

CSE 4621

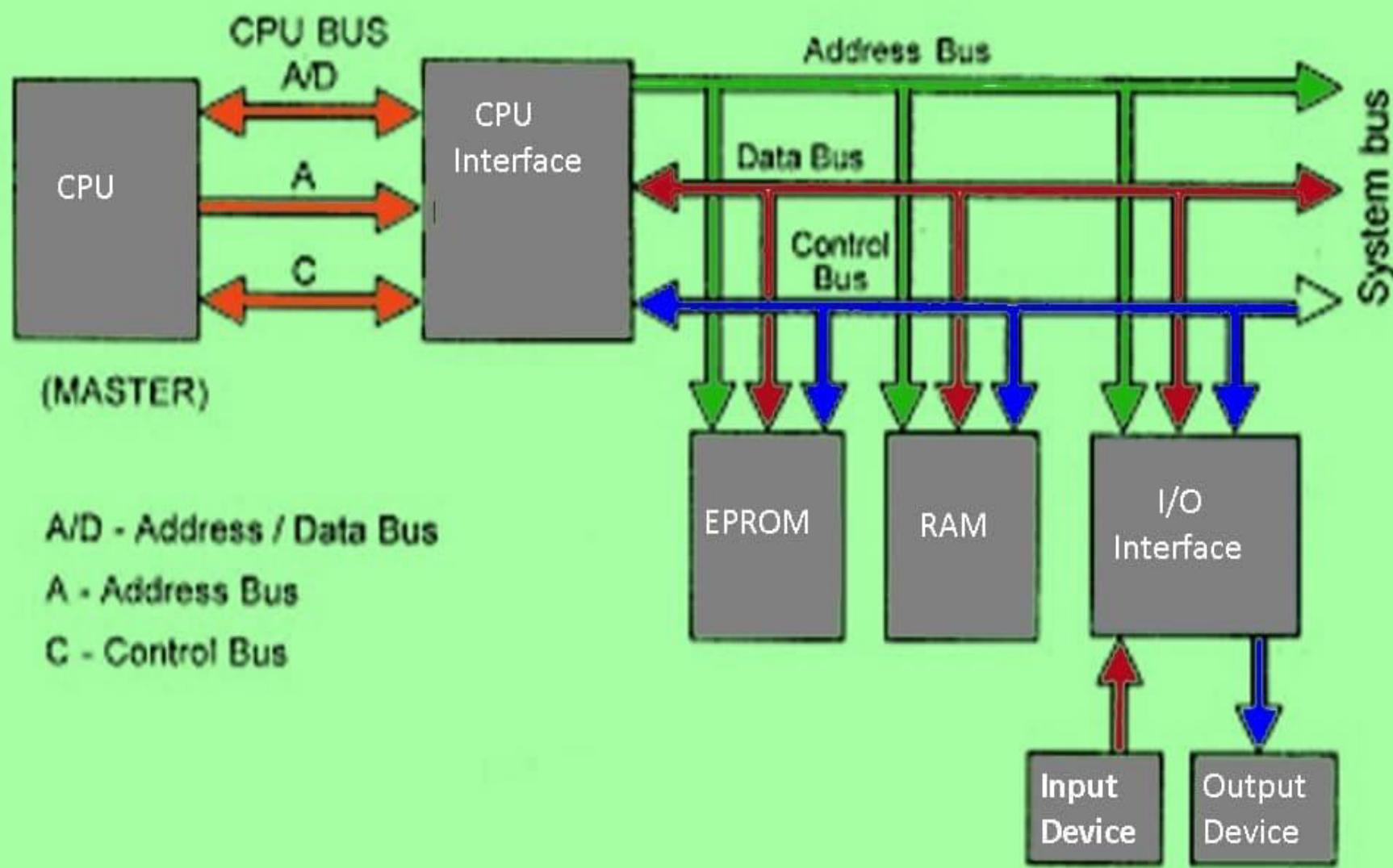
Microprocessor and Interfacing

Class-20-21

8085 Microprocessor :

- Architecture**
- Timing Diagram**

Interfacing Memory and I/O Devices



8085 Microprocessor Architecture

- ❖ Accumulator
- ❖ Arithmetic and logic Unit
- ❖ General purpose register
- ❖ Program counter
- ❖ Stack pointer
- ❖ Temporary register
- ❖ Flags
- ❖ Instruction register and Decoder
- ❖ Timing and Control unit
- ❖ Interrupt control
- ❖ Serial Input/output control
- ❖ Address buffer and Address-Data buffer
- ❖ Address bus and Data bus
- ❖ Incrementer/Decrementer Address Latch

