The 25th International Conference on Field-Programmable Logic and Applications (FPL 2015)
September 3, 2015

Ultra-Fast NoC Emulation on a Single FPGA

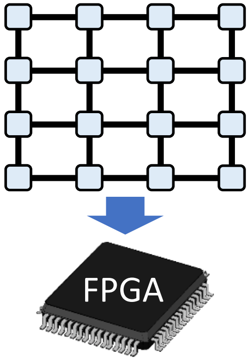
Thiem Van Chu, Shimpei Sato, and Kenji Kise

Tokyo Institute of Technology

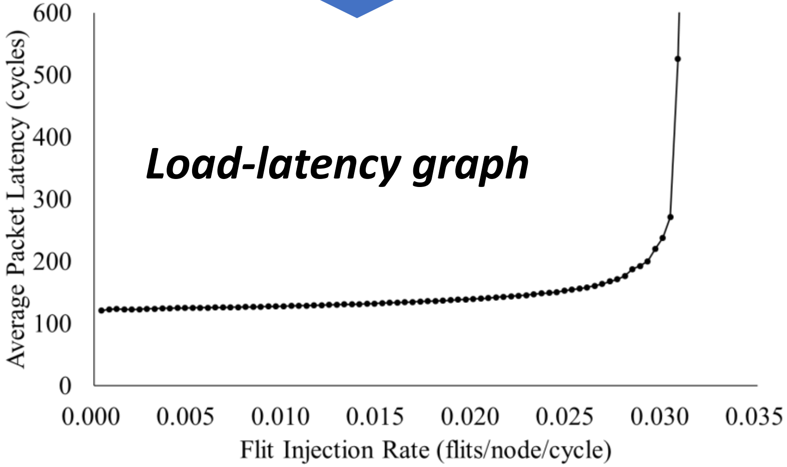
Contributions

Methodologies for emulating Network-on-Chip (NoC) architectures with up to 1000s of nodes on a single FPGA

Cycle-accurate & 5000x simulation speedup over BookSim[1], a widely used software simulator

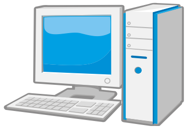


Up to ***16384 nodes***



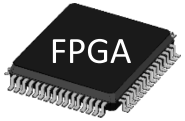
Load-latency graph

High end PC
Core i7 4770 CPU
32GB RAM



BookSim[1]: several ***days***

Proposal: several ***minutes***

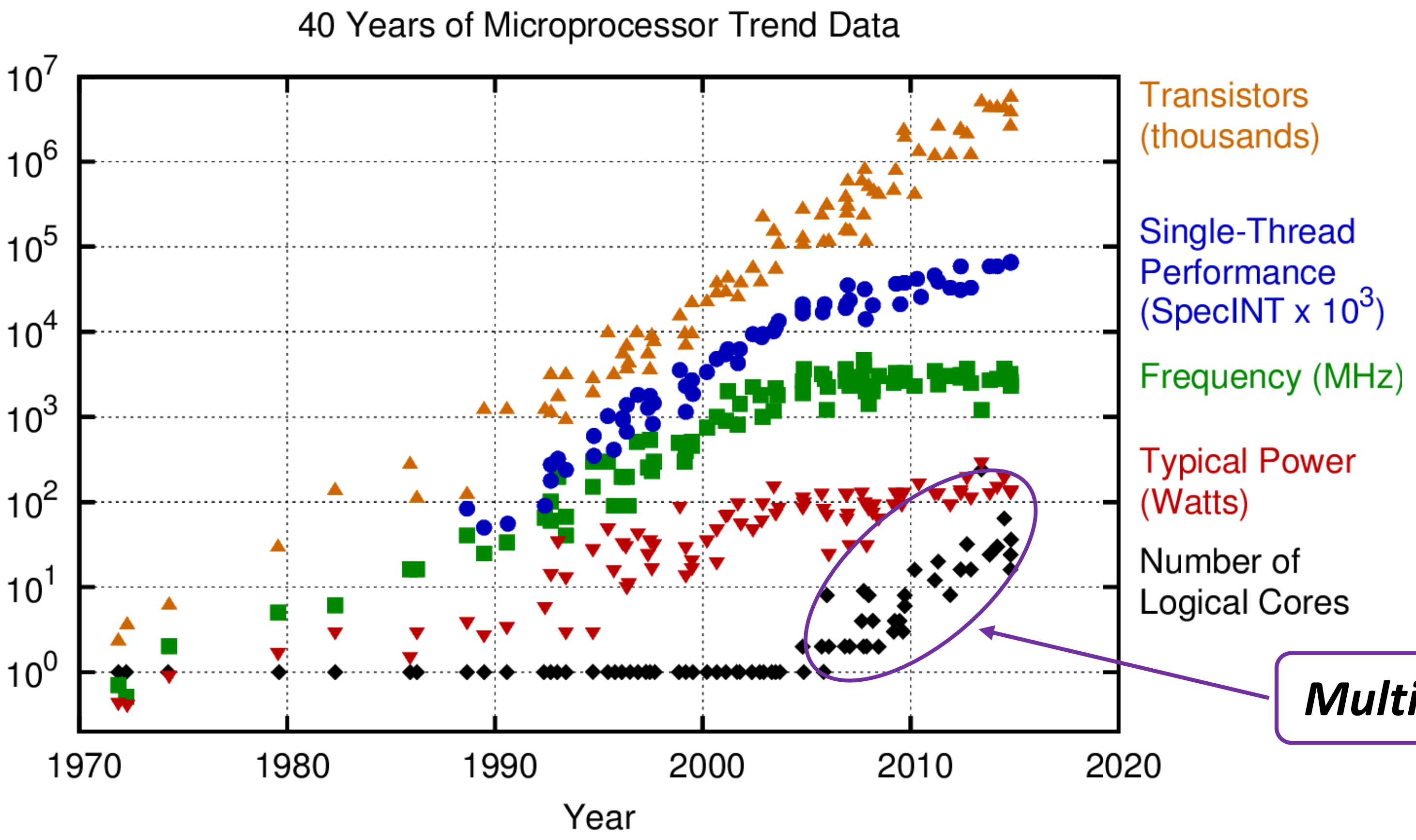


Virtex-7 FPGA

[1] http://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim

Multi/Many-Core Architectures Have Become Mainstream

1000s cores in near future architectures



Multi/many-core

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

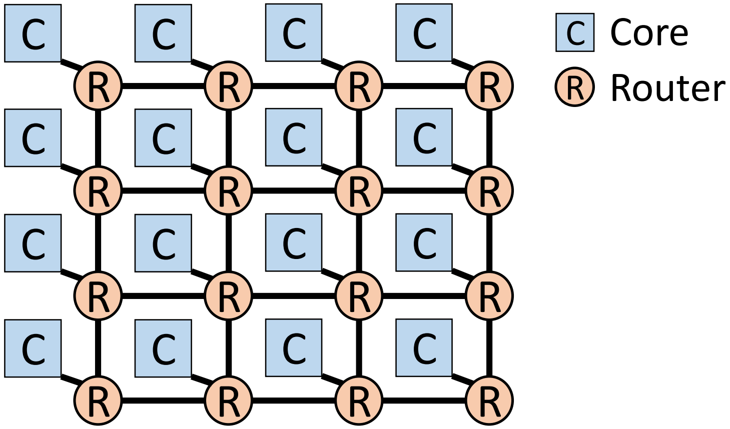
60+ cores

100 cores

Network-on-Chip (NoC)

- The interconnection network of many-core architectures

- NoC simulation plays a vital role in designing many-core architectures



A many-core architecture with 2D Mesh NoC

Software Simulators

- Flexible and easy to debug
- ***Too slow*** to simulate large architectures
- ***Parallelization is non-trivial***
  - Without sacrificing accuracy, only a limited degree of parallelization can be achieved

	Simple Models	Moderate Models	Detailed Models
Typical Simulation Time of 1000s-Core Architectures	Several ***days***	Several ***months***	Several ***years***

