ReCoSoC 2013

Some Design Issues for 3D NoCs: From Circuits to Systems

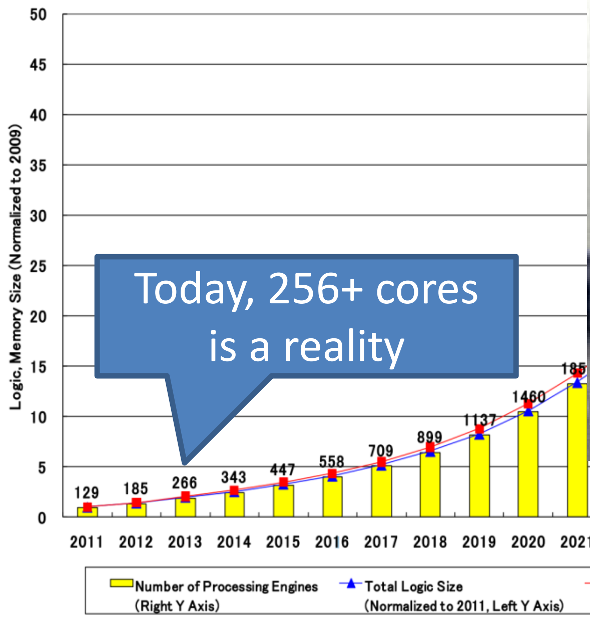
Frédéric Pétrot
with many inputs from
Abbas Sheibanyrad, Florentine Dubois, Sahar Foroutan, Maryam Bahmani
TIMA
System Level Synthesis Group
Grenoble, France

Outline

- 2D (Asynchronous) NoCs
- 3D-Integration and NoCs
  - Vertical Link Asynchronous Serialization
  - Memory on logic and 3D-NoC
  - Vertically-Partially-Connected 3D-NoC
  - A few open problems
- Conclusion

Technology (R)Evolution !

- Fabrication technolo
  - Integration of syst
    transistors in only
  - Hundreds and eve
    components
  - Moore's law is for



Today, 256+ cores is a reality

ITRS 2011 Update

Transistor, 1947

System-on-Board

System-on-Chip

ore Systems

rmance

ower dissipation issues and clock
kews limit clock frequency

arallelism

r Consumption

imple Cores

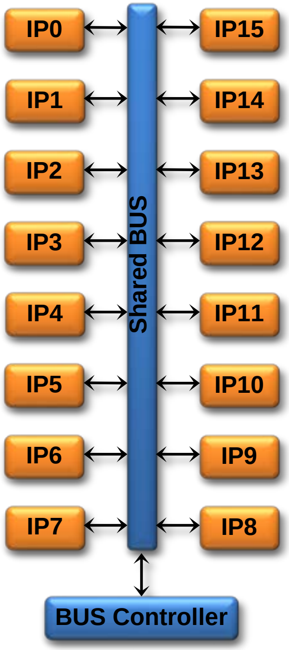
ower down idle cores

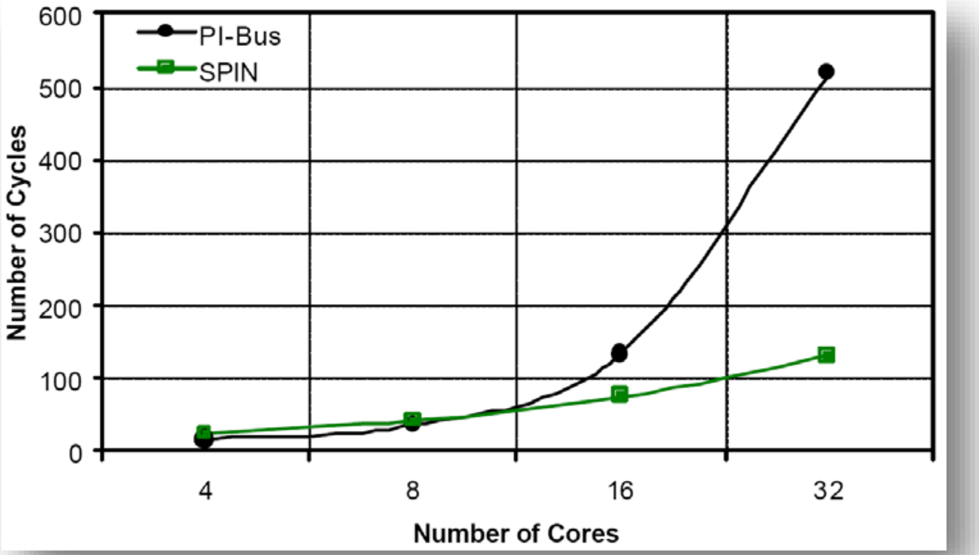
Rapid Design

- Reuse repeated tiles into common infrastructure

Scalability !

- Key role of communication infrastructure
  - Bus: shared medium, broadcast all
  - NoC: data travels along a path, many active concurrently

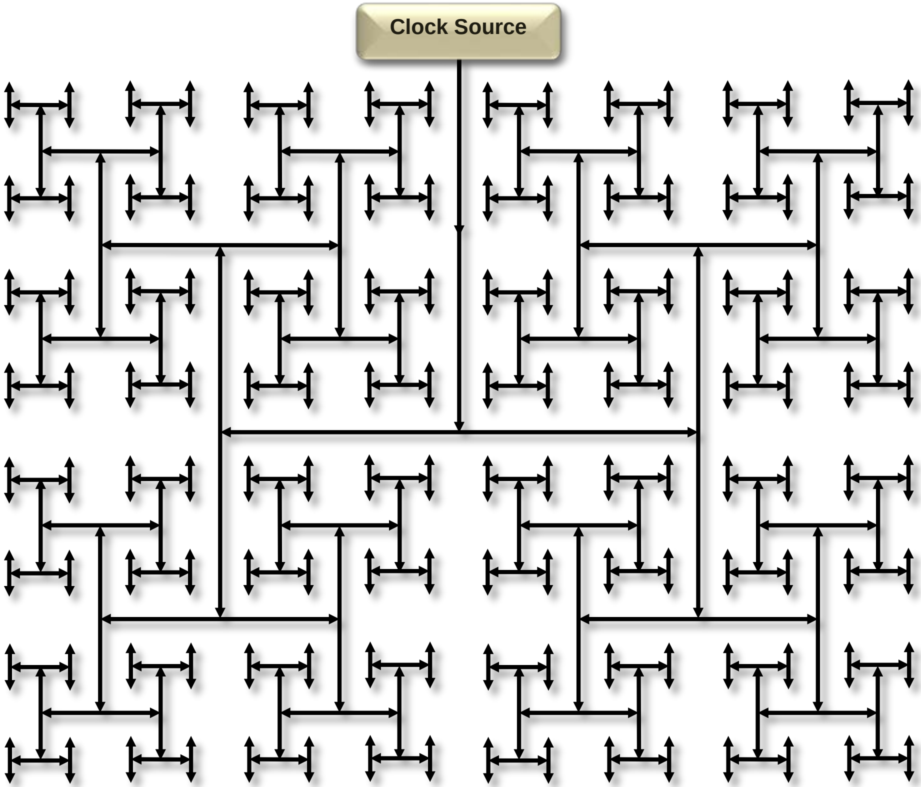


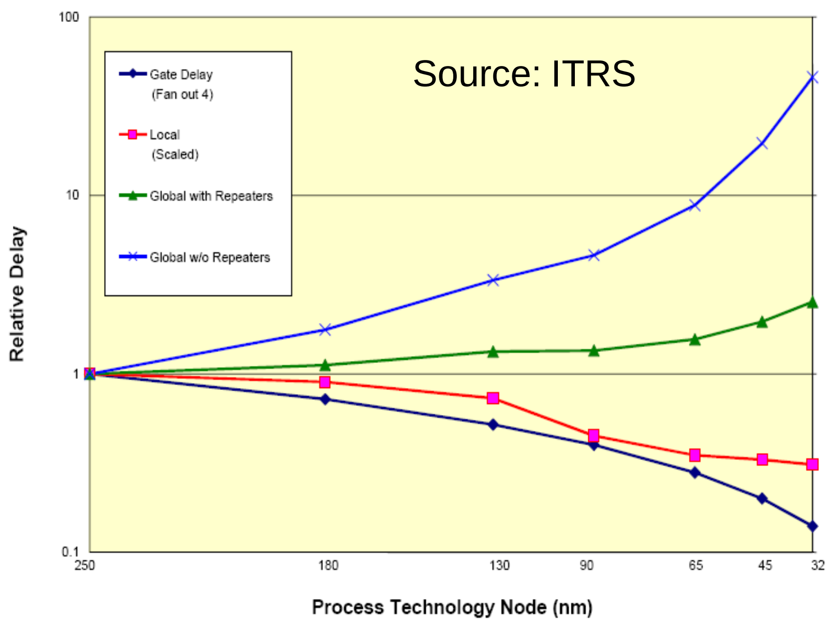


Latency as a function of core number

Clock Distribution

- Deep Submicron Technologies
  - Aggravation of physical problems
  - Predominant effect of long wires on delay and consumption



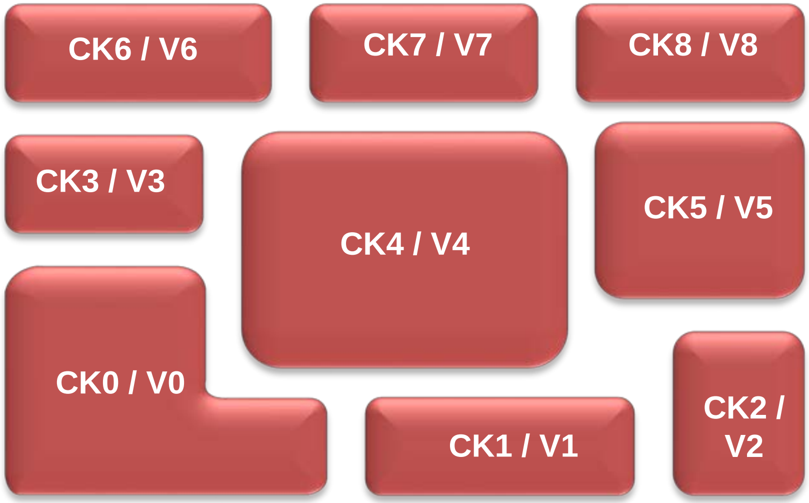


- Nightmare of Global Synchronization
  - Global distribution of single clock signal over chip impossible
  - Clock skew claiming larger relative part of total cycle time
  - Clock distribution network demanding increasing portions of power and area budgets
  - Fabrication Process Variation
  - Temperature Variation

GALS/DVFS Paradigm, Prominent Solution

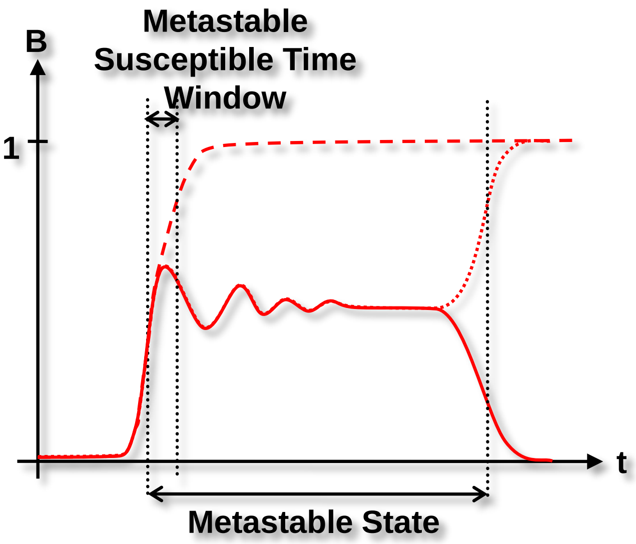
- Reducing the Problem to a number of smaller Sub-Problems
  - Several Independent Clock/Voltage Clusters

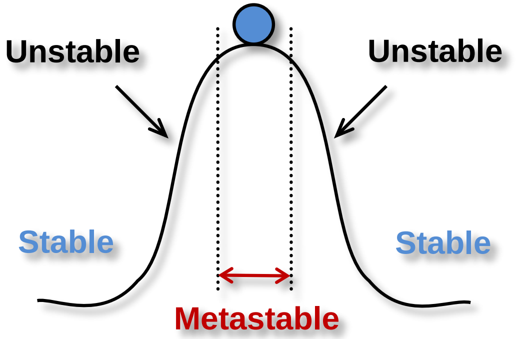
- Networks-on-Chip as a Structured Approach
  - The network: Globally Asynchronous part
  - Subsystems: Locally Synchronous parts



BUT …

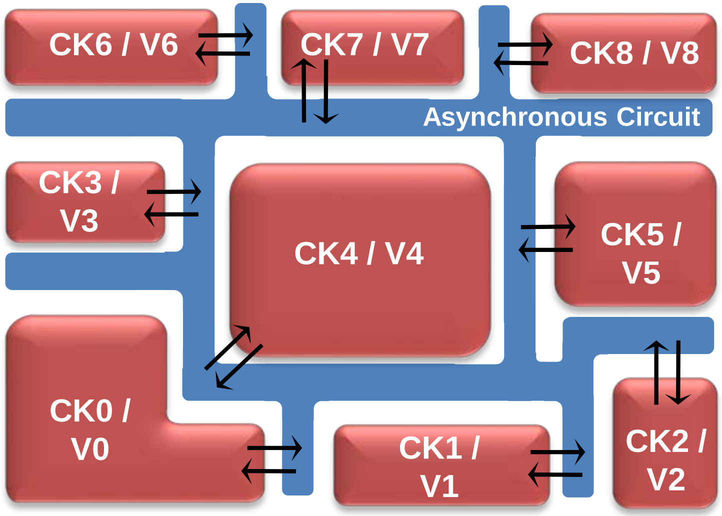
- How separated synchronous domains can robustly communicate together?
  - Transferring data between different frequency domains requires safe synchronization
  - Metastability, unavoidable state of bistable devices, is a major concern





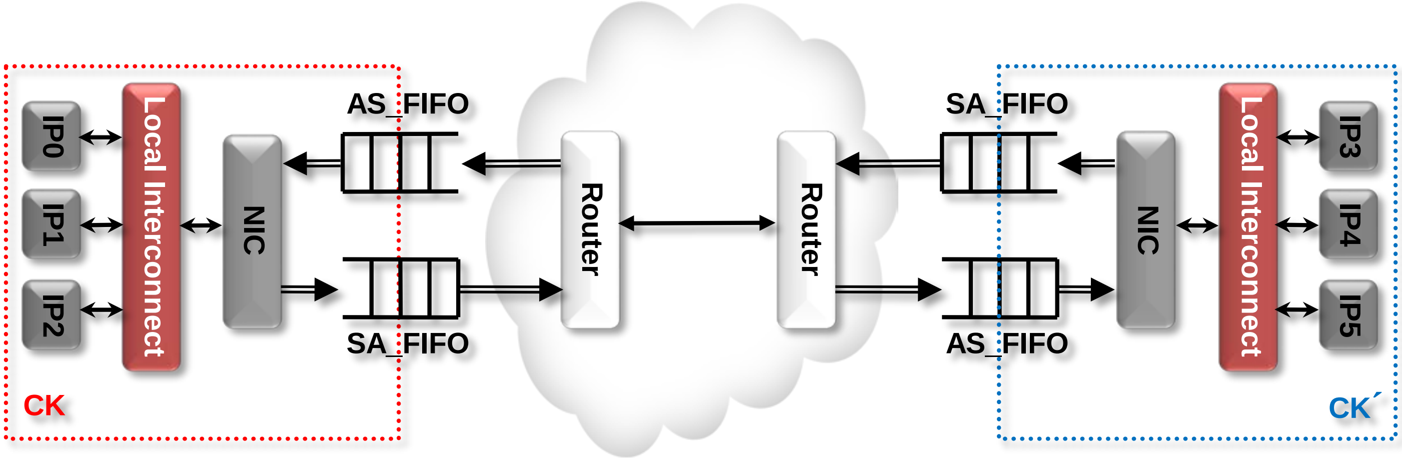
Asynchronous NoC, A Viable Solution

- Reduces the need for synchronization
  - In Network Interface Controller only: 2DFFs to resolve metastability
  - Local Frequency and Voltage control possible (VFI for DVFS)
- Reusability in a Plug-and-Play Fashion
- Almost Zero Idle Dynamic Power Dissipation
- As Fast as Possible
  - Global Timing Independence
- Scalability
  - Cluster Size Independence

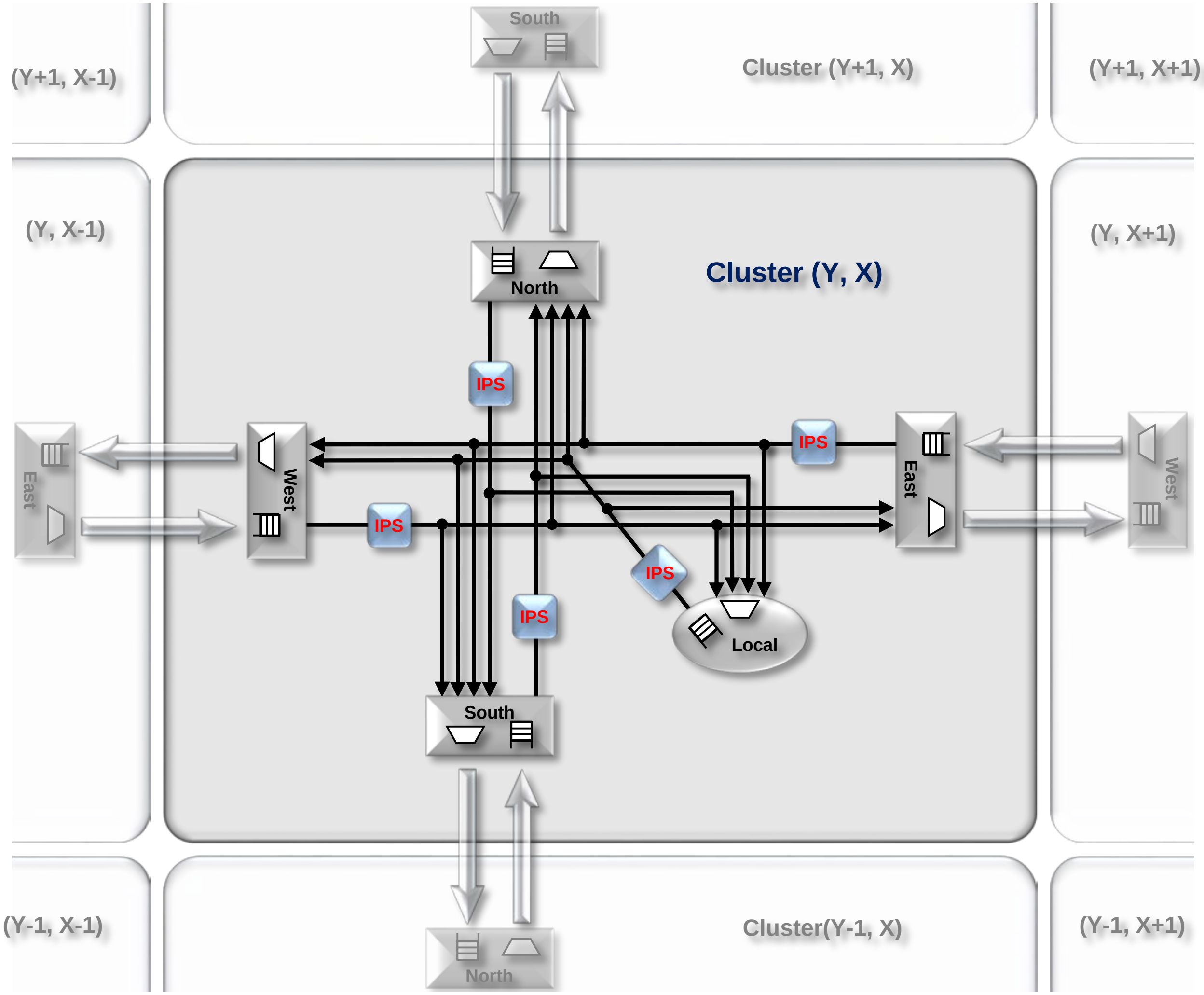


Asynchronous NoC

- Fully asynchronous architecture for the network
- Synchronous compliant interfaces
  - Synchronous-to-Asynchronous FIFO
  - Asynchronous-to-Synchronous FIFO

