

Lab 4

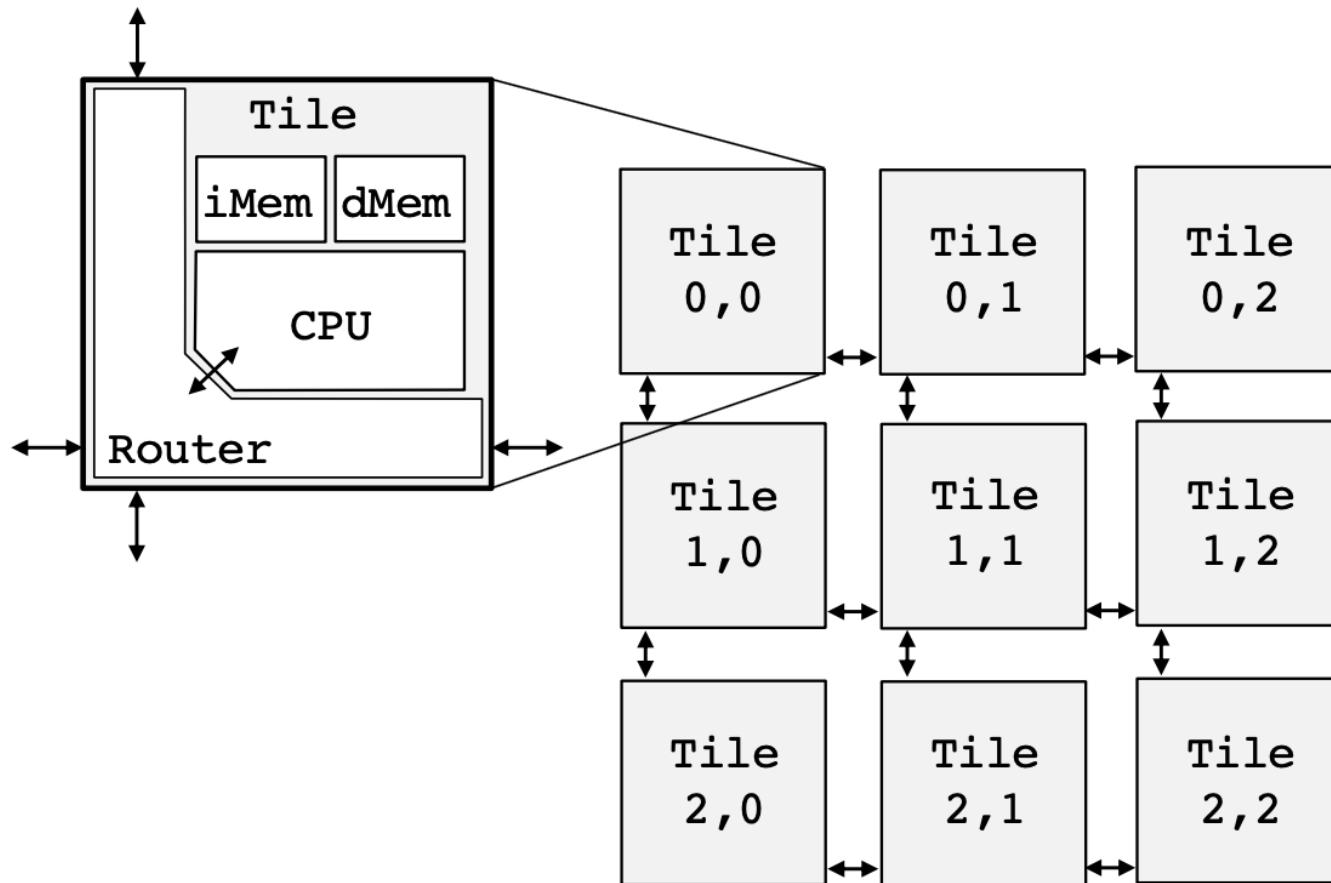
(Intro to Networks-on-Chip)

VLSI Design

Dan Holcomb
Nov 10 2020

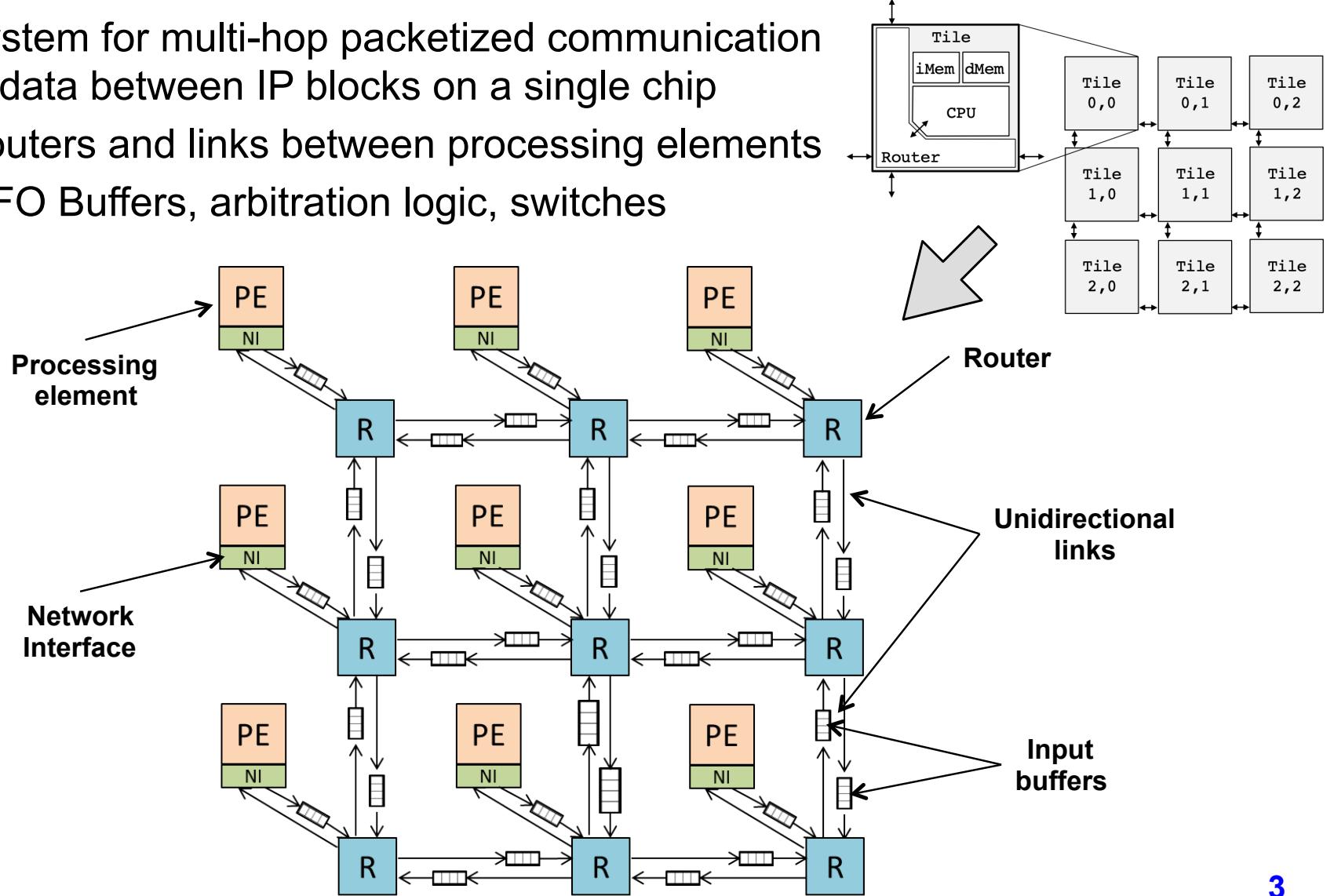
Lab 4: Network-on-Chip Tile

❑ Synthesis and Place & Route of NoC Tile



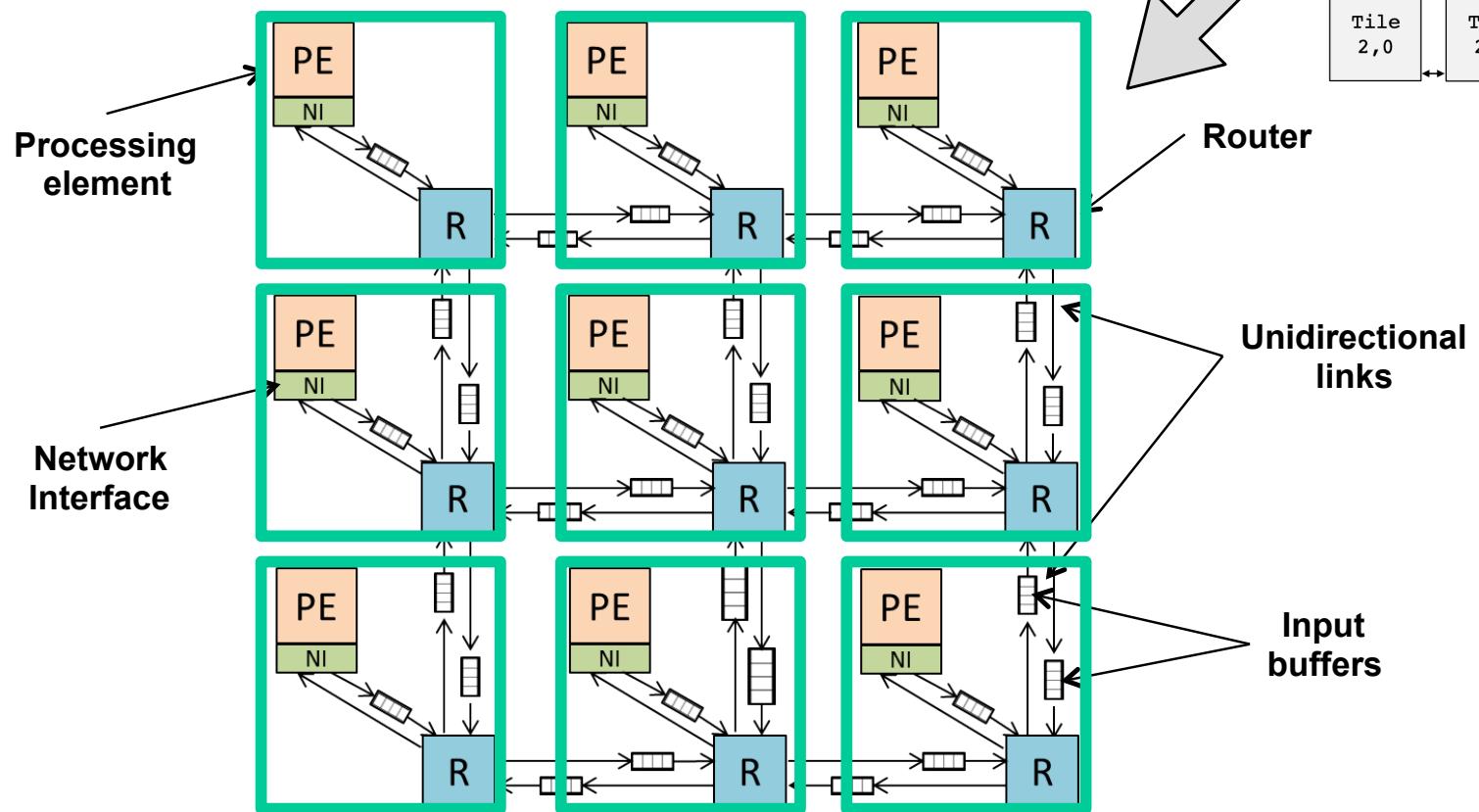
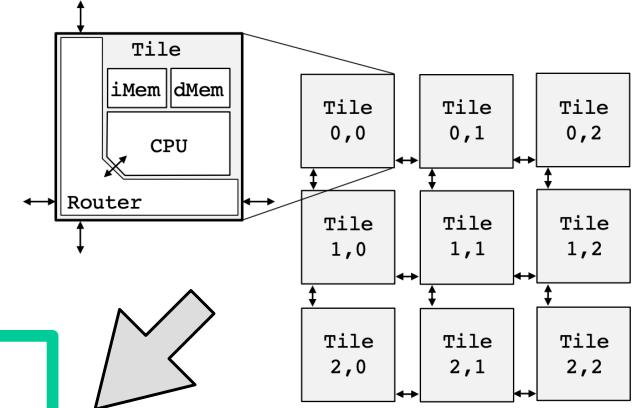
What is an NoC?

- ❑ System for multi-hop packetized communication of data between IP blocks on a single chip
- ❑ Routers and links between processing elements
- ❑ FIFO Buffers, arbitration logic, switches



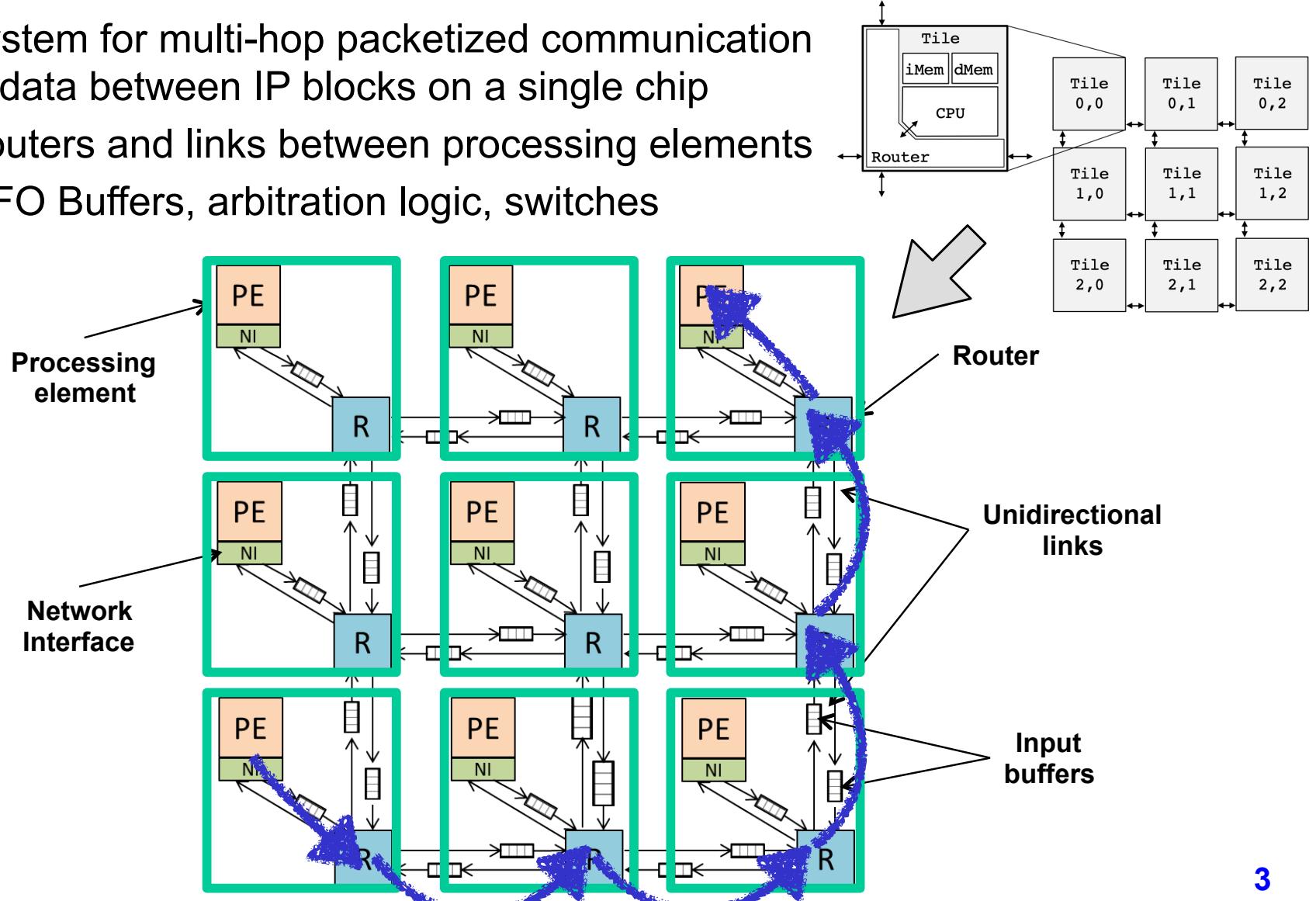
What is an NoC?

