

## UNIT II

# MICROPROCESSOR AND MICROCONTROLLER

- ❖ Introduction
- ❖ Architecture of 8085
- ❖ Pin Configuration
- ❖ Addressing Modes
- ❖ Instruction set
- ❖ Timing diagram of 8085
- ❖ Concepts of 8051 microcontroller
- ❖ Block diagram

**Q.1 What is microprocessor ?**

**Ans. :** It is a program controlled semiconductor device (IC), which fetches, decode and executes instructions.

**Q.2 What are the basic units of a microprocessor ?**

**Ans. :** The basic units or blocks of a microprocessor are ALU, an array of registers and control unit.

**Q.3 What is the function of microprocessor in a system?**

**Ans. :**

- The microprocessor is the master in the system, which controls all the activity of the system.
- It issues address and control signals and fetches the instruction and data from memory.
- Then it executes the instruction to take appropriate action.

**What is a Microcontroller ?**

- A single chip computer or A CPU with all the peripherals like RAM, ROM, I/O Ports, Timers, ADCs etc... on the same chip.
- For ex : Motorola's 6811, Intel's 8051, Zilog's Z8 and PIC 16X etc.

## Salient Features of 8085 Microprocessor

- 8085 is an 8 bit microprocessor, manufactured with N-MOS technology.
- It has 16 - bit address bus and hence can address up to  $2^{16} = 65536$  bytes (64 kB) memory locations through  $A_0 - A_{15}$ .
- The first 8 lines of address bus and 8 lines of data bus are multiplexed  $AD_0 - AD_7$ . Data bus is a group of 8 lines  $D_0 - D_7$ .
- It supports external interrupt request. 8085 consists of 16 bit program counter (PC) and stack pointer (SP).
- Six 8 - bit general purpose register arranged in pairs : BC, DE, HL.
- It requires a signal +5 V power supply and can operate at 3 MHz, 5 MHz and 6 MHz serial in/serial out port.
- It is enclosed with 40 pins DIP (Dual in line package).

## 2.2 Internal Architecture of 8085

[ AU : Dec.-2016, 8 Marks ]

The architecture of 8085 consist various components like :

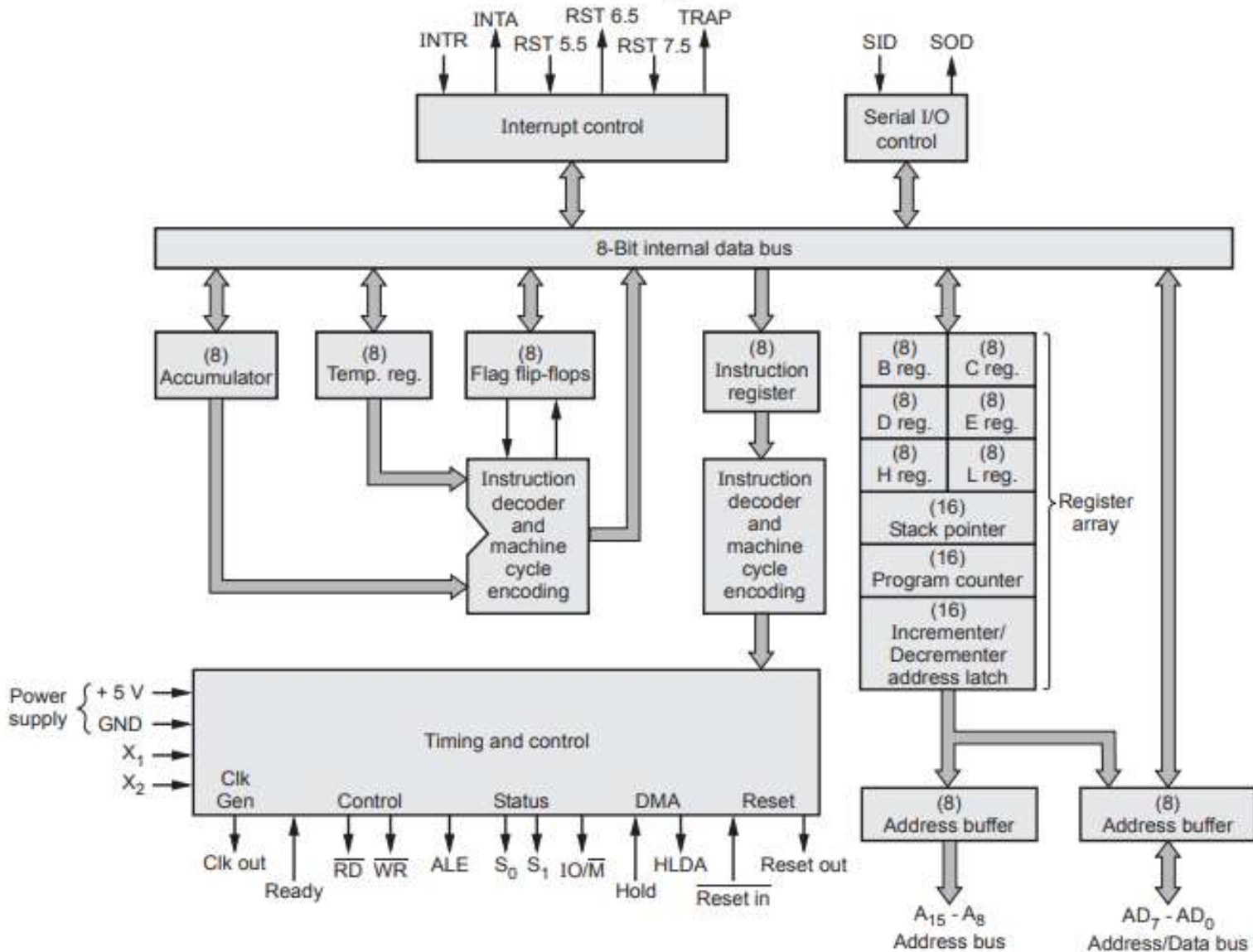


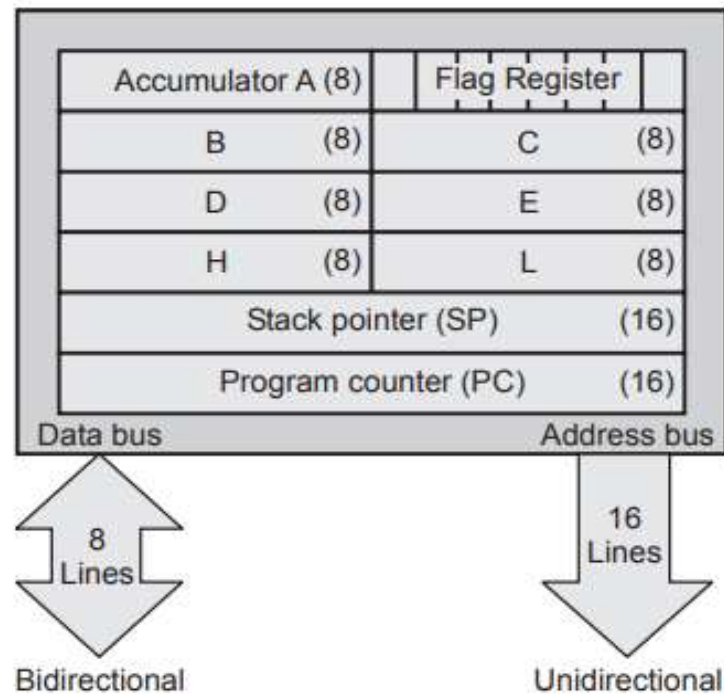
Fig 2.2.1 Architecture of 8085

# Components of Architecture of 8055

1. Registers
2. Accumulator(A)
3. Flags
4. Arithmetic and Logic Units(ALU)
5. Program Counter(PC)
6. Stack Pointer(SP)
7. Timing and Control Unit
8. Interrupt Control Unit

## 1. Registers of 8085 :

- The 8085 have six general - purpose registers to store 8-bit data during program execution.
- These registers are identified as B, C, D, E, H, and L.
- They can be combined as register pairs - BC, DE, and HL - to perform some 16 - bit operations.



## 2. Accumulator (A) :

- The accumulator is an 8 - bit register that is part of the arithmetic/logic unit (ALU).
- This register is used to store 8-bit data and to perform arithmetic and logical operations.
- The result of an operation is stored in the accumulator.

### 3. Flags :

- The ALU includes five flip-flops that are set or reset according to the result of an operation.
- The microprocessor uses the flags for testing the data conditions.
- They are Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags. The most commonly used flags are Sign, Zero and Carry.
- The bit position for the flags in flag register is,

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S	Z		AC		P		CY



### **i . Sign flag (S) :**

- After execution of any arithmetic and logical operation, if  $D_7$  of the result is 1, the sign flag is set.
- Otherwise it is reset.
- $D_7$  is reserved for indicating the sign; the remaining is the magnitude of number.
- If  $D_7$  is 1, the number will be viewed as negative number. If  $D_7$  is 0, the number will be viewed as positive number.

### **ii. Zero flag (z) :**

If the result of arithmetic and logical operation is zero, then zero flag is set, otherwise it is reset.

### **iii. Auxiliary Carry flag (AC) :**

If  $D_3$  generates any carry when doing any arithmetic and logical operation, this flag is set. Otherwise it is reset.