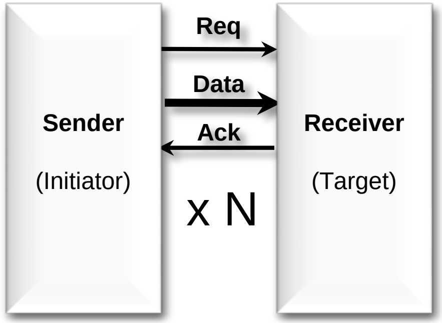


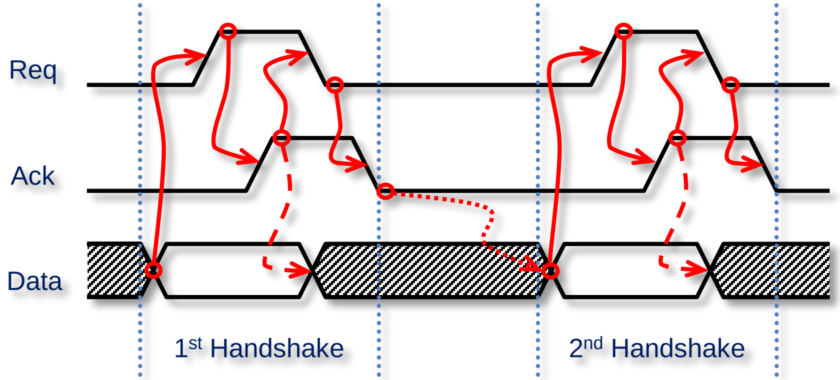
Distributed Router



Asynchronous design in a nutshell

Local handshake instead of global clock





4-Phase protocol

Data availability triggers the computation:

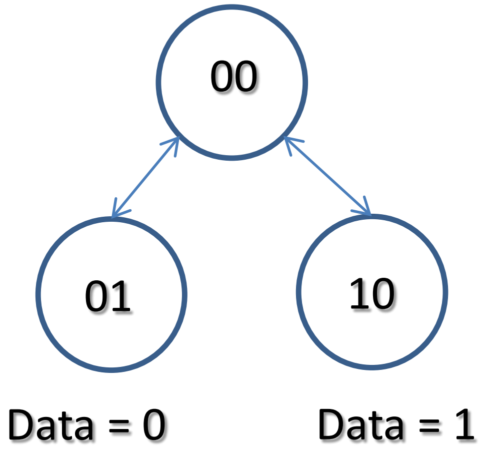
```
wait for valid inputs
   output = f(inputs)
complete input transactions
wait for output ready to receive
   send output
complete output transaction
```

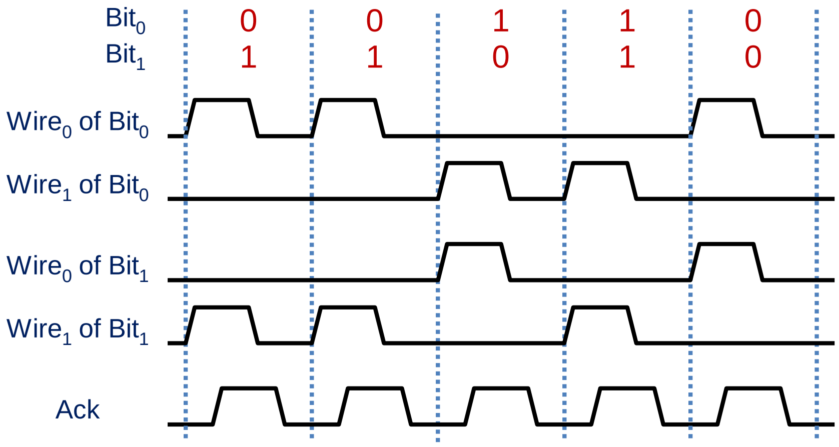
Asynchronous design in a nutshell

Signaling required to minimize delay assumptions, *e.g.* between Data and Request

=> *Embed Request into Data*

Dual-Rail coding: simplest delay insensitive code

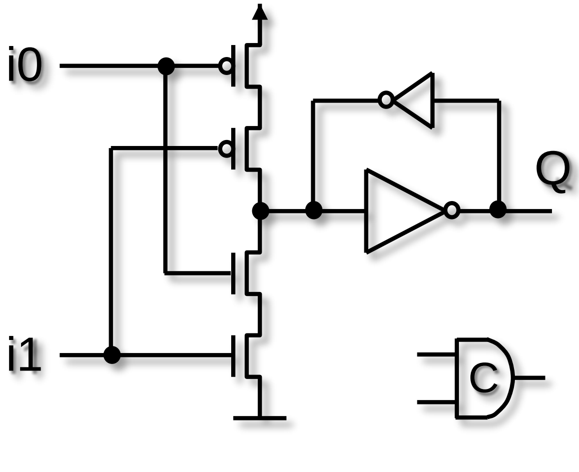



Bit_0
Bit_1
0 1
0 1
1 0
1 1
0 0
Wire_0 of Bit_0
Wire_1 of Bit_0
Wire_0 of Bit_1
Wire_1 of Bit_1
Ack

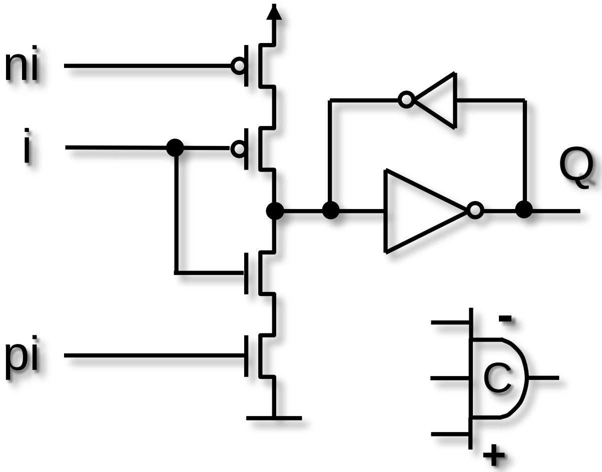
Completion detection indicates computation done:

Ack <= Wire$_0$ or Wire$_1$

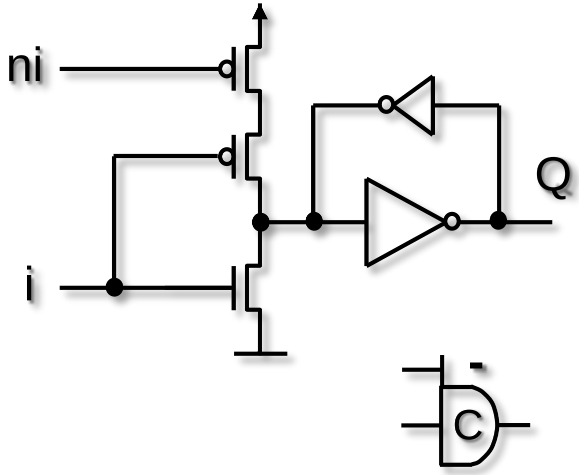
Asynchronous Design Cells (Muller gates)



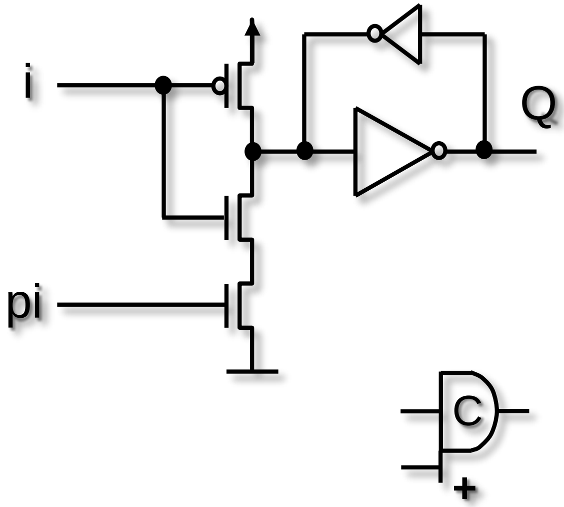
i1	i0	Q
0	0	0
1	1	1
1	0	Q’
0	1	Q’



i	ni	pi	Q
0	0	x	0
1	x	1	1
1	x	0	Q’
0	1	x	Q’



i	ni	Q
0	0	0
1	x	1
0	1	Q’



i	pi	Q
0	x	0
1	1	1
1	0	Q’

Experimental Results

- STMicroelectronics CMOS 90nm GPLVT
- A boundless throughput of connected block as a Hypothesis for throughput measurement

Type	Transistors	Surface	Min Latency	Max Latency	Max Throughput
2-Place SA_FIFO	1338	1422 µm^2	177 pS		2.39 GEvents/S
3-Place SA_FIFO	1969	2054 µm^2	207 pS		2.36 GEvents/S
8-Place SA_FIFO	5126	5215 µm^2	219 pS		2.22 GEvents/S
2-Place AS_FIFO	1388	1452 µm^2	271 pS + T	271 pS + 2T	1.50 GEvents/S
3-Place AS_FIFO	1942	2011 µm^2	247 pS + T	247 pS + 2T	2.61 GEvents/S
8-Place AS_FIFO	5054	5107 µm^2	263 pS + T	263 pS + 2T	2.89 GEvents/S
6-Place SS_FIFO	2985	2940 µm^2	362 pS + T	362 pS + 2T	2.61 GEvents/S
8-Place SS_FIFO	3956	3869 µm^2	366 pS + T	366 pS + 2T	4.60 GEvents/S

Asynchronous NoCs

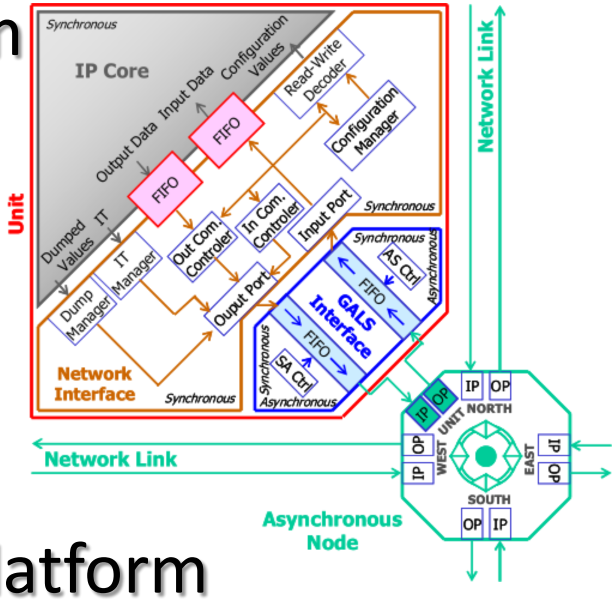
Academic work:

- ANoC, CEA-LETI , 2005, 130 nm (actual chip)
- Alpin, CEA-LETI, 2007, 65 nm (actual chip)
- QNoC, Technion, 2004
- Mango, TU Denmark, 2005, 120 nm
- ASPIN, LIP6, 2008, 90 nm
- Hermes, PUCRS, 2010, 65 nm
- …

Industrial outcome:

- ANoC used in STMicro manycore platform STHorm (2012, 28 nm)

Source: E. Beigné, CEA-LETI

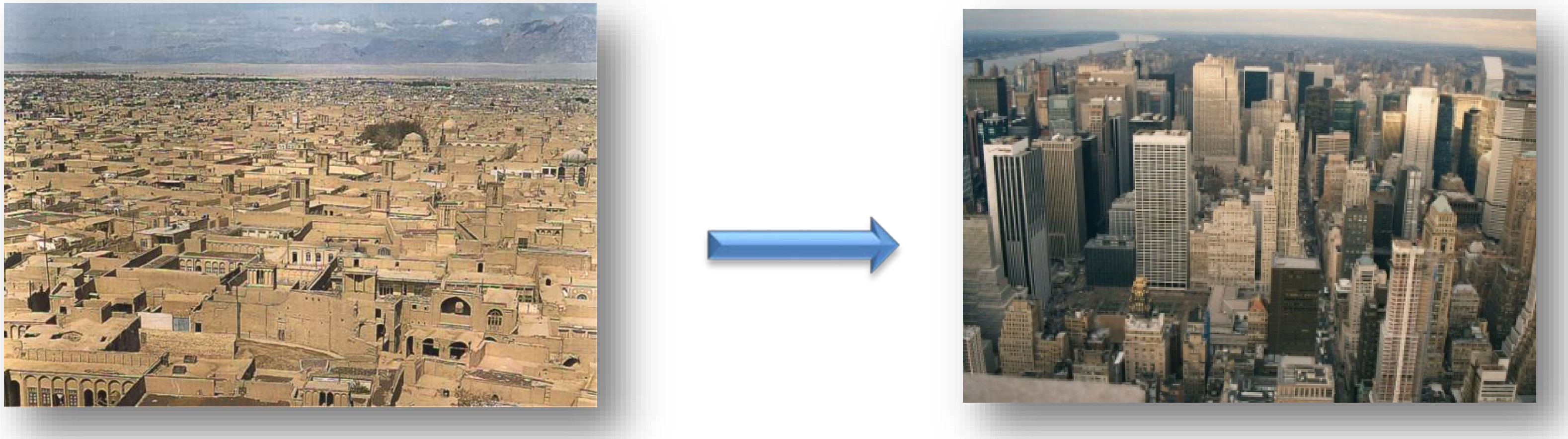


Outline

- 2D (Asynchronous) NoCs
- 3D-Integration and NoCs
    - Vertical Link Serialization
    - Memory on logic and 3D-NoC
    - Vertically-Partially-Connected 3D-NoC
    - A few open problems
- Conclusion

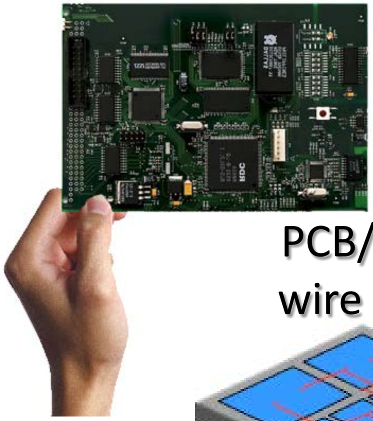
… when the population grows!

- As population grows, tendency to build vertically rather than horizontally
  - Increase the density
    - Land more and more expensive
  - Decrease the length and the number of long paths
    - Average time and energy for moving from one point to another unaffordable
  - However new methods required for going Up & Down!

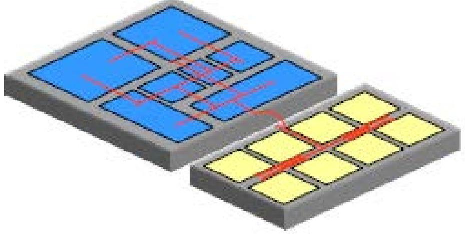


Three-Dimensional Integration

System-on-Board



PCB/SiP with wire bonding

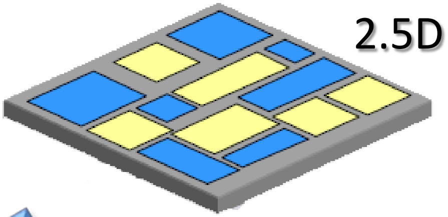


no External Connections

Large Die

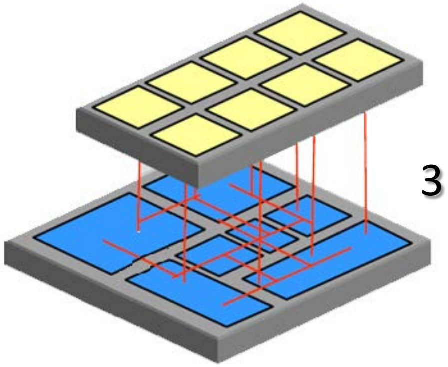
System-on-Chip

2.5D



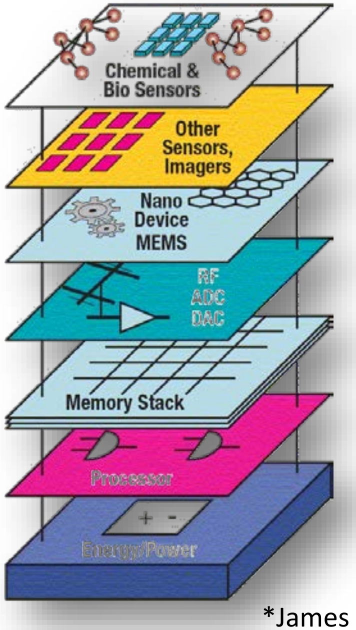
Short and Direct Connections

Small Dies



3D

* Ability to Integrate Different Dies with Different Technologies in a Same Package



*James Lu

Three-Dimensional Integration

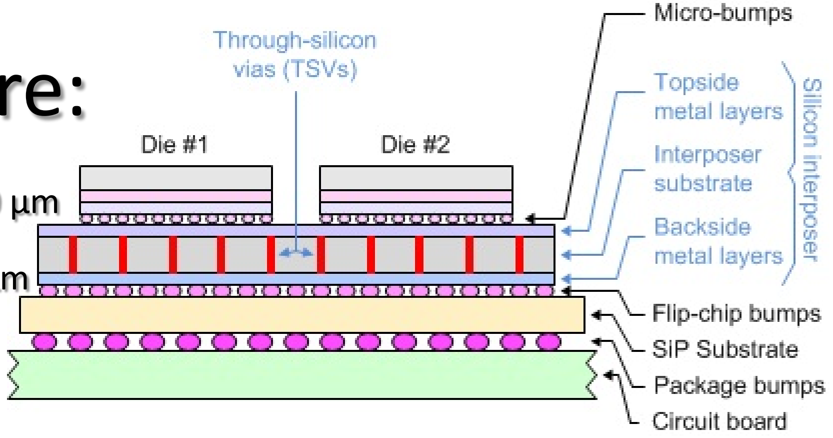
- Today 2.5D is here:

Clive Maxwell, EETimes

Tiny bumps pitch ~10 µm

Bumps pitch ~100 µm

BGA pitch ~600 µm



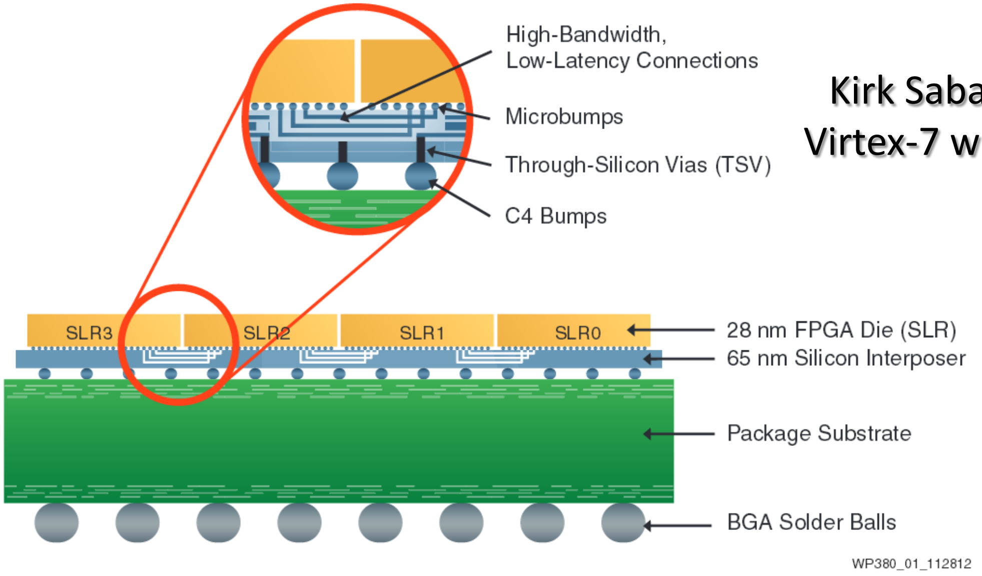


Figure 1: **Virtex®-7 2000T FPGA Enabled by SSI Technology**

Kirk Saban, Xilinx, Virtex-7 white paper

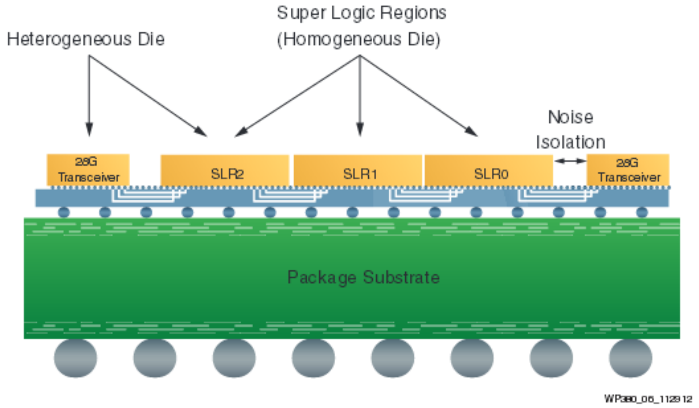


Figure 6: **Heterogeneous 3D FPGA with Integrated 28G Transceivers**

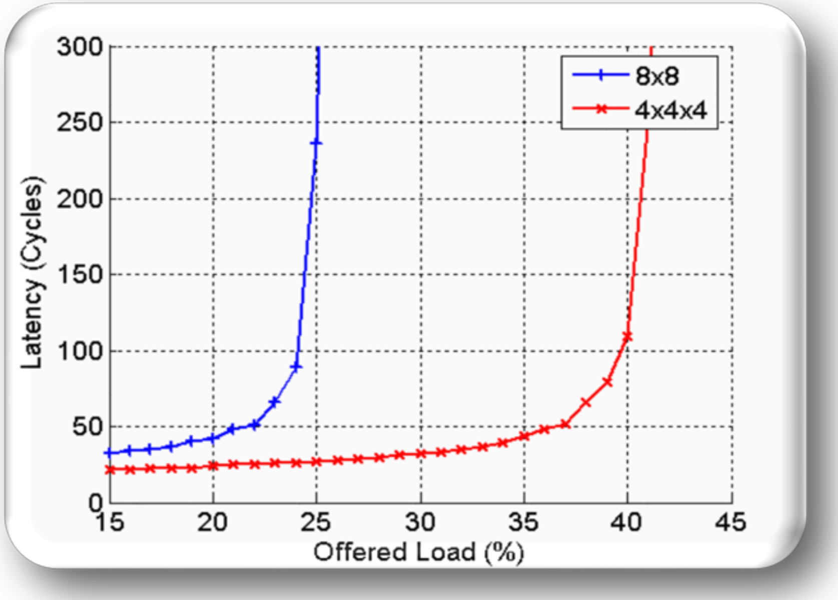
Three-Dimensional Integration

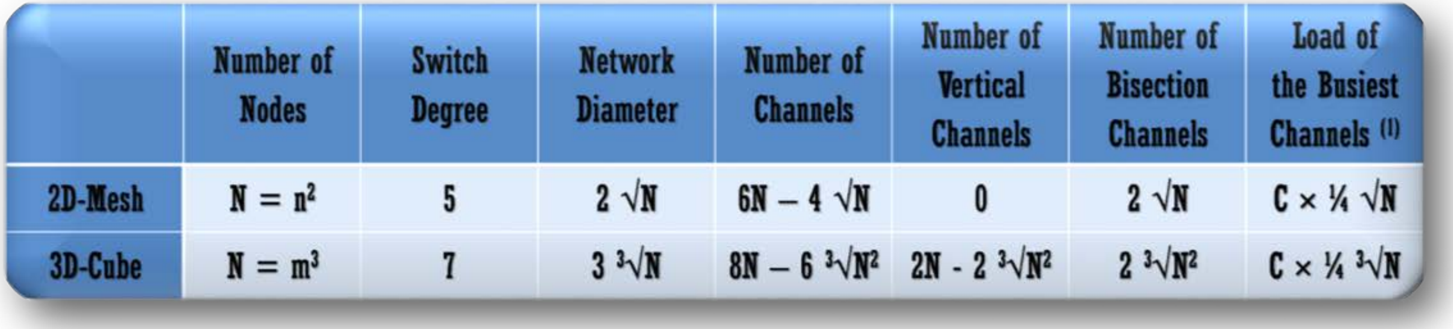
- Tomorrow, true 3D using TSVs?
  – ITRS 2011 update predictions

	2011-2014	2015-2018
TSV diameter	4-8 µm	2-4 µm
TSV pitch	8-16 µm	4-8 µm
TSV depth	20-50 µm	20-50 µm
Number of tiers	2-3	2-4

Courtesy Ivan Miro Panades
CEA-LETI

The Third Dimension

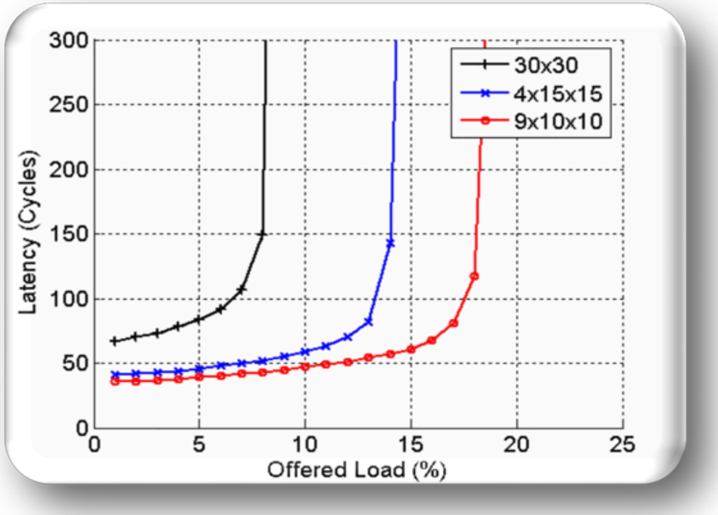


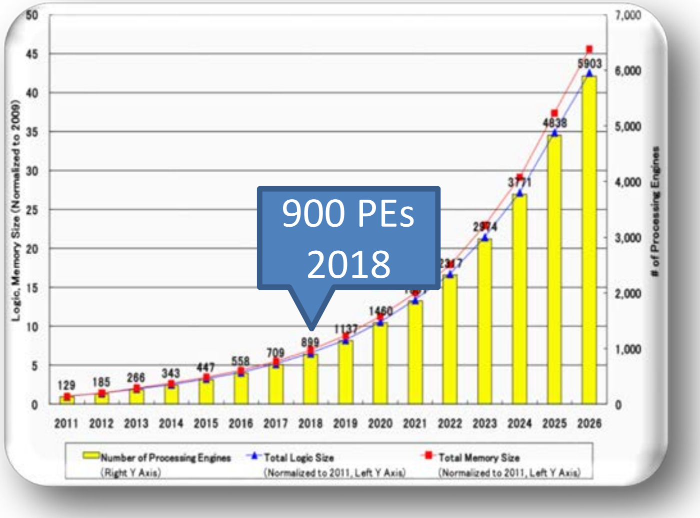


	Number of Nodes	Switch Degree	Network Diameter	Number of Channels	Number of Vertical Channels	Number of Bisection Channels	Load of the Busiest Channels [1]
2D-Mesh	$N = n^2$	5	$2\sqrt{N}$	$6N - 4\sqrt{N}$	0	$2\sqrt{N}$	$C \times \frac{1}{4}\sqrt{N}$
3D-Cube	$N = m^3$	7	$3\sqrt[3]{N}$	$8N - 6\sqrt[3]{N^2}$	$2N - 2\sqrt[3]{N^2}$	$2\sqrt[3]{N^2}$	$C \times \frac{1}{4}\sqrt[3]{N}$

[1] Assuming uniform destination distribution and dimension-ordered routing, C is the average load injected to the network by each node

How Many Layers?





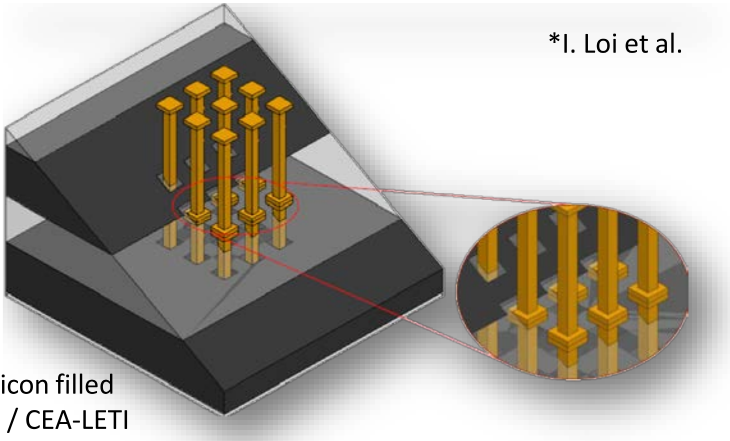
900 PEs
2018

	Number of Nodes	Switch Degree	Network Diameter	Number of Channels	Number of Vertical Channels	Number of Bisection Channels	Load of the Busiest Channels [1]
30x30	900	5	60	5280	0	60	C × ¼ × 30
4x15x15	900	7	34	6510	1350	120	C × ¼ × 15
9x10x10	900	7	29	6640	1600	180	C × ¼ × 10

[1] Assuming uniform destination distribution and dimension-ordered routing, C is the average load injected to the network by each node

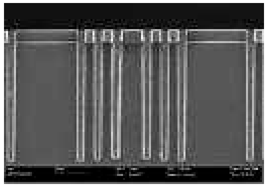
Through-Silicon-Via

- The most promising Technology of Vertical Interconnection
  - Low Resistance and Capacitance
    - High Bandwidth
    - Low Power Consumption



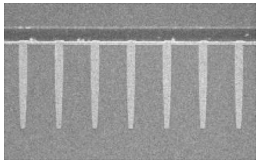
*I. Loi et al.

Via First, Polysilicon filled
Source : G. Pares / CEA-LETI



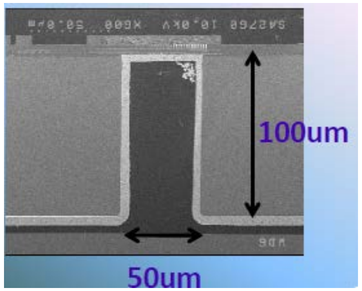
- Via-First (higher density of TSVs)
  - Diameter ≈ 5 µm
  - Pitch ≈ 10 µm
  - Depth ≈ 20-50 µm

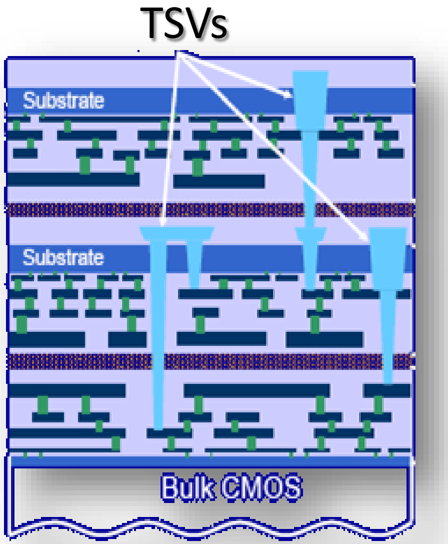
Via Middle, Cu filled
Source : D.Y. Chen,
TSMC/IEDM 2009



- Via-Last (lower cost of the process)
  - Diameter ≈ 35 µm
  - Pitch ≈ 50 µm
  - Depth ≈ 40-150 µm

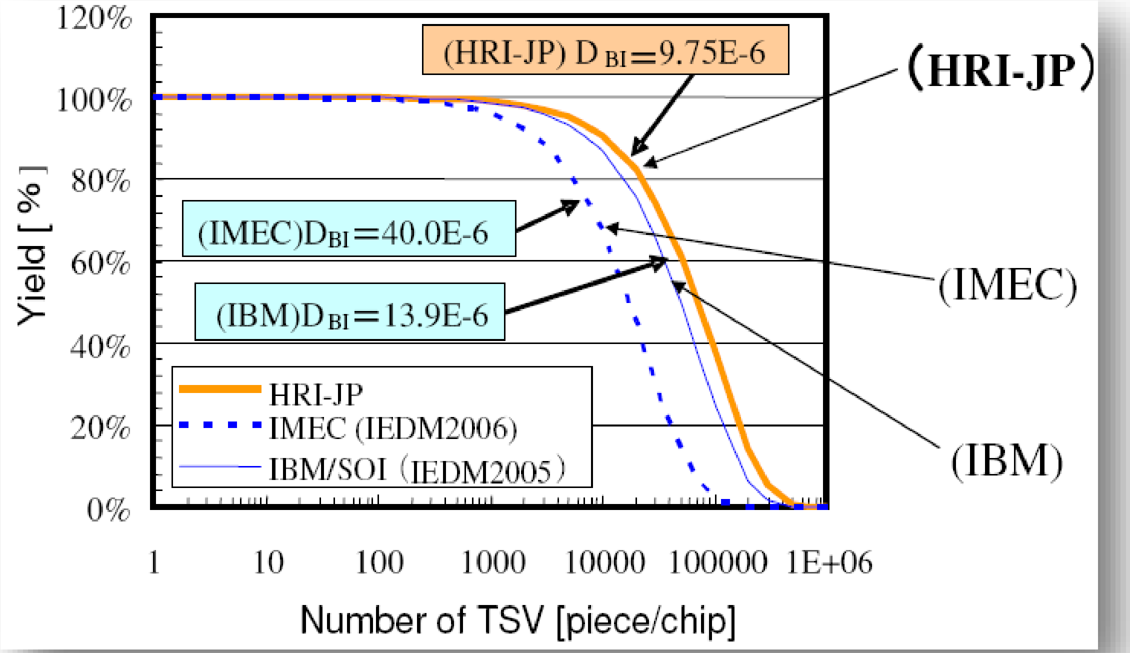
Via Last, Cu liner
Source : Allvia, 2011





…but, is there any problem ?

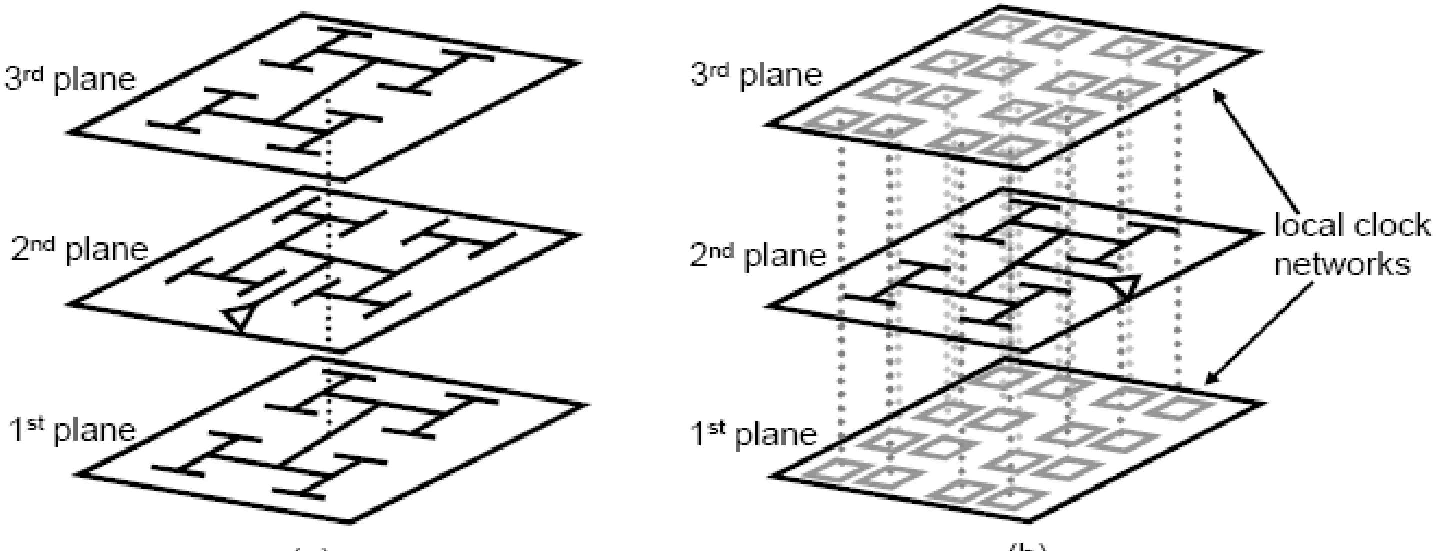
- Large area overhead because of large TSV pitch, mainly due to large pads to compensate misalignment of dies
- Guard zone to active area, ESD protection and level shifters
- Important risk of failure due to several additional fabrication step
  - Misalignment
  - Dislocation
  - Void formation
  - Oxide film formation over Copper interfaces
  - Pad detaching
  - Defects due to temperature
  - …



Three-Dimensional Integrated Circuits are limited by the number of TSVs to be exploited

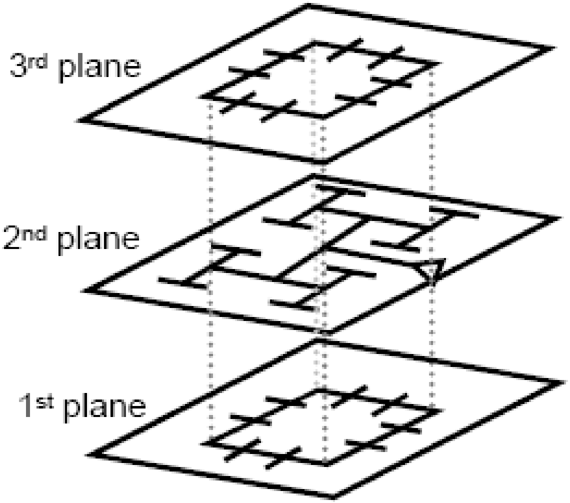
*C. Seiculescu et al.

Clock Distribution in 3 Dimensions



(a) (b)

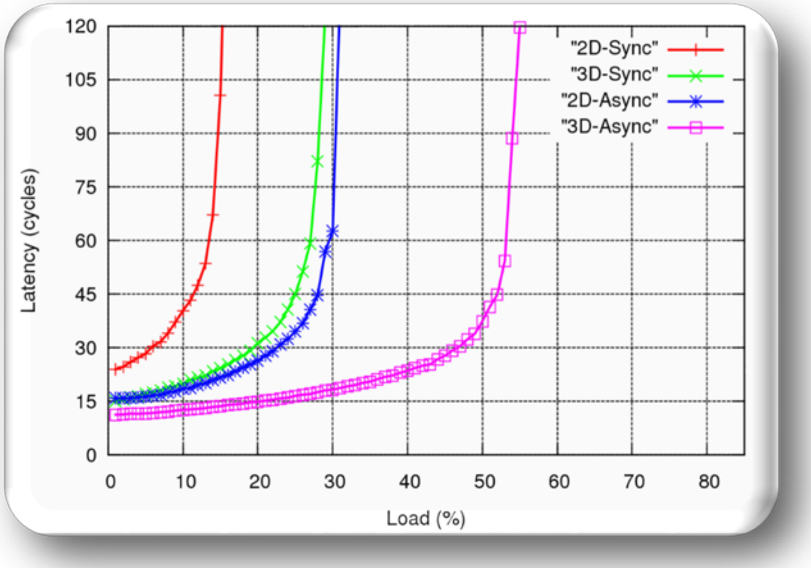
- Skew less distribution of clock across dies unrealistic
  - Batch and chip variability
  - Technology heterogeneity
- Distribution of power is an issue too!

