

Internet-based Tool for System-On-Chip Project Testing and Grading

Christopher E. Neely, Christopher K. Zuver, John W. Lockwood
Applied Research Laboratory
Department of Computer Science and Engineering
Washington University
1 Brookings Drive, Campus Box 1045
Saint Louis, MO 63130
lockwood@arl.wustl.edu

ABSTRACT

A tool has been developed to automate the testing and grading of design projects implemented in reprogrammable hardware. The server allows multiple students to test circuits in FPGA hardware over the internet. A web interface allows students to upload their placed and routed designs to the server, which batches the jobs together and (1) sequentially programs an FPGA board, (2) inputs test vectors, (3) generates a report that details the results, and (4) grades the design as either “pass” or “fail.” The single server allows entire class to share the same FPGA board.

1. INTRODUCTION

Courses that use field programmable gate arrays (FPGAs) to teach microelectronic design courses are becoming increasingly popular. FPGAs have been used as a tool in a semester-long course in Reconfigurable System-on-Chip Design to teach graduate students how to build network hardware using hardware description languages. Students created an extensible Internet firewall, which were tested using the Field-Programmable Port Extender (FPX) platform [1] [3] and the Project Test Server.

A firewall is usually placed on the edge on a network and protects the network by identifying and discarding malicious and spurious traffic comprised of Internet protocol (IP) packets. The firewalls developed in this class by the students were verified by passing network packets into their designs and validating the resulting packets that were emitted.

Some of the students encountered difficulty during their transition from working simulations to functioning hardware due to their unfamiliarity with synthesizable VHDL constructs. A development environment was needed so students could quickly test their designs with actual hardware. The Internet-based server allowed testing to be performed outside of the classroom and regular hours.

The test server abstracts the physical configuration of the FPGA board and other low-level details commonly associated with FPGA design. The server grades lab assignments by comparing the response to input traffic to a desired response. It also allows students to upload custom test traffic for their

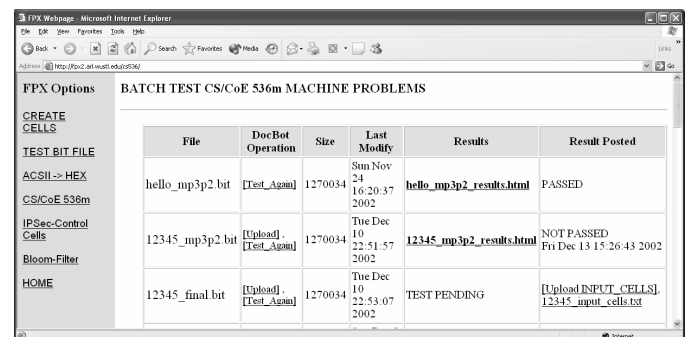
own experiments and displays the resulting output packets graphically. This paper describes the operation of the test server along with its benefits to course grading.

2. Design Environment

Student that took the course were provided with an extensive design environment including build scripts, a VHDL testbench, compile scripts, IP traffic formatters, network layer VHDL handlers, and synthesis scripts. Once a design is validated in functional simulation with ModelSim, the student synthesized and built the design using the Xilinx backend tools to create a configuration file (bitfile). A web-based GUI allows that bitfile to be uploaded to a server and programmed into an FPGA.

3. Project Test Server

It is possible that multiple files may be uploaded at once, therefore upon submission the bitfile is added to a queue of waiting configurations for testing. Prior to the next batch run, the test status in the results manager is reported as “Test Pending,” as shown in Figure 1.



File	DocBot Operation	Size	Last Modify	Results	Result Posted
hello_mp3p2.bit	[Test_Again]	1270034	Sun Nov 24 16:20:37 2002	hello_mp3p2_results.html	PASSED
12345_mp3p2.bit	[Upload] [Test_Again]	1270034	Tue Dec 10 22:51:57 2002	12345_mp3p2_results.html	NOT PASSED Fri Dec 13 15:26:43 2002
12345_final.bit	[Upload] [Test_Again]	1270034	Tue Dec 10 22:53:07 2002	TEST PENDING	[Upload INPUT_CELLS]. 12345_input_cells.txt

Figure 1: User Status and Result Manager

3.1 Testing Environment

When bitfiles reach the head of the queue the test server reprograms the FPGA and dynamically configures the network to route packets through the student’s firewall. The server displays “Test Running” while programming the FPGA and sending Internet protocol (IP) traffic to the firewall design. The input packets can be predefined for an assignment or can

be specified by the students using the same format used by their VHDL simulation testbench.

