The 2nd HiPChips Conference, HPCA-2023
Montreal, Canada

Tsinghua University

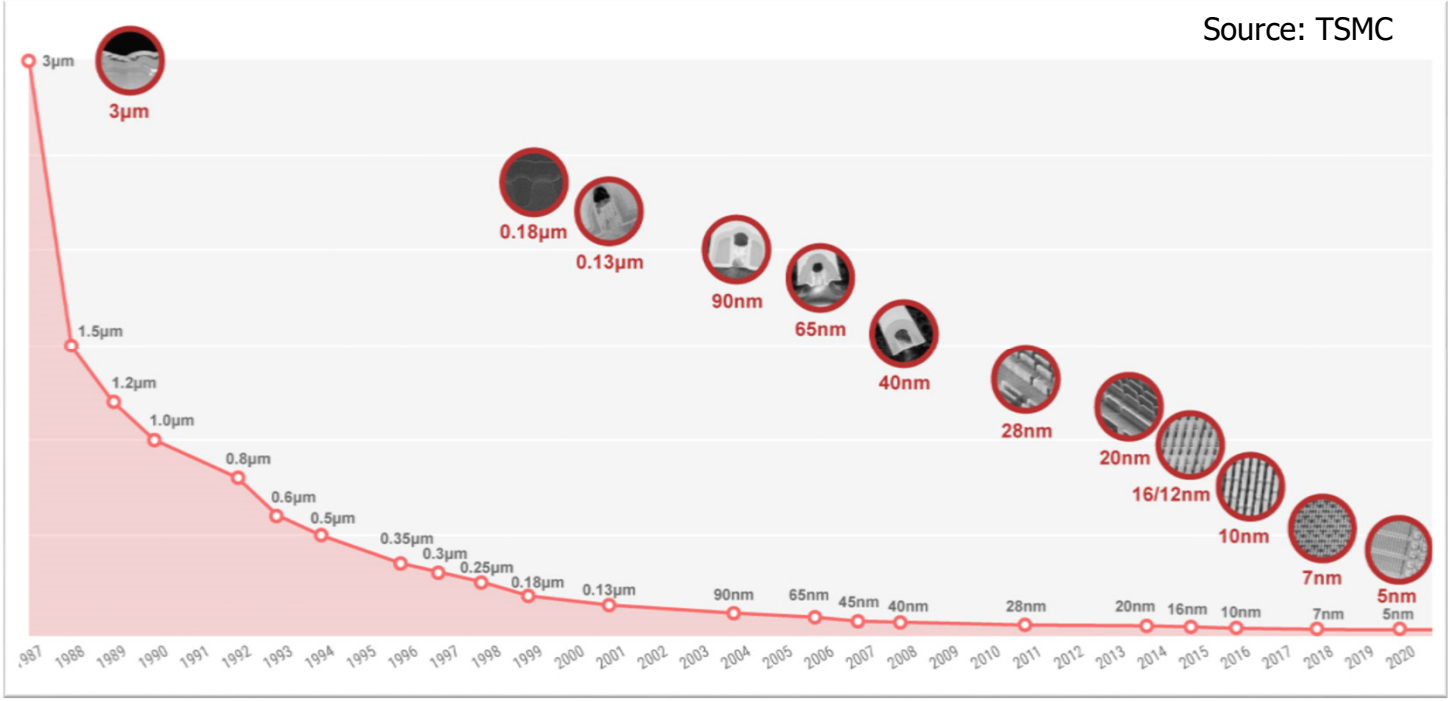
A Scalable Methodology for Designing Efficient Interconnection Network of Chiplets

