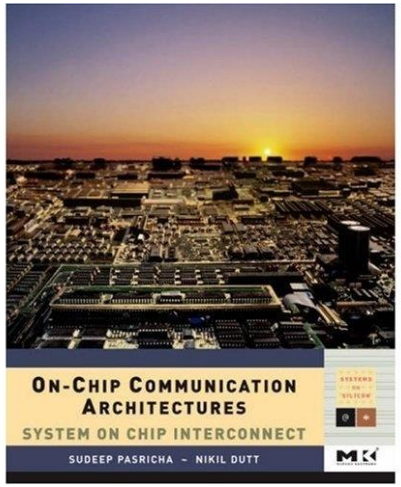
Topic 3

SoC Interconnect Architecture III*

Xinfei Guo
xinfei.guo@sjtu.edu.cn

October 21st, 2024



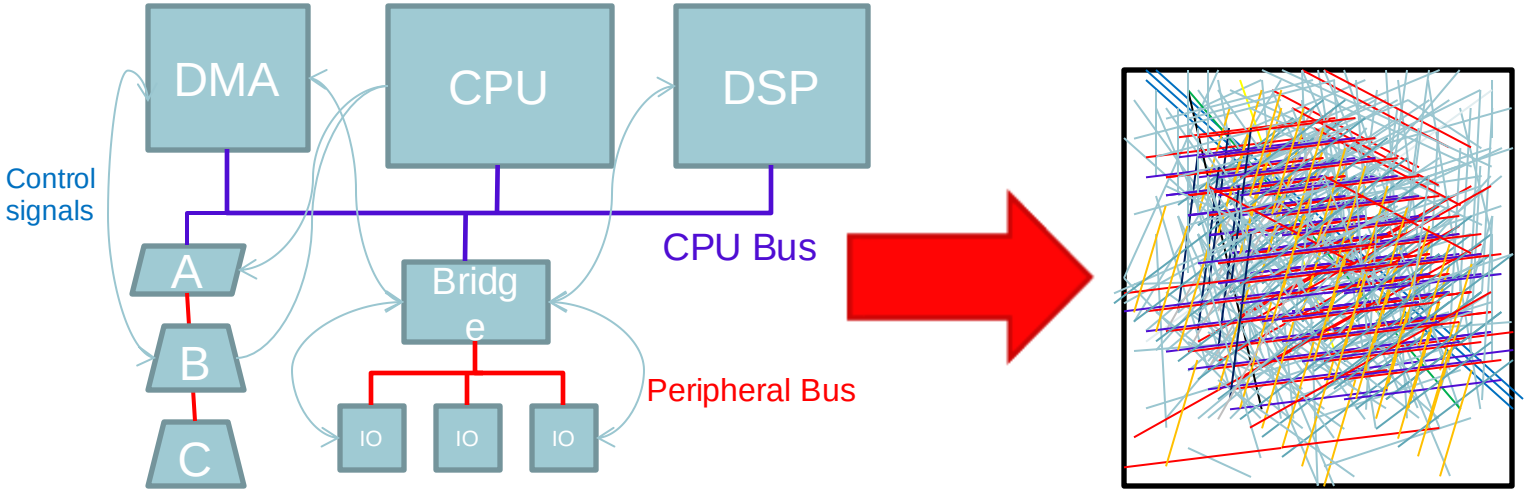
*Many materials are adapted from *"On-Chip Communication Architectures" by Sudeep Pasricha & Nikil Dutt, Morgan Kaufmann, 2008*

T3 learning goals

- How SoC components are connected together…
  - Introduction & Motivation
  - Bus
  - **Network on chip (NoC)**

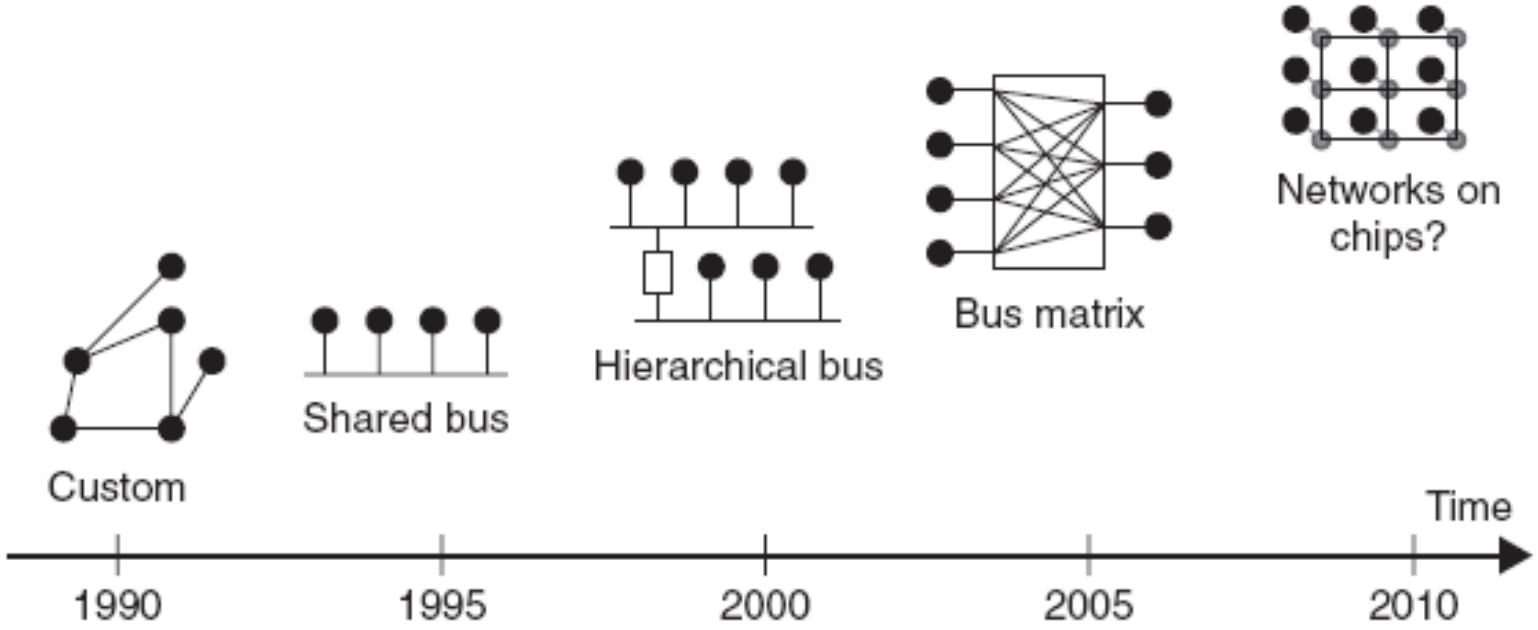
Traditional SoC Nightmare

- Variety of dedicated interfaces
- Design and verification complexity
- Unpredictable performance
- Many underutilized wires



Scalability also become questionable

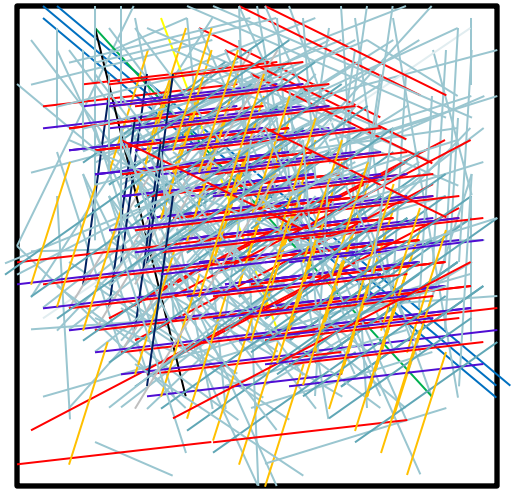
Evolution of on-chip Communication

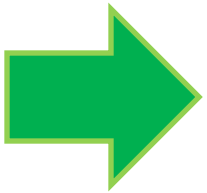


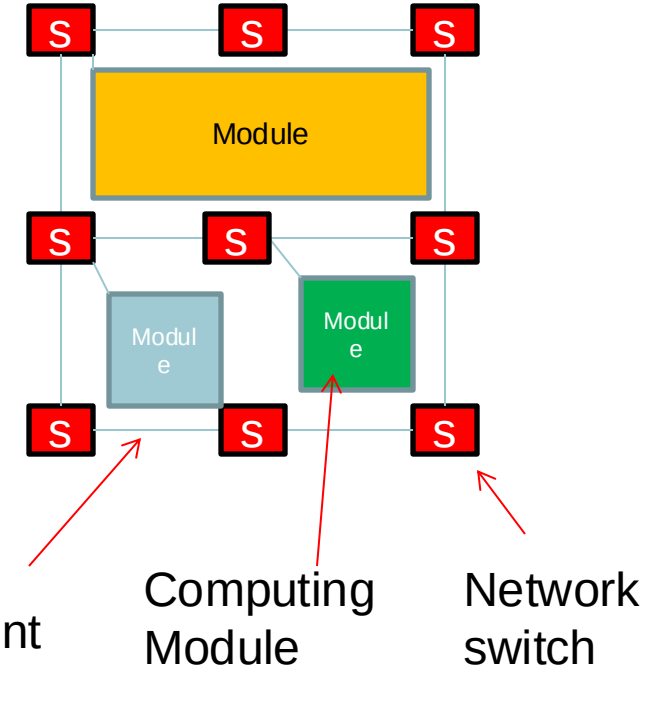
Network on Chip: A paradigm Shift in VLSI

From: Dedicated signal wires

To: Shared network

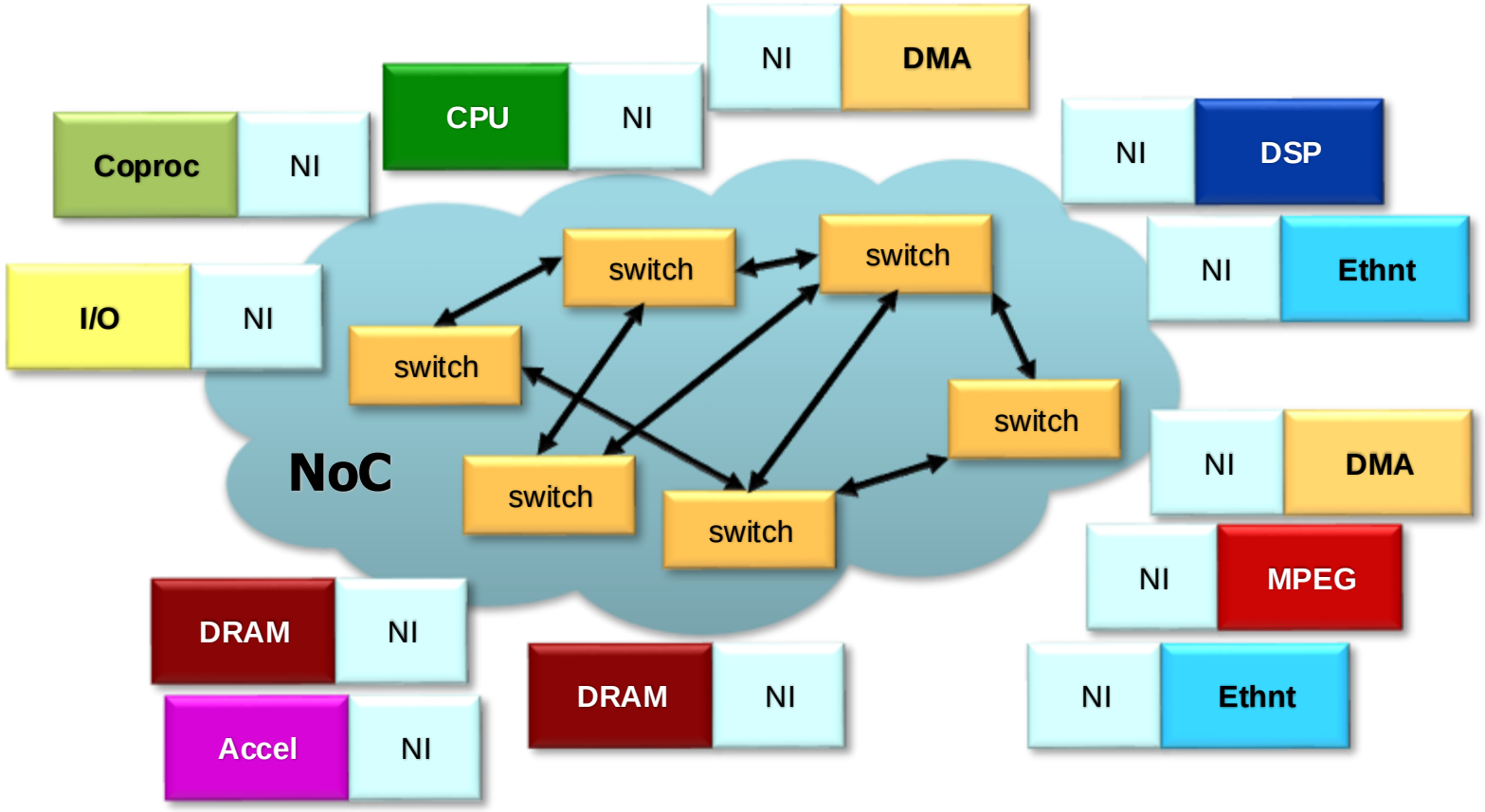






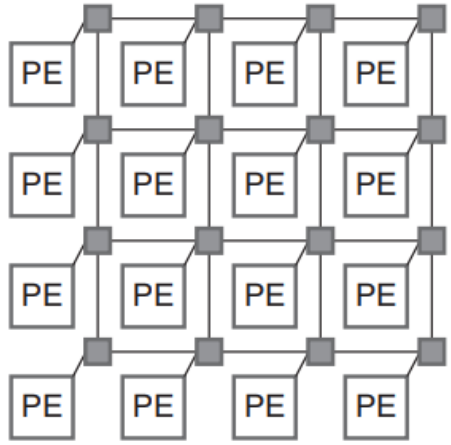
Network on Chip: A paradigm Shift in VLSI

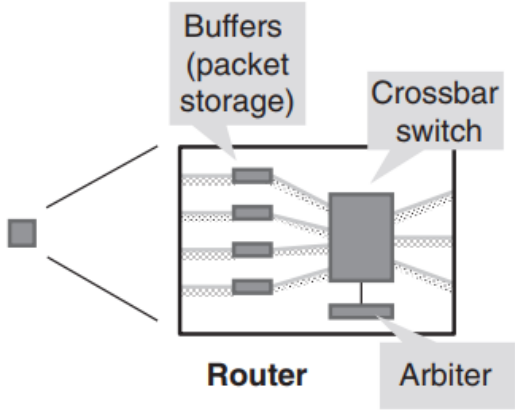
- Buses replaced with Networked architectures
  - Better electrical properties
  - Higher bandwidth
  - Energy efficiency
  - Scalable