### **iv. Parity flag (P):**

If the result of arithmetic and logical operation contains even number of 1's then this flag will be set and if it is odd number of 1's it will be reset.

### **v. Carry flag (CY) :**


If any arithmetic and logical operation result any carry then carry flag is set, otherwise it is reset.

#### 4. Arithmetic and Logic Unit (ALU) :

 [ AU : Dec.-2016, 2 Marks ]

- It is used to perform the arithmetic operations like addition, subtraction, multiplication, division, increment and decrement and logical operations like AND, OR and EX-OR.
- It receives the data from accumulator and registers.
- According to the result it set or reset the flags.

#### 5. Program Counter (PC) :

 [ AU : Dec.-2016, 2 Marks ]

- This 16-bit register sequencing the execution of instructions.
- It is a memory pointer. Memory locations have 16-bit addresses, and that is why this is a 16-bit register.
- The function of the program counter is to point to the memory address of the next instruction to be executed.
- When an opcode is being fetched, the program counter is incremented by one to point to the next memory location.

## 6. Stack Pointer (SP) :

- The stack pointer is also a 16-bit register used as a memory pointer.
  - It points to a memory location in R/W memory, called the stack.
  - The beginning of the stack is defined by loading a 16-bit address in the stack pointer (register).
- i. **Temporary Register** : It is used to hold the data during the arithmetic and logical operations.
  - ii. **Instruction Register** : When an instruction is fetched from the memory, it is loaded in the instruction register.
  - iii. **Instruction Decoder** : It gets the instruction from the instruction register and decodes the instruction. It identifies the instruction to be performed.
  - iv. **Serial I/O Control** : It has two control signals named SID and SOD for serial data transmission.

## 7. Timing and Control Unit :

- It has three control signals ALE, RD (Active low) and WR (Active low) and three status signals IO/M(Active low),  $S_0$  and  $S_1$ .
- ALE is used for provide control signal to synchronize the components of microprocessor and timing for instruction to perform the operation.
- RD (Active low) and WR (Active low) are used to indicate whether the operation is reading the data from memory or writing the data into memory respectively.
- IO/M (Active low) is used to indicate whether the operation is belongs to the memory or peripherals.
- If,

IO/M (Active Low)	$S_1$	$S_2$	Data bus status(output)
0	0	0	Halt
0	0	1	Memory WRITE
0	1	0	Memory READ
1	0	1	IO WRITE
1	1	0	IO READ
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

## 8. Interrupt Control Unit :

It receives hardware interrupt signals and sends an acknowledgement for receiving the interrupt signal.

## 2.2.1 8085 Bus Structure

### i. Address bus :

- The address bus is a group of 16 lines generally identified as  $A_0$  to  $A_{15}$ .
- The address bus is unidirectional : bits flow in one direction-from the MPU to peripheral devices.
- The MPU uses the address bus to perform the first function : identifying a peripheral or a memory location.

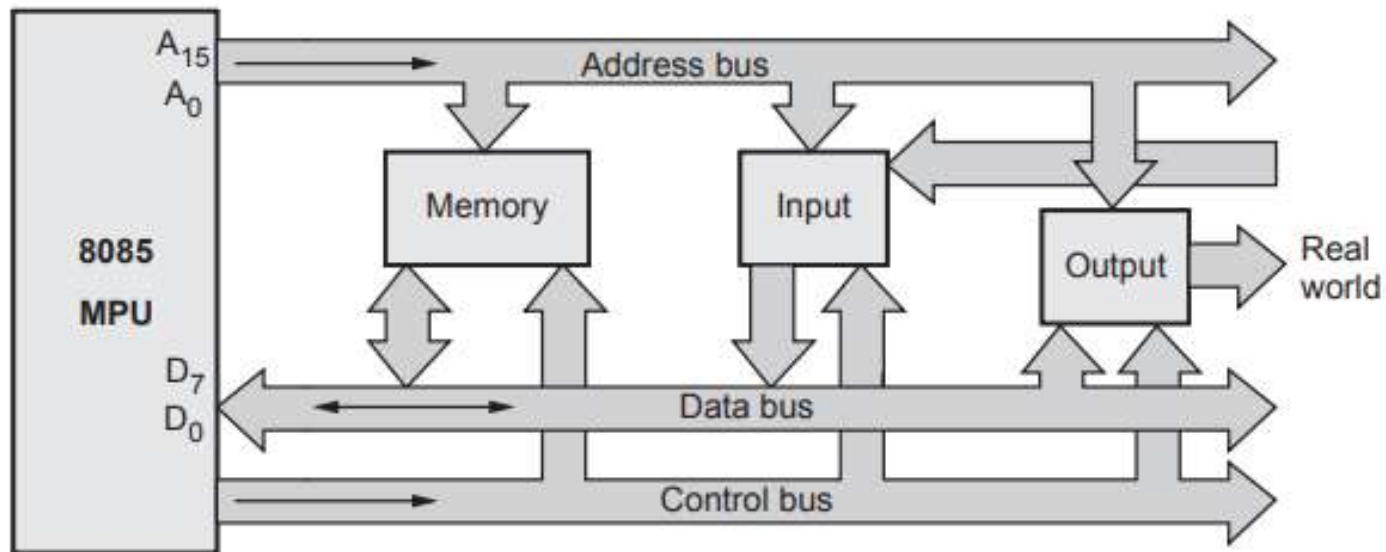


Fig. 2.2.2

## ii. Data bus :

The data bus is a group of eight lines used for data flow.

- These lines are bi-directional - data flow in both directions between the MPU and memory and peripheral devices.
- The MPU uses the data bus to perform the second function: transferring binary information.
- The eight data lines enable the MPU to manipulate 8 - bit data ranging from 00 to FF (2<sup>8</sup> = 256 numbers).
- The largest number that can appear on the data bus is 11111111.

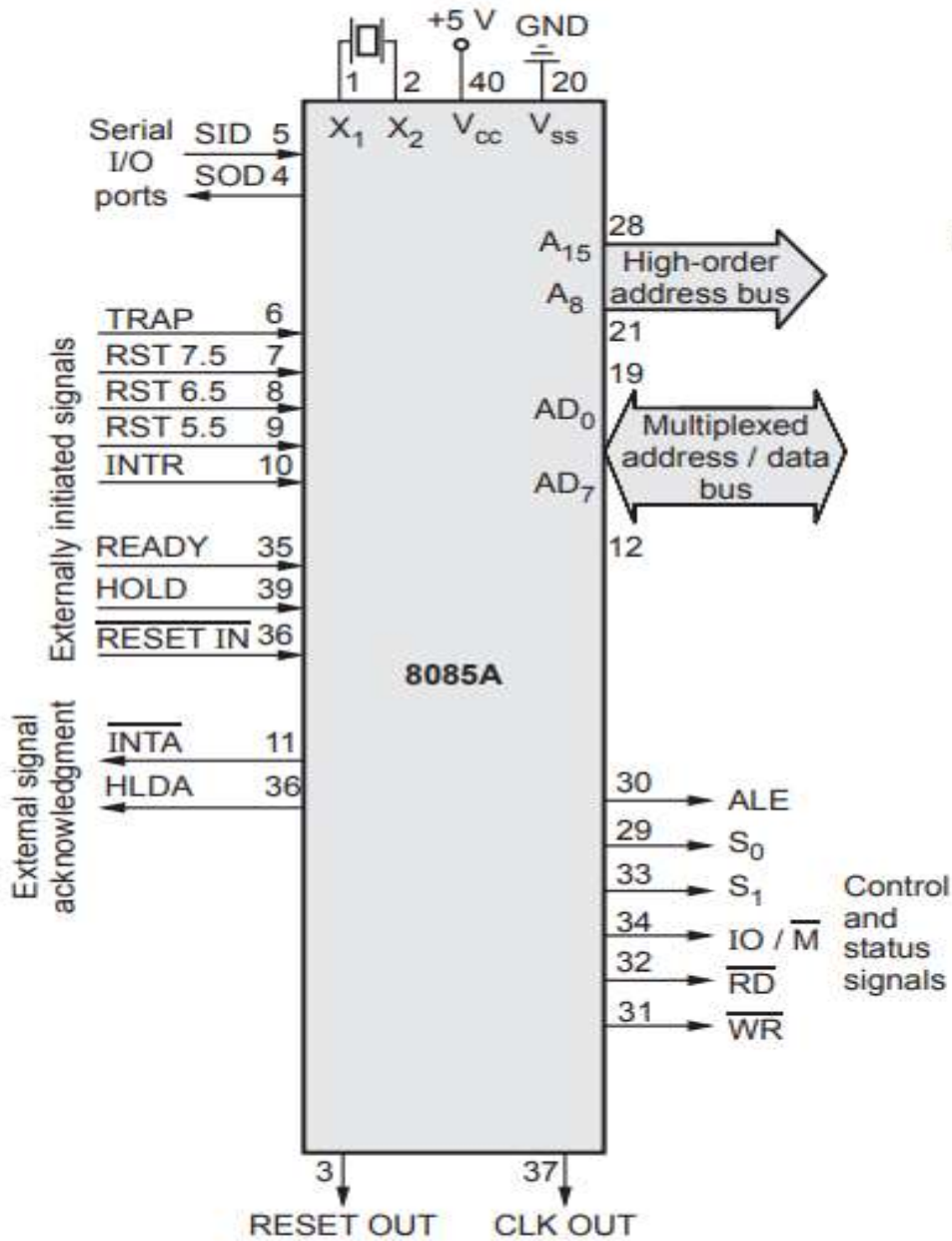
## iii. Control bus :

- The control bus carries synchronization signals and providing timing signals.
- The MPU generates specific control signals for every operation it performs. These signals are used to identify a device type with which the MPU wants to communicate.

