8085 Microprocessor

The 8085 microprocessor has an 8-bit data bus and a 16-bit address bus. Thus, it can address 64 KB of memory. The 8085 microprocessor has an 8-bit ALU that can perform 8-bit operations. The lower order address bus is multiplexed with the data bus to minimize the chip size. The 8085 microprocessor is an 8-bit processor available as a 40-pin IC package (shown in the figure below) and uses +5 V for power. It can run at a maximum frequency of 3 MHz.

Three control signals are available on the chip:

- **RD:** it is an active low signal. This indicates that the selected IO or Memory device is to be read, and data is available on the data bus.
- **WR:** it is an active low signal which indicates that the data on the data bus is to be written into a selected memory or IO location.
- **ALE:** it is a +ve going pulse generated every time the 8085 begins an operation (machine cycle), which indicates that the bits on AD7-AD0 are address bits.

The status signals are available on the chip:

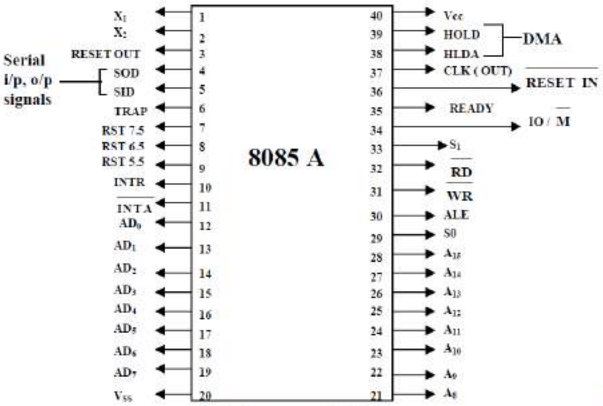
- **IO/M:** This is a status signal to differentiate between IO and Memory operations. If it is high, then IO operation; if it is low, then Memory operation.
- **S_1 and S_0:** status signals similar to IO/M can identify various operations rarely used in the systems.

8085 Microprocessor Pin Diagram

The 8085 has extensions to support new interrupts, with three maskable interrupts (RST 7.5, RST 6.5 and RST 5.5), one non-maskable interrupt (TRAP), and one externally serviced interrupt (INTR). 8085 microprocessor is important part of GATE CSE syllabus. Here are a few prerequisites to know for the 8085 Microprocessor:

- **Bit**: A bit is a single binary digit.
- **Word**: A word refers to the basic data size or bit size that can be processed by the arithmetic and logic unit of the processor. A 16-bit binary number is called a word in a 16-bit processor.
- **Memory Word**: The number of bits stored in a register or memory element is called a memory word.
- **Bus**: A bus is a group of wires (lines) that carry similar information.
- **System Bus**: A system bus is a group of wires used for communication between the microprocessor and peripherals.
- **Address Bus**: It carries the address, a unique binary pattern used to identify a memory location or an I/O port.
- **Data Bus**: The bus transfers data between memory and processor or I/O device and processor.

- **Control Bus**: The control bus carries control signals, which consist of signals for selecting memory or I/O device from the given address, the direction of data transfer, and synchronization of data transfer in case of slow devices.



Pin Diagram of 8085

8085 Microprocessor Architecture

The architecture of 8085 consists of three main sections, ALU (Arithmetic and Logical Unit), timing and control unit and Registers.

Arithmetic and Logic Unit (ALU)

The ALU performs numerical and logical operations.

- The ALU performs the following arithmetic and logical operations.
  - Addition, Subtraction
  - Logical AND, Logical OR, Logical Ex - OR
  - Complement (logical NOT)
  - Increment, Decrement
  - Left shift, Right shift
  - Clear, etc.
- ALU includes the accumulator, the temporary register, the arithmetic and logic circuits and flags. It always stores the results of operations in an Accumulator.

Timing & Control Unit

It generates timing and control signals, which are necessary for the execution of instructions.

- It controls data flow between the CPU and peripherals (including memory).
- It provides status, control and timing signals required to operate memory and I/O devices.

Program Counter (PC): This 16-bit register deals with sequencing the execution of instructions. The microprocessor uses this register to sequence the execution of the instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched.

Stack Pointer (SP): The stack pointer is a 16-bit register used as a memory pointer. It points to a memory location in read-write memory called the stack.

Instruction Register/Decoder: Temporary store for the current instructions of a program. The latest instruction was sent here from memory before execution. The decoder then takes instruction and decodes or interprets the instruction. Decoded instructions are then passed to the next stage.

Memory Address Register: Holds address received from PC of next program instruction.

