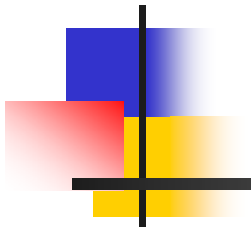
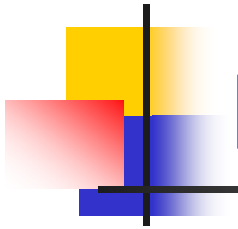


Parallel Computers

Alexandre David

1.2.05

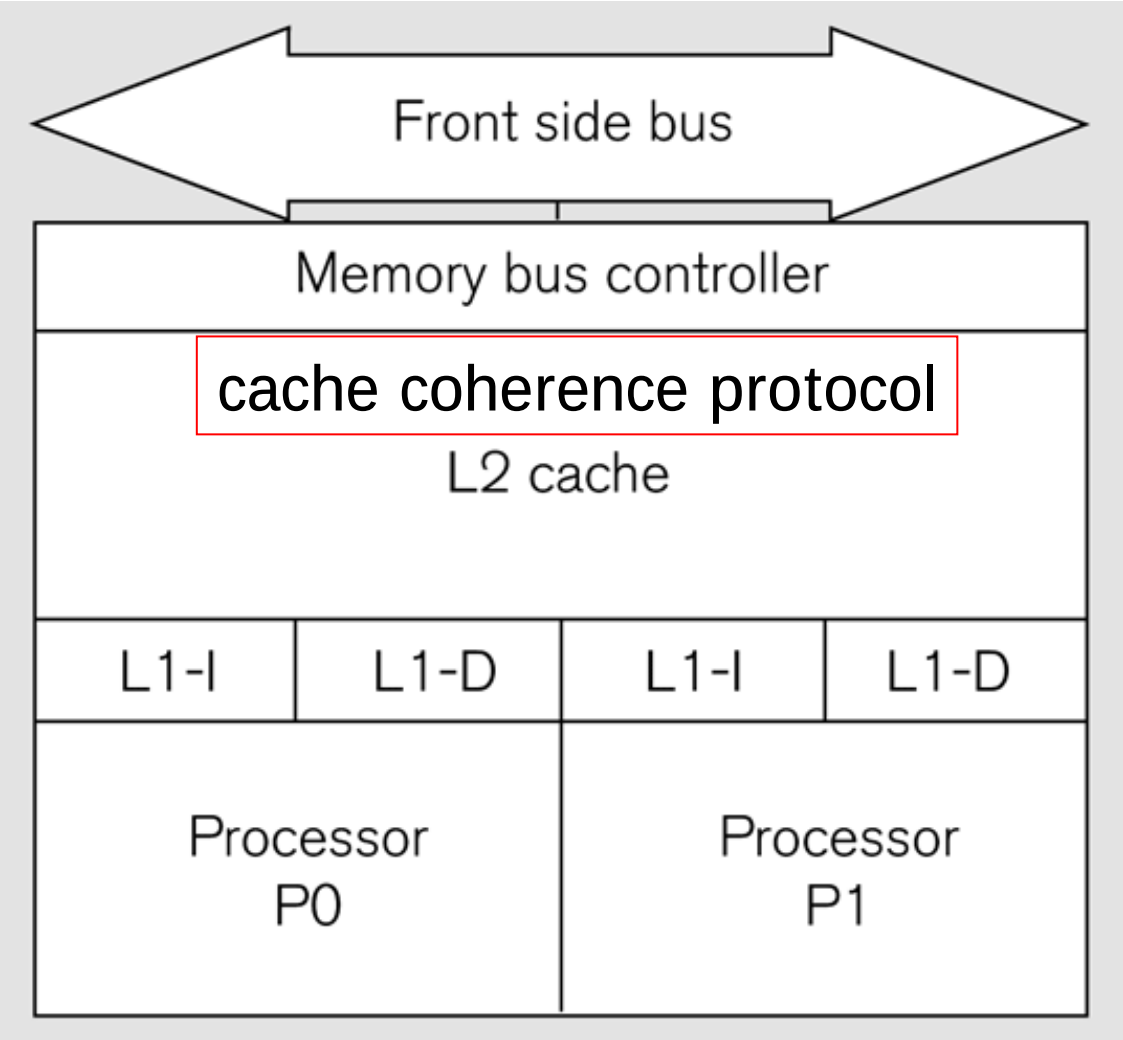
adavid@cs.aau.dk



How much do we need to know?

- Important to know the architecture of parallel hardware.
- Not all details are important to programmers
  - keep portability
  - keep up with technological changes
- The point: Get a meaningful model.

Intel Core Duo



Modified
Exclusive
Shared
Invalid

more shared L2
low latency

Example

The point: Relatively expensive.

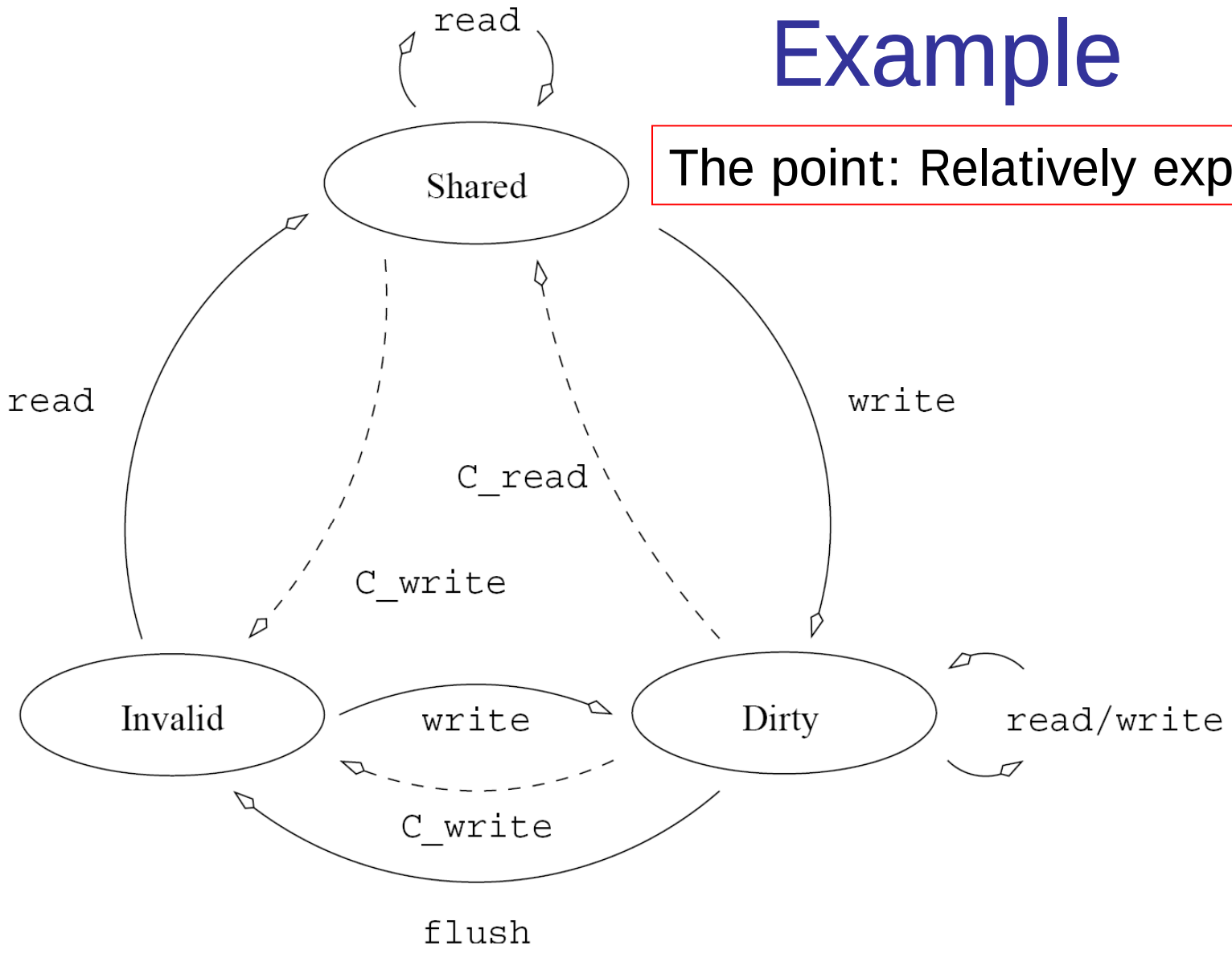
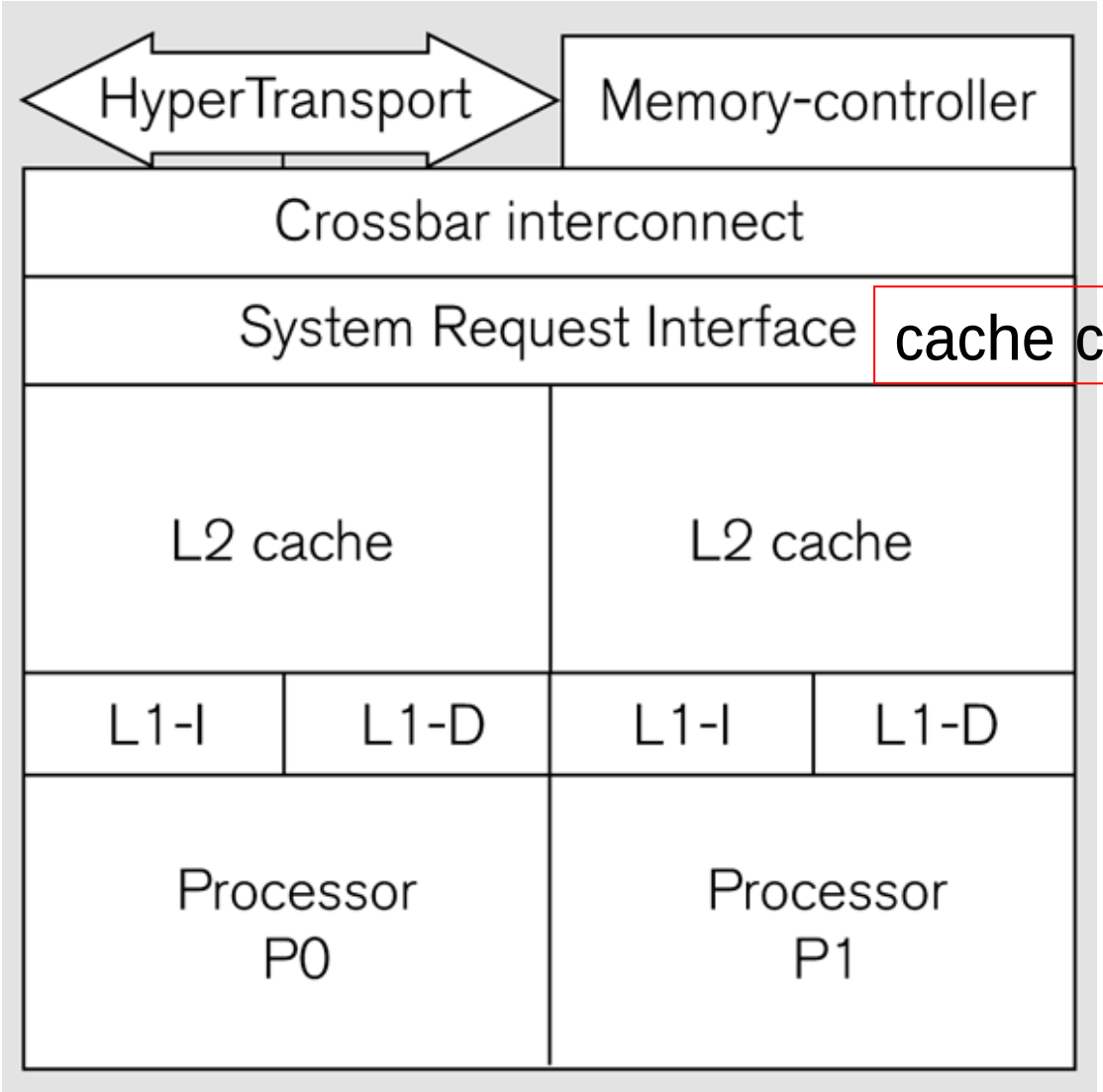