FPGA-Accelerated Simulation

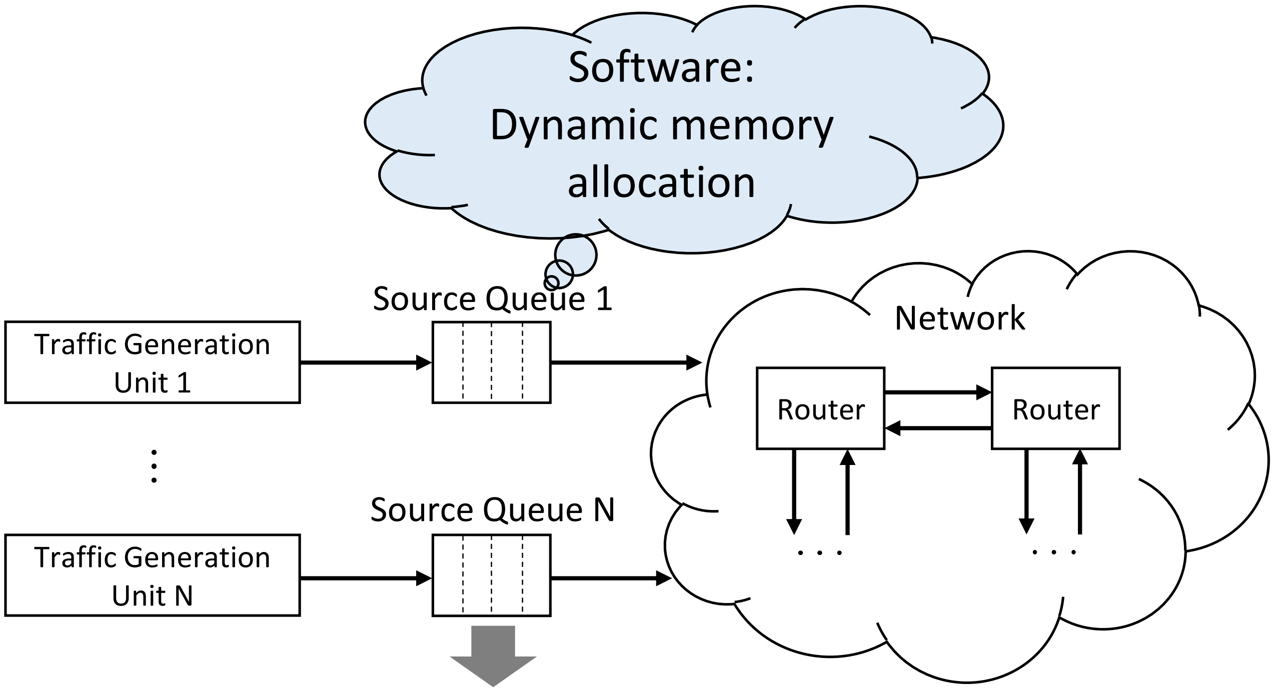
- Avoids impractical designs
- Possible to reuse RTL code
- Can achieve an ultra-fast simulation speed
  - Many operations can be simulated simultaneously in a tick of FPGA's clock
  - Adding detail to a model requires more hardware, but does not necessary degrade performance

FPGA-Accelerated Simulation

Challenges

Single FPGA →
- Limited resources

~~Multiple FPGAs~~
~~Off-chip memory~~ →
- More resources
- More complex
- Off-chip communication



