OpenSMART: Single-cycle Multi-hop NoC Generator in BSV and Chisel

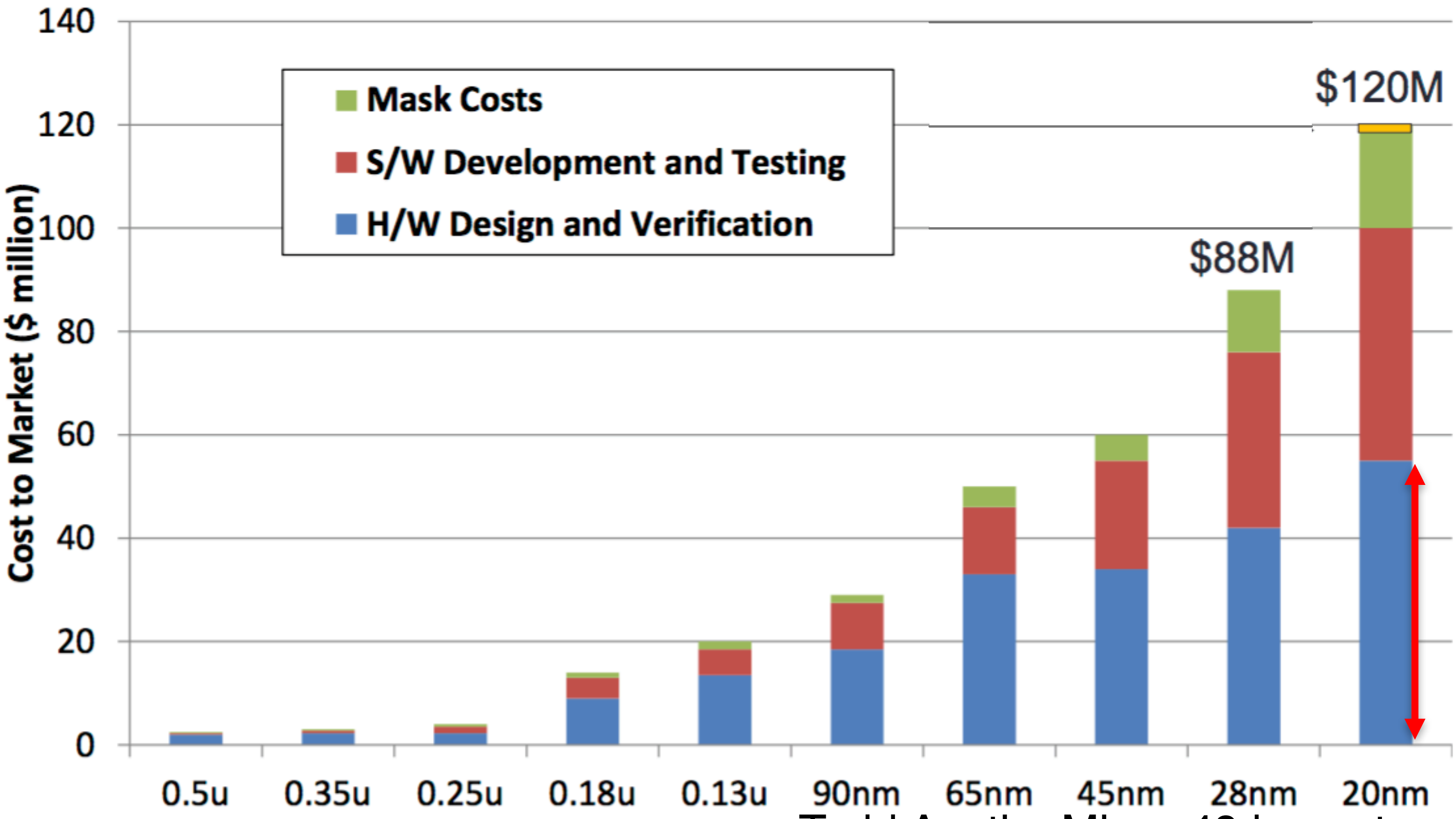
Hyoukjun Kwon and Tushar Krishna

Georgia Institute of Technology
Synergy Lab (http://synergy.ece.gatech.edu)

