Fast Flexible FPGA-Tuned Networks-on-Chip

Michael K. Papamichael <papamix@cs.cmu.edu>, **James C. Hoe** <jhoe@ece.cmu.edu>

Computer Architecture Lab at Carnegie Mellon

http://www.ece.cmu.edu/~mpapamic/connect

Networks-on-Chip (NoCs) and FPGAs

Rapid growth of FPGA capacity & features

- Extended SoC and full-system prototyping
- FPGA-based high-performance computing

→**Need for flexible NoCs to support communication**

Map existing ASIC-oriented NoC designs on FPGAs?

- Inefficient use of FPGA resources
- ASIC-driven NoC arch. not optimal for FPGA

CONNECT FPGA-tuned NoC Architecture

- Embodies FPGA-motivated design principles
- Very lightweight, minimizes resource usage
- Publicly released flexible NoC generator

The CONNECT NoC Architecture

FPGAs peculiar HW realization substrate

- Relative cost of speed vs. logic vs. wires vs. mem.
- Unique mapping and operating characteristics

CONNECT focuses on 4 FPGA characteristics:

1. Abundance of Wires
2. Storage Shortage & Peculiarities
3. Frequency Challenged
4. Reconfigurable Nature

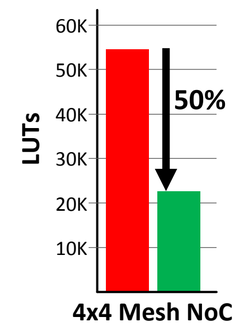
FPGA characteristics uniquely influence NoC design decisions, which often go against ASIC-driven NoC conventional wisdom.

CONNECT vs. ASIC-Oriented RTL

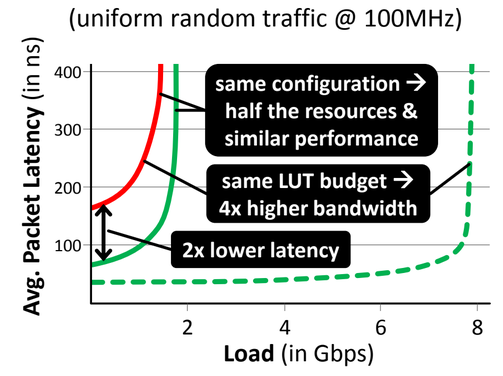
16-node 4x4 Mesh Network-on-Chip (NoC)

- **SOTA:** state-of-the-art high-quality ASIC-oriented RTL*
- **CONNECT:** identical config. CONNECT-generated RTL

FPGA Resource Usage (same NoC configuration)



Network Performance (uniform random traffic @ 100MHz)



*NoC RTL from http://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/Router

The CONNECT Approach: Tailoring NoCs to FPGAs

1 Abundance of Wires

- **Densely connected wiring substrate**
  - (Over)provisioned to handle worst case
  - Wires are "free" comp. to other resources

CONNECT NoC Implications

→ **Datapaths & channels as wide as possible**

→ **Adjust packet format**

E.g. carry control info on the dedicated side links

→ **Adapt traditional credit-based flow control**

"Peek" flow control in CONNECT uses wider links

2 Storage Shortage & Peculiarities

- **Modern FPGAs offer storage in two forms**
  - Block RAMs and LUT RAMs (use logic resources)
  - Only come in specific aspect ratios and sizes
- **In high demand, especially Block RAMs**

CONNECT NoC Implications

→ **Optimize for aspect ratios and sizes**

Multiple logical flit buffers in each physical buffer

→ **Use LUT RAM for flit buffers**

Leave Block RAM resources for rest of design

3 Frequency "Challenged"

- **Much lower frequencies compared to ASICs**
  - LUTs inherently slower than ASIC standard cells
  - Large wire delays when chaining LUTs
- **Rapidly diminishing returns of pipelining**

CONNECT NoC Implications

→ **Design router as single-stage pipeline**

Also dramatically reduces network latency

→ **Adjust network to meet perf. goals**

E.g. increase link width or adapt topology

4 Reconfigurable Nature

- **Reconfigurable nature of FPGAs**
  - Sets them apart from ASICs
  - Allows support for diverse range of applications

CONNECT NoC Implications

→ **Support application-specific customization**

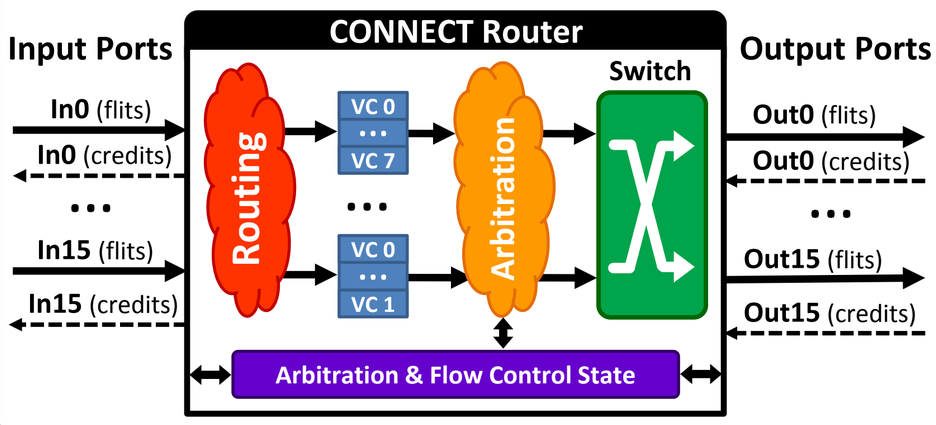
Flexible parameterized NoC architecture

Automated NoC design generator (demo!)