- ❑ System for multi-hop packetized communication of data between IP blocks on a single chip
- ❑ Routers and links between processing elements
- ❑ FIFO Buffers, arbitration logic, switches



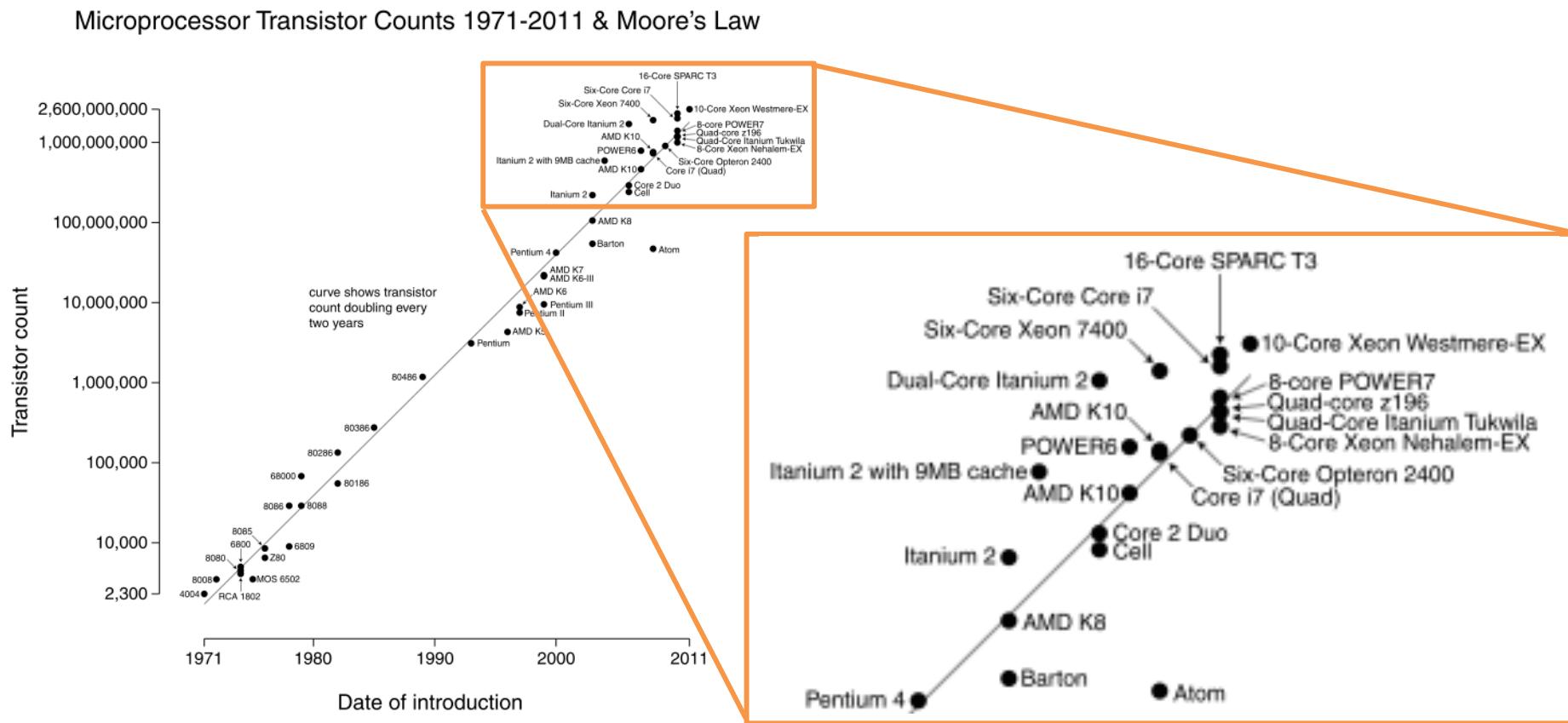
What is an NoC?

- System for multi-hop packetized communication of data between IP blocks on a single chip
- Routers and links between processing elements
- FIFO Buffers, arbitration logic, switches



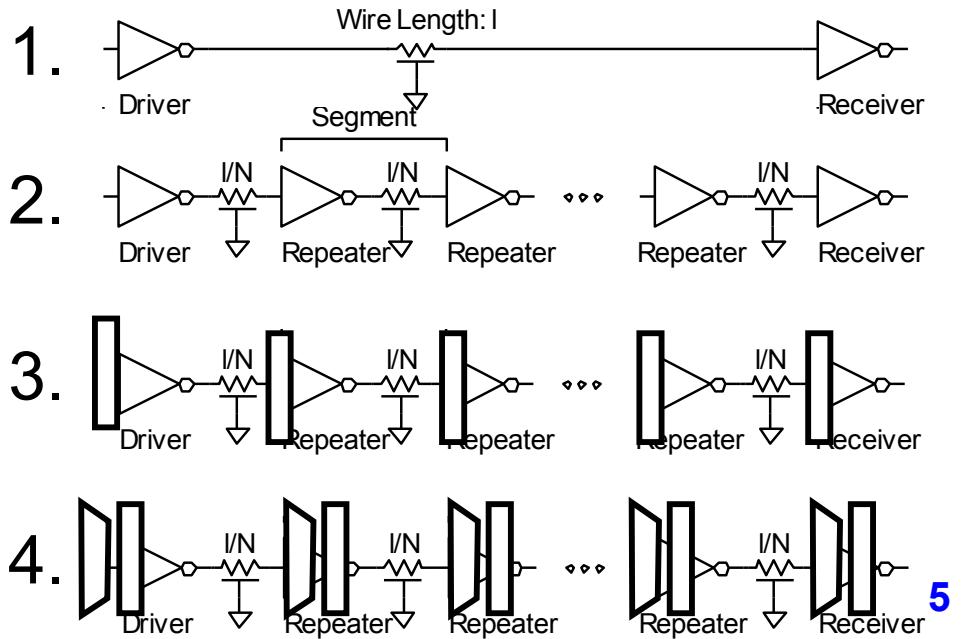
NoCs and Moore's Law

- ❑ How to use 10^9 transistors? Chip Multi-Processors
 - ❑ High-bandwidth communication needed
 - ❑ Core to core, core to/from memory controller



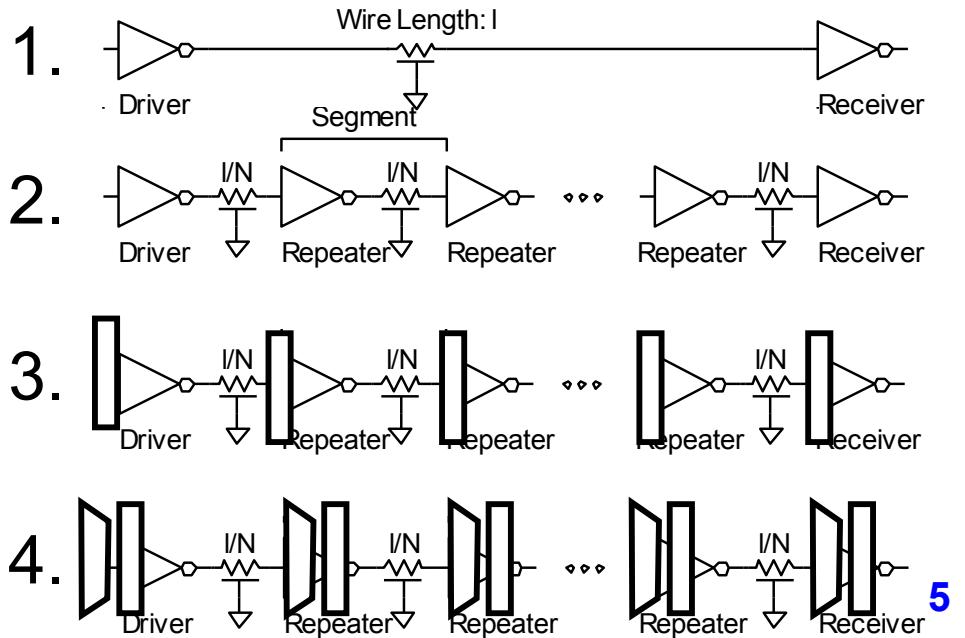
From Wires to NoCs

- What is benefit of each transformation?
1. Long wires are slow
 2. Add repeaters
 3. Add registers for pipelined links
 4. Add multiplexing so that pipelined link stages can be shared across multiple source-destination pairs



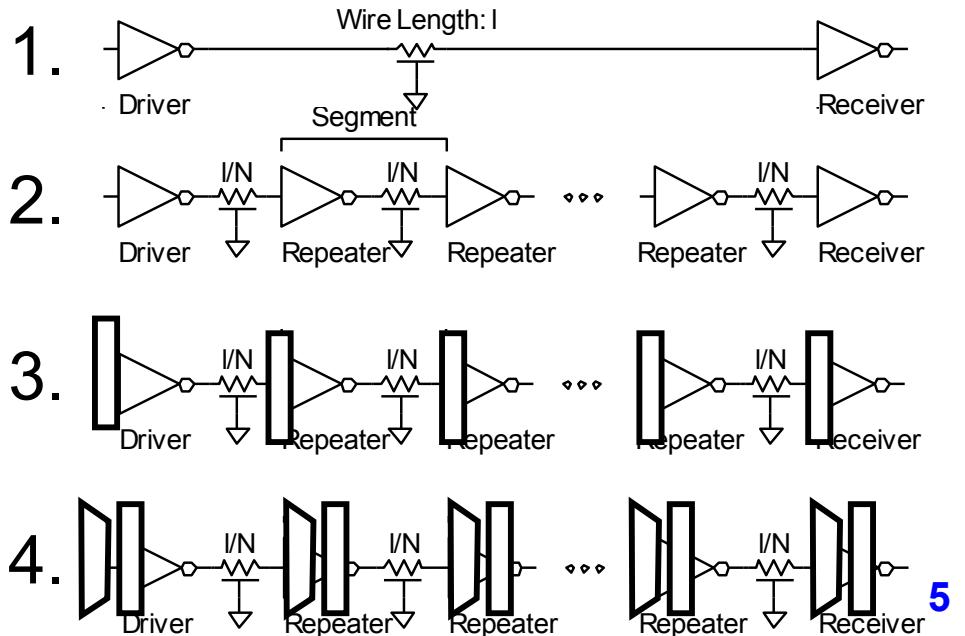
From Wires to NoCs

- What is benefit of each transformation?
1. Long wires are slow
 2. Add repeaters (for delay)
 3. Add registers for pipelined links
 4. Add multiplexing so that pipelined link stages can be shared across multiple source-destination pairs



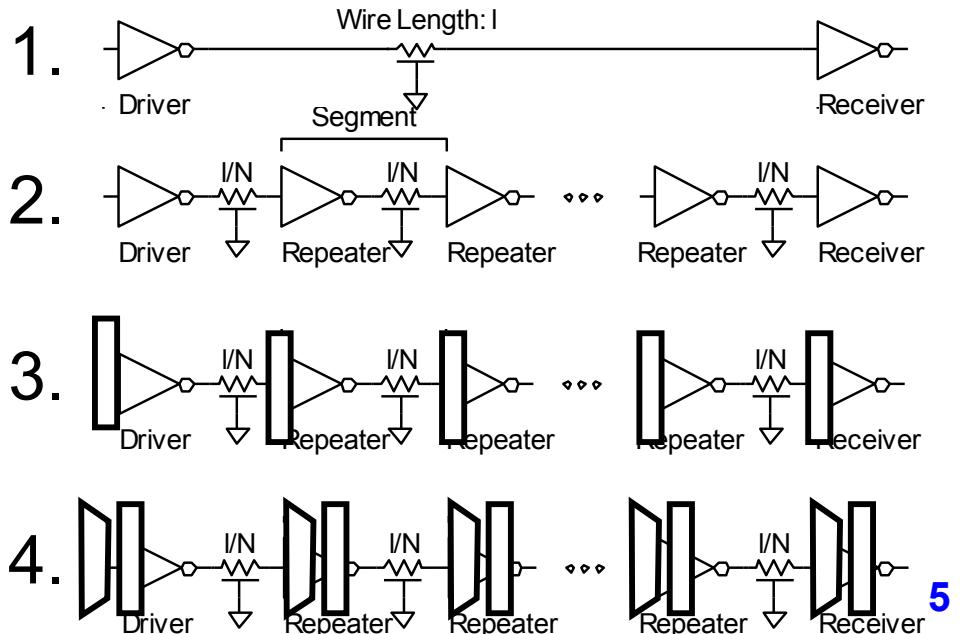
From Wires to NoCs

- What is benefit of each transformation?
1. Long wires are slow
 2. Add repeaters (for delay)
 3. Add registers for pipelined links (for throughput)
 4. Add multiplexing so that pipelined link stages can be shared across multiple source-destination pairs



