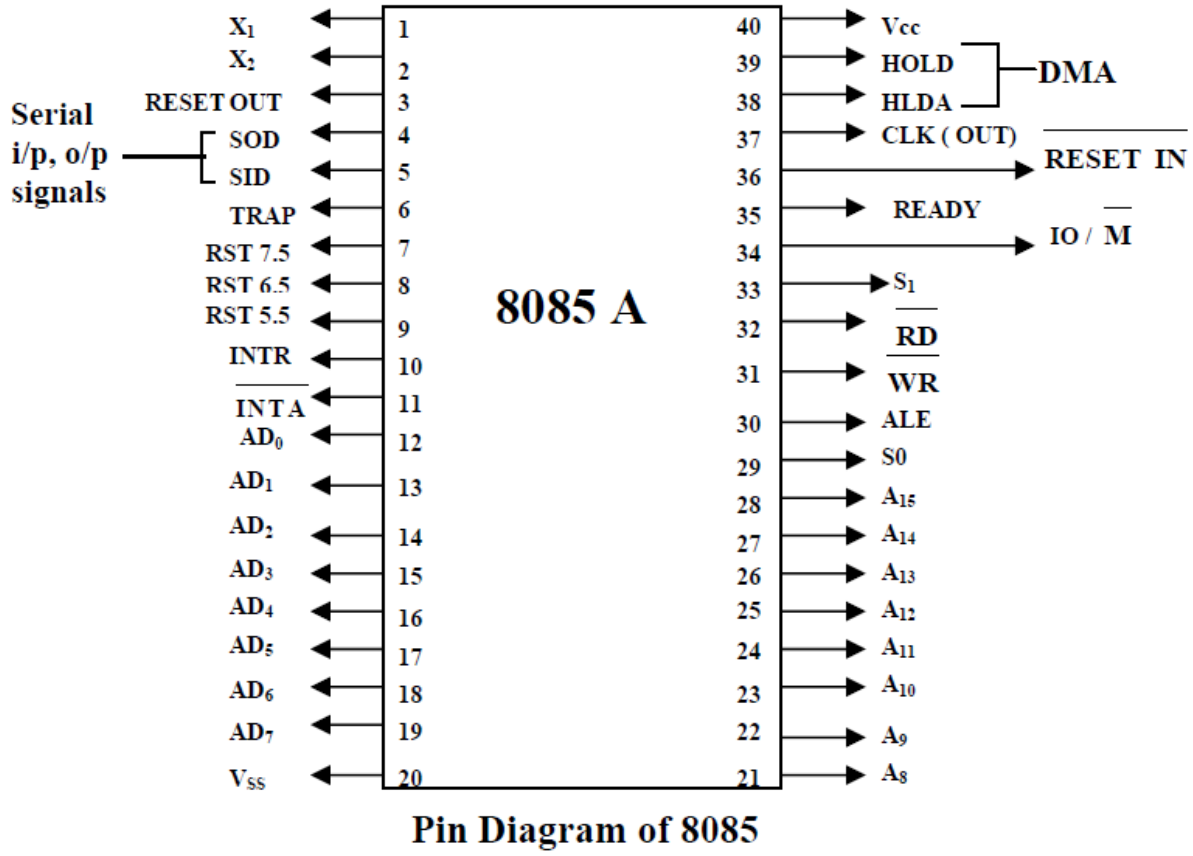
Experiment 11

Aim: To study the architecture and instruction set of 8085 Microprocessor.

Tools Used: (Research)

Theory: The 8085 Microprocessor or Intel 8085 ("eighty-eighty-five") is an 8-bit microprocessor produced by Intel and introduced in March 1976. It is created with N-MOS technology. It has an 8-bit data bus. This means that 8 bits of data can flow around in the innards of the microprocessor. Apart from the data bus, it also has a 16-bit address bus, which addresses up to 64KB. It also has a 16-bit program counter & a 16-bit stack pointer. There are six 8-bit registers which are arranged in pairs: BC, DE, HL. It requires a voltage supply of +5V to operate at 3.2MHZ single-phase clock frequency. The Intel 8085 comes as a 40-pin IC package.



Pin Diagram of 8085

Fig 1: Pin Diagram of 8085 Microprocessor

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).

Three control signals are available on chip:

- RD: it is a active low signal. Which indicate that the selected IO or Memory device is to be read and data is available on the data bus.

- WR: it is a active low signal which indicate that the data on the data bus are 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 indicate that the bits on AD7-AD0 are address bits.

Three status signals are available on chip:

- IO/M: this is a status signal used to differentiate between IO and Memory operations. If it is high then IO operation and If it is low then Memory operation.
- S1 and S0: status signals similar to IO/M, can identify various operations that are rarely used in the systems.

