NOC Router Micro-Architecture

COE838/EE8221: Systems-on-Chip Design
http://www.ecb.torontomu.ca/~courses/coe838/

Dr. Gul N. Khan
http://www.ecb.torontomu.ca/~gnkhan
Electrical, Computer & Biomedical Engineering
Toronto Metropolitan University

Overview

- Introduction to NoC Switch/Router
- Router Architecture
- Buffer and Crossbar Switch
- Virtual Channel Router
- NoC based SoC

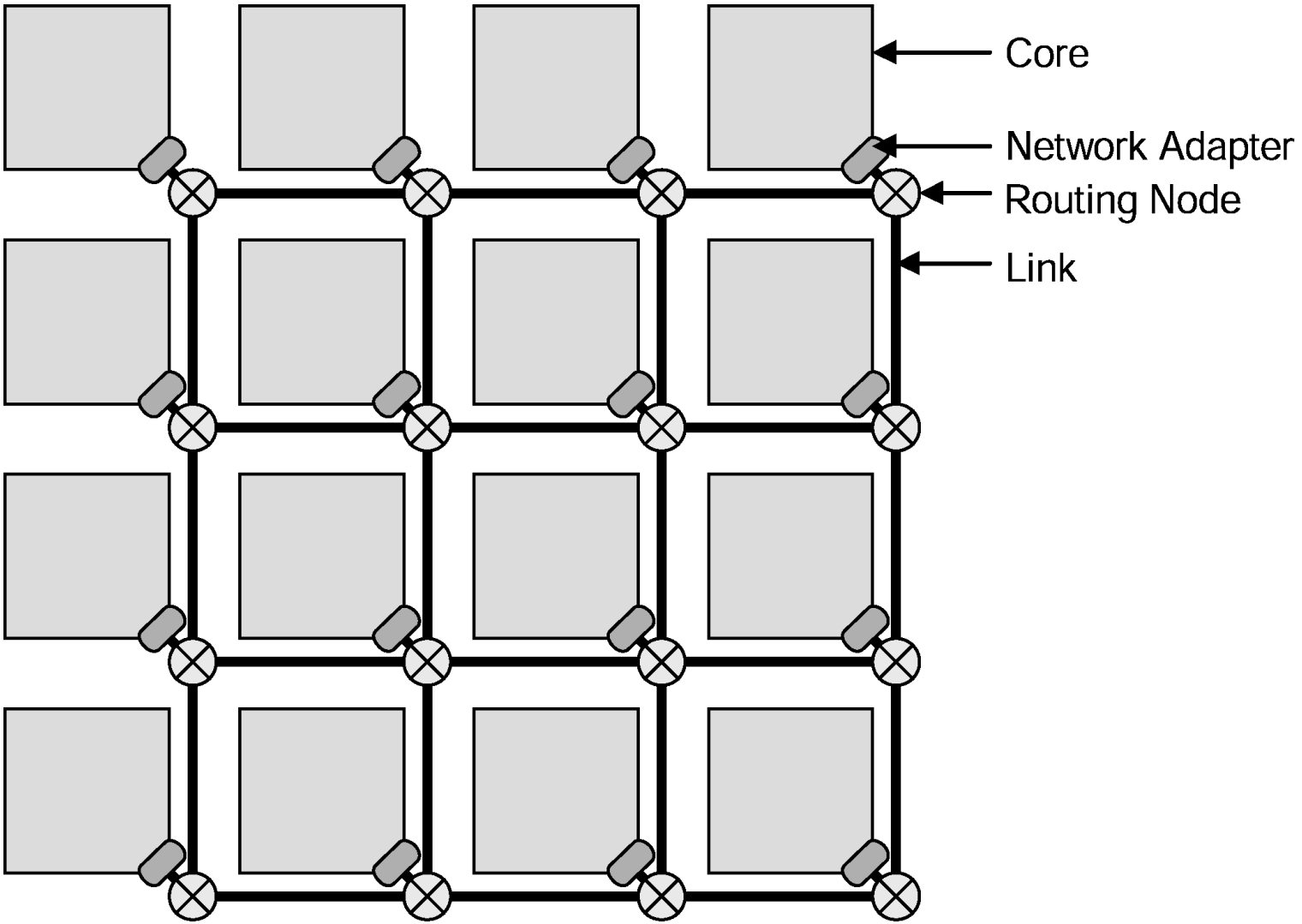
On-chip networks by N. E. Jerger, T. Krishna, Li-Shiuan Peh. Publisher Morgan & Claypool, 2017.
Chapter 12: On-Chip Communication Architectures – SoC Interconnect by S. Pasricha & N. Dutt.

NoC/Router: Overview

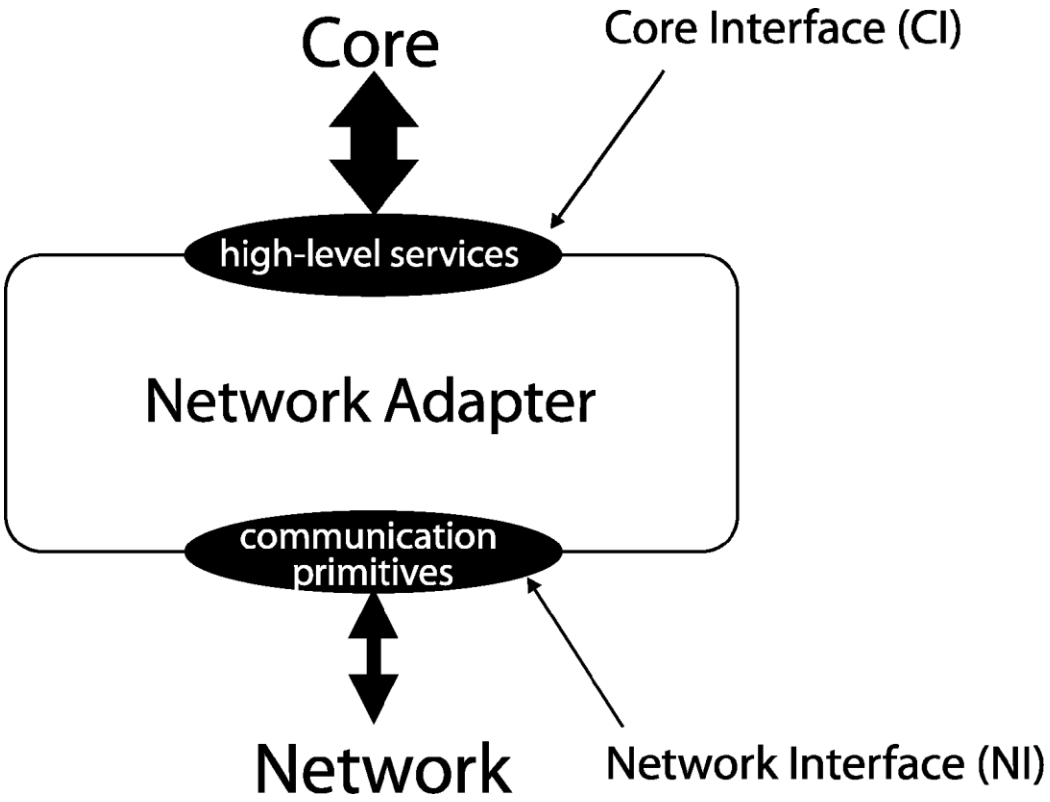
- NoC Topology: Connectivity
- Routing: Channels and paths
- Flow control: Resource Allocation

  (Buffer, Links, VC, etc.)
- Router microarchitecture: Design and Implementation of Routing, Flow control and Router pipeline