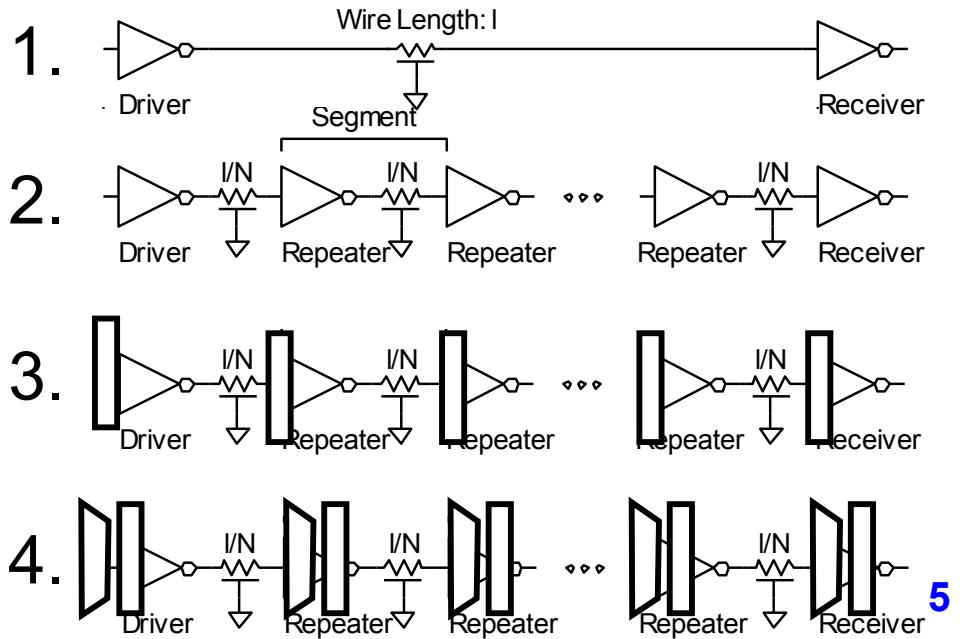
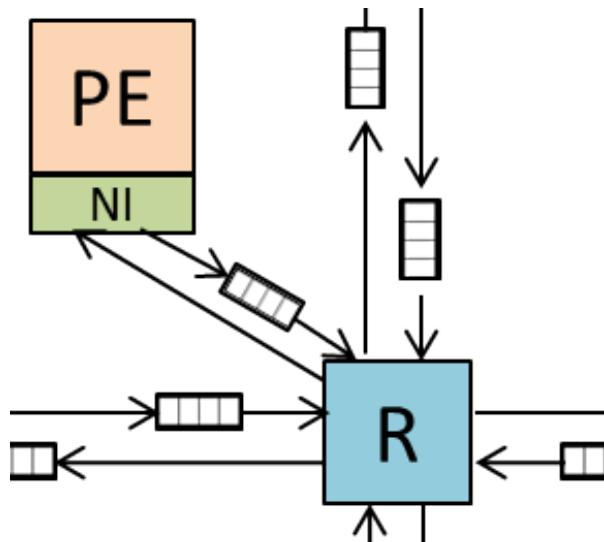
From Wires to NoCs

- What is benefit of each transformation?
1. Long wires are slow
 2. Add repeaters (for delay)
 3. Add registers for pipelined links (for throughput)
 4. Add multiplexing so that pipelined link stages can be shared across multiple source-destination pairs (for utilization)



From Wires to NoCs

- What is benefit of each transformation?
1. Long wires are slow
 2. Add repeaters (for delay)
 3. Add registers for pipelined links (for throughput)
 4. Add multiplexing so that pipelined link stages can be shared across multiple source-destination pairs (for utilization)
- NoC is system of on-chip routers for packetized communication over shared link stages

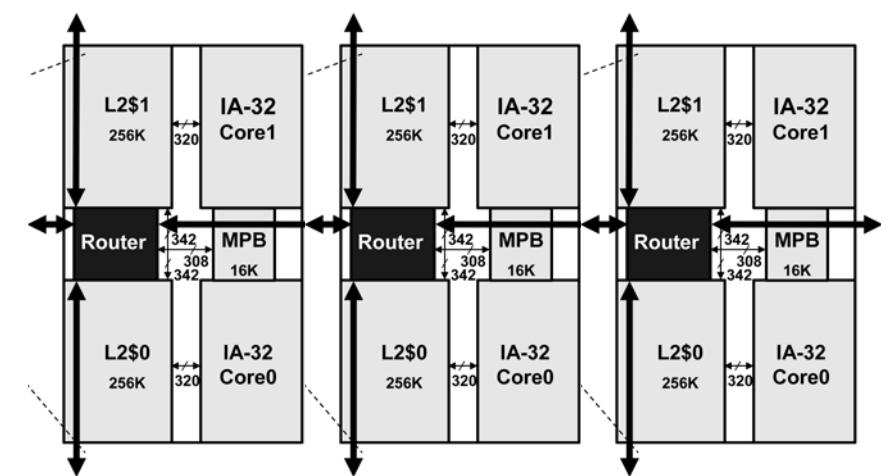
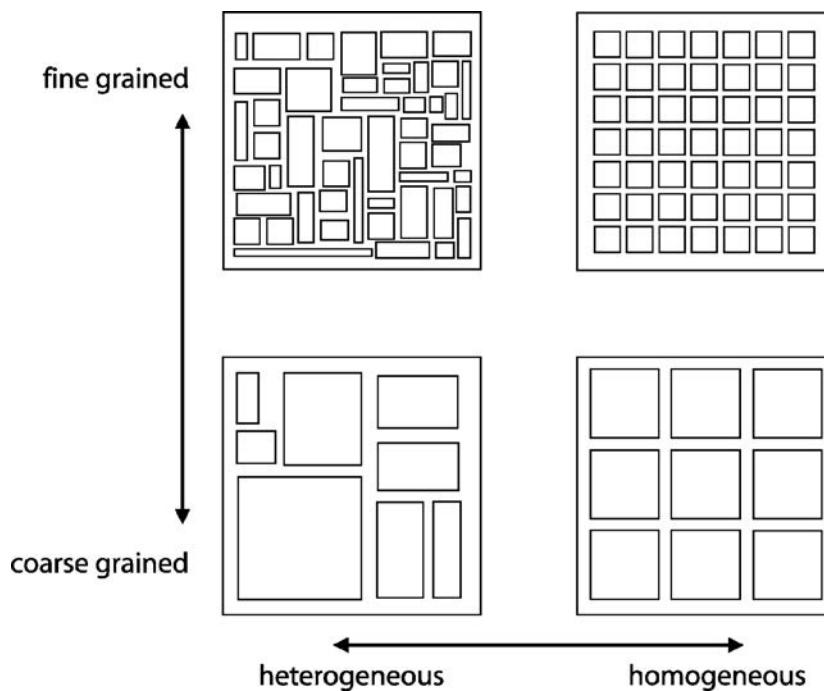


Scalability of NoC

- ❑ Distributed system: no communication back to centralized controller, make routing decisions locally
 - Easy to increase number of tiles
- ❑ Design complexity is $O(1)$: design tile once, instantiate multiple times
 - Replication justifies high design effort
- ❑ Area cost of communication is $O(\text{num_cores})$
 - vs at least $O(\text{num_cores}^2)$ for point-to-point
- ❑ Avoid problems with driving global wires

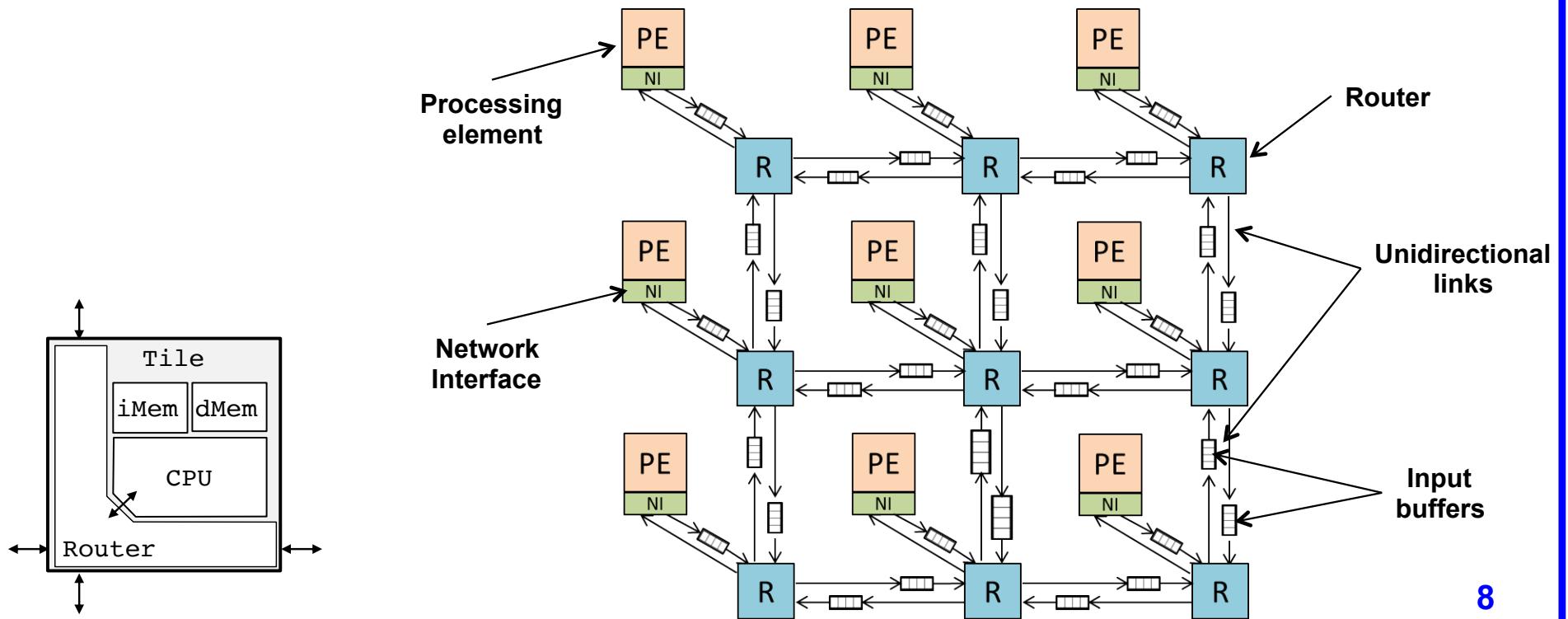
Chip Multiprocessor (CMP) NoCs

- ❑ Abutable “tiles” comprising processing element (PE) and associated router (like bitslice)
- ❑ PE is often core+cache, but sometimes I/O or memory controller (to DRAM off-chip)
- ❑ Homogeneous floorplan with around 10 -100 tiles



Routers

- ❑ Router connects to processor and neighboring tiles
 - E.g. 5 input ports, 5 output ports (for N, S, E, W, and NI)
 - FIFO buffering at inputs
 - xbar and control logic

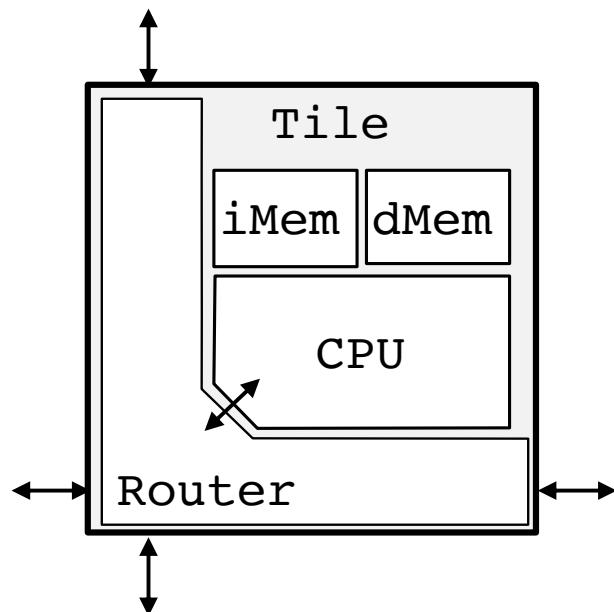


Characteristics of an NoC

- ❑ Topology — How are routers connected?
- ❑ Routing — What path from source to destination?
- ❑ Flow Control — When does data proceed to next hop?
- ❑ Plus architectural decisions
 - How wide are channels between routers?
 - How deep are buffers?
- ❑ Plus VLSI implementation choices
 - Circuit design for buffers and links between routers?
 - Clock frequency and supply voltage?

