

《计算机网络》实验报告

年级、专业、班级	21 级计卓 1 班		姓名	李宽宇
实验题目	Ethernet 多端口交换机设计与开发			
实验时间	11 月 4 日	实验地点	DS3 304	
实验成绩		实验性质	<input type="checkbox"/> 验证性 <input type="checkbox"/> 设计性 <input checked="" type="checkbox"/> 综合性	
<p>教师评价：</p> <p><input type="checkbox"/>算法/实验过程正确； <input type="checkbox"/>源程序/实验内容提交 <input type="checkbox"/>程序结构/实验步骤合理；</p> <p><input type="checkbox"/>实验结果正确； <input type="checkbox"/>语法、语义正确； <input type="checkbox"/>报告规范；</p> <p>其他：</p> <p>评价教师签名：</p>				
<p>一、 实验目的</p> <ul style="list-style-type: none">● 了解硬件系统开发流程。● 熟悉 Vivado 开发平台。● 深入了解交换机的工作原理，以及如何设计与实现一个真实的交换机系统，同时验证交换机的基础功能。				
<p>二、实验项目内容</p> <p>利用电脑和 Xilinx ZBOX 板卡实现 Ethernet 交换机核心功能。具体功能包括如下：</p> <ul style="list-style-type: none">● ARM 处理器模块功能实现，实现 CPU 处理功能● MAC 模块功能实现，实现网络数据的接收发送基本功能● DMA 模块功能实现，实现 MAC 模块与存储器间的数据交换● AXI 互联模块功能实现，实现不同 Ethernet 端口间通信互联● 交换机功能模块实现，实现交换机端口的状态管理维护、地址转发表的查询与管理维护、基于源地址学习的转发表学、以及帧收发等功能● 交换机配置软件实现。实现交换机的管理配置和操作命令				

报告创建时间：

三、实验过程或算法（源程序）

3.1 交换功能 IP 核完善封装，更新交换功能 IP 核

- 1) 打开已解压的 zynq_switch 工程
- 2) 在 zynq_switch_i-zynq_switch 模块中右击 packet_pipeline_v1_0, 选中 Edit in IP Packager 选项。
- 3) 选择 IP 核编辑工程的地址以及工程名称, 点击 OK, 打开 IP 核编辑工程界面。
- 4) 修改 eth_parser.v, mac_cam_lut.v

