SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

SECA1404 – Microprocessor & Microcontroller Based Systems

COURSE OUTCOMES

On completion of the course, student will be able to

CO1 - Understand the architecture and functional blocks of Processor 8085.

CO2 - Understand the addressing modes and instructions of Microprocessor 8085.

CO3 - Learn the architecture and functions of important interface chips.

CO4 - Understand the architecture and functional blocks of Processor 8086.

CO5 - Learn the architecture and functions of 8051 and basics of Arduino controller.

CO6 - Design and implement Microprocessor and Microcontroller based system.

Syllabus :

UNIT 1 BASIC CONCEPTS **9 Hrs.**

8085 Microprocessor - Architecture and its operation, Concept of instruction execution and timing diagrams, fundamentals of memory interface - Addressing modes.

UNIT 2 8085 INSTRUCTION SET AND ASSEMBLY LANGUAGE PROGRAMMING **9 Hrs.**

Instruction classifications, Writing and executing simple programs - Arithmetic and logic operations – Data transfer - Branching - Looping – Indexing - Counter and time delays - Writing subroutine - Conditional call and return instruction, simple programs.

UNIT 3 INTERFACING **9 Hrs.**

Basic Interface concepts, memory mapped I/O and I/O mapped I/O, Interrupt and vectored interrupt, Programmable peripheral interface 8255 - Programmable Interval timer 8253 - Programmable interrupt controller 8259 - Programmable DMA controller 8257.

UNIT 4 8086 ARCHITECTURE **9 Hrs.**

Architecture – Minimum mode and Maximum mode operation – Address Generation - Addressing modes - Overview of 8086 instruction set - Instruction format - Assembler Directives – Designing a Single Board Computer.

UNIT 5 MICROCONTROLLER **9 Hrs.**

Introduction - Architecture of 8051 - Memory organization - Addressing modes - Instruction set – Assembly Language Programming - Jump, Loop and Call Instructions - Arithmetic and Logic Instructions - Bit Operations -Programs – Introduction to Arduino.

TEXT / REFERENCE BOOKS

1. Ramesh Goankar, "Microprocessor architecture programming and applications with 8085 / 8088", 5th Edition, Penram International Publishing.
2. A.K.Ray and Bhurchandi, "Advanced Microprocessor", 1st Edition, TMH Publication.
3. Kenneth J.Ayala, "The 8051 microcontroller Architecture, Programming and applications" 2nd Edition ,Penram international.
4. Doughlas V.Hall, "Microprocessors and Digital system", 2nd Editon, Mc Graw Hill,1983.
5. Md.Rafiquzzaman, "Microprocessors and Microcomputer based system design", 2nd Editon,Universal Book Stall, 1992.
6. Hardware Reference Manual for 80X86 family", Intel Corporation, 1990.
7. Muhammad Ali Mazidi and Janice Gillispie Mazidi, "The 8051 Microcontroller and Embedded Systems", 2nd Edition, Pearson.
8. "Arduino Made Simple" by Ashwin Pajankar.

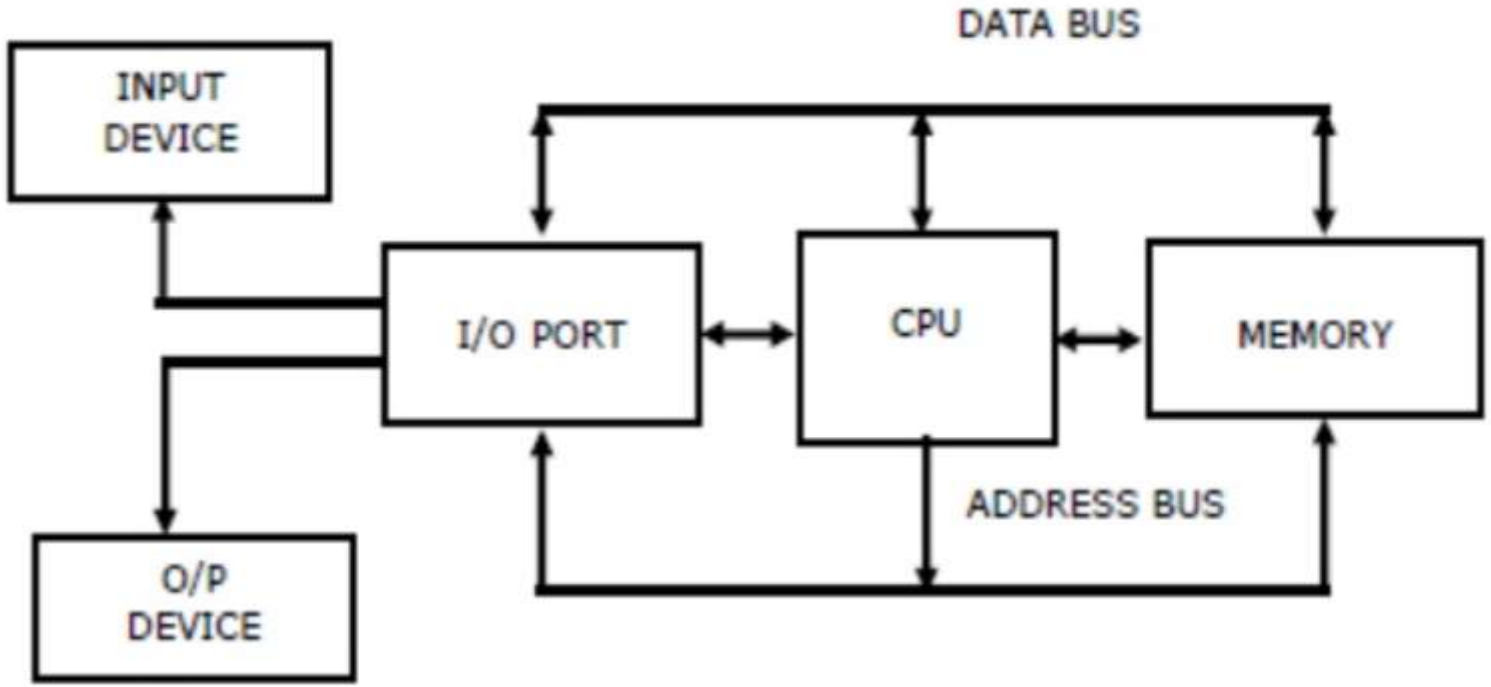
SECA1401 – Microprocessor & Microcontroller Based Systems

Topics to be covered in this video Lecture

- Introduction
- Key points of Microprocessor & Microcontroller
- Features of 8085
- Bus Structure
- Architecture of 8085
- Timing & Control Unit
- Concepts of Registers
- PIN Description
- Concepts of Interrupts
- Concepts of Timing Diagrams
- Model University Questions
- Tips & Tricks to score Good Marks

Introduction to Microcomputer – Concepts of Microprocessor & Microcontroller

General Structure of a Microcomputer system



Fig(1): General Structure

Key Points:

- CPU (Microprocessor or Microcontroller)
- Input Unit
- Output Unit
- Ports & Bus

(Fig-1) Image Adopted from: https://krisjavahmi.wordpress.com/2016/02/13/what-is-required-at-the-beginning

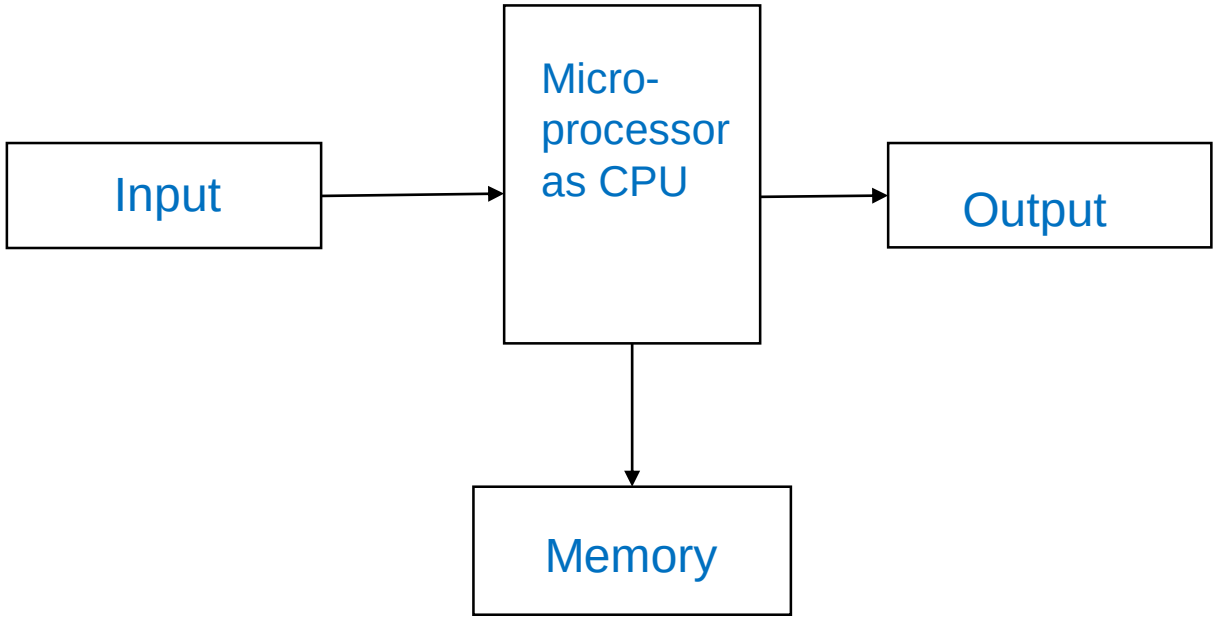
Introduction to Microcomputer – Concepts of Microprocessor & Microcontroller

Microprocessor: Is a Program controlled semiconductor device which reads data from memory, Decodes and executes the instructions. Also known as Central Processing Unit in computers (CPU).
- A. Nagoor Kani, Microprocessor & Microcontrollers- Edition

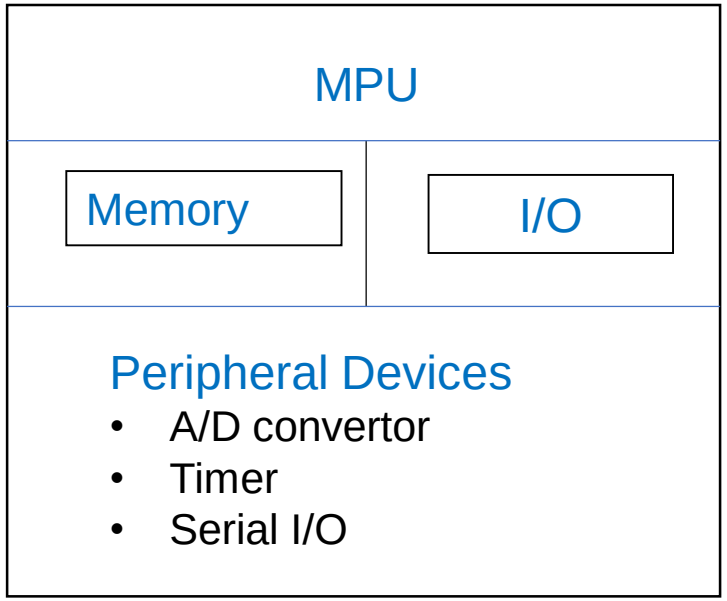
Microcontroller: Is a Programable semiconductor IC capable of performing Arithmetic and Logical operations. Microcontroller contains ALU, Registers, I/o Ports, Timing & Control circuits, **Memory**.
- A. Nagoor Kani, Microprocessor & Microcontrollers- Edition

Microprocessor	**Microcontroller**
Moves code and data between processor and External memory	Moves code and data within the controller
Contains ALU/CPU, Registers, Control Units	Contains ALU, Registers, Timing &Control Units
Memory is connected externally with the CPU	Memory is present inside as an internal unit
Used to design General purpose systems	Used to design Application Specific Systems

Introduction to Microcomputer – Concepts of Microprocessor & Microcontroller



Fig(2):Microcomputer with Microprocessor as CPU



Fig(3):Microcontroller Block

Points to remember:

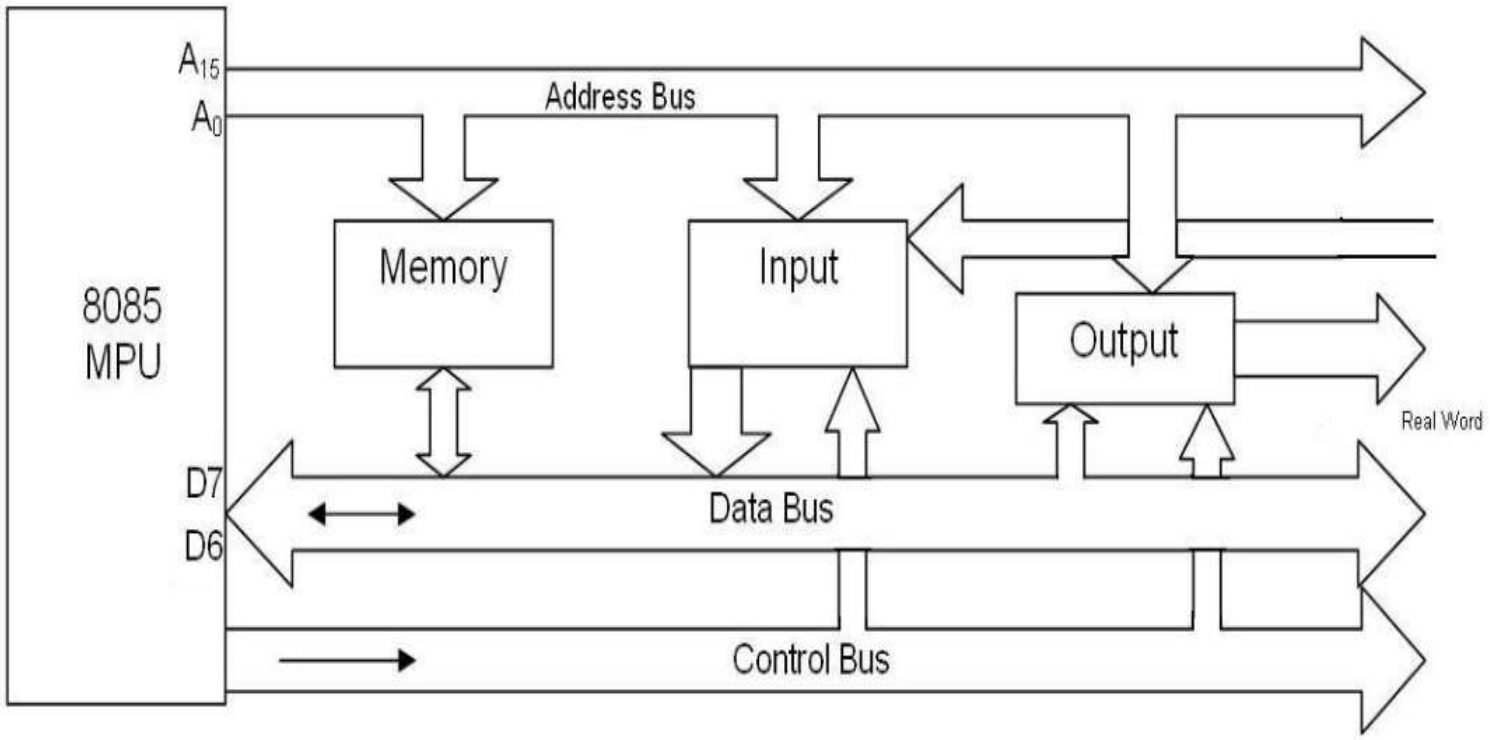
- CPU = ALU & Control Unit
- MPU = Microprocessor Unit

Introduction – Continued

Features of 8085:

- 8-bit general purpose Microprocessor
- It is a 40 pin DIP IC &Requires +5 v power supply
- Can operate with 3 -6 MHz clock frequency
- It has 16 Bit Address lines
- 8 bit Data lines[Multiplexed with lower 8 bit address lines] (AD0- AD7)
- 16 Bit (PC)Program counter & 16 Bit Stack pointer (SP)
- Six 8 bit General purpose registers B,C,D,E,H,L (can be arranged in pairs BC,DE,HL)

Introduction – Bus Structure of 8085



Fig(4):8085 Bus structure

Points to remember:

Bus:

It is a collection of wire which is used to transfer the data or address information or control signals from source to destination.

