



PCIE 级联

应用指南

文档版本 01

发布日期 2018-12-10

版权所有 © 深圳市海思半导体有限公司 2018。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深圳市海思半导体有限公司的商标。

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。

深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：support@hisilicon.com



前 言

概述

本文分别从硬件环境准备、软件环境准备等方面介绍 Demo 板 PCIE 级联操作的相关指导，同时介绍了 PCIE 的基础知识、PCIE 级联的业务实现和 PCIE MPI 接口函数等，可为用户在使用 PCIE 级联功能时提供参考。



说明

- 未有特殊说明，Hi3559CV100 与 Hi3559AV100 内容一致。
- 未有特殊说明，下文中的 Hi35xxVxxx 表示的芯片包含 Hi3559AV100/Hi3559CV100/Hi3519AV100。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100

读者对象





本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 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	表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
 警告	表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 注意	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	表示能帮助您解决某个问题或节省您的时间。
 说明	表示是正文的附加信息，是对正文的强调和补充。

修订记录

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

文档版本 01 (2018-12-10)

第 1 次正式版本发布。

5.3 小节，PCIV_MAX_CHN_NUM【定义】涉及修改

文档版本 00B06 (2018-07-10)

第 6 次临时版本发布。

添加 Hi3519AV100 的相关内容

文档版本 00B05 (2018-06-20)

第 5 次临时版本发布。

5.2 小节，HI_MPI_PCIV_Create【注意】涉及修改

5.5.2 小节涉及修改

文档版本 00B04 (2018-05-15)

第 4 次临时版本发布。

1.3、1.4.1 和 3.2 小节涉及修改

5.2 小节，HI_MPI_PCIV_MallocChnBuffer【注意】涉及修改



文档版本 00B03 (2018-04-04)

第 3 次临时版本发布。

5.2 小节，HI_MPI_PCIV_MallocChnBuffer、HI_MPI_PCIV_FreeChnBuffer 和 HI_MPI_PCIV_GetBaseWindow 【参数】涉及修改

5.3 小节，新增 PCIV_MAX_DMABLK；PCIV_PIC_ATTR_S 【定义】和 【成员】涉及修改

5.5.1 小节涉及修改

文档版本 00B02 (2018-03-15)

第 2 次临时版本发布。

5.3 小节，新增 [PCIV_MAX_CHN_NUM](#)、[PCIV_MAX_CHIPNUM](#) 和 PCIV_MAX_VBCOUNT；所有数据类型添加 【注意事项】

PCIV_PIC_ATTR_S 【成员】涉及修改

文档版本 00B01 (2018-01-30)

第 1 次临时版本发布。



目 录

前 言.....	i
1 Demo 板 PCIE 级联操作指南	1
1.1 硬件环境准备	1
1.2 软件环境准备	1
1.3 主片及从片启动	2
1.4 PCIV 功能验证	3
1.4.1 依赖的驱动说明	3
1.4.2 验证操作步骤	3
2 Demo 板从启动硬件及软件配置	4
2.1 从启动硬件配置	4
2.2 在级联场景时钟设计注意事项	4
2.3 从启动软件配置	6
2.3.1 PCIE 时钟选择模式配置	6
2.3.2 PCIE_MCC 驱动配置	6
3 PCIE 基础知识	7
3.1 概述	7
3.2 PCIE DMA 方式数据传输	8
3.3 PCIE 共享内存方式数据传输	8
3.4 PCIE MCC 消息应用	10
4 PCIE 级联业务实现	11
4.1 视频预览	11
4.2 码流传输	11
4.3 解码回放图像显示	12
4.4 内存配置	13
5 PCIV 开发参考	14
5.1 PCIV 概述	14
5.2 PCIV MPI 参考	14
5.3 数据类型	31
5.4 错误码	42



5.5 Proc 调试信息	42
5.5.1 PCIV	42
5.5.2 PCIV FMW	44



插图目录

图 2-1 PCIe 主片场景时钟图.....	4
图 2-2 PCIe 从片场景时钟图.....	5
图 2-3 PCIe 模式图	5
图 2-4 两片 PCIe X1 级联示意图.....	5
图 2-5 两片 PCIe X2 级联示意图.....	6
图 3-1 PCIe 地址与 AHB 侧地址的映射关系图.....	9
图 3-2 PCIe 地址与 AHB 侧地址的映射关系图.....	9
图 4-1 视频预览数据流处理流程.....	11
图 4-2 码流传输数据流处理流程.....	12
图 4-3 解码回放图像数据流处理流程.....	12
图 4-4 PCIe 窗口 PF 区域和 MMZ 区域的示意	13



表格目录

表 1-1 启动文件清单（主从片均采用 Flash 启动）	1
表 1-2 启动文件清单（主片采用 Flash 启动，从片采用 DDR 启动）	2
表 2-1 PCIe 时钟源选择	4
表 3-1 PCIe 地址空间和命令	7



1 Demo 板 PCIE 级联操作指南

1.1 硬件环境准备

- PCIE 级联调测，需要准备两块或多块硬件板卡，两片或多片 Hi35xxVxxx 通过 PCIE 级联使用：
 - 其中一片工作在 PCIE 主模式（主片 RC 模式）。
 - 其他片工作在 PCIE 从模式（从片 EP 模式）。多片通过 PCIE 进行级联时，主片通过 PCIE 桥与多个从片进行级联。需要为板卡正确连接电源线、串口线、网线以及视频输入输出线。
- 目前 Hi35xxVxxx 级联均支持两种启动方式：
 - 主片与从片均从 Flash 启动。
 - 工作在 PCIE 主模式的 Hi35xxVxxx 从 Flash 启动，工作在 PCIE 从模式的 Hi35xxVxxx 需要将启动模式更改为 DDR 启动模式（由主片来引导从片启动）。

1.2 软件环境准备

芯片需要的 boot、内核以及文件系统可在发布包的 pub/images 目录下获得这些镜像文件。

- 主从片均采用 Flash 启动方式：
按照《Hi35xxVxxx SDK 安装以及升级使用说明》烧写主/从片上的 boot、内核以及文件系统；主/从片成功启动后，配置网络及 NFS。
- 主从片 Flash 启动方式下，启动文件清单如表 1-1 所示。

表1-1 启动文件清单（主从片均采用 Flash 启动）

项目		文件名称	描述
主片	ARM	hi35xxvxxx.bin	烧写到主片 Flash
		uImage_hi35xxvxxx	烧写到主片 Flash



项目		文件名称	描述
		rootfs_hi35xxvxxx_2k_24bit.yaffs2	烧写到主片 Flash
从片	ARM	u-boot-hi35xxvxxx.bin	烧写到从片 Flash
		uImage_hi35xxvxxx_slave	烧写到从片 Flash
		rootfs_hi35xxvxxx_2k_24bit.yaffs2	烧写到从片 Flash

如果主片采用 Flash 启动方式，从片采用由主片进行引导 DDR 启动方式。该方式下，启动文件清单如表 1-2 所示。

表1-2 启动文件清单（主片采用 Flash 启动，从片采用 DDR 启动）

项目		文件名称	描述
主片	ARM	u-boot-hi35xxvxxx	烧写到主片 Flash
		u-boot-hi35xxvxxx	烧写到主片 Flash
		rootfs_hi35xxvxxx_2k_24bit.yaffs2	烧写到主片 Flash
从片	ARM	u-boot.bin	由主片导入从片 DDR 内存
		uImage	由主片导入从片 DDR 内存
		cramfs.initrd.img	由主片导入从片 DDR 内存



说明

上文中的 hi35xxvxxx 并不代表具体的文件名称，真正的文件名称以该芯片版本发布包中为准。

1.3 主片及从片启动

主片和从片启动有以下 2 种方式：

- 如果主从片均采用 Flash 启动方式。
 - 启动主片和从片的步骤请参考文档《Hi35xxVxxx SDK 安装以及升级使用说明》
- 如果主片采用 Flash 启动方式，从片采用 DDR 启动方式，由主片来引导。
 - 详细操作请参考其发布包中 hi35xxvxxx sdk/osdrv/components/pcie_mcc/pcie_mcc/目录下的《主片引导从片的启动方法》

但是当其主片为 Hi35xxVxxx 时候，需要在主片加载相应驱动并执行主片相应的可执行程序。

注意：一旦程序出现异常需要重启单板，则主从片均要重启，重新搭建级联环境。



1.4 PCIV 功能验证

1.4.1 依赖的驱动说明

PCIV 为基于 PCIE 驱动和 Hi35xxVxxx 媒体驱动的上层业务模块，依赖的驱动包括 PCIE 底层驱动、PCIE 消息驱动（即 MCC）和所有 MPP 媒体驱动。

- PCIE 底层驱动包括 hi35xx_dev_host.ko、hi35xx_dev_slv.ko，pcit_dma_host.ko 和 pcit_dma_slv.ko 以及 irq_map_host.ko 和 irq_map_slv.ko；
- PCIE 消息驱动包括 mcc_drv_host.ko，mcc_usrdev_host.ko，mcc_drv_slv.ko，mcc_usrdev_slv.ko；
- MPP 媒体驱动：load35xxvxxx 脚本中所有需要加载的模块驱动，主从片分别运行不同的 ko 加载脚本；
- PCIV 模块本身的相关驱动：
仅支持单系统模式，加载 hi35xxvxxx_pciv.ko，hi35xxvxxx_pciv_fm.w.ko

