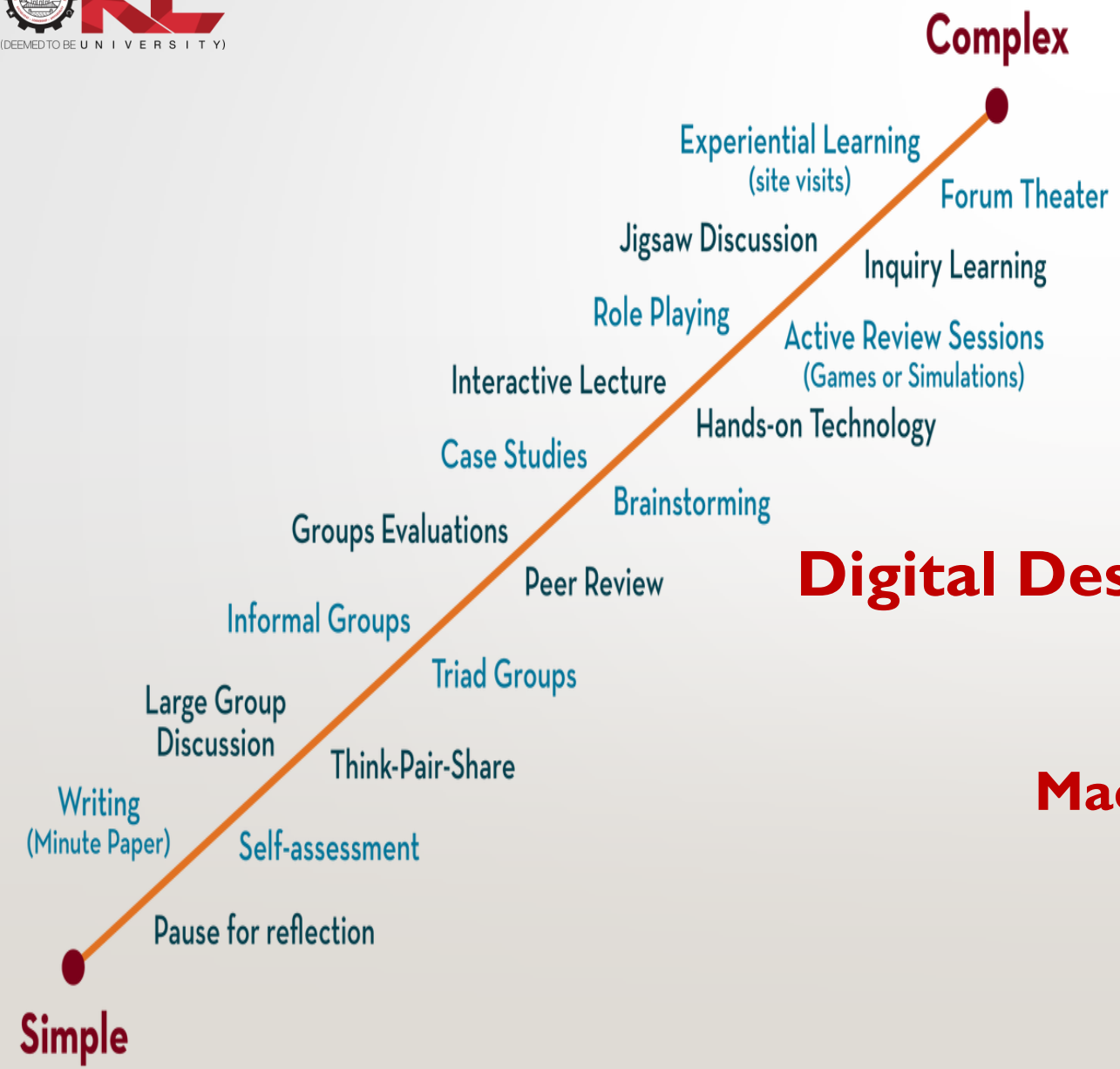


Department of BES-II

Digital Design and Computer Architecture

23EC1202

Topic:

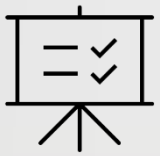
Machine Cycle, Instruction Sets

Session No: 24

AIM OF THE SESSION

To familiarize students with the basic concept of instruction set and machine cycle in the context of computer architecture.