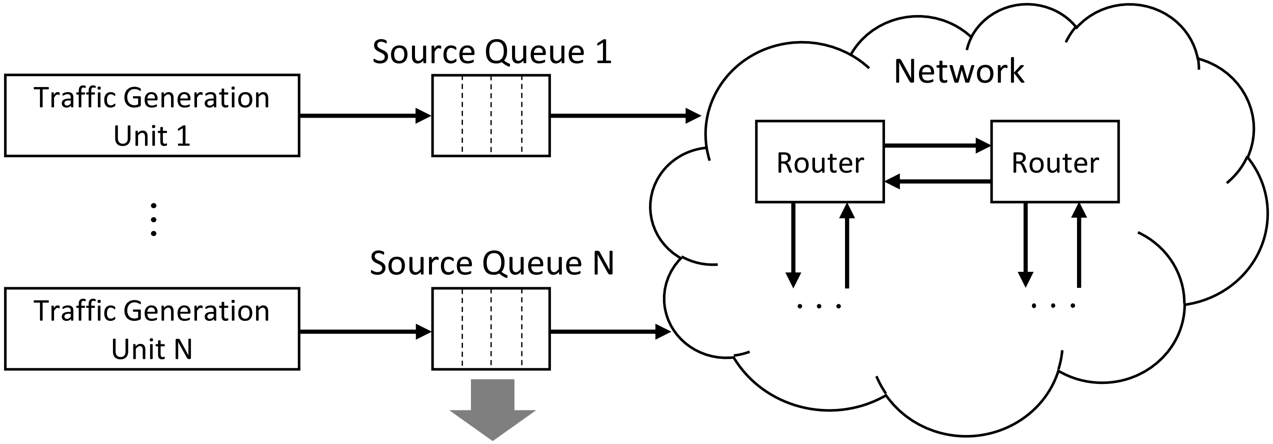
Software:
Dynamic memory allocation

Memory constraint

Must be ***very large*** to ensure that no generated packet is dropped

Proposals

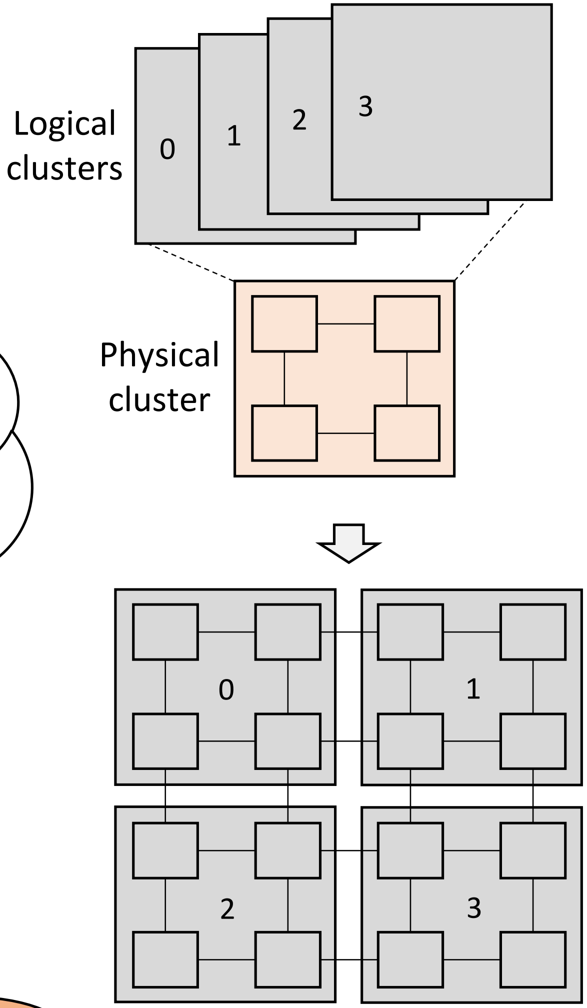
Method 1: Decoupling Time Counters



~~Must be ***very large*** to ensure that no generated packet is dropped~~

Eliminate the memory constraint

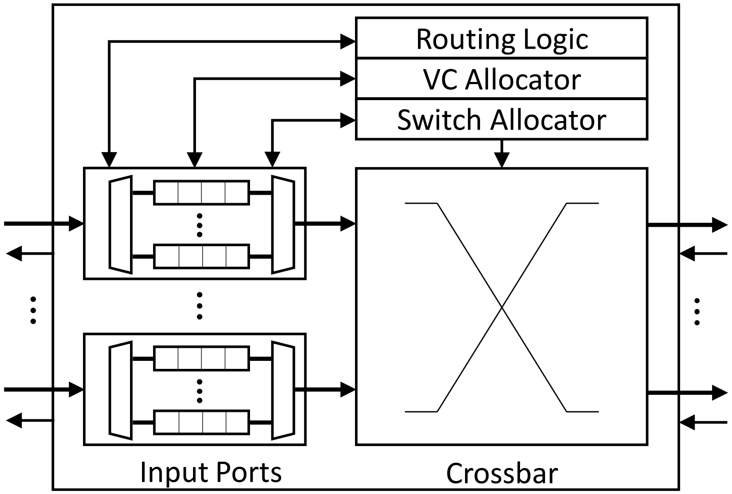
Method 2: Time-Multiplexing



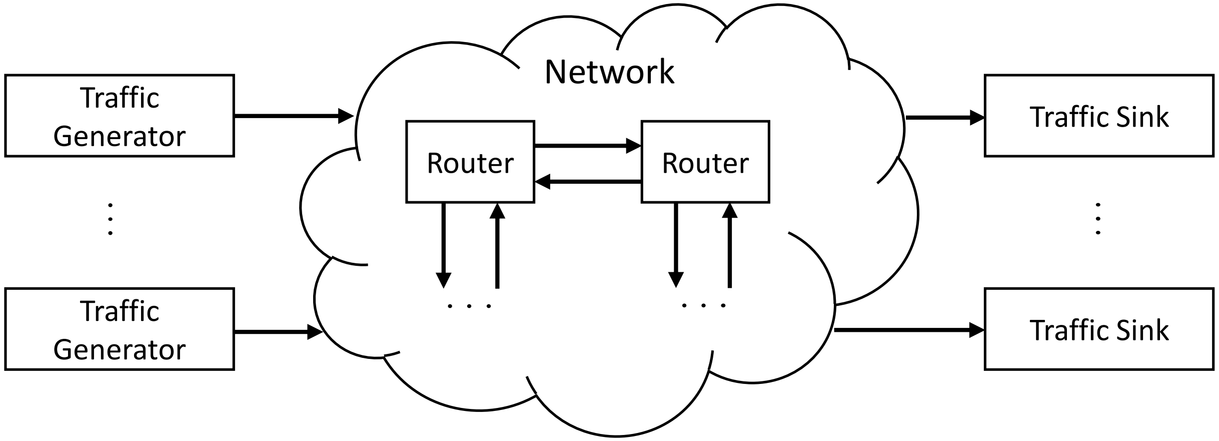
Simulate architectures with 1000s of cores on a single FPGA

Emulation Model

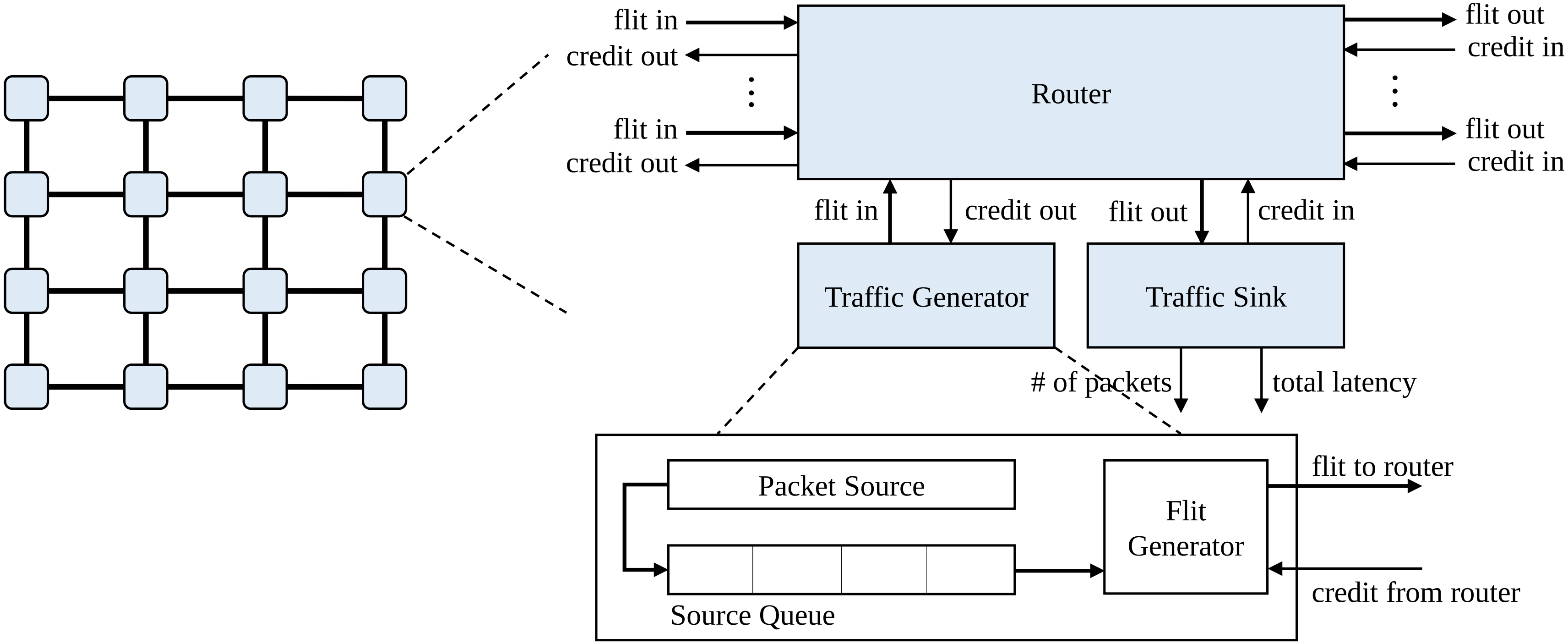
The primary router model



- Three basic components
  - ***Router***: a state-of-the-art pipelined router architecture
  - ***Traffic generator***: generates and injects synthetic workloads
  - ***Traffic sink***: collects performance characteristics



Emulation Model: 2D Mesh

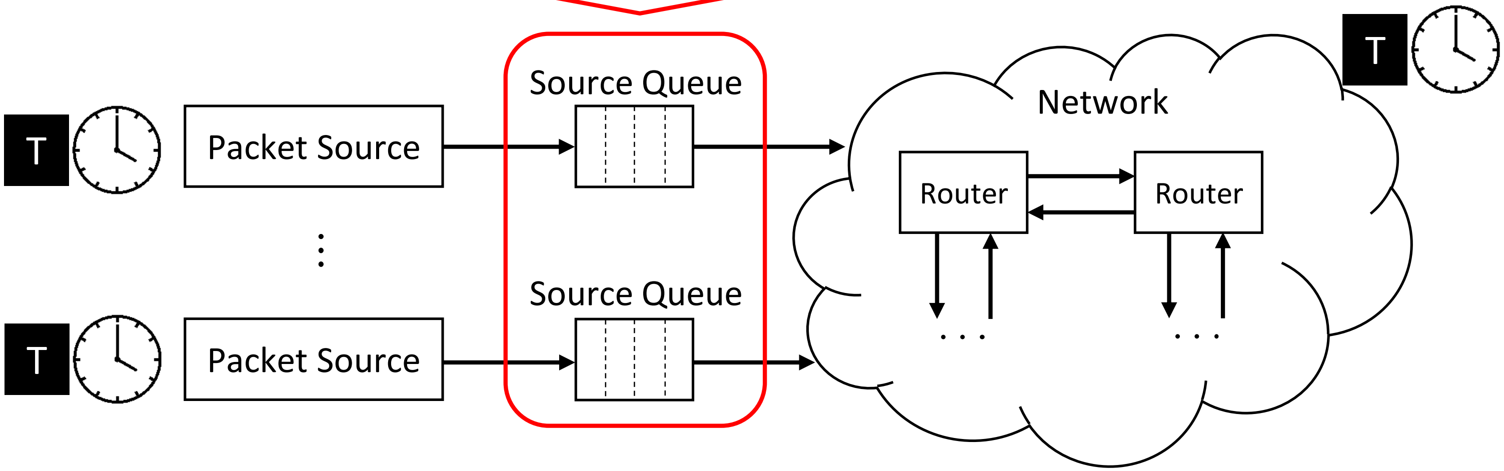


- ***Packet source***: models injection processes (e.g. Bernoulli process)
- ***Source queue***: every packet generated by the packet source is stored in the source queue until it can enter the network
- ***Flit generator***: models traffic patterns (e.g. uniform random)

Decoupling Time Counters

- ***Conventional approach***
  - Every packet source is synchronized with the network