History of 8085

- | | |
|-------------------------|-------------------|
| 1971 –Intel 4004 | - 4 bit μp |
| 1972 –Intel 8008 | - 8 bit μp |
| 1973 –Intel 8080 | - 8 bit μp |
| 1974 –Motorolla 6800 | - 8 bit μp |
| 1976 –Zilog 80 | - 8 bit μp |
| 1976 –Intel 8085 | - 8 bit μp |
| 1978 –Intel 8086 | - 16 bit μp |

Microprocessor Evolution and Types

NAME	YEAR	TRANSISTORS	DATA WIDTH	CLOCK SPEED
8080	1974	6,000	8 bits	2 MHz
8085	1976	6,500	8 bits	5 MHz
8086	1978	29,000	16 bits	5 MHz
8088	1979	29,000	8 bits	5 MHz
80286	1982	134,000	16 bits	6 MHz
80386	1985	275,000	32 bits	16 MHz
80486	1989	1,200,000	32 bits	25 MHz
PENTIUM	1993	3,100,000	32/64 bits	60 MHz
PENTIUM II	1997	7,500,000	64 bits	233 MHz
PENTIUM III	1999	9,500,000	64 bits	450 MHz
PENTIUM IV	2000	42,000,000	64 bits	1.5 GHz

Microprocessor Evolution and Types

Evolution of Intel Processor

4-bit	8-bit	16-bit	32-bit	64-bit
4004	8008	8086	80386	Dual Core
4040	8080	8088	80486	Core 2
	8085	80186	Pentium/80586	Core i7
		80188	PII	Core i5
		80286	PIII	Core i3
			PIV	
			Dual Core	

Microprocessor Evolution and Types

Processor	Year Of Introduction	No. Of Transistors	Initial Clock Speed	Address Bus	Data Bus(in bit)	Addressable Memory
4004	1971	2300	108 kHz	10 bit	4	640 bytes
8008	1972	3500	200 kHz	14 bit	8	16 k
8080	1974	6000	2 MHz	16 bit	8	64 k
8085	1976	6500	5 MHz	16 bit	8	64 k
8086	1978	29000	5 MHz	20 bit	16	1 M
8088	1979	29000	5 MHz	20 bit	8	1 M
80286	1982	134000	8 MHz	24 bit	16	16 M
80386	1985	275000	16 MHz	32 bit	32	4 G
80486	1989	1.2 M	25 MHz	32 bit	32	4 G
Pentium	1993	3.1 M	60 MHz	32 bit	32/64	4 G
Pentium Pro	1995	5.5 M	150 MHz	36 bit	32/64	64 G
Pentium II	1997	8.8 M	233 MHz	36 bit	64	64 G
Pentium III	1999	9.5 M	650 MHz	36 bit	64	64 G
Pentium 4	2000	42 M	1.4 GHz	36 bit	64	64 G

Microprocessor Evolution and Types:

Intel 8086 Microprocessor

Clock Speed	Data Width	Data Bus	Address Bus	Addressable Memory
5 MHz	16 bit	16 bit	20 bit	$2^{20} = 1 \text{ Mega byte}$

- ▶ Intel 8086 microprocessor is a **16 bit microprocessor**
- ▶ **Can process data and memory addresses** that are represented by 16 bits at a time.

Microprocessor Evolution and Types:

Intel 8085 Microprocessor

Clock Speed	Data Width	Data Bus	Address Bus	Addressable Memory
5 MHz	8 bit	8 bit	16 bit	$2^{16} = 64$ Kilo byte

- ▶ Intel 8086 microprocessor is a **8 bit microprocessor**
- ▶ **Can process data and memory addresses** that are represented by 8 bits at a time.

