Building Networks with Open, Reconfigurable Hardware

1st Asia NetFPGA Developer's Workshop

Monday, June 14, 2010

John W. Lockwood

CEO & Lead Consultant, Algo-Logic Systems

With input from:

NetFPGA Group,Stanford University

Hardware and tools available for university programs thanks to grants, donations, and/or partnerships from:

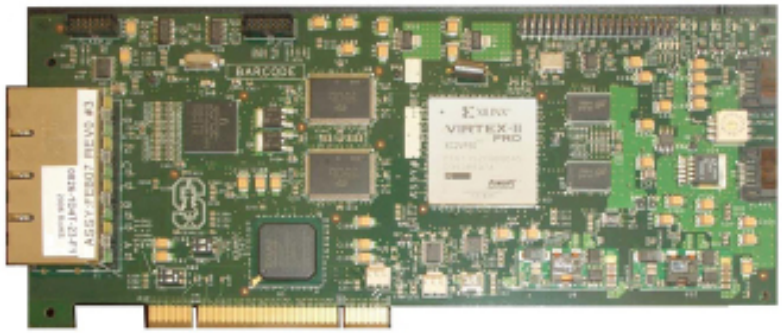
NetFPGA's Defining Characteristics

- **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

NetFPGA = Networked FPGA

A line-rate, flexible, open networking platform for teaching and research



=

- Network Interface Card
- Hardware Accelerated Linux Router
- IPv4 Reference Router
- Traffic Generator
- Openflow Switch
- List of All Projects
- Add Your Project

NetFPGA Reference Platform

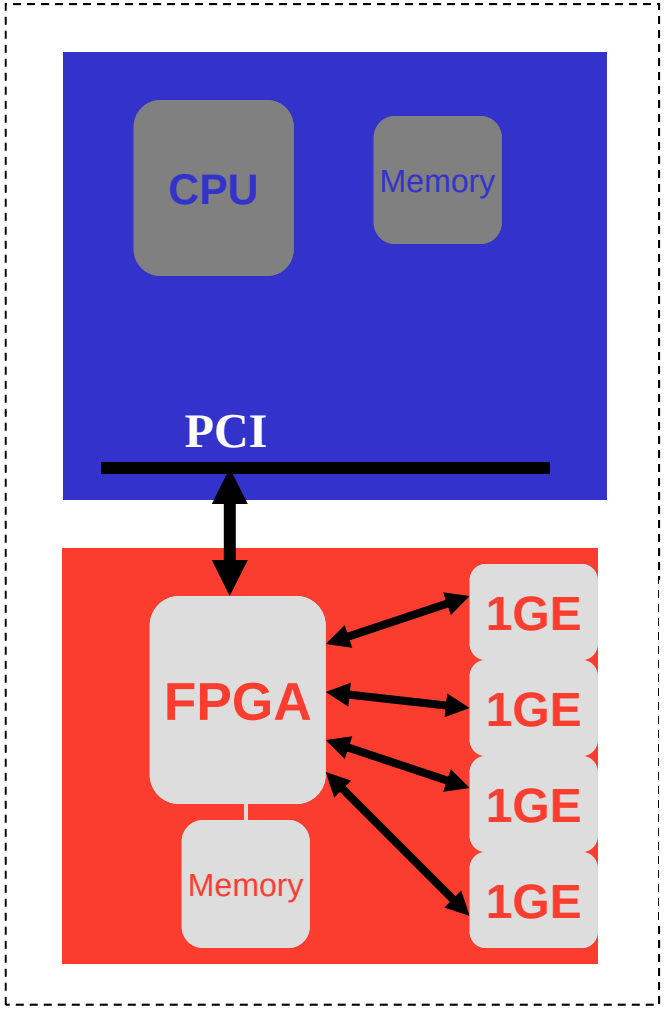
NetFPGA

=

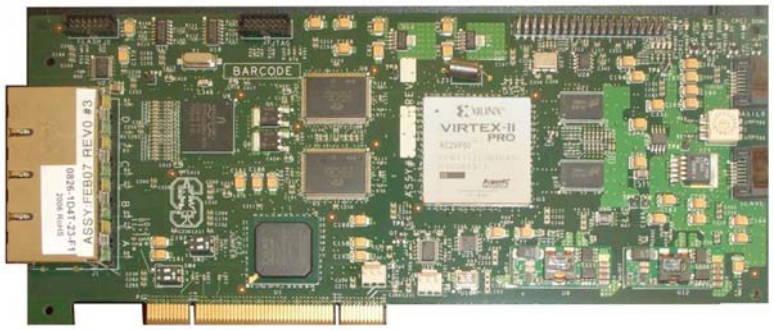
Software running on a standard PC

+

A hardware accelerator built with Field Programmable Gate Array driving Gigabit network links

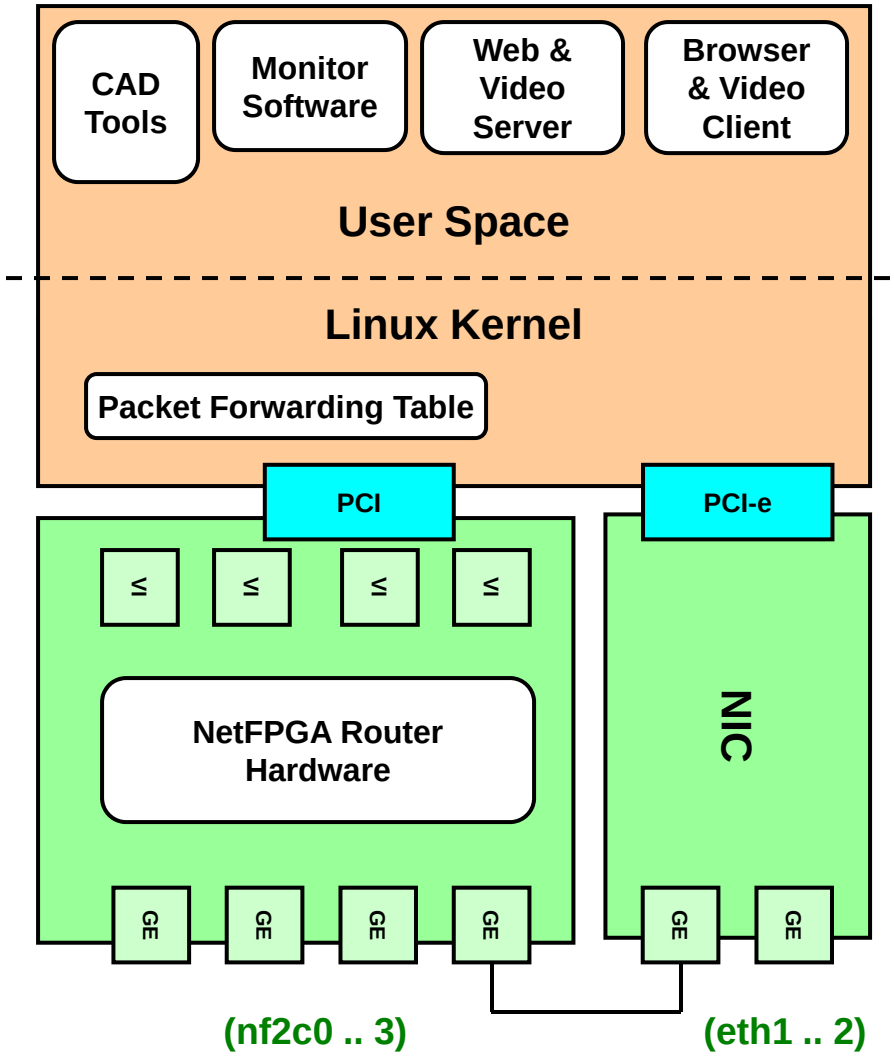


PC with NetFPGA

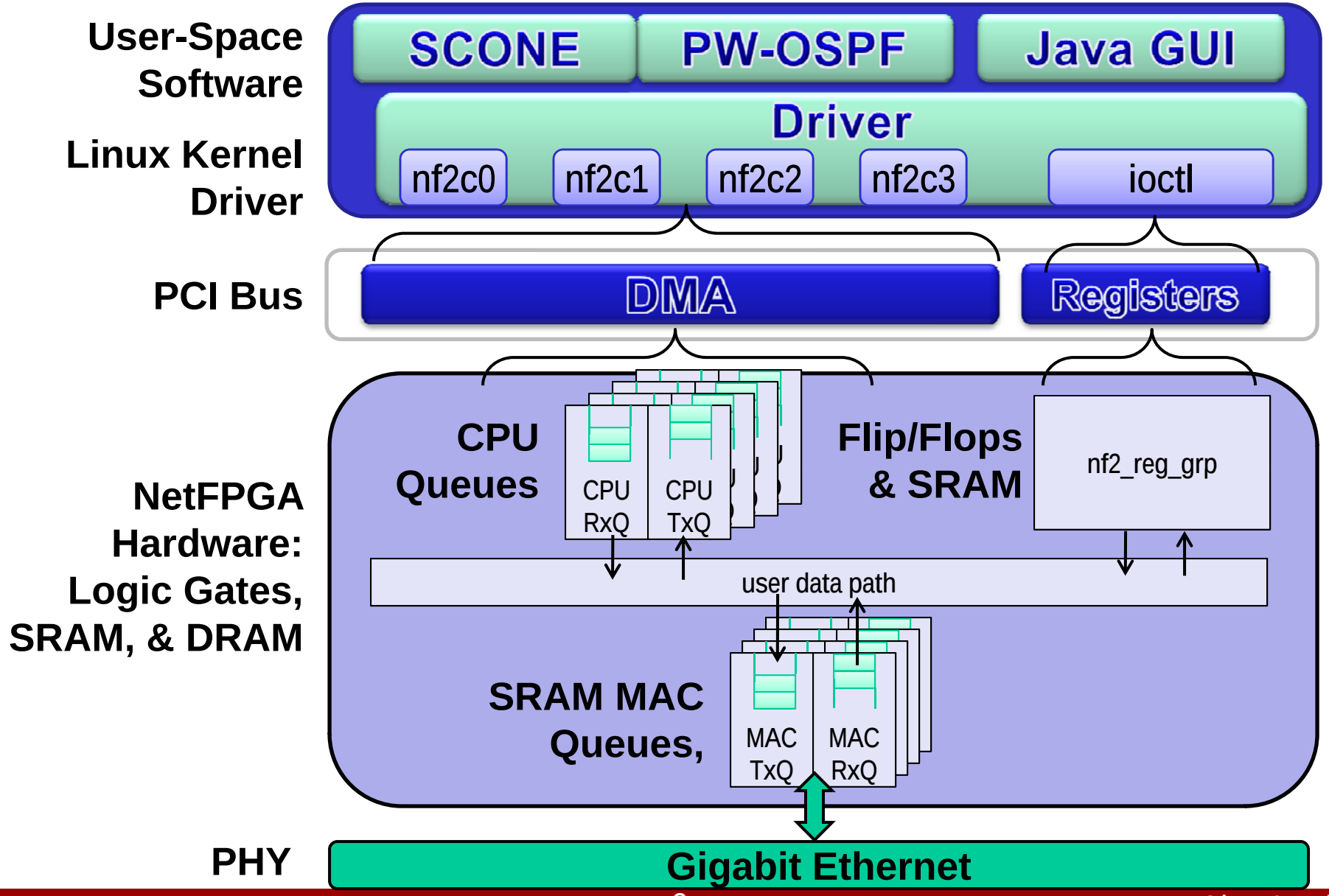


NetFPGA Board

NetFPGA System



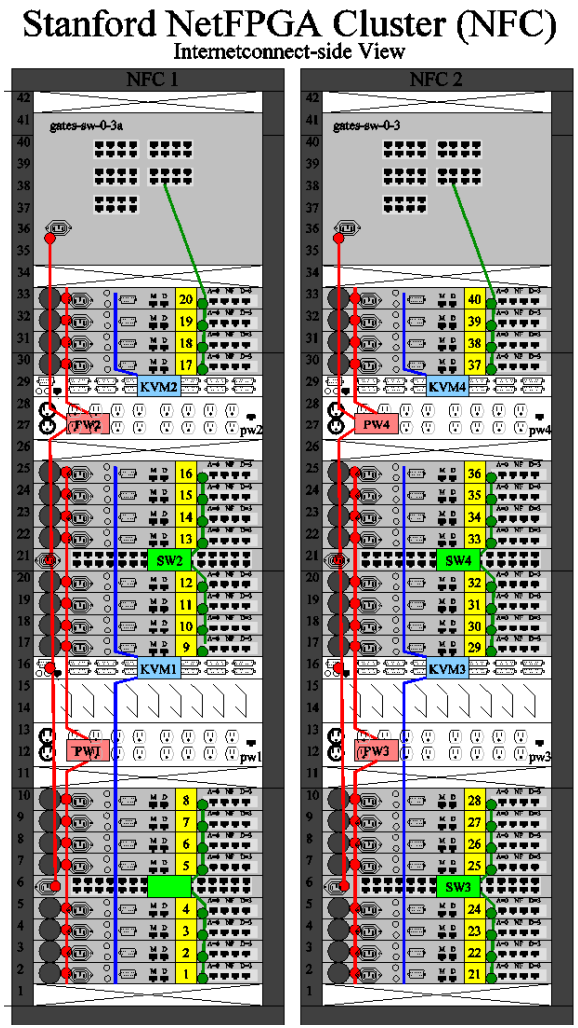
Hardware & Software Components



Open Platform, Low-cost Hardware

- **NetFPGA Cards & Pre-built Systems**
  - Available from 3rd Party Vendors
- **Or, build your system from parts**
  - Details in the on-line Guide