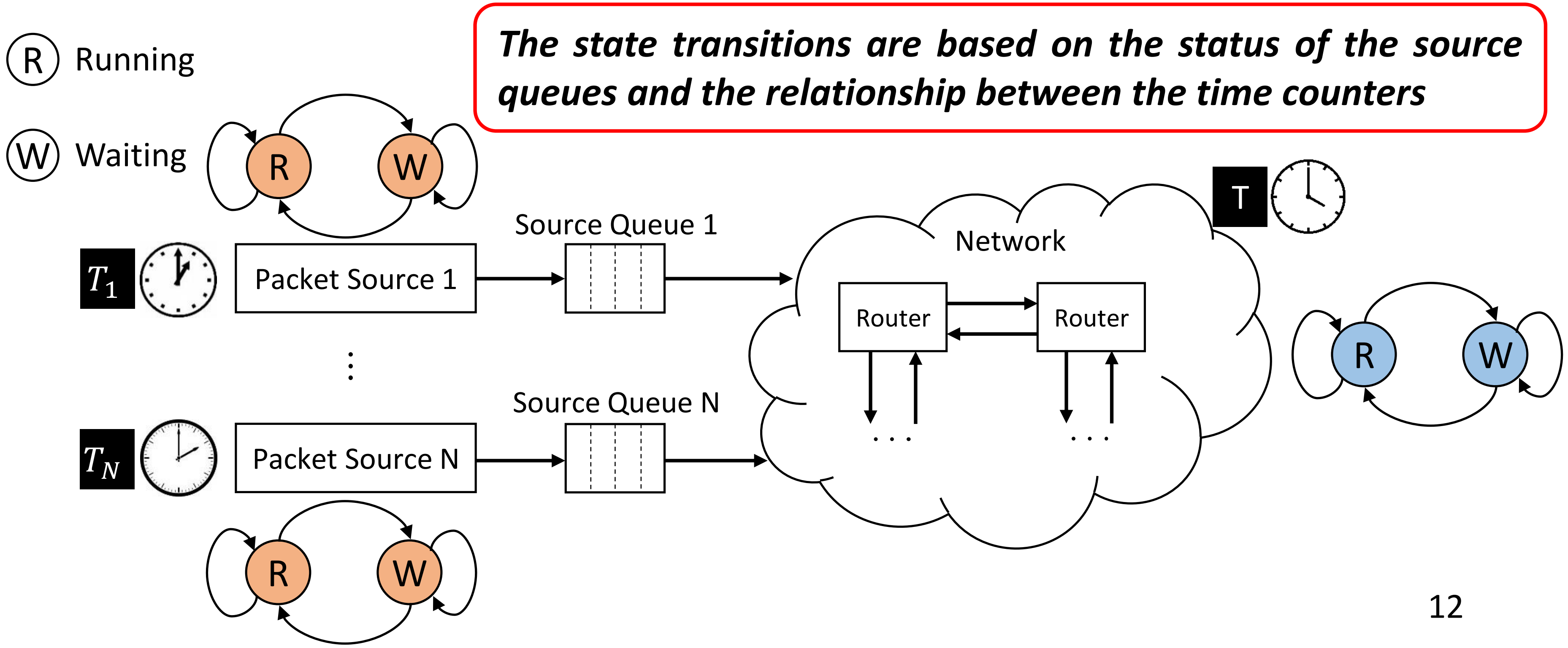
The source queues must be very large to cope with the case when the packet sources generate so many packets that the network becomes very congested



Decoupling Time Counters

- ***Proposal***
  - Each packet source, as well as the network, has its own time counter and operates based on a separate state machine

The state transitions are based on the status of the source queues and the relationship between the time counters



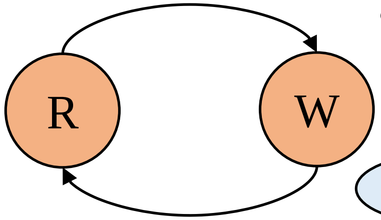
Decoupling Time Counters

W Waiting

R Running

State Machine of Packet Source i

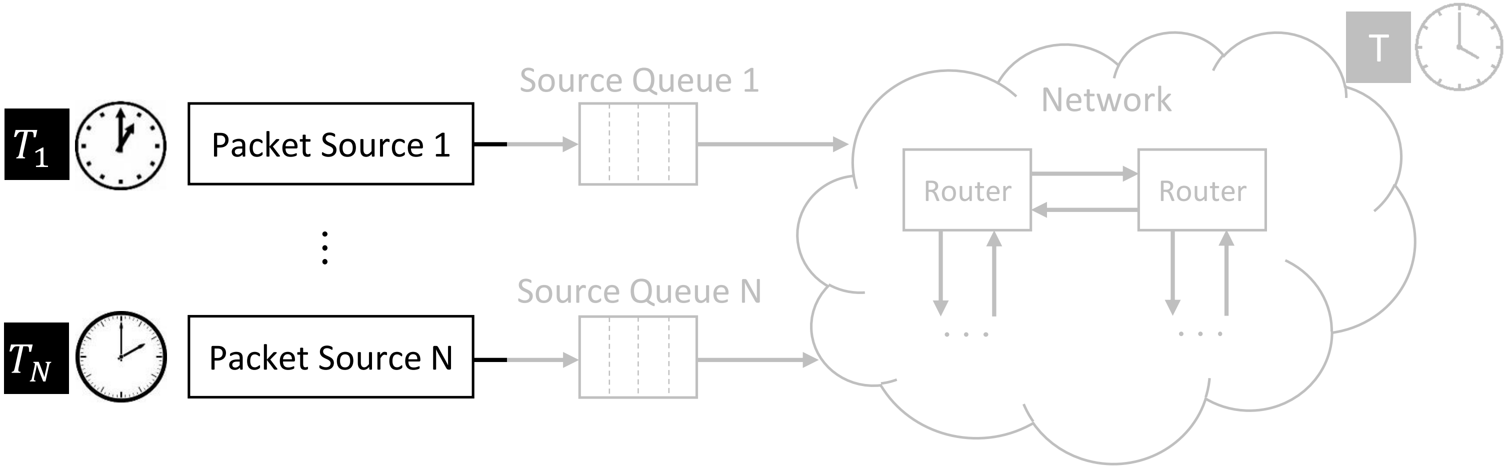
PS_i intends to generate a packet but SQ_i is full