SDK 发布包中已有相关脚本来加载以上所有驱动。

1.4.2 验证操作步骤

验证 PCIV 功能的样例代码位于 SDK 发布包的 mpp/sample/pciv/目录下，包括主片和从片分别使用的 Sample 代码、封装的消息通讯代码和封装的数据通讯代码（具体说明请参考 Sample 目录下的 readme 文档）。



注意

执行以下验证操作前，请先使用 Sample 目录下的 Makefile 编译生成相关可执行文件。

在硬件、软件环境准备好后，执行完整的 PCIV 功能验证（其中的脚本内容请直接阅读 SDK 发布包中相应脚本）的步骤如下：

- 步骤 1. 主从片分别进入 mpp /ko 目录，执行脚本 load35xxvxxx，加载 PCIE 相关 ko 和 MPP 相关 ko。
- 步骤 2. 从片进入 mpp/sample/pciv/目录，执行从片的 Sample 程序 sample_pciv_slave，此时终端上将打印 start check pci target id:0……并被阻塞，等待主片启动。
- 步骤 3. 主片进入 mpp/sample/pciv/目录，执行主片的 Sample 程序 sample_pciv_host，完成与从片的握手过程后，即启动从片与主片之间的数据传输业务。
- 步骤 4. 如果需要停止 Sample 程序，可以在主片上按回车键，主片会发消息给从片销毁相关业务，并退出本片程序；从片接收消息并在销毁相关业务后退出。

----结束



2 Demo 板从启动硬件及软件配置

2.1 从启动硬件配置

BOOT 源选择为 PCIe 启动，要求将 BOOT_SEL4，配置为 1。

PCIe 从启动时，注意时钟硬件选择管脚的设置：

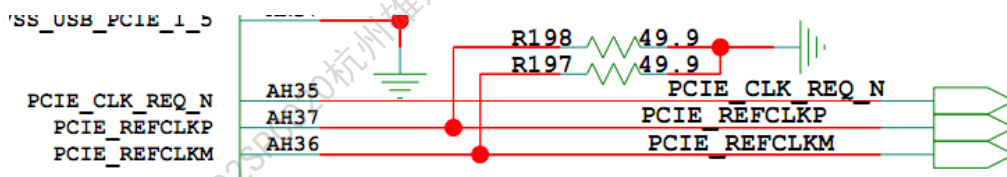
表2-1 PCIe 时钟源选择

信号	方向	功能
PCIE_CLK_REQ_N	I	PCIE 参考时钟源选择（内部下拉）。 0：内部时钟； 1：外部时钟。

2.2 在级联场景时钟设计注意事项

1. 作为主片的时候，时钟输出管脚靠近芯片必须加 49.9 欧姆电阻对地。

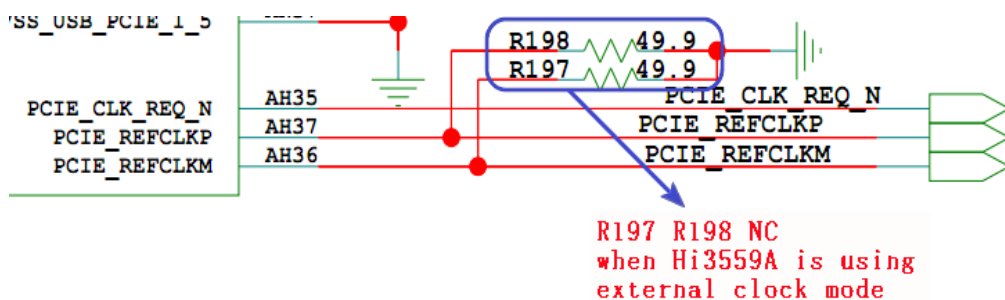
图2-1 PCIe 主片场景时钟图



2. 作为从片且设置为外部时钟时，时钟输入管脚不需加 49.9 欧姆电阻对地。



图2-2 PCIE 从片场景时钟图



注意

时钟选择错误将导致从片 PCIe 无法使用外部时钟模式。

图2-3 PCIe 模式图

PCIE_USB3_MODE[1:0] :

00:PCIE X2 Mode

01:PCIE X1 + USB3 P1

10:USB3 P0 + USB3 P1

11:No such Configuration

几种典型的 PCIe 级联示意图。

图2-4 两片 PCIe X1 级联示意图

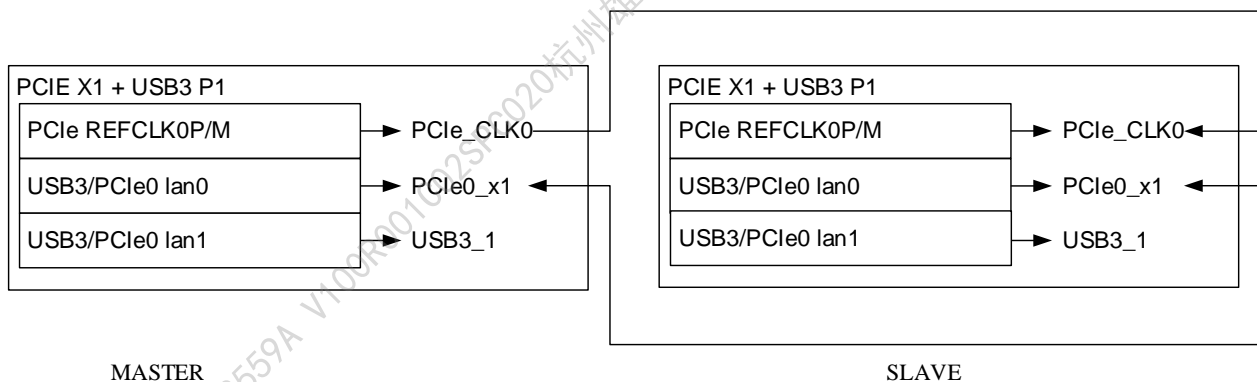
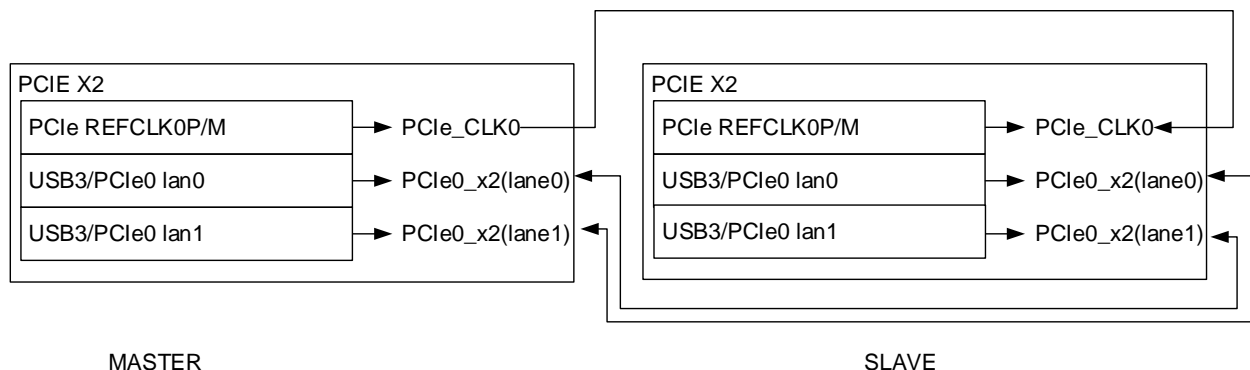




图2-5 两片 PCIe X2 级联示意图



2.3 从启动软件配置

2.3.1 PCIE 时钟选择模式配置

《Hi35xxVxxxultra-HD Mobile Camera SoC 用户指南》或《Hi3519AV100 4K Smart IP Camera SoC 用户指南》文档中“外围设备->PCI Express->工作方式->时钟设置”章节的相关描述。

2.3.2 PCIE_MCC 驱动配置

芯片只有一个 PCIE 控制器，PCIE_MCC 驱动的默认配置已在 PCIE_MCC 驱动代码的./hi35xx_dev/arch/config.h 文件中进行设置，不需要修改。



3 PCIE 基础知识

3.1 概述

PCIE 是外围设备互连（Peripheral Component Interconnect Express）的简称，作为一种通用的总线接口标准，在目前的计算机系统中得到了非常广泛的应用。PCIE 总线的时钟频率一般使用 2.5GHz，在 32bit 系统中，最基本的 PCI Express x1 模式理论极限速度可达 500MB/s（ $2.5\text{Gbps} \times (1 \text{ B}/8\text{bit}) \times (8\text{b}/10\text{b}) \times 2$ ）；但 PCIE 无法一直维持在峰值传输的状态，一般只能保持在 50~60% 的传输效率，即 250MB/s~300MB/s 的速度。如果需要更高的传输速度，可以使用 PCI Express x 2、x 4、x 8 或 x 16 模式，数据理论传输速率分别达到 1GBps、2GBps、4GBps 和 8GBps。

PCIE 设备上有三种地址空间，对应三种 PCIE 总线命令，具体如表 3-1 所示。



说明

CPU 可以访问 PCIE 设备上的所有地址空间。

表3-1 PCIE 地址空间和命令

地址空间	描述	命令	说明
I/O 空间	提供给设备驱动程序使用	I/O 操作命令	对设备对应的 I/O 地址空间进行访问，此类访问不可预取。
存储空间		Memory 操作命令	对设备的 Memory 空间进行访问，其中 Memory 操作命令又可分为 Prefetchable（可预取）和 Non-prefetchable（不可预取）两种类型。
配置空间	提供 Linux 内核中的 PCIE 初始化代码使用	配置访问命令	对设备的配置空间进行读写访问，用来初始化设备，给设备分配资源。

