

实验 6. 基于 NetFPGA 平台的路由实验

一. 实验目的

1. 熟悉 NetFPGA 的运行环境
2. 熟悉 Reference Router 工作原理
3. 掌握使用 Reference Router 进行实验

二. 实验简介

1. NetFPGA 介绍:

NetFPGA 是斯坦福大学高性能网络研究小组开发的一款开放式网络研究平台，其构建在 CentOS 操作系统下的开发环境，为人们提供了开源的软件与硬件代码，使得研究人们能非常方便地复用前人开发的模块，在其上验证网络协议与算法或开发网络设备等，图 1 为整个 NetFPGA 开发平台的整体框架图。

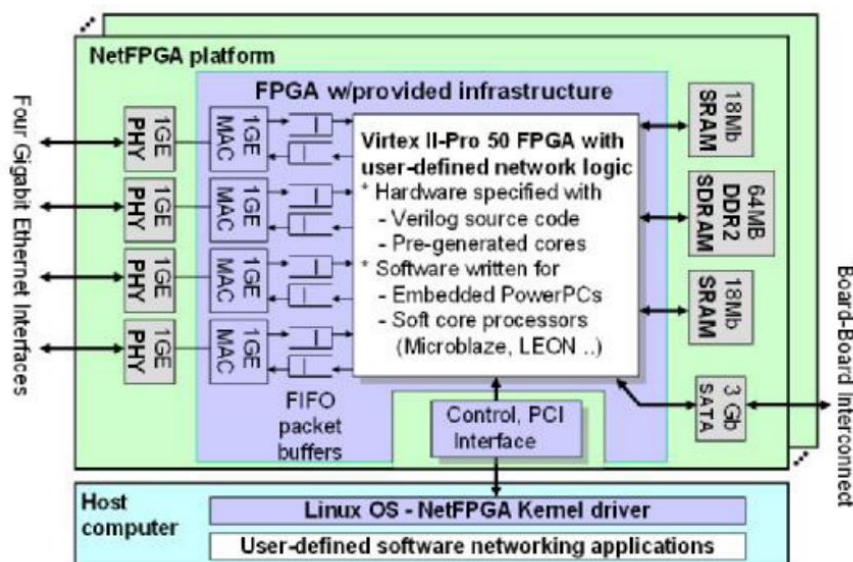


图 1 NetFPGA 整体框架图

整个 NetFPGA 开发平台由软件与硬件构成，硬件部分包括开发板卡以及基于该板卡的使用 verilog 语言描述的一个可综合的参考设计。软件部分包括基于 LINUX 操作系统的设备驱动程序以供用户能够将 NetFPGA 板卡识别为一普通的 PCI 设备以提供上位机与硬件的通信，以及特定工程对应的相关软件。NetFPGA 上配备有 PCI 插槽，且直接由 PCI 接口进行供电，其开发需要辅以个人计算机使用，其板卡外形如右图所示。在板卡上有两块 FPGA 芯片，其中的 Spartan 3E 芯片内部固化了一个 PCI 接口控制器的 IP 核，用做上位机与主控核心的 V2Pro 芯片的中间媒介，上位机通过 PCI 总线再经由



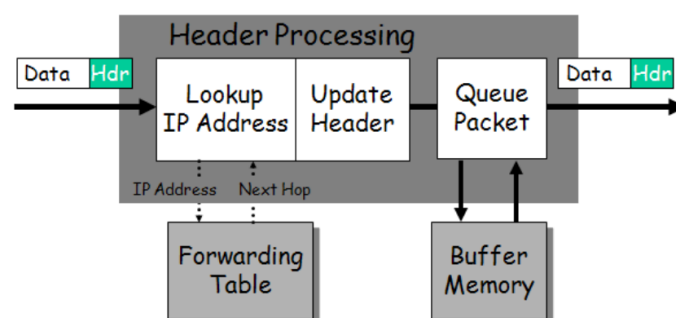
Spartan 芯片对主 FPGA 进行编程配置或内部寄存器的读写操作。大多数的网络设备都可以分为数据平面和控制平面，其中，控制平面通过特定逻辑决策如何处理数据包，数据平面执行控制平面的决策，处理数据包。对应到 NetFPGA 平台，通过 Verilog 编程可以在 NetFPGA 板卡上实现数据平面的功能，通过在上位机上编程可以实现控制平面的功能。因此，研究人员能够根据需求，利用 NetFPGA 实现不同的网络功能以验证和测试自己的设计，极大的增加了网络研究的便捷性。