Network-on-Chip

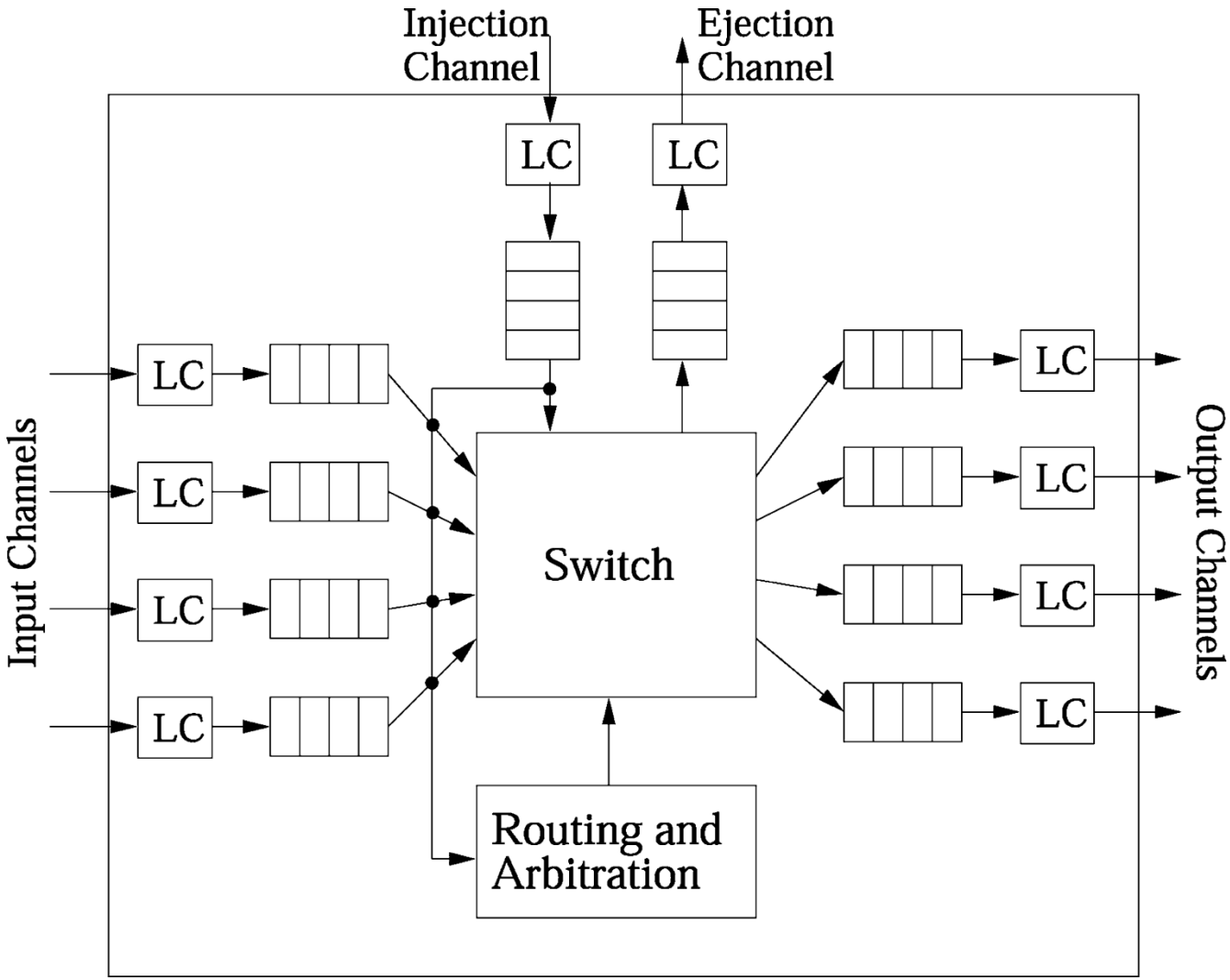


Core to Network Connection



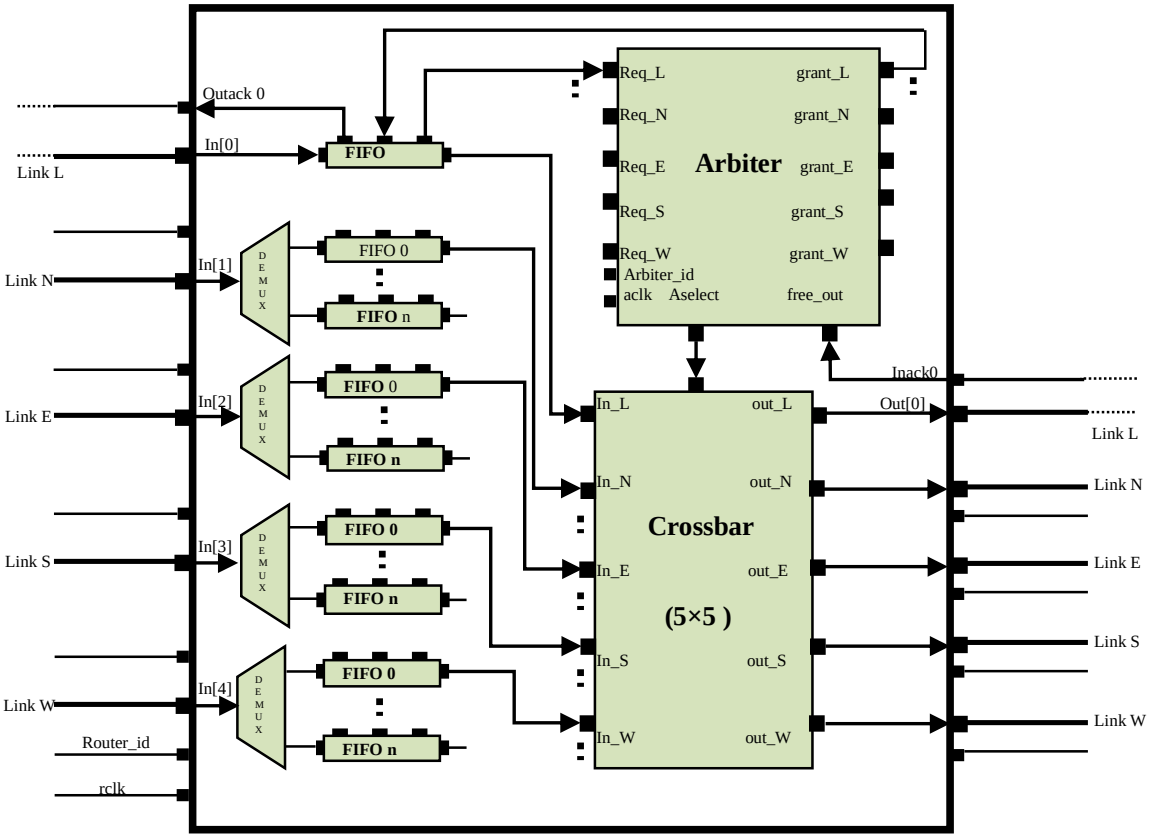
NoC

NoC Router



Generic Router/Switch

Generic NoC Router



Buffers, Arbiter, Crossbar Switch, etc.

Baseline Router Components

- Input Buffers (FIFOs)
- Route Computation
- Arbiter (Switch Allocator)
- Crossbar Switch

Most NoC routers have buffers at the inputs.

Buffer store flits in the router being traversed.

Router Components

There are mainly three components:

Buffer: Buffers are organized as FIFOs for each input channel. There is no need of buffer for the output channels.

Arbiter: It is a switch allocator for the generic router. For VC (virtual channel) router arbiter will also perform VC allocation.

