Noida Institute of Engineering and Technology, Greater Noida

Introduction

Unit: 1

Computer Organization & Architecture

B Tech 3rd Sem

Khushboo
Assistant Professor
NIET, Greater Noida

Content

- Introduction
- Functional units of digital system and their interconnections
- Buses, bus architecture, Types of buses
- Bus arbitration
- Register, bus and memory transfer.
- Processor organization
- General registers organization
- Stack organization
- Addressing modes

Course Objective

To understand the types of organizations, structures and functions of computer, design of arithmetic and logic unit and float point arithmetic. To understand the concepts of memory system, communication with I/O devices and interfaces.

Prerequisite and Recap

- **Basic knowledge of Digital Logic Design**
- **Basic Idea of various gates and circuit**

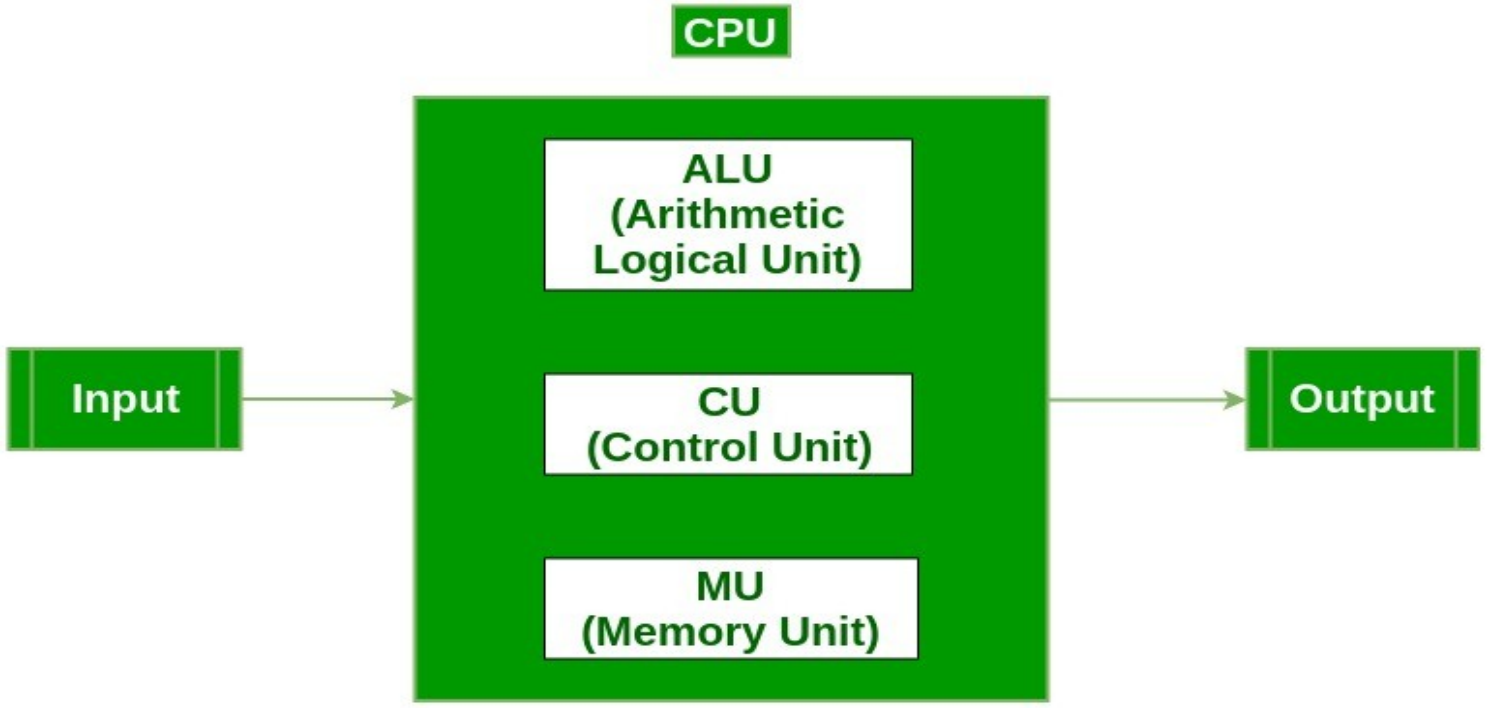
Introduction to Topic 1

Name of Topic	Objective of Topic	Mapping with CO
Functional units of digital system and their interconnections	Students will be able to understand about different units of computer and how they are connected	CO 1

Introduction

- Computer Organization and Architecture provides in-depth knowledge of internal working, structuring, and implementation of a computer system
- Computer Architecture is concerned with the way hardware components are connected together to form a computer system.
- Computer Organization is concerned with the structure and behaviour of a computer system as seen by the user.

Functional Units of Computer System



Block Diagram

Functional Units of Computer System

1. **Input Unit :**

- The input unit consists of input devices that are attached to the computer.
- These devices take input and convert it into binary language that the computer understands.
- Common input devices are keyboard, mouse, joystick, scanner etc.

2. **Central Processing Unit (CPU) :**

- The CPU is called the brain of the computer because it is the control centre of the computer.
- It first fetches instructions from memory and then interprets them so as to know what is to be done.
- CPU executes or performs the required computation and then either stores the output or displays on the output device

Functional Units of Computer System

3. Arithmetic and Logic Unit (ALU) :

- It performs mathematical calculations and takes logical decisions.
- Arithmetic calculations include addition, subtraction, multiplication and division.
- Logical decisions involve comparison of two data items to see which one is larger or smaller or equal.

4. Control Unit :

- It coordinates and controls the data flow in and out of CPU and also controls all the operations of ALU, memory registers and input/output units.
- It decodes the fetched instruction, interprets it and sends control signals to input/output devices until the required operation is done properly by ALU and memory.

Functional Units of Computer System

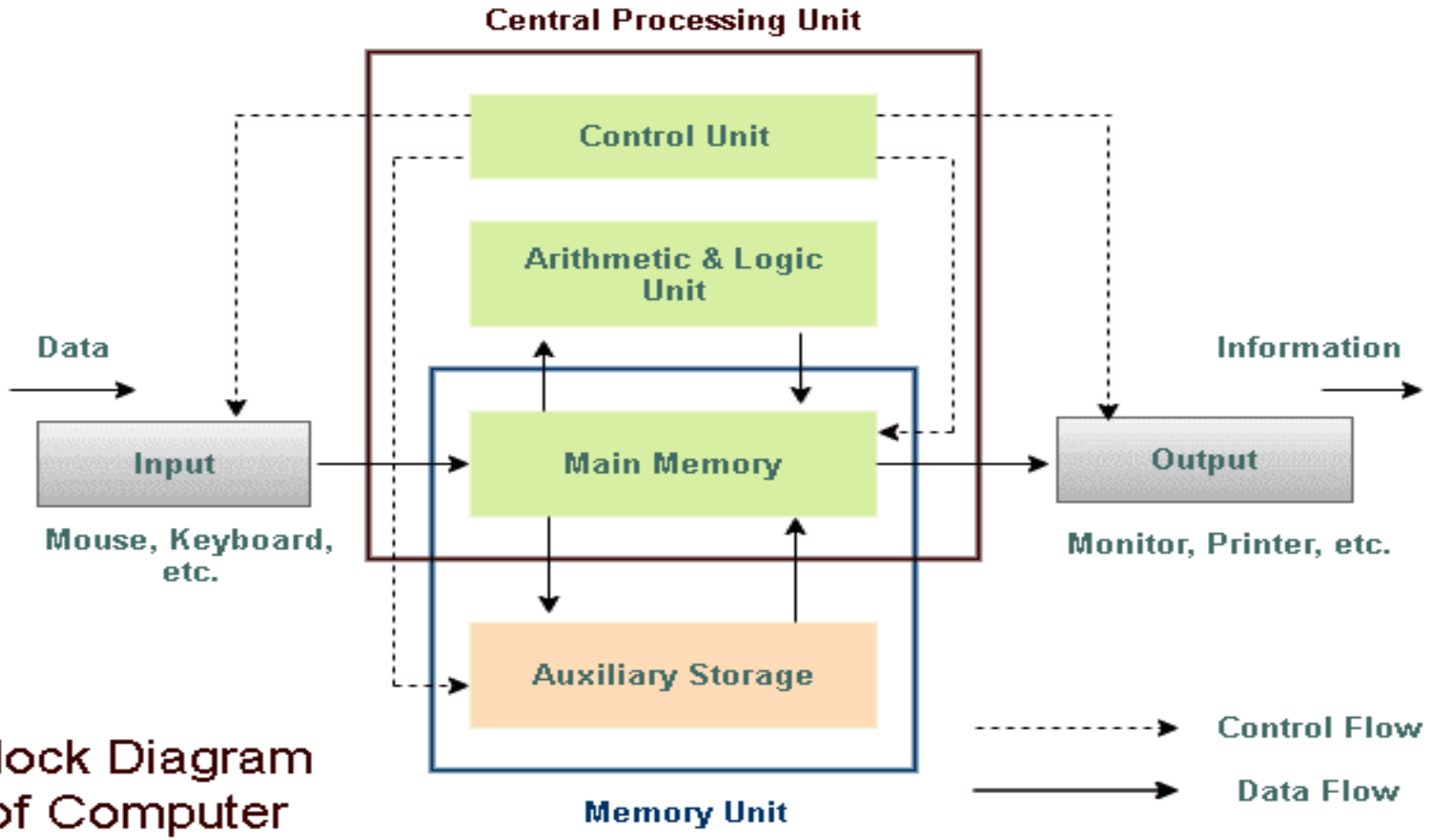
5. Memory :

- Memory attached to the CPU is used for storage of data and instructions and is called internal memory.
- The internal memory is divided into many storage locations, each of which can store data or instructions

6. Output Unit :

- The output unit consists of output devices that are attached with the computer.
- It converts the binary data coming from CPU to human understandable form.
- The common output devices are monitor, printer, plotter etc.

Interconnection of Functional unit



Block Diagram of Computer

Memory

1. **Primary / Main memory:**

- The memory unit that establishes direct communication with the CPU is called **Main Memory**. The main memory is often referred to as RAM (Random Access Memory).
- It holds the data and instructions that the processor is currently working on.

2. **Secondary Memory / Mass Storage:**

- The contents of the secondary memory first get transferred to the primary memory and then are accessed by the processor, this is because the processor does not directly interact with the secondary memory.
- The memory units that provide backup storage are called **Auxiliary Memory**. For instance, magnetic disks and magnetic tapes are the most commonly used auxiliary memories.

Daily Quiz

1. _____ is considered as the brain of the computer.
2. _____ is smallest unit of the information.
3. _____ is the decimal equivalent of the binary number 10111.
4. _____ section is used to perform logic operations such as comparing, selecting, matching of data.
5. _____ unit is used to store data and instructions.
6. The functions of sequencing and execution are performed by using
   a) Input Signals
   b) Output Signals
   c) Control Signals
   d) CPU
7. _____ is a way in which the components of a computer are connected to each other.

