



Hi3520 PCI 级联应用

Application Notes

文档版本 00B02

发布日期 2009-09-30

BOM编码 N/A

深圳市海思半导体有限公司为客户提供全方位的技术支持，用户可与就近的海思办事处联系，也可直接与公司总部联系。

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

版权所有 © 深圳市海思半导体有限公司 2009。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

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。



目 录

前 言.....	1
1 Demo板PCI级联操作指南.....	1-1
1.1 硬件环境准备	1-1
1.2 软件环境准备	1-1
1.3 从板启动	1-2
1.4 数据传输功能验证	1-3
1.4.1 驱动加载顺序	1-3
1.4.2 PCI数据传输功能验证.....	1-3
2 PCI基础知识简介.....	2-1
2.1 PCI简要概述.....	2-1
2.2 PCI地址映射.....	2-1
2.3 PCI DMA方式数据传输.....	2-2
2.4 PCI共享内存方式数据传输.....	2-2
2.5 PCI MCC消息应用	2-2
2.6 内存配置	2-2
3 PCI级联业务实现.....	3-1
3.1 视频预览流程	3-1
3.2 码流传输流程	3-1
3.3 解码图象显示流程	3-1
4 PCIV开发参考.....	4-1
4.1 PCIV概述	4-1
4.2 PCIV MPI参考	4-1
4.3 PCIV内核态功能简介	4-1
5 FAQ.....	5-1
5.1 主片和从片的Boot、Kernel、FS有何区别	5-1
5.2 文件系统如何制作	5-1
5.3 如何提高PCI传输效率.....	5-1
5.4 视频预览画面切换需要注意什么	5-1



5.5 从片到从片能不能直接使用DMA方式传输数据	5-1
5.6 从到从能不能直接通过窗口进行共享内存访问	5-2
5.7 从到从能不能直接使用MCC消息进行通信	5-2
5.8 PCI共享内存访问时，数据时与时错是什么原因	5-2
5.9 如何访问Hi3511 设备配置空间寄存器	5-2
5.10 如何通过PCI传递自定义消息	5-2
5.11 视频预览为什么要进行主从同步处理	5-2



前言

概述

本文介绍了 Demo 板通过 PCI 级联时的使用指南，同时介绍了 PCI 的基础知识、通过 PCI 级联的业务实现和 PCI MPI 接口函数等，可为用户在应用 PCI 级联时提供参考。

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
Hi3520 H.264 编解码处理器	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

约定

通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display” 格式	“Terminal Display” 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。



命令行格式约定

格式	意义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用加粗字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用斜体表示。
[]	表示用“[]”括起来的部分在命令配置时是可选的。
{ x y ... }	表示从两个或多个选项选取一个。
[x y ...]	表示从两个或多个选项选取一个或者不选。
{ x y ... } *	表示从两个或多个选项选取多个，最少选取一个，最多选取所有选项。
[x y ...] *	表示从两个或多个选项选取多个或者不选。

表格内容约定

内容	说明
-	表格中的无内容单元。
*	表格中的内容用户可根据需要进行配置。

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
2009-09-30	00B02	增加 1.5 小节。
2009-09-02	00B01	第一次发布。



1 Demo 板 PCI 级联操作指南

1.1 硬件环境准备

PCI 级联调测，需要准备两块 Hi3520 Demo 板，一个 12V 电源，两根串口线，网线以及视频输入输出线若干。两块单板分别工作在 PCI 主模式（主片）和 PCI 从模式（从片），从片的金手指插入到主片的 PCI 插槽上，确保视频输入输出接口位于外侧即可。只有主片连接电源线，从片不需要再接电源。分别连接好两块单板的串口线、网线以及视频输入输出线。

Hi3520 Demo 板默认工作在 PCI 主模式，如果要工作在 PCI 从模式需要将启动模式更改为 DDR 启动模式（拨码开关 S1 的第 8 位拨到 R60 端），断开电阻 R373，R483，R484。调测前请先确认 PCI 从片上，以下内容已全部更改：