One space in SQ_i becomes available

No packet is dropped

But..
Some packets may not be generated on time

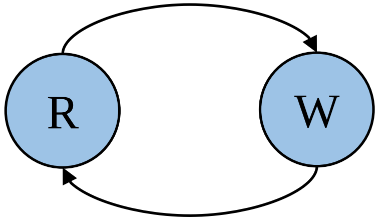


Decoupling Time Counters

W Waiting

R Running

$\exists i \;\; SQ_i$ is empty & $T_i < T$

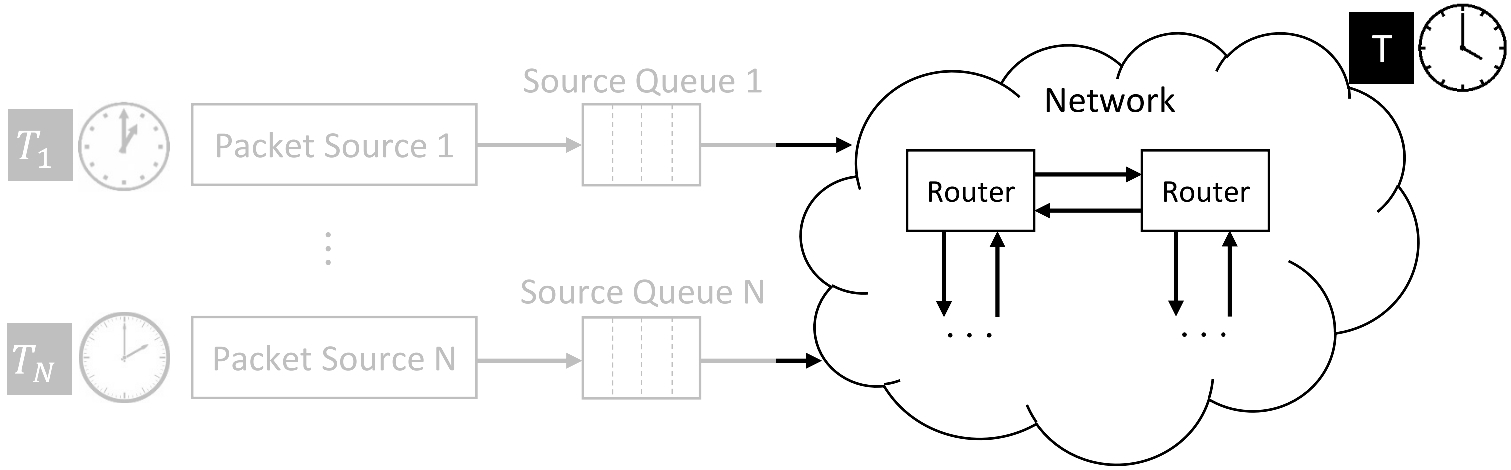


$\forall i \;\; SQ_i$ is NOT empty
OR $T_i = T$

State Machine of the Network

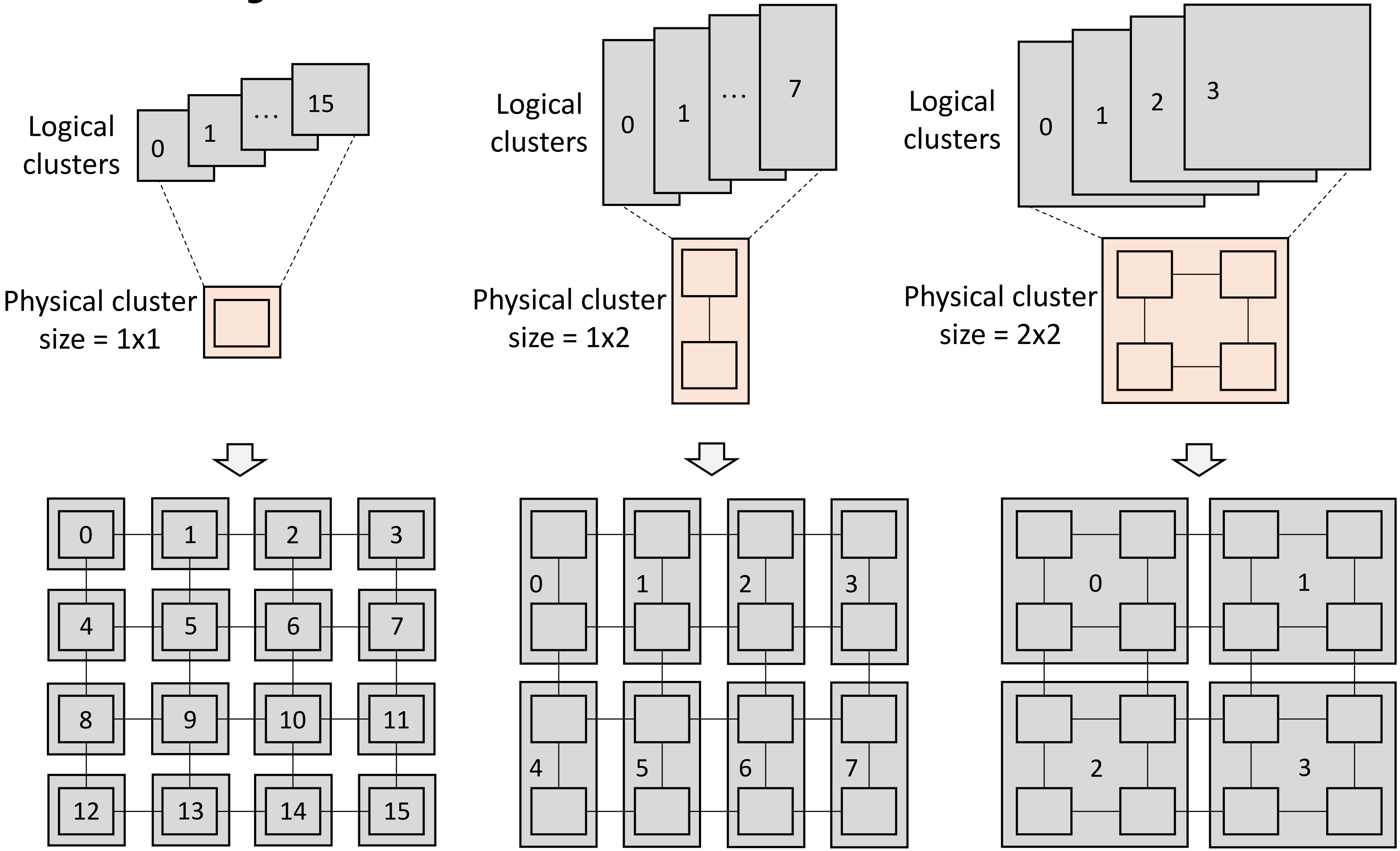
$\forall i$ either SQ_i contains at least one packet or the time of PS_i is synchronized with the time of the network

The specified synthetic workload is simulated accurately



Time-Multiplexing

To complete one cycle of the network, the ***physical cluster*** sequentially emulates a number of ***logical clusters***



Logical clusters

Physical cluster size = 1x1

Logical clusters

Physical cluster size = 1x2

Logical clusters

Physical cluster size = 2x2

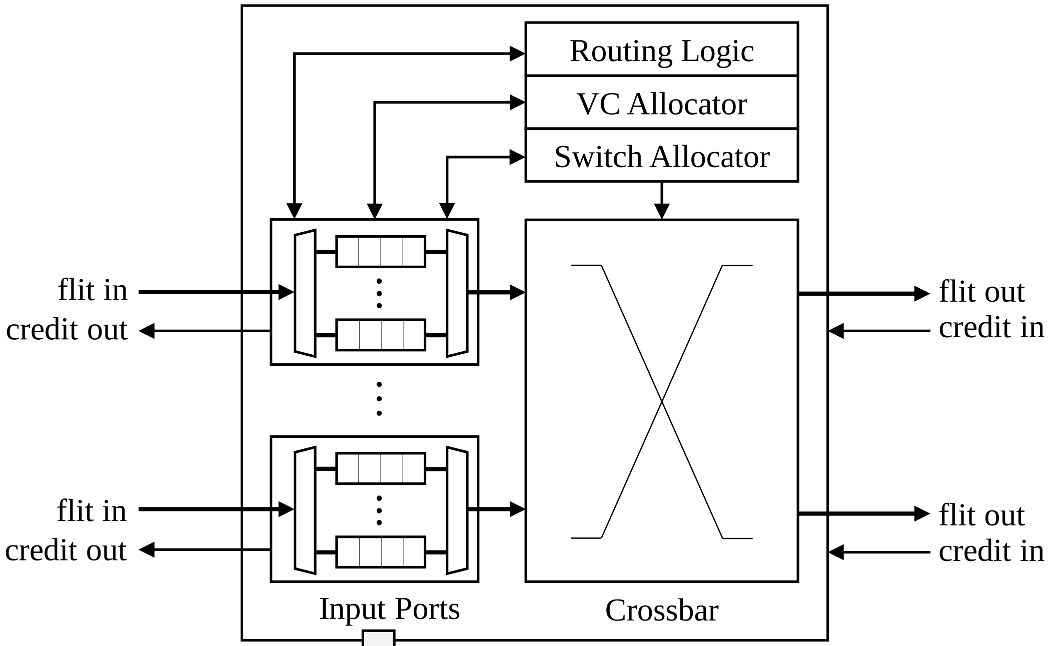
Time-Multiplexing

- ***Combinational logic*** and ***block RAMs (BRAMs)*** are utilized much more efficiently because they can be shared between many NoC nodes
- Example: 128x128 mesh NoC

Direct implementation
5x128x128 = ***81,920 BRAMs***

Using time-multiplexing
< 500 BRAMs

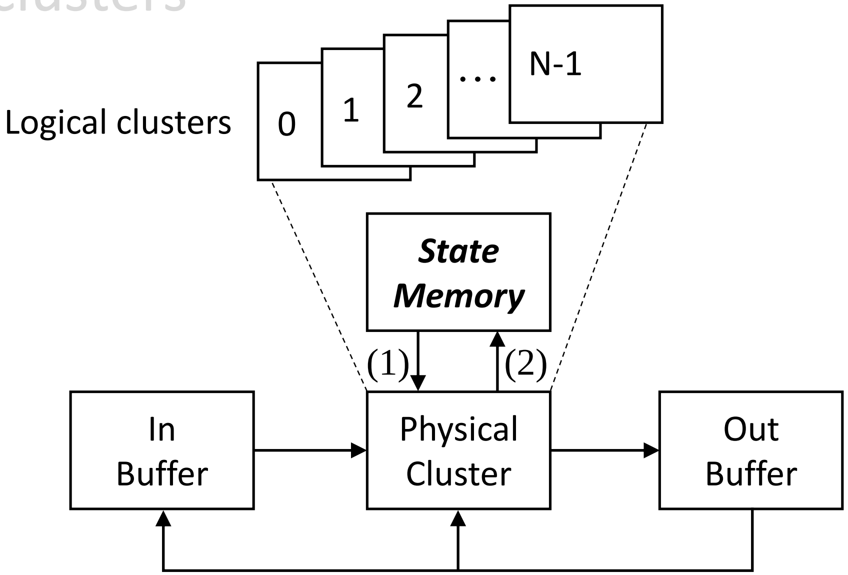
Method for translating to emulation code
... Please see the paper



