# Accelerator Architectures for Machine Learning (AAML)

## Lecture 10: Chiplet DNN Accelerator

Tsung Tai Yeh
Department of Computer Science
National Yang-Ming Chiao Tung University

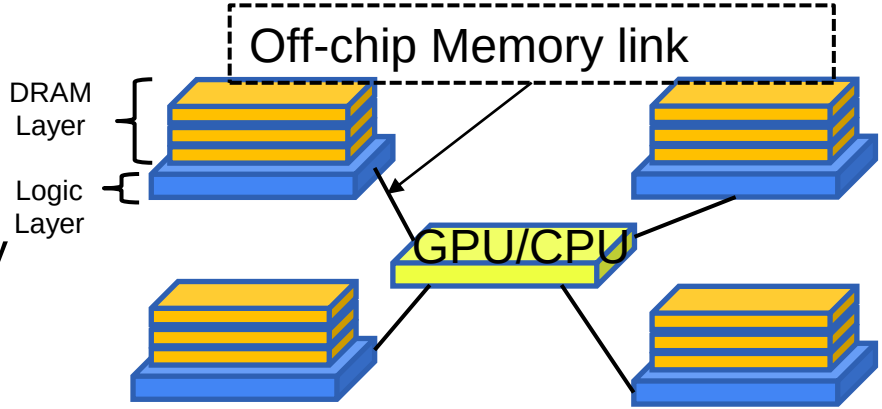
# Acknowledgements and Disclaimer

- Slides was developed in the reference with
  Joel Emer, Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, ISCA 2019 tutorial
  Efficient Processing of Deep Neural Network, Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, Morgan and Claypool Publisher, 2020
  Yakun Sophia Shao,  EE290-2: Hardware for Machine Learning, UC Berkeley, 2020
  CS231n Convolutional Neural Networks for Visual Recognition, Stanford University, 2020
  CS224W: Machine Learning with Graphs, Stanford University, 2021

# Outline

- Chiplet-based system
- Simba: Multi-Chip Module Architecture
- On-chip Network Topology

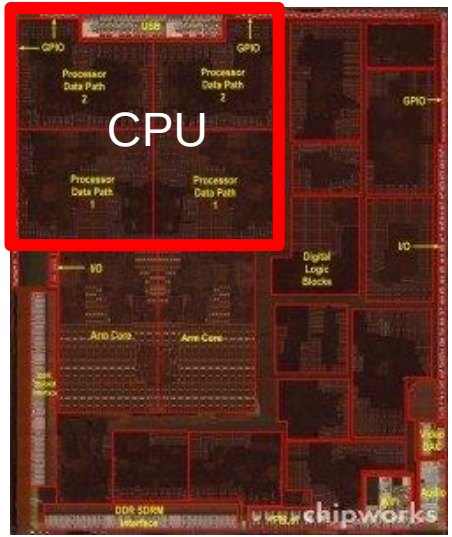
# Chiplet-Based System



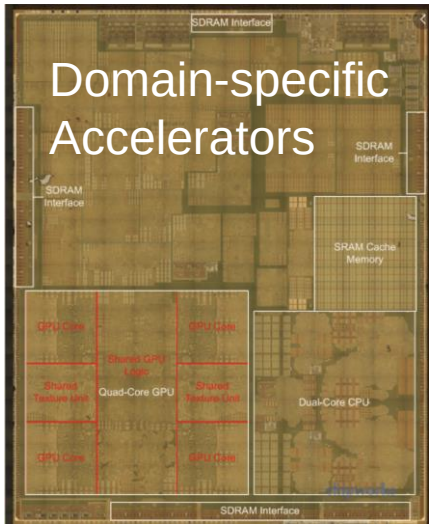

- **Motivation**
  - Difficult to pack more functionality on a single chip
  - High cost on the large chip
    - Verification cost is high
    - Manufacturing defects in densely packed logic can reduce the wafer yield
  - Rise of heterogeneity
  - Demand of big data
- **Chiplet-based system**
  - The integration of multiple discrete chips within the same package
  - Multi-chip module & silicon interposer

# Rise of Heterogeneity

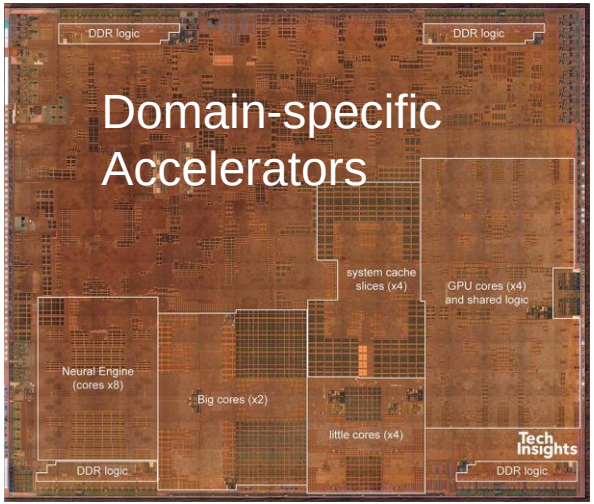
- More and more heterogeneous chips are placed in one die




**2010 Apple A4**
65 nm TSMC 53 mm$^2$
**4 accelerators**




**2014 Apple A8**
20 nm TSMC 89 mm$^2$
**28 accelerators**




**2019 Apple A12**
7 nm TSMC 83 mm$^2$
**42 accelerators**

https://edge.seas.harvard.edu/files/edge/files/alp.pdf

# Advantages of using Chiplets

- **Cost**
  - Smaller chips have lower manufacturing costs
  - Not all functionality in an SoC benefit from bleeding edge technology process
- **Flexibility**
  - Small simple chiplets can be more easily repurposed across market segments
- **Sustainability**
  - Not always chase the bleeding edge process technology (need high investment in manufacture processing)

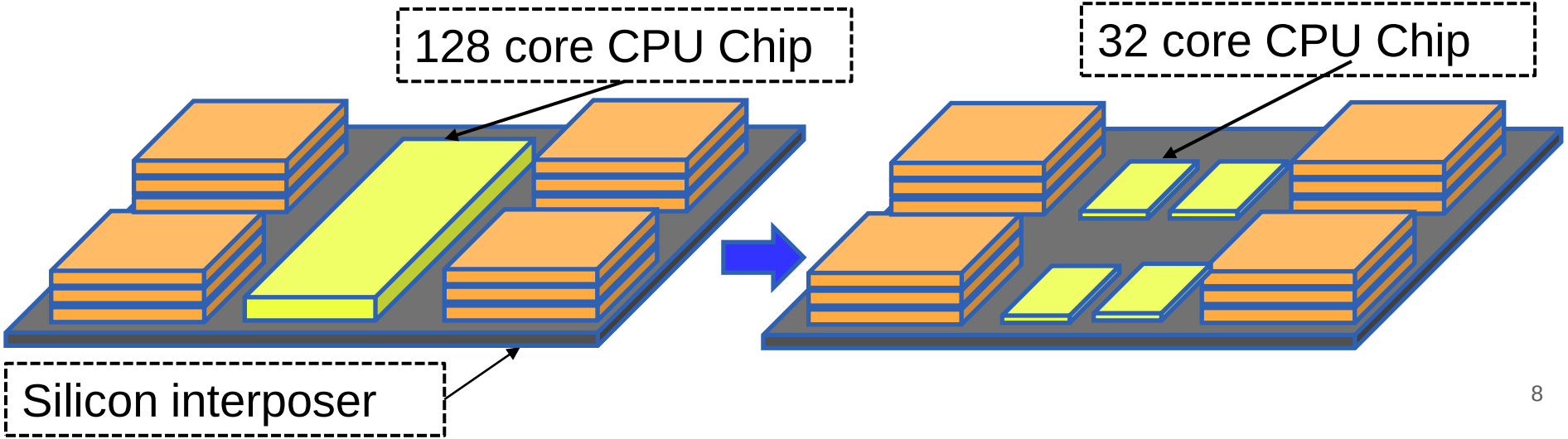
# Challenges of Chiplets

- **Communication**
  - Non-uniform communication latencies
  - Flexible inter-connection
  - Constrained bandwidth from interposer and package substrate
- **Open fabrics and interfaces**
  - Communicate across heterogeneous IPs
- **Heterogeneous integration**
  - Integrate pieces of custom silicon in different technology nodes

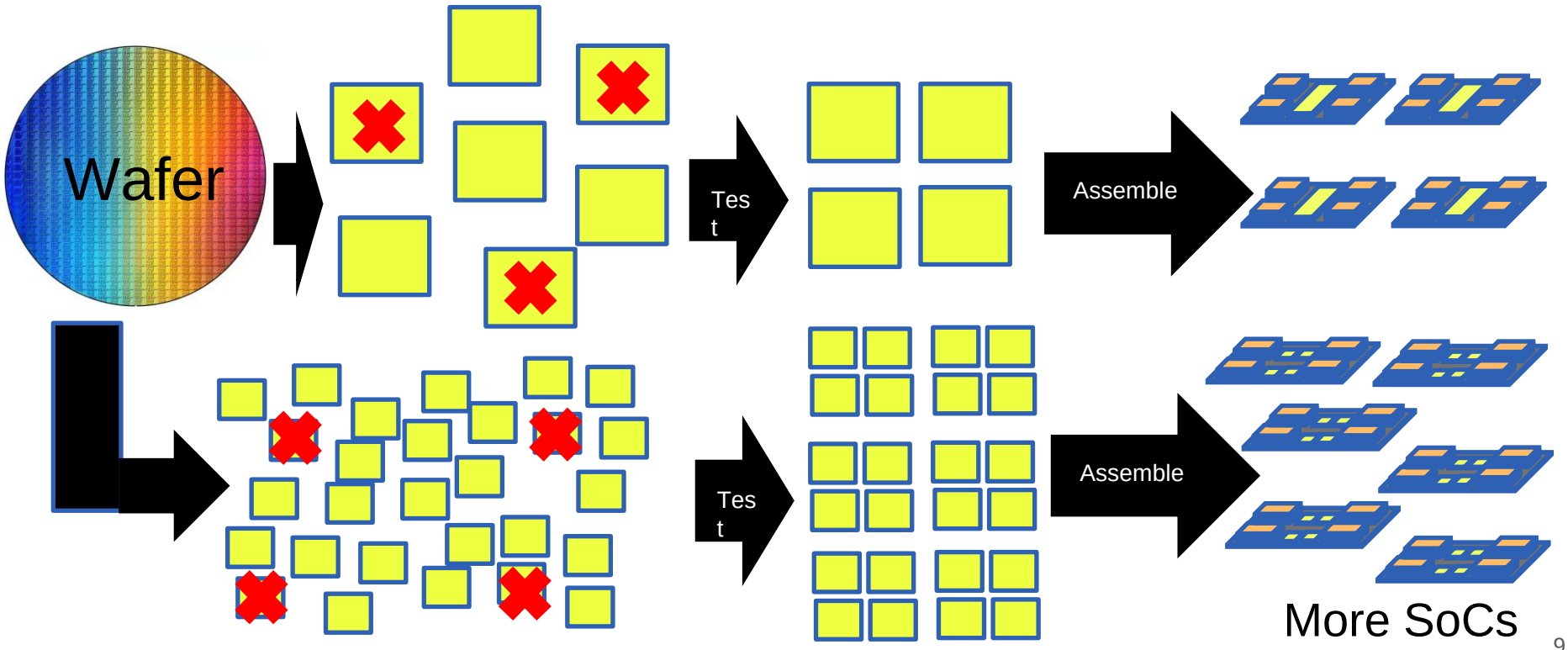
# Why Disintegrate Chips to Chiplets ?

- Big chips are expansive
- Break into several small pieces
  - Cheaper to manufacture

An **interposer** is like a silicon bridge that makes the connection to different sockets on the board

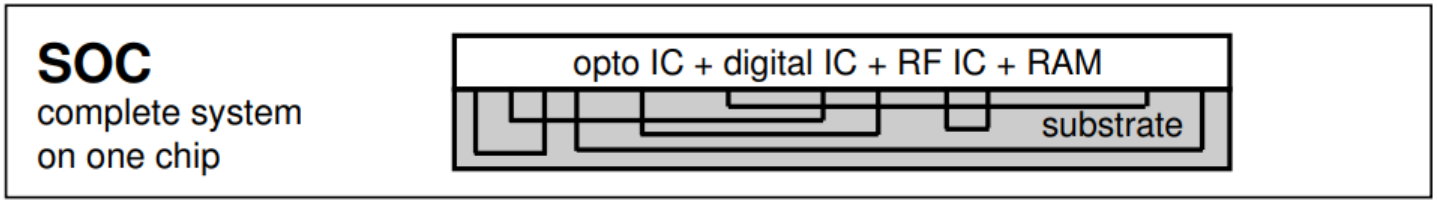


# Wafer Utilization in Fine-grain Disintegration



# System-on-Chip (SoC)

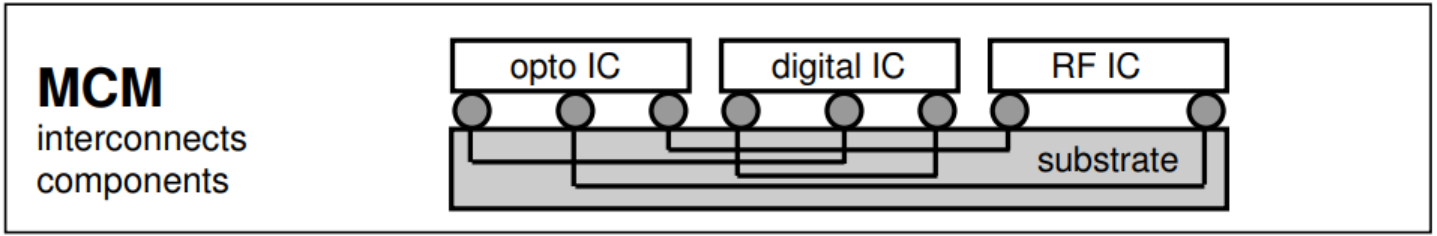
- **System-on-Chip (SoC)**
  - A whole system on a single chip
    - Processor, memory, I/Os, DSPs …
  - Optimized for power usage and minimized latency – short communication
  - What are the problem of a monolithic chip ?




