CHAPTER 6
ADVANCED TOPICS

Contents

- **Multiprocessing Systems**
  - **Real and Pseudo-Parallelism**
  - **Flynn’s Classification**
  - **Instruction Level, Thread Level and Process Level Parallelism**
  - **Inter-process Communication, Resource Allocation and Deadlock**
  - **Features of Typical Operating System**
- **Different Microprocessor Architectures**
  - **Register Based and Accumulator Based Architecture**
  - **RISC and CISC Architectures**
  - **Digital Signal Processors**

Multiprocessing Systems

- Traditionally, the computer has been viewed as a ***sequential machine***.
- Processor executes programs by executing machine instructions in a sequence and one at a time.
- At the micro operation level, multiple control signals are generated at the same time.
- ***Instruction pipelining*** and the overlapping of fetch & execute instructions from the same program in ***parallel***.
- Opportunities in parallelism, usually to enhance performance and availability.
- Multiprocessing is an example of parallelism which uses multiple CPUS sharing the common resources such as memory, storage device etc.

Characteristics of multiprocessing system

- An interconnection of two or more CPUs with memory and input-output equipment.
- All processors share access to common memory and are of similar comparable capability.
- Al processors share access to I/o devices either through the same channels that provide paths to the same devices.
- System is controlled by an integrated operating system that provides interaction between processors and their programs.

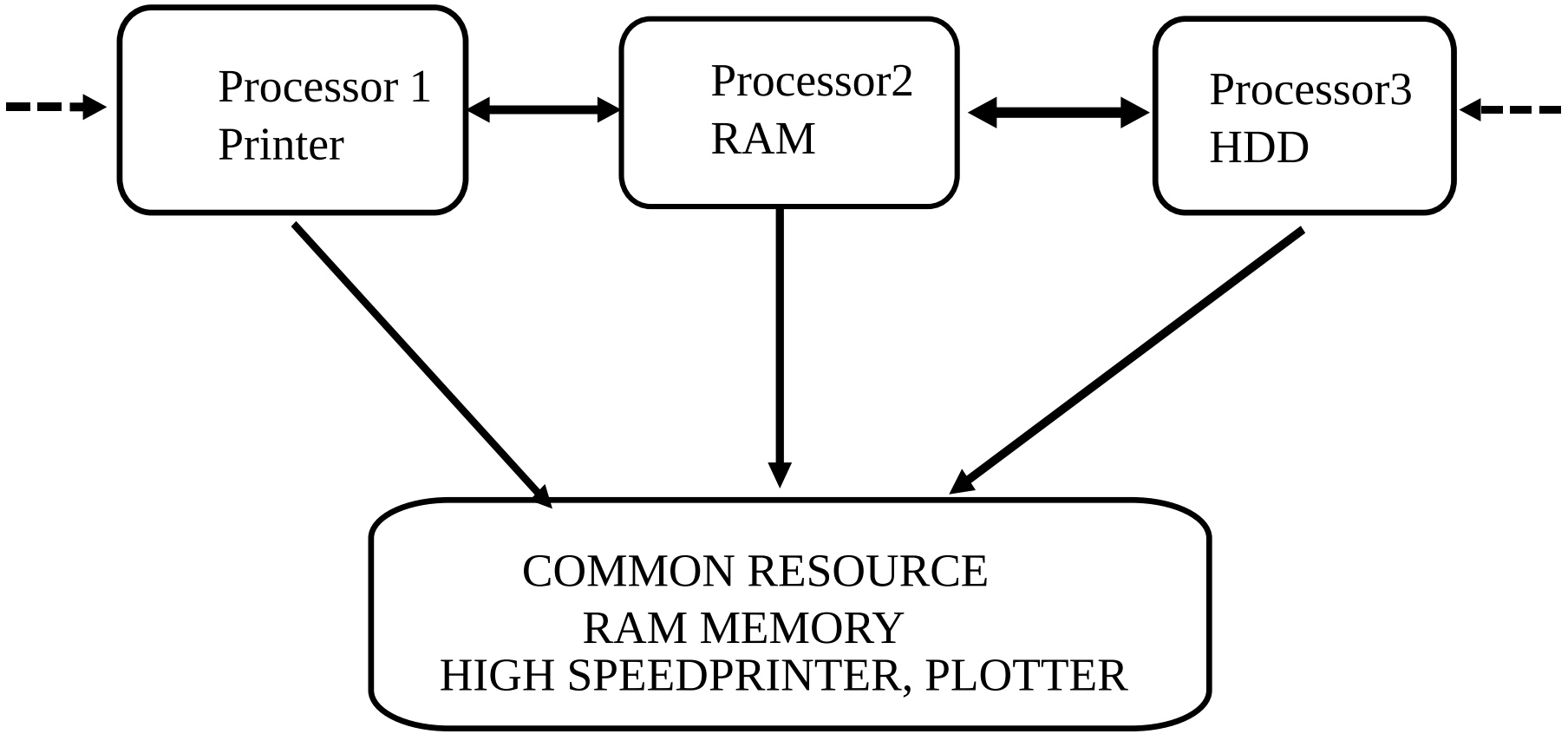


Fig: Organization of Multiprocessing System

- The organization of multiprocessor system can be divided into three types:
  1. **Time shared or common bus System:**
  2. **Multiport memory System:**
  3. **Central control unit System:**