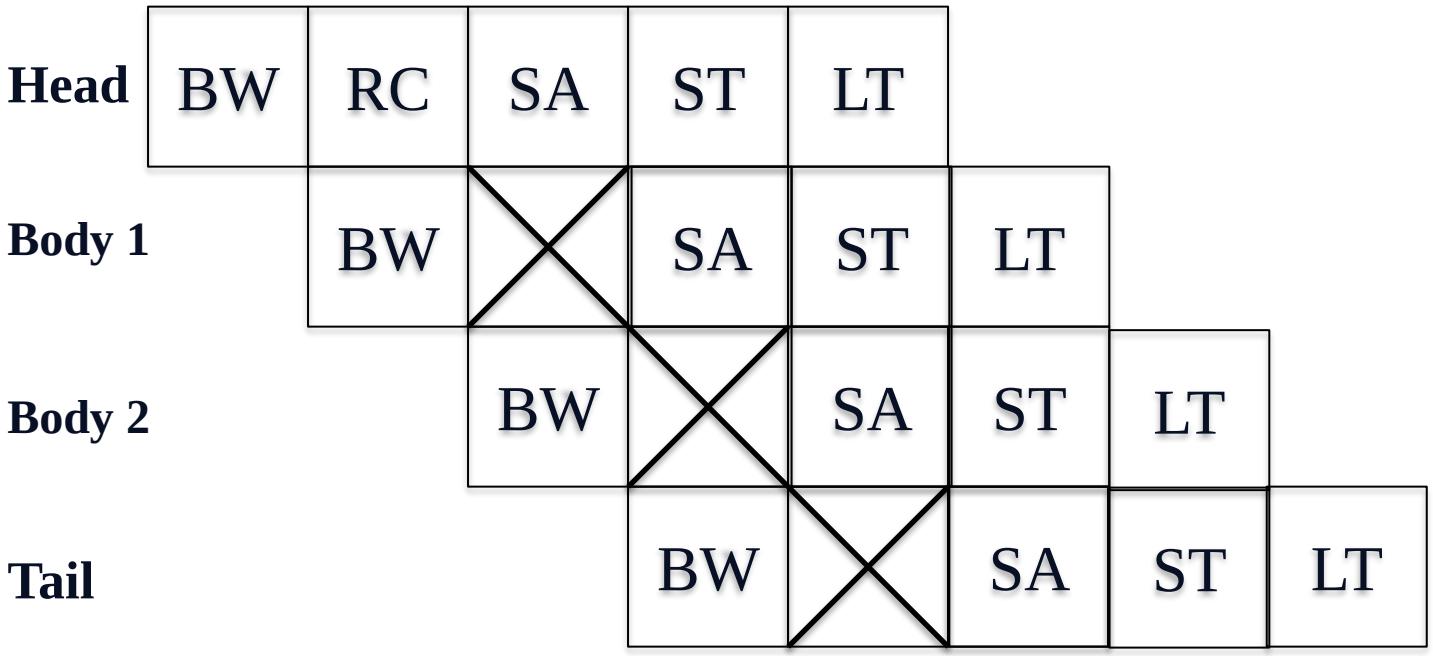
Switch: Crossbar switch is the most suitable for higher performance.

Baseline Router Pipeline

5-stage pipeline

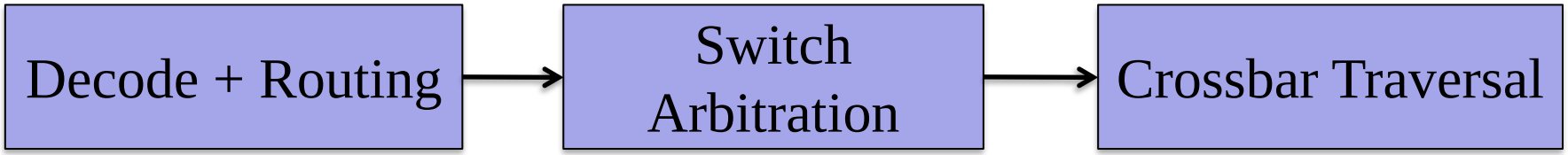
- BW: Buffer Write
- RC: Routing Computation
- SA: Switch Allocation
- ST: Switch Traversal
- LT: Link Traversal

Baseline Router Pipeline



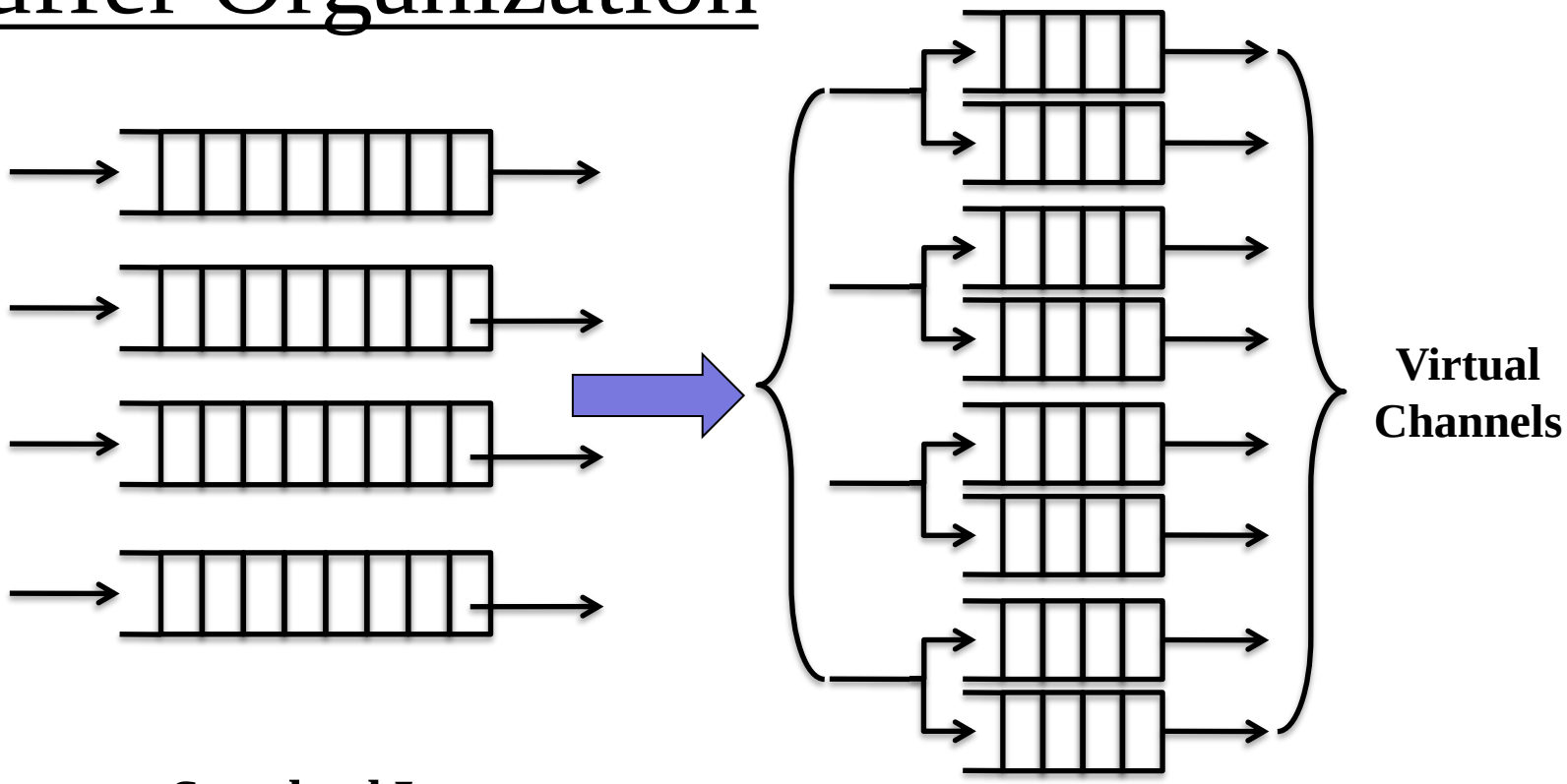
- RC (Routing calculation) done once per packet
- Body and tail flits inherit this info from the head flit

Dependencies and Modules
Wormhole Router



- Dependence between output of one module and input of another
  - Determine critical path through the router.
  - Cannot bid for switch port until routing performed.

