Multiprocessors

EECE6029

Yizong Cheng

2/26/2016

Multiple Processor Systems

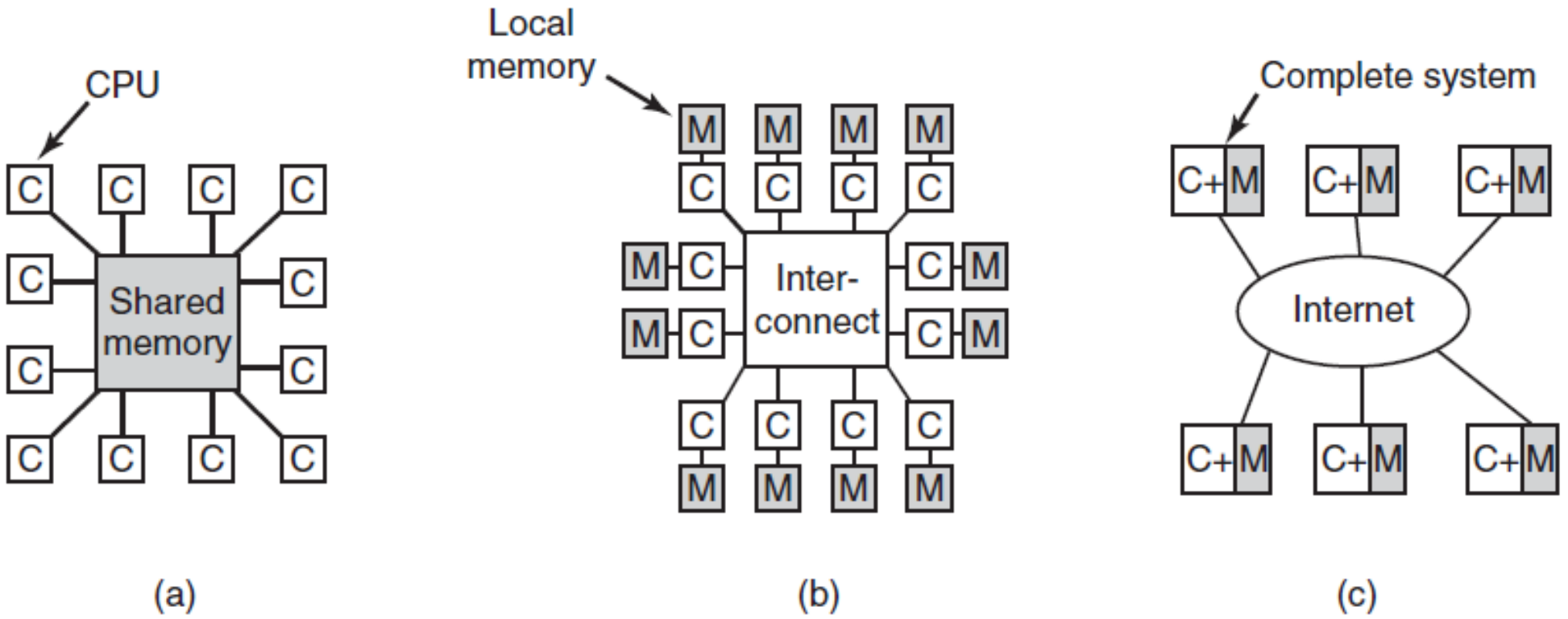

Figure 8-1. (a) A shared-memory multiprocessor. (b) A message-passing multicomputer. (c) A wide area distributed system.

Uniform Memory Access Multiprocessors

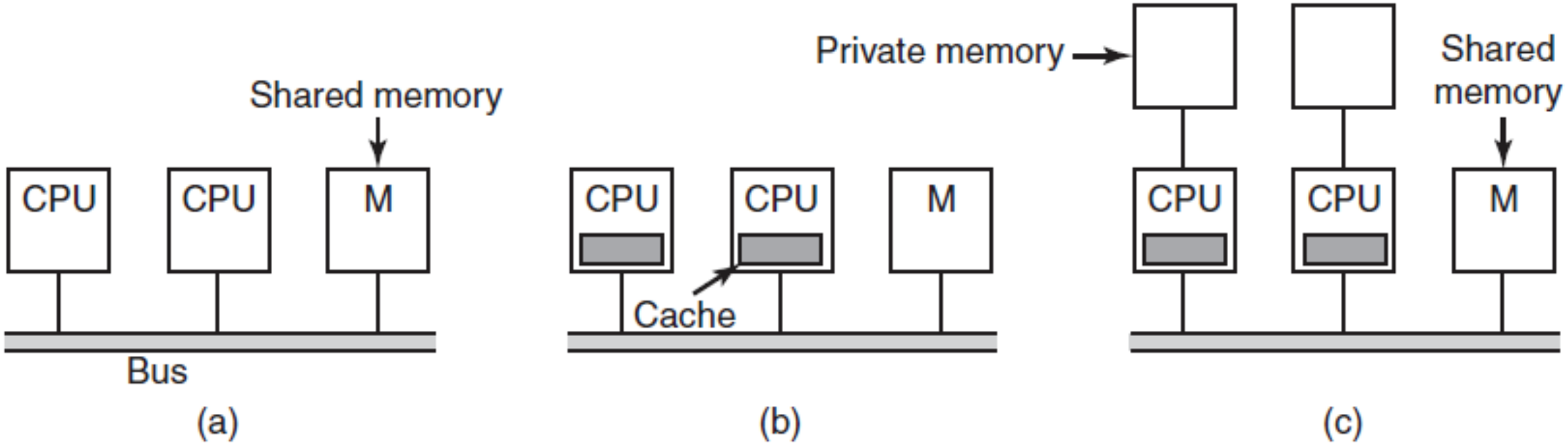

Figure 8-2. Three bus-based multiprocessors. (a) Without caching. (b) With caching. (c) With caching and private memories.

