

Chapter 17

Parallel Processing

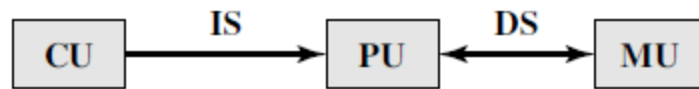
2020.6
Howon Kim

정보보호 및 지능형 IoT연구실 - <http://infosec.pusan.ac.kr>
부산대 지능형융합보안대학원 - <http://aisec.pusan.ac.kr>

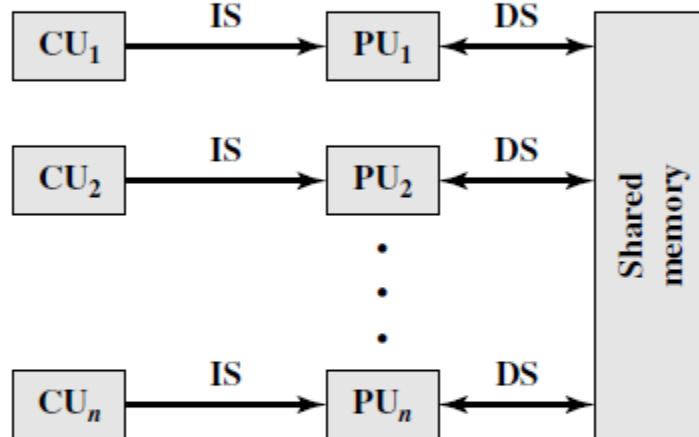
Multiple Processor Organization

- Flynn's Taxonomy is a classification of computer architectures:
 - Single instruction, single data stream - SISD
 - Single instruction, multiple data stream - SIMD
 - Multiple instruction, single data stream - MISD
 - Multiple instruction, multiple data stream- MIMD

Parallel Organizations

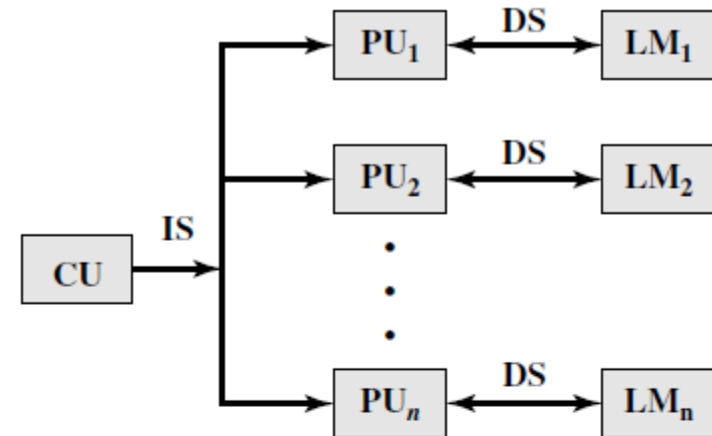


(a) SISD

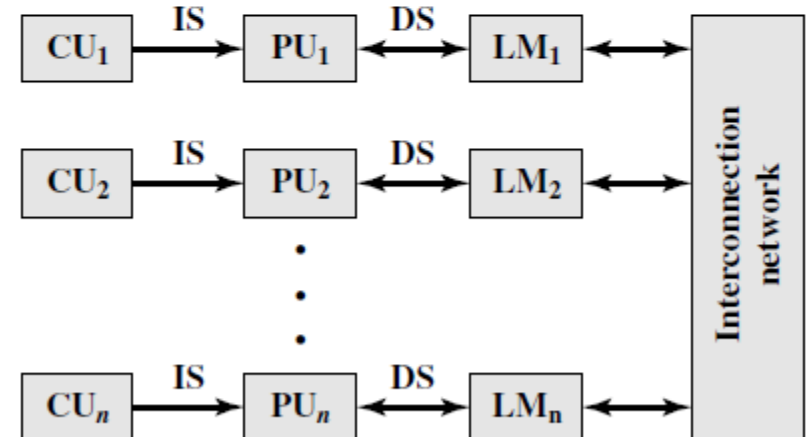


(c) MIMD (with shared memory)

CU = Control unit	SISD = Single instruction,
IS = Instruction stream	= single data stream
PU = Processing unit	SIMD = Single instruction,
DS = Data stream	multiple data stream
MU = Memory unit	MIMD = Multiple instruction,
LM = Local memory	multiple data stream