内核在启动时负责对所有 PCIE 设备进行初始化，配置所有的 PCIE 设备，包括中断号以及 I/O 基址，并在文件 /proc/bus/pci/devices 中列出所有找到的 PCIE 设备，以及这些设备的参数和属性。



请查阅有关 PCIE 规范的资料获取 PCIE 协议详细说明。本文重点介绍关注业务应用中常用的知识。

3.2 PCIE DMA 方式数据传输

PCIE 模块内建 DMAC，可直接由 PCIE 接口发起 DMA 操作，此时不需要 ARM 的干预，可获得更好的系统性能。

支持 PCIE 从到主、主到从的 DMA 读写传输，主要用于传输预览图像、解码图像以及码流等数据。预览图像和解码图像的传输由 PCIV 模块在内核态调用接口完成，而码流数据的传输则需要用户调用 PCIV 模块封装的接口来完成。

软件提供的 DMA 写操作时传输接口需要以下输入参数：

- 目标物理地址（即 PCIE 地址）
- 源物理地址（即 AHB 地址）
- 传输长度

从到主的 DMA 写操作：源地址使用从片的 AHB 地址（即 DDR 地址），目标地址使用主片的 DDR 地址（即其 PCIE 地址）。

主到从的 DMA 写操作：源地址使用主片的 AHB 地址（即 DDR 地址），目标地址使用从片的窗口 PF 地址空间对应的 PCIE 地址。

软件提供的 DMA 读操作时传输接口需要以下输入参数：

- 目标物理地址（即 AHB 地址）
- 源物理地址（即 PCIE 地址）
- 传输长度

从到主的 DMA 读操作：源地址使用主片的 DDR 地址（即其 PCIE 地址），目标地址使用从片的 AHB 地址（即 DDR 地址）。

主到从的 DMA 读操作：源地址使用从片的窗口 PF 地址空间对应的 PCIE 地址，目标地址使用主片的 AHB 地址（即 DDR 地址）。

3.3 PCIE 共享内存方式数据传输

作为 Hi35xxVxxx PCIE Host 的可以通过 AHB-PCIE window 实现 ARM core 对 PCIE 总线上的其它设备的访问。

AHB-PCIE window 上存在三种地址空间，分别是：

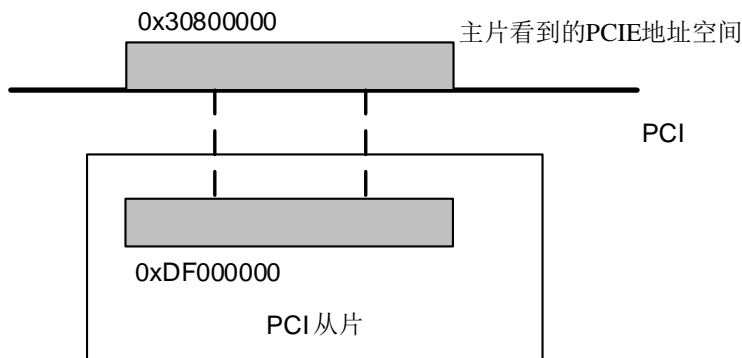
- 非可预取内存空间（NP Memory）
- 可预取内存空间（PF Memory）
- IO 空间（IO）

PCIE 主设备由操作系统统一分配从设备配置空间中的 BAR0，BAR1，BAR2 寄存器的基址，并通过配置空间访问写入；从设备则可以在 AHB 总线上通过寄存器配置 NP



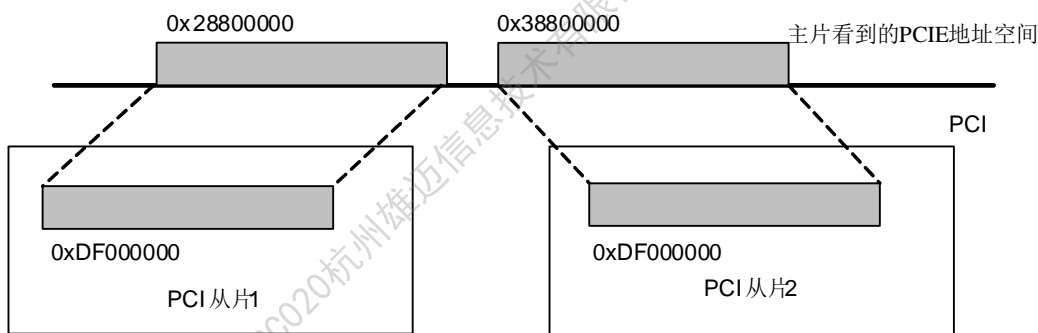
Memory 和 PF Memory 的 AHB 侧基地址以及范围大小（最大 8M）。因此主设备上看到的各个从设备的窗口的 PCIE 地址与从设备本身的 AHB 侧基地址一一对应，即可使用 Window 中的可预取内存空间实现主从片间的内存共享，具体如图 3-1 和图 3-2 所示。

图3-1 PCIE 地址与 AHB 侧地址的映射关系图



如图 3-1 所示，有 1 个 PCIE 控制器，主片的 PCIE 地址 0x30800000 与从片 1 上的 AHB 地址 0xDF000000 可以认为存在映射关系，在主片上对 0x30800000 区域的读写访问可以通过 PCIE 总线反应到从片的 0xDF000000 区域；主片上的 PCIE 基址 0x30800000 是通过读取从片 1 的 BAR0 寄存器而得到（用户程序可以通过 PCIV 模块封装的接口获取），而从片上的 AHB 基址 0xDF000000 是由驱动程序写入到 PCIE iATU 寄存器组中的目标地址低位寄存器（PCIE 支持 64 位地址操作，目标地址高位寄存器写 0），用户可以在加载 hi35xx_dev_slv.ko 模块时通过修改模块参数来更改此 AHB 基址。

图3-2 PCIE 地址与 AHB 侧地址的映射关系图



如图 3-2 所示，有 2 个 PCIE 控制器，主片的 PCIE 地址 0x28800000 与从片 1 上的 AHB 地址 0xDF000000 可以认为存在映射关系，主片的 PCIE 地址 0x38800000 与从片 2 上的 AHB 地址 0xDF000000 可以认为存在映射关系。



注意

PF Memory 在从片上的地址范围是可以不断移动的（即窗口的移动），但鉴于整个系统的稳定性以及 PCIE 消息模块对 NP 基址的依赖性，不建议在从片启动后再移动窗口。

3.4 PCIE MCC 消息应用

PCIE MCC 消息模块基于 PCIE 的 Window 窗口机制以及系统全局软中断（Hi 中断），实现 PCIE 主从设备间的消息通讯功能。

用户态接口包括：获取 PCIE 本地以及对端的 ChipId 号，主从片间的相互检测机制（即通讯握手），消息端口的打开、关闭，消息的读和写、以及 Select 接口等。

从 MCC 模块获取到的 ChipId 号，在 PCIE 主设备上为 0，在 PCIE 从设备上则为 PCIE slot 号（由于 PCIV 模块内部的消息通讯也基于 MCC，因此 PCIV 中的 ChipId 与此一致）。

MCC 的消息缓存池使用 PCIE 窗口中的可预取内存空间，且固定使用前 1M 的地址范围，例如加载 hi35xx_dev_slv.ko 模块时，配置窗口范围为 0xDF000000 的 8M 地址范围，则从 0xDF000000 开始的 1M 空间分配给 MCC 模块使用，用户程序不应该再去使用它。

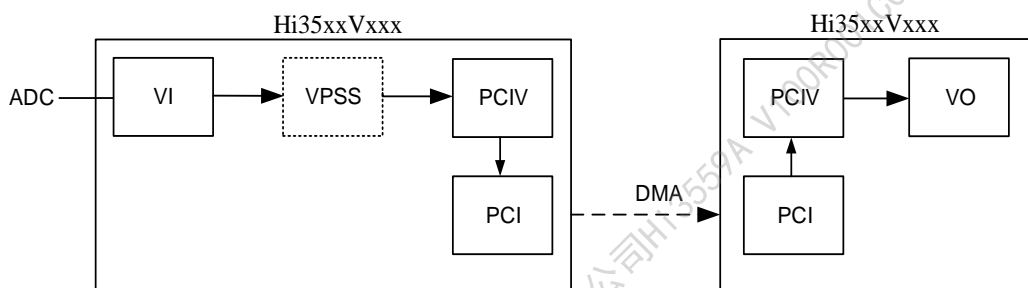


4 PCIE 级联业务实现

4.1 视频预览

视频预览主要用于芯片之间级联，视频预览是将从片的 VI 图像传送到 PCIE 总线上的主片的 VO 设备上显示。基本的数据流处理如图 4-1 所示。

图4-1 视频预览数据流处理流程



数据流的控制和传输由 MPP 系统在内核态完成，用户只需要调用 MPI 接口完成相应配置、使用 PCIE 消息机制完成部分命令的传递。传输通路建立以后，正常图像传输则不需用户干预。



说明

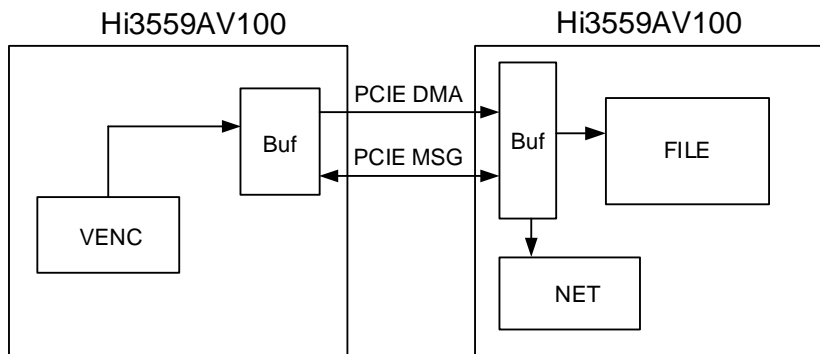
- PCIV 相关接口的详细说明和注意事项请参见“5 PCIV 开发参考”。
- PCIE 消息的相关接口则使用 mcc 模块提供的 ioctl 命令。
- 从片在送 PCIV 之前，建议先通过 VPSS 模块进行所需处理。

