Addressing heterogeneity, failures and variability in high-performance NoCs

José Duato

Parallel Architectures Group (GAP)
Technical University of Valencia (UPV)
Spain

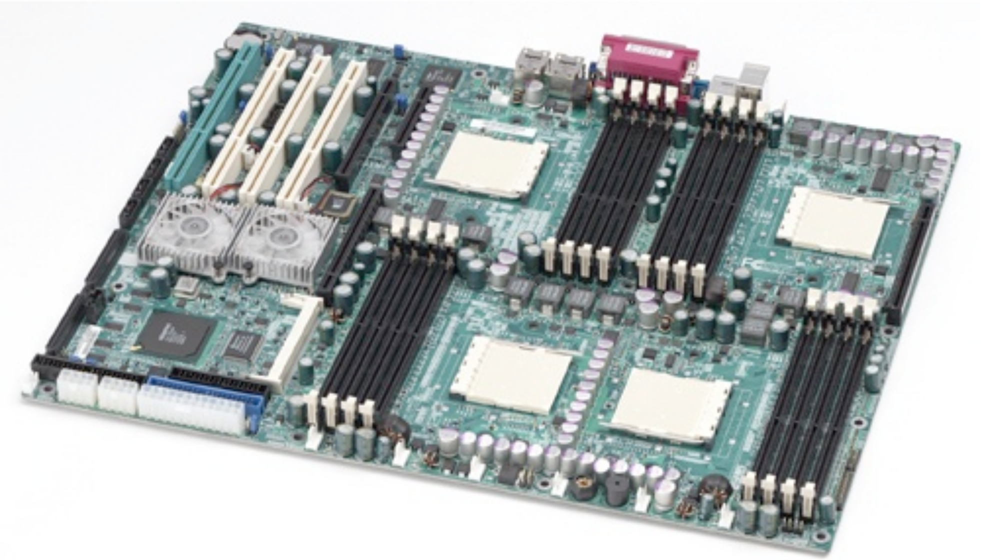
Outline

- Current proposals for NOCs
- Sources of heterogeneity
- Current designs
- Our proposal
- Addressing bandwidth constraints
- Addressing heat dissipation
- The role of HyperTransport and QPI
- Some current research efforts
- Conclusions

Current Server Configurations

- Cluster architectures based on 2- to 8-way motherboards with 4-core chips

Perspective View



What is next?

- Prediction is very difficult, especially about the future (Niels Bohr, physicist, 1885-1962)
- Extrapolating current trends, the number of cores per chip will increase at a steady rate
- Main expected difficulties
  - Communication among cores
    - Buses and crossbars do not scale
    - A Network on Chip (NoC) will be required

What is next?

- Main expected difficulties
  - Heat dissipation and power consumption
    - Known power reduction techniques already implemented in the cores
    - Either cores are simplified (in-order cores) or better heat extraction techniques are designed
  - Memory bandwidth and latency
    - VLSI technology scales much faster than package bandwidth
    - Multiple interconnect layers increase memory latency
    - Optical interconnects, proximity communication, and 3D stacking address this problem

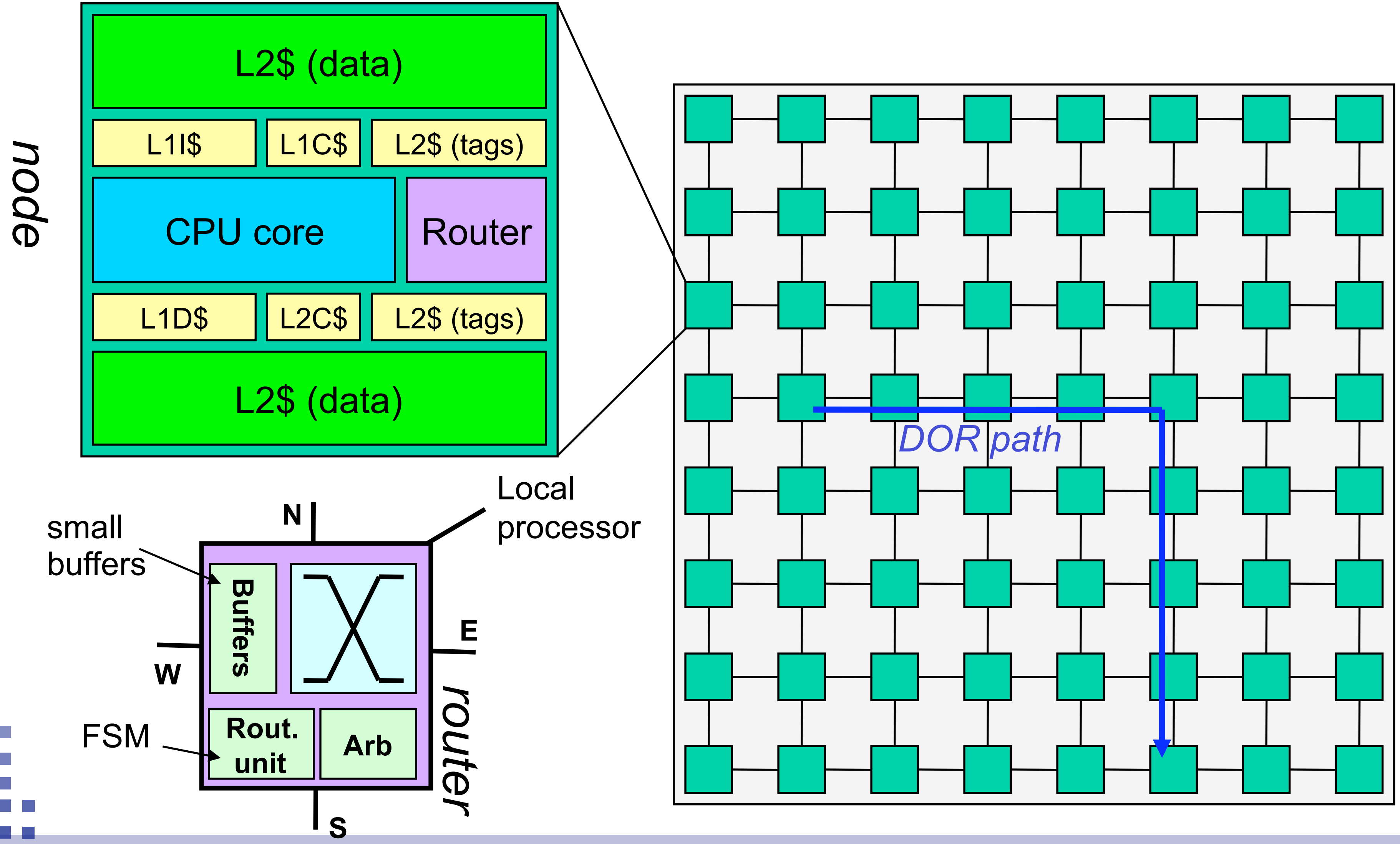
Most current proposals for NOCs…

- Homogeneous systems
  - Regular topologies and simple routing algorithms
  - Load balancing strategies become simpler
  - A single switch design for all the nodes
- Goals
  - Minimize latency
  - Minimize resource consumption (silicon area)
  - Minimize power consumption
  - Automate design space exploration

Most current proposals…

- Inherit solutions from first single-chip switches
  - Wormhole switching
    - ➢Low latency
    - ➢Small buffers (low area and power requirements)
  - 2D meshes
    - ➢Match the 2D layout of current chips
    - ➢Minimize wiring complexity
  - Dimension-order routing
    - ➢Implemented with a finite-state machine (low latency, small area)

Most current proposals…



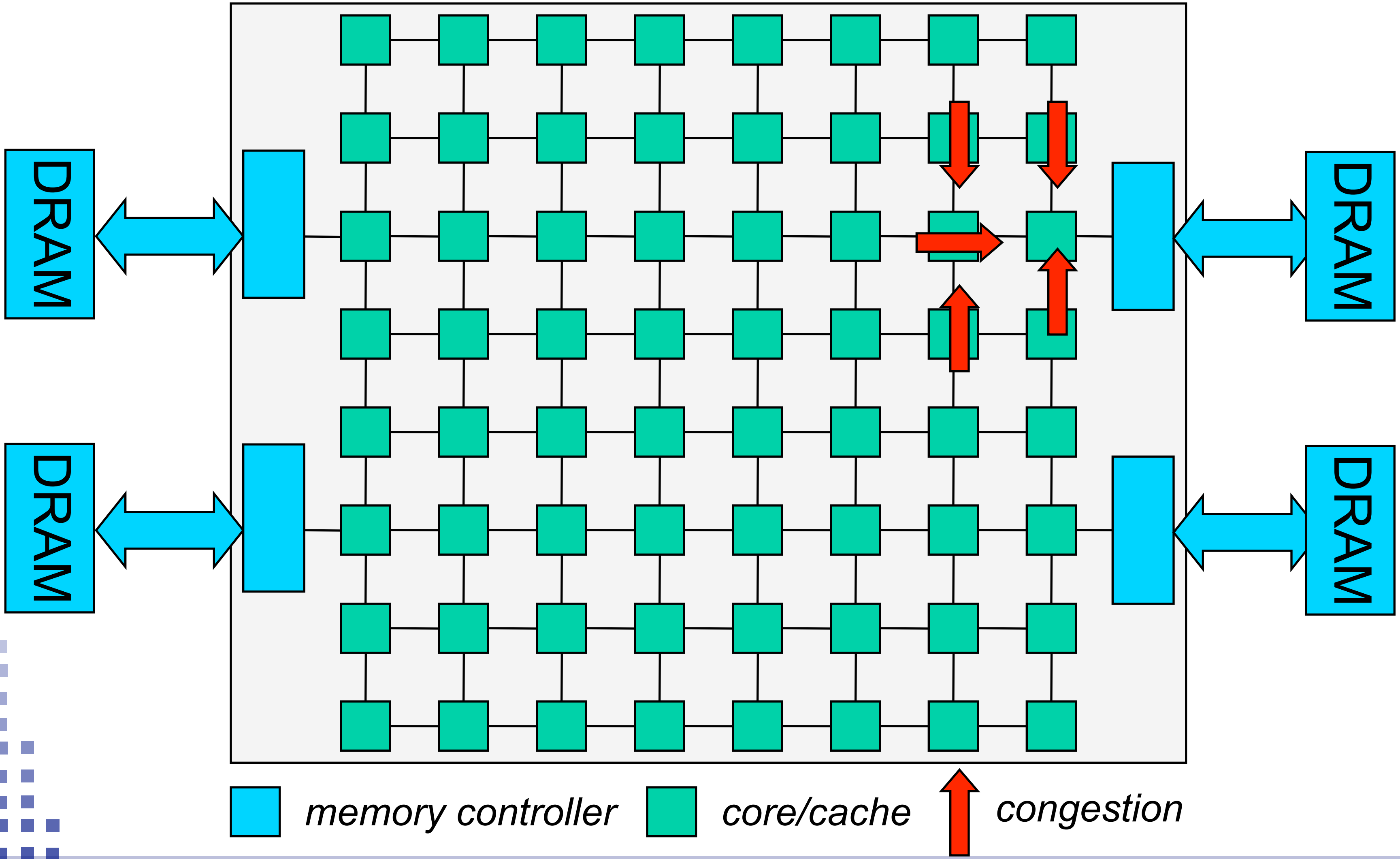
Sources of Heterogeneity

- Architectural sources
  - Access to external memory
  - Devices with different functionalities
  - Use of accelerators
  - Simple and complex cores
- Technology sources
  - Manufacturing defects
  - Manufacturing process variability
  - Thermal issues
  - 3D stacking
- Usage model sources
  - Virtualization
  - Application specific systems

Architectural sources

- Due to the existence of different kinds of devices
- Access to external memory
  - On-chip memory controllers
  - Different number of cores and memory controllers
    - ➢Example: GPUs with hundreds of cores and less than ten memory controllers

