



NetFPGA: Introducción y Arquitectura

Manuel F. Jaimes

Abril 10 de 2014

Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

Qué sigue

Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

Qué sigue

Qué es NetFPGA

Plataforma abierta de bajo costo, diseñada principalmente como una herramienta para la enseñanza de hardware de red

Plataforma de desarrollo que permite a los estudiantes y equipos de investigación, construir, diseñar y prototipar hardware de redes de alto rendimiento usando FPGA's



Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

Qué sigue

NetFPGA v1

Su desarrollo inicio en el año 2001 y el primer prototipo funcional fue usado en una clase de posgrado en la universidad de Stanford en 2003.

Características:

- 6 inches x 9 inches $\approx$ 15cm x 23 cm
- 3 x Altera EP20K400 APEX
- Eight-port Ethernet controller
- 3 x 1MByte SRAMs
- Ancillary logic.

Limitaciones:

1. **Board**
2. **Carencia de velocidad**
3. **Falta de una CPU interna**
4. **Herramientas CAD**

Herramientas

- Synopsys tolos
- Altera Quartus tool

Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

Qué sigue

NetFPGA 1G

Su desarrollo inicio en verano de 2004 y en Septiembre de 2005 aparece el primer prototipo de 1G. La NetFPGA de 1G consiste en:

- Tarjeta PCI
- FPGA Xilinx Virtex-II Pro
- 4 x interfaces Gigabit Ethernet
- Repositorio de código.

Objetivos

- Estudiantes aprenden a construir sistemas de redes sólo a través de software.
- La relación con la capa física y capa de enlace es limitada a la teoría.
- Crear diseños reutilizables.

Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

Qué sigue

NetFPGA 1G – Especificaciones



- FPGA Xilinx Virtex-II Pro 50
- Double-Data Rate Random Access Memory (DDR2 DRAM) – *Packet buffering*
- 4.5 MB Static Random Access Memory (SRAM) – *Forwarding table*
- FPGA Spartan 2
- JTAG
- Interfaces estándar Gigabit Ethernet.
- Procesamiento a velocidad de enlace en todos los puertos en todo momento.
- 2 conectores SATA-style Multi-Gigabit I/O (MGIO)
- Dimensiones: 10 inches x 4 inches ≈ 25cm x 10cm

Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

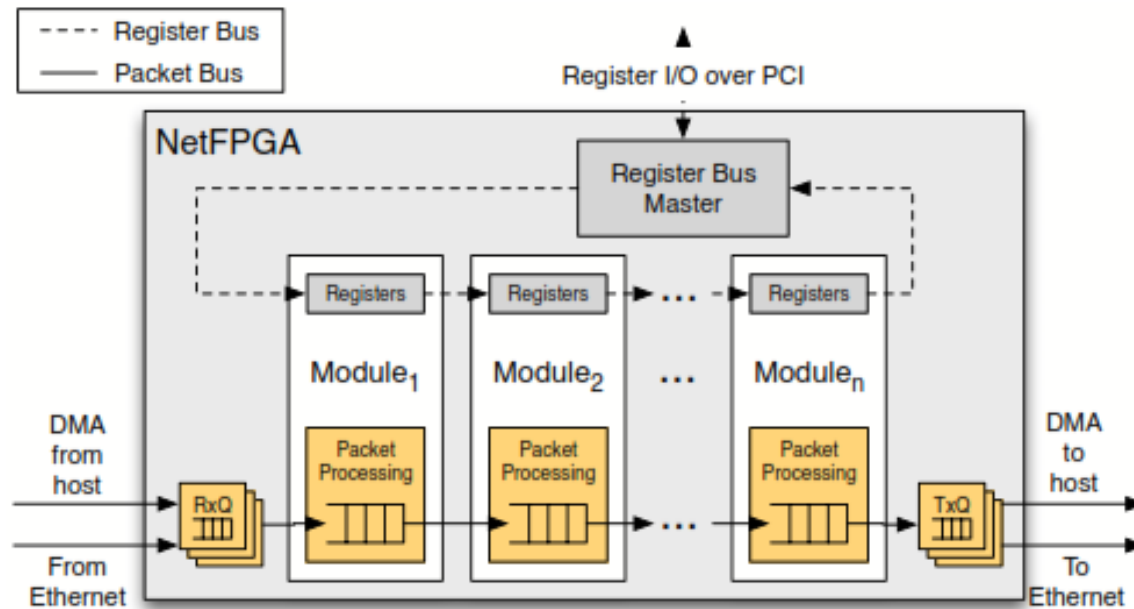
Pipeline de referencia

Flujo de diseño

Qué sigue

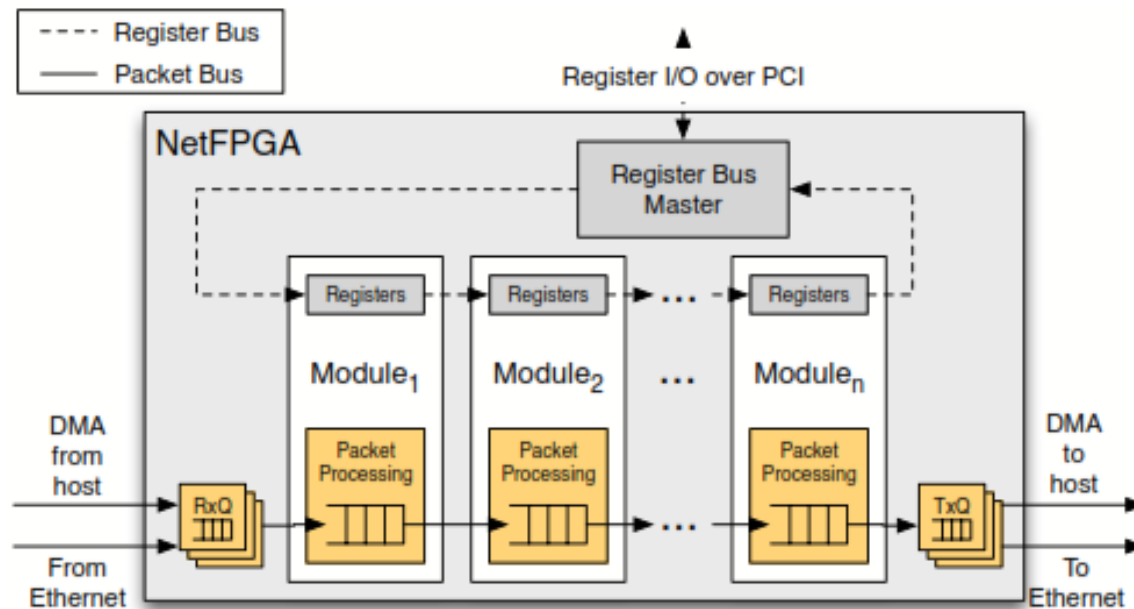
NetFPGA 1G – Detalles del pipeline

- Los módulos de la NetFPGA están conectados como una secuencia de estados dentro del *pipeline*
 - Los estados se comunican usando una interfaz FIFO
 - Estado $i+1$ le dice al estado i que tiene espacio para un paquete
 - Estado i escribe un paquete en $i+1$
 - Estado i puede anteponer cualquier información que quiera transmitir en el comienzo del paquete



NetFPGA 1G – Detalles del pipeline

- **Packet bus:** Transfiere paquetes de una etapa a la siguiente usando una interfaz FIFO.
 - ctrl bus
 - data bus
 - write signal
 - ready signal
- **Register bus:** The register bus strings together register modules in each stage in a pipelined daisy-chain that is looped back in a ring.



Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

***Pipeline* de referencia**

Flujo de diseño

Qué sigue

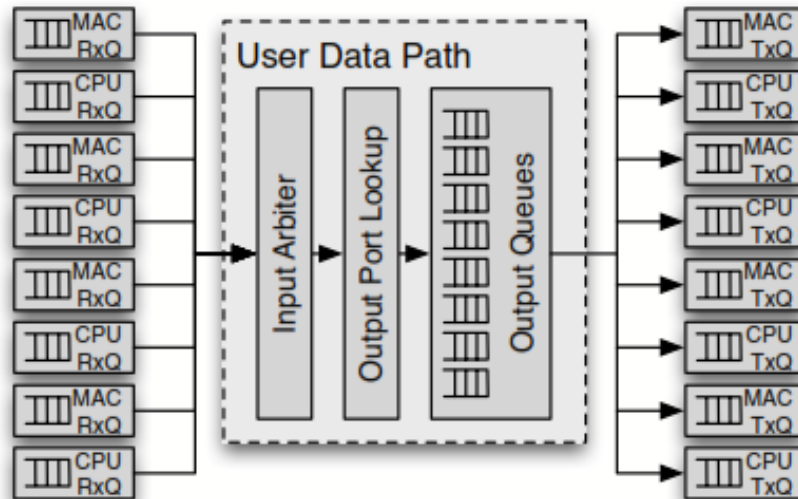
NetFPGA 1G – Pipeline de referencia

1. Rx Queues:

- Recibe los paquetes de los puertos de E/S de la tarjeta.
- Adjunta un *module header* indicando: Longitud del paquete y puerto de ingreso.
- Interfaz unificada para el resto del sistema.

2. Input Arbiter:

- Selecciona a cuales de las *Rx Queues* atiende (*Round-robin*)



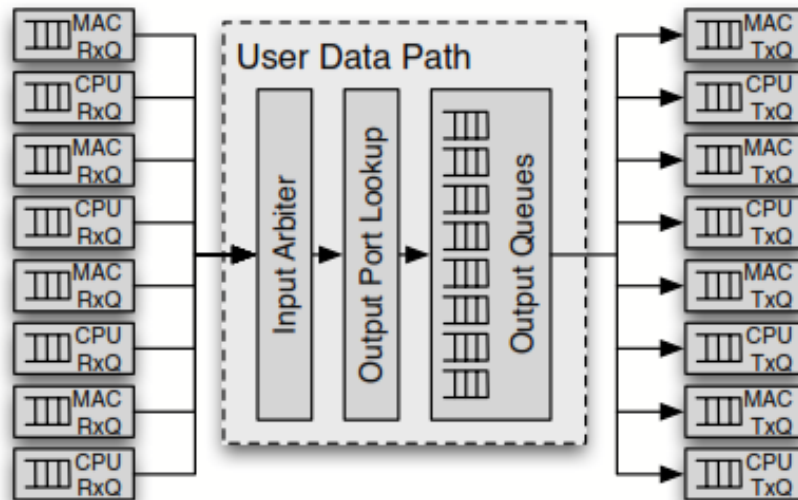
NetFPGA 1G – Pipeline de referencia

3. Output Port Lookup:

- Selecciona en cual cola de salida (*Output queue*) debe ser puesto el paquete y si es necesario modificarlo.
 - Decrementa el TTL
 - *Check y update* del IP *checksum*
 - Modifica el *module header* inicial para indicar el puerto de salida

4. Outputs Queues:

- Pone el paquete en uno de los 8 buffers de salida (4 CPU y 4 Ethernet) usando la información del *module header*.

