EECS251B
Advanced Digital Circuits and Systems

Lecture 5&6 – System Interconnect

Vladimir Stojanović

Tuesdays and Thursdays 9:30-11am
Cory 521
Power and Performance Trends

- With clock frequencies saturating CPUs, started using many cores to leverage parallelism and deal with fabrication yields.
Manycore System Roadmap

64-tile system (64-256 cores)
- 4-way SIMD FMACs @ 2.5 – 5 GHz
- 5-10 TFlops on one chip
- Need 5-10 TB/s of off-chip I/O
- Even larger bisection bandwidth

Number of cores

1 2 3 4 5 6 7 8 9
128 256 512

- 2 cm x 2 cm

Intel 48 core - Xeon
Intel Flops
Tilera TILE64
NVIDIA GT200
Sun Niagara
Cavium Octeon
Raza XLR
Cell
Rock
Niagra
Barcelona
Neha
IBM CELL
DRAM interface
bus
Power4
Opteron
Core2
Power6
Itanium
P4
Athlon
P3
P2
Pentium
486
386
286
The rise of manycore machines

Only way to meet future system feature set, design cost, power, and performance requirements is by programming a processor array

- Multiple parallel general-purpose processors (GPPs)
- Multiple application-specific processors (ASPs)

"The Processor is the new Transistor" [Rowen]

Intel 4004 (1971):
- 4-bit processor,
- 2312 transistors,
- 100 KIPS,
- 10 micron PMOS,
- 11 mm² chip

1000s of processor cores per die

Sun Niagara
- 8 GPP cores (32 threads)

IBM Cell
- 1 GPP (2 threads)
- 8 ASPs

Intel Network Processor
- 1 GPP Core
- 16 ASPs (128 threads)

Cisco CRS-1
- 192 Tensilica GPPs

Picochip DSP
- 1 GPP core
- 248 ASPs

“The Processor is the new Transistor” [Rowen]
Interconnect bottlenecks

Bottlenecks due to energy and bandwidth density limitations

Need to jointly optimize on-chip and off-chip interconnect network
Scaling to many cores

• Networks-on-chip
  • Many meshes
    • Slow, latency varies greatly
  • Easy to implement
  • Large crossbars
    • Fast, predictable latency
  • Hard to build and scale
  • Rings

TILE64 [Bell08]
Rainbow-Falls 2-stage Crossbar

Bisection Bandwidth
461GB/s

[Patel09]
Recent trends

[Cerebras Systems] WSE-2
2.6T Transistors
850,000 AI optimized cores
15kW
40GB on-chip SRAM
Mem BW 20PB/s (on-chip)
On-chip Fabric BW 220Pb/s

[AMD] Milan/Rome CPUs
>100B Transistors
8 CPU die 1 I/O die
64 cores/128 Threads
280W

[Intel] Ponte Vecchio GPU
>100B Transistors
47 Active Tiles
120GB on-package HBM
Multi-package interconnect
Rack-scale systems

Dojo Training Tile

V1 Dojo Training Matrix

Dojo Unique Innovation: Flattened Hierarchies

HotChips'22
Expansion of memory-semantic fabrics

**DGX A100 256 SuperPOD**
- IB HDR spine switches
- ... IB HDR leaf switches ...
- ... 32 nodes (256 GPUs) ...

**DGX H100 256 SuperPOD**
- NVS
- NVS
- NVS
- ...
- NVS
- New NVLink Switch

Fully NVLink-connected
Massive bisection bandwidth

<table>
<thead>
<tr>
<th></th>
<th>A100 SuperPod</th>
<th>H100 SuperPod</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DGX / 8 GPUs</td>
<td>2.5</td>
<td>2,400</td>
<td>150</td>
</tr>
<tr>
<td>32 DGXs / 256 GPUs</td>
<td>80</td>
<td>6,400</td>
<td>100</td>
</tr>
</tbody>
</table>
Network topology spectrum

