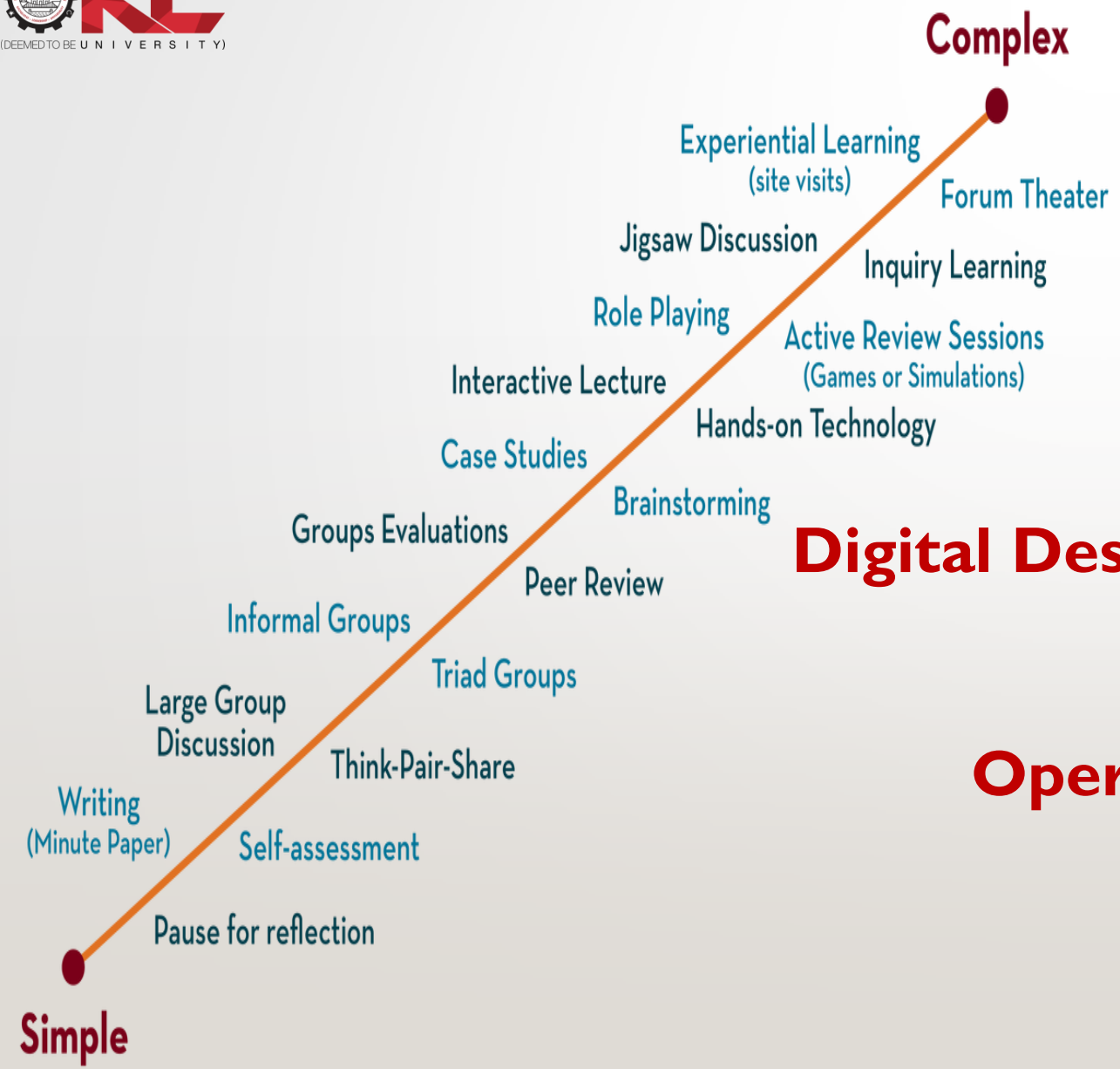


Department of BES-II

Digital Design and Computer Architecture
23EC1202

Topic:

Operands, Instruction formats, Addressing modes

Session No: 23

AIM OF THE SESSION

To familiarize students with the basic concept of Operands, Addressing modes, Instruction formats.