Buffer Organization

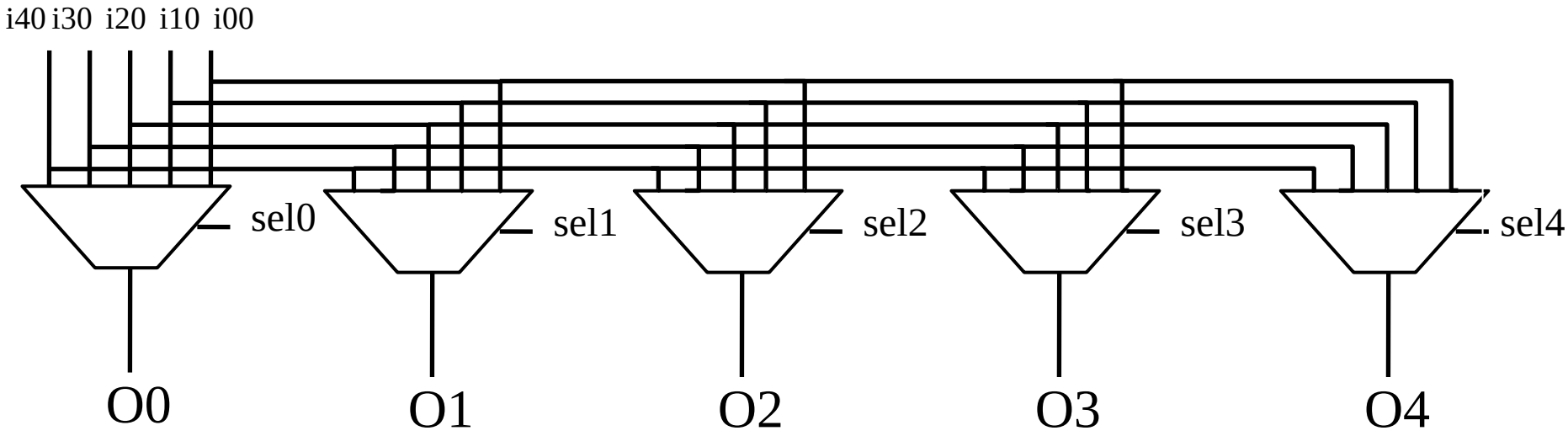


- One buffer per input channel. Multiple VCs share the buffer.
- Multiple queues (VCs) per physical channel (prevent deadlock)
  - Each VC must have minimum 1 flit buffer
  - More complex logic circuits.

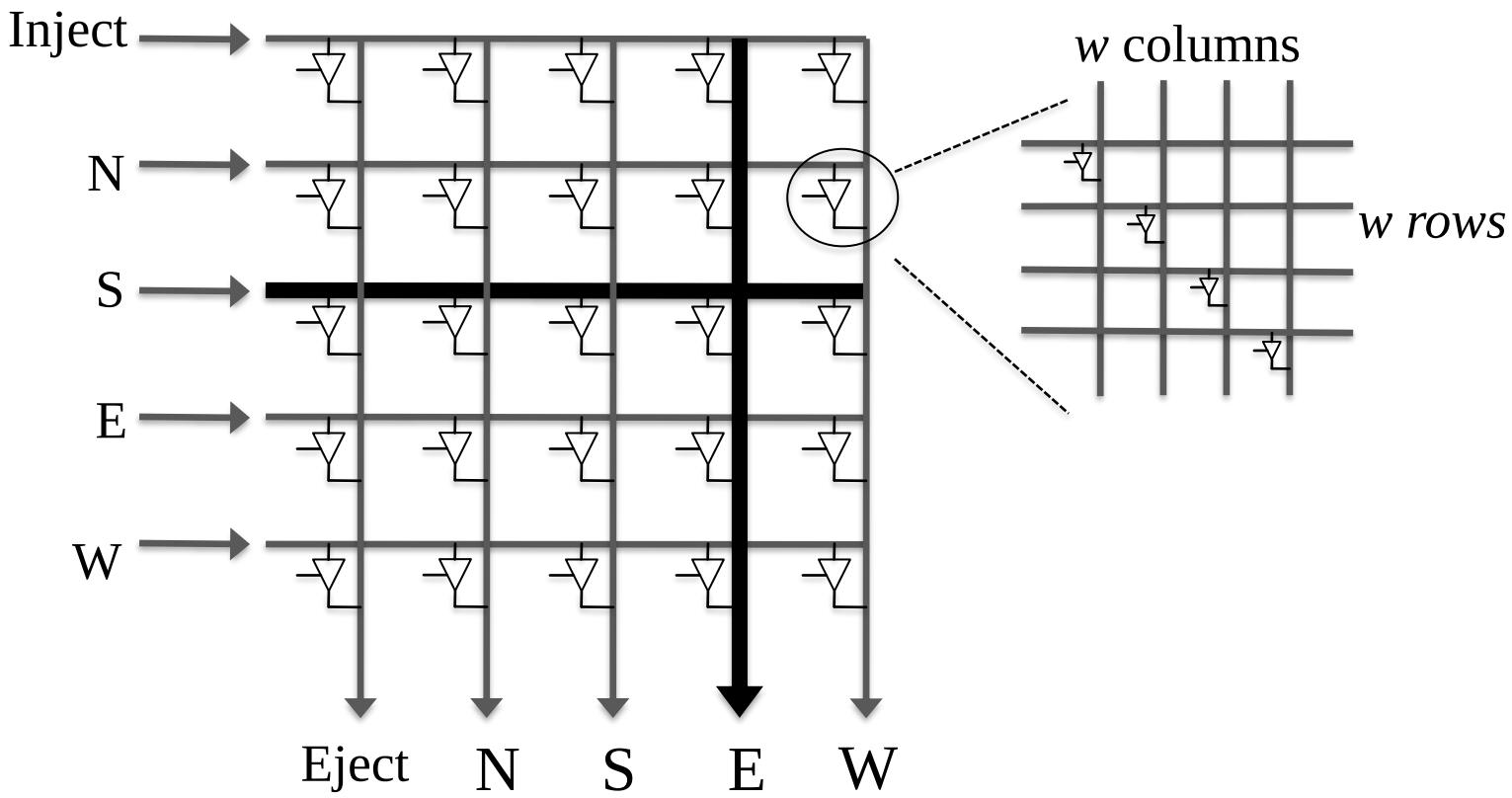
Switch Organization

- Central part of the router Datapath

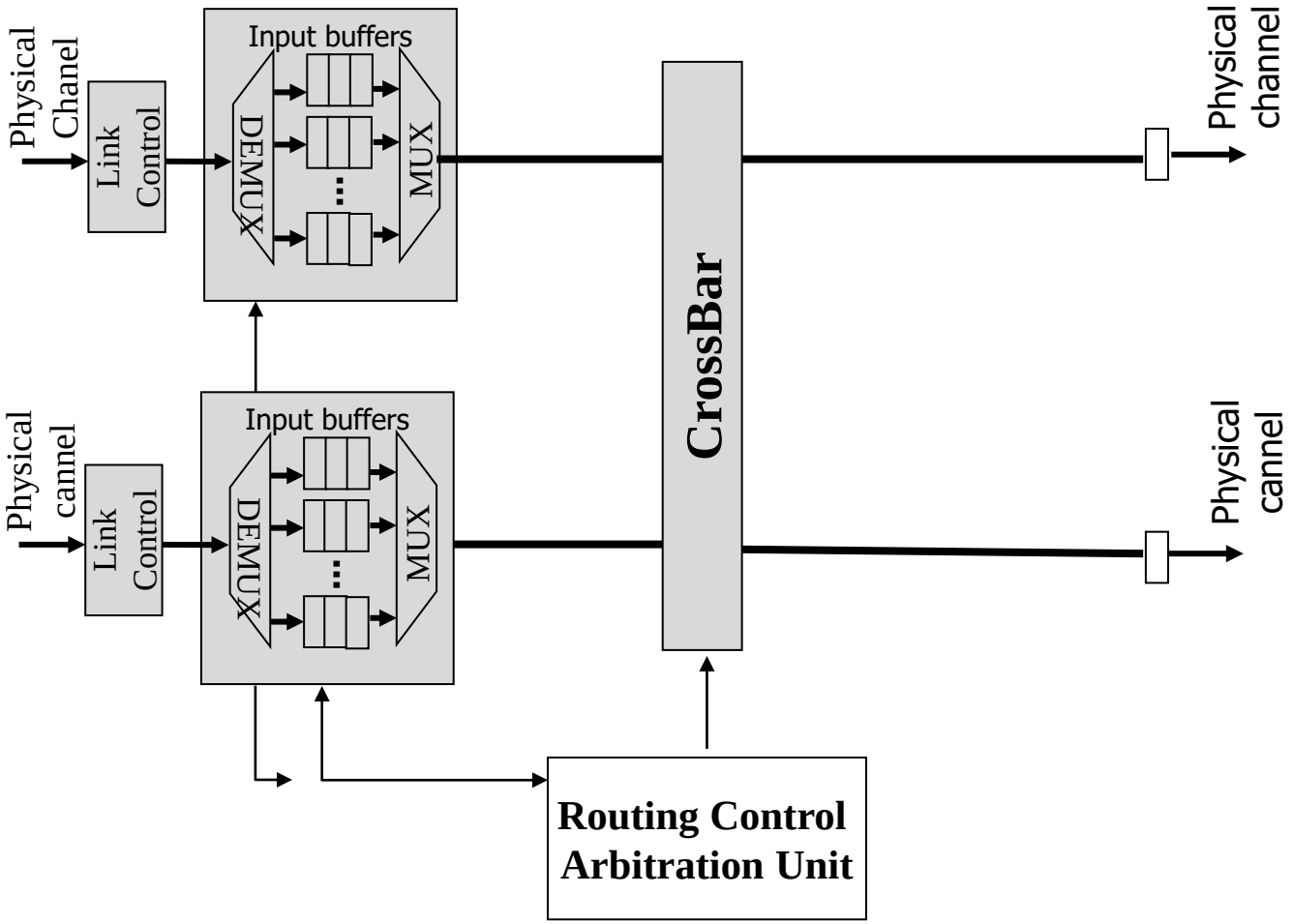
  Switches connect the data from input to output.
- Crossbar can be made of many multiplexers
- Long wires that may be a problem.



