1.1 INTRODUCTION

The microprocessor is one of the most important components of a digital computer. It acts as a brain of a computer system. Computers are of two types: analog computers and digital computers. A digital computer makes processing of digital signals or numbers while analog computer processes analog signals (Continuous quantity) A digital computer is a programmable machine. Its main component is: CPU (Central Processing Unit), memory, input device and output device.

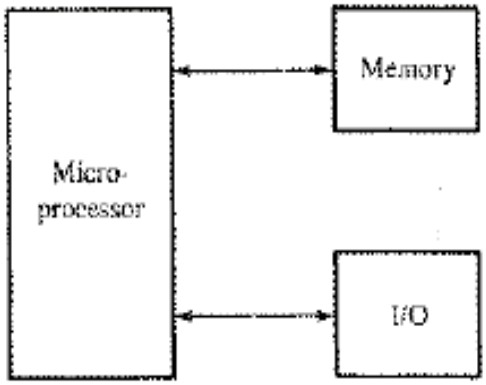

Figure 1.1.1 A Programmable machine

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-4]

The CPU executes instructions given by the programmer. The input device is used to feed programs and data to the computer. The memory is storage device. It stores programs, data and result. The output device displays or prints programs, data and/or results according to the instruction given to the computer.

"A microprocessor is a multipurpose, programmable, clock driven, register- based electronic device that reads binary instructions from storage device called memory, accepts binary data as input and processes data according to those instructions, and provides results as output."

The physical component digital computer system or programmable machine are called hardware. A set of instructions written for microprocessor to perform a task is called a program, and group of programs is called software.

The microprocessor operates in binary digits, 0 and 1, also known as bits. Each microprocessor recognizes and processes a group of bits called the word, and microprocessors are classified according to their word length.

Word Length of a Microprocessor:

The word length of microcomputer or microprocessor is given as "n-bit" where, n= 4, 8, 16, 32 or 64. An 8-bit microprocessor can process 8-bit data at a time. Its ALU (Arithmetic Logic Unit) is of 8 bit, its general purpose registers which holds data for processing, are 8-bit. Similarly, a 16-bit processor handles 16-bit data at a time and its ALU, general purpose registers, are of 16 bits. A processor of longer word length is more powerful and can process data at faster speed.

History of microprocessor:

The first microprocessor INTEL 4004, a 4-bit PMOS microprocessor was introduced in the year 1971 by Intel Corporation, U.S.A. after this a 4-bit microprocessor Intel404, an enhanced version of Intel 4004 was developed. Many other companies also developed 4-bit microprocessor.

In 1972, Intel introduced the first 8- bit microprocessor, Intel 8008 which also uses PMOS technology. The microprocessor using PMOS technology were slow and not compatible with TTL logic. In 1976 Intel developed an improved version of 8-bit NMOS microprocessor, Intel 8085 which uses only one +5V.

Microprocessor	Year of introduction	Word length (bit)	Memory addressing capacity	Pins	Clock	Remarks
4004	1971	4	1 KB	16	750 KHz	First microprocessor
8085	1976	8	64 KB	40	3-6 MHz	Popular 8-bit microprocessor
8086	1978	16	1 MB	40	5-10 MHz	Popular 8-bit microprocessor
8088	1980	8/16	1 MB	40	5-8 MHz	Widely used in PC/XT
80286	1982	16	16 MB real, 4 GB virtual	68	6-12.5 MHz	Widely used in PC/AT
Pentium	1993	32	4 GB real, 32-bit address & 64-bit data bus	237 PGA	60-200 MHz	Contains 2 ALUs
Pentium Pro	1995	32	64 GB real, 36-bit address bus	387 PGA	150-200 MHz	It is data flow processor
Itanium	2001	64	64 address lines	423 PGA	733 MHz-1.5 GHz	64-bit (Explicitily Parallel Instruction Computing) Processor

A Microprocessor-Based System

From the above description, we can draw the following block diagram to represent a microprocessor-based system as shown

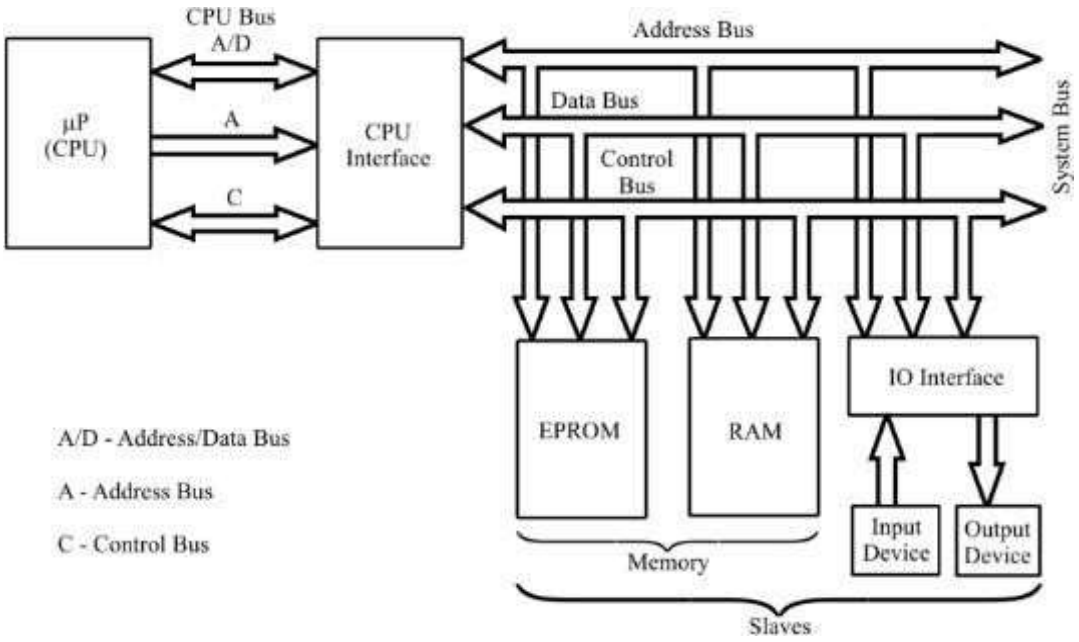

Figure 1.1.1 Microprocessor based system (organization of microcomputer)

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-6]

In this system, the microprocessor is the master and all other peripherals are slaves. The master controls all peripherals and initiates all operations. The buses are group of lines that carry data, address or control signals. The CPU interface is provided to demultiplex the multiplexed lines, to generate the chip select signals and additional control signals. The system bus has separate lines for each signal.

All the slaves in the system are connected to the same system bus. At any time, instant communication takes place between the master and one of the slaves. All the slaves have tristate logic and hence normally remain in high impedance state. The processor selects a slave by sending an address. When a slave is selected, it comes to the normal logic and communicates with the processor.

The EPROM memory is used to store permanent programs and data. The RAM memory is used to store temporary programs and data. The input device is used to enter program, data and to operate system. The output device is also used for examining the results. Since the speed of IO devices does not match with speed of microprocessor, an interface device is provided between system bus and IO device.

A Microprocessor-Based System

From the above description, we can draw the following block diagram to represent a microprocessor-based system as shown

etc. Result are stored either in registers or in memory or sent to output devices.

b) Register Unit:

It contains various register. The registers are used primarily to store data temporarily during the execution of a program. Some of the registers are accessible to the uses through instructions.

c) Control Unit:

It provides necessary timing & control signals necessary to all the operations in the microcomputer. It controls the flow of data between the processor and peripherals (input, output & memory). The control unit gets a clock which determines the speed of the processor.

There are three buses:

Address Bus:

It is a group of wires or lines that are used to transfer the addresses of Memory or I/O devices. It is unidirectional. In Intel 8085 microprocessor, Address bus was of 16 bits. This means that Microprocessor 8085 can transfer maximum 16-bit address which means it can address 65,536 different memory locations. This bus is multiplexed with 8-bit data bus. So the most significant bits (MSB) of address goes through Address bus (A7-A0) and LSB goes through multiplexed data bus (AD0-AD7).

Data Bus:

Data Bus is used to transfer data within Microprocessor and Memory/Input or Output devices. It is bidirectional as Microprocessor requires to send or receive data. The data bus also works as address bus when multiplexed with lower order address bus. Data bus is 8 Bits long. The word length of a processor depends on data bus, that's why Intel 8085 is called 8-bit Microprocessor because it have an 8 bit data bus.

Control Bus:

Microprocessor uses control bus to process data that is what to do with the selected memory location. Some control signals are Read, Write and Opcode fetch etc. Various operations are performed by microprocessor with the help of control bus. This is a dedicated bus, because all timing signals are generated according to control signal. The microprocessor is the master, which controls all the activities of the system. To perform a specific job or task, the microprocessor has to execute a program stored in memory. The program consists of a set of instructions stored in consecutive memory location. In order to execute the program, the microprocessor issues address and control signals, to fetch the instruction and data from memory one by one. After fetching each instruction, it decodes the instruction and carries out the task specified by the instruction.

Memory

To execute a program:

- The user enters its instructions in binary format into the memory.
- The microprocessor then reads these instructions and whatever data is needed from memory, executes the instructions and places the results either in memory or produces it on an output device.

The three cycle instruction execution model

- To execute a program, the microprocessor ―reads‖ each instruction from memory, ―interprets‖ it, then ―executes‖ it.
- To use the right names for the cycles
- The microprocessor fetches each instruction, decodes it, and then executes it. This sequence is continued until all instructions are performed.

