JGI JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

Master of Computer Applications

23MCAC105 – Advanced Computer Architecture

Credits: 3

L: T: P – 3-0-0

Prepared by

Dr. A. Rengarajan, Professor

JGI JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

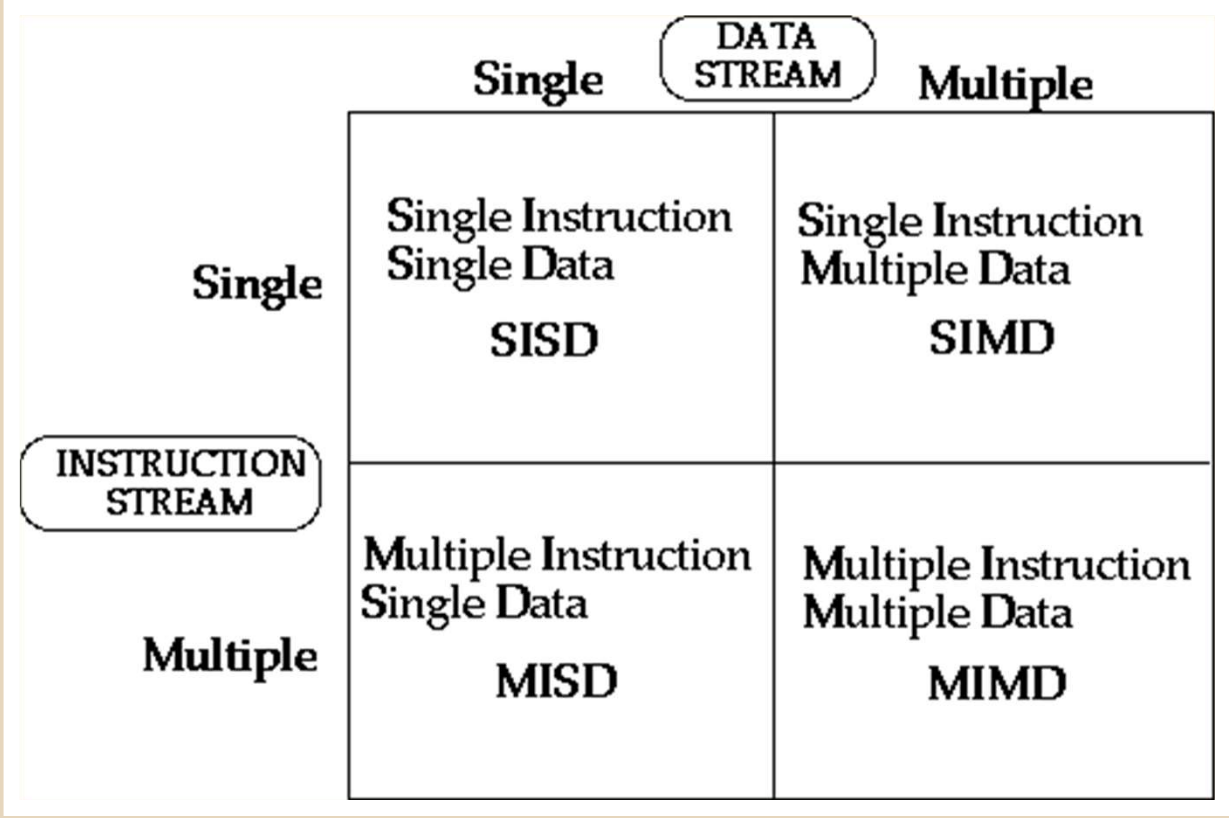
Module – 5

MULTIPROCESSOR ARCHITECTURE

- Taxonomy of Parallel Architectures
- Centralized Shared Memory Architecture
- Synchronization
- Memory Consistency
- Symmetric and Distributed Shared Memory Architectures
- SISD, MISD, MIMD, Single Instruction Multiple Data Stream (SIMD) Architectures.