2. CentOS 简介：

CentOS (Community ENTerprise Operating System) 是 Linux 发行版之一，它是来自于 Red Hat Enterprise Linux 依照开放源代码规定释出的源代码所编译而成。由于出自同样的源代码，有些要求高度稳定性的服务器以 CentOS 替代商业版的 Red Hat Enterprise Linux 使用。两者的不同，在于 CentOS 并不包含封闭源代码软件。

3. Reference Router 简介

NetFPGA 小组在发行板卡及驱动的同时，也提供了一些基于 NetFPGA 的参考设计，包括网卡、路由器、OpenFlow 交换机等。本次实验我们将使用参考路由器 (Reference Router)，利用 NetFPGA 和它对应的上位机一起实现路由器的功能。路由器的功能从逻辑上可以分为两个平面：数据平面和控制平面。数据平面负责根据路由表转发数据包；控制平面负责运行路由协议，与其他路由器交互获取网络拓扑，并动态生成路由表。对应到实验平台，通过 Verilog 编程，在 NetFPGA 板卡中实现数据平面的功能。数据包在 NetFPGA 板卡中的处理流程如下图所示。从端口接收到数据包后，在路由表中查找输出端口，之后修改数据包包头，如修改目的 MAC 地址，更新 TTL、校验值等，最后将数据包送到输出缓冲队列，并进一步通过输出端口发送。控制平面的路由协议运行在上位机上 (PC 机上)。本次实验使用的协议为简化版的 OSPF 协议，通过 C 语言实现。数据平面的路由表存储在特定的寄存器中，NetFPGA 驱动提供了读写板卡中寄存器的接口，控制平面利用该接口更新板卡中的路由表。



