# MODULE 1

## Microprocessor

A microprocessor is a multipurpose,programmable logic device that reads binary instructions from memory,accepts binary data as input and processes data according to those instructions,and provide result as output.

The microprocessor can be broadly divided into:

1.ALU

2.Registers

3.Control unit

## Evolution of microprocessors

First generation:

Intel 4004-4 bit-1971->Designed specifically for calculator

Intel 8008-8 bit-1972

Second generation

Intel 8085-8 bit-1976

Development of microprocessor has been towards a complete microcomputer system with CPU,ROM,RAM all in single package.

Ex:Intel 8048

Third generation

Intel 8086-16 bit-1976

Ex:Intel 8086,8088,Zylog,M6800

Fourth generation

Intel 80386-32 bit-1981

Fifth generation

Pentium I,Pentium II,Pentium III&IV,Intel core 2 Duo,Intel Xeon etc.

## Introduction of 8085 microprocessor

The microprocessor is the central processing unit or cpu of a micro computer.it is the heart of the computer.

## 3.1. Intel 8085:

It is an 8 bit NMOS microprocessor.it is an 40 pin IC(integrated circuit) package fabricated on a single LSI (Large scale Integration) chip. It uses a single +5 volt d.c.(Direct Current) supply for its operation. It's clock speed is 3 MHZ and clock cycle is 320 ns.

## Architecture of Intel 8085

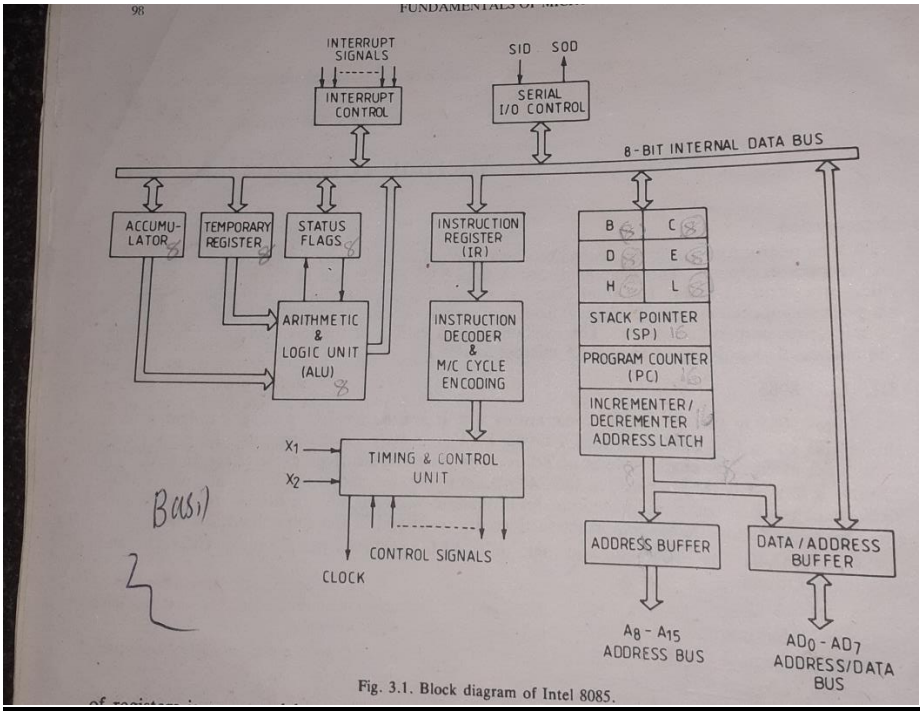



Fig. 3.1. Block diagram of Intel 8085.

It consists of 3 main sections.

1.Arithmetic Logic Unit(ALU)

2.Timing and control unit

3.Registers

### 3.1.1.Arithmetic Logic Unit:

It performs various arithmetic and logical operations like:

I. Addition
II. Subtraction
III. Logical AND
IV. Logical OR
V. Logical XOR
VI. Complement
VII. Increment
VIII. Decrement
IX. Left shift
X. Clear

**3.1.2.Timing and control unit:**

It generates timing and control signals which are necessary for the execution of the instructions.it controls the data flow between cpu and peripherals.

It controls the entire operation of microprocessor. It acts as brain of the computer system

**3.1.3. Registers**

Registers are used by the microprocessor for temporary storage and manipulation of data and instructions. Data remain in the registers till they are sent to the memory or I/O devices

Intel 8085 microprocessor has the following registers:

I. One 8 bit accumulator i.e. register A
II. six 8 bit general purpose registers i.e. B,C,D,E,H,L
III. one 16 bit stack pointer ,SP
IV. one 16 bit Program counter, PC
V. Instruction Register.
VI. Status register
VII. Temporary register

**Accumulator**

-8 bit register

-It is also referred as Register A

-The Register A holds the first operand during program execution.

-Final result of arithmetic or logical operation is placed in the accumulator

**General purpose Register**

-six 8 bit general purpose registers are B,C,D,E,H,L

-To hold 16 bit data, two 8 bit registers can be combined. This is called register pair.

-valid register pairs are B-C, D-E, H-L

-H-L pair used to address memory location

**PROGRAM COUNTER:**

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

**STACK POINTER:**

-It is a 16 bit special purpose register

-SP hold the address of stack Top.

-Stack is a sequence of memory location defined by the programmer

-Stack is used to save the content of a register during execution of a program

-The last memory location of the occupied portion of the stack is called stack Top.

**Instruction Register**

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

**Temporary Register**

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

It is not accessible to programmer

**Status flags**

-it contains the status of the current result

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

These are the set of 5 flip-flops −

- Sign (S)
- Zero (Z)
- Auxiliary Carry (AC)
- Parity (P)
- Carry (CS)

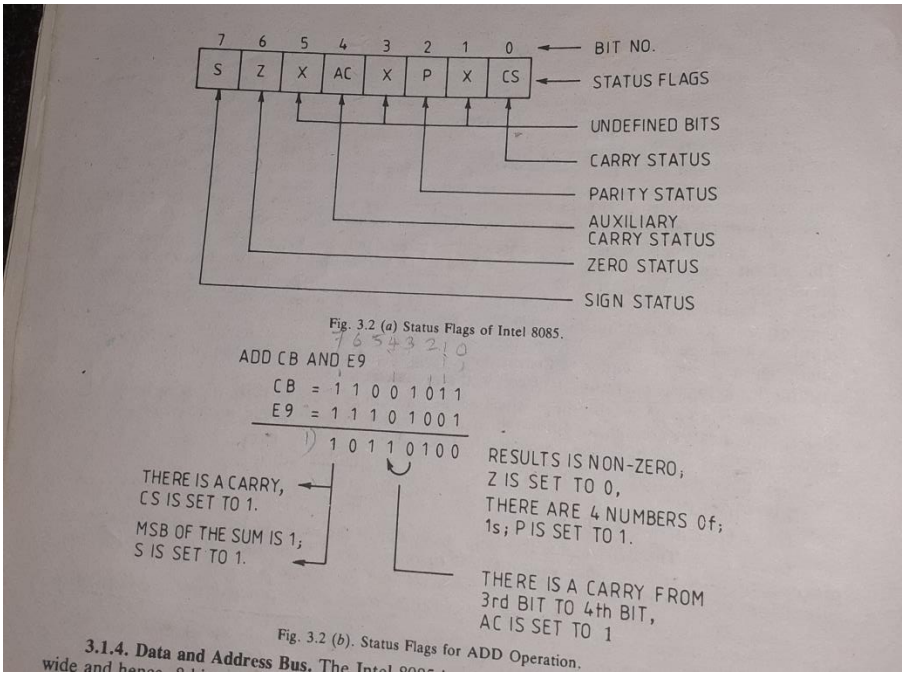



Fig. 3.2 (*a*) Status Flags of Intel 8085.




Fig. 3.2 (*b*). Status Flags for ADD Operation.