Architectural sources



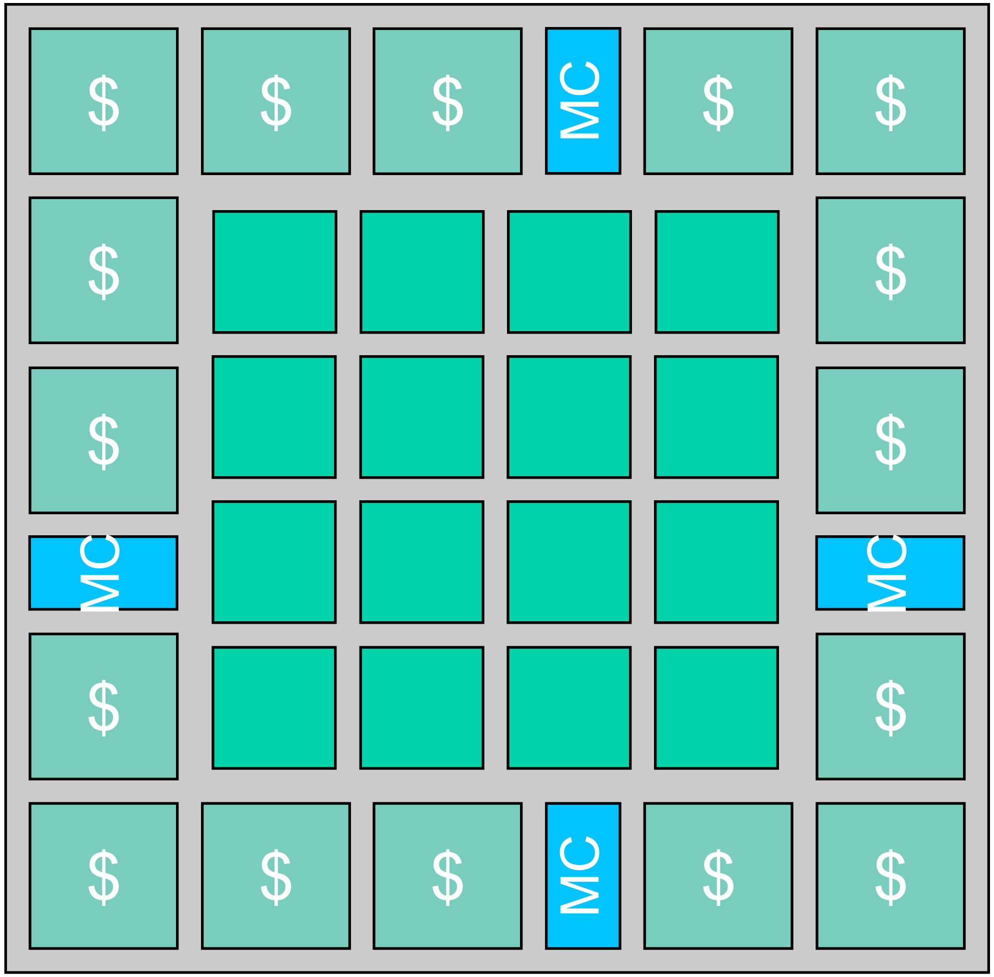
Architectural sources

- Access to external memory
  - On-chip memory controllers
  - Different number of cores and memory controllers
    - Example: GPUs with hundreds of cores and less than ten memory controllers
  - Consequences
    - Heterogeneity in the topology
    - Asymmetric traffic patterns
    - Congestion when accessing memory controllers

Architectural sources

- Devices with different functionalities
  - Cache blocks with different sizes and shapes (than processor cores)

Architectural sources



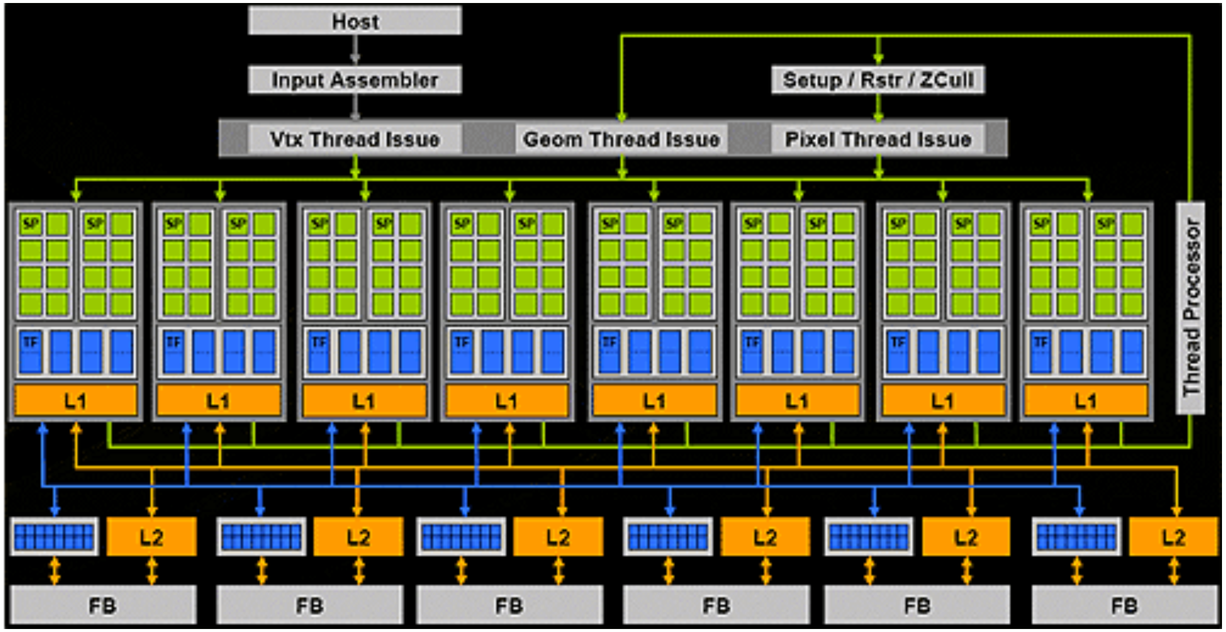
Architectural sources

- Devices with different functionalities
  - Cache blocks with different sizes and shapes (than processor cores)
  - Consequences
    - ➢Heterogeneity in the topology
    - ➢Asymmetric traffic patterns
      - Different link bandwidths might be required
      - Different networks might be required (e.g. 2D mesh + binary tree)

Architectural sources

- Using accelerators
  - Efficient use of available transistors
  - Increases the Flops/Watt ratio
  - Next device: GPU (already planned by AMD)

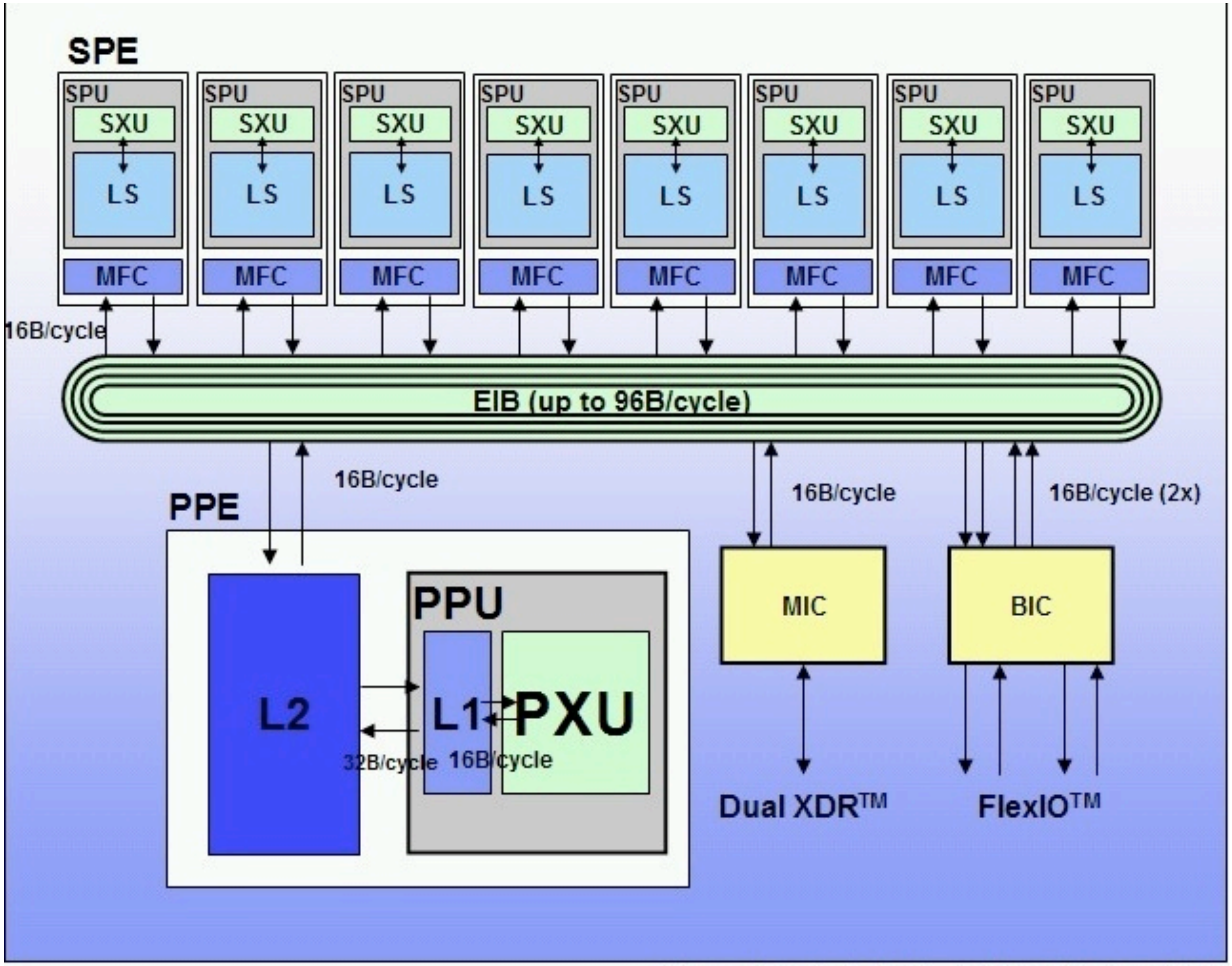
Architectural sources



Architectural sources

- Using accelerators
  - Efficient use of available transistors
  - Increases the Flops/Watt ratio
  - Next device: GPU (already planned by AMD)
- Simple and complex cores
  - Few complex cores to run sequential applications efficiently
  - Simple cores to run parallel applications and increase Flops/watt ratio
  - Example: Cell processor