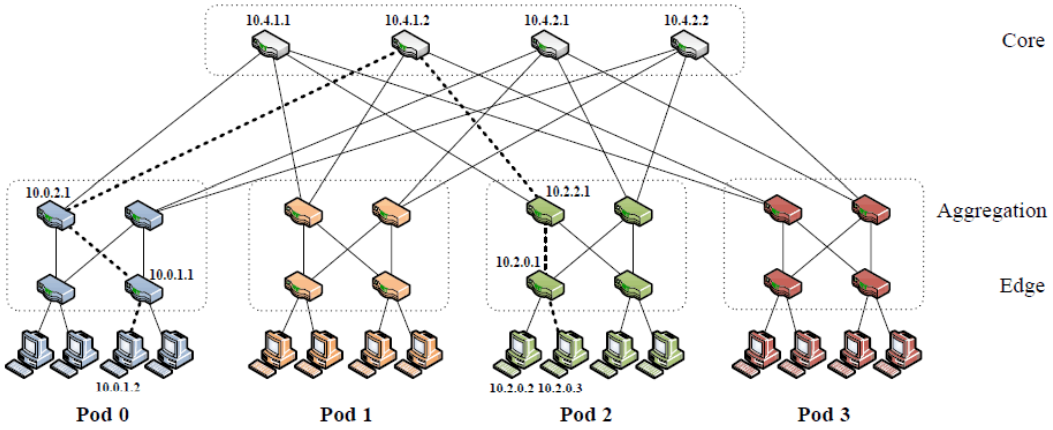
Stanford's Rackmount Deployment

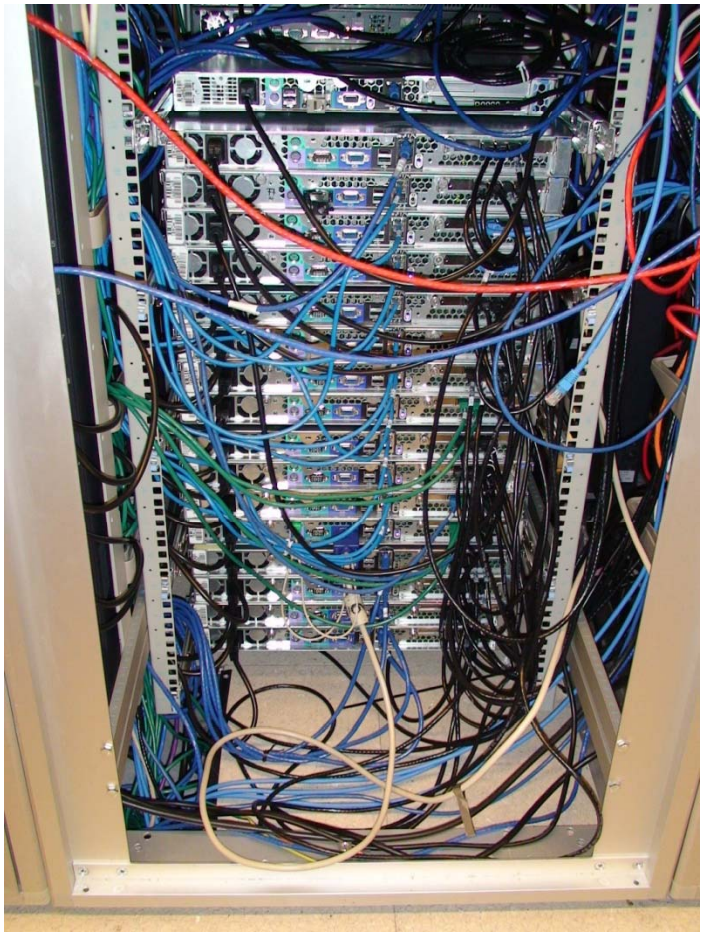


Statistics

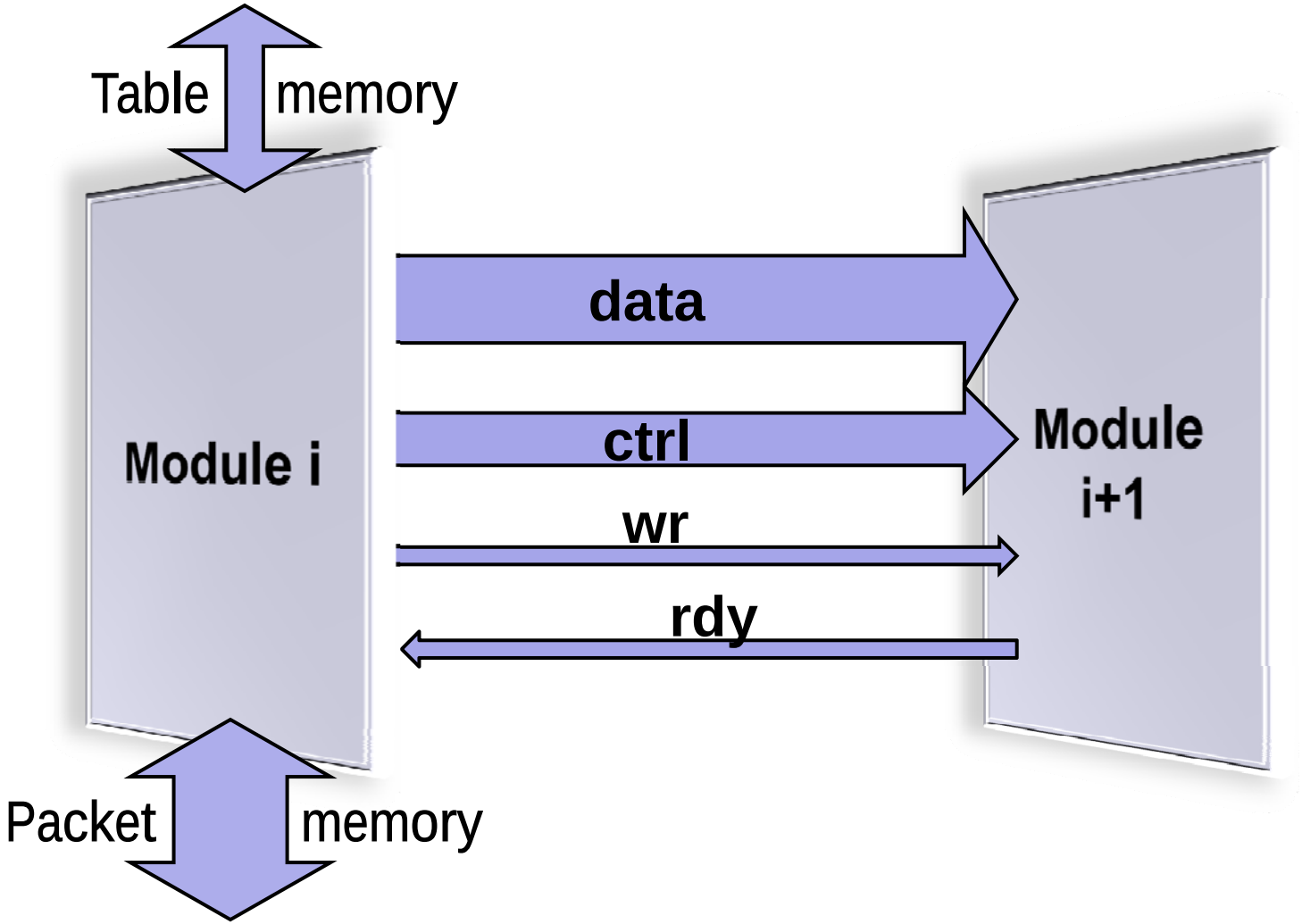
- Rack of 40
  - 1U PCs
  - NetFPGAs
- Manged
  - Power,
  - Console
  - VLANs
- Provides 160 Gbps of full line-rate processing bandwidth

