Other types of operating system

- Real-Time Operating System (RTOS)
- Network Operating System (NOS)
- Parallel processing
- Multithreading Operating System

Real-Time Operating System (RTOS)

- A real-time system has to respond to input within a finite and specified time often referred to as deadline. The correctness depends not only on the logical result but also the time it was delivered.
- RTOS manages the resources so that particular operation executes in precisely the same amount of time every time it occurs.
- These systems are required in special applications where the response is needed immediately such as industrial control systems, weapon systems and medical products.

RTOS cont…

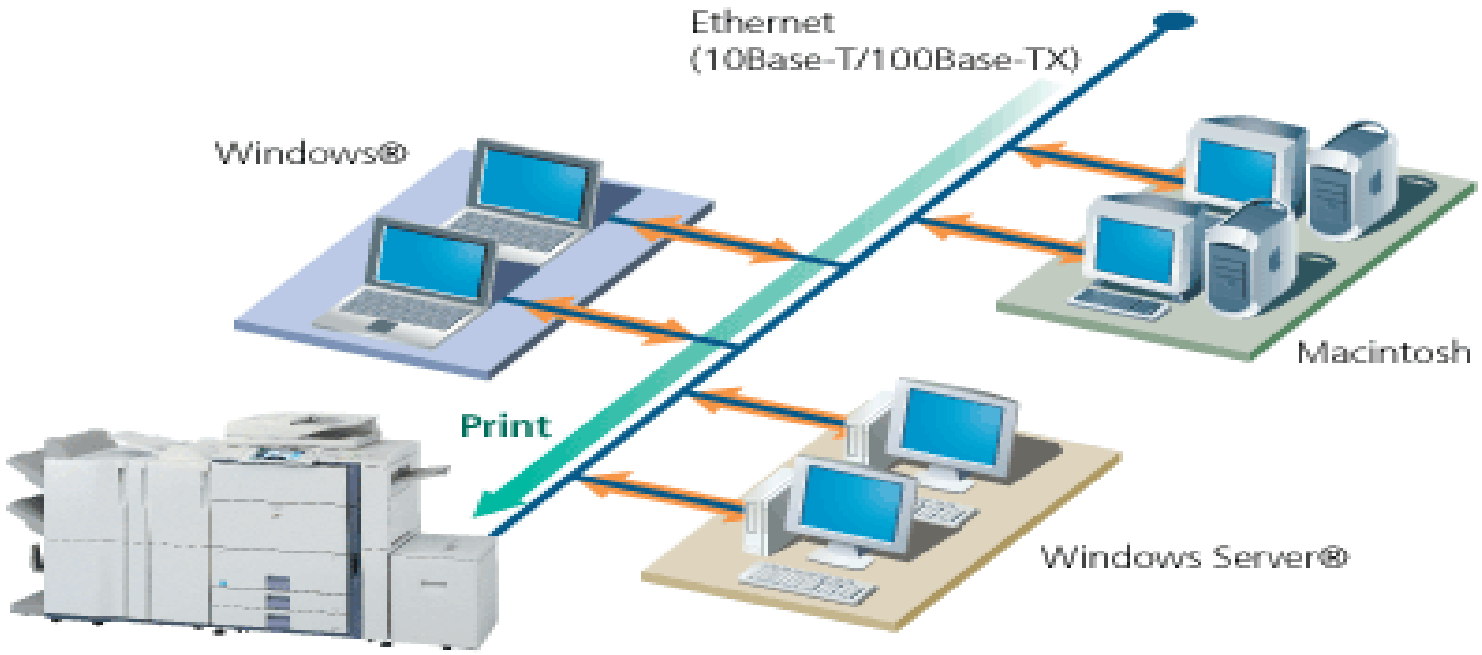
- RTOS systems can be hard and soft real time.
  - Hard-real time: is purely deterministic and time constraint system. For example users expected the output for the given input in 10sec then system must process the input data and give the output exactly by 10th second. Here in the above example 10sec is the deadline to complete process for given data. It should not give the output by 11th second or by 9th second, exactly by 10th second it should give the output. In the hard real time system meeting the deadline is very important. If the system fails to meet the deadline even once, the entire system performance is worthless and will fail.

RTOS cont…

- Soft-real time: even if the system fails to meet the deadline, possibly more than once, the system is not considered to have failed. In this case the results of the requests are not worthless value after its deadline, rather it degrades as time passes after the deadline (Streaming audio-video, mp3 players).

Network operating system (NOS)

- NOS is an OS specifically designed to run on servers and enables the servers to serve the requests of client computers on the network. The main purpose of this operating system is to provide services to clients and allow the access of shared resources on the network.

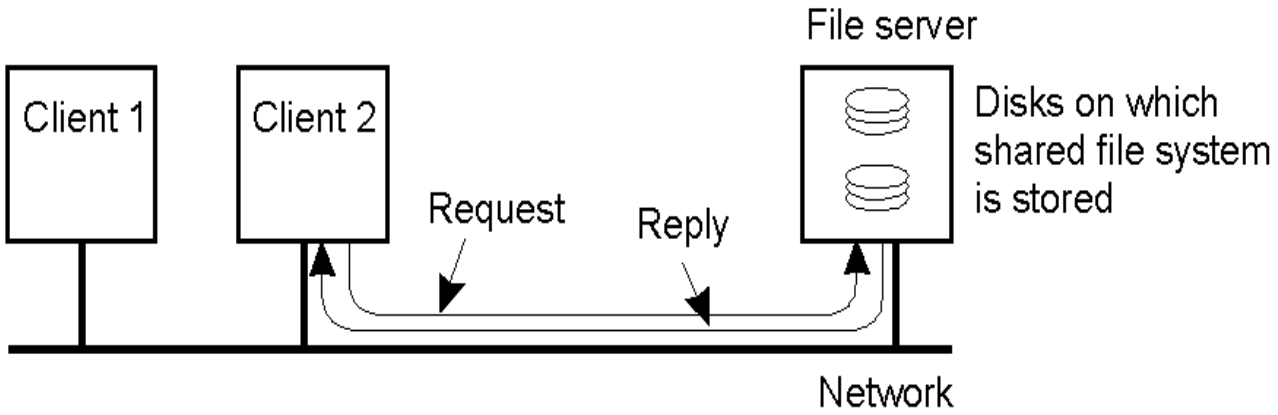


NOS: Microsoft Windows Server 2003, 2008, and 2012, UNIX (Mac OS X server, BSD), Linux (RHEL, SUSE, ubuntu server), Novell NetWare, and.

NOS cont…

- In network operating system, the users are fully aware of the existence of the multiple computers on the network.

- Access to resources of various machines is done by:
  - Remote logging into the remote machine. Each computer runs its own operating system and it has its own local users. When a user wants to access any other machine, he must require some kind of remote login to access the other machine.
  - Transferring data from remote machines to local machines, via the FTP, SSH, etc.



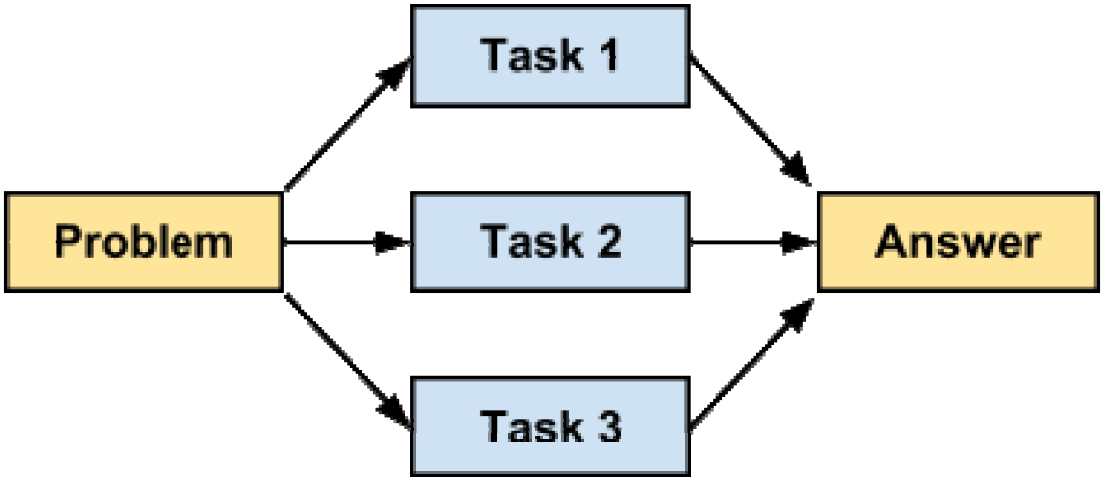
Parallel processing

- In the early days of computing, programs were entirely serial, which limited them to performing a single task at a time. The next instruction could not be performed until the previous instruction was complete.

Parallel processing cont…

- However, many tasks can be completed more efficiently by allowing work to be performed simultaneously. This need was addressed by the development of concurrent computing methods, which use a set of two or more processors or computers to solve a larger problem.

- Concurrency refers to the sharing of resources in the same time. For instance, several processes share the same CPU (or CPU cores) or share memory or an I/O device.



Parallel processing cont…

- Even in the case that they have a single processor, they often have two or more cores which are capable of working in parallel. This allows tasks to be accomplished independently from one another.

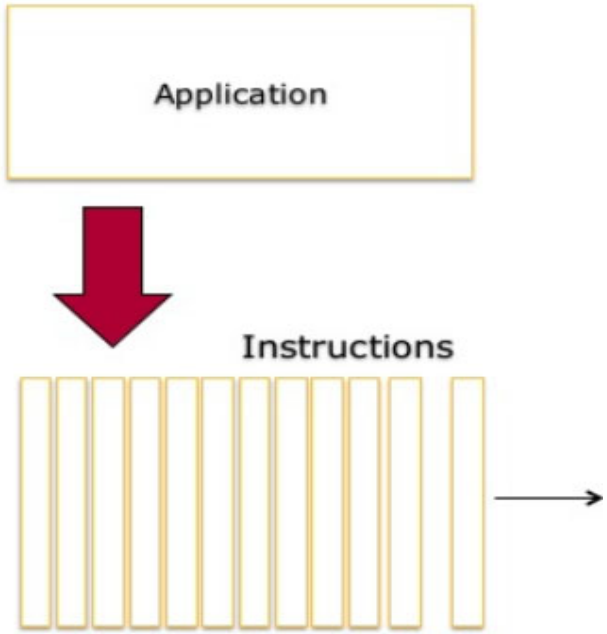
Parallel processing cont…

- Using parallelism is an obvious way to achieve speedup when it is required and multiprocessing is used to achieve parallelism.

- Today, there are problems that are larger than what a single CPU can handle. Thus, it is becoming more common to use multiple processors to solve large-scale complex computational tasks for improving the execution time of computationally intensive programs.

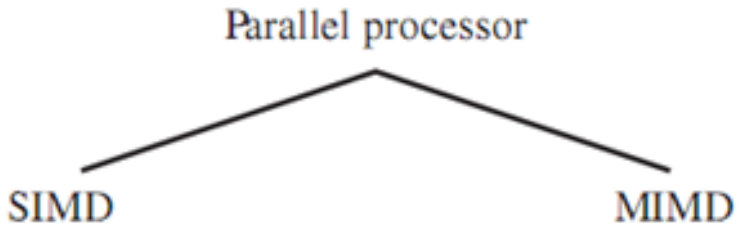
Flynn taxonomy

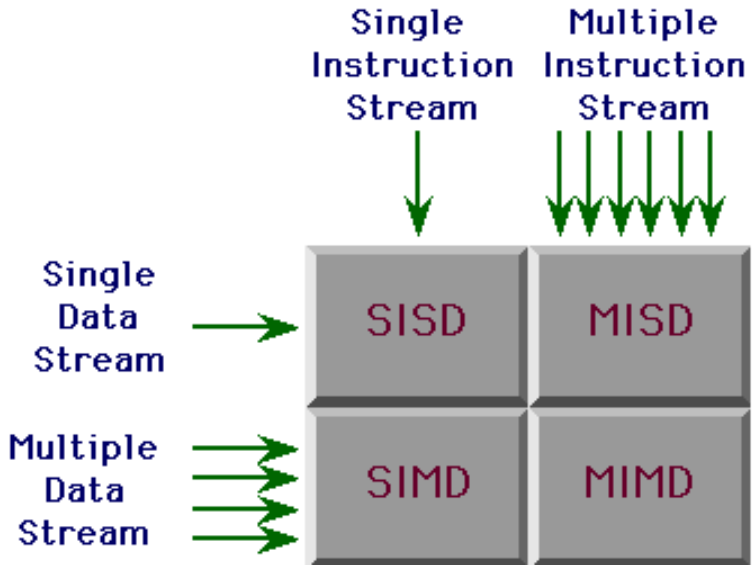
- A CPU operates by fetching instructions and operands from memory, executing the instructions, and placing the final results in memory.
- Hence, to run programs, we need:
  - Instruction stream: sequence of instructions
  - Data stream: sequence of data
- There are different ways to classify Parallel processing methods. The widely used classification is called Flynn's Taxonomy which is based upon the number of **concurrent** instruction and data streams available in the architecture.
- According to Flynn, Parallel processing may occur in the instruction stream, in the data stream, or in both.