Architectural sources

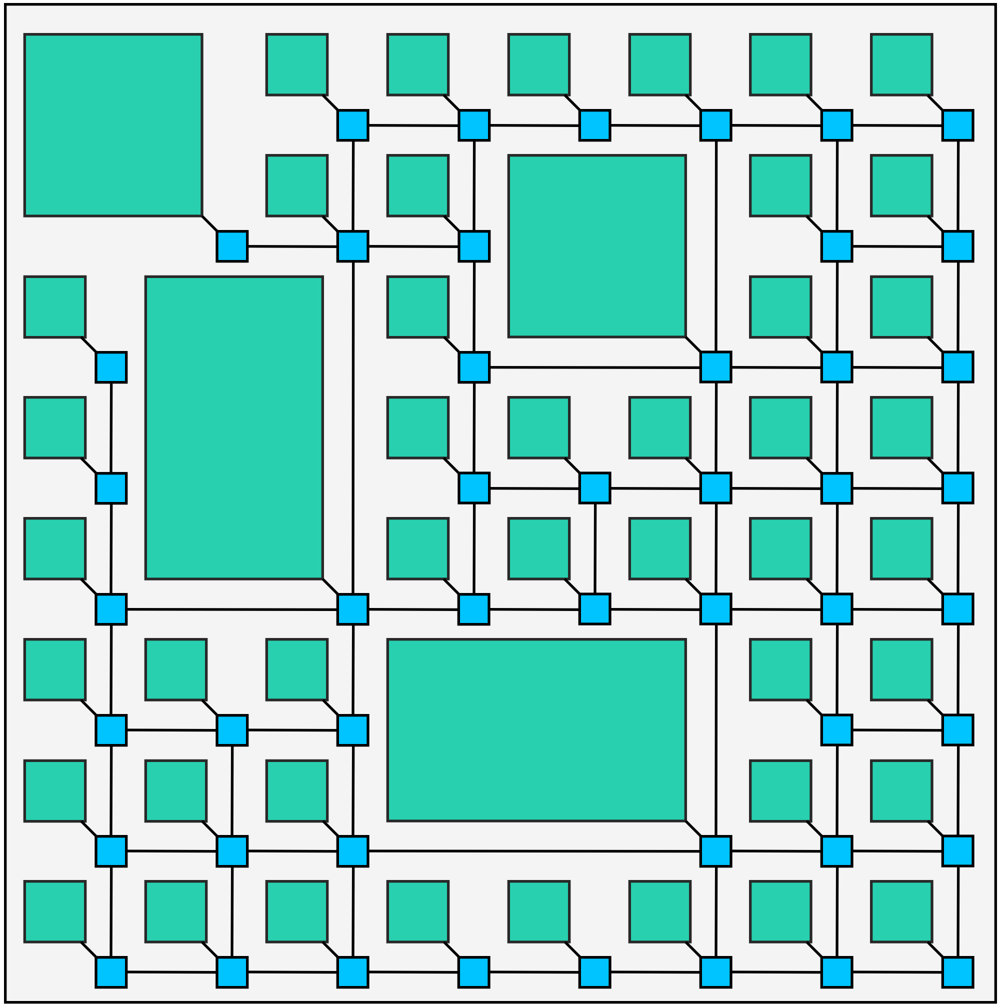


Source: M. Gschwind et al., Hot Chips-17, August 2005

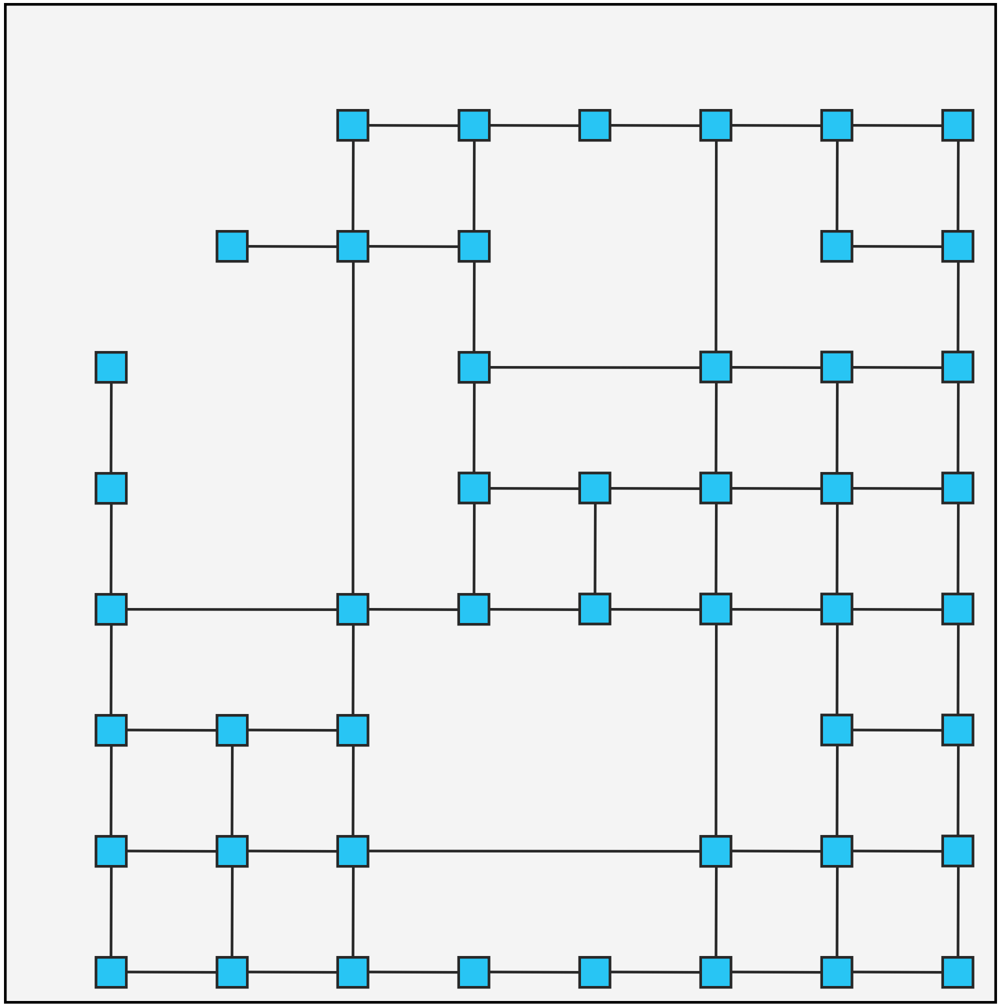
Architectural sources

- Using accelerators
  - Efficient use of available transistors
  - Increases the Flops/Watt ratio
  - Next device: GPU (already planned by AMD)
- Simple and complex cores
  - Few complex cores to run sequential applications efficiently
  - Simple cores to run parallel applications and increase Flops/watt ratio
  - Example: Cell processor
- Consequences
  - Heterogeneity in the topology (different sizes)
  - Asymmetric traffic patterns

Architectural sources



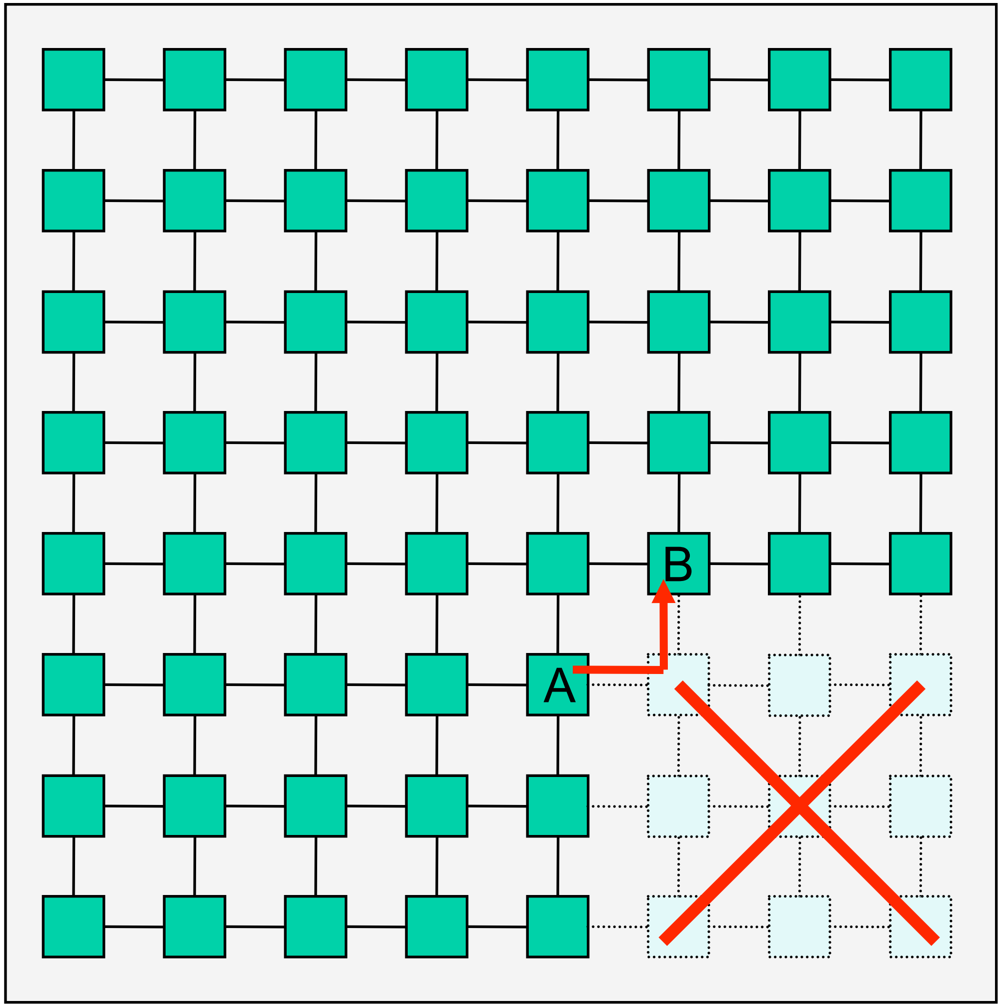
Architectural sources



Technology sources

- Manufacturing defects
  - Increase with integration scale
  - Yield may drop unless fault tolerance solutions are provided
  - Solution: use alternative paths (fault tolerant routing)
- Consequences
  - Asymmetries introduced in the use of links (deadlock issues)

Technology sources

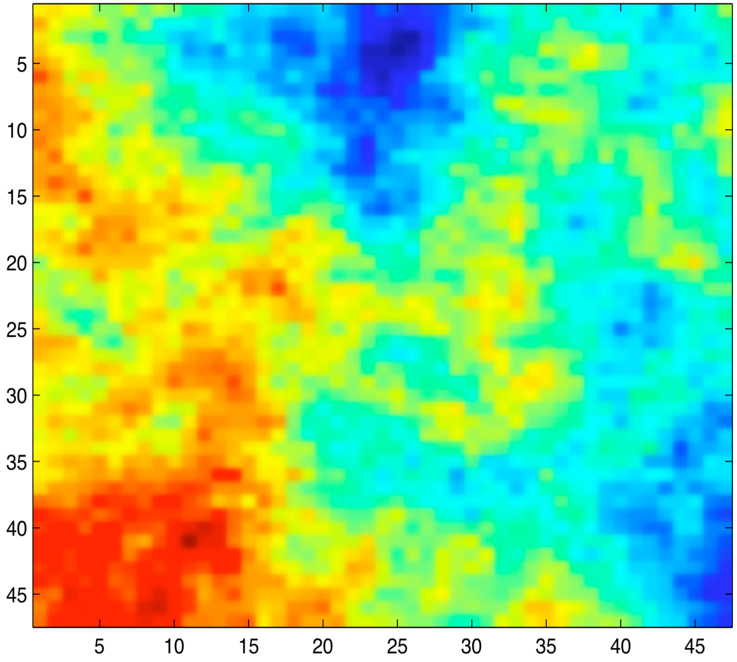


Technology sources

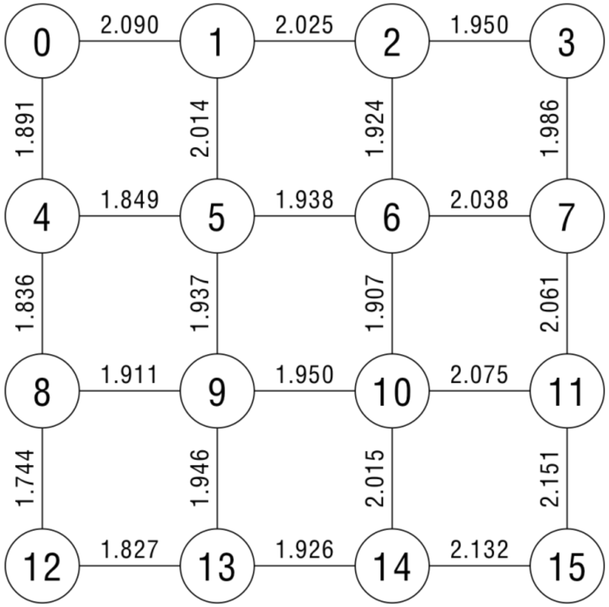
- Manufacturing process variability
  - Smaller transistor size ⇨ Increased process variability
    - Variations in L_{eff} ⇨ Variations in V_{th}
    - Variations in wire dimensions ⇨ Resistance and capacitance variations
    - Variations in dopant levels ⇨ Variations in V_{th}
  - Clock frequency fixed by slowest device
  - Unacceptable as variability increases

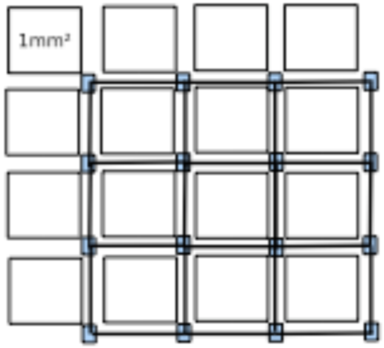
Systematic front-end variability

- Link frequency/delay distributions in NoC topology (32nm 4x4 NoC)



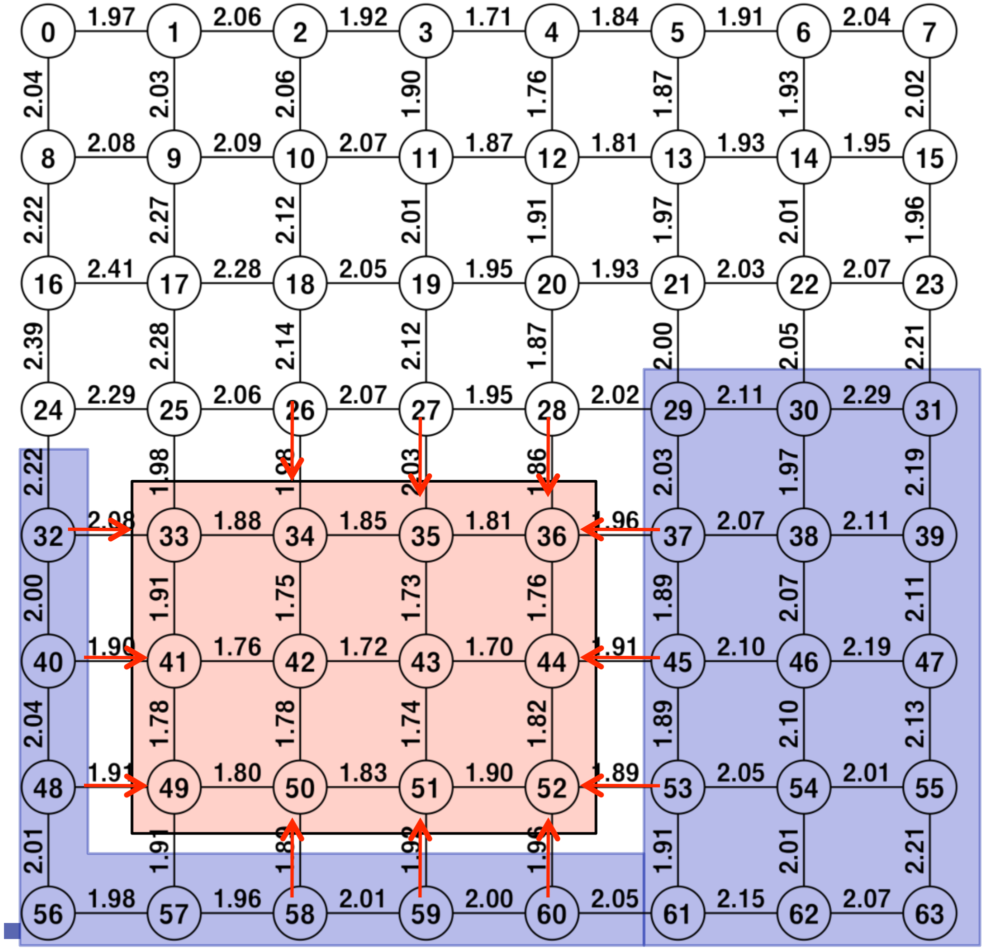
Lgate map





Systematic front-end variability

- Link frequency/delay distributions in NoC topology (32nm 8x8 NoC)



- In larger networks the scenario is even worse

Technology sources

- Manufacturing process variability
  - Possible solutions
    - Different regions with different speeds
    - Links with different speeds
    - Disabled links and/or switches
  - Consequences
    - Unbalanced link utilization
    - Irregular topologies

