- **What is a microprocessor?**

Microprocessor is a controlling unit of a micro-computer, fabricated on a small chip capable of performing ALU (Arithmetic Logical Unit) operations and communicating with the other devices connected to it. Microprocessor consists of an ALU, register array, and a control unit. ALU performs arithmetical and logical operations on the data received from the memory or an input device. Register array consists of registers identified by letters like B, C, D, E, H, L and accumulator. The control unit controls the flow of data and instructions within the computer.

- **What is a bus?Explain different types of buses in 8085.**

Bus is a group of conducting wires which carries information, all the peripherals are connected to microprocessor through Bus.There are three types of buses Address bus,data bus and control bus.

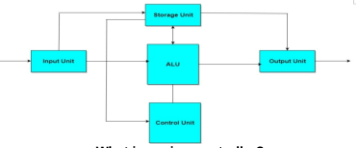
Address bus: It is a group of conducting wires which carries address only.Address bus is unidirectional because data flow in one direction, from microprocessor to memory or from microprocessor to Input/output devices (That is, Out of Microprocessor). Length of Address Bus of 8085 microprocessor is 16 Bit (That is, Four Hexadecimal Digits), ranging from 0000 H to FFFF H, (H denotes Hexadecimal).

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).

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.

- **Explain the architecture of computer System?**

Computer is an electronic machine that makes performing any task very easy. In computer, the CPU executes each instruction provided to it, in a series of steps, this series of steps is called Machine Cycle, and is repeated for each instruction. One machine cycle involves fetching of instruction, decoding the instruction, transferring the data, executing the instruction.



- **What is a microcontroller?**

A microcontroller is a small and low-cost microcomputer, which is designed to perform the specific tasks of embedded systems like displaying microwave's information, receiving remote signals, etc. The general microcontroller consists of the processor, the memory (RAM,ROM, EPROM), Serial ports, peripherals (timers, counters), etc.

- **Microcontroller & Microprocessor- Differentiate**

Microcontroller: Microcontrollers are used to execute a single task within an application * Its designing and hardware cost is low * Easy to replace * It is built with CMOS technology, which requires less power to operate.* It consists of CPU, RAM, ROM, I/O ports.
Microprocessor : Microprocessors are used for big applications. * Its designing and hardware cost is high. * Not so easy to replace. * Its power consumption is high because it has to control the entire system. * It doesn't consist of RAM, ROM, I/O ports. It uses its pins to interface to peripheral devices.

- **Memory classification of computer systems**

A memory is just like a human brain. It is used to store data and instructions. Computer Memory is the storage space in the computer, where data is to be processed and instructions required for processing are stored. The memory is divided into large number of small parts called cells. Each location or cell has a unique address, which varies from zero to memory size minus one. For example, if the computer has 64k words, then this memory unit has 64 * 1024 = 65536 memory locations. The address of these locations varies from 0 to 65535.
Cache Memory: Cache memory is a very high speed semiconductor memory which can speed up the CPU. It acts as a buffer between the CPU and the main memory. It is used to hold those parts of data and program which are most frequently used by the CPU. The parts of data and programs are transferred from the disk to cache memory by the operating system, from where the CPU can access them.
Primary Memory: Primary memory holds only those data and instructions on which the computer is currently working. It has a limited capacity and data is lost when power is switched off. It is generally made up of semiconductor devices. These memories are not as fast as registers. The data and instruction required to be processed resides in the main memory. It is divided into two subcategories RAM and ROM.
Secondary Memory: This type of memory is also known as external memory or non-volatile. It is slower than the main memory. These are used for storing data/information permanently. CPU directly does not access these memories, instead they are accessed via input-output routines. The contents of secondary memories are first transferred to the main memory, and then the CPU can access it. For example, disk, CD-ROM, DVD, etc.

- **RAM Computer Memory**

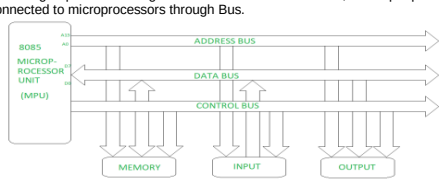
RAM memory is very fast, it can be written to as well as read, it is volatile (so all data stored in RAM memory is lost when it loses power) and, finally, it is very expensive compared to all types of secondary memory in terms of cost per gigabyte. It is because of the relative high cost of RAM compared to secondary memory types that most computer systems use both primary and secondary memory.
DRAM: DRAM stands for Dynamic RAM, and it is the most common type of RAM used in computers. The oldest type is known as single data rate (SDR) DRAM, but newer computers use faster dual data rate (DDR) DRAM. DDR comes in several versions including DDR2 , DDR3, and DDR4, which offer better performance and are more energy efficient than DDR. However different versions are incompatible, so it is not possible to mix DDR2 with DDR3 DRAM in a computer system. DRAM consists of a transistor and a capacitor in each cell.
SRAM: SRAM stands for Static RAM, and it is a particular type of RAM which is faster than DRAM, but more expensive and bulker, having six transistors in each cell. For those reasons SRAM is generally only used as a data cache within a CPU itself or as RAM in very high-end server systems. A small SRAM cache of the most imminently -needed data can result in significant speed improvements in a system

