Evaluating Ruche Networks: Physically Scalable, Cost-Effective, Bandwidth-Flexible NoCs

Dai Cheol “Tommy” Jung, Michael Taylor

University of Washington

ISCA 2025

Network on Chips

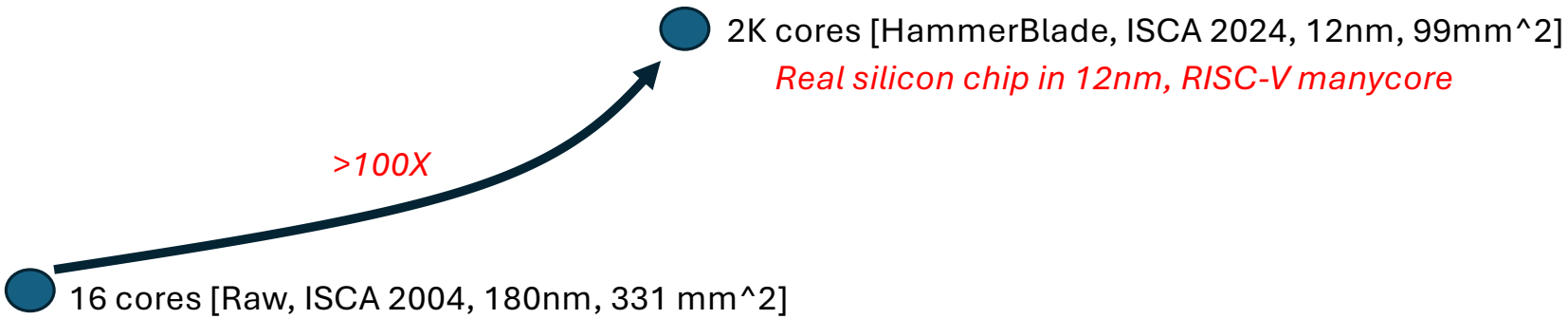
Are reaching a new level of scale

16 cores [Raw, ISCA 2004, 180nm, 331 mm^2]