Interconnect Topologies

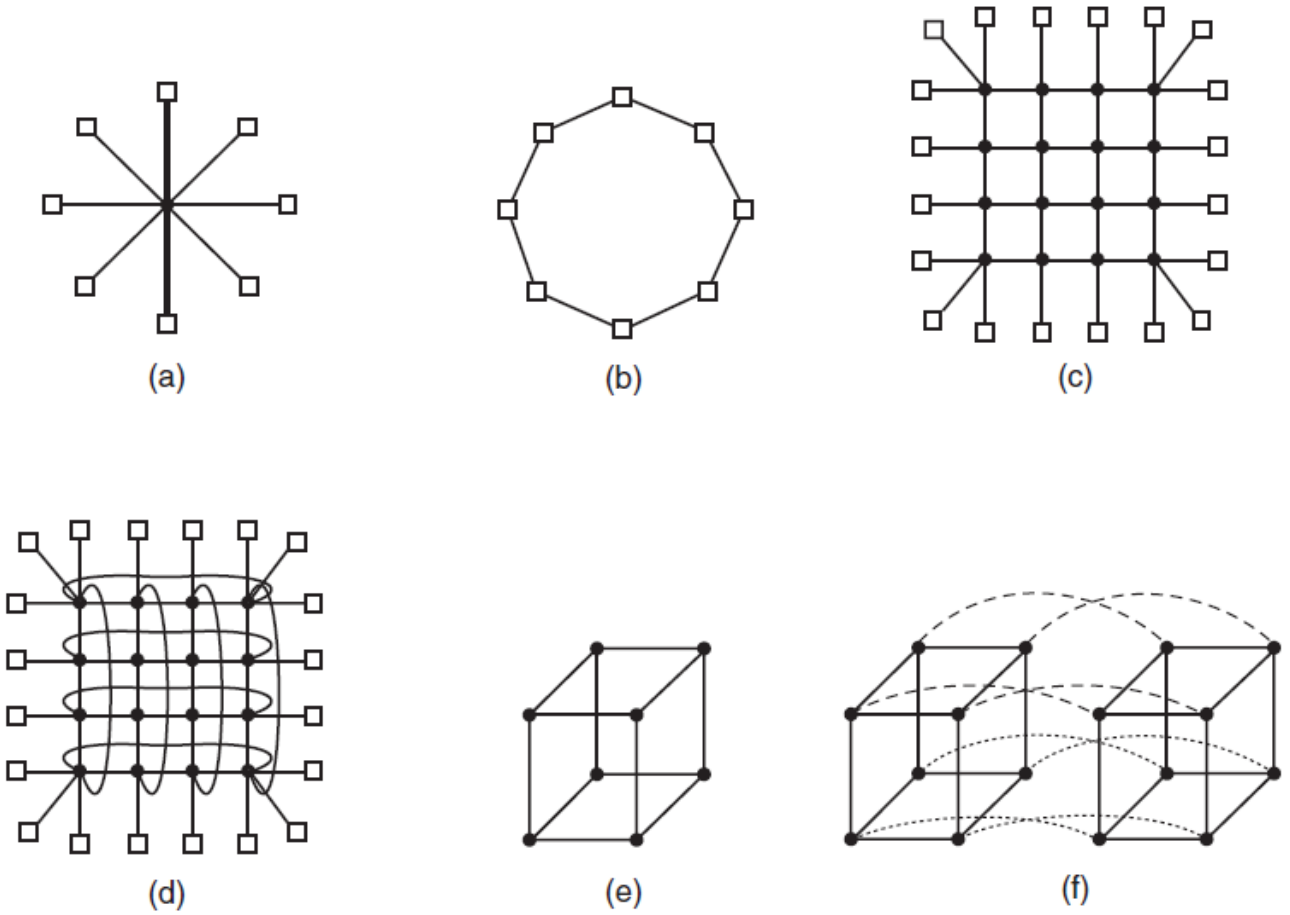

Figure 8-16. Various interconnect topologies. (a) A single switch. (b) A ring. (c) A grid. (d) A double torus. (e) A cube. (f) A 4D hypercube.