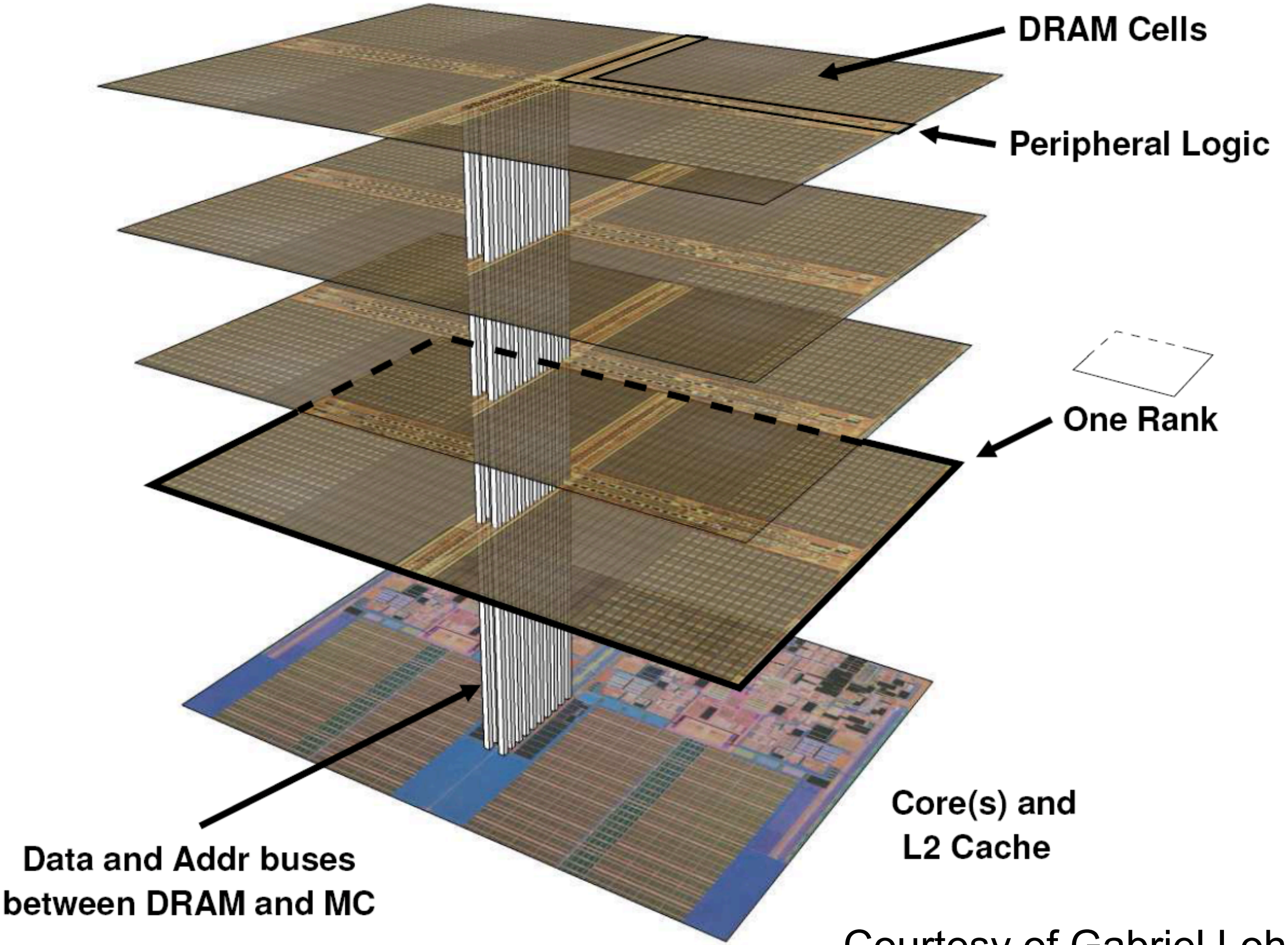
Technology sources

- Thermal issues
  - More transistors integrated as long as they are not active at the same time
  - Temperature controllers will dynamically adjust clock frequency for different clock domains
- Consequences
  - Functional heterogeneity
  - Performance drops due to congested (low bandwidth) subpaths (passing through slower regions)

Technology sources

- 3D stacking
  - The most promising technology to alleviate the memory bandwidth problem
  - Will aggravate the temperature problem (heat dissipation)
- Consequences
  - Traffic asymmetries (# vias vs wires in a chip)

Technology sources

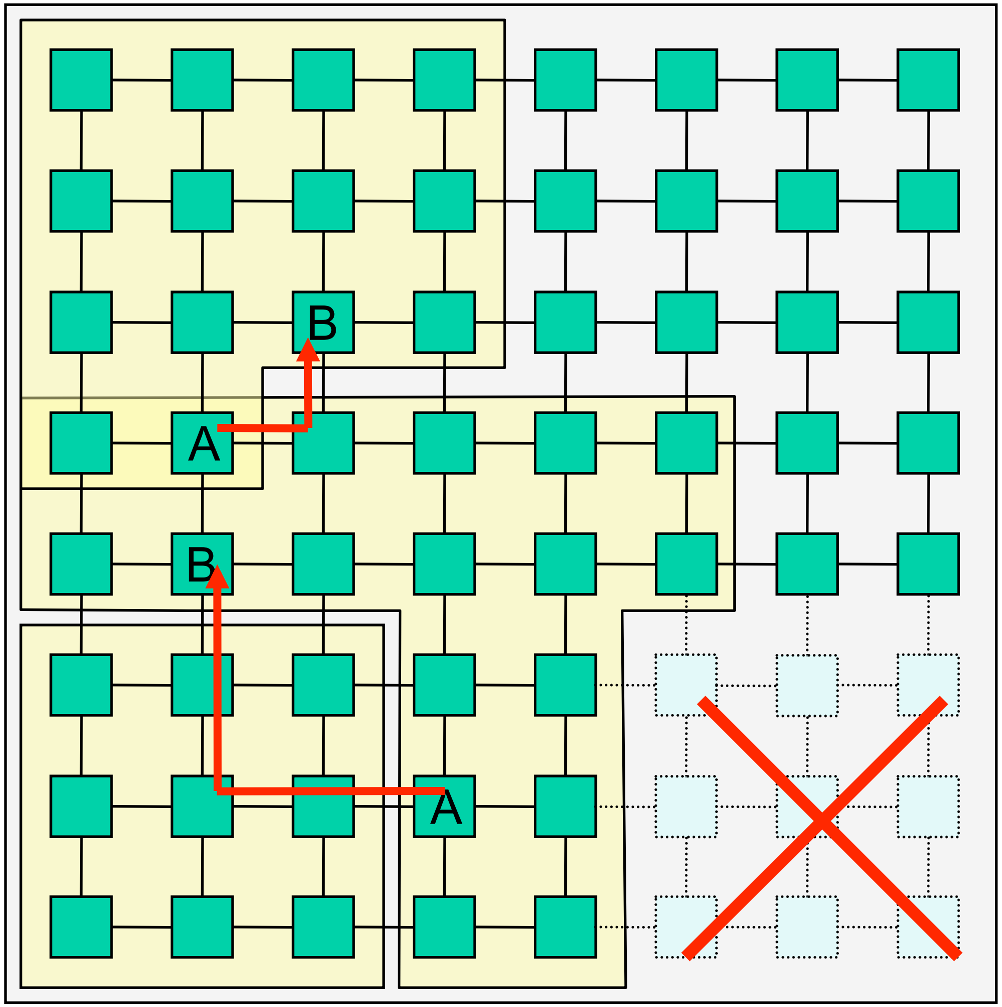


Courtesy of Gabriel Loh, ISCA 2008

Usage Model sources

- Virtualization
  - Enables running applications from different customers in the same computer while guaranteeing security and resource availability
  - Resources dynamically assigned (increases utilization)
  - At the on-chip level
    - Traffic isolation between regions
      - Deadlock issues (routing becomes complex)
      - Shared caches introduce interferences among regions
      - Memory controllers need to be shared

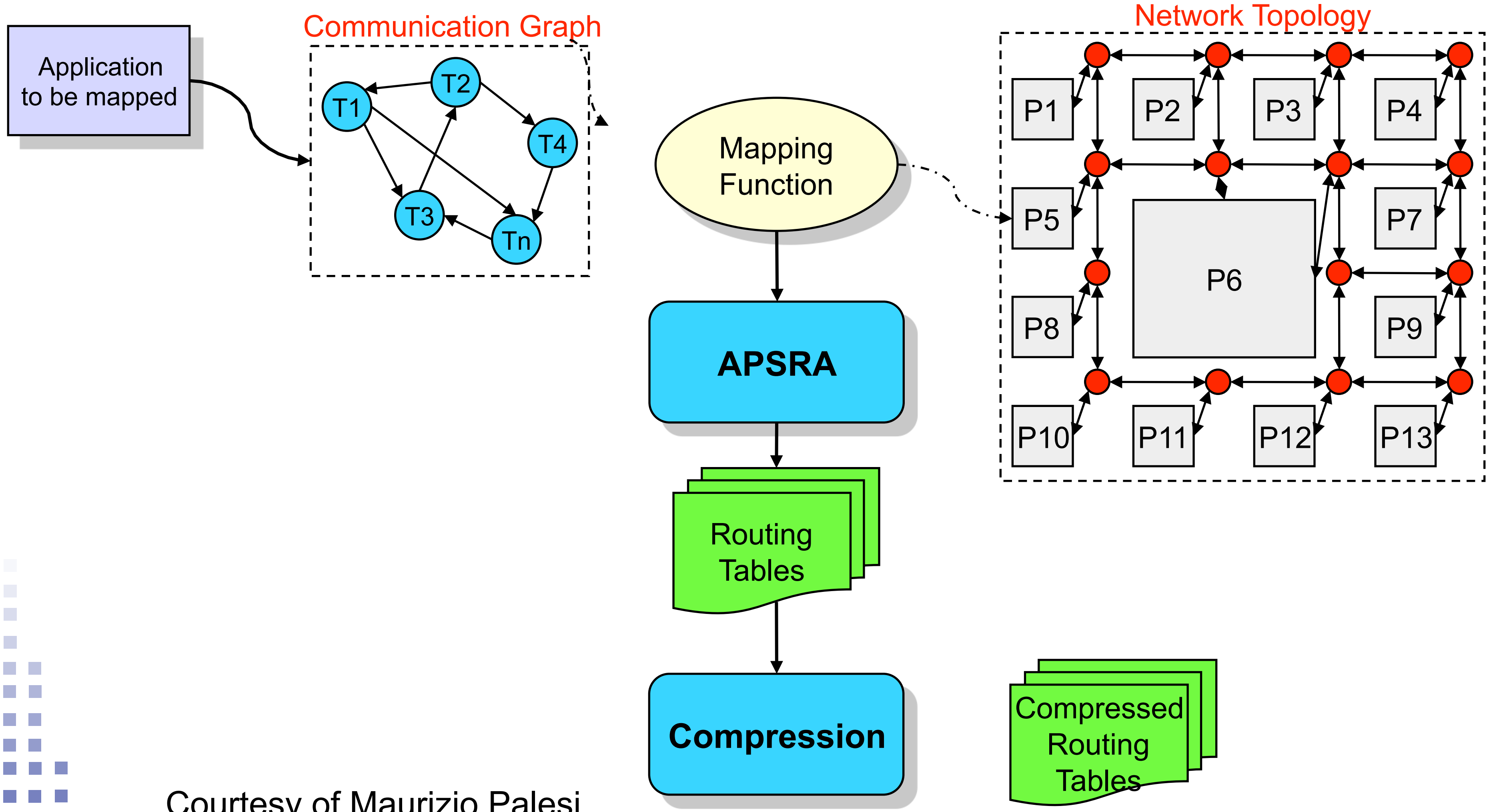
Usage Model sources



Usage Model sources

- Application specific systems
  - The application to run is known beforehand (embedded systems)
    - Non-uniform traffic and some links may not be required
  - Heterogeneity can lead to silicon area and power savings

Usage Model sources



Application to be mapped

Communication Graph

Mapping Function

APSRA

Routing Tables

Compression

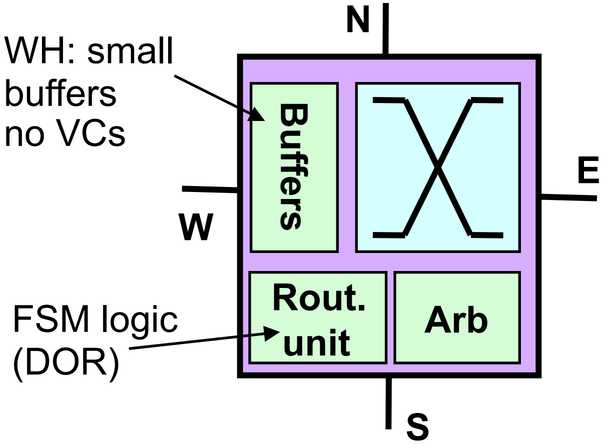
Compressed Routing Tables

Network Topology

Courtesy of Maurizio Palesi
and Shashi Kumar, CODES 2006

Current Designs

- Current trends when designing the NoC
  - Topology: 2D mesh (fits the chip layout)
  - Switching: wormhole (minimum area requirements for buffers)
  - Routing: implemented with logic (FSM finite-state-machine), DOR
    - ➢ Low latency, area and power efficient
    - ➢ But, … not suitable for new challenges
      - Manufacturing defects
      - Virtualization
      - Collective communication
      - Power management



