

CHAPTER 1

MICROPROCESSOR ARCHITECTURE AND ITS OPERATIONS

A microprocessor is a computer processor that incorporates the functions of a central processing unit on a single integrated circuit (IC), or at most a few integrated circuits. The microprocessor is a multipurpose, clock driven, register based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory and provides results as output. Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary number system.

The integration of a whole CPU onto a single or a few integrated circuits greatly reduced the cost of processing power. Integrated circuit processors are produced in large numbers by highly automated processes, resulting in a low unit price. Single-chip processors increase reliability because there are many fewer electrical connections that could fail. As microprocessor designs improve, the cost of manufacturing a chip (with smaller components built on a semiconductor chip the same size) generally stays the same according to Rock's law.

Before microprocessors, small computers had been built using racks of circuit boards with many medium- and small-scale integrated circuits. Microprocessors combined this into one or a few large-scale ICs. Continued increases in microprocessor capacity have since rendered other forms of computers almost completely obsolete (see history of computing hardware), with one or more microprocessors used in everything from the smallest embedded systems and handheld devices to the largest mainframes and supercomputers.

Evolution of Microprocessor

The microprocessor has become more essential part of many gadgets. The evolution of microprocessors was divided into five generations such as first, second, third, fourth and fifth generation and the characteristics of these generations are discussed below.

First Generation Microprocessors

The first generation microprocessors were introduced in the year 1971-1972. The instructions of these microprocessors were processed serially, they fetched the instruction, decoded and then executed it. When an instruction of the microprocessor was finished, then the microprocessor updates the instruction pointer & fetched the following instruction, performing this consecutive operation for each instruction in turn.

Second Generation Microprocessors

In the year 1970, small amount of transistors were available on the integrated circuit in the second generation microprocessors. Examples of the second generation microprocessors are 16-bit arithmetic 7 pipelined instruction processing, MC68000 Motorola microprocessor. These processors are introduced in the year 1979, and Intel 8080 processor is another example of the microprocessor. The second generation of the microprocessor is defined by overlapped fetch, decode and execute the steps. When the first generation is processed in the execution unit, then the second instruction is decoded and the third instruction is fetched.

The difference between the first generation microprocessor and second generation microprocessors was mainly the use of new semiconductor technologies to manufacture the chips. The result of this technology resulted in a fivefold increase in instruction, speed, execution and higher chip densities.

Third Generation Microprocessors

The third generation microprocessors were introduced in the year 1978, as denoted by Intel's 8086 and the Zilog Z8000. These were 16-bit processors with a performance like mini computers. These types of microprocessors were different from the previous generations of microprocessors in that all main workstation industrialists began evolving their own ISC based microprocessor architectures.

Fourth Generation Microprocessors

As many industries converted from commercial microprocessors to in house designs, the fourth generation microprocessors are entered with outstanding design with a million transistors. Leading edge microprocessors like Motorola's 88100 and Intel's 80960CA could issue & retire more than one instruction per clock cycle.

Fifth Generation Microprocessors

Fifth generation microprocessors employed decoupled super scalar processing, and their design soon exceeded 10 million transistors. In fifth generation, PCs are a low-margin, high volume business conquered by a single microprocessor.

Types of Microprocessor

Complex Instruction Set Microprocessors

The short term of Complex Instruction Set Microprocessors is CISM and they classify a microprocessor in which orders can be performed together along with other low level activities. These types of processors performs the different tasks like downloading, uploading, recalling data into the memory card and recalling data from the memory card. Apart from these tasks, it also does complex mathematical calculations in a single command.

Reduced Instruction Set Microprocessor

The short term of Reduced Instruction Set Microprocessor is RISC. These types of processors are made according to the function in which the microprocessor can carry out small things in specific command. In this way these processors completes more commands at a faster rate.

Superscalar Microprocessors

Superscalar processor facsimiles the hardware on the processor to perform various tasks at a time. These processors can be used for ALUs or multipliers. They have different operational units and these processors can carry out more than a one command by continuously transmitting several instructions to the extra operational units inside the processor.

The Application Specific Integrated Circuit

The short term of Application Specific Integrated Circuit processor is an ASIC. These processors are used for particular purposes that includes of automotive emissions control or personal digital assistant's computer. This type of processor is made with proper specification, but apart from these it can also be made with off the shelf gears.

Digital Signal Multiprocessors

Digital signal processors are also called as DSP's, these processors are used to encode and decode the videos or to convert the D/A (digital to analog) & A/D (analog to digital). They need a microprocessor that is excellent in mathematical calculations. The chips of this processor are employed in RADAR, home theaters, SONAR, audio gears, TV set top boxes and Mobile phones

There are many companies like Intel, Motorola, DEC (Digital Equipment Corporation), TI (Texas Instruments) associated with many microprocessors such as 8085 microprocessors, ASIC, CISM, RISC, DSPs and 8086 microprocessors like Intel

Advantages and Disadvantages of Microprocessors

The advantages of microprocessors are

The processing speed is high

Intelligence has been brought to systems

Flexible.

Compact size.

Easy maintenance

Complex mathematics

Some of the disadvantages of microprocessor are it might get overheated and the limitation of the microprocessor imposes on size of data.

The applications of the microprocessors mainly involve in controllers in home appliances, wireless communication equipments, office publication and automation, consumer electronic goods, calculators, accounting system, video games, industrial controllers and data acquisition systems.

Instrumentation : It is very useful in the field of instrumentation. Frequency counters, function generators, frequency synthesizers, spectrum analyses and many other instruments are available, when microprocessors are used as controller. It is also used in medical instrumentation.

Control : Microprocessor based controllers are available in home appliances, such as microwave oven, washing machine etc., microprocessors are being used in controlling various parameters like speed, pressure, temperature etc. These are used with the help of suitable transducers.

Communication : Microprocessors are being used in a wide range of communication equipments. In telephone industry, these are used in digital telephone sets. Telephone exchanges and modem etc. The use of microprocessor in television, satellite communication have made teleconferencing possible. Railway reservation and air reservation system also uses this technology. LAN and WAN for communication of vertical information through computer network.

Consumer : The use of microprocessor in toys, entertainment equipment and home applications is making them more entertaining and full of features. The use of microprocessors is more widespread and popular. Now the Microprocessors are used in :

Calculators

Accounting system

Games machine

Complex Industrial Controllers

Traffic light Control

Data acquisition systems

Multi user, multi-function environments

Military applications

Communication system

Advantages of microprocessor

Microprocessor is that these are general purpose electronic processing devices which can be programmed to execute a number of tasks.

Microprocessor is its speed, which is measured in **hertz**. For instance, a microprocessor with 3 GHz, shortly GHz is capable of performing 3 billion tasks per second.

Microprocessor is that it can quickly move data between the various memory locations.

Disadvantages of microprocessor

The microprocessor has a limitation on the size of data.

Most of the microprocessor does not support floating point operations.

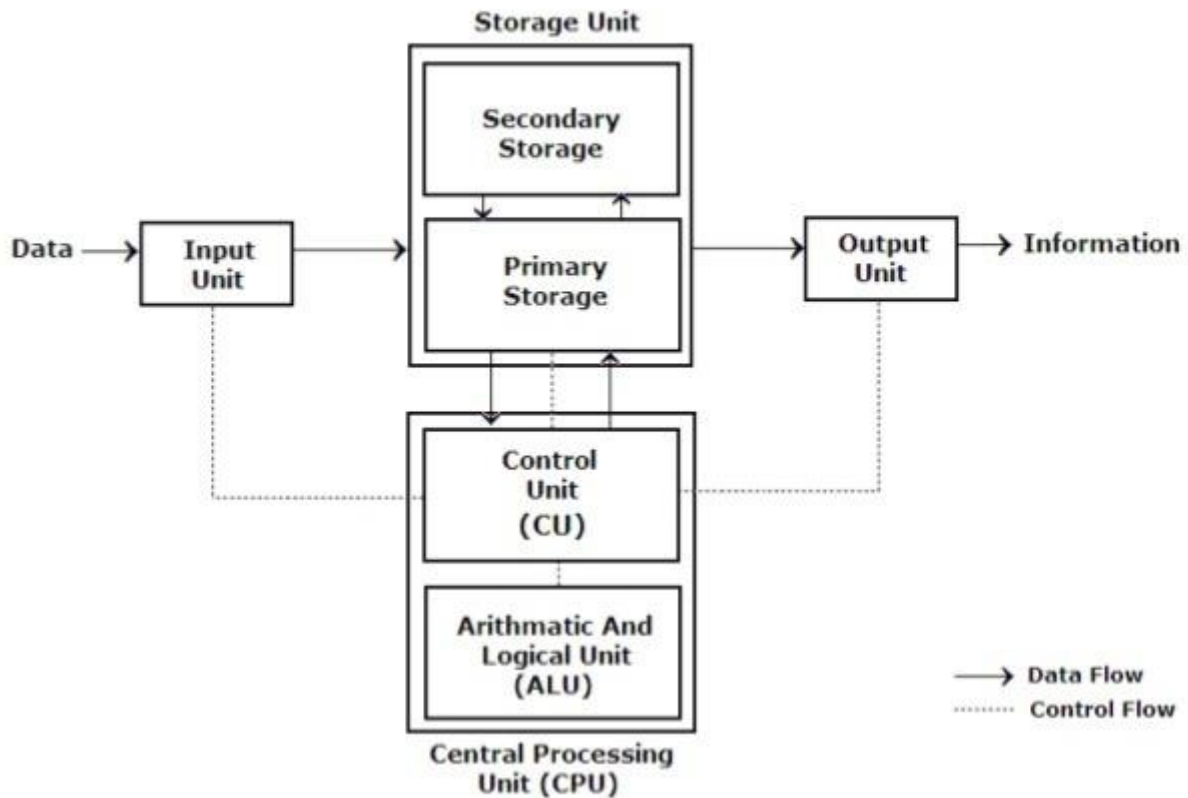
The main disadvantage is it's over heating physically.