Figure 2.22 State diagram of a simple three-state coherence protocol.

AMD Dual Core Opteron

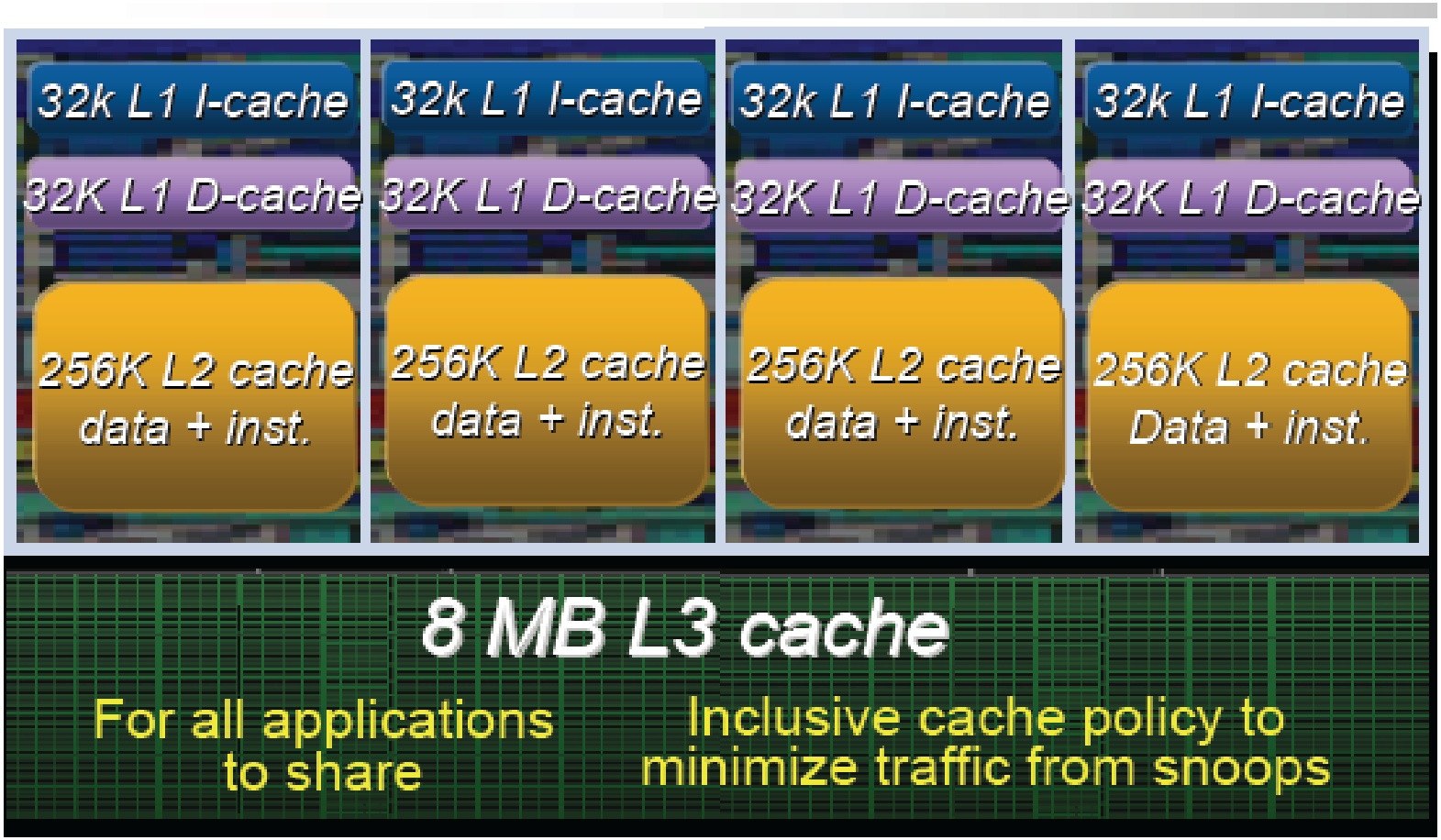


easier for SMP

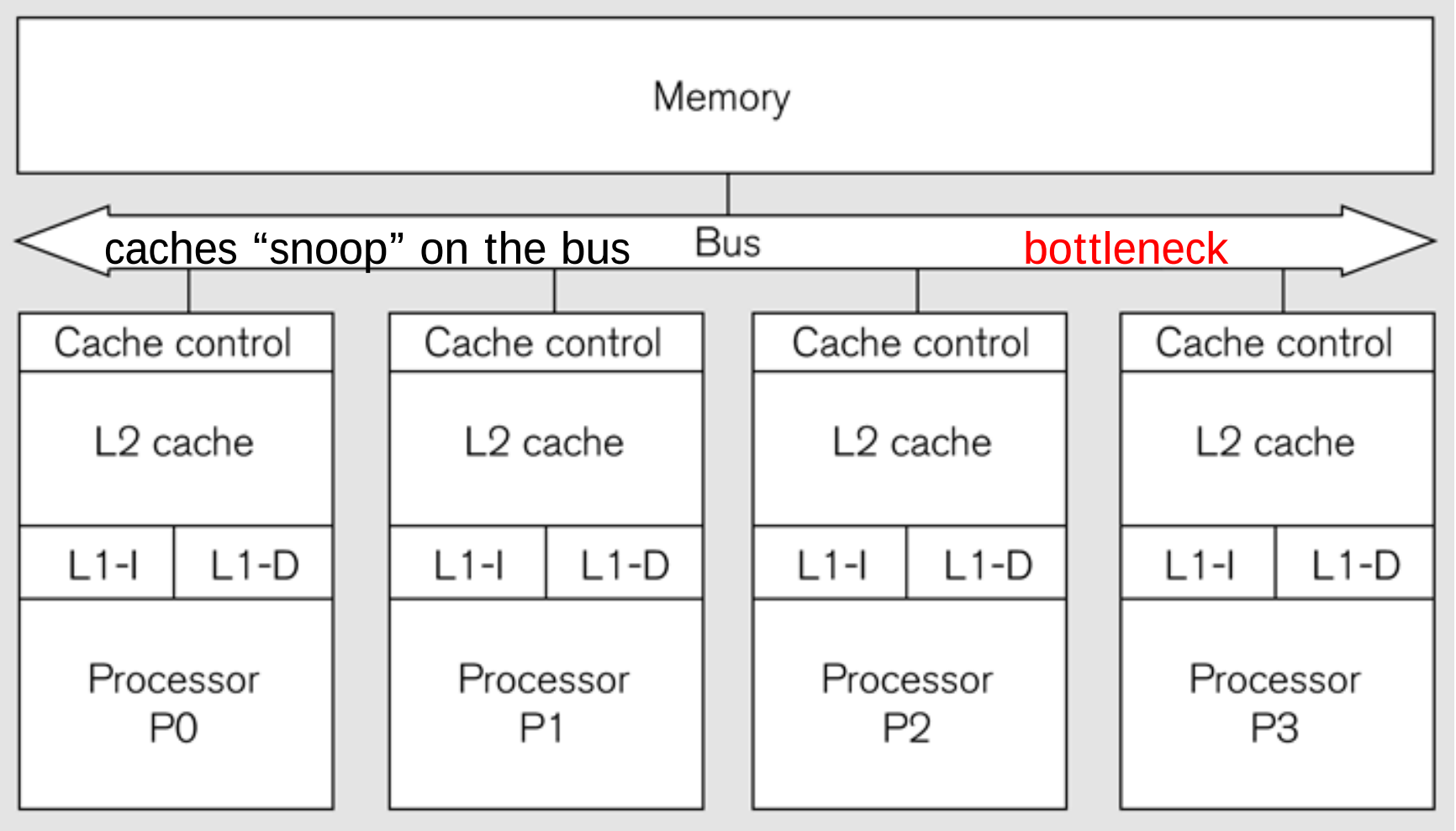
cache coherence protocol

Modified
Owned
Exclusive
Shared
Invalid

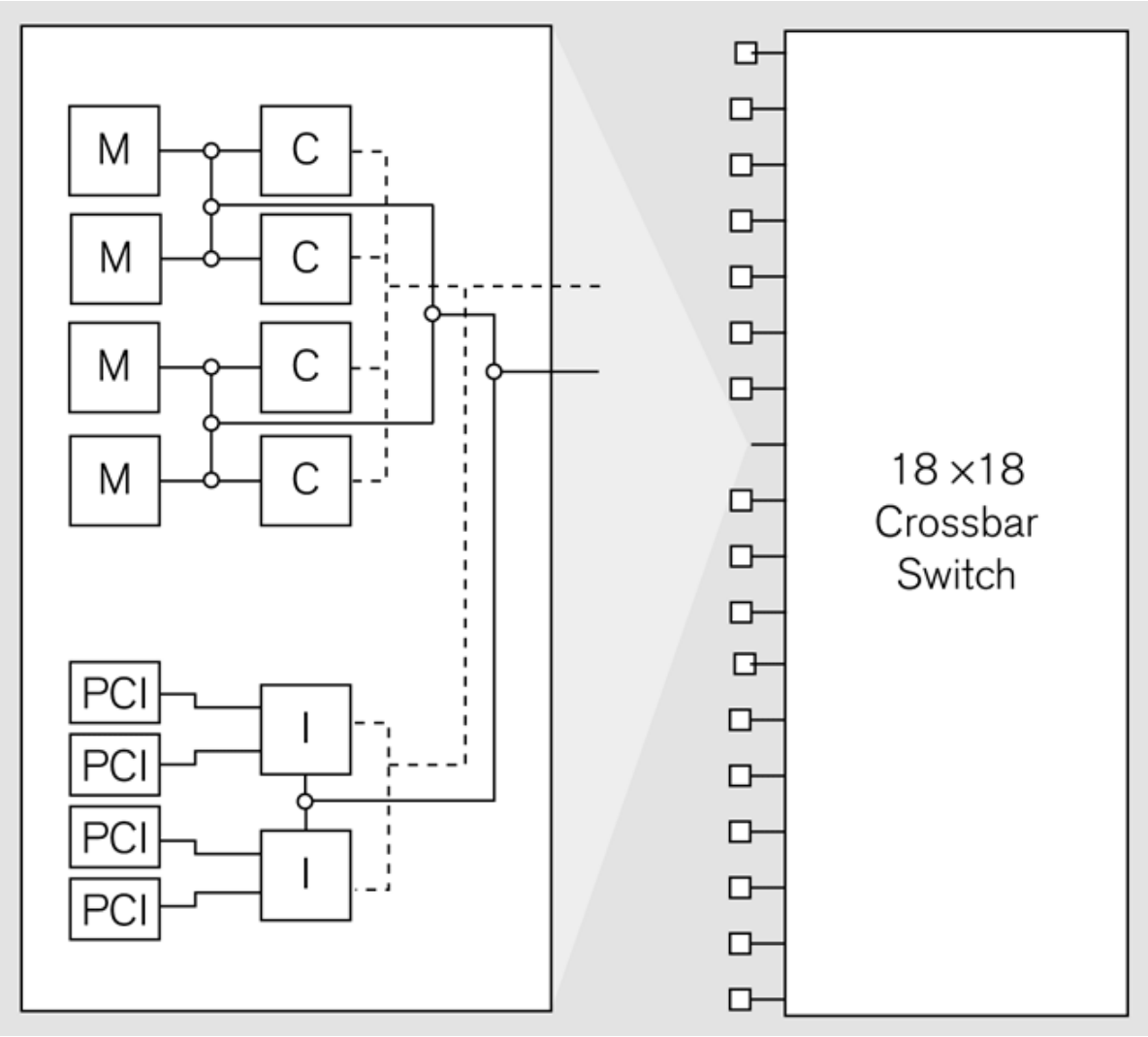
Core i7


32k L1 I-cache
32K L1 D-cache
256K L2 cache
data + inst.
32k L1 I-cache
32K L1 D-cache
256K L2 cache
data + inst.
32k L1 I-cache
32K L1 D-cache
256K L2 cache
data + inst.
32k L1 I-cache
32K L1 D-cache
256K L2 cache
Data + inst.
8 MB L3 cache
For all applications