Network-on-Chip (NoCs)

- NoCs borrow ideas and concepts from computer networks → apply them to the embedded SoC domain.
- A Network-on-chip (NoC) is a **packet switched** on-chip communication network designed using a layered methodology Packed data into a packet
  - "routes packets, not wires"
- NoCs use packets to route data from the source to the destination processing element (PE) via a network fabric that consists of
  - Network interfaces/adapters (NI)
  - Switches (routers)
  - Interconnect links (wires)

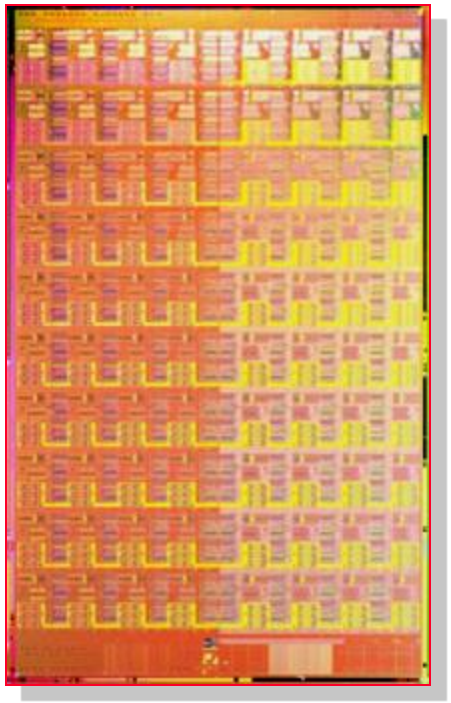


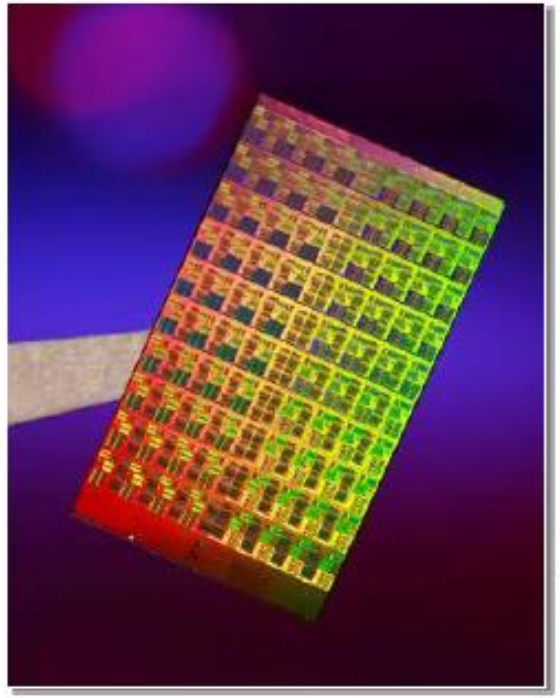


An example

Teraflops Research Chip

100 Million Transistors • 80 Tiles • 275mm^2





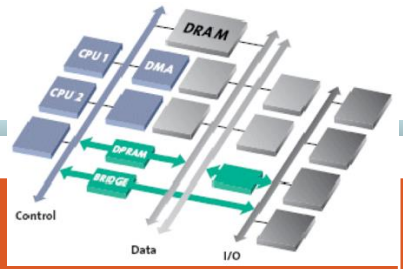
First tera-scale programmable silicon

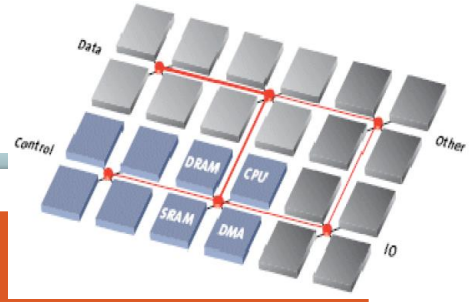
- Teraflops performance
- Tile design approach
- On-die mesh network
- Novel clocking
- Power-aware capability
- Supports 3D-memory

Not designed for IA or product

Source: ossum, Intel @ MPSoC'07

NoC vs. Bus





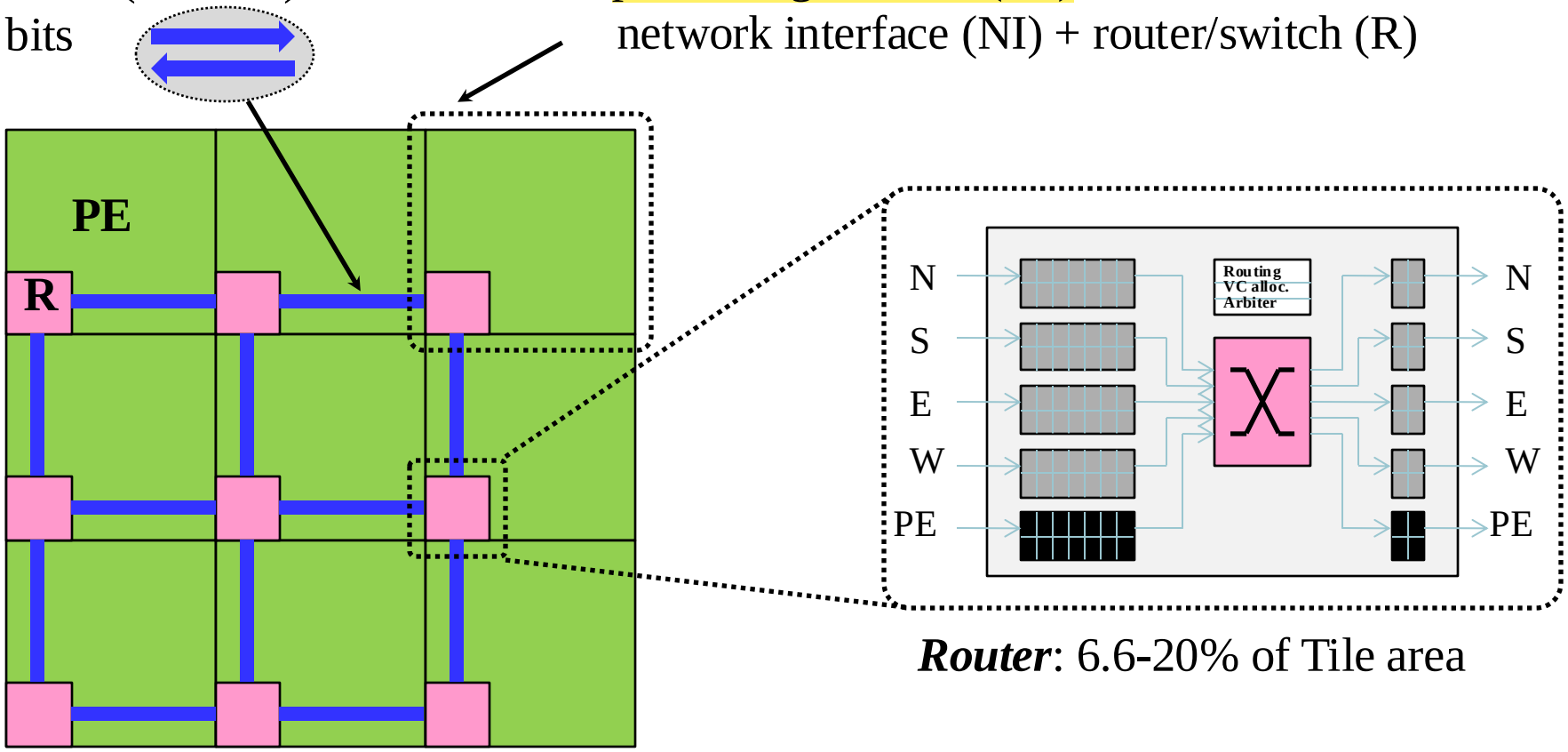
Bus	NoC
Longer connections → higher parasitic capacitance	Performance does not downgrade with network scaling
Arbitration grows and becomes a bottleneck	Arbitration and routing are distributed
Bandwidth is limited and shared by all cores	Aggregated bandwidth scales with network size
Latency is wire-speed once arbitration granted control	Multiple hops increase latency
Well-known and simple concepts	Further study needed

source: Dr. Konstantinos Tatas

NoC: A closer look

Physical link (channel)
e.g., 64 bits

Tile = processing element (PE)/Node +
network interface (NI) + router/switch (R)

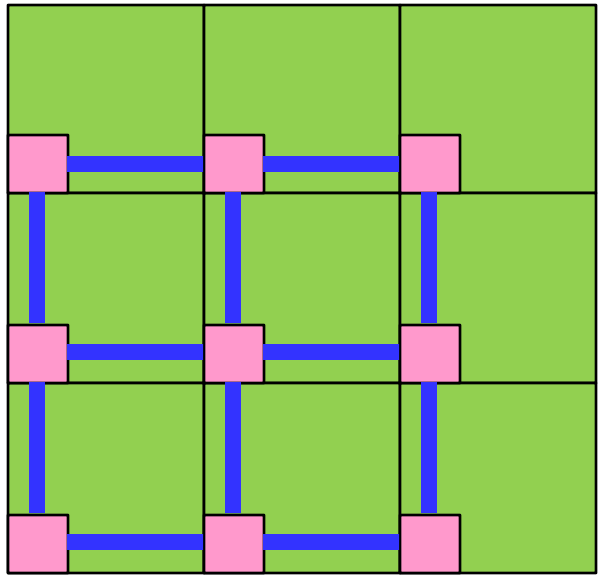


Router: 6.6-20% of Tile area

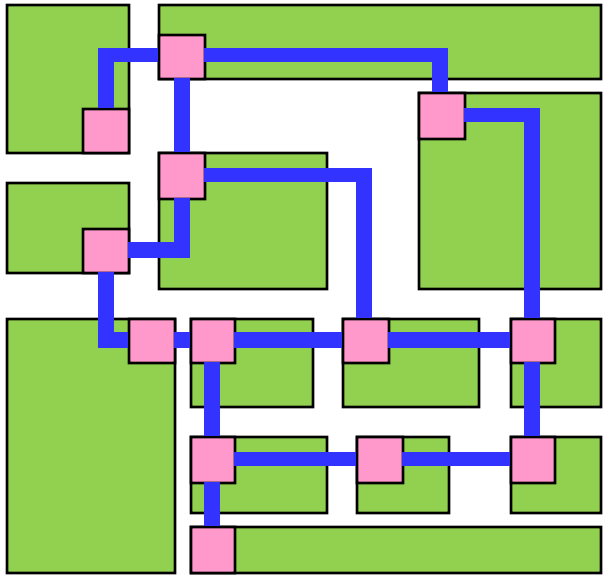
3x3 homogeneous NoC

source: Cristinel Ababei@ North Dakota State University

Homogeneous vs. Heterogeneous



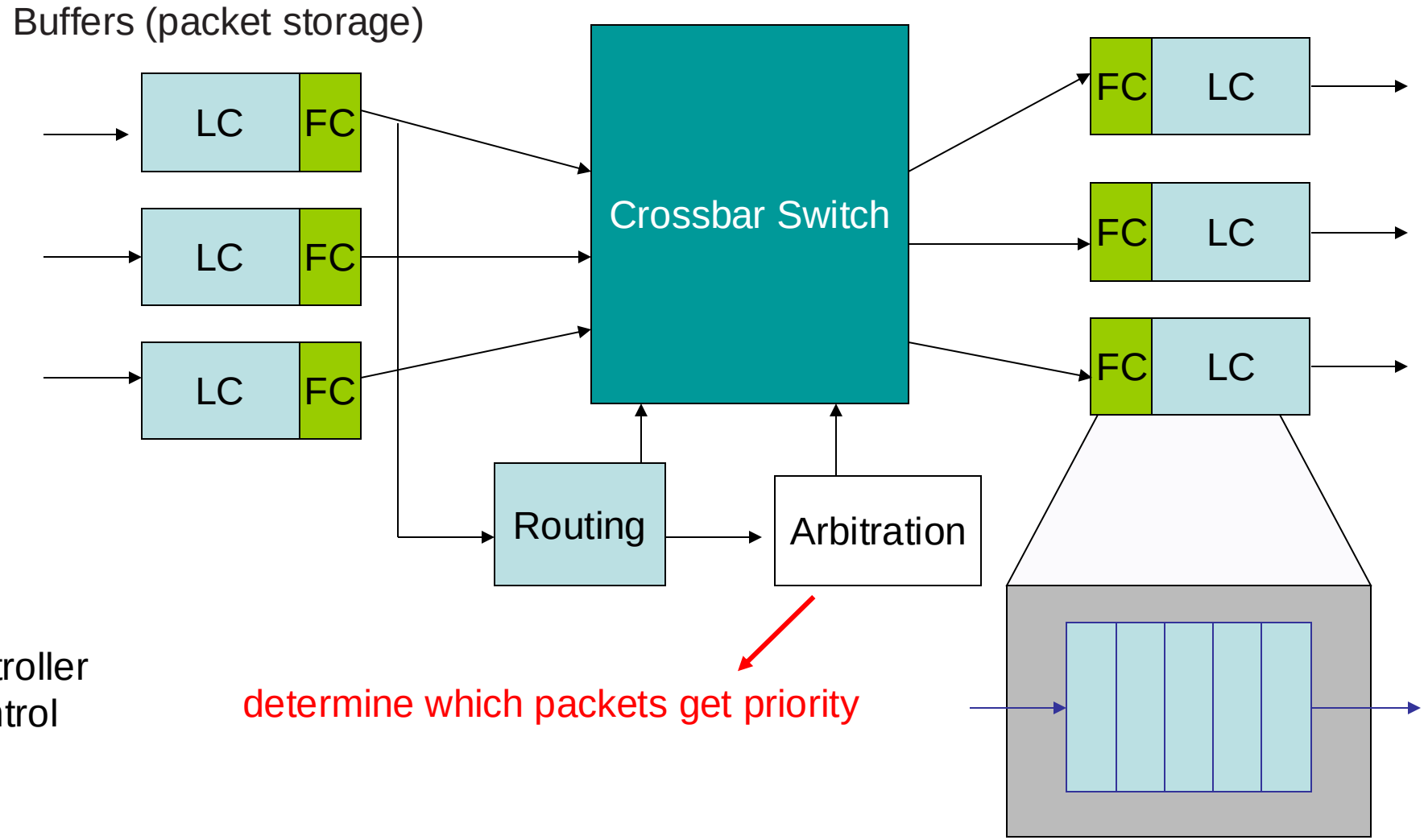
- Homogenous:
  - Each tile is a simple processor
  - Tile replication (scalability, predictability)
  - Less performance
  - Low network resource utilization