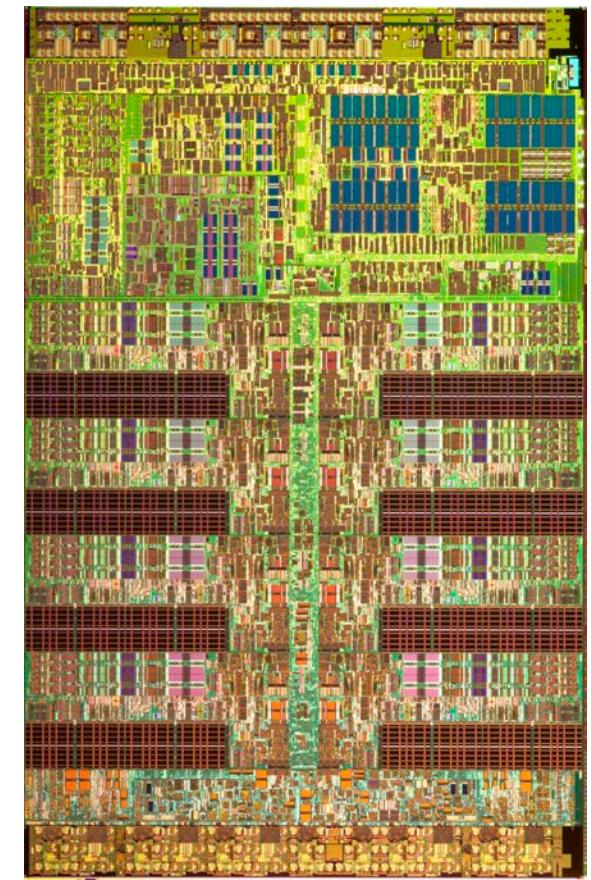
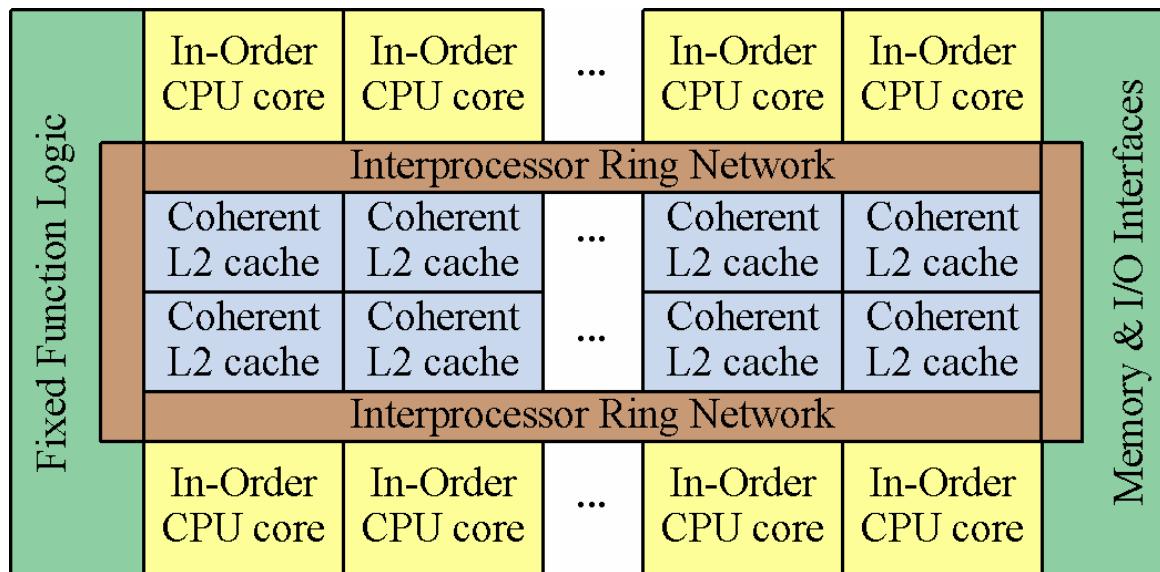
Topology

- ❑ Router in each tile communicates with tile's own processing element and with neighboring tiles
- ❑ Definition of neighbor depends on network topology
- ❑ Topology Examples...



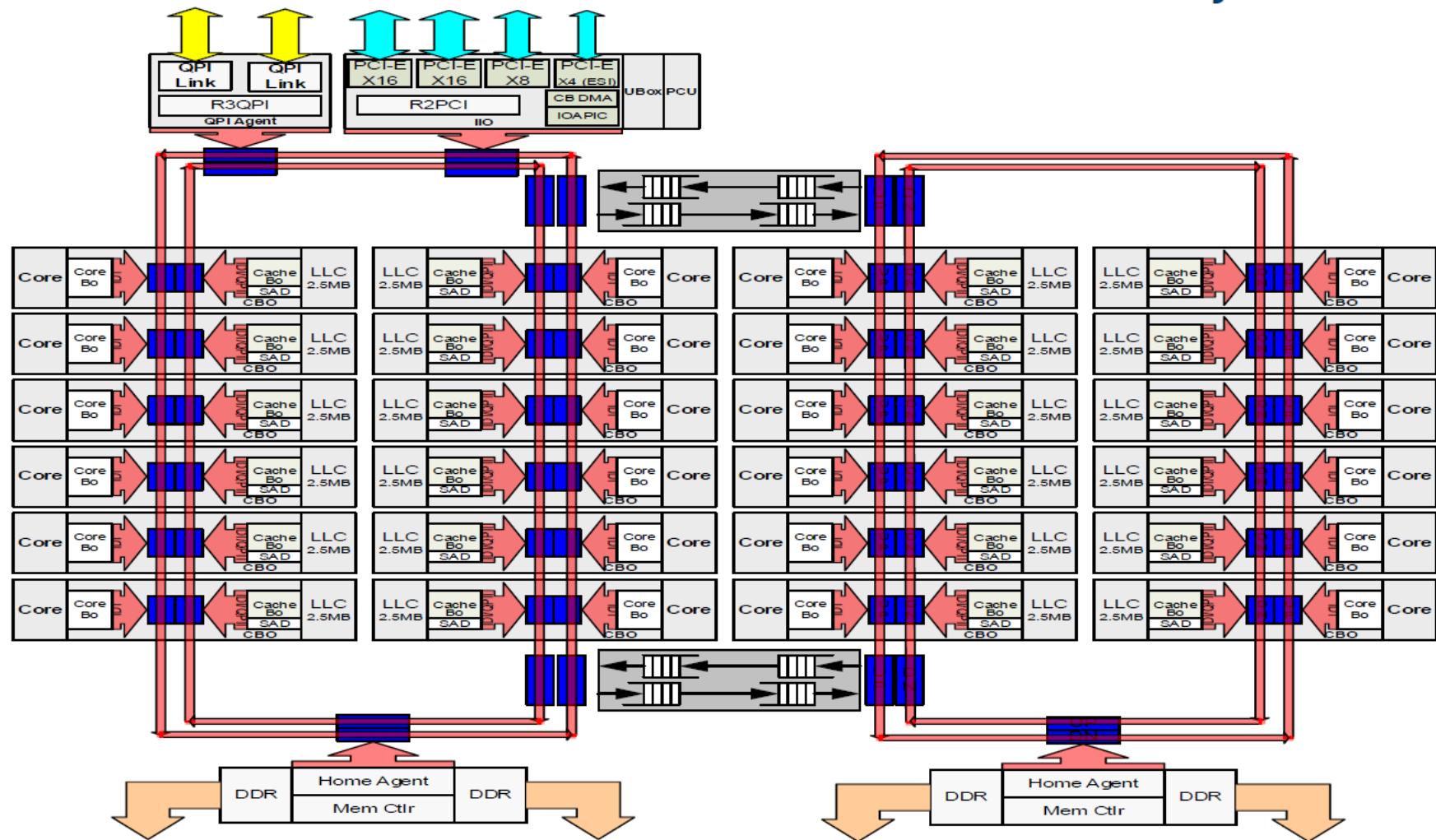
Ring Topology

- Each tile has 2 neighbors (clockwise, counterclockwise)
- Convenient for broadcast (cache coherence) because all traffic can be routed past all cores
- Basic ring scales poorly beyond tens of tiles, but hierarchical rings possible
- Used in STI Cell Broadband Engine and Intel (Nehalem, Sandy Bridge, Haswell)
~8-20 core processors from ~2009-17



Haswell Ring

Intel® Xeon® Processor E5 v4 Product Family HCC

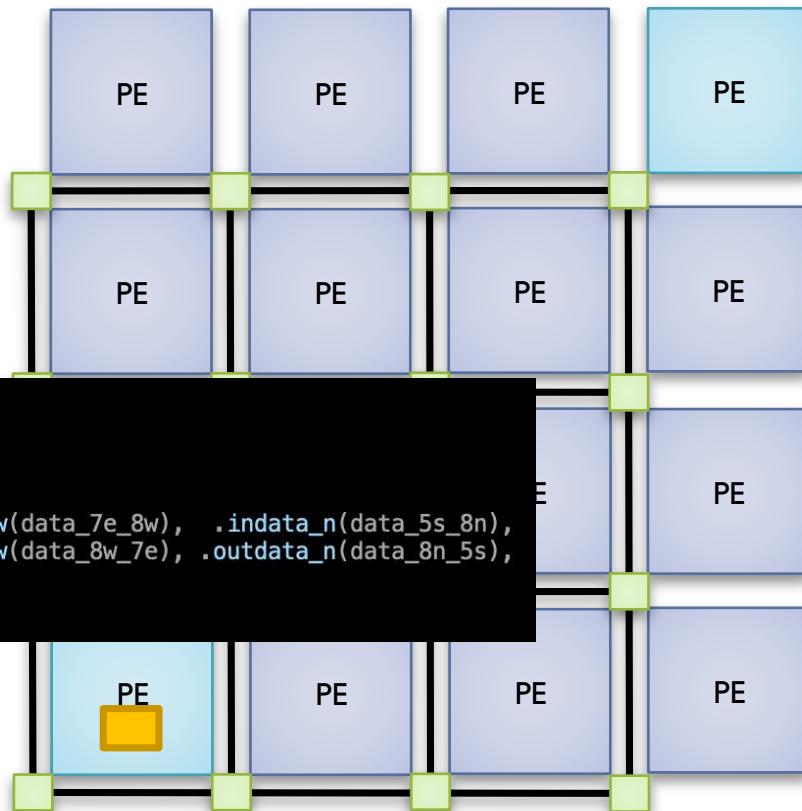


Mesh Topology

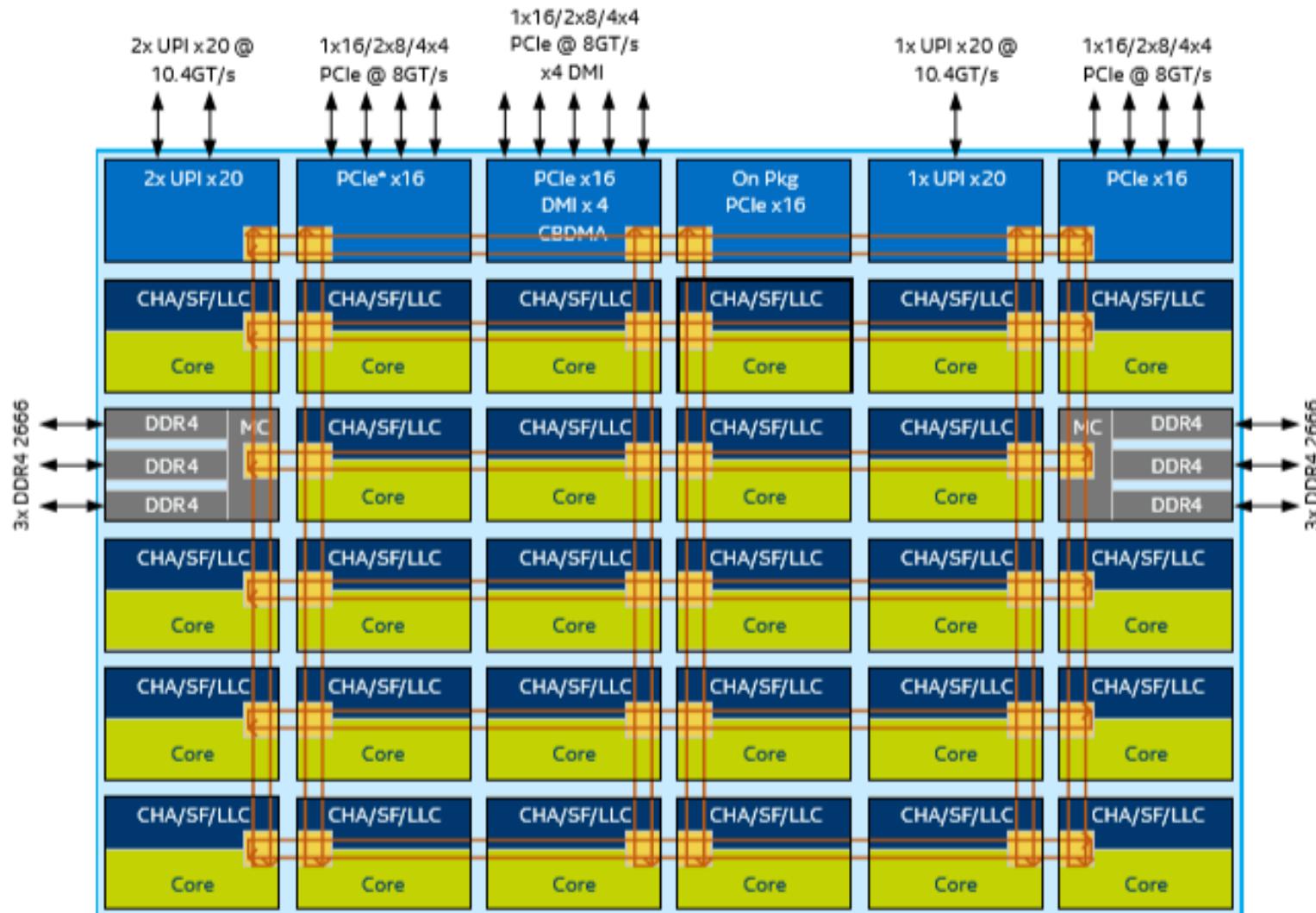
- ❑ Short links (length equals tile width)
- ❑ Hop count along paths can be high
- ❑ Some ports on edge tiles don't connect to anything

- ❑ Popular topology for NoCs with >> 10 cores
- ❑ Used in Lab 4

```
Tile tile8(  
    .clk(clk), .rst_n(rst_n), .TileID(ID8),  
    .init_mem(init_mem), .Address(address),  
    .Data_input(data8), .Instruction_input(inst8),  
    .indata_e( 13'b0 ), .indata_s( 13'b0 ), .indata_w(data_7e_8w), .indata_n(data_5s_8n),  
    .outdata_e( ), .outdata_s( ), .outdata_w(data_8w_7e), .outdata_n(data_8n_5s),  
    .cpu2router(cpu2router_8), .router2cpu(router2cpu_8)  
); // (2, 2)
```



