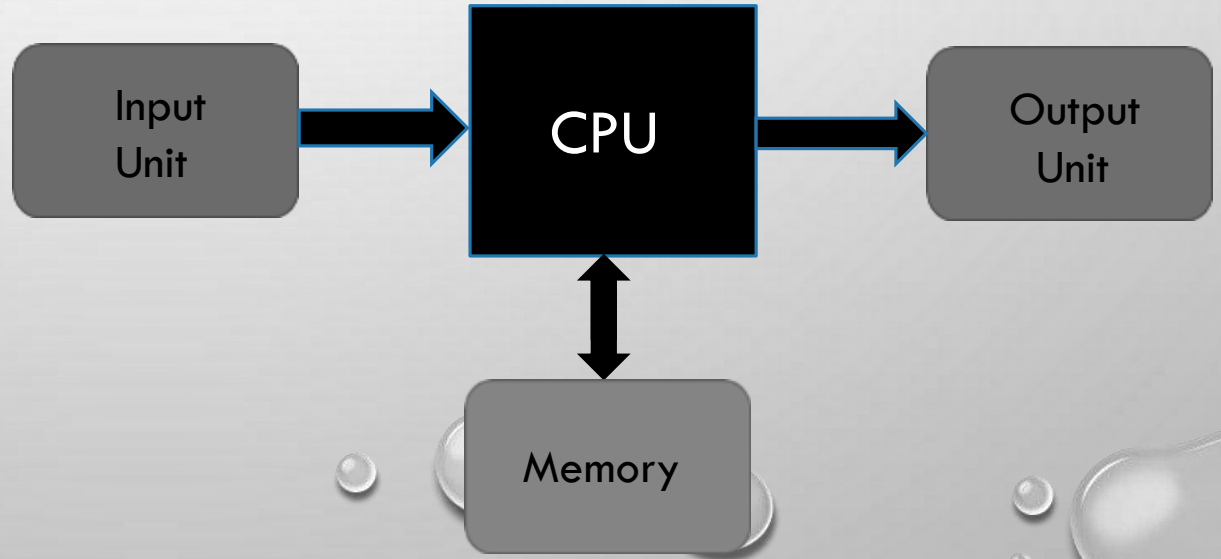
Computer System Organization

❖Digital Computer

Digital computer is a program able machine that received data as input storage and multiple data and provides output in a useful format with the high speed and accuracy.

- Types of digital computer

a) Central Processing Unit (CPU)
b) Memory Unit
c) Input Unit
d) Output Unit

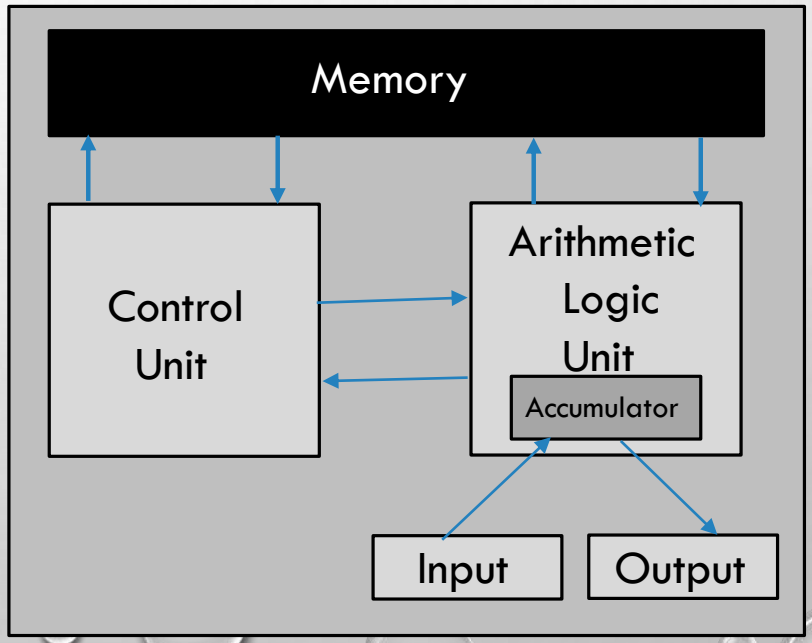


CPU ORGANIZATION

The CPU is the brain of a computer system. It is responsible for all processing in the system. It also controls the operation of other units of a computer system. It receives data input, executes instructions, and processes information. It communicates with input/output (I/0) devices, which send and receive data to and from the CPU.

A Typical CPU Consists of the following interconnected functional units.

a) Arithmetic / Logic unit
b) Control unit
c) Registers



a) Arithmetic logic Unit (ALU)

- ALU is fundamental building block of the CPU. All the computing functions are maintained in this unit.
- This unit executes all arithmetic operations (ADDition, SUBtraction, INCrementing, DECrementing etc) and logical operations (ANDing, ORing, XORing, NOTing etc).
- "It consists of a complicated set of logic circuit, registers and accumulator to carry out all the operations .
- The accumulator is the main register of the ALU. In execution of the most of the instructions the accumulator is used to store a input data or output result.
- The data upon which operations are performed can come from memory or from external input. The resulted data may be transferred to memory or to output unit.
- The operation to be performed is specified by signals from the control unit.

b) Control Unit

- Control unit is another important unit of a CPU. It directs the operation of the other units by providing timing and control signals.
- It is made up of the control signal generating circuitry (clock) and the command (instruction) decoder.
- The control unit contains the necessary logic to interpret instructions and togenerate the signals necessary for the execution of those instructions.
- Another purpose of control unit is, controlling the data flow between CPU andperipheral devices/peripheral chips.

c) Registers

- Registers are sets of flip-flops which can hold data. They are primarily used to storethe data temporarily during the execution/runtime of the program.
- A CPU contains several kinds of registers that can be classified according to the instructions provided to the CPU. The registers are basically 8 bit, 16 bit or 32 bit according to the type.
- Each CPU has a limited number of registers, so they can only be used for very shortterm data storage while that data is being processed

❖ COMPUTER INSTRUCTION

computer instruction is a binary code that specifies a sequence of micro operations for the computer. A binary instruction code is stored in one word of memory. The data required for the instruction code is also stored in memory. The special pointers are used to help processor to differentiate between instruction code and data. The instruction pointer helps the processor to keep track where in memory the instruction codes are stored. Similarly, the data pointers keep track of where the data area in memory starts.