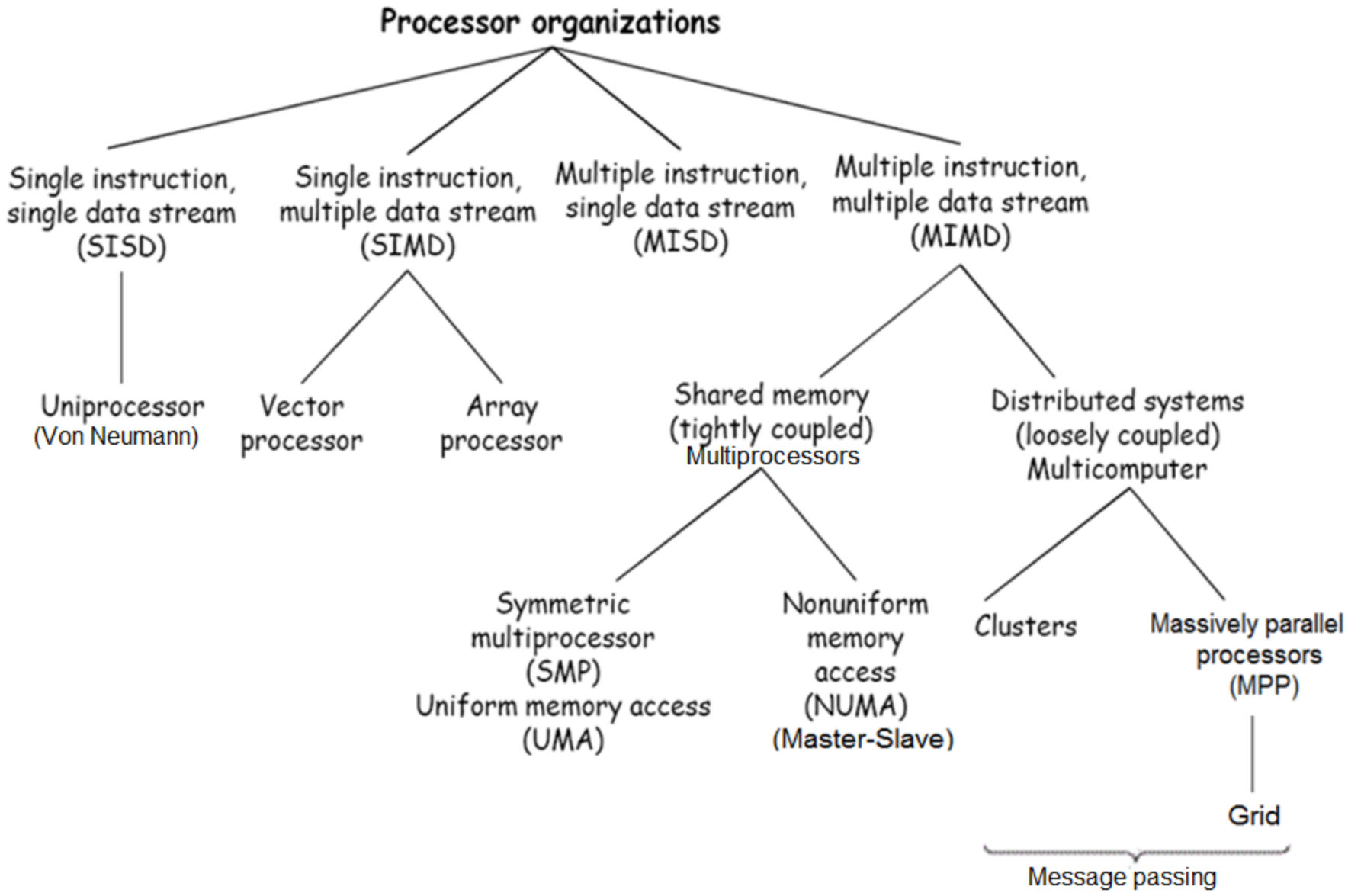
Flynn taxonomy cont…

- Flynn has proposed a broad classification based on the number of simultaneous instruction streams and data streams seen by the processor during program execution:
  - Single instruction stream, single data stream (SISD)/Scalar
  - Single instruction stream, multiple data stream (SIMD)
  - Multiple instruction stream, single data stream (MISD)
  - Multiple instruction stream, multiple data stream (MIMD)





Processor Architectures



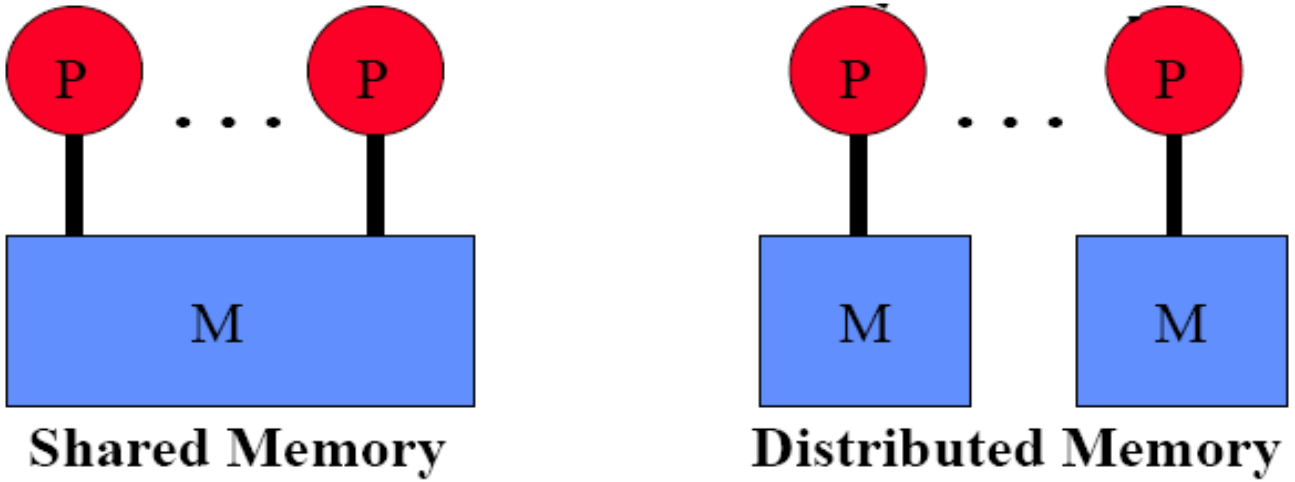
Processing element/ unit

- A parallel processor is a collection of processing elements that communicate to solve large problem fast.

- A Processing Element/ Processing Unit (PE/PU) is a hardware element that runs instructions. Depends on scenario, the PE can be ALU, core, or CPU.

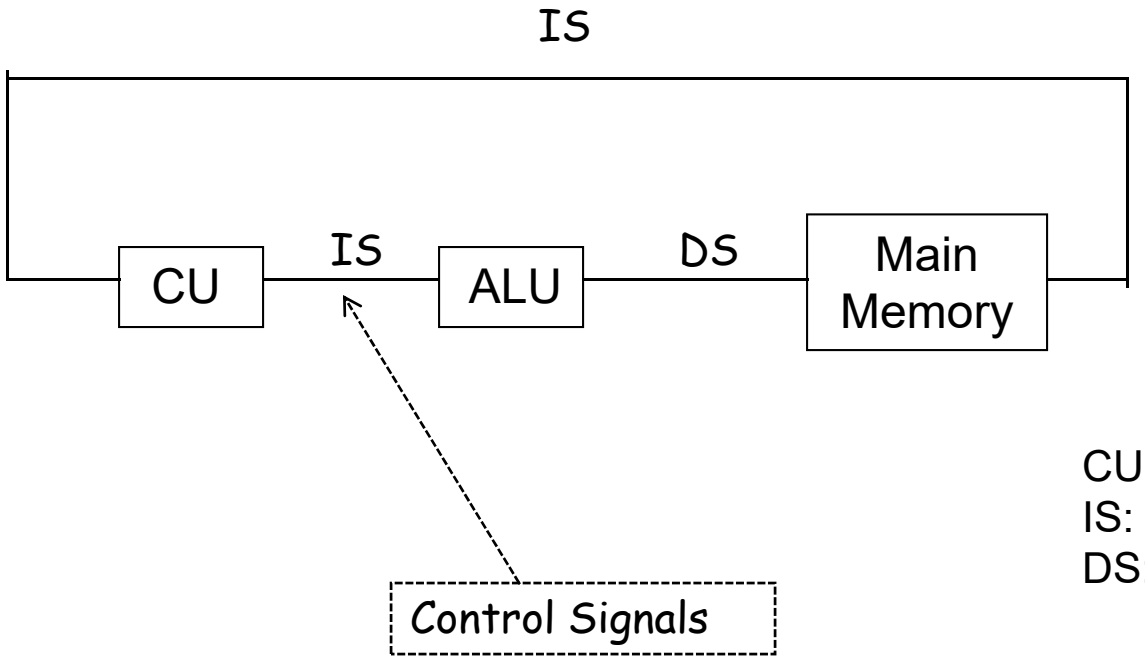
Shared vs. Distributed memory

- In shared memory (SM) cooperation model all processing units share the same global memory and have the same address spaces.
- In distributed/private memory (DM) cooperation model, there is a private memory for each processing unit.



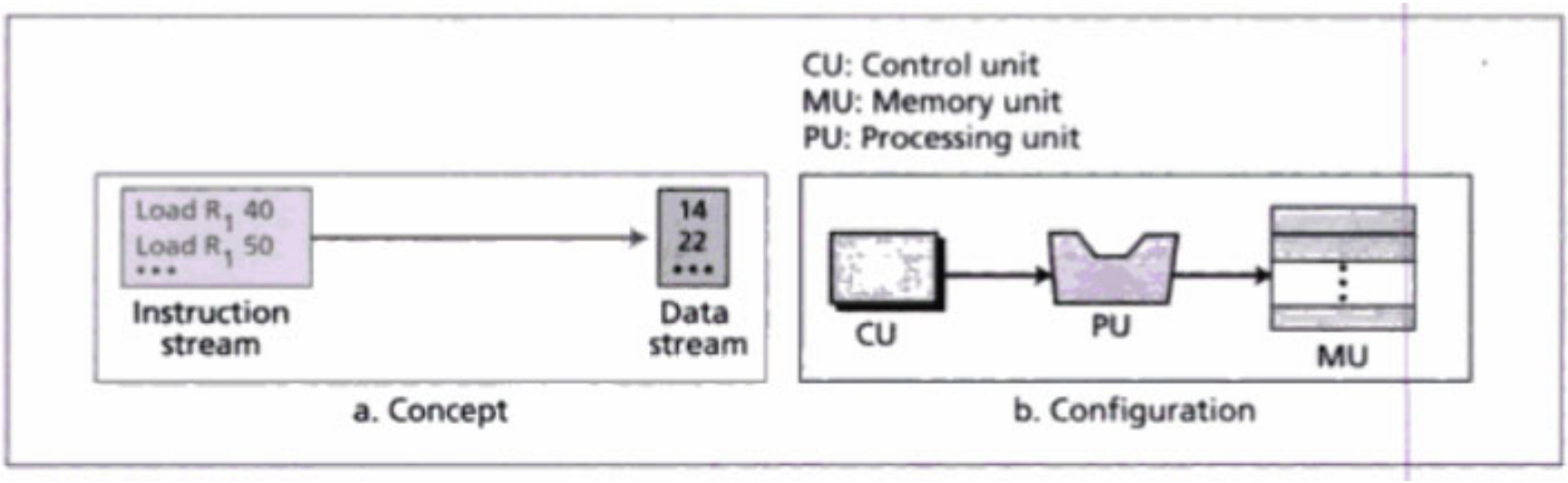
SISD

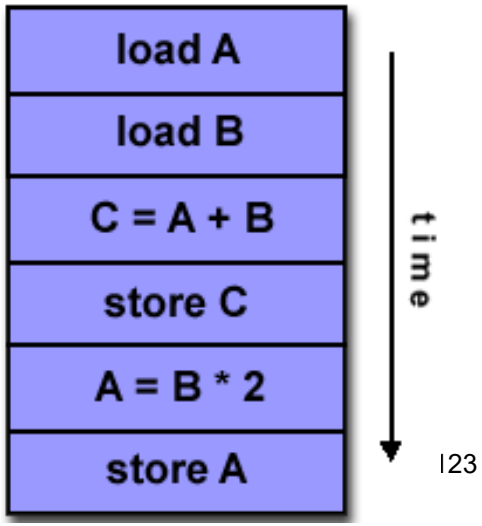
- In a single processor/Uni-processor/scalar processor computer, a single stream of instructions is generated by the program. The instructions operate on a single stream of data items (scalar).
- It is Von Neumann architecture.
- Algorithms for SISD computers provide no parallelism in either the instruction or data streams.



CU: Control Unit
IS: Instruction Stream
DS: Data Stream

SISD cont…

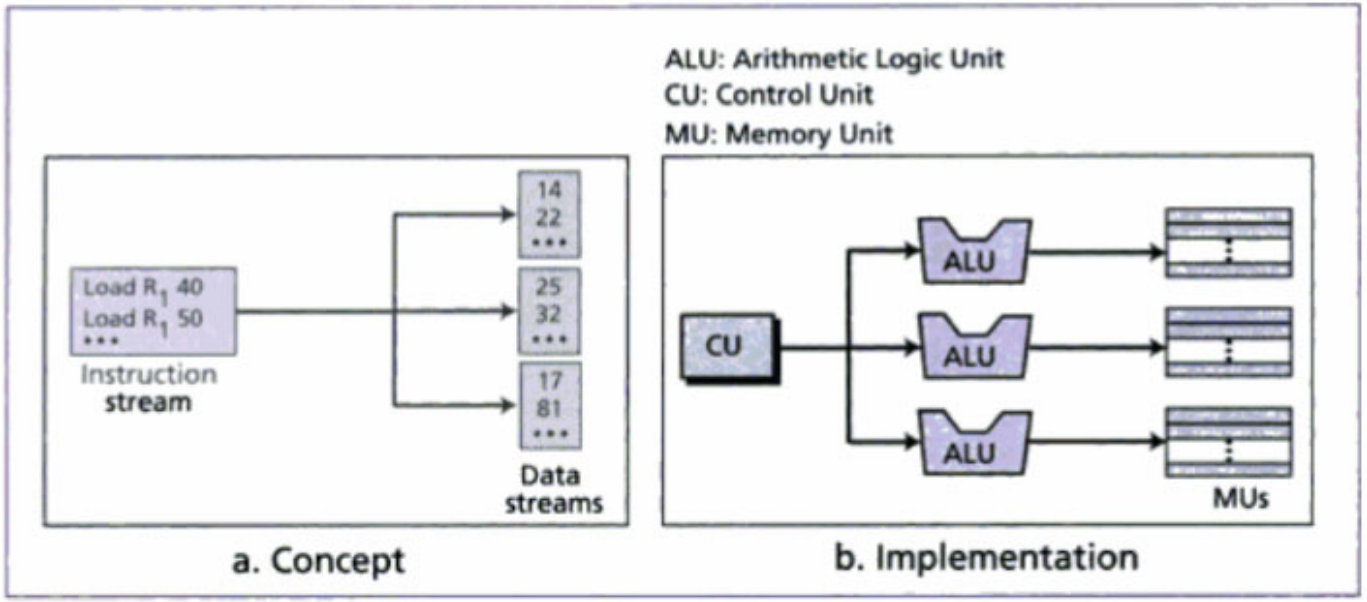

CU: Control unit
MU: Memory unit
PU: Processing unit
Load R1 40
Load R1 50
Instruction stream
14
22
Data stream
a. Concept
CU
PU
MU
b. Configuration


load A
load B
C = A + B
store C
A = B * 2
store A
time

SIMD

- The control unit broadcast the same instruction to each processor. Each instruction is executed on a different set of data by different PUs in parallel. The processing elements (PEs) work synchronously.