Jacob Minz, 2005

# Chiplet/Multi-Chip-Module (MCM)

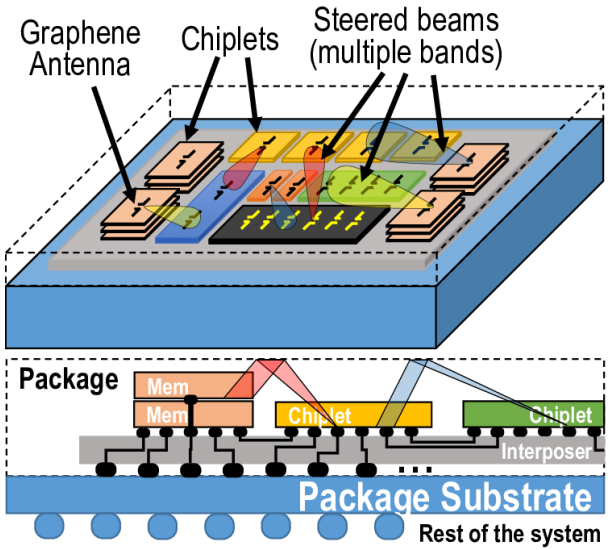
- **Chiplet (Multi-chip-module)**
  - Take heterogeneous components and connects them into one package
  - Leverage packing technology to integrate chips in a package
  - More functionality, simplified design, and higher yielded than SoC



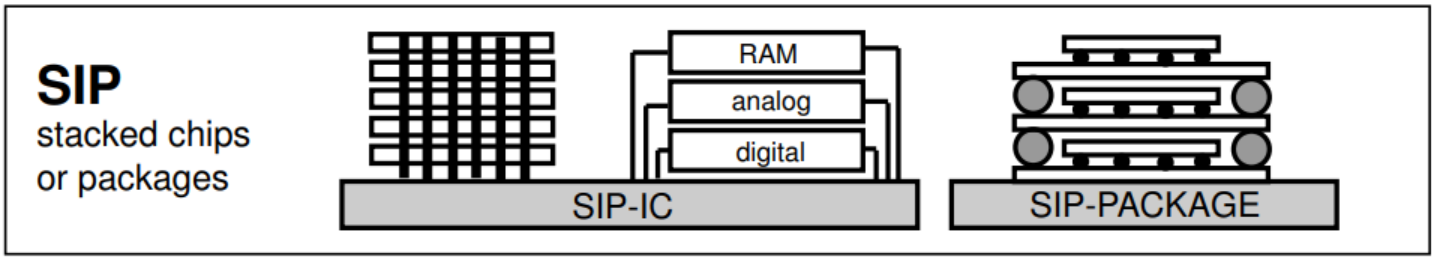

Jacob Minz, 2005

# System in Package - SiP

- **Stacking of ICs**
  - Wire bond -> 2.5D and 3D
  - Group of ICs mounted on a common substrate
  - Heterogeneous integration – More than Moore
  - Uses interposer and TSV technology
- **Advantages**
  - Simple design and verification, low time to market, increase yield



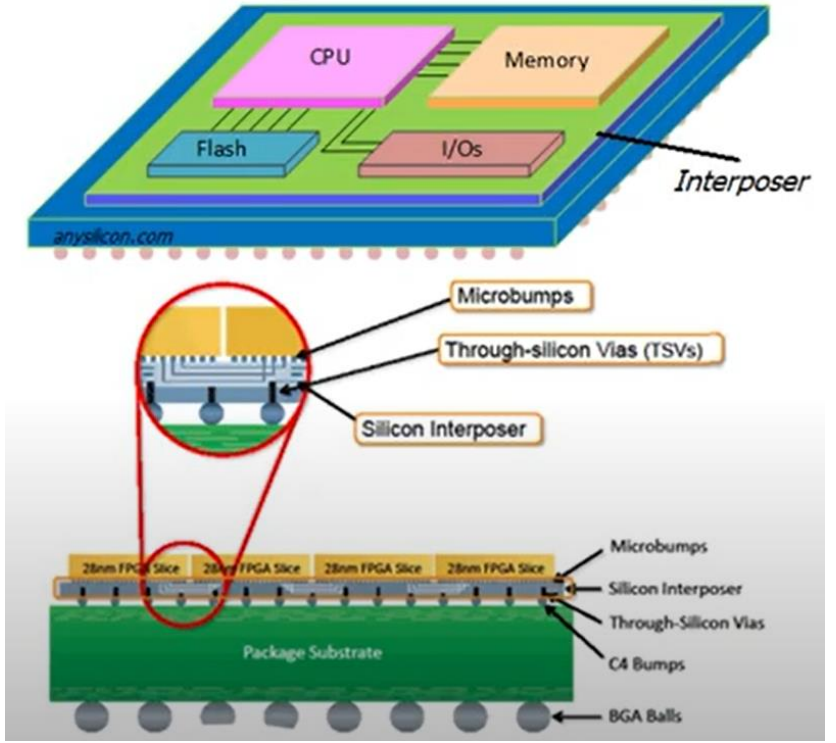

Sergi Abadal, 2020




Jacob Minz, 2005

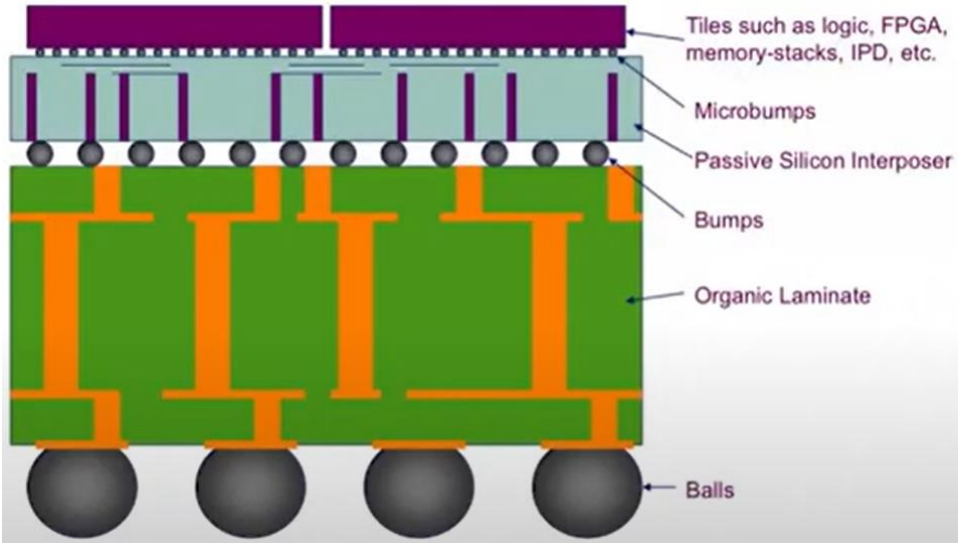
# What is a Silicon Interposer

- **Silicon interposer**
  - Silicon layer between substrate and dies
  - Serves as a connection
    - Among dies and to the I/Os
  - Uses Through Silicon Vias (TSV)
  - Can be active or passive



# Passive Interposer

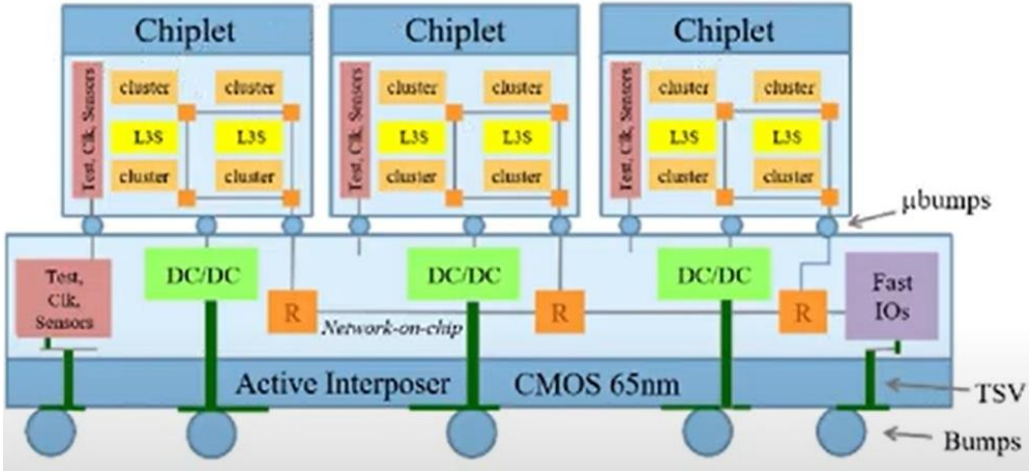
- **Passive Interposer**
  - Acts as an interconnection
    - Holds all the dies together
  - No active devices in substrate
  - Cannot perform functions
  - Higher wiring density




Navid Asadi, 2022

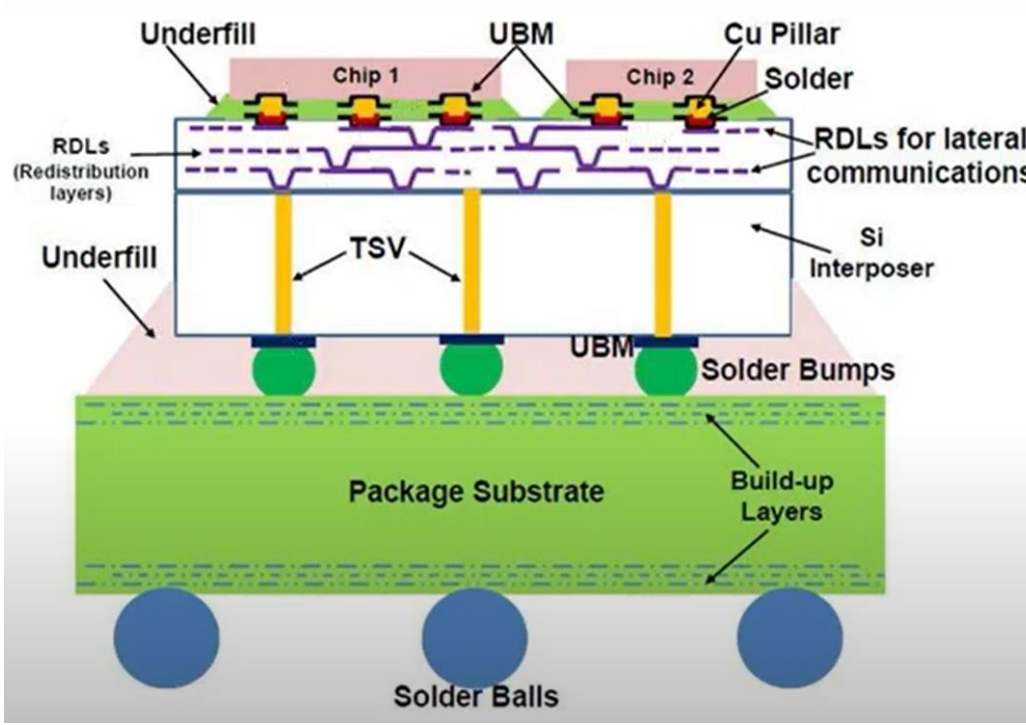
# Active Interposer

- **Active Interposer**
  - Acts as interconnection
  - Fully functional chips embedded in the silicon substrate
  - Lower TSV density
    - Keep-Out-Zone (KOZ)
      - The thermal stresses can cause different reliability issues
        - Carrier mobility results in timing violations in the implemented circuit
      - Forbidden to any circuit in KOZ – no mobility changes can be occurred

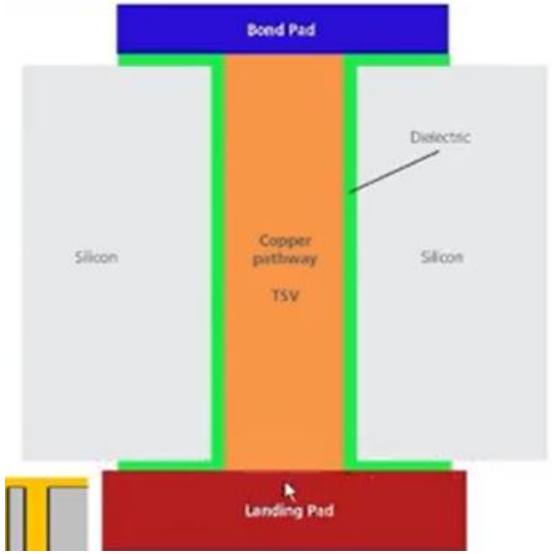


# Structure of the Interposer

- **TSV**
  - Through Silicon Vias
- **RDL**
  - Redistribution layer
- **UBM**
  - Under bump metallization



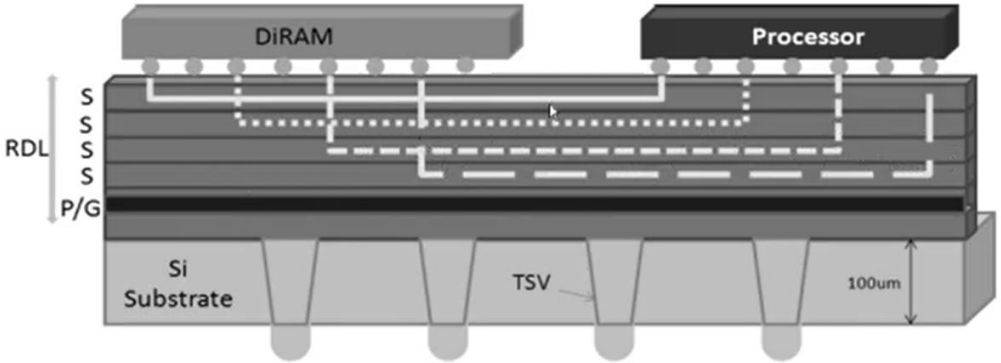
# TSV – Through Silicon Vias

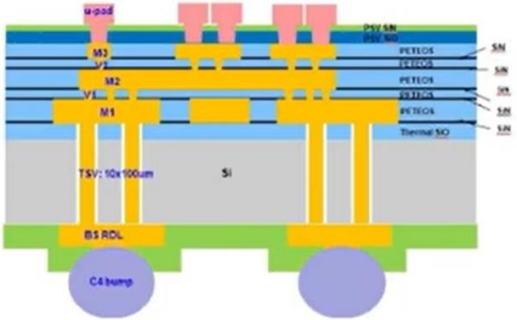



- **TSV – Through Silicon Vias**
  - Vertical pathway through silicon interposer
- **Very high aspect ratio**
  - Under 20 um wide, up to 200 um tall
- Composed of copper pathway, SiO2 isolator, and Barrier
- **Processing to make TSV**
  - Etching – make a hole
  - Oxidation – create a isolated layer that prevent the current leak to the silicon
  - Deposition – material to prevent the diffusion between cooper and oxide
  - Filling – filling the copper
  - Chemical Metal Polishing (CMP)