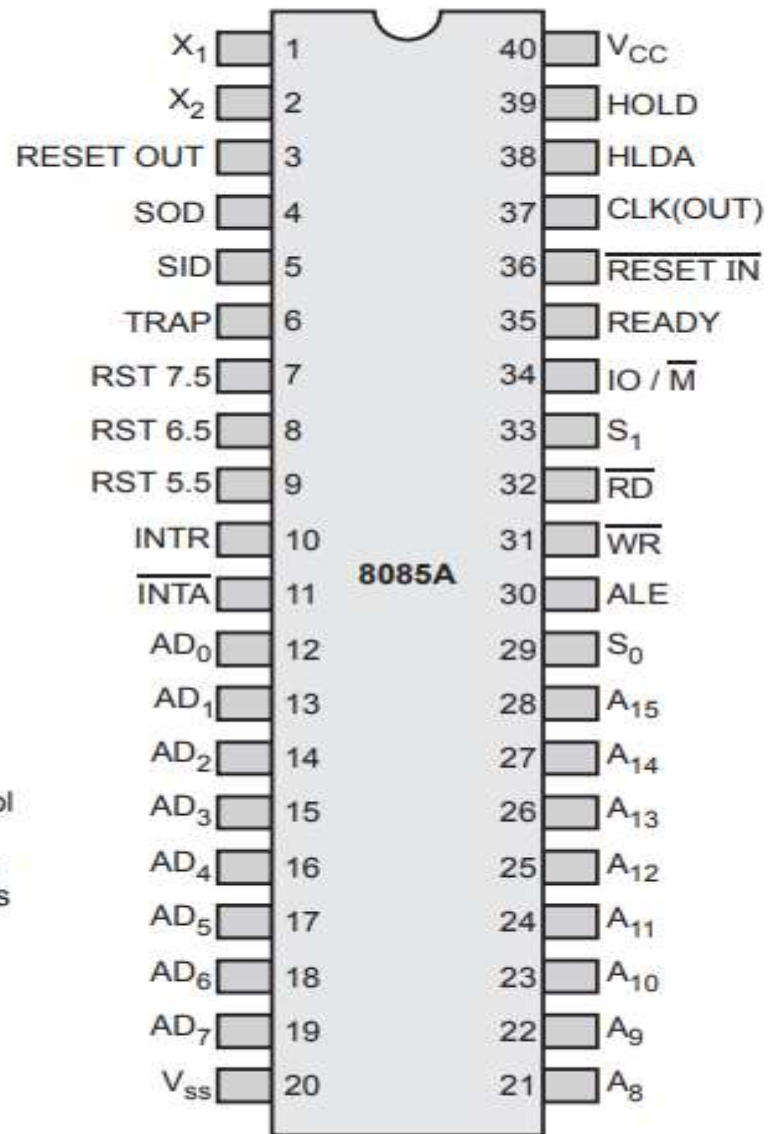
## 2.3 Pin Diagram and Pin Description of 8085

- 8085 is a 40 pin IC, DIP package.
- The signals from the pins can be grouped as follows
  1. Power supply and clock signals
  2. Address bus
  3. Data bus
  4. Control and status signals
  5. Interrupts and externally initiated signals
  6. Serial I/O ports





(a) Functional pin diagram



(b) Pin configuration

## 1. Power supply and clock frequency signals

- $V_{cc}$  + 5 volt power supply
- $V_{ss}$  Ground
- $X_1, X_2$  : Crystal or R/C network or LC network connections to set the frequency of internal clock generator.
- The frequency is internally divided by two. Since the basic operating timing frequency is 3 MHz, a 6 MHz crystal is connected externally.
- CLK (output) - Clock output is used as the system clock for peripheral and devices interfaced with the microprocessor.

## 2. Address Bus :

- $A_8 - A_{15}$  (output; 3 - state)
- It carries the most significant 8 bits of the memory address or the 8 bits of the I/O address.

## 3. Multiplexed Address / Data Bus :

- $AD_0 - AD_7$  (input/output; 3-state) .
- These multiplexed set of lines used to carry the lower order 8 bit address as well as data bus.
- During the opcode fetch operation, in the first clock cycle, the lines deliver the lower order address  $A_0 - A_7$ .
- In the subsequent IO / memory, read / write clock cycle the lines are used as data bus.
- The CPU may read or write out data through these lines.

#### 4. Control and Status signals :

- ALE (output) - Address Latch Enable.
- This signal helps to capture the lower order address presented on the multiplexed address / data bus.
- RD (output 3 - state, active low) - Read memory or IO device.
- This indicates that the selected memory location or I/O device is to be read and that the data bus is ready for accepting data from the memory or I/O device.
- WR (output 3 - state, active low) - Write memory or IO device.
- This indicates that the data on the data bus is to be written into the selected memory location or I/O device.
- IO/M (output) - Select memory or an IO device.
- This status signal indicates that the read / write operation relates to whether the memory or I/O device.
- It goes high to indicate an I/O operation.
- It goes low for memory operations.

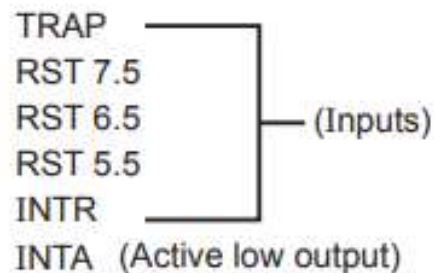
## 5. Status Signals :

It is used to know the type of current operation of the microprocessor.

IO/M (Active low)	$S_1$	$S_2$	Data bus status(output)
0	0	0	Halt
0	0	1	Memory WRITE
0	1	0	Memory READ
1	0	1	IO WRITE
1	1	0	IO READ
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

## 6. Interrupts and externally initiated operations :

- They are the signals initiated by an external device to request the microprocessor to do a particular task or work.
- There are five hardware interrupts called,



- On receipt of an interrupt, the microprocessor acknowledges the interrupt by the active low INTA (Interrupt Acknowledge) signal.

### Reset In (input, active low)

- This signal is used to reset the microprocessor.
- The program counter inside the microprocessor is set to zero.
- The buses are tri - stated.

### Reset Out (output)

- It indicates CPU is being reset.
- Used to reset all the connected devices when the microprocessor is reset

## 7. Direct Memory Access (DMA) :

### Tri state devices :

- 3 output states are high and low states and additionally a high impedance state.
- When enable E is high the gate is enabled and the output Q can be 1 or 0 (if A is 0, Q is 1, otherwise Q is 0). However, when E is low the gate is disabled and the output Q enters into a high impedance state.

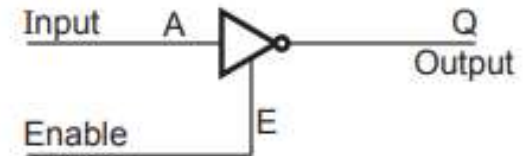


Fig. 2.3.2

E	A	Q	State
1 (high)	0	1	High
1	1	0	Low
0 (low)	0	0	High impedance
0	1	0	High impedance

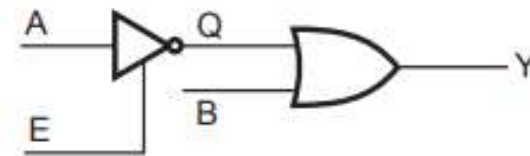



Fig. 2.3.3

For both high and low states, the output Q draws a current from the input of the OR gate.

## 8. Single Bit Serial I/O ports :

- SID (input) - Serial input data line
- SOD (output) - Serial output data line
- These signals are used for serial communication.

## 2.4 8085 Addressing Modes

 [ AU : Dec.-2016, 8 Marks]

- The instructions MOV B, A or MVI A, 82H are to copy data from a source into a destination.
- In these instructions the source can be a register, an input port, or an 8-bit number (00H to FFH).
- Similarly, a destination can be a register or an output port.
- The sources and destination are operands.
- The various formats for specifying operands are called the Addressing Modes.
- For 8085, they are :
  1. Immediate addressing.
  2. Register addressing.
  3. Direct addressing.
  4. Indirect addressing.



### **1. Immediate addressing**

- Data is present in the instruction. Load the immediate data to the destination provided.

**Example :** MVI R, data

### **2. Register addressing**

- Data is provided through the registers.

**Example :** MOV Rd, Rs

### **3. Direct addressing**


- Used to accept data from outside devices to store in the accumulator or send the data stored in the accumulator to the outside device.
- Accept the data from the port 00H and store them into the accumulator or send the data from the accumulator to the port 01H.

**Example :** IN 00H or OUT 01H

### **4. Indirect Addressing**

- This means that the Effective Address is calculated by the processor.
- And the contents of the address (and the one following) is used to form a second address.
- The second address is where the data is stored.
- Note that this requires several memory accesses; two accesses to retrieve the 16-bit address and a further access (or accesses) to retrieve the data which is to be loaded into the register.