It should not contact with the other external devices.

The microprocessor is does not have any internal peripheral like ROM, RAM and other I/O devices.

Schematic diagram of digital computer system

Block diagram of computer



1. Input Unit: The data and instructions are inputted into the computer using keyboard which is one of the most commonly used input devices. Other commonly used input devices are mouse, floppy disk drive, magnetic tape etc. Thus, we can conclude that, all the input devices accepts the data and instruction from outside world, convert it to a form that the computer can understand, supply the converted data to the computer system for further processing.

2. Storage Unit: The storage unit of a computer holds data and instructions that are entered through the input unit, before they are processed. It stores programs, data as well as intermediate results and results for output. Its main function is to store information.

The various storage devices can be divided into two main categories:

(a) Primary Storage (Main Memory):

This memory is generally used to hold the program being currently executed in the computer, the data being received from input device, the intermediate and final results of a program. The primary memory is temporary in nature. The data is lost when the computer is switched off. The data is electronically represented in the main memory chip's circuitry and while it remains in the main memory, central processing unit (CPU) can access it directly at a very fast speed. Primary storage has limited storage capacity because it is very expensive and generally made up of semiconductor devices.

(b) Secondary Storage (Auxiliary Memory):

It is used like an archive. It stores operating system, data files, compilers, assemblers, application programs etc. The Program and data if needed by central processing unit (CPU) are first transferred from the secondary memory to main memory. The results are also stored in secondary memory to main memory. It is a mass storage memory, slower but cheaper. It is non-volatile in nature i.e. data is not lost even if the power supply is switched off. Some of the most commonly used secondary storage devices are Hard Disk, Compact Disk etc. Their access time is in milliseconds.

3. Central Processing Unit (CPU) :

The control unit and arithmetic logic unit of computer are together known as central processing unit (CPU). The CPU is like brain and performs following functions: It performs all calculations, it takes all decisions, and it controls all units of a computer. A personal computer may have CPU-IC such as Intel 8088, 80386, 80486, Celeron, Pentium, Pentium Pro, Pentium II, III, IV, Dual Core, AMD etc.z

(a) Control Unit:

It controls all other units in a computer. The control unit instructs the input unit where to store the data after receiving it from user. It controls the flow of data and instructions from the storage unit to Arithmetic logic unit (ALU), it does not perform any actual processing of data. It manages and coordinates the entire computer system and synchronizes its working, thus referred to as “Central Nervous System” or “Brain of the Computer”.

(b) Arithmetic and Logic Unit:

The function of an Arithmetic logic unit (ALU) is to perform arithmetic and logical operations such as addition, subtraction, multiplication, division, AND, OR, NOT, Exclusive OR etc. It also performs increment, decrements, shift and clear operations. It is the place where the actual execution of instructions takes place during processing operation. To be precise, all calculations and comparisons are made in the Arithmetic and Logical Unit (ALU).\

4. Output Unit:

An output unit performs the reverse operation of that of an input unit. As computers, work with binary code the results produced are also in binary form. So before applying results to the outside world it must be converted to human acceptable (readable) form. So, it supplies information obtained from processing to outside world. Units called output interfaces accomplish this task

8085 is pronounced as "eighty-eighty-five" microprocessor. It is an 8-bit microprocessor designed by Intel in 1977 using NMOS technology.

It has the following configuration –

8-bit data bus

16-bit address bus, which can address upto 64KB

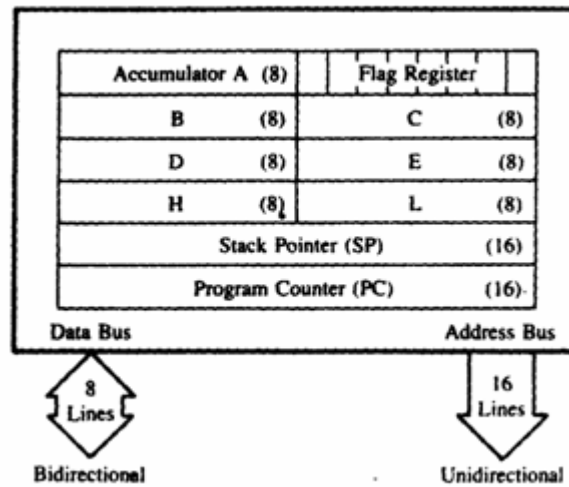
A 16-bit program counter

A 16-bit stack pointer

Six 8-bit registers arranged in pairs: BC, DE, HL

Requires +5V supply to operate at 3.2 MHZ single phase clock

The Registers of 8085 Microprocessor:



A microprocessor is a multipurpose, programmable, clock-driven, register-based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provide results as output. A 8085 microprocessor, is a second generation 8-bit microprocessor and is the base for studying and using all the microprocessor available in the market.

Registers in 8085:

(a) General Purpose Registers –

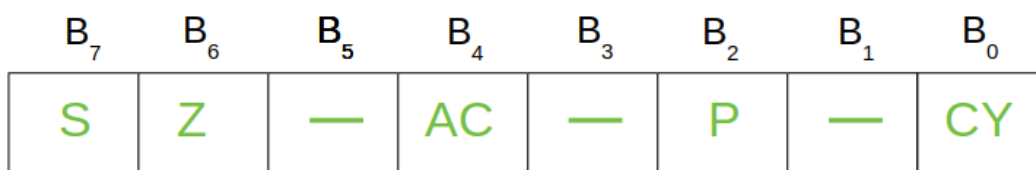
The 8085 has six general-purpose registers to store 8-bit data; these are identified as- B, C, D, E, H, and L. These can be combined as register pairs – BC, DE, and HL, to perform some 16-bit operation. These registers are used to store or copy temporary data, by using instructions, during the execution of the program.

(b) Specific Purpose Registers –

Accumulator:

The accumulator is an 8-bit register (can store 8-bit data) that is the part of the arithmetic and logical unit (ALU). After performing arithmetical or logical operations, the result is stored in accumulator. Accumulator is also defined as register A.

Flag registers:



fig(a)-Bit position of various flags in flag registers of 8085

The flag register is a special purpose register and it is completely different from other registers in microprocessor. It consists of 8 bits and only 5 of them are useful. The other three are left vacant and are used in the future Intel versions. These 5 flags are set or reset (when value of flag is 1, then it is said to be set and when value is 0, then it is said to be reset) after an operation according to data condition of the result in the accumulator and other registers. The 5 flag registers are:

Sign Flag: It occupies the seventh bit of the flag register, which is also known as the most significant bit. It helps the programmer to know whether the number stored in the accumulator is positive or negative. If the sign flag is set, it means that number stored in the accumulator is negative, and if reset, then the number is positive.

Zero Flag: It occupies the sixth bit of the flag register. It is set, when the operation performed in the ALU results in zero (all 8 bits are zero), otherwise it is reset. It helps in determining if two numbers are equal or not.

Auxillary Carry Flag: It occupies the fourth bit of the flag register. In an arithmetic operation, when a carry flag is generated by the third bit and passed on to the fourth bit, then Auxillary Carry flag is set. If not flag is reset. This flag is used internally for BCD(Binary-Coded decimal Number) operations.

Note – This is the only flag register in 8085 which is not accessible by user.

Parity FlagL: It occupies the second bit of the flag register. This flag tests for number of 1's in the accumulator. If the accumulator holds even number of 1's, then this flag is set and it is said to even parity. On the other hand if the number of 1's is odd, then it is reset and it is said to be odd parity.