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Demonstrate the basic concept of operands.
2. Describe Addressing modes.
3. List out the Operands, Addressing modes, and Instruction formats.
4. Describe the Instruction formats

LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Define the basic concept of operands.
2. Describe Addressing modes.
3. Summarize the Operands, Addressing modes, and Instruction formats.

Instruction Formats

- Instruction format refers to the layout or structure of a binary instruction in machine language.
- The instruction format is crucial for understanding how a computer operates internally.
- There are several components to an instruction, and different architectures may have different formats.
- Common elements include the **opcode (operation code), operand(s)** and sometimes a specifier for the instruction length or additional operation specifics.

Operands

Definition

Operands are values used in arithmetic or logical operations.

Common types include integers, floating point numbers, and booleans.

Example

In the operation x = y + z,
y and z are the operands. The computer retrieves their values from memory, performs the addition, and stores the result in x.

Significance

Choosing the right operand type can greatly impact program speed and efficiency. Understanding operand types is key to optimizing program performance.

Instruction Formats

1. Zero Address Instruction

2. One Address Instruction

3. Two Address Instruction

4. Three Address Instruction

Zero Address Instructions

PUSH	A	TOP = A
PUSH	B	TOP = B
ADD		TOP = A+B
PUSH	C	TOP = C
PUSH	D	TOP = D
ADD		TOP = C+D
MUL		TOP = (C+D)*(A+B)
POP X		M[X] = TOP

One Address Instructions

LOAD	A	A C <- M [A]
ADD	B	A C <- A C + M [B]
STORE	T	M [T] <- A C
LOAD	C	A C <- M [C]
ADD	D	A C <- A C + M [D]
MUL	T	A C <- A C • M [T]
STORE	X	M [X] <- A C

Two Address Instructions

MOV R1 ,A	R 1 <-- M [A]
ADD R1 , B	R 1 <-- R 1 + M [B]
MOV R2 , C	R 2 <-- M [C]
ADD R2 , D	R 2 <-- R 2 + M [D]
MUL R1 , R2	R 1 <-- R 1 * R 2
MOV X , R1	M [X] <-- R 1

Three Address Instructions

ADD R1 , A , B	R 1 <-- M [A] + M [B]
ADD R2 , C , D	R 2 <-- M [C] + M [D]
MUL X , R1 , R2	M [X] <-- R 1 * R 2

Addressing Modes

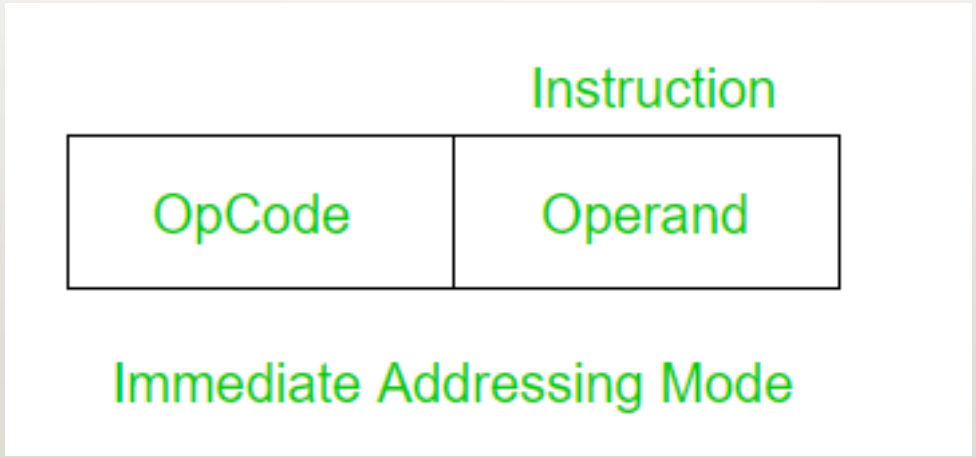
- ✓ The term addressing modes refers to the way in which the operand of an instruction is specified.
- ✓ The most common addressing techniques are:
  - **Immediate**
  - **Direct**
  - **Indirect**
  - **Register**
  - **Register Indirect**
  - **Displacement**

Immediate Addressing Mode

- ✓ In Immediate addressing mode the source operand is always data.