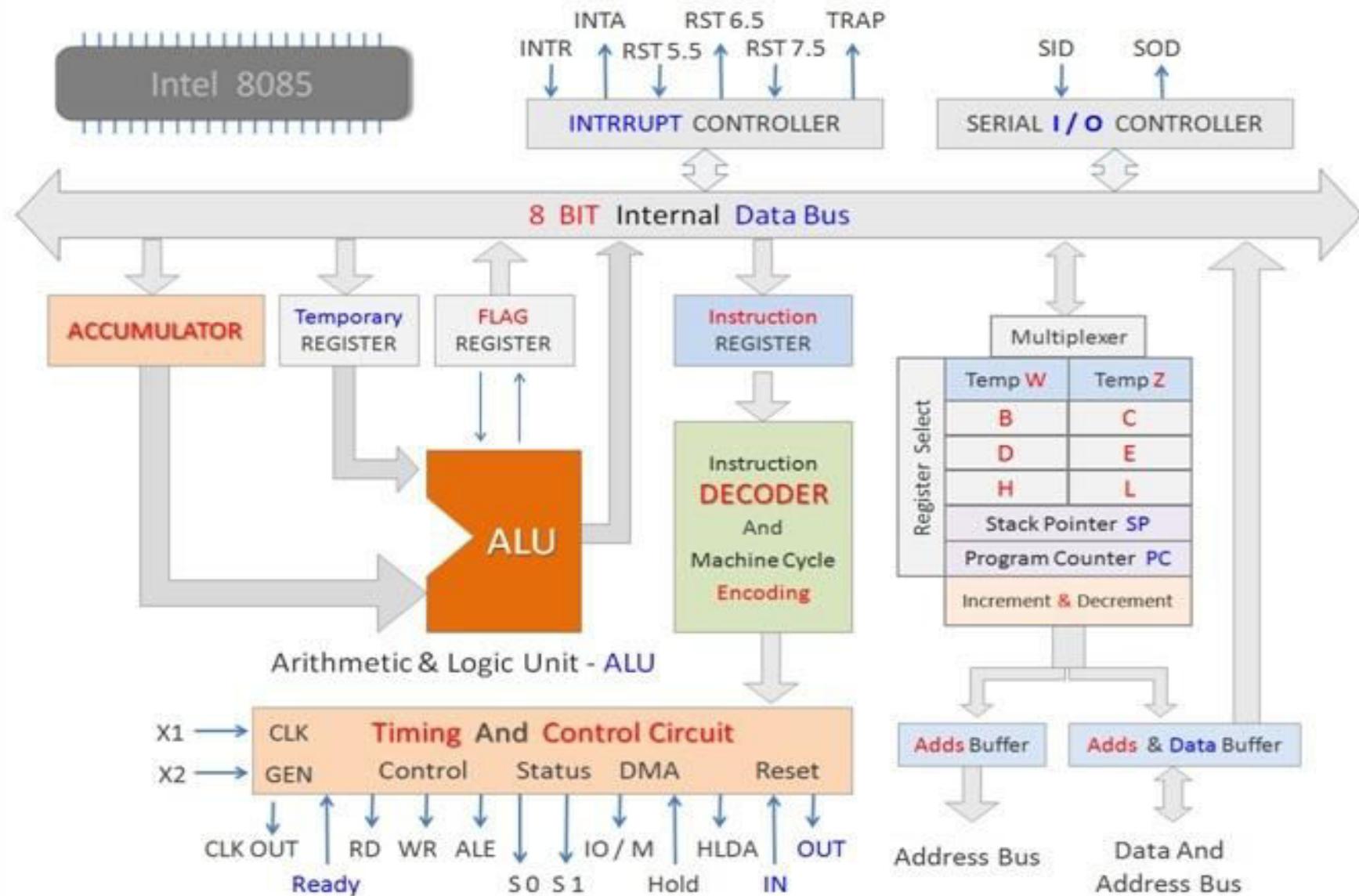
8085 Microprocessor Architecture

- 8-bit general purpose μp
- Capable of addressing 64 k of memory
- Has 40 pins
 - It is a 40 pin IC package fabricated on a single Large Scale Integration (LSI) chip
- Requires +5 v power supply
- Can operate with 3 MHz clock
- It has 80 basic instructions and 246 opcodes