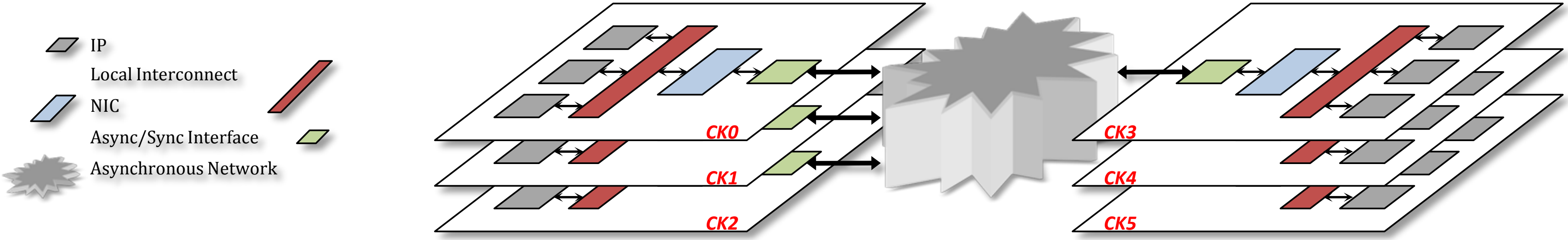


(c) * E. G. Friedman, V. F. Pavlidis

Asynchronous 3D-NoC

- Insensitive to delay variation due to temperature or process variation
- Exploitation of the whole (high) bandwidth of TSVs
- Speed ratio of 2 as a worst-case assumption
  - Using STMicroelectronics 90nm GPLVT transistors, 400MHz as the maximum frequency of usual SoCs
  - Using the same technology, 1100 Mflits/s as throughput of an asynchronous NoC





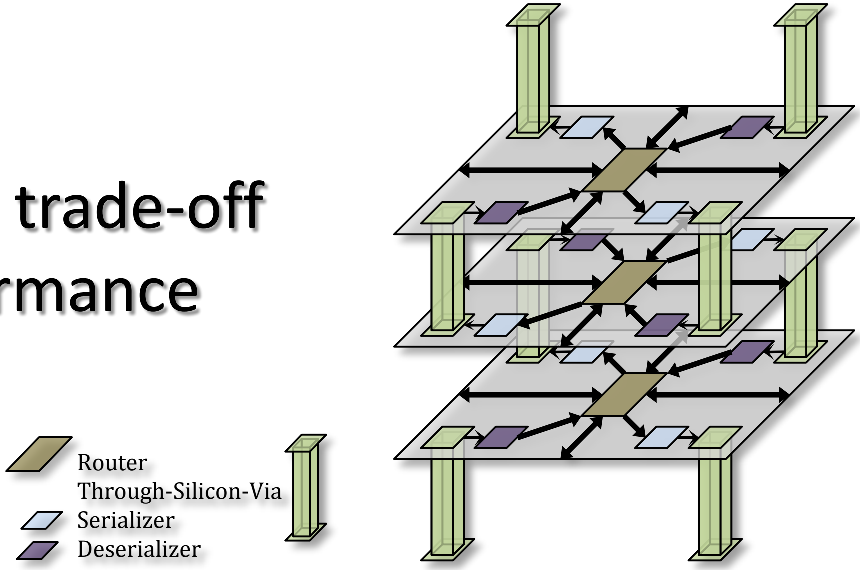
Outline

- 2D (Asynchronous) NoCs
- 3D-Integration and NoCs
  - Vertical Link Serialization
  - Memory on logic and 3D-NoC
  - Vertically-Partially-Connected 3D-NoC
  - A few open problems
- Conclusion

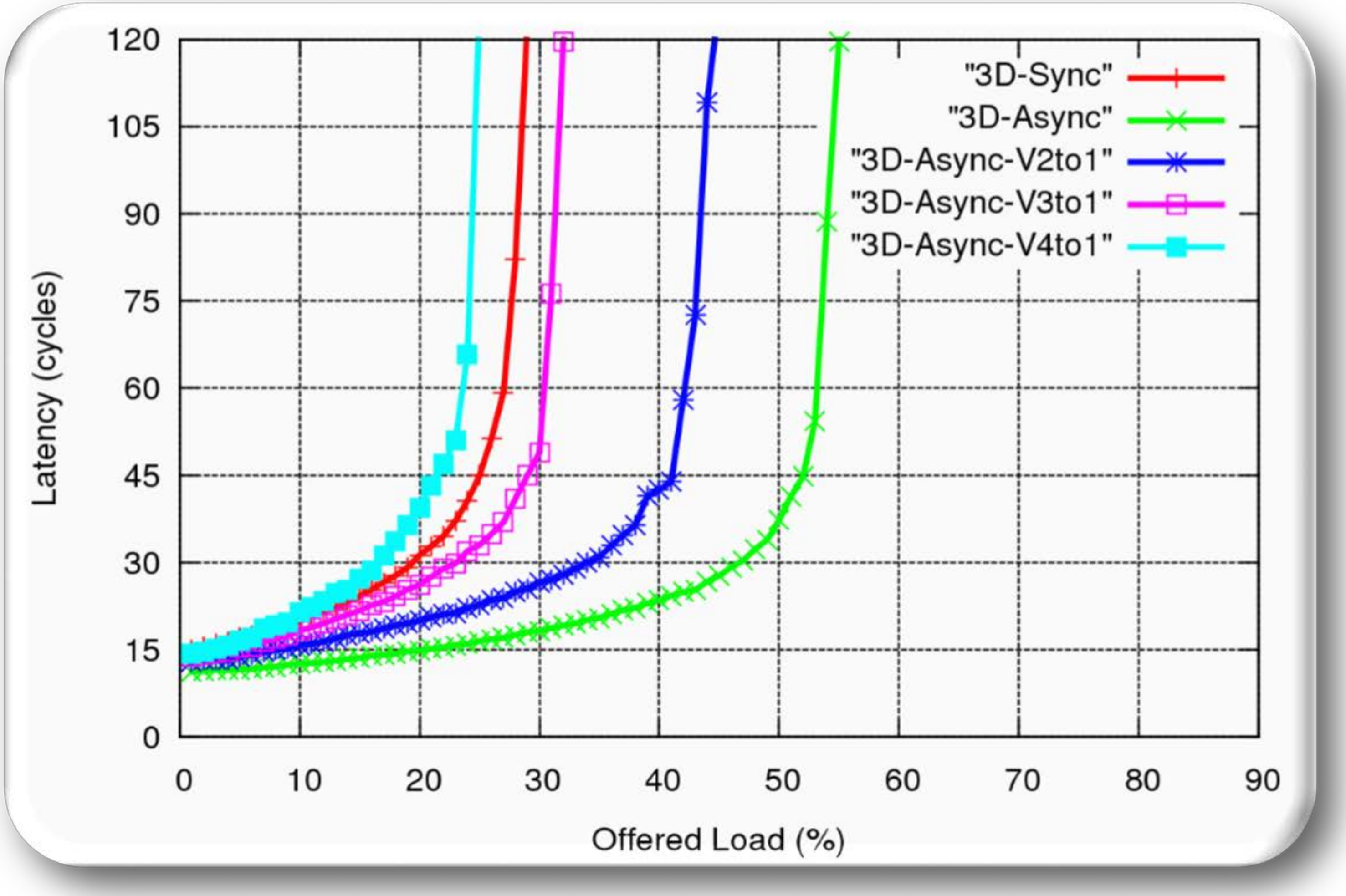
… why not Serialized Vertical Links!

- Remembering
  - Using TSVs guarantees faster vertical data transfer with lower power consumption than horizontal links of moderate size
  - but, pitch of TSVs large, several additional steps of TSV fabrication add potential yield reduction
  - Only small fraction of vertical link capacity exploited in a 3D-NoC
  - Large number of physical connections for each link of the router

- Serialization of data on TSVs is a trade-off between the cost and the performance



Vertically Serialized Asynchronous 3DNoC



Circuit Implementation

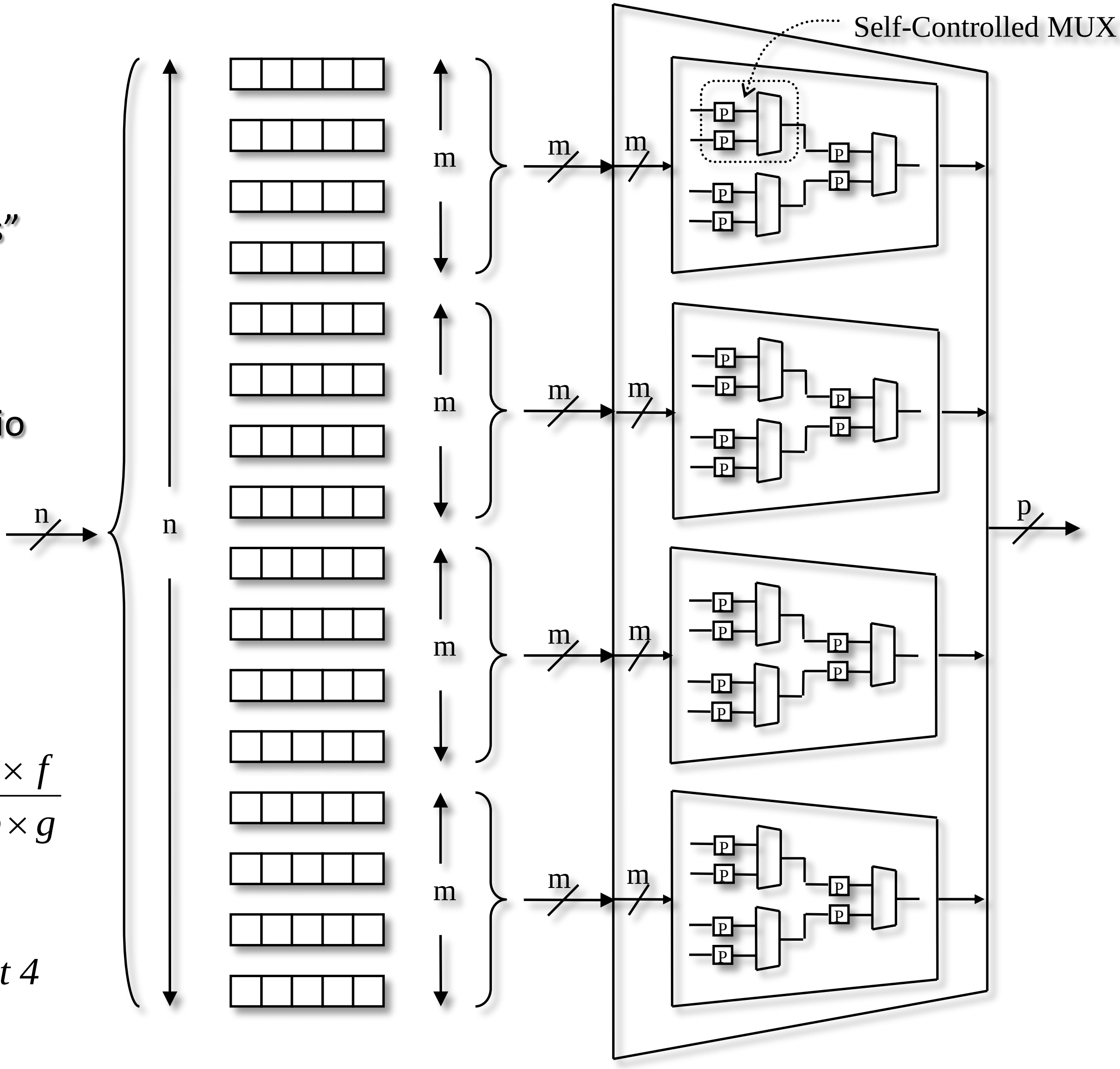
- Serializer of n:p composed of p Serializer of m:1
  - Serializer of m:1 is a tree of "Self-Controlled Multiplexors"

$$m = Serialization\ Ratio = \frac{n}{p}$$

- *R,* The Serialization Bandwidth Ratio as the throughput cost factor
  - *f,* the transfer rate of parallel input data
  - g, the transfer rate of serialized output data

$$R = Serialization\ Bandwidth\ Ratio = \frac{n \times f}{p \times g}$$

$$R = \frac{32 \times 750 Mflits/s}{8 \times 2800 Mflits/s} = 1.07, \textit{ and not 4}$$



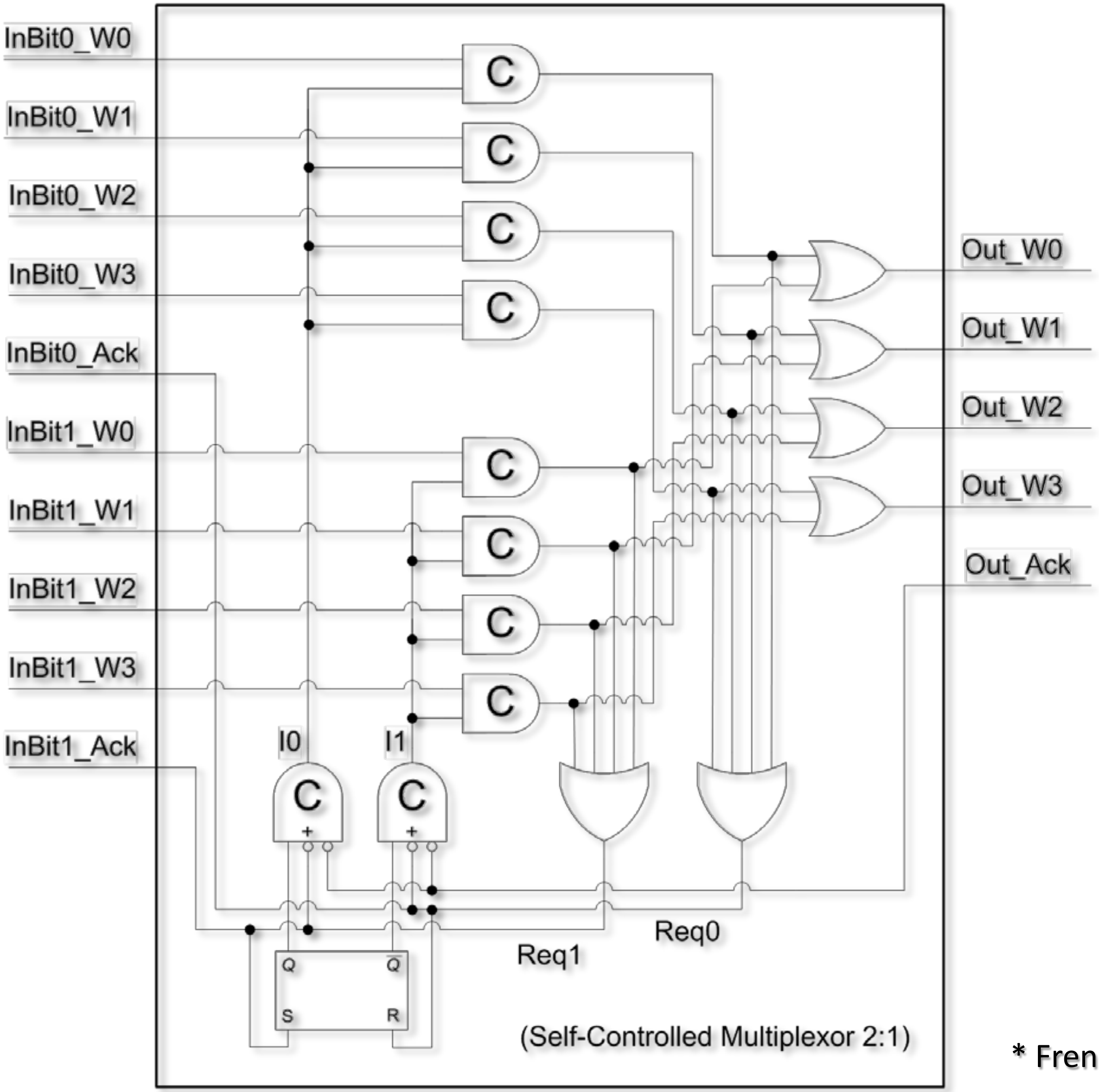
SPICE Simulation Results

- Horizontal Link Throughput: 710 Mflits/sec
  - Router Throughput : 1100 Mflits/sec
  - Inter-Core wire (2mm) delay : 125 ps
- Serialized (8:1) Vertical Link Throughput: 2080 Mflits/sec
  - Serialization Throughput: 2500 Mflits/sec
  - TSV delay: 20 ps

- Speed ratio : (710*32)/(2080*4) = 2.73 (and not 8 !)