Early manycore processor with NOC

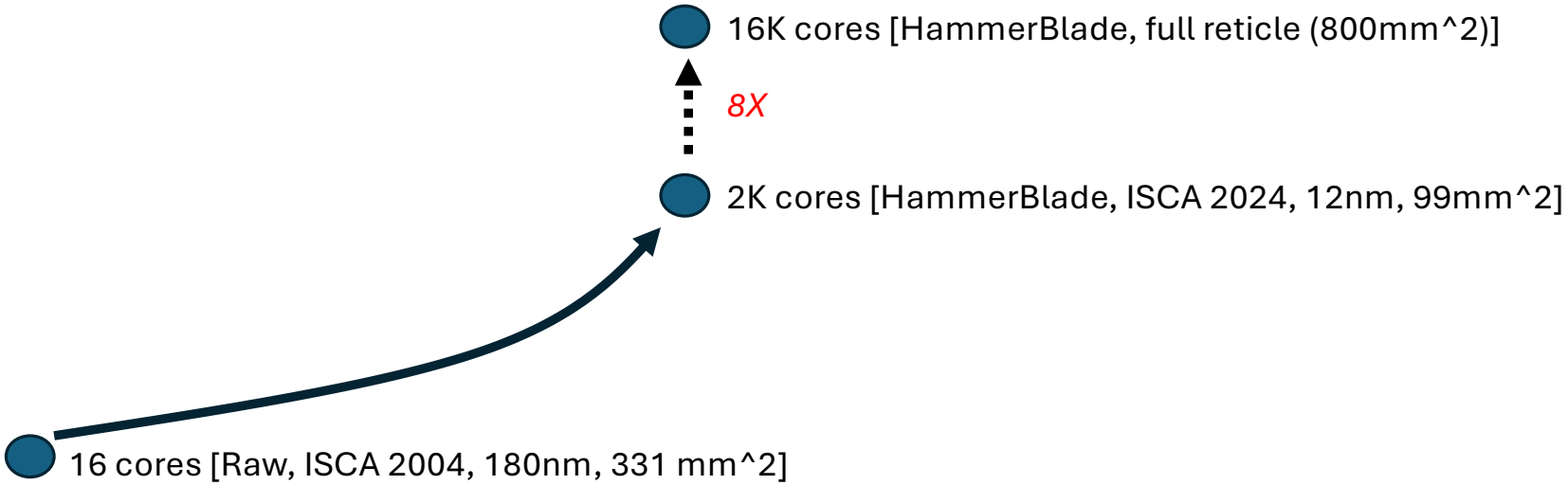
Network on Chips

Are reaching a new level of scale



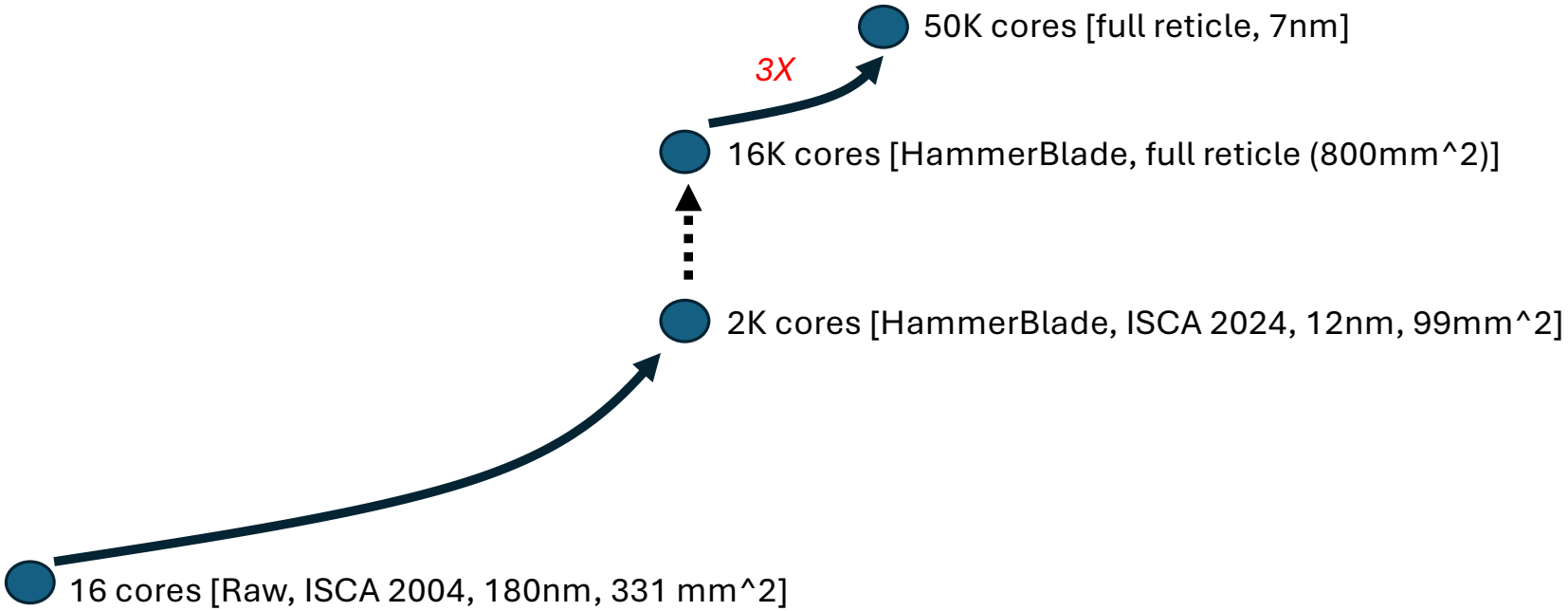
Network on Chips

Are reaching a new level of scale



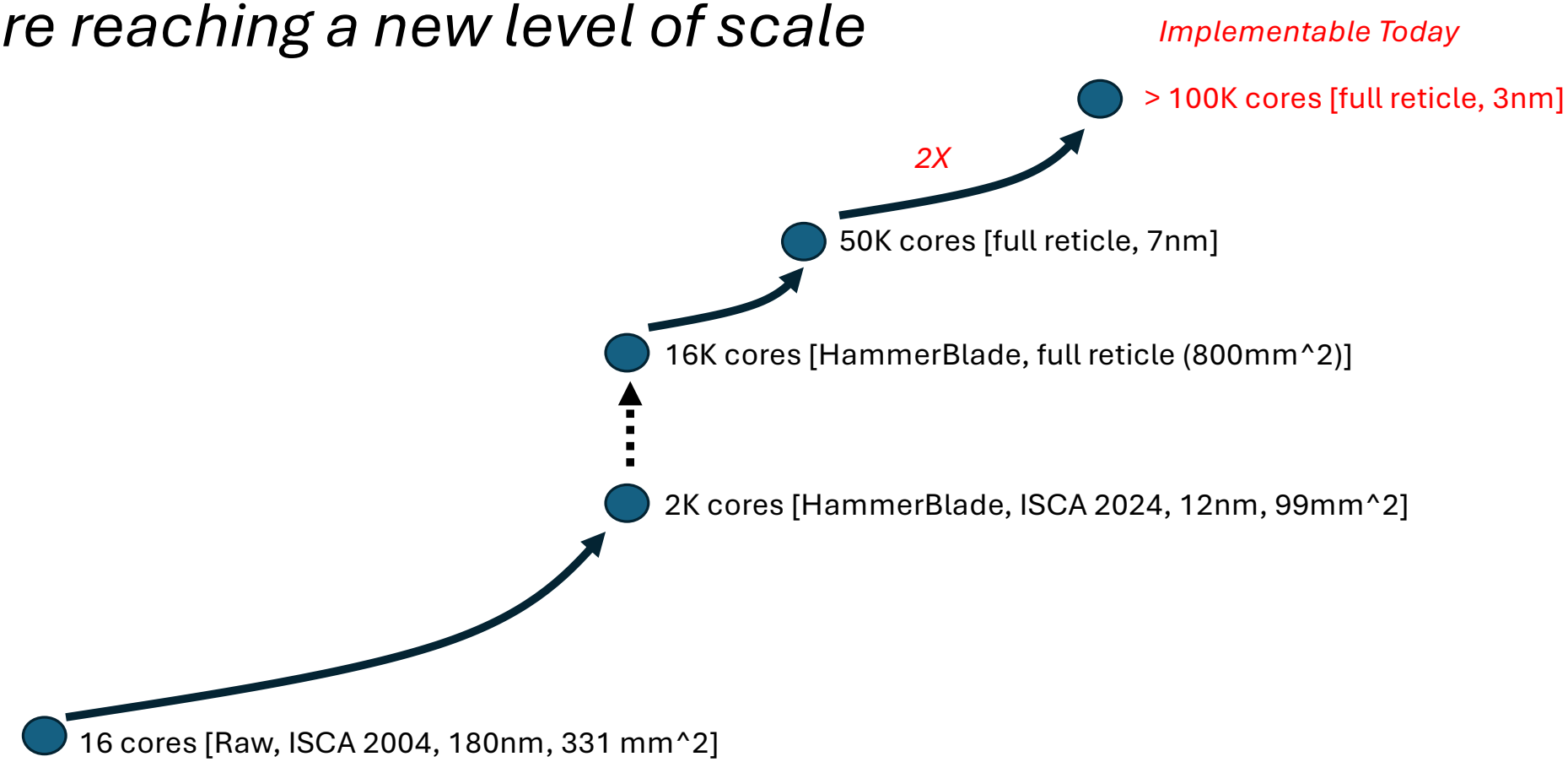
Network on Chips

Are reaching a new level of scale



Network on Chips

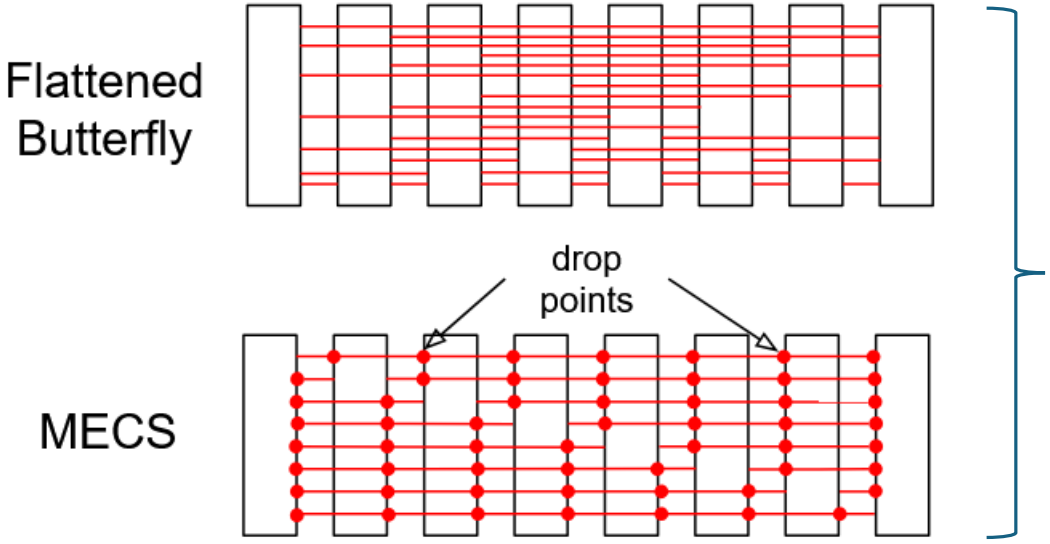
Are reaching a new level of scale



Important Properties for Large NoCs: Physical Scalability

As network scales up, we want

- unchanging wire lengths between nodes to control wire delay (critical path)
- unchanging small input crossbars that can be routed easily (routability)
- unchanging number of wires per node as you scale up the number of nodes



Example “advanced” NOCs that lack physical scalability.

Important Properties for Large NoCs: Tileable Design

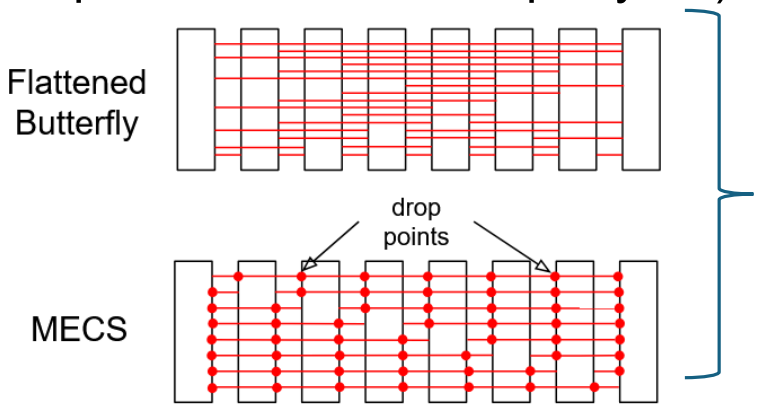
– one layout that can be replicated to scale up the design
  – transistors, wires and all.

– repeatedly stamp out a small variety of small layouts
  e.g. not 100,000 unique cores; or 1 giant manycore that is 100,000X bigger

– a small core can often take 10 hours to synthesize, place and route on 16 X86 CPUs.

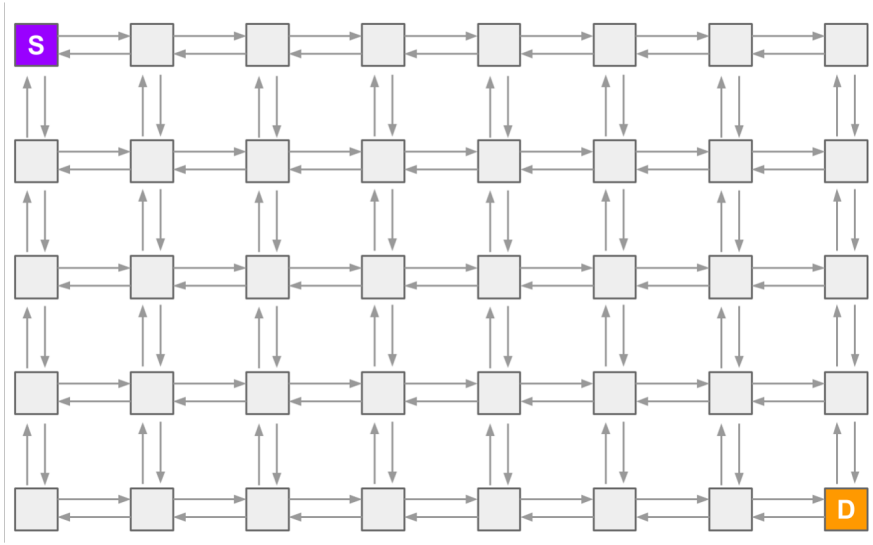
– a complete layout needs to be done many times to converge the design.