(b) SIMD (with distributed memory)

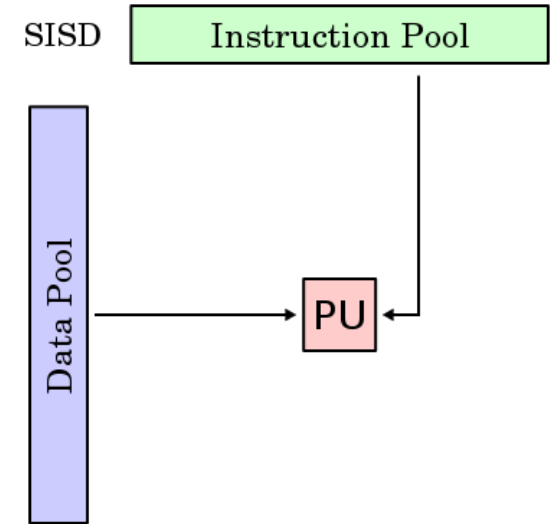


(d) MIMD (with distributed memory)

Figure 17.2 Alternative Computer Organizations

Single Instruction, Single Data Stream - SISD

- Single processor
- Single instruction stream
- Data stored in single memory
- Uni-processor

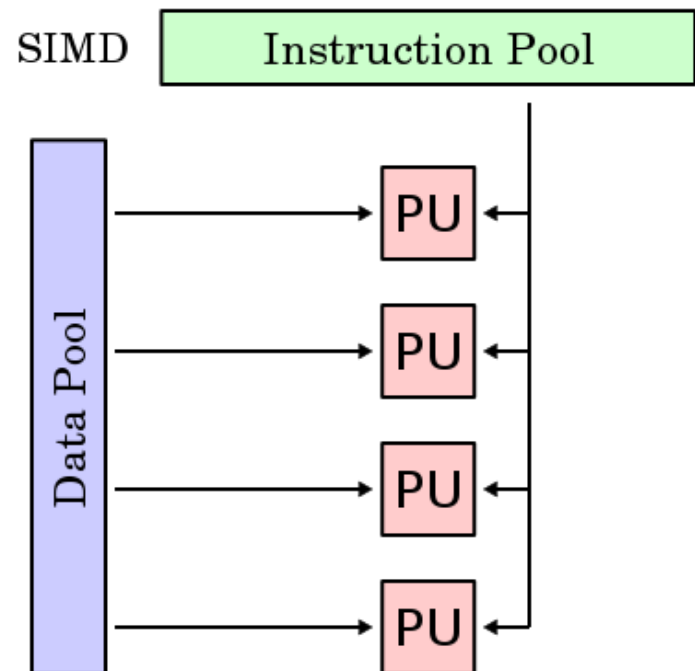


Examples of SISD architecture are the traditional uniprocessor machines like a PC (currently manufactured PCs have multiple processors) or old mainframes.

Single Instruction, Multiple Data Stream - SIMD

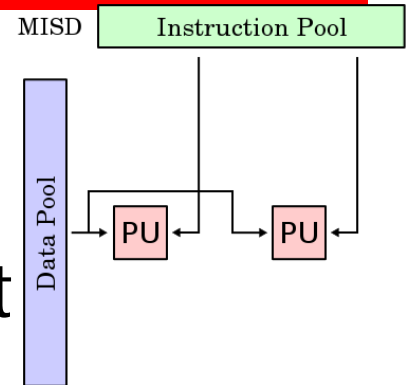
- Single machine instruction
- Controls simultaneous execution
- A lot of processing elements
- Lockstep basis
- Each processing element has associated data memory
- Each instruction executed on different set of data by different processors

Example: Vector and array processors,
GPU



Multiple Instruction, Single Data Stream - MISD

- Sequence of data
- Transmitted to set of processors
- Each processor executes different instruction sequence