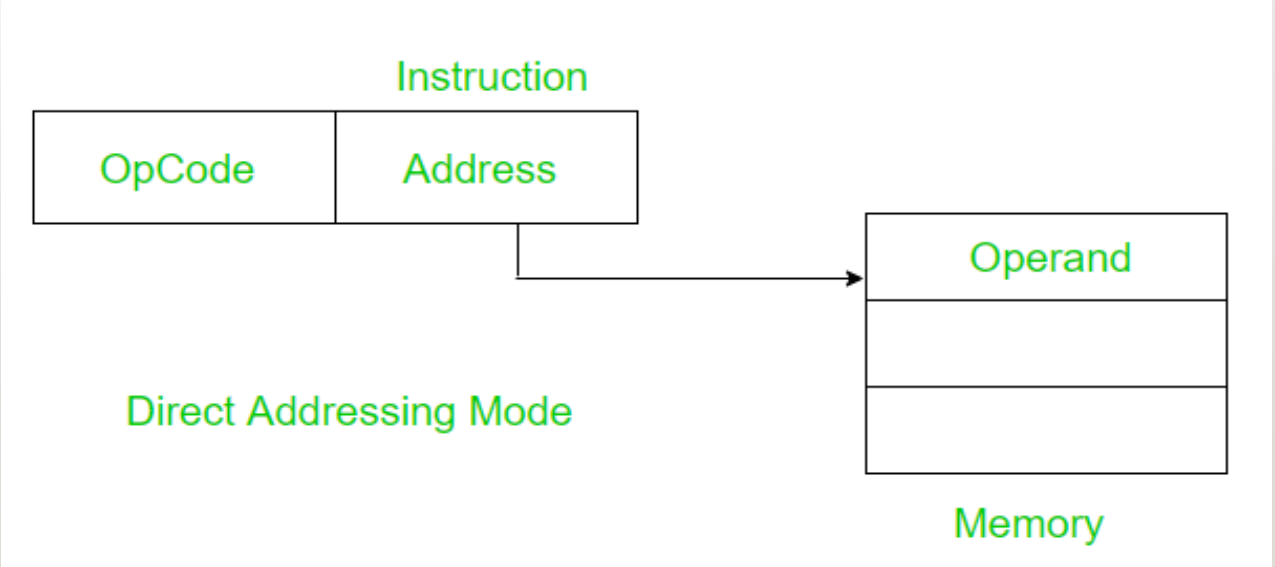
Eg: MOV AX, 2000H

ADD AL,45H



Direct Addressing Mode

- ✓ A very simple form of addressing is direct addressing, in which the address field contains the effective address of the operand: EA = A
- ✓ It requires only one memory reference and no special calculation.
- ✓ Eg: MOV AX, [1592H]
  MOV BL, [0300H]