**3.1.4. Data and Address Bus.** The Intel 8085

**Carry Flag:**

After the execution of an arithmetic operation, if a carry is produced then the carry flag CS is set to 1, otherwise 0

**Sign Flag:**

It is set to 1 if the MSB(Most Significant Bit) of the result of an arithmetic or logical operation is 1, otherwise it is 0.

**Zero Flag:**

It is set to 1 if the result of an arithmetic or logical operation is zero. for non zero result,it is 0.

**Parity Flag:**

It is set to 1 when the result of the operation contains even no. of 1& it is set to 0 if there are odd no.of 1.

**Auxilary Carry Flag:**

If there is a carry from bit 3 to 4,the AC flag is set to 1 otherwise it is 0.

**Program Status Word(PSW):**

It is a combination of 8-bits where five bits indicates the 5 status flags & three bits are undefined.

Ex 11001011+

11101001

1) 10110100

Cs=1
P=1
AC=1
Z=0
S=1

**Data and Address Bus**

- Address bus is 16 bit wide as memory address are of 16 bit.
- 8 MSB(Most significant Bit) of the address are transmitted by Address Bus or A-bus [ A8-A15].
- 8 LSB(Least significant Bit) of the address are transmitted by the Address/data bus,or AD-bus [AD0-AD7].
- The address or data bus transmits data & address at different moments.it can transmit data or address at a time.
- AD-bus operates in time shared mode. This technique is known as multiplexing.

**********

**Additional Explanation for 8085 Architecture: (Not necessary..page no 7,8,9)**

To execute a program:

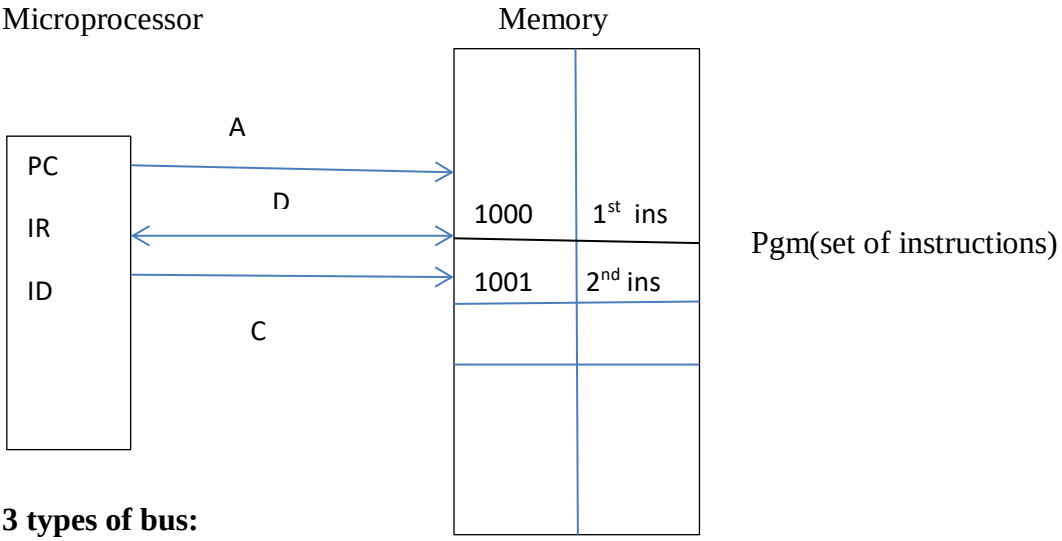
1.Fetch instructions from memory

2.Decoding(understand the binary pattern)

3.Execution(ex:ADD/SUBTRACT)

Register-inside the microprocessor

Memory-outside microprocessor




**3 types of bus:**

**A-address bus**

**D-data bus**

**C-Control bus**

To fetch instructions, we need address which is stored in memory in different address.Suppose now PC contain the address 1000(Program Counter Contain address of the next instruction),then this 16 bit address is sent to memory through address bus and it split into two 8 bit ie A8-A15 and AD0-AD7(multiplexed bus-it carries both address and data).At that time microprocessor sent a Read signal($\overline{RD}$) to read the instructions from memory. Now Microprocessor fetched instruction at location 1000 from memory. This instruction sent through Address/Data bus(ie AD0-AD7),then to 8 bit internal Data Bus. Through this internal Data bus ,this fetched instruction will reach at Instruction Register(IR).

At this point, we had completed first instruction fetch. Then it will be given to Instruction Decoder(ID).ID understand the binary pattern and this decoded information is given to Timing and Control unit.

When we are doing fetching and decoding of instruction fetched from address 1000,then PC will be incremented by 1.ie

PC=PC+1

=1000+1

= 1001

If we assume that the fetched instruction at location 1000 was ADD instruction,then we can see what will happen:

Timing and control unit will generate a control signal(which tells what to do(here in our ex..ADD)

Then this control signal was received by ALU, which will perform Arithmetic operation.

To do Arithmetic operation ,ALU need 2 operands.

In 8085:

ADD B means A ← A+B

Register A(Accumulator Register-8 bit)is used to store first operand and result.Register B contains second operand.

Assume that Register A contains 02 and Register B contains 03

Register B sent the value to Temporary Register through 8 bit internal data bus ..which means Temporary Register is used to hold the second operand temporarily. Now temporary register hold the value 03.Temporary register is an 8 bit register, it is for Microprocessor and it cannot be used by the programmer.

02 from Accumulator register and 03 from Temporary register added by the ALU , and the result 05 will be sent to the Accumulator register via 8 bit internal data bus.

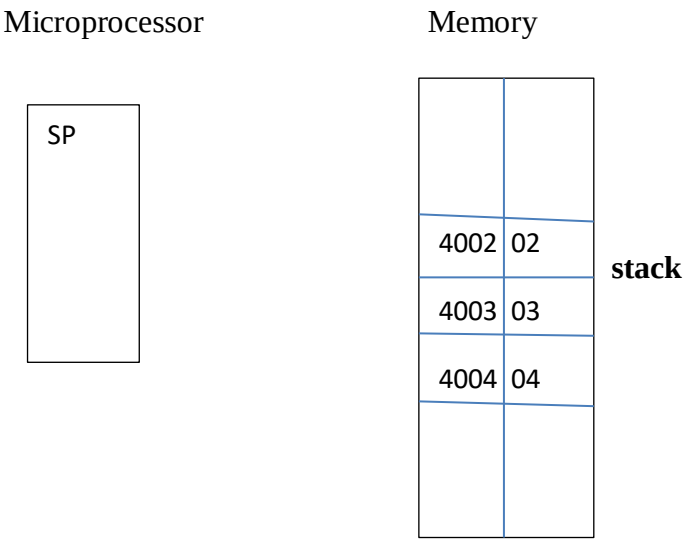
ALU can read from or write into the status register.

Ex:27+

19

46

Status of current result is stored in **status register**.it is an 8 bit register(5 valid bits+3 undefined bits).status flags of intel 8085 is given above with figure.




Stack Pointer Sp contains Address of the Top of the stack(TOS).Now SP contains 4002.

When address of TOS is 4002,then Value will be 02.

