Q1a: Define Microprocessor.

Definition of a Microprocessor

A microprocessor is a single integrated circuit (IC) that incorporates the core functions of a computer's central processing unit (CPU).

- **Key Points:**

- **The "Brain" of a Computer:** It executes instructions, performs calculations, and manages the flow of data within a computer system.
- **Small and Powerful:** Microprocessors pack millions or even billions of transistors into a tiny chip, enabling complex processing in compact devices.
- **Essential for Modern Devices:** They power a vast range of devices from smartphones and laptops to cars, appliances, and industrial equipment.
- **Components:** Typical components of a microprocessor include:
  - Arithmetic Logic Unit (ALU) - Performs arithmetic and logical operations
  - Control Unit (CU) - Decodes instructions and coordinates the operations of other units
  - Registers - Small, high-speed memory locations for temporary data storage

Q1b: Explain Flag register of 8085 microprocessor.

The Flag Register

The Flag register in the 8085 is an 8-bit register, with only 5 bits actively used as flags. These flags act as individual flip-flops that are set (1) or reset (0) to reflect specific conditions arising from arithmetic, logical, and other operations performed by the ALU (Arithmetic and Logic Unit).

The 5 Flags:

1. **Sign Flag (S):**
   - Set (1) if the result of an operation is negative (the Most Significant Bit, or MSB, of the result is 1).
   - Reset (0) if the result is positive.
2. **Zero Flag (Z):**
   - Set (1) if the result of an operation is zero.
   - Reset (0) if the result is not zero.
3. **Auxiliary Carry Flag (AC):**
   - Set (1) if there is a carry-out from the lower nibble (lower 4 bits) into the upper nibble (upper 4 bits) of a result.
   - Used primarily in instructions that perform decimal arithmetic.
4. **Parity Flag (P):**
   - Set (1) if the result has even parity (contains an even number of 1s).
   - Reset (0) if the result has odd parity.
5. **Carry Flag (CY):**
   - Set (1) if there is a carry-out from the most significant bit (MSB) of a result during addition, or a borrow during subtraction.
   - Reset (0) otherwise.

How the Flags are Used:

- **Conditional Jumps:** Instructions like JZ (Jump if Zero), JNZ (Jump if Not Zero), JC (Jump if Carry), etc. use the status of these flags to determine whether to branch to different parts of the program.

- **Decision Making:** The processor can examine flag states to modify calculations or behaviors based on previous operations.

Example:

```
; Assume the accumulator (A) holds the value 50
SUB B  ; Subtract the value in register B from the accumulator
JZ LABEL  ; If the result is zero, jump to the code section marked as LABEL
```

Q1c: Explain format of instruction of 8085 microprocessor with example.

8085 Instruction Formats

Instructions in the 8085 microprocessor can be 1, 2, or 3 bytes long. The structure varies depending on the specific instruction and the addressing modes used.

General Structure

- **Opcode (Operation Code):** The first byte of an instruction. It specifies the operation to be performed (e.g., MOV, ADD, JMP).
- **Operands (Optional):** The second and third bytes, if present, provide data or addresses required by the operation. Operands can be:
  - **Registers:** 8-bit registers within the 8085 (B, C, D, E, H, L, or the accumulator A).
  - **Immediate Data:** 8-bit or 16-bit data embedded directly into the instruction.
  - **Memory Addresses:** 16-bit addresses of memory locations.

8085 Instruction Examples

1. **Single-Byte Instruction (No Operands):**
   - **Instruction:** NOP (No Operation)
   - **Opcode:** 00000000
   - **Explanation:** Does nothing - the processor simply moves to the next instruction.
2. **Two-Byte Instruction (Immediate Data):**
   - **Instruction:** MVI A, 42H (Move Immediate to Accumulator)
   - **Opcode:** 00111110
   - **Operand:** 42H (Hexadecimal value to be loaded)
   - **Explanation:** Loads the value 42H into the accumulator.
3. **Three-Byte Instruction (16-bit Memory Address):**
   - **Instruction:** LDA 2050H (Load Accumulator Direct)
   - **Opcode:** 00111010
   - **Operand:** 2050H (16-bit memory address)
   - **Explanation:** Loads the content of the memory location at address 2050H into the accumulator.