to share
Inclusive cache policy to
minimize traffic from snoops

SMP


Memory
caches “snoop” on the bus
Bus
bottleneck
Cache control
L2 cache
L1-I
L1-D
Processor
P0
Cache control
L2 cache
L1-I
L1-D
Processor
P1
Cache control
L2 cache
L1-I
L1-D
Processor
P2
Cache control
L2 cache
L1-I
L1-D
Processor
P3

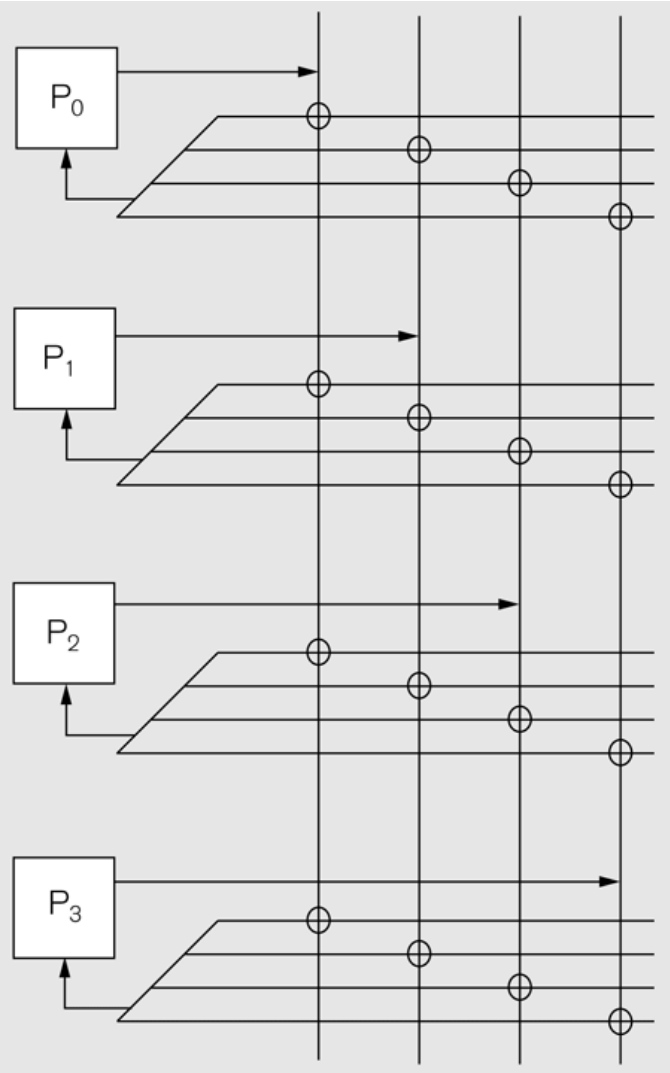
Larger SMP – Sun Fire



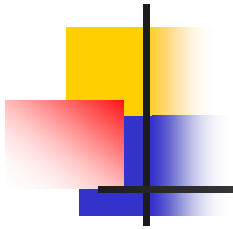
18 boards connected by a crossbar switch.
Snooping buses.
Directory based cache coherence protocol.
Scalable/higher latency.

Note: Expensive hardware.