Push

SP-

POP

SP+

******** ******************************

### 3.1.5. Pin Diagram of 8085

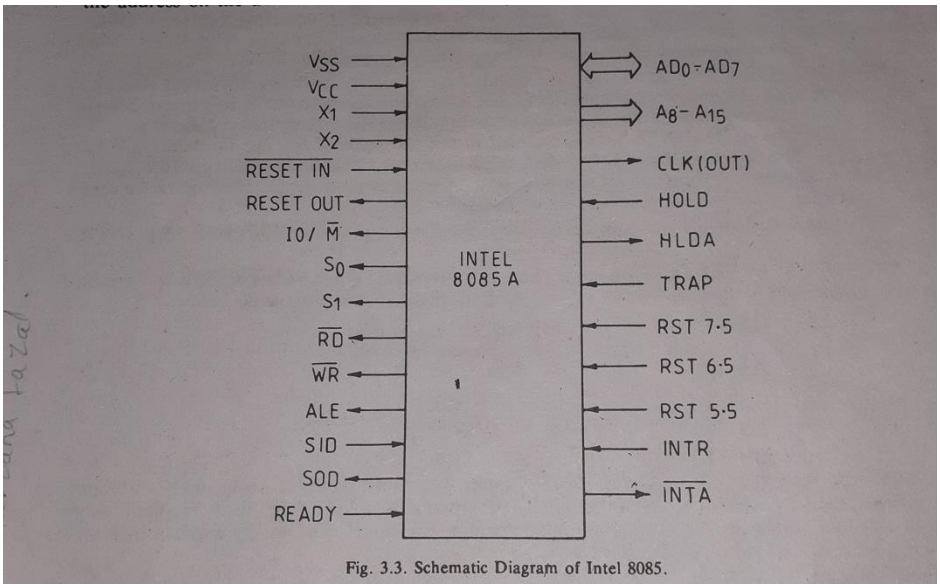



Fig. 3.3. Schematic Diagram of Intel 8085.

- 8085 is a 8 bit microprocessor
- It consists of 40 pins

- Address bus :16 bit

  A0 - A7

  A8 - A15

- Data bus :8 bit

  D0 -D7

**1.$A_8$-$A_{15}$ :**

-These are address bus and are used for the most significant bits of the memory address or 8 bits of I/O address.
-unidirectional bus

-it carries address only

**2.$AD_0$-$AD_7$ :**

-A0-A7 and D0-D7 is combined. So it can be called as Multiplexed bus

-It is a Bidirectional bus

-It is used to carry address/data

-they are used for the least significant 8 bits of the memory address or I/O address

**3.ALE (ADDRESS LATCH ENABLE):**

-It is an address latch enable signal. It goes high during first clock cycle of a machine cycle

- It shows the status of address and data line

(AD0-AD7 Carry both Address And Data)

- ALE- 1 , AD0-AD7 act as address line (ie A0-A7)
- ALE - 0 , AD0-AD7 act as data line (ie D0-D7)

**4.IO / $\overline{M}$**

-it is a 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.

-that is it specifies whether the read or write operation is from memory or from an i/o device.

**1- I/O READ OR I/O WRITE**

**0- Memory READ OR Memory WRITE**

**5.$\overline{RD}$**

-it is a signal to control READ operation .when it goes low the selected memory or I/O device is read.

- active low signal, ie 0.

$\overline{RD}$ =0, means it will perform **read** operation either from memory or from an I/O device

**6.$\overline{WR}$**

-it is a signal to control WRITE operation .when it goes low the data on the data bus is written into the selected memory or I/O operation

-Active Low Signal, ie 0.

$\overline{WR}$= 0 , it will perform **write** operation either into memory or into an I/O device

$\overline{RD}$ & $\overline{WR}$ ...both of them wouldn't become '0' at the same time.

**7. S0,S1**

-these are status signal sent by the microprocessor to distinguish the various types of operations like Read/write/fetch/halt

| S1 | S0 | OPERATION |
|---|---|---|
| 0 | 0 | HALT |
| 0 | 1 | WRITE |
| 1 | 0 | READ |
| 1 | 1 | FETCH |

| $\overline{IO}$ / M | S1 | S0 | OPERATION |
|---|---|---|---|
| 0 | 0 | 0 | HALT |
| 0 | 0 | 1 | MEMORY WRITE |
| 0 | 1 | 0 | MEMORY READ |
| 0 | 1 | 1 | OPCODE FETCH |
| 1 | 0 | 1 | I/O WRITE |
| 1 | 1 | 0 | I/O READ |
| 1 | 1 | 1 | INTERRUPT ACKNOWLEDGEMENT |

**8.READY**

-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 the peripheral is ready .if it is low the microprocessor waits till it goes high

**9.HOLD**

-it indicates that another device is requesting for the use of the address and data bus. Having received a **HOLD request** the microprocessor relinquishes (releases) the use of the buses as soon as the current machine cycle is completed.

**10.HLDA (HOLD ACKNOWLEDGEMENT)**

-it is a signal for **HOLD acknowledgement**. It indicates that the cpu has been received the **HOLD** request and it will release the Address/Data bus in next clock cycle

**11.INTR**

-it is an **interrupt request signal**. Among interrupts it has the lowest priority. An interrupt is used by io devices to transfer data to the microprocessor without wasting its time.

(Interrupts are the signals generated by external devices to request the microprocessor to perform a task)

**12. INTA**

-it is an interrupt acknowledgement sent by the microprocessor after INTR is received.
-active low signal,ie 0

- 0-process
- 1-off

**13. RST5.5, RST6.5, RST 7.5**

-these are interrupts.
**RST(Restart Instruction)**

Interrupts are of 2 types:

1.**Maskable interrupt**-handled by programmer

**2.NMI(NON Maskable Interrupt)**-It is not handled by programmer

Example of NMI-**TRAP**

**TRAP**-TRAP has the highest priority. It is used in emergency situation. it is an non-maskable interrupt.

Order of priority-

- TRAP high priority
- RST 7.5
- RST 6.5
- RST 5.5
- INTR lowest priority

These 5 pins handle interrupt signal.

**14. RESET IN**

- active low signal, ie 0
- 0 indicates microprocessor reset
- That is it reset Program Counter(PC) to initial position...ie 0

**15. RESET OUT**

- reset the devices connected to the microprocessor when the microprocessor was reset
- RESET OUT signal externally provided through this pin
- (Reset devices connected to microprocessor)

**16. X1,X2**

-these are terminals to be connected to an external crystal oscillator which drives an internal circuitry of the microprocessor to produce a suitable clock for the operation of microprocessor
-it Maintains internal frequency of 8085

**17. CLK**

-it is a clock output for user

-clock frequency- MAX 3 MHZ

**18. SID**

-it is data line for serial input.

- serial input data- from input device, data comes bit by bit serially into the microprocessor through this pin

**19.SOD**

-it is data line for serial output

Serial output data-After processing, output generated bit by bit serially

- Both SID ,SOD used for serial data communication

**20.Vcc**

+5 volt supply

**21.Vss**

- Ground signal
- Voltage source supply 0

## 3.2. Instruction Cycle

- The necessary steps that a cpu carries out to fetch an instruction and necessary data from memory, and to execute it constitute instruction cycle.
- Total time required to execute instruction:

**Instruction Cycle(IC)=Fetch Cycle(FC)+Execute Cycle(EC)**

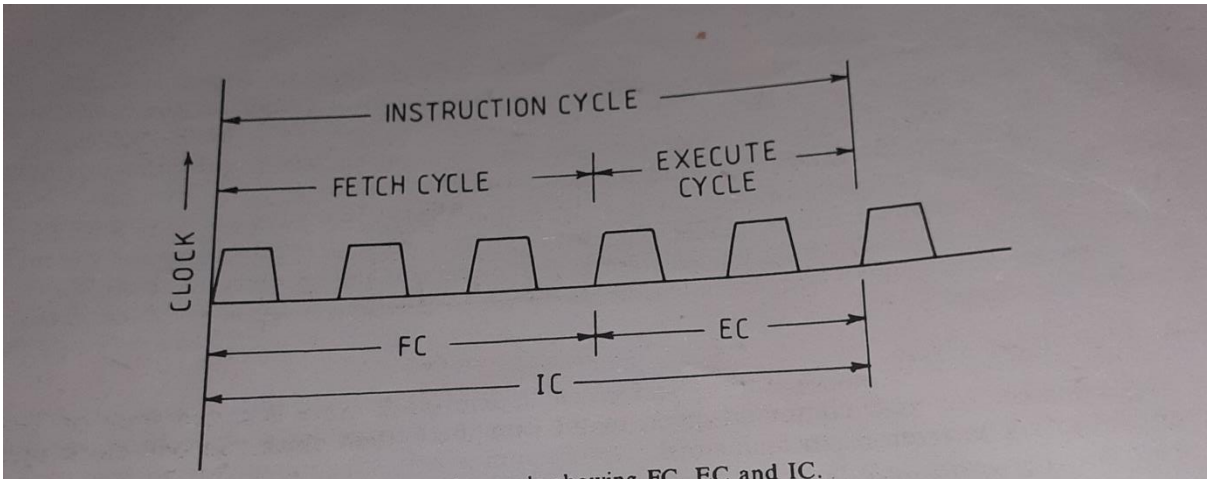



Fig. 3.4. Instruction cycle showing FC, EC and IC.