- Heterogeneous:
  - IPs can be: General purpose/DSP processor, Memory, FPGA, IO core
  - Better fit to application domain
  - Most modern systems are heterogeneous
  - Topology synthesis: more difficult
  - Needs specialized routing

source: Cristinel Ababei@ North Dakota State University

A typical SoC Router

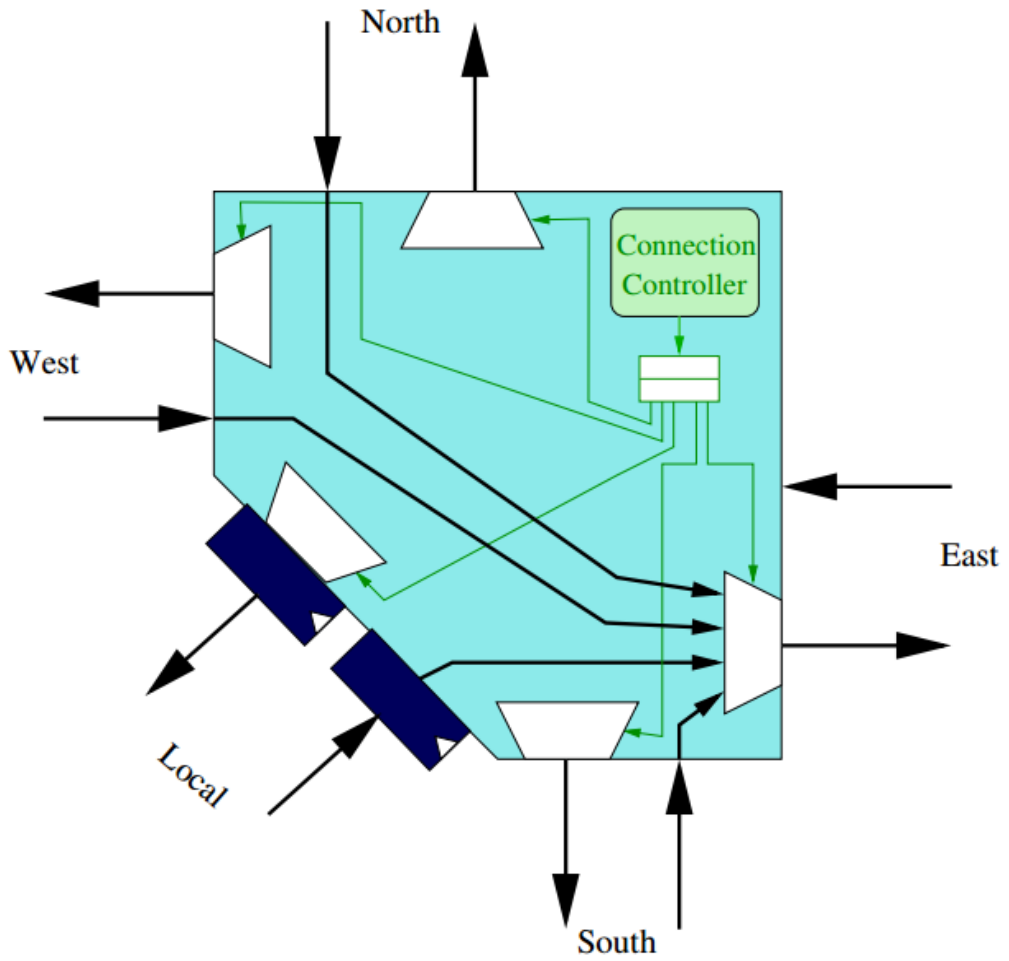


LC: Link Controller
FC: Flow Control

determine which packets get priority

source: Jeremy Chan

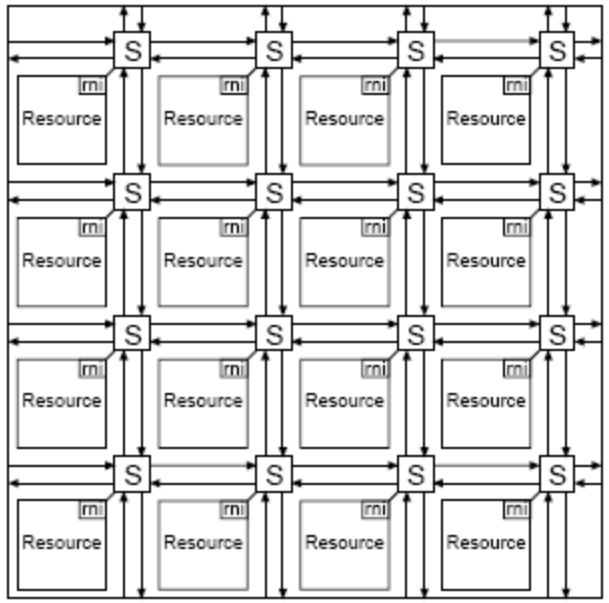
NoC Router – Crossbar Switch

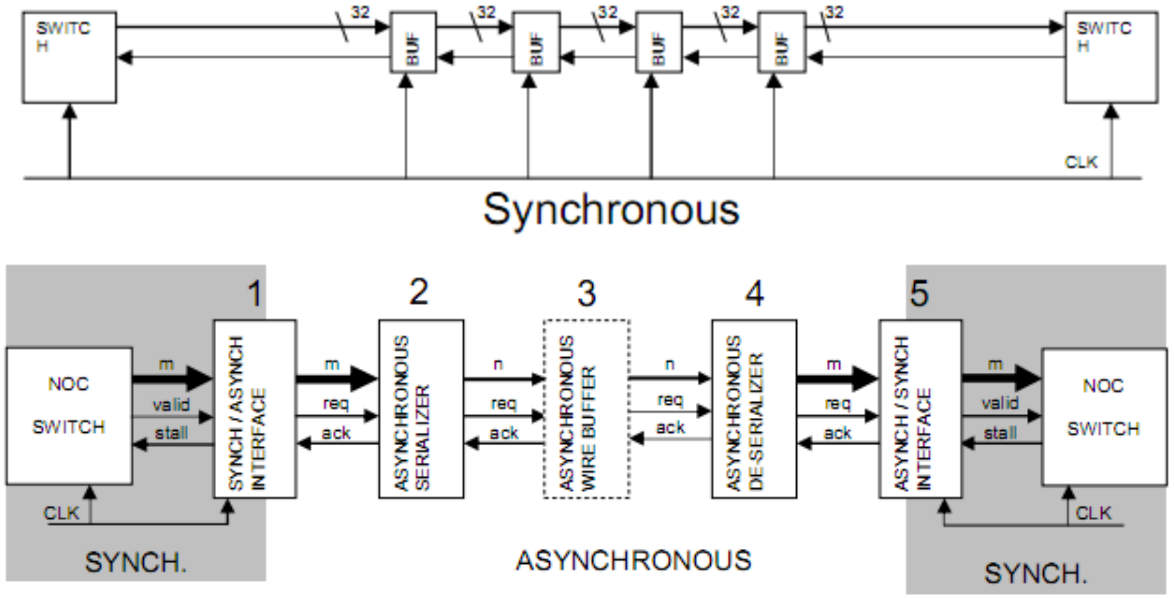


source: Konstantinos Tatas

NoC Links

- Connects two routers in both directions on a number of wires (e.g., 32 bits)
- In addition, wires for control are part of the link too
- Can be pipelined (include handshaking for asynchronous)





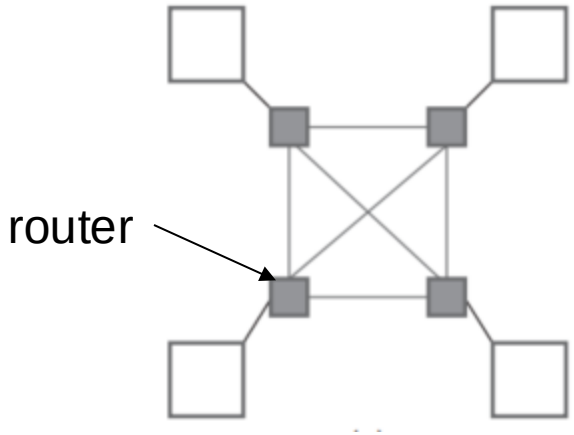
source: Cristinel Ababei@ North Dakota State University

NoC Topologies (0)

- Regular/irregular
- Direct/indirect
  - each node has a direct point-to-point link to a subset of other nodes in the system, called neighboring nodes

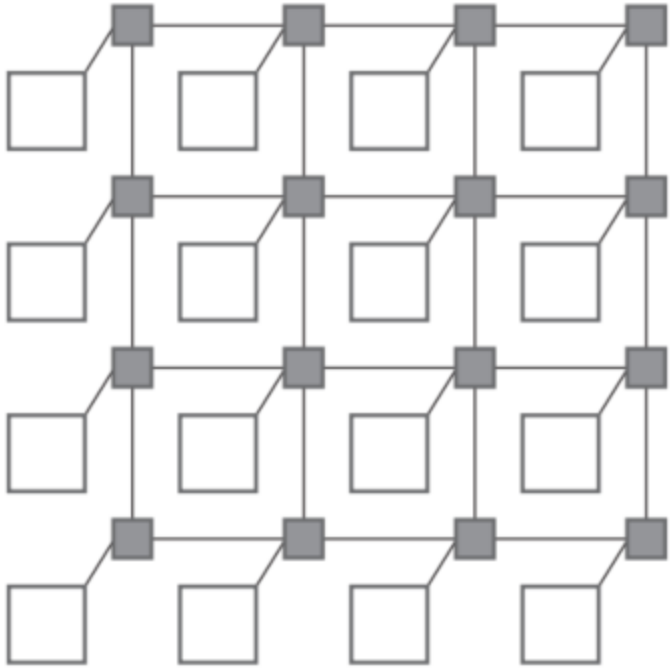
NoC Topology (1)

- **Direct Networks**
  - Each node has direct point-to-point link to a subset of other nodes in the system called neighboring nodes
  - Nodes consist of computational blocks and/or memories, as well as a NI block that acts as a router
  - As the number of nodes in the system increases, the total available communication bandwidth also increases
  - Fundamental trade-off is between connectivity and cost



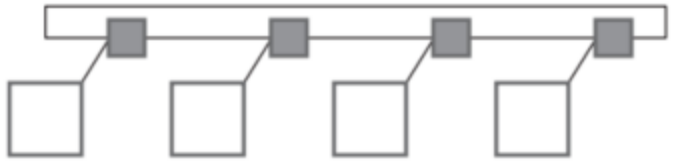
NoC Topology (2)

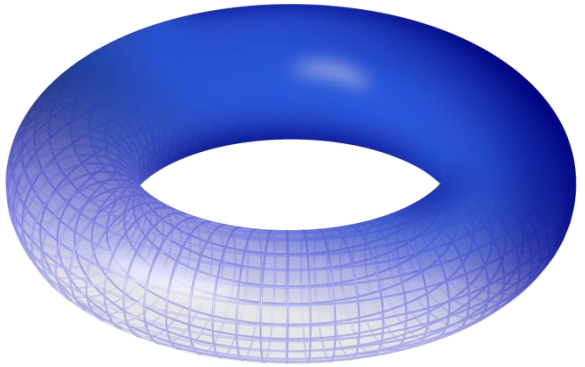
- Most direct network topologies have an orthogonal implementation, where nodes can be arranged in an *n* dimensional orthogonal space.
  - Routing for such networks is fairly simple
- 2D mesh is most popular topology
  - All links have the same length Reusing same links
    - Eases physical design
  - Area grows linearly with the number of nodes
    - Must be designed in such a way as to avoid traffic accumulating in the center of the mesh



NoC Topology (3)

- Torus topology, also called a k-ary n-cube, is an ndimensional grid with k nodes in each dimension
  - *k*-ary 1-cube (1-D torus) is essentially a ring network with k nodes
    - Limited scalability as performance decreases when more nodes





  - *k*-ary 2-cube (i.e., 2-D torus) topology is similar to a regular mesh
    - Except that nodes at the edges are connected to switches at the opposite edge via wrap-around channels
    - Long end-around connections can, however, lead to excessive delays

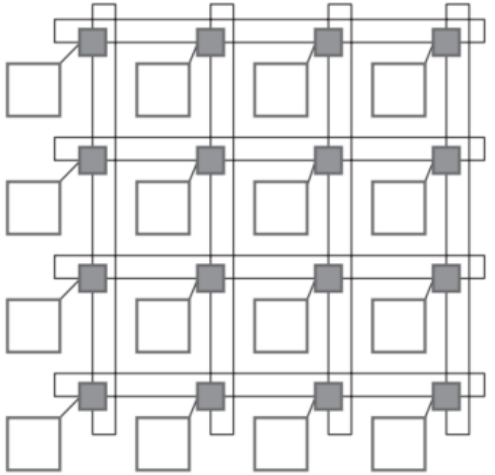
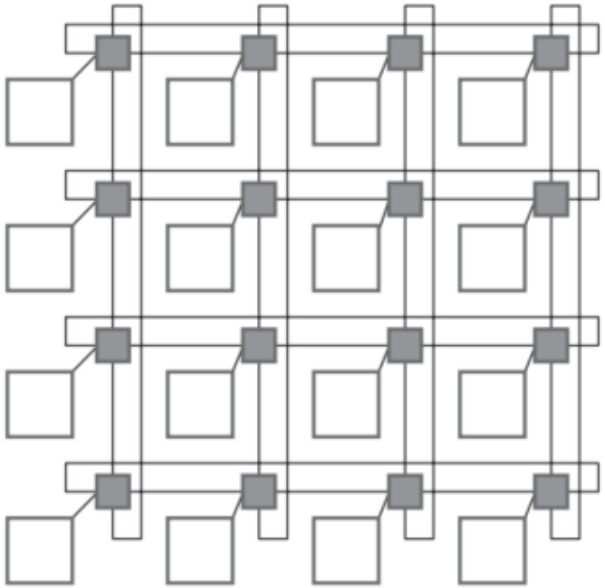


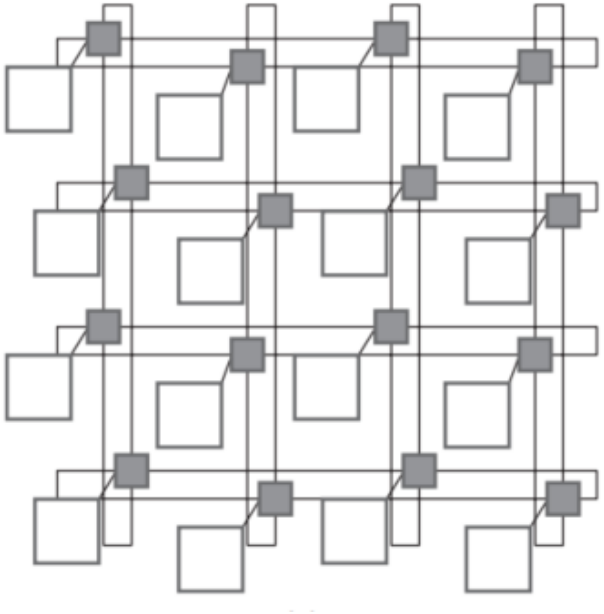
Image: Wikipedia

NoC Topology (4)