- **ROM Computer Memory**

ROM stands for read-only memory, and the name stems from the fact that while data can be read from this type of computer memory, data cannot normally be written to it. It is a very fast type of computer memory which is usually installed close to the CPU on the motherboard. ROM is a type of non-volatile memory, which means that the data stored in ROM persists in the memory even when it receives no power – for example when the computer is turned off. In that sense it is similar to secondary memory, which is used for long term storage.
PROM: It stands for Programmable Read-Only Memory, and it is different from true ROM in that while a ROM is programmed (i.e. has data written to it) during the manufacturing process, a PROM is manufactured in an empty state and then programmed later using a PROM programmer or burner.
EPROM: It stands for Erasable Programmable Read-Only Memory, and as the name suggests, data stored in an EPROM can be erased and the EPROM reprogrammed. Erasing an EPROM involves removing it from the computer and exposing it to ultraviolet light before re-burning it.
EEPROM: It stands for Electrically Erasable Programmable Read-Only Memory, and the distinction between EPROM and EEPROM is that the latter can be erased and written to by the computer system it is installed in. In that senseEEPROM is not strictly read-only. However in many cases the write process is slow, so it is normally only done to update program code such as firmware or BIOS code on an occasional basis

- **8085 microprocessor bus organization architecture**

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

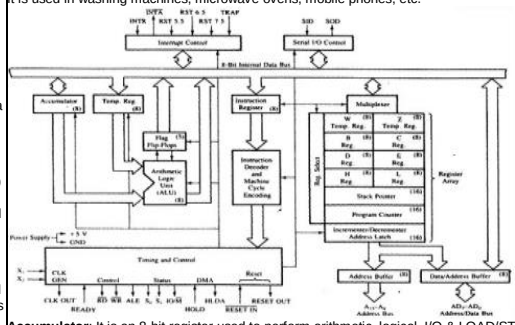


(Adrs bus, Data bus, Control buss definitions needed)

- **8085 microprocessor Architecture with a neat diagram.**

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
It is used in washing machines, microwave ovens, mobile phones, etc.



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.
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.
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.

- **Pin configuration of the 8085 microprocessor**

Properties: Single + 5V Supply, 4 Vectored Interrupts (One is Non Maskable) Serial In/Serial Out Port. Decimal, Binary, and Double Precision Arithmetic. Direct Addressing Capability to 64K bytes of memory. The Intel 8085A is a new generation, complete 8 bit parallel central processing unit (CPU). The 8085A uses a multiplexed data bus. The address is split between the 8 bit address bus and the 8 bit data bus. Figures are at the end of the document.
A6 - A1s (Output 3 State) Address Bus; The most significant 8 bits of the memory address or the 8 bits of the I/0 address,3 stated during Hold and Halt modes. AD0 - 7 (Input/Output 3state) Multiplexed Address/Data Bus; Lower 8 bits of the memory address (or I/0 address) appear on the bus during the first clock cycle of a machine state. It then becomes the data bus during the second and third clock cycles. 3 stated during Hold and Halt modes.
ALE (Output) Address Latch Enable: It occurs during the first clock cycle of a machine state and enables the address to get latched into the on chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. ALE can also be used to strobe the status information.ALE is never 3stated.

- **Use of HOLD pin of the microprocessor**

It supports Direct Memory Access (DMA). When an interfacing device needs to access the microprocessor, DMA controller places a high input on HOLD line. Microprocessor then relinquishes control of the bus and acknowledges the receipt of the request to the DMA controller. When the DMA operation is over, HOLD line will be brought down by the DMA controller and in turn, CPU will exit from the hold state.

- **Use of ALE pin of 8085?**

A positive going pulse on the ALE line indicates that the bits on AD7 — AD0
Are address bits. This signal is utilized to handle the low order address from the multiplexed bus and make a separate set of eight address lines.

- **Use of READY pin of 8085?**

Microprocessor waits until the signal on this line is high to access data from a peripheral device. This is used to delay the microprocessor Read or Write cycles until a slow peripheral device is ready to access the data.

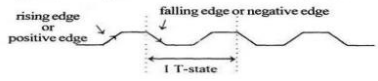
- **Timing diagram**

Timing Diagram is a graphical representation. It represents the execution time taken by each instruction in a graphical format. The execution time is represented in T-states.**Instruction Cycle:**The time required to execute an instruction is called instruction cycle.**Machine Cycle:**The time required to access the memory or input/output devices is called machine cycle.**T-State:**The machine cycle and instruction cycle takes multiple clock periods.A portion of an operation carried out in one system clock period is called as T-state

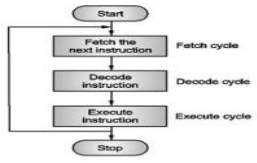
- **T-state**

The machine cycle and instruction cycle takes multiple clock periods. A portion of an operation carried out in one system clock period is called as T-state.