- 拨码开关 S1 位于单板正面，金手指的上方。
- 电阻 373 位于单板正面，紧贴 Hi3520 芯片，朝向金手指的一面。
- 电阻 483 位于单板反面，与电阻 373 正对。
- 电阻 484 位于单板正面，与电阻 373 并列相隔 8 个电阻（印刷的字符被一个小孔隔断了，导致 R484 看着有点象 R404）。

详细的说明请参见《Hi3520 DMEB 原理图》和《Hi3520 Demo 单板用户指南》。



注意

- 从板的金手指插入到主片的 PCI 插槽后，需确保从片的视频输入输出接口位于外侧，否则就说明插反了。
- 只有主片需要连接电源线，从片由 PCI 供电，不能再接独立电源。

1.2 软件环境准备

Hi3520 芯片包括两个 ARM 核，单颗芯片就需要两套 boot、内核以及文件系统。这些镜像文件，可以在发布包的 pub/images 目录下获得。



首先按照《Hi3520 SDK 安装以及升级使用说明》中的说明，烧写主片上的 boot、内核以及文件系统；主片成功启动后，配置网络及 NFS。

从片启动也需要六个镜像文件，其中除主 ARM 的文件系统需要重新制作成 cramfs.initrd 格式的镜像外，其他五个文件可以与主片使用相同的镜像文件。将镜像文件存放在主片能够直接访问到的地方，例如存放在 nfs 服务器上的 pci-boot 目录。六个文件的名称，必须分别更改为引导程序默认的名称，如表 1-1 所示。

表1-1 启动文件清单

项目		文件名称	描述
主片	主 ARM	u-boot-hi3520v100_200M.bin	烧写到主片 Flash
		kernel-hi3520v100_full_release.img	烧写到主片 Flash
		rootfs-FULL_REL-Flash.jffs2	烧写到主片 Flash
	从 ARM	u-boot-hi3520v100_slave_200M.img	烧写到主片 Flash
		kernel-hi3520v100_full_release_slave.img	烧写到主片 Flash
		rootfs-SLV_FULL_REL.cramfs.initrd.img	烧写到主片 Flash
从片	主 ARM	u-boot-hi3520v100_200M.bin	与主片主 ARM 相同 改名为 u-boot.bin
		kernel-hi3520v100_full_release.img	与主片主 ARM 相同 改名为 kernel.img
		rootfs-FULL_REL-Inird.cramfs.initrd	需要重新制作 改名为 cramfs.initrd.img
	从 ARM	u-boot-hi3520v100_slave_200M.bin	与主片从 ARM 相同 改名为 u-boot-slave.bin
		kernel-hi3520v100_full_release_slave.img	与主片从 ARM 相同 改名为 kernel-slave.img
		rootfs-SLV_FULL_REL.cramfs.initrd.img	与主片从 ARM 相同 cramfs-slave.initrd.img

1.3 从片启动

启动从片前需要检查一下，下面各个步骤的配置是否正确。

步骤 1 主片的 PCI 设置为主模式，在主片的 Bootargs 启动参数中指定"pcimod=host pciclkssel=1"。例如 setenv bootargs 'mem=32M ...pcimod=host pciclkssel=1'，其他参数配置需参考《Hi3520 SDK 安装以及升级使用说明》。



- 步骤 2 主片系统引导成功后，可以使用 `cat /proc/bus/pci/devices` 命令查看从片是否被正常的识别到了。
- 步骤 3 主片插入 MMZ 模块。主片启动后，需要先插入 MMZ 模块，例如 `modprobe mmz mmz=ddr,0,0xe4000000,32M`
- 步骤 4 主片插入 PCI 引导模块。`modprobe pci_multi_boot path=/mount/pci_boot/`。其中 `path` 参数指定了从片启动所需镜像文件所在的目录，可以根据实际情况改动。
- 步骤 5 卸载 PCI 引导模块 `rmmmod pci_multi_boot`。模块插入成功后，从片应该已经被引导起来了，此时需要卸载引导模块，为后续工作清理环境。MMZ 模块也可以卸载掉（执行 `rmmmod mmz`），需要的时候再重新配置。
- 步骤 6 从片启动后，建议也配置好网络及 NFS，方便调测。
- 结束