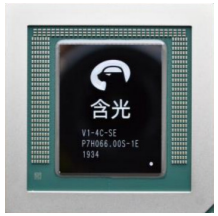
Yinxiao Feng*, Dong Xiang, Kaisheng Ma

Tsinghua University

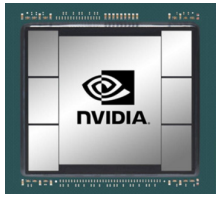
* Speaker

Background: Post-Moore Era





Alibaba (2020)
Hanguang-800
TSMC 12nm
709 mm²



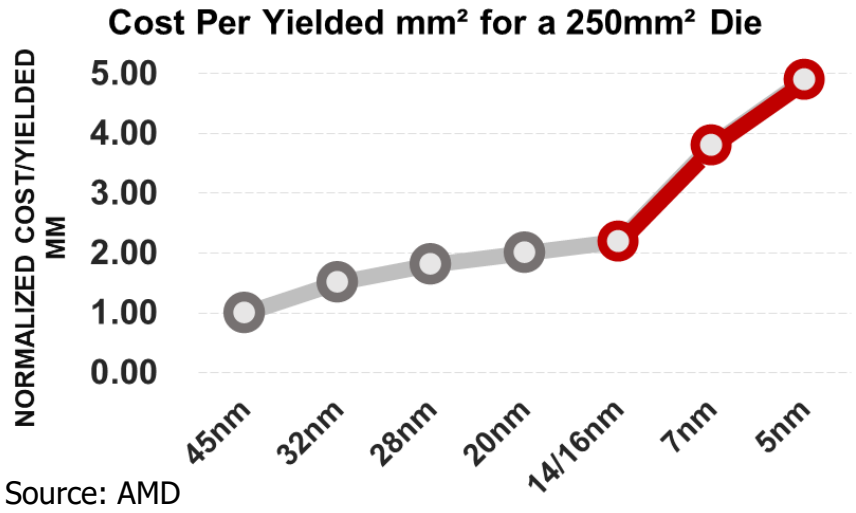
Nvidia (2020)
GA100
TSMC 7nm
826 mm²

Lithographic reticle area limit
26mm×33mm=858 mm²

Device size approaches physical limits, and technology development slows down

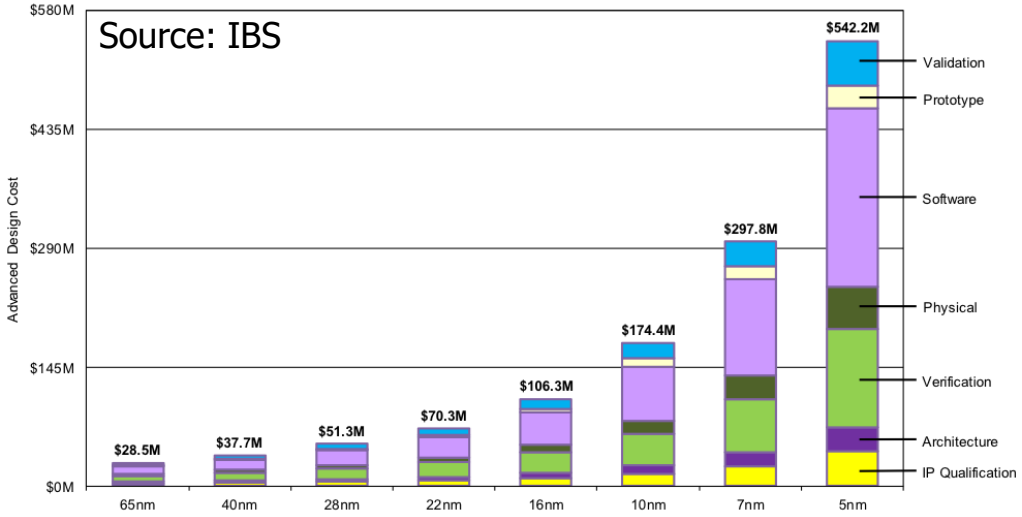
Transistors = Density × Area

Background: Cost Crisis



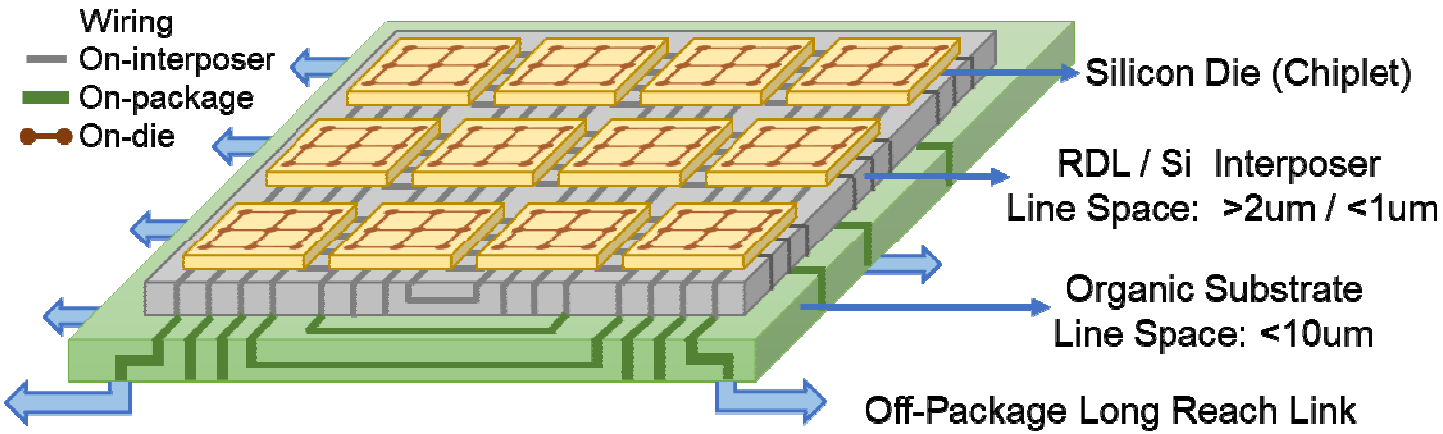
Source: AMD

The manufacturing (RE) cost increases rapidly



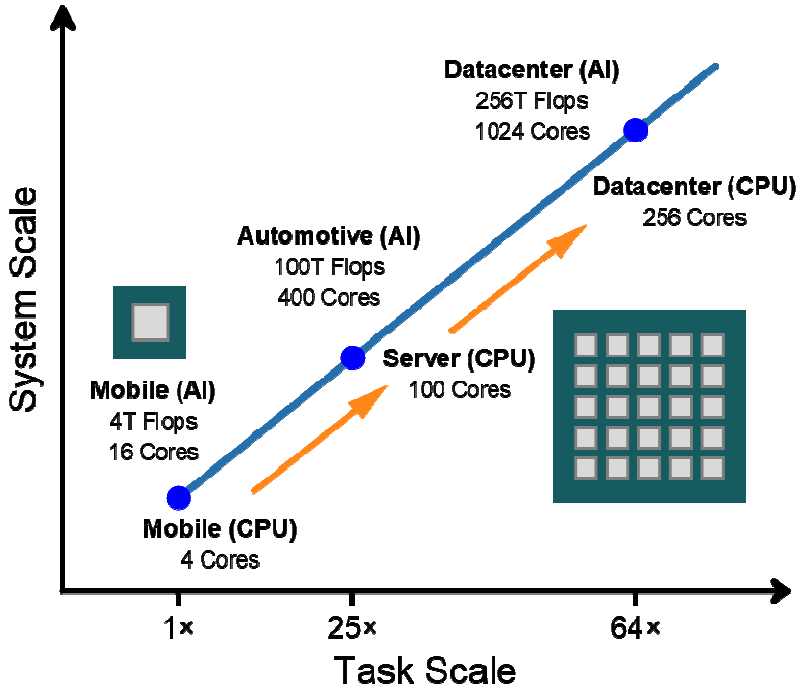
The non-recurring engineering (NRE) cost also increases rapidly

Advanced Packaging Technology



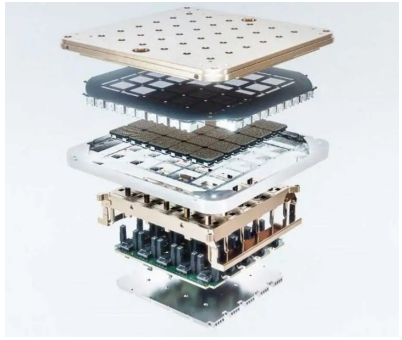
- Large size
- High density
- Abundant wiring

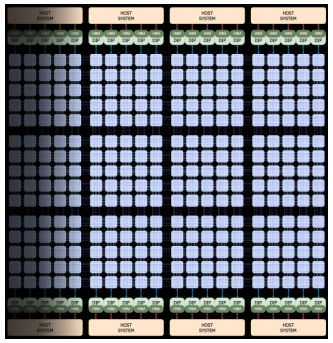
Chiplet Architecture



- Multiple chiplets
  - Break the area limit
  - Better yield
  - Chiplet reuse
  - **Scalable**

Scalable Multi-chiplet Systems



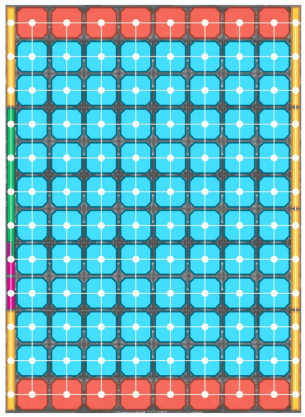


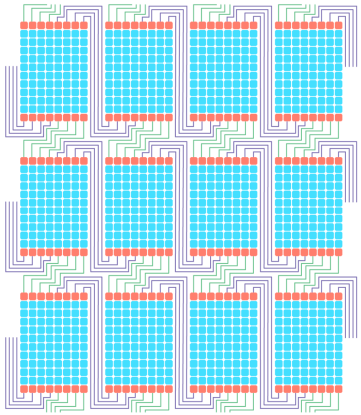
Tesla (2021)
Dojo
2D-mesh on chip
2D-mesh + Z-plane system