Crossbar Switch

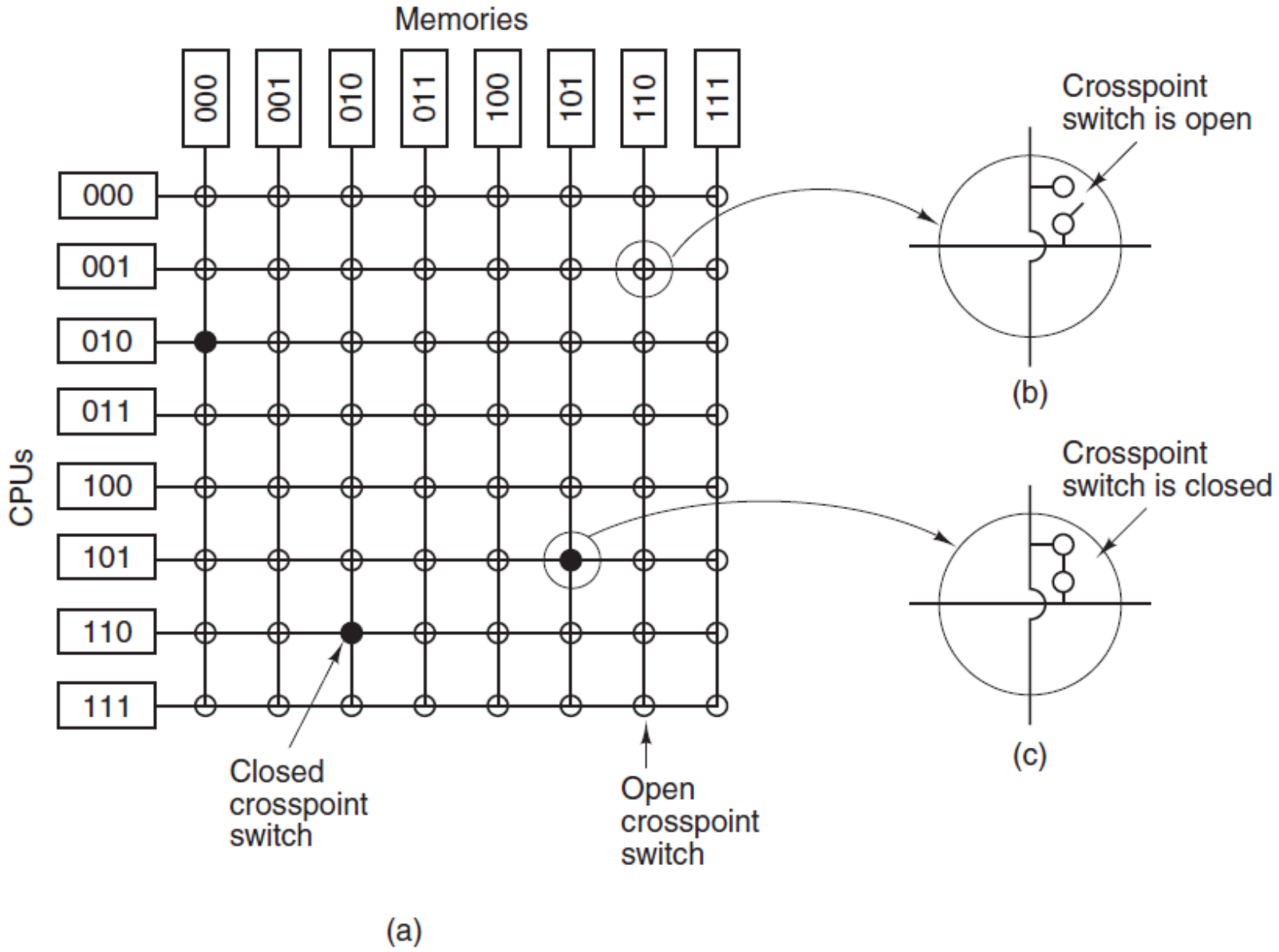

Figure 8-3. (a) An 8 × 8 crossbar switch. (b) An open crosspoint. (c) A closed crosspoint.

A 2 x 2 Switch

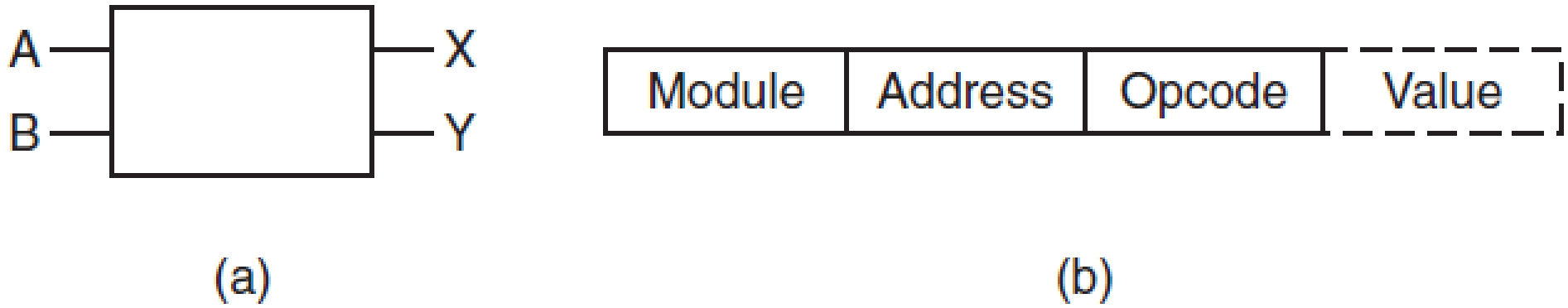

Figure 8-4. (a) A 2×2 switch with two input lines, A and B, and two output lines, X and Y. (b) A message format.

Multistage Switching Network

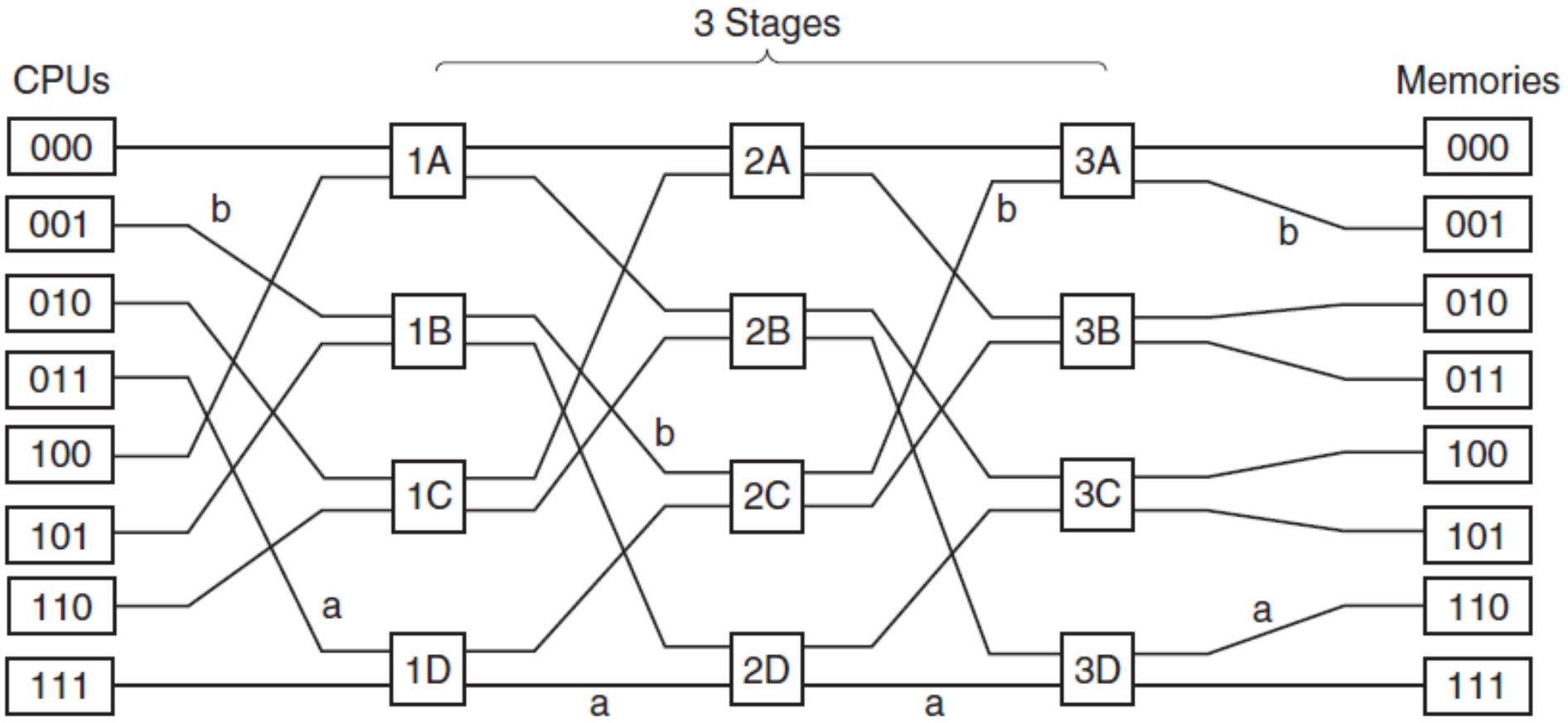

Figure 8-5. An omega switching network.

