Experience with the NetFPGA Program

John W. Lockwood
Algo-Logic Systems
Algo-Logic.com

With input from the
Stanford University NetFPGA Group
& Xilinx XUP Program

Sunday, February 21, 2010

FPGA-2010 Pre-Conference Workshop
on Open-Source for FPGA
NetFPGA = Networked FPGA

A line-rate, flexible, open networking platform for teaching and research
NetFPGA’s Defining Characteristics

• **Line-Rate**
  – Processes back-to-back packets
    • Without dropping packets
    • At full rate of Gigabit Ethernet Links
  – Operating on packet headers
    • For switching, routing, and firewall rules
  – And packet payloads
    • For content processing and intrusion prevention

• **Open-source Hardware**
  – Similar to open-source software
    • Full source code available
    • BSD-Style License
  – But harder, because
    • Hardware modules must meeting timing
    • Verilog & VHDL Components have more complex interfaces
    • Hardware designers need high confidence in specification of modules
Maintain an Organized Web Page

• Simple is better

Learn More
Project summary, videos, publications, tutorials

Get Started
Obtain NetFPGA hardware, download gateware & software, review reference designs

Develop
Create user account, contribute your code, document your project

• But no too simple
Provide a Reference Platform

NetFPGA

= Software running on a standard PC + A hardware accelerator built with Field Programmable Gate Array driving Gigabit network links
Make Systems Widely Available

• Pre-built systems available
  – From 3rd Party Vendor
• Details available
  – To build system from parts
Scale up Deployments

Stanford NetFPGA Cluster (NFC)

Statistics
- Rack of 40
  - 1U PCs
  - NetFPGAs
- Managed
  - Power,
  - Console
  - VLANs
- Provides 160 Gbps of full line-rate processing bandwidth
UCSD-NetFPGA Cluster
Provide Reference Design
Use a Modular Architecture

NetFPGA

Register I/O over PCI

Register Bus Master

Module_1

Module_2

... ...

Module_n

Packet Processing

Packet Processing

Packet Processing

RxQ

From Ethernet

DMA from host

TxQ

To Ethernet

DMA to host

Register Bus

Packet Bus
Include Control Software

Click the Details tab of the Quickstart window

This is the reference router pipeline – a canonical, simple-to-understand, modular router pipeline
Provide Reference Code & Modules

NF2 Directory Tree

- **bin** (scripts for running simulations and setting up the environment)
- **bitfiles** (contains the bitfiles for all projects that have been synthesized)
- **lib** (stable common modules and common parts needed for simulation/synthesis/design)
- **projects** (user projects, including reference designs)
Tree Structure (2)

```
lib
  C  (common software and code for reference designs)
  java  (contains software for the graphical user interface)
  Makefiles  (makefiles for simulation and synthesis)
  Perl5  (common libraries to interact with reference designs and aid in simulation)
  python  (common libraries to aid in regression tests)
  scripts  (scripts for common functions)
  verilog  (modules and files that can be reused for design)
```
Tree Structure (3)

projects

- **doc** (project specific documentation)
- **include** (contains file to include verilog modules from lib, and creates project specific register defines files)
- **regress** (regression tests used to test generated bitfiles)
- **src** (contains non-library verilog code used for synthesis and simulation)
- **sw** (all software parts of the project)
- **synth** (contains user .xco files to generate cores and Makefile to implement the design)
- **verif** (simulation tests)
Provide Test-Driven Designs

• Regression tests
  – Have repeatable results
  – Define the supported features
  – Provide clear expectation on functionality

• Example: Internet Router
  – Drops packets with bad IP checksum
  – Performs Longest Prefix Matching on destination address
  – Forwards IPv4 packets of length 64-1500 bytes
  – Generates ICMP message for packets with TTL <= 1
  – Defines how packets with IP options or non IPv4
    … and dozens more …

  Every feature is defined by a regression test
Teach Hands-on Tutorials

NetFPGA @ SIGCOMM - Seattle, WA

NetFGPA @ SIGMETRICS - San Diego, CA

EuroSys - Glasgow, Scotland, U.K.

