MICROPROCESSOR

A microprocessor incorporates the functions of a computer's central processing unit (CPU) on a single Integrated (IC), or at most a few integrated circuit.

It is a multipurpose, programmable device that accepts digital data as input, processes it according to instructions stored in its memory, and provides result as output. It is example of sequential digital logic, as it has internal memory. Microprocessors operate on numbers and symbols represented in the binary numeral system.

General purpose microprocessor in personal computer and used for computation, text editing, multimedia display, and communication over the Internet.

A microprocessor is a general purpose system.

Microprocessor (MPU): A semiconductor device (integrated circuit) manufactured by using the LSI technique. It includes the ALU, register arrays, and control circuits on a single chip. The term MPU is also synonymous with the microprocessor.

Microprocessor-based product: A machine or product that uses a microprocessor to run or execute its operations. It is represented by three components: microprocessor, memory and I/O (input/output).

Microcontroller: A device that includes microprocessor, memory, and I/O signal lines on a single chip, fabricated using VLSI technology.

Microcomputer: A computer that is designed using a microprocessor as its CPU. It includes microprocessor, memory, and I/O (input/output).

Bus: A group of lines used to transfer bits between the microprocessor and other components of the computer system.

ROM (Read-Only memory): A memory that stores binary information permanently. The information can be read from this memory but cannot be altered.

R/WM (Read/Write memory): A memory that stores binary information during the operation of the computer. This memory is used as a writing pad to write user programs and data. The information stored in this memory can be read and altered easily.

The functions of various components of a micro-processor-based system can be summarized as follows:

1. **The microprocessor**

(i) Reads instructions from memory.

(ii) Communicates with all peripherals (memory and I/Os) using the system bus.

(iii) Controls the timing of information flow.

(iv) Performs the computing tasks specified in a program.

2. **The Memory:**

(i) Stores binary information, called instructions and data.

(ii) Provides the instructions and data to the microprocessor on request.

(iii) Stores results and data for the microprocessor.

3. **The Input Device:**

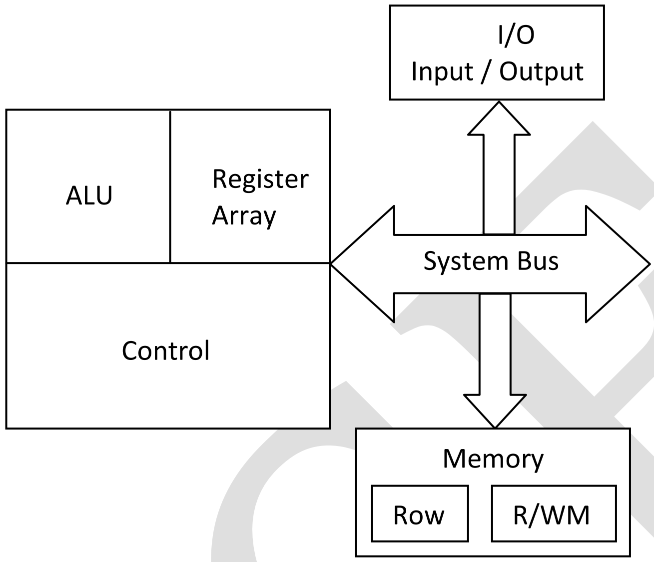
(i) Enters data and instructions under the control of a program such as a monitor program.

4. **The Output Device:**

(i) Accepts data from the microprocessor as specified in a program.

5. **The Bus:**

(i) Carries bits between the microprocessor and memory and I/Os.



Microprocessor-based system with Bus Architecture

Microprocessor: The microprocessor is a clock driven semiconductor device consisting of electronic logic circuits manufactured by using either a large-scale integration (VLCI) technique. The microprocessor can be divided into three segments i.e. (1) Arithmetic/logic unit (ALU) 2. Register 3. Control Unit

(1) **Arithmetic Logic Unit:** The arithmetic and logic unit (ALU) performs the following arithmetic and logic operations: Addition, Subtraction, Logical AND, Logical OR, Logical EXCLUSIVE OR, Complement, Increment, Decrement, Left shift, Clear.

(2) **Registers:** Registers are used by the microprocessor for temporary storage and manipulation of data and instructions. Data remain in the register till they are sent to the memory or I/O devices. Intel 8085 microprocessor has the following registers.

(3) **Timing and 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. It controls the entire operation of the microprocessor and peripherals connected to it. Thus it is seen that the control unit of the CPU acts as the brain of the computer system.

How does the microprocessor work?