A. Time shared or common bus System:

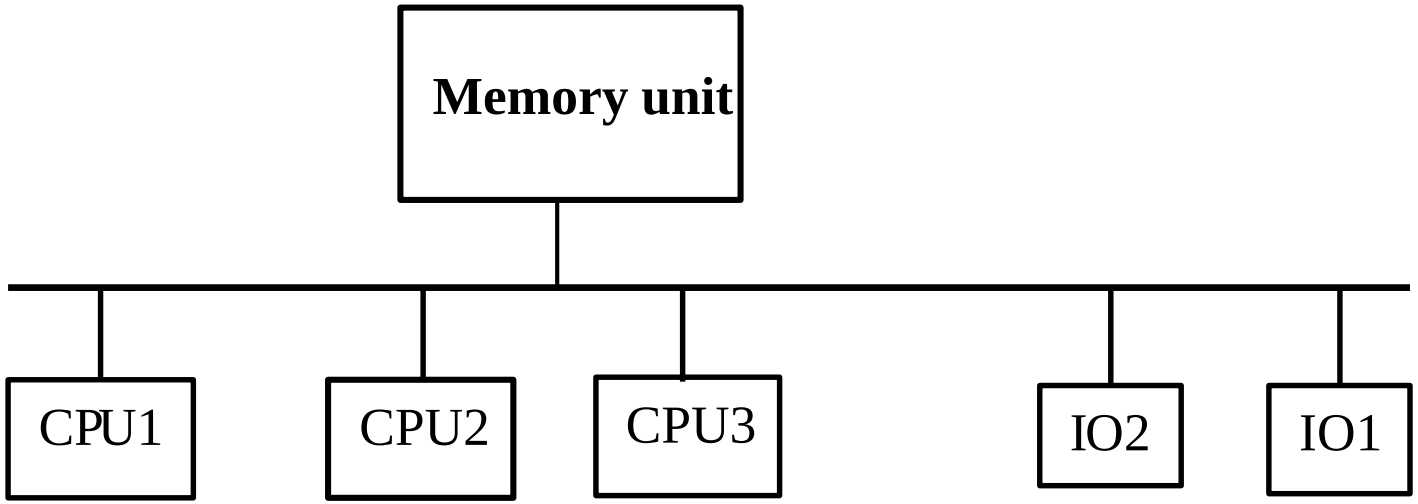


Fig: Time Shared System

- Numerous CPUs, I/O modules and memory connected to the same bus.
- The system must distinguish the source and the destination module for data transfer.
- Only one can be active at a time. i.e. only one device can master the bus temporarily.
- When one is active others should be locked out, events are divided on time basis.

Advantages:

1. **Simplicity:**
   - The physical interface and the addressing time sharing logic of each processor remains the same as in a single processor system, so it is very simplest approach.
2. **Flexibility:**
   - It is easy to expand the system by attaching more CPUs to the bus.
3. **Reliability:**
   - The failure of any attached device should not the failure of the whole system.

- **Drawback:**
  - The speed of the system is limited by the cycle time because all memory references must pass through the common bus.