- **Mesh**
- **CMesh**
- **Clos**
- **Crossbar**

**Increasing diameter**

- Easy to design
- Hard to program

**Increasing radix**

- Hard to design
- Easy to program

Radix – Number of inputs and outputs of each switching node
Diameter – largest minimal hop count over all node pairs

**In power constrained** systems – Need to look at networks in a **cross-cut** approach
Connect physical implementation (channels, routers, power) with **network topology, routing and flow-control**
Lecture Roadmap

- Networking Basics
- Building Blocks
- Evaluation
Lecture Roadmap

• Networking Basics
  • Topologies
  • Routing
  • Flow-Control

• Building Blocks

• Evaluation
Message definitions

- Basic trade-off
  - Minimize overheads (large size)
  - Efficient use of resources (small size)

[Dally&Towles: Principles and Practices of Interconnection Networks - Dally04]
Latency Components

- **Zero-load latency**
  - **Average latency w/o contention**

  ![Flowchart]

  \[
  T_0 = H_{\text{min}} t_r + \frac{D_{\text{min}}}{v} + \frac{L}{b}
  \]

  - **Router delays**
  - **Channel delays**
  - **Serialization delay**

  - \(H_{\text{min}}\) – average minimum number of hops
  - \(t_r\) – Router delay
  - \(D_{\text{min}}\) – average minimum distance
  - \(v\) – signal velocity
  - \(L\) – packet length in bits
  - \(b\) – router-to-router channel bandwidth

  \[T_0 = 2t_r + (t_{xy} + t_{yz}) + \frac{L}{b}\]
Ideal network throughput (capacity)

- Maximum traffic that can be sustained by all cores
- Mesh throughput
  - 50% of data crosses the bisection assuming uniform random traffic
- Bisection bandwidth \( = 2\sqrt{Nb} \)
- Data crossing the bisection \( = \frac{1}{2} N b_{core} \)
- Maximum throughput \( \Theta_{ideal} = N b_{core} = 4\sqrt{Nb} \)

\( N \) = number of cores
\( b \) = router-to-router link bandwidth
\( b_{core} \) = rate at which each core generates traffic

To maximize bandwidth, a topology should saturate the bisection bandwidth
Network performance plots

Zero-load latency includes effects of routing and flow-control

![Graph showing network performance plots with labels for Topology, Routing, and Flow-control.](Image)
Tori

- Low-radix, large diameter networks
- N-ary, K-cube (mesh)
  - N nodes per dimension
  - K dimensions
- Cubes have 2x larger bisection bandwidth

[Dally04]
TILE64

- 64 cores at 750 MHz
- Memory BW 25 GB/s
- 240 GB/s bis. Bw

[Bell08]
TILE64 Networks

[Wentzlaff07]

STN – Static network
TDN – Tile Dynamic network
UDN – User Dynamic network
MDN – Memory Dynamic network
IDN – I/O Dynamic network

32 bit channels on all networks

Wormhole, dimension-order routed

5-port routers with credit-based flow-control

STN – Scalar operand network
TDN and MDN implement the memory sub-system
UDN/IDN – Directly accessible by processor ALU (message-based, variable length)
Improving Tori - Express cubes

- Increase bisection bandwidth, reduce latency
  - Add expressways - long “express” channels

One dimension of 16-ary express cube with 4-hop express channels

Add extra channels to diversify and/or increase bisection
Butterflies

- N-ary, K-fly
  - N nodes per switch
  - K stages
- Example
  - 2-ary 4 fly

[Dally04]
Path diversity problem

• Butterflies have no path diversity
• Bad performance for some traffic patterns
  • e.g. shuffle permutation
• Wide spread in BW
• Inherently blocking
• Fixed in Clos topologies

[Dally04]
Clos networks

- Redundant paths – more uniform throughput

[Clos53]
Logical to Physical Mapping

Router group

8-ary 3-stage Clos

• Same topology – different physical mapping
Topology comparison

Table 1: Comparison of network parameters – Networks sized to support 128 bits/cycle per tile under uniform random traffic.