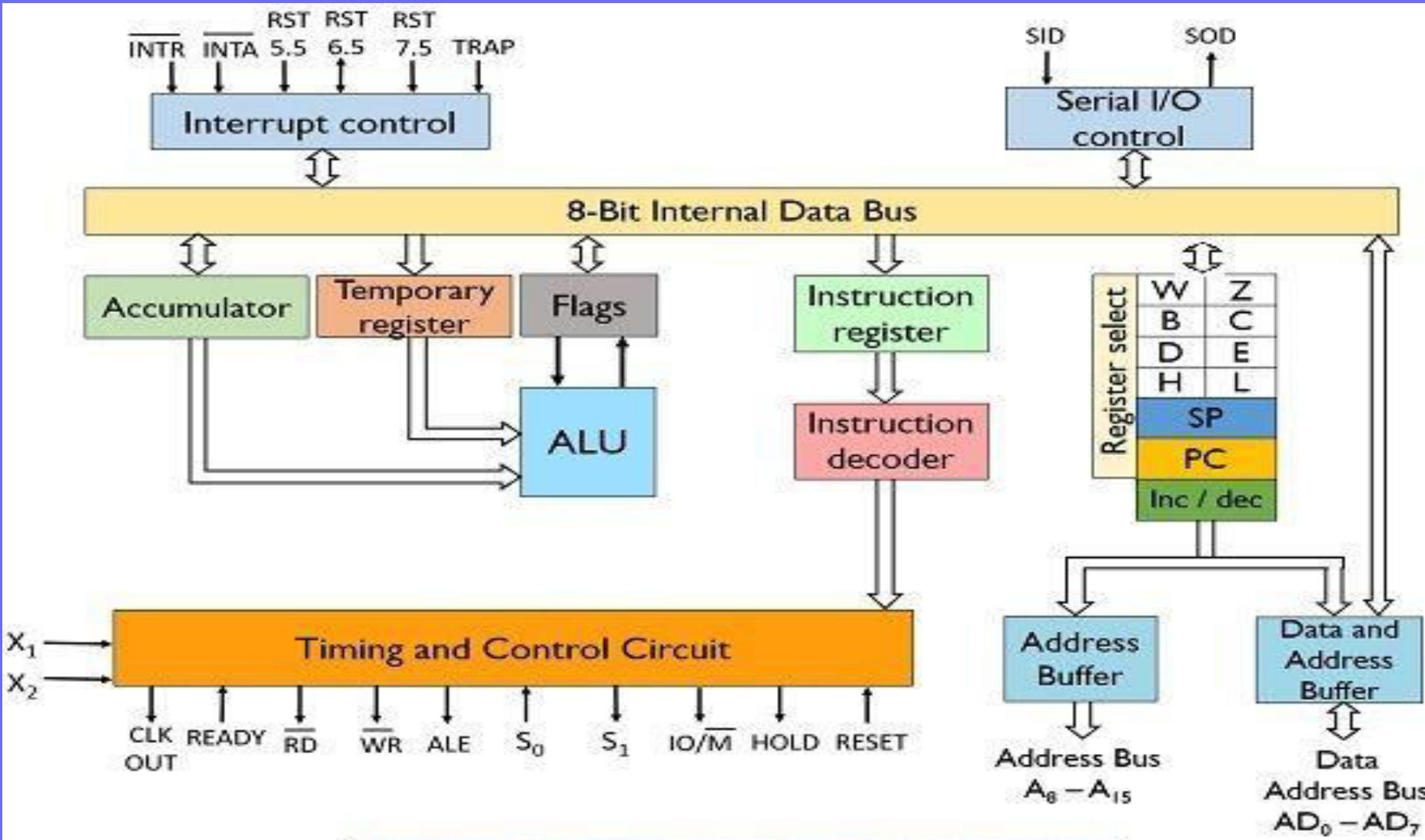
8085 Microprocessor : ALU

- The ALU performs the following arithmetic and logical operations.
 - Addition
 - Subtraction
 - Logical AND
 - Logical OR
 - Logical EXCLUSIVE OR
 - Complement (logical NOT)
 - Increment (add 1)
 - Decrement (subtract 1)
 - Left shift
 - Clear