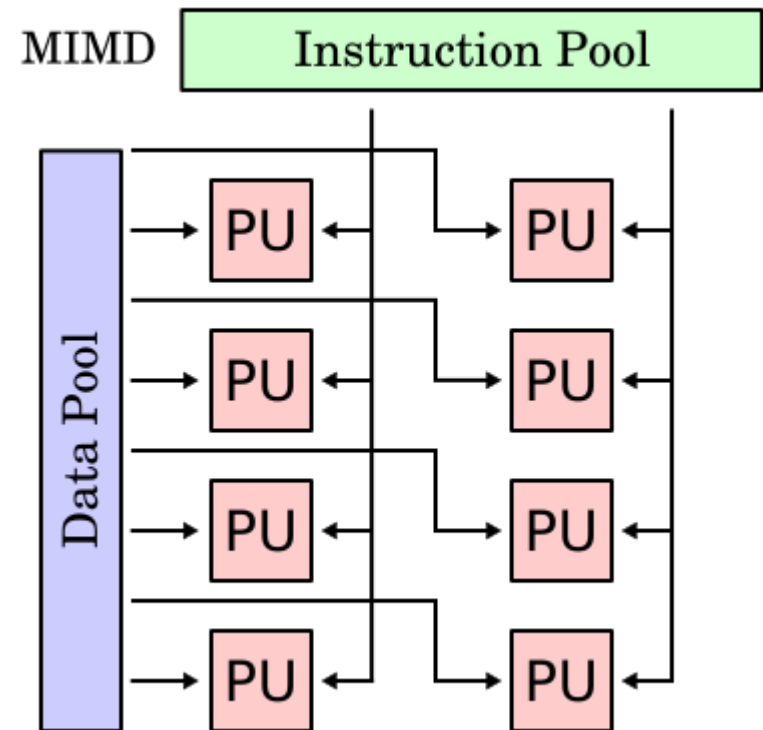
- where many functional units perform different operations on the same data.
- Some people say that the pipeline architectures may belong to this type, though a purist might say that the data is different after processing by each stage in the pipeline. (파이프라인은 진정한 의미에서 MISD로 볼 수는 없음)
- Fault-tolerant computers executing the same instructions redundantly in order to detect and mask errors, in a manner known as task replication, may be considered to belong to this type.
- Not many instances of this architecture exist.

Example: Space Shuttle flight control computer

Multiple Instruction, Multiple Data Stream- MIMD

- Set of processors
- Simultaneously execute different instruction sequences
- Different sets of data
- SMPs, clusters and NUMA systems

A multi-core superscalar processor is an MIMD processor.