### 3.2.1.Fetch Cycle( FC)

- Necessary steps carried out to fetch instruction(opcode) from memory constitute fetch cycle

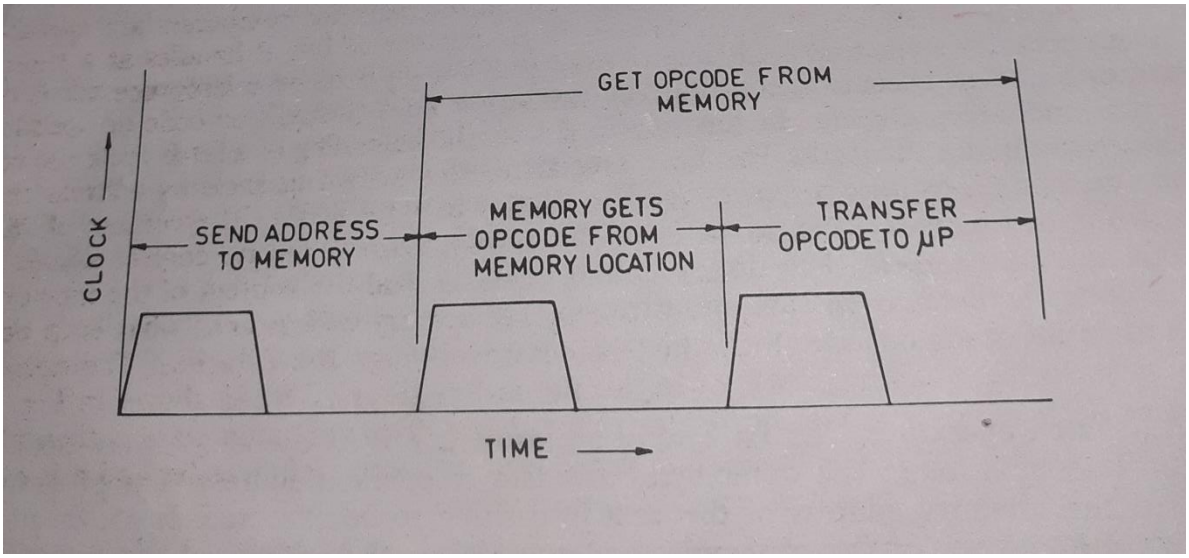
[Steps.. in short:- 1)Pc->MAR->memory , 2) Memory->MDR->IR]

1.Send contents of pc to memory

2.Memory gets the opcode from memory location

3.Opcode sent to cpu

The entire operation requires 3 clock cycle



3.2.2.**Execute Cycle (EC)**

**1.**Opcode from memory first goes to MDR(Address or data buffer)

**2.Then it goes to IR**

3.Then it was sent to decoder circuitry

4.After decoding, execution begins

5.If operand is in GPRS ,it will immediately perform

6.If instruction contains operand address ,which are in memory then cpu has to perform read operation

7.Read cycle similar to fetch cycle

8.After getting data from memory, it will perform execution

**3.2.3.Machine cycle**

-necessary steps carried out to perform the operation of accessing either memory or I/O device

-ie necessary steps carried out to perform fetch,read or write constitute a machine cycle.

- Basic operations of machine cycle

1. Opcode fetch
2. Memory read
3. Memory write
4. I/O read
5. I/O write

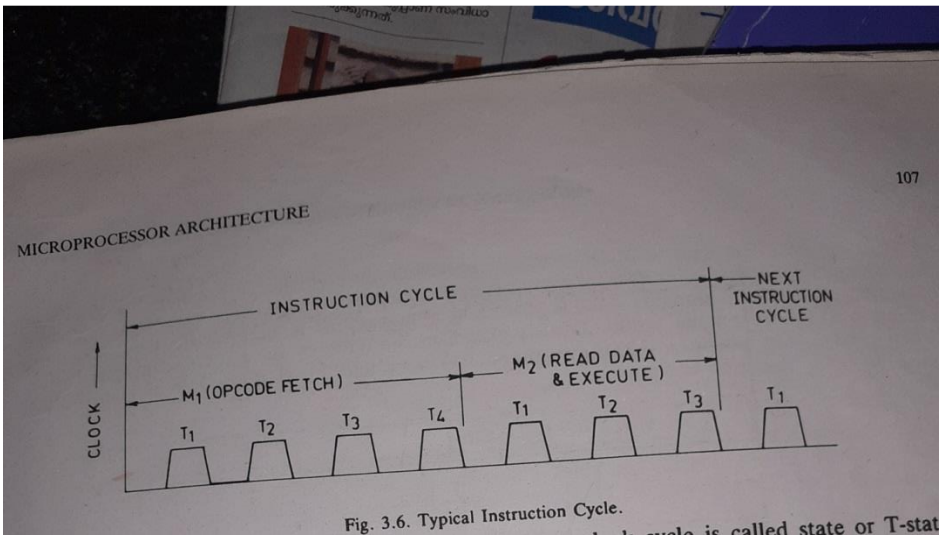



FIG:Instruction cycle

### 3.2.4.Instruction and data flow

**Instruction flow:**

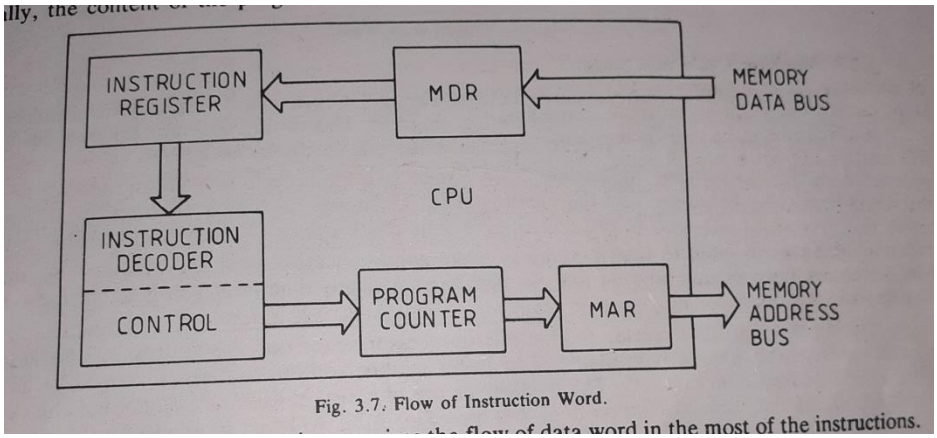



Fig:flow of instruction word

1. Contents oc pc transferred to MAR
2. Contents of MAR transferred to memory via address bus
3. Memory send opcode to microprocessor via data bus
4. Opcode comes in MDR
5. Opcode placed in IR

6. Instruction is decoded by decoder circuitary&executed

7. Finally pc get incremented

Data flow:

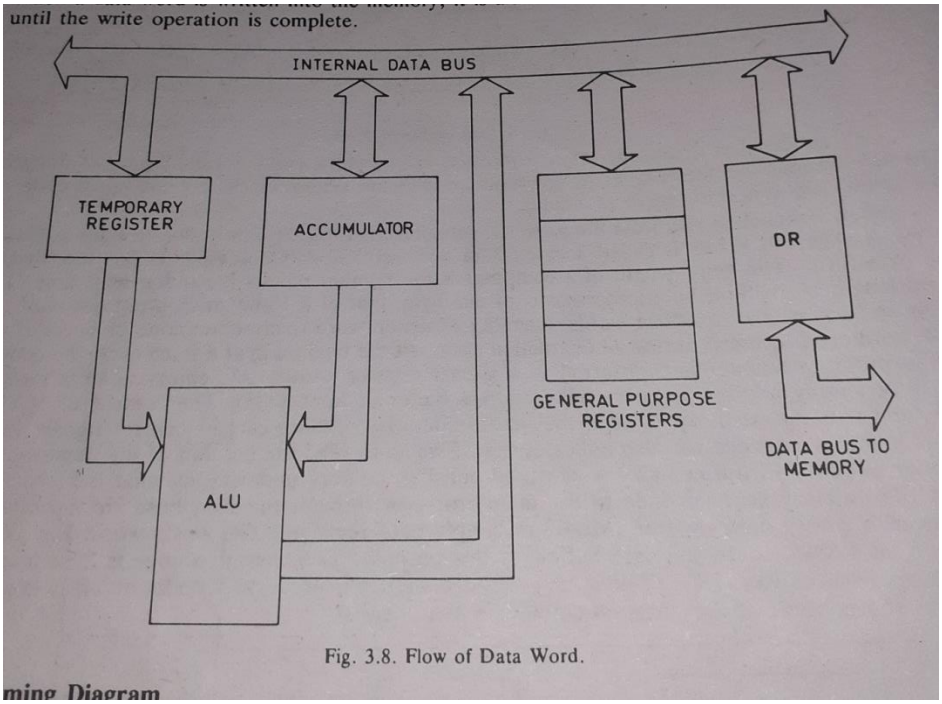



Fig:Flow of data word

The execution of an instruction requires the flow of data word in the most of the instructions. A data word is received either from the memory or input devices. The data word flows to the processor through the data bus and is placed in the accumulator or any other general purpose register depending upon the instructions. After the execution of an instruction the data is placed in a register or a memory location. After the execution of a program the result is placed in the memory or sent to an output device. When the data word is written into the memory, it is also held in MDR until the write operation is complete.

### 3.3.Timing diagram

The necessary steps carried out in a machine cycle can be represented graphically. Such a graphical representation is called timing diagram .Timing diagram for opcode fetch,memory read,memory write,I/O read,I/O write are given below:

.

Points to be remembered from previous topic,ie pin diagram of 8085…**No need to write** this again:

$\overline{RD}$ 0

$\overline{WR}$ 0

Io/$\overline{M}$ : 0 memory read/write

1 i/o read/write

| s0 | s1 | operation |
|---|---|---|
| 0 | 1 | read |
| 1 | 0 | write |
| 1 | 1 | opcode fetch |

ALE 1 , AD0-AD7 ACT AS ADDRESS BUS

ALE 0 ,AD0-AD7 ACT AS DATA BUS

TIMING DIAGRAM FOR OPCODE FETCH

microprocessor. The memory is disabled when RD goes high during $T_3$. The fetch cycle is pleted by $T_3$. The opcode is decoded in $T_4$. The microprocessor examines READY signal . If it is high, the microprocessor enters into $T_3$ state. If it is low, the microprocessor inserts it state in between $T_2$ and $T_3$.

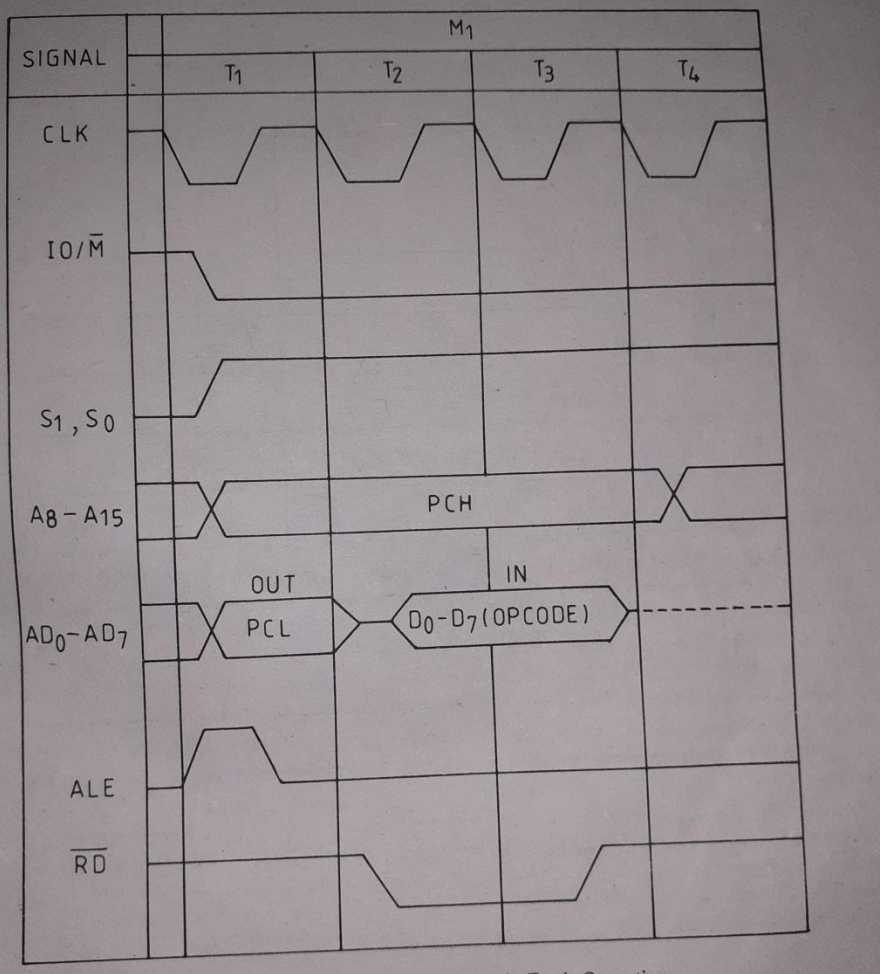


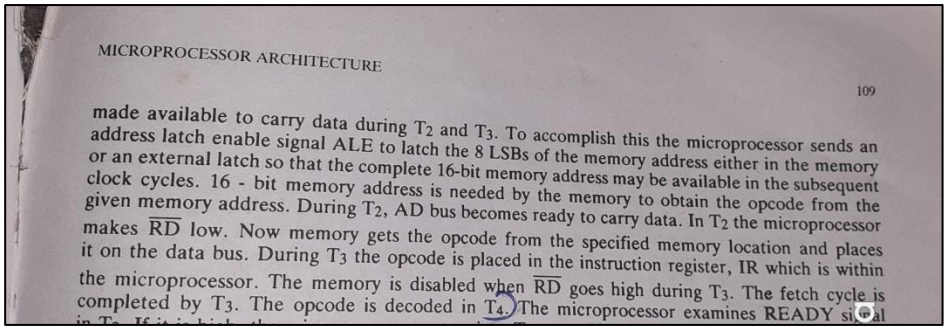

Fig. 3.9. Timing Diagram for Opcode Fetch Operation.



memory read, memory write, I/O read and ...

**3.3.1. Timing Diagram for Opcode Fetch Cycle.** In a fetch cycle the microprocessor fetches the opcode of an instruction from the memory. Fig. 3.9 shows the timing diagram for an opcode fetch cycle. $T_1$, $T_2$, $T_3$ and $T_4$ are consecutive four clock cycles. The microprocessor issues a low IO/$\overline{M}$ signal to indicate that it wants to make communication with the memory. Again the microprocessor sends out high $S_0$ and $S_1$ signals to indicate that it is going to perform fetch operation.