- Folding torus topology overcomes the long link limitation of a 2-D torus
  - Links have the same size
- Meshes and tori can be extended by adding bypass links to increase performance at the cost of higher area



2-D Torus



Folding Torus

NoC Topology (5)

- Octagon
  - Well-established direct topology found in NoCs.
  - ring of 8 nodes connected by 12 bi-directional links.
  - links provide two-hop communication between any pair of nodes in the ring
  - In case a platform consists of more than eight nodes, the octagon is extended to multidimensional space

Maximum

more scalable

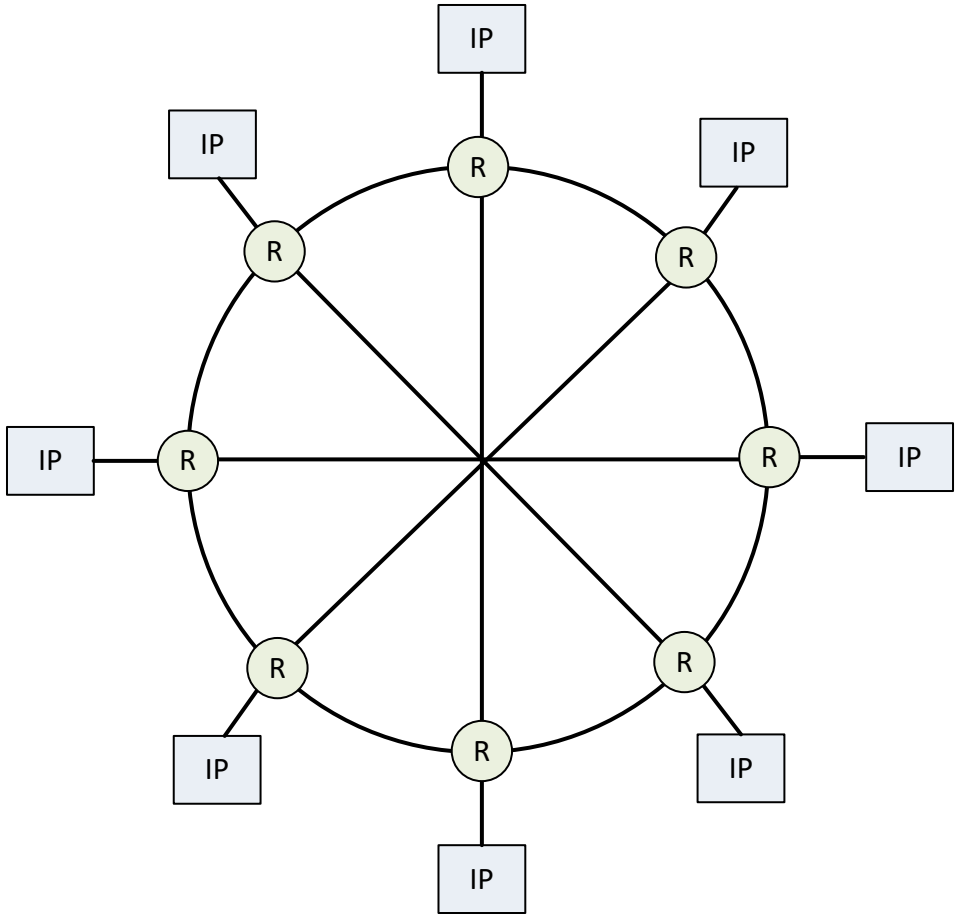


Image: Konstantinos Tatas

Polygon

- widely accepted topology
- packets travel in a loop from one router to the next. Direct traffic between IPs
- We can add chords to the circle
- if chords are inserted only between opposite routers, the topology is called a spidergon.

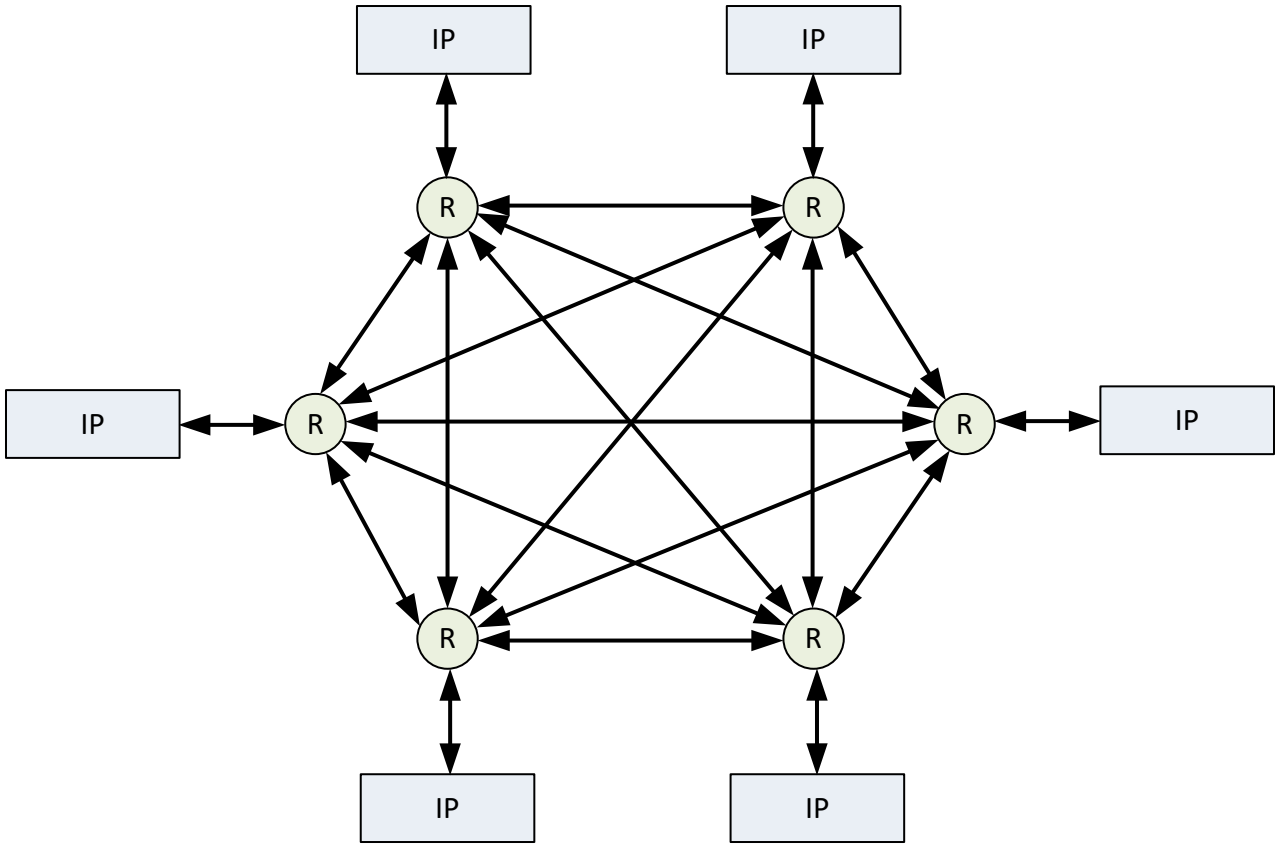
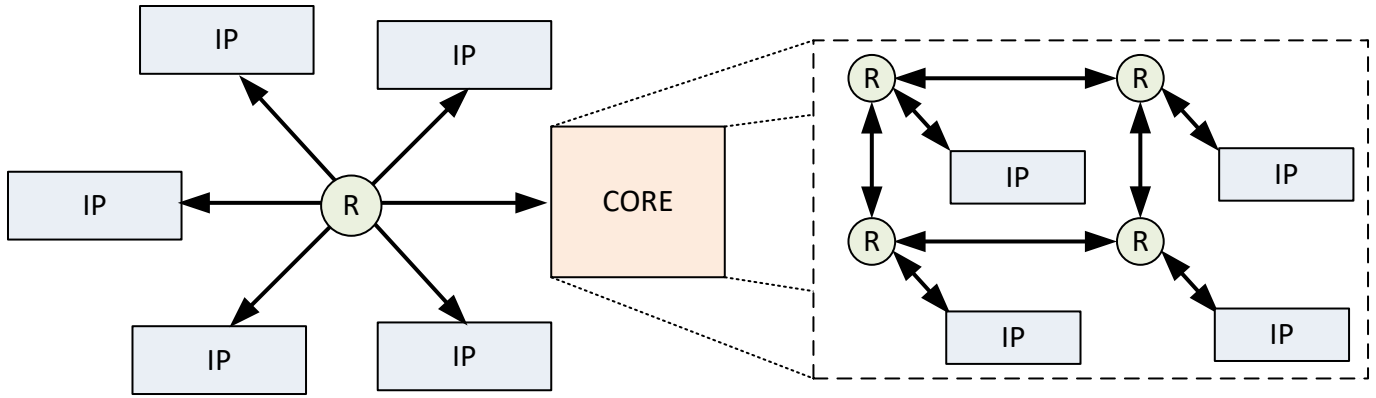


Image: Konstantinos Tatas

Star



- Central router in the middle of the star
- Computational resources, or subnetworks, in the spikes of the star.
- The capacity requirements of the central router are quite large
- significant possibility of congestion in the middle of the star

Image: Konstantinos Tatas

NoC Topology (6)

- **Indirect network topologies**
  - each node is connected to an external switch, and switches have point-to-point links to other switches
    - The NI associated with each node connects to a port of a switch. Switches do not perform any information processing, and correspondingly nodes do not perform any packet switching.
  - communication involves climbing up and down some part of the tree
  - the root and its neighbors have higher traffic.
  - Increasing the number of links near the root of the tree essentially allocates more bandwidth on the channels that have higher traffic.

Fat Tree

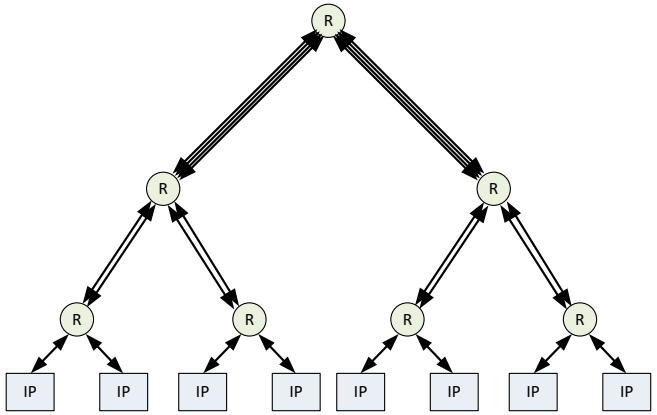
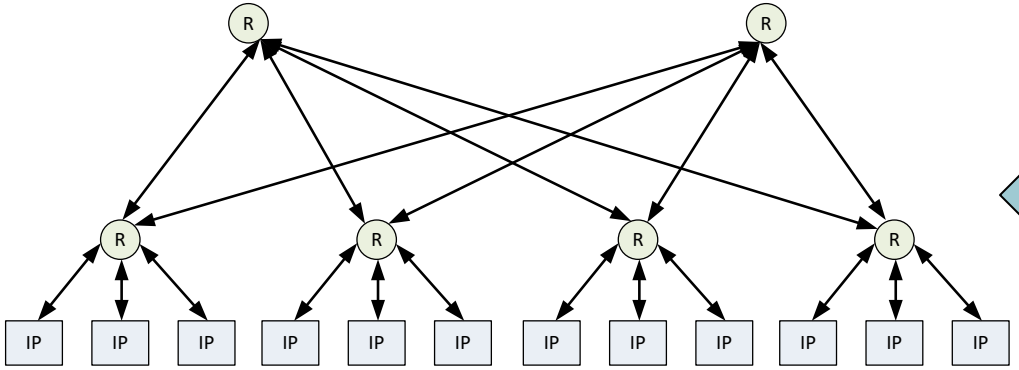


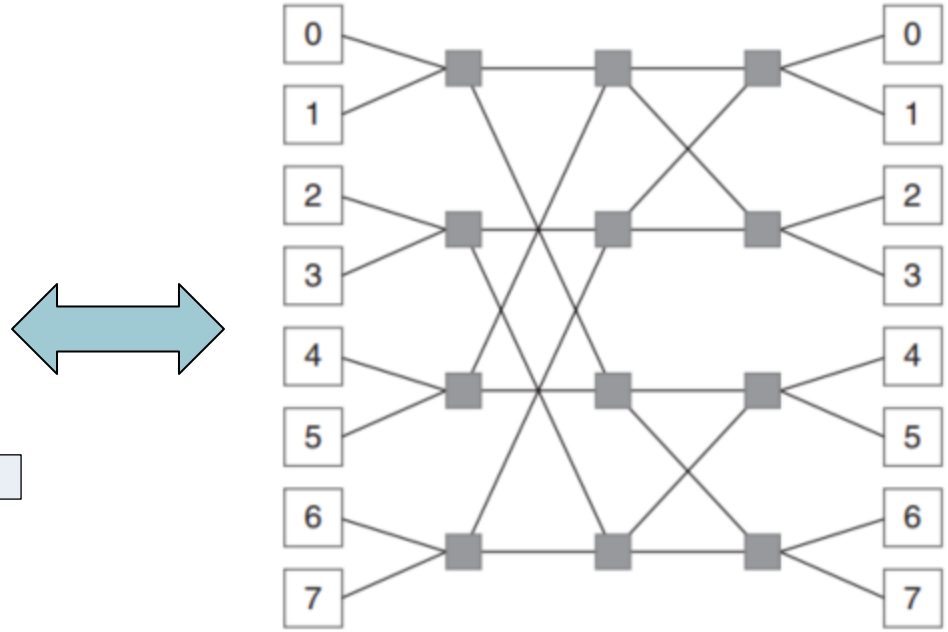
Image: Konstantinos Tatas

NoC Topology (7)

- Butterfly network
  - a *k*-ary, *n*-fly butterfly network consists of k^n nodes, and *n* stages of $k^{n-1}k$ x *k* crossbar switches
  - multi-stage network, which implies that information may be temporarily blocked or dropped in the network if contention occurs