Daily Quiz with Answers

1. _____ is considered as the brain of the computer. (CPU)
2. _____ is smallest unit of the information? (Bit)
3. _____ is the decimal equivalent of the binary number 10111? (23)
4. _____ section is to perform logic operations such as comparing, selecting, matching, and merging of data? (ALU/Logic section)
5. _____ is used to store data and instructions. (Memory)
6. The functions of sequencing and execution are performed by using
   a) Input Signals
   b) Output Signals
   c) Control Signals
   d) CPU
7. _____ is a way in which the components of a computer are connected to each other. (Computer Architecture)

Recap

- Computer Organization and Architecture provides in-depth knowledge of internal working, structuring, and implementation of a computer system.
- Computer Architecture is concerned with the way hardware components are connected together to form a computer system.
- Computer Organization is concerned with the structure and behaviour of a computer system as seen by the user.
- Computer consists of different blocks such as Input, output, Control unit, memory unit and ALU.

Introduction to Topic 2

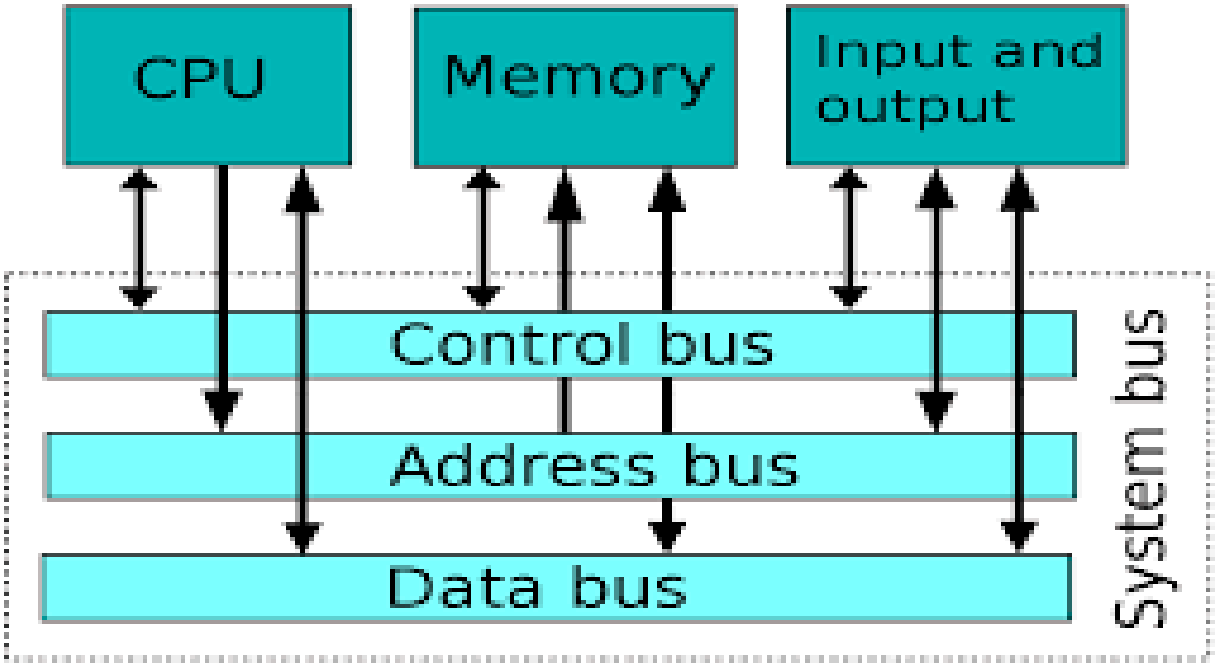
Name of Topic	Objective of Topic	Mapping with CO
•Buses, bus architecture, Types of buses	Students will be able to know the architecture and types of buses.	CO 1

BUS

BUS :

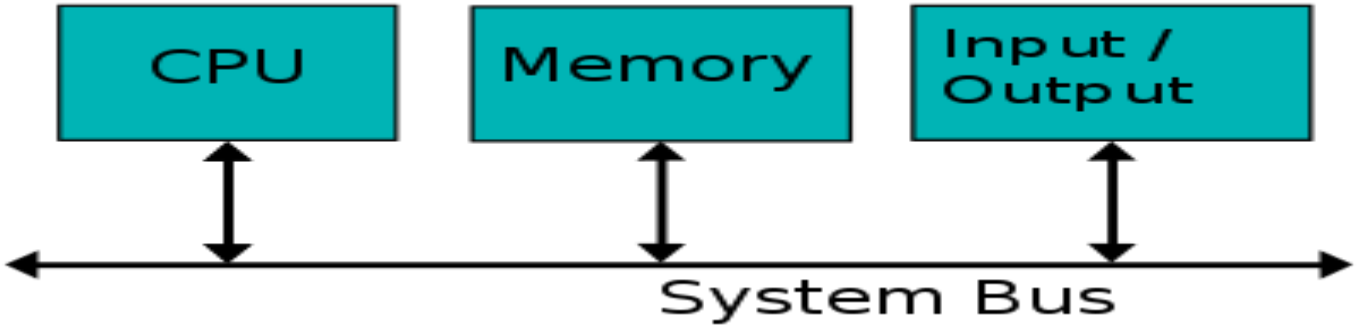
- A **bus** is a **common pathway** through which information flows from one computer component to another.
- It is a subsystem that is used to transfer data and other information between devices.
- Means various devices in computer like(Memory, CPU, I/O and Other) are communicate with each other through buses.

Bus Architecture



SYSTEM BUS

- A **system bus** is a single computer bus that connects the major components of a computer system, combining the functions of a data bus to carry information, an address bus to determine where it should be sent or read from, and a control bus to determine its operation



SYSTEM BUS

- Types of Computer BUS:
    1. **Data Bus**
    2. **Address Bus**
    3. **Control Bus**

SYSTEM BUS

1. **Data bus**

- It is a bidirectional pathway that carries the actual data (information) to and from the main memory.
- Data Lines provide a path for moving data between system modules.
- It is bidirectional which means data lines are used to transfer data in both directions.
- CPU can read data on these lines from memory as well as send data out of these lines to a memory location or to a port.
- The no. of lines in data lines are either 8,16,32 or more depending on architecture.

SYSTEM BUS

2. Address bus

- Address Lines are collectively called as address bus.
- It is a **unidirectional** pathway that allows information to travel in only one direction.
- No. of lines in address are usually 16,20,24, or more depending on type and architecture of bus
- It is an internal channel from CPU to Memory across which the address of data(not data) are transmitted.
- It is used to identify the source or destination of data.
- Here the communication is one way that is, the address is send from CPU to Memory and I/O Port but not Memory and I/O port send address to CPU on that line and hence these lines are unidirectional.

SYSTEM BUS

3. Control bus

- It **carries the control and timing signals** needed to coordinate the activities of the entire computer.
- They are used by CPUs for Communicating with other devices within the computer.
- They are bidirectional.
- Typical Control Lines signals are

  Memory Read

  Memory Write

  I/O Read

  I/O Write ,etc

BUS STRUCTURE

1. **Single Bus Structure –**

   All units are connected to the same bus.

2. **Multiple Bus Structure**

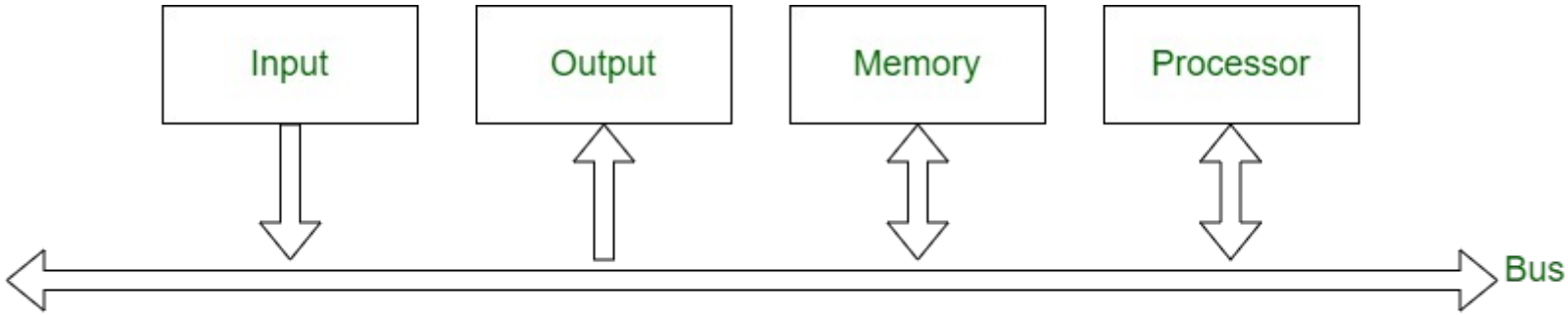
   a)Traditional Configuration

   Uses three buses – local bus, system bus and expanded bus.

   a)High Speed BUS Configuration

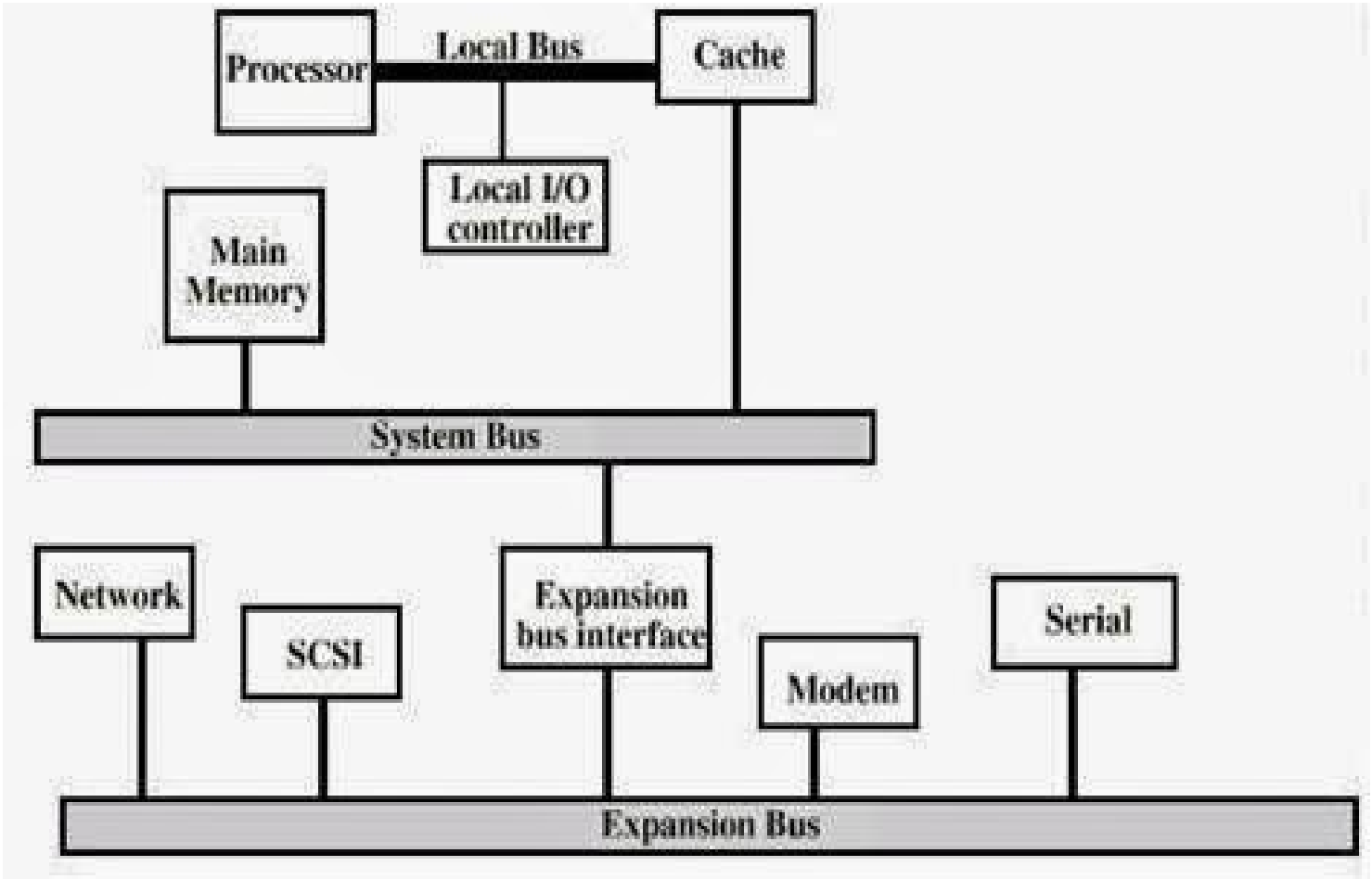
   Uses high speed bus along with three buses – local bus, system bus and expanded bus used in traditional configuration.