4.2 码流传输

码流传输是将芯片的视频编码码流数据或者解码码流数据传送到 PCIE 总线上的其他芯片设备。基本的数据流处理如图 4-2 所示。



图4-2 码流传输数据流处理流程



码流发送端首先从 VENC 通道中获取编码码流数据，将其拷贝至准备好的 stream buffer 中，然后通过 PCIE 的 DMA 将码流数据发送到 PCIE 对端的 stream buffer 中，对端再将码流取出通过网络发送或存文件，每次传输的读写位置信息可以通过 PCIE 消息发送到对端以便进行发送和接收的同步控制。发送端和接收端的 stream buffer 需要用户自行实现，推荐采用不定长的循环 buffer，每次传输多帧数据。



说明

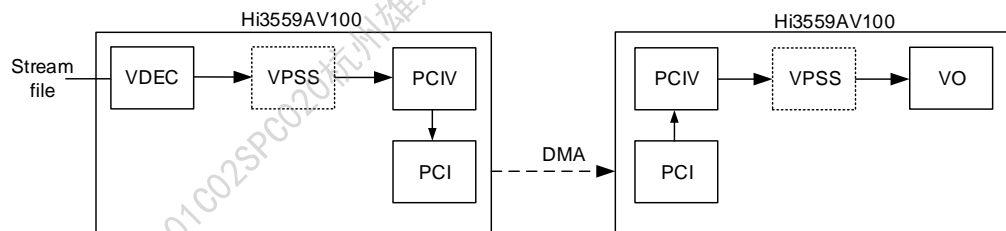
- PCIE DMA 的传输使用 PCIV 模块提供的相应接口。
- PCIE 消息则使用 MCC 模块提供的相应接口。
- 注意：码流的发送和接收都需要四字节对齐。

详细操作及流程可以参考 SDK 中的样例程序。

4.3 解码回放图像显示

解码回放图像是将芯片的 VDEC 解码后图像传送到 PCIE 总线上的其他芯片的 VO 设备上显示。基本的数据流处理如图 4-3 所示。

图4-3 解码回放图像数据流处理流程



具体的数据流处理和接口调用与 PCIE 预览流程类似，主要区别如下：

- 用户需要创建解码通道并向其发送码流进行解码；
- PCIV 相应传输通路建立后，PCIV 模块可以接收 VDEC 通道解码后图像数据，处理后通过 PCIE DMA 发送到对端；也可接收 VPSS 处理后的图像数据，通过 PCIE DMA 发送到对端。



对端的数据接收及 VO 显示与预览流程一致。



说明

- 从片在送 PCIV 之前，建议先通过 VPSS 模块进行所需处理。
- 如 VO 显示高清图像，需要在主片在 VO 之前调用 VPSS 进行处理。

4.4 内存配置

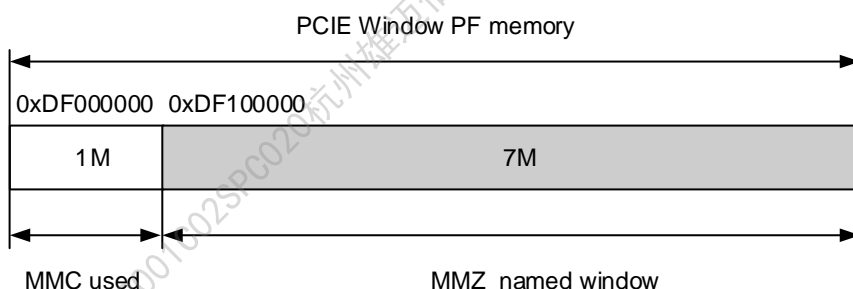
与 PCIE 业务相关的内存配置时，需要注意以下事项：

- PCIE 从设备加载 `mcc_drv_slv.ko` 模块时，配置窗口 PF 地址范围最大为 8M。需要注意，前 1M 固定分配给 MCC 模块。
- 图像或码流的传输过程中，使用的源地址和目标地址都是由 MMZ 管理，用户可通过 MMZ 的 VideoBuffer 相关接口获取。其中图像的传输由 PCIV 模块内部从 VideoBuff 中获取内存，而码流传输需要用户在用户态调用接口获取 VideoBuffer 内存。
- 从到主的 DMA 数据传输（包括图像和码流）时，源地址和目标地址都可以是各自 DDR 上的任意可用地址。
- 主到从或者从到从的 DMA 数据传输时，目标地址必须使用 PCIE 窗口映射的 PCIE 地址。

以主到从的码流传传输为例，主片上的码流发送 buffer 可位于 DDR 上任意有效地址空间（可以使用 VideoBuffer 相关接口申请内存）；而从片上的码流接收 buffer 则必须在 PCIE 窗口的 PF Memory 空间范围内，获取其内存可以调用 PCIV 相应接口：首先调用 `HI_MPL_PCIV_WinVbCreate` 创建基于 window mmz 的缓存池，然后调用 `HI_MPL_PCIV_Malloc` 接口从其中获取缓存块。

如图 4-4 所示，命名为 window 的 MMZ 区域位于整个 PF 窗口区域的后 7M 范围内，此 MMZ 区域是在从片的初始化加载脚本中创建，基址为 `0xDF100000`，对应的整个从片 PCIE 窗口的 PF 基址是 `0xDF000000`，如果用户需要修改 PCIE 窗口 PF 基址，则必须同时修改 window MMZ 的基址。

图4-4 PCIE 窗口 PF 区域和 MMZ 区域的示意





5 PCIV 开发参考

5.1 PCIV 概述

PCIV 模块主要提供 PCIE 多片间图像数据传输等相关 MPI 接口。具体包括：



说明

PCIV 模块不提供 PCIE 消息通讯、编码与 PCIV 通道的绑定等接口，这些接口由其他模块提供或者由用户实现。

- 预览图像的传输：图像发送端绑定 VI 通道或者绑定 VPSS 通道，图像接收端绑定 VO 通道或者 VENC 通道或者 VPSS 通道，发送端和接收端进行相应配置并启动后，即可将发送端的 VI 图像发送到接收端的 VO 通道进行显示或者送给 VPSS/VENC 进行后续处理。
- 解码回放图像的传输：图像发送端绑定 VDEC 通道或者绑定 VPSS 通道，图像接收端绑定 VO 通道或者 VENC 通道或者 VPSS 通道，发送端和接收端进行相应配置并启动后，即可将 VDEC 通道解码后图像发送到接收端的 VO 通道进行显示或者送给 VPSS/VENC 进行后续处理。
- 码流数据的传输：提供启动 PCIE DMA 传输的 MPI 接口，用户可以调用此接口将编码码流或者解码码流或者其他数据通过 PCIE DMA 传输到 PCIE 对端。

5.2 PCIV MPI 参考

本模块所有 MPI 接口需要在完整，正确的级联环境下调用，否则可能会引发不可预知的异常。本功能模块提供以下 MPI：

- [HI_MPI_PCIV_Create](#)：创建 PCIV 通道。
- [HI_MPI_PCIV_Destroy](#)：销毁 PCIV 通道。
- [HI_MPI_PCIV_SetAttr](#)：设置 PCIV 通道属性。
- [HI_MPI_PCIV_GetAttr](#)：获取 PCIV 通道属性。
- [HI_MPI_PCIV_Start](#)：启动 PCIV 通道。
- [HI_MPI_PCIV_Stop](#)：停止 PCIV 通道。
- [HI_MPI_PCIV_DmaTask](#)：启动 PCIE DMA 任务。
- [HI_MPI_PCIV_Malloc](#)：分配 PCIV 缓存。



- [HI_MPI_PCIV_Free](#): 释放 PCIV 缓存。
- [HI_MPI_PCIV_MallocChnBuffer](#): 分配 PCIV 通道内存。
- [HI_MPI_PCIV_FreeChnBuffer](#): 释放 PCIV 通道内存。
- [HI_MPI_PCIV_GetLocalId](#): 获取本地 PCIE 节点的设备 ID 号。
- [HI_MPI_PCIV_EnumChip](#): 获取所有与运行此函数的 PCIE 节点连接其它 PCIE 节点的设备 ID 号。
- [HI_MPI_PCIV_GetBaseWindow](#): 获取 PCIE 窗口信息。
- [HI_MPI_PCIV_WinVbCreate](#): 创建 PCIV 专用缓存池。
- [HI_MPI_PCIV_WinVbDestroy](#): 销毁 PCIV 专用缓存池。
- [HI_MPI_PCIV_Show](#): 显示 PCIV 图像。
- [HI_MPI_PCIV_Hide](#): 隐藏 PCIV 图像。
- [HI_MPI_PCIV_SetPreProcCfg](#): 设置 PCIV 前处理配置。
- [HI_MPI_PCIV_GetPreProcCfg](#): 获取 PCIV 前处理配置。