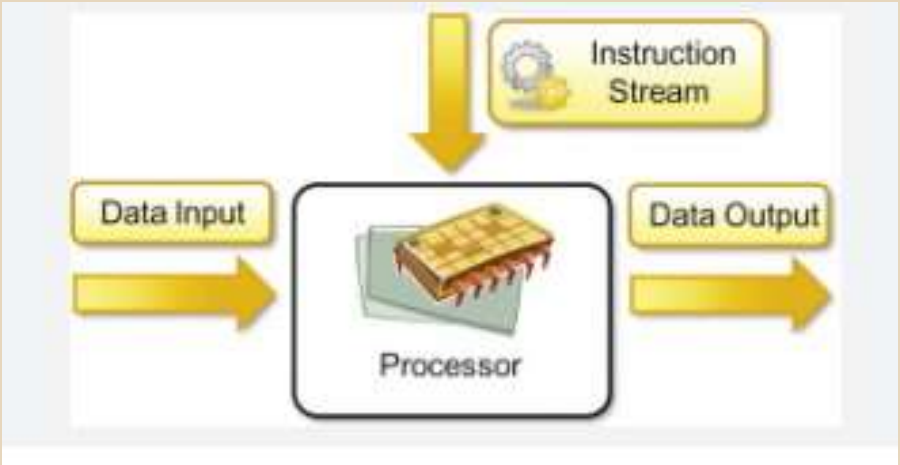
Taxonomy of parallel architectures

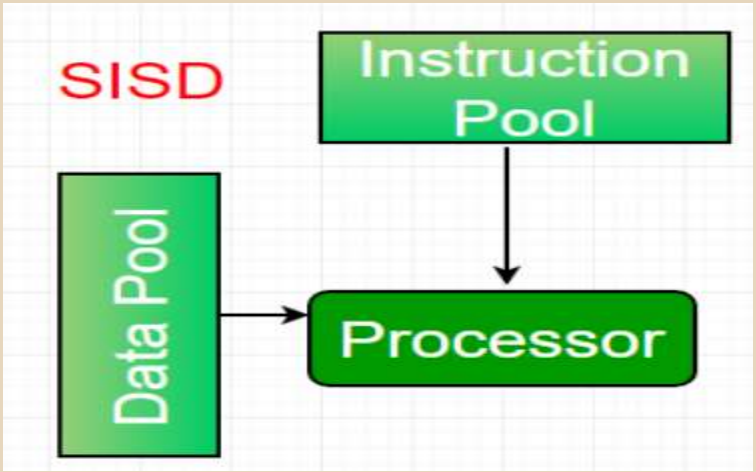
- ✓ 4 categories of Flynn's classification of multiprocessor systems by their instruction and data streams



SISD

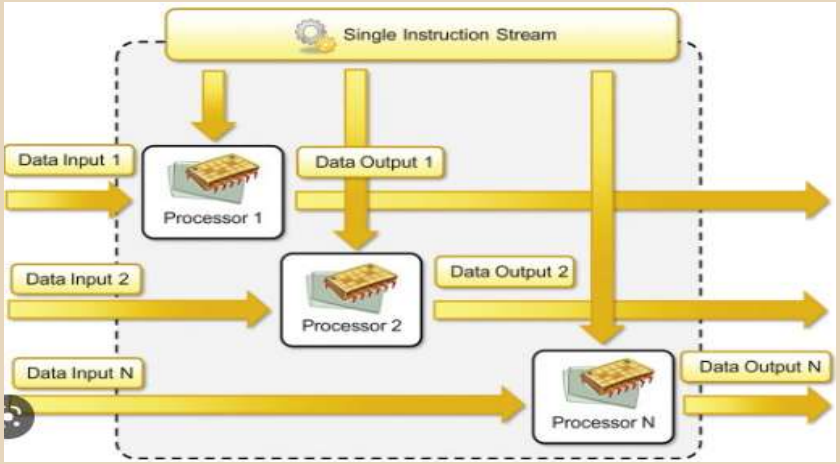
- ✓ SISD machines executes a single instruction on individual data values using a single processor.
- ✓ Based on traditional Von Neumann uniprocessor architecture, instructions are executed sequentially or serially, one step after the next.
- ✓ Until most recently, most computers are of SISD type.