1.2 ARCHITECTURE OF INTEL 8085 MICROPROCESSOR

Features of 8085

- Intel 8085 is an 8-bit, NMOS microprocessor.
- It is a 40 pin I.C. package fabricated on a single LSI chip.
- The Intel 8085 uses a single +5Vd.c. supply for its operation
- Its clock speed is about 3 MHz
- The clock cycle is of 320 ns.
- It has 80 basic instructions and 256 opcodes.

Figure 1.2.1 shows the functional block diagram of Intel 8085. It consists of three main sections:

[1] Arithmetic and Logic Unit (ALU)

[2] Timing & Control Unit

[3] Set of registers.

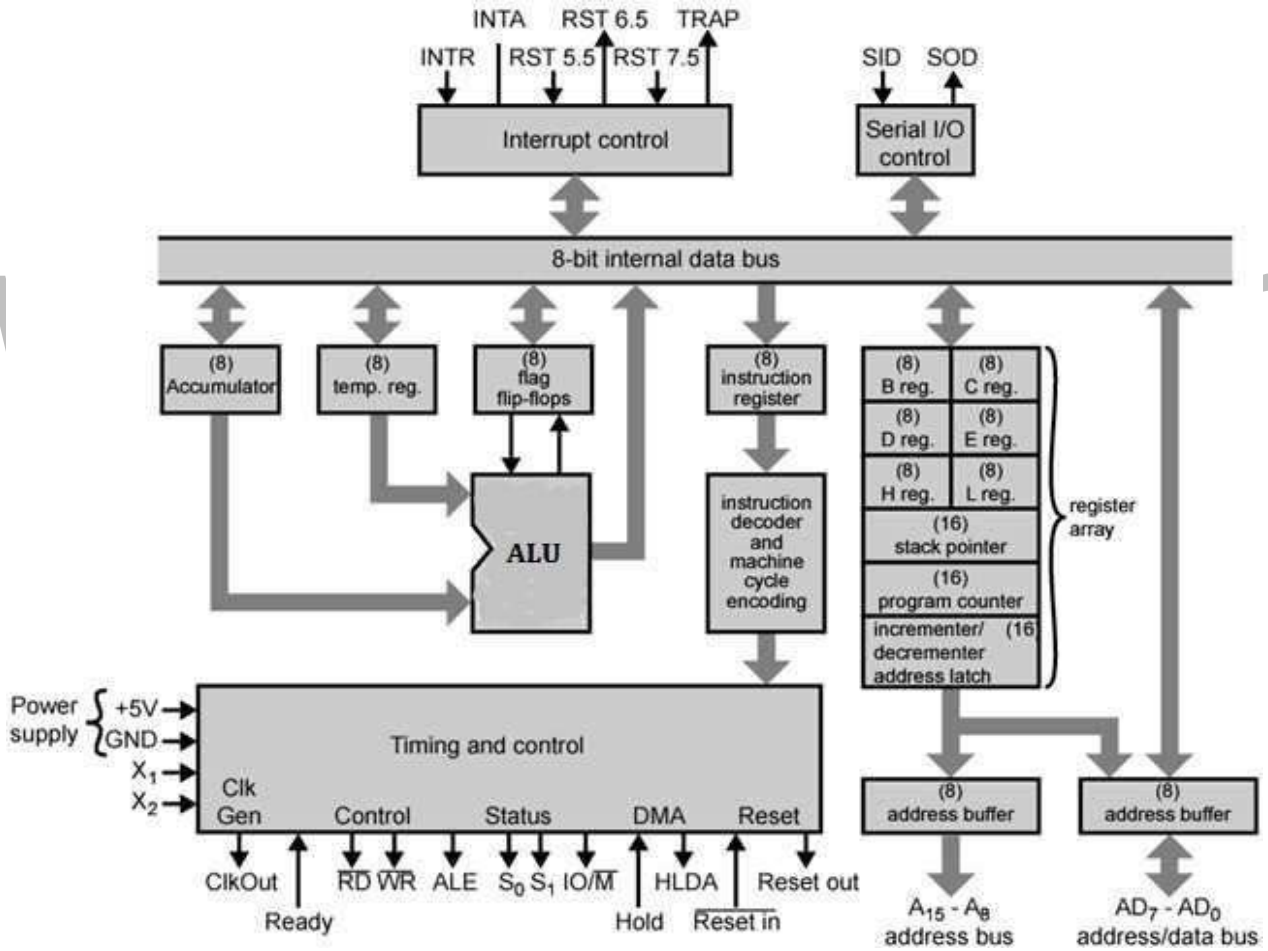

Figure 1.2.1 Architecture 8085 processor

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-72]

1.2 ARCHITECTURE OF INTEL 8085 MICROPROCESSOR

Features of 8085

- Intel 8085 is an 8-bit, NMOS microprocessor.
- It is a 40 pin I.C. package fabricated on a single LSI chip.
- The Intel 8085 uses a single +5Vd.c. supply for its operation
- Its clock speed is about 3 MHz
- The clock cycle is of 320 ns.

v. Logical EXCLUSIVE OR

vi. Complement (logical NOT)

vii. Increment (add 1)

viii. Decrement (Subtract 1)

ix. Left shift, Rotate left, Rorate Right

x. Cler, etc.

2. Timing & Control Unit:

- It generates timing & control signals which are necessary for the execution of instructions.
- It controls data flow between CPU and peripherals including memory.
- It provides status, control and timing signals which are required for the operation of memory and I/O devices.
- It controls entire operations of the microprocessor and peripherals connected to it.
- Thus it is seen that the control unit of the CPU acts as the brain of the computer system.

3. Set of Registers:

- Fig 1.2.1 shows the various registers of Intel 8085 which are used by the microprocessor for temporary storage and manipulation of data and intsructions.
- Intel 8085 microprocessor has the following registers:

i. One 8-bit Accumulator (ACC) i.e register A

ii. Six 8-bit general purpose registers. These are B, C, D, E, H & L

iii. One 16-bit stack pointer, SP

iv. One 16-bit program Counter, PC

v. Intruction register

vi. Temporary register

- In addition to the above mentioned regiter the 8085 microprocessor contains a set of five flipflops which serve as flag (or status flags). A flag is a flipflop which indicates some condition which arises after execution of an arithmetic or logical instruction.

Accumulator:

- The accumulator (register A) is an 8-bit register associated with ALU.
- It is used to hold one of the operands of an arithmetic or logical operation.
- It serves as one input to the ALU. The other operand for an arithmetic or logical opeartion may be stored either in the memroy or in one of the general purpose registers.
- Final result of an arithmetic or logical operation is placed in the accumulator.

General Purpose Registers

- The 8085 microprocessor contains six 8-bit general purpose registers. They are B, C, D, E, H and L.
- To hold 16-bit data a combination of two 8-bit registers can be employed which known as register pairs.
- The valid register pairs in 8051 are B-C, D-E and H-L. the programmer cannot form the register pair by selecting any two registers of his choice.
- The H-L pair is used to act as memory pointer and for this purpose it holds the 16-bit address of a memory location.
- The general purpose registers and the accumulator are accessible to programmer. He can store data in these registers during writing his program.

Program Counter (PC):

- It is a 16-bit special purpose register used to hold the memory address of the next instruction which is to be executed.
- It keeps the track of memory addresses of the instructions in the program while they are being executed.

- The microprocessor increments the content of the program counter during the execution of an instruction so that it points to the address of the next instruction in the program at the end of the execution of an instruction.

Stack Pointer (SP):

- It is a 16-bit special purpose register.
- "Stack is a sequence of memory location set aside by the programmer to store/retrieve the content of accumulator, flags, program counter and general purpose registers during the execution of a program.
- Any portion of the memory can be used as stack. Since the stack works on LIFO (last in first out) principle, its operation is faster compared normal store/ retrieve of memory locations.
- "The SP holds the address of the top element of data stored in the stack.
- "The stack is defined and stack pointer is initialized by the programmer at the beginning of a program which needs stack operation. Stack is also used by the microprocessor. For example, it stores the contents of program counter when it jumps to a subroutine using CALL instruction.

Instruction Register:

- The instruction register holds the opcode (operation code or instruction code) of the instruction which is being decoded and executed.

Temporary Register:

- It is an 8-bit register associated with ALU. It holds data during an arithmetic/logical operation.
- It is used by microprocessor.
- It is not accessible to programmer.

Flags (Program Status Word i.e. PSW):

- The Intel 8085 microprocessor contains five flip-flops to serve as status flags.
- The flip-flops are set or reset according to the conditions which arises during an arithmetic or logical operation.
- The five status flags of Intel 8085 are:
  i. Carry flag (CS)

ii. Parity Flag (P)

iii. Auxiliary Carry Flag (AC)

iv. Zero Flag (Z)

v. Sign Flag (S)

- In fig 1.2.2, five bits indicates the five status flags and three bits are undefined. "The combination of these 8-bits is called Program Status Word (PSW)". PSW and the accumulator are treated as a 16-bit unit for stack operation.