hyoukjun@gatech.edu

April 25, 2017

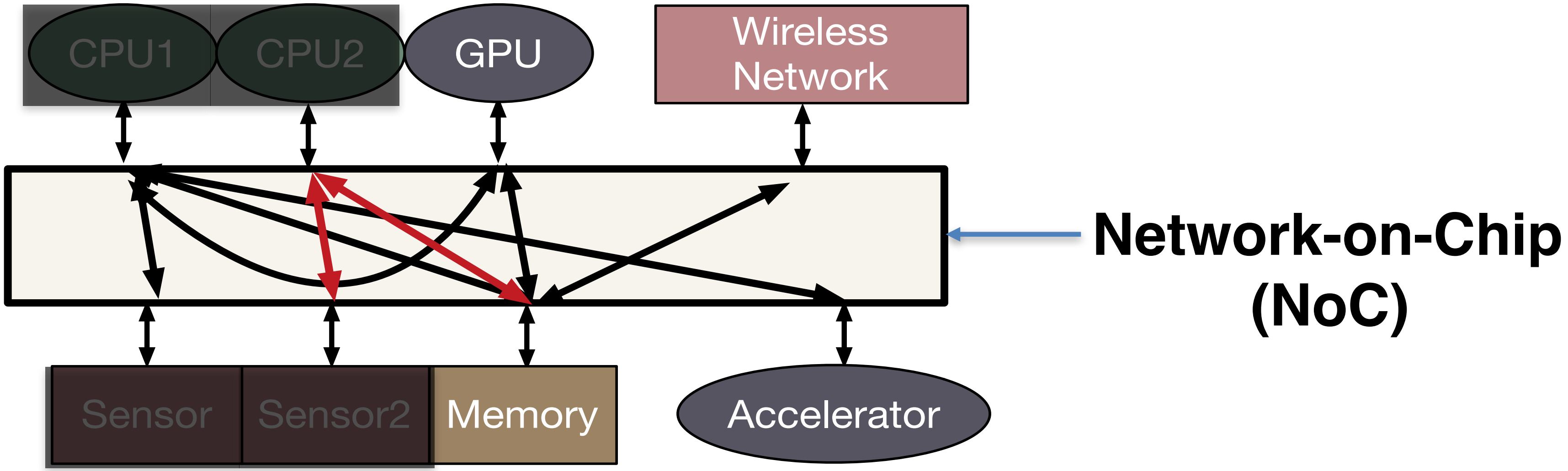
Hardware Development Cost



source: Todd Austin, Micro-49 keynote

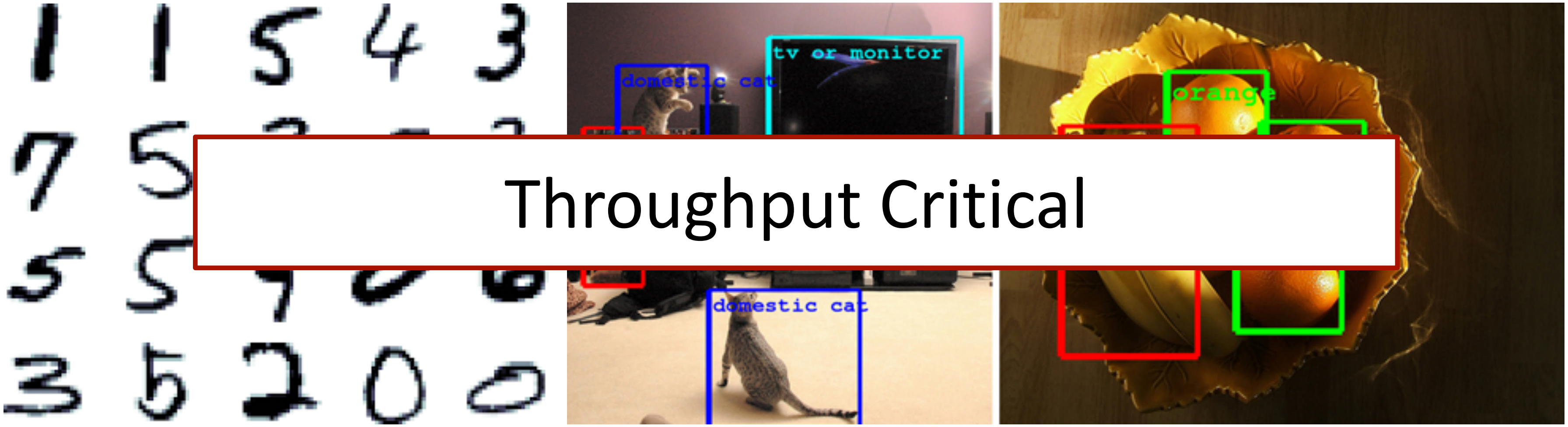
- **Low cost challenge**

Many-IP Heterogeneous System



- **Scalability challenge**
- **Flexibility challenge**

Diverse System Requirements



Throughput Critical

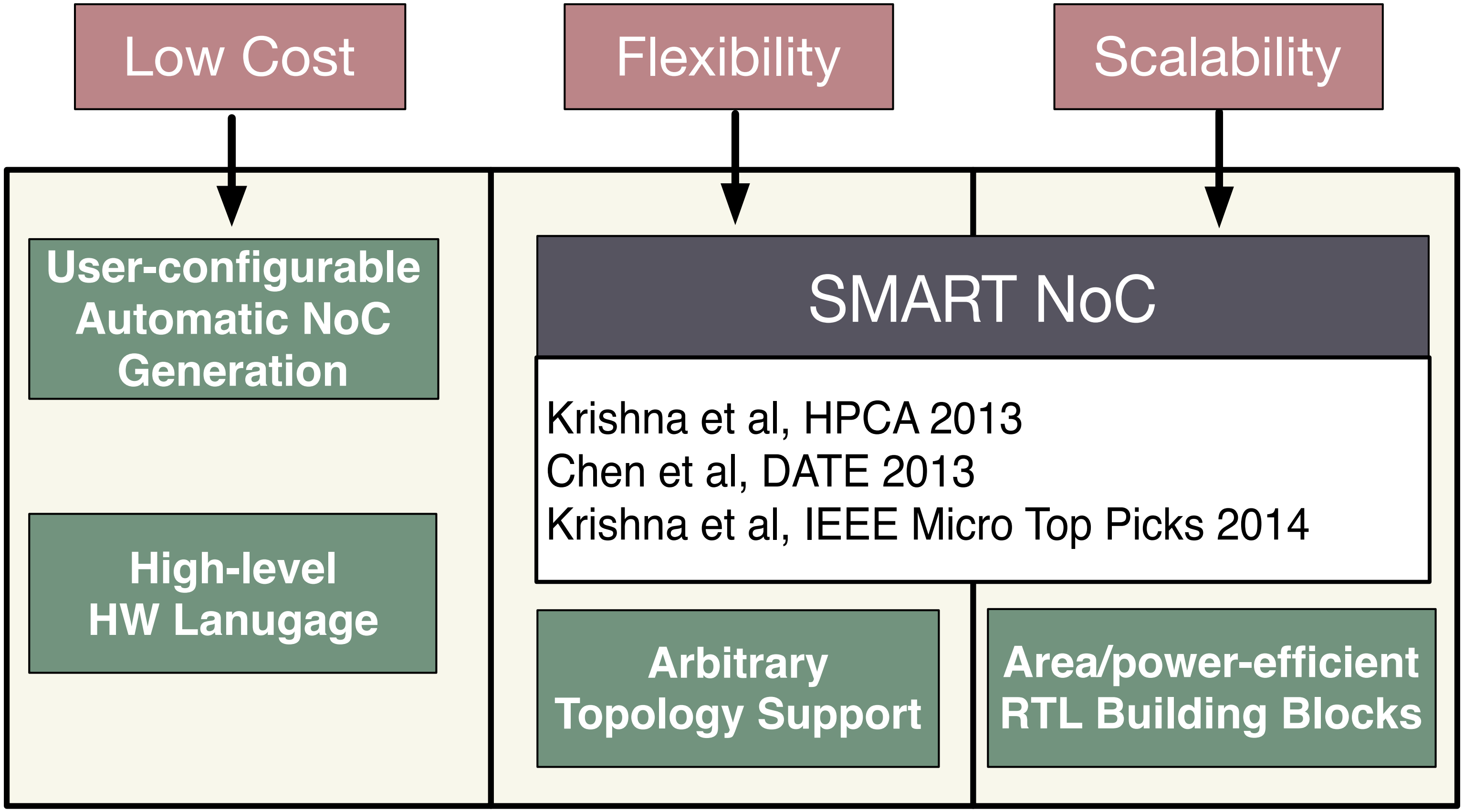
Latency Critical

source: MNIST, Engadget, TheStack

Challenges for NoCs

- **Low-cost**
  - Low design/verification costs of custom/generic NoCs
  - Design automation of high-performance, low-energy NoCs
- **Scalability**
  - Many-IP heterogeneous system support
  - Low latency
  - Low energy
  - Low area
- **Flexibility**
  - Diverse connectivity
  - Diverse latency/throughput requirements

OpenSMART


Krishna et al, HPCA 2013
Chen et al, DATE 2013
Krishna et al, IEEE Micro Top Picks 2014

Low Cost
Flexibility
Scalability
User-configurable Automatic NoC Generation
High-level HW Lanugage
SMART NoC
Arbitrary Topology Support
Area/power-efficient RTL Building Blocks

OpenSMART

Outline

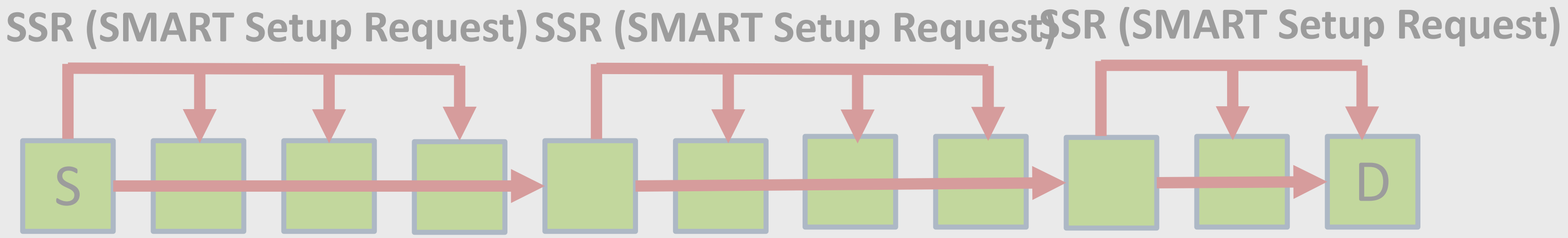
- **Motivation: Scalable, Flexible, and Low-cost NoCs**
- **Background: SMART NoCs**
- **OpenSMART**
  - Design Flow
  - Building Blocks
  - Walk-through Examples
- **Case Studies**
  - Mesh vs. SMART
  - High-radix vs. Low-radix
- **Conclusions**

Outline

- Motivation: Scalable, Flexible, and Low-cost NoCs
- **Background: SMART NoCs**
- OpenSMART
  - Design Flow
  - Building Blocks
  - Walk-through Examples
- Case Studies
  - Mesh vs. SMART
  - High-radix vs. Low-radix
- Conclusions

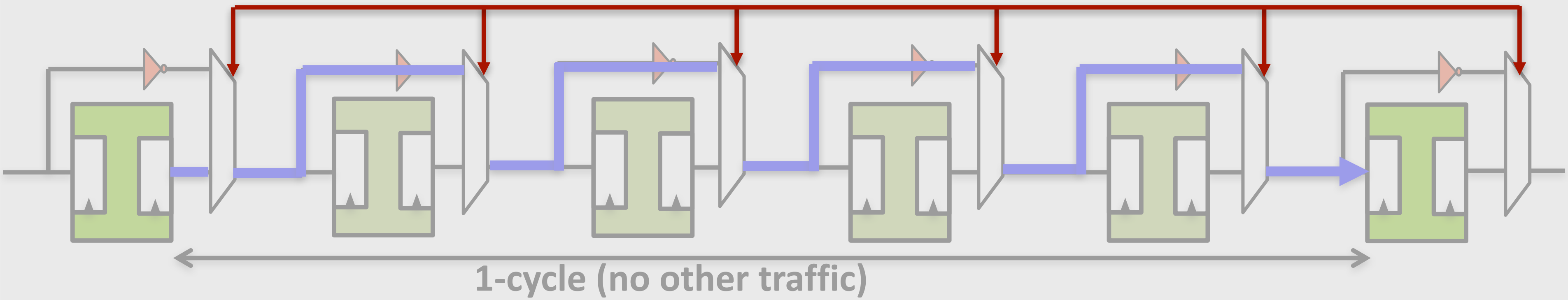
SMART NoC

- **Single-cycle Multi-hop Asynchronous Repeated Traversal**


SSR (SMART Setup Request)
SSR (SMART Setup Request)
SSR (SMART Setup Request)
S
D

SMART: achieves the performance of ***dedicated*** connections over a network of ***shared*** links

Krishna et al, IEEE Micro Top Picks 2014,

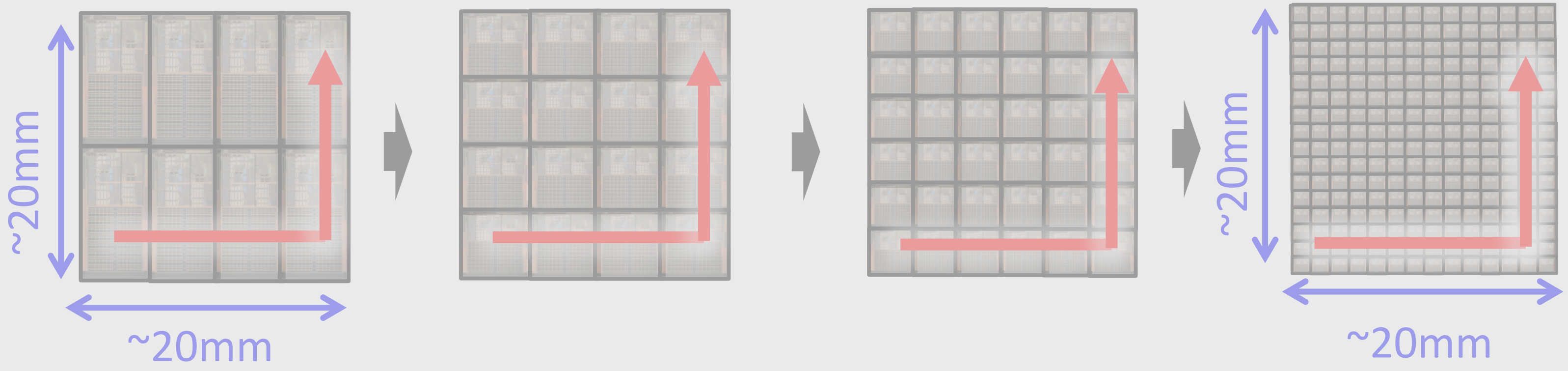


Is 1-cycle Network Possible?