INSTRUCTIONAL OBJECTIVES



This Session is designed to:

1. Demonstrate to decode and understand simple instructions sets.
2. Describe instructions related to data movement, such as load and store operations, and understand their role in transferring data between memory and registers.
3. List out the Steps in machine cycle.

LEARNING OUTCOMES

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

1. Define the concept of an instruction set.
2. Describe how instructions are fetched, decoded, and executed within a processor.
3. Summarize the machine cycle process.

Instruction Set

An **instruction set** is a list of all the instructions that a processor can execute.

Types of Operations

1. Data Transfer
2. Arithmetic
3. Logical
4. Input Output [I/O]
5. System Control
6. Transfer Control

Format:

Mnemonic Destination, Source

- The destination can be a register within the CPU, a memory address where the result is to be stored.
- The source can be an immediate value (a direct numerical value), a register, or a memory address containing the data to be used in the operation.

Data Transfer - Instruction Set

- **Move (Transfer)** :Transfer word or block from source to destination

  Example: mov A, 09h

  mov AX, BX

- **Store**: Transfer word from processor to memory

  Example: STR T

  STR [R1], R3

- **Load (fetch)**: Transfer word from memory to processor.

  Example: Load A, [R1]

  Load C

- **Exchange**: Swap contents of source and destination

  Example: xchg ah, al

  xchg ax, bx

- **Clear (reset)**: Transfer word of 0s to destination

  Example: clr ax

  clr a

- **Set**: Transfer word of 1s to destination

  Example: set A

Arithmetic - Instruction Set

- **Add:** Compute sum of two operands

  Example: add al, 07h

  add ax, bx

- **Subtract:** Compute difference of two operands

  Example: sub ah, 05h

  sub ah, al

- **Multiply:** Compute product of two operands

  Example: mov ax,1234h

  mov bx,100h

  mul bx

- **Divide:** Compute quotient of two operands

  Example: mov ax,8003h

  mov cx,100h

  div cx

- **Negate:** Change sign of operand

  Example: neg RT, RA

- **Increment :** Add 1 to operand

  Example: inc A

- **Decrement:** Subtract 1 from operand

  Example: dec A

Logical - Instruction Set

AND: Performs the logical operation AND bitwise

Example: AND AL, 0Fh

AND AL, 01h

OR: Performs the logical operation OR bitwise

Example: OR AH, 0Bh

OR AH, 05h

NOT: Performs the logical operation NOT bitwise

Example: Operand1: 0101 0011

After NOT -> Operand1: 1010 1100

Exclusive OR: Performs the Exclusive-OR bitwise

Example: Operand1: 0101, Operand2: 0011

After XOR -> Operand1: 0110

Shift: Left (right) shift operand, introducing constant at end

Example: SHR AX, 2

SHL AX, 2

Rotate: Left (right) shift operation, with wraparound end

Example: ROR AH, 4

ROL AH, 4

Input / Output - Instruction Set

- **Input (Read):** Transfer data from specified I/O port or device to destination (e.g., main memory or processor register)
- **Output (Write):** Transfer data from specified source to I/O port or device.
- **Start I/O:** Transfer instructions to I/O processor to initiate I/O operation.
- **Test I/O:** Transfer status information from I/O system to specified destination

System Control - Instruction Set

- System control instructions are those which are used for system setting and it can be used only in privileged state.
- Typically, these instructions are reserved for the use of operating systems.