UCSD's NetFPGA Cluster

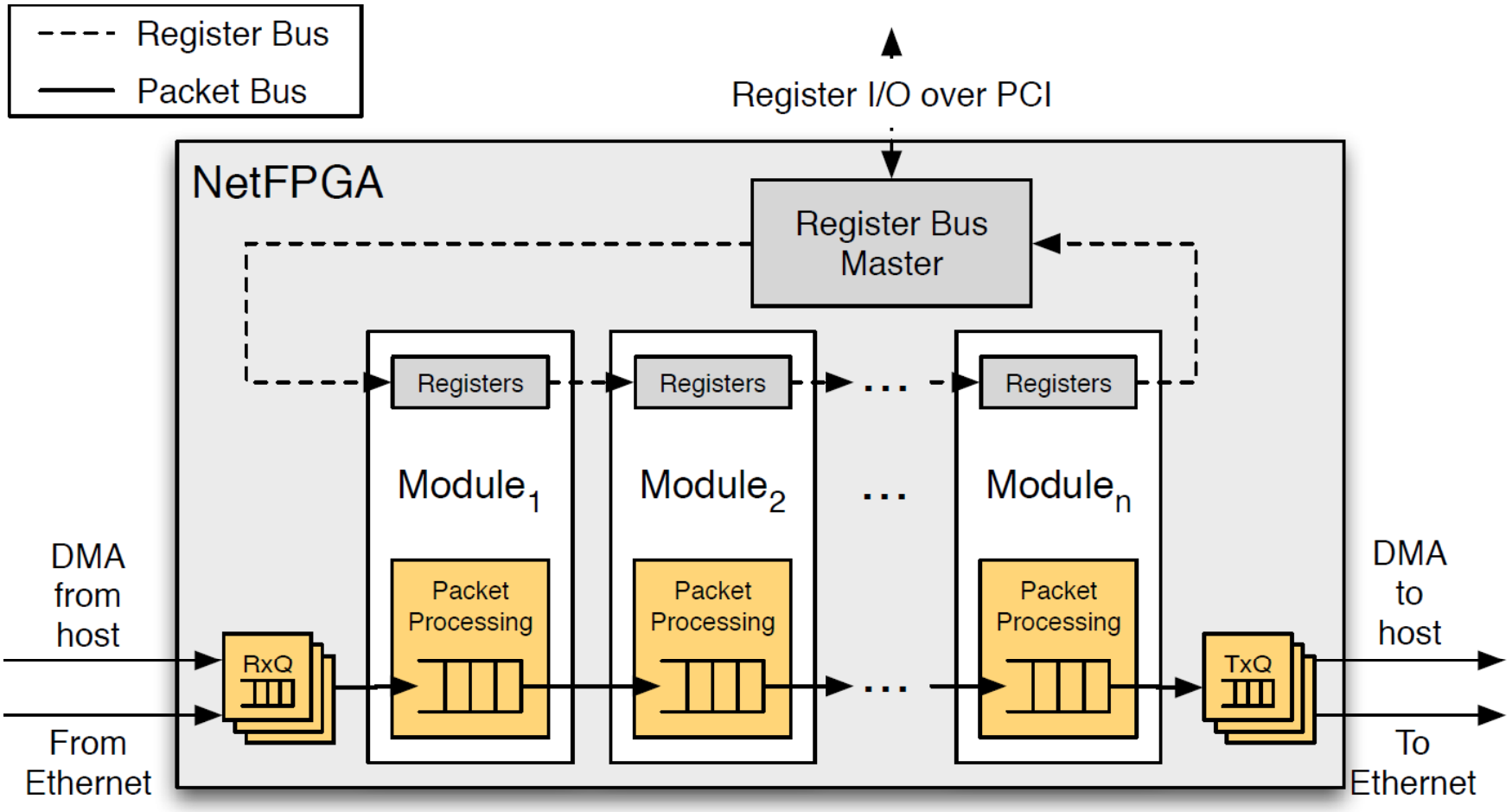




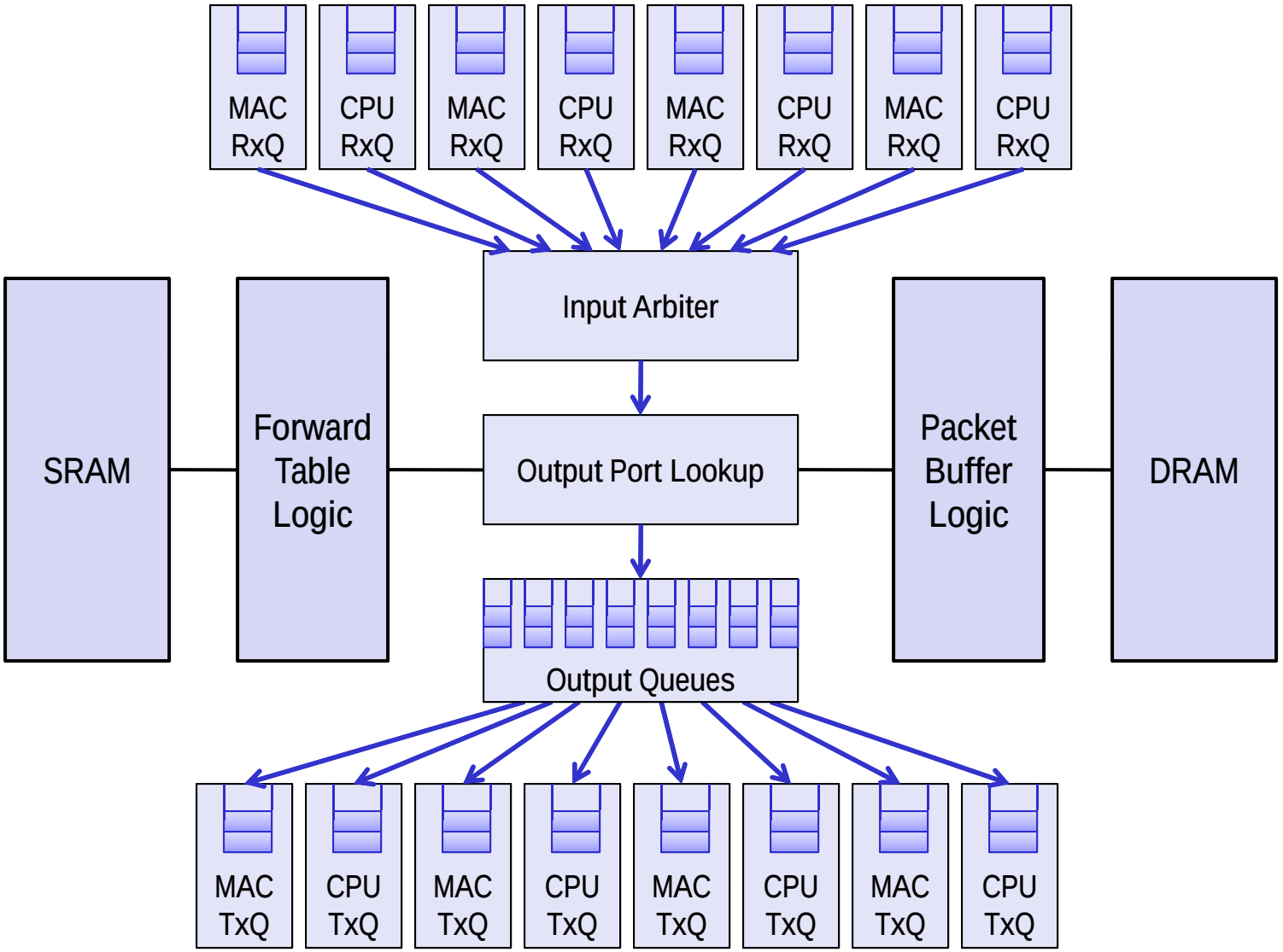
Inter-module Communication



Modular Architecture

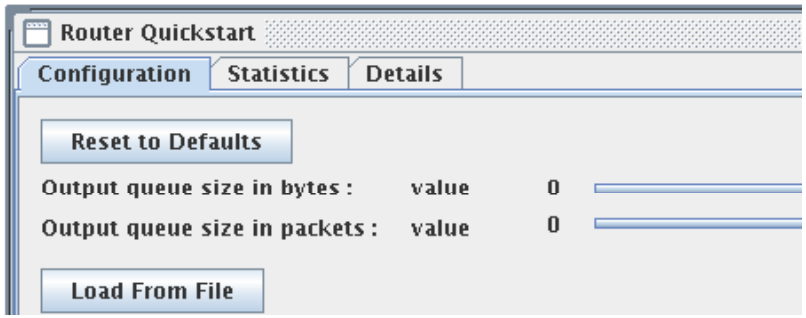


NetFPGA Base Reference Router

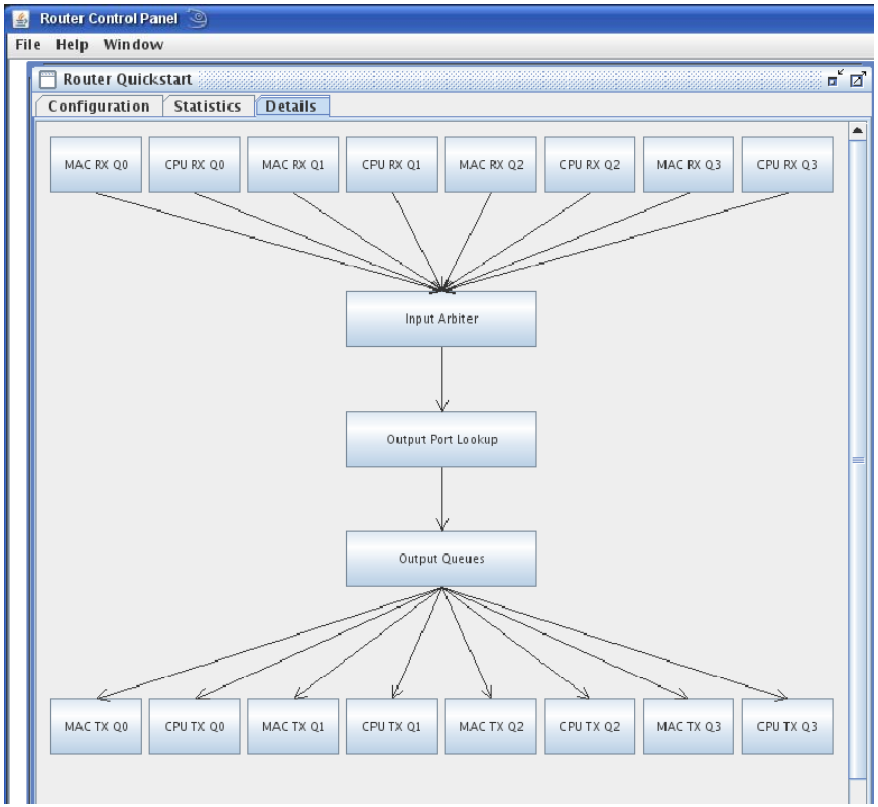


Reference Router Control Software

Click the Details tab of the Quickstart window



This is the reference router pipeline – a canonical, simple-to-understand, modular router pipeline

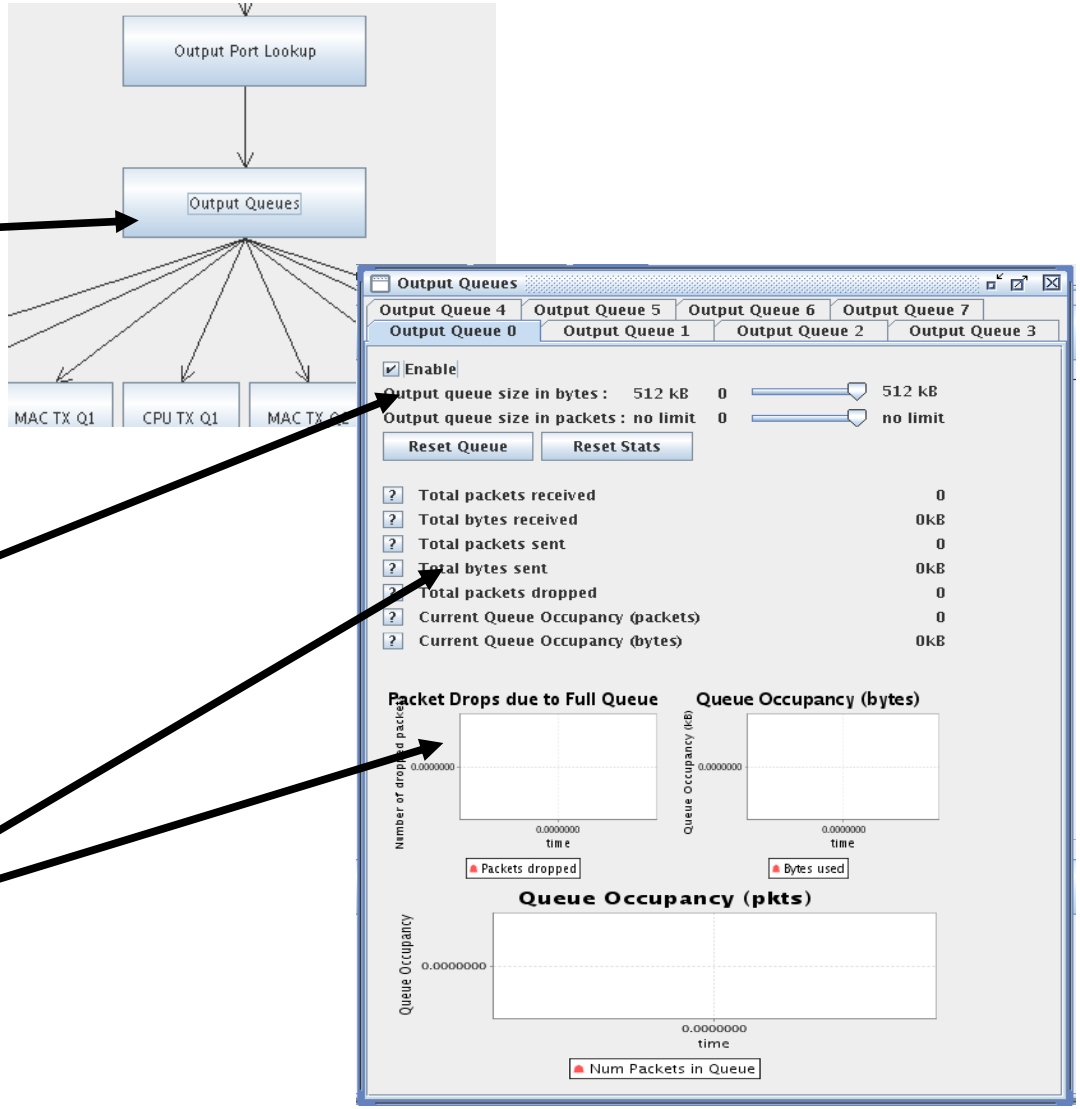


Control Software - Continued

Output Queues module (from Details tab)

Configuration details

Statistics



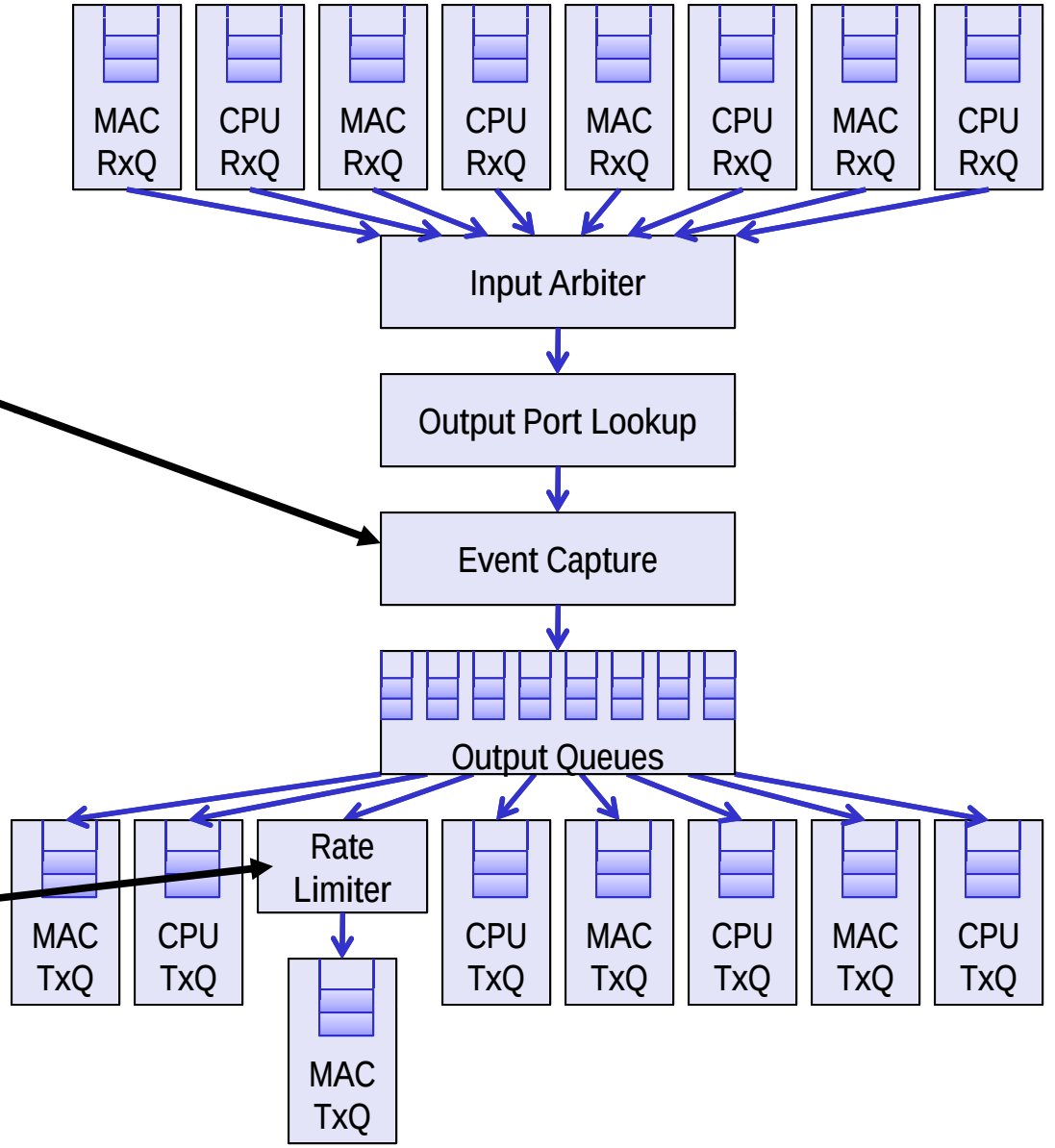
Enhanced Router Pipeline

Event Capture

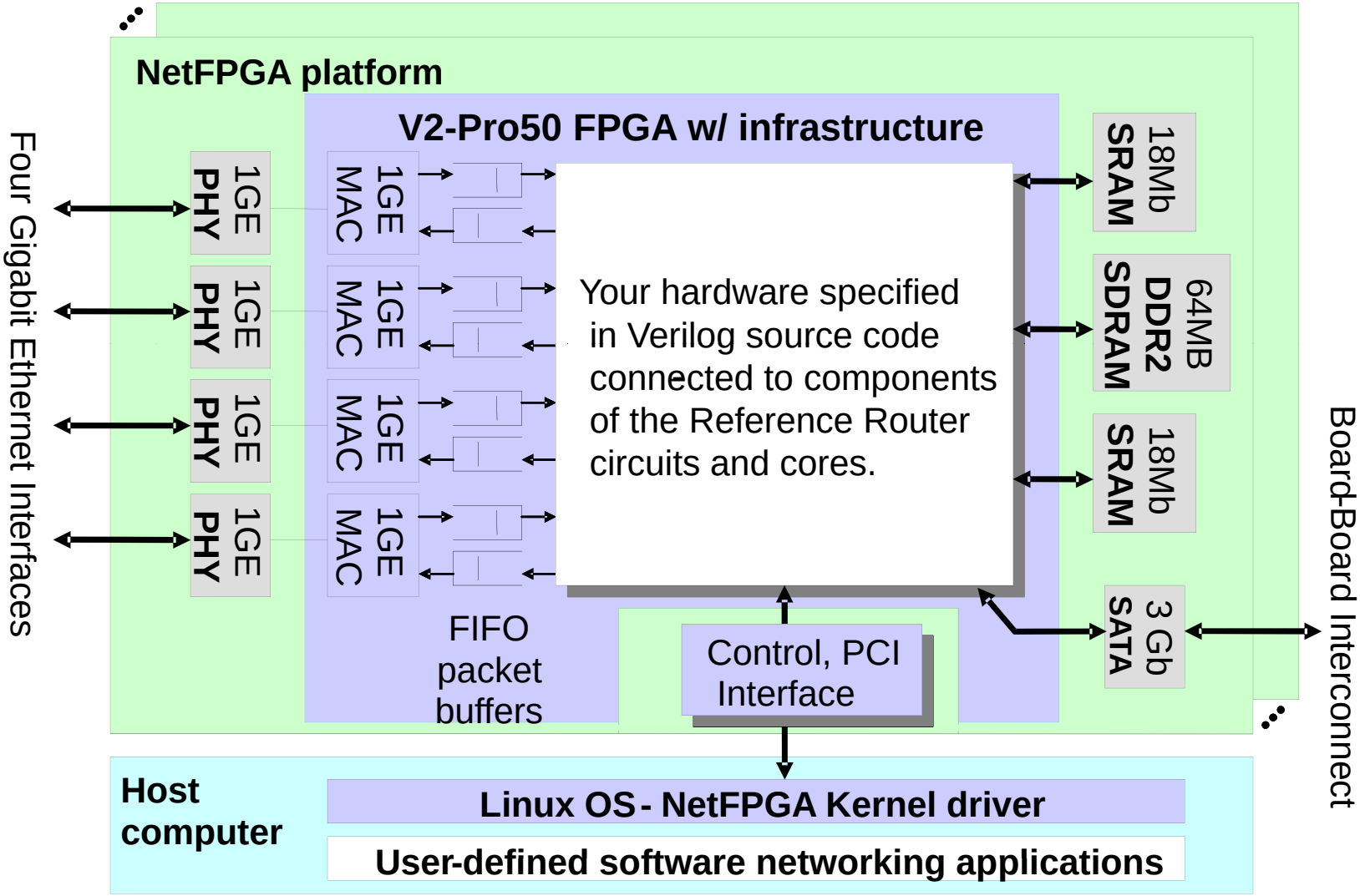
to capture output queue events (writes, reads, drops)

