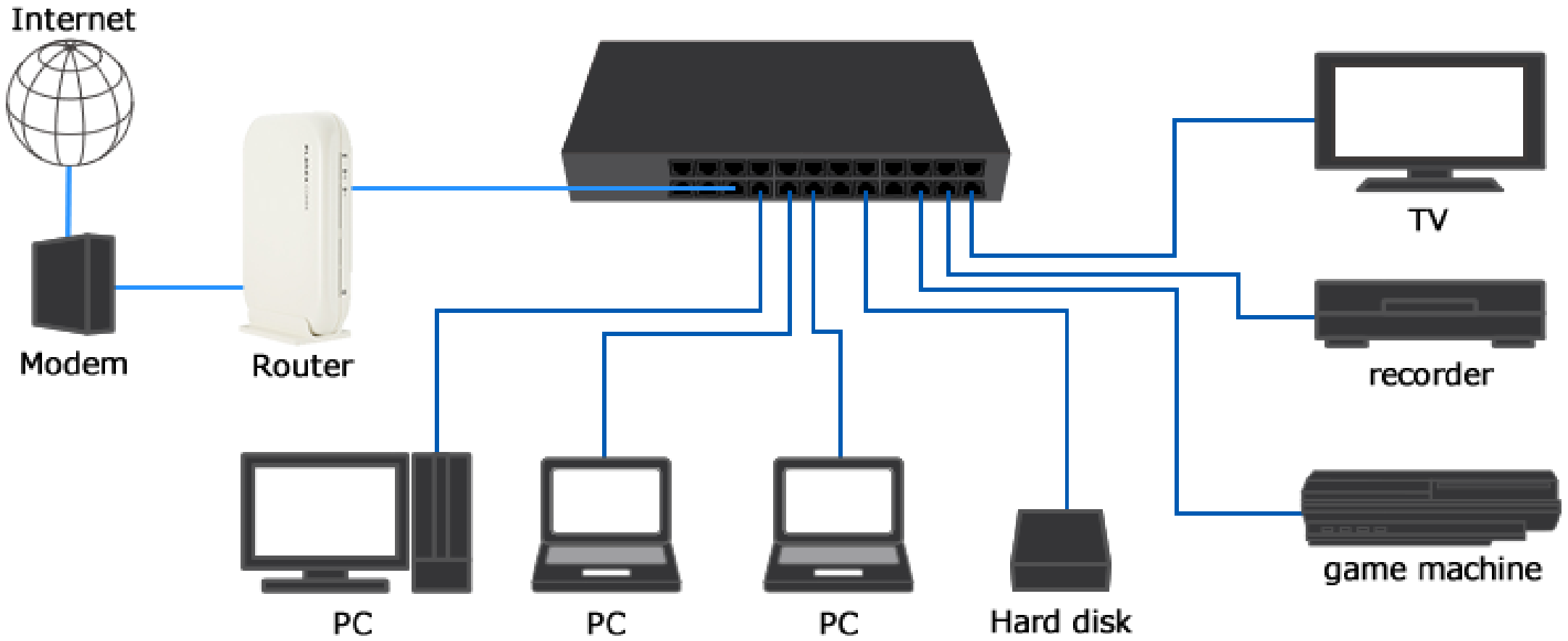
Investigating the Feasibility of FPGA-based Network Switches

Jiuxi Meng[1], Nadeen Gebara[1], Ho-Chueng Ng[1], Paolo Costa[1,2], Wayne Luk[1]

[1]Imperial College London

[2]Microsoft

What is a network switch?



http://www.fiber-optic-cable-sale.com/network-switch-router.html

Motivation

- Application Specific Integrated Circuit (ASIC) based switches
  - Long development time
  - Expensive
  - Not future proof
  - Aggregate bandwidth of 12.8Tbps (**Broadcom's Tomahawk® 3**)

- Field Programmable Gate Array (FPGA)
  - Short development time
  - Customizable -> Cost-efficient
  - Widely used in data centre (e.g. **SmartNIC** from Microsoft)
  - Increasing in aggregate transceiver bandwidth (8Tbps with **Intel® Stratix® 10**)

Can we replace ASIC switches with FPGA switches?

Contributions

1. Efficient buffer-sharing architecture for crossbar switches

2. Performance and resource usage trade-offs of crossbar switches

3. Technology independent model as a prediction tool

Background: switch fabrics

- Shared memory
  - cost too high
- Shared bus
  - too slow
- Crossbar switch
  - widely used
  - non-blocking
  - simple structure, implemented with multiplexers on FPGA

Switch Architectures: buffering approaches

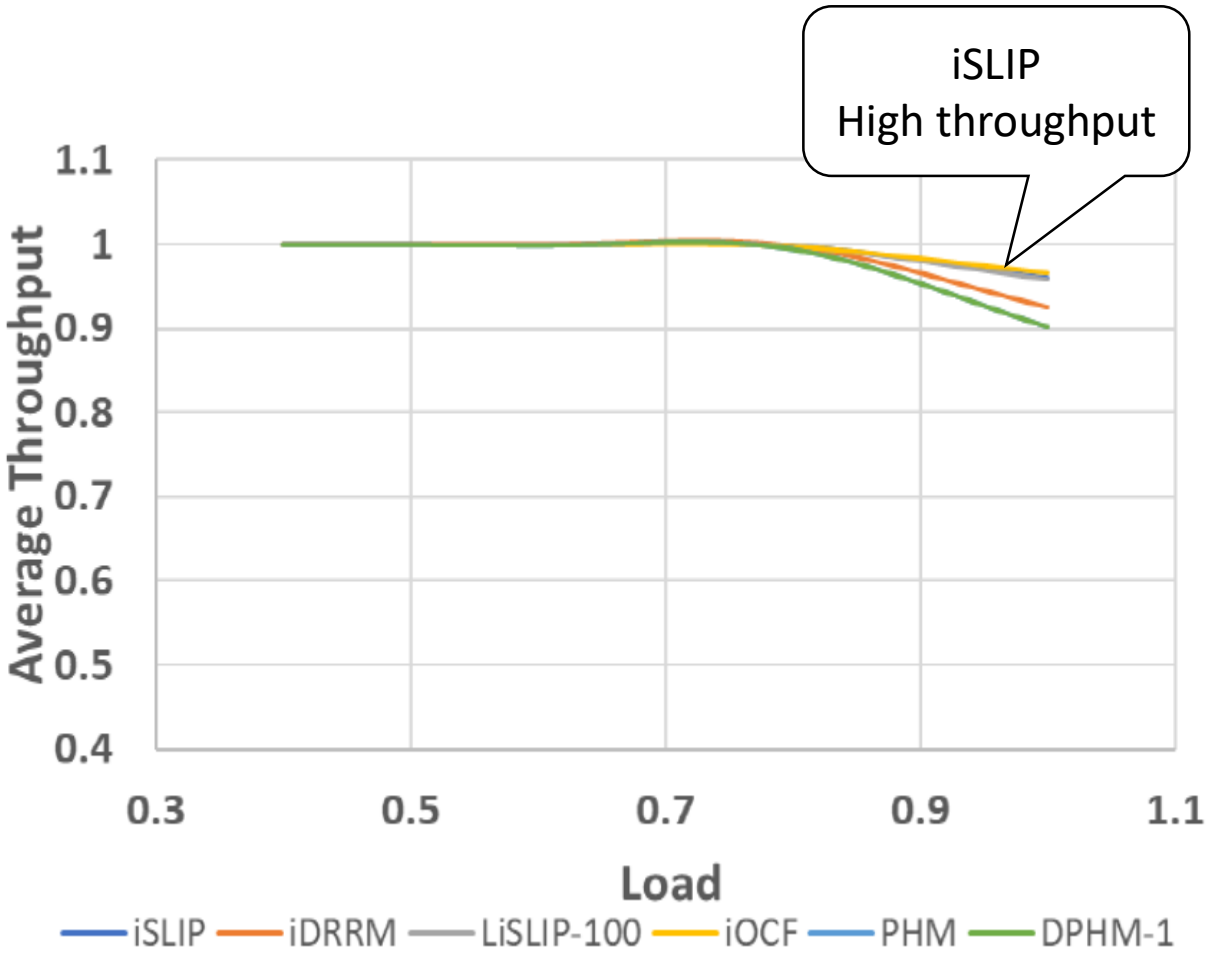
1) **I**nput **B**uffered crossbar switch (IB)
2) **O**utput **B**uffered crossbar switch (OB)
3) **C**ombined **I**nput and **O**utput **B**uffered switch (CIOB)
4) **C**ombined **I**nput and **C**rosspoint **Buffered** switch (CICB)
5) **G**rouped **C**rosspoint **Q**ueued switch (GCQ)
   - hierarchical crossbar targeting FPGA technoogy