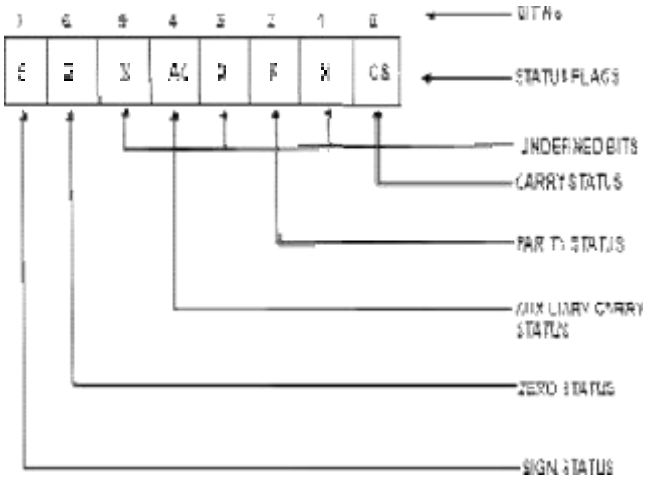

Figure 1.2.2 Flag Register

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-72]

Carry Flag (CS):

- After execution of an arithmetic instruction if carry is produced, the carry flag CS is set to 1. Otherwise it is 0.
- The carry flag is set or reset in case of addition as well as subtraction.
- After the addition of two 8-bit numbers, if the sum is larger than 8-bits, a carry is produce; and the carry flag is set to 1.
- In case of subtraction, if borrow occurs, the carry flag is set to 1.
- The carry flag holds carry out of the most significant bit resulting from the execution of an arithmetic operation.

Parity Flag (P):

- The parity flag P is set to 1, if the result of an arithmetic or logical operation contains even number of 1s.
- It is reset i.e it is 0, if the result contains odd number of 1s.

Auxiliary Carry Flag (AC):

- The auxiliary carry flag AC hold carry out of the bit number 3 to the bit number 4 resulting from the execution of an arithmetic operation.
- The counting of bits starts from 0, and hence Bit No.3 is actually the fourth bit from the least significant bit.

Zero Flag (Z):

- The zero status flag Z is set to 1, if the result of an arithmetic or logical operation is Zero otherwise it is set to 0.

Sign Flag (S):

- The sign flag S is set to 1, if the result of an arithmetic or logical operation is negative. If the result is positive, the sign flag is set to 0.
- The sign flag has significance only when signed arithmetic operation is performed. To represent a signed numbed the most significant bit is reserved by the programmer to represent the sign of a number.
- In other words, the MSB is used as a sign bit.
- If the number is negative, the sign bit is 1. For positive, sign bit is 0.
- In case of 8-bit sign operation, the remaining 7 bits are used to represent the magnitude of a number.

Data and Address bus:

- The Intel 8085 is an 8-bit microprocessor. Its data bus is 8-bit wide and hence, 8bit of data can be transmitted in parallel from or to the microprocessor.
- The Intel 8085 requires a 16-bit wide address bus as the memory addresses are of 16 bits.
- The 8 most significant bits of the address are transmitted by the address bus, a-bus (pins A8 to A15).
- The least significant bits of address are transmitted by address/ data bus, AD-bus (pins AD0 to AD7).
- The address/ data bus transmits data and address at different moments. At a particular moment it transmits either data or address. Thus AD-bus operates in time shared mode.

- This technique is known as multiplexing. First all 16-bit memory address is transmitted by the microprocessor; the 8 MSBs of the address on A-bus and the 8 LSBs of the address on AD- bus. Thus effective width of address bus becomes 16
  bit wide.

1.5 I/O Ports and Data Transfer Concepts

I/O Ports

There are two methods in which I/O devices can be connected to the Microprocessor.

- Memory mapped I/O
- I/O mapped I/O

Memory mapped I/O:

In this method I/O device is treated like the memory. Here there is no IO/M signal. If the processor wants to read the data from a I/O device it will place the address of the I/O device on the address bus. Then the I/O device will get selected. The memory which is having the same address will also get selected.so we have to use separate address for memory and separate address for I/O device.

I/O mapped I/O:

Here we have the IO/M signal. So we can select either the memory or I/O device for read and write operation.

Data Transfer Concepts

- Parallel data transfer
- Serial data transfer

Parallel data transfer

- Programmed I/O
- Interrupt I/O
- DMA

Programmed I/O:

Here the processor has to check whether the I/O device is ready or not through the Ready signal of the I/O device. If the ready signal is high, then it will send the data to the I/O device. Otherwise it will continuously check the Ready signal. The processor is busy in checking the Ready signal. The drawback is wastage of time.

Interrupt I/O:

In this method the I/O device will interrupt the Processor through the INTR signal to indicate to the processor that it is ready to accept the next data. Then the processor will

send the INTA signal. Then the processor stops its normal execution and start transferring the data to the I/O device.

DMA:

Using DMA I/O device can directly transfer the data to the Memory using the Address and Data buses of Processor.

Serial data Transfer

Some of the external I/0 devices receive only the serial data. Normally serial communication is used in the MultiProcessor environment. 8051 has two pins for serial communication.

- SID- Serial Input data.
- SOD-Serial Output data.

Serial data communication can be categorized on the basis of how data transmission occurs. These are:

1. **Simplex:**

   In simplex communication, the hardware is set such that the data exchange takes place in only one direction. Example: Computer to Printer communication.

2. **Half Duplex:**

   The half-duplex communication allows the data exchange in both directions, but not at the same time. Example: Walkie talkie.

3. **Full Duplex:**

   It permits the information transfer in both directions at the same time. Example: Telephone lines.

Interrupts

The processor has the following interrupts:

INTR is a maskable hardware interrupt. The interrupt can be enabled/disabled using STI/CLI instructions or using more complicated method of updating the FLAGS register with the help of the POPF instruction.

When an interrupt occurs, the processor stores FLAGS register into stack, disables further interrupts, fetches from the bus one byte representing interrupt type, and jumps to

interrupt processing routine address of which is stored in location 4 * <interrupt type>. Interrupt processing routine should return with the IRET instruction.

NMI is a non-maskable interrupt. Interrupt is processed in the same way as the INTR interrupt. Interrupt type of the NMI is 2, i.e. the address of the NMI processing routine is stored in location 0008h. This interrupt has higher priority than the maskable interrupt.

Software interrupts can be caused by:

- INT instruction - breakpoint interrupt. This is a type 3 interrupt.
- INT <interrupt number> instruction - any one interrupt from available 256 interrupts.
- INTO instruction - interrupt on overflow
- Single-step interrupt - generated if the TF flag is set. This is a type 1 interrupt. When the CPU processes this interrupt it clears TF flag before calling the interrupt processing routine.

Synchronous Data Transfer

- The word 'Synchronous 'means 'taking place at the same time.'
- Thus, to establish communication between our processor and the device, we need to set a common clock pulse. This common pulse synchronizes the peripheral device with the 8085 microprocessor.
- This method is used when the speed of the microprocessor, Intel 8085, in this case, and the external peripheral device match with each other.
- If the device is ready to send data, it can indicate via the READY pin of 8085. Once the speeds match, the data transfer immediately begins, once a signal is issued by the microprocessor to begin transferring. The microprocessor need not wait for an extended period because of the matching speeds.
- This technique of data transfer is seldom used to communicate with I/O devices though. Because I/O devices compatible with the microprocessor's speed are usually not found.
- Hence, this method of data transfer is most commonly employed for communicating with compatible memory devices.

Asynchronous Data Transfer

- When the speed of the I/O device is slower than that of the microprocessor, we prefer the Asynchronous Data Transfer Method. As the speeds of both the devices differ, the I/O device's internal timing is entirely independent of the microprocessor.
- Thus, they are termed to be 'asynchronous 'from each other. The term asynchronous means 'at irregular intervals.'

1.7 INTERRUPTS

Interrupt is a signal send by an external device to the processor, to the processor to perform a particular task or work. Mainly in the microprocessor based system the interrupts are used for data transfer between the peripheral and the microprocessor.

When a peripheral is ready for data transfer, it interrupts the processor by sending an appropriate signal to the interrupt pin of the processor. If the processor accepts the interrupt then the processor suspends its current activity and executes an interrupt service subroutine to complete the data transfer between the peripheral and processor. After executing the interrupt service routine the processor resumes its current activity. This type of data transfer scheme is called interrupt driven data transfer scheme.

Types of Interrupts

The interrupts are classified into software interrupts and hardware interrupts.

- The software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, lf a software interrupt instruction is encountered, then the processor executes an interrupt service routine (ISR).
- The hardware interrupts are initiated by an external device by placing an appropriate signal at the interrupt pin of the processor. If the interrupt is accepted, then the processor executes an interrupt service routine (ISR).

Software Interrupts of 8085

The software interrupts are program instructions. When the instruction is executed, the processor executes an interrupt service routine stored in the vector address of the software interrupt instruction. The software interrupts of 8085 are RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6 and RST 7.

Table 1.7.1 Interrupt vector address

Interrupt	Vector address
RST 0 RST 1	0000_H 0008_H
RST 2 RST 3	0010_H 0018_H
RST 4 RST 5	0020_H 0028_H
RST 6 RST 7	0030_H 0038_H

Interrupt	Vector address
RST 7.5	$003C_H$
RST 6.5	0034_H
RST 5.5	$002C_H$
TRAP	0024_H

The software interrupt instructions are included at the appropriate (or required) place in the main program. When the processor encounters the software instruction, it pushes the content of PC (Program Counter) to stack. Then loads the Vector address in PC and starts executing the Interrupt Service Routine (ISR) stored in this vector address. At the end of ISR, a return instruction - RET will be placed. When the RET instruction is executed, the processor POP the content of stack to PC. Hence the processor control returns to the main program after servicing the interrupt. Execution of ISR is referred to as servicing of interrupt. All software interrupts of 8085 are vectored interrupts. The software interrupts cannot be masked and they cannot be disabled.