The microprocessor works on the sequence of the process i.e. **read, interpret and perform.**

The instructions are stored sequentially in the memory. The microprocessor fetches the first instructions from its memory sheet, decodes it, and executes that instructions. The sequence of fetch decode, and execute is continued until the microprocessor comes across an instruction to stop. During the entire process the microprocessor uses the system bus to fetch the binary instructions and data from the memory. It uses registers from the register section to store data temporarily and it performs the computing function in the ALU section finally, it sends out the result in binary using the same bus lines, to the seven segment LEDs.

Example:

1. Assume that a program and data are already entered in the R/W memory.
2. The program includes binary instructions to add given data and to display the answer at the seven-segment LEDs.
3. When the microprocessor is given a command to execute the program, it reads and executes one instruction at a time and finally sends the result to the seven-segment LEDs for display.

8085 PROGRAMMING MODEL:

A model is a conceptual representation of a real objects. It can take many forms, such as text, description, a drawing, or a built structure.

The model includes: Six registers, one accumulator, one flag register, 16 bit registers, Stack pointers, Program counter.

Registers: 1. The 8085 has six general-purpose registers to store 8-bit data.

2. These are B, C, D, E, H and L.

3. Above registers are combined into register pairs as BC, DE and HL to perform 16 bit operations.

4. By using data copy instructions programmer com store or copy data into the registers.

Accumulator: 1. The accumulator is an 8 bit register which is a part of the arithmetic/logic unit (ALU).

2. The 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.

Flags: There are five flop-flops. Which acts as status flags. Each of these flip-flops holds 1-bit flag that indicates certain condition which arises during arithmetic and logic operation. The following status flags have been provided in 8085.

Carry (CS): It holds carry out of the most significant bit resulting from the execution of an arithmetic operation. If there is a carry from addition or a borrow from subtraction or comparison the carry flag (S is set to 1) otherwise 0.

Zero (Z): The zero status flag Z is set to 1 if the result of an arithmetic or logical operation is zero. For non-zero result it is set to 0.

Sign (S): The sign status flag is set to 1 if the most significant bit of the result of an arithmetic or logical operations is 1, otherwise 0.

Parity (P): The parity status flag is set to 1 if the most significant bit of the result of the operation contains even number of 1s. It is set to zero when there is odd number of 1s.

Auxiliary Carry (AC): The auxiliary carry status flag holds carry out of bit 3 to 4 resulting from the execution of an arithmetic operation.

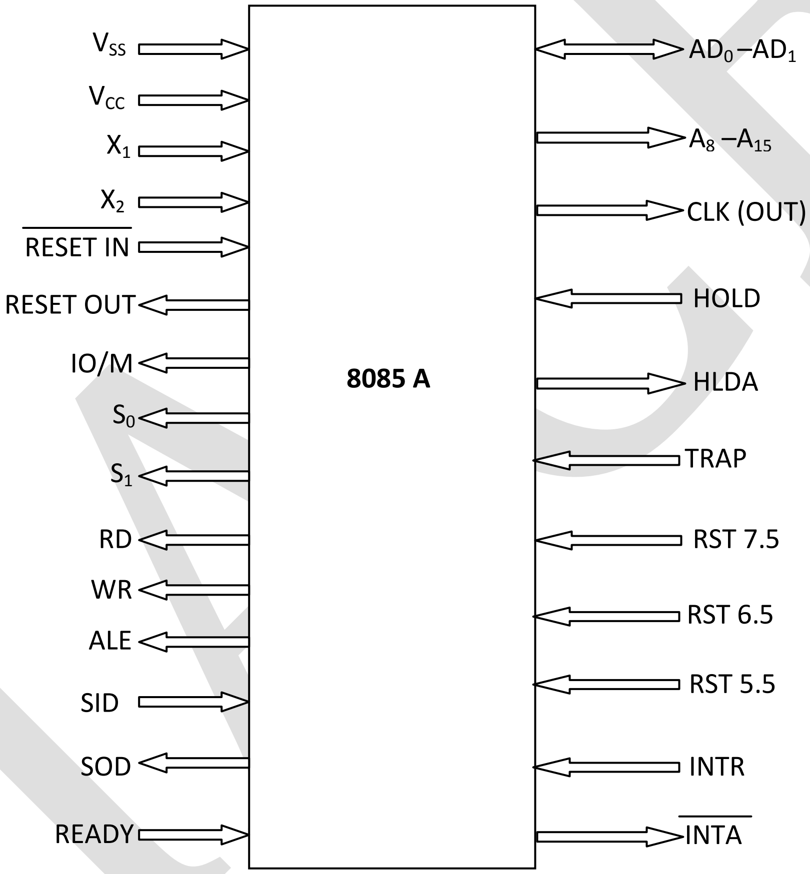
Data and Address Bus: The 8085 is an eight bit microprocessor. Its data bus is 8-bit wide and hence 8-bits of data can be transmitted in parallel from or to the microprocessor. The 8085 microprocessor requires 16-bit wide address bus as the memory addresses are of 16-bits. The 8 most significant bits of the address are transmitted by the address bus. A bus (Pins A_8 to A_{15}). The 8 least significant bits of the address/data bus transmits data and address at different moments. At a particular moment it transmits either data or address. Thus the AD bus operates in time shared mode. This techniques is known as multiplexing.