SINGLE BUS STRUCTURE

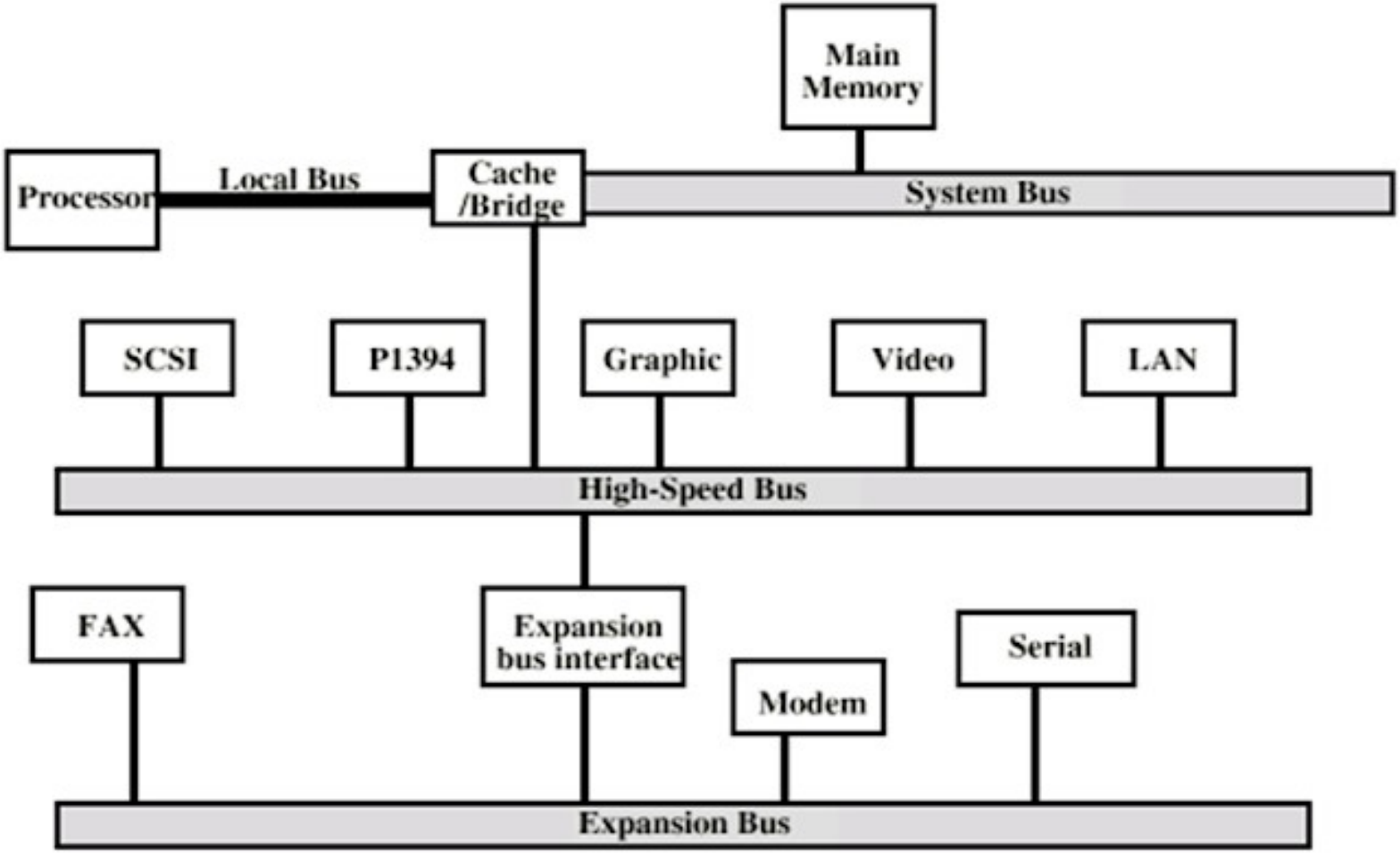


Single Bus Structure

TRADITIONAL BUS CONFIGURATION



HIGH SPEED BUS CONFIGURATION



Daily Quiz

1. _____ bus carries information between processors and peripherals.
2. _____ are unidirectional bus.
3. _____ bus controls the sequencing of read/write operations.
4. _____ is a common pathway through which information flows from one computer component to another.
5. Control bus is Unidirectional. T/F

Daily Quiz

1. _____ bus carries information between processors and peripherals. (Data Bus)
2. _____ are unidirectional bus. (Address Bus)
3. _____ bus controls the sequencing of read/write operations. (Control Bus)
4. _____ is a common pathway through which information flows from one computer component to another. (Bus)
5. Control bus is Unidirectional. **False**

Recap

Bus Type	Description
Address bus	A unidirectional pathway – information can only flow one way
Data bus	A bi-directional pathway – information can flow in two directions
Control bus	Carries the control and timing signals needed to coordinate the activities of the entire computer

Introduction to Topic 3

Name of Topic	Objective of Topic	Mapping with CO
•Bus arbitration	Students will be able to know the different schemes of bus arbitration	CO 1

Bus Arbitration

Bus Arbitration

It refers to the process by which the current bus master accesses and then leaves the control of the bus and passes it to the another bus requesting processor unit.

- **Bus master :**The controller that has access to a bus at an instance.

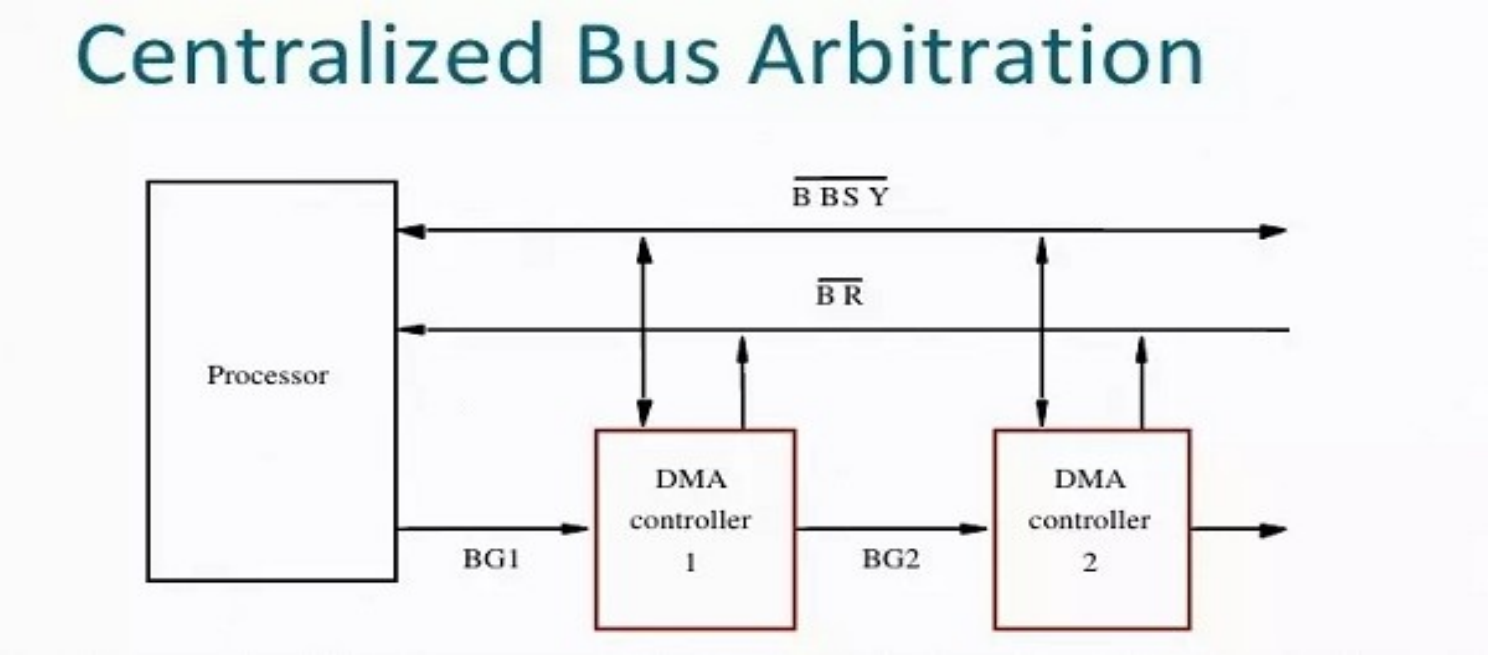
- **Bus Arbiter:** It decides who would become current bus master.

Bus Arbitration

There are two approaches to bus arbitration:

1. **Centralized bus arbitration**

   A single bus arbiter performs the required arbitration and it can be either a processor or a separate DMA controller.



Bus Arbitration

2. Distributed bus arbitration

All devices participate in the selection of the next bus master.

Bus Arbitration

Methods of BUS Arbitration

There are three arbitration schemes which run on centralized arbitration.

1. **Daisy Chaining method**
2. **Polling method**
3. **Independent Request method**

Daisy Chaining method

Advantages –

1.Simplicity and Scalability.

2.The user can add more devices anywhere along the chain

Disadvantages –

1.The value of priority assigned to a device is depends on the position of master bus.

2.Propagation delay is arises in this method.

3.If one device fails then entire system will stop working.

Polling or Rotating Priority Method

Advantages –

1.This method does not favor any particular device and processor.

2.The method is also quite simple.

3.If one device fails then entire system will not stop working.

Disadvantages –

1.Adding bus masters is difficult as increases the number of address lines of the circuit.