– except for the largest companies, companies are limited by CAD tool licenses
  ($300K per 16-cpu software license per year)



Each core would need
a unique layout
in these designs

Not surprisingly, most Chip NoCs are 2D meshes because they are physically scalable and tileable



Tesla DOJO
Esperanto ET-SoC-1
Meta MTIA 8x8
Celerity
OpenPiton
KiloCore

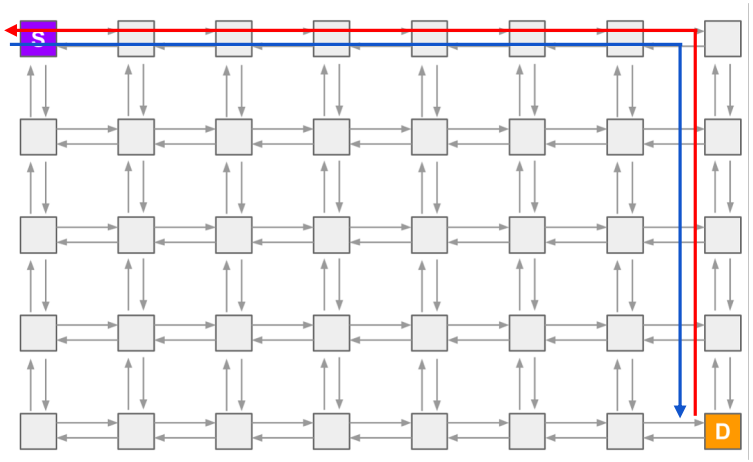
Epiphany-V
Execution Migration Machine
TILE64
Intel Teraflops
TRIPS processor
MIT Raw

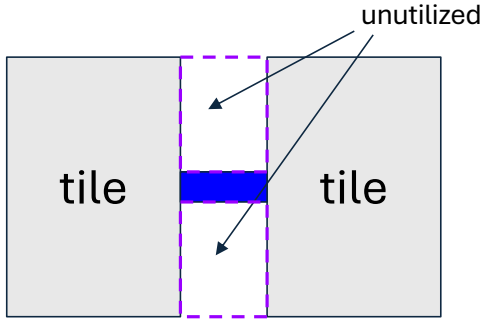
If 2D Mesh Networks are so great for physical design, why are other topologies a topic of repeated research?

Packets are limited to **one hop per cycle**, even though **the raw speed of wires** can be 3-5x faster.

– As the number of nodes increases:
  - **network diameter** becomes substantial.
  - **bisection bandwidth** becomes more and more important

– The natural width of the mesh channel often does not fully utilize the available number of wiring tracks between neighbor tiles.





Wider Mesh Links

One option that is proposed in the literature is to use wider links to increase wiring track utilization.

However, this has many side effects that are often ignored:

- Microarchitectural upsizing
  e.g. To supply a link with 8X the width, Cache SRAMs need 8X the sense amps → greater area and power

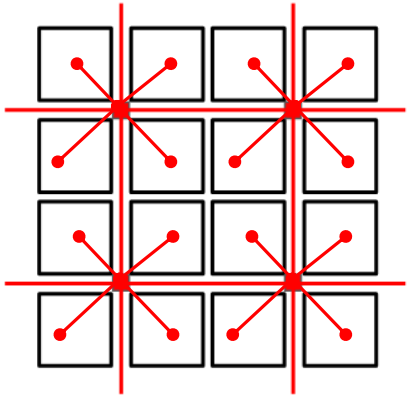
- Serialization/Deserialization Latencies/Areas
  e.g. Must accumulate enough data to send; most hold data and serialize into narrower interface → greater latency (+8 cycles)

- No improvement for network diameter / latency

Concentration for Meshes

Another idea that is often proposed is concentration, where groups of multiple tiles are attached to a single network node, reducing the network diameter.

Negative side-effects:



Reduces the bisection bandwidth of the network, unless wider mesh links are used (see prev slide)

Reduces the collective injection bandwidth of the nodes that are sharing routers

In practice, reduces wiring track available for long-distance routing, since more resources are used for local routing, and node local routing tracks are often left unused.

Ruche Networks:

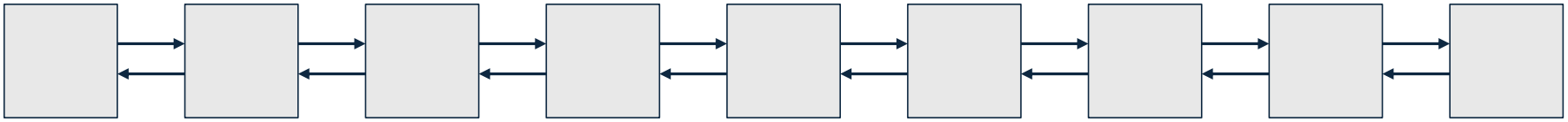
Are a wire-maximal NoC that considers Physical Design as a first class constraint

Can convert excess wiring tracks into network performance

Propagate packets at closer to the full speed of wire (lower diameter)

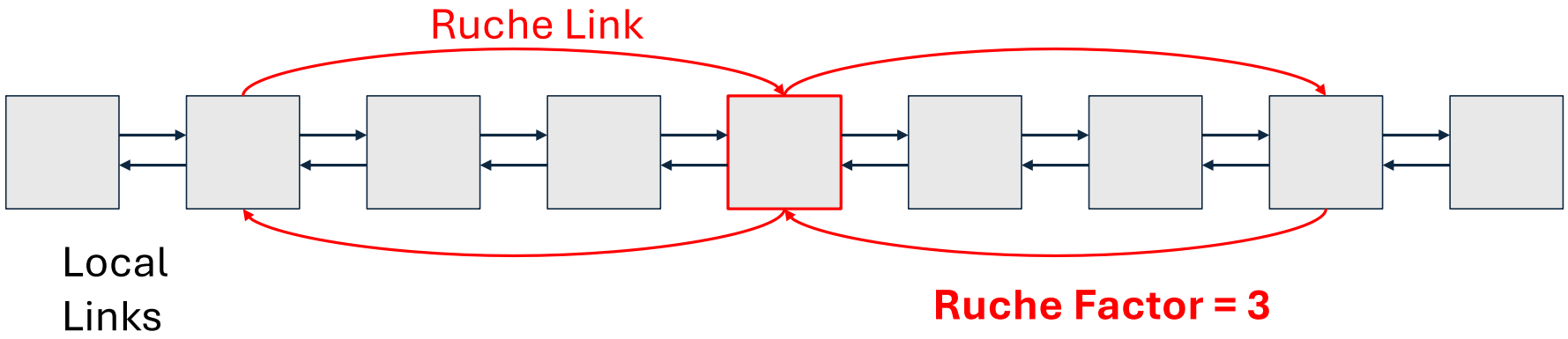
Have low hardware complexity (critical path + routability)

Traditional Mesh Network (in 1D)



