Assignment-1

Sec-A

1. Define Program Counter:

ans The Program Counter (PC) is a register in the CPU that holds the address of the next instruction to be executed.

2. What are the primary registers present in CPU?

ans Primary CPU registers include the Program Counter (PC), Accumulator (ACC), Instruction Register (IR), memory Address Register (MAR), and memory Buffer Register (MBR).

3. What does an instruction contain?

ans An instruction typically contains an opcode (operation code) and operands (data or addresses used in the operation).

4. Why is MAR connection with memory unidirectional?

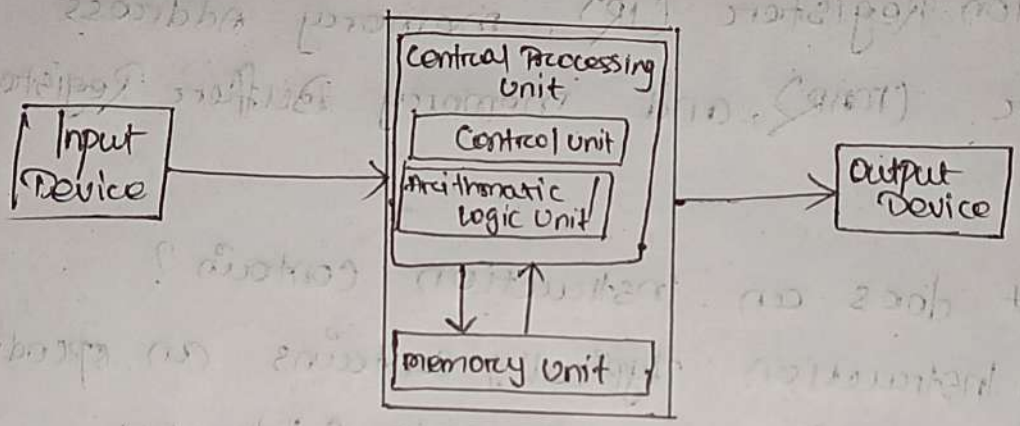
ans: The MAR (Memory Address Register) sends addresses to memory, but does not receive data back, hence the connection is unidirectional.

Sec-B

1. Describe with Diagram Von Neumann Architecture:

ans The Von Neumann Architecture consists of a single memory space for both instruction and data. It includes a CPU (with ALU, CU, registers), memory, and I/O units connected via buses.

Diagram:



2. Differentiate Big Endian System and Little Endian Systems:

Big Endian System	Little Endian System
→ Definition: In Big-endian, the most significant byte (MSB) of a multi-byte value is stored at the lowest memory address. → Byte Order: most Significant byte stored first → Ex: 0x12 stored at lower address → Usage: Used in some Network Protocols	Definition: In little Endian the least Significant byte (LSB) of a multi-byte value is stored at the lowest memory address. → Least significant byte stored first. → 0x34 stored at lower address → Common in Intel Processors.

SOC-C

1. Discuss different types of instruction formats with example:

• Zero-address: Used in stack-based architectures.

Example: ADD (operates on top two stack elements).

• One-address: Uses accumulator and one operand.

Example:
```
LOAD A
ADD B
STORE C
```

• Two-address: Specifies two operands.

Example: ADD A, B (A = A + B).

• Three-address: Uses three operands.

Example: ADD A, B, C (A = B + C).

2. List the instructions using one-address instruction for X = (A*B) + (C*D):

One Assuming accumulator-based one-address instructions:

```
LOAD A ; ACC = A
MUL B ; ACC = A * B
STORE TEMP1 ; TEMP = A * B
LOAD C ; ACC = C
MUL D ; ACC = C * D
STORE TEMP2 ; TEMP2 = C * D
```

LOAD TEMP1 ; ACC = A * B

ADD TEMP2 ; ACC = (A * B) + (C * D)

STORE X ; X = (A * B) + (C * D)