1. AN INTRODUCTION OF MICROPROCESSOR

Evolution Of Microprocessor

*First generation:
a) Intel’s 4004 was the first microprocessor available in the market. It was a four bit pmos microprocessor introduced in 1971, designed to be used in calculators.
b) In 1972 Intel introduced first general purpose 8-bit microprocessor Intel 8008.
c) It was followed by Intel’s 8080 in 1973 and Motorola’s 6800 in the same year.
For e.g.
Intel’s 4004 (4-bit), 8008 (8-bit)
Motorola’s 6800 (8-bit)

*Second generation:
a) In 1974 Intel’s 8080, Zilog’s Z-80, Motorola’s M6800 were introduced. All these were 8-bit microprocessors.
b) During second generation, the development of microprocessor has been in a direction to complete microprocessor system (microcontroller) i.e. CPU, ROM, RAM, clock, I/O ports all in single package.
c) In 1976, Intel’s 8085, 8-bit microprocessor was introduced.
d) In 1977, 12-bit microprocessor Intel’s IM6100 and Toshiba’s T8190 developed.
For e.g.
INTEL’S 8085 (8-bit)
Zilog’s Z80 (8-bit)

*Third generation:
a) Intel introduced first 16-bit microprocessor 8086 in 1978.
b) It was followed by Zilog’s Z-8000 in 1979 and Motorola’s 68000 in 1980.
c) In third generation, memory space was 64 KB. The other features were full arithmetic execution and efficient higher level language addressing.
For e.g.
Intel’s 8086 (16-bit)
Zilog’s Z-8000 (16-bit)

*Fourth generation:
a) In 1981, Intel introduced first 32-bit microprocessor 80386; it can address physical memory of 4 GB.
b) Other 32-bit microprocessor Hewlett Packard’s HP-32.
c) In 1987, Motorola’s 68020, a 32-bit microprocessor was introduced.
For e.g.
INTEL 80386 (32-bit)
INTEL 80486 (32-bit)

*Fifth generation:
a) Intel made improvement in microprocessor design to provide greatest speed.
b) Also it can run on new OS like UNIX, LINUX.
c) The processor in this generation is called as Pentium. It is 61-bit microprocessor.
For e.g.
Intel’s 80586, Intel Pentium IV.
8085 MICROPROCESSOR

INTRODUCTION

The microprocessor communicates and operates in binary numbers i.e. 0 and 1 which are called as bits. Bit is an abbreviation of binary digit. These digits are represented in terms of electrical voltages. Generally 0 represents low voltage and 1 represents high voltage. A group of bits are called word. And the microprocessors are classified according to the word length i.e. for e.g. a microprocessor with 8-bit word length is known as 8-bit microprocessor. Similarly, a microprocessor with 32-bit word length is known as 32-bit microprocessor. Each microprocessor has a fix set of instructions known as machine language but it is difficult for us to communicate in language of 0’s and 1’s. Therefore, the binary instructions are given abbreviated names are called as mnemonics which forms the assembly language.

DEFINATION: -

Microprocessor is a multipurpose programmable logical device which takes the data (binary) from the storage device called as memory and process the data in the processor and store it again back into the memory location or in the output device. Thus, a microprocessor has been divided into three components:

   I. Microprocessor
   II. Memory device and
   III. Input / output device.

![Microcomputer System Diagram]

Functions Of Microprocessor

* To fetch, decode and execute instruction.
* To transfer data from one block to another or from one block to I/O lines.
* To give proper response to different externally produced interrupt according to their priority.
* To provide control and timing signal to the whole system according to the instruction.

* Features of 8085 microprocessor

1) 8085 is an 8-bit microprocessor which means it has 8-bit data bus
2) 8085 is available in 40 pin DIP package.
3) It has 16-bit address bus which means it can address a memory of 64 kilo bytes.
4) To select external memory or I/O devices 8085 microprocessor uses I/O mapped I/O system.
5) To communicate with external devices 8085 microprocessor uses interrupt method (hardware interrupt)
6) The microprocessor requires +5V power supply can operate with 3 MHz single phase clock

* Function of ALU with block diagram

1) The arithmetic and logic unit is 8-bit unit.
2) It performs arithmetic, logic and rotate operations.
3) It consists of binary adder to perform addition and subtraction by 2’s compliment method.
4) The result is typically stored in accumulator.
5) Accumulator, temporary register and flag register is closely associated with ALU
6) The temporary register is used to hold the data during an arithmetic or logical operations.
7) The flags are set and reset according to the result of the operations in status register.