Flat Topology

Long Diameter

Limited Scalability





Tenstorrent (2021)
Wormhole
Folded torus on chip
2D-mesh system

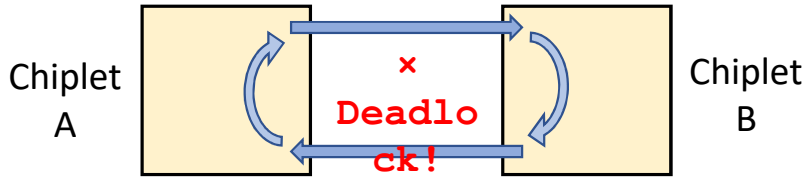
Challenges

- Scaling Challenge

2D-Mesh	nD-Mesh	Hypercube
$2(\sqrt{N}-1)$	$n(\sqrt[n]{N}-1)$	$\log_2 N$

  - 2D-Mesh topology (typical NoC)
    - Chiplet number = 64
    - Diameter = **14**

- Routing Challenge
  - Cross chiplet deadlock



  - Adaptivity (fault tolerance)

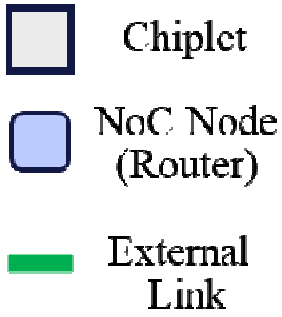
Poor network performance and not fully exploiting chiplet architecture!

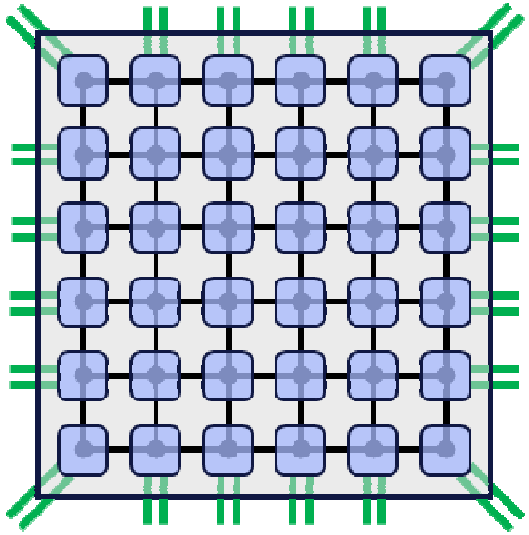
Motivations

Tsinghua University

- Building multiple systems of different scales from **identical** chiplets
- Chiplet is based on **2D-mesh** (Typical NoC design)
- **Scalable** (variable topology)
- **Deadlock-free** adaptive routing algorithm
- **Efficient**

Network on Chiplet

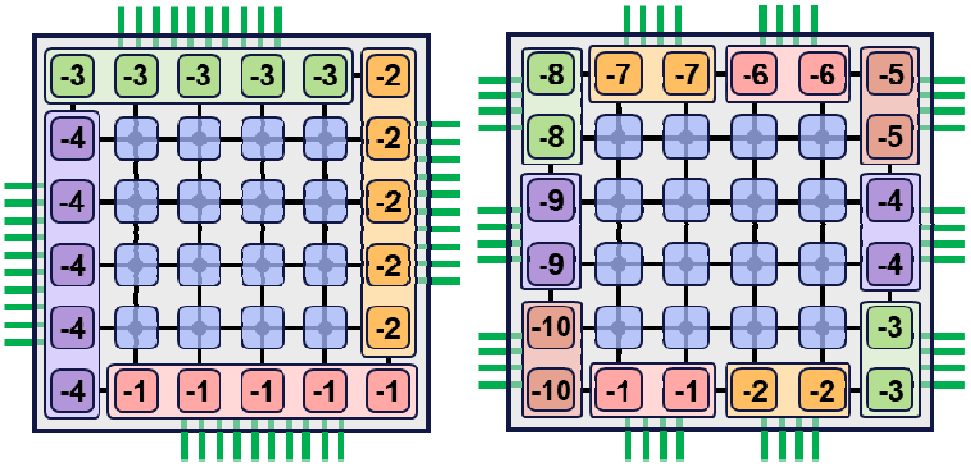




- 2D-mesh topology
- Typical virtual-channel-based router
- Virtual cut-through (VCT) switching
- Every edge node has an external link

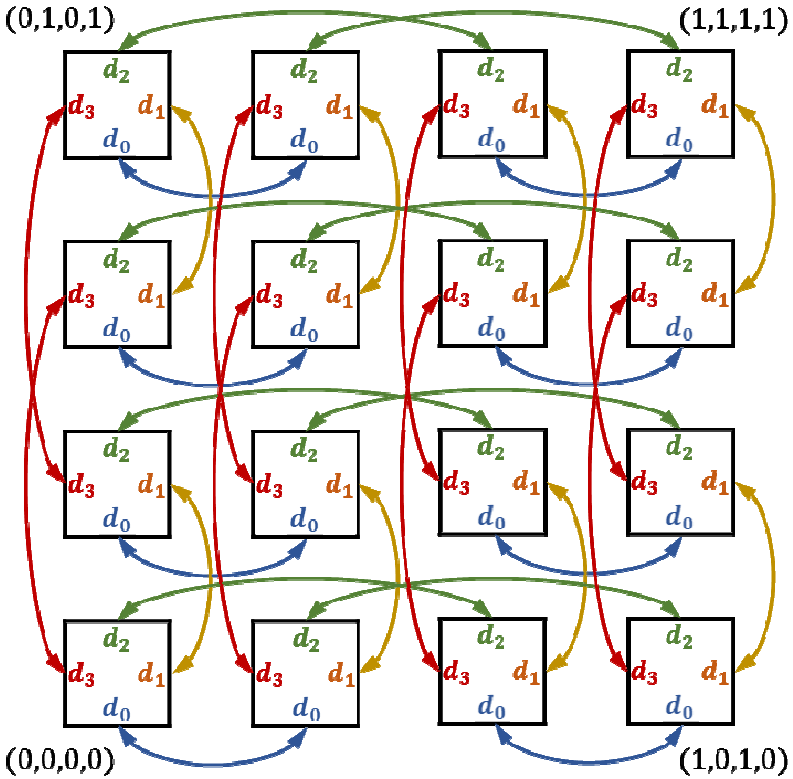
Minimal changes to the traditional NoC architecture!