Hardware Interrupts of 8085

An external device, initiates the hardware interrupts of 8O85 by placing an appropriate signal at the interrupt pin of the processor. The processor keeps on checking the interrupt pins at the second T -state of last machine cycle of every instruction. If the processor finds a valid interrupt signal and if the interrupt is unmasked and enabled, then the processor accepts the interrupt. The acceptance of the interrupt is acknowledged by sending an INTA signal to the interrupted device.

The processor saves the content of PC (program Counter) in stack and then loads the vector address of the interrupt in PC. (If the interrupt is non-vectored, then the interrupting device has to supply the address of ISR when it receives INTA signal). It starts executing ISR in this address. At the end of ISR, a return instruction, RET will be placed. When the processor executes the RET instruction, it POP the content of top of stack to PC. Thus the processor control returns to main program after servicing interrupt. The hardware interrupts of 8085 are TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR. Further the interrupts may be classified into vectored and non-vectored interrupts.

Vectored Interrupt

In vectored interrupts, the processor automatically branches to the specific address in response to an interrupt.

Non-Vectored Interrupt

But in non-vectored interrupts the interrupted device should give the address of the interrupt service routine (ISR).

In vectored interrupts, the manufacturer fixes the address of the ISR to which the program control is to be transferred.

The type of signal that has to be placed on the interrupt pin of hardware interrupts of 8085 are defined by INTEL.

- The TRAP interrupt is edge and level sensitive. Hence, to initiate TRAP, the interrupt signal has to make a low to high transition and then it has to remain high until the interrupt is recognized.
- The RST 7.5 interrupt is edge sensitive (positive edge). To initiate the RST 7.5, the interrupt signal has to make a low to high transition and it need not remain high until it is recognized.
- The RST 6.5, RST 5.5 and INTR are level sensitive interrupts. Hence for these interrupts the interrupting signal should remain high, until it is recognized.

Maskable & Non-Maskable Inetrrupts:

The hardware vectored interrupts are classified into maskable and non-maskable interrupts.

- TRAP is non-maskable interrupt
- RST 7.5, RST 6.5 and RST 5.5 are maskable interrupt.

Masking is preventing the interrupt from disturbing the main program. When an interrupt is masked the processor will not accept the interrupt signal. The interrupts can be masked by moving an appropriate data (or code) to accumulator and then executing SIM instruction. (SIM - Set Interrupt Mask). The status of maskable interrupts can be read into accumulator by executing RIM instruction (RIM - Read Interrupt Mask).

All the hardware interrupts, except TRAP are disabled, when the processor is resetted. They can also be disabled by executing Dl instruction. (Dl-Disable Interrupt).

- When an interrupt is disabled, it will not be accepted by the processor. (i.e., INTR, RST 5.5, RST 6.5 and RST 7.5 are disabled by DI instruction and upon hardware reset).
- To enable (to allow) the disabled interrupt, the processor has to execute El instruction (El-Enable Interrupt).

Interrupt Driven Data Transfer Scheme

The interrupt driven data transfer scheme is the best method of data transfer for effectively utilizing the processor time. In this scheme, the processor first initiates the I/O device for data transfer. After initiating the device, the processor will continue the execution of instructions in the program. Also at the end of an instruction the processor will check for a valid interrupt signal. If there is no interrupt, then the processor will continue the execution.

When the I/O device is ready, it will interrupt the processor. On receiving an interrupt signal, the processor will complete the current instruction execution and saves the processor status in stack. Then the processor calls an interrupt service routine (ISR) to service the interrupted device. At the end of ISR the processor status is retrieved from stack and the processor starts executing its main program. The sequence of operations for an interrupt driven data transfer scheme is shown in figure below.

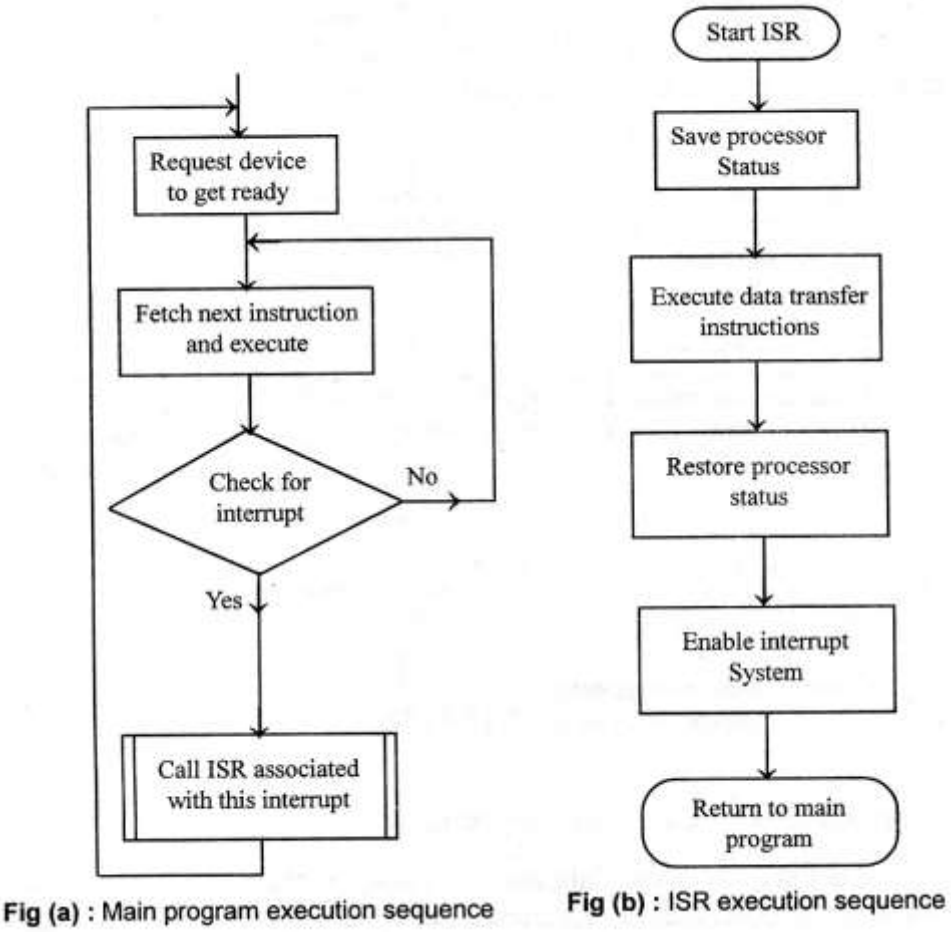

Fig (a) : Main program execution sequence

Fig (b) : ISR execution sequence

Figure 1.7.1 sequence of operations for an interrupt

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-295]

1.4 MEMORY STRUCTURE IN 8085 PROCESSOR

Memory Structure and Its Requirements:

- Read/Write memory is a group of registers to store binary information. Fig 1.4.1 shows a typical R/W memory chip; it has "2n = M" registers (where n= no. of address lines) and each can store N no. of bits.
- It has N no. of bidirectional (or separate input-output) data lines.
- It also has one Chip select (CS), and two control lines Read (RD) to enable the output buffer and Write (WR) to enable the input buffer.
- Figure. 1.4.1 shows the logic diagram of typical EPROM (Erasable Programmable Read Only memory) Memory with "2n = M" registers (where n= no. of address lines).
- It has "n" no. of address lines, one Chip select (CS) and one Read (RD).
- This memory chip must be programmed before it can be used as Read-Only memory.