Nonuniform Memory Access (NUMA)

- There is a single address space visible to all CPUs.
- Access to remote memory is via LOAD and STORE instructions.
- Access to remote memory is slower than access to local memory.

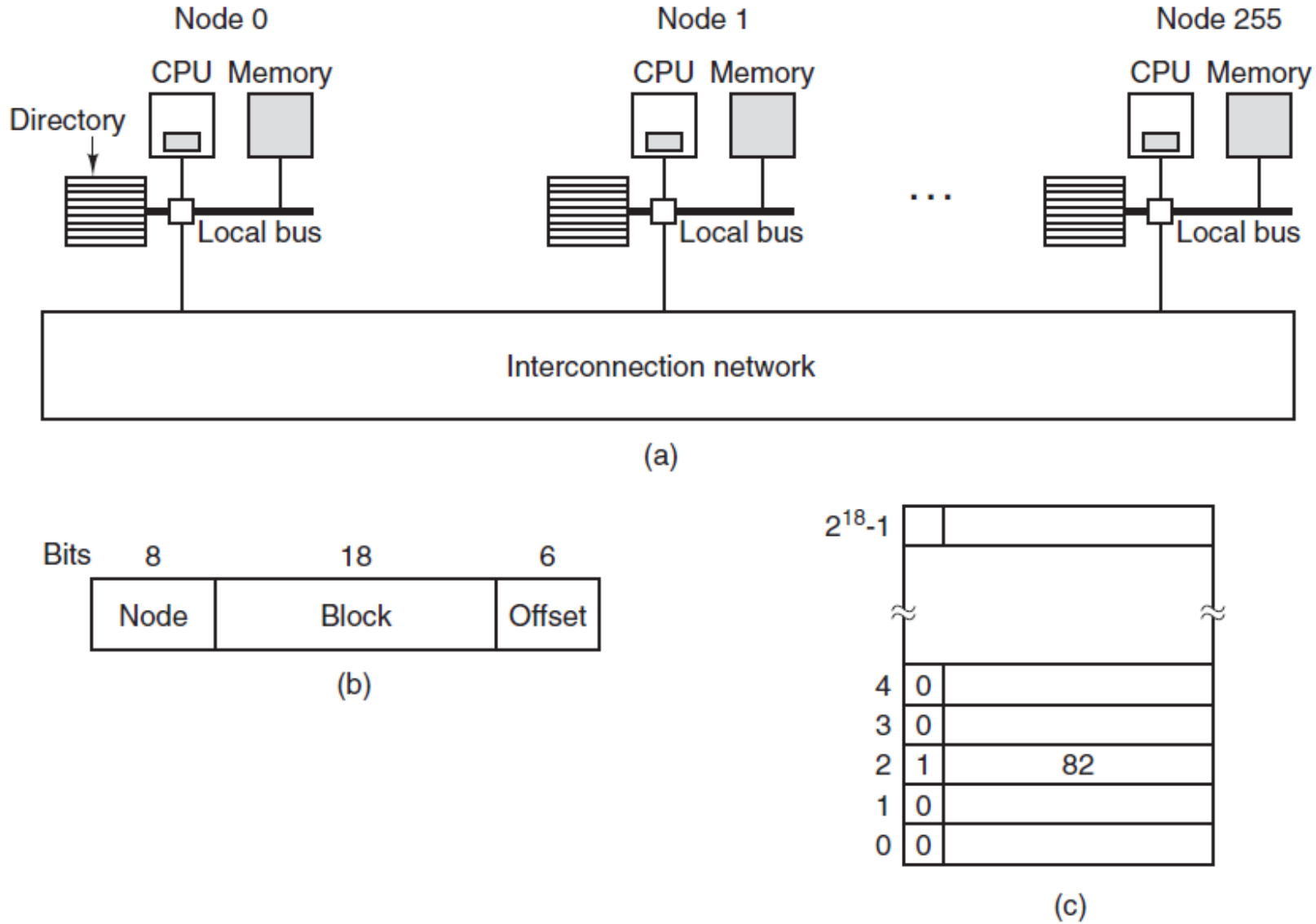

Figure 8-6. (a) A 256-node directory-based multiprocessor. (b) Division of a 32-bit memory address into fields. (c) The directory at node 36.

Each CPU Has Its Own Operating System

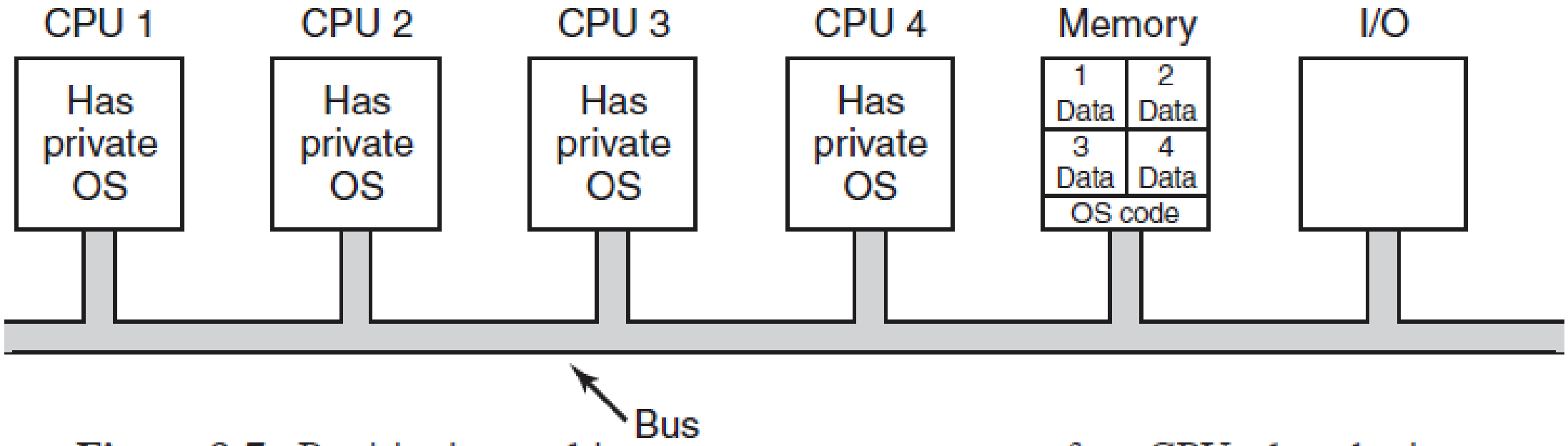

Figure 8-7. Partitioning multiprocessor memory among four CPUs, but sharing a single copy of the operating system code. The boxes marked Data are the operating system's private data for each CPU.

