**Assignment 1 Solutions (COMA - 203124210)**

**MCQ/One Sentence Answers:**

1. **Define Microprocessor and explain difference between microprocessor and microcontroller.**
   * A microprocessor is a general-purpose CPU, whereas a microcontroller includes CPU, memory, and peripherals in one chip.
2. **Define: Bit, Nibble, Byte, Word, and Instruction**
   * **Bit:** Smallest unit of data (0 or 1)
   * **Nibble:** 4 bits
   * **Byte:** 8 bits
   * **Word:** Typically 16 or 32 bits (depends on architecture)
   * **Instruction:** A command executed by the processor
3. **What is meaning of an 8-bit processor?**
   * It means the processor can process 8-bit data at a time.
4. **What is Mnemonics?**
   * Mnemonics are symbolic names for machine instructions in assembly language.
5. **Define: Assembly Language, Machine Language, High-Level Language**
   * **Assembly Language:** Uses mnemonics for instructions.
   * **Machine Language:** Binary code understood by the CPU.
   * **High-Level Language:** Human-readable programming languages like C, Python.
6. **What is an Assembler?**
   * A program that converts assembly language into machine code.
7. **What is a Compiler?**
   * A program that translates high-level code into machine code.
8. **Which information is given by the address line?**
   * It specifies the location of data in memory.
9. **Data bus is always bidirectional. (True/False)**
   * **True**
10. **Microcontroller is faster than Microprocessor. Justify.**

* A microcontroller has built-in memory and peripherals, reducing external communication delays.

1. **Why do we need a clock in a microprocessor?**

* It synchronizes the operations of the processor.

1. **Advantages of Assembly Language over High-Level Language?**

* More control, efficient execution, but harder to program.

1. **Full Forms:**

* **RAM:** Random Access Memory
* **ROM:** Read-Only Memory
* **ALU:** Arithmetic Logic Unit
* **SSI:** Small Scale Integration
* **MSI:** Medium Scale Integration
* **VLSI:** Very Large Scale Integration
* **ASCII:** American Standard Code for Information Interchange

1. **What is a Bus?**

* A pathway for transmitting data between system components.

1. **What are Peripherals?**

* External devices connected to a microprocessor.

1. **How much memory can 8085 microprocessor access?**

* 64 KB.

1. **ALU stands for?**

* Arithmetic Logic Unit.

1. **What is the purpose of the Control Unit of a Microprocessor?**

* It manages and coordinates processor operations.

1. **Justify: “Microprocessor is a programmable device.”**

* It can be programmed to perform different tasks using instructions.

1. **Justify: “The direction of address bus is unidirectional.”**

* The CPU sends addresses to memory, but not vice versa.

1. **Difference between 4-bit and 8-bit Microprocessor?**

* An 8-bit processor handles larger data chunks than a 4-bit processor.

1. **Difference between Assembly Language and Machine Language?**

* Assembly uses mnemonics, machine language uses binary.

1. **What is the use of Registers in Microprocessor?**

* They store temporary data for quick access.

1. **CISC vs RISC?**

* **CISC:** Complex Instruction Set Computing (more complex instructions).
* **RISC:** Reduced Instruction Set Computing (simpler, faster instructions).

1. **What is Op-code?**

* The part of an instruction that specifies the operation.

1. **What is Operand?**

* The data on which an operation is performed.

1. **General-Purpose vs Special-Purpose Microprocessor?**

* **General-Purpose:** Used for multiple applications.
* **Special-Purpose:** Designed for specific tasks.

**Answers:**

1. **Maximum Memory Locations and I/O Devices:**
   * The 8085 microprocessor has a **16-bit address bus**, allowing it to address **64KB (65536 bytes) of memory**.
   * It can address **256 I/O devices** using an 8-bit address for I/O ports.
2. **HOLD and HLDA Pins:**
   * **HOLD:** Used by external devices to request control of the system bus.
   * **HLDA (Hold Acknowledge):** The 8085 responds with HLDA when it grants control of the bus to the requesting device.
3. **Difference Between Op-Code Fetch (OF) and Memory Read (MR) Cycles:**
   * **Op-Code Fetch Cycle:** Fetches the operation code (instruction) from memory to the microprocessor.
   * **Memory Read Cycle:** Reads data from memory but not an instruction.
4. **X1 and X2 Pins Importance:**
   * X1 and X2 are **crystal oscillator pins** used to connect an external clock to provide a stable clock signal to the microprocessor.
5. **Interrupt & Hardware Interrupts:**
   * **Interrupt:** A mechanism that temporarily stops normal execution and jumps to a predefined routine.
   * **8085 Hardware Interrupts:**
     + **TRAP (Highest priority)**
     + **RST 7.5**
     + **RST 6.5**
     + **RST 5.5**
     + **INTR (Lowest priority)**
6. **READY Pin Usage:**
   * The READY pin is used to synchronize slower peripherals with the microprocessor. It ensures that the microprocessor waits until the peripheral is ready for data transfer.
7. **RESETIN and RESETOUT Pins:**
   * **RESETIN:** Resets the microprocessor and clears registers.
   * **RESETOUT:** Indicates that the microprocessor is resetting external devices.
8. **Instruction Cycle:**
   * The complete process of fetching, decoding, and executing an instruction.
9. **Machine Cycle:**
   * The time required to complete one operation such as memory read/write or instruction fetch.
10. **T-State and Time Period Calculation:**