# RDL – Redistribution Layer

- Horizontal pathway along the interposer
  - Connect dies to one another
- Connects the solder bumps to the TSV by re-routing
- Cooper lines etched into SiO2 layers and polished

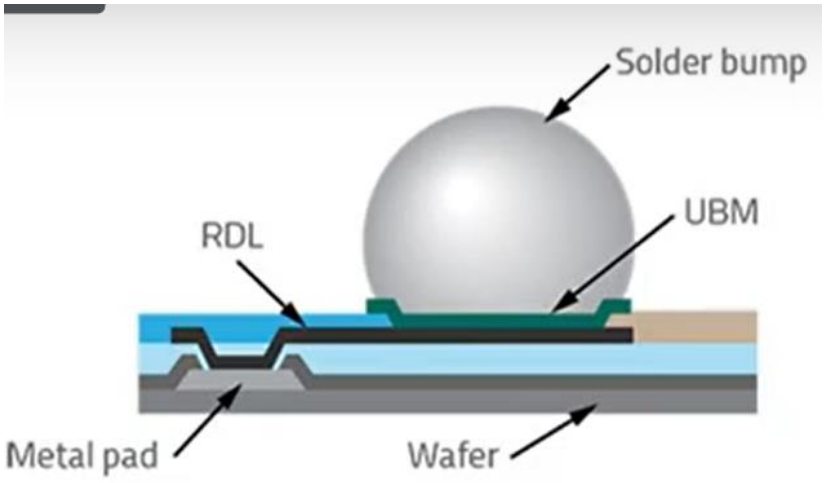



Navid Asadi, 2022

# UBM – Under Bump Metallization

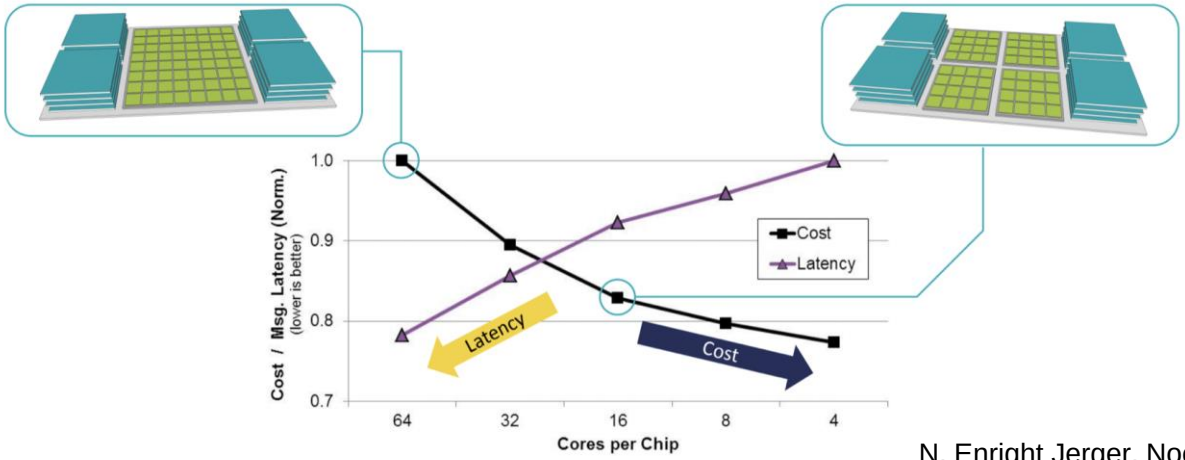
- **UBM – Under Bump Metallization**
  - Thin pad connecting the solder bump and the copper in the RDL
  - Serves as barrier to stop diffusion
  - Acts as a mechanical connection




Navid Asadi, 2022

# Fragmented Architecture

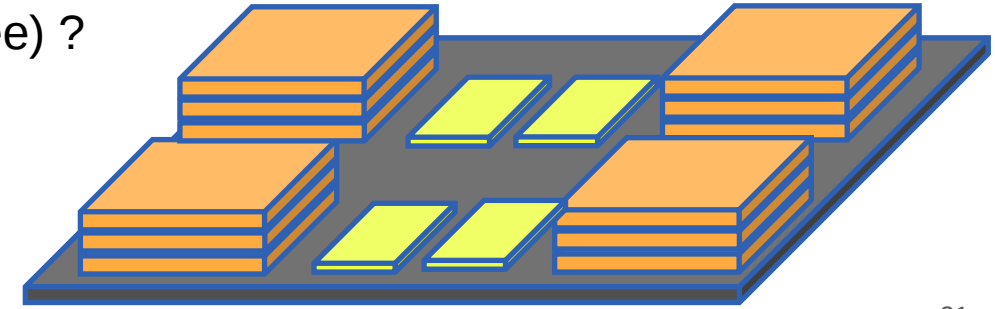
- Disintegrated SoCs can reduce costs of large chips
- But performance can degrade when the disintegration granularity is getting smaller, why ?
  - High latency connections (inter/intra chiplet)




N. Enright Jerger, Nocarc, 2020

# NoC on Interposer

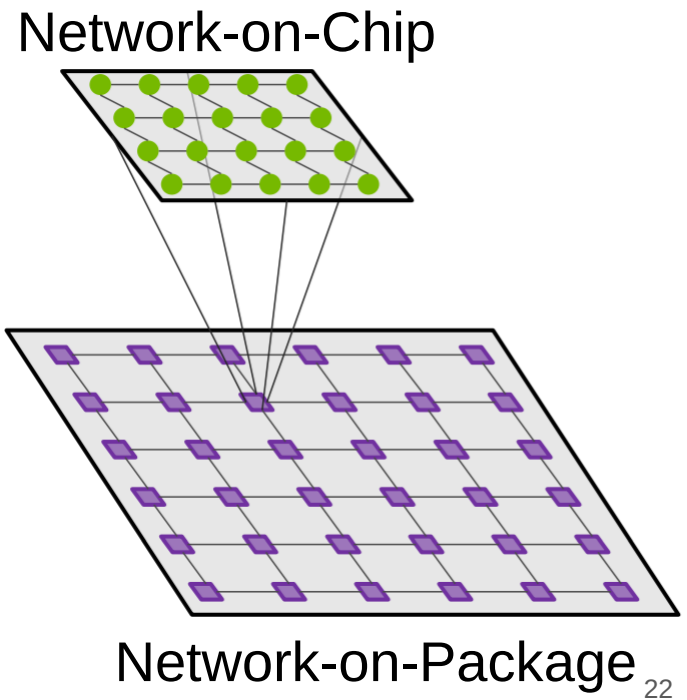
- A NoC that spans both chiplets and interposer
- Each chiplet may be designed independently
- Questions
  - How to choose NoC topology on chiplet ?
  - How to choose local routing algorithm within chiplet (deadlock free) ?



**Active silicon interposer**

# Simba: Multi-Chip Module Architecture

- **Network-on-Chip (NoC)**
  - 4 x 5 mesh topology
  - 16 PEs, one global PE, one RISC-V
  - Cut-through routing with multi-cast
  - 10 ns per hop, 68 GB/s/PE
- **Network-on-Package (NoP)**
  - 6 x 6 mesh topology
  - 36 chiplets in package
  - A NoP router per chiplet
  - Configurable routing
  - 20 ns per hop, 100 GB/s/chiplet

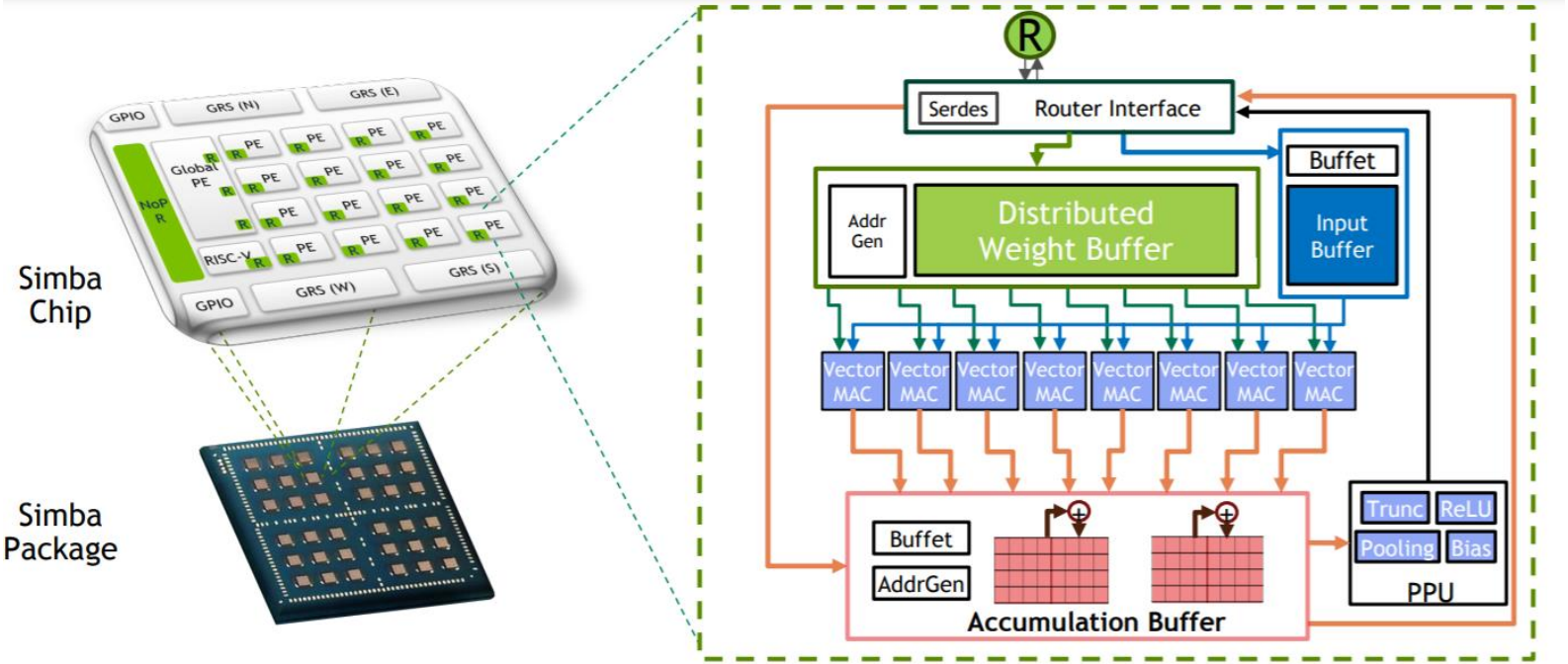





Shao et. al, MICRO, 2019

# Simba: Scalable MCM-Based Architecture

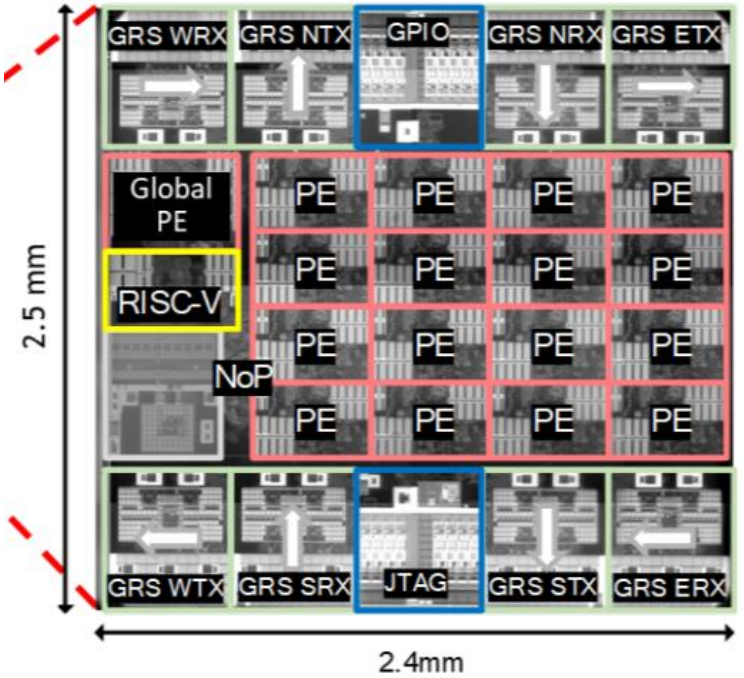
- Spatial Architecture with distributed memory




Shao et. al, MICRO, 2019

# Simba Chiplet and Package

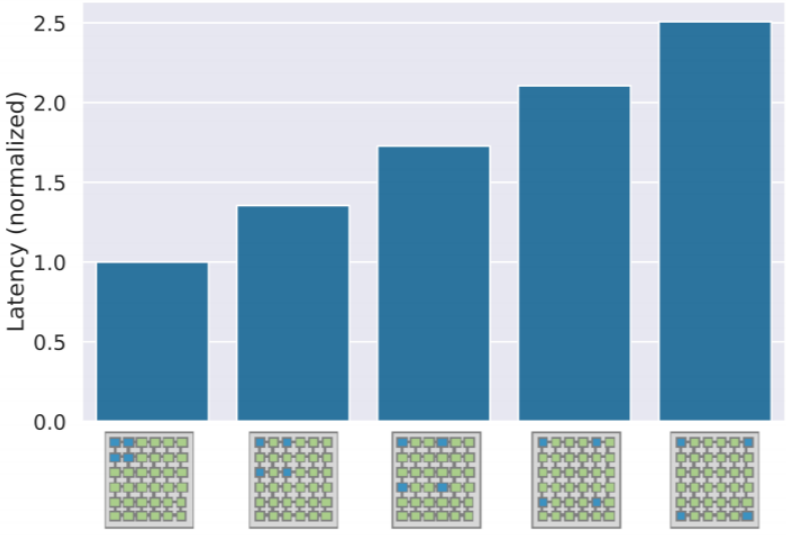
- 6 $mm^2$ chipet in TSMC 16 nm
- 36 chiplets/package
- Chip-to-chip interconnect
  - Ground-referenced signaling
- Efficient compute tiles
  - 128 TOPS
  - 0.11 pJ/Op
  - 8-bit integer datapath