Taxonomy of Parallel Processor Architectures

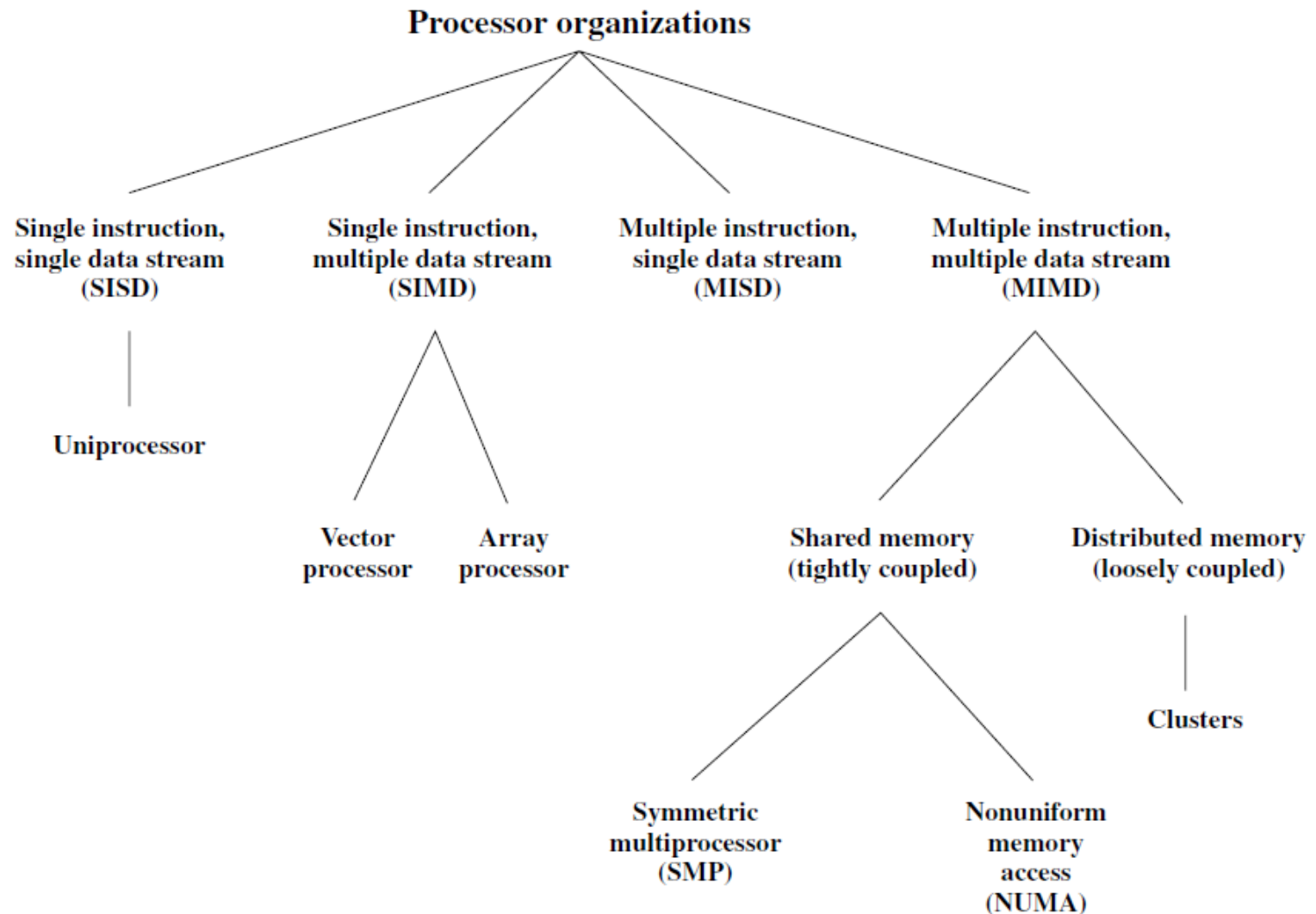


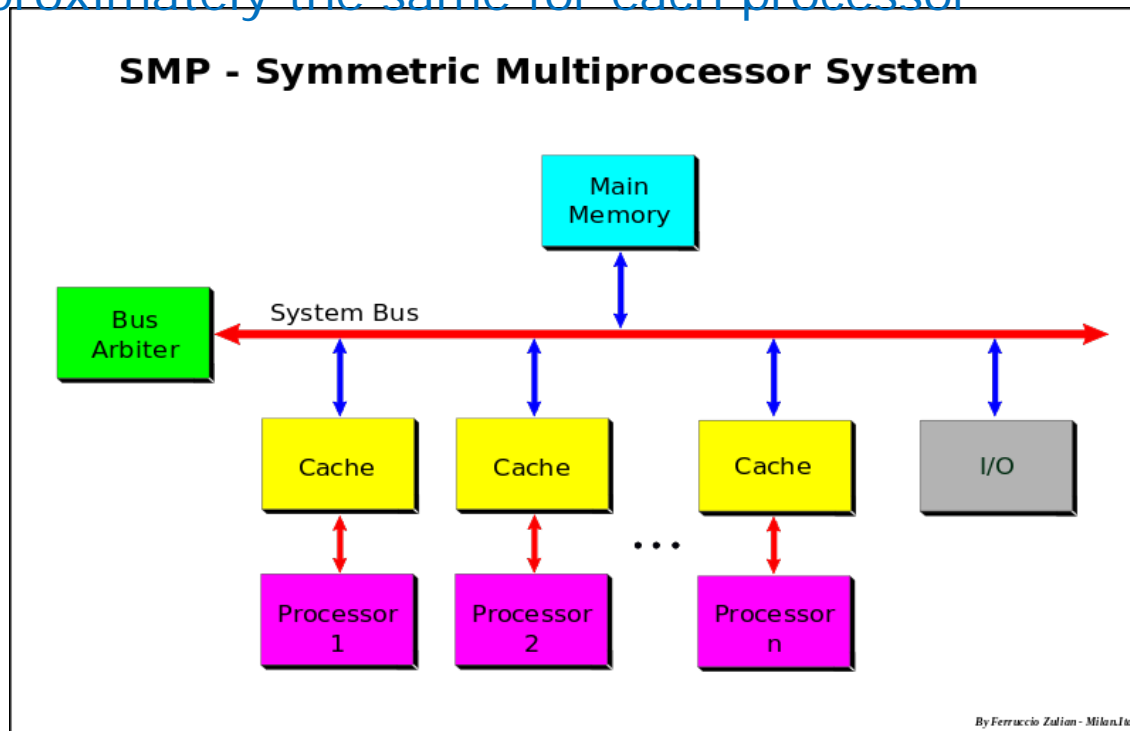
Figure 17.1 A Taxonomy of Parallel Processor Architectures

