4kByte data Buffer. Read and write share these resources. MAX outstanding is 96.	NA
最大数据size	64Byte (CPU), 256Byte (NIC直通)	NA
最大ID位宽，transaction ID，TXNID	8-bits	NA
支持AXI Slave接口	AMBA总线接口，用于CPU及片上设备访问PCIE	Y
地址位宽	IP接口位宽 使用位宽	64-bits
数据位宽	-	128-bits
最大outstanding读/写能力	-	8/8
最大burst长度	读长度//写长度	16/16 (Len=1~15默认可支持)
最大ID位宽	-	16
AXI总线版本	选择是AXI3还是AXI4总线。暂时只能支持AXI4.0。	AXI4



Feature	Description	Hi309a
支持AXI Master接口	AMBA总线接口，用于DMA、Inbound Memory访问	Y
地址位宽	64-bits（有效位宽48bit）	64
数据位宽	-	128
最大outstanding读/写能力	Inbound最大能力 读能力/写能力，引擎（主要是EP DMA）最大能力（读能力/写能力）	（读单独16） （写单独16） 备注：总线延时按照1μs评估性能。
最大burst长度	读长度/写长度	8/8
最大ID位宽	当前参数化可以支持ID位宽可配置。	8-bits
AXI总线版本	选择是AXI3还是AXI4总线	AXI4
Frequency，频率要求	-	-
AXI clock，AP工作时钟	PCIE IP MAX 1GHz.	400MHz
APB clock，APB口工作时钟	usually 250MHz	400MHz
Core clock，PCIE core工作时钟	Max 1GHZ	500MHz，使用SerDes提供的时钟
Pcie aux clock, aux 时钟，初始化PCIE使用	usually 200MHz，250MHz	200MHz
PCS aux clock, pcs aux 时钟，低功耗使用	usually 100MHz，MAX 1GH	100MHz
asynchronous process position	1) between AP and TL; 2) Between TL and DL. 3) between PCS and SerDes; 4) between mac and MAC_MUX	1) between AP and TL;
Technology	12nm	T12
性能:latency		
GEN4: latency（include the latency of Port Tx, Port Rx and SerDes）	CPU access	同PCIE50，100ns内
Outband bandwidth	-	-
Inband AXI master bandwidth	-	-



Feature	Description	Hi309a
Inband native interface bandwidth	-	NA
P2P bandwidth	-	NA
EP DMA bandwidth	-	NA
Virtio bandwidth	-	NA
NVME bandwidth	-	NA
DFX	-	DFX
系统配合		
支持PERST# pin复位	Chip need to implement function. Detect this pin and generate interrupt to local CPU. Local CPU configure PCIe's soft reset to reset PCIe. 芯片CRG实现，用于整个PCIE的复位、初始化启动	NA
Function Level Reset (FLR)	EP feature. When EP receive FLR CFG command, PCIe will report interrupt to local CPU. Local CPU need to stop business for those function, and then local CPU reset function register.	N
WAKE#	Pin detect	N
Volume Management Device (VMD)	-	N
RAS (software platform),	-	N
Fast OOB Hotplug	Fast Out-of-band Hotplug (Presence Detect pin in less than 50uS)	N
Performance Monitors, 性能监控, 性能调优。	-	N
CLKREQ	-	N



Feature	Description	Hi309a
建链时间，从控制器初始化、SerDes 初始化，到建链到Gen5速率的时间	-	NA
IODC（IO data direct write cache）	RP特性，IO设备发送数据，CPU收到数据后，直接写到cache中。CPU构造数据放到cache中，IO设备直接从cache读数据。	N
Inbound的Write Merge功能	和SMMU配合，对Mmemory write操作，实现merge功能。	N
ECR		
ATS Memory Attributes	-	N
ACS Enhanced Capability	-	N
QTI PCIe Security ECR	-	N



7 SATA

7.1 Key Terms

The terms in [表7-1](#) have special meanings in this section.

表 7-1 Description of SATA key terms

Terms	Description
SATAH	SATA Host controller.
AHCI	Serial ATA Advanced Host Controller Interface.
ATAPI	AT Attachment Packet Interface.
CCC	Command Completion Coalescing
D2H	D2H is an acronym for “device to HBA” . In the Serial ATA specifications, this acronym stands similarly for “device to host” . D2H is often used to describe the direction of transfer for a FIS.
H2D	H2D is an acronym for “HBA to device” . In the Serial ATA specifications, this acronym stands similarly for “host to device” . H2D is often used to describe the direction of transfer for a FIS.
FIS	FIS is an acronym for “Frame Information Structure” . A FIS is a packet or frame of information that is transferred between the host and device. Refer to the Serial ATA 1.0a specification for more information.
FBS	FIS-Based Switching.
FPDMA	First-Party DMA Access.



Terms	Description
HBA	HBA is an acronym for “host bus adapter” . A host bus adapter refers to the silicon that implements the AHCI specification to communicate between system memory and Serial ATA devices.
NCQ	Native Command Queuing.
PIO	Programmed Input/Output.

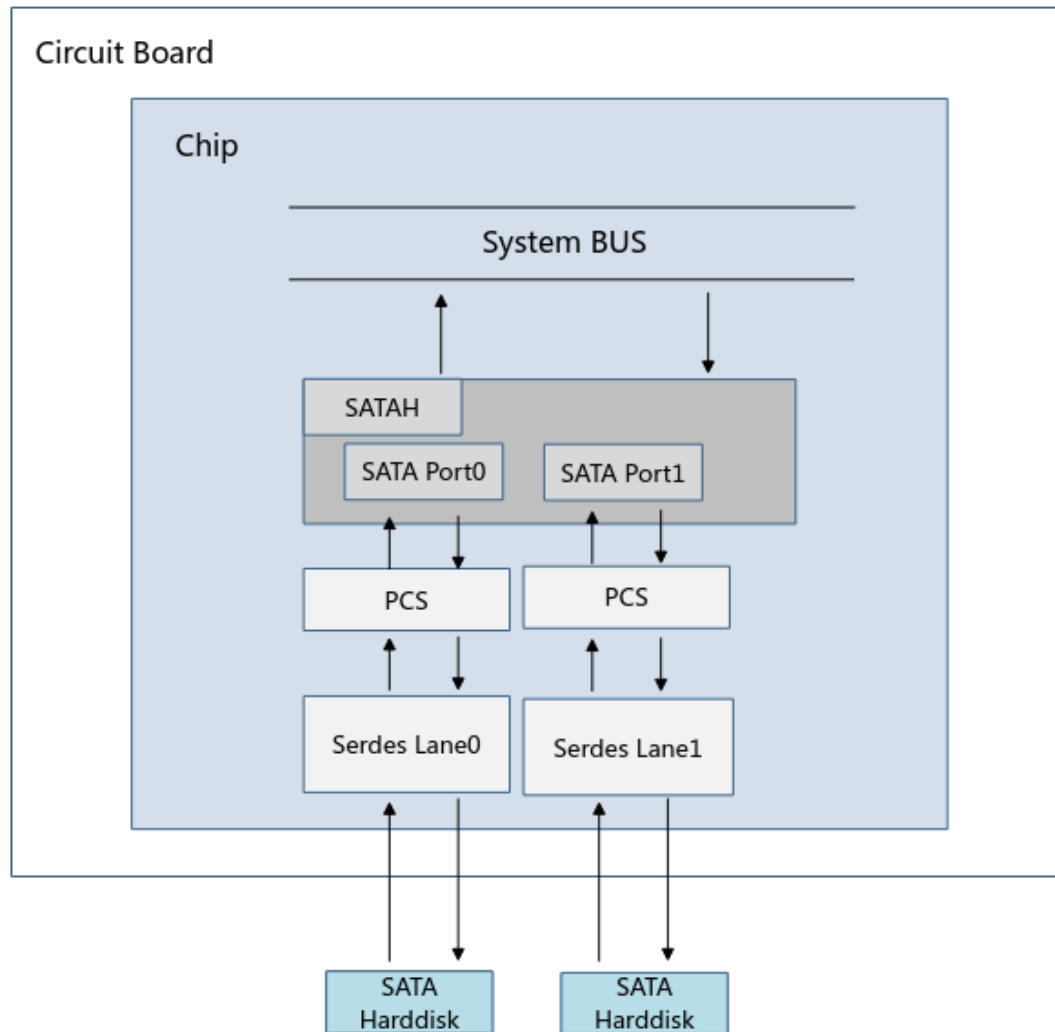
7.2 Overview

SATAH(SATA Host) is a SATA controller, comply with the protocol of :

- comply with “Serial ATA International Organization: Serial ATA Revision 3.0” and “Serial ATA Advanced Host Controller Interface (AHCI) 1.3” .
- downward compatible with “Serial ATA International Organization: Serial ATA Revision 2.6” and “Serial ATA Advanced Host Controller Interface (AHCI) 1.2” .

The system architecture of SATAH is as follows :

图 7-1 The System Architecture of SATAH



The system implement SATA Controller as one SATA HBA with four SATA PORTs.

7.3 Features

SATAH's feature list is as follows:

- Supports SATA 3.0 and AHCI 1.3 protocol, down compatible with SATA 2.6 and AHCI 1.2 protocol.
 - Supports PIO, DMA , NCQ operation.
 - Automatic negotiation for 1.5Gbps, 3.0Gbps and 6.0Gbps rates.
 - Supports 2 SATA Ports under a single SATA HBA.
- Supports 64 bits address. 128 bits data
- Support Logic DFX debugging features.
 - Supports RAM ECC inject and report interface.



说明

- SATAH does not support the RAID disk array.

7.4 Compliance

SATA controller compatible or downward compatible with the following protocol:

- Serial ATA International Organization: Serial ATA Revision 3.0/2.6
- Serial ATA Advanced Host Controller Interface (AHCI) 1.3/1.2

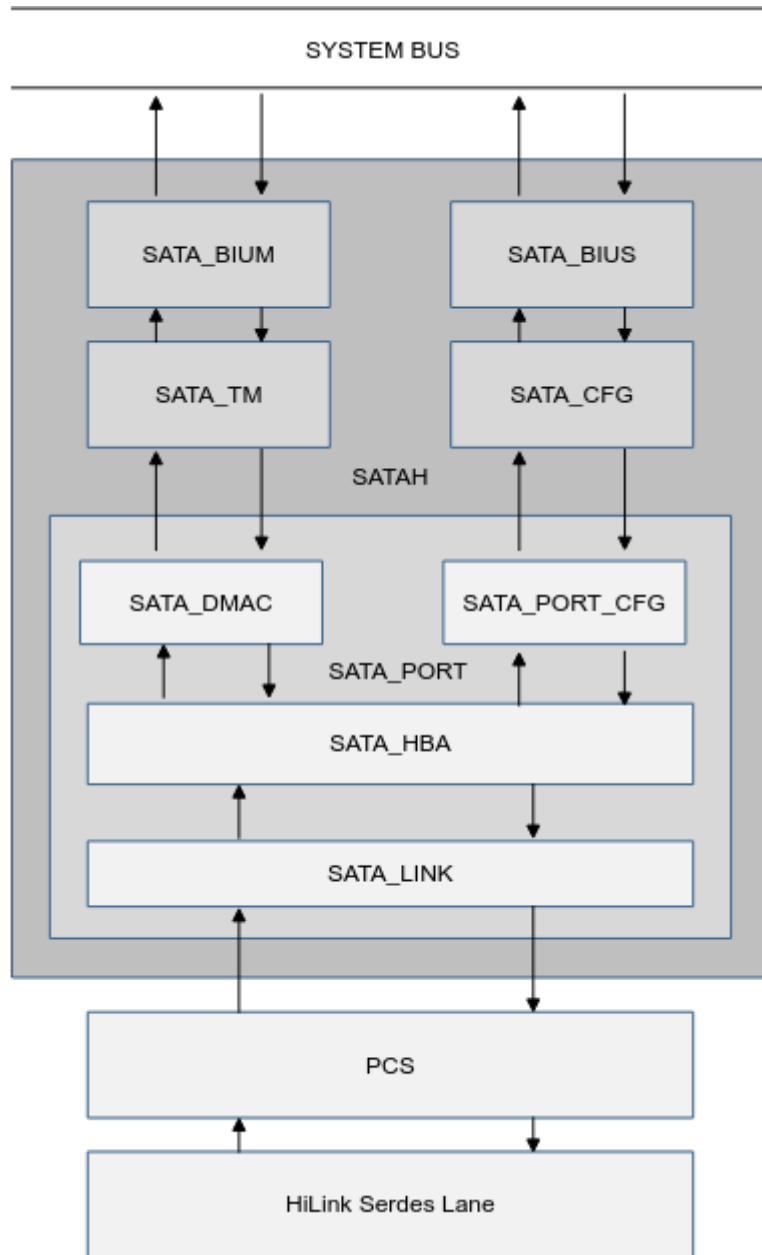
说明

- When the PxIS.OFS interrupt is generated, if the driver clears the PxCMD.ST and receives FIS from the harddisk, an exception occurs. The symptom is that after the next link FIS cannot be sent. However, this exception will be automatically cleared after the OOB is restarted next time.
- SATAH have two SATA PORT, when using SATA, ensure that the clocks of SATA PORT0 is enabled, otherwise, an exception occurs when do the HBA reset.

7.5 Function Description

7.5.1 Logic Block Diagram

图 7-2 SATAH Structure Diagram



The functions of sub module implementation are as follows:

- **SATA_PORT**: This module implements all functions of the SATA Port as specification defined, including: configurable port registers implementation, DMA control, AHCI protocol description of SATA main controller, SATA link layer and the transport layer protocol. One SATAH can be instant of up to 4 **SATA_PORT**.
- **SATA_BIUM**: The module implements the Master AXI interface function.
- **SATA_BIUS**: This module implements the interface function of PCI device.



- SATA_TM: This module implements the Master AXI interface for different SATA_PORT access to complete the arbitration.
- SATA_CFG: The module implements the access to the global register through the SATA_BIUS interface, and provides access to the SATA_PORT_CFG register interface. And the global CCC function is implemented in this module.
- The function of the sub module of the SATA_PORT module is described as follows:
 - SATA_DMAC implement send and receive DMA function, including data sending and receiving FIFO control.
 - SATA_HBA implement the state machine and related control logic, and includes the SATA protocol described in the transport layer function.
 - SATA_LINK to achieve the functions of LINK layer in SATA protocol.
 - SATA_PORT_CFG implement the SATA Port registers.

7.5.2 Device Custom Content

7.5.2.1 Interrupt Mapping

SATA Controller implements “Pin based interrupts”, it reports to system.

表 7-2 SPI Interrupt Mapping Table

SPI ID	Interrupt Name	Interrupt Function Description
531	sata_int	SATA global interruption.

7.6 Programming Interface

SATA controller using standard AHCI 1.3programming interface, detailed see protocol “Serial ATA Advanced Host Controller Interface (AHCI) 1.3”。

说明

As protocol description(10.4.3 HBA Reset in AHCI 1.3 : If the HBA becomes unusable for multiple ports, and a software reset or port reset does not correct the problem, software may reset the entire HBA by setting GHC.HR to ‘1’), software users should not random invoke HBA reset interface because it may case ship problem when command is running, actual one should try softreset reset (10.4.1 in AHCI 1.3) or port reset (10.4.2 in AHCI 1.3) first.

7.7 Application Configurations

7.7.1 SATA Data Structure

SATA controller use the data structure to follow the AHCI 1.3protocol, detailed see protocol “Serial ATA Advanced Host Controller Interface (AHCI) 1.3”。

This table indicates basic capabilities of the HBA to driver software.



表 7-3 Generic Host Control Offset 00h: CAP – HBA Capabilities

Bit	Type	Reset	Description
31	RO	0x1	Supports 64-bit Addressing (S64A)
30	RO	0x1	Supports Native Command Queuing (SNCQ)
29	RO	0x1	Supports SNotification Register (SSNTF)
28	RO	0x0	Not Supports Mechanical Presence Switch (SMPS)
27	RO	0x1	Supports Staggered Spin-up (SSS)
26	RO	0x1	Supports Aggressive Link Power Management (SALP)
25	RO	0x1	Supports Activity LED (SAL)
24	RO	0x1	Supports Command List Override (SCLO)
23:20	RO	0x3	Interface Speed Support (ISS)
19	RO	0x0	Reserved
18	RO	0x1	Supports AHCI mode only (SAM)
17	RO	0x1	Supports Port Multiplier (SPM)
16	RO	0x1	FIS-based Switching Supported (FBSS)
15	RO	0x1	PIO Multiple DRQ Block (PMD)
14	RO	0x1	Slumber State Capable (SSC)
13	RO	0x1	Partial State Capable (PSC)
12:8	RO	0x1F	Number of Command Slots (NCS)
7	RO	0x1	Command Completion Coalescing Supported (CCCS)
6	RO	0x0	Enclosure Management is not Supported (EMS)
5	RO	0x1	Supports External SATA (SXS)
4:0	RO	0x1	Number of Ports (NP)



7.8 Initialization Example

7.8.1 Initialization Configuration Process

表7-4 describes the clock reset control registers corresponding to the SATA controllers (for details about the register functions, see the description of the corresponding subsystem registers).

表 7-4 SATA controllers and corresponding system CRG registers

Device ID	Register	Offset	Brief Function Description
SATA controller	SC_SATA_ICG_EN	0x468	SATA clock gating signal register
	SC_SATA_ICG_DIS	0x46C	SATA clock gating signal register
	SC_SATA_RESET_REQ	0xB68	SATA warm reset request register
	SC_SATA_RESET_DREQ	0xB6C	SATA warm reset deassertion request register
	SC_MEM_CTRL_SATA	0x3008	SATA memory control register
	SC_SATA_ICG_ST	0x5468	SATA clock gating signal status register
	SC_SATA_RESET_ST	0x5B68	SATA warm reset request status register

- To warm-reset the clock when the SATA controller is working, ensure that bus transmission is stopped. There is strict requirement on the sequence of the clock resets.
- BASE_ADDR of SPPE_SUBCTRL is 0x000_D885_0000.

For example, to enable the SATA controller clock to enter the working status, perform the following steps:

- 步骤1** Set SPPE_SUBCTRL [SC_SATA_RESET_REQ] (offset address: 0xB68) to 0x3F to warm-reset all the clocks.
- 步骤2** Query SPPE_SUBCTRL [SC_SATA_RESET_ST] until the value is 0x3F, indicating that all the clocks are reset.
- 步骤3** Set SPPE_SUBCTRL [SC_SATA_ICG_DIS] (offset address: 0x46C) to 0x3F to disable all the clocks.



- 步骤4** Query SPPE_SUBCTRL [SC_SATA_ICG_ST] until the value is 0x0, indicating that all the clocks are disabled.
- 步骤5** Set SPPE_SUBCTRL [SC_SATA_RESET_DREQ] (offset address: 0xB6C) to 0x3F to deassert the warm-reset on all the clocks.
- 步骤6** Set SPPE_SUBCTRL [SC_SATA_ICG_EN] (offset address: 0x468) to 0x3F to enable all the clocks.
- 步骤7** Query SPPE_SUBCTRL [SC_SATA_ICG_ST] and SPPE_SUBCTRL [SC_SATA_RESET_ST] to check whether the clocks and reset status are as expected.

----结束

7.8.2 LED light status

Each SATA Port provides one bit Activity lighting interface, the meaning is described as follows:

It will pulled up after the connection is established, but the following conditions will be low:

- When device is not ATAPI device and have commands is running .
- When device is ATAPI device and configure allow light in ATAPI mode and have commands is running。



8 SerDes Ctrl

8.1 概述

为了满足产品多种应用场景，并实现有竞争力的成本和功耗，芯片提供了灵活的片上高速接口SerDes组网控制模块，可以根据不同产品应用场景，实现网口（XGE/GE/QSGMII）、SATA接口、PCIE等SerDes Lane的灵活组网。

一个SerDes单元（Macro）包含多个SerDes Lane，同一个SerDes单元上的所有Lane最多只能工作在两个数据速率。

SerDes可以按Lane为单位使能和关闭，不要求同一个SerDes单元上的所有Lane同时使能。

8.2 关键术语

如表8-1的术语将在本章节表示特殊的含义。

表 8-1 关键术语描述

术语	描述
SerDes Lane	一个SerDes Lane有两对差分信号，一对差分信号接受数据，一对差分信号发送数据。
SGMII	Serial Gigabit Media Independent Interface，一种GE网口标准。
XAUI	10 Gigabit Attachment Unit Interface，一种10GE网口标准。
XGE	Ten-GigabitEthernet，一种10GE网口标准。
QSGMII	Quad Serial Gigabit Media Independent Interface，一种GE合并的网口标准，4个MAC合并后出一路5G的数据。
Macro	一个SerDes单元，包含多个SerDes Lane。



8.3 功能特性

支持的主要特性如下：

- Hi309a提供6条SerDes Lane，分布在2个Macro中。Hi309a提供10条SerDes Lane，分布在3个Macro中；具体对应方式可以参考《Hi309a SerDes复用关系》。
- 同一个Macro上支持多种速率模式，同时最多支持两种不同频点共存，具体共存方式可参考《Hi309a SerDes复用关系》。
- Hi309a支持总共10路SerDes，可复用为以下模式：
 - 支持10路以太网接口，其中4路支持XGE（BASE-R/BASE-R with BR FEC）。
 - 支持6路QSGMII。
 - 支持2路SATA。
 - 支持8路PCIE@GEN3，支持8*X1和2*X2和1*X4。
- SerDes需要芯片外部提供两路时钟频率分别为156.25MHz和100MHz的差分参考时钟。参考时钟选择如下：
 - GE SerDes参考时钟：156.25/100MHz差分时钟输入。
 - XGE SerDes参考时钟：156.25MHz差分时钟输入。
 - QSGMII SerDes参考时钟：156.25MHz/100MHz差分时钟输入。
 - SATA SerDes参考时钟：100MHz差分时钟输入。
 - PCIe SerDes参考时钟：100MHz差分时钟输入。
- 支持按Macro和Lane为单位进行掉电控制。
- SerDes 复用控制以Lane为单位，灵活实现各种不同的需求场景。

8.4 遵循协议

无

8.5 功能描述

8.5.1 相关模块

在系统中，SerDes Ctrl模块一端会直接接出芯片管脚，与外部芯片对接；一端会和多个模块进行交互，包括SATA、PCIe、GE、XGE、QSGMII。

8.5.2 Macro

一个SerDes单元（Macro）包含多个SerDes Lane，每个Macro包含两个PLL，同一个SerDes单元上的所有Lane最多工作在两个数据速率。

📖 说明

当Hilink工作在100BASE-FX接口模式下，数字眼图与眼图与发送的码型强相关，此时SerDes工作在1.25Gbps，采样100MFX模式下链路传输的125Mbps信号，CDR不能正常工作，此时采样点可能会在数据的边沿而出现眼图糊掉的现象，导致数字眼图不可用，上层的MAC逻辑通过过采样算法能保证业务没有问题。



8.5.3 存储映射和访问

8.5.3.1 SerDes 基地址

每个Macro占用512KB的地址空间，和其他模块一起映射到系统主空间，各个Macro基地址如下表8-2所示。

表 8-2 SerDes 访问地址空间

编号	基地址
Macro0	0xF3800000
Macro1	0xF3880000
Macro2	0xF3900000

SerDes subctrl是对应每个Macro的控制、状态等寄存器访问接口，支持CPU总线和JTAG两种访问方式，如下表8-3所示。

表 8-3 SerDes subctrl 访问地址空间

编号	基地址
SerDes subctrl0	0xF5090000
SerDes subctrl1	0xF50A0000
SerDes subctrl2	0xF50B0000
HilinkMux_SubCtrl	0xF50C0000

8.5.3.2 SerDes Macro 的内部地址划分

Hilink的寄存器按照功能划分为三类：

- Memory空间，包含RAM、ROM。
- 配置clock slice（即PLL）的寄存器。
- 配置data slice（即lane，通道）的寄存器。

RAM和ROM空间主要用于控制加载Firmware，用于SerDes的工艺参数校准和自适应算法的运行。这部分参数不需要软件参与，由MCU自动运行，软件只需要完成Firmware的加载和启动。

clock slice的寄存器用于设置PLL的参数，以便PLL产生正确的工作时钟送给data slice。

data slice的寄存器用于设置数据通道的参数，比如预加重参数、通道均衡参数、并口位宽、速率模式等。



图 8-1 Hilink Macro 的内部地址分配

	Address																	
Bit #	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Clockslic	0	0	0	0	AT ¹	Slice ID ²			Sub-block ID		register address							
Dataslic	0	0	0	1	AT ¹	Slice ID ²			Sub-block ID		register address							
Reserved0	0	0	1	RESERVED (0x0_8000 – 0x0_FFFF)														
(ROM)	0	1	ROM address(0x1_0000 – 0x1_5FFF)															
(RAM0)	0	1	RAM0 address(0x1_6000 – 0x1_FFFF)															
(RAM0)	1	0	RAM0 address(0x2_0000 – 0x2_1FFF)															
(RAM1)	1	0	RAM1 address(0x2_2000 – 0x2_5FFF)															
other	1	1	RESERVED (0x3_6000 – 0x3_FFFF)															

1. AT (Access Type): 0 = unicast; 1 = broadcast.
2. Slice ID: For dataslice, the range is 0-7.
3. Sub-block ID: For clockslic, 00 = CSANA, 01 = CSDIG , 10 = CP, 11 = SS; For dataslice, 00 = DSANA, 01= DSDIG, 10= DSRXD, 11 = DSTXD.

说明

对于有多条通道的Macro（比如2lane+1pll），Macro内部以用地址Address bits[12:10]来区分，见图8-1中Slice ID一行。SliceID从0开始编号，所以2lane+1pll的Macro，有0b000对应通道0，0b001对应通道1。

对于软件，需要实现以下寄存器进行操作。

说明

寄存器偏移地址只支持按word操作。

1. SerDes内部通用寄存器读写命令。
地址为 SerDes基地址+SerDes偏移地址
2. SerDes内部Clock Slice寄存器读写操作。
地址为 SerDes基地址 + (0x0000_0000 + CS寄存器偏移地址)
3. SerDes内部Data Slice中的DSDIG进行读写操作。
 - a. 写操作：
同时对所有的data slice进行写操作：
地址为 SerDes基地址 + (0x0000_6100 + DSCLK寄存器偏移地址)
单独对一条lane的data slice进行写操作：
地址： SerDes基地址 + (0x0000_4100 + DSCLK寄存器偏移地址+ Dataslice_no * 0x0000_0400)
 - b. 读操作：
不支持同时对所有的data slice进行读操作。
对于单独一条lane的data slice进行读操作：
地址： SerDes基地址 + (0x0000_4100 + DSCLK寄存器偏移地址+ Dataslice_no * 0x0000_0400)
4. SerDes内部Data Slice中的DSANA进行读写操作。
写操作：



同时对所有的data slice进行写操作：

地址为 SerDes基地址 + (0x0000_6000 + TX寄存器偏移地址)

单独对一条lane的data slice进行写操作：

地址： SerDes基地址 + (0x0000_4000 + TX寄存器偏移地址+ Dataslice_no * 0x0000_0400)

读操作：

不支持同时对所有的data slice进行读操作。

对于单独一条lane的data slice进行读操作：

地址： SerDes基地址 + (0x0000_4000 + TX寄存器偏移地址+ Dataslice_no * 0x0000_0200)

5. SerDes内部Data Slice中的RXD进行读写操作。

a. 写操作：

同时对所有的data slice进行写操作：

地址为 SerDes基地址 + (0x0000_4200 + RX寄存器偏移地址)

单独对一条lane的data slice进行写操作：

地址： SerDes基地址 + (0x0000_4200 + RX寄存器偏移地址+ Dataslice_no * 0x0000_0400)

b. 读操作：

不支持同时对所有的data slice进行读操作。

对于单独一条lane的data slice进行读操作：

地址： SerDes基地址 + (0x0000_4200 + RX寄存器偏移地址+ Dataslice_no * 0x0000_0400)

以访问Macro0 DS1 RXD CSR54（PRBS误码统计寄存器）为例，寄存器地址计算说明如下：

地址： SerDes基地址 + (0x0000_4200 + RX寄存器偏移地址+ Dataslice_no * 0x0000_0400)

SerDes基地址: 0xF3800000

RXD CSR54寄存器偏移地址: 0xD8，具体寄存器含义可以参见《HiLink32LRT12_datasheet_V1.3.pdf》查询。

Dataslice_no:1，即DS1

最终计算地址为0xF3800000+（0x4200+0xD8+1*0x400）= 0xF38046D8

8.5.4 Lane 映射

SerDes Lane复用关系如图8-2所示。

图 8-2 SerDes Lane 复用关系

Macro name	Lane #	Macro Type	GE	2.5GE	10GE	QSGMII	PCIe3.0	PCIe3.0	PCIe3.0	SATA
频点	单位: GHz		12.5/15	3.125G	10.3125G	5G	2.5/5/8G	2.5/5/8G	2.5/5/8G	1.5G/3G/6G
使用时钟fclk	单位: MHz		156.25/100	12.5	15.4875	15/15	15/16	15/16	15/16	12
			156.25/100	156.25	156.25	156.25/100	100	100	100	100
Macro0 2个PLL 4lane	lane0	OSS+1CS+4DS	GE4	2.5GE4		QSGMII2	PCIe6_X1			
	lane2		GE5	2.5GE5		QSGMII3	PCIe7_X1			
	lane3		GE10	2.5GE10	10GE0/复用10GE2					
	lane4		GE11	2.5GE11	10GE1/复用10GE3					
Macro1 2个PLL 2lane	lane5	OSS+1CS+2DS	GE0/复用GE10	2.5GE0/复用2.5GE10	10GE2	QSGMII0	PCIe4_X1	PCIe4_X2	PCIe4_X2	SATA0
	lane6		GE3/复用GE11	2.5GE3/复用2.5GE11	10GE3	QSGMII1	PCIe0_X1	PCIe4_X2	PCIe4_X2	SATA1
	lane7		GE1/GE8	2.5GE1/2.5GE8	复用10GE2	复用QSGMII0	PCIe5_X1	PCIe5_X4	PCIe5_X2	复用SATA1
	lane8		GE2/GE9	2.5GE2/2.5GE9	复用10GE3	复用QSGMII1	PCIe1_X1	PCIe5_X4	PCIe5_X2	
Macro2 2个PLL 4lane	lane9	OSS+1CS+4DS	GE6	2.5GE6	复用10GE0	QSGMII4	PCIe2_X1	PCIe5_X4	PCIe2_X1	
	lane9		GE7	2.5GE7	复用10GE1	QSGMII5	PCIe3_X1	PCIe5_X4	PCIe3_X1	



使用限制：

1. 同一行只能工作在其中一种协议。
2. 同一个MACRO两个PLL同时只支持两种速率共存。
3. 若LANE6复用10GE2/QSGMII0时，LANE4不能使用GE0/2.5GE0/10GE2/QSGMII0。
4. 若LANE7复用10GE3/QSGMII1时，LANE5不能使用GE3/2.5GE3/10GE3/QSGMII1。
5. 若LANE4使用10GE2/QSGMII0时，GE1/GE2不可用。
6. 若LANE9使用QSGMII5时，GE8/GE9不可用。
7. 部署接口总带宽超过44Gbps后，会导致不区分优先级丢包。
8. 芯片存在部分Macro或者部分lane在上电时不使用，但若若干年后会使用情况，故即使是不使用的Macro和lane，也要在芯片上电后进行校准，然后再做Macro或者lane的power down。

PCIE/SATA/XFI HILINK的限制：

1. PCIE Gen3和XFI/SATA共存，先做GEN3初始化和建链，然后做 XFI/SATA建链，该共存使用有限制，不支持PCIE降速。
2. 如果PCIE Gen3和QSGMII/GE共存，都使用100M参考时钟，不使用SSC，可以支持。

PCIE控制器：

309a内共集成6个x1、1个x2、1个x4的PCIE控制器，PCIE0~PCIE3仅支持x1模式，PCIE4支持x1、x2模式，PCIE5支持x1、x2、x4模式，PCIE6~PCIE7仅支持x1模式，最高可支持以下几种模式：

1. x1*8。
2. x2+x4+x1*2。
3. x1*4+x4。
4. x1*4+x2*2。

其中，x2可用作x1模式使用，x4可用作x1或者x2模式使用。

MAC接口排布如[图8-3](#)所示。



图 8-3 MAC 接口排布

AXI Master	Port #	(FE/1000BASE-X/SGMII/2.5GBASE-X) 说明: 1000BASE-X/SGMII 支持提到 3.125Gbps	10GE (BASE-R/BASE-R with BRFC) 或者 5GBASE-R/ 5GBASE-KR	QSGMII	RGMI/MLI	PPE Tnl
Master 000	Port0	GE0	10GE2	QSGMII0		PPE Tnl0
	Port1	GE1	NotUsed	NotUsed		PPE Tnl1
	Port2	GE2	NotUsed	NotUsed		PPE Tnl2
	Port3	GE12	NotUsed	NotUsed	RGMI/MLI	PPE Tnl3
	Port4	GE3	10GE3	QSGMII1		PPE Tnl4
	Port5	GE13	NotUsed	NotUsed		PPE Tnl5
	Port6	GE14	NotUsed	NotUsed		PPE Tnl6
	Port7	GE15	NotUsed	NotUsed		PPE Tnl7
	Port8	GE4		QSGMII2		PPE Tnl8
	Port9	GE16		NotUsed		PPE Tnl9
	Port10	GE17		NotUsed		PPE Tnl10
	Port11	GE18		NotUsed		PPE Tnl11
	Port12	GE5		QSGMII3		PPE Tnl12
	Port13	GE19		NotUsed		PPE Tnl13
	Port14	GE20		NotUsed		PPE Tnl14
	Port15	GE21		NotUsed		PPE Tnl15
	Port16	GE6		QSGMII4		PPE Tnl16
	Port17	GE22		NotUsed		PPE Tnl17
	Port18	GE23		NotUsed		PPE Tnl18
	Port19	GE24		NotUsed		PPE Tnl19
	Port20	GE7		QSGMII5		PPE Tnl20
	Port21	GE8		NotUsed		PPE Tnl21
	Port22	GE9		NotUsed		PPE Tnl22
	Port23	GE25		NotUsed		PPE Tnl23
	Port24	GE10	10GE0			PPE Tnl24
	Port25	GE11	10GE1			PPE Tnl25

1. 上述表格总的同一行中，同时只能支持一种速率模式。
2. “NotUsed”表示当使用某种速率模式时，其他对应的port无法工作。
3. Port3、Port5、Port6、Port7、Port9、Port10、Port11、Port13、Port14、Port15、Port17、Port18、Port19、Port23端口通过QSGMII来实现，不能作为单独GE口实现。

8.5.5 Lane 映射的注意事项

- 每条 lane 只能使用一种协议进行工作。
- 若LANE6复用10GE2/QSGMII0时，LANE4不能使用GE0/2.5GE0/10GE2/QSGMII0。
- 若LANE7复用10GE3/QSGMII1时，LANE5不能使用GE3/2.5GE3/10GE3/QSGMII1。
- 若LANE4使用10GE2/QSGMII0时，GE1/GE2不可用。
- 若LANE9使用QSGMII5时，GE8/GE9不可用。
- 当部署接口的总带宽超过 44Gbps 时，可能会导致不分优先级的丢包现象。
- 芯片存在部分Macro或者部分Lane在上电时不使用，但若若干年后会使用的情况，故即使是不使用的Macro和Lane，也要在芯片上电后进行校准，然后再做Macro或者Lane的power down。
- PCIE/SATA/XFI HILINK的限制。
 - a. PCIE Gen3和XFI/SATA共存，先做GEN3初始化和建链,然后做 XFI/SATA建链,该共存使用有限制,不支持PCIE降速。
 - b. PCIE Gen1/2和XFI/SATA共存，SATA使用PLL0，PCIE使用PLL1，先做PCIE初始化和建链，然后做XFI/SATA初始化和建链，可以支持。
 - c. 如果PCIE Gen3 和QSGMII/GE共存，都使用100M参考时钟，不使用SSC，可以支持。



说明

XFI速率：10.3125G、3.125G其中之一，QSGMII 5Gbps。

8.5.6 外部参考时钟

所有Macro共用一组差分参考时钟QL_REFCLK_INP0/M0/ QL_REFCLK_INP/M1，参考时钟频率分别为156.25/100MHz。

8.6 应用配置

8.6.1 SerDes 频率配置

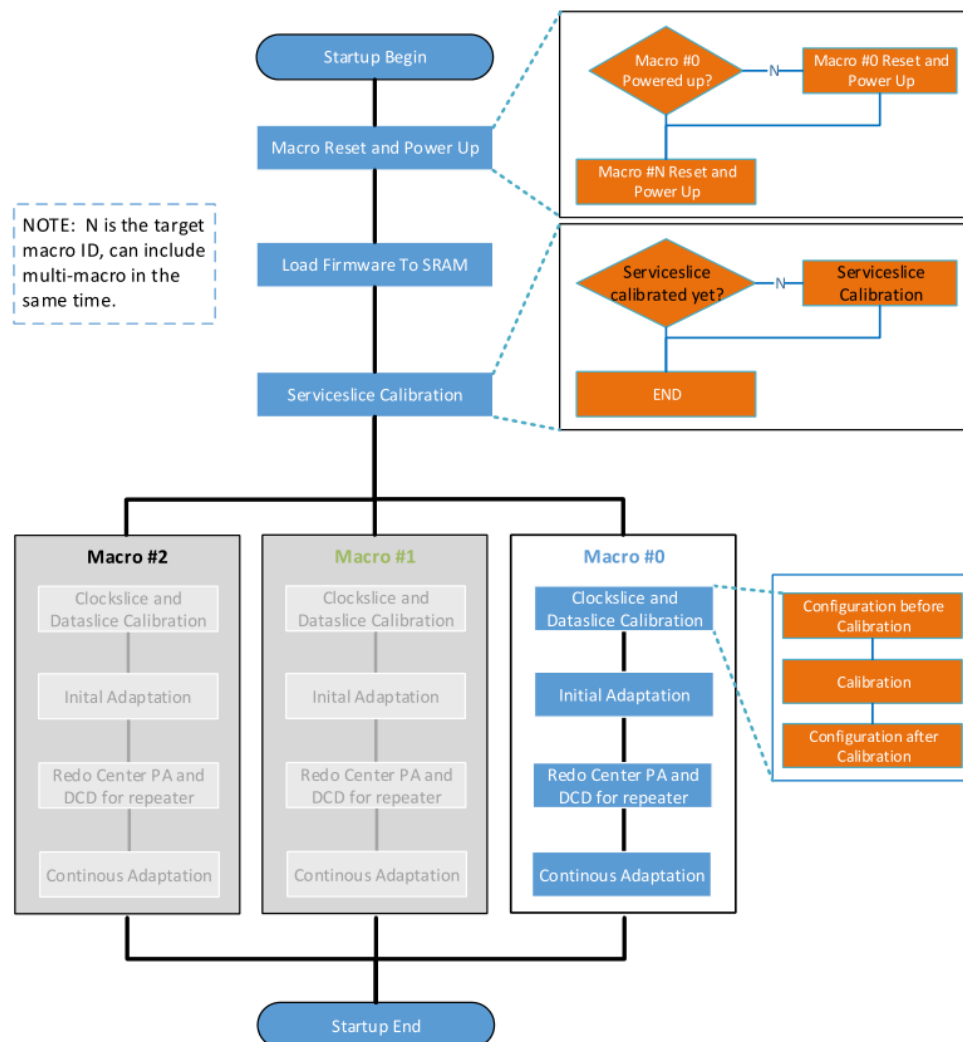
SerDes各种模式下的频率配置表，参见《Hi309a HiLink32LRT12_Frequency Plan_2.0》中Frequency Plan章节的描述；具体每个Macro的速率组合下的配置表，参见复用方案中速率组合表页签。

8.6.2 Macro 掉电

只能通过控制对应Macro的subctrl中SC_HILINK_MACRO_MACROPWRDB寄存器的配置，来power down 整个Macro。

8.7 初始化

图 8-4 SerDes 初始化流程图



由于各个接口的速率存在差异，各接口的启动流程如下：

- 步骤1 撤销SerDes powerdown和grst。
- 步骤2 Clock Slice Configuration。
- 步骤3 进行CS Calibration。
- 步骤4 进行DS Calibration前配置。
- 步骤5 进行DS Calibration（软件或 hardware calibration，优先使用hardware calibration）。
- 步骤6 进行DS Calibration后配置。



步骤7 设置恢复时钟输出和mclk分频。

----结束

在线速率重新配置时，需要软件进行以下步骤：

步骤1 SerDes进行全局复位操作grst。

步骤2 Clock Slice Configuration。

步骤3 进行CS Calibration。

步骤4 进行DS Calibration前配置。

步骤5 进行DS Calibration（软件 或 hardware calibration，优先使用hardware calibration）。

步骤6 进行DS Calibration后配置。

步骤7 设置恢复时钟输出和mclk分频。

----结束

说明

除了初次上电需要配置SerDes powerdown外，后续的线速率初始化配置只配置grst。

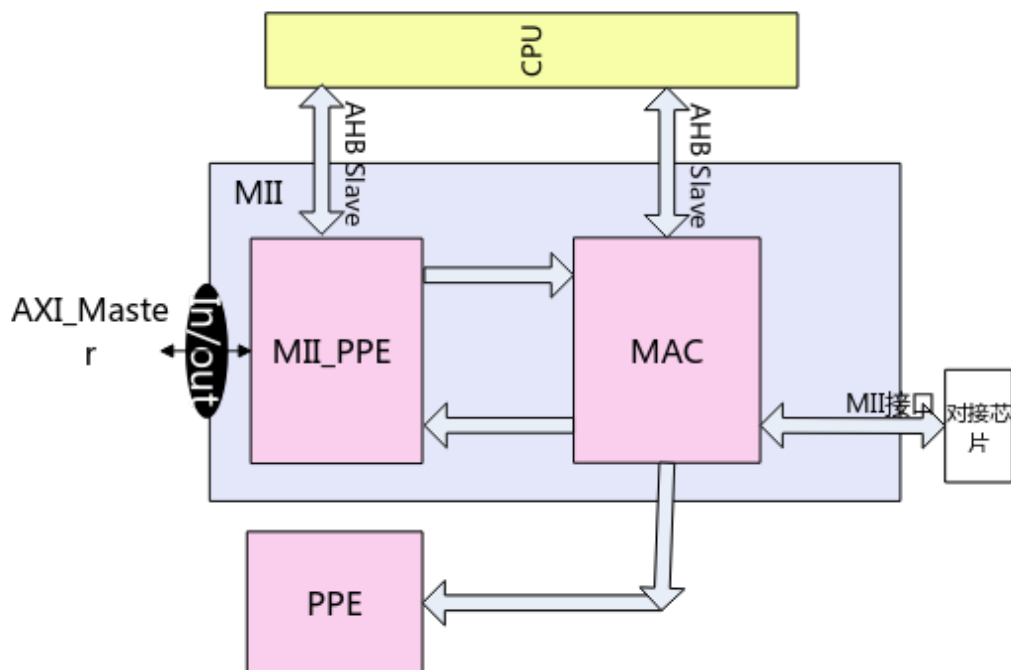
9 RGMII

9.1 概述

CPU-DIE的调试网口由将简化版的PPE和MII接口组成，用于调试网口或canbus网口。

该网口不支持报文解析，不支持多模，仅支持报文的简单收发。MII接口是GE支持的接口中的一种。

图 9-1 MII 连接示意图





9.2 关键术语

表 9-1 关键术语解释

术语	术语解释
MAC	Media Access Control，媒体接入控制
MII	Medium Independent Interface，介质独立接口
PPE	Packet Parse Engine，包解析引擎模块
RvMII	Reverse MII，时钟的I/O方向反转的MII
RGMII	Reduced Gigabit MII，精简吉比特介质独立接口