- SIMD architectures comprise a number of processors (arrays of ALUs), each executing the same set of instructions.



SIMD applications

- SIMD machines are useful for computations that repeat the same calculation on many sets of data such as multimedia applications like image processing.

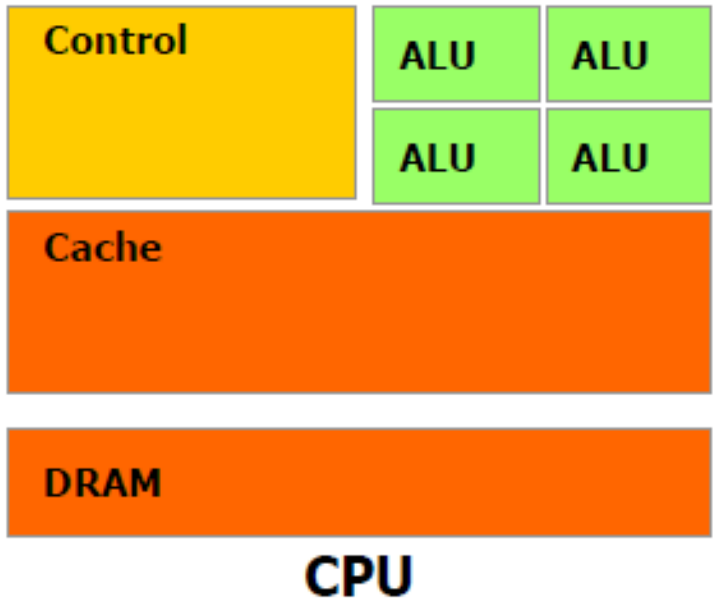
```
for (i=0;i<n;i++)
c[i]=a[i]+b[i].
```

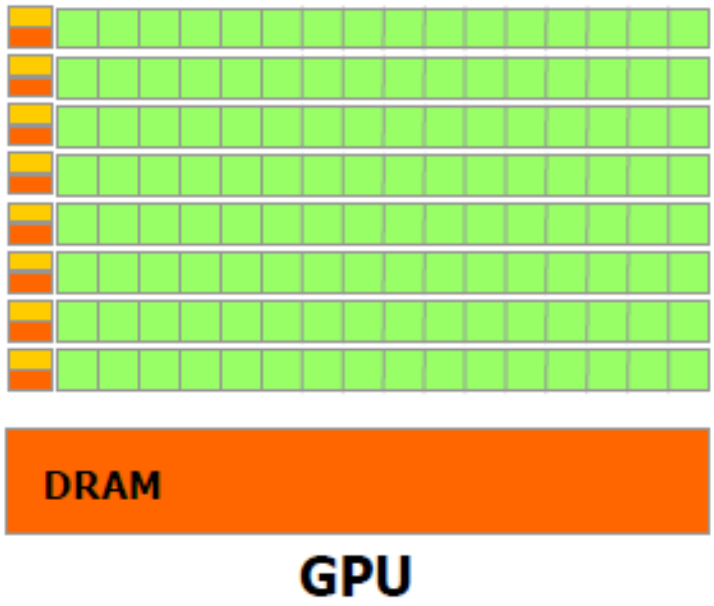
C = A + B
With 4 processors:

C11 = A11 + B11
C12 = A12 + B12
C21 = A21 + B21
C22 = A22 + B22

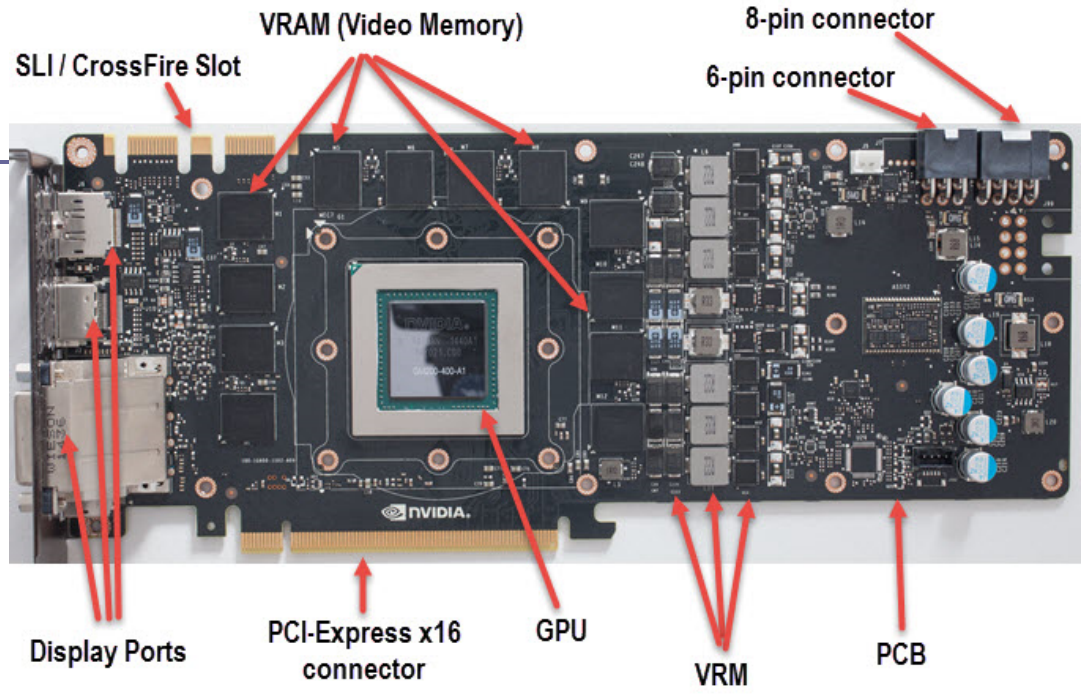
SIMD applications cont…

- SIMD architecture is used in designing **GPU**.
- GPUs have lots of small PEs, compared to a few larger ones on the CPU. This allows for many parallel computations, like calculating a color for each pixel on the screen.






msi
GEFORCE RTX


VRAM (Video Memory)
8-pin connector
SLI / CrossFire Slot
6-pin connector
NVIDIA.
Display Ports
PCI-Express x16 connector
GPU
VRM
PCB

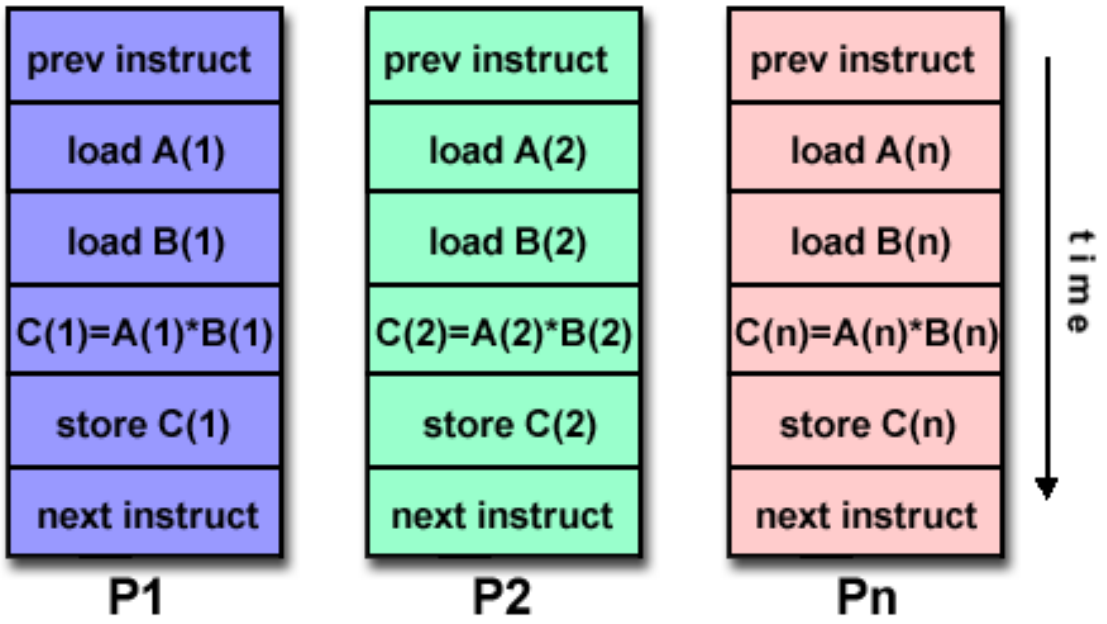
Concepts

- Synchronous: means that they are built in such a way to guarantee that all processing elements will receive the same instruction at the same time, and thus all will potentially be able to execute the same operation simultaneously.

- Deterministic (repeatable/ predictable) : at any one point in time, there is only one instruction being executed, even though multiple units may be executing it. So, every time the same program is run on the same data, using the same number of execution units, exactly the same result is guaranteed at every step in the process. SIMD architectures are deterministic.

SIMD cont…

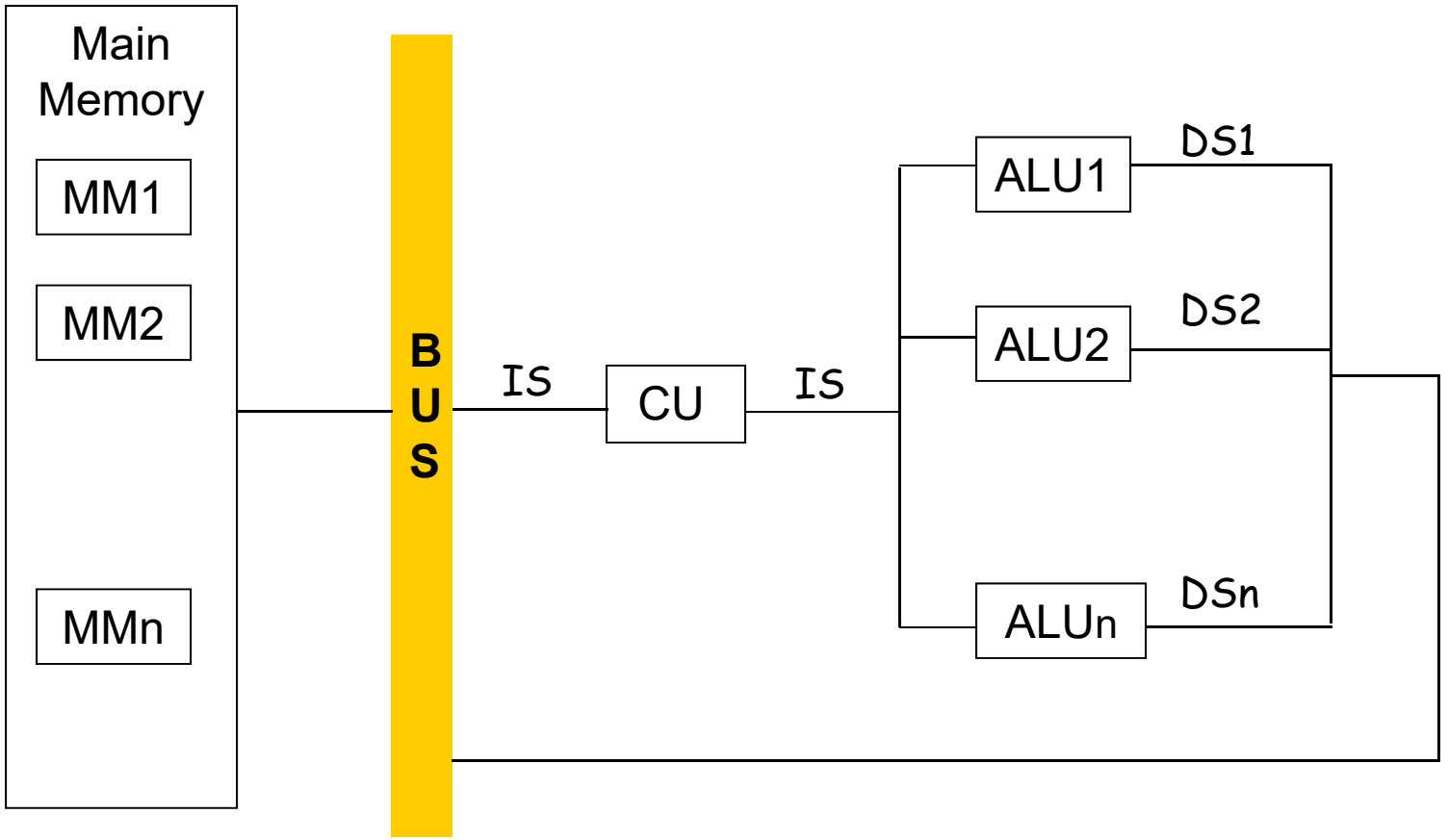
- The memory in both SIMD and MIMD can be either:
  - Shared memory with bus interconnection
  - Distributed/private memory with network interconnection and message passing