Interface Grouping



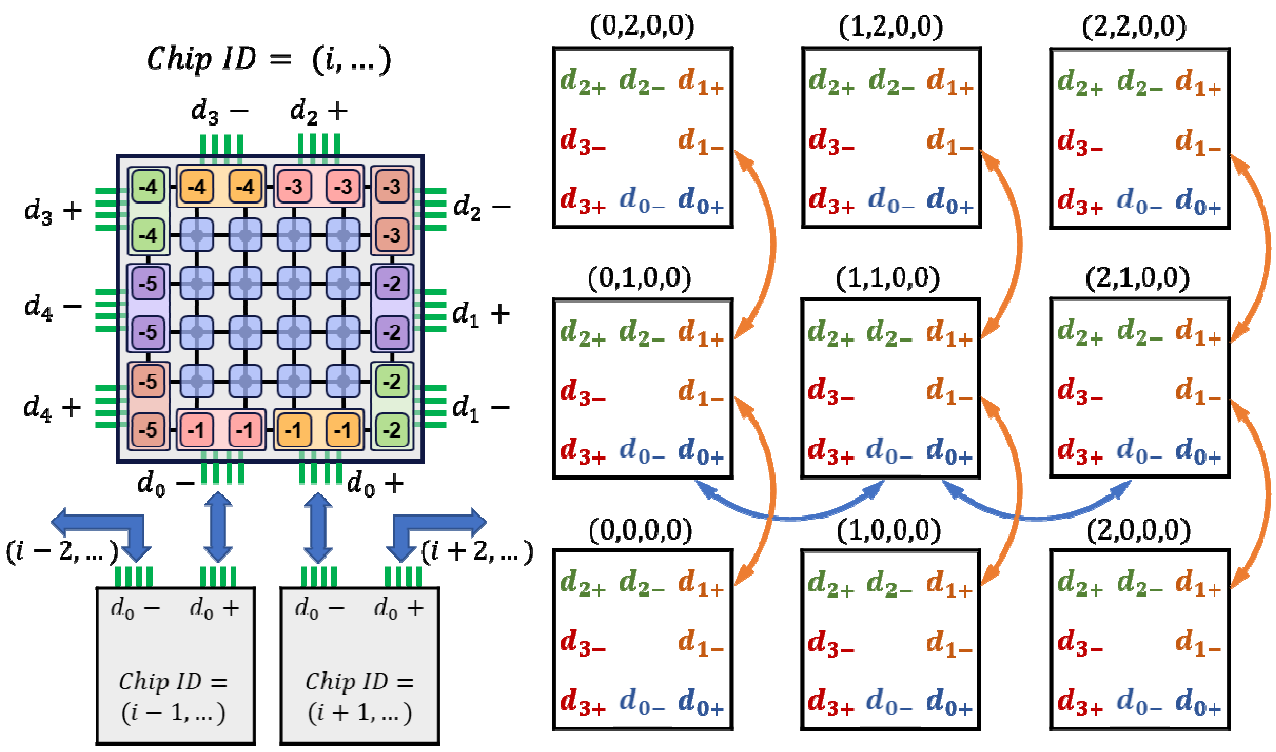
- 2D-mesh: radix - 4
- Hypercube: radix - $\log_2 N$
- **Software-defined** interface grouping
  - Variable chiplet radix
- Interleaving
  - Higher bandwidth (muti-channel)
  - Higher utilization

Interconnection: Hypercube

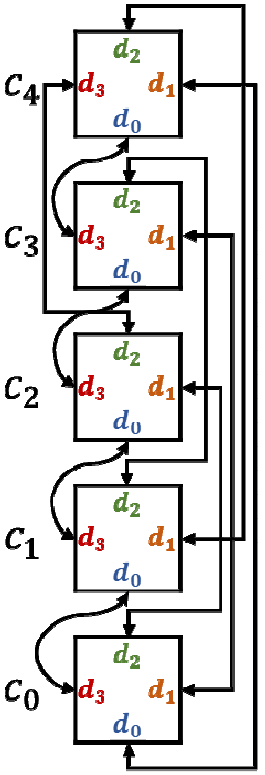


- Connecting the same dimensions
- Chiplet number $\mathrm{N} = 2^n$
  - Clustering into n interface group
  - Diameter: n
- Not wasting port

Interconnection: nD-mesh & dragonfly

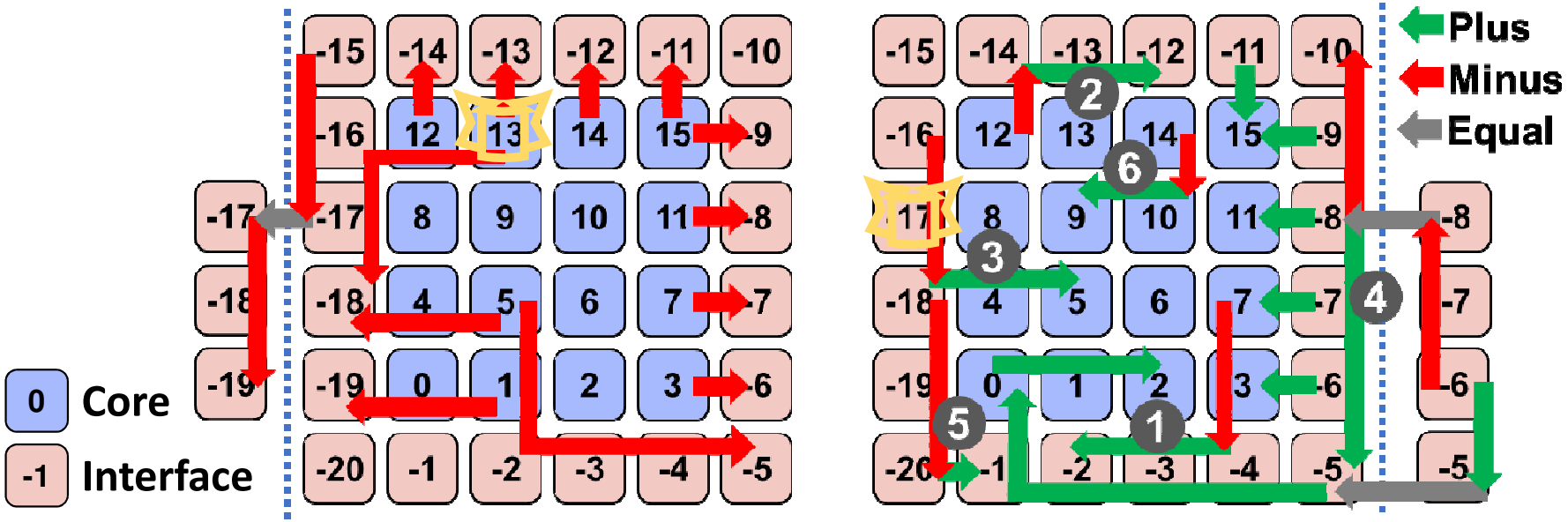


nD-mesh



Dragonfly

Minus First Routing



The general idea of designing the MFR algorithm is to find a "minus-first" path for any node pair based on the labels.