Is wire fast enough to support 1-cycle network? Yes

- Wire traversal length within 1ns (1Ghz): **10-16mm** ☺
- Wire delay over technology: **constant** ☹
- Chip dimension: **remain similar (~20mm)** ☺

On-chip wires are fast enough to transmit across the chip **within 1-2 cycles** at 1GHz even if the technology scales



Features of SMART

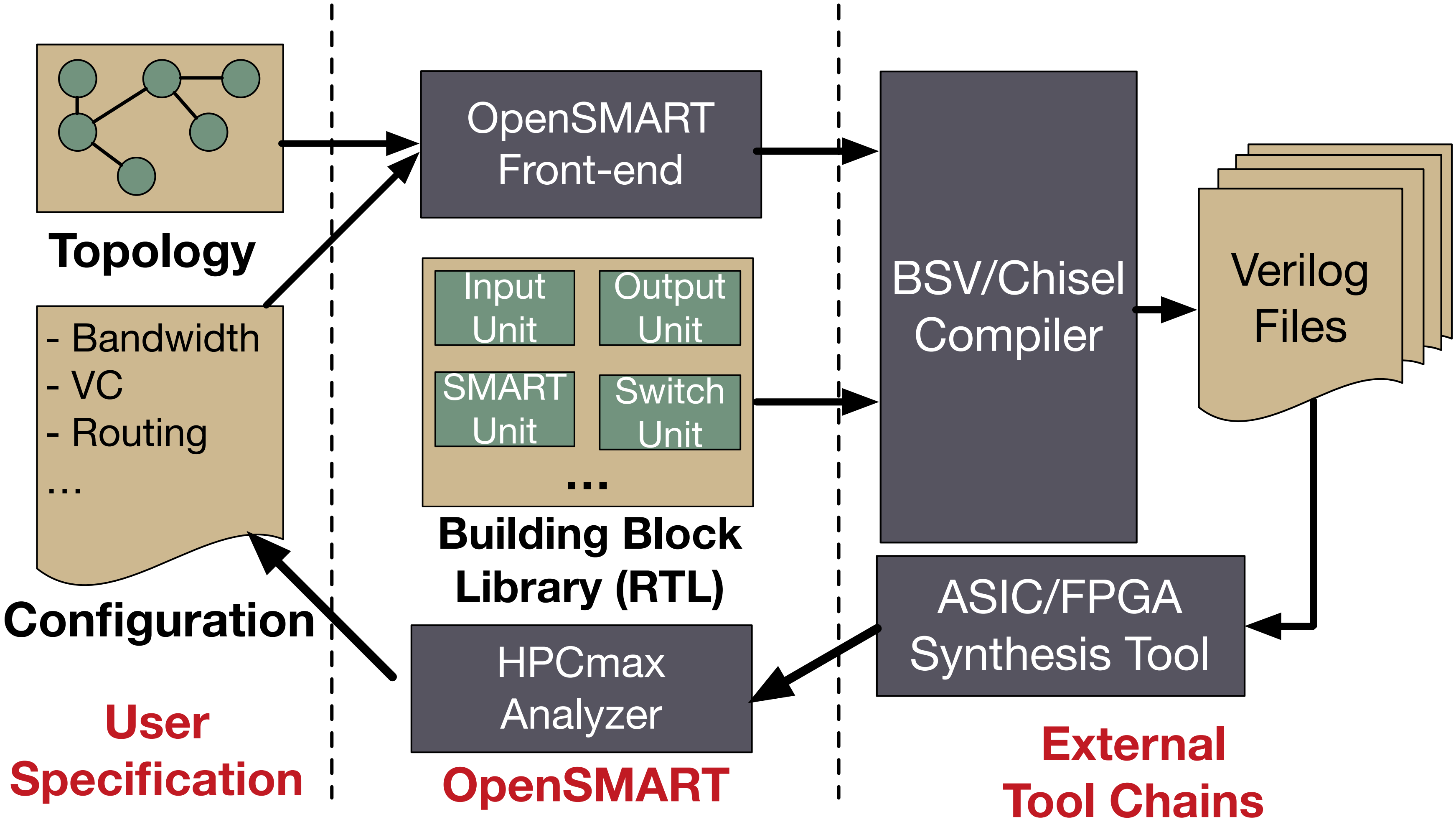
- **Low latency network**
  - **Dynamic** bypass of intermediate routers **between any two routers**
  - Limit: **HPCmax** (hops per cycle max), *maximum number of "hops" that the underlying wire allows the flit to traverse within a clock cycle*

- **Separate control path**
  - HPCmax bits from every router along each direction
  - Arbitration of multiple bypass requests on the same link
  - **No ACK** required

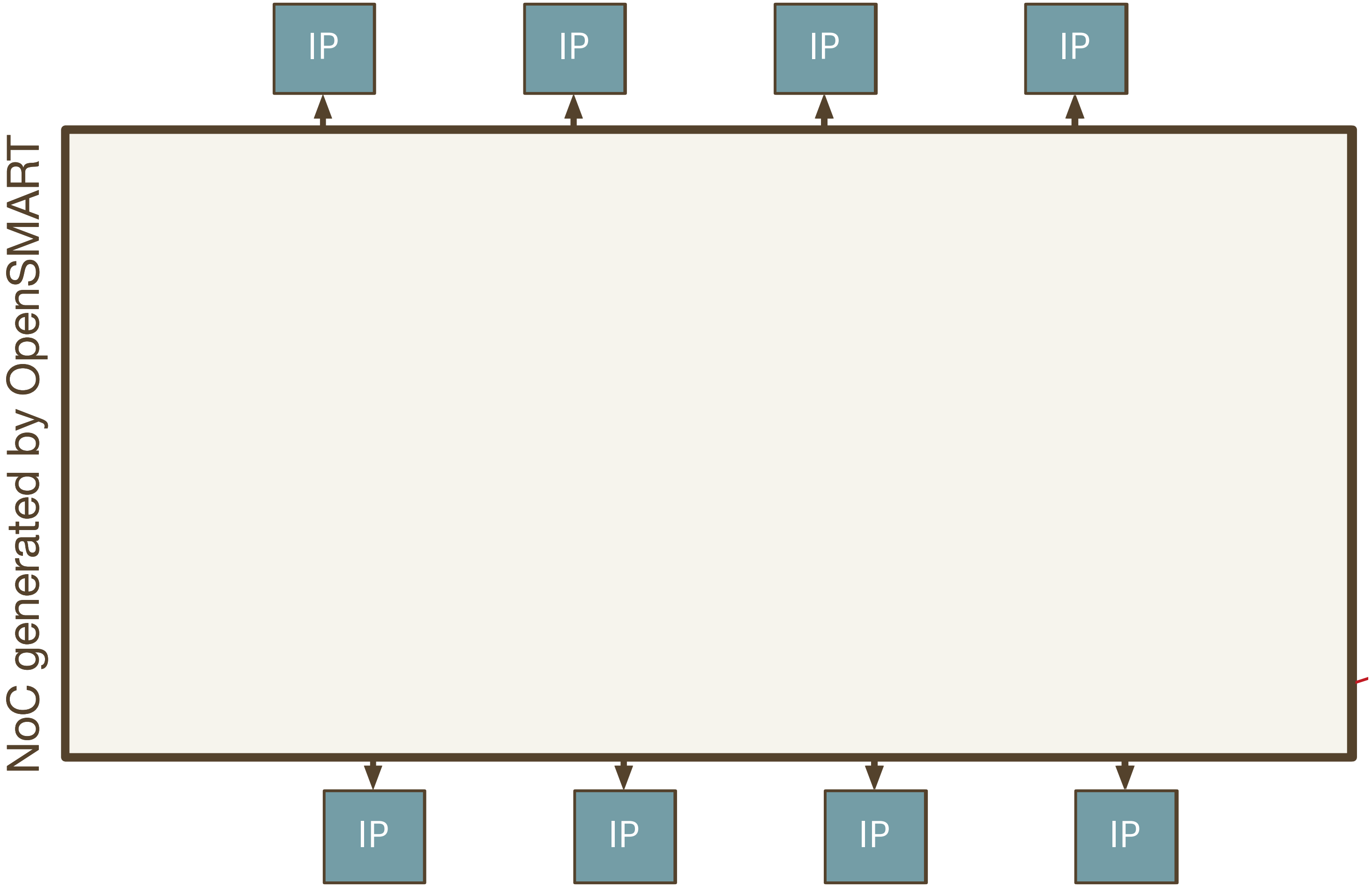
Outline

- Motivation: Scalable, Flexible, and Low-cost NoCs
- Background: SMART NoCs
- **OpenSMART**
  - **Design Flow**
  - Building Blocks
  - Walk-through Examples
- Case Studies
  - Mesh vs. SMART
  - High-radix vs. Low-radix
- Conclusions

OpenSMART Design Flow



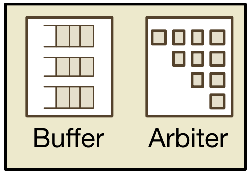
OpenSMART NoC



Outline

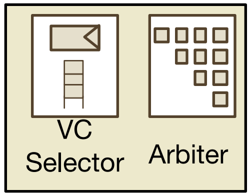
- Motivation: Scalable, Flexible, and Low-cost NoCs
- Background: SMART NoCs
- **OpenSMART**
  - Design Flow
  - **Building Blocks**
  - Walk-through Examples
- Case Studies
  - Mesh vs. SMART
  - High-radix vs. Low-radix
- Conclusions