❖ TYPES OF INSTRUCTIONS

1. Data transfer instructions
2. Arithmetic instructions
3. Logical instructions
4. Branch instructions
5. Control instructions
6. I/O instructions

1. **Data Transfer Instructions:** This group consists of large number of instruction that perform data transfer between:

    - Register to register
    - Register to memory
    - Memory to register
    - Memory to memory

For example, consider the following data transfer instructions:

(a) LD: (Load) The LD instruction transfers the data content from memory CPU register, usually an accumulator. (Memory read)

(b) ST: (Store) The ST instruction transfers the data content from processor register into memory (Memory Write)

2) Arithmetic instructions : This group consists of instructions that perform arithmetic operations such as addition, subtraction, increment and decrement.

For example, consider the following arithmetic instructions:

(a) ADD: The instruction ADD is used to add two operands. The destination operand is always in processor register (usually accumulator) while the source operand can be a register, immediate data, or in memory.
(b) INR: The INR instruction is used for incrementing an operand by one. It works on a single operand that can be either in a register or in memory.

3. Logical instructions: This group consists of instructions that perform logical operations such as AND, OR, XOR, compare & complement on a bit-by-bit basis .These instruction sets also include the following shift instructions:

- Arithmetic shift
- Logical shift
- Rotate shift

For example, consider the following logic instructions:

(a) AND: The AND instruction is used to clear bits of an operand selectively byANDing the operand with another operand that has 0's in the bit positions that must be cleared.
(b) COM: The COM instruction logically complements the value of the specified operand by inverting all the bits of the operand

4 Branch instructions: This group consists of instructions that transfer the program execution (conditionally or unconditionally) to the memory address specified within the instruction or supplied by the processor or given by extra hardware.

For example, consider the following branch instructions:

(a) JMP address: Load PC with the address specified within the instruction. This instruction unconditionally transfer the program execution to the memory address specified.
(b) CALL address: This instruction unconditionally transfers the program control to subroutine.

5. Control Instructions: This group consists of instructions that control the stack and machine operations. The stack instructions perform the data transfer between registers and stack memory. The machine instructions handle the interrupt and halt conditions of microprocessor.

For example, consider the following control instructions:

(a) NOP: No operation is performed. It is a dummy instruction that is fetched and decoded but has no effect. Processor
(b) HLT: Halts processor to execute any further instruction. The remains in HALT state until interrupt occurs.
(c) EI: Set the interrupt enable flip-flop to enable interrupts.

6. I/O Instructions: This group consists of instructions that perform input and output operations. These I/O instructions are used for interacting with user, Le. getting data from the user or showing data to the user on a monitor or printer. An input instruction allows a peripheral to transfer a word to either a register or memory. An output instruction enables a processor to transfer a word into the buffer register of a peripheral device.

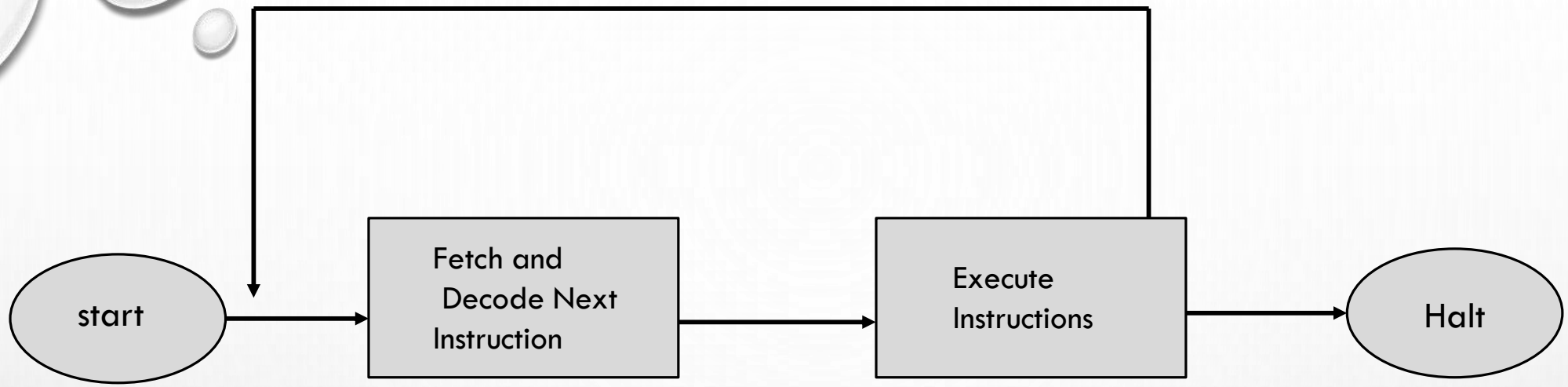
(a) IN port-address: Load the word available at the input port whose address is specified within the instruction into the accumulator.
(b) OUT port-address: Send out the content of the accumulator to the port whose address is specified within the instruction.