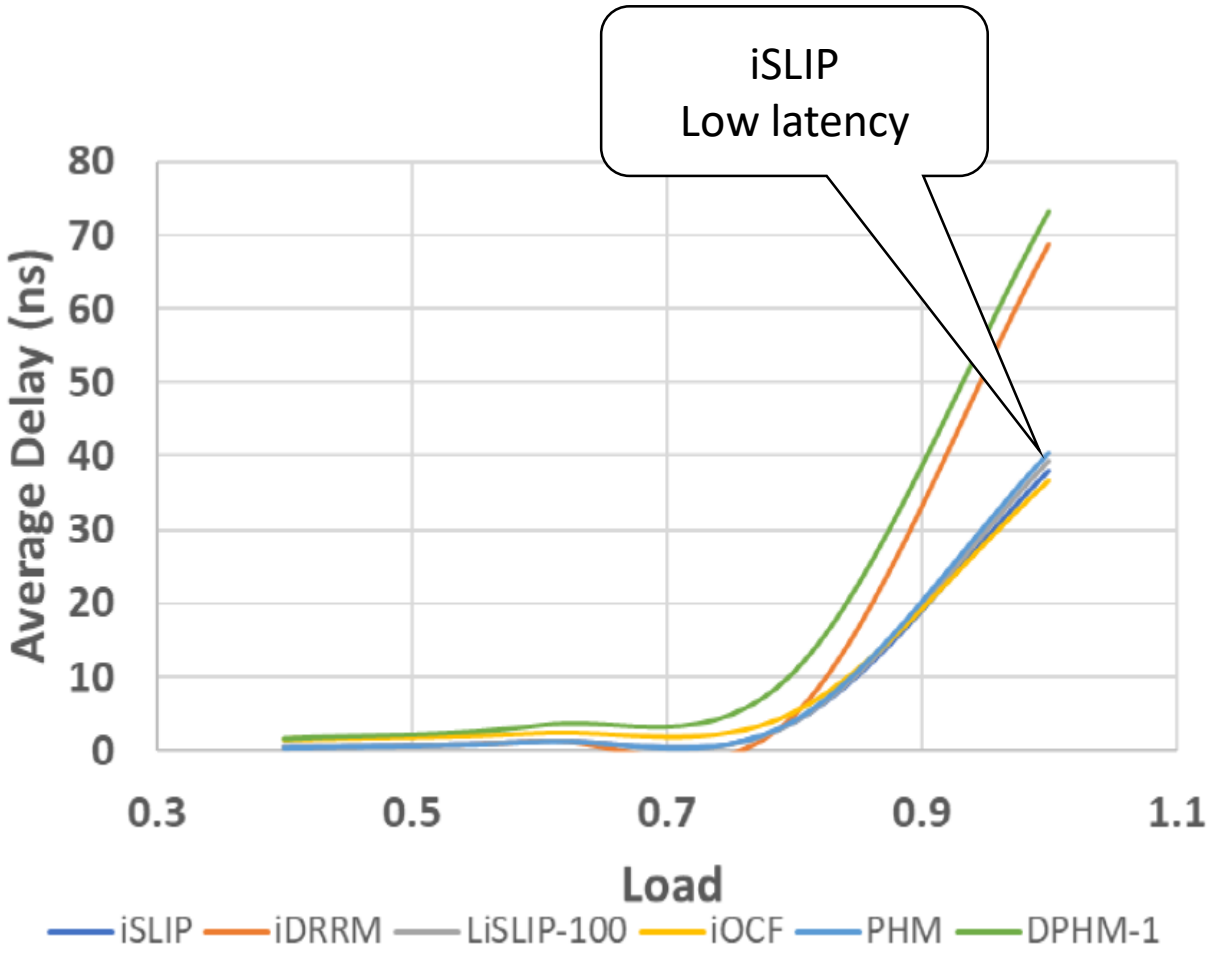
Architecture comparison

	IB	OB	CIOB	CICB	GCQ
Input buffer	N^2	0	N	N	N^2/S
Output buffer	0	N	N	0	0
Crosspoint buffer	0	0	0	N^2	$(N/S)^2$
speed up	0	N	S	0	S

N: port size, W: port width, S: speeding up the block RAM in GCQ by S times to emulate an SxS switch