Crossbar



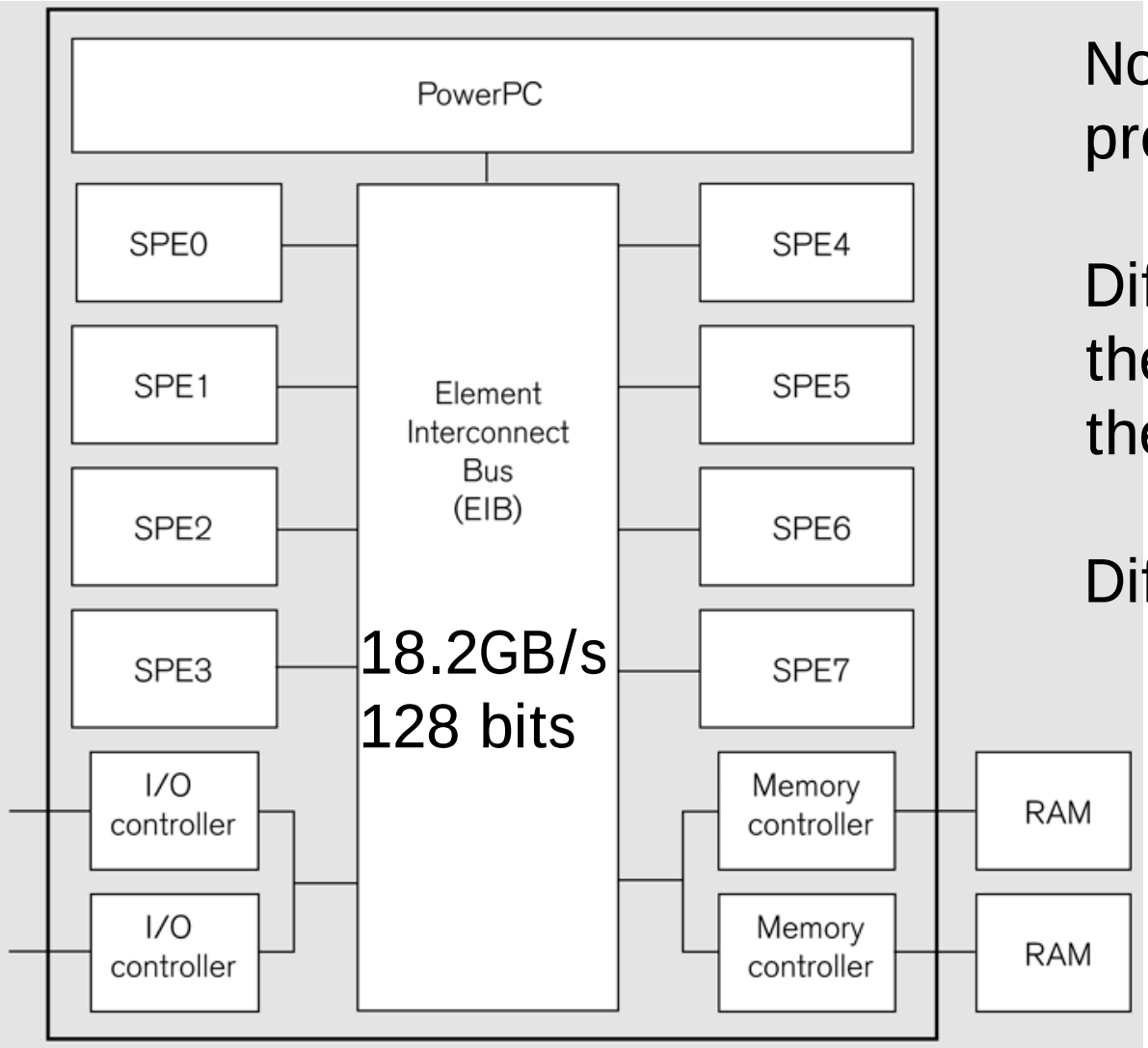
- N x N connections.
- Expensive, limited.



Heterogeneous chips

- GPUs
  - 800 ALU on ATI's latest 4800 series.
  - --logic, ++computational units
- FPGAs
  - PCI boards available
  - reconfigurable
- Cell
  - Dual-threaded PPC – PPU, 64 bits
  - 8x SPU

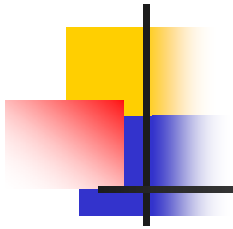
Cell architecture



No cache coherence protocol.

Different philosophy: the PPU is a coordinator, the SPUs do the job.

Difficult to program.

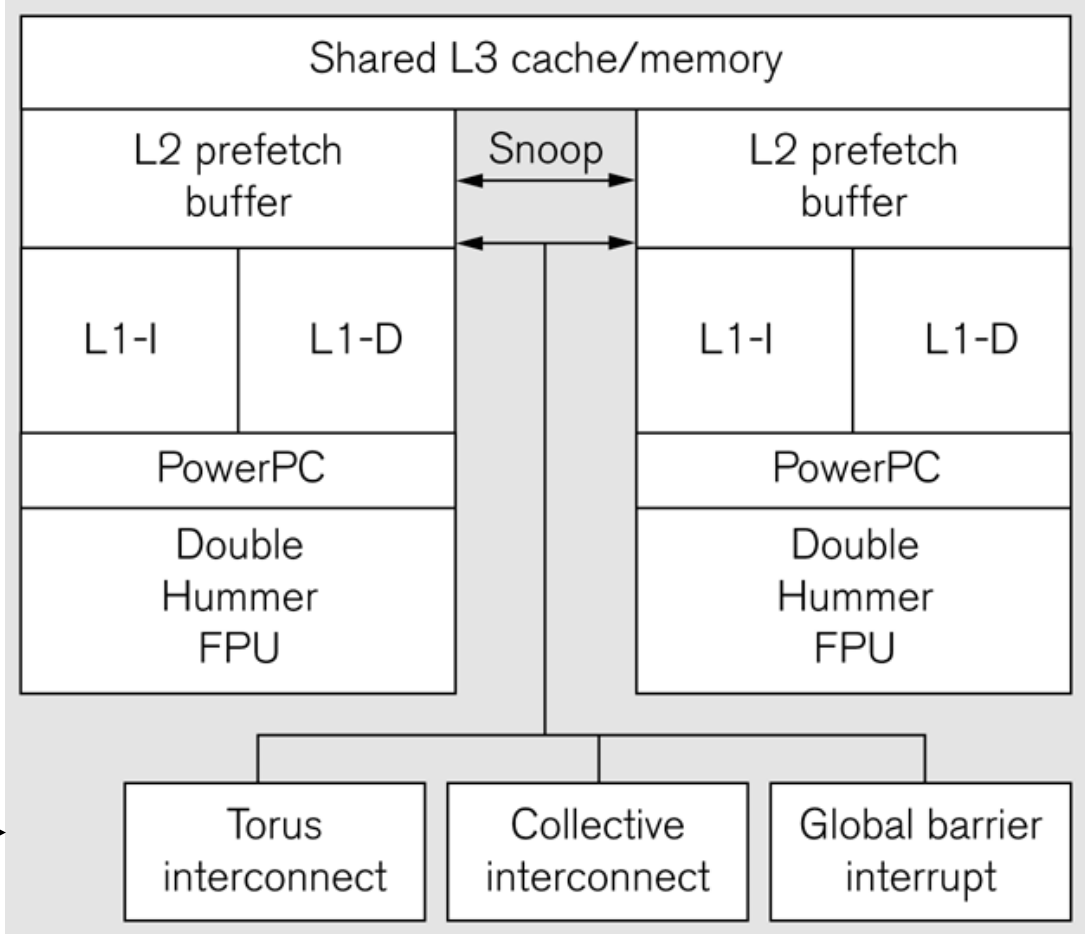


Clusters

- “Cheap” PCs connected together.
  - GB ethernet
  - Infiniband
  - ...
  - Memory private to each machine, use message based communication.
  - Scalable but high latency.
  - Sold by racks.