Addressing Modes

The way operands are specified determines the "addressing mode" of the instruction. The 8085 supports modes like:

- **Register addressing:** The operand is a register.
- **Direct addressing:** The operand is a 16-bit memory address.
- **Immediate addressing:** The operand is data within the instruction.
- **Register indirect addressing:** The operand's address is held within a register pair.

Remember:

- The specific format depends on the instruction and the addressing mode used.
- Opcodes and addressing modes are how the 8085 interprets the bytes that make up an instruction.

Q1c: Explain function of ALU, Control Unit and CPU of 8085 microprocessor.

1. ALU (Arithmetic Logic Unit)

- **Heart of Calculations:** The ALU performs the core arithmetic and logical operations within the microprocessor.
- **Operations:**
  - Arithmetic: Addition, subtraction, increment, decrement, etc.
  - Logical: AND, OR, XOR, NOT, comparisons, etc.
- **Flags:** Sets status flags (Carry, Zero, Sign, Parity) based on the results of its operations. These flags are used for conditional branching and decision-making by the processor.

2. Control Unit

- **The Orchestrator:** Governs the overall operation of the microprocessor.
- **Key Functions:**
  - **Instruction Decoding:** Interprets the opcode of the current instruction fetched from memory.
  - **Control Signal Generation:** Produces control signals that synchronize and manage the actions of all other units within the microprocessor (ALU, registers, memory interface, etc.).
  - **Data Flow Management:** Coordinates the movement of data between the ALU, registers, and memory/I/O devices.

3. CPU (Central Processing Unit)

- **The Brain of the System:** The CPU is the combination of the ALU and the Control Unit.
- **Responsibilities:**
  - **Instruction Execution:** Fetches instructions from memory, decodes them using the control unit, and executes them using the ALU and other components.
  - **Program Control:** Manages the flow of instructions within a program, including branching and jumps based on conditions.
  - **System Management:** Handles communication with external devices and responds to interrupts.

How They Work Together

1. The Control Unit fetches an instruction from memory.
2. The Control Unit decodes the instruction and generates the necessary control signals.
3. The ALU, if needed, performs the required arithmetic or logical operation.
4. Results may be stored in registers, written to memory, or sent to output devices.
5. The Control Unit directs the processor to fetch the next instruction, continuing the cycle.

Q2a: Explain function of ALE signal with diagram.

What is the ALE Signal?

- The ALE signal is a control signal generated by the 8085 microprocessor.
- It is a positive-going pulse that occurs during the first clock cycle (T1 state) of each machine cycle.

Purpose of the ALE Signal

The primary function of the ALE signal is to demultiplex the lower-order address/data bus (AD0-AD7). This bus is shared (multiplexed) to carry both:

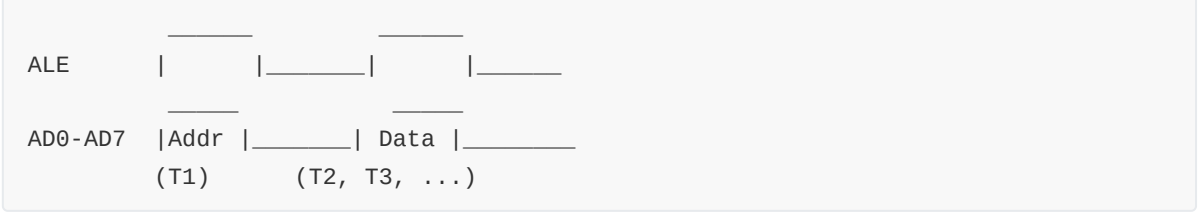
1. **Lower 8-bits of the Address (during T1 state):** The 8085 needs to send out the 16-bit address of a memory location or I/O port. The lower 8 bits of the address are carried on lines AD0-AD7.
2. **Data (during subsequent states):** The same lines are used to transmit or receive actual data to/from the memory or I/O device.