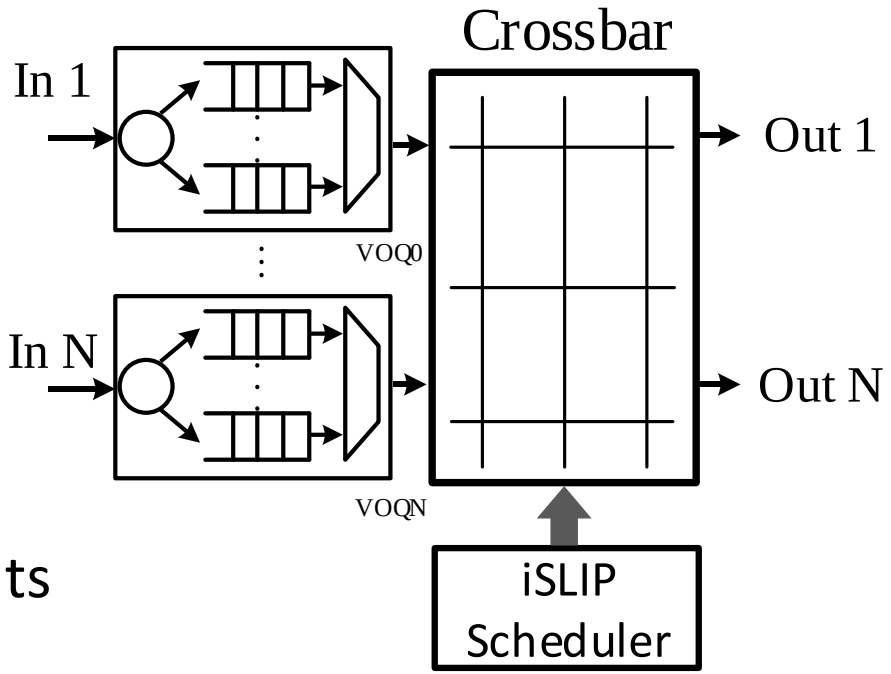
Scheduling algorithm comparison



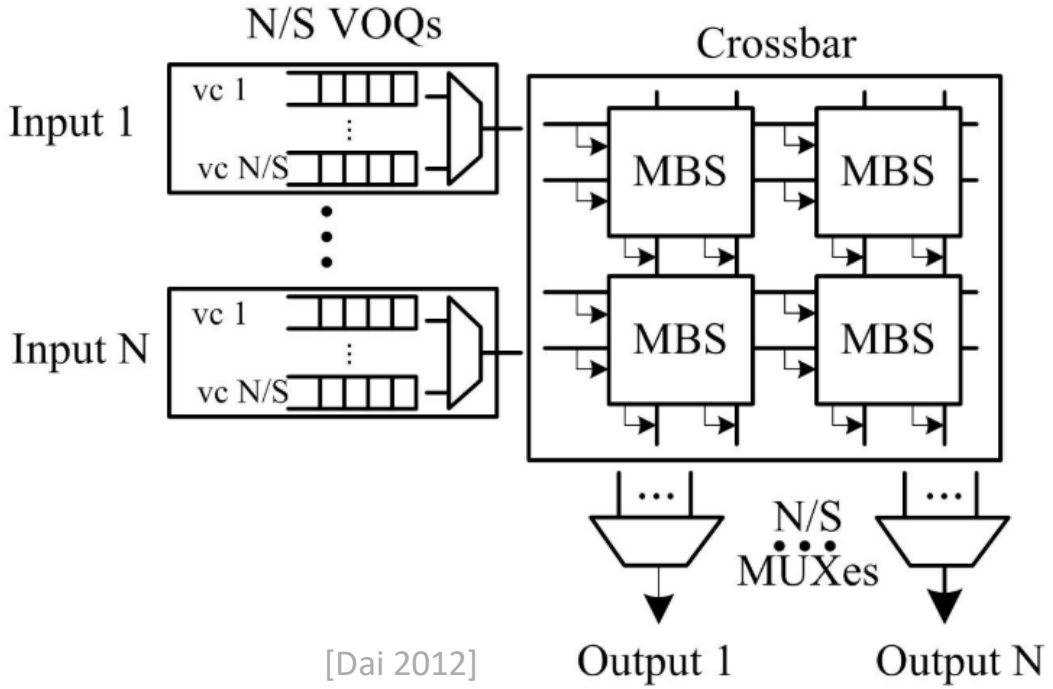


Final Choices

- iSLIP based input buffered (IB) switch
  - Widely used in commercial switches
  - Easy to implement
  - Good performance based on our simulation results



- GCQ based switch
  - Fine tuned for FPGA
  - Memory resource efficient

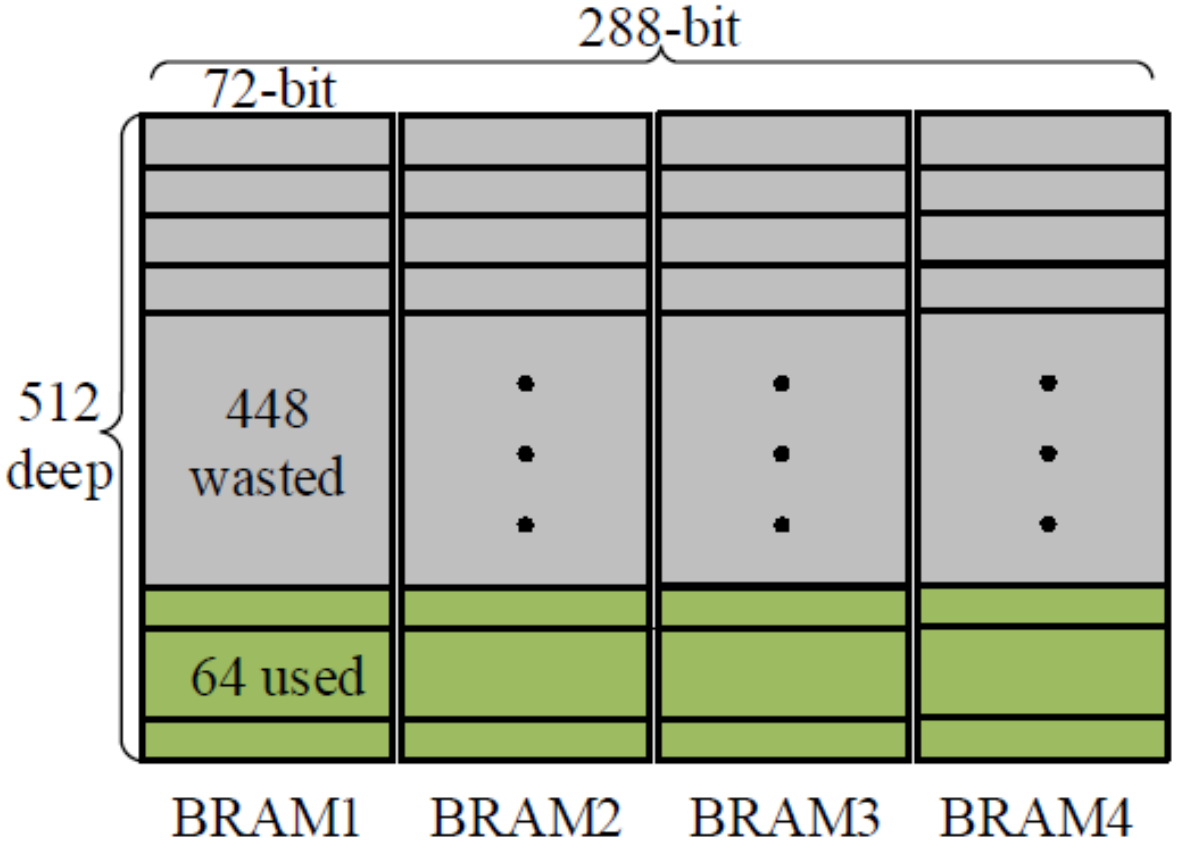


[Dai 2012]

