NetFPGA - An Open Platform for Gigabit-rate Network Switching and Routing

John W. Lockwood, Nick McKeown, Greg Watson, Glen Gibb, Paul Hartke,
Jad Naous, Ramanan Raghuraman, and Jianying Luo

Stanford University *
Computer Systems Laboratory: Gates 3A-337, Stanford, CA 94305-9030
{jwlockwd, nickm, gwatson, grg, phartke, jnaous, ramananr, jyluo}@stanford.edu

Abstract

The NetFPGA platform enables students and researchers to build high-performance networking systems in hardware. A new version of the NetFPGA platform has been developed and is available for use by the academic community. The NetFPGA 2.1 platform now has interfaces that can be parameterized, therefore enabling development of modular hardware designs with varied word sizes. It also includes more logic and faster memory than the previous platform. Field Programmable Gate Array (FPGA) logic is used to implement the core data processing functions while software running on embedded cores within the FPGA and/or programs running on an attached host computer implement only control functions. Reference designs and component libraries have been developed for the CS344 course at Stanford University. Open-source Verilog code is available for download from the project website.

1 Introduction

Hardware-accelerated network switches and routers enabled rapid growth of the Internet. Today, Gigabit Ethernet switches are widely deployed to switch traffic within Local Area Networks (LANs) and route Internet Protocol (IP) packets across Wide Area Networks (WANs). Commercial vendors use Application Specific Integrated Circuits (ASICs) and/or Field Programmable Gate Arrays (FPGAs) to accelerate the switching, routing, and processing of network data. Today's students must understand how to process packets in hardware if they wish to build real systems.

At most universities, students build network systems using software. Most hands-on courses only teach how to write software programs that send and receive packets from user-space sockets. Advanced courses typically only cover methods that interface software to the kernel of an operating system. While software-based systems can process a limited volume of network traffic, they are not suitable for switching, routing, and processing large volumes of packets that appear on a fully-loaded Gigabit Ethernet links.

2 The NetFPGA Platform

The NetFPGA 2.1 platform contains a large Xilinx Virtex-II Pro FPGA that is programmed with user-defined logic and a small Xilinx Spartan FPGA that programs the Virtex and implements the PCI interface to a host processor. The platform includes two Static RAMs (SRAMs) that operate synchronously with the FPGA and two second-generation Double Date Rate (DDR2) SDRAM devices that operate asynchronously with the FPGA. A quad-port physical-layer transceiver (PHY) is provided enabling the platform to send and receive packets over four, standard twisted-pair Ethernet cables. Two Serial ATA (SATA) connectors on the platform allow multiple NetFPGAs within a system to exchange data at high speeds. A photograph of the NetFPGA 2.1 platform is shown in Figure 1.

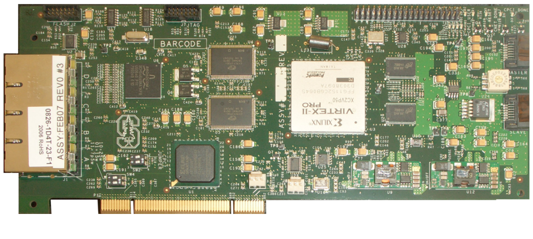

Figure 1. Photo of the NetFPGA 2.1

The NetFPGA library of logic includes a Verilog skeleton design that instantiates four Gigabit Ethernet Media Access Controllers (GMACs) and interfaces the logic to the SRAM and DDR2 memory. Internal module interfaces use a standard request-grant First-In-First-Out (FIFO) protocol.

[*] This research is supported by gifts from Xilinx, Agilent, Micron, Broadcom, and Cisco as well as grants from NSF EIA-0305729 and CNS-0551703. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or other sponsors of this project.

2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)
0-7695-2849-X/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

The NetFPGA uses standard Computer Aided Design (CAD) tool flows to implement the circuits that run on the FPGA. These tools are available at minimal or no cost to universities via corporate-sponsored academic programs. The function of the logic is simulated using the Mentor Graphics ModelSim tool. Verilog source code is synthesized into a netlist that defines the configuration and topology of the hardware resources. Logic is placed and routed using the Xilinx ISE tools. Bitfiles are programmed into the Virtex FPGA through the PCI interface so that no download cable is needed in order to reconfigure the logic.