How ALE Demultiplexes the Bus

1. **T1 State:**
   - The ALE signal goes high.
   - The 8085 places the lower 8 bits of the address on lines AD0-AD7.
   - An external latch (usually an 8282 or 8283 octal latch) connected to these lines "latches" or captures this address information.
2. **Subsequent States (T2, T3, ...):**
   - ALE goes low.
   - The lower-order address lines (AD0-AD7) are now free to be used as a data bus for transferring data.

Diagram

A simple timing diagram can help visualize this:



Key Points:

- The ALE signal is crucial for the 8085 to correctly interface with memory and I/O devices.

- The external latch holds the lower order address bits, freeing the 8085 to continue its fetch or write operation.

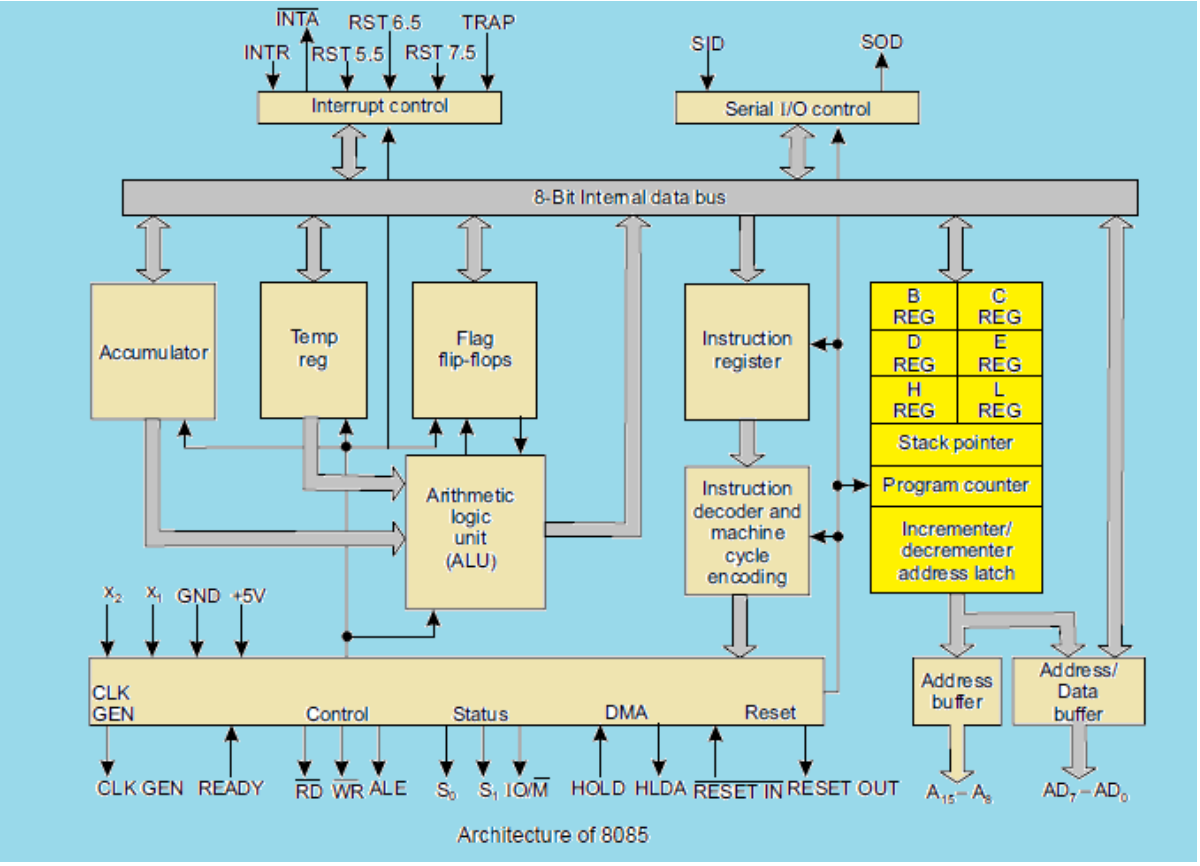
Q2b: Compare microprocessor and microcontroller