Time period, T = 1/f ; where f = Internal clock frequency



- **Instruction cycle in 8085**

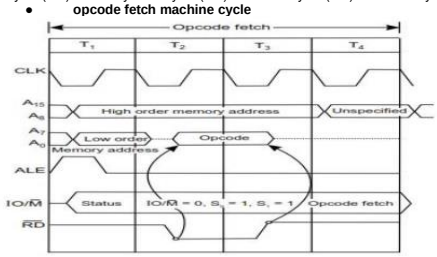


The time required to fetch an instruction and necessary data f rom memory and to execute it, is called an instruction cycle. Or the total time required to execute an instruction is given by: IC = FC + EC (IC = Instruction Cycle, FC = Fetch Cycle, EC = Execute Cycle)
Decode the instruction (Decode Cycle): The opcode fetched f rom the memory goes to the data register, DR and then to instruction register, IR. From the IR it goes to the decoder circuitry which decodes the instruction. Decoder circuitry is within the microprocessor.
Execute the Instruction (Execute Cycle): After the instruction is decoded, execution begins.
If the operand resides in the general purpose registers, execution is immediately performed. The time taken in decoding and execution of an instruction is one clock cycle. In some situations, an execution cycle may involve one or more read or write cycles or both.

- **Machine cycles of 8085 microprocessors**

The time needed for completing one operation of accessing memory, I/O or acknowledging an external request is termed as Machine cycle. It is composed of T-states. One subdivision of the operation completed in one clock period is termed as T-state. The 8085 microprocessor has 5 basic machine cycles. They are ○ Opcode fetch cycle (4T)○ Memory read cycle (3 T) ○Memory write cycle (3 T) ○I/O read cycle (3 T) ○I/O write cycle (3 T)

- **opcode fetch machine cycle**



●Each instruction of the processor has one byte opcode. ●The opcodes are stored in memory. So, the processor executes the opcode fetch machine cycle to fetch the opcode from memory. ●Hence, every instruction starts with the opcode fetch machine Cycle. ●The time taken by the processor to execute the opcode fetch cycle is 4T.● In this time, the first 3 T-states are used for fetching the opcode from memory and the remaining T-states are used for internal operations by the processor.

S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/$\overline{M}$ = 0, S1 =1 and S0 = 1. These status signals do not change throughout the OF machine cycle.
4	T_2	The microprocessor makes the RD' line LOW to enable memory read and increments the Program Counter.
5		The contents on D7 – D0 (i.e. the Opcode) are placed on the address / data bus.
6	T_3	The microprocessor transfers the Opcode on the address / data bus to Instruction Register (IR).
7		The microprocessor makes the RD' line HIGH to disable memory read.
8	T_4	The microprocessor decodes the instruction.

Memory read machine cycle
●The memory read machine cycle is executed by the processor to read a data byte from memory. ●The processor takes 3T states to execute this cycle. The instructions which have more than one byte word size will use the machine cycle after the opcode fetch machine cycle

T state	Operation
T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
	The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
	The status signals are changed as IO/$\overline{M}$ = 0, S1 =1 and S0 = 0. These status signals do not change throughout the memory read machine cycle.
T_2	The microprocessor makes the RD' line LOW to enable memory read and increments the Program Counter.
	The contents on D7 – D0 (i.e. the data) are placed on the address / data bus.
T_3	The data loaded on the address / data bus is moved to the microprocessor.
	The microprocessor makes the RD' line HIGH to disable the memory read operation.

- **Memory write machine**

The memory write machine cycle is executed by the processor to write a data byte in a memory location. ●The processor takes 3T states to execute this machine cycle.

T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
	The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
	The status signals are changed as IO/$\overline{M}$ = 0, S1 =0 and S0 = 1. These status signals do not change throughout the memory write machine cycle.
T_2	The microprocessor makes the WR' line LOW to enable memory write.
	The contents of the specified register are placed on the address / data bus.
T_3	The data placed on the address / data bus is transferred to the specified memory location.
	The microprocessor makes the WR' line HIGH to disable the memory write operation.

- **I/O read machine cycle**

*The I/O Read cycle is executed by the processor to read a data byte from I/O port or from the peripheral, which is I/O,mapped in the system. * The processor takes 3T states to execute this machine cycle. * The IN instruction uses this machine cycle during the execution.

T state	Operation
T_1	The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
	The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
	The status signals are changed as IO/$\overline{M}$ = 0, S1 =1 and S0 = 0. These status signals do not change throughout the I/O read machine cycle.
T_2	The microprocessor makes the RD' line LOW to enable I/O read.
	The contents on D7 – D0 (i.e. the data) are placed on the address / data bus.
T_3	The data loaded on the address / data bus is moved to the microprocessor ie., to the accumulator.
	The microprocessor makes the RD' line HIGH to disable the I/O read operation.