➢ INSTRUCTION EXECUTION

The main function of the computer is to execute programs. The programs itself are made up of instructions residing in the memory. Instruction is command which is given by the user to computer. Execution is the process by which a computer performs instruction. Instruction execution means a program to be executed by a processor consists of a set of instructions stored in memory.

Instruction Cycle

Instruction cycle comprises of repeated execution of three steps - one is called Fetch, second is decode and next is called Execute. An instruction cycle (also known as the fetch- decode-execute cycle) is the basic operational process of a computer as shown in Fig. It is the process by which a computer retrieves a program instruction from its memory, determines what actions the instruction dictates, and carries out those actions.



Instruction cycle

In the basic computer, each instruction cycle consists of the following steps :

1. Fetch the instruction from memory into the instruction register(IR).
2. Decode the fetched instruction.
3. Locate the operands used by the instruction.
4. Fetch operands from memory (if necessary).
5. Execute the instruction.
6. Store the results in the proper place.
7. Go back to step 1 to fetch the next instruction.

S.NO	**RISC**	**CISC**
1.	Reduced instructions set computer	Complex instruction set computer
2.	It has more instructions	It has more instructions
3.	It has faster execution	It has slower execution
4.	It has less complex microcode	It has complex microcode
5.	it has few addressing mode	It has large variety of addressing mode
6.	It has fixed length instruction format	It has variable length instructions format
7.	It has hardwired control	It has micro programmed control
8.	It mostly comprise of register based instructions	It mostly comprise of memory based instructions
9.	It support large number of general purpose registers	It supports small number of general purpose registers
10.	Best suited for real time applications	Cannot be used for real time application
11.	It is highly pipelined	It is less pipelined
12.	It has single cycle instructions execution	It has multicycle instruction execution
13.	It does not use microprogramming	It uses microprogramming
14.	Decoding is very easy	Decoding is very difficult
15.	Examples of RISC : ULTRA SPARC, POWER PC etc.	Example of CISC : intel80486, Celeron, pentium etc.

➢ Instruction level parallelism

It is a measure of number of instructions in a computer program that can be executed simultaneously. The basic ides of ILP is to execute several instructions in parallel. It uses pipelining to overlap the execution of instruction and improve performance.

Pipelining

Pipelining is one of the effective methods to increase parallelism in CPU operation where different stages perform repeated functions on different operands. It is a technique of splitting up a sequential operation (task) into sub-operations (sub-tasks), with each sub- operation being executed in a special dedicated section that operates concurrently with all other sections. The result of each section is transferred to the next section in the pipeline. The final result is obtained after the operand is passed through all sections. The name "pipeline" implies the computation of several operations in distinct sections at the same time.

In this section we will look at two approaches of instruction-level parallelism:

(a) Instruction Pipelining
(b) Super-Scalar Architecture

Instruction Pipelining

The instruction pipeline represents the stages in which an instruction is moved through the processor, including its being fetched, decoded, buffered, and then executed. Without a pipeline, a CPU gets the first instruction from memory, performs the operation it calls for, and then goes to get the next instruction from memory, and so forth. While fetching the instruction, the arithmetic part of the processor is idle. It must wait until it gets the next instruction. The staging of instruction fetching is continuous. The result is an increase in the number of instructions that can be performed during a given time period. It is a way of speed up instruction execution.

To apply the concept of instruction pipelining, the execution of single instruction within CPU is broken down into a following number of stages:

1) Instruction Fetch (F): Read instruction from memory
2) Instruction Decode (D): Decode the instruction for operation
3) Read Data (R): Read data from memory
4) Instruction Execution (E): Perform the operation
5) Store Result (S): Store the result in destination

superscalar architecture

A superscalar architecture consists of multiple pipelines that are working in parallel. It includes all features of pipelining but, in addition, several instructions can be initiated simultaneously and executed independently. It is applicable to both RISC & CISC, but usually in RISC

In superscalar, multiple independent instruction pipelines are used. Each pipeline consists of multiple stages, so that each pipeline can handle multiple instructions at a time.

Advantage of Superscalar Architecture

- Maximum performance
- Propagation delay is Less
- Faster CPU throughput
- Compatibility between generations

Disadvantage of Superscalar Architecture

- Problem in scheduling
- Handling of data dependencies
- Complex
- Hardware Time cost of the dispatcher
- Resource conflicts

➢ PROCESS LEVEL PARALLELISM (PLP)

(Processor level parallelism refers to multiple processors (CPUs) working together on the same problem. Instruction level parallelism helps performance but only to a factor of 5 to 10, but processor-level parallelism gains a factor of 50, 100, and even more.

