Fine-Grained Bandwidth Adaptivity in Networks-on-Chip Using Bidirectional Channels

Robert Hesse, Jeff Nicholls, Natalie Enright Jerger

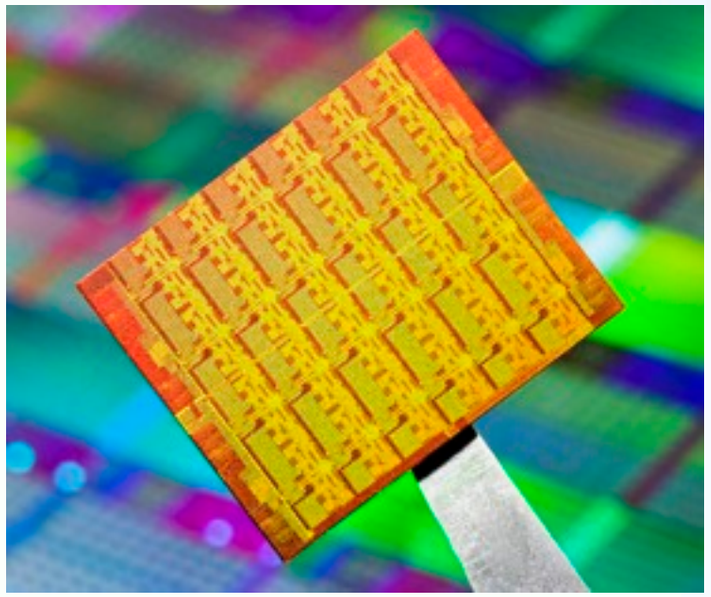
University of Toronto

May 10, 2012

Motivation

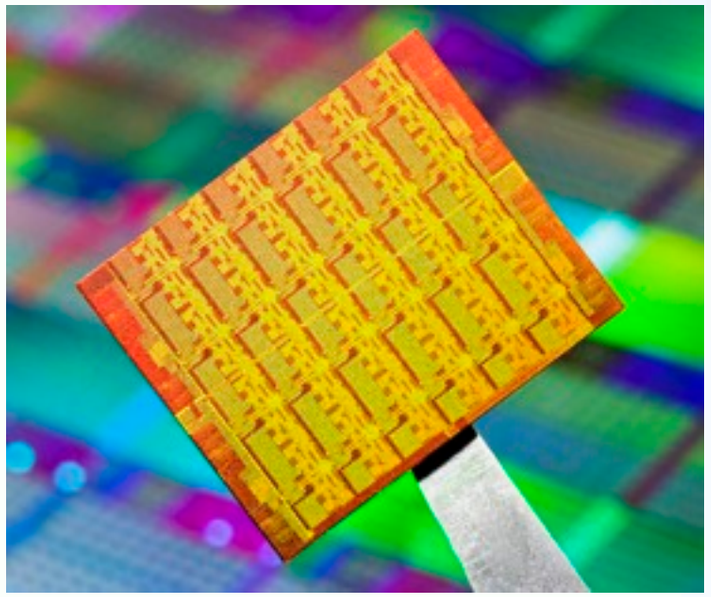
Motivation

- NoCs are crucial for scaling CMPs



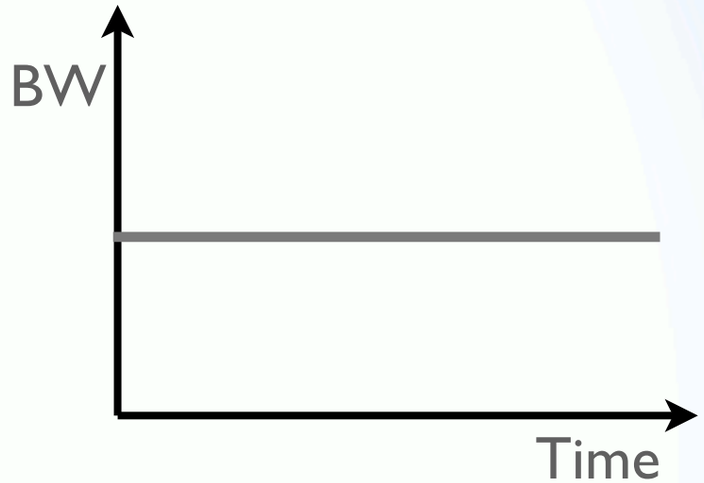
Motivation

- NoCs are crucial for scaling CMPs
- Problem:



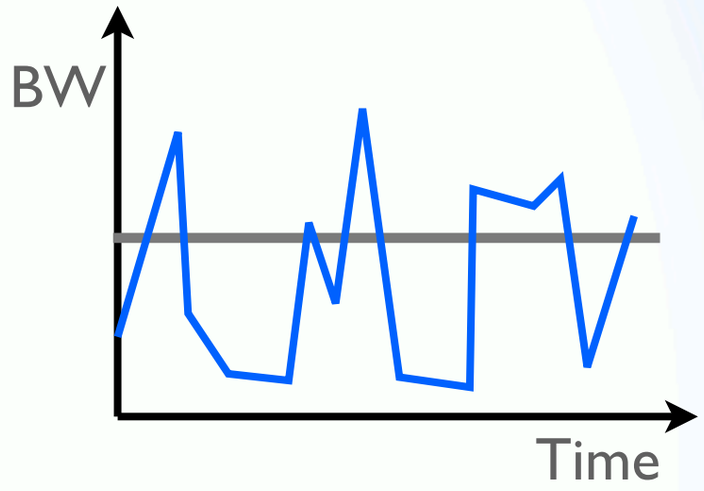
Motivation

- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static



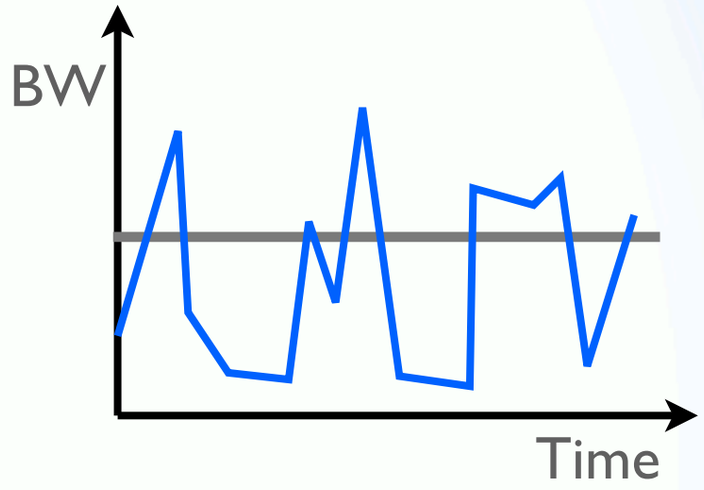
Motivation

- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic



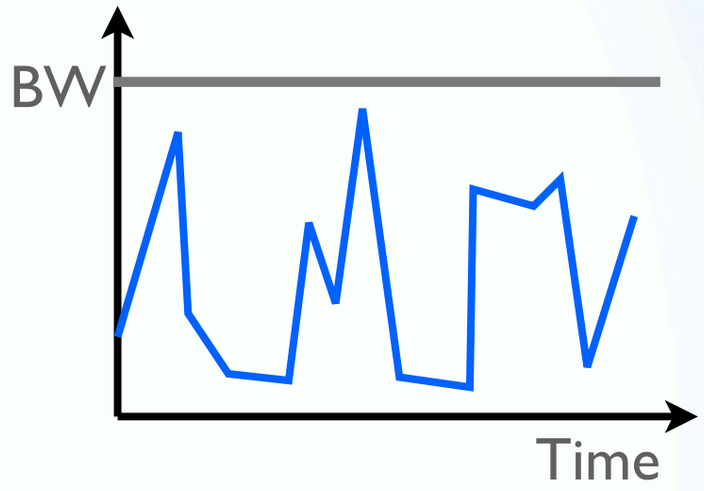
Motivation

- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic
- Current solution:



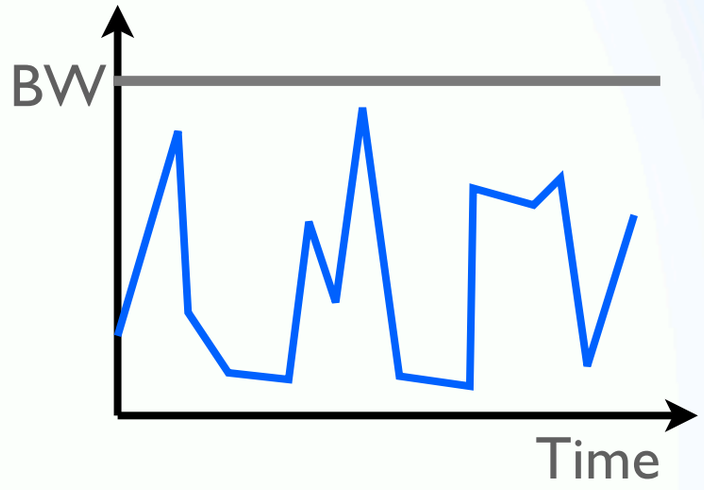
Motivation

- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic
- Current solution:
  - Over-provisioned link BW



Motivation

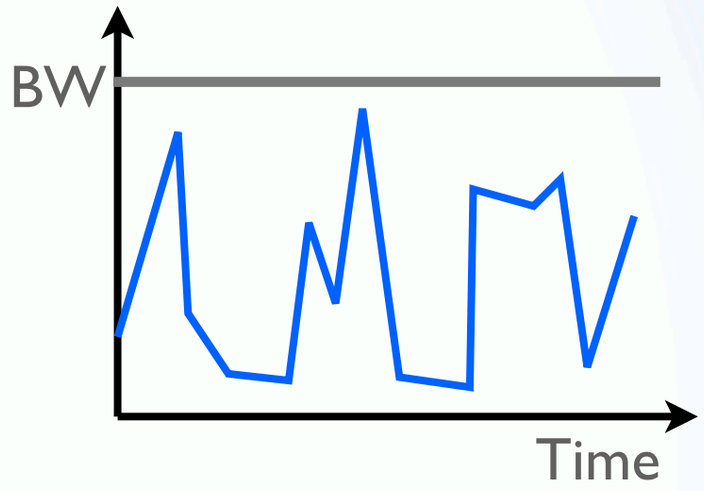
- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic
- Current solution:
  - Over-provisioned link BW



Average channel utilization: < 5%

Motivation

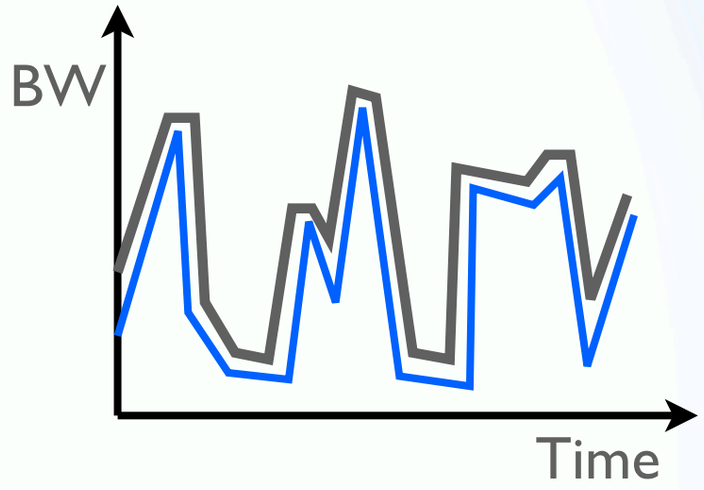
- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic
- Current solution:
  - Over-provisioned link BW
- Our solution:



Average channel utilization: < 5%

Motivation

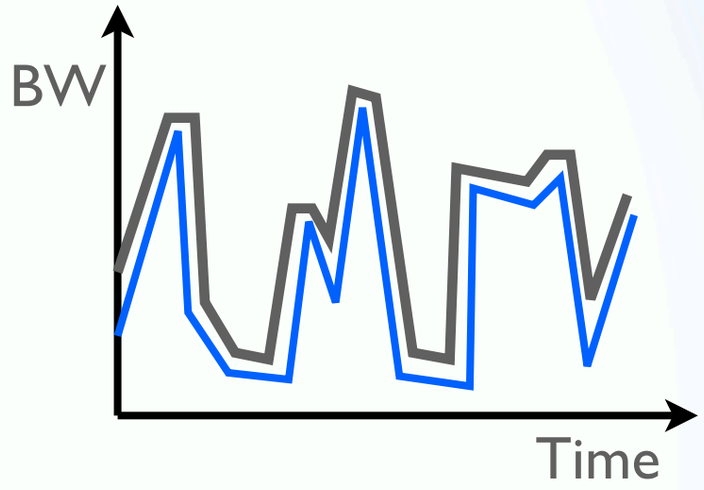
- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic
- Current solution:
  - Over-provisioned link BW
- Our solution:
  - Adapt link BW to demands



Average channel utilization: < 5%

Motivation

- NoCs are crucial for scaling CMPs
- Problem:
  - NoC bandwidth resources are static
  - Bandwidth requirements are highly dynamic
- Current solution:
  - Over-provisioned link BW
- Our solution:
  - Adapt link BW to demands

