Adaptive 3D Cache Architecture for Manycores

Eric Guthmuller

CEA-Leti, DACLE, LISAN

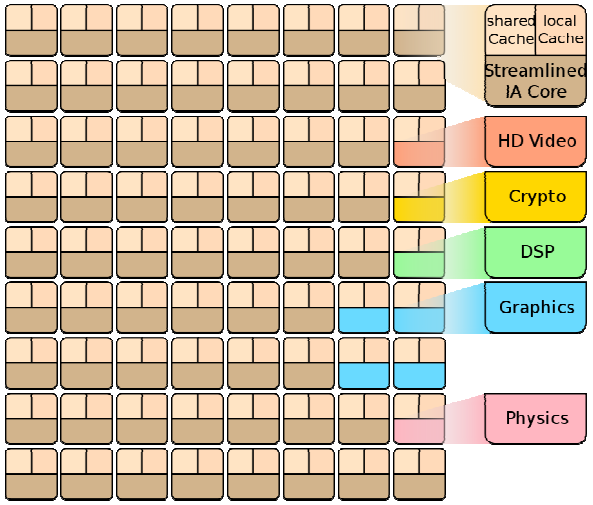
UPMC, LIP6, ALSoC

GDR SoC SiP - 2012-11-15

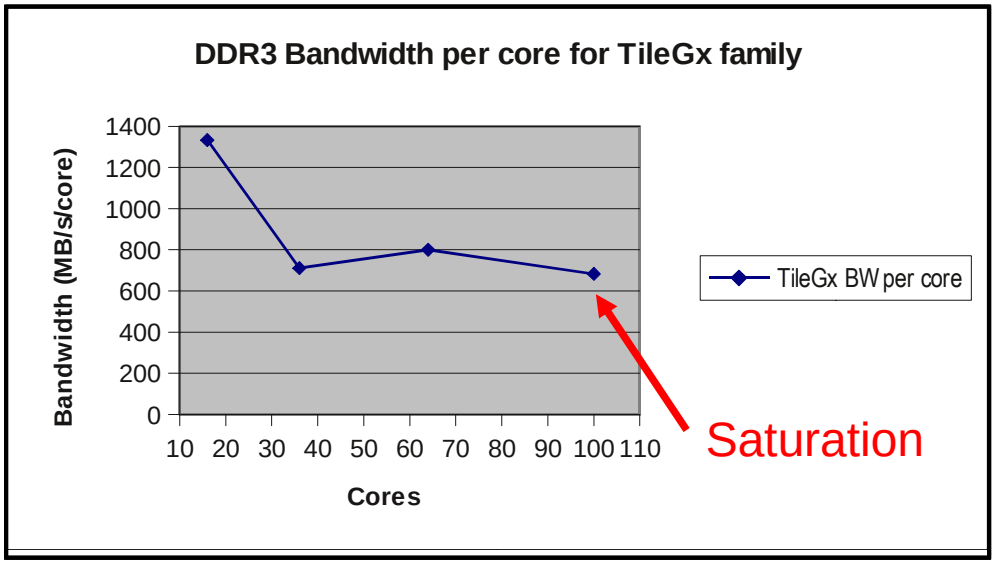
Introduction

- Number of cores in computing ICs is exploding
- Applications use more and more memory amount
- Cores need a fast access to the memory

=> Memory Wall



Intel Terascale Architecture
(source wikipedia)



External memory bandwidth per processor in the Tile Gx architecture of Tilera

Introduction

- Memory bandwidth per core in Nvidia GPUs over time:

Year	GPU	Shader Procs (# Cores)	PCIe Speed	GDDR Bandwidth
2003	GeForce PCX4300	2	40	5.33
2004	GeForce PCX5300	5	40	5.33
2005	GeForce 6500	7	40	5.33
2006	GeForce 7900GT	64	40	42.2
2007	GeForce 8800GT	112	80	57.6
2008	GeForce 9800GT	112	80	57.6
2009	GeForce GTX280	240	80	141.7
2010	GeForce GTX480	480	80	177.4
2011	GeForce GTX580	512	80	192.4

source: http://denalimemoryreport.com

Outline

- Context and motivations
  - The cube law
  - 3D technologies
- TSAR manycore architecture
  - Architecture
  - Cache Coherency
- Adaptive distributed 3D cache
  - TSAR 3D L3
  - Adaptivity
  - Architecture
- Results
  - Tile allocation
  - Fault tolerance
- Hardware implementation

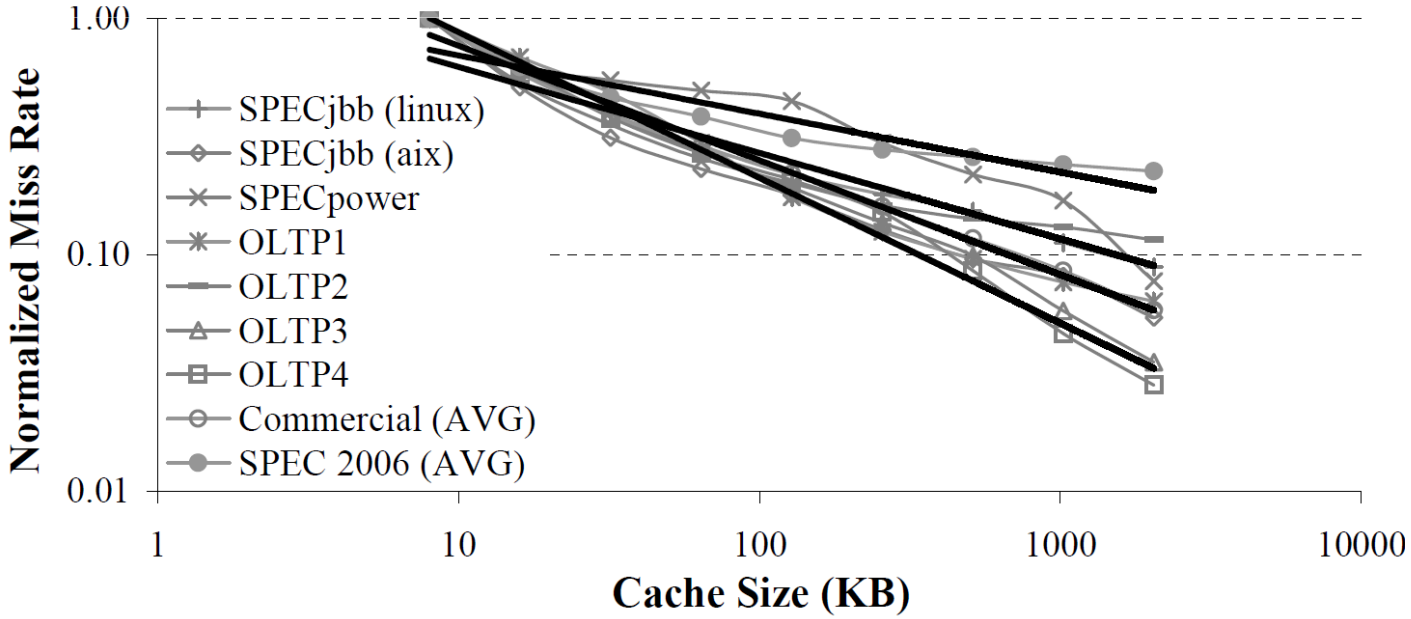
Outline

- Context and motivations
  - The cube law
  - 3D technologies
- TSAR manycore architecture
  - Architecture
  - Cache Coherency
- Adaptive distributed 3D cache
  - TSAR 3D L3
  - Adaptivity
  - Architecture
- Results
  - Tile allocation
  - Fault tolerance
- Hardware implementation

Context and Motivations - Cube Law

- The cache miss rate is a S^x function where S is the size of the cache (see [1])
- x is between 0.36 and 0.62 on commercial workloads

  => Mean of 0.5



Context and Motivations - Cube Law

- With no sharing between processors: memory accesses are multiplied by the number of cores
- To keep constant the external memory bandwidth when the number of cores is multiplied by N, the total cache size must be multiplied by N^3
- With 16% of data sharing (mean for PARSEC benchmarks [1]):

  => $(1-0.16)^3*N^3 = 0.6*N^3$ (with N big)

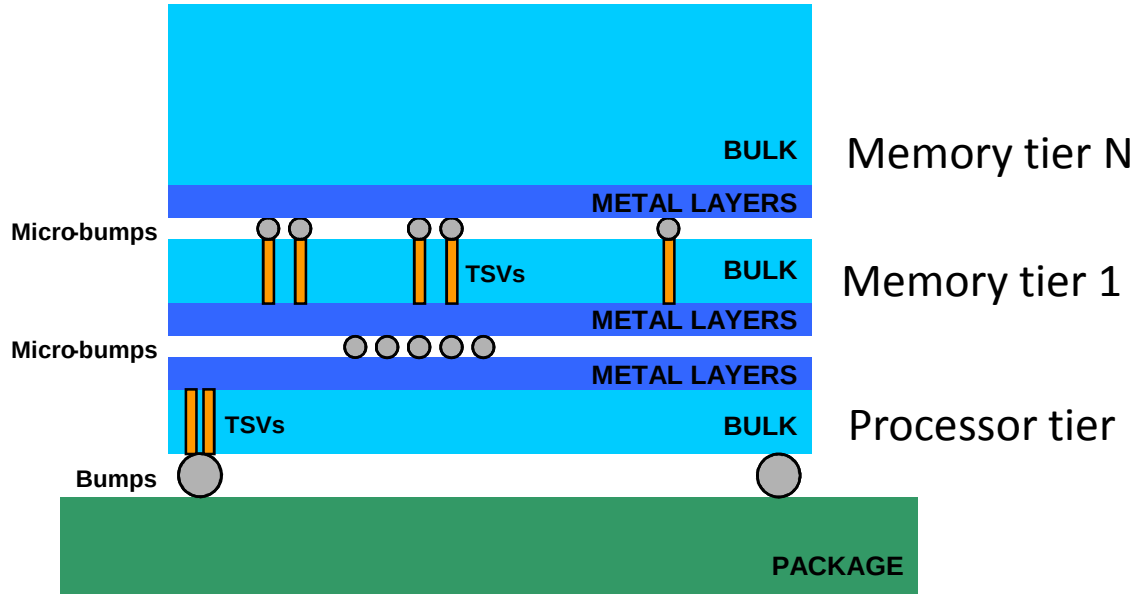
- Manycores need very big caches and/or fast memory

  => Go to 3D caches

Context and Motivations - 3D Stack

- Memory on top of processors
- Multiple layers of logic => Through Silicon Vias (TSV)

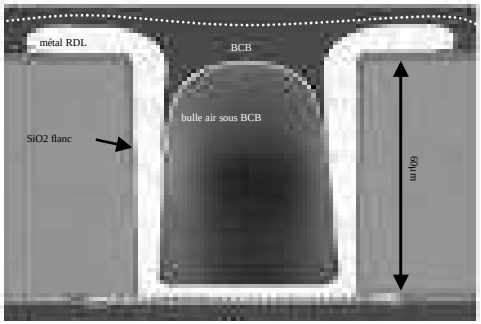
- Example of stacking:



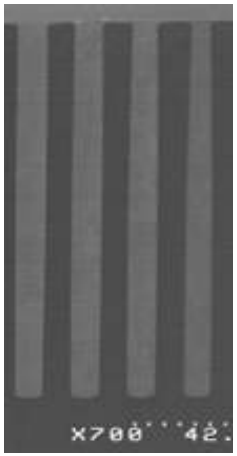
Source: Denis Dutoit; LETI

Context and Motivations - 3D TSV

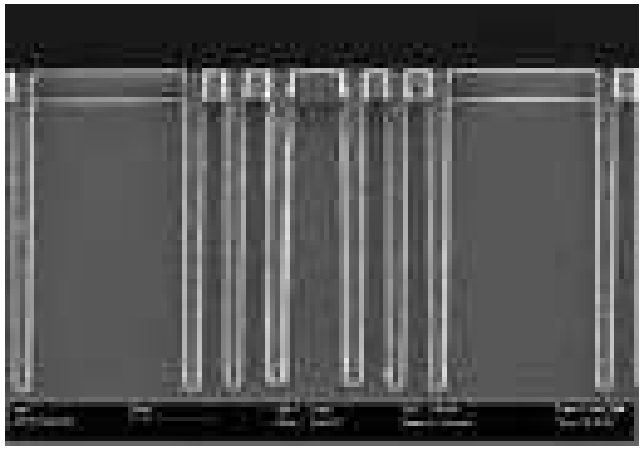
- Different TSV technologies (lined / filled)