ORGANISATION OF MICROCOMPUTER SYSTEM:

Since microprocessor is a programmable logical device, it has the capability to make decisions and perform various computing functions. This functions are partly divided into three:
1. A.L.U.
2. Register Array
3. Control Unit.

(1) A.L.U. [Arithmetic Logical Unit]: All the arithmetic operations are such as addition and subtraction is carried out in this unit. Similarly, all the logical operations such as OR, EX-OR, etc. are also done this unit.

(2) Register Array: This area of microprocessor consist of various registers. These registers are primarily used to store data temporarily during the execution of the program. Some of the registers are accessible to the user through instructions.

(3) Control Unit: The control unit provides the necessary timing and control signals to all the operations in the microprocessor. It controls the flow of data between microprocessor, memory and peripheral.

REGISTER ARRAY

The 8085 microprocessor has eight 8-bits addressable registers and two 16-bits registers. The eight 8-bit addressable registers are A, B, C, D, E, H, L and F. Among this registers they are categorized as follows:

1) Special Purpose Registers. (A, F)
2) General Purpose Registers. (B, C, D, E, H, L)
3) Two 16-bits registers.
1) **Special Purpose Registers**: There are two special purpose registers

   i. Accumulator (A)
   ii. Flags (F)

**Accumulator**: The accumulator is the gate of microprocessor. It is an 8-bit addressable register. All the process of addition, subtraction as well as all the logical operations (for eg EX-OR, AND, OR) are done in accumulator and the result is stored in the accumulator itself.

   ii.) **Flags (F)**: Flag register has 5 flip-flop conditions which are set and reset according to data contents of accumulator and registers.

<table>
<thead>
<tr>
<th></th>
<th>Z</th>
<th>X</th>
<th>AC</th>
<th>X</th>
<th>P</th>
<th>X</th>
<th>CY</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   x- Don’t care condition.

**S- Sign**: Whenever the answer or the result in the accumulator is negative, sign flag will be set or else if it is positive, sign flag will be reset.

**Z-Zero**: Whenever the answer in the accumulator is zero, zero flag will be set or else it will be reset.

**AC –Auxillary carry**: Whenever there is an addition of 8-bit and the carry is generated due to the addition of the third bit and carry forward to the fourth bit, auxillary carry will be set or else it will be reset.  E.g.

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>+</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**P – Parity**: If the data contents of the accumulator has even number of one’s, parity flag will be set i.e. it is an even parity and whenever the data contents of the accumulator has odd numbers of one’s a parity flag will be reset i.e. it is an odd parity. E.g

   If the data contents of the accumulator is [0A]
   i.e.00001010
   That means parity flag is set. (two ones)

   If the data contents of the accumulator is [07]
   i.e.00000111
   That means parity flag will be reset. (three ones)

**CY – Carry**: Whenever there is an addition of 8-bit and an overflow bit is generated due to the addition of MSB , that bit is stored in carry flag.

   e.g.

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>+</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
**General purpose register:**
The 8085 microprocessor has six general purpose registers to perform the first function listed below:

- a) Store 8-bit data
- b) Perform arithmetic and logical operations
- c) Test for condition
- d) Sequence the execution of instruction
- e) Store the data temporarily during execution of the program

These registers are identified as B, C, D, E, H, and L as shown in the figure below.

They are combined as register pairs – BC, DE, and HL—to perform some 16-bit operation.

<table>
<thead>
<tr>
<th>Accumulator</th>
<th>Flag register</th>
</tr>
</thead>
<tbody>
<tr>
<td>B (8)</td>
<td>C (8)</td>
</tr>
<tr>
<td>D (8)</td>
<td>E (8)</td>
</tr>
<tr>
<td>H (8)</td>
<td>L (8)</td>
</tr>
</tbody>
</table>

- Stack Pointer (SP) (16)
- Program Counter (PC) (16)

**THE 8085 PROGRAMMABLE REGISTER**

These registers are programmable, meaning that a programmer can use them to load or transfer data from the register by using instructions. For example, the instruction MOV B, C transfers the data from the register C to register B.