Input buffers can be implemented using BRAMs

Datapath

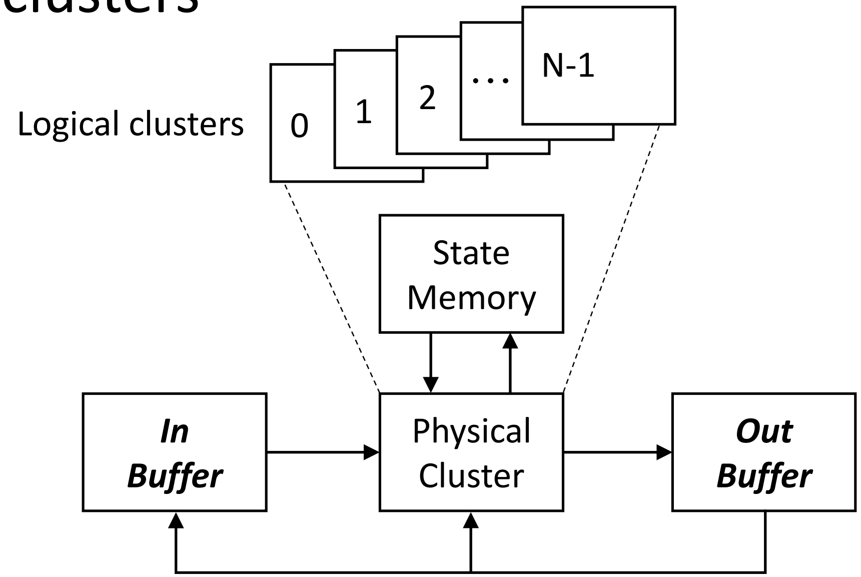
- Physical cluster emulates different logical clusters using different states loaded from ***state memory***
- In buffer and out buffer store data passed between logical clusters



(1) Load a state & Emulate the corresponding logical cluster
(2) Store the updated state

Datapath

- Physical cluster emulates different logical clusters using different states loaded from state memory
- ***In buffer*** and ***out buffer*** store data passed between logical clusters