## 2.5 Instruction Set of 8085

 [ AU : Dec.-2016, 8 Marks ]

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

### 2.5.1 Classification of Instruction Set

1. Data Transfer Instruction
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Instructions
5. Control Instructions

## 1. Data Transfer Instructions

- These instructions move data between registers, or between memory and registers.
- These instructions copy data from source to destination.
- While copying, the contents of source are not modified.

Opcode	Operand	Description
Copy from source to destination MOV	Rd, Rs M, Rs Rd, M	This instruction copies the contents of the source register into the destination register; the contents of the source register are not altered. If one of the operands is a memory location, its location is specified by the contents of the HL registers. <b>Example :</b> MOV B, C or MOV B, M
Move immediate 8 - bit MVI	Rd, data M, data	The 8 - bit data is stored in the destination register or memory. If the operand is a memory location, its location is specified by the contents of the HL registers. <b>Example :</b> MVI B, 57H or MVI M, 57H
Load accumulator LDA	16 - bit address	The contents of a memory location, specified by a 16-bit address in the operand are copied to the accumulator. The contents of the source are not altered. <b>Example :</b> LDA 2034H
Load accumulator indirect LDAX	B/D Reg. pair	The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the accumulator. The contents of either the register pair or the memory location are not altered. <b>Example :</b> LDAX B

## 2. Arithmetic Instructions

- These instructions perform the operations like :
  1. Addition
  2. Subtract
  3. Increment / Decrement

### 1. Addition

- Any 8-bit number, or the contents of register, or the contents of memory location can be added to the contents of accumulator.
- The result (sum) is stored in the accumulator.
- No two other 8-bit registers can be added directly.
- **Example :** The contents of register B cannot be added directly to the contents of register C.

### 2. Subtraction

- Any 8-bit number, or the contents of register, or the contents of memory location can be subtracted from the contents of accumulator.
- The result is stored in the accumulator.
- Subtraction is performed in 2's complement form.
- If the result is negative, it is stored in 2's complement form.
- No two other 8-bit registers can be subtracted directly.

### 3. Increment / Decrement

- The 8-bit contents of a register or a memory location can be incremented or decremented by 1.
- The 16-bit contents of a register pair can be incremented or decremented by 1.
- Increment or decrement can be performed on any register or a memory location.

Opcode	Operand	Description
Add register or memory to accumulator ADD	R M	The contents of the operand (register or memory) are added to the contents of the accumulator and the result is stored in the accumulator. If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the addition. <b>Example :</b> ADD B or ADD M
Add register to accumulator with carry ADC	R M	The contents of the operand (register or memory) and the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator. If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the addition. <b>Example :</b> ADC B or ADC M
Add immediate to accumulator ADI	8 - bit data	The 8-bit data (operand) is added to the contents of the accumulator and the result is stored in the accumulator. All flags are modified to reflect the result of the addition. <b>Example :</b> ADI 45H

### 3. Logical Instructions

- These instructions perform logical operations on data stored in registers, memory and status flags.
- The logical operations are :
  - AND
  - OR
  - XOR
  - Rotate
  - Compare
  - Complement

#### **AND, OR, XOR**

- Any 8-bit data, or the contents of register, or memory location can logically have AND operation, OR operation, XOR operation with the contents of accumulator.
- The result is stored in accumulator.

#### **Rotate**

- Each bit in the accumulator can be shifted either left or right to the next position.

## Compare

- Any 8-bit data, or the contents of register, or memory location can be compared for : equality, greater than, and less than with the contents of accumulator.
- The result is reflected in status flags.

## Complement

- The contents of accumulator can be complemented.
- Each 0 is replaced by 1 and each 1 is replaced by 0.

Opcode	Operand	Description
Compare register or memory with accumulator CMP	R M	The contents of the operand (register or memory) are compared with the contents of the accumulator. Both contents are preserved. The result of the comparison is shown by setting the flags of the PSW as follows : if (A) < (reg/mem) : carry flag is set if (A) = (reg/ mem) : zero flag is set if (A) > (reg/mem) : carry and zero flags are reset <b>Example :</b> CMP B or CMP M
Compare immediate with accumulator CPI	8 - bit data	The second byte (8-bit data) is compared with the contents of the accumulator. The values being compared remain unchanged. The result of the comparison is shown by setting the flags of the PSW as follows : if (A) < data : carry flag is set if (A) = data : zero flag is set if (A) > data : carry and zero flags are reset <b>Example :</b> CPI 89H
Logical AND register or memory with accumulator ANA	R M	The contents of the accumulator are logically ANDed with the contents of the operand (register or memory) and the result is placed in the accumulator. If the operand is a memory location, its address is specified by the contents of HL registers. S, Z, P are modified to reflect the result of the operation. CY is reset. AC is set. <b>Example :</b> ANA B or ANA M



#### 4. Branching Instructions

- The branching instruction alters the normal sequential flow.
- These instructions alter either unconditionally or conditionally.

Opcode	Operand	Description
Jump unconditionally JMP	16 - bit address	The program sequence is transferred to the memory location specified by the 16 - bit address given in the operand. <b>Example :</b> JMP 2034H or JMP XYZ
Jump conditionally Operand :	16 - bit address	The program sequence is transferred to the memory location specified by the 16-bit address given in the operand based on the specified flag of the PSW as described below. <b>Example :</b> JZ 2034H or JZ XYZ
Opcode	Description	Flag status
JC	Jump on Carry	CY = 1
JNC	Jump on no Carry	CY = 0
JP	Jump on positive	S = 0
JM	Jump on minus	S = 1
JZ	Jump on zero	Z = 1
JNZ	Jump on no zero	Z = 0
JPE	Jump on parity even	P = 1
JPO	Jump on parity odd	P = 0

## 5. Control Instructions

- The control instructions control the operation of microprocessor.

Opcode	Operand	Description
No operation NOP	none	No operation is performed. The instruction is fetched and decoded. However no operation is executed. <b>Example :</b> NOP
Halt and enter wait state HLT	none	The CPU finishes executing the current instruction and halts any further execution. An interrupt or reset is necessary to exit from the halt state. <b>Example :</b> HLT
Disable interrupt DI	none	The interrupt enable flip - flop is reset and all the interrupts except the TRAP are disabled. No flags are affected. <b>Example :</b> DI
Enable interrupt EI	none	The interrupt enable flip - flop is set and all interrupts are enabled. No flags are affected. After a system reset or the acknowledgement of an interrupt, the interrupt enable flip - flop is reset, thus disabling the interrupts. This instruction is necessary to reenables the interrupts (except TRAP). <b>Example :</b> EI

# Timing diagram of 8085

## 2.6 Timing Diagram of 8085

- Timing Diagram is a graphical representation.
- The timing diagram is the diagram which provides information about the various conditions of signals such as high/low, when a machine cycle is being executed.
- Without the knowledge of timing diagram it is not possible to match the peripheral devices to the microprocessors.
- These peripheral devices includes memories, ports etc. Such devices can only be matched with microprocessors with the help of timing diagram.
- 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**

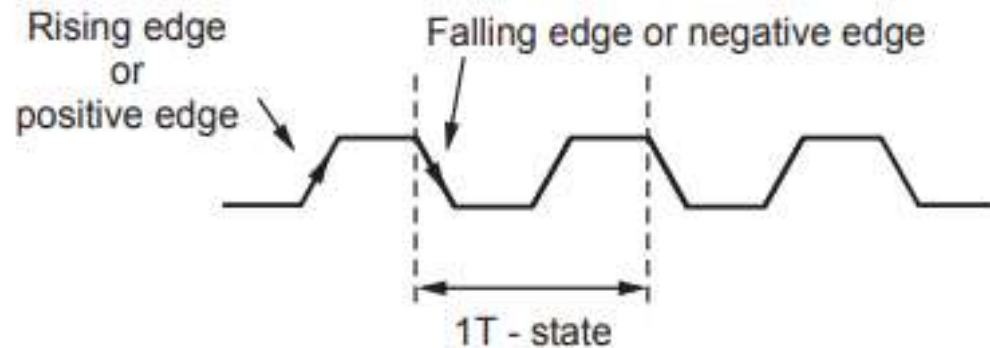
- The time required to access the memory or input/output devices is called machine cycle.

## **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.

## 2.6.1 Machine cycles of 8085

- The 8085 microprocessor has 5 (seven) basic machine cycles. They are
  1. Opcode fetch cycle (4T)
  2. Memory read cycle (3T)
  3. Memory write cycle (3T)
  4. I/O read cycle (3T)
  5. I/O write cycle (3T)
- Time period,  $T = 1/f$ ; where  $f =$  Internal clock frequency



**Fig. 2.6.1 Clock signal**

## 1. Opcode fetch machine cycle of 8085 :

- 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, 3 T - states are used for fetching the opcode from memory and the remaining T - states are used for internal operations by the processor.