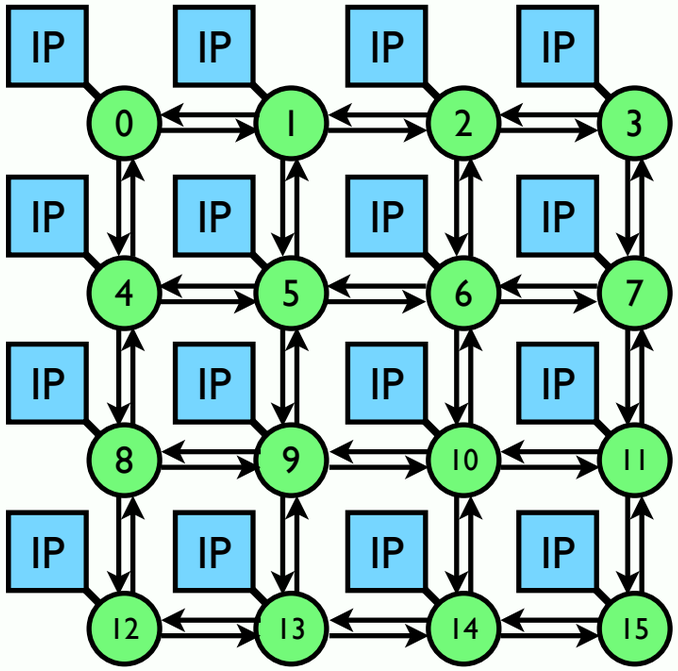


Average channel utilization: < 5%

Save up to 75% of BW resources

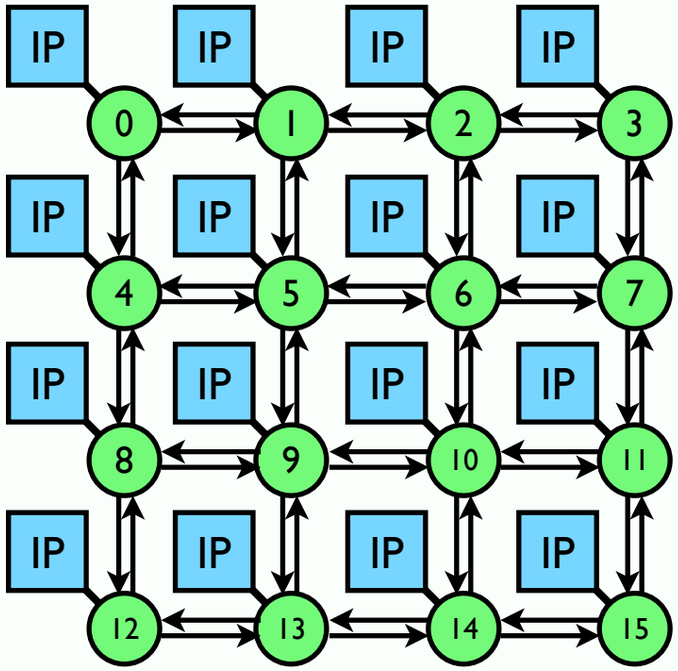
Motivation – Static NoC

Motivation – Static NoC



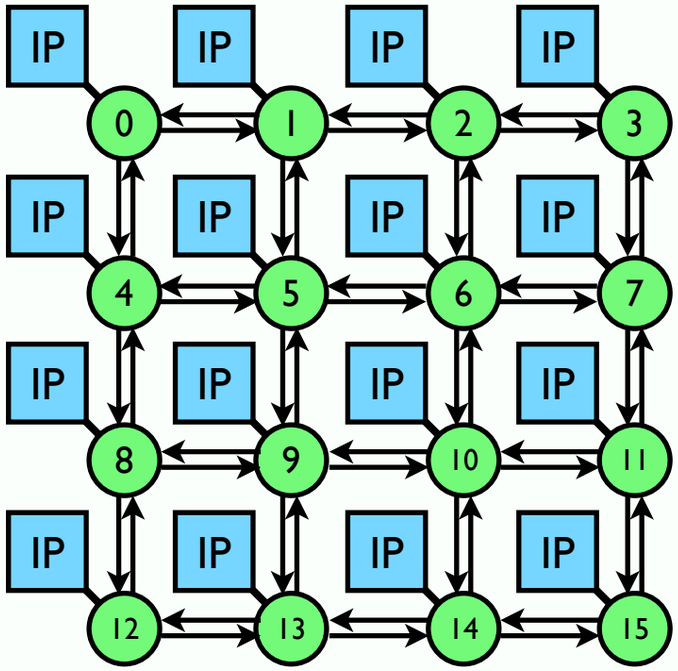
- Static Topology

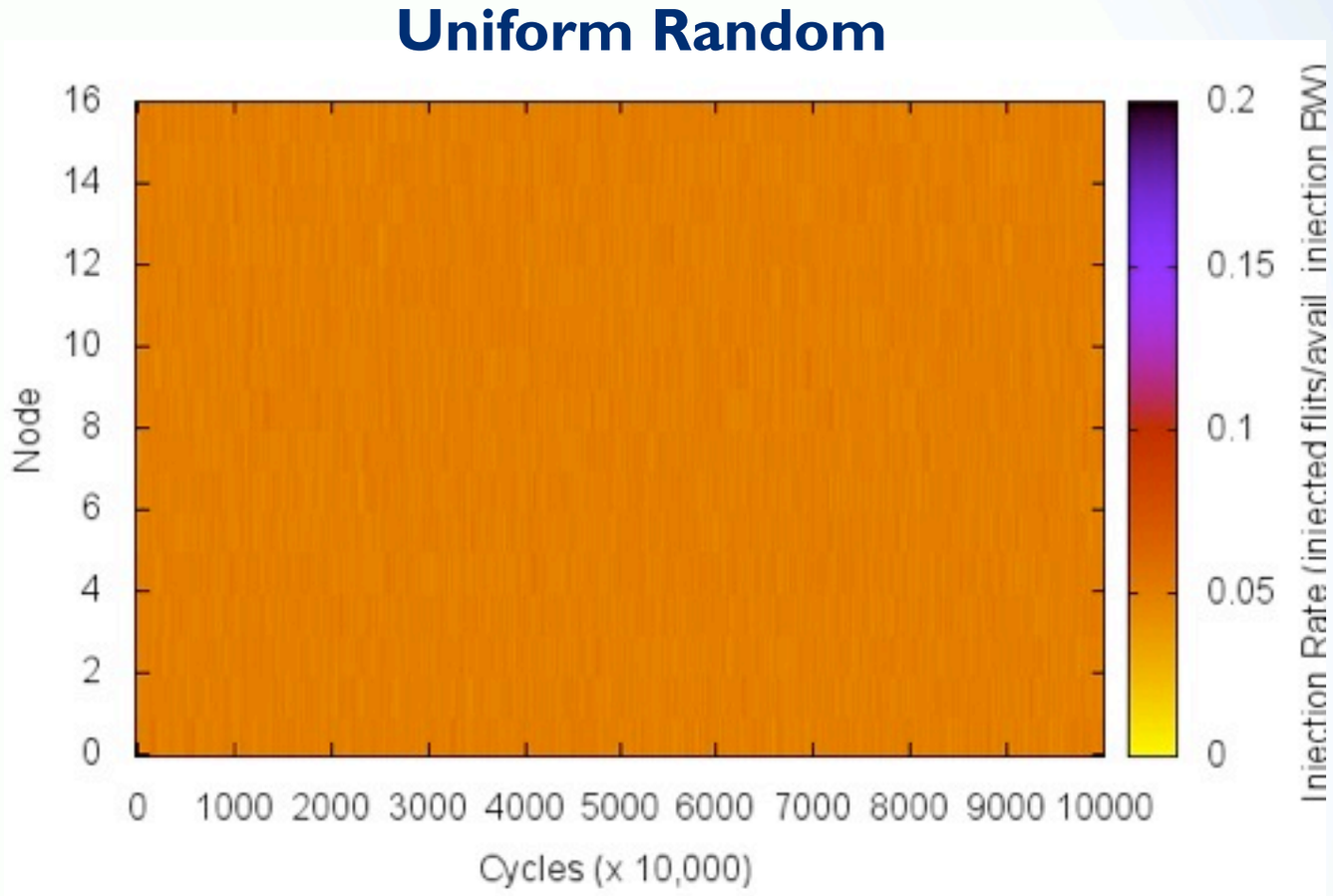
Motivation – Static NoC



- Static Topology
- Static Bandwidth

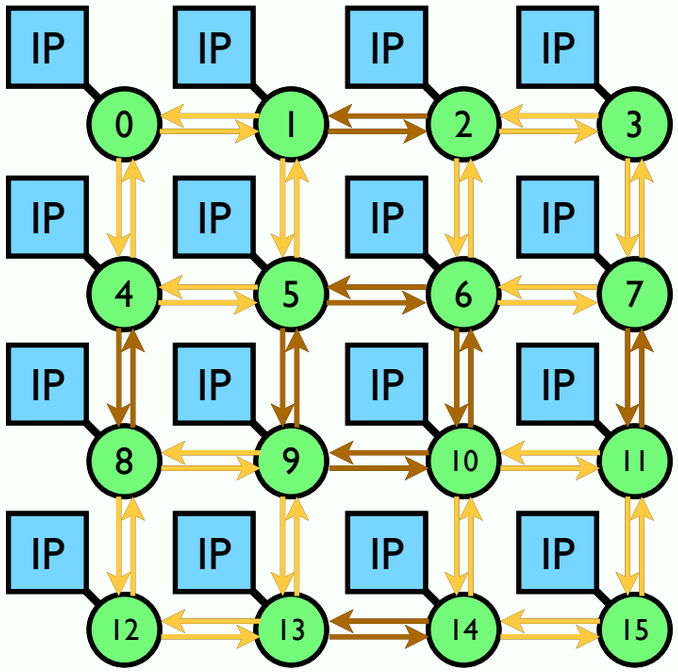
Motivation – Static NoC

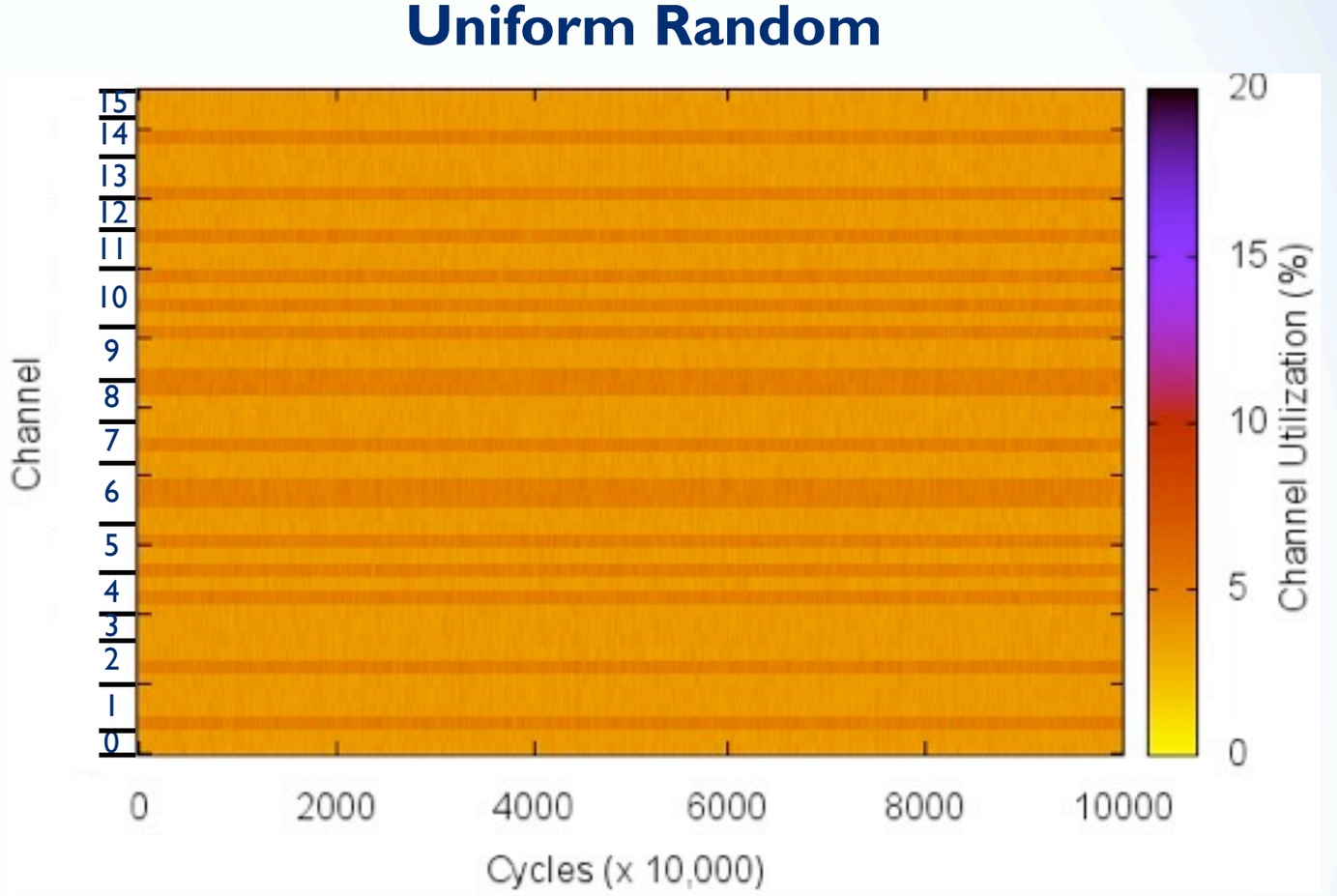




- Static Topology
- Static Bandwidth
- Static workloads for evaluation

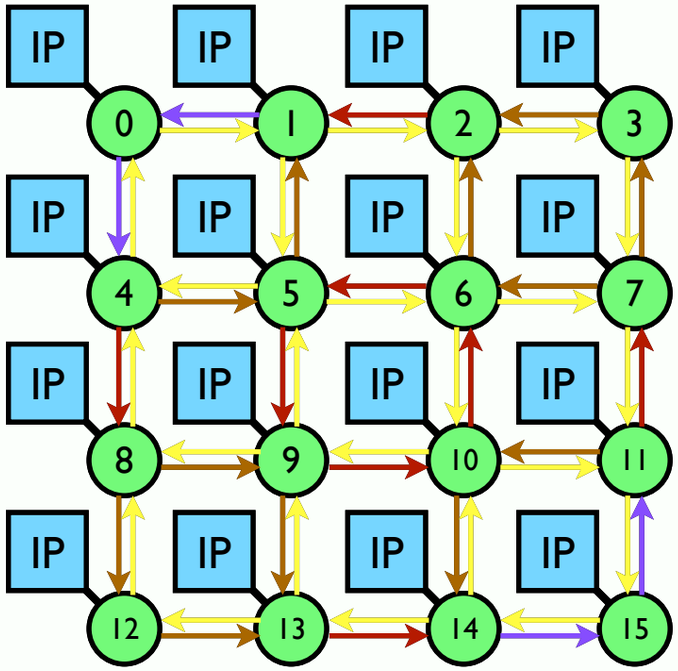
Motivation – Static NoC

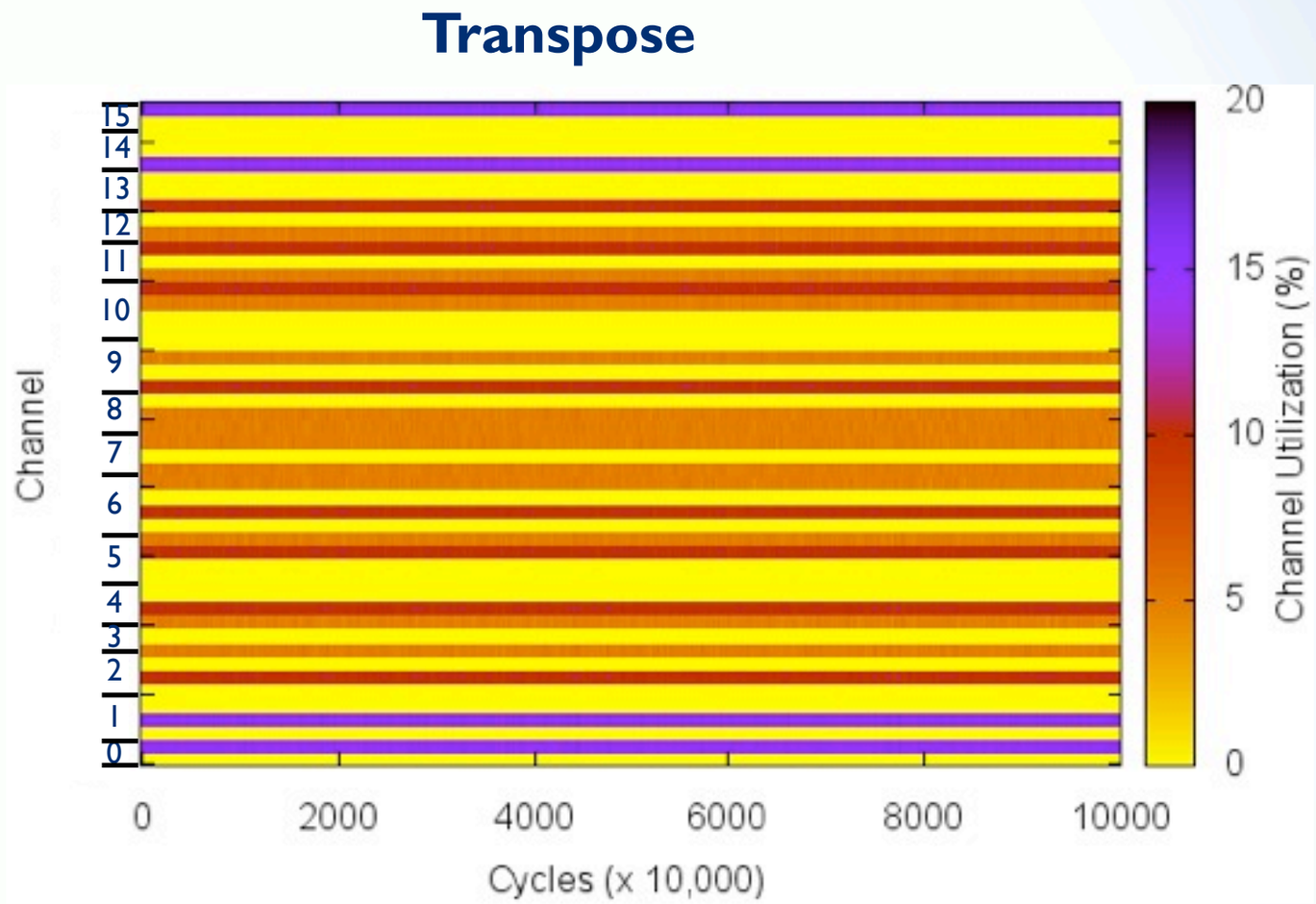




- Static Topology
- Static Bandwidth
- Static workloads for evaluation

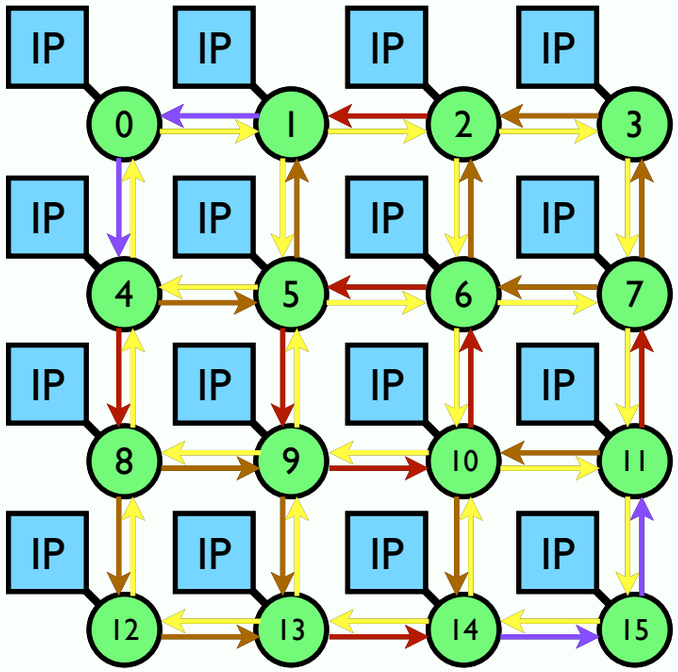
Motivation – Static NoC

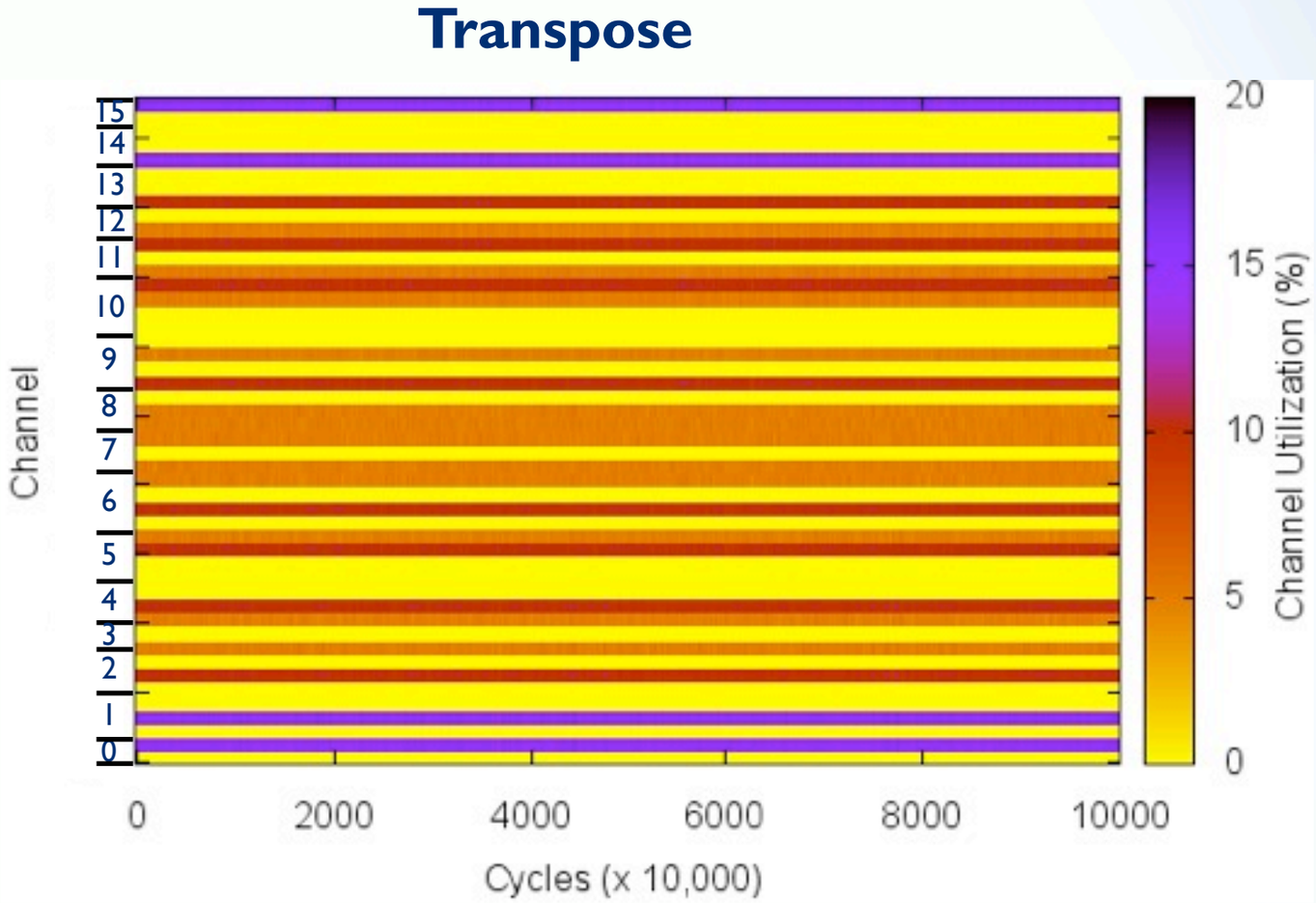




- Static Topology
- Static Bandwidth
- Static workloads for evaluation

Motivation – Static NoC

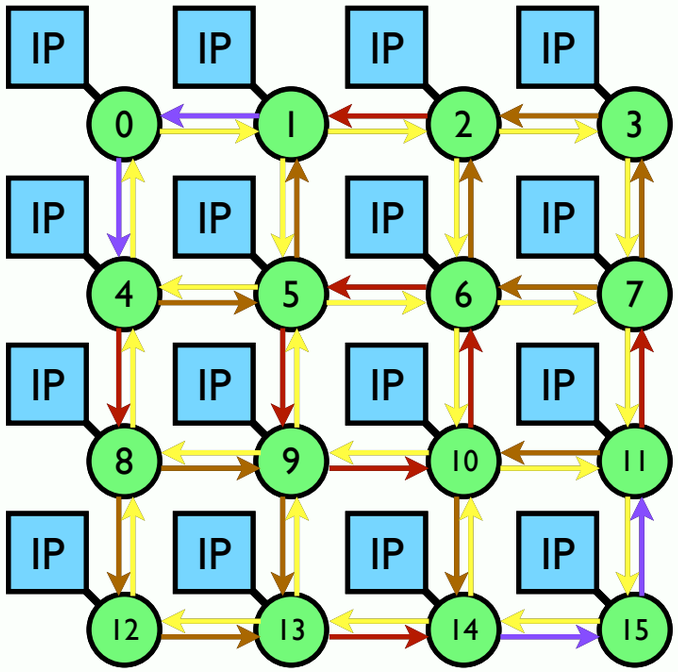


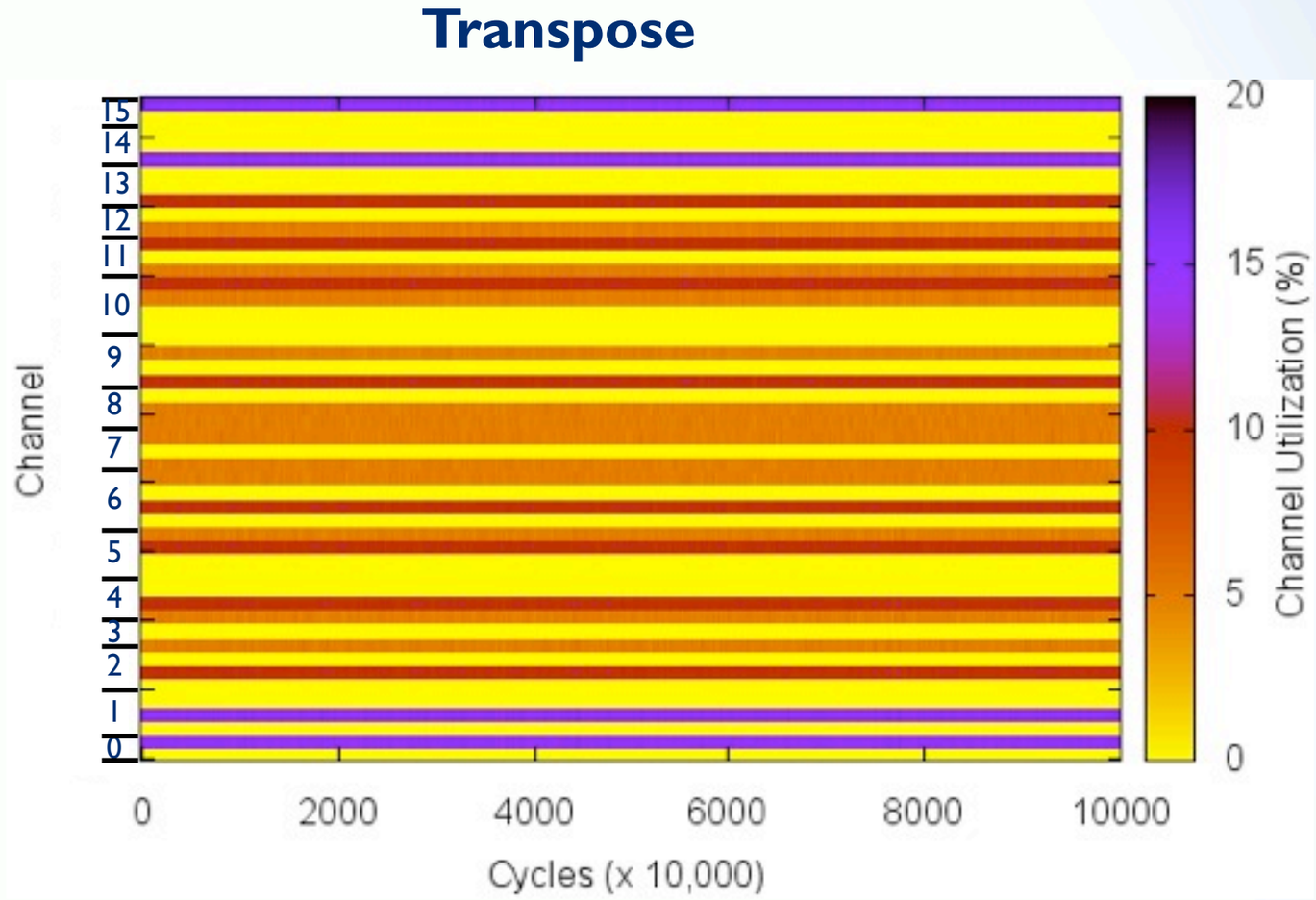


- Static Topology
- Static Bandwidth
- Static workloads for evaluation

- Specified at design time for worst case scenario

Motivation – Static NoC

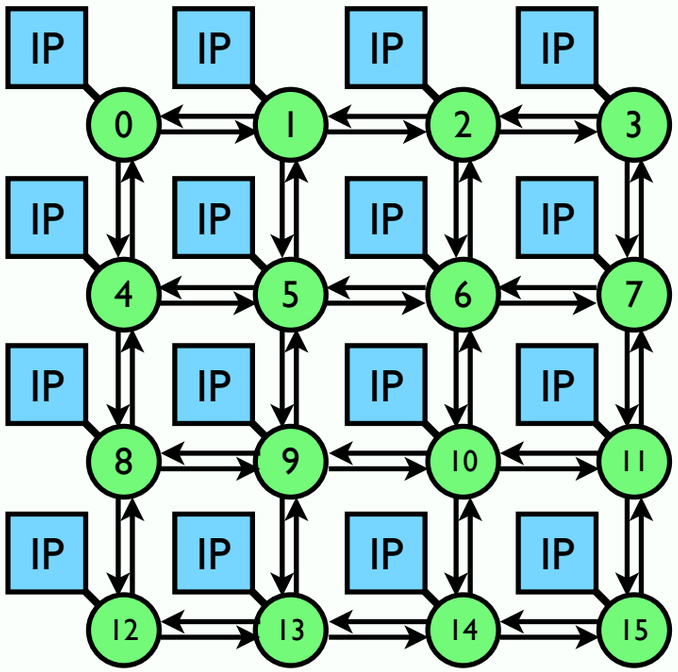




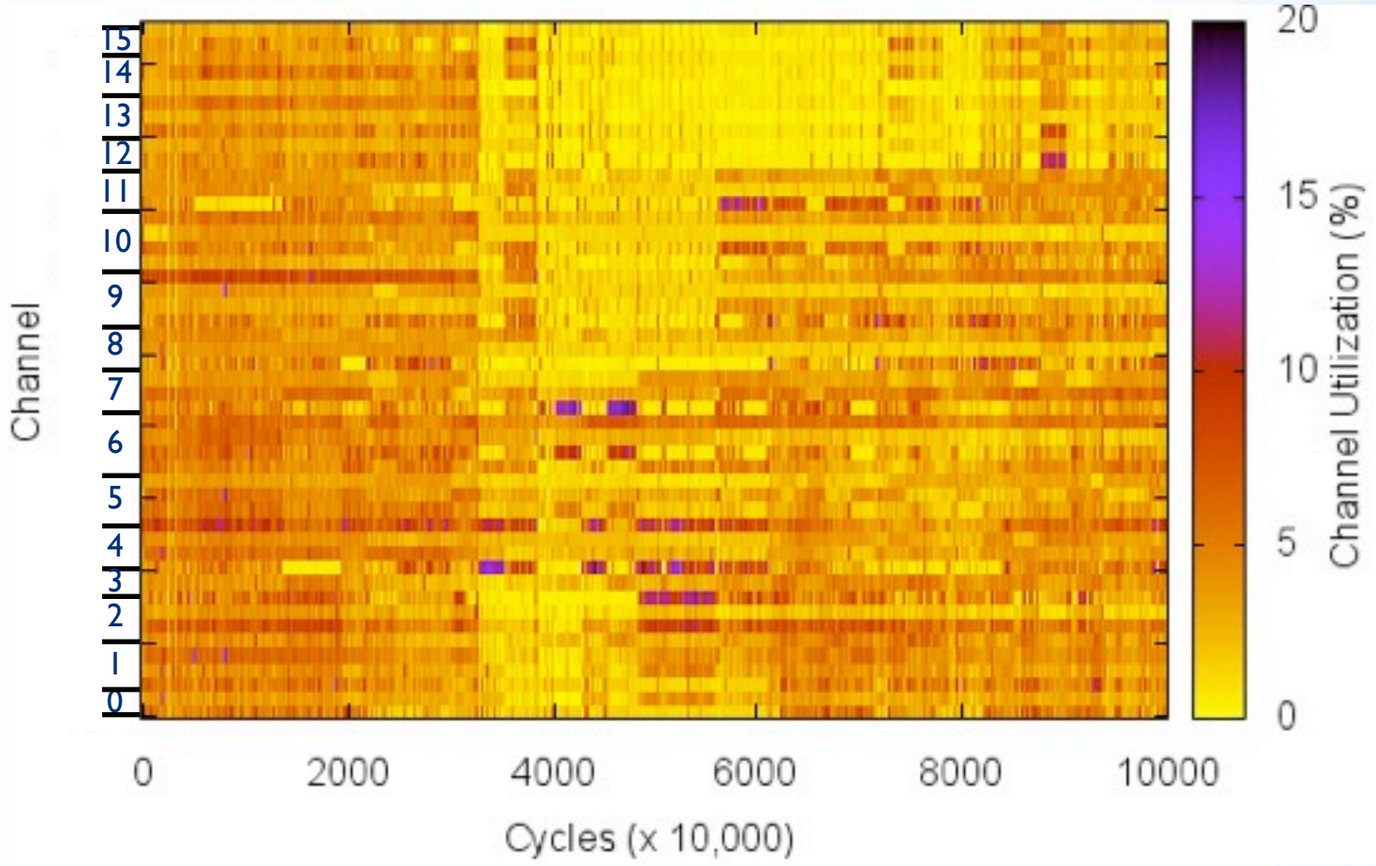
- Static Topology
- Static Bandwidth
- Static workloads for evaluation

- Specified at design time for worst case scenario
- Static NoCs can handle temporally- and spatially-stable traffic well