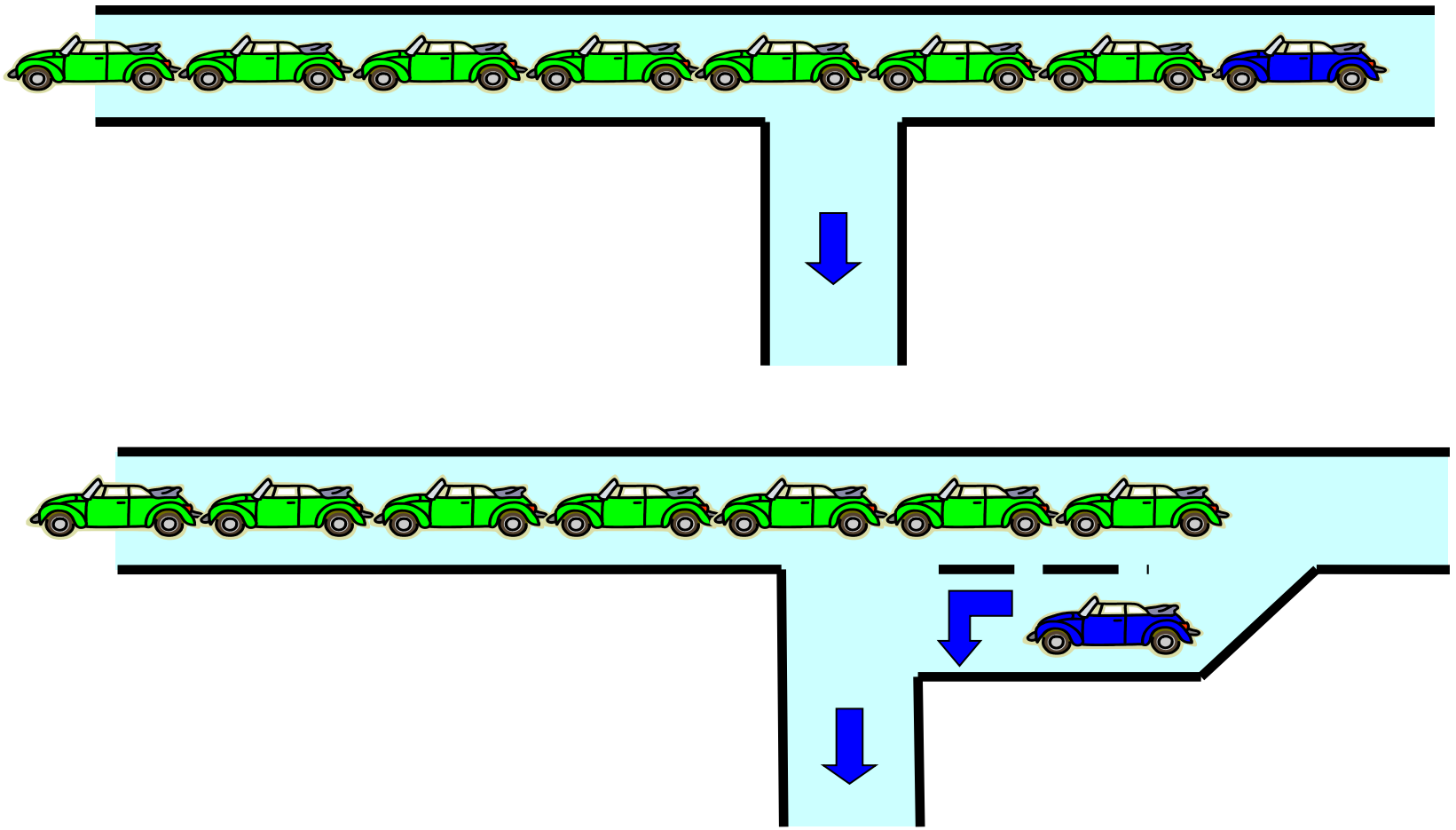
2D-Mesh NoC Router Switch Another Design



Switch Microarchitecture



VC: Virtual-Channels



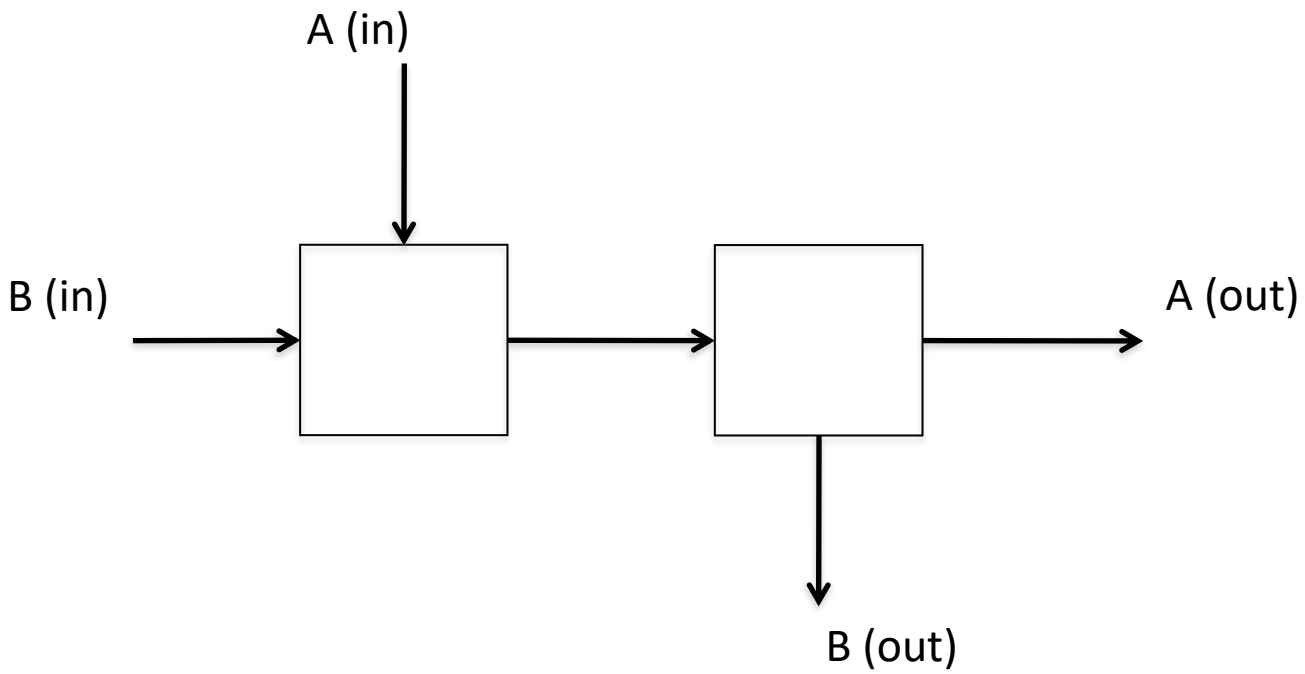
VC: Virtual Channels

- First proposed for deadlock avoidance.
- Handles contention and deadlock by providing multiple paths for packets
- Can be applied to any flow control
  - It can be used with wormhole.
- Multiple flit queues per input port
  - Share **same** physical link (channel)
- Link utilization is improved
  - Flits on different VC can pass the blocked packet