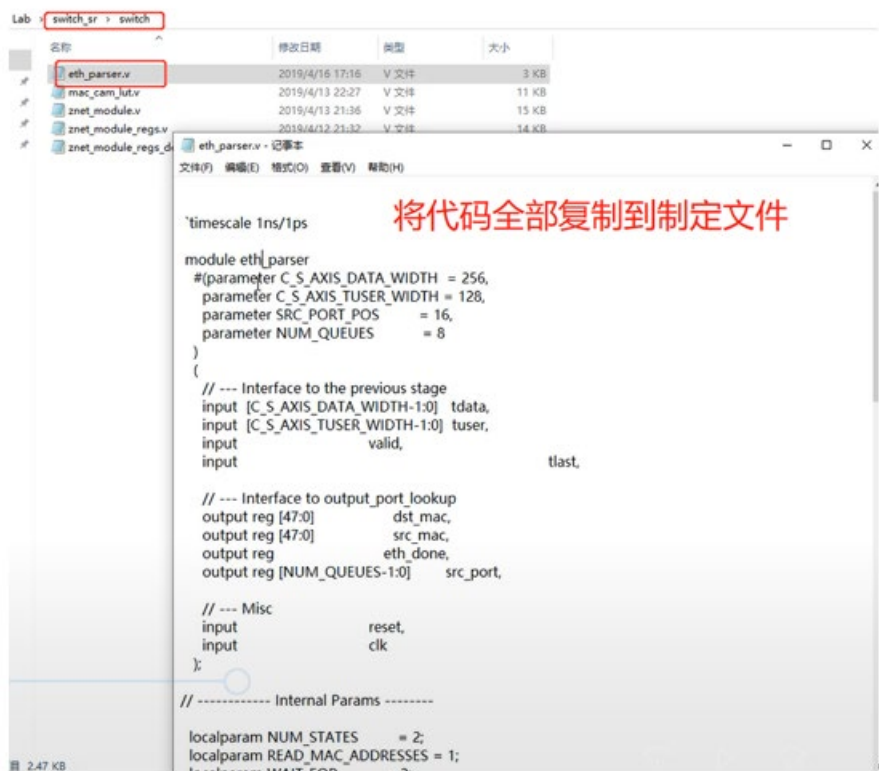


图 1

- 5) 双击 Source 界面中 mac_cam_lut.v, 并右击选择 Add Sources.. 选项, 补充完整 cam.v 模块及其子模块, 使之实现功能: 根据源 MAC 地址实现地址自学习, 根据目的 MAC 地址获取输出端口。并点击保存。
- 6) 点击工程左侧 Project Manager 下的 Package IP, 依次点击 Packaging Steps 中的各个部分, 查看 IP 核配置属性, 点击 Review and Package, 打开 Review and Package 界面, 点击 “Re_Package IP”, 对 IP 进行再次封装。



图 2

7) 回到 zynq_switch 实验工程，点击工程界面中 IP 需要更新的提示中“Report IP Status”，或者点击菜单栏 Tools->Report->Report IP Status，查看 IP status。