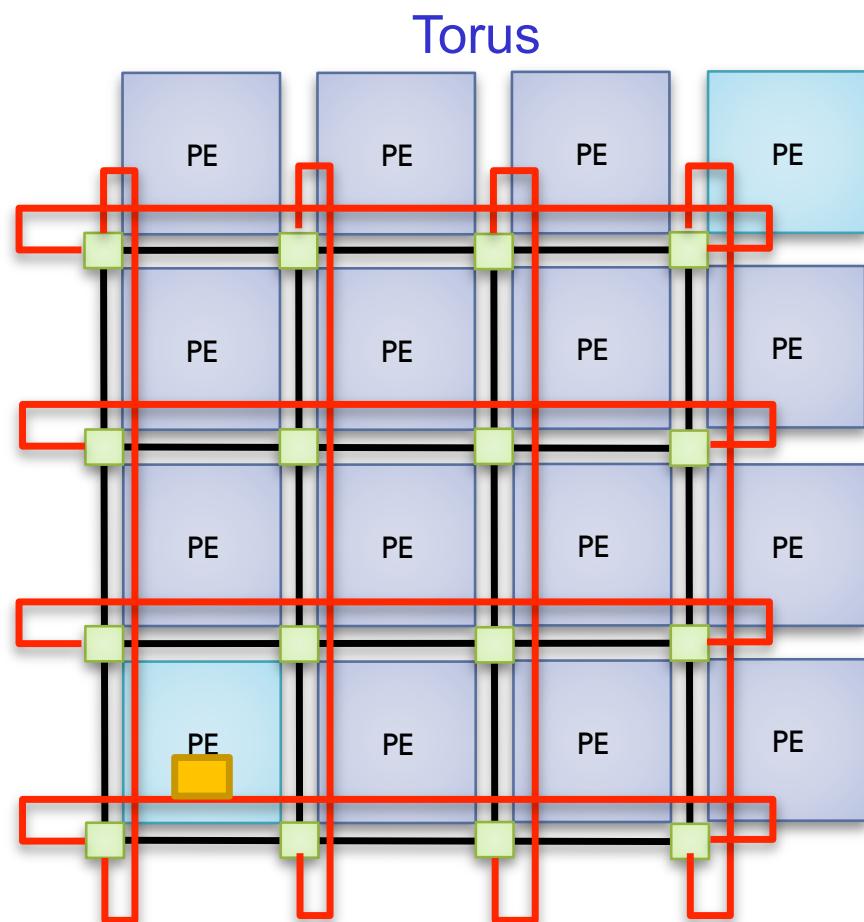
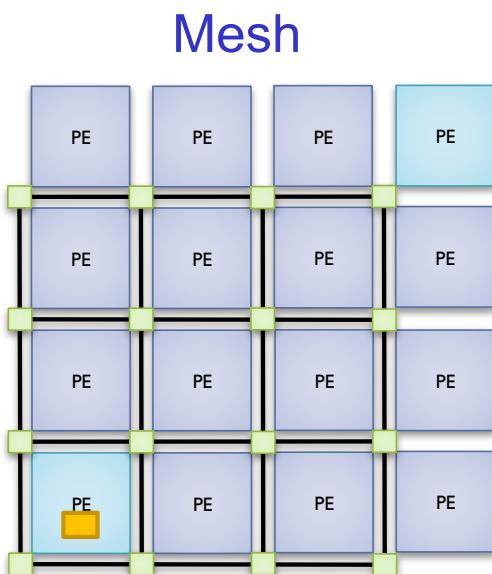
Skylake Mesh



CHA – Caching and Home Agent ; SF – Snoop Filter; LLC – Last Level Cache;
Core – Skylake-SP Core; UPI – Intel® UltraPath Interconnect

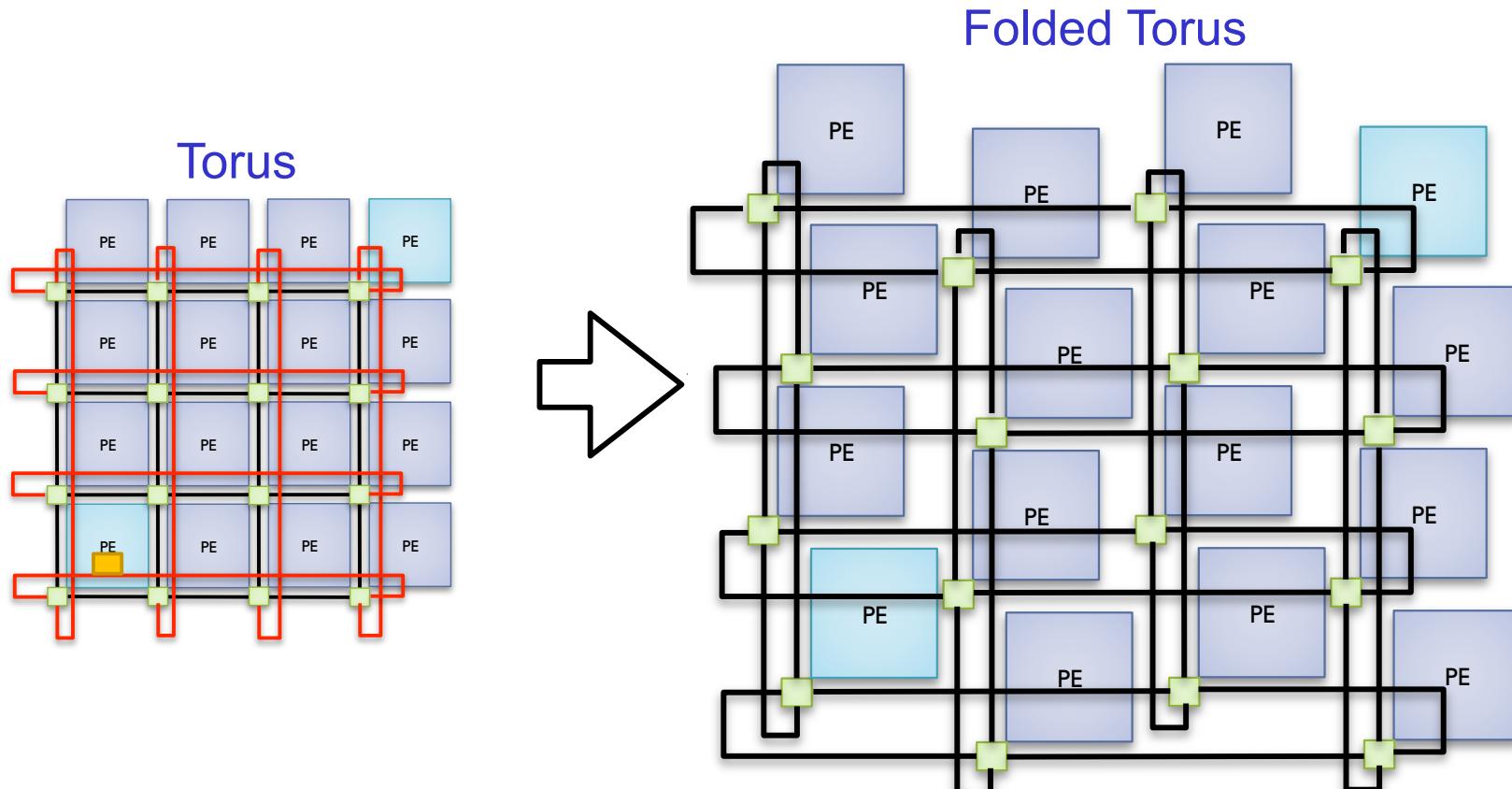
Torus

- ❑ Like a mesh but edges wrap around to other side
- ❑ Unsuitable to planar VLSI implementation due to long wrap-around links



Folded Torus

- ❑ Planar adaptation of torus
- ❑ Uniform link lengths, each 2x length of tile
- ❑ No underutilization of edge tiles (vs mesh)

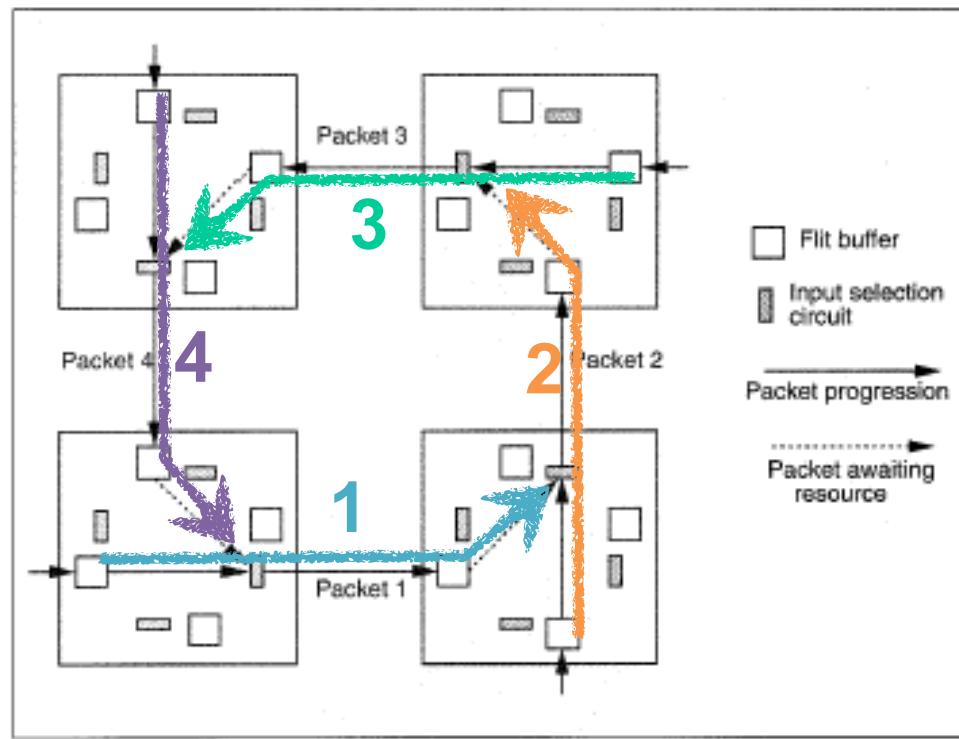


Routing

- ❑ Determines the path that packet takes through network from source to destination
 - Deterministic — always take same path for same source and destination
 - Oblivious — don't consider network state (deterministic is special case of oblivious)
 - Adaptive — make routing decisions based on network state to avoid congestion
- ❑ Perfect global information not possible, so the added complexity of adaptive routing is not usually justified
- ❑ Deterministic routing is most popular
- ❑ What deterministic routing to use?

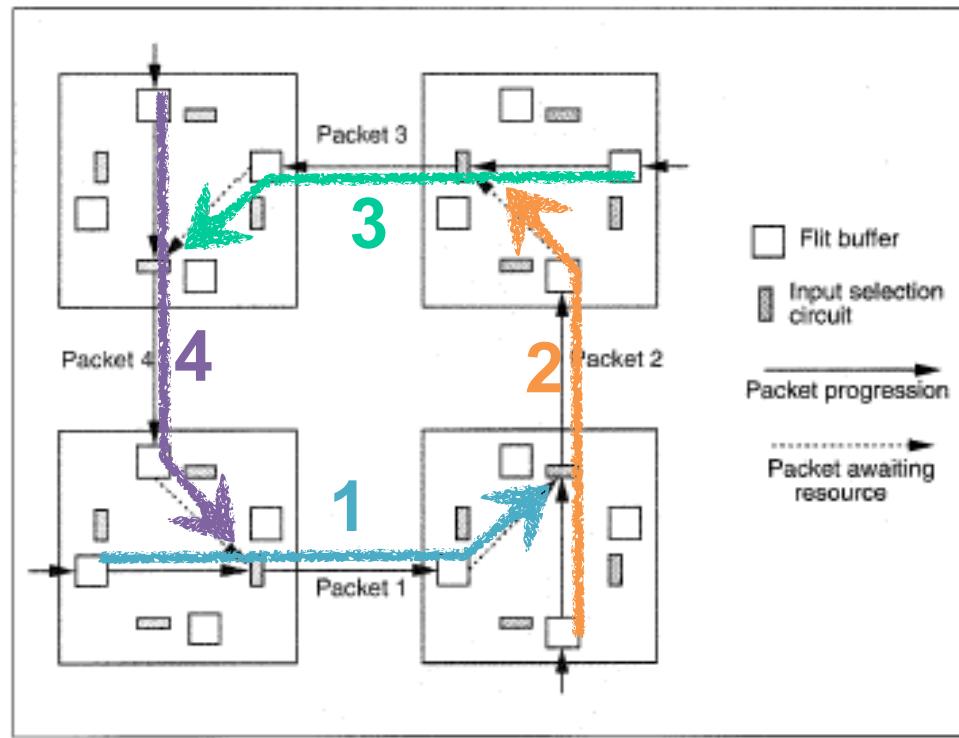
Routing Deadlocks

- ❑ Deadlock occurs if network gets stuck in a condition where no packets can make progress
- ❑ Deadlock requires circular dependencies
- ❑ Use routing rules to prevent circular dependencies



Routing Deadlocks

- ☐ Deadlock occurs if network gets stuck in a condition where no packets can make progress
- ☐ Deadlock requires circular dependencies
- ☐ Use routing rules to prevent circular dependencies



- ☐ 1 waiting for 2
- ☐ 2 waiting for 3
- ☐ 3 waiting for 4
- ☐ 4 waiting for 1

Hence, progress of packet 1 depends on progress of packet 1!

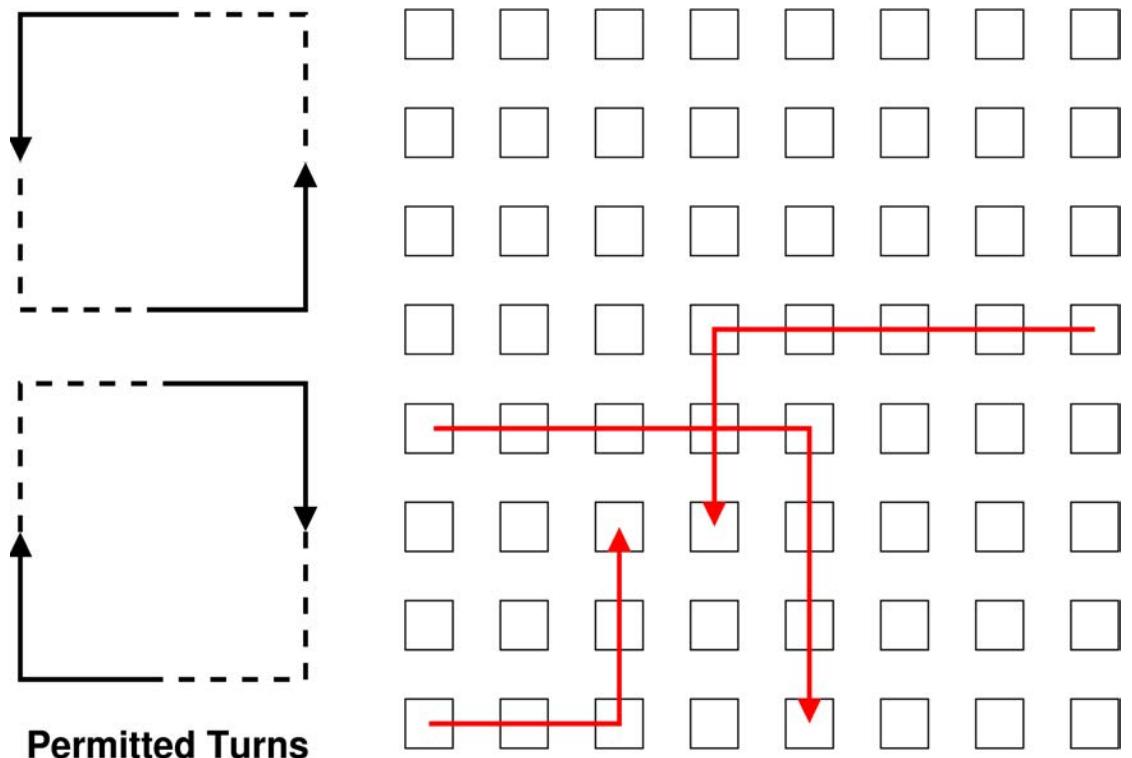
This is a deadlock

XY Dimension Ordered Routing

- ❑ Always route along X first, then Y
- ❑ Instance of restricted turns model
- ❑ Restricted turns proved deadlock free by Glass & Ni in 1992

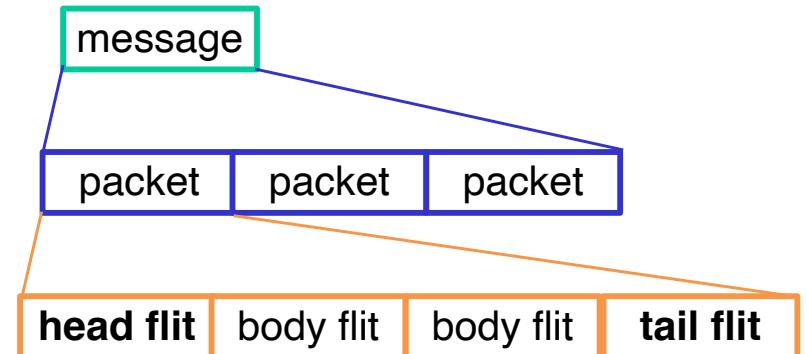
In XY D.O.R., given a router's own address and the dest. address of incoming packet:

How should router decide whether to send packet to N,S,E,W or NI port?