9.3 功能特性

调试网口MII具有以下特性：

- 支持10M/100M 速率的MII/RvMII，不支持半双工，支持10M/100M/1000M RGMII。
- 支持接收方向检查CRC，对于CRC错误的报文，在BD解析描述符L2E置位。
- 发送仅支持CPU直接添加任务，是否成功可以查询。
- 发送完报文后，支持向发送描述符指定的地址清0，软件查询该地址是否为0确认报文是否完成发送。
- 接收仅支持CPU直接分配缓存。
- 接收支持收到软件配置的报文个数后上报中断，或收到报文后超时上报中断。
- 支持流控帧的收发和响应。
 - 支持流控请求响应功能，接收流控帧响应使能可配置。
 - 支持流控帧发送功能。
 - 支持识别流控帧，并将流控请求通知发送通道。
 - 支持发送流控帧的MAC源地址可配置。
 - 支持发送流控帧的间隙和流控帧的时间域值可配置。
 - 支持同时识别标准目的MAC地址和可配的非标准目的MAC地址流控帧的识别。
- 支持符合802.3协议的帧发送处理功能。
 - 支持插入标准长度的前导码、SFD。
 - 支持根据配置决定是否对发送报文进行PAD填充，即把小于60 Bytes的不带FCS的帧填充至60Bytes后，再添加FCS。
 - 支持根据配置决定是否对发送报文添加FCS。
 - 支持发送帧间隙可配置。
- 支持符合802.3协议的帧接收处理功能。



- 支持剥离前导码和SFD。
- 支持剥离802.3 snap帧的PAD，剥离PAD的同时则必须剥离FCS；如果不剥离PAD，则可以配置决定是否剥离FCS。
- 支持帧错误标识功能，即帧接收过程中如果发现FCS校验错误，则置帧错误标志为有效，寄存器可配置是否标示错误。
- 提供帧统计计数信息和状态信息。
 - 支持流控帧、VLAN帧、广播帧、单播帧、组播帧、数据帧等收发数量统计信息。
 - 支持特殊帧或者错误帧收发统计信息，主要包括超长帧（对于超长帧会上报二层错）、超短帧、CRC错误帧、过滤帧等。
 - 支持收发帧长统计，并对各个长度范围内收发帧进行个数统计。
 - 正常报文收发字节统计支持48bit统计位宽。
- 支持MAC环回测试功能。
 - 支持内环回测试，即从应用侧发送的报文在GE内环回到接收方向给应用侧。
 - 支持外环回测试，即从链路侧发出报文在GE内环回到发送方向给链路侧。
- 支持最长收发报文和最短收发报文长度可配置，长于最长报文配置的报文将会被截断，置上错误标示发送出去，短于64Byte的报文可配置直接过滤。
- 支持对接收报文进行目的MAC地址过滤，目的MAC和本地MAC地址不同的报文直接丢弃，支持4个独有的不带掩码的本地MAC地址配置和2个带掩码的本地MAC地址配置。
- 支持RGMII通过配置选择可以走PPE模块的LSW通路。

9.4 遵循协议

802.3协议。

9.5 功能描述

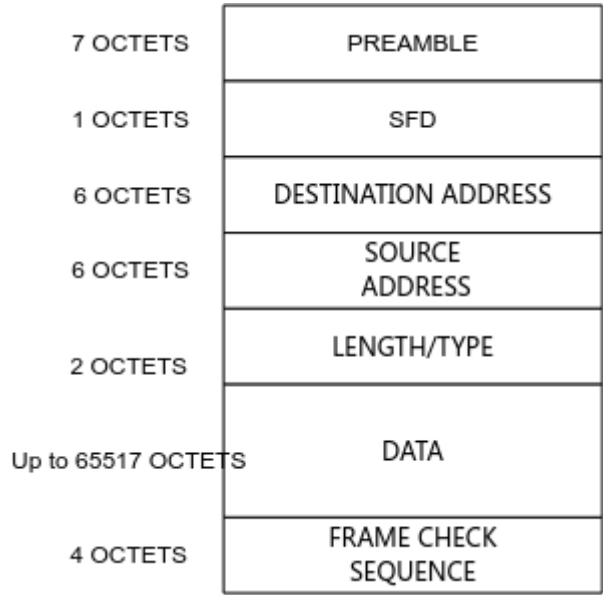
9.5.1 数据结构

9.5.1.1 报文数据格式

标准的802.3协议帧格式如图9-2所示。



图 9-2 802.3 帧格式



9.5.2 MII 接口信号描述

表 9-2 MII 接口信号描述

管脚	内部信号名称	方向	描述
MII0_RXCLK	MII0_RXCLK	I	MII0接口接收时钟
MII0_RXD	MII0_RXD	I	MII0接口接收数据线
MII0_RXDV	MII0_RXDV	I	MII0接口接收数据有效
MII0_RXER	MII0_RXER	I	MII0接口接收数据错误
MII0_TXCLK	MII0_TXCLK	I	MII0接口发送时钟
MII0_TXD	MII0_TXD	O	MII0接口发送数据线
MII0_TXEN	MII0_TXEN	O	MII0接口发送使能信号
MII0_TXER	MII0_TXER	O	MII0接口发送数据错误
MII0_COL	MII0_COL	I	MII0接口接收帧错误指示
MII0_CRS	MII0_CRS	I	MII0接口接收总线空闲状态指示信号

9.6 软件指南

注意：具体寄存器的使用请参考《Hi309a 寄存器手册》。



9.6.1 发送报文

- 步骤1** 软件在主存中构造好报文，并且将发送描述符构造好，填好发送描述符的WB_ADDR，并将发送描述符的CL置1。
- 步骤2** 将发送描述符的地址推送到发包寄存器PPE_CFG_CPU_ADD_ADDR。
- 步骤3** 查询状态寄存器PPE_CFG_CPU_ADD_ADDR_FAIL是否添加成功，如果添加失败，将失败状态清0（对应位写1清）后重复**步骤2**和**步骤3**。
- 步骤4** 查询WB_ADDR的值是否为0，如果为0，表示报文已经进入硬件处理，软件可以回收Buffer，否则暂停一段时间再查询。

----结束

9.6.2 接收报文

- 步骤1** 查询当前缓存个数寄存器PPE_CURR_CFF_DATA_NUM[24:16]，如果为0x100，表示内部缓存已满，不能再注入缓存，否则可以向缓存分配寄存器PPE_CFG_RX_CFF_ADDR注入缓存，软件需要保证注入缓存不能溢出，注入的缓存要先进行清理，硬件提供最大256个缓存空间。
- 步骤2** 网口收到报文，上报中断。
- 步骤3** 软件采样轮询的方式查询所有缓存（写入PPE的所有缓存）的描述符状态是否有效（描述符中DataLength有值,且没有二层错误和drop指示，则是一个有效的报文），然后处理报文。
- 步骤4** 清除收包中断PPE_RINT[10]。
- 步骤5** 用完的缓存可以选择继续注入到PPE内部（注入的缓存要先进行清理）。

----结束

9.6.3 系统内环配置方法

- 步骤1** 配置应用侧环回使能GE_LOOP_REG寄存器cf2mi_lp_en（GE偏移地址0x1dc[2]）为1。

----结束

9.6.4 流控帧过滤

MAC CORE的接收通道可以通过配置相应的寄存器对流控帧进行过滤。首先对接收到的流控帧DA域进行检查，如果该流控帧的DA域与01-80-C2-00-00-01完全匹配，或者与流控帧目的地址寄存器值完全匹配，则该流控帧MAC地址匹配。继续判断该流控帧的长度/类型域是否为88-08-00-01，如果匹配在流控帧过滤使能的情况下则丢弃该流控帧，并进行接收方向过滤掉的帧统计和接收方向过滤掉的字节统计。流控帧虽然被丢弃但流控时间被存储，发送方向将按照流控时间停止发送报文流。

说明

接收流控帧过滤使能时，当接收到流控帧该帧将被丢弃并统计进接收方向丢弃的帧和字节统计寄存器。全双工模式下响应流控帧使能，发送方向才能按照流控时间停止发送报文，并在CRC检测有效且CRC正确、帧长大于等于64Byte的情况下被统计进接收到的流控帧统计寄存器。

流控帧过滤配置如下：



- 步骤1** 配置流控帧目的地址低32位寄存器FD_FC_ADDR_LOW（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0020，bit[31:0]），配置流控帧目的地址高16位寄存器FD_FC_ADDR_HIGH（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0024，bit[15:0]），配置流控帧目的地址。
- 步骤2** 配置接收方向相应流帧使能寄存器PAUSE_EN[rx_fdfc]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0048，bit[0]）为1，全双工模式下响应流控帧使能。
- 步骤3** 配置流控帧过滤使能寄存器REC_FILT_CONTROL[pause_frm_pass]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0064，bit[4]）为0，流控帧过滤使能。
- 结束

9.6.5 广播帧过滤

MAC CORE的接收通道可以通过配置相应的寄存器对接收到的广播帧进行过滤，当该功能有效时检查接收帧的DA域是否全1，匹配则该帧将被丢弃，并进行接收方向过滤掉的帧统计和接收方向过滤掉的字节统计。

配置广播帧过滤使能寄存器REC_FILT_CONTROL[bc_drop_en]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0064，bit[2]）为1，广播帧过滤使能。

9.6.6 多播帧过滤

MAC CORE的接收通道可以通过配置相应的寄存器对接收到的多播帧进行过滤，当该功能有效时检查接收帧的DA域与2个多播帧地址匹配寄存器的配置值是否相等，检查时会根据地址屏蔽寄存器配置值将相应地址bit位屏蔽掉（即屏蔽掉的地址bit不进行比较），如果不匹配，则该帧将被丢弃，并进行接收方向过滤掉的帧统计和接收方向过滤掉的字节统计。

说明

检查多播帧DA域才有使用带掩码MAC地址的需求，因此带掩码的MAC地址用于多播帧MAC地址匹配。

多播帧过滤配置如下：

- 步骤1** 配置本机MAC地址0寄存器1STATION_ADDR_LOW_0（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0200，bit[31:0]），配置本机MAC地址0寄存器2STATION_ADDR_HIGH_0（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0204，bit[15:0]），第8位为1，其他位根据需求可选；配置本机MAC地址1寄存器2STATION_ADDR_LOW_1（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0208，bit[31:0]），配置本机MAC地址1寄存器2STATION_ADDR_HIGH_1（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x020C，bit[15:0]），第8位为1，其他位根据需求可选。
- 步骤2** 配置本机MAC地址0寄存器1的掩码STATION_ADDR_MSK_LOW_0（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0230，bit[31:0]）；配置本机MAC地址0寄存器2的掩码STATION_ADDR_MSK_HIGH_0（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0234，bit[15:0]）；这两个寄存器与**步骤1**中所配的前两个寄存器按位与的结果构成一个带掩码的MAC地址。配置本机MAC地址1寄存器1的掩码STATION_ADDR_MSK_LOW_1（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0238，bit[31:0]）；配置本机MAC地址1寄存器2的掩码STATION_ADDR_MSK_HIGH_1（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x023C，bit[15:0]），第8位为1，其他位



根据需求可选；这两个寄存器与**步骤1**中所配的后两个寄存器按位与的结果构成一个带掩码的MAC地址。

步骤3 配置本机MAC地址0寄存器2STATION_ADDR_HIGH_0 [station_addr0_en]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0204，bit[16]）为1，开启该MAC地址的使能。

步骤4 配置多播帧过滤使能寄存器REC_FILT_CONTROL[mc_drop_en]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0064，bit[1]）为1，多播帧过滤使能。

----结束

9.6.7 单播帧过滤

MAC CORE的接收通道可以通过配置相应的寄存器对接收到的单播帧进行过滤，当该功能有效时检查接收帧的DA域是否与4个MAC源地址中至少一个完全匹配，否则该帧将被丢弃，并进行接收方向过滤掉的帧统计和接收方向过滤掉的字节统计。

单播帧过滤配置如下：

步骤1 配置本机MAC地址2寄存器1STATION_ADDR_LOW_2（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0210，bit[31:0]），配置本机MAC地址2寄存器2STATION_ADDR_HIGH_2（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0214，bit[15:0]），第8位为0，其他位根据需求可选；这两个寄存器的组合值为一个本机MAC源地址。配置本机MAC地址3寄存器2STATION_ADDR_LOW_3（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0218，bit[31:0]），配置本机MAC地址3寄存器2STATION_ADDR_HIGH_3（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x021C，bit[15:0]），第8位为0，其他位根据需求可选；这两个寄存器的组合值为一个本机MAC源地址。配置本机MAC地址4寄存器1STATION_ADDR_LOW_4（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0220，bit[31:0]），配置本机MAC地址4寄存器2STATION_ADDR_HIGH_4（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0224，bit[15:0]），第8位为0，其他位根据需求可选；这两个寄存器的组合值为一个本机MAC源地址。配置本机MAC地址5寄存器2STATION_ADDR_LOW_5（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0228，bit[31:0]），配置本机MAC地址5寄存器2STATION_ADDR_HIGH_5（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x022C，bit[15:0]），第8位为0，其他位根据需求可选；这两个寄存器的组合值为一个本机MAC源地址。

步骤2 配置本机MAC地址2寄存器2 STATION_ADDR_HIGH_2 [station_addr0_en]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0214，bit[16]）为1，本机MAC地址3寄存器2 STATION_ADDR_HIGH_3 [station_addr0_en]，本机MAC地址4寄存器2 STATION_ADDR_HIGH_4[station_addr0_en]，本机MAC地址5寄存器2 STATION_ADDR_HIGH_5 [station_addr0_en]，开启该MAC地址的使能。

步骤3 配置单播帧过滤使能寄存器REC_FILT_CONTROL[uc_drop_en]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x0064，bit[0]）为1，单播帧过滤使能。

----结束

9.6.8 PAD 剥离

802.3帧在封装数据时，封装完的报文不足64字节会填充PAD。



须知

VLAN封装的帧及以太网帧无法剥离PAD；
剥离PAD时连CRC一起剥离，无法只剥离PAD不剥离CRC。

步骤1 配置接收控制寄存器RECV_CONTROL [strip_pad_en]（基地址0xF4C0_0000+1000*ch_num，偏移地址为0x01E0，bit[3]）为1。

----结束

9.6.9 发送截断

发送的数据的最大长度可以通过寄存器配置，如果发送的数据长度大于配置值，发送超过该长度报文则截断数据，并置为CRC错：

配置最大帧长寄存器MAX_FRM_SIZE(基地址0x30000000+ 1000* ch_num，偏移地址为0x003C，bit [15:0])值为0x05EE，设置发送数据帧的最大长度为0x05EE个字节。

表 9-3 发送方向截断报文及统计计数相关说明（1000M，MAX_FRM_SIZE 配置值为N）

报文类型	CPU发送的报文字节长度M(无CRC)	MAC发送的报文字节长度（有CRC）	OCTETS_TRANSMITTED_BAD统计值
VLAN帧	$N+4 < M < N+10$	N+9	M+4
	$N+10 \leq M$		N+14
非VLAN帧	$N < M < N+6$	N+5	M+4
	$N+6 \leq M$		N+10

说明

OCTETS_TRANSMITTED_BAD(基地址0x30000000+ 1000* ch_num，偏移地址为0x0104，bit[14:0])。

表 9-4 发送方向截断报文及统计计数相关说明（10/100M，MAX_FRM_SIZE 配置值为N）

报文类型	CPU发送的报文字节长度M(无CRC)	MAC发送的报文字节长度（有CRC）	OCTETS_TRANSMITTED_BAD统计值
VLAN帧	$M = N+5$	N+9	N+9
	$M = N+6$ 或 $N+7$		N+10



报文类型	CPU发送的报文字节长度M(无CRC)	MAC发送的报文字节长度（有CRC）	OCTETS_TRANSMITTED_BAD统计值
	$M = N+8$ 或 $N+9$		$N+11$
	$M = N+10$ 或 $N+11$		$N+12$
	$M = N+12$ 或 $N+13$		$N+13$
	$N+13 < M$		$N+14$
非VLAN帧	$M = N+1$	$N+5$	$N+5$
	$M = N+2$ 或 $N+3$		$N+6$
	$M = N+4$ 或 $N+5$		$N+7$
	$M = N+6$ 或 $N+7$		$N+8$
	$M = N+8$ 或 $N+9$		$N+9$
	$N+9 < M$		$N+10$

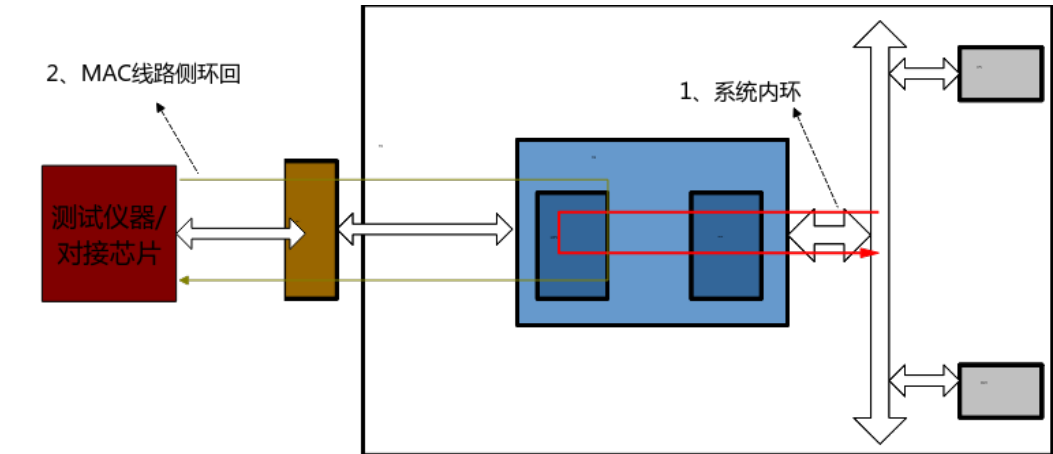
说明

OCTETS_TRANSMITTED_BAD(基地址0x30000000+ 1000* ch_num，偏移地址为0x0104，bit [14:0])。

9.6.10 各种环回介绍

MII主要有下面2种环回。

图 9-3 GMAC 的各种环回





- 环回1：系统内环
用来测试系统内部是否正常，可以用来确认CPU、总线、DDR等是否正常工作。
- 环回2：MAC线路侧环回
除了用来测试仪器/对接芯片、PHY通路是否正常，还可以确认MAC是否正常工作。

9.6.11 软复位

MII同系统总线有握手信号，不能在任意时刻对MII进行单独复位，在加速器空闲情况下可进行软复位，软复位会清除当前加速器所有配置参数和未完成的任务。

须知

在解复位后必须读取到该模块的时钟以及解复位状态，确认时钟复位正常后，才能进行下一步的模块功能操作。

时钟状态查询ETH SUBCTRL中RGMII时钟状态寄存器

图 9-4 RGMII 时钟状态寄存器

SC_RGMII_ICG_ST	RGMII时钟状态寄存器	0x5360
-----------------	--------------	--------

时钟状态查询PERI SUBCTRL中RGMII时钟状态寄存器

图 9-5 RGMII 时钟状态寄存器

SC_RGMII_CLK_ST	RGMII时钟状态寄存器	0x5468
-----------------	--------------	--------

复位状态查询ETH SUBCTRL中RGMII复位状态寄存器

图 9-6 RGMII 复位状态寄存器

SC_RGMII_RESET_ST	RGMII复位状态寄存器	0x5a70
-------------------	--------------	--------

时钟状态查询PERI SUBCTRL中RGMII复位状态寄存器

图 9-7 RGMII 复位状态寄存器

SC_RGMII_RESET_ST	RGMII软复位状态寄存器	0x5B28
-------------------	---------------	--------

复位配置步骤如下：

步骤1 软件停止向PPE推送发送描述符地址，一直到复位流程全部结束。



- 步骤2** 配置通道使能寄存器PORT_EN [tx_en]（MII寄存器：偏移地址为0x0044，bit[2]）为1，开启发送通道，配置端口通道使能寄存器PORT_EN [rx_en]（GE寄存器：偏移地址为0x0044，bit[1]）为0，关闭接收通道。
- 步骤3** 配置端口复位指示寄存器PPE_CFG_TNL_TO_BE_RST[tnl_to_be_rst]（PPE寄存器：偏移地址为0x4f8，bit[0]）为1，指示硬件软件需要复位。
- 步骤4** 查询通道是否可以复位寄存器PPE_CURR_TNL_CAN_RST[tnl_could_be_rst]（PPE寄存器：偏移地址为0x4fc，bit[0]）是否为1，如果不为1，隔10ms重新执行该步骤，若返回值没有改变，则疑似系统总线接口数据传输异常，此时建议软件对系统进行软复位。
- 步骤5** 配置MII复位寄存器SC_MII_RESET_REQ（PERI Subctrl寄存器）为0x7；配置MII复位寄存器SC_MII_RESET_REQ（ETH Subctrl寄存器）为0x3F。
- 步骤6** 配置MII解复位寄存器SC_MII_RESET_DREQ（PERI Subctrl寄存器）为0x7；配置MII复位寄存器SC_MII_RESET_DREQ（ETH Subctrl寄存器）为0x3F。

----结束

9.6.12 初始化

- 步骤1** 配置MII复位寄存器SC_MII_RESET_REQ（PERI Subctrl）为0x7；配置MII复位寄存器SC_MII_RESET_REQ（ETH Subctrl寄存器）为0x3F。
- 步骤2** 配置MII解复位寄存器SC_MII_RESET_DREQ（PERI Subctrl）为0x7；配置MII复位寄存器SC_MII_RESET_DREQ（ETH Subctrl寄存器）为0x3F。
- 步骤3** 根据模式配置PERI Subctrl 中SC_RGMII_CRG_CFG寄存器。
- 步骤4** 配置端口通道使能寄存器PORT_EN [tx_en]（偏移地址为0x0044，bit[2]）为0，关闭发送通道。
- 步骤5** 配置端口通道使能寄存器PORT_EN [rx_en]（偏移地址为0x0044，bit[1]）为0，关闭接收通道。
- 步骤6** 配置全双工半双工模式寄存器DUPLEX_TYPE [duplex_sel]（偏移地址为0x0008，bit[0]）为双工模式。
- 步骤7** 配置端口模式寄存器PORT_MODE [port_mode]（偏移地址为0x0040，bit[3:0]）为对接单板的工作模式。
- 步骤8** 配置端口模式改变使能寄存器MODE_CHANGE_EN [mode_change_en]（偏移地址为0x01B4，bit[0]）为1。
- 步骤9** 配置端口通道使能寄存器PORT_EN [tx_en]（偏移地址为0x0044，bit[2]）为1，打开发送通道。

配置端口通道使能寄存器PORT_EN [rx_en]（偏移地址为0x0044，bit[1]）为1，打开接收通道。

----结束

9.6.13 RvMII 模式的配置

- 步骤1** 配置PERI Subctrl 中MII CRG寄存器，SC_RGMII_CRG_CFG[1:0]为3，SC_RGMII_CRG_CFG[8:2] 需要结合当前的晶振频点配置为2.5M（10M RvMII）或25M（100M RvMII）。



图 9-8 MII CRG 寄存器

SC_RGMII_CRG_CFG	RGMII CRG寄存器	0x241C	32	0x00000000	reserved	31:12	-	0x00000000	保留。
					rgmii_to_ahb_sel	11	3B	0x0	RGMII F2B 时钟源头选择： 0: CRG提供； 1: RGMII 提供。
					sel_rgmii_to_ahb	10	3B	0x0	RGMII 时钟源头选择： 0: 本地时钟源； 1: 外部时钟源。
					reserved	9	-	0x0	保留。
					rgmii_div_num	0:2	3B	0x02	RGMII 时钟分频系数，需要结合不同晶振（50M/60M/100M）需求，计算分频系数，接口频率 = 晶振频率 / rgmii_div_num * 100000000。RGMII 晶振频率，对应 0: 0M 接口频率需求，0 配置；rgmii_div_num = 1: 40: 25，需要同时设置 D0_S10KCTL和RGMII_S10KCTL两寄存器，保证时钟匹配。
					rgmii_to_ahb_sel	1	3B	0x0	RGMII F2B 时钟源头选择： 0: CRG提供； 1: RGMII 提供。
					rgmii_to_ahb_sel	0	3B	0x0	RGMII 时钟源头选择： 0: 本地时钟源； 1: 外部时钟源。

步骤2 后续按照初始化流程进行。

----结束

9.6.14 RGMII 模式的配置

步骤1 配置PERI Subctrl 中MII CRG寄存器， SC_RGMII_CRG_CFG[1:0]为0b10， SC_RGMII_CRG_CFG[11]为0b1。

图 9-9 MII CRG 寄存器

SC_RGMII_CRG_CFG	RGMII CRG寄存器	0x241C	32	0x00000000	reserved	31:12	-	0x00000000	保留。
					rgmii_to_ahb_sel	11	3B	0x0	RGMII F2B 时钟源头选择： 0: CRG提供； 1: RGMII 提供。
					sel_rgmii_to_ahb	10	3B	0x0	RGMII 时钟源头选择： 0: 本地时钟源； 1: 外部时钟源。
					reserved	9	-	0x0	保留。
					rgmii_div_num	0:2	3B	0x02	RGMII 时钟分频系数，需要结合不同晶振（50M/60M/100M）需求，计算分频系数，接口频率 = 晶振频率 / rgmii_div_num * 100000000。RGMII 晶振频率，对应 0: 0M 接口频率需求，0 配置；rgmii_div_num = 1: 40: 25，需要同时设置 D0_S10KCTL和RGMII_S10KCTL两寄存器，保证时钟匹配。
					rgmii_to_ahb_sel	1	3B	0x0	RGMII F2B 时钟源头选择： 0: CRG提供； 1: RGMII 提供。
					rgmii_to_ahb_sel	0	3B	0x0	RGMII 时钟源头选择： 0: 本地时钟源； 1: 外部时钟源。

步骤2 后续按照初始化流程进行。

----结束

9.6.15 RGMII/RvMII/MII 对接 PPE 的 LSW 配置

该配置放在MII 初始化的过程中进行，不允许动态配置。

步骤1 配置ETH Subctrl 中SC_CTRL_RGMII寄存器， rgmii_to_lsw_mode为1。

图 9-10 SC_CTRL_RGMII 寄存器

SC_CTRL_RGMII	RGMII lsw 模式 信号配置寄存器	0x2A04	32	reserved	31:1	-	0x00000000	保留。
				rgmii_to_lsw_mode	0	RW	0x0	RGMII数据上到PPE的lsw模式： 0: RGMII与PPE通过Lsw模式交互； 1: RGMII与PPE通过Lsw模式交互。

----结束

9.6.16 MII 的 ddr 访存地址扩展配置

由于MII支持的ddr访存地址为36bit，将MII的ddr访存地址进行了扩展，让MII能够访问ddr空间不局限在低2GB地址范围内。该配置放在MII 初始化的过程中进行，不允许动态配置。

步骤1 配置ETHSUB Subctrl 中RGMII_DAW寄存器，为MII访存 DDR地址空间的高位。



图 9-11 RGMII_DAW 寄存器

SC_AWADDR_RGMII_DAW	RGMII AXI MASTER访问 端口扩展配置	0x2A08	32	reserved	31:12	-	0x0000	保留。
				ec_ewaddr_rgmii_daw	11:0	RW	0x000	RGMII AXI MASTER写的端口扩展配置
SC_ARADDR_RGMII_DAW	RGMII AXI MASTER读访问 端口扩展配置	0x2A0C	32	reserved	31:12	-	0x0000	保留。
				ec_araddr_rgmii_daw	11:0	RW	0x000	RGMII AXI MASTER读的端口扩展配置

----结束

9.7 性能

支持100Mbps速率以太网报文传输。

9.7.1 实际测试结果

支持10/100/1000Mbps速率以太网报文传输。

9.8 DFX 应用指导

9.8.1 中断

表 9-5 中断信息

中断名	中断寄存器中的bit位置	中断含义
rx_st_heat_int_sts	15	接收状态机心跳检测超时中断状态寄存器
tx_st_heat_int_sts	14	发送状态机心跳检测超时中断状态寄存器
tx_bd_addr_fifo_int_sts	13	发送描述符地址FIFO溢出中断状态寄存器
ge_fifo_int_sts	12	GE的FIFO异常中断状态和GE接收通道失步中断（失步中断仅GE0~7存在）寄存器。 初始化完成后该位为1是正常的（初始化前会报出），请清除，如果清除后还有该中断，说明链路有异常或MAC FIFO有异常，需要处理。
ddr_rw_int_sts	11	读写DDR总线错误中断状态寄存器
rx_pkt_int_sts	10	接收的报文满足配置条件的中断状态寄存器（参考地址为0x740寄存器的描述）
buf_fifo_over_int_sts	9	软件分配的缓冲出现溢出丢失中断状态寄存器
rx_no_buf_int_sts	8	接收报文因没有缓存丢弃中断状态寄存器
rx_drop_int_sts	7	接收报文因FIFO满丢弃中断状态寄存器



中断名	中断寄存器中的bit位置	中断含义
tx_drop_int_sts	6	发送报文因FIFO空丢弃中断状态寄存器
buf_sram_ecc_2b_int_sts	5	存储缓存的SRAM出现2bit ECC错误中断状态寄存器
buf_sram_ecc_1b_int_sts	4	存储缓存的SRAM出现1bit ECC错误中断状态寄存器
tx_bd_sram_ecc_2b_int_sts	3	发送报文描述符的SRAM出现2bit ECC错误中断状态寄存器
tx_bd_sram_ecc_1b_int_sts	2	发送报文描述符的SRAM出现1bit ECC错误中断状态寄存器
tx_sram_par_int_sts	1	发送报文的SRAM出现奇偶校验错误中断状态寄存器
rx_sram_par_int_sts	0	接收报文的SRAM出现奇偶错误中断状态寄存器



10_{RMII}

10.1 概述

SOC提供三个RMII调试网口。

10.2 关键术语

表 10-1 关键术语解释

术语	术语解释
RMII	Reduced Media Independant Interface，精简MII接口

10.3 特性

SOC提供三个RMII调试网口，相关特性列举如下：

1. 提供RMII信号接口，50MHz时钟由外部提供或由内部生成时钟。
2. 支持10M/100Mbps模式。
 - a. 提供10M/100Mbps的Ethernet MAC，芯片可以通过外接物理层芯片（PHY），输出以太网接口。
 - b. Ethernet MAC符合IEEE802.3标准。
3. 以太网端口都支持全双工模式下的流控帧的发送。
 - a. 支持接收流控帧（Pause帧）并根据流控帧算法处理流控。
 - b. 支持发送流控帧。
 - c. 支持发送流控帧水线可配置，流控水线根据收包队列深度而设置。
4. 支持全双工模式下的内环回和外环回。
 - a. 支持报文从MAC发送侧（不经过RMII接口）直接环回到MAC接收测的内环回功能，即RMII内环回。



- b. 支持报文从MAC接收侧直接环回到MAC发送侧的外环回功能，即RMII外环回。
- 5. 支持中断上报，包括错误中断。
- 6. 支持帧长有效性检测，丢弃超长帧和超短帧，以及短帧填充。
 - a. 当输入帧帧长小于最小帧长设置时，可配置该帧接收或者丢弃。
 - b. 超短帧透传不使能时，最小帧长为64Byte。
 - c. 超短帧透传使能时，最小帧长范围（42～63Byte）可配置。
 - d. 超短帧帧长默认值为42Bytes。
 - e. 当超短帧帧长配置值小于42Byte时，按照42Byte处理。
 - f. 当输入帧帧长大于最长帧设置时，直接将该帧丢弃。
 - g. 超长帧帧长范围（1518～1535Byte）可配置。
 - h. 超长帧帧长默认值为1518Byte。
- 7. 支持对接收帧进行CRC检验，可丢弃校验错误的帧。
- 8. 支持为发送帧添加CRC检验码，可配置。
- 9. 提供64个接收和发送共用的帧管理队列。
- 10. 对广播帧、组播帧和单播帧支持可配置是否接收或者丢弃。
- 11. RMII接口支持发送端/接收端上下沿可配，接收端上升沿接收。支持时钟调相功能。

10.4 软件指南

10.4.1 初始化配置

步骤1（可选）配置RMII 调相参数和正反沿选择参数，默认是正沿时钟，SLAVE模式。

表 10-2 PERI Subctrl 寄存器描述中 SC_RMII_CFG（0xF200_0000+0x2410）

bit	名称	类型	描述	复位值	配置值
[18]	sc_gcrg_rmi i_tx_clk_sel	RW	主/从模式发送时钟正反向时钟选择，配置1则选择反向时钟。	0x0	根据需求配置
[17]	sc_gcrg_rmi i_rx_clk_sel	RW	主/从模式接收时钟正反向时钟选择，配置1则选择反向时钟。	0x0	根据需求配置
[16]	sc_gcrg_rmi i_clk_sel	RW	master/slave模式选择，0是master模式，1是slave模式。	0x1	根据需求配置
[15:12]	sc_rmii_pip e_tx_clk_dly _sel	RW	RMII 发送侧时钟调相选择。 0x0：不调相； 0x1：调相1/800MHz； 以此类推。	0x0	根据需求配置



bit	名称	类型	描述	复位值	配置值
[11:8]	sc_rmii_pip e_rx_clk_dly _sel	RW	RMII 接收侧时钟调相选择。 0x0: 不调相; 0x1: 调相1/800MHz; 以此类推。	0x0	根据需 求配置
[7:0]	sc_rmii_div_ cfg	RW	RMII 800MHz时钟分频系数: 800/16=50MHz	0x0B	根据需 求配置

步骤2 配置RMII的总线USER配置，配置地址：0xF200_0000+ 0x4210；配置值：0
x6000_0000。

步骤3 解复位RMII 调相时钟生成模块。

配置SC_RMII_RESET_ST（0xF200_0000+ 0xAE0）中bit[1]为0b0，解复位RMII 调相
模块，该寄存器中其他值保持默认态，即SC_RMII_RESET_ST配置为0x1D。

步骤4 解复位RMII IP。

配置SC_RMII_RESET_ST（0xF200_0000+ 0xAE0）为0x00。

步骤5 配置端口状态获取方式UD_MAC_PORTSEL(0x200)，以太网交换模块可以自适应物理
层芯片工作状态或者由软件配置其工作状态。

Bits	Access	Name	Description	配置值
UD_MAC_PORTSEL: 0xF5A8_0000+0x0200				0x0000_00 01
[31:1]	RW	reserved	保留。	-
[0]	RW	stat_ctrl	端口工作状态信息 选择控制寄存器。 0: 使用从MDIO接 口获得的状态信 息; 1: 使用CPU设定的 状态信息。	1

步骤6 （可选）如果配置端口通过软件配置工作状态，则软件需要根据实际运用环境配置
UD_MAC_PORTSET（0x208）中的速度、连接状态和双工模式状态信息，同时将这些
信息配置到物理层芯片的相关寄存器中。

Bits	Access	Name	Description	配置值
UD_MAC_PORTSET: 0xF5A8_0000+0x0208				0x0000_0007
[31:3]	RW	reserved	保留。	-



Bits	Access	Name	Description	配置值
[2]	RW	speed_stat_dio	CPU设定端口速度模式。 0: 10Mbit/s模式; 1: 100Mbit/s模式。	1
[1]	RW	link_stat_dio	CPU设定端口链接状态。 0: 无链接; 1: 有链接。	1
[0]	RW	duplex_stat_dio	CPU设定端口双工模式。 0: 半双工模式; 1: 全双工模式。	1

步骤7 配置UD_GLB_QLEN_SET（0x344）寄存器中的输入队列深度和输出队列深度。

1. 输入队列深度表示接收数据时可以缓存的帧的最大的数量。
2. 输出队列深度表示发送数据时可以缓存的帧的最大的数量。

由于输入队列和输出队列共用64个管理空间，因此设置的输入队列深度和输出队列深度之和不能超过64，输入队列深度和输出队列深度至少为1。如果软件设置的输入队列深度和输出队列深度之和大于64，则优先保证软件先配置的项，另外一项则为64减去先配置项的结果。

Bits	Access	Name	Description	配置值
UD_GLB_QLEN_SET: 0xF5A8_0000+0x0344				0x2020
[31:14]	RW	reserved	保留。	-
[13:8]	RW	iq_len	输入（收包）队列长度配置。 注意： iq_len不得配置为0，否则将会强行置1。 配置的iq_len和eq_len之和不得大于64，否则优先满足iq_len的值（非0），eq_len取64-iq_len。	0x20
[7:6]	RW	reserved	保留。	-
[5:0]	RW	eq_len	输出（发包）队列长度配置。 注意： eq_len不得配置为0，否则将会强行置1。	0x20



步骤8 初始化接收帧队列缓冲区，复位后，软件需要申请与配置的输入队列深度个数相同的缓冲区，每个缓冲区大小为2KB，然后将缓冲区首地址依次写入到帧输入队列中，写入的次数应刚好等于配置的输入队列的深度。软件需要保证配置的首地址是以Word为单位，并且保证在收帧时不能释放该缓冲区。软件需要在内部建立一个索引表，索引地址为对应的输入队列的地址，索引项内容为分配给该输入队列地址的缓冲区的首地址。

Bits	Access	Name	Description	配置值
UD_GLB_IQ_ADDR: 0xF5A8_0000+0x358				-
Bits	Access	Name	Description	-
[31:0]	RW	startaddr_iq	CPU配置的输入帧对应的存储空间的首地址。接收帧根据此地址来申请总线。	-

步骤9 （可选）如果软件通过中断来收、发报文，配置中断使能寄存器GLB_IRQ_ENA（F5A8+0x1334），具体参见[中断配置](#)。

步骤10 （可选）配置以太网交换模块模块软复位，可以对以太网交换模块内部的逻辑电路和帧管理队列进行复位，使以太网交换模块模块回到初始化状态，但以太网交换模块内部所有的寄存器保留软复位前的值。软件解除软件复位后，需要重新申请收包缓冲区并对输入队列进行初始化，否则以太网交换模块不能接收网络包。

Bits	Access	Name	Description	配置值
GLB_SOFT_RESET: 0xF5A8_0000+0x1308				-
[31:1]	RW	reserved	保留。	-
[0]	RW	soft_reset	内部软复位寄存器，该软复位不复位寄存器。 0：不进行软复位； 1：进行软复位。 进入软复位状态后，若要撤消软复位，需将该位置0。	-

说明

- 以太网交换模块软复位后，软件设置的寄存器是不变的。具体哪些寄存器可以软复位，请参见每个寄存器的说明。
- 默认情况下，MAC转发表和VLAN转发表不使能，如需使用，请直接配置相关寄存器。

----结束

10.4.2 收帧过程

初始化时软件需要完成以下操作：



1. 软件需要申请与配置的输入队列深度相同数量的缓冲区，每个缓冲区的大小为 2KB，并将这些缓冲区的起始地址依次写入帧输入队列中，写入次数应等于配置的输入队列深度。
2. 软件需要确保在收帧过程中配置的缓冲区不会被释放。如果配置的起始地址不是按Word对齐的，则需确保该地址对应的字节地址是可以写入的。
3. 软件需要在内部建立一个索引表，其中索引地址对应于输入队列的地址，索引项的内容为分配给该输入队列地址的缓冲区的起始地址。

CPU通过帧描述符获取帧存放的地址索引，软件需要通过内部的接收队列查找表来获得当前接收帧缓冲区的起始地址。该查找表在初始化输入队列时建立，并且每当分配一个新的队列时都需要更新。如果该查找表与以太网交换模块硬件的输入队列不匹配，将会导致收帧错误，此时软件可以通过读取以太网交换模块输入队列的当前读写地址来进行校正，或者进行软复位以重建索引表。CPU可以采用中断方式或轮询方式来接收帧。

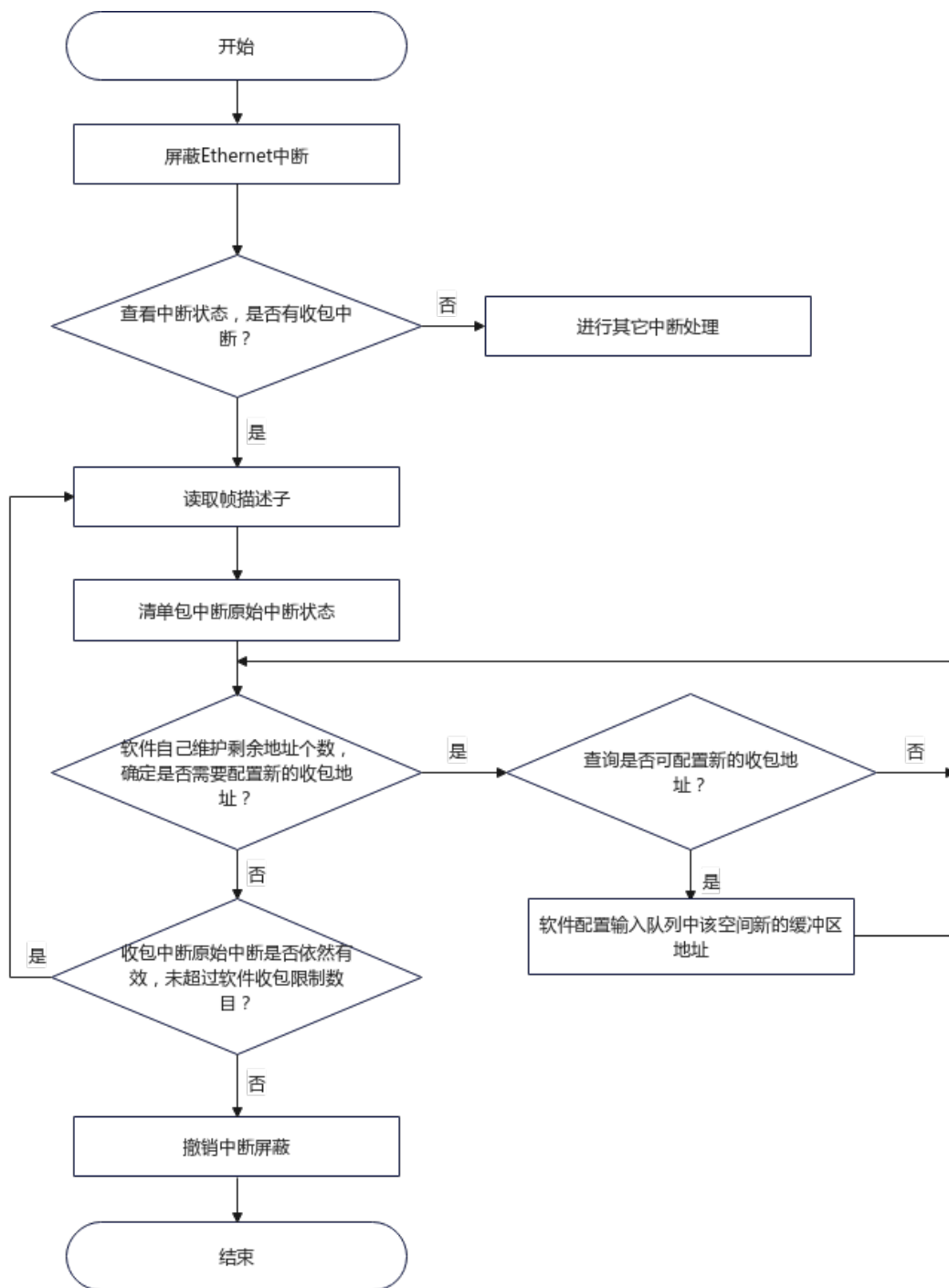
表 10-3 MAC CPU 接收帧描述子数据结构

位	名称	描述
[31:20]	fd_vlanid	从输入帧中的VLAN TAG拷贝过来的VLAN ID（不包括VLAN TAG中的Pri和CFI域）。
[19]	fd_vid_vld	指示帧描述子中是否带VID域，即帧描述子中是否含有fd_vlanid域。 0：帧描述子中不带VID域（fd_vlanid）； 1：帧描述子中带VID域（fd_vlanid）。
[18]	frm_vlan_vld	指示输出给CPU端口时的报文中是否带VLAN TAG。 0：输出报文中不带VLAN TAG； 1：输出报文中带VLAN TAG。
[17:12]	fd_in_addr	输入队列（iq）待接收帧帧的相对地址，作为帧存放绝对地址的索引值（0~iq_len-1）。
[11:0]	fd_in_len	输入队列待接收帧帧长度。