2. Multiport memory System:

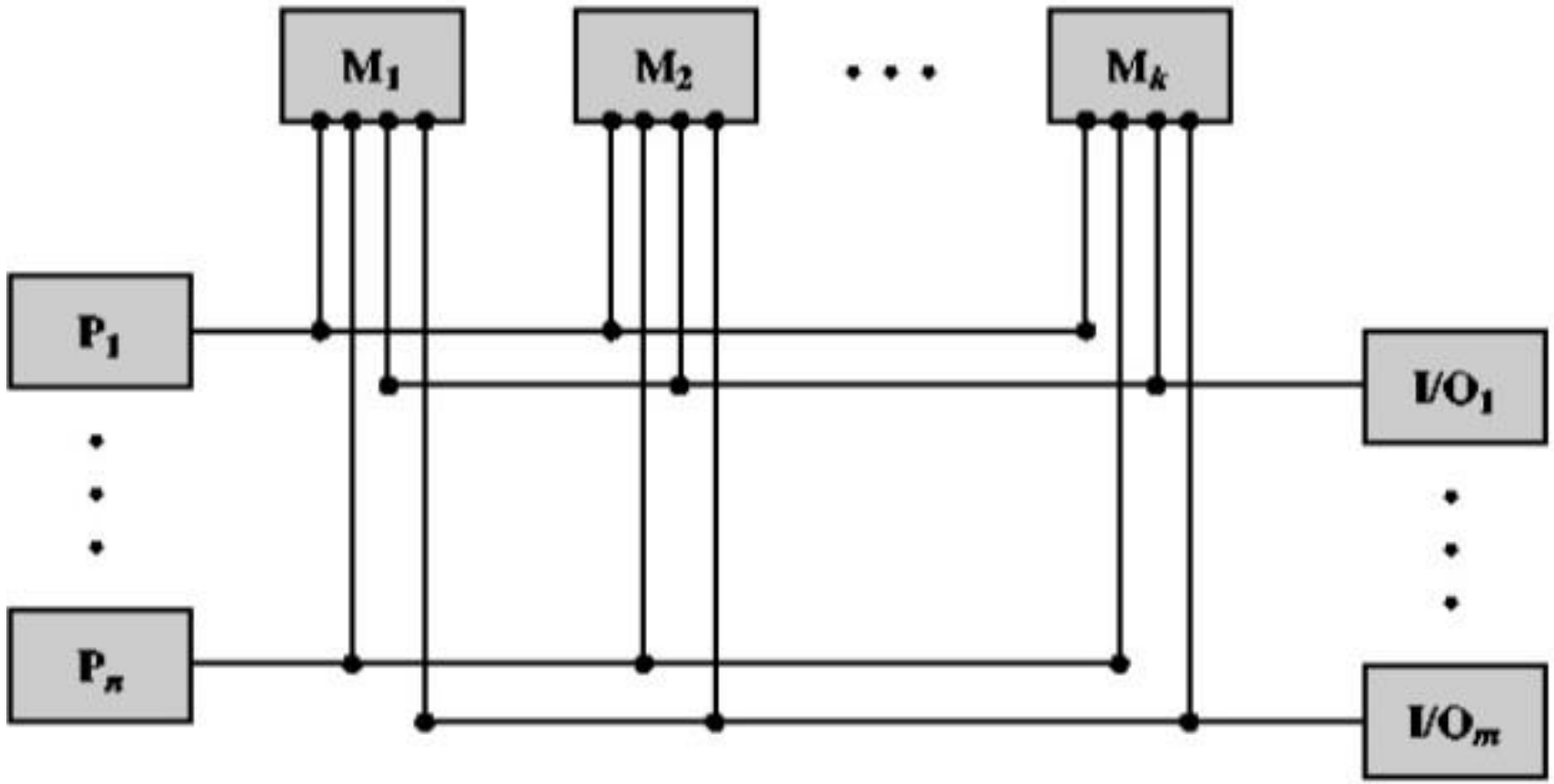


Fig: Time Shared System

- Each processor and /O module has dedicated path to each memory module.
- system has more performance and complexity than earlier one.
- For this system, it is possible to configure portions of memory as private to one or more CPUs and or I/O modules.
- This feature allows increasing security against unauthorized access and the storage of recovery routines in areas of memory not susceptible to modification by other processors.

C. Central control unit System:

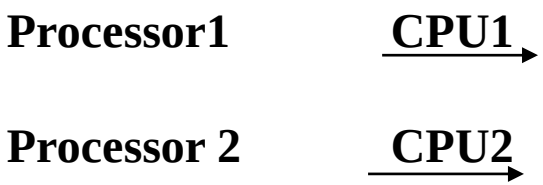
- It manages the transfer of separate data streams back and forth between independent modules like CPU, memory and I/O.
- The controller can buffer requests and perform arbitration and timing functions.
- It can also pass status and control messages between CPUS.
- All the co-ordination is concentrated in the central control unit un-disturbing the modules.
- It is more flexible and complex as well.

Real and Pseudo -Parallelism

- **Parallelism:**
  - simultaneous use of multiple computer resources to solve the computational problem.
  - Problem is broken into discrete parts that can be solved currently.
  - Each part is further broken into series of instructions which get executed simultaneously.
  - Two forms:
    1. Real parallelism
    2. Pseudo-parallelism

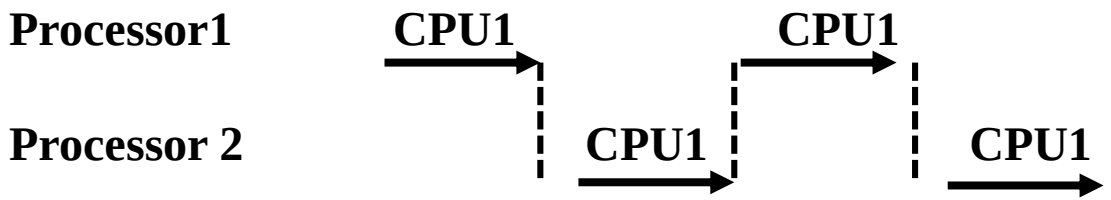
1. Real parallelism

- Real parallelism consists of the parallel modes of physical devices so that each can carry parallel operations to each other.
- Two or more processors operate at once.



2. Pseudo parallelism

- Consists of the same device carrying the parallel operation.
- Logically manage the parallelism for system.
- Concurrent processing using parallelism is the pseudo parallelism which operates either in time division or using other types of parallel algorithms.
- Two processors are switched and executed concurrently through single processor.



Flynn’s Classification

- There are different ways to classify parallel computers. One of the more widely used classifications, in use since ***1966***, is called ***Flynn's classification.***
- Flynn's classification distinguishes multi-processor computer architectures according to how they can be classified along the two independent dimensions of ***Instruction*** and ***Data***.
- Each of these dimensions can have only one of two possible states: ***Single*** or ***Multiple***.

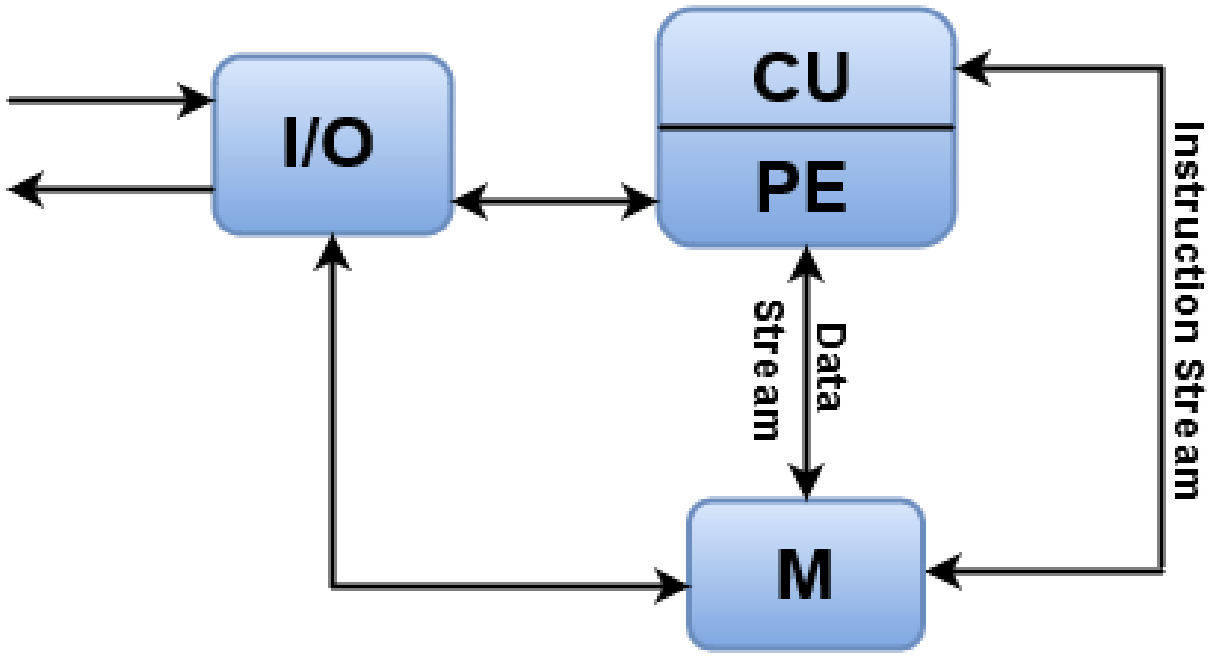
The matrix below defines the 4 possible classifications according to Flynn:

S I S D **Single Instruction, Single Data**	**S I M D** **Single Instruction, Multiple Data**
M I S D **Multiple Instruction, Single Data**	**M I M D** **Multiple Instruction, Multiple Data**

1. Single Instruction, Single Data (SISD):

- A serial (non-parallel) computer.
- **Single Instruction:** Only one instruction stream is being acted on by the CPU during any One clock cycle.
- **Single Data:** Only one data stream is being used as input during any one clock cycle.
- Single instruction is performed on a single set of data in a sequential form.
- Deterministic execution.
- This is the oldest and even today, the most common type of computer.
- **Examples:** Older generation mainframes, minicomputers and workstations; most modern day PCs.

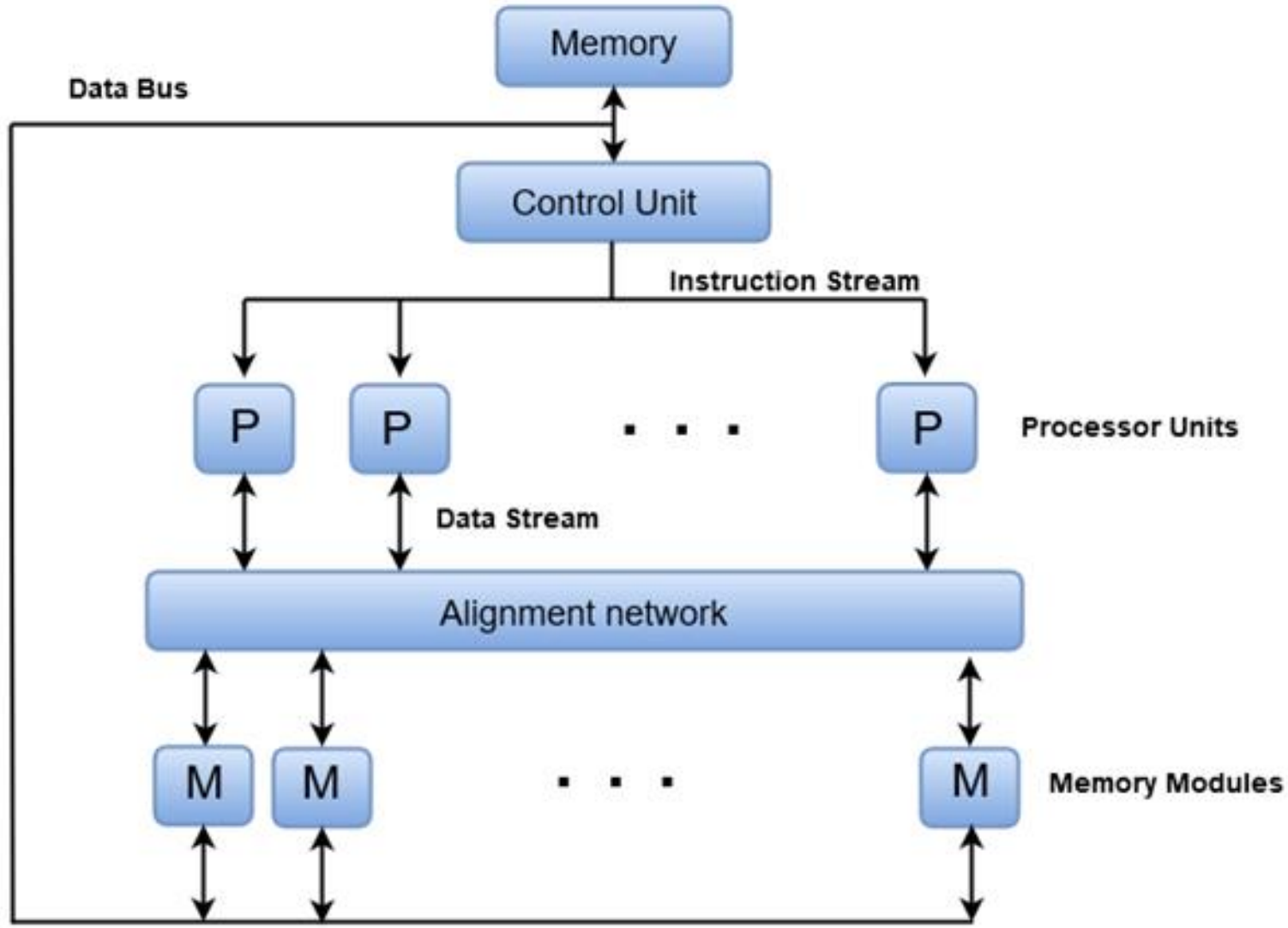
SISD:



2. Single Instruction, Multiple Data (SIMD):

- A type of parallel computer.
- **Single Instruction:** All processing units executes the same instruction at any given clock cycle.
- **Multiple data:** Each processing unit can operate on a different data element.
- Best suited for specialized problems characterized by a high degree of regularity, such as graphics/image processing.
- Single instruction is performed on multiple data. A good example is the *"For"* loop statement. Over here instruction is the same but the data stream is different.
- **Example:** ***Array processors:*** Connection Machine CM-2, MasPar MP-1 & MP-2, ILLIAC IV
- ***Vector Processors***: IBM 9000, Cray X-MP, Y-MP & C90, Fujitsu VP, NEC SX-2, Hitachi S820, ETA10

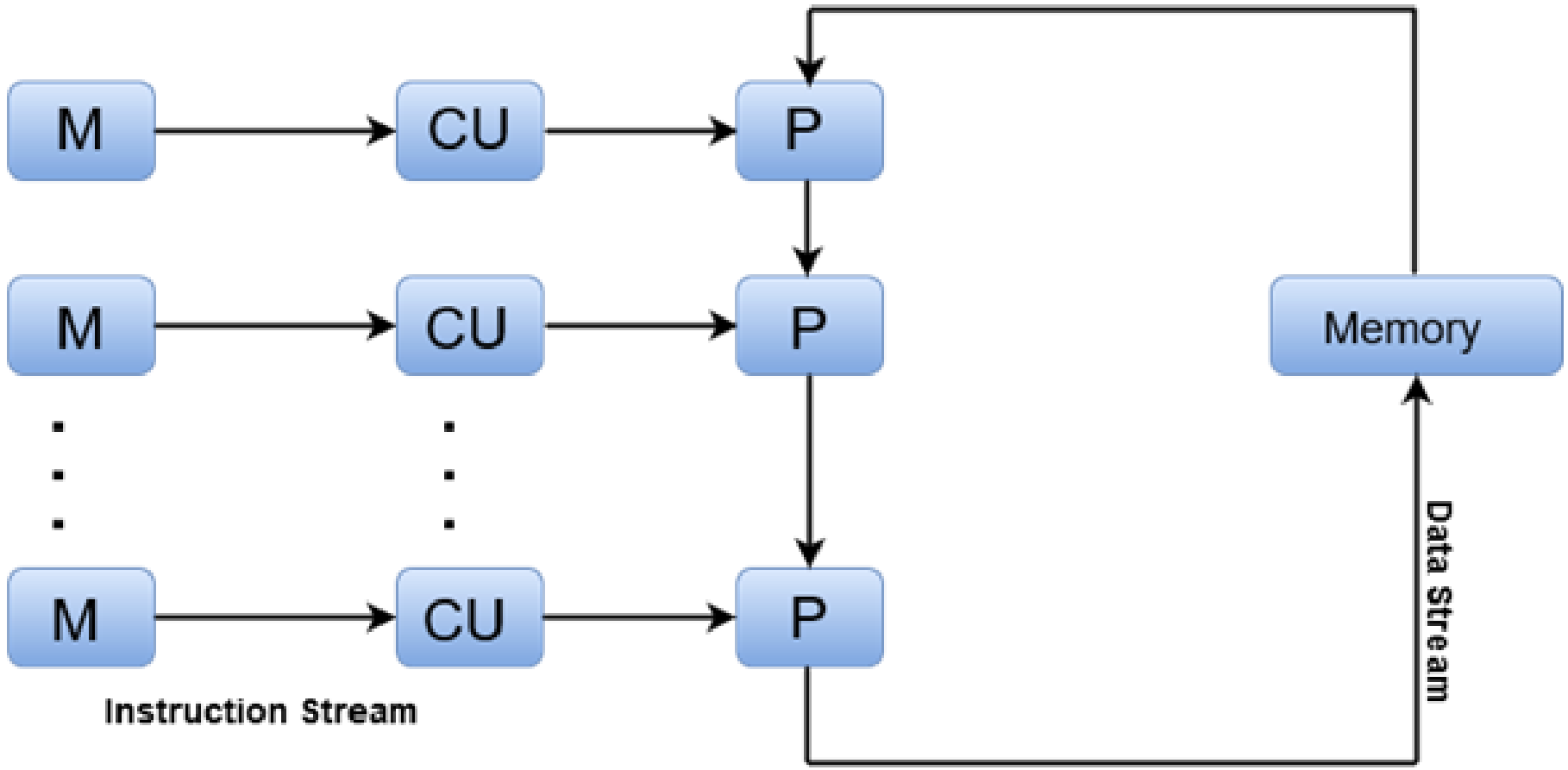
SIMD:


Memory
Data Bus
Control Unit
Instruction Stream
P
P
P
Processor Units
Data Stream
Alignment network
M
M
M
Memory Modules

3. Multiple Instruction, Single Data (MISD):

- A type of parallel computer.
- **Multiple Instruction:** Each processing unit operates on the data independently via separate instruction streams.
- **Single Data:** A single data stream is fed into multiple processing units.
- N numbers of processors are working on different set of instruction on the same set of data.
- Few actual examples of this class of parallel computer have ever existed. One is the experimental Carnegie-Mellon C.mmp computer (1971).
- Some conceivable uses might be:
  - multiple frequency filters operating on a single signal stream.
  - Multiple cryptography algorithms attempting to crack a single coded message.