Rate Limiter

to create a bottleneck



NetFPGA Hardware Block Diagram



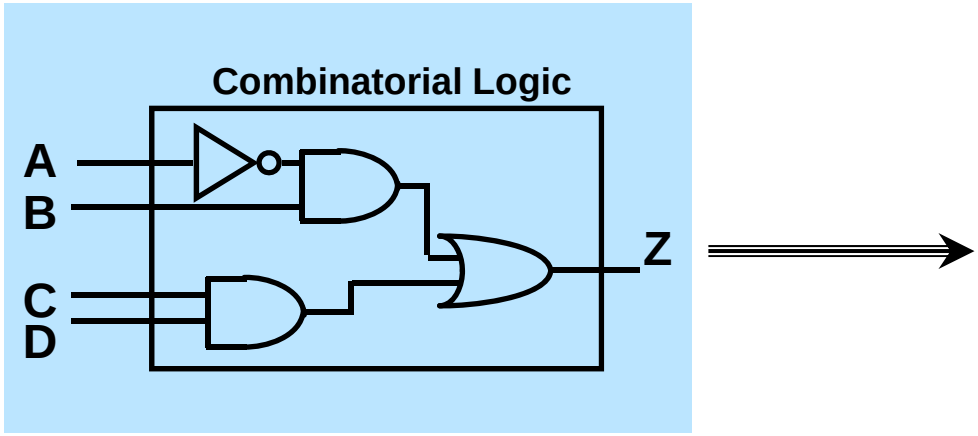
Hardware Description Languages

- **Concurrent**
  - By default, Verilog statements evaluated concurrently
- **Express *fine grain* parallelism**
  - Allows *gate-level* parallelism
- **Provides Precise Description**
  - Eliminates ambiguity about operation
- **Synthesizable**
  - Generates hardware from description

FPGA Look-Up Tables

Combinatorial logic is stored in Look-Up Tables (LUTs)

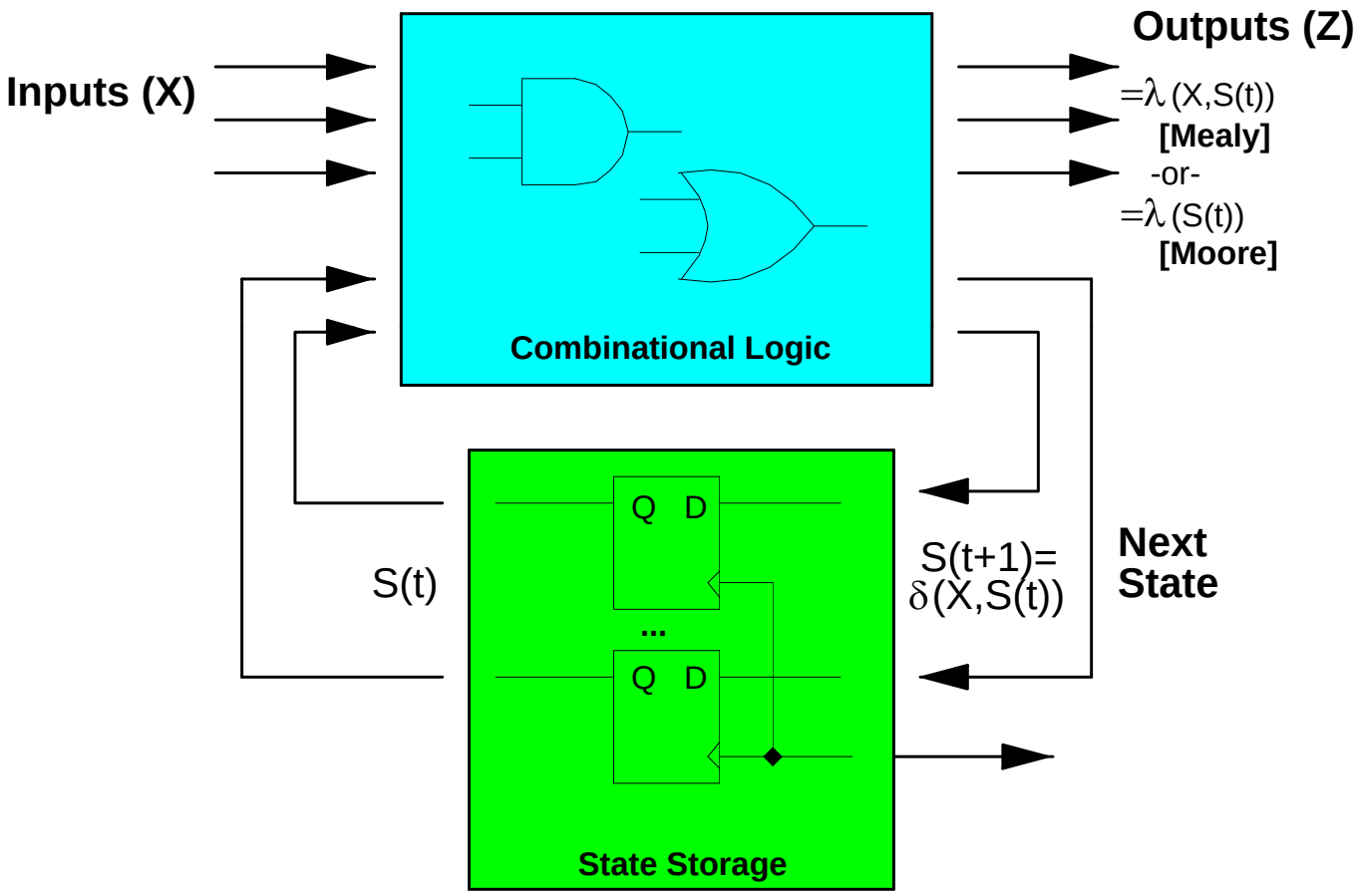
- Also called Function Generators (FGs)
- Capacity is limited only by number of inputs, not complexity
- Delay through the LUT is constant



A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
	.	.	.	
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Diagram From: Xilinx, Inc

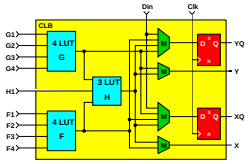
Finite State Machines



Field Programmable Gate Arrays

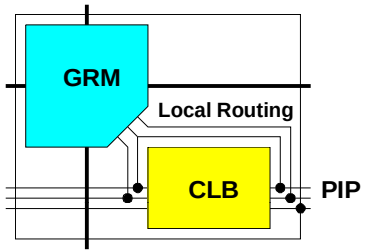
CLB

- Primitive element of FPGA



Routing Module

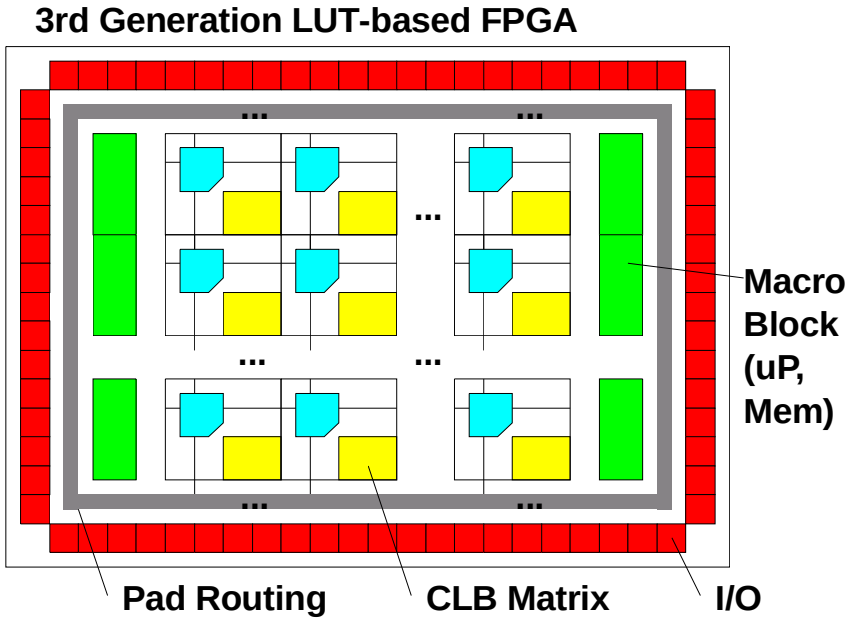
- Global routing
- Local interconnect



Macro Blocks

- Block Memories
- Microprocessor

I/O Block



NetFPGA 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)

Structure of Libray Source Code

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)

Structure of Project Source Code

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)

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

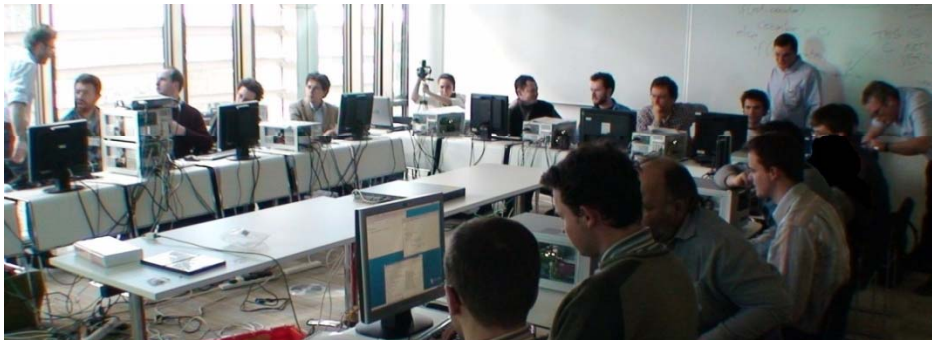
Welcome to the Worldwide Community

NetFPGA @ SIGCOMM - Seattle, WA

Workshop in Beijing, China

NetFGPA @ SIGMETRICS - San Diego, CA

Workshop in Bangalore, India



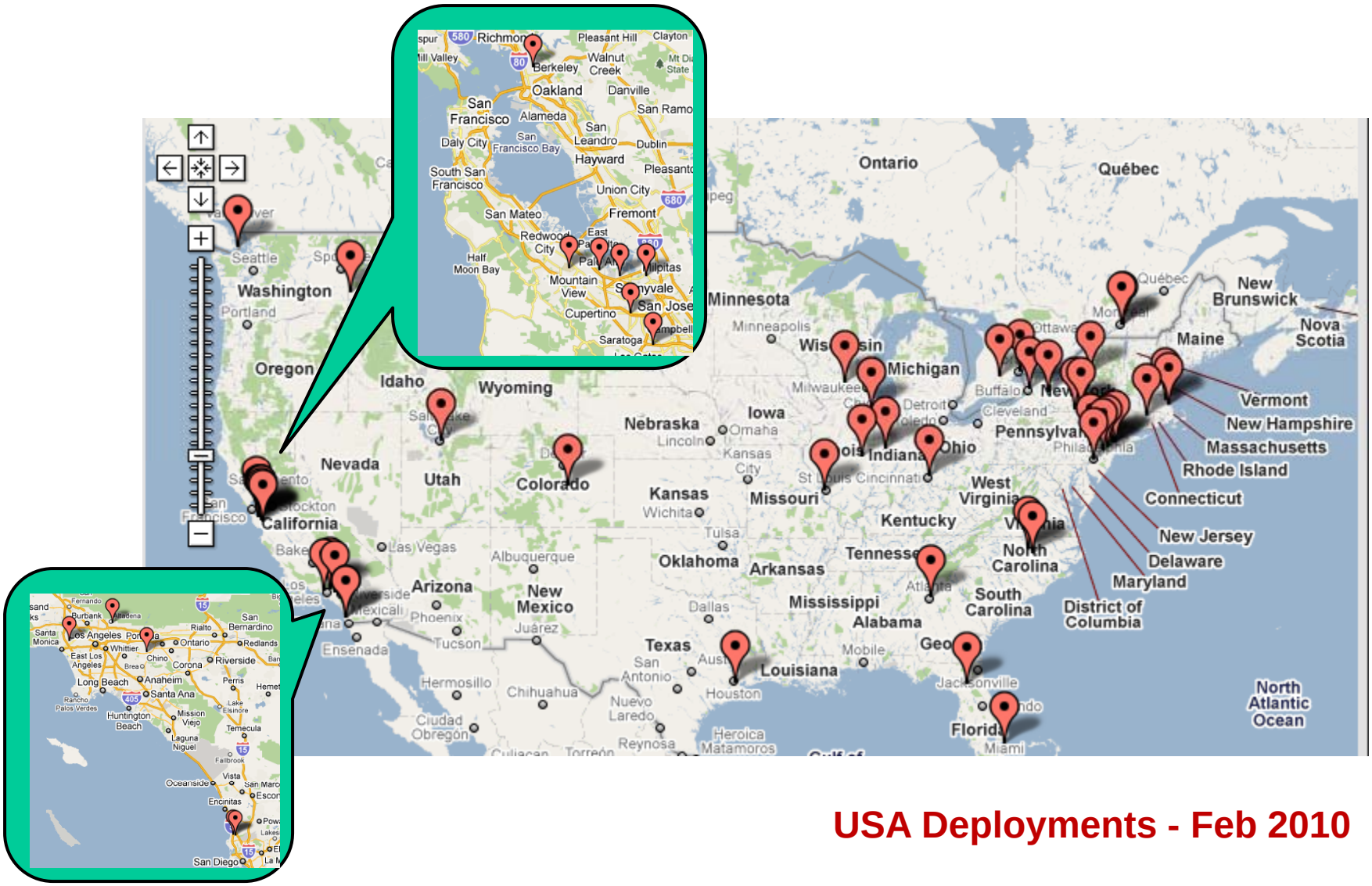
EuroSys - Glasgow, Scotland, U.K.