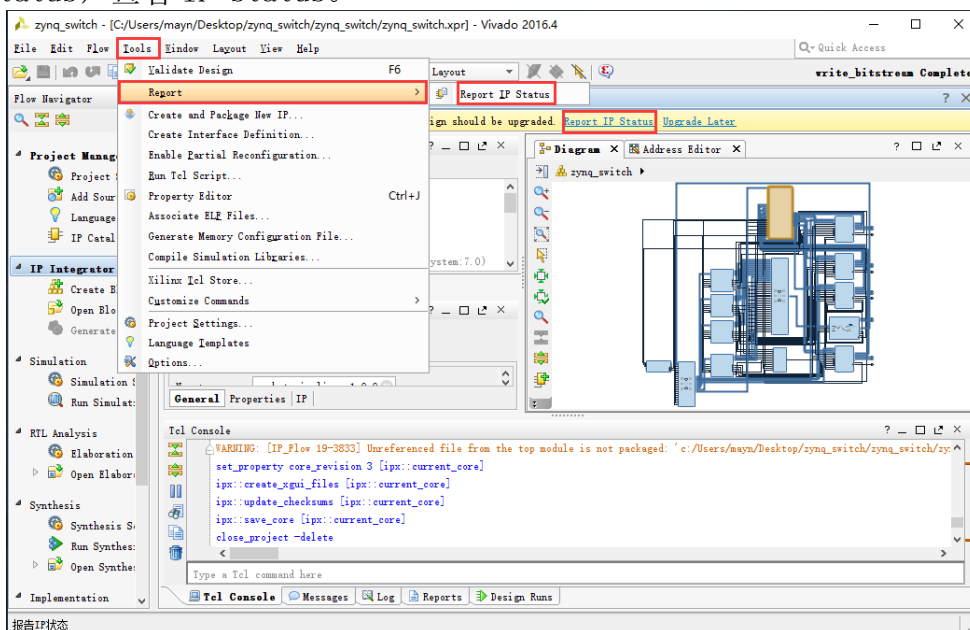


图 3

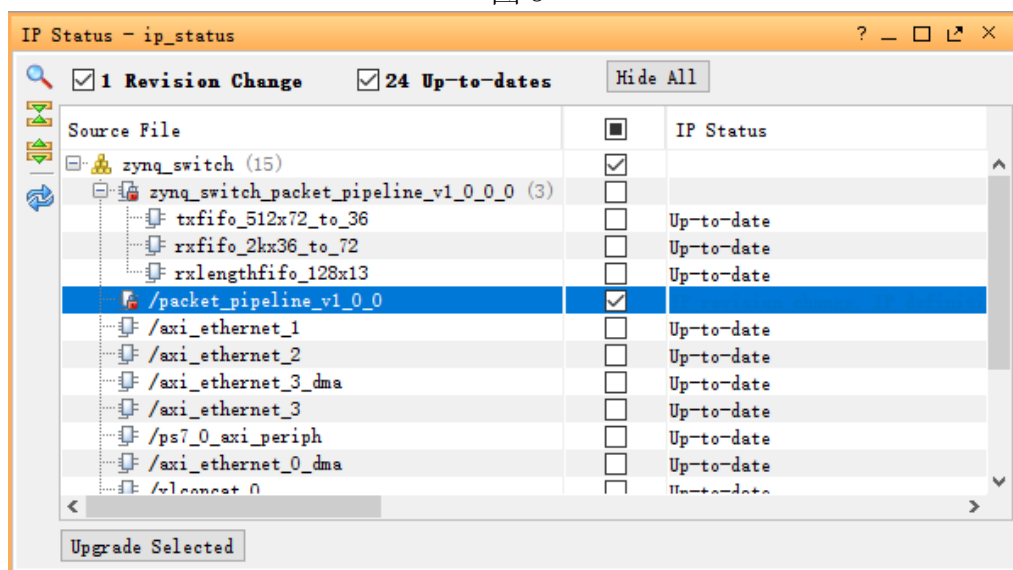


图 4

8) 在 IP status 窗口选中 packet_pipeline_v1_0_0，点击“Upgrade Selected”对工程中的交换功能 IP 核进行更新，等待更新完毕，点击 OK。
 9) 在 Generate Output Products 窗口点击 Generate，如果出现警告，可暂时忽略。该警告是因为交换功能 IP 中有些接口在本实验中没有被用到。
 10) 完成整个硬件设计后，点击“Generate Bitstream”按钮，在弹出的

对话框中点击 OK，综合、实现、布局布线生成 FPGA 配置文件。

11)完成“Generate Bitstream”后，需要导出 FPGA 配置文件：

zynq_switch.bit。点击菜单“File”，选择“Export”，选择“Export Bitstream File”。选择预先下载的 linux 文件夹路径，输入File name: zynq_switch，点击 Save 按钮。

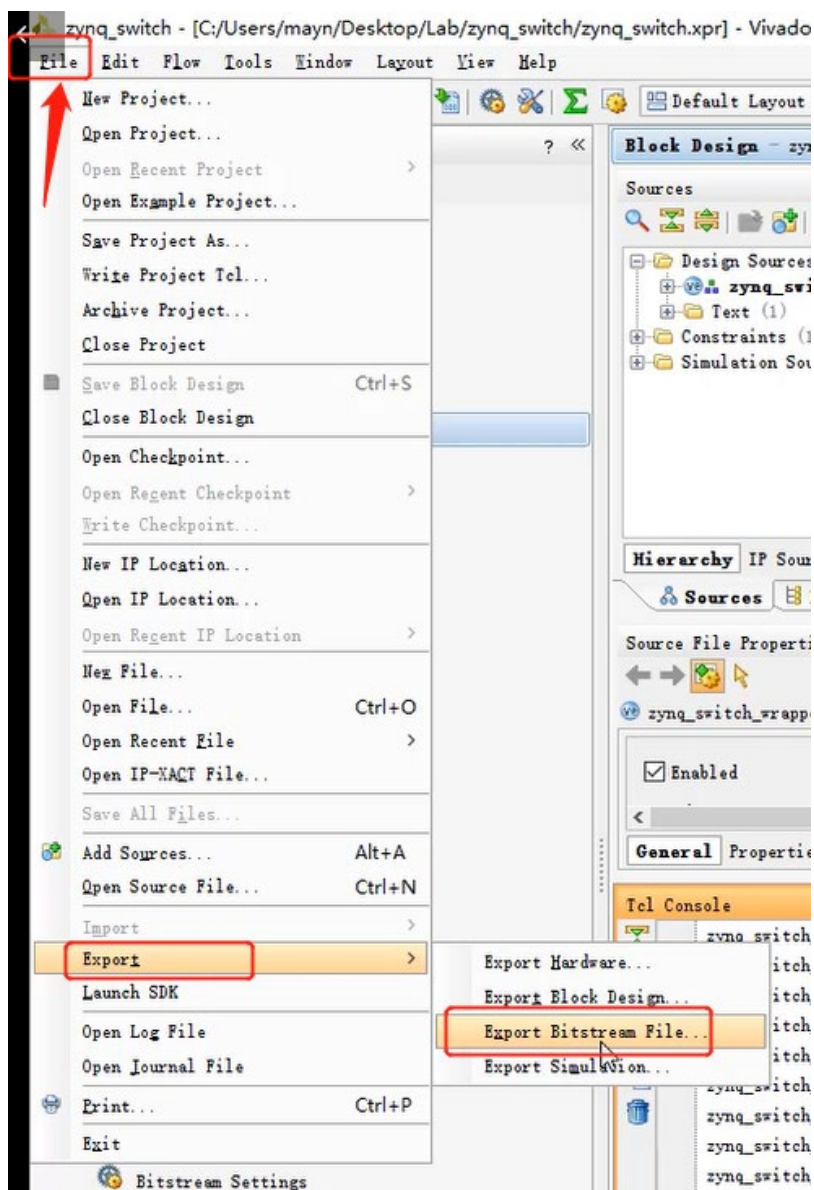


图 5

3.1 结果



图 6

3.2 创建 Boot.bin 启动文件

1) 打开 Xilinx SDK 工具，点击“Xilinx Tools”菜单选项，选择“Create Boot Image”