* **T-State:** The smallest time unit of execution, corresponding to one clock pulse.
* **Time Period Calculation:**
  + Given clock frequency = **5 MHz**
  + Time period T=1Frequency=15×106=0.2T = \frac{1}{\text{Frequency}} = \frac{1}{5 \times 10^6} = 0.2T=Frequency1​=5×1061​=0.2 μs.

1. **Highest Priority Interrupt:**

* **TRAP** is the highest priority interrupt in the 8085 microprocessor.

1. **Function of ALE and IO/M Signals:**

* **ALE (Address Latch Enable):** Used to demultiplex the address/data bus.
* **IO/M Signal:** Differentiates between memory (IO/M = 0) and I/O operations (IO/M = 1).

1. **Definitions:**

* **Assembly Language:** Low-level language using mnemonics (e.g., MOV, ADD).
* **Machine Language:** Binary code that a microprocessor directly understands.
* **High-Level Language:** Human-readable programming languages like C, Python.

1. **Definitions:**

* **Assembler:** Converts assembly language into machine code.
* **Compiler:** Translates high-level language into machine code.
* **Mnemonics:** Short codes used in assembly language (e.g., MOV, ADD).
* **Instruction:** A command executed by the microprocessor.

1. **Microprocessor vs. Microcontroller:**

* **Microprocessor:** Only CPU; requires external memory and peripherals (e.g., 8085, 8086).
* **Microcontroller:** CPU + memory + I/O ports in one chip (e.g., 8051, PIC).

1. **Time to Execute an Instruction of 18 T-States:**

* Given clock frequency = **5 MHz**
* Time per T-State = **0.2 μs**
* Time required = 18×0.218 \times 0.218×0.2 **= 3.6 μs**

**1. Bus Organization / Bus Structure of 8085 Microprocessor:**

(3-4 Marks)

A **bus** is a group of lines used for communication between the microprocessor and other components. The 8085 microprocessor has the following buses:

1. **Address Bus (16-bit)**
   * Carries memory addresses from CPU to memory and I/O devices.
   * Unidirectional (data flows only from microprocessor to external devices).
2. **Data Bus (8-bit)**
   * Used for data transfer between CPU, memory, and I/O.
   * Bidirectional (data flows both ways).
3. **Control Bus**
   * Carries control signals like **RD, WR, ALE, IO/M** to manage data transfer.
4. **Internal Bus**
   * Connects internal registers and ALU for processing.

Diagram:

mathematica

CopyEdit

+---------------------------+

| 8085 Microprocessor |

+---------------------------+

| Address Bus (A15-A0) → Memory/I/O |

| Data Bus (D7-D0) ↔ Memory/I/O |

| Control Bus (RD, WR, ALE, IO/M) |

+---------------------------+

**2. Flag Register of 8085 Microprocessor:**

(3-4 Marks)

The **Flag Register** is an 8-bit register that stores the status of the result after arithmetic and logical operations. Out of 8 bits, 5 are active flags:

1. **Sign Flag (S)** – Set if the result is negative (D7 = 1).
2. **Zero Flag (Z)** – Set if the result is zero.
3. **Auxiliary Carry (AC)** – Set if carry is generated from lower nibble (used in BCD).
4. **Parity Flag (P)** – Set if the result has an even number of 1s.
5. **Carry Flag (CY)** – Set if there is a carry from the most significant bit (MSB).