During the first clock cycle, $T_1$, the microprocessor sends out the address of the memory location where the opcode is available. The 16-bit memory address is sent through the address bus A and address/data bus AD. The 8 MSBs of the memory address are sent over the A bus, and 8 LSBs of the memory address over AD bus. Since the AD bus is needed to transfer data during subsequent clock cycles, it is used in time-multiplexed mode. Therefore, it has to be





made available to carry data during $T_2$ and $T_3$. To accomplish this the microprocessor sends an address latch enable signal ALE to latch the 8 LSBs of the memory address either in the memory or an external latch so that the complete 16-bit memory address may be available in the subsequent clock cycles. 16 - bit memory address is needed by the memory to obtain the opcode from the given memory address. During $T_2$, AD bus becomes ready to carry data. In $T_2$ the microprocessor makes $\overline{RD}$ low. Now memory gets the opcode from the specified memory location and places it on the data bus. During $T_3$ the opcode is placed in the instruction register, IR which is within the microprocessor. The memory is disabled when $\overline{RD}$ goes high during $T_3$. The fetch cycle is completed by $T_3$. The opcode is decoded in $T_4$. The microprocessor examines READY si[illegible]al

**3.3.2. Memory Read.** In a memory read cycle the microprocessor reads the content of a memory location. The content is then placed either in the accumulator or any other register of the CPU. Now let us take an example of two byte long instruction.

MVI A, 05

In the coded form it is written as

3E, 05

where 3E is opcode for MVI A instruction and 05 is data.

This instruction requires two machine cycles, $M_1$ and $M_2$. The first machine cycle $M_1$ is to fetch the operation code 3E from the memory. The timing diagram for opcode fetch operation has already been shown in Fig. 3.9. The second machine cycle $M_2$ is for reading the data (05) from the memory as shown in Fig. 3.11. It is a memory read cycle. IO/$\overline{M}$ goes low indicating that the address is for memory. $S_1$ and $S_0$ are set to 1 and 0 respectively for read operation, see Table 3.1. On the address lines $A_8$—$A_{15}$, the 8 MSBs of the memory address of the data (05) are sent. During $T_1$, 8 LSBs of the memory address of the data are sent on $AD_0$—$AD_7$. During $T_2$, 8 LSBs of the address is latched and $AD_0$—$AD_7$ are made free for data transmission. $\overline{RD}$ goes low in $T_2$ to enable the memory for read operation. Now data is placed on data bus. During $T_3$ the data enters into the CPU. In $T_3$, $\overline{RD}$ goes high and disables the memory. For MVI A, 05 instruction the data enters into the accumulator. The timing diagram for an opcode fetch cycle and a memory read cycle are same except the status signals $S_1$ and $S_0$. MR cycle consists of only three clock cycles.

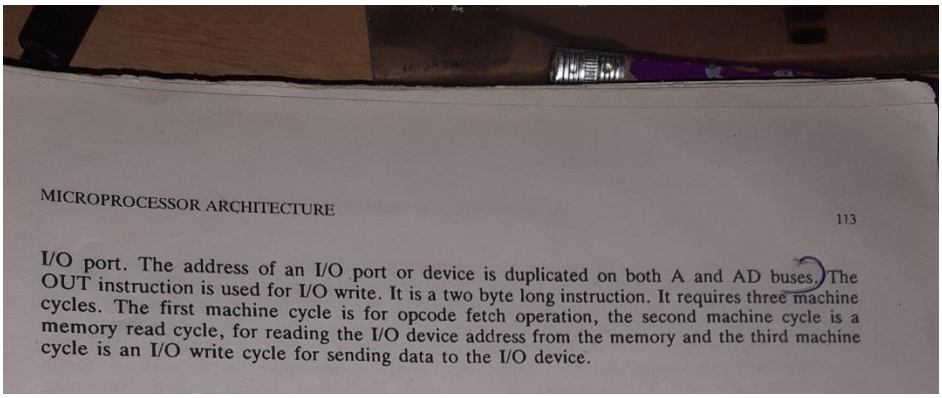
Now consider a three byte long instruction, for example; LXI H, 2500H. This instruction requires three machine cycles, one fetch cycle and two consecutive memory read cycles. The 1st machine cycle is to fetch the opcode, the 2nd machine cycle to read 8 LSBs of the 16-bit data (2500), and the 3rd machine cycle to read 8 MSBs of the 16-bit data (2500). The instruction LDA 2400H requires four machine cycles. The 1st machine cycle is to fetch the operation code, the 2nd machine cycle for reading the 8 LSBs of the memory address (2400H), the third machine cycle for reading 8 MSBs of the memory address. In the fourth machine cycle the microprocessor sends the memory address 2400H to get its content into the accumulator.

**3.3.3 I/O Read.** In an I/O read cycle the microprocessor reads the data available at an input port or input device. The data is placed in the accumulator. An I/O read cycle is similar to memory read cycle. The only difference between a memory read cycle and I/O read cycle is that signal IO/$\overline{M}$ goes high in case of I/O read. It indicates that the address on the A-bus is for an input device. In timing diagram all other signals will remain same as shown in Fig. 3.11. In case of I/O device or I/O port the address is only 8-bits long and therefore the address of I/O devices is duplicated on both A and AD buses.

The IN instruction is two byte long. It requires three machine cycles for execution. The first machine cycle is opcode fetch cycle, the second machine cycle is a memory read cycle to read the input device or input port address and the third machine cycle is I/O read cycle to read the data from the device or port.

**3.3.4. Memory Write.** In a memory write cycle the CPU sends data from the accumulator or any other register to the memory. The timing diagram for a memory write cycle is shown in Fig. 3.12. The status signals $S_0$ and $S_1$ are 1 and 0 respectively for write operation. $\overline{WR}$ goes low in $T_2$ indicating that the write operation is to be performed. During $T_2$ the AD-bus is not disabled as it is done in case of memory or I/O read operation. But the data to be sent out to the memory is placed on the AD-bus. As soon as $\overline{WR}$ goes high in $T_3$, the write operation is terminated. The instructions MOV M, A; STA 2500H etc. use memory write cycle.

**3.3.5. I/O Write.** In an I/O write cycle the CPU sends data to an I/O port or I/O device from the accumulator. An I/O write cycle is similar to a memory write cycle. In case of I/O write cycle IO/$\overline{M}$ goes high indicating that the address sent out by the CPU is for I/O device or



MICROPROCESSOR ARCHITECTURE 113

I/O port. The address of an I/O port or device is duplicated on both A and AD buses. The OUT instruction is used for I/O write. It is a two byte long instruction. It requires three machine cycles. The first machine cycle is for opcode fetch operation, the second machine cycle is a memory read cycle, for reading the I/O device address from the memory and the third machine cycle is an I/O write cycle for sending data to the I/O device.