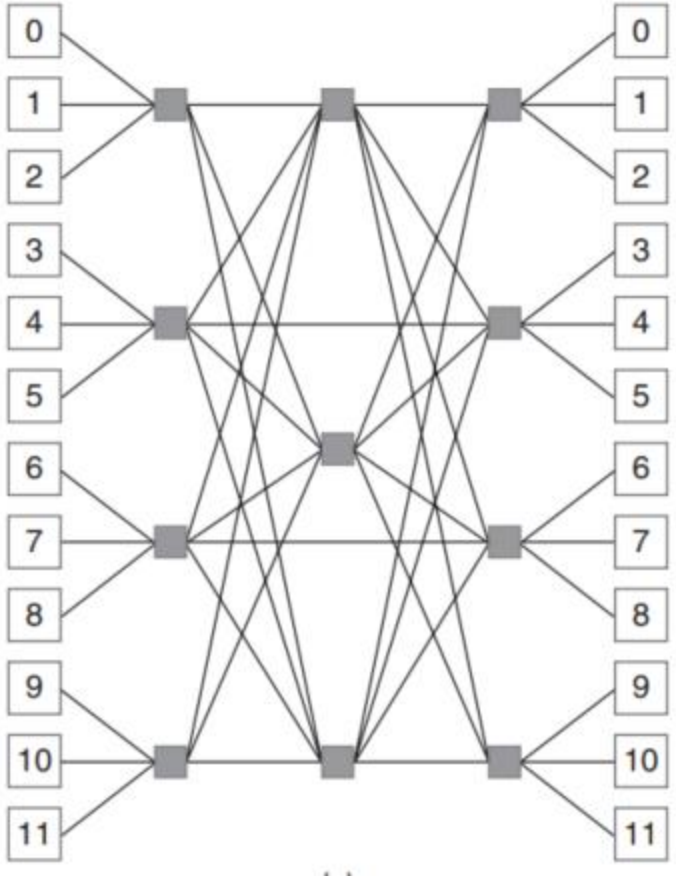
Butterfly Fat Tree



source destination

NoC Topology (8)

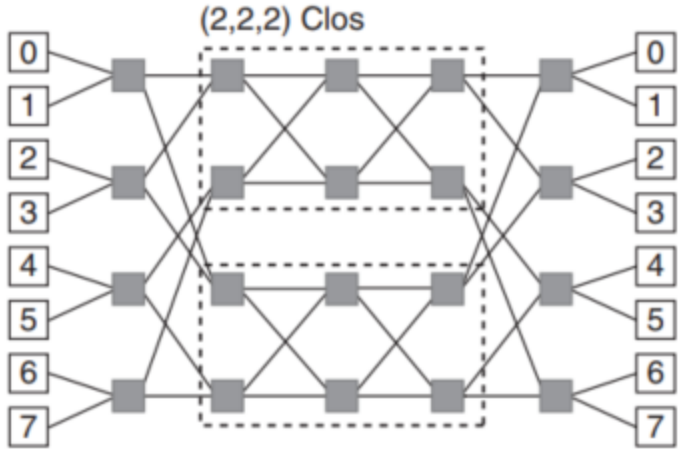
- Clos. Network
  - three-stage network in which each stage is made up of a number of crossbar switches. A symmetric Clos is characterized by a triple (*m, n, r*), where *m* is the number of middle-stage switches, *n* is the number of input/output nodes on each input/output switch, and *r* is the number of input and output switches.



(3, 3, 4) clos network

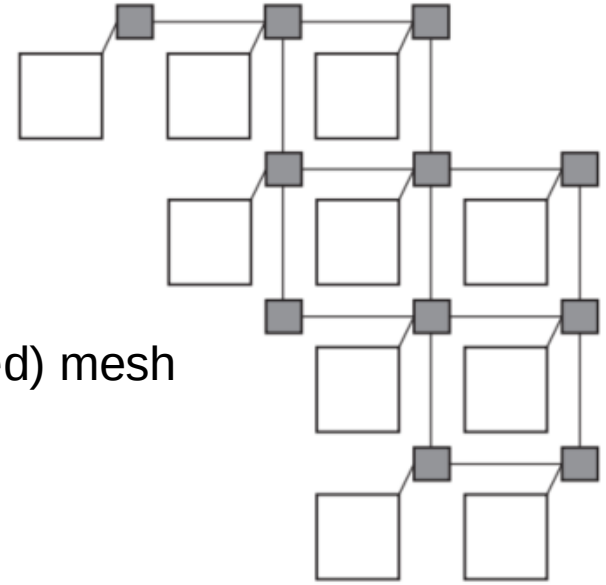
NoC Topology (9)

- Benes network
  - Also a type of Clos network
  - (2, 2, 4) rearrangeable Clos network topology constructed using two (2, 2, 2) Clos networks with 4x4 middle switches.
  - paths may have to be rearranged to provide a connection, requiring an appropriate controller.

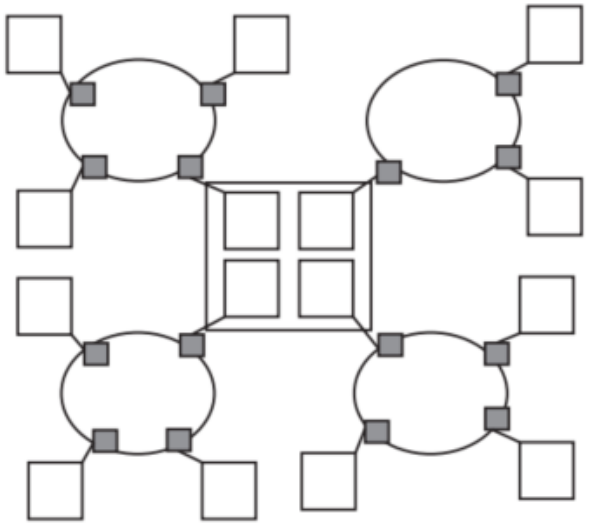


NoC Topology (10)

- Irregular Networks or ad hoc network
  - a mix of shared bus, direct, and indirect network topologies.
  - increase available bandwidth as compared to traditional shared buses, and reduce the distance between nodes as compared to direct and indirect networks.
  - are typically customized for an application

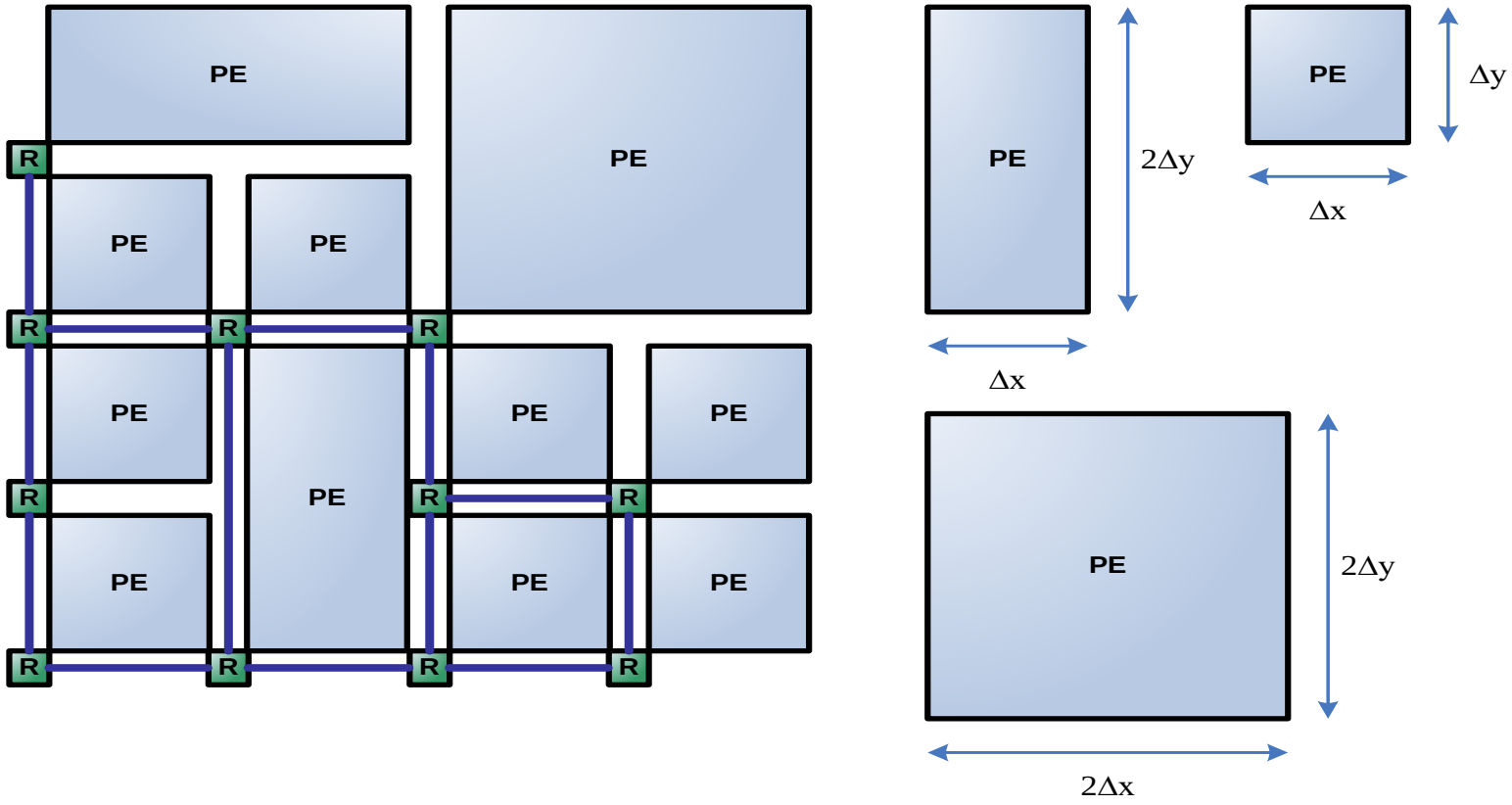


optimized (reduced) mesh



cluster-based hybrid (mesh ring) topology

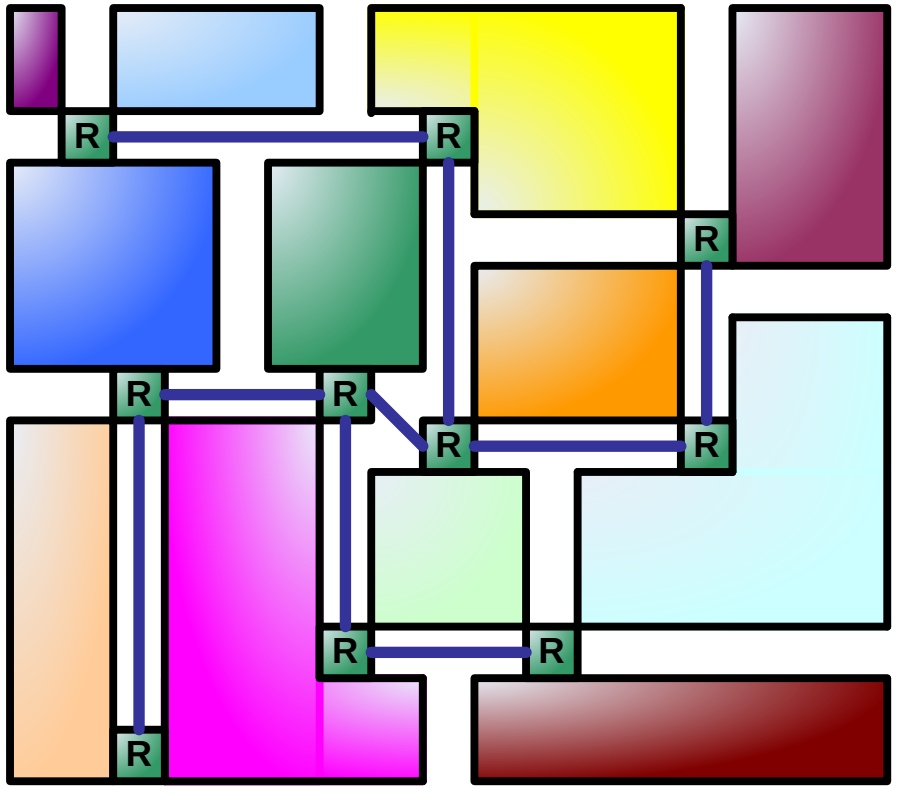
Example 1: Partially Irregular 2D-Mesh Topology



- Contains oversized rectangularly shaped PEs.

source: *Ben Abdallah, Abderazek*

Example 2: Irregular Mesh



- This kind of chip does not limit the shape of the PEs or the placement of the routers. It may be considered a "custom" NoC

source: *Ben Abdallah, Abderazek*

How to Select a Topology for SoC ?

- Application decides the topology type
- If PEs = few tens

Star, Mesh topologies are recommended

- If PEs = 100 or more

Hierarchical Star, Mesh are recommended

- Some topologies are better for certain designs than others

A quick pulse

What to expect from the above discussions…

- Be able to understand differences between direct and indirect networks
- Be able to identify the network topology and analyze its benefits/disadvantage
- Understand how one topology evolves from one architecture to another

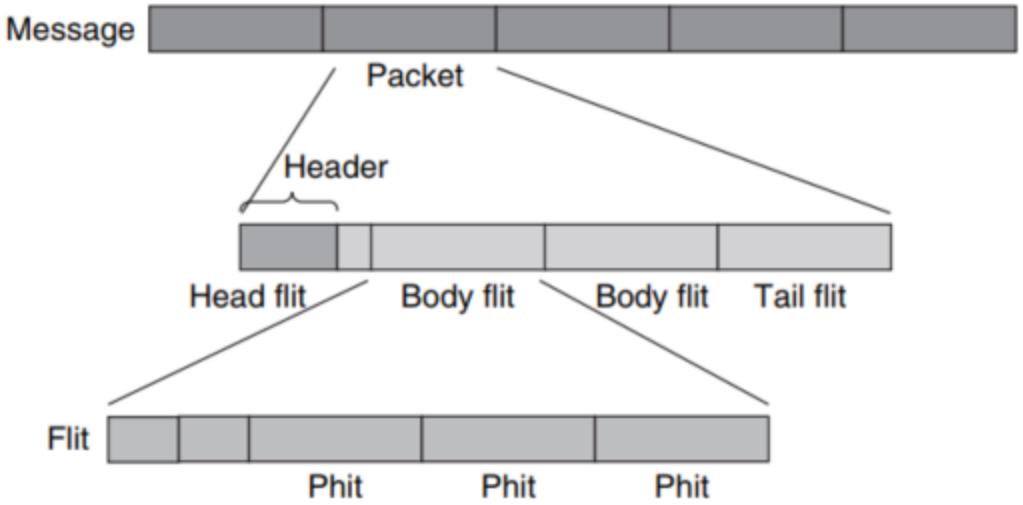
Switching & Routing

- **Switching** establishes the type of connection between source and destination. It is tightly coupled to **routing**. **Can be seen as a flow control mechanism as a problem of resource allocation.**

- **Routing** is the route/path (a sequence of channels) of streets from source to destination. “The routing method steers the car”. Routing determines the path followed by a message through the network to its final destination.

