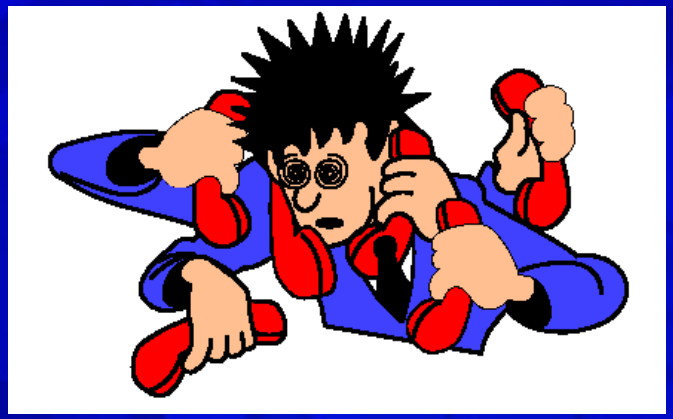
Chapter 6 Multiprocessors and Thread-Level Parallelism



chenwz@zju.edu.cn

2014年4月13日

6.1 Introduction

- **advances in uniprocessor architecture were nearing an end**
- **processor performance growth was at its highest rate**
- **parallel processors will definitely have a bigger role in the future**

Which is right ?

6.2 A Taxonomy of Parallel Architectures

- *Single instruction stream, single data stream* (SISD) —This category is the uniprocessor.
- *Single instruction stream, multiple data streams* (SIMD) —Vector architectures are the largest class of processors of this type.

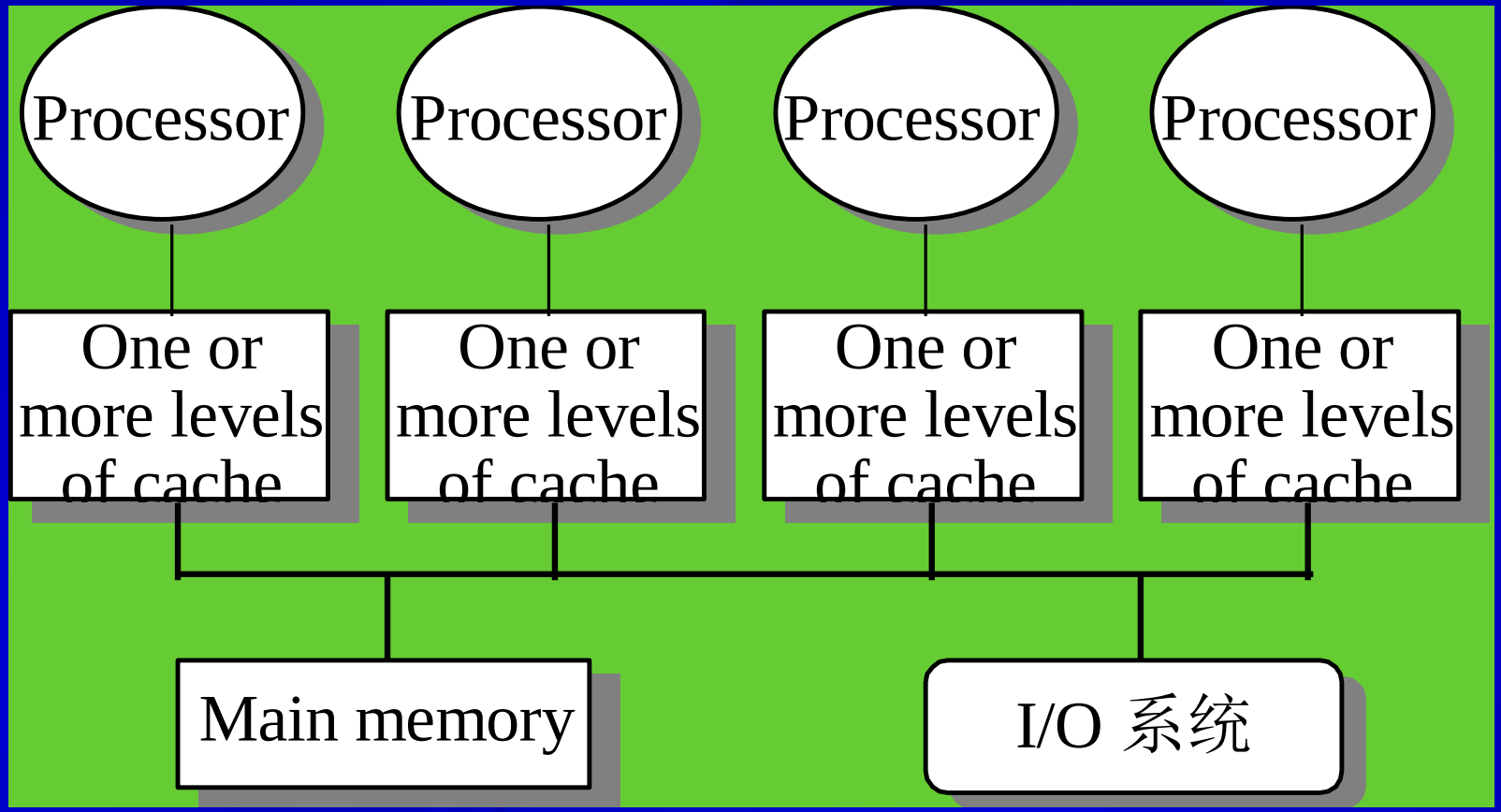
- *Multiple instruction streams, single data stream* (MISD) —No commercial multiprocessor of this type has been built to date, but may be in the future.
- *Multiple instruction streams, multiple data streams* (MIMD) —Each processor fetches its own instructions and operates on its own data. The processors are often off-the-shelf microprocessors.