Workshop in Beijing, China

Workshop in Bangalore, India
Grow a World-wide Community

- Over 1,000 NetFPGA users with 1,300+ cards deployed at 150+ universities in 17 Countries worldwide

Worldwide Hardware Deployments - Feb 2010
NetFPGA Hardware in North America

USA Deployments - Feb 2010
NetFPGA Hardware in Europe

European Deployments - Feb 2010
NetFPGA Hardware in Asia

China, Korea, Japan, Taiwan, and India Deployments - Feb 2010
Build a set of contributed projects

<table>
<thead>
<tr>
<th>Project (Title &amp; Summary)</th>
<th>Base Version</th>
<th>Status</th>
<th>Organization</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 Reference Router</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Guide</td>
</tr>
<tr>
<td>Quad-Port Gigabit NIC</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Guide</td>
</tr>
<tr>
<td>Ethernet Switch</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>Buffer Monitoring System</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Guide</td>
</tr>
<tr>
<td>Hardware-Accelerated Linux Router</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Guide</td>
</tr>
<tr>
<td>DRAM-Router</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>DRAM-Queue Test</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>Packet Generator</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>OpenFlow Switch</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>NetFlow Probe</td>
<td>1.2</td>
<td>Functional</td>
<td>Brno University</td>
<td>Wiki</td>
</tr>
<tr>
<td>AirFPGA</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki and Paper</td>
</tr>
<tr>
<td>Fast Reroute &amp; Multipath Router</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>NetThreads</td>
<td>1.2.5</td>
<td>Functional</td>
<td>University of Toronto</td>
<td>Wiki</td>
</tr>
<tr>
<td>Precise Traffic Generator</td>
<td>1.2.5</td>
<td>Functional</td>
<td>University of Toronto</td>
<td>Wiki</td>
</tr>
<tr>
<td>URL Extraction</td>
<td>2.0</td>
<td>Functional</td>
<td>Univ. of New South Wales</td>
<td>Wiki</td>
</tr>
<tr>
<td>zFilter Sprouter (Pub/Sub)</td>
<td>1.2</td>
<td>Functional</td>
<td>Ericsson</td>
<td>Wiki</td>
</tr>
<tr>
<td>Windows Driver</td>
<td>2.0</td>
<td>Functional</td>
<td>Microsoft Research</td>
<td>Wiki</td>
</tr>
</tbody>
</table>

... continued on next page ...

As of Feb 2010

.. And more on http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/ProjectTable
## NetFPGA Designs (continued..)

... continued from previous page ...

<table>
<thead>
<tr>
<th>Windows Driver</th>
<th>2.0</th>
<th>Functional</th>
<th>Microsoft Research</th>
<th>Wiki</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>Open Network Lab</td>
<td>2.0</td>
<td>Functional</td>
<td>Washington University</td>
<td>Wiki</td>
</tr>
<tr>
<td>DFA</td>
<td>2.0</td>
<td>Functional</td>
<td>UMass Lowell</td>
<td>Wiki</td>
</tr>
<tr>
<td>G/PaX</td>
<td>.2</td>
<td>Functional</td>
<td>Xilinx</td>
<td>Wiki</td>
</tr>
<tr>
<td>RCP Router</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>Deficit Round Robin (DRR)</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>OpenFlow-MPLS Switch</td>
<td>2.0</td>
<td>Functional</td>
<td>Ericsson</td>
<td>Wiki</td>
</tr>
<tr>
<td>PTP-enabled Router</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>Vlan Tag Handler</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>Port Aggregator</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford University</td>
<td>Wiki</td>
</tr>
<tr>
<td>IP Lookup w/Blooming Tree</td>
<td>1.2.5</td>
<td>In Progress</td>
<td>University of Pisa</td>
<td>Wiki</td>
</tr>
<tr>
<td>KOREN Testbed</td>
<td>.2</td>
<td>In Progress</td>
<td>Chungnam-Korea</td>
<td>Wiki</td>
</tr>
<tr>
<td>Virtual Data Plane</td>
<td>1.2</td>
<td>In Progress</td>
<td>Georgia Tech</td>
<td>Wiki</td>
</tr>
<tr>
<td>Deficit Round Robin (DRR) Input Arbiter</td>
<td>1.2</td>
<td>In Progress</td>
<td>Universidade Federal do Rio Grande do Sul (Brazil)</td>
<td>Wiki</td>
</tr>
<tr>
<td>Counter Braids</td>
<td>2.0</td>
<td>Functional</td>
<td>Stanford (Lu, Jianying)</td>
<td>Wiki</td>
</tr>
<tr>
<td>Tunneling NIC</td>
<td>2.0</td>
<td>In Progress</td>
<td>Stanford</td>
<td>Wiki</td>
</tr>
<tr>
<td>Multicore Prototype for Real-Time Switching</td>
<td>1.0</td>
<td>Available on Request</td>
<td>University of Waterloo</td>
<td>Wiki</td>
</tr>
<tr>
<td>End-to-End Ethernet Authorization</td>
<td>2.0</td>
<td>In Progress</td>
<td>Euskal Herriko Unibertsitateko</td>
<td>Wiki</td>
</tr>
<tr>
<td>Ultra-high Speed Congestion-control</td>
<td>2.0</td>
<td>In Progress</td>
<td>University of North Carolina</td>
<td>Wiki</td>
</tr>
</tbody>
</table>