- **Various addressing modes of the 8085 microprocessor**

Immediate Addressing Mode: the source operand is always data. If the data is 8-bit, then the instruction will be of 2 bytes, if the data is of 16-bit then the instruction will be of 3 bytes.
Examples: MVI B 45 (move the data 45H immediately to register B), LXI H3050 (load the H-L pair with the operand 3050H immediately), JMP address (jump to the operand address immediately)
Register Addressing Mode: the data to be operated is available inside the register(s) and register(s) is(are) operands. Therefore the operation is performed within various registers of the microprocessor.
Examples: MOV A, B (move the contents of register B to register A), ADD B (add contents of registers A and B and store the result in register A), INR A (increment the contents of register A by one)
Direct Addressing Mode: the data to be operated is available inside a memory location and that memory location is directly specified as an operand. The operand is directly available in the instruction itself.
Examples: LDA 2050 (load the contents of memory location into accumulator A), LHLD address (load contents of 16-bit memory location into H-L register pair), IN 35 (read the data from port whose address is 35)
Register Indirect Addressing Mode: the data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair.
Examples: MOV A, M (move the contents of thememory location pointed by the H-L pair to the accumulator), LDAX B (move contents of B-C register to the accumulator), LXIH 9570 (load immediate the H-L pair with the address of the location 9570)

- **various instruction formats of 8085**

An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts: one is the task to be performed, called the operation code (opcode), and the second is the data to be operated on, called the operand. The operand (or data) can be specified in various ways. It may include 8-bit (or 16-bit) data, an internal register, a memory location, or 8-bit (or 16-bit) address. In some instructions, the operand is implicit.
One-word or 1-byte instructions: A 1-byte instruction includes the opcode and operand in the same byte. Operand(s) are internal register and are coded into the instruction.
Two-word or 2-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-word or 3-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.

●

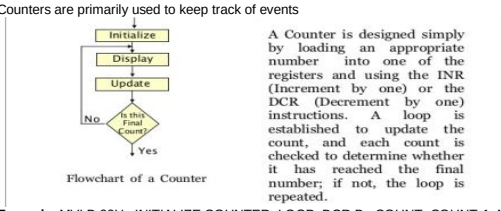
- **Data transfer instructions of the 8085 microprocessor.**

These instructions are used to move data between the registers, or between memory and the registers. These instructions perform a simple task – to copy data from a source to a destination. While copying the contents stored in the source are not altered. They are also referred to as copy instructions.

Sr. No	Data Transfer	Example
1.	Between registers	Register B -> Register D
2.	Specific data byte to register or a memory location	Data Byte -> Register B
3.	Between memory location and register	Memory Location -> Register A
4.	Between the I/O device and the accumulator	Input Device -> Register A
5.	Between a register pair and the stack	Register Pair data -> Stack Location

- **Counter**

Counters are primarily used to keep track of events



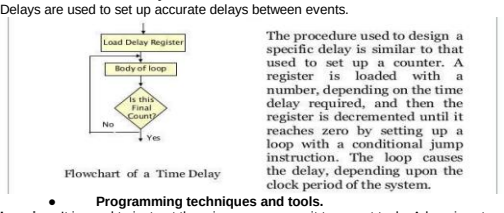
Flowchart of a Counter

A Counter is designed simply by loading an appropriate number into one of the registers and using the INR (Increment by one) or the DCR (Decrement by one) instructions. A loop is established to update the count, and each count is checked to determine whether it has reached the final number; if not, the loop is repeated.

Example: MVI B,00H ; INITIALIZE COUNTER, LOOP: DCR B ; COUNT=COUNT-1, MOV A,B ; MOVE COUNT TO ACC, OUT 01H ; DISPLAY IT AT PORT 01H, JNZ LOOP ; IF COUNT>0 REPEAT

- **Time delay**

Delays are used to set up accurate delays between events.



Flowchart of a Time Delay

The procedure used to design a specific delay is similar to that used to set up a counter. A register is loaded with a number, depending on the time delay required, and then the register is decremented until it reaches zero by setting up a loop with a conditional jump instruction. The loop causes the delay, depending upon the clock period of the system.