Zimmer et al., VLSI 2019

# Network-on-Package Latency Sensitivity

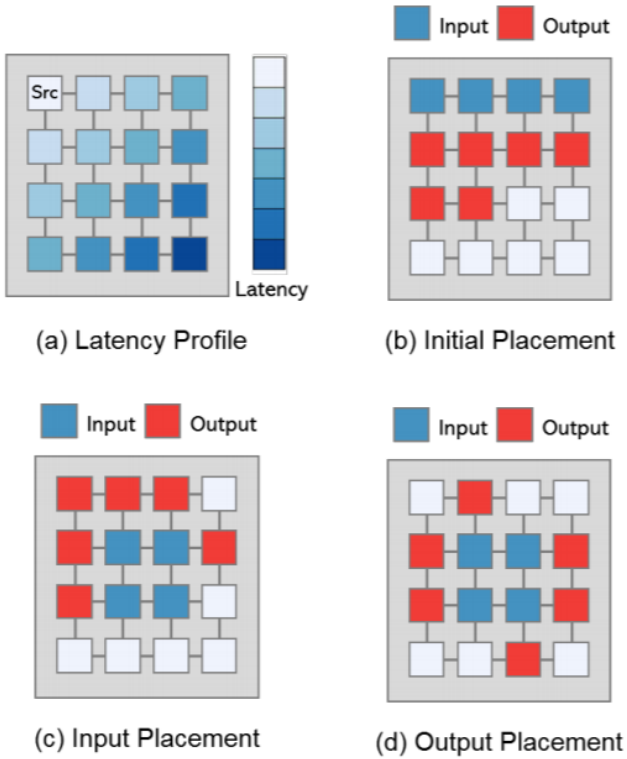
- **Non-uniform bandwidth**
  - Intra-chiplet bandwidth is higher than inter-chiplet
  - Sending data to remote chiplets incurs additional latency
- Mapping res4a_branch1 to 4 chiplets with different placements
- **Communication latency** is critical in a large-scale system




Shao et. al, MICRO, 2019

# Communication-aware Data Placement

- **Non-uniform work partition**
  - PEs closer to the data producers will perform more work -> maximize physical data locality
  - PEs are far away do less work to decrease the tail latency




Shao et. al, MICRO, 2019

# Open Challenges on Chiplet-based Systems

- **Active Interposer**
  - How to intelligently use and offload additional functionality to interposer ?
- **Process**
  - How to re-integrated SoC with varying die-to-die process ?
  - Independent clock domains
  - DVFS management
  - Mitigate overhead of clock crossings

N. Enright Jerger, Nocarc 2020

# Open Challenges on Chiplet-based Systems

- **3D NoC**
  - How to span NoC through multiple process technologies ?
- **Chiplet placement**
  - Chiplet placement determines NoC traffic patterns
  - Interaction between in-package memory stacks and external memories ?

N. Enright Jerger, Nocarc 2020

# Opportunity on Chiplet-based Systems

- **Heterogeneous SoCs**
  - Rise of accelerators to provide performance, power efficiency, and security
  - Cost-effective LARGE SoCs
- **Additional Challenges**
  - Interfaces
  - QoS
  - Coherence

N. Enright Jerger, Nocarc 2020

# Takeaway Questions

- What are advantages of chiplet-based system?
  - (A) Increase wafer utilization
  - (B) Reduce the network communication latency
  - (C) Simplifying the manufacture of a single chip
- What are challenges of chiplet-based system?
  - (A) Inter-chip communication increases the latency when the granularity of chip is getting small
  - (B) Non-uniform bandwidth requirement
  - (C) Increase the speed of the processor

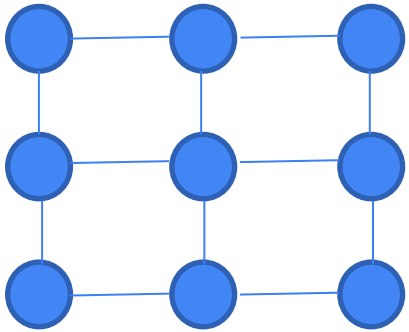
# On-chip vs Off-chip Networks

- **Off-chip network**
  - I/O bottleneck
  - Connect clusters of workstations
- **On-chip network**
  - Communication latency is critical on multi-core chips
    - Aim to increase core utilization
    - Synchronization between threads
  - A scalable **low-latency** and **high-bandwidth** communication fabric
  - **Power constraints** -> place many interconnection components under tight chip area (~30% chip power is from Intel's 80-core TeraFLOPS network)

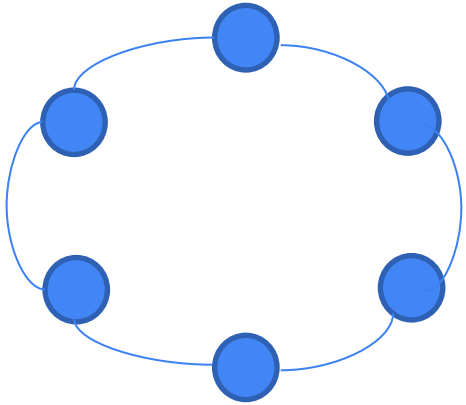
# On-chip Network Topology

- Determine the physical layout and connections between nodes and channels in the network
- Determines the number of hops/routers and a message traverse
- **The number of links** affects the complexity of a topology
- Impact **network latency and energy consumption**

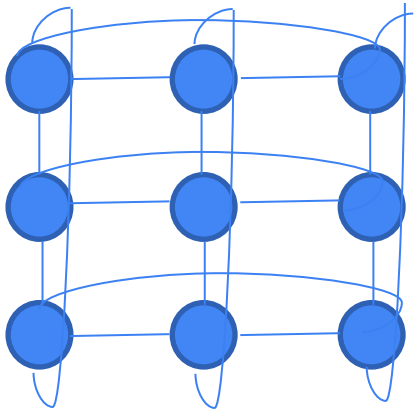
# Network Topology



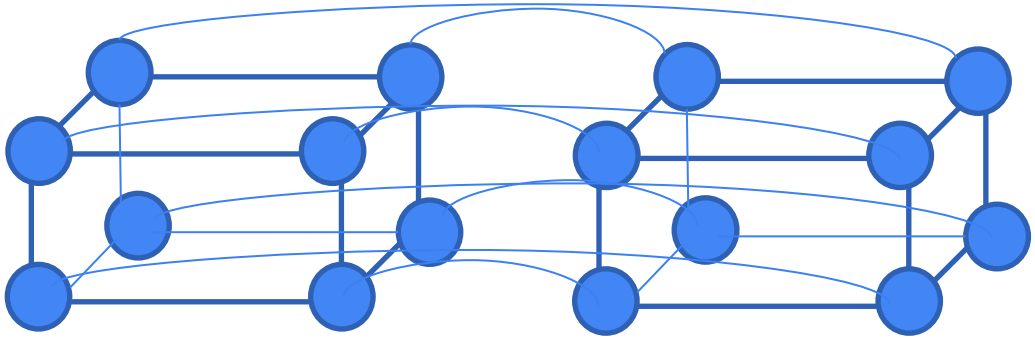
**Mesh(Grid)**



**Ring**



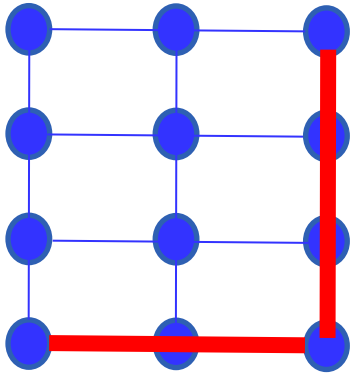
**Torus**



**Hypercube**

# NoC Routing

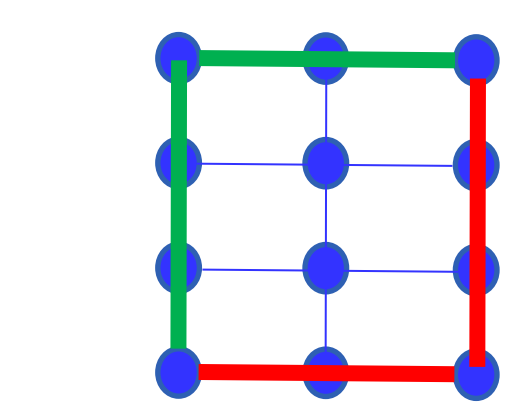
- Distribute traffic among paths supplied by network topology
- Avoid hotspots and minimize contention
- **Deterministic routing**
  - Given the source and destination, there is a unique route
- **Dimension-ordered routing (DOR)**
  - Packet first routed to correct position in higher dimension before attempting to route in next dimension.
  - For example in a 2D mesh, route first in X dimension, then in Y dimension.

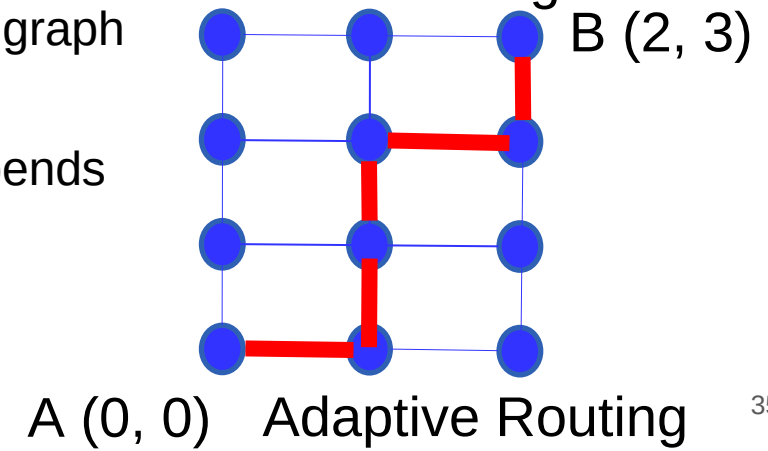



DOR Routing

# NoC Routing

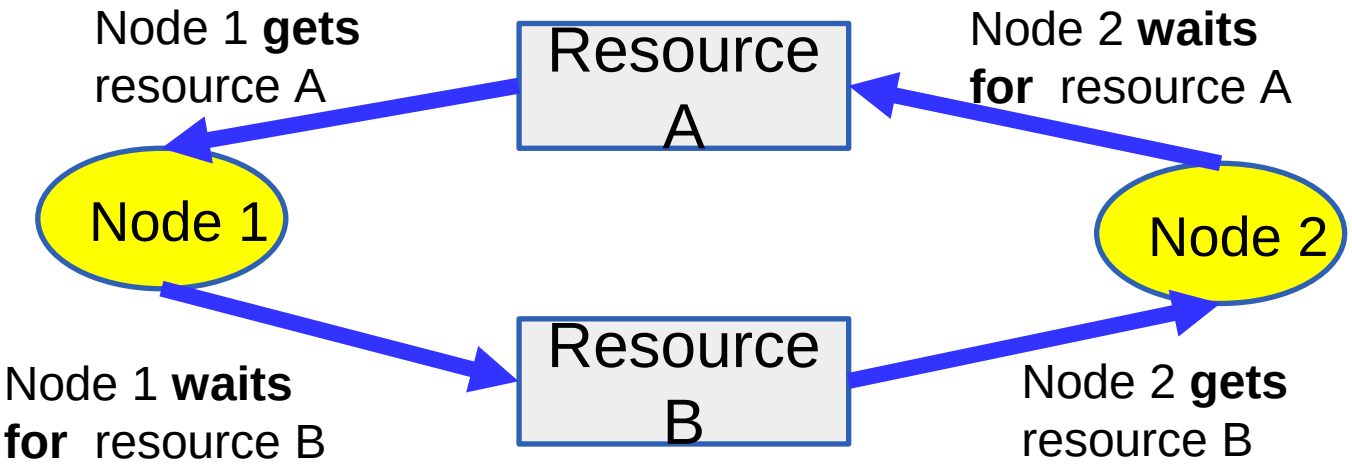
- **Oblivious routing**
  - Routing paths are chosen without regard to the state of the network. (why ?)
  - Keep routing algorithm simple
  - Aim to minimize the minimize the maximum congestion on any edge in a graph
- **Adaptive routing**
  - A message takes from node A to B depends on network traffic situation
  - E.g. congestion is encountered at (1,0)