Intel 8085 Block Diagram

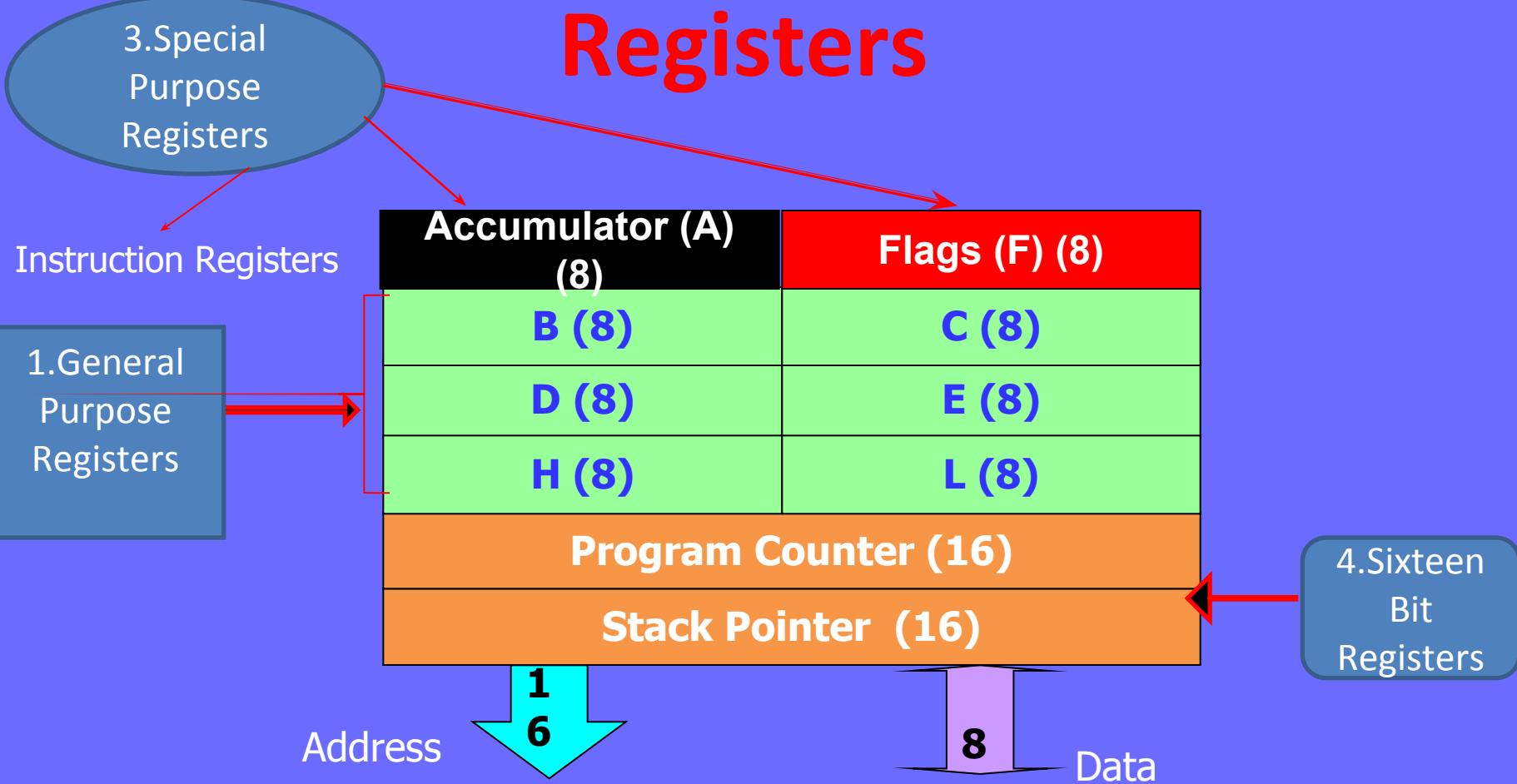


Intel 8085 Block Diagram



Architecture of 8085 Microprocessor

Registers



2.Temporary Registers
(Temp Data, W & Z)