- **Programming techniques and tools.**

Looping: It is used to instruct the microprocessor unit to repeat tasks.A loop is set up by instructing MPU to change sequence of execution and perform the task given.This is accomplished by Jump Instructions.
Loops are of 2 types: • Continuous(repeats a task continuously) • Conditional(repeats a task until certain data conditions are met)
Indexing: It means counting or referencing objects with sequential numbers.Data bytes are stored in memory location,and those data bytes are referred to by their memory location.
Counters: This programming technique uses INR or DCR instructions.A loop is established to update count and each count is checked to determine whether it has reached the final number and if not reached,then the loop is repeated.
Time Delay: It is a similar programming technique used to set up a counter. Register is loaded with a number depending on the time delay required and then the register is decremented until it reaches zero.Register works on the principle of time delay within setting up a loop with a conditional jump instruction and the loop causes delay depending upon the lock period of the system. It is achieved by two methods: • Using a register • Using a register pair

- **Subroutine**

• A subroutine is a small program written separately from the main program to perform a particular task that you may repeatedly require in the main program. Essentially, the concept of a subroutine is that it is used to avoid the repetition of smaller programs. • Subroutines are written separately and are stored in a memory location that is different from the main program. You can call a subroutine multiple times from the main program using a simple CALL instruction. And a RET instruction is used at the end of the subroutine to return to the main program.

- **Subroutine related instructions of the 8085 microprocessor.**

The 8085 microprocessor has 4 instructions to implement subroutines. The unconditional and conditional CALL and RET instructions. • In unconditional CALL, when a subroutine is called the content of the PC is stored on the stack and the program execution is transferred to the subroutine address. When the unconditional RET instruction is executed, the memory address stored on the stack is retrieved and the sequence of execution is resumed in the main program. • The conditional CALL and RET instructions are based on the four flag register conditions -S,Z,CY and P. The conditions are tesed by checking the respective flags. In conditional CALL , execution control is transferred to the subroutine if the condition is met. In conditional RET instruction, the control returns to the main program if the condition is met, otherwise the sequence in the subroutine is continued.

- **Stack**

The stack is a reserved area of the memory in RAM where we can store
Temporary information. Interestingly, the stack is a shared resource as it can
Be shared by the microprocessor and the programmer. The programmer
Can use the stack to store data. And the microprocessor uses the stack to execute subroutines. The 8085 has a 16-bit register known as the 'Stack Pointer.' its function is to hold the memory address of the stack. This control is given to the programmer. The programmer can decide the starting address of the stack by loading the address into the stack pointer register at the beginning of a program.

- **Implement a stack using instructions of 8085**

The stack works on the principle of First In Last Out. The memory location of the most recent data entry on the stack is known as the Stack Top. Two operations used to control the movement of data into a stack and from a stack. These two instructions are PUSH and POP. PUSH – This is the instruction to write information on the stack. POP – This is the instruction to read information from the stack. There are two methods to add data to the stack: Direct method and Indirect method. using either LXI or the SPHL instruction.

- **Software interrupts**

Software Interrupt is invoked by the use of INT instruction. This event immediately stops execution of the program and passes execution over to the INT handler. The INT handler is usually a part of the operating system and determines the action to be taken. It occurs when an application program terminates or requests certain services from the operating system. Software interrupts can be classified into two types: 1. Normal Interrupts. 2.Exception
They are – *RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6, RST 7.*

- **Hardware interrupts**

Hardware Interrupt is caused by some hardware device such as a request to start an I/O, a hardware failure or something similar. Hardware interrupts were introduced as a way to avoid wasting the processor's valuable time in polling loops, waiting for external events.For example, when an I/O operation is completed such as reading some data into the computer from a tape drive. Hardware interrupts can be classified into two types: 1. Maskable Interrupt. 2. Non Maskable Interrupt. They are *INTR, RST 7.5, RST 6.5, RST 5.5, TRAP*

- **Various interrupts of 8085**

When microprocessor receives any interrupt signal from peripheral(s) which are requesting its services, it stops its current execution and program control is transferred to a sub-routine by generating CALL signal and after executing sub-routine by generating RET signal again program control is transferred to main program from where it had stopped.
Hardware Interrupts: When microprocessors receive interrupt signals through pins (hardware) of microprocessor, they are known as *Hardware Interrupts*. There are 5 Hardware Interrupts in 8085 microprocessors. They are – *INTR, RST 7.5, RST 6.5, RST 5.5, TRAP*
Software Interrupts: are those which are inserted in between the program which means these are mnemonics of microprocessor. There are 8 software interrupts in the 8085 microprocessor. They are – *RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6, RST 7.*
Vectored Interrupts: *Vectored Interrupts* are those which have fixed vector address (starting address of sub-routine) and after executing these, program control is transferred to that address.
Non-Vectored Interrupts: are those in which the vector address is not predefined. The interrupting device gives the address of the sub-routine for these interrupts. INTR is the only non-vectored interrupt in an 8085 microprocessor.
Maskable Interrupts: are those which can be disabled or ignored by the microprocessor. These interrupts are either edge-triggered or level-triggered, so they can be disabled. INTR, RST 7.5, RST 6.5, RST 5.5 are maskable interrupts in 8085 microprocessor.
Non-Maskable Interrupts: are those which cannot be disabled or ignored by microprocessor. TRAP is a non-maskable interrupt. It consists of both levels as well as edge triggering and is used in critical power failure conditions.