10.4.2.1 中断方式收帧

CPU使能帧接收中断时，根据是否有帧需要接收，硬件会产生帧接收中断（单包中断和多包中断）int_rx_up（每接收1个报文上报1个中断）和int_rxd_up（每接收到指定报文数目时上报1个中断）通知CPU收帧。中断方式收帧流程如图10-1所示。

图 10-1 中断方式收帧流程



步骤1 进入中断处理程序后，屏蔽以太网交换模块中断GLB_IRQ_ENA（0x1334）。

步骤2 查看中断状态GLB_IRQ_STAT（0x1330）[int_rx_up]是否有收帧中断。如果有，则转**步骤3**；如果没有，则进行其他以太网交换模块中断处理。

步骤3 读取帧描述子UD_GLB_IQFRM_DES（0x354），根据fd_in_addr对应的帧首地址读取对应长度（fd_in_len）的帧数据。



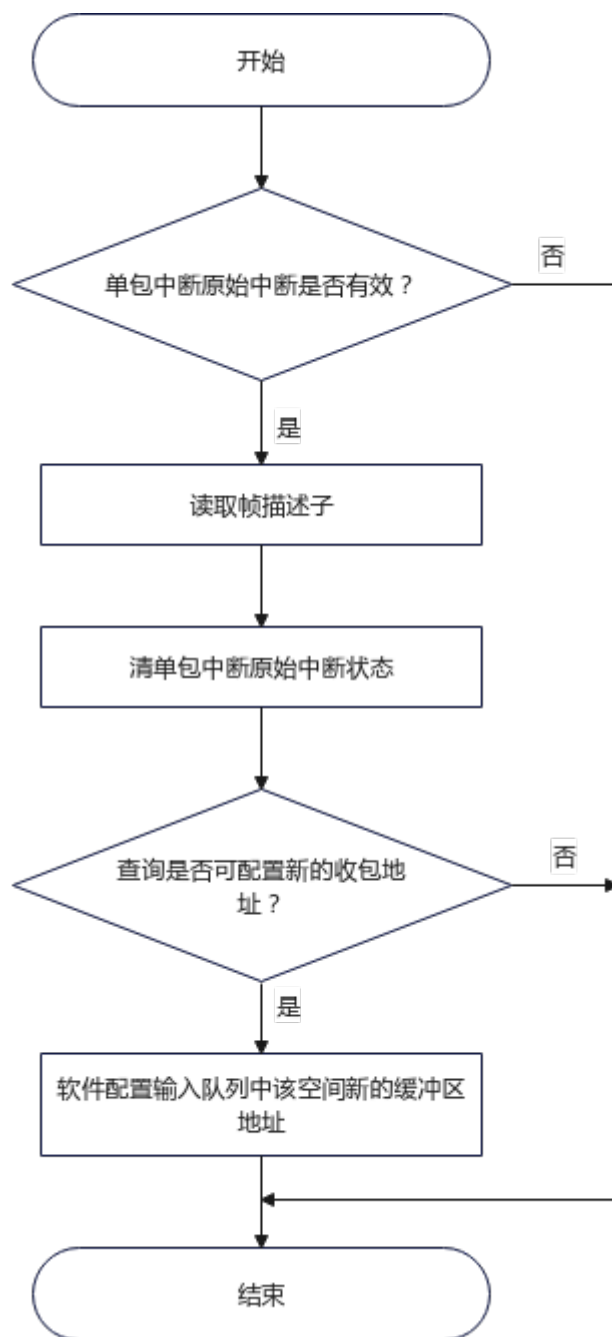
- 步骤4** 清单包中断原始中断信号GLB_IRQ_RAW（0x1338）[iraw_rx_up/]为0，通知硬件接收完成。
- 步骤5** 软件根据自己维护的剩余可用地址个数，确定是否需要配置新的收包地址。如果不需要配置，则转**步骤3**。完成一次收包。
- 步骤6** 读取UD_GLB_ADDRQ_STAT（0x36C）[cpu_addr_in_rdy]，查看是否可配新的收包地址。如果不可配，则返回**步骤5**。
- 步骤7** 软件通过寄存器UD_GLB_IQ_ADDR（0x358）将新的缓冲区地址配置给输入队列。返回**步骤5**。
- 步骤8** 读取并判断单包中断原始中断信号GLB_IRQ_RAW（0x1338）[iraw_rx_up]。如果该位有效且软件可继续收包（未超过收包限制数目），则返回**步骤3**。
- 步骤9** 撤销以太网交换模块中断屏蔽。

----结束

10.4.2.2 查询方式收帧

不使能帧接收中断GLB_IRQ_ENA（0x1334）[ien_cpu_rx]，CPU自动查询单包中断原始中断信号GLB_IRQ_RAW（0x1338）[iraw_rx_up]，为1表示有帧需要CPU接收。查询方式收帧流程如**图10-2**所示。

图 10-2 查询方式收帧流程图



步骤1 读取单包中断中断信号GLB_IRQ_RAW (0x1338) [iraw_rx_up]。如果该位无效，则直接结束。

步骤2 读取帧描述子UD_GLB_IQFRM_DES (0x354)，根据fd_in_addr对应的帧首地址读取对应长度 (fd_in_len) 的帧数据。

步骤3 写1清除单包中断原始中断信号GLB_IRQ_RAW (0x1338) [iraw_rx_up]，通知硬件接收完成。

步骤4 读取UD_GLB_ADDRQ_STAT (0x36C) [cpu_addr_in_rdy]，查看是否可配新的收包地址。如果无效，则直接结束。



步骤5 软件通过寄存器UD_GLB_IQ_ADDR（0x358）将新的缓冲区地址配置给输入队列。

----结束

10.4.3 发帧过程

CPU有帧待发送时，判断当前队列是否有空间发送。如果有空间发送，则分别写该帧Buffer的首地址和帧长到输出队列帧描述子中。CPU先写入首地址，然后写入发送帧的帧长。写发送帧的帧长触发硬件将该输出帧的首地址和帧长信息写入到输出队列中等待发送。因此，软件应该控制不能任意写帧长寄存器，每写该寄存器一次会导致发送一个数据包。

帧格式如下：

目的MAC	源MAC	Type	Data	FCS
-------	------	------	------	-----

CPU发送的帧在SDRAM中缓存时，不包括帧描述子。写帧描述子到UD_GLB_EQ_ADDR（0x360）及UD_GLB_EQFRM_LEN（0x364），通知以太网交换模块将该帧（描述子）入队。查询UD_GLB_ADDRQ_STAT（0x36C）可获得当前CPU发送队列使用情况。CPU发送帧描述子数据结构如表10-4所示。

表 10-4 RMII CPU 发送帧描述子数据结构

位	名称	描述
[31:0]	start_addr_eq	发送一个帧的首地址。
[10:0]	fm_len	帧长度，单位：Byte。

须知

fm_len小于20 Bytes以及大于1600 Bytes的帧将被丢弃，即可发送范围在20 ~ 1600之间。

如果需要发送带VLAN TAG的帧，可以将VLAN配置寄存器中由逻辑添加VLAN发送出去，此种方式下逻辑只能添加1种VLAN TAG；如果需要添加多种VLAN TAG时，需要软件将VLAN TAG添加好以后，由逻辑重新计算CRC后发送出去。

CPU发帧可以采用中断方式或查询方式。

10.4.3.1 中断方式发帧

如果软件需要发送帧，但是当前输出队列已满，则软件可以使能该中断。当输出队列为空后，产生中断，通知软件发送等待的帧。软件可以使用该中断一次发送一组帧，同时在该中断有效时释放上次发送的一组帧的缓冲区。

步骤1 CPU使能以太网交换模块发包队列由不空到空中断，配置GLB_IRQ_ENA（0x1334）[int_freeq_up]为0b1，其余bit保持不变。

步骤2 等待中断产生，中断产生后说明以太网交换模块输出队列由不空到空，可以发送帧。



步骤3 配置待输出帧的首地址UD_GLB_EQ_ADDR（0x360）。

步骤4 配置待输出帧的长度UD_GLB_EQFRM_LEN（0x364），完成1帧的发送配置。

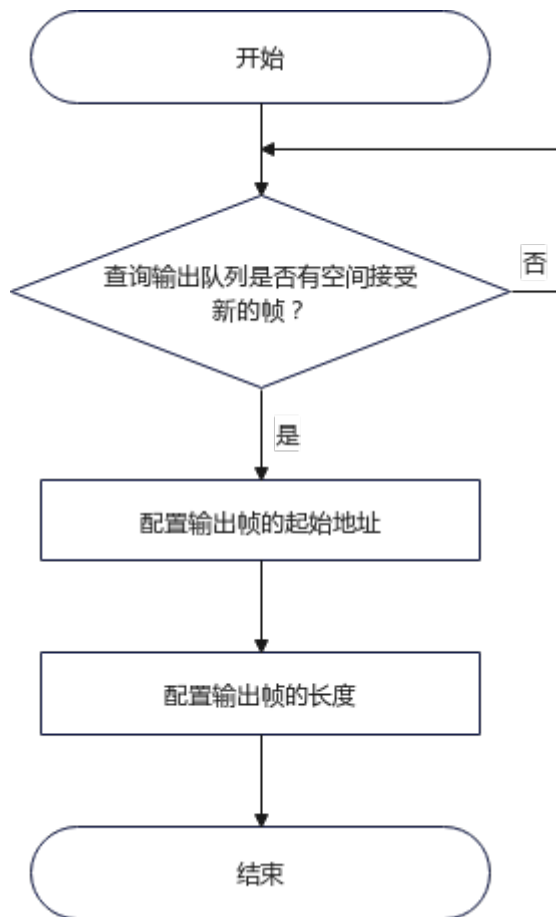
----结束

10.4.3.2 查询方式发帧

软件查询自己内部的发帧计数，如果计数小于配置的输出队列长度，则可以直接将待发送的帧通知以太网交换模块，同时建立对应的发帧索引表，索引表项内容为写入到以太网交换模块输出队列的输出帧首地址。当以太网交换模块发送完1帧后，通过输出队列的地址通知CPU释放对应的发送缓冲区，CPU通过该输出队列的地址查找到对应的发送缓冲区并释放。

CPU发帧流程图如图10-3所示。

图 10-3 CPU 发帧流程图



步骤1 读取UD_GLB_ADDRQ_STAT（0x36C）[eq_in_rdy]，检查以太网交换模块输出队列是否有剩余空间接收新的发送帧。如果没有，则继续等待并查询。

步骤2 配置待输出帧的首地址UD_GLB_EQ_ADDR（0x360）。

步骤3 配置待输出帧的长度UD_GLB_EQFRM_LEN（0x364），完成1帧的发送配置。

----结束



10.4.4 MAC 转发配置（可选）

- 步骤1** 对于输入的广播帧，按照寄存器GLB_MACTCTRL（0x1334）设置来决定是否转发到CPU端口和另一外部端口。
- 步骤2** 对于目的MAC地址等于本机MAC地址的输入帧，直接转发给CPU端口（配置强制转发时该输入帧也可以转发到下行端口）。
- 步骤3** 对于目的MAC地址不等于本机MAC地址的输入帧，使用目的MAC地址查MAC转发控制表，若匹配到MAC转发控制表，按照表项设置进行转发；若没有匹配到MAC转发控制表，（区分组播帧和单播帧）按照寄存器GLB_MACTCTRL（0x1314）设置进行转发。

Bits	Access	Name	Description
GLB_MACTCTRL:0xF5A8_0000+0x1314			
[31:8]	RW	reserved	保留。
[7]	RW	mact_ena_up	端口所有MAC过滤器总使能位。 0：不使用任何MAC过滤器； 1：使用MAC过滤器。
[6]	RW	reserved	保留。
[5]	RW	broad2cpu_up	端口输入的广播帧是否转发给CPU端口。 0：不转发给CPU端口； 1：转发给CPU端口。
[4]	RW	reserved	保留。
[3]	RW	multi2cpu_up	端口输入的非过滤表中的组播帧是否转发给CPU端口。 0：不转发给CPU端口； 1：转发给CPU端口。
[2]	RW	reserved	保留。
[1]	RW	uni2cpu_up	端口输入的非过滤表中的单播帧是否转发给CPU端口。 0：不转发给CPU端口； 1：转发给CPU端口。
[0]	RW	reserved	保留。

- 步骤4** 可通过地址GLB_MAC0_L32～GLB_MAC7_H16（地址0xF5A8_1400～0xF5A8_143C）来配置MAC转发控制表，MAC转发控制表配置方法如下。
1. 本机MAC通过地址GLB_HOSTMAC_L32和GLB_HOSTMAC_H16来配置。
 2. MAC0表项通过地址GLB_MAC0_L32和GLB_MAC0_H16来配置。
 3. MAC1表项通过地址GLB_MAC1_L32和GLB_MAC1_H16来配置。
 4. MAC2表项通过地址GLB_MAC2_L32和GLB_MAC2_H16来配置。



5. MAC3表项通过地址GLB_MAC3_L32和GLB_MAC3_H16来配置。
6. MAC4表项通过地址GLB_MAC4_L32和GLB_MAC4_H16来配置。
7. MAC5表项通过地址GLB_MAC5_L32和GLB_MAC5_H16来配置。
8. MAC6表项通过地址GLB_MAC6_L32和GLB_MAC6_H16来配置。
9. MAC7表项通过地址GLB_MAC7_L32和GLB_MAC7_H16来配置。

----结束

10.4.5 软复位配置

RMII模块支持通过系统控制器进行软复位操作。

- 步骤1** 配置RMII寄存器UD_MAC_PORTSET (0xF5A8_0208) bit[1]=0。关闭RMII端口链路状态，使软件无法收包，同时软件不再发包。
- 步骤2** 等到软件处理完上行、下行端口的当前收发包后，清除收发包队列，同时队列长度保持软复位之前的值，相关指针、队列计数值均要回0。
- 步骤3** 配置系统控制寄存器SOFT_RST0_CTL (0xF5A8_2044) 的bit[16]=0b1,bit[29]=0b1，复位RMII。
- 步骤4** 配置系统控制寄存器SOFT_RST0_CTL (0xF5A8_2044) 的bit[16]=0b0,bit[29]=0b0，解复位RMII。
- 步骤5** 如果要重新开始收发包，重新走初始化流程。

----结束

10.4.6 内环配置

配置使能UD_MAC_SET的in_loop_en为1。

10.4.7 中断配置

10.4.7.1 中断状态寄存器

指示产生的中断类型，具体内容请参见GLB_IRQ_STAT (0xF5A8_0000+0x1330)。

10.4.7.2 中断使能寄存器

控制是否产生相应的中断，具体内容请参见GLB_IRQ_ENA (0xF5A8_0000+0x1334)。如果使能某中断，其中断状态会写入到相应的中断状态寄存器中。

10.4.7.3 原始中断状态寄存器

可以读取某种类型的原始中断传送给CPU，具体内容请参见GLB_IRQ_RAW (0xF5A8_0000+0x1338)。中断状态清除必须清除该中断的原始中断，原始中断清除后，中断状态会自动清除掉。



10.4.8 RMII 接口管脚复用配置

RMII管脚均为非默认管脚，需要进行管脚复用配置。（SD7601 IO_CONFIG 寄存器列表）



11 LOCAL BUS

11.1 概述

Local Bus可对外提供异步静态存储器接口，可以连接SRAM（Static Random Access Memory）、ROM（Read Only Memory）、NOR FLASH等异步静态存储器，用于数据存储等功能。同时也可以用于连接带异步存储器接口（包括Intel时序和Moto时序）的控制芯片。

LocalBus总线时钟至少100MHz，最小可配置的读写访问周期要小于100ns，最大可配置的访问周期要大于1 μ s。

11.2 特性

Local Bus具有以下特性：

- 支持8个片选（CS0~CS7），每个片选存储器空间可独立配置，独立配置时序参数。
- 片选有效电平之间的最小间隔可配。
- 对应8套时序参数寄存器，时序参数的配置范围为1~64个内部工作时钟周期。
- 片选与读信号、写信号之间的有效长度及间隔可配置。
- 支持访问带有异步静态存储器接口（包括Intel时序和Moto时序）的控制芯片，Intel时序和Moto时序通过时序寄存器配置得到（注：Hi309a只支持数据线与时地址线复用的Moto模式接口）。
- 支持异步Page模式读操作。
- 支持地址段的写保护功能。
- 数据位宽支持16bits或8bits可配置。
- 支持数据线的D15为数据线最高位、D0为数据线最低位；
- 大小端可配置，在采用16bits数据时，如果D15为高位，数据线可配置为按照[15:8,7:0]或[7:0,15:8]的顺序传输。
- 无访问时，数据线默认输入。
- 支持通过寄存器配置片选的数据宽度。



- 支持28根地址线，支持地址偏移。
- 当A31为高位时，支持地址偏移，只对16bits数据位宽有效，当地址偏移使能位为0，则lb_addr[0]对应器件的addr[0]。如果地址偏移使能位为1，则lb_addr[1]对应器件的addr[0]。
- 启动片选的地址偏移选择，支持引脚电平可配。所有片选的地址偏移支持寄存器可配置。
- 支持独立的读使能信号OE_N。
- 支持独立的写使能信号WE_N。
- 支持字节的写使能信号LB_BS_N[1:0]，时序与WE_N相同。
- 支持读写合一信号LBTCL，用于Moto时序的器件。支持高读低写和高写低读寄存器可配置。
- 支持地址段的写保护功能，保护的地址范围寄存器可配置。并且支持保护地址段的写操作产生中断，中断可屏蔽。
- 支持WE_N、OE_N、LBTCL同时有效。
- 支持从CS片选有效到开始采外部Ready的时间可配置。
- 支持外部Ready（每个片选配置寄存器单独配置），支持32BIT的Ready超时保护机制（默认值配置为最大），默认使能，是否使能通过寄存器可配置。支持查询和中断模式，支持中断可屏蔽。
- 支持8个Ready超时状态寄存器（反映屏蔽后中断状态），8个中断屏蔽寄存器（中断可屏蔽），8个Ready超时原始状态寄存器（反映屏蔽前中断状态），各对应8个片选。
- 超时后置1状态寄存器，供查询状态，上报中断。同时将CS_N、WE_N、OE_N置为无效，将LBCTL恢复为默认值，结束本次操作。
- Local Bus总线支持PAGE读操作。
- CS与读信号、写信号之间允许1时钟周期的时序可配。
- 支持LAL（地址锁存指示有效信号）有效长度（Tlale）可配置（1-31Cycle）。
- 支持地址有效到片选有效长度（Tacs，0-31Cycle）可配置；支持地址有效到读使能（Taoe，0-31Cycle）、地址有效到写使能（Tawe，0-31Cycle）和地址有效到读写合一信号（Tal，0-31Cycle）有效长度可配置。
- 读操作时片选有效长度可配置（Tcsrp，2-511Cycle）；写操作时片选有效长度（Tcswp，2-511Cycle）可配置。
- 支持片选无效到读使能无效长度（Tcoen，0-63Cycle）可配置；支持片选无效到写使能无效长度（Tcwen，0-31Cycle）可配置；支持片选无效到读写合一信号恢复到默认值长度（Tcln，0-63Cycle）可配置；支持从片选无效到写数据无效时间（Tcdn，0-31Cycle）可配置。
- 支持片选无效到LAL有效或下一次操作地址有效之间的长度（Tcl，1-63Cycle）可配置。
- Page读时，支持从地址有效到第一次读操作之间的时间（Tcsrp，2-511Cycle）可配置，支持两次读操作之间的时间（Tprc，1-64Cycle）可配置。
- Page读时，支持从片选有效到开始采样外部Ready的时间（Tlgt_samp，1-63Cycle）可配置。
- 支持总空间最大2G访问，具体空间以系统分配空间为准。



11.3 关键术语

表 11-1 Local Bus 关键术语描述

术语	描述
SRAM	Static Random Access Memory，静态随机访问存储器。
ROM	Read Only Memory，只读存储器。

11.3.1 应用场景

Local Bus的应用如图11-1所示。

图 11-1 Local Bus 的典型应用框图

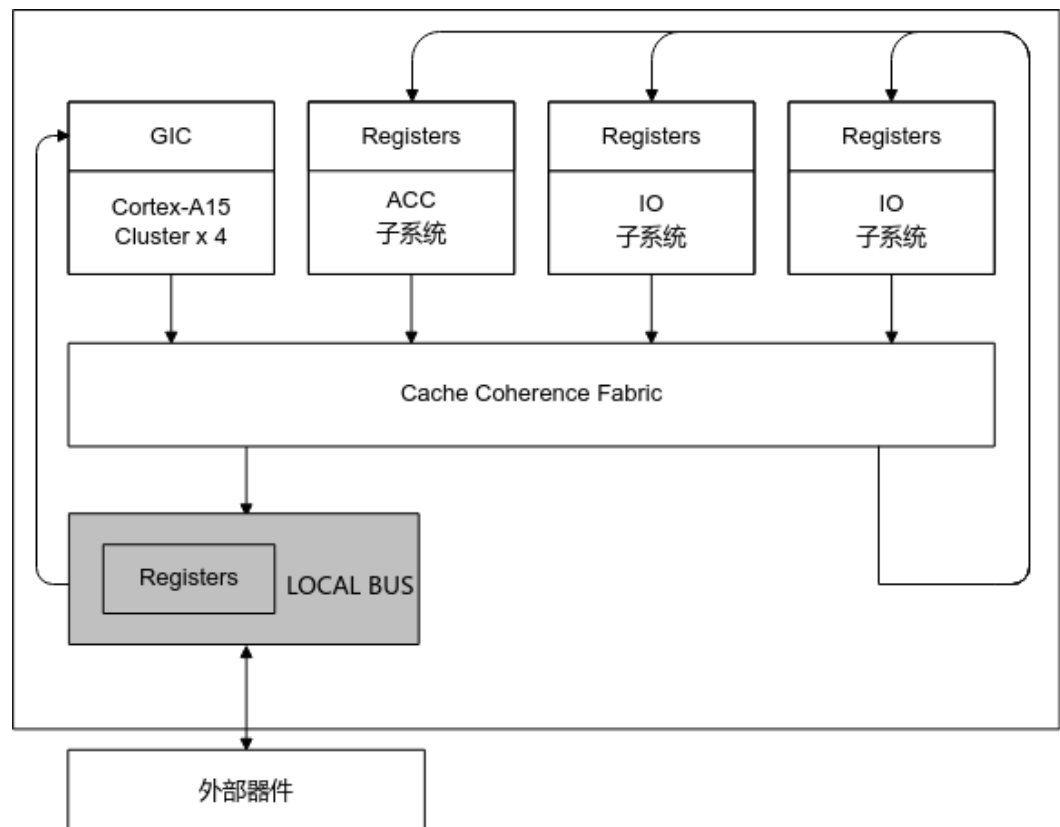


图11-1中的外部器件可以是SRAM、NORFLASH等静态存储器，也可以是Intel或Moto时序的器件。Local Bus作为总线从设备挂在总线上，CPU可以通过总线配置Local Bus的寄存器，或通过总线向Local Bus发起读写操作，Local Bus根据当前的时序参数配置，产生符合外部器件的时序，访问外部器件。



须知

为防止总线冲突，要求对接器件在片选（CS_N）输入无效的情况下，数据总线（DATA[15:0]）必须为输入或者高阻。

11.3.2 接口信号

Local Bus接口信号如表11-2所示。

表 11-2 Local Bus 接口信号描述

管脚	内部信号名称	方向	描述
BOOTWIDTH0	LB_DATA_WIDTH	I	Local Bus启动时存储接口数据位宽指示信号。 0：8bit； 1：16bit。
BOOTADDRSEL	LB_ADDR_OFFSET	I	Local Bus启动时存储接口地址是否偏移指示信号。与GPIO0复用。 0：不偏移； 1：偏移。
BOOTMULSEL（内部固定不出管脚）	LB_MUL_EN	I	地址线和数据线是否复用。 0：表示不复用； 1：表示复用。
LBREADY	LB_READY	I	外部输入Local Bus控制器的异步等待信号，有效电平可配。
LBBSL0~LBBSL1	LB_BS_N0~ LB_BS_N1	O	Local Bus控制器输出的Byte选择信号，低电平有效。
LBWEN	LB_WE_N	O	Local Bus控制器输出的写使能信号，低电平有效。
LBOEN	LB_OE_N	O	Local Bus控制器输出的读使能信号，低电平有效。
LBLBCTL	LB_LBCTL	O	Local Bus控制器输出的读写合一信号，高读低写还是低写高读寄存器可配置。
LBCS0N~LBCS7N	LB_CS_N_0~LB_CS_N_7	O	Local Bus控制器输出的通道0~7的片选信号，低电平有效。



管脚	内部信号名称	方向	描述
LBDQ0 ~ LBDQ15	LB_DATA0 ~ LB_DATA15	I/O	Local Bus控制器双向数据总线。
LBADR0 ~ LBADR27	LB_ADDR0 ~ LB_ADDR27	O	Local Bus控制器输出的地址总线。

须知

在数据线和地址线非复用情况，LBDQ0 ~ LBDQ15和LBADR0 ~ LBADR27是外部的数据和地址管脚，LBPADDR不作为外部管脚。

11.3.3 工作方式

11.3.3.1 灵活的时序配置

支持8套时序参数配置寄存器，能够连接8个外部设备，并且每个设备的时序都可以独立设置。与传统的Local Bus相比，无需选择Intel模式或Motorola模式进行配置，可以直接依据具体设备调整相应的时序。

时序参数配置分为非Page读模式和Page读模式两种类型：

- 非Page读模式：时序参数以地址有效作为时间基准点，从地址有效到片选、读使能、写使能（读使能和写使能适用于Intel模式）、读写合一信号（适用于Motorola模式）的有效时间均可配置；读写操作期间片选的有效长度也可以配置；片选无效与读使能无效、写使能无效、读写合一信号无效、写数据无效之间的时间间隔同样可以配置；两次操作（包括读读、读写、写读、写写）之间的间隔时间也可以配置。当连接具有外部Ready信号的设备时，外部Ready信号有效至片选无效的时间间隔也可以配置。
- Page读模式：从地址有效到片选、读使能的时间可以配置；从地址有效到首次读取数据的时间可以配置；首次读取后两次连续读取之间的时间间隔可以配置；两次Page读操作之间的间隔时间也可以配置。

11.3.3.2 异步(Ready)模式

支持异步（Ready）模式，其中外部Ready信号的有效状态（高电平或低电平）可以配置，并且从片选信号有效到开始采样外部器件的Ready信号之间的时间间隔也可以进行配置。

在异步等待模式下，系统具备防死锁机制，以避免因对接器件的Ready信号长时间无响应而导致系统陷入死锁状态。若从设备未能在规定时间内提供有效的Ready信号，LocalBus将在片选信号有效之后开始计时；一旦计时达到预设的CF_RDY_TIMEOUT值而Ready信号仍未有效，则系统将主动终止此次读取或写入操作，并向CPU发送中断信号。

不再接收



11.3.3.3 字节读写模式

以片选0为例说明，通过配置寄存器LB_CS0_CTRL[cf_rble_0](偏移地址为0x0020，bit[18])来控制LB_BS_N的时序，如果配置为0，表示字节使能信号LB_BS_N和写使能时序相同；如果配置为1，那么在写操作时，LB_BS_N时序与写使能时序相同，读操作时将一直为低。

11.3.3.4 数据字节顺序可配

数据字节顺序通过cf_byte_order配置，数据线序固定配置cf_data_order为0。在数据位宽为16bit和8bit形式时，表11-3给出了cf_byte_order和cf_data_order不同配置时，数据的输出值。

表 11-3 数据位宽 16bit 时的数据值

cf_byte_order=0 cf_data_order=0	cf_byte_order=1 cf_data_order=0
data[15:0]	{data[7:0],data[15:8]}

11.3.3.5 地址线序和地址偏移可配

表 11-4 外接器件的地址产生

外接器件数据位宽	外部器件地址
16bits	LBADR31 ~ LBADR0=CPU地址右移一位高位补0。
8bits	LBADR31 ~ LBADR0=CPU地址。

外接器件的地址和外接器件的数据位宽和总线上的地址有关，具体关系如表11-4所示。

Local Bus地址信号的A0是最低位。在8bit模式下，A0是最低位，单位为Byte。而在16bit模式下，A0也是最低位，但此时的单位为Half Word。需要注意的是，这种定义与 Freescale 的 PowerPC 芯片 Local Bus 地址信号有所不同：在 PowerPC 芯片中，8 bit模式时 A0 为最低位，单位为Byte；但在 16 bit模式下，A1 才是最低位，单位为Half Word。因此，为了兼容现有的某些 PowerPC 芯片背板应用，可以通过系统控制器来配置 Local Bus 地址是否需要进行偏移。该偏移设置可以针对不同的片选独立配置。

假设地址位宽为28bit，数据位宽为16bit，地址偏移通过cf_addr_offset配置。则在cf_addr_offset不同配置时，管脚处的地址如表11-5所示。

表 11-5 cf_addr_offset 不同配置

外接器件的数据位宽	cf_addr_offset=0	cf_addr_offset=1
8bit	lb_addr[0]是地址的低位	lb_addr[0]是地址的低位。



外接器件的数据位宽	cf_addr_offset=0	cf_addr_offset=1
16bit	lb_addr[0]是地址的低位	lb_addr[1]是地址的低位。

图 11-2 16bit cf_addr_offset=1 时

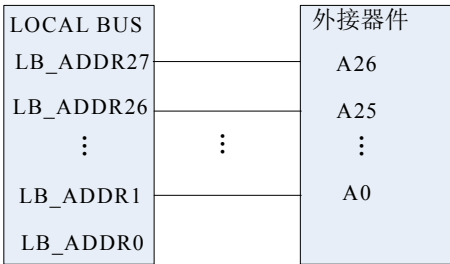
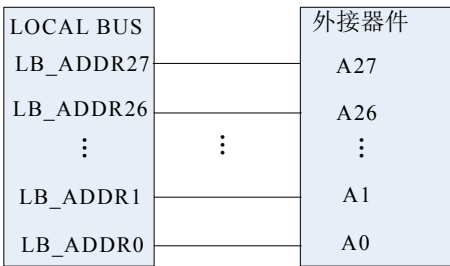


图 11-3 8bit 或 16bit (cf_addr_offset=0) 时



11.3.3.6 NorFLASH 写保护功能

- NOR FLASH器件接口一般带有一个WP#输入管脚，实现对NOR FLASH的硬件写保护功能。Local Bus不支持该信号控制。
- Local Bus支持写保护功能，对于写保护的使能有两种方式，一种是低级写保护使能，一种是高级写保护使能。通过LB_CTRL(偏移地址0x00CC，bit[15:0])中的相应bit位来决定，例如地址段0的写保护使能，配置LB_CTRL(偏移地址0x00CC，bit[15:14])，配置为01，表示低级写保护使能，配置为10高级写保护使能。如果是高级写保护。还需要配置写保护使能关键字LB_HL_PROT_KEY0 [cf_hl_prot_key0] (偏移地址为0x0130，bit[31:16])。
- 如果写保护地址段中包含命令字地址，且需要对命令字地址可以进行写操作，需要配置命令字地址LB_NOR_CMDi_ADDR(i=0-3) (偏移地址为0x0110、0x118、0x120、0x128) 和相应的掩码LB_NOR_CMDi_MASK(i=0-3) (偏移地址为0x0114、0x11C、0x124、0x12C)，命令字地址和相应掩码共有四对，可根据写保护地址段的命令字个数选择。
- Local Bus最多可以配置8个写保护地址段，如果使能了写保护功能，却对写保护地址进行了操作（命令字地址除外，名字地址可以正常写入），会产生中断。



11.3.3.7 NorFLASH 复位

NOR FLASH器件接口一般带有一个RST#输入管脚，实现对NOR FLASH的复位功能。
Local Bus不支持该信号控制。

11.3.3.8 Local Bus 时序参数配置

可配置的时序参数如表11-6所示。

表 11-6 Local Bus 的可配置时序参数表

符号	说明	最小值	最大值	单位
Tacs	从地址有效到片选有效的时间。	0	31	Cycle
Tcsrp	在读操作时，表示片选有效时间。在Page读操作时，表示从地址有效第一次读取数据的时间。	2	511	Cycle
Taoe	地址有效到读使能（oe_n）有效的时间	0	31	Cycle
Tcoen	如果读使能（oe_n）先于片选（cs_n）无效，表示读使能（oe_n）无效到片选（cs_n）无效的时间； 如果片选（cs_n）先于读使能（oe_n）无效，表示片选（cs_n）无效到读使能（oe_n）无效的时间。	0	60	Cycle
Tal	从地址信号有效到读写合一信号（lbctl）有效（如：默认为写，假设为高，则低有效）时间	0	31	Cycle
Tawe	从地址信号有效到写使能信号有效的时 间。	0	31	Cycle
Tcln	如果读写合一信号（lbctl）先于片选（cs_n）无效，表示从读写合一信号（lbctl）无效（恢复默认值）到片选（cs_n）无效时间； 如果片选（cs_n）先于读写合一信号（lbctl）无效，表示从片选（cs_n）无效到读写合一信号（lbctl）无效的时 间。	0	60	Cycle
Tcswp	写操作时，片选有效时间。	2	511	Cycle
Tcwen	如果写使能（we_n）先于片选（cs_n）无效，表示写使能（we_n）无效到片选（cs_n）无效的时间； 如果片选（cs_n）先于写使能（we_n）无效，表示从片选无效到写使能无效的 时间。	0	31	Cycle
Tcdn	从片选无效到写数据无效的时间	0	31	Cycle



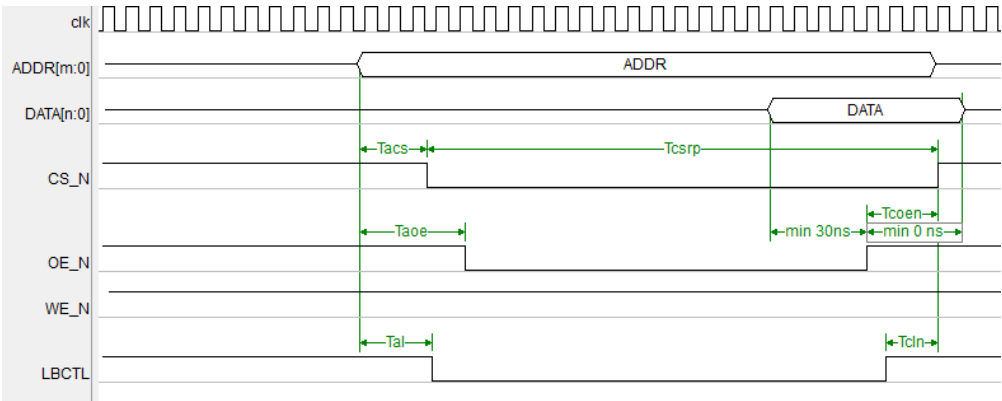
符号	说明	最小值	最大值	单位
Tcl	在非复用情况下，表示从片选无效到地址有效的时间。	3	63	Cycle
Tprc	Page读时，两次读的时间间隔。	1	31	Cycle
Tlgta_samp	从片选有效到开始采样外部Ready（LGTA）的时间。如果从片选有效开始采，则此值设置为1。	1	63	Cycle

11.3.3.9 Local Bus 典型时序

Local Bus可以配置为地址线和数据线复用或非复用，外部器件是带Ready的器件还是不带Ready的器件。下面各图将列出各种情况的时序，以及各种情况下的配置参数。各个参数的具体含义可以在[表11-6](#)中查找。

- 不复用、无外部Ready情况

图 11-4 Local Bus 不复用无外部 Ready 单次读操作



须知

不同项目略有差异。

因为Local Bus接口为异步接口，所以在应用或者设置Local Bus时序参数时，需要考虑Local Bus IP从芯片内部和PAD之间的延迟，上述两个表列出了延迟参数的最大最小值。

输出方向信号寄存器到PAD_I最大延时6.94ns，PAD最大延时7ns。

输入方向PAD到寄存器最大延时4.47ns，Oe_n上升沿前一个时钟周期采样，5.2ns(192M)

封装延时：pin delay 0.2ns。

总延时：23.81ns。

预留6.19ns余量。数据有效和oe_n上升沿之间，最小30ns。

SI的影响没有列入（边沿不单调或边沿台阶的影响，需要单板硬件考虑）

片选上升沿与数据无效之间最小是0ns。

图 11-5 Local Bus 不复用无外部 Ready 单次写操作

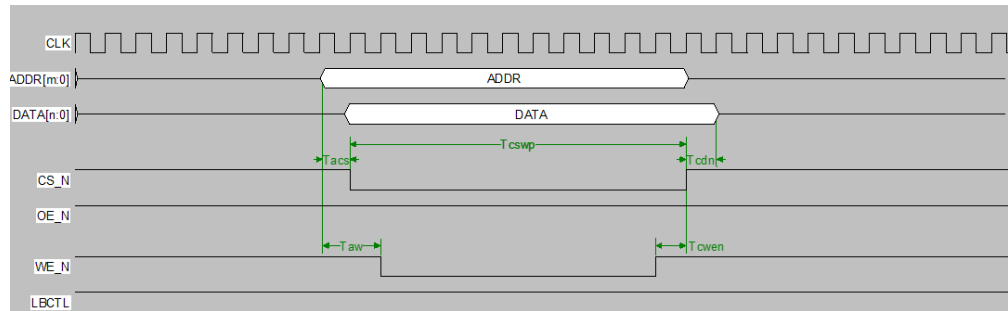


图 11-6 Local Bus 不复用无外部 Ready 连续读操作

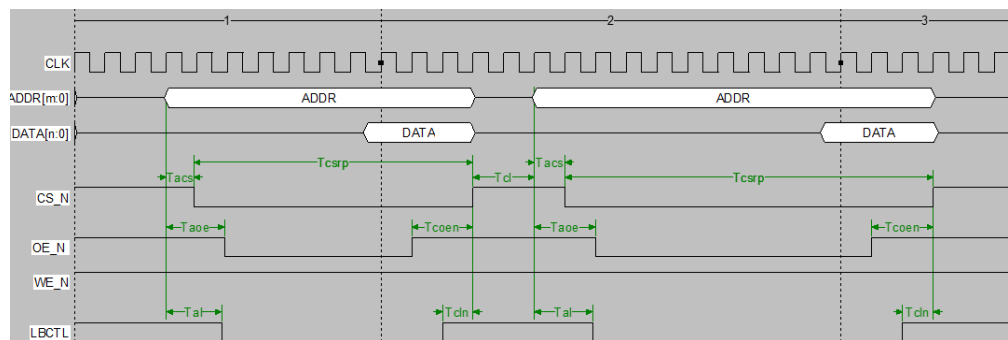


图 11-7 Local Bus 不复用无外部 Ready 连续写操作

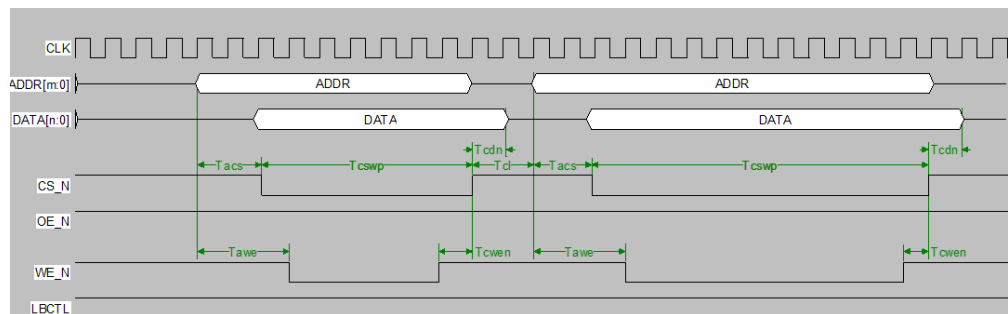


图 11-8 Local Bus 不复用无外部 Ready 读写切换操作

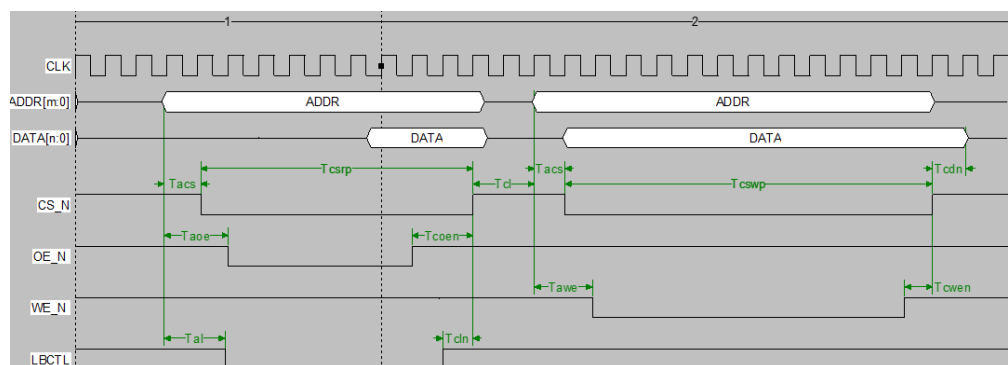


图 11-9 Local Bus 不复用无外部 Ready 写读切换操作

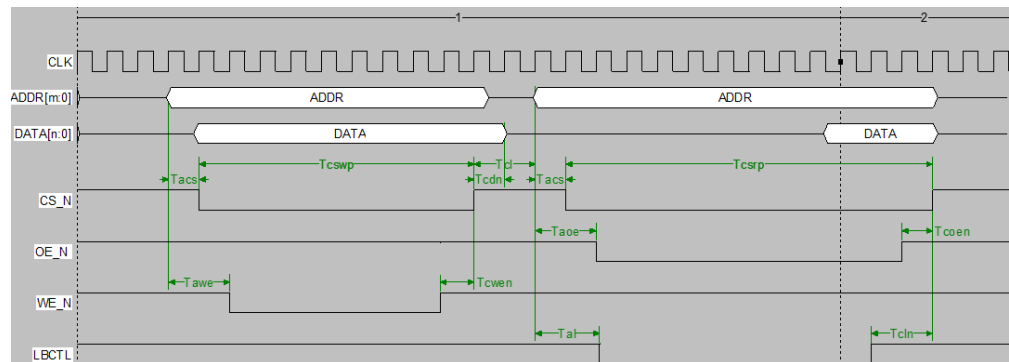
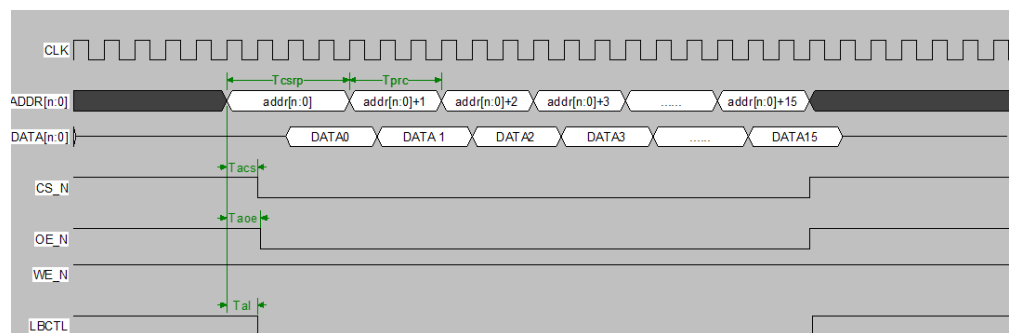