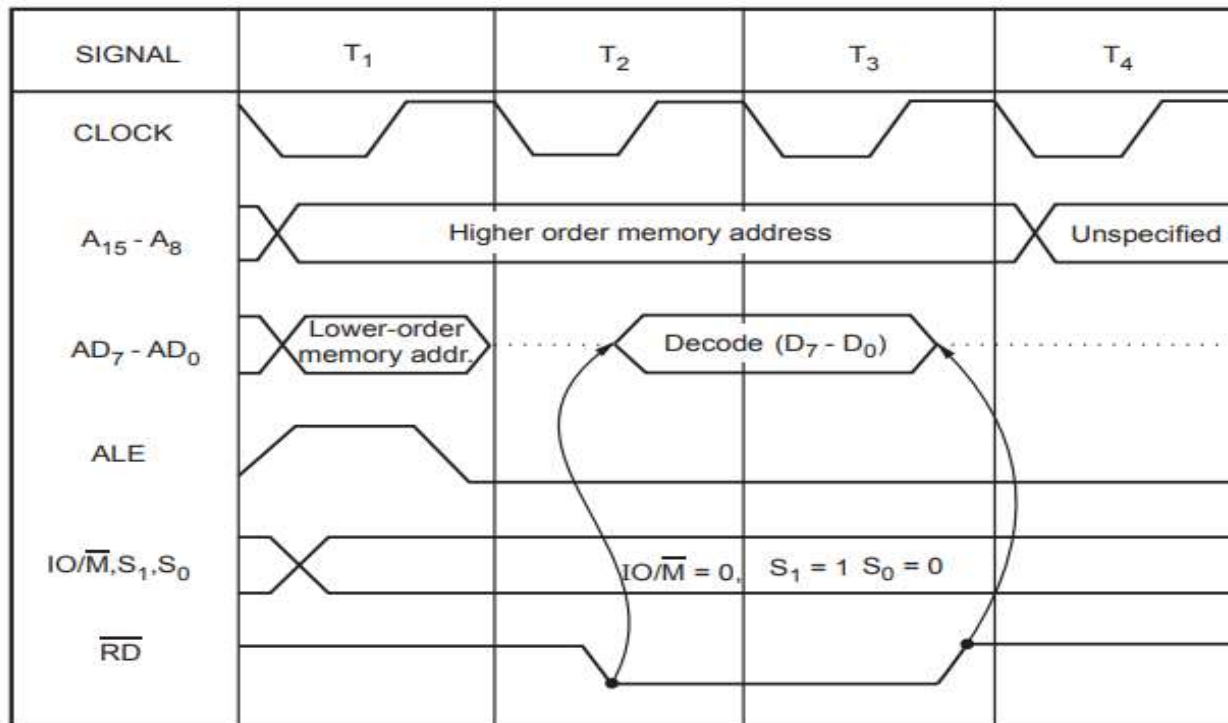


Fig. 2.6.2 Timing diagram of opcode fetch machine cycle

## 2. Memory Read Machine Cycle of 8085

- 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. (See Fig. 2.6.3 on next page.)

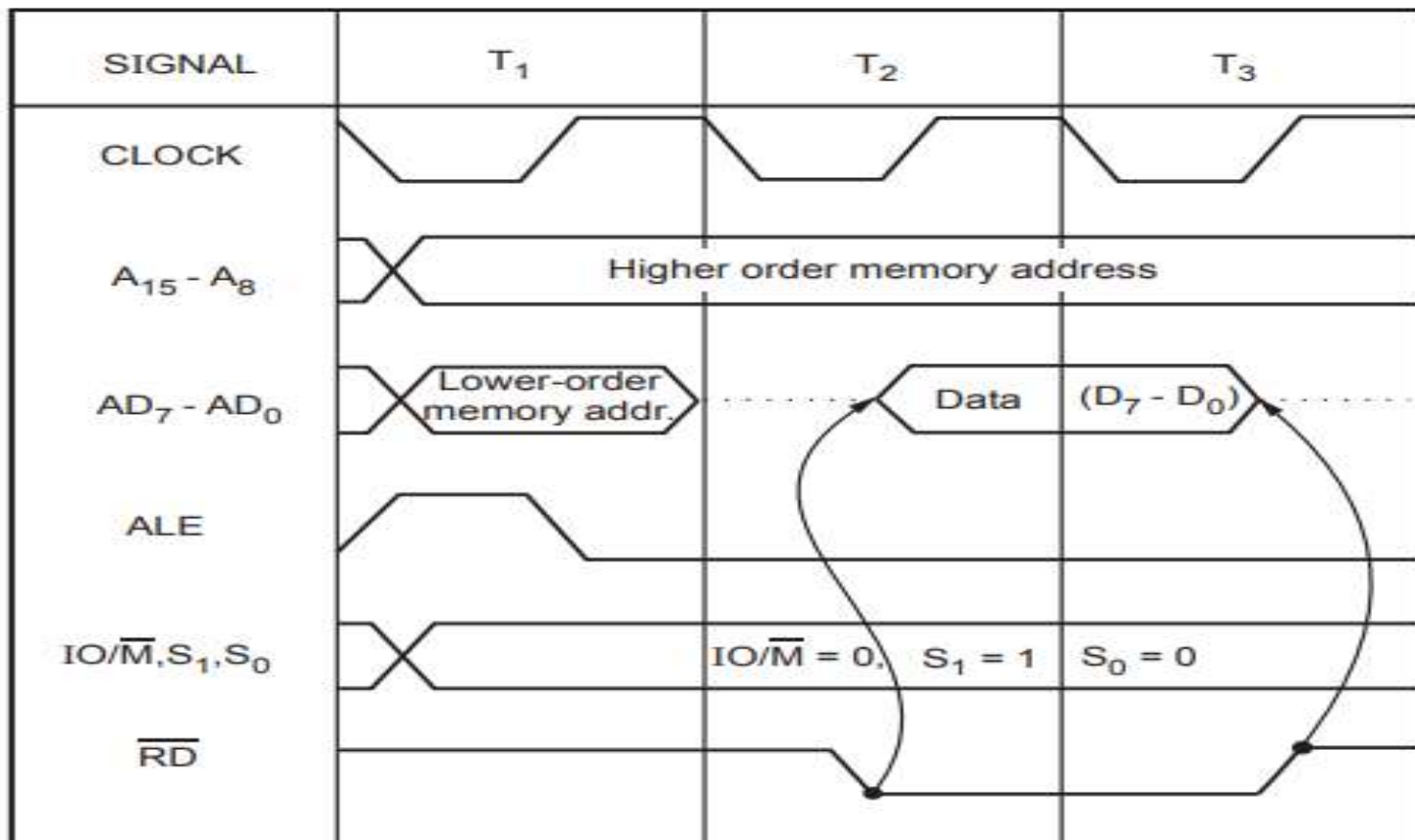


Fig. 2.6.3 Memory Read Machine Cycle

### 3. Memory Write Machine Cycle of 8085

- 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.  
(See Fig. 2.6.4 on next page.)