- **SIM and RIM instructions**

Enable Interrupt(EI) – The interrupt enable flip-flop is set and all interrupts are enabled following the execution of the next instruction followed by EI. No flags are affected. After a system reset, the interrupt enable flip-flop is reset, thus disabling the interrupts. This instruction is necessary to enable the interrupts again (except TRAP).
Disable Interrupt (DI) – This instruction is used to reset the value of enable flip-flop hence disabling all the interrupts. No flags are affected by this instruction.
Set Interrupt Mask (SIM) – It is used to implement the hardware interrupts (RST 7.5, RST 6.5, RST 5.5) by setting various bits to form masks or generate output data via the Serial Output Data (SOD) line. First the required value is loaded in accumulator then the SIM will take the bit pattern from it.
Read Interrupt Mask (RIM) – This instruction is used to read the status of the hardware interrupts (RST 7.5, RST 6.5, RST 5.5) by loading into the A register a bytewhich defines the condition of the mask bits for the interrupts. It also reads the condition of SID (Serial Input Data) bit on the microprocessor.

- **Peripheral device**

To communicate with the outside world microcomputers use **peripherals (I/O devices)**.Commonly used peripherals are: A/D converter, D/A converter, CRT, printers, Hard disks, floppy disks, magnetic tapes etc. Peripherals are connected to the microcomputer through electronic circuits known as interfacing circuits.

- **Features of 8255A**

The prominent features of 8255A are as follows − ●It consists of 3 8-bit IO ports i.e. PA, PB, and PC. ●Address/data bus must be externally demux'd. ●It is TTL compatible. ●It has improved DC driving capability.

- **PPI or 8255A**

A programmable peripheral interface(PPI) is a multiport device. The ports may be programmed in a variety of ways as required by the programmer. The device is very useful for interfacing peripheral devices. The term PIA, Peripheral Interface Adapter is also used by some manufacturers.
The Intel 8255 is a programmable peripheral interface (PPI). It has two versions, namely the Intel 8255A and Intel 8255A-5. General descriptions for both are the same. There are some differences in their electrical characteristics. Its main functions are to interface peripheral devices to the microcomputer.
8255A: The 8255A is a general purpose programmable I/O device designed to transfer the data from I/O to interrupt I/O under certain conditions as required. It can be used with almost any microprocessor.It consists of three 8-bit bidirectional I/O ports (24I/O lines) which can be configured as per the requirement.

- **Operating Modes**

Mode 0 - Simple Input/output: The 8255 has two 8-bit ports (Port A and Port B) and two 4-bit ports (Port Cupper and Port Clower). Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.
Mode 1-Strobed Input/output: Mode 1 is strobed input/output mode of operation. The Port A and Port B both are designed to operate in this mode of operation. When Port A and Port B are programmed in Mode 1, six pins of Port C are used for their control.
Mode 2 -Bidirectional Port: Mode 2 is strobed bidirectional mode of operation. In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as a handshake for port B.

- **8254 programmable interval timer**

8254 is a device designed to solve the timing control problems in a microprocessor. It has 3 independent counters, each capable of handling clock inputs up to 10 MHz and size of each counter is 16 bit. It operates in +5V regulated power supply and has 24 pin signals. All modes are software programmable. The 8254 is an advanced version of 8253 which did not offer the feature of read back command.

Data Bus Buffer: It is a tri-state, bi-directional, 8-bit buffer, which is used to interface the 8253/54 to the system data bus. It has three basic functions − ●Programming the modes of 8253/54. ●Loading the count registers. ●Reading the count values.

Read/Write Logic: It includes 5 signals, i.e. RD, WR, CS, and the address lines A0 & A1. In the peripheral I/O mode, the RD and WR signals are connected to IOR and IOW, respectively. In the memory mapped I/O mode, these are connected to MEMR and MEMW
Control Word Register: This register is accessed when lines A0 & A1 are at logic 1. It is used to write a command word, which specifies the counter to be used, its mode, and either a read or write operation. Following table shows the result for various control inputs.

- **DMA controller in 8085**

It is a hardware device that allows I/O devices to directlyaccess memory with less participation of the processor. DMA controller needs the same old circuits of an interface to communicate with the CPU and Input/Output devices. DMA Controller temporarily borrows the address bus,data bus and control bus from the microprocessor and transfers the data directly from the external device to memory location and vice versa.

- **8086 microprocessor**

The 8086 Microprocessor is an enhanced version of the 8085 Microprocessor that was designed by Intel in 1976. It is a 16-bit Microprocessor in a 40 pin, Dual Inline Packaged IC. It Has 20 address lines and16 data lines that provides up to 1MB storage. It consists of a powerful instruction set, which provides operations like multiplication and division easily. It supports two modes of operation, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for systems having multiple processors and Minimum mode is suitable for systems having a single processor.