Diagram:

diff

CopyEdit

+----+----+----+----+----+----+----+----+

| CY | X | P | X | AC | X | Z | S |

+----+----+----+----+----+----+----+----+

(X = Unused bits)

**3. Programming Model / Registers of 8085:**

(4-5 Marks)

The **8085 programming model** consists of several registers:

1. **General Purpose Registers:**
   * **B, C, D, E, H, L** (Each is 8-bit, can form 16-bit pairs like BC, DE, HL).
2. **Accumulator (A):**
   * 8-bit register used in arithmetic and logical operations.
3. **Flag Register:**
   * Holds condition flags (CY, Z, P, S, AC).
4. **Stack Pointer (SP):**
   * 16-bit register used for stack operations.
5. **Program Counter (PC):**
   * 16-bit register that holds the address of the next instruction.

Diagram:

mathematica

CopyEdit

+----+----+----+----+----+----+----+

| A | B | C | D | E | H | L |

+----+----+----+----+----+----+----+

| Program Counter (PC) |

| Stack Pointer (SP) |

**4. Generation of Control Signals in 8085:**

(4-5 Marks)

The **control signals** are generated using a combination of clock pulses and status signals from the microprocessor. These signals include:

1. **ALE (Address Latch Enable):**
   * Used to demultiplex the address/data bus.
2. **RD (Read Signal):**
   * Indicates a memory or I/O read operation.
3. **WR (Write Signal):**
   * Indicates a memory or I/O write operation.
4. **IO/M Signal:**
   * Differentiates between memory and I/O operations.
5. **INTA (Interrupt Acknowledge):**
   * Indicates that the CPU has acknowledged an interrupt request.

**Control Logic Circuit:**

diff

CopyEdit

+----------------------+

| 8085 CPU |

+----------------------+

| ALE | RD | WR | IO/M |

+----------------------+

**5. De-multiplexing of Data and Address Bus (AD7-AD0):**

(4-5 Marks)

The lower **8 bits of the address (A7-A0) and data bus (D7-D0)** share the same lines (AD7-AD0). To separate them, **Address Latch Enable (ALE)** is used with a latch (74LS373).

**Process:**

1. During the first clock cycle, AD7-AD0 carries the **lower address** and ALE is high.
2. The latch stores the lower address.
3. In the next cycle, AD7-AD0 is used for **data transfer**.

**Diagram:**

lua

CopyEdit

+----+ +----+

AD7-AD0| | | |

------| Latch |--- A7-A0 (Address Bus)

| |

| |--- D7-D0 (Data Bus)

**6. Addressing Modes of 8085:**

(4-5 Marks)

The **8085 microprocessor has different addressing modes** to access data:

1. **Immediate Addressing:**
   * Data is directly specified in the instruction.
   * Example: MVI A, 32H (Load A with 32H).
2. **Register Addressing:**
   * Operand is in a register.
   * Example: MOV B, C (Copy C into B).
3. **Direct Addressing:**
   * Memory address is given in the instruction.
   * Example: LDA 2000H (Load A from address 2000H).
4. **Indirect Addressing:**
   * Address is stored in a register pair.
   * Example: MOV A, M (Load A from memory pointed by HL).
5. **Implied Addressing:**
   * Operation is fixed with the instruction.
   * Example: CMA (Complement accumulator).

**7. Types of Instructions Based on Number of Bytes:**

(4-5 Marks)

Instructions in **8085 are classified based on the number of bytes**:

1. **One-Byte Instructions:**
   * Only the opcode, no additional data.
   * Example: CMA, MOV A, B.
2. **Two-Byte Instructions:**
   * Opcode + One-byte operand.
   * Example: MVI A, 32H (Move 32H to A).
3. **Three-Byte Instructions:**
   * Opcode + 16-bit address.
   * Example: LDA 2000H (Load A from address 2000H).

| **Size** | **Example** | **Explanation** |
| --- | --- | --- |
| 1-Byte | MOV B, C | Moves C to B |
| 2-Byte | MVI A, 32H | Loads 32H into A |
| 3-Byte | LDA 2000H | Loads A from address 2000H |

**1. Functional Block Diagram / Architecture of 8085 and Its Working:**

(7-8 Marks)

The **8085 microprocessor** is an 8-bit microprocessor that follows **Von Neumann architecture** (shared memory for instructions and data). It consists of various functional blocks:

**Architecture of 8085:**

(Draw a detailed labeled diagram with the following components)