图 11-10 Local Bus 不复用无外部 Ready Page 读操作



- 不复用、有外部Ready情况

图 11-11 Local Bus 不复用有外部 Ready 单次读操作

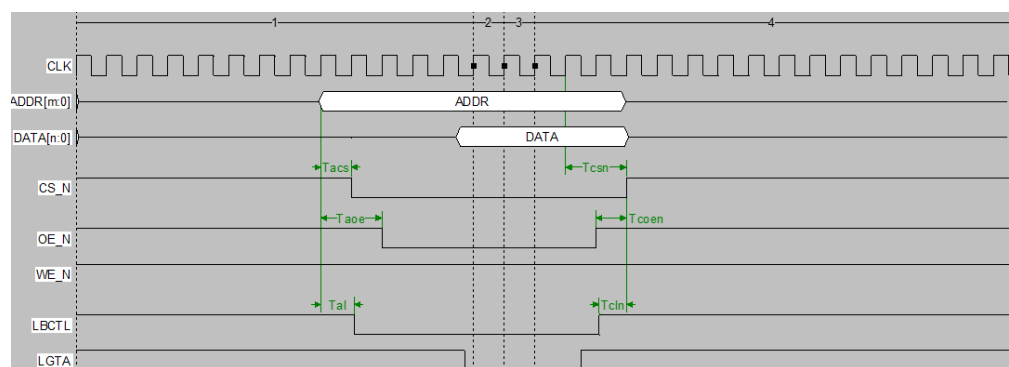


图 11-12 Local Bus 不复用有外部 Ready 单次写操作

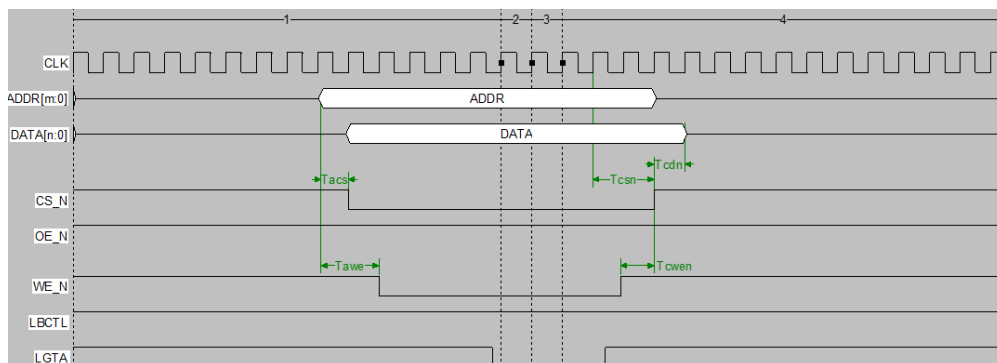


图 11-13 Local Bus 不复用有外部 Ready 连续读操作

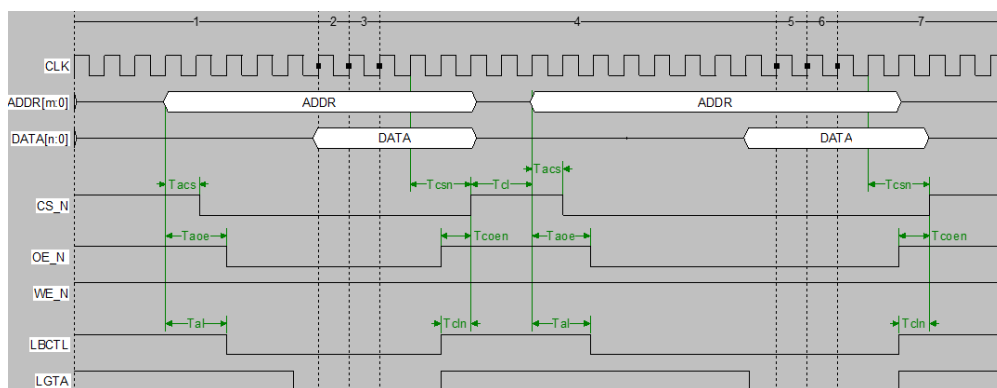


图 11-14 Local Bus 不复用有外部 Ready 连续写操作

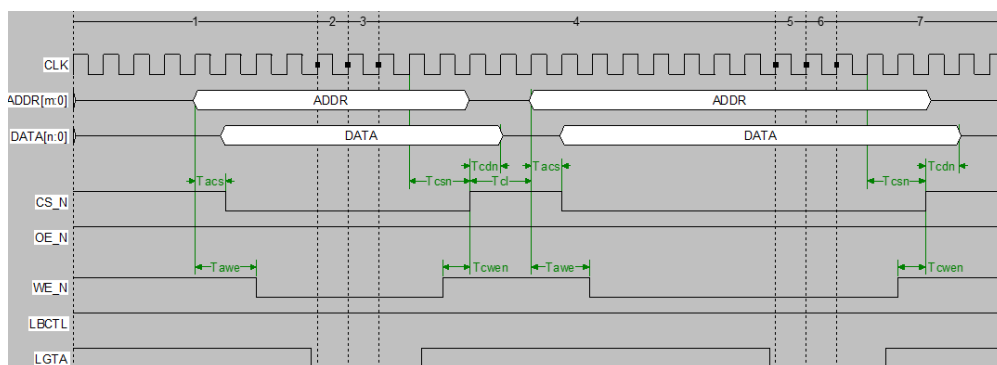


图 11-15 Local Bus 不复用有外部 Ready 读写操作

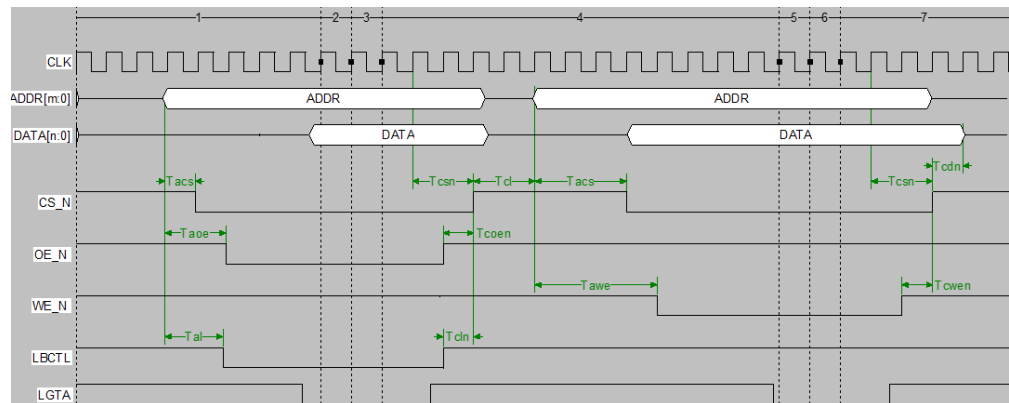


图 11-16 Local Bus 不复用有外部 Ready 写读切换操作

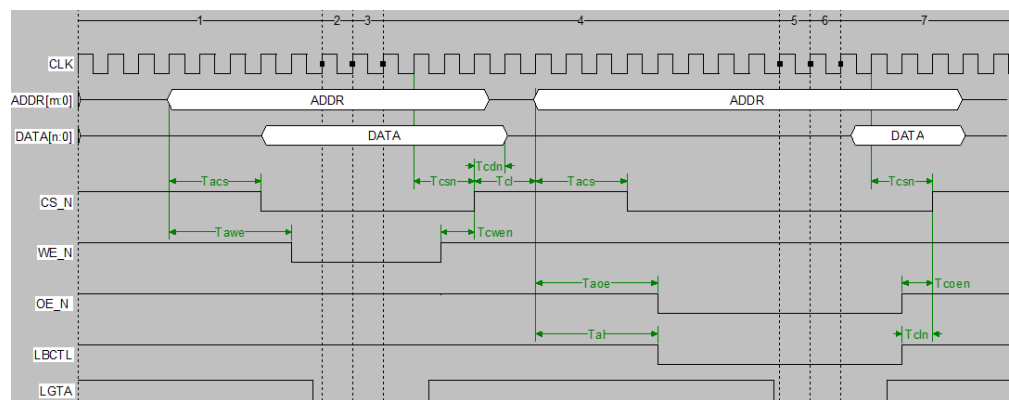
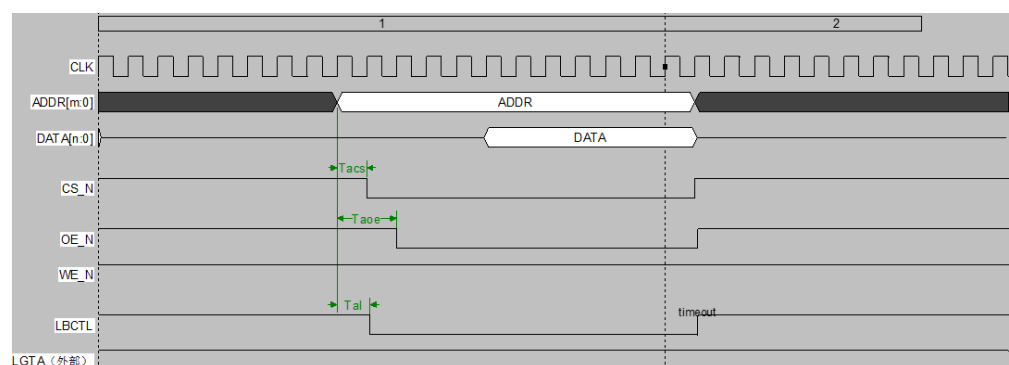


图 11-17 Local Bus 不复用有外部 Ready 外部 TA 超时



- 复用、无外部Ready情况

图 11-18 Local Bus 复用无外部 Ready 单次读操作

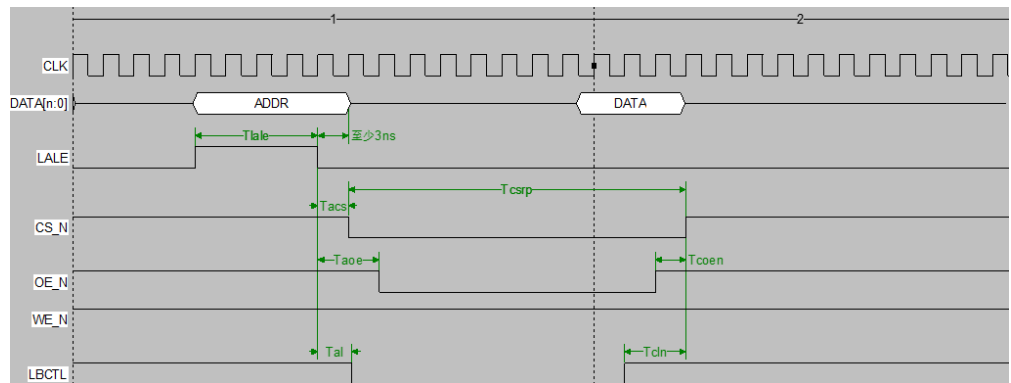


图 11-19 Local Bus 复用无外部 Ready 单次写操作

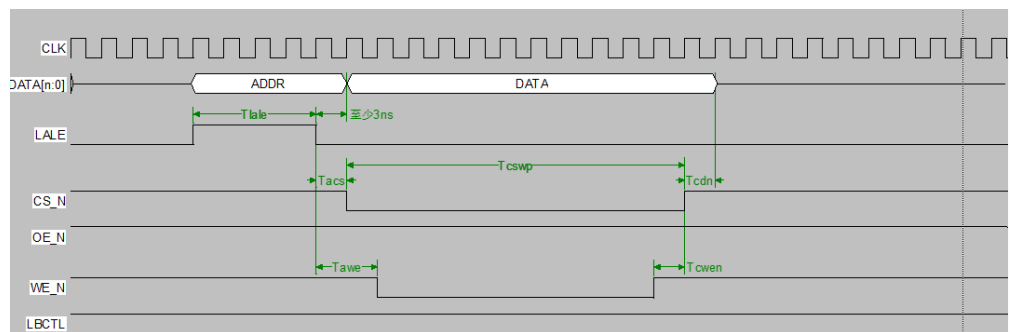


图 11-20 Local Bus 复用无外部 Ready 连续读操作

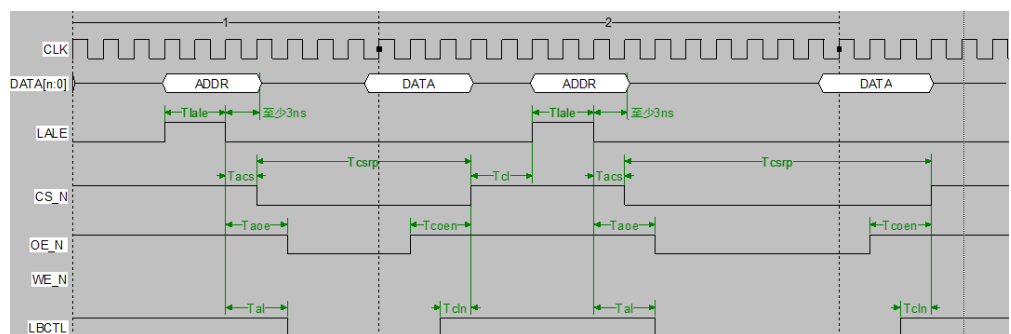




图 11-21 Local Bus 复用无外部 Ready 连续写操作

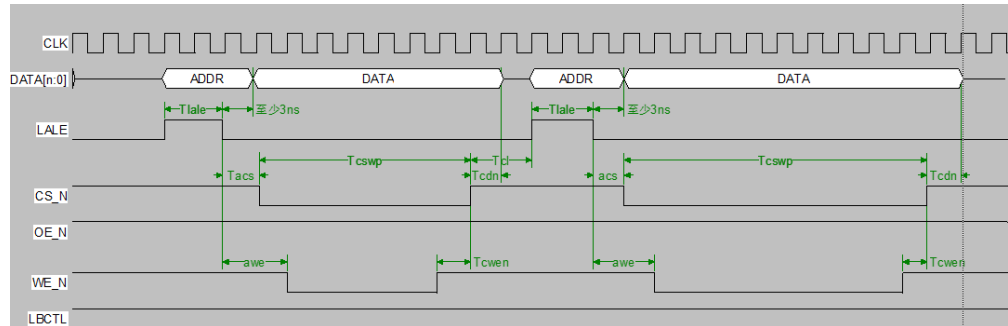


图 11-22 Local Bus 复用无外部 Ready 读写切换操作

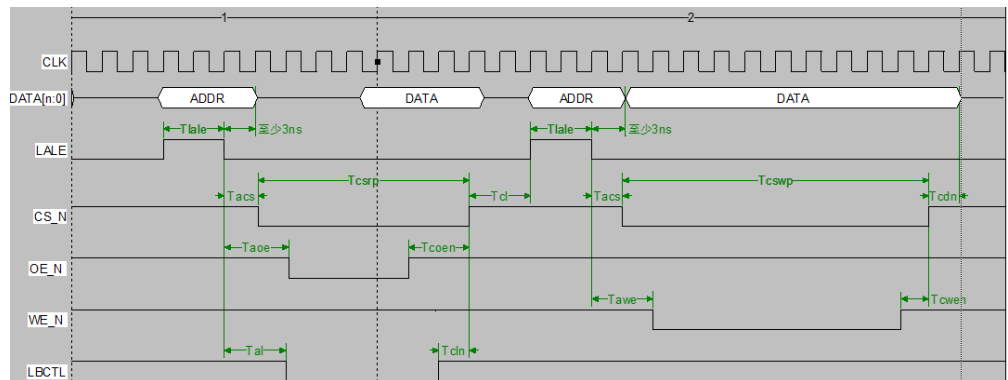


图 11-23 Local Bus 复用无外部 Ready 写读切换操作

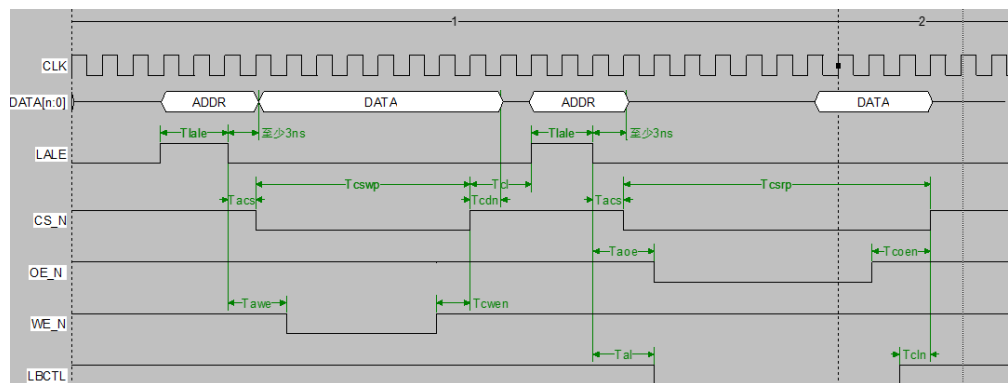
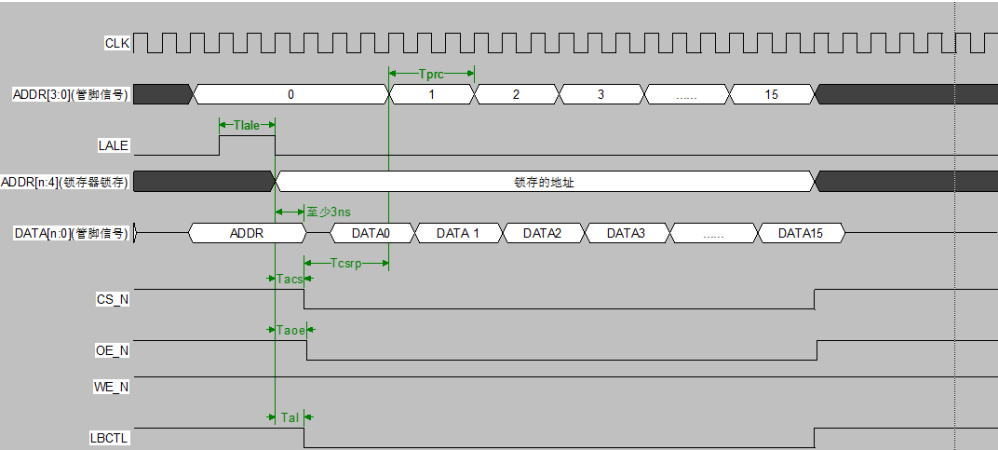




图 11-24 Local Bus 复用无外部 Ready Page 读操作



- 复用、有外部Ready情况

图 11-25 Local Bus 复用有外部 Ready 单次读操作

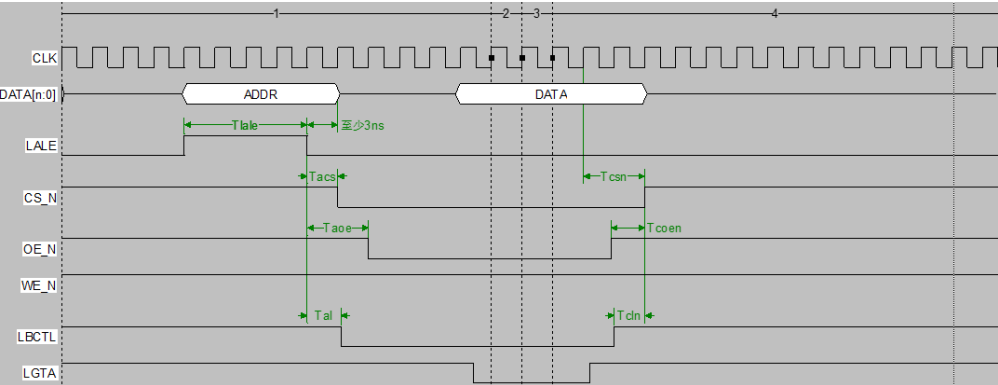


图 11-26 Local Bus 复用有外部 Ready 单次写操作

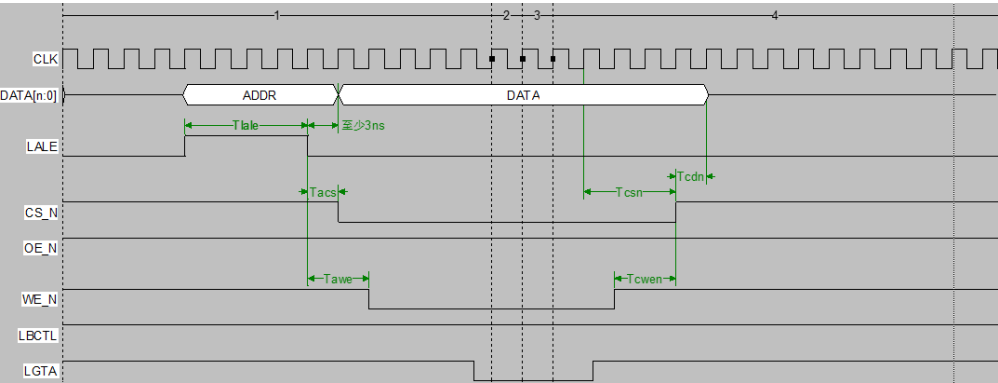


图 11-27 Local Bus 复用有外部 Ready 连续读操作

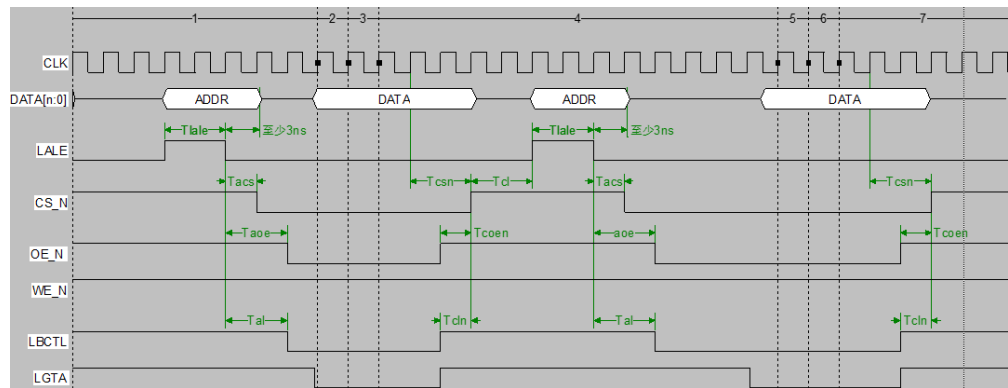


图 11-28 Local Bus 复用有外部 Ready 连续写操作

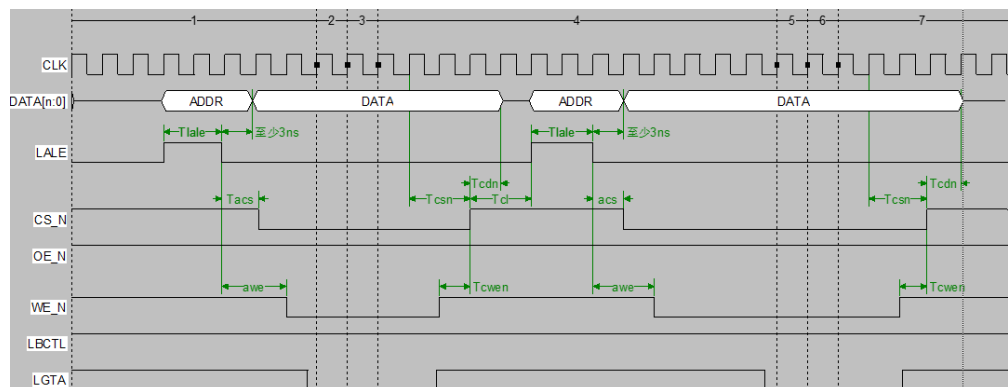


图 11-29 Local Bus 复用有外部 Ready 读写切换操作

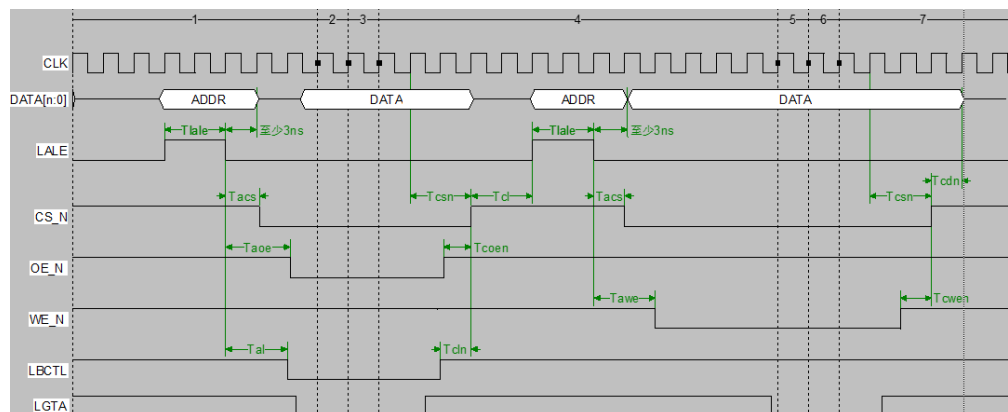




图 11-30 Local Bus 复用有外部 Ready 写读切换操作

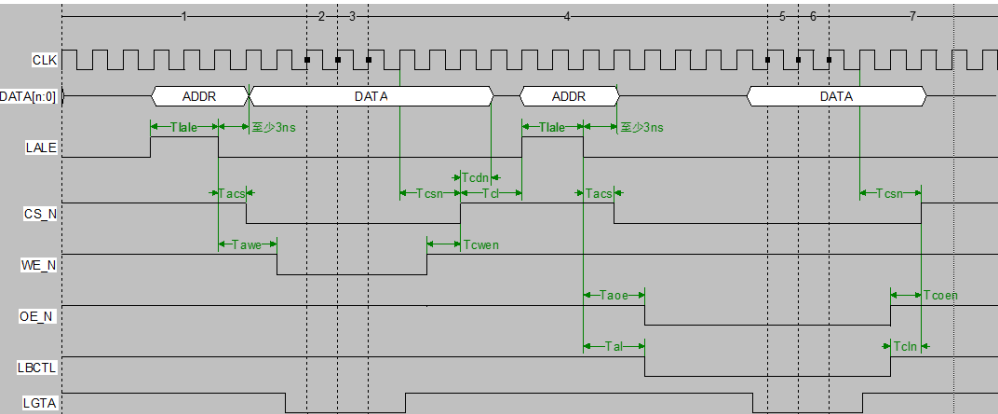
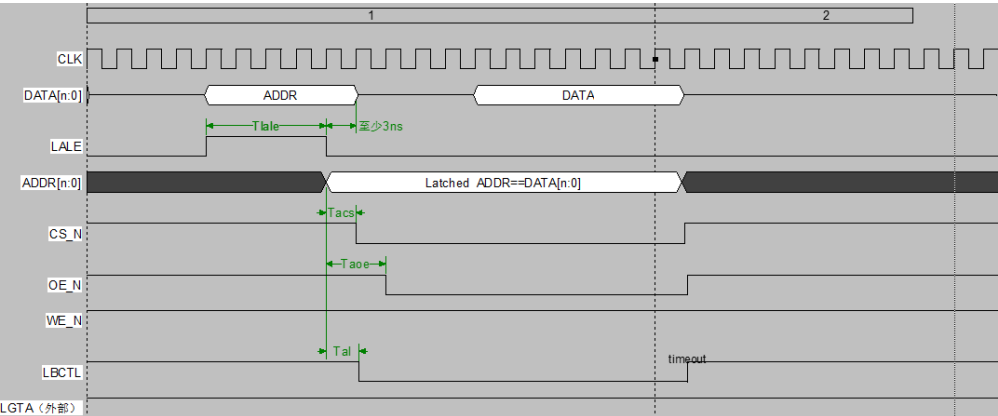


图 11-31 Local Bus 复用有外部 Ready 外部 Ready 超时



11.4 工作流程

11.4.1 时钟门控

11.4.1.1 打开时钟

上电时LOCALBUS工作时钟默认为打开状态。在不使用LOCALBUS的情况下，可以关闭LOCALBUS工作时钟，以降低芯片功耗。

打开门控时钟具体配置步骤如下：

- 步骤1 读peri_sub控制器寄存器SC_LB_CLK_ST，获取当前时钟门控信息。
- 步骤2 如果SC_LB_CLK_ST为低，进入步骤3；SC_LB_CLK_ST为高，表示LOCALBUS时钟已经打开，直接跳过下面的步骤结束操作。



步骤3 配置系统控制寄存器SC_LB_CLK_EN为高，打开LOCALBUS模块的时钟门控。等待10μs后，读取SC_LB_CLK_ST，如果为高，则进行其他初始化操作，否则认为出现异常。

----结束

11.4.1.2 关闭时钟

关闭门控时钟具体配置步骤如下：

步骤1 读控制器寄存器SC_LOCALBUS_CLK_ST，获取当前时钟门控信息。

步骤2 如果SC_LOCALBUS_CLK_ST为高，进入**步骤3**；SC_LOCALBUS_CLK_ST为低，表示LOCALBUS时钟已经关闭，直接跳过下面的步骤结束操作。

步骤3 配置系统控制寄存器SC_LOCALBUS_CLK_DIS为高，打开LOCALBUS模块的时钟门控。

----结束

11.4.2 软复位

对接器件容许的情况下，可以任意时间对LOCALBUS工作逻辑进行软复位，不会对系统造成影响。

- 通过配置控制器中SC_LBUS_RESET_REQ为0b1，可实现对LOCALBUS工作逻辑的单独软复位。
- 通过配置控制器中SC_LBUS_RESET_DREQ为0b1，可实现对LOCALBUS工作逻辑的单独软复位撤离。（上电初始化期间，若配置软复位撤离，延时10μs后读取软复位状态寄存器SC_LBUS_RESET_ST，如果为0，则执行下一步，否则认为出现异常）。

11.4.3 初始化

需要根据LOCALBUS控制器外接存储器件的时序参数和特性，对LOCALBUS控制器进行正确配置，具体配置步骤如下（以配置片选0为例）：

步骤1 根据外部器件配置片选0控制寄存器LB_CS0_CTRL（偏移地址为0x0020，bit[18:0]）。如配置是否支持支持外部Ready器件，是否支持Page读，Page读大小，数据位宽等。

步骤2 根据外部器件的时序特性，配置片选0的时序参数寄存器，LB_CS0_T_SET0（偏移地址为0x0040），LB_CS0_T_SET1偏移地址为0x0044），LB_CS0_T_SET2偏移地址为0x0048）。

步骤3 如果片选0访问的器件，存在写保护区域，则配置LB_CTRL(偏移地址0x00cc，bit[15:0])使能写保护区域，例如使能保护地址段0，配置写保护地址段0的起始地址LB_PROT_ADDR_START_0（偏移地址为0x00d0），配置写保护地址段0的结束地址LB_PROT_ADDR_OVER_0（偏移地址为0x00d4），详细的写保护功能在**异步等待模式**中介绍。

步骤4 如果在1中配置是带有外部Ready的器件，并且使能了超时保护功能，需要配置片选0的超时保护时间LB_RDY_TOUT_0（偏移地址为00AC）。

----结束

11.4.4 基地址配置

8个片选的基地址可以配置，具体步骤如下（以配置CS0为例）。



- 步骤1** 通过寄存器LB_CSx_BASE_ADDR（偏移地址为0x0000）配置CSx所挂接memory的基地址。
- 步骤2** 通过寄存器LB_CSx_CTRL[cf_mem_size_x]（偏移地址为0x0020）配置所挂接memory的存储容量。
- 步骤3** 读取调试寄存器DBG_ST0[0]（偏移地址为0x0140），如果等于1，说明有地址交叠，需要返回到**步骤1**，或者**步骤2**重新配置memory的地址分配。

----结束

须知

配置基地址要静态配置，最好在操作空闲期配置。

11.4.5 固定时序模式

需要根据LocalBus控制器外接存储器件的时序参数和特性，对LocalBus控制器进行正确配置。

具体配置步骤如下（以配置CS0为例）：

- 步骤1** 配置LB_CFG_T_UPDATE_EN[cf_t_update_en]（偏移地址为0x0154，bit[0]）为0。
- 步骤2** 根据外部器件的类型，配置LB_CS0_CTRL（偏移地址为0x0020）。
- 步骤3** 根据外部器件的时序要求，配置LB_CS0_T_SET0、LB_CS0_T_SET1、LB_CS0_T_SET2（偏移地址为0x0044、0x0048、0x004C）。
- 步骤4** 读取时序调试寄存器DBG_T_CFG（偏移地址为0x0148），如果等于1，说明时序配置存在错误，需要返回到**步骤3**，重新配置时序参数，直到DBG_T_CFG的值为0。
- 步骤5** 配置LB_CFG_T_UPDATE_EN[cf_t_update_en]（偏移地址为0x0154，bit[0]）为1。

----结束

11.4.6 异步等待模式

当Local Bus接带有外部Ready的器件时，需要进行如下配置（以片选0为例）：

- 步骤1** 配置LB_CS0_CTRL[cf_rdy_mode_0]（偏移地址为0x0020，bit[9]）为1，使能外部Ready。
- 步骤2** 配置LB_CS0_CTRL[cf_rdy_pol_0]（偏移地址为0x0020，bit[10]），配置外部Ready的有效极性，为0表示低有效，为1表示高有效。
- 步骤3** 配置LB_CS0_CTRL[cf_rdy_tout_en_0]（偏移地址为0x0020，bit[16]），配置为0关闭超时保护功能，配置为1使能超时保护功能。
- 步骤4** 如果使能超时保护功能，需要配置超时保护时间，LB_RDY_TOUT_0（偏移地址为00AC）。
- 步骤5** 配置LB_CS0_T_SET2[cf_t_lgta_samp_0]（偏移地址为0x0048，bit[28:23]），从片选有效到开始采样外部Ready的时间，如果此参数小于等于1，则从片选有效开始采样。



步骤6 配置LB_CS0_T_SET2[cf_t_csn_0](偏移地址为0x0048, bit[5:0]), 外部Ready有效3拍到片选无效的时间（异步模式特有时序参数）。

----结束

11.4.7 Page 读模式

Page读模式需要如下配置（以片选0为例）：

步骤1 配置LB_CS0_CTRL[cf_page_en_0]（偏移地址为0x0020, bit[14]），配置为1，使能Page读模式。

步骤2 配置LB_CS0_CTRL[cf_Page_size_0]（偏移地址为0x0020, bit[16:15]），配置Page size的大小。

步骤3 配置LB_CS0_T_SET0[cf_t_csrp_0](偏移地址为0x0040, bit[26:18]), 配置从地址有效到第一次读数据的时间。

步骤4 配置LB_CS0_T_SET2[cf_t_prc_0] (偏移地址为0x0048, bit[10:6]), 配置两次读之间的时间。

----结束

11.4.8 写保护功能

11.4.8.1 普通写保护功能

Local Bus的写保护功能需要如下配置（以写保护地址段0为例）：

步骤1 配置LB_CTRL [cf_wr_prot_0](偏移地址为0x00cc, bit[15:14]), 配置为01表示低级写保护使能。

步骤2 配置写保护地址段0的起始地址LB_PROT_ADDR_START_0（偏移地址为0x00d0），配置写保护地址段0的结束地址LB_PROT_ADDR_OVER_0（偏移地址为0x00d4）。

步骤3 如果在保护地址段内存在命令字地址，且需要使命令字写操作可以执行。则需要配置命令字地址LB_NOR_CMDi_ADDR(i=0-3)（偏移地址为0x0110、0x118、0x120、0x128）和相应的掩码LB_NOR_CMDi_MASK(i=0-3)（偏移地址为0x0114、0x11c、0x124、0x12c），命令字地址和相应掩码共有四对，可根据写保护地址段0的命令字个数选择。

----结束

11.4.8.2 高级写保护功能

步骤1 配置LB_CTRL [cf_wr_prot_0](偏移地址为0x00cc, bit[15:14]), 配置为10表示高级写保护使能。

步骤2 配置高级写保护使能关键字LB_HL_PROT_KEY0 [cf_hl_prot_key0] (偏移地址为0x0130, bit[31:16])。

步骤3 配置写保护地址段0的起始地址LB_PROT_ADDR_START_0（偏移地址为0x00d0），配置写保护地址段0的结束地址LB_PROT_ADDR_OVER_0（偏移地址为0x00d4）。

步骤4 如果在保护地址段内存在命令字地址，且需要使命令字写操作可以执行。则需要配置命令字地址LB_NOR_CMDi_ADDR(i=0-3)（偏移地址为0x0110、0x118、0x120、0x128）和相应的掩码LB_NOR_CMDi_MASK(i=0-3)（偏移地址为0x0114、0x11c、



0x124、0x12c），命令字地址和相应掩码共有四对，可根据写保护地址段0的命令字个数选择。

----结束

11.5 DFX

11.5.1 写保护错误查询

在寄存器手册中可以查询到写保护错误状态，偏移地址为0x00A4，其中bit[7:0]分别表示写保护地址段0到写保护7的错误状态。

11.5.2 超时错误查询

在寄存器手册中可以查询到超时错误状态，偏移地址为0x00A4，其中bit[15:8]分别表示片选0超时到写片选7超时错误状态。

11.5.3 注意事项

11.5.3.1 一致性要求

软件配置流程：

步骤1 配置 LB_CFG_T_UPDATE_EN寄存器的cfg_t_update_en= 0。

步骤2 分别配置local bus的4个片选基地址。

步骤3 配置 LB_CFG_T_UPDATE_EN寄存器的cfg_t_update_en= 1。

----结束



警告

一致性要求场景要增加一次读操作。

11.5.3.2 CPU 数据位宽与 localbus 接口数据位宽

CPU数据数据位宽是localbus接口数据位宽的4倍和2倍时，数据的对应关系如下：

1. CPU数据位宽32bit，localbus接口数据位宽8bit时。
 - a. 写操作，先发cpu_data[7:0]，然后发cpu_data[15:8]；然后发cpu_data[23:16]；最后发cpu_data[31:24]。
 - b. 读操作，先读lb_data[7:0]，然后读lb_data[15:8]；然后读lb_data[23:16]；最后读lb_data[31:24]。在localbus内部拼接成lb[31:0]给CPU。
2. CPU数据位宽32bit，localbus接口数据位宽16bit时：
 - a. 写操作，先发cpu_data[15:0]，再发cpu_data[31:16]。



- b. 读操作，先读lb_data[15:0]，再读lb_data[31:16]。在localbus内部拼接成lb[31:0]给CPU。



12 IFC 控制器

12.1 关键术语

表 12-1 关键术语解释

术语	术语解释
IFC	Integrated Flash Controller，集成Flash控制器。

12.2 SPI FLASH 控制器

12.2.1 概述

SFC模块是一个SPI FLASH控制器，主要功能完成系统对SPI FLASH的访问控制。

SFC的主要特点如下：

- 支持两个片选CS0和CS1。每个片选的存储空间最大支持到128Mbit(3Byte地址模式)/8Gbit(4Byte地址模式)，两片选可以支持不同容量的FLASH。每片选可分别映射到系统地址空间，映射基地址可配置。
- 支持地址Alias（CS0可以有第二个映射系统地址空间）。
- 支持地址remap，CS0器件接口的地址范围通过寄存器sfc_if_remap_addr配置。
- 支持Standard SPI、Dual-Output/Dual-Input SPI、Quad-Output/Quad-Input SPI、Dual-I/O SPI、Quad-I/O SPI、Full DIO SPI、Full QIO SPI等多种接口类型。上电后默认支持Standard SPI接口类型，可通过寄存器配置切换接口类型。
- 支持3Byte和4Byte两种FLASH地址模式。上电后默认支持3Bytes地址模式，可通过寄存器配置切换地址模式。
- SPI FLASH读写操作支持总线直接读写和寄存器编程读写两种方式。
- 支持写保护操作，写保护操作支持韧性，在韧性锁锁定后，FLASH中写保护区域不能被改写。



须知

SFC模块共设计了两个片选，芯片使用片选1，片选0不可使用。

12.2.2 依赖关系

SFC控制器包含有1个AXI Slave口，用于FLASH访问，一个为APB接口，用于寄存器配置。

12.2.3 操作步骤

SFC 初始化步骤

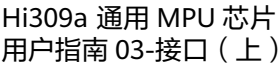
（可选）如果需要调整Timing参数，则配置SFC的Timing寄存器为对应值（当前提供的配置值为寄存器默认值）。

-	Offset Address 0x3048											Register Name SFC_TIME_CFG											Total Reset Value 0x0040_0F66										
Bit	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0	
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0											
Name	reserved			sfc_if_rpd											reserved			sfc_if_bta			sfc_if_csh			sfc_if_css									
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	1	1	0	
-	Bits		Access	Name			Description																										
-	[31:28]		RO	reserved			保留																										
-	[27:16]		RW	sfc_if_rpd			复位之后首次访问FLASH的延时，单位为时钟周期																										
-	[15:12]		RO	reserved			保留																										
-	[11:8]		RW	sfc_if_bta			数据访问间隔时间(t_bta) 实际设定时间为(lbc_if_bta+1)T 时间范围：1~16T(至少1T)																										



-	Offset Address 0x3048												Register Name SFC_TIME_CFG												Total Reset Value 0x0040_0F66											
Bit	3	3	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0		
Name	reserved				sfc_if_rpd												reserved				sfc_if_bt_a				sfc_if_csh				sfc_if_css							
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	1	1	0			
-	Bits		Access		Name				Description																											
-	[7:4]		RW		sfc_if_csh				数据传输完成后CS有效保持时间(t_csh) 当设定为0时无CS有效保持时间，非0时，为实际设定时间 时间范围：0~15T																											
-	[3:0]		RW		sfc_if_css				数据传输开始前CS有效建立时间(t_css) 当设定为0时无CS有效建立时间，非0时，为实际设定时间 时间范围：0~15T																											

（可选）根据实际FLASH大小配置BUS_FLASH_SIZE（可以直接获知器件大小或可通过发Read ID命令给FLASH查询获得）



-	Offset Address 0x3038												Register Name SFC_MEM0_CFG												Total Reset Value 0x000B_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	reserved												sfc_if0_mem_size				sfc_if0_base_addr																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
-	Bits		Access		Name				Description																															
-	[31:20]		RO		reserved				保留。																															
-	[19:16]		RW		sfc_if0_mem_size				接口外接器件容量。 0x00: No Device; 0x01: 64KB; 0x02: 128KB; 0x03: 256KB; 0x04: 512KB; 0x05: 1MB; 0x06: 2MB; 0x07: 4MB; 0x08: 8MB; 0x09: 16MB; 0x0A: 32MB; 0x0B: 64MB; 0x0C: 128MB; 0x0D: 256MB; 0x0E: 512MB; 0x0F: 1GB。																															
-	[15:0]		RW		sfc_if0_base_addr				外部器件起始地址高16位。																															
-	Offset Address 0x303C												Register Name SFC_MEM1_CFG												Total Reset Value 0x000B_0400															

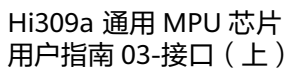


-	Offset Address 0x3038												Register Name SFC_MEM0_CFG												Total Reset Value 0x000B_0000											
Bit	3	3	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0			
Name	reserved												sfc_if0_mem_size				sfc_if0_base_addr																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
-	Bits		Access		Name								Description																							
Bit	3	3	2	2	2	2	2	2	2	2	2	21	2	1	1	1	1	1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved												sfc_if1_mem_size				sfc_if1_base_addr																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0			
-	Bits		Access		Name								Description																							
-	[31:20]		RO		reserved								保留。																							



- Offset Address 0x3038														Register Name SFC_MEM0_CFG										Total Reset Value 0x000B_0000												
Bit	3	3	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0		
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0														
Name	reserved														sfc_if0_mem_size			sfc_if0_base_addr																		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
-	Bits		Access		Name				Description																											
-	[19:16]		RW		sfc_if1_mem_size				接口外接器件容量。 0x00: No Device; 0x01: 64KB; 0x02: 128KB; 0x03: 256KB; 0x04: 512KB; 0x05: 1MB; 0x06: 2MB; 0x07: 4MB; 0x08: 8MB; 0x09: 16MB; 0x0A: 32MB; 0x0B: 64MB; 0x0C: 128MB; 0x0D: 256MB; 0x0E: 512MB; 0x0F: 1GB。																											
-	[15:0]		RW		sfc_if1_base_addr				外部器件起始地址高16位。																											

（可选）配置SFC_IF_CFG以对控制器与板级延迟的配合，SPI fLASH部分模式功能支持、时钟选择等整体功能进行配置，包括时钟分频、模式选择、大小端、采样片选，HOLD/RST选择等。