OpenSMART Building Blocks



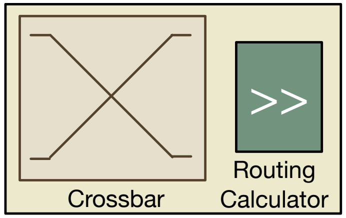
Input Unit

input buffer + input VC arbitration



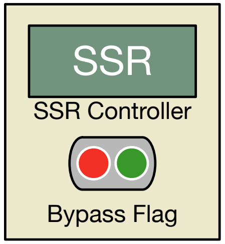
Output Unit

output VC selection + output port arbitration + credit management



Switch Unit

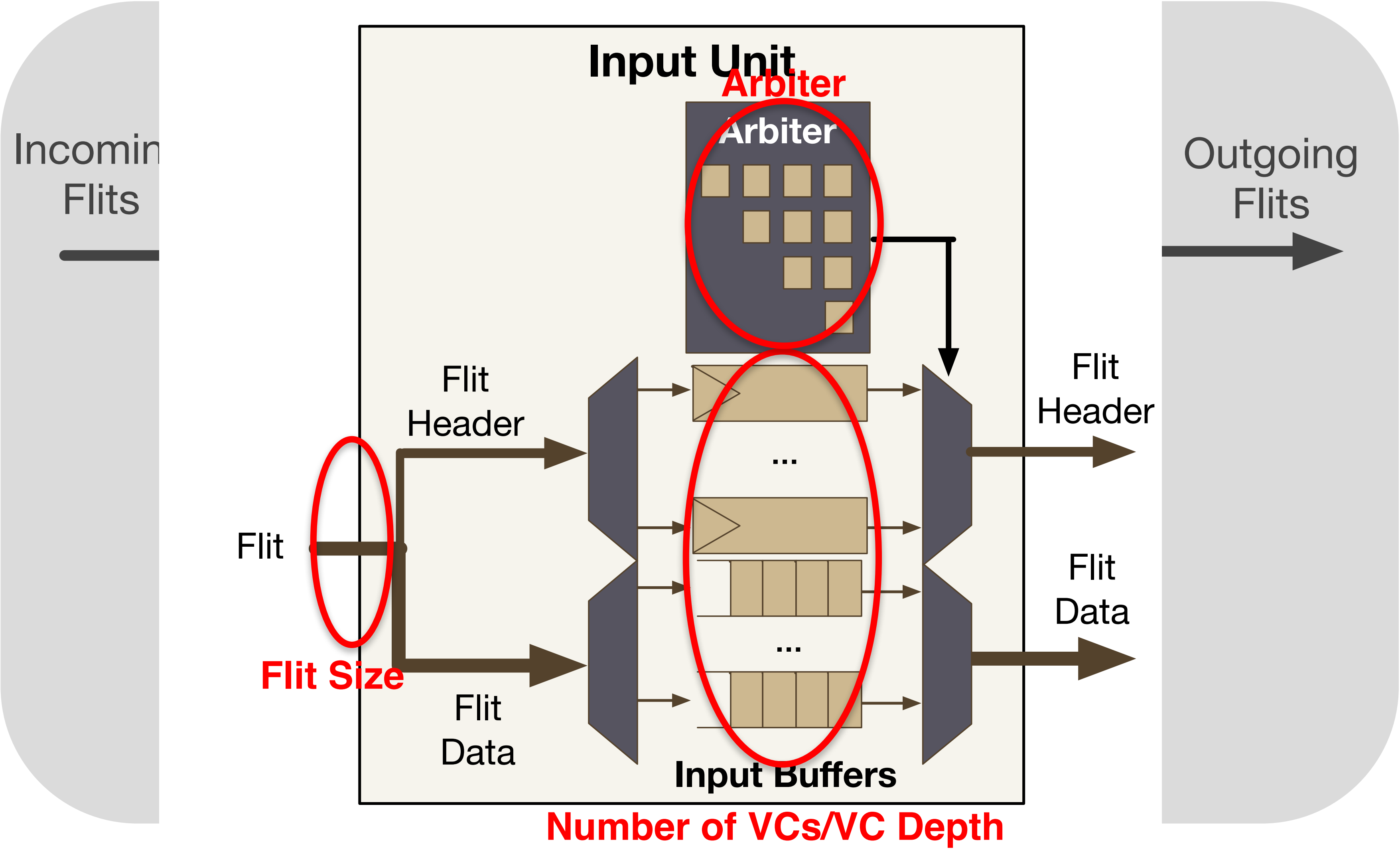
switching (via crossbar) + routing calculation



SMART Unit

SSR communication & arbitration + bypass flag

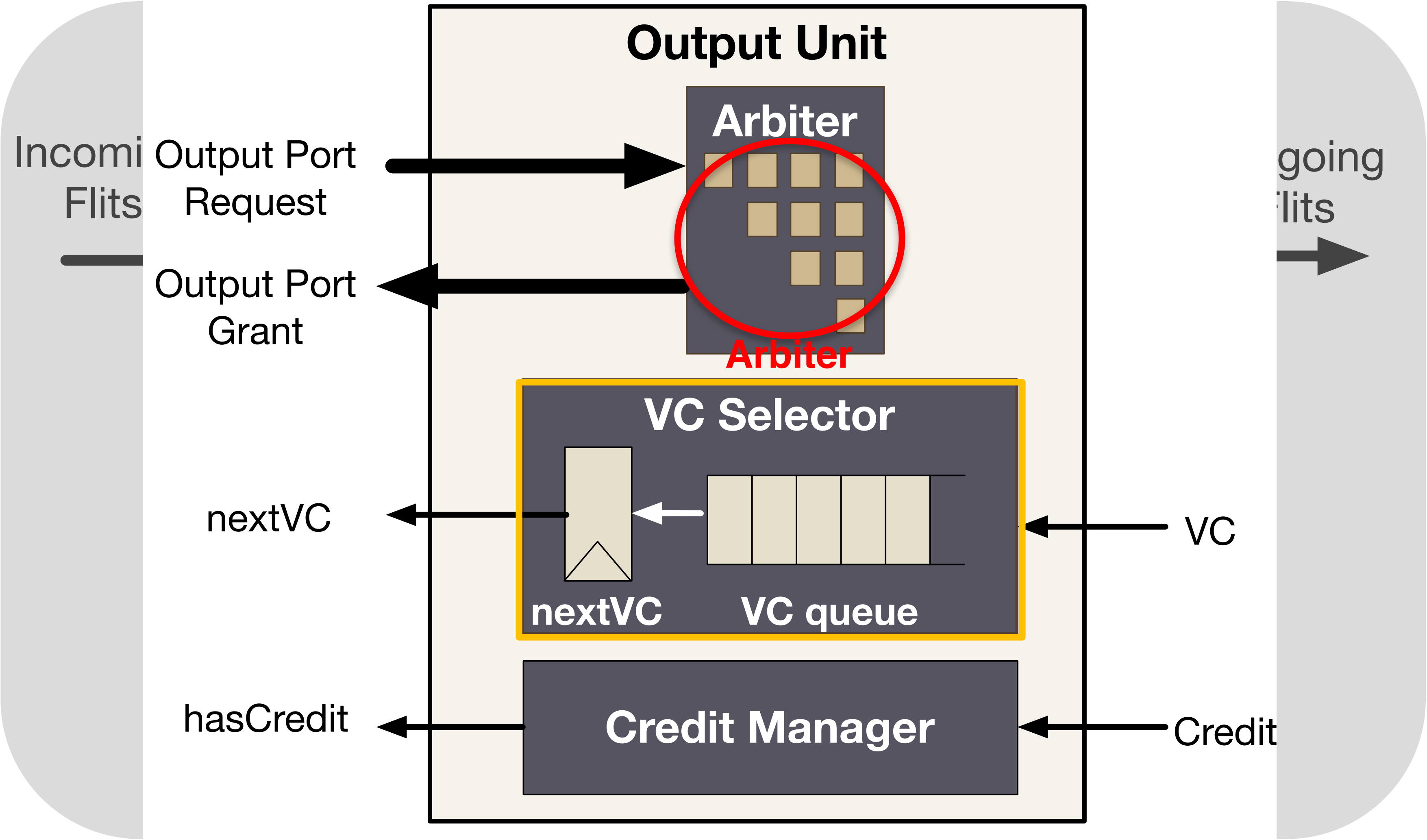
OpenSMART Router



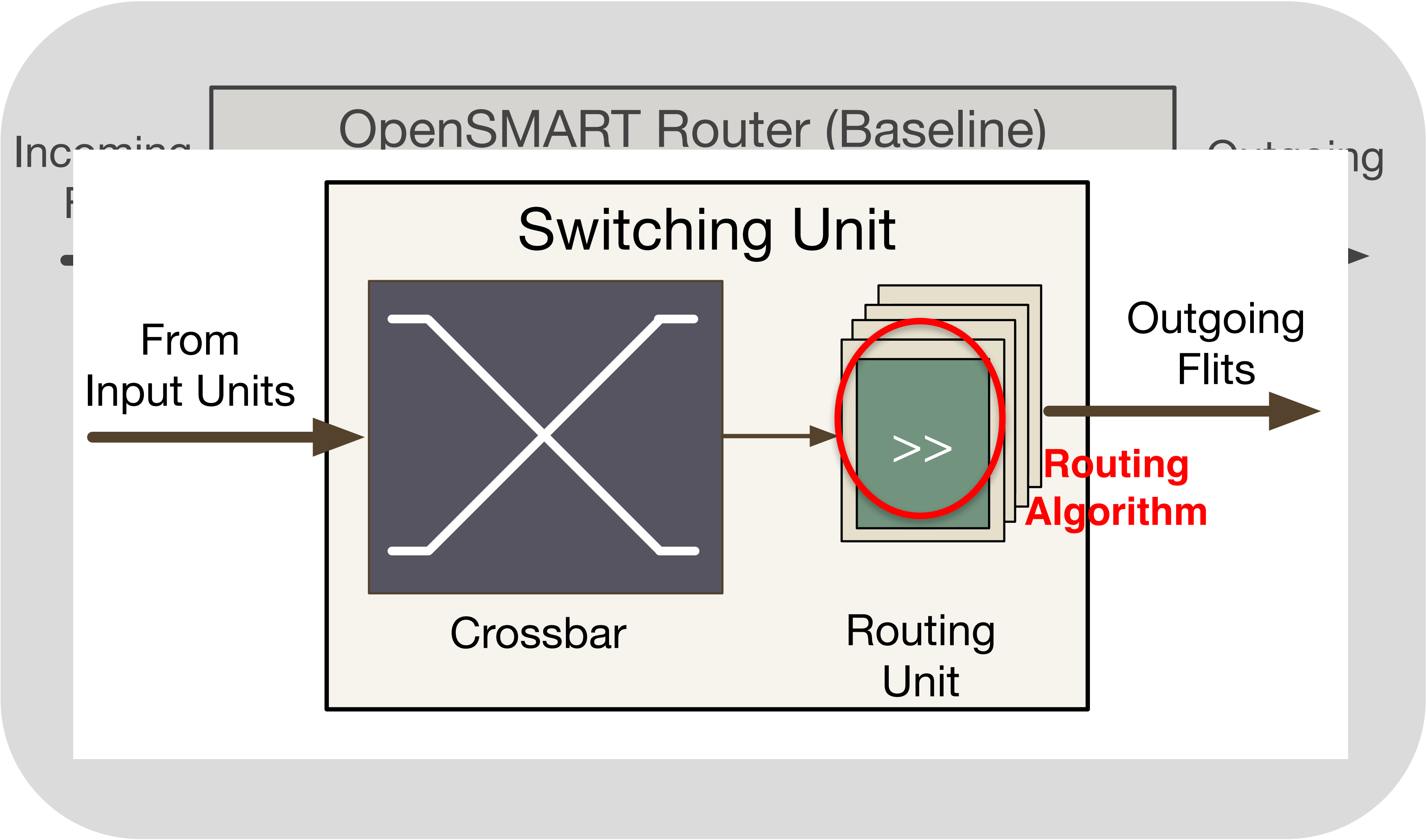
Flit Size

Number of VCs/VC Depth

OpenSMART Router



OpenSMART Router



OpenSMART Router (Baseline)

