Network Systems Architecture

ECPE 293B – Spring 2011

HW Project 1 Report

Learning Ethernet Switch

Vikas Azad, Christopher Bennett, Michael Bruckel & Alberto Sanchez

University of the Pacific

Project Report Submitted On: February 26, 2011

Project Report Due On: February 25, 2011

1. Objectives

Each team will create a 4-port learning Ethernet Switch on the NetFPGA platform. The main purpose is for the students to increase their familiarity with the hardware platform and simulation environment by creating a simple network device. This experience is a fundamental stepping stone towards building an IP router later in the semester. In order to complete the project, the 4-port Ethernet NIC design will be extended in a new project to create a 4-port learning Ethernet switch. Most of the changes will take place in the output\_port\_lookup module.

1. Theoretical Background

A learning Ethernet Switch is a computer networking device that connects multiple devices with each other. This connection takes place within a Local Area Network (LAN) only. In contrast to Ethernet hubs which do not manage any of the traffic and broadcast out all other ports, switches receive data packets and inspect them to figure out source and destination. Switches store MAC addresses of each host and its corresponding port in a table. This table can then be used to forward packets based on MAC addresses. The switch broadcasts the packet out all the other ports if the destination is unknown (switch does not have an entry for the destination MAC address in the table). On the other hand, the switch will forward the packet out a single port if destination mapping exists. These features will be present in the Ethernet Switch developed in this project.

The Ethernet Switch will be developed on top of Stanford’s NetFPGA. The NetFPGA is a complete network hardware platform implemented with Field Programmable Gate Array logic. This PCI card contains a large Xlinx FPGA, 4 Gigabit Ethernet ports, Static RAM, and DDR2 RAM. The NetFPGA project is aimed at helping students learn how to build switches and routers in a classroom environment.

1. List of Resources

* NetFPGA System
* Server 2 Private Machine: netfpga02
* ecs-network
* 4-port Ethernet NIC implementation

1. Switch Design

The following sections describe the features of the Ethernet Switch developed in this project:

* 1. Overview of switch functionality
  2. Forwarding Table

In order to implement the table, the switch uses a combination of Content Addressable Memory (CAM) and Block Memory Generator (RAM). MAC addresses are stored in the CAM and their corresponding port numbers are stored in the RAM using the same indexes. By storing both the MAC address and its corresponding port number at the same index in both memories, the switch can easily determine the mapping. For instance, if MAC address 01-23-45-67-89-ab is stored at index 2 in the CAM, then its corresponding port is stored at index 2 in the RAM.

The Verilog code uses the content matching capabilities of the CAM to determine whether a MAC address exists in the table. In a search instance, the CAM will return the index number where the MAC address is located at, otherwise it will return FALSE. After retrieving the index number, the same value is used as the index number to get the port number from the RAM. In other words, the RAM is fed the index number returned by the CAM and it will return a port number.

If a search in the CAM is not able to determine a matching MAC address, a corresponding port number will not exist in the RAM and the packet will be broadcasted.

* 1. Packet decoding and counting
  2. Access and update of counter registers
  3. Testing.

To perform the simulation tests, the ModelSim HDL simulator was used. As in the Ethernet Hub, the Switch was tested using the provided Perl library to generate packets and feed them into the system. The make\_pkts.pl file in the Perl library is the main component of the simulator test. The file generates test packets entering from the MAC or CPU ports and generates the expected output packets so that they could be compared.

To test the Switch design, the make\_pkts script was modified to send and receive three different packets, one packet with an unknown destination, one packet with a known destination, and one broadcast packet.

* 1. Command-line register access program

1. Discussion & Conclusion
2. Appendix