Feature	Microprocessor	Microcontroller
Complexity	Less complex	More complex
Instruction Set	Larger and more versatile	Smaller and more application-specific
Memory	Requires external memory (RAM, ROM)	Has built-in memory (RAM, ROM, Flash)
Peripherals	Requires external peripherals (e.g., display, I/O)	Has built-in peripherals (e.g., timers, ADCs, DACs)
Cost	Generally lower cost	Generally higher cost
Applications	General-purpose computing	Embedded systems, specific tasks
Examples	Intel 8085, Intel x86	Atmel AVR, PIC, ARM Cortex-M

Q2c: Draw & explain block diagram of 8085 microprocessor.

Absolutely! Let's break down the block diagram of the Intel 8085 microprocessor.

Block Diagram



Architecture of 8085

Key Components and their Functions

1. **Accumulator:** An 8-bit register that's central to arithmetic and logical operations performed by the ALU.
2. **Arithmetic and Logic Unit (ALU):** Performs arithmetic operations (addition, subtraction, etc.) and logical operations (AND, OR, NOT, etc.). It sets flags (Carry, Zero, Sign, etc.) based on the results.
3. **Temporary Register:** A temporary holding location for data used during instruction execution.
4. **Instruction Register:** Holds the currently fetched instruction.
5. **Instruction Decoder and Machine Cycle Encoder:** Decodes the instruction in the instruction register and generates control signals to coordinate the microprocessor's actions during a machine cycle.
6. **Register Array:** Contains six general purpose 8-bit registers (B, C, D, E, H, and L), which can be used individually or in pairs (BC, DE, HL) for 16-bit operations.
7. **Program Counter (PC):** A 16-bit register that holds the memory address of the next instruction to be fetched.
8. **Stack Pointer (SP):** A 16-bit register pointing to the top of the stack in memory. The stack is used for storing return addresses of subroutines and temporarily storing data.
9. **Timing and Control Unit:** Generates timing and control signals for all operations within the microprocessor and synchronizes with external devices.
10. **Interrupt Control:** Handles incoming interrupt requests (if any), acknowledging them and allowing them to temporarily disrupt the current program execution.
11. **Serial I/O Control:** Facilitates serial input and output, useful for slower communication with certain types of peripherals.
12. **Address Bus (A8 - A15):** The upper 8-bits of the 16-bit address bus, used to send the most significant portion of an address.
13. **Address/Data Bus (AD0 - AD7):** A multiplexed bus. It carries the lower 8 bits of an address during the beginning of a machine cycle and data during data transfer operations.

How it Works (Simplified)

1. **Fetch:** The PC provides an address; the instruction is fetched from memory and placed into the Instruction Register.
2. **Decode:** The Instruction Decoder decodes the instruction to understand what needs to be done.
3. **Execute:** The Control Unit generates signals to coordinate the ALU, registers, and other components as they perform the necessary operations.
4. **Repeat:** The process continues, fetching and executing instructions sequentially.

Let me know if you want specific aspects of the block diagram explained in more detail!

Q2a: Explain 16 bits registers of 8085 microprocessor.

16-Bit Registers in the 8085

The 8085 microprocessor, while primarily an 8-bit processor, features several 16-bit registers that are crucial for memory addressing and specific operations:

- **Program Counter (PC):**
  - Holds the 16-bit memory address of the next instruction to be fetched and executed by the processor.
  - Essential for maintaining the correct sequence of program execution.
- **Stack Pointer (SP):**
  - Points to the current top of the stack in memory.
  - The stack is a Last-In, First-Out (LIFO) data structure used for storing return addresses during subroutine calls, temporary data, and interrupt handling.
- **Register Pairs (BC, DE, HL):**
  - While B, C, D, E, H, and L are individual 8-bit registers, they can be paired together to form 16-bit registers:
    - BC
    - DE
    - HL
  - These register pairs allow for operations on 16-bit data and for holding 16-bit memory addresses.