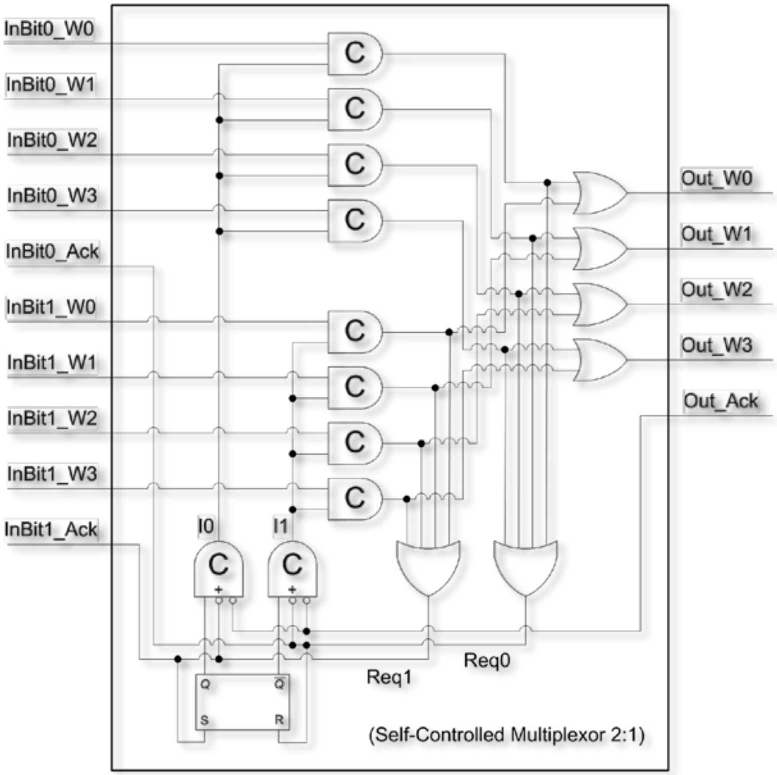
	Self-Controlled Multiplexer 2:1	Self-Controlled Demultiplexor 1:2	Serializer 4:1	Deserializer 1:4	Serializer 8:1	Deserializer 1:8
Transistor count	130	132	390	396	910	924
Latency	80 ps	70 ps	150 ps	130 ps	220 ps	190 ps
Throughput	2.9 Gflits/sec	3.2 Gflits/sec	2.5 Gflits/sec	2.8 Gflits/sec	2.5 Gflits/sec	2.8 Gflits/sec

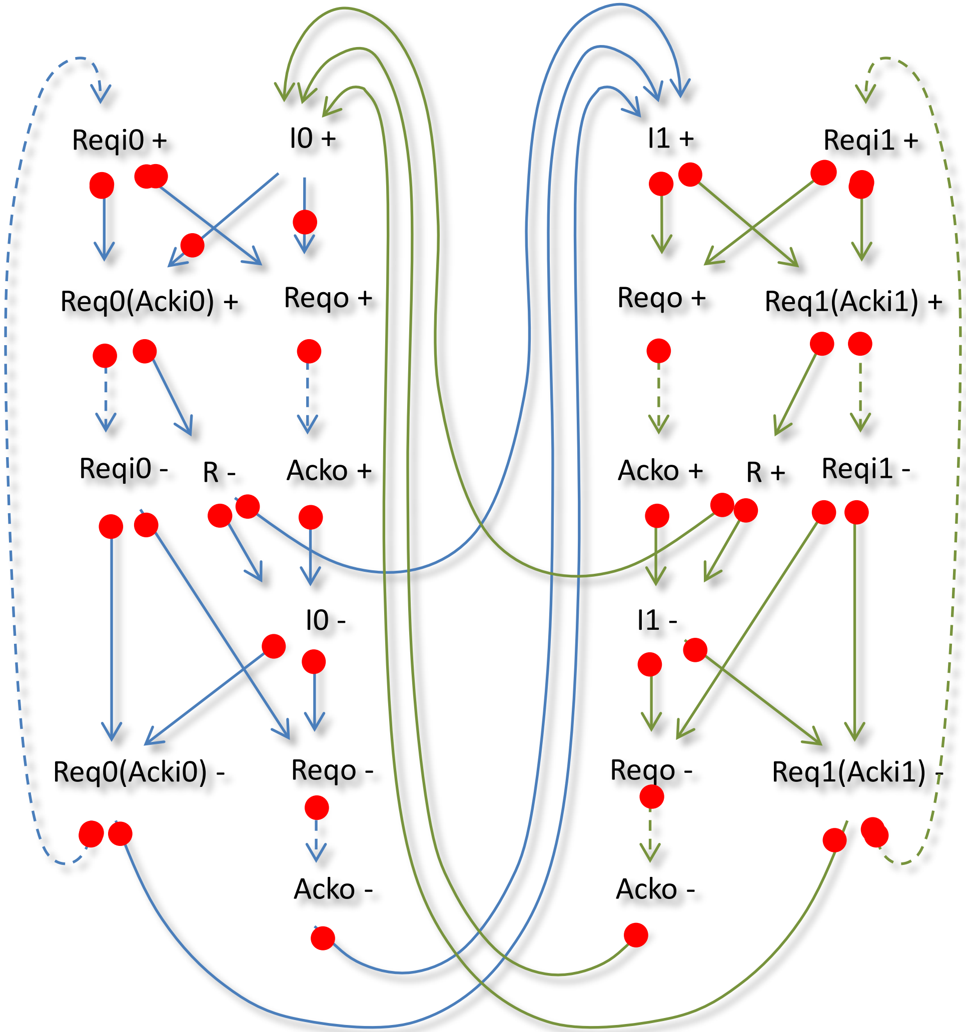
Self-Controlled Multiplexor



* French Patent 09/53637

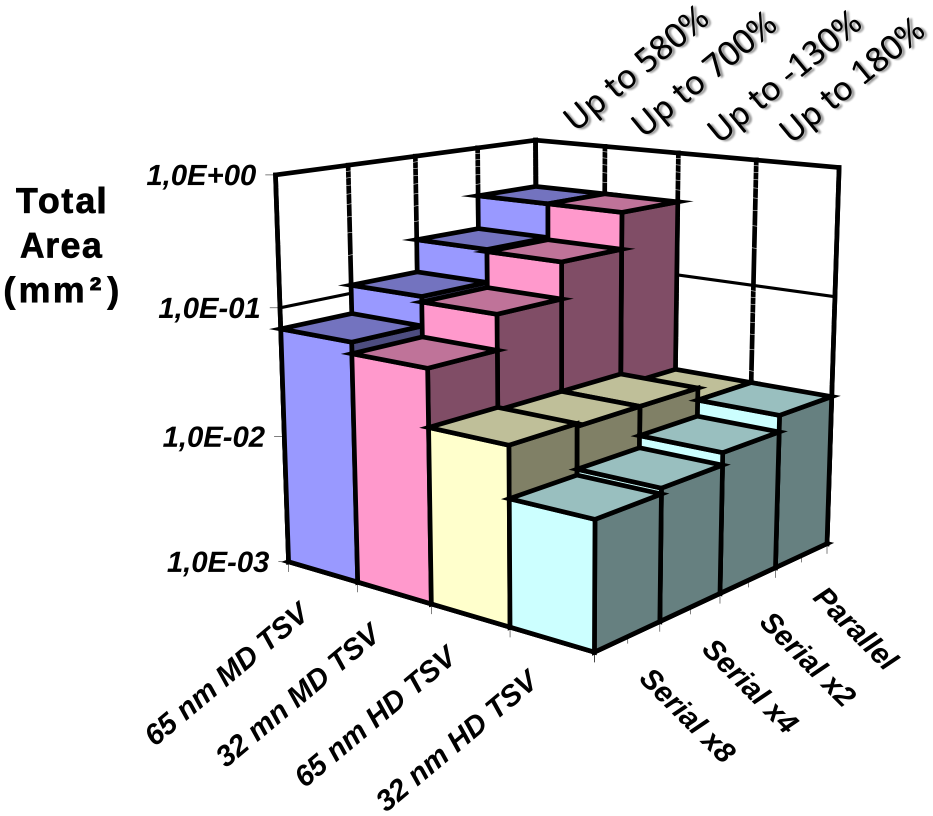
Signal Transitions





Serialization Cost Analysis

	MD TSV	HD TSV	65 nm	32 nm
Parallel	0.4 mm²	0.016 mm²	0 mm²	0 mm²
Serial x2	0.2 mm²	0.008 mm²	0.012 mm²	0.0039 mm²
Serial x4	0.1 mm²	0.004 mm²	0.016 mm²	0.0056 mm²
Serial x8	0.05 mm²	0.002 mm²	0.019 mm²	0.0067 mm²



Latency of Ser+TSV+Des: 2.0 ns in 65 nm

Power: 15 mW (identical to router power)

VLSI integration by CEA Leti

Outline

- 2D (Asynchronous) NoCs
- 3D-Integration and NoCs
  - Vertical Link Serialization
  - Memory on logic and 3D-NoC
  - Vertically-Partially-Connected 3D-NoC
  - A few open problems
- Conclusion

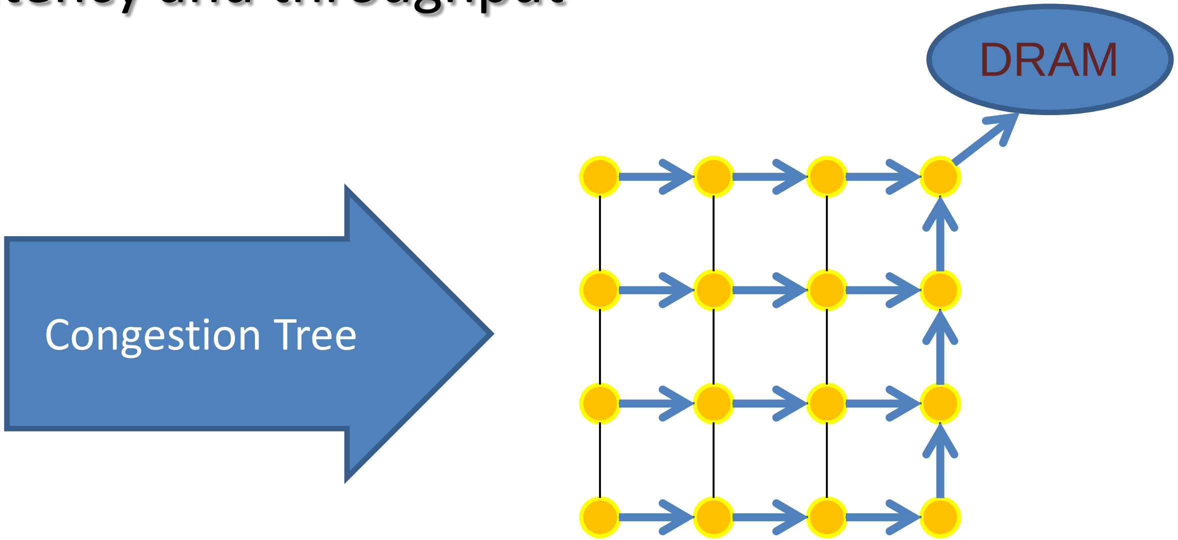
Memory on Logic in 3D-NoCs

Objectives

- Impact of "*DRAM on Logic*" stacking from the NoC Performance-Cost viewpoint
- 3DSoCs with DRAM die(s) on top of the NoC
  - Best-effort, Wormhole NoCs
  - Average performance of the system is addressed
- Distribution of a *limited buffering budget* between NoC links at design time in order to improve system average performance.

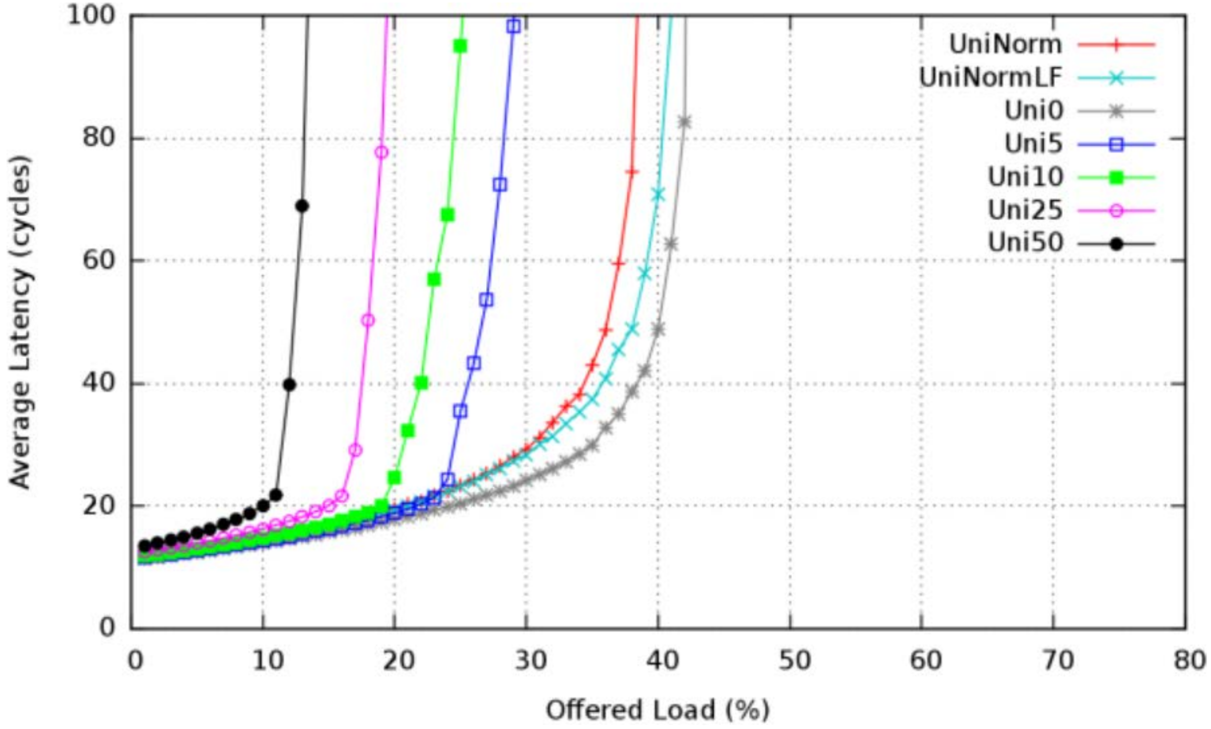
External DRAM: Bottleneck of the System

- External DRAM pad in classic shared memory MPSoCs creates a drastic hotspot in the SoC and becomes the main bottleneck of the system
- Congested DRAM interface ➔ retro-propagation of traffic ➔ Congestion tree ➔Saturation of the whole SoC
- Significantly Deteriorates the average performance in terms of latency and throughput



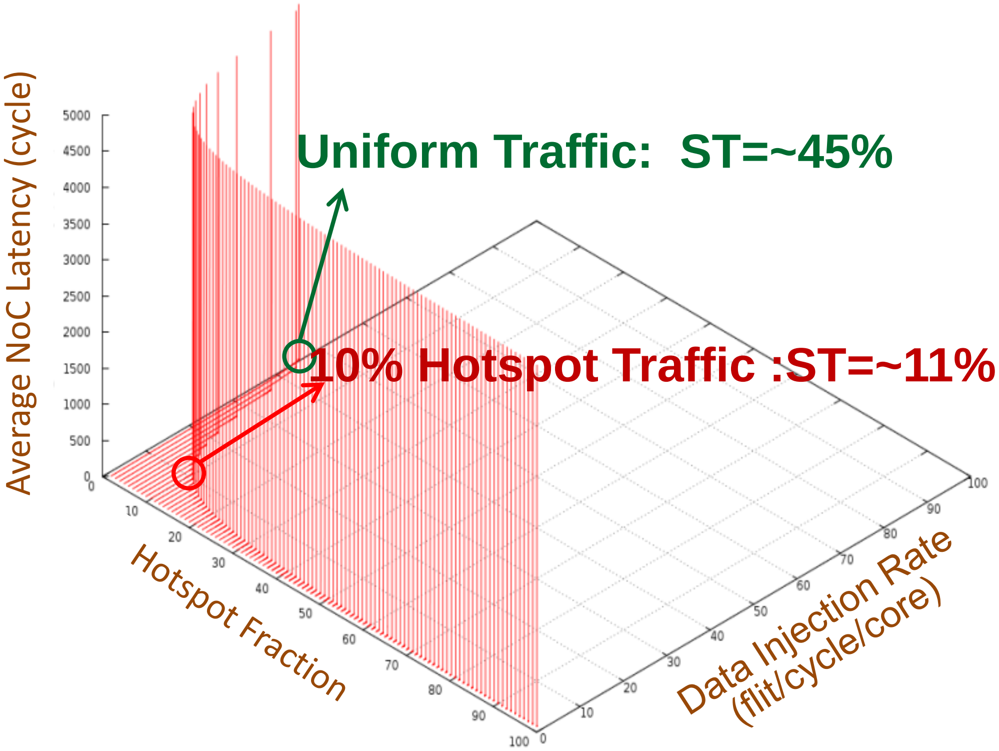
NoC Average Performance

- Saturation threshold: maximum load accepted by the network
- Average Latency / Offered Load curves



Issue: Hotspots & NoC Performance Degradation

NoC Saturation Threshold (ST)



Uniform Traffic: ST=~45%

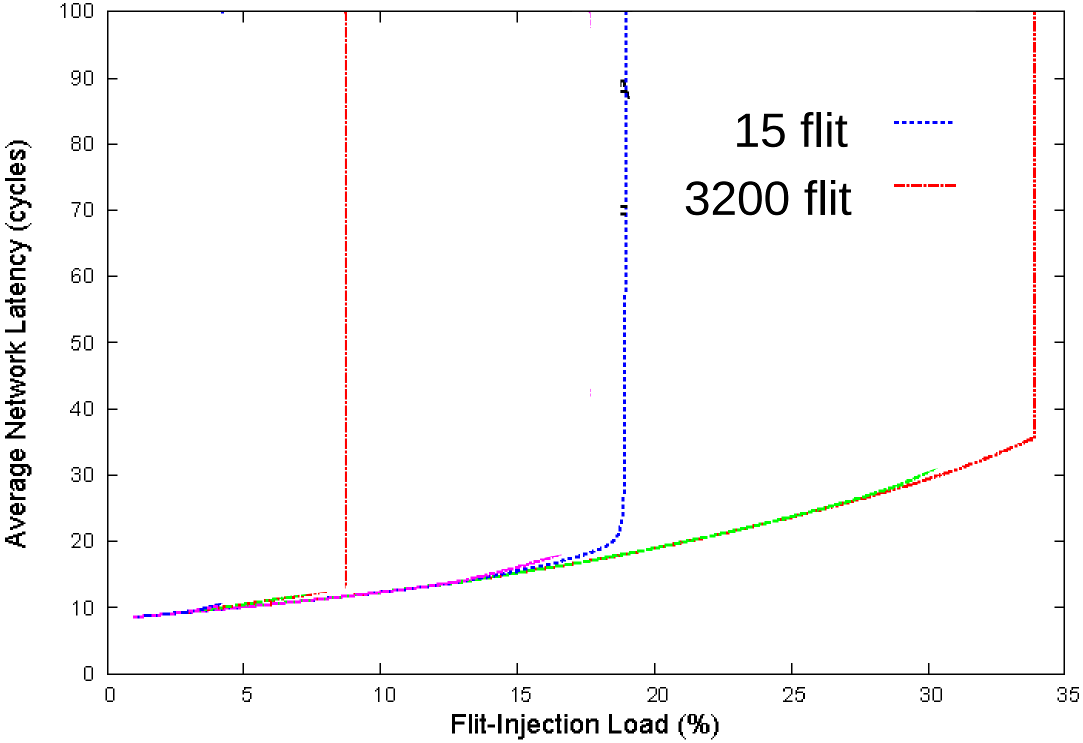
10% Hotspot Traffic :ST=~11%

Stacked DRAM

- One of the major benefits of 3D technology: DRAM/Logic stacking in a single chip (vs. traditional off-chip DRAMs)
- TSVs offer wide and fast vertical interfaces between logic and DRAM, e.g. Wide I/O :
  - 4x128-bit interfaces (4 channels), 200 MHz I/O bus clock
  - Total peak bandwidth of 12.8GB/second (3.2GB/second/Channel)
- NoC and DRAM-port work approximately with the same throughput
- DRAM port is not anymore the major bottleneck of the SoC
- Saturated link buffers ➔ retro-propagation of traffic ➔ Saturation of the whole SoC

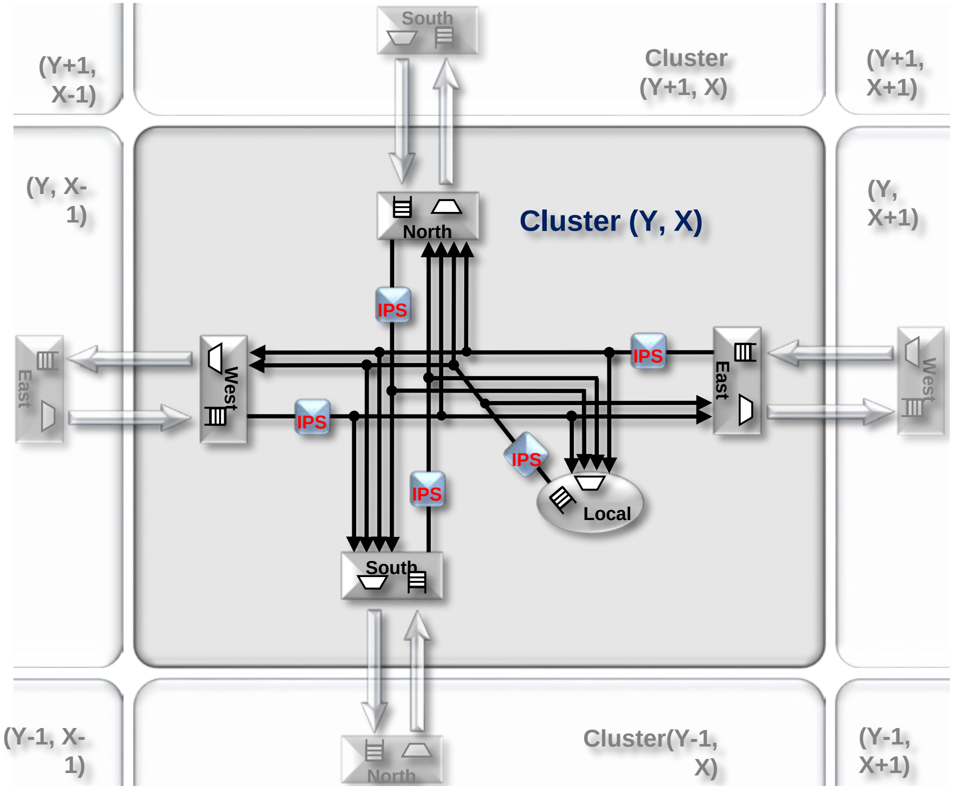
Buffers Sizing at Design Time

- Assignment of large buffers to network links can significantly enhance the NoC saturation point