HI_MPI_PCIV_Create

【描述】

创建 PCIV 通道。

【语法】

```
HI_S32 HI_MPI_PCIV_Create(PCIV_CHN pcivChn, const PCIV_ATTR_S *pPcivAttr);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围: [0, PCIV_MAX_CHN_NUM)。	输入
pPcivAttr	PCIV 通道属性。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为 错误码 。

【需求】

- 头文件: [hi_comm_pciv.h](#)、[mpi_pciv.h](#)
- 库文件: [libmpi.a](#) [libpciv.a](#)



【注意】

- 重复创建通道会返回错误。
- PCIV 通道属性包含以下参数：
 - 目标图像属性 (stPicAttr): 配置图像显示端的图像宽、高、像素格式等。
 - 图像缓冲块大小 (u32BlkSize): 每块图像缓冲块应该与一帧目标图像大小一致。
 - 图像缓冲块个数 (u32Count): 取值范围为 2~PCIV_MAX_BUF_NUM。建议缓冲块的个数为 5 个, 同时可以根据传输业务适当的增加或者减少。
 - 图像缓冲块物理地址 (u64PhyAddr): 图像接收方的 buffer 中每块缓冲块的物理地址, 接收方通过调用接口 [HI_MPI_PCIV_MallocChnBuffer](#) 获取, 然后用户可以通过 PCIE 消息将地址信息传递给图像发送方。
 - 对端 PCIE 通道信息 (stRemoteObj): 与当前本地 PCIE 通道对应的对端 PCIE chip 号及 PCIV 通道号。
- 主从片在创建 PCIV 通道的时候, 需要用户保证主从片间有绑定关系的 PCIV 通道属性一致。

HI_MPI_PCIV_Destroy

【描述】

销毁 PCIV 通道。

【语法】

```
HI_S32 HI_MPI_PCIV_Destroy (PCIV_CHN pcivChn);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围: [0, PCIV_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为 错误码 。

【需求】

- 头文件: hi_comm_pciv.h、mpi_pciv.h
- 库文件: libmpi.a libpciv.a



【注意】

- 如果未创建通道，直接返回成功。
- 销毁通道前必须先停止通道，否则返回错误。

HI_MPI_PCIV_SetAttr

【描述】

设置 PCIV 通道的属性。

【语法】

```
HI_S32 HI_MPI_PCIV_SetAttr(PCIV_CHN pcivChn, const PCIV_ATTR_S  
*pPcivAttr);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入
pPcivAttr	PCIV 通道属性结构体指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 设置属性前必须先创建通道。
- 如果通道已经启动，则不允许设置通道属性。

HI_MPI_PCIV_GetAttr

【描述】

获取 PCIV 通道的属性。

【语法】



```
HI_S32 HI_MPI_PCIV_GetAttr(PCIV_CHN pcivChn, PCIV_ATTR_S *pPcivAttr);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入
pPcivAttr	PCIV 通道属性结构体指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

必须先创建了通道或者设置通道属性，然后才能获取通道属性。

HI_MPI_PCIV_Start

【描述】

启动 PCIV 通道。

启动通道的具体行为取决于其绑定的通道类型：

- 绑定的是 VI 通道：先将 VI 图像加工处理为配置的目标图像，然后通过 PCIE 发送到对端。
- 绑定的是 VDEC 通道：先从 VDEC 通道获取解码后图像，加工处理为配置的目标图像，然后通过 PCIE 发送到对端。
- 绑定的是 VO 通道：从 PCIV 通道的 buffer 物理地址中取出帧图像，将其发送到后端绑定的 VO 通道进行显示或接收前端绑定的虚拟 VO 设备的图像，加工处理为配置的目标图像，然后通过 PCIE 发送到对端。

【语法】

```
HI_S32 HI_MPI_PCIV_Start(PCIV_CHN pcivChn);
```

【参数】



参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 启动通道前必须先创建通道、设置属性并绑定，否则返回错误。
- 如果已经启动通道，则直接返回成功。

HI_MPI_PCIV_Stop

【描述】

停止 PCIV 通道。

【语法】

```
HI_S32 HI_MPI_PCIV_Stop(PCIV_CHN pcivChn);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。



【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

如果未启动通道，则直接返回成功。

HI_MPI_PCIV_DmaTask

【描述】

创建 PCIE DMA 传输任务。

用于调用者自行启动一次或一系列的 PCIE DMA 传输任务。

【语法】

```
HI_S32 HI_MPI_PCIV_DmaTask(PCIV_DMA_TASK_S *pTask);
```

【参数】

参数名称	描述	输入/输出
pTask	PCIE 任务结构体指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为 错误码 。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 此接口为阻塞接口，即等到 DMA 任务完成后接口才返回。
- 源地址和目标地址必须是合法的物理地址。

HI_MPI_PCIV_Malloc

【描述】

申请 PCIE 相关内存。



用于申请 PCIE 主从片间数据传输的相关内存，可以一次申请多块指定大小的内存块。

【语法】

```
HI_U32 HI_MPI_PCIV_Malloc(HI_U32 u32BlkSize, HI_U32 u32BlkCnt, HI_U64  
au64PhyAddr[]);
```

【参数】

参数名称	描述	输入/输出
u32BlkSize	内存块大小。	输入
u32BlkCnt	内存块个数。	输入
au64PhyAddr	物理地址数组。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 在 PCIE 主片上调用此接口时，直接从 MPP 的公共 VideoBuffer 中申请缓存块。需要保证 VideoBuffer 公共缓存池中有足够满足条件的缓存块，否则返回错误，VideoBuffer 公共缓存池的相关概念请参见《HiMPP IPC Vx.0 媒体处理软件开发参考》中系统控制章节相关内容。
- 在 PCIE 从片上调用此接口时，从 PCIV 专用缓存池中申请缓存块，因此必须先调用 HI_MPI_PCIV_WinVbCreate 接口创建 PCIV 专用缓存池（其内存从名为“window”的 MMZ 空间中分配）。
- 此接口需要用户保证输入的地址数量参数 u32BlkCnt 与地址数组 au64PhyAddr 长度匹配。
- 与接口 HI_MPI_PCIV_Free 配合使用，否则会造成释放失败或程序异常。

HI_MPI_PCIV_Free

【描述】

释放 PCIV 相关内存。

【语法】



```
HI_S32 HI_MPI_PCIV_Free(HI_U32 u32BlkCnt, HI_U64 au64PhyAddr[]);
```

【参数】

参数名称	描述	输入/输出
u32BlkCnt	内存块个数，取值范围大于零。	输入
au64PhyAddr	物理地址数组。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

与接口 [HI_MPI_PCIV_Malloc](#) 配合使用，否则会造成释放失败或程序异常。

HI_MPI_PCIV_MallocChnBuffer

【描述】

分配 PCIV 通道内存。

用于申请接收方 PCIV 通道的相关内存，可以一次申请多块指定大小的内存块。

【语法】

```
HI_U32 HI_MPI_PCIV_MallocChnBuffer(PCIV_CHN pcivChn, HI_U32 u32BlkSize,  
HI_U32 u32BlkCnt, HI_U64 au64PhyAddr[]);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)	输入
u32BlkSize	内存块大小。	输入
u32BlkCnt	内存块个数。 取值范围：[0, PCIV_MAX_BUF_NUM]	输入



参数名称	描述	输入/输出
au64PhyAddr	物理地址数组。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 在 PCIE 主片上调用此接口时，直接从 MPP 的公共 VideoBuffer 中申请缓存块。需要保证 VideoBuffer 公共缓存池中有足够满足条件的缓存块，否则返回错误，VideoBuffer 公共缓存池的相关概念请参见《HiMPP IPC Vx.0 媒体处理软件开发参考》中的“系统控制”章节相关内容。
- 调用此接口分配该通道号对应的通道内存后，在创建该通道号对应的通道时必须使用已经分配的这几块内存，否则返回错误。
- 必须在创建该通道号对应的通道之前，调用此接口分配该通道号对应的通道内存。
- 在 PCIE 从片上不需要分配通道内存，因此该接口只能在主片上调用，在从片上调用此接口时，返回错误。
- 当接口分配内存失败时，会将传出参数 au64PhyAddr 所指向的数组中的元素都置为 0，用来防止用户使用已经释放了的内存。
- 此接口需要用户保证输入的地址数量参数 u32BlkCnt 与地址数组 au64PhyAddr 长度匹配。
- 此接口需与接口 HI_MPI_PCIV_FreeChnBuffer 配合使用，否则会造成释放失败或程序异常。

HI_MPI_PCIV_FreeChnBuffer

【描述】

释放 PCIV 通道内存。

【语法】

```
HI_S32 HI_MPI_PCIV_FreeChnBuffer(PCIV_CHN pcivChn, HI_U32 u32BlkCnt);
```

【参数】



参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)	输入
u32BlkCnt	内存块个数。 取值范围：[0, PCIV_MAX_BUF_NUM]	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 必须在通道停止或销毁后，再调用此接口释放通道号对应的通道内存，否则返回错误。
- 此接口需与接口 [HI_MPI_PCIV_MallocChnBuffer](#) 配合使用，否则会造成释放失败或程序异常。

HI_MPI_PCIV_GetLocalId

【描述】

获取本地 PCIE 节点的设备 ID 号。

【语法】

```
HI_S32 HI_MPI_PCIV_GetLocalId(HI_VOID);
```

【返回值】

返回值	描述
s32ChipId	PCIE 设备 ID 号。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

**【注意】**

- 主片的 PCIE 设备号为 0;
- 从片的 PCIE 设备号大于 0;