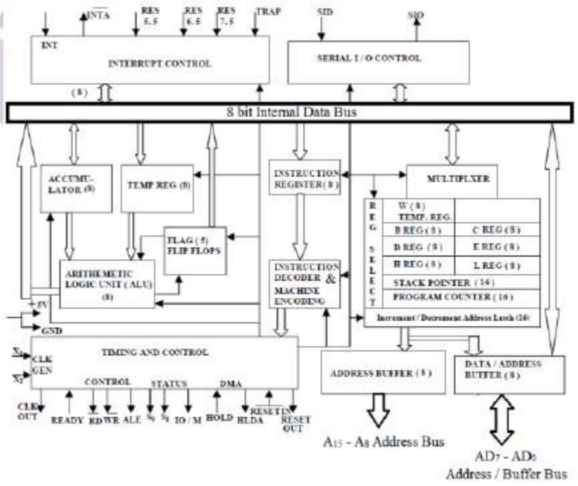
Control Generator: It generates a signal within µP to carry out the instructions which have been decoded.

Register Selector: This block controls the use of the register stack.

General Purpose Registers: µP requires extra registers for versatility. It can be used to store additional data during a program.

8085 Microprocessor Architecture Diagram

The architecture diagram of the 8085 microprocessor is shown below:



8085 System Bus: The microprocessor communicates with memory and other devices (input and output) using three buses: Address Bus, Data Bus and Control Bus.

Address Bus: The Address bus consists of 16 wires. The address bus's size determines the memory size, which can be used. To communicate with memory, the microprocessor sends an address on the address bus to the memory. The address bus is unidirectional, *i.e.,* numbers are only sent from the microprocessor to memory.

Data Bus: Bus is bidirectional. The size of the data bus determines what arithmetic can be done. The data bus also carries instructions from memory to the microprocessor.

Memory size = 2^A x D were, A denotes the address lines, and D denotes the data lines.

Control Bus: Control buses are various lines that have specific functions for coordinating and controlling μP operations. The control bus carries control signals partly unidirectional and partly bidirectional. Control signals are things like reading or writing.

Registers: 8085 has six general-purpose registers to store 8-bit data, identified as B, C, D, E, H and L. They can be combined as register pairs BC, DE and HL to perform 16-bit operations.

Accumulator: The accumulator is an 8-bit register included as a part of the Arithmetic Logic Unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator.

Flag Register: The ALU includes five flip-flops. They are called Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags. The microprocessor uses these flags to test data conditions. The software instructions test the flags' conditions (set or reset).

Bit Position of Various Flags in Flag Register of 8085

The combination of the flag register and the accumulator is called Program Status Word (PSW), and PSW is the 16-bit unit for stack operation.

D_7	D_6	D_5	D_4	D_3	D_2
S	Z		AC		P

Operations of Microprocessor

The microprocessor performs the following four operations using the address bus, data bus, and control bus:

- Memory Read: Reads data (or instruction) from memory.
- Memory Write: Writes data (or instruction) into memory.
- I/O Read: Accepts data from the input device.
- I/O Write: Sends data to the output device.

8085 Microprocessor Instruction Format

An instruction is a command to the microprocessor to perform a given task on specified data. Each instruction has two parts, one is a task to be performed, called the operation code (opcode), and the second is the data to be operated on called the operand. The 8085 instruction set is classified according to a word size.

- **One-Byte Instructions:** A 1-byte instruction includes the opcode and operand in the same byte. Operands are internal registers and are coded into the instruction.
- **Two-Byte Instructions:** In a two-byte instruction, the first byte specifies the operation code and the second byte specifies the operand. The source operand is a data byte immediately following the opcode.
- **Three-Byte Instructions:** In a three-byte instruction, the first byte specifies the opcode and the following two bytes specify the 16-bit address. Note that the second byte is the low-order address and the third byte is the high-order address.

8085 Microprocessor Addressing Modes

The various formats for specifying operands are called the addressing modes. For 8085, they are

- **Immediate Addressing:**
    - Data is provided in the instruction.
    - Load the immediate data to the destination provided.
    - Example: MVI A, 12 H

- **Register Addressing:**
    - Data is provided through the registers.
    - Example: MOV B, C
- **Direct Addressing:**
    - Used to accept data from outside devices to store in the accumulator or send the data stored in the accumulator to the outside device.
    - Example: MOV A, [1000]
- **Indirect Addressing:**
    - The processor calculates the effective address, and the contents of the address is used to form a second address. The second address is where the data is stored.
    - Example: MOV A, [[1000]]
- **Implicit addressing:**

- In this addressing mode, the data itself specifies the data to be operated upon.
- Example: CMA; Complement the contents of the accumulator

8085 Microprocessor Instruction Set

An instruction is a binary pattern designed inside a microprocessor to perform a specific function. Each instruction is represented by an 8-bit binary value. The instruction set can be categorized into 5 types:

1. **Data Transfer Instructions**:
   - These instructions transfer data from one register to another, from memory to register or register to memory.
   - When an instruction for a data transfer group is executed, data is transferred from the source to the destination without altering the contents of the source.
   - Examples: MOV, MVI, LXI, LDA, STA, etc.
2. **Arithmetic Instructions:**
   - These instructions are used to perform arithmetic operations such as addition, subtraction, increment or decrement of the content of a register or memory.
   - Examples: ADD, ADC, ADI, DAD, SUB, INR, DCR, etc.
3. **Logical Instructions:**
   - These instructions perform logical operations such as AND, OR, compare, rotate etc.
   - Examples: ANA, ANI, ORA, ORI, XRA, CMA, CMC, STC, CMP, RLC, RAL, RAR, etc.
4. **Branching Instructions:**
   - These instructions are used to perform a conditional and unconditional jump, subroutine call and return, and restart.
   - Examples: JZ, JNZ, JC, JNC, JP, JM, JPE, JPO, CALL, RET, RST, etc.
5. **Machine Control Instructions:**
   - These instructions control machine functions such as Halt, Interrupt, or do nothing.
   - The microprocessor operations related to data manipulation can be summarized in four functions: copying data, performing arithmetic operations, performing logical operations, testing for a given condition and alerting the program sequence.
   - Example: PUSH, POP, HLT, XTHL, NOP, EI, DI, etc.

8085 Microprocessor Example

Example-1: Write an 8085 assembly program for multiplying two 8-bit numbers.

- MVI A,00; Load immediate data into the accumulator.

- MVI B,02; Load immediate data into register B.
- MVI C,04; Load immediate data into register C.
- LOOP: ADD B; Add the content to the accumulator.
- DCR C; Decrement the content of register C by 1.
- JNZ LOOP
- STA 1000 H; Store the content of the accumulator in memory location 1000 H
- HLT ; Halt

Example-2: With the 8085 assembly program, find the largest number in an array of data.

- LXI H, 1000; Load the address of the first element of the array in the HL pair
- MOV B, M; Load the Count
- INX H; Set the first element as the largest data
- MOV A, M; Get the first data in A
- DCR B; Decrements the count
- LOOP: INX H
- CMP M; Compare A and M
- JNC AHEAD; if no carry (A>M), then go to AHEAD
- MOV A, M; Set the new value as the largest
- AHEAD: DCR B
- JNZ LOOP; Repeat comparisons till count = 0
- STA 2000; Store the largest value at 2000
- HLT

Direct Memory Access: Direct memory access (DMA) facilitates data transfer operations between main memory and I/O subsystems with limited CPU intervention. Most I/O devices provide two methods for transferring data between a device and memory.

Programmed I/O (PIO): It is fairly easy to implement but requires the processor to constantly read or write a single memory word (8-bit, 16-bit or 32-bits, depending on the device interface) until the data transfer is complete. Although PIO is not necessarily slower than DMA, it does consume more processor cycles and can be detrimental in a multi-processing environment.

DMA: It allows a system to issue an I/O command to a device, initiate a DMA transaction and then place the process in a waiting queue. The system can now continue by selecting another process for execution, thereby utilizing the CPU cycles typically lost when using PIO. The DMA controller will inform the system when its current operation has been completed by issuing an interrupt signal. Although the data is transferred 1 memory unit at a time from the device, the transfer to the main memory now circumvents the CPU because the DMA controller can directly access the memory unit.

Steps Involved in the Mode of DMA Transfer

The device wishing to perform DMA asserts the processor's bus request signal. The processor completes the current bus cycle and then asserts the bus grant signal to the device. The device then asserts the bus grant ack signal.

- The processor senses the change in the state of the bus, grants an ack signal and starts listening to the data, and addresses the bus for DMA activity.
- The DMA device performs the transfer from the source to the destination address.
- During these transfers, the processor monitors the addresses on the bus and checks if any location modified during DMA operations is cached in the processor. If the processor detects a cached address on the bus, it can take one of the two actions:
    - The processor invalidates the internal cache entry for the address involved in the DMA write operation
    - The processor updates the internal cache when a DMA write is detected
- Once the DMA operations have been completed, the device releases the bus by asserting the bus release signal.
- The processor acknowledges the bus release and resumes its bus cycles from where it left off.

The 8085 microprocessor has two pins for the DMA mode of I/O communication: HOLD (Hold) and HLDA (Hold Acknowledge).

- **HOLD:** This is an active-high input signal to the 8085 from another master requesting the use of the address and data buses. After receiving the HOLD request, the Microprocessor relinquishes the buses in the following machine cycle. All buses are tri-stated, and a Hold Acknowledge signal is sent out. The Microprocessor regains control of buses after HOLD goes low.
- **HLDA:** This is an active-high output signal indicating that the MPU is relinquishing control of the buses. Typically, an external peripheral such as a DMA controller, sends a request for a high signal to the HOLD pin. The processor completes the execution of the current machine cycle; floats (high impedance state) the address, the data, and the control lines; and sends the Hold Acknowledge (HLDA) signal. The DMA controller takes control of the buses and transfers data directly between the source and destination, thus bypassing the microprocessor. At the end of the data transfer, the controller terminates the request by sending a low signal to the HOLD pin, and the microprocessor regains control of the buses.