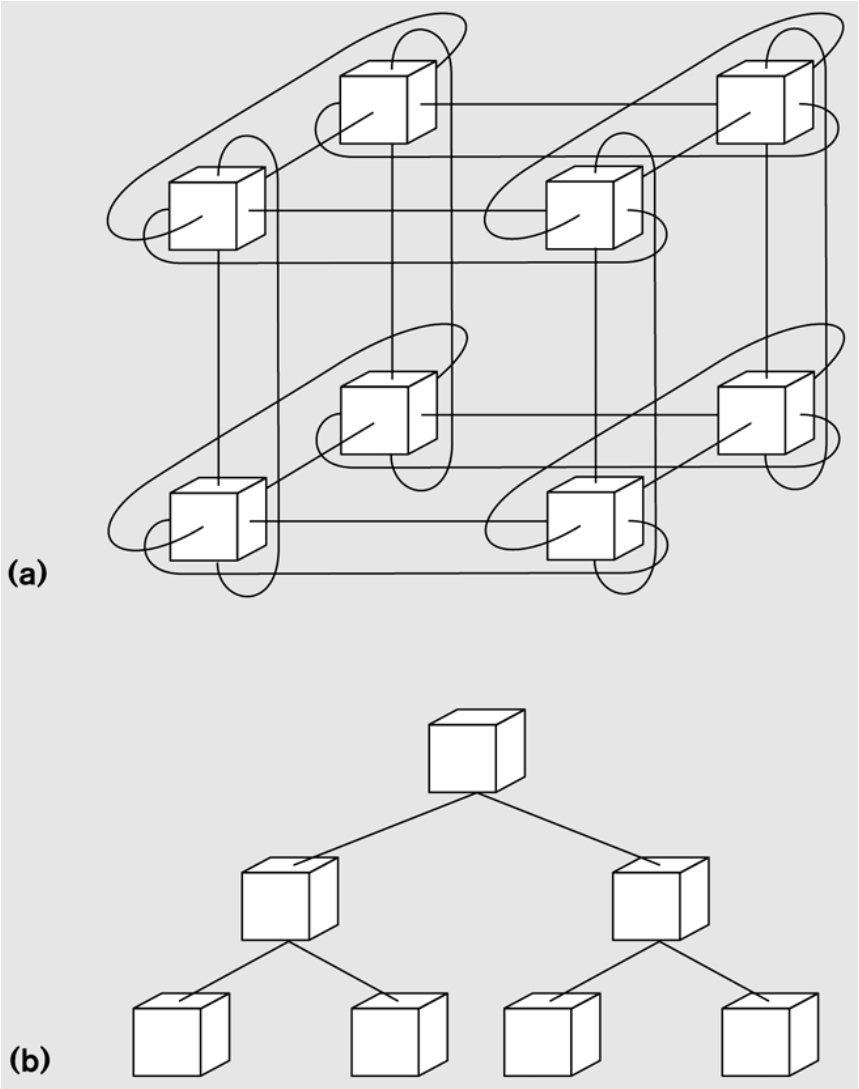
BlueGene

65536 x
@ 700MHz



interesting part →

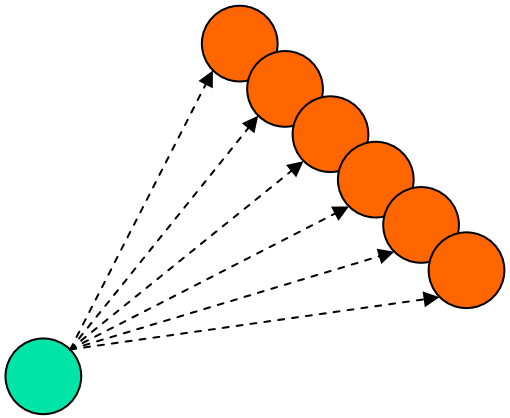
Interconnect



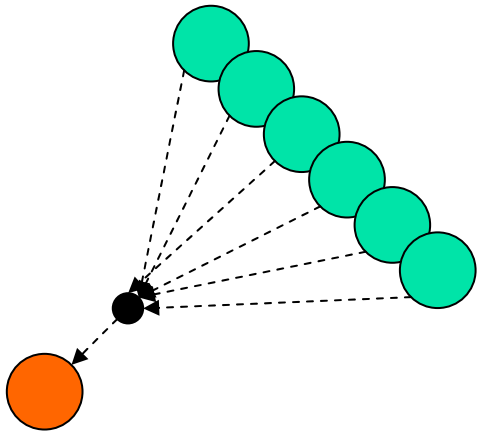
3-D torus for standard data transfers.

Collective network for fast reductions.
Very powerful.

Broadcast/Reduction



Broadcast



Reduce

Cut-through routing
Understanding communication

- Simplified packet routing:
  - Packets take the same path (1x routing information).
  - In sequence packet delivery (no sequencing).
  - Error detection at message level, cheap detection (for good networks).
  - Fixed size unit for packets = flow control digits (flits).

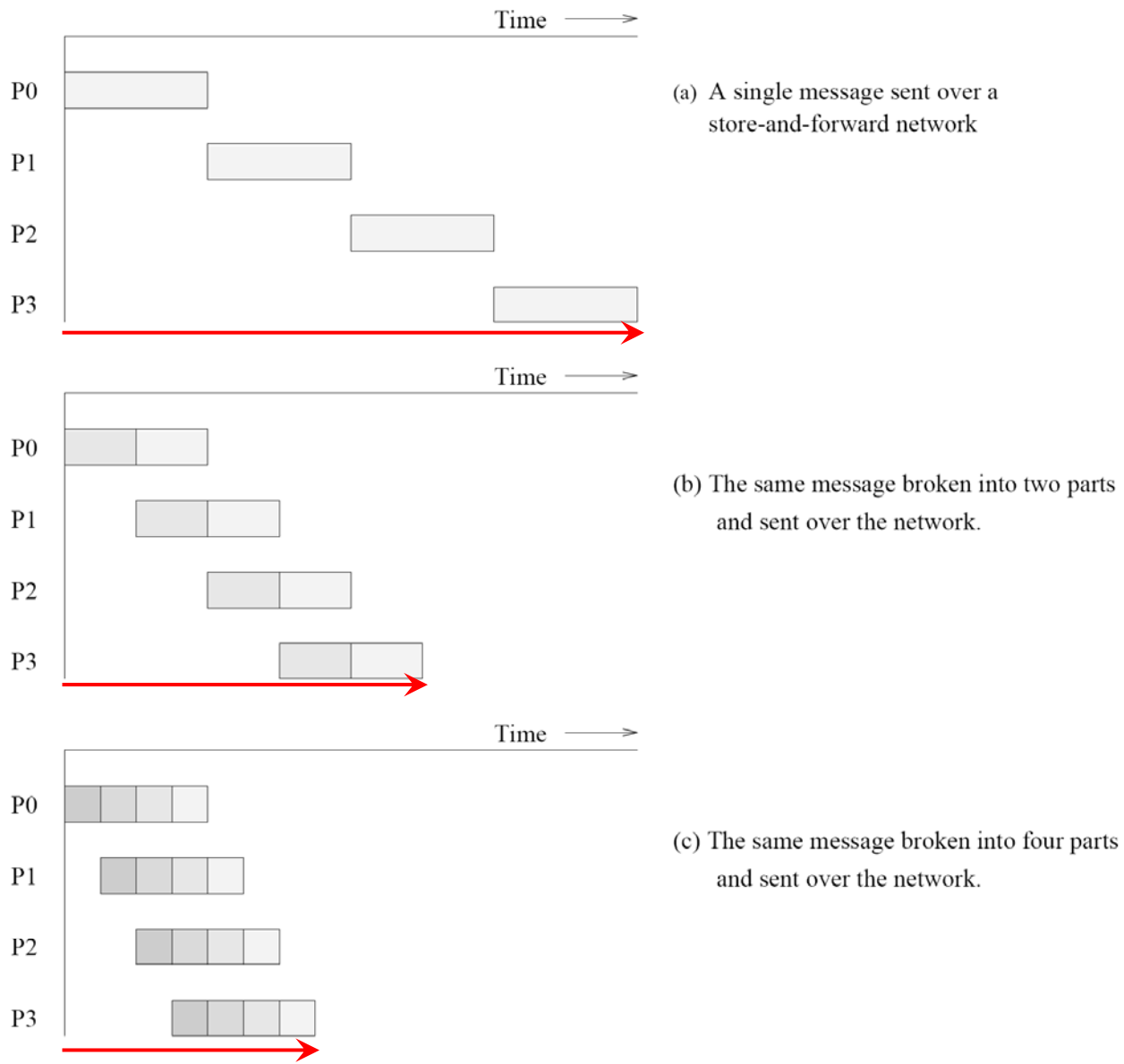
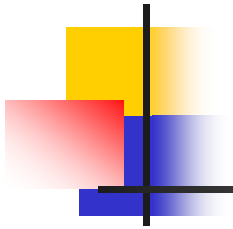
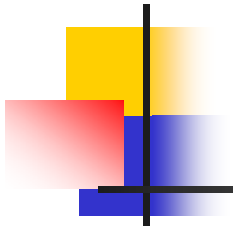


Figure 2.26 Passing a message from node P_0 to P_3 (a) through a store-and-forward communication network; (b) and (c) extending the concept to cut-through routing. The shaded regions represent the time that the message is in transit. The startup time associated with this message transfer is assumed to be zero.



Lessons

- Very different architectures.
  - SMP
  - Distributed
- But we want one meaningful model.
- Hints:
  - local accesses - cheap
  - non-local accesses - expensive



RAM model

- Sequential execution unit with unbounded memory.
  - every operation takes 1 unit of time
- Limited
  - ok for algorithms – reason on complexity
  - unrealistic

Application of the RAM model

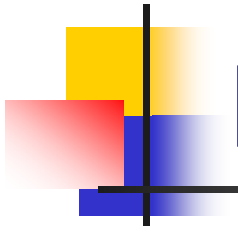
```
1  location=-1;
2  for(j=0; j<n; j++)
3  {
4    if(A[j]==searchee)
5    {
6      location=j;
7      break;
8    }
9  }
```

```
 1  location=-1;
 2  hi=n-1;
 3  lo=0;
 4  while(lo!=hi)
 5  {
 6    mid=lo+floor((hi-lo+1)/2);
 7    if(A[mid]==searchee)
 8      break;
 9    if(A[mid]>searchee)
10      hi=mid;
11    else
12      lo=mid+1;
13  }
```

Expected: O(n), O(log n)

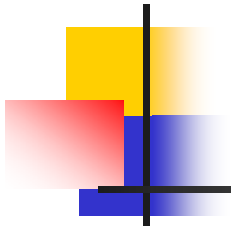
update of `location` missing

(array must be sorted)