Types:

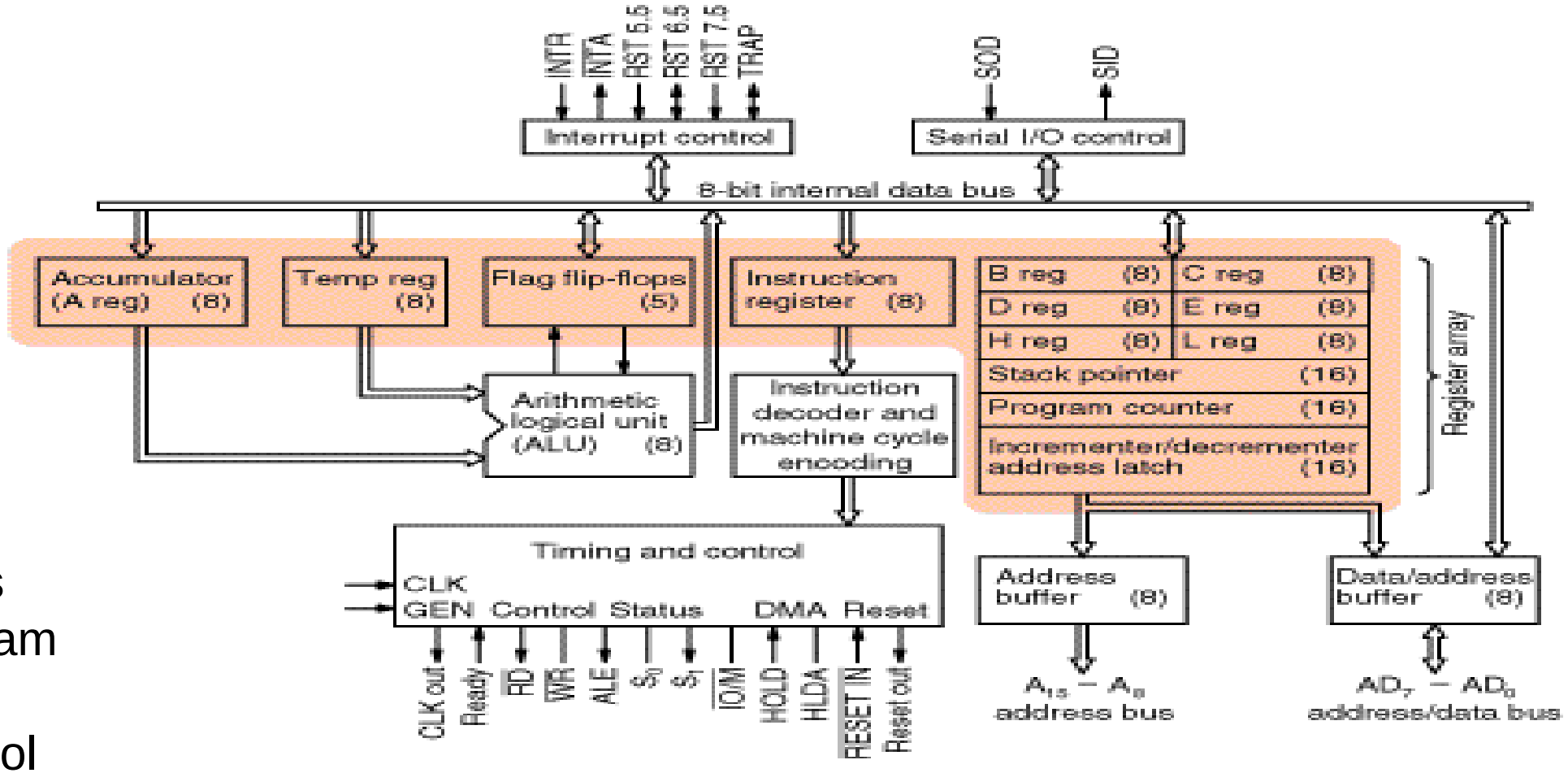
1. Address Bus: To transfer 16 bit address (Unidirectional)
2. Data Bus: To carry 8 bit data(Bidirectional)
3. Control Bus: To carry control signals

(Fig-4) Image Adopted from: https://scanftree.com/microprocessor/Bus-Structure-8085

Architecture of 8085

Key Points:

- ALU
- Registers
- Flag Registers
- Stack & Program Counters
- Timing& Control Unit signals



Fig(5):8085 Architecture Diagram

(Fig-5) Image Adopted from: http://ce.sharif.edu/courses/86-87/1/ce126/resources/root/8085%20Microprocessor.pdf

Arithmetic and Logic Unit (ALU)

Performs the following operations:

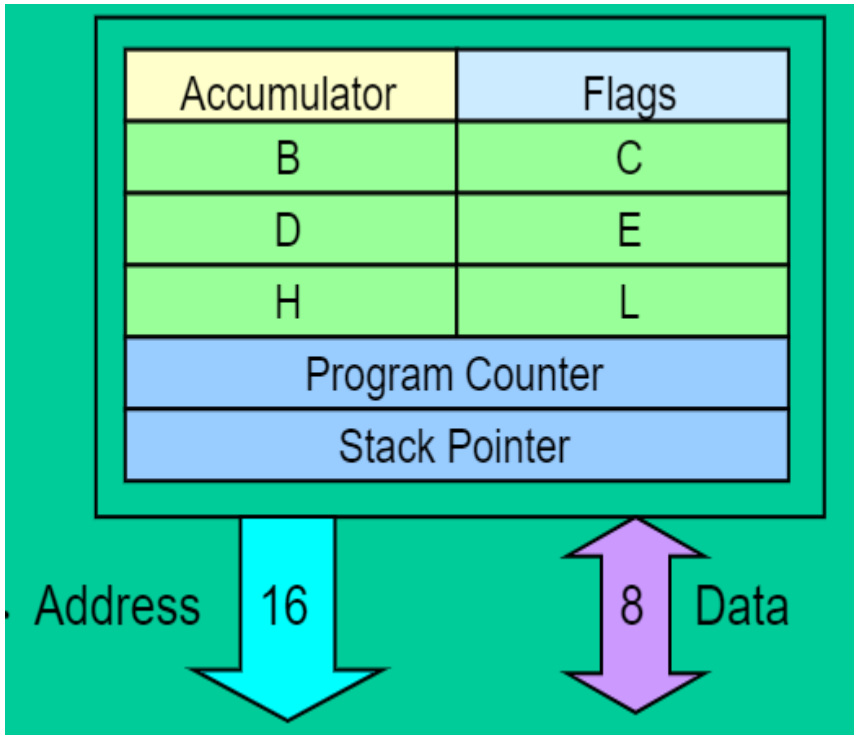
- Addition ,Subtraction
- Logical AND ,Logical OR, Logical EXCLUSIVE OR
- Complement (logical NOT), Compare operations
- Increment (add 1),Decrement (subtract 1) Operations
- Left shift / right shift

✓Accumulator (A) register & Temporary Registers holds the data during operation & After processing the result is stored in Accumulator(A)

✓Flag registers are set /reset based on the operation performed

Registers

- **Accumulator**
  - ✓The accumulator is an 8-bit register that is a part of 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. The accumulator is also identified as register A.
- **General Purpose Register:** B, C, D, E, H, and L.
- **Temporary Register-** W and Z register which is not available to programmer. Because it is used internally located above B and C register



Fig(6):8085 Registers

(Fig-6) Image Adopted from: inspirit.net.in/books/academic/8085%20Microprocessor%20-%20Ramesh%20Gaonkar.pdf

Special Purpose Register

Program Counter (PC)

- ✓16-bit Special Purpose register deals with sequencing the execution of instructions
- ✓This register is a memory pointer.
- ✓The function of the program counter is to point to the memory address from which the next byte is to be fetched.
- ✓When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location

Stack Pointer (SP)

- The stack pointer is also a 16-bit register which is used to point memory location called stack. Memory area in which data to be retrieved are placed. The beginning of the stack is defined by loading 16-bit address in the stack pointer. Accessed Last in First out(LIFO)

Special Purpose Register

Instruction Register / Decoder

- Temporary storage for the current instruction of a program
- Latest instruction sent here from memory prior to execution
- Decoder then takes instruction and ‘decodes’ or interprets the instruction
- Decoded instruction then passed to next stage
- Not programable
- Can’t accessed by instructions

Example: Assume A = 56H,When an instruction MOV C,A is given ,

Process Flow:

- The opcode for MOV C,A is 4FH.Its placed in data bus from memory
- From Data Bus 4FH is placed in Instruction Decoder (After this ,its decoded)
- Content of Accumulator(56H) is moved temporary register in ALU
- Now from Temporary register the content is moved to C register

Flag Registers

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

→ Data bits

→ Flag Registers

Sign Flag(S): After the arithmetic or logical operation D7 bit is 1.sign flag will be SET. Otherwise RESET. Sign flag indicates that

Example 1: Lets assume an operation has resulted a value of 80H .It can be represented as

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

Now D7(MSB) =1 .Then it indicates a negative number (sign flag is set)

If D7 = 0 , the number is considered as positive

Zero Flag(Z): when arithmetic operation results in zero, then Zero (Z) flag is set to one

Example: Lets assume that a value of 5 is stored in accumulator (A register). Now a command to subtract the accumulator(A) with a value of 5 is given. Then the result will be ZERO. Now D6 bit =1.

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	0	0	0

Flag Registers

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

→ Data bits

→ Flag Registers

Auxiliary Carry Flag(AC): Used in BCD number systems. In any arithmetic or logical operation a overflow from D3 to D4 bit occurs then AC Flag is =1 (set) otherwise RESET.

Example: Let us assume that, A=1BH& B=28H,when we add these two numbers the result is 4BH.In this operation a carry overflow occurs from D3 to D4.

Hence D4 bit = SET

Parity Flag(P): Number of one's present in accumulator. If there is even number of one's Even parity will set. If there is odd number of one's odd parity will set.

Carry Flag(CY): The flag is set if there is overflow out of bit 7.

When we add 45H & B3H as shown then we have a overflow at D7 bit, Then CY Flag is = 1(SET)

45H – 0 1 0 0 0 1 0 1

B3 H - 1 1 1 1 0 0 1 1

1] 0 0 1 1 1 0 0 0

Timing and Control Unit

- It synchronizes all microprocessor operations with the clock signal
- Contains an oscillator and a sequencer. Oscillator generates clock signals which are used to synchronize the registers

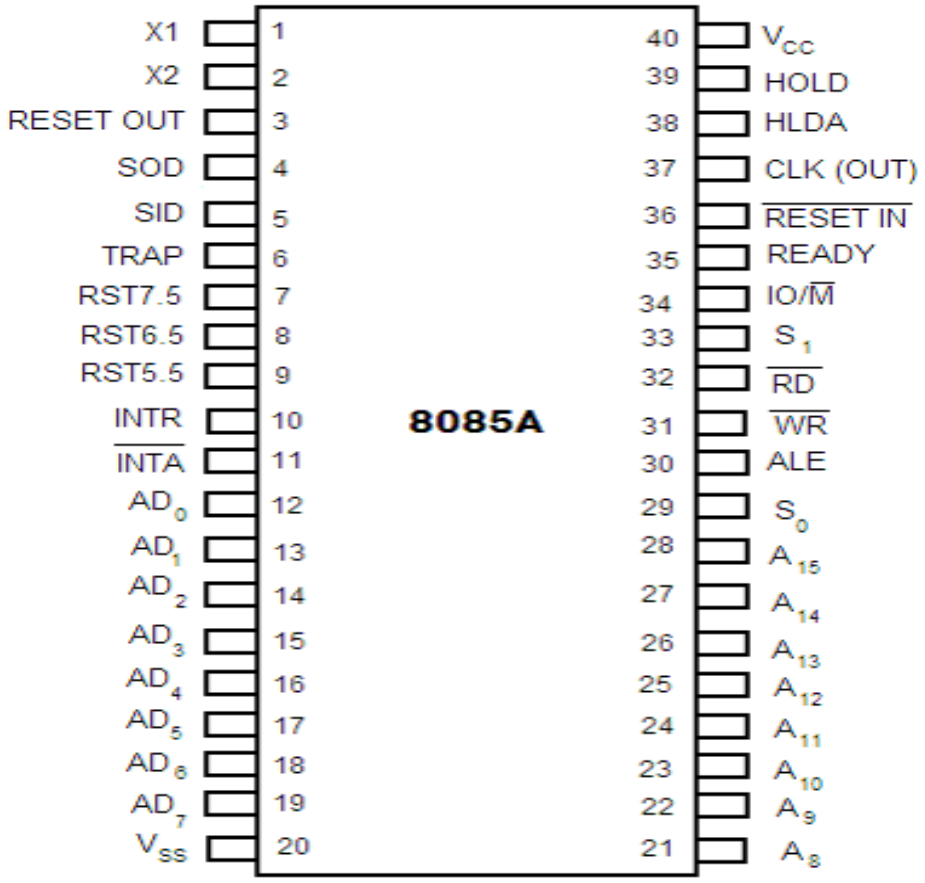
  Status Signals :S0,S1,IO/M'

  Control signals : READY,RD',WR',ALE

  DMA signals: HOLD,HLDA

  RESET Signals: RESETIN',RESETOUT'
- It controls the operation of different units of the CPU.
- It controls the data flow between CPU and Memory, CPU and peripheral devices.
- It provides control, status and reset signals to perform any memory and input output related operation.

8085 PIN DESCRIPTION



Fig(7):PIN Diagram of 8085

ALE (Address Latch Enable**)**

Address latch enable, during the first clock state of a machine cycle, it become high and enables the address to get latched either in to the memory or external latch.

Read(RD') (active low output):

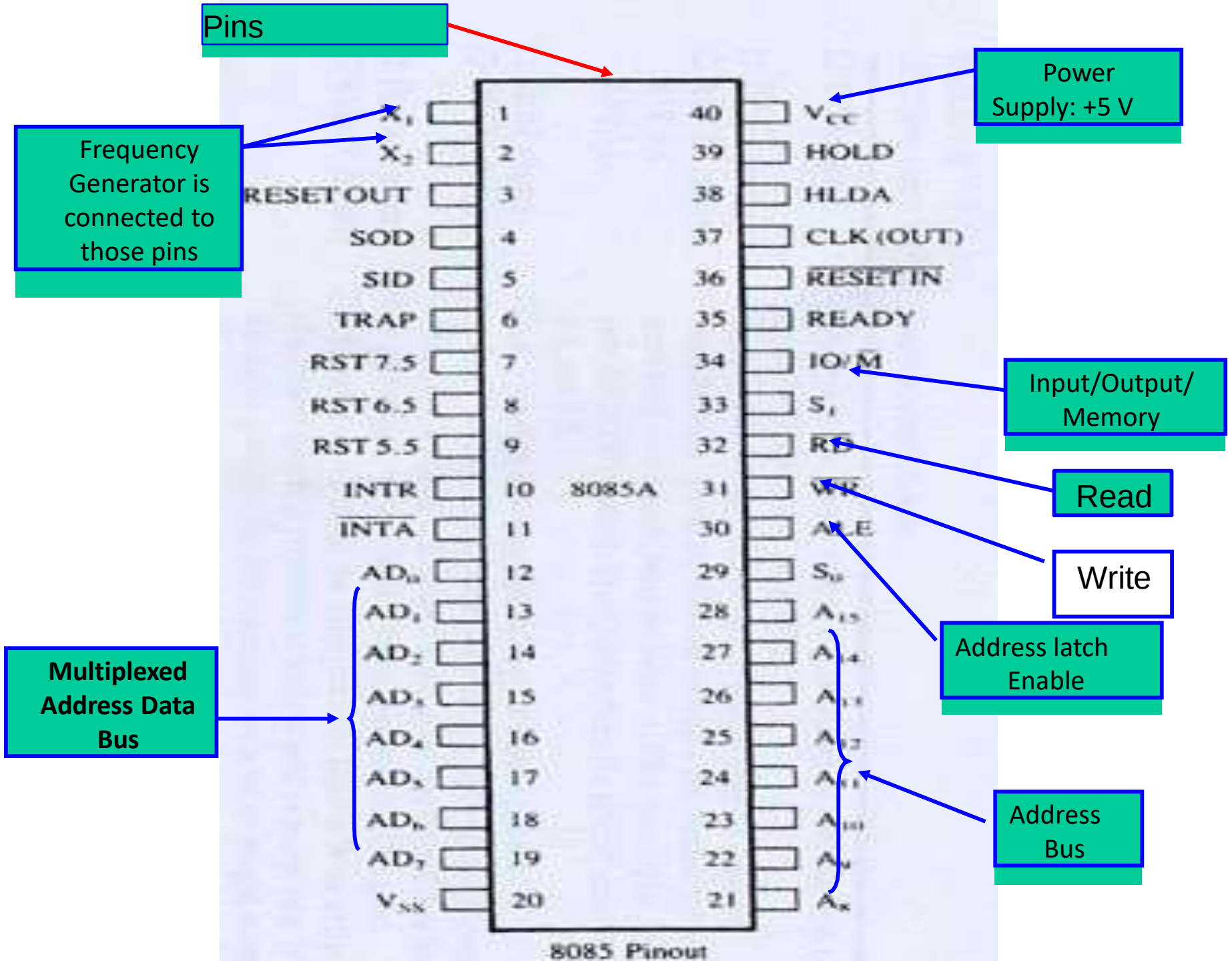
The Read signal indicates that data are being read from the selected I/O or memory device and that they are available on the data bus.