3.2 Test Results and Grading

The server creates and injects the traffic into the network, records the response traffic, and displays a test summary. The resulting packets are then parsed by the test server and displayed as an HTML document in both decimal and hexadecimal representation, as shown in Figure 2. The HTML-based parser color codes and labels each section of the IP packets so to make the test results more readable.

In a class assignment environment, student's designs are compared by the server to the solution key which contains the resulting traffic when passed through a known working implementation. A status of pass or fail is displayed in conjunction with the HTML parsing, as shown in Figure 1. This provides feedback to student about their design's validity and simplifies assigning of grades by the instructor.

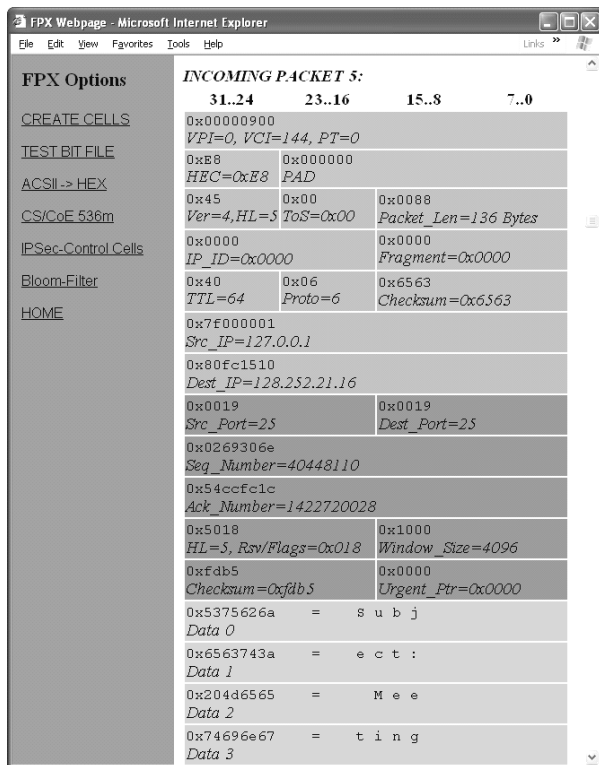


Figure 2 : HTML Test Report

4. Implementing Testbench Server

The server was created using a combination of Hypertext Markup Language (HTML) and Common Gateway Interface (CGI) scripts. The hardware testbench server was implemented on a GHz class PC running NetBSD and open-source software called NCHARGE designed for interfacing with the FPX platform [2]. NCHARGE provides the functionality to program bitfiles over the network into the FPGA and configure the network routing through the WUGS20 switch. The CGI scripts maintain queue functions

ensuring each bitfile independent access to the FPGA's resources and a fresh test environment. Once the IP traffic is transmitted through the circuit the bitfile is replaced by the next test in the queue. Typical tests complete in 15 seconds with results immediately available to the user.

To ensure that students always submit their own work, the CGI scripts compares bitfiles to check for identical submissions. The timestamp of upload and also the test results are logged and stored internally. The bitfiles are saved for future reference by the instructor.

5. From Simulation to In-System Testing

Traversing from simulation to In-System testing can be difficult. The testbench server eases this transition by accepting the same user defined traffic used by the VHDL testbench in simulation. The packets are represented by an ASCII text file, which lists the contents of the traffic in sequential order. This data file is created using C scripts in the student's design environment which automatically generate IP header information and formats the payload into IP packet (IP/TCP/UDP) form. This text file can be uploaded to the server which transforms the text file into bits formatted for hardware testing. When a run is successful the hardware's output is identical to the student's functional simulation.

6. Conclusions

The networked project test server allows an entire class to prototype designs using a single FPGA over the Internet. The server has proved an invaluable tool for teaching FPGA design by providing students with informative results within seconds of submission. Through the web-based GUI, bitfiles can be uploaded, packets can be injected, and the results can be viewed graphically. The server along with the design environment has allowed students to prototype hardware for an Internet firewall and quickly proceed from a concept to a fully functional implementation in hardware.

7. References

- [1] John W. Lockwood, Christopher Neely, Christopher Zuver, "CS536Course Website," Washington University, <http://www.cs.wustl.edu/~lockwood/class/cs536/index.html> (current December 2002).
- [2] Todd Sproull, John W. Lockwood, David E. Taylor, Control and Configuration Software for a Reconfigurable Networking Hardware Platform, IEEE Symposium on Field-Programmable Custom Computing Machines, (FCCM), Napa, CA, April 24, 2002
- [3] John W. Lockwood, Jon S. Turner, David E. Taylor, Field Programmable Port Extender (FPX) for Distributed Routing and Queuing, ACM International Symposium on Field Programmable Gate Arrays (FPGA'2000), Monterey, CA, February 2000, pp. 137-144.