AR 1
Ø80x80µm



AR 10,
Ø10x100µm

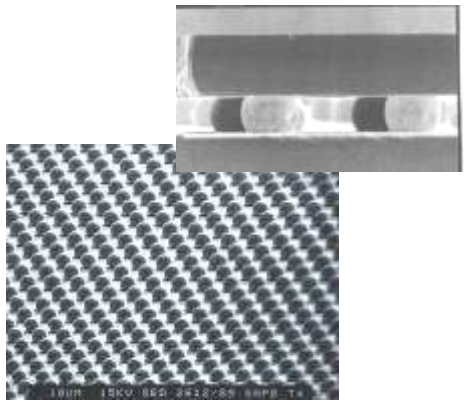


Trench AR 20,
5x100µm

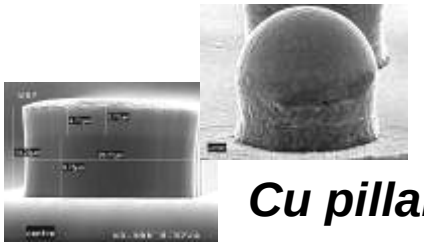
AR= Aspect Ratio

Source: Patrick Leduc; LETI

Context and Motivations - 3D Bumps

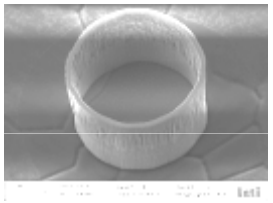


Classic Flip chip
(Ball or stud bump)



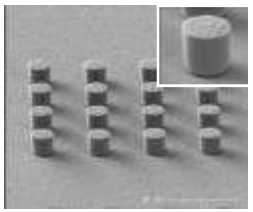
Cu pillars

SnAg Solder



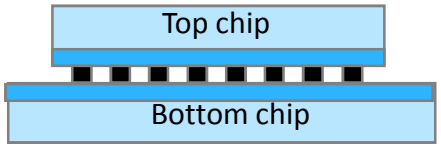
µtubes

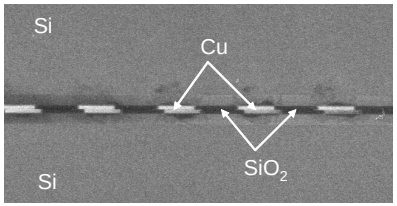
Microinsertion



Solder-free µinserts

Nickel Microinsertion





Cu-Cu Direct bonding

Need very flat surface
Cleaning

> 100 µm ***100-30 µm range*** ***30-10 µm range*** ***Down to 5 µm***

Pitch

Source: Patrick Leduc; LETI

Context and Motivations - Conclusion

⇒ Use dense 3D on-chip memory to build a n+1 level cache

- 3D memory distributed amongst the circuit with many 3D links

  => High bandwidth and fault tolerance

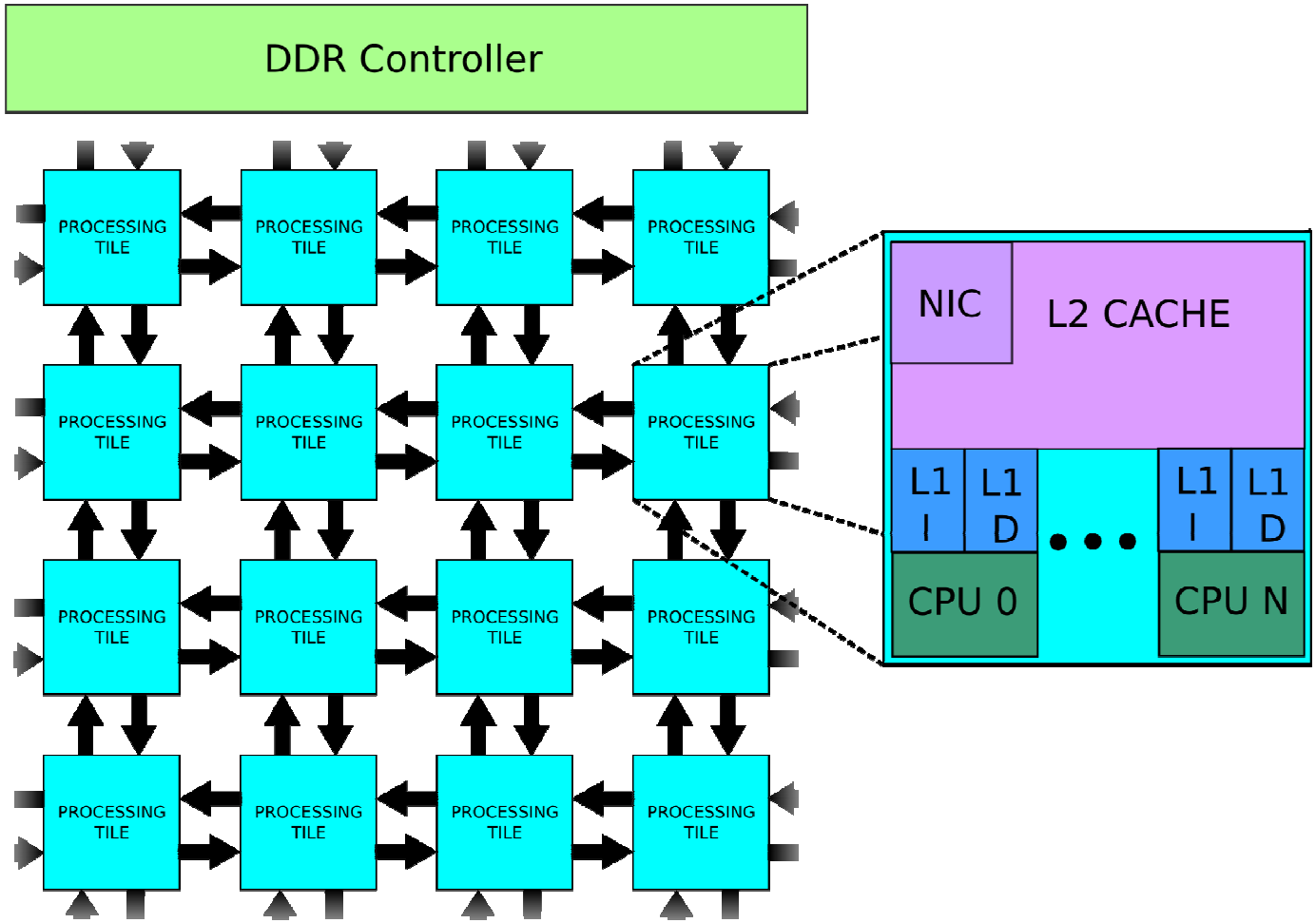
- Goal: stackable and reusable architecture to adapt memory quantity to the application needs, including fault tolerance

Outline

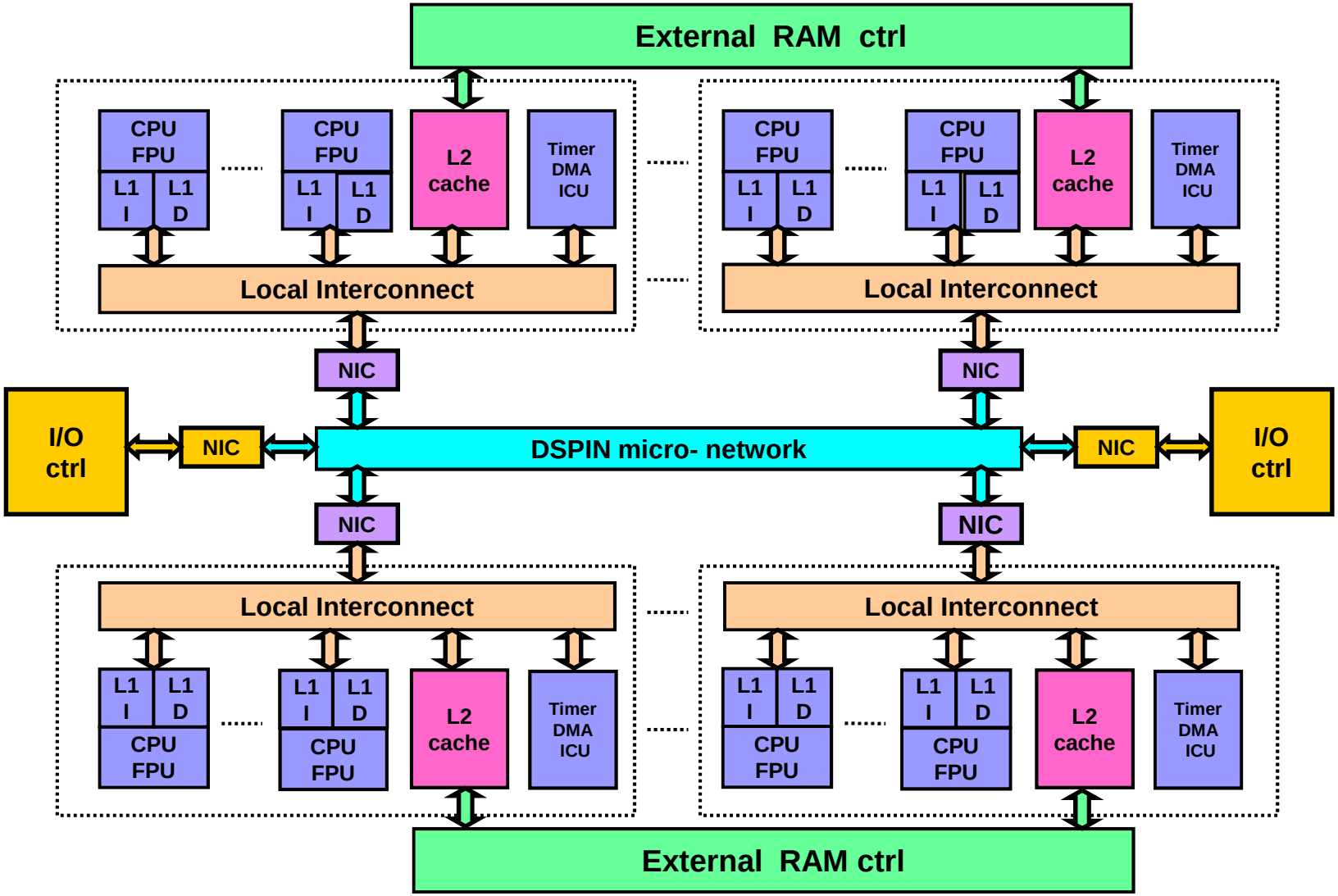
- Context and motivations
  - The cube law
  - 3D technologies
- TSAR manycore architecture
  - Architecture
  - Cache Coherency
- Adaptive distributed 3D cache
  - TSAR 3D L3
  - Adaptivity
  - Architecture
- Results
  - Tile allocation
  - Fault tolerance
- Hardware implementation

TSAR Manycore [2] - Architecture

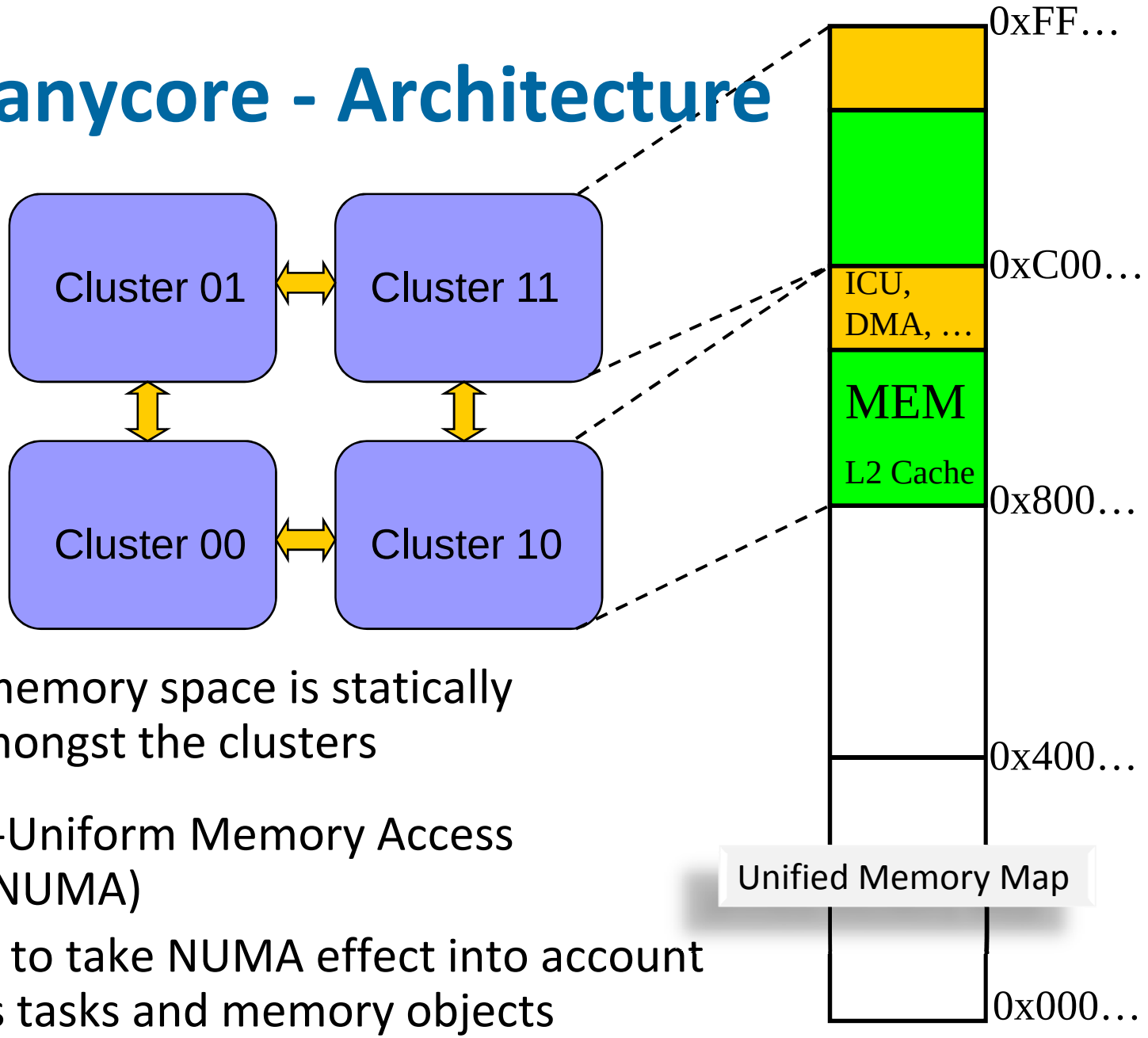
MP²SOC with a 2D-mesh NoC (DSPIN) => GALS