Flow Control

- ❑ Packets are divided into multiple “flits”
(FLow control unITS)
- ❑ Messages > Packets > Flits
- ❑ Flit crosses channel per cycle
- ❑ Flit size == channel width
- ❑ Three flit types:
 - Head: carries destination address and other header info, along with a smaller payload in remaining space
 - Body: no header, hence more payload
 - Tail: like body, but tells router that packet is complete
- ❑ Flow control: When can each flit proceed across channel to next hop?

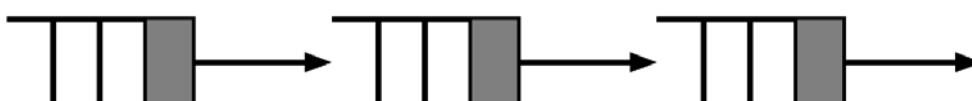


Store and Forward vs Wormhole

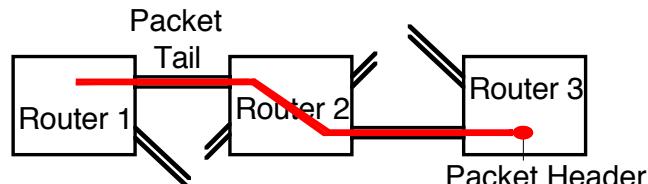
- ❑ SaF: start sending flits of packet once all flits are present at current hop and next hop has enough space for entire packet
- ❑ Wormhole: send flit whenever space is available
- ❑ Which flow control has better latency?
- ❑ Which requires larger buffers?
- ❑ Wormhole is dominant flow control today. Problems?



Store-and-Forward (SaF)



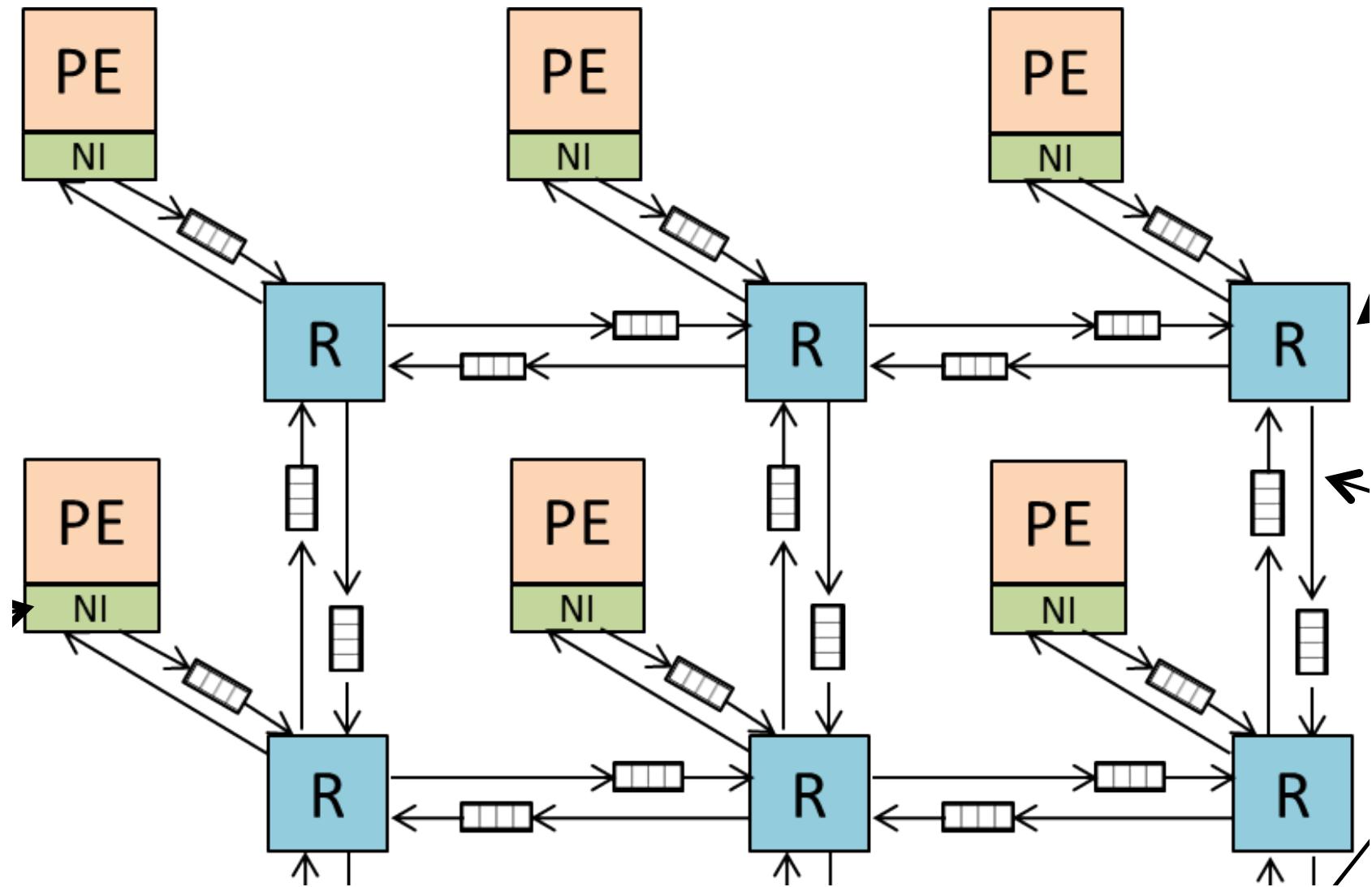
Wormhole



Head of Line Blocking

- ❑ Packet is waiting for next buffer to become free
 - Any packets behind it are also forced to wait
 - Even if their next hop buffer is available
- ❑ Why problematic for wormhole flow control in particular?
 - Flits of a packet can be stretched out across routers
 - If head flit is blocked, then traffic at many routers may be blocked by body and tail flits of packet
- ❑ Solution is virtual channels

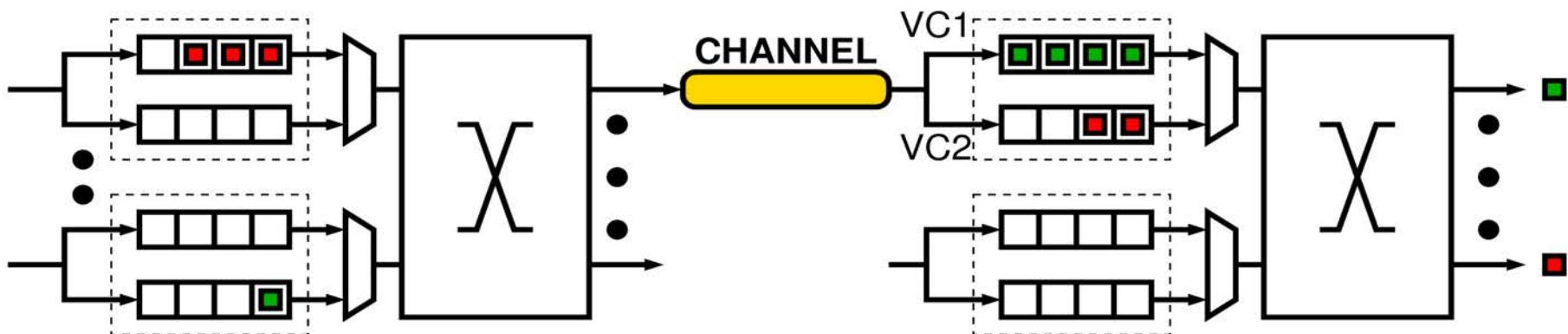
Example of Head of Line Blocking



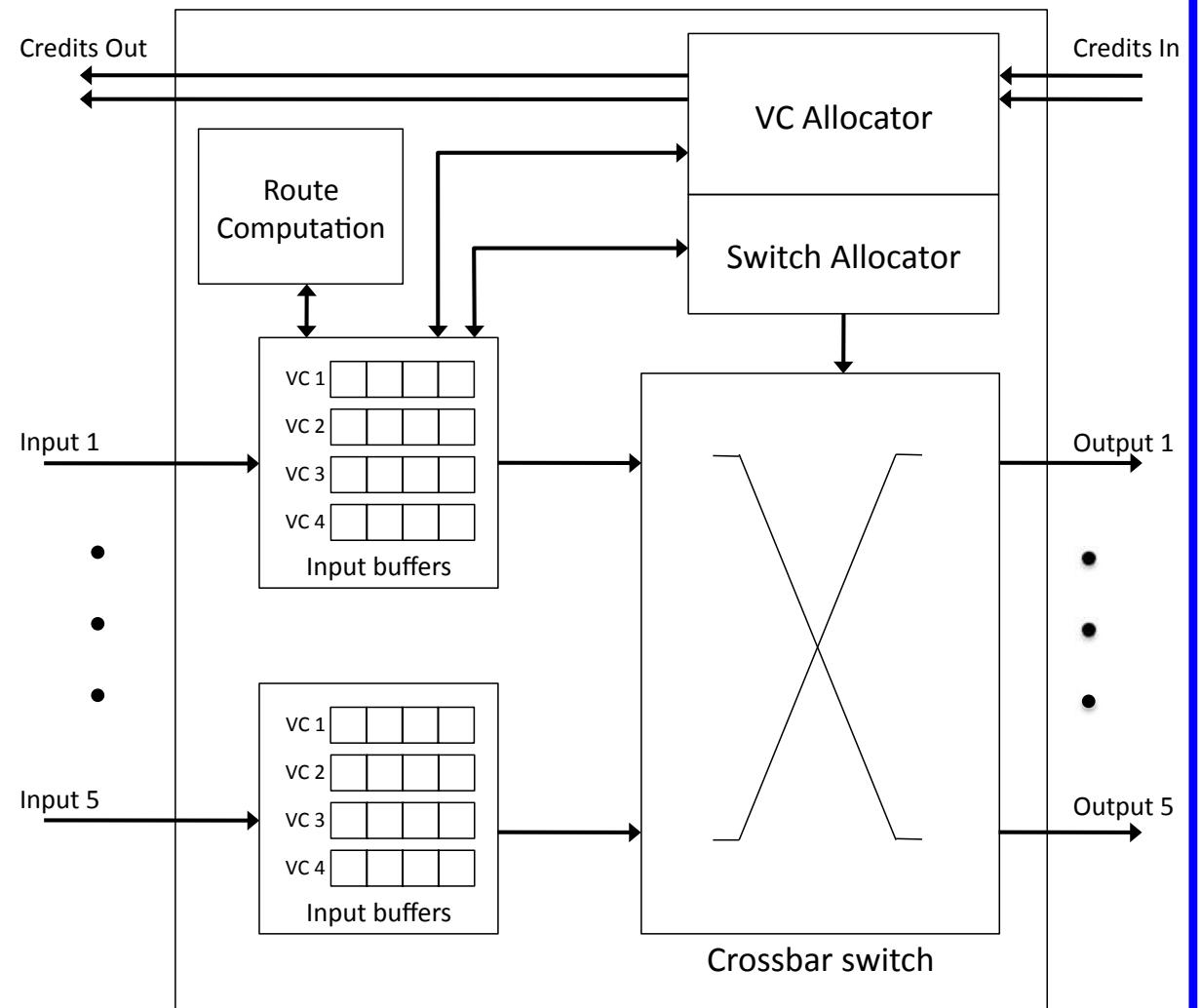
**Covered through this slide
on 11/10/2020**

Virtual Channels (VCs)