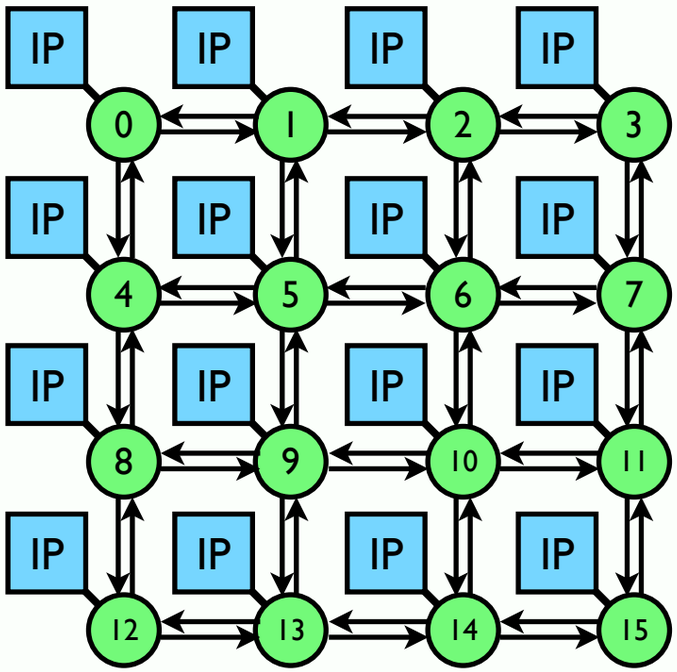
Motivation – Real NoC Traffic

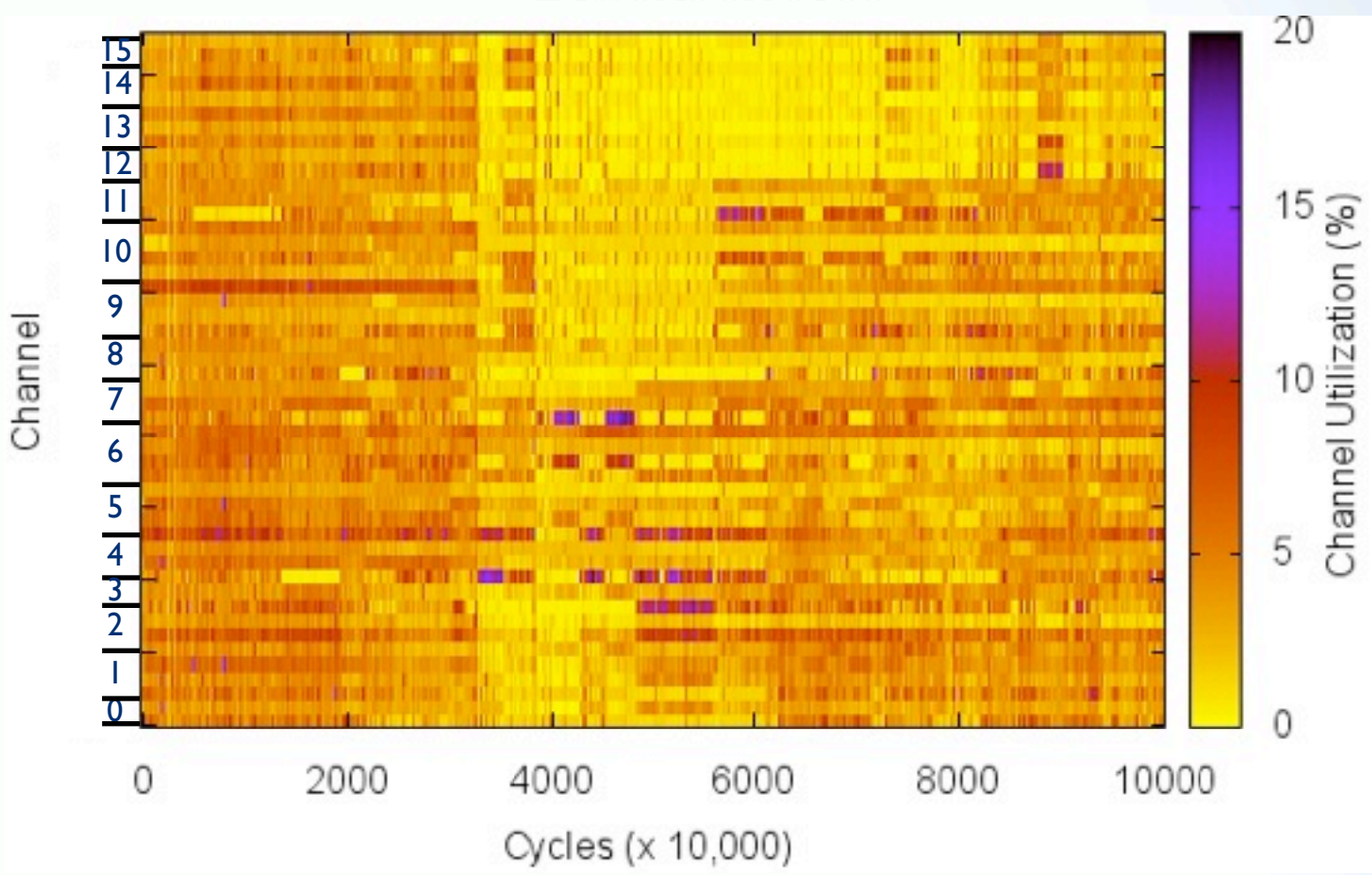


Blackscholes



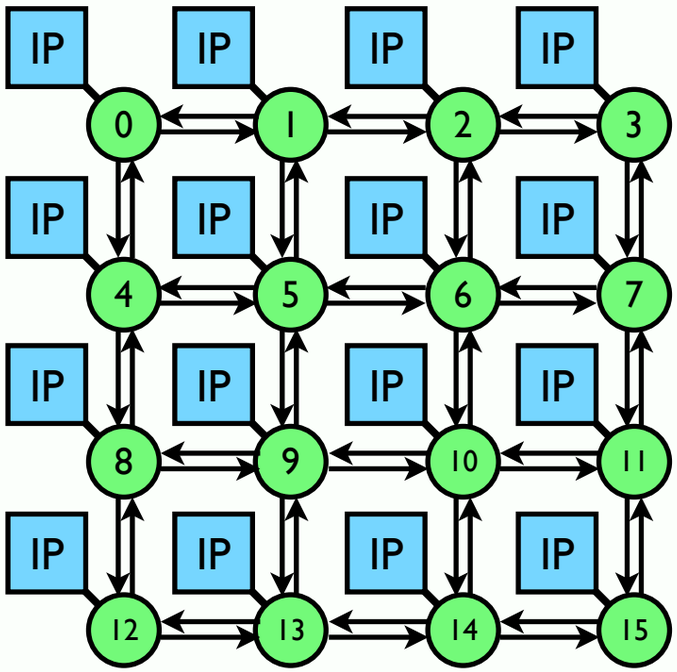
Motivation – Real NoC Traffic

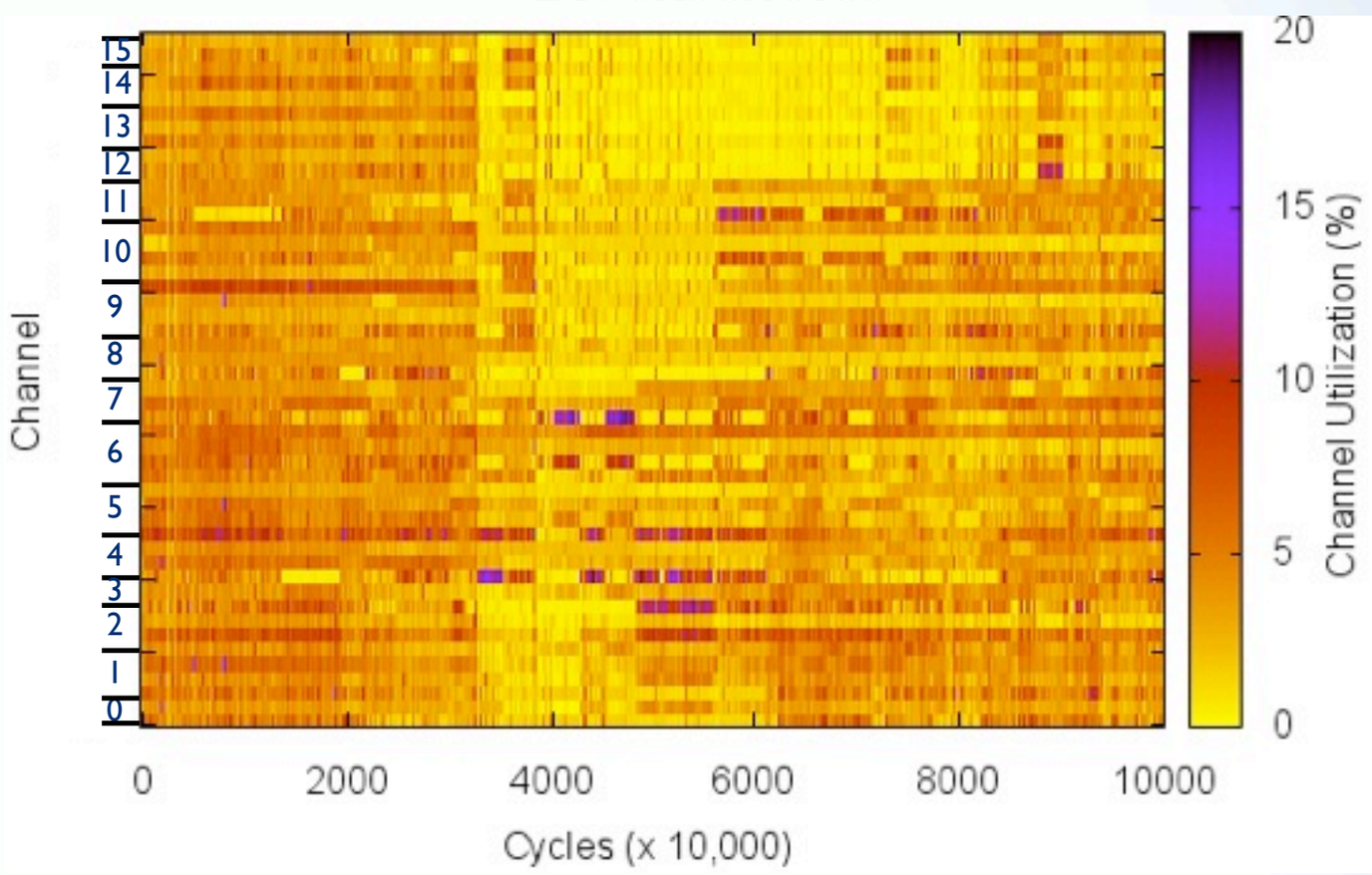




- Highly dynamic workloads

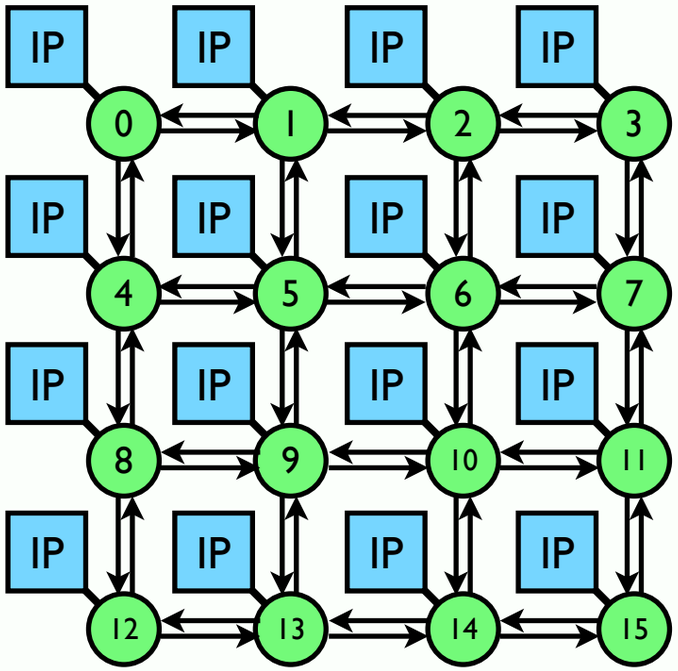
Motivation – Real NoC Traffic

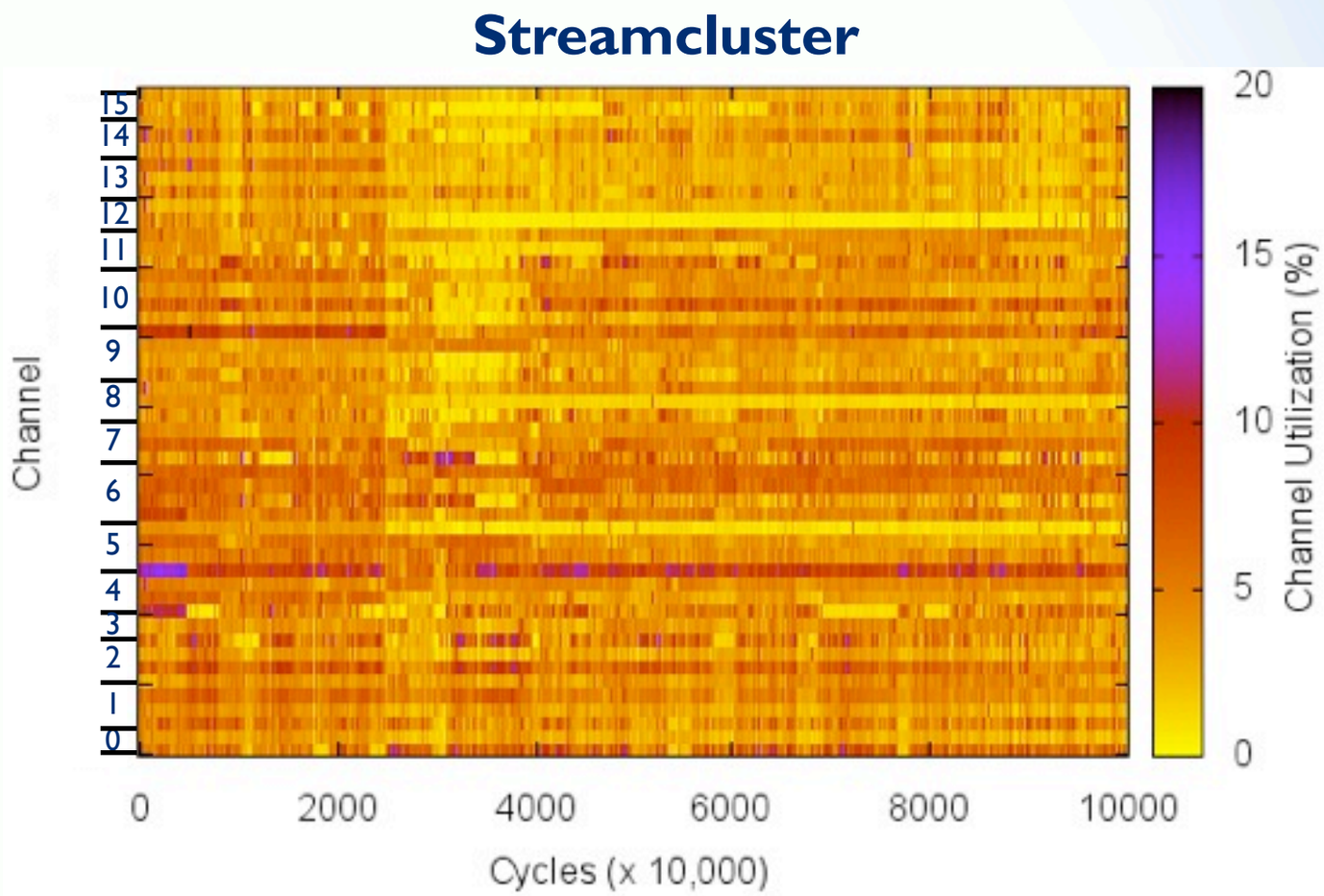




- Highly dynamic workloads
- Large temporal and spatial BW variance

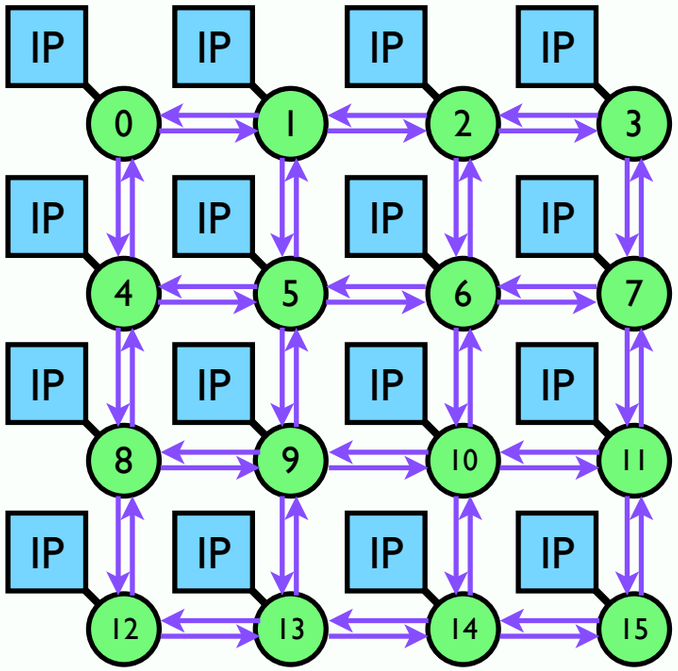
Motivation – Real NoC Traffic

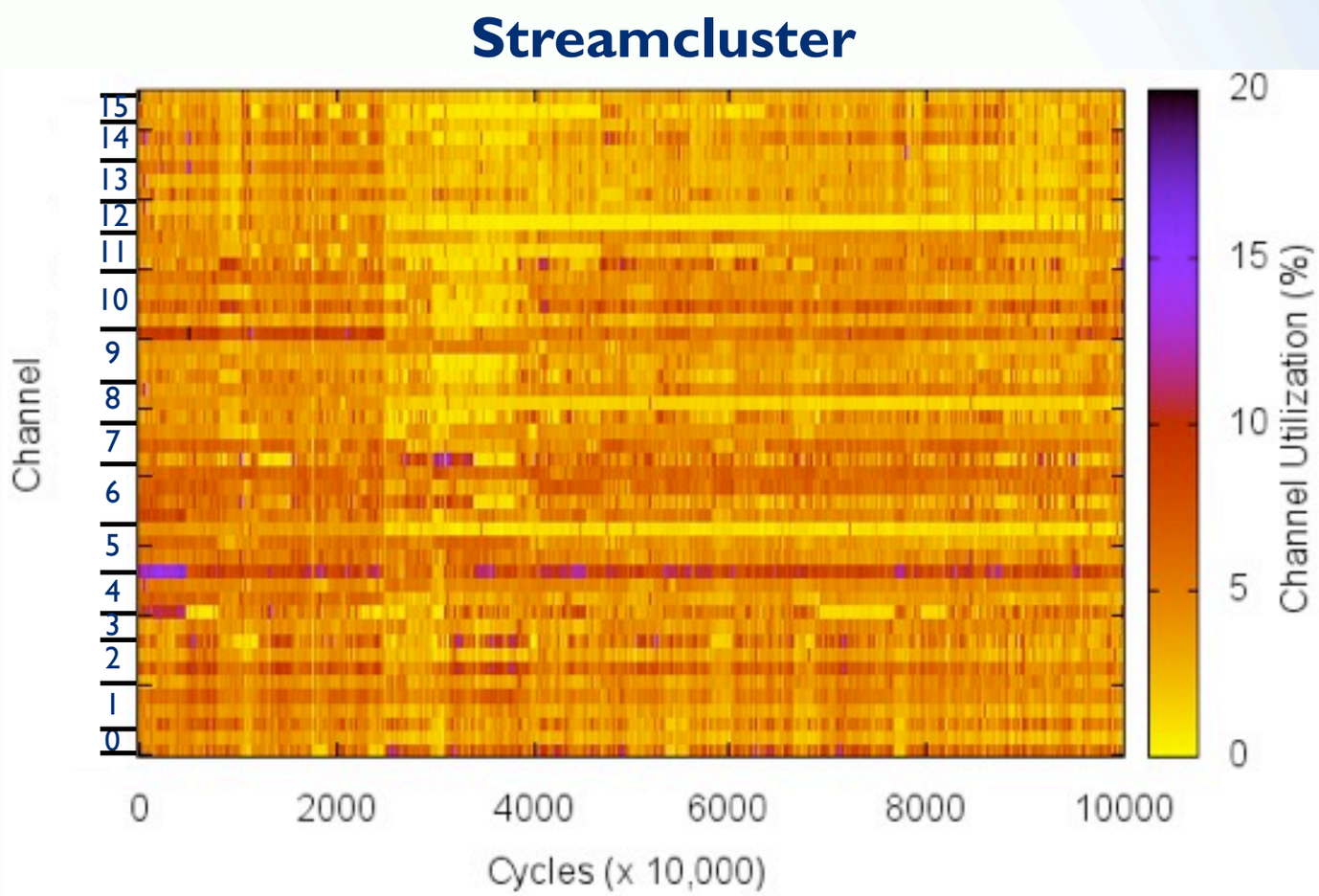




- Highly dynamic workloads
- Large temporal and spatial BW variance

Motivation – Real NoC Traffic

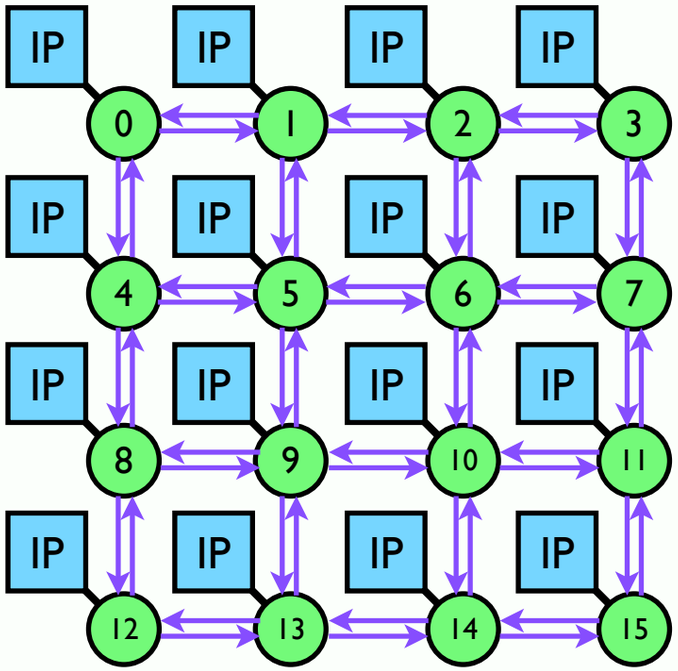


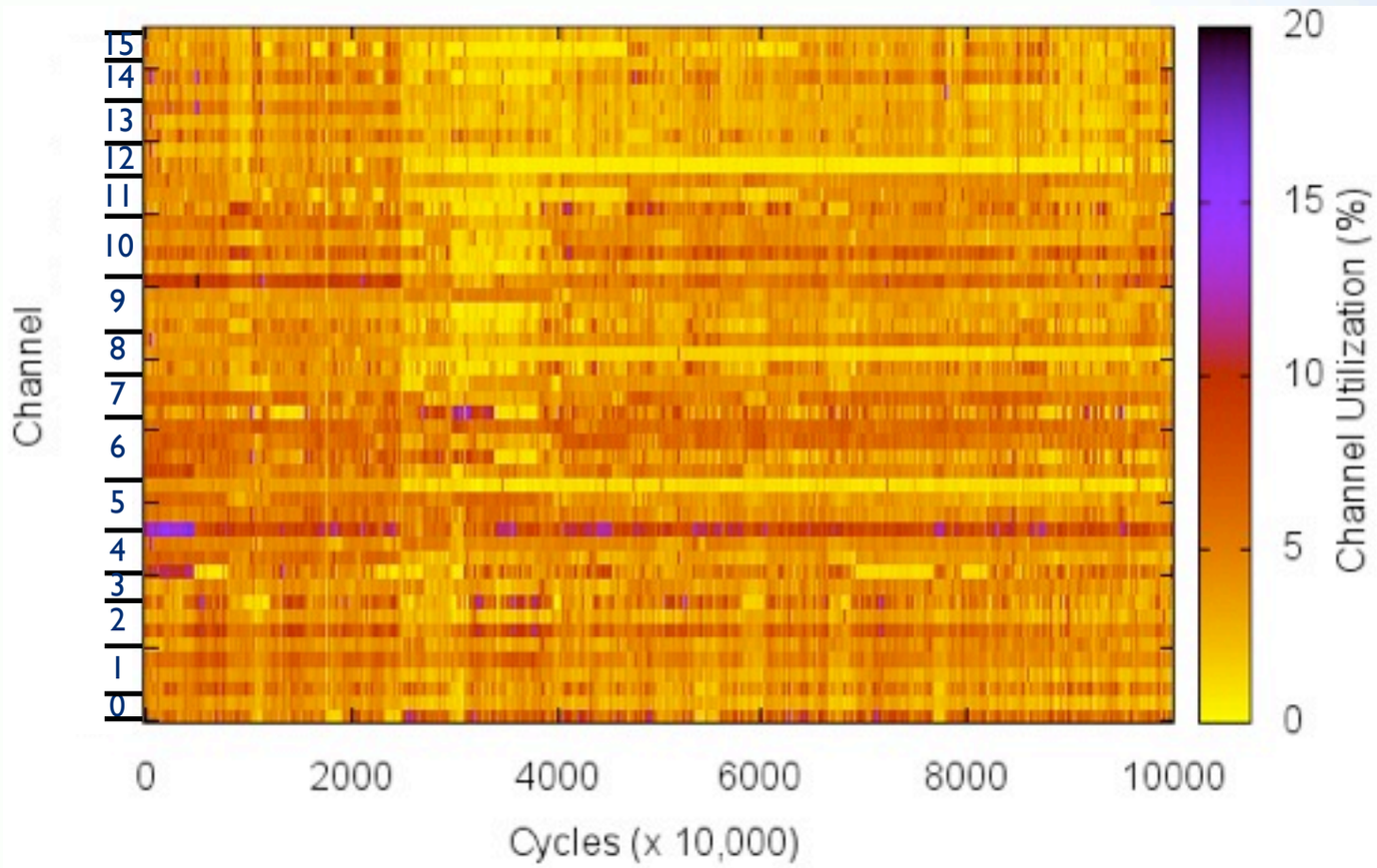


- Highly dynamic workloads
- Large temporal and spatial BW variance

- Significant area and power overhead with traditional NoC implementation

Motivation – Real NoC Traffic



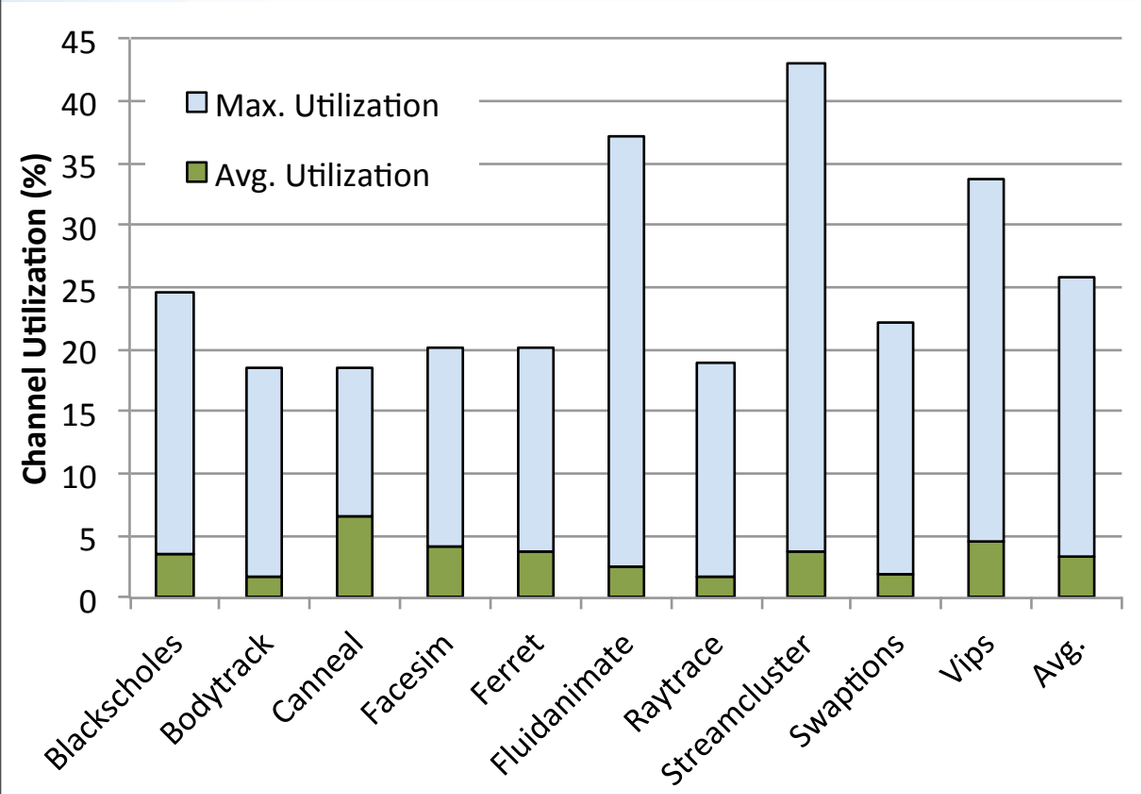


- Highly dynamic workloads
- Large temporal and spatial BW variance

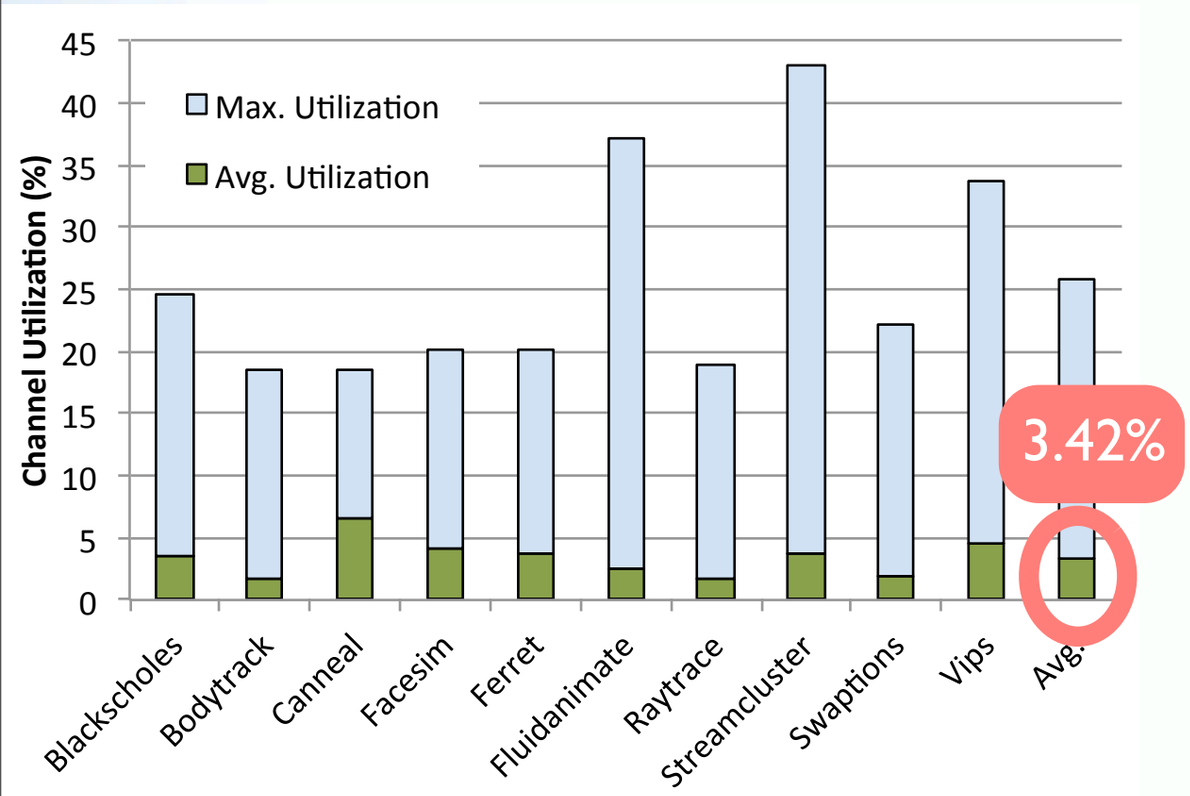
- Significant area and power overhead with traditional NoC implementation
- Channels are underutilized most of the time

Channel Utilization

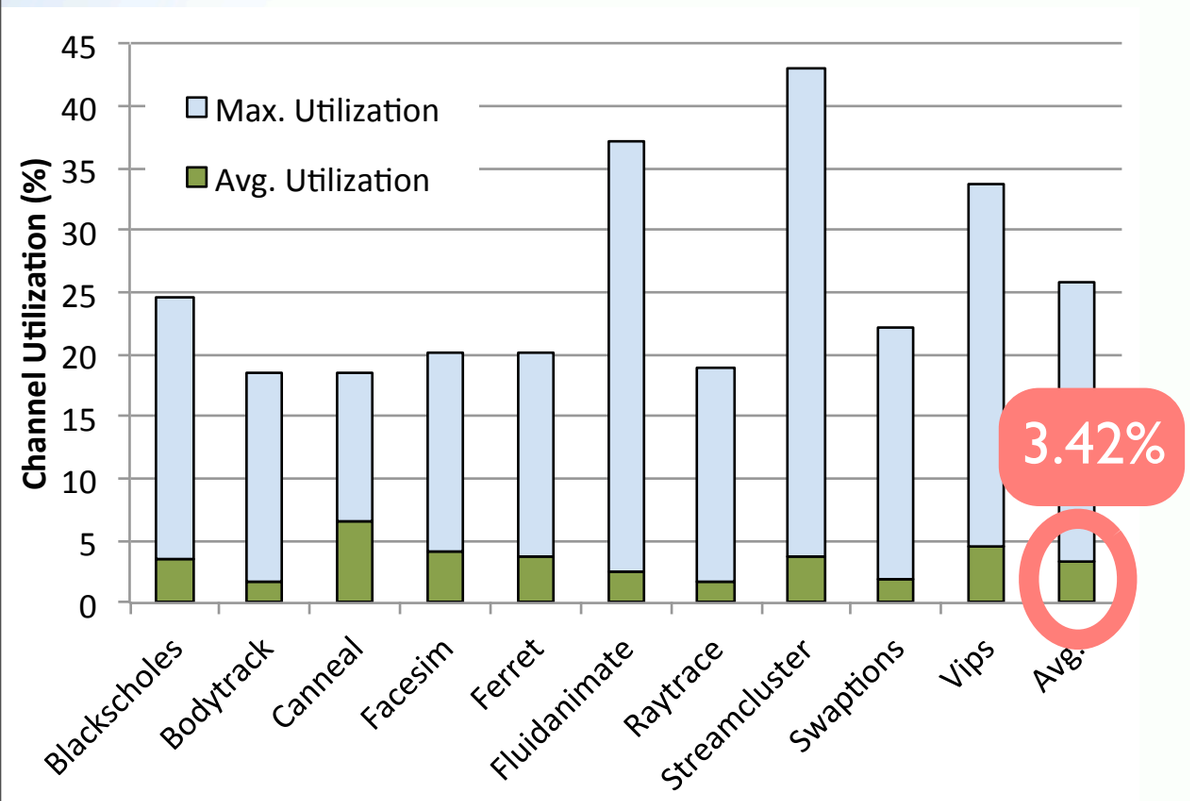
Channel Utilization



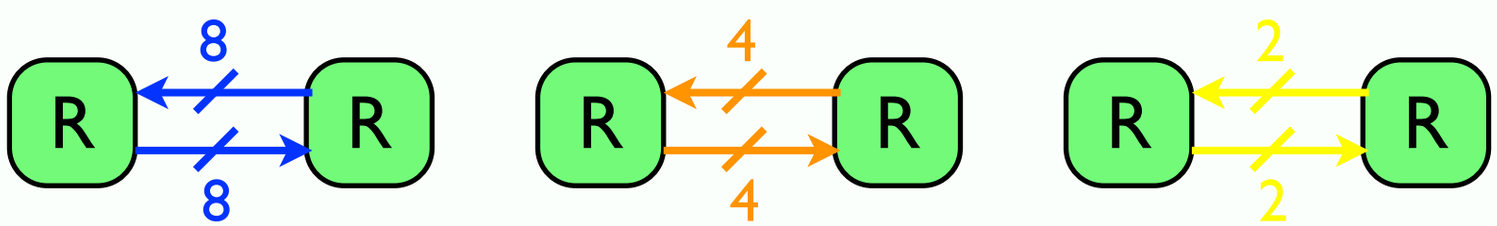
Channel Utilization



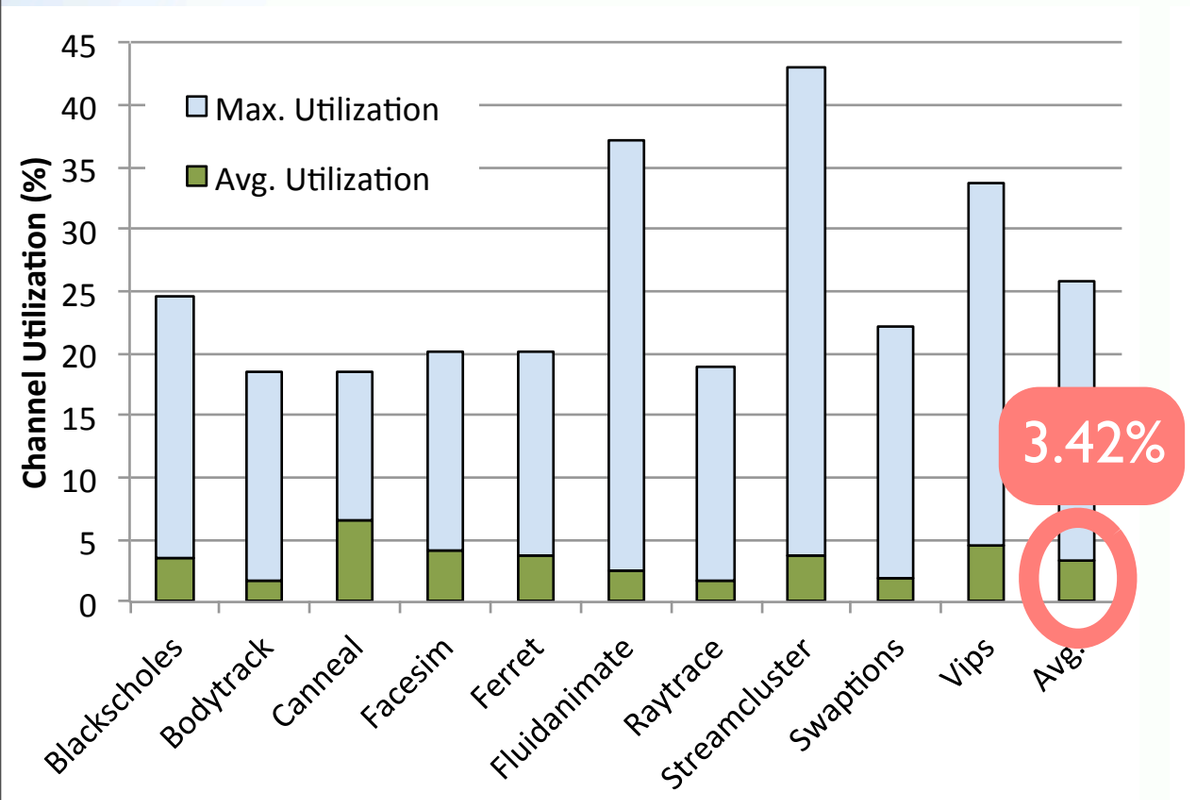
Channel Utilization

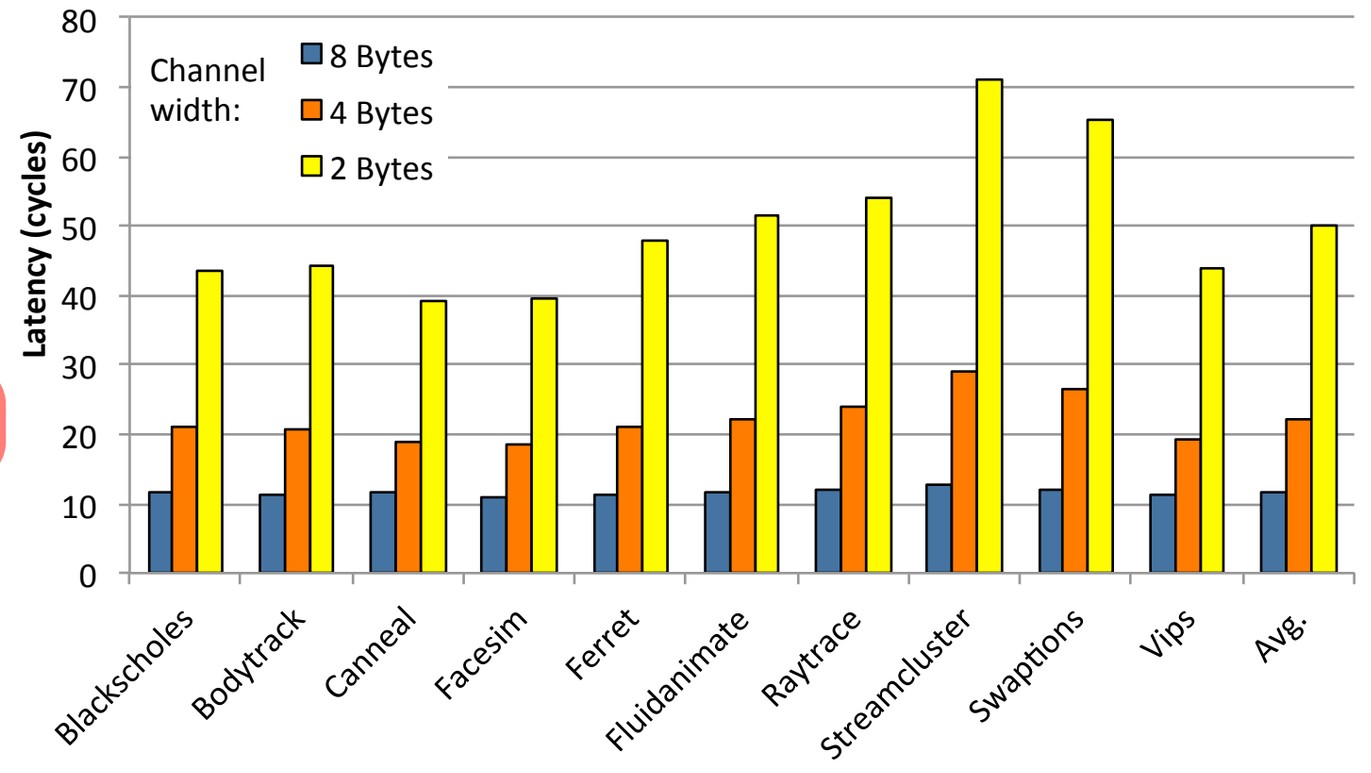


- Adjust channel width (flit width):

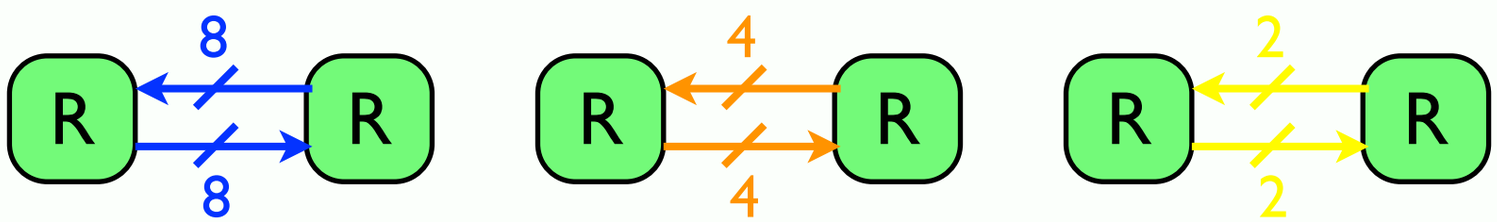


Channel Utilization

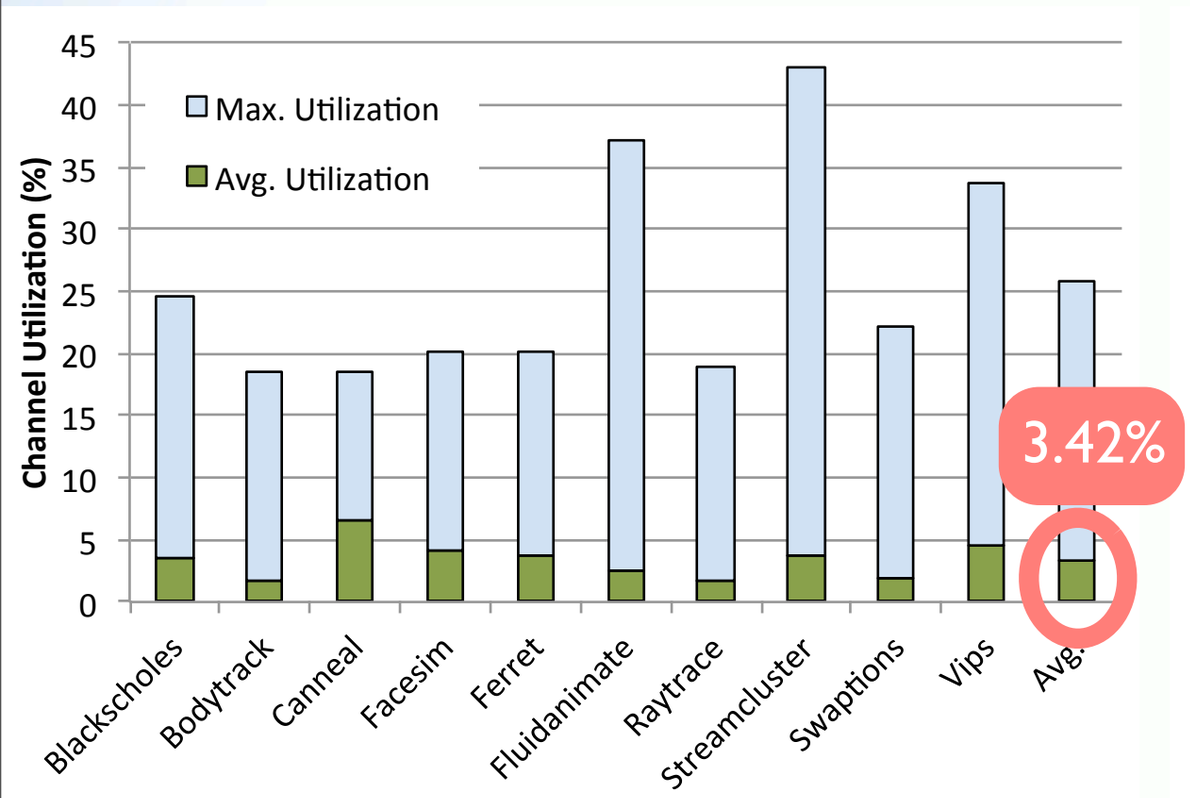


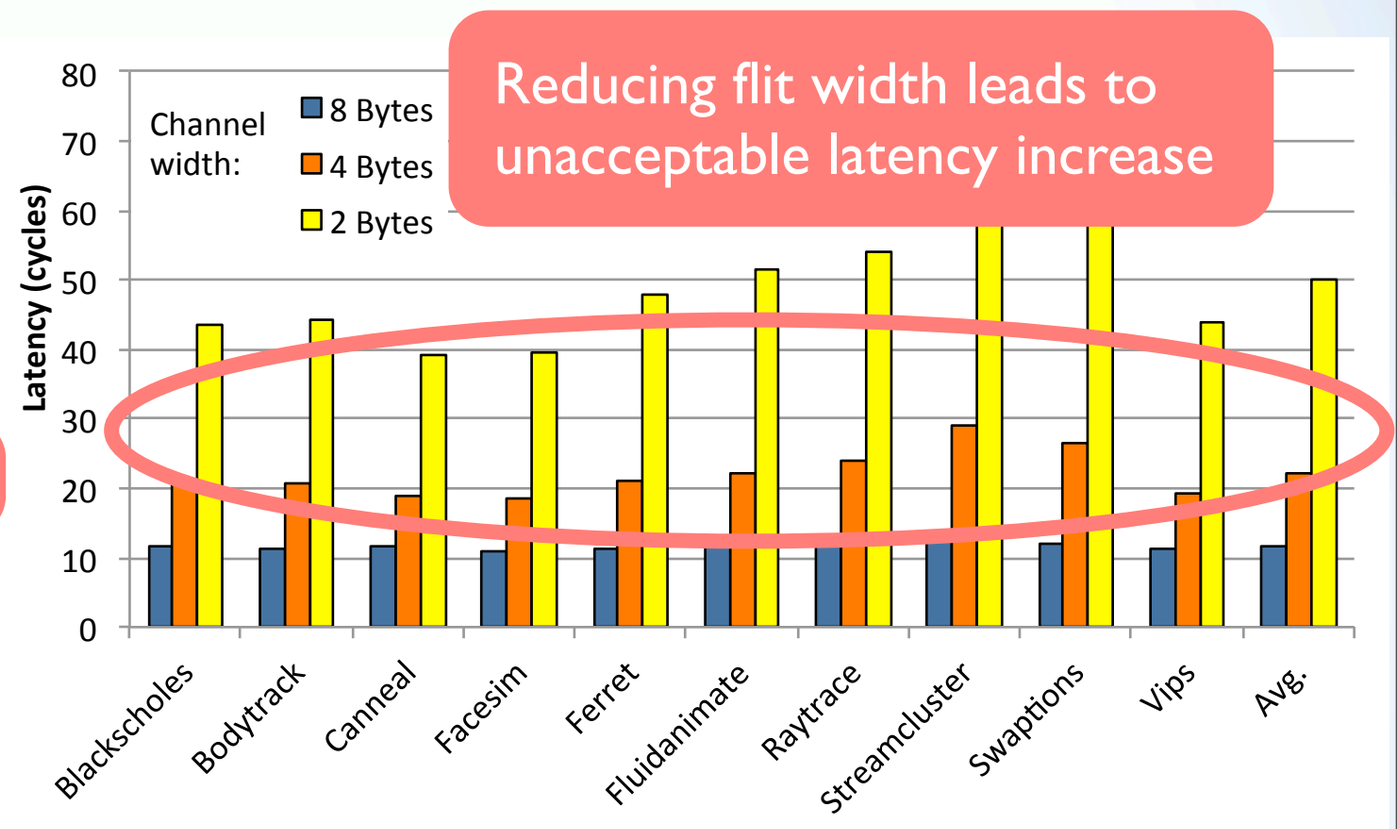


- Adjust channel width (flit width):

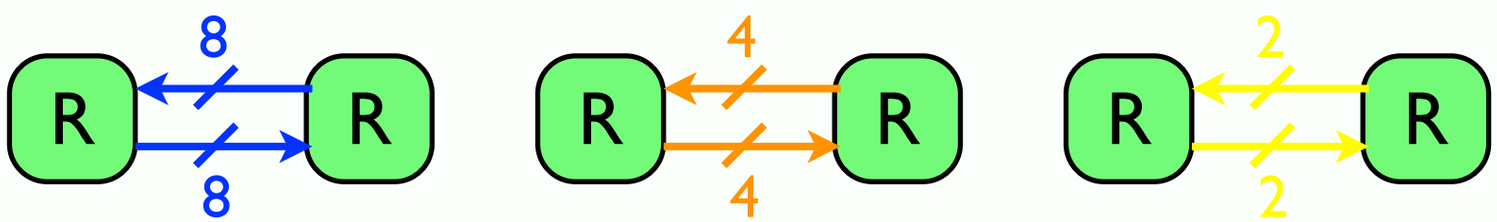


Channel Utilization



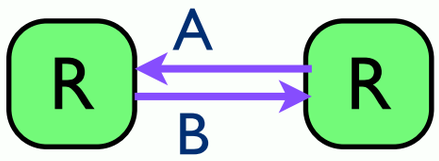


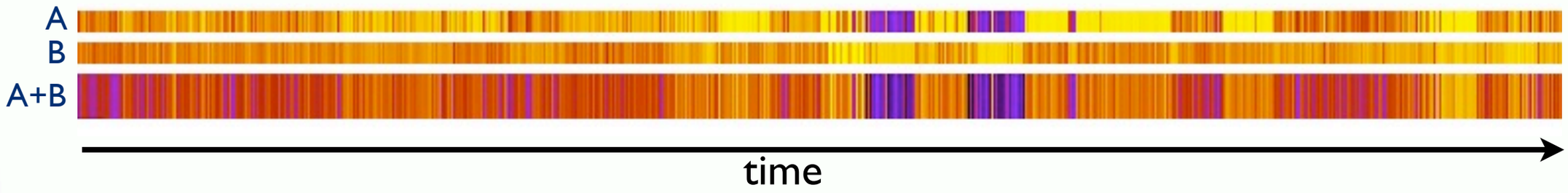
- Adjust channel width (flit width):



Bidirectional Channels

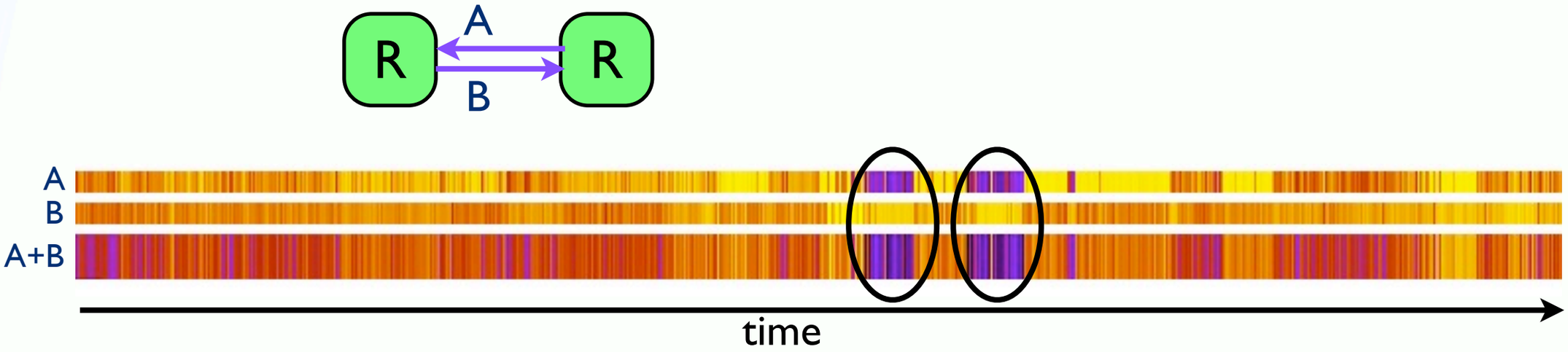
- Bidirectional channels to share channel resources:





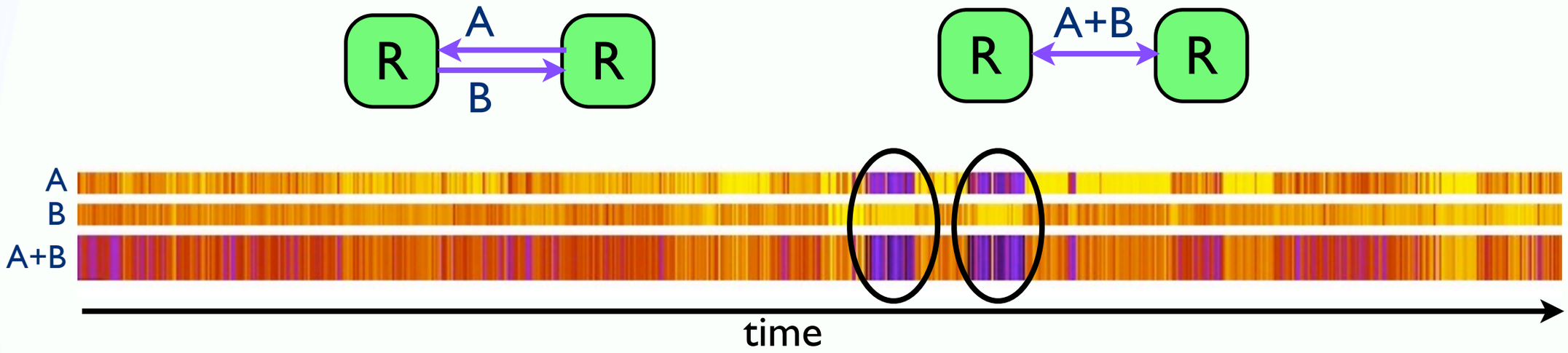
Bidirectional Channels

- Bidirectional channels to share channel resources:



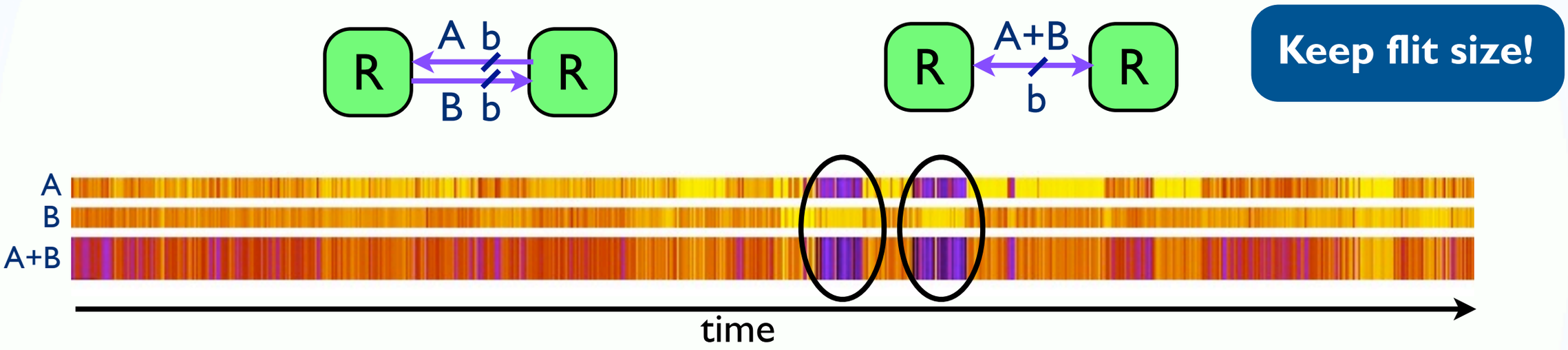
Bidirectional Channels

- Bidirectional channels to share channel resources:



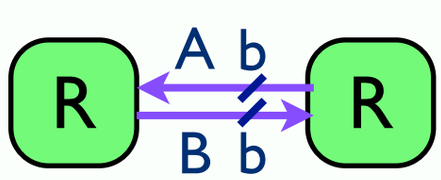
Bidirectional Channels

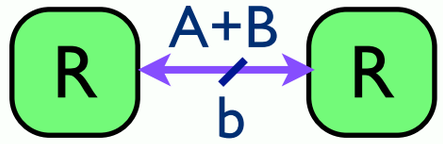
- Bidirectional channels to share channel resources:



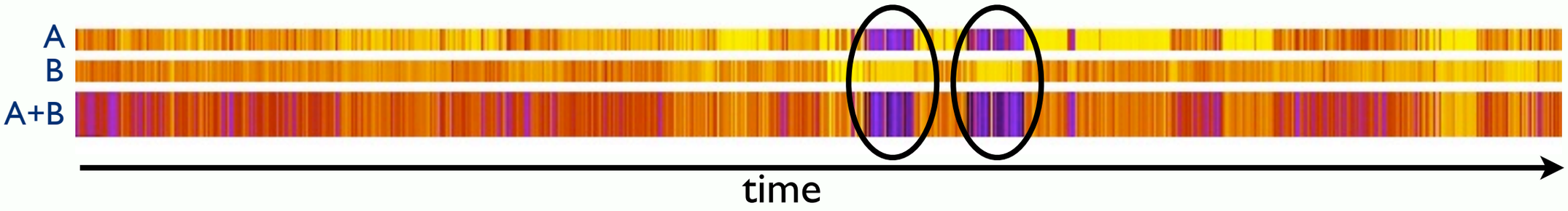
Bidirectional Channels

- Bidirectional channels to share channel resources:

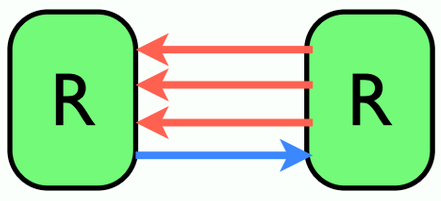


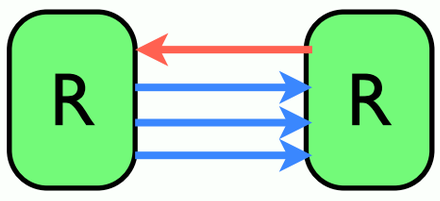


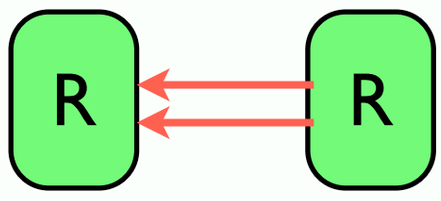
Keep flit size!



- Adding flexibility with fine-grained BW adaptivity

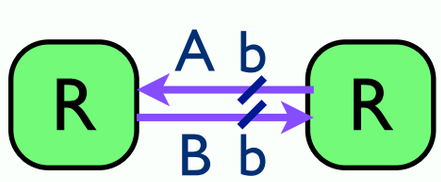


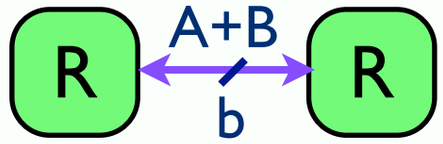




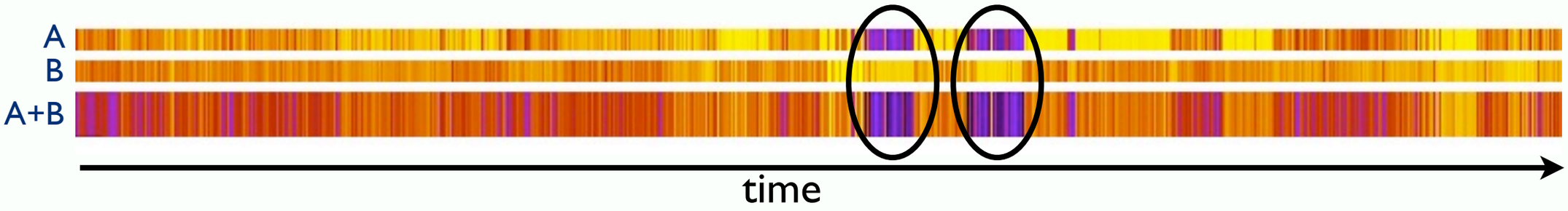
Bidirectional Channels

- Bidirectional channels to share channel resources:

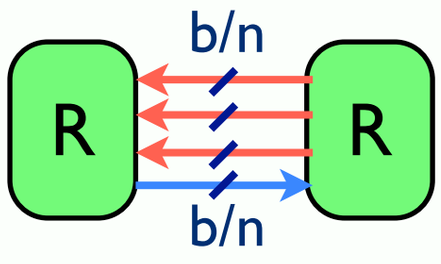


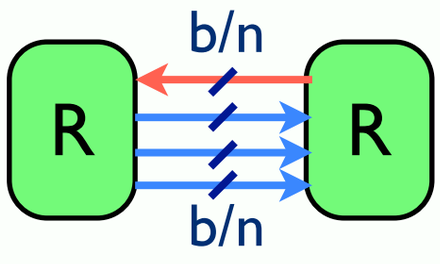


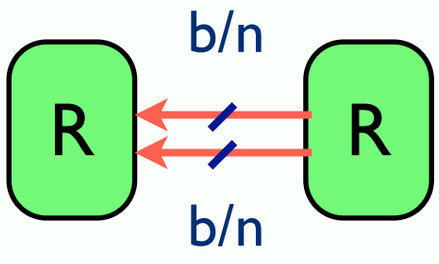
Keep flit size!



- Adding flexibility with fine-grained BW adaptivity



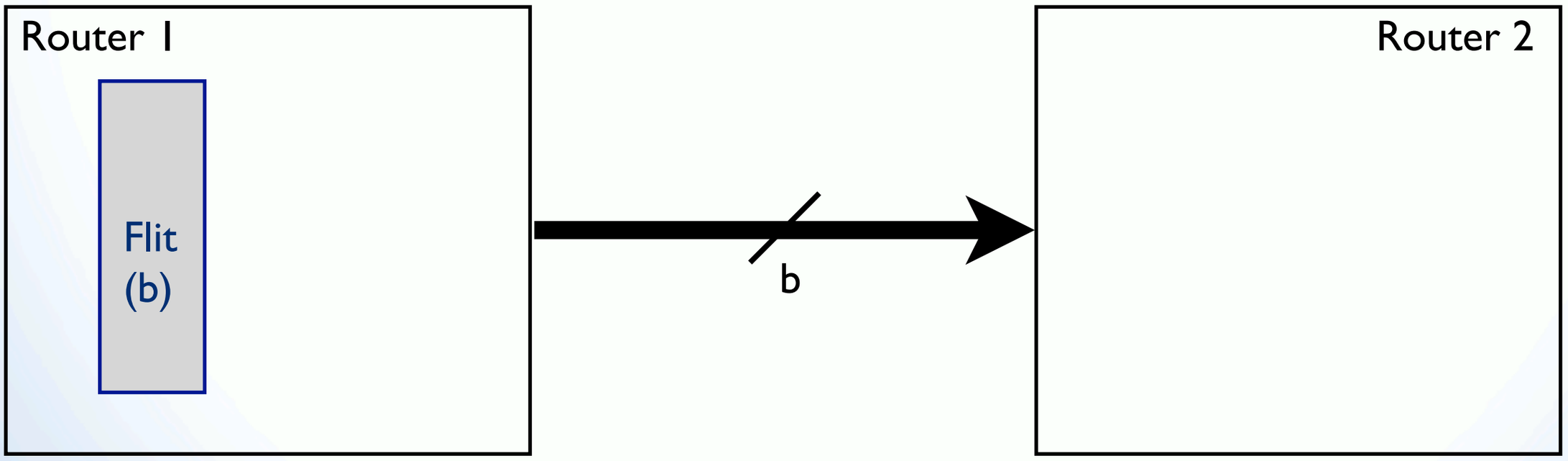




Need to sub-divide flits

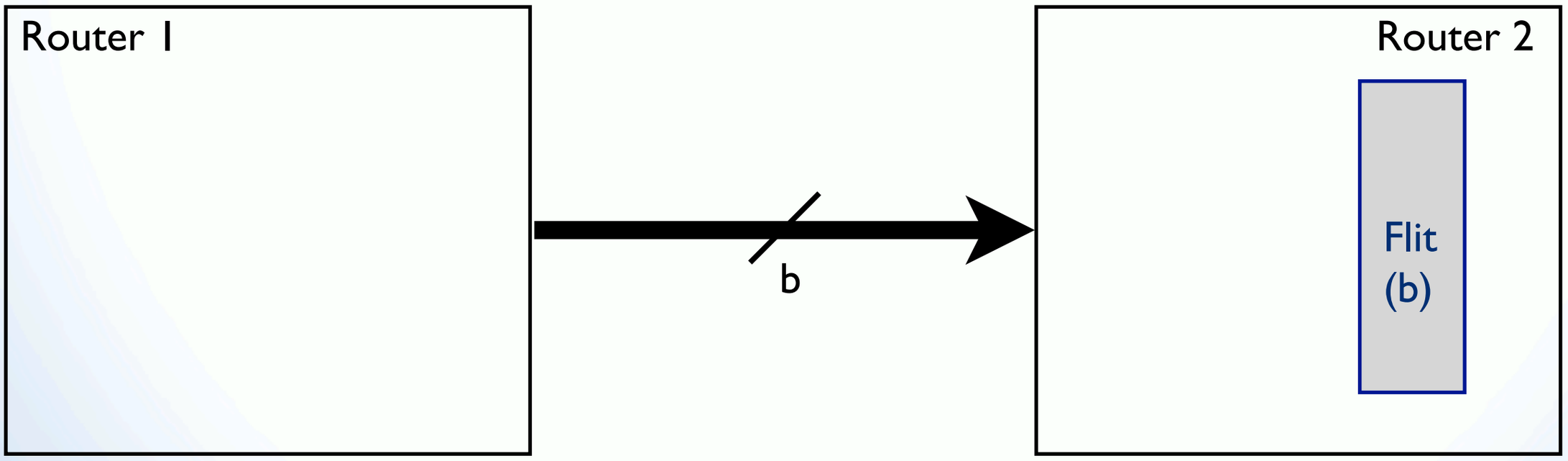
Decoupling Flit Width From Channel Width

- Conventionally in NoC, flit width is coupled to channel width



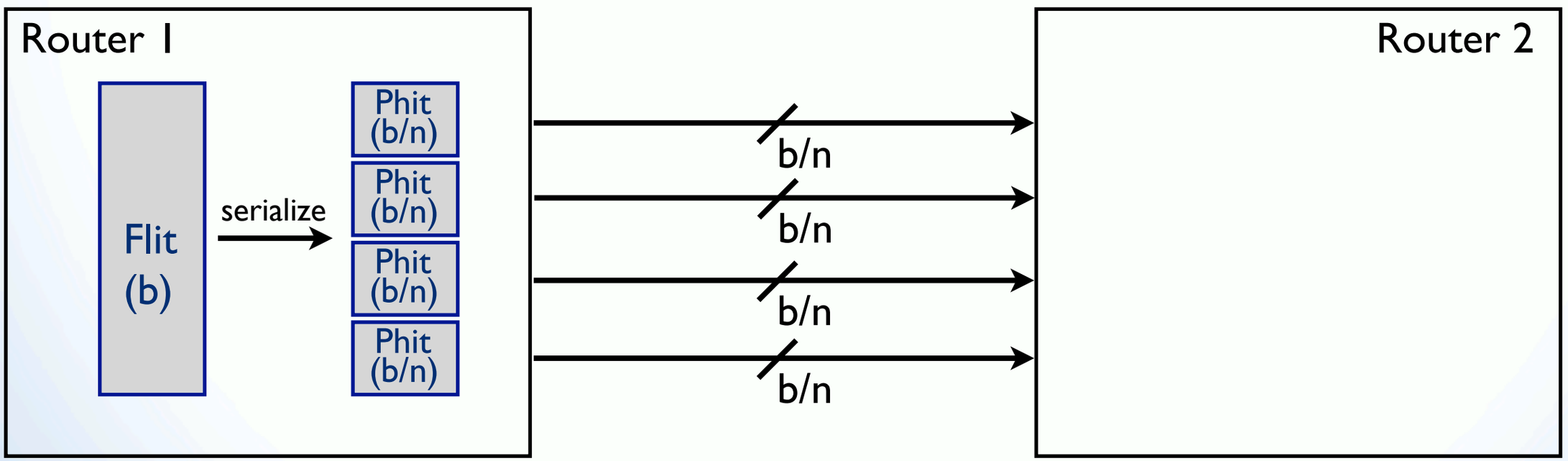
Decoupling Flit Width From Channel Width

- Conventionally in NoC, flit width is coupled to channel width



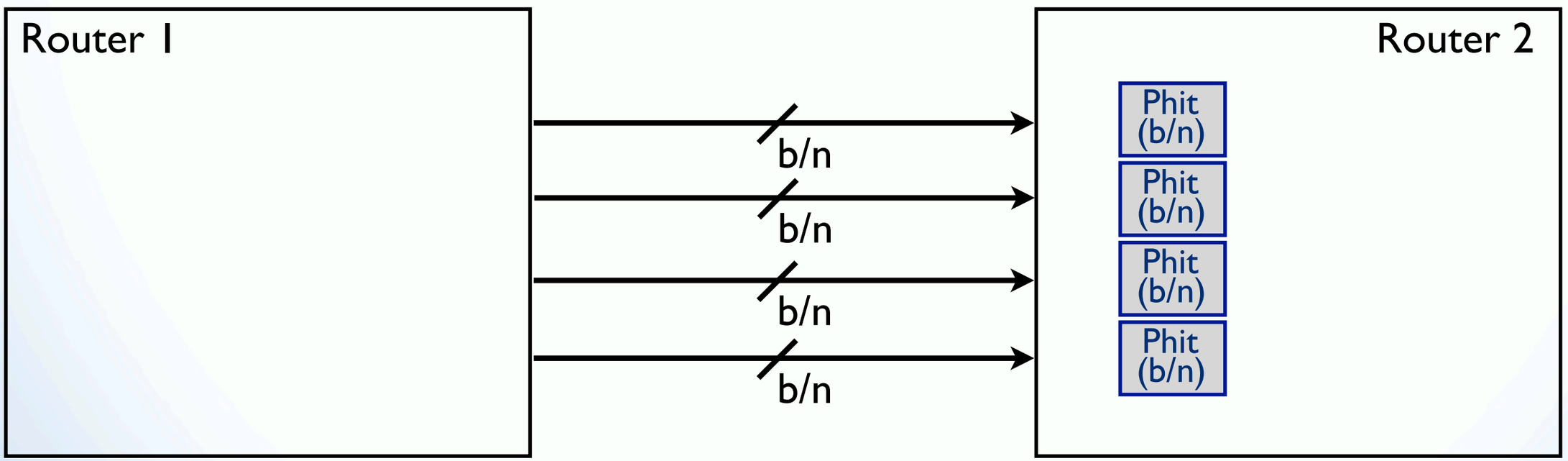
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



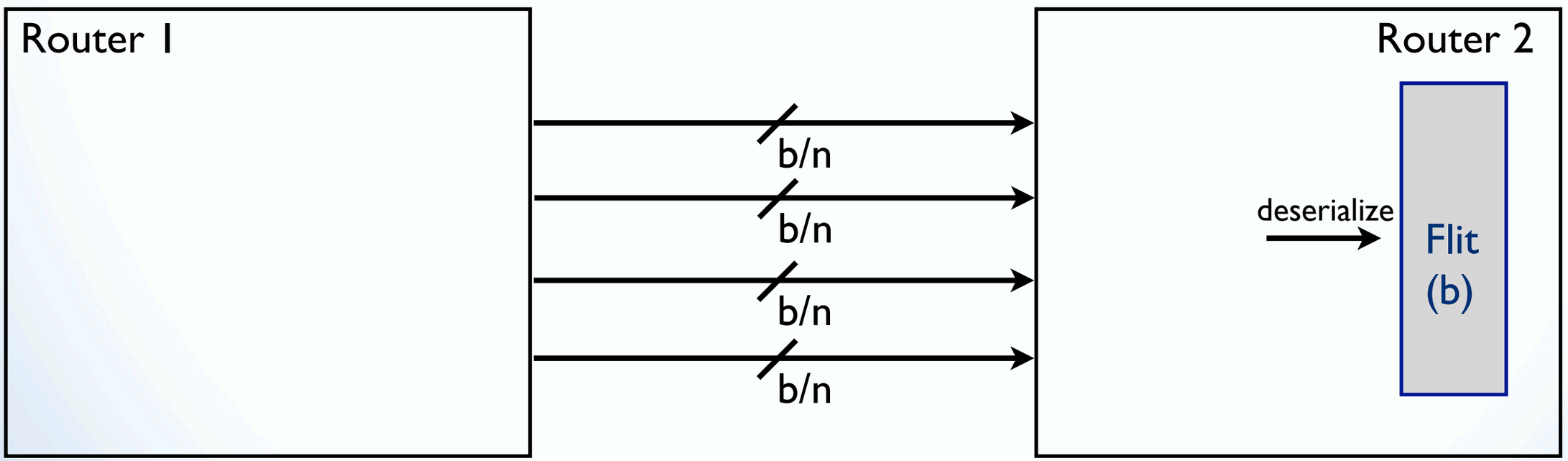
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



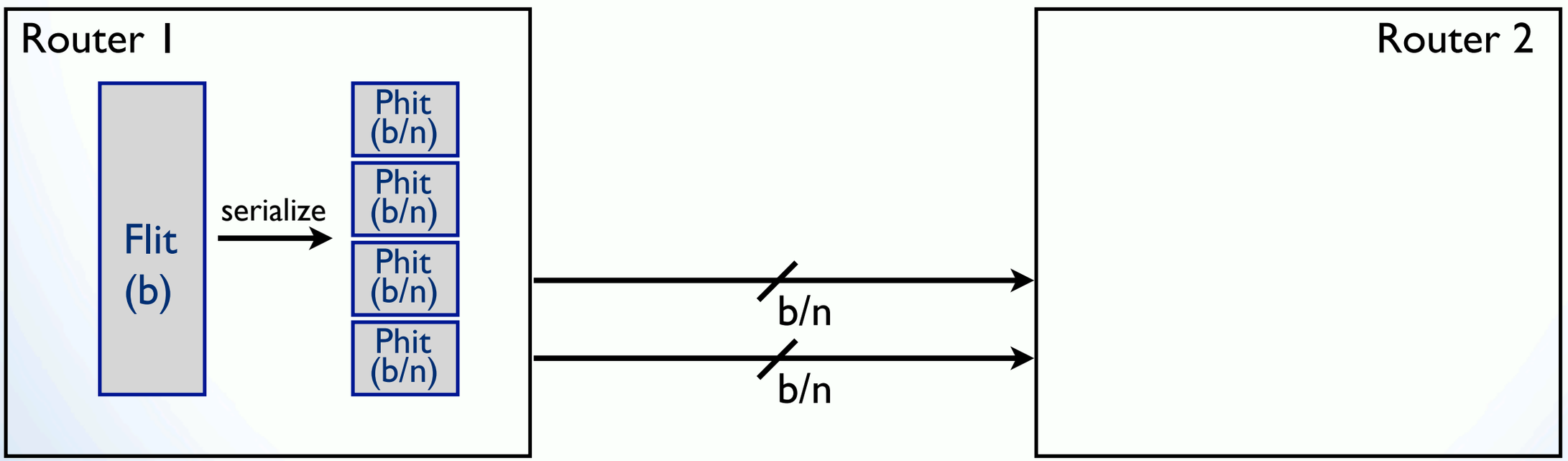
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



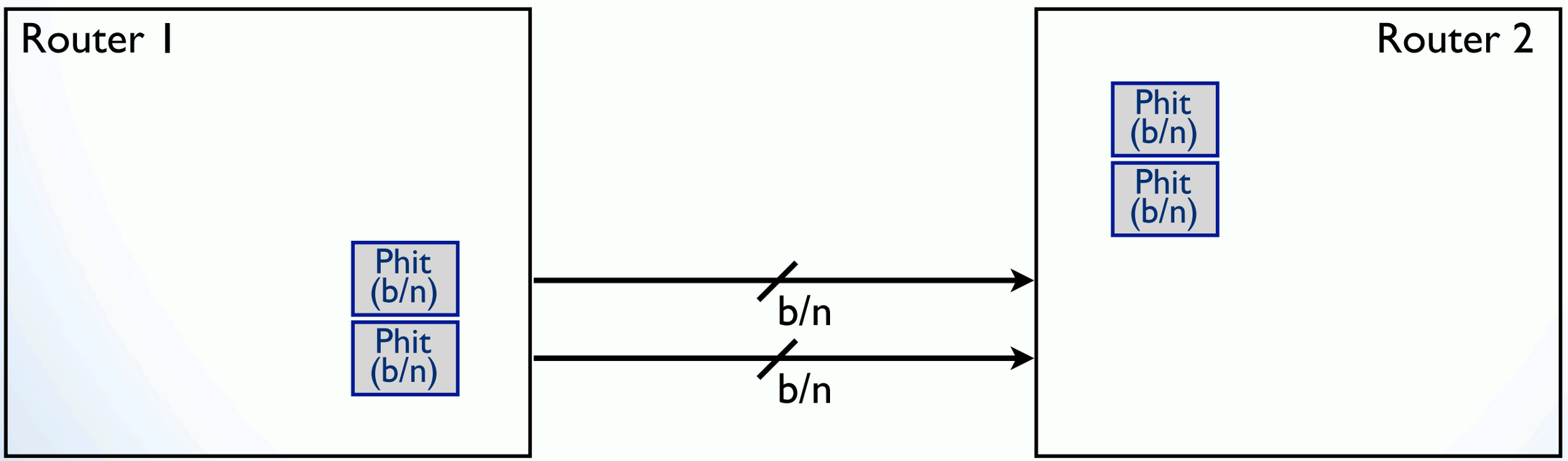
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



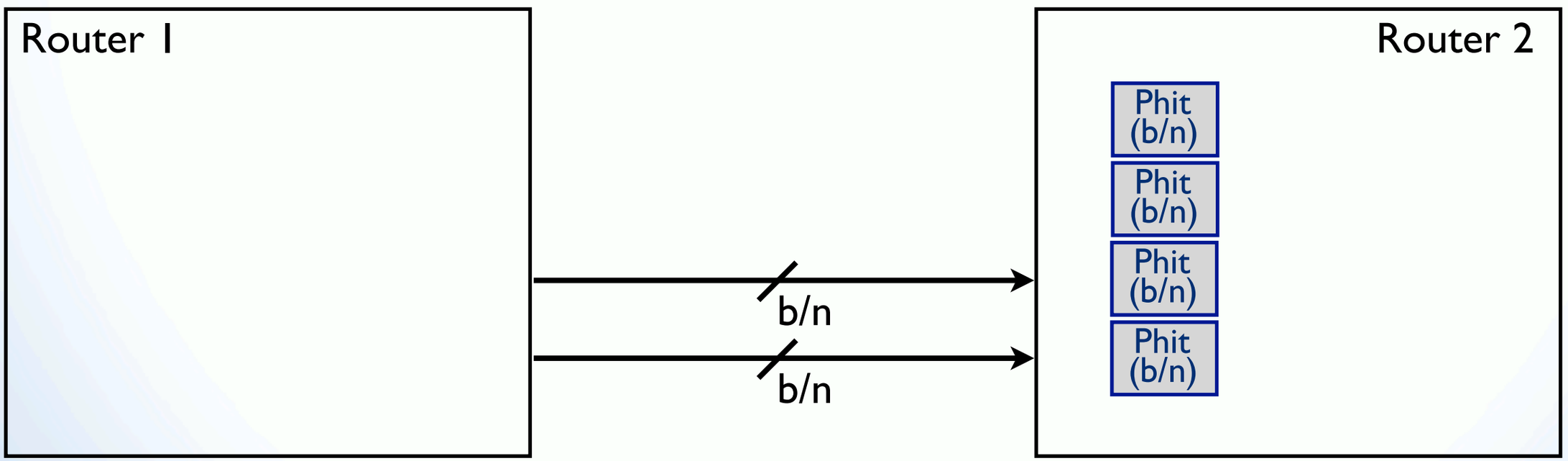
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



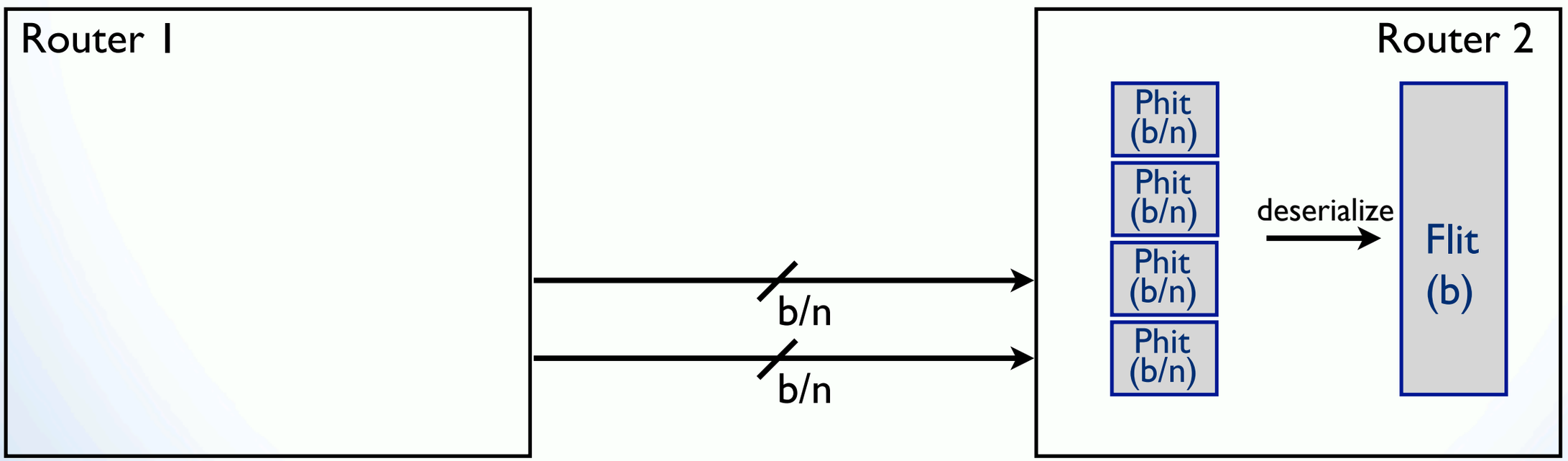
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



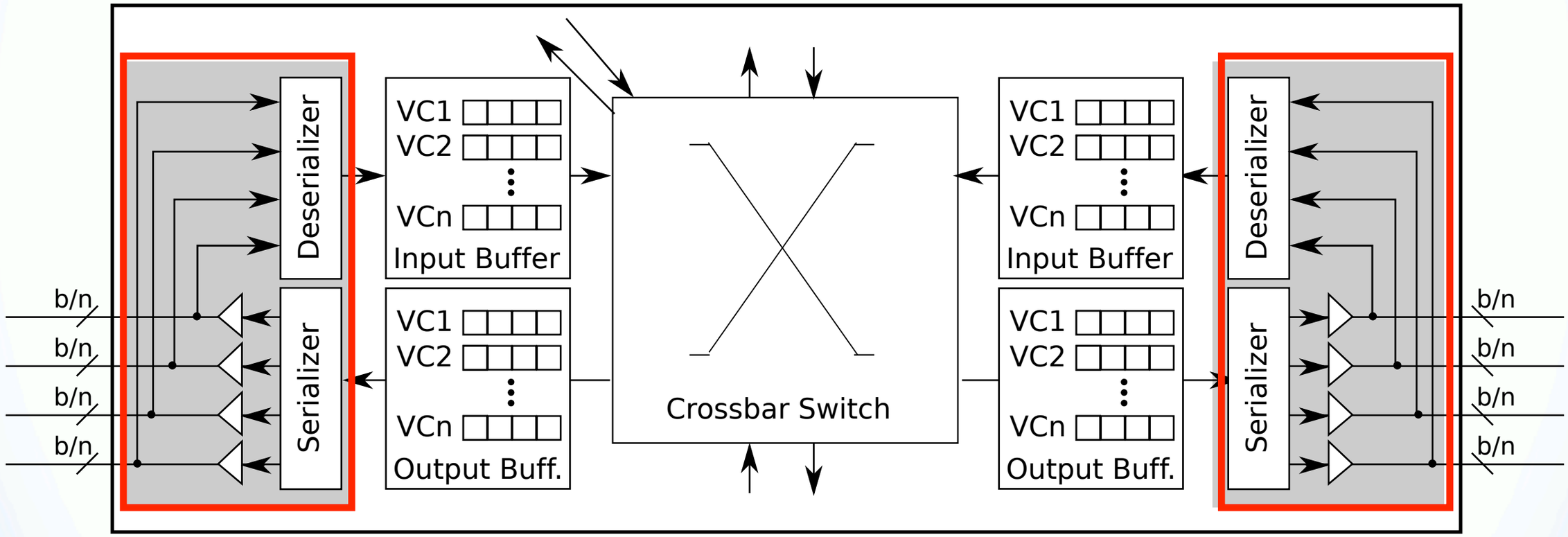
Phit-Serial Communication

- Conventionally in NoC, flit width is coupled to channel width
- Break flits (flow control units) into phits (physical transfer units) to decouple channel width from flit width



Microarchitecture

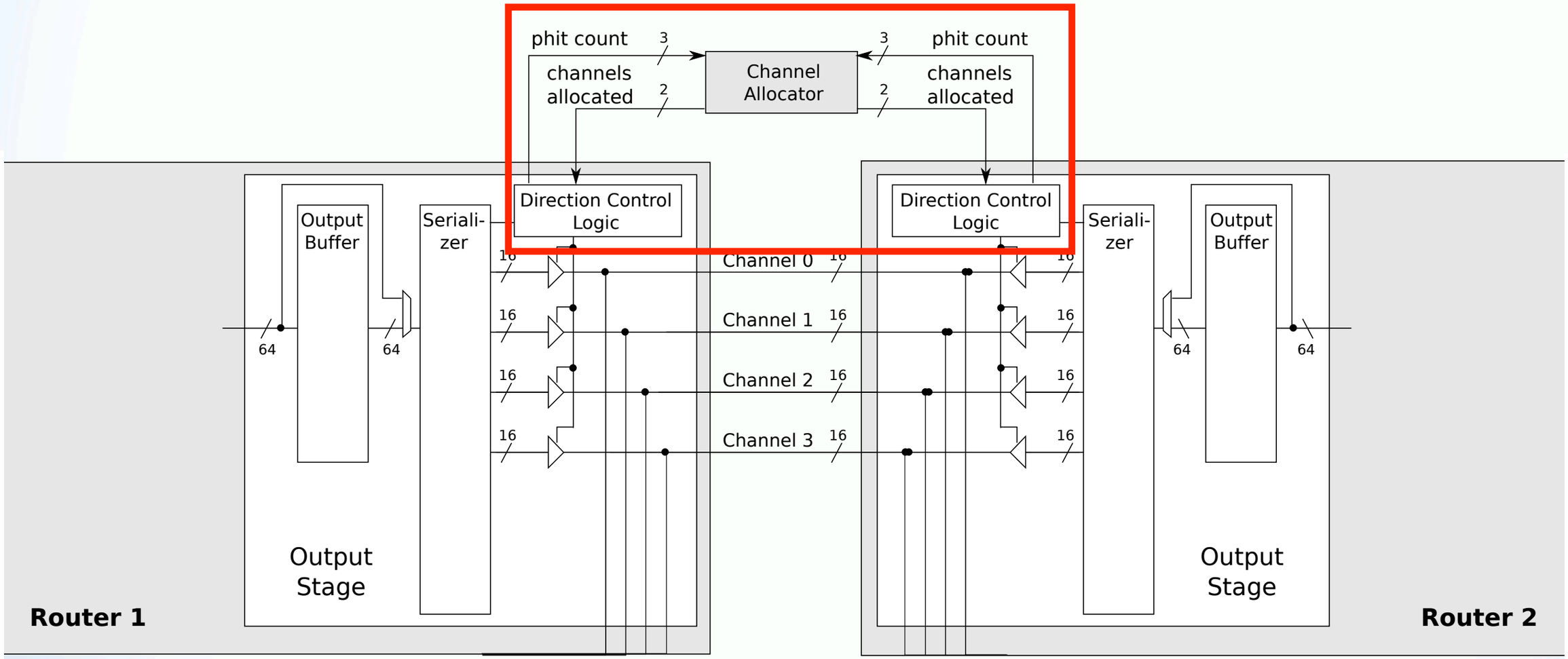
- Bandwidth-Adaptive Router (BAR): Only minimal modifications to standard VC router necessary