NetFPGA Deployments

– Over 1,350 NetFPGA users with 1,300+ cards deployed at 150+ universities in 17 Countries worldwide

Worldwide Hardware Deployments - Feb 2010

NetFPGA Hardware in North America

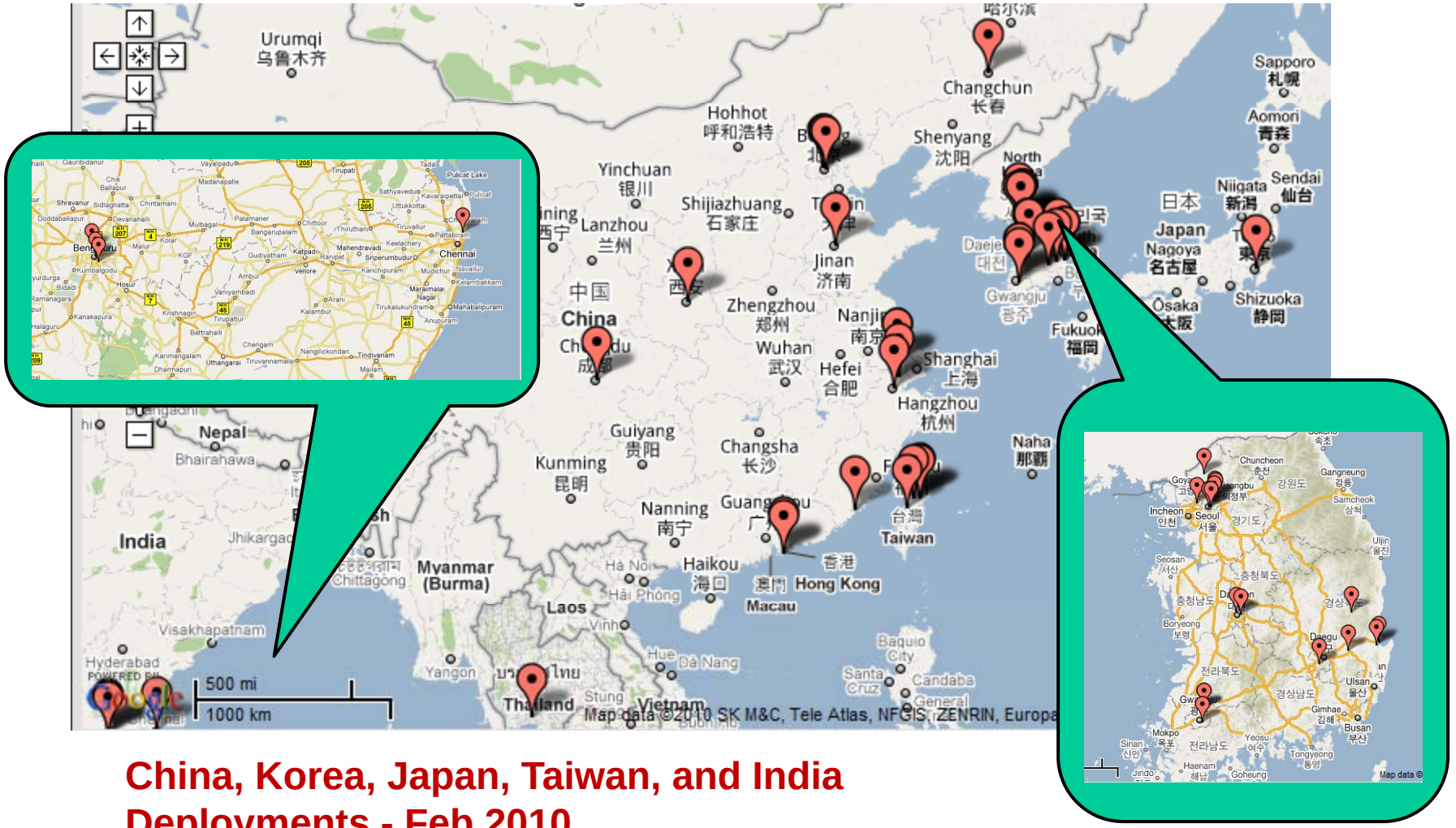


USA Deployments - Feb 2010

NetFPGA Hardware in Europe

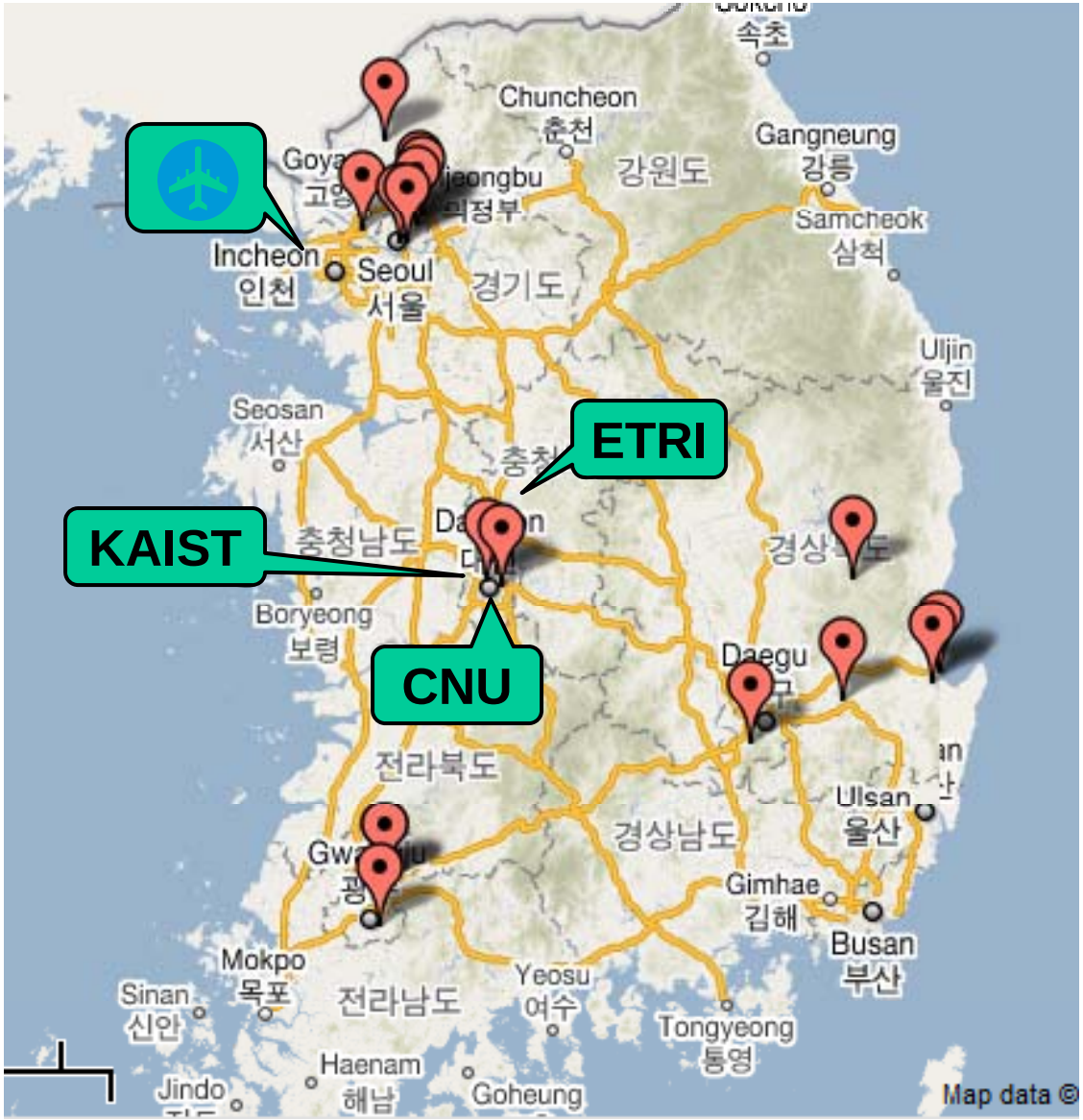
European Deployments - Feb 2010

NetFPGA Hardware in Asia



China, Korea, Japan, Taiwan, and India Deployments - Feb 2010

NetFPGA Deployments in Korea



Contributed projects

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

… continued on next page …

As of Feb 2010

.. And more on http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/ProjectTable

NetFPGA Designs (continued..)

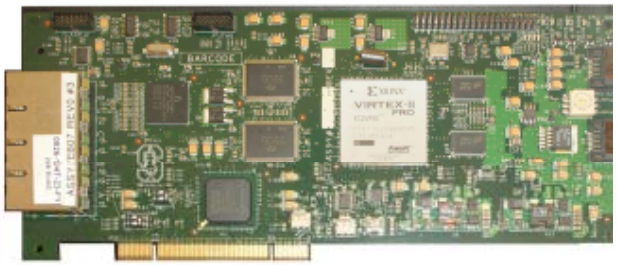
… continued from previous page …

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

.. And more on http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/ProjectTable

Other Network FPGA Platforms

- **NetFPGA 1G**
  - 4 * 1 GigE
  - 4 Gbps



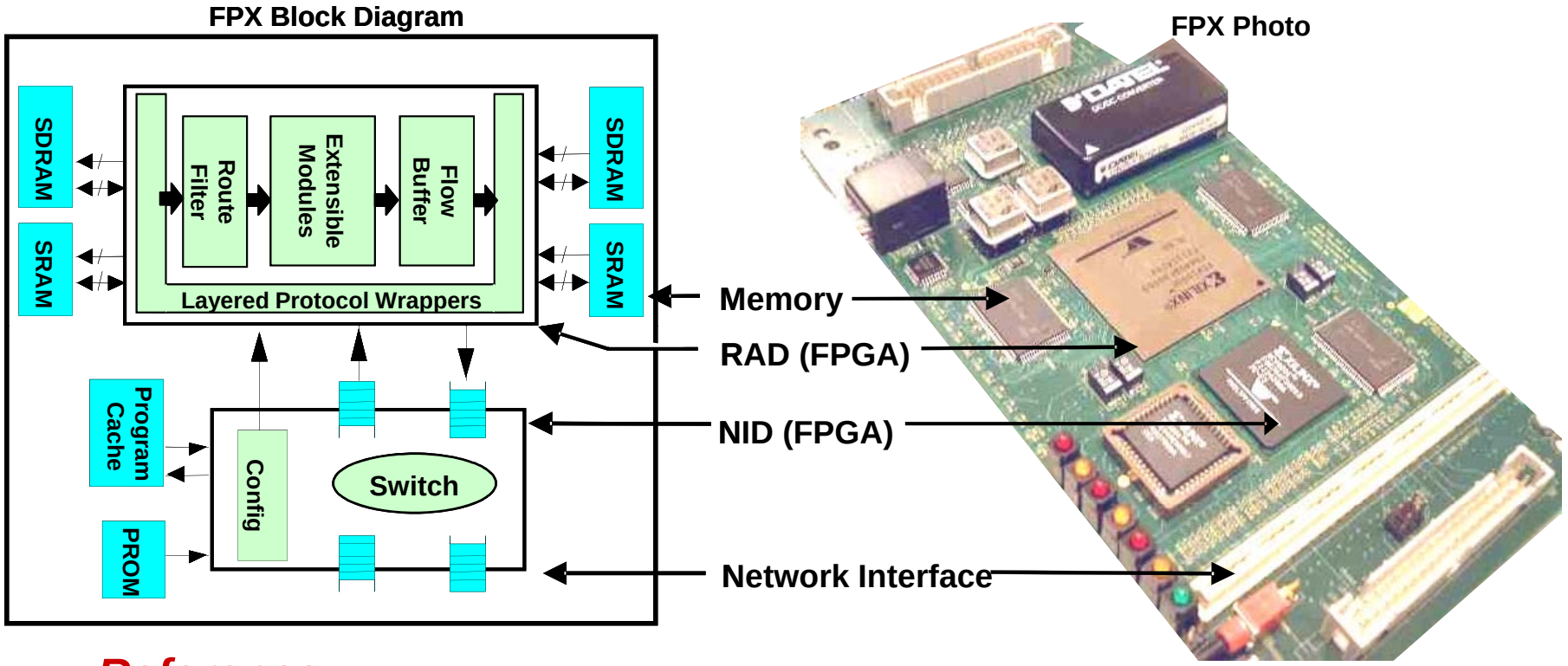
- **FPX**
  - 2 * OC-48
  - 4.8 Gbps

- **Upcoming 10G**
  - 4 * 10GE
  - 40 Gbps

Motivation for Network Security

- **Viruses can be costly to businesses**
  - Annoyance to average users
  - Use networks to propagate
- **Sample Attacks**
  - Nimda, Code Red, Slammer
  - MSBlast
    - Infected over 350,000 hosts
  - SoBigF
    - Infected 1 million users in first 24 hours
    - Infected > 200 million in the first week
    - Caused an estimated $1 billion in damages to repair.
- **End-systems difficult to maintain**
  - Operating systems become outdated
  - Users introduce new machines on network

Example of SOC Firewall-FPX Platform

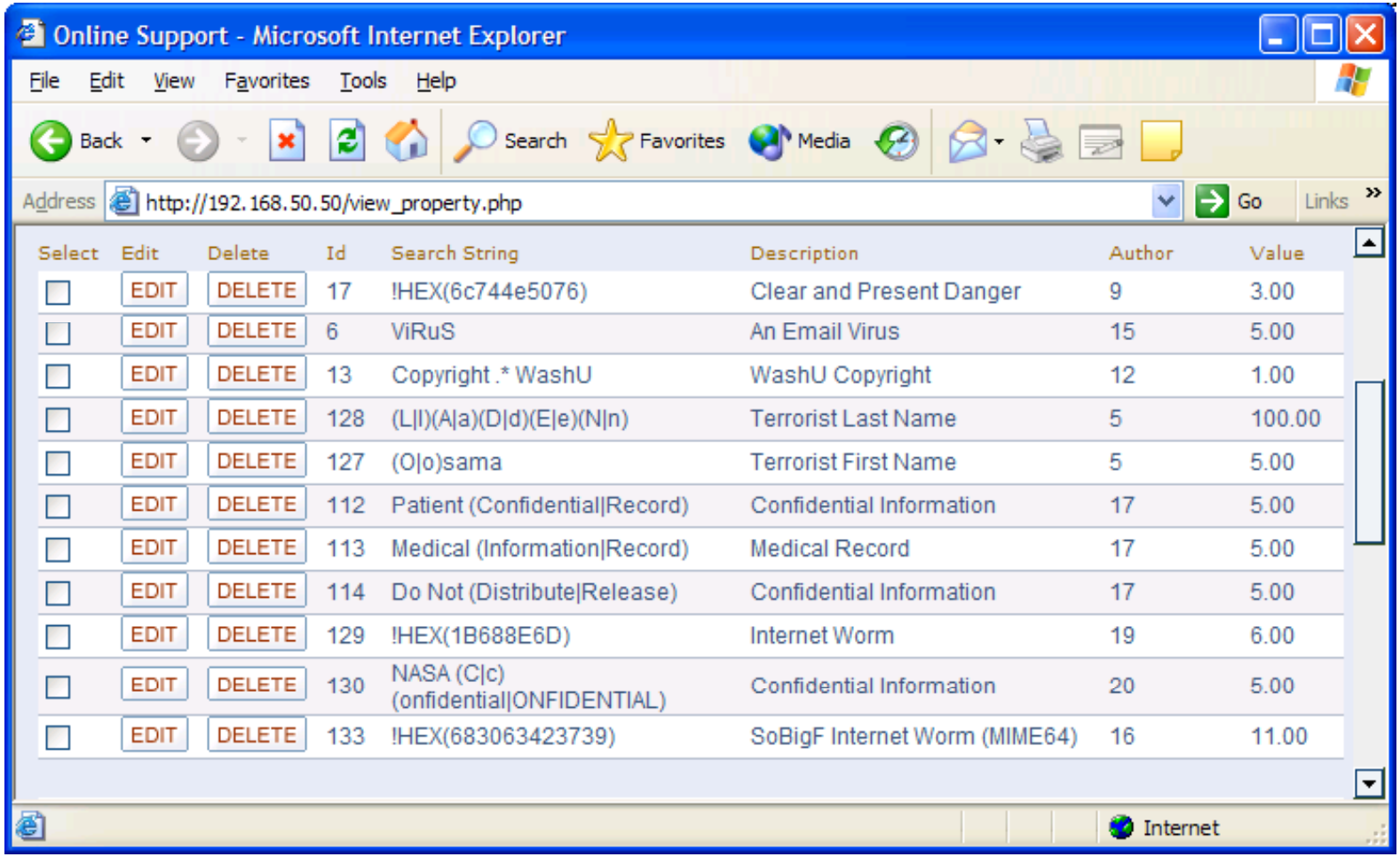


- ***Reference***
  - *An Extensible, System-On-Programmable-Chip, Content-Aware Internet Firewall*, by John W. Lockwood, Christopher Neely, Christopher Zuver, James Moscola, Sarang Dharmapurikar, and David Lim; Field Programmable Logic and Applications (FPL), Lisbon, Portugal, Paper 14B, Sep 1-3, 2003.
- **More Information:**
  - http://www.reprogrammablenetworks.com/