Transfer of Control - Instruction Set

The most common transfer-of-control operations found in instruction set are:

1. Branch
2. Skip
3. Procedure call.

BRP X: Branch to location X if result is positive

BRN X: Branch to location X if result is negative

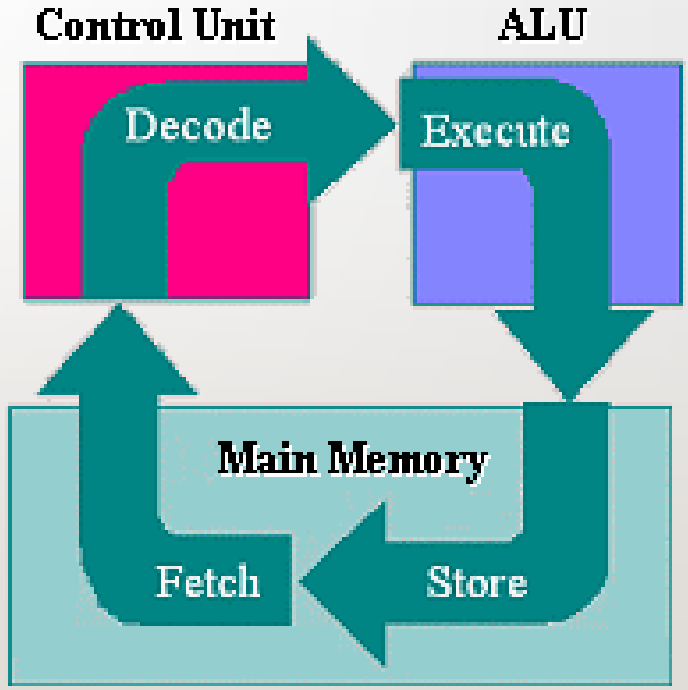
BRZ X: Branch to location X is result is zero

BRO X: Branch to location X if overflow occurs

Machine Cycle

It is the sequence of steps a computer's central processing unit (CPU) takes to process instructions and perform tasks.

The CPU follows a series of stages in order to retrieve, decode, and carry out a command.



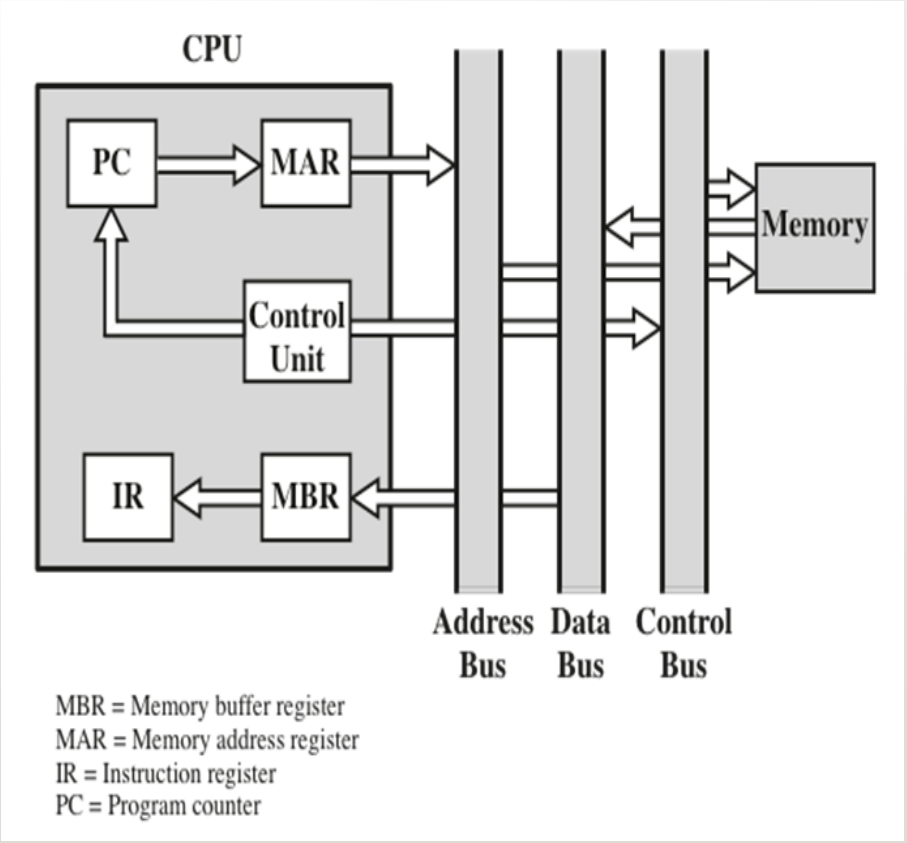
Machine Cycle

There are multiple phases to a typical machine cycle:

- **Fetch Phase:** The CPU retrieves the instruction from memory. The address of the instruction to be fetched is determined by the program counter.
- **Decode Phase:** The fetched instruction is decoded to determine the operation to be performed and the operands involved.
- **Execute Phase:** The actual operation specified by the instruction is carried out. This could involve arithmetic or logical operations, data manipulation, or control transfer.

Fetch Phase

- During the fetch cycle, an instruction is read from memory. Figure shows the flow of data during this cycle.
- The PC contains the address of the next instruction to be fetched. This address is moved to the MAR and placed on the address bus.
- The control unit requests a memory read, and the result is placed on the data bus and copied into the MBR and then moved to the IR.
- Meanwhile, the PC is incremented by 1, preparatory for the next fetch.



Fetch Phase

The illustrated fetch cycle above can be summarized by the following points:

PC => MAR

MAR => memory => MBR

MBR => IR

PC=PC+1

After the CPU has finished fetching an instruction, the CU checks the contents of the IR and determines which type of execution is to be carried out next. This process is known as the decoding phase. The instruction is now ready for the execution cycle.

Decode Phase

The fetched instruction is interpreted or decoded to identify the operation to be carried out and the operands involved.

- Instruction Decoding
- Operand Extraction
- Control Signal Generation
- Address Calculation (if applicable)

Execute Phase

Once an instruction has been loaded into the instruction register (IR), and the control unit (CU) has examined and decoded the fetched instruction and determined the required course of action to take, the execution cycle can commence.

The actions within the execution cycle can be categorized into the following four groups:

- CPU - Memory: Data may be transferred from memory to the CPU or from the CPU to memory.
- CPU - I/O: Data may be transferred from an I/O module to the CPU or from the CPU to an I/O module.
- Data Processing: The CPU may perform some arithmetic or logic operation on data via the arithmetic-logic unit (ALU).
- Control: An instruction may specify that the sequence of operation may be altered.

SELF-ASSESSMENT QUESTIONS

1. Which type of instruction involves transferring data between memory and registers?

(a). Control transfer instructions
(b). Arithmetic instructions
(c). Data movement instructions
(d). Logic instructions

2. What is a common trade-off when designing an instruction set?

(a). Execution speed vs. code size
(b). Memory size vs. processor speed
(c). Number of registers vs. cache size
(d). Pipelining vs. parallel processing

SELF-ASSESSMENT QUESTIONS

3. Which of the following is not a general operation of machine cycle in a CPU

(a) Fetch
(b) Decode
(c) Return
(d) Store

4. A unit that decodes, interprets each instruction and generates the required enable signal for ALU and other units is called

(a) Arithmetic Unit
(b) CPU
(c) Logical Unit
(d) Control Unit

TERMINAL QUESTIONS

Short answer questions:

1. Explore the role of the decoding phase in a machine cycle.
2. Specify the role of the fetching phase in a machine cycle.

Long answer questions:

1. Interpret the concept of a machine cycle in computer architecture, outlining its phases.
2. Discuss the data transfer and arithmetic logic instruction sets with examples.
3. Consider a scenario where you are developing a simple combinational circuit of **basic logical operations**. Make use of the related **instruction sets** and accomplish the task.
4. Consider a scenario where you are developing a simple calculator application for a mobile device. You want to implement **basic arithmetic operations.** Make use of the related **instruction sets** and accomplish the task.

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.javatpoint.com/instruction-cycle
2. https://www.tutorialspoint.com/what-is-an-instruction-set-in-a-computer

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – Digital Design & Computer Architecture