Routing Details

Algorithm 2 MFR(C,P) AMONG CHIPLETS

Input:
current node C,
packet P,

1: $Offset$ = different dimensions of chiplet coordinates.
2: assert($Offset$! = $\emptyset$)
3: B = IF node of dimension $d_B \in Offset$ that need to transfer first
4: **if** C == B **then**
5: transfer to adjacent chiplet through outward channel
6: **else**
7: transfer to B based on MFR(C,P) WITHIN CHIPLET
8: **end if**

- **Key:** Within a chiplet
  - Each core can access any interface through a minus path
  - Each interface can access any core through a plus path

Algorithm 3 MFR(C,P) WITHIN CHIPLET

Input:
current node C,
packet P,

Function: CORE_TO_CORE(C,P)
transfer based on NEGTIVE_FIRST_ROUTING(C,D).

Function: IF_TO_IF(C,P)
transfer along the IF ring, out of the chiplet through the external link if necessary; only one turn from minus to plus channels is allowed

Function: IF_TO_CORE(C,P)
transfer along the IF ring until
the label of adjacent core node $\leq$ destination[1], then transfer based on CORE_TO_CORE(C,P) by only plus channels.

Function: CORE_TO_IF(C,P)
transfer to an IF node by only minus channels, then transfer based on IF_TO_IF(C,P).

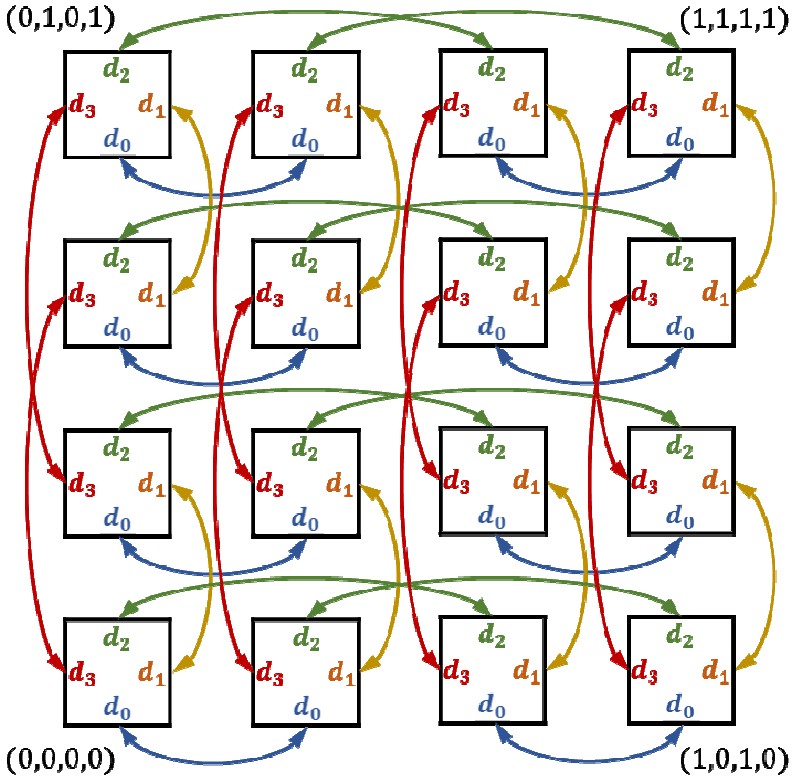
Case study: Hypercube

Algorithm 4 MFR(C,P) FOR HYPERCUBE

Input:

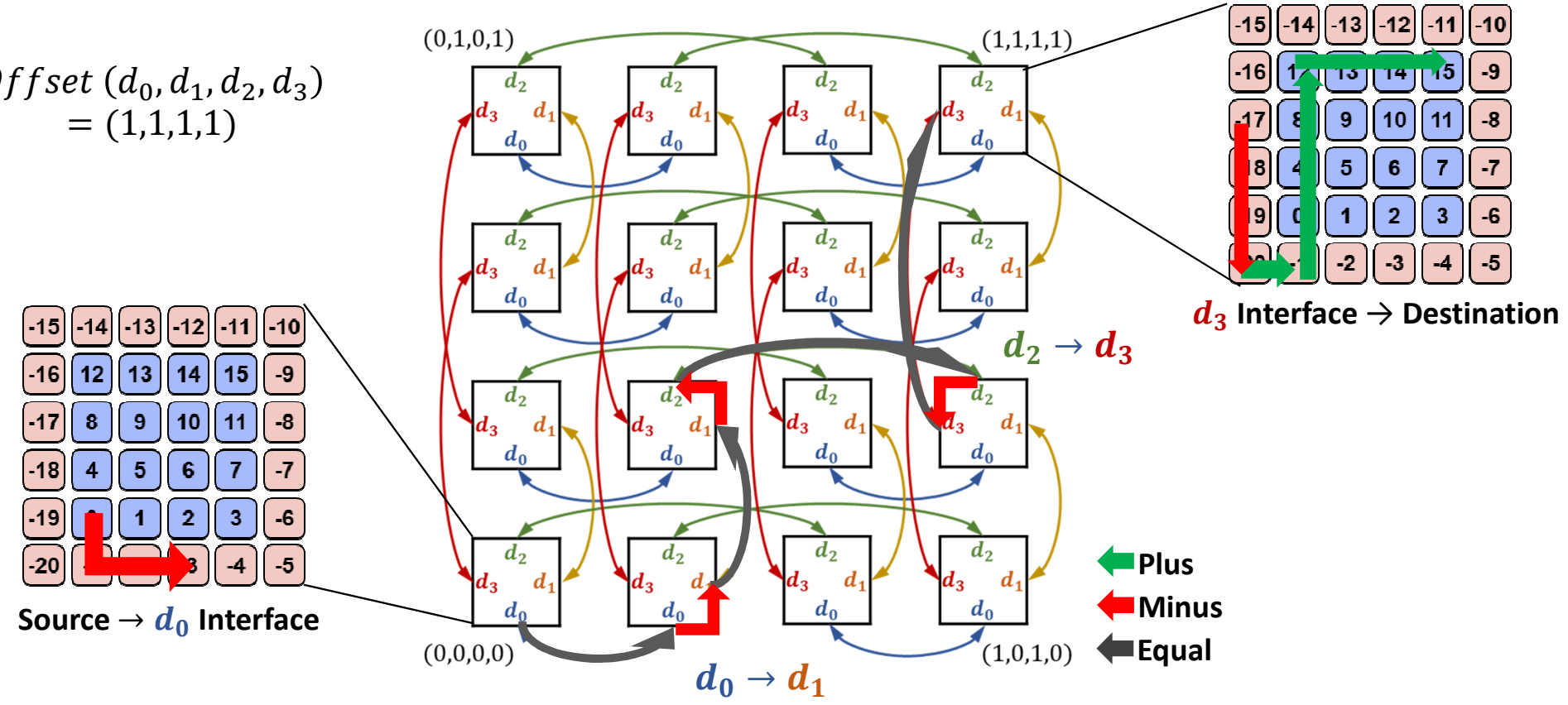
current node C,
packet P;