1. Efficient buffer-sharing architecture

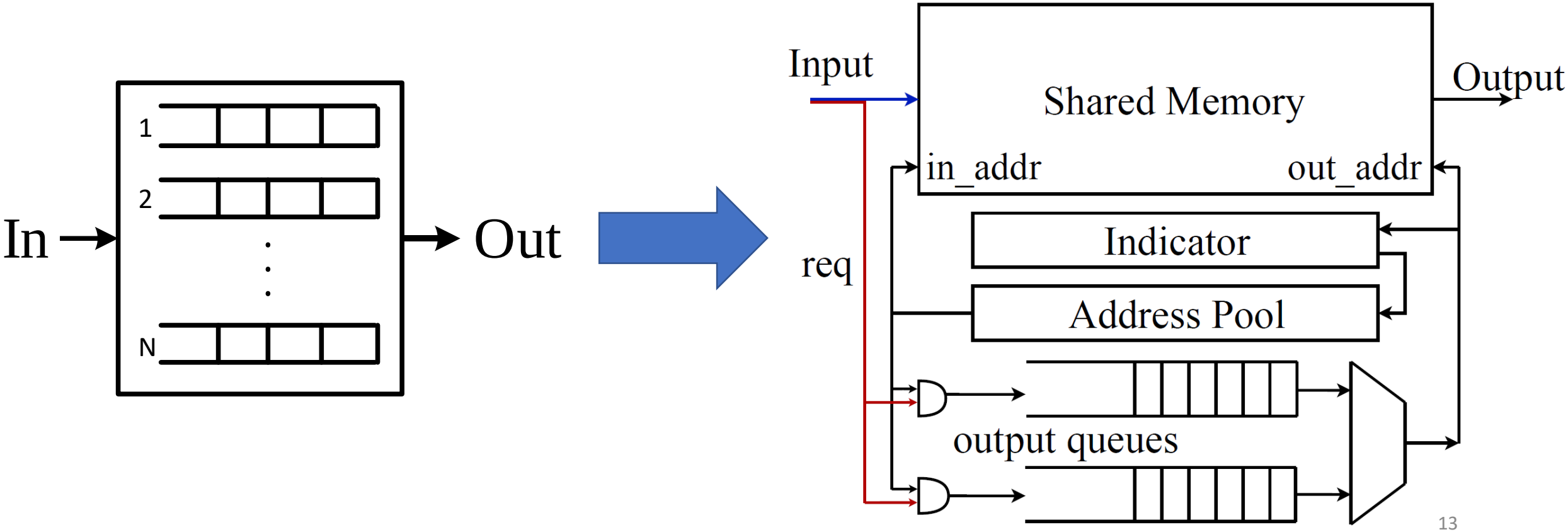
Challenges of implementing switch buffers on FPGAs

- FPGA has limited on-chip memory
- Memory efficiency is crucial



A resource efficient buffering architecture

- replace Virtual Output Queues (VOQs) with shared memory
- architecture similar to IBM's Prizma switch



Design setup

- Design Flow and Tools: Vivado Design Suite 2017.2
  - platform: Virtex Ultrascale+ XCVU9P
  - method: *out_of_context* mode to avoid IO insertion

- Buffer Depth: 64
  - 40 for zero packet loss from Omnet++ simulation

- Packet Format:

Field	Source	Data	Destination
Width	$\log_2 N$	$W - \log_2 N - N$	N

Design setup

Platform	Operating frequency	Line rate	Data width
Virtex UltraScale+ xcvu9p	40MHz	10Gb/s	256
	40MHz	25Gb/s	640
	40MHZ	50Gb/s	1280

iSLIP scheduler implementation result

N	LUT	FF	Max frequency (MHz)
128	255739	34571	**39.9**

Why Operating freq. = 40 MHz?

1. iSLIP scheduler runs at 39.9MHz with 128 ports

2. Fair comparison with GCQ design

2. Performance and resource usage trade-offs

Implementation results

Virtex Ultrascale+ XCVU9P

	iSLIP			SM-iSLIP			GCQ		
N	10G	25G	50G	10G	25G	50G	10G	25G	50G
8	✓	✓	✓	✓	✓	✓	✓	✓	✓
16	✓	✓	T	✓	✓	✓	✓	✓	✓
32	✓	X	X	✓	✓	✓	✓	✓	✓
64	X	X	X	✓	C	X	✓	✓✓	X
96	X	X	X	X	X	X	X	X	X