Fixed priority or Independent Request method

Advantages –

- This method generates fast response.

Disadvantages –

- Hardware cost is high as large no. of control lines are required.

Daily Quiz

1. To resolve the clash over the access of the system BUS we use _____
   a) Multiple BUS
   b) BUS arbitrator
   c) Priority access
   d) None of the mentioned

2. The device which is allowed to initiate data transfers on the BUS at any time is called _____
   a) BUS master
   b) Processor
   c) BUS arbitrator
   d) Controller

3. _____ BUS arbitration approach uses the involvement of the processor.
   a) Centralised arbitration
   b) Distributed arbitration
   c) Random arbitration
   d) All of the mentioned

Daily Quiz with Answers

1. To resolve the clash over the access of the system BUS we use _____
   a) Multiple BUS
   b) **BUS arbitrator**
   c) Priority access
   d) None of the mentioned

2. The device which is allowed to initiate data transfers on the BUS at any time is called _____
   a) BUS master
   b) Processor
   c) BUS arbitrator
   d) Controller

3. _____ BUS arbitration approach uses the involvement of the processor.
   a) Centralised arbitration
   b) Distributed arbitration
   c) Random arbitration
   d) All of the mentioned

Daily Quiz

4. When the processor receives the request from a device, it responds by sending _____
   a) Request signal
   b) BUS grant signal
   c) Response signal
   d) None of the mentioned

5. Once the BUS is granted to a device _____
   a) It activates the BUS busy line
   b) Performs the required operation
   c) Raises an interrupt
   d) All of the mentioned

6. After the device completes its operation _____ assumes the control of the BUS.
   a) Another device
   b) Processor
   c) Controller
   d) None of the mentioned

Daily Quiz with Answers

4. When the processor receives the request from a device, it responds by sending _____
   a) Request signal
   b) BUS grant signal
   c) Response signal
   d) None of the mentioned

5. Once the BUS is granted to a device _____
   a) It activates the BUS busy line
   b) Performs the required operation
   c) Raises an interrupt
   d) All of the mentioned

6. After the device completes its operation _____ assumes the control of the BUS.
   a) Another device
   b) Processor (After the device completes the operation it releases the BUS and the processor takes over it.**)**
   c) Controller
   d) None of the mentioned

Daily Quiz

7. The BUS busy line is used _____
   a) To indicate the processor is busy
   b) To indicate that the BUS master is busy
   c) To indicate the BUS is already allocated
   d) None of the mentioned

Daily Quiz with Answers

7. The BUS busy line is used _____
   a) To indicate the processor is busy
   b) To indicate that the BUS master is busy
   c) To indicate the BUS is already allocated
   d) None of the mentioned

Recap

- Bus arbitration refers to the process by which the current bus master accesses and then leaves the control of the bus and passes it to the another bus requesting processor unit.

- There are two approaches to bus arbitration: Centralized and Distributed

- There are three arbitration schemes which run on centralized arbitration – Daisy chain, Polling and Independent request

Introduction to Topic 4

Name of Topic	Objective of Topic	Mapping with CO
Register, bus and memory transfer.	Students will be able to know transfer between the registers, bus and memory transfer	CO 1

Register, bus and memory transfer

Register

- They are used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU.
- They are used to hold the temporary data.
- There are various types of Registers those are used for various purpose.

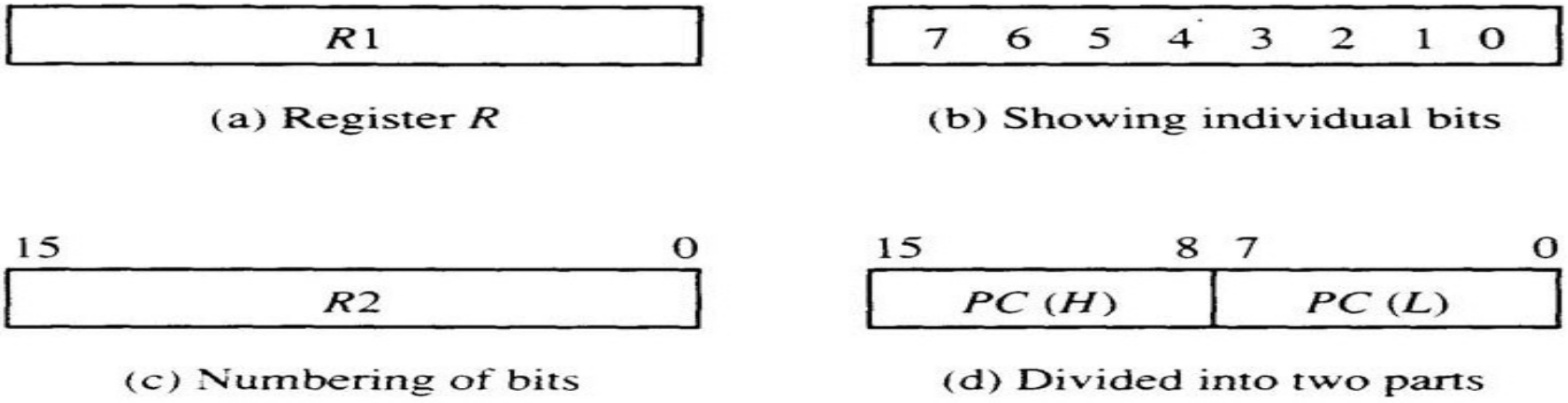
Register Symbol	Number of bits	Register Name	Register Function
DR	16	Data register	Holds memory operands
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

Register, bus and memory transfer

Register

- Computer registers are designated by capital letters (sometimes followed by numerals) to denote the function of the register.
- The register that holds an address for the memory unit is memory address register and is designated by the name **MAR**.
- The program counter register is called **PC**, **IR** is the instruction register and **R1** is a processor register

Block diagram of register.



Register, bus and memory transfer

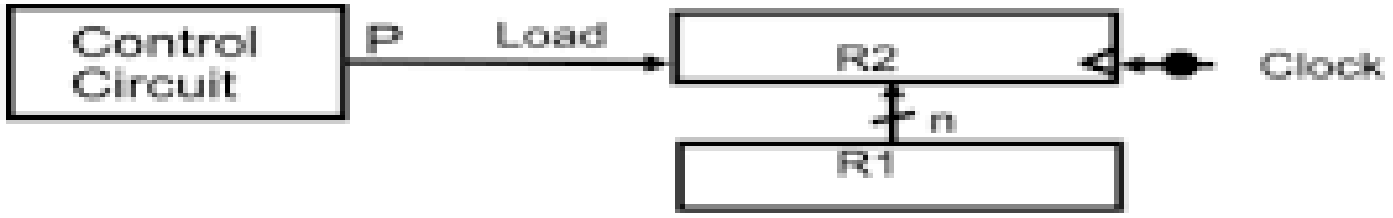
Register Transfer

- Information transfer from one register to another is designated in symbolic form by means of a replacement operator.
- *R*2 ← *R*1
- It denotes a transfer of the content of register *R*1 into register *R*2. It designates a replacement of the content of *R*2 by the content of *R*1 without changing the content of R1 after transfer.
- If the Register transfer is to occur only under a predetermined control condition, this can be shown by means of an *if-then* statement.

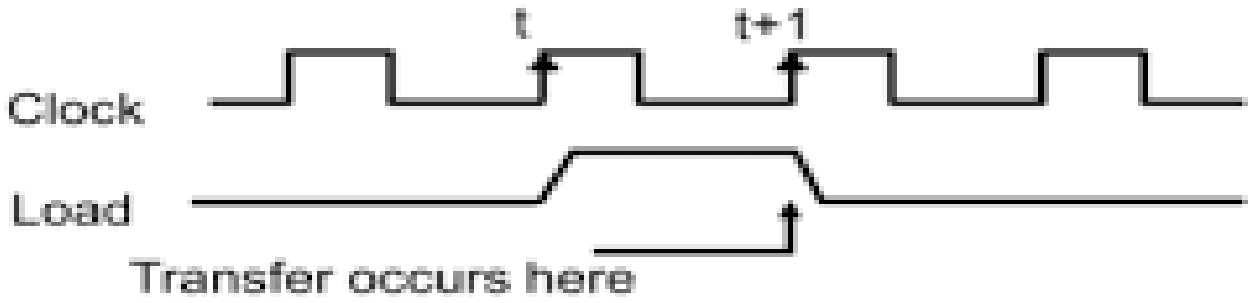
- *If* (P = 1) *then* (R2 ← R1)
- P: R2 ← R1,

where **P is a control function** that can be either 0 or 1

Register, bus and memory transfer



Transfer from R1 to R2 when P = 1



Timing diagram

Register, bus and memory transfer

Basic Symbols for Register Transfers

Symbol	Description	Examples
Letters (and numerals)	Denotes a register	*MAR*, *R2*
Parentheses ( )	Denotes a part of a register	*R2*(0-7), *R2*(*L*)
Arrow ←	Denotes transfer of information	*R2* ← *R1*
Comma ,	Separates two microoperations	*R2* ← *R1*, *R1* ← *R2*