TSAR Manycore - Architecture



TSAR Manycore - Architecture



- The physical memory space is statically distributed amongst the clusters
- TSAR is a Non-Uniform Memory Access architecture (NUMA)

  => The OS has to take NUMA effect into account when it places tasks and memory objects

TSAR Manycore - Cache Coherency

- Thanks to the scalable bandwidth provided by the NoC technology, the TSAR architecture use a write-through policy between L1 and L2 caches
  - ⇒ the L1 cache architecture and the coherency protocol are simplified
  - ⇒ the cache coherency is ensured by the L2 cache (Home Agent)

- As snooping does not scale with a large number of cores, TSAR relies on a L2 cache directory-based approach
  - ⇒ Each L2 cache tracks all the copies stored in the L1 caches in order to update them

TSAR Manycore - Cache Coherency

- TSAR implements the DHCCP protocol (Distributed Hybrid Cache Coherence Protocol):
  - multicast / update when the number of copies is smaller than a predefined threshold
  - broadcast / invalidate when the number of copies is larger than a predefined threshold

- The DHCCP has been analyzed, from the point of view of dead-lock prevention. Three types of traffic have been identified:
  - Direct read/write transactions (L1 caches => L2 caches)
  - Coherency transactions (L2 caches <=> L1 caches)
  - External memory accesses (L2 caches => Ext. RAM)

- 6 communication channels

TSAR Manycore - Cache Coherency

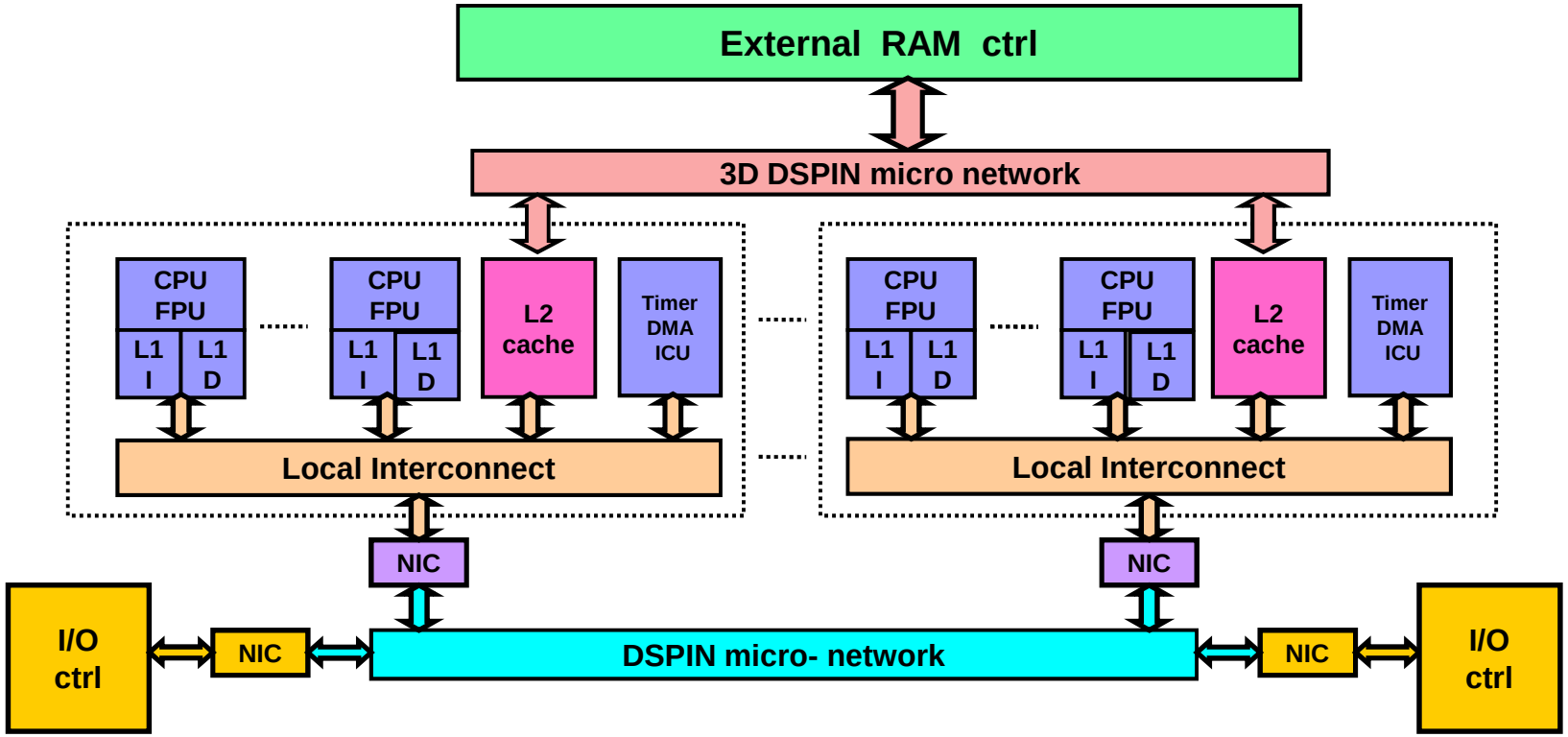
- Write-through:
  - Simple coherency protocol
  - Simple hardware
  - BUT more network traffic
  - Caches need to be inclusive

- Write-back:
  - Network traffic reduced
  - Automatic replication and distribution
  - Cache storage better exploited
  - BUT complex protocol
  - Problem of false sharing

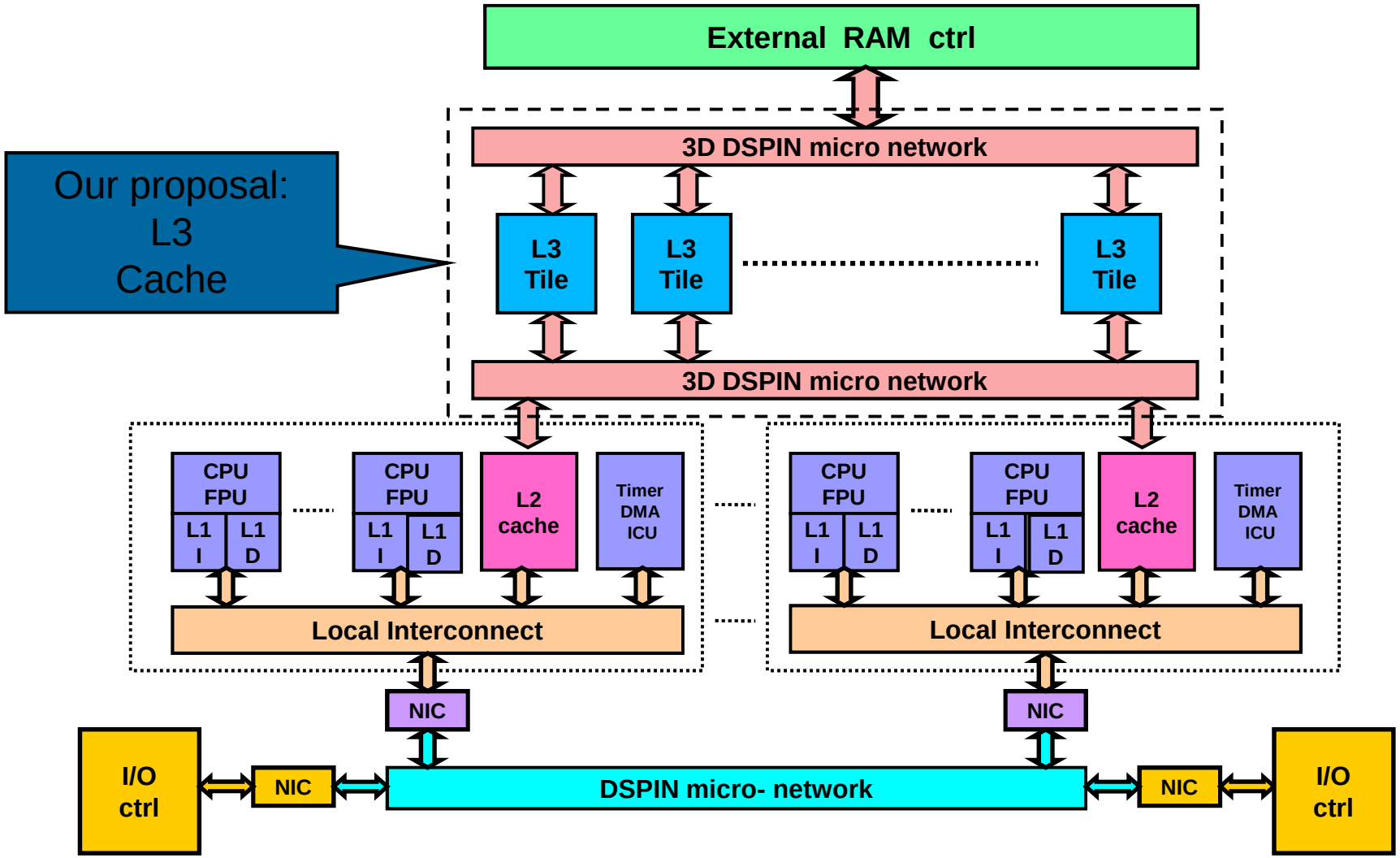
Outline

- Context and motivations
  - The cube law
  - 3D technologies
- TSAR manycore architecture
  - Architecture
  - Cache Coherency
- Adaptive distributed 3D cache
  - TSAR 3D L3
  - Adaptivity
  - Architecture
- Results
  - Tile allocation
  - Fault tolerance
- Hardware implementation