2) 在弹出的“Create Boot Image”界面，在 Output BIF file path 中选择生成文件存放的路径名（不能涵盖中文路径），这里选择 linux 文件夹。

3) 按顺序添加三个文件：zynq_fsbl.elf, zynq_switch.bit, u-boot.elf 在 Boot image partitions 里点击 Add，在弹出的 Add partition 对话框里，看到 Partition type 为 bootloader 时，点击 File path 按钮，选中 linux 文件夹下的 zynq_fsbl.elf，点击 OK。

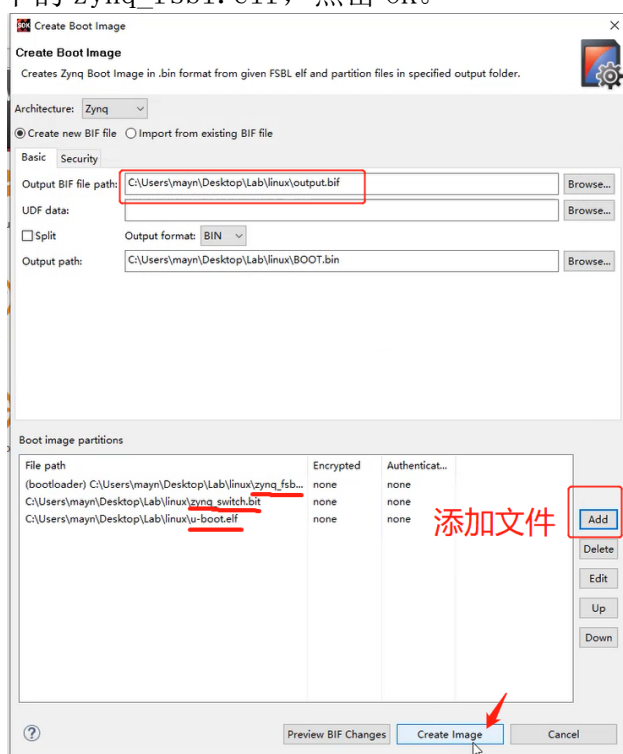


图 7

4) 点击“Create Image”按钮，生成 Boot.bin 文件，在 linux 文件夹下面可以看到 BOOT.bin 文件。

3.2 结果



图 8

3.3 交换机配置软件开发

创建编译 petalinux 工程

1)在虚拟机新建文件夹 zynq_switch，并进入到该文件夹目录下
打开虚拟机（密码 11）

```
mkdir zynq_switch
cd zynq_switch
```

2)在使用 Petalinux 工具之前应该首先指明相应的环境变量，在 Linux 终端中执行以下命令：

```
source /opt/pkg/petalinux/settings.sh
```

3)将 “*_hw_platform_0” 这个文件夹，复制到安装 Petalinux 的 Linux 系统中的 zynq_switch 文件夹中。

4)创建一个 petalinux 工程：

```
petalinux-create --type project --template zynq --name zbox
cd zbox
```

5)给 petalinux 工程导入硬件工程描述：进入 znet 工程文件夹，通过

```
petalinux-config
--get-hw-description=" ../zynq_nic_wrapper_hw_platform_0"
```

6)文件夹路径，命令 cd 导入 HDF 文件。

7)进入 zbox 工程文件夹，配置内核： **petalinux-config -c kernel** (10min 左右)

8)修改 Device tree 文件：

```
project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi
```

9)在 zbox 工程路径下创建 switch-config 应用：

```
petalinux-create -t apps --template c --name switch-config - enable
```

10)zbox/project-spec/meta-user/recipes-apps/switch-config/files/switch-config.c 把提供的 switch-config.c 文件中的代码复制到上述文件，保存。

11)在 zbox 工程路径下创建 router-config 应用：

```
petalinux-create -t apps --template c --name router-config --enable
```

12)znet/project-spec/meta-user/recipes-apps/router-config/files/router-config.c，把提供的 router-config.c 文件中的代码复制到上述文件，保存。

13)在 zbox 工程路径下，编译应用：

petalinux-build -v

14)在 zbox/images/linux 路径下，可以看到：image.ub，上述两个应用已经封装进 image.ub 文件，将该文件拷贝到 linux 文件夹下。

3.3 结果

image.ub	2023/11/7 17:45	UB 文件	9,271 KB
----------	-----------------	-------	----------