Incoming

Outgoing

Switching Unit

From
Input Units

Crossbar

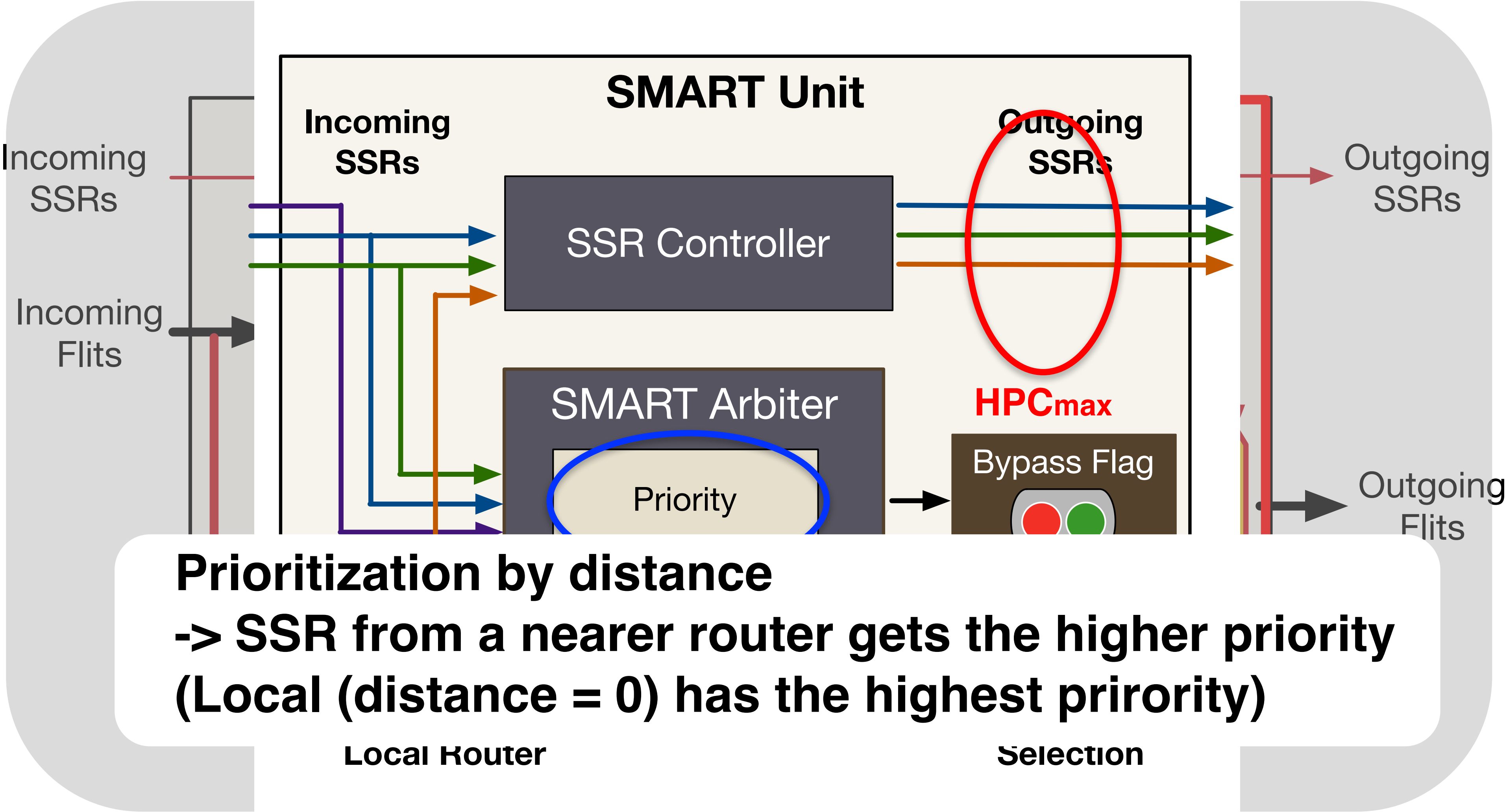
>>

Routing
Unit

Outgoing
Flits

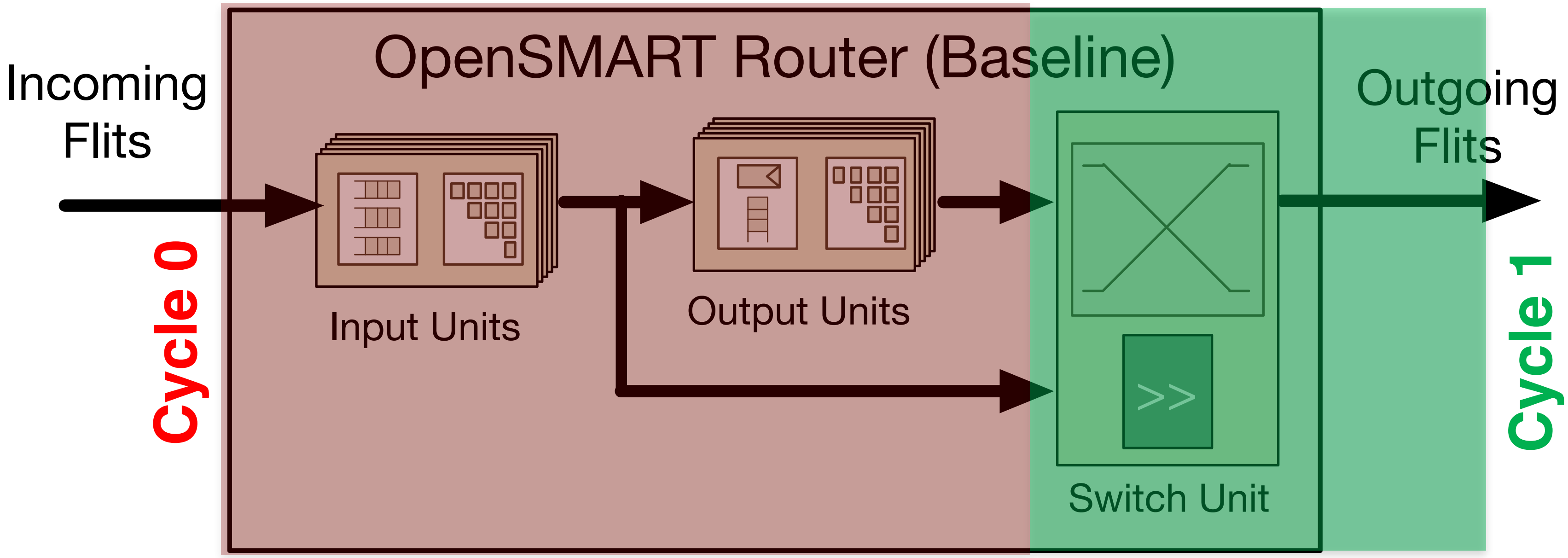
**Routing
Algorithm**

OpenSMART Router (SMART)

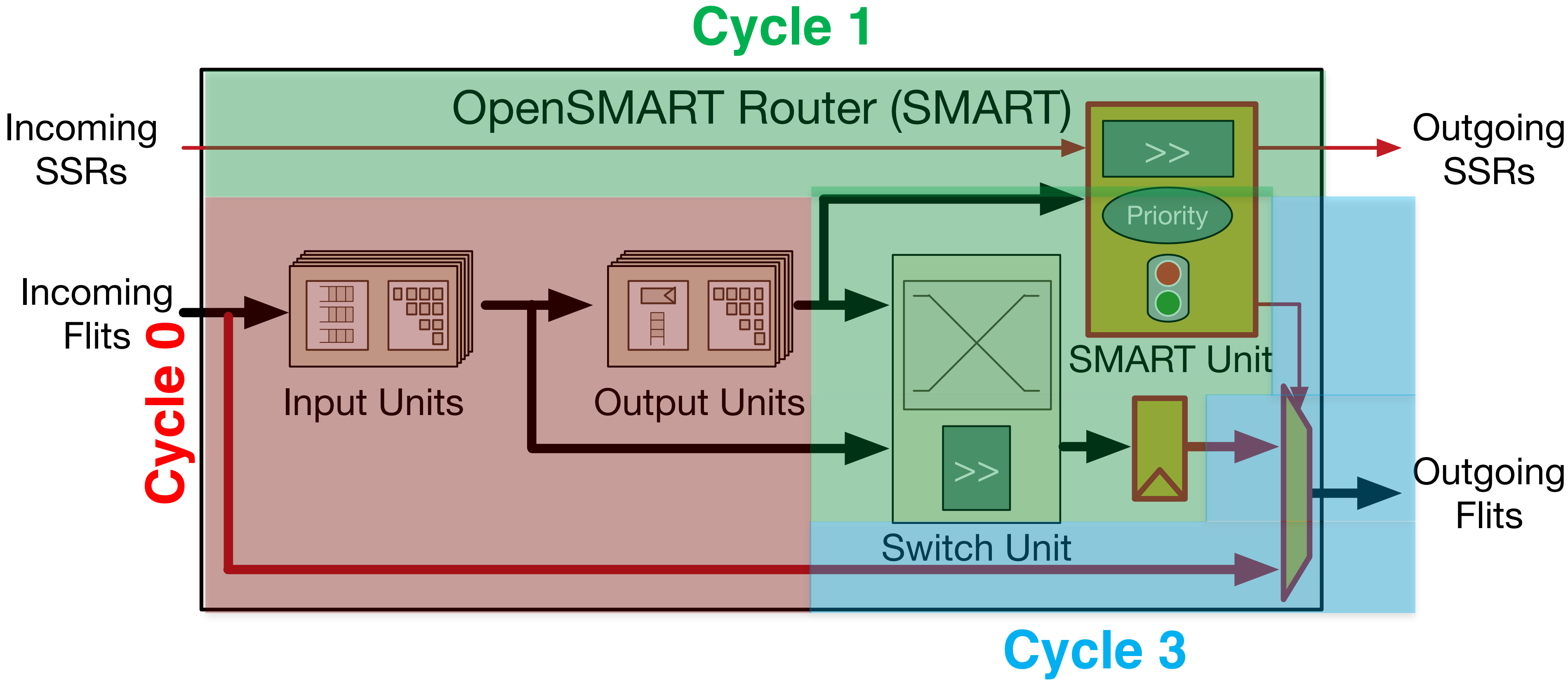


Prioritization by distance
-> SSR from a nearer router gets the higher priority
(Local (distance = 0) has the highest prirority)