- Offset Address 0x304C												Register Name SFC_IF_CFG										Total Reset Value 0x0000_0001										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	Reserved			sfc_miso_sample					Reserved	sfc_mosi_sft			Reserved	sfc_clk_sft			Reserved			sfc_crc_t_wpn_en			sfc_crc_rst_mosi	sfc_crc_qst_mosi	sfc_clk_div			sfc_crc_hl_srst			Reserved	sfc_crc_fifo
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-	Bits		Access		Name				Description																							
-	[31:29]		RO		reserved				保留。																							

-	Offset Address 0x304C											Register Name SFC_IF_CFG											Total Reset Value 0x0000_0001									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved			sfc_miso_sample					reserved		sfc_mosi_sft		reserved		sfc_clk_sft		reserved				sfc_crst_wen		sfc_fcst_qst2m0side		sfc_clk_div			sfc_hlslbl_firnst		reserved		sfc_rifcdcpol
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-	Bits		Access	Name		Description																										
-	[28:24]		RW	sfc_miso_sample		SFC_MISO采样时间(以sys_clk为单位)。 sfc_miso_sample[28]设置在SFC_CLK电平采样时间。 0: 低电平; 1: 高电平。 sfc_miso_sample[27:24]设置采样时间点。 0x00: TIME_CNT为0时采样; 0x01: TIME_CNT为1时采样; 0x02: TIME_CNT为2时采样; 0x03: TIME_CNT为3时采样; 0x04: TIME_CNT为4时采样; 0x05: TIME_CNT为5时采样; 0x06: TIME_CNT为6时采样; 0x07: TIME_CNT为7时采样; 0x08: TIME_CNT为8时采样; 0x09: TIME_CNT为9时采样; 0x0A: TIME_CNT为10时采样; 0x0B: TIME_CNT为11时采样;																										

-	Offset Address 0x304C											Register Name SFC_IF_CFG											Total Reset Value 0x0000_0001															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Name	reserved			sfc_miso_sample					reserved	sfc_mosi_sft			reserved	sfc_clk_sft			reserved			sfc_rst_wen		sfc_fcrc	sfc_fcrc	sfc_clk_div				sfc_hl	sfc_hl	reserved	sfc_reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1						
-	Bits		Access		Name			Description																														
								0x0C: TIME_CNT为12时采样; 0x0D: TIME_CNT为13时采样; 0x0E: TIME_CNT为14时采样; 0x0F: TIME_CNT为15时采样; 其他: 保留。 sfc_miso_sample[27:24]设置值必须小于等于sfc_clk_div。 TIME_CNT为递减操作。																														
-	[23]		-		reserved			保留。																														
-	[22:20]		RW		sfc_mosi_sft			SFC_MOSI偏移量设置(以sys_clk为单位)。 000: 不偏移; 001: 向后偏移1个sysclk; 010: 向后偏移2个sysclk; 011: 向后偏移3个sysclk; 100: 向后偏移4个sysclk; 101: 向后偏移5个sysclk; 110: 向后偏移6个sysclk; 111: 向后偏移7个sysclk。																														



- Offset Address 0x304C												Register Name SFC_IF_CFG												Total Reset Value 0x0000_0001											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved			sfc_miso_sample				reserved				sfc_clk_sft				reserved				sfc_clk_div				reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1			
-	Bits		Access		Name				Description																										
-	[19]		-		reserved				保留。																										
-	[18:16]		RW		sfc_clk_sft				SFC_CLK偏移量设置(以sys_clk为单位)。 000: 不偏移; 001: 向后偏移1个sysclk; 010: 向后偏移2个sysclk; 011: 向后偏移3个sysclk; 100: 向后偏移4个sysclk; 101: 向后偏移5个sysclk; 110: 向后偏移6个sysclk; 111: 向后偏移7个sysclk。																										
-	[15:12]		-		reserved				保留。																										



- Offset Address 0x304C												Register Name SFC_IF_CFG												Total Reset Value 0x0000_0001													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	reserved			sfc_miso_sample				reserved				sfc_clk_sft				reserved				sfc_rst_en				sfc_rst2mosi		sfc_clk_div		sfc_clk_div				sfc_clk_div		reserved		sfc_clk_div	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1					
-	Bits		Access		Name				Description																												
-	[11]		RW		sfc_wp_en				Write Protect。 0: No Write Protect; 1: Write Protect。 在Qaud接口模式下，SFC_MOSI[2]输出数据。 在其他模式下，在sfc_wp_en为1时，SFC_MOSI[2]输出sfc_wp_en信号。																												
-	[10]		RW		sfc_rst_en				Reset SPI FLASH Device。 0: Normal; 1: Reset SPI FLASH。 在Qaud接口模式下，SFC_MOSI[3]输出数据。 在其他模式下，在sfc_rst2mosi为1时，输出Reset信号。																												
-	[9]		RW		sfc_rst2mosi				SFC将Reset控制信号输出至sfc_mosi bit3。 0: 输出HOLD信号； 1: 输出RESET信号。																												



- Offset Address 0x304C													Register Name SFC_IF_CFG										Total Reset Value 0x0000_0001																		
Bit	3	3	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	9	8	7	6	5	4	3	2	1	0								
Name	reserved			sfc_miso_sample				reserved				sfc_mosi_sft				reserved				sfc_clk_sft				reserved				sfc_clk_div		sfc_clk_div				sfc_clk_div				reserved		reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1								
-	Bits		Access		Name				Description																																
-	[8]		RW		sfc_quad_mode				SFC输出数据接口组成控制。 0：非Qaud IO模式(DQ3~DQ2输出功能接口)； 数据接口DQ2输出WP信号。 数据接口DQ3输出RESET或者HOLD信号。 数据接口DQ1~DQ0输出数据。 1： Qaud IO模式； 数据接口DQ3~DQ0输出数据接口。																																

（可选）配置SFC_AUTOPRO_CMD/SFC_AUTOPRO_CFG以完成对总线自动方式的配置，即控制器自动完成读写操作，无需软件参与。

-	Offset Address 0x3050												Register Name SFC_AUTOPRO_CMD										Total Reset Value 0x0000_0605										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved																sfc_wr_enbale_cmd						sfc_rdstatus_cmd										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1

-	Bits	Access	Name	Description
-	[31:16]	RO	reserved	保留。
-	[15:8]	RW	sfc_wr_enbale_cmd	写使能命令字。
-	[7:0]	RW	sfc_rdstatus_cmd	读状态命令字。

-	Offset Address 0x3054												Register Name SFC_AUTOPRO_CFG										Total Reset Value 0x0FF8_0018									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



-	Offset Address 0x3050												Register Name SFC_AUTOPRO_CMD												Total Reset Value 0x0000_0605											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved																sfc_wr_enbale_cmd						sfc_rdstatus_cmd													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1				
-	Bits		Access		Name				Description																											
Name	reserved				sfc_rdstatus_num_max				sfc_rdstatus_cycle								sfc_rdstatus_data_cfg				sfc_rdstatus_data_cfg		sfc_rdstatus_data_cfg		sfc_rdstatus_data_cfg		sfc_rdstatus_data_cfg		sfc_rdstatus_data_cfg		sfc_rdstatus_data_cfg		sfc_wip_location			
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
-	Bits		Access		Name				Description																											



- Offset Address 0x3050												Register Name SFC_AUTOPRO_CMD										Total Reset Value 0x0000_0605														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved															sfc_wr_enbale_cmd							sfc_rdstatus_cmd													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1				
-	Bits			Access		Name				Description																										
-	[31:28]			RO		reserved				保留。																										
-	[27:20]			RW		sfc_rdstatus_num_max				总线自动方式中，读状态寄存器字发送最大次数。																										
-	[19:12]			RW		sfc_rdstatus_cycle				总线写操作写数据发送完后的，读状态命令发送间隔。实际cycle数为配置值乘以4。																										
-	[11:10]			RW		sfc_rdstatus_data_cfg				总线模式接收rdstatus数据的操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: 保留。																										
-	[9:8]			RW		sfc_rdstatus_cmd_cfg				总线模式发送rdstatus命令操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: 保留。																										
-	[7]			RW		sfc_rdstatus_data_ddr_en				读状态数据是否选用DDR选择。 0: Disable; 1: Enable。																										
-	[6:5]			RW		sfc_wr_enable_cmd_cfg				总线模式发送write enable命令操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: 保留。																										



- Offset Address 0x3050												Register Name SFC_AUTOPRO_CMD												Total Reset Value 0x0000_0605												
Bit	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0				
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0														
Name	reserved															sfc_wr_enbale_cmd							sfc_rdstatus_cmd													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1				
-	Bits		Access		Name				Description																											
-	[4]		RW		sfc_rdstatus_en				总线方式进行写操作访问时，控制器自动发送读状态命令使能。 0：不使能； 1：使能。																											
-	[3]		RW		sfc_wr_enable_en				总线方式进行写操作访问时，控制器自动发送write enable命令字使能。 0：不使能； 1：使能。																											
-	[2:0]		RW		sfc_wip_locate				WIP(Write In Progress)位在FLASH状态寄存器的位置。 000：WIP位于FLASH状态寄存器的第0bit；(default) 001：WIP位于FLASH状态寄存器的第1bit； 010：WIP位于FLASH状态寄存器的第2bit； 011：WIP位于FLASH状态寄存器的第3bit； 100：WIP位于FLASH状态寄存器的第4bit； 101：WIP位于FLASH状态寄存器的第5bit； 110：WIP位于FLASH状态寄存器的第6bit； 111：WIP位于FLASH状态寄存器的第7bit。																											

（可选）使用的器件，所对应的page_size，以及发送的读写命令字，通过寄存器进行配置，此外对于读操作，是否采用性能更高的自动完成地址连续读命令对应读数据的读取：



-	Offset Address 0x3058												Register Name SFC_BUS_COM_CFG										Total Reset Value 0x0000_0002									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														sfc_bus_data_size	sfc_page_size
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
-	Bits		Access		Name		Description																									
-	[31:2]		RO		reserved		保留。																									
-	[1]		RW		sfc_bus_data_cho		总线方式DATA传输最大数据量选择。 0：一次读写操作数据量为大小由sfc_bus_data_max决定； 1：sfc_bus_data_max配置无效，对于读操作为整个深度空间，对于写操作，则为一个pagesize。																									
-	[0]		RW		sfc_page_size		SPI FALSH器件 pagesize大小。 0：256Byte； 1：512Byte。																									
-	Offset Address 0x3064												Register Name SFC_BUS_CMD_CFG										Total Reset Value 0x0000_0203									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



-	Offset Address 0x3058												Register Name SFC_BUS_COM_CFG												Total Reset Value 0x0000_0002																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Name	reserved																														sfc_bus_dlp_cmd	sfc_bus_mode_cmd	sfc_bus_wr_cmd	sfc_bus_rd_cmd								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0									
-	Bits		Access		Name								Description																													
Name	sfc_bus_dlp_cmd				sfc_bus_mode_cmd								sfc_bus_wr_cmd								sfc_bus_rd_cmd																					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1								
-	Bits		Access		Name								Description																													
-	[31:24]		RW		sfc_bus_dlp_cmd								BUS模式下DLP命令字。																													
-	[23:16]		RW		sfc_bus_mode_cmd								BUS模式下模式命令字。																													
-	[15:8]		RW		sfc_bus_wr_cmd								BUS模式下写命令字。																													



- Offset Address 0x3058												Register Name SFC_BUS_COM_CFG												Total Reset Value 0x0000_0002															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Name	reserved																														sfc_bus_rd_cmd								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0							
-	Bits		Access		Name								Description																										
-	[7:0]		RW		sfc_bus_rd_cmd								BUS模式下读命令字。																										

（可选）如果需要修改读操作接口类型，提高读数据性能，或者配置读操作地址模式（3/4B），这通过该寄存器完成：



Offset Address											Register Name											Total Reset Value															
0x305C											SFC_BUS_RD_FRM_CFG											0x0000_0000															
3	3	2	2	2	2	2	2	2	2	2	21	2	1	1	1	1	1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0					
1	0	9	8	7	6	5	4	3	2			0	9	8	7	6	5	4	3	2	1																
reserved											sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Bits		Access		Name								Description																									
[31:26]		RO		reserved								保留。																									
[25:24]		RW		sfc_bus_data_max								总线模式 DATA 传输最大数据量选择。 00: 32 Byte; 01: 64 Byte; 10: 128 Byte; 11: 256 Byte。																									
[23]		RW		sfc_bus_dlp_en								总线模式数据是否发送DLP操作选择。 0: Disable; 1: Enable。																									
[22]		RW		sfc_bus_rd_data_ddr_en								总线模式数据读操作是否选用DDR选择。 0: Disable; 1: Enable。																									



Offset Address 0x305C											Register Name SFC_BUS_RD_FRM_CFG											Total Reset Value 0x0000_0000														
3	3	2	2	2	2	2	2	2	2	21	2	1	1	1	1	1	1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0				
1	0	9	8	7	6	5	4	3	2		0	9	8	7	6	5	4	3	2	1																
reserved										sfc_bus_rd_data_max	sfc_bus_rd_data_max	sfc_bus_rd_data_max	sfc_bus_dumy_cycle				reserved				sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max							
										sfc_bus_rd_data_max	sfc_bus_rd_data_max	sfc_bus_rd_data_max									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max							
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max				reserved				sfc_bus_rd_data_max			
																									sfc_bus_rd_data_max</											



Offset Address											Register Name											Total Reset Value													
0x305C											SFC_BUS_RD_FRM_CFG											0x0000_0000													
3	3	2	2	2	2	2	2	2	2	21	2	1	1	1	1	1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0				
1	0	9	8	7	6	5	4	3	2		0	9	8	7	6	5	4	3	2	1															
reserved											sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max	sfc_bus_data_max
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Bits		Access		Name							Description																								
[17:12]		RW		sfc_bus_dummy_cycle							总线模式发送DUMMY cycle时，发送Dummy cycle的数量，以SFC_CLK为单位。 如设定为0，不发送DUMMY cycle，如果设定不为0，发送DUMMY cycle的数量为设定值。 配置此寄存器仅在读操作时有效。 由于SFC无法识别当前命令字是否需要发送Dummy Cycle，所以是否发送Dummy Cycle由S/W保证。																								
[11:9]		RO		reserved							保留。																								
[8]		RW		sfc_bus_rd_mode_en							总线模式下，模式命令字是否发送。 0：Disable； 1：Enable。																								
[7]		RW		sfc_bus_addr_width							总线模式地址传输位宽选择。 0：24 bit； 1：32 bit。																								

Offset Address										Register Name										Total Reset Value																	
0x305C										SFC_BUS_RD_FRM_CFG										0x0000_0000																	
3	3	2	2	2	2	2	2	2	2	21	2	1	1	1	1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0							
1	0	9	8	7	6	5	4	3	2		0	9	8	7	6	5	4	3	2	1																	
reserved										sfc_bus_rd_addr_max	sfc_bus_rd_addr_en	sfc_bus_rd_addr_en	reserved	sfc_bus_drammy_cycle										reserved	sfc_bus_rd_addr_cfg	sfc_bus_rd_addr_en	sfc_bus_rd_addr_en	reserved	sfc_bus_rd_addr_cfg								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
Bits		Access		Name							Description																										
[6]		RW		sfc_bus_rd_addr_dds_en							总线模式下，地址读操作是否选用DDR选择。 0: Disable; 1: Enable。																										
[5:4]		RW		sfc_bus_rd_addr_cfg							总线模式ADDR读操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: Not Send Out。 在寄存器sfc_qaud_mode配置为0时，禁止配置为2。																										
[3:2]		-		reserved							保留。																										

Offset Address										Register Name										Total Reset Value										
0x305C										SFC_BUS_RD_FRM_CFG										0x0000_0000										
3	3	2	2	2	2	2	2	2	2	21	2	1	1	1	1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2		0	9	8	7	6	5	4	3	2	1										
reserved					sf		s		s		sfc		re		sfc		reserv		sf		sf		sf		sf		re		sf	
					c_		f		f		bus		se		bus		ed		c		c		c		c		se		c	
					b		b		b		rd		rv		du		ed		f		f		f		f		rv		b	
					us		u		u		ata				my				_		_		_		_		_		_	
					d		d		d		cfg				cycle				_		_		_		_		_		_	
					at		s		s										r		r		r		r		r		r	
					a		s		s										d		d		d		d		d		d	
					m		s		s										d		d		d		d		d		d	
					ax		d		d										r		r		r		r		r		r	
							p		p										d		d		d		d		d		d	
							e		e										a		a		a		a		a		a	
							n		n										d		d		d		d		d		d	
																			r		r		r		r		r		r	
																			e		e		e		e		e		e	
																			n		n		n		n		n		n	
																			o		o		o		o		o		o	
																			d		d		d		d		d		d	
																			e		e		e		e		e		e	
																			t		t		t		t		t		t	
																			h		h		h		h		h		h	
																			r		r		r		r		r		r	
																			e		e		e		e		e		e	
																			n		n		n		n		n		n	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0		0		0		0	
																			0		0		0</							

(可选) 如果需要修改写操作接口类型, 提高写数据性能, 或者配置写操作地址模式 (3/4B), 这通过该寄存器完成:



-	Offset Address 0x3060											Register Name SFC_BUS_WR_FRM_CFG											Total Reset Value 0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Name	reserved											sfc_bus_wr_data_cfg	reserved											sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
-	Bits		Access		Name		Description																																
-	[31:23]		RO		reserved		保留。																																
-	[22]		RW		sfc_bus_wr_data_ddr_en		总线模式数据写操作是否选用DDR选择。 0: Disable; 1: Enable。																																



-	Offset Address 0x3060										Register Name SFC_BUS_WR_FRM_CFG										Total Reset Value 0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	reserved										sfc_bus_wr_data_cfg	reserved										sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	reserved	sfc_bus_wr_data_cfg										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
-	Bits		Access	Name			Description																														
-	[21:20]		RW	sfc_bus_wr_data_cfg			总线模式DATA写操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: Not Send Out。 在寄存器sfc_quad_mode配置为0时，禁止配置为2。																														
-	[19:9]		-	reserved			保留。																														



- Offset Address 0x3060												Register Name SFC_BUS_WR_FRM_CFG												Total Reset Value 0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved											sfc_bus_wr_data_cfg	reserved											sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	reserved	sfc_bus_wr_data_cfg							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
-	Bits		Access		Name		Description																												
-	[8]		RW		sfc_bus_wr_mode_en		总线模式模式命令字写操作是否发送。 0: Disable; 1: Enable。																												
-	[7]		-		reserved		保留。																												
-	[6]		RW		sfc_bus_wr_addr_ddr_en		总线模式地址写操作是否选用DDR选择。 0: Disable; 1: Enable。																												



- Offset Address 0x3060													Register Name SFC_BUS_WR_FRM_CFG										Total Reset Value 0x0000_0000														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	reserved										sfc_bus_wr_data_cfg	reserved										sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	sfc_bus_wr_data_cfg	reserved	sfc_bus_wr_data_cfg											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
-	Bits		Access	Name			Description																														
-	[5:4]		RW	sfc_bus_wr_addr_cfg			总线模式ADDR写操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: Not Send Out。 在寄存器sfc_quad_mode配置为0时，禁止配置为2。																														
-	[3:2]		RO	reserved			保留。																														

-	Offset Address 0x3060											Register Name SFC_BUS_WR_FRM_CFG										Total Reset Value 0x0000_0000										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved											sfc_bus_wr_data_busr_wrd_data_dr_en	reserved										sfc_wr_data_dr_cfg	reserved	sfc_busr_wrd_data_dr_cfg	reserved	sfc_bus_wr_cmd_cfg					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	Bits		Access	Name		Description																										
-	[1:0]		RW	sfc_bus_wr_cmd_cfg		总线模式CMD写操作选择。 00: Extend(1 bit); 01: Dual (2 bit); 10: Quad (4 bit); 11: Not Send Out。 在寄存器sfc_quad_mode配置为0时，禁止配置为2。																										

初始化SFC中断SFC_IMSK寄存器，根据实际应用场景选择，上电默认为屏蔽中断。



-	Offset Address 0x3018												Register Name SFC_IMSK												Total Reset Value 0x0000_00FF															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	reserved																								sfc_spcmd_imsk	sfc_bus_il_imsk	sfc_soft_il_imsk	sfc_rdsta_time_out_imsk	sfc_avldt_imsk	sfc_fifo_imsk										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1								
-	Bits		Access		Name		Description																																	
-	[31:8]		RO		reserved		保留。																																	
-	[7]		RW		sfc_spcmd_imsk		SFC指令寄存器写操作违反指令黑名单写保护，产生中断屏蔽后的中断寄存器。																																	
-	[6]		RW		sfc_bus_il_imsk		SFC总线方式访问违反写保护，产生中断屏蔽后的中断寄存器。																																	
-	[5]		RW		sfc_soft_il_imsk		SFC软件方式访问违反写保护，产生中断屏蔽后的中断寄存器。																																	
-	[4]		RW		sfc_rdsta_time_out_imsk		总线自动方式读状态发送超限错误中断屏蔽寄存器。																																	

-	Offset Address 0x3018												Register Name SFC_IMSK												Total Reset Value 0x0000_00FF																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																					
Name	Reserved																								sfc_dlp_err_imsk	sfc_bad_if_cfg_imsk	sfc_ivld_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk	sfc_softdone_imsk
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1																				
-	Bits		Access	Name		Description																																															
-	[3]		RW	sfc_dlp_err_imsk		DLP检测错误中断屏蔽寄存器。																																															
-	[2]		RW	sfc_bad_if_cfg_imsk		接口错误配置中断屏蔽寄存器。																																															
-	[1]		RW	sfc_ivld_imsk		数据无效访问中断屏蔽寄存器。																																															
-	[0]		RW	sfc_softdone_imsk		SFC中寄存器方式数据传输完成中断屏蔽后的中断寄存器。																																															

(可选) 如果需要使用软件写保护, 则依据寄存器访问要求, 配置写保护首地址、大小、写保护命令片选, 以及16组屏蔽指令寄存器。

Offset Address 0x30D4												Register Name SFC_PROT_EN										Total Reset Value 0x0001_0001										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sfc_bus_wp_en												sfc_soft_wp_en																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:16]	RW		sfc_bus_wp_en		总线方式写保护使能，该使能为总线方式，用于对特定空间进行写保护。 0：不使能； 1：使能。 sfc_bus_wp_en[i]对应总线方式，sfc_prot_addr(i)的保护地址段使能。 sfc_prot_addr_lock[i]配置为1进行写保护韧性锁定时，IFC会将sfc_bus_wp_en[i]改写为1，且不可通过寄存器配置改写。																											
[15:0]	RW		sfc_soft_wp_en		软件方式写保护使能，该使能为软件方式，用于对特定空间进行写保护。 0：不使能； 1：使能。 sfc_soft_wp_en[i]对应总线方式，sfc_prot_addr(i)的保护地址段使能。 sfc_prot_addr_lock[i]配置为1进行写保护韧性锁定时，IFC会将sfc_soft_wp_en[i]改写为1，且不可通过寄存器配置改写。																											

- Offset Address 0x30C0												Register Name SFC_PROT_EXCMD0										Total Reset Value 0x0000_0000										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



- Offset Address 0x30C0												Register Name SFC_PROT_EXCMD0												Total Reset Value 0x0000_0000																						
Name	sfc_prot_excmd3								sfc_prot_excmd2								sfc_prot_excmd1								sfc_prot_excmd0																					
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
-	Bits		Access		Name								Description																																	
-	[31:24]		RW		sfc_prot_excmd3								写保护扩展命令字3。																																	
-	[23:16]		RW		sfc_prot_excmd2								写保护扩展命令字2。																																	
-	[15:8]		RW		sfc_prot_excmd1								写保护扩展命令字1。																																	
-	[7:0]		RW		sfc_prot_excmd0								写保护扩展命令字0。																																	

- Offset Address 0x30C4												Register Name SFC_PROT_EXCMD1												Total Reset Value 0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Name	sfc_prot_excmd7											sfc_prot_excmd6											sfc_prot_excmd5											sfc_prot_excmd4										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
-	Bits		Access		Name								Description																															
-	[31:24]		RW		sfc_prot_excmd7								写保护扩展命令字7。																															



-	Offset Address 0x30C4			Register Name SFC_PROT_EXCMD1	Total Reset Value 0x0000_0000
-	[23:16]	RW	sfc_prot_excmd6	写保护扩展命令字6。	
-	[15:8]	RW	sfc_prot_excmd5	写保护扩展命令字5。	
-	[7:0]	RW	sfc_prot_excmd4	写保护扩展命令字4。	

- Offset Address 0x30C8												Register Name SFC_PROT_EXCMD2										Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	sfc_prot_excmd11											sfc_prot_excmd10										sfc_prot_excmd9								sfc_prot_excmd8							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
-	Bits			Access			Name					Description																									
-	[31:24]			RW			sfc_prot_excmd11					写保护扩展命令字11。																									
-	[23:16]			RW			sfc_prot_excmd10					写保护扩展命令字10。																									
-	[15:8]			RW			sfc_prot_excmd9					写保护扩展命令字9。																									
-	[7:0]			RW			sfc_prot_excmd8					写保护扩展命令字8。																									



- Offset Address 0x30CC												Register Name SFC_PROT_EXCMD3												Total Reset Value 0x0000_0000																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
Name	sfc_prot_excmd15											sfc_prot_excmd14											sfc_prot_excmd13											sfc_prot_excmd12																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

配置写保护地址范围寄存器，如表12-2、表12-3和表12-4所示。

表 12-2 SFC_PROT_ADDR0~3 寄存器

Bits	寄存器说明	配置值
SFC_PROT_ADDR0~3：偏移地址0x30B0~0x30BC		
[31:16]	数据写保护结束地址0~3。 配置写保护结束地址高16bit。 低16bit固定为0xFFFF。 受IFC_LOCK写保护。 sfc_prot_addr_lock[i]为1时，该域不可被改写，i的取值范围是0~3。	按需配置需写保护的地址范围，默认为全0。



Bits	寄存器说明	配置值
[15:0]	数据写保护起始地址0~3。 配置写保护起始地址高16bit。 低16bit固定为0x0000。 受IFC_LOCK写保护。 sfc_prot_addr_lock[i]为1时，该域不可被改写，i的取值范围是0~3。	按需配置需写保护的地址范围，默认为全0。

表 12-3 SFC_PROT_ADDR4~15 寄存器

Bits	寄存器说明	配置值
SFC_PROT_ADDR4~15：偏移地址0x3200~0x322C		
[31:16]	数据写保护结束地址4~15。 配置写保护结束地址高16bit。 低16bit固定为0xFFFF。 受IFC_LOCK写保护。 sfc_prot_addr_lock[i]为1时，该域不可被改写，i的取值范围是4~15。	按需配置需写保护的地址范围，默认为全0。
[15:0]	数据写保护起始地址4~15。 配置写保护起始地址高16bit。 低16bit固定为0x0000。 受IFC_LOCK写保护。 sfc_prot_addr_lock[i]为1时，该域不可被改写，i的取值范围是4~15。	按需配置需写保护的地址范围，默认为全0。

表 12-4 SFC_PROT_ADDR4~15 寄存器

Bits	寄存器说明	配置值
SFC_PROT_ADDR4~15：偏移地址0x3200~0x320C		
[31:16]	数据写保护结束地址4~15。 配置写保护结束地址高16bit。 低16bit固定为0xFFFF。 受IFC_LOCK写保护。 sfc_prot_addr_lock[i]为1时，该域不可被改写，i的取值范围是4~15。	按需配置需写保护的地址范围，默认为全0。



Bits	寄存器说明	配置值
[15:0]	数据写保护起始地址4~15。 配置写保护起始地址高16bit。 低16bit固定为0x0000。 受IFC_LOCK写保护。 sfc_prot_addr_lock[i]为1时，该域不可被改写，i的取值范围是4~15。	按需配置需写保护的地址范围，默认为全0。

配置指令黑名单寄存器；

表 12-5 SFC_PROT_SPCMD0~1 寄存器

Bits	寄存器说明	配置值
SFC_PROT_SPCMD0~1： 偏移地址0x30E0~0x30E4		
[31:24]	写保护黑名单指令寄存器3+i*4，i取值范围是0~1。 在sfc_prot_spcmd_lock[3+i*4]为1时，该寄存器不可被改写。 在sfc_prot_spcmd_lock[3+i*4]为1时，写入指令寄存器的值与sfc_prot_spcmd[3+i*4]相等，则写入指令寄存器操作失效，并上报异常中断。 指令寄存器包括： sfc_bus_rd_cmd（总线模式读指令）、 sfc_bus_wr_cmd（总线模式写指令）、 sfc_soft_cmd（软件模式指令）、 sfc_wr_enable_cmd（写使能指令）、 sfc_rdstatus_cmd（读状态指令）。	按需配置需黑名单保护的寄存器配置值，默认为全0。
[23:16]	写保护黑名单指令寄存器2+i*4，i取值范围是0~1。 在sfc_prot_spcmd_lock[2+i*4]为1时，该寄存器不可被改写。 在sfc_prot_spcmd_lock[2+i*4]为1时，写入指令寄存器的值与sfc_prot_spcmd[2+i*4]相等，则写入指令寄存器操作失效，并上报异常中断。 指令寄存器包括： sfc_bus_rd_cmd（总线模式读指令）、 sfc_bus_wr_cmd（总线模式写指令）、 sfc_soft_cmd（软件模式指令）、 sfc_wr_enable_cmd（写使能指令）、 sfc_rdstatus_cmd（读状态指令）。	按需配置需黑名单保护的寄存器配置值，默认为全0。



Bits	寄存器说明	配置值
[15:8]	<p>写保护黑名单指令寄存器$1+i*4$，i取值范围是0~1。</p> <p>在sfc_prot_spcmd_lock[1+i*4]为1时，该寄存器不可被改写。</p> <p>在sfc_prot_spcmd_lock[1+i*4]为1时，写入指令寄存器的值与sfc_prot_spcmd[1+i*4]相等，则写入指令寄存器操作失效，并上报异常中断。</p> <p>指令寄存器包括：</p> <p>sfc_bus_rd_cmd（总线模式读指令）、 sfc_bus_wr_cmd（总线模式写指令）、 sfc_soft_cmd（软件模式指令）、 sfc_wr_enable_cmd（写使能指令）、 sfc_rdstatus_cmd（读状态指令）。</p>	按需配置需黑名单保护的寄存器配置值，默认为全0。
[7:0]	<p>写保护黑名单指令寄存器$3+i*4$，i取值范围是0~1。</p> <p>在sfc_prot_spcmd_lock[3+i*4]为1时，该寄存器不可被改写。</p> <p>在sfc_prot_spcmd_lock[3+i*4]为1时，写入指令寄存器的值与sfc_prot_spcmd[3+i*4]相等，则写入指令寄存器操作失效，并上报异常中断。</p> <p>指令寄存器包括：</p> <p>sfc_bus_rd_cmd（总线模式读指令）、 sfc_bus_wr_cmd（总线模式写指令）、 sfc_soft_cmd（软件模式指令）、 sfc_wr_enable_cmd（写使能指令）、 sfc_rdstatus_cmd（读状态指令）。</p>	按需配置需黑名单保护的寄存器配置值，默认为全0。

配置韧性锁，对写保护相关的配置寄存器进行韧性锁定。

表 12-6 SFC_PROT_CMD_LOCK 寄存器

Bits	寄存器说明	配置值
SFC_PROT_CMD_LCOK：偏移地址0x30D8		
[31:24]	保留。	-
[23:8]	<p>写保护扩展命令字寄存器SFC_PROT_EXCMD锁定使能。</p> <p>0：不锁定； 1：锁定。</p> <p>sfc_prot_cmd_lock[15:0]分别对应sfc_prot_excma15~0。</p> <p>sfc_prot_cmd_lock任意bit上电复位值为0，配置为1后，不可被改写。</p>	按需配置需写保护韧性配置锁定值，默认为全0。



Bits	寄存器说明	配置值
[7:0]	写保护黑名单命令字寄存器SFC_PROT_SPCMD锁定使能。 0：不锁定，不使能指令黑名单写保护； 1：锁定，使能指令黑名单写保护。 sfc_prot_spcmd_lock[7:0]分别对应sfc_prot_spcmd7~0。 sfc_prot_spcmd_lock任意bit上电复位值为0，配置为1后，不可被改写。	按需配置需写保护韧性配置锁定值，默认为全0。

表 12-7 SFC_PROT_ADDR_LOCK 寄存器

Bits	寄存器说明	配置值
SFC_PROT_ADDR_LCOK：偏移地址0x30DC		
[31:16]	保留。	-
[15:0]	写保护地址寄存器SFC_PROT_EXCMD锁定使能。 0：不锁定； 1：锁定。 sfc_prot_addr_lock[15:0]分别对应sfc_prot_addr15~0。 sfc_prot_addr_lock任意bit上电复位值为0，配置为1后，不可被改写。	按需配置需写保护韧性配置锁定值，默认为全0。

写保护的逻辑：

图 12-1 总线访问写保护

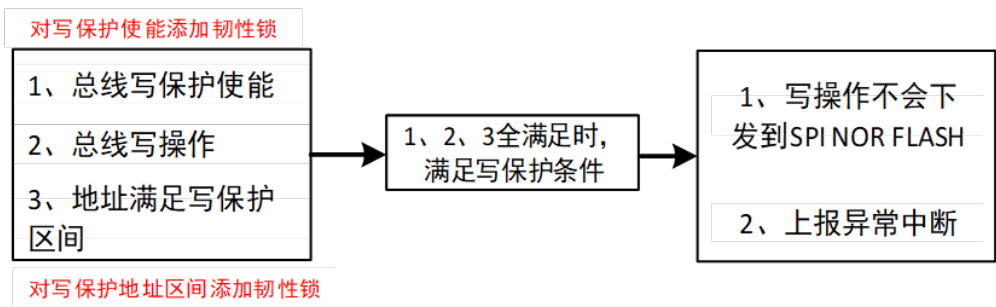


图 12-2 软件访问写保护

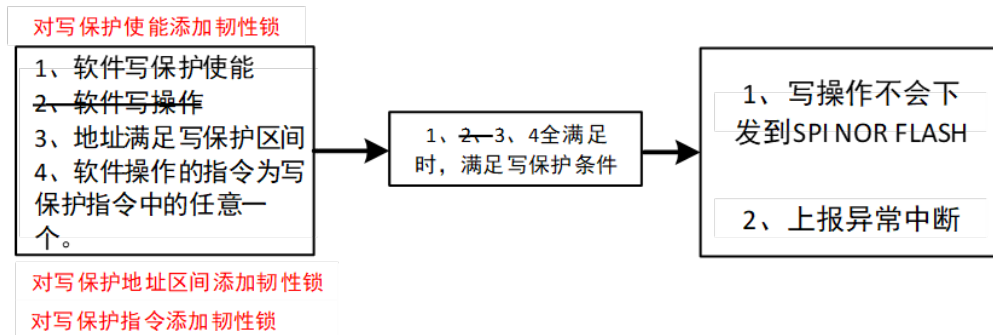
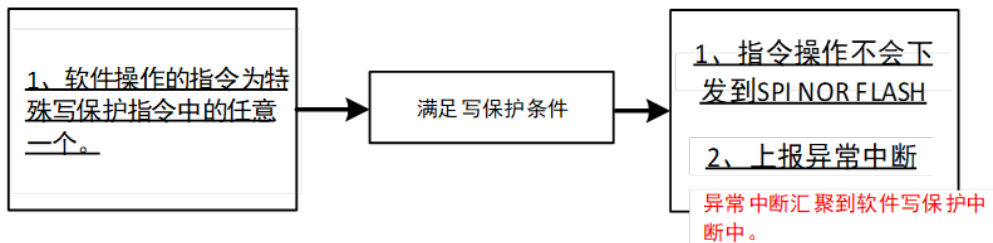


图 12-3 黑名单保护



读 FLASH 操作步骤

提供两种读取FLASH途径的方法：

- 通过AXI总线读取FLASH，此方法读取FLASH操作效率高，SFC模块自动将AXI总线的读操作时序映射为SPI时序。
对于AXI总线的任何读取Memory操作，SFC模块都将其转换成SPI接口的读操作。上电后默认执行Normal read操作（opcode=0x3），这种读操作对应的dummy byte个数为0。上电系统启动后，可以通过软件配置读命令的opcode和dummy byte个数进行其他的读操作，如Fast Read（opcode=0xB），dummy byte为1。对于一个AXI的wrap读操作，SFC拆成两个burst处理。
- 通过设置SFC的配置寄存器读取FLASH，需要软件配置寄存器。

具体步骤如下：

配置软件方式命令字即操作数据量寄存器



- Offset Address 0x3074												Register Name SFC_SOFT_CMD_CFG												Total Reset Value 0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	sfc_soft_dlp_cm								sfc_soft_mode_cm								reserved				sfc_soft_data_num				sfc_soft_cmd												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
-	Bits			Access		Name				Description																											
-	[31:24]			RW		sfc_soft_dlp_cm				软件模式下DLP命令字。																											
-	[23:16]			RW		sfc_soft_mode_cmd				软件模式下模式命令字。																											
-	[15:13]			RO		reserved				保留。																											
-	[12:8]			RW		sfc_soft_data_num				软件模式时数据发送数目，最大数据传输量为32 Byte，最大配置数据为31。实际配置数据为配置值加1。																											
-	[7:0]			RW		sfc_soft_cmd				软件模式下命令字。																											

说明

配置值参考对应器件手册的Command真值表。

12.2.4 应用说明

无

12.2.5 可维可测

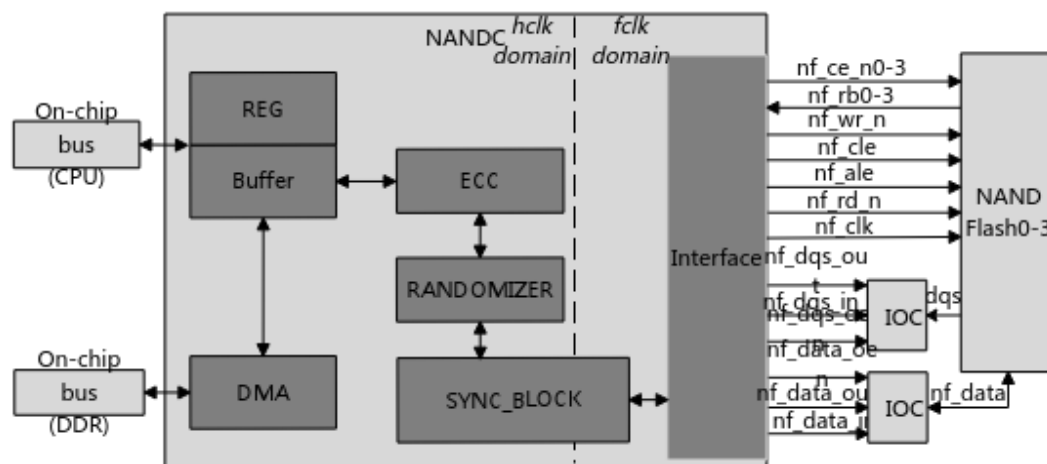
无

13 NAND Controller

13.1 微架构概览

NANDC(NAND FLASH Controller)作为片外NAND FLASH的控制器，主要实现对片外NAND FLASH的数据存取。其连接示意图如图13-1所示。

图 13-1 NANDC 连接示意图



NANDC的外部接口主要包括：On-chip bus接口、NAND FLASH接口（CS和RB信号独立，其他共用）。外部模块（如CPU）可以直接通过On-chip bus接口访问NANDC的内部寄存器和Buffer，NANDC也可以使用DMA功能与外部模块（如DDR）之间进行数据交互。

NANDC内部包含Buffer、ECC、RANDOMIZER、Interface等模块。

- REG：寄存器模块。
- Buffer：内部Buffer用于软件数据缓存。CPU可。
- 以直接通过总线访问。
- ECC：ECC模块用于ECC的编码/解码。



- RANDOMIZER：用于数据的随机化/还原。
- Interface：用于与NAND FLASH接口进行对接。
- DMA：用于外部模块（如DDR）与NAND FLASH之间进行数据交互。
- SYNC_BLOCK：用于处理异步时钟域之间的信号以及数据。

13.2 特性

NANDC主要提供如下特性：

- 支持总线和接口时钟异步。
- 提供一个9KB（8192Byte+1024Byte）片内缓存，提高读取速度。
- 提供内置DMA，用于自动在DDR和NAND FLASH之间交换数据。
- 支持多种规格的NAND FLASH器件：
 - 支持2K、4K、8K、16K Page_size的器件。
 - 支持64、128、256、512 Pages/Block的器件。
 - 支持8bit 数据接口位宽的器件。
 - 支持传统异步接口器件、支持ONFI器件的NV_DDR同步接口、支持ONFI器件的NV_DDR2同步接口（不支持差分）以及TOGGLE接口器件。
- 支持外接4片选，均有独立的RB信号（需要保证外接4片选均使用同一型号的器件）。
- 支持ECC功能：
 - 支持支持8bit/1KB、16bit/1KB、24bit/1KB、40bit/1KB、64bit/1KB的BCH码ECC校验和纠错（1KB指1KB数量级，并不是严格意义上的1024Byte）。
 - 支持ECC校验码生成的使能和关闭、支持ECC自动纠错的使能和关闭。
 - 支持ECC0透传模式下的读写操作。
 - 支持ECC调试功能，只在Buffer中生成ECC校验码或者只在Buffer中进行ECC自动纠错。
 - 支持不使用的ECC控制模块的时钟关闭，降低功耗。
 - 支持读空白Page（数据全为0xFF）时，不会误报ECC不可纠错。
- 支持Randomizer功能。仅在8K Page_size 和16K Page_size配置下开启有效。
- 支持手动配置命令组合，从而完成各种命令。
- 支持NANDC自动下发八种操作：读、写、擦、two-plane读、two-plane写、two-plane擦、cache读、cache写。其中读、写、two-plane读、two-plane写，cache写，cache读为DMA操作。
- 非ECC0模式下支持整个Page的读写、只读OOB、只读数据头，ECC0模式下支持不定长数据读写、整个Page读写。
- 支持Interleave高性能模式的使能和关闭。
- 支持10种中断：操作完成、器件进入busy、器件擦写失败、ECC校验可纠错和不可纠错、总线操作错误、DMA传输错误、DQS超时错误、NANDC内部ECC模块的mem出现1bit错误和超过1bit错误。4个片选分别有一套中断。

13.3 关键术语

表13-1的术语将在本章节表示特殊的含义。



表 13-1 NANDC 模块关键术语描述

术语	描述
RB	Ready/Busy。NAND FLASH的闲/忙指示信号，用于指示器件的操作是否完成。
Page	Page是NAND FLASH读写操作中可寻址的最小单元。每个Page由若干个Byte组成。
Block	NAND FLASH的存储阵列中，可擦除的最小数据单元。一个Block由若干个Page组成。
Plane	NAND FLASH中存储阵列的最小数据单元。每个Plane由若干个Block组成。每个Plane对应一个大小与Page一致的页缓存器，用于存储临时数据。数据写入或读出Plane时，都要经过页缓存器的缓存。
OOB	Out Of Band带外。OOB数据有两种概念：1.对NAND FLASH来说，每个Page中的冗余区域被称为OOB区，此区域用于存放ECC数据、坏块标识以及和文件系统相关的数据。2.对于软件来说，看不到ECC数据，OOB区只包含坏块标识和文件系统数据。本文中的OOB为第二种含义，即只包含坏块标识和文件系统数据。
BB	Bad Block坏块。NAND FLASH的Block出现损坏，无法正常进行擦除操作，且坏块内的Page无法进行正常擦写。系统为区分此类Block，需要对Block进行标记，0xFF表示好块，非0xFF的值表示坏块。
Randomzier	数据随机化。为了提高数据在NAND FLASH中存放的稳定性，减少bit翻转，需要对写入的数据进行随机化处理，以保证0和1均匀分布。
NFC_OP	手动配置操作。为方便描述，使用手动配置操作寄存器的名称NFC_OP来表示手动配置操作。
NFC_OP_CTRL	自动配置操作。为方便描述，使用自动配置操作寄存器的名称NFC_OP_CTRL来表示自动配置操作。
ECCx	表示ECC模式。如ECC8表示8bits ECC模式；ECC0表示ECC功能关闭，即数据透传模式。
ONFI	Open NAND FLASH Interface Specification。
NV_DDR	请参考ONFI3.0标准。
NV_DDR2	请参考ONFI3.0标准。
TOGGLE	三星东芝的接口标准。请参考支持TOGGLE模式的器件手册。
异步模式	请参考ONFI3.0标准中的SDR模式。
同步模式	本文将NV_DDR、NV_DDR2、TOGGLE模式统称为同步模式。
DQS	NAND FLASH的DQS管脚。在同步模式下，读FLASH的操作中，FLASH主动输出DQS信号并伴随数据输出。