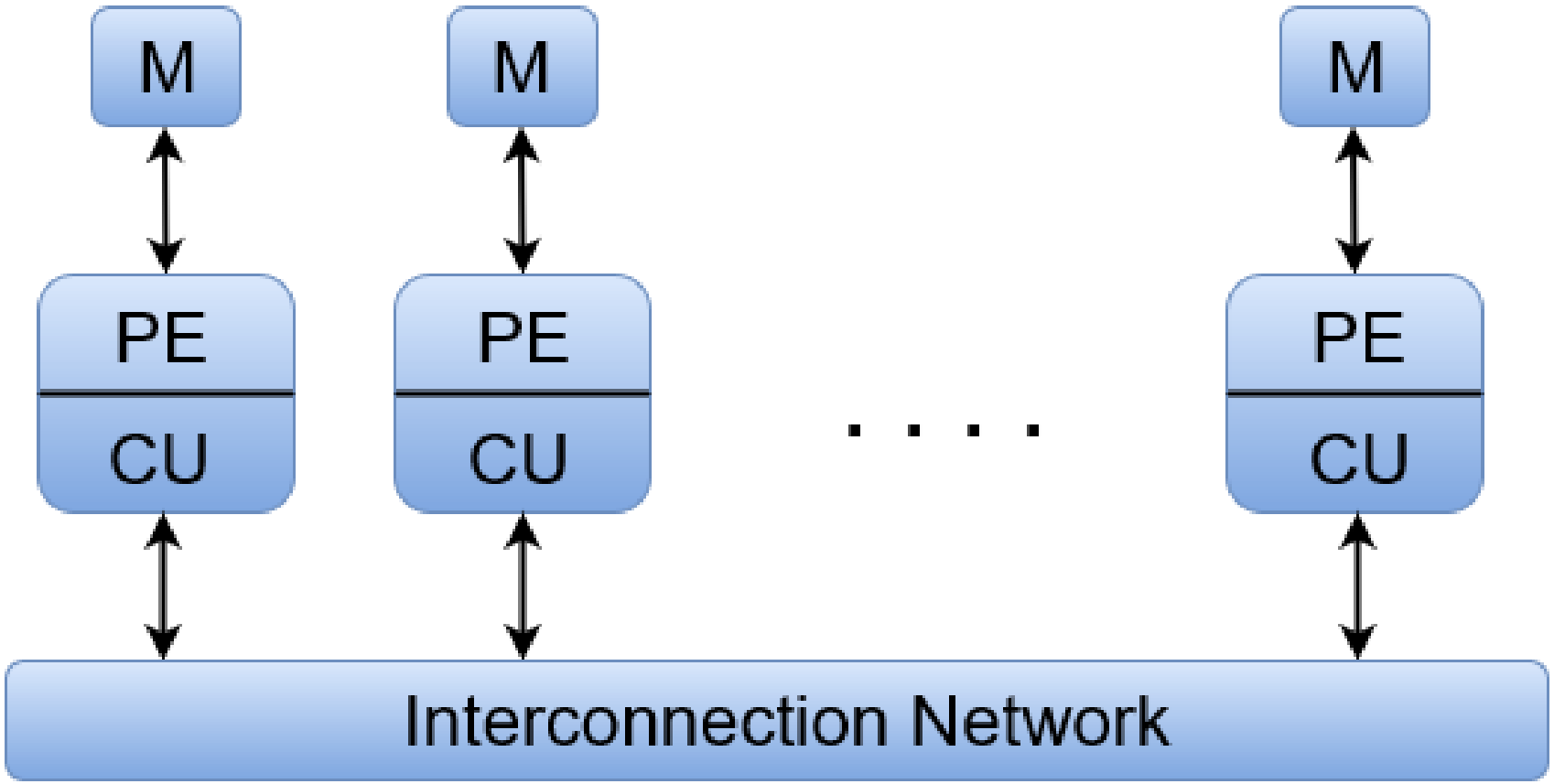
MISD:



4. Multiple Instruction, Multiple Data (MIMD):

- A type of parallel computer.
- **Multiple Instruction:** Every processor may be executing a different instruction stream.
- **Multiple Data:** Every processor may be working with a different data stream.
- There is an interaction of N numbers of processors on a same data stream shared by all processors.
- Execution can be synchronous or asynchronous, deterministic or non-deterministic.
- Currently, the most common type of parallel computer - most modern supercomputers fall into this category.
- **Examples:** Most current supercomputers, networked parallel computer clusters and "grids", multi-processor SMP computers, multi-core PCs.
- *Note: many MIMD architectures also include SIMD execution sub-components.*

MIMD:



Instruction Level, Thread Level and Process Level Parallelism

- **Instruction-level parallelism (ILP) :**
  - Is a measure of how many of the operations in a computer program can be performed simultaneously.
  - Consider the following program:
    1. e = a + b
    2. f = c + d
    3. g = e * f
  - Operation 3 depends on the results of operations 1 and 2, so it cannot be calculated until both of them are completed.
  - However, operations 1 and 2 do not depend on any other operation, so they can be calculated simultaneously.
  - If we assume that each operation can be completed in one unit of time then these three instructions can be completed in a total of two units of time, giving an ILP of 3/2.

- **Ordinary programs:**
  - Sequential execution model
  - Order specified by the programmer.
- **Ilp:**
  - Allows the compiler and the processor to overlap the execution of multiple instructions or even to change the order in which instructions are executed.
  - Amount is very large in graphics and scientific computing.
  - Cryptography exhibit much less parallelism.
- **Micro-architectural techniques that are used to exploit ILP include:**
  - **Instruction pipelining** where the execution of multiple instructions can be partially overlapped.
  - **Superscalar execution**, VLIW (very long instruction word), and the closely related Explicitly Parallel Instruction Computing concepts, in which multiple execution units are used to execute multiple instruction in parallel.