WRITE(WR') (active low output):

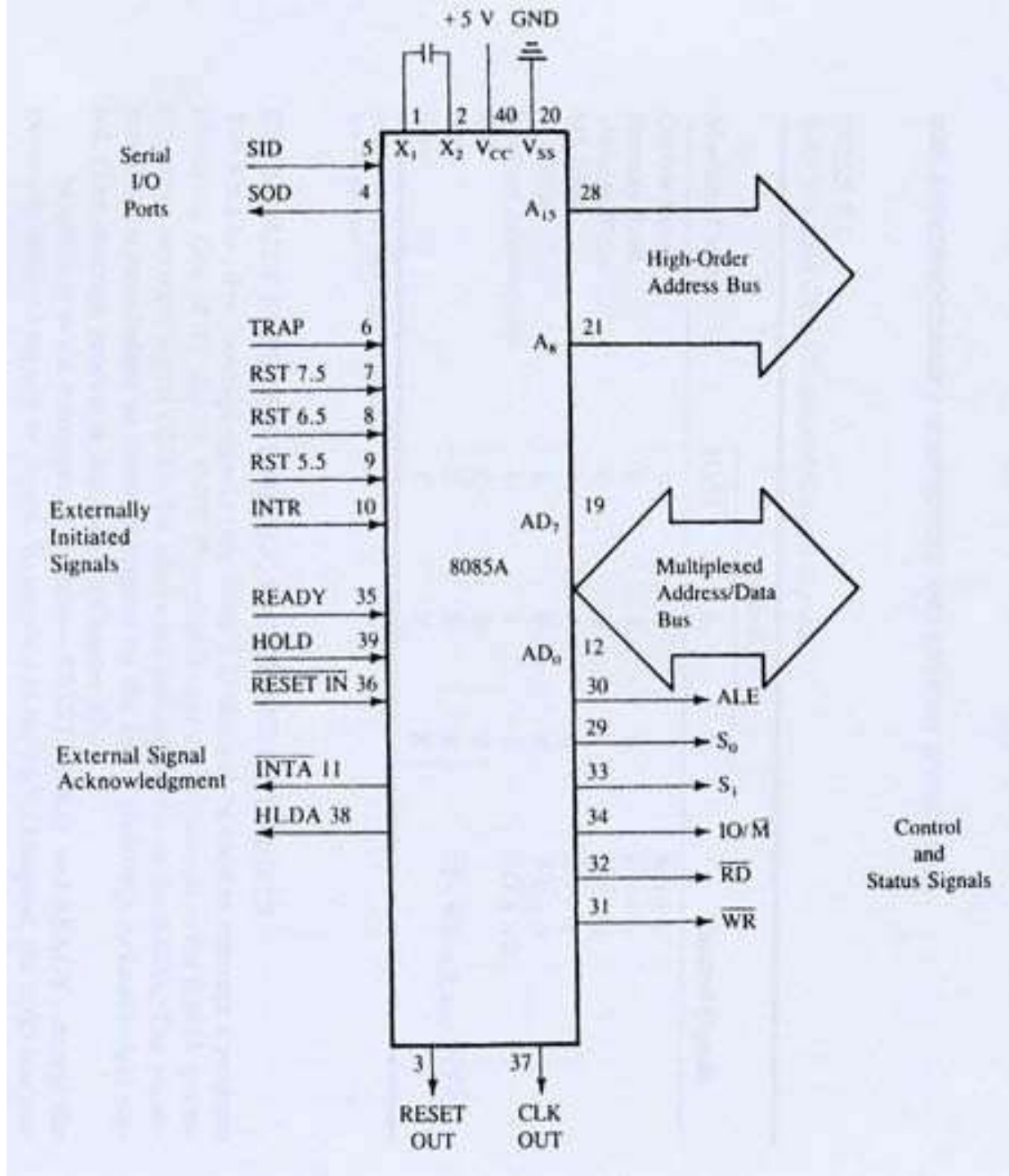
The Write signal indicates that data on the data bus are to be written into a selected memory or I/O location.

IO/M':

It is a signal that distinguishes between a memory operation and an I/O operation. When IO/M= 0 it is a memory operation and IO/M= 1 it is an I/O operation.


Pins
Power Supply: +5 V
Frequency Generator is connected to those pins
Input/Output/ Memory
Read
Write
Address latch Enable
Multiplexed Address Data Bus
Address Bus
X1
X2
RESET OUT
SOD
SID
TRAP
RST 7.5
RST 6.5
RST 5.5
INTR
INTA
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
VSS
VCC
HOLD
HLDA
CLK (OUT)
RESET IN
READY
IO/M
S1
RD
WR
ALE
S0
A15
A14
A13
A12
A11
A10
A9
A8
8085A

8085 Pinout


+5 V
GND
1
2
40
20
X1
X2
VCC
VSS
Serial I/O Ports
SID
5
SOD
4
TRAP
6
RST 7.5
7
RST 6.5
8
RST 5.5
9
Externally Initiated Signals
INTR
10
READY
35
HOLD
39
RESET IN
36
External Signal Acknowledgment
INTA
11
HLDA
38
8085A
A15
28
A8
21
High-Order Address Bus
AD7
19
AD0
12
Multiplexed Address/Data Bus
ALE
30
S0
29
S1
33
IO/M
34
RD
32
WR
31
Control and Status Signals
3
37
RESET OUT
CLK OUT

8085 PIN DESCRIPTION - Continued

S1 and S0 (output): These are status signals used to specify the type of operation being performed;

The microprocessor performs primarily four operations:

I. Memory Read: Reads data (or instruction) from memory.

II. Memory Write: Writes data (or instruction) into memory.

III. I/O Read: Accepts data from input device.

IV. I/O Write: Sends data to output device.

Table1: Status Signal

Operation	S0	S1
Opcode Fetch	1	1
Read	0	1
Write	1	0
Halt	0	0

8085 PIN DESCRIPTION – Key points

Address and Data Buses:

- A8 – A15 (output, 3-state): Most significant eight bits of memory addresses and the eight bits of the I/O addresses.
- AD0 – AD7 (input/output, 3-state): Lower significant bits of memory addresses and the eight bits of the I/O addresses during first clock cycle. Behaves as data bus

Externally Initiated and Interrupt Signals:

- RESETIN': When the signal on this pin is low, the PC is set to 0, the processor is reset.
- RESETOUT': This signal indicates that the processor is being reset. The signal can be used to reset other devices.
- READY: When this signal is low, the processor waits for an integral number of clock cycles until it goes high.

Serial I/O Signals:

- SID: Serial input signal: Bit on this line is loaded to D7 bit of register A using RIM instruction.
- SOD: Serial output signal: Output SOD is set or reset by using SIM instruction.

Interrupts - Basics

- When 8085 receives an interrupt signal from a peripheral device, it stops its current process and the control is transferred to the external device.

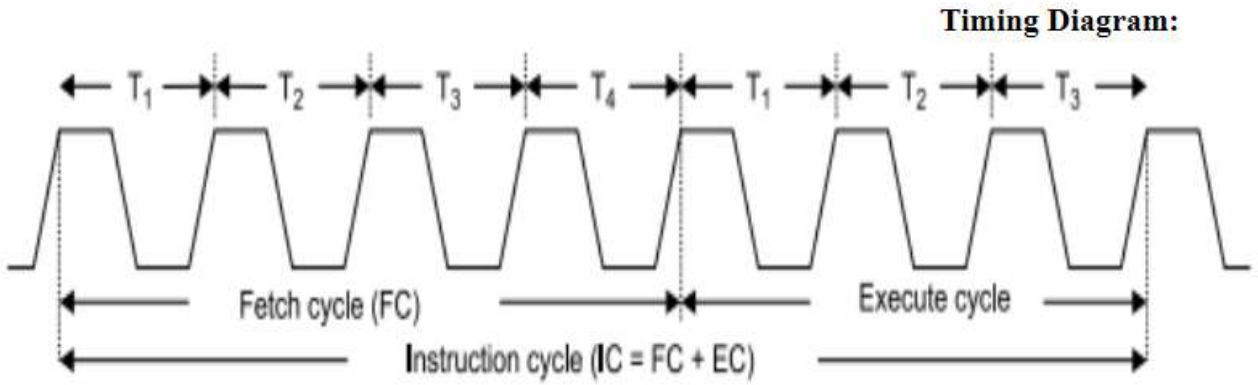
Table2: Interrupts Priority Table

Name	Priority	Address
TRAP	1	0024H
RST 7.5	2	003CH
RST6.5	3	0034H
RST 5.5	4	002CH

Points to remember

- **Types of Interrupts:**
  - Hardware & Software Interrupts
  - Vectored & Non vectored Interrupts
  - Maskable & Non Maskable Interrupts
- **HOLD:** This signal indicates that a peripheral like DMA (direct memory access)
- **HLDA:** This signal acknowledges the HOLD request.
- **INTR:** Interrupt request is a general-purpose interrupt.
- **INTA :** This is used to acknowledge an interrupt.
- **RST** 7.5, RST 6.5, RST 5.5 – restart interrupt: These are vectored interrupts and have highest priority than INTR interrupt
- **TRAP:** This is a non-maskable interrupt and has the highest priority.

Timing Diagram



Fig(8): 8085 Timing Cycle

8085 processor performs

- Opcode fetch
- Operand fetch
- Memory read/write
- I/O read/write

Key Points to remember

- Instruction Cycle: Time taken by the processor to execute an instruction
  It can involve 1 to 6 Machine cycles
  Instruction cycle = Fetch cycle+ Execute cycle
- Machine cycle: Time required to complete one operation. Contains 3 to 6 T-States.
- T-States: Denotes one Clock period

Figure Adapted from: https://www.zseries.in/embedded%20lab/8085%20microprocessor/timing%20diagram.php

Fetching & Exécution Cycles

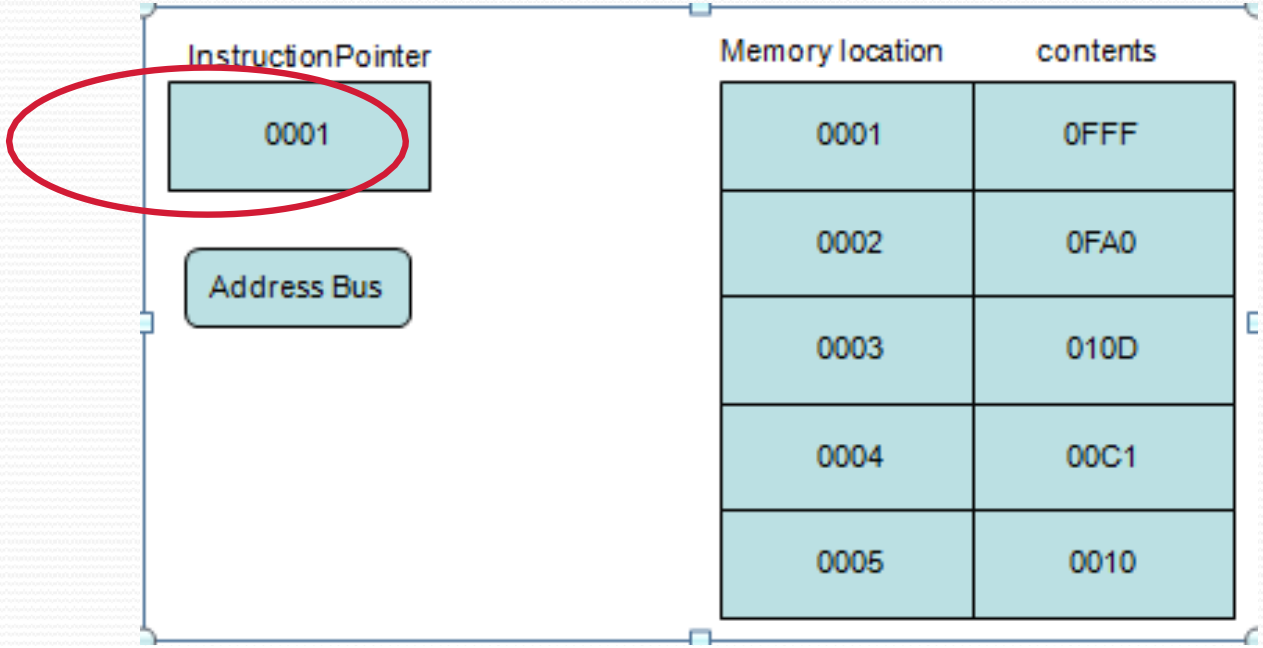
- **Fetching Cycles**
  - The fetch cycle takes the instruction required from memory, stores it in the instruction register, and

    increment the program counter on one so that it points to the next instruction.

- **Execute cycle**
  - The actual actions which occur during the execute cycle of an instruction.
  - Depend on both the instruction itself and the addressing mode specified to be used to access the data that may be required.

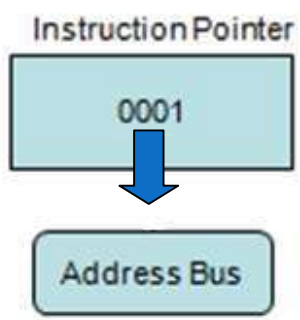
Fetching an instruction

Step 1: Instruction pointer (program counter) hold the address of the next instruction to be fetch.



Fetching an instruction….Cont….

Step 2

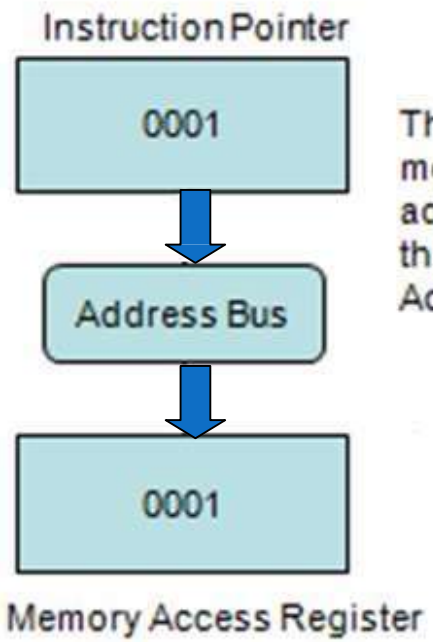


Contents of the Program Counter are passed across the Address Bus

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

Fetching an instruction....Cont....

Step 3

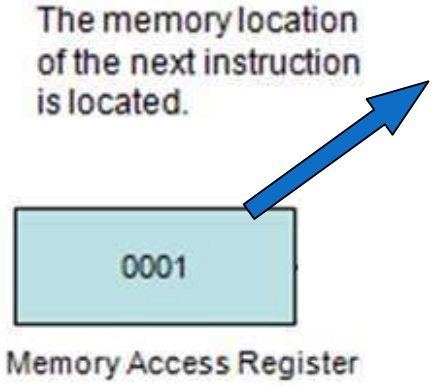


The address moves over the address bus to the Memory Access Register

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

Fetching an instruction....Cont....