3D Cache - TSAR 3D L3

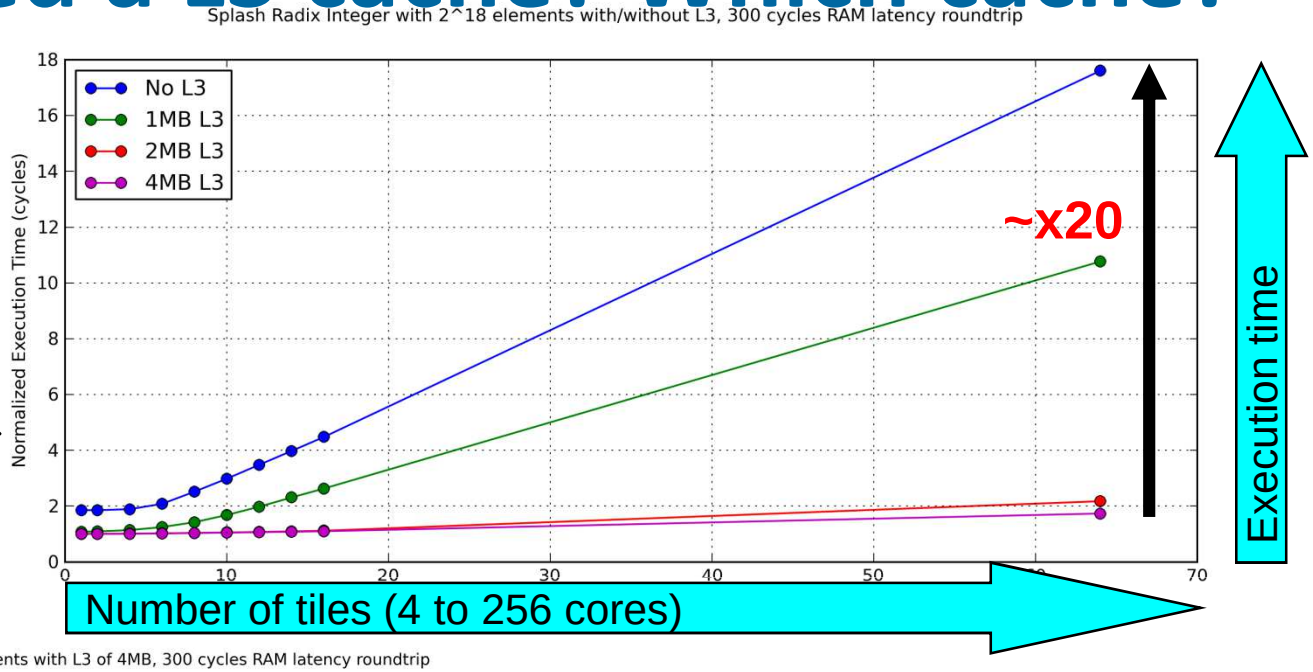


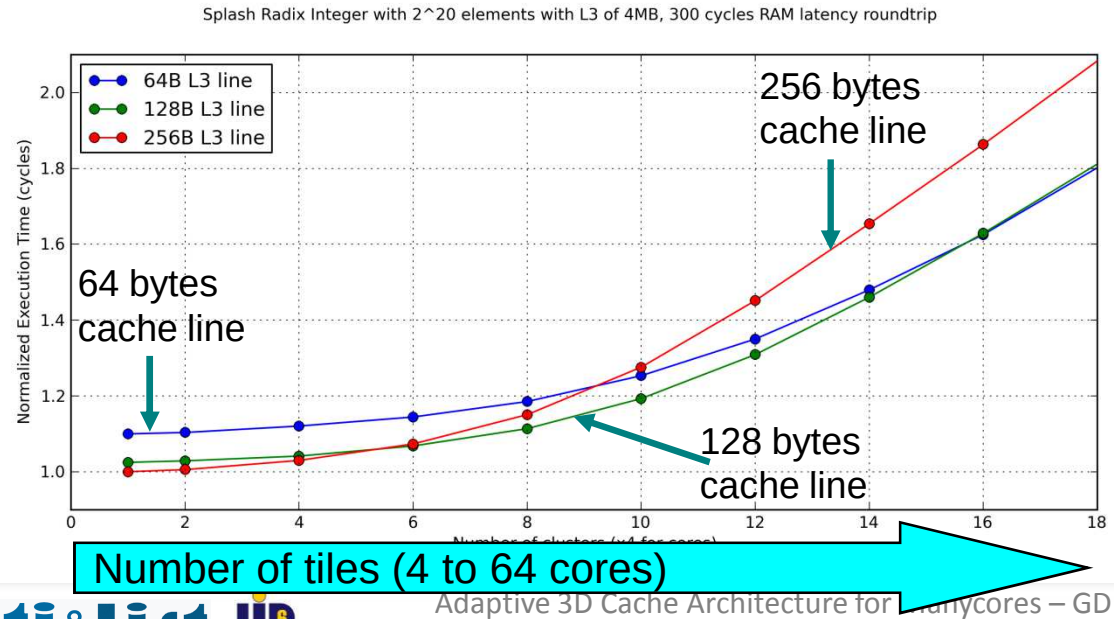
3D Cache - TSAR 3D L3



Do We Need a L3 cache? Which cache?