OpenSMART Router (1cycle)



OpenSMART Router (2cycle/SMART)



Outline

- Motivation: Scalable, Flexible, and Low-cost NoCs
- Background: SMART NoCs
- **OpenSMART**
  - Design Flow
  - Building Blocks
  - **Walk-through Examples**
- Case Studies
  - Mesh vs. SMART
  - High-radix vs. Low-radix
- Conclusions

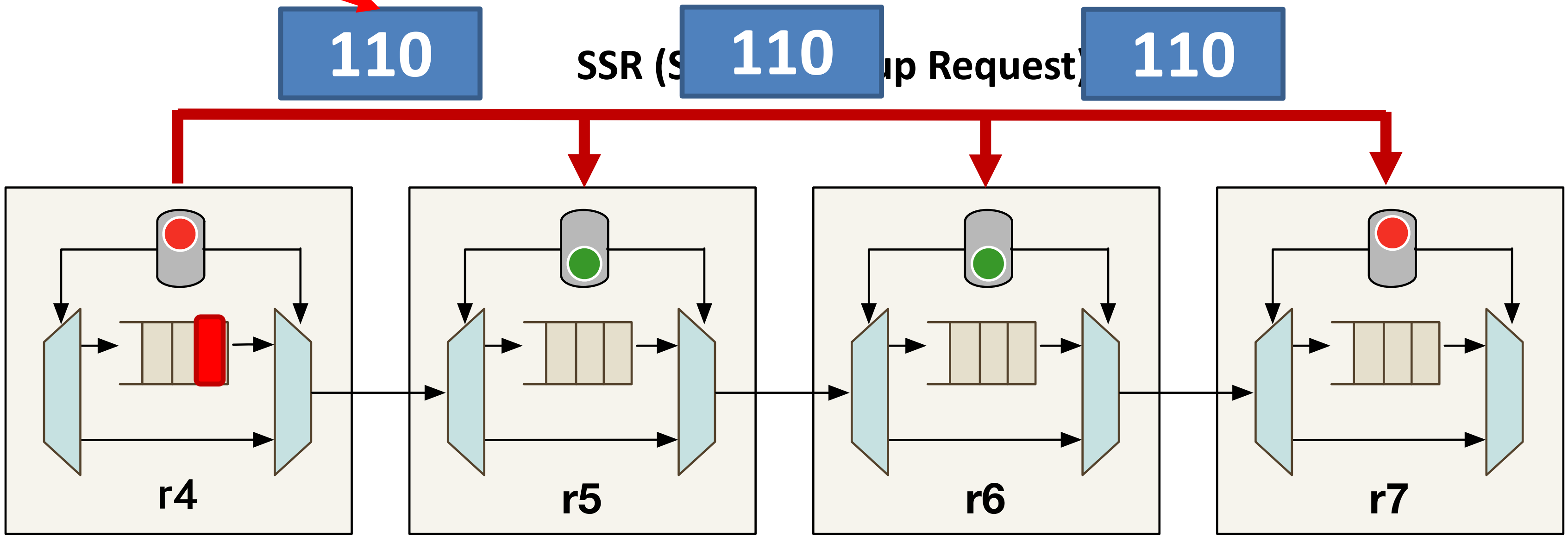
Walk-through Example 1

- Router r4 sends a flit to router r7
- HPCmax = 3

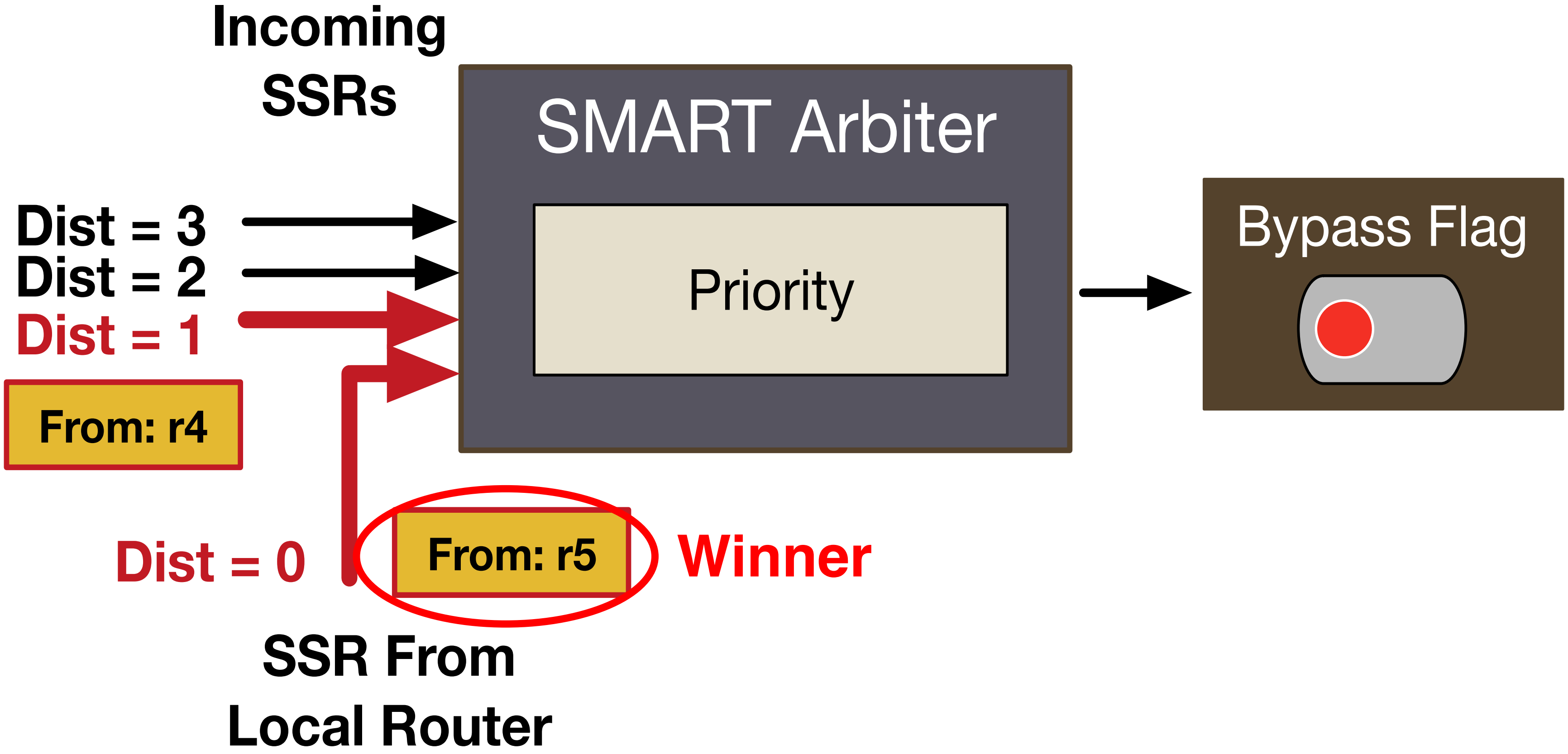
Cycle 1: Multi-hop Bypass

Cycle 0: SSR Setup

bypass, bypass, stop



Walk-through Example 2



SMART Unit in r5

OpenSMART: Features

- **Language**
  - **BSV** and **Chisel**
- **Flow control**
  - **VC** and **SMART**
- **Buffer management**
  - **Credit-based** buffer management
- **Router microarchitecture**
  - **1- and 2-cycle** state-of-the-art packet switching router
  - **SMART** router

OpenSMART: Features

- **Routing calculation**
  - **XY**, **YX**, and **source-routing**
  - **One-hot** encoding hop count + **shift-based** routing calculation
  - For SMART, routing calculation is done during bypasses

- **VC selection**
  - **FIFO**-based **dynamic** VC selection
  - Next VC is stored in a separate register
  - For SMART, VC selection is done during bypasses