**Control Unit**: The control unit provides the necessary timing and control signals to all the operations in the microprocessor. It controls the flow of data between the microprocessor, memory, and peripherals.
BUS ARCHITECTURE

System Bus: System bus is a communication path between microprocessor and peripherals, system bus is nothing but a group of wires to carry bits. All the peripherals and the memory share the same bus. However, the microprocessor communicates with only one peripheral at a time. The control unit of microprocessor provides the timings.

The 8085 microprocessor performs different functions using three set of communication lines are called as buses. They form one group called as system bus.

1) Address Bus
2) Data Bus
3) Control Bus

BUS STRUCTURE

Address Bus: Address bus is a group of 16 lines generally identified as shown in the figure above. Address bus is unidirectional i.e. the bit flows only in one direction from microprocessor to peripheral. The microprocessor uses address bus to perform the first function i.e. identifying the peripheral or a memory location. In a computer system, each peripheral or memory location is identified with a binary number called as an address and the address bus is used to carry 16-bit addresses. The microprocessor with its 16 address lines is capable of addressing $2^{16} = 65536 = 64$ memory location. Hence, it is theoretically possible for 8085 microprocessor to address memory location from $0000_{16}$ to $FFFF_{16}$.

Data Bus: Data bus is a group of 8 lines which are used for the flow of data. Since there are 8 data lines, it is possible to have $2^8 = 256$ different values of data ranging from $00_{16}$ to $FF_{16}$.

Data bus is bi-directional i.e. the data flows in both the directions between microprocessor and the peripherals. The microprocessor uses the data bus to transfer the data. The data bus determines the word length and the register size of microprocessor. Thus, 8085 microprocessor is called an 8-bit microprocessor. This means that the bus can be used as address bus as well as data bus. When the ALE signal is high it transmits the lower 8-bit of address bus. When the ALE signal is low it
passes the data. It uses the lines from AD to AD7. The advantage of multiplexing is reduced the number of pins on the microprocessor.

**Control Bus**: It includes the various signal lines that carries signals. It consist of two main groups of control signals namely RD and WR and status signal namely IO/M and one special signal called as ALE to identify the nature of operation.

1. **ALE (Address Latch Enable)**: This signal is used to latch the lower order address form the multiplexed bus. When the ALE signal is high, it transmit the lower 8-bits of the address bus when the ALE signal is low it transmits the data.

2. **RD (Read)**: This is a read control signal. The bar indicates that it is active low. This signal indicates the selected input, output or memory device is to read and the data available on the data bus.

3. **WR (Write)**: This is a write control signal. the bar indicates that it is active low. This signal indicates that the data on the data bus is to be written on the selected memory or input, output operation.

4. **IO/M**: This is status signal. It is used to differentiate between input, output and memory operations. When it is high, it indicates input output operation and when it is low it indicates the memory operation.

5. **S1 and S0**: These are also status signals which are similar to IO/M used to identify various operation.

**Addressing Mode**:

The type of source and destination address utilized by an instruction is determined by an instruction addressing mode. The 8085 microprocessor has 5 addressing modes.

1. **Implied Addressing Mode**
2. **Register Addressing Mode**
3. **Immediate Addressing Mode**
4. **Direct Addressing Mode**
5. **Register Indirect Addressing Mode**.

1. **Implied Addressing Mode**: The addressing mode of 8085 microprocessor instruction is implemented by the instruction function. e.g.
   - i) **STC (Set Carry Flag)**: This instruction deals with carry flag only with no other registers or memory locations.
   - ii) **CMA (Compliment Accumulator)**: This instruction deals only with the accumulator and with no other registers.

2. **Register Addressing Mode**: This mode is used to move the data between the internal registers and hence the instruction source and destination address specifies which of these registers are involved in the transfer. E.g.

   i) **MOV B, A**: Where B is the destination address and A is the source address. In the above instruction, the data of registers B and the contents of register A is remains unchanged.

3. **Immediate Addressing Mode**: In this type of mode the source address does not specify any register or memory location but it specifies the actual source data which is present within the instruction. Therefore, It is immediately available. E.g. **MVI A, FE** – Where A is the destination and FE is the source data and MVI is the operation. In such type of instruction, the value of the source data is transferred to register A.
4. **Direct Addressing Mode**: Using this addressing mode, an operand may be either from or returned to a memory location. The address is specified in the instruction itself. Since the memory address are 16-bit i.e. the address requires 2 bytes and for this reason it is also called as extended addressing mode.