MIMD - Overview

- General purpose processors
- Each can process all instructions necessary
- Further classified by method of processor communication

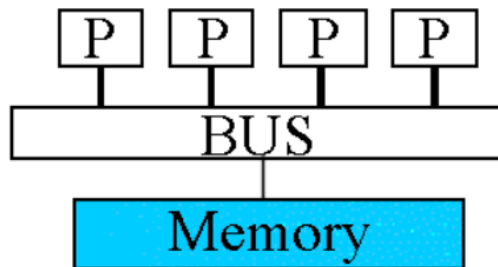
Tightly Coupled - SMP

- Processors share memory
- Communicate via that shared memory
- **S**ymmetric **M**ultiprocessor (SMP)
 - Share single memory or pool
 - Shared bus to access memory
 - Memory access time to given area of memory is approximately the same for each processor



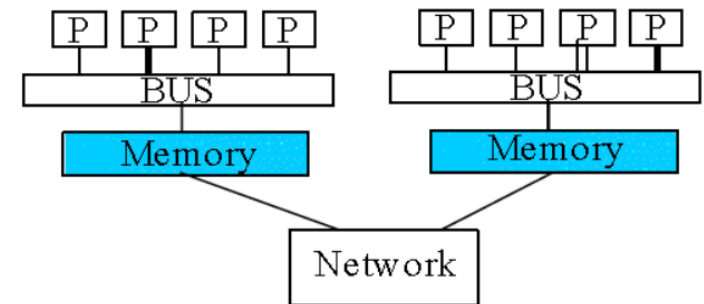
Tightly Coupled - NUMA

- Nonuniform memory access
- Access times to different regions of memory may differ



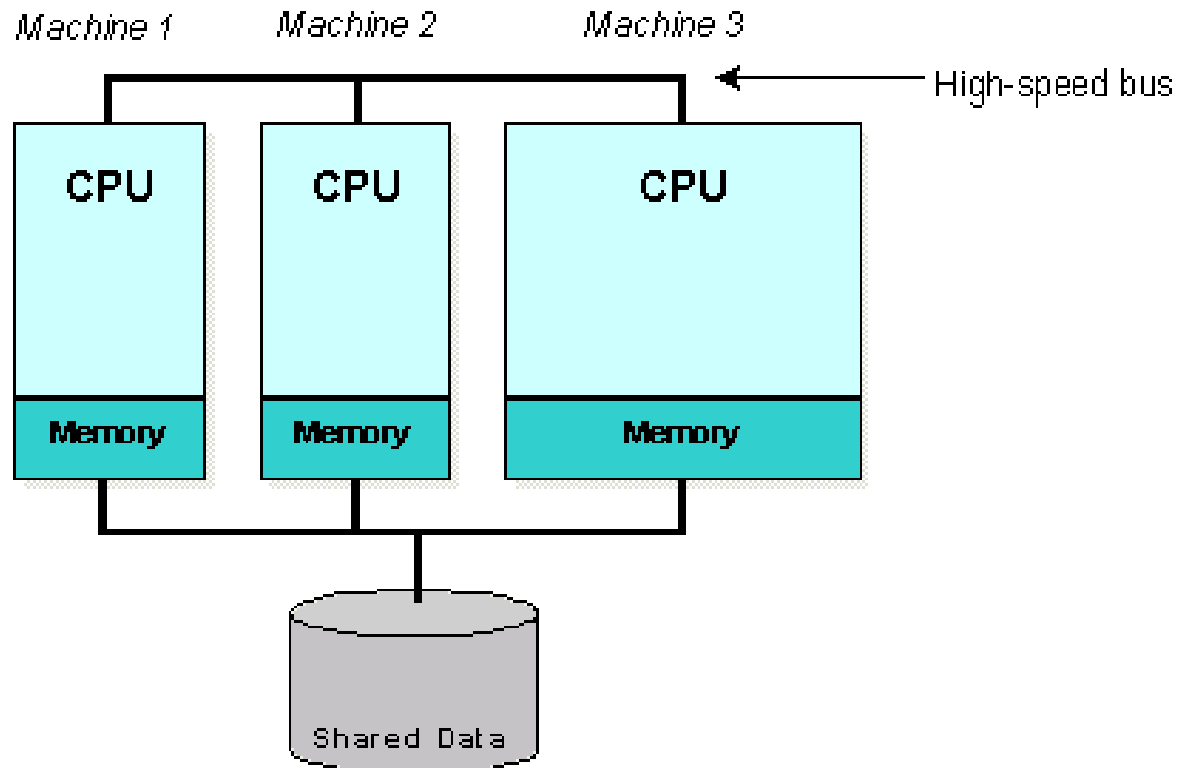
Uniform memory access (UMA) : Each processor has uniform access time to memory - also known as symmetric multiprocessors (SMPs) (example: SUN ES1000)