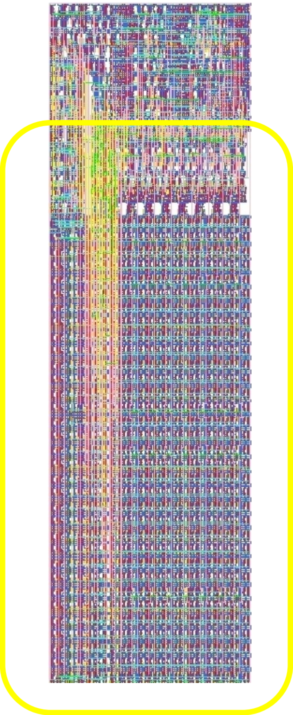


Buffers are Costly

- More than 80% of the area of a router belongs to its buffers



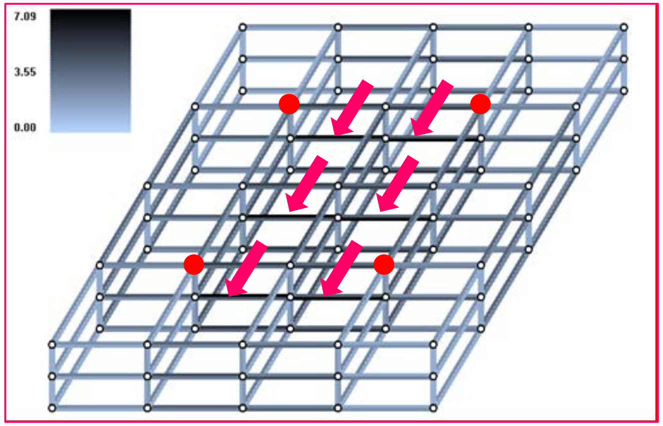
8-flit Buffer



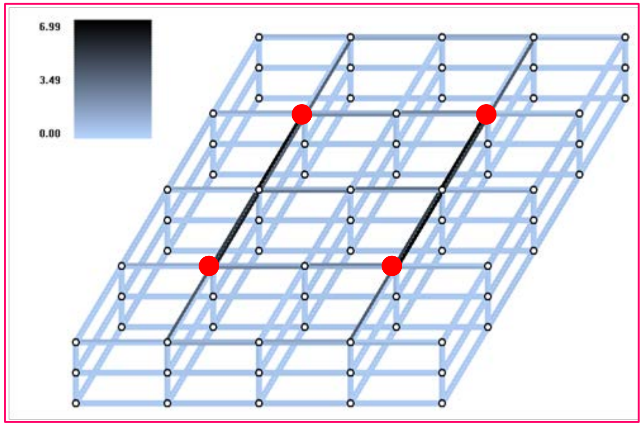
- Therefore, an efficient buffer dimensioning method is required to deal with the tradeoff between NoC cost and performance.

Detection of Bottleneck Links

1. Architectural parameters: e.g. NoC topology, routing algorithm, buffer lengths
2. Traffic Parameters: e.g. number and location of memory controllers, hotspot fraction



h=5%



h=25%

Example: 5% of hotspot traffic vs. 25% of hotspot traffic (ZXY routing algorithm, 8 flits buffer length, uniform background traffic)

Buffer Sizing Approach

- Initial analysis to identify network saturation point: assumes almost infinite buffers, *i.e.* 3200 flits

- Determine average utilization for every buffer *b* (*Utilization(b)*) at load (*ST – epsilon*)

- Compute normalization factor:
  $F = (Max_{len} - Min_{len})/Bottleneck$
  - Max_{len}, Min_{len}: size of largest and smallest buffer (designer choice)
  - *Bottleneck*: maximal observed buffer utilization (given by simulation)

- Normalize buffer size:
  $Size(b) = Utilization(b) * F + Min_{len}$

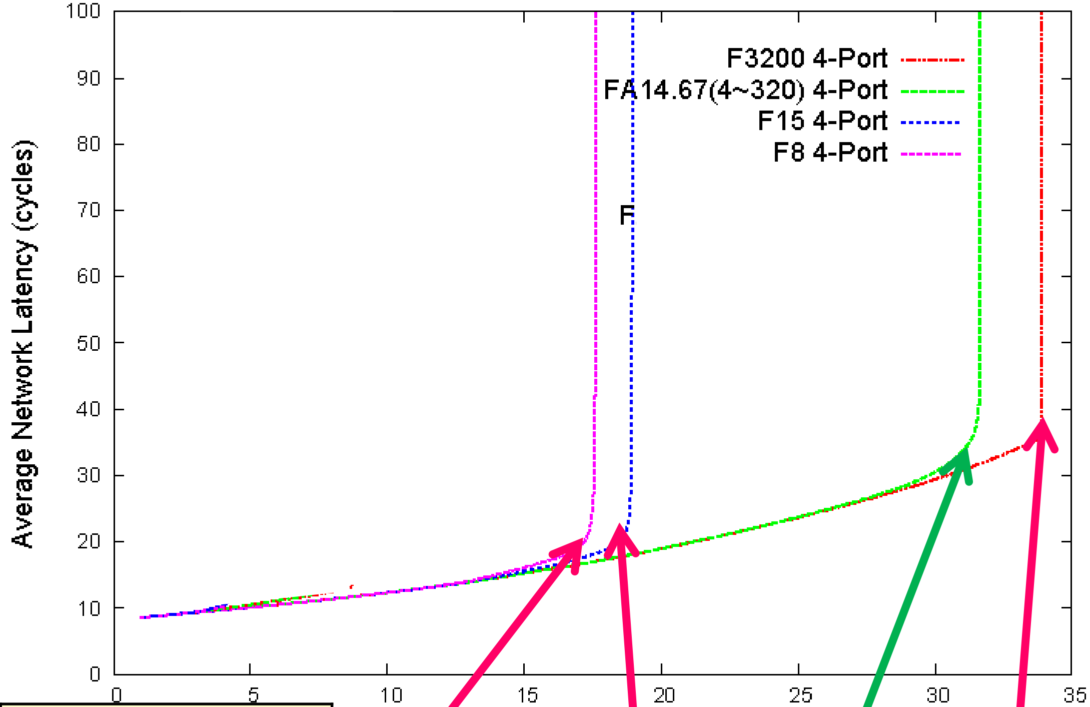
Buffer Sizing Approach: Cost Analysis

- Assignment of buffers between 4 … 320 flits to each link in 5x5x3 mesh wormhole NoC
- Wide IO memories with 1 or 4 ports
- Number of large buffers is very limited
  - Only 24 links out of 340 have buffers greater than 32 flits
- Resulting average buffer length: 14.67 flits

Buffer Length	4	5	6	7	8	9	12	13	14	17	18	19	20	21	22	23	26	36	39	42	43	48	58	59	97	98	128	131	318	319	320
Number of links (4 ports)	-	124	46	46	12	46	10	-	4	3	4	13	1	4	2	1	-	1	3	2	2	4	2	2	-	-	1	3	1	2	1
Number of links (1 port)	328	-	4	-	-	-	-	2	-	-	-	-	-	-	-	1	2	-	-	-	-	-	-	-	1	1	-	-	-	-	1

Cost/Performance Comparison

Significant growth in saturation point with the same cost as a15-flit buffer/link everywhere



Reference 8 flit

15 flit

3200 flit

Average: 14.67 flit

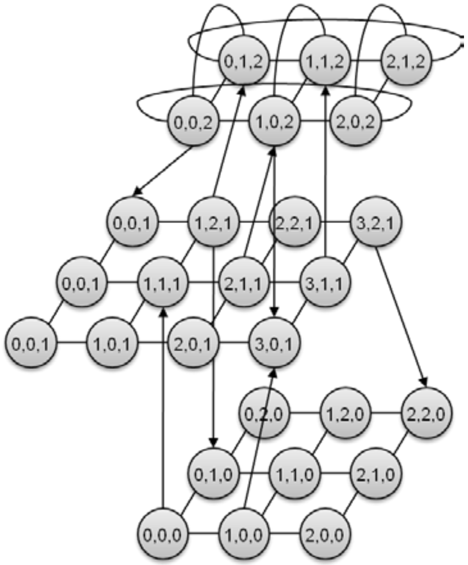
Buffer Length (flits)	Cost	Performance
8 **(reference)**	1	1
15	1.87 X	1.07 X
3200	400 X	1.92 X
Average 14.67 (4~320)	**1.83 X**	**1.79 X**

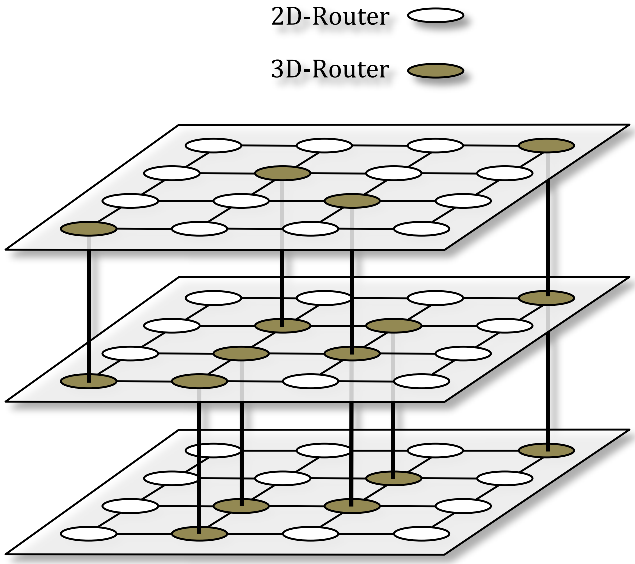
Outline

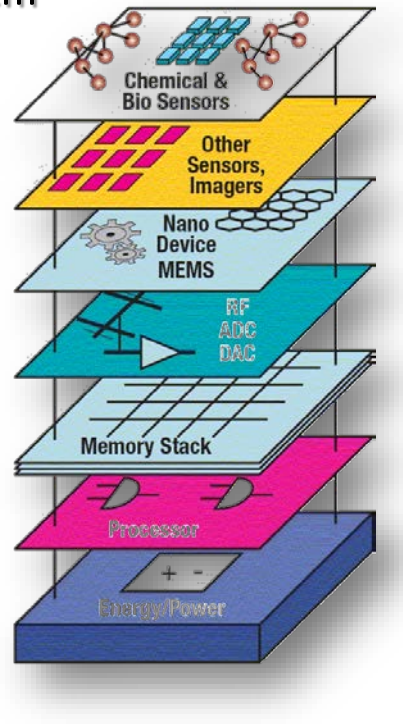
- 2D (Asynchronous) NoCs
- 3D-Integration and NoCs
  - Vertical Link Serialization
  - Memory on logic and 3D-NoC
  - Vertically-Partially-Connected 3D-NoC
  - A few open problems
- Conclusion

Vertically-Partially-Connected 3D-NoC !

- Limited number of vertical connections (TSVs)
- Network with different dies fabricated with different technologies
  - Heterogeneity
  - Irregularity
- Vertically-Partially-Connected Topology as an efficient solution
  - Routing strategy in such an irregular topology is the major problem

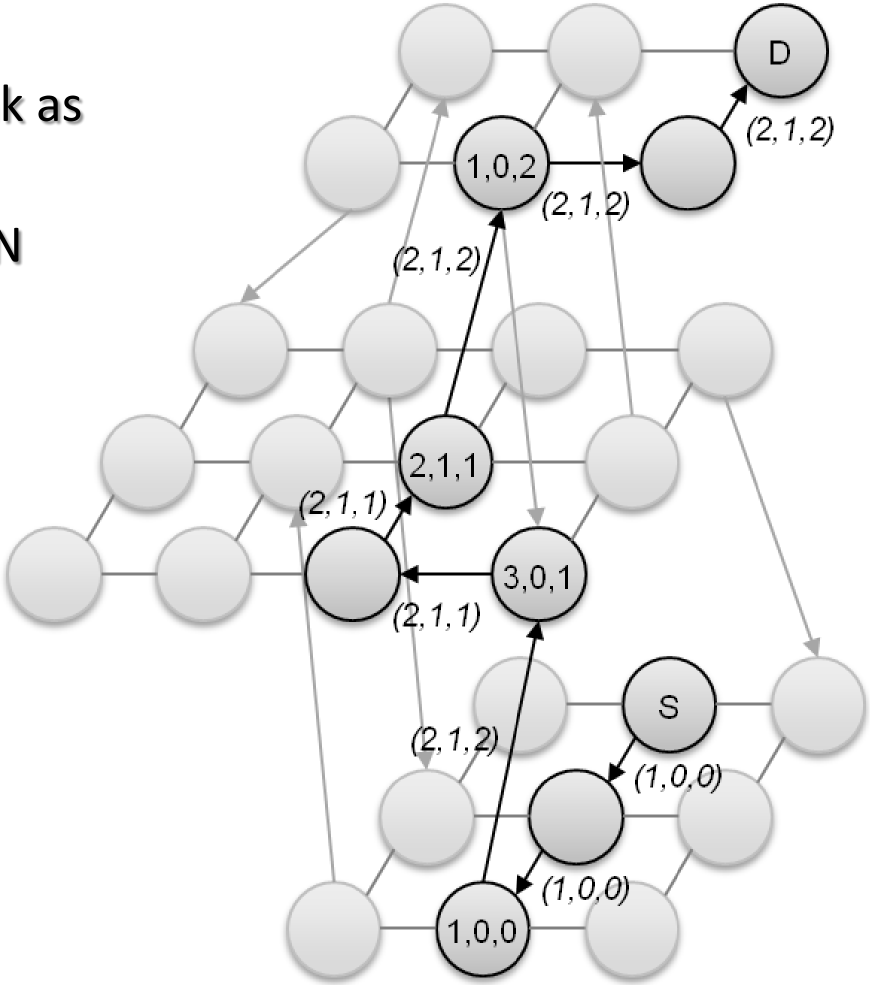






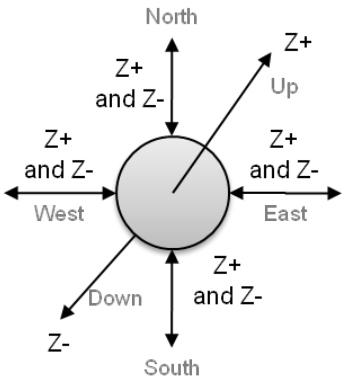
Elevator-First Routing Algorithm

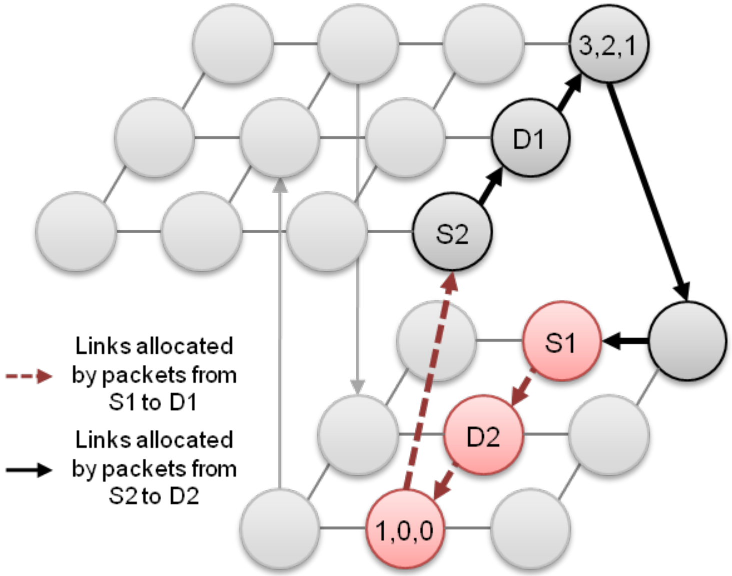
- Each router registers
  - A router in its layer with UP link as ascending elevator
  - A router in its layer with DOWN link as descending elevator



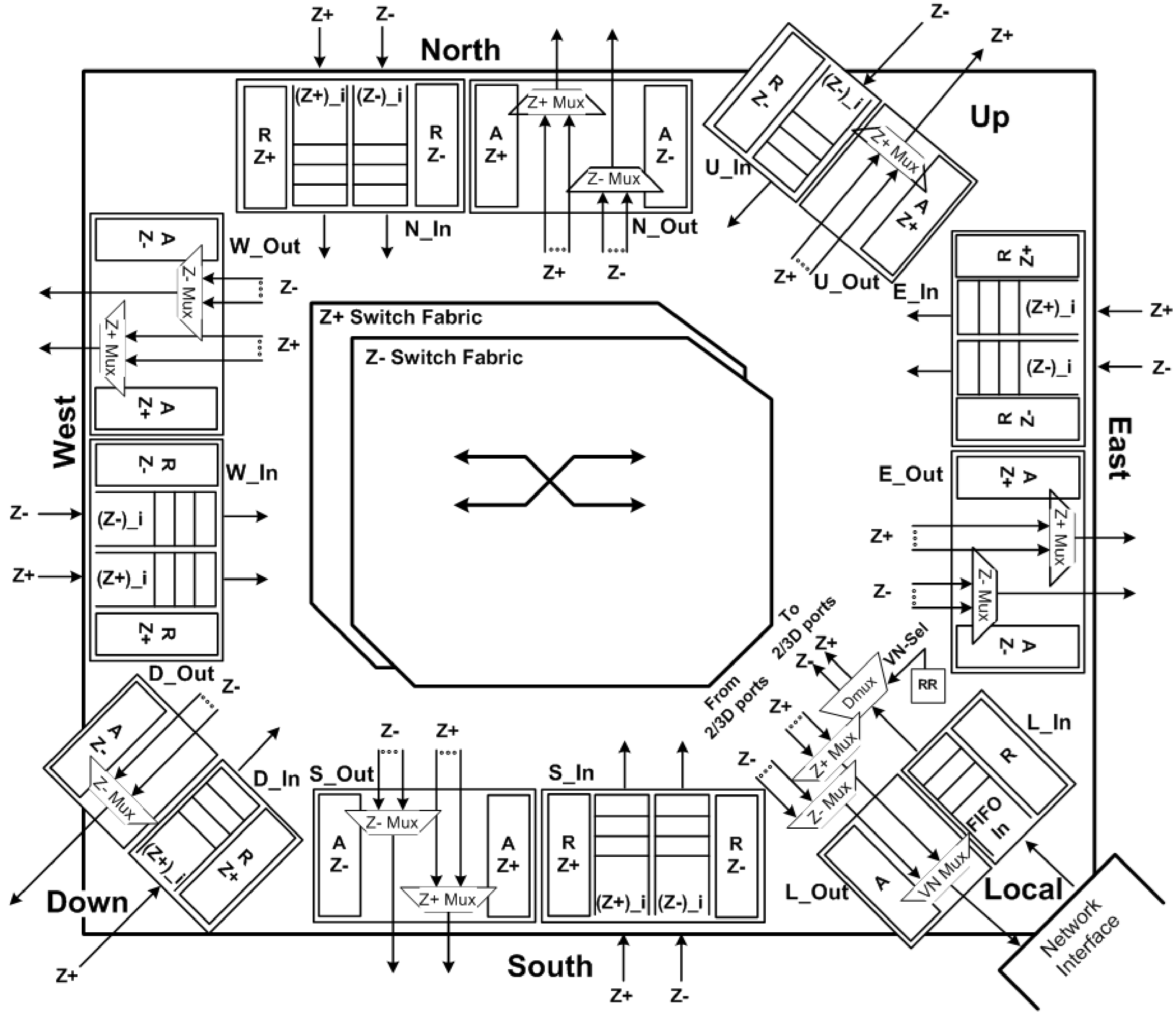
… and … Deadlock !