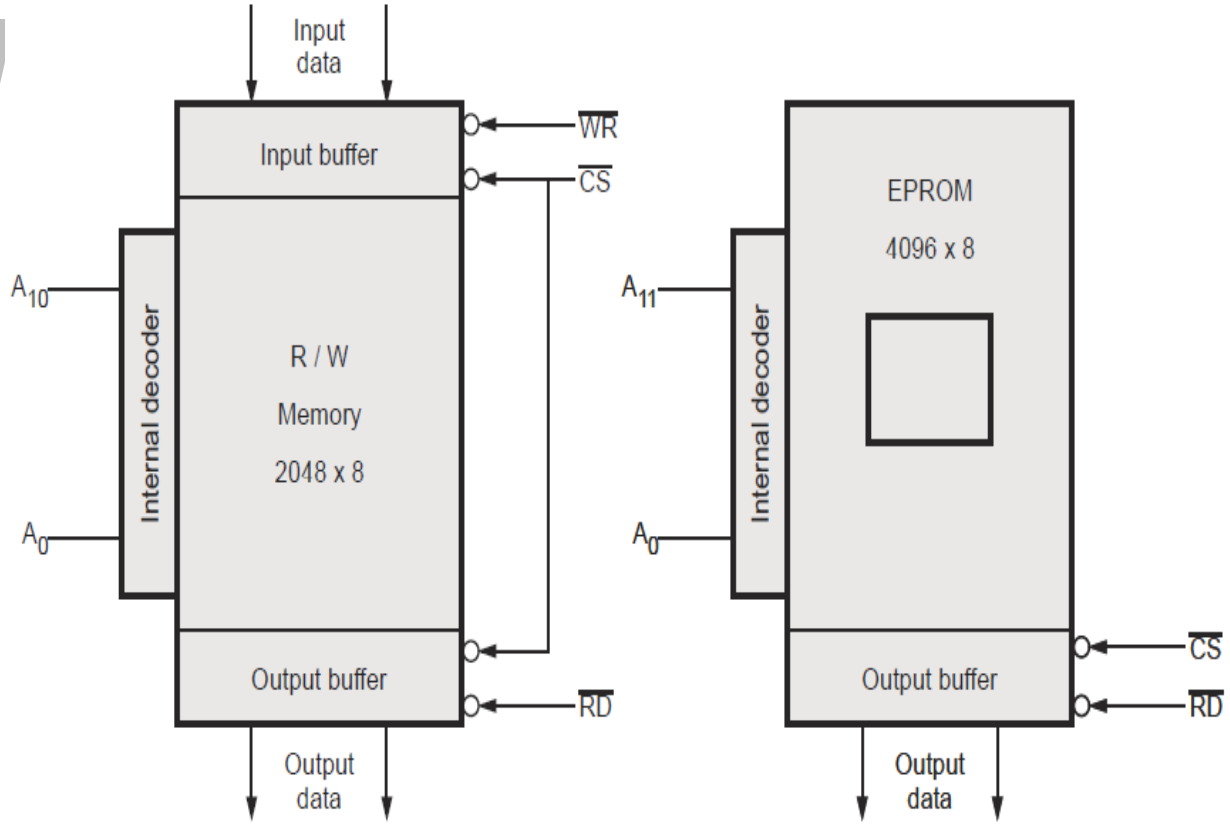

Figure 1.4.1 logic diagram of typical EPROM

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-91]

The 8085 microprocessor uses a16-bit wide address bus for addressing memories and I/O devices. Using 16-bit wide address bus it can access 216 = 64K bytes of memory and I/O devices. The 64K addresses are to be assigned to memories and I/O devices for their addressing. There are two schemes for the allocation of addresses to memories and input / output devices:

1. Memory mapped I/O scheme.
2. I/O mapped I/O scheme.

Memory mapped I/O scheme:

- In memory mapped I/O scheme there is only one address space.
- Address space is defined as the set of all possible addresses that microprocessor can generate.
- Some addresses are assigned to memories and some addresses to I/O devices.
- An I/O device is also treated as memory location and one address is assign to it.
- Suppose that memory locations are assigned the addresses 2000 H to 24FF H then each one address is assign to each memory location. The addresses for I/O devices are different from the addresses which have been assigned to memories.
- The addresses which have not been assigned to memories can be assigned to each I/O device.
- In this scheme all the data transfer instructions of the microprocessor can be valid for data transfer from the memory location or I/O device whose address is in H-L pair.

I/O mapped I/O scheme:

- In this scheme the addresses assigned to memory location can also be assigned to I/O devices.
- Since the same address may be assigned to memory location or an I/O device, the microprocessor must issue a signal to distinguish whether the address on the address bus is for a memory location or an I/O device.
- The 8085 issues an IO/M signal for this purpose.
- Two extra instructions IN and OUT are used to address I/O devices.

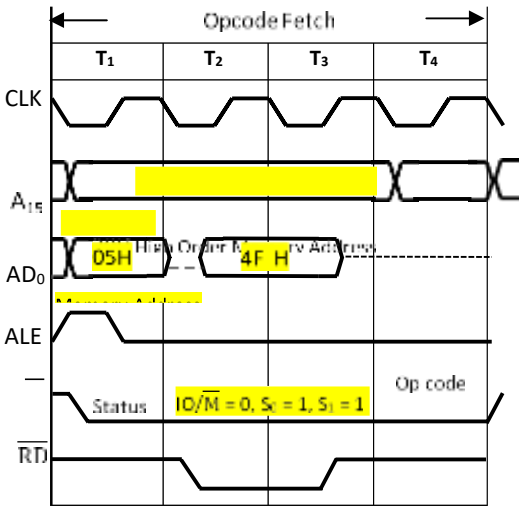

Figure 1.4.2 Timing Diagram: Transfer of Byte from Memory to Microprocessor

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-66]

- Figure 1.4.2 shows the timing diagram of how a data byte is transferred from memory to the microprocessor from which the need for demultiplexing the bus AD0-AD7 becomes easier to understand.
- This figure shows that the address on the high order bus (20H) remains on the bus for three clock periods.
- However, the low order address (05H) is lost after the first clock period.
- This address needs to be latched and used for identifying the memory address.
- If the bus AD7-AD0 is used to identify the memory location (2005H), the address will change to 204FH after the first clock period.
- Figure. 1.4.2 shows a schematic that uses a latch and the ALE signal to demultiplex the bus. The bus AD7-AD0 is connected as input to latch 74LS373.
- The ALE signal is connected to Enable (G) pin of the Latch, and the Output control (OC) signal of the latch is grounded.
- Figure 1.4.2 shows that the ALE goes high during T1. When the ALE is high, the latch is transparent; that means output changes according to the input data.
- During T1 output of the latch is 05H.
- When ALE goes low, the data byte 05H is latched until the next ALE, and the output of the latch represents the low-order address bus A7-A0 after the latching operation.

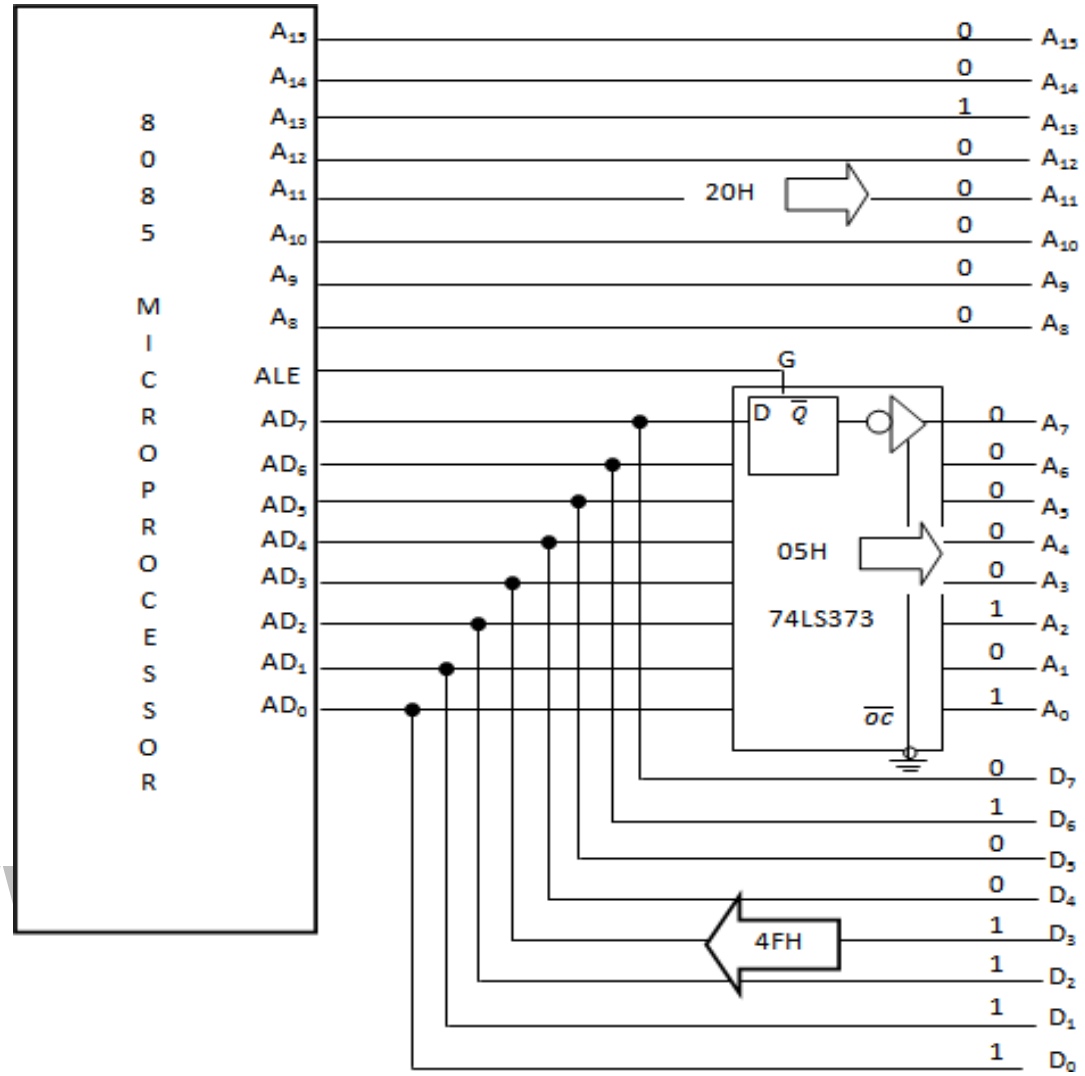

Figure 1.4.3 Schematic of Latching Low-Order Address Bus

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-82]

Generating Control Signals:

- Figure 1.4.3 shows the RD (Read) as a control signal. Because this signal is used both for reading memory and for reading an input device, it is necessary to generate two different Read signals: one for memory and another for input. Similarly, two separate Write signals must be generated.
- This signal is Indeed with RD and WR signals by using the 74LS32 quadruple two-input OR gates, as shown in Figure 1.4.3.
- When both the input goes low , the outputs of the gates go low and generate MEMR (memory Read) and MEMW (memory Write) control Signals.