Switching

- Determine how data flows through routers in the network
- Define granularity of data transfer and applied switching technique
  - Flit (flow control digit) is unit of switching, a synchronization unit between routers
  - Phit (physical control digit) is a unit of data that is transferred on a link in a single cycle
  - The size of a phit is typically the width, in bits, of the communication link
  - Typically, phit size = flit size
- Two main modes
  - circuit (or path) switching
  - packet switching



Example:

- Phits: 1 bit to 64 bits
- Flits: 16 bits to 512 bits
- Packets: 128 bits to 1024 bits

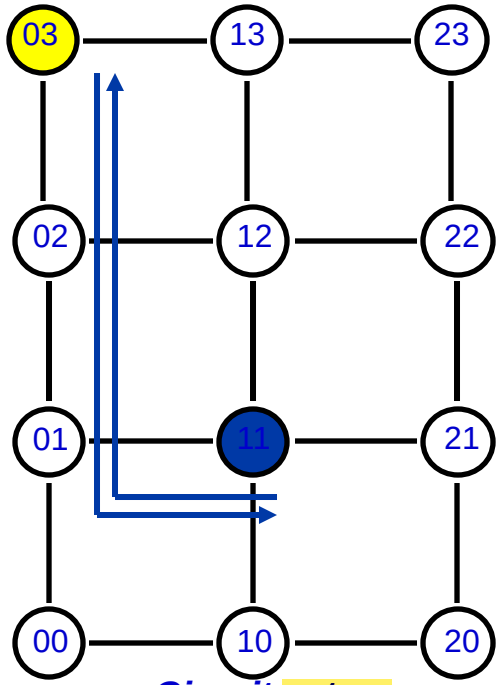
Size: Phit, Flit, Packet

- There are no fixed rules for the size of phits, flits and packets
  - Message: arbitrarily long
  - Packets: restricted maximum length
- Typical values
  - Phits: 1 bit to 64 bits
  - Flits: 16 bits to 512 bits
  - Packets: 128 bits to 1024 bits

Circuit Switching

- Physical path between the source and the destination is reserved prior to the transmission of data
  - routing information set up during initialization
  - It is a form of **bufferless** flow control Does not need to record something during the process
  - the messages from the sender are sent in their entirety to the receiver once a path (**circuit**) is reserved.
- Message header flit traverses the network from the source to the destination, reserving links along the way
  testing the way only with few bits, to reserve the whole link
  - Advantage: low latency transfers, once path is reserved
  - Disadvantage: pure circuit switching does not scale well with NoC size
    - Several links are occupied for the duration of the transmitted data, even when no data is being transmitted
      - For instance in the setup and tear down phases

Circuit Switching



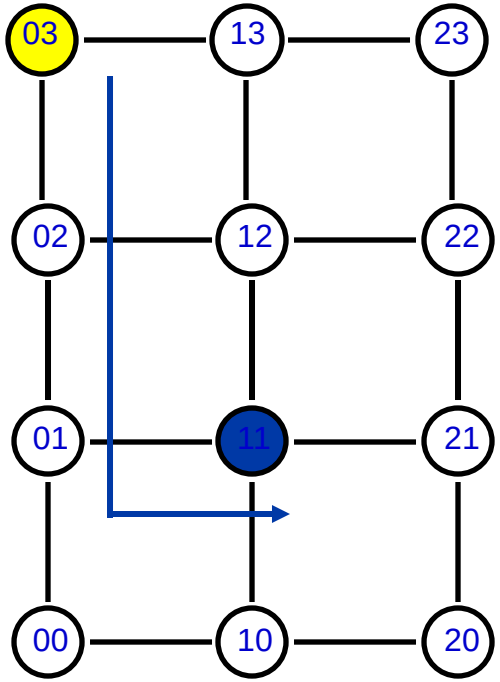
Circuit set-up

Two traversals – latency overhead

Waste of bandwidth

Request packet can be buffered

header find path and reserve



Circuit utilization

Third traversal – latency overhead

Contention-free transmission

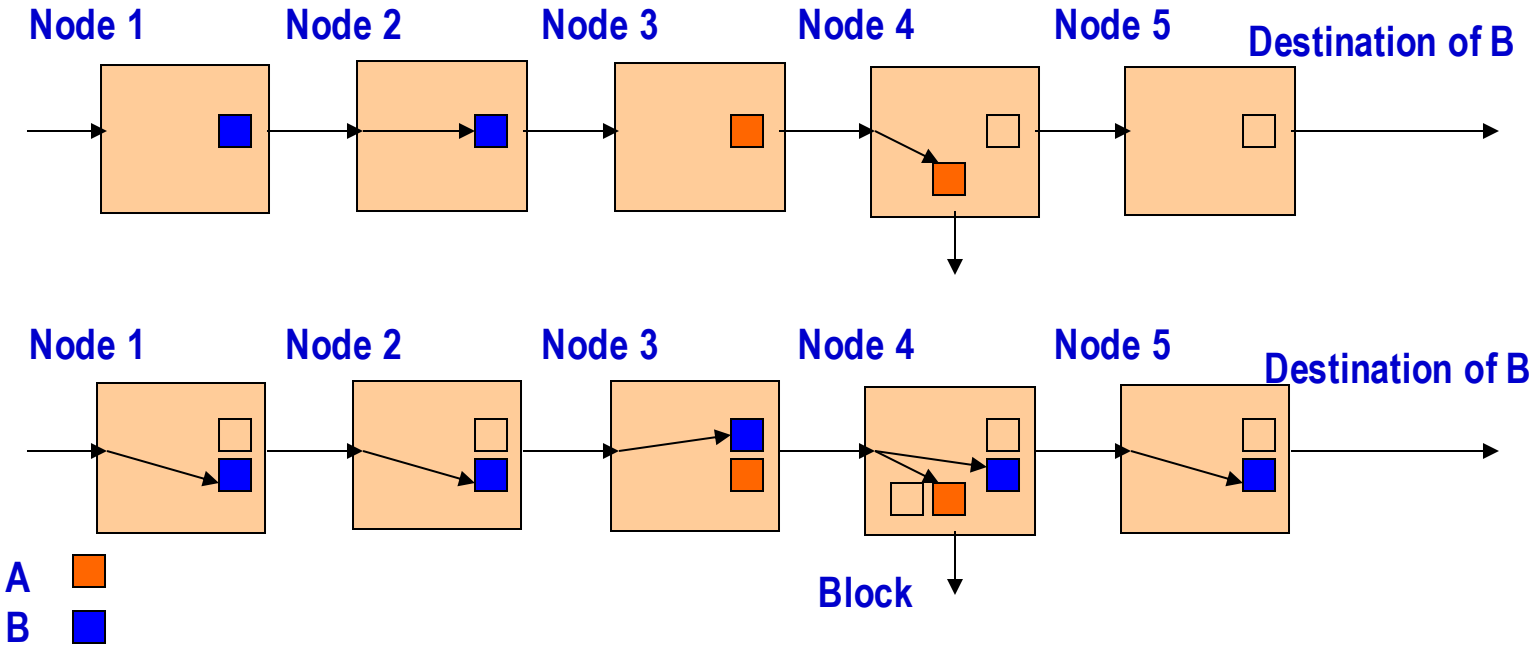
Poor resource utilization

utilize reserved path

- A request (R) propagates from source to destination, which is answered by an acknowledgement (A)
- Then data is sent to deallocate the channels/links

Virtual Circuit Switching

- Creates virtual circuits that are multiplexed on links
- Virtual-channel flow control decouples the allocation of channel state from channel bandwidth.
- Number of virtual links (or virtual channels (VCs)) that can be supported by a physical link depends on buffers allocated to link
- Allocating one buffer per virtual link
  - can be expensive due to the large number of shared buffers
- Allocate one buffer per physical link
  - uses time division multiplexing (TDM) to statically schedule usage
  - less expensive routers



Packet Switching

- It is a form of buffered flow control
- Packets are transmitted from source and make their way independently to receiver
  - possibly along different routes and with different delays
- Zero start up time, followed by a variable delay due to contention in routers along packet path

Multiple packets can arrive the same router at the same time

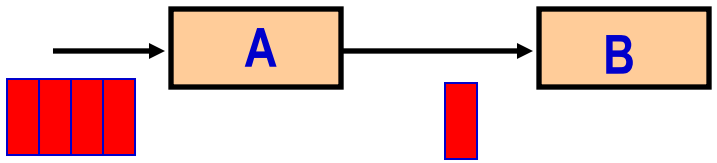
  - QoS guarantees are harder to make

Quality of Service (QoS)

- QoS refers to the level of commitment for packet delivery
  - refers to bounds on performance (bandwidth, delay, and jitter=packet delay variation)
- Two basic categories
  - Best effort (BE)
    - only correctness and completion of communication is guaranteed
    - usually packet switched
    - worst case times cannot be guaranteed
  - Guaranteed service (GS)
    - makes a tangible guarantee on performance, in addition to basic guarantees of correctness and completion for communication
    - usually (virtual) circuit switched

Three main packet switching scheme variants

- 1. Store and Forward (SAF) switching
  - packet is sent from one router to the next only if **the receiving router has buffer space for entire packet**
  - buffer size in the router is at least equal to the size of a packet
  - Disadvantage: excessive buffer requirements
- 2. Virtual Cut Through (VCT) switching
  - **forwards first flit of a packet as soon as space for the entire packet is available** in the next router
  - reduces router latency over SAF switching
  - same buffering requirements as SAF switching
- 3. Wormhole (WH) switching
  - flit is forwarded to receiving router if space exists for that flit



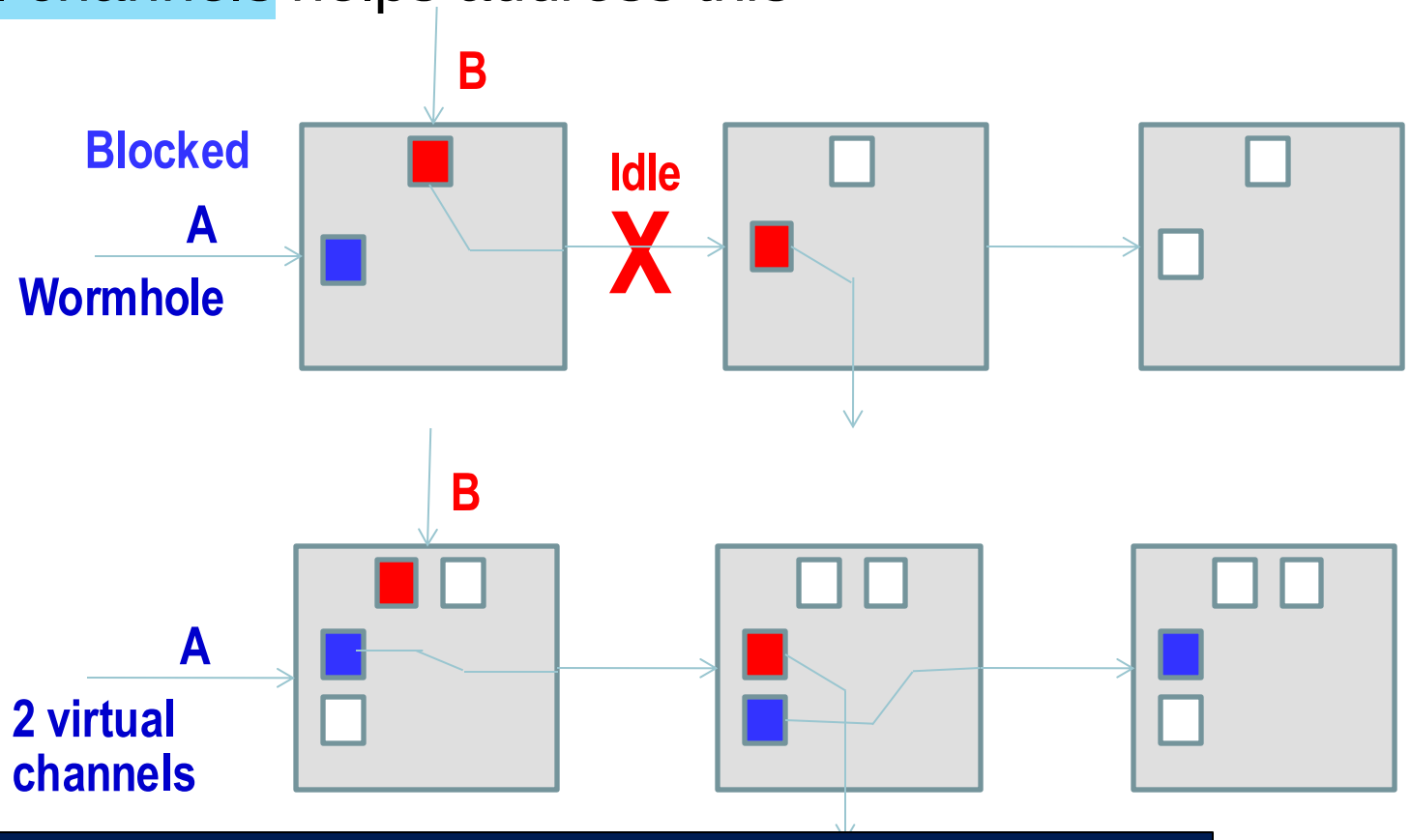
Pipelining on a flit (flow control unit) basis

(1) After A receives a flit of the packet, A asks B if B is ready to receive a flit
(2) B → A, ack
(3) A sends a flit to B.

flit size < packet size
Smaller data space is needed than store-and-forward

Wormhole Switching Issues

- Wormhole switching suffers from packet blocking problems
- An idle channel cannot be used because it is owned by a blocked packet…
  - Although another packet could use it!
- Using virtual channels helps address this



Costly as additional buffer is added to each router

Routing algorithms

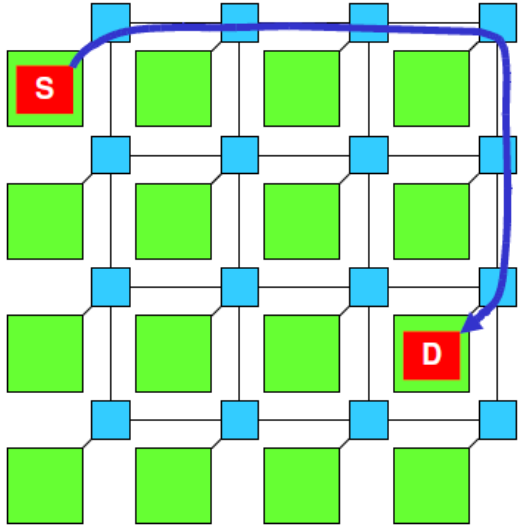
- **Routing** is the route/path (a sequence of channels) of streets from source to destination. "The routing method steers the car".
- Routing determines the path followed by a message through the network to its final destination.
- Responsible for correctly and efficiently routing packets or circuits from the source to the destination
  - Path selection between a source and a destination node in a particular topology
- Ensure load balancing
- Latency minimization
- Flexibility w.r.t. faults in the network
- Deadlock and livelock free solutions
- Routing schemes/techniques/algos can be classified/looked-at as:
  - Static or dynamic routing
  - Distributed or source routing
  - Minimal or non-minimal routing

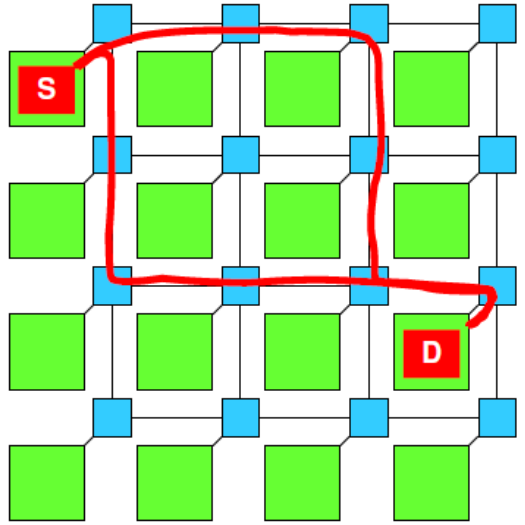
source: Cristinel Ababei@ North Dakota State University

Static/deterministic vs. Dynamic/adaptive Routing

paths are predetermined between a certain source and destination.