1.4 数据传输功能验证

1.4.1 驱动加载顺序

主片上驱动模块加载顺序如下：

- 步骤 1 插入 PCI 硬件适配层模块 `modprobe pci_hwhal_host`。
- 步骤 2 插入 PCI 数据传输模块 `insmod hi3520_pci_trans.ko`。
- 结束

从片上驱动模块加载顺序如下：

- 步骤 1 插入 PCI 硬件适配层模块 `modprobe pci_hwhal_slave shm_phys_addr=0xC0000000 shm_size=0x1000000`；这里带的参数是 PCI 共享内存的起始地址和长度，可以根据业务需要修改成其他值；`shm_size` 最大可配置成 16M，最小不应小于 1M。这块内存中，最前面 1M 的空间用于 MCC 消息通信，必须保留；后面 15M 用于数据传输，可以根据实际业务配置其大小。
- 步骤 2 插入 PCI 数据传输模块 `insmod hi3520_pci_trans.ko`。
- 结束

1.4.2 PCI 数据传输功能验证

成功加载 PCI 驱动模块后，即可进行简单的功能验证。进入 `source\drv\hisi-pci\pcit` 目录，可以看到 `pcit_test_device.c` 和 `pcit_test_host.c` 两个测试程序，其中 `pcit_test_host.c` 在主片上运行，`pcit_test_device.c` 在从片上运行。在 `pcit` 上一级目录中的 `Makefile` 会编译这两个文件，并生成 `device` 和 `host` 两个可执行程序。将其复制到 `nfs` 可以访问的目录，例如前面放置从片启动文件的 `pci_boot` 目录。

这两个程序可以完成下面三个功能验证：

- 通过 Window 窗共享内存将数据从主片传送到从片



在主片运行 ./host 1； 在从片运行 ./device 1。

主片会向共享内存中写入 0xa5，从片应该能够读取并打印出 0xa5；读取完毕后，程序自动退出。

- doorbell 中断互发测试

在主片运行 ./host 2； 在从片运行 ./device 2。

主片循环向共享内存写入一个计数值，计数每增加一次就会向从片发送一次 doorbell 中断。从片接收到中断后，从共享内存中读取计数值并打印出来；再向主片回复一个 doorbell 中断。程序执行后，在主从片上都会打印出递增的计数值。如果要停止测试，需要 CTRL+C 结束。

- 通过 DMA 方式将数据从从片传送到主片

在主片运行 ./host 3； 在从片运行 ./device 3。

从片将递增的计数值先写入一块临时缓存中，再通过 PCI DMA 发送到主片的 DDR 中。主片接收到 PCI DMA 事件后，会从约定的缓存中读取计数值。在主从片上都会打印出递增的计数值。如果要停止测试，需要 CTRL+C 结束。

这个测试程序中使用的缓存都是约定好的，使用宏定义 TEST_DST_ADDR 和 TEST_SRC_ADDR 标识，运行测试前可能需要根据单板的内存分布，重新调整这两个值。

1.5 PCIV 功能验证

1.5.1 PCI 从片文件系统制作

进入发布包的 mpp_master/ko/ 目录执行 mkimg_pci 脚本。

此脚本包含以下操作：

- 步骤 1 将启动从片需要的 6 个原始镜像文件从 pub/images 目录中拷贝到 pci_boot 目录下。
- 步骤 2 将从片文件系统的压缩文件 rootfs-FULL_REL-Flash.tgz 从 pub/tarball 中拷贝到当前目录并解压。
- 步骤 3 将从片需要的 ko 和应用程序拷贝到文件系统的/root/目录下。
- 步骤 4 执行 mkimg.rootfs 制作 PCI 从片的文件系统镜像。
- 步骤 5 将文件系统镜像拷贝到 pci_boot 目录。