Internal Architecture of 8085 Microprocessor: 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 the actual 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

ALU includes the accumulator, the temporary register, the arithmetic and logic circuits and flags. It always stores result of operations in Accumulator.

Timing & Control Unit:

- It generates timing and control signals, which are necessary for the execution of instructions.
- It controls data flow between CPU and peripherals (including memory).
- It provides status, control and timing signals, which are required for the operation of memory and I/O devices.

8085 System Bus: 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 size of the address bus determines the size of memory, which can be used. To communicate with memory the microprocessor sends an address on the address bus to the memory. Address bus is unidirectional, i.e., numbers only sent from microprocessor to memory.
- Data Bus: Bus is bidirectional. Size of the data bus determines what arithmetic can be done. Data bus also carries instructions from memory to the microprocessor.

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

- Control Bus: Control bus are various lines which have specific functions for coordinating and controlling μP operations. The control bus carries control signals partly unidirectional, partly bidirectional. Control signals are things like read or write.

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

- Accumulator: The accumulator is an 8 - bit register included as a part of 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 conditions (set or reset) of the flags are tested through the software instructions. 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.
- Program Counter (PC): This 16 - bit register deals with sequencing the execution of instruction. 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 also 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. Latest instruction sent here from memory prior to execution. Decoder then takes instruction and decodes or interprets the instruction. Decoded instruction then passed to next stage.
- Memory Address Register: Holds address, received from PC of next program instruction.
- Control Generator: It generates 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.

Operations of Microprocessor: The microprocessor performs the following four operations using 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 input device.
- I/O Write: Sends data to output device.

The 8085 Instruction Format: An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts, one is 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 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. Source operand is a data byte immediately following the opcode.
- **Three-Byte Instructions:** In a three - byte instruction, the first byte specifies the op - code 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.

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

- **Immediate Addressing:**
  i) Data is provided in the instruction.
  ii) Load the immediate data to the destination provided.
  iii) Example: MVI A, 12 H.
- **Register Addressing:**
  i) Data is provided through the registers.
  ii) Example: MOV B, C.
- **Direct Addressing:**
  i) Used to accept data from outside devices to store in the accumulator or send the data stored in the accumulator to the outside device.
  ii) Example: MOV A, [1000].
- **Indirect Addressing:**
  i) Effective address is calculated by the processor and the contents of the address is used to form a second address. The second address is where the data is stored.
  ii) Example: MOV A, [[1000]].
- **Implicit addressing:**
  i) In this addressing mode the data itself specifies the data to be operated upon.
  ii) Example: CMA; Complement the contents of accumulator.

8085 Instruction set: An instruction is a binary pattern designed inside a microprocessor to perform a specific function. Each instruction is represented by 8 bit binary value. Instruction set can be categorised int0 5 types:

Data transfer instructions:

- These instructions are used to transfer data from one register to another register, from memory to register or register to memory.
- When an instruction of 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.

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.

Logical instructions:

- These instructions are used to perform logical operations such as AND, OR, compare, rotate etc.
- Examples: ANA, ANI, ORA, ORI, XRA, CMA, CMC , STC, CMP, RLC, RAL, RAR, etc.

Branching Instructions:

- These instructions are used to perform conditional and unconditional jump, subroutine call and return, and restart.
- Examples: JZ, JNZ, JC, JNC, JP, JM, JPE, JPO, CALL, RET, RST, etc.

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.

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

- MVI A,00; Load immediate data into 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 of to accumulator.
- DCR C; Decrement the content of register C by 1.
- JNZ LOOP
- STA 1000 H; Store the content of accumulator to memory location 1000 H
- HLT; Halt

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

- LXI H, 1000; Load the address of the first element of the array in HL pair
- MOV B, M; Load the Count
- INX H; Set first element as 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 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. The majority of 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-bits, 16-bits 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 still transferred 1 memory unit at a time from the device, the transfer to main memory now circumvents the CPU because the DMA controller can directly access the memory unit.
- Steps involved in the mode of DMA transfer are as follows.
  i) Device wishing to perform DMA asserts the processors bus request signal.
  ii) Processor completes the current bus cycle and then asserts the bus grant signal to the device.
  iii) The device then asserts the bus grant ack signal.
  iv) The processor senses in the change in the state of bus grant ack signal and starts listening to the data and address bus for DMA activity.
  v) The DMA device performs the transfer from the source to destination address.
  vi) 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:
    - Processor invalidates the internal cache entry for the address involved in DMA write operation
    - Processor updates the internal cache when a DMA write is detected
  vii) Once the DMA operations have been completed, the device releases the bus by asserting the bus release signal.
  viii) Processor acknowledges the bus release and resumes its bus cycles from the point it left off.