- Two Virtual Networks in the plane to avoid *deadlocks*
  - One for ascending packets (Z+)
  - One for descending packets (Z-)
  - Ascending vertical links are Z+
  - Descending vertical links are Z-

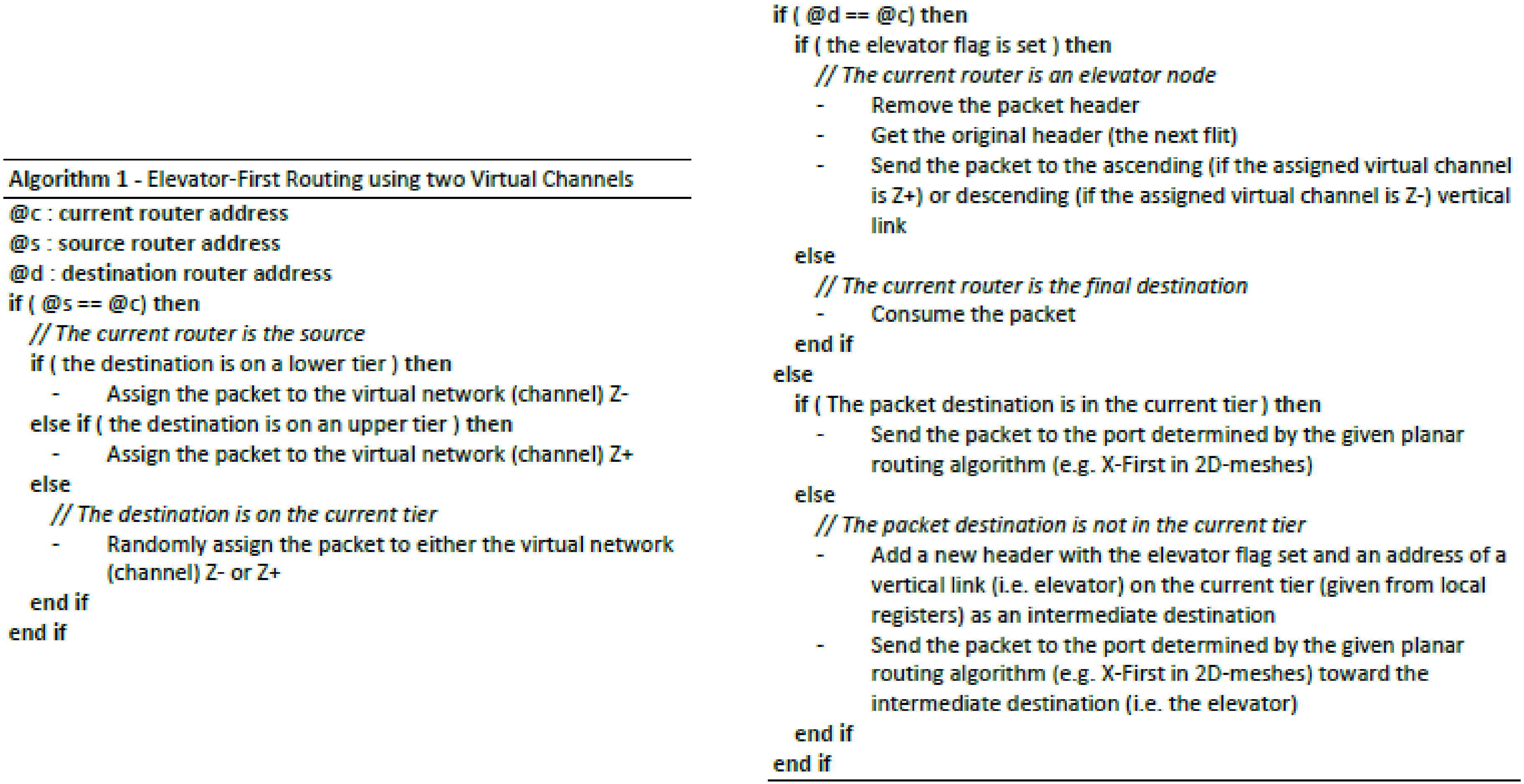



3,2,1
D1
S2
S1
D2
1,0,0
Links allocated by packets from S1 to D1
Links allocated by packets from S2 to D2

Elevator-First Router



The Algorithm

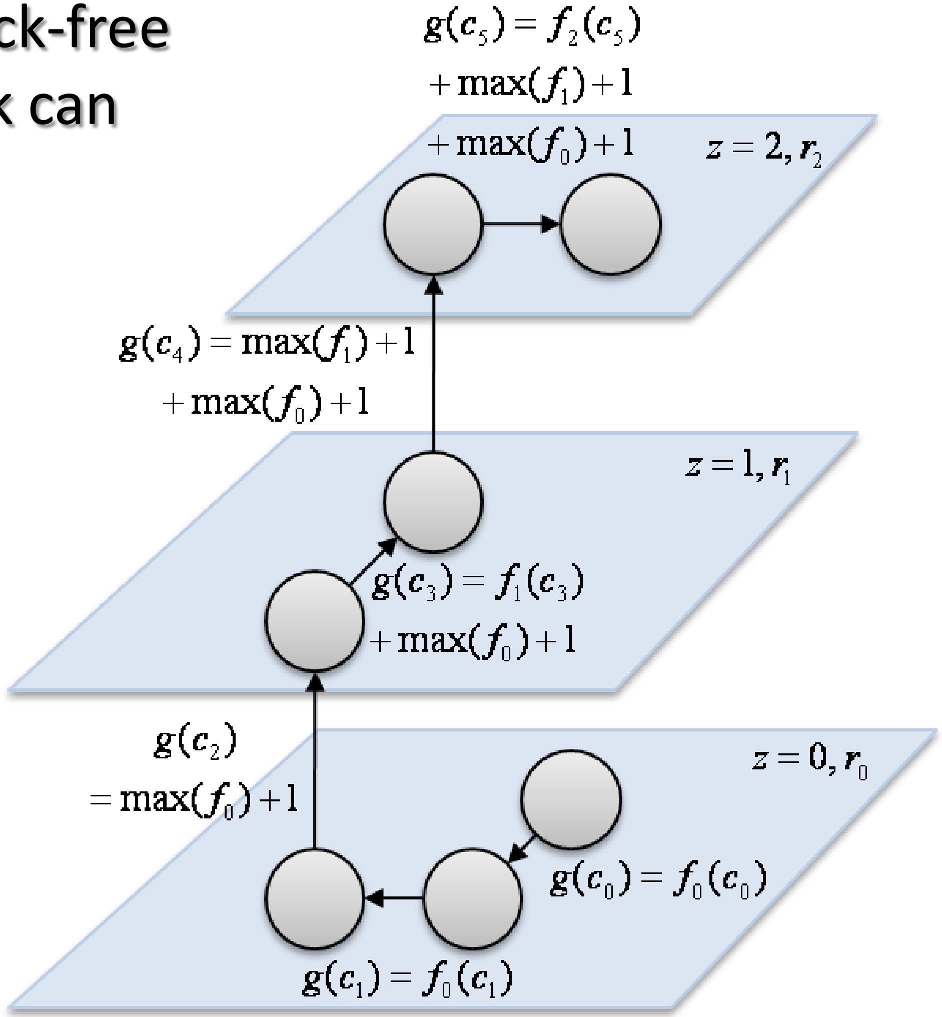


Algorithm 1 - Elevator-First Routing using two Virtual Channels

```
@c : current router address
@s : source router address
@d : destination router address
if ( @s == @c) then
   // The current router is the source
   if ( the destination is on a lower tier ) then
      -   Assign the packet to the virtual network (channel) Z-
   else if ( the destination is on an upper tier ) then
      -   Assign the packet to the virtual network (channel) Z+
   else
      // The destination is on the current tier
      -   Randomly assign the packet to either the virtual network
          (channel) Z- or Z+
   end if
end if
if ( @d == @c) then
   if ( the elevator flag is set ) then
      // The current router is an elevator node
      -   Remove the packet header
      -   Get the original header (the next flit)
      -   Send the packet to the ascending (if the assigned virtual channel
          is Z+) or descending (if the assigned virtual channel is Z-) vertical
          link
   else
      // The current router is the final destination
      -   Consume the packet
   end if
else
   if ( The packet destination is in the current tier ) then
      -   Send the packet to the port determined by the given planar
          routing algorithm (e.g. X-First in 2D-meshes)
   else
      // The packet destination is not in the current tier
      -   Add a new header with the elevator flag set and an address of a
          vertical link (i.e. elevator) on the current tier (given from local
          registers) as an intermediate destination
      -   Send the packet to the port determined by the given planar
          routing algorithm (e.g. X-First in 2D-meshes) toward the
          intermediate destination (i.e. the elevator)
   end if
end if
```

Formal Proof of Deadlock-Freedom

- A routing algorithm is deadlock-free if the channels in the network can be numbered such as every routing path uses strictly increasing (decreasing) channel numbers


$g(c_5) = f_2(c_5) + \max(f_1) + 1 + \max(f_0) + 1$
$z = 2, r_2$
$g(c_4) = \max(f_1) + 1 + \max(f_0) + 1$
$z = 1, r_1$
$g(c_3) = f_1(c_3) + \max(f_0) + 1$
$g(c_2) = \max(f_0) + 1$
$z = 0, r_0$
$g(c_0) = f_0(c_0)$
$g(c_1) = f_0(c_1)$

… an example …

b = max (n, m)

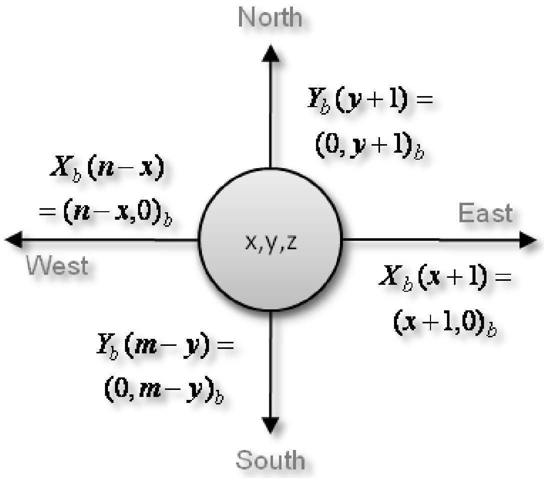
$$X_b : x \rightarrow (x,0)_b = x$$

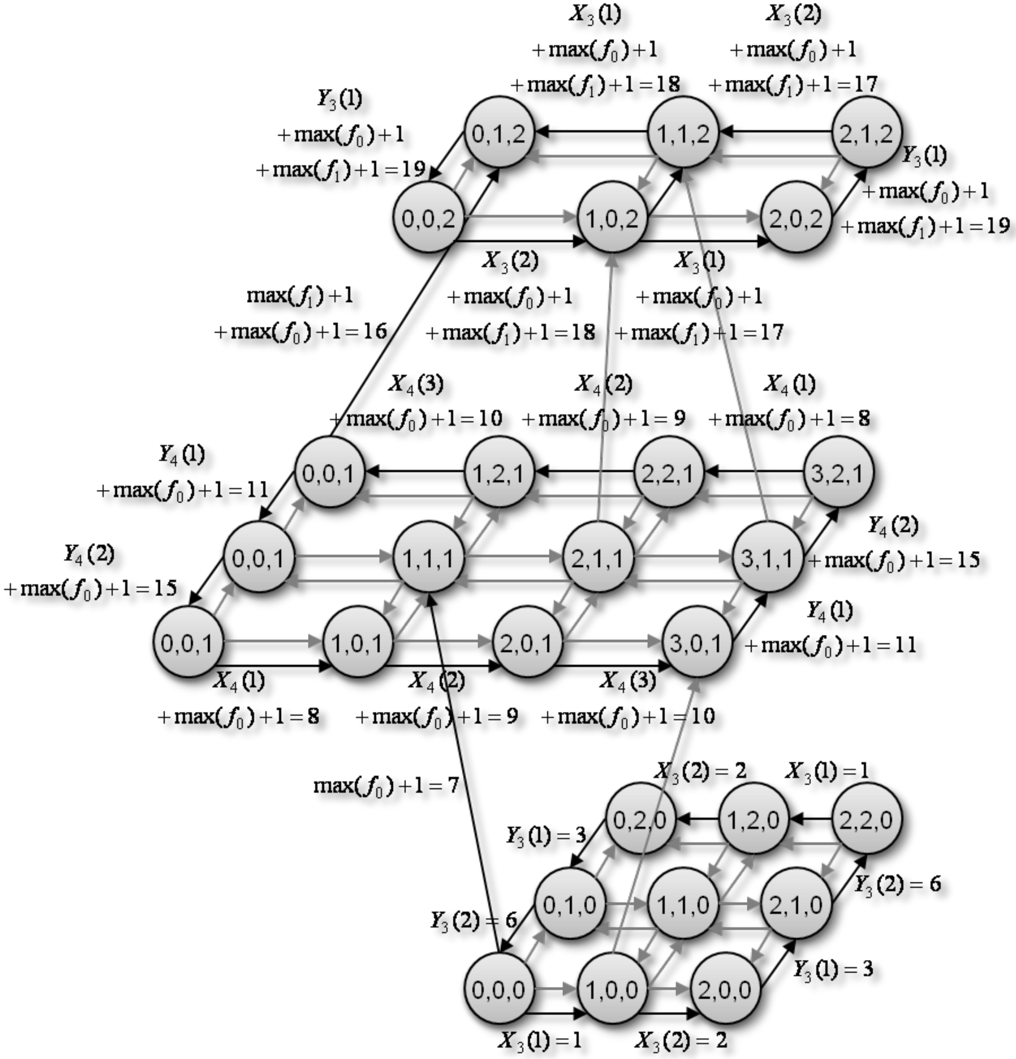
$$Y_b : y \rightarrow (0,y)_b = yb$$

$$f_z(c) \rightarrow \left| \begin{array}{l} X_b(n-x) \ \ if\ c = West \\ Y_b(y+1) \ \ if\ c = North \\ X_b(x+1) \ \ if\ c = East \\ Y_b(m-y) \ \ if\ c = South \end{array} \right.$$

$$\forall (x,y,z),$$

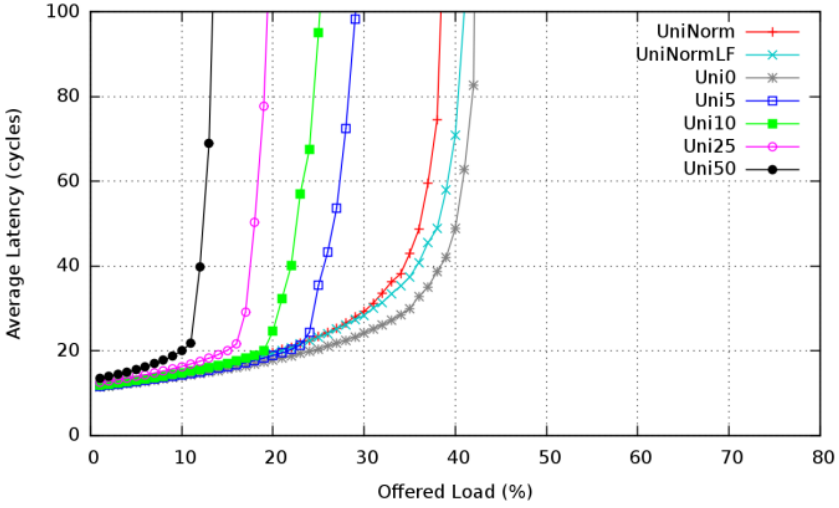
$$X_b(x) < X_b(x+1) < Y_b(y) < Y_b(y+1)$$

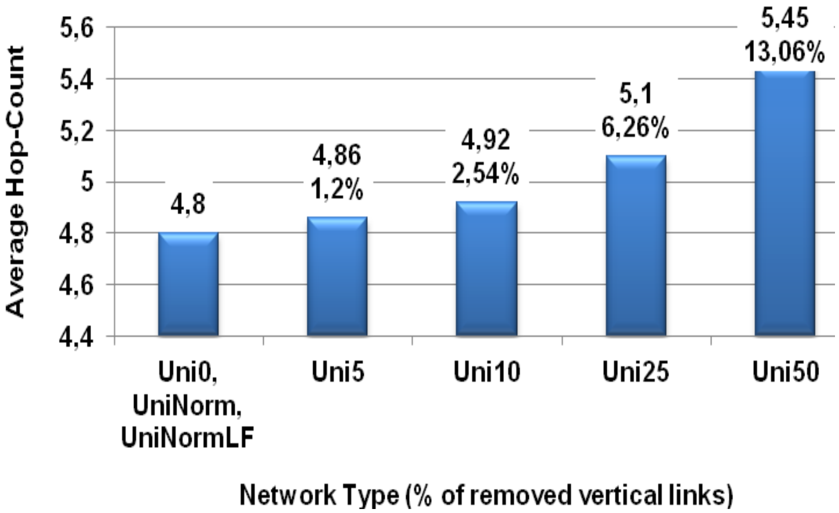


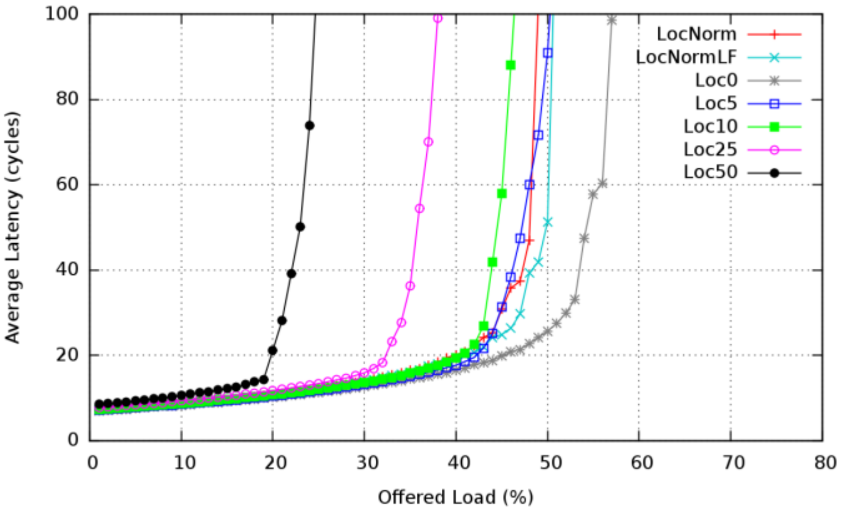


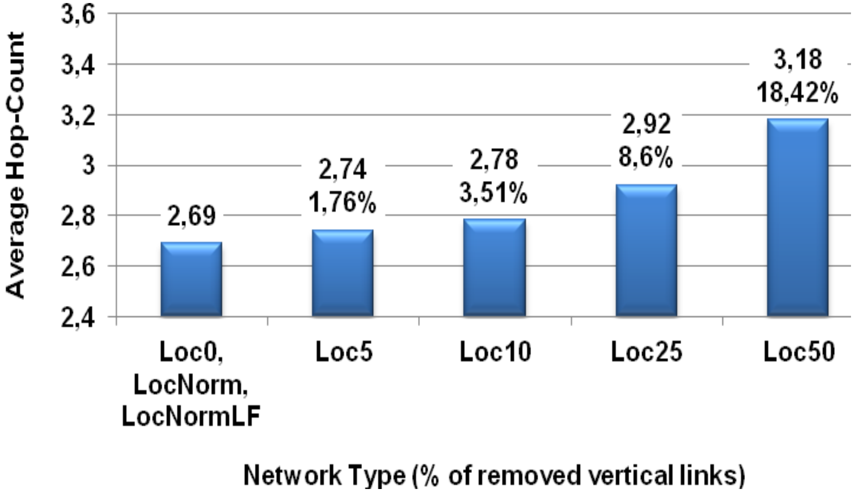
$X_3(1)$ $+\max(f_0)+1$ $+\max(f_1)+1=18$
$X_3(2)$ $+\max(f_0)+1$ $+\max(f_1)+1=17$
$Y_3(1)$ $+\max(f_0)+1$ $+\max(f_1)+1=19$
0,1,2
1,1,2
2,1,2
$Y_3(1)$ $+\max(f_0)+1$ $+\max(f_1)+1=19$
0,0,2
1,0,2
2,0,2
$X_3(2)$ $+\max(f_0)+1$ $+\max(f_1)+1=18$
$X_3(1)$ $+\max(f_0)+1$ $+\max(f_1)+1=17$
$\max(f_1)+1$ $+\max(f_0)+1=16$
$X_4(3)$ $+\max(f_0)+1=10$
$X_4(2)$ $+\max(f_0)+1=9$
$X_4(1)$ $+\max(f_0)+1=8$
$Y_4(1)$ $+\max(f_0)+1=11$
0,0,1
1,2,1
2,2,1
3,2,1
$Y_4(2)$ $+\max(f_0)+1=15$
0,0,1
1,1,1
2,1,1
3,1,1
$Y_4(2)$ $+\max(f_0)+1=15$
$Y_4(1)$ $+\max(f_0)+1=11$
0,0,1
1,0,1
2,0,1
3,0,1
$X_4(1)$ $+\max(f_0)+1=8$
$X_4(2)$ $+\max(f_0)+1=9$
$X_4(3)$ $+\max(f_0)+1=10$
$\max(f_0)+1=7$
$X_3(2)=2$
$X_3(1)=1$
$Y_3(1)=3$
0,2,0
1,2,0
2,2,0
$Y_3(2)=6$
0,1,0
1,1,0
2,1,0
$Y_3(2)=6$
0,0,0
1,0,0
2,0,0
$Y_3(1)=3$
$X_3(1)=1$
$X_3(2)=2$