Thread Level Parallelism

- Also known as **task parallelism**, **function parallelism** and **control parallelism.**
- Is a form of parallelization of computer code across multiple processors in parallel computing environments.
- focuses on distributing execution processes (threads) across different parallel computing nodes.
- It contrasts to data parallelism as another form of parallelism.
- A single program might have several threads (or functions) that could be executed separately or in parallel.

- As a simple example, if we are running code on a 2-processor system (CPUs "a" & "b") in a parallel environment and we wish to do tasks "A" and "B" , it is possible to tell CPU "a" to do task "A" and CPU "b" to do task 'B" simultaneously, thereby reducing the run time of the execution.
- Most real programs fall somewhere on a continuum between Thread parallelism and Data parallelism.
- The pseudocode below illustrates task parallelism:
- Program:

```
if CPU="a" then
        do task "A"
else if CPU="b" then
        do task "B"
end if...
end program
```

- The goal of the program is to do some net total task ("A+B").
  - If we write the code as above and launch it on a 2-processor system, then the runtime environment will execute it as follows.
  - In an single program multiple data (SPMD) system, both CPUs will execute the code.
  - In a parallel environment, both will have access to the same data.
  - The "if" clause differentiates between the CPU's. CPU "a" will read true on the "if" and CPU "b" will read true on the "else if", thus having their own task.
  - Now, both CPU's execute separate code blocks simultaneously, performing different tasks simultaneously.

    •Code executed by CPU "a":

    …

    do task "A"

    …

    end program

    •Code executed by CPU "b":

    …

    do task "B"

    …

    end program

Data parallelism

- Is parallelism inherent in program loops,
- Focuses on distributing the data across different computing nodes to be processed in parallel.
- Many scientific and engineering applications exhibit data parallelism.
- A loop-carried dependency is the dependence of a loop iteration on the output of one or more previous iterations.
- Loop-carried dependencies prevent the parallelization of loops.

Process Level Parallelism

- Use of one or more central processing units (cpus) within a single computer system.
- Also refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them.
- Individual CPUs were subdivided into multiple "cores", each being a unique execution unit.
- CPUs with multiple cores are sometimes called "sockets". The result is a node with multiple CPUs, each containing multiple cores.
  - During the past 20+ years, the trends indicated by ever faster networks, distributed systems, and multi-processor computer architectures (even at the desktop level) clearly show that **parallelism is the future of computing**.
  - In this same time period, there has been a greater than 1000x increase in supercomputer performance, with no end currently in sight.

Inter-process Communication, Resource Allocation and Deadlock

- **Inter-process communication** (**IPC**):
  - is a set of methods for the exchange of data among multiple threads in one or more processes.
  - Processes may be running on one or more computers connected by a network.
  - IPC methods are divided into methods for **message passing**, **synchronization**, **shared memory**, and **remote procedure calls (RPC).**
  - The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data being communicated.

- There are several reasons for providing an environment that allows process cooperation:
  - Information sharing
  - Speedup
  - Modularity
  - Convenience
  - Privilege separation
- IPC may also be referred to as inter-thread communication and inter-application communication.

- The single operating system controls the use of system resources in a multiprocessing environment.
- The functions of multiprocessor operating system are:
  - An interface between users and machine
  - Resource management
  - Memory management
  - Prevent deadlocks
  - Abnormal program termination
  - Process scheduling
  - Managers security

Resource Allocation:

- Is necessary for any application to be run on the system.
- The computer requires to allocate certain resources for it to be able to run.
- Could be access to a section of the computer's memory, data in a device interface buffer, one or more files, or the required amount of processing power.
- Computers using single processors appear to be running multiple programs at once because the processor quickly alternates between programs, processing what is needed in very small amounts of time. This process is known as **multitasking or *time slicing***.
- The time allocation is automatic, however higher or lower priority may be given to certain processes, essentially giving high priority programs more/bigger **slices** of the processor's time.

Deadlock:

- A process requests resources; if the resources are not available at that time, the process enters a wait state.
- Waiting processes may never again change state, because the resources they have requested are held by other waiting processes. This situation is called a deadlock.
- Processes need access in reasonable order.
- Suppose a process holds resource A and requests resource B, at same time another process holds B and requests A; both are blocked and remain in deadlock.

Operating System

- Is a system software which control and manage the hardware through bios (basic i/o system).
- Bios is the part of operating system which is built in the system and it run application s/w on a hardware.
- It is an interface b/w user and system/s and is complex s/w.
- The operating systems key elements are:
  - A technical layer of software for driving software components.
  - A file system for organizing and accessing files logically.
  - Simple command language enabling users to run their own programs and manipulating files.

Features of OS

- System call
- Device drivers
- File system
- User interface

Characteristics:

- Memory management
- Process management
- Device management
- File management
- Security
- Job accounting
- Control over system performance
- Interaction with operators
- Error detecting
- Coordination between other software

Different Microprocessor Architectures

- **Accumulator Based Architecture** :
  - Accumulator is a most significant register then compared to other registers and most of the arithmetic and logic operations are performed using the accumulator performed via the accumulator.
  - The internal architecture of 8085 shows that the registers B, C, D, E, H and L are connected with the ALU through the accumulator and temporary register.
  - Data can only enter into the ALU from accumulator and the out pot of the ALU can be stared in accumulator through data bus.