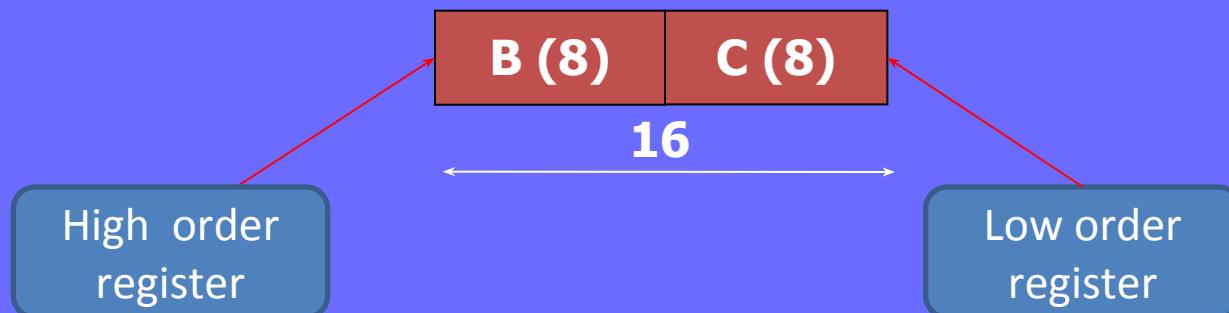
General purpose Registers

- ❖ Six general purpose 8-bit registers: B, C, D, E, H, L
- ❖ They can also be combined as register pairs to perform

16-bit operations: BC, DE, HL

Data Pointer or
Memory Pointer (M)

- ❖ Registers are programmable (Data load, Move etc.)



Temporary Registers

❖ Temporary data Register

The ALU has two inputs. One is **Accumulator** & other from **Temp data Register**

Ex : ADD B \square (A \square A+B)

First B is transferred to Temp then add with A

❖ W & Z (16)

Ex : CALL, XCHG \square (HL $\square\square$ DE)

Note

The programmer Can not access this temp Registers

First DE transferred to WZ then exchange with HL

Special purpose Registers

1. Accumulator (A)

- ❖ Single 8-bit register that is part of the ALU
- ❖ Used in
 - ✓ Arithmetic/logic operations
 - ✓ Load
 - ✓ store
 - ✓ As well as I/O operation

2. Instruction Registers (Discuss later)

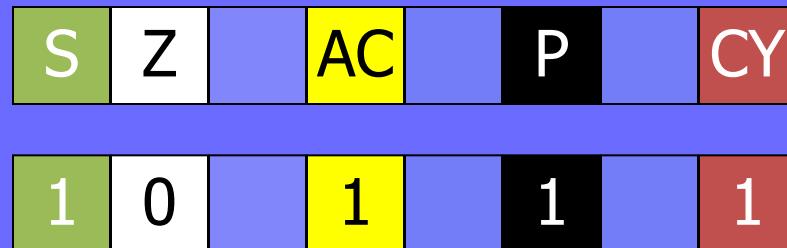
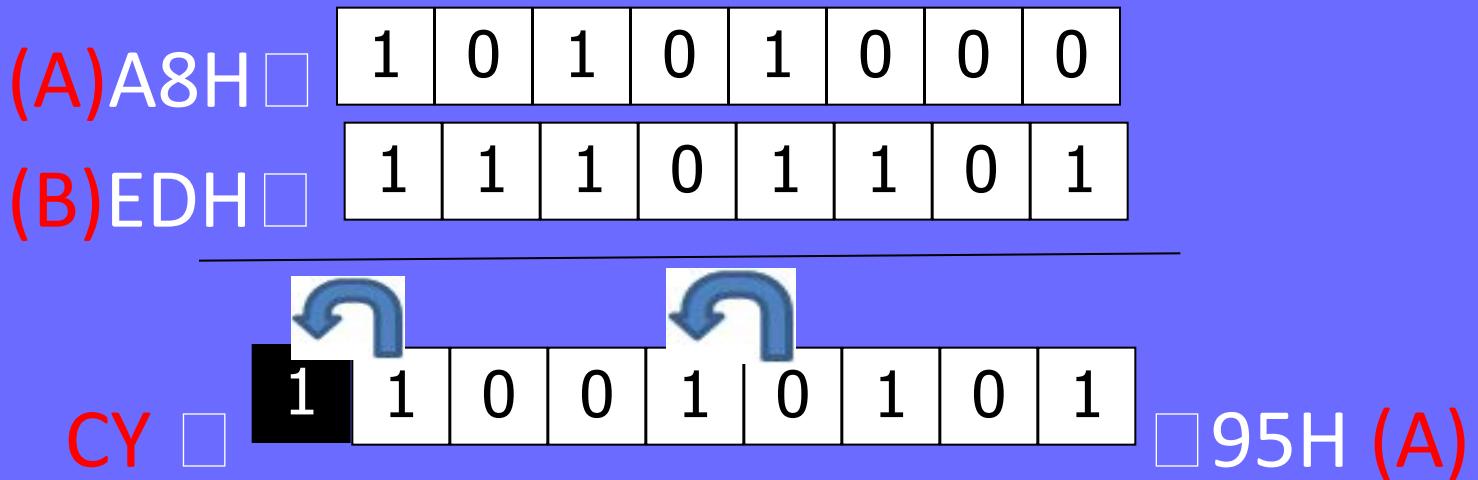
Special purpose Registers