HI_MPI_PCIV_EnumChip

【描述】

获取所有与运行此函数的 PCIE 节点连接其它 PCIE 节点的设备 ID 号。

- 在主片上可以获取所有从片的 PCIE 节点的设备 ID 号。
- 在从片上则可以获取与此从片连接的主片的 PCIE 节点的设备 ID 号。

【语法】

```
HI_S32 HI_MPI_PCIV_EnumChip(HI_S32 as32ChipID[PCIV_MAX_CHIPNUM]);
```

【参数】

参数名称	描述	输入/输出
as32ChipID	保存与此节点连接的其它 PCIE 节点的设备 ID 号的数组	输出

【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

【需求】

- 头文件: hi_comm_pciv.h、mpi_pciv.h
- 库文件: libmpi.a libpciv.a

【注意】

- 在主片上调用此接口时, as32ChipId 数组为所有从片的 PCIE 设备 ID 号 (大于 0)。
- 在从片上调用此接口时, as32ChipId 数组只有 as32ChipId[0]保存主片的 PCIE 设备 ID 号 (等于 0)。

HI_MPI_PCIV_GetBaseWindow

【描述】

获取 PCIE 窗口信息。

- 在主片上可以获取所有从片的 PCIE 窗口信息。



- 在从片上则只能获取本片的可预取空间的 AHB 地址信息。

【语法】

```
HI_S32 HI_MPI_PCIV_GetBaseWindow(HI_S32 s32ChipId, PCIV_BASEWINDOW_S  
*pBase);
```

【参数】

参数名称	描述	输入/输出
s32ChipId	PCIE 设备 ID 号。 主片设备 ID 号为 0，从片设备 ID 大于 0。	输入
pBase	PCIE 窗口信息结构体指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- 在主片上调用此接口时，s32ChipId 为指定从片的 PCIE 设备 ID 号（大于 0）。
- 在从片上调用此接口时，s32ChipId 必须置为 0。

HI_MPI_PCIV_WinVbCreate

【描述】

创建 PCIV 专用缓存池。

一般用于在 PCIE 从片上创建一个专用的 VideoBuffer 缓存池（基于名称为“window”的 MMZ 区域）。此专用 VideoBuffer 缓存池与接口 HI_MPI_VB_SetConf 配置的缓存池概念类似；创建此缓存池的目的是用于实现 PCIE 主到从或从到从的数据传输及共享。

【语法】

```
HI_S32 HI_MPI_PCIV_WinVbCreate(PCIV_WINVBCFG_S *pCfg);
```

【参数】



参数名称	描述	输入/输出
pCfg	PCIV 缓存池结构体指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

在 PCIE 主片上不支持创建专用的 VideoBuffer 缓存池，因此该接口只能在从片上调用，在主片上调用此接口时，返回错误。

HI_MPI_PCIV_WinVbDestroy

【描述】

销毁 PCIV 专用缓存池。

【语法】

```
HI_S32 HI_MPI_PCIV_WinVbDestroy(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】



- 与 [HI_MPI_PCIV_WinVbCreate](#) 接口配对使用。
- 在 PCIE 主片上不支持创建或销毁专用的 VideoBuffer 缓存池，因此该接口只能在从片上调用，在主片上调用此接口时，返回错误。

HI_MPI_PCIV_Show

【描述】

显示（传输）PCIV 图像。

与 [HI_MPI_PCIV_Hide](#) 接口配合使用，用于控制是否通过 DMA 传输图像数据到 PCIE 对端设备。

【语法】

```
HI_S32 HI_MPI_PCIV_Show(PCIV_CHN pcivChn);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为 错误码 。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

PCIV 通道创建时默认为显示 PCIV 图像，即发送端会传输实际图像数据到接收端。

HI_MPI_PCIV_Hide

【描述】

隐藏（不传输）PCIV 图像。

与 [HI_MPI_PCIV_Show](#) 接口配合使用，用于控制是否通过 DMA 传输图像数据到 PCIE 对端设备。

【语法】



```
HI_S32 HI_MPI_PCIV_Hide(PCIV_CHN pcivChn);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- PCIV 通道创建时默认为显示 PCIV 图像，即发送端会传输实际图像数据到接收端。
- 显示隐藏 PCIV 图像接口，一般用于接收端进行 VO 同步回放的应用场景。如果隐藏 PCIV 图像，发送端仍会从 VI 或 VDEC 取图像数据，但图像数据并不通过 PCIE 传输到接收端，而是只传输时间戳等信息，接收端会将带时间戳的图像信息发送给 VO，但图像数据不正确，因此此时 VO 应该设置为不显示。

HI_MPI_PCIV_SetPreProcCfg

【描述】

设置 PCIV 前处理配置。

【语法】

```
HI_S32 HI_MPI_PCIV_SetPreProcCfg(PCIV_CHN pcivChn, const
PCIV_PREPROC_CFG_S *pstCfg);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入
pstCfg	PCIV 前处理配置结构体指针。	输入



【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

前处理的配置在系统初始化时有默认值，如果不调用 [HI_MPI_PCIV_SetPreProcCfg](#) 进行更改，前处理则使用默认配置；如需要修改其中部分选项，应该先调用 [HI_MPI_PCIV_GetPreProcCfg](#) 获取配置，修改相应选项后，再调用 [HI_MPI_PCIV_SetPreProcCfg](#) 进行设置。

HI_MPI_PCIV_GetPreProcCfg

【描述】

获取 PCIV 前处理配置。

【语法】

```
HI_S32 HI_MPI_PCIV_GetPreProcCfg(PCIV_CHN pcivChn, PCIV_PREPROC_CFG_S
*pstCfg);
```

【参数】

参数名称	描述	输入/输出
pcivChn	PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。	输入
pstCfg	PCIV 前处理配置结构体指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。



【需求】

- 头文件：hi_comm_pciv.h、mpi_pciv.h
- 库文件：libmpi.a libpciv.a

【注意】

- PCIV 的前处理指的是来自 VI 或 VDEC 的源图像在发送到 PCIE 对端设备之前做的相应视频处理，包括图像缩放、搬移、滤波、格式转换等。
- 前处理的配置在系统初始化时有默认值，因此没有调用设置接口，也可以直接调用获取接口来获取当前的前处理配置。

5.3 数据类型

PCIV 数据类型定义如下：

- **PCIV_VIDEVICE_S**：定义 PCIV 中 VI 设备数据类型。
- **PCIV_VODEVICE_S**：定义 PCIV 中 VO 设备数据类型。
- **PCIV_VDECDEVICE_S**：定义 PCIV 中 VDEC 设备数据类型。
- **PCIV_VENCDEVICE_S**：定义 PCIV 中 VENC 设备数据类型。
- **PCIV_VPSSDEVICE_S**：定义 PCIV 中 VPSS 设备数据类型。
- **PCIV_REMOTE_OBJ_S**：定义 PCIV 对端目标结构体。
- **PCIV_PIC_ATTR_S**：定义 PCIV 图像属性结构体。
- **PCIV_ATTR_S**：定义 PCIV 属性结构体。
- **PCIV_WINVBCFG_S**：定义 PCIE Window VideoBuffer 结构体。
- **PCIV_BASEWINDOW_S**：定义 PCIE Window 信息结构体。
- **PCIV_DMA_BLOCK_S**：定义 PCIE DMA 任务块结构体。
- **PCIV_DMA_TASK_S**：定义 PCIE DMA 任务结构体。
- **PCIV_FILTER_TYPE_E**：PCIV 前处理中的滤波系数组类型。
- **PCIV_PREPROC_CFG_S**：PCIV 前处理配置结构体。
- **PCIV_MAX_CHN_NUM**：PCIV 最大通道数。
- **PCIV_MAX_CHIPNUM**：级联系统中最大 PCIV 设备数量。
- **PCIV_MAX_VBCOUNT**：在从片 PCIE 窗口地址中创建的最多 VB 池个数。
- **PCIV_MAX_BUF_NUM**：PCIV 通道最大 buff 个数。
- **PCIV_MAX_DMABLK**：PCIE 通道最大 DMA 任务个数。

PCIV_VIDEVICE_S

【说明】

定义 PCIV 中 VI 设备数据类型。

【定义】

```
typedef struct hiPCIV_VIDEVICE_S
```




```
{  
    VI_DEV viDev;  
    VI_CHN viChn;  
} PCIV_VIDEVICE_S;
```

【成员】

成员名称	描述
viDev	VI 设备号。 取值范围：[0, VIU_MAX_DEV_NUM)。
viChn	VI 通道号。 取值范围：[0, VIU_MAX_CHN_NUM)。

【注意事项】

VIU_MAX_DEV_NUM, VIU_MAX_CHN_NUM 具体数值大小参考《HiMPP V4.0 媒体处理软件开发参考》的“视频输入”章节。

PCIV_VODEVICE_S

【说明】

定义 PCIV 中 VO 设备数据类型。

【定义】

```
typedef struct hiPCIV_VODEVICE_S  
{  
    VO_DEV voDev;  
    VO_CHN voChn;  
} PCIV_VODEVICE_S;
```

【成员】

成员名称	描述
voDev	VO 设备号。 取值范围：[0, VO_MAX_DEV_NUM)。
voChn	VO 通道号。 取值范围：[0, VO_MAX_CHN_NUM)。

【注意事项】

VO_MAX_DEV_NUM, VO_MAX_CHN_NUM 具体数值大小参考《HiMPP V4.0 媒体处理软件开发参考》的“视频输出”章节。



PCIV_VDECDEVICE_S

【说明】

定义 PCIV 中 VDEC 设备数据类型。

【定义】

```
typedef struct hiPCIV_VDECDEVICE_S
{
    VDEC_CHN vdecChn;
} PCIV_VDECDEVICE_S;
```