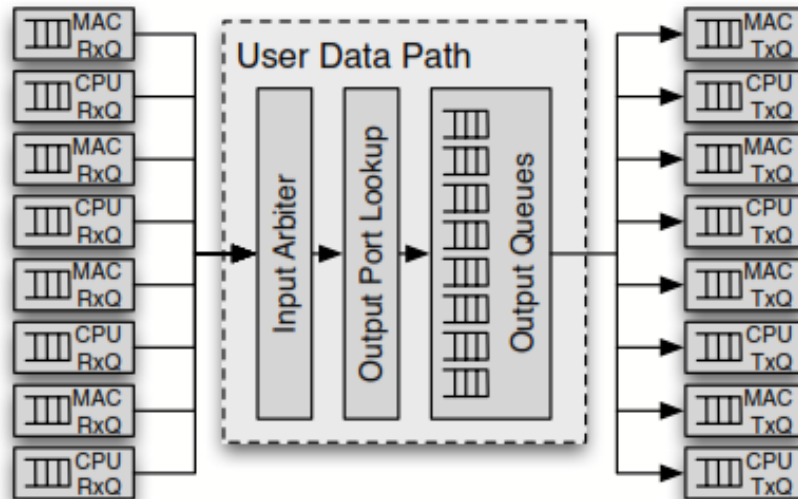


NetFPGA 1G – Pipeline de referencia

5. Tx Queues:

- Análogas a las *Rx Queues*
- Quita los *module headers*.
- Pone el paquete en el puerto de salida

En el diseño de este pipeline cada estado es un modulo separado.



Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

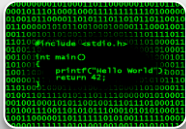
Qué sigue

NetFPGA 1G – Flujo de diseño



Definir la metodología

- Implementar un diseño existente
- Modificar un diseño
- Crear un nuevo diseño



Desarrollo de la metodología



Simulación



Síntesis



Descarga del *bitfile*



Agenda

Que es NetFPGA

NetFPGA v1

NetFPGA 1G

Especificaciones

Detalles del *pipeline*

Pipeline de referencia

Flujo de diseño

Qué sigue

Que sigue...

NetFPGA: Reusable Router Architecture for Experimental Research

Jad Nasou
Stanford University
California, USA
jnassou@stanford.edu

Sara Bolokui
Stanford University
California, USA
sbolokui@stanford.edu

Glen Gibb
Stanford University
California, USA
ggibb@stanford.edu

Nick McKown
Stanford University
California, USA
nickm@stanford.edu

ABSTRACT

Our goal is to enable fast prototyping of networking hardware (e.g., modified Ethernet switches and IP routers) for teaching and research. To this end, we built and made available the NetFPGA platform. Starting from open-source reference designs, students and researchers create their designs in Verilog, and then download them to the NetFPGA board where they can process packets at line-rate for speeds of 1 GbE. The board is becoming widely used for teaching and research, and so it has become important to make it easy to re-use modules and designs. We have created a standard interface between modules, making it easy to plug modules together in pipelines, and to create new re-usable designs. In this paper we describe our modular design, and how we have used it to build several systems, including our IP router reference design, and some extensions to it.

Categories and Subject Descriptors

B.8.1 [Logic Design]: Design Techniques—physical circuits, physical circuits; C.2.5 [Computer-Communication Networks]: Local and Wide-area Networks—network, high-speed, network; C.2.6 [Computer-Communication Networks]: Networking—routers

General Terms

Designs

Keywords

NetFPGA, modular design, reuse

1. INTRODUCTION

The benefits of reuse are well understood. It allows developers to quickly build on the work of others, reduce time to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made for distribution for profit or commercial advantage and that copies bear the name and the identification of the first page. To copy otherwise, to republish, to post on a server or to redistribute to lists, requires prior specific permission. Copyright 2008 ACM 978-1-60959-110-1/08/08...\$5.00.

market, and increases the veracity (and therefore the quality) of code. Software re-use is widely practiced and widely accepted, particularly in the open-source community, as well as in corporate development practices and commercial tools.