Impact of the external memory bandwidth on performances

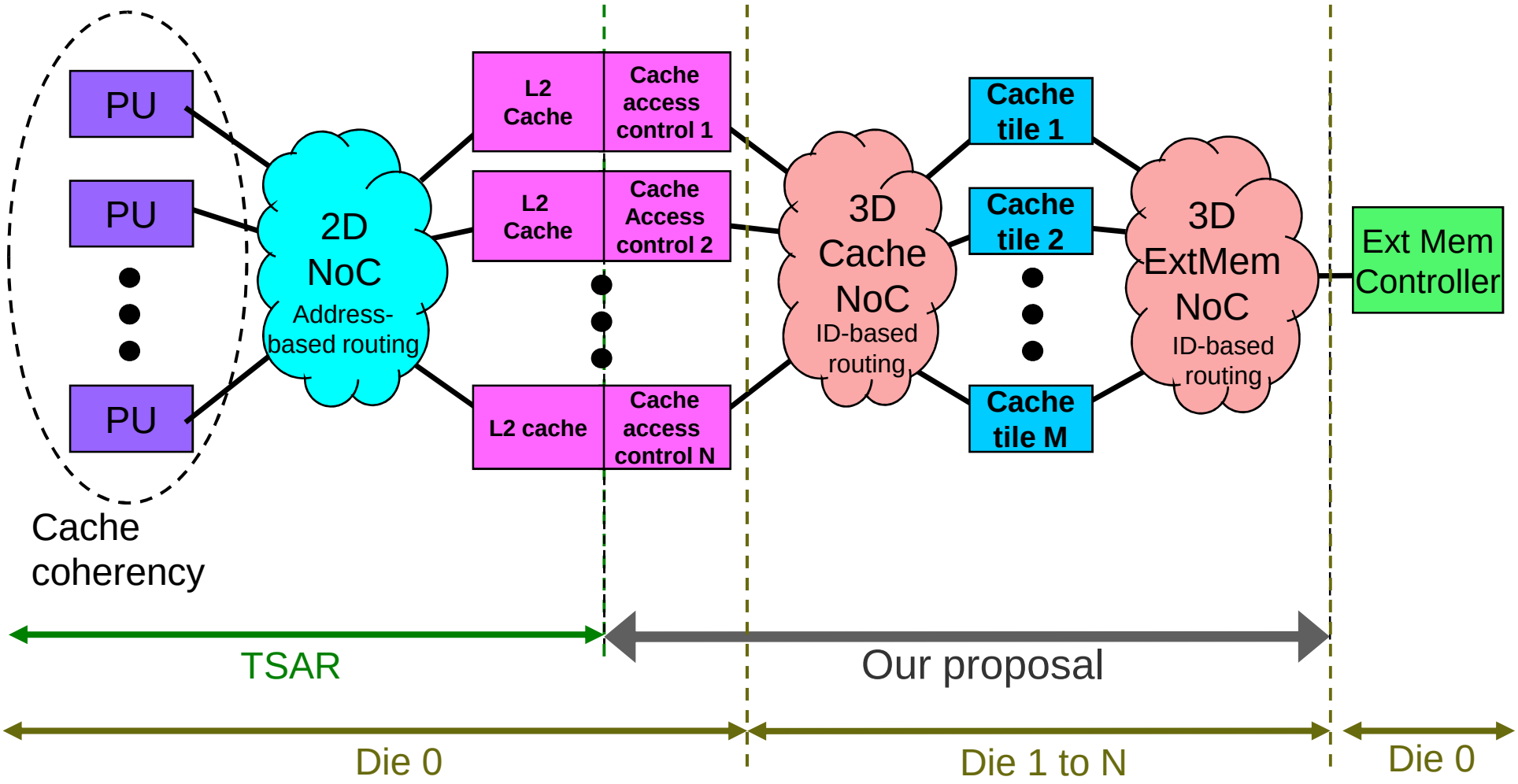




Optimal cache line size depends on the available external memory bandwidth

3D Cache - TSAR 3D L3

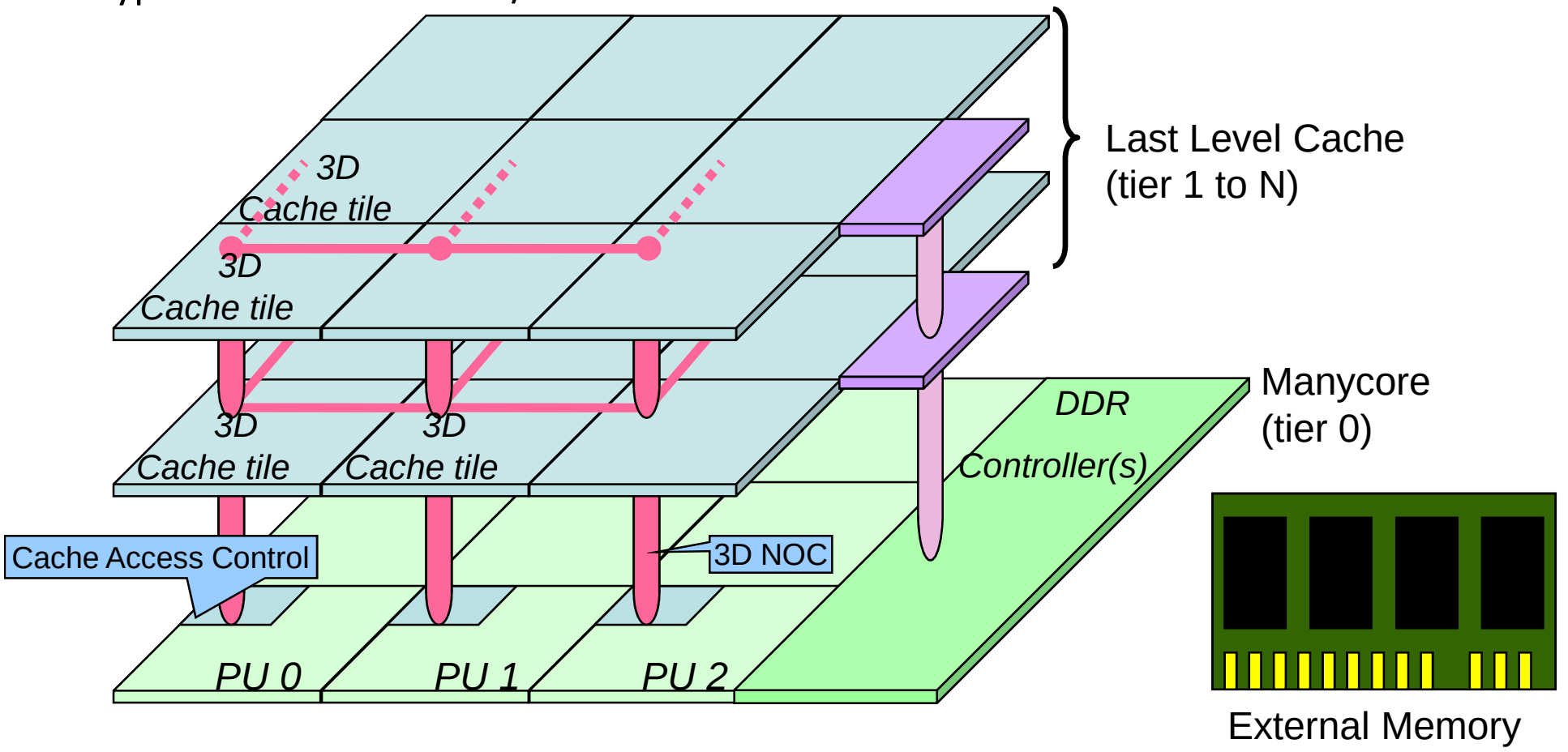
- An architecture built around 3 NoCs



3D Cache - TSAR 3D L3

- Proposed 3D architecture: regular 3D mesh

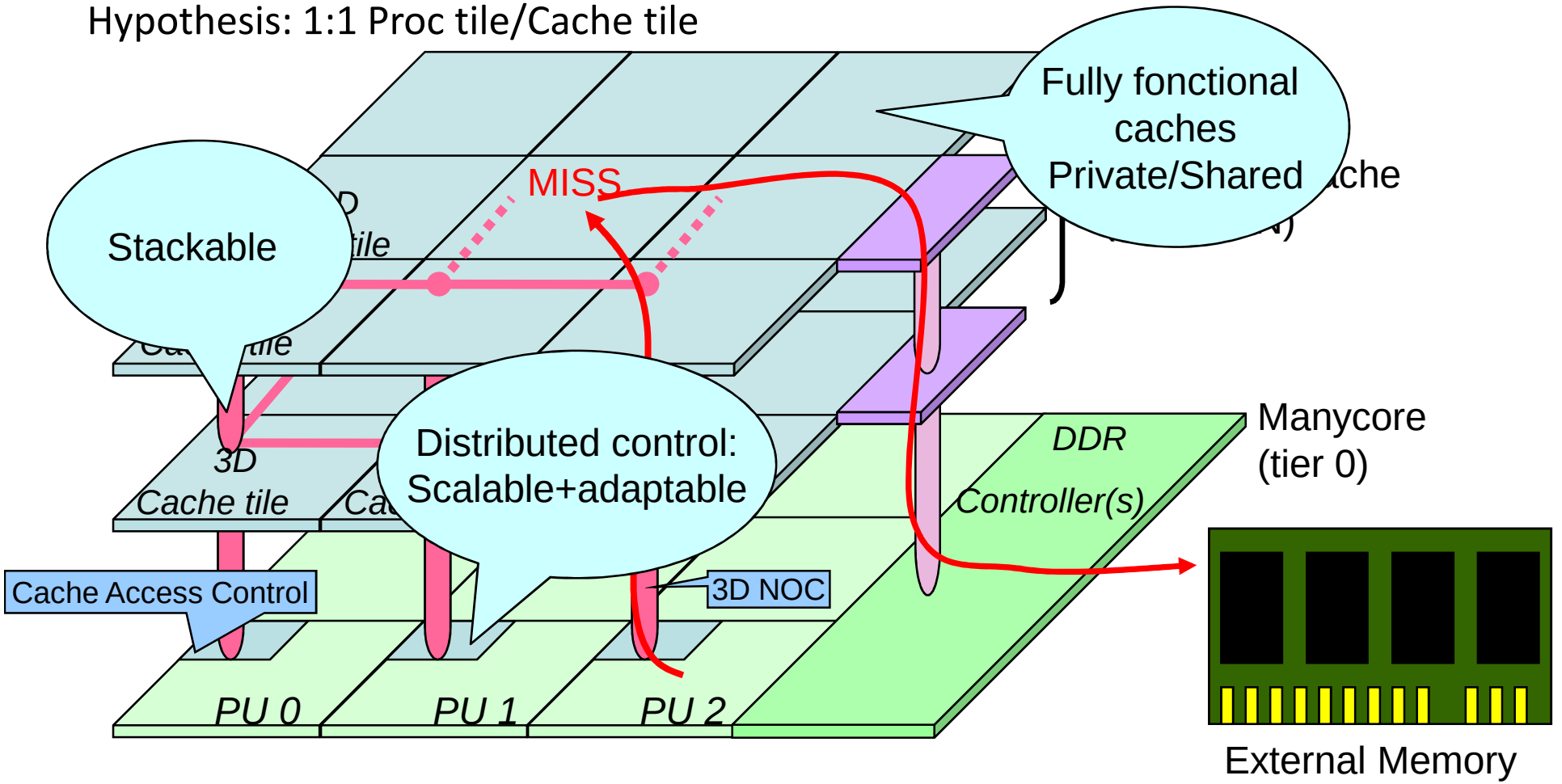
Hypothesis: 1:1 Proc tile/Cache tile



3D Cache - TSAR 3D L3

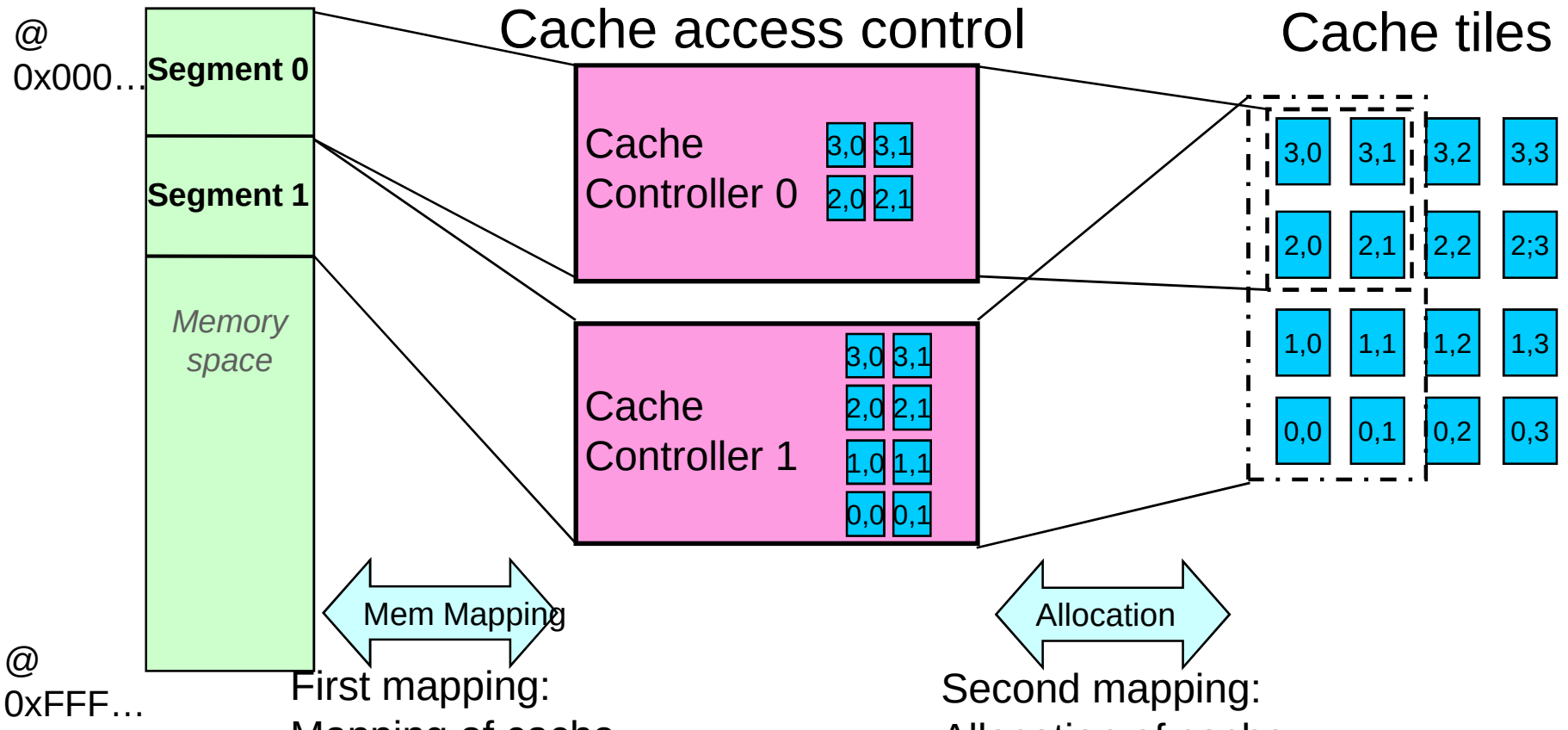
- Proposed 3D architecture: regular 3D mesh

Hypothesis: 1:1 Proc tile/Cache tile



3D Cache - Adaptivity

- Allocation of cache tiles to memory segments

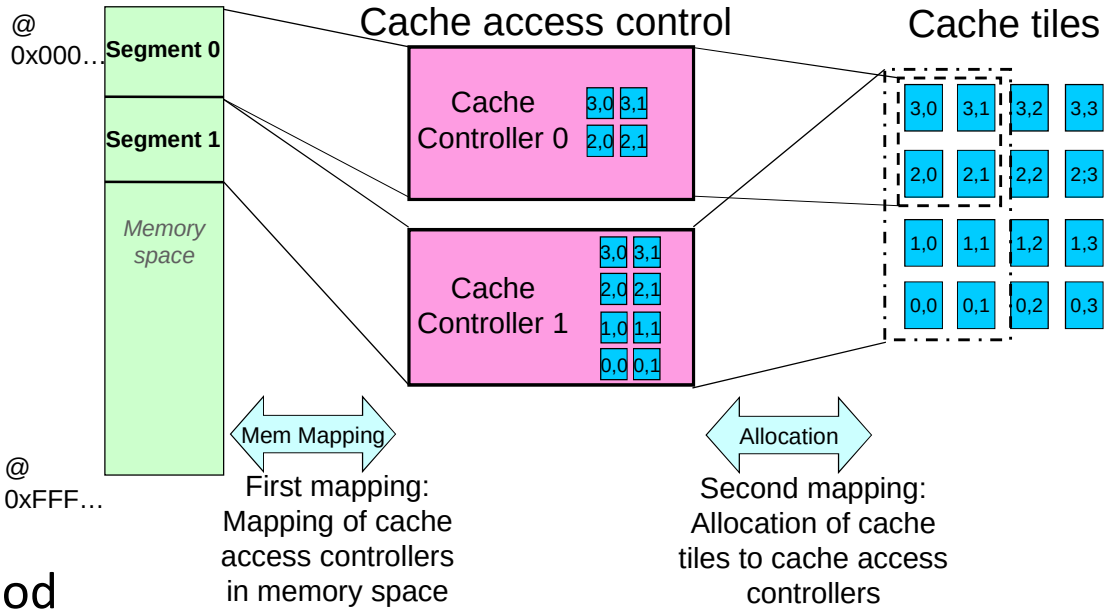


First mapping:
Mapping of cache access controllers in memory space

Second mapping:
Allocation of cache tiles to cache access controllers

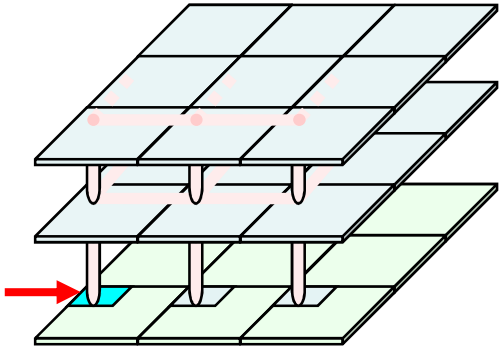
3D Cache - Adaptivity

- Allocation of cache tiles to memory segments

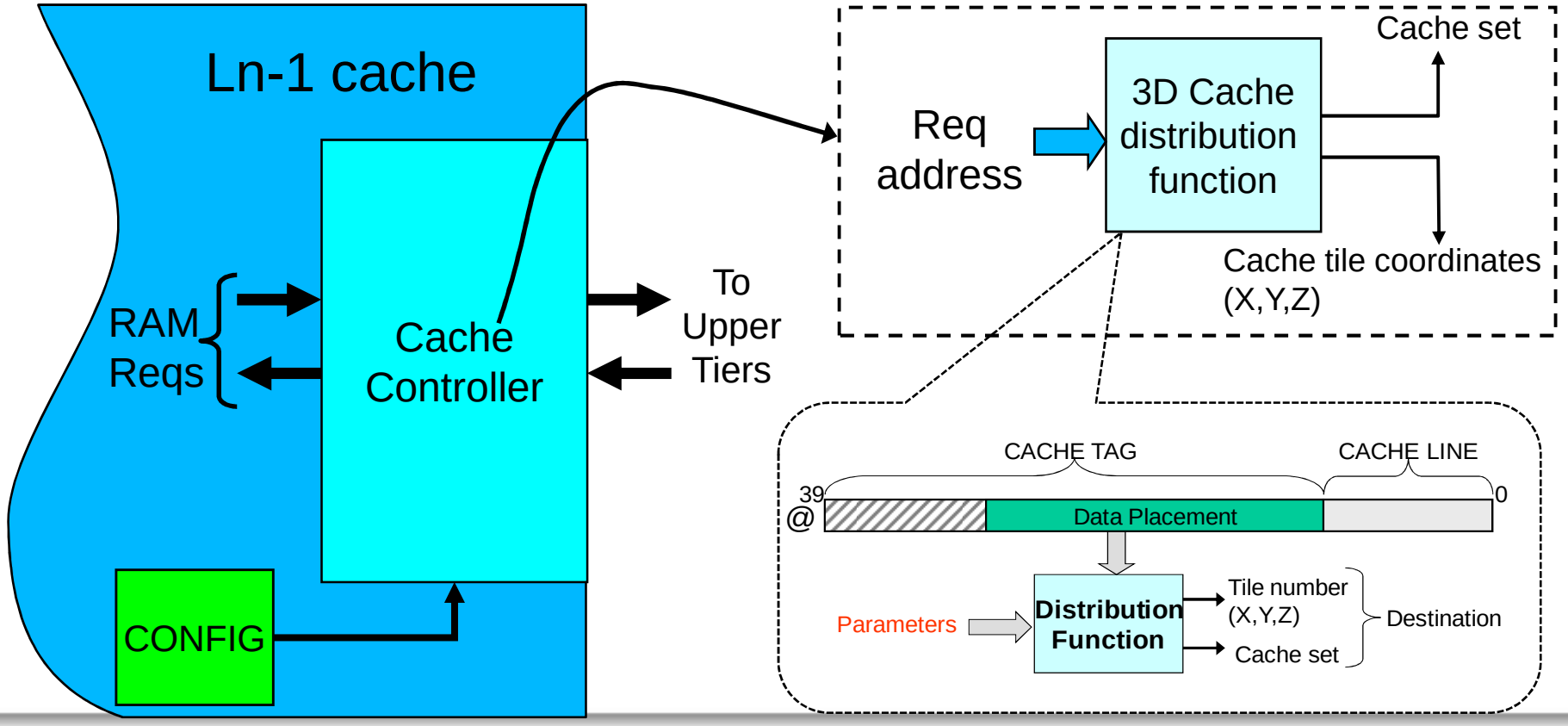


- 3D: low latency to neighborhood
- Different cache quantities for different memory segments
- Shared cache tiles: the most accessed memory segment takes the most part of the cache tile
- Conclusion: 2 levels of adaptability:
  - Programmable by the soft (OS) at runtime
  - Very dynamic when cache tiles are shared