Regular Expression Matching

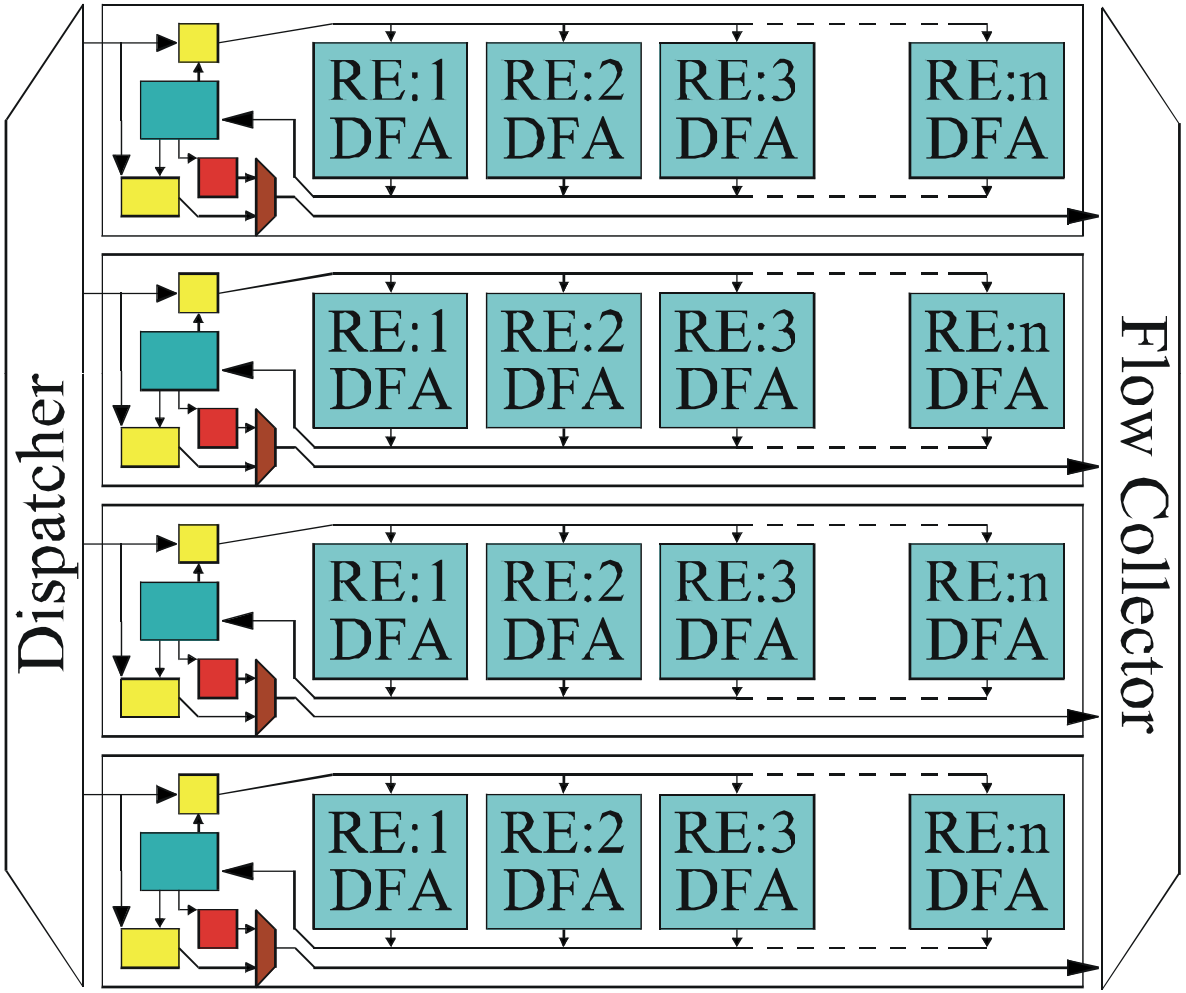
- **String Matching**
  - Allow for matching of fixed-length strings
    - HEX(683063423739) finds SoBig.F
- **Regular Expressions**
  - Allows for a wide range of matches
    - Case variations : (W|w)(A|a)(R|r)(H|h)(O|o)(L|l)
    - Wild-card characters: Albert ? Einstein
    - Strings of wildcards : A.* Einstein
  - Content may begin anywhere in packet
- **Morphable Patterns**
  - Will require systems that can evolve

GUI to Select Search Strings

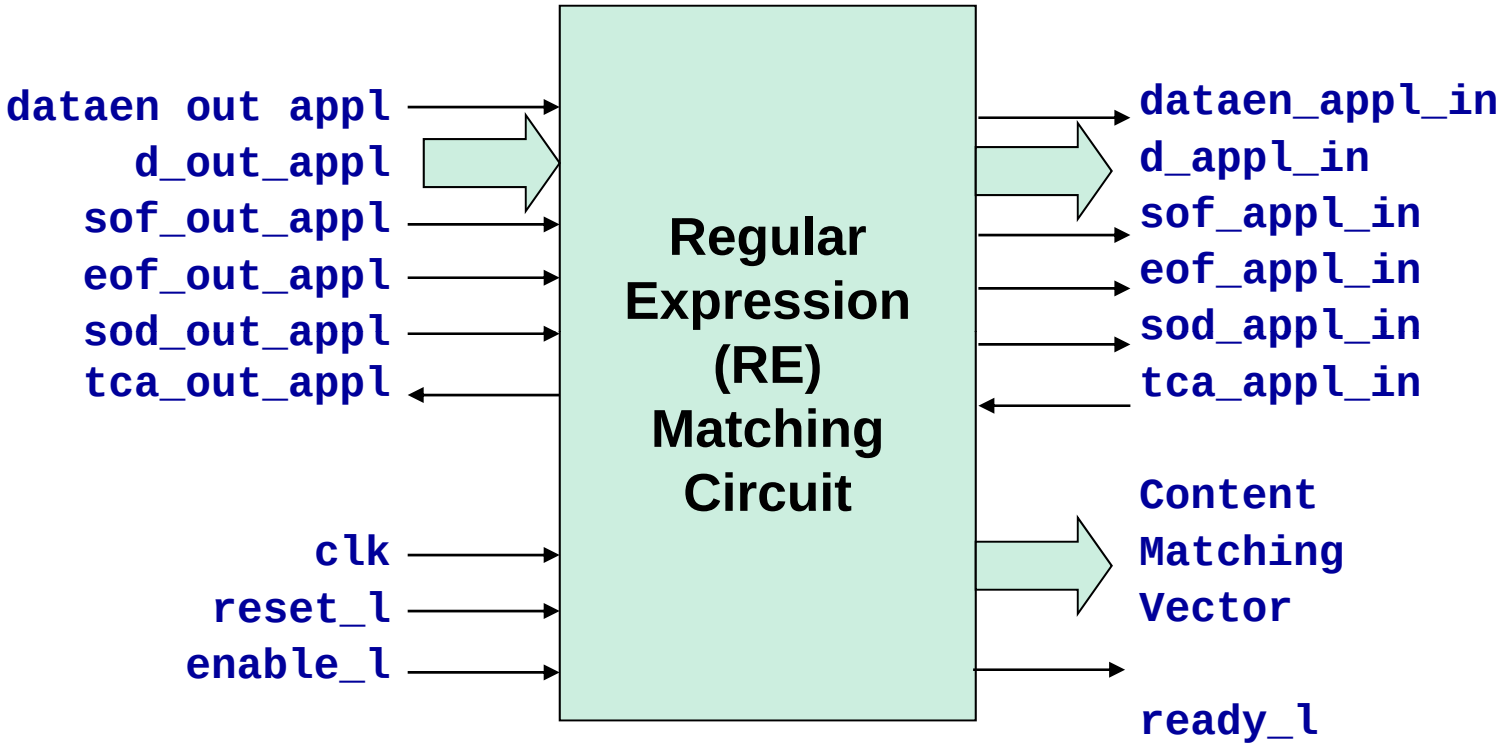


Select	Edit	Delete	Id	Search String	Description	Author	Value
☐	EDIT	DELETE	17	!HEX(6c744e5076)	Clear and Present Danger	9	3.00
☐	EDIT	DELETE	6	ViRuS	An Email Virus	15	5.00
☐	EDIT	DELETE	13	Copyright .* WashU	WashU Copyright	12	1.00
☐	EDIT	DELETE	128	(L|l)(A|a)(D|d)(E|e)(N|n)	Terrorist Last Name	5	100.00
☐	EDIT	DELETE	127	(O|o)sama	Terrorist First Name	5	5.00
☐	EDIT	DELETE	112	Patient (Confidential|Record)	Confidential Information	17	5.00
☐	EDIT	DELETE	113	Medical (Information|Record)	Medical Record	17	5.00
☐	EDIT	DELETE	114	Do Not (Distribute|Release)	Confidential Information	17	5.00
☐	EDIT	DELETE	129	!HEX(1B688E6D)	Internet Worm	19	6.00
☐	EDIT	DELETE	130	NASA (C|c) (onfidential|ONFIDENTIAL)	Confidential Information	20	5.00
☐	EDIT	DELETE	133	!HEX(683063423739)	SoBigF Internet Worm (MIME64)	16	11.00