The 8085 microprocessor has two pins available for 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 the control of buses after HOLD goes low.

HLDA: This is an active high output signal indicating that the MPU is relinquishing the control of the buses. Typically, an external peripheral such as DMA controller sends a request 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 source and destination, thus bypassing the microprocessor. At the end of data transfer, the controller terminates the request by sending a low signal to the HOLD pin, and the microprocessor regains control of the buses.

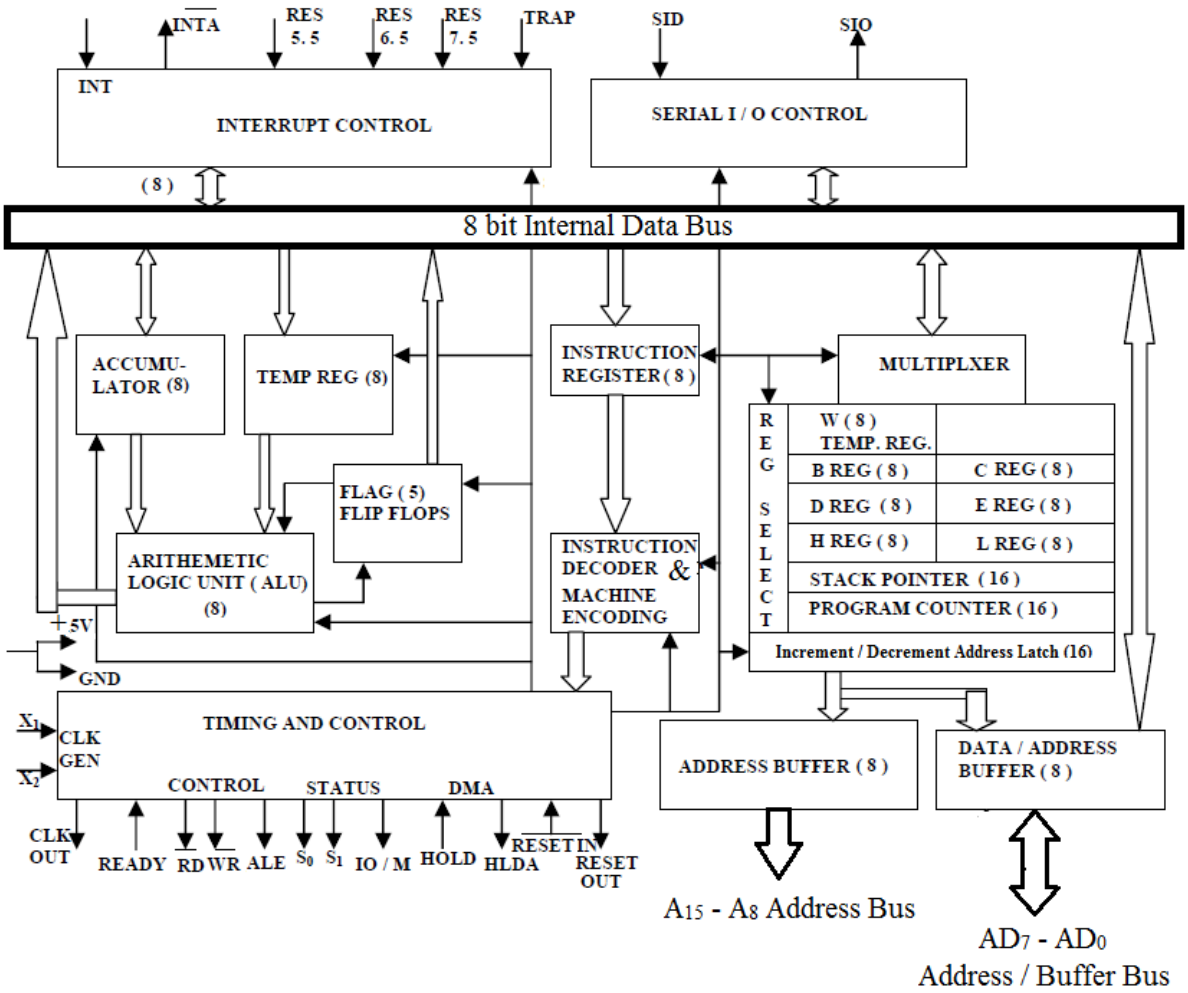


Fig 2: Internal Architecture of 8085 Microprocessor

Results and Conclusion: The 8085 Microprocessor has been studied successfully.

CRITERIA	TOTAL MARKS	MARKS OBTAINED	COMMENTS
(A) CONCEPT	2		
(B) IMPLEMENTATION	2		
(C) PERFORMANCE	2		
TOTAL	6		