13.4 时钟与复位

13.4.1 时钟

NANDC涉及到两个时钟域：总线时钟（hclk）和接口时钟（nf_clk）。其中nf_clk有多档可以配置，由BI_SUBCTRL的寄存器SC_NF_CLK_SEL.nf_clk_sel进行选择，表格中clk_2x频率即为nf_clk的两倍频率，详情请参考BI_SUBCTRL寄存器手册。

时钟关闭和使能开关，有BI_SUBCTRL模块的SC_NF_CLK_EN. icg_en_nf，SC_NF_CLK_DIS. icg_dis_nf。

说明

在NANDC的寄存器手册中涉及到NAND FLASH接口的时间参数的单位为接口时钟（nf_clk），其他时间参数的单位为总线时钟（hclk），使用时需要注意寄存器手册的详细描述。

13.4.2 复位

NANDC的复位包括软复位和硬复位。

- 软复位包含两个时钟域：总线时钟和接口时钟。分别由模块BI_SUBCTRL中的寄存器SC_NF_RESET_REQ[srst_req_nf]和SC_NF_RESET_REQ [srst_req_nf]控制。在对NANDC进行软复位时，必须对总线时钟和接口时钟都进行复位，否则会导致NANDC内部状态错误。解复位由SC_NF_RESET_DREQ[srst_dreq_nf]和SC_NF_RESET_DREQ[srst_dreq_nf]控制。硬复位用于复位LOCK功能相关寄存器，由下电复位信号控制。

NANDC 上电默认处于解复位状态。

具体软复位流程请参考“[NANDC软复位流程](#)”。

13.5 基本功能

本节主要描述NAND FLASH的基本操作以及NANDC的基本功能。

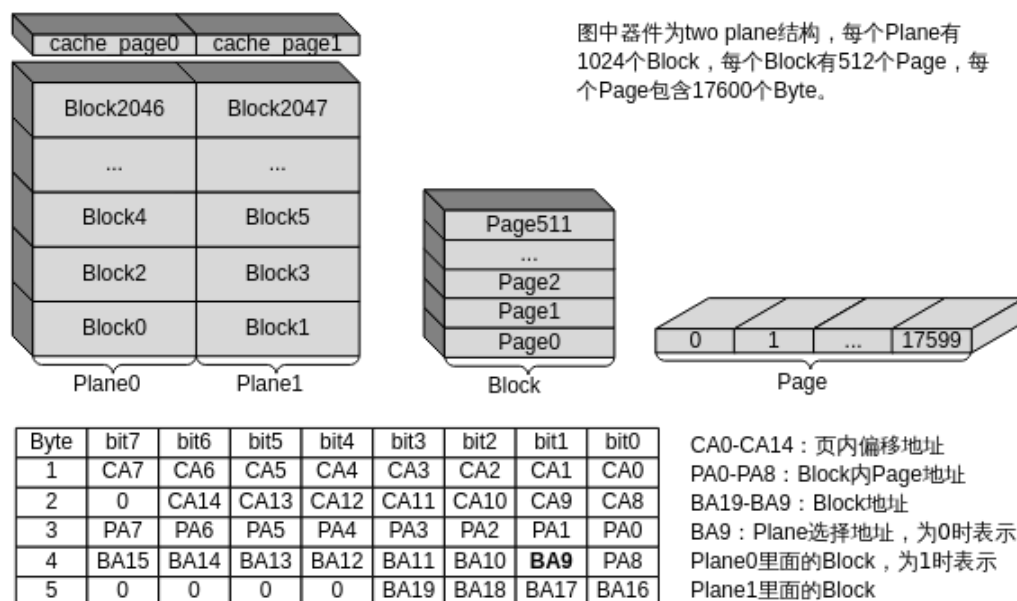
13.5.1 NAND FLASH 存储结构

NAND FLASH的存储结构如[图13-2](#)所示。

一个完整的FLASH地址由5个Byte组成，{Byte2,Byte1}用于表示Page内的偏移地址，{Byte5,Byte4,Byte3}用于表示Page的绝对地址（由Block地址和Block内Page偏移地址组成），其中PA8~PA0表示Block内Page偏移地址。读写操作中，需要发送给器件5Byte的地址，即{Byte5,Byte4,Byte3,Byte2,Byte1}。擦除操作中，不需要页内偏移地址，发送给器件3Byte地址，即{Byte5,Byte4,Byte3}。



图 13-2 NAND FLASH 存储结构



器件的每个Plane有一个单独的缓存页，用于存放临时数据。执行读操作时，器件首先将数据从存储阵列读到页面缓存页，启动输出操作后，才将缓存页的数据输出到器件接口；执行写操作时，器件首先将接口接受的数据存放在缓存页，启动内部编程操作时，才将数据写到存储阵列中。

对于支持two plane的器件，执行two plane读操作时，器件接收到two plane读指令后，两个Plane同时开始从各自的Plane读取数据，并存放各自Plane的缓存页，完成后再依次输出到接口。执行two plane写操作时，器件依次把接口输入的数据存放到对应Plane的缓存页，完成后，两个Plane同时启动编程操作，数据被写入到各自的存储阵列中。从two plane的操作过程可以看出，器件内部执行操作时，是两个Plane同时工作，对比单Plane操作一次只操作一个Plane，可以大大节省busy时间。

13.5.2 NAND FLASH 的基本操作

NANDC通过控制NAND FLASH接口的控制信号线以及数据输入输出来完成对NAND FLASH的操作。下面以异步接口模式下的写读操作为例，来说明NAND FLASH的操作过程。其他类型的操作请参考器件手册。

图 13-3 NAND FLASH 的写操作示意图

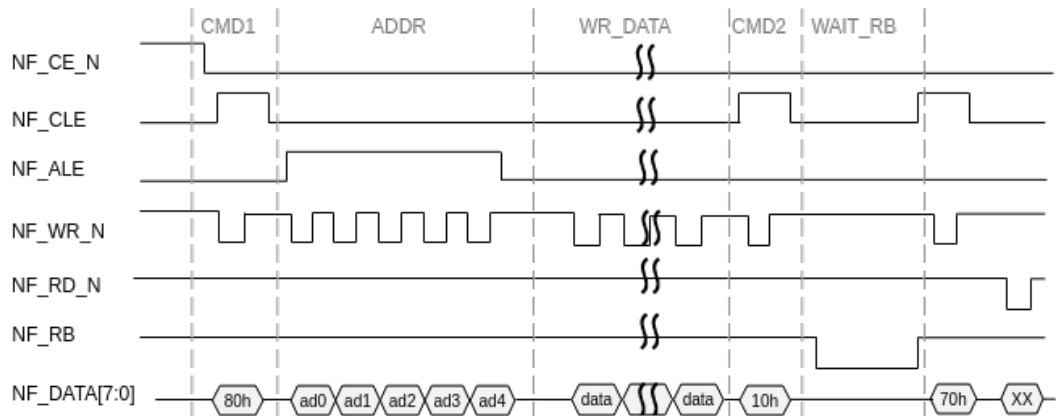


图13-3为NAND FLASH的写操作示意图。NANDC依次将命令1、地址、数据、命令2发送给FLASH，FLASH接收到命令序列之后，内部开始执行编程操作，同时其RB信号拉低。当FLASH内部的写操作执行完成，其RB信号拉高。至此，写操作执行完成。为了保证写操作正确执行，可以再进行读状态寄存器操作来查询FLASH的状态。

图 13-4 NAND FLASH 的读操作示意图

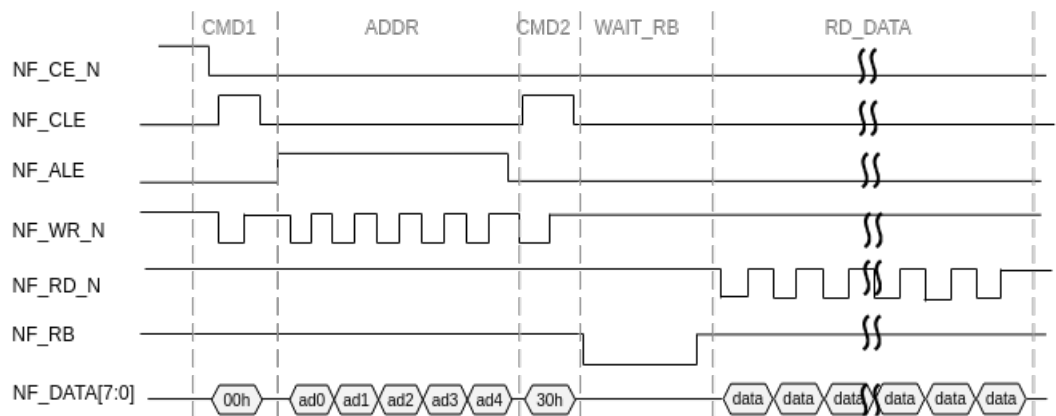
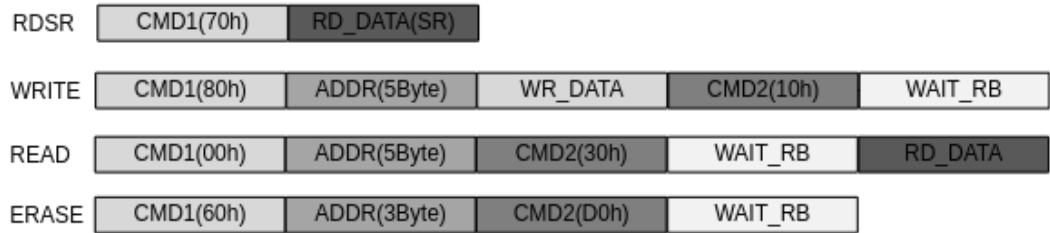


图13-4为NAND FLASH的读操作示意图。NANDC依次将命令1、地址、命令2发送给FLASH，FLASH接收到命令序列之后，内部开始准备数据，同时其RB信号拉低。当FLASH内部数据准备完成之后，其RB信号拉高，此时NANDC开始从接口读取数据，直到数据读取完成。

13.5.3 NANDC 的基本操作

NANDC最基本的功能是能够对NAND FLASH下发各类操作。从上节中NAND FLASH的操作示意图中可以看出，NAND FLASH的操作可以分解为六种基本的操作要素：CMD1（命令1）、ADDR（地址）、WR_DATA（写数据）、CMD2（命令2）、WAIT_RB（等待RB）、RD_DATA（读数据）。如图13-5所示，图中列举了NAND FLASH最常用的4种操作：读器件状态操作（RDSR）、写操作（WRITE）、读操作（READ）、擦除操作（ERASE）。

图 13-5 几种典型操作的分解图



基于这六种操作要素，NANDC可以兼容目前市面所见的绝大部分器件。在实际应用中，NANDC提供两种操作方式来下发NAND FLASH命令：

- 手动配置操作方式（NFC_OP）。此种方式允许软件灵活的配置各类操作要素，由此可以实现各种不同的操作要素组合，这种方式能够实现绝大部分NAND FLASH操作。
- 自动配置操作方式（NFC_OP_CTRL）。此种方式则由NANDC完成特定的某些操作的组合，软件只需要对相关寄存器进行配置，则可以很方便下发比较复杂的操作指令。NANDC只支持某些比较常用的操作的自动下发，用于提高操作性能。

例如对于写Page操作，软件若使用手动配置方式，则需要软件先根据写Page操作的特点，将其分解为CMD1、ADDR、WR_DATA、CMD2、WAIT_RB等要素，然后配置相关寄存器下发这一串操作序列；若使用自动配置方式，软件则不需要关心写Page操作由哪些操作要素组成，NANDC自动配置这一串操作序列。关于这两种操作方式，将在后续章节进行更为详细的描述。

13.5.4 NANDC 操作状态

无论使用手动配置操作方式还是使用自动配置操作方式下发操作，软件都需要知道NANDC的操作何时结束。为此，NANDC提供两种方式来告知软件NANDC的操作是否结束：

- 查询NANDC状态寄存器：NANDC提供状态寄存器来记录当前NANDC的状态，当软件下发操作之后，可以不断的查询NANDC状态寄存器以确认操作结束。
- 检测操作结束中断：NANDC提供专门的操作结束中断供软件使用，当软件下发操作之后，可以等待操作结束中断以确认操作结束。

关于这两种方式的具体使用方法请参考“[典型应用配置](#)”章节。

13.5.5 其他功能

NANDC除了能够对NAND FLASH实现操作之外，还提供其他辅助功能：

- ECC功能：NAND FLASH的存储介质的物理特性导致数据在存储的过程中容易发生bit翻转，因此有必要对Page中的数据进行ECC保护。软件在对FLASH写入数据时，NANDC先对原始数据进行ECC保护，生成ECC码，并将原始数据和ECC码一同写入到FLASH中；软件读取FLASH中的数据时，NANDC先对读取的数据进行ECC纠错，并将还原后的数据返回给软件。
- Randomizer功能：对于部分大容量的MLC器件，Page中的数据的0、1分布影响其中数据的稳定性。Randomizer功能能够对写入Page数据预先进行加扰处理，使数据的0、1分布更均匀，以提高数据存储的稳定性，在读出数据时，又可以将其还原为原始数据。



- TIMEOUT机制：对于WAIT_RB，NANDC提供TIMEOUT机制，防止由于器件问题出现RB不拉高而导致操作挂死。
- Interleave功能：仅在NFC_OP_CTRL模式下使用，用于大数据量的读写，可以提高平均读写性能。

下面的章节将详细的介绍这些特性。

13.6 ECC 功能

ECC功能用于对写入FLASH的数据进行保护，防止FLASH在存储过程中出现bit翻转。写操作时，NANDC首先在内部Buffer中处理软件数据，按照预定的数据结构生成ECC保护码，然后将软件数据和ECC保护码一起写入到NAND FLASH；读操作时，NANDC在Buffer中进行ECC校验，若遇到可以纠正错误（一个纠错单元中，错误比特数小于或等于最大纠错能力），则自动校正错误的地方，并且在寄存器中保存错误信息，同时产生相应的错误中断，软件可以相关寄存器中查询错误比特个数以及错误类型。

在ECC开启模式下，NANDC支持整Page的读写、只读OOB、只读数据头，在ECC0透传模式下，NANDC可以支持读ID、读器件状态等操作。具体操作流程可以参考后续章节。

13.6.1 ECC 错误信息查询

NANDC进行ECC错误校验时，每个纠错单元的错误比特数用一个字节存放。若此纠错单元出现可纠正错误，则对应的错误信息寄存器记录此数值；若出现不可纠正错误，则对应的错误信息寄存器的值为0xFF。NANDC最大支持4个片选，每个片选对应一套ECC错误信息寄存器。

ECC错误信息寄存器为NFC_CSx_PLNy_ERR_NUMz。其中x=0、1、2、3，分别表示cs0、cs1、cs2、cs3；y=0、1，分别表示plane0、plane1；z=0、1、2、3，分别表示第0-3、4-7、8-11、12-15 K数据段。

对于只读数据头操作，err_num均存放于NFC_CSx_PLNy_ERR_NUM0[7:0]。

对于只读OOB操作，64bit ECC模式16K Page_size配置下，NANDC会读取最后两个数据段，此时会有两个纠错单元被纠错，因此错误信息存放于NFC_CSx_PLNy_ERR_NUM0[15:0]。除此之外，其他配置下，NANDC只读取一个数据，错误信息存放于NFC_CSx_PLNy_ERR_NUM0[7:0]。

说明

16K Page_size的只读数据头/OOB操作中，只需要关注对应的错误寄存器的位域，其他位域的值不用关心。

13.6.2 ECC0 透传模式

ECC0透传模式用于支持NANDC读取器件ID、读取器件状态等不涉及到FLASH存储阵列数据的操作，或者是需要直接读写FLASH存储阵列数据的场景。

在ECC0透传模式下，NANDC不对数据进行任何处理。由于没有了数据结构的限制，软件可以指定数据长度。NFC_OP和NFC_OP_CTRL均支持ECC0透传模式。具体操作流程请参考后续章节。

说明

1. ECC0读写不支持只读数据头、只读OOB操作、不支持NFC_OP_CTRL的two plane读写操作。
2. ECC0模式下，8K Pagesize美光器件的同步模式中，使用NFC_OP_CTRL读操作将有可能导致挂死，因为此时NANDC将读取9120Byte的数据，超过了器件的Page范围。

13.6.3 空白页处理

在非Randomizer模式下，开启ECC读取空白页，即Page中的值全部为0xFF，NANDC不会上报不可纠错。此功能可以方便软件判断某个Page是否写入了数据，例如擦除某个Block之后，读取其中的Page不会出现不可纠错的错误的信息。

13.7 Boot 功能

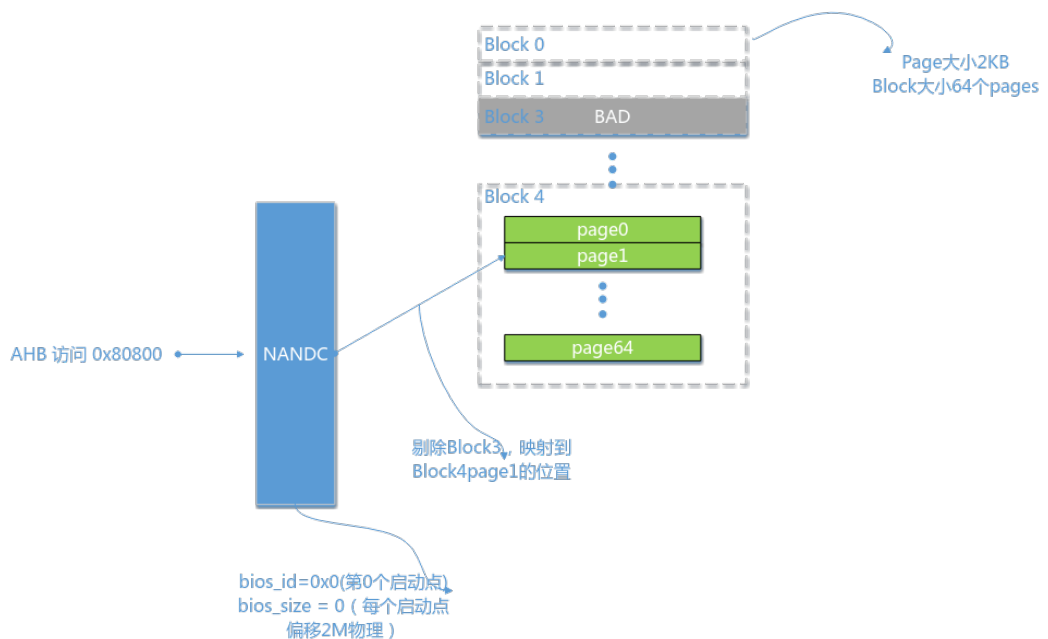
NANDC提供普通Boot模式，需要根据管脚的值来选择ECC、Page-size等信息。

NANDC复位之后，默认处于Boot模式，由NFC_CON[op_mode]决定。Boot模式控制器只会从片选0读取数据。

CPU可以读取NANDC BootSize 2M大小的数据。

NANDC自动将CPU请求的地址映射到NANDFLASH存储矩阵对应的位置中（映射地址时会剔除烧写时标记的坏块），按照器件接口时序将其读取，并返回总线。

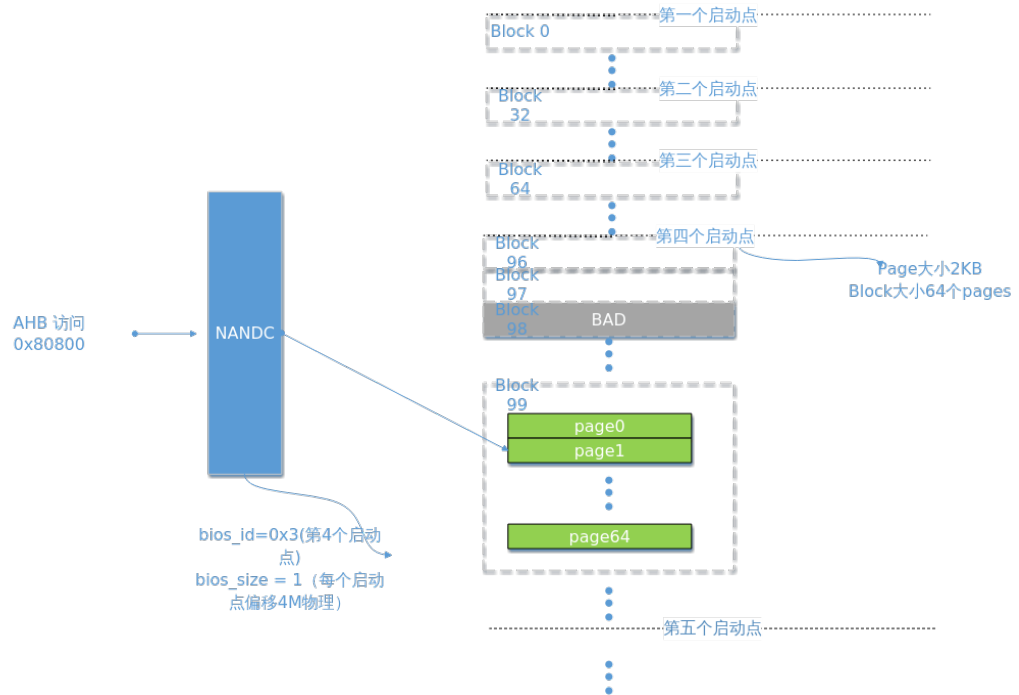
图 13-6 Boot 空间映射关系图



为了增加NAND FLASH Boot数据的可靠性。NANDC启动点多选，启动空间的起始地址可配置，NFC_BOOT_SET[bios_id]可选择0~7个启动点，NADC按照每个启动点偏移2M/4M（NFC_BOOT_SET[bios_size]决定）的物理大小进行偏移，之后再进行地址映射。

举例当前配置4M偏移，第4个启动点，访问地址映射关系如下图13-7所示。

图 13-7 多点启动 boot 空间映射关系图

**须知**

不管选择哪个采样点，CPU侧看到的空间地址范围都是从0开始的。

13.7.1 坏块识别

在将映射地址映射到Flash中的某个物理地址之前，控制器会先检查当前块及其之前所有块是否为良好状态，以确保映射地址的准确性。控制器通过读取每个物理块的第一个页（page0）和最后一个页的坏块标记位BB的值来判断该块是否完好。如果这两个页中任何一个的BB位不是0xFF，则控制器将认为当前块是坏块。

须知

坏块标记位（BB）在OOB的第一和第二个Byte，控制器只判断第一个Byte。具体在page中的位置因FLASH的数据结构不同而不同，具体参考章节“[数据结构](#)”。

13.8 TIMEOUT 功能

在部分操作中，NANDC需要等待NAND FLASH返回某些信号才能继续下一步操作。为了防止异常情况下的NAND FLASH的信号返回异常而导致NANDC挂死的情况，NANDC提供TIMEOUT机制：

同步模式下读操作中，NAND FLASH的DQS返回超时；



NFC_OP_CTRL的写、擦操作中，器件的RB信号拉高超时。

13.8.1 DQS 超时

在同步模式下，NANDC进行读数据操作时，需要等待NAND FLASH返回DQS信号。读操作发起后，NANDC开始计数，超时时长为0xFFFF个接口2x时钟周期(nf_clk_2x)，若超过此时间器件仍然未返回DQS信号，NANDC会出现DQS超时中断。DQS超时同步模式下的NFC_OP和NFC_OP_CTRL的读数据类操作均有效，如NFC_OP发起的读ID操作、读数据操作、NFC_OP_CTRL发起的只读数据头操作等等。

应用范围

DQS超时机制仅适用于读操作发起后等待器件返回的第一个DQS上升沿。一旦器件成功返回了第一个DQS上升沿，在这次读操作中，NANDC就不会再检查DQS是否超时。通常情况下，这已经足够应对大多数DQS异常情况。然而，对于某些特殊情况，NANDC仍可能无法妥善处理。比如，在一次读操作期间，尽管器件最初返回了第一个DQS上升沿，但随后没有继续发送任何DQS信号，这将导致NANDC陷入等待状态并最终挂起。已知的一种可能导致此问题的情况是：当对同步模式下的美光器件执行读操作且读取的数据量超过了该Page的实际大小时，器件会停止对超出部分数据的DQS响应，从而导致NANDC挂起。

使用方法

若操作中出现DQS超时中断，软件必须对NANDC进行软复位操作后才能重新正常工作。

13.8.2 RB 超时

NANDC使用NAND FLASH的RB管脚来判断器件是否处于busy状态。对于NFC_OP_CTRL的写入和擦除操作，NANDC提供了RB超时机制以防止因RB异常而导致的NANDC挂死。而对于手动配置的操作模式，NANDC不提供RB超时机制。

根据操作的类型的不同，RB超时可以分为三类：

- 对于two plane的写、two plane擦操作，plane0的命令下发后的busy时间，NANDC提供寄存器NFC_TIMEOUT_PLN0.nfc_timeout0_pln0用于超时设置。若plane0出现超时，plane1的命令以及数据将会被丢弃，并直接上报操作失败。
- 对于单plane的写、cache写、two plane的plane1写，NANDC提供NFC_TIMEOUT0_WR.nfc_timeout0用于超时设置，当此超时触发时，NANDC将开始查询器件状态的命令，查询次数和查询间隔均可配置，若达到最大查询次数或者某次状态查询到操作失败，NANDC将结束查询并上报操作失败。目前不支持查询状态次数的设置，需要将最大查询次数设置为0。
- 对于单plane的擦除、two plane的plane1擦除，NANDC提供NFC_TIMEOUT0_ERS.nfc_timeout0_ers用于超时设置，当此超时触发时，NANDC将开始查询器件状态的命令，查询次数和查询间隔均可配置，若达到最大查询次数或者某次状态查询到操作失败，NANDC将结束查询并上报操作失败。目前不支持查询状态次数的设置，需要将最大查询次数设置为0。

若在操作过程中出现RB超时，软件需要确定出现RB超时的原因，有可能是TIMEOUT相关参数配置不正确，或者NAND FLASH出现异常。



13.9 手动配置操作功能（NFC_OP）

NANDC的手动配置操作功能是根据NAND FLASH的操作特点设计的操作方法，软件可以使用此功能配置出不同的操作序列。

13.9.1 可配置操作要素

NFC_OP提供了七种操作要素：CMD1（命令1）、ADDR（地址）、WR_DATA（写数据）、CMD2（命令2）、WAIT_RB（等待RB）、RD_DATA（读数据）、RD_ST（读器件状态）。读器件状态RD_ST不属于操作要素的一种，但由于读器件状态操作的使用非常频繁，因此NANDC将其作为一个基本的操作要素。

图 13-8 NFC_OP 的操作要素

CMD1	ADDR	WR_DATA	CMD2	WAIT_RB	RD_DATA	RD_ST
------	------	---------	------	---------	---------	-------

当使用NFC_OP下发操作时，NANDC按照如下顺序依次判断并执行操作：CMD1_EN—>ADDR_EN—>WR_DATA_EN—>CMD2_EN—>WAIT_RB_EN—>RD_DATA_EN—>RD_ST_EN，请参见图13-8。

13.9.2 相关配置

在下发NFC_OP操作之前，需要根据相关要素的使能来配置其他寄存器，具体关系如下：

- CMD1：配置寄存器NFC_CMD.CMD1以指定要发送的命令。
- ADDR：配置NFC_ADDH.addr_h和NFC_ADDL.addr_l指定发送的地址，配置NFC_OP.address_cycles指定所发送的地址字节数。

说明

配置NFC_OP.address_cycles应该放在最后一步，因为一旦写寄存器NFC_OP，就会触发操作。

- WR_DATA：在ECC使能模式下，NANDC按约定数据结构的数据长度进行写。在ECC0透传模式下，需要配置NFC_DATA_NUM.nfc_data_num指定写的长度，并使能NFC_DATA_NUM.nfc_wr_rand表示当前为ECC0的写操作。此外，软件需要在Buffer中准备好相应长度的数据。
- CMD2：配置寄存器NFC_CMD.CMD2以指定要发送的命令。
- WAIT_RB：控制器自动等待器件的RB变高。
- RD_DATA：在ECC使能模式下，NANDC按约定数据结构的数据长度进行读操作。在ECC0透传模式下，需要配置NFC_DATA_NUM.nfc_data_num以指定读的数据长度，并关闭使能NFC_DATA_NUM.nfc_wr_rand。读操作完成后，数据将存放在Buffer中。
- RD_ST：配置NFC_CMD.read_status_cmd，用于指定读器件状态所使用的命令。操作完成后，读回的器件状态存放在NF_CSx_STATUS（x表示片选0、1、2、3）。

当上述相关寄存器配置完毕之后，再写寄存器NFC_OP，NANDC就会开始下发操作。寄存器NFC_OP中需要配置的项除了七种操作要素之外，还包含片选NFC_OP.nf_cs、地址周期NFC_OP.address_cycles、NFC_OP.read_id_en、NFC_OP.rw_reg_en，其中



NFC_OP.read_id_en只用在Toggle接口模式下读ID操作时使用。NFC_OP.rw_reg_en则在同步模式下读写器件内部寄存器的操作中使用，例如SET/GET FEATURES、READ STATUS、READ ID。更详细的流程可以参考典型操作流程“[典型应用配置](#)”。

13.9.3 多种读 Page 方式

NFC_OP提供了三种读取Page数据的方式：仅读取数据头、仅读取OOB功能以及完整读取整个Page。当采用仅读取数据头的方式时，NANDC只会读取Page中的第一个数据段；若选择仅读取OOB，则NANDC仅读取Page中包含OOB数据的那个数据段；而当需要完整读取整个Page时，NANDC将会完整地读取并输出该Page的所有数据。寄存器NFC_OP.PARA.rd_sel用于在NFC_OP模式下选择具体的读取Page方式。针对不同的读取Page方式，NFC_ADDRL的配置也会有所不同：

- 在执行仅读取数据头的操作时，应将NFC_ADDRL[15:0]（即页内偏移地址）设为0，此时NANDC会将指定Page的前1K字节（其长度等于sec_len）的数据读入Buffer，并从Buffer的起始位置开始存储这些数据。
- 对于只读OOB操作，NFC_ADDRL[15:0]需要配置为FLASH中OOB区所在的数据段的起始地址，具体页内偏移地址请参见[表13-2](#)。对于2K、4K、8K Page_size的配置，NANDC会把FLASH器件中指定page的OOB数据存放在NANDC内部Buffer偏移地址为Page_size起始的地方；对于16K的配置，OOB则会存放于第二次Buffer操作的OOB起始地址。具体地址请参考“[数据结构](#)”章节中Buffer中OOB的存放地址。

说明

对于16K Page_size，64bit ECC的配置，由于OOB区的BB和CTRL分别存放于两个不同的数据段，NANDC需要读出Page中的最后两个数据段到Buffer中。

- 对于读整个Page，页内偏移地址配置为0，NANDC将整个Page的数据读到Buffer中。

表 13-2 只读 OOB 的页内偏移地址

ECC (bit)	ecc_len (Byte)	sect_num	oob_len (Byte)	sect_len ^c (Byte)	NFC_ADDRL[15:0]
4 (8/1K)	14	2	30+2	1040	(sec_len+ecc_len)*(sec_num-1)
		4		1032	
8 (16/1K)	28	2	6+2	1028	
		4	14+2		
24/1K	42	2	30+2	1040	
		4		1032	
		8		1028	
28/1K	49	2	30+2	1040	(sec_len+ecc_len)*(sec_num-1)-1
		4		1032	



ECC (bit)	ecc_len (Byte)	sect_num	oob_len (Byte)	sect_len ^c (Byte)	NFC_ADDRL[15:0]
		8		1028	
40/1K	70	8	30+2	1028	(sec_len+ecc_len)*(sec_num-1)
		16		1026	sec_len*15+ecc_len*14
64/1K	112	8	30+2	1028	(sec_len+ecc_len)*(sec_num-1)
		16		1026	sec_len*15+ecc_len*14

13.10 自动配置操作功能（NFC_OP_CTRL）

NANDC提供自动配置操作功能，使用NFC_OP_CTRL操作，可以简化软件的配置，提高操作效率。NFC_OP_CTRL中涉及到读写的操作均为DMA读写，NANDC会自动实现NAND FLASH和DDR的数据交互。

13.10.1 操作类型

NFC_OP_CTRL提供八种操作：擦、写、读、two plane擦、two plane写、two plane读，cache读，cahce写。这八种操作的命令序列请参见表13-3。

表 13-3 NFC_OP_CTRL 支持的七种操作

操作类型	器件类型	命令序列 ^a (十六进制)	需要配置地址 ^b
读	不区分	00-30	plane0的DDR和FLASH地址
写	不区分	80-10	plane0的DDR和FLASH地址
擦	不区分	60-D0	plane0的FLASH地址
two plane 读	非toshiba	00-32-00-30, 06-E0-06-E0	plane0和plane1的 DDR和FLASH地址
	toshiba	60-60-30, 00-05-E0-00-05-E0	
two plane 写	非toshiba	80-11-80-10	plane0和plane1的 DDR和FLASH地址
	toshiba	80-11-81-10	



操作类型	器件类型	命令序列 ^a (十六进制)	需要配置地址 ^b
two plane 擦	非toshiba	60-D1-60-D0	plane0和plane1 的 FLASH地址
	toshiba	60-60-D0	
copyback	不区分	00-35-85-10	FLASH的源地址 和目的地址
cache read	不区分	00-31-31-31-3F	FLASH 起始页地 址 DDR起始地址
cache write	不区分	80-15-80-15-80-10	FLASH 起始页地 址 DDR起始地址

说明

- a: 命令序列仅给出命令字，命令字中间可能会穿插数据序列或者地址序列。
- b: 每个片选都有对应的DDR地址、FLASH地址、copyback源/目的地址配置寄存器。需要区分片选进行配置。

在下发命令之前，需要预先配置好对应cs的DDR地址（读写操作需要）和FLASH地址，然后配置NFC_OP_CTRL寄存器便可以下发指定操作，NANDC将自动完成命令、地址和数据的下发和传输。由于美光器件和东芝三星器件的two plane操作不同，NANDC提供寄存器NFC_CON.toshiba_nand来区分两种two plane操作。

13.10.2 自动查询器件状态

NFC_OP_CTRL对擦、写、two plane擦、two plane写，cache写这五种操作支持自动查询状态。在操作完成后，NANDC会自动发送读器件状态操作，并根据返回值判断此次操作是否执行成功。不同片选器件返回的状态值分别存放于寄存器NF_CS0_STATUS~NF_CS3_STATUS中。

NANDC只判断器件状态返回值的bit[0]和bit[6]。bit[0]表示器件的擦除或者写操作失败（包括copyback写），bit[6]表示器件是否处于BUSY状态。当某一次NFC_OP_CTRL上报失败中断后，需要查看对应的寄存器NF_CS_STATUS的bit[0]和bit[6]来判断失败原因。

13.10.3 多种读 Page 方式

NFC_OP_CTRL支持三种读Page方式：只读数据头、只读OOB、读整个Page。通过配置寄存器NFC_OP_CTRL.rd_op_sel可以选择不同的读Page类型。在ECC使能模式下，NFC_OP_CTRL的三种读Page操作均支持单plane读和two plane读。cache读只支持整个page。

不同于NFC_OP方式的读操作，在NFC_OP_CTRL中的读操作不需要关心页内偏移地址。所有读操作中，NFC_CSx_PLNy_ADDRL[15:0]（x=0/1/2/3，y=0/1）均配置为0。

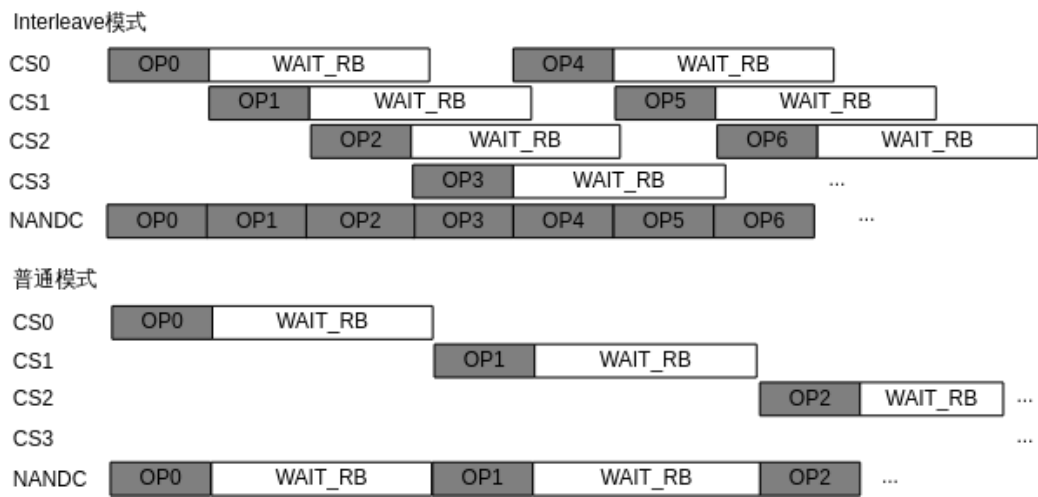


13.11 Interleave 功能

为了提高操作性能，充分地利用等待RB的空闲时间，NANDC为NFC_OP_CTRL提供Interleave功能，即在某个CS处于BUSY期间，NANDC可以下发另外一个CS的操作，以减少整体操作的耗时。

图13-9给出了Interleave模式和普通模式的对比示意图。在Interleave模式下，NANDC可以在接口处于BUSY或者空闲时接受其他CS的操作。CS0的OP0（命令地址数据等）操作完成后，CS0进去BUSY状态，此时NANDC可以执行CS1的OP1。依次操作完成后，若CS0操作完成，软件可以继续给CS0下发操作，使NANDC处于连续执行操作的状态，这样可以减少操作的等待时间。而在普通模式下，NANDC必须等待上一次操作完全结束之后才能执行下一个命令。

图 13-9 Interleave 调度模式与普通模式对比



13.12 中断处理

NANDC对外部提供一个中断信号，其内部的所有中断共用一个外部中断信号线。每个片选均有一套独立的中断寄存器，因此当软件检测到外部中断之后，需要轮询所有片选的中断状态寄存器以确定中断来源。

NANDC内部包含多种中断源，中断类型请参考表13-4。

表 13-4 中断源类型

中断	描述
blkpage_rd	开加扰后，非ECC0模式，读取空白页产生的中断。
ecc_mem_err	ECC模块的纠错IP的mem产生1bit错误中断。
ecc_bad_mem_err	ECC模块的纠错IP的mem产生超过1bit错误中断。



中断	描述
dqs_timeout	同步模式下，读类型的操作中，NAND FLASH长时间不返回DQS超时中断。
dma_err	DMA接口的总线返回错误响应中断。
ahb_op_err	NANDC对器件进行操作器件，CPU访问Buffer中断。 注意：ahb_op_err不区分片选，无论哪一片选发生ahb_op_err中断，都会在cs0的中断状态寄存器中反映。
err_invalid	读数据操作中，数据出现可纠正错误中断。
err_valid	读数据操作中，数据出现不可纠正错误中断。
op_fail	NFC_OP_CTRL的写、擦、two plane写、two plane擦、cache写 操作失败中断。
busy	器件进入busy状态中断。
op_done	操作完成中断。

13.13 数据结构

在ECC0透传模式下，NANDC不对软件数据进行任何处理，读写操作中，数据数据保持原样地写入/读出。

在非ECC0模式下，NANDC对写入/读出NAND FLASH的数据进行ECC编码/解码操作。ECC的编码/解码都是以一个纠错单元为单位进行，不同的ECC类型和Page_size配置对应不同的长度的纠错单元和ECC码。

非ECC0模式下，NANDC涉及到的数据格式有三种：软件数据格式，Buffer中的数据格式，FLASH中的数据格式。以写操作为例，下面分别说明这三种数据格式。

软件数据格式

在启动写Page操作之前，软件需要准备好软件数据。软件数据由DATA+OOB组成，其中DATA的长度data_len（Page_size），OOB的长度则由当前ECC配置和Page_size配置决定。具体对应关系请参考表13-5。

如果使用手动配置操作的方式下发写Page操作，则软件需要先将软件数据直接存放到Buffer；若使用自动配置操作的方式下发写Page操作，则软件将数据存放在指定的DDR中。

Buffer 中的数据格式

在写Page操作过程中，NANDC在Buffer中进行ECC编码的过程：

- 对于2K、4K、8K Page_size的器件，NANDC内部Buffer足够大，在Buffer中只进行一次数据处理就可以完成一次写操作。软件数据从NANDC内部Buffer的0地址开始存放，NANDC将总长为（Page_size+oob_len）的软件数据分割成（Page_size/1024）个数据段DATA_SECT，每个数据段长度为sec_len，ECC模块以数据段为单元，为每个数据段进行ECC编码，每个数据段生成长度为ecc_len的



ECC码，生成的ECC码存放在Buffer中8320Byte开始后的区域，各个数据段的ECC码连续排布在此区域。

- 对于16K Page_size的器件，受限于内部Buffer大小，NANDC需要在Buffer中进行两次数据处理才能完成整个Page的操作。总长16416Byte的软件数据分割成16数据段（每段长度为1026Byte）。在第一次操作中，NANDC对前8个数据段（1026Byte*8）的DATA生成8段ECC码（ecc_len*8）；在第二次操作中，NANDC对剩下的8个数据段（1026Byte*8中包含32Byte的BB+CTRL）生成8段ECC码（ecc_len*8）。每次操作时，8段ECC码都固定从Buffer的8320Byte地址开始连续存放。

FLASH 中的数据格式

数据从Buffer写入到NAND FLASH的过程中，NANDC以DATA_SECT+ECC的格式交替存放所有数据段，但对于最后一个数据段的OOB数据，其存放位置有特殊要求。

- BB数据需要放在FLASH页中偏移为Page_size的地方，即对于2K、4K、8K、16K的配置，BB数据将被放置到2048、4096、8192、16384Byte开始的两个字节。
- CTRL数据需要放置在整个数据的末尾。

其对应关系见表13-5。表中给出了NANDC所支持的数据结构参数。

表 13-5 数据结构参数

ECC (bit)	ecc_len ^a (Byte)	data_len /Page_size (Byte)	oob_len ^b (Byte)	sec_len ^c (Byte)
4/512B ^d (8/1K ^e)	14	2048	30+2 ^f	1040
		4096		1032
8/512B (16/1K)	28	2048	6+2	1028
		4096	14+2	
24/1K	42	2048	30+2	1040
		4096		1032
		8192		1028
28/1K	49	2048	30+2	1040
		4096		1032
		8192		1028
40/1K	70	8192	30+2	1028
		16384		1026
64/1K	112	8192	30+2	1028
		16384		1026



说明

- ecc_len表示每个纠错单元生成的ECC码的长度。
- oob_len表示上层软件可见的冗余区长度，由CTRL+BB组成。
- sec_len表示每个纠错单元的数据区长度， $\text{sec_len}=1024+(\text{oob_len}*1024)/\text{Page_size}$
- 此处使用4bit/512B和8bit/1KB这两种表示方法主要是为了兼容某些器件手册中对ECC规格的描述。
- ECC的每个纠错单元为1K，此处1K并不是严格意义上的1024Byte，而是指长度处于1K这一数量级。
- OOB数据由CTRL数据和BB数据组成，BB数据固定为2Byte。