6.3 Two classes of MIMD

Depend On:

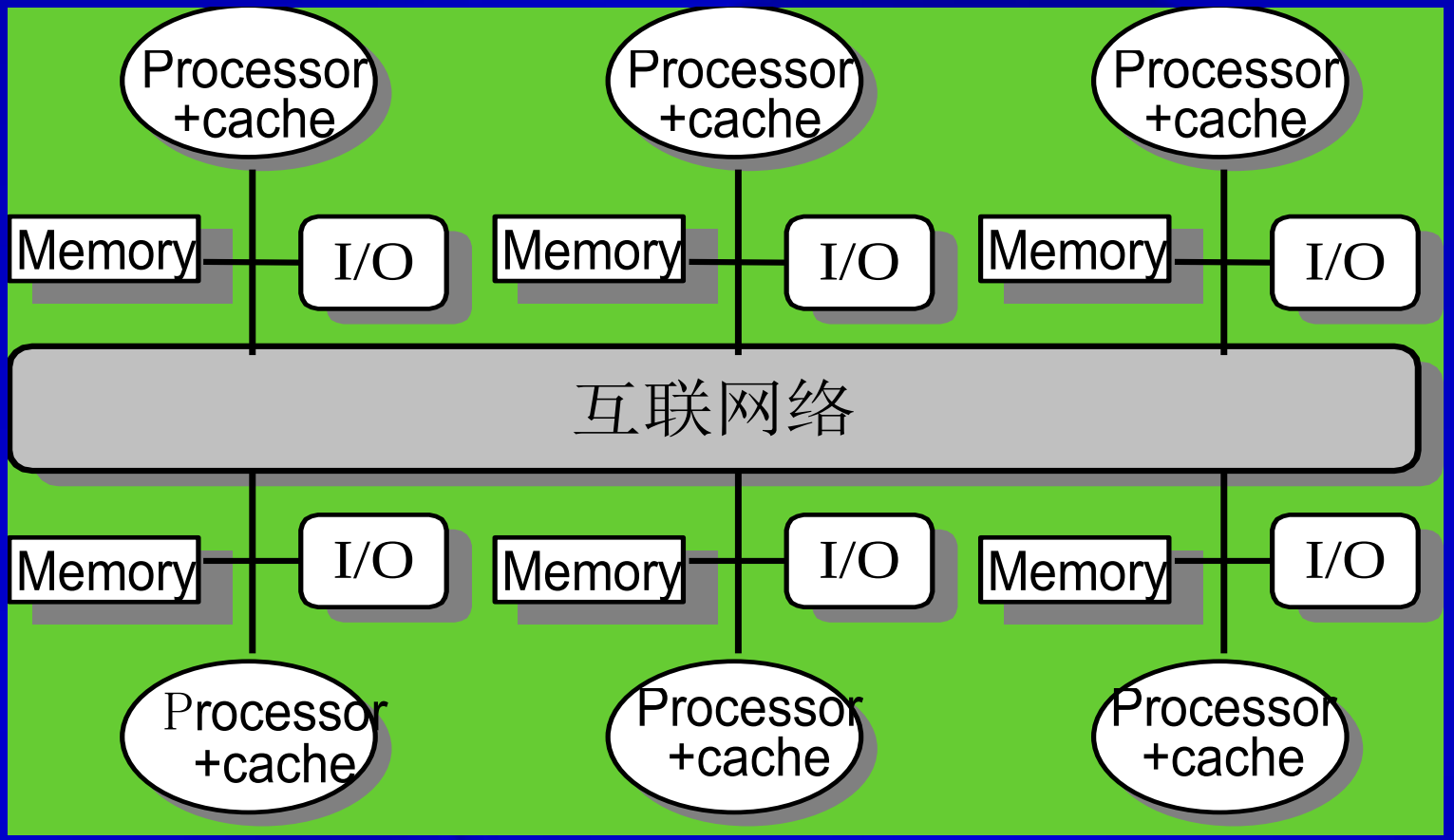
- the number of processors
- memory organization
- Interconnect strategy

6.3.1
centralized shared-memory architecture



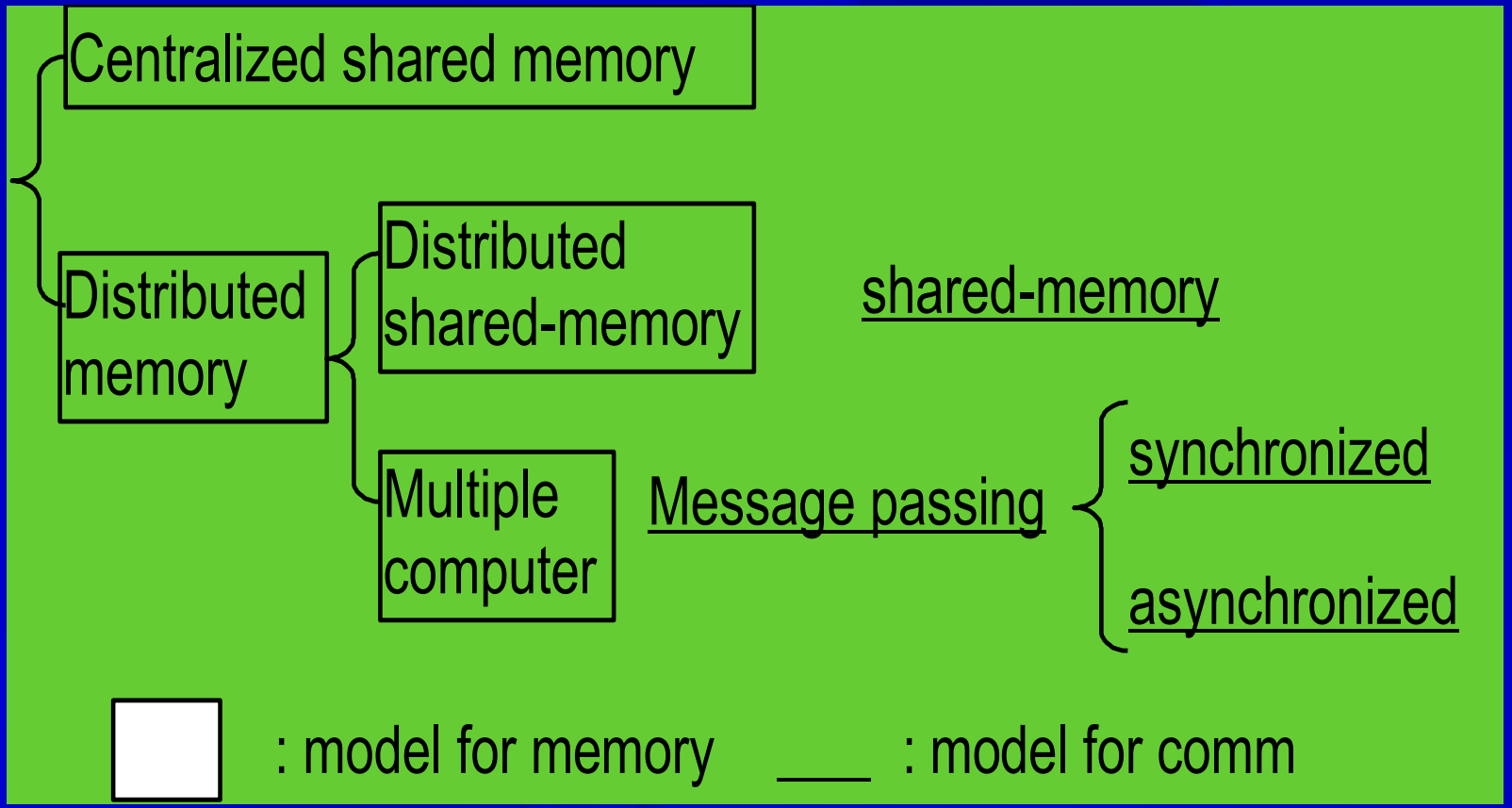
- With small processor counts
- With large caches
- Called *symmetric (shared-memory) multiprocessors* (*SMPs*)
- Called *UMA* for *uniform memory access*