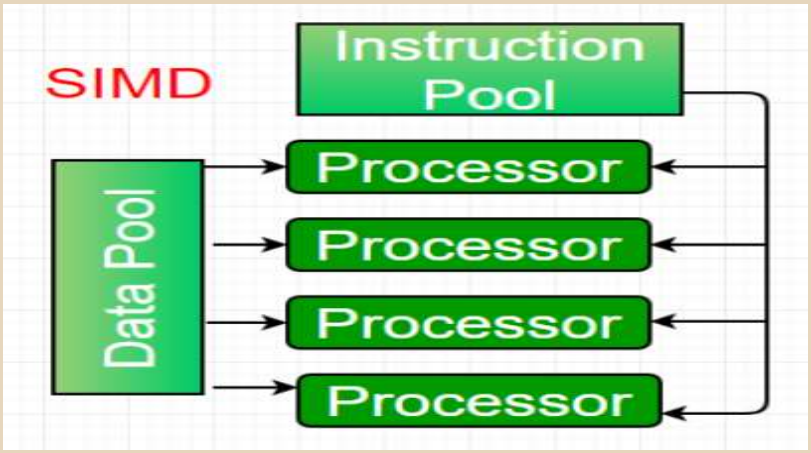




SIMD

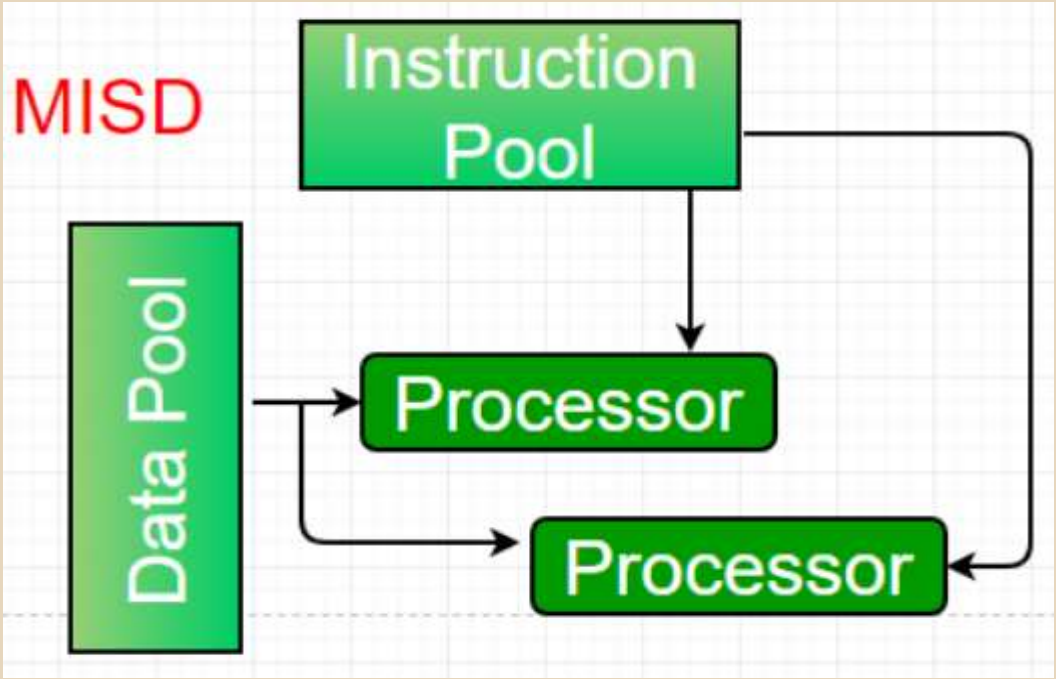
- ✓ An SIMD machine executes a single instruction on multiple data values simultaneously using many processors.
- ✓ Since there is only one instruction, each processor does not have to fetch and decode each instruction. Instead, a single control unit does the fetch and decoding for all processors.
- ✓ SIMD architectures include array processors.





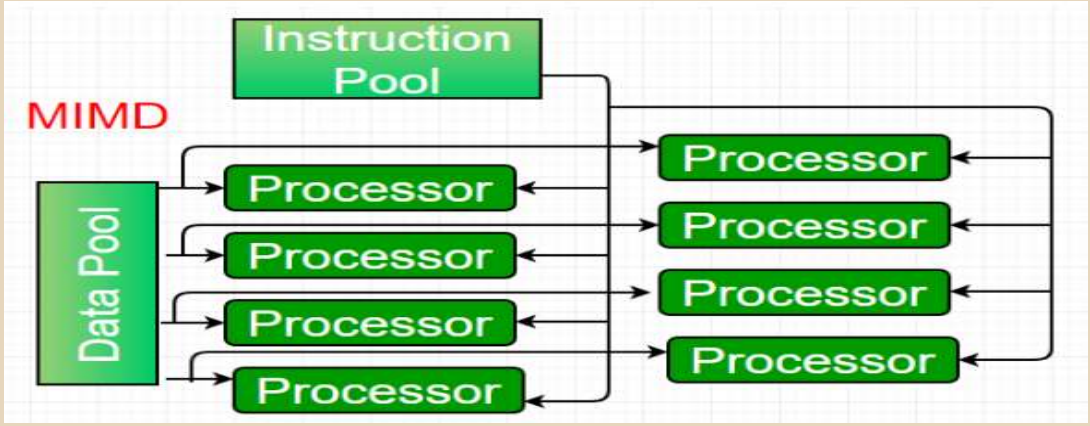
MISD

- ✓This category does not actually exist. This category was included in the taxonomy for the sake of completeness.