Master-Slave Multiprocessor

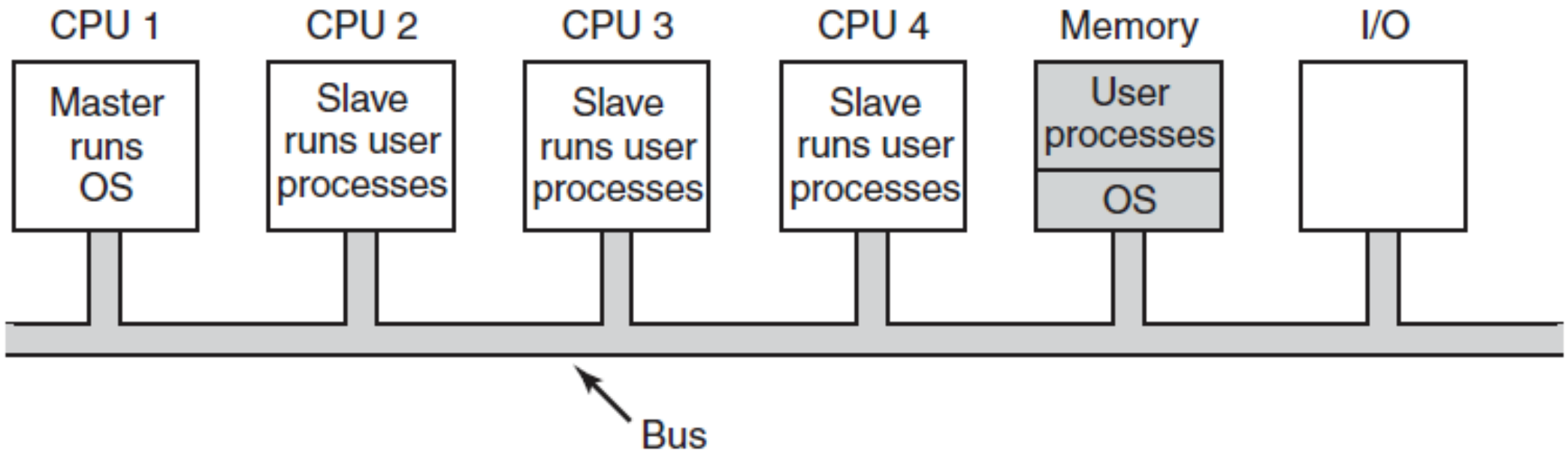

Figure 8-8. A master-slave multiprocessor model.

Symmetric Multiprocessor

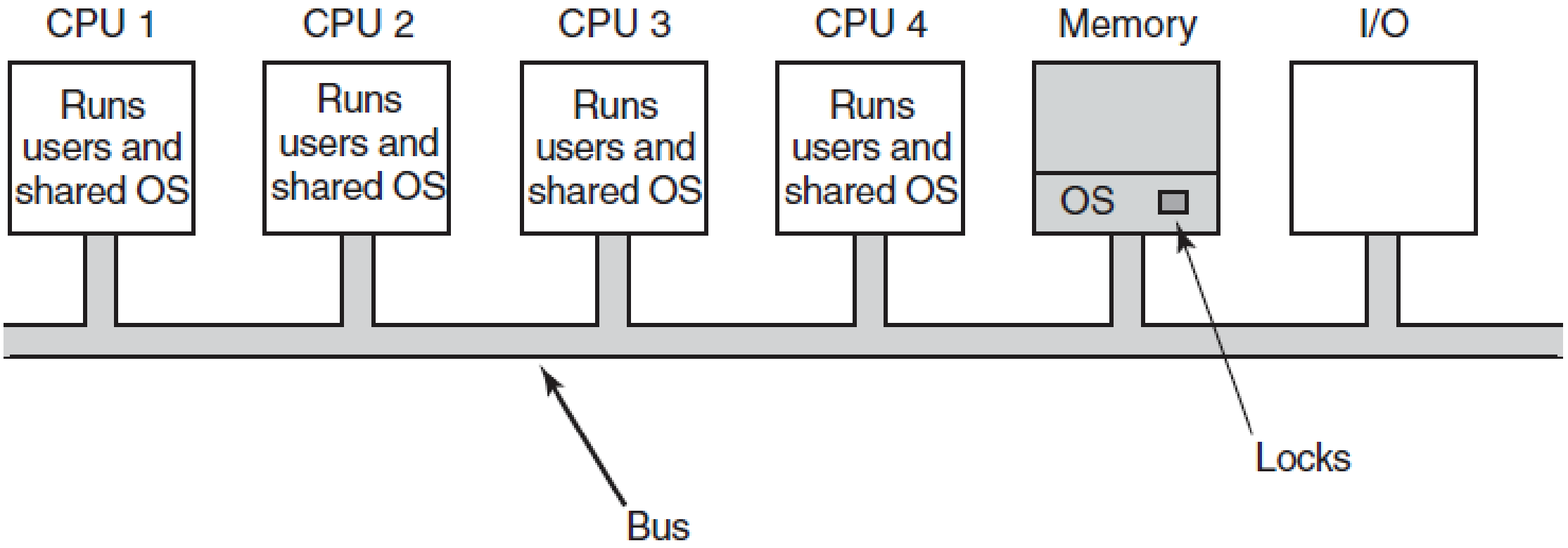

Figure 8-9. The SMP multiprocessor model.