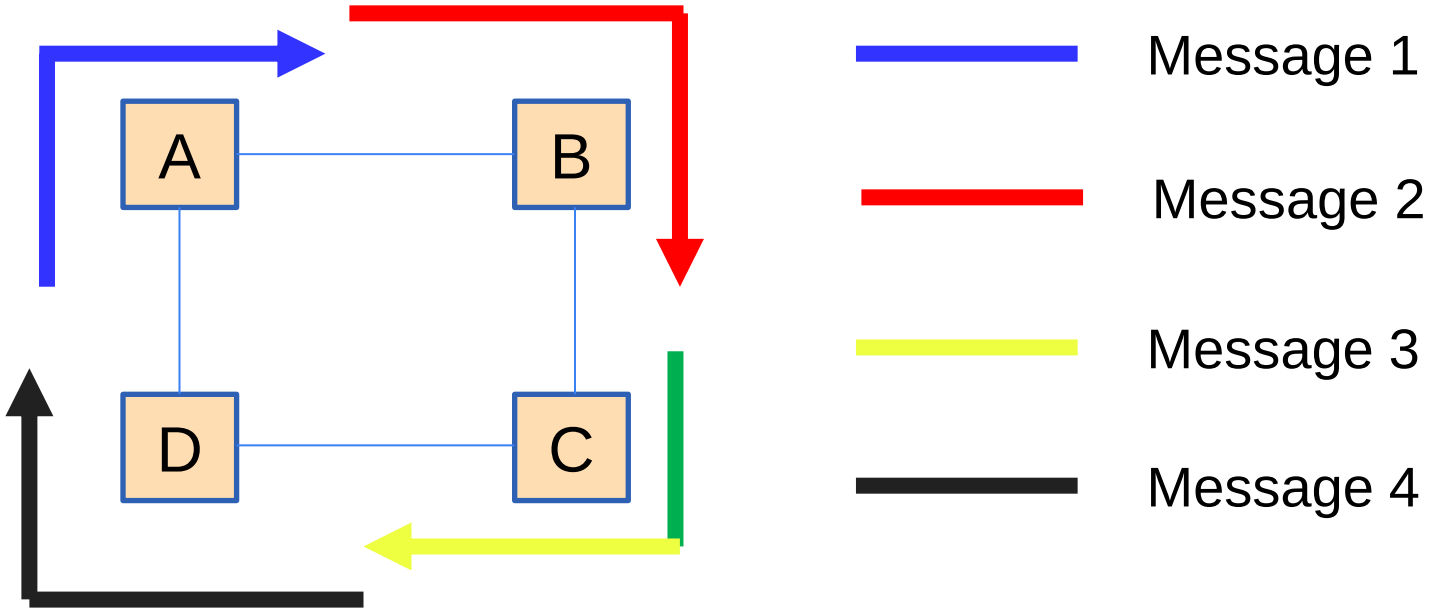
# Deadlock

- Deadlock occurs when there is a cycle of resource dependencies
- A node holds on a resource (A) and attempts to acquire another resource (B)
- A is not relinquished until B is acquired



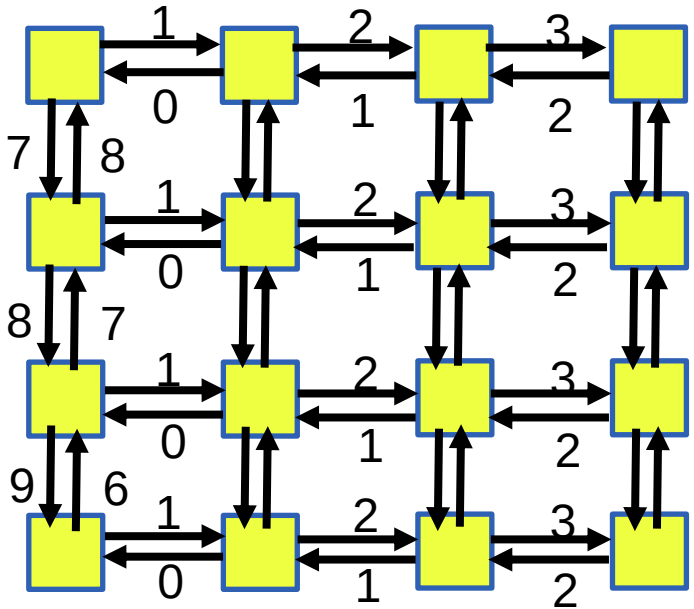
# Deadlock in Routing

- Each message is attempting to make a left turn through 4-way switch



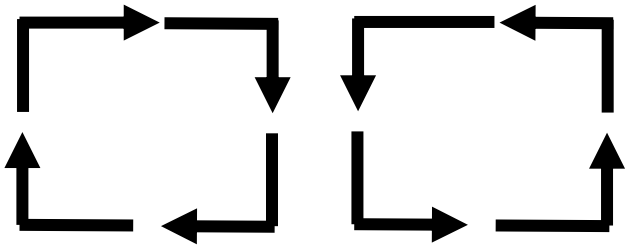
# Deadlock-Free Proof

- All routes will traverse edges in increasing (decreasing) order
- Example: k-ary 2-d array with dimension routing
- First route along x-dimension, then along y

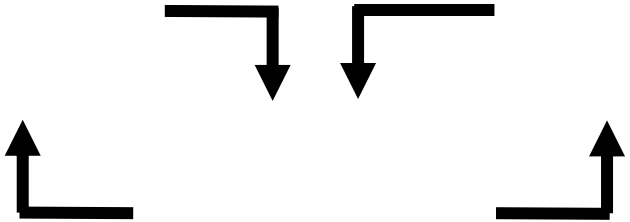


# Deadlock-avoidance routing

- To prevent cyclic dependencies, certain turns should be disallowed
- A message in X-Y turns travels east or west is allowed to turn north or south
- Messages traveling north and south are permitted no turns



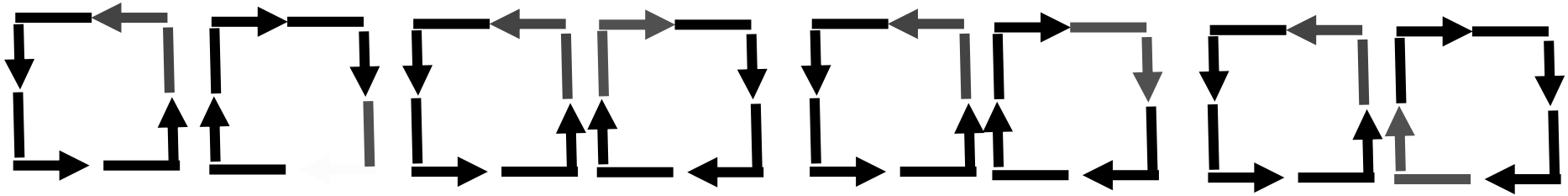
All turns



X-Y turns

# Adaptive Turn Model Routing

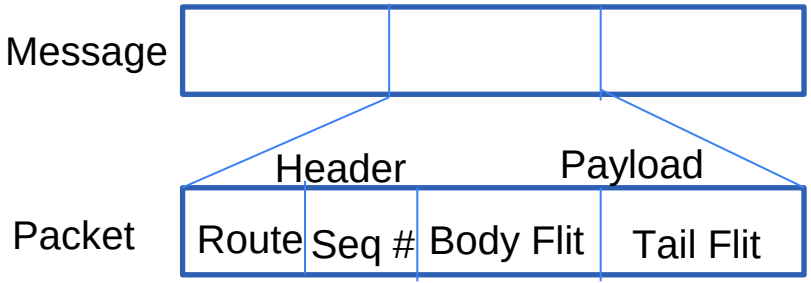
- Eliminate the minimum set of turns needed to achieve deadlock freedom
- Retain some path diversity and potential for adaptivity

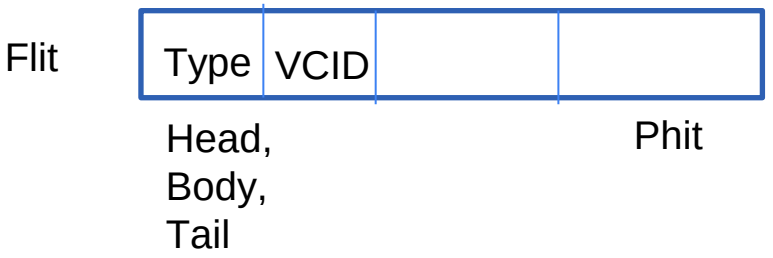


**West-First** **North-First** **Negative-First** **Can allow deadlocks**

# Messages, Packets, and Flits

- A **message** is segmented into multiple **packets**
- **Packets** have header information that allows the receiver to re-construct the original message
- A packet is divided into fixed-length **flits**, do not have additional headers, short for flow control units
- Flits of a packet must take the same route (why ?)



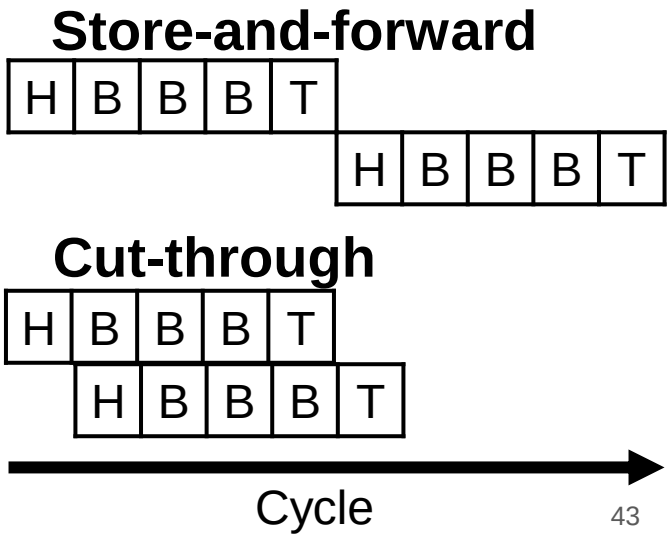



# Flow Control

- Determine the allocation of buffer and link resources shared among many messages on the network
- **Bufferless**
  - Flits are dropped if there is contention for a link
  - NACKs are sent back
  - The original sender has to re-transmit the packet
- **Circuit switching**
  - Links are pre-reserved without buffers at each hop -> save power
  - The request may be held at an intermediate router until the channel is available -> poor link bandwidth utilization

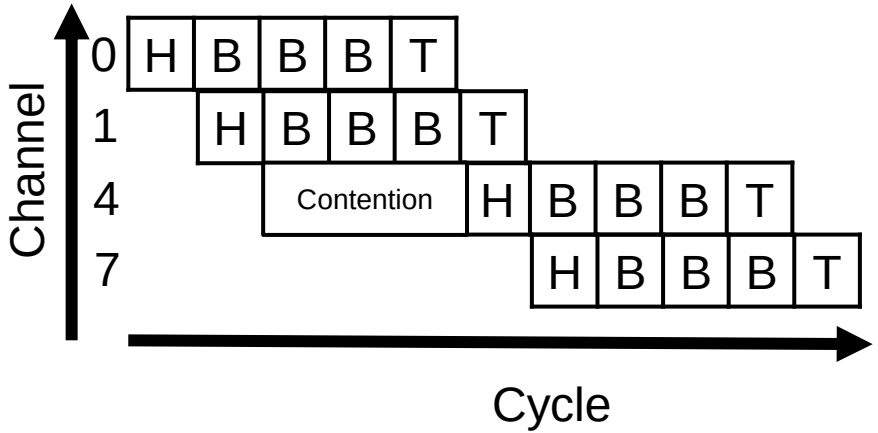
# Buffered Flow Control

- Unlike circuit switching, per-node buffering is required
- **Store-and-forward**
  - Each node waits until an entire packet has been received before forwarding packet to the next node
  - Long delays are incurred at each hop
- **Cut-through**
  - Transmit a packet without completing to receive the entire packet



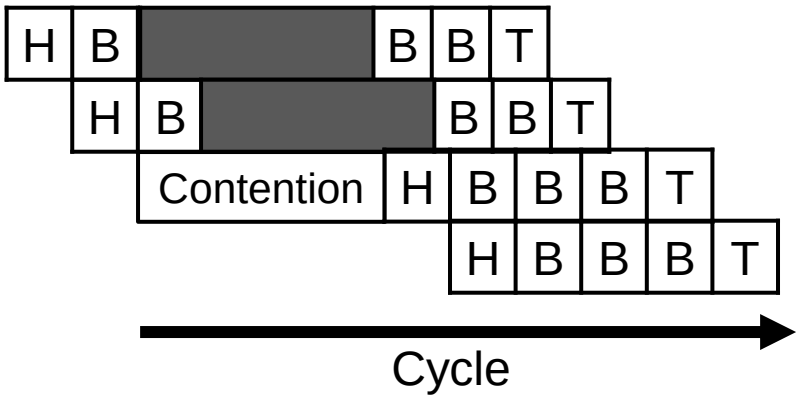
# Buffered Flow Control

- **Cut-through with delay**
  - Bandwidth and storage are allocated in packet-sized units
  - Packets move forward only if there is enough storage at the next downstream to hold the entire packet
  - No flits can proceed until all 5 flit buffers are available



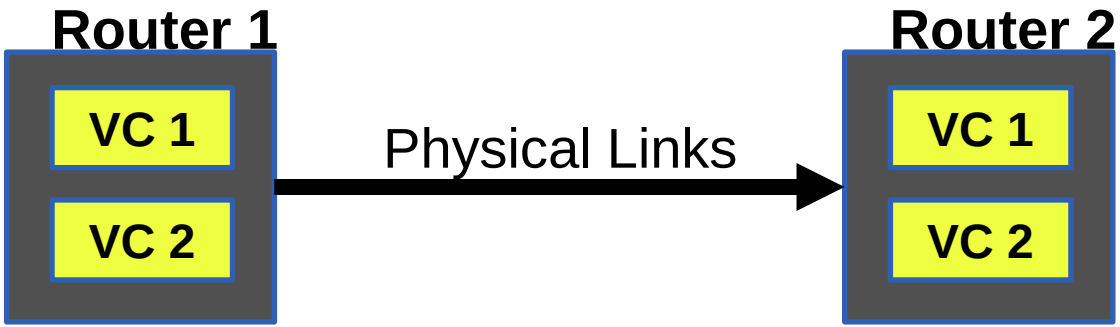
# Buffered Flow Control

- **Wormhole**
  - Allow flits to move on to the next router before the entire packet is received at the current channel
  - Allocate buffering to flits rather than entire packets
  - Allow smaller flit buffers to be used in each router
  - When a packet is blocked, all of the physical links held by that packet are idle