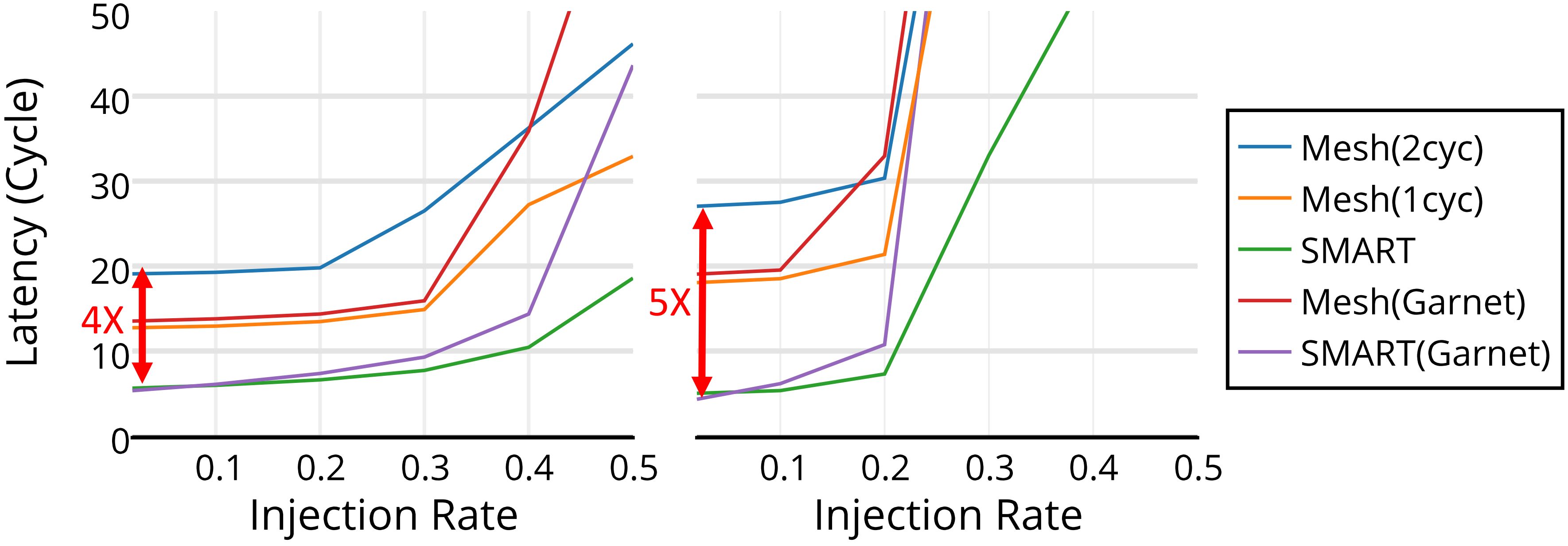
Outline

- Motivation: Scalable, Flexible, and Low-cost NoCs
- Background: SMART NoCs
- OpenSMART
  - Design Flow
  - Building Blocks
  - Walk-through Examples
- **Case Studies**
  - **Mesh vs. SMART**
  - High-radix vs. Low-radix
- Conclusions

Case Study Configuration

- **Network Topology:** 8x8 mesh
- **Number of VCs:** 4
- **Traffic:** Uniform-random and bit-complement
- **HPCmax :** 7
- **Flit Size :** 52 bit (data: 32 bit)
- **Synthesis Environment :** Synopsys Design Compiler with NanGate 15nm PDK standard cell library // Xilinx Vivado (target: VC709 board)
- **Simulation Method:** Cycle-accurate RTL simulation using BSV testbench and Garnet simulation