Multiprocessor Synchronization

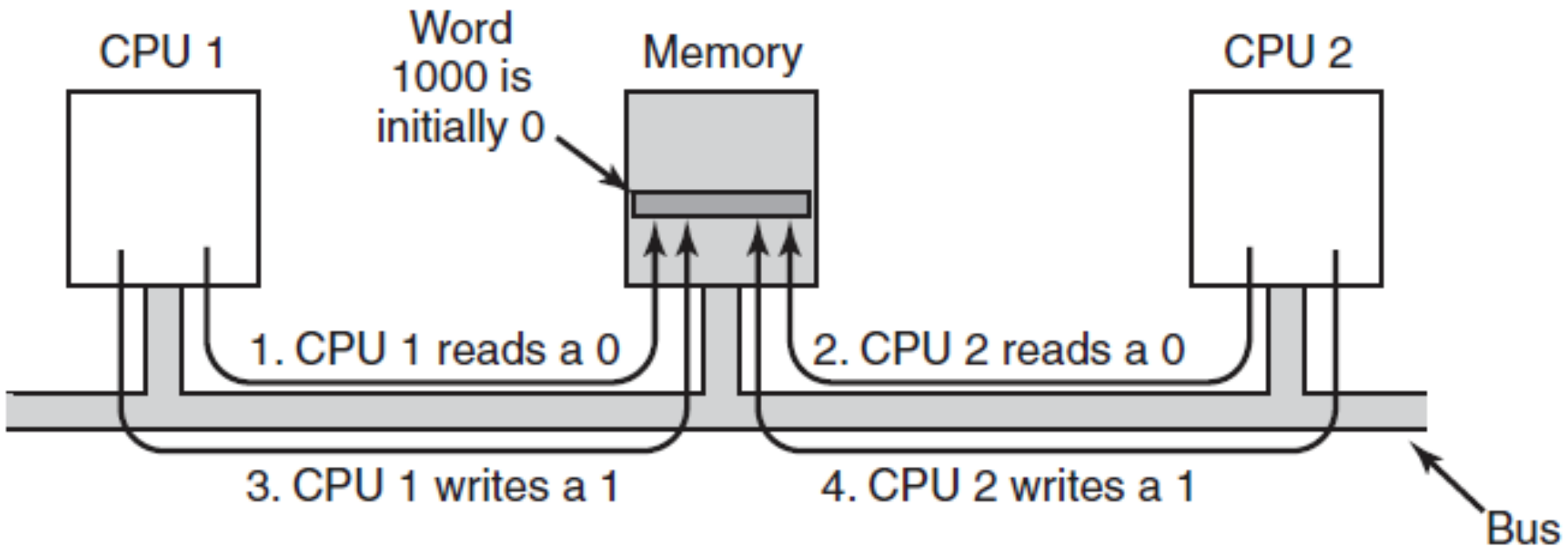

Figure 8-10. The TSL instruction can fail if the bus cannot be locked. These four steps show a sequence of events where the failure is demonstrated.

Cache Thrashing Avoidance

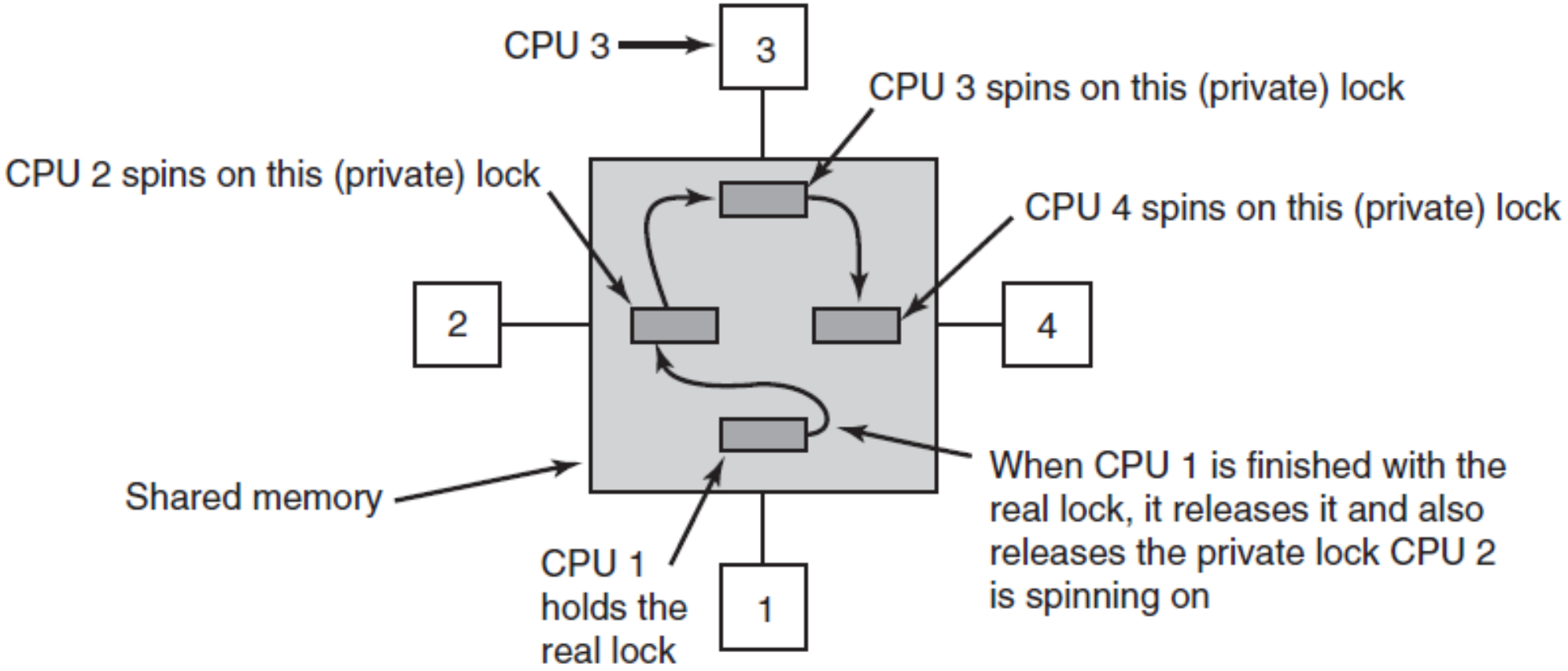

Figure 8-11. Use of multiple locks to avoid cache thrashing.