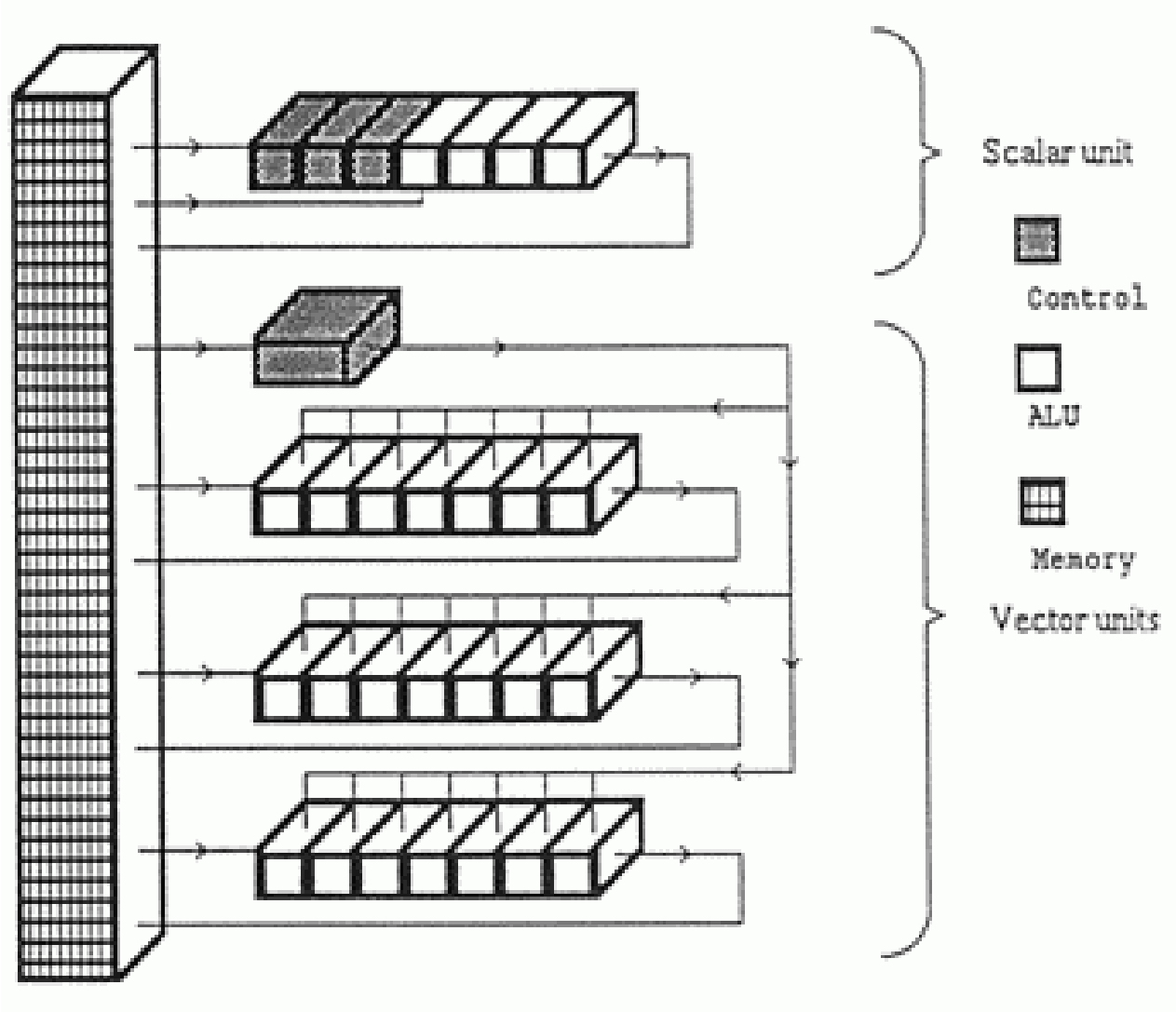
SIMD with shared memory

- Many scientific and engineering computations require the same operation to be performed on every element of a list or matrix.
- Main memory is divided into modules for generating multiple data streams.
- Each processor takes the data from its own memory module and hence it has on distinct data streams.
- Every processor unit must be allowed to complete its instruction before the next instruction is taken for execution. Thus, the execution of instructions is synchronous.

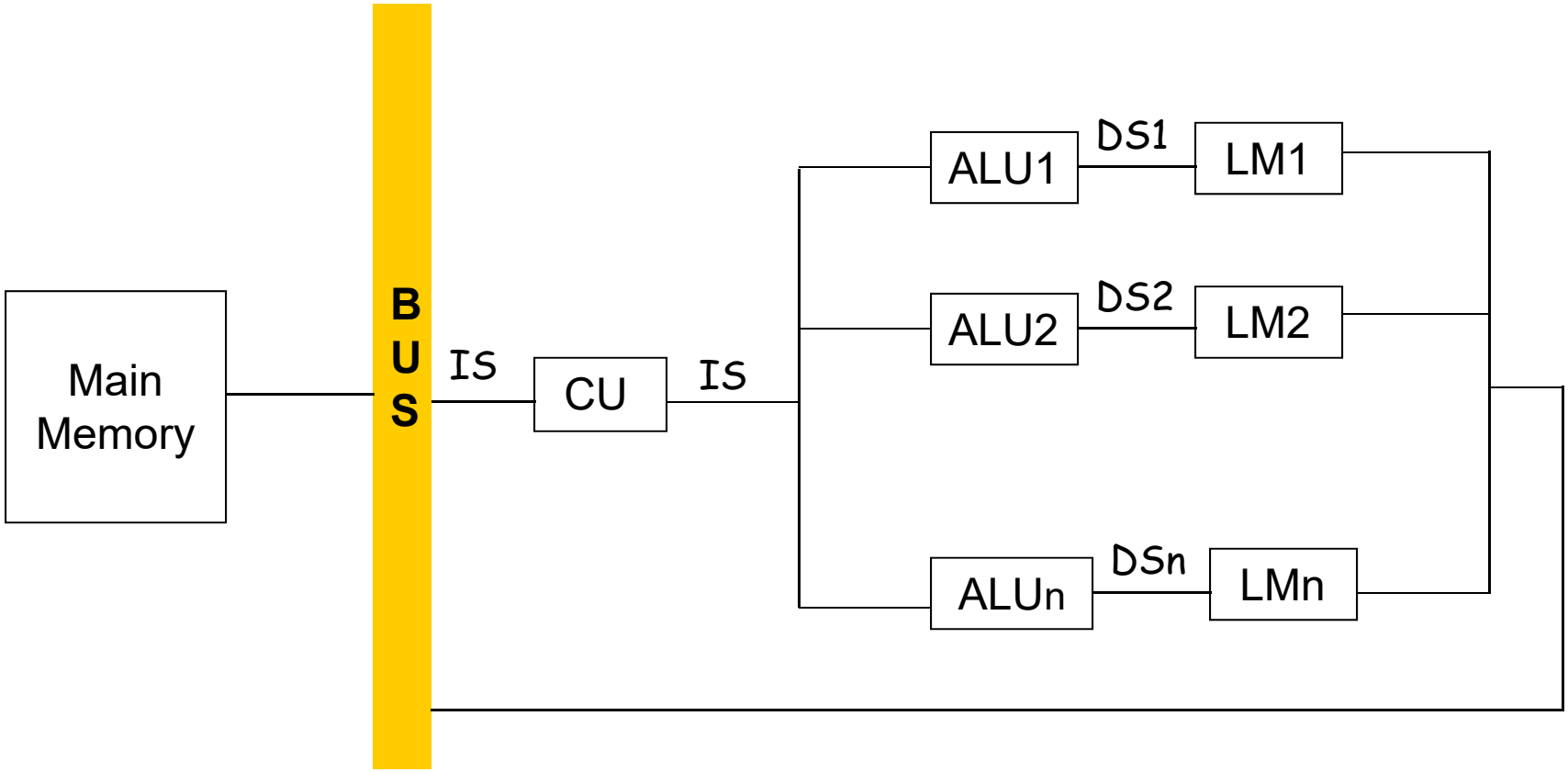
SIMD with shared memory cont…



SIMD with shared memory cont…

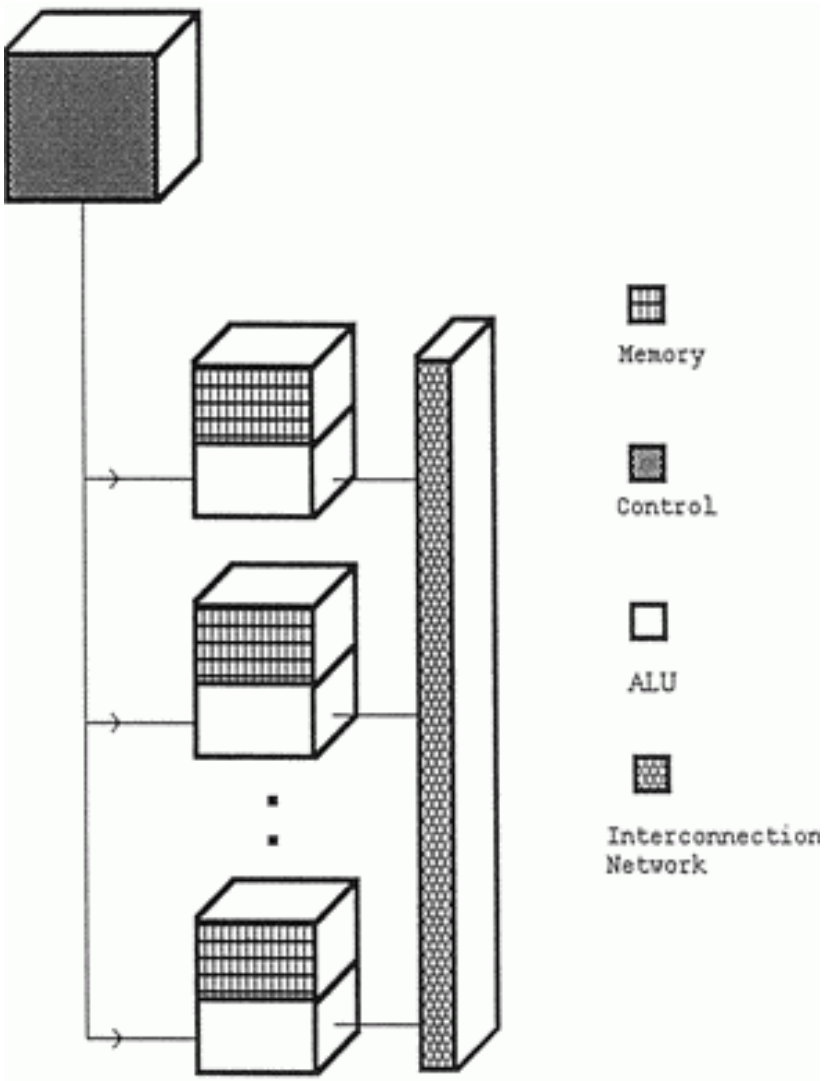


SIMD with distributed memory



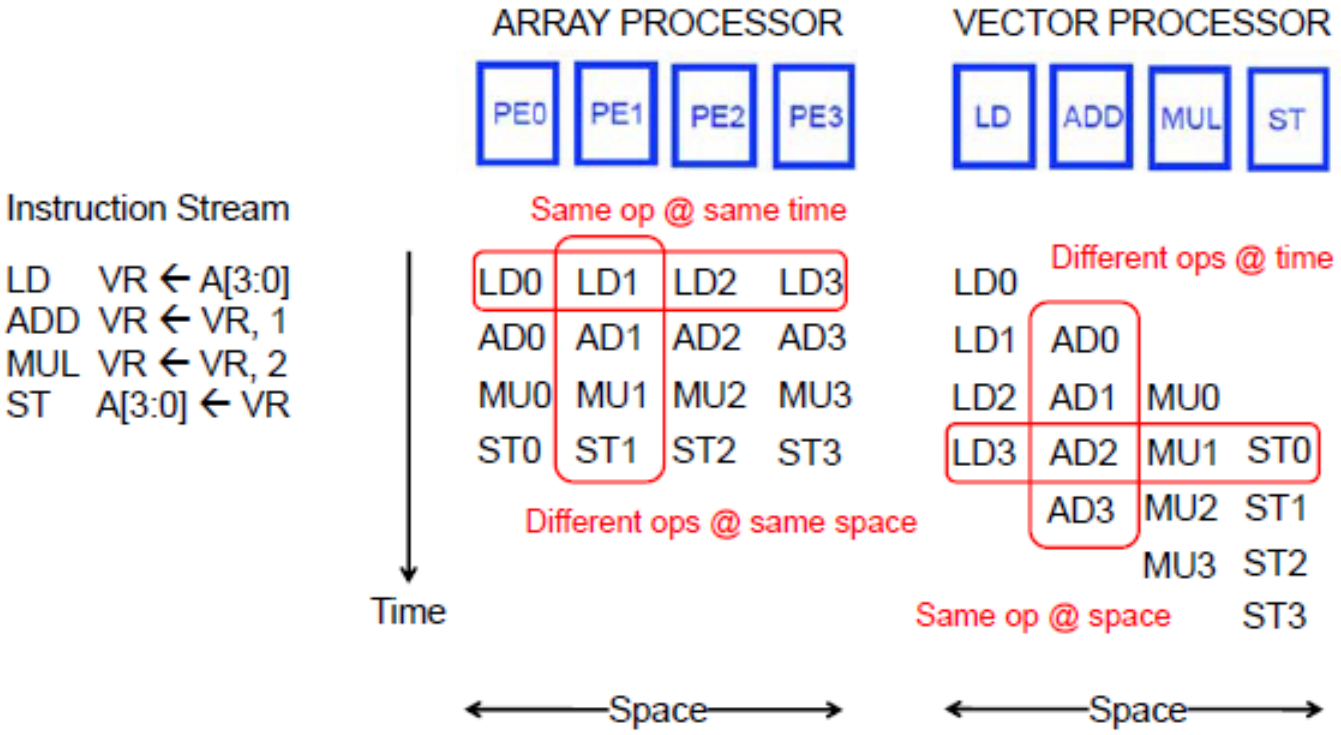
LM: Local Memory (cache)

SIMD with distributed memory cont…



SIMD cont…

- SIMD systems may be further subdivided based on Time-space duality into:
  - Array processing: Instruction operates on multiple data elements at the same time. The processors operate synchronously.
  - Vector processing: Instruction operates on multiple data elements in consecutive time steps. The processors may operate asynchronously or synchronously.