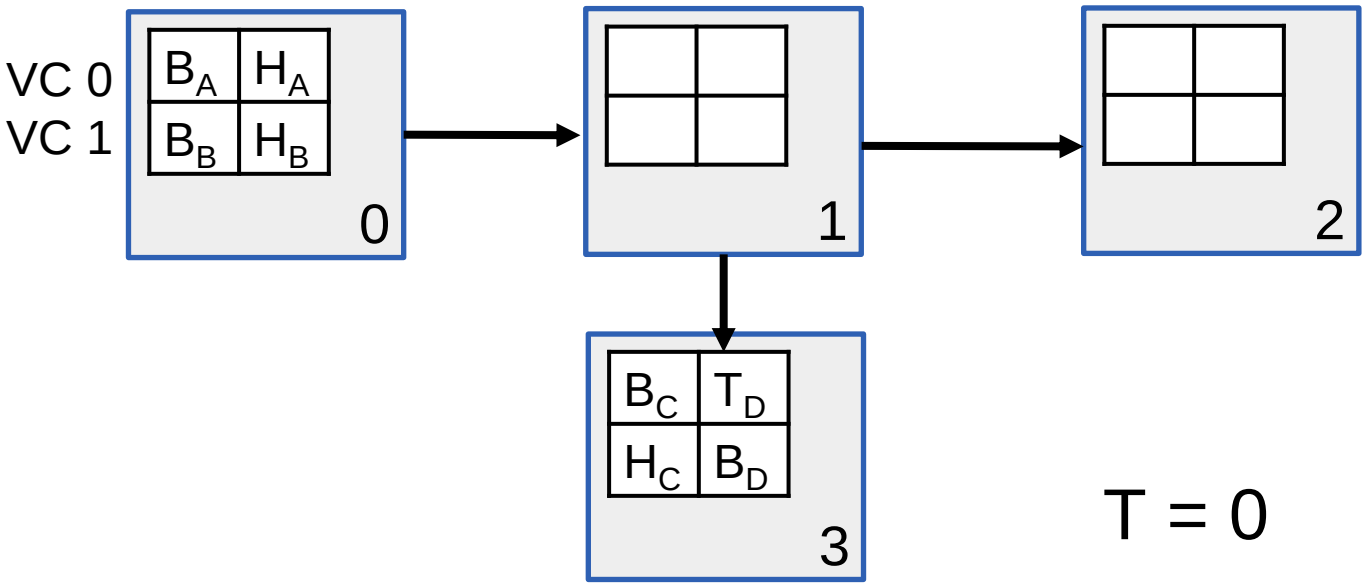
# Virtual Channels

- **Head-of-line blocking (HB)**
  - When a packet is blocked and stalls subsequent packets behind it
  - Even when there are available resources for the stalled packets
- **Virtual channel (VC)**
  - Multiple VCs share the physical link between router
  - Multiple separate queues with each input port, reduce HB



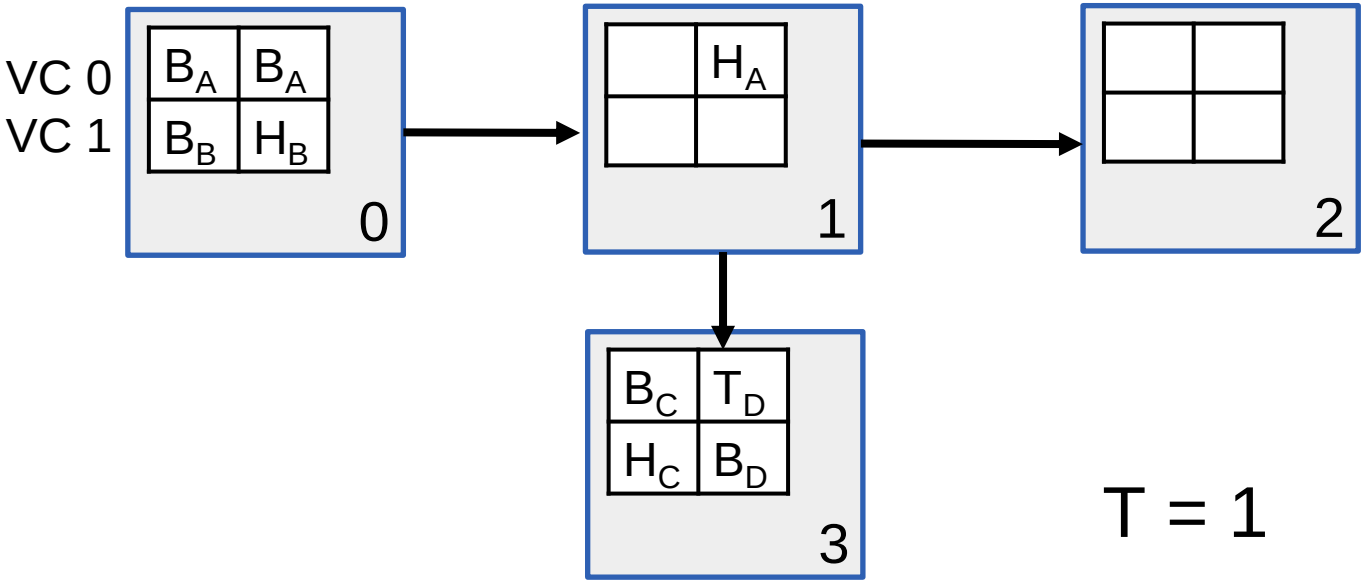
# Virtual Channel Flow Control

- Two packets A and B are divided into 4 filts each (H: head, B: Body, T: Tail)



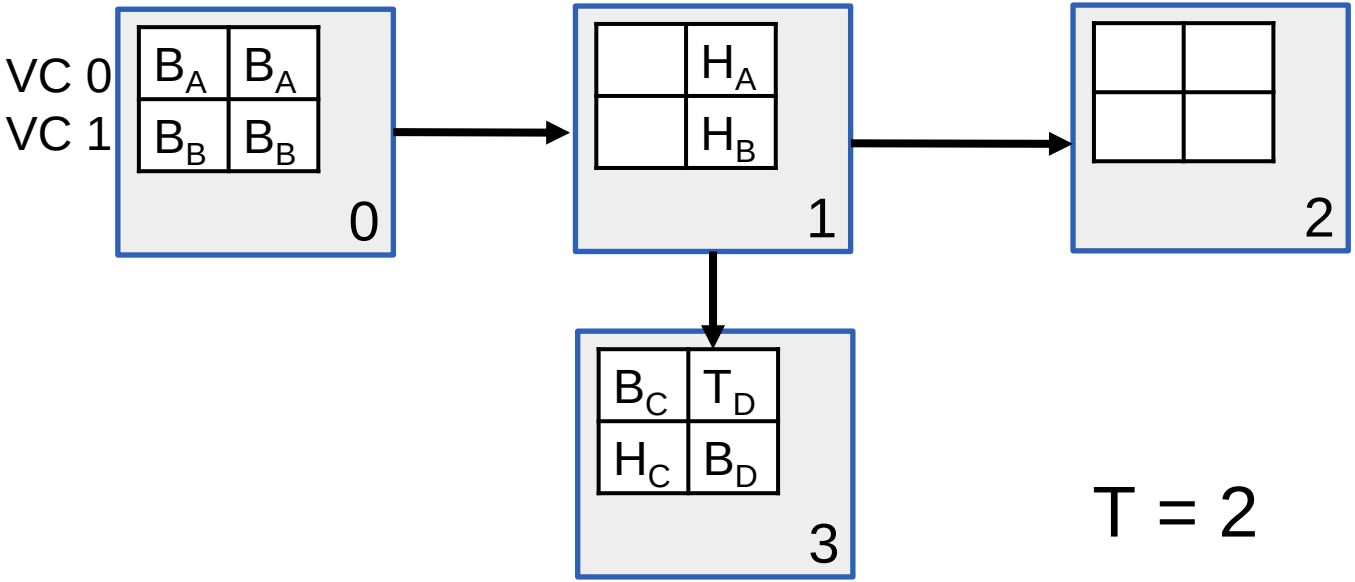
# Virtual Channel Flow Control

- Packet A destination is router 3
- The head flit of Packet A is allocated VC 0 on router 1



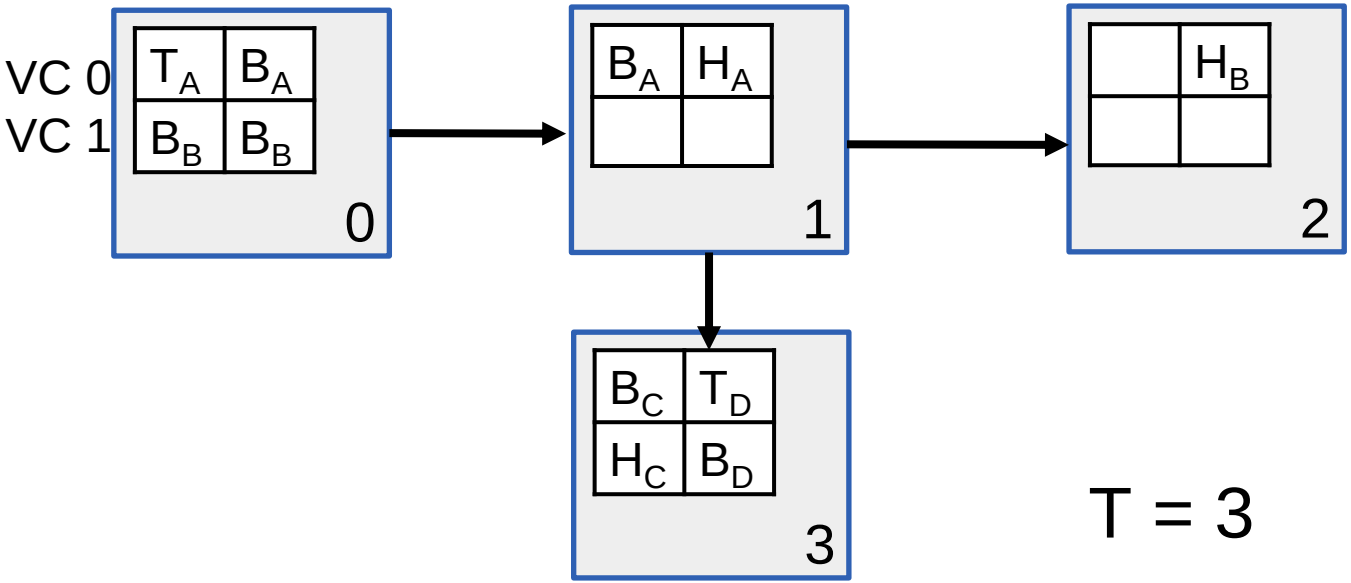
# Virtual Channel Flow Control

- The head flit of packet B travels to router 1 and is stored in VC 1
- $H_A$ fails to receive VC for router 3 that is occupied by others



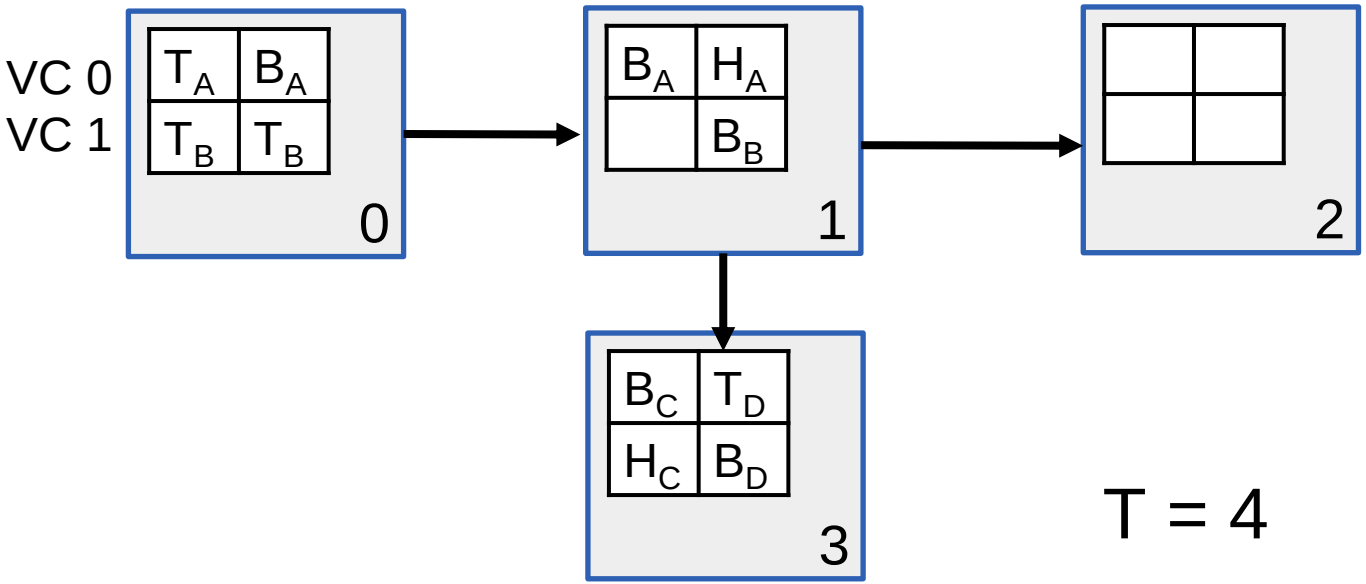
# Virtual Channel Flow Control

- The first body flit of A travels to router 1 and is store in VC 1
- The first body flit of B allocates VC 0 at router 2