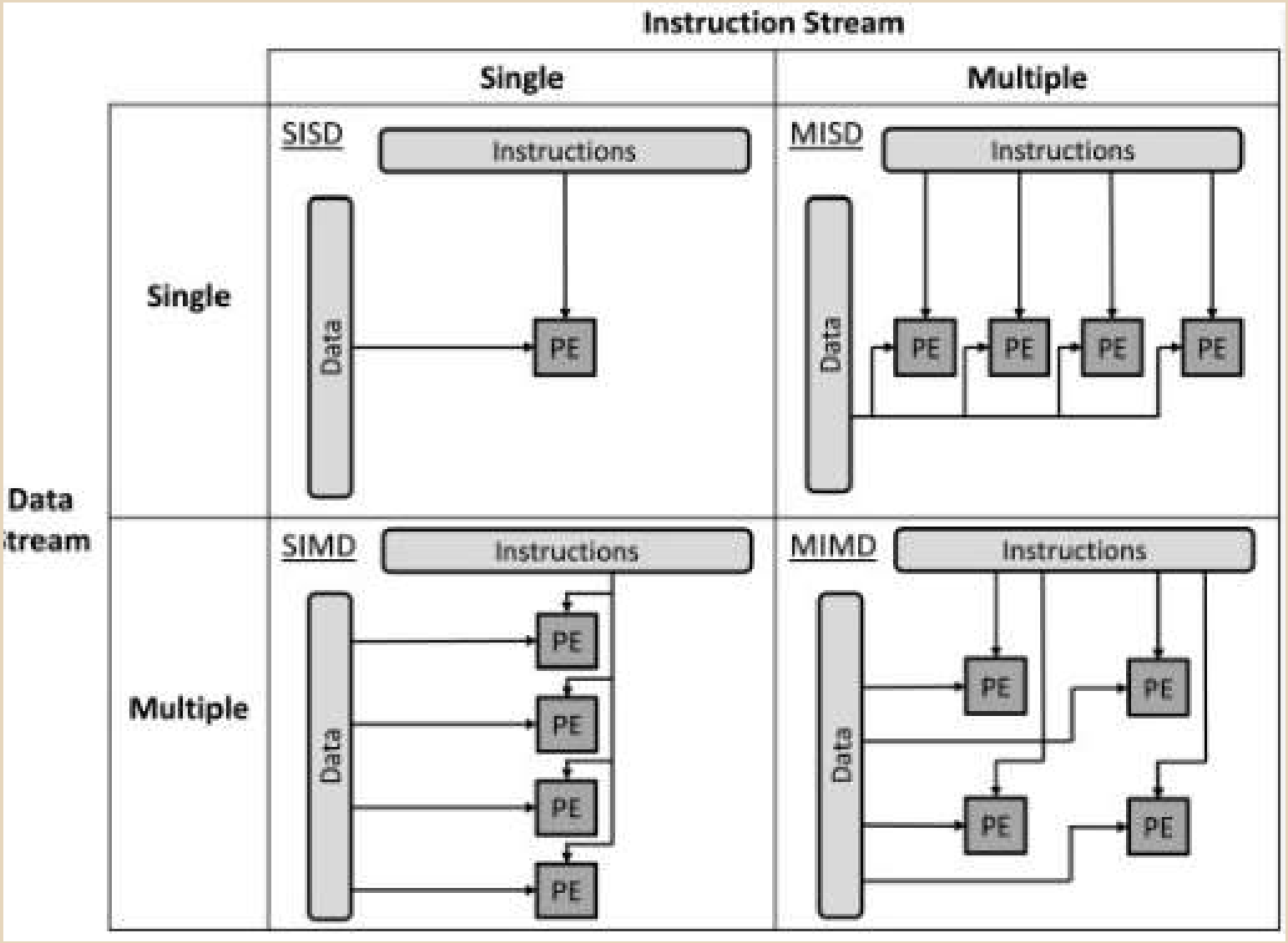


MIMD

- ✓ MIMD machines are usually referred to as multiprocessors or multi computers.
- ✓ It may execute multiple instructions simultaneously, contrary to SIMD machines.
- ✓ Each processor must include its own control unit that will assign to the processors parts of a task or a separate task.
- ✓ It has two subclasses: Shared memory and distributed memory