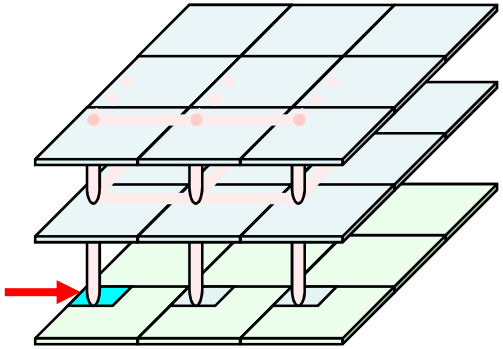
3D Cache - Architecture



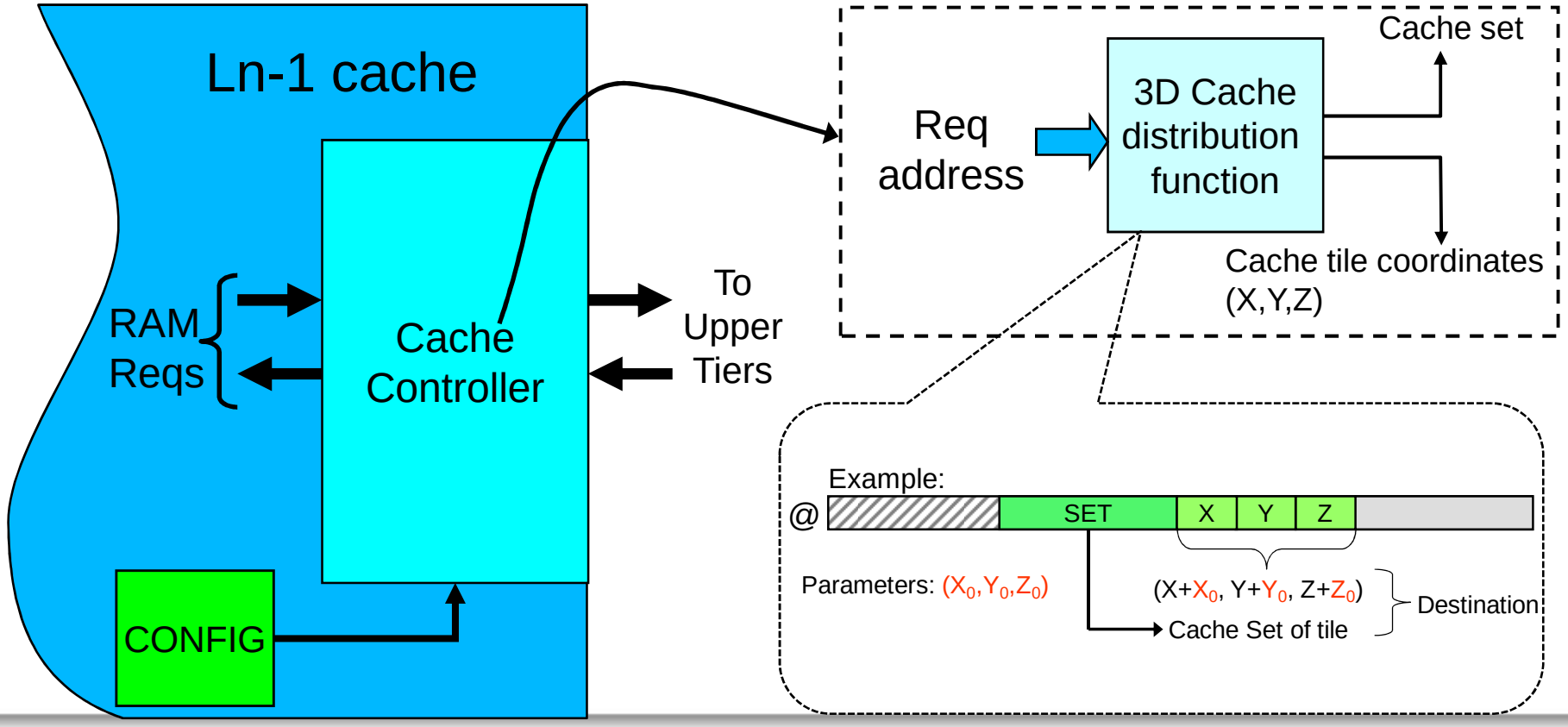
- Cache controllers integrated in the computing tier
- Selects the tile coordinates and its cache set to use



3D Cache - Architecture



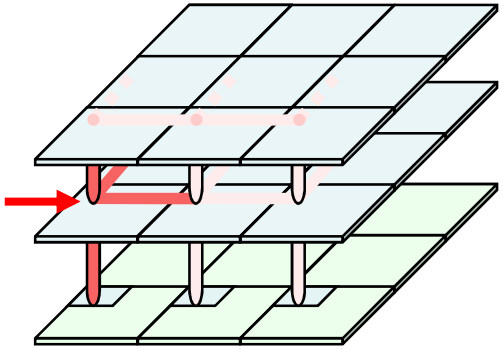
- Cache controllers integrated in the computing tier
- Selects the tile coordinates and its cache set to use



3D Cache - Architecture

- Four 3D 64 bits DSPIN NoC channels:

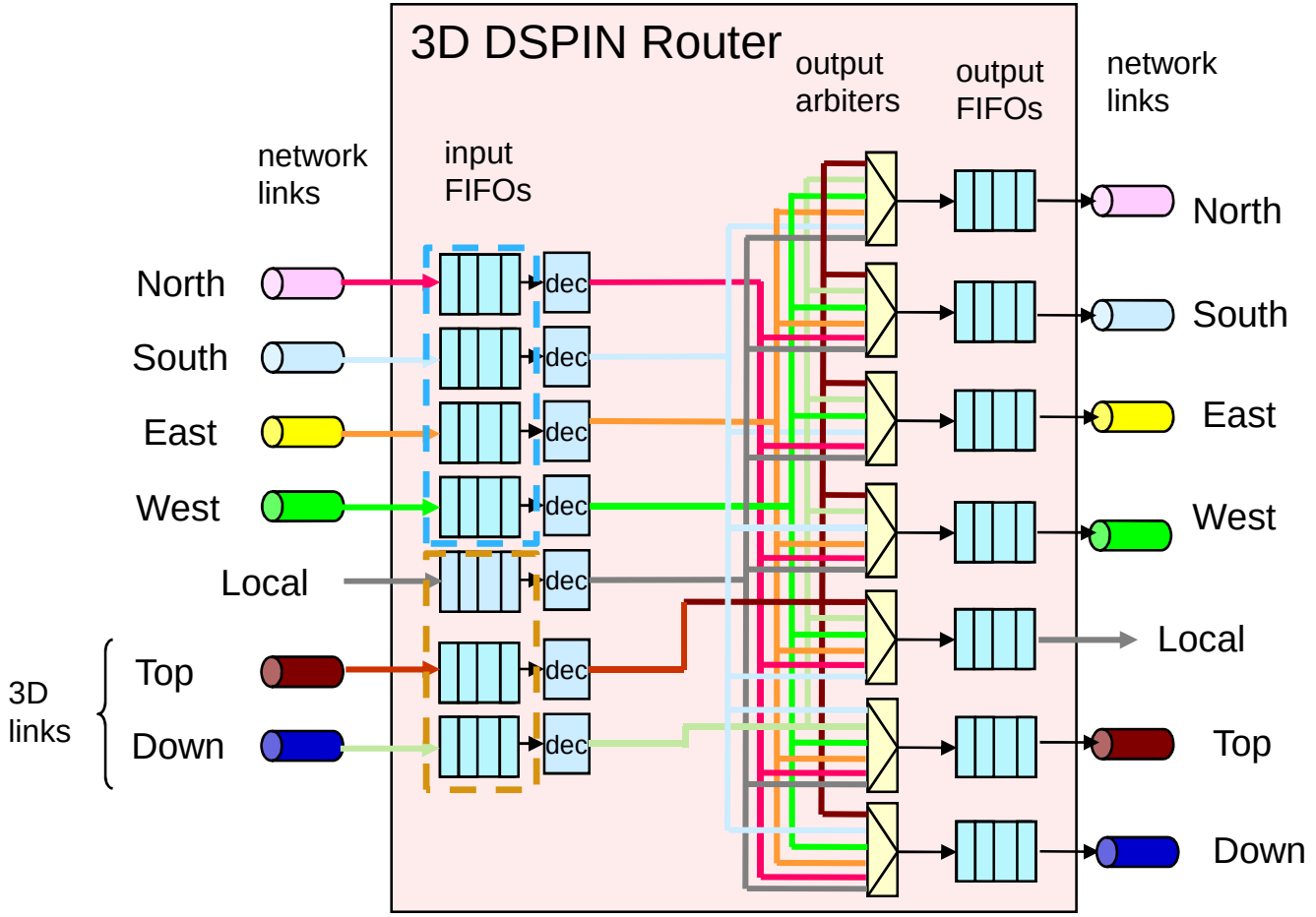
L2->L3 (req+rsp) + L3->Ext Mem (req + rsp)