Packets arrive to the input module of the NetFPGA from the four network interfaces via Gigabit Ethernet, the host CPU via the PCI bus, and from other NetFPGA boards interconnected by Multi-Gigabit Transceivers (MGT) via SATA connectors. A *user filter* follows the input module to perform de-encapsulation, decryption, and other user-defined functions on the packet. Next, an *output port lookup* is performed to determine to which port(s) the packet should be forwarded. For Ethernet switching, the decision is based on the MAC address at the start of the Ethernet frame while for IP routing the decision is based upon the Destination IP address field in the packet header. Other policies that use combinations of fields can be implemented by modifying the logic in the lookup algorithm.

A Linux device driver allows software on the host PC to communicate with the user-defined logic in the Virtex. The host PC can directly read and write memory-mapped registers within the FPGA and access all memory on the NetFPGA platform. In total, 128 MBytes of memory is allocated for each NetFPGA installed in a computer. Counters track statistics on the traffic that flows through the hardware. The values of these counters can be read by software on the host through the PCI bus as memory-mapped registers.

3 Stanford's CS344 Router Design Course

Stanford's CS344 is a graduate-level, project-based course that teaches graduate students how to build an Internet router in just eight weeks. Masters students that took the class in the spring of 2004, 2005, or 2007 developed hardware and software components for the NetFPGA to implement a fully functional router. Each team consisted of one student to work on hardware and one to two students to write the router control software.

In CS344, student teams implement fully functional, high-performance routers and switches. Together, all teams build a complete network topology that routes packets through a mesh of multiple routers. The topology of a sample network implemented with five NetFPGA routers interconnecting a cluster of PCs and workstations to network servers and an Internet gateway is shown in Figure 2. Hardware forwarding tables populated by the routing software routes packets along the shortest path between machines.

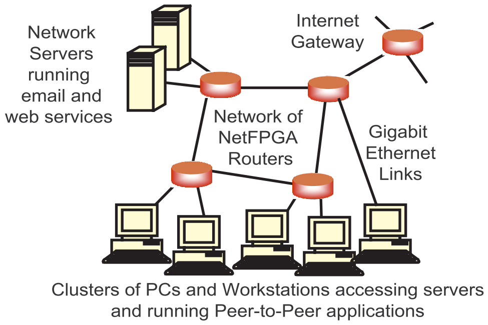

Figure 2. Topology with NetFPGA Routers

When students first started the course in 2005, they were given a Verilog HDL code base for a two-port Ethernet switch and a NetFPGA 2 platform. As students progressed through the course, they extended and modified parts of the code to build the functionality for a four-port Internet router. Students used CAD tools to design, verify, and synthesize NetFPGA hardware modules. Software running on the host computer managed the routing tables. Hardware performed the address lookup, MAC address learning, packet buffering, and packet scheduling. Software implemented network management and participated in a distributed, dynamic routing protocol by responding to ARP, ICMP, and route update messages.

Based on feedback provided by students, we found that students learned where to draw boundaries within their logic to implement modules, how to debug hardware incrementally, and why extensive test plans are needed to test a large system. At the end of the class, the teams demonstrated that their router interoperated with all the routers built by the other teams in the class.

4 Hardware and Community Workshops

NetFPGA hardware is available from `DigilentInc`. Workshops are planned in 2007 so that instructors at other institutions can learn how to readily adopt the NetFPGA courseware and use this open platform for their own projects and courses. The dates and locations of these workshops can be found on the Stanford NetFPGA homepage.

References

[1] Stanford University, NetFPGA Homepage, `http://netfpga.stanford.edu`, Apr. 2007

[2] G. Watson, N. McKeown, M. Casado, "NetFPGA: A tool for network research and education", in *Workshop on Architecture Research using FPGA Platforms*, Feb. 2006.

[3] John W. Lockwood, "Platform and Methodology for Teaching Design of Hardware Modules in Internet Routers and Firewalls", in *MSE'2001*, Jun. 2001.