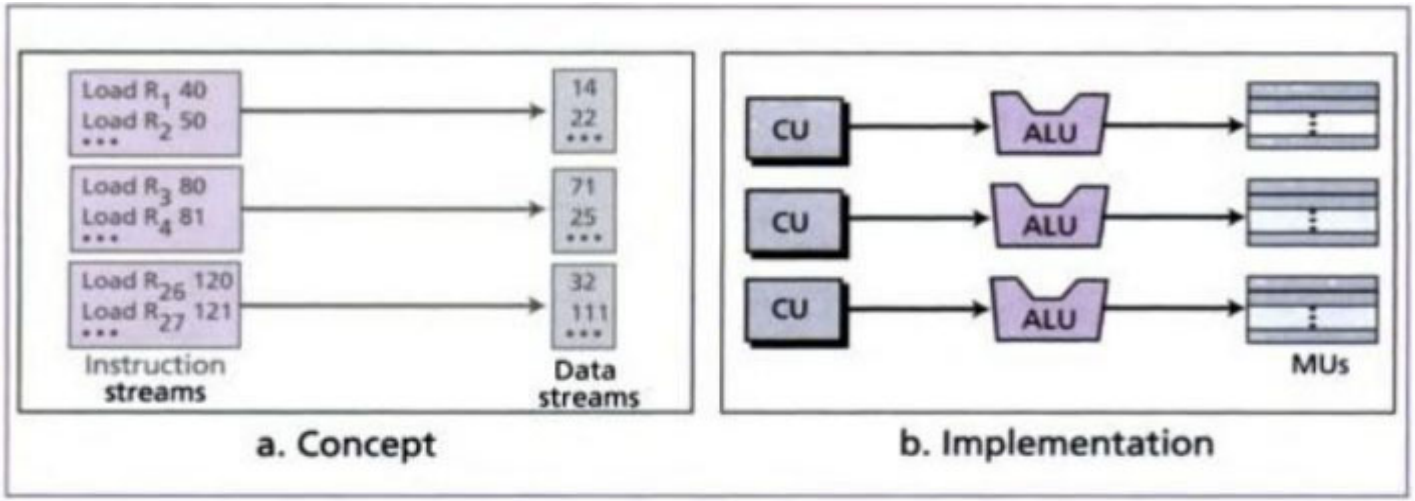


MIMD

- In this model, CPU consists of a collection of fully independent processing units each with its own control unit and its own ALU. Thus, each processing unit executes its own program.

- Each processor unit has a separate program and one instruction stream is generated from each program for each processor. Each instruction stream operates upon different data (superscalar).

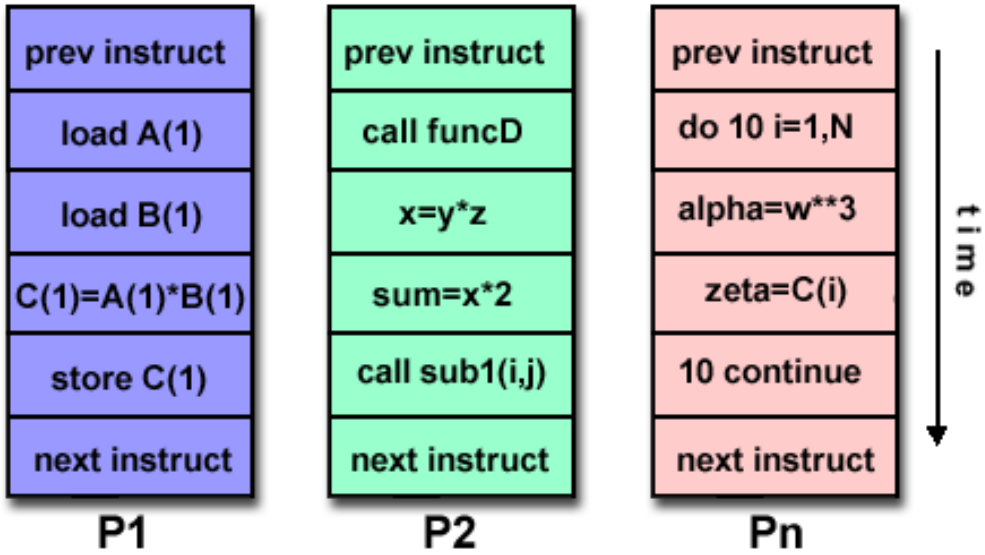
- Each processor operates under the control of an instruction stream issued by its own control unit. So, processors can operate asynchronously in general.



MIMD cont…

- Since it is general purpose, most of today's computers use this MIMD architecture.

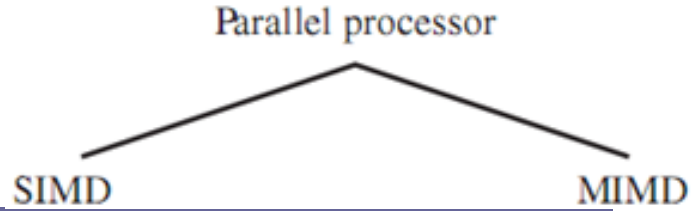
- SIMD computers requires less hardware than MIMD. However, since SIMD processors are specially designed, they tend to be expensive and have long design cycles. Not all applications are suited to SIMD processors.

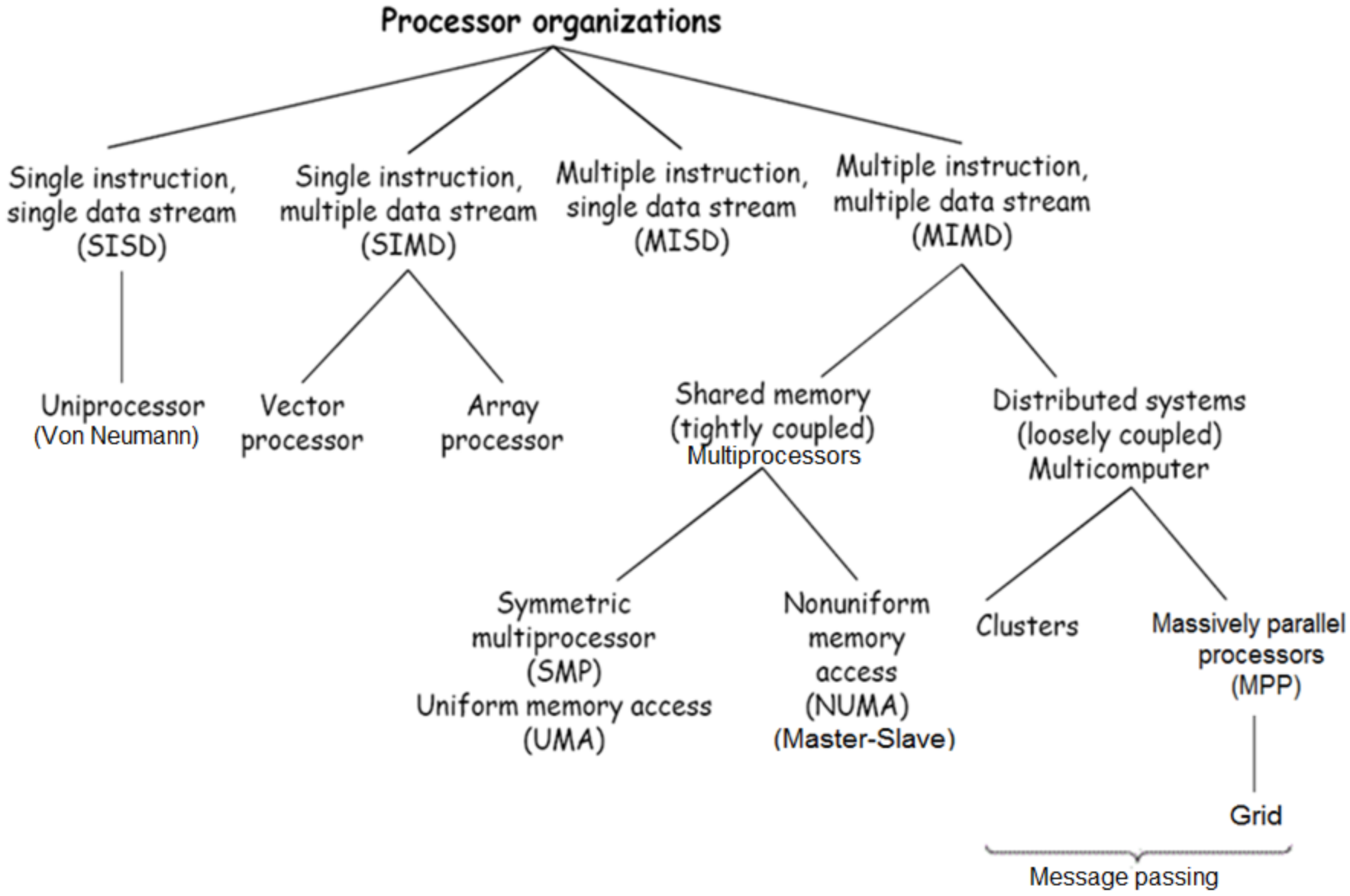


MIMD cont…

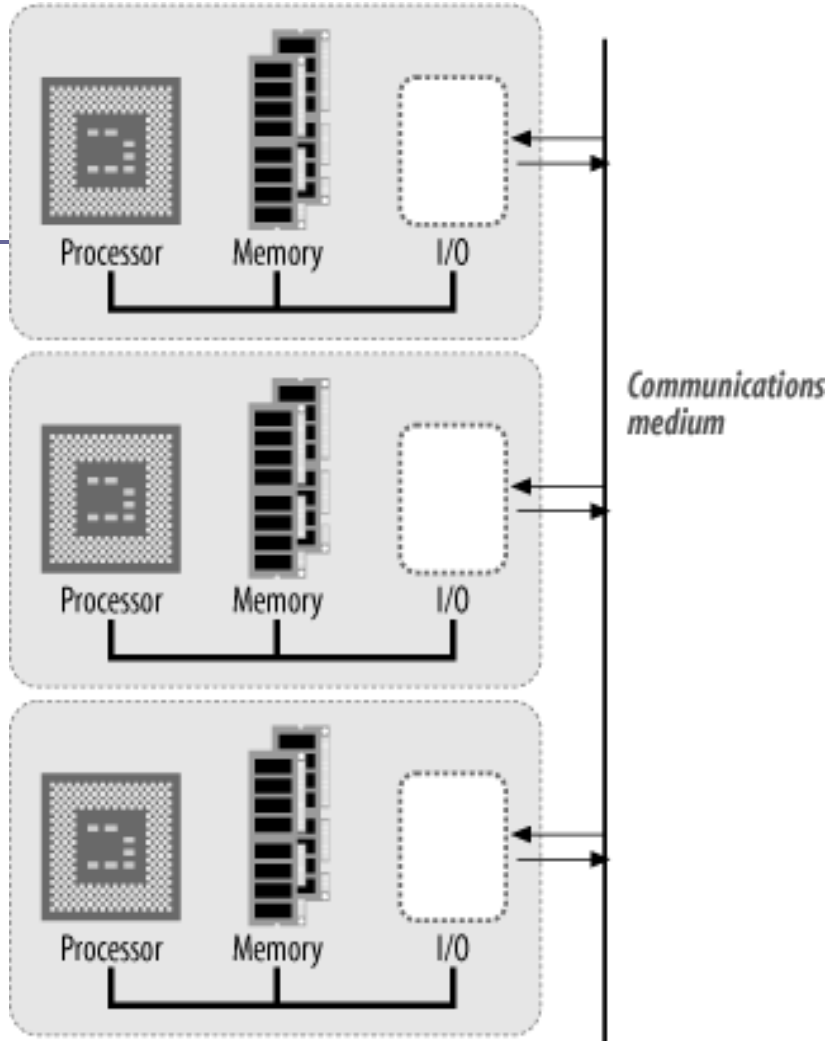
- There are two types of MIMD machines:
  - MIMD with shared memory/ Multi-processors/ tightly coupled/ Parallel machines: have a shared memory and can be divided in two categories based on how they access memory:
    - Centralized Shared-memory/Uniform memory access(UMA)/ Symmetric multiprocessors (SMP): all memory accesses take the same amount of time
    - Distributed Shared-memory/Non Uniform memory access(NUMA): time to access a remote memory is longer than the time to access a local memory
  - MIMD with distributed memory/Multi-computers/loosely coupled/ Distributed machines/ Cluster/message passing: large number of independent processing units each with its own memory that communicate via a dedicated network using message passing.
    - Cluster
    - MPP
      - Grid