【成员】

成员名称	描述
vdecChn	VDEC 通道号。 取值范围：[0, VDEC_MAX_CHN_NUM)。

【注意事项】

VDEC_MAX_CHN_NUM 具体数值大小参考《HiMPP V4.0 媒体处理软件开发参考》的“视频解码”章节。

PCIV_VENCDEVICE_S

【说明】

定义 PCIV 中 VENC 设备数据类型。

【定义】

```
typedef struct hiPCIV_VENCDEVICE_S
{
    VENC_CHN vencChn; /* venc_channel number */
} PCIV_VENCDEVICE_S;
```

【成员】

成员名称	描述
vencChn	VENC 通道号。 取值范围：[0, VENC_MAX_CHN_NUM)。

【注意事项】

VENC_MAX_CHN_NUM 具体数值大小参考《HiMPP V4.0 媒体处理软件开发参考》的“视频编码”章节。



PCIV_VPSSDEVICE_S

【说明】

定义 PCIV 中 VPSS 设备数据类型。

【定义】

```
typedef struct hiPCIV_VPSSDEVICE_S
{
    VPSS_GRP vpssGrp; /* vpss group number */
    VPSS_CHN vpssChn; /* vpss channel number */
} PCIV_VPSSDEVICE_S;
```

【成员】

成员名称	描述
vpssGrp	VPSS GROUP 号。 取值范围：[0, VPSS_MAX_GRP_NUM)。
vpssChn	VPSS 通道号。 取值范围：[0, VPSS_MAX_CHN_NUM)。

【注意事项】

VPSS_MAX_GRP_NUM, VPSS_MAX_CHN_NUM 具体数值大小参考《HiMPP V4.0 媒体处理软件开发参考》的“视频处理子系统”章节。

PCIV_REMOTE_OBJ_S

【说明】

定义 PCIV 对端目标结构体。

【定义】

```
typedef struct hiPCIV_REMOTE_OBJ_S
{
    HI_S32 s32ChipId;
    PCIV_CHN pcivChn;
} PCIV_REMOTE_OBJ_S;
```

【成员】

成员名称	描述
s32ChipId	对端 PCIE 设备的芯片序号。 取值范围：[0, PCIV_MAX_CHIPNUM)。



成员名称	描述
pcivChn	对端 PCIE 设备的 PCIV 通道号。 取值范围：[0, PCIV_MAX_CHN_NUM)。

【注意事项】

无

PCIV_PIC_ATTR_S

【说明】

定义 PCIV 图像属性结构体。

【定义】

```
typedef struct hiPCIV_PIC_ATTR_S
{
    HI_U32    u32Width;
    HI_U32    u32Height;
    HI_U32    u32Stride[3];
    VIDEO_FIELD_E    u32Field;
    PIXEL_FORMAT_E    enPixelFormat;
    DYNAMIC_RANGE_E    enDynamicRange;
    COMPRESS_MODE_E    enCompressMode;
    VIDEO_FORMAT_E    enVideoFormat;
} PCIV_PIC_ATTR_S;
```

【成员】

成员名称	描述
u32Width	图像宽度。要大于 0，2 对齐。
u32Height	图像高度。要大于 0，2 对齐。
u32Stride	图像跨度。u32Stride[0]，大于等于 u32Width，16 对齐。 u32Stride[1]，u32Stride[2]保留参数，暂未使用。
u32Field	图像帧场选择。 此项不需要配置。
enPixelFormat	图像像素格式。目前支持 PIXEL_FORMAT_YVU_SEMIPLANAR_420、 PIXEL_FORMAT_YVU_SEMIPLANAR_422。
enDynamicRange	目标图像动态范围。具体描述请参见《HiMPP V4.0 媒体 处理软件开发参考》的“系统控制”章节。



成员名称	描述
enCompressMode	目标图像压缩模式。具体描述请参见《HiMPP V4.0 媒体处理软件开发参考》的“系统控制”章节。
enVideoFromat	目标图像视频格式。具体描述请参见《HiMPP V4.0 媒体处理软件开发参考》的“系统控制”章节。

【注意事项】

无

PCIV_ATTR_S

【说明】

定义 PCIV 属性结构体。

【定义】

```
typedef struct hiPCIV_ATTR_S
{
    PCIV_PIC_ATTR_S  stPicAttr;
    HI_S32            s32BufChip;
    HI_U32            u32BlkSize;
    HI_U32            u32Count;
    HI_U64            u64PhyAddr[PCIV_MAX_BUF_NUM];
    PCIV_REMOTE_OBJ_S stRemoteObj;
} PCIV_ATTR_S;
```

【成员】

成员名称	描述
stPicAttr	图像属性。
s32BufChip	图像缓存所在 PCIE 设备的芯片序号。 取值范围：[0, PCIV_MAX_CHIPNUM)。 目前此参数无意义，不需要配置。
u32BlkSize	图像缓存块的大小。
u32Count	图像缓存块的个数。 取值范围：[2, PCIV_MAX_BUF_NUM]。
u64PhyAddr	图像缓存块的物理地址。
stRemoteObj	对端 PCIE 设备及 PCIV 通道信息。



【注意事项】

无

PCIV_WINVBCFG_S

【说明】

定义 PCIE Window VideoBuffer 结构体。

【定义】

```
typedef struct hiPCIV_WINVBCFG_S
{
    HI_U32 u32PoolCount;
    HI_U32 u32BlkSize[PCIV_MAX_VBCOUNT];
    HI_U32 u32BlkCount[PCIV_MAX_VBCOUNT];
} PCIV_WINVBCFG_S;
```

【成员】

成员名称	描述
u32PoolCount	缓存池个数。最大个数为 PCIV_MAX_VBCOUNT 。
u32BlkSize	缓存块大小。
u32BlkCount	缓存块个数。

【注意事项】

无

PCIV_BASEWINDOW_S

【说明】

定义 PCIE Window 信息结构体。

【定义】

```
typedef struct hiPCIV_BASEWINDOW_S
{
    HI_S32 s32ChipId;
    HI_U64 u64NpWinBase;
    HI_U64 u64PfWinBase;
    HI_U64 u64CfgWinBase;
    HI_U64 u64PfAHBAddr;
} PCIV_BASEWINDOW_S;
```

【成员】



成员名称	描述
s32ChipId	PCIE 设备的芯片序号。 取值范围：[0, PCIV_MAX_CHIPNUM 。
u64NpWinBase	NP 基地址。
u64PfWinBase	PF 基地址。
u64CfgWinBase	CFG 基地址。
u64PfAHBAddr	PF AHB 侧基地址。

【注意事项】

无

PCIV_DMA_BLOCK_S

【说明】

定义 PCIE DMA 任务块结构体。

【定义】

```
typedef struct hiPCIV_DMA_BLOCK_S
{
    HI_U64  u64SrcAddr;
    HI_U64  u64DstAddr;
    HI_U32  u32BlkSize;
} PCIV_DMA_BLOCK_S;
```

【成员】

成员名称	描述
u64SrcAddr	源地址。
u64DstAddr	目标地址。
u32BlkSize	任务块大小。 主片时，取值范围：（0, 7M]。 DMA 传输的地址空间的有效性由用户自己保证。

【注意事项】

无



PCIV_DMA_TASK_S

【说明】

定义 PCIE DMA 任务结构体。

【定义】

```
typedef struct hiPCIV_DMA_TASK_S
{
    HI_U32  u32Count;
    HI_BOOL bRead;
    PCIV_DMA_BLOCK_S *pBlock;
} PCIV_DMA_TASK_S;
```

【成员】

成员名称	描述
u32Count	任务个数。
bRead	是否 DMA 读操作。
pBlock	DMA 任务块结构体指针。

【注意事项】

无

PCIV_FILTER_TYPE_E

【说明】

定义 PCIV 前处理中的滤波系数组类型。

【定义】

```
typedef enum hiPCIV_FILTER_TYPE_E
{
    PCIV_FILTER_TYPE_NORM,
    PCIV_FILTER_TYPE_EX,
    PCIV_FILTER_TYPE_EX2,
    PCIV_FILTER_TYPE_BUTT
} PCIV_FILTER_TYPE_E;
```

【成员】

成员名称	描述
PCIV_FILTER_TYPE_NORM	普通滤波系数组。



成员名称	描述
PCIV_FILTER_TYPE_EX	扩展滤波系数组。
PCIV_FILTER_TYPE_EX2	扩展滤波系数组 2。

【注意事项】

无

PCIV_PREPROC_CFG_S

【说明】

定义 PCIV 前处理配置结构体。

【定义】

```
typedef struct hiPCIV_PREPROC_CFG_S
{
    PCIV_PIC_FIELD_E      enFieldSel;
    PCIV_FILTER_TYPE_E    enFilterType;
} PCIV_PREPROC_CFG_S;
```