<table>
<thead>
<tr>
<th>Topology</th>
<th>$N_C$</th>
<th>$b_C$</th>
<th>$N_{BC}$</th>
<th>$N_{BC} \cdot b_C$</th>
<th>$N_R$</th>
<th>radix</th>
<th>$H$</th>
<th>$T_R$</th>
<th>$T_C$</th>
<th>$T_{TC}$</th>
<th>$T_S$</th>
<th>$T_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh</td>
<td>224</td>
<td>256</td>
<td>16</td>
<td>4,096</td>
<td>64</td>
<td>5x5</td>
<td>2-15</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>7-46</td>
</tr>
<tr>
<td>CMesh</td>
<td>48</td>
<td>512</td>
<td>8</td>
<td>4,096</td>
<td>16</td>
<td>8x8</td>
<td>1-7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3-25</td>
</tr>
<tr>
<td>Clos</td>
<td>128</td>
<td>128</td>
<td>64</td>
<td>8,192</td>
<td>24</td>
<td>8x8</td>
<td>3</td>
<td>2</td>
<td>2-10</td>
<td>0-1</td>
<td>4</td>
<td>14-32</td>
</tr>
<tr>
<td>Crossbar</td>
<td>*64</td>
<td>*128</td>
<td>*64</td>
<td>8,192</td>
<td>1</td>
<td>64x64</td>
<td>1</td>
<td>10</td>
<td>n/a</td>
<td>0</td>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

$N_C$ = number of channels, $b_C$ = bits/channel, $N_{BC}$ = number of bisection channels, $N_R$ = number of routers, $H$ = number of routers along data paths, $T_R$ = router latency, $T_C$ = channel latency, $T_{TC}$ = latency from tile to first router, $T_S$ = serialization latency, $T_0$ = zero load latency. *Crossbar “channels” are the shared crossbar buses.
Routing Algorithms

• **Deterministic routing algorithms**
  • Always same path between x and y
    • Poor load balancing (ignore inherent path diversity)
    • Quite common in practice
      • Easy to implement and make deadlock-free.

• **Oblivious algorithms**
  • Choose a route w/o network’s present state
    • E.g. random middle-node in Clos

• **Adaptive algorithms**
  • Use network’s state information in routing
    • Length of queues, historical channel load, etc
Deterministic Routing

2-ary 3-fly

6-ary 2-cube

Destination-tag
Butterflies

Dimension-order
Tori

[Dally04]
Oblivious Routing

• Valiant’s algorithm (Randomized Routing) [Dally04]

Folded Clos (Fat Tree)

Randomly select nearest common ancestor switch

8-ary 3-fly Clos

Randomly select middle switch

6-ary 2-cube

Randomly select middle node
Dimension-order to/from node
Flow Control

• Bufferless flow-control (Circuit Switching)

• Buffered flow-control (Packet Switching)
  • Packet-based (store&forward, cut-through)
  • Flit-based (wormhole, virtual channels)

• Buffer Management
  • Credit-based, on-off, flit-reservation
Circuit switching

Pros

• Simple to implement (simple routers, small buffers)

Cons

• High latency (R+A) and low throughput
Example - Pipelined Circuit Switching

64 core 2D mesh, 125 mW/router

Network efficiency 3 pJ/bit
Packet-buffered Flow Control

Buffer and channel allocated to the whole packet

• **Store-and-forward**
  
  Start next hop after whole packet received

  ![Store-and-forward diagram]

  5-flit packet

  \[ T_0 = H \left( t_r + \frac{L}{b} \right) \]

• **Cut-through**
  
  Start next hop after head flit received

  ![Cut-through diagram]

  5-flit packet

  \[ T_0 = H t_r + \frac{L}{b} \]

  Contention for channel 2

  Both ineffective in use of buffer storage

  Contention latency increased in channels

[Dally04]
Flit-buffered Flow Control

Buffer and channel allocated to flits

• Wormhole

I – idle, W – waiting, A – allocated

More efficient buffer usage than cut-through
But, may block a channel mid-packet

[Dally04]
Flit-buffered Flow Control

- Wormhole vs. Virtual-Channel

[Dally92]