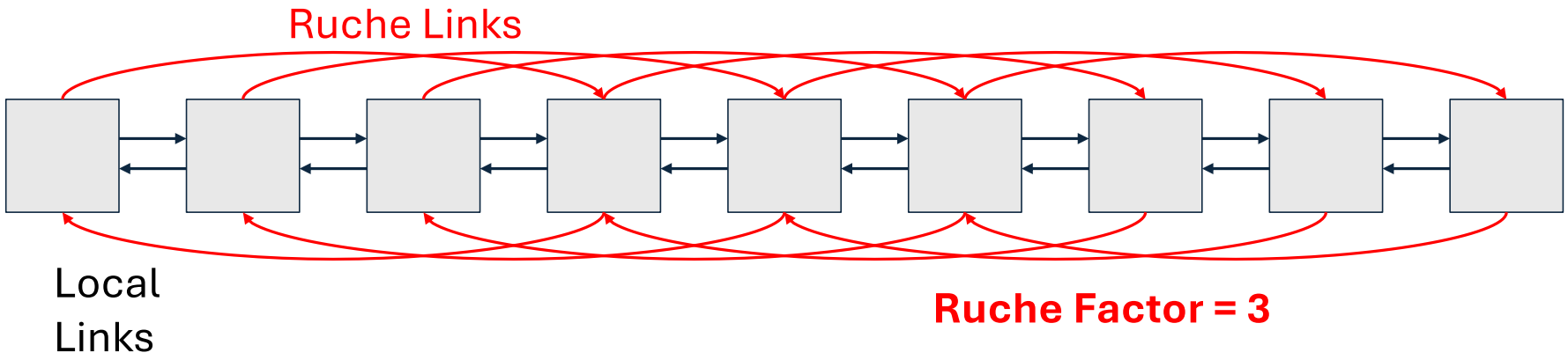
Local
Links

Augmenting a Node with Ruche Links



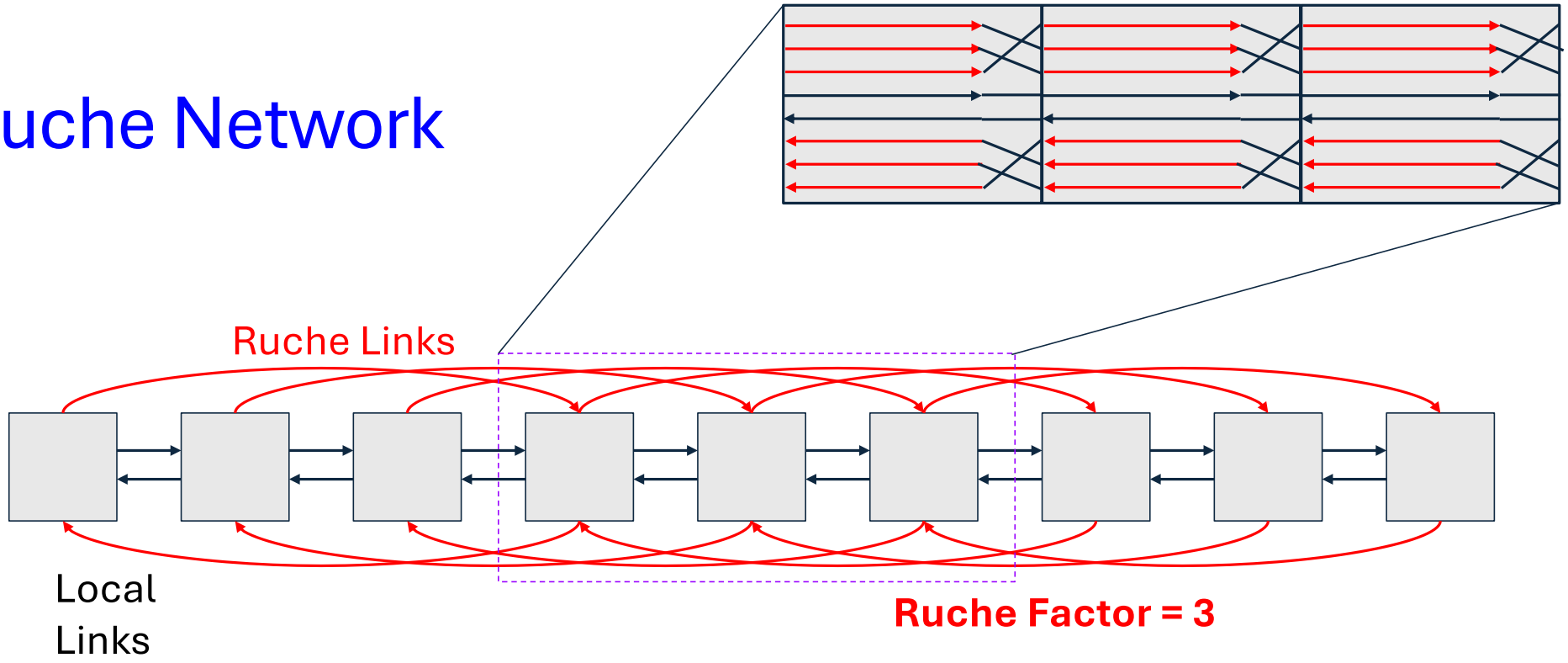
Ruche Factor = # of hops that ruche links extend

Ruche Network



bisection bandwidth: 1x → 4x
network diameter: 8 → 4

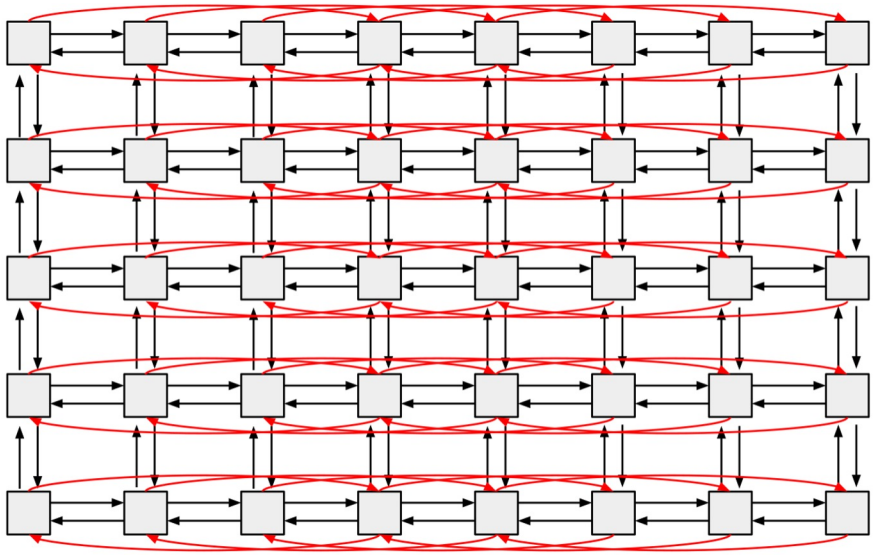
Ruche Network



By ***bit interleaving*** the Ruche Links, and shifting the wires inside each tile, each tile in the Ruche network has 100% identical logic and wiring, so the tiles can be replicated and placed directly abutting each other. This addresses the *tileability* requirement.