----结束

1.5.2 Sample 程序执行

- 步骤 1 主片进入 mpp_master/ko 目录，执行脚本 load_pci_host.sh，启动 PCI 从片，并加载 PCI 相关 KO 和 MPP 相关 KO。
- 步骤 2 从片进入/root/mpp_ko/ 目录，执行脚本 load_pci_device.sh，加载 PCI 相关 KO 和 MPP 相关 KO。



- 步骤 3 从片进入/root/ 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 4 主片进入 mpp_master/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 PCIV 视频预览业务。
- 步骤 5 如果需要停止 Sample 程序，可以在主片上输入两次回车键，主片会发消息给从片销毁相关业务，并退出本片程序；从片需要按 CTL+C 退出。

----结束



2 PCI 基础知识简介

2.1 PCI 简要概述

PCI 是外围设备互连 (Peripheral Component Interconnect) 的简称, 作为一种通用的总线接口标准, 它在目前的计算机系统中得到了非常广泛的应用。PCI 总线的时钟频率一般使用 33MHz, 在 32bit 系统中, 理论极限速度可以达 132Mbytes/s ($32 \times 33\text{M}/8$); 但 PCI 无法一直维持在峰值传输的状态, 一般只能保持在 40~50% 的传输效率, 即 40~60Mbytes/s 的速度。如果需要更高的传输速度, 也可以使用 66MHz 的时钟频率, 只是频率越高, 对硬件设计的要求也会越高。

PCI 设备上有三种地址空间: PCI 的 I/O 空间、PCI 的存储空间和 PCI 的配置空间。CPU 可以访问 PCI 设备上的所有地址空间, 其中 I/O 空间和存储空间提供给设备驱动程序使用, 而配置空间则由 Linux 内核中的 PCI 初始化代码使用。内核在启动时负责对所有 PCI 设备进行初始化, 配置好所有的 PCI 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 PCI 设备, 以及这些设备的参数和属性。与此对应 PCI 总线命令可以分为 I/O 操作命令、Memory 操作命令、配置访问命令。



说明

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。

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。

PCI 协议比较复杂, 关于它的详细说明, 请查阅有关 PCI 规范的资料。这里不去讲述这些复杂的协议, 只关注业务应用中最常用的一些知识。

2.2 PCI 地址映射

待补充。



2.3 PCI DMA 方式数据传输

待补充。

2.4 PCI 共享内存方式数据传输

待补充

2.5 PCI MCC 消息应用

待补充。

2.6 内存配置

待补充。



3 PCI 级联业务实现

3.1 视频预览流程

待补充。

3.2 码流传输流程

待补充。

3.3 解码图象显示流程

待补充。



4 PCIV 开发参考

4.1 PCIV 概述

待补充。

4.2 PCIV MPI 参考

待补充

4.3 PCIV 内核态功能简介

待补充。



5 FAQ

5.1 主片和从片的 Boot、Kernel、FS 有何区别

待补充。

5.2 文件系统如何制作

待补充。

5.3 如何提高 PCI 传输效率

待补充。

5.4 视频预览画面切换需要注意什么

待补充。

5.5 从片到从片能不能直接使用 DMA 方式传输数据

待补充。



5.6 从到从能不能直接通过窗口进行共享内存访问

待补充。

5.7 从到从能不能直接使用 MCC 消息进行通信

待补充。

5.8 PCI 共享内存访问时，数据时对时错是什么原因

待补充。

5.9 如何访问 Hi3511 设备配置空间寄存器

待补充。

5.10 如何通过 PCI 传递自定义消息

待补充。

5.11 视频预览为什么要进行主从同步处理

待补充。