- Intra- & inter-router flow control is still flit-based

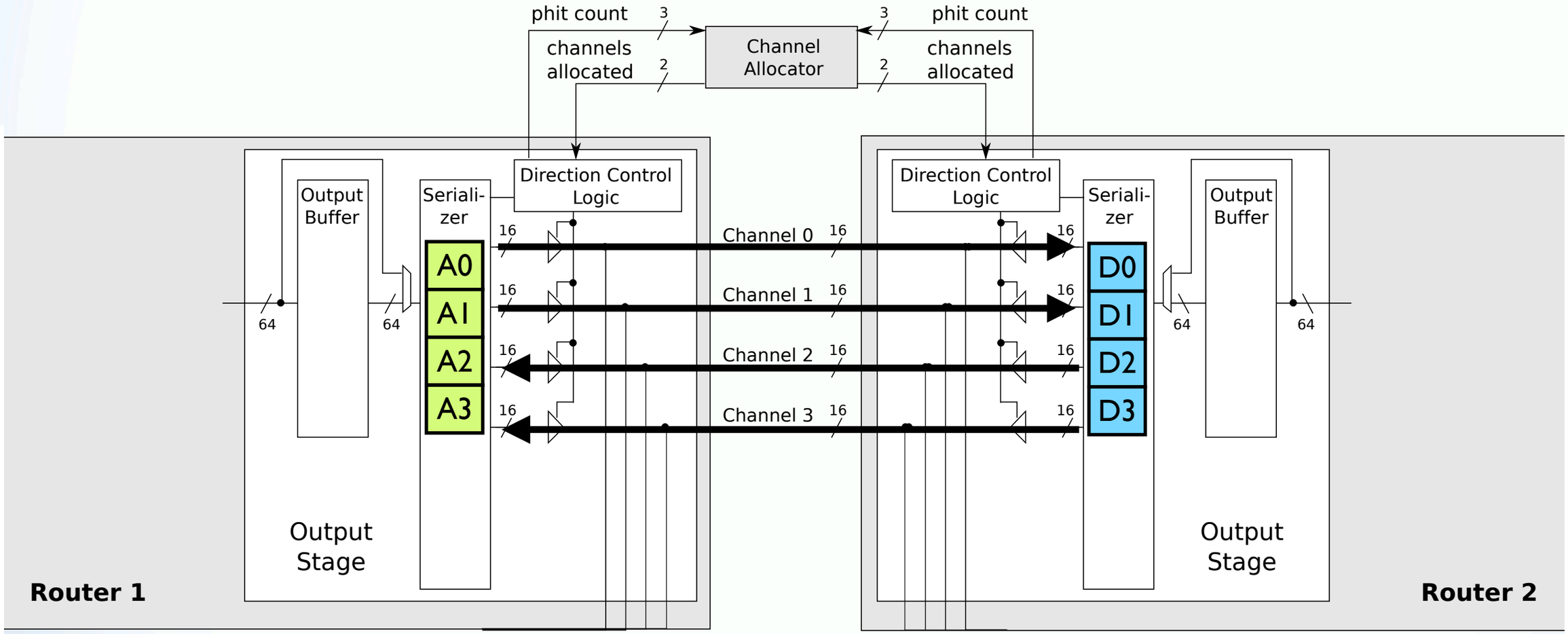
Bandwidth Allocation

- Pressure-based allocation of channels to directions



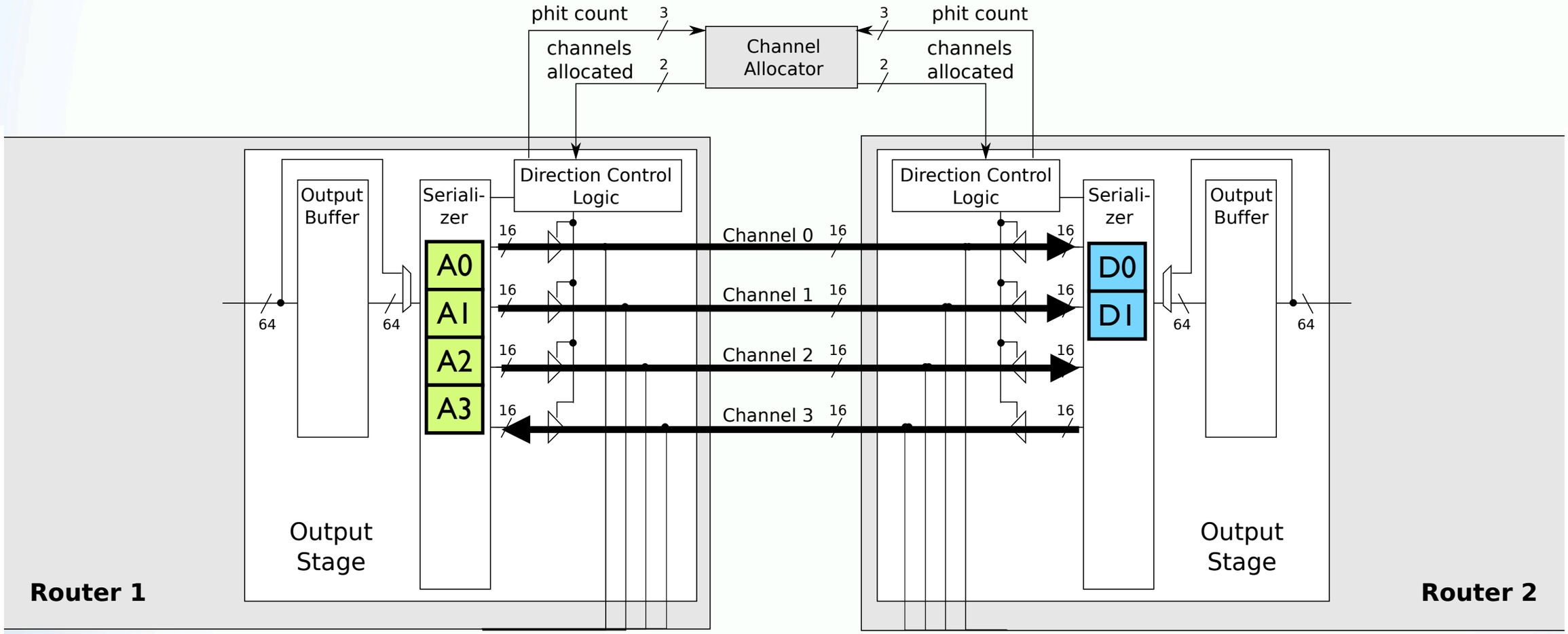
Bandwidth Allocation

- Pressure-based allocation of channels to directions



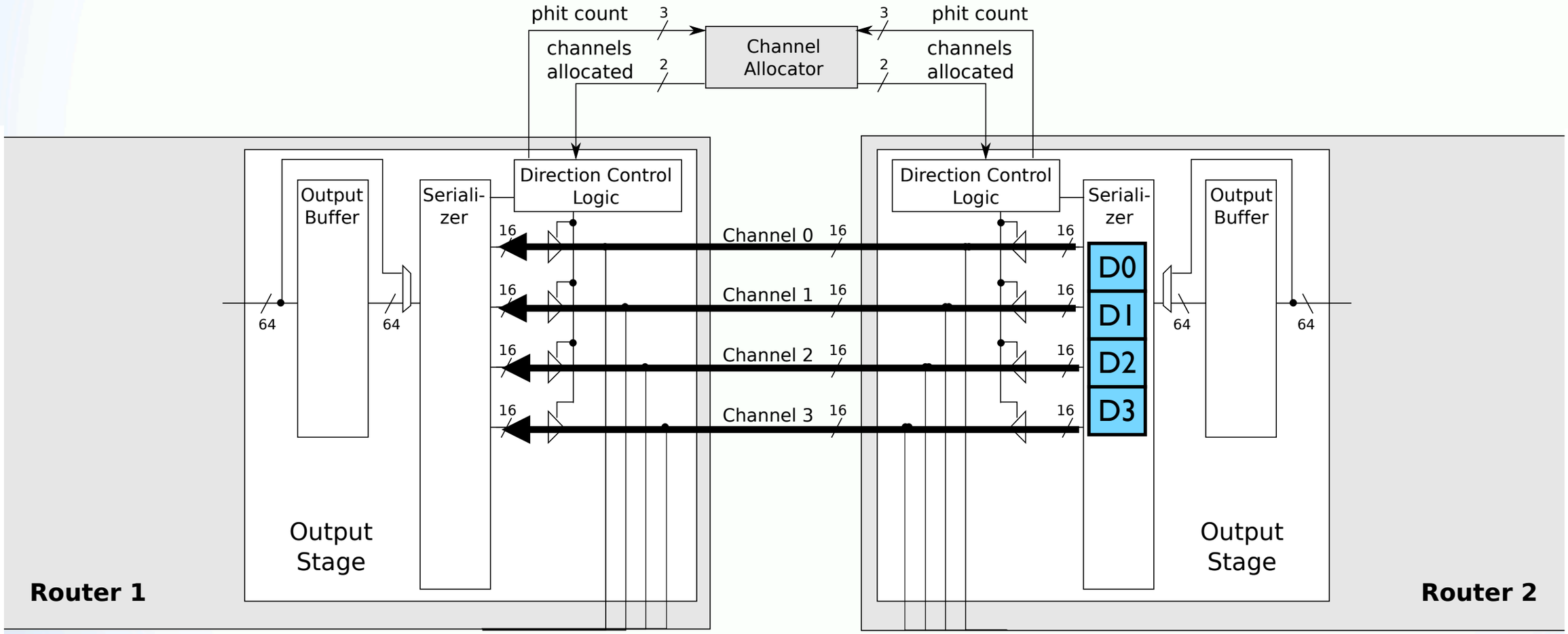
Bandwidth Allocation

- Pressure-based allocation of channels to directions

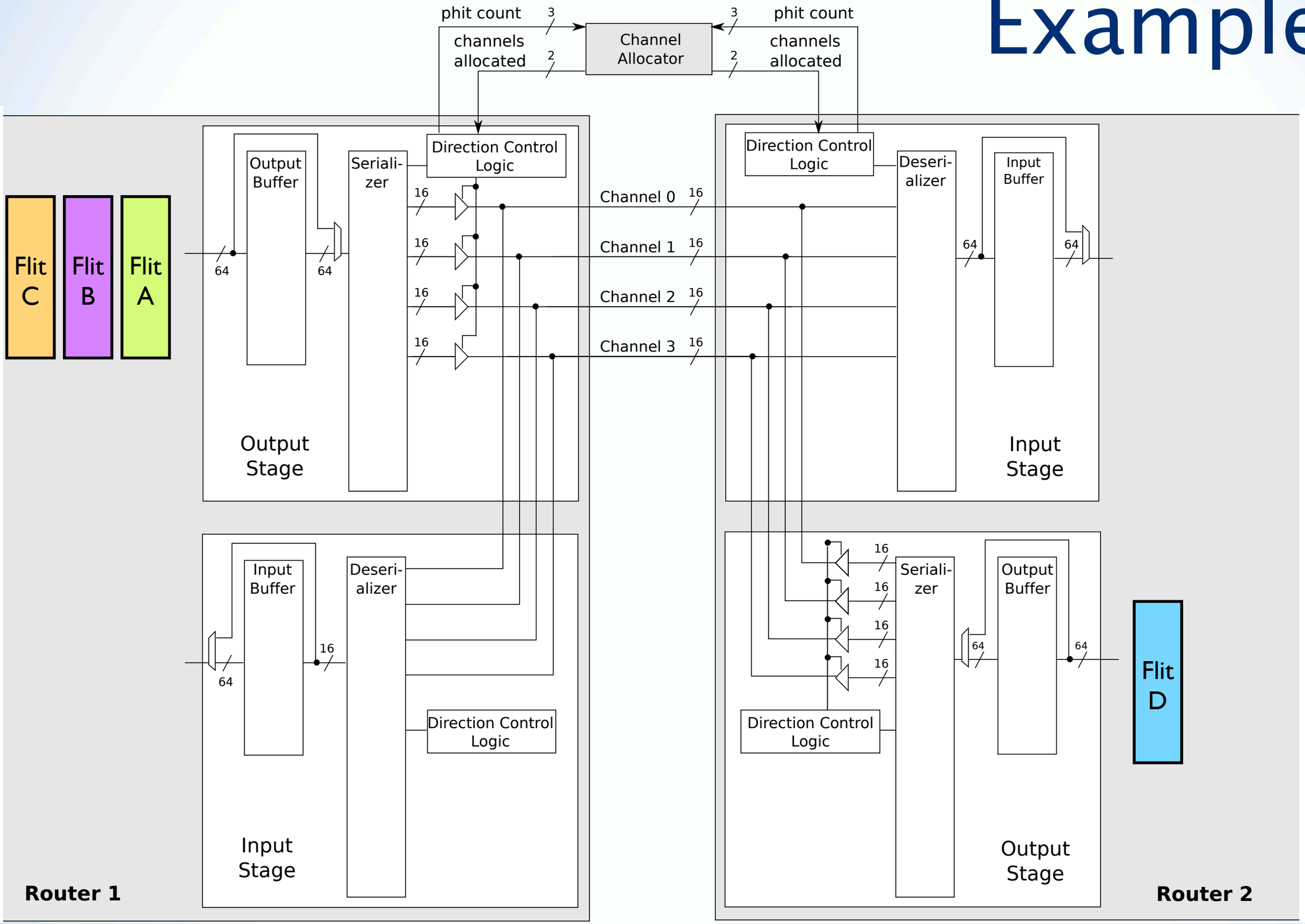


Bandwidth Allocation

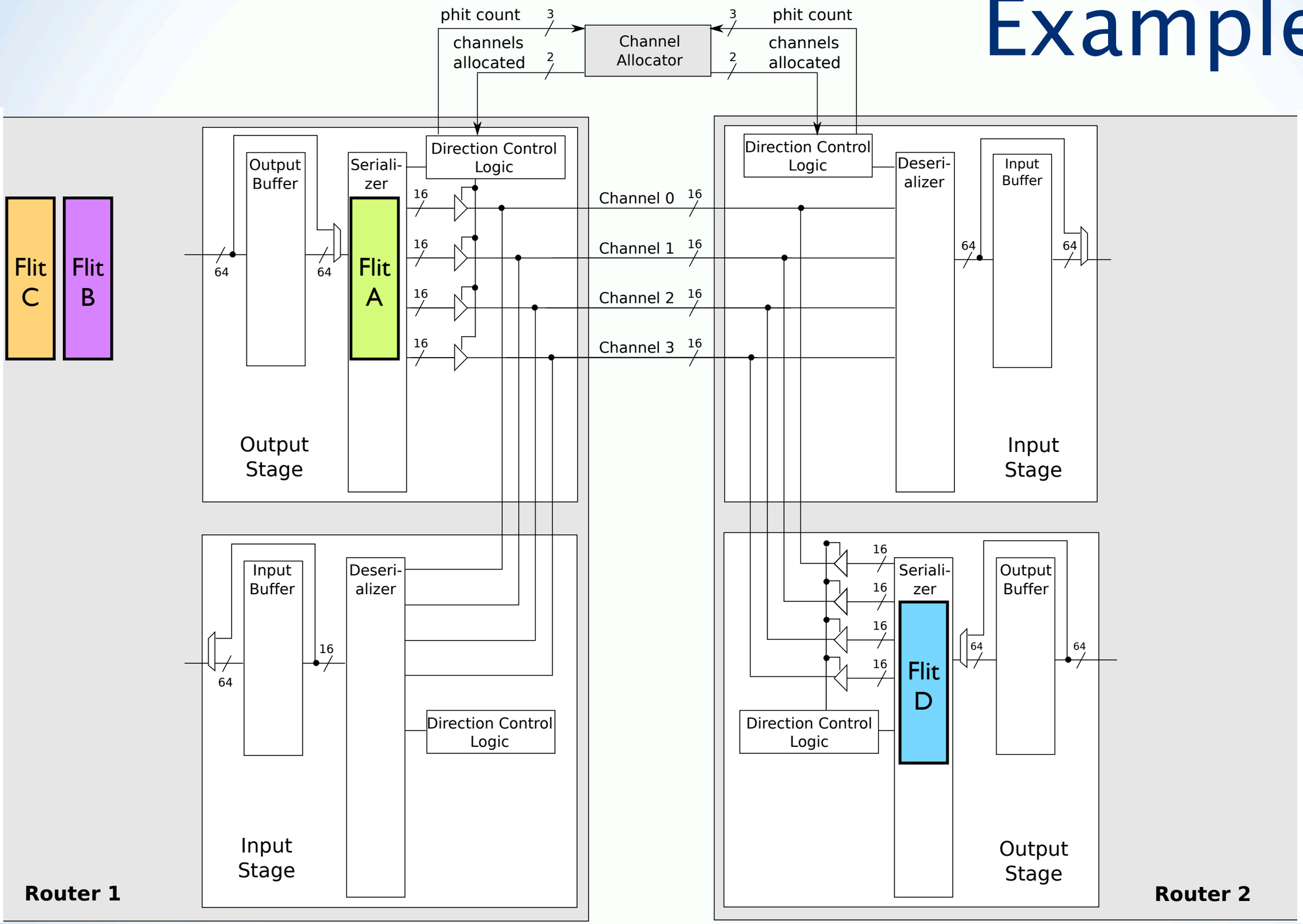
- Pressure-based allocation of channels to directions



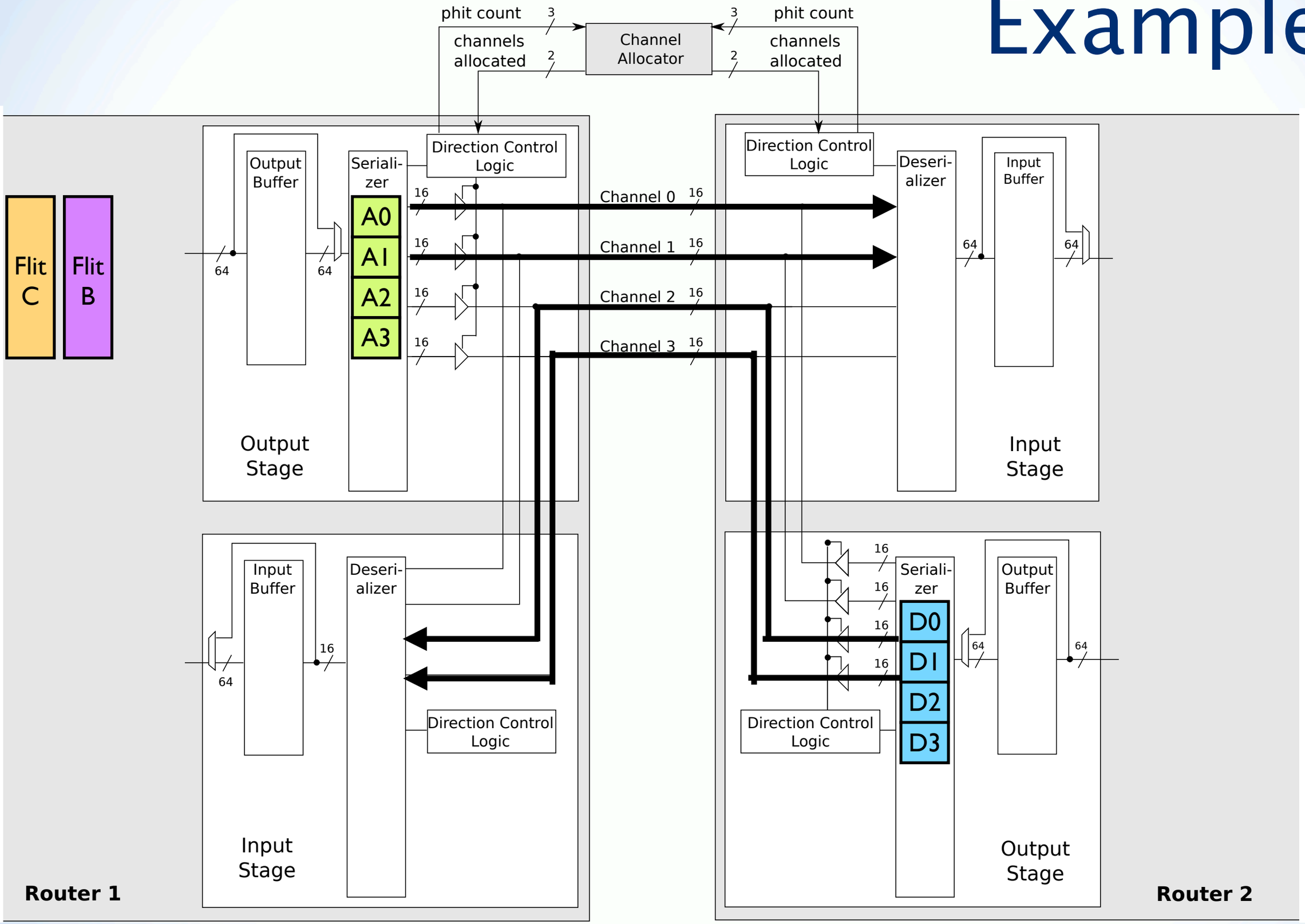
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Flit C
Flit B
Flit A
Output Buffer
Seriali-zer
Direction Control Logic
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Direction Control Logic
Deseri-alizer
Input Buffer
64
64
Input Buffer
Deseri-alizer
16
64
Direction Control Logic
Input Stage
Router 1
16
Seriali-zer
Output Buffer
64
64
Flit D
Direction Control Logic
Output Stage
Router 2

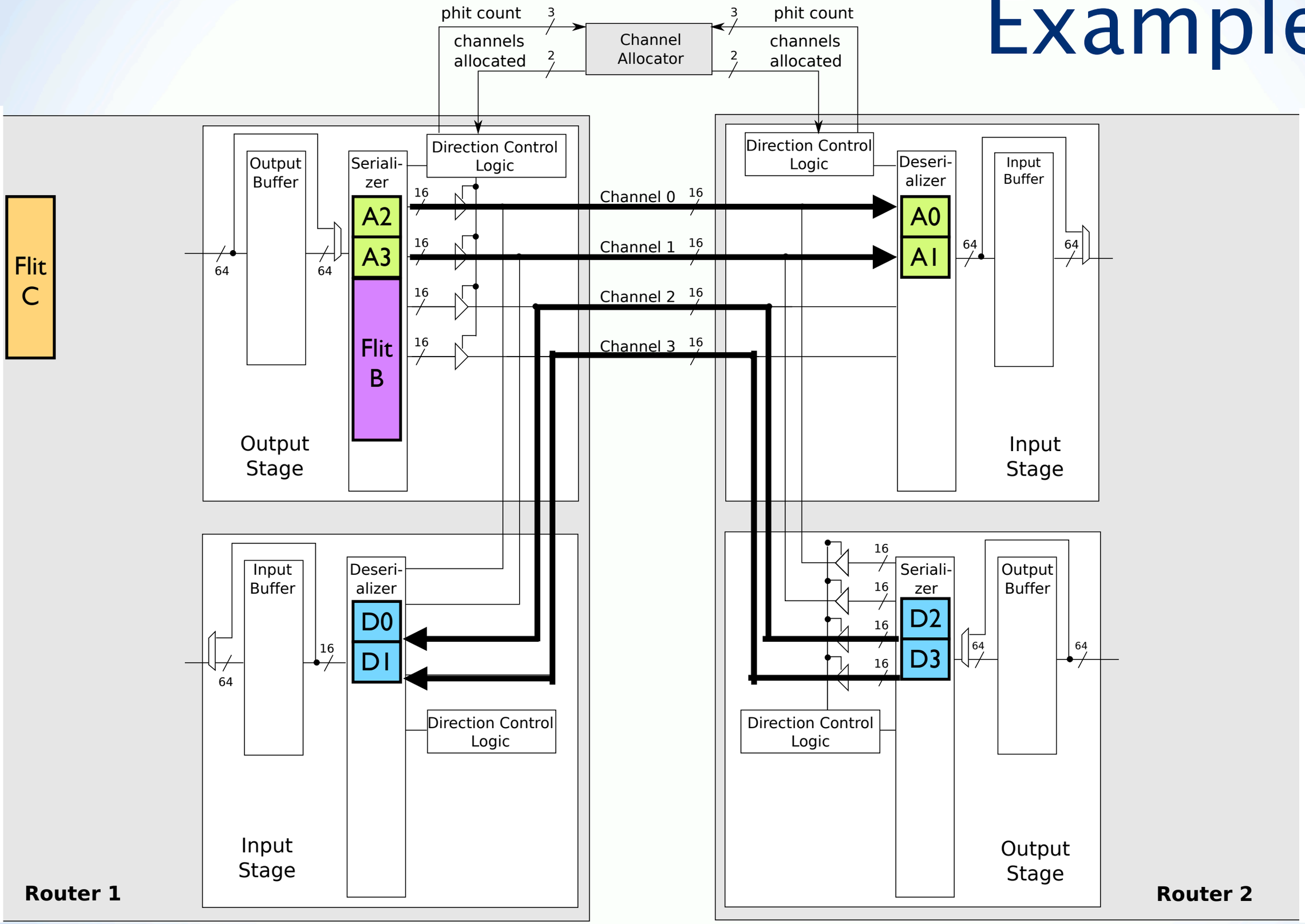
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Flit C
Flit B
Output Buffer
Serializer
Flit A
Direction Control Logic
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Input Buffer
Deserializer
Direction Control Logic
Input Stage
Router 1
Direction Control Logic
Deserializer
Input Buffer
64
64
Serializer
Flit D
Output Buffer
Output Stage
Router 2

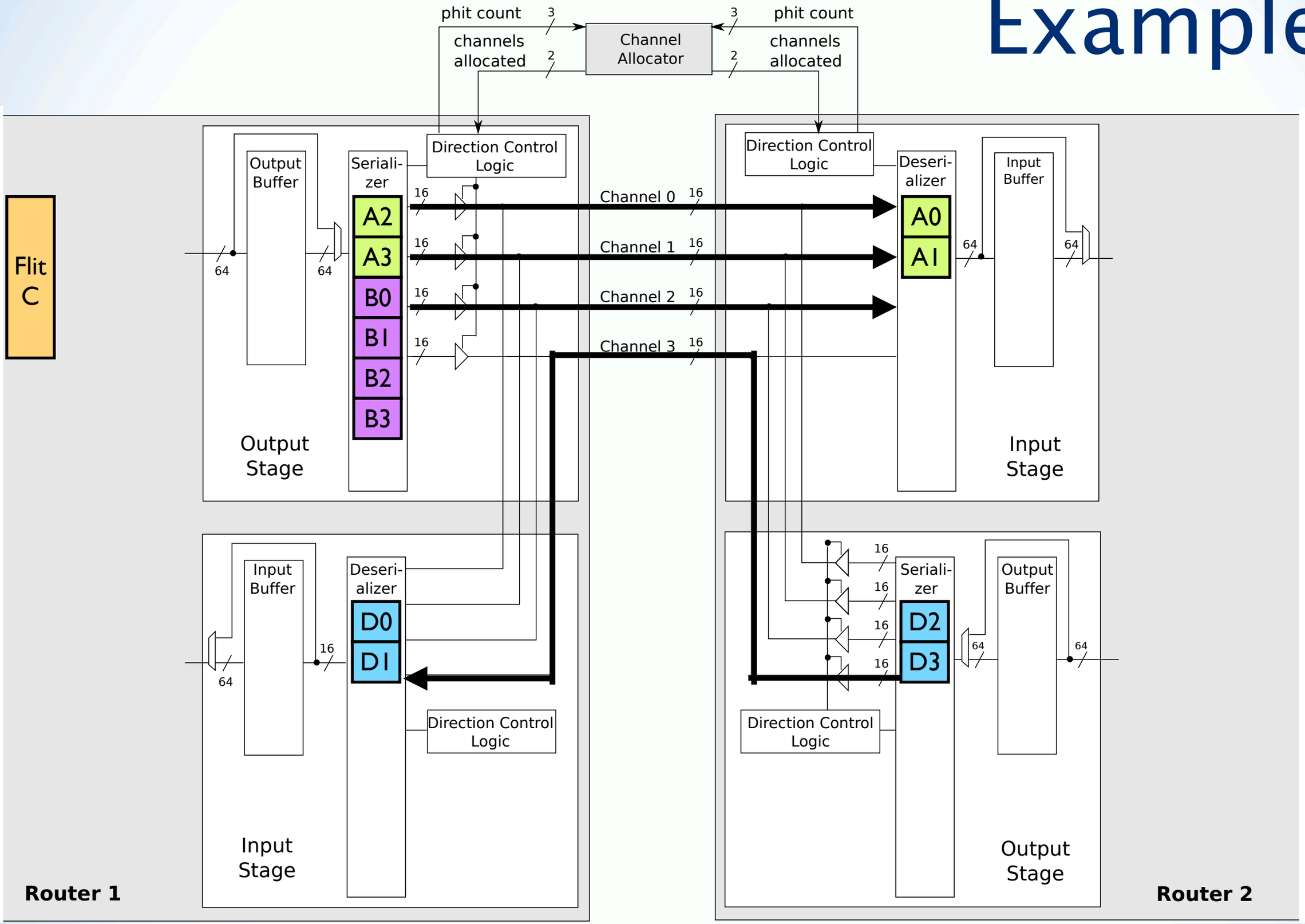
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Flit C
Flit B
Output Buffer
Serializer
A0
A1
A2
A3
Direction Control Logic
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Deserializer
Input Buffer
Input Stage
Router 1
Router 2
D0
D1
D2
D3

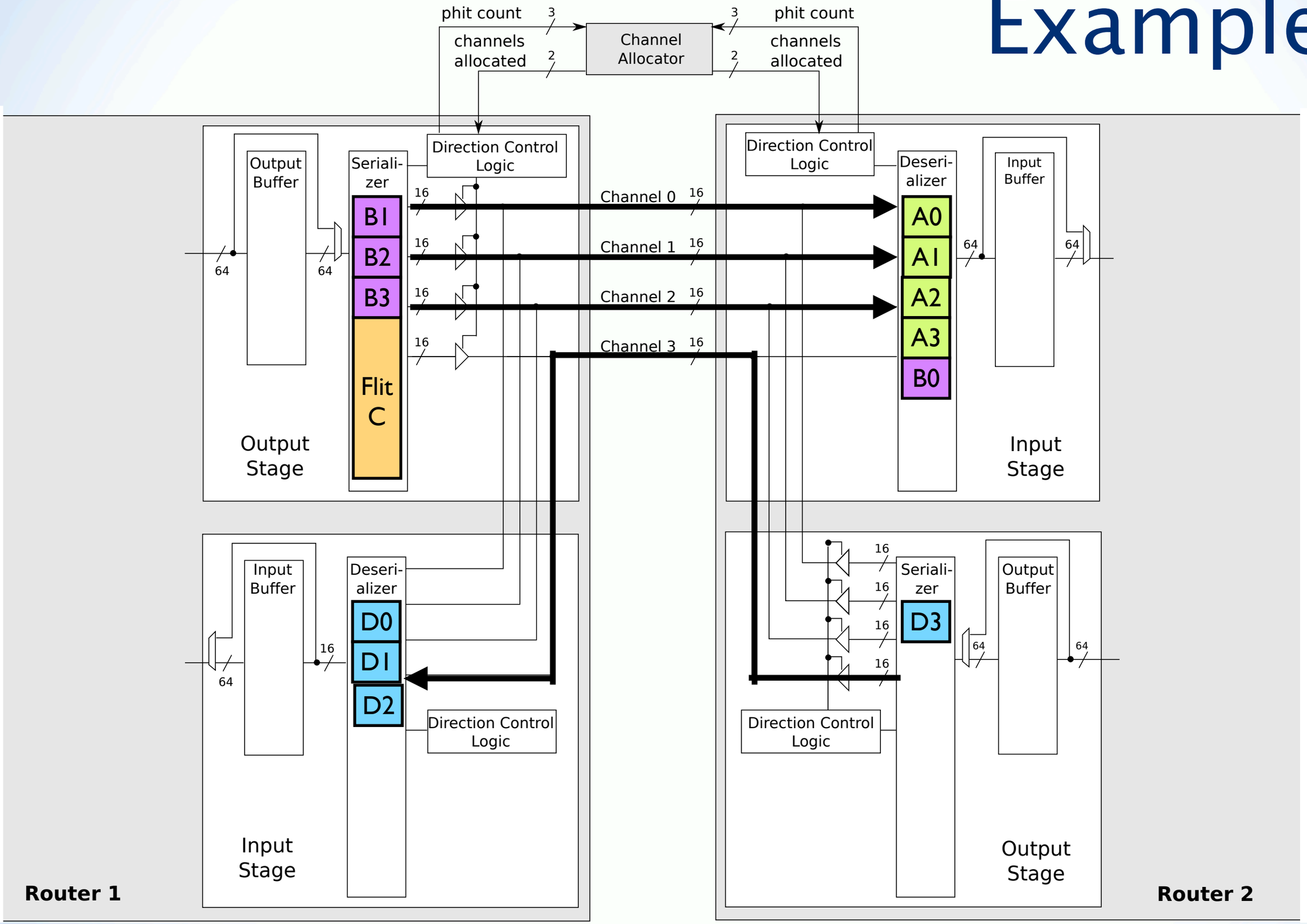
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Flit C
Output Buffer
Serializer
Direction Control Logic
A2
A3
Flit B
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Direction Control Logic
Deserializer
A0
A1
Input Buffer
64
64
Input Stage
Input Buffer
Deserializer
D0
D1
16
64
Direction Control Logic
Input Stage
Router 1
16
Serializer
D2
D3
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