PRAM model

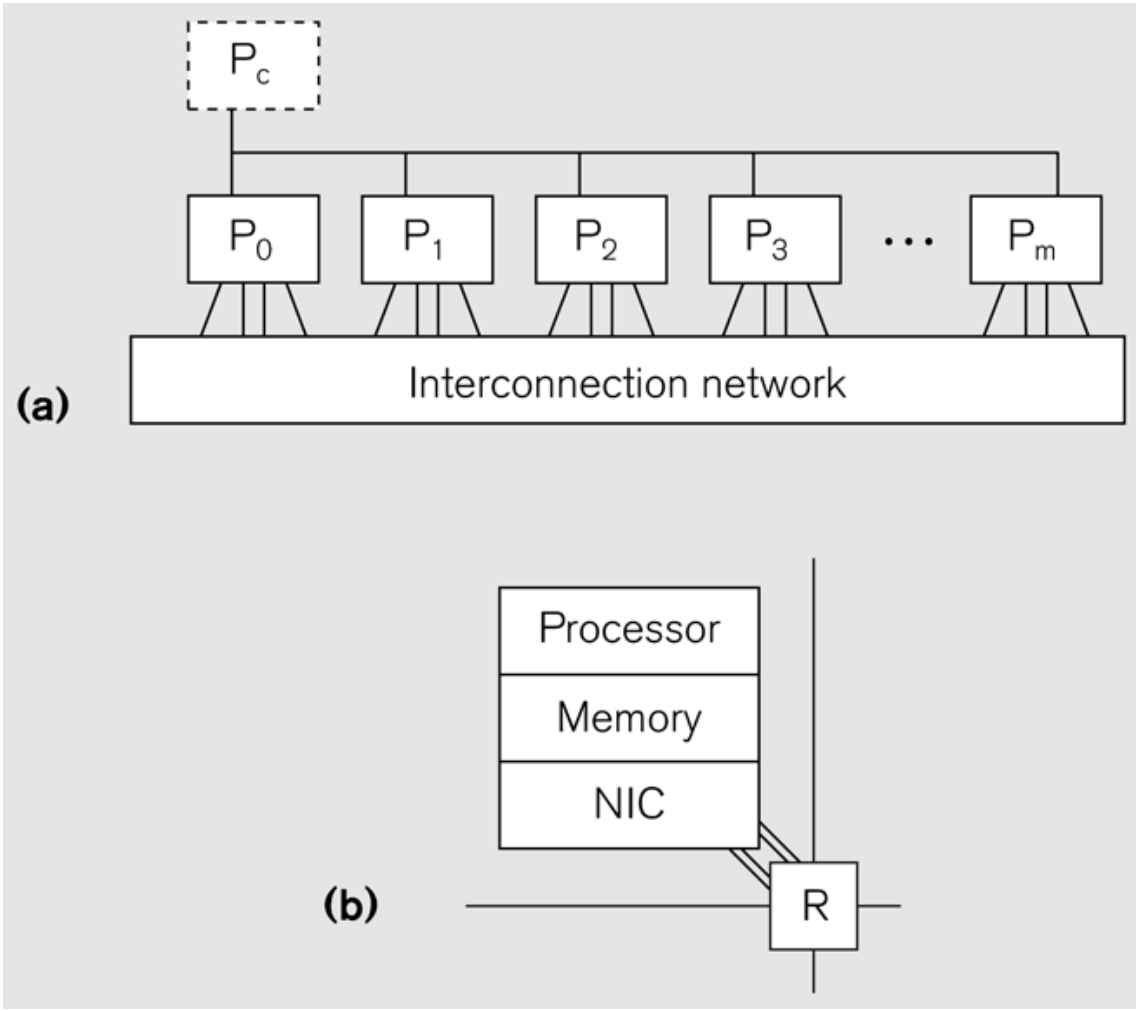
- Several execution units accessing one shared unbounded memory
  - global access
  - synchronous access – one global clock
  - contention resolved by pre-defined rules
    - EREW, CREW, CRCW, ERCW
    - least powerful, least convenient: EREW
    - most powerful, most convenient: CRCW
    - lesson: reason on CRCW but apply on EREW because it is possible to simulate one with the other (in polynomial time)
  - like RAM: good for algorithms, complexity...



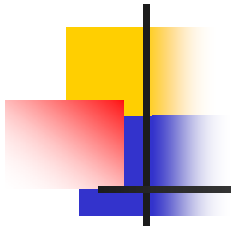
CTA (Candidate Type Architecture)

- Account for communication costs.
  - Applies to clusters & SMPs.
  - Local/non-local accesses.
  - Goal: Achieve in practice the predicted running time. PRAM is misleading in that respect.
  - The catch: Not easy to estimate communication costs.
- Model:
  - interconnected processors with RAM
  - topology not specified but this impacts communication costs.

CTA



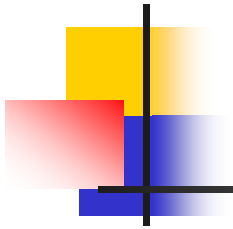
SMP
Cluster
Cell
...
Memory latency specified in function of the real architecture.
Non-local: λ.



Typical λ

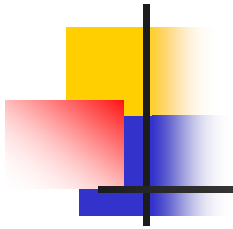
Architecture Family	Computer	Lambda
Chip Multiprocessor*	AMD Opteron	100
Shared-memory Multiprocessor	Sun Fire E25K	400–660
Co-processor	Cell	N/A
Cluster	HP BL6000 w/GbE	4,160–5,120
Supercomputer	BlueGene/L	8960

*CMP's λ value measures a transfer between L1 data caches on chip.



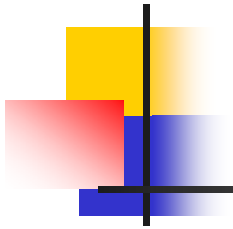
Lesson

- Use locality
  - temporal & spatial
  - sometimes redundant computation is better than sending data around
- Exact number of processors supplied at runtime.
  - scale/not tied to one setup
  - Note: λ increases with P.



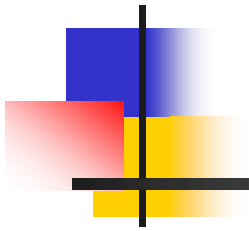
Memory reference mechanisms

- Shared memory
  - avoid race conditions, needs synchronization
- One-sided
  - not common
  - private (local) & shared non-coherent memory
- Message passing – 2-sided
  - MPI
  - Complex communication protocols.



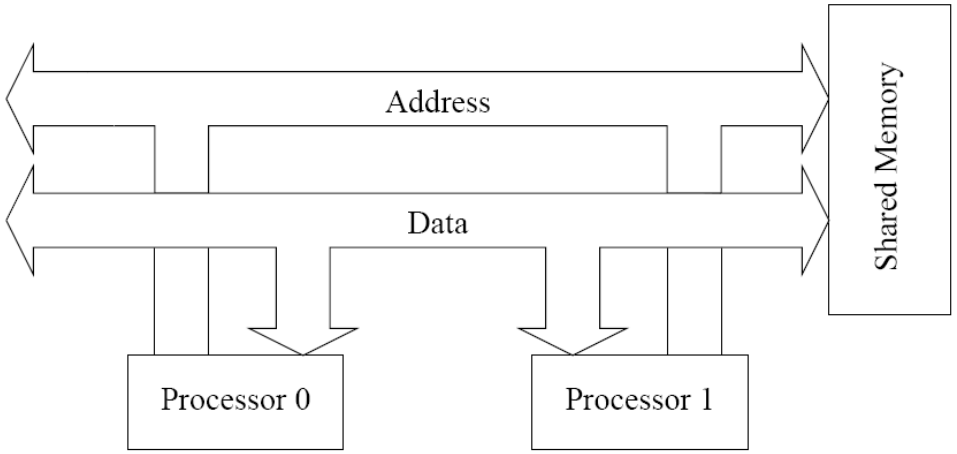
Memory consistency models

- Sequential consistency – expensive.
  - serialize the operations of all processors
  - operations obey specified order
- Relaxed consistency – weaker.
  - variations
- Keep in mind: There are hardware tricks to get sequential consistency (CAS/TAS).



Interconnects

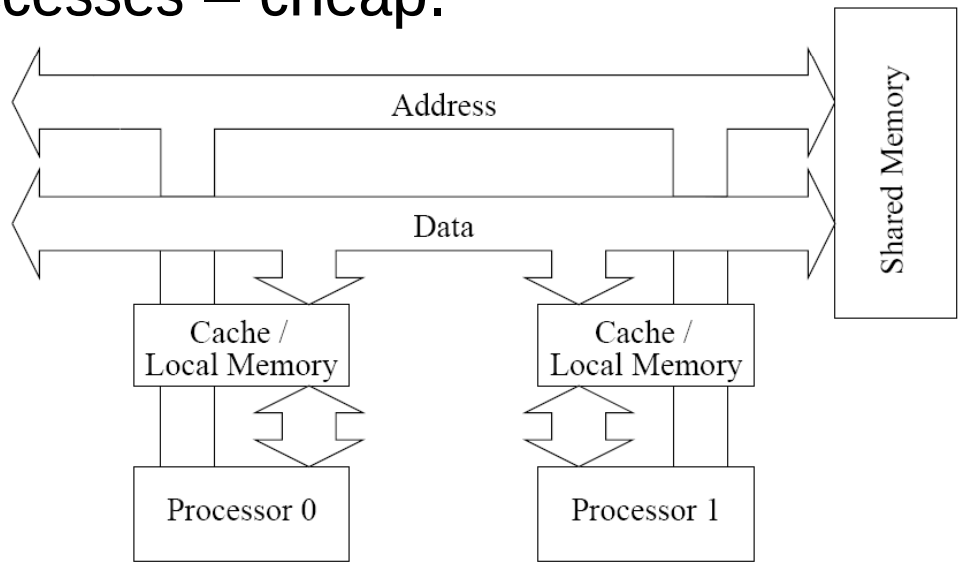
Bus Based Networks



(a)