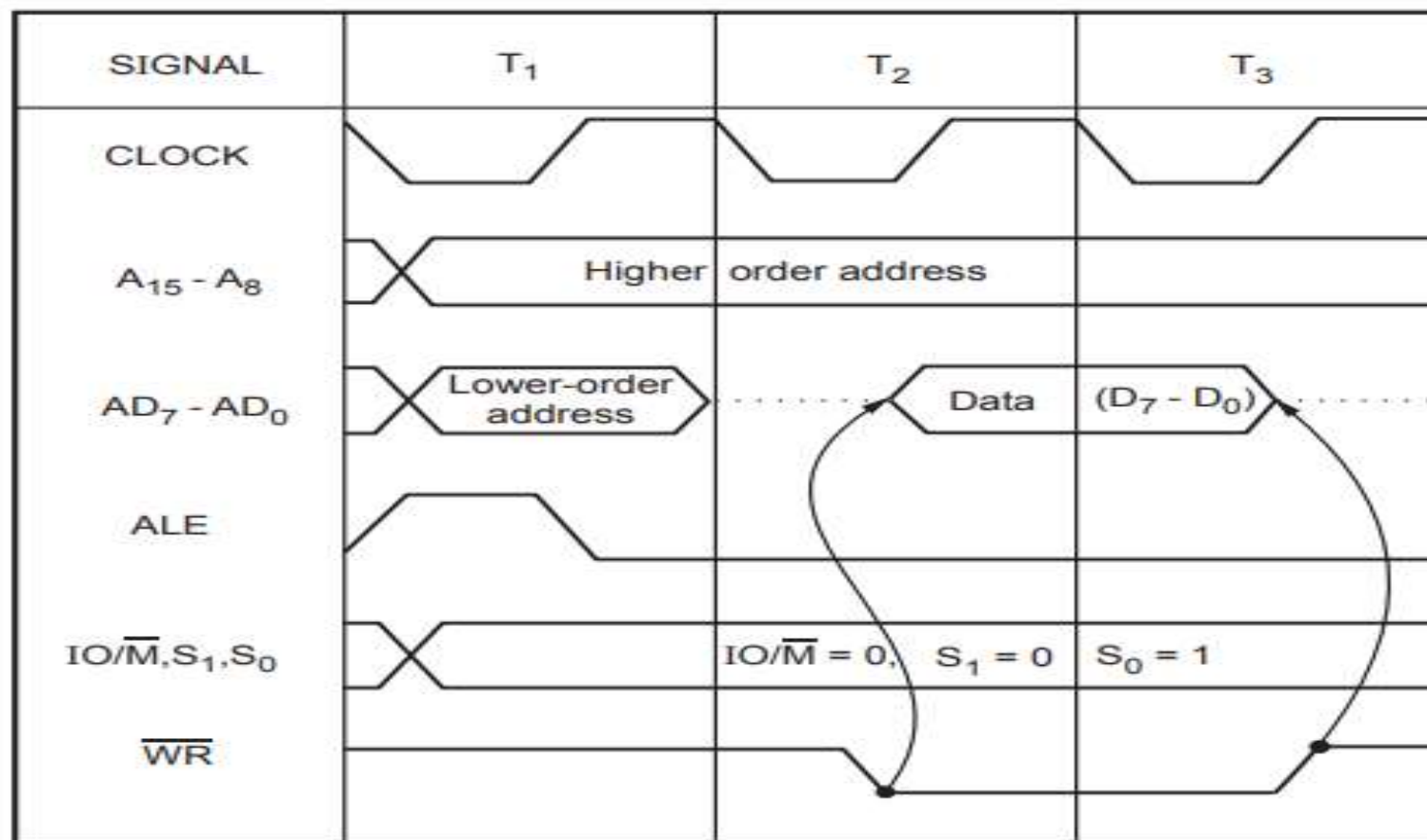


Fig. 2.6.4 Memory Write Machine Cycle



#### 4. 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, 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.

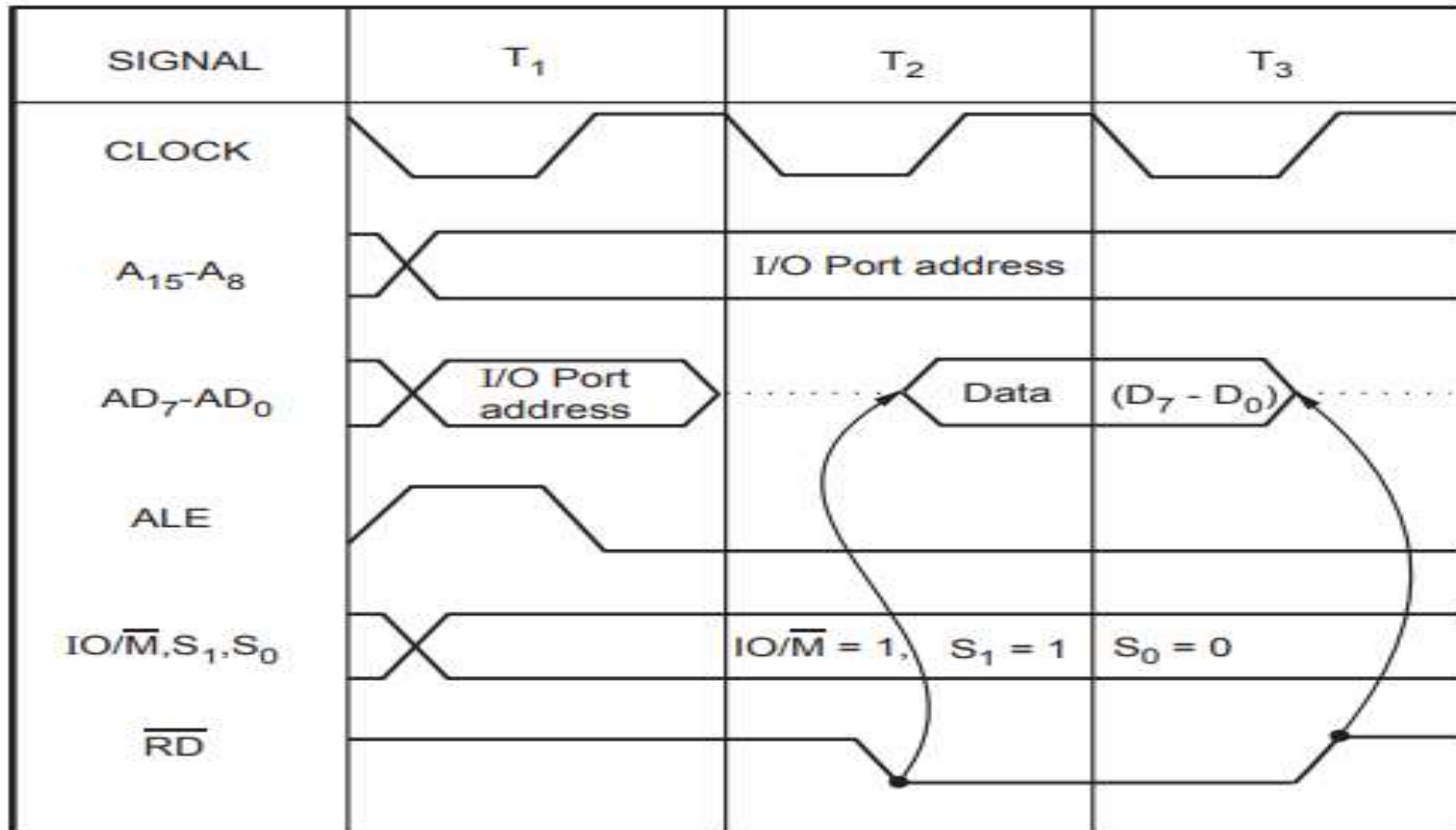
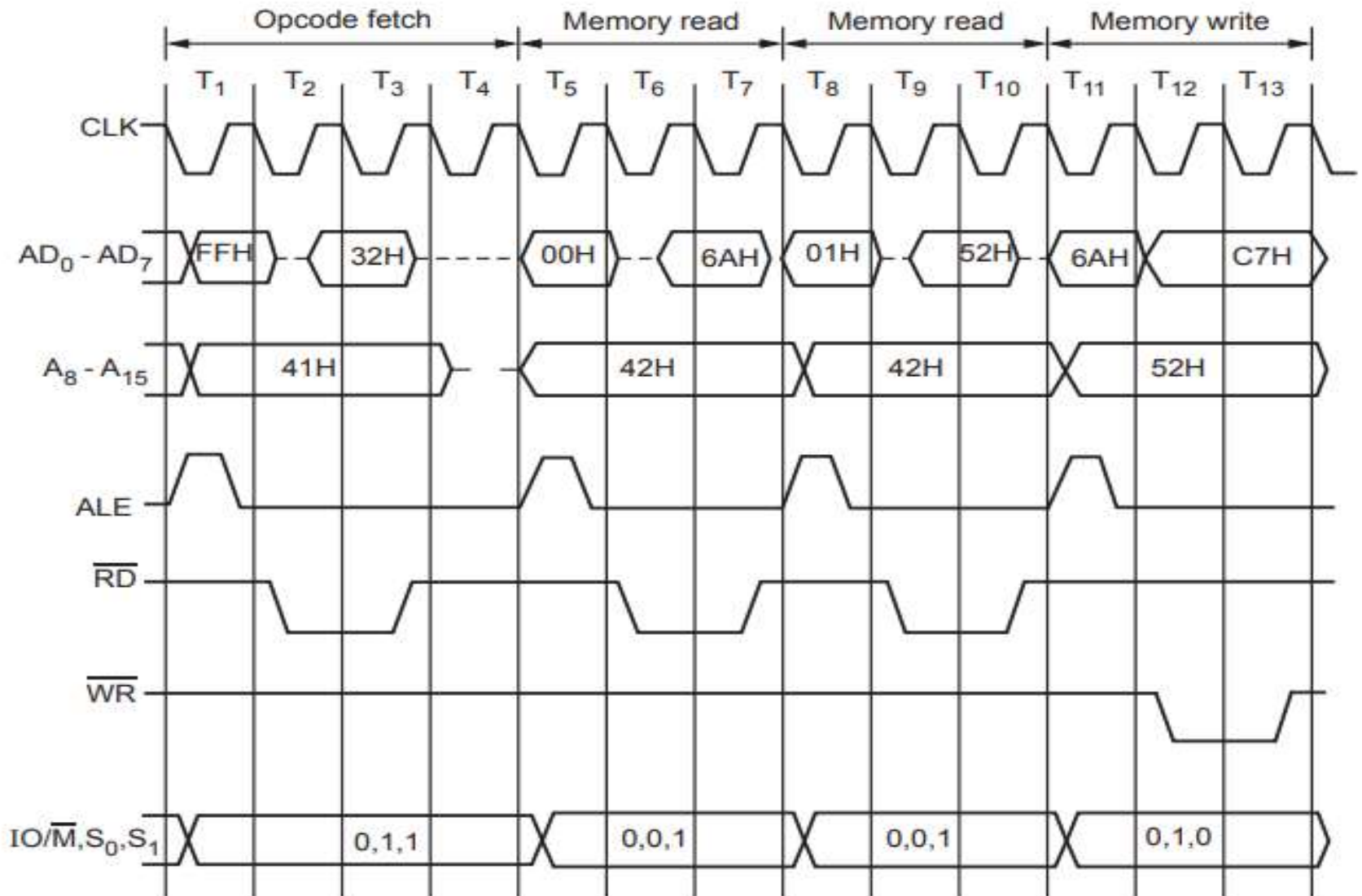


Fig. 2.6.5 I/O Read Cycle

## 2.6.2 Timing Diagram for STA 526AH

- STA means Store Accumulator - The contents of the accumulator is stored in the specified address (526A).
- The opcode of the STA instruction is said to be 32H. It is fetched from the memory 41FFH (see Fig. 2.6.6). - of machine cycle
- Then the lower order memory address is read (6A) - Memory Read Machine Cycle
- Read the higher order memory address (52) - Memory Read Machine Cycle
- The combination of both the addresses are considered and the content from accumulator is written in 526A - Memory Write Machine Cycle
- Assume the memory address for the instruction and let the content of accumulator is C7H. So, C7H from accumulator is now stored in 526A.