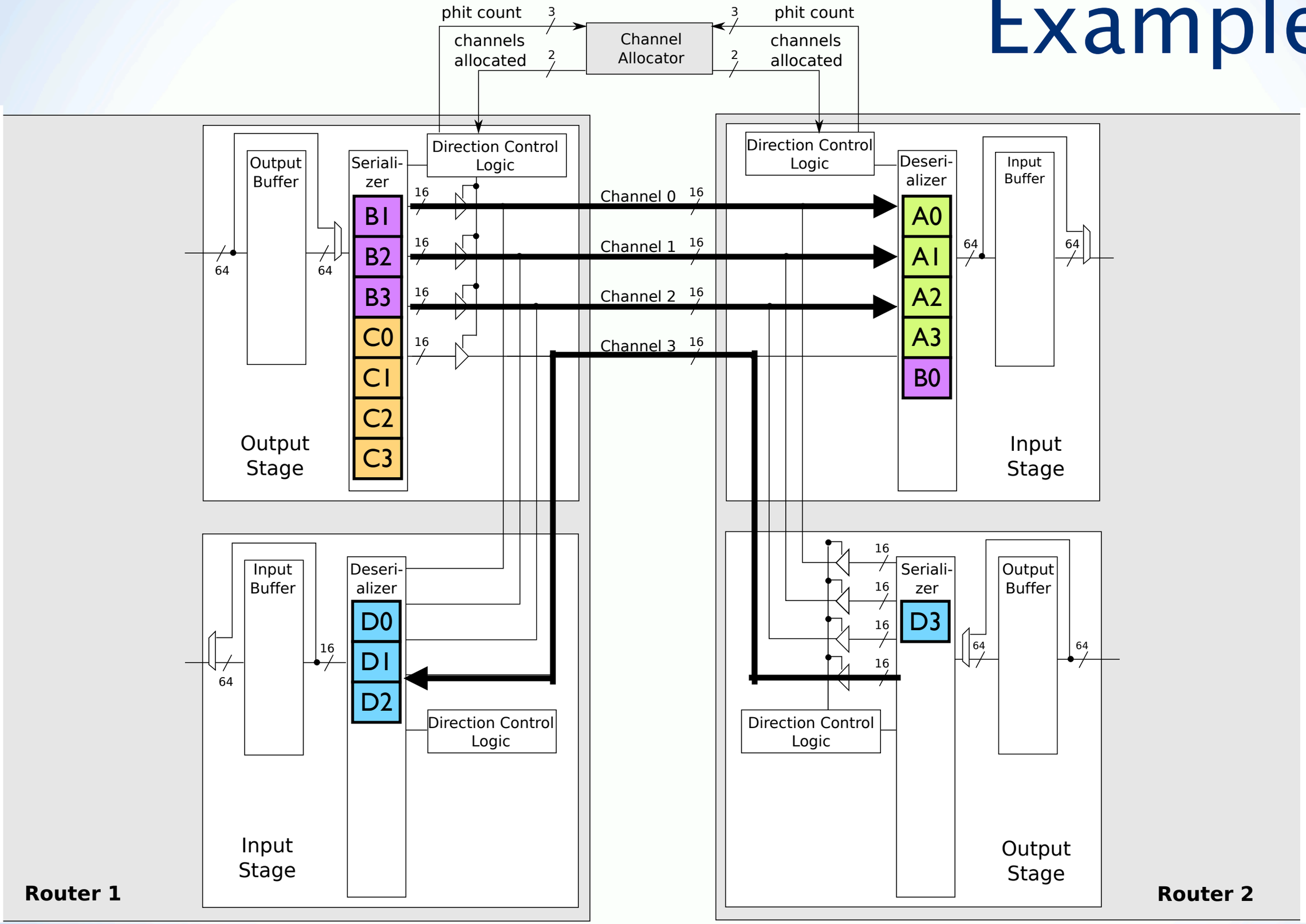
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Flit C
Output Buffer
Seriali-zer
Direction Control Logic
A2
A3
B0
B1
B2
B3
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Direction Control Logic
Deseri-alizer
A0
A1
Input Buffer
64
64
Output Stage
Input Stage
Input Buffer
Deseri-alizer
D0
D1
16
64
Direction Control Logic
Input Stage
Router 1
16
Seriali-zer
D2
D3
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

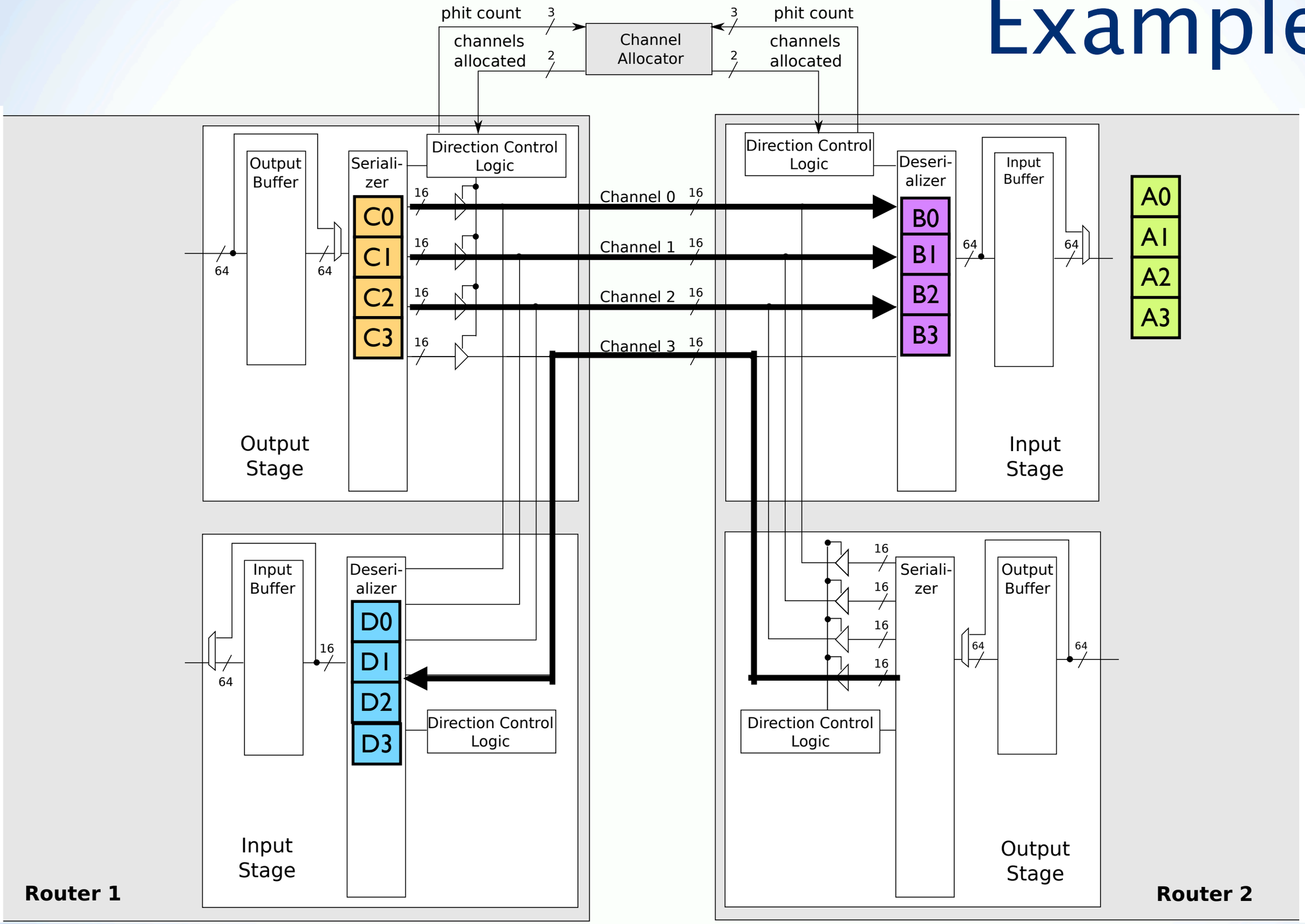
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Serializer
B1
B2
B3
Flit C
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Direction Control Logic
Deserializer
A0
A1
A2
A3
B0
Input Buffer
64
64
Input Stage
Input Buffer
Deserializer
D0
D1
D2
64
16
Direction Control Logic
Input Stage
Router 1
16
Serializer
D3
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

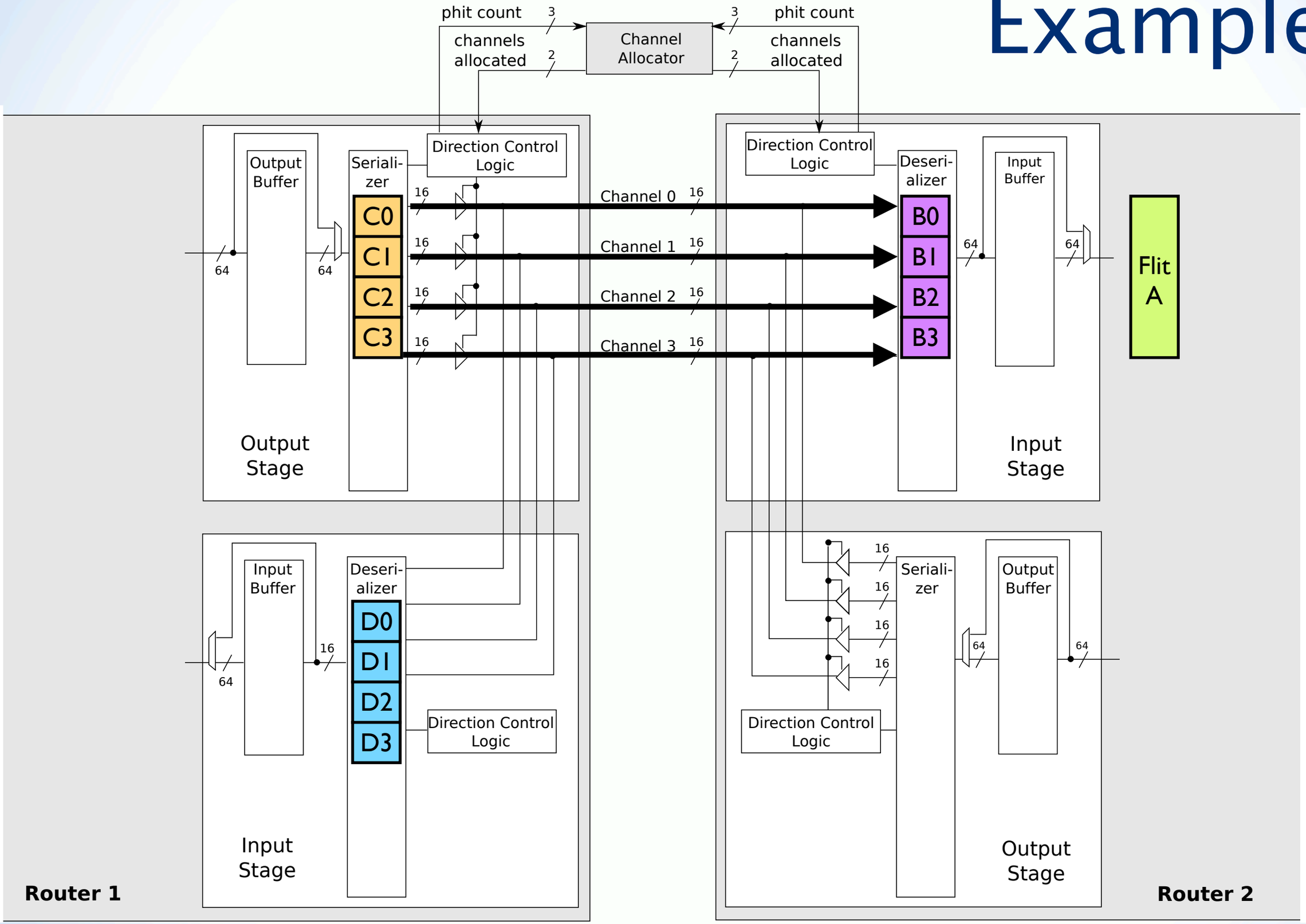
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Seriali-zer
B1
B2
B3
C0
C1
C2
C3
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Direction Control Logic
Deseri-alizer
A0
A1
A2
A3
B0
Input Buffer
64
64
Input Stage
Input Buffer
Deseri-alizer
D0
D1
D2
64
16
Direction Control Logic
Input Stage
Router 1
Seriali-zer
D3
Output Buffer
Direction Control Logic
Output Stage
Router 2

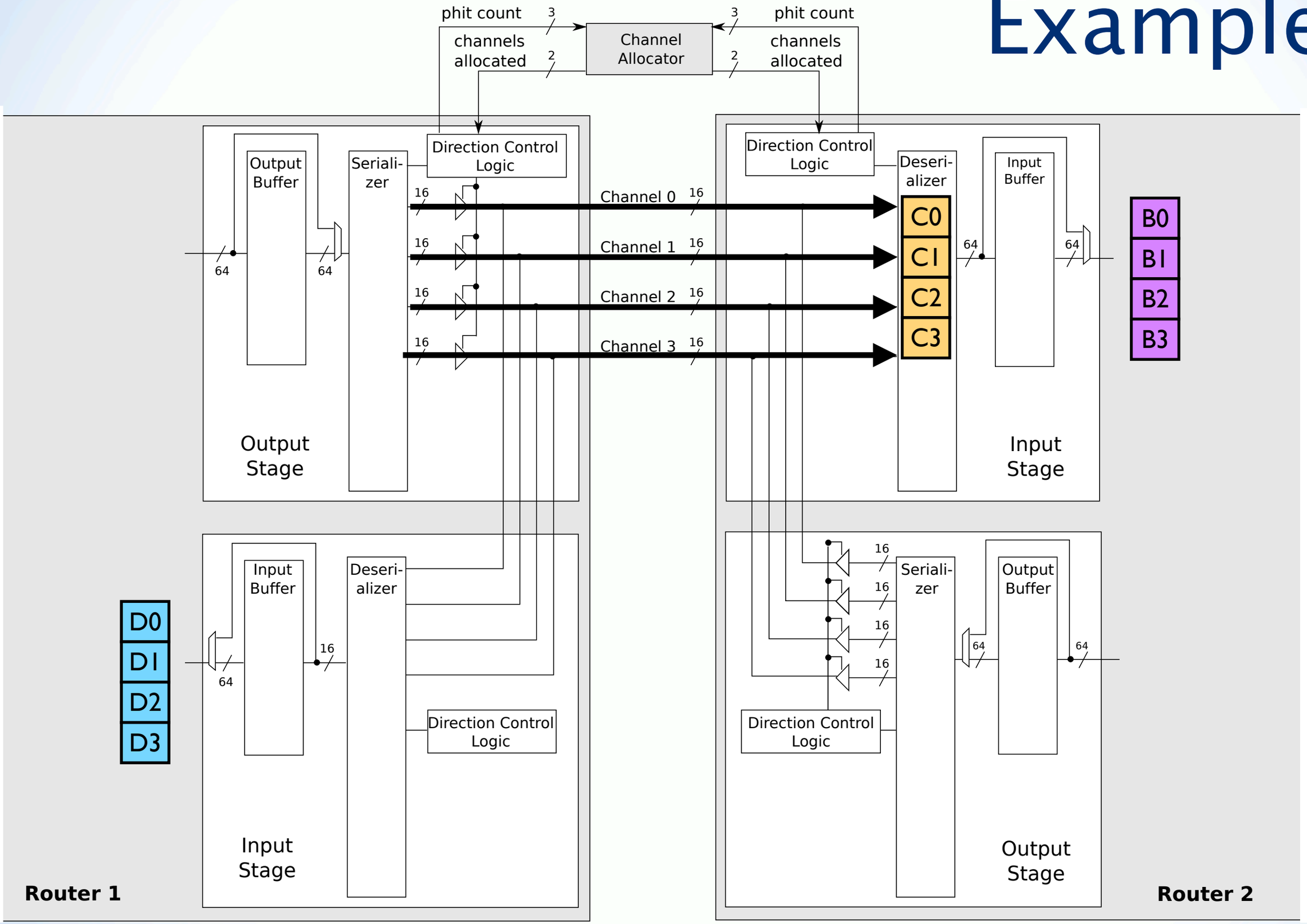
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Serializer
C0
C1
C2
C3
64
64
16
Channel 0
16
Channel 1
16
Channel 2
16
Channel 3
16
Output Stage
Direction Control Logic
Deserializer
B0
B1
B2
B3
Input Buffer
64
64
A0
A1
A2
A3
Input Stage
Input Buffer
64
16
Deserializer
D0
D1
D2
D3
Direction Control Logic
Input Stage
Router 1
16
16
16
16
Serializer
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

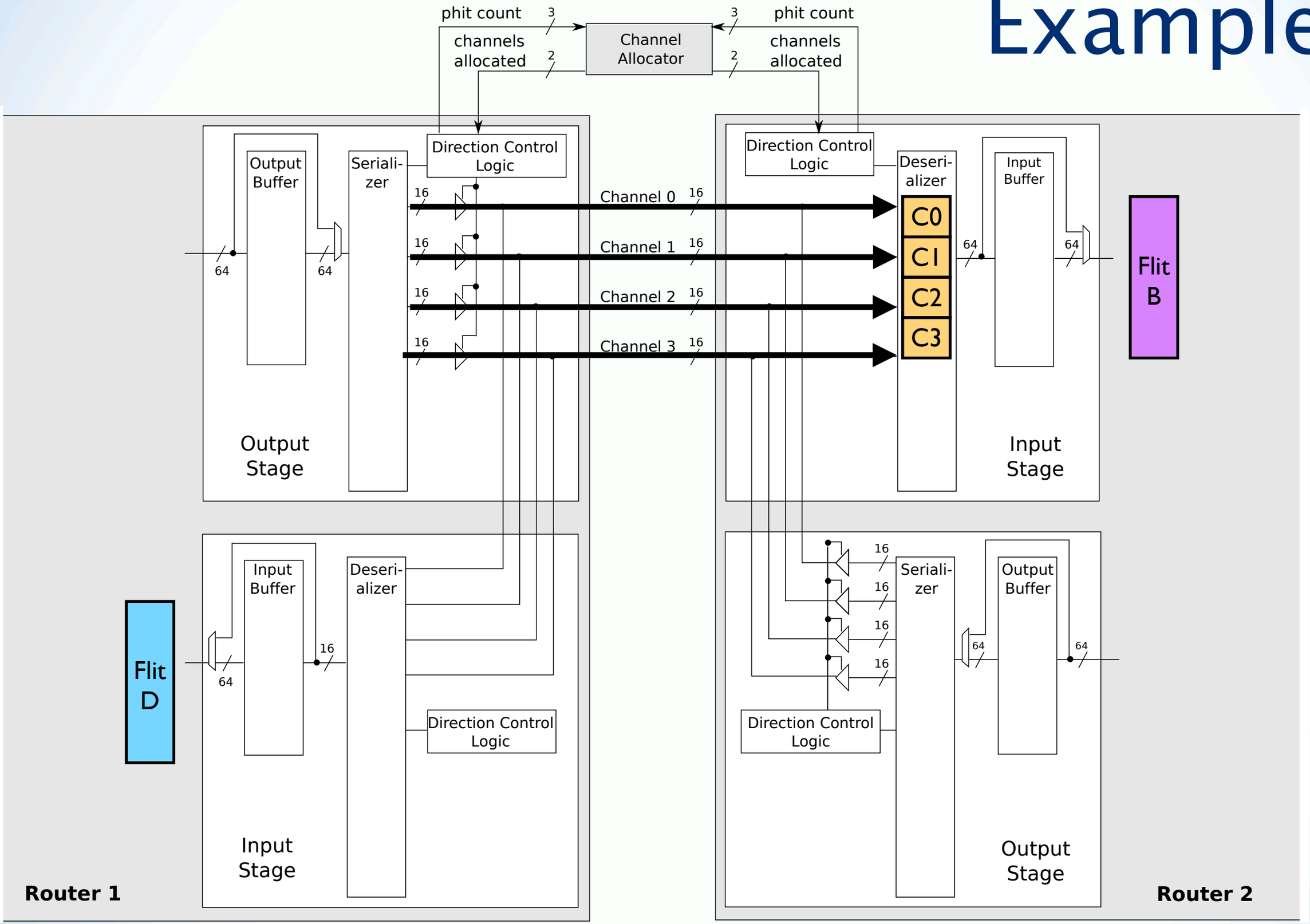
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Seriali-zer
C0
C1
C2
C3
64
64
16
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Direction Control Logic
Deseri-alizer
B0
B1
B2
B3
Input Buffer
64
64
Flit A
Input Stage
Input Buffer
Deseri-alizer
D0
D1
D2
D3
64
16
Direction Control Logic
Input Stage
Router 1
16
Seriali-zer
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

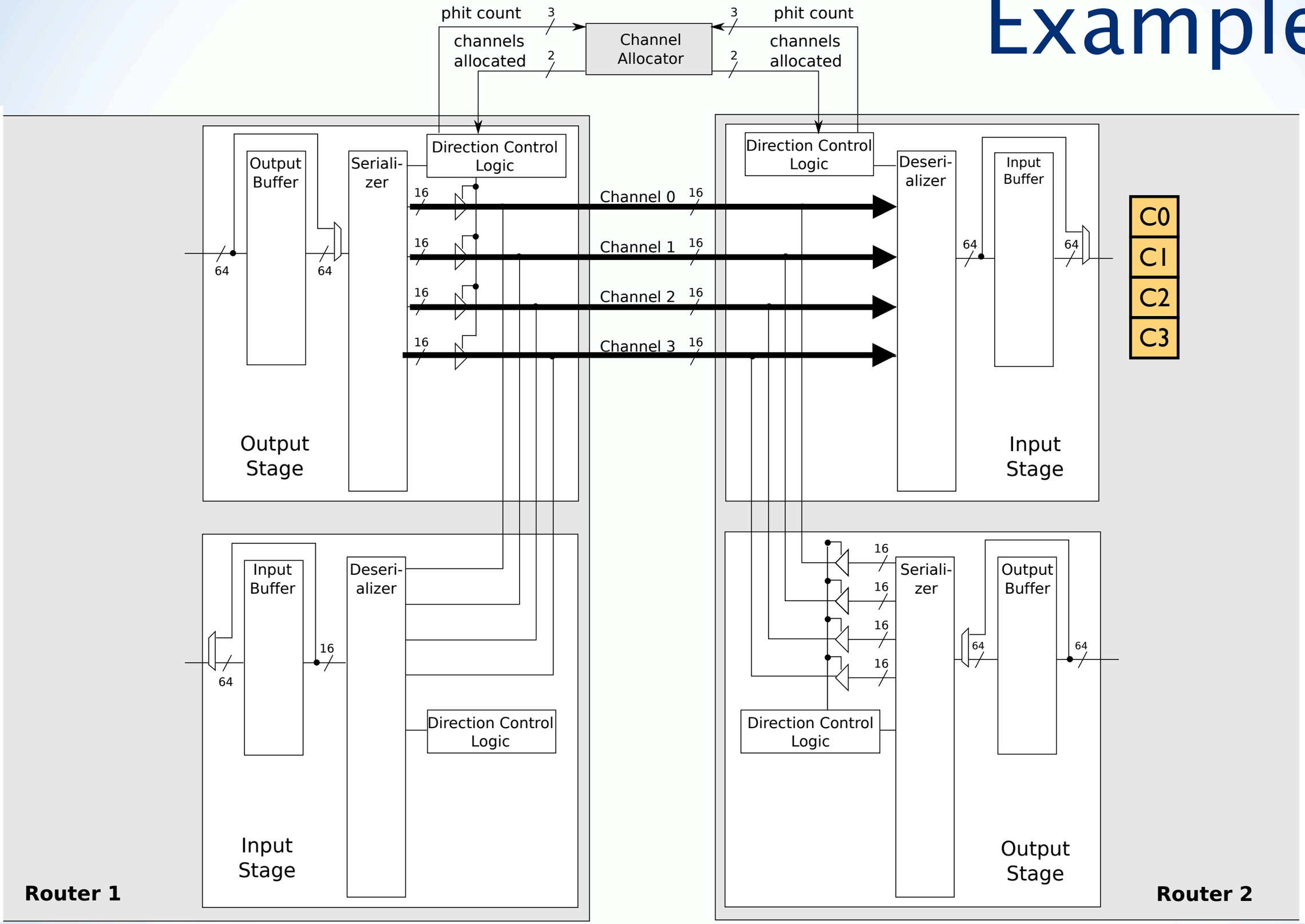
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Seriali-zer
64
64
16
Channel 0
16
Channel 1
Channel 2
Channel 3
Direction Control Logic
Deseri-alizer
C0
C1
C2
C3
Input Buffer
64
64
B0
B1
B2
B3
Output Stage
Input Stage
D0
D1
D2
D3
Input Buffer
64
16
Deseri-alizer
Direction Control Logic
Input Stage
Router 1
16
16
16
16
Seriali-zer
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

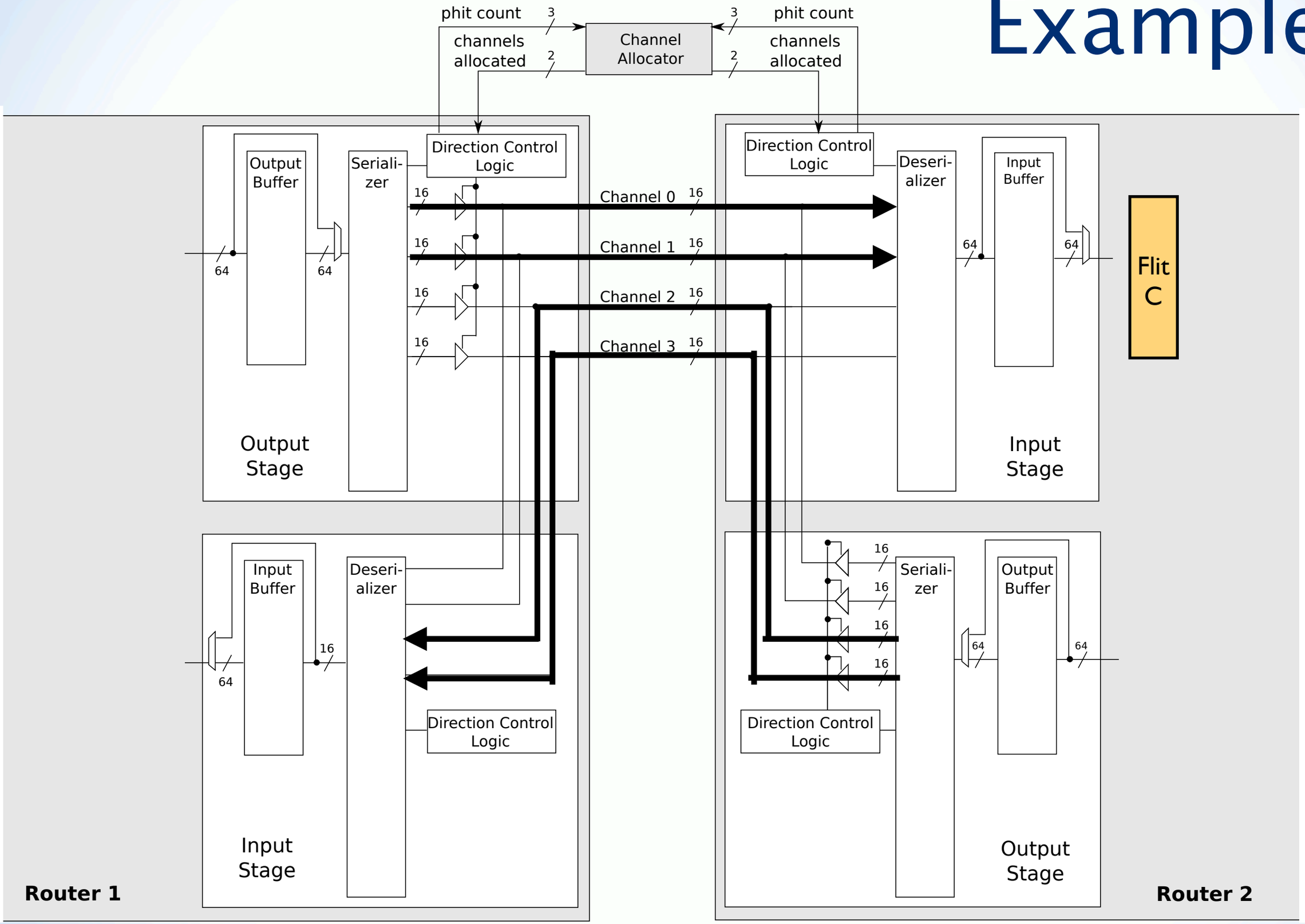
Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Seriali-zer
64
64
16
Channel 0
16
Channel 1
Channel 2
Channel 3
Output Stage
Direction Control Logic
Deseri-alizer
C0
C1
C2
C3
Input Buffer
64
64
Flit B
Flit D
Input Buffer
64
16
Deseri-alizer
Direction Control Logic
Input Stage
Router 1
16
Seriali-zer
Output Buffer
64
64
Direction Control Logic
Output Stage
Router 2

Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Seriali-zer
Deseri-alizer
Input Buffer
64
16
Channel 0
Channel 1
Channel 2
Channel 3
C0
C1
C2
C3
Output Stage
Input Stage
Router 1
Router 2

Example


phit count
3
channels allocated
2
Channel Allocator
3
phit count
2
channels allocated
Direction Control Logic
Output Buffer
Seriali-zer
Deseri-alizer
Input Buffer
16
64
Channel 0
Channel 1
Channel 2
Channel 3
Output Stage
Input Stage
Flit C
Router 1
Router 2

Related Work

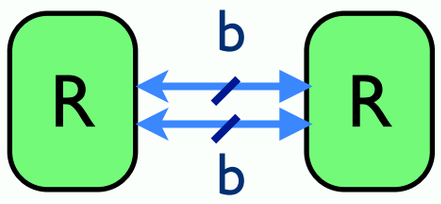
Related Work

- Previous Work:

Related Work

- Previous Work:
  - BiNoC (Lan et al., NOCS 2009)
    - only coarse-grained BW adaptivity

BINOC



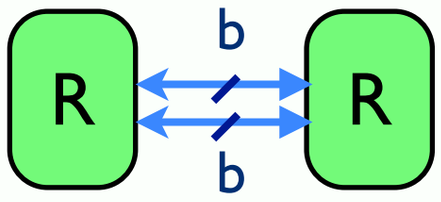
Link BW: **2*b**

XBar ports: **2*P**

Related Work

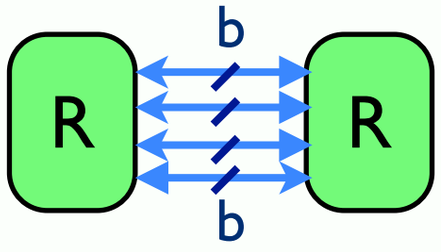
- Previous Work:
  - –BiNoC (Lan et al., NOCS 2009)
    - only coarse-grained BW adaptivity
  - –Oblivious Routing in On-Chip Bandwidth-Adaptive Networks (Cho et al., PACT 2009)

BINOC



Link BW: **2*b**
XBar ports: **2*P**

BWADAPTIVE

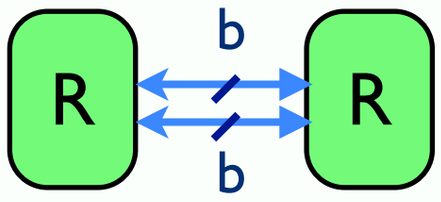


Link BW: **N*b**
XBar ports: **N*P**

Related Work

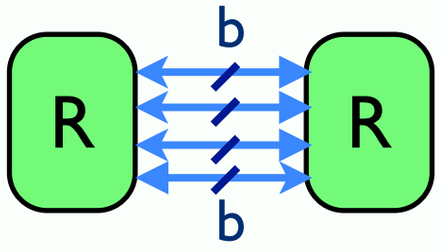
- Previous Work:
  - BiNoC (Lan et al., NOCS 2009)
    - only coarse-grained BW adaptivity
  - Oblivious Routing in On-Chip Bandwidth-Adaptive Networks (Cho et al., PACT 2009)
- Objective is different:

BINOC



Link BW: **2*b**
XBar ports: **2*P**

BWADAPTIVE

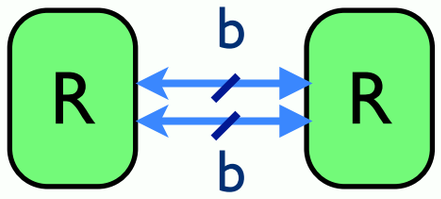


Link BW: **N*b**
XBar ports: **N*P**

Related Work

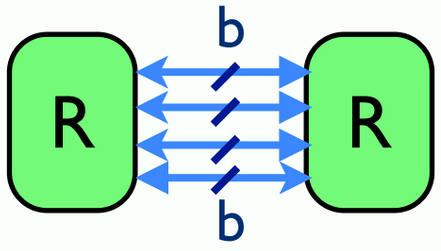
- Previous Work:
  - BiNoC (Lan et al., NOCS 2009)
    - only coarse-grained BW adaptivity
  - Oblivious Routing in On-Chip Bandwidth-Adaptive Networks (Cho et al., PACT 2009)
- Objective is different:

BINOC



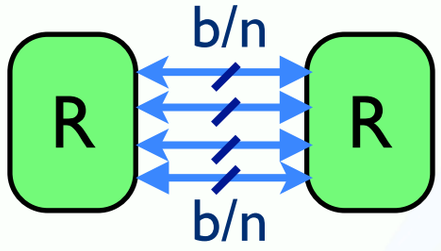
Link BW: **2*b**
XBar ports: **2*P**

BWADAPTIVE



Link BW: **N*b**
XBar ports: **N*P**

BAR

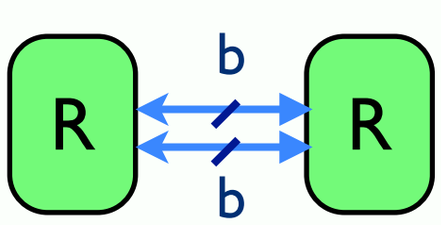
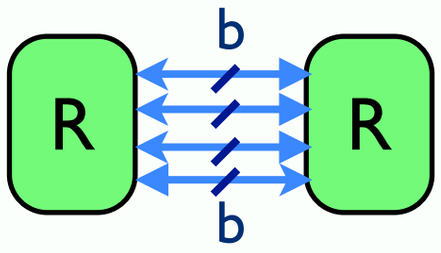
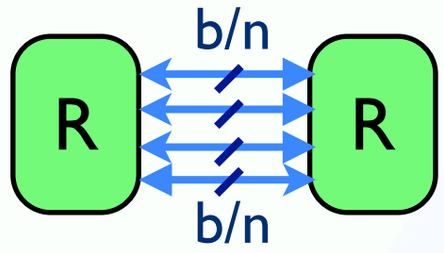


Link BW: ≤ **b**
XBar ports: **P**

Related Work

- Previous Work:
  - BiNoC (Lan et al., NOCS 2009)
    - only coarse-grained BW adaptivity
  - Oblivious Routing in On-Chip Bandwidth-Adaptive Networks (Cho et al., PACT 2009)
- Objective is different:

Remember: BW demands << b

BINOC	BWADAPTIVE	BAR
		
Link BW: **2*b**	Link BW: **N*b**	Link BW: ≤ **b**
XBar ports: **2*P**	XBar ports: **N*P**	XBar ports: **P**

Evaluation

- Synthetic and real workloads (PARSEC)
- Comparing 4 router designs:
  - BAR: our BW adaptive router design
  - STANDARD: typical virtual-channel router
  - BINOC: BiNoC router
  - BWADAPTIVE: existing BW adaptive router

Area Comparison

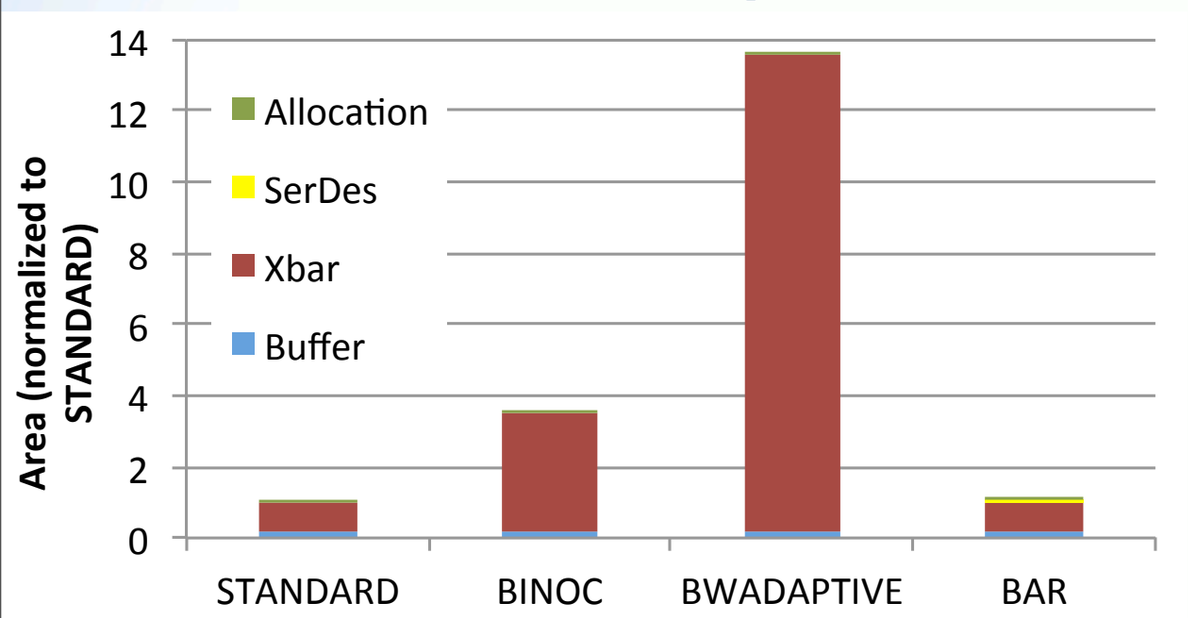
Area Comparison

- Orion 2.0 with 45nm process

Architecture	STANDARD	BINOC	BWADAPT.	BAR
Total # of Buf.	5	10	20	10
Total Channels	5-in 5-out	10-inout	20-inout	20-inout
Each Buf. Size	32 flits	16 flits	8 flits	16 flits
Total Buf. Size	160 flits	160 flits	160 flits	160 flits
Crossbar	5x5	10x10	20x20	5x5
Flit width	8 byte	8 byte	8 byte	8 byte

Area Comparison

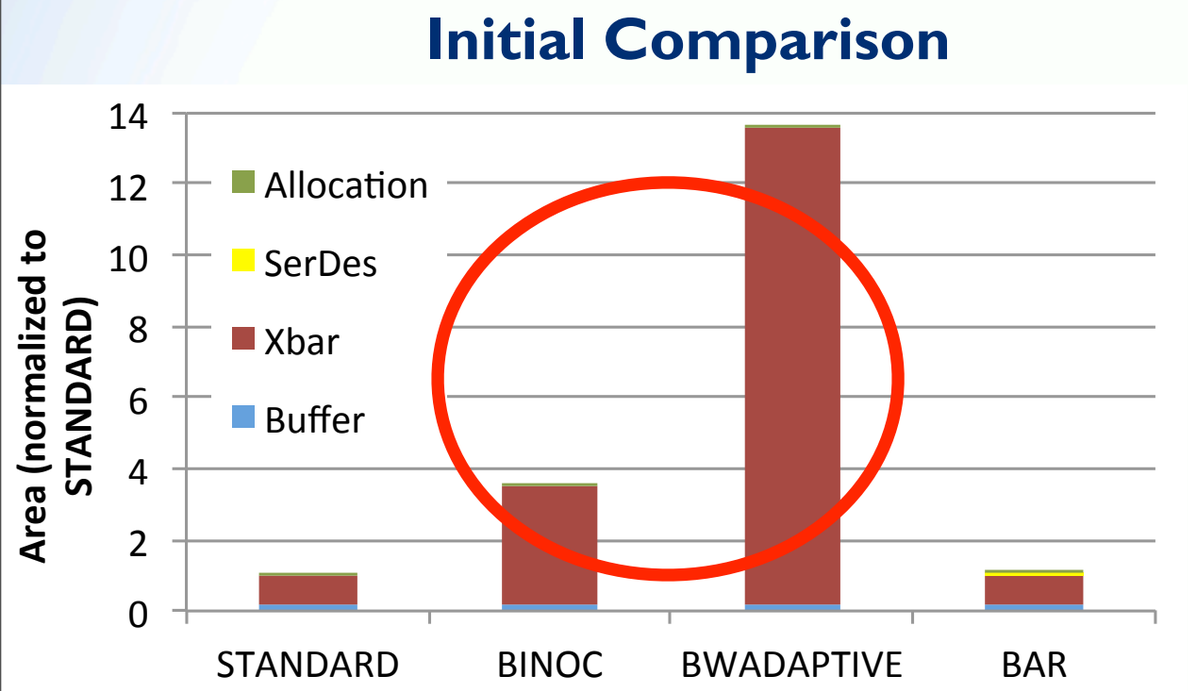
- Orion 2.0 with 45nm process



Architecture	STANDARD	BINOC	BWADAPT.	BAR
Total # of Buf.	5	10	20	10
Total Channels	5-in 5-out	10-inout	20-inout	20-inout
Each Buf. Size	32 flits	16 flits	8 flits	16 flits
Total Buf. Size	160 flits	160 flits	160 flits	160 flits
Crossbar	5x5	10x10	20x20	5x5
Flit width	8 byte	8 byte	8 byte	8 byte

Area Comparison

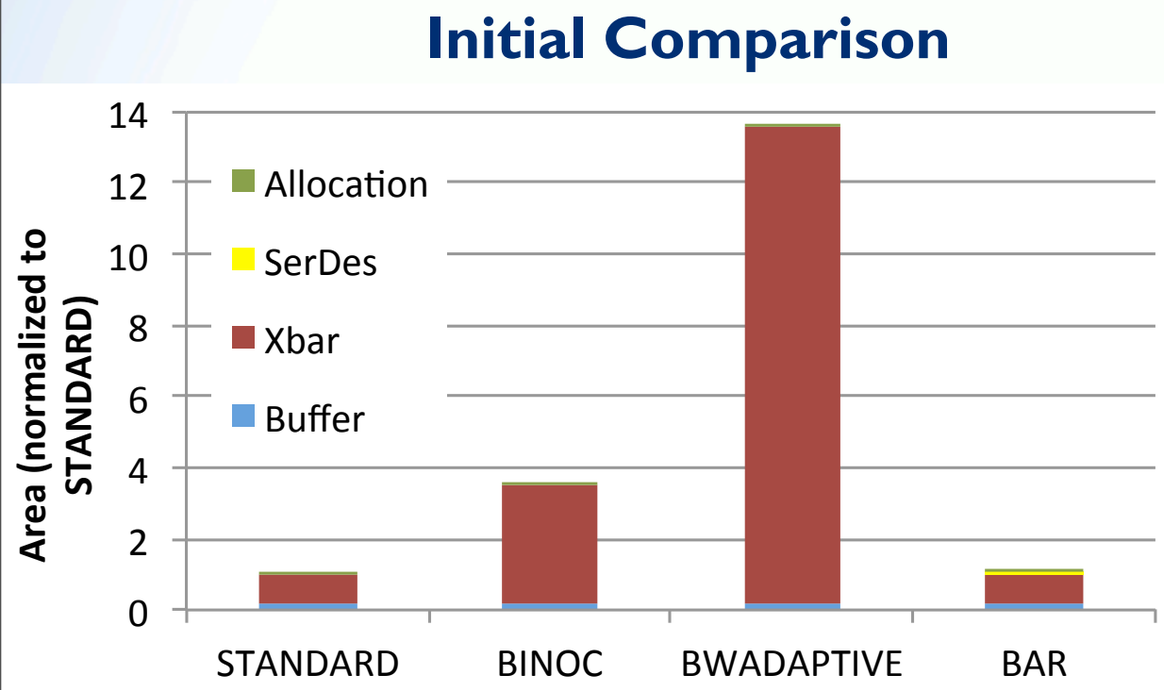
- Orion 2.0 with 45nm process

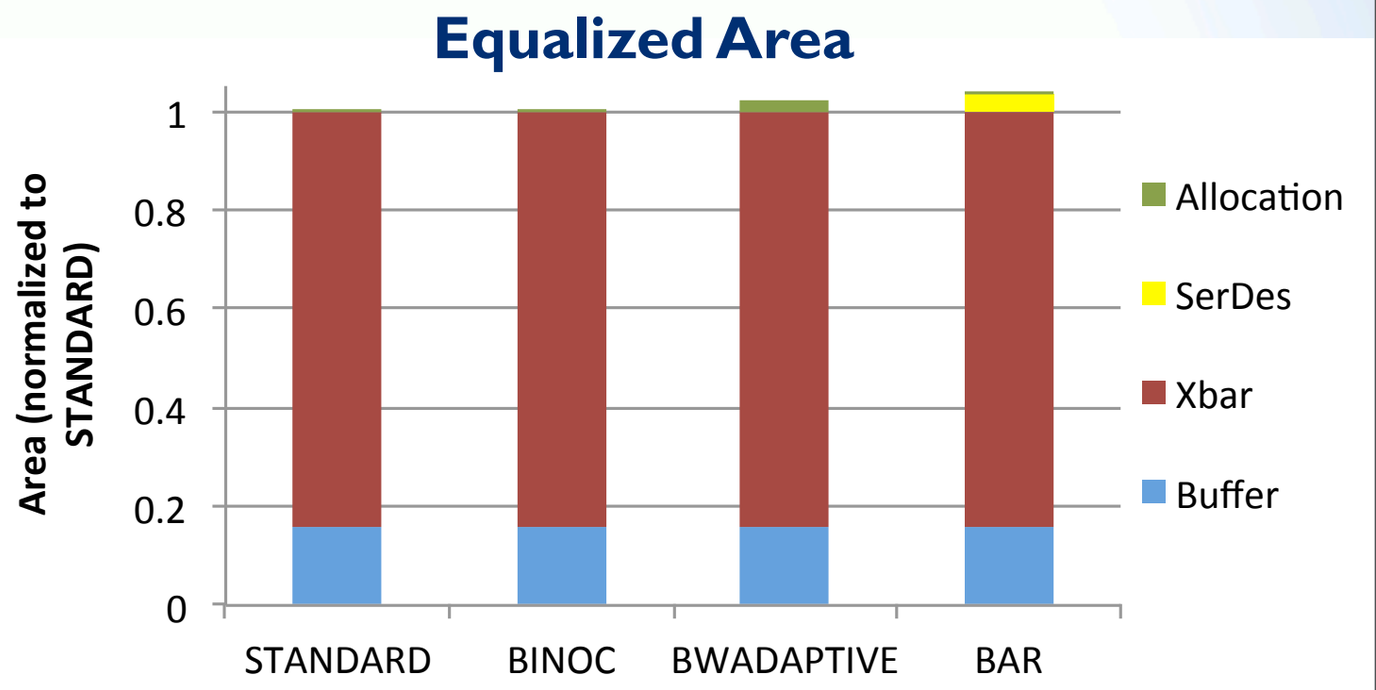


Architecture	STANDARD	BINOC	BWADAPT.	BAR
Total # of Buf.	5	10	20	10
Total Channels	5-in 5-out	10-inout	20-inout	20-inout
Each Buf. Size	32 flits	16 flits	8 flits	16 flits
Total Buf. Size	160 flits	160 flits	160 flits	160 flits
Crossbar	5x5	10x10	20x20	5x5
Flit width	8 byte	8 byte	8 byte	8 byte

Area Comparison

- Orion 2.0 with 45nm process

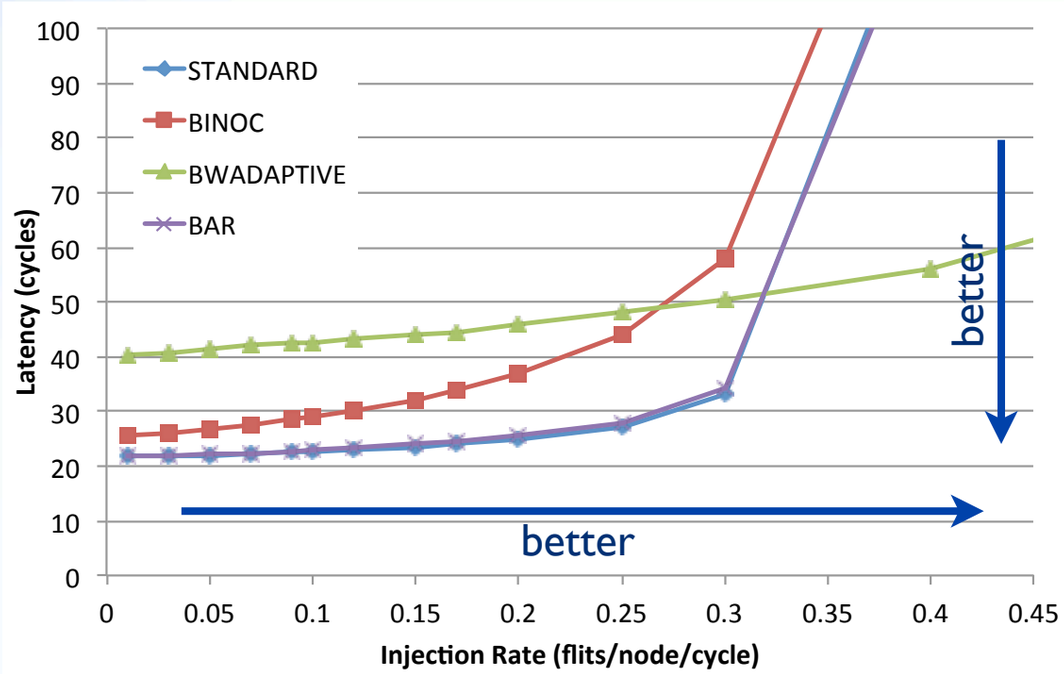




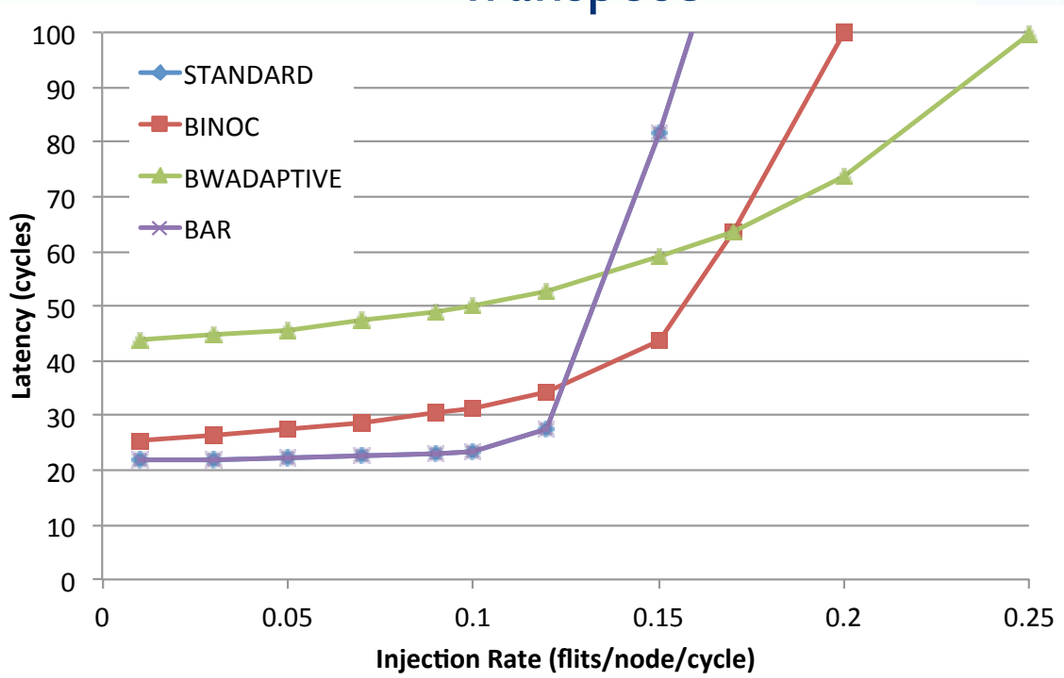
Architecture	STANDARD	BINOC	BWADAPT.	BAR
Total # of Buf.	5	10	20	10
Total Channels	5-in 5-out	10-inout	20-inout	20-inout
Each Buf. Size	32 flits	32 flits	32 flits	16 flits
Total Buf. Size	160 flits	160 flits	160 flits	160 flits
Crossbar	5x5	10x10	20x20	5x5
Flit width	8 byte	4 byte	2 byte	8 byte