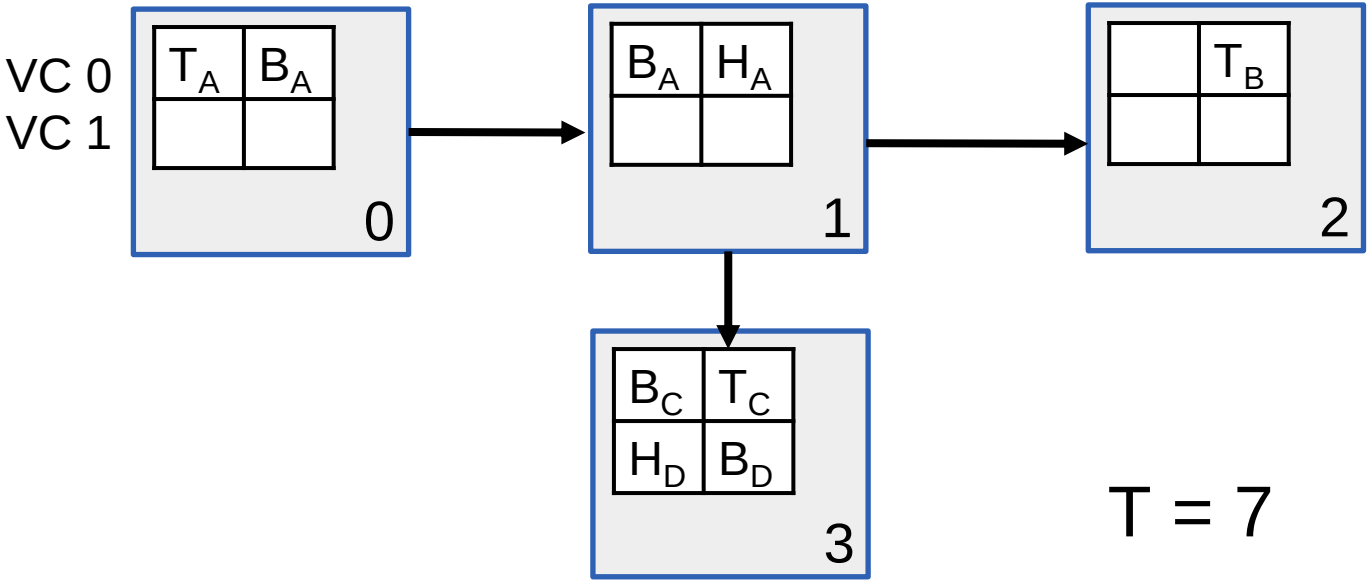
# Virtual Channel Flow Control

- The first body flit of packet B continues to router 1



# Virtual Channel Flow Control

- At time 7, all of the flits of B have arrived at router 2
- The head flit of packet A is still blocked waiting for a free VC on router 3



# Advantages of Virtual Channel

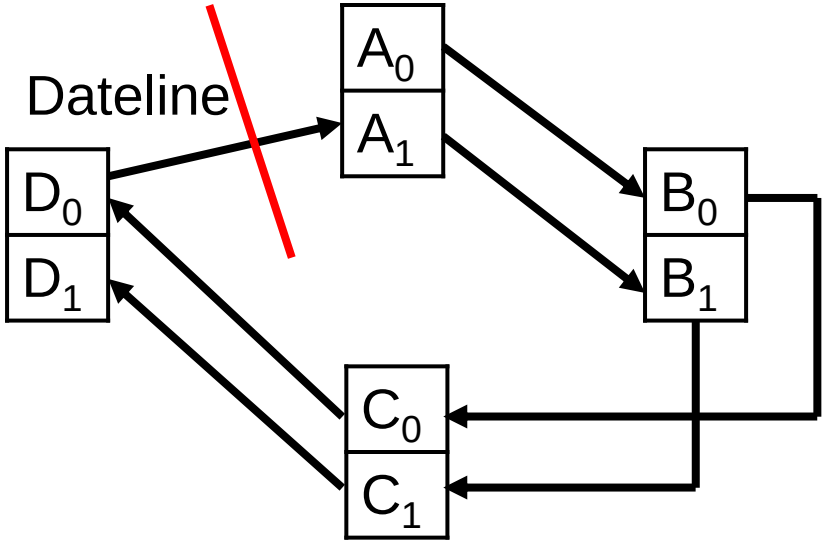
- **Wormhole flow control**
  - Using a single VC, packet B would be blocked behind packet A at router 1
- **Virtual Channels**
  - Allow packet B to proceed toward its destination despite the blocking of packet A
  - Flits of different packets can be interleaved on the same physical link (allow N packets transit over a given link)
  - The packet must carry an ID to indicate its VC
  - **Circular dependencies are removed** by assigning different network flows to disjoint VCs

# Summary of Flow Control Techniques

| | Links | Buffers | Comments |
|---|---|---|---|
| Circuit-Switching | Messages | Buffer-less | Require setup and ACK. |
| Store and Forward | Packet | Packet | Head flit must wait for the completion of entire packet |
| Cut Through | Packet | Packet | Head can begin next link traversal before tail arrives at current node |
| Wormhole | Packet | Flit | Head of line blocking |
| Virtual Channel | Flit | Flit | Interleave flits of different packets on links |

# Deadlock-Free Flow Control

- All messages are sent through VC 0 until they cross the dateline
- After crossing the **dateline**, message are assigned to VC 1 and cannot be allocated to VC 0 during the remainder of their network traversal
- Channel dependency graph (CDG) is **acyclic**

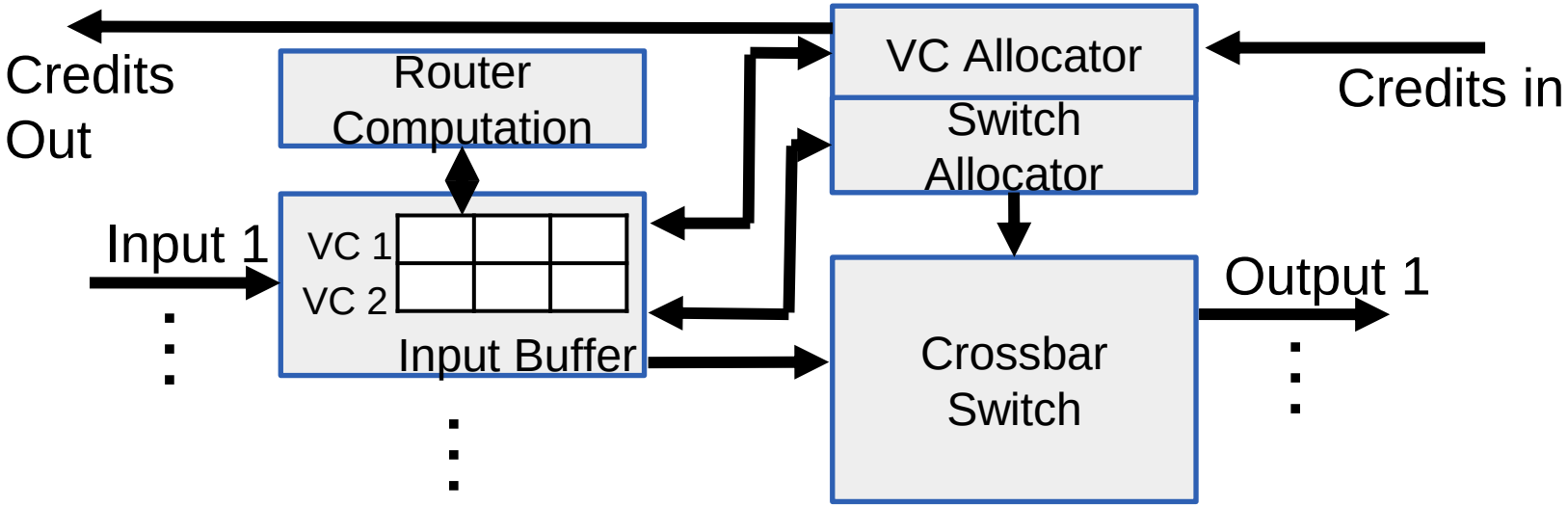


# Buffer Backpressure

- Most on-chip networks **CANNOT** tolerate the dropping of packets
- Buffer backpressure mechanism for stalling flits
- Flits **must not** transmitted when the next hop will not have buffers available to house them
- **Credit-based**
  - Keep track of free buffers in the downstream node; the downstream node sends back signals when a buffer is freed; need enough buffers to hide the round-trip latency
- **On/Off**
  - Upstream node sends back a signal when its buffer are close to being full
  - Reduce upstream signaling and counters, but can waste buffer space

# Router Microarchitecture

- The buffer stores flits
- Allocators determine which flits are selected to crossbar
- The crossbar switch moves flits from input to output port



# Allocator and Arbiters

- An allocator matches ***N*** requests to ***M*** resources
- An arbiter matches ***N*** requests to ***1*** resource
- **Virtual Channel Allocator**
  - Resolve contention for output VC and grants them to input VCs
  - Only the head flit of a packet needs to access VC allocator
- **Switch Allocator**
  - Grants crossbar switch ports to input VCs
  - Accessed by all flits and grants access to the switch on a cycle-by-cycle basis
- The allocation logic decides cycle time -> need fast pipeline for high clock frequency router

# Matrix Arbiter

- The least recently served requestor has the highest priority
- Initially, the priority of request is D > C > B > A

T = 0

| | | | | |
|---|---|---|---|---|
| A | X | 0 | 0 | 0 |
| B | 1 | X | 0 | 0 |
| C | 1 | 1 | X | 0 |
| D | 1 | 1 | 1 | X |

T = 1

| | | | |
|---|---|---|---|
| X | 0 | 0 | 1 |
| 1 | X | 0 | 1 |
| 1 | 1 | X | 1 |
| 0 | 0 | 0 | X |

T = 2

| | | | |
|---|---|---|---|
| X | 0 | 1 | 1 |
| 1 | X | 1 | 1 |
| 0 | 0 | X | 0 |
| 0 | 0 | 1 | X |

T = 3

| | | | |
|---|---|---|---|
| X | 1 | 1 | 1 |
| 0 | X | 0 | 0 |
| 0 | 1 | X | 0 |
| 0 | 1 | 1 | X |

T = 4

| | | | |
|---|---|---|---|
| X | 0 | 0 | 0 |
| 1 | X | 0 | 0 |
| 1 | 1 | X | 0 |
| 1 | 1 | 1 | X |

T = 5

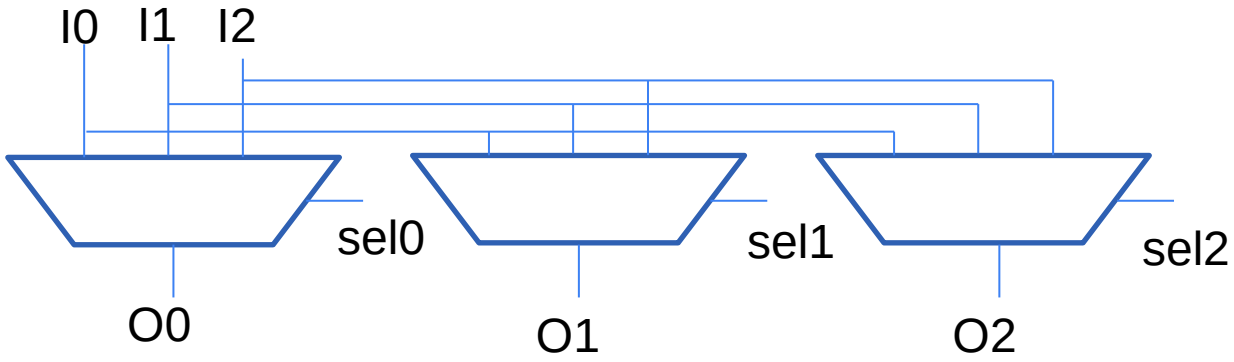
| | | | |
|---|---|---|---|
| X | 0 | 0 | 1 |
| 1 | X | 0 | 1 |
| 1 | 1 | X | 1 |
| 0 | 0 | 0 | X |

T = 6

| | | | |
|---|---|---|---|
| X | 0 | 1 | 1 |
| 1 | X | 1 | 1 |
| 0 | 0 | X | 0 |
| 0 | 0 | 1 | X |

# Switch Design

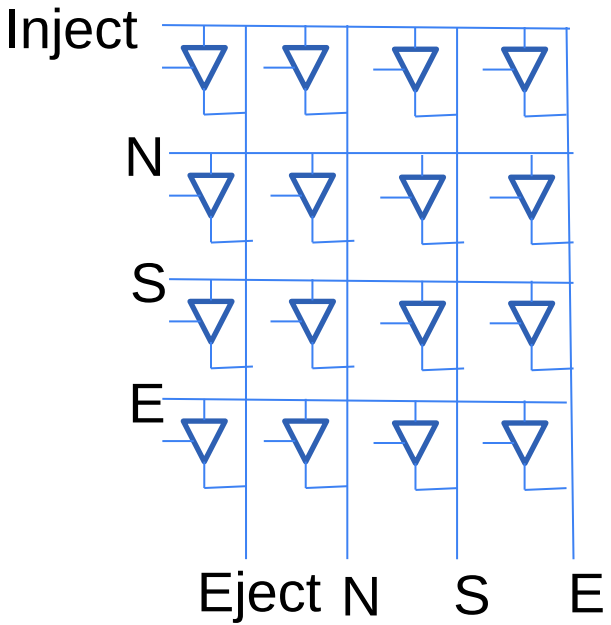
- **Crossbar Designs**




  - Input selects signals to each multiplexer
  - Input ports should be connected to which output ports
  - A crossbar contains many multiplexers

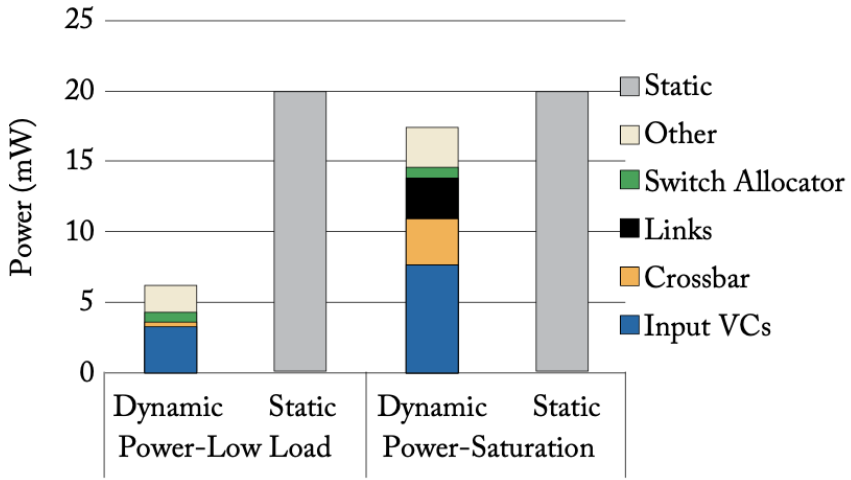
# Switch Design

- **Crosspoint crossbar switch**
  - Each horizontal and vertical line is *w* bits wide (1 phit)
  - Using select signals feeding each crosspoint
  - High speed switch with more stringent power budgets
  - A switch's area and power scale at **$O((pw)^2)$**
    p: # of crossbar port
    w is the crossbar port width in bits