Step 4



Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

Fetching an instruction....Cont....

Step 5

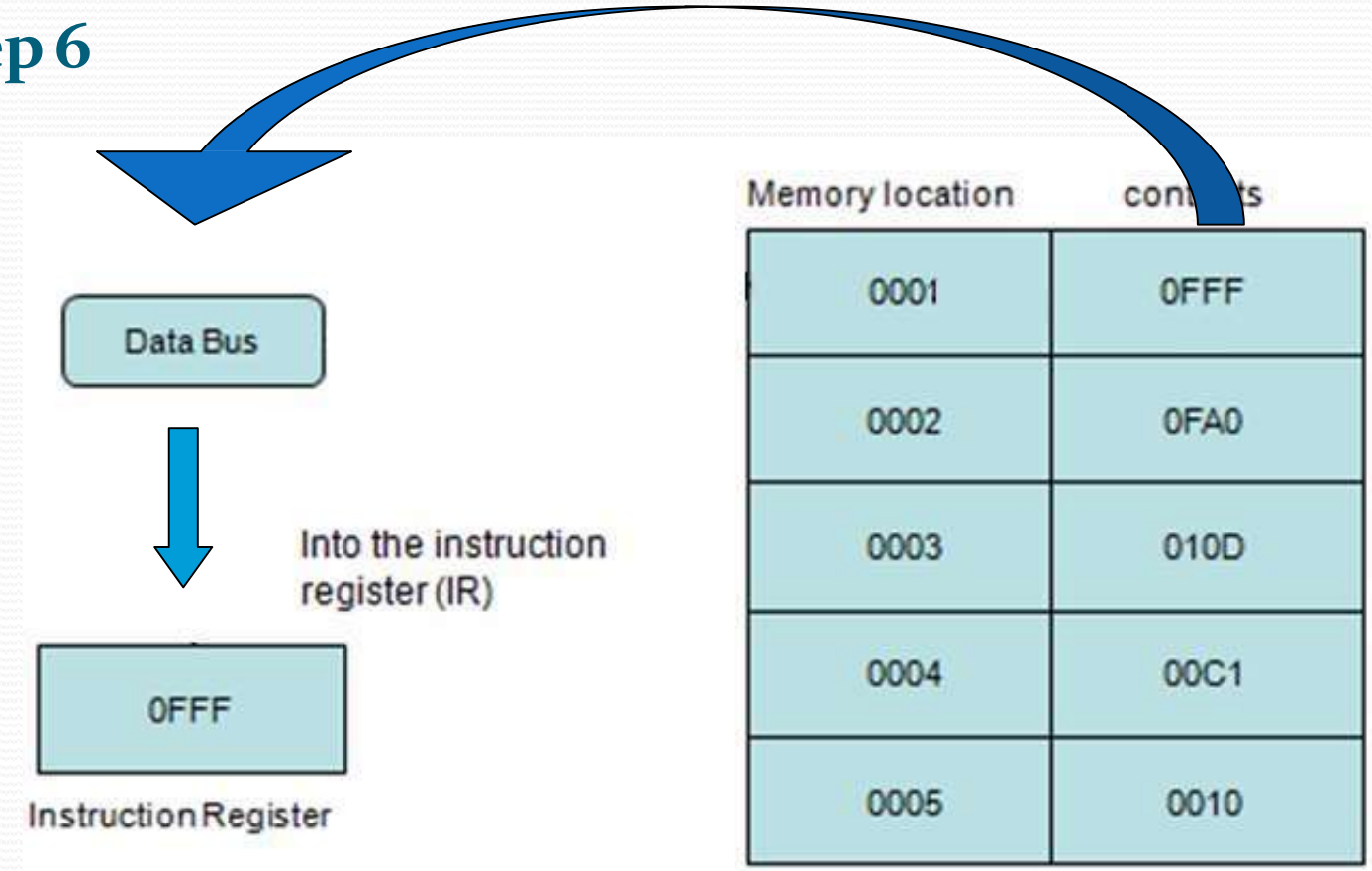
Data Bus

The contents of memory at the given location are moved across the data bus

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

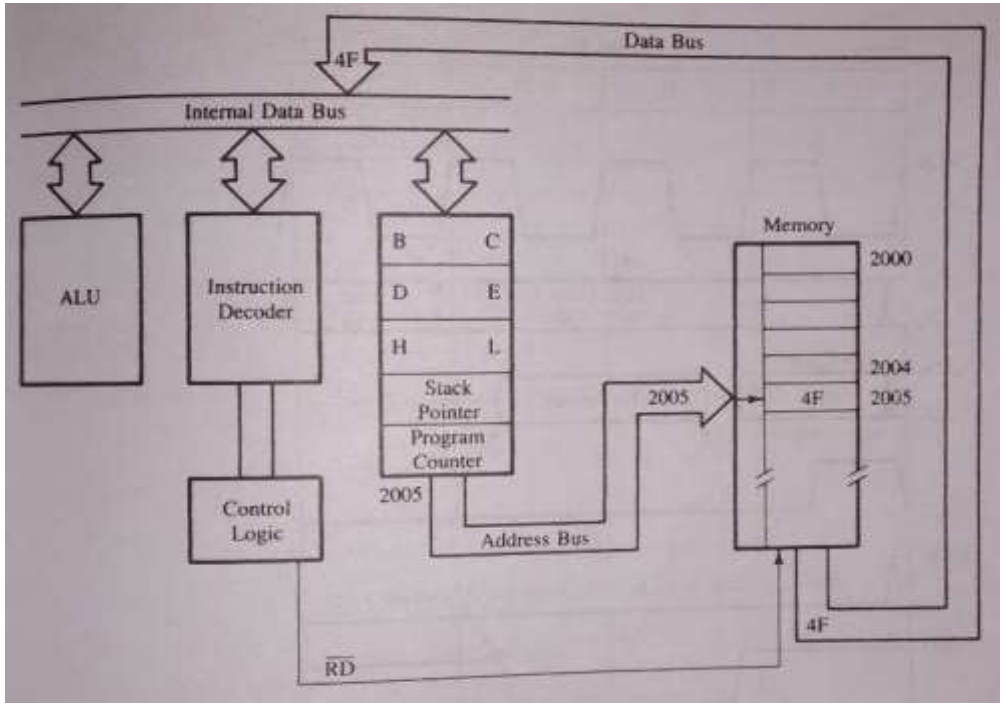
Fetching an instruction....Cont....

Step 6



Data flow from memory to MPU

Steps and data flow, when the instruction code 01001111 (4FH – MOV C,A) Stored in the location 2005H, is being fetch.



Step 1: MPU places the 16 bit memory address from PC on the address bus

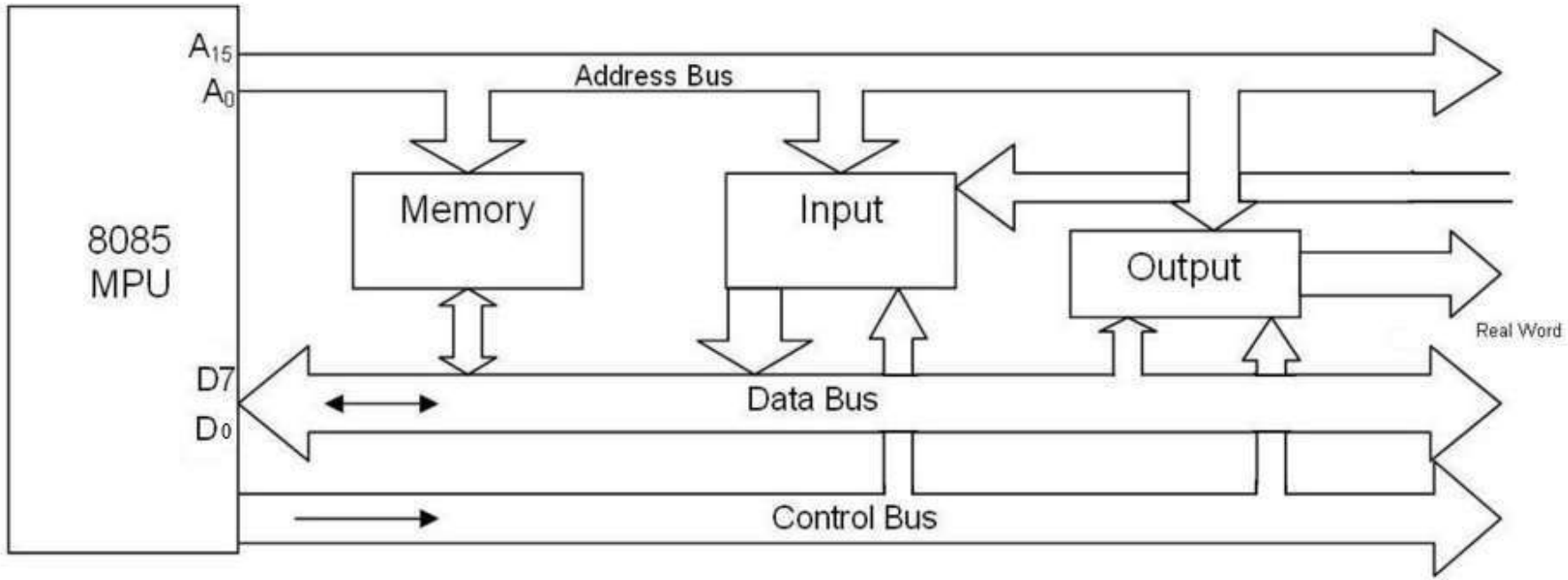
Step 2: Control unit send the signal RD to enable memory chip

Step 3: The byte from the memory location is placed on the data bus.

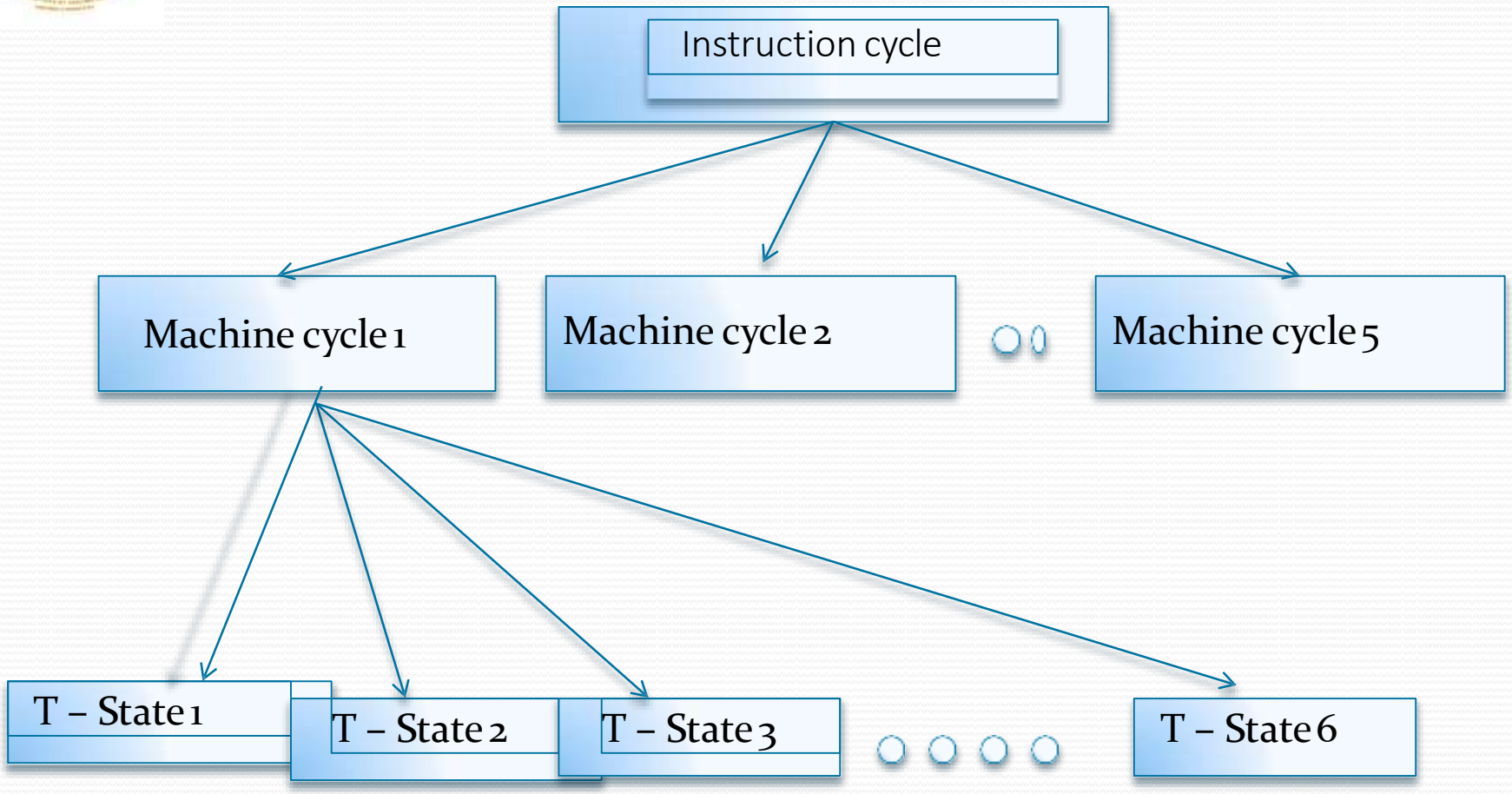
Step 4: The byte is placed on the instruction decoder of the MPU and task is carried out according to the instruction.

Buses Structure

- Various I/O devices and memories are connected to CPU by a group of lines called as bus.



8085 Bus structure


Instruction cycle
Machine cycle 1
Machine cycle 2
Machine cycle 5
T – State 1
T – State 2
T – State 3
T – State 6

The 8085 microprocessor has 7 basic machine cycles. They are

1.Opcode fetch cycle (4T)

2.Memory read cycle or operand fetch(3 T)

3.Memory write cycle (3 T)

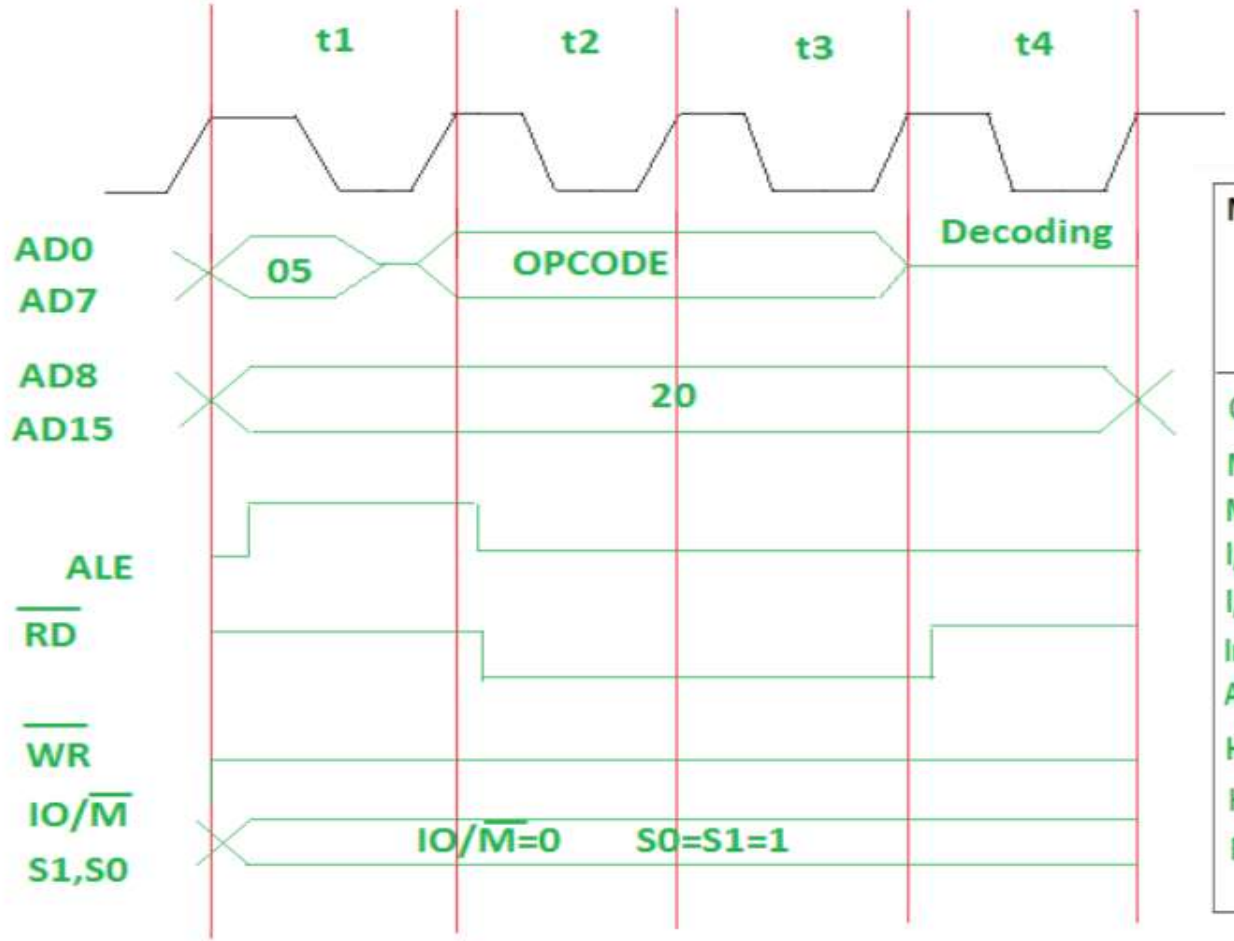
4.I/O read cycle (3 T)