Time Sharing

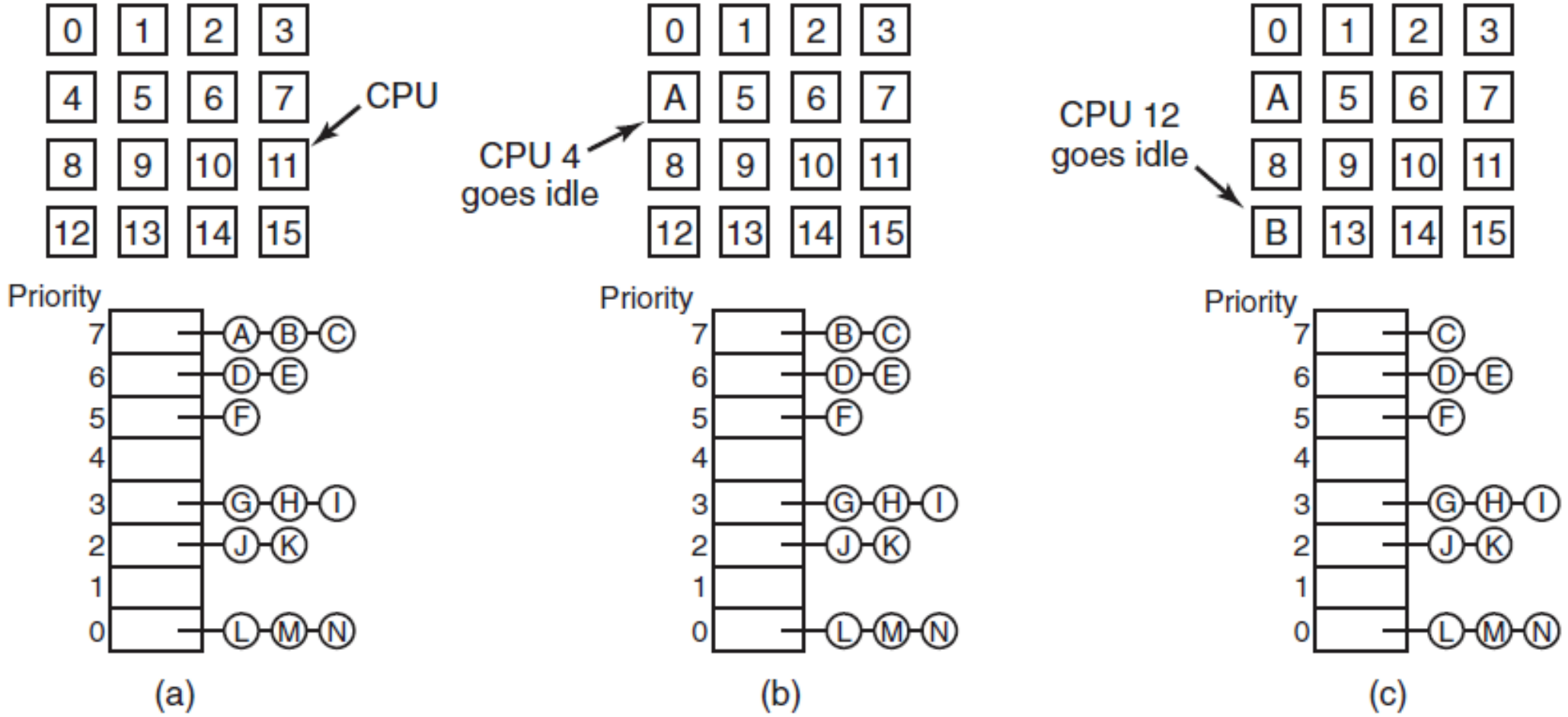

Figure 8-12. Using a single data structure for scheduling a multiprocessor.

Space Sharing

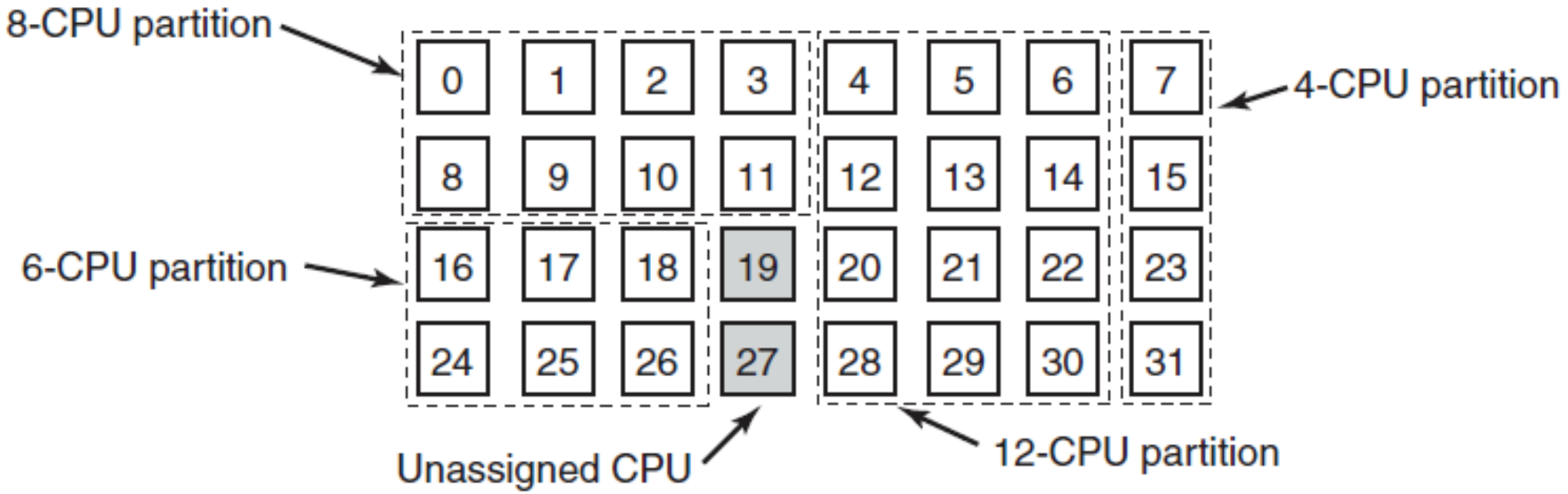

Figure 8-13. A set of 32 CPUs split into four partitions, with two CPUs available.