.. And more on [http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/ProjectTable](http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/ProjectTable)
A Sample Project: AirFPGA

- **Software-Defined Radio (SDR) platform**
  - Collects IQ signals from
    - USB radio (SDR-IQ, SDR-14)
    - ADC daughterboard

- **Stream waveforms over the Internet**
  - Using one or more Gigabit Ethernet links

- **View waveforms remotely**
  - Using open tools like Linrad
Use an Open Software License

• Use an open software license
  – NetFPGA uses BSD-Style License (Not GNU)
    • BSD / MIT license provides unrestricted use of code
    • GNU license limits use in commercial systems

• Provide Code Checkout / Package Manager
  – Yum

• Provide Documentation
  – Wiki Guide
Hardware
Find a 3rd Party to Manufacture & Distribute HW

- Digilent, Inc for NetFPGACards
Find a vendor to ship pre-built systems

- Accent Technologies
  - Pre-built PC Systems
Occasionally Refresh the Design

Xilinx Virtex5
XCV5TX240T-2
FG1759

QDRII+ SRAM
3x 36bit interfaces, 300MHz+
(each i/f: 1x K7R643684MFC30)

SFP+ Cage
SFI 10Gbps

PHY AEL2005
XAUI 4 GTXs

PHY AEL2005
XAUI 4 GTXs

PHY AEL2005
XAUI 4 GTXs

PCIe x8, Gen1 endpoint edge connector

PCIe x8, Gen1 endpoint edge connector

2 x Samtec x10 Connector

NetFPGA 10G

RLDRAM II
2x 32bit interfaces, 300MHz+
Upcoming 40 Gbps NetFPGA
Algo-Logic Systems

- Founded by a professor and PhD researchers from Stanford University and Washington University, the Algo-Logic ® team has extensive experience building routers, data center switches, and network processing circuits in ASICs and FPGA logic. Algo-Logic specializes in mapping network algorithms into hardware logic. The founders are experts in developing, documenting, and prototyping logic and systems of reprogrammable networks.

- **Web**
  - http://Algo-Logic.com
- **Email**
  - Solutions@Algo-Logic.com
- **Phone**
  - (650) 395-7026
- **Fax**
  - (650) 498-8296

- **Office Address**
  - 530 Lytton Ave
    Second Floor
    Palo Alto, CA  94301
Algo-Logic Design Services

• **Implementations of Network Algorithms**
  – Hardware Logic in synthesizable high-level HDLs, Verilog, and VHDL
  – Ultra low latency processing
  – System-level software with APIs in C and C++

• **Architectures for Next-Generation Networks**
  – Data-center networks
  – Trading-floor networks
  – Network security
  – Content-aware networks

• **Systems Architecture and Development**
  – FPGA design
  – Verification and test bench development
  – Customization of IP cores
  – System-level Integration

*Design, develop, and verify line-rate network processing systems at multi-Gbps rates.*
Ensure that your network systems meet your performance requirements.

- Network Architecture
  - Data-center networks
  - Network security
  - Content-aware networks
  - On-chip interconnect

- Performance Analysis
  - Mathematical system modeling
  - Real and synthetic trace simulation
  - Live measurements in hardware prototypes

- Networked Sensors
  - Distributed, real-time data collection
  - Integrated, network-wide results
Algo-Logic Corporate Training

- Customized corporate training services to help your team build your own network systems in hardware.
  - Experience in teaching the world-wide NetFPGA tutorials
  - Customized training specifically tailored for your company’s skill level and requirements
  - Hands-on training with live hardware systems
Help Companies Build Solutions