# Low Power NoC Microarchitecture

- The power distribution for a mesh router with four VCs at 32 nm
- At high loads, **buffers and crossbar (links)** contribute **55%** and **34%** dynamic power, **static power** contributes **53%**
- At low load, **static power** contributes over **75%** total power consumption




Chen et. al., ICCAD, 2011

# Dynamic Voltage and Frequency Scaling (DVFS)

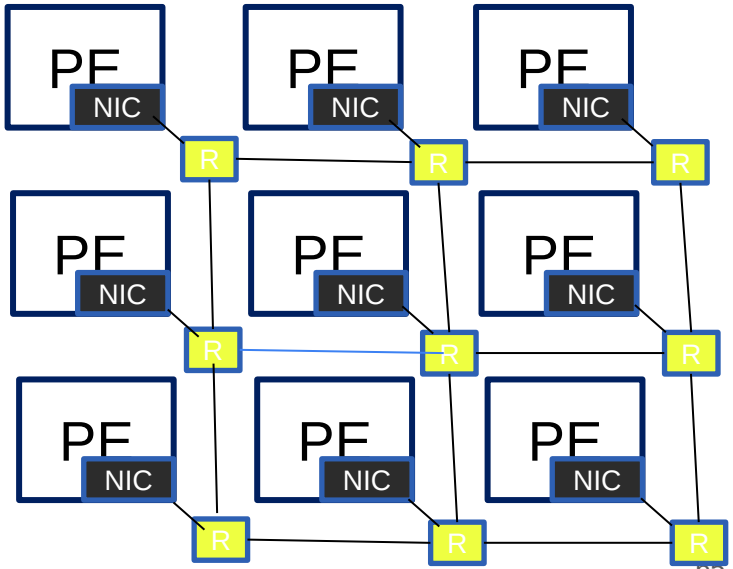
- **DVFS** operates lower voltage frequency state with less traffic router
- DVFS challenges on NoC
  - **Excess delays** from bi-synchronous FIFOs for multiple voltage-frequency islands
  - **Additional converters** are needed for multiple voltage rails

# Power-efficient NoC Designs

- **Reducing capacitance**
  - Low-wiring and equalized links -> Wires dominate network power
  - Reducing # of pipeline stage
  - Optimizing buffers, crossbar, and arbiter circuits/microarchitecture
  - E.g. Complex arbiters can be split into multiple simple ones
- **Reducing switching activity**
  - Reduce dynamic power
  - Clock-gating reduces the amount of switching activity of latches between inactive circuits
  - Efficient encoding reduces bit-toggles

# 2D Network-on-Chip

- Typical 2D NoC has multiple PEs arranged in a grid-like mesh structure
- The PEs are interconnected through underlying packet-based network fabric
- Each router connects to four adjacent routers

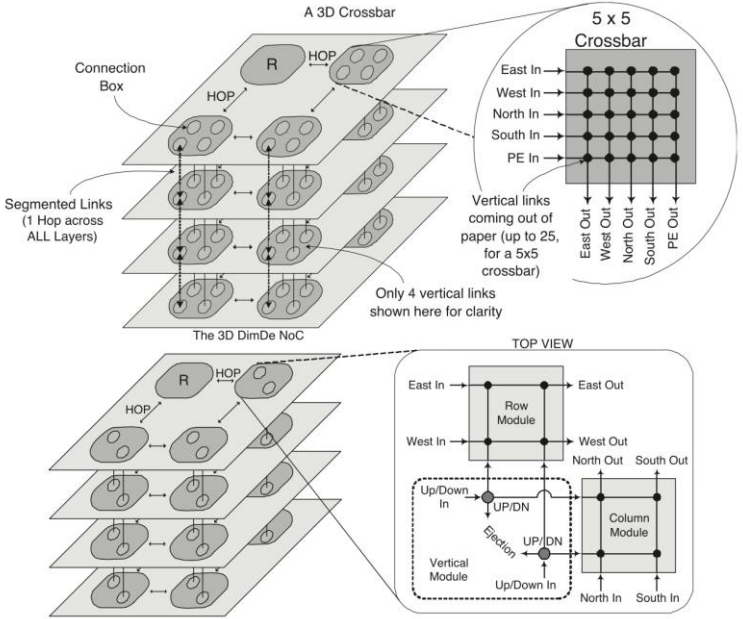


# 3D NoC Router Design

- **Symmetric NoC Router Design**
  - Simply add 2D routers in each 3D layer
- **Challenges**
  - Asymmetry delays in a 3D architecture between vertical and horizontal interconnects
  - Slow inter-wafer (vertical dimension) communication
  - Buffering and arbitration delay in moving up/down the layers
  - Crossbar scalability in 3D layers (area/power)

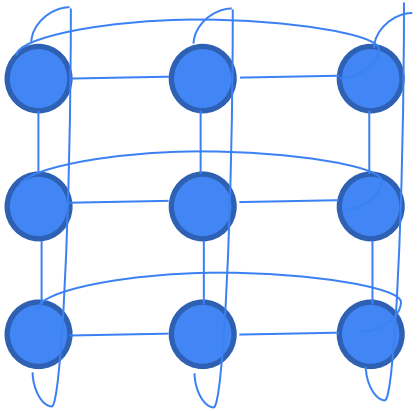
# 3D NoC Router Design

- **3D Dimensionally Decomposed NoC Router Design**
  - Use a limited amount of inter-layer links
  - Incoming traffic can be decomposed into two independent streams
  - **East-West** (packet movement in X dimension)
  - **North-South** (packet movement in Y dimension)
  - Small crossbars and isolation of two flows




Kim et. al, ISCA, 2007

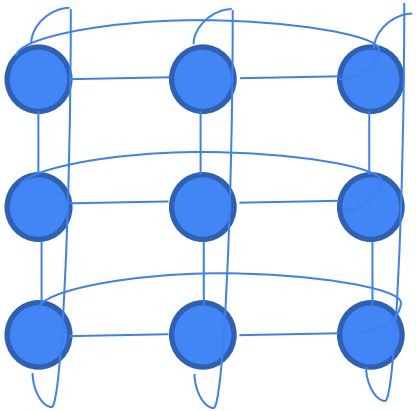
# NoC Topology Metrics



**Torus**

- **Degree**
  - The number of links at each node
  - What is the degree of torus topology ?
- **Bisection bandwidth**
  - The bandwidth across a cut that partitions the network into two equal parts
  - What is the bisection bandwidth of torus topology ?
  - Use to define the worst-case performance of a particular network

# NoC Topology Metrics

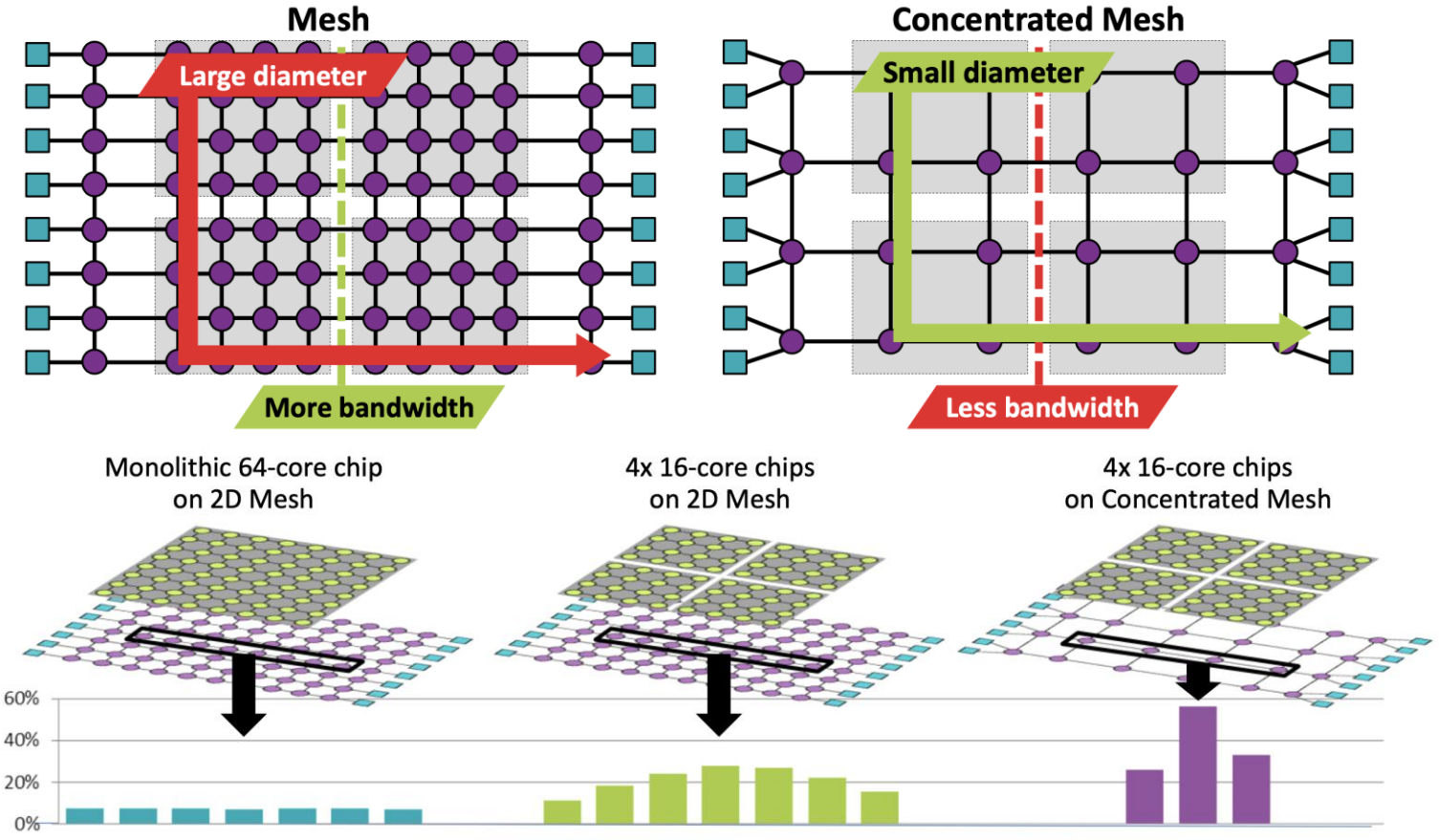


**Torus**

- **Diameter**
  - The max distance between any two nodes in the topology
  - What is the diameter of torus topology ?
  - Estimate the maximum latency in the topology
- **Hop Count**
  - The number of hops a message takes from source to destination
  - The number of links it traverses
  - The **maximum hop count** is **diameter** of the network

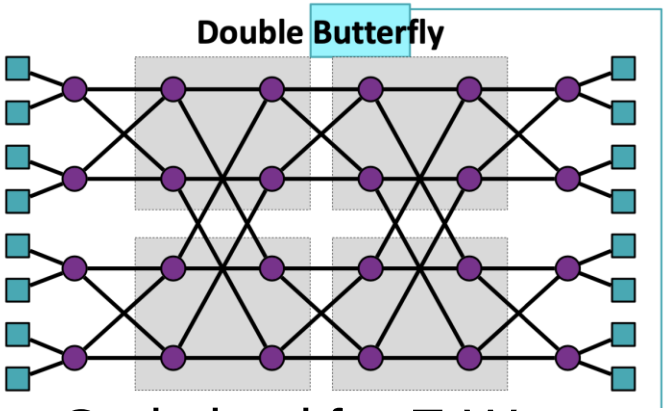
# Diameter vs Bisection Bandwidth

N. Enright Jerger, Nocarc 2020

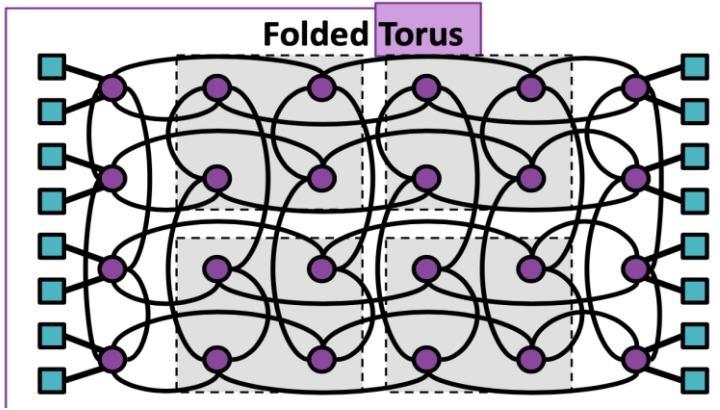


# ButterDonut Topology

N. Enright Jerger, Nocarc 2020



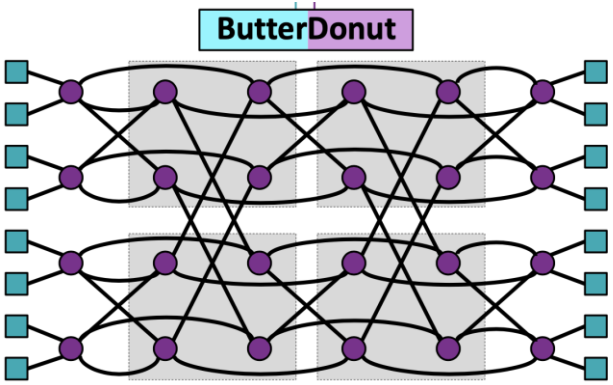

Optimized for E-W traffic




Fewer hops, +20% links

Smaller diameter

Lower Avg. Hop Count




+10% links vs Double Butterfly

Higher Bisection Bandwidth

# Takeaway Questions

- What are problems of bufferless flow control?
  - (A) Might increase the packet re-transmit rate
  - (B) Long delays are incurred at each hop
  - (C) Poor link bandwidth utilization
- How does virtual channel solve the head-of-line blocking?
  - (A) Adds the buffer on each hop
  - (B) Multiple VCs share the physical link
  - (C) Multiple separate queues with each input port

# Takeaway Questions

- How to reduce the NoC's static power consumption?
  - (A) Reduce the length of wiring
  - (B) Optimize the crossbar micro-architecture
  - (C) Using the clock gating
- What is the bisection bandwidth of the mesh network?
  - (A) 4
  - (B) 5
  - (C) 3