1: $Offset$ = different dimensions of chip coordinates. each dimension is associated with an IF.
2: **while** $Offset\ != \emptyset$ **do**
3: transfer to IF node B ($d_B = \min\{Offset\}$ with largest label) based on IF_TO_IF(C,P) or CORE_TO_IF(C,P) by only minus channel
4: transfer to adjacent chiplet through outward channel
5: **end while**
6: **if** $P.destination$ is IF node **then**
7: transfer based on $IF\_to\_IF(C, P)$
8: **else if** $P.destination$ is core node **then**
9: transfer based on $IF\_to\_Core(C, P)$
10: **end if**



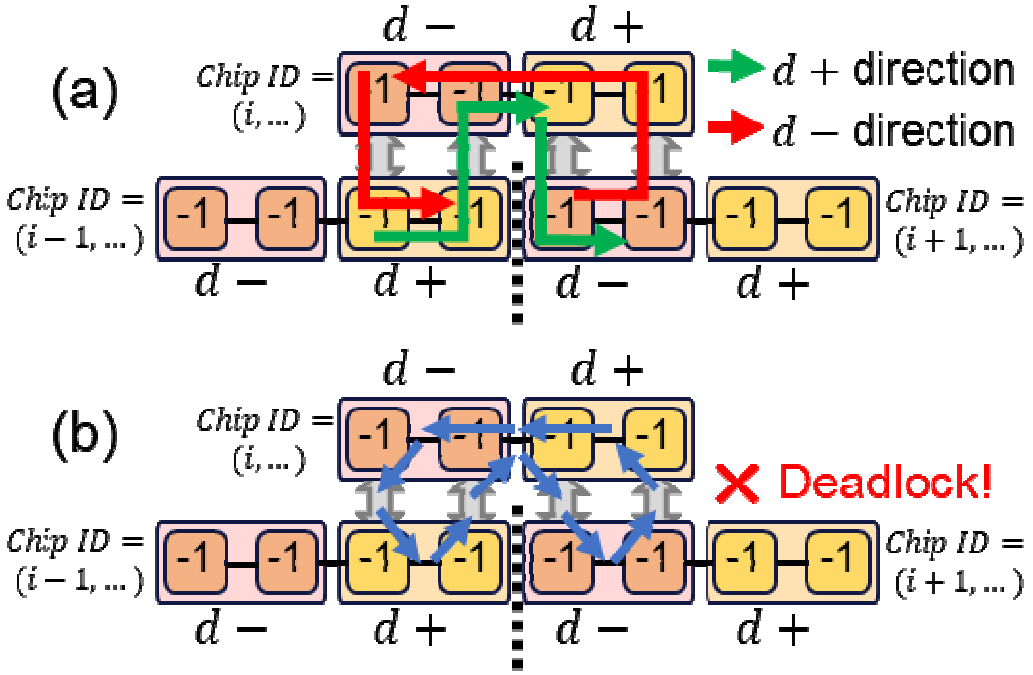
(0,0,0,0,0) → (1,1,1,1,15)

$Offset\ (d_0, d_1, d_2, d_3) = (1,1,1,1)$



Equal Channel Deadlock

- Difference to the traditional MFR: **equal channel**

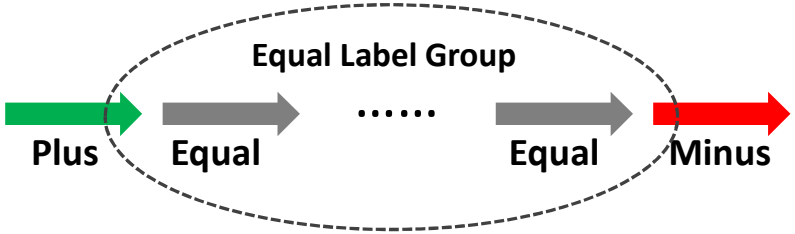


Deadlock-Free Proof

- Difference to the traditional MFR: **equal channel**

Theorem: *The MFR algorithm with **equal channels** is deadlock-free if the following conditions are met:*

1. *A packet that has passed through the plus channel is prohibited from turning to the minus channel; **(same with traditional MFR)***
2. *In the equal label groups, virtual channels are used to separate packets entering along the plus-channel (if any) from other packets;*
3. *There is no deadlock in the equal label groups.*

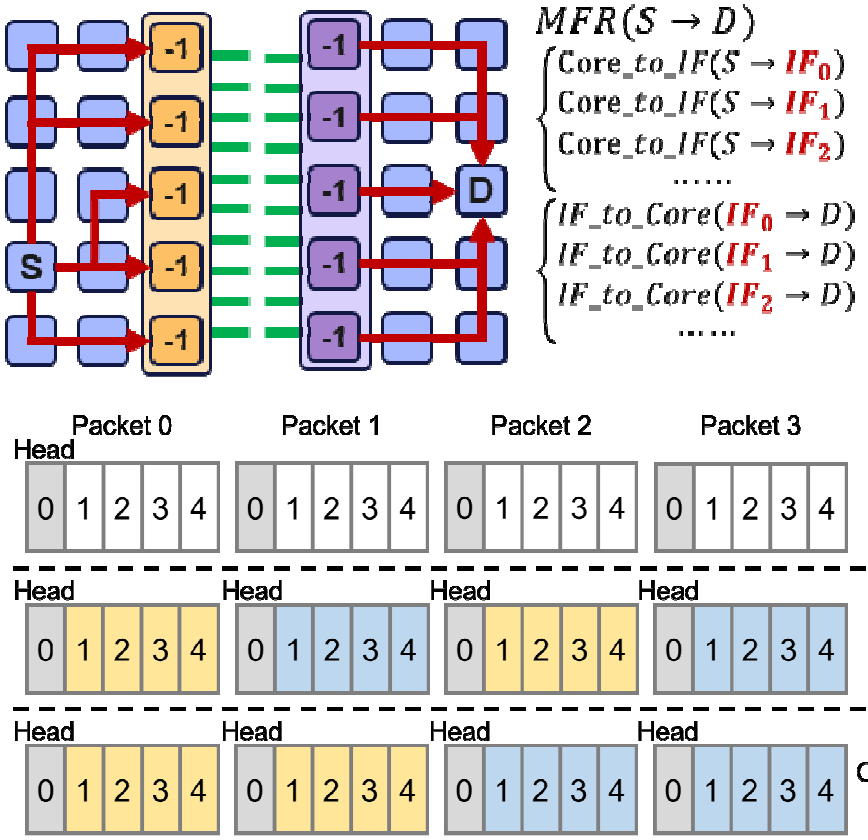


Discussion

- Hypercube
  - No equal to plus dependency
  - No deadlock in the equal label groups
- Deadlock-Free without extra virtual channel