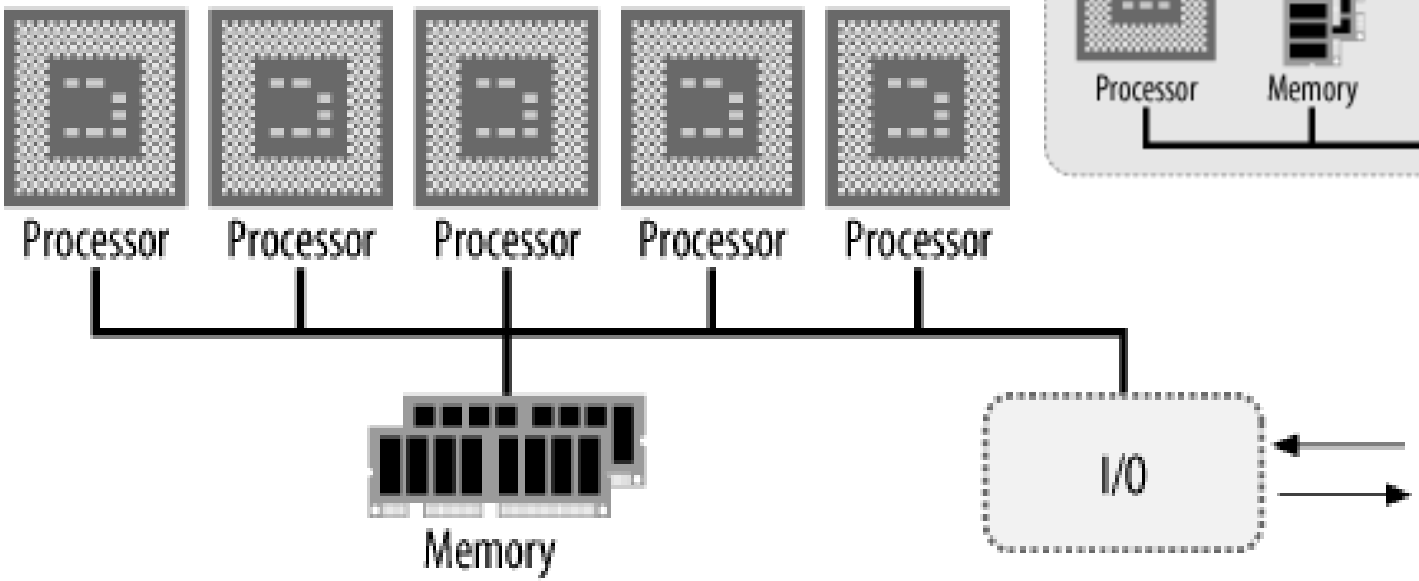
Processor Architectures





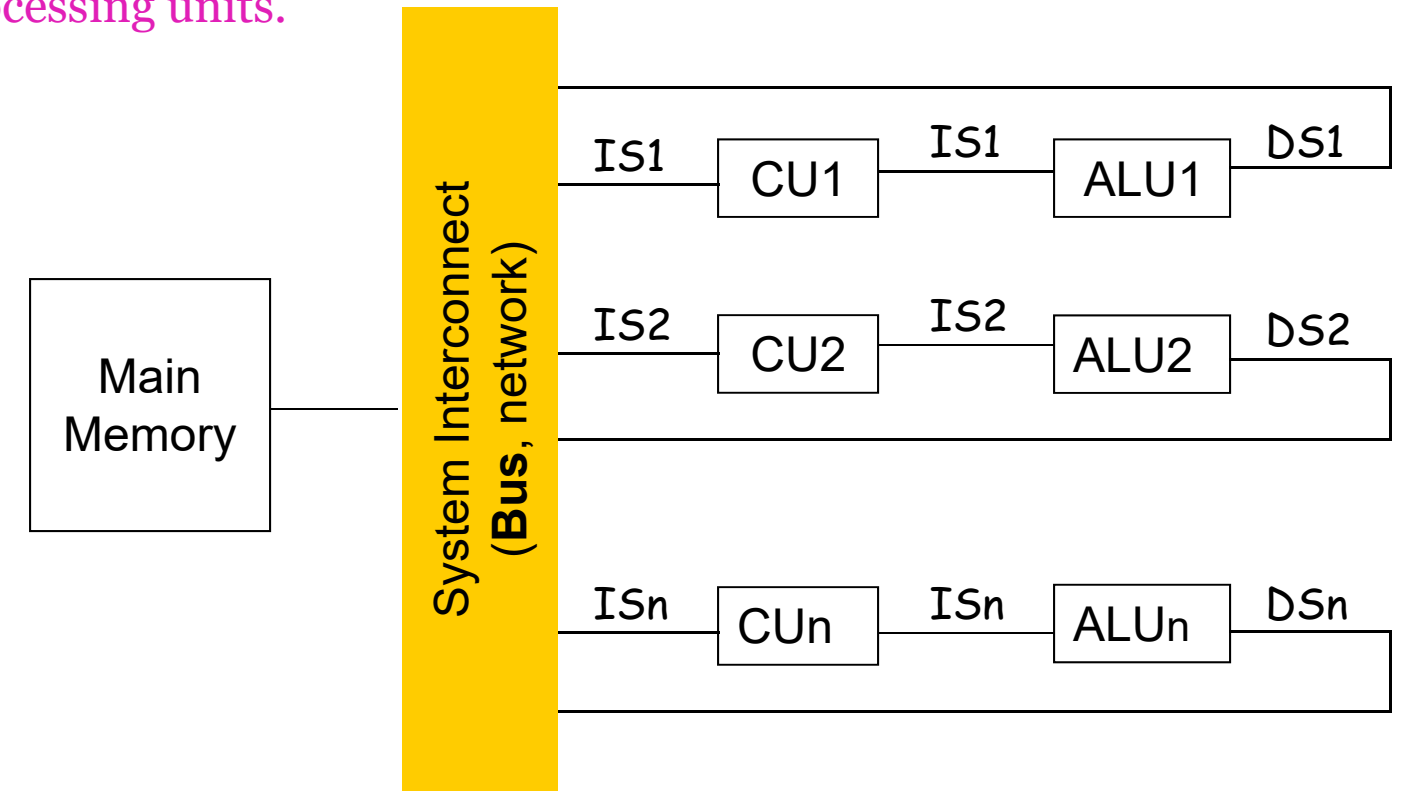
MIMD Shared vs. Distributed





MIMD: UMA

- Centralized shared-memory/Uniform memory access(UMA)/ Symmetric multiprocessors (SMP): the processors share a common memory and they communicate with each other via that shared memory.
- In UMA systems, all memory accesses take the same amount of time by the processing units.

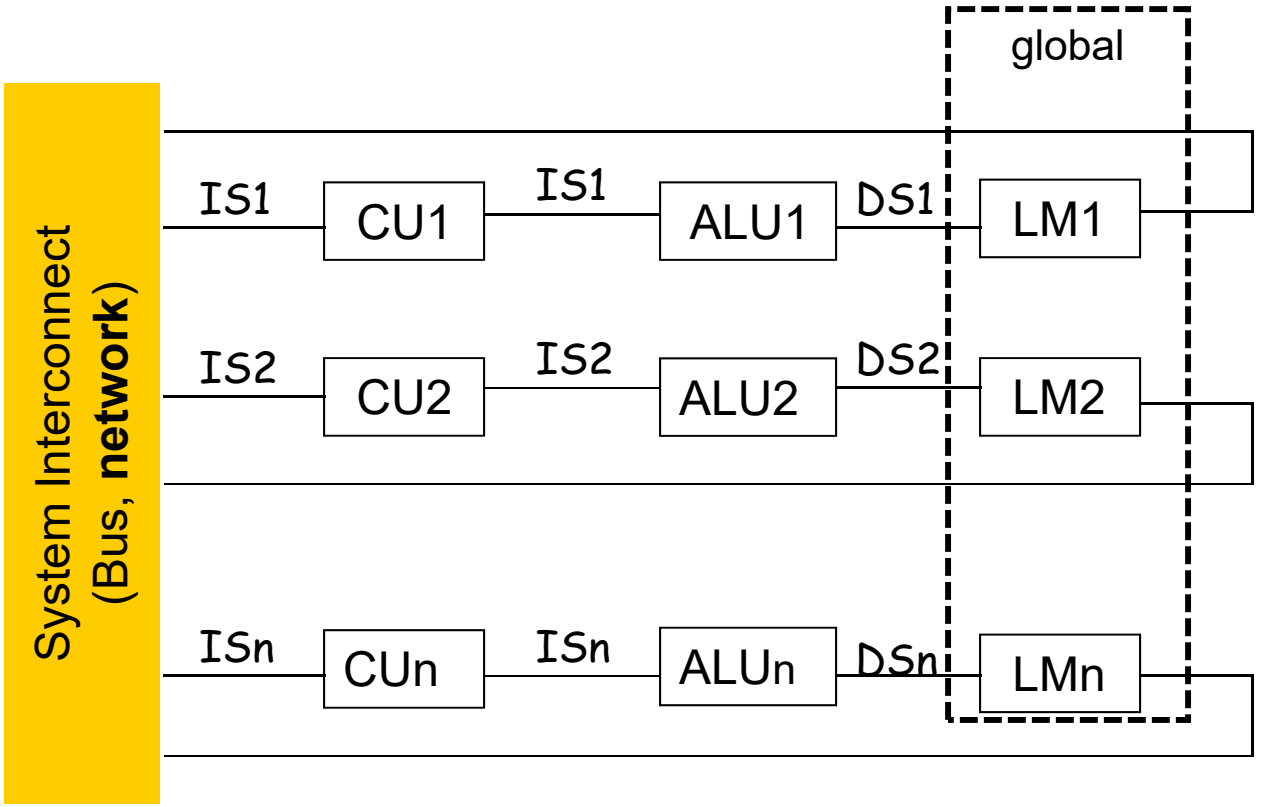


MIMD: NUMA

- Distributed Shared-memory/Non Uniform memory access(NUMA), each processing unit has its own local memory. A Processor can share its local memory with other processors (distributed shared memory) and the collection of all local memories create the global memory being shared.

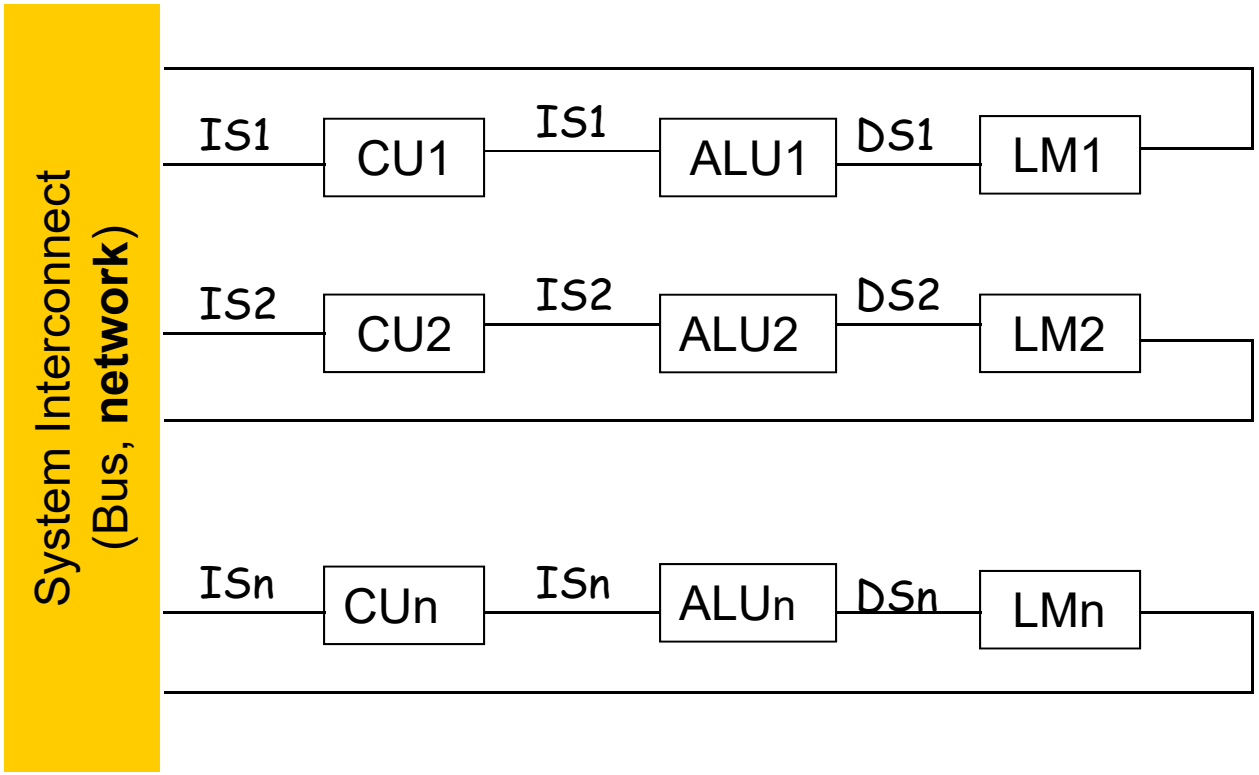
- In this way, global memory is distributed to all the processors **but** they see this memory as a contiguous entity. In this case, the access to a local memory is uniform for its corresponding processor that is attached to the local memory while it is not uniform for some other remote processors. Thus, all memory words are not accessed uniformly.

MIMD: NUMA cont…



MIMD with distributed memory

- A collection of independent Uni-processors are interconnected to form a cluster or grid.
- Communication among the computers is either via fixed paths or via some network facility.