- When IO/M goes high, it indicates the peripheral I/O operation. The table shows that this signal is complemented and ANDed with RD and WR signals to generate IOR (I/O Read) and IOW (I/O Write) control Signals.

$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$	**Control Signal**
0	0	1	Memory read
0	1	0	Memory write
1	0	1	I/O read
1	1	0	I/O write

Address Decoding:

- The address of a memory location is sent out by the microprocessor. The corresponding memory chip or I/O device is selected by a decoding circuit.
- The decoding task can be performed by a decoder, a comparator, a bipolar PROM or PLA (Programmed logic array).
- For memory interfacing we can use Decoder IC 74LS138 which has G1, G2A and G2B enable signals.
- To enable 74LS138, G1 should be high, and G2A and G2B should be low. Also, 74LS138 has three select lines A, B & C. By applying proper logic to select ines any one of the output cab be selected.
- Where, 74LS138 has Y0, Y1, …. Y7 output lines. An output lines goes low when it is selected. Other output lines remain high. Thus as shown in Table1.1 (Truth table for 74LS138), when G1 is low or G2A is high or G2B is high, all output lines become high.
- Thus 74LS138 acts as decoder only when G1 is high, and G2A and G2B are low.

INPUTS						OUTPUTS							
ENABLE			SELECT										
G1	G2A	G2B	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	H	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	**L**	H	H	H	H	H	H	H
H	L	L	L	L	H	H	**L**	H	H	H	H	H	H
H	L	L	L	H	L	H	H	**L**	H	H	H	H	H
H	L	L	L	H	H	H	H	H	**L**	H	H	H	H
H	L	L	H	L	L	H	H	H	H	**L**	H	H	H
H	L	L	H	L	H	H	H	H	H	H	**L**	H	H
H	L	L	H	H	L	H	H	H	H	H	H	**L**	H
H	L	L	H	H	H	H	H	H	H	H	H	H	**L**

Following fig shows the simple interfacing of 8K memory with 8085

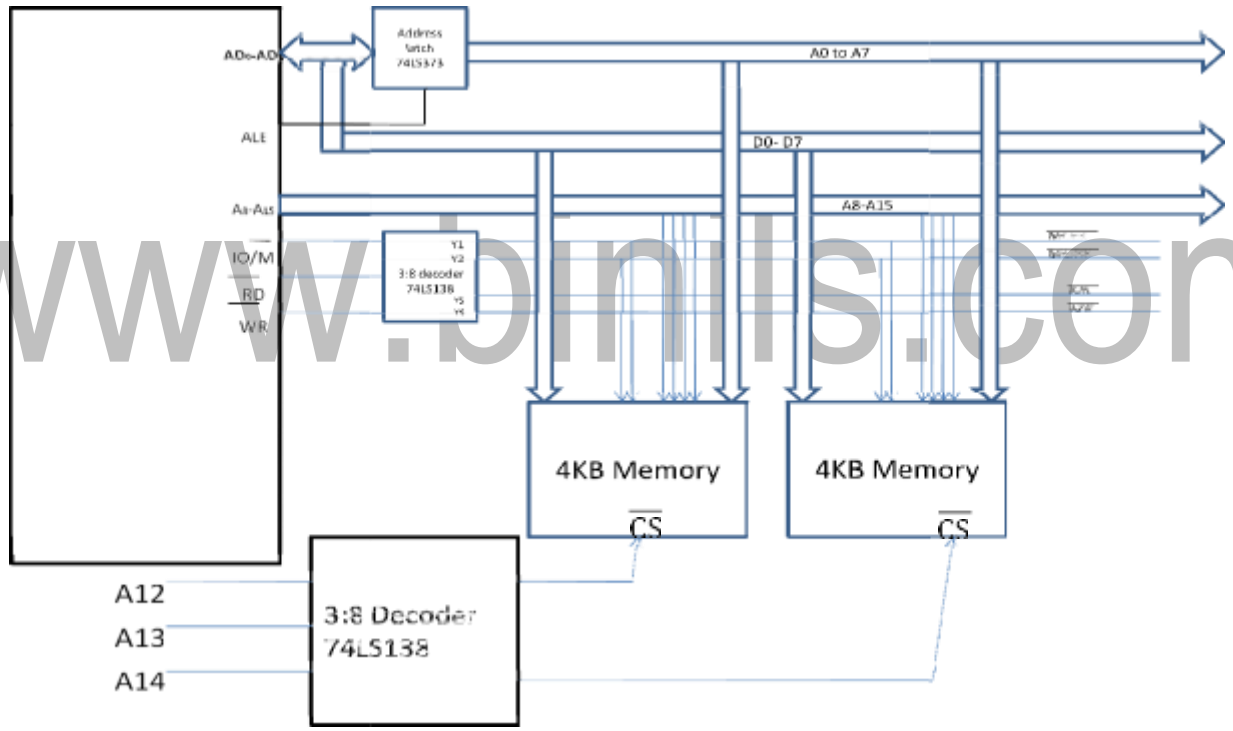

Figure 1.4.4 simple interfacing of 8K memory with 8085

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-82]

1.3 PINOUTS IN 8085 MICROPROCESSOR

The description of various pins is as follows,

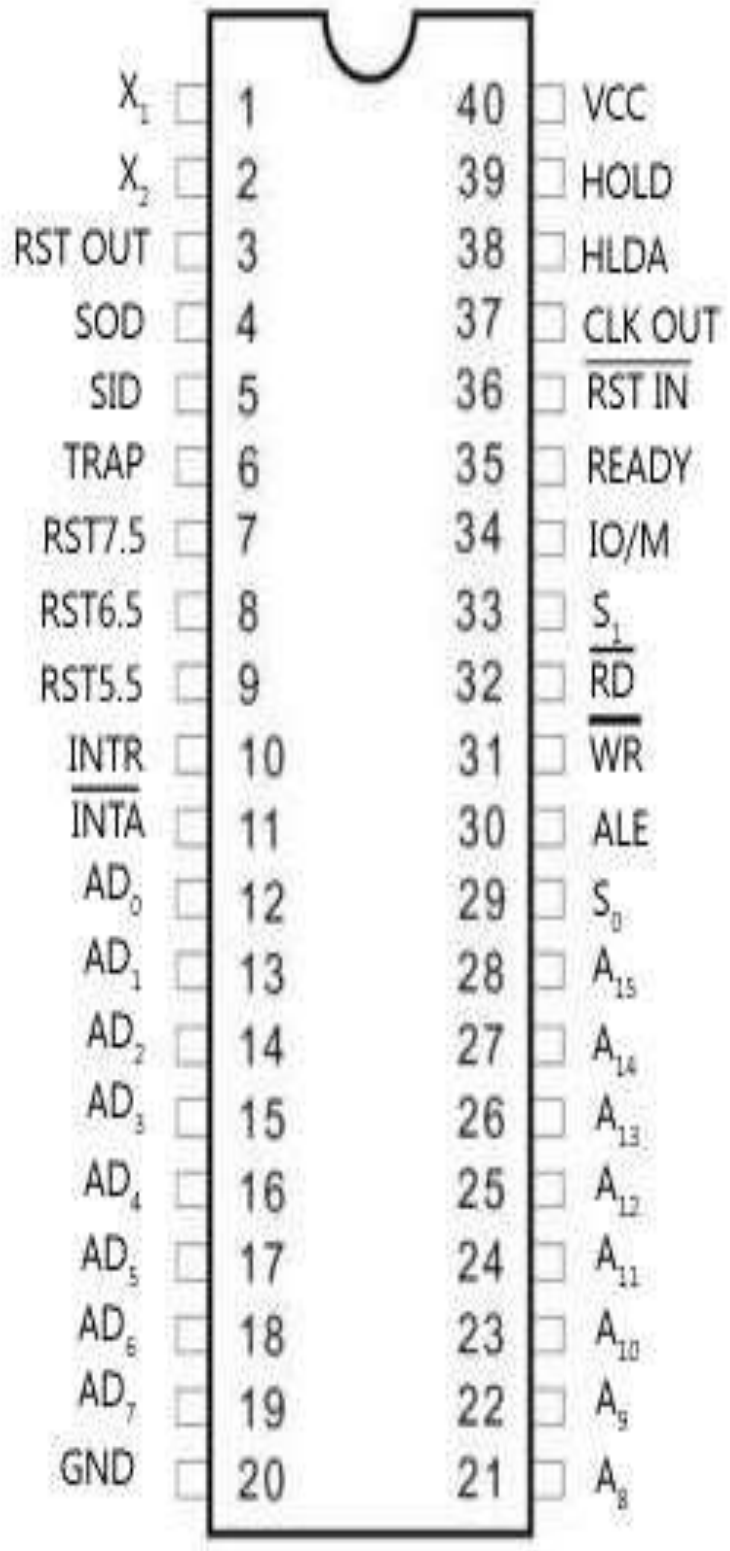

Figure 1.3.1 Pinouts of 8085 processor

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-41]

A8 – A15:

- These are the output lines
- These are the parallel address lines called as Address bus and are used for the most significant bits of the memory address or 8 bits of I/O address.