- **Features of the 8086**

●It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing. ●It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing. ●It is available in 3 versions based on the frequency of operation − 8086 → 5MHz 8086-2 → 8MHz (c)8086-1 → 10 MHz ● It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance. ●Fetch stage can prefetch up to 6 bytes of instructions and stores them in the queue. ●Execute stage executes these instructions. ●It has 256 vectored interrupts. ●It consists of 29,000 transistors.

- **Differentiate between 8085 and 8086**

Size − 8085 is an 8-bit microprocessor, whereas 8086 is a 16-bit microprocessor.

Address Bus − 8085 has a 16-bit address bus while 8086 has a 20-bit address bus.

Memory − 8085 can access up to 64Kb, whereas 8086 can access up to 1 Mb of memory.

Instruction − 8085 doesn't have an instruction queue, whereas 8086 has an instruction queue.

Pipelining − 8085 doesn't support a pipelined architecture while 8086 supports a pipelined Architecture.

I/O − 8085 can address 2^8 = 256 I/O's, whereas 8086 can access 2^16 = 65,536 I/O's.

Cost − The cost of 8085 is low whereas that of 8086 is high.

- **Bus Interface Unit (BIU):**

It provides the interface of 8086 to external memory and I/O devices via the System Bus. It performs various machine cycles such as memory read, I/O read etc. to transfer data between memory and I/O devices.
●It generates the 20 bit physical address for memory access. ●It fetches instructions from the memory. ●It transfers data to and from the memory and I/O. ●Maintains the 6 byte prefetch instruction queue(supports pipelining). BIU mainly contains the 4 Segment registers, the Instruction Pointer, a prefetch queue and an Address Generation Circuit.

- **Instruction pointer in 8086**

It is a 16 bit register. It holds the offset of the next instructions in the Code Segment. ●IP is incremented after every instruction byte is fetched. ●IP gets a new value whenever a branch instruction occurs. ●CS is multiplied by 10H to give the 20 bit physical address of the Code Segment. ●Address of the next instruction is calculated as CS x 10H + IP.
Example: CS = 4321H IP = 1000H, then CS x 10H = 43210H + offset = 44210H. This is the address of the instruction.

- **various segment registers used in 8086**

Code Segment register: CS holds the base address for the Code Segment. All programs are stored in the Code Segment and accessed via the IP.
Data Segment register: DS holds the base address for the Data Segment.
Stack Segment register: SS holds the base address for the Stack Segment.
Extra Segment register: ES holds the base address for the Extra Segment.
Address Generation Circuit: ●The BIU has a Physical Address Generation Circuit. ●It generates the 20 bit physical address using Segment and Offset addresses using the formula: ●Physical Address = Segment Address x 10H + Offset Address

Use of the instruction queue

6 Byte Prefetch Queue: ●It is a 6 byte queue (FIFO). ●Fetching the next instruction (by BIU from CS) while executing the current instruction is called pipelining. ●Gets flushed whenever a branch instruction occurs.

- **Execution Unit (EU)**

The main components of the EU are General purpose registers, the ALU, Special purpose registers, Instruction Register and Instruction Decoder and the Flag/Status Register.
1.Fetches instructions from the Queue in BIU, decodes and executes arithmetic and logic operations using the ALU. 2.Sends control signals for internal data transfer operations within the microprocessor. 3.Sends request signals to the BIU to access the external module. 4.It operates with respect to T-states (clock cycles) and not machine cycles.

- **General purpose registers in 8086**

AX register: It holds operands and results during multiplication and division operations. Also an accumulator during String operations.
BX register: It holds the memory address (offset address) in indirect addressing modes.
CX register: It holds count for instructions like loop, rotate, shift and string operations.
DX register: It is used with AX to hold 32 bit values during multiplication and division.

- **Special purpose registers**

Stack Pointer: Points to Stack top. Stack is in Stack Segment, used during instructions like PUSH, POP, CALL, RET etc.
Base Pointer:BP can hold an offset address of any location in the stack segment. It is used to access random locations of the stack.
Source Index: It holds an offset address in the Data Segment during string operations.
Destination Index: It holds offset addresses in Extra Segment during string operations

- **Use of flag registers in 8086**

The Flag register is a Special Purpose Register. Depending upon the value of result after any arithmetic and logical operation the flag bits become set (1) or reset (0).There are total 9 flags in 8086 and the flag register is divided into two types:**Status Flags and Control Flags**