PIN CONFIGURATION: A_8 –A_{15} (Output): These are address bus and are used for the most significant bits of the memory address or 8 bit of I/O address.

AD_0 – AD_7 (input /output): These are time multiplexed address/data bus i.e. they serve dual purpose. They are used for the least significant 8-bits of the memory address or I/O address during the first clock cycle of a machine cycle. Again they are used for data during second and third clock cycles.

ALE (output): It is an address latch enable signal. It goes high during first clock cycle of a machine cycle and enables the lower 8-bits of the address to be latched either into the memory or external latch.

IO/$\overline{M}$ (Output): It is a status signal which distinguishers whether the address is for memory or I/O. When it goes high the address on the address bus is for I/O devices.



S_0, S_1 (output): These are status signals sent by the microprocessor to distinguish the various types of operation as follow.

S_1	S_0	Operations
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

RD (output): It is a signal to control READ operation. When it goes low the selected memory or I/O device is read.

$\overline{WR}$ (output): It is a signal to control WRITE operation. When it goes low the data on the data bus is written into the selected memory or I/O location.

READY (input): It is used by the microprocessor to sense whether a peripheral is ready to transfer data or not. If READY is high the peripheral is ready. If it is low the microprocessor waits till it goes high.

HOLD (input): It indicates that another device is requesting the use of the address and data bus.

HLAD (output): It is a signal for HOLD acknowledgement.

Program Counter (PC) and Stack Pointer (SP)

Program Counter: The microprocessor uses the PC register to sequence the execution of the instructions. PC is used to point the memory address from which the next byte is to be fetched. When a byte is being fetched, the program counter is incremented by one to point to the next memory location.

Stack Pointer: It is 16 bit register used as a memory pointer. It points to a memory location is R/W memory called the stack.

The 8085 Instructions Set: An instruction is a binary pattern designed inside a microprocessor to perform a specified function. The entire group of instructions called the instruction set, determines what function the microprocessor can perform.

The 8085 instructions set can be tabulated into five functional categories.

1. **Data Transfer (copy) Operations:**

This group of instructions copies data from one location to another location i.e. from source location to destination location.

This various types of data transfer are as follows.

(i) Between registers (ii) Specific data byte to a register

(iii) Between memory location and a register

(iv) Between an Input / Output device and the accumulator.

2. **Arithmetic Operations:** These instructions perform arithmetic operation such as addition, subtraction, increment and decrement.

Addition: In this any 8 bit number, content of the register, the contents of a memory location can be added to the content present in the accumulator and result is stored in the accumulator.

Subtraction: In this any 8 bit number, content of the register, the contents of a memory location can be subtracted to the content present in the accumulator and result is stored in the accumulator.

Increment / Decrement: Memory location or 8 bit contents of a register can be incremented or decremented by 1. Also the contents of the 16 bit register pair can be incremented or decremented by 1. They can be performed in any one of the register or in a memory location.

3. **The logical Operations:**

(i) **AND, OR, Exclusive-OR:** In this any 8 bit number, content of the register, the content of memory can be ANDed, ORed, or Exclusive-ORed with the contents present in the accumulator. The result is stored in the accumulator.

(ii) **Rotate:** Each bit in the accumulator can be shifted to the left or the right position.

(iii) **Compare:** Instructions can be compared for equality, greater than, or less than, with the contents of the accumulator.

(iv) **Complement:** The contents of the accumulator can be complemented all 0's are placed by 1s and all 1s are replaced by 0s.

4. **Branching Operations:**

(i) **Jump:** These instructions test for a certain condition and alter the program sequence when the condition is met.

(ii) **Call:** Return and Restart: These instructions change the sequence of a program either by calling a subroutine or returning from subroutine.

5. **Machine Control Operations:** These instructions control machine functions such as Halt, Interrupt or do nothing.

***** IACE *****