- nD-mesh
  - No equal to plus dependency
  - $d+$ direction and $d-$ direction messages can lead to deadlocks in the equal label groups
- One extra virtual channel

Interleaving



- Fine-grained interleaving
  - High bandwidth utilization
  - Most even traffic
- Coarse-grained interleaving
  - Implement-friendly
  - Reduce power consumption under low network load

Evaluation

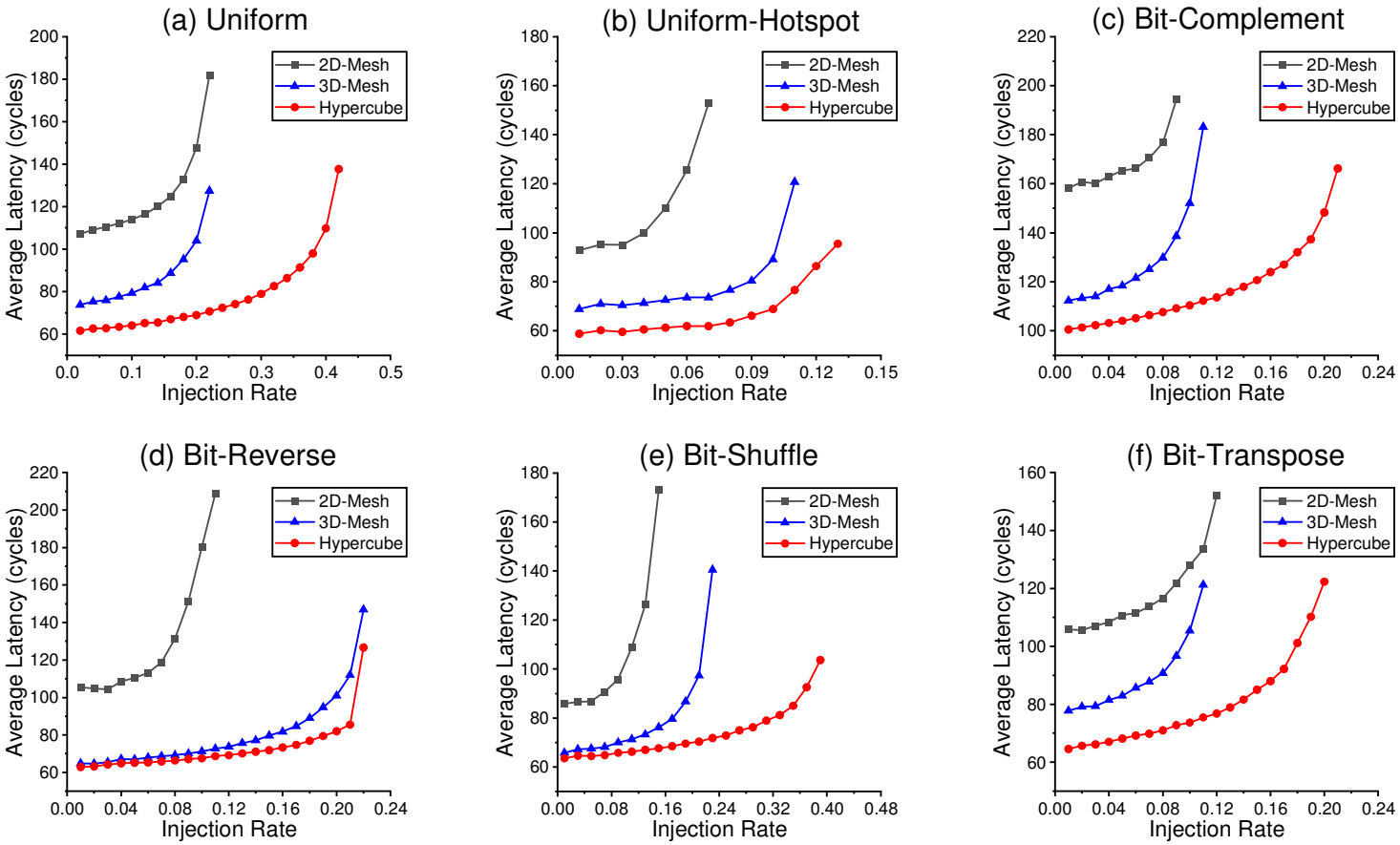
- Cycle-accurate C++ Simulator
  - Typical virtual channel router
  - Virtual cut-through switching
  - 4-stage pipeline
    - Routing
    - VC allocation
    - Switch allocation
    - Transmission
  - Preemptively-scheduled crossbar
  - Credit-based flow control
  - Individually-configured interface

TABLE II
PARAMETERS SETUP

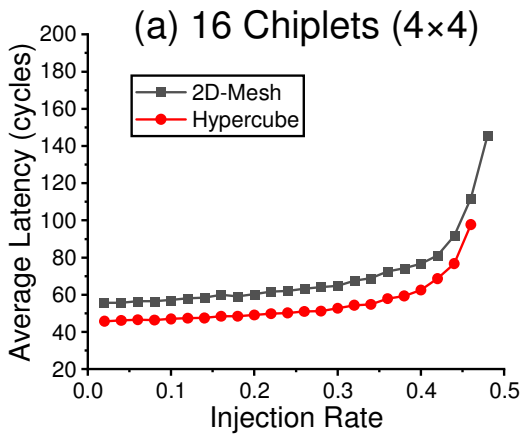
Parameter	Value
Flit Width	32 bits
Packet Length	32 flits
Input Buffer Size	1024 bits (32 flits) for internal buffers 2048 bits (64 flits) for interface buffers
Virtual Channel Number	2 channels/port
On-chip Link Bandwidth	128 bits/cycle (4 flits/cycle)
Off-chip Link Bandwidth	64 bits/cycle (2 flits/cycle)
Pipeline	4 stages; 1 cycle/stage
Intra-Chip Link Extra Delay	5 cycle
Simulation Time	6000 cycles (1000 for warming up)