6.3.2 distributed-memory architecture



- With large processor counts
- the need for a high bandwidth interconnect
- Advantage:
  - cost-effective way to scale the memory bandwidth, if most of accesses are to local memory in the node.
  - It reduces the latency for access to the local memory.

6.4 Models for Communication and Memory Architecture



6.4.1 Models for memory Architecture

- **Distributed shared memory**（**DSM or scalable shared memory)**
  - logical uniform address space but physical distributed memory, so any one of the processors can access any one of the memories.
  - Shared memory means sharing the address space, which is different from centralized shared memory.
  - UMA(uniform memory access) ---- centralized share memory.
  - NUMA(non-uniform memory access) ---- distributed shared memory.

- **multiple computers**
  - Address space consists of multiple private address spaces。cannot be addressed by a remote processor.
  - Each processor-memory module is essentially a separate computer (*multicomputers)*.

6.4.2 Models for Communication

- shared memory

- message passing
  - Synchronous message passing (RPC)
  - Asynchronous message passing(MPI)

For Shared memory

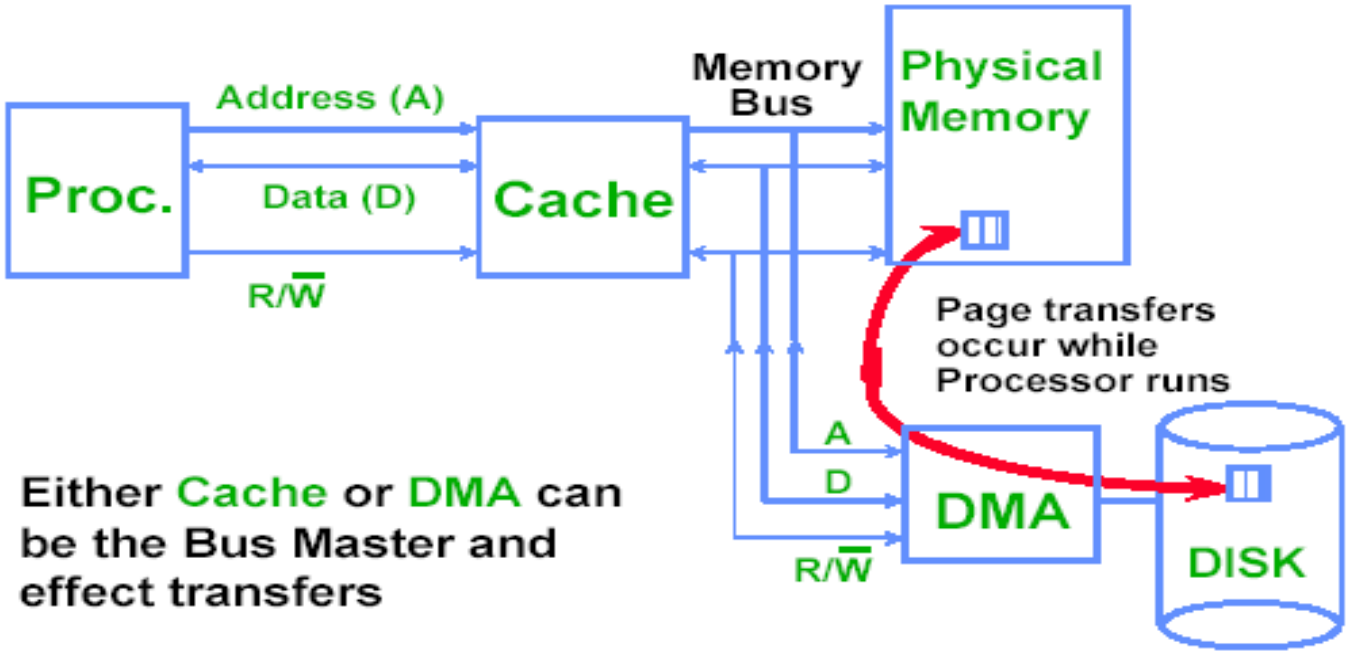
- compatibility with mechanism used in centralized multiprocessors
- easy programming, simplify compiler design
- lower overhead for communication and better use of bandwidth, due to implicit nature of communication and implement memory protection in hardware instead of in OS.
- The ability to use hardware-controlled caching to reduce the frequency of remote communication by supporting automatic caching of both shared and private data.

For Message passing

- The hardware can be simpler
- communication is explicit, simpler to understand
- forcing programmers and compilers to pay attention to communication.
- ….