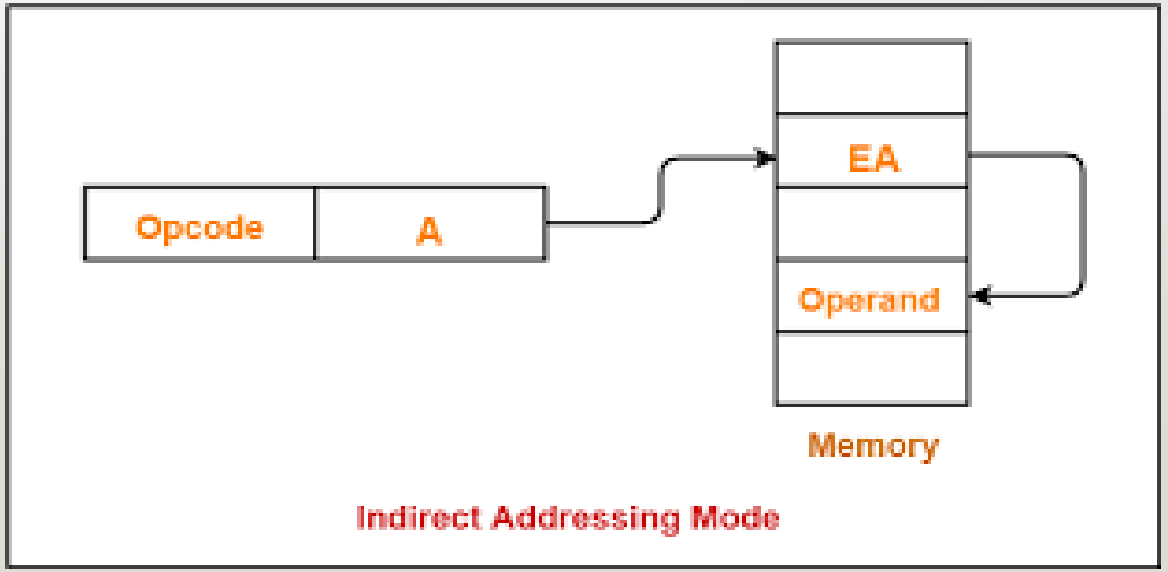


Indirect Addressing Mode

- ✓ In Direct Addressing, the length of the address field is usually less than the word length, thus limiting the address range. One solution is to have the address field refer to the address of a word in memory, which in turn contains a full-length address of the operand. This is known as indirect addressing.

  EA = (A)

- ✓ Eg: MOV AX, @2005H
  LOAD R1, (1345H)



Indirect Addressing Mode

Register Addressing Mode

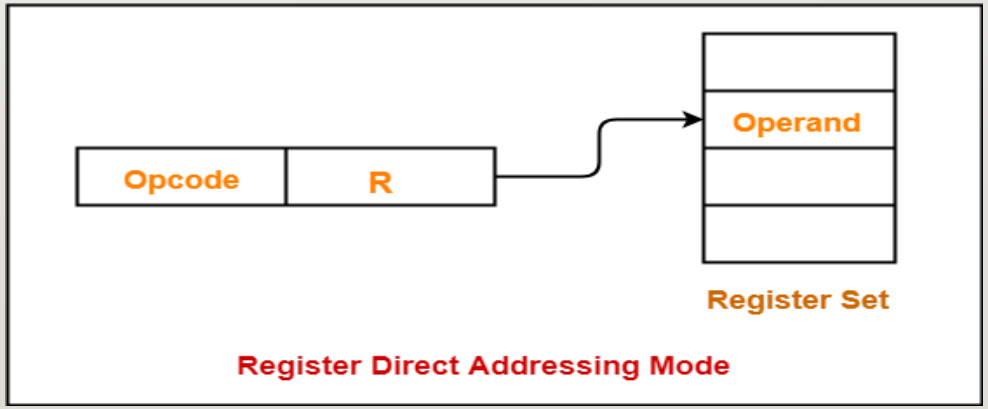
- Register addressing is similar to direct addressing. The only difference is that the address field refers to a register rather than a main memory address.

  EA = R

- The advantages of register addressing are that only a small address field is needed in the instruction and no memory reference is required. The disadvantage of register addressing is that the address space is very limited.