Non-uniform memory access (NUMA) : Time for memory access depends on location of data; also known as Distributed Shared memory machines. Local access is faster than non-local access. Easier to scale than SMPs (example: SGI Origin 2000)



Loosely Coupled - Clusters

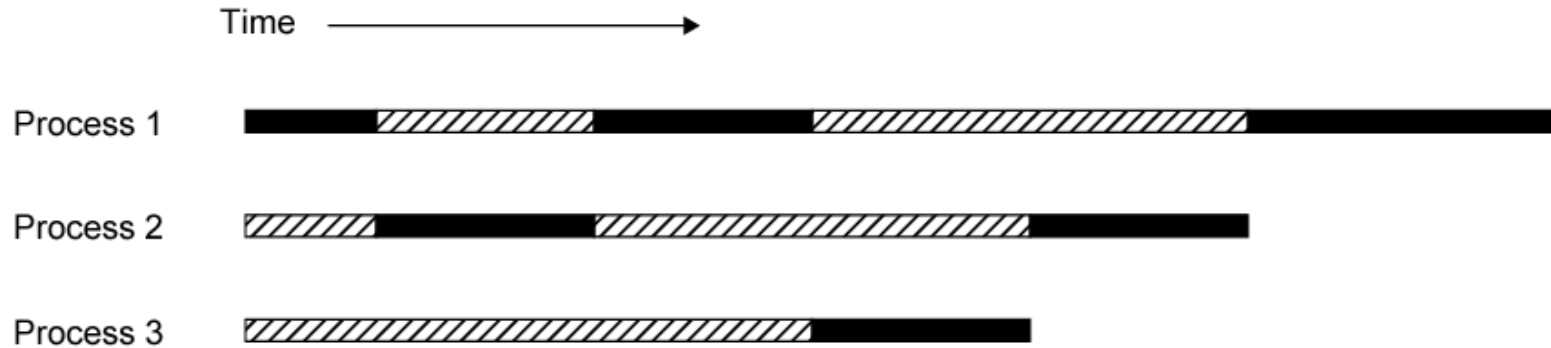
- Collection of independent uniprocessors or SMPs
- Interconnected to form a cluster
- Communication via fixed path or network connections



Symmetric Multiprocessors

- A stand alone computer with the following characteristics
 - Two or more similar processors of comparable capacity
 - Processors share same memory and I/O
 - Processors are connected by a bus or other internal connection
 - Memory access time is approximately the same for each processor
 - All processors share access to I/O
 - Either through same channels or different channels giving paths to same devices
 - All processors can perform the same functions (hence symmetric)
 - System controlled by integrated operating system
 - providing interaction between processors
 - Interaction at job, task, file and data element levels

Multiprogramming and Multiprocessing



(a) Interleaving (multiprogramming, one processor)