Virtual Channel Flow Control

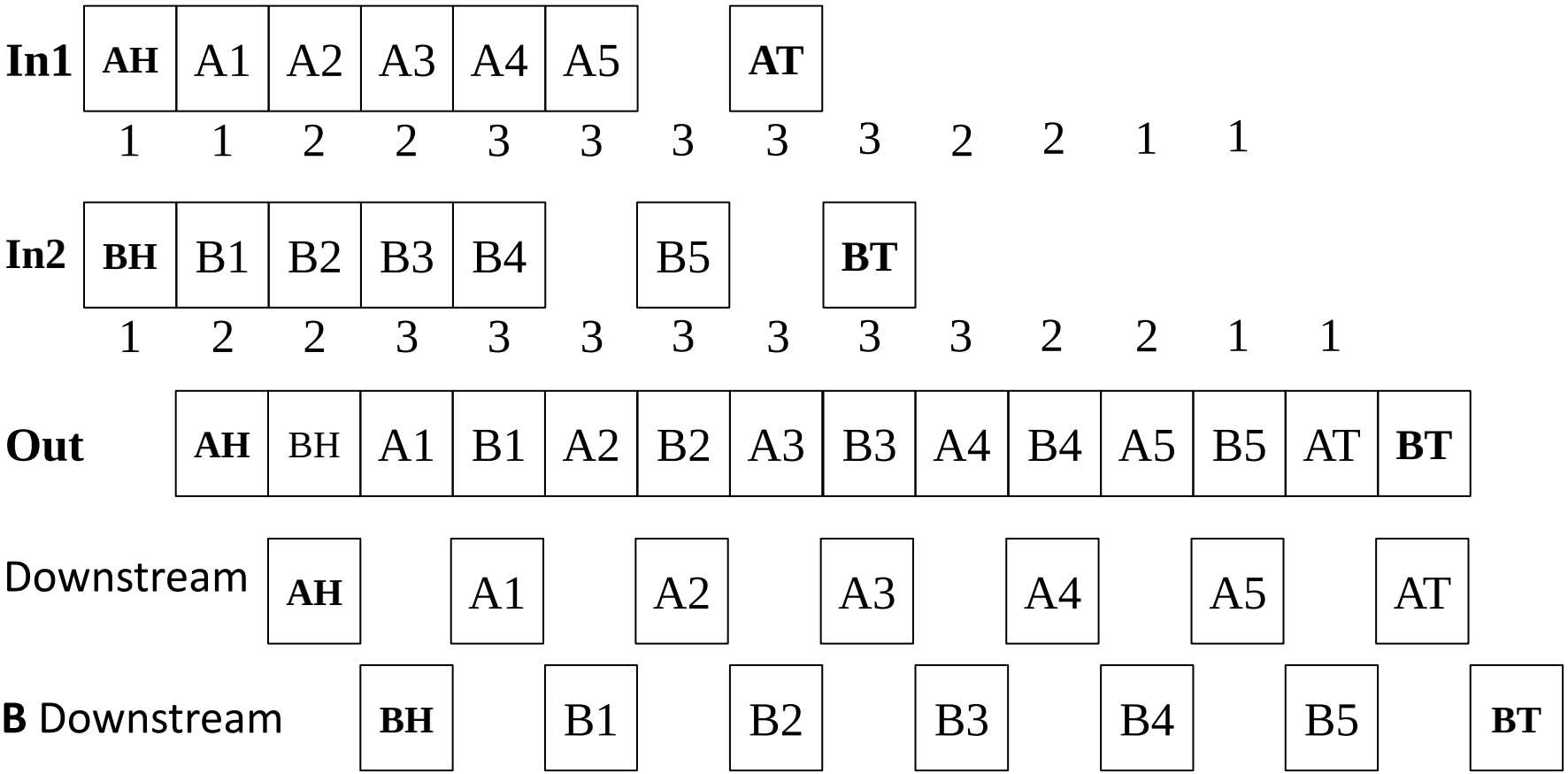
- VCs provide more than one input/output path per channel.

Consider two packets A and B travelling through a router towards different paths for destinations



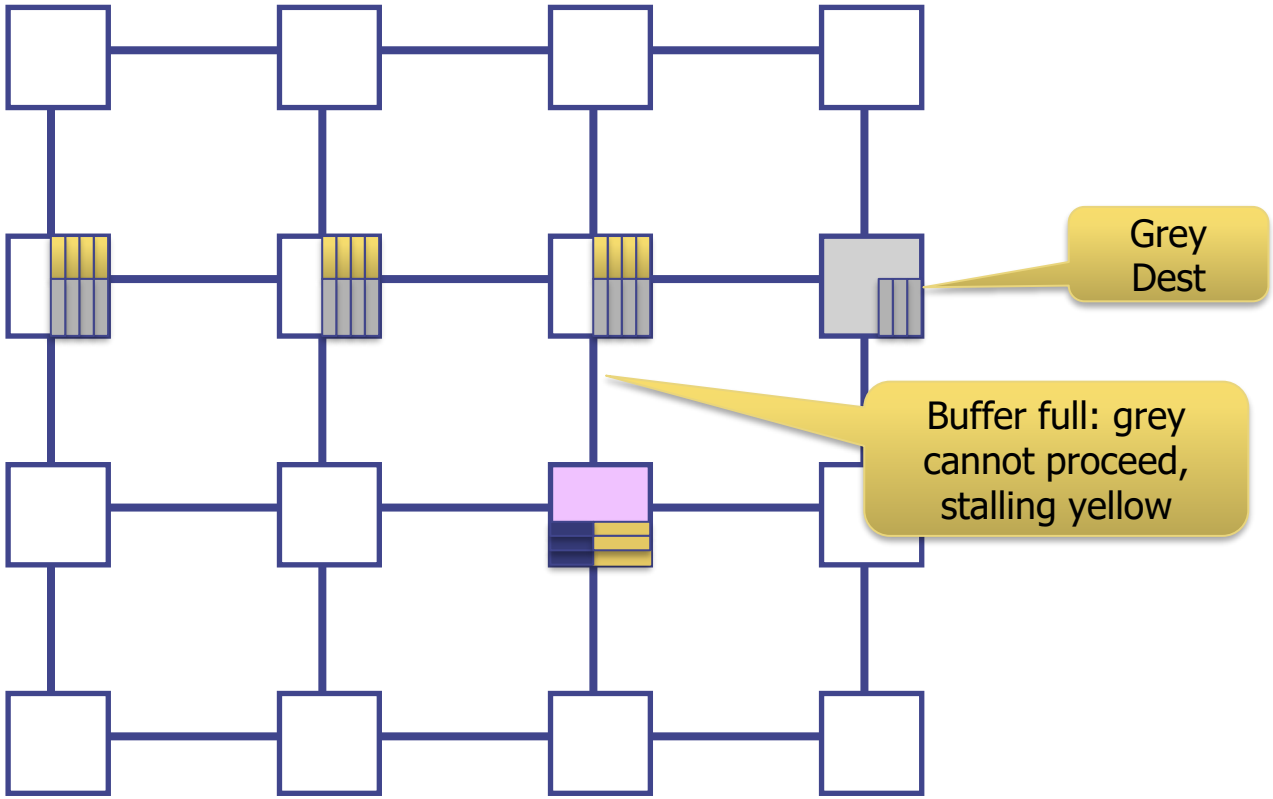
Virtual Channel Flow Control

AH and AT (Packet A Head and Tail respectively)
BH and BT (Packet A Head and Tail respectively