- **Static routing**: fixed paths are used to transfer data between a particular source and destination
  - does not take into account current state of the network
- advantages of static routing:
  - easy to implement, since very little additional router logic is required
  - in-order packet delivery if single path is used
- **Dynamic/adaptive routing**: routing decisions are made according to the current state of the network
  - considering factors such as availability and load on links
- path between source and destination may change over time
  - as traffic conditions and requirements of the application change
- more resources needed to monitor state of the network and dynamically change routing paths
- able to better distribute traffic in a network





source: Cristinel Ababei @ North Dakota State University

Distributed vs. Source Routing

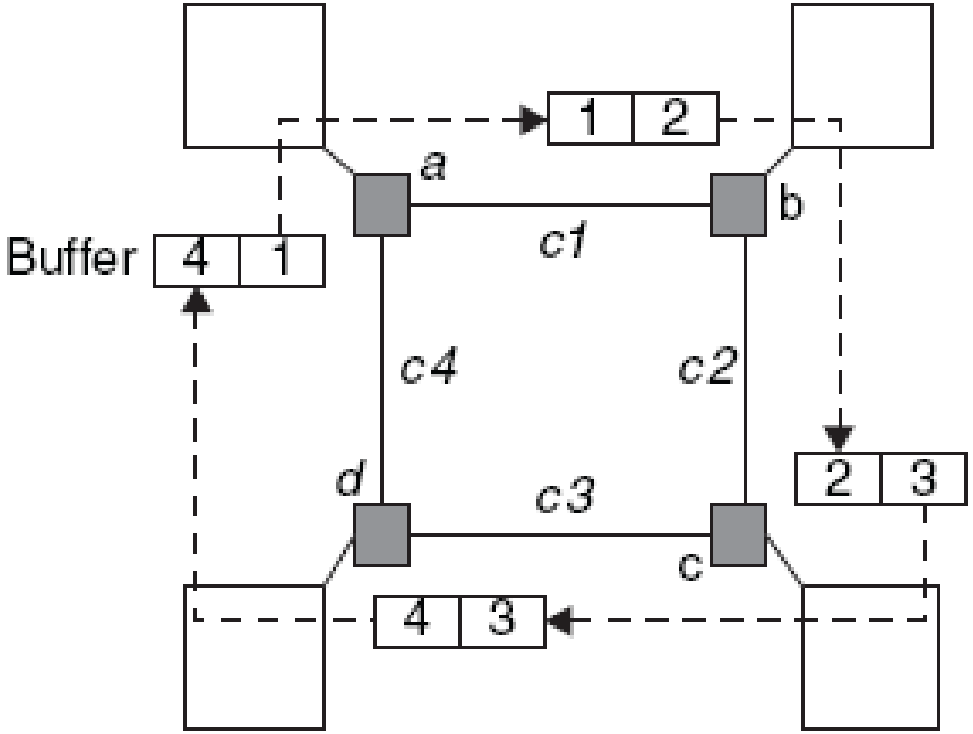
- Distributed routing: each packet carries the destination address
  - e.g., XY co-ordinates or number identifying destination node/router
  - routing decisions are made in each router by looking up the destination addresses in a routing table or by executing a hardware function
- Source routing: packet carries routing information
  - pre-computed routing tables are stored at a nodes' NI
  - routing information is looked up at the source NI and routing information is added to the header of the packet (increasing packet size)
  - when a packet arrives at a router, the routing information is extracted from the routing field in the packet header
  - does not require a destination address in a packet, any intermediate routing tables, or functions needed to calculate the route

Minimal and Non-minimal routing

- Minimal routing: length of the routing path from the source to the destination is the shortest possible length between the two nodes
  - e.g. in a mesh NoC topology (where each node can be identified by its XY co-ordinates in the grid) if source node is at (0, 0) and destination node is at (i, j), then the minimal path length is |i| + |j|
  - source does not start sending a packet if minimal path is not available
- Non-minimal routing: can use longer paths if a minimal path is not available
  - by allowing non-minimal paths, the number of alternative paths is increased, which can be useful for avoiding congestion
  - disadvantage: overhead of additional power consumption

Deadlock

- Occurs when a group of agents, usually packets, are unable to progress because they are waiting on one another to release resources (usually buffers and channels).



Each of routers are fully occupied,
as each packets are waiting for destinated router
to be released.

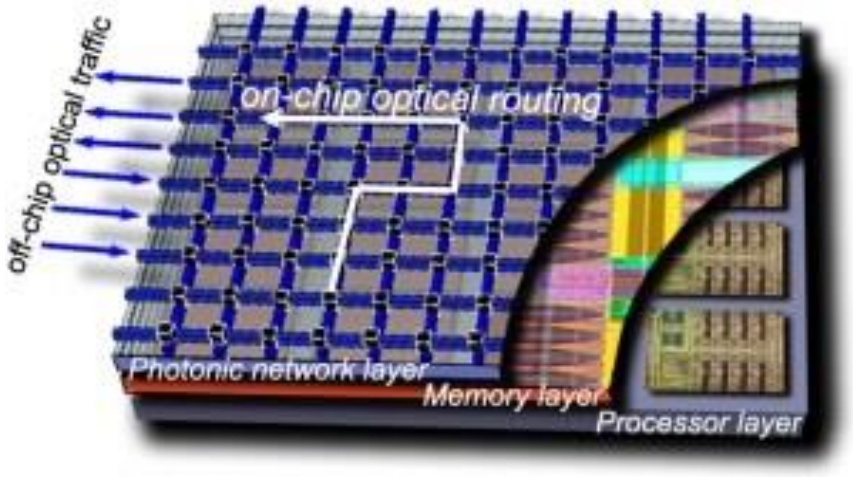
NoC Trends

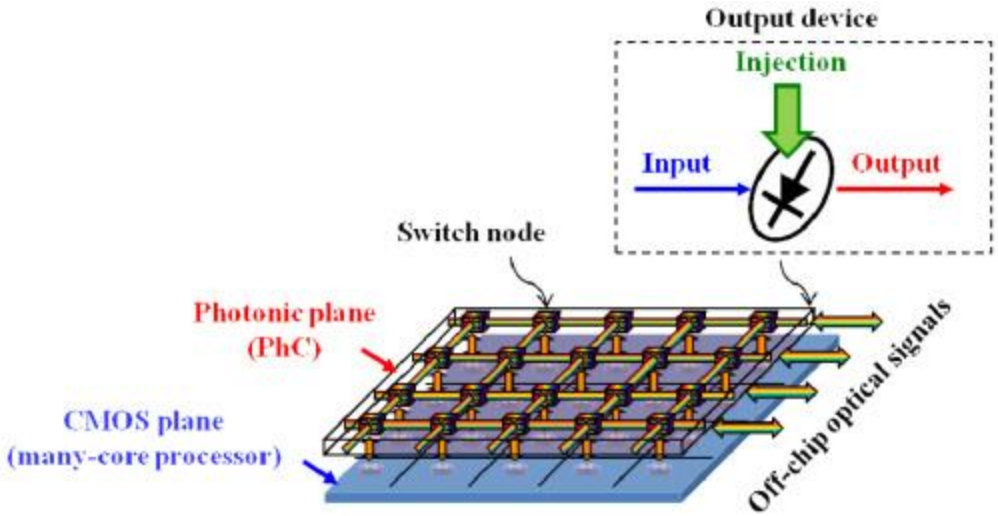
Costly : routing, buffers

- Hybrid interconnection structures
  - NoC and Bus based
  - Custom (application specific), heterogeneous topologies
- New interconnect paradigms
  - Optical, Wireless, Carbon nanotubes?
- 3D NoC
- Reconfigurability features
- GALS, DVFS, VFI

Optical/Photonic NoC

- Ultra-fast communication speed (light speed!)