1. **Accumulator (A):**
   * 8-bit register used for arithmetic and logical operations.
2. **General-Purpose Registers (B, C, D, E, H, L):**
   * Six 8-bit registers used for temporary storage.
   * Can be paired (BC, DE, HL) for 16-bit operations.
3. **Program Counter (PC):**
   * 16-bit register that holds the address of the next instruction.
4. **Stack Pointer (SP):**
   * 16-bit register used for stack memory operations.
5. **Instruction Decoder & Control Unit:**
   * Decodes instructions and generates control signals.
6. **ALU (Arithmetic Logic Unit):**
   * Performs arithmetic (ADD, SUB) and logic (AND, OR, XOR) operations.
7. **Flag Register:**
   * Stores status flags (S, Z, AC, P, CY).
8. **Timing and Control Unit:**
   * Generates control signals (RD, WR, ALE, etc.).
9. **Interrupt Control Unit:**
   * Handles interrupts (TRAP, RST7.5, RST6.5, RST5.5, INTR).
10. **Address and Data Bus:**

* **Address Bus (16-bit):** For memory addressing.
* **Data Bus (8-bit):** For data transfer.

**2. Pin Diagram of 8085 Microprocessor and Explanation of Each Pin:**

(7-8 Marks)

**Pin Diagram of 8085:**

(Draw a **detailed** pin diagram of **40-pin IC** with proper labels.)

**Pin Descriptions:**

1. **Power Supply & Clock Signals:**
   * **VCC (Pin 40):** +5V Power supply.
   * **VSS (Pin 20):** Ground.
   * **X1, X2 (Pins 1, 2):** Connect external clock crystal.
   * **CLK OUT (Pin 37):** Provides clock signal to external devices.
2. **Address and Data Bus:**
   * **A15 - A8 (Pins 21-28):** Higher address bus lines.
   * **AD7 - AD0 (Pins 12-19):** Multiplexed lower address/data bus.
3. **Control & Status Signals:**
   * **ALE (Pin 30):** Address Latch Enable, separates address/data bus.
   * **IO/M (Pin 31):** Differentiates between memory (0) and I/O (1).
   * **RD (Pin 32):** Read signal (Active Low).
   * **WR (Pin 29):** Write signal (Active Low).
4. **Interrupt Signals:**
   * **TRAP (Pin 6):** Non-maskable interrupt (Highest Priority).
   * **RST7.5, RST6.5, RST5.5 (Pins 7, 8, 9):** Maskable interrupts.
   * **INTR (Pin 10):** General interrupt request.
   * **INTA (Pin 11):** Interrupt Acknowledge.
5. **Other Signals:**
   * **RESET IN (Pin 36):** Resets microprocessor.
   * **RESET OUT (Pin 3):** Resets connected devices.

**3. Categories of 8085 Instructions That Manipulate Data:**

(7-8 Marks)

**8085 Instructions Are Categorized Based on Function:**

1. **Data Transfer Instructions:**
   * Move data between registers, memory, and I/O.
   * Example: MOV A, B (Move B into A), LDA 2000H (Load A from address 2000H).
2. **Arithmetic Instructions:**
   * Perform arithmetic operations like ADD, SUB, INR, DCR.
   * Example: ADD B (Add B to A), SUB C (Subtract C from A).
3. **Logical Instructions:**
   * Perform bitwise operations like AND, OR, XOR.
   * Example: ANA B (AND A with B), XRA C (XOR A with C).
4. **Branching Instructions:**
   * Change program execution sequence (JUMP, CALL, RET).
   * Example: JMP 2050H (Jump to 2050H), CALL 3000H (Call subroutine at 3000H).
5. **Control Instructions:**
   * Control microprocessor operations (HLT, NOP).
   * Example: HLT (Halt execution), NOP (No operation).

**4. 4K RAM and 4K Byte Memory Interfacing Circuit with 8085:**

(7-8 Marks)

**Memory Addressing in 8085:**

* **4K RAM requires 12 address lines (A11 - A0)** since **2^12 = 4K (4096 bytes).**
* **8085 has 16 address lines (A15 - A0), so A15 - A12 are used for chip selection.**

**Circuit Design:**

(Draw a **detailed schematic** of **8085 connected to 4K RAM and 4K ROM**, including **address decoding logic** using NAND gates or decoders.)

1. **Address Bus Connections:**
   * **A0 - A11:** Directly connected to RAM/ROM.
   * **A12 - A15:** Used for **Chip Select (CS) logic**.
2. **Data Bus (D0 - D7):**
   * Connected to RAM/ROM for data transfer.
3. **Control Signals:**
   * **IO/M = 0 (Memory operation).**
   * **RD (Read from memory).**
   * **WR (Write to memory).**
4. **Chip Select Logic:**
   * A **Decoder (74LS138 or NAND Gate)** selects memory blocks based on A12 - A15.
   * Example: **A15 = 0, A14 = 0, A13 = 0 → Selects RAM at 0000H - 0FFFH**.