Our Proposal

- bLBDR (Broadcast Logic-Based Distributed Routing)
  - Removes routing tables both at source nodes and switches
  - Enables
    - ➢FSM-based (low-latency, power/area efficient) unicast routing
    - ➢Tree-based multicast/broadcast routing with no need for tables
    - ➢Most irregular topologies (i.e. for manufacturing defects) are supported
      - Most topology-agnostic routing algorithms supported (up*/down*, SR)
      - DOR routing in a 2D mesh topology is supported
    - ➢Definition of multiple regions for virtualization and power management

System environment

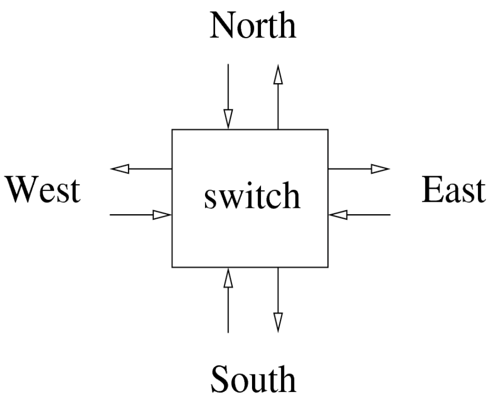
- For bLBDR to be applied, some conditions must be met:
  - Message headers must contain X and Y offsets, and every switch must know its own coordinates
  - Every end node can communicate with any other node through a minimal path
- bLBDR, on the other hand:
  - Can be applied on systems with or without virtual channels
  - Supports both wormhole and virtual cut-through switching

Our Proposal

– FSM-based implementation

- A set of AND, OR, NOT gates
- 2 flags per switch output port for routing
- An 8-bit register per switch output port for topology/regions definition

Description



payload	header	
	Ydst	Xdst

Routing and connectivity bits required per switch (12 bits, 3 per output port)

bits at the North port: Rne, Rnw

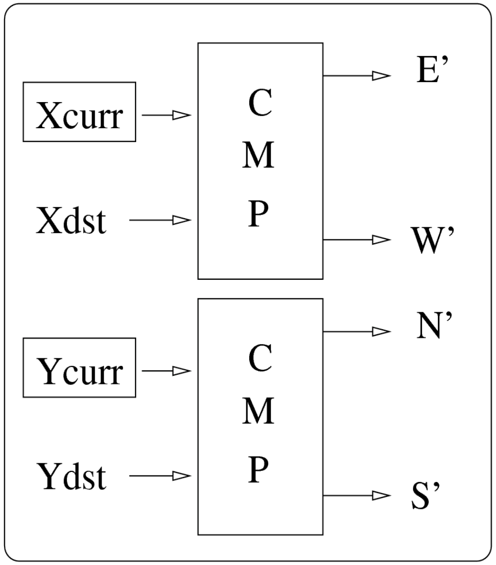
bits at the West port: Rwn, Rws

bits at the East port: Ren, Res

bits at the South port: Rse, Rsw

Cn, Ce, Cw, Cs

First part of the routing logic



Second part of the routing logic

$$N'' = N' \,.\, \overline{E'} \,.\, \overline{W'} + N' \,.\, E' \,.\, Rne + N' \,.\, W' \,.\, Rnw$$

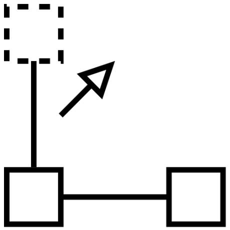
$$E'' = E' \,.\, \overline{N'} \,.\, \overline{S'} + E' \,.\, N' \,.\, Ren + E' \,.\, S' \,.\, Res$$

$$W'' = W' \,.\, \overline{N'} \,.\, \overline{S'} + W' \,.\, N' \,.\, Rwn + W' \,.\, S' \,.\, Rws$$

$$S'' = S' \,.\, \overline{E'} \,.\, \overline{W'} + S' \,.\, E' \,.\, Rse + S' \,.\, W' \,.\, Rsw$$

$N = N'' \,.\, Cn$ $\qquad$ $W = W'' \,.\, Cw$

$E = E'' \,.\, Ce$ $\qquad$ $S = S'' \,.\, Cs$



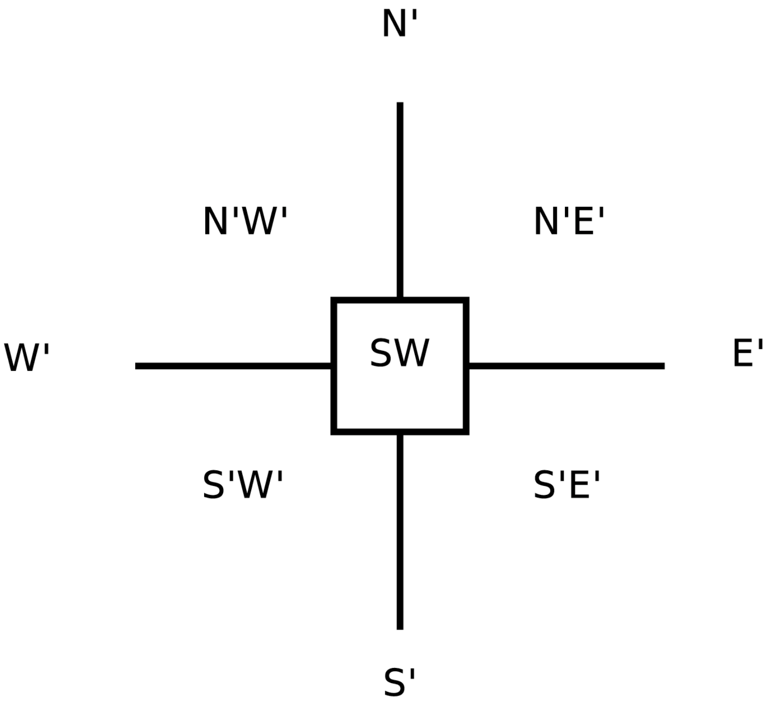
Rne = 0

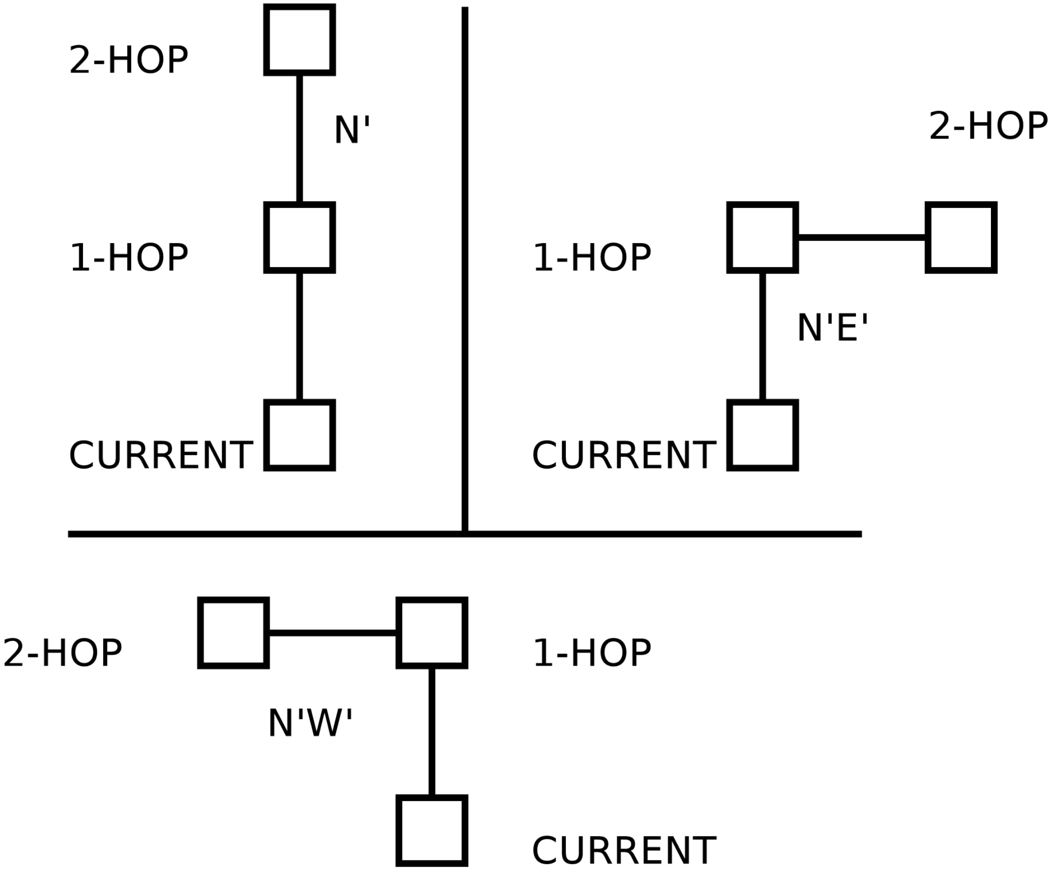
Rnw = 1

Cn = 1

Cw = 0

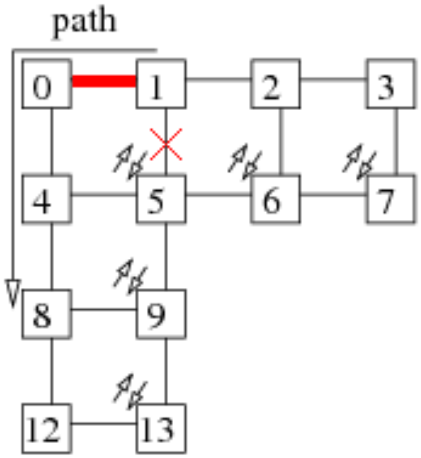
Description (2)



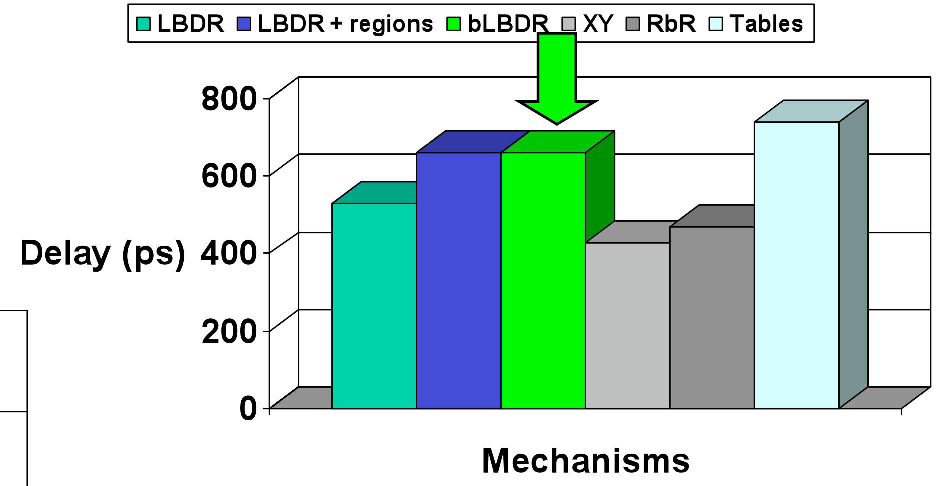


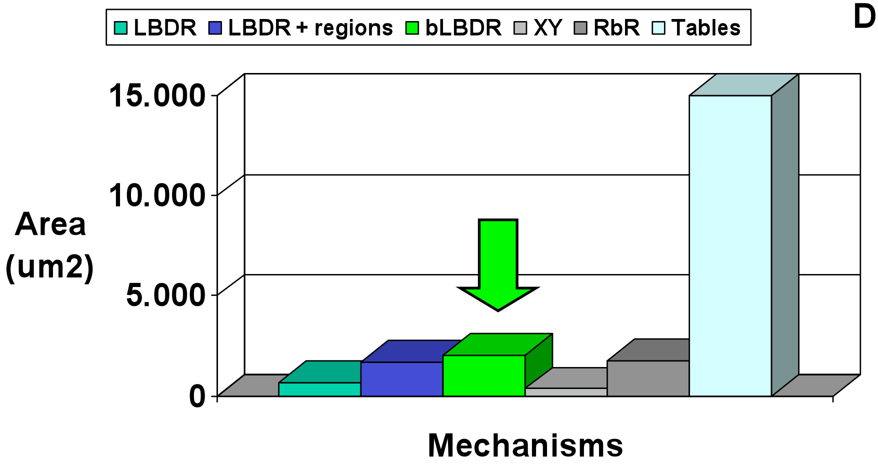
Description (3)