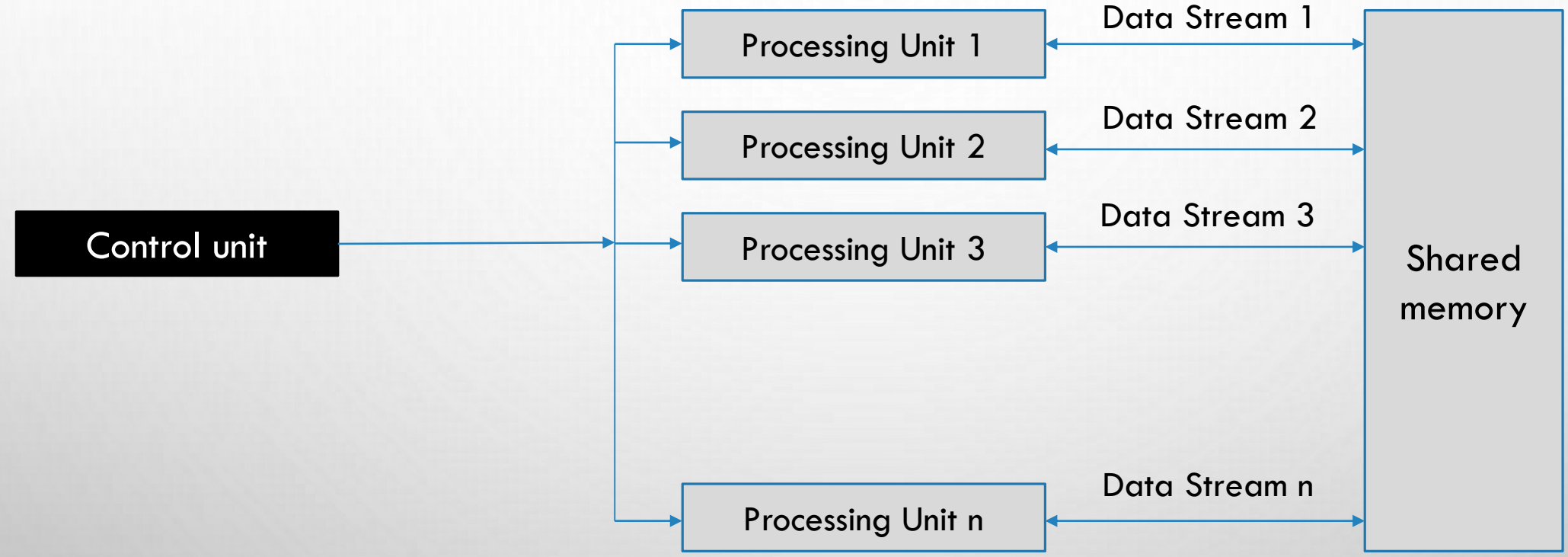
Three types of Processor-Level Parallelism are:

- Array Processors
- Multiprocessors
- Multicomputers

- **Array processors**

Array processors consist of a large number of identical processors with single control unit. A single control unit controls the same sequence of instructions on different sets of data (in parallel).

The operations are first divided into several streams of operands (called array elem and run for each stream in the parallel processing units. Thus, providing a single instru stream and multiple data streams. An array processor is also referred as Single Instruc stream Multiple Data-stream (SIMD) processor.



SIMD Array processer

Memory Organization

❖ MEMORY

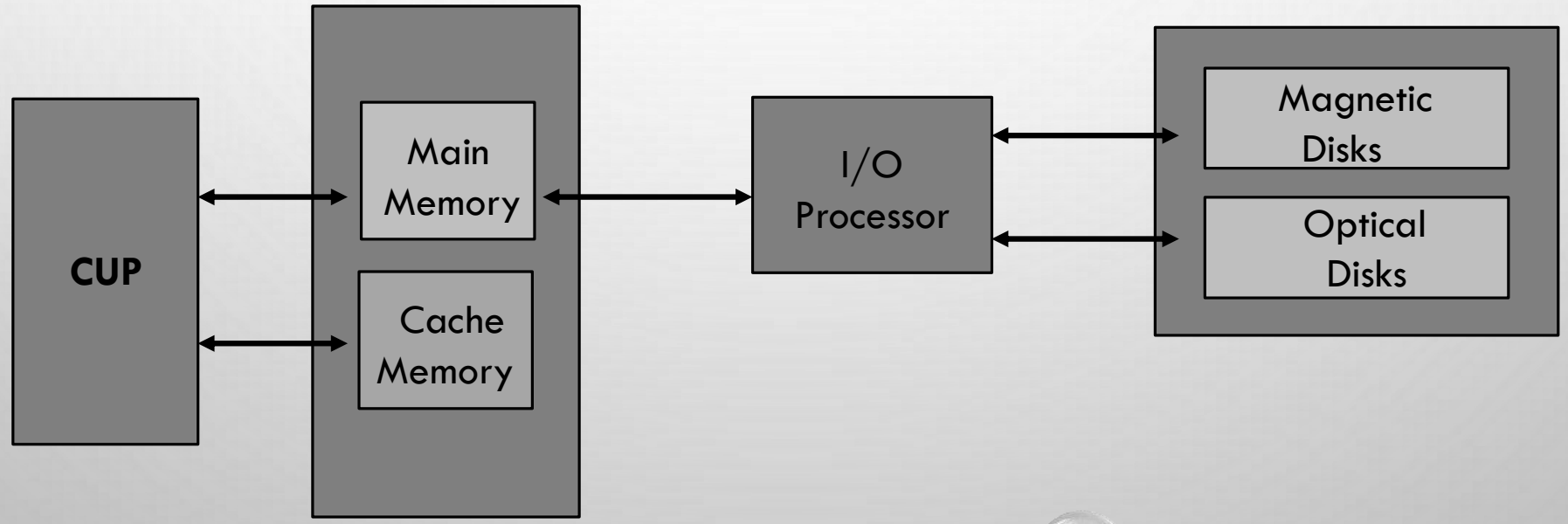
Memory is one of the main components of the computer system that is used to store the data and instructions for processing. It is actually a storage area within the computer where all the inputs are stored before processing and the outputs are stored after processing of inputs.

The memory of a computer can be divided into two broad categories:

1. Primary Memory
2. Secondary Memory

✓ Memory Hierarchy

The memory unit is an essential component in any digital computer. Personal computer (PC) systems have a hierarchical memory structure consisting of main memory, cache memory and secondary memory as shown in Fig.