  Eg: MOV AX,BX

  ADD R1,R2

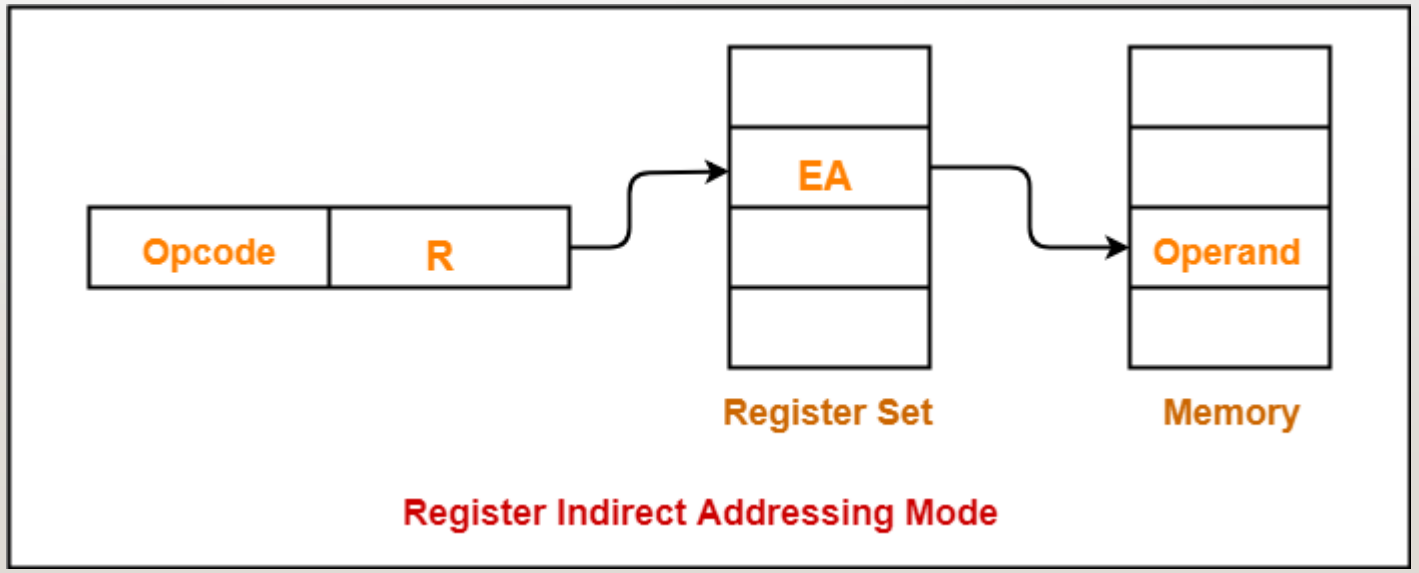


Register Indirect Addressing Mode

- Register indirect addressing is similar to indirect addressing, except that the address field refers to a register instead of a memory location.
- It requires only one memory reference and no special calculation.

EA = (R)

Eg: ADD AL, [BX]
MOV AX, [BX]