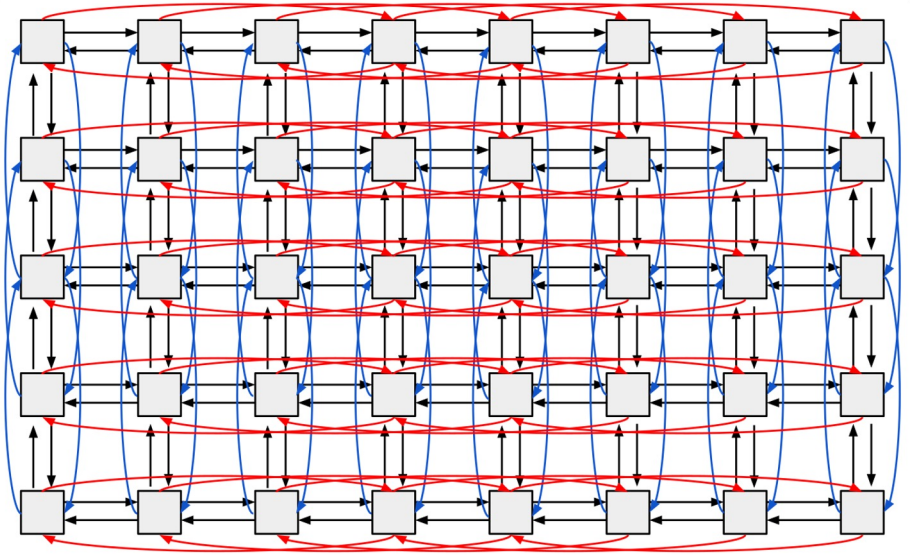
Half Ruche vs Full Ruche

half ruche



Ruche Factor X = 3

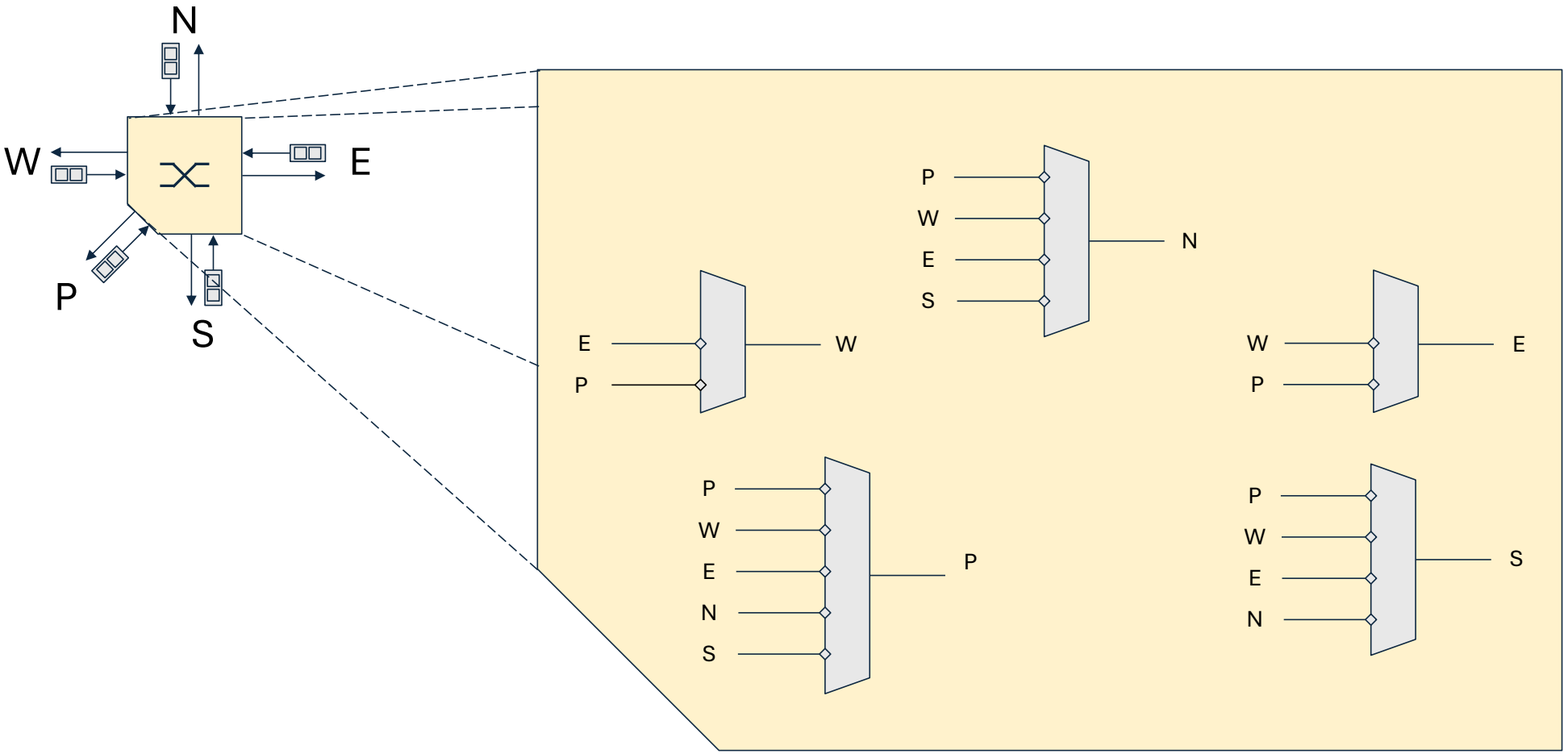
full ruche



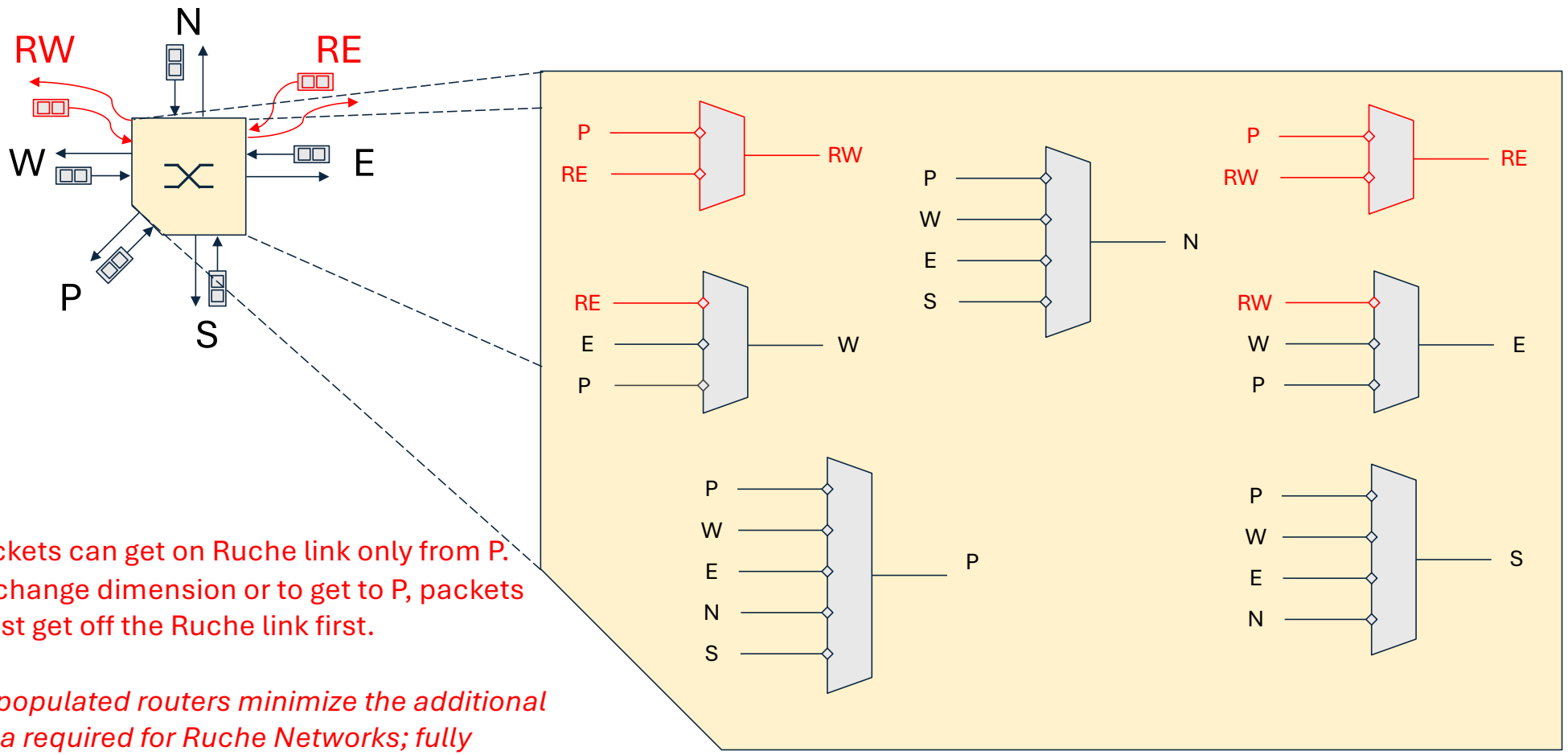
Ruche Factor X = 3
Ruche Factor Y = 2

Tweak to absorb raw wiring tracks in rectangular meshes, with rectangular tiles.

Dimensioned-Ordered 2D Mesh Router (X then Y)



Half Ruche *Depopulated* Router (X then Y)



Packets can get on Ruche link only from P. To change dimension or to get to P, packets must get off the Ruche link first.

Depopulated routers minimize the additional area required for Ruche Networks; fully populated routers are also okay!

Fully Ruched Populated/Depopulated

Spectrum of tradeoffs for router flexibility / PPA

Input Ports

Output Ports	RS	RN	RE	RW	S	N	E	W	P
RS		△	x	x		△	x	x	x
RN	△		x	x	△		x	x	x
RE				△					△
RW			△						△
S			x	x		o	o	o	o
N			x	x	o		o	o	o
E				△				o	o
W			△				o		o
P	△	△	x	x	o	o	o	o	o

Crossbar Connectivity Matrix

o = Original 2-D mesh connection
△ = Added by depopulated router
x = Added by fully-populated router

Depopulation reduces XBAR connectivity by 16.

Comprehensive NoC Evaluation

Ruche networks were previously proposed in NOCS 2020, and implemented in our 2048-core RISC-V manycore, HammerBlade [ISCA 2024] which had a Depopulated Half-Ruched NOC with Ruche Factor 3.

This paper's contribution is to evaluate *physically scalable* NoCs:

- Folded Torus
- 2-D Mesh
- 2X Multi-mesh
- Ruche NoCs (various configurations)

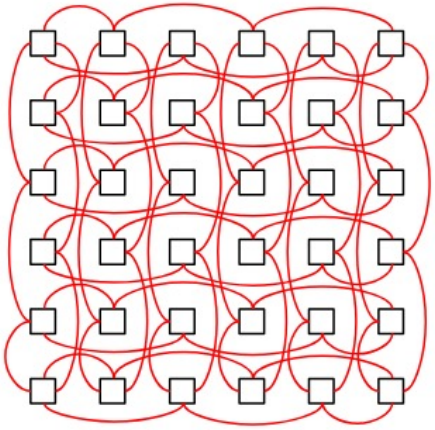
Evaluation Methods:

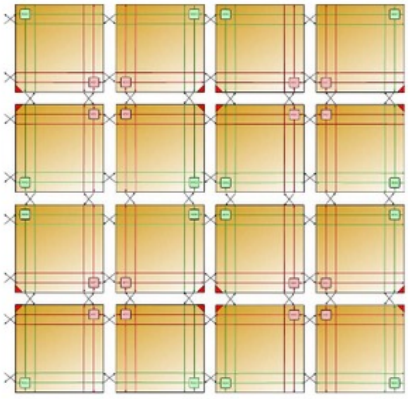
- Cycle-accurate RTL-level simulation.
- Cycle time, Area, and Energy based on 12-nm process.
- Benchmark evaluation using *cellular manycore* simulator (a.k.a HammerBlade).
- Full-system, execution-driven simulation (not trace-driven).