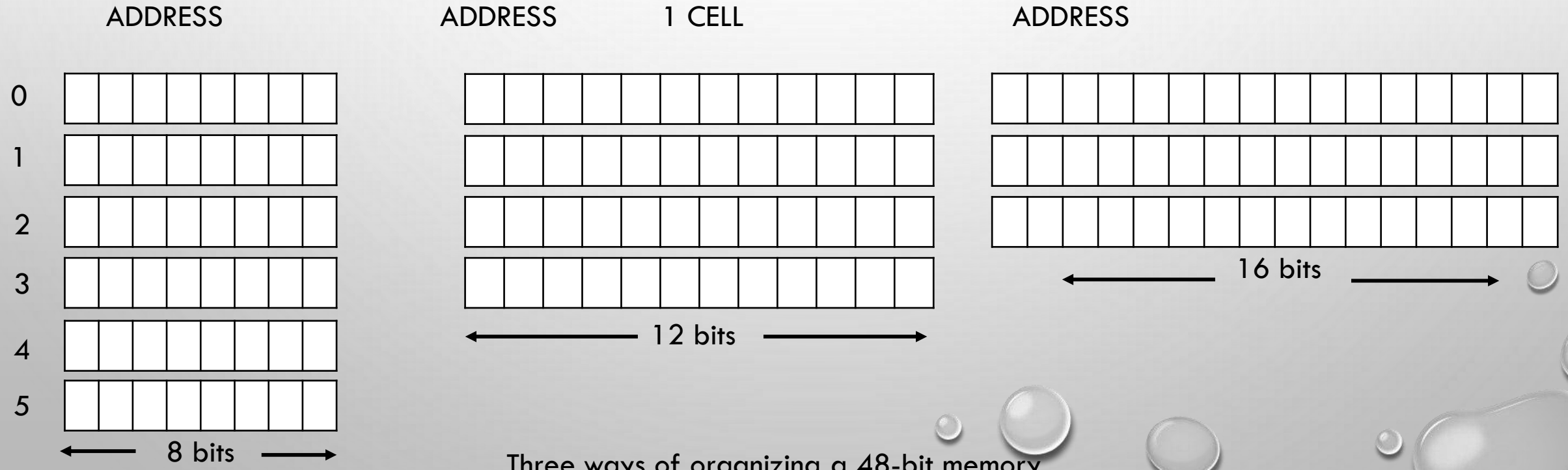


Computer memory Hierarchy System

✓ MEMORY ADDRESS

Memory consists of a number of storage locations (or cells). Each location in the storage has a unique number, called memory address or storage address. The CPU uses the address bus to communicate with memory address.

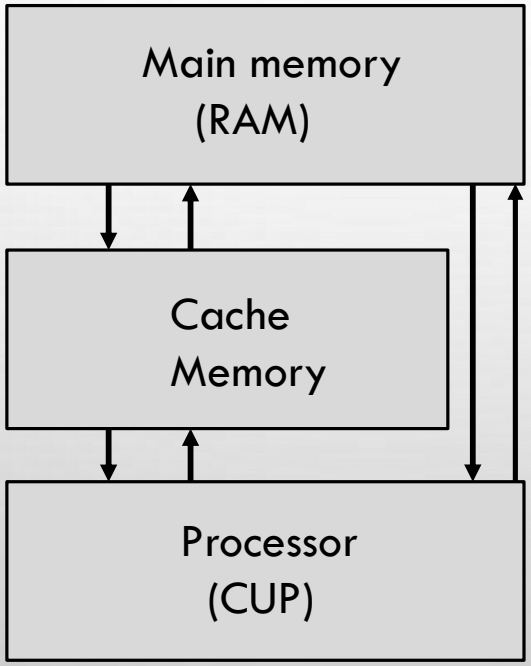
If a memory has n cells, they will have addresses 0 to n - 1. All cells in a memory contain the same number of bits. If a cell consists of k bits, it can hold any one of 2 different bit combinations. Figure shows three different organizations for a 48-bit memory.



Three ways of organizing a 48-bit memory

✓ CACHE MEMORY

The CPU accesses the main memory to read the instructions or data. The speed at which the CPU executes the instructions is much faster than the speed at which the instructions are fetched from the main memory. In order to make it compatible with storage, a small very high-speed memory is used between main memory (RAM) and CPU as shown in Fig. 2.8. This type of memory is called Cache Memory/Buffers.



❖ RAID

RAID stands for Redundant Array of Independent Disks. RAID technologies setup multiple storage disks into array, which enables storing of same data on multiple disks .

➢ RAID-0

RAID Level 0 is referred to disk stripping. It split up data among the all available disk drives. It is non-redundant, hence it does not protect against data lost due to drive failures The only advantage of RAID-0 is high throughput.