3.Flag



- ❖ S = After the execution of an arithmetic operation, if bit 7 of the result is 1, then sign flag is set. (1 □ Negative 0 □ Positive)
- ❖ Z = Bit is set if ALU operation results a zero in the Accumulator
- ❖ AC = Bit is set, when a carry is generated by bit 3 & passed on bit 4.
- ❖ P = Parity bit is set when the result has even number of 1s.
- ❖ CY = carry is set when result generates a carry. Also a borrow flag.

Example : ADD B



Program counter & Stack

Pointer

- ❖ PC acts as a pointer to the NEXT instruction depends on upon nature of instruction (1 or 2 or 3 Bytes)
- ❖ PC automatically increments to point to the next memory during the execution of the present instruction.
(In Jump or CALL , PC changes to address of subprogram)
- ❖ Stack is reserved area of the memory
(Temporary information storage - LIFO algorithm)
- ❖ After every stack operation SP points to next available location of the stack (Recent Entry)

ALU (Arithmetic & Logic Unit)

- ❖ To perform arithmetic operations like

Addition & Subtraction

- ❖ To perform logical operations like

✓ AND

✓ OR

✓ NOT (Complement)

Instruction Register & Decoder

- ❖ The processor first fetches the opcode of instruction from memory & stores opcode in the instructions registers , it is then sent to instruction decoder
- ❖ The Instruction decoder decodes the it & accordingly gives for further processing depending on nature of instructions

Address buffer & Address/Data buffer

Address Buffer

- ❖ 8 bit unidirectional buffer
- ❖ The address bits are always sent from the MPU to peripheral devices, not reverse
- ❖ Used to drive external High order address bus (A_8 - A_{15})

Address Buffer & Address/Data Buffer

- ❖ 8 bit Bidirectional buffer
- ❖ The data bits are sent from the MPU to peripheral devices, as well as from the peripheral devices to the MPU
- ❖ Used to drive multiplexed address/data bus
i.e Low order address bus (A_0 - A_7) & data bus (D_0 - D_7)