Carry Flag: It occupies the zeroth bit of the flag register. If the arithmetic operation results in a carry(if result is more than 8 bit), then Carry Flag is set; otherwise it is reset.

(c) Memory Registers –

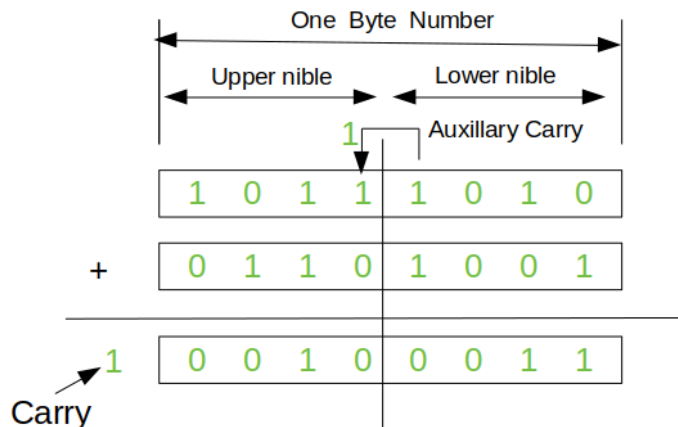
There are two 16-bit registers used to hold memory addresses. The size of these registers is 16 bits because the memory addresses are 16 bits. They are :-

Program Counter: This register is used 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. When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location.

Stack Pointer: It is used as a memory pointer. It points to a memory location in read/write memory, called the stack. It is always incremented/decremented by 2 during push and pop operation.

Example

Here two binary numbers are added. The result produced is stored in the accumulator. Now lets check what each bit means. Refer to the below explanation simultaneously to connect them with the example.



B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
0	0	—	1	—	0	—	1

Value of flags

Sign Flag (7th bit): It is reset(0), which means number stored in the accumulator is positive.

Zero Flag (6th bit): It is reset(0), thus result of the operations performed in the ALU is non-zero.

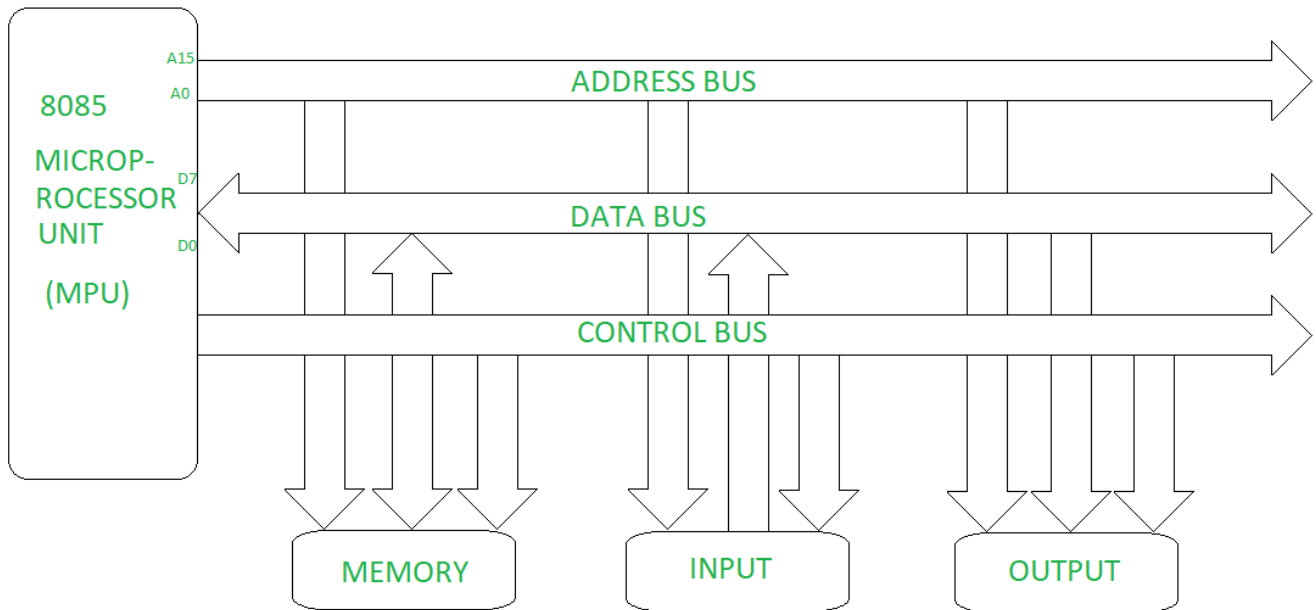
Auxiliary Carry Flag (4th bit): We can see that b3 generates a carry which is taken by b4, thus auxiliary carry flag gets set (1).

Parity Flag (2nd bit): It is reset(0), it means that parity is odd. The accumulator holds odd number of 1's.

Carry Flag (0th bit): It is set(1), output results in more than 8 bit.

System bus of 8085 Microprocessor System

Bus is a group of conducting wires which carries information, all the peripherals are connected to microprocessor through Bus.



Bus organization system of 8085 Microprocessor

There are three types of buses.

1. Address bus –

It is a group of conducting wires which carries address only. Address bus is unidirectional because data flow in one direction, from microprocessor to memory or from microprocessor to Input/output devices (That is, Out of Microprocessor).

Length of Address Bus of 8085 microprocessor is 16 Bit (That is, Four Hexadecimal Digits), ranging from 0000 H to FFFF H, (H denotes Hexadecimal). The microprocessor 8085 can transfer maximum 16 bit address which means it can address 65, 536 different memory location.

The Length of the address bus determines the amount of memory a system can address. Such as a system with a 32-bit address bus can address 2^{32} memory locations. If each memory location holds one byte, the addressable memory space is 4 GB. However, the actual amount of memory that can be accessed is usually much less than this theoretical limit due to chipset and motherboard limitations.

2. Data bus –

It is a group of conducting wires which carries Data only. Data bus is bidirectional because data flow in both directions, from microprocessor to memory or Input/Output devices and from memory or Input/Output devices to microprocessor.

Length of Data Bus of 8085 microprocessor is 8 Bit (That is, two Hexadecimal Digits), ranging from 00 H to FF H. (H denotes Hexadecimal).

When it is write operation, the processor will put the data (to be written) on the data bus, when it is read operation, the memory controller will get the data from specific memory block and put it into the data bus.

The width of the data bus is directly related to the largest number that the bus can carry, such as an 8 bit bus can represent 2 to the power of 8 unique values, this equates to the number 0 to 255. A 16 bit bus can carry 0 to 65535.

3. Control bus –

It is a group of conducting wires, which is used to generate timing and control signals to control all the associated peripherals, microprocessor uses control bus to process data, that is what to do with selected memory location. Some control signals are:

Memory read

Memory write

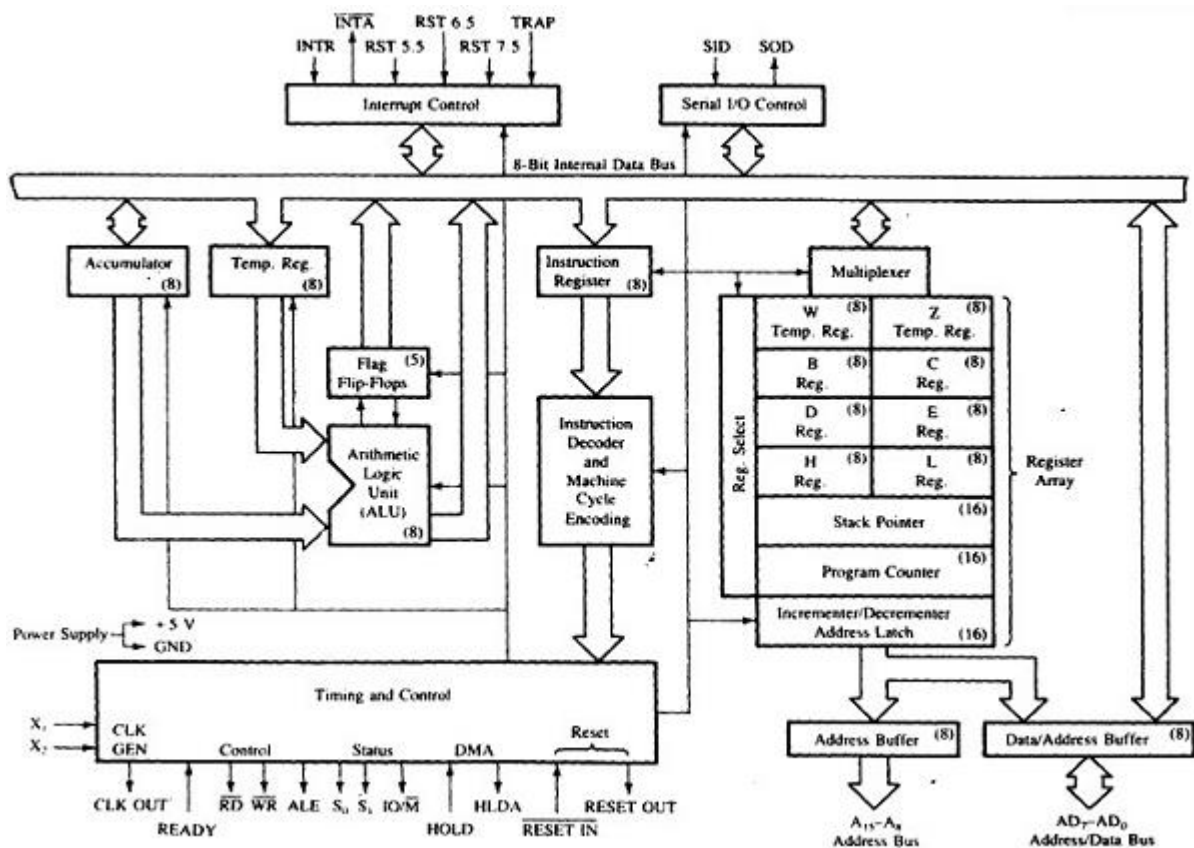
I/O read

I/O Write

Opcode fetch

If one line of control bus may be the read/write line. If the wire is low (no electricity flowing) then the memory is read, if the wire is high (electricity is flowing) then the memory is written.

Architecture of 8085 Microprocessor System



8085 consists of the following functional units –

Accumulator

It is an 8-bit register used to perform arithmetic, logical, I/O & LOAD/STORE operations. It is connected to internal data bus & ALU.

Arithmetic and logic unit

As the name suggests, it performs arithmetic and logical operations like Addition, Subtraction, AND, OR, etc. on 8-bit data.

General purpose register

There are 6 general purpose registers in 8085 processor, i.e. B, C, D, E, H & L. Each register can hold 8-bit data. These registers can work in pair to hold 16-bit data and their pairing combination is like B-C, D-E & H-L.

Program counter

It is a 16-bit register used to store the memory address location of the next instruction to be executed. Microprocessor increments the program whenever an instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

Stack pointer

It is also a 16-bit register works like stack, which is always incremented/decremented by 2 during push & pop operations.

Temporary register

It is an 8-bit register, which holds the temporary data of arithmetic and logical operations.

Flag register

It is an 8-bit register having five 1-bit flip-flops, which holds either 0 or 1 depending upon the result stored in the accumulator.

These are the set of 5 flip-flops –

Sign (S)

Zero (Z)

Auxiliary Carry (AC)

Parity (P)

Carry (C)

Instruction register and decoder

It is an 8-bit register. When an instruction is fetched from memory then it is stored in the Instruction register. Instruction decoder decodes the information present in the Instruction register.

Timing and control unit

It provides timing and control signal to the microprocessor to perform operations. Following are the timing and control signals, which control external and internal circuits –

Control Signals: READY, RD', WR', ALE

Status Signals: S0, S1, IO/M'

DMA Signals: HOLD, HLDA

RESET Signals: RESET IN, RESET OUT

Interrupt control

As the name suggests it controls the interrupts during a process. When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

There are 5 interrupt signals in 8085 microprocessor: INTR, RST 7.5, RST 6.5, RST 5.5, TRAP.

Serial Input/output control

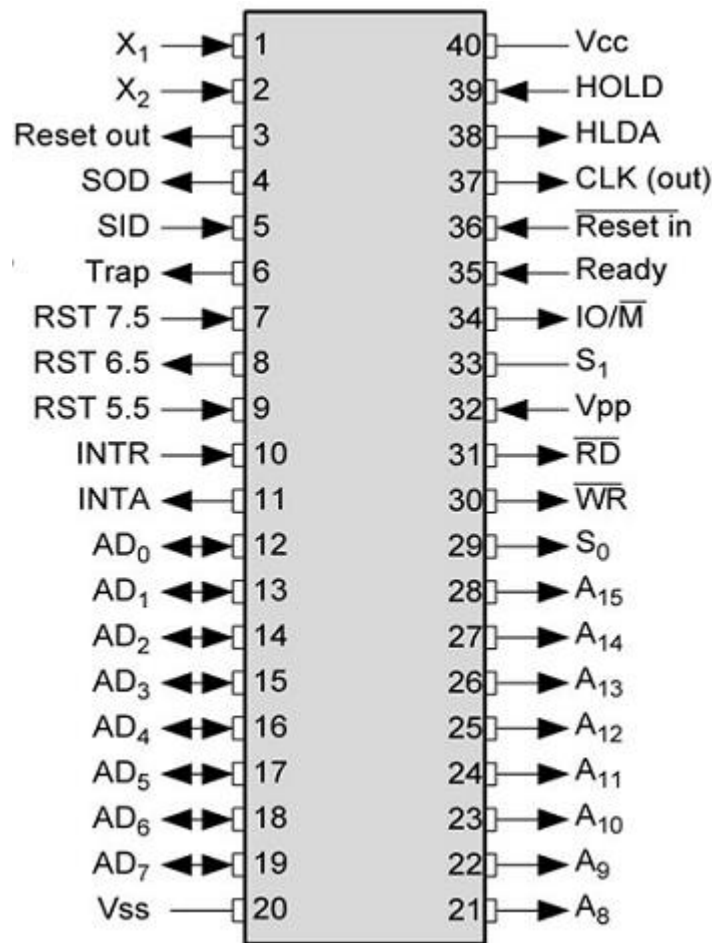
It controls the serial data communication by using these two instructions: SID (Serial input data) and SOD (Serial output data).

Address buffer and address-data buffer

The content stored in the stack pointer and program counter is loaded into the address buffer and address-data buffer to communicate with the CPU. The memory and I/O chips are connected to these buses; the CPU can exchange the desired data with the memory and I/O chips.

Address bus and data bus

Data bus carries the data to be stored. It is bidirectional, whereas address bus carries the location to where it should be stored and it is unidirectional. It is used to transfer the data & Address I/O devices.



The pins of a 8085 microprocessor can be classified into seven groups –

Address bus

A_{15} - A_8 , it carries the most significant 8-bits of memory/IO address.

Data bus

AD_7 - AD_0 , it carries the least significant 8-bit address and data bus.

Control and status signals

These signals are used to identify the nature of operation. There are 3 control signal and 3 status signals.

Three control signals are RD, WR & ALE.

RD – This signal indicates that the selected IO or memory device is to be read and is ready for accepting data available on the data bus.

WR – This signal indicates that the data on the data bus is to be written into a selected memory or IO location.

ALE – It is a positive going pulse generated when a new operation is started by the microprocessor. When the pulse goes high, it indicates address. When the pulse goes down it indicates data.

Three status signals are IO/M, S0 & S1.

IO/M

This signal is used to differentiate between IO and Memory operations, i.e. when it is high indicates IO operation and when it is low then it indicates memory operation.

S1 & S0

These signals are used to identify the type of current operation.

Power supply

There are 2 power supply signals – VCC & VSS. VCC indicates +5v power supply and VSS indicates ground signal.

Clock signals

There are 3 clock signals, i.e. X1, X2, CLK OUT.

X1, X2 – A crystal (RC, LC N/W) is connected at these two pins and is used to set frequency of the internal clock generator. This frequency is internally divided by 2.

CLK OUT – This signal is used as the system clock for devices connected with the microprocessor.

Interrupts & externally initiated signals

Interrupts are the signals generated by external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. We will discuss interrupts in detail in interrupts section.

INTA – It is an interrupt acknowledgment signal.

RESET IN – This signal is used to reset the microprocessor by setting the program counter to zero.

RESET OUT – This signal is used to reset all the connected devices when the microprocessor is reset.

READY – This signal indicates that the device is ready to send or receive data. If READY is low, then the CPU has to wait for READY to go high.

HOLD – This signal indicates that another master is requesting the use of the address and data buses.

HLDA (HOLD Acknowledge) – It indicates that the CPU has received the HOLD request and it will relinquish the bus in the next clock cycle. HLDA is set to low after the HOLD signal is removed.

Serial I/O signals

There are 2 serial signals, i.e. SID and SOD and these signals are used for serial communication.

SOD (Serial output data line) – The output SOD is set/reset as specified by the SIM instruction.

SID (Serial input data line) – The data on this line is loaded into accumulator whenever a RIM instruction is executed.