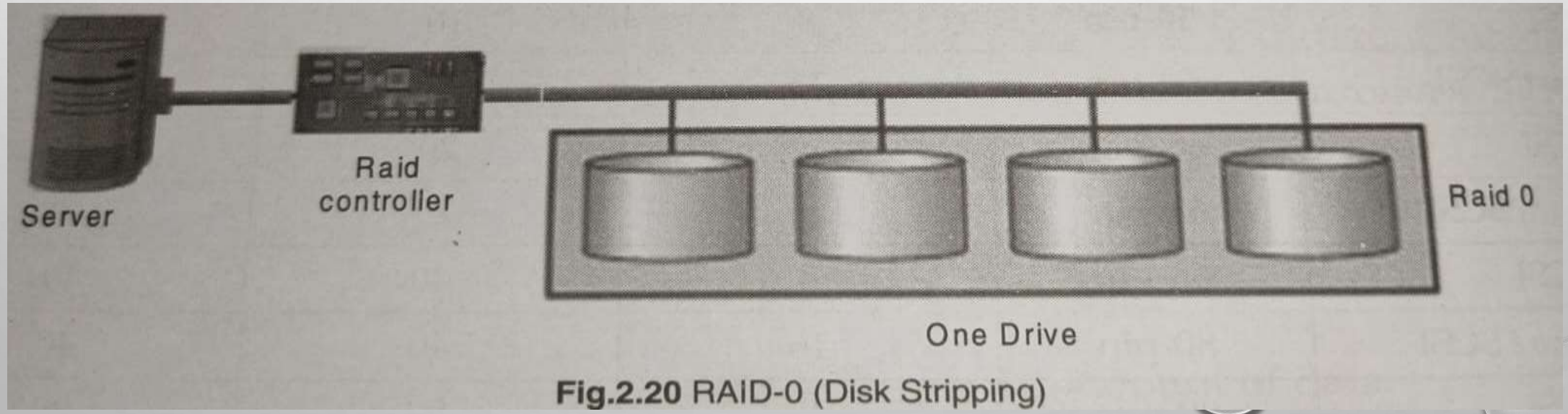


Fig.2.20 RAID-0 (Disk Stripping)

➢ RAID-1

RAID Level 1 is referred to disk mirroring. It involves an array of two hard disks. One disk stores the original data and another disk stores replicate of original data (thus, mirroring). If one disk fails, another disk continues to operate. It is redundant, thus it protect against data lost due to drive failures, which enable data security. It read or writes both drives at same time. The RAID 1 offers fast read performance, but slow write performance as compared to single disk storage.

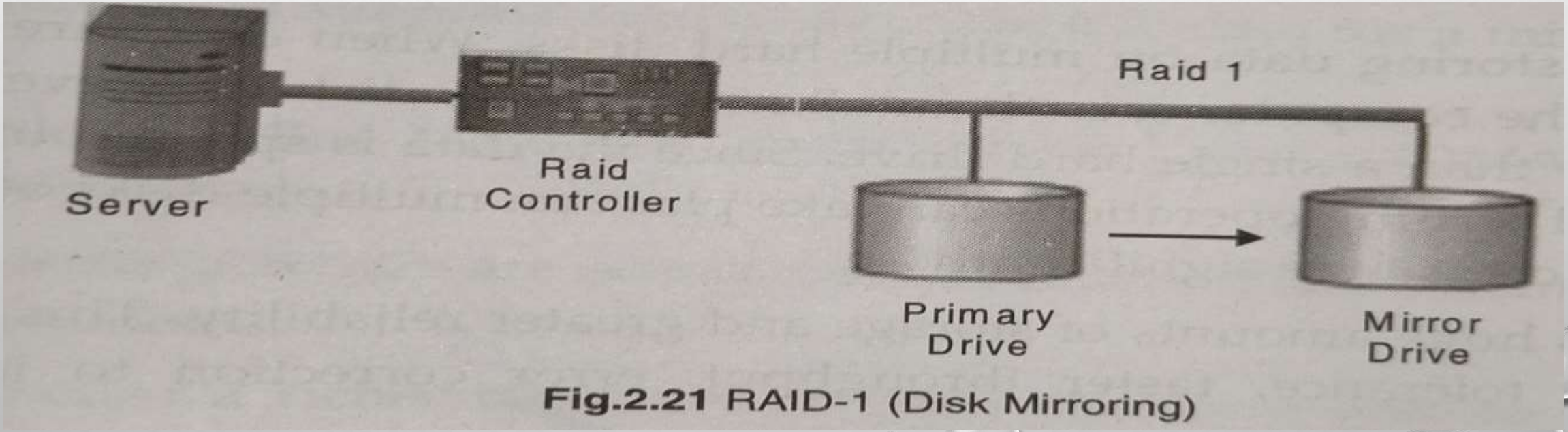


Fig.2.21 RAID-1 (Disk Mirroring)

RAID-3

RAID Level 2 is referred to disk stripping with dedicated parity disk. It split up data byte-by- byte across disk array but one disk in array stores parity information. The parity info is used to detect errors. On failure of any disk in array, a new disk is restored using the parity information. RAID-3 is common in single-user systems with long record Applications.