[Dally04]
Virtual-channels — Bandwidth Allocation

Inputs compete for bandwidth
Flit-by-Flit

# flits in VC buffer (cap 3)

Fair Arbitration

Winner-take-all
Arbitration

Reduced latency
No throughput penalty
Virtual-channel Router

Each channel only as deep as round-trip credit latency
More buffering, more virtual channels

[Dally04]
Credit-based buffer management

\[ F \geq \frac{t_{\text{crt}} b}{L_f} \]

- \( F \) - Flit buffer depth
- \( L_f \) - Flit length
- \( b \) - channel bandwidth
- \( t_{\text{crt}} \) - credit round-trip delay

[Dally04]
Lecture Roadmap

• Networking Basics

• Building Blocks
  • Channels
  • Routers

• Evaluation
Building block costs

- Simple routers and channels roughly balanced
- Narrower networks scale better

Router Area Breakdowns

90nm technology
Channels: Electrical technology

- Design constraints
  - 22 nm technology
  - 500 nm pitch
  - 5 GHz clock

- Design parameters
  - Wire width
  - Repeater size
  - Repeater spacing
Channels: Equalized interconnects

- FFE shapes transmitted pulse
- DFE cancels first trailing ISI tap
- Lower energy cost due to output voltage swing attenuation

[Mensik07, Kim08, Kim09]
Repeated interconnects vs Equalized interconnects

Data-dependent energy (DDE) is 4-10x lower for equalized interconnects, while fixed energy (FE) is comparable.
Routers

Input VC state

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Global state</td>
</tr>
<tr>
<td>R</td>
<td>Route</td>
</tr>
<tr>
<td>O</td>
<td>Output VC</td>
</tr>
<tr>
<td>P</td>
<td>Pointers</td>
</tr>
<tr>
<td>C</td>
<td>Credit count</td>
</tr>
</tbody>
</table>

Output VC state

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Global state</td>
</tr>
<tr>
<td>I</td>
<td>Input VC</td>
</tr>
<tr>
<td>C</td>
<td>Credit count</td>
</tr>
</tbody>
</table>
Router pipeline

- Pipelined routing of a packet

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Head flit</th>
<th>Body flit 1</th>
<th>Body flit 2</th>
<th>Tail flit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RC</td>
<td>VA</td>
<td>SA</td>
<td>ST</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RC – route computation
VA – virtual channel allocation
SA – switch allocation
ST – switch traversal

Pipeline stalls (virtual-channel allocation stall – output VC)

VC stall need not slow transmission over the input channel as long as there is sufficient buffer space (in this case, six flits) to hold the arriving head and body flits until they are able to begin switch traversal.
Speculation and Lookahead

Speculative allocation

Lookahead routing
(pass routing for next hop in head flit)
Crossbar switches

No Speedup – 68% capacity

2x Input Speedup – 90% capacity

2x Output Speedup – 87% capacity

2x Input & Output Speedup – 137% capacity

\[ \Theta = s_o \left(1 - \left(\frac{k-1}{k}\right)^{s_k/s_o}\right) \]
**Router design space exploration - Setup**

- \( w \): Flit size (bits)
- \( p \): Ports = 5

6-bit Destination Address for 64-core system

[Shamim09]
Matrix Crossbar

input 1
input 2
...  
input k

input 1 output 1
input 2 output 2
... output k
Mux Crossbar

Multiplexer Crossbar

in1[w-1:0] → out1[w-1:0]
in2[w-1:0] → out2[w-1:0]
in3[w-1:0] → out3[w-1:0]
in4[w-1:0] → out4[w-1:0]
in5[w-1:0] → out5[w-1:0]
Example System

- 64 tiles.
- 1GHz frequency
- 1 Message = 512-bits
- 4 Messages per input port (2048-bits)
- Router Aspect Ratio 1
  - $p = 5, 8, 12$
  - $w = 32, 64, 128$ (bits)
- Matrix xbar
- Mux xbar

Design space
18 Routers
# 5x5 Router Floorplan (128bit)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td></td>
</tr>
</tbody>
</table>