E.g.

LDA C050 :- In this type of instruction, register A is loaded with the data contents of memory location. Similarly, the data contents of the accumulator can be stored to memory location by using the instruction STA C070.

5. **Register Indirect Addressing Mode**: Using register indirect addressing mode, the operand is either read from or returned into the memory location. The address of which is currently stored in register pair HL. The instruction does not contain the actual address but implies the address to be used currently in the HL pair. The actual memory address is therefore obtained indirectly.

E.g. MOVA, M- using this instruction, the accumulator is loaded with the data contents of memory location whose address is specified by register pair HL.

MOV M, B – Using this instruction, the data contents of register B is transferred to memory location. Whose address is specified by HL pair
Two 16- bits registers

a) Stack Pointer  
b) Program Counter

**STACK POINTER**

The stack in an 8085 micro-computer system can be described as set of memory location in R/W memory, specified by a programmer in a main program. These memory location are used to store binary information temporarily during the execution of program.

The beginning of the stack is defined in the program by using the instruction LXI Sp, which loads a 16-bit memory address in the stack pointer of the microprocessor. Once the stack location is defined, storing of data bytes begins at the memory address that is one less than the address in the S.P. register for e.g. if the stack pointer register is loaded with memory address 2099 h (LXI SP ,2099 h), the storing of data bytes begins at 2098 h and continuous in the reverse numerical order (decreasing memory) address such as 2090,2097 H, etc.). Therefore, as a general practice, the stack is initialized at the highest available memory location to prevent the program being destroyed by the stack information.

Data byte in the register pairs of the microprocessor can be stored on the stack (2 data byte at a time) reverse the order by using the instruction PUSH. Data byte can be transferred from stack to respective registers by using the instruction POP. The stack pointer register tracks the storage and retrieval of the information. Since 2 data bytes are being stored at a time, the 16 bit memory address in the stack pointer register is decremented by two, and when the data byte is retrieved, the address is incremented by 2.

**PROGRAM COUNTER**

This is 16-bit register deals with sequencing the execution of instruction. This register is a memory pointer. The memory location have 16-bit address and that is why it is a 16–bit register.

The microprocessor uses this register to sequence the execution of instruction. The function of the P.C. is to point to memory address from which the next byte is to be fetched the P.C. is incremented by one to point to the next memory location.
* Functional pin diagram of microprocessor 8085

HOLD:
1) It indicates that a peripheral such as DMA (Direct Memory Access) controller is requesting the use of address and data buses.

2) Having received a HOLD request the microprocessor release the use of the buses as soon as the current machine cycle is completed. Internal processing may continue.

3) The processor regain the bus after the removal of the HOLD signal.

HLDA:
1) It is a signal for HOLD ACKNOWLEDGEMENT.

2) A HLDA output indicates to a peripheral that a HOLD request has been received and that the microprocessor will relinquish control of buses in the next clock cycle.

3) After the removal of HOLD request HLDA goes low.
**INTR:**
1) INTR is a level triggered maskable interrupt Request input signal.
2) This is a general purpose interrupt with lowest priority.
3) When interrupt signal is given on this line, the microprocessor execute interrupt acknowledge cycle to read interrupt information from interrupting device.
4) The INTR is enable or disable by software.
5) When this arise, program counter does not increment its content.

**RESET IN:**
1) When the signal on this pin goes low, the program counter is set to 0000H.
2) The buses are tri-stated and the microprocessor unit is held in reset condition as long as RESET is applied.
3) It also reset interrupt enable and HLDA flip flop.

**READY:**
1) It is input signal used by microprocessor to sense whether a peripheral is ready to transfer data or not.
2) This signal is used to delay the microprocessor until a slow responding peripheral is ready to send or accept data.
3) If READY is high the peripheral is ready. If it is low, the microprocessor waits for an integral number of clock cycle until it goes high.
4) It is used to synchronize slower peripheral to faster microprocessor.

**IO / M:**
1) It is a status signal indicates whether the address bus is for I/O device or memory.
2) When it goes high, the address on the address bus referring I/O device and when it goes low, the address on the address bus referring memory.
3) It is tri-stated during HOLD and HALT.