Addressing Modes in 8085

These are the instructions used to transfer the data from one register to another register, from the memory to the register, and from the register to the memory without any alteration in the content. Addressing modes in 8085 is classified into 5 groups –

Immediate addressing mode

In this mode, the 8/16-bit data is specified in the instruction itself as one of its operand.

For example: MVI K, 20F: means 20F is copied into register K.

Register addressing mode

In this mode, the data is copied from one register to another.

For example: MOV K, B: means data in register B is copied to register K.

Direct addressing mode

In this mode, the data is directly copied from the given address to the register. For example: LDB 5000K: means the data at address 5000K is copied to register B.

Indirect addressing mode

In this mode, the data is transferred from one register to another by using the address pointed by the register. For example: MOV K, B: means data is transferred from the memory address pointed by the register to the register K.

Implied addressing mode

This mode doesn't require any operand; the data is specified by the opcode itself. For example: CMP.

Interrupts in 8085

Interrupts are the signals generated by the external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR.

Interrupt are classified into following groups based on their parameter –

Vector interrupt – In this type of interrupt, the interrupt address is known to the processor. For example: RST7.5, RST6.5, RST5.5, TRAP.

Non-Vector interrupt – In this type of interrupt, the interrupt address is not known to the processor so, the interrupt address needs to be sent externally by the device to perform interrupts. For example: INTR.

Maskable interrupt – In this type of interrupt, we can disable the interrupt by writing some instructions into the program. For example: RST7.5, RST6.5, RST5.5.

Non-Maskable interrupt – In this type of interrupt, we cannot disable the interrupt by writing some instructions into the program. For example: TRAP.

Software interrupt – In this type of interrupt, the programmer has to add the instructions into the program to execute the interrupt. There are 8 software interrupts in 8085, i.e. RST0, RST1, RST2, RST3, RST4, RST5, RST6, and RST7.

Hardware interrupt – There are 5 interrupt pins in 8085 used as hardware interrupts, i.e. TRAP, RST7.5, RST6.5, RST5.5, INTA.

Note – NTA is not an interrupt, it is used by the microprocessor for sending acknowledgement. TRAP has the highest priority, then RST7.5 and so on.

Interrupt Service Routine (ISR)

A small program or a routine that when executed, services the corresponding interrupting source is called an ISR.

TRAP

It is a non-maskable interrupt, having the highest priority among all interrupts. By default, it is enabled until it gets acknowledged. In case of failure, it executes as ISR and sends the data to backup memory. This interrupt transfers the control to the location 0024H.

RST7.5

It is a maskable interrupt, having the second highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 003CH address.

RST 6.5

It is a maskable interrupt, having the third highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 0034H address.

RST 5.5

It is a maskable interrupt. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 002CH address.

INTR

It is a maskable interrupt, having the lowest priority among all interrupts. It can be disabled by resetting the microprocessor.

When INTR signal goes high, the following events can occur –

The microprocessor checks the status of INTR signal during the execution of each instruction.

When the INTR signal is high, then the microprocessor completes its current instruction and sends active low interrupt acknowledge signal.

When instructions are received, then the microprocessor saves the address of the next instruction on stack and executes the received instruction.

Timing Diagrams of 8085

It is one of the best way to understand to process of micro-processor/controller. With the help of timing diagram we can understand the working of any system, step by step working of each instruction and its execution, etc.

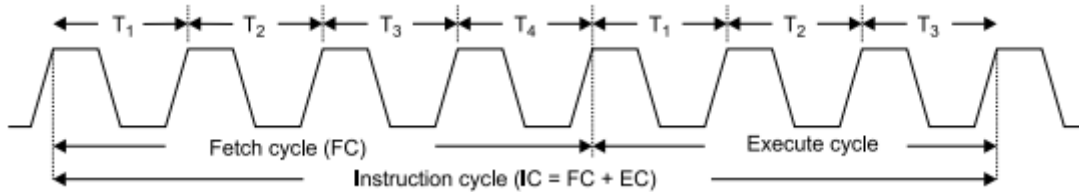
It is the graphical representation of process in steps with respect to time. The timing diagram represents the clock cycle and duration, delay, content of address bus and data bus, type of operation ie. Read/write/status signals.

Important terms related to timing diagrams:

1. Instruction cycle: this term is defined as the number of steps required by the cpu to complete the entire process ie. Fetching and execution of one instruction. The fetch and execute cycles are carried out in synchronization with the clock.
2. Machine cycle: It is the time required by the microprocessor to complete the operation of accessing the memory devices or I/O devices. In machine cycle various operations like opcode fetch, memory read, memory write, I/O read, I/O write are performed.
3. T-state: Each clock cycle is called as T-states.

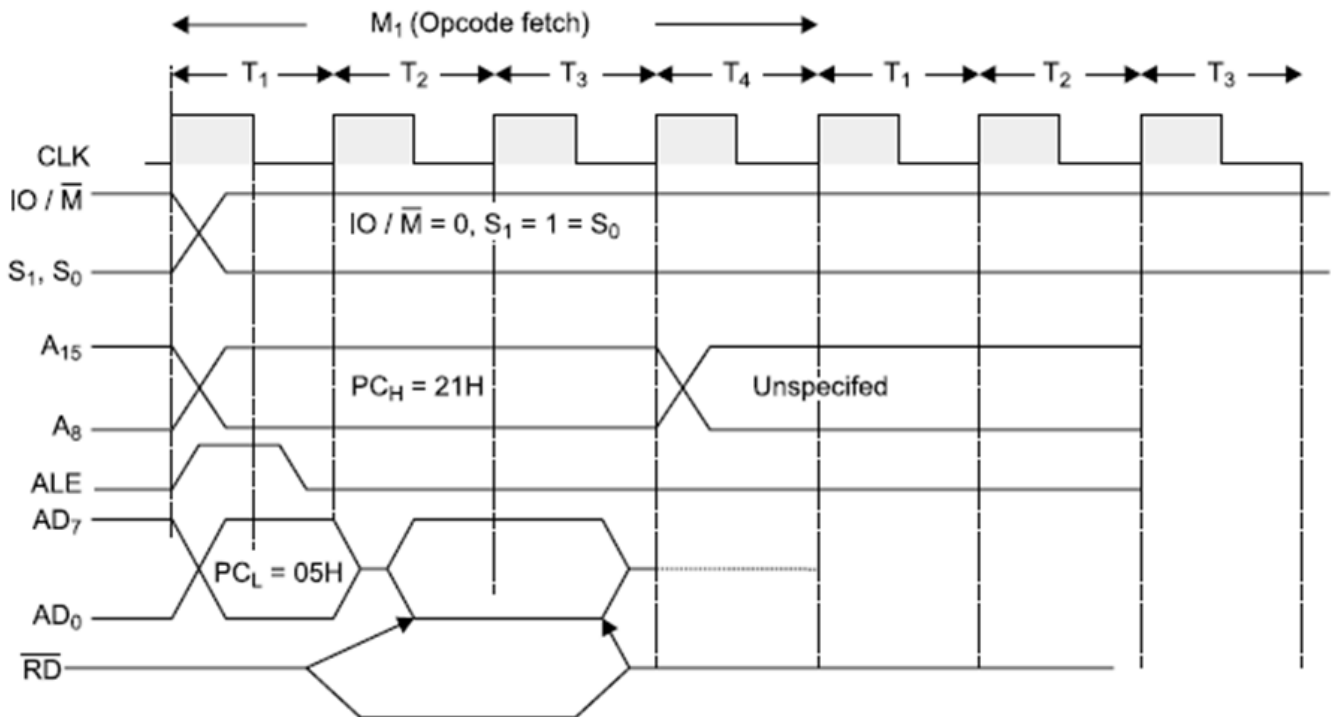
Rules to identify number of machine cycles in an instruction:

1. If an addressing mode is direct, immediate or implicit then No. of machine cycles = No. of bytes.
2. If the addressing mode is indirect then No. of machine cycles = No. of bytes + 1. Add +1 to the No. of machine cycles if it is memory read/write operation.
3. If the operand is 8-bit or 16-bit address then, No. of machine cycles = No. of bytes + 1.
4. These rules are applicable to 80% of the instructions of 8085.



Opcode fetch:

The microprocessor requires instructions to perform any particular action. In order to perform these actions microprocessor utilizes Opcode which is a part of an instruction which provides detail (ie. Which operation μp needs to perform) to microprocessor.