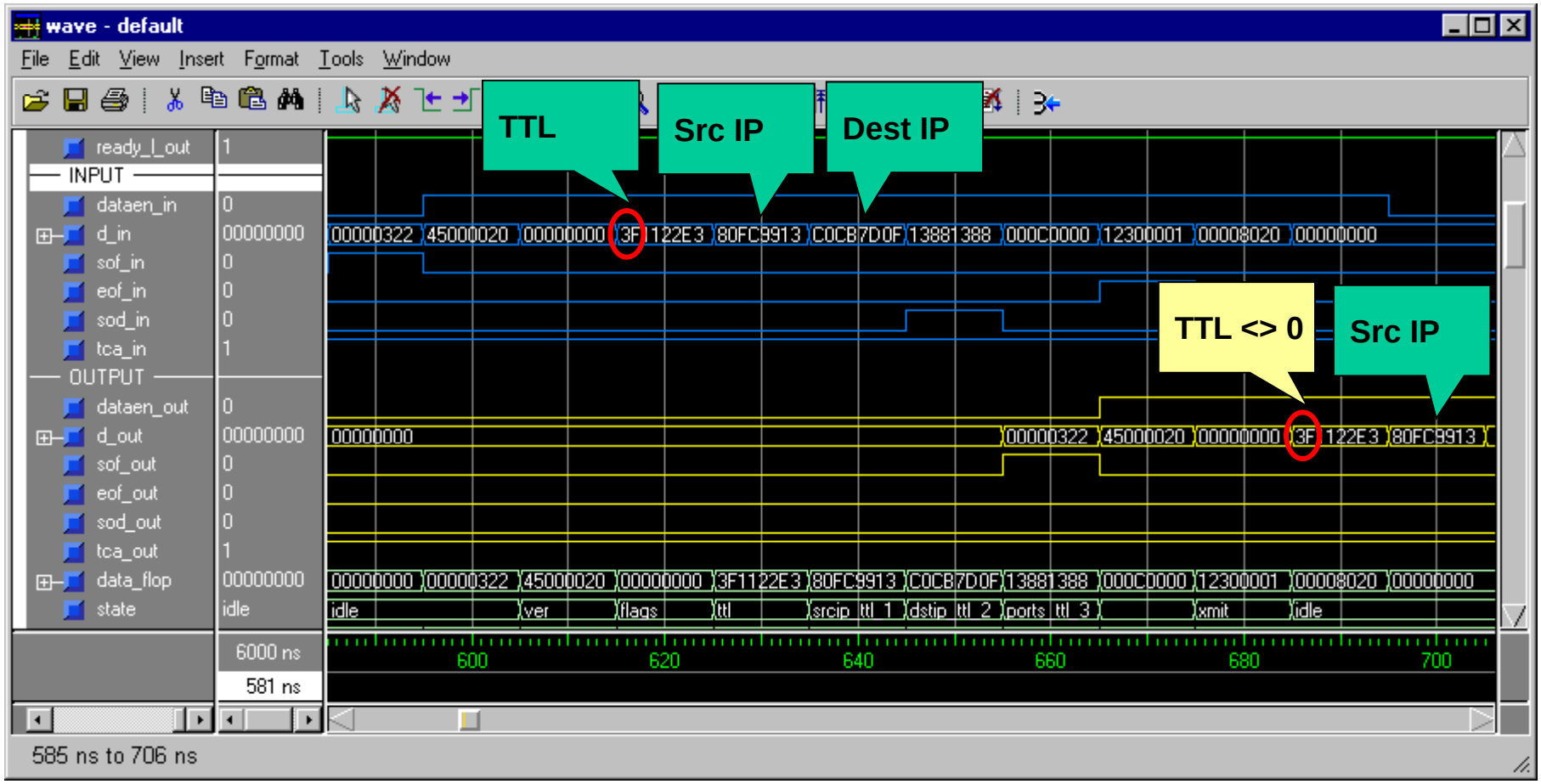
Compiler to build Parallel Scan Engines



Content Matching Module



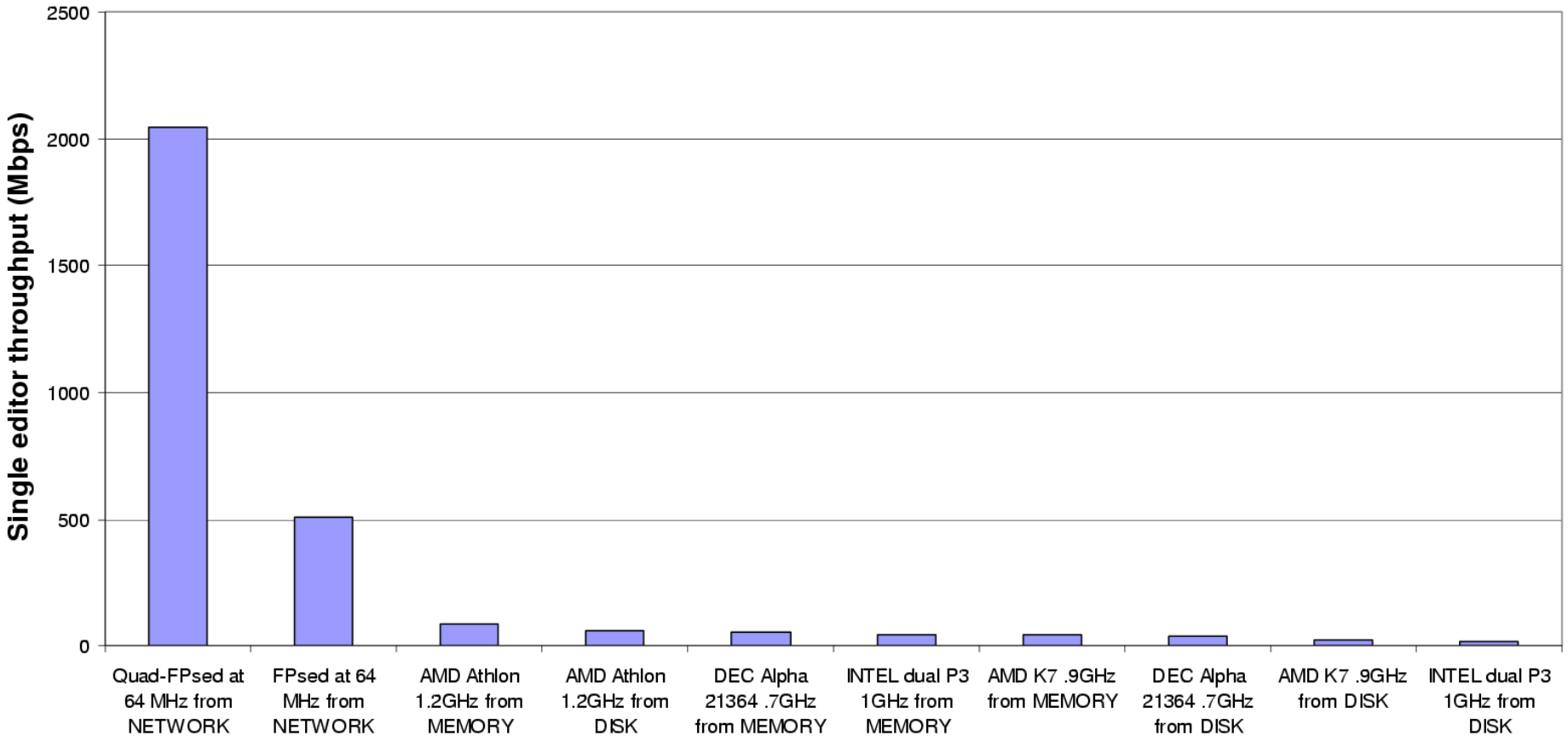
Sample ModelSim Waveform



Throughput Comparison

- **Sed was run on different Linux PCs**
  - Dual Intel Pentium III @ 1 GHz
    - 13.7 Mbps when data is read from disk
    - 32.72 Mbps when data is read from memory
  - Alpha 21364 @ 667 MHz
    - 36 Mbps when data is read from disk
    - 50.4 Mbps when data is read from memory
- **Software results are 40x slower than FPsed**

String Processing Benchmarks

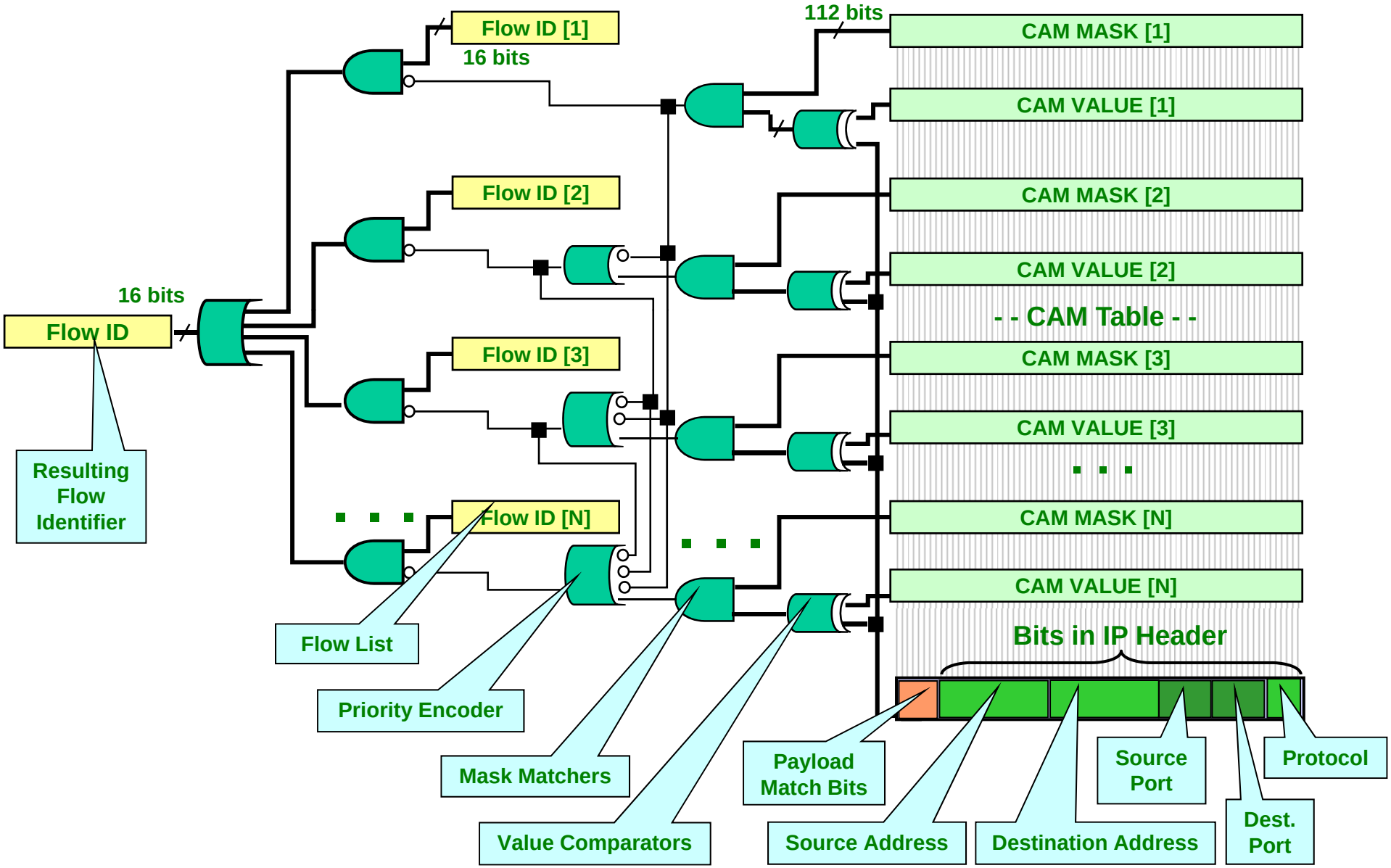


Associative Match in FPGA TCAM

All values shown In hex

- **Content :** Content Vector = “00000011” *(binary)* = x”03” *(hex)*
- **Source IP Address** : 128.252.5.5 *(dotted.decimal)*
- **Destination IP Address** : 141.142.2.2 *(dotted.decimal)*
- **Source Port** : 4096 *(decimal)* = 1000 *(hex)*
- **Destination Port** : 80 *(decimal)* = 50 *(hex)*
- **Protocol** : TCP (6)

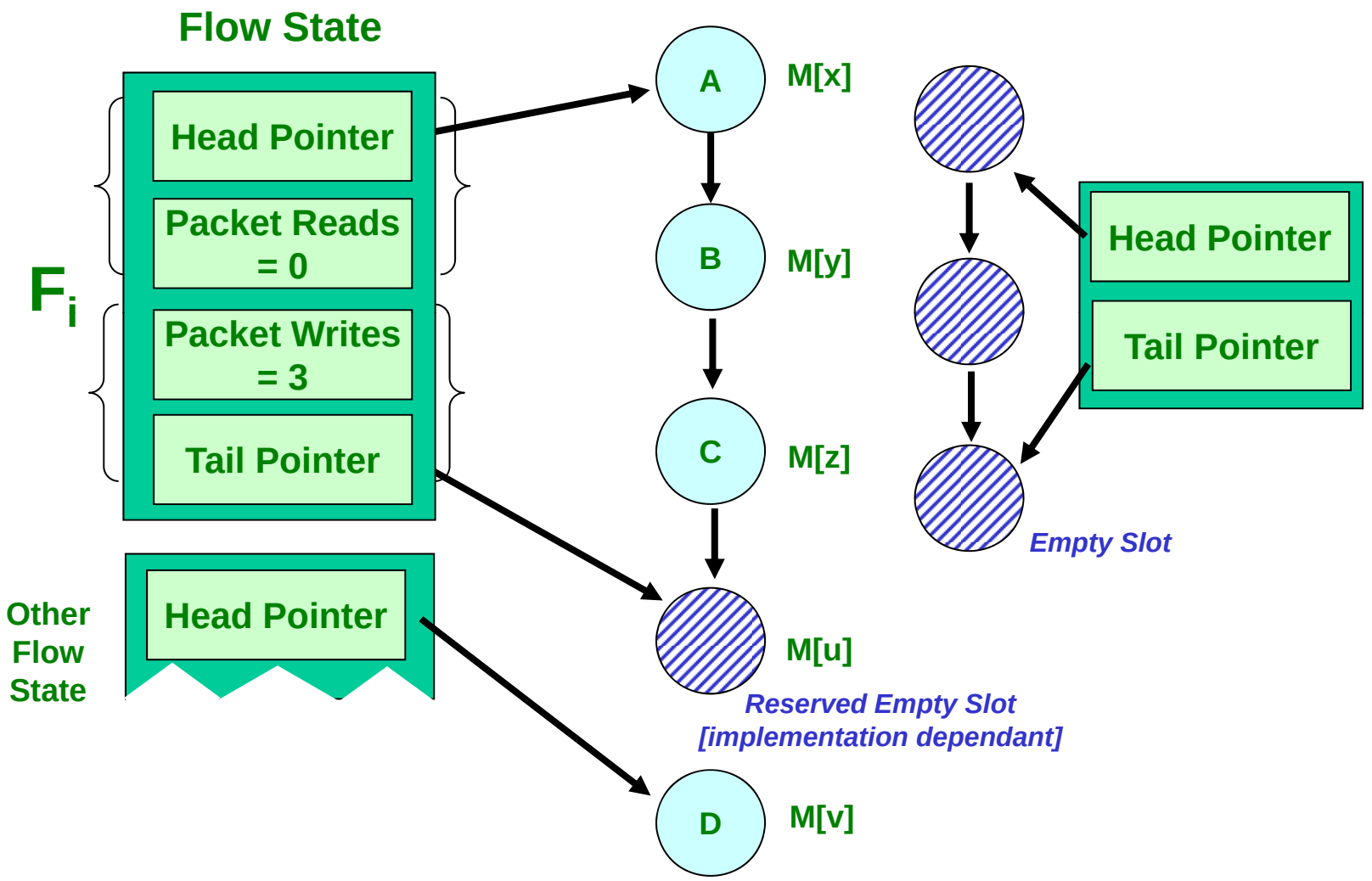
TCAM Function in FPGA Hardware



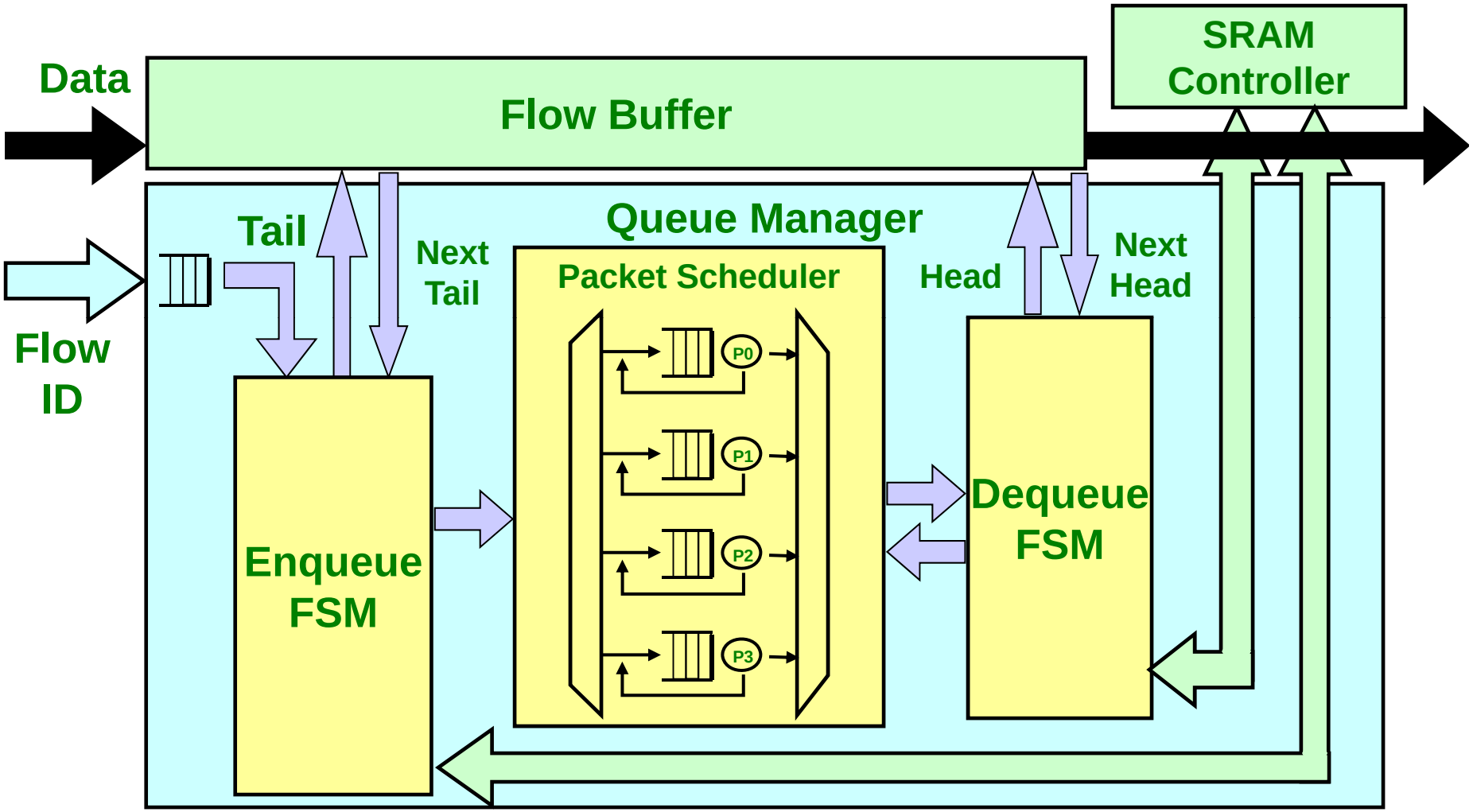
Data Buffering & Per-flow Queuing

- **Flow Buffering**
  - Maintain linked list of packets
  - Track head and tail pointers for each list
  - Track free memory
- **Queue Management**
  - Enqueue packet
  - Dequeue packet
- **Schedule Traffic Flows**