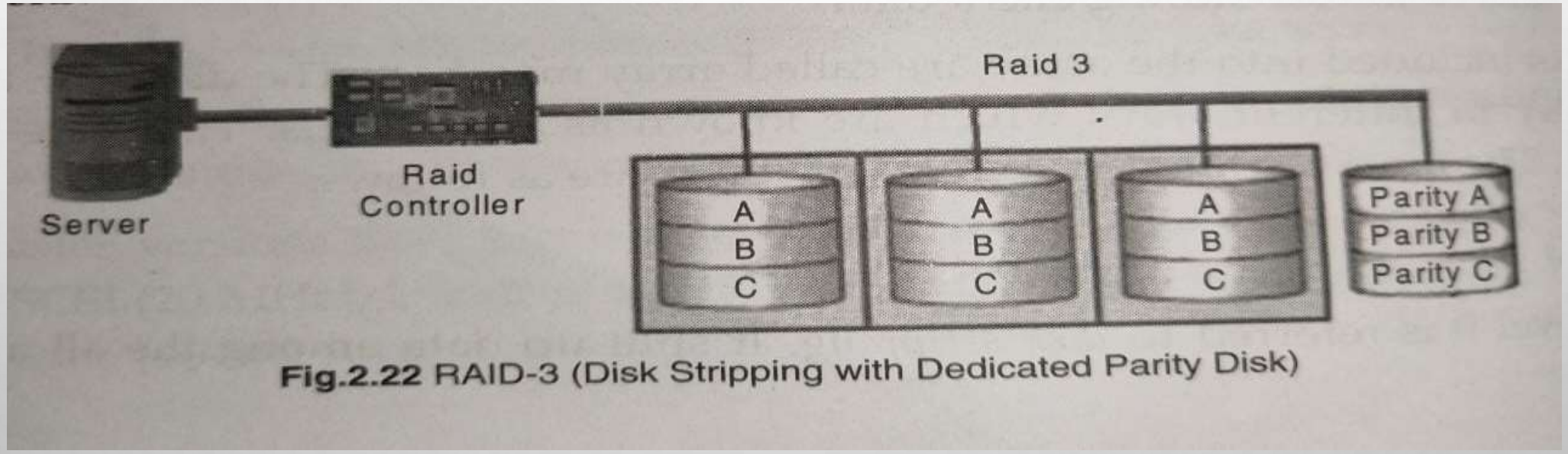


Fig.2.22 RAID-3 (Disk Stripping with Dedicated Parity Disk)

RAID-5

RAID Level 5 is referred to disk stripping with parity. Instead of using a dedicated parity disk, the parity info is stripped across all disks in array. It is one of the most commonly used RAID level. If any disk fails, it can recover the lost data. RAID-5 is common in multi-user systems

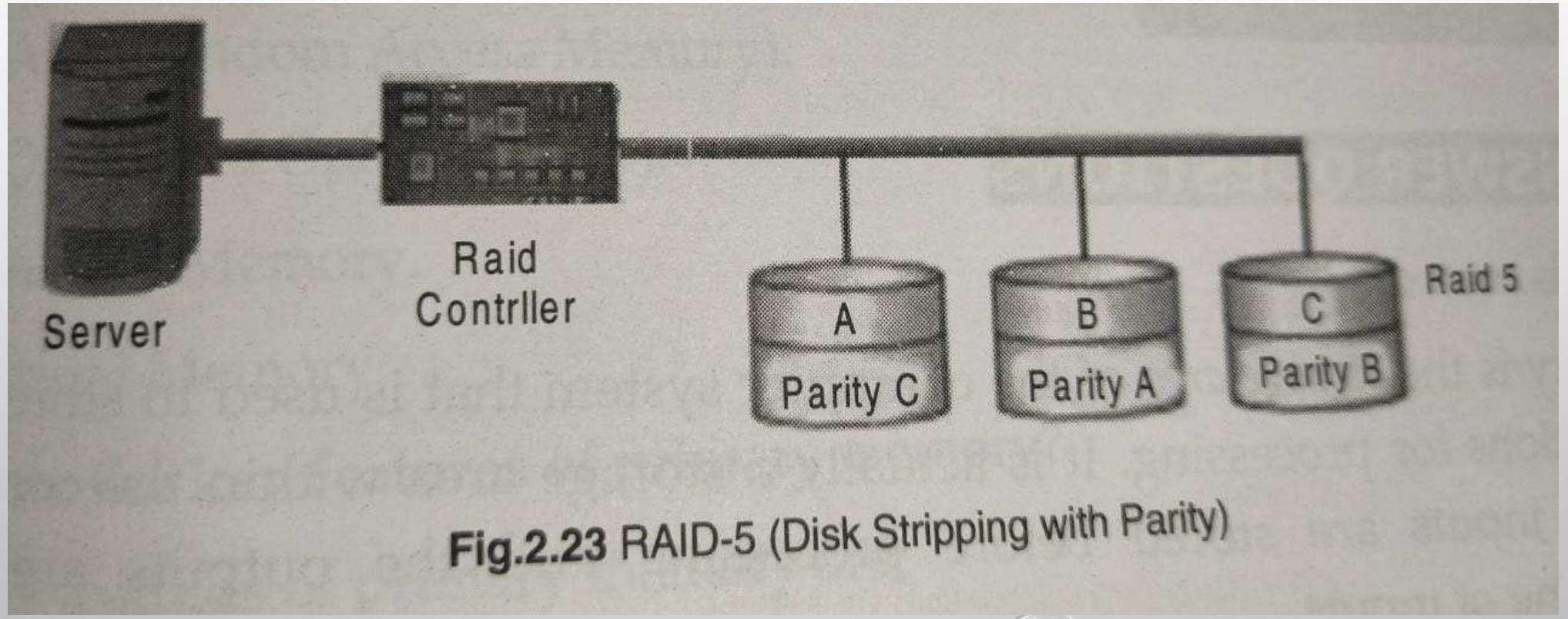


Fig.2.23 RAID-5 (Disk Stripping with Parity)

Instruction Set Architecture

❖ Instruction Format

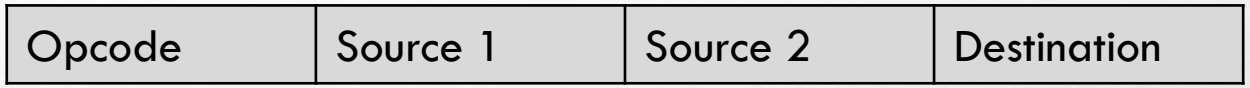
An instruction is a command given to a computer to perform a specific operation on some given data. When the assembler processes an instruction, it converts the instruction from its mnemonic form to a standard machine-language (binary) format called an instruction format. An instruction format defines the layout of the bits of an instruction, in terms of its constituent's parts. Every Computer has its own particular instruction code format.