✓: Successful implementation

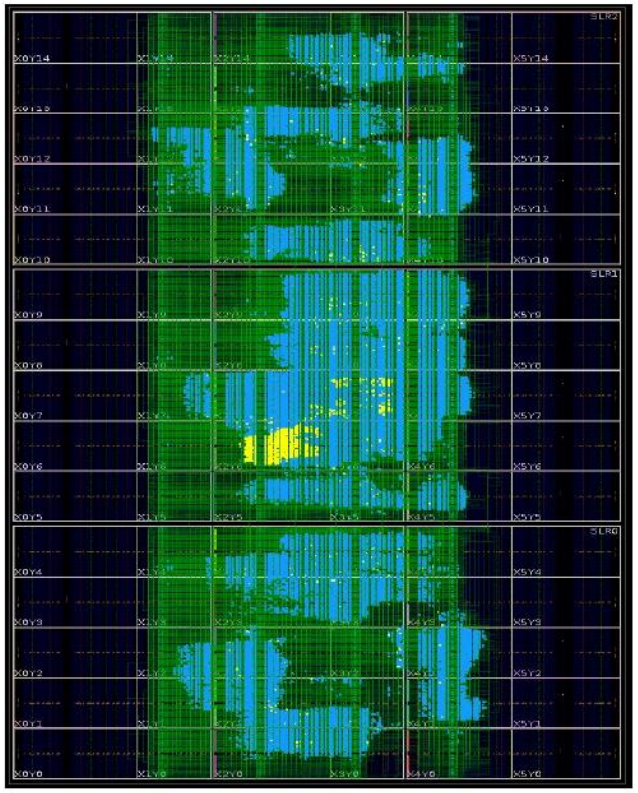
✓✓: Best aggregate throughput

C: Failed to resolve global congestion

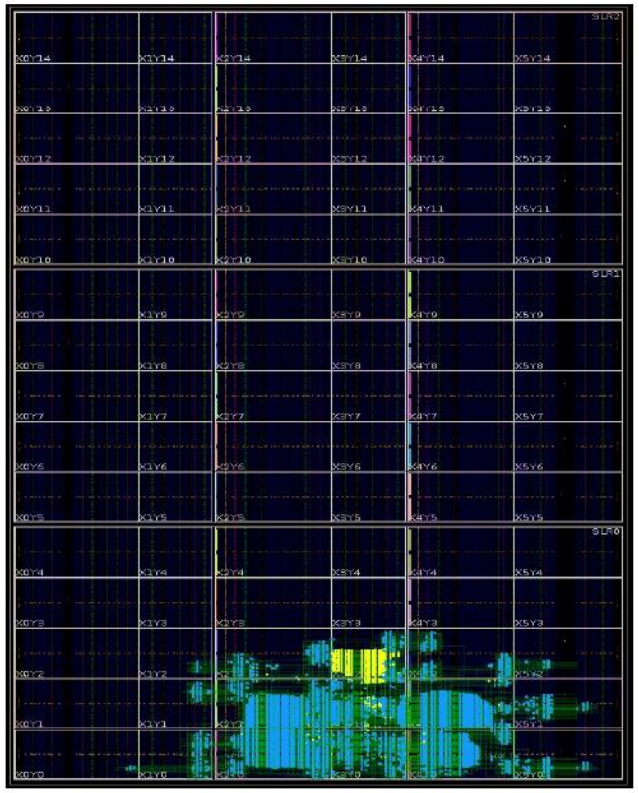
T: Failed to resolve timing closure

X: Implementation failed due to resource shortage

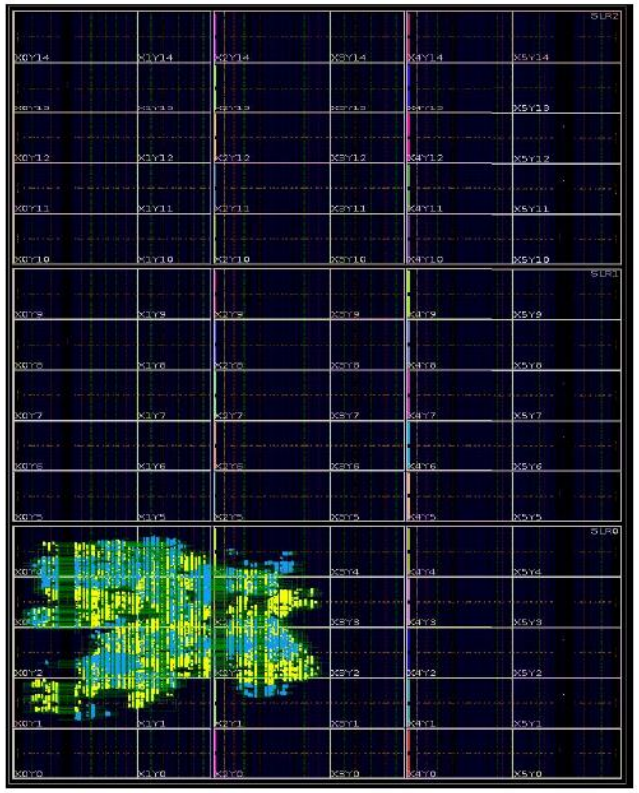
Device view of three designs



(a) iSLIP 10G 16-port

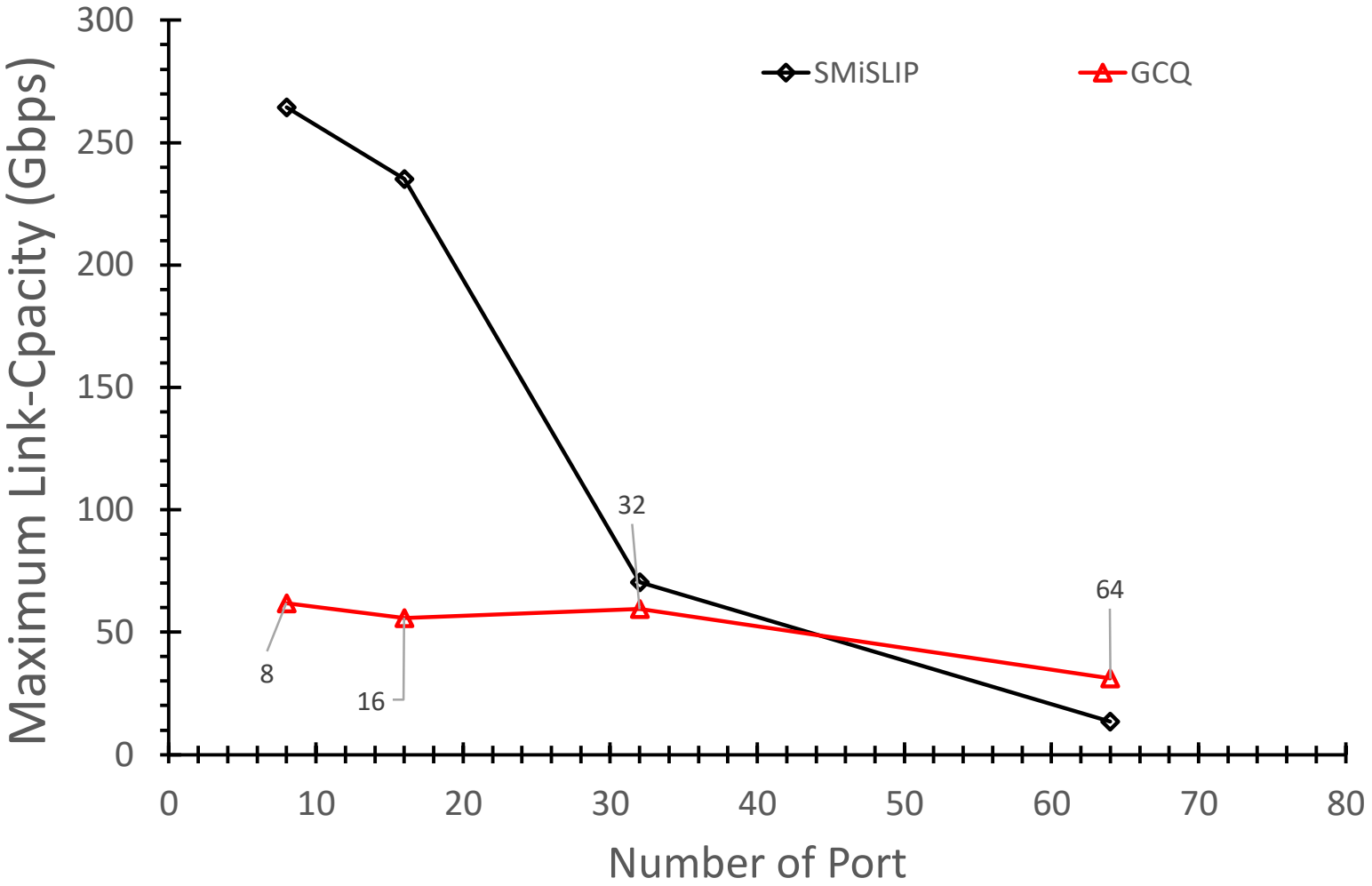


(b) SMiSLIP 10G 16-port



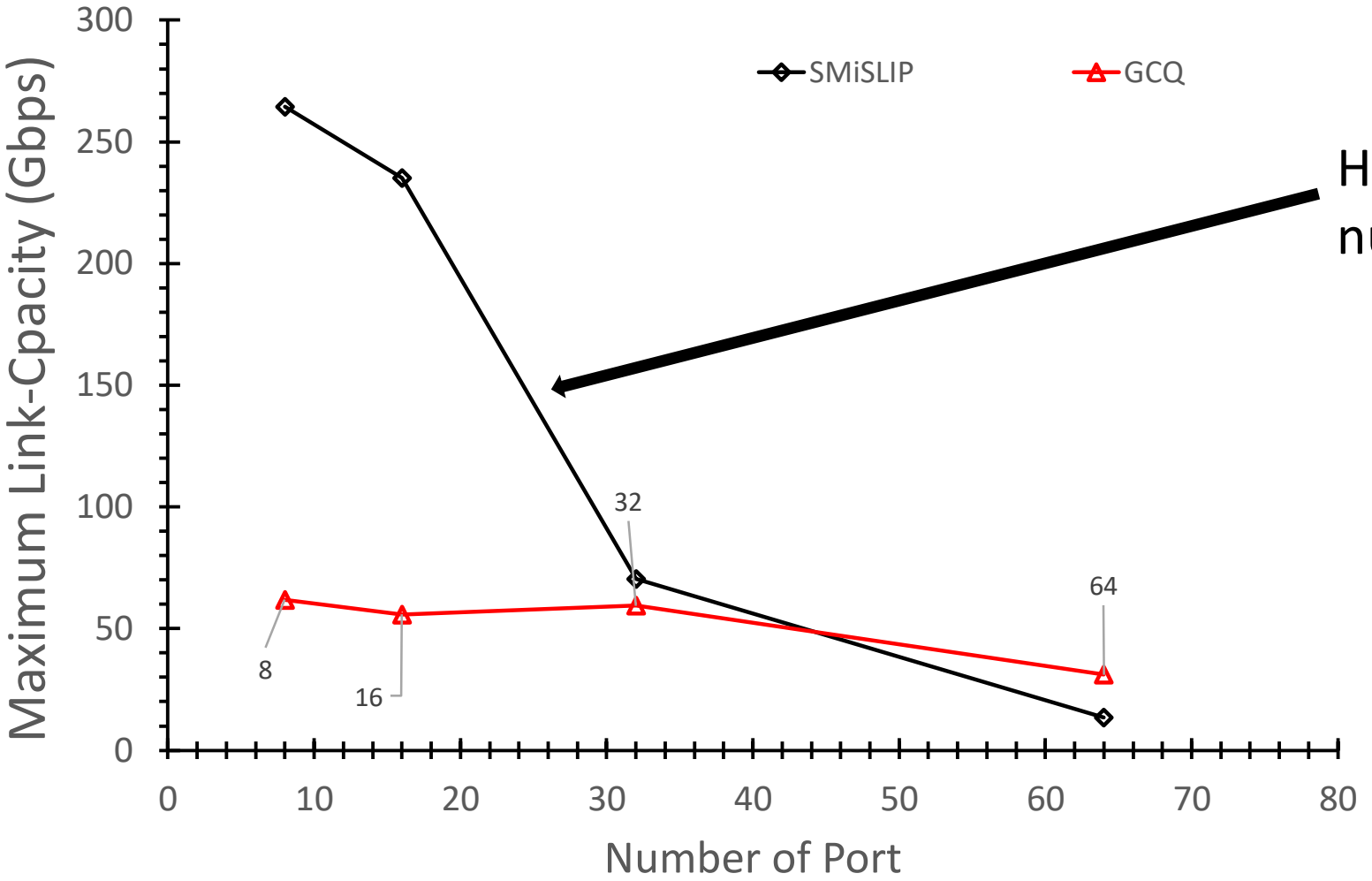
(c) GCQ 10G 16-port

Link capacity