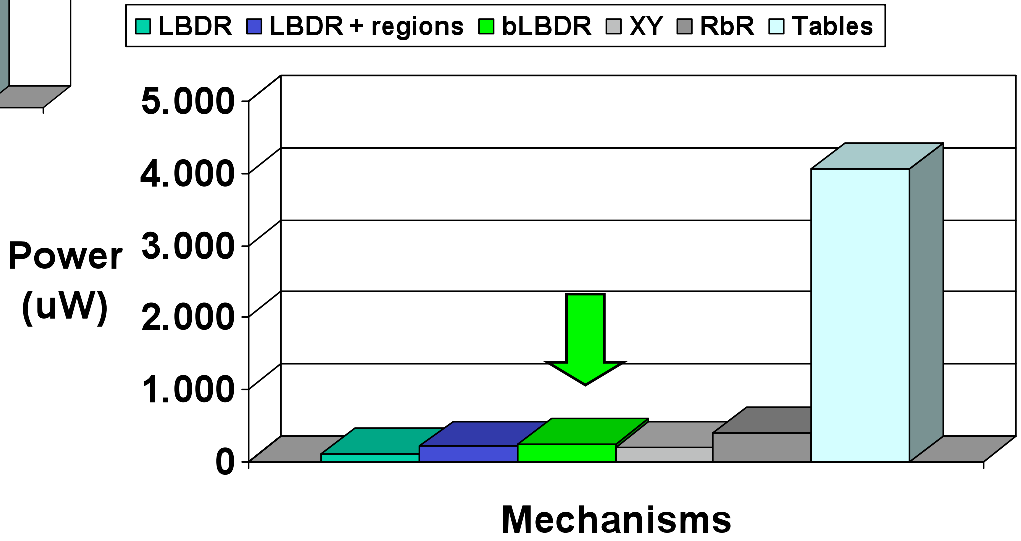
Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



Performance



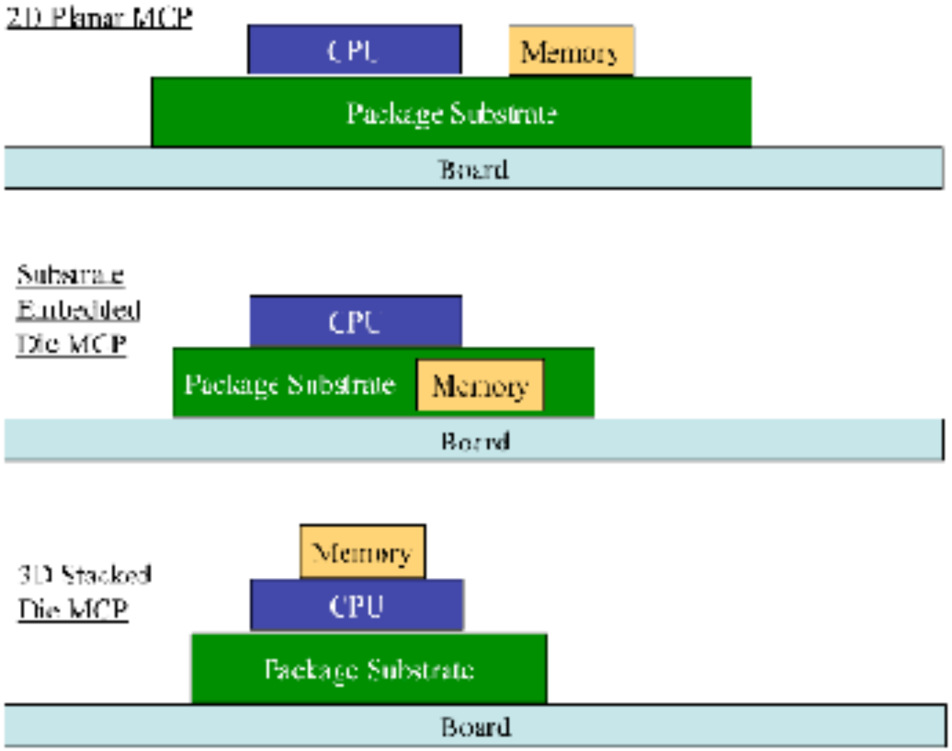




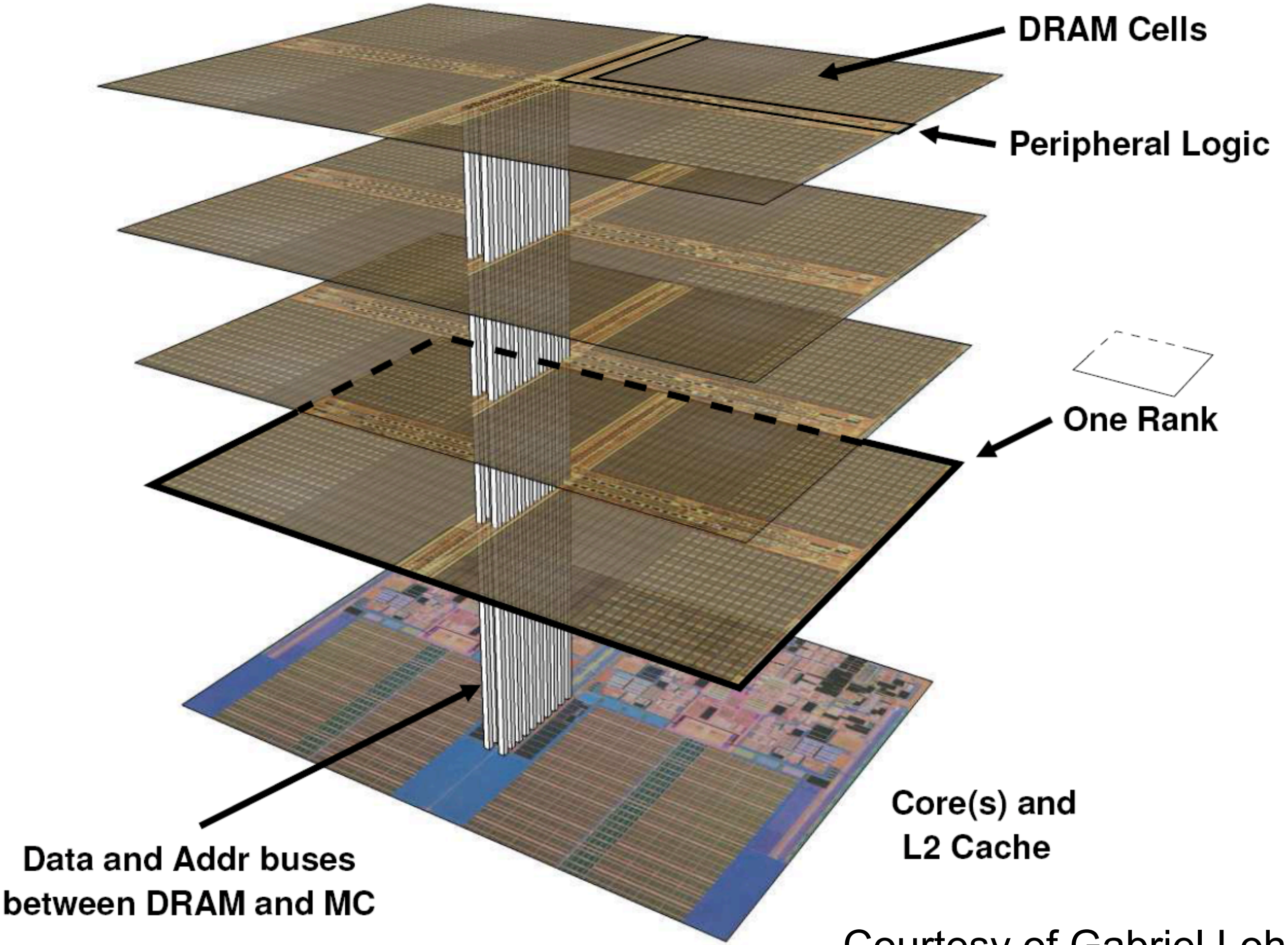
8x8 mesh, TSMC library 90nm technology (we thank Maurizio Palesi for the evaluation results)

Addressing Bandwidth Constraints

- 3D stacking of DRAM seems the most viable and effective approach



DRAM and Cores in a Single Stack

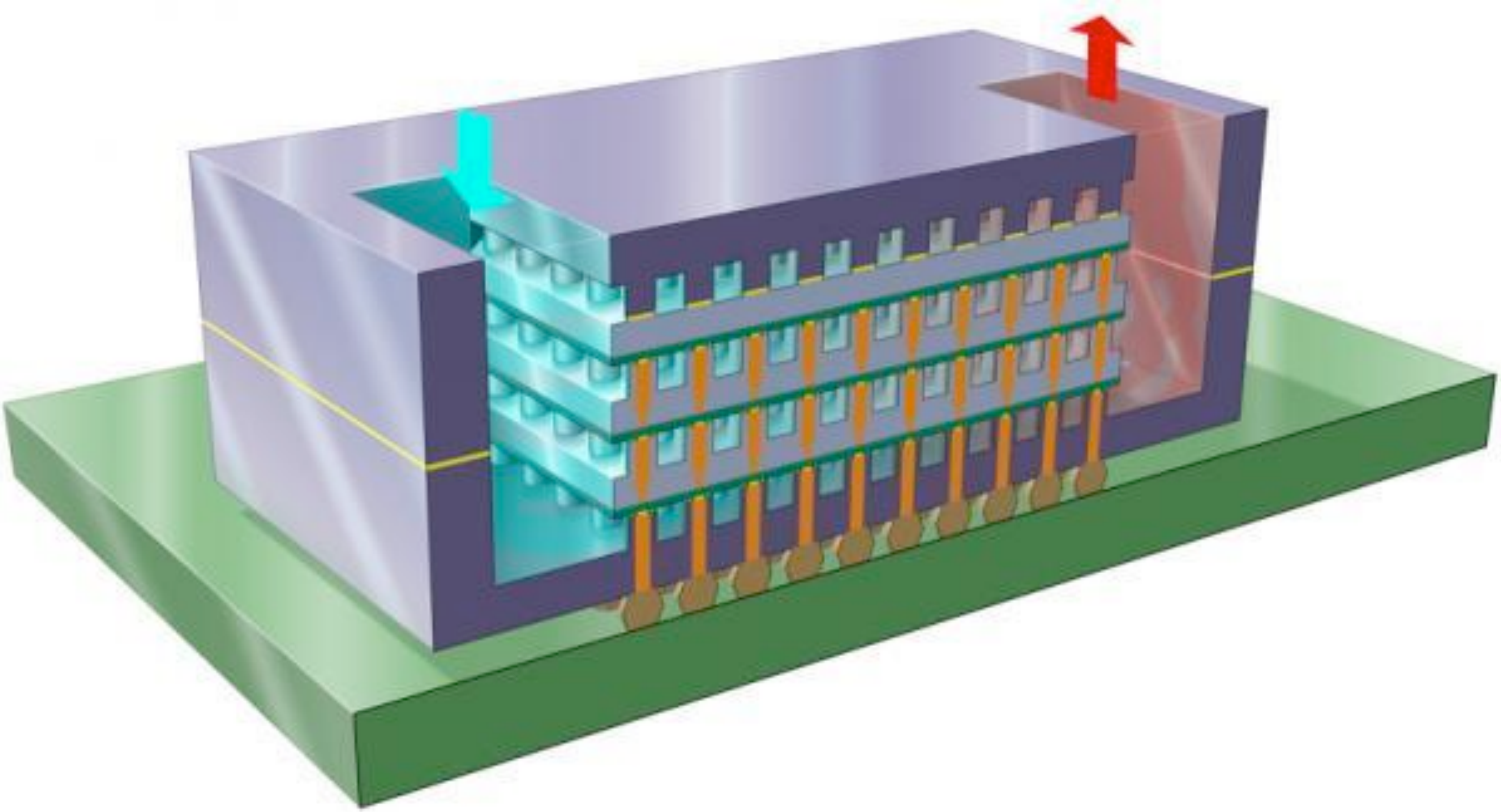


Courtesy of Gabriel Loh, ISCA 2008

Addressing Heat Dissipation

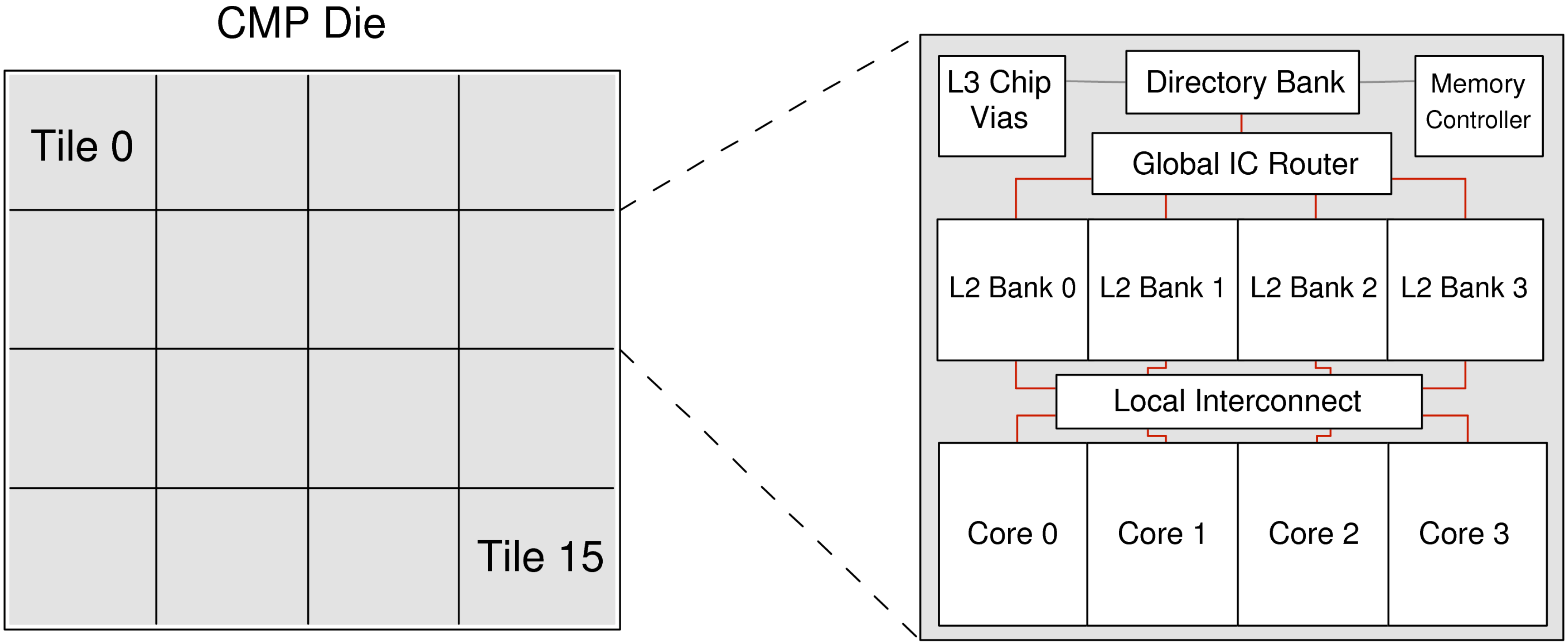
- Most feasible techniques to reduce power consumption have already been implemented in current cores
- Increasing the number of cores will increase power consumption. Options are:
  - Using simpler cores (e.g. in-order cores)
    - Niagara 2 has a chip TDP of 95W, and a core TDP of 5.4W, which results in a 32nm scaled core TDP of 1.1W
    - Atom has a chip TDP of 2.5W, and a core TDP of 1.1W, which results in a 32nm scaled core TDP of 0.5W
  - Using new techniques to increase heat dissipation
    - Liquid cooling inside the chip