三. 实验内容

1. 熟悉 NetFPGA 运行环境

(1) 查看 NetFPGA 端口配置

进入系统，选择 CentOS 系统，用户名 root，密码 client

打开终端：在桌面右键点击桌面，选择在终端中打开。

在输入命令 `ifconfig`，查看各端口配置，NetFPGA 端口为 `nf2c0-nf2c3`。

若无显示，则输入 `ifconfig -a` 以显示所有可用端口信息。

使用命令 `ifconfig nf2c* up` 来激活所有的 NetFPGA 接口,*代表数字 0-3。

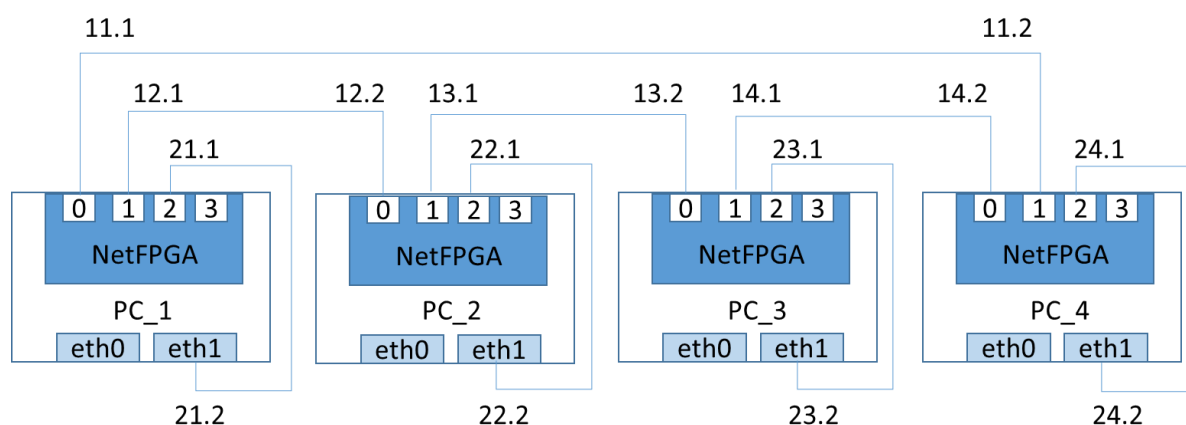
(2) NetFPGA 可用命令

以 `nf_`开头的命令是 NetFPGA 软件包提供的，可用于测试，下载等功能。常用命令：

- ✧ `nf_info`:查看 NetFPGA 版卡信息
- ✧ `nf_download`:将 Bit 文件下载到 NetFPGA 中。
- ✧ `nf_register_gen.pl`: 生成项目所需寄存器文件
- ✧ `nf_run_test.pl`: 使用 Modelsim 进行时序仿真。
- ✧ `nf_regress_test.pl` 进行 `regress test` 来测试生成的 bit 文件功能

2. 利用 NetFPGA 平台进行路由实验

本次实验需要第 1-2、3-4、5-6、7-8 分别组成一个大组，每个大组协同完成（由于原来只有 7 个小组，为了使人数均衡，我们从其他小组选出了 3 名同学加到第 1-2 组，新的分组名单见网络学堂）。每个小组内有 2 块 NetFPGA 板卡，每个 NetFPGA 有四个端口，按照下图将 NetFPGA 连成环形拓扑，并将主机的一个网卡与 NetFPGA 的一个端口相连，作为客户端使用。通过在 NetFPGA 中下载路由器的比特文件，将其配置成路由器。运行 OSPF 路由协议，学习路由表，使得各客户端之间能够互相连通。通过相关实验，观察路由表，测试路由时间，学习 Reference Router 代码。



3. 实验步骤

3.1 路由器数据平面配置

编译好的路由器数据平面比特文件为：

```
/root/netfpga/bitfiles/reference_router.bit
```

将该文件下载到 NetFPGA 板卡中，实现数据平面功能。具体步骤如下：

(1) 点击桌面，选择在终端中打开。

(2) 初始化 NetFPGA 板卡: `cpci_reprogram.pl -all`

(3) 下载 reference_router 的 bit 文件:

```
nf_download /root/netfpga/bitfiles/reference_router.bit
```

3.2 路由器控制平面配置

控制平面对应的工程为 scone, 代码放在 `/root/netfpga/projects/scone/sw` 下, 启动脚本为 `./scone`。在启动之前, 需要首先配置路由器端口的 IP 和 MAC 地址。具体步骤如下:

(4) 通过配置文件 (cpuhw) 配置路由器端口信息

```
gedit /root/netfpga/projects/scone/sw/cpuhw
```

在打开的文件中编辑对应的 NetFPGA 端口信息, 格式为:

eth0(端口号) 192.168.1.1(端口 IP) 255.255.255.0(子网掩码)00:00:00:00:00:01(端口 MAC)

将所有端口(eth0—eth3)按照拓扑中所给出的地址配置好, 同时保证相连的两个小组内的 MAC 地址不冲突。编辑完成后保存并关闭

注意: cpuhw 中, eth0 对应 nf2c0 即板卡 0 号端口, eth1 对应 nf2c1 即板卡 1 号端口。面向 netfpga 板卡时, 0 号端口为最左侧端口。

(5) 运行 OSPF 协议 (SCONE 程序)

```
cd netfpga/projects/scone/sw/
```

```
./scone
```

注意: 运行 scone 之前需保证所有 nf2c* 在激活状态 (即 ifconfig, 否则会提示 “write: Network is down”)

3.3 客户端配置

(6) 配置本机端口地址, `ifconfig eth1 192.168.*.*`

注意: 主机中与 NetFPGA 板卡相连网卡的编号可能为 eth0, 也可能为 eth1, 每个主机都可能不同。配置好 IP 地址后, 可以通过 ifconfig 命令的输出, 查看该网卡是否收到数据包进行区分, 与 NetFPGA 相连的网卡会持续收到数据包。

(7) 添加本机默认网关 (请思考: 网关 IP 是什么)

```
route add default gw <IP of gateway>
```

添加本机网关后, 可以 ping 通网关, 如果 ping 不通, 请重新执行步骤 1-5, 并再次尝试 ping 网关。

3.4 路由实验

(8) 通过 GUI 图形界面观察路由表结构:

新打开一个终端, 并在终端中输入命令

```
cd netfpga/lib/java/gui/
```

```
./router.sh
```

在弹出的界面中会有端口信息, 路由表信息和 ARP 信息。:

(9) 启动视频服务器 (PC_1 作服务器), 并使用客户端 (PC_2, PC_3) 使用该服务器

服务器端输入以下命令, 运行流媒体服务器:

```
/opt/real/Bin/rmserver /opt/real/rmserver.cfg
```

客户端打开 Realplayer:

应用程序—>影音—>realplayer->打开 URL,

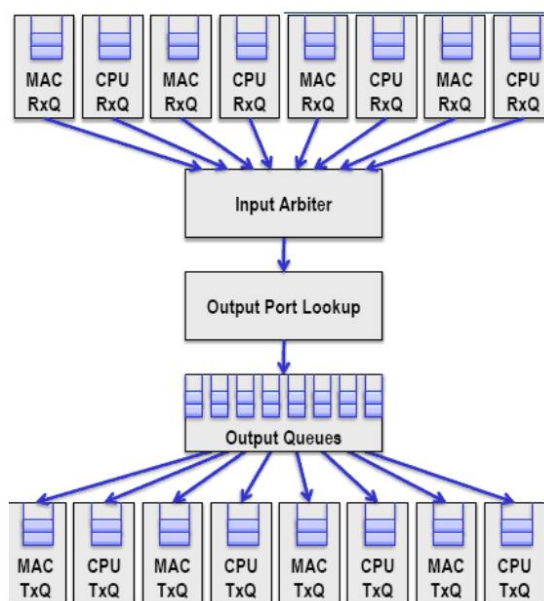
输入地址 `rtsp://<服务器的 IP 地址>/realvideo10.rm`

注：如果提示证书错误，将系统时间调整为 2013 年即可

(10) 测试 OSPF 收敛时间

- 每台主机打开两个新的命令行窗口，ping 另外三台（或两台）主机
- 断开 NetFPGA 之间的某条连接
- 观察 ping 包通断情况及路由表的状态变化
- 如果没有变化，恢复该连接，断开另一条连接呢？

(11) 观察 reference_router 模块结构



reference_router 模块结构见上图。保持 scone 打开,观察 GUI 界面中的统计信息

(statistics).在 GUI(detail)界面中双击某一模块,查看其统计信息,调整模块参数,观察统计信息是否有改变。

3.5（开放性任务）了解数据平面 reference_router 和控制平面 scone 代码结构

(12) 了解数据平面 reference_router 代码结构

以 Input_Arbiter 为例,在终端中输入命令:

```
gedit /root/netfpga/lib/verilog/core/input_arbiter/rr_input_arbiter/src/input_arbiter.v
```

阅读 Input_arbiter 的代码,它实现了一个轮询获取八个队列数据的功能,结合上述结构框图来理解 Input_arbiter 的详细功能,分析该部分的状态机。

(13) 了解控制平面 scone 程序代码结构

scone 程序代码放在 `/root/netfpga/projects/scone/sw/` 路径下,OSPF 协议的主要代码为 `./or_pwospf.c`,找到并分析触发更新路由表的函数,根据该部分代码分析更新路由表的过程。

四. 实验验收

1. 组网 IP 配置不限于使用上文的 IP 配置,可以自行分配网段。根据实际分配情况,绘制网络拓扑结构图(与上文提到的网络拓扑图类似,标明每个接口实际配

置的 ip、mac 地址以及接口连接情况)。

2. 演示浏览视频的过程，以证明组网正常工作
3. 根据 3.10 的内容，演示 OSPF 收敛过程，解释 ping 的过程及路由表变化
4. （选做）根据 3.10 的内容，针对每台机器，由 GUI 界面提供的路由表，绘制其断开连接之前的最小生成树和断开连接之后的最小生成树，并以此解释网络拓扑变化导致的路由变化
5. （开放性题目，选做）根据相关代码，阐述某模块的功能实现方式
6. 回答助教的问题

五. 实验报告

本次实验全部当堂完成，不需要写实验报告。