Threads Running Out of Phase

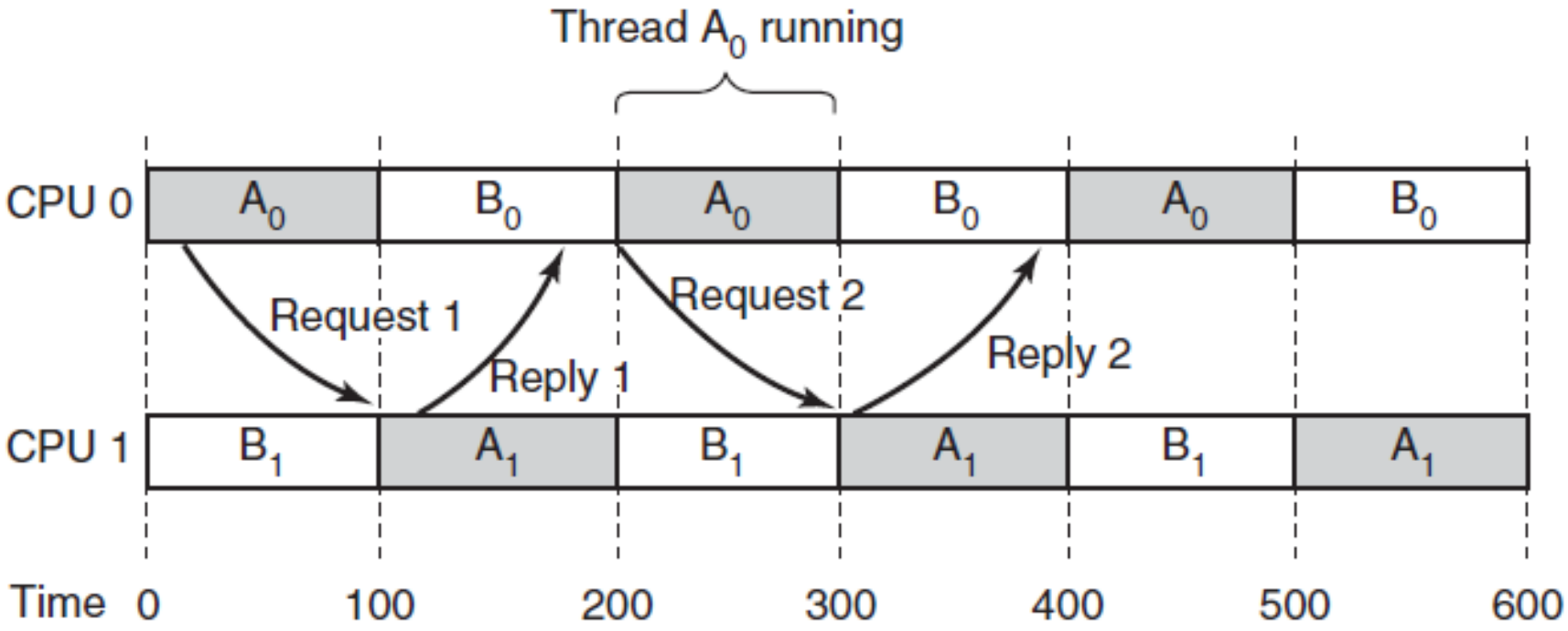

Figure 8-14. Communication between two threads belonging to thread *A* that are running out of phase.

Gang Scheduling

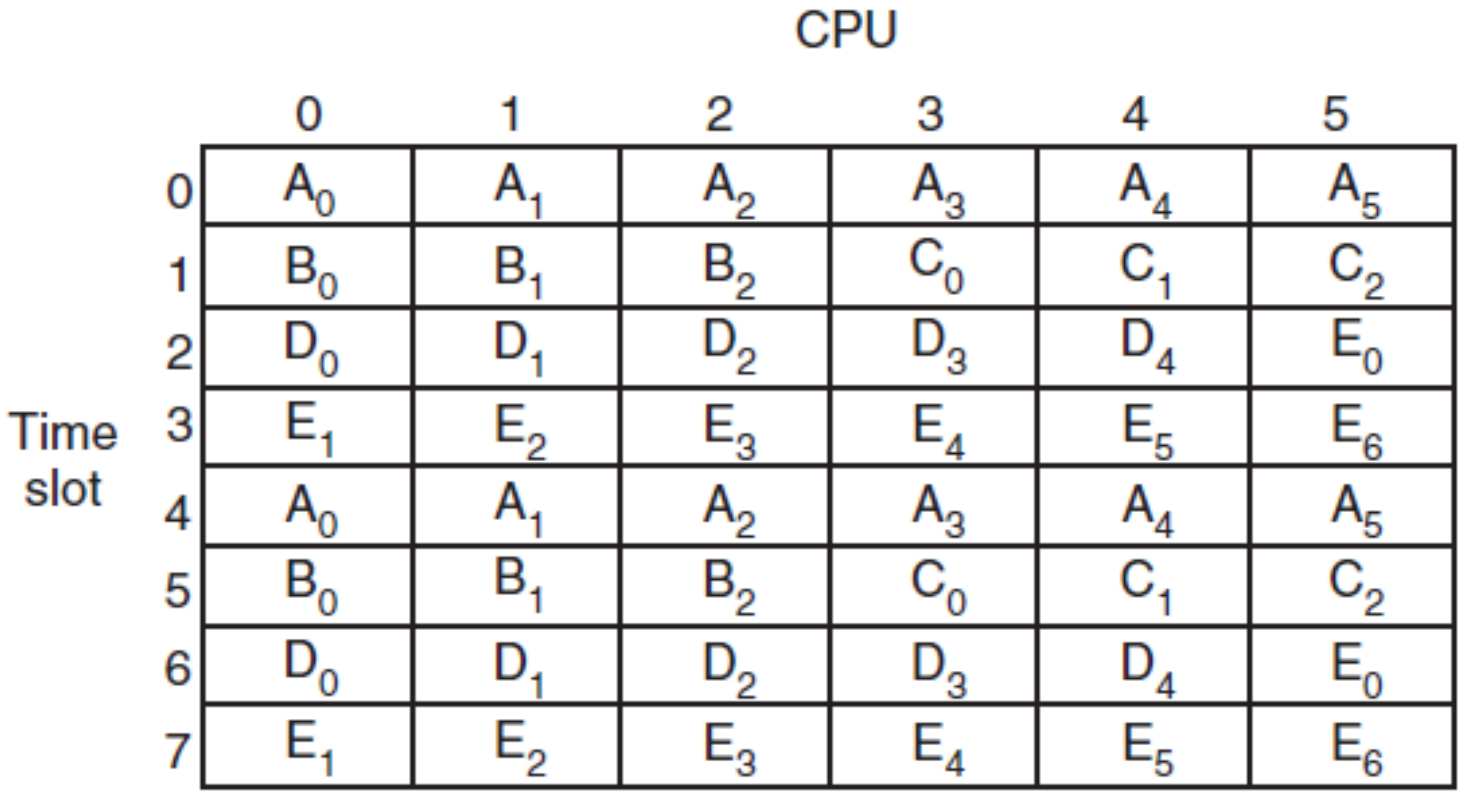

Figure 8-15. Gang scheduling.