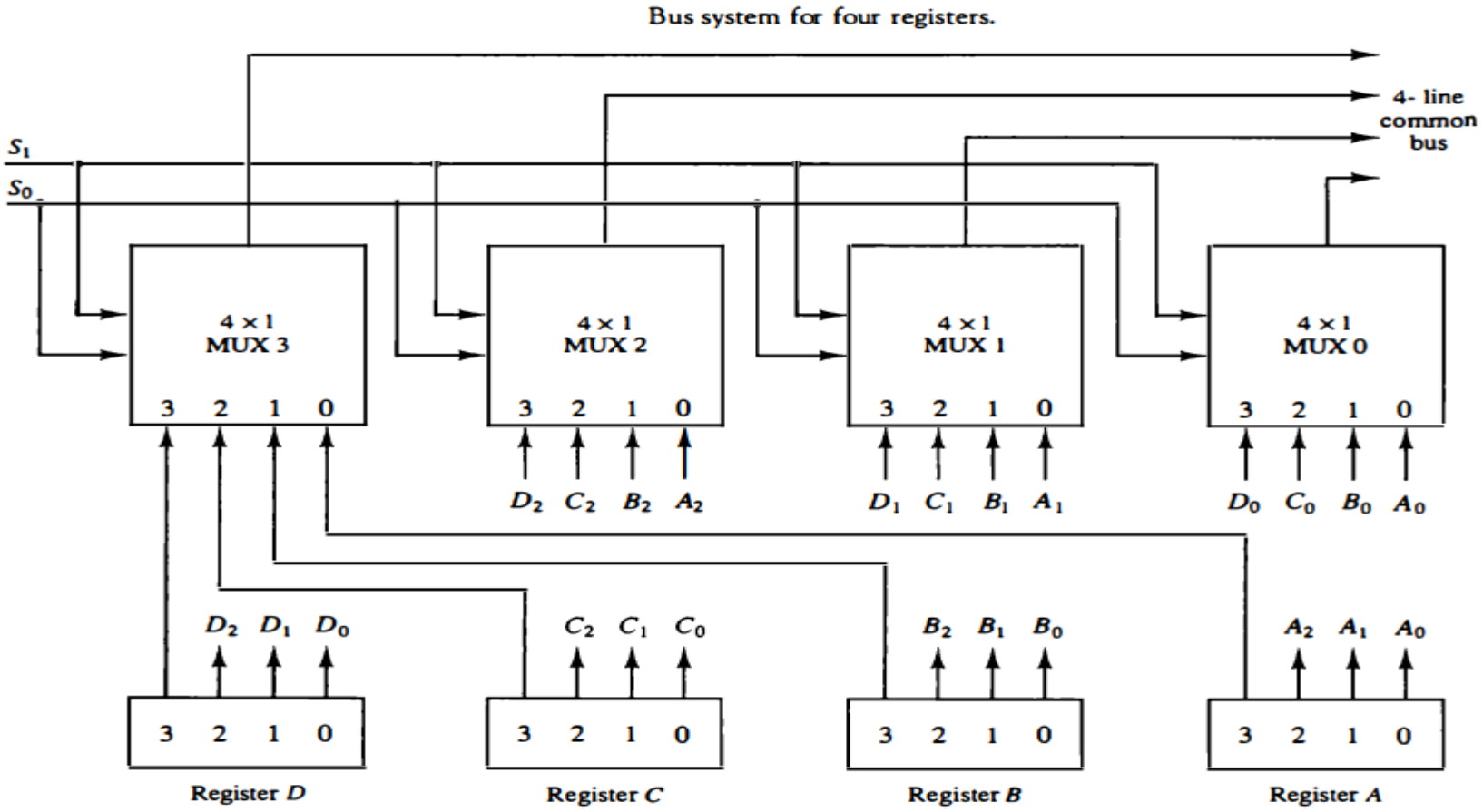
Register, bus and memory transfer

Bus transfer

- A bus structure consists of a set of common lines, one for each bit of a register.
- Control signals determine which register is selected by the bus during each transfer.
- Multiplexers can be used to construct a common bus.
- Multiplexers select the source register whose binary information is then placed on the bus.
- The select lines are connected to the selection inputs of the multiplexers and choose the bits of one register

Register, bus and memory transfer

Bus system for 4 registers



Register, bus and memory transfer

Functional table for bus

S1	S0	REGISTER SELECTED
0	0	A
0	1	B
1	0	C
1	1	D

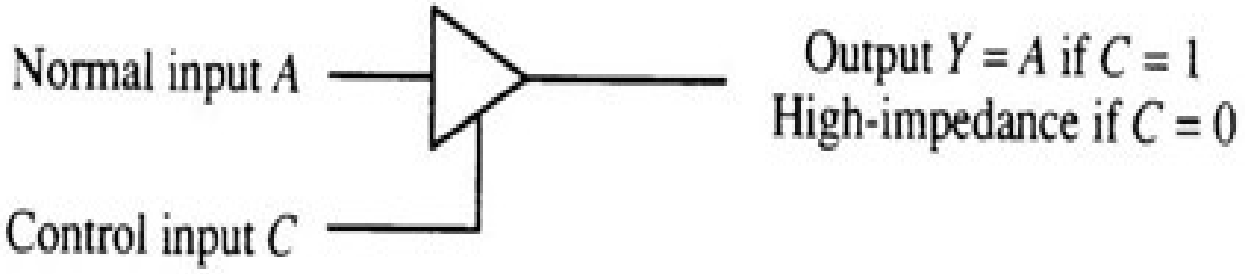
Register, bus and memory transfer

- In general, a bus system will multiplex *k* registers of *n* bits each to produce an *n*- line common bus.
- This requires *n* multiplexers – one for each bit
- The size of each multiplexer must be *k* x 1
- Transfer of information from the bus to one of many destination registers can be accomplished by connecting bus lines to the inputs of all designation registers and activating the load control of particular destination register selected.
- Symbolic statement-
- BUS ⟵ C, R1 ⟵ BUS,
- If bus is known to exist in the system, R1 ⟵ C

Bus Transfer using Three state buffer

- Instead of using multiplexers, *three-state gates* can be used to construct the bus system
- A three-state gate is a digital circuit that exhibits three states.
- Two of the states are signals equivalent to logic 1 and 0
- The third state is a *high-impedance* state – this behaves like an open circuit, which means the output is disconnected and does not have a logic significance.

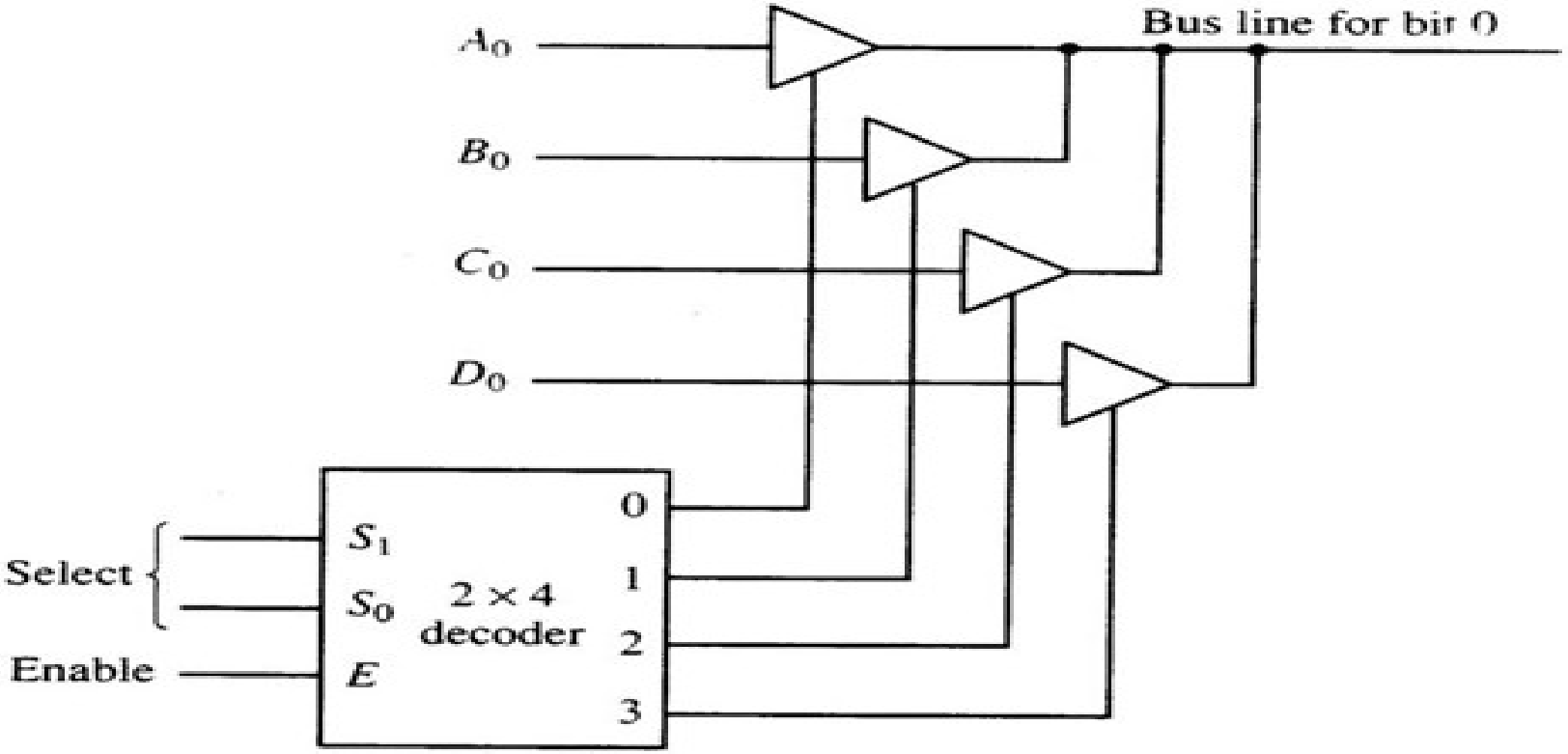
Graphic symbols for three-state buffer.



Bus Transfer using Three state buffer

- The three-state buffer gate has a normal input and a control input which determines the output state.
- With control 1, the output equals the normal input
- With control 0, the gate goes to a high-impedance state
- This enables a large number of three-state gate outputs to be connected with wires to form a common bus line without endangering loading effects

Bus Transfer using Three state buffer



Bus line with three state-buffers.

Bus Transfer using Three state buffer

- Decoders are used to ensure that no more than one control input is active at any given time
- This circuit can replace the multiplexer.
- To construct a common bus for four registers of *n* bits each using three-state buffers, we need *n* circuits with four buffers in each
- **Only one decoder** is necessary to select between the four registers

Memory transfer

Most of the standard notations used for specifying operations on memory transfer are stated below.

- The transfer of information from a memory unit to the user end is called a **Read** operation.
- A memory word is designated by the letter **M**.
- We must specify the address of memory word while writing the memory transfer operations.
- The **address register** is designated by **AR** and the **data register** by **DR**.
- Thus, a read operation can be stated as:

$$\text{Read: } DR \leftarrow M[AR]$$

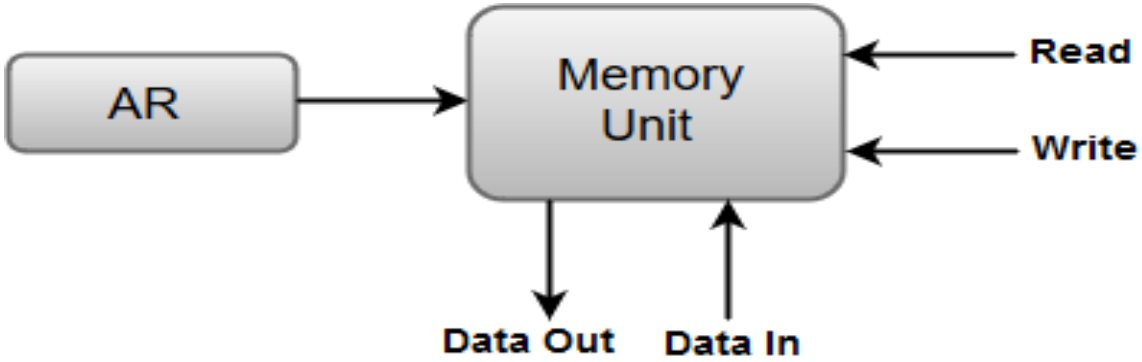
- The **Read** statement causes a transfer of information into the data register (DR) from the memory word (M) selected by the address register (AR).

Memory transfer

- The transfer of new information to be stored in the memory is called a **Write** operation.
- The Write statement causes a transfer of information from register R1 into the memory word (M) selected by address register (AR).
- Write: M [AR] ← R1

Memory transfer

Memory Transfer Block diagram



Above Diagram showing connections to memory unit.