No local cache

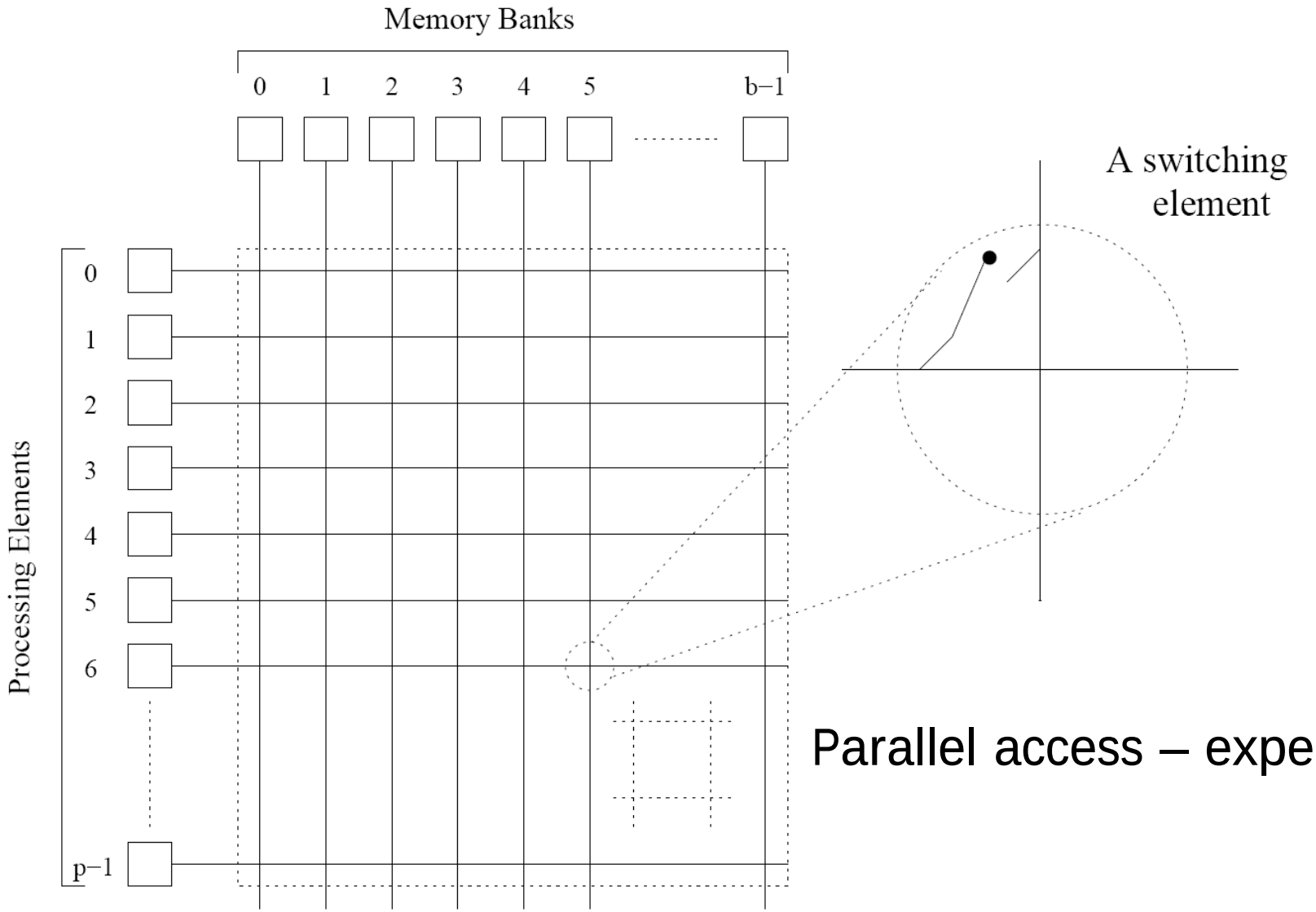
Serialize accesses – cheap.



(b)

Local cache

Crossbar Networks



Parallel access – expensive.

Omega networks

Multi-stage network – compromise cost/performance.
N nodes – log n stages.

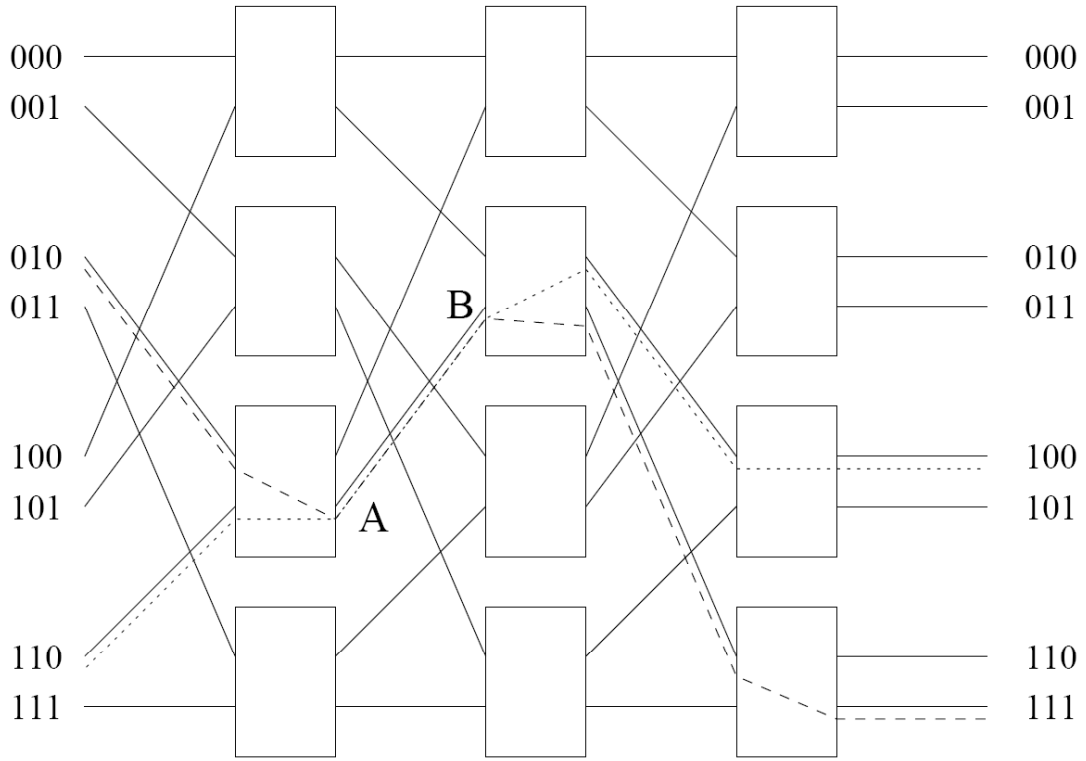


Figure 2.13 An example of blocking in omega network: one of the messages (010 to 111 or 110 to 100) is blocked at link AB.

Linear Arrays and Meshes

Figure 2.15 Linear arrays: (a) with no wraparound links; (b) with wraparound link.