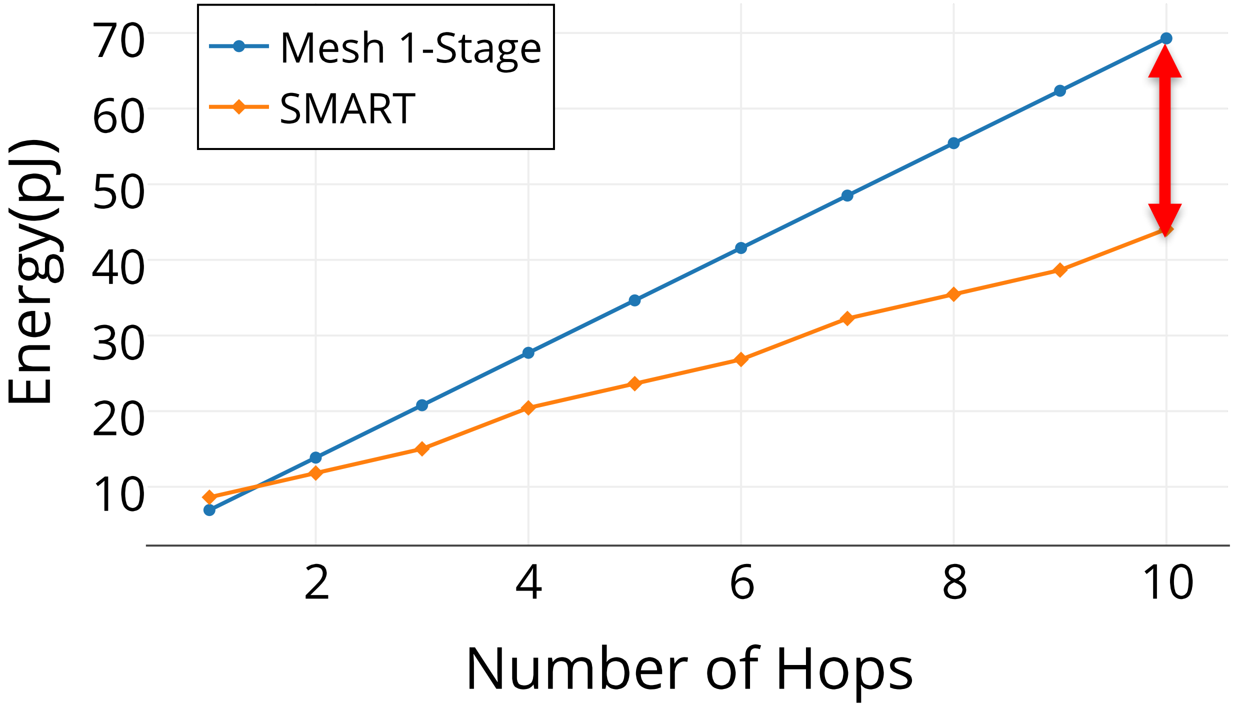
Latency



(a) Uniform Random

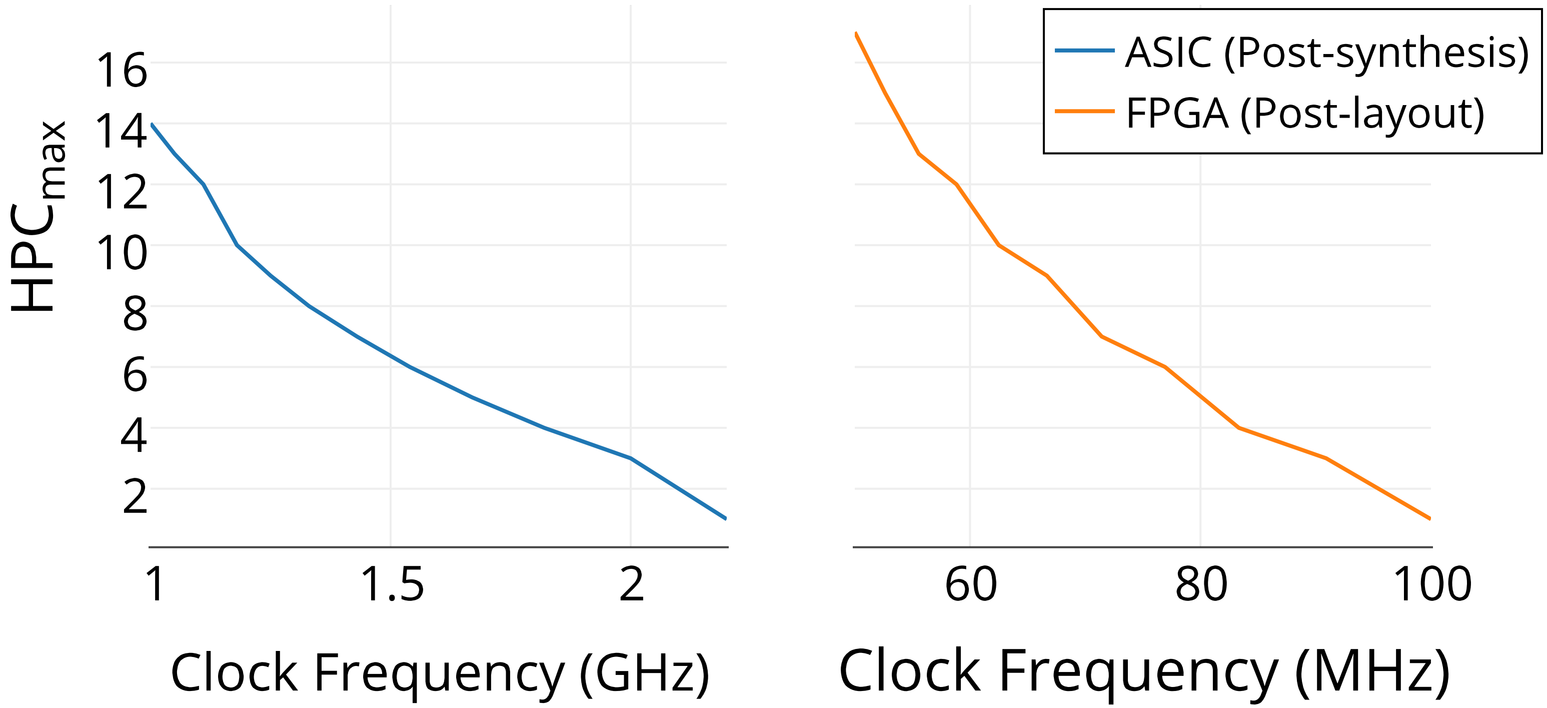
(b) Bit-complement

Energy Consumption



Repeaters require less energy than clocked latches

HPCmax

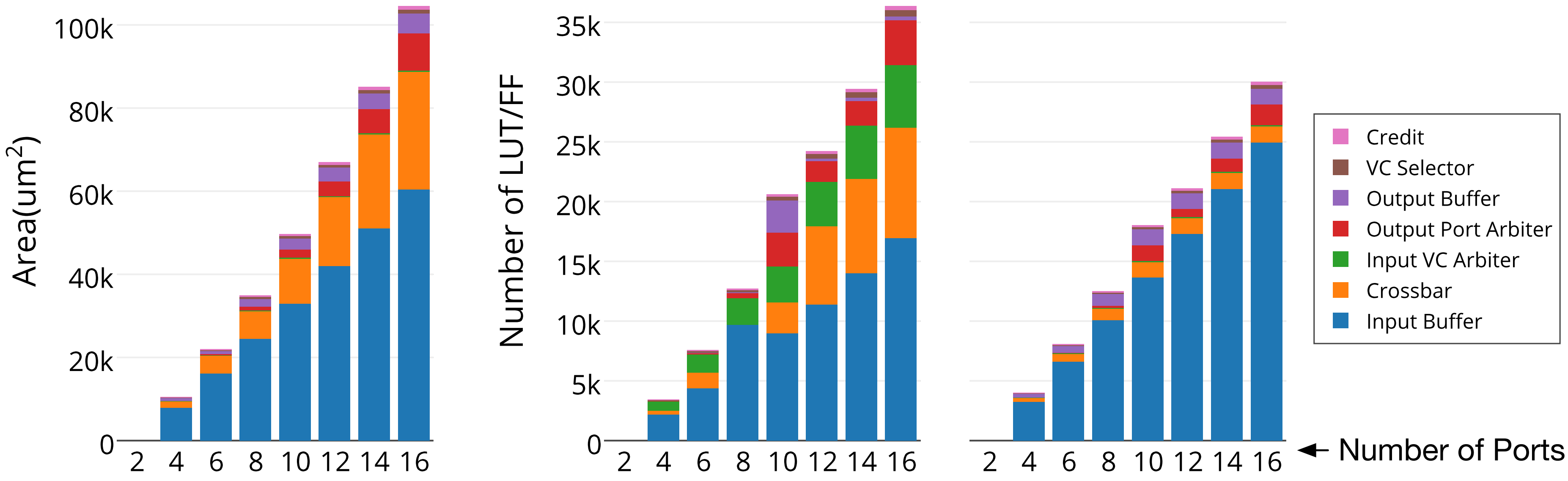


(a) HPCmax on ASIC **(b) HPCmax on FPGA**

Outline

- Motivation: Scalable, Flexible, and Low-cost NoCs
- Background: SMART NoCs
- OpenSMART
  - Design Flow
  - Building Blocks
  - Walk-through Examples
- **Case Studies**
  - Mesh vs. SMART
  - **High-radix vs. Low-radix**
- Conclusions

Router Area

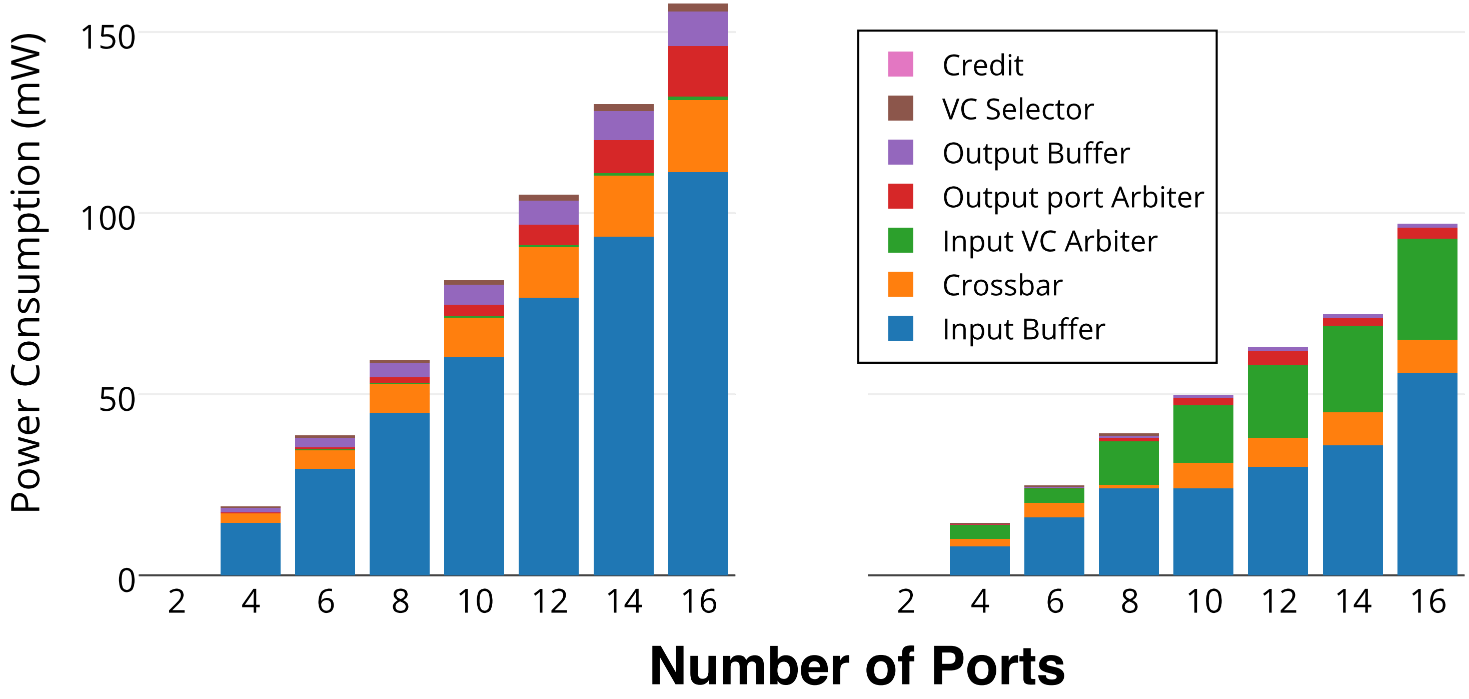


(a) ASIC area

(b) FPGA LUTs

(c) FPGA FFs

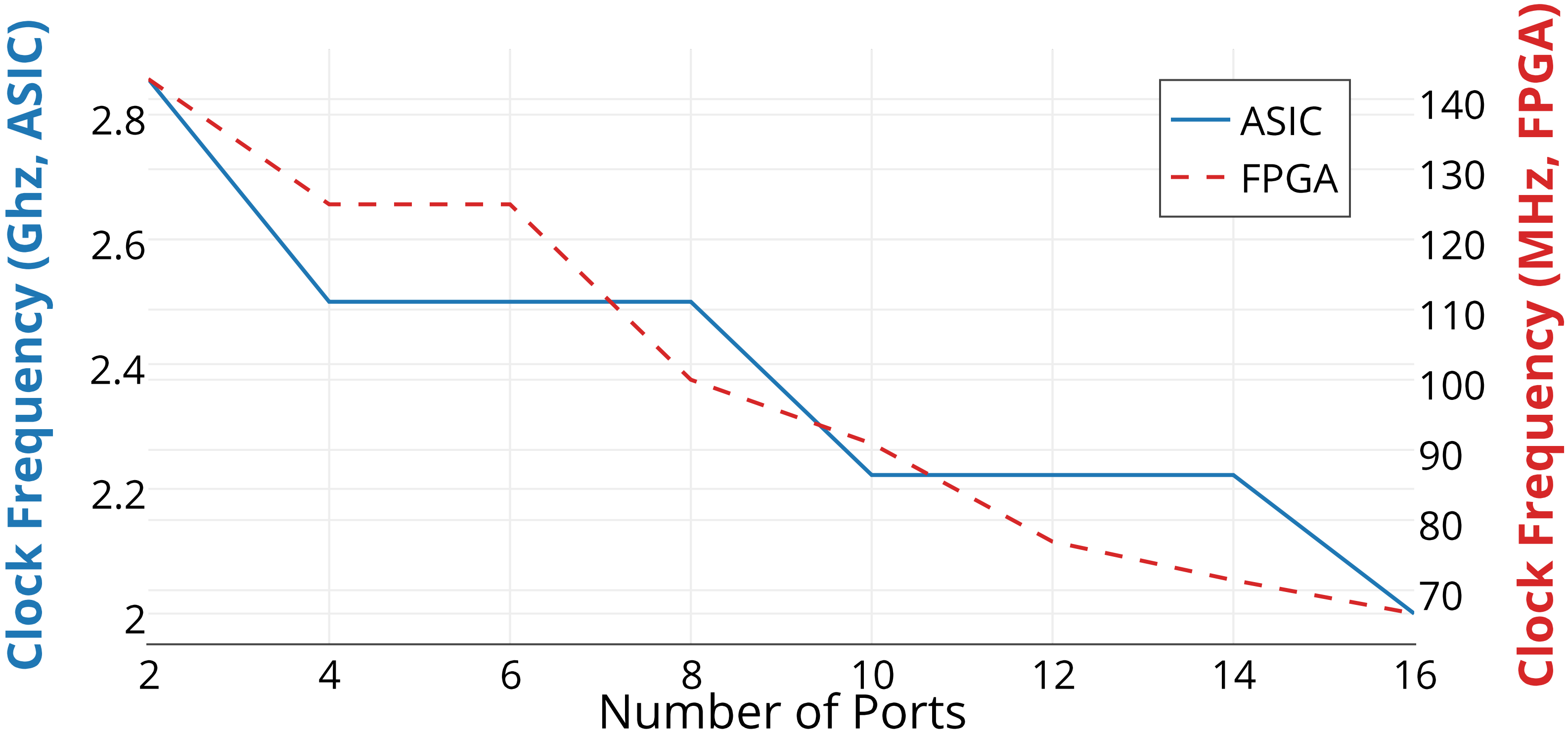
Router Power



(a) ASIC

(b) FPGA

Maximum Clock Frequency



Outline

- Motivation: Scalable, Flexible, and Low-cost NoCs
- Background: SMART NoCs
- OpenSMART
  - Design Flow
  - Building Blocks
  - Walk-through Examples
- Case Studies
  - Mesh vs. SMART
  - High-radix vs. Low-radix
- **Conclusions**

Conclusion

- NoCs are crucial components to support many-IP heterogeneous systems
- ***OpenSMART*** provides automatic generation of NoCs RTL for many-IP heterogeneous systems
- ***OpenSMART*** generates not only state-of-the-art packet switching network but also low latency network, SMART.

Thank you!

Paper and Source code

- Paper is available this link : http://synergy.ece.gatech.edu/wp-content/uploads/sites/332/2017/03/OpenSMART_ISPASS17.pdf

- Source code is available via this link: https://hyoukjun.github.io/OpenSMART/