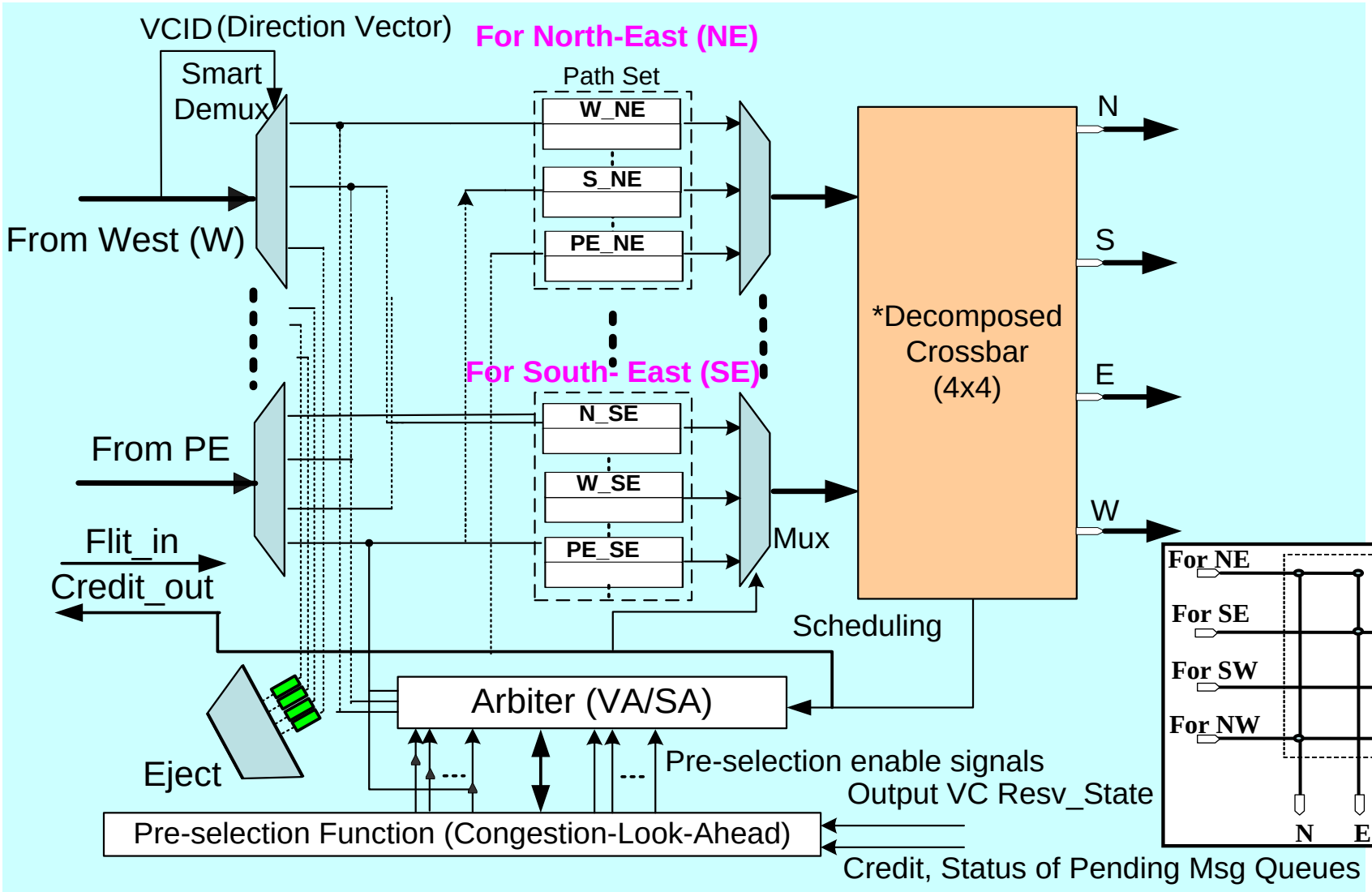
Virtual Channel Flow Control

- Packets compete for VC flit-by-flit.
- For downstream links, flits of each packet are available every other cycle in the example.