**TIMING DIAGRAM FOR MEMORY READ**

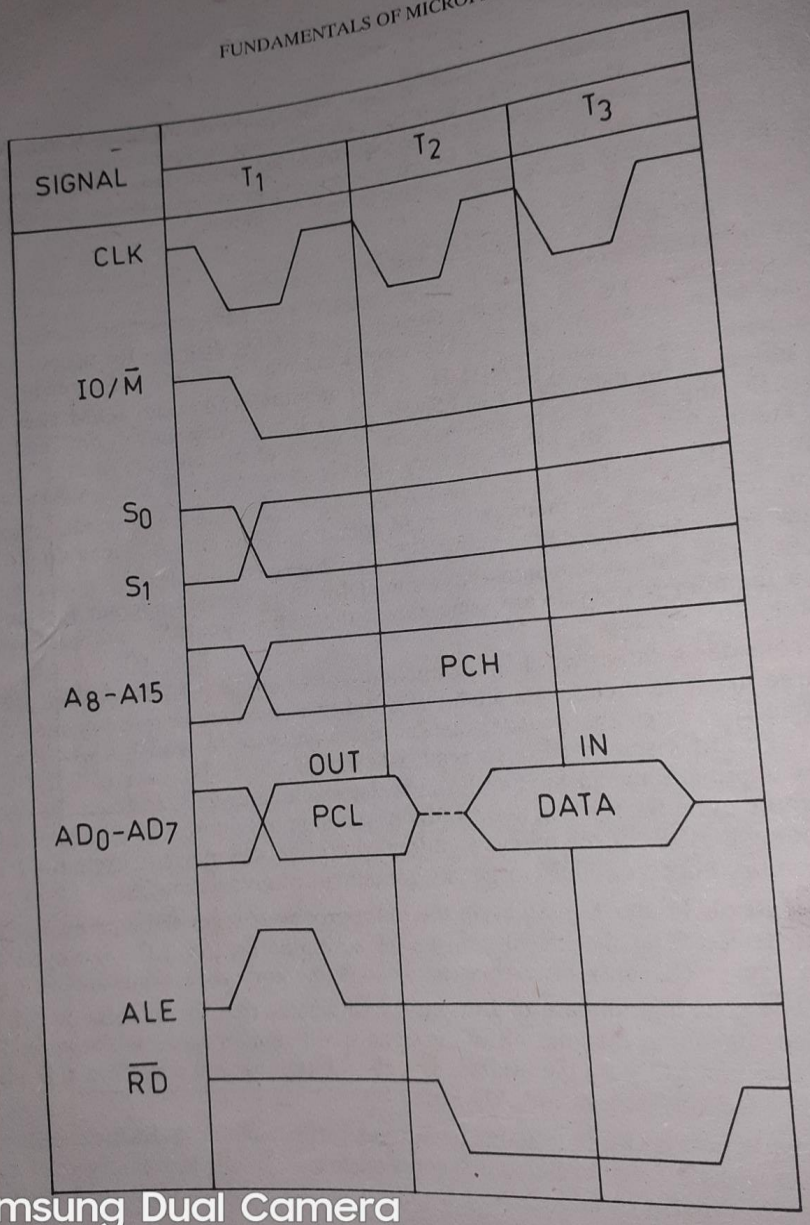





Fig. 3.12 Timing Diagram For Memory Read Operation.

the memory is placed on the AD-bus. As soon as $\overline{WR}$

TIMING DIAGRAM FOR MEMORY WRITE

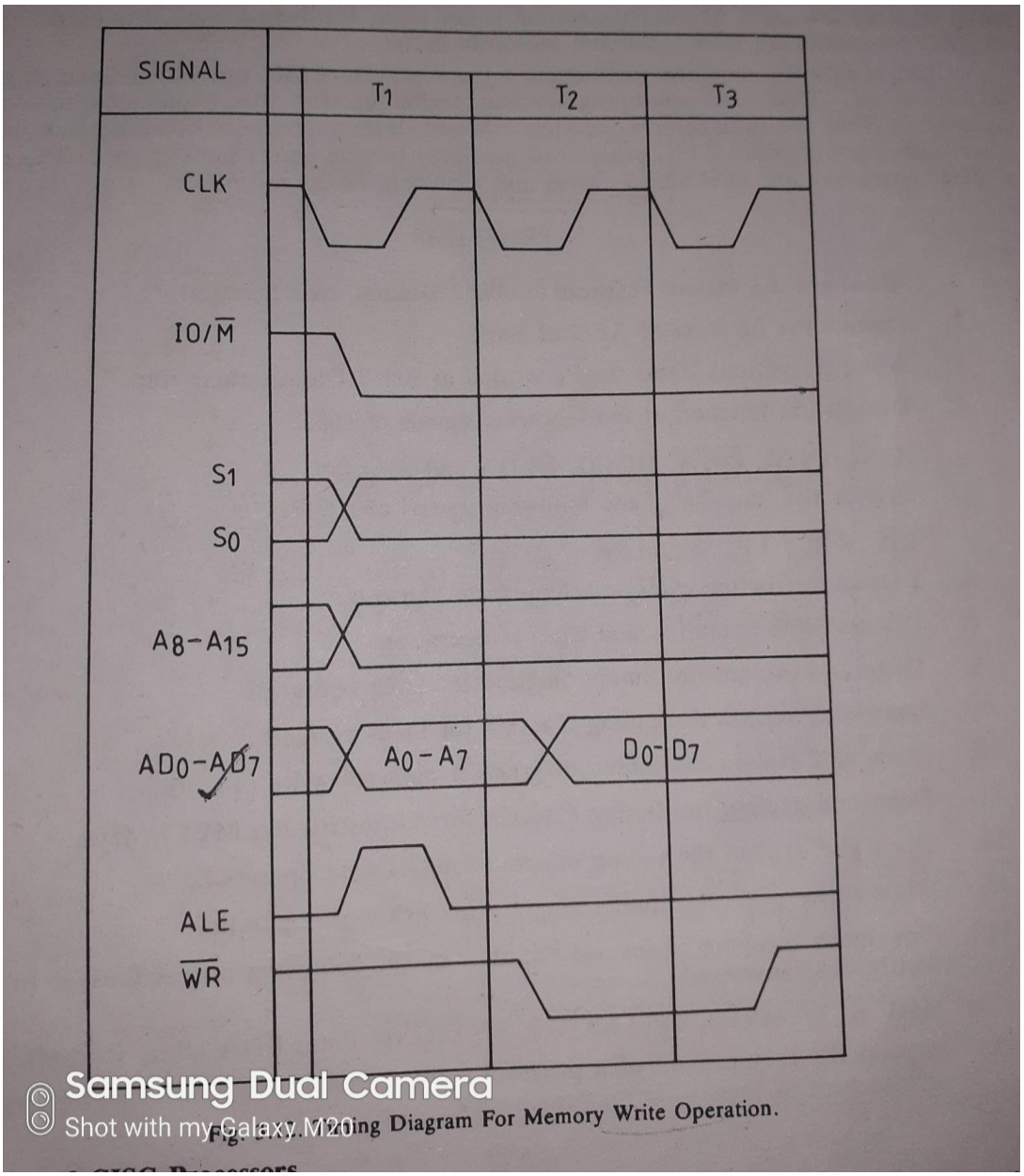
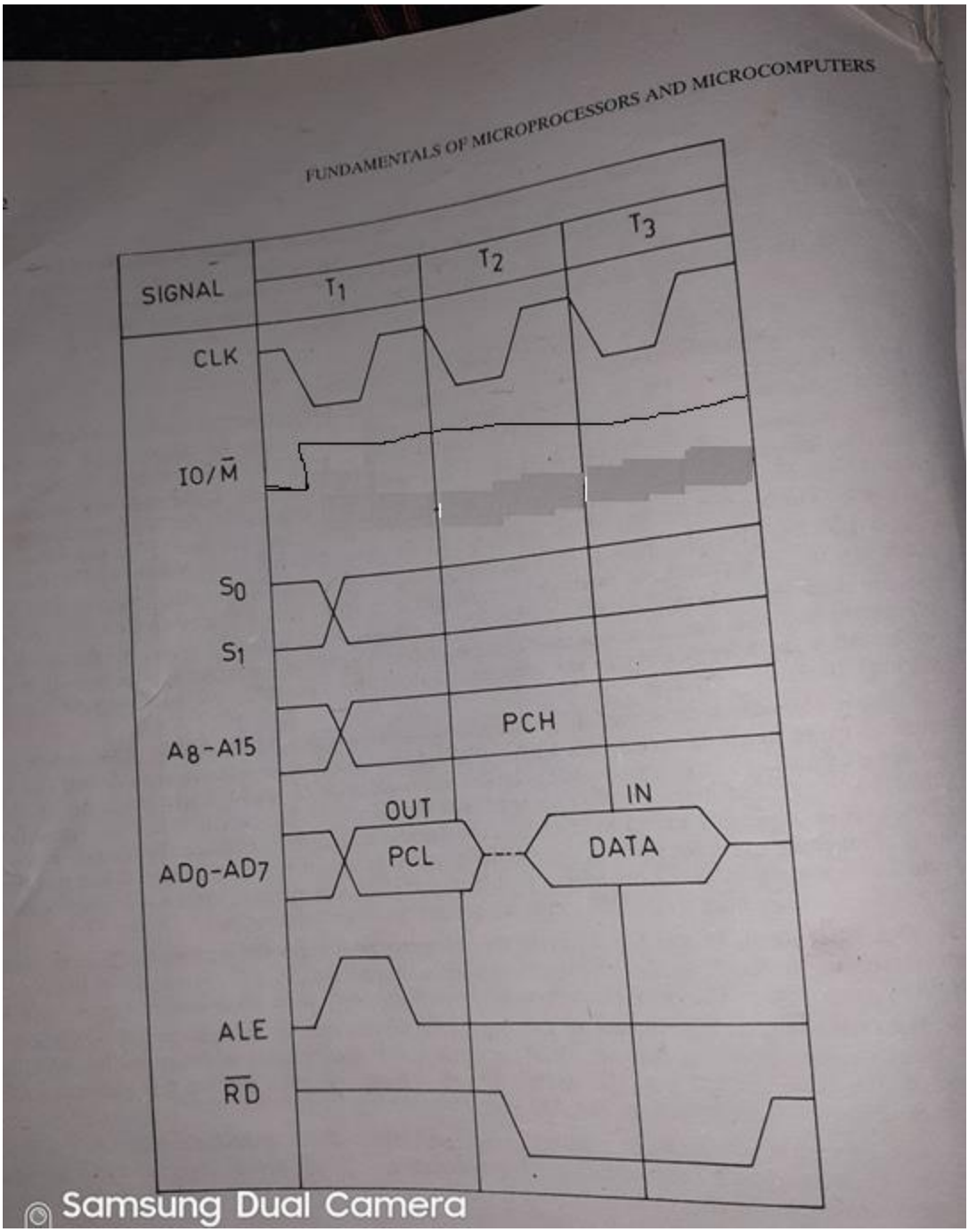