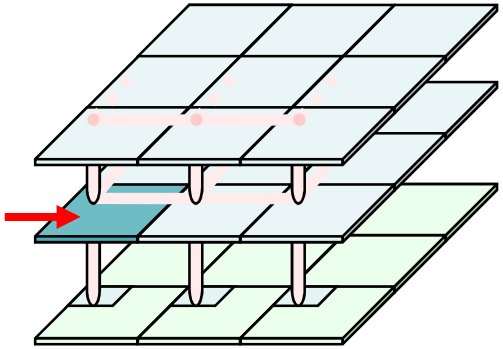
mesochronous FIFOs

bisynchronous FIFOs

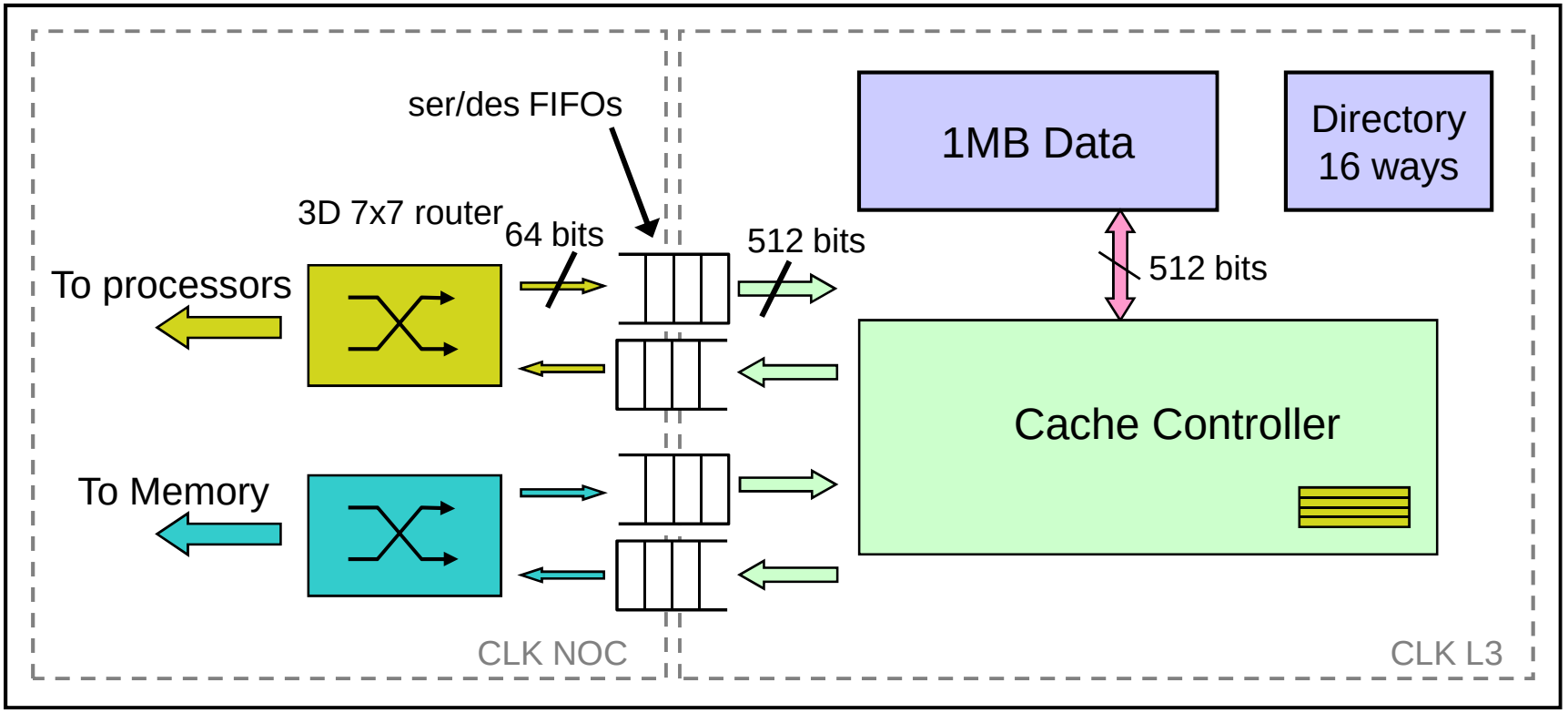
Each tier has its own clock



3D Cache - Architecture



- Low frequency high throughput tile
- 1 MB cache tile

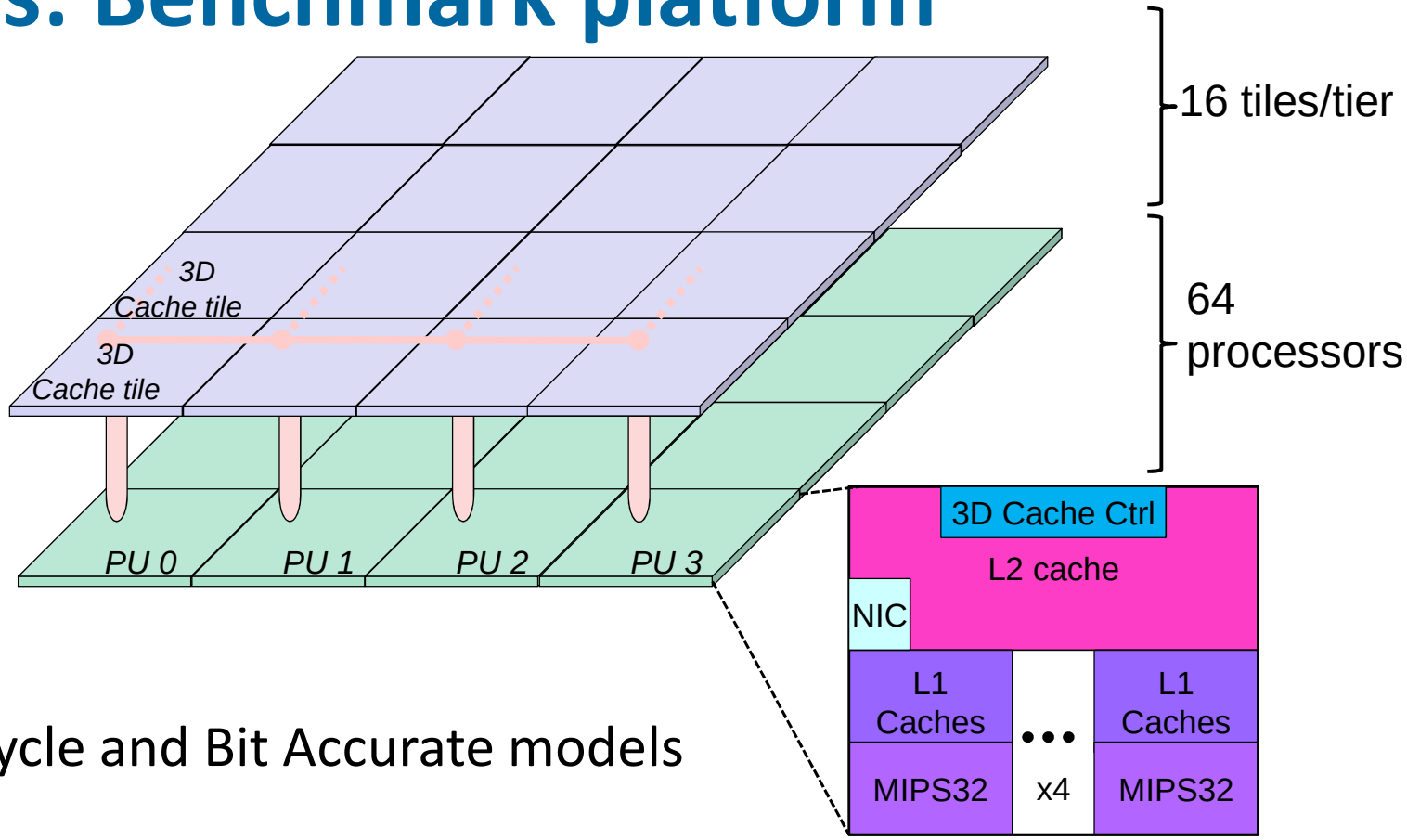


Cache Tile

Outline

- Context and motivations
  - The cube law
  - 3D technologies
- TSAR manycore architecture
  - Architecture
  - Cache Coherency
- Adaptive distributed 3D cache
  - TSAR 3D L3
  - Adaptivity
  - Architecture
- Results
  - Tile allocation
  - Fault tolerance
- Hardware implementation

Results: Benchmark platform



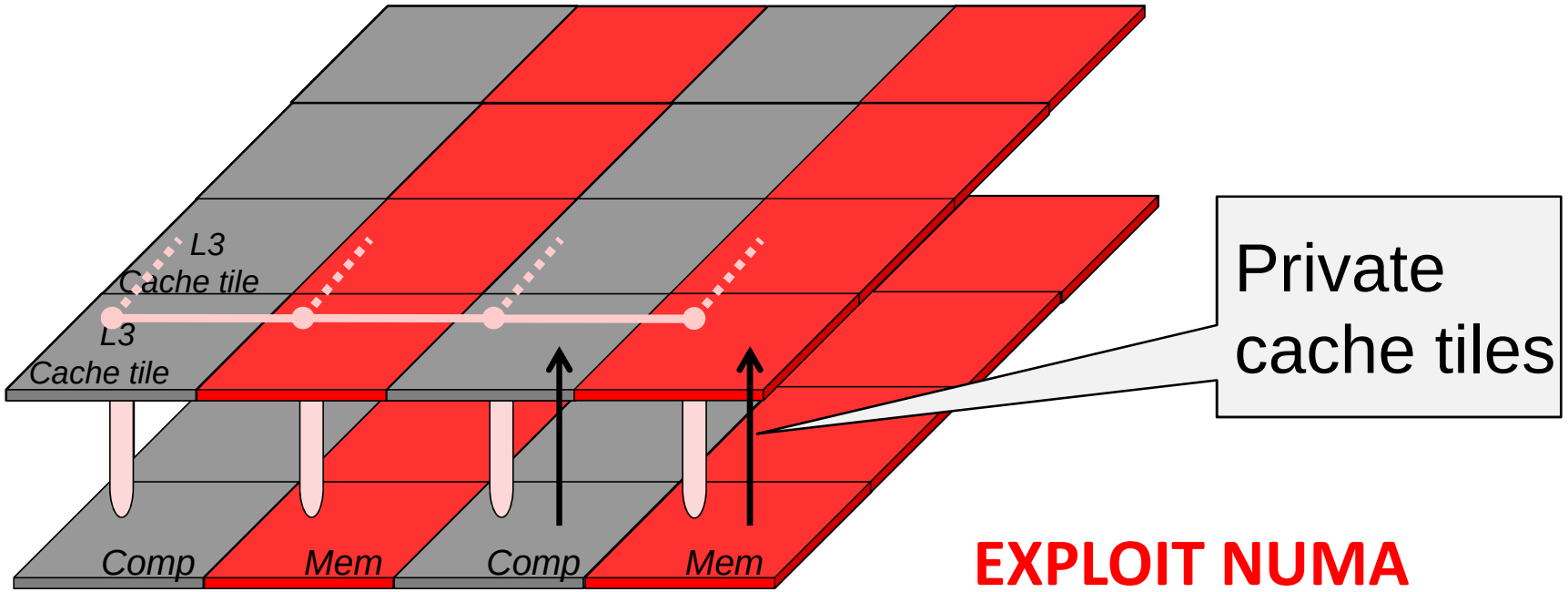
- SystemC Cycle and Bit Accurate models
  - SoClib [4]
- 16 tiles of 4 cores
- 16x1MB 3D cache tiles per memory tier
- L3 tiles frequency == Processor frequency

Results: Benchmark platform

- Bandwidth to External Memory: we emulate an equivalent 1024 cores circuit

- Splash 2 multithreaded benchmarks on MutekH [5] OS:
  - Radix: a radix sorting algorithm
  - FFT: a distributed FFT
  - LU: a matrix factorization algorithm
  - Ocean: a sea current solver

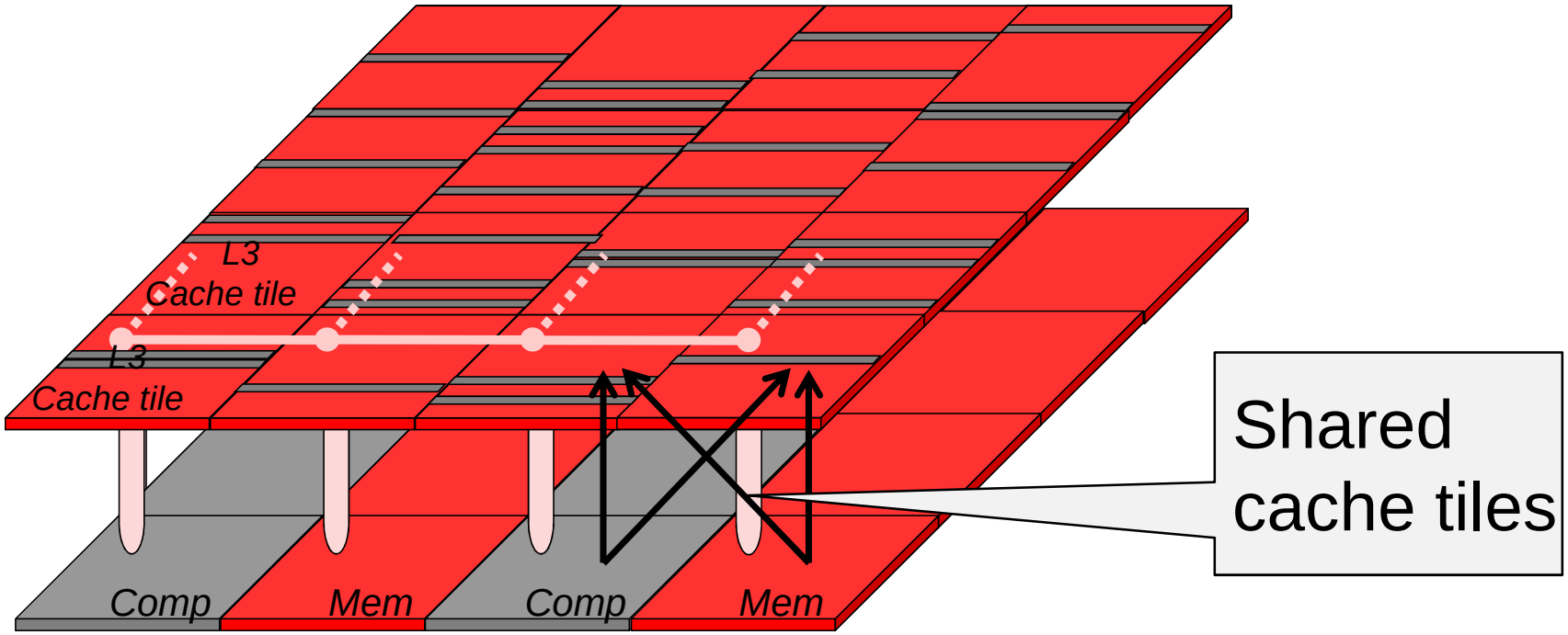
First case: Private caches



EXPLOIT NUMA ALLOCATION !

- Two applications, 32 cores each:
  - Memory intensive (red)
  - Computational (grey)
- 1st case: private 3D cache tiles

Second case: Shared caches



- 2nd case: shared 3D cache tiles
- Competition between the two applications on cache utilization

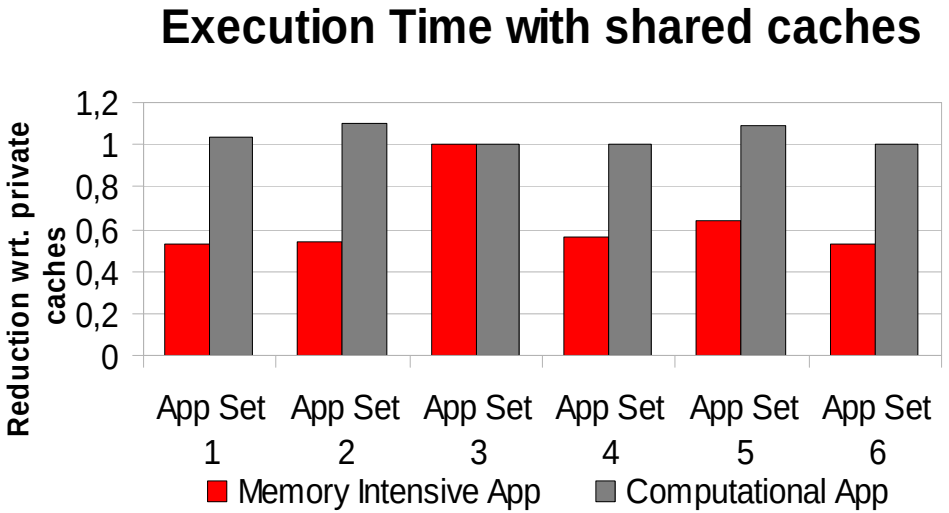
  => The most demanding application will occupy more cache space