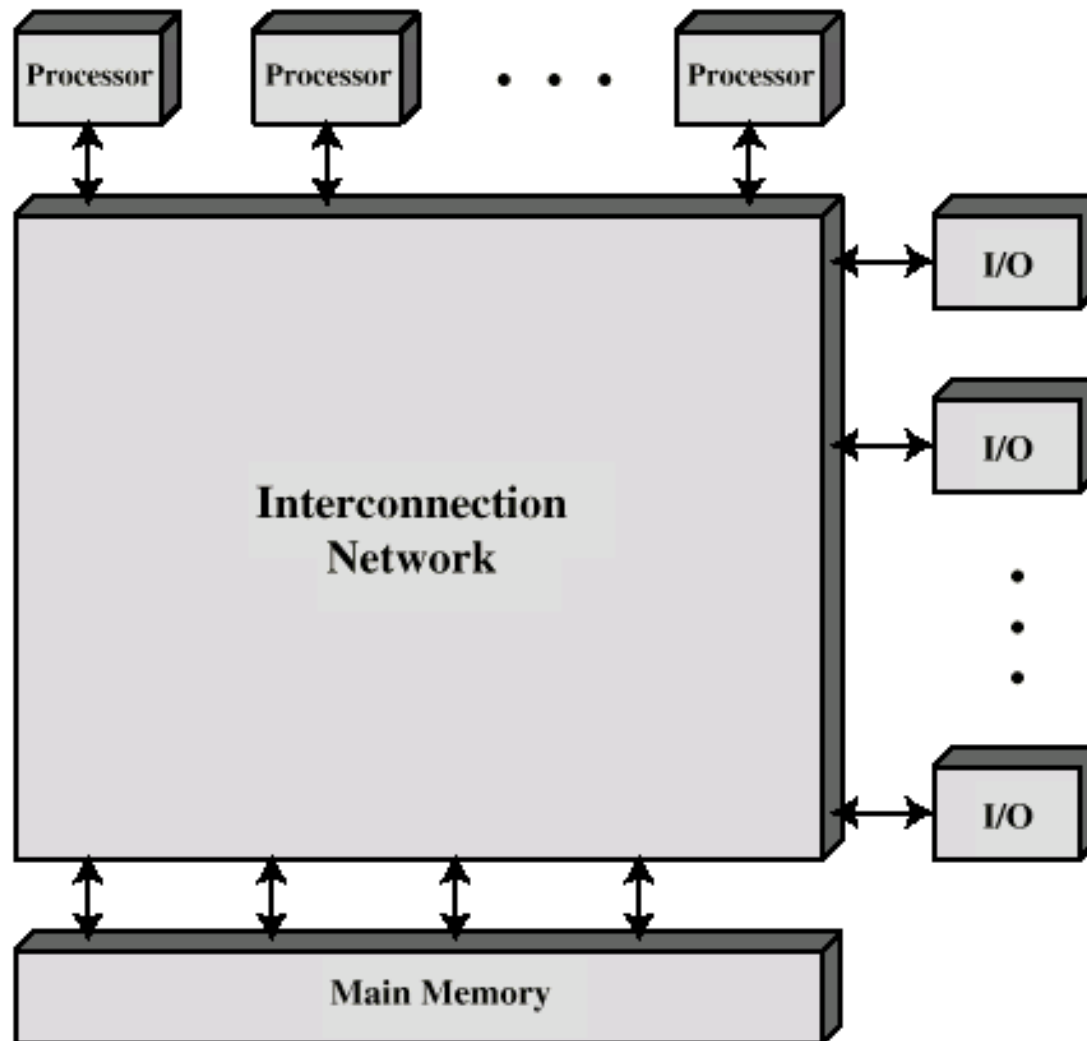
(b) Interleaving and overlapping (multiprocessing; multiple processors)

▨ Blocked ■ Running

SMP Advantages

- Performance
 - If some work can be done in parallel
- Availability
 - Since all processors can perform the same functions, failure of a single processor does not halt the system
- Incremental growth
 - User can enhance performance by adding additional processors
- Scaling
 - Vendors can offer range of products based on number of processors