Static Network Performance

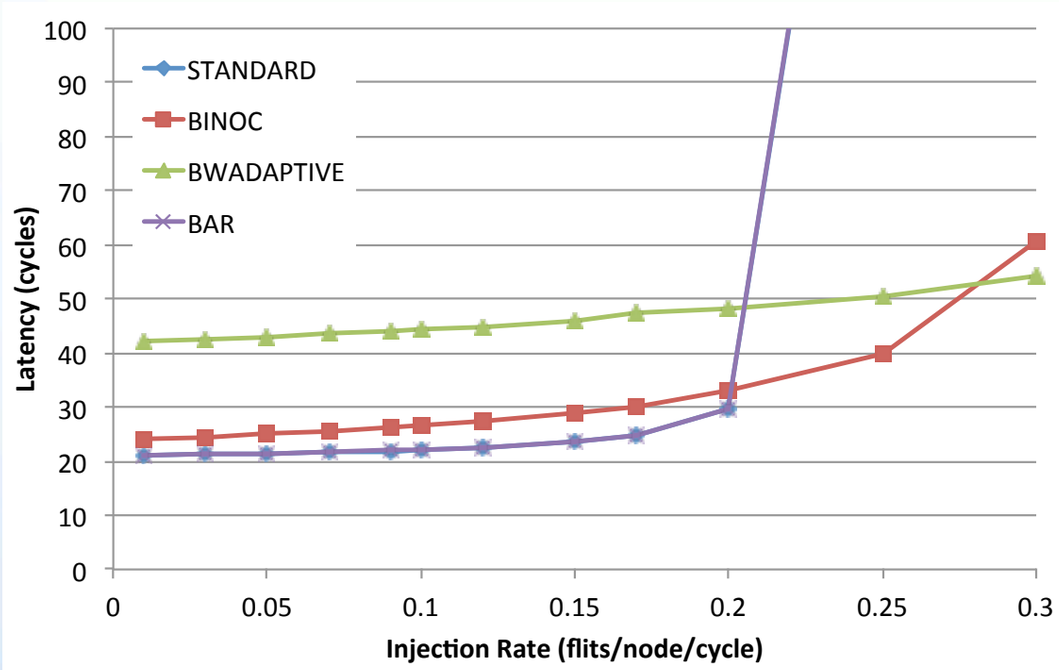
Uniform Random



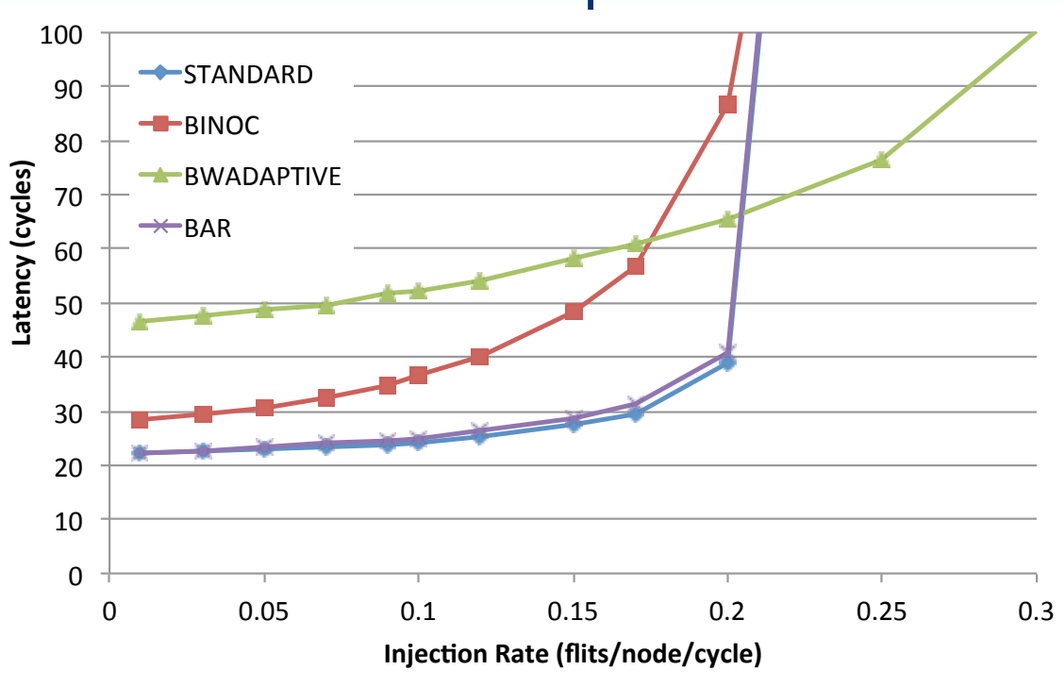
Transpose



Shuffle

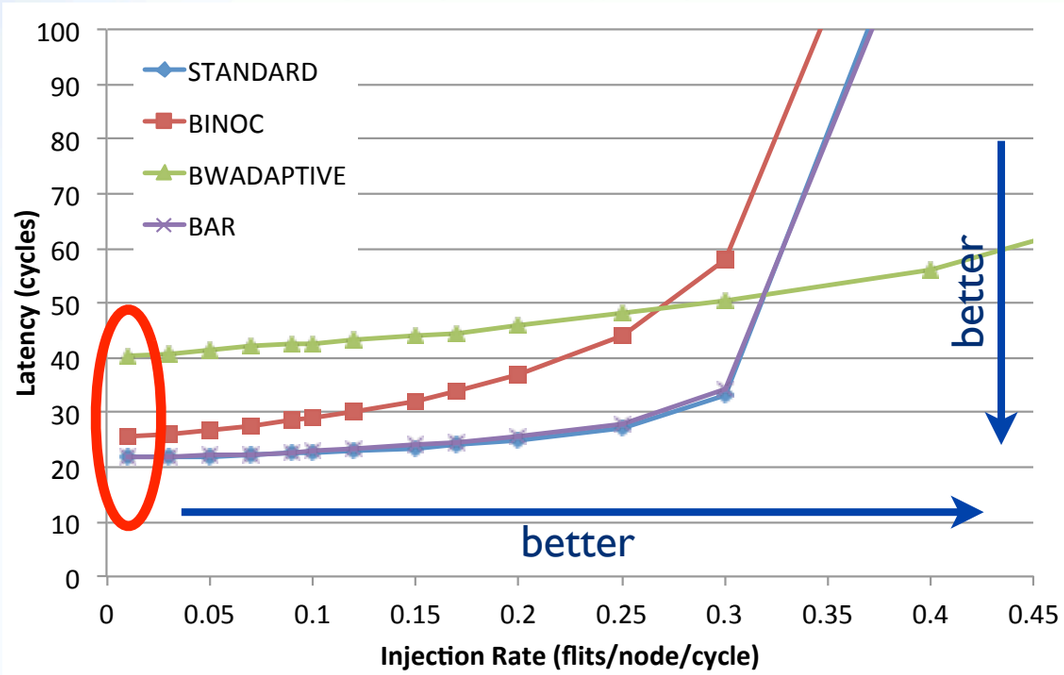


Bit-Complement

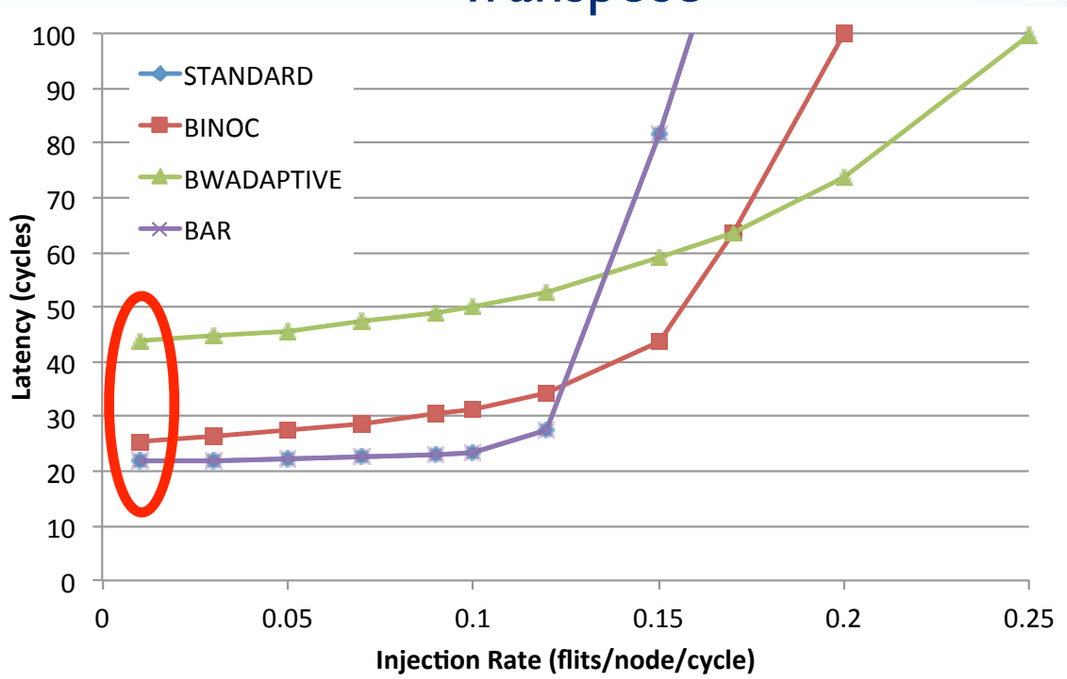


Static Network Performance

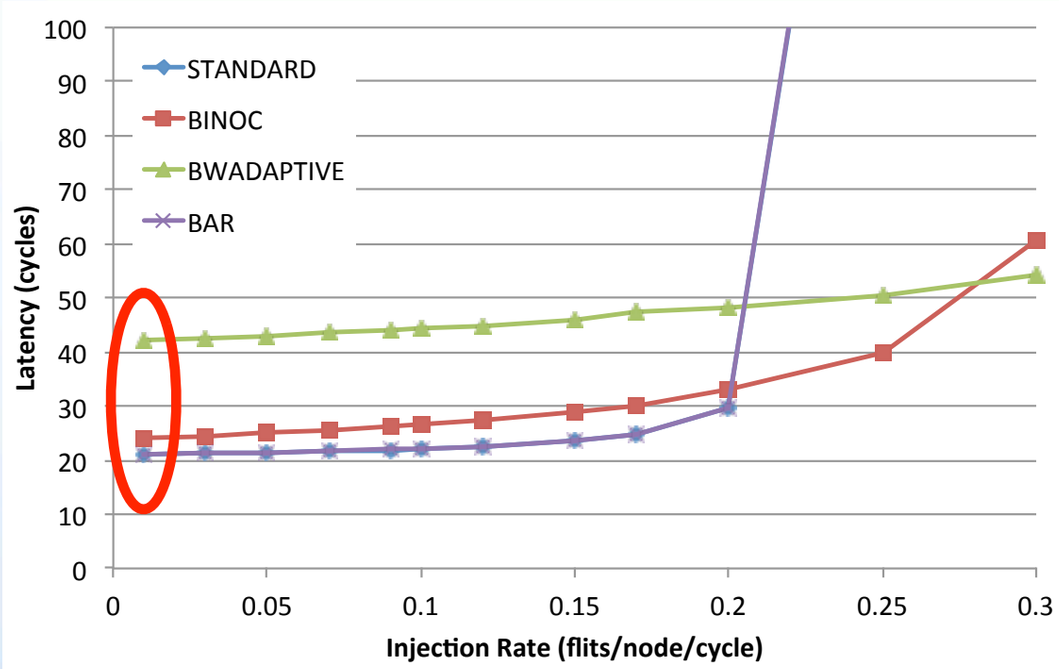
Uniform Random



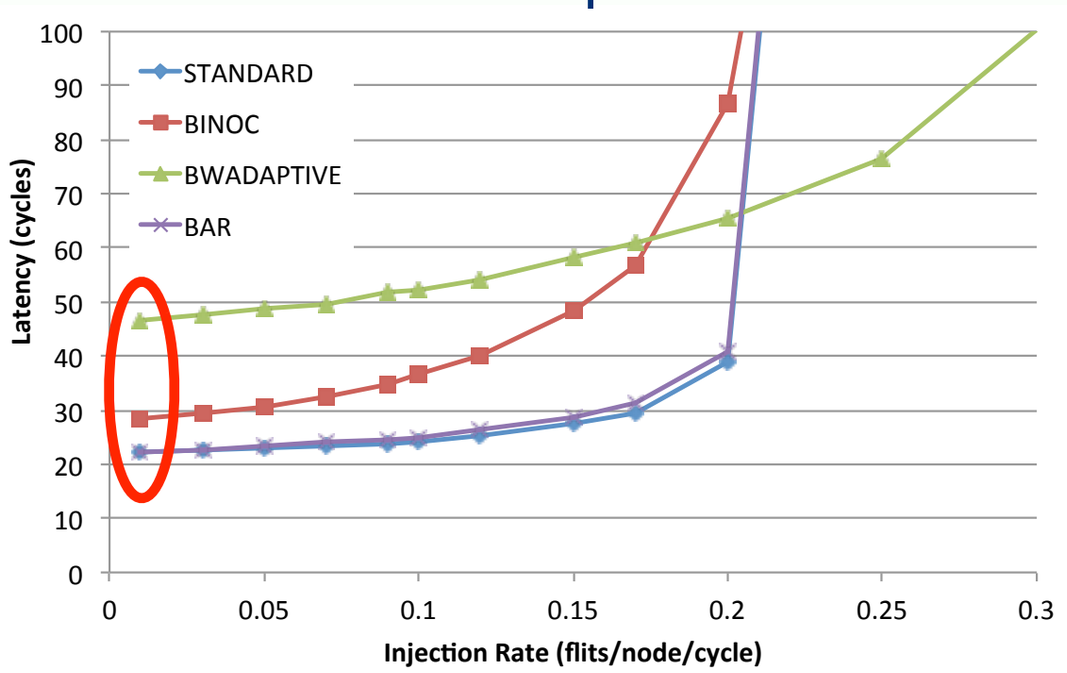
Transpose



Shuffle

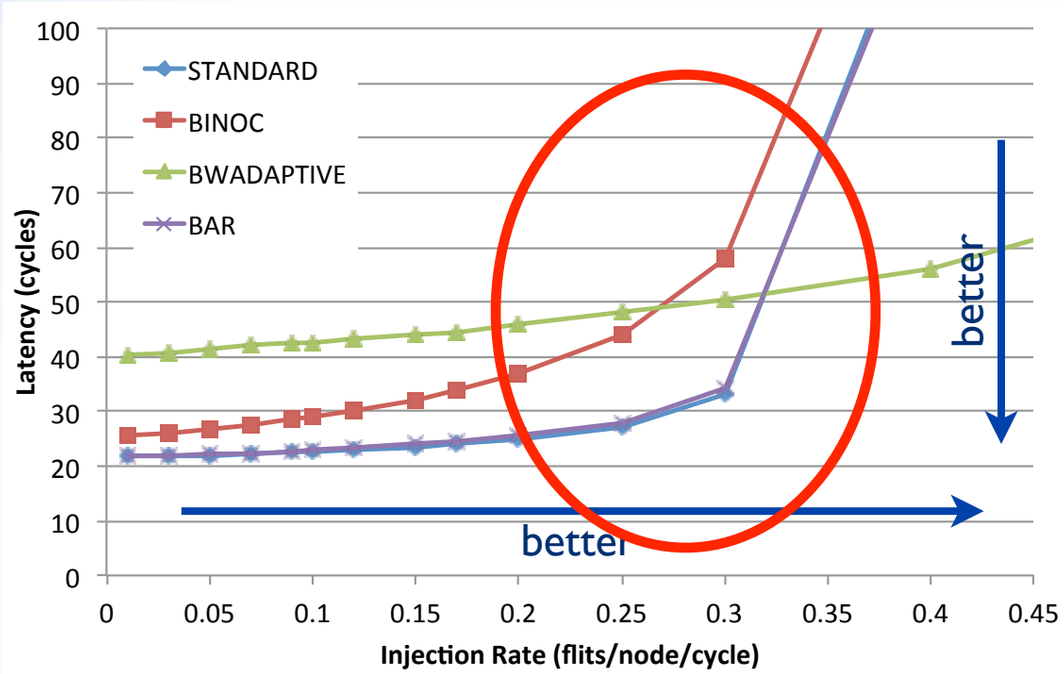


Bit-Complement

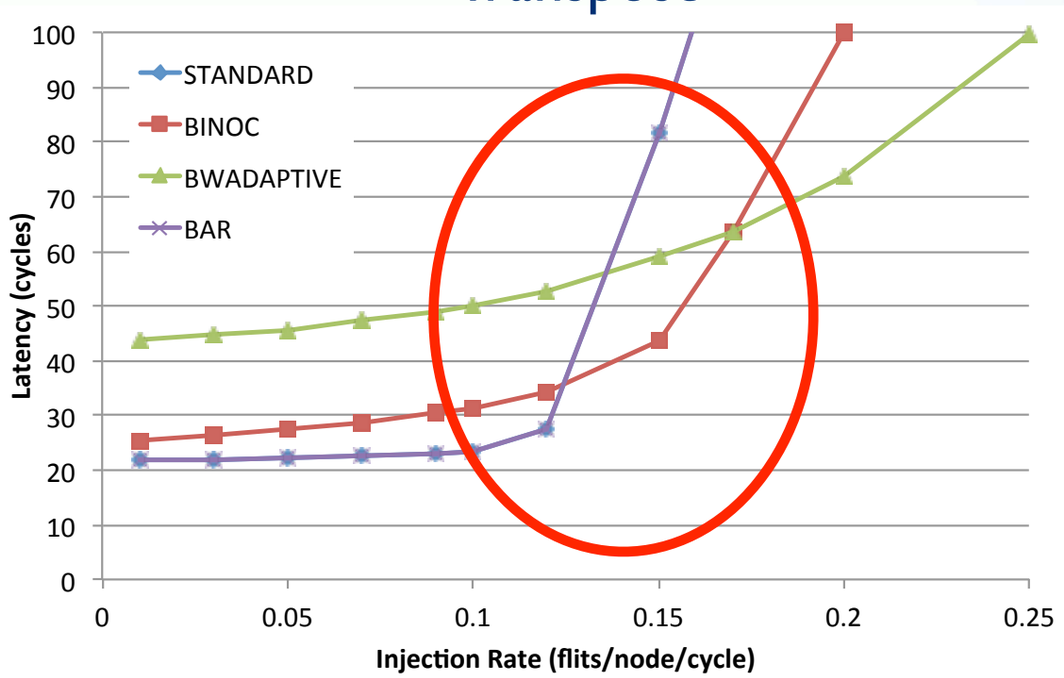


Static Network Performance

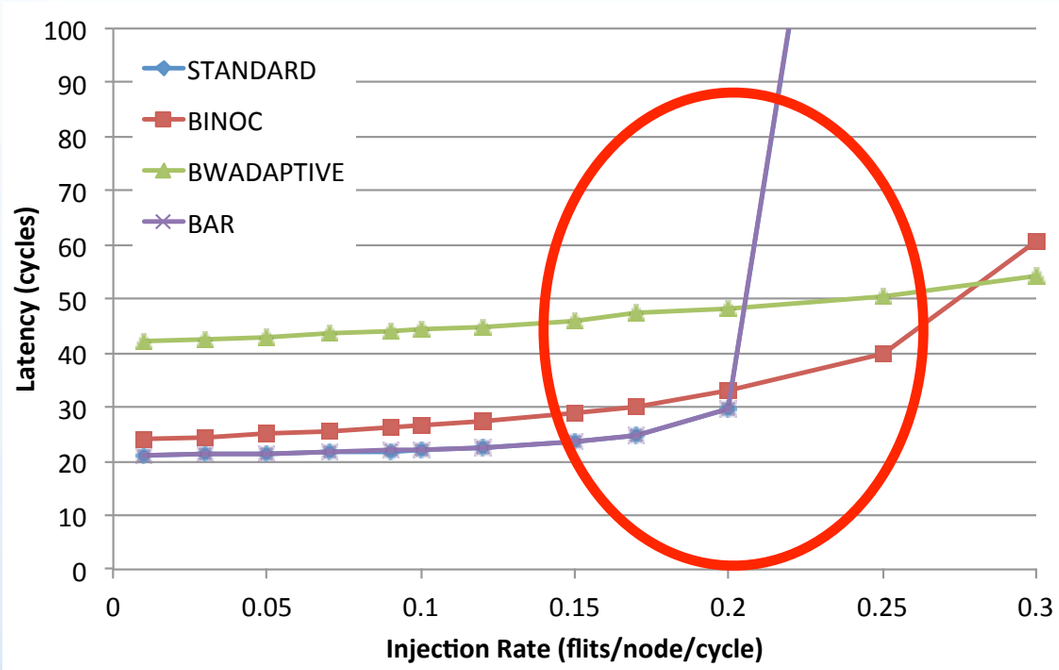
Uniform Random



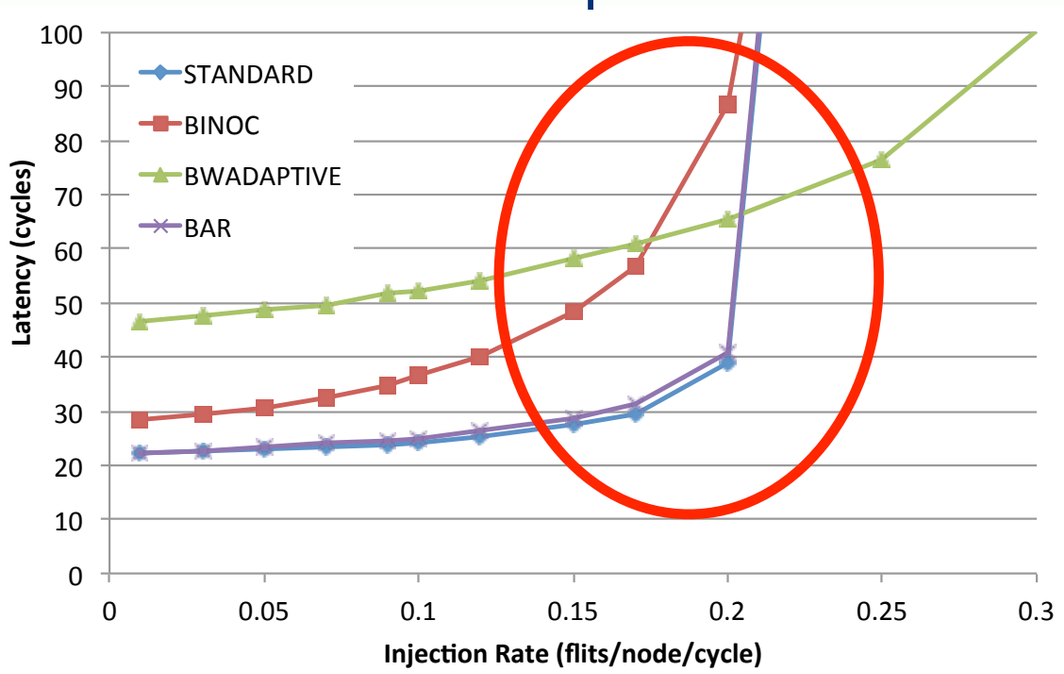
Transpose



Shuffle

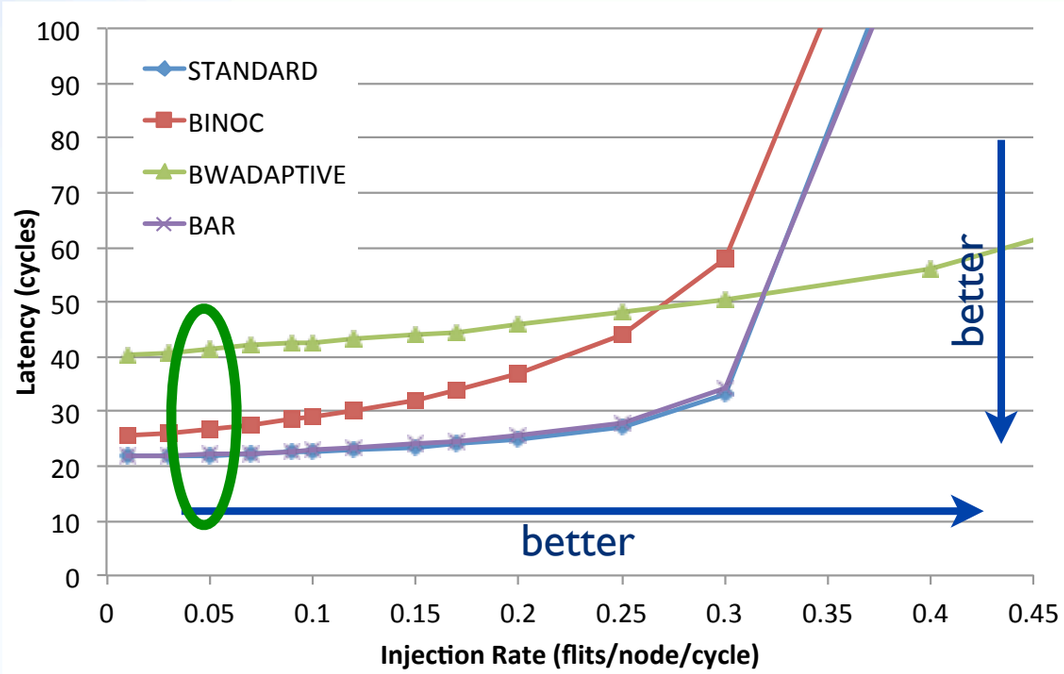


Bit-Complement

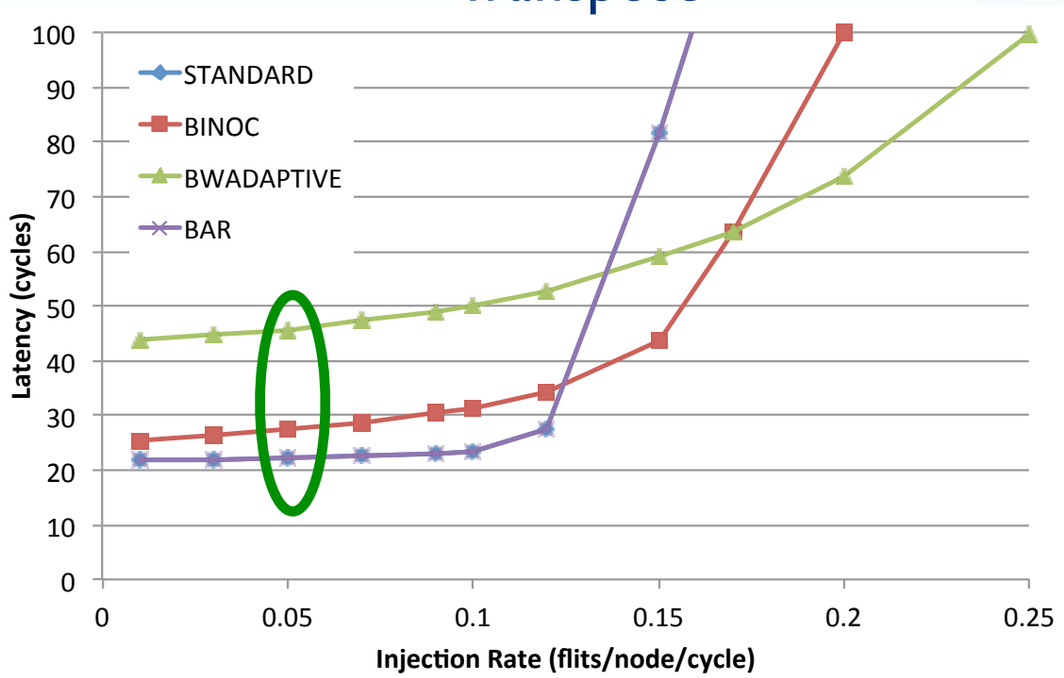


Static Network Performance

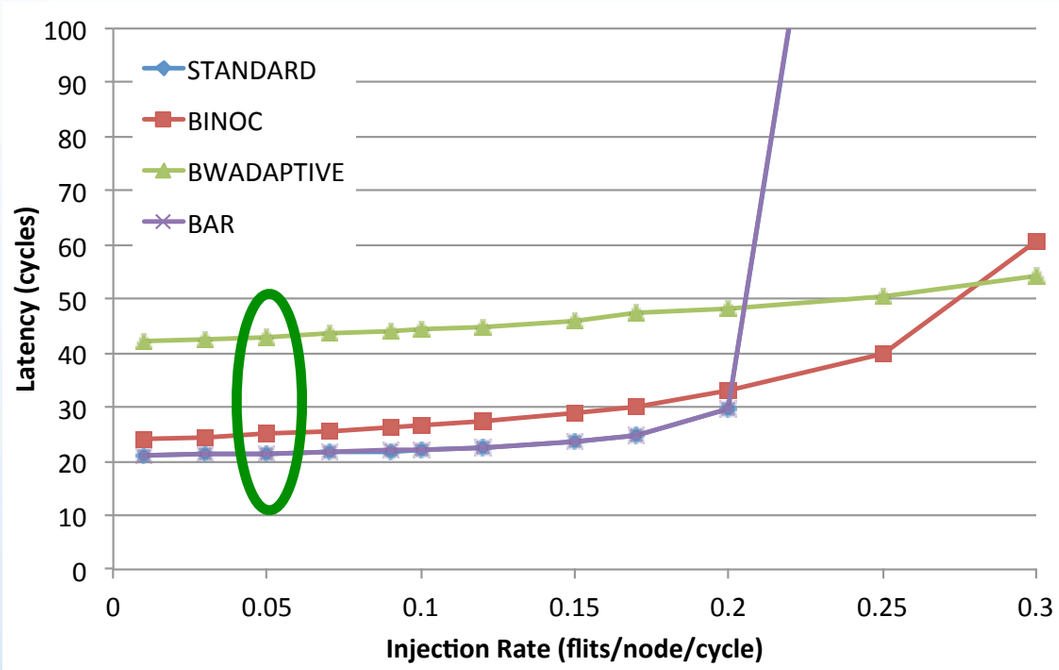
Uniform Random



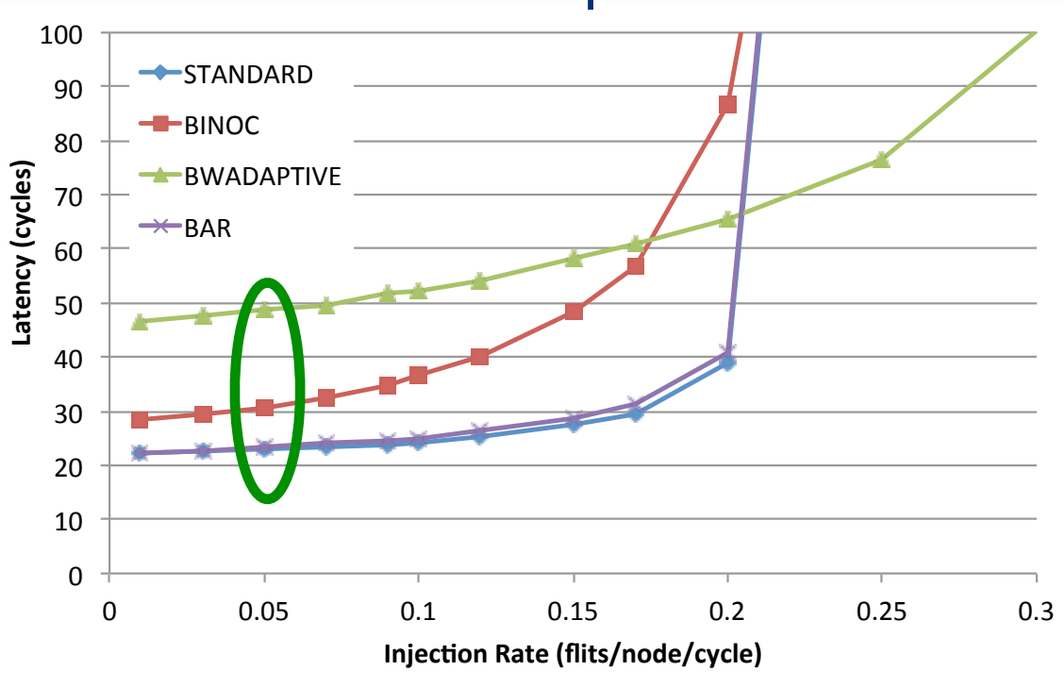
Transpose



Shuffle



Bit-Complement



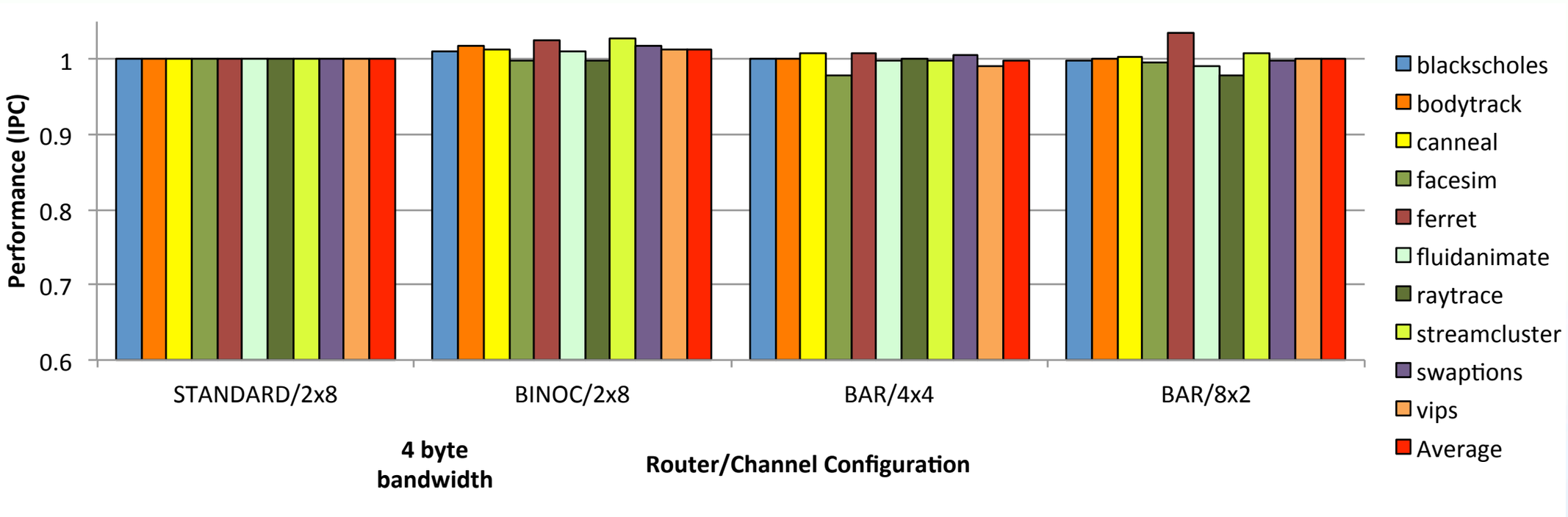
Overall System Performance

- Full-system simulation:
  - Cycle accurate x86 simulation: FeS2 + BookSim
  - PARSEC benchmarks (16 threads)
  - 16 P4-like CPUs, 4x4 mesh NoC

Overall System Performance

- Full-system simulation:
  - Cycle accurate x86 simulation: FeS2 + BookSim
  - PARSEC benchmarks (16 threads)
  - 16 P4-like CPUs, 4x4 mesh NoC

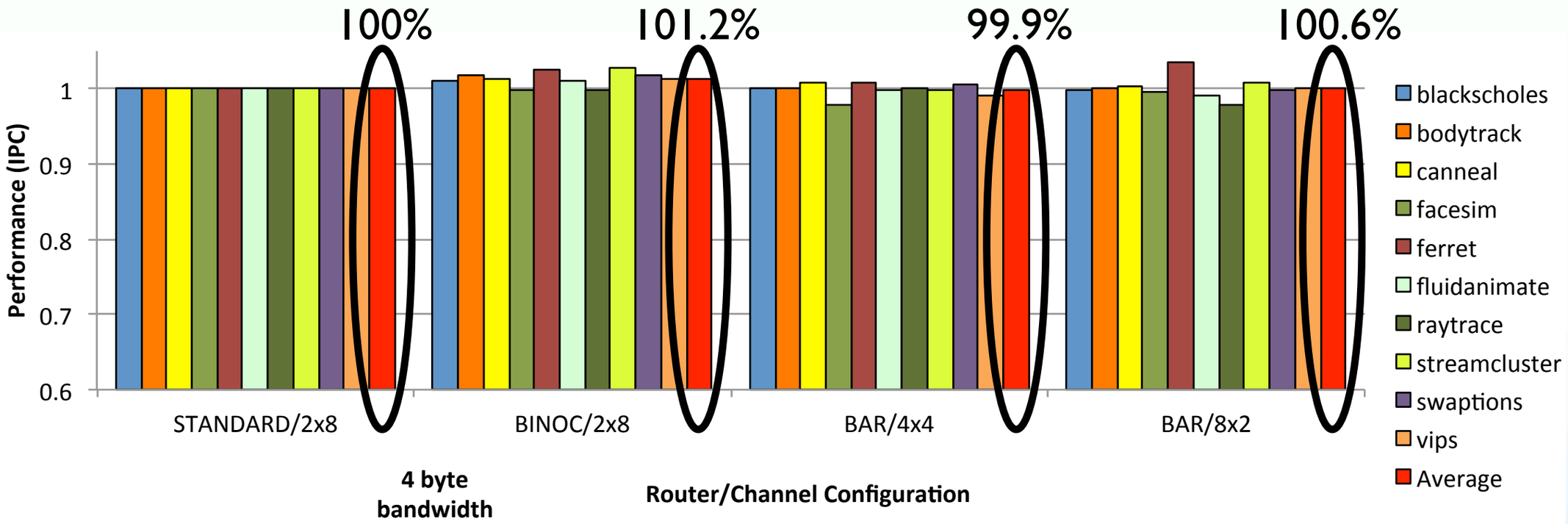
100% channel resources



Overall System Performance

- Full-system simulation:
  - Cycle accurate x86 simulation: FeS2 + BookSim
  - PARSEC benchmarks (16 threads)
  - 16 P4-like CPUs, 4x4 mesh NoC

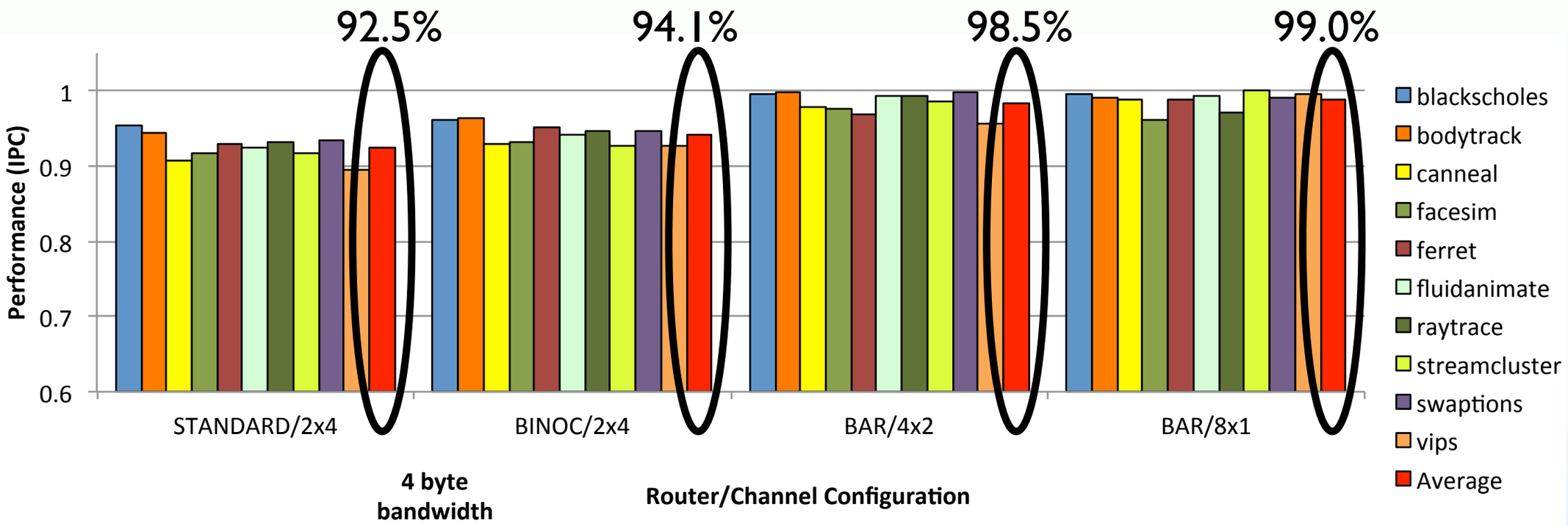
100% channel resources



Overall System Performance

- Full-system simulation:
  - Cycle accurate x86 simulation: FeS2 + BookSim
  - PARSEC benchmarks (16 threads)
  - 16 P4-like CPUs, 4x4 mesh NoC

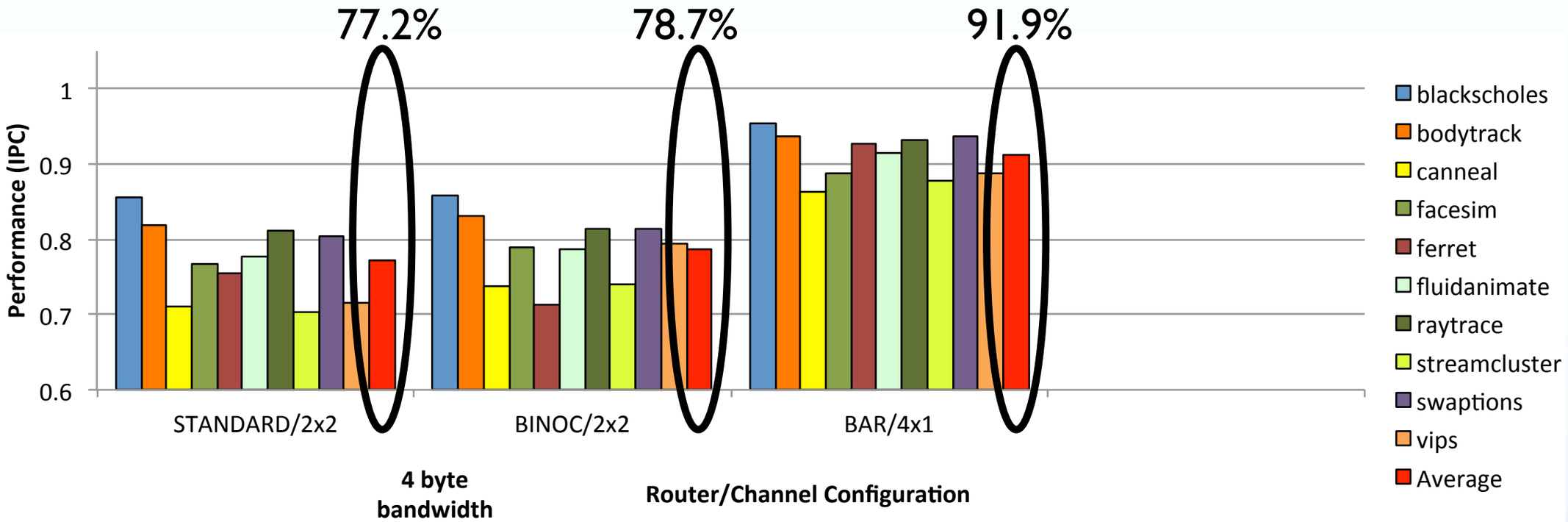
50% channel resources



Overall System Performance

- Full-system simulation:
  - Cycle accurate x86 simulation: FeS2 + BookSim
  - PARSEC benchmarks (16 threads)
  - 16 P4-like CPUs, 4x4 mesh NoC

25% channel resources



Summary

Summary

- We introduce fine-grained BW adaptivity

Summary

- We introduce fine-grained BW adaptivity
- Phit-serial communication to decouple flit width from channel width

Summary

- We introduce fine-grained BW adaptivity
- Phit-serial communication to decouple flit width from channel width
- Improved channel utilization without significantly increasing latency

Summary

- We introduce fine-grained BW adaptivity
- Phit-serial communication to decouple flit width from channel width
- Improved channel utilization without significantly increasing latency
- 50% reduction channel resources: 99% perf.

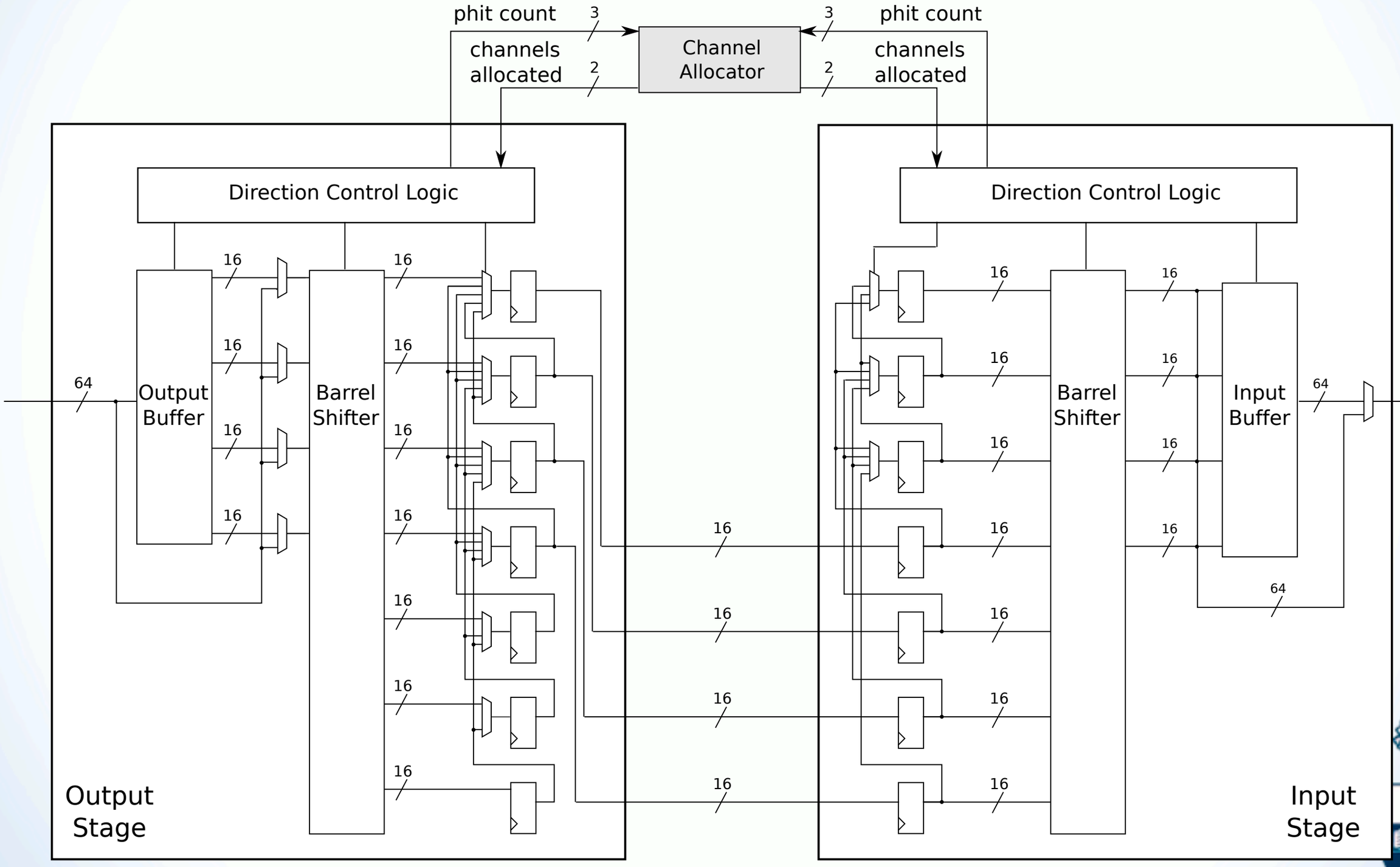
Summary

- We introduce fine-grained BW adaptivity
- Phit-serial communication to decouple flit width from channel width
- Improved channel utilization without significantly increasing latency
- 50% reduction channel resources: 99% perf.
- 75% reduction in channel resources: 92% perf.

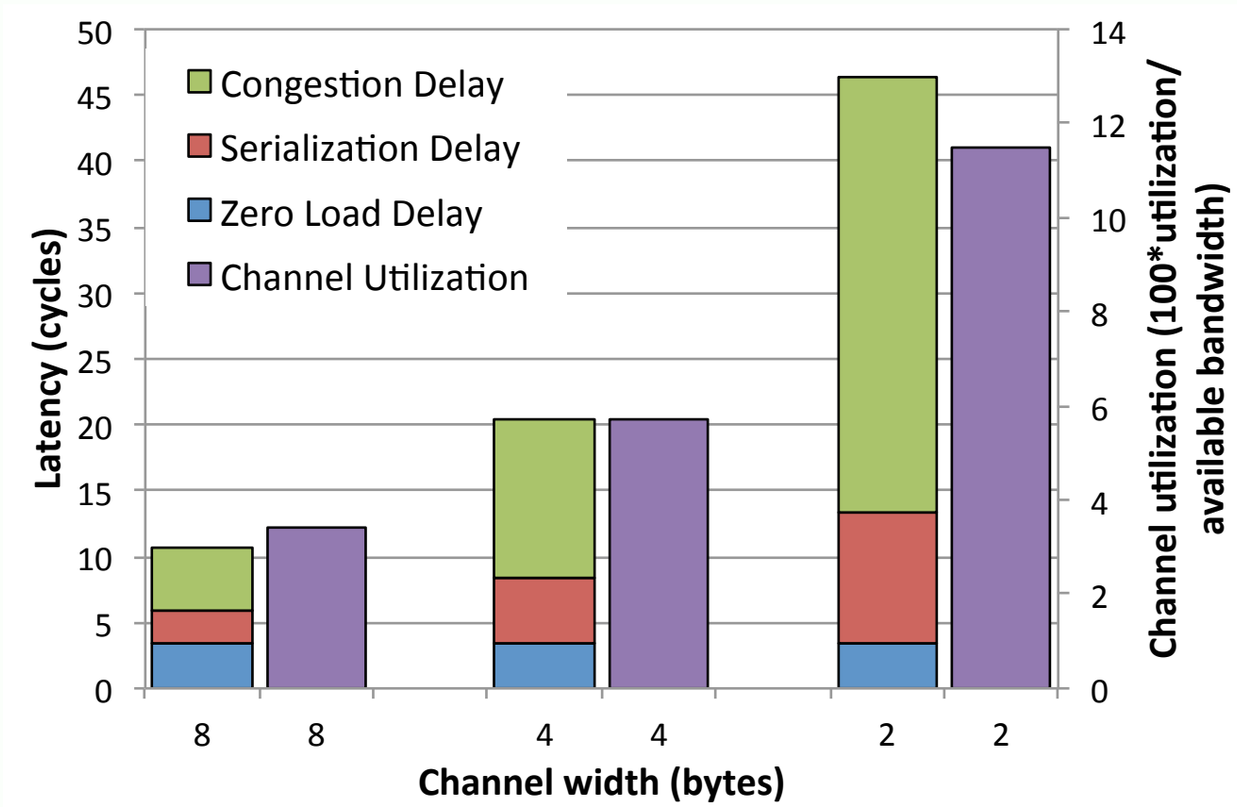
Thank you!

robert.hesse@utoronto.ca

Hardware Implementation



Additional Delays



Implementation Results

Router	Area (μm^2)	Power (mW)	Crit. path (ns)	Wires
STANDARD	316194	17.31	1.1	640
BAR	319358	17.63	1.1	345