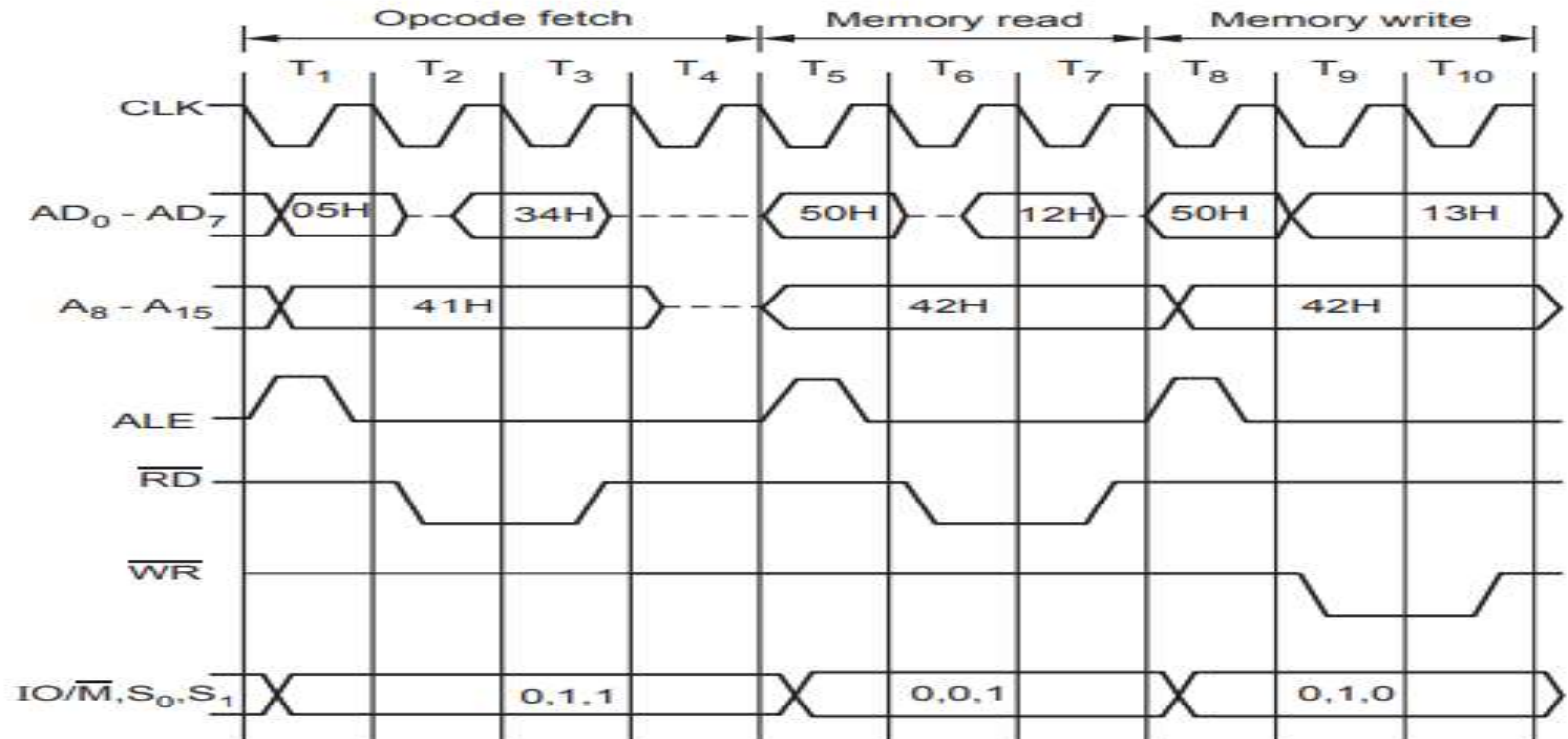


**Fig. 2.6.6 Timing Diagram for STA 526A H**

### 2.6.3 Timing Diagram for INR M

- Fetching the Opcode 34H from the memory 4105H (of cycle).
- Let the memory address (M) be 4250H. (MR cycle - to read Memory address and data).
- Let the content of that memory is 12H.
- Increment the memory content from 12H to 13H. (MW machine cycle)

Address	Mnemonics	Opcode
4105	INR M	34H




# Microcontroller

## 2.7.2.2 Microcontroller

- A microcontroller is a highly integrated single chip, which consists of on chip CPU (Central Processing Unit), RAM (Random Access Memory), EPROM/PROM/ROM (Erasable Programmable Read Only Memory), I/O (input/output) - serial and parallel, timers, interrupt controller.
- For example, Intel 8051 is 8-bit microcontroller and Intel 8096 is 16-bit microcontroller.

### 2.7.3 Distinguish between Microprocessor and Microcontroller

 [ AU : Dec.-2016, 2 Marks ]

Sr. No.	Microprocessor	Microcontroller
1	A microprocessor is a general purpose device which is called a CPU.	A microcontroller is a dedicated chip which is also called single chip computer.
2	A microprocessor do not contain onchip I/O ports, timers, memories etc.	A microcontroller includes RAM, ROM, serial and parallel interface, timers, interrupt circuitry (in addition to CPU) in a single chip.
3	Microprocessors are most commonly used as the CPU in microcomputer systems.	Microcontrollers are used in small, minimum component designs performing control - oriented applications.
4	Microprocessor instructions are mainly nibble or byte addressable.	Microcontroller instructions are both bit addressable as well as byte addressable.
5	Microprocessor instruction sets are mainly intended for catering to large volumes of data.	Microcontrollers have instruction sets catering to the control of inputs and outputs.
6	Microprocessor based system design is complex and expensive.	Microcontroller based system design is rather simple and cost effective.
7	The instruction set of microprocessor is complex with large number of instructions.	The instruction set of a microcontroller is very simple with less number of instructions. For ex : PIC microcontrollers have only 35 instructions.
8	A microprocessor has zero status flag.	A microcontroller has no zero flag.

## 2.7.4 Feature of 8051 Microcontroller

- Features of 8051 microcontroller are given as follows.
  1. 8-bit CPU
  2. 16-bit Program Counter
  3. 8-bit Processor Status Word (PSW)
  4. 8-bit Stack Pointer
  5. Internal RAM of 128 bytes
  6. Special Function Registers (SFRs) of 128 bytes
  7. 32 I/O pins arranged as four 8-bit ports (P0 - P3)
  8. Two 16-bit timer/counters : T0 and T1
  9. Two external and three internal vectored interrupts
  10. One full duplex serial I/O

## 2.8 Architecture & Block Diagram of 8051 Microcontroller

The architecture of the 8051 microcontroller can be understood from the block diagram. It has Harward architecture with RISC (Reduced Instruction Set Computer) concept. The block diagram of 8051 microcontroller is shown in Fig 2.8.1. It consists of an 8-bit ALU, one 8-bit PSW (Program Status Register), A and B registers, one 16-bit program counter , one 16-bit Data pointer register (DPTR), 128 bytes of RAM and 4kB of ROM and four parallel I/O ports each of 8-bit width.