2-D Folded Torus: A Cousin/Alternative to Ruche 2?

- Bisection bandwidth doubled (like Ruche 2)
- Network diameter halved (like Ruche 2)
- Physically scalable and tileable (except edges?).
- Router radix remains the same as 2-D mesh.
  - *Seems* to require less router resources than Ruche.
  - *Except*, to avoid deadlock, 2-D torus requires *virtual channels and dateline logic*, unlike Ruche.

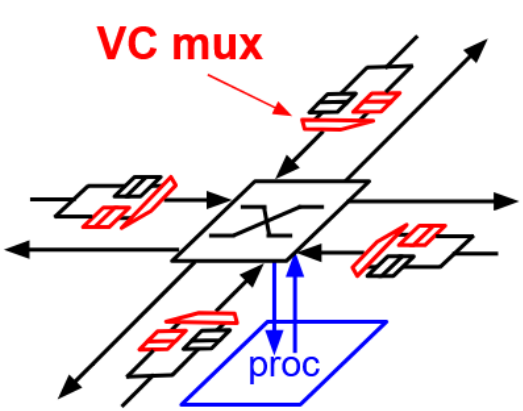
Requires more detailed comparison ..
More on this in the talk and paper.



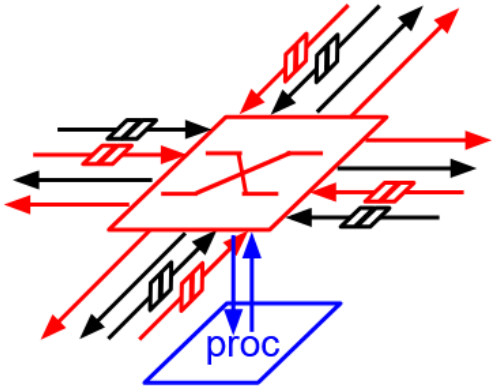


Tenstorrent, ISSCC 2022

2-D Torus's VC Router versus Ruche



(c) Virtual Channel Router



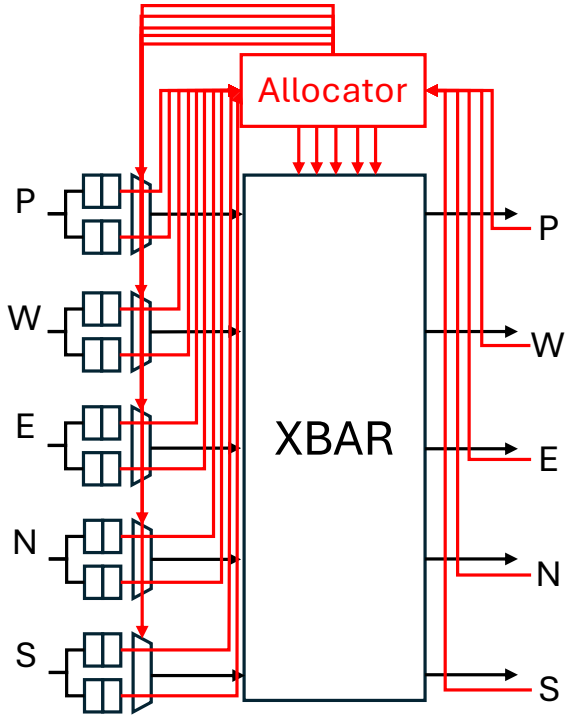
(b) Full Ruche

VC Router

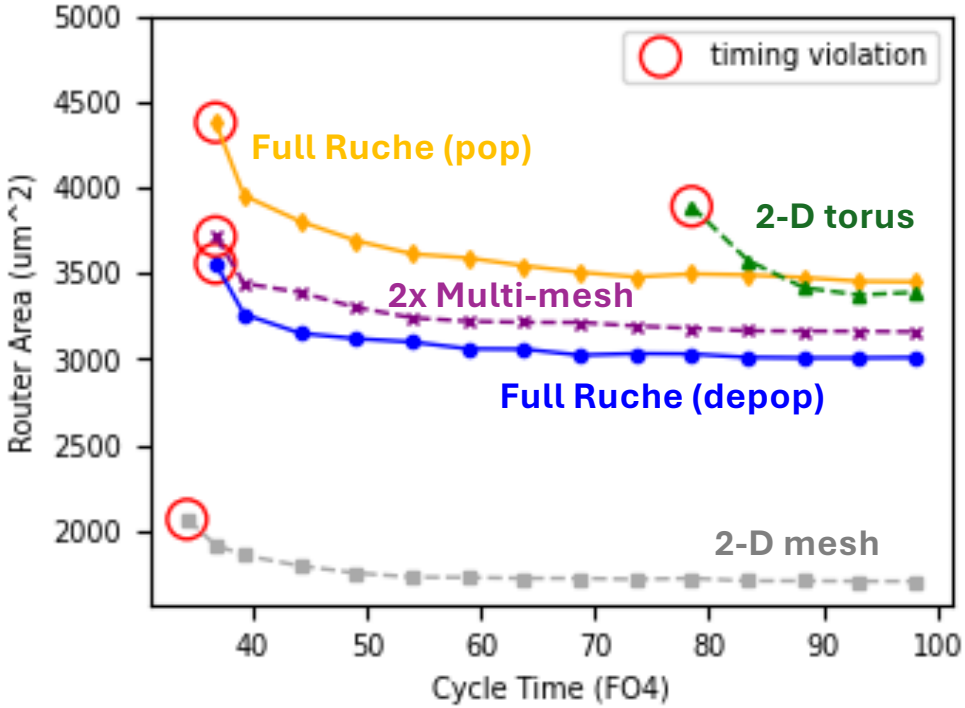
- Must Expend resources to 'VC mux' → Does not help with bandwidth…
- Similar resource utilization to Ruche networks, but half the router bandwidth.

VC Allocation also has longer critical paths

- VC Allocation must make a *centralized decision* that ensures fairness and maximal matching.
- VC Allocation logic generally have a longer critical path delay than arbitration logic.
- VC routers need to be pipelined to keep down cycle time.
- Speculative VC routers reduces hop latency but significantly increase area and power.



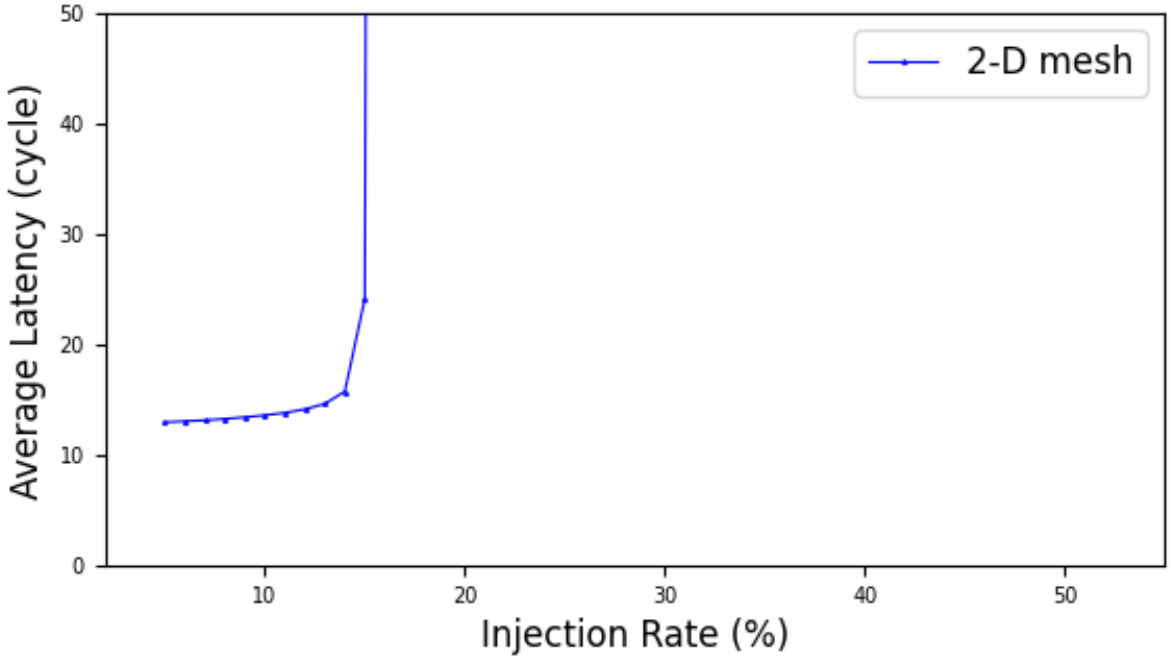
Lower Area and Cycle time



- Depopulation significantly reduces XBAR area.
- Depopulated Ruche has lower XBAR area than 2x Multi-mesh.