Handling Heat Dissipation



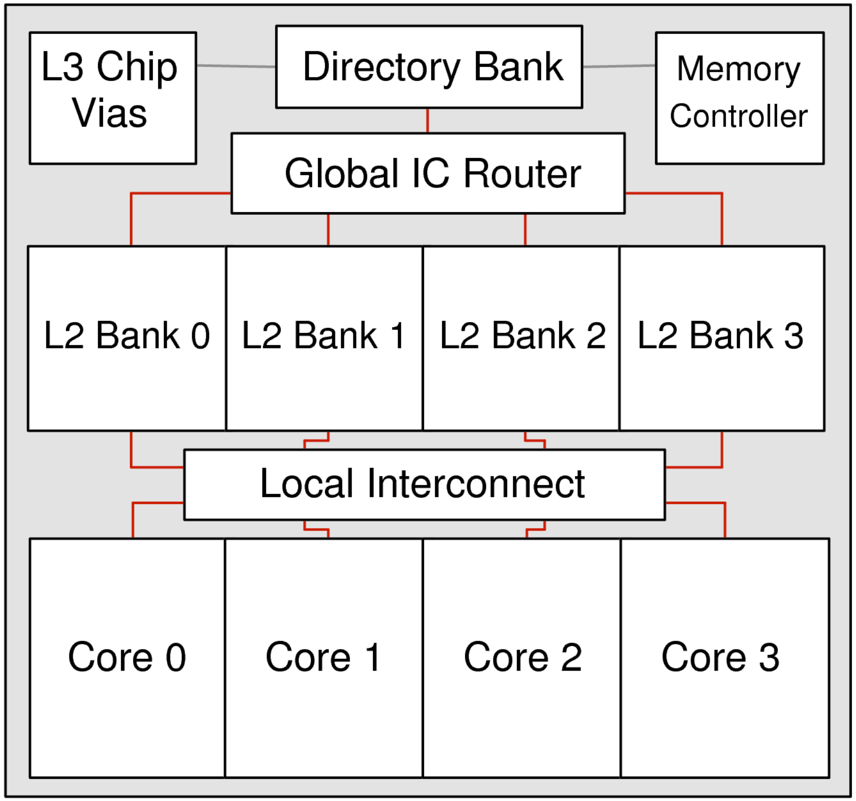
The Role of HyperTransport and QPI

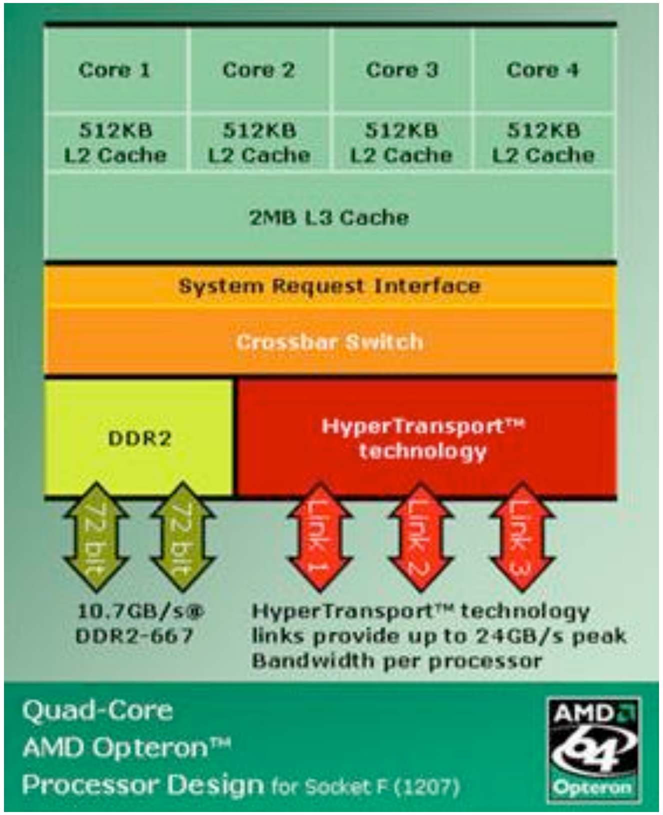
- Tiled architectures reduce design cost and NoC size, and share memory controllers



The Role of HyperTransport and QPI

- Tile architecture versus 4-core Opteron architecture: HT/QPI-based NoCs?

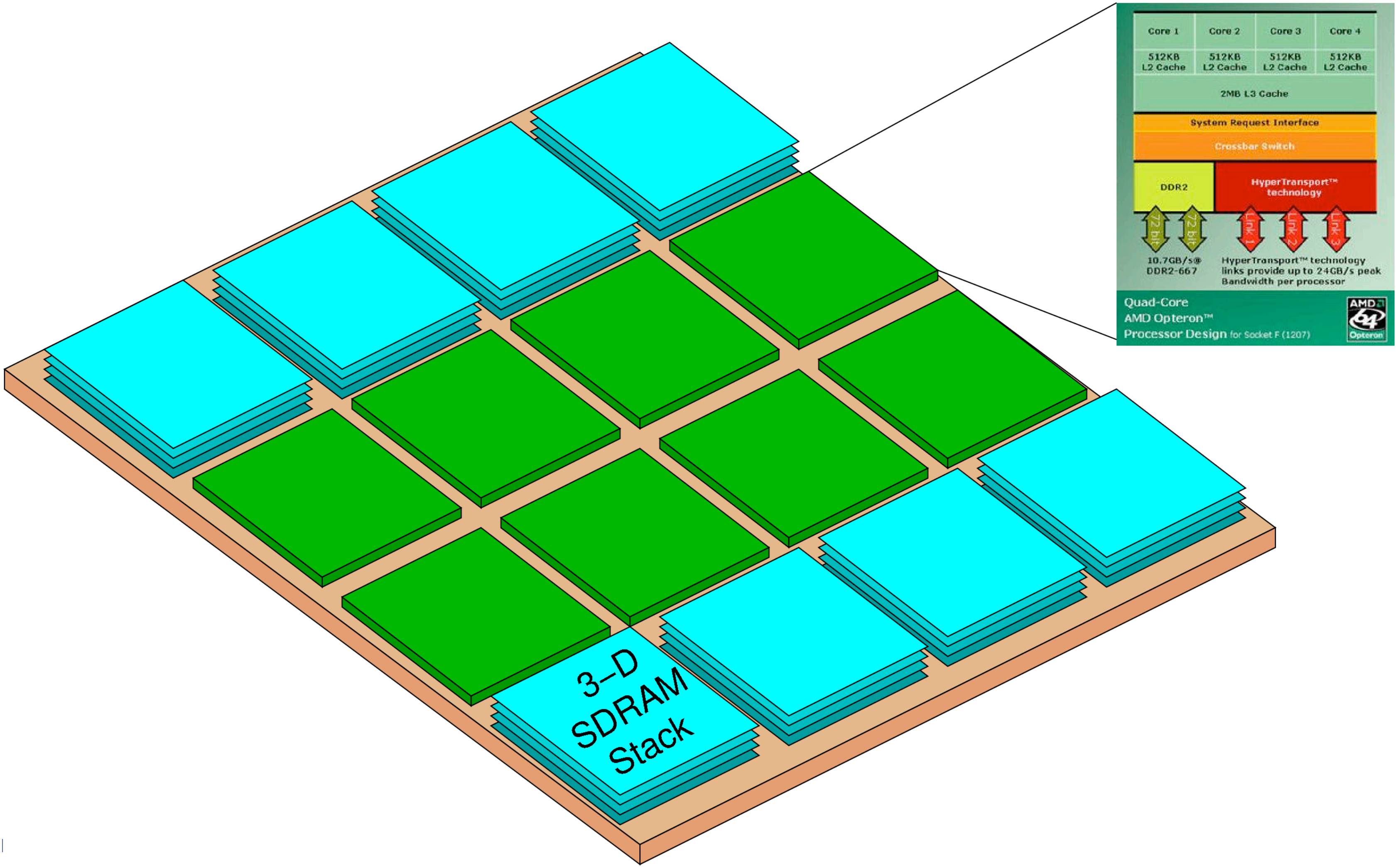




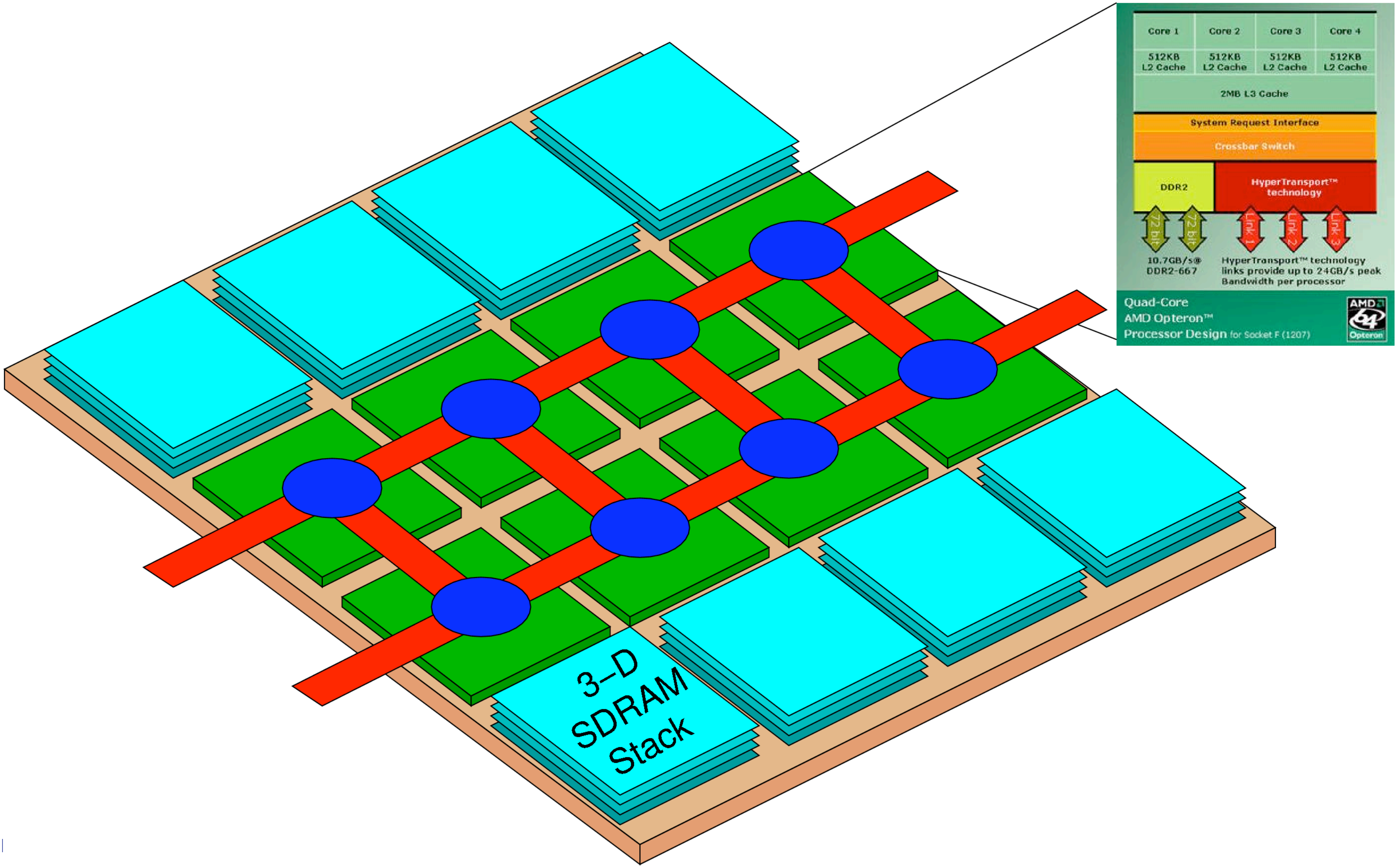
Reducing Design Cost and Time to Market

- Instead of stacking a multi-core die and several DRAM dies…
- Silicon carrier with multiple (smaller) multi-core dies and 3D DRAM stacks
  - Shorter time to market. Just shrink current dies to next VLSI technology
  - Better heat distribution, yield, and fault tolerance
  - Opportunities for design space exploration and optimizations
    - ➢Number of dies of each kind, component location, interconnect patterns, etc.
  - Two-level interconnect: network on-chip and network on-substrate
    - ➢Network on-substrate: Not a new concept; already implemented in SoCs
    - ➢Network on-substrate implemented with metal layers or silicon waveguides
    - ➢Perfect fit for HT/QPI: current chip-to-chip interconnects moved to substrate