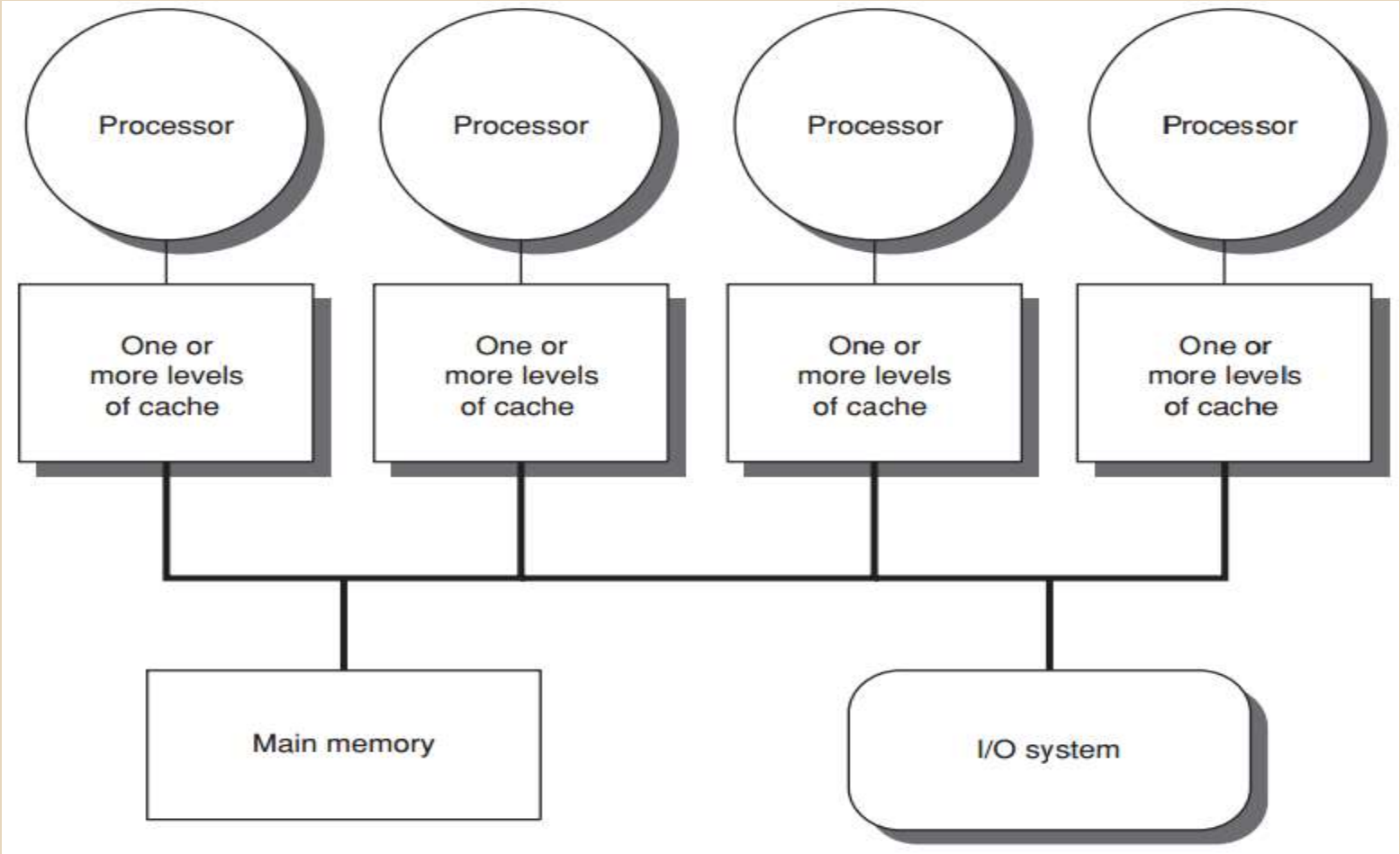
Summary – Flynn's Classification



Centralized Shared Memory

- ✓ Eight or fewer cores
- ✓ Share a single centralized memory
- ✓ All processors have equal access to
- ✓ All processors have uniform latency from memory
- ✓ Uniform memory access (UMA) multiprocessors

Centralized Shared Memory



Synchronization

- ✓ In a centralized system: all processes reside on the same system utilize the same clock.
- ✓ In a distributed system: like synchronize everyone's watch in the classroom.
- ✓ **Global Time** is utilized to provide timestamps for processes and data.
  - **Physical clock**: concerned with "People" time
  - **Logical clock**: concerned with relative time and maintain logical consistency

Memory Consistency

JGI JAIN DEEMED-TO-BE UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND IT

- ✓ In a shared memory multiprocessor …
  - Multiple processors can read and write shared memory
  - Shared memory might be cached in more than one processor
  - Cache coherence ensures **same view** by all processors
- ✓ But cache coherence does not address the problem of
  - **How Consistent** the view of shared memory must be?
    - When should processors see a value that has been updated?
  - Is **reordering** of reads/writes to **different** locations allowed?
    - In a uniprocessor, it is allowed and not considered an issue
    - But in a multiprocessor, it is considered an issue
- ✓ Memory consistency specifies constraints on the …
  - Order in which memory operations can appear to execute

Memory Consistency

- ✓ Consider the code fragments executed by P1 & P2

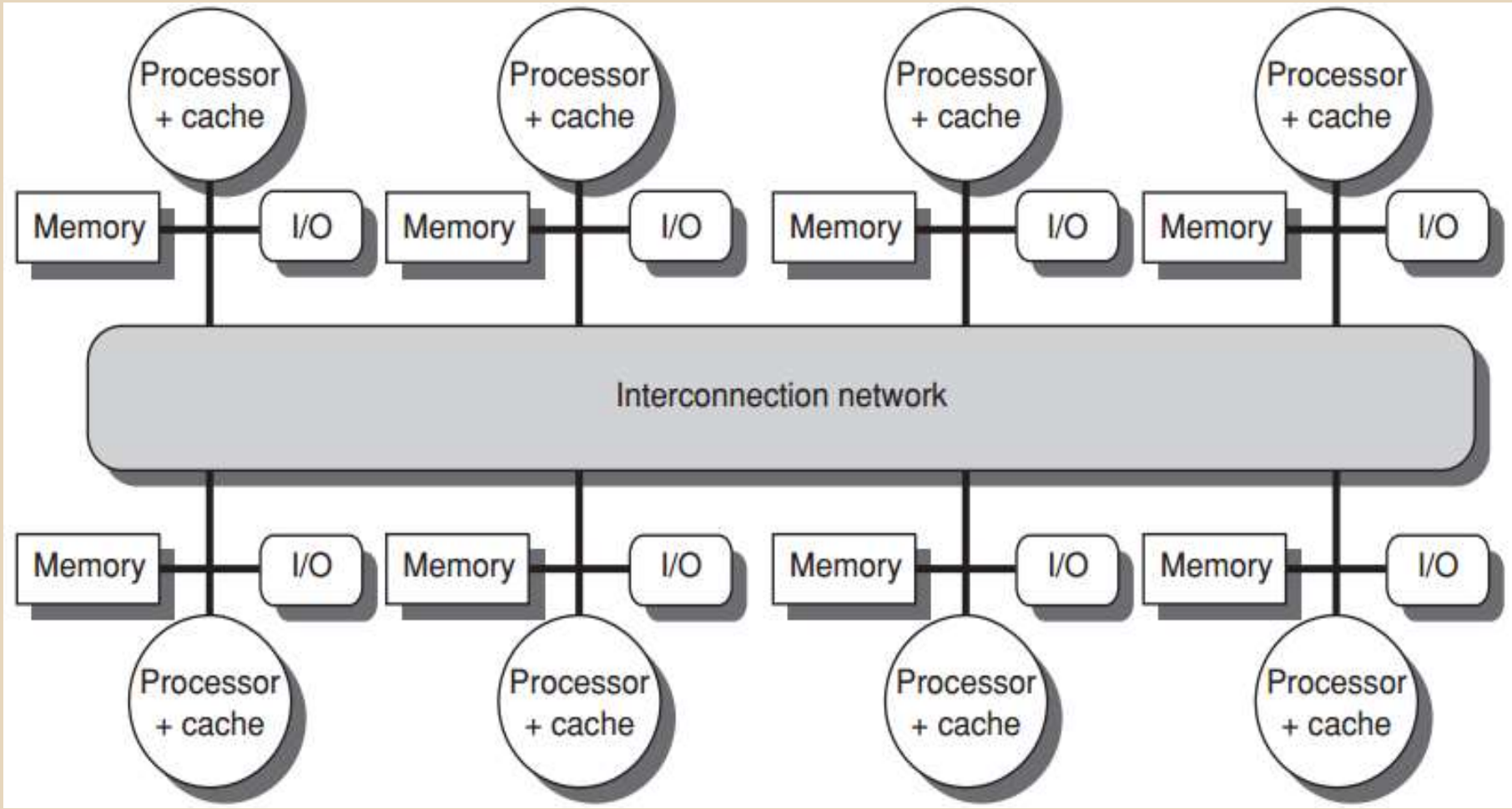
```
P1: A = 0;                  P2: B = 0;
    ...                         ...
    A = 1;                      B = 1;
L1: if (B == 0) ...         L2: if (A == 0) ...
```