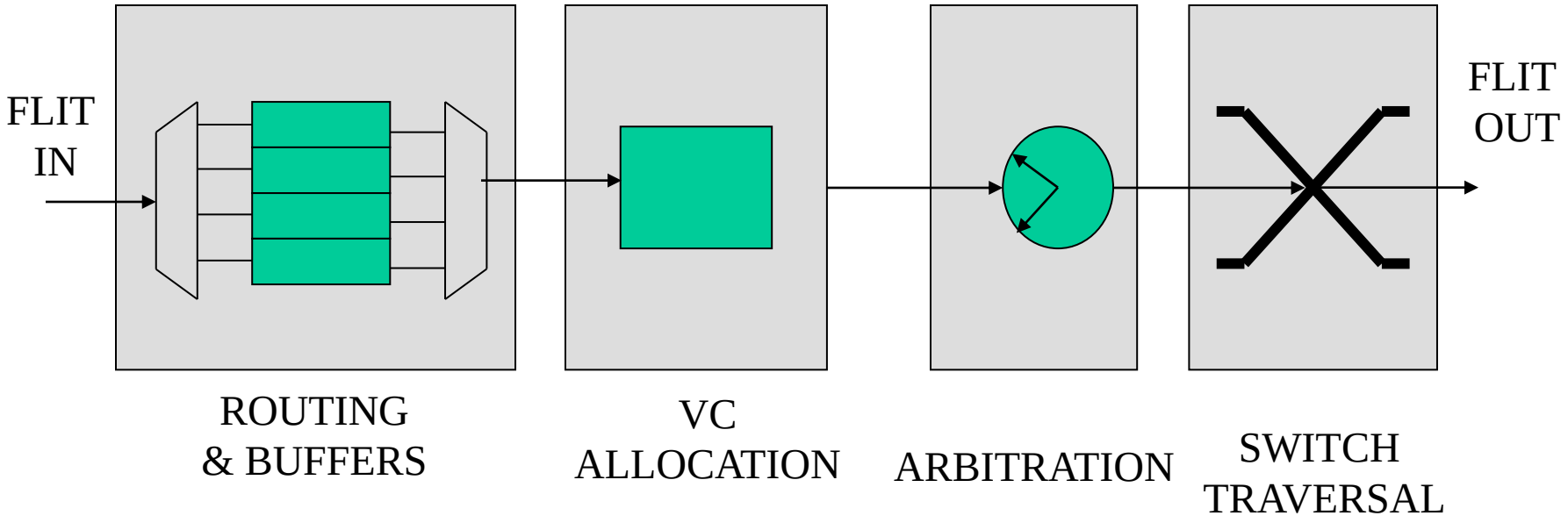


6 flit buffers/input port
3 flit buffers/VC

Virtual Channel 2D Router

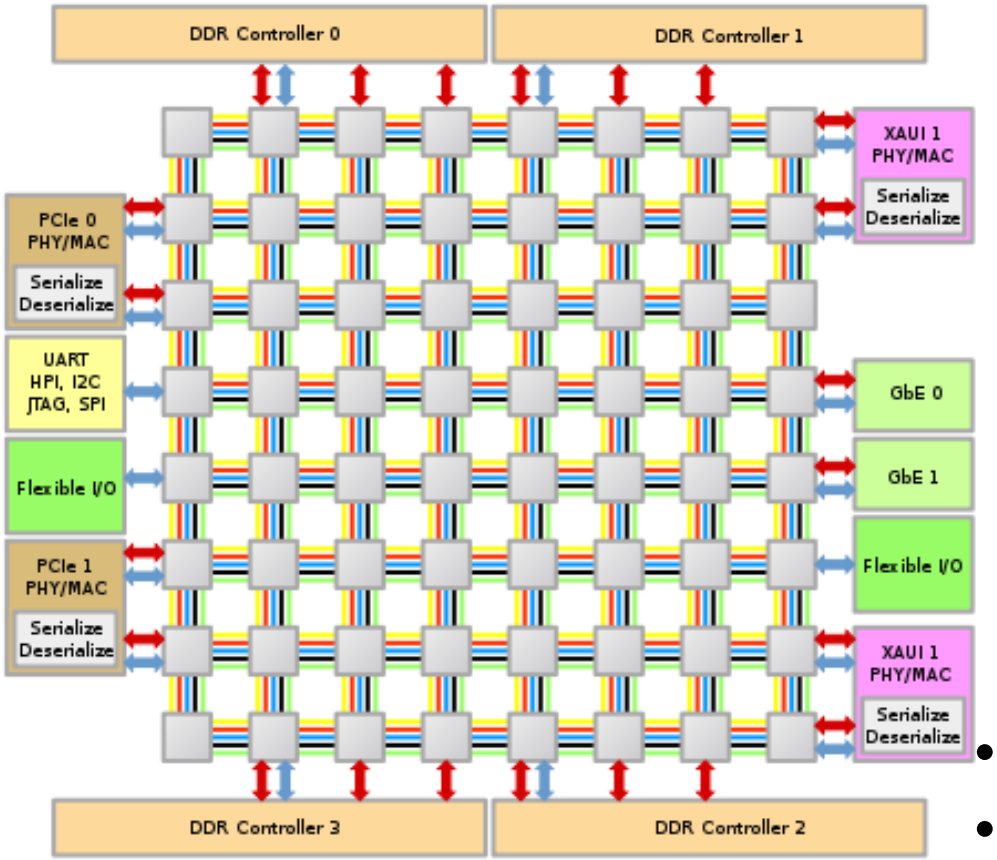


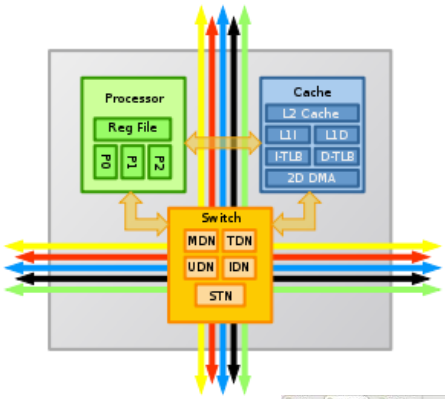
VC-based Router Pipeline

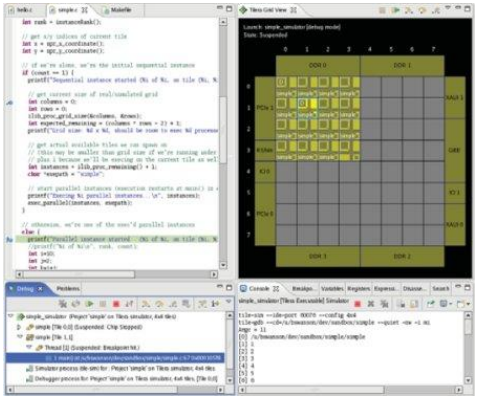


NoC based CPU SoCs

Tilera Tile64



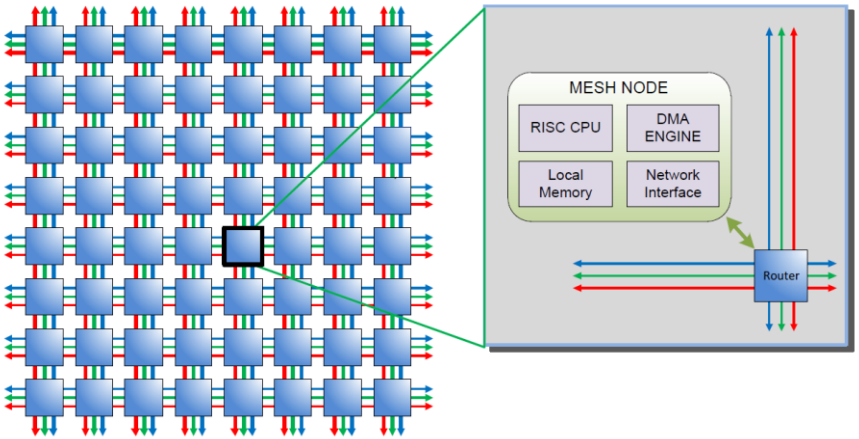


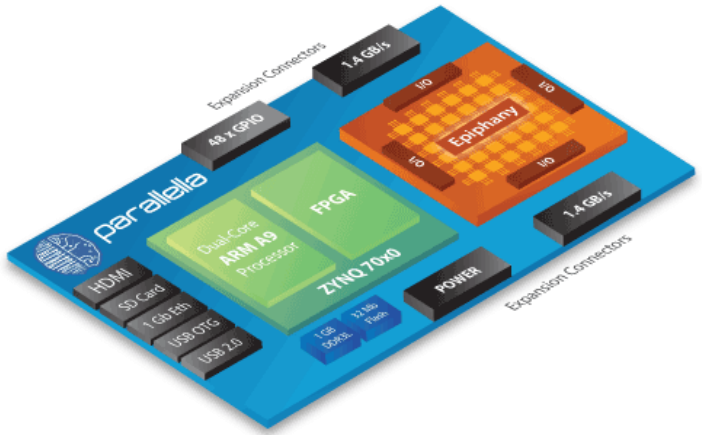


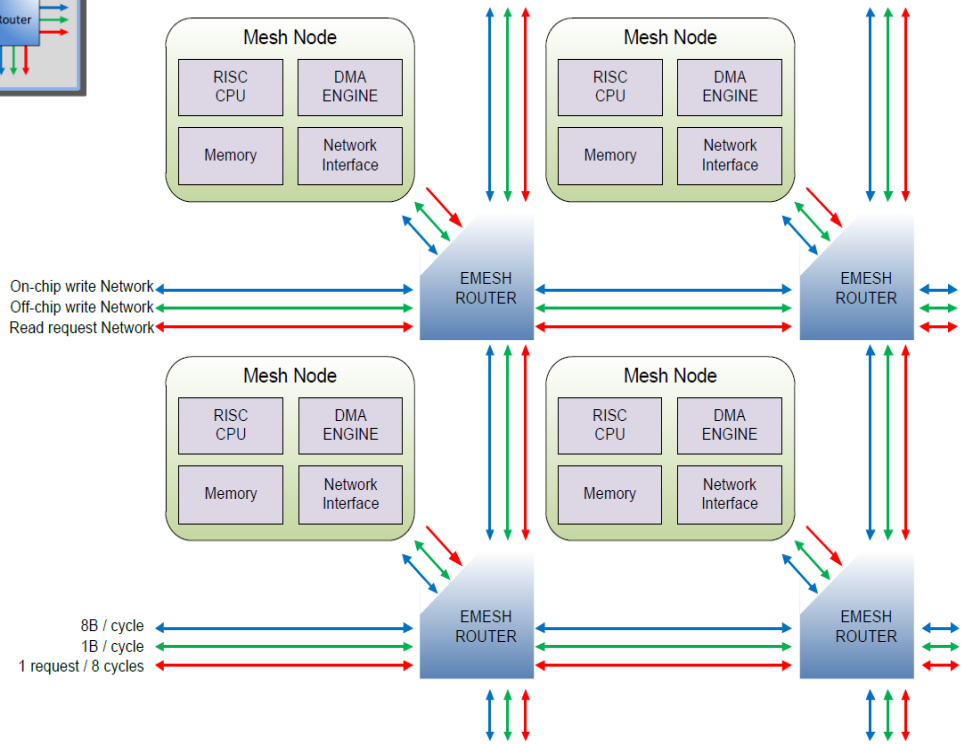
- 2D Mesh topology
- XY Routing
- Wormhole switching, no VCs
- Credit-based Flow Control

CPU/SoC based on NoC

Parallella – the Adapteva daughter card







Eminent CPU based on NoC

System	Topology	Routing	Switching	Flow ctrl
MIT RAW	2D mesh (32bit)	XY DOR	WH, no VC	Credit
UPMC SPIN	Fat Tree (32bit)	Up*/down*	WH, no VC	Credit
QuickSilver ACM	H-Tree (32bit)	Up*/down*	1-flit, no VC	Credit
UMass Amherst aSOC	2D mesh	Shortest-path	Pipelined CS, no VC	Timeslot
Sun T1	Crossbar (128bit)	-	-	ACK/NACK
Cell BE EIB	Ring (128bit)	Shortest-path	Pipelined CS, no VC	Credit
TRIPS (operand)	2D mesh (109bit)	YX DOR	1-flit, no VC	On/off
TRIPS (on-chip)	2D mesh (128bit)	YX DOR	WH, 4 VCs	Credit
Intel SCC	2D torus (32bit)	XY,YX DOR, odd-even TM	WH, no VC	On/off
TILE64 iMesh	2D mesh (32bit)	XY DOR	WH, no VC	Credit
Intel 80-core NoC	2-D mesh (32bit)	Source routing	WH, 2 lanes	On/off

Buses Versus NoCs

Buses

- Simple and familiar
- Bandwidth is limited, and speed decreases as no. of cores increase
- No concurrency, with centralized arbitration bottleneck

- Provides better chip area and power but it is not scalable.

NoCs

- Unfamiliar methodology
- Bandwidth grows as more cores added, with link speed unaffected
- Concurrent spatial reuse, distributed arbitration

- Extra delay of routers
- No performance guarantee
- Area and power overhead