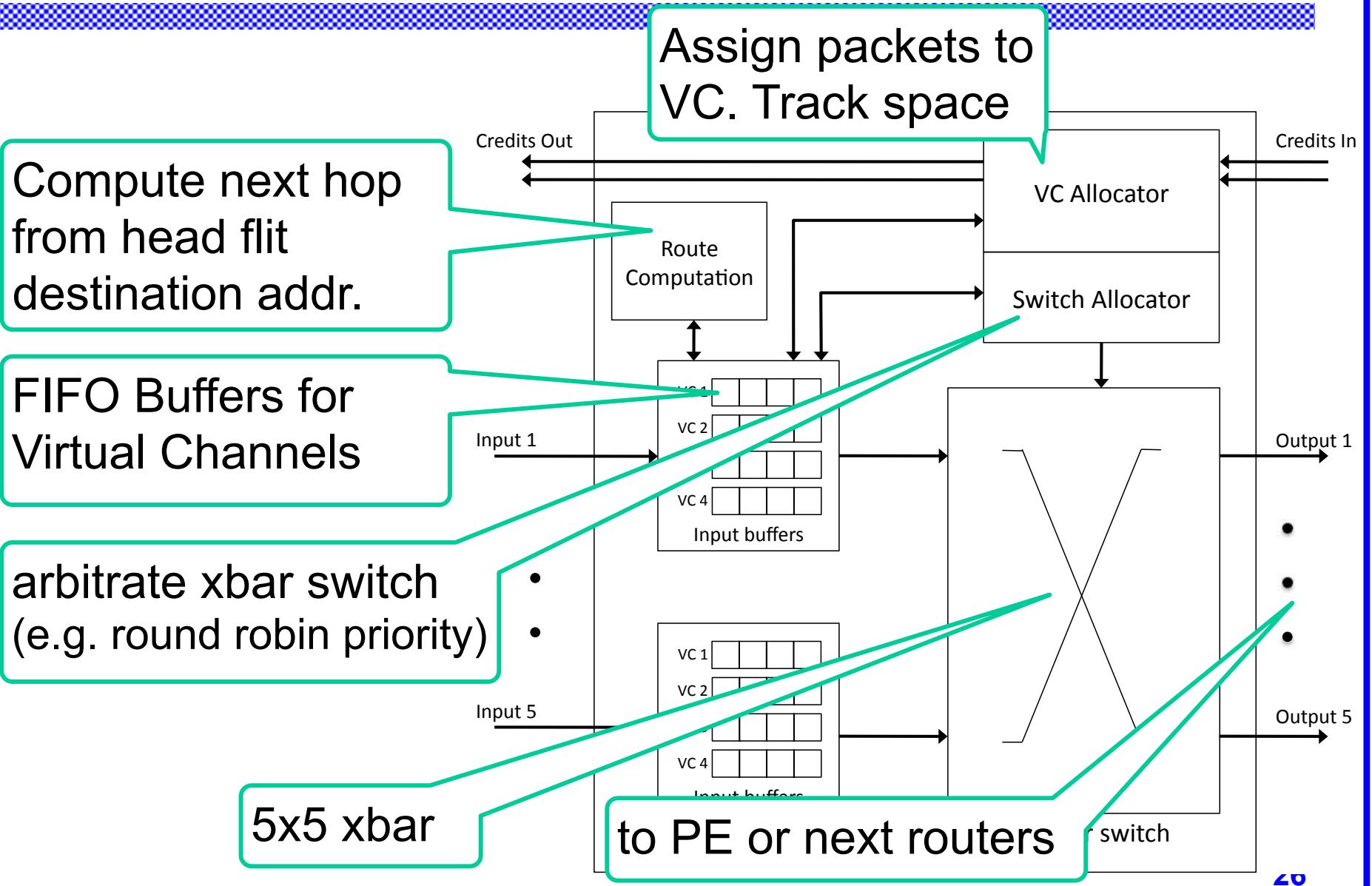
- ❑ Virtual channel buffers allow flits of different packets to be interleaved on same physical channel
- ❑ Prevents head-of-line blocking
- ❑ Example: Tail flit of green packet is blocked, so flits of red packet proceed across channel into VC2
- ❑ Most wormhole networks use VCs



Components of a Router

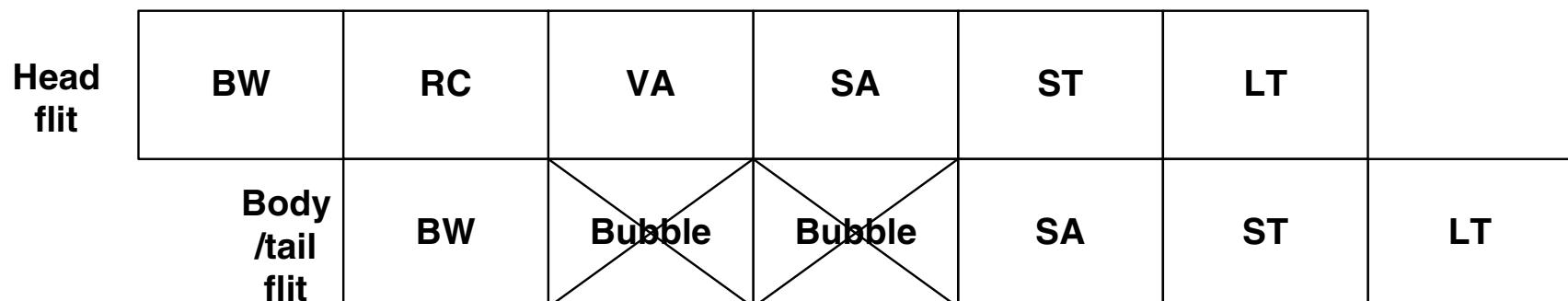


Components of a Router



Router Pipeline

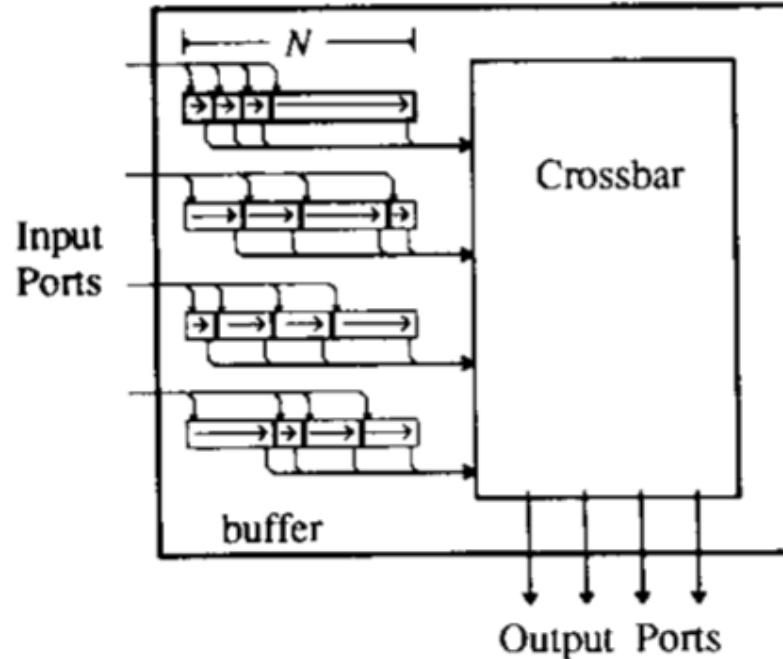
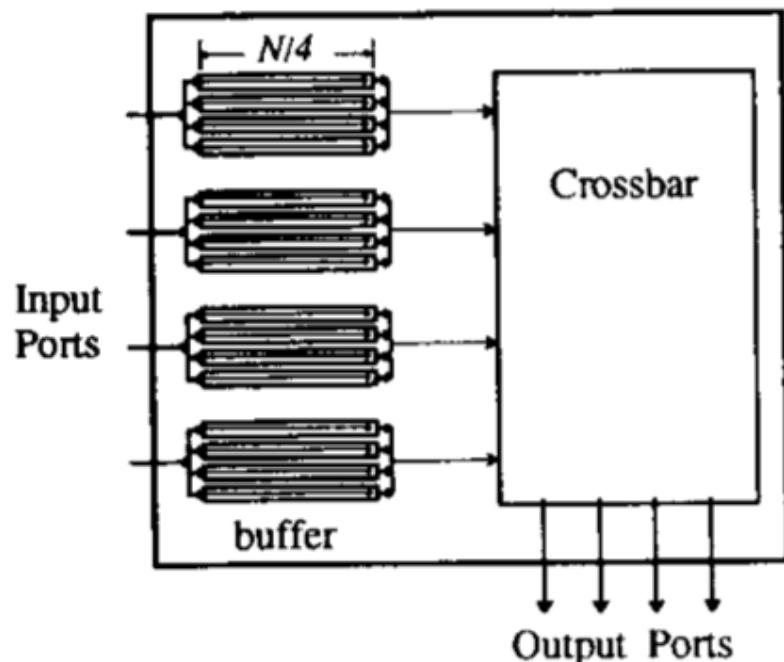
- ❑ Pipelining increases frequency and bandwidth
- ❑ 4 or 5 stage pipeline in high performance routers
- ❑ Only head flit uses all stages of pipeline, body and tail just follow
- ❑ Crossbar switch traversal is often critical path in router and also largest area



(a) Basic 5-stage pipeline (BASE)

Buffer Organization

- Should buffers share slots across virtual channels?
- Possibly better buffer utilization
- Increased control complexity when sharing, and possibility of undesirable interactions between VCs



Crossbar Circuit

- Recall that reuse enables high design effort in NoCs

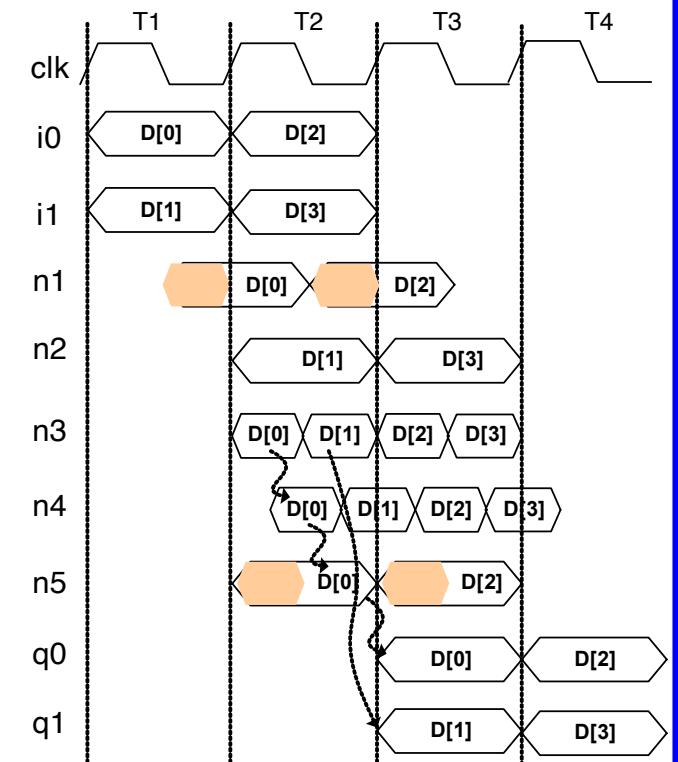
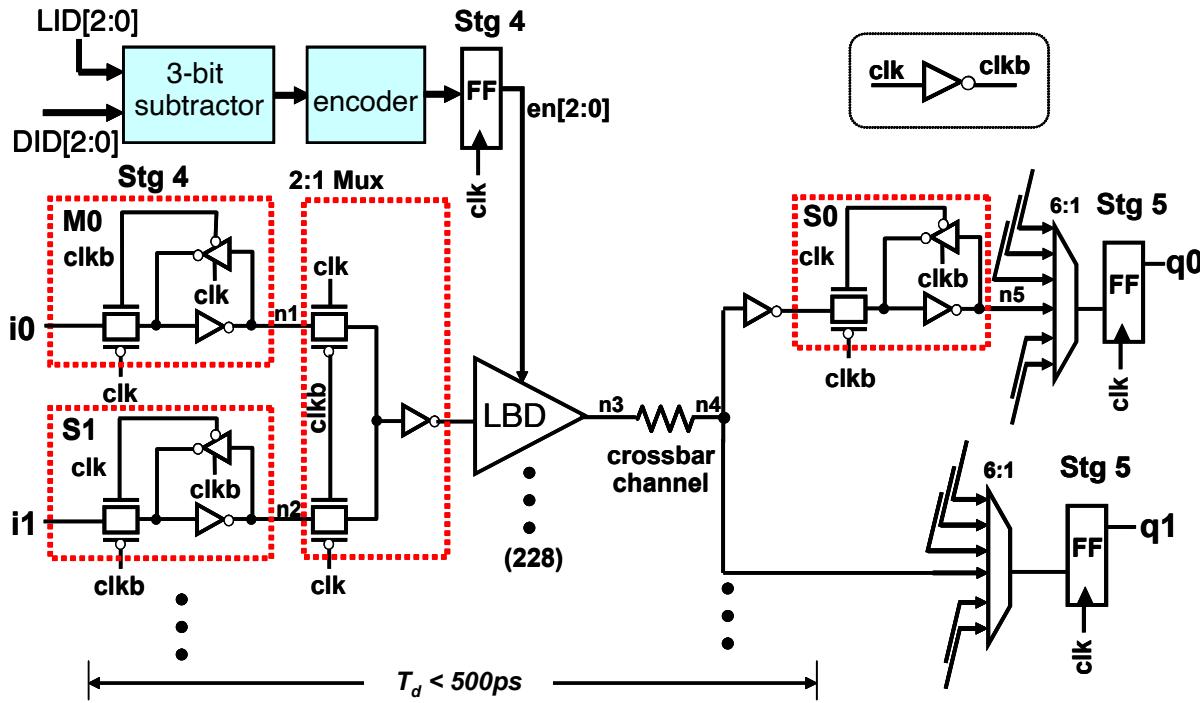
A Six-Port 57GB/s Double-Pumped Nonblocking Router Core

Sriram Vangal^{*++}, Nitin Borkar^{*} and Atila Alvandpour^{*}

^{*}Microprocessor Technology Labs, Intel Corporation, Hillsboro, OR, USA

^{**}Electronic Devices, Dept. of Electrical Engineering, Linköping University, Linköping, Sweden

2005 Symposium on VLSI Circuits Digest of Technical Papers



Crossbar Circuit

- Recall that reuse enables high design effort in NoCs

A Six-Port 57GB/s Double-Pumped Nonblocking Router Core

Sriram Vangal^{*†}, Nitin Borkar^{*} and Atila Alvandpour^{*}

^{*}Microprocessor Technology Labs, Intel Corporation, Hillsboro, OR, USA

^{*}Electronic Devices, Dept. of Electrical Engineering, Linköping University, Linköping, Sweden

2005 Symposium on VLSI Circuits Digest of Technical Papers

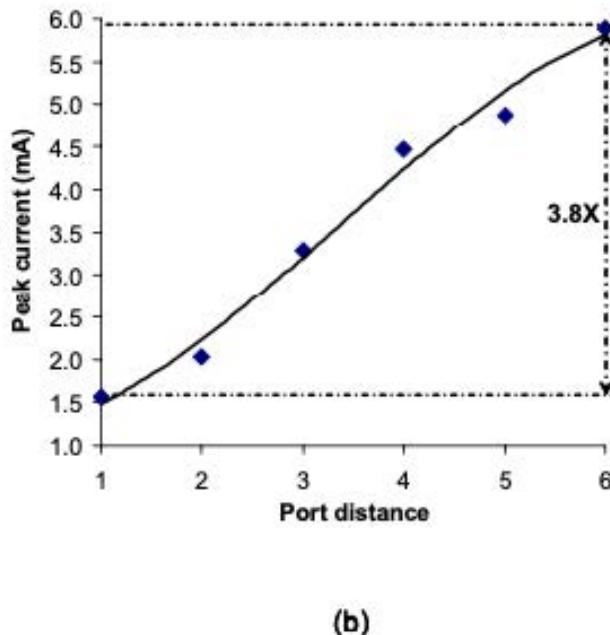
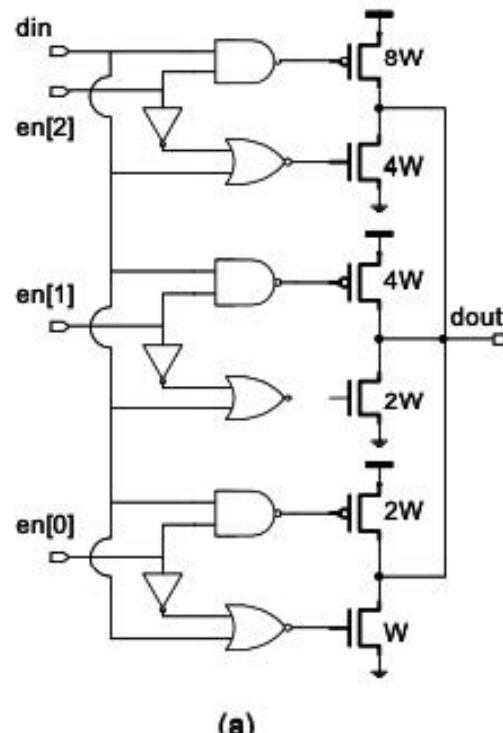


Fig. 5 (a) LBD schematic (b) Peak current per LBD vs. port distance.