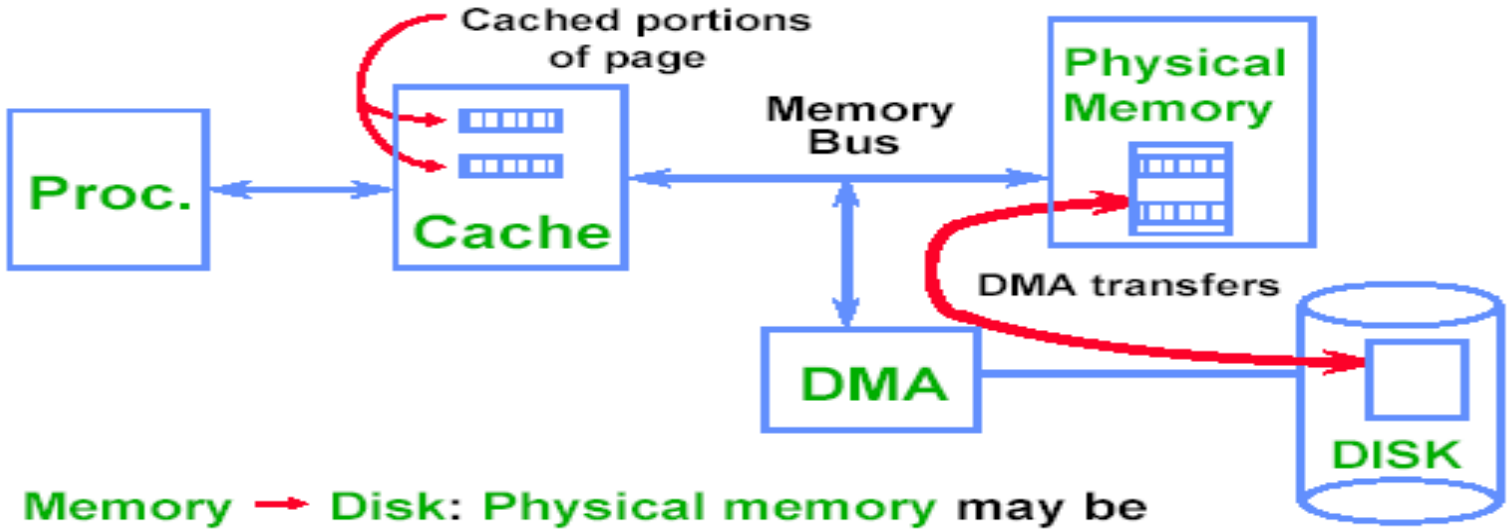
6.5 What Is Multiprocessor Cache Coherence?

Warmup: Parallel I/O



DMA stands for Direct Memory Access

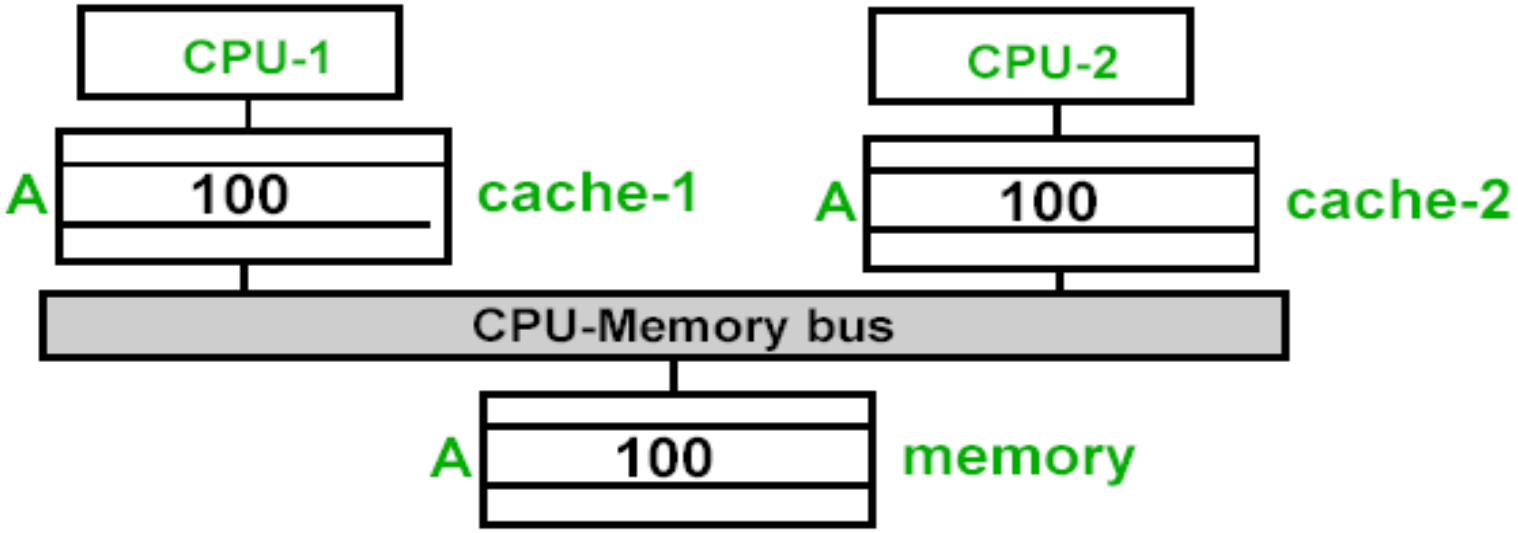
Problems with Parallel I/O



Memory → Disk: Physical memory may be stale if Cache is _____.

Disk → Memory: Cache may have data corresponding to _____.

Memory Consistency in SMPs



Suppose CPU-1 updates A to 200.