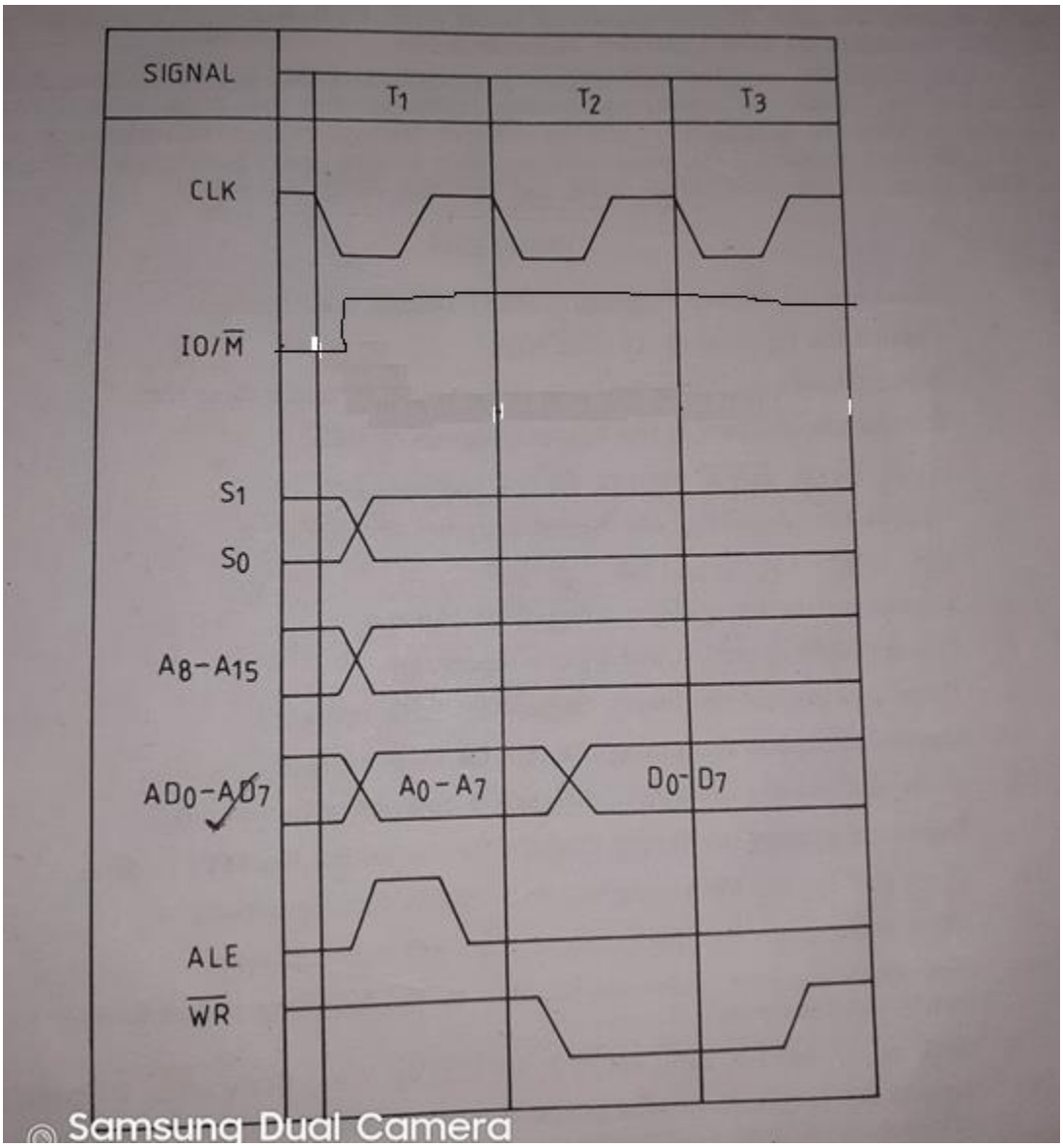



Fig. Timing Diagram For Memory Write Operation.

**TIMING DIAGRAM FOR I/O READ**



## TIMING DIAGRAM FOR I/O WRITE




## INTERRUPTS OF 8085

- Interrupts are the signals generated by the external device to request the microprocessor to perform some task
- Interrupts are classified into different groups based on the parameter:
  - Vector interrupt
  - Non-vector interrupt
  - Maskable interrupt

- Non-maskable interrupt
- Software interrupt
- Hardware interrupt

**Vector interrupt**

(interrupt have fixed vector address)

- Interrupt address is known to processor
- [ie from which device interrupt is coming is known to processor]
- These interrupt signal generated by pins like:
- RST 7.5,RST 6.5,RST 5.5,TRAP

**Non-Vector interrupt**

- Interrupt address is not known to the processor
- (interrupt didn't have fixed vector address)
- Ex:INTR

**Maskable Interrupt**

- -we can disable the interrupt by writing some instructions into the program
- Ex:RST 7.5,RST 6.5,RST 5.5

**Non-Maskable interrupt**

- We cannot disable the  interrupt by writing some instructions into the program
- Ex:TRAP

**Software interrupt**

-programmer has to add the instruction into the program to execute the interrupt

(Interrupt given by the programmer)

-these are the interrupts caused by special instructions

8 s/w interrupts in 8085

RST0,RST1,RST2,RST3,RST4,RST5,RST6,RST7

**Hardware interrupt**

- Interrupt caused by h/w device(ie external device)
- 5 h/w interrupt
- Pins connected to h/w interrupt:

TRAP high priority

RST 7.5

RST 6.5

RST 5.5

INTR low priority

**********