(a) Maximum link capacity vs. port number

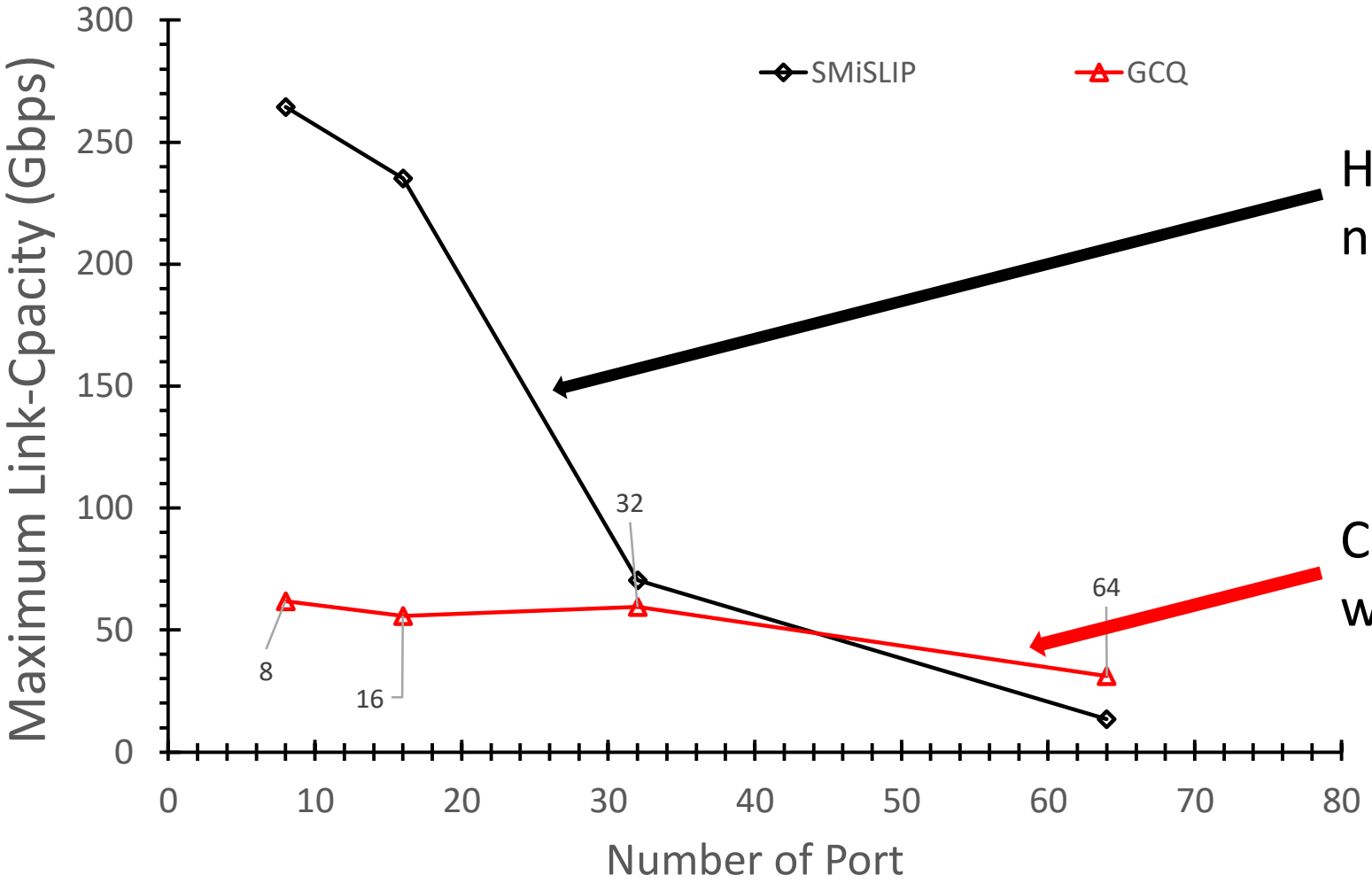
Link capacity



Highly affected by number of port

(a) Maximum link capacity vs. port number

Link capacity



Highly affected by number of port

Changing steadily with number of port

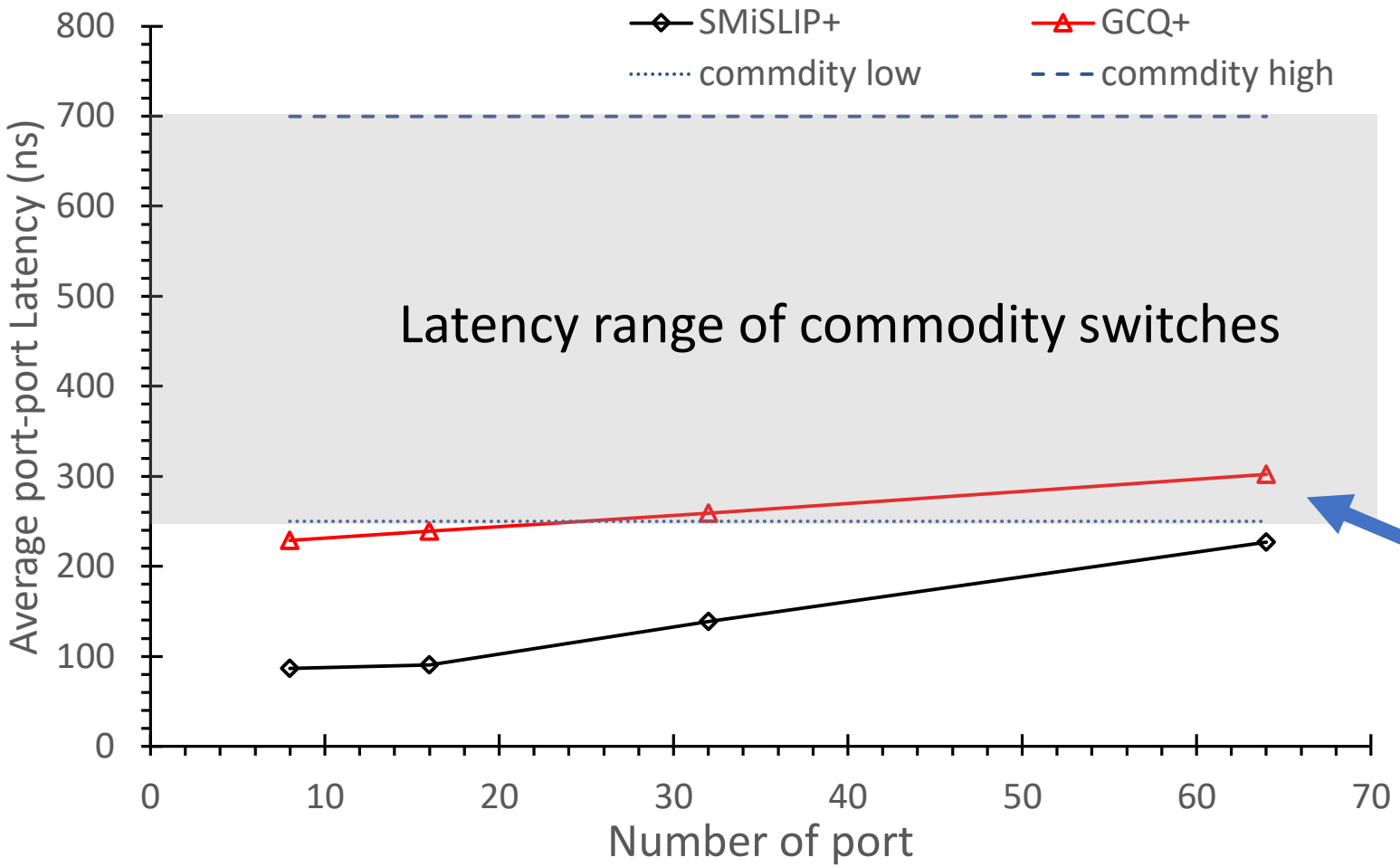
(a) Maximum link capacity vs. port number

Commercial Products for data centers

Brand	Series	Number of port	Port speed	Latency
Juniper	QFX	32 48 64 96 128	10G 25G 40G 50G 100G	550ns minimum
Cisco	Nexus 3200	32 64	10G 40G	450 - 700ns
Cisco	Nexus 3500	24 48	10G	Sub 250ns
FS	N series	20 24 32 48	10G - 100G	480 - 680ns
Mellanox	InfiniBand	36 40 per edge	40G 56G 100G 200G	Sub 90ns

https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000480-en.pdf
https://www.cisco.com/c/en/us/products/switches/nexus-3000-series-switches/models-comparison.html
https://www.fs.com/uk/c/network-switches-3079
http://www.mellanox.com/pdf/products/SwitchSystem_Brochure.pdf

Port to port latency



(b) Average port-port latency vs. port number, "+" means with transceiver latency added