Write: M[AR] ← DR

Read: DR ← M[AR]

Daily Quiz

1. The transfer of information from a memory unit to the user end is called a _____operation.
2. _____ are used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU.
3. A decoder converts n inputs to _____ outputs. ()
4. Which of the following are building blocks of encoders?
   a) NOT gate
   b) OR gate
   c) AND gate
   d) NAND gate
5. The transfer of new information to be stored in the memory is called a _____ operation.
6. Which of the following can be represented for decoder?
   a) Sequential circuit
   b) Combinational circuit
   c) Logical circuit
   d) None of the mentioned

Daily Quiz with Answers

1. The transfer of information from a memory unit to the user end is called a _____operation. (**Read**)
2. _____ are used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. (**Registers**)
3. A decoder converts n inputs to _____ outputs. (2^n)
4. Which of the following are building blocks of encoders?
   a) NOT gate
   b) OR gate
   c) AND gate
   d) NAND gate
5. The transfer of new information to be stored in the memory is called a _____ operation. (**Write**)
6. Which of the following a decoder represents?
   a) Sequential circuit
   b) **Combinational circuit**
   c) Logical circuit
   d) None of the mentioned

Recap

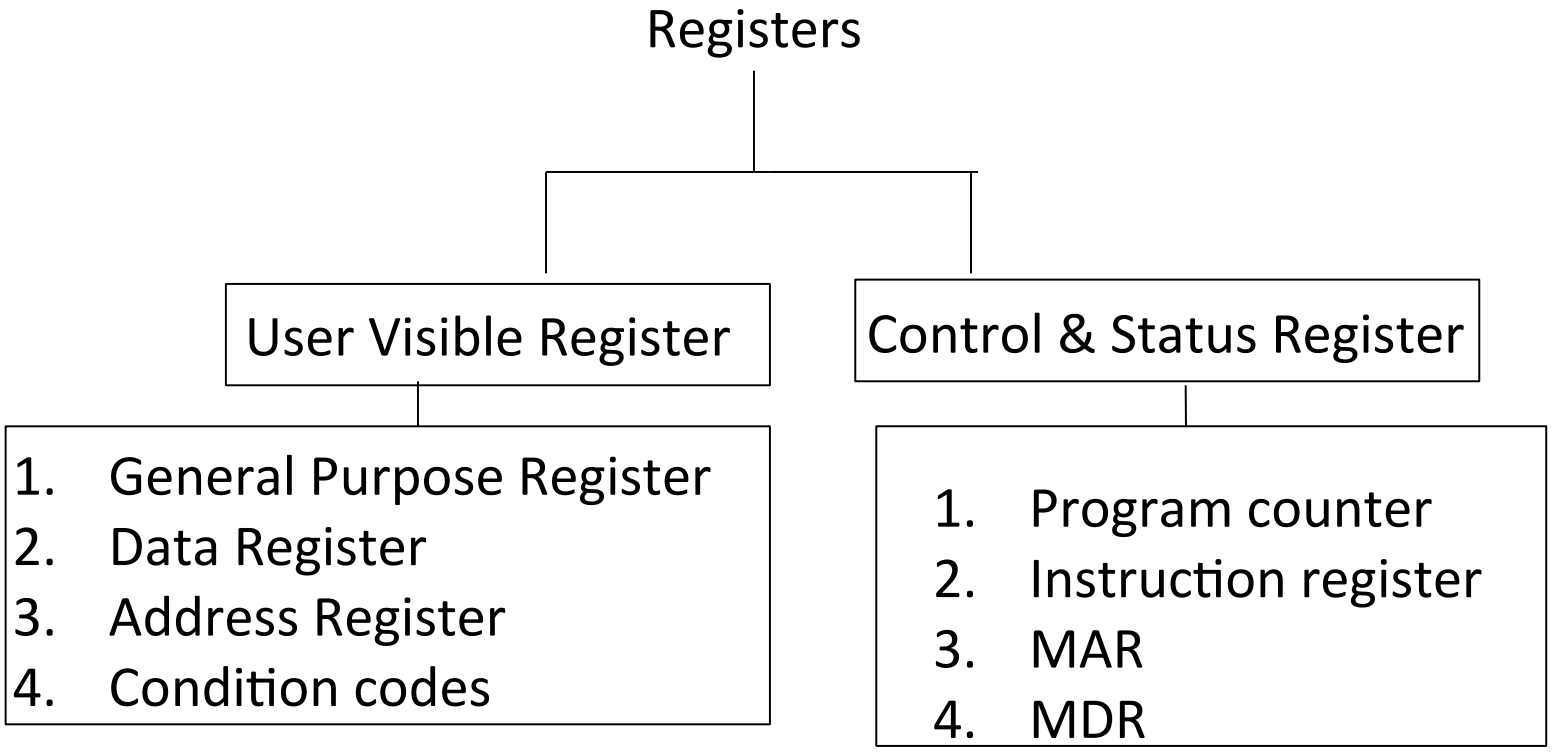
- Registers are used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU.
- Multiplexers can be used to construct a common bus.
- A three-state gate is a digital circuit that exhibits three states – 0,1 and high impedance state.
- The transfer of information from a memory unit to the user end is called a **Read** operation.
- The transfer of new information to be stored in the memory is called a **Write** operation.

Introduction to Topic 2

Name of Topic	Objective of Topic	Mapping with CO
Processor organization, General register organization and stack organization	Students will be able to know various process organization and data stored in register and stack	CO 1

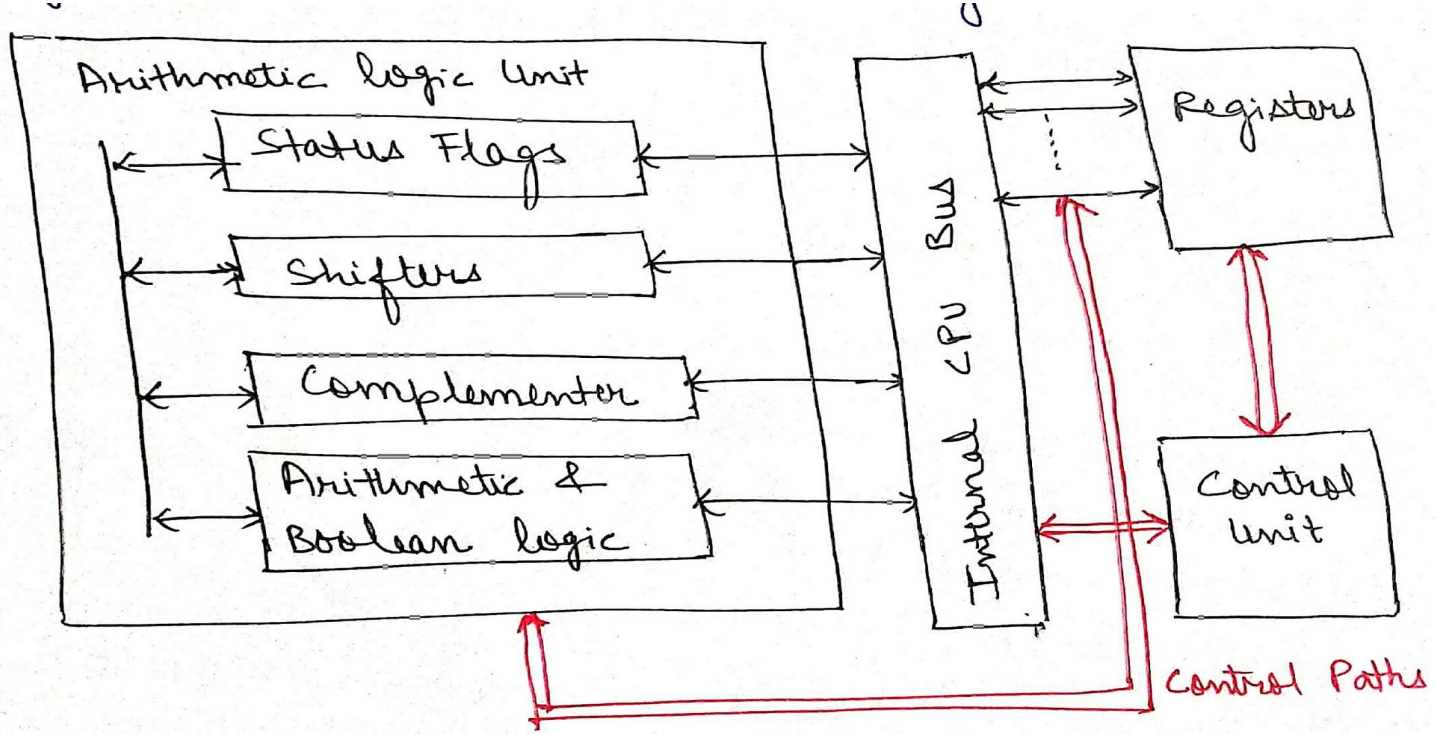
Registers

- Registers are used to store data temporarily.



Processor organization

- Processor organization means how the components of processor are connected and accomplish their task.



Scanned with CamScanner

fig:- Internal structure of CPU

Processor organization-General register organization

Most computers fall into one of three types of CPU organizations:

- Single accumulator organization.
- General register organization.
- Stack organization.

1.Single accumulator organization

- The instruction format in this type of computer uses one address field.
- All operations are performed with an implied accumulator register.
- **Example :** ADD X

- where X is the address of the operand. The ADD instruction in this case results in the operation AC <--AC + M[X].
- AC is the accumulator register and M[X] symbolizes the memory word located at address X.

Processor organization

2. General register organization

- The instruction format in this type of computer needs three register address fields.
- ADD R1, R2, R3 to denote the operation R 1 <---R2 + R3 .
- ADD R1, R2, would denote the operation R 1 <---R1 + R2. Only register addresses for R 1 and R2 need be specified in this instruction.
- General register-type computers employ two or three address fields in their instruction format.
- Each address field may specify a processor register or a memory word.
- ADD R1, X
  R1 ←---- R1 + M [X]

Processor organization

3. Stack organization

- The stack-organized CPU ,Computers with stack organization would have PUSH and POP instructions which require an address field. Thus the instruction
- PUSH X
  It will push the word at address X to the top of the stack. Stack pointer is updated automatically.

- ADD
  This instruction in stack computer consists of an operation code only with no address field.

General register organization

General registers organization

•In this type of organization, computer uses two or three address fields in their instruction format.

•Each address field may specify a general register or a memory word. If many CPU registers are available for heavily used variables and intermediate results

For example:

•MULT R1, R2, R3

•This is an instruction of an arithmetic multiplication written in assembly language. It uses three address fields R1, R2 and R3.

General register organization

- The meaning of this instruction is:

  R1 <-- R2 * R3

- This instruction also can be written using only two address fields as:

  MULT R1, R2

- In this instruction, the destination register is the same as one of the source registers. This means the operation

  R1 <-- R1 * R2

- The use of large number of registers results in short program with limited instructions.

General register organization

The advantages of General register based CPU organization

•Efficiency of CPU increases as there are large number of registers are used in this organization.