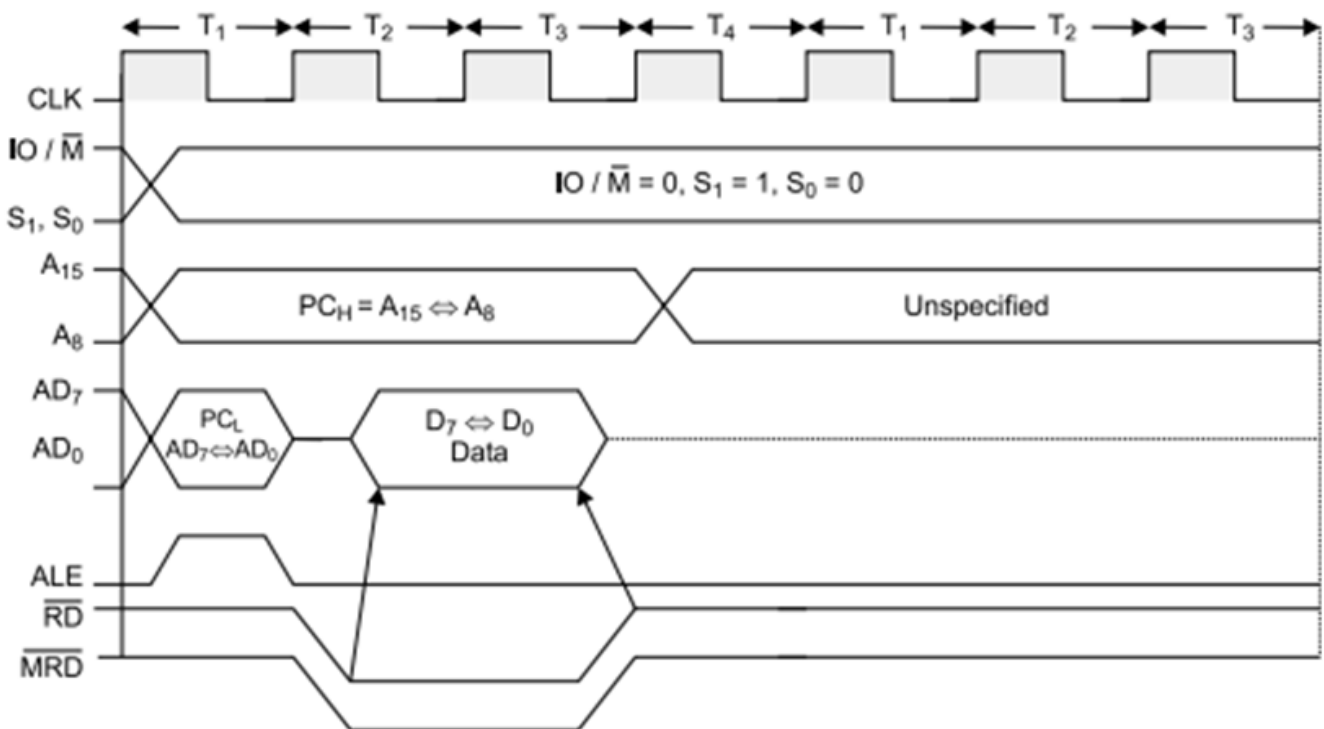


Operation:

- During T1 state, microprocessor uses IO/\overline{M} , S_0 , S_1 signals are used to instruct microprocessor to fetch opcode.

- Thus when $\text{IO}/\overline{\text{M}}=0$, $\text{S}_0=\text{S}_1=1$, it indicates opcode fetch operation.
- During this operation 8085 transmits 16-bit address and also uses ALE signal for address latching.
- At T2 state microprocessor uses read signal and make data ready from that memory location to read opcode from memory and at the same time program counter increments by 1 and points next instruction to be fetched.
- In this state microprocessor also checks READY input signal, if this pin is at low logic level ie. '0' then microprocessor adds wait state immediately between T2 and T3.
- At T3, microprocessor reads opcode and store it into instruction register to decode it further.
- During T4 microprocessor performs internal operation like decoding opcode and providing necessary actions.
- The opcode is decoded to know whether T5 or T6 states are required, if they are not required then μp performs next operation.

Read and write timing diagram for memory and I/O Operation

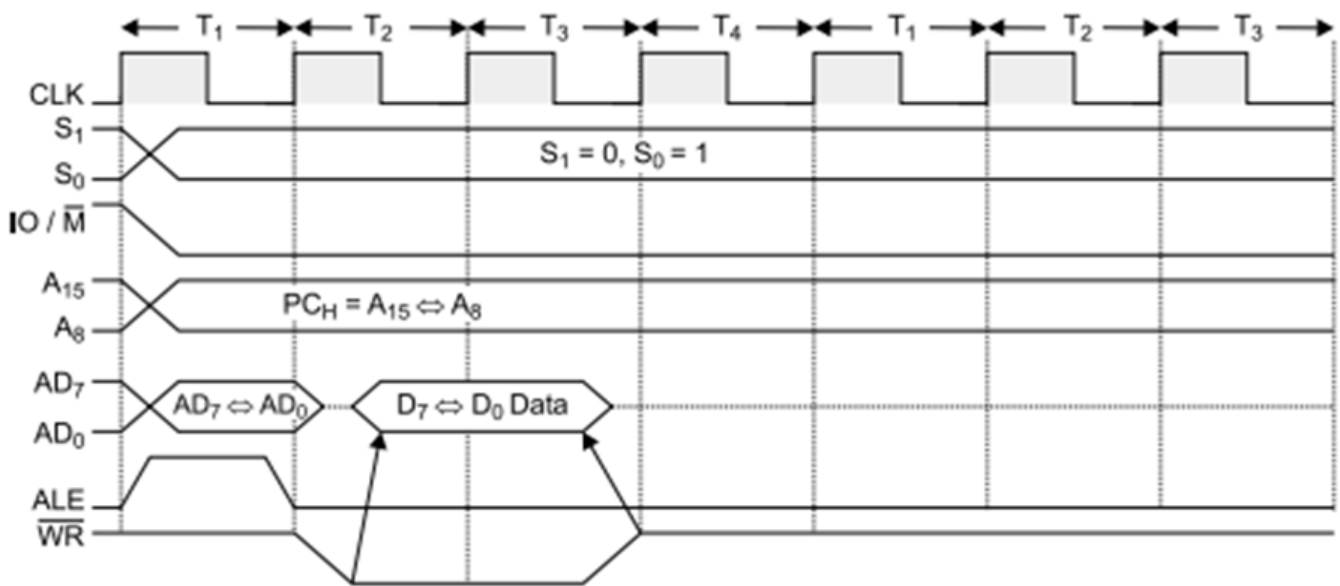


Operation:

- It is used to fetch one byte from the memory.
- It requires 3 T-States.
- It can be used to fetch operand or data from the memory.

- During T1, A8-A15 contains higher byte of address. At the same time ALE is high. Therefore Lower byte of address A0-A7 is selected from AD0-AD7.
- Since it is memory ready operation, IO/M(bar) goes low.
- During T2 ALE goes low, RD(bar) goes low. Address is removed from AD0-AD7 and data D0-D7 appears on AD0-AD7.
- During T3, Data remains on AD0-AD7 till RD(bar) is at low signal.

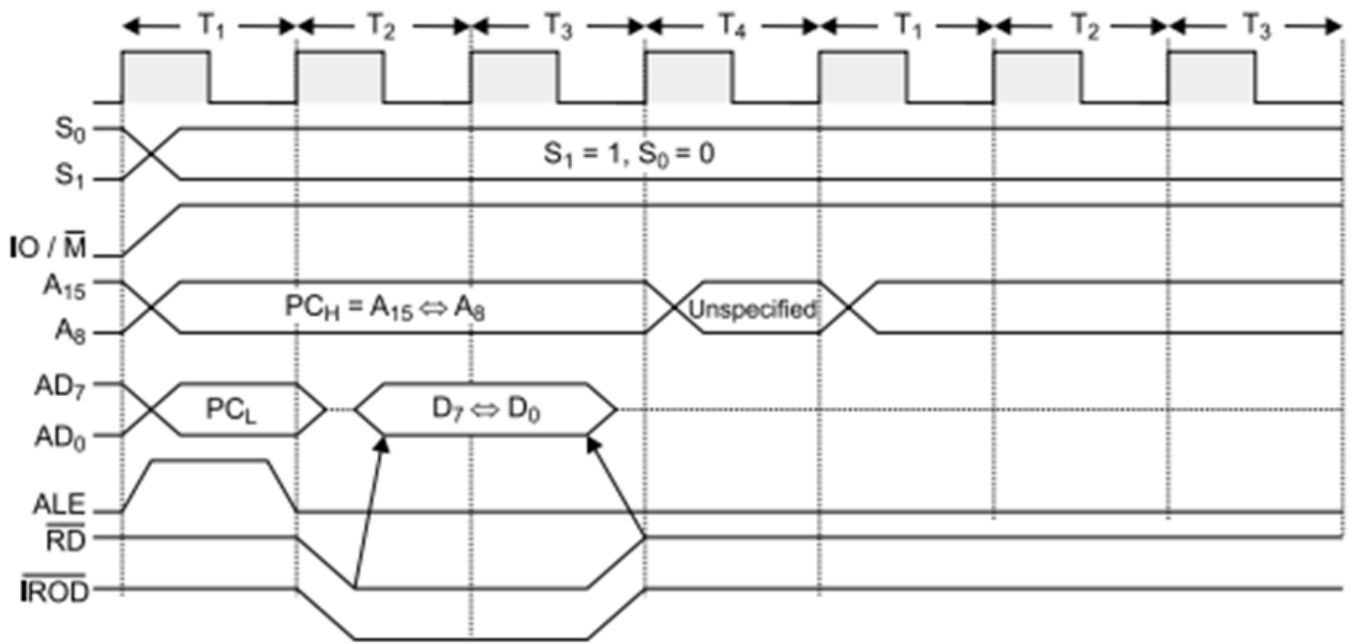
Memory Write:



Operation:

- It is used to send one byte into memory.
- It requires 3 T-States.
- During T1, ALE is high and contains lower address A0-A7 from AD0-AD7.
- A8-A15 contains higher byte of address.
- As it is memory operation, IO/M(bar) goes low.
- During T2, ALE goes low, WR(bar) goes low and Address is removed from AD0-AD7 and then data appears on AD0-AD7.
- Data remains on AD0-AD7 till WR(bar) is low.

IO Read:



Operation:

It is used to fetch one byte from an IO port.

It requires 3 T-States.

During T_1 , The Lower Byte of IO address is duplicated into higher order address bus A_8 - A_{15} .

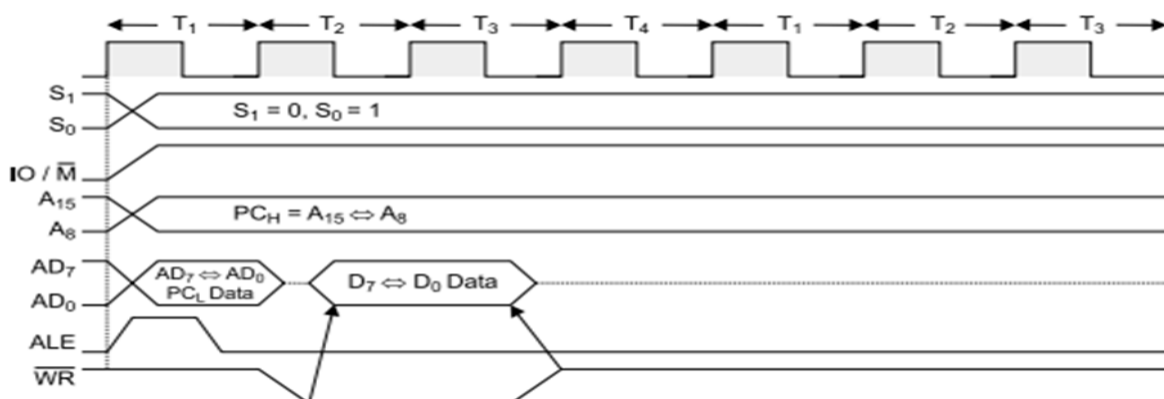
ALE is high and AD_0 - AD_7 contains address of IO device.

IO/\bar{M} (bar) goes high as it is an IO operation.

During T_2 , ALE goes low, \overline{RD} (bar) goes low and data appears on AD_0 - AD_7 as input from IO device.

During T_3 Data remains on AD_0 - AD_7 till \overline{RD} (bar) is low.

IO Write:



Operation:

- It is used to write one byte into IO device.
- It requires 3 T-States.
- During T1, the lower byte of address is duplicated into higher order address bus A8-A15.
- ALE is high and A0-A7 address is selected from AD0-AD7.
- As it is an IO operation IO/M (bar) goes low.
- During T2, ALE goes low, WR (bar) goes low and data appears on AD0-AD7 to write data into IO device.
- During T3, Data remains on AD0-AD7 till WR(bar) is low.

CHAPTER 2

INSTRUCTIONS SET OF 8085 MICROPROCESSOR AND ALP

Instruction Set of 8085

- ☐ An instruction is a binary pattern designed inside a microprocessor to perform a specific function.
- ☐ The entire group of instructions that a microprocessor supports is called Instruction Set.
- ☐ 8085 has 246 instructions.
- ☐ Each instruction is represented by an 8-bit binary value.
- ☐ These 8-bits of binary value is called Op-Code or Instruction Byte.

Classification of Instruction Set

- Data Transfer Instruction
- Arithmetic Instructions
- Logical Instructions
- Branching Instructions
- Control Instructions

Following is the table showing the list of Data-transfer instructions with their meanings.

Opcode	Operand	Meaning	Explanation
MOV	Rd, Sc M, Sc Dt, M	Copy from the source (Sc) to the destination(Dt)	This instruction copies the contents of the source register into the destination register without any alteration. Example – MOV K, L
MVI	Rd, data M, data	Move immediate 8-bit	The 8-bit data is stored in the destination register or memory. Example – MVI K, 55L
LDA	16-bit address	Load the accumulator	The contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator. Example – LDA 2034K
LDAX	B/D Reg. pair	Load the accumulator indirect	The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the accumulator. Example – LDAX K
LXI	Reg. pair, 16-bit data	Load the register pair immediate	The instruction loads 16-bit data in the register pair designated in the

			<p>register or the memory.</p> <p>Example – LXI K, 3225L</p>
LHLD	16-bit address	Load H and L registers direct	<p>The instruction copies the contents of the memory location pointed out by the address into register L and copies the contents of the next memory location into register H.</p> <p>Example – LHLD 3225K</p>
STA	16-bit address	16-bit address	<p>The contents of the accumulator are copied into the memory location specified by the operand.</p> <p>This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address.</p> <p>Example – STA 325K</p>
STAX	16-bit address	Store the accumulator indirect	<p>The contents of the accumulator are copied into the memory location specified by the contents of the operand.</p> <p>Example – STAX K</p>
SHLD	16-bit address	Store H and L registers direct	<p>The contents of register L are stored in the memory location specified by the 16-bit address in the operand and the contents of H register are stored into the next memory location by incrementing the operand.</p> <p>This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address.</p> <p>Example – SHLD 3225K</p>
XCHG	None	Exchange H and L with D and E	<p>The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E.</p> <p>Example – XCHG</p>
SPHL	None	Copy H and L registers to the	<p>The instruction loads the contents of the H and L registers into the stack pointer register. The contents of the</p>

		stack pointer	<p>H register provide the high-order address and the contents of the L register provide the low-order address.</p> <p>Example – SPHL</p>
XTHL	None	Exchange H and L with top of stack	<p>The contents of the L register are exchanged with the stack location pointed out by the contents of the stack pointer register.</p> <p>The contents of the H register are exchanged with the next stack location (SP+1).</p> <p>Example – XTHL</p>
PUSH	Reg. pair	Push the register pair onto the stack	<p>The contents of the register pair designated in the operand are copied onto the stack in the following sequence.</p> <p>The stack pointer register is decremented and the contents of the high order register (B, D, H, A) are copied into that location.</p> <p>The stack pointer register is decremented again and the contents of the low-order register (C, E, L, flags) are copied to that location.</p> <p>Example – PUSH K</p>
POP	Reg. pair	Pop off stack to the register pair	<p>The contents of the memory location pointed out by the stack pointer register are copied to the low-order register (C, E, L, status flags) of the operand.</p> <p>The stack pointer is incremented by 1 and the contents of that memory location are copied to the high-order register (B, D, H, A) of the operand.</p> <p>The stack pointer register is again incremented by 1.</p> <p>Example – POPK</p>
OUT	8-bit port address	Output the data from the accumulator to a port with 8bit address	<p>The contents of the accumulator are copied into the I/O port specified by the operand.</p> <p>Example – OUT K9L</p>

IN	8-bit port address	Input data to accumulator from a port with 8-bit address	The contents of the input port designated in the operand are read and loaded into the accumulator. Example – IN5KL
----	--------------------	--	--

Following is the table showing the list of Arithmetic instructions with their meanings.