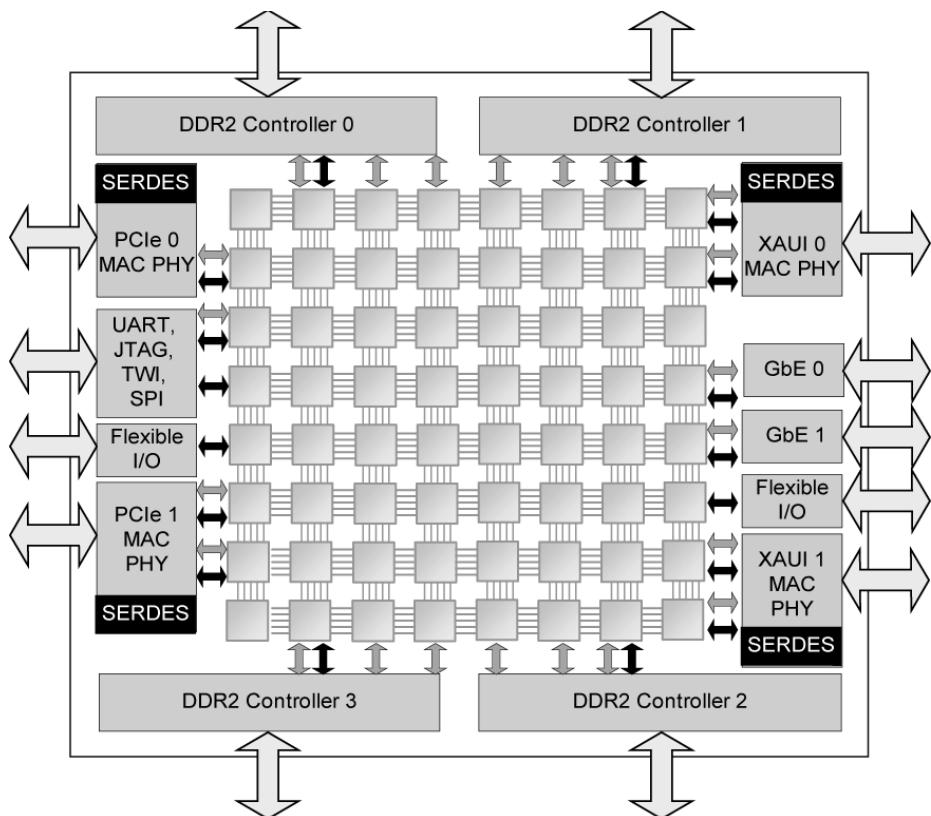
Changing drive strength according to physical distance from driver to receiver.

Drive longer wires with a smaller effective resistance

Why showing peak current and not average power consumption?

Tilera TILE64

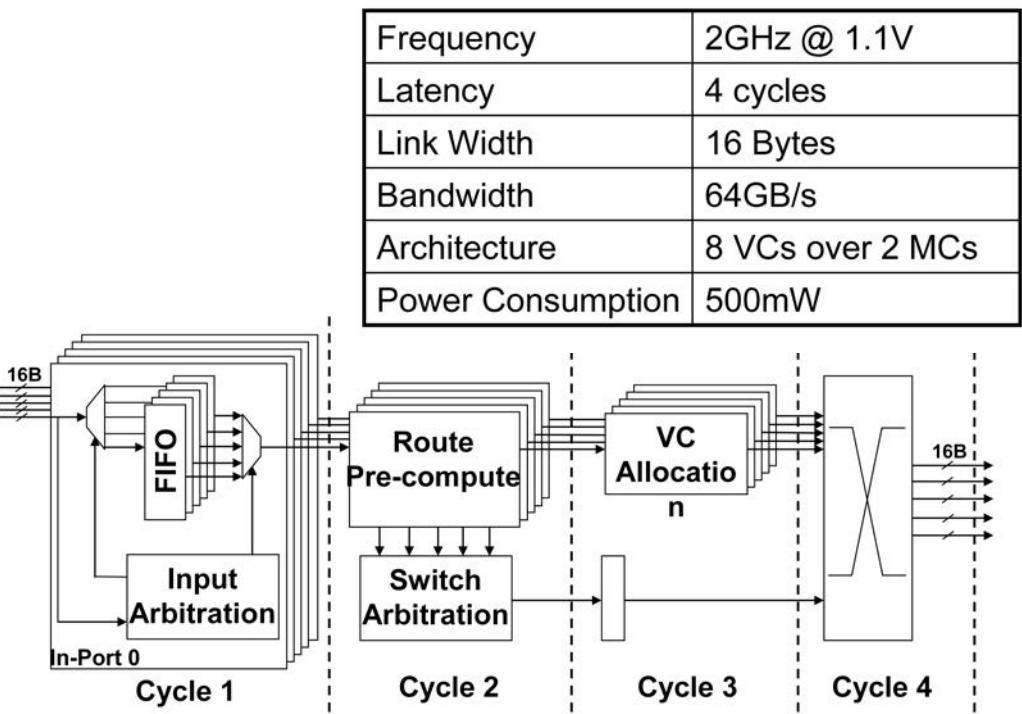
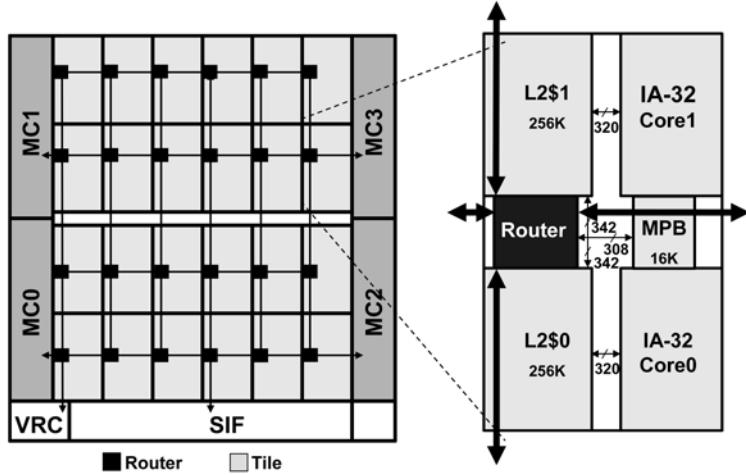
- 64 VLIW core tiles in 8x8 mesh
- Memory controllers and IOs around edges
- Simple router: non-pipelined, low-cost, 700MHz
- 5 independent networks for different traffic classes



- 2 Meshes for Memory Communication
 - Memory Dynamic Network (MDN)
 - Tile Dynamic Network (TDN)
- 3 Meshes for Register Mapped Communication
 - I/O Dynamic Network (IDN)
 - User Dynamic Network (UDN)
 - Static Network (STN)

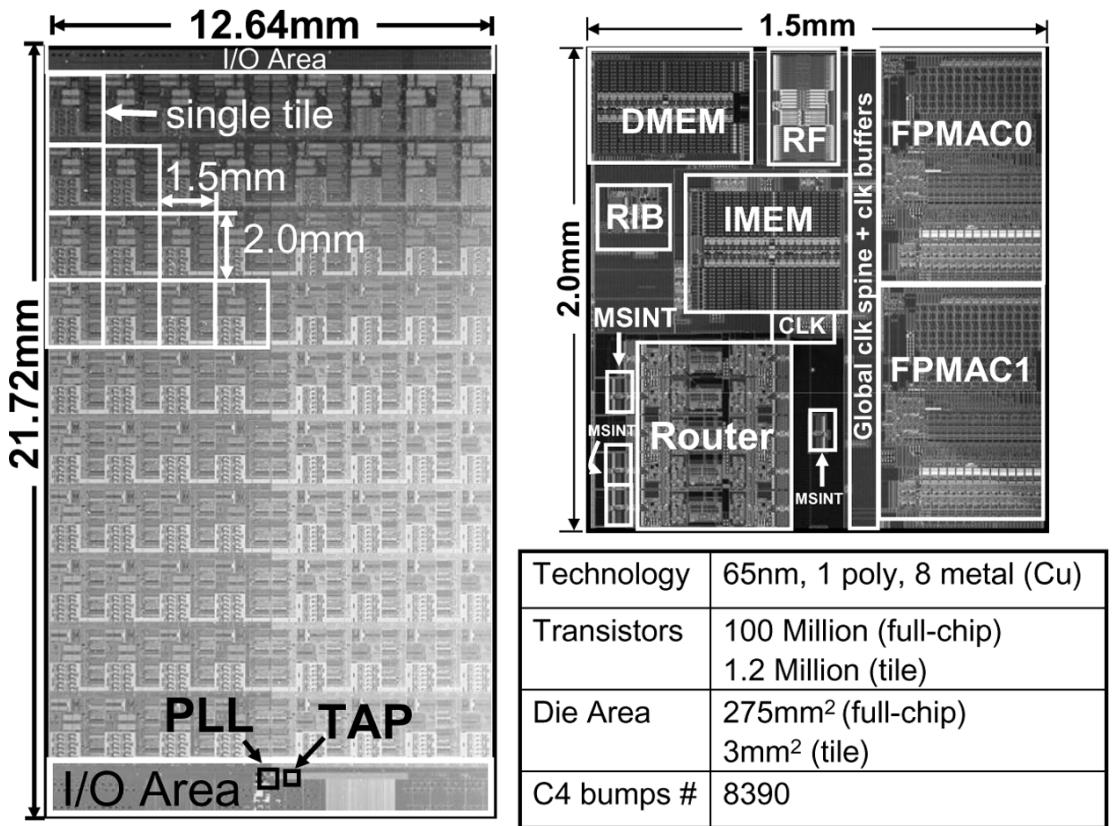
Intel 48-core

- Mesh topology with memory controllers at edges
- 24 tiles with 2 cores and 2 caches per tile
- XY Dimension ordered routing
- 4 stage high performance router pipeline
- 45nm technology. DVFS with 8 voltage islands



Intel TeraFLOPS

- 80 tiles in 8x10 mesh
- Capable of running at 5GHz
(5 stage pipeline and small tiles)
- 2 VCs per channel,
each with depth 16
- Uses
double-pumped
crossbar

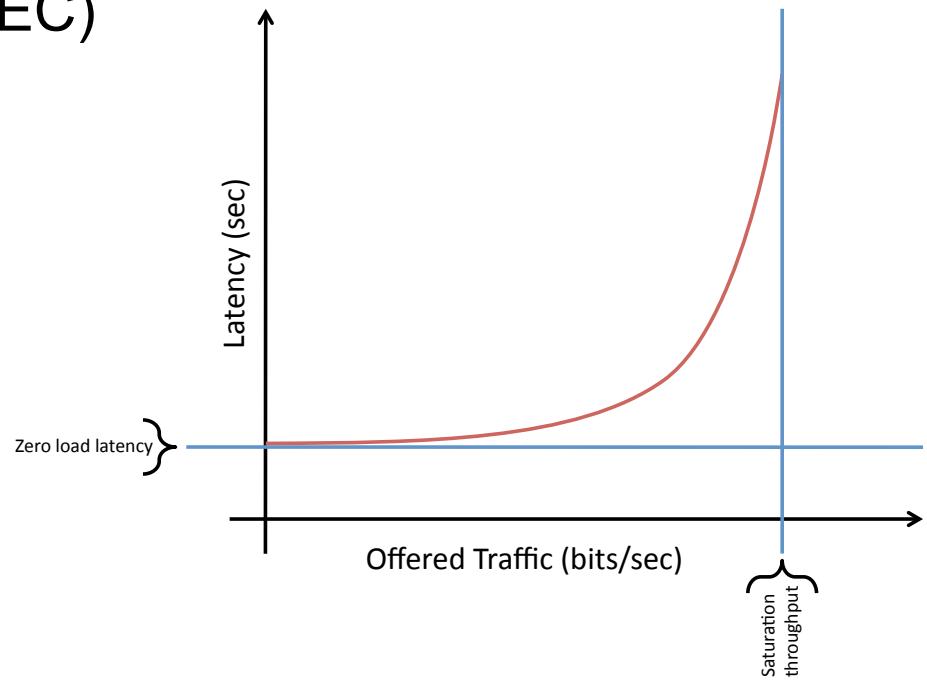


Quality of Service Metrics in NoCs

- Bandwidth of network or of source-destination pairs
- Latency from a source to destination
- With idle network or congested network
- Jitter in arrival time of packets
(not VLSI clock jitter, but same idea)

Traffic Modeling

- ❑ Networks start to clog when injected traffic approaches saturation throughput
- ❑ Performance can suffer well before network throughput limits if traffic has hot spots
 - Test with synthetic traffic generators
 - Or benchmarks (e.g. PARSEC)



Review Questions

- ❑ Between Torus and Folded Torus, why is one better suited for VLSI chips?
- ❑ Explain the difference between wormhole flow control and store-and-forward flow control?
- ❑ What is a flit, and how does the size of a flit relate to the width of channels and input buffers?
- ❑ Bonus: if a certain class of traffic (e.g. memory requests) is latency critical, how might a network be modified to prioritize this traffic and decrease its latency?