- ✓ Can both if statements L1 & L2 be true?
  - Intuition says **NO**, it can't be
  - At least **A** or **B** must have been assigned **1** before **if**
- ✓ But reading of **B** and **A** might take place before writing 1
  - Reading of **B** in P1 is independent of writing **A = 1**
    - Read hit on **B** might take place before Bus Upgrade on writing **A**
  - Same thing might happen when reading **A** in P2

Distributed Shared Memory

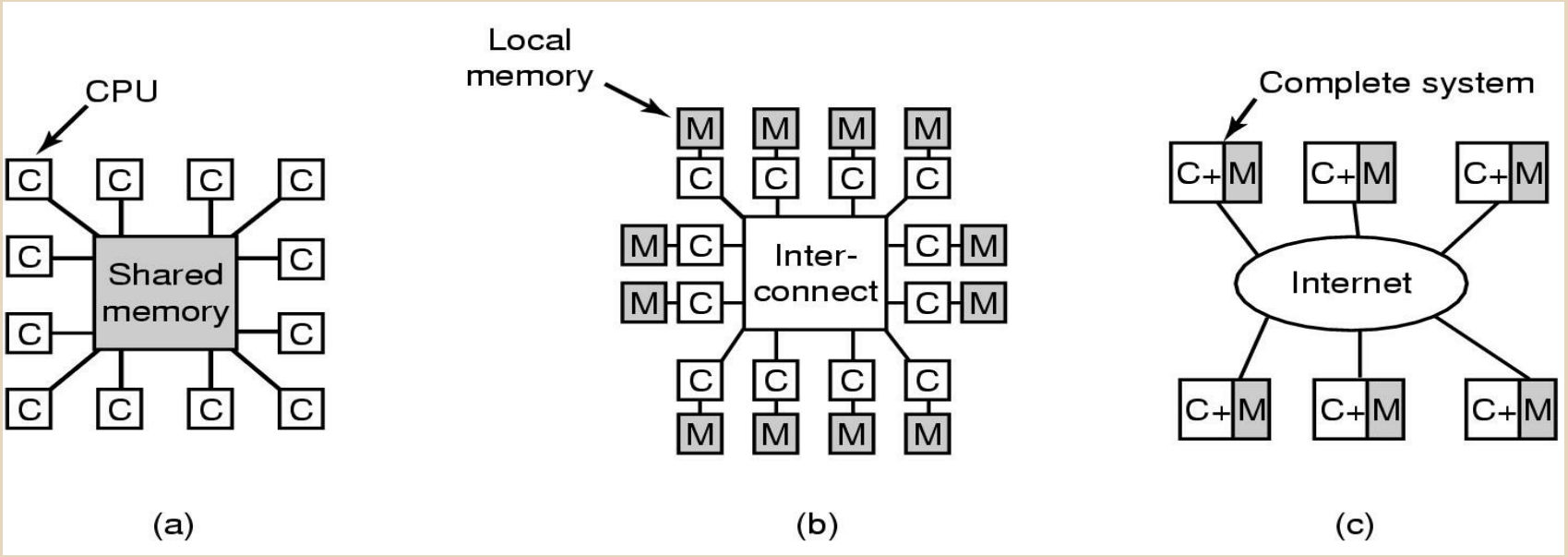
- ✓ Distributing mem among the nodes increases bandwidth & reduces local-mem latency
- ✓ **NUMA**: non uniform memory access time depends on data word location in mem
- ✓ **Disadvantages:**
  - ❑ more complex inter-processor communication
  - ❑ more complex software to handle distributed mem