AD0 – AD7:

- Input / Output lines
- These are time multiplexed address/data bus i.e. they serve dual purpose.
- They are used for the least significant 8 bits of memory address or I/O address during the first clock cycle of a machine cycle.
- Again they are used for data during second and third clock cycles.

ALE:

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

IO/M:

- It is an output status signal which distinguishes whether the address is for memory or I/O.
- When it goes high the address on the address bus is for an I/O device.
- When it goes low the address on the address bus is for a memory location.

S0, S1:

- These are the output status signals sent by the microprocessor to distinguish the various types of operations as given below.
- Status codes for 8085,

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

RD:

- It is an output control signal.

- It controls the READ operation.
- When it goes low selected memory or I/O device is read.

WR:

- It is an output control signal.
- It controls the WRITE operation.
- When it goes low data on the data bus is written into the selected memory or I/O location.

READY: (input)

- It is used by the microprocessor to sense whether a peripheral is ready to transfer data or not.
- A slow peripheral may be connected to the microprocessor through READY line.
- If READY is high peripheral is ready.
- If it is low microprocessor waits till it goes high.

HOLD (input):

- It indicates that another device is requesting for the use of address and data bus.
- After receiving the HOLD request microprocessor handover the controls of buses to that devices as soon as current machine cycle is completed. Internal processing may continue.
- The processor regains the control over the buses after removal of HOLD signal.
- When hold is acknowledged, address bus, data bus, RD, WR, and IO/M are tri-stated.
- HOLD is sampled in T2 Clock Cycle.

HLDA (output):

- It is signal for HOLD acknowledgement.
- It indicates that the HOLD request has been received.
- After removal of a HOLD request the HLDA goes low.
- The CPU takes over the buses half clock cycle after the HLDA goes low.

INTR (input):

- It is an interrupt request signal.
- Among interrupts it has lowest priority.

- When it goes high the program counter does not increment its content.
- The microprocessor suspends its normal sequence of instructions.
- After completing the instruction at hand it attends the interrupting device.
- The INTR line is sampled in the last state of the last machine cycle of an instruction.
- The INTR is enabled or disabled by software.
- An interrupt is used by I/O device to transfer data to microprocessor without wasting time.
- If CPU is in HOLD state or interrupt enable flip-flop is reset, an interrupt request is not honored.

INTA: (output)

- It is an interrupt acknowledgement sent by the microprocessor after INTR is received.

RST 5.5, 6.5 and TRAP: (inputs)

- These are the interrupts.
- When an interrupt is recognized the next instruction is executed from a fixed location in the memory as given below:

Interrupt	Vectored location
TRAP	0024
RST5.5	002C
RST6.5	0034
RST7.5	003C

- RST 7.5, RST6.5 and RST 5.5 are the restart interrupts. They cause an internal restart to be automatically inserted, each them has programmable mask.
- The TRAP has highest priority among the interrupts and it is non-maskable interrupt.

- The order of priority of interrupts is as follows:

  TRAP (Highest Priority)

  RST7.5

  RST6.5 RST5.5

  INTR (Lowest Priority)

RESET IN: (input)

- It resets the program counter to 0.
- It also resets interrupt enable and HLDA flip-flops.
- It does not affect any other flag or register except the instruction register.
- The CPU is held in reset condition as long as RESET is applied.

RESET OUT: (output)

- It indicates that the CPU is being reset.

X1, X2: (input)

- These are terminals connected to an external crystal oscillator which drives an internal circuitry of the microprocessor to produce a suitable clock for the operation of microprocessor.

CLK: (output)

- It is a clock output for user, which can be used for other digital ICs. Its frequency is same at which microprocessor operates.

SID: (input)

- It is a data line for serial input.
- The data on this line is loaded into the 7th nit of the accumulator when RIM instruction is executed.

SOD: (output)

- It is a data line for serial output. The 7th bit of the accumulator is output on SOD line when SIM instruction is executed.

Vcc:

- +5 volts supply.

Vss:

- Ground reference.

1.6 TIMING DIAGRAM

The machine cycles are the basic operations performed by the processor, while instructions are executed. The time taken for performing each machine cycle is expressed in terms of T-states.

Timing Diagram is a graphical representation of the execution of an instruction by a processor.The execution time is represented in T-states. One T-state is the time period of one clock cycle of the microprocessor.

Parts of an instruction

Instruction:-

- It is a command which direct the processor to execute certain task.
  Ex:- MOV A,B

Operator:

- what operation the MP will perform.
  Ex:-MOV

Operand:-

- The source of data on which instruction is to be operated
  Ex:- A,B

Classification of instruction based on byte length

- 1 byte instruction
- 2 byte instruction
- 3 byte instruction

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

- It is the time corresponding to one clock period.
- A portion of an operation carried out in one system clock period is called as T-state.

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

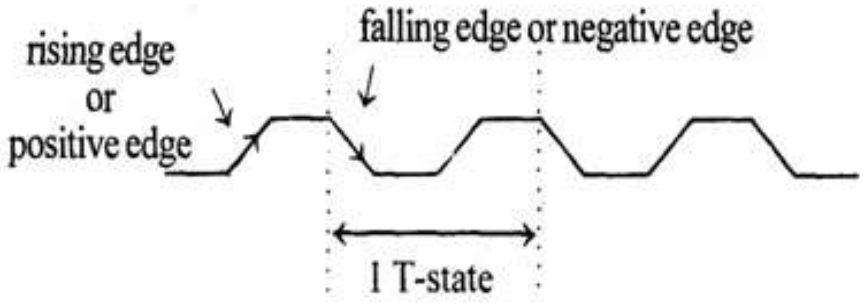

Figure 1.6.1 T-State

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-65]

Machine Cycles of 8085

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)

IO/M	S0	S1	Processor state
High impedance	0	0	Halt
0	0	1	Memory read
0	1	0	Memory write
0	1	1	Op-code fetch
1	0	1	I/O Read
1	1	0	I/O Write
1	1	1	Interrupt acknowledgement

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 opcode fetch machine cycle.
- The time taken by the processor to execute the opcode fetch cycle is 4T.
- In this time, the first, 3T-states are used for fetching the opcode from memory and the remaining T-states are used for decoding. & execution.

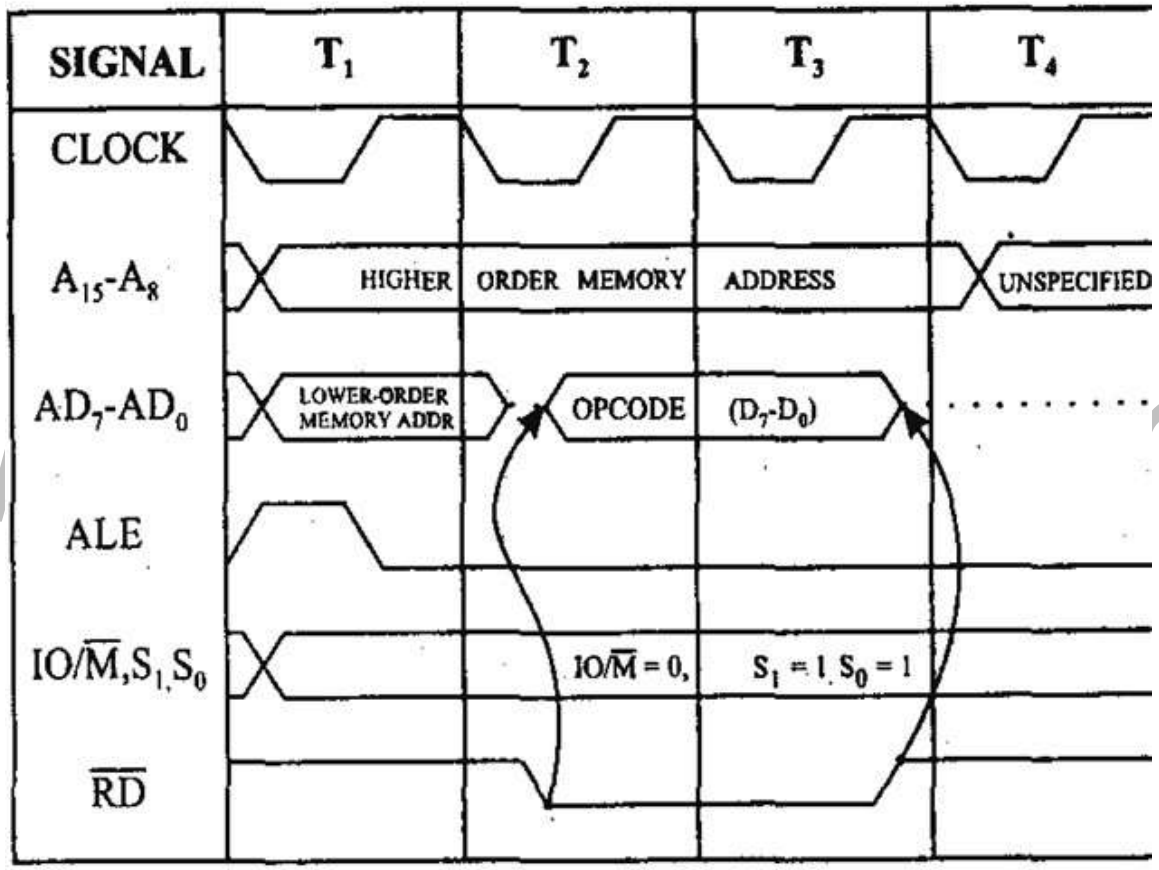

Figure 1.6.2 Opcode fetch machine cycle

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-65]

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.