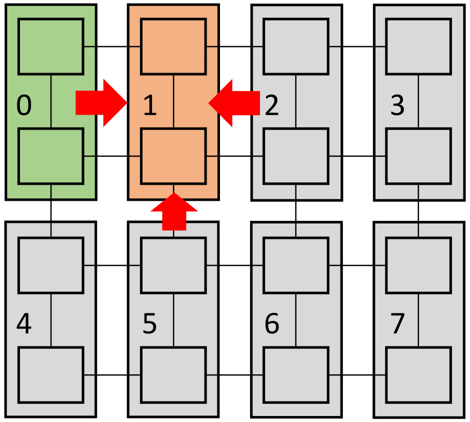
Example

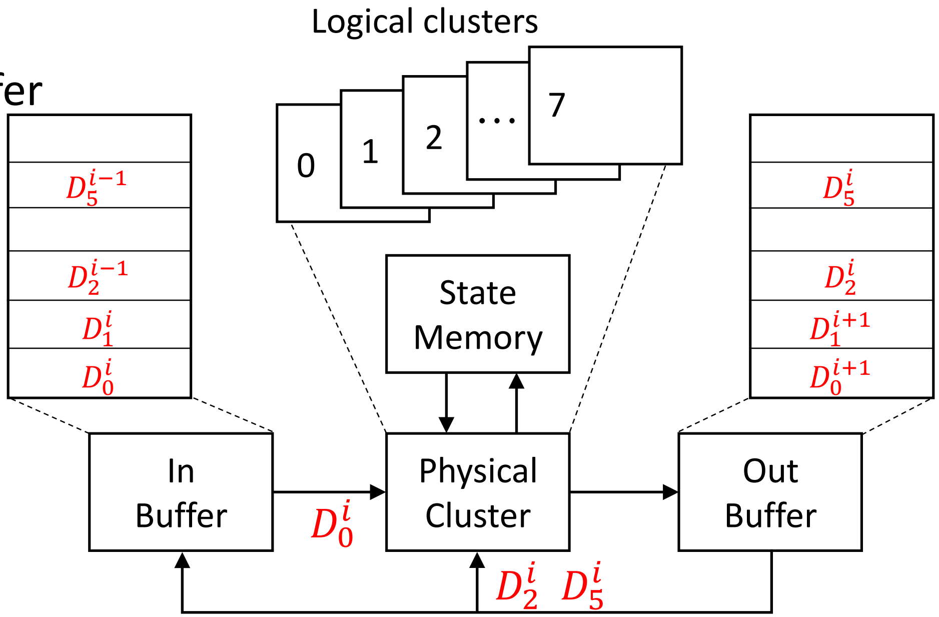
🟩 Emulated 🟧 Emulating ⬜ To be emulated

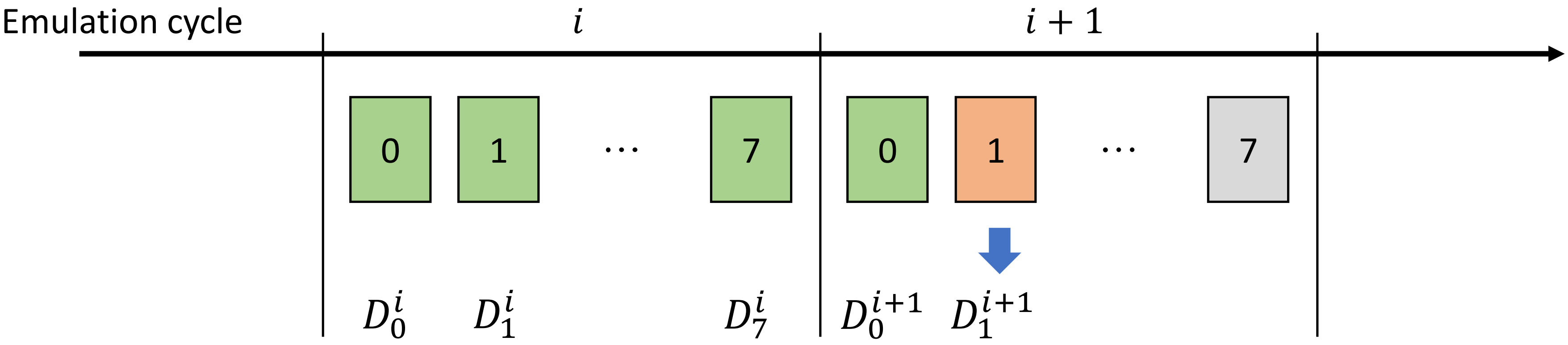
Need D_0^i D_2^i D_5^i

D_0^i : In Buffer

D_2^i D_5^i : Out Buffer







Xilinx VC707 board

Evaluation and Analysis

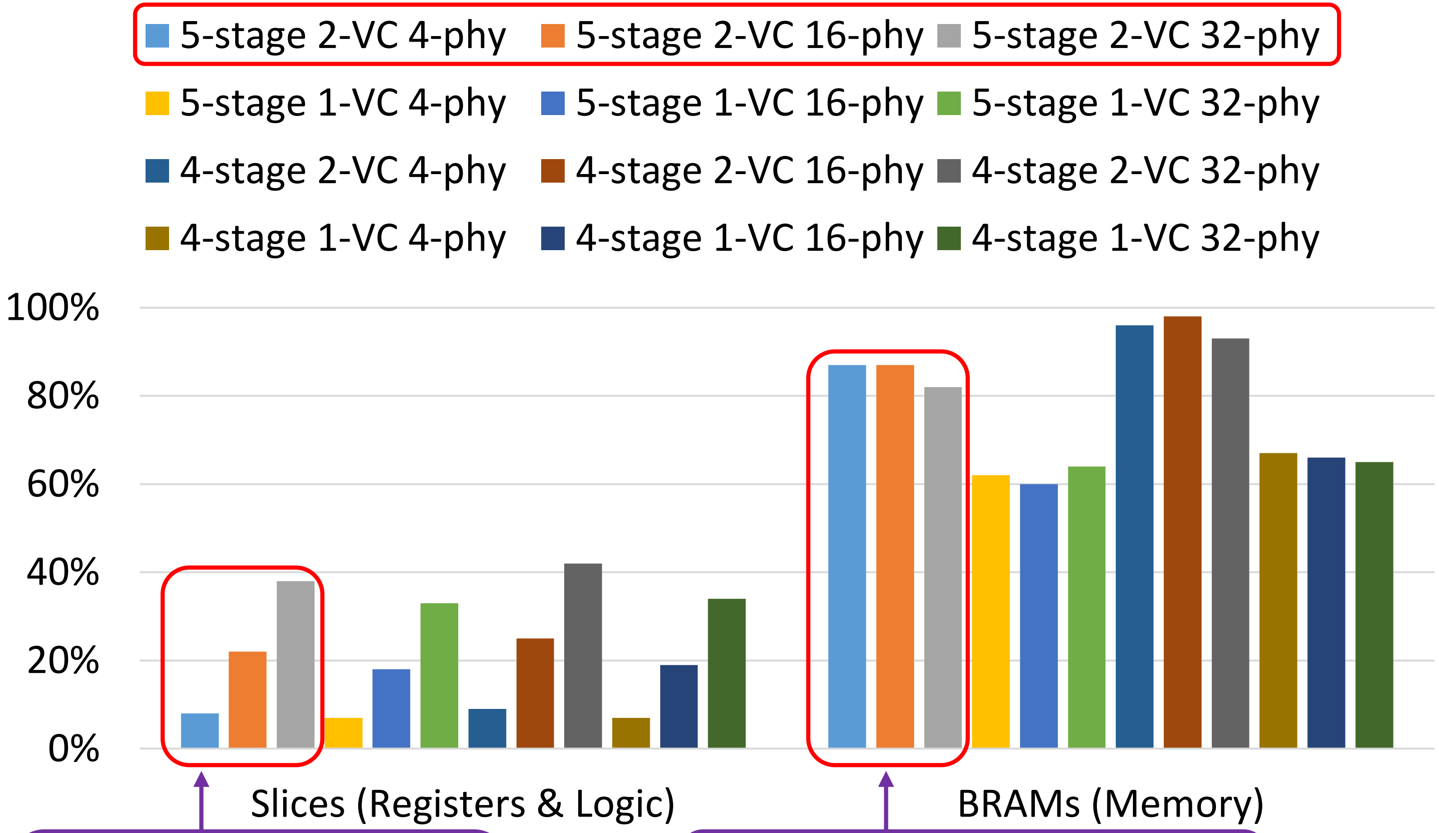
- 128×128 mesh NoC (***16,384 nodes***) on a Xilinx VC707 board
- Four NoC designs
  - ***5-stage 2-VC***: canonical 5-stage pipelined VC router architecture with 2 VCs per port
  - ***5-stage 1-VC***: canonical 5-stage pipelined VC router architecture with 1 VC per port
  - ***4-stage 2-VC***: canonical 4-stage pipelined VC router architecture with 2 VCs per port
  - ***4-stage 1-VC***: canonical 4-stage pipelined VC router architecture with 1 VC per port
- Three configurations
  - ***4-phy*** (2×2): use four nodes to emulate the entire 128×128 mesh network
  - ***16-phy*** (4×4): use 16 nodes to emulate the entire 128×128 mesh network
  - ***32-phy*** (8×4): use 32 nodes to emulate the entire 128×128 mesh network
- Metrics
  - Hardware usage
  - Verification against BookSim, a widely used cycle-accurate software simulator
  - Simulation performance: speedup over BookSim

Configuration Parameters

Topology	128x128 mesh ***(16,384 nodes)***
Router architecture	5-stage pipelined VC router ***or*** 4-stage pipelined VC router (employing look-ahead routing)
# of VCs per port	2 ***or*** 1
Routing algorithm	Dimension-order (XY)
Flow control	Credit-based
VC/Switch allocator	Separable output first
Arbiter type	Fixed priority
Flit size	25-bit ***or*** 22-bit
VC size	4-flit
Packet length	8-flit
Injection process	Bernoulli
Traffic pattern	Uniform random
Source queue length	8-entry

Hardware Usage

Simulation of a same 128 × 128 NoC design by using 4 physical nodes (4-phy), 16 physical nodes (16-phy), and 32 physical nodes (32-phy)



Use more physical nodes → Consume more registers & logic

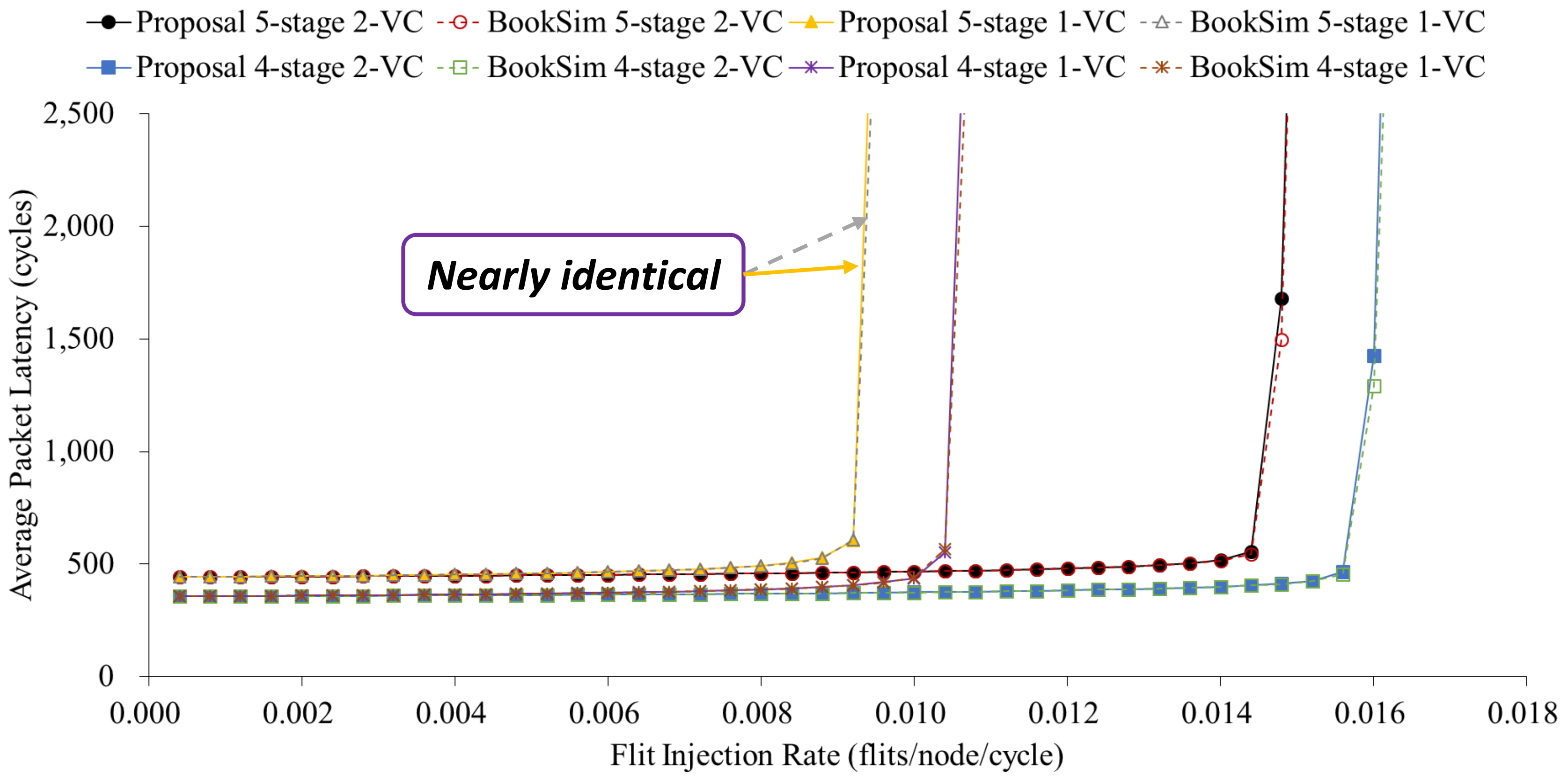
Only a minor difference in the number of occupied BRAMs