The key to software re-use is to create a good API—an interface that is intuitive and simple to use, and useful to a large number of developers. Indeed, the whole field of software engineering is devoted to this. In the hardware design world, there are no happy standard open-source hardware projects analogous to Linux, mostly because complicated designs only recently started to fit on FPGAs. Still, this might seem surprising as there are many strong incentives to reuse Verilog code (or other HDLs)—the cost of developing Verilog is much higher (per line of code) than for software development languages, and the importance of carefully verifying the code is much higher, as even small bugs can cost millions of dollars to fix. Indeed, companies who build ASICs place great importance on building reusable blocks or modules. And some companies exist to produce and sell reusable IP (intellectual property) blocks for use by others. In data, the companies with open-source reusable hardware have been smaller, with Openocm being the most well-known.

Reusing hardware is difficult because of the dependence of the particular design is part of: e.g., clock speed, I/Os, and so on. Unlike software projects, there is no underlying underlying operating system to provide a common platform for all configurations.

Our goal is to make networking hardware designs more reusable for teachers, students and researchers—particularly on the low-cost NetFPGA platform. We have created a simple modular design methodology that allows networking hardware designs to be reusable code. We are creating a library of modules that can be strung together in different ways to create new systems.

NetFPGA is a solution for networking hardware—it allows students and researchers to experiment with new ways to process packets at line-rate. For example, a student in a class might create a 4-port Gigabit Ethernet switch, or a researcher might add a novel feature to a 4-port Gigabit IP router. Features can be processed in arbitrary ways, under the control of the user. NetFPGA has an industry-

NetFPGA—An Open Platform for Teaching How to Build Gigabit-Rate Network Switches and Routers

Oliver Chibb, Member, IEEE, John W. Lockwood, Member, IEEE, Tad Nassou, Paul Harkis, and Nick McKown, Fellow, IEEE

Abstract—The NetFPGA platform enables students and researchers to build high-performance networking systems using field-programmable gate arrays (FPGAs) hardware. A new version of the NetFPGA platform has been developed and is available for use by the academic community. The NetFPGA platform has modular interfaces that enable development of complex hardware designs by integration of simple building blocks. FPGA logic is used to implement the core data processing functions while software running on an attached host computer or embedded core within the device implements control functions. Reference designs and component libraries have been developed for the COTS cores of Stanford University, Stanford, CA, and taught at a series of tutorials held in the United States, United Kingdom, India, China, Australia, and Europe. The open-source Verilog, C, Perl, and Java reference designs are available for download from the project website.

Index Terms—Field-programmable gate arrays (FPGAs), Internet, network, protocols, routing, switches.

1. INTRODUCTION

Behind the rapid growth of the Internet, Gigabit Ethernet switches are widely deployed in interconnected computers to local area networks (LANs). Multigigabit/sec links are used to transport Internet protocol (IP) packets across wide area networks (WANs). At most universities, students only learn to build networking systems with software. The students that do take a hands-on course in computer networking typically write software programs that send and receive packets through open-source switches. Students who take advanced courses write software for the kernel that interface directly with a Linux operating system. While systems implemented with software may be able to send and receive some of the packets to and from a gigabit/sec Ethernet line card, software alone is not suitable for switching, routing, and processing most of the traffic that appears on high-speed networks.

Manuscript received July 20, 2007; revised November 20, 2007. Published in the final version of this paper on February 15, 2008. This work was supported by Xilinx, Cisco, NetScout, Agilent, Intel, Marvell, Cypress, Broadcom, Digital, Synopsys, and others. The authors would like to thank the National Science Foundation, the National Defense Science and Engineering Graduate Fellowship, and the National Science Foundation for their support of this work. The authors are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305-5080. E-mail: {oliver@stanford.edu, jwl@stanford.edu, jwl@stanford.edu, paulh@stanford.edu, nickm@stanford.edu}.

Color versions of one or more of the figures in this paper are available online at <http://www.netfpga.org/>.

Digital Object Identifier 10.1109/TPDS.2008.00666



Fig. 1. Photograph of NetFPGA installed in a desktop PC.