- 2-D torus router has higher area and cycle time.

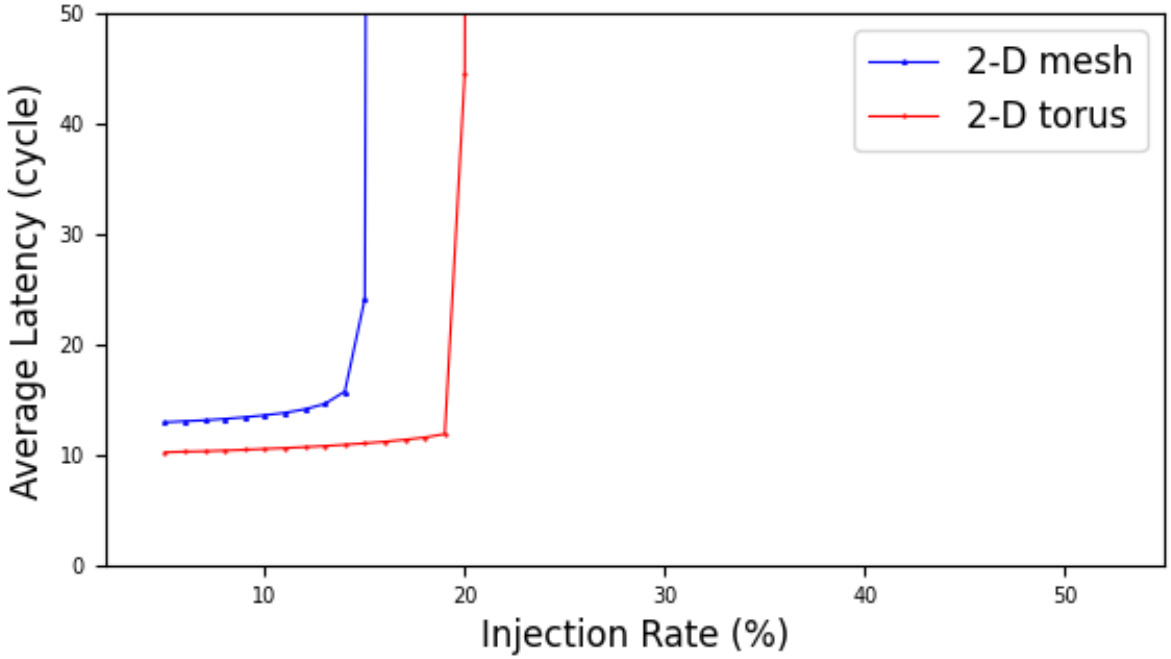
Synthetic Traffic Analysis



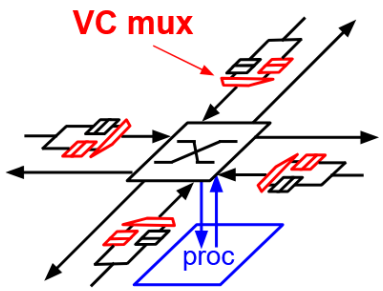
Uniform Random 16x16

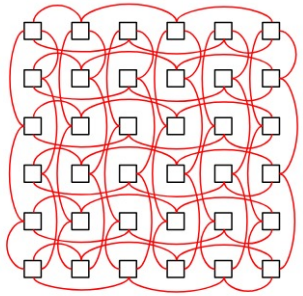
- 2-D mesh saturation throughput = 16%

2-D Torus struggles to deliver higher throughput.



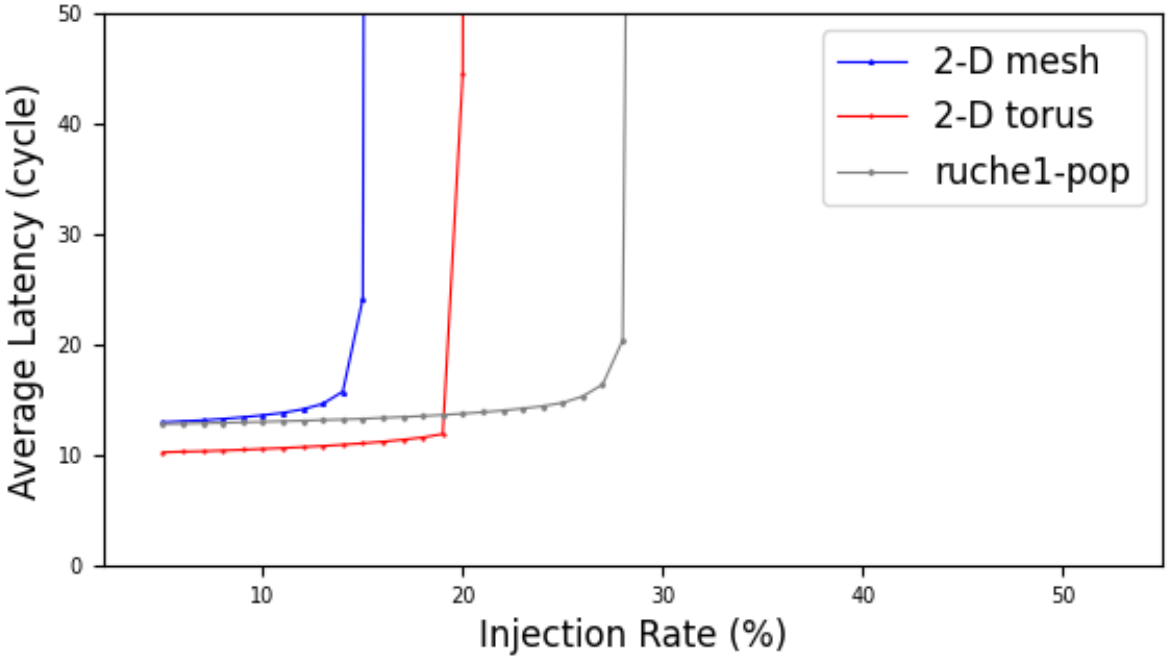
Uniform Random 16x16



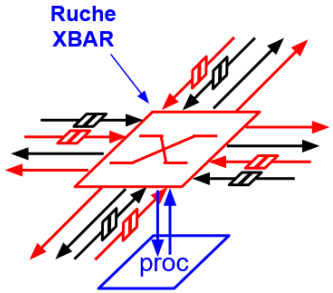


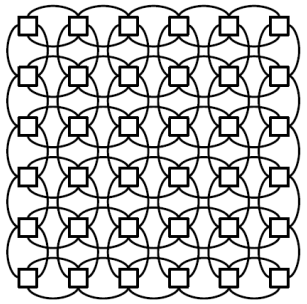
- 2-D torus saturation throughput = 20%
- Way less than what is expected by doubling the 'bisection bandwidth'...

Even Ruche without skip links has higher throughput.