图 9

3.4 交换机功能验证

至此，已经得到以下的文件

名称	修改日期	类型	大小
BOOT.bin	2023/11/7 16:38	BIN 文件	4,460 KB
image.ub	2023/11/7 17:45	UB 文件	9,271 KB
output.bif	2023/11/7 16:38	BIF 文件	1 KB
u-boot.elf	2019/4/14 21:12	ELF 文件	3,142 KB
zynq_fsbl.elf	2019/4/14 21:12	ELF 文件	182 KB
zynq_switch.bit	2023/11/7 16:31	BIT 文件	3,951 KB

图 10

1) 创建 SD 卡启动系统

- 需要确保 SD 卡的 FAT 分区格式
- 若重新修改硬件设计，只需要替换 SD 卡 FAT 分区的 BOOT.bin 文件。
- 完成之后，按照如下步骤操作：
 - 把烧写好系统的 Micro SD 卡插入卡槽；
 - 把 TypeC 连线接上；
 - 将 PS 网口接入局域网（电脑网口）。

● 系统启动正常：两个 LED 灯点亮，绿色灯指示电源正常，蓝色灯指示 Linux 系统启动正常。

2) 打开串口调试工具 putty，配置如下

- 端口号选择：COM4（可在计算机“设备管理器”中确定 COM 端口号）；
- 比特率设置：115200；
- 连接类型选择：Serial（中文版选择“串口”）。

3) 交换机配置软件

● 交换机配置使用 switch-config 命令，可以使用该命令读写交换机寄存器，查找表。

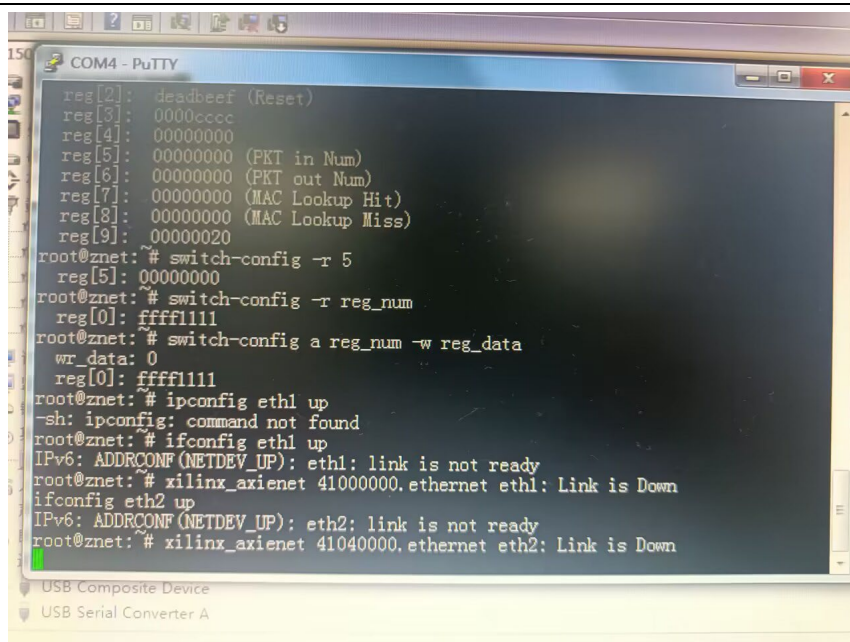


图 11

- 配置交换机查找表

```
switch-config -a reg_num -w reg_data
```

4) 验证实验操作步骤

- 步骤 1 SD 卡拷贝，物理连接。

- 步骤 2 启动开发板的相应端口。

启动命令: ifconfig ethX up (ethX 是 eth1、eth2、eth3、eth4, up 接网线的对应端口)