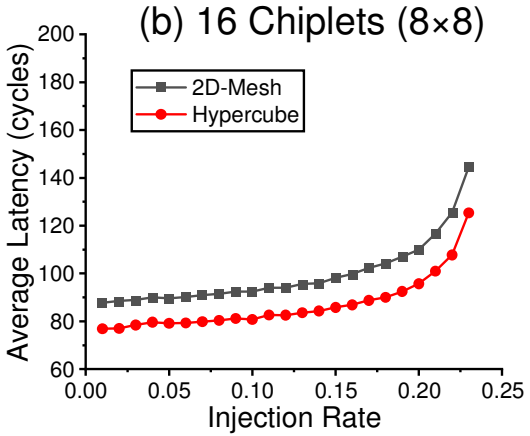
Evaluation on Traffic Patterns

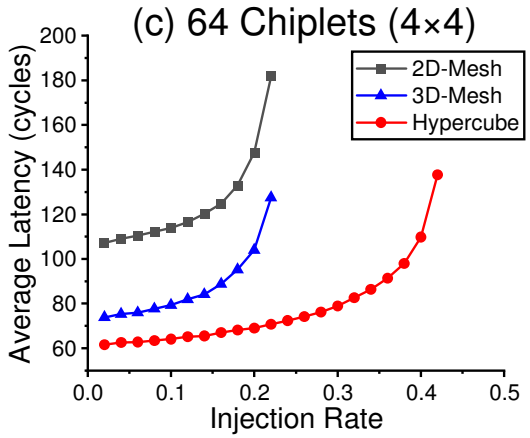
Tsinghua University

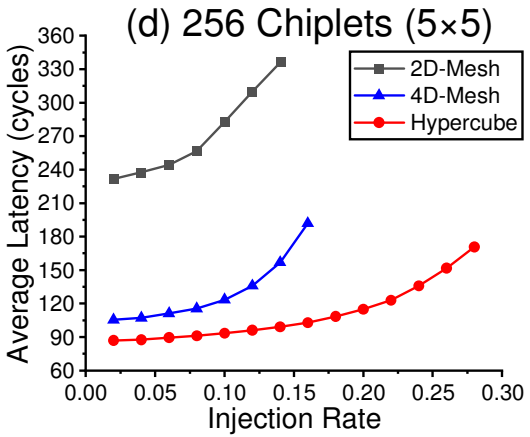


Exploration on System Scales



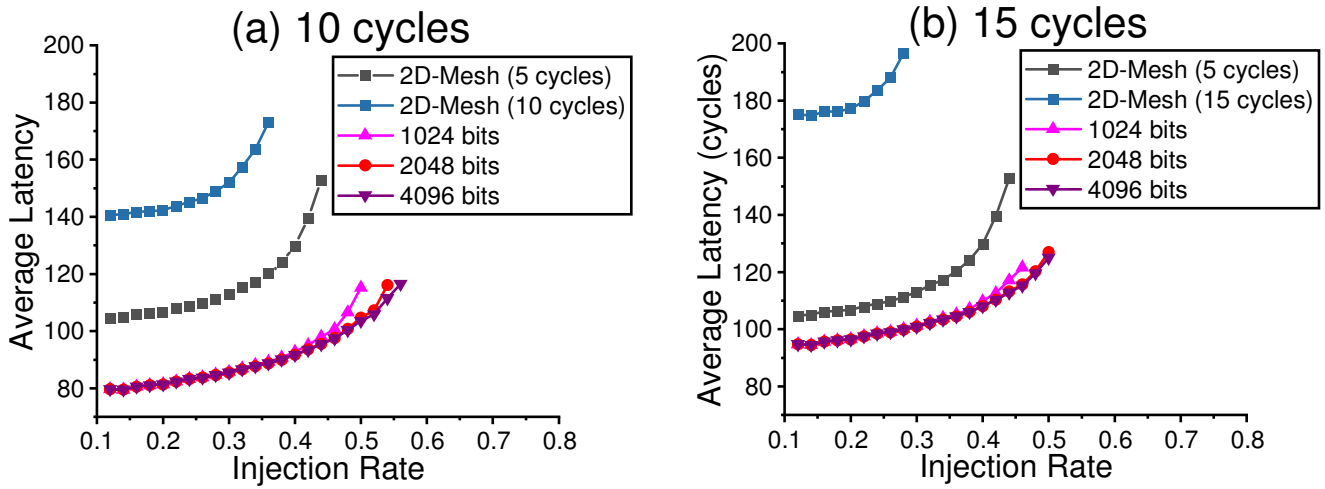






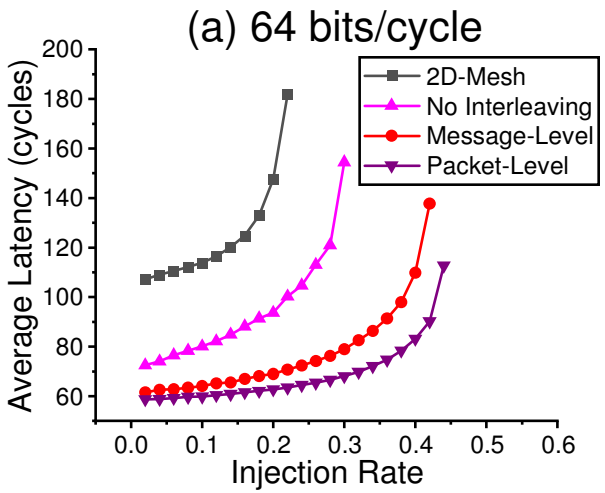
Exploration on Chiplet-to-Chiplet Link Configuration

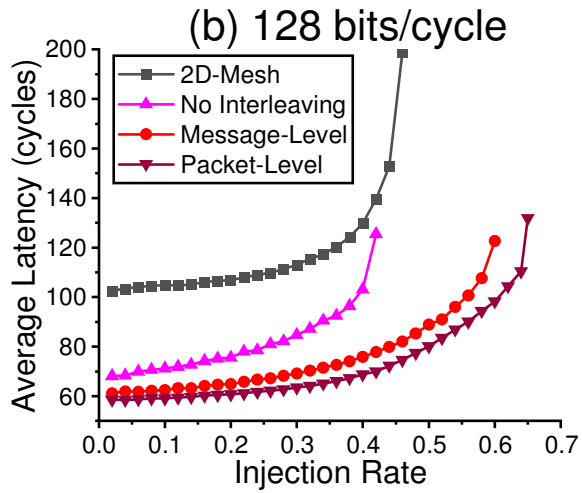
Tsinghua University



Exploration on Interleaving

Tsinghua University





Summary

- A methodology to design efficient and scalable chiplet interconnect networks
  - A software-defined interface grouping method
  - A specification for connecting 2D-mesh-NoC-based chiplets into high-radix networks.
  - A deadlock-free adaptive routing algorithm based on MFR
  - The network interleaving method
  - Evaluated on a chiplet-specific cycle-accurate C++ simulator
  - The evaluation and exploration results demonstrate our approach's high-performance and flexibility.

Thank you for listening