GCQ based switch has higher port-to-port latency than iSLIP based one

3. Technology independent model

Challenges for estimation

- Three types of memory
  - BRAM, Distributed RAM and UltraRAM

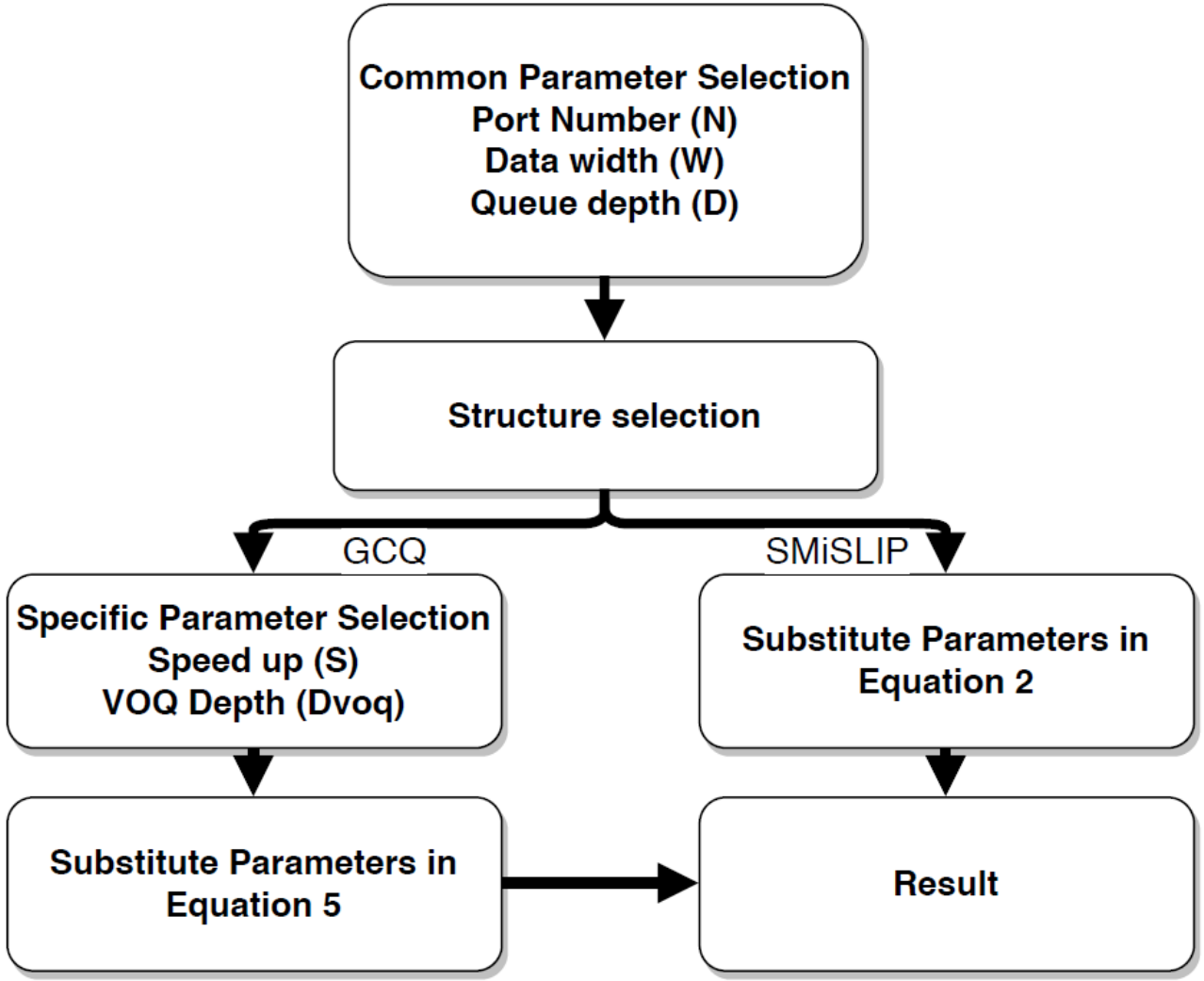
- Tools select different types of memory when one runs out

- Usage affected by design method
  - e.g. distribute small buffers to avoid wasting memory

Memory resource technology independent model

$$U_{sm} = N \times (\underbrace{DNW}_{shared\_mem} + \underbrace{2DNlog_2(DN)}_{mem\_pool\&out\_q} + \underbrace{DN^2}_{indicator}) \quad (2)$$

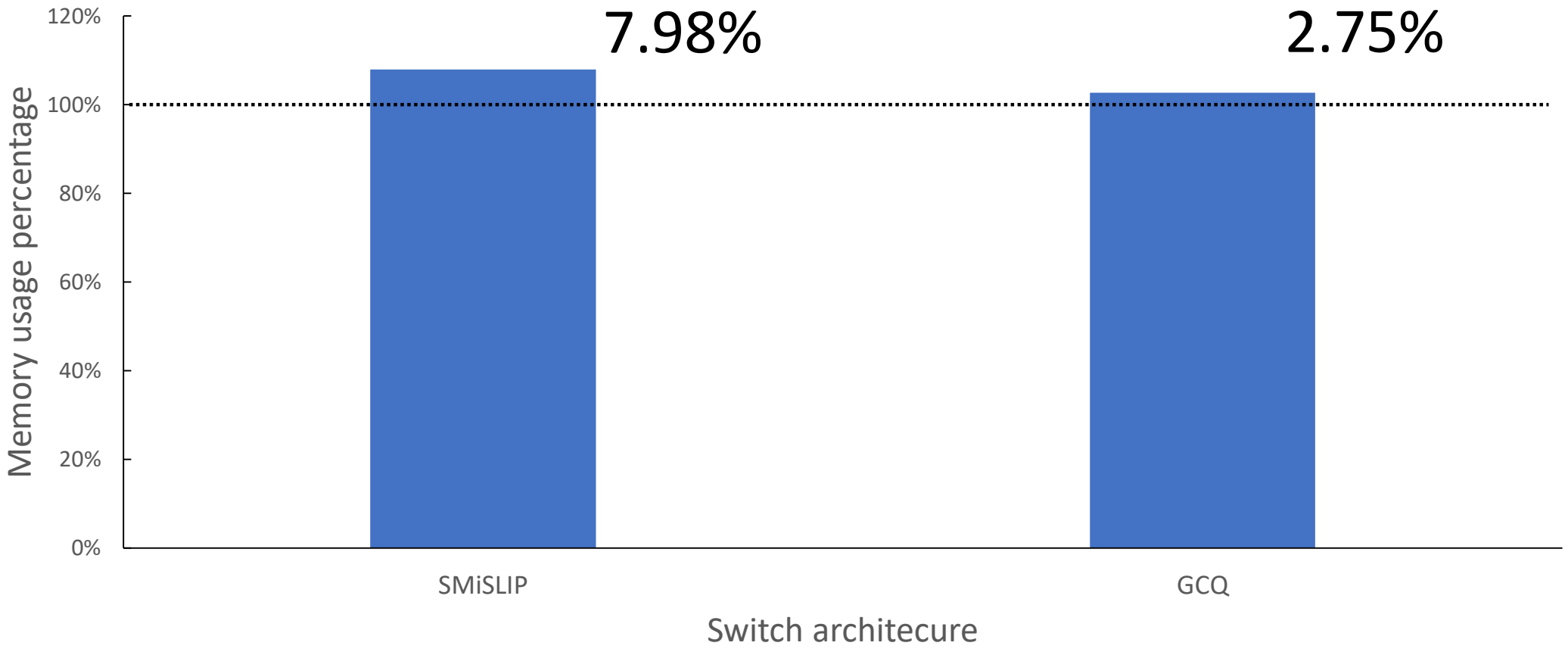
$$U_{GCQ} = \underbrace{(\frac{N}{S})^2 \times (DSW + 2DSlog_2(DS) + DS^2)}_{U_{MBS}} + \underbrace{N(N/S) \times W \times D_{VOQ})}_{U_{VOQ}} \quad (5)$$



Case study: memory model use case

	iSLIP			SM-iSLIP			GCQ		
N	10G	25G	50G	10G	25G	50G	10G	25G	50G
8	✓	✓	✓	✓	✓	✓	✓	✓	✓
16	✓	✓	T	✓	✓	✓	✓	✓	✓
32	✓	X	X	✓	✓	✓	✓	✓	✓
64	X	X	X	✓	C	X	✓	✓✓	X
96	X	X	X	C	X	X	X	X	X
128	X	X	X	X	X	X	X	X	X

Result from model



Future work

- Tune our model in variant platforms
- Investigate pipelined iSLIP scheduler and its impact
- Support more functions for the switches

Summary

- Shared memory: reduce memory usage on FPGA switches
- SMiSLIP: lower latency but hard to scale
- GCQ: link capacity less affected by number of ports,
  but higher latency

Can we replace ASIC switches with FPGA switches?

For now:

- more likely for small and medium scale, e.g. ~40 ports
- next-gen FPGA with hardened NoC, e.g. Versal ACAP