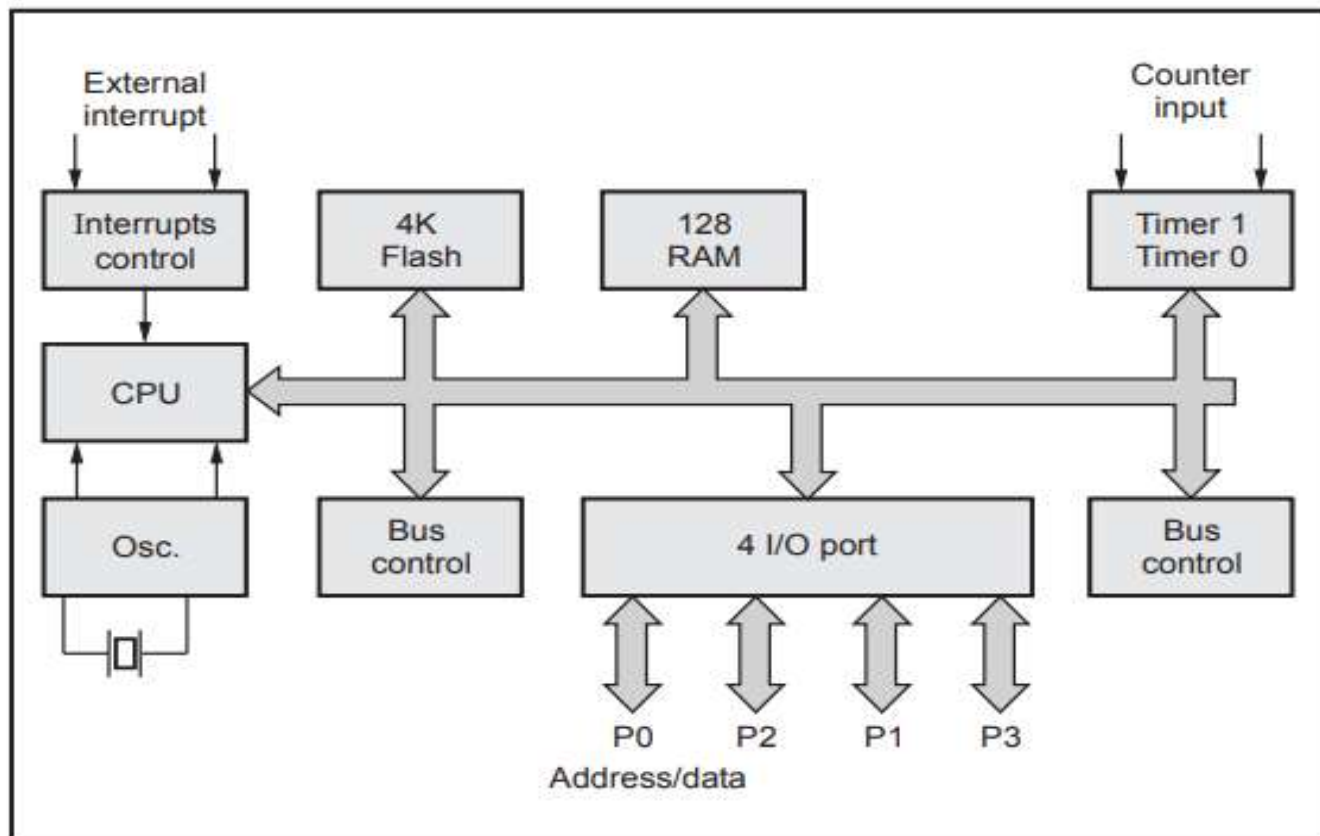


Fig. 2.8.1 Block diagram of 8051 microcontroller



## 1. Central Processor Unit (CPU)

- CPU is the brain of any processing device of the microcontroller.
- It monitors and controls all operations that are performed on the microcontroller units.
- The user has no control over the work of the CPU directly.
- It reads program written in ROM memory and executes them and do the expected task of that application.

## 2. Interrupts

- As its name suggests, interrupt is a subroutine call that interrupts of the microcontrollers main operations or work and causes it to execute any other program, which is more important at the time of operation.
- The feature of interrupt is very useful as it helps in case of emergency operations.
- An Interrupts gives us a mechanism to put on hold the ongoing operations, execute a subroutine and then again resumes to another type of operations.

### 3. Memory

- Microcontroller requires a program which is a collection of instructions.
- This program tells microcontroller to do specific tasks.
- These programs require a memory on which these can be saved and read by microcontroller to perform specific operations of a particular task.
- The memory which is used to store the program of the microcontroller is known as code memory or program memory of applications.
- It is known as ROM memory of microcontroller also requires a memory to store data or operands temporarily of the micro controller.
- The data memory of the 8051 is used to store data temporarily for operation is known RAM memory.
- 8051 microcontroller has 4K of code memory or program memory, that has 4kB ROM and also 128 bytes of data memory of RAM.

#### **4. Bus**

- Basically bus is a collection of wires which work as a communication channel or medium for transfer of data.
- These buses consist of 8, 16 or more wires of the microcontroller.
- Thus, these can carry 8 bits, 16 bits simultaneously.
- There are two types of buses are
  - i. Address Bus
  - ii. Data Bus

##### **i. Address Bus :**

- Microcontroller 8051 has a 16 bit address bus for transferring the data.
- It is used to address memory locations and to transfer the address from CPU to Memory of the microcontroller.
- It has four addressing modes that are
  - a. Immediate addressing modes.
  - b. Bank address (or) Register addressing mode.
  - c. Direct addressing mode.
  - d. Register indirect addressing mode.

##### **ii. Data Bus :**

- Microcontroller 8051 has 8 bits of the data bus, which is used to carry data of particular applications.

## **5. Oscillator**

- Generally, we know that the microcontroller is a device, therefore it requires clock pulses for its operation of microcontroller applications.
- For this purpose, microcontroller 8051 has an on-chip oscillator which works as a clock source for Central Processing Unit of the microcontroller.
- The output pulses of oscillator are stable.
- Therefore, it enables synchronized work of all parts of the 8051 microcontroller.

## **6. Input/Output Port**

- Normally microcontroller is used in embedded systems to control the operation of machines in the microcontroller.
- Therefore, to connect it to other machines, devices or peripherals we require I/O interfacing ports in the microcontroller interface.
- For this purpose microcontroller 8051 has 4 input, output ports to connect it to the other peripherals

## **7. Timers/Counters**

- 8051 microcontroller has two 16 bit timers and counters.
- These counters are again divided into a 8 bit register.
- The timers are used for measurement of intervals to determine the pulse width of pulses.

## 2.9 PIN Diagram of 8051 Microcontroller

- The 8051 microcontroller is available as a 40 pin DIP chip and it works at +5 volts DC.
- Among the 40 pins, a total of 32 pins are allotted for the four parallel ports P0, P1, P2 and P3 i.e each port occupies 8-pins .
- The remaining pins are VCC, GND, XTAL1, XTAL2, RST, EA, PSEN.

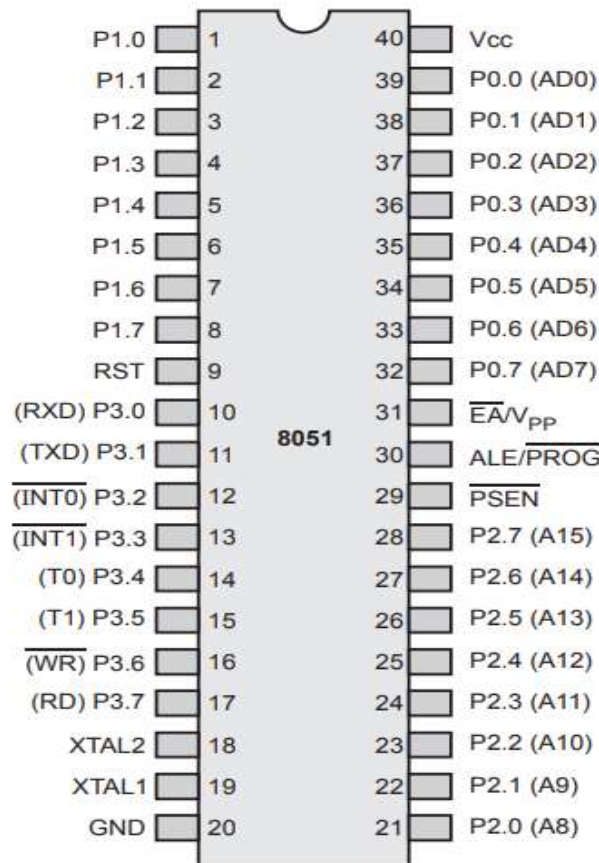


Fig. 2.9.1 PIN diagram of 8051 microcontroller

## 1. XTAL1, XTAL2 :

- These two pins are connected to quartz crystal oscillator which runs the on-chip oscillator.
- The quartz crystal oscillator is connected to the two pins along with a capacitor of 30 pF as shown in the circuit.

## 2. RST :

- The RESET pin is an input pin and it is an active high pin.
- When a high pulse is applied to this pin the microcontroller will reset and terminate all activities.
- Upon reset all the registers except PC will reset to 0000 value and PC register will reset to 0007 value.

## 3. $\overline{\text{EA}}$ (External Access) :

- This pin is an active low pin.
- This pin is connected to ground when microcontroller is accessing the program code stored in the external memory and connected to Vcc when it is accessing the program code in the on chip memory.
- This pin should not be left unconnected.

## 4. $\overline{\text{PSEN}}$ (Program Store Enable) :

- This is an output pin which is active low.
- When the microcontroller is accessing the program code stored in the external ROM, this pin is connected to the OE (Output Enable) pin of the ROM.

## 5. ALE (Address Latch Enable) :

- This is an output pin, which is active high.
- When connected to external memory, port 0 provides both address and data i.e address and data are multiplexed through port 0 .
- This ALE pin will demultiplex the address and data bus.
- When the pin is high, the AD bus will act as address bus otherwise the AD bus will act as data bus.

## 6. P0.0 - P0.7 (AD<sub>0</sub> - AD<sub>7</sub>) :

- The port 0 pins multiplexed with address/data pins.
- If the microcontroller is accessing external memory these pins will act as address/data pins otherwise they are used for port 0 pins.

## 7. P2.0 - P2.7 (A<sub>8</sub> - A<sub>15</sub>) :

- The port 2 pins are multiplexed with the higher order address pins.
- When the microcontroller is accessing external memory these pins provide the higher order address byte otherwise they act as port 2 pins.

## 8. P1.0 - P1.7 :

- These 8-pins are dedicated for port 1 to perform input or output port operations.

## 9. P3.0 - P3.7 :

- These 8-pins are meant for port 3 operations and also for some control operations like read, write, timer0, timer1, INT0, INT1, RxD and TxD
-

## 2.11 Applications of 8051 Microcontroller

- Some of the applications of 8051 is mainly used in daily life & industrial applications also some of that applications are shown below
  - i. Light sensing and controlling devices
  - ii. Temperature sensing and controlling devices
  - iii. Fire detections and safety devices
  - iv. Automobile applications
  - v. Defense applications
- Some industrial applications of microcontroller and its applications
  - i. Industrial instrumentation devices
  - ii. Process control devices