Cluster

Clusters

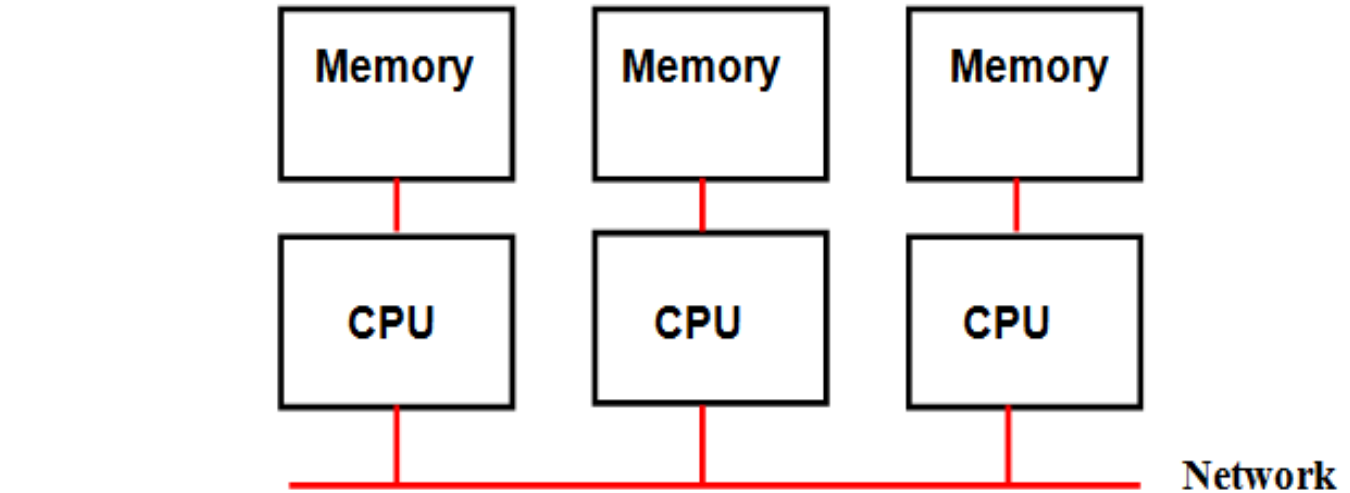
Loose vs. Tight coupling

- Coupling or dependency is the degree to which a system relies on each one of the other systems. Systems can be classified based on coupling in two ways:
  - Loosely coupled (اتصال ضعيف)
  - Tightly coupled (اتصال محكم)

- Shared memory systems are always tightly coupled.
- Distributed memory systems are always loosely coupled.

Loosely coupled

- Each processor has its own bus, memory, clock, and IO subsystem.
- Each processor communicates with other processors through the network medium (e.g., LAN, phone lines, WAN).
- Each processor runs its own independent local OS.

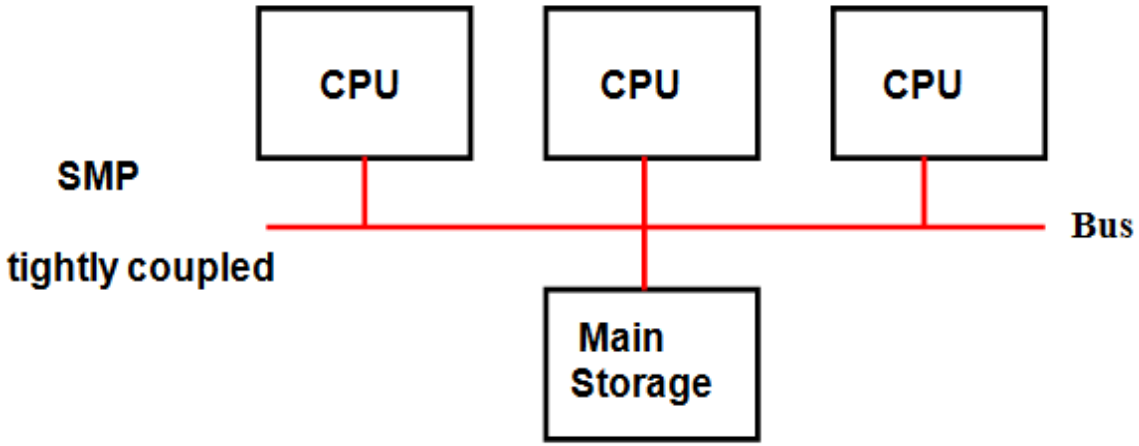


loosely coupled

Tightly coupled

tightly coupled = high bandwidth between processors
loosely coupled = low bandwidth between processors

- Processors share the clock, memory, bus, devices, and sometimes cache.
- Processors run a single instance of OS.
- Processors communicate frequently through a common memory.

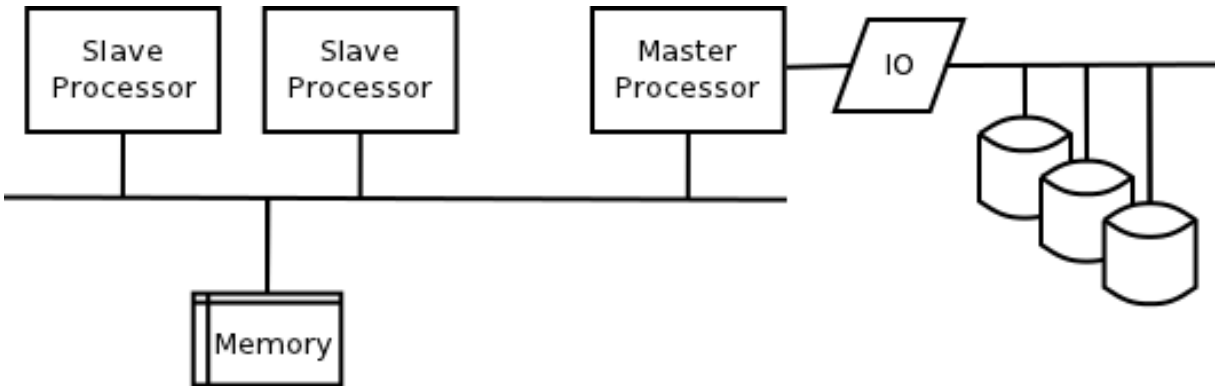


Tightly coupled cont…

- Tightly coupled systems can be classified into:
  - Symmetric
  - Asymmetric

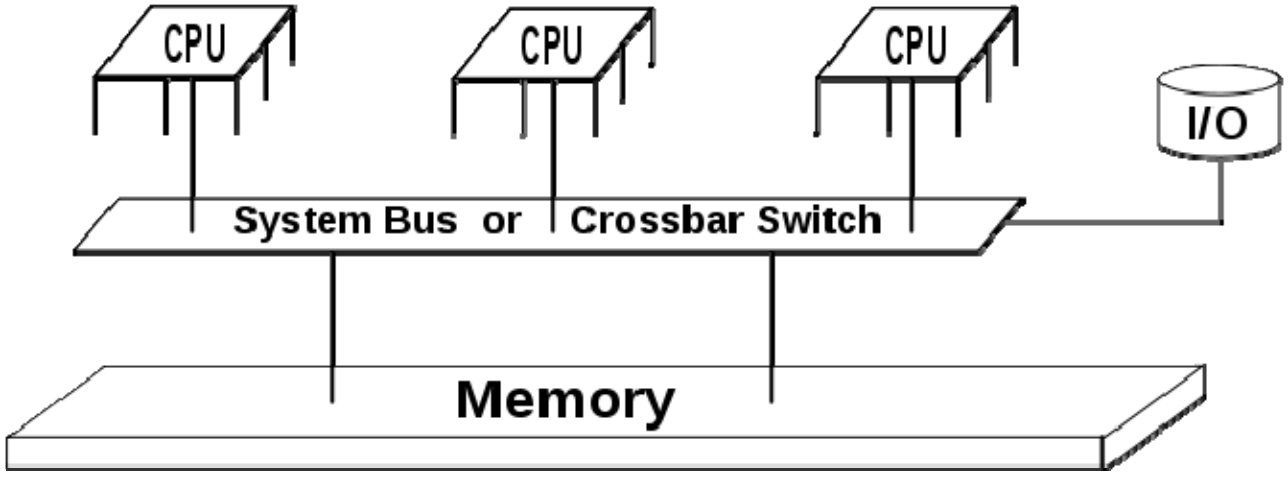
Asymmetric Multiprocessing

- Asymmetric Multiprocessing (ASMP/AMP) has one master CPU and the remainder CPUs are slaves.
- The master distributes tasks among the slaves, and I/O is usually done by the master only.
- The operating system typically sets aside one or more processors for its exclusive use. The remainders of the processors run user applications.
- As a result, the single processor running the operating system can fall behind the processors running user applications. This forces the applications to wait while the operating system catches up, which reduces the overall throughput of the system.
- if the processor that fails is an operating system processor, the whole computer can go down.

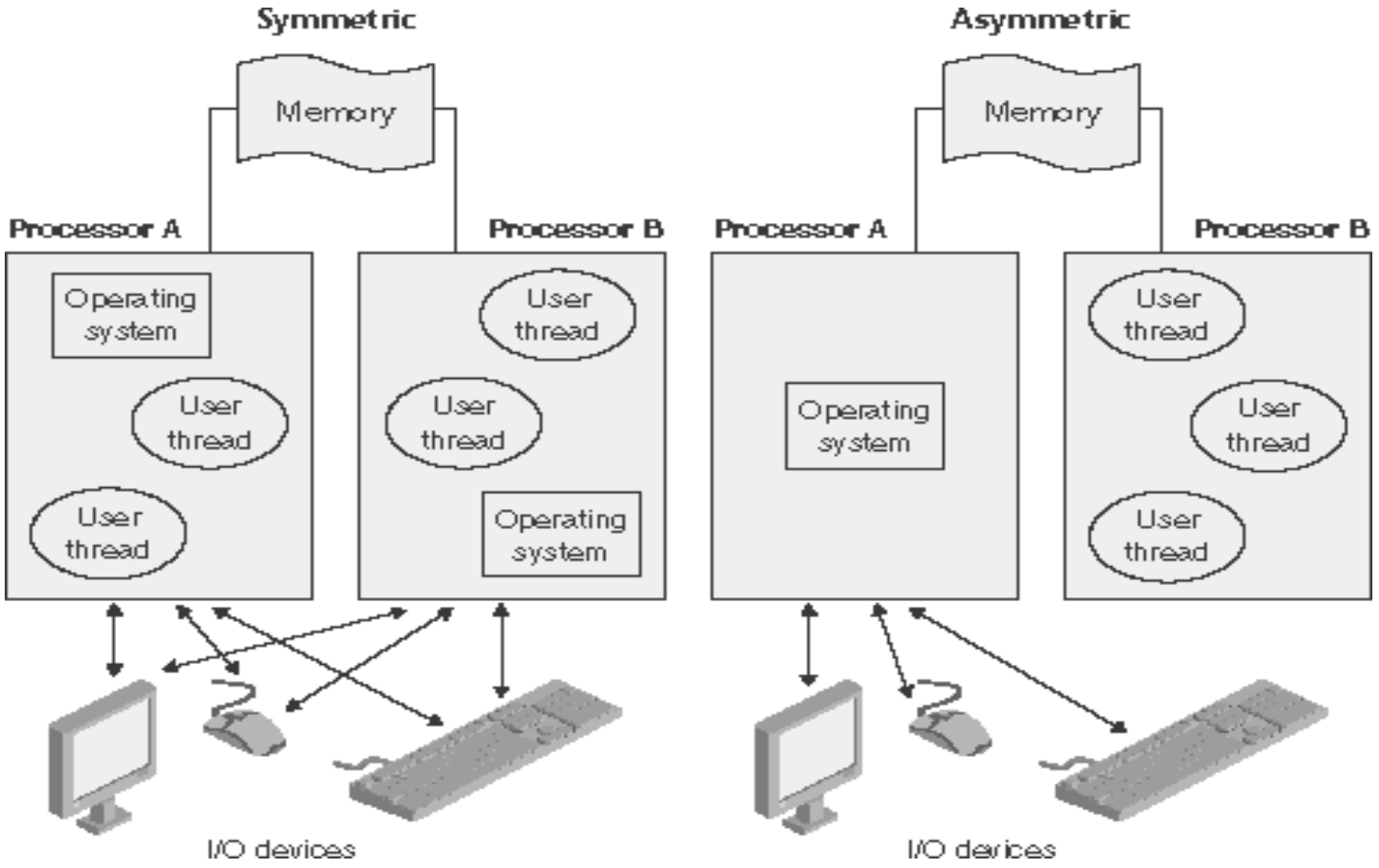


Symmetric Multiprocessing

- Symmetric multiprocessing (SMP) treats all processors as equals, and I/O can be processed on any CPU.
- Any processor can run any type of thread. Because the operating system threads can run on any processor, the chance of hitting a CPU bottleneck is greatly reduced.
- All processors are allowed to run a mixture of application and operating system code. A processor failure in the SMP model only reduces the computing capacity of the system.