Distributed Shared Memory



Multiprocessor Systems

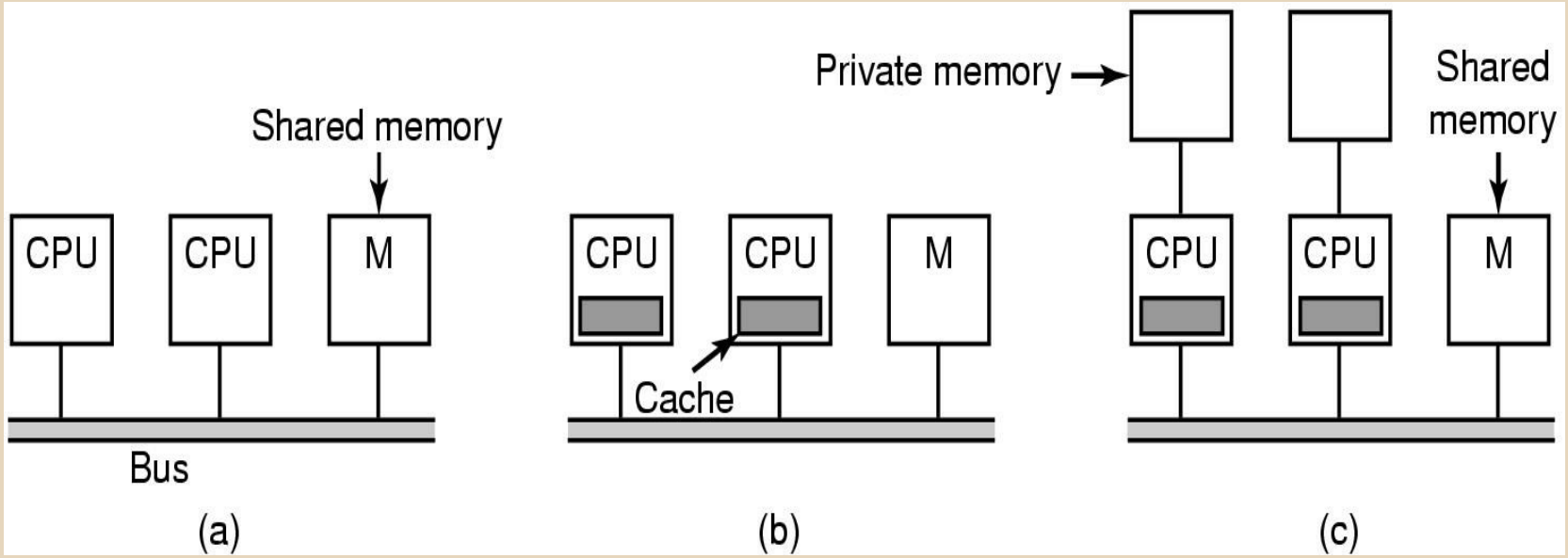
- ✓ Continuous need for faster computers
  - shared memory model
  - message passing multiprocessor
  - wide area distributed system



Multiprocessors

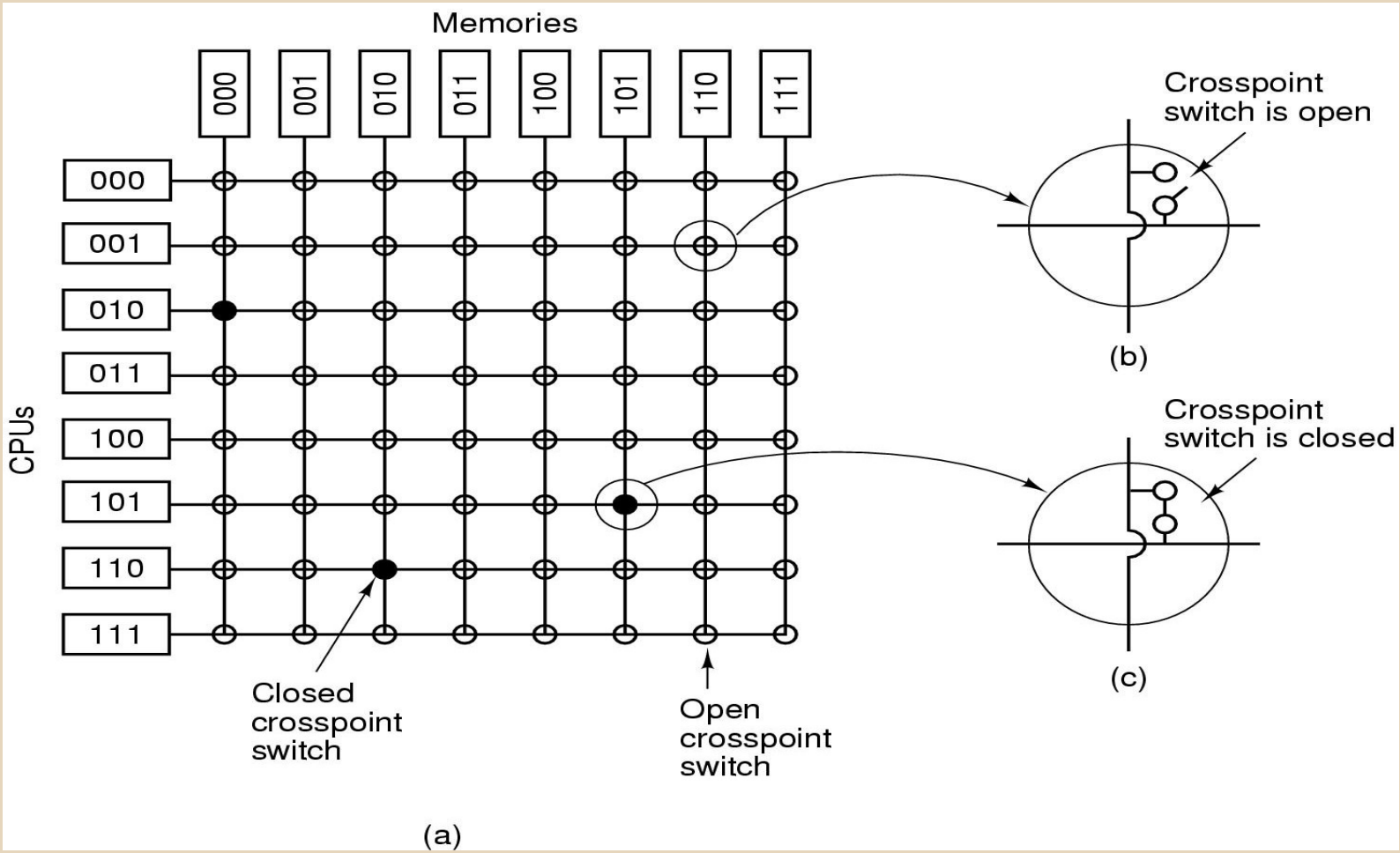
Definition:

A computer system in which two or more CPUs share full access to a common RAM



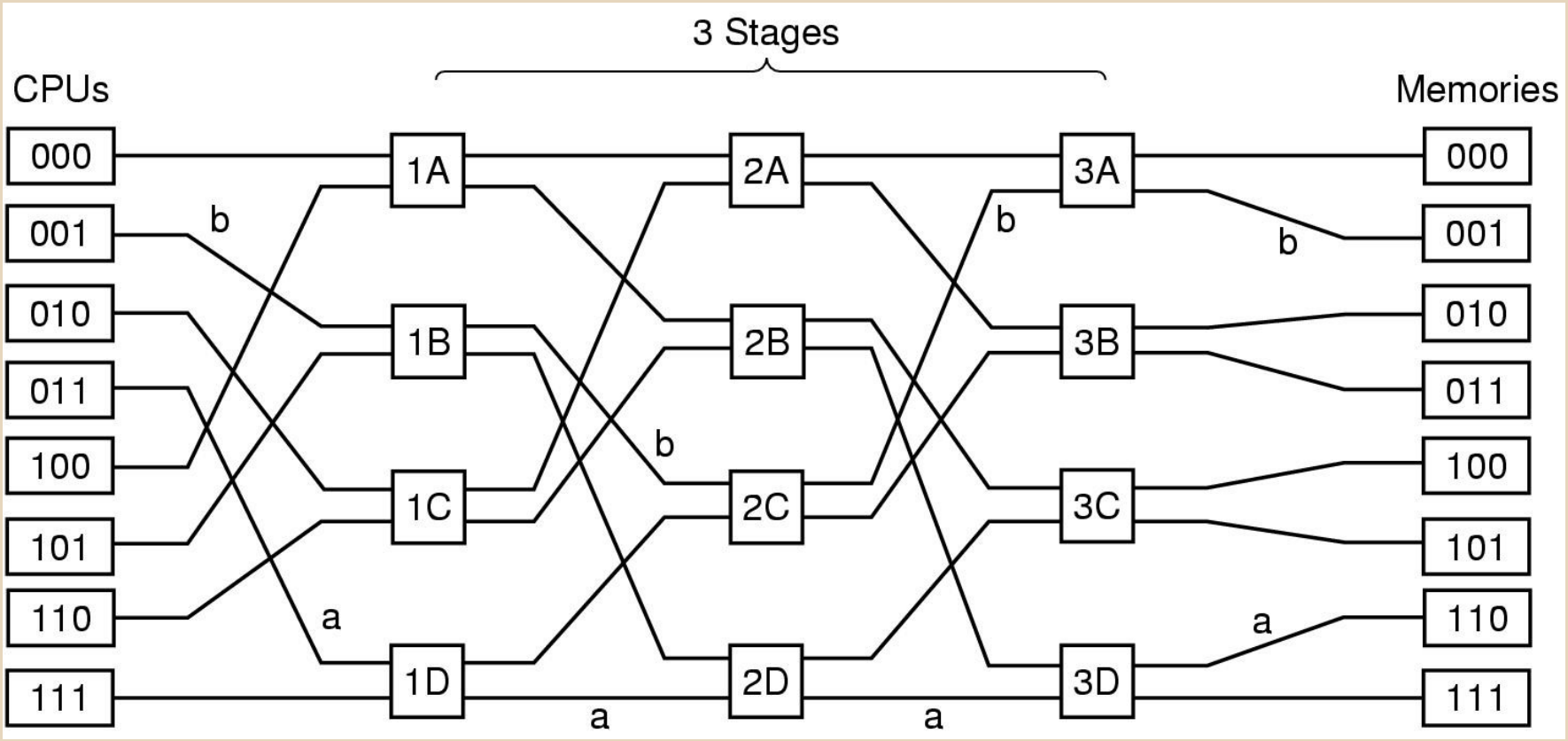
Bus-based multiprocessors

Multiprocessors



UMA Multiprocessor using a crossbar switch

Multiprocessors



Omega Switching Network

Distributed Shared Memory

Replication

(a) Pages distributed on 4 machines

(b) CPU 0 reads page 10

(c) CPU 1 reads page 10