Register Based Architecture

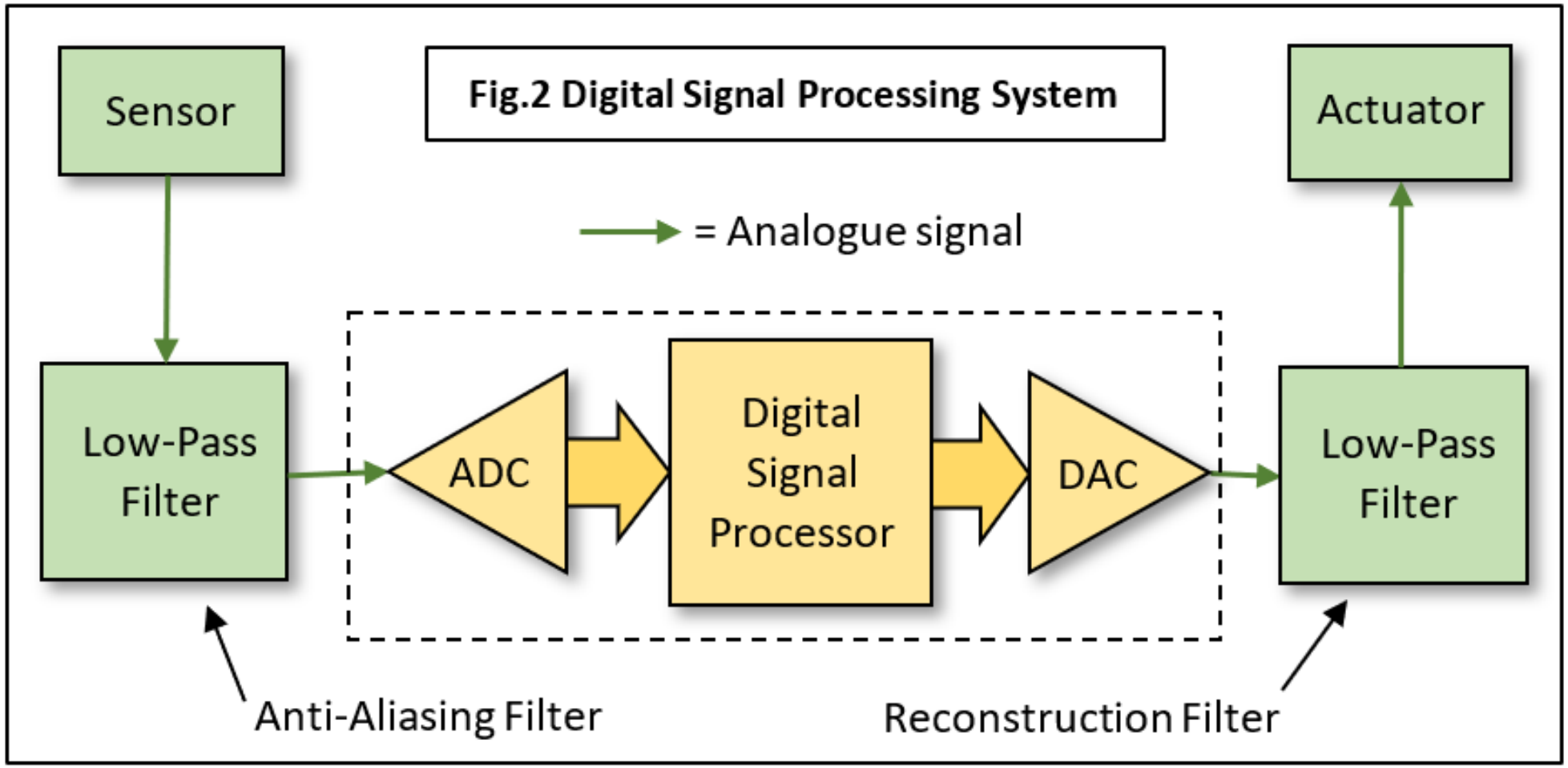
- All the arithmetic and logical operation consists of the operands of any registers (ABCD etc.).
- The input/output operations here register A and register B are similar.
- The internal architecture of 8086 shows that the registers A, B, C, D and others directly work with the ALU.
- Data can enter into the ALU from any registers.
- The advantage of register based architecture is that extendibility and flexibility in programming.
- The processor will be enhanced in register based architecture.
- The disadvantage is requirement of complex circuitry.

RISC and CISC Architectures

	RISC	CISC
Acronym	It stands for ‘Reduced Instruction Set Computer’.	It stands for ‘Complex Instruction Set Computer’.
Definition	The RISC processors have a smaller set of instructions with few addressing nodes.	The CISC processors have a larger set of instructions with more addressing nodes.
Memory unit	RISC has no memory unit and uses a separate hardware to implement instructions.	CISC has a memory unit to implement complex instructions.
Program	It has a hard-wired unit of programming.	It has a micro-programming unit.
Design	It is a complex complier design.	It is easier complier design.
Calculations	The calculations are faster and precise.	The calculations are slow and precise.
Decoding	Decoding of instructions is simple.	Decoding is complex.
Time	Very less execution time.	Very high execution time.
External memory	It does not require external memory for calculations.	It requires external memory for calculations.
Pipelining	Pipelining does function correctly.	Pipelining does not function correctly.
Code expansion	Can be a problem.	Is not a problem.
Applications	Used in high end applications such as video processing, telecommunications and image processing.	Used in low end applications such as security systems, home automations, etc.

Digital Signal Processors

- The real time signals such as pressure, temperature, voice is continuous time varying signals are known as analog signals.
- The process of conversion of the analog signals into digital signal, which reduces the redundancy and make more immune to noise is called **digital signal processing.**
- The digital signal processors take input the digital data for that every analog data is to be converted into digital by using A/D converter.


Sensor
Fig.2 Digital Signal Processing System
Actuator
= Analogue signal
Low-Pass Filter
ADC
Digital Signal Processor
DAC
Low-Pass Filter
Anti-Aliasing Filter
Reconstruction Filter

DSP applications

- Telecommunications
- Music processing
- Speech generation and recognition
- Radar
- Image processing
- In digital filters