5.I/O write cycle (3 T)

6.Interrupt Acknowledge

7.Bus Idle cycle

Timing Diagram – Opcode Fetch



Timing diagram for opcode fetch

Machine Cycle	Status			Control Signals		
	IO/M	S1	S0	RD	WR	INTA
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/0 Read	1	1	0	0	1	1
I/O Write	1	0	1	1	0	1
Interrupt Acknowledge	1	1	1	1	1	0
HALT	Z	0	0	Z	Z	1
HOLD	Z	X	X	Z	Z	1
RESET	Z	X	X	Z	Z	1

Fig(9): Opcode Fetch Timing Cycle

(Fig-9) Image & Table Adopted from: https://www.geeksforgeeks.org/instruction-cycle-8085-microprocessor

Timing: Transfer of byte from memory to MPU

- **How a data byte is transfer from memory to the MPU.**
- **It shows the five different group of signals with clock**

Step 1: At T1 higher order memory address 20H is placed on the A15 – A8 and the lower order memory address 05H is placed on the bus AD7-AD0, and ALE signal high. IO/M goes low(memory related signal).

Step 2: During T2 RD $\overline{\text{signal}}$ is sent out. $\overline{\text{RD}}$ is active during two clock periods.

Step 3 : During T3, Memory is enabled then instruction byte 4FH is placed on the data bus and transferred to MPU. When RD goes high it causes the bus to go into high impedance state.

Step 4: During T4, the machine code or byte is decoded by the instruction decoder and content of A is copied into register.

Example: MOV C,A
Address:2005 , Opcode = 4F

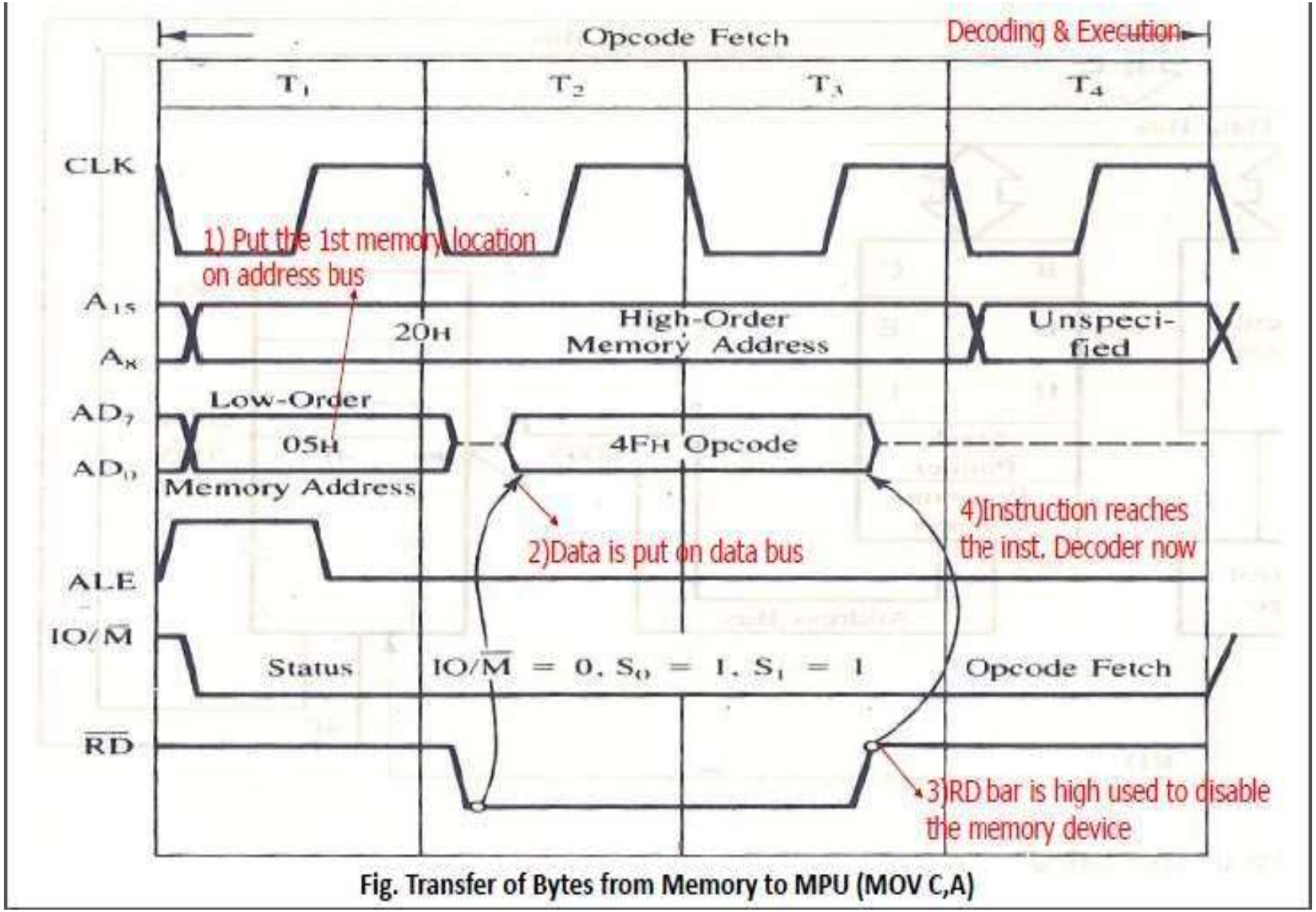
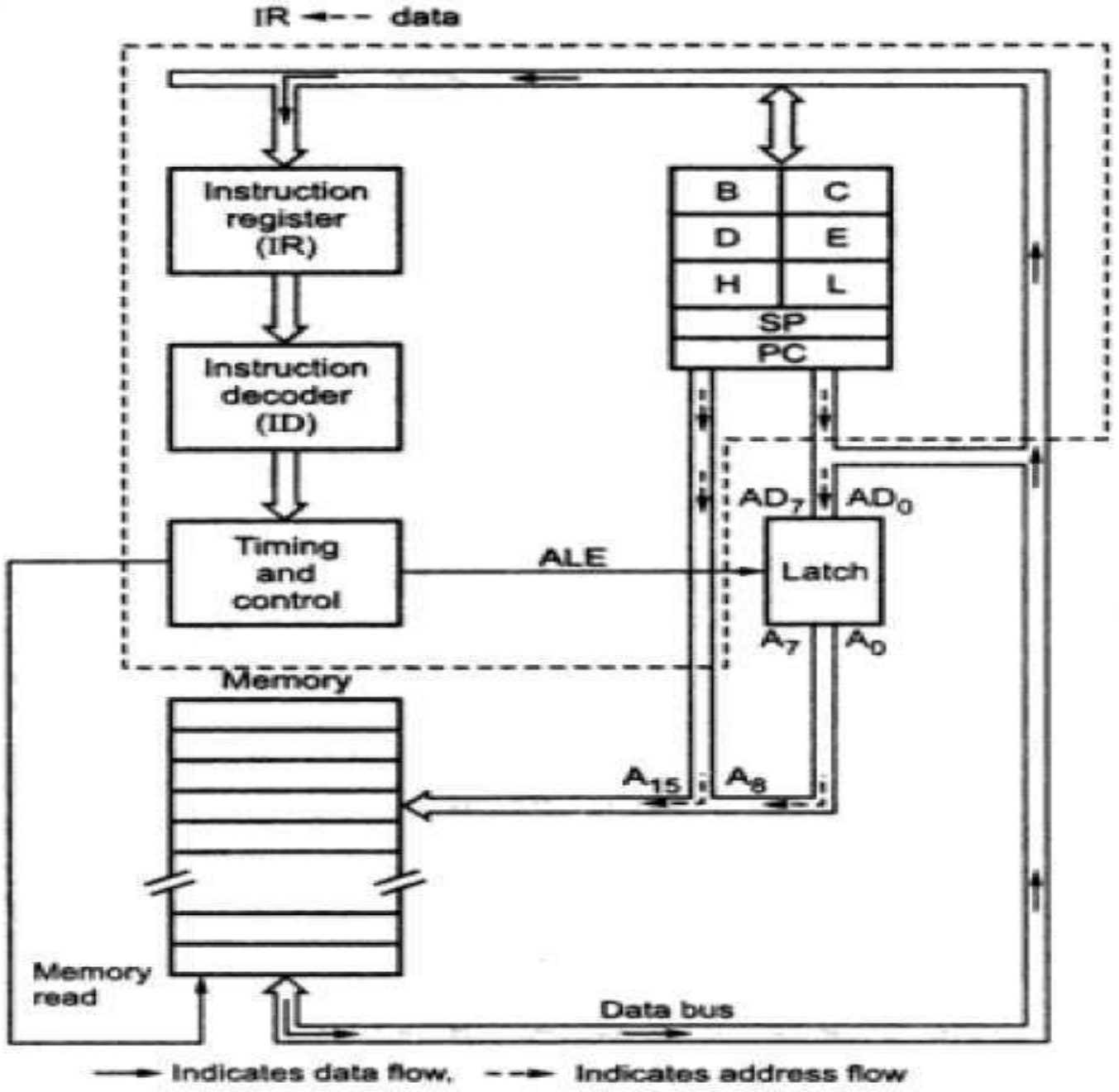


Fig. Transfer of Bytes from Memory to MPU (MOV C,A)

Data Flow from Memory to MP

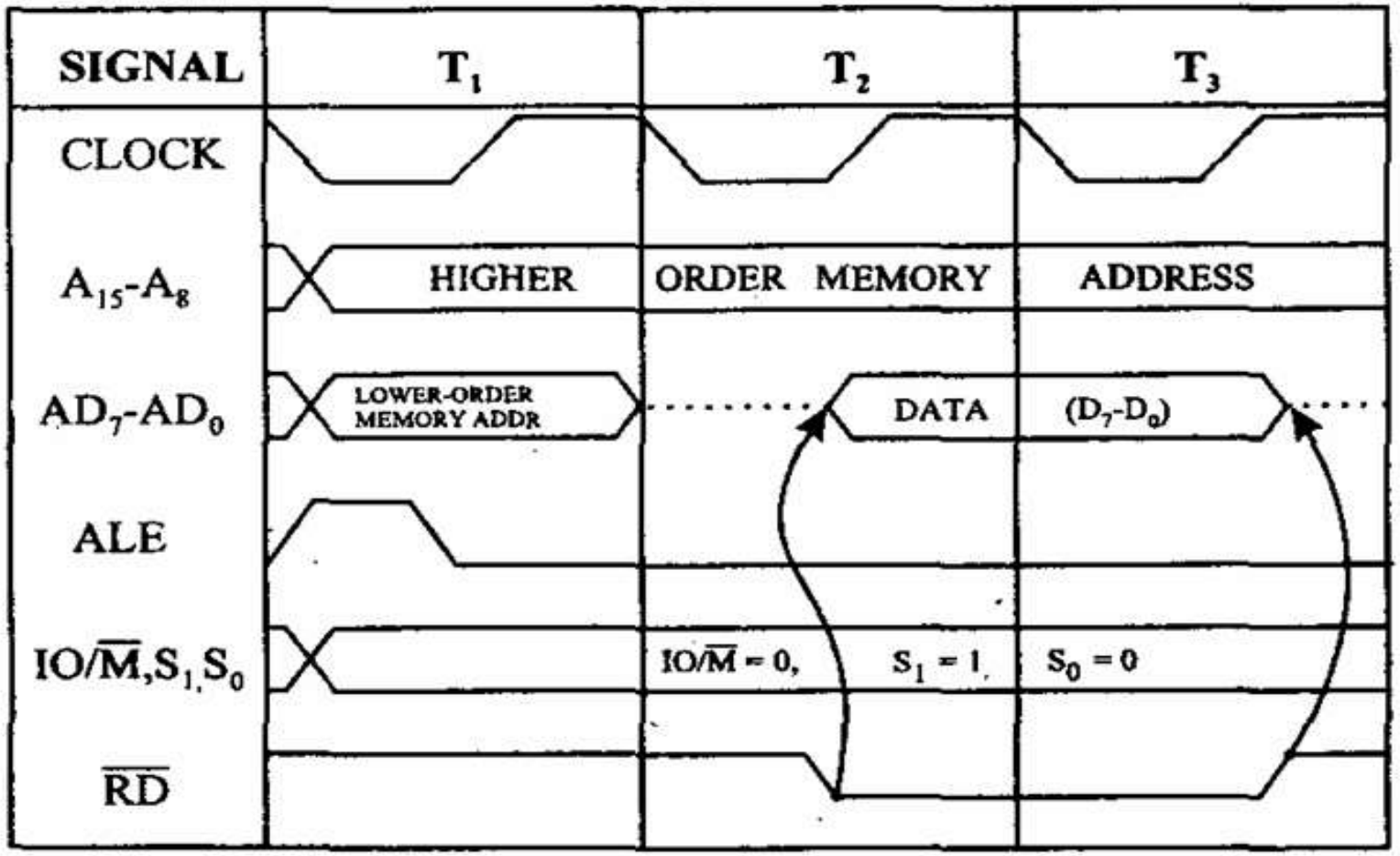


Memory Read or Operand Fetch Machine Cycle

- This cycle is executed by the processor to read a data byte from memory or to fetch operand in a multi byte instruction. For ex. 2 or 3 byte instruction because in 1 byte instruction the machine code is an opcode; so operation is always an opcode fetch
- The instructions which have more than one byte word size will use the machine cycle after the opcode fetch machine cycle.
- The memory read machine cycle is exactly the same as the opcode fetch except:
  - **IO/M= 0(memory operation),**
  - **s1 = 1 and so = 0. (memory read)**
  - **WR = 1 & RD = 0**
  - It only has 3 T-states
- First cycle is opcode fetch cycle.
- So this cycle requires

  4T(opcode) +3T(memory read)=7 T states to execute.