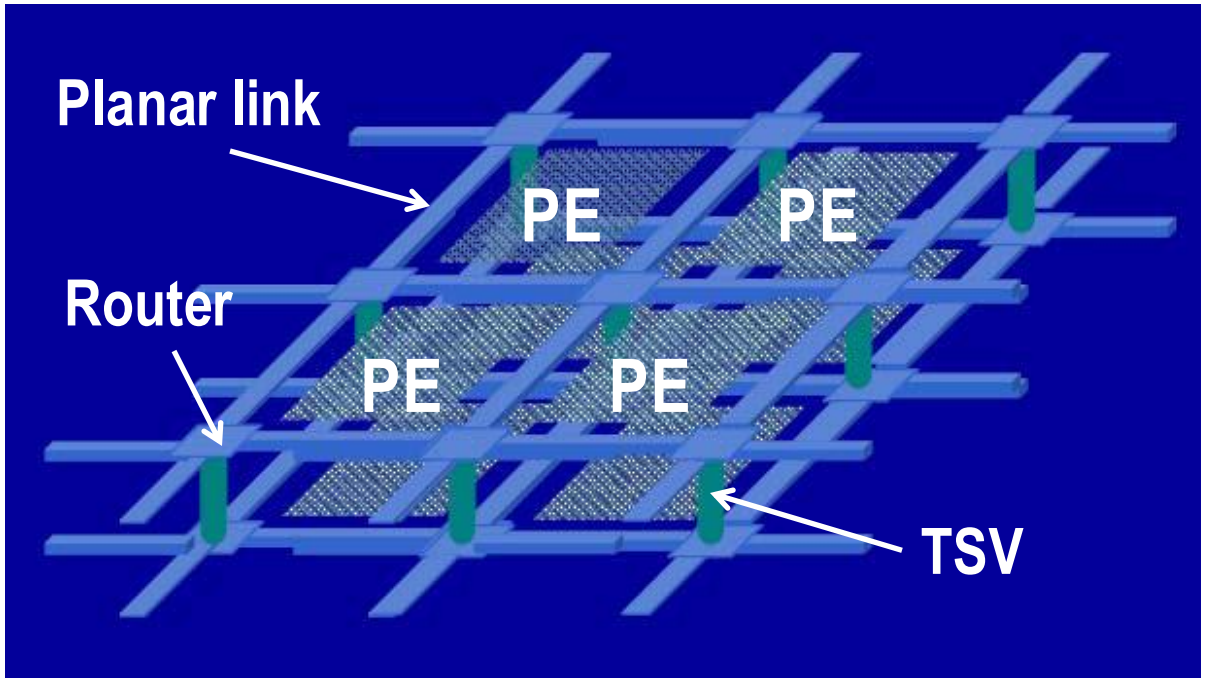




source: https://omnest.com/casestudy-photonic.php

3D NoC

- Shorter channel length
- Reduced average number of hops



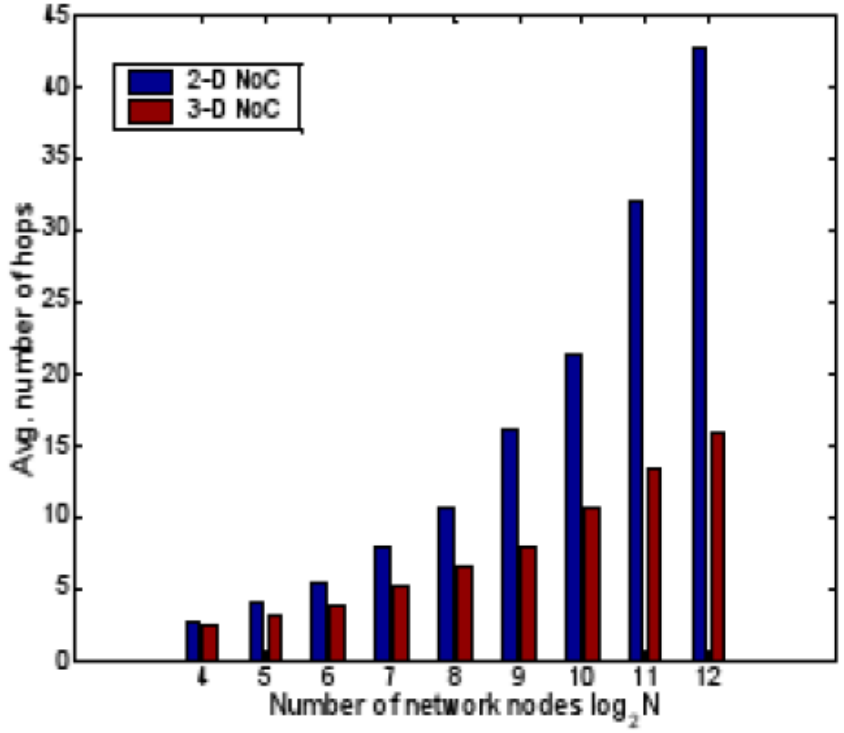
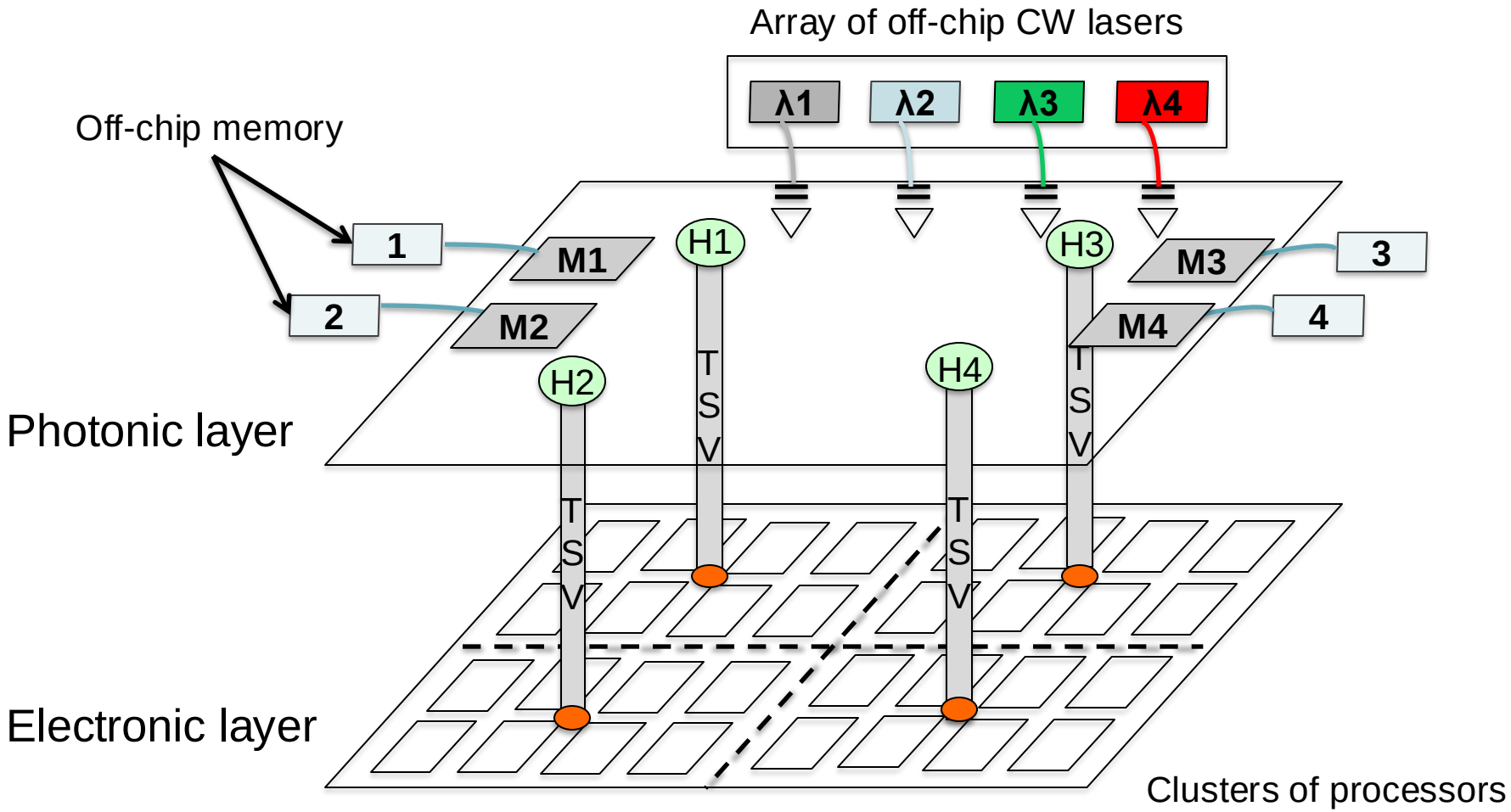


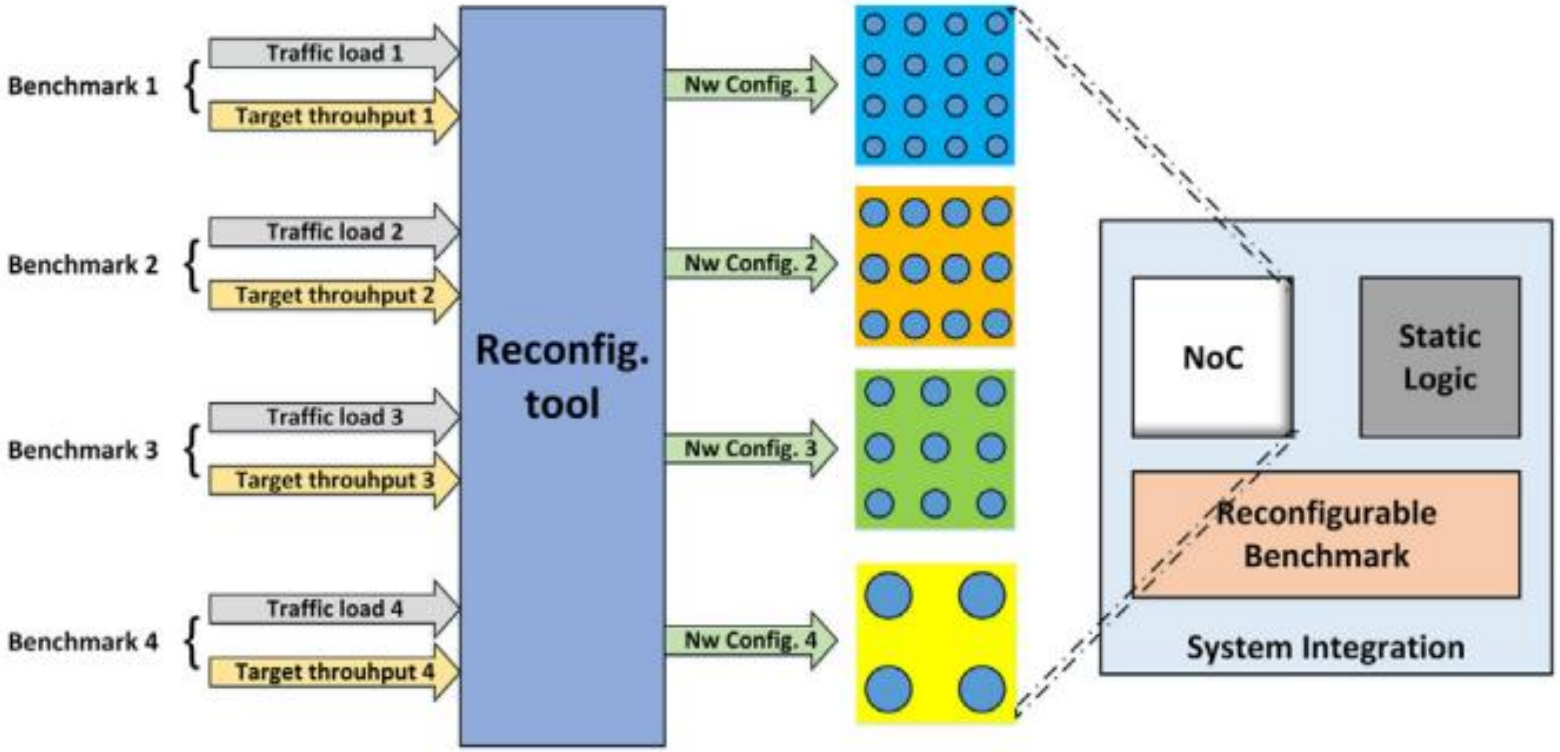
Image: Dr. Cristinel Ababei

3D+Photonic NoC



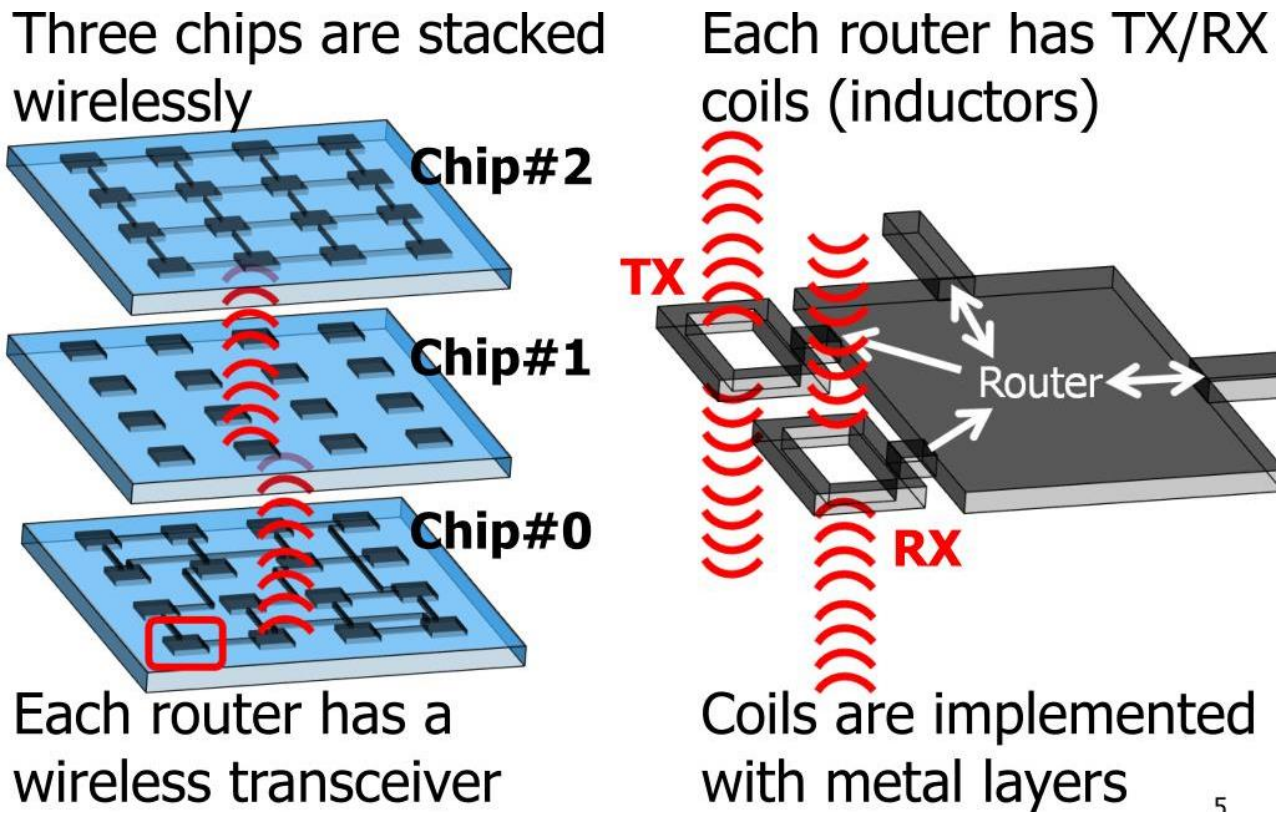
source: Anja von Beuningen, etc. PLATON: A Force-Directed Placement Algorithm for 3D Optical Networks-on-Chip

Reconfigurable NoC



source: Ramy A, Mostafa H, et al. Design of a reconfigurable network-on-chip for next generation FPGAs using Dynamic Partial Reconfiguration[J]. Microelectronics Journal, 2021, 108: 104964.

Wireless NoC



source: https://www.slideserve.com/nadine-brown/low-latency-wireless-3d-nocs-via-randomized-shortcut-chips-powerpoint-ppt-presentation

Companies, Simulators

- For info on NoC related companies, simulators, other tools, conference pointers, etc. please see:
  - http://networkonchip.wordpress.com/
- International Symposium on Networks-on-Chip (NOCS)
  - https://esweek.org/nocs/
  - https://nocs2022.github.io/

NoC Properties

- Reliable and predictable electrical and physical properties → Predictability
- Regular geometry → Scalability
- Flexible QoS guarantees
- Higher bandwidth
- Reusable components
  - Buffers, arbiters, routers, protocol stack

Summary

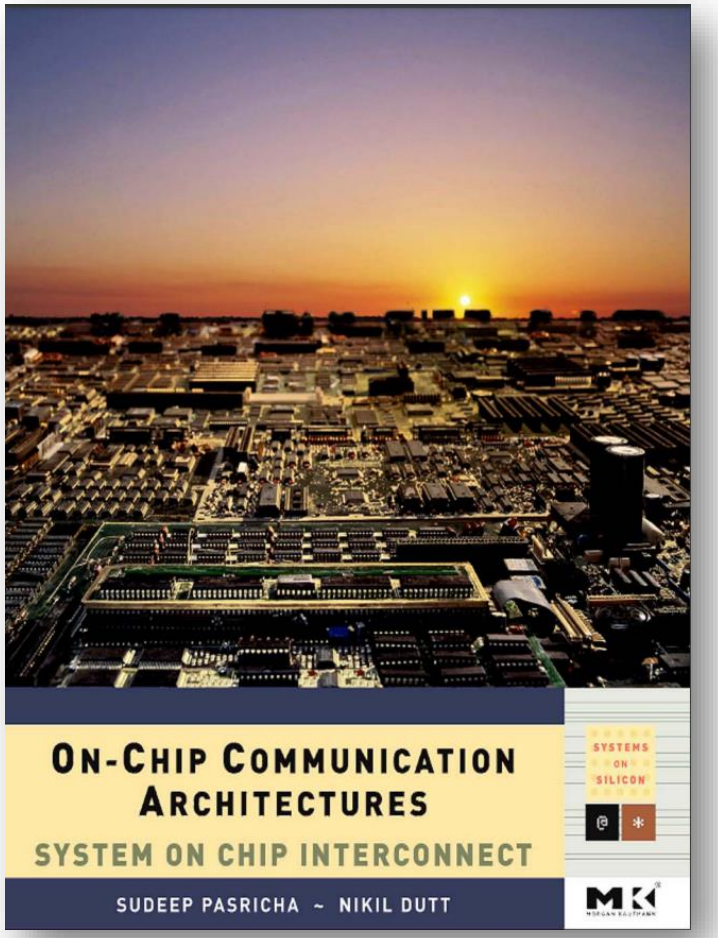
- **SoC complexity is increasing rapidly, due to**
  - Digital convergence
  - Process technology shrinking into DSM era
- **On-chip communication architectures are critical components in SoC designs**
  - To meet power, performance, cost, reliability constraints
  - Also rapidly increasing in complexity with increasing no. of cores
- **Basic concepts of (widely used) SoC communication architectures**
  - Bus-based architectures
  - advanced networks-on-chip (NoC)

Where are we Heading?

- ASIC Design Flow

Action Items

- Reading Materials
  - Ch. 5.4
  - **Ch. 12 of Book "On-chip communication architectures" (on canvas > Reading materials > Literature)**



Acknowledgement

Slides in this topic are inspired in part by material developed and copyright by:

- Dr. Wayne Luk (Imperial College)
- Dr. Gul N. Khan (Ryerson University)
- Dr. Andreas Gerstlauer (UT Austin)
- Dr. Anand Raghunathan (Purdue)
- Dr. Sudeep Pasricha (Colorado)
- Dr. Nikil Dutt (UC Irvine)
- Dr. Konstantinos Tatas
- Dr. Jeremy Chan
- Dr. Ben Abdallah, Abderazek
- Dr. Cristinel Ababei