**Chip Area**

- 16word 128bits SRAM
- 16word 128bits SRAM
- 16word 128bits SRAM

**Power Rings**

- VDD Power Ring
- VSS Power Ring

**Router Chip Area**
8x8 Routers Floorplan (128bit)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td></td>
</tr>
</tbody>
</table>

The diagram shows a 2x2 grid of 16-bit SRAMs, with each SRAM labeled as "16word 128bits SRAM." The grid is connected by input (in1, in2, in3, in4) and output (out1, out2, out3, out4) ports, each labeled with their respective numbers.
12x12 Routers Floorplan (128bit)
Area vs Port Width and Radix

- Mux crossbar always better
- 5-12 port routers scale well (sub $p^2$, $b^2$)
Power vs Port Width and Radix

- Mux crossbar always better
- 5-12 port routers scale well (sub $p^2$, $b^2$)
Router Power Breakdown

Xbar and Buffer power roughly even

Improve Xbar with Ckt/channel design (equalized, low-swing)

Use less buffers (circuit switching, token flow control) [Anders08, Kumar08]
Router Area per core vs. # Ports

Area Fraction VS Ports

Area Fraction (%)

- 32bit matrix
- 64bit matrix
- 128bit matrix
- 32bit mux
- 64bit mux
- 128bit mux

Ports
### Effects of Concentration

- **Mesh to Cmesh**
  - **5p routers to 8p routers**

<table>
<thead>
<tr>
<th>Matrix Design</th>
<th>Area (mm²)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 x 5p32b-mat</td>
<td>1.1664</td>
<td>332.304</td>
</tr>
<tr>
<td>1 x 8p64b-mat</td>
<td>0.4356</td>
<td>246.3924</td>
</tr>
<tr>
<td>4 x 5p64b-mat</td>
<td>1.2996</td>
<td>484.4544</td>
</tr>
<tr>
<td>1 x 8p128b-mat</td>
<td>0.8836</td>
<td>568.2672</td>
</tr>
<tr>
<td>2 x 8p32b-mat</td>
<td>0.5832</td>
<td>264.6312</td>
</tr>
<tr>
<td>1 x 12p64b-mat</td>
<td>0.6889</td>
<td>546.8928</td>
</tr>
<tr>
<td>2 x 8p64b-mat</td>
<td>0.8712</td>
<td>492.7848</td>
</tr>
<tr>
<td>1 x 12p128b-mat</td>
<td>1.7424</td>
<td>1584.54</td>
</tr>
<tr>
<td>8 x 5p32b-mat</td>
<td>2.3328</td>
<td>664.608</td>
</tr>
<tr>
<td>1 x 12p128b-mat</td>
<td>1.7424</td>
<td>1584.54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mux Design</th>
<th>Area (mm²)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 x 5p32b-mux</td>
<td>1.1664</td>
<td>268.3056</td>
</tr>
<tr>
<td>1 x 8p64b-mux</td>
<td>0.3721</td>
<td>203.268</td>
</tr>
<tr>
<td>4 x 5p64b-mux</td>
<td>1.2544</td>
<td>410.5872</td>
</tr>
<tr>
<td>1 x 8p128b-mux</td>
<td>0.7225</td>
<td>391.0116</td>
</tr>
<tr>
<td>2 x 8p32b-mux</td>
<td>0.5832</td>
<td>215.8464</td>
</tr>
<tr>
<td>1 x 12p64b-mux</td>
<td>0.5625</td>
<td>389.5896</td>
</tr>
<tr>
<td>2 x 8p64b-mux</td>
<td>0.7442</td>
<td>406.536</td>
</tr>
<tr>
<td>1 x 12p128b-mux</td>
<td>1.2769</td>
<td>926.2188</td>
</tr>
<tr>
<td>8 x 5p32b-mux</td>
<td>2.3328</td>
<td>536.6112</td>
</tr>
<tr>
<td>1 x 12p128b-mux</td>
<td>1.2769</td>
<td>926.2188</td>
</tr>
</tbody>
</table>