→ **Adhere to standard common interface**

NoC appears as plug-and-play black box

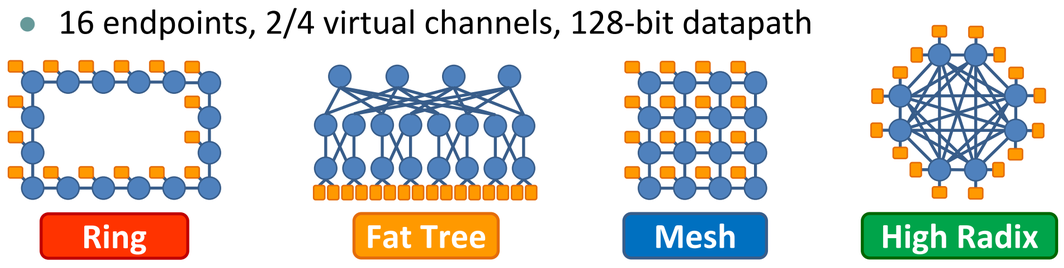
CONNECT Architecture



- **Topology-Agnostic Architecture**
- **Fully parameterized, including:**
  - # in/out ports, # virtual channels
  - flit width, buffer depths
  - Flexible user-specified routing
  - Four allocation algorithms
  - Two flow-control mechanisms
- **"Virtual Link" Support**
  - Contiguous delivery of multi-flit packets
- **FPGA-friendly "Peek" Flow Control**
  - Lightweight alternative to credit-based

CONNECT Network Examples and Results

- **Four sample CONNECT Networks** (router, endpoint)
  - 16 endpoints, 2/4 virtual channels, 128-bit datapath



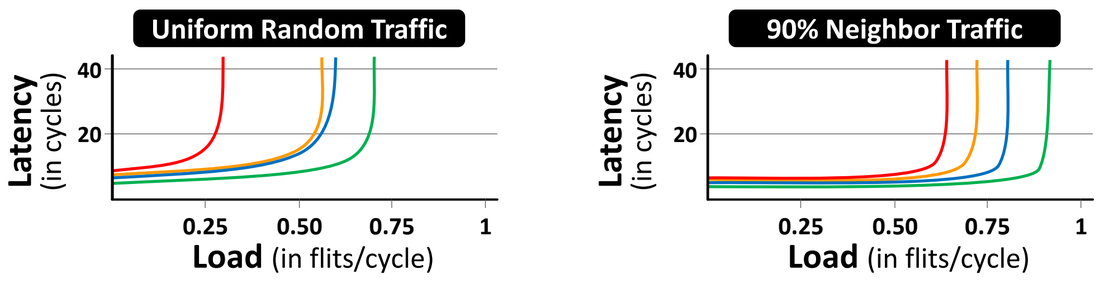
All above networks are interchangeable from user perspective.

FPGA Synthesis Results

Network	Network Configuration			Xilinx Virtex-6 LX760T	
	# Routers	# VCs	Ports/Router	% LUTs	Freq. (in MHz)
Ring	16	4	3	3%	158
Fat Tree	20	4	4	4%	143
Mesh	16	4	5	5%	113
High Radix	8	2	9	9%	75

< 10% LUT Utilization for all networks, No Block RAMs

Network Performance Results



There is no one-size-fits-all NoC! Tune NoC to application.

Please see our FPGA 2012 paper for more synthesis and performance results.

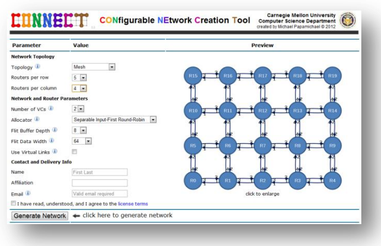
Conclusions & Public Release

- **Significant gains from tuning for FPGA**
  - FPGAs and ASICs have different design "sweet spot"
- **Compared to ASIC-driven NoCs, CONNECT offers**
  - Significantly lower network latency **and**
  - ~50% lower LUT usage or 3-4x higher network performance
- **Take advantage of reconfigurable nature of FPGA**
  - Tailor NoC to specific communication needs of application

CONNECT NoC Generator Public Release

http://www.ece.cmu.edu/~mpapamic/connect

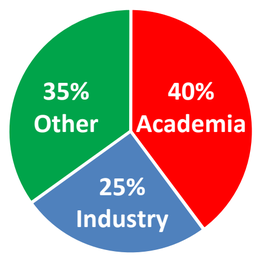
- **NoC Generator with web-based interface**
  - Supports multiple pre-configured topologies
  - Includes graphical custom topology editor
- **Some Release Stats** (since release in March 2012)
  - **2000+** unique visitors
  - **200+** network generation requests

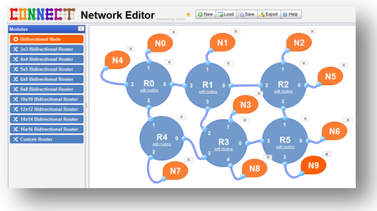


Most Popular Topologies

1. Mesh/Torus – 51%
2. Double Ring – 14%
3. Ring/Line – 14%
4. Fully Connected – 10%
5. Custom – 6%
6. Star – 5%

User Breakdown





This work was funded by NSF. We thank Xilinx for their FPGA and tool donations. We thank Bluespec for their tool donations. We also thank Derek Chiou, Daniel Becker and the Stanford CVA group.