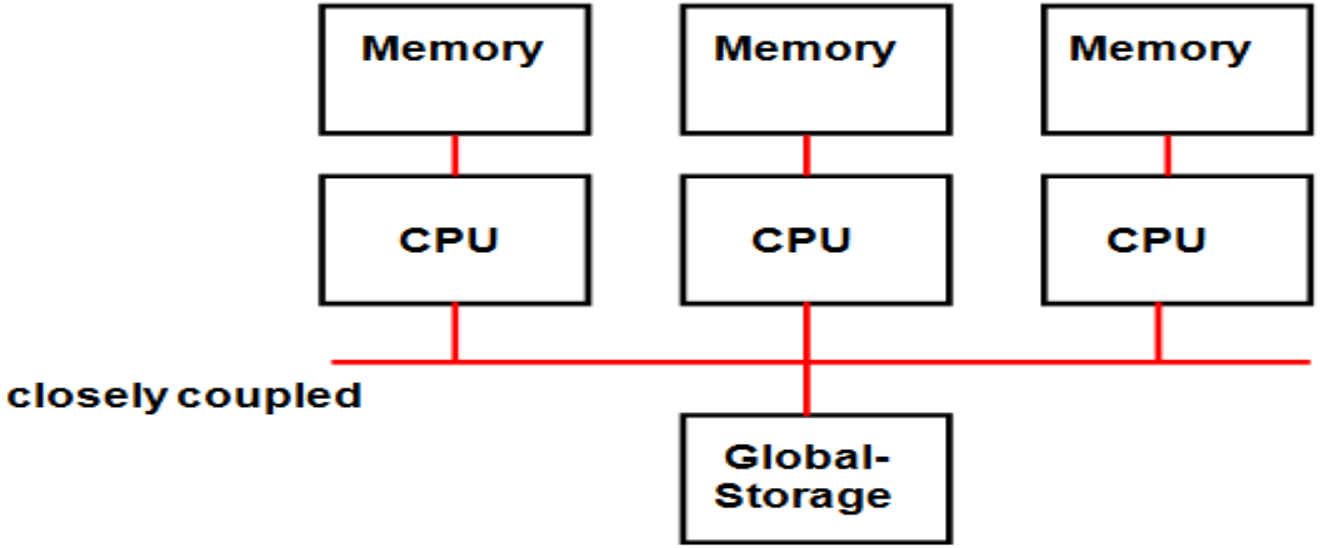
Tightly coupled cont…



each CPU in symmetric multiprocessing runs the same copy of the OS, while in asymmetric multiprocessing, they split responsibilities.

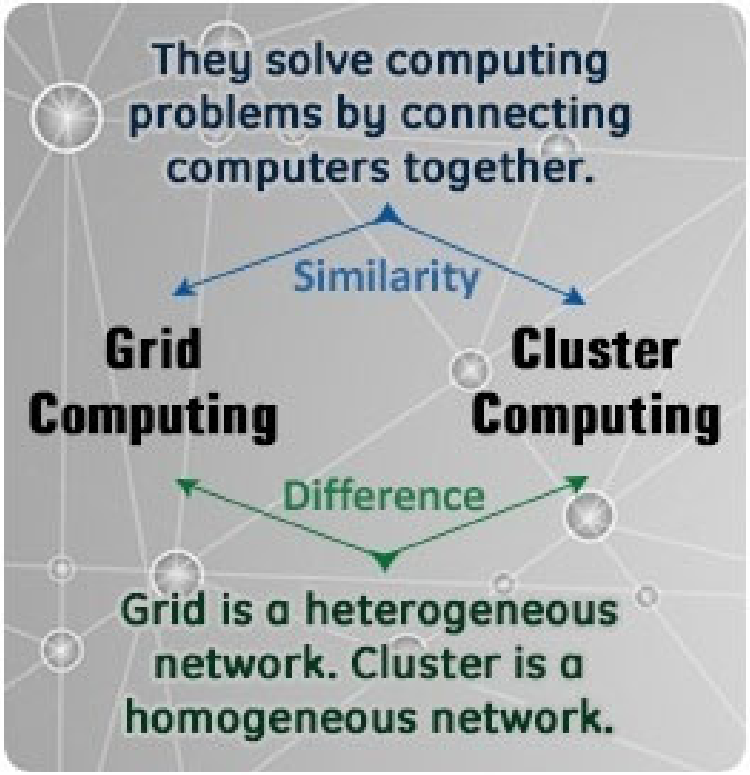
Closely coupled

- The closely coupled configuration is an extension of the loosely coupled configuration. It adds global storage, a storage unit that can be accessed by all processors. Global storage requires its own processor.
- An example of this system is the datacenters that have multiple supercomputers clustered together all working to run tasks like weather prediction or stock-market trend regression.



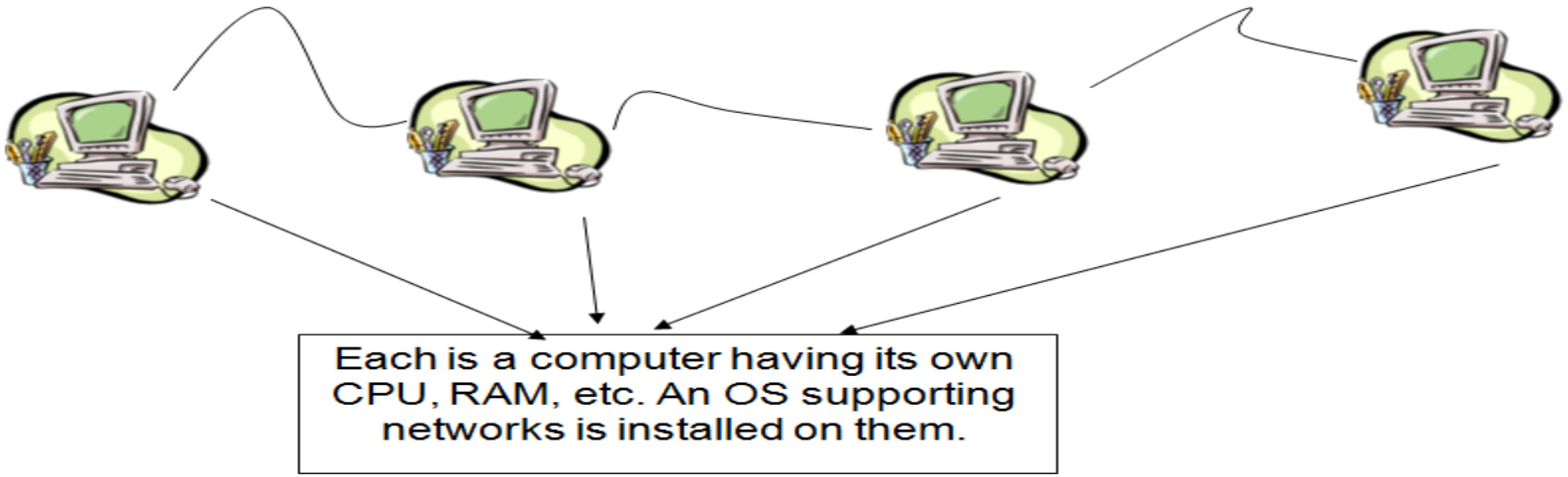
Distributed Processing

- Distributed system is a collection of many computers that are connected to each other via network.
- Distributed computing is when multiple systems are involved in performing a single computing task. The computing is divided among multiple computing systems to achieve the end result.
  - Grid computing
  - Cluster computing
  - Cloud computing



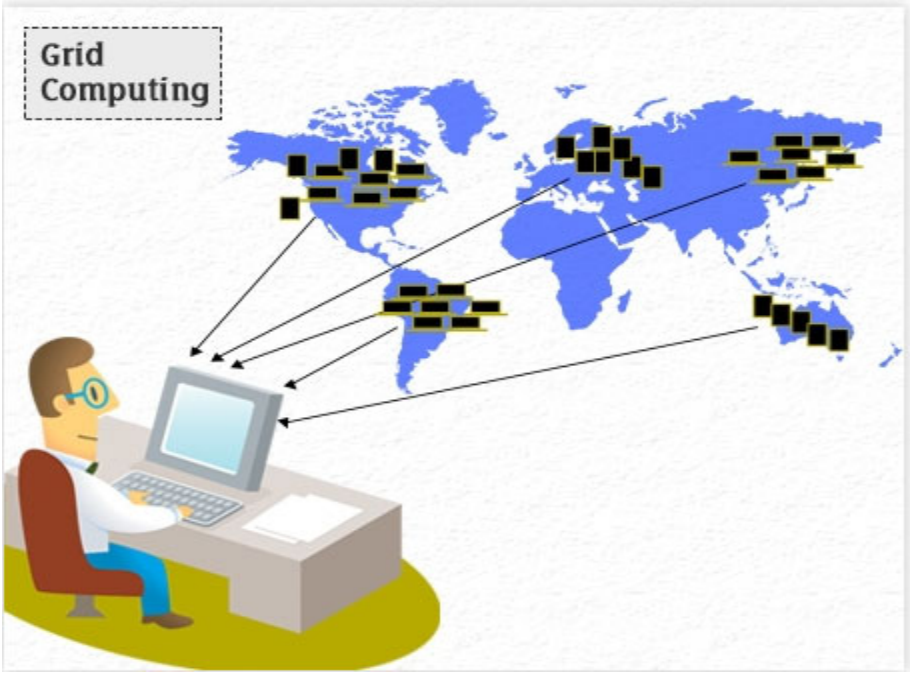
Grid computing

- Grid computing is a specialized implementation of distributed computing. It is a loosely coupled system.
- Grid Computing is a method of computer processing in which different parts of a program are run simultaneously on two or more autonomous (independent) computers that are communicating with each other over a network.
- Here, users are not aware of multiplicity of machines. Access to remote resources is similar to access to local resources.



Grid computing cont…

- Thousands of computers are employed in the process. All the devices that have computing power like desktops, laptops, tablets, mobiles, supercomputers, mainframes, servers, and meteorological sensors are connected together to form a single network. They are all connected together using the Internet. A software that is capable of dividing the program over many computers is used for this purpose. This entire infrastructure of connected computers is called a grid.



Grid computing cont…

- The term *grid computing* is frequently used to describe heterogeneous nodes distributed across the globe over a LAN, MAN, or WAN with different OS and different Hardware working together.
- Here individual user gets access to the resources (like processors, storage, data etc.) on demand with little or no knowledge of the fact that where those resources are physically located. For example, we use electricity for running air-conditioners, televisions etc. through wall sockets without concerned about the fact that from where that electricity is coming and how it is being generated.
- Every node in a grid behaves like an independent entity. This means it manages its resources by itself.

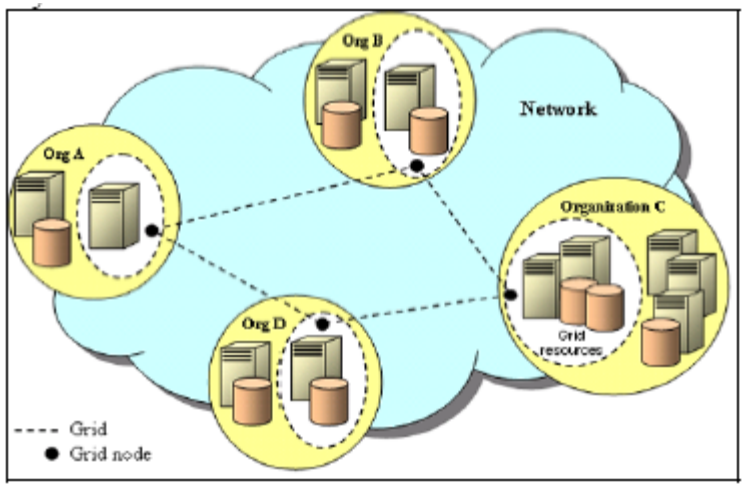


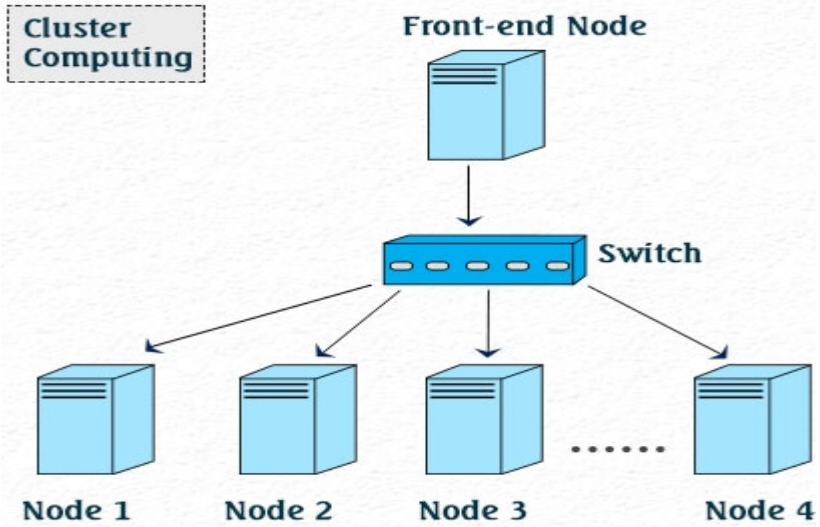
Figure 1. Grid Computing

Cluster computing

- Clustering is distributed computing, but all the resources are located in close proximity, i.e. within a single data center.
- Clustering is a closely coupled system.
- In order to be able to work together, multiple processors need to be able to share information with each other. This is accomplished using a shared-memory environment.

Cluster Computing cont…

- Clusters are generally restricted to computers on the same subnetwork or LAN. The aim is to combine the resources of several computers so that they function as a single unit.
- Cluster computing is a type of computing in which several nodes are made to run the same task as a single entity.
- The various homogeneous nodes involved in cluster have the same OS and same Hardware and the are normally connected to each other using a fast LAN.



Cluster Computing cont…

- **Unlike grids, the cluster from outside is seen as a single unit powerful system.**

- Therefore outsider client can not access individual systems in the cluster and can only give the program to the head cluster to use services. The head cluster receives the programs and distribute them between the existing systems and manage the performance of the parallel works and finally gives the results to the outsider client.

Cluster vs. grid: Differences

- Grid computing is something similar to cluster computing. The big difference is that a cluster is homogenous while grids are heterogeneous. The computers that are part of a grid can run different operating systems and have different hardware whereas the cluster computers all have the same hardware and OS.

- Another difference lies in the way resources are handled. In case of Cluster, the whole system (all nodes) behave like a single system view and resources are managed by centralized resource manager. In case of Grid, every node is autonomous i.e. it has its own resource manager and behaves like an independent entity.

Cluster vs. grid: Similarities

- Both the techniques involve solving computing problems that are not within the scope of a single computer by connecting computers together.

- These two techniques want to increase efficiency and throughput by networking of computers.

Multithreading Operating System

- Multi-threading is the ability of an operating system to execute the different parts of one program, called threads, simultaneously. In Multitasking, OS executes more than one program simultaneously, while in multithreading, OS executes different parts of a program simultaneously.