【成员】

成员名称	描述
enFieldSel	源图像的帧场选择。
enFilterType	滤波系数组类型。

【注意事项】

无

PCIV_MAX_CHN_NUM

【说明】

PCIV 最大通道数。

【定义】

Hi3559AV100:

```
#define PCIV_MAX_CHN_NUM    128
```

Hi3519AV100:

```
#define PCIV_MAX_CHN_NUM    16
```



【成员】

无

PCIV_MAX_CHIPNUM

【说明】

级联系统中最大 PCIV 设备数量。

【定义】

```
#define PCIV_MAX_CHIPNUM 32
```

【成员】

无

PCIV_MAX_VBCOUNT

【说明】

在从片 PCIE 窗口地址中创建的最多 VB 池个数。

【定义】

```
#define PCIV_MAX_VBCOUNT 8
```

【成员】

无

PCIV_MAX_BUF_NUM

【说明】

PCIV 通道最大 buff 个数。

【定义】

```
#define PCIV_MAX_BUF_NUM 16
```

【成员】

无

PCIV_MAX_DMABLK

【说明】

PCIE 通道最大 DMA 任务个数。

【定义】

```
#define PCIV_MAX_DMABLK 128
```

【成员】



无

5.4 错误码

错误代码	宏定义	描述
0xA01A8002	HI_ERR_PCIV_INVALID_CHNID	PCIV 通道号无效
0xA01A8003	HI_ERR_PCIV_ILLEGAL_PARAM	PCIV 参数设置无效
0xA01A8004	HI_ERR_PCIV_EXIST	PCIV 通道已经存在
0xA01A8005	HI_ERR_PCIV_UNEXIST	PCIV 通道不存在
0xA01A8006	HI_ERR_PCIV_NULL_PTR	输入参数空指针错误
0xA01A8007	HI_ERR_PCIV_NOT_CONFIG	PCIV 通道属性未配置
0xA01A8008	HI_ERR_PCIV_NOT_SUPPORT	操作不支持
0xA01A8009	HI_ERR_PCIV_NOT_PERM	操作不允许
0xA01A800C	HI_ERR_PCIV_NOMEM	分配内存失败
0xA01A800D	HI_ERR_PCIV_NOBUF	分配 VideoBuffer 失败
0xA01A800E	HI_ERR_PCIV_BUF_EMPTY	PCIV 缓存为空
0xA01A800F	HI_ERR_PCIV_BUF_FULL	PCIV 缓存为满
0xA01A8010	HI_ERR_PCIV_SYS_NOTREADY	系统未初始化
0xA01A8012	HI_ERR_PCIV_BUSY	系统忙
0xA01A8040	HI_ERR_PCIV_TIMEOUT	任务超时

5.5 Proc 调试信息

5.5.1 PCIV

【调试信息】

以下为截取 Hi3559A V100 的 pciv 调试信息：

```
~ # cat /proc/umap/pciv
```

```
[PCIV] Version: [Hi3559AV100_MPP_V1.0.0.0 B010 Release], Build Time[Jan 20 2018, 13:07:12]
```



-----PCIV CHN ATTR-----

PciChn	Width	Height	Stride	Field	PixFmt	BufCnt	BufSize	PhyAddr0
0	1920	1080	1920	frm	sp420	5	3110400	69f38000

-----PCIV CHN STATUS-----

PciChn	RemtChp	RemtChn	GetCnt	SendCnt	RespCnt	LostCnt	NtfyCnt	BuffStatus
0	1	0	410077	410077	410073	0	0	1 1 0 1 1

-----PCIV MSG STATUS-----

PciChn	RdoneGap	MaxRDGap	MinRDGap	WdoneGap	MaxWDGap	MinWDGap
0	0	0	0	22032	24291	20992

~ #

【调试信息分析】

记录当前已启用的 PCIV 通道的属性和工作状态。

【参数说明】

参数		描述
PCIV CHN ATTR	PciChn	PCIV 通道号。
	Width	目标图像宽度。
	Height	目标图像高度。
	Stride	目标图像 Stride[0]。
	Field	目标图像帧场模式。
	PixFmt	目标图像像素格式。
	BufCnt	目标图像缓冲块个数。
	BufSize	目标图像缓冲块大小，以字节为单位。
	PhyAddr0	目标图像第一个缓冲块的起始物理地址。
PCIV CHN STATUS	PciChn	PCIV 通道号。
	RemtChp	对端 PCIE 设备序号。
	RemtChn	对端 PCIV 通道号。
	GetCnt	发送端表示从 VI 或 VDEC 或虚拟 VO 获取的图像帧数目； 接收端表示通过 PCIE DMA 接收到的图像帧数目。



参数		描述
	SendCnt	发送端表示通过 PCIE DMA 成功发送的图像帧数目； 接收端表示将接收的图像发送到 VO/VPSS/VENC 的图像帧数目。
	RespCnt	发送端表示 PCIE DMA 任务回调次数； 接收端表示 VO/VPSS/VENC 使用完图像后的释放图像帧数目。
	LostCnt	发送端表示未成功通过 PCIE DMA 发送的图像帧数目； 接收端表示接收到的图像未成功发送到 VO/VPSS/VENC 的图像帧数目。
	NtfyCnt	从片发给主片的图像帧，主片使用完后发给从片的通知次数
	BuffStatus	发送端表示目标图像缓冲块的空闲状态； 接收端表示本地接收图像缓冲块的空闲状态。 0: free 状态； 1: busy 状态； 2: hold 状态。
PCIV MSG STATUS	PciChn	PCIV 通道号。
	RdoneGap	发送端收到接收端消息的时间间隔。
	MaxRDGap	发送端收到接收端消息的最大时间间隔。
	MinRDGap	发送端收到接收端消息的最小时间间隔。
	WdoneGap	接收端收到发送端消息的时间间隔。
	MaxWDGap	接收端收到发送端消息的最大时间间隔。
	MinWDGap	接收端收到发送端消息的最小时间间隔。

5.5.2 PCIV FMW

【调试信息】

```
~ # cat /proc/umap/pcivfmw
```

```
[PCIVF] Version: [Hi3559AV100_MPP_V1.0.0.0 B010 Release], Build Time:[May 28 2018, 13:46:34]
```

```
-----PCIV FMW CHN INFO-----
PciChn      GetCnt      SendCnt      RespCnt      LostCnt      NewDo      OldUndo
PoolId0
```



0 552 0 0 552 0 0 0

-----PCIV FMW CHN PREPROC INFO-----

PciChn	FiltT	Field
0	0	both

-----PCIV FMW CHN PIC ATTR INFO-----

PciChn	Width	Height	Stride	PixFormat	Dynamic	Compress	VideoFormat
0	3840	2160	3840	YVU_SP_420	SDR10	SEG	TILE_64x16

-----PCIV CHN QUEUE INFO-----

PciChn	busynum	frenum	state	Timer
0	0	6	0	1638

-----PCIV CHN CALL VGS INFO-----

PciChn	JobSuc	JobFail	EndSuc	EndFail	MoveSuc	MoveFail
OsdSuc	OsdFail	ZoomSuc	ZoomFail	MoveCb	OsdCb	ZoomCb
0	0	0	0	0	0	0
0	0	0	0	0	0	0
~	#					

【调试信息分析】

记录当前已启用的 PCIV FMW 通道的属性和工作状态。

【参数说明】

参数		描述
PCIV FMW CHN INFO	PciChn	PCIV 通道号。
	Width	目标图像宽度。
	Height	目标图像高度。
	Stride0	目标图像 Stride[0]。
	GetCnt	发送端表示从 VI 或 VDEC 获取的图像帧数目； 接收端表示通过 PCIE DMA 接收到的图像帧数目。
	SendCnt	发送端表示成功调用 VGS 进行源图像处理的次数； 接收端表示将接收的图像发送到 VO/VPSS/VENC 的图像帧数目。
	RespCnt	发送端表示源图像 VGS 回调次数； 接收端表示 VO/VPSS/VENC 使用完图像后的释放图像帧数目。



参数		描述
	LostCnt	发送端表示未成功调用 VGS 处理源图像的次数（申请 VB 内存失败或者创建 VGS 任务失败）； 接收端表示接收到的图像未成功发送到 VO/VPSS/VENC 的图像帧数目。
	NewDo	vpss 查询为 NewDo 的次数。
	OldUndo	vpss 查询为 OldUndo 的次数。
	PoolId0	图像缓冲块所在的 VideoBuffer 缓存池序号。
PCIV FMW CHN PREPRO C INFO (仅图像 发送端有 效) (请 参考数据 结构 PCIV_PR EPROC_C FG_S)	PciChn	PCIV 通道号。
	FiltT	源图像滤波系数类型。
	Field	源图像缩放时的帧场选择。
PCIV FMW CHN PIC ATTR INFO	PciChn	PCIV 通道号。
	Width	目标图像宽度。
	Height	目标图像高度。
	Stride	目标图像 Stride。
	PixFormat	目标图像像素格式。
	Dynamic	目标图像动态范围。
	Compress	目标图像压缩模式。
	VideoFormat	目标图像视频格式。
PCIV CHN QUEUE INFO (仅 图像发送 端有效)	PciChn	PCIV 通道号。
	busynum	PCIV 通道上 busy 队列的节点数目
	freenum	PCIV 通道上 free 队列的节点数目
	state	PCIV 通道发送解码后图像时的通道状态
	Timer	PCIV 通道上，发送解码后图像的定时器运行次数
PCIV	PciChn	PCIV 通道号。



参数		描述
CHN CALL VGS INFO (仅 图像发送 端有效)	JobSuc	PCIV 成功提交给 VGS 的 job 次数
	JobFail	PCIV 没有成功提交给 VGS 的 job 次数
	EndSuc	PCIV 结束 VGS job 成功的次数
	EndFail	PCIV 结束 VGS job 失败的次数
	MoveSuc	PCIV 添加 VGS move task 成功的次数
	MoveFail	PCIV 添加 VGS move task 失败的次数
	OsdSuc	PCIV 添加 VGS osd task 成功的次数
	OsdFail	PCIV 添加 VGS osd task 失败的次数
	ZoomSuc	PCIV 添加 VGS zoom task 成功的次数
	ZoomFail	PCIV 添加 VGS zoom task 失败的次数
	MoveCb	VGS move 回调的次数
	OsdCb	VGS osd 回调的次数
	ZoomCb	VGS zoom 回调的次数