**ALE:**
1) ALE means Address Latch Enable. It’s a special output signal generated by microprocessor to indicate the beginning of an operation.
2) It is a positive going pulse generated during first clock cycle of machine state and it indicate that the bits on AD_7 to AD_0.
3) This signal enables the lower 8-bit of the address from the multiplexed bus to latch in to external flip flop or peripheral devices. After AD_0 to AD_7 changes over to data bus.
4) ALE is never tri-stated.

**RD:**
1) This is read control signal. This is active low signal.
2) This signal indicates that selected I/O or memory device is to be read and data is available on data bus.
3) It is tri-stated during HOLD and HALT.
WR:
1) This is write control signal. This is active low signal.
2) This signal indicates that the data on data bus are to written in to selected memory or I/O location.
3) It is tri-stated during HOLD and HALT.

S₀, S₁:
1) These are status signals sent by microprocessor to distinguish the various operations or type of machine cycle in progress.
2) Status code for 8085 microprocessor is:

<table>
<thead>
<tr>
<th>S₁</th>
<th>S₀</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>HALT</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>WRITE</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>READ</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>FETCH</td>
</tr>
</tbody>
</table>

Reset out:
1) It indicates that the microprocessor is being reset.
2) It is connected to peripheral to reset them when microprocessor is reset.

SOD:
1) It means Serial Output Data. It is a data line for serial output.
2) The 7th bit of the accumulator is outputted on SOD line when SIM instruction is executed.
3) The SOD line is eliminated the need for an output port in the software–controlled serial I/O.

TRAP:
1) This signal is used to interrupt the microprocessor.
2) It has highest priority among all the interrupts.
3) It is a non-maskable interrupt. It is unaffected by any interrupt enable or mask.
4) When an interrupt is recognized the next instruction is executed from a fixed location in the memory i.e. 0024H.

RST 7.5:
1) RESART INTERRUPT: This signal is used to interrupt the microprocessor.
2) It is an vectored interrupt
3) When an interrupt is recognized the next instruction is executed from a fixed location in the memory i.e. 7.5 * 8 = 003CH.
4) It is a maskable interrupt.
5) They cause an internal restart to be automatically inserts.
**RST 6.5:**
1) **RESART INTERRUPT**: This signal is used to interrupt the microprocessor.
2) It is an vectored interrupt
3) When an interrupt is recognized the next instruction is executed from a fixed location in the memory i.e. $6.5 \times 8 = 0034H$.
4) It is a maskable interrupt.
5) They cause an internal restart to be automatically inserts.

**RST 5.5:**
1) **RESART INTERRUPT**: This signal is used to interrupt the microprocessor.
2) It is an vectored interrupt
3) When an interrupt is recognized the next instruction is executed from a fixed location in the memory i.e. $5.5 \times 8 = 002CH$.
4) It is a maskable interrupt.
5) They cause an internal restart to be automatically inserts.

* **Multiplexed address / data bus in 8085 microprocessor**:

1) Microprocessor 8085 has 8-bit data bus and 16-bit address bus.
2) The least significant 8-bit of address bus are passed on the same eight lines as that of data bus i.e. on the signal lines $AD_7 - AD_0$
3) These signal lines are bi-directional
4) They are used for dual purpose for lower order 8-bit address and as well as 8-bit of data. This is known as multiplexing and such bus is known as multiplexed bus.
5) To multiplexed means, first to select one and then other.
6) In executing an instruction, during earlier part of cycle these lines are used as the lower order address bus. During later part of cycle, these lines are used as data bus.
7) The 8085 has a special signal called ALE (Address Latch Enable) for informing the peripheral when the address / data bus is sending address and when it is functioning as a data bus.
8) If signal of pin ALE is high (i.e. 1), then the bits on $AD_7 - AD_0$ are address bits also they are data bits.

**Address and data bus structure of 8085 and how address and data is demultiplexed.**

1. With the added functions of the 8085, one 40 pin DIP did not have enough pins for all the inputs and outputs.

2. For that reasons the manufacturer uses pins 12 to 19 as **dual-purpose address/data bus lines** ($AD_0 - AD_7$). The unit is said to have an 8-bit multiplexed address/data bus.

3. The least significant 8 address lines share pins with the 8 data bus lines.
4. To multiplexed means to first select one and then another. Therefore to multiplexed address/data bus means to first use of the bus is send the address and next to send or receive data via same bus.