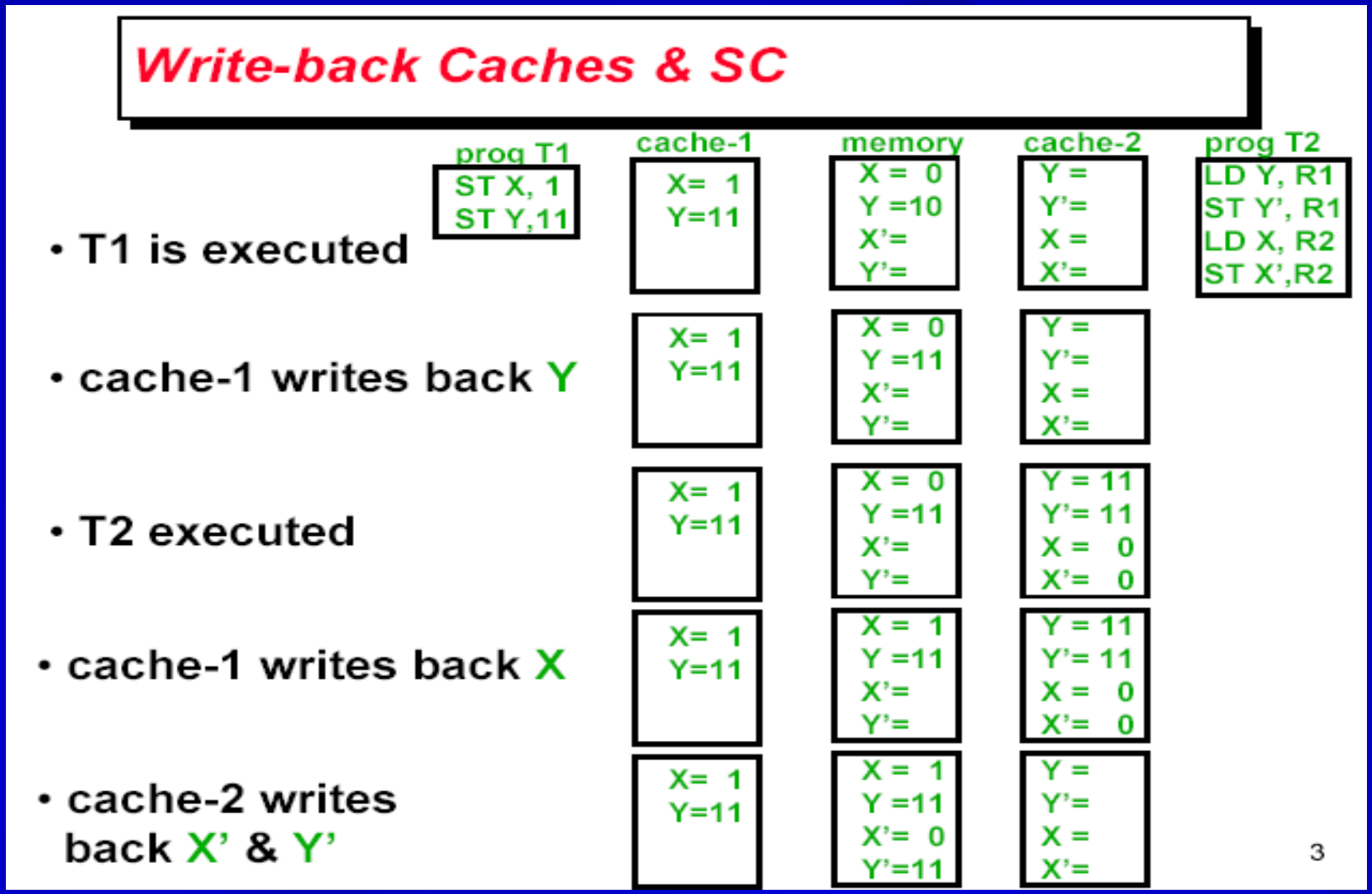
write-back: memory and cache-2 have stale values

write-through: cache-2 has a stale value

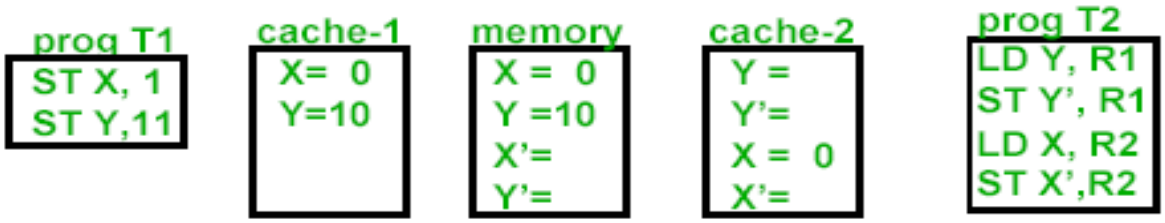
Example1: read

Time	Event	Cache contents for CPU A	Cache contents for CPU A	Memory contents for Location X
0				1
1	CPU A reads X	1		1
2	CPU B reads X	1	1	1
3	CPU A stores 0 into X	0	1	0