An instruction format with a distinct addressing mode field is shown in Fig :

Opcode	Mode Field	Address Field

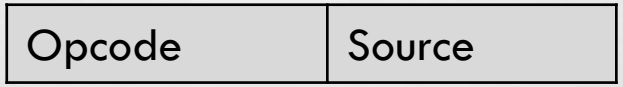
The various types of instruction format based upon number of address fields are discussed below:

- Three Address Instructions



- Two Address Instructions

- One Address Instructions



- Zero Address Instructions

❖ addressing modes

Each instruction of a computer specifies an operation on certain data called operand. The operand may be in accumulator, general purpose registers or memory location. The various techniques to specify the operand for the instruction are called addressing modes

The most common addressing modes are:

- Immediate addressing mode
- Direct addressing mode
- Indirect addressing mode
- Register addressing mode
- Register indirect addressing mode
- Auto Increment and Decrement
- Relative addressing mode.
- Base Register addressing mode
- Indexed addressing mode

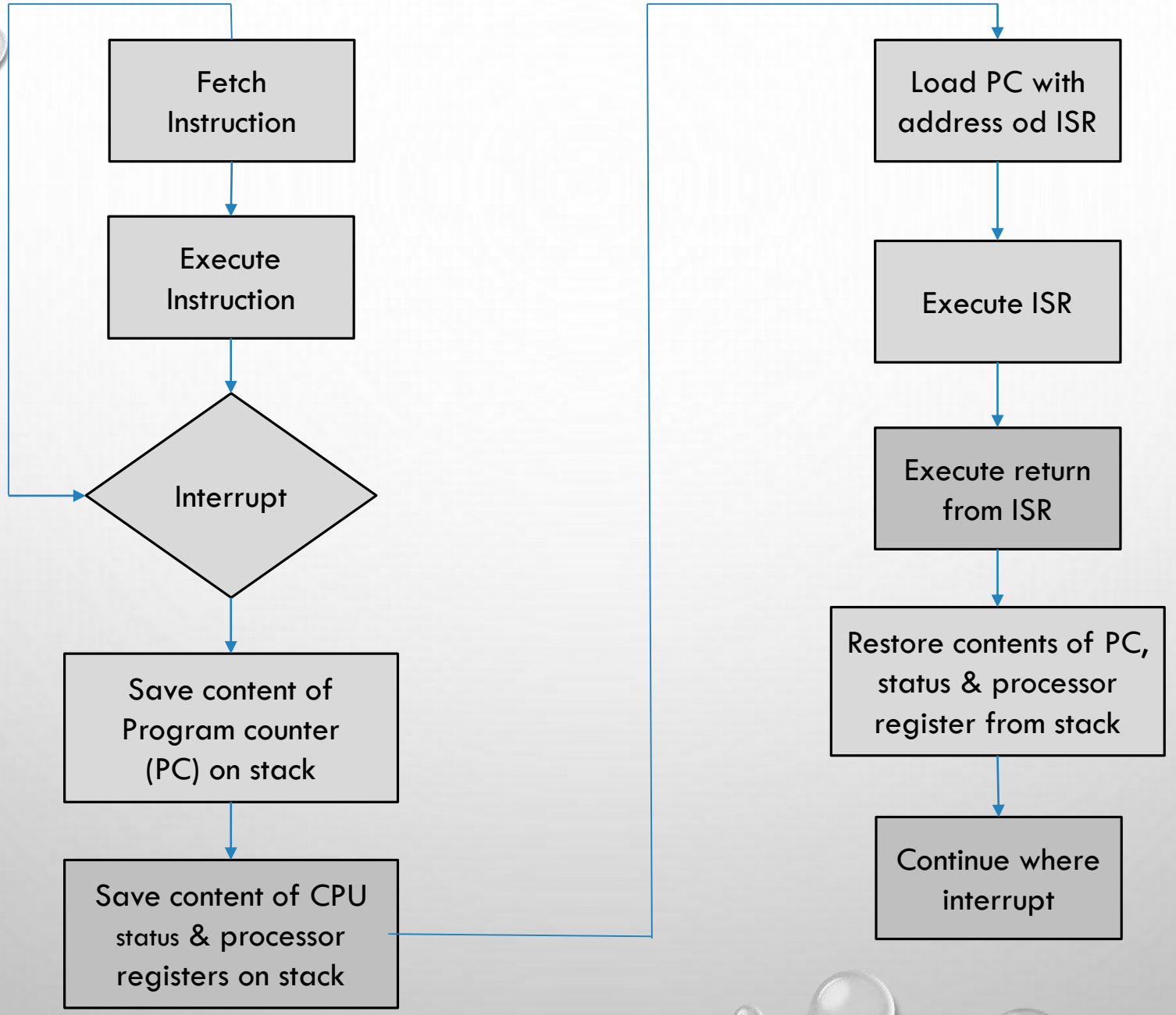
❖ Program Interrupts

An interrupt is an event inside a computer system that causes the CPU to suspend its current activities and perform a particular task to service the interrupt. It is used to get CPU's attention. This approach helps to maximize the CPU utilization. Interrupts are the primary means by which external devices obtain the services of the CPU.

➢ Interrupt types

In general, interrupts can be classified in the following ways:

- External and Internal interrupts
- Hardware and Software interrupts.
- Maskable and Non-Maskable interrupts



Flowchart of Interrupt processor