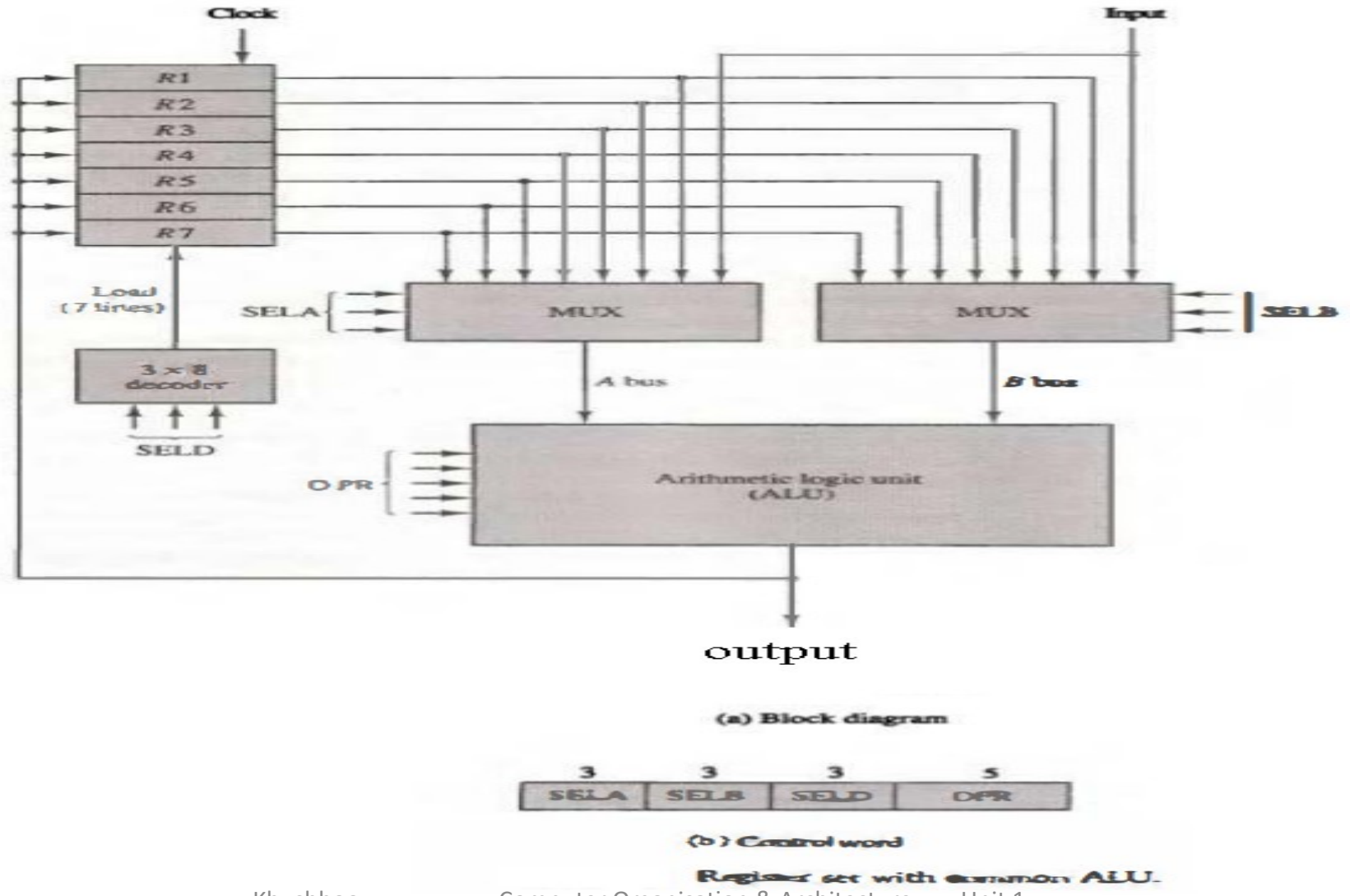
•Less memory space is used to store the program since the instructions are written in compact way.

The disadvantages of General register based CPU organization

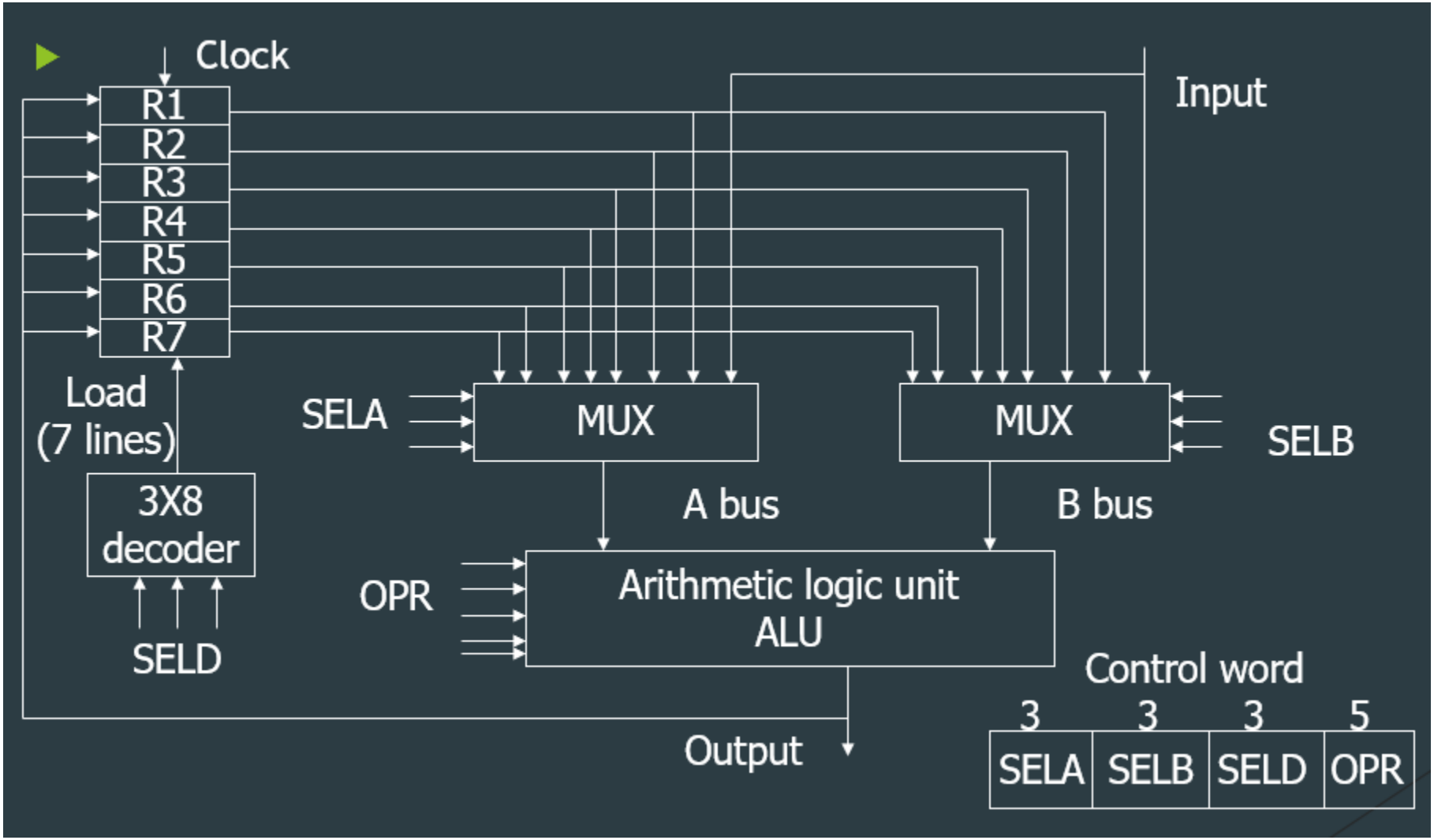
•Care should be taken to avoid unnecessary usage of registers. Thus, compilers need to be more intelligent in this aspect.

•Since large number of registers are used, thus extra cost is required in this organization.

General register organization



General register organization



Encoding of Register Selection Fields

Binary Code	SELA	SELB	SELD
000	INPUT	INPUT	NONE
001	R1	R1	R1
010	R2	R2	R2
011	R3	R3	R3
100	R4	R4	R4
101	R5	R5	R5
110	R6	R6	R6
111	R7	R7	R7

Encoding of ALU Operations

OPR Select	Operation	Symbol
00000	Transfer A	TSFA
00001	Increment A	INCA
00010	Add A + B	ADD
00101	Subtract A - B	SUB
00110	Decrement A	DECA
01000	AND A and B	AND

Example of Microoperations

Subtract microoperation:

R1 ←----- R2 – R3

Binary control word for subtract microoperation-

Field:	SELA	SELB	SELD	OPR
Symbol:	R2	R3	R1	SUB
Control word:	010	011	001	00101

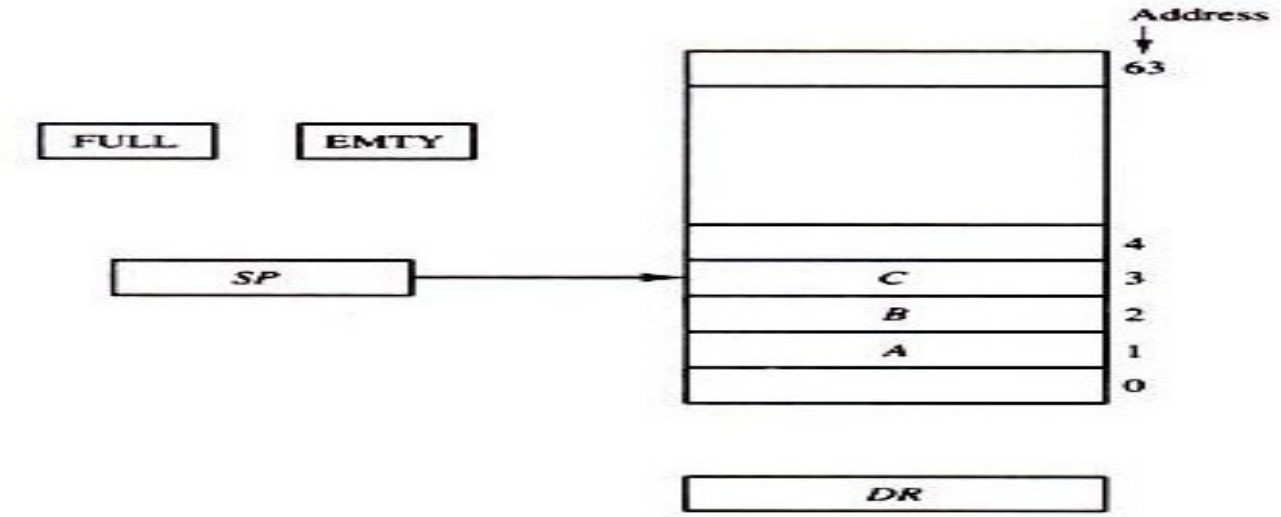
General register organization

- The control unit that operates the CPU bus system directs the information flow through the registers and ALU by selecting the various components in the system. For example, to perform the operation
- R1 <--R2 + R3. The control must provide binary selection variables to the following selector inputs:
- 1. MUX A selector (SELA): to place the content of R2 into bus A .
- 2 . MUX B selector (SELB): to place the content o f R 3 into bus B .
- 3 . ALU operation selector (OPR): to provide the arithmetic addition
- A+B.
- Decoder destination selector (SELD): to transfer the content of the output bus into R 1 .