Opcode	Operand	Meaning	Explanation
ADD	R M	Add register or memory, to the accumulator	The contents of the register or memory are added to the contents of the accumulator and the result is stored in the accumulator. Example – ADD K.
ADC	R M	Add register to the accumulator with carry	The contents of the register or memory & M the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator. Example – ADC K
ADI	8-bit data	Add the immediate to the accumulator	The 8-bit data is added to the contents of the accumulator and the result is stored in the accumulator. Example – ADI 55K
ACI	8-bit data	Add the immediate to the accumulator with carry	The 8-bit data and the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator. Example – ACI 55K
LXI	Reg. pair, 16bit data	Load the register pair immediate	The instruction stores 16-bit data into the register pair designated in the operand. Example – LXI K, 3025M
DAD	Reg. pair	Add the register pair to H and L registers	The 16-bit data of the specified register pair are added to the contents of the HL register. Example – DAD K
SUB	R	Subtract the	The contents of the register or the

	M	register or the memory from the accumulator	memory are subtracted from the contents of the accumulator, and the result is stored in the accumulator. Example – SUB K
SBB	R M	Subtract the source and borrow from the accumulator	The contents of the register or the memory & M the Borrow flag are subtracted from the contents of the accumulator and the result is placed in the accumulator. Example – SBB K
SUI	8-bit data	Subtract the immediate from the accumulator	The 8-bit data is subtracted from the contents of the accumulator & the result is stored in the accumulator. Example – SUI 55K
SBI	8-bit data	Subtract the immediate from the accumulator with borrow	The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E. Example – XCHG
INR	R M	Increment the register or the memory by 1	The contents of the designated register or the memory are incremented by 1 and their result is stored at the same place. Example – INR K
INX	R	Increment register pair by 1	The contents of the designated register pair are incremented by 1 and their result is stored at the same place. Example – INX K
DCR	R M	Decrement the register or the memory by 1	The contents of the designated register or memory are decremented by 1 and their result is stored at the same place. Example – DCR K
DCX	R	Decrement the register pair by 1	The contents of the designated register pair are decremented by 1 and their result is stored at the same place.

			Example – DCX K
DAA	None	Decimal adjust accumulator	<p>The contents of the accumulator are changed from a binary value to two 4-bit BCD digits.</p> <p>If the value of the low-order 4-bits in the accumulator is greater than 9 or if AC flag is set, the instruction adds 6 to the low-order four bits.</p> <p>If the value of the high-order 4-bits in the accumulator is greater than 9 or if the Carry flag is set, the instruction adds 6 to the high-order four bits.</p> <p>Example – DAA</p>

The following table shows the list of Logical instructions with their meanings.

Opcode	Operand	Meaning	Explanation
CMP	R M	Compare the register or memory with the accumulator	The contents of the operand (register or memory) are M compared with the contents of the accumulator.
CPI	8-bit data	Compare immediate with the accumulator	The second byte data is compared with the contents of the accumulator.
ANA	R M	Logical AND register or memory with the accumulator	The contents of the accumulator are logically AND with M the contents of the register or memory, and the result is placed in the accumulator.
ANI	8-bit data	Logical AND immediate with the accumulator	The contents of the accumulator are logically AND with the 8-bit data and the result is placed in the accumulator.
XRA	R M	Exclusive OR register or memory with the accumulator	The contents of the accumulator are Exclusive OR with M the contents of the register or memory, and the result is placed in the accumulator.
XRI	8-bit data	Exclusive OR immediate with the accumulator	The contents of the accumulator are Exclusive OR with the 8-bit data and the result is placed in the accumulator.
ORA	R M	Logical OR register or memory with the accumulator	The contents of the accumulator are logically OR with M the contents of the register or memory, and result is placed in the accumulator.
ORI	8-bit data	Logical OR	The contents of the accumulator are logically OR

		immediate with the accumulator	with the 8-bit data and the result is placed in the accumulator.
RLC	None	Rotate the accumulator left	Each binary bit of the accumulator is rotated left by one position. Bit D7 is placed in the position of D0 as well as in the Carry flag. CY is modified according to bit D7.
RRC	None	Rotate the accumulator right	Each binary bit of the accumulator is rotated right by one position. Bit D0 is placed in the position of D7 as well as in the Carry flag. CY is modified according to bit D0.
RAL	None	Rotate the accumulator left through carry	Each binary bit of the accumulator is rotated left by one position through the Carry flag. Bit D7 is placed in the Carry flag, and the Carry flag is placed in the least significant position D0. CY is modified according to bit D7.
RAR	None	Rotate the accumulator right through carry	Each binary bit of the accumulator is rotated right by one position through the Carry flag. Bit D0 is placed in the Carry flag, and the Carry flag is placed in the most significant position D7. CY is modified according to bit D0.
CMA	None	Complement accumulator	The contents of the accumulator are complemented. No flags are affected.
CMC	None	Complement carry	The Carry flag is complemented. No other flags are affected.
STC	None	Set Carry	Set Carry

The following table shows the list of Branching instructions with their meanings.

Opcode			Operand	Meaning	Explanation
JMP			16-bit address	Jump unconditionally	The program sequence is transferred to the memory address given in the operand.
Opcode	Description	Flag Status	16-bit address	Jump conditionally	The program sequence is transferred to the memory address given in the operand based on the specified flag of the PSW.
JC	Jump on Carry	CY=1			
JNC	Jump on no Carry	CY=0			

JP	Jump on positive	S=0			
JM	Jump on minus	S=1			
JZ	Jump on zero	Z=1			
JNZ	Jump on no zero	Z=0			
JPE	Jump on parity even	P=1			
JPO	Jump on parity odd	P=0			
Opcode	Description	Flag Status	16-bit address	Unconditional subroutine call	The program sequence is transferred to the memory address given in the operand. Before transferring, the address of the next instruction after CALL is pushed onto the stack.
CC	Call on Carry	CY=1			
CNC	Call on no Carry	CY=0			
CP	Call on positive	S=0			
CM	Call on minus	S=1			
CZ	Call on zero	Z=1			

CNZ	Call on no zero	Z=0			
CPE	Call on parity even	P=1			
CPO	Call on parity odd	P=0			
RET			None	Return from subroutine unconditionally	The program sequence is transferred from the subroutine to the calling program.
Opcode	Description	Flag Status	None	Return from subroutine conditionally	The program sequence is transferred from the subroutine to the calling program based on the specified flag of the PSW and the program execution begins at the new address.
RC	Return on Carry	CY=1			
RNC	Return on no Carry	CY=0			
RP	Return on positive	S=0			
RM	Return on minus	S=1			
RZ	Return on zero	Z=1			
RNZ	Return on no zero	Z=0			
RPE	Return on parity even	P=1			

RPO	Return on parity odd	P=0																					
PCHL			None	Load the program counter with HL contents	The contents of registers H & L are copied into the program counter. The contents of H are placed as the high-order byte and the contents of L as the loworder byte.																		
RST			0-7	Restart	<div><p>The RST instruction is used as software instructions in a program to transfer the program execution to one of the following eight locations.</p><table><tr><th>Instruction</th><th>Restart Address</th></tr><tr><td>RST 0</td><td>0000H</td></tr><tr><td>RST 1</td><td>0008H</td></tr><tr><td>RST 2</td><td>0010H</td></tr><tr><td>RST 3</td><td>0018H</td></tr><tr><td>RST 4</td><td>0020H</td></tr><tr><td>RST 5</td><td>0028H</td></tr><tr><td>RST 6</td><td>0030H</td></tr><tr><td>RST 7</td><td>0038H</td></tr></table><p>The 8085 has additionally 4 interrupts, which can generate RST instructions internally and doesn't require any external hardware. Following are those instructions and their Restart addresses –</p></div>	Instruction	Restart Address	RST 0	0000H	RST 1	0008H	RST 2	0010H	RST 3	0018H	RST 4	0020H	RST 5	0028H	RST 6	0030H	RST 7	0038H
Instruction	Restart Address																						
RST 0	0000H																						
RST 1	0008H																						
RST 2	0010H																						
RST 3	0018H																						
RST 4	0020H																						
RST 5	0028H																						
RST 6	0030H																						
RST 7	0038H																						

			Interrupt	Restart Address
			TRAP	0024H
			RST 5.5	002CH
			RST 6.5	0034H
			RST 7.5	003CH

Following is the table showing the list of Control instructions with their meanings.

Opcode	Operand	Meaning	Explanation
NOP	None	No operation	No operation is performed, i.e., the instruction is fetched and decoded.
HLT	None	Halt and enter wait state	The CPU finishes executing the current instruction and stops further execution. An interrupt or reset is necessary to exit from the halt state.
DI	None	Disable interrupts	The interrupt enable flip-flop is reset and all the interrupts are disabled except TRAP.
EI	None	Enable interrupts	The interrupt enable flip-flop is set and all the interrupts are enabled.
RIM	None	Read interrupt mask	This instruction is used to read the status of interrupts 7.5, 6.5, 5.5 and read serial data input bit.
SIM	None	Set interrupt mask	This instruction is used to implement the interrupts 7.5, 6.5, 5.5, and serial data output.