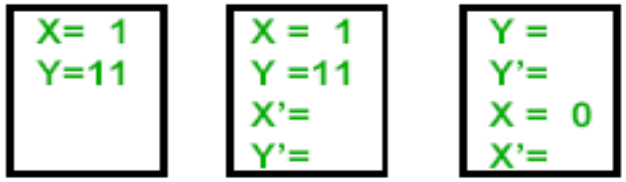
Example2: write



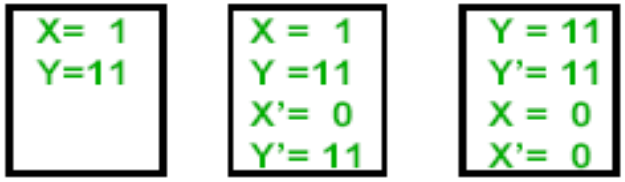
Write-through Caches & SC



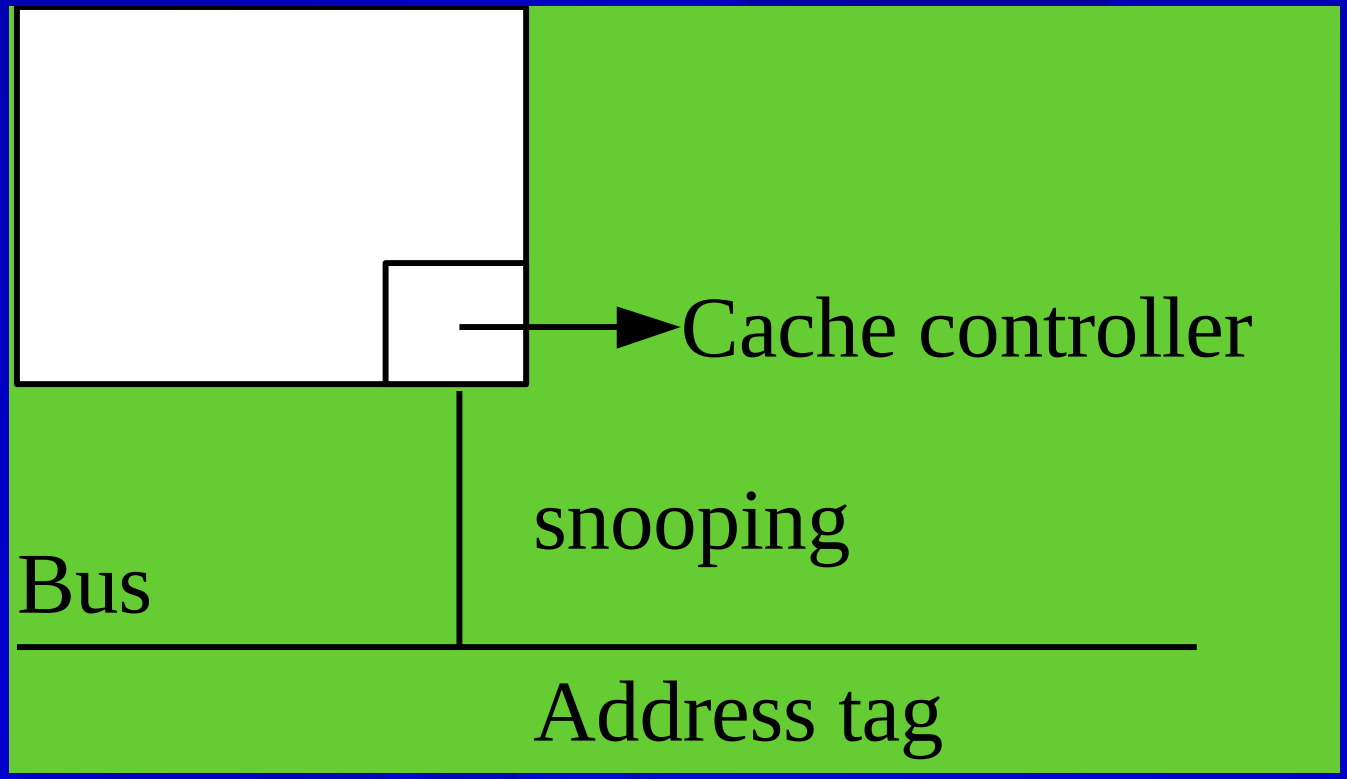
- T1 executed



- T2 executed



6.6 Snooping

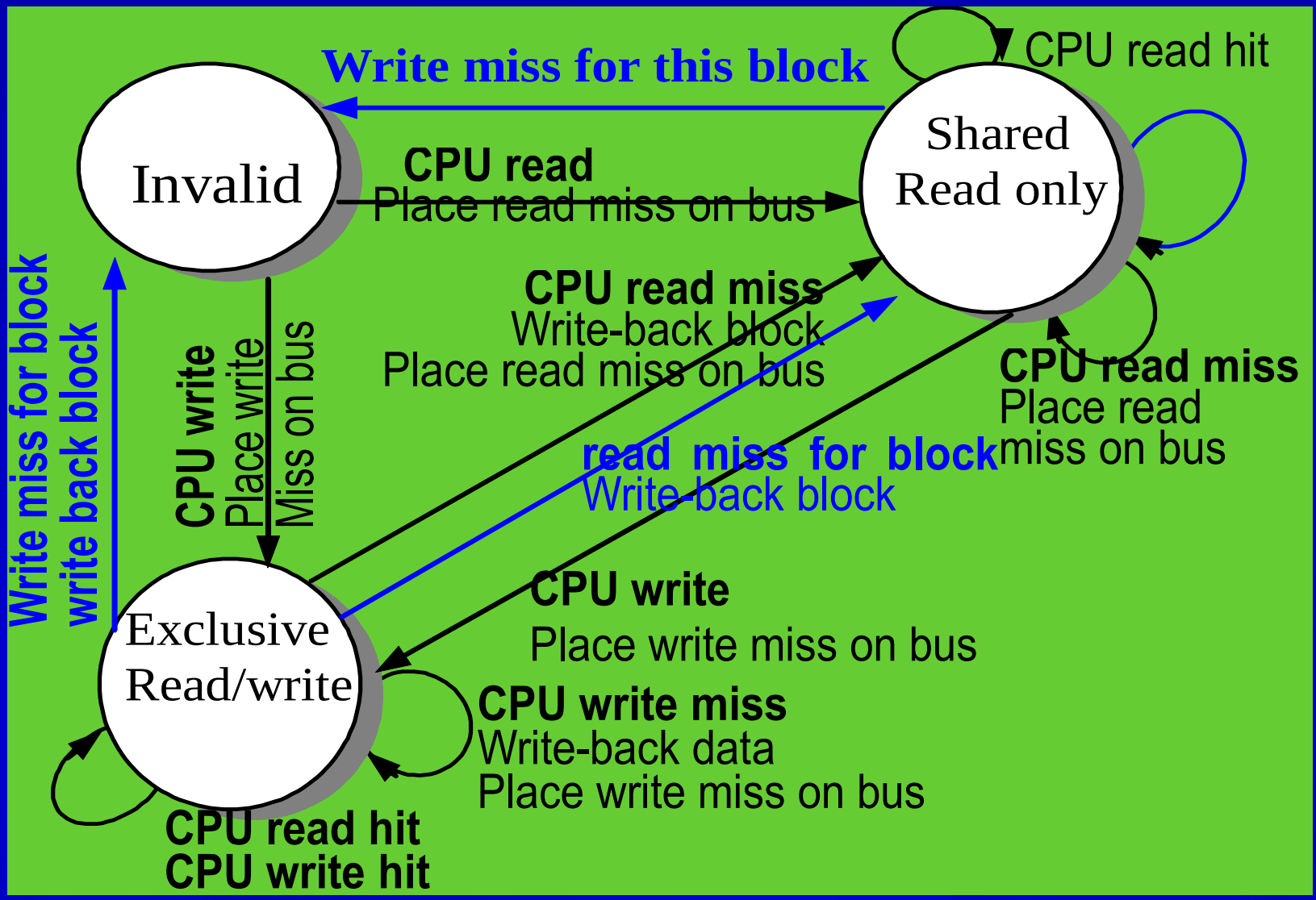


6.6.1 Write invalidate protocol

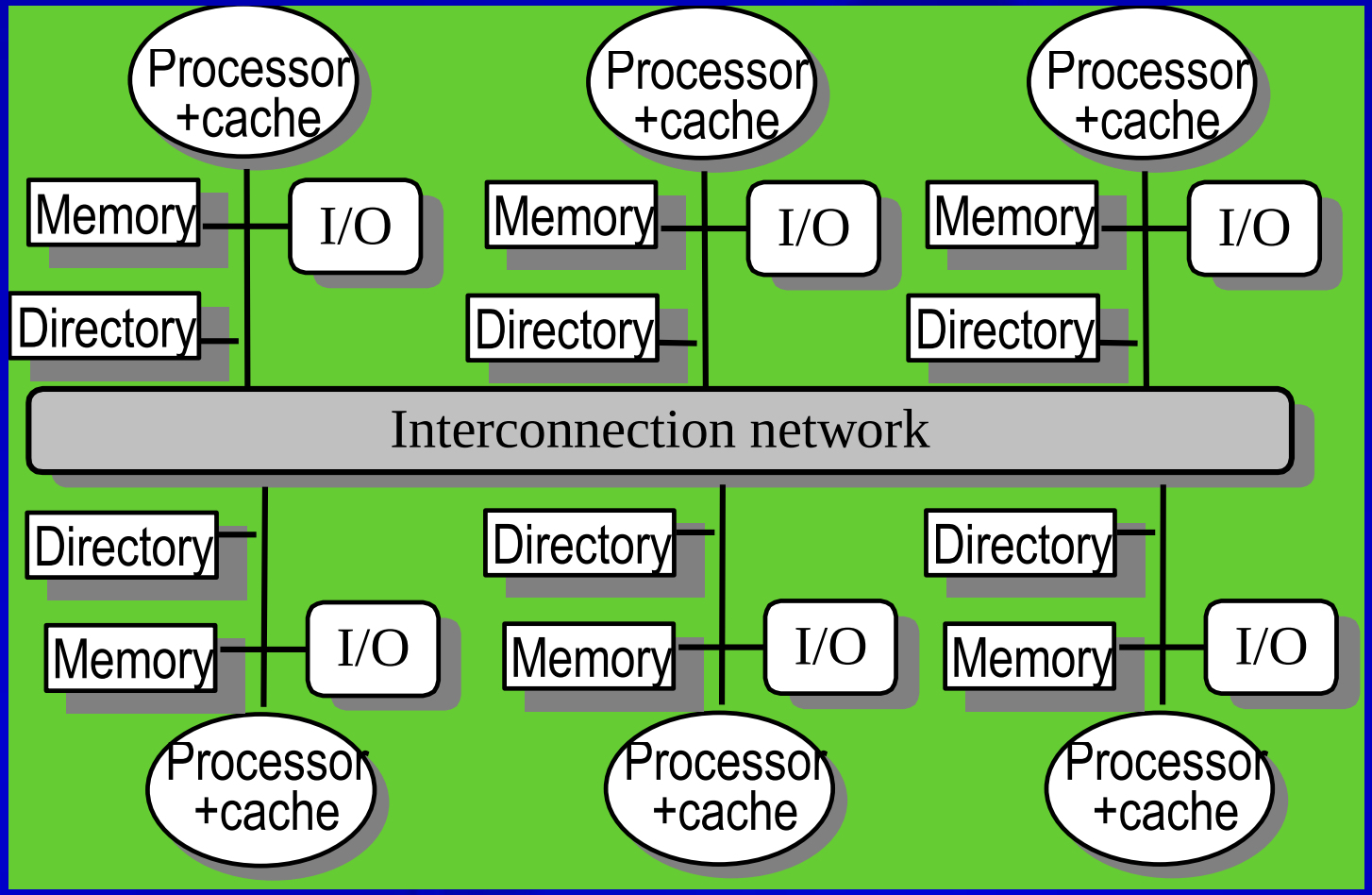