Key Functions of 16-bit Registers

1. **Memory Addressing:** The 8085 has a 16-bit address bus, meaning it can address up to 64KB of memory. The 16-bit registers are used to store and manipulate memory addresses for data storage and retrieval.
2. **Subroutine Calls and Returns:** When a subroutine is called (using instructions like CALL), the processor needs to store the address where it should return to after the subroutine is finished. The Program Counter is pushed onto the stack for safekeeping.
3. **Data Manipulation:** Some instructions treat these register pairs as a single unit for performing 16-bit operations (e.g., addition, loading immediate 16-bit values).

Q2b: Explain de-multiplexing lower order address and data lines with diagram of 8085 microprocessor.

Q2c: Draw and explain pin diagram of 8085.

Q3a: Draw clock and reset circuit of 8051 microcontroller.

Q3b: Explain internal RAM of 8051.

Q3c: Explain block diagram of 8051.

Q3a: Explain different timer modes of 8051 microcontroller.

Q3b: Explain function of DPTR and PC.

Q3c: Explain interrupts of 8051 microcontroller.

Q4a: Explain data transfer instruction with example for 8051.

Q4b: List and explain different addressing modes of 8051 microcontroller.

Q4c: Write a program to copy block of 8 data starting from location 100h to 200h.

Q4a: Write a program to add two bytes of data and store result in R0 register.

Q4b: Explain indexed addressing mode with example.

Q4c: Explain stack operation of 8051 microcontroller, PUSH and POP instruction.

Q5a: Explain branching instruction with example.

Q5b: Interface 8 leds with 8051 microcontroller and write a program to turn on and off.

Q5c: Interface LCD with 8051 microcontroller and write a program to display “welcome”.

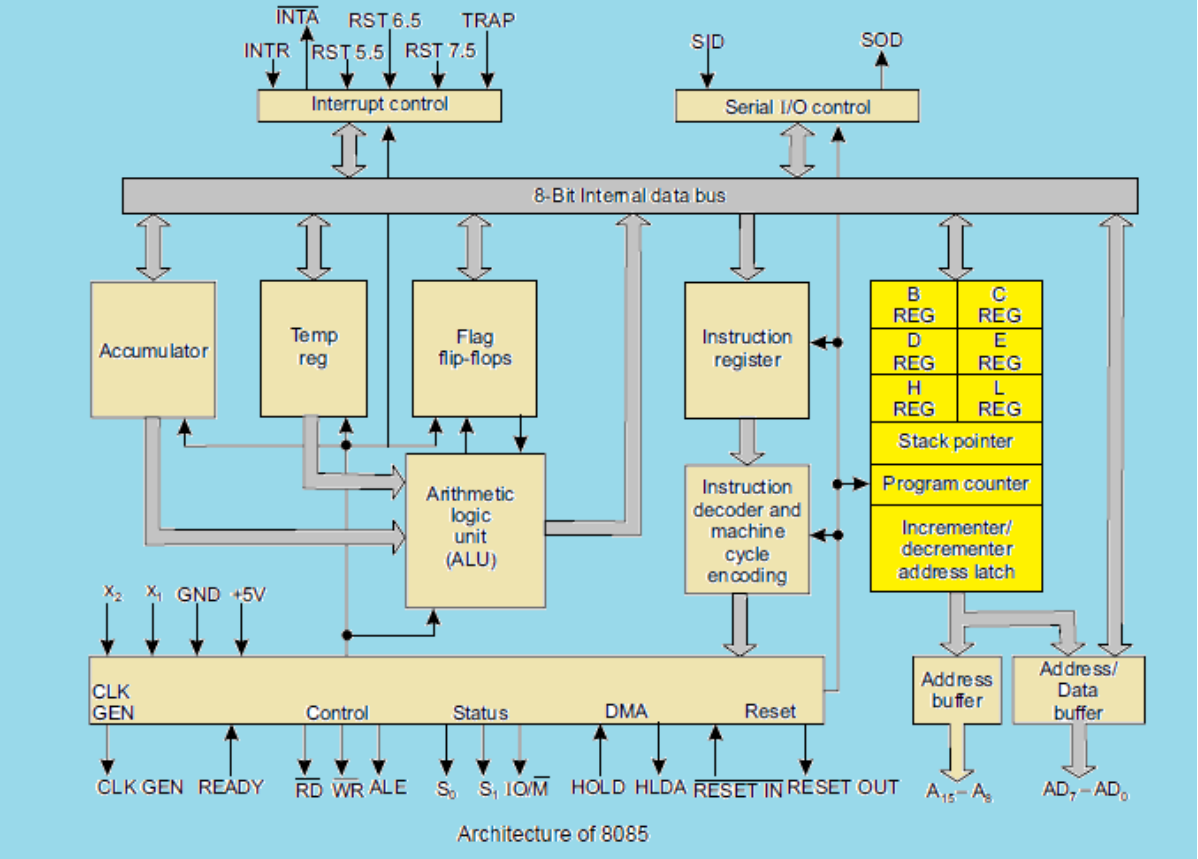
Q5a: Explain logical instruction with example.