- 步骤 3 设置两主机 ip。(两台主机在同一网段即可), 分别为 192.168.0.1/24, 192.168.0.2/24

- 步骤 4 在 PC 网口适配器属性-配置-高级 里面, 把速度固定在 100M 全双工, 重启计算机。

- 步骤 5 执行 ping 命令, 验证两主机之间是否能互 ping。

- 步骤 6 通过 switch-config 命令, 查看交换机的输入和输出数据包数目。

3.4 结果

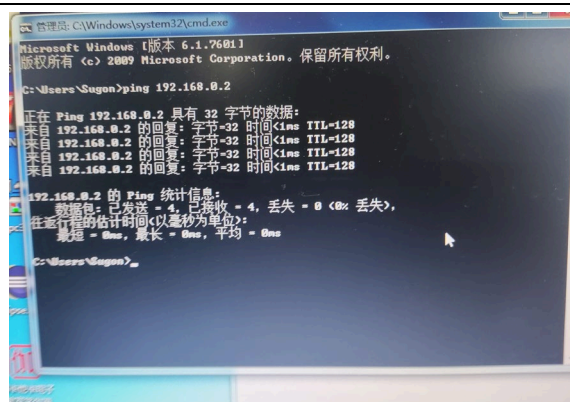


图 12: 3.4 验证两主机之间是否能互 ping

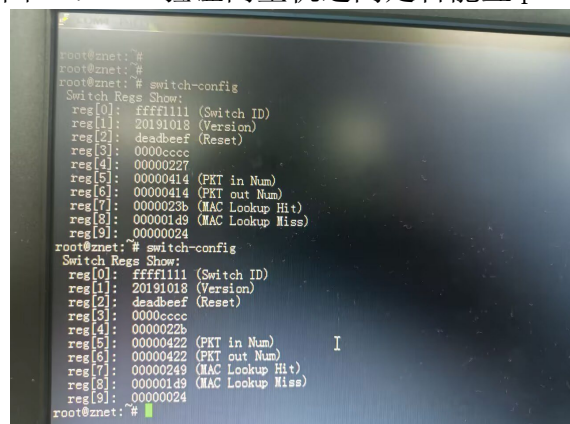


图 13: 3.4 查看交换机的输入和输出数据包数目

四、实验结果及分析和（或）源程序调试过程

1.实验结果

3.1IP 核完善封装、更新，生成 bit 流文件结果

zynq_switch.bit	2023/11/7 16:31	BIT 文件	3,951 KB
-----------------	-----------------	--------	----------

图 14

3.2 创建 Boot.bin 启动文件结果

BOOT.bin	2023/11/7 16:38	BIN 文件	4,460 KB
----------	-----------------	--------	----------

图 15

3.3 交换机配置软件开发结果

image.ub	2023/11/7 17:45	UB 文件	9,271 KB
----------	-----------------	-------	----------

图 16

3.4 交换机功能验证结果

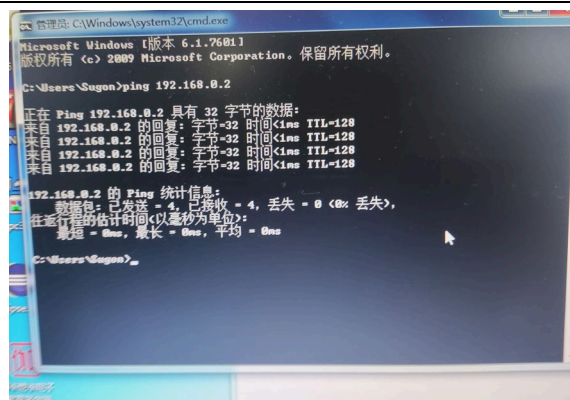


图 16: 3.4 验证两主机之间是否能互 ping

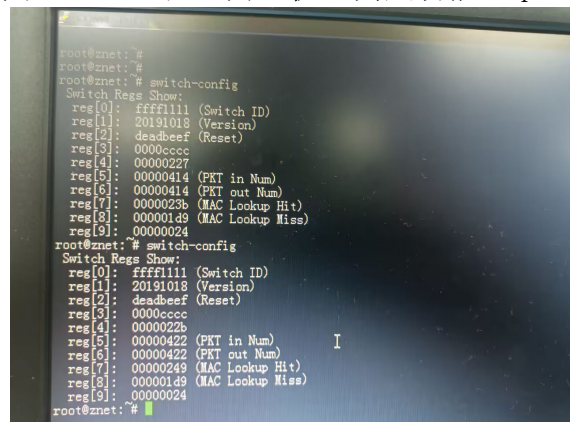


图 17: 3.4 查看交换机的输入和输出数据包数目

2.总结与体会

通过实验，我深入了解了硬件系统开发流程，掌握了 Vivado 和 Petalinux 工具的使用，同时对交换机的各个功能模块有了深入的认识。

通过对数据链路层、网络层和交换机内部组件的详细学习，我对网络设备的运作机制有了更深刻的理解。

在设计与实现交换机系统的过程中，我学到了如何使用硬件描述语言 Verilog，来构建和连接硬件设计模块。

总的来说，这次实验不仅加深了我对硬件系统开发流程的理解，还为我提供了宝贵的实践经验。