- Works well for small flits and number of ports

[Balfour06]
Orion 2.0 vs P & R design

Ratio (Power of Synthesized designs / Dynamic (no leakage) Power of Analytical Models)

[Kahng09]  [Shamim09]

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Xbar</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.8</td>
<td>1.4</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>1.2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6</td>
<td>0.9</td>
</tr>
</tbody>
</table>

5 ports: 32 bits, 64 bits, 128 bits
8 ports: 32 bits, 64 bits, 128 bits
12 ports: 32 bits, 64 bits, 128 bits

[Kahng09] [Shamim09]
Lecture Roadmap

• Networking Basics
• Building Blocks
• Evaluation
Clos with electrical interconnects

8-ary 3-stage Clos

- 10-15 mm channels
- Equalized
- Pipelined Repeaters

- Two 8 x 8 Routers
- Eight 8 x 8 Routers
Simulation setup

- Cycle-accurate microarchitectural simulator
- Traffic patterns based on partition application model
  - Global traffic – UR, P2D, P8D
  - Local traffic – P8C
- 64-tile system, 512-bit messages
- Events captured during simulations to calculate power
Partition application model

• Tiles divided into logical partitions and communication is within partition

• Logical partitions mapped to physical tiles
  • Co-located tiles → Local traffic
  • Distributed tiles → Global traffic

[Joshi’09]
Latency vs BW

- flatFlyX2 vs mesh/cmeshX2
  - Saturation BW \( \rightarrow \) comparable (UR, P8D, P2D)
  - Latency \( \rightarrow \) flatFlyX2 has lower latency

- clos vs mesh/cmeshX2/flatFlyX2
  - Saturation BW \( \rightarrow \) uniform for all traffic, comparable to UR of mesh
  - Latency \( \rightarrow \) uniform for all traffic, comparable to UR of mesh

Ideal Throughput \( \theta_T = 8 \) kb/cyc for UR

\[ T = 8 \text{ kb/cyc} \]

[Joshi09b]
Mesh vs CMeshX2

- Repeater-inserted interconnects
  - CMeshX2 lower power than mesh at comparable throughput

- Equalized interconnects
  - CMeshX2 has further 1.5x reduction in power
  - Channel gains masked by router power
Power vs BW plots – repeater inserted pipelined vs equalized

1.5-2x lower power with equalized channels at comparable throughput
- Channel DDE reduces by 4-10x using equalized links
- Channel fixed power and router power need to be tackled
Saturation throughput improves using VCs
Small change in power at comparable throughput
Power vs BW – no VC vs 4 VCs, repeater inserted pipelined

25-50% lower power using VCs at comparable throughput
Power vs BW—no VC case, repeater inserted pipelined vs 4 VCs, equalized

- 2-3x lower power obtained using equalized interconnects and VCs at comparable throughput
Power split

- VCs an indirect way to increase impact of channel power
  - Narrower networks, lower power for same throughput, keep utilization high
Cross-cut approach for on-chip system interconnects design needed
- Application mapping
- Topology, Routing, Flow-control
- Improving Routers and Channels equally important
  - New circuit design (low-swing, equalized)
  - System – DVFS, bus-encoding
To probe further (tools and sites)

- **DSENT - A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling**
  - [https://dspace.mit.edu/handle/1721.1/85863](https://dspace.mit.edu/handle/1721.1/85863)

- **Orion Router Design Exploration Tool**
  - [https://github.com/eigenpi/vnoc20](https://github.com/eigenpi/vnoc20)

- **Router RTLs**
  - Bob Mullins’ Netmaker
    - [http://www-dyn.cl.cam.ac.uk/~rdm34/wiki](http://www-dyn.cl.cam.ac.uk/~rdm34/wiki)

- **Network simulators**
  - Garnet ([http://www.princeton.edu/~niketa/garnet.html](http://www.princeton.edu/~niketa/garnet.html))
  - Booksim ([http://noc.s.stanford.edu/booksim.html](http://noc.s.stanford.edu/booksim.html))


Bibliography