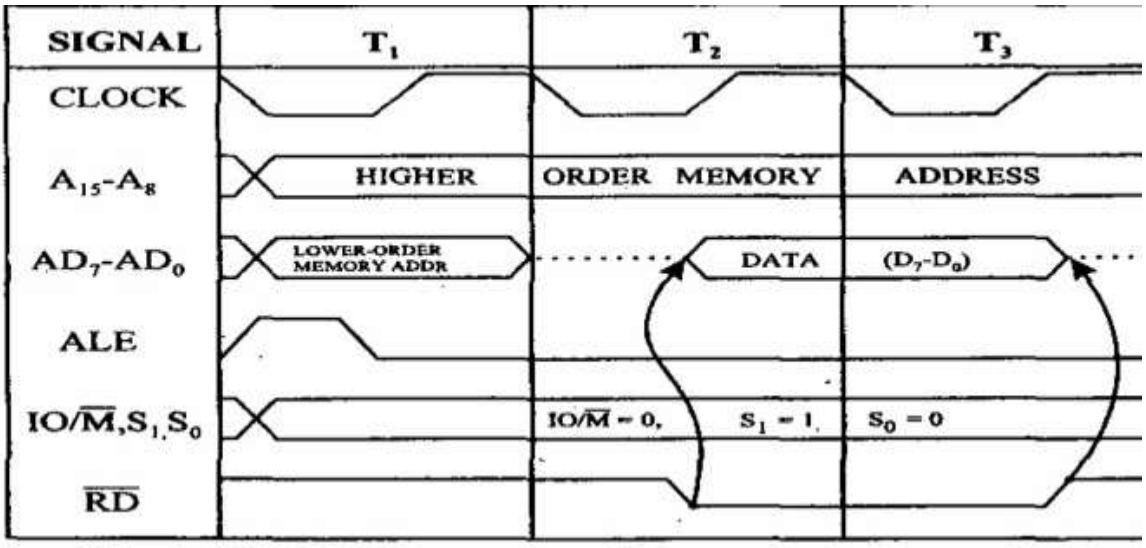

Figure 1.6.3 Memory Read Machine Cycle

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-66]

Memory Write Machine Cycle

- 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.
- The 8085 places the address (2065H) on the address bus
- Identifies the operation as a 'memory write' (IO/M=0, s1=0, s0=1).
- Places the contents of the accumulator on the data bus and asserts the signal WR.
- During the last T-state, the contents of the data bus are saved into the memory location.

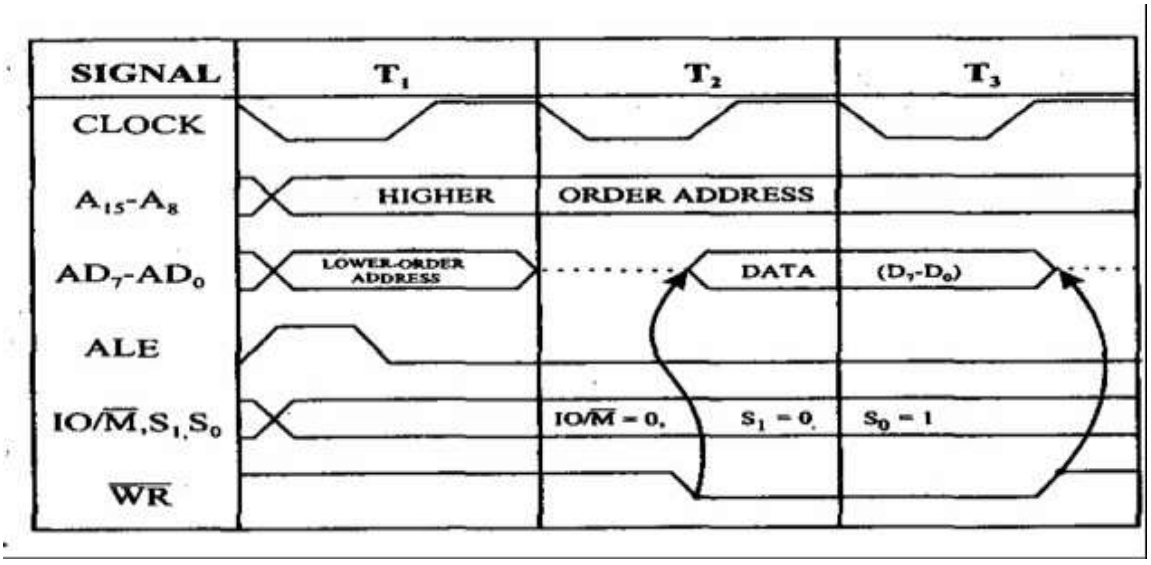

Figure 1.6.4 Memory write Machine Cycle

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-68]

I/O Read Cycle Of 8085

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

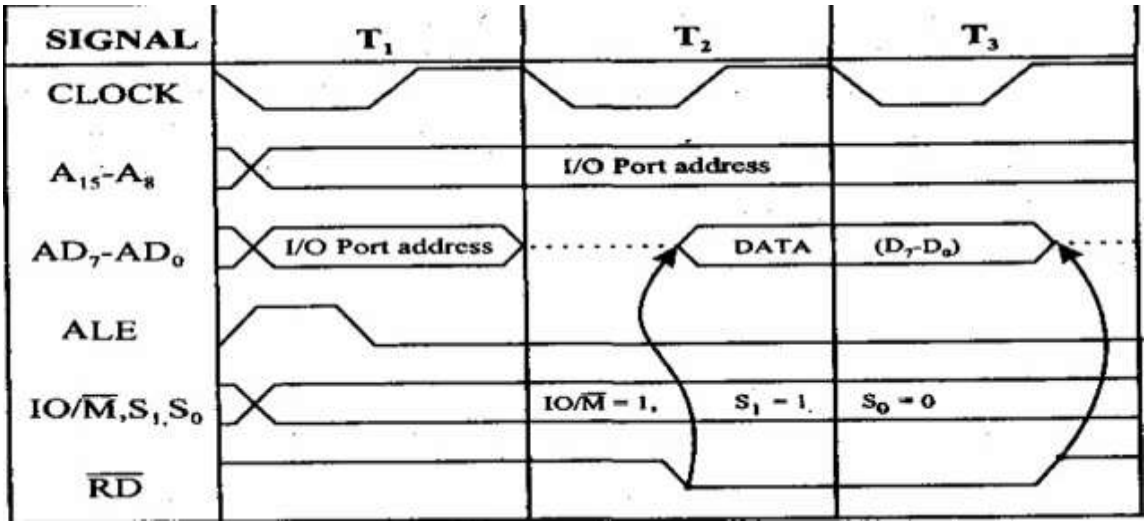

Figure 1.6.5 I/O Read Machine Cycle

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-79]

I/O Write Cycle Of 8085

- The I/O write machine cycle is executed by the processor to write a data byte in the I/O port or to a peripheral, which is I/O, mapped in the system.
- The processor takes, 3T states to execute this machine cycle.

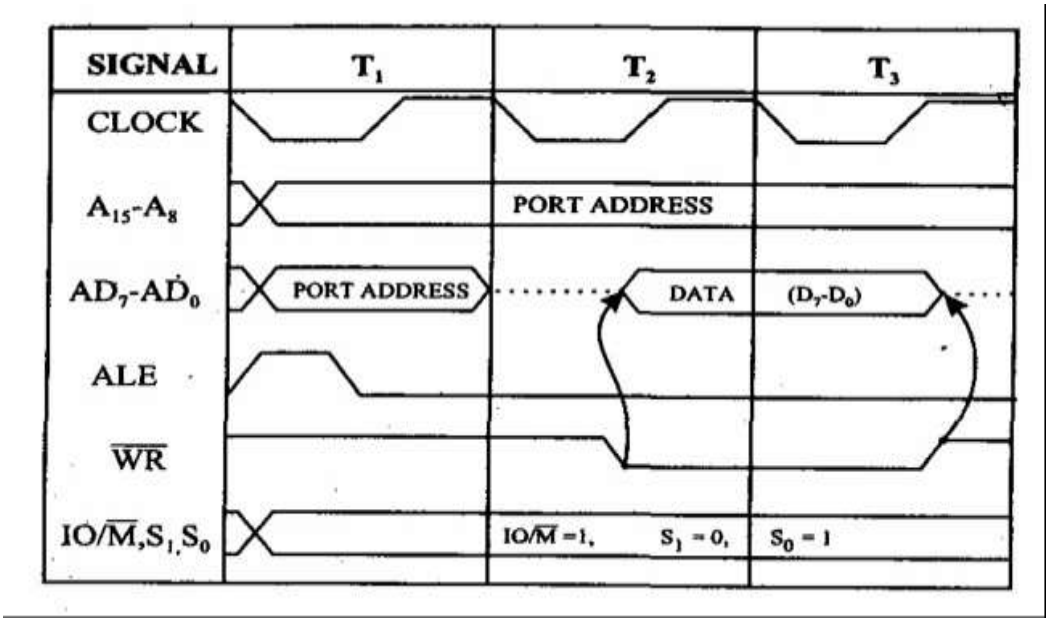

Figure 1.6.5 I/O Write Machine Cycle

[Source: "Microprocessor Architecture Programming and Application" by R.S. Gaonkar, page-79]