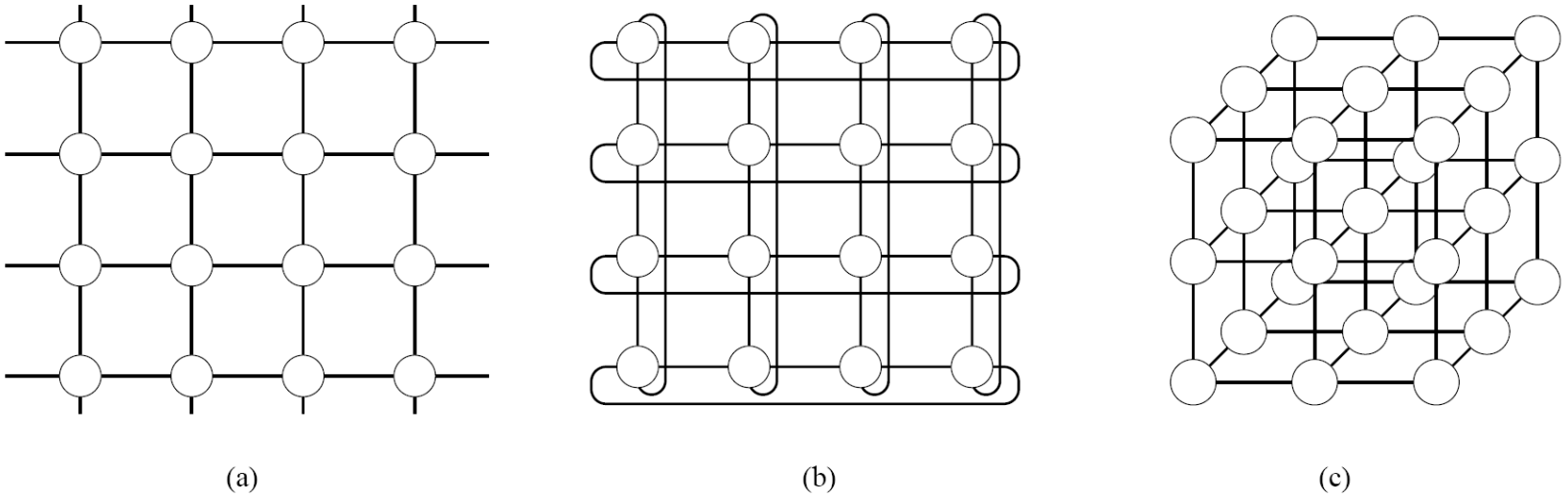
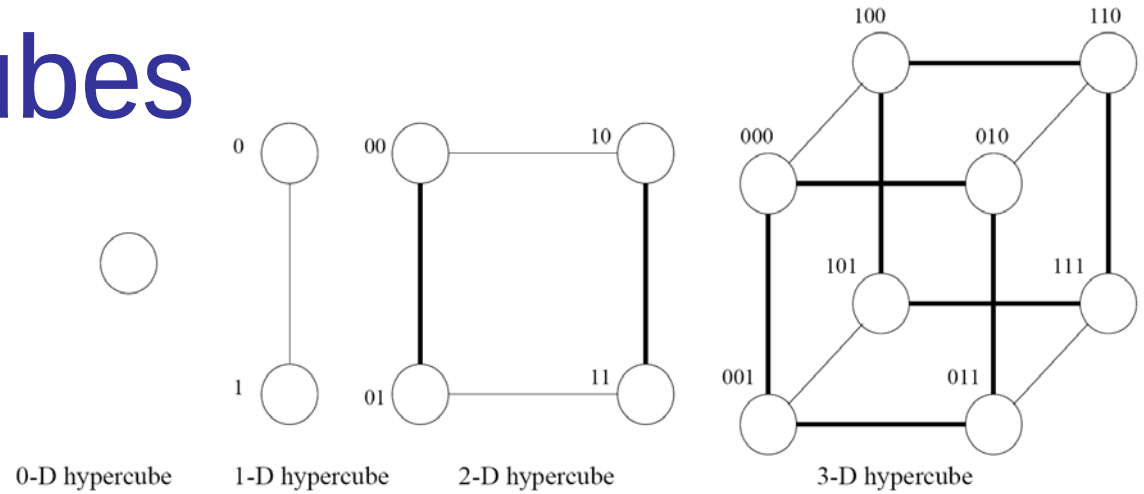


Figure 2.16 Two and three dimensional meshes: (a) 2-D mesh with no wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.

Hypercubes



2^d nodes,
d=dimension,
good routing,
relatively expensive,
low congestion

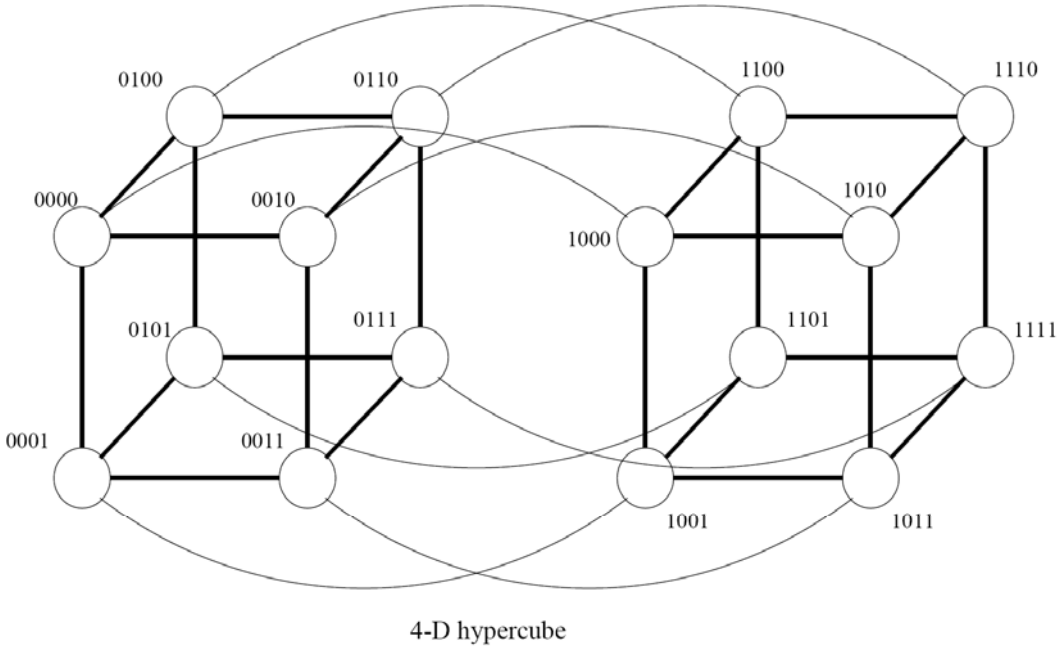


Figure 2.17 Construction of hypercubes from hypercubes of lower dimension.

Fat trees

More bandwidth where it is needed.

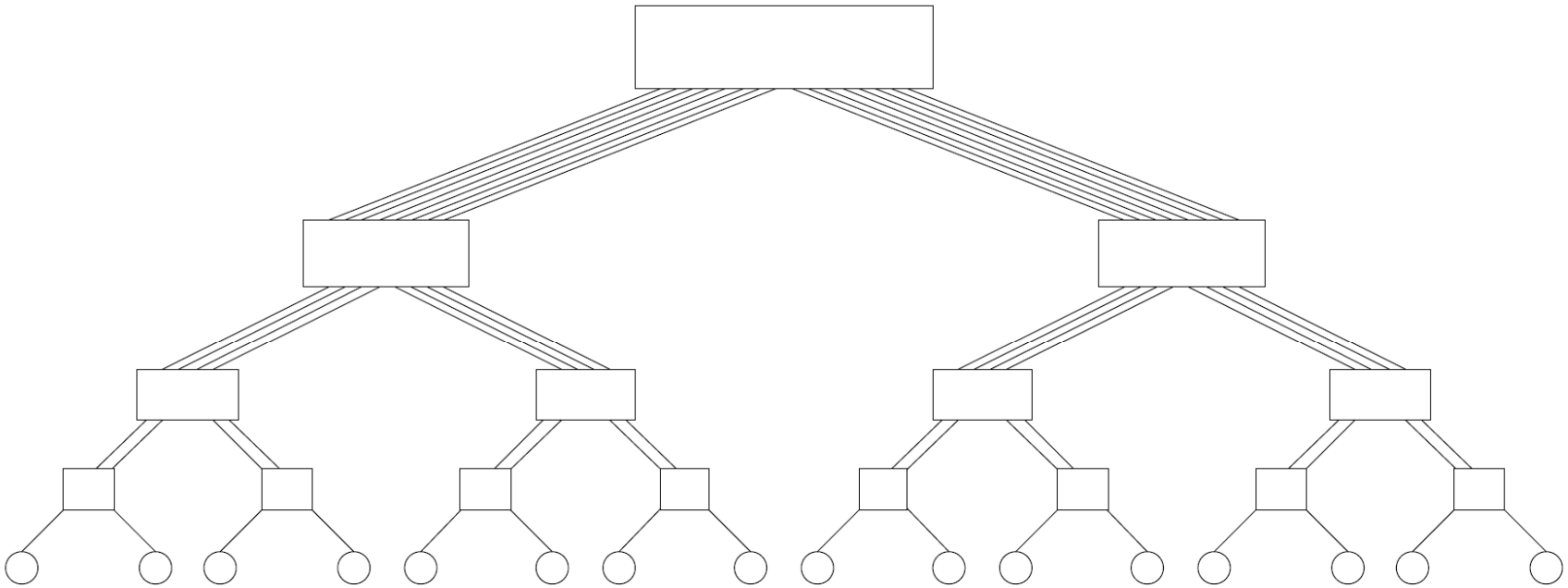
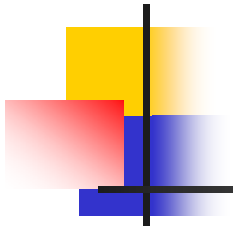
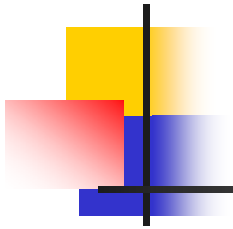


Figure 2.19 A fat tree network of 16 processing nodes.



Evaluating The Networks

- All the previous topologies have advantages and disadvantages.
- Important factors: cost and performance.
- Define criteria to characterize cost and performance.



Criteria

- Diameter: maximum distance $p_a \leftrightarrow p_b$.
- Connectivity: measure of multiplicity of paths.
- Bisection width: minimum number of links to cut in order to partition the network in 2 equal halves.
- Bisection bandwidth: minimum volume of communication allowed between 2 halves.
- Cost: number of communication links, i.e., wires.

Comparing The Topologies

Table 2.1 A summary of the characteristics of various static network topologies connecting p nodes.

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	$p-1$	$p(p-1)/2$
Star	2	1	1	$p-1$
Complete binary tree	$2\log((p+1)/2)$	1	1	$p-1$
Linear array	$p-1$	1	1	$p-1$
2-D mesh, no wraparound	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2-D wraparound mesh	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	$2p$
Hypercube	$\log p$	$p/2$	$\log p$	$(p\log p)/2$
Wraparound k-ary d-cube	$d\lfloor k/2\rfloor$	$2k^{d-1}$	$2d$	dp