Example Based on 4-Core Opteron

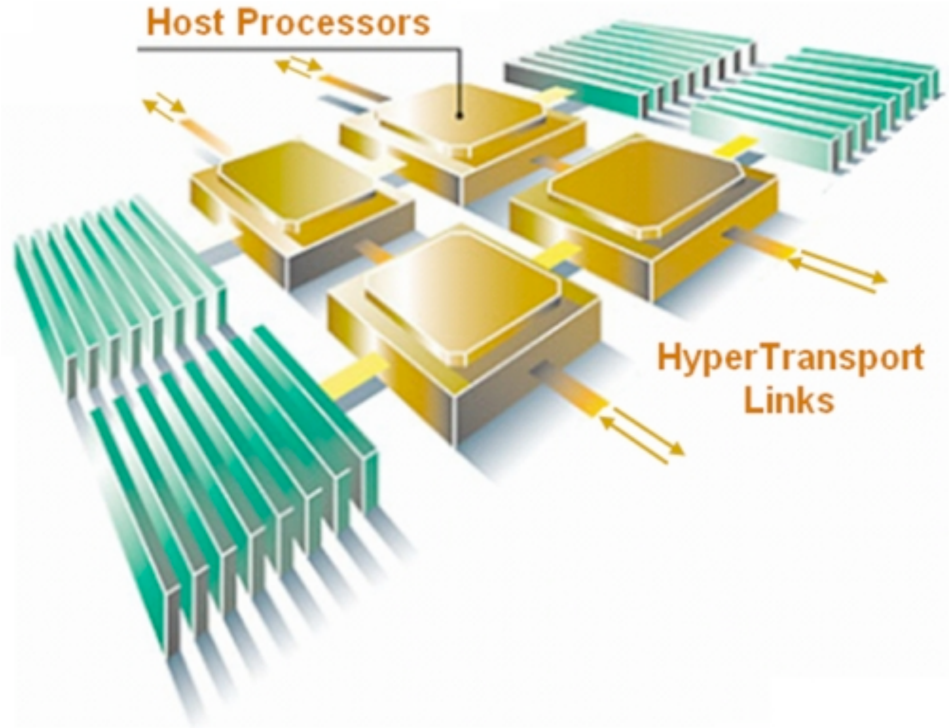


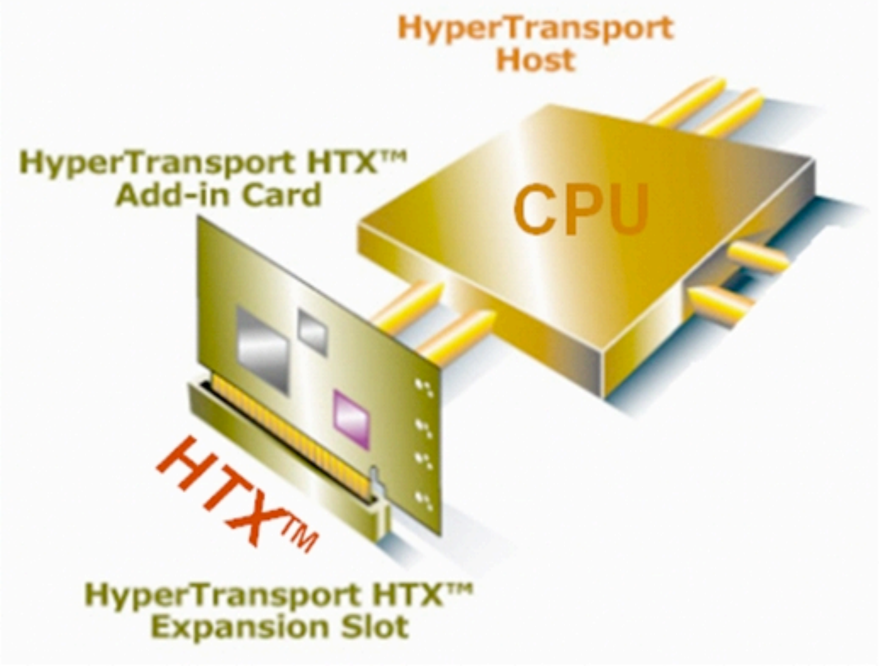
Example Based on 4-Core Opteron



Some Current Research Efforts

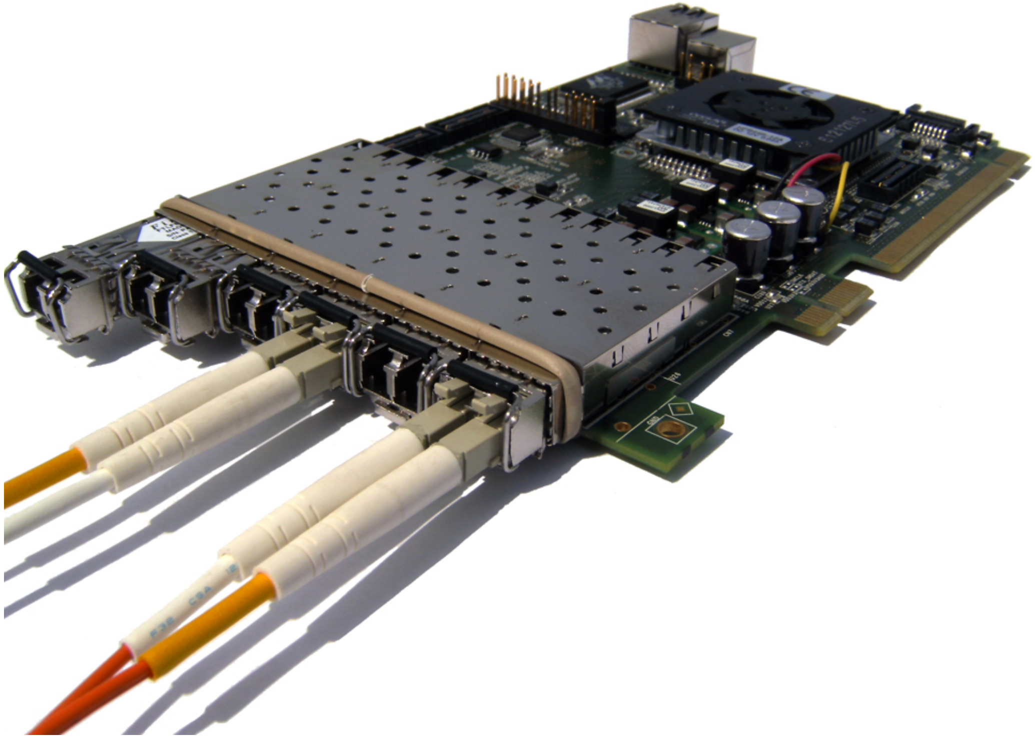
- Implementation and evaluation of High Node Count HT extensions…




HyperTransport Host
HyperTransport HTX™ Add-in Card
CPU
HTX™
HyperTransport HTX™ Expansion Slot

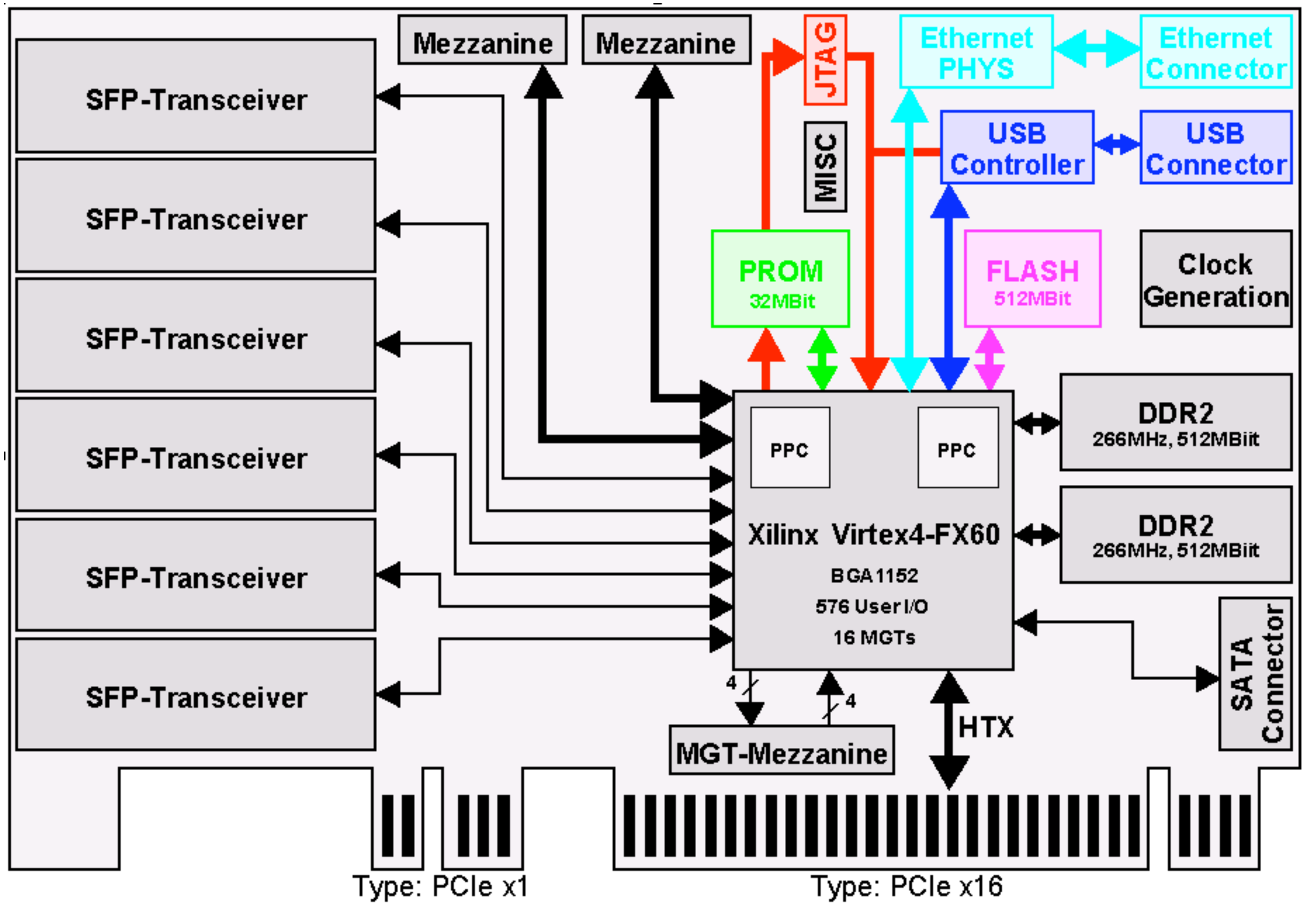
Some Current Research Efforts

- … based on HTX reference card from University of Heidelberg, to model at system level what in the future will be within a single package



Some Current Research Efforts

- The FPGA implements protocol translation, matching store, routing, and NI



Expected Results

- Working prototype with 1024 cores
  - FPGA implementation of protocol translation to HNCHT
  - Optimized libraries for MPI and GASNet
  - Evaluation with sample parallel applications
  - Extension of cache coherence protocols for using remote memory
- Limitations
  - Cache coherence protocols not scalable
  - Long latency when accessing remote memory
  - Low bandwidth when accessing remote memory with load/store (limited by MSHRs and load-store queue size in the Opteron)

Conclusions

- Future multi-core chips face three big challenges: power consumption (and heat dissipation), memory bandwidth, and on-chip interconnects
- Despite the simplicity and beauty of homogeneous designs, designers will be forced to consider heterogeneity
- There exist many sources of heterogeneity, imposed by either architecture, technology, or usage models. No way to escape!
- It is very challenging, but not impossible, to provide efficient, cost-effective architectural support for heterogeneity in a NoC
- Some solutions have been proposed for heat dissipation. The question is whether they will become cost effective
- 3D stacking is the most promising approach to address memory bandwidth. Two flavors (single and multiple stacks) offer very different trade-offs
- HT/QPI fits very well with on-chip and on-substrate interconnect requirements

Thank you!

Addressing heterogeneity, failures and variability in high-performance NoCs

José Duato
Parallel Architectures Group (GAP)
Technical University of Valencia (UPV)
Spain