Block Diagram of Tightly Coupled Multiprocessor

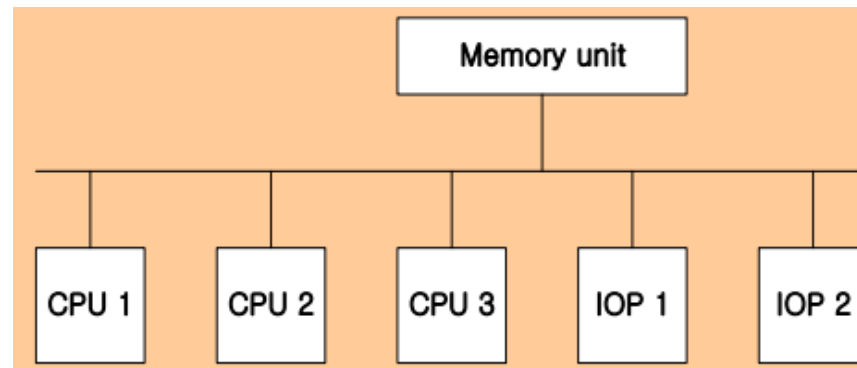


Organization Classification

- Time shared or common bus
- Multiport memory
- Central control unit

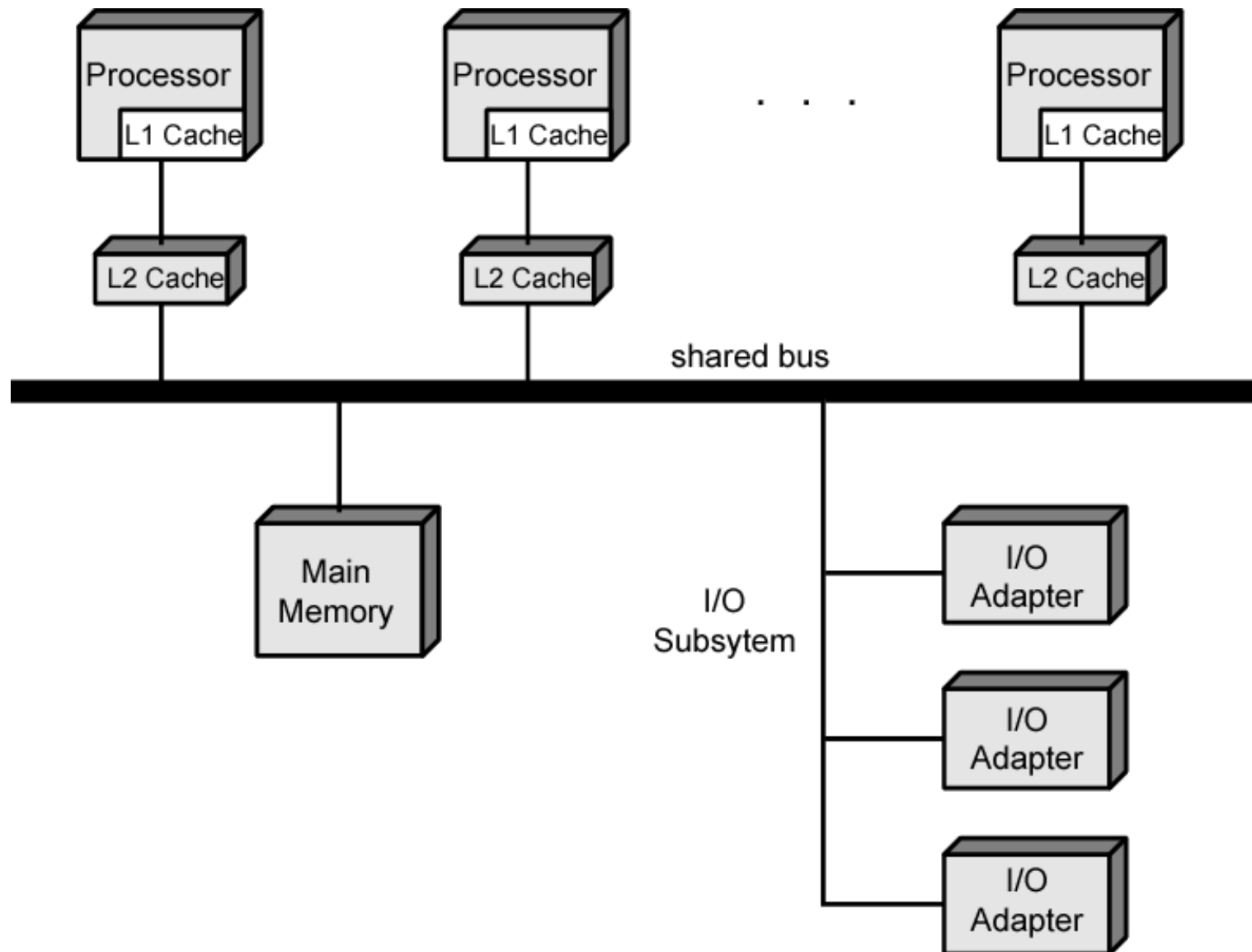
Time Shared Bus

- Simplest form
- Only one processor can communicate with the memory or another processor at any given time
- Structure and interface similar to single processor system
- Following features provided
 - Addressing - distinguish modules on bus
 - Arbitration - any module can be temporary master
 - Time sharing - if one module has the bus, others must wait and may have to suspend
- Now have multiple processors as well as multiple I/O modules



Time-shared single common bus system

Symmetric Multiprocessor Organization



Time Shared Bus - Advantages

- Simplicity
- Flexibility
- Reliability

Time Shared Bus - Disadvantage

- Performance limited by bus cycle time
- Each processor should have local cache
 - Reduce number of bus accesses
- Leads to problems with cache coherence
 - Solved in hardware - see later



Thank you !