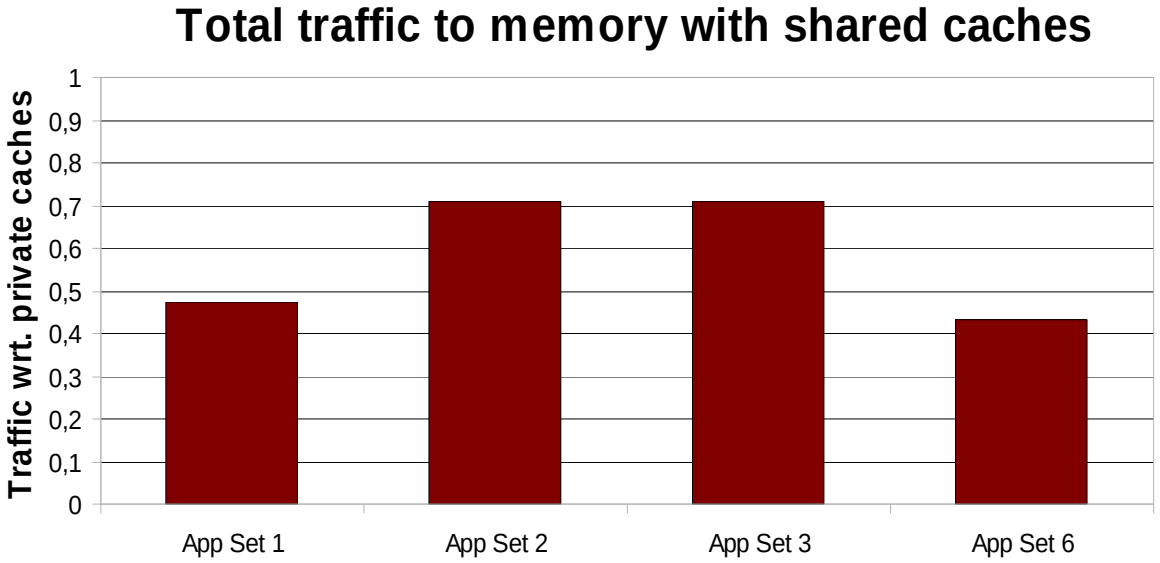
Results: Tiles Allocation

- We study execution time for **shared** caches wrt. **private** caches
- Mix of Splash FFT and Splash LU applications for test cases

The most memory intensive application performs better with shared caches

=> Quality of Service vs Best Effort





The overall traffic to the external memory is reduced by up to 55%

Results: Tiles allocation

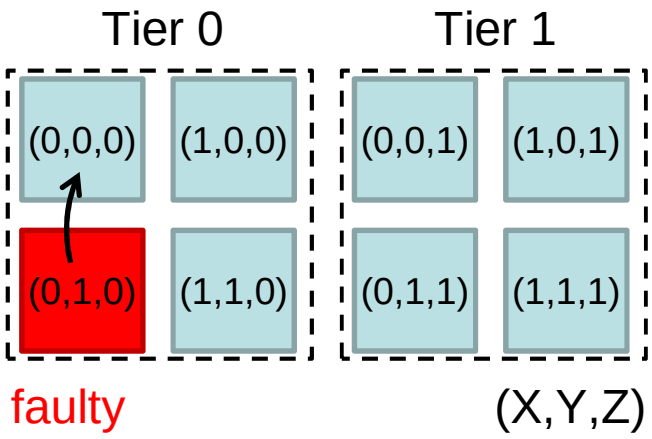
App Set	Memory Intensive App	Computational App	L3 cache
1	FFT 2^{18}	FFT 2^{14}	1 tier (16MB)
2	FFT 2^{20}	FFT 2^{18}	4 tiers (64MB)
3	LU 1024	LU 512	1 tier (16MB)
4	FFT 2^{18}	LU 512	1 tier (16 MB)
5	FFT 2^{18}	LU 1024	1 tier (16 MB)
6	FFT 2^{20}	LU 512	4 tiers (64MB)

Results: Fault Tolerance

V	Faulty	Good
1	(x_f,y_f,z_f)	(x_g,y_g,z_g)
.	…	…

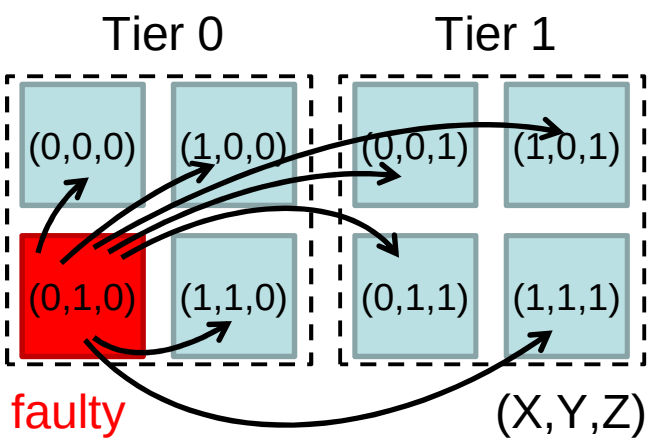
- We consider only faulty cache tiles
- A list of faulty tiles in each cache access controller and a decision algorithm
- Two possible algorithms if we select a faulty tile

First order algorithm



VERSUS

Second order algorithm



Results: Fault Tolerance

- We compare the two algorithms while we introduce faults
- 1 tier 3D cache (16 tiles) with FFT 2^{18}
  - ⇒ 1 faulty tile: 13.5% execution time increase for first order algo versus 2.9% for second order
- 4 tiers 3D cache (64 tiles) with FFT 2^{20}
  - ⇒ 1 faulty tile: 1.8% execution time increase for first order algo versus 0% for second order
  - ⇒ 4 faulty tiles: 9.3% execution time increase for first order algo versus 2.7% for second order

Outline

- Context and motivations
  - The cube law
  - 3D technologies
- TSAR manycore architecture
  - Architecture
  - Cache Coherency
- Adaptive distributed 3D cache
  - TSAR 3D L3
  - Adaptivity
  - Architecture
- Results
  - Tile allocation
  - Fault tolerance
- Hardware implementation

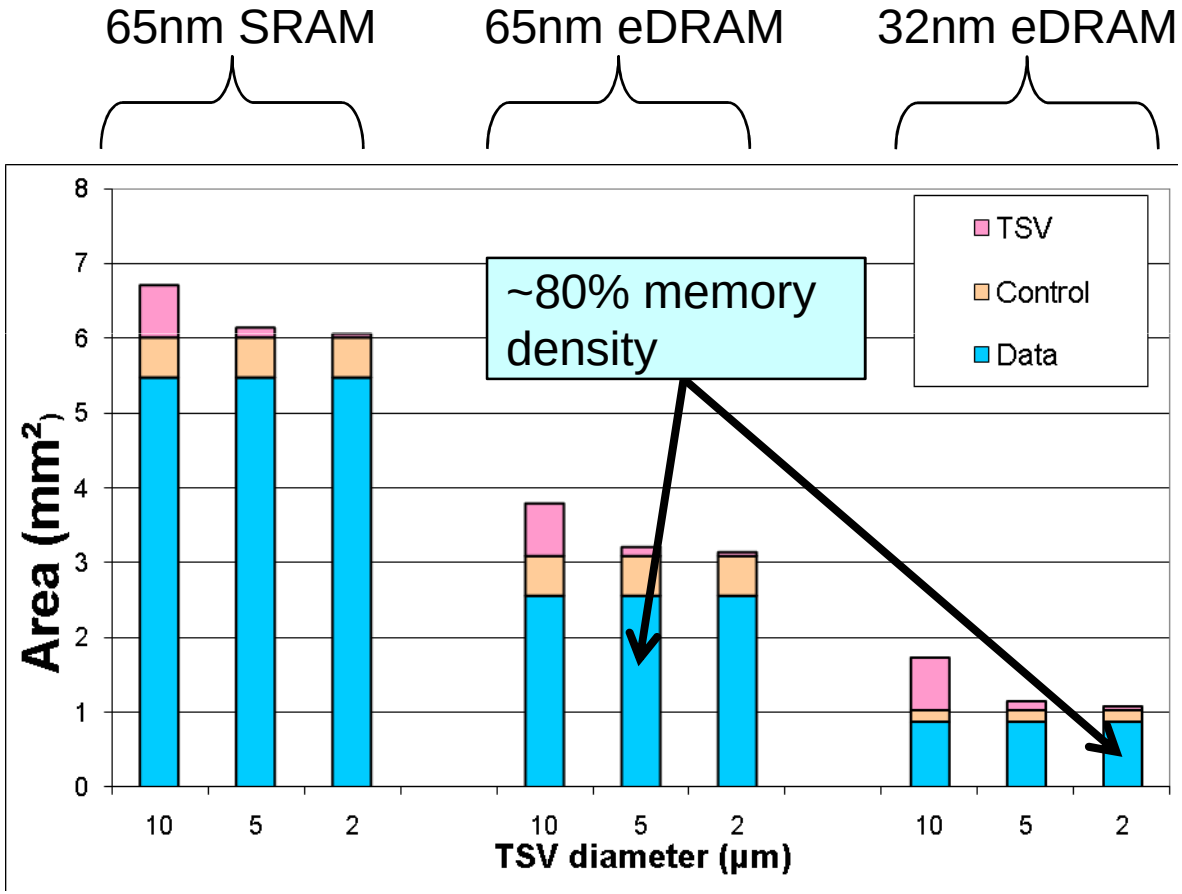
Hardware Implementation - Synthesis

- Synthesis of a 1MB cache tile in 65nm with SRAM memories
  ⇒ Extrapolation to 32nm and eDRAM
- 3 TSV sizes : 10µm (WideIO), 5µm and 2µm
- Power TSVs: according to the Wide IO example
  ⇒ 68 TSVs, 10µm diam. (272 for 5µm and 1700 for 2µm)
- Signal TSVs: 284 for the 3D NoCs (worst case tile)
  - L2/L3 NoC
  - L3/ExtMem NoC

Hardware Implementation - Synthesis

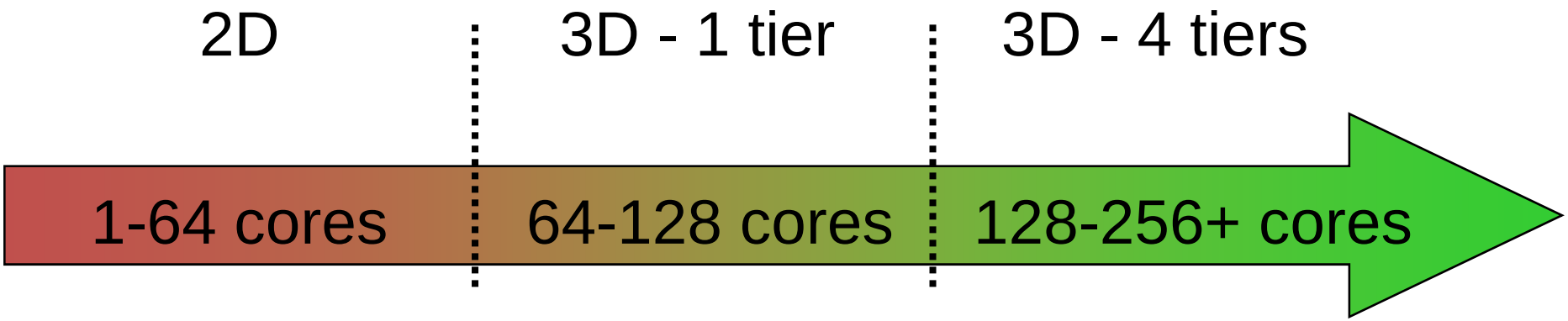
- Synthesis of a 1MB cache tile in 65nm with the two 3D NoCs

- TSV arrays:
  - 3D NoC signals
  - Memory tier power supply
- Control:
  - Directory
  - Logic
  - NoC routers
- Data:
  - Cached data



Conclusion

- How many cores in a manycore architecture without saturating memory access ?
  - TSAR, NUMA cache coherent manycore
  - 4 x 2133 MHz DDR3 interfaces
  - Constant computing density (GFLOPS/mm²)
  - HPC application



Conclusion

- The memory bandwidth is the limiting factor for manycore architectures

- 3D big caches are a solution
  - Even if they have only a $S^{0.5}$ impact on memory accesses
- We propose a low power and high performance 3D cache architecture with
  - Adaptivity
  - Fault tolerance
- A 28nm hardware implementation has been performed with low 3D overhead

References

[1] B. M. Rogers, A. Krishna, G. B. Bell, K. Vu, X. Jiang, et Y. Solihin, “Scaling the bandwidth wall: challenges in and avenues for CMP scaling,” SIGARCH Comput. Archit. News, vol. 37, n□. 3, p. 371–382, juin 2009.

[2] TSAR architecture: https://www-soc.lip6.fr/trac/tsar/

[3] Guthmuller, E.; Miro-Panades, I.; Greiner, A.; , “Adaptive Stackable 3D Cache Architecture for Manycores,” VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on , vol., no., pp.39-44, 19-21 Aug. 2012

[4] SoClib project: http://www.soclib.fr

[5] MutekH Operating System: http://www.mutekh.org

leti

LABORATOIRE D'ÉLECTRONIQUE
ET DE TECHNOLOGIES
DE L'INFORMATION

list

LABORATOIRE D'INTÉGRATION
DES SYSTÈMES
ET DES TECHNOLOGIES

Thank you

INSTITUT CARNOT CEA LETI

MINATEC

DE LA RECHERCHE À L'INDUSTRIE
cea

INSTITUT CARNOT CEA LIST

digiteo