Timing Diagram – Memory Read



Key points:

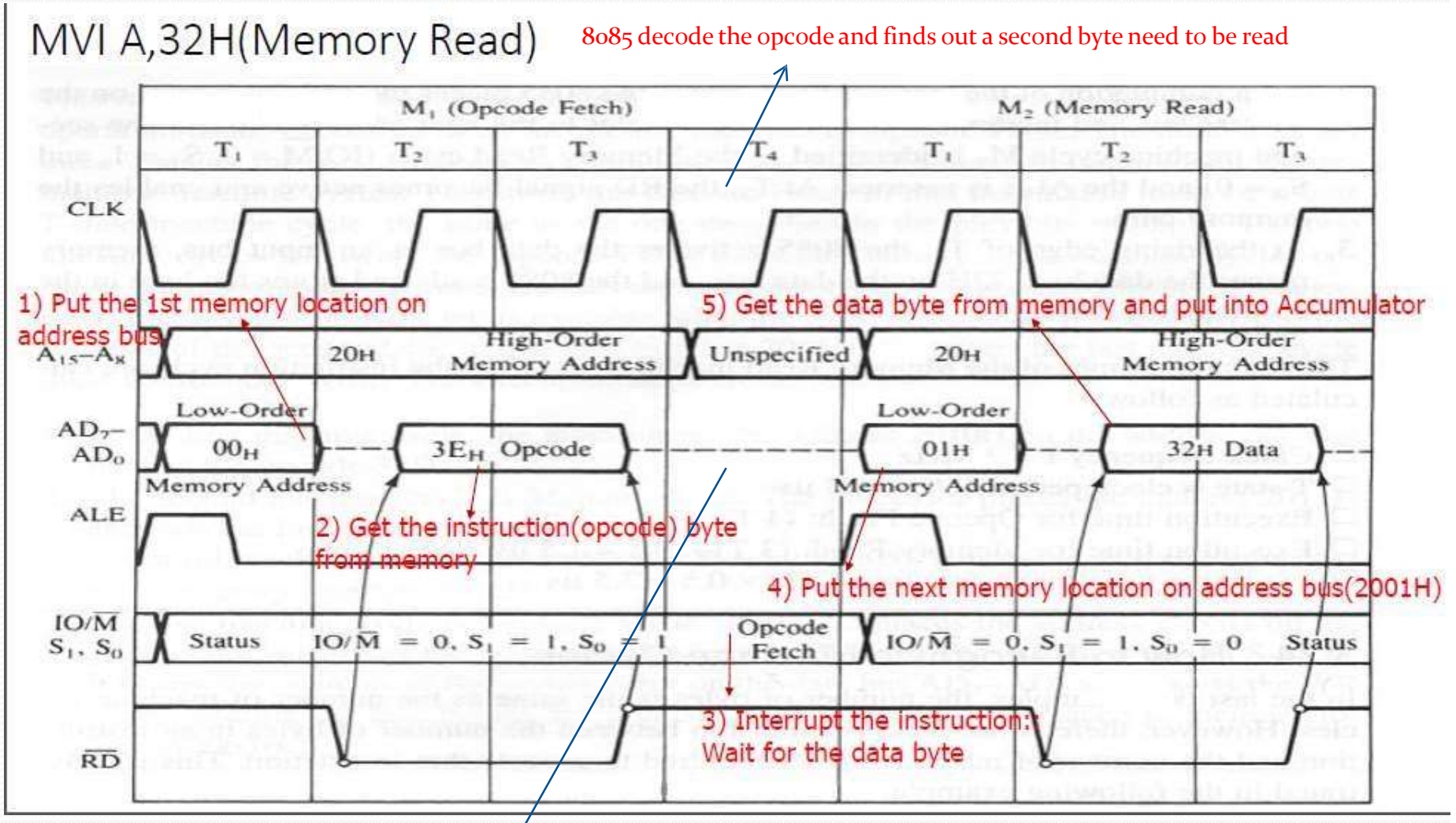
- Fetches one byte from memory
- It has 3 T states
- Used to fetch opcode or operand from the memory

Fig(10): Memory Read Timing Diagram

Image Adapted from: https://www.zseries.in/embedded%20lab/8085%20microprocessor/timing%20diagram.php

Timing for execution of the instruction MVI A, 32H

MVI A, 32H is 2 byte instruction, so hex code for MVI A is 3E.



The execution times of the memory read machine cycle and the instruction cycle are calculated as

Clock Frequency f = 2MHz

T state = clock period (1/f) = 0.5µsec

Execution time for opcode fetch = 4T * 0.5 = 2µsec

Execution time for Memory Read = 3T * 0.5 = 1.5µsec

Execution time for Instruction = 7T * 0.5 = 3.5µsec

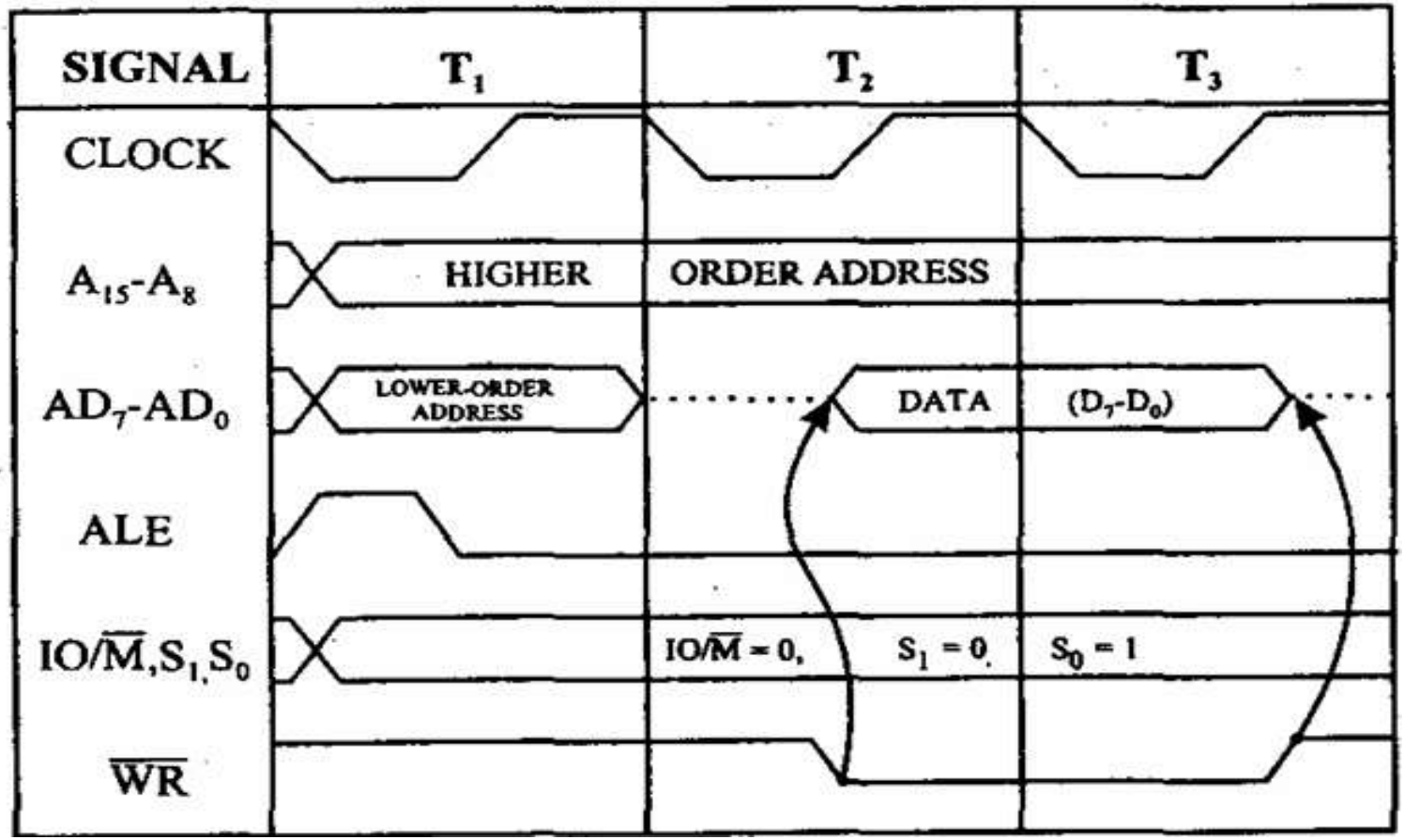
Memory Write Machine Cycle

- The memory write machine cycle is executed by the processor to write a data byte in a memory location.
- The memory write machine cycle is exactly the same as the memory read except:
  - **IO/M= o(memory operation),**
  - **s1 = o and so = 1. (memory read**
  - **WR = o & RD = 1**
  - It only has 3 T-states

- First cycle is opcode fetch cycle.
- So this cycle requires

  4T(opcode) +3T(memory write)=7 T states to execute.

Timing Diagram – Memory write



Fig(11): Memory Write Timing Diagram

Key points:

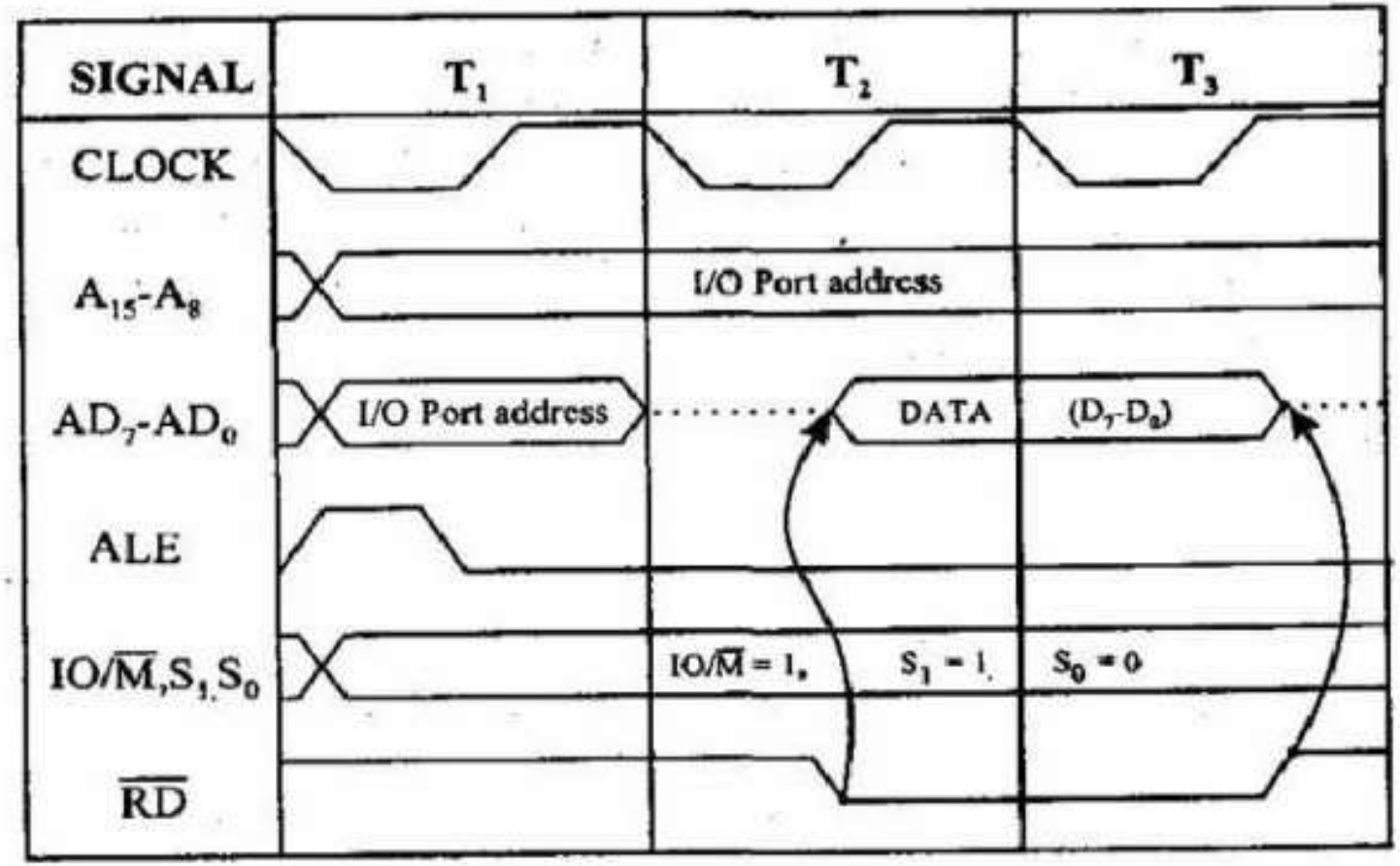
- Used to send one byte into memory
- It has 3 T states

Image Adapted from:
https://www.zseries.in/embedded%20lab/8085%20microprocessor/timing%20diagram.php

Input/ Output Read Machine Cycle

- Microprocessor executes this cycle to read content of I/O port or to read 8 bit data present on an input ports through data bus.
- The I/O read machine cycle is exactly the same as the memory read except:
  - **IO/M= 1(I/O operation),**
  - **s0 = 0 and s1 = 1. (read)**
  - **~~WR~~ = 1 & RD = 0**
  - **It only has 3 T-states**
- This instruction reads the data from an input device and places the data byte in the accumulator.
- It accept the data from input device by using I/O read signal(IOR).

Timing Diagram – I/O Read



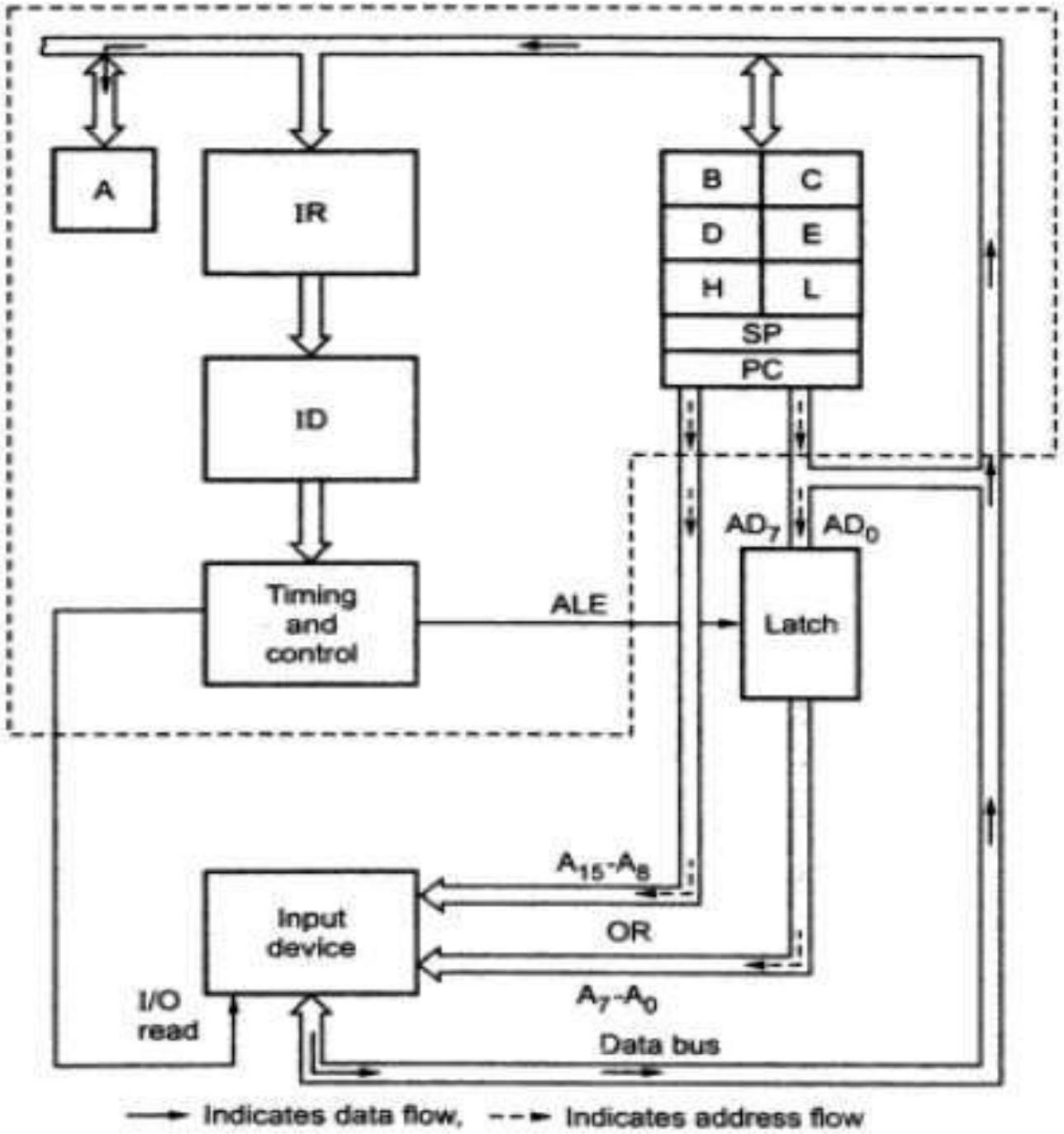
Fig(12): I/O Read Timing Diagram

Key points:

- Fetches one byte from I/O Port
- It has 3 T states

Image Adapted from:
https://www.zseries.in/embedded%20lab/8085%20microprocessor/timing%20diagram.php

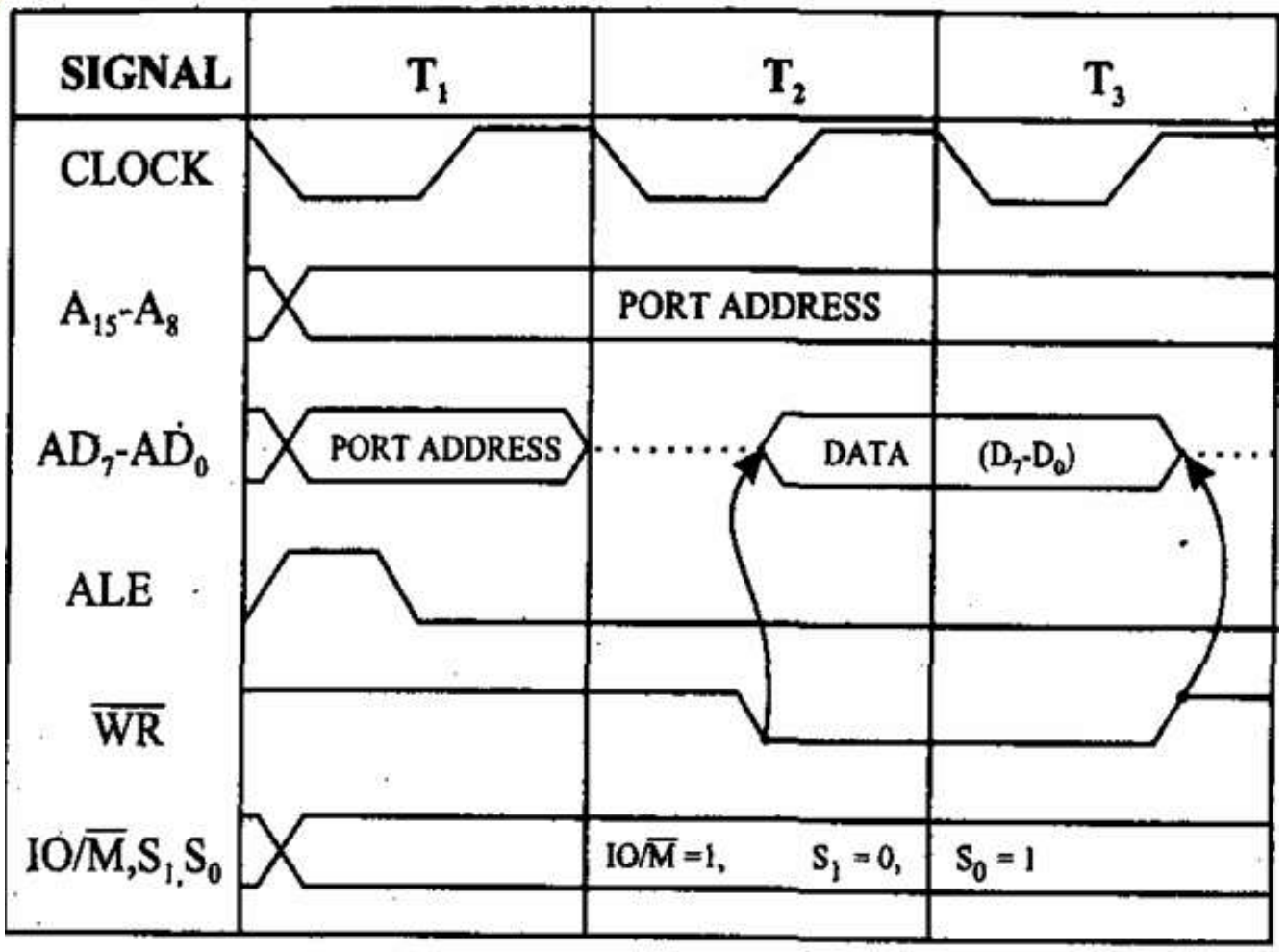
Data Flow form input device to MP



Input/output write Machine Cycle

- Microprocessor executes this cycle to write a data into an I/O port or to write 8 bit data present on an output ports.
- The I/O write machine cycle is exactly the same as the memory write except:
  - **IO/M= 1(I/O operation),**
  - **S1 = 1 and S0 = 0. (write)**
  - **WR = 0 & RD = 1**
  - It only has 3 T-states
- This instruction places the content of the accumulator on the data bus.
- It transmit the data to output device by using I/O write signal(IOW).

Timing Diagram – I/O Write



Key points:

- Used to write one byte into IO device
- It has 3 T states

Fig(13): I/O Read Timing Diagram

Image Adapted from: https://www.zseries.in/embedded%20lab/8085%20microprocessor/timing%20diagram.php

Timing Diagram Examples

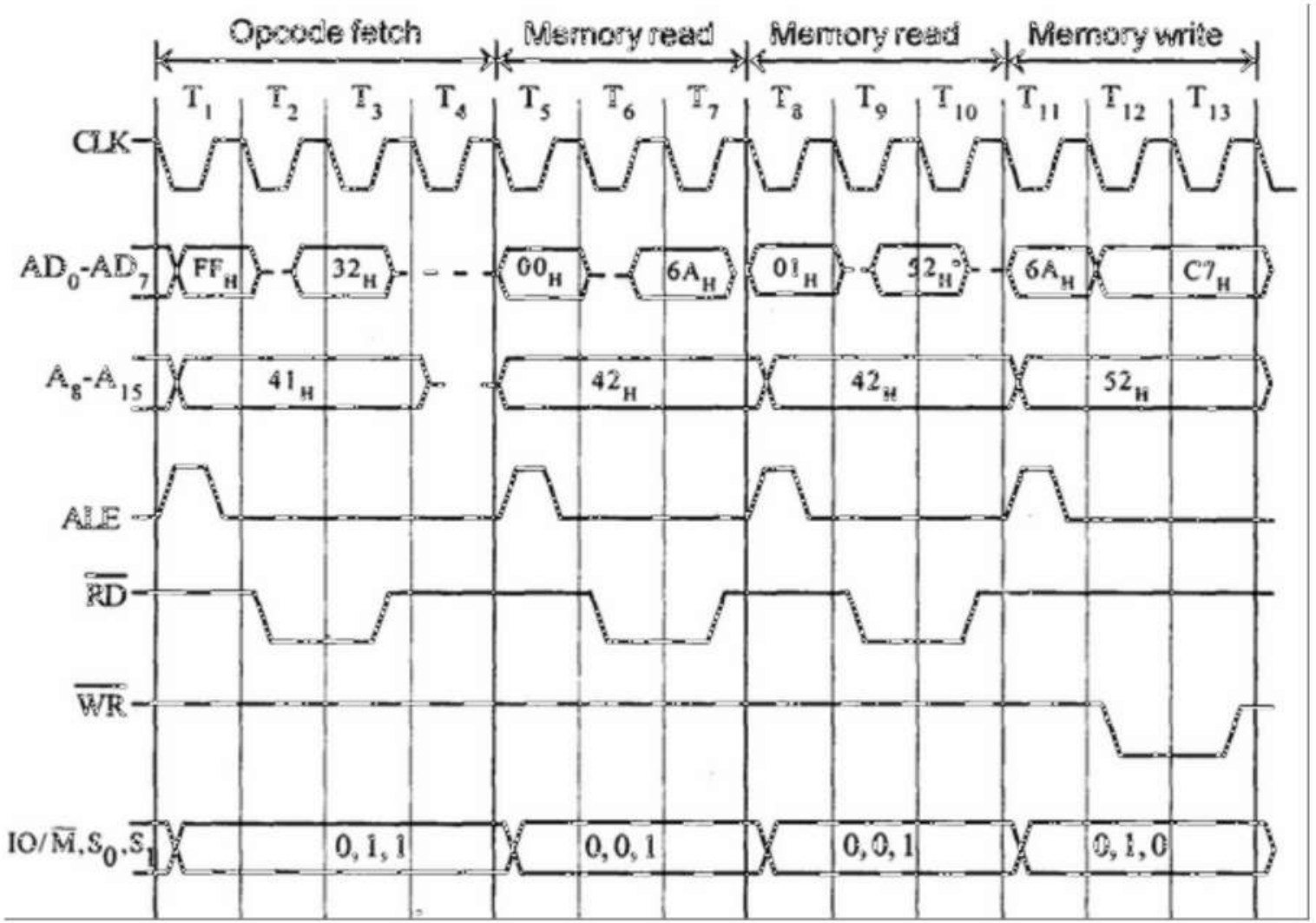
STA 526A

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). - *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.

Address	Mnemonics	Opcode
41FF	STA 526A$_H$	32$_H$
4200		6A$_H$
4201		52$_H$

STA....


Opcode fetch
Memory read
Memory read
Memory write
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13
CLK
AD0-AD7
FFH 32H 00H 6AH 01H 52H 6AH C7H
A8-A15
41H 42H 42H 52H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
0, 0, 1
0, 1, 0

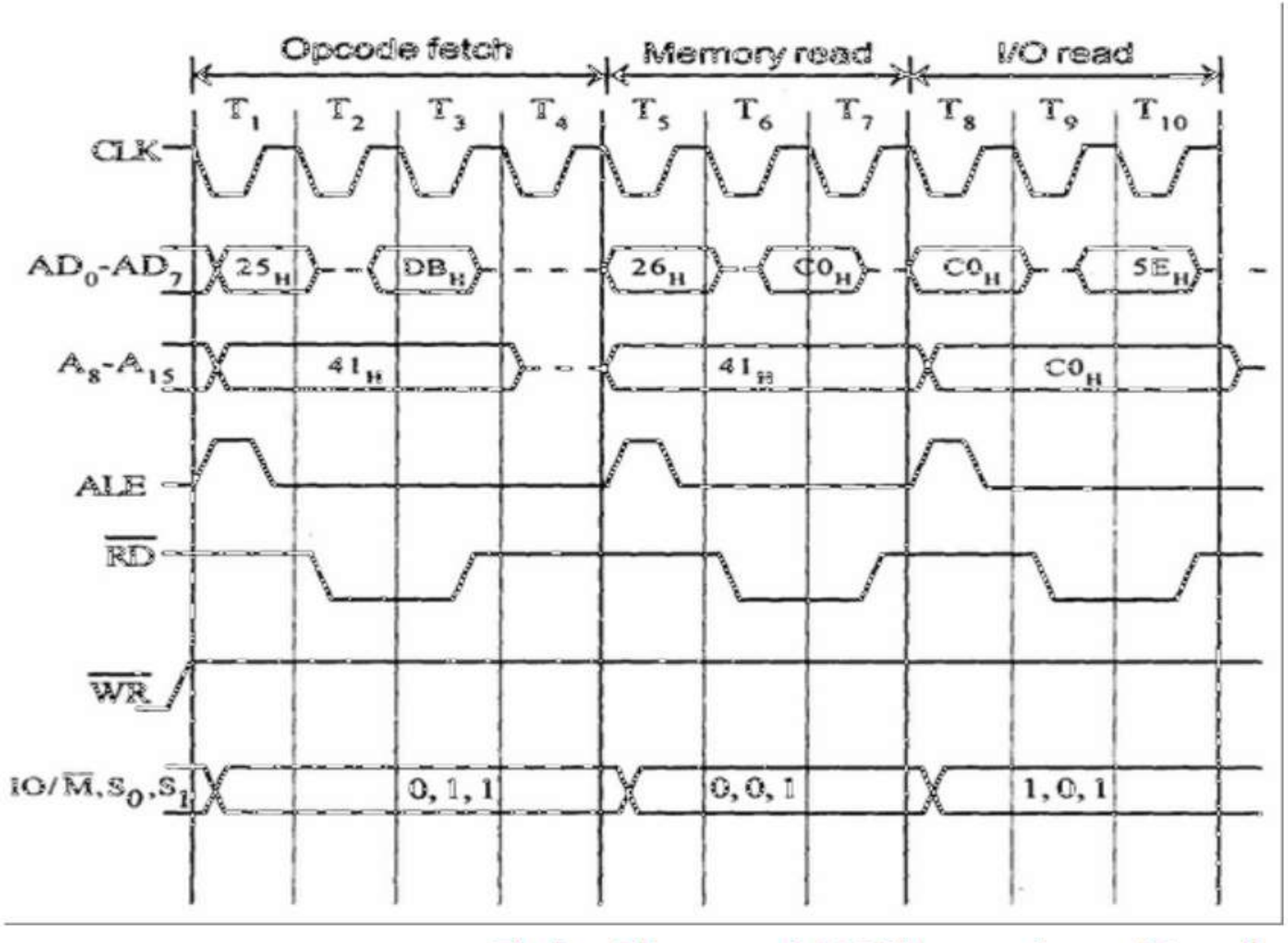
IN C0

Timing diagram for IN C0H.

- Fetching the Opcode DBH from the memory 4125H.
- Read the port address C0H from 4126H.
- Read the content of port C0H and send it to the accumulator.
- Let the content of port is 5EH.

Address	Mnemonics	Opcode
4125	IN $C0_H$	DB_H
4126		$C0_H$

IN...


Opcode fetch
Memory read
I/O read
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
CLK
AD0-AD7
25H
DBH
26H
C0H
C0H
5EH
A8-A15
41H
41H
C0H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
1, 0, 1

INR M

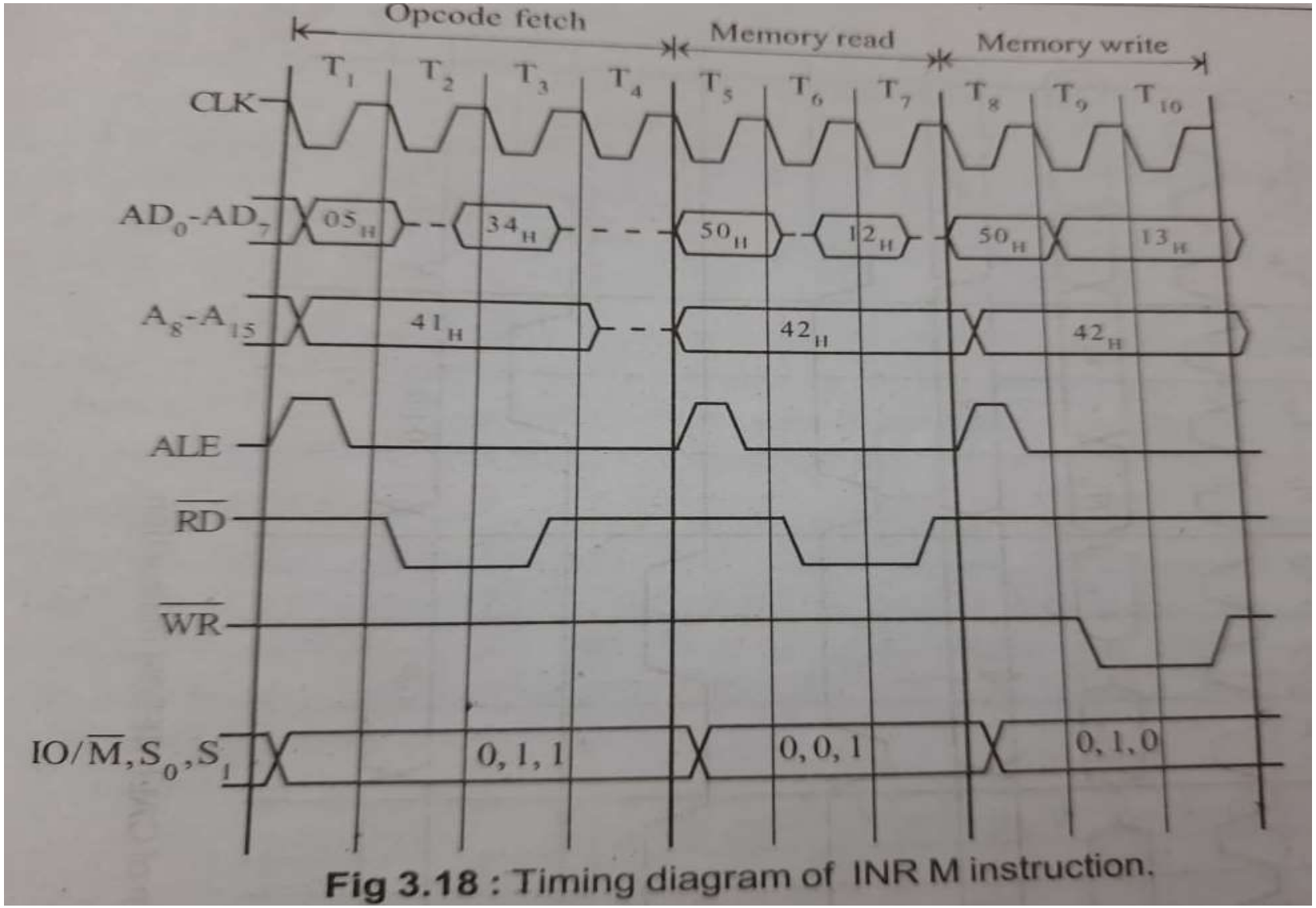


Fig 3.18 : Timing diagram of INR M instruction.