Processor Activity	Bus activity	Contents of CPU A's cache	Contents of CPU B's cache	Contents of Memory Location X
				0
CPU A Reads X	Cache miss for X	0		0
CPU B Reads X	Cache miss for X	0	0	0
CPU A writes A 1 to X	Invalidation for X	1		0
CPU B Reads X	Cache miss for X	1	1	1

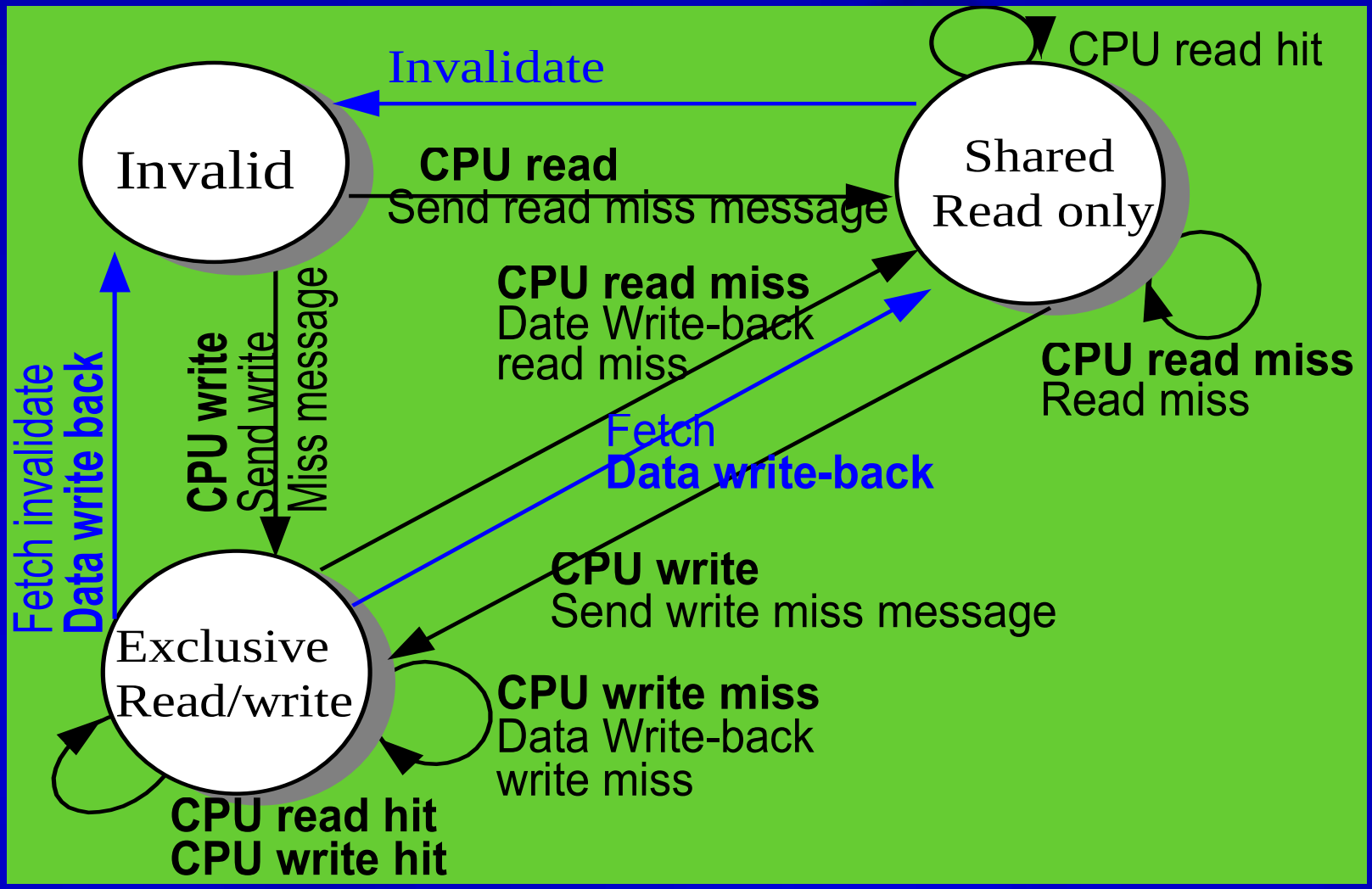
6.6.2Write update or write broadcast protocol