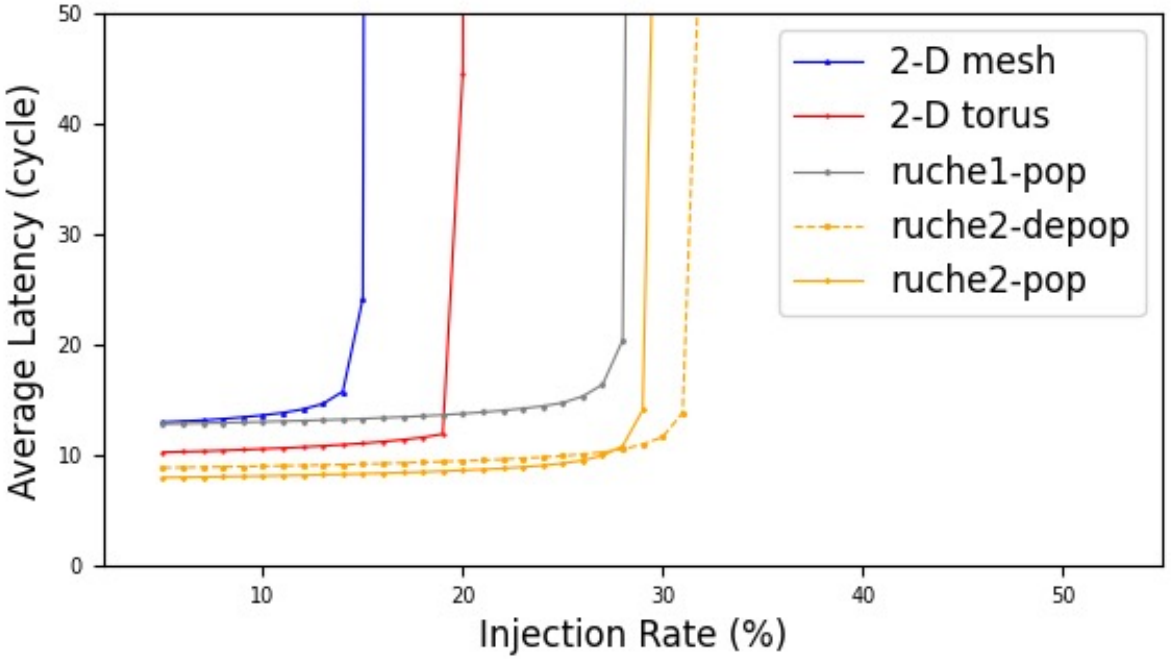
Uniform Random 16x16



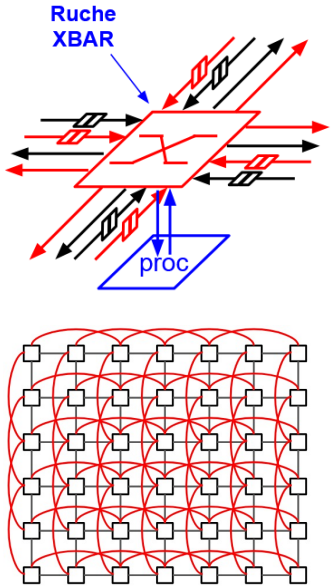


- Ruche-1 saturation throughput = 28%
- Ruche-1 without skip links is already better than 2-D torus.

Longer Ruche Links = ↓Latency, ↑Throughput

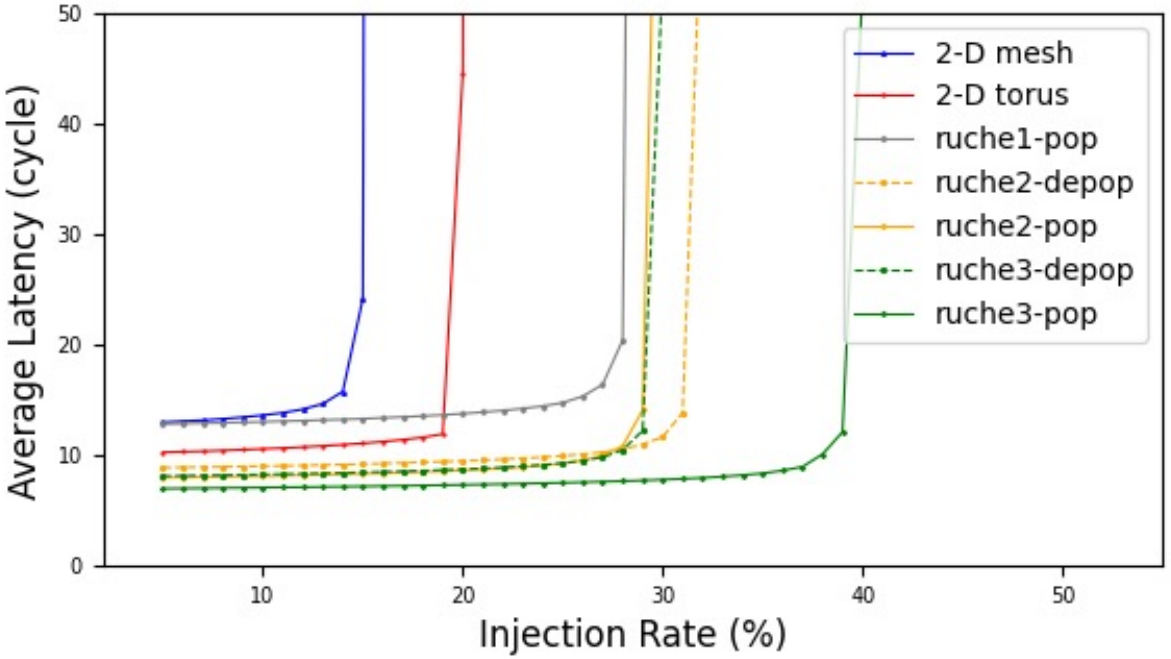


Uniform Random 16x16

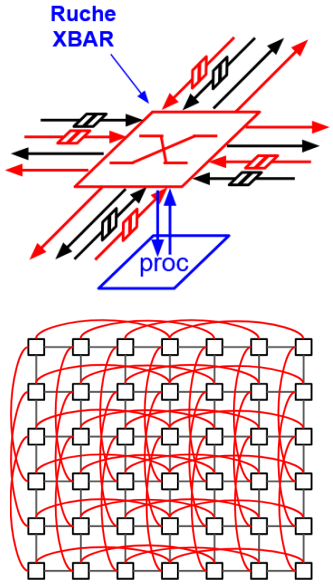


- Ruche-2 reduces zero-load latency and improves throughput more.

Longer Ruche Links = ↓Latency, ↑Throughput



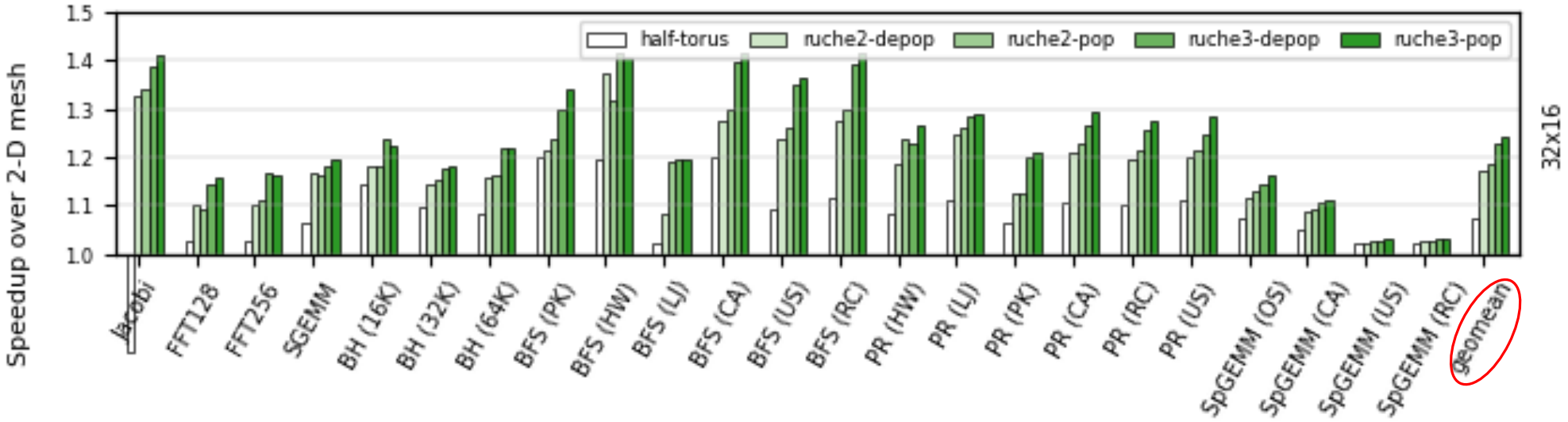
Uniform Random 16x16



- Ruche-3 lowers latency and improve throughput even more.
- Fully populated XBAR provides more routing flexibility.

Sounds all good in theory… but how about with real applications?

Speedup on Real Applications



- Most benefits over torus came from the simplest Ruche NoC: “Ruche-2 Depop”.
- Higher Ruche Factor and populated XBAR for incremental gains.

Geomean Summary

(vs mesh)	Torus	Ruche 2 depop	Ruche 2 pop	Ruche 3 depop	Ruche 3 pop
Speedup	1.08 X	1.17 X	1.19 X	1.23 X	**1.24 X**
Load Latency	1.11 X	1.27 X	1.30 X	1.35 X	**1.40 X**
Energy Efficiency	**0.91 X**	1.18 X	1.18 X	1.20 X	**1.22 X**
Area Increase	1.071 X	**1.058 X**	1.085 X	1.063 X	1.090 X
Area-normalized Speedup	**1.01 X**	1.11 X	1.10 X	**1.16 X**	1.14 X

Main Insights

- Ruching is an easy tool for converting spare wiring tracks into higher bandwidth, reduced network diameter and lower latency.
- Versus fatter channels or concentration, requires fewer additional changes to your core implementation, and more uniform conversion of chip resources to network performance.
- Folded Torus has some similarities to Ruche Factor 2 but suffers because of the requirement of virtual channels for deadlock.
- Ruche Factor, Populated/Depopulated and full/half Ruched are design choices you can make for Ruche networks.

Thank you

This work was supported in part by ACE and CHIMES, two of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and by NSF PPoSS Award 2118628.