Performance …







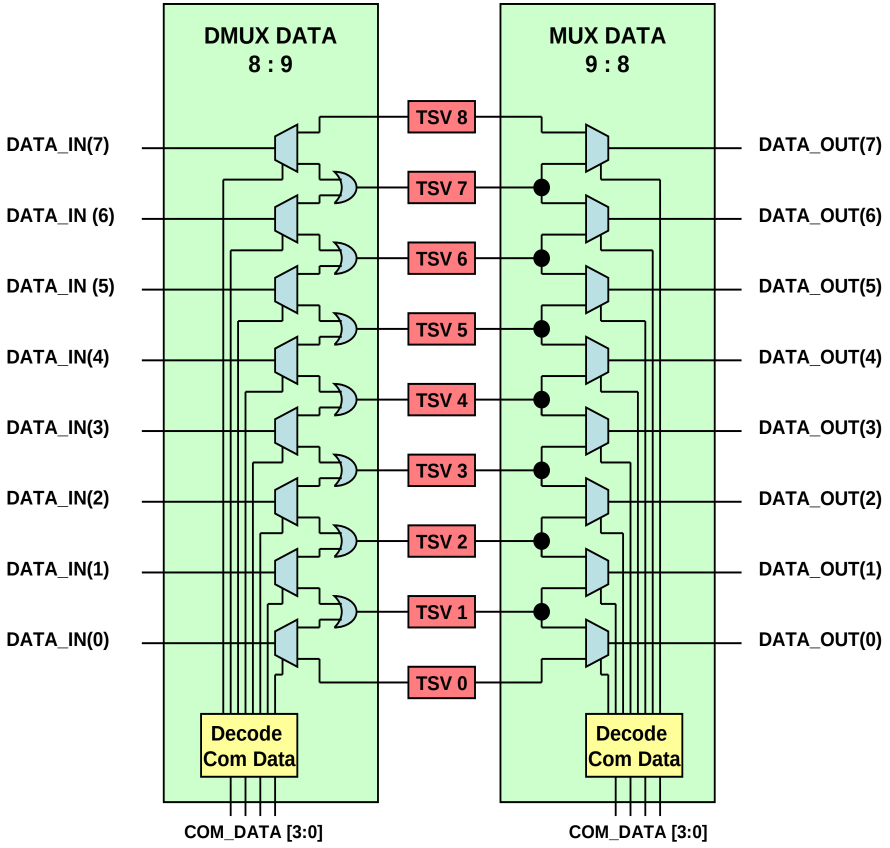


Outline

- 2D (Asynchronous) NoCs
- 3D-Integration and NoCs
  - Vertical Link Serialization
  - Memory on logic and 3D-NoC
  - Vertically-Partially-Connected 3D-NoC
  - A few open problems
- Conclusion

TSV failure tolerance

- Using spare TSV?
- Or by redirecting traffic into the network!
  - Provisions for network reconfiguration
  - Cost? Decision making?
  - Dynamicity?

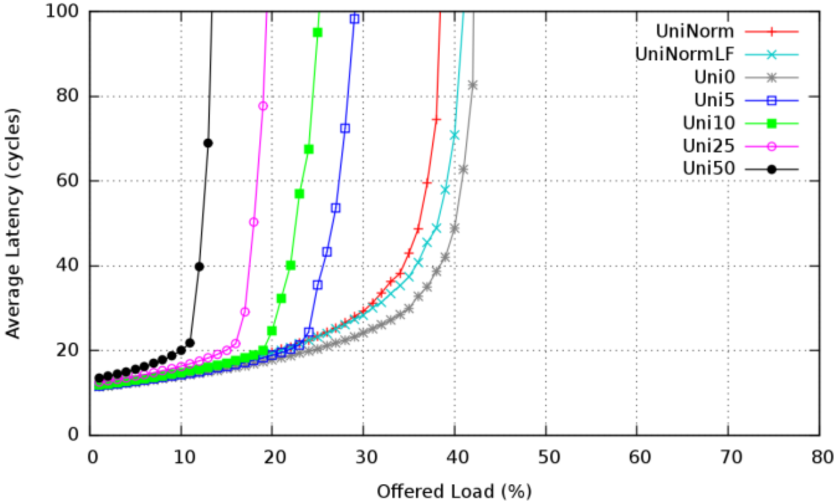


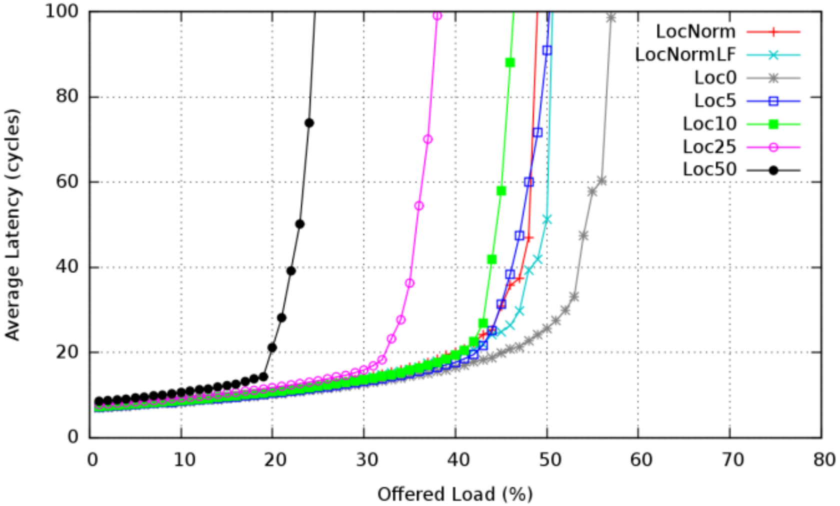
Vertically-Partially-Connected 3D-NoC

- Elevator placement:

  for a given traffic pattern/application domain, how many elevators should be used?

Example for uniform random and localized traffic





Very technology dependent:

- TSV (and associated circuit) size
- TSV throughput
- Yield

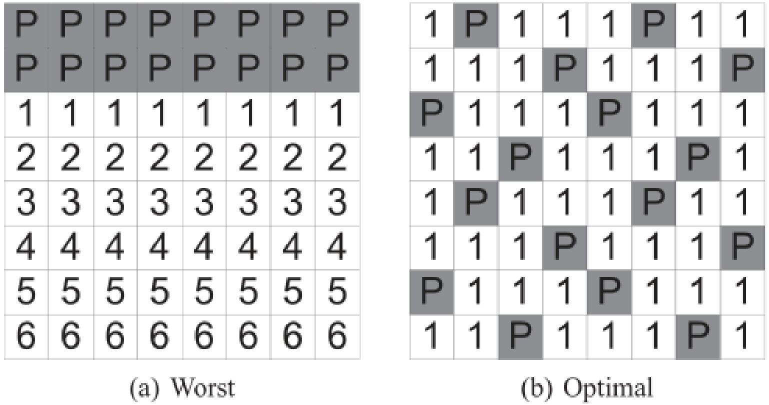
Vertically-Partially-Connected 3D-NoC

- Elevator placement:

  for a given traffic pattern/application domain, where should the elevators be placed?

Example for a 8x8 3D mesh, uniform random traffic, adaptive routing
Optimal minimizes hop-count, …
What about deterministic routing ? Hot-spot or domain specific traffic ?

Source: Xu *et al*



P	P	P	P	P	P	P	P
P	P	P	P	P	P	P	P
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6

(a) Worst

1	P	1	1	1	P	1	1
1	1	1	P	1	1	1	P
P	1	1	1	P	1	1	1
1	1	P	1	1	1	P	1
1	P	1	1	1	P	1	1
1	1	1	P	1	1	1	P
P	1	1	1	P	1	1	1
1	1	P	1	1	1	P	1

(b) Optimal

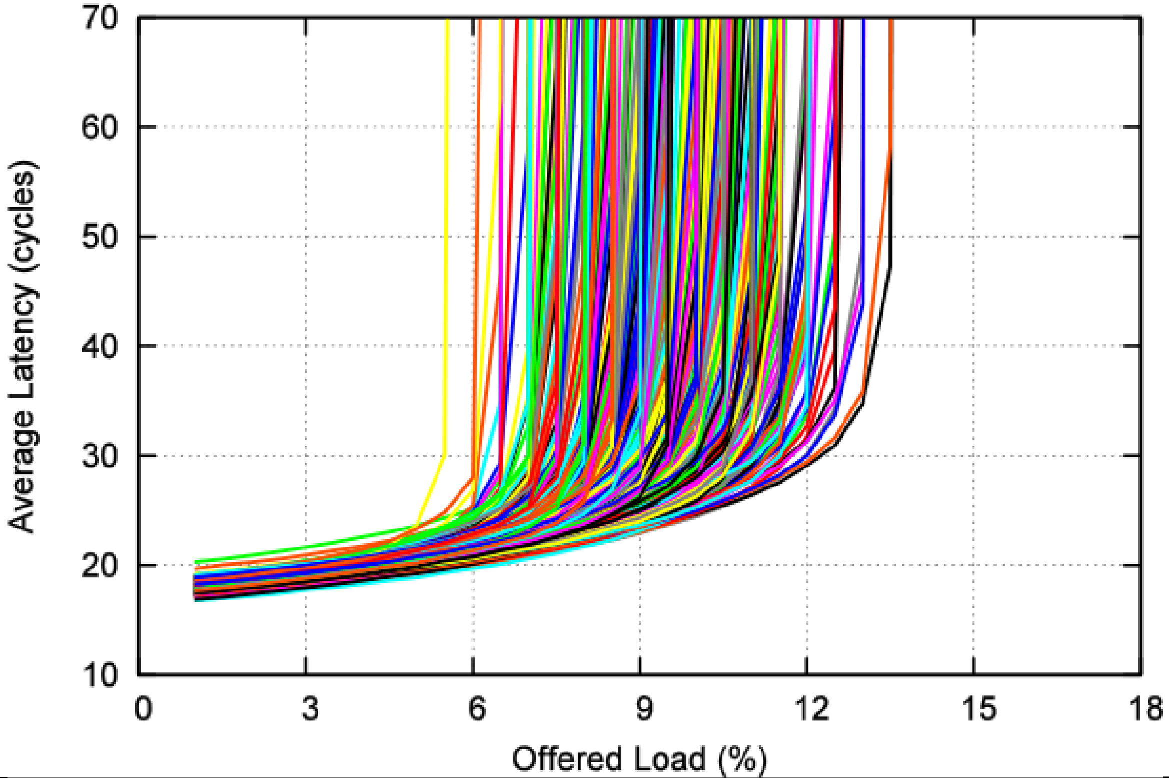
Placing 16 pillars on a 8x8 3D mesh

Vertically-Partially-Connected 3D-NoC

- Elevator assignment:
  for a given placement of the elevators, which elevator is to be assigned to any given node ?

Different random assignments for 50% elevators (uniform random traffic used)

Not just a theoretical question!



Conclusion

3D Interconnections are key to future SoC design

- Zillions of NoC papers in the last decade, but only now companies start using them, …
- Many, many 3D works in last years, but 2.5D still the current roadmap
- 3D technological options are still quite open

Source: ITRS'11 interconnect analysis

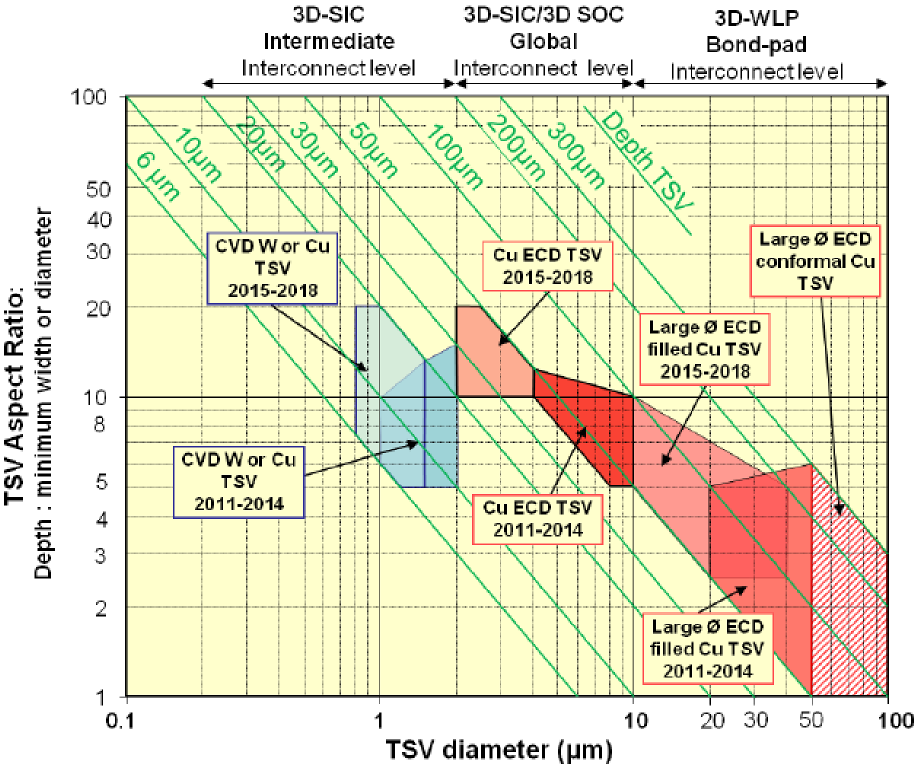


Figure INTC23 Cu and W-based TSV Options as a Function of TSV Diameter and Aspect Ratio, in Accordance with the 3D Interconnect Hierarchy and Roadmap

Conclusion

Let’s get prepared for 3D NoCs using technology independent approaches !

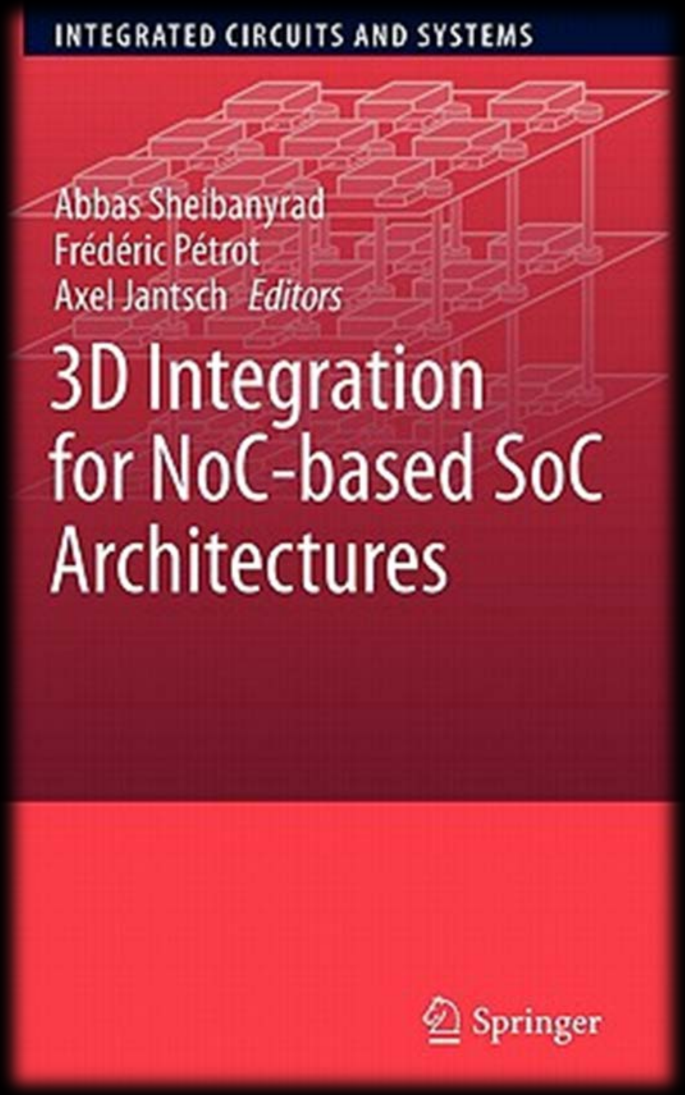
- QDI asynchronous circuits are robust
  - Well suited to temperature changes, technology variability, and aging
  - Design and test complexity is *the* issue
- Serialization limits the number of TSVs
  - Without timing performance degradation
  - With an acceptable area as compared to spared TSVs
  - Power consumption is however high

Conclusion

Let’s get prepared for 3D NoCs using technology independent approaches !

- Smart buffer usage enhances NoC behavior
  - Static distribution of buffering capacities useful for memory on logic case
  - What about optimizing sharing for dynamic behaviors?
- Vertically-Partially-Connected Topologies limit the number of TSVs
  - Deadlock free algorithms can be derived
  - Degrade performances
  - Optimizations on several aspects may limit degradation

Dankeschön, …



INTEGRATED CIRCUITS AND SYSTEMS

Abbas Sheibanyrad
Frédéric Pétrot
Axel Jantsch *Editors*

3D Integration for NoC-based SoC Architectures

Springer