Status Flags – There are 6 flag registers in 8086 microprocessor which become set(1) or reset(0) depending upon condition after either 8-bit or 16-bit operation.
Sign Flag (S) – After any operation if the MSB (B(7)) of the result is 1, it indicates the number is negative and the sign flag becomes set, i.e. 1. If the MSB is 0, it indicates the number is positive and the sign flag becomes reset i.e. 0. from 00H to 7F, sign flag is 0 from 80H to FF, sign flag is 1
Zero Flag (Z) – After any arithmetic or logical operation if the result is 0 (00)H, the zero flag becomes set i.e. 1, otherwise it becomes reset i.e. 0. 00H zero flag is 1. from 01H to FFH zero flag is 0
Auxiliary Carry Flag (AC) – This flag is used in BCD number system(0-9). If after any arithmetic or logical operation D(3) generates any carry and passes on to B(4) this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0. This is the only flag register which is not accessible by the programmer1-carry out from bit 3 on addition or borrow into bit 3 on subtraction 0-otherwise
Parity Flag (P) – If after any arithmetic or logical operation the result has even parity, an even number of 1 bits, the parity register becomes set i.e. 1, otherwise it becomes reset i.e. 0. 1-accumulator has even number of 1 bits 0-accumulator has odd parity
Carry Flag (CY) – Carry is generated when performing n bit operations and the result is more than n bits, then this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0. During subtraction (A-B), if A>B it becomes reset and if (A<B) it becomes set. Carry flag is also called the borrow flag. 1-carry out from MSB bit on addition or borrow into MSB bit on subtraction 0-no carry out or borrow into MSB bit
Overflow Flag (O) – This flag will be set (1) if the result of a signed operation is too large to fit in the number of bits available to represent it, otherwise reset (0). After any operation, if D[6] generates any carry and passes to D[7] OR if D[6] does not generate carry but D[7] generates, the overflow flag becomes set, i.e., 1. If D[6] and D[7] both generate carry or both do not generate any carry, then the overflow flag becomes reset, i.e., 0.

- **Addressing modes of 8086.**

The way of specifying data to be operated by an instruction is known as addressing modes. This specifies that the given data is an immediate data or an address. It also specifies whether the given operand is register or register pair. Types of addressing modes:1. **Register mode –** In this type of addressing mode both the operands are registers. Example:MOV AX, BX ADD AL, BL
2. **Immediate mode –** In this type of addressing mode the source operand is a 8 bit or 16 bit data. Destination operand can never be immediate data. Example:MOV AX, 2000 ADD AL, 45
3. **Displacement or direct mode –** In this type of addressing mode the effective address is directly given in the instruction as displacement. Example:MOV AX, [DISP] MOV AX, [0500]
4. **Register indirect mode –** In this addressing mode the effective address is in SI, DI or BX. Example:MOV AX, [DI] ADD AL, [BX] MOV AX, [SI]
5. **Based indexed mode –** In this effective address is sum of base register and index register. Base register: BX, BP Index register: SI, DI

8086 microprocessor	8088 microprocessor
The data bus is of 16 bits.	The data bus is of 8 bits.
It has 3 available clock speeds (5 MHz, 8 MHz (8086-2) and 10 MHz (8086-1)).	It has 3 available clock speeds (5 MHz, 8 MHz)
The memory capacity is 512 kB.	The memory capacity is implemented as a single 1 MX 8 memory banks.
It has memory control pin (M/IO) signal.	It has complemented memory control pin (IO/M) signal of 8086.
It has Bank High Enable (BHE) signal.	It has Status Signal (SSO).
It can read or write either 8-bit or 16-bit word at the same time.	It can read only 8-bit word at the same time.
Input/Output voltage level is measured at 2.5 mA.	Input/Output voltage level is measured at 2.0 mA
It has 6 byte instruction queue.	It has 4 byte instruction queue as it can fetch only 1 byte at a time.

- **Segmentation in 8086.**

● Segmentation means dividing the memory into logically different parts called segments. 8086 has a 20-bit address bus, hence it can access 1MB memory. It is not possible to work with a 20 bit address as it is not a byte compatible number i.e. (20 bits is two and a half bytes). ●To avoid working with this incompatible number, we create a virtual model of the memory. Here the memory is divided into 4 segments: Code, Stack Data and Extra. The max size of a segment is 64KB and the minimum size is 16 bytes. ●Now programmer can access each location with a VIRTUAL ADDRESS. The Virtual Address is a combination of Segment Address and Offset Address. Segment Address indicates where the segment is located in the memory (base address) and Offset Address gives the offset of the target location within the segment. ●Since both, Segment Address and Offset Address are 16 bits each, they both are compatible numbers and can be easily used by the programmer. Hence, we can access 1 MB memory using only a 16 bit offset address for most part of the program.

- **Physical address is calculated in 8086.**

Physical Address (20 bit) = Segment Address(16 bit) X 10H + Offset Address (16 bit)

Segment	Offset Registers	Function
CS	IP	Address of the next instruction
DS	BX, DI, SI	Address of data
SS	SP, BP	Address in the stack
ES	BX, DI, SI	Address of destination data (for string operations)

The value of Code Segment (CS) Register is 4042H and the value of different offsets is as follows:BX: 2025H , IP: 0580H , DI: 4247H. Calculate the effective address of the memory location pointed by the CS register.
Solution: The offset of the CS Register is the IP register. Therefore, the effective address of the memory location pointed by the CS register is calculated as follows:
Effective address= Base address of CS register X 10H + Address of IP
= 4042H X 10H + 0580H
= (40420 + 0580)H
= 41000H