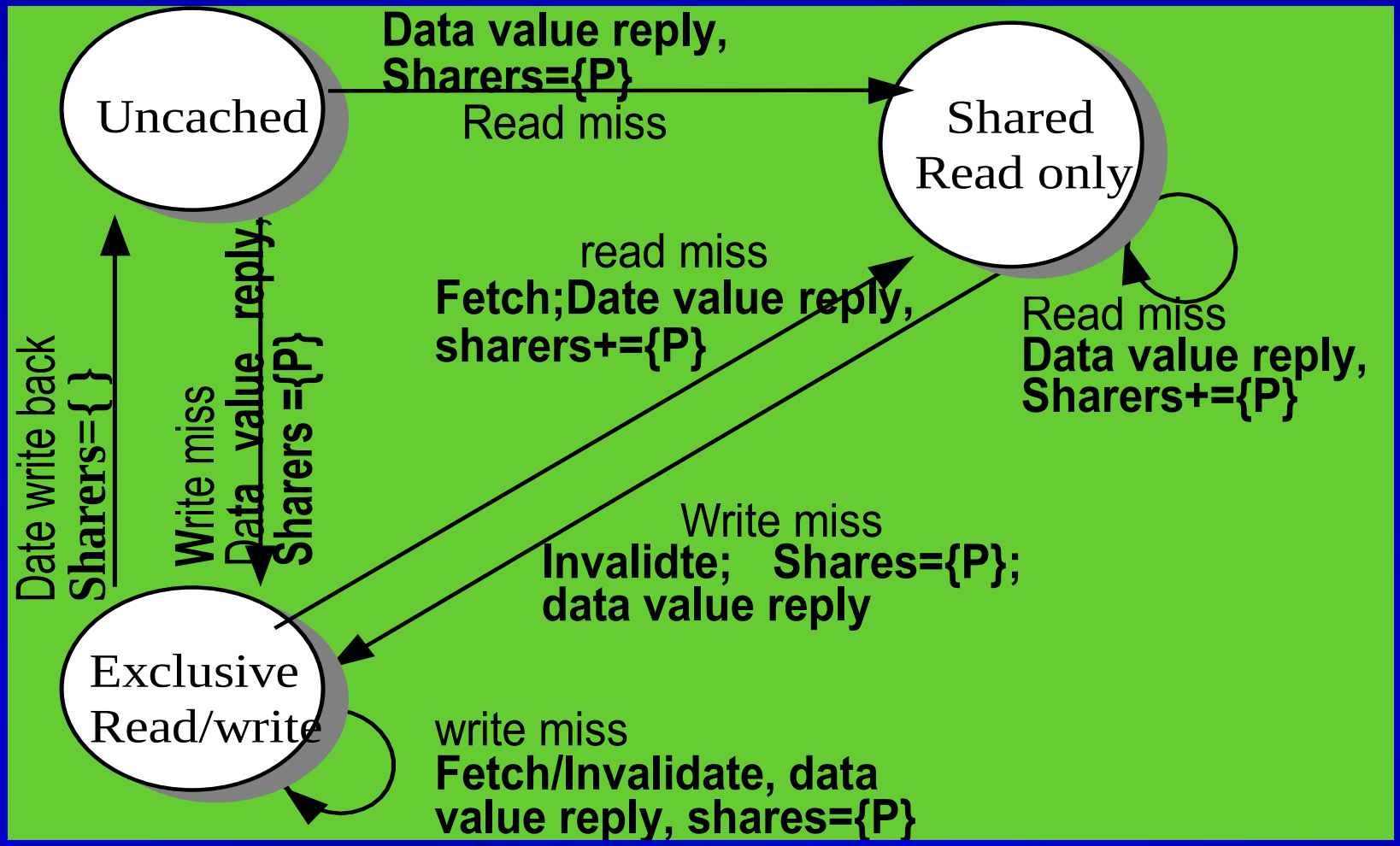
Processor Activity	Bus activity	Contents of CPU A's cache	Contents of CPU B's cache	Contents of Memory Location X
				0
CPU A Reads X	Cache miss for X	0		0
CPU B Reads X	Cache miss for X	0	0	0
CPU A writes A 1 to X	Write broadcast Of X	1	1	1
CPU B Reads X		1	1	1


Write miss for this block
CPU read hit
Invalid
Shared
Read only
CPU read
Place read miss on bus
CPU read miss
Write-back block
Place read miss on bus
CPU read miss
Place read
miss on bus
read miss for block
Write-back block
Write miss for block
write back block
CPU write
Place write
Miss on bus
CPU write
Place write miss on bus
Exclusive
Read/write
CPU write miss
Write-back data
Place write miss on bus
CPU read hit
CPU write hit

6.7 Directory protocol




Invalidate
CPU read hit
Invalid
CPU read
Send read miss message
Shared
Read only
CPU read miss
Date Write-back
read miss
CPU read miss
Read miss
Fetch
Data write-back
CPU write
Send write
Miss message
Fetch invalidate
Data write back
CPU write
Send write miss message
Exclusive
Read/write
CPU write miss
Data Write-back
write miss
CPU read hit
CPU write hit


Uncached
Data value reply,
Sharers={P}
Read miss
Shared
Read only
read miss
Fetch;Date value reply,
sharers+={P}
Read miss
Data value reply,
Sharers+={P}
Date write back
Sharers={ }
Write miss
Data value reply,
Sharers ={P}
Write miss
Invalidte; Shares={P};
data value reply
Exclusive
Read/write
write miss
Fetch/Invalidate, data
value reply, shares={P}