C commercial vendors of high-speed networking equipment use application-specific integrated circuits (ASICs) and/or field-programmable gate arrays (FPGAs) to accelerate the switching, routing, and processing of packet data. To be competitive, students need to understand how these hardware-accelerated systems operate. Using the NetFPGA platform, students can build and prototype their own hardware-accelerated networking systems and run live traffic through the hardware. Using the NetFPGA, traffic on four gigabit Ethernet links can be processed at the full gigabit/sec rate. By processing data with hardware, back-to-back packets can be scanned at gigabit rates [1, 12].

For teachers of hardware or networking classes, a set of reusable instructions and components has been developed. By starting with this infrastructure, students need not build systems from scratch. Hardware and software provided with the NetFPGA enables the platform to receive and transmit packets, buffer data in memory, and communicate with the host processor. A reference router enables the NetFPGA to operate as a full line-rate Internet router. The coreware, consisting of Web pages, homework assignments, machine problems, and presentation slides, provides instructions with the resources to teach students how to build an Ethernet switch and an Internet router.

The NetFPGA platform fits into a standard desktop or rack-mount system. The hardware supports a PC with an expansion card that has four ports of Gigabit Ethernet attached to the peripheral communication interconnect (PCI) bus. The FPGA on the platform directly handles all data-path switching, routing, and processing of Ethernet and Internet packets, leaving software to handle only control-path functions. A photograph of a PC with the NetFPGA installed is shown in Fig. 1.

2. THE NETFPGA PLATFORM

A photograph of the NetFPGA Version 2.1 platform is shown in Fig. 2. At the center of the NetFPGA platform is a large Xilinx FPGAs device. Surrounding the FPGA are

NetFPGA: A Tool for Network Research and Education

Greg Watson, Nick McKown and Martin Casado
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-5030
{gwatson, nickm, casado}@stanford.edu

Abstract—NetFPGA is a platform that allows students to build real networking hardware, using industry-standard design tools (e.g., Verilog, then deploy and debug their hardware in an open-source network. In the educational domain, design research, a standard builds on Ethernet switch, or an Internet router and makes it interoperable with other standard solutions. NetFPGA-1 has been used in classes at Stanford for several years, and is just being replaced by NetFPGA-2, which has four Gigabit Ethernet interfaces. NetFPGA-2 is designed for teaching and research. It is open, easy to use, and enough enough to give away for free.

1. INTRODUCTION

The NetFPGA project is motivated by a need for tools to teach computer network systems at the undergraduate and graduate level. Specifically we observed that students only gain practical experience with networking at layers three and above (routing protocols, transport protocols, etc.). Students' exposure to the Link and Physical layers is limited to in-class teaching—they don't get to build network systems. There are many reasons for this, some of the obvious ones being:

- the inherent complexity of building custom hardware systems,
- the practical reality that interesting network systems need multiple devices before they become interesting, and thus are not feasible to build,
- the observation that digital design classes focus more on microprocessors than networks.

We believe that a custom platform will address the cost and complexity arguments. By developing interesting network-based project classes that are easy to use we believe we can get network systems design adopted more widely in digital design classes. We set out to build a platform that students could use to build interesting and realistic network systems (NICs, switches, routers) within a 10-week class. Our success with the tool has encouraged us and others to expand its use into research.

NetFPGA has gone through two major design iterations. This paper starts with the first version and explains how it was used in particular classes at Stanford. The limitations of this first version are noted, and lead into the description of the second version which we are just starting to use. We conclude with

The work was funded by the NSF under grant 02-0462 and grant 02-046277. Martin Casado is funded by the NSF Scholarship and Fellowship Program.

the current status and a mention of some other groups who are using NetFPGA for their research.