INR

Opcode	Operand	Description
INR	R M	Increment register or memory by 1

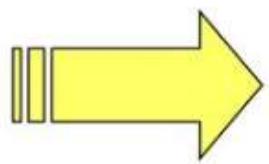
- The contents of register or memory location are incremented by 1.
- The result is stored in the same place.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- **Example:** INR B or INR M

Example

Instruction: **INR E**

Register contents before Execution

E 1C $_h$



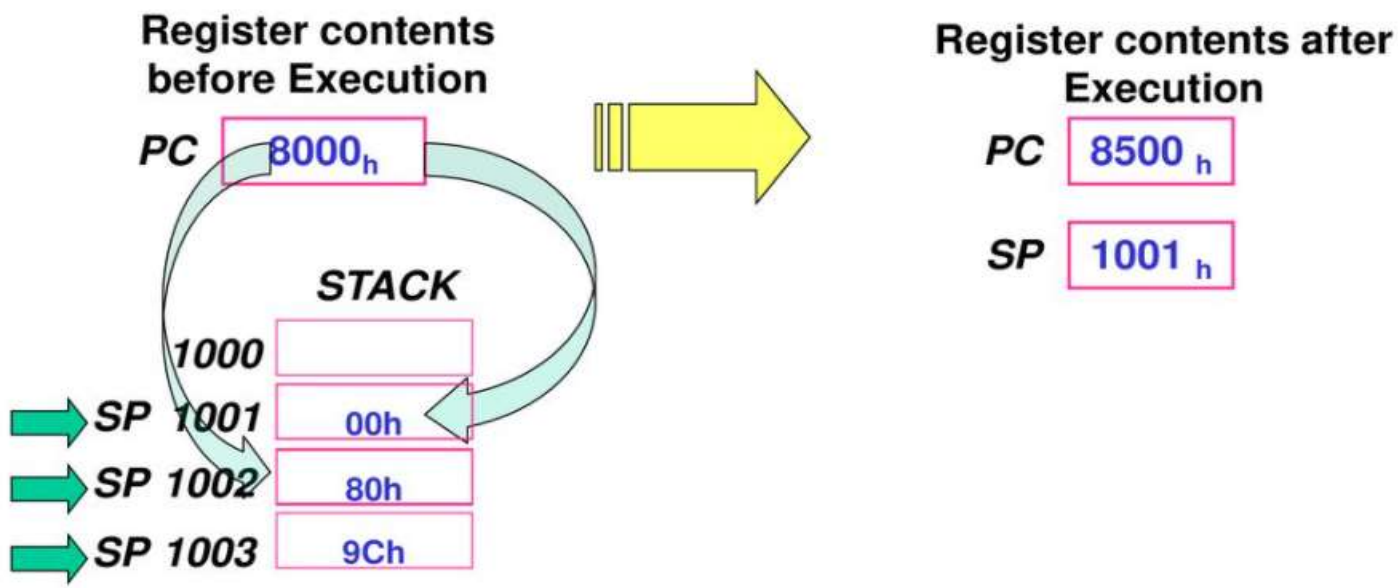
Register contents after Execution

E 1D $_h$

Note: Except Carry Flag, all flags are affected depend upon the result.

Example

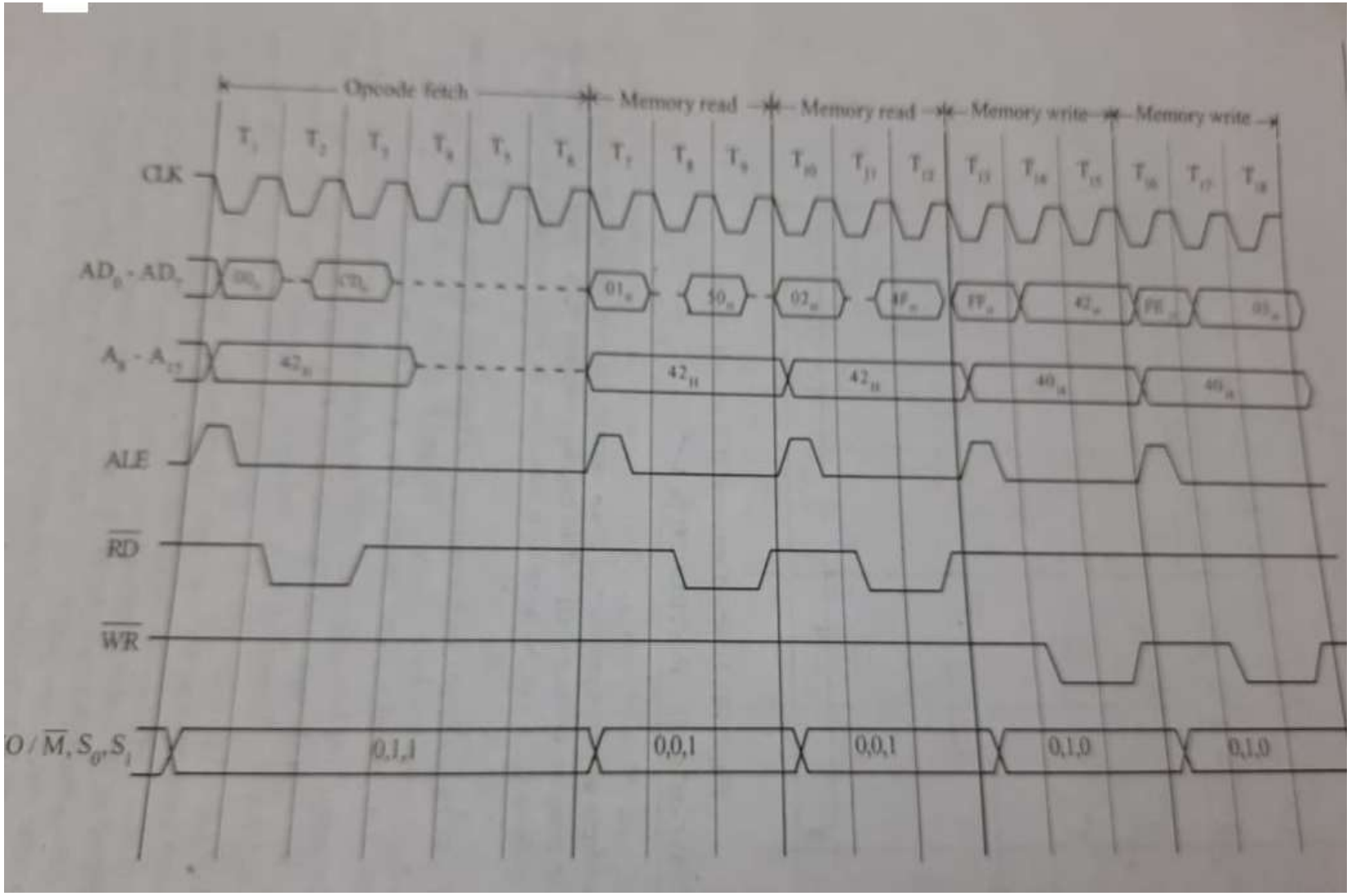
Instruction: **CALL 8500h**



Note: No Flags are affected.

CALL 4F50

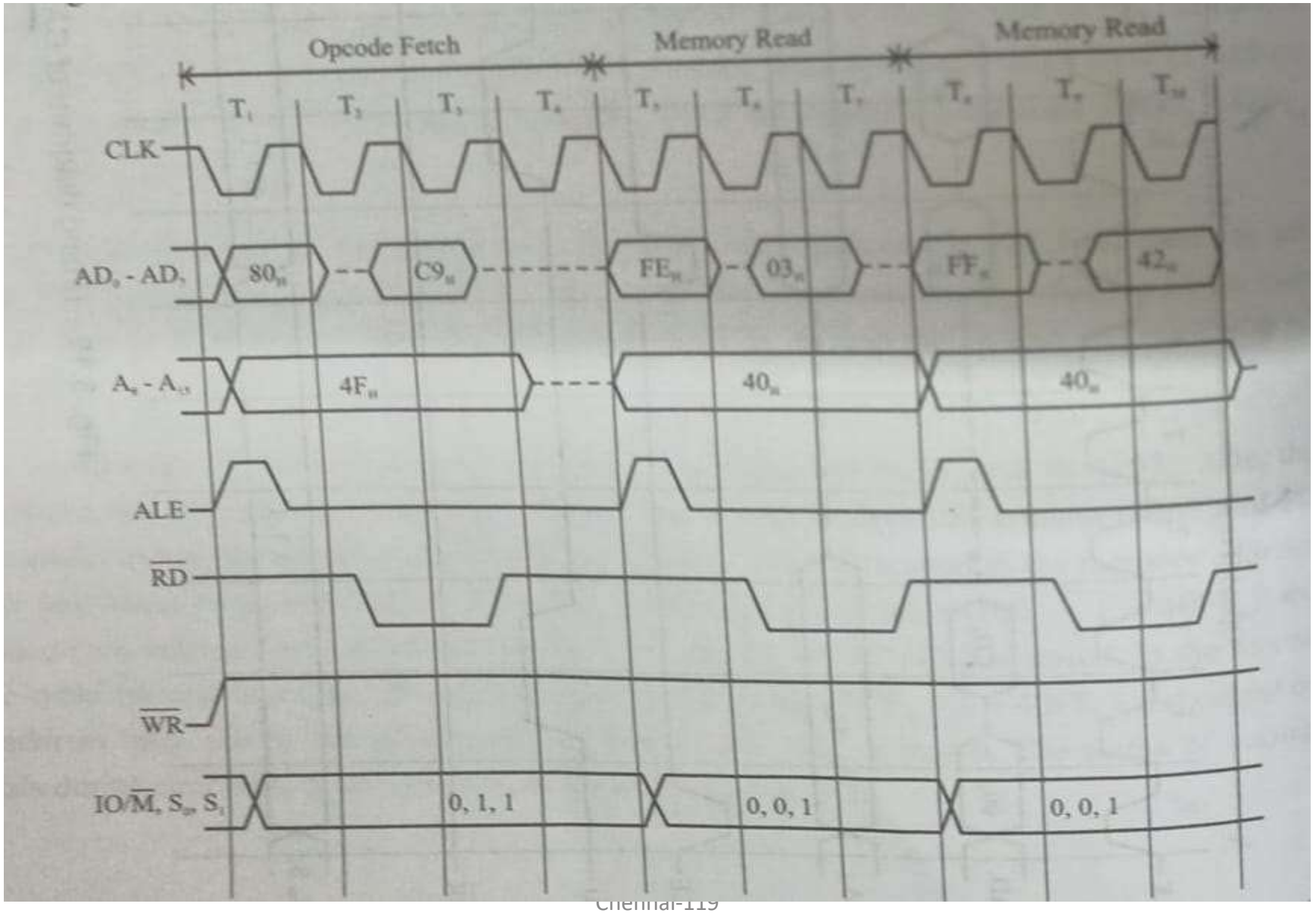
Call 4F50


Opcode fetch
Memory read
Memory read
Memory write
Memory write
CLK
ALE
$\overline{RD}$
$\overline{WR}$
$IO/\overline{M}, S_0, S_1$
0,1,1
0,0,1
0,0,1
0,1,0
0,1,0

PUSH B


OPCODE FETCH CYCLE WITH 6 T-STATES
MEMORY WRITE CYCLE
MEMORY WRITE CYCLE
T1
T2
T3
T4
T5
T6
T1
T2
T3
T1
T2
T3
CLK
A15-A8
20 High-Order Memory Address
Decodes Opcode & the next memory address
3F High-Order Memory Address
3F High-Order Memory Address
AD7-AD0
06 Low-Order Memory Address
Opcode Read C5
FF Low-Order Memory Address
Opcode Write AA
FE Low-Order Memory Address
Opcode Write BB
ALE
ALE = 1
ALE = 0
ALE = 1
ALE = 0
ALE = 1
ALE = 0
IO/M
IO/M = 0
IO/M = 0
IO/M = 0
S1
S1 = 1
S1 = 0
S1 = 0
S0
S0 = 1
S0 = 1
S0 = 1
RD
RD = 0
RD = 1
RD = 1
WR
WR = 1
WR = 0
WR = 0

RET Instruction



Addressing modes for 8085:

The method of specifying the data or operand to be operated.

Types:

- Immediate Addressing Mode
- Register Addressing Mode
- Direct Addressing Mode
- Indirect Addressing Mode /(Register Indirect Addressing Mode)
- Implied Addressing Mode

Addressing modes for 8085:

Indirect Addressing Mode:

- The content of the register is used to specify the address of the operand

Ex: **MOV A, M** →Move the content of the memory location whose address is given in H and L register in the accumulator

ADD M →Addition of the content of the memory location whose address is given in H and L register and in the accumulator

Implied Addressing Mode:

- Opcode specifies the address of the operand

EX: **CMA** → Complements the content of the accumulator

RAL→ Rotate the content of the accumulator left through carry

Addressing modes for 8085:

Immediate Addressing Mode:

- The data is present in the instruction itself is called Immediate Addressing Mode.

Ex: **MVI A, 03H-**→The data 03 is moved to accumulator.

ADI 03H →The data 03 is added immediately to the accumulator.

Register Addressing Mode:

- The data is moved from one register to another register is called Register Addressing Mode .

Ex: **MOV A, B** → The data is moved from one register to another register

ADD C →The data present in C register is added with data present in accumulator and saved in accumulator.

Direct Addressing Mode:

- The address of the operand always exist with in the instruction.

Ex: **LDA 4500H** →Load the content of the memory location 4500 in to the accumulator

STA 4600H →Store the content of the accumulator in the memory location 4600

Tips & Tricks

- Read the question carefully
- Understand the Marks weightage for each question
- Draw the diagrams clearly
- Stop Mugging and try to understand the basic concepts
- Note down the key points and whenever possible recollect it
- Don’t fear about Failure, Just try your Best
- Present neatly

Model Questions
(Previous years University questions shown as samples)

Unit -1 / Part-B

1. Draw the Block diagram of 8085 and explain Each block. (Sathyabama – BE-ECE/EEE/P-EEE -SEP 2018)
2. Explain the Architecture of 8085 MPU with its functional Block Diagram.(Sathyabama – ECEEEE/NOV-2018, MAY 2015)
3. Explain the addressing modes & its types. (Sathyabama – ECEEEE/NOV-2018, MAY 2015)
4. Draw and Describe the timing Diagram for IO Read and Memory Write cycle diagram and its input, output waveforms. (Sathyabama – BE-CSE/IT-MAY2018)
5. Illustrate the Timing and Control circuitry for 8085 microprocessor.

Model Questions
(Previous years University questions shown as samples)

Unit -1 / Part-A

1. Write register Indirect addressing mode with example for 8085 processor. (Sathyabama – CSE/IT/ECE/EEE/MAY-2018,)
2. Give the format of Flag register.(Sathyabama – CSE/ECE/EEE/MAY-2018,)
3. Explain the SID & SOP pins. (Sathyabama – ECEEEE/ MAY 2015)
4. What are the Interrupts available in 8085. (Sathyabama – BE-CSE/IT/ ECE-SEP-2018)
5. Differentiate microprocessor and microcontrollers.
6. Define Vectored Interrupts.
7. What is Maskable Interrupt?

THANK YOU