Stack Organization

- A stack is a storage device that stores information in such a manner that the item stored last is the first item retrieved.
- The stack in digital computers is essentially a memory unit with an address register that can count only. The register that holds the address for the stack is called a stack pointer (SP) because its value always points at the top item in the stack.
- The physical registers of a stack are always available for reading or writing. It is the content of the word that is inserted or deleted.

Register stack:



Stack Organization

PUSH:

•If the stack is not full (FULL =0), a new item is inserted with a push operation. The push operation consists of the following sequences of micro operations:

SP ← SP + 1 Increment stack pointer

M [SP] ← DR WRITE ITEM ON TOP OF THE STACK

IF (SP = 0) then (FULL ← 1) Check is stack is full

EMTY ← 0 Mark the stack not empty

Stack Organization

POP:

•A new item is deleted from the stack if the stack is not empty (if EMTY = 0). The pop operation consists of the following sequences of micro operations:

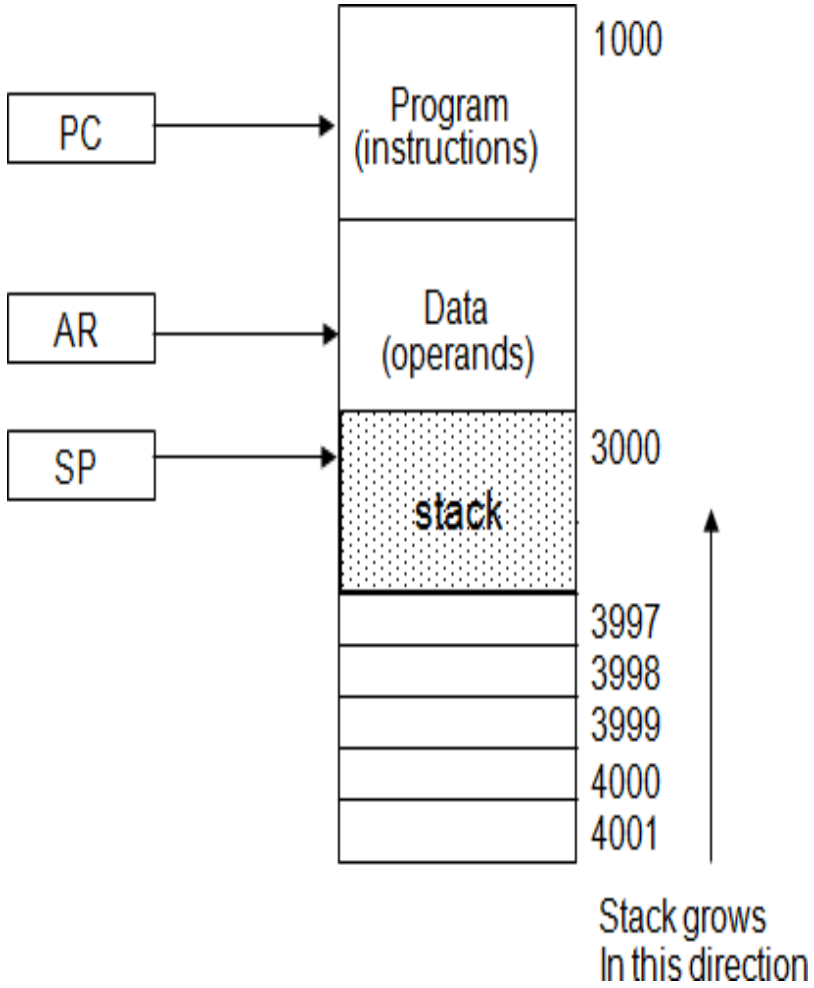
DR ← M [SP] Read item on top of the stack
SP ← SP - 1 Decrement stack pointer
IF (SP = 0) then (EMTY ← 1) Check if stack is empty
FULL ← 0 Mark the stack not full

•The top item is read from the stack into DR. The stack pointer is then decremented. If its value reaches zero, the stack is empty, so EMTY is set to 1.

Stack Organization

Memory Stack

- The implementation of a stack in the CPU is done by assigning a portion of memory to a stack operation and using a processor register as a stack pointer.



Stack Organization

Memory Stack

- The program counter PC points at the address of the next instruction in the program which is used during the fetch phase to read an instruction.
- The address registers AR points at an array of data which is used during the execute phase to read an operand.
- The stack pointer SP points at the top of the stack which is used to push or pop items into or from the stack.
- The three registers are connected to a common address bus, and either one can provide an address for memory.

Stack Organization

PUSH

- A new item is inserted with the push operation as follows:

$$SP \leftarrow SP - 1$$
$$M[SP] \leftarrow DR$$

- The stack pointer is decremented so that it points at the address of the next word.
- A memory write operation inserts the word from DR into the top of the stack.

POP

- A new item is deleted with a pop operation as follows:

$$DR \leftarrow M[SP]$$
$$SP \leftarrow SP + 1$$

Stack Organization

Polish Notation

- A stack organization is very effective for evaluating arithmetic expressions. The common arithmetic expressions are written in infix notation, with each operator written between the operands.

- Consider the simple arithmetic expression

A•B + C•D

A + B Infix notation
+AB Prefix or Polish notation
AB+ Postfix or reverse Polish notation (RPN)

- The reverse Polish notation is in a form suitable for stack manipulation.
- The expression A•B + C•D is written in reverse Polish notation as AB•CD•+

Introduction to Topic 2

Name of Topic	Objective of Topic	Mapping with CO
Addressing Modes	Students will be able to know the different addressing modes.	CO 1

Addressing Modes

- The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced.
- The decoding step in the instruction cycle determines the operation to be performed, the addressing mode of the instruction, and the location of the operands

Addressing Modes

1. **Implied mode**:

- The operands are specified implicitly in the definition of the instruction – complement accumulator or zero-address instructions.
- CMA , CLC

2. **Immediate mode**:

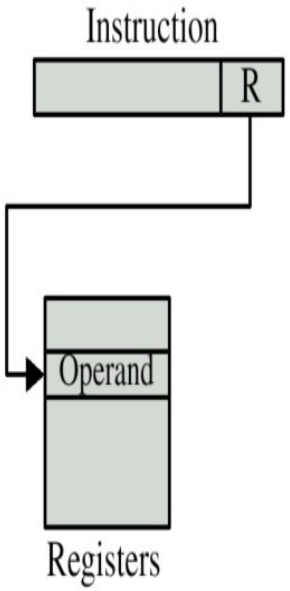
- The operands value are specified in the instruction.
- Immediate mode instructions is said to be useful for initializing registers to a constant value.
- MOV AL, 35 H

3. **Register mode**:

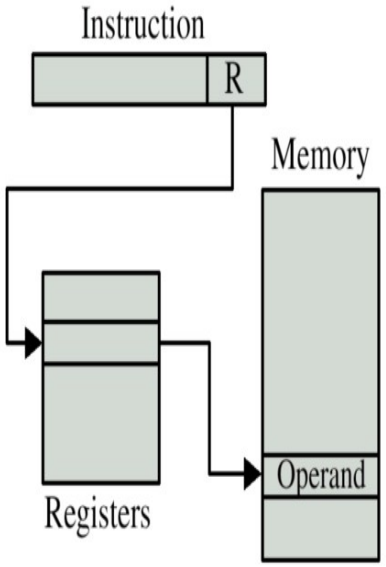
- The operands are in registers and the register is present in CPU.
- The data is in the register that is specified by the instruction.
- MOV A,C (move the content of C register to A register)

Addressing Modes

Register Addressing



Register Indirect Addressing



Addressing Modes

4. Register Indirect mode:

- In this mode the instruction specifies a register in the CPU whose contents give the address of the operand in memory.
- The selected register contains the address of the operand rather than the operand itself.
- MOV AX,[CX] (move the content of memory location addressed by the register CX to the register AX)

5. Autoincrement/Autodecrement mode:

- This is similar to register indirect mode except that the register is incremented or decremented after or before its value is used to access memory.

a) Increment mode- After accessing the operand the contents of this register are automatically incremented to point to next consecutive memory location.

Addressing Modes

Example: Add R1, (R2+)
OR

R1 = R1 + M[R2]
R2 = R2 + d

Useful for stepping through arrays in a loop.
R2 – start of the array, d- size of an element.

b)Decrement Mode:
Before accessing the operand , the contents of this register are automatically decremented to point to the previous consecutive memory location.
Example- Add R1, (-R2)
OR

R2 = R2 – d
R1 = R1 + M [R2]

Addressing Modes

6. Direct Address mode:

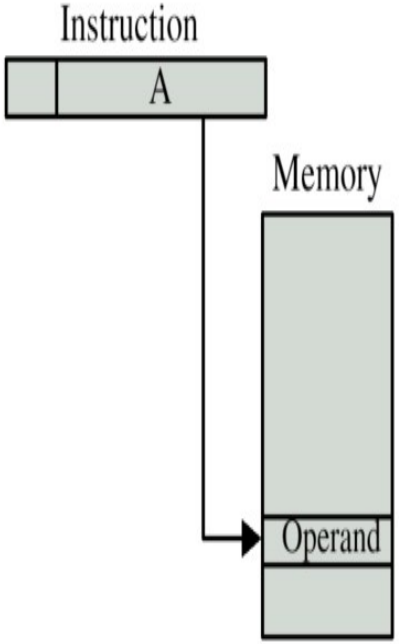
- In this mode the effective address is equal to the address part of the instruction.
- The operand resides in memory and its address is given directly by the address field of the instruction.
- Example – ADD AL, [0301] (add the content of address 0301 to A)

7. Indirect Address mode :

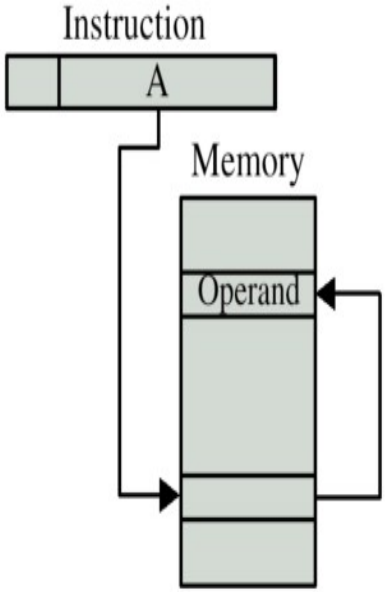
- In this mode the address field of the instruction gives the address where the effective address is stored in memory.
- In this address field of instruction gives the address where the effective address is stored in memory.

Addressing Modes

Direct Addressing



Indirect Addressing



Addressing Modes

- **Relative address mode:** the effective address is the summation of the address field and the content of the PC
- **Indexed addressing mode:** the effective address is the summation of an index register and the address field
- **Base register address mode:** the effective address is the summation of a base register and the address field

- **effective address = address part of instruction + content of CPU register**

Summary

In previous slides we discuss in details

- Functional units of digital system & their interconnections
- Buses, types of buses, bus architecture and bus arbitration.
- Register, bus and memory transfer.
- Processor organization, general registers organization, stack organization
- Addressing modes