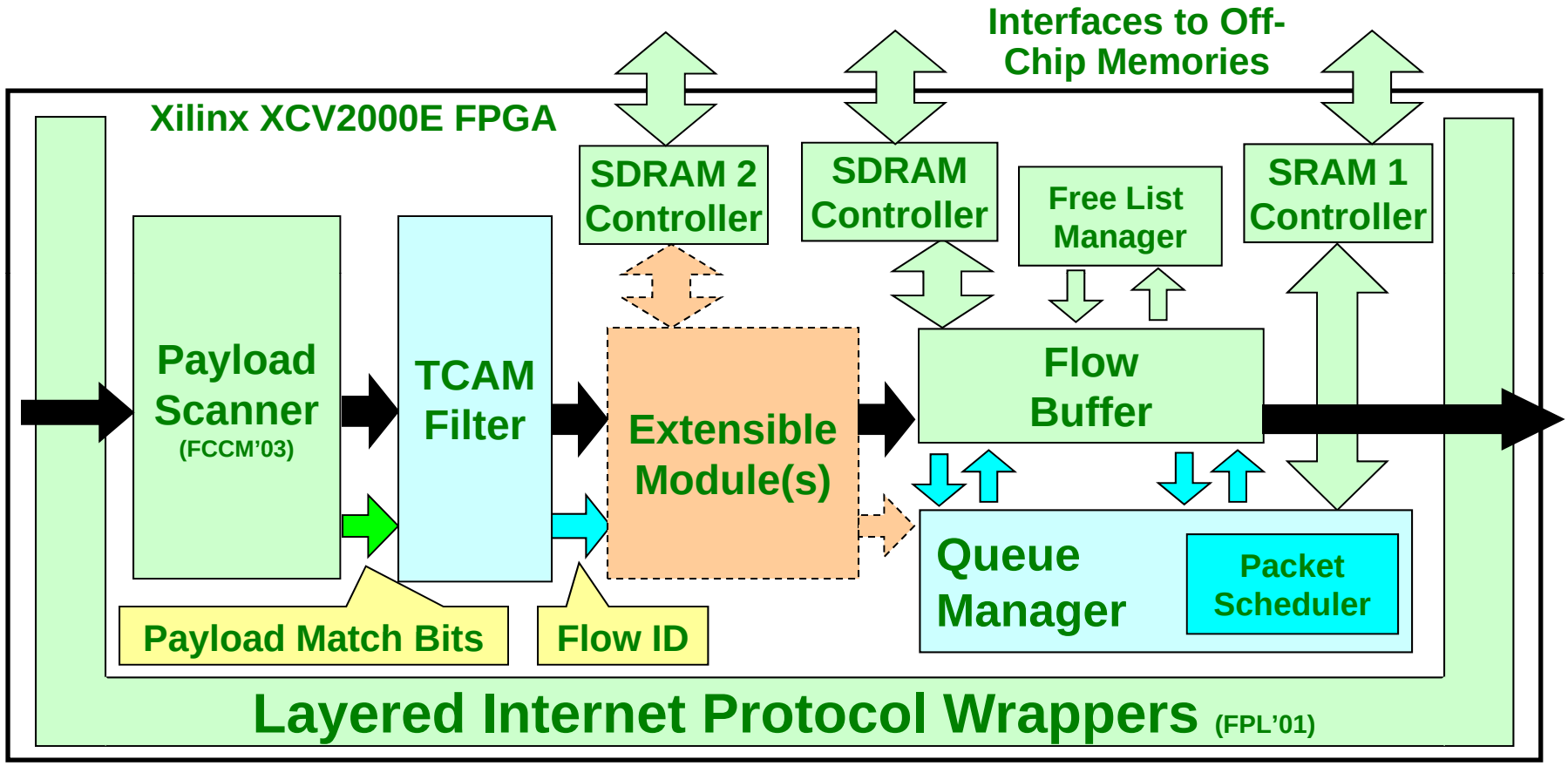
Flow Buffering



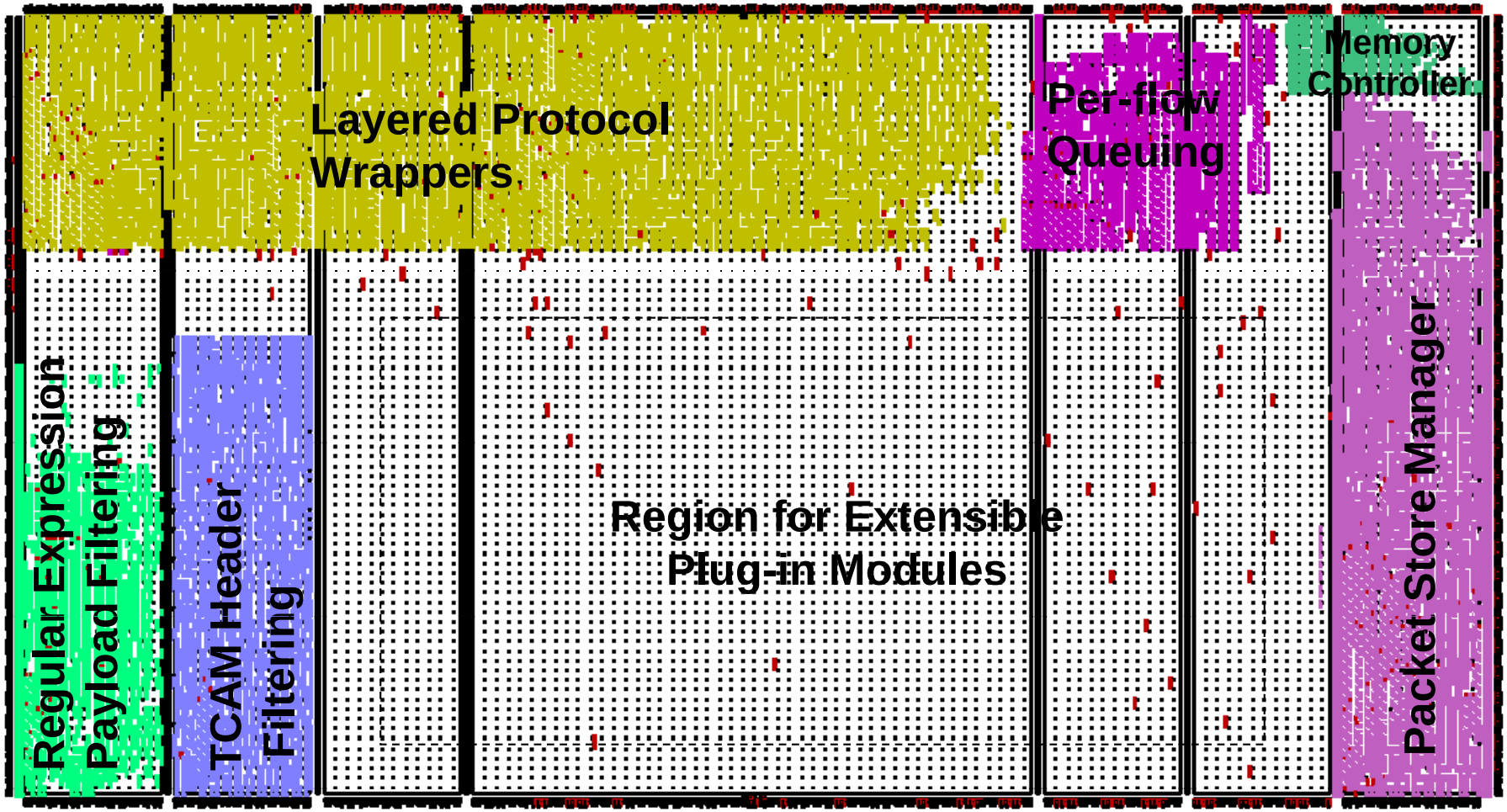
Queue Management



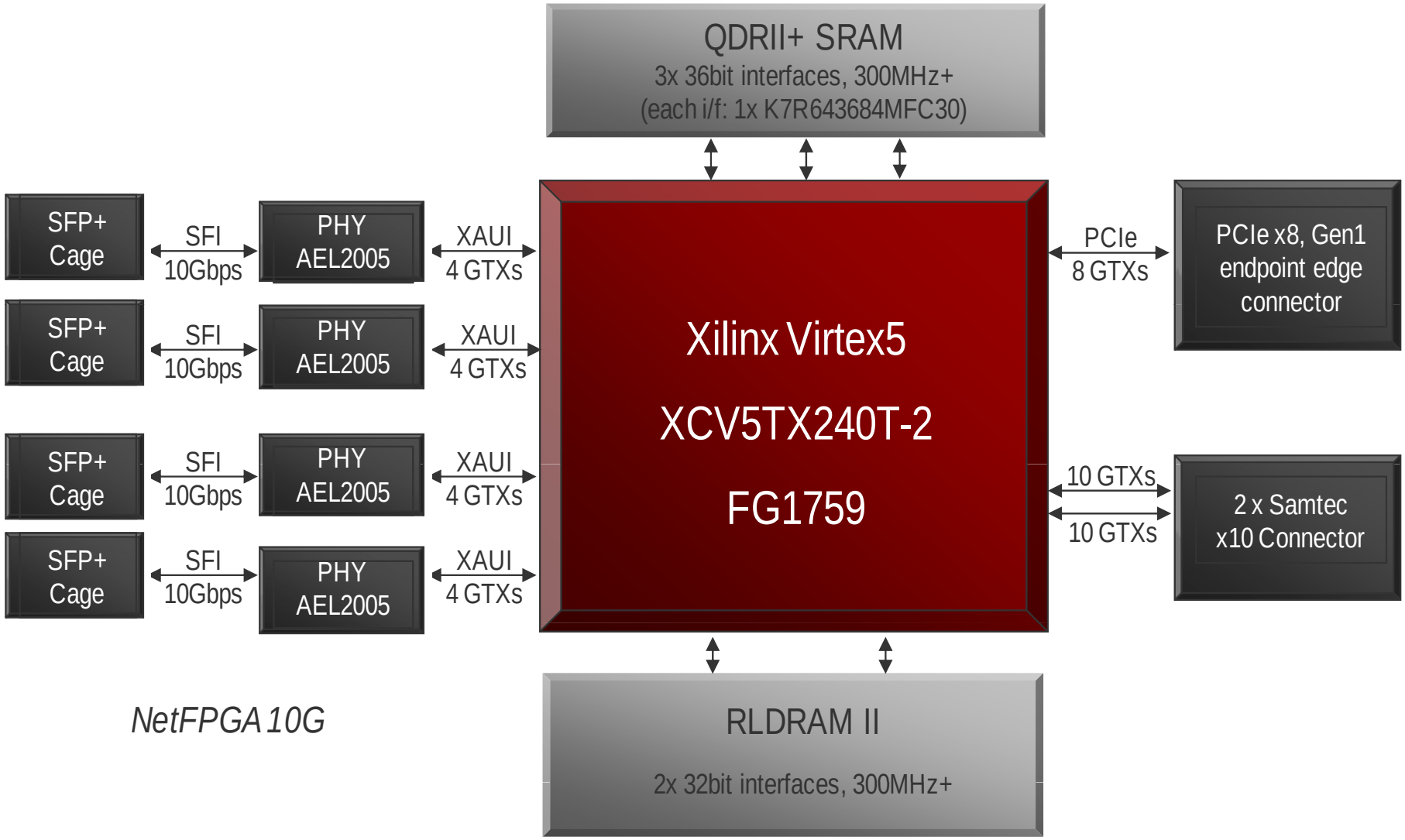
Architecture of the SOC Firewall



Resulting SOC-Firewall FPGA Layout



Upcoming NetFPGA 10G (4 * 10 GE)



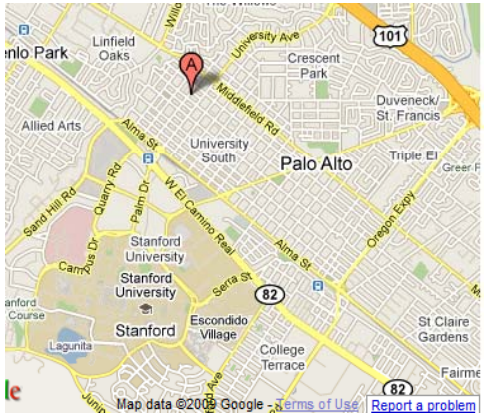
NetFPGA 10G

40 Gbps NetFPGA PCB

Algo-Logic Systems

- **Founded by Dr. John W. Lockwood, 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

Consulting Services

- ***Design and Build Hardware-Accelerated Network Systems***

  - 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

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 Network**
  - 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.

Extended Training

- ***Customized 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

1st Asia NetFPGA Developer Workshop

June 13-14, 2010 @ KAIST, Daejeon, Korea

http://fif.kr/AsiaNetFPGAws

- **Program Committee Chairs**
  - John W. Lockwood
    - (Algo-Logic Systems)
  - Seung-Joon Seok
    - (Kyungnam University)
- **Local Arrangement Co-Chairs**
  - Sue Moon
    - (KAIST)
  - Jaeyong Lee
    - (Chungnam National Univ.)
- **Advisory**
  - Nick McKeown
    - (Stanford University)
  - Dae Young Kim
    - (Chungnam University)
- **Many Thanks**
  - Xilinx
  - NetFPGA Group
  - Sponsors:

NetFPGA

FIF Future Internet Forum

ETRI Electronics and Telecommunications Research Institute

충남대학교 인터넷침해대응기술연구센터

ALGORITHMS IN LOGIC ALGO-LOGIC HTTP://ALGO-LOGIC.COM

Program Committee Members

- **Jun Bi**
  - (Tsinghua University)
- **Gordon Brebner**
  - (Xilinx)
- **Michael Ciesla**
  - (UNSW)
- **Yu-Huang Chu**
  - (Chunghwa Telecom)
- **Sun Moo Kang**
  - (NIA)
- **Eric Keller**
  - (Princeton University)
- **Jongwon Kim**
  - (GIST)
- **Taekyoung Kwon**
  - (Seoul National Univ.)
- **Martin Labrecque**
  - (University of Toronto)
- **Yukuen Lai**
  - (Chung-Yuan Christian Univ.)
- **Jaeyong Lee**
  - (Chungnam National Univ.)
- **John W. Lockwood**
  - (Algo-Logic Systems)
- **Kevin Xie**
  - (Xilinx)
- **Sue Moon**
  - (KAIST)
- **Andrew, W. Moore**
  - (Cambridge University)
- **Akihiro Nakao**
  - (University of Tokyo)
- **Seung Yeob Nam**
  - (Yeungnam University)
- **Seung-Joon Seok**
  - (Kyungnam University)
- **Seung-Yong Park**
  - (Yonsei University)
- **Hwangjun Song**
  - (POSTECH)
- **Charlie Wiseman**
  - (Washington University)
- **Chong Ho Yoon**
  - (Korea Aerospace Univ.)
- **Martin Zadnik**
  - (Brno University of Tech.)
- **Hongyi Zeng**
  - (Stanford University)
- **Young Cho**
  - (USC-ISI)

Welcome


uter Science Department
NetFPGA
1st Asia NetFPGA Developers' Workshop
June 14, 2010
KAIST, KOREA
Sponsored by Future Internet Forum, ALGO-LOGIC, ETRI, CNU IIRTRC, KAIST CASFI