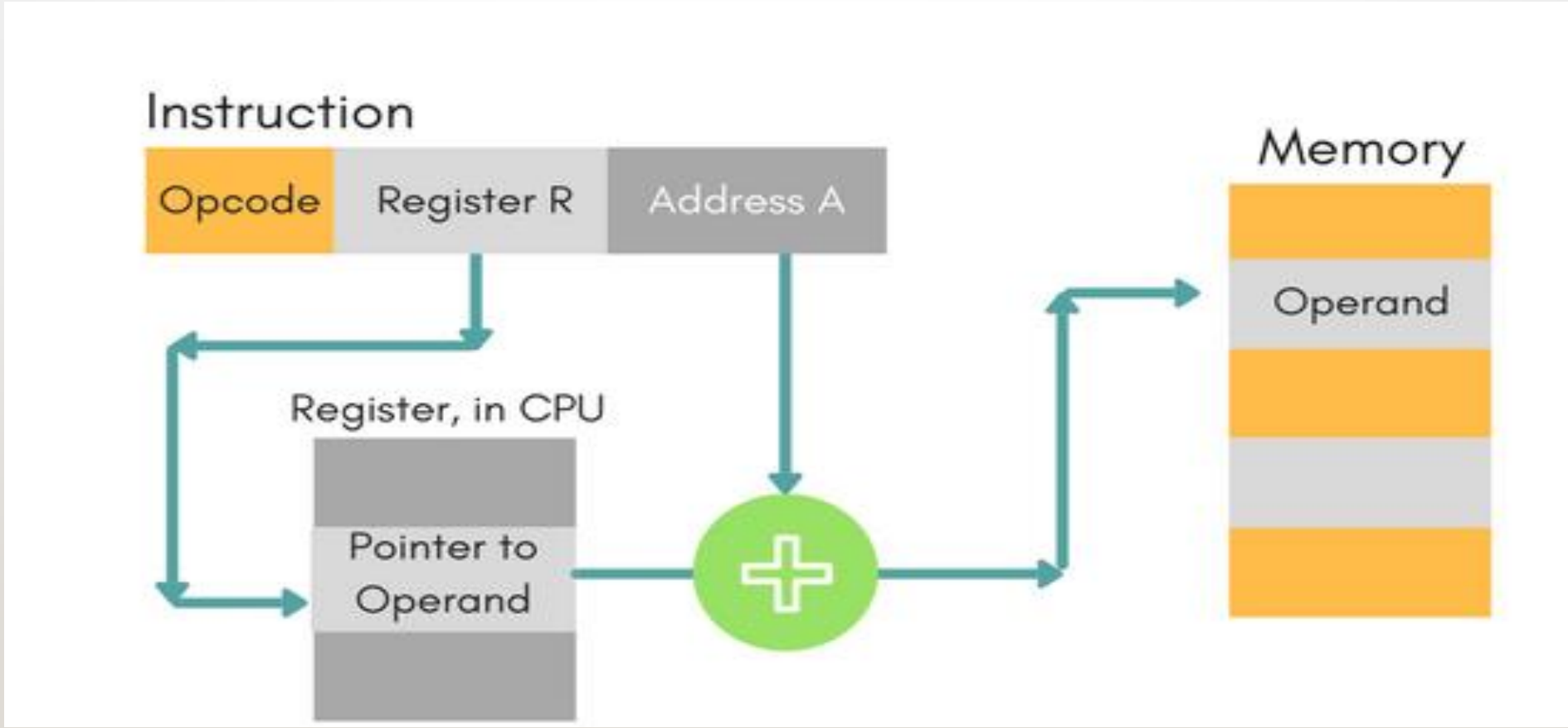
Displacement Addressing Mode

- A very powerful mode of addressing combines the capabilities of direct addressing and register indirect addressing, which is broadly categorized as displacement addressing.

$$EA = A + (R)$$

- Displacement addressing requires that the instruction have two address fields, at least one of which is explicit. The value contained in one address field (value = A) is used directly. The other address field, or an implicit reference based on opcode, refers to a register whose contents are added to A to produce the effective address.

Displacement Addressing Mode



SELF-ASSESSMENT QUESTIONS

1. What is an addressing mode in computer architecture?

A. A configuration for the CPU's power management.
B. A method for organizing instruction sets.
C. The way in which the operand of an instruction is specified.
D. The mode in which the CPU addresses the control unit.

2. In immediate addressing mode, where is the operand located?

A. In a CPU register.
B. In the instruction itself.
C. In the main memory.
D. In a cache memory.

SELF-ASSESSMENT QUESTIONS

3. What is an instruction format in computer architecture?

A. A method to store data in memory.
B. A set of rules that defines how an instruction is encoded.
C. The algorithm used by the CPU to execute instructions.
D. The layout of the CPU's control unit.

4. Which component of an instruction specifies the operation to be performed?

A. Operand
B. Opcode
C. Address field
D. Flag register

TERMINAL QUESTIONS

Short answer questions:

1. Provide the common types of operands with an example.

Long answer questions:

1. Examine the concepts of immediate, direct, and indirect addressing modes in computer architecture.
2. Elaborate on various instruction formats with structure and example.
3. Examine the concepts of register, register indirect, displacement addressing modes in computer architecture.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books:

1. Computer Organization by Carl Hamacher, Zvonko Vranesic and Saftwat Zaky.
2. Computer System Architecture by M. Morris Mano
3. Computer Organization and Architecture by William Stallings

Sites and Web links:

1. https://www.geeksforgeeks.org/addressing-modes/
2. https://www.geeksforgeeks.org/computer-organization-instruction-formats-zero-one-two-three-address-instruction/

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – Digital Design & Computer Architecture