5. The 8085 have special signal for informing the peripherals when the address/data bus is sending on address and when the bus is functioning as data bus. The special signal is called the **Address Latch Enable (ALE) control signal**.

6. During the first cycle, the address is sent out. The lower 8-bits are latched into the peripherals by the **Address Latch Enable**. During the rest of the machine cycle the **data bus** is used for memory or I/O data.

I/O mapped I/O and memory mapped I/O schemes.

**OR**

**Note on addressing I/O devices.**

**Ans.** : Microprocessor is connected to various other devices such as memory and I/O devices. There are two scheme by which these devices can be addressed :

1) **Memory mapped I/O scheme** :
   1. In memory mapped I/O scheme, whenever an address appearing on address bus is for on I/O device, then there is no other information on corresponding memory location.
   2. For e.g. : suppose address 0002 is for output device and whenever this address appears on address bus and is to be used to select output device, there is no information on memory location 0002.

3. In other words, when an address is used to select I/O device, then that address is not used for any memory location.
II) I/O mapped I/O:
1. In I/O mapped I/O, same address may be used for I/O device as well as for memory location.
2. Microprocessor 8085 uses I/O mapped I/O scheme to address I/O devices.
3. Using status signal IO / \( \overline{M} \), differentiation between I/O operations and memory operations is done.
4. When the \( IO / \overline{M} \) is high, the address on the address bus is for an I/O device and when it is low, the address on the address bus is for a memory location.
5. The instruction IN and OUT are used to address I/O devices.
6. The various operation to be carried out are identified by using status \( S_0 \) and \( S_1 \) as follows:

<table>
<thead>
<tr>
<th>Status Signals</th>
<th>Machine Cycle Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO / ( \overline{M} )</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>floating</td>
<td>0 0</td>
</tr>
<tr>
<td>floating</td>
<td>unused</td>
</tr>
<tr>
<td>floating</td>
<td>unused</td>
</tr>
</tbody>
</table>
Functional block diagram of microprocessor 8085
Serial I/O Control

* Most often I/O devices work with serial data transmission
* The 8085 microprocessor has two pins to implement serial transmission SID (Serial Input Data) and SOD (Serial Output Data)
* The 8085 RIM instruction transfer data from SID to bit 7 of microprocessor.
* A single bit may be output via SOD pin of 8085 for this SIM instruction is used.

Instruction Register and Decoder

* During an instruction fetch, the first byte of the instruction i.e., the op-code is transferred to the 8-bit instruction register.
* The content of instruction register is, in turn available to the instruction decoder.
* The output of decoder, geted by timing signal, controls the register, ALU and data/address buffer.
* The output of decoder and internal clock generator produce the state and machine cycle timing signal

* T-State, Machine Cycle and Instruction Cycle:

![Diagram](image)

T-State : The subdivision of an operation which is performed in one clock period is called as T-state

Machine Cycle : Machine cycle is defined as the time required to complete any operation of accessing either memory or I/O which is the subpart of an instruction.
In 8085, the machine cycle may consist of three to six T-states.

Instruction Cycle : An instruction cycle is defined as the time required to complete the execution of an instruction.
The 8085 instruction cycle consist of one to five machine cycles.

The above diagram shows T-states, Machine cycles and Instruction cycle required for execution of an instruction. From the figure it is cleared that an instruction consists of number of machine cycle and a machine cycle consists of number of T-states.
Interrupt

Ans.: 
1. An interrupt is a subroutine called, initiated by external device through hardware (hardware interrupt) or microprocessor itself (software interrupt).

2. An interrupt can also be viewed as a signal, which suspends the normal sequence of microprocessor and then microprocessor gives service to that device which has given the signal. After completing the service, microprocessor again returns to the main program.

3. Microprocessor is connected to different peripheral devices. To communicate with these devices, microprocessor 8085 uses interrupt method.

4. An interrupt is an input signal, which transfers control to specific routine known as Interrupt Service Routine (ISR). After executing ISR, control is again transfer to main program.

5. Microprocessor 8085 has two types of interrupts:
   (i) Software interrupt
   (ii) Hardware interrupt

   The software interrupts has priority than any hardware interrupt.

6. Software interrupts are not requested by external peripheral devices. All software interrupts are non-maskable. Some hardware interrupts are maskable.

Hardware interrupts provided by 8085. List them according to their priority.