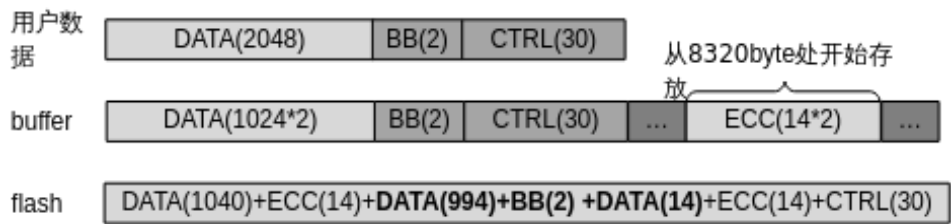
关于软件数据格式、Buffer中数据格式以及FLASH中数据格式的更加详细的数据结构将在下面的章节进行描述。

13.13.1 4bit ECC 模式(纠错 8bit/1KB)

13.13.1.1 2KB Page_size

对2KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-10所示。

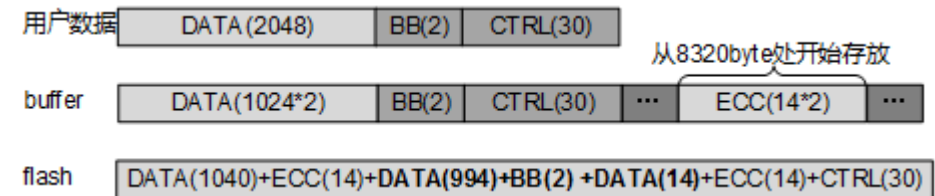
图 13-10 4bit ECC 模式 2K Page_size 的数据结构



13.13.1.2 4KB Page_size

对4KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-11所示。

图 13-11 4bit ECC 模式 4K Page_size 的数据结构

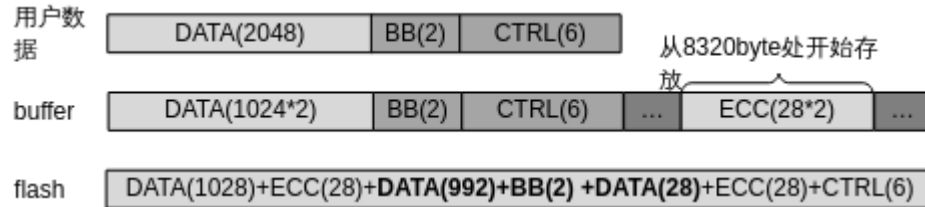


13.13.2 8bit ECC 模式(纠错 16bit/1KB)

13.13.2.1 2KB Page_size

对2KB Page_size的配置，软件可用的冗余区大小为8Byte。数据在Buffer和FLASH中的结构如图13-12所示。

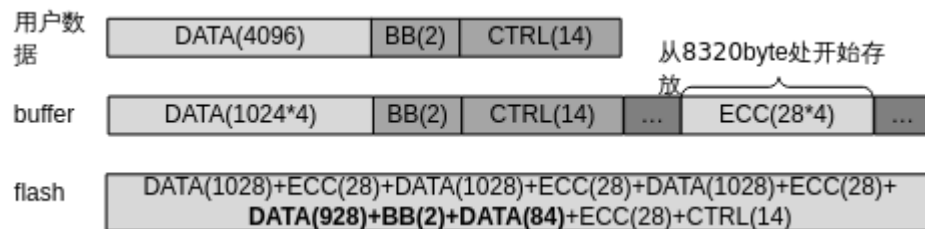
图 13-12 8bit ECC 模式 2K Page_size 的数据结构



13.13.2.2 4KB Page_size

对4KB Page_size的配置，软件可用的冗余区大小为16Byte。数据在Buffer和FLASH中的结构如图13-13所示。

图 13-13 8bit ECC 模式 4K Page_size 的数据结构

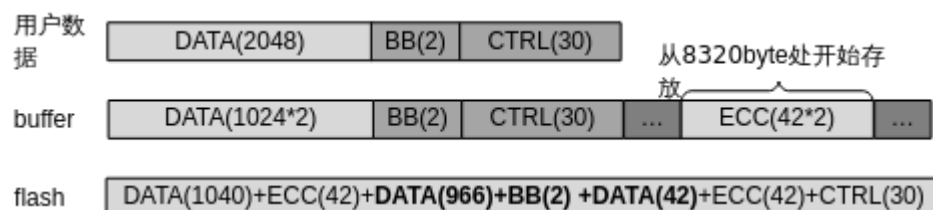


13.13.3 24bit ECC 模式(纠错 24bit/1KB)

13.13.3.1 2KB Page_size

对2KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-14所示。

图 13-14 24bit ECC 模式 2K Page_size 的数据结构

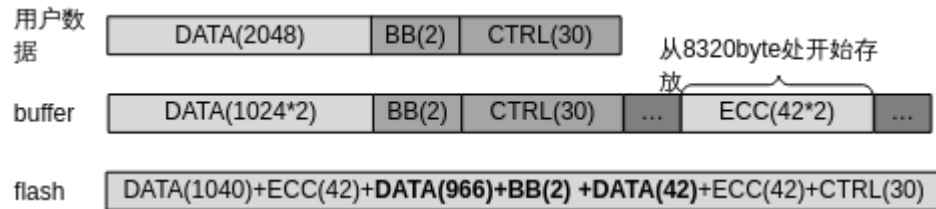




13.13.3.2 4KB Page_size

对4KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-15所示。

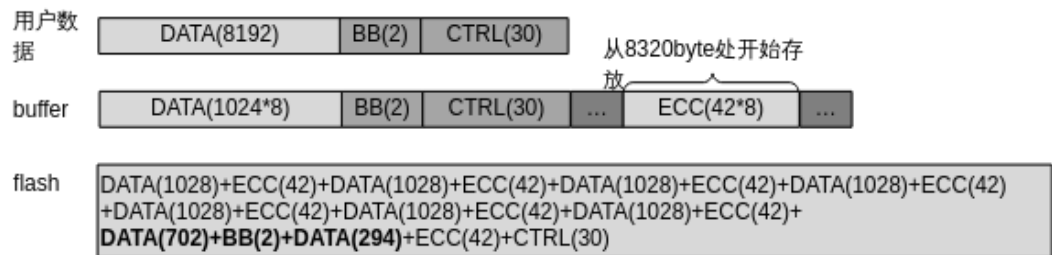
图 13-15 24bit ECC 模式 4K Page_size 的数据结构



13.13.3.3 8KB Page_size

对8KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-16所示。

图 13-16 24bit ECC 模式 8K Page_size 的数据结构

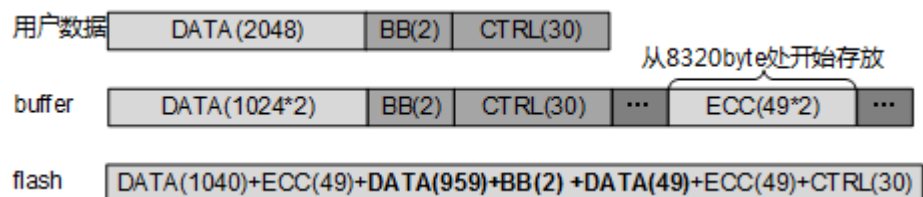


13.13.4 28bit ECC 模式(纠错 28bit/1KB)

13.13.4.1 2KB Page_size

对2KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-17所示。

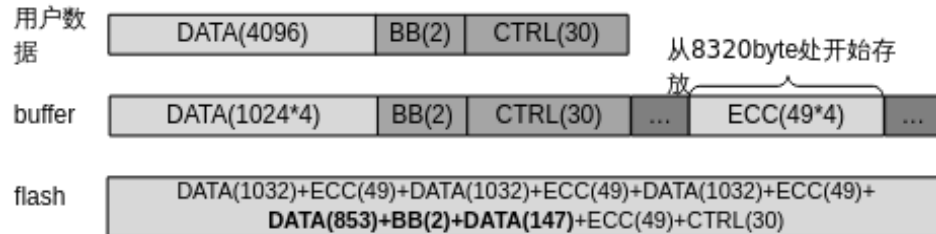
图 13-17 28bit ECC 模式 2K Page_size 的数据结构



13.13.4.2 4KB Page_size

对4KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-18所示。

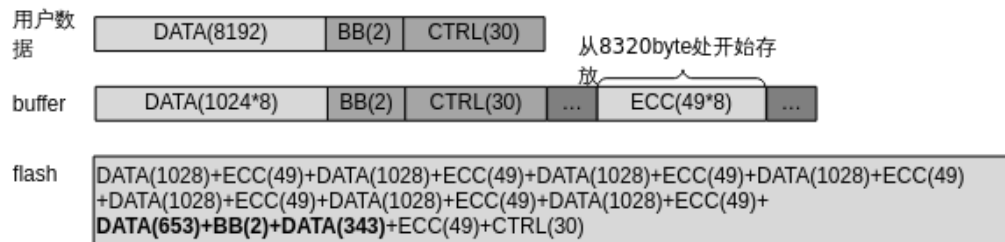
图 13-18 28bit ECC 模式 4K Page_size 的数据结构



13.13.4.3 8KB Page_size

对8KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-19所示。

图 13-19 28bit ECC 模式 8K Page_size 的数据结构

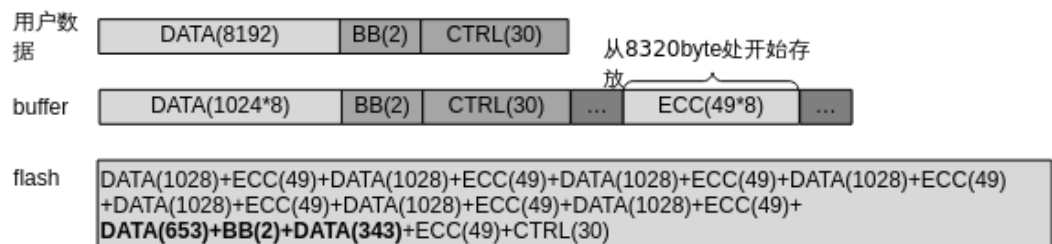


13.13.5 40bit ECC 模式(纠错 40bit/1KB)

13.13.5.1 8KB Page_size

对8KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-20所示。

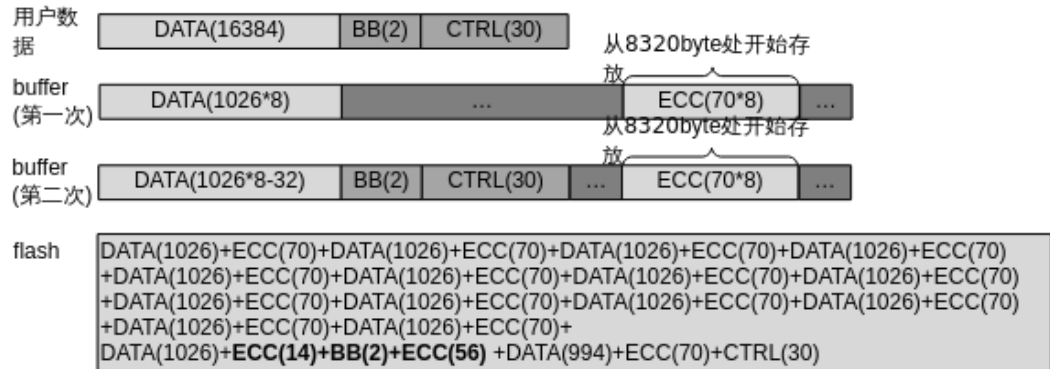
图 13-20 40bit ECC 模式 8K Page_size 的数据结构



13.13.5.2 16KB Page_size

对16KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-21所示。

图 13-21 40bit ECC 模式 16K Page_size 的数据结构

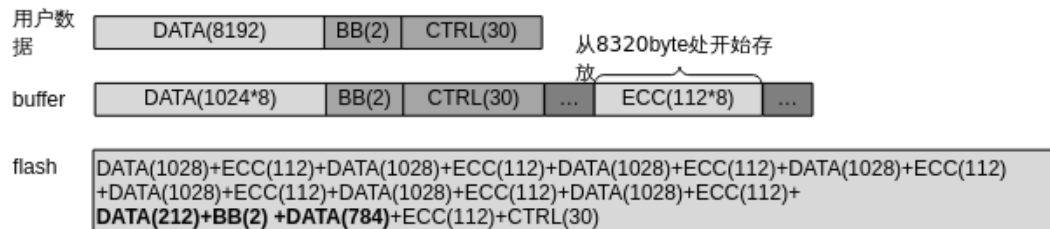


13.13.6 64bit ECC 模式(纠错 40bit/1KB)

13.13.6.1 8KB Page_size

对8KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-22所示。

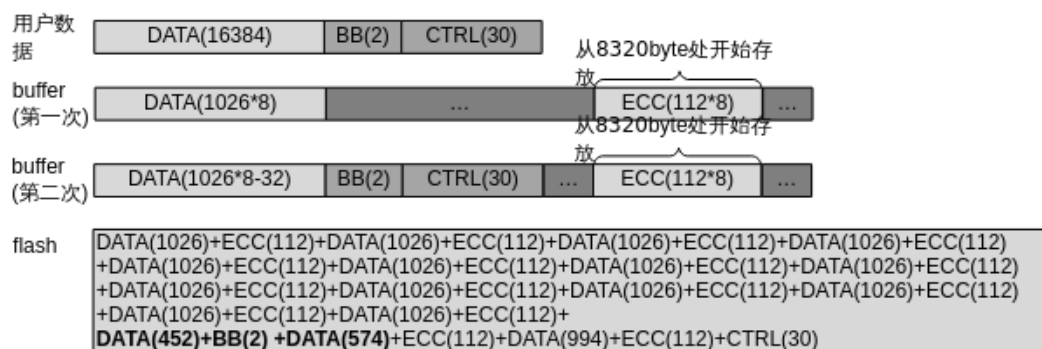
图 13-22 64bit ECC 模式 8K Page_size 的数据结构



13.13.6.2 16KB Page_size

对16KB Page_size的配置，软件可用的冗余区大小为32Byte。数据在Buffer和FLASH中的结构如图13-23所示。

图 13-23 64bit ECC 模式 16K Page_size 的数据结构



13.14 典型应用配置

13.14.1 NANDC 软复位流程

NANDC的软复位由BI_SUBCTRL模块控制，需要配置BI_SUBCTRL模块的寄存器实现软复位：

- 步骤1** 配置寄存器SC_NF_RESET_REQ. srst_req_nf为0x1，使能NANDC软复位请求。
- 步骤2** 配置寄存器SC_NF_ICG_DIS. icg_dis_nf为0x1，关闭NANDC的时钟使能。
- 步骤3** 配置寄存器SC_NF_RESET_DREQ. srst_dreq_nf为0x1，解除NANDC软复位请求。
- 步骤4** 配置寄存器SC_NF_ICG_EN. icg_en_nf为0x1，使能NANDC的时钟。软复位流程结束。

----结束

13.14.2 初始化

NANDC的初始化步骤如下:

- 步骤1** 对NANDC进行软复位，请参考“[NANDC软复位流程](#)”。
- 步骤2** 配置寄存器NFC_ARB_EN.interleave_mode为0，确保处于非Interleave模式下。
- 步骤3** 清除所有中断状态，并根据需要配置中断使能。
- 步骤4** 读取器件的ID，根据ID确定器件的规格并配置NFC_CON中的ecc_type、pagesize等以及NFC_BOOT_SET.block_size，NFC_CON.bus_width默认值为1，表示数据线位宽16bit，实际使用FLASH位宽为8bit，需要将NFC_CON.bus_width配置为0。
- 步骤5** 配置寄存器NFC_TIMEOUT1_WR和寄存器NFC_TIMEOUT1_ERS为0。
- 步骤6** 对器件进行复位操作，复位器件。
- 步骤7** 确定工作模式，配置相关时间参数。若工作在异步模式，则需要配置NFC_PWIDTH；若工作在同步模式则需要配置NFC_SYNC_TIMING。此外根据需要配置操作间隔寄存器NFC_OPIDLE。
- 步骤8** 如果DLL需要初始化的话，需要根据具体功能配置寄存器NFC_DLL_CTRL，改变设置后，需要等待寄存器DLL_STATUS.dll_locked和DLL_STATUS.dll_ready均为1，表示DLL初始化完成。



步骤9 NANDC初始化完成，可根据具体的操作配置寄存器下发操作。

----结束

13.14.3 NANDC 操作结束判断流程

NANDC支持两种操作：NFC_OP操作和NFC_OP_CTRL操作。无论进行哪一种操作，NANDC完成此次操作后，NFC_INTS_CSx.csx_op_done（x表示片选）都会被置1。软件可以使用查询和中断这两种方式判断NANDC是否完成操作。步骤如下：

步骤1 若使用查询方式，则查询寄存器NFC_INTS_CSx.csx_op_done（x表示片选），直到其变为1；若使用中断方式，则等待中断触发并判断NFC_INTS_CSx.csx_op_done是否变为1（x表示片选）。若检测到NFC_INTS_CSx.csx_op_done为1，则表示NANDC完成对片选x的操作，软件可以下发新的操作了。软件完成判断后需要清除此中断状态，以免影响下一次操作的判断。

说明

若使用查询方式，NANDC还提供了另一个寄存器NFC_STATUS用于指示NANDC是否完成操作。软件可以使用查询方式查询NFC_STATUS.nfx_ready是否为（x表示片选）来判断。但是使用查询NFC_STATUS.nfx_ready的方式，需要在配置NFC_OP或者NFC_OP_CTRL之后，读一次NFC_OP或者NFC_OP_CTRL寄存器，然后才能进行查询NFC_STATUS.nfx_ready的流程，否则在配置NFC_OP或者NFC_OP_CTRL之后直接查询NFC_STATUS.nfx_ready可能会读到不准确的状态。

----结束

13.14.4 器件复位流程

NAND FLASH器件上电后需要被复位之后才能使用。异步接口模式下，器件复位操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。
- 步骤3** 配置寄存器NFC_CMD为器件复位的操作命令0xFF。
- 步骤4** 配置寄存器NFC_OP为0x44，启动cs0的器件复位操作。
- 步骤5** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。

----结束

13.14.5 读 ID 流程

对器件进行复位操作后，需要读取器件ID来确定器件类型，下面以异步模式下的读ID操作为例：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。
- 步骤3** 配置寄存器NFC_ADDRL为读ID的操作地址0x00，配置寄存器NFC_CMD.cmd1为器件复位的操作命令0x90。
- 步骤4** 配置寄存器NFC_DATA_NUM.nfc_data_num为0x5，表示读取5个Byte的ID值。



步骤5 配置寄存器NFC_OP为0x66，启动对cs0的读器件ID操作；若此时器件处于ONFI同步模式，则需要配置NFC_OP为0x2066；若器件处于TOGGLE模式，则需要配置NFC_OP为0x3066。

步骤6 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。

步骤7 软件读取Buffer的前5个Byte获取ID值。ID值按接口读出的先后顺序依次存放到Buffer的0、1、2、3、4地址。

----结束

13.14.6 读器件状态流程

使用NFC_OP对器件进行读器件状态操作有两种方式，一种是软件配置使能，另一种是直接使用控制器提供的RD_ST操作要素使能。

13.14.6.1 软件配置使能读器件状态操作流程

步骤1 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。

步骤2 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。

步骤3 配置命令寄存器NFC_CMD.read_status_cmd为0x70，配置读写数据寄存器NFC_DATA_NUM.nfc_data_num为1。

步骤4 配置寄存器NFC_OP为0x42，启动读器件状态操作。

步骤5 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。

步骤6 软件读取Buffer中的0地址的1Byte数据，即为器件状态值。

----结束

13.14.6.2 使用 RD_ST 使能读取状态寄存器的操作流程

步骤1 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。

步骤2 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。

步骤3 配置命令寄存器NFC_CMD.read_status_cmd为0x70。

步骤4 配置寄存器NFC_OP为0x1，启动器件的读器件状态操作。

步骤5 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。

步骤6 软件读取状态寄存器NF_CS0_STATUS.nf_cs0_status，获取器件状态。

----结束

13.14.7 接口模式切换流程

对于美光器件，部分器件支持异步和NV_DDR、异步和NV_DDR2的模式互相切换；对于三星器件，一般仅支持异步或者TOGGLE模式，不需要切换；对于东芝器件，部分器件支持异步和TOGGLE模式互相切换。下面以支持切换的美光器件为例，对异步模式和同步模式的互相切换流程进行说明。



13.14.7.1 异步切换到 NV_DDR

对于支持同步模式的ONFI器件，上电复位后默认都处于异步模式，若要使用NV_DDR同步模式，则需要下发SET FEATURE操作进行切换操作。下面以美光器件从异步模式0切换到NV_DDR模式0为例，操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。
- 步骤3** 配置寄存器NFC_ADDRL写入SET FEATURE的操作地址0x01，配置寄存器NFC_CMD为SET FEATURE的操作命令0xEF。
- 步骤4** 配置寄存器NFC_DATA_NUM.nfc_data_num为0x4，表示SET FEATURE需要对器件内部寄存器写入4Byte的配置值。写入Buffer的0、1、2、3地址的字节分别为0x10、0、0、0。
- 步骤5** 配置寄存器NFC_OP为0x274，启动对cs0的SET FEATURE操作。
- 步骤6** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤7** 配置NFC_CON.nf_mode为0x2，将NANDC切换到NV_DDR模式。

----结束

13.14.7.2 NV_DDR 切换到异步模式

对于支持同步模式的ONFI器件，切换到同步模式后，可以再次通过SET FEATURE操作将器件切回到异步模式。下面以美光器件从NV_DDR模式0切换到异步模式0为例，操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。
- 步骤3** 配置寄存器NFC_ADDRL写入SET FEATURE的操作地址0x01，配置寄存器NFC_CMD为SET FEATURE的操作命令0xEF。
- 步骤4** 配置寄存器NFC_DATA_NUM.nfc_data_num为0x4，表示SET FEATURE需要对器件内部寄存器写入4Byte的配置值。写入Buffer的0、1、2、3地址的字节分别为0、0、0、0。
- 步骤5** 配置寄存器NFC_OP为0x2274，启动对cs0的SET FEATURE操作。
- 步骤6** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤7** 配置NFC_CON.nf_mode为0x0，将NANDC切换到异步模式。

----结束

说明

在配置NFC_OP模式切换时，不应在SET FEATURE操作中启用自动读取设备状态的功能（NFC_OP.read_status_en）。因为一旦NANDC执行读取设备状态的操作，其接口模式可能会与NAND FLASH当前的接口模式不一致（由于SET FEATURE操作已将NAND FLASH切换到了另一种模式），这可能导致NANDC出现故障。



13.14.8 擦除操作流程

NFC_OP和NFC_OP_CTRL均支持擦除操作，其中NFC_OP_CTRL还支持two plane擦除操作，下面分别对这些操作进行说明。

13.14.8.1 NFC_OP 擦除

NFC_OP方式的擦除片选0的操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_ADDRL写入需要擦除Block地址，向寄存器NFC_CMD写入擦除命令0x70D060。
- 步骤3** 配置寄存器NFC_OP写入0x66D，启动擦除操作(对cs0进行操作)。
- 步骤4** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤5** 读操作片选寄存器NF_CS0_STATUS.nf_cs0_status，判断擦除是否成功。

----结束

13.14.8.2 NFC_OP_CTRL 擦除/two plane 擦除

NFC_OP_CTRL方式下的擦除片选0的操作流程如下：

- 步骤1** 配置寄存器NFC_CS0_PLN0_ADDRL为需要擦除的Block地址。如果是two plane操作，则还需要配置plane1的Block地址到寄存器NFC_CS0_PLN1_ADDRL。
- 步骤2** 配置寄存器NFC_OP_CTRL为0x87，下发擦除操作。若要下发two plane操作，则配置寄存器NFC_OP_CTRL为0x89。
- 步骤3** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤4** 检测NFC_INTS_CS0.cs0_op_fail判断操作是否成功。

----结束

13.14.9 写操作流程

NFC_OP和NFC_OP_CTRL均支持写操作，其中NFC_OP_CTRL还支持two plane写操作，下面分别对这些操作进行说明。

13.14.9.1 NFC_OP 写

NFC_OP方式下，16K Page_size的写操作需要分两次Buffer操作，其他配置则只需要一次Buffer操作。

2、4、8K Page_size下，写片选0的操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 分别配置地址寄存器寄存器NFC_ADDRH、NFC_ADDRL。向寄存器NFC_CMD写入写命令0x701080。
- 步骤3** 软件按对应Page_size的数据结构，对Buffer写入数据。



步骤4 配置寄存器NFC_OP写入0xA7D，启动写操作。

步骤5 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。

步骤6 读操作片选寄存器NF_CS0_STATUS.nf_cs0_status，判断写操作是否成功。

----结束

16K Page_size下，写片选0的操作流程如下：

步骤1 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。

步骤2 分别配置地址寄存器寄存器NFC_ADDRH、NFC_ADDRL。向寄存器NFC_CMD写入写命令0x701080。

步骤3 配置NFC_SEGMENT_ID.last_flag为0，表示16K数据的前8K操作。软件按对应Page_size的数据结构，对Buffer写入数据。

步骤4 配置寄存器NFC_OP写入0xA74，启动前8K的写操作。

步骤5 查询方式下，检测NFC_STATUS.nfc_ready，如果为1，进入下一步，否则，重复此步骤；中断方式下，响应中断，若检测到中断状态寄存器NFC_INTS_CS0.cs0_op_done为1，则进入下一步。

步骤6 配置NFC_SEGMENT_ID.last_flag为1，表示16K数据的后8K操作。软件按对应Page_size的数据结构，对Buffer写入数据。

步骤7 配置寄存器NFC_OP写入0x1D，启动后8K的写操作。

步骤8 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。

步骤9 读操作片选寄存器NF_CS0_STATUS.nf_cs0_status，判断写操作是否成功。

----结束

13.14.9.2 NFC_OP_CTRL 写/two plane 写/FLASH cache 写

NFC_OP_CTRL方式下的写片选0的操作流程如下：

步骤1 在DDR中准备好数据，配置DDR基地址寄存器：NFC_CS0_PLN0_BADDRH/L_Dx(x=0、1、2、3，根据Page_size配置所需的寄存器)，NFC_CS0_PLN0_BADDRH/L_OOB。除OOB基地址外，每个基地址最多可以存放4KB的数据。对于two plane操作还需配置plane1对应的基地址寄存器：NFC_CS0_PLN1_BADDRH/L_Dx，NFC_CS0_PLN1_BADDRH/L_OOB。cache 写操作只需要配置NFC_CS0_PLN0_BADDRH/L_D0和NFC_CS0_PLN0_BADDRH/L_OOB。

步骤2 配置器件地址寄存器NFC_CS0_PLN0_ADDRH、NFC_CS0_PLN0_ADDRL。对于two plane操作还需要配置plane1对应的地址寄存器NFC_CS0_PLN1_ADDRH、NFC_CS0_PLN1_ADDRL。

步骤3 若进行cache写操作，须配置NFC_CACHE_CTRL寄存器，配置cache连续写的页数。

步骤4 配置寄存器NFC_OP_CTRL为0x83，下发写操作。若要执行two plane操作，则配置寄存器NFC_OP_CTRL为0x85。若要执行cache写操作，则配置寄存器NFC_OP_CTRL为0x283。

步骤5 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。



步骤6 检测NFC_INTS_CS0.cs0_op_fail判断操作是否成功。

----结束

13.14.10 读操作流程

NFC_OP和NFC_OP_CTRL均支持读操作，其中NFC_OP_CTRL还支持two plane读操作，下面分别对这些操作进行说明。

13.14.10.1 NFC_OP 读

NFC_OP方式下，16K Page_size的读操作需要分两次Buffer操作，其他配置则只需要一次Buffer操作。

2、4、8K Page_size下，读片选0的操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_OP_PARA.rd_sel为0x0，表示将要进行整Page的读操作。
- 步骤3** 分别配置地址寄存器寄存器NFC_ADDRH、NFC_ADDRL。向寄存器NFC_CMD写入写命令0x0070_3000。
- 步骤4** 配置寄存器NFC_OP写入0xA6E，启动读操作。
- 步骤5** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤6** 软件按对应Page_size的数据结构，从Buffer读出数据。

----结束

16K Page_size下，读片选0的操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_OP_PARA.rd_sel为0x0，表示将要进行整Page的读操作。
- 步骤3** 分别配置地址寄存器寄存器NFC_ADDRH、NFC_ADDRL。向寄存器NFC_CMD写入写命令0x703000。
- 步骤4** 配置NFC_SEGMENT_ID.last_flag为0，表示16K数据的前8K操作。
- 步骤5** 配置寄存器NFC_OP写入0xA6E，启动前8K的写操作。
- 步骤6** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤7** 软件按对应的数据结构读取Buffer中数据。
- 步骤8** 配置NFC_SEGMENT_ID.last_flag为1，表示16K数据的后8K操作。
- 步骤9** 配置寄存器NFC_OP写入0x2，启动后8K的写操作。
- 步骤10** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤11** 软件按对应的数据结构读取Buffer中数据。

----结束



13.14.10.2 NFC_OP_CTRL 读/two plane 读/FLASH cache 读

NFC_OP_CTRL方式下的读片选0的操作流程如下：

- 步骤1** 配置DDR基地址寄存器：NFC_CS0_PLN0_BADDRH/L_Dx(x=0、1、2、3，根据Page_size配置所需的寄存器)，NFC_CS0_PLN0_BADDRH/L_OOB。除OOB基地址外，每个基地址最多可以存放4KB的数据。对于two plane操作还需配置plane1对应的基地址寄存器：NFC_CS0_PLN1_BADDRH/L_Dx，NFC_CS0_PLN1_BADDRH/L_OOB。cache 读操作只需要配置NFC_CS0_PLN0_BADDRH/L_D0和NFC_CS0_PLN0_BADDRH/L_OOB。
- 步骤2** 配置器件地址寄存器NFC_CS0_PLN0_ADDRH、NFC_CS0_PLN0_ADDRL。对于two plane操作还需要配置plane1对应的地址寄存器NFC_CS0_PLN1_ADDRH、NFC_CS0_PLN1_ADDRL。
- 步骤3** 若进行cache读操作，须配置NFC_CACHE_CTRL寄存器，配置cache连续读的页数。
- 步骤4** 配置寄存器NFC_OP_CTRL为0x81，下发读操作。若要执行two plane操作，则配置寄存器NFC_OP_CTRL为0x8B。若要执行cache读操作，则配置寄存器NFC_OP_CTRL为0x281。
- 步骤5** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤6** 软件访问DDR中的数据。

----结束

13.14.11 只读 OOB 操作流程

NFC_OP和NFC_OP_CTRL均支持只读OOB作，其中NFC_OP_CTRL还支持two plane的只读OOB操作，下面分别对这些操作进行说明。

13.14.11.1 NFC_OP 只读 OOB

NFC_OP方式下只读OOB，一次Buffer操作即可以完成，其操作流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_OP_PARA.rd_sel为0x1，表示将要进行只读OOB的读操作。对于16K Page_size的配置，还需要配置NFC_SEGMENT_ID.last_flag为0x1，对于其他Page_size，配置NFC_SEGMENT_ID.last_flag为0x0。
- 步骤3** 分别配置地址寄存器寄存器NFC_ADDRH、NFC_ADDRL。向寄存器NFC_CMD写入写命令0x0070_3000。

说明

对于FC_OP方式的只读OOB，除了配置器件的page地址之外，还需要配置OOB的页内偏移地址，具体偏移地址请参考[表13-2](#)。

- 步骤4** 配置寄存器NFC_OP写入0xA6E，启动只读OOB操作。
- 步骤5** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤6** 软件按对应Page_size的数据结构，从Buffer读出数据。

----结束



13.14.11.2 NFC_OP_CTRL 只读 OOB/two plane 只读 OOB

NFC_OP_CTRL方式下的读片选0的操作流程如下：

- 步骤1** 配置OOB的DDR基地址寄存器NFC_CS0_PLN0_BADDRH/L_OOB。对于two plane操作还需配置plane1对应的基地址寄存器NFC_CS0_PLN1_BADDRH/L_OOB。
- 步骤2** 配置器件地址寄存器NFC_CS0_PLN0_ADDRH、NFC_CS0_PLN0_ADDRL。对于two plane操作还需要配置plane1对应的地址寄存器NFC_CS0_PLN1_ADDRH、NFC_CS0_PLN1_ADDRL。
- 说明**
- 对于NFC_OP_CTRL模式的只读OOB操作，配置器件地址时只需要给出page地址即可，不需要配置页内偏移地址。
- 步骤3** 配置寄存器NFC_OP_CTRL为0x181，下发只读OOB操作。若要执行two plane操作，则配置寄存器NFC_OP_CTRL为0x18B。
- 步骤4** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤5** 软件访问DDR中的OOB数据（存放于NFC_CS0_PLN0_BADDRH/L_OOB）。

----结束

13.14.12 ECC0 透传操作流程

NFC_OP和NFC_OP_CTRL均支持ECC0透传模式的读写操作，下面分别对这些操作进行说明。

13.14.12.1 NFC_OP 的 ECC0 透传操作

NFC_OP方式下的操作片选0的流程如下：

- 步骤1** 配置寄存器NFC_CON.op_mode为0x1，配置寄存器NFC_ARB_EN.interleave_mode为0，确保NANDC处于normal模式下的非Interleave模式。
- 步骤2** 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。
- 步骤3** 若为写操作，则软件在Buffer写入数据。
- 步骤4** 分别配置地址寄存器寄存器NFC_ADDRH、NFC_ADDRL。向寄存器NFC_CMD写入写命令0x0070_1080（写操作）或者读命令0x0070_3000（读操作）。
- 步骤5** 配置寄存器NFC_DATA_NUM.nfc_data_num为操作的数据长度，若为写操作还需要配置NFC_DATA_NUM.nfc_wr_rand使能。
- 步骤6** 配置寄存器NFC_OP写入0xA7D（写操作）或者0xA6E（读操作），启动操作。
- 步骤7** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤8** 若为写操作，则读取寄存器NF_CS0_STATUS.nf_cs0_status，判断写操作是否成功。若为读操作则访问DDR中的数据。

----结束

13.14.12.2 NFC_OP_CTRL 的 ECC0 透传操作

NFC_OP_CTRL方式下操作片选0的流程如下：



- 步骤1** 配置寄存器NFC_CON.ecc_type为0，确保处于ECC0模式。
- 步骤2** 配置DDR基地址寄存器：NFC_CS0_PLN0_BADDRH/L_Dx(x=0、1、2、3，根据Page_size配置所需的寄存器)，NFC_CS0_PLN0_BADDRH/L_OOB。除OOB基地址外，每个基地址最多可以存放4KB的数据。
- 步骤3** 配置OOB数据长度NFC_DMA_LEN.len_oob。
- 说明**
- ECC0透传模式下，NFC_OP_CTRL读操作数据长度固定，对应不同的Page_size分别为、4KB+232B、8KB+928B、16KB+1824B，而写操作数据长度则可以配置，为Page_size+NFC_DMA_LEN.len_oob。
- 步骤4** 若为写操作，则软件需要准备好对应DDR中的数据。
- 步骤5** 配置器件地址寄存器NFC_CS0_PLN0_ADDRH、NFC_CS0_PLN0_ADDRL。
- 步骤6** 配置寄存器NFC_OP_CTRL为0x83（写操作）或者0x81（读操作），下发读操作。
- 步骤7** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。
- 步骤8** 若为写操作，则检测NFC_INTS_CS0.cs0_op_fail判断操作是否成功；若为读操作，则可以访问对应DDR中的数据。

----结束

13.14.13 Interleave 操作流程

Interleave模式下仅支持NFC_OP_CTRL的八种操作。在此模式下，任何一个cs没有命令执行时，都可以对其下发操作。

- 步骤1** 配置寄存器NFC_ARB_EN.interleave_mode使能，开启Interleave模式。
- 步骤2** 查询状态寄存器，检测到某片选处于ready状态，软件准备操作，对于写操作，需要确保操作的DDR地址已经存放好数据，对于读操作需要保证存放数据的DDR空间是否已经ready，对于擦除操作确认要操作的器件块是否需要擦除。
- 步骤3** 软件准备好以后，配置相关DDR基地址寄存器和器件操作寄存器，最后配置NFC_OP_CTRL下发操作，nfc_ready拉低。
- 步骤4** 继续查询其他片选的状态，如果处于ready状态，需要操作的话，继续下发操作，直到软件不需要操作或者没有片选处于ready状态。
- 步骤5** 判断NANDC的操作是否结束，请参考“[NANDC操作结束判断流程](#)”。对于已经结束的片选，可以发起新的操作。
- 步骤6** 如此循环，直到所有操作完成。

----结束

13.15 性能和功耗调优

无



13.16 注意事项

在使用NANDC的过程中，需要注意某些特殊配置或者场景可能会导致NANDC出现异常。

- NANDC的功能均是由寄存器下发，若控制器寄存器配置错误将导致NANDC工作异常。如配置寄存器NFC_OP为0x0，由于写此寄存器会触发操作，但没有开启任何使能，则可能造成NANDC出现异常。
- 寄存器中的某些位为保留位，若随意配置也可能导致NANDC工作不正常。
- NANDC不支持RB超时后查询器件状态次数配置功能，因此在初始化时，需要将寄存器NFC_TIMEOUT1_WR和寄存器NFC_TIMEOUT1_ERS都配置为0。若这两个寄存器的配置值不为0，则在出现RB超时后可能导致NANDC异常。

若NANDC出现挂死等异常情况，需要对NANDC进行复位处理。



14 eMMC/SD

14.1 特性

eMMC/SD host controller（以下简称eMMC）用于处理对device卡（SD卡/MMC）的读写等操作，属于自研IP。eMMC地址空间64位全地址访问，适用于SOC的最大物理地址大小；eMMC默认工作电压1.8V；SD低速模式3.3V，高速模式支持1.8V。

表 14-1 自研 eMMC/SD SPEED MODE

SPEED	Frequency(4line)	Signaling mode	Sampling
SDR400K	400kHz	1.8/3.3V	下沿发上沿采
DS	25MHz	3.3V	下沿发上沿采
HS	50MHz	3.3V	上沿发上沿采
SDR12	25MHz	1.8V/3.3V	下沿发上沿采
SDR25	50MHz	1.8V	上沿发上沿采
SDR50	100MHz	1.8V	上沿发上沿采
SDR104	200MHz	1.8V	上沿发上沿采
DDR50	50MHz	1.8V	CMD上沿发上沿采 DAT DDR 双沿
HS200	200MHz	1.8V/1.2V	Single data rate
HS400	400MHz	1.8V/1.2V	Dual data rate

自研eMMC IP支持符合以下协议的设备：（PG2应用场景仅需要支持eMMC颗粒）

- SecureDigitalMemory（SDmem-version3.0协议）



- SDIO（SDIO V3.00协议）
- MultiMediaCard（eMMC4.5/MMC4.41协议）
- Nano Memory Card（兼容eMMC4.5协议）

eMMC控制器具有以下特点：

- 支持SD协议的Default Speed（DS），High Speed（HS），SDR12，SDR25，SDR50和SDR104速率。
- 支持1/4bit模式；不支持8线模式（不允许配置8线模式；）。
- 支持eMMC协议的，协议的Default Speed，High Speed SDR，支持HS200模式，最高速率要求支持208MHz；最大读/写带宽均不小于104MB/s。
- 支持CPU数据传输，支持SDMA数据传输，
- 支持ADMA2数据，传输支持链表访问，链表模式最大支持4KB数据传输。
- Slave接口符合APB3协议（slvae空间4KB）
- Master接口符合AXI4协议（数据位宽支持32bit，地址支持64bit），写OST最大支持16，读OST最大支持17，支持乱序，不支持超过16的长Burst
只支持INC操作，支持非对齐地址，最小颗粒度为32bit（4Byte），不支持Narrow Burst，不支持Exclusive操作，不支持Lock操作
- 支持命令CRC7与数据CRC16校验，卡检测（需要GPIO控制双沿触发）与初始化。
- 支持Single/Multi_Block传输，块数据传输最大支持512Byte，支持最大Block Number为65536。
- 支持双array异步FIFO设计(FIFO_DEPTH=256,FIFO_WIDTH=32bit)，收发FIFO独立，支持水线配置。
- 支持FIFO上溢出与下溢出中断告警，防止数据传输错误。
- 支持命令完成信号与中断上报主机控制器。
- 支持Tuning功能，Sample和Driver 相位可配；调相窗口数量可编程（理论上16个窗口即可，即最大频率为1.6GHz，FNPLL=FSD*N/2）。
- 支持eMMC BOOT启动模式，不支持eMMC在DDR模式下进行Boot操作。
- 309a项目eMMC支持跨片访问；（PG2不支持）
配置peri_subctrl，0x00_A40C_220C=0x0表示访问本片，0x1表示跨片访问。