2. NETFPGA VERSION 1

The first version of NetFPGA was started in late 2003 and the first prototypes were used in a post-graduate level project class at Stanford in 2003. The original vision was for a rack of NetFPGA-1 boards located somewhere in Stanford. Students would have physical access to the boards, but rather they would download and debug their designs remotely. We chose to provide only remote access because the long term goal was for Stanford to host the boards and make them available to other institutions, which would provide physical access.

A. The NetFPGA-1 system

The NetFPGA-1 board itself measures 10 inches by 10 inches and contains three Altera FPGAs (FPGA devices, an 8-port Ethernet controller, three 1MByte SRAMs and an on-chip logic). There is no on-board CPU. One of the three FPGAs, called the Control FPGA (CFPGA), is pre-programmed and controls the Ethernet controller for the two Core FPGAs (CFPGA). All communication with the board is via the Ethernet ports—the only other physical connection are for power and

Students develop their designs using Synopsys tools for synthesis and simulation, and the Altera Quartus tool for place-and-route. We provide some scripts and libraries to simplify the development and debug process.

When their design has been simulated and synthesized, students upload the CFPGA configuration file to a NetFPGA-1 board via a simple web interface. The students then send packets to their board, and capture packets received from their board, in order to verify the hardware functionality. Ultimately the students disconnect their board to the campus network and their design then carries regular internet traffic. We used a locally developed tool, called the Virtual Network System [1] to map the NetFPGA-1 ports into the campus internet.

In most FPGA-based projects we have seen, including those at Stanford, there is an emphasis on hands-on debugging of the hardware using logic analyzers. For NetFPGA-1 we consider this a last resort—students are strongly encouraged to simulate first and debug last. The hands-off nature of the NetFPGA-1

NetFPGA: Reusable Router Architecture for Experimental Research

NetFPGA—An Open Platform for Teaching How to Build Gigabit-Rate Network Switches and Routers

NetFPGA: A Tool for Network Research and Education

Que sigue...

 [Home](#) [Applications](#) [Events](#) [News](#) [Videos](#) [Wiki](#) [Forums](#) [About](#)

The NetFPGA is:

a line-rate, flexible, and open platform for [research](#), and classroom experimentation. More than 2,000 NetFPGA systems have been deployed at over [150 institutions in over 40 countries around the world](#).

Watch the [Introductory Tutorial videos](#), now online!

[NetFPGA 1G \(4x1G\)](#)

2637 Boards Deployed

[NetFPGA 10G \(4x10G\)](#)

478 Boards Deployed

 [Learn More](#)

 [Get Started](#)

 [Develop](#)

Website: www.netfpga.org

Manuel Jaimes

Bienvenido



¿Quién Soy?

Estudiante Colombiano de Ingeniería de Sistemas en la Universidad Autónoma de Bucaramanga [UNAB](#), coordinador de [NetFPGA Latinoamérica](#), apasionado por las ciencias de la computación y fiel oyente de la W radio. Hincha del más veces [campeón de Colombia](#).

¿Qué hago?

Trato de profundizar en el área de las redes de computadoras mediante la plataforma [NetFPGA](#).

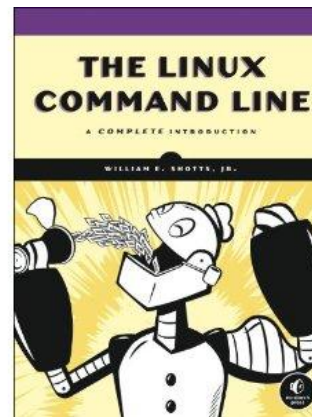
Contacto

E-mail: 4mjaimes@gmail.com



Navegador
Bienvenido
NetFPGA
Instalación
Verificación
Diseños de referencia
Estructura de carpetas
Proyectos
Blog

Website: <https://sites.google.com/site/4mjaimes/>



Website: <http://linuxcommand.org/>





Gracias

Manuel F. Jaimes

Contacto: mjaimes60@unab.edu.co

Abril 10 de 2014