1. 8085 provides 5 hardware interrupts:
   (i) TRAP
   (ii) RST 7.5
   (iii) RST 6.5
   (iv) RST 5.5
   (v) INTR

   2. These interrupts are vectored interrupts. It means that when these interrupts are given, it is directed (or vectored) to transfer the control to specific memory location given by:

   TRAP = 4.5×8=0024H
   RST 7.5×8=003CH
   RST 6.5×8=0034H
   RST 5.5×8=002CH

   3. Among these interrupts, TRAP is non-maskable interrupt which can not be disabled. But the other four interrupts are maskable interrupts, which can be disabled.

   4. The TRAP has highest priority and the INTR has lowest priority among the hardware interrupts. The hardware interrupts in descending order of priority are listed below:

   i) TRAP – highest priority
   ii) RST 7.5
   iii) RST 6.5
   iv) RST 5.5
   v) INTR – lowest priority
Software interrupts

1. The normal operation of a microprocessor can be interrupted by special instruction. Such an interrupt is called a **software interrupt**.

2. 8085 provides 8 user defined software interrupts RST 0 to RST 7 where RST means restart.

3. These interrupts are vectored interrupts and when these interrupts are called the control is transferred to the memory location as shown below:

<table>
<thead>
<tr>
<th>Interrupt (mnemonics)</th>
<th>Call Location (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST 0</td>
<td>0000</td>
</tr>
<tr>
<td>RST 1</td>
<td>0008</td>
</tr>
<tr>
<td>RST 2</td>
<td>0010</td>
</tr>
<tr>
<td>RST 3</td>
<td>0018</td>
</tr>
<tr>
<td>RST 4</td>
<td>0020</td>
</tr>
<tr>
<td>RST 5</td>
<td>0028</td>
</tr>
<tr>
<td>RST 6</td>
<td>0030</td>
</tr>
<tr>
<td>RST 7</td>
<td>0038</td>
</tr>
</tbody>
</table>

4. Software interrupts are not used to handle asynchronous events. They are used to call software routines like singles step, break point etc.

5. These interrupts are requested by executing interrupt instructions. They can also be requested due to arithmetic errors.

6. After execution of these interrupts, program counter is incremented. The microprocessor does not execute any interrupt acknowledge cycle. The microprocessor executes normal instruction cycle.

7. These interrupts cannot be ignored or masked. They have more priority than any hardware interrupt.

8. They are not used to interface peripherals. That means they does not improve throughput of the system. They are used in program debugging.

**Differentiate between hardware and software interrupts.**

1) **Hardware interrupts**:
   i) Hardware interrupts are used to handle asynchronous events.
   ii) These interrupts are requested by external device.
   iii) After execution of these interrupts program counter is not incremented.
   iv) The microprocessor executes either interrupt acknowledge cycle or ideal machine cycle to acknowledge this interrupt.
   v) These interrupts may be non-maskable or maskable.
   vi) They have lower priority than any software interrupt.
   vii) These interrupts affects on interrupts control logic.
   viii) It improves throughput of the system.
2) **Software interrupts** :

i) Software interrupts are not used to handle asynchronous events.
ii) These interrupts are not requested by external device but by microprocessor itself.
iii) After execution of these interrupts, program counter is incremented.
iv) The microprocessor does not execute any interrupt acknowledge cycle. It executes normal instruction cycle.
v) They cannot be masked or ignored.
vi) Software interrupts have more priority than any hardware interrupt.
vii) These interrupts do not affect on interrupt control logic.
viii) They do not improve throughput of the system.

**Differentiate between non-maskable and maskable interrupts.**

1) **Non maskable interrupts** :

i) These interrupts cannot be masked or cannot be made pending.
ii) Non maskable interrupt disable all maskable interrupts.
iii) It is used for emergency purpose like power failure, smoke detector, parity check error etc.
iv) It has higher priority than maskable interrupts.
v) It is always vectored interrupt.
vi) Response time for non maskable interrupt is low.

2) **Maskable interrupts** :

i) These interrupts can be masked or made pending.
ii) A maskable interrupt can not disable any non maskable interrupt.
iii) It is used to interface with peripheral devices.
iv) It has lower priority than non-maskable interrupts.
v) It may be vectored or non-vectorred interrupt.
vi) Response time for maskable interrupts is high.