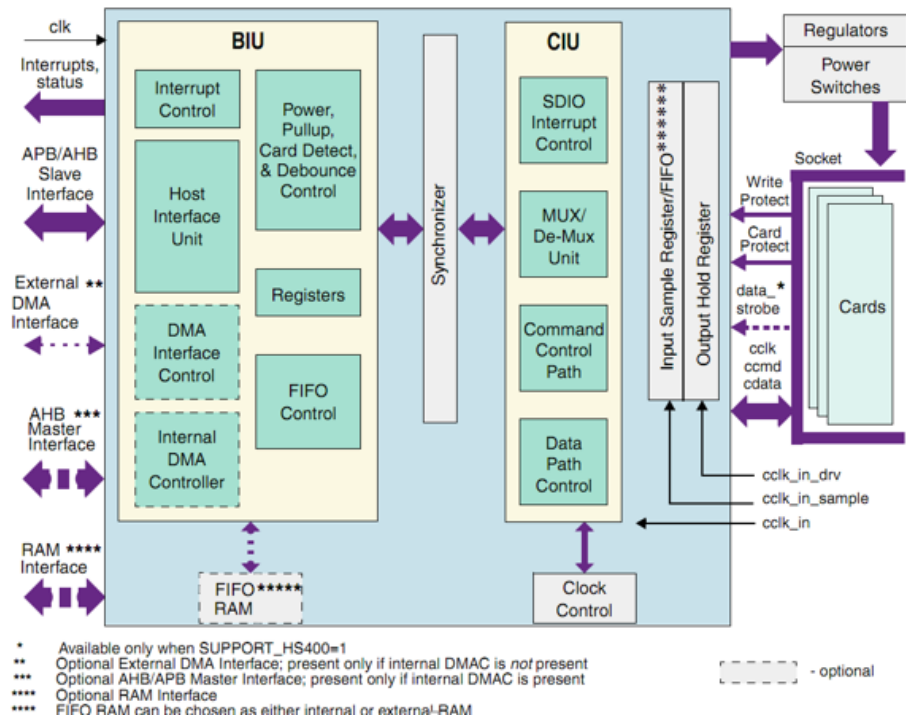
14.2 功能描述

14.2.1 整体结构



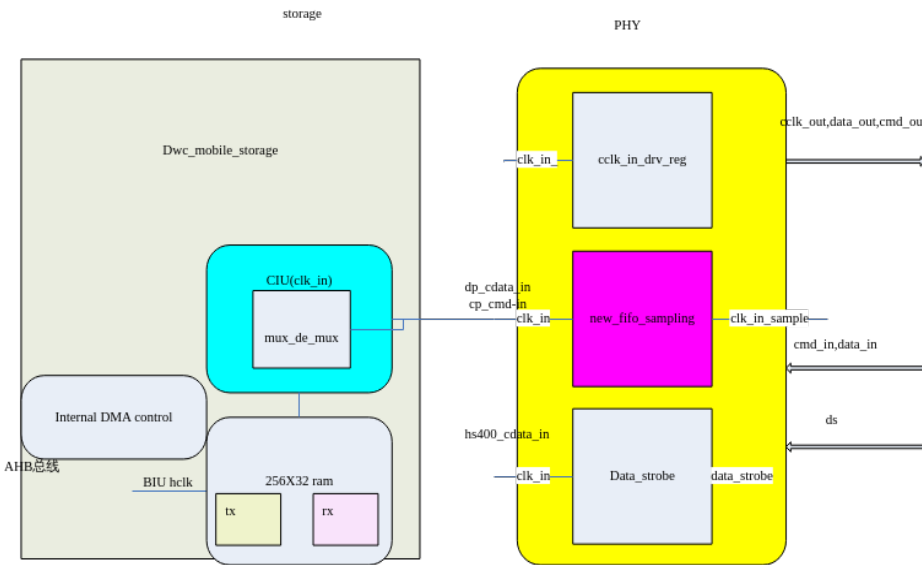
14.2.1.1 原有逻辑结构

图 14-1 Block Diagram



14.2.1.2 修改后的逻辑结构简图

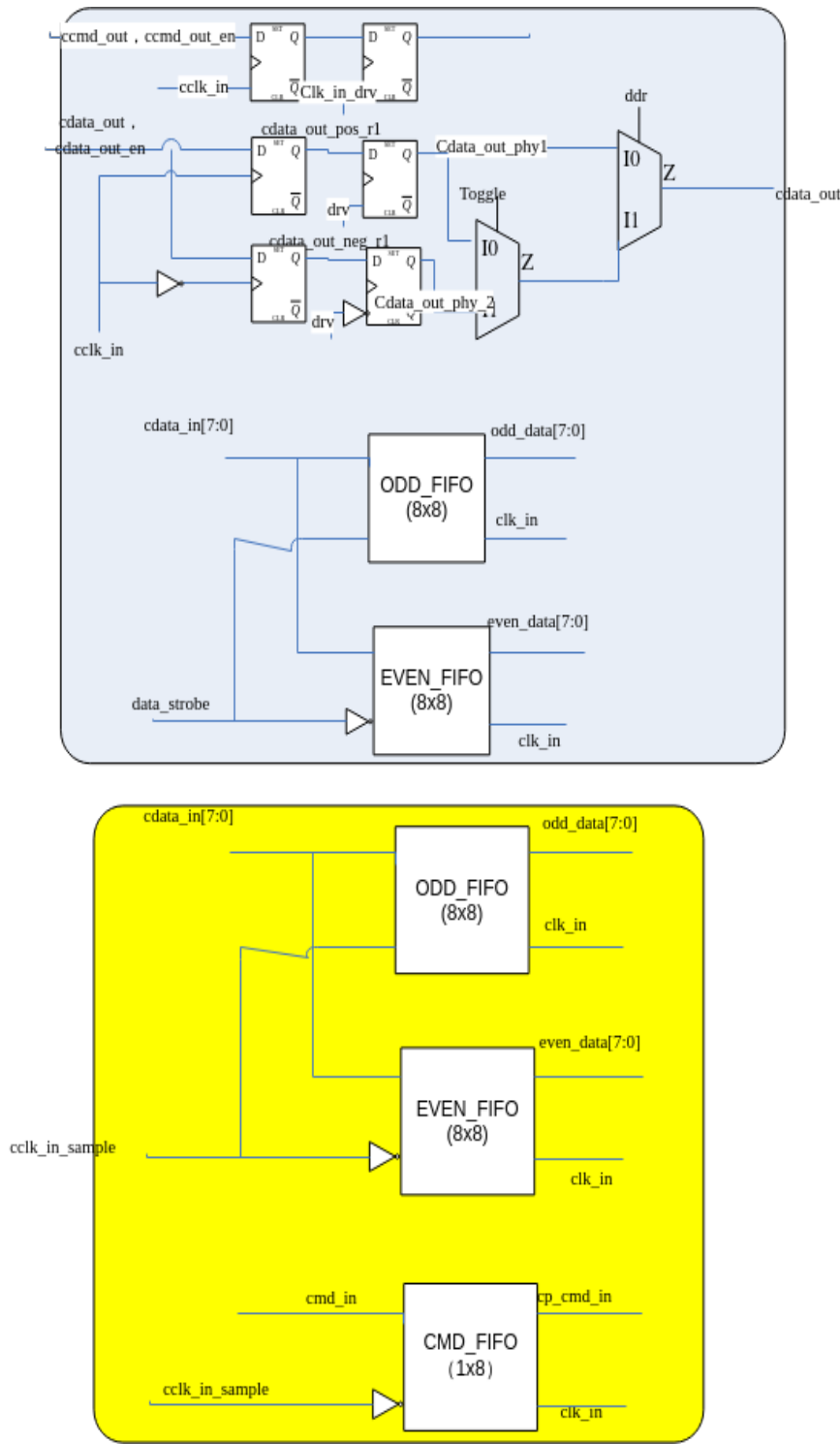
图 14-2 逻辑结构简图





14.2.2 PHY 模块内部结构

图 14-3 PHY 模块内部结构图





14.3 SD_EMMC 时序参数

表 14-2 SD_EMMC 时序参数

模式	支持的输入时钟 (MHz)	卡侧时钟	最大数据位 宽	电压
SDR104	200MHz	200MHz	4bit	1.8V
SDR50	100MHz	100MHz	4bit	1.8V
DDR50	50MHz	50MHz	4bit	1.8V
SDR25	50MHz	50MHz	4bit	1.8V
SDR12	25MHz	25MHz	4bit	1.8V
HS	50MHz	50MHz	4bit	3.3V
DS	25MHz	25MHz	4bit	3.3V
HS400	200MHz	200MHz	8bit	1.8V
HS200	200MHz	200MHz	8bit	1.8V
DDR-4bit	50MHz	50MHz	4bit	3.3V
DDR-8bit	100MHz	50MHz	8bit	3.3V
MMC HS	50MHz	50MHz	4bit	3.3V

14.3.1 MMC backward-compatible card interface timing

14.3.1.1 输入时序参数

图 14-4 输入时序参数

Table 85 — Backward-compatible card interface timing

Parameter	Symbol	Min	Max.	Unit	Remark ¹
Clock CLK²					
Clock frequency Data Transfer Mode (PP) ³	f _{PP}	0	26	MHz	C _L ≤ 30 pF
Clock frequency Identification Mode (OD)	f _{OD}	0	400	kHz	
Clock low time	t _{WL}	10		ns	C _L ≤ 30 pF
Clock rise time ⁴	t _{TLH}		10	ns	C _L ≤ 30 pF
Clock fall time	t _{THL}		10	ns	C _L ≤ 30 pF
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t _{ISU}	3		ns	C _L ≤ 30 pF
Input hold time	t _{IH}	3		ns	C _L ≤ 30 pF



14.3.1.2 输出时序参数

图 14-5 输出时序参数

Parameter	Symbol	Min	Max.	Unit	Remark ¹
Outputs CMD, DAT (referenced to CLK)					
Output set-up time	t _{OSU}	11.7		ns	C _L ≤ 30 pF
Output hold time	t _{OH}	8.3		ns	C _L ≤ 30 pF
NOTE 1. The card must always start with the backward-compatible interface timing. The timing mode can be switched to high-speed interface timing by the host sending the SWITCH command (CMD6) with the argument for high-speed interface select. NOTE 2. CLK timing is measured at 50% of VDD. NOTE 3. For compatibility with cards that support the v4.2 standard or earlier, host should not use > 20 MHz before switching to high-speed interface timing. NOTE 4. CLK rise and fall times are measured by min (V _{IH}) and max (V _{IL}).					

14.3.2 MMC HS (52M)

14.3.2.1 输入时序参数

图 14-6 输入时序参数

Table 84 — High-speed card interface timing

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK¹					
Clock frequency Data Transfer Mode (PP) ²	f _{PP}	0	52 ³	MHz	C _L ≤ 30 pF Tolerance: +100KHz
Clock frequency Identification Mode (OD)	f _{OD}	0	400	kHz	Tolerance: +20KHz
Clock low time	t _{WL}	6.5		ns	C _L ≤ 30 pF
Clock rise time ⁴	t _{TLH}		3	ns	C _L ≤ 30 pF
Clock fall time	t _{THL}		3	ns	C _L ≤ 30 pF
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t _{ISU}	3		ns	C _L ≤ 30 pF
Input hold time	t _{IH}	3		ns	C _L ≤ 30 pF

14.3.2.2 输出时序参数

图 14-7 输出时序参数

Outputs CMD, DAT (referenced to CLK)					
Output delay time during data transfer	t _{ODLY}		13.7	ns	C _L ≤ 30 pF
Output hold time	t _{OH}	2.5		ns	C _L ≤ 30 pF
Signal rise time ⁵	t _{RISE}		3	ns	C _L ≤ 30 pF
Signal fall time	t _{FALL}		3	ns	C _L ≤ 30 pF



14.3.3 MMC DDR-8BIT(52M)

14.3.3.1 示意图

图 14-8 时序图

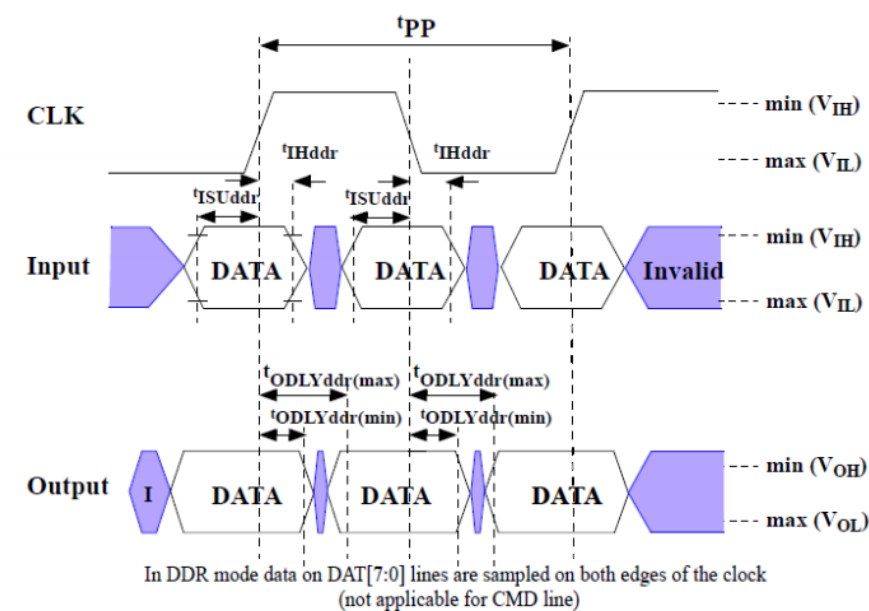


Figure 78 — Timing diagram: data input/output in dual data rate mode

14.3.3.2 输入输出时序，区分 CMD（单沿）和 DATA（双沿）

表 14-3 双数据速率接口时序

Parameter	Symbol	Min.	Max.	Unit	Remark
Input CLK ⁽¹⁾					
Clock duty cycle	-	45	55	%	Includes jitter, phase noise
Clock rise time	t_{TLH}	-	3	ns	$CL \leq 30pF$
Clock fall time	t_{THI}	-	3	ns	$CL \leq 30pF$
Input CMD (referenced to CLK-SDR mode)					



Parameter	Symbol	Min.	Max.	Unit	Remark
Input set-up time	tISUddr	3	-	ns	CL≤20pF
Input hold time	tIHddr	3	-	ns	CL≤20pF
Output CMD (referenced to CLK-SDR mode)					
Output delay time during data transfer	tODLY	-	13.7	ns	CL≤20pF
Output hold time	IOH	2.5	-	ns	CL≤20pF
Signal rise time	tRISE	-	3	ns	CL≤20pF
Signal fall time	tFALL	-	3	ns	CL≤20pF
Input DAT (referenced to CLK-DDR mode)					
Input set-up time	tISUddr	2.5	-	ns	CL≤20pF
Input hold time	tIHddr	2.5	-	ns	CL≤20pF
Output DAT (referenced to CLK-DDR mode)					
Output delay time during data transfer	tODLYddr	1.5	7	ns	CL≤20pF
Signal rise time (DAT0-7) (2)	tRISE	-	2	ns	CL≤20pF



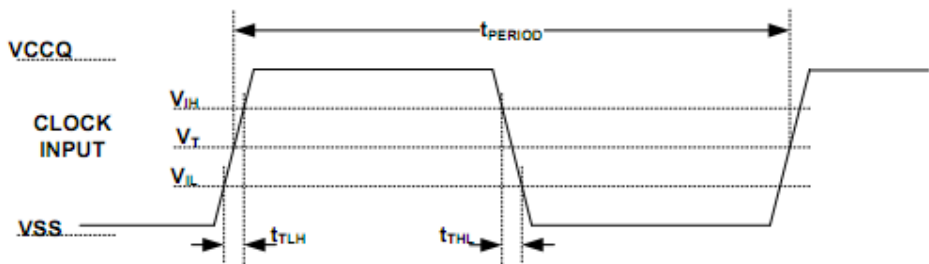
Parameter	Symbol	Min.	Max.	Unit	Remark
Signal fall time (DA T0-7)	tFALI	-	2	ns	CL≤20pF
NOTE 1: CLK timing is measured at 50% of VDD. NOTE 2: Inputs DAT rise and fall times are measured by min (V _{IH}) and max (V _{IL}), and outputs DAT rise and fall times are measured by min (V _{OH}) and max (V _{OL})					

14.3.4 MMC HS200

14.3.4.1 时钟要求

HS200 的最大频率为 200MHz。主机可以使用 HS200 模式允许的任意频率，直至最大频率。

图 14-9 HS200 时钟信号时序



说明

- 1. V_{IH}表示V_{IH}(min.)，V_{IL}表示V_{IL}(max.)。
- 2. V_T = 0.975V - Clock Threshold (V_{CCQ} = 1.8V)和V_T = 0.65V - Clock Threshold (V_{CCQ} = 1.2V)，指示定时测量的时钟参考点。

表 14-4 HS200 时钟信号时序

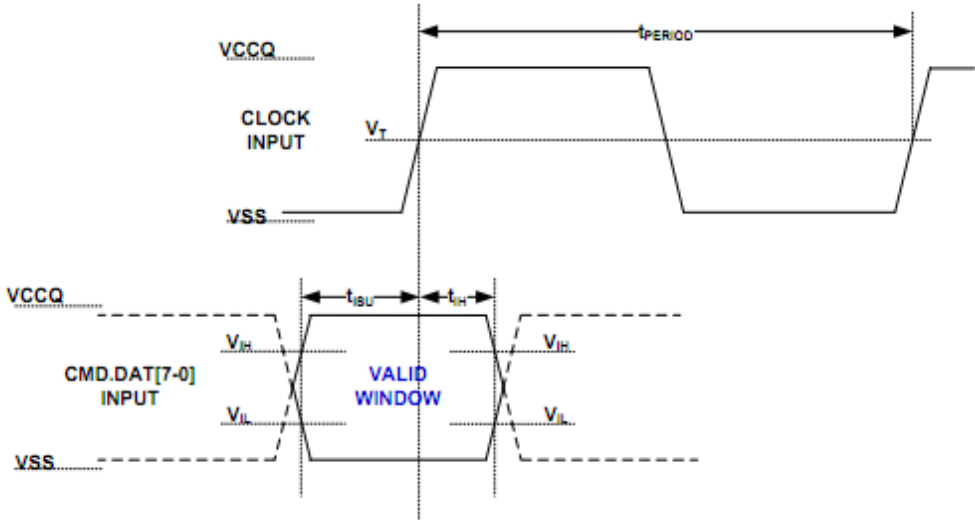
Symbol	Min.	Max.	Unit	Remark
t _{PERIOD}	5	-	ns	200MHz (Max), between rising edges



Symbol	Min.	Max.	Unit	Remark
t_{TLH}, t_{THL}	-	0.2- t_{PERIOD}	ns	$t_{TLH}, t_{THL} < 1\text{ns}$ (max.) at 200MHz, $C_{DEVICE}=6\text{pF}$, The absolute maximum value of t_{TLH}, t_{THL} is 10ns regardless of clock frequency.
Duty Cycle	30	70	%	-

14.3.4.2 输入时序要求

图 14-10 HS200 设备输入时序



说明

- 1. t_{ISU} 和 t_{IH} 分别是在 $V_{IL}(\text{max.})$ 和 $V_{IH}(\text{min.})$ 测量得到的。
- 2. V_{IH} 表示 $V_{IH}(\text{min.})$ ， V_{IL} 表示 $V_{IL}(\text{max.})$ 。

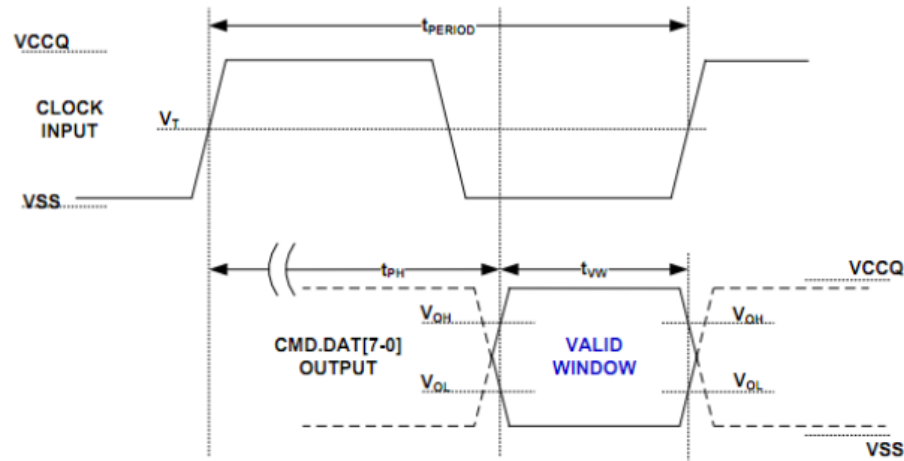
表 14-5 HS200 设备输入时序

Symbol	Min.	Max.	Unit	Remark
t_{ISU}	1.40	-	ns	$C_{DEVICE} \leq 6\text{pF}$
t_{IH}	0.8	-	ns	$C_{DEVICE} \leq 6\text{pF}$



14.3.5 输出时序

图 14-11 HS200 Device output timing



NOTE VoH denotes $V_{oH}(\text{min.})$ and V_{oL} denotes $V_{oL}(\text{max.})$.

表 14-6 191—Output timing

Symb ol	Min.	Max.	Unit	Remark
t_{pH}	0	2	UI	Device output momentary phase from CLK input to CMD or DAT lines output. Does not include a long term temperature drift.
ΔTPH	-350 ($\Delta T = -20^{\circ}\text{C}$)	+1550 ($\Delta T = -90^{\circ}\text{C}$)	PS	Delay variation due to temperature change after tuning. Total allowable shift of output valid window (T_{VW}) from last system Tuning procedure ΔTPH is 2600ps for ΔT from -25°C to 125°C during operation。
t_{VW}	0.575	-	UI	$t_{VW} = 2.88\text{ns}$ at 200MHz Host path may add Signal Integrity induced noise, skews, etc. Expected T_{VW} at Host input is larger than 0.475UI



14.3.6 MMC HS400

14.3.6.1 输入方向

图 14-12 HS400 Device Data input timing

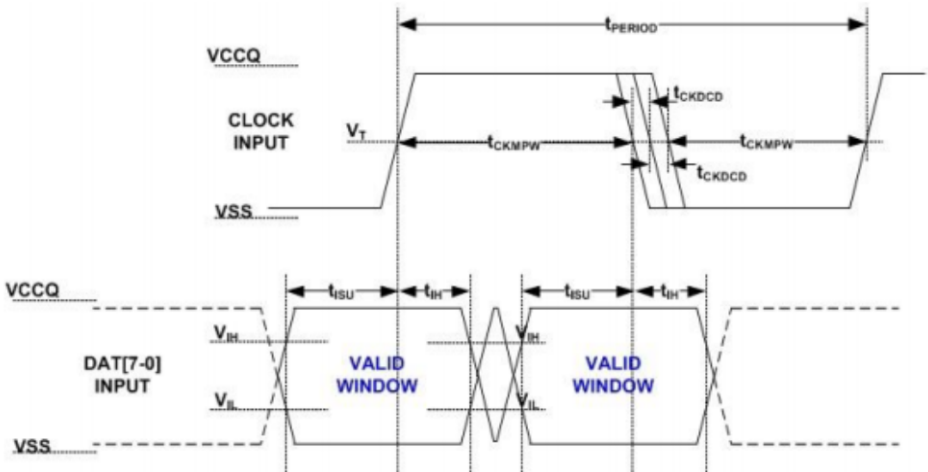


表 14-7 HS400 Device input timing

Paramet er	Symb ol	Min.	Max.	Unit	Remark
Input CLK					
Cycle time data transfer mode	t_{PERIOD}	5	-	-	200MHz(Max), between rising edges With respect to V_T .
Slew rate	SR	1.125	-	V/ns	With respect to V_{IH}/V_{IL} .
Duty cycle distortio n	t_{CKDCD}	0.0	0.3	ns	Allowable deviation from an ideal 50% duty cycle. With respect to V_T . Includes jitter, phase noise
Minimu m pulse width	t_{CKMPW}	2.2	-	ns	With respect to V_T .
Input DAT (referenced to CLK)					



Parameter	Symbol	Min.	Max.	Unit	Remark
Input set-up time	tISUddr	0.4	-	ns	$C_{Device} \leq 6pF$ With respect to V_{IH}/V_{IL} .
Input hold time	tIHddr	0.4	-	ns	$C_{Device} \leq 6pF$ With respect to V_{IH}/V_{IL} .
Slew rate	SR	1.125	-	V/ns	With respect to V_{IH}/V_{IL} .

14.3.7 输出方向

图 14-13 HS400 Device output timing

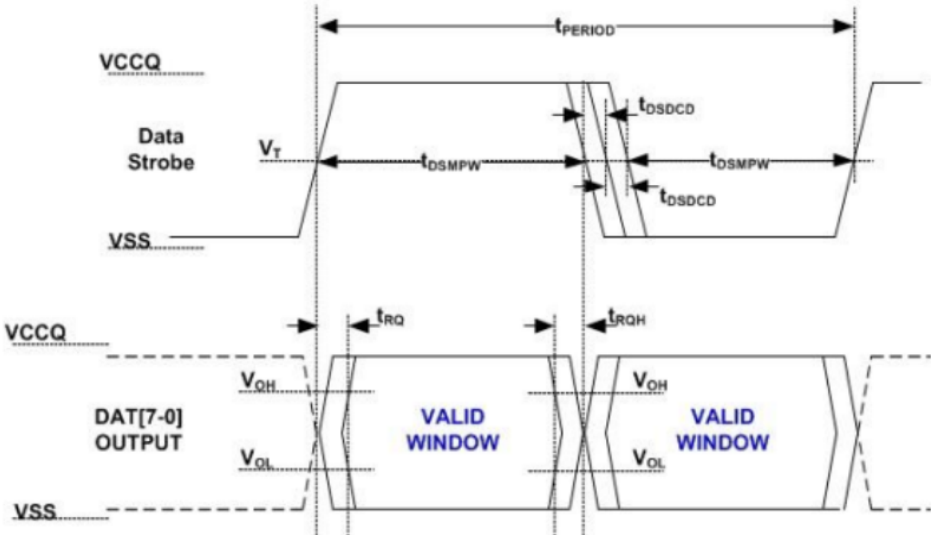


表 14-8 HS400 Device Output timing

Parameter	Symbol	Min.	Max.	Unit	Remark
Data Strobe					
Cycle time data transfer mode	t_{PERIOD}	5	-	-	200MHz(Max), between rising edges With respect to V_T



Parameter	Symbol	Min.	Max.	Unit	Remark
Slew rate	SR	1.125	-	V/ns	With respect to V_{OH}/V_{OL} and HS400 reference load
Duty cycle distortion	tDSD CD	0.0	0.2	ns	Allowable deviation from the input CLK duty cycle distortion (tCKDCD) With respect to V_T Includes jitter, phase noise
Minimum pulse width	tDS MPW	2.0	-	ns	With respect to V_T
Read pre-amble	t _{RPRE}	0.4	-	t _{PERIOD}	Max value is specified by manufacturer. Value up to infinite is valid
Read post-amble	t _{RPST}	0.4	-	t _{PERIOD}	Max value is specified by manufacturer. Value up to infinite is valid
Output DAT (referenced to Data Strobe)					
Output skew	tRQ	-	0.4	ns	With respect to V_{OH}/V_{OL} and HS400 reference load
Output hold skew	tRQH	-	0.4	ns	With respect to V_{OH}/V_{OL} and HS400 reference load
Slew rate	SR	1.125	-	V/ns	With respect to V_{OH}/V_{OL} and HS400 reference load

表 14-9 HS400 Capacitance

Parameter	Symbol	Min.	Type	Max	Unit	Remark
Pull-up resistance for CMD	R _{CM D}	4.7	-	100 ⁽¹⁾	Kohm	-



Parameter	Sym bol	Min .	Type	Max	Unit	Remark
Pull-up resistance for DAT0-7	R _{DAT}	10	-	100 ⁽¹⁾	Kohm	-
Pull-down resistance for Data Strobe	R _{DS}	10	-	100 ⁽¹⁾	Kohm	-
Internal pull up resistance DAT1-DAT7	R _{int}	10	-	150	Kohm	-
Bus signal line capacitance	CL	-	-	13	pF	-
Single Device capacitance	C _{Devi ce}	-	-	6	pF	-

14.3.8 SD 卡 HS (50M)

14.3.8.1 时序要求

Parameter	Sym bol	Mi n.	Max.	Unit	Remark
Clock CLK (All values are referred to min (V _{IH}) and max (V _{IL}))					
Clock frequency Data Transfer Mode	f _{PP}	0	50	MHz	C _{CARD} ≤ 10 pF (1 card)
Clock low time	t _{WL}	7	-	ns	C _{CARD} ≤ 10 pF (1 card)
Clock high time	t _{WH}	7	-	ns	C _{CARD} ≤ 10 pF (1 card)
Clock rise time	t _{TLH}	-	3	ns	C _{CARD} ≤ 10 pF (1 card)
Clock fall time	t _{THL}	-	3	ns	C _{CARD} ≤ 10 pF (1 card)
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t _{ISU}	6	-	ns	C _{CARD} ≤ 10 pF (1 card)
Input hold time	t _{IH}	2	-	ns	C _{CARD} ≤ 10 pF (1 card)



Parameter	Sym bol	Mi n.	Max.	Unit	Remark
Outputs CMD, DAT (referenced to CLK)					
Output Delay time during Data Transfer Mode	t_{ODLY}	-	14	ns	$C_L \leq 40$ pF (1 card)
Output Hold time	t_{OH}	2.5	-	ns	$C_L \leq 15$ pF (1 card)
Total System capacitance for each line ¹	C_L	-	40	pF	1 card

14.3.9 SD 卡 SDR12,SDR25, SDR50, SDR104

14.3.9.1 时钟要求

图14-14 shows clock signal timing and 表14-10 shows required values of this timing. Clock timing is requirement for the host. t_{CLK} is used to define rise / fall timing. Rise and fall time shall be less than $0.2 * t_{CLK}$. SDCLK input shall satisfy the clock timing over all variable conditions, and is measured as close as possible to SD socket pins to the card while CMD and DAT[3:0] are in quiet state (not toggling). V_{IH} denote $V_{IH}(\min.)$ and V_{IL} denotes $V_{IL}(\max.)$ in 图14-14.

图 14-14 Clock Signal Timing

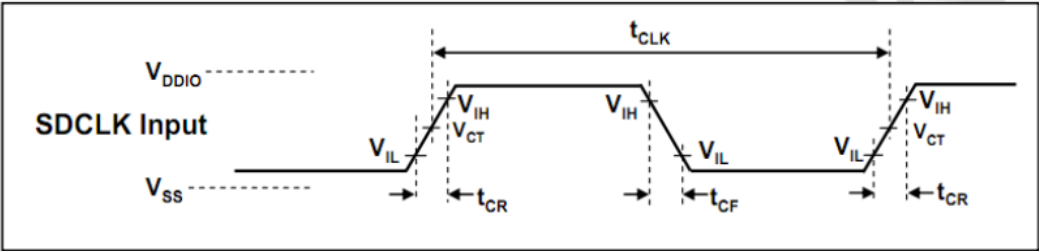


表 14-10 Clock Signal Timing

Symbol	Min.	Max.	Unit	Remark
t_{CLK}	4.80	-	ns	208MHz (Max.), Between rising edge, $V_{CT}= 0.975V$
t_{CR}, t_{CF}	-	$0.2 * t_{CLK}$	ns	$t_{CR}, t_{CF} < 0.96ns$ (max.) at 208MHZ, $C_{CARD}=10pF$ $t_{CR}, t_{CF} < 2.00ns$ (max.) at 100MHZ, $C_{CARD}=10pF$



Symbol	Min.	Max.	Unit	Remark
Clock Duty	30	70	%	-

Table 6-13 shows DC specification of 1.8V signaling. As signaling level is generated by regulator in host and card, some of the values are defined by fixed value rather than based on VDD.

Parameter	Symbol	Min.	Max.	Unit	Remark
Supply Voltage	V _{DD}	2.70	3.60	V	
Regulator Voltage	V _{DDIO}	1.70	1.95	V	Generated by V _{DD}
Output High Voltage	V _{OH}	1.40	-	V	I _{OH} = -2mA
Output Low Voltage	V _{OL}	-	0.45	V	I _{OL} = 2mA
Input High Voltage	V _{IH}	1.27	2.00	V	
Input Low Voltage	V _{IL}	V _{SS} -0.30	0.58V	V	

Table 6-13 : Threshold Level for 1.8V Signaling

14.3.9.2 输入时序

图14-15 shows card input timing and 表14-11 ~表14-12 shows required values of card input timing. The new parameter Clock Threshold (V_{CT}) is introduced to indicate clock reference point. V_{CT} is defined as 0.975V. Data setup time and hold time are measured at Data Threshold (V_{IL}(max.) and V_{IH}(min.)). V_{IH} denote V_{IH}(min.) and V_{IL} denotes V_{IL}(max.) in 图14-15.

图 14-15 Card Input Timing

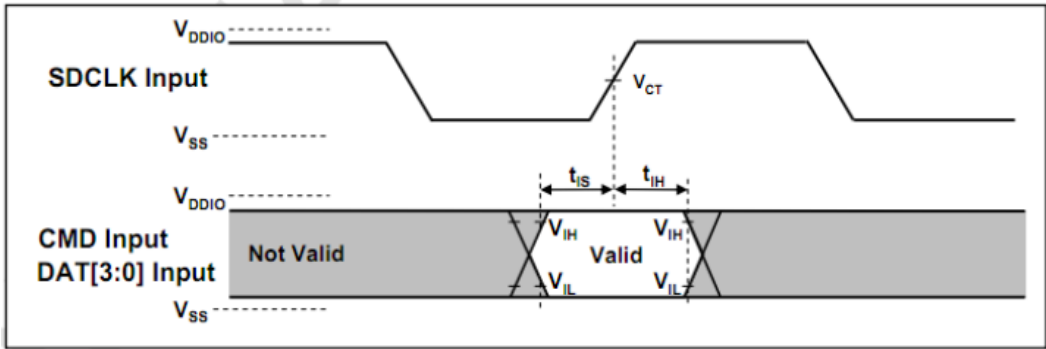


表 14-11 SDR104 Input Timing

Symbol	Min.	Max	Unit	SDR104 mode
t _{IS}	1.40	-	ns	C _{CARD} = 10pF, V _{CT} = 0.975V
t _{IH}	0.80	-	ns	C _{CARD} = 5pF, V _{CT} = 0.975V



表 14-12 SDR50 Input Timing

Symbol	Min.	Max	Unit	SDR50 mode
t_{IS}	3.00	-	ns	$C_{CARD} = 10\text{pF}$, $V_{CT} = 0.975\text{V}$
t_{IH}	0.80	-	ns	$C_{CARD} = 5\text{pF}$, $V_{CT} = 0.975\text{V}$

14.3.9.3 输出时序 SDR12 SDR25 SDR50

图 14-16 Output Timing of Fixed Data Window

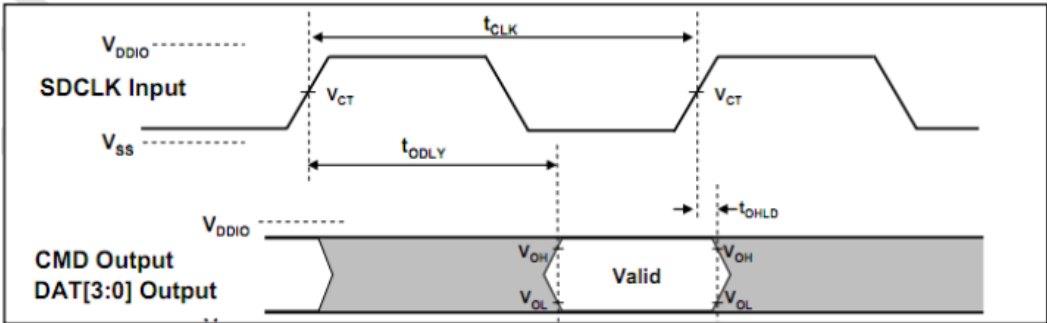


表 14-13 Output Timing of Fixed Data Window

Symbol	Min.	Max	Unit	Remark
t_{ODLY}	-	7.5	ns	$t_{CLK} \geq 10.0\text{ns}$, $CL = 30\text{pF}$, using driver Type B, for SDR50.
t_{ODLY}	-	14	ns	$t_{CLK} \geq 20.0\text{ns}$, $CL = 40\text{pF}$, using driver Type B, for SDR25 and SDR12.
t_{OH}	1.5	-	ns	Hold time at the t_{ODLY} (min.), $C_L = 15\text{pF}$

14.3.9.4 输出时序 SDR104 最高可支持 208M

图 14-17 Output Timing of Variable Data Window

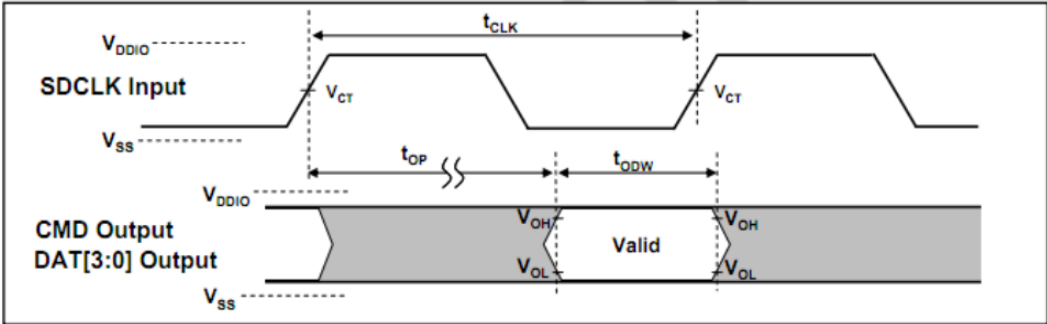




表 14-14 Output Timing of Variable Data Window

Symbol	Min.	Max	Unit	Remark
t _{OP}	0	2	UI	Card Output Phase
Δt _{OP}	-350	+1550	ps	Delay variation due to temperature change after tuning
t _{ODW}	0.60	-	UI	t _{ODW} =2.88ns at 208MHz

14.3.10 SD 卡 DDR50-4bit

14.3.10.1 时钟要求

图 14-18 Clock Signal Timing

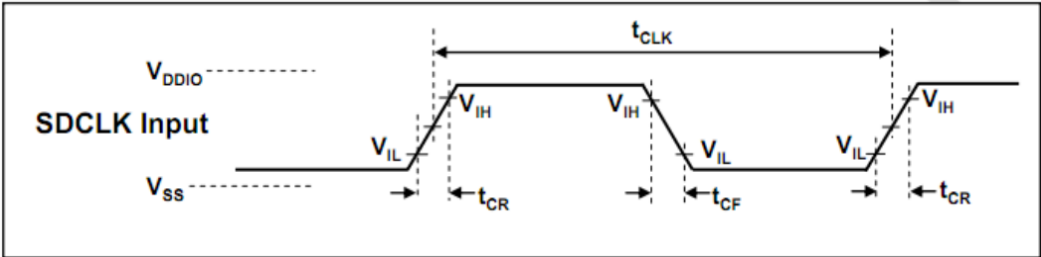
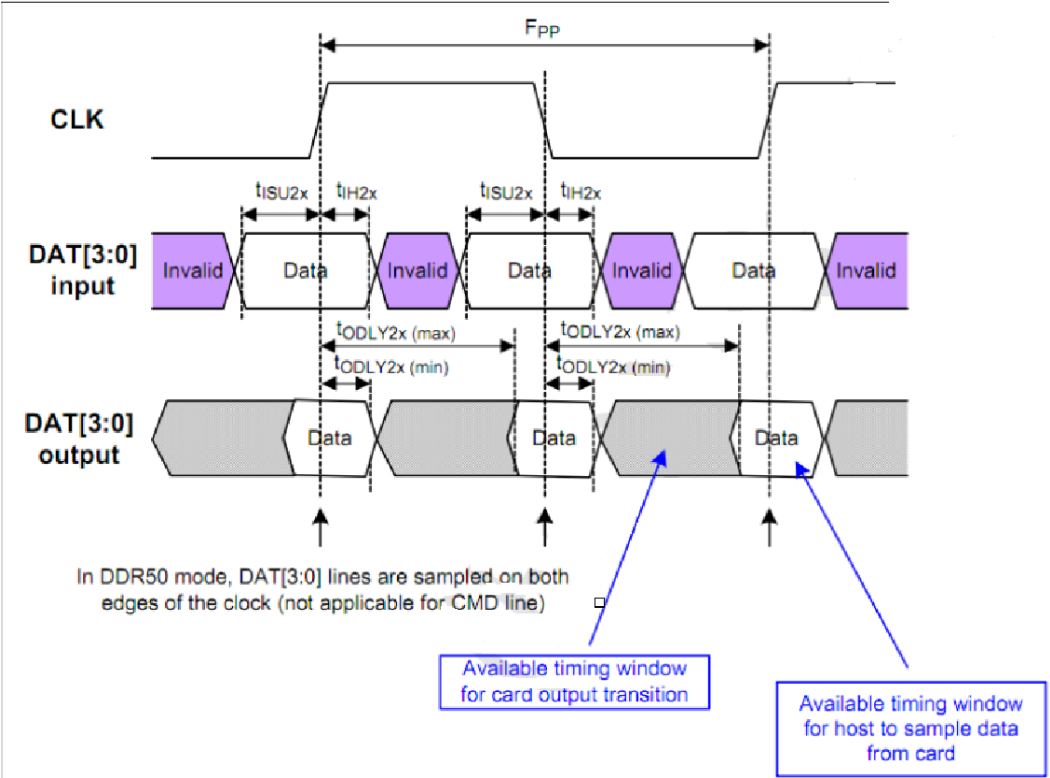


表 14-15 Clock Signal Timing

Symbol	Min.	Max	Unit	Remark
t _{CLK}	20	-	ns	50MHz (Max.). Between rising edge
t _{CR} ,t _{CF}	-	0.2*t _{CLK}	ns	t _{CK} ,t _{CF} < 4.00ns (max.) at 50MHz, C _{CARD} =10pF
Clock Duty	45	55	%	-



图 14-19 Timing Diagram DAT Inputs/Outputs Referenced to CLK in DDR50 Mode



14.3.10.2 输入输出的时序要求，区分 CMD 和 DATA

表 14-16 Bus Timings - Parameters Values (DDR50 mode)

Parameter	Sym bol	Min	Max	Unit	Remark
Input CMD (referenced to CLK rising edge)					
Input set-up time	t_{ISU}	6	-	ns	$C_{CARD} \leq 10pF$ (1 card)
Input hold time	t_{IH}	0.8	-	ns	$C_{CARD} \leq 10pF$ (1 card)
Output CMD (referenced to CLK rising edge)					
Output Delay time duringData Transfer Mode	t_{ODLY}	-	13.7	ns	$C_L \leq 30pF$ (1 card)
Output hold time	t_{OH}	1.5	-	ns	$C_L \geq 15pF$ (1 card)
Inputs DAT (referenced to CLK rising and falling edges)					



Parameter	Sym bol	Min	Max	Unit	Remark
Input set-up time	t _{ISU2x}	3	-	ns	C _{CARD} ≤10pF (1 card)
Input hold time	t _{IH2x}	0.8	-	ns	C _{CARD} ≤10pF (1 card)
Outputs DAT (referenced to CLK rising and falling edges)					
Output Delay time during Data Transfer Mode	t _{ODLY} 2x	-	7.0	ns	C _L ≤25pF (1 card)
Output hold time	t _{ODLY} 2x	1.5	-	ns	C _L ≥15pF (1 card)

14.4 时钟定义

14.4.1 SDIOV2P9 内部支持 SDIO/SD/eMMC 三种协议

eMMC:支持DS/HS/HS DDR-4BIT/HS DDR-8BIT/HS200/HS400

SD:支持DS/HS/SDR12/SDR25/SDR50/DDR50/SDR104

SDIO:支持DS/HS/SDR12/SDR25/SDR50/DDR50/SDR104

14.4.2 各模式下的时序要求

表 14-17 各模式下的时序要求

模式	支持的输入时钟 (MHz)	卡侧时钟	最大数据位 宽	电压
SDR104	200MHz	200MHz	4bit	1.8V
SDR50	100MHz	100MHz	4bit	1.8V
DDR50	50MHz	50MHz	4bit	1.8V
SDR25	50MHz	50MHz	4bit	1.8V
SDR12	25MHz	25MHz	4bit	1.8V
HS	50MHz	50MHz	4bit	3.3V
DS	25MHz	25MHz	4bit	3.3V
HS400	200MHz	200MHz	8bit	1.8V
HS200	200MHz	200MHz	8bit	1.8V
DDR-4bit	50MHz	50MHz	4bit	3.3V



模式	支持的输入时钟 (MHz)	卡侧时钟	最大数据位 宽	电压
DDR-8bit	100MHz	50MHz	8bit	3.3V
MMC HS	50MHz	50MHz	4bit	3.3V

14.5 典型问题定位流程

- EMMC DMA方式搬运数据失败，没有进行DMA传输，且没有收到完成中断；通过定位波形，EMMC未获取DDR给出正确的描述符，在device操作下，不符合SMMU 访问device空间的保序读操作，访问挂死；
- 解决方案：在EMMC初始化之前，配置user域属性，将axcache修改为normal属性，即axcache=0xE或0xF即可。

14.6 软件约束

1. Hi309a应用场景是智能小站，解决方案对芯片侧暂无SMMU地址转换的需求，所以，PG2替换成自研eMMC，继续保持PG1的访问模式，不支持SMMU地址转换。
2. eMMC替换成自研，axsize固定为0x010，axlen支持0~15，保序读写，由于SMMU device保序读操作仅支持以下操作类型（读axsize=0x010，axlen=0），为了满足SMMU操作要求，约束软件通过cacheable访问内存，将axcache配置为0xF。
3. 异常情况下解复位流程要求：
系统默认情况下eMMC开时钟解复位，当处于异常情况下时，解复位流程见下：
复位→关时钟→解复位→开时钟
4. eMMC master限制访问device空间，受限SMMU仅支持固定的结构访问，要求不能随便配置axache，地址0x0x00_A458_504C默认值为0x00000F0F。（与第二点一致）
5. eMMC IP内部（4K空间）支持pstrp访问（支持BYTE访问），eMMC 外部的空间（4K~64K）仅支持WORD访问。
6. 在启动切换到高频模式下，需要先关闭eMMC外部时钟门控，地址0x0x00_A458_5004默认值为0x00000001，确保时钟无毛刺切换。
7. Dfx功能使能：0x5050寄存器bit[3]为dfx_en，用于控制dfx功能是否打开，安全配置，控制0x24寄存器地址的20bit~24bit，以及0x800寄存器以后的DFX寄存器。请根据产品安全策略是否打开DFX调试寄存器。