Accuracy

- The proposed methods do not affect the simulation accuracy
  - Synthetic workloads can be modeled accurately without using a large amount of memory
  - No compromise in simulation accuracy is made
- Verification
  - Compare the output results in simulating 4 NoC designs of the FPGA-based emulator and BookSim
    - ***5-stage 2-VC***
    - ***5-stage 1-VC***
    - ***4-stage 2-VC***
    - ***4-stage 1-VC***

Verification: Proposal vs BookSim

Solid Lines: Proposal (FPGA-based)
Dotted Lines: BookSim (Software-based)

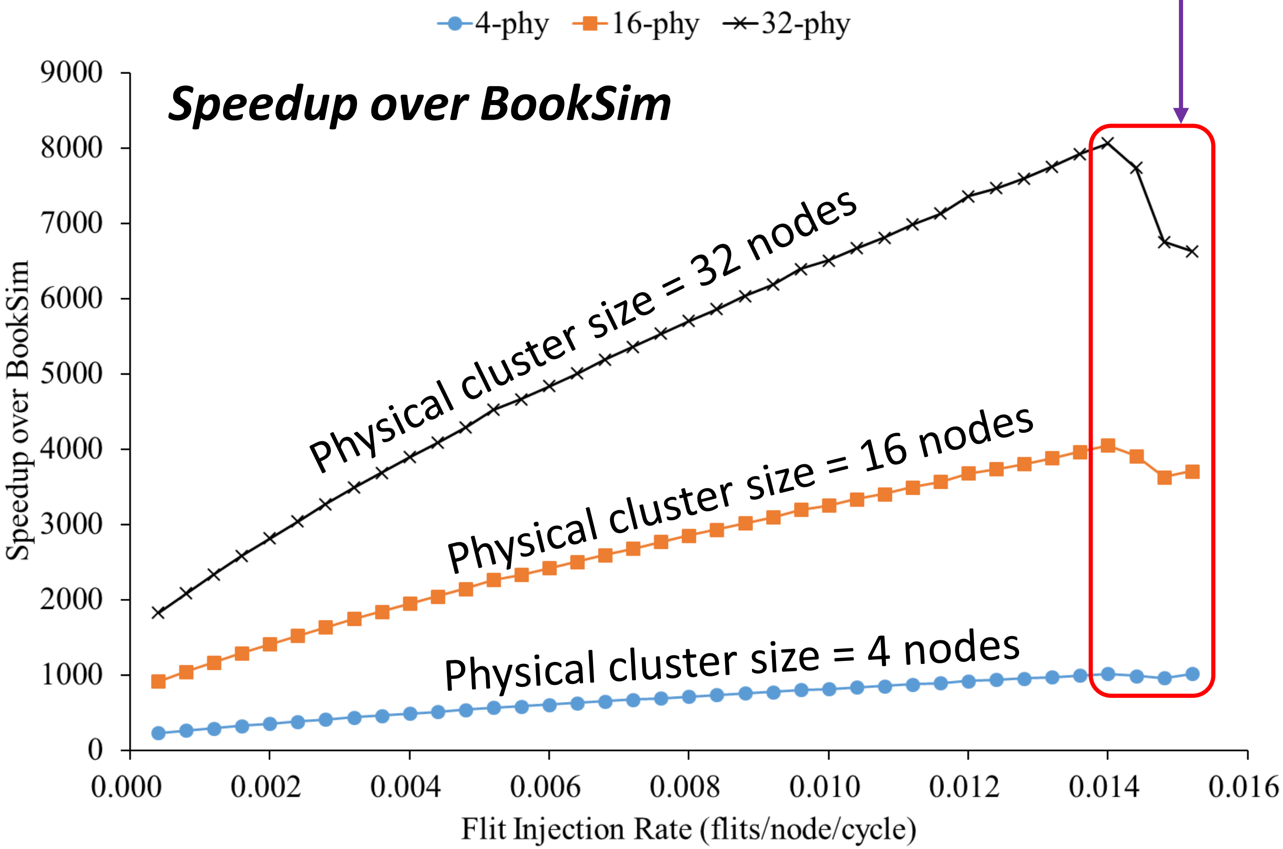


Nearly identical

Simulation Performance

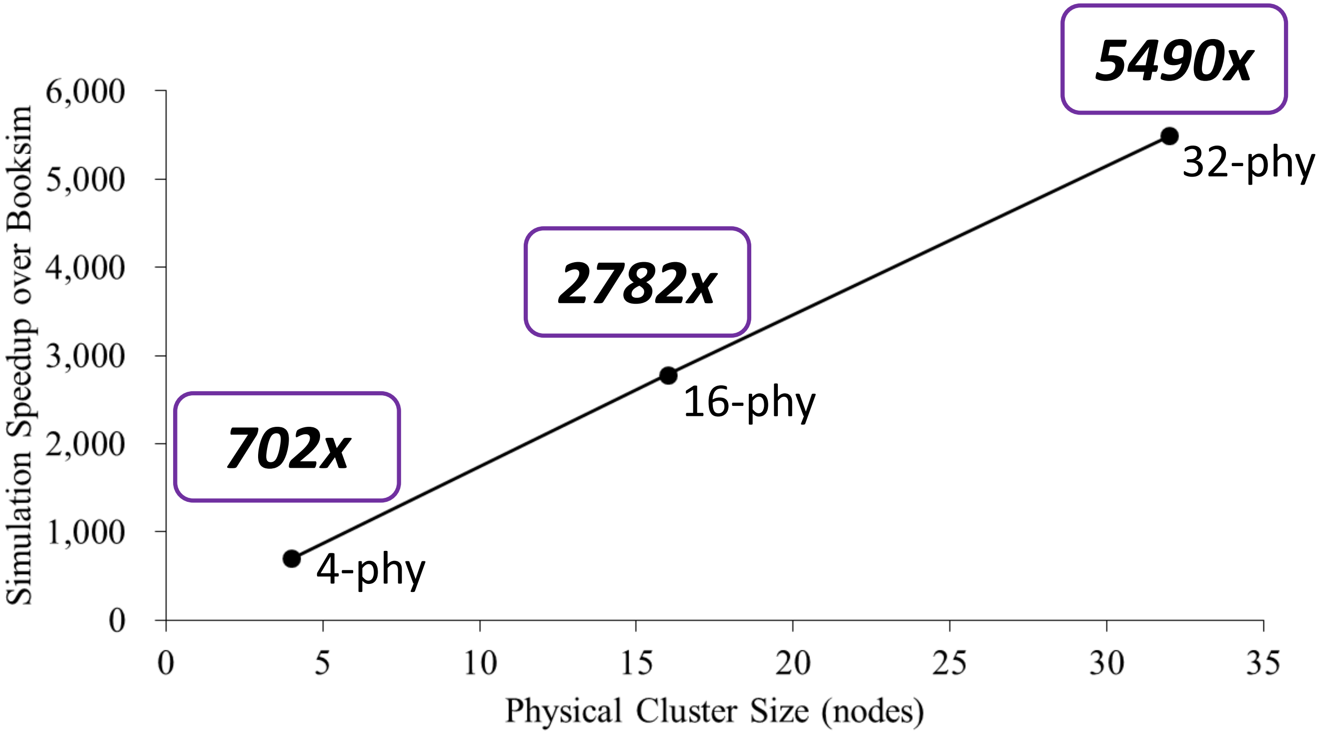
Topology	128x128 mesh
Router architecture	5-stage
# of VCs per port	2

The drop is caused by stalling the emulated network in the first proposed method which helps to eliminate the memory constraint



(Amount of Network Activity)

Simulation Performance

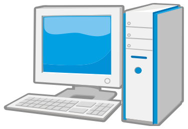


702x

2782x

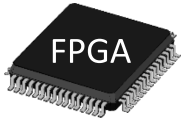
5490x

High end PC
Core i7 4770 CPU
32GB RAM



BookSim: ≈***5.5 days***

Proposal: ≈ ***1.5 minutes***



Virtex-7 FPGA

Conclusions & Future Work

- Conclusions
  - Two methods are proposed to enable ultra-fast and accurate emulation of large-scale NoC architectures on a single FPGA
  - ***More than 5000x simulation speedup over BookSim*** is achieved when emulating an 128x128 NoC with state-of-the-art router architectures
- Future work
  - Support full-system simulations
  - Support a wide range of benchmarks/workloads

Q & A

Thank you!