Q5b: Interface 7 segment with 8051 microcontroller.

Q5c: Interface LM 35 with 8051 microcontroller and explain block diagram of temperature controller.

Absolutely! Let's break down the block diagram of the Intel 8085 microprocessor.

Block Diagram



Architecture of 8085

Key Components and their Functions

1. **Accumulator:** An 8-bit register that's central to arithmetic and logical operations performed by the ALU.
2. **Arithmetic and Logic Unit (ALU):** Performs arithmetic operations (addition, subtraction, etc.) and logical operations (AND, OR, NOT, etc.). It sets flags (Carry, Zero, Sign, etc.) based on the results.
3. **Temporary Register:** A temporary holding location for data used during instruction execution.
4. **Instruction Register:** Holds the currently fetched instruction.
5. **Instruction Decoder and Machine Cycle Encoder:** Decodes the instruction in the instruction register and generates control signals to coordinate the microprocessor's actions during a machine cycle.
6. **Register Array:** Contains six general purpose 8-bit registers (B, C, D, E, H, and L), which can be used individually or in pairs (BC, DE, HL) for 16-bit operations.
7. **Program Counter (PC):** A 16-bit register that holds the memory address of the next instruction to be fetched.

8. **Stack Pointer (SP):** A 16-bit register pointing to the top of the stack in memory. The stack is used for storing return addresses of subroutines and temporarily storing data.
9. **Timing and Control Unit:** Generates timing and control signals for all operations within the microprocessor and synchronizes with external devices.
10. **Interrupt Control:** Handles incoming interrupt requests (if any), acknowledging them and allowing them to temporarily disrupt the current program execution.
11. **Serial I/O Control:** Facilitates serial input and output, useful for slower communication with certain types of peripherals.
12. **Address Bus (A8 - A15):** The upper 8-bits of the 16-bit address bus, used to send the most significant portion of an address.
13. **Address/Data Bus (AD0 - AD7):** A multiplexed bus. It carries the lower 8 bits of an address during the beginning of a machine cycle and data during data transfer operations.

How it Works (Simplified)

1. **Fetch:** The PC provides an address; the instruction is fetched from memory and placed into the Instruction Register.
2. **Decode:** The Instruction Decoder decodes the instruction to understand what needs to be done.
3. **Execute:** The Control Unit generates signals to coordinate the ALU, registers, and other components as they perform the necessary operations.
4. **Repeat:** The process continues, fetching and executing instructions sequentially.

Let me know if you want specific aspects of the block diagram explained in more detail!