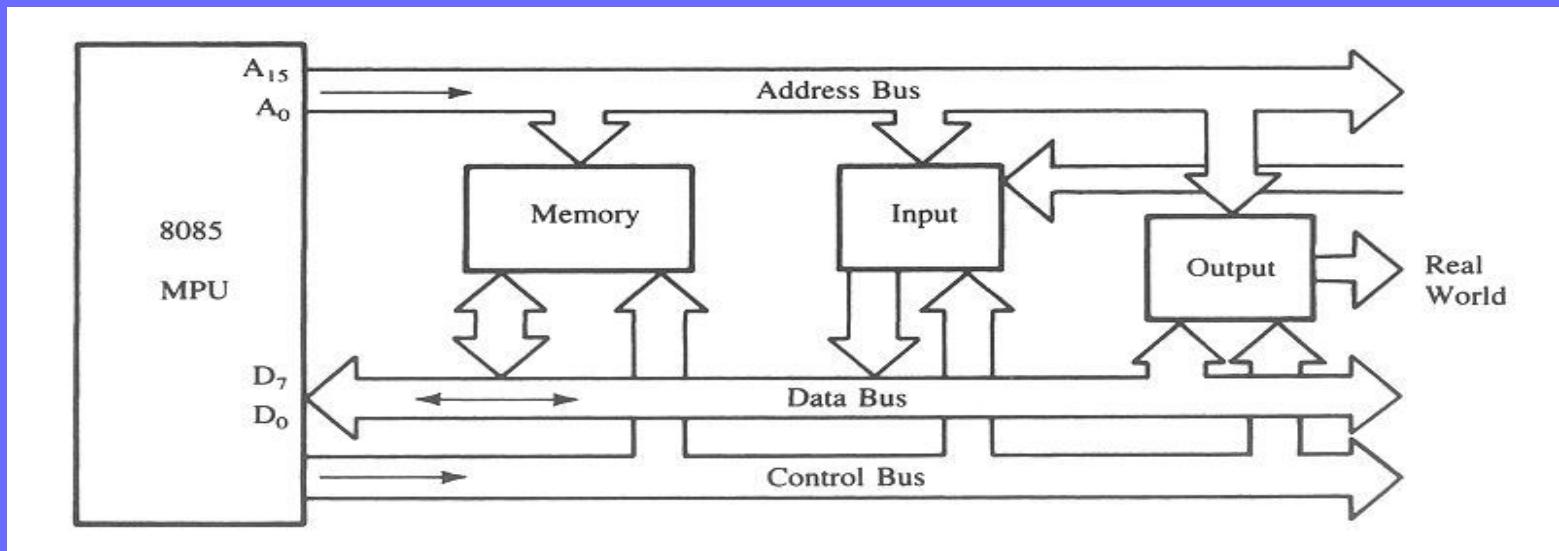
Bus: A shared group of wires used for communicating signals among devices

- Address Bus : The device and the location within the device that is being accessed

Total $2^{16} = 65,536$ (64k) Memory Locations

Address Locations: 0000H– FFFFH

- Data Bus : The data value being communicated
Data Range: 00H – FFH
- Control Bus: Describes the action on the address & data buses



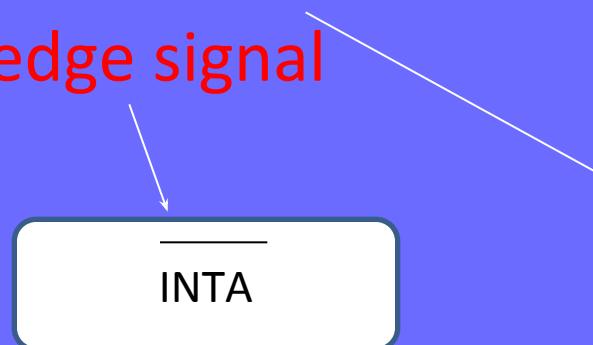
Incrementer/ Decrementer

Address Latch

- ❖ This 16 bit Register is used to increment or decrement the contents of the PC or SP as a part of execution instructions related to them

Interrupt Control

- ❖ When the Microprocessor receives an interrupt signal, it suspends the currently executing program and jumps to (Special Routine or Sub program) an Interrupt Service Routine (ISR) to respond to the incoming interrupt
- ❖ Five Interrupt inputs & one Acknowledge signal



Interrupt name
INTR
RST 5.5
RST 6.5
RST 7.5
TRAP

Serial I/O Control

❖ It provides two lines SOD & SID for Serial Communication

1.SOD (Serial Output Data)

□ used to send Data serially

2.SID (Serial Input Data)

□ used to Receive Data serially

Timing & Control /circuitry

Timing and control unit is a very important unit as it synchronizes the registers and flow of data through various registers and other units

Control Signals : READY, RD, WR, ALE

Status Signals : S0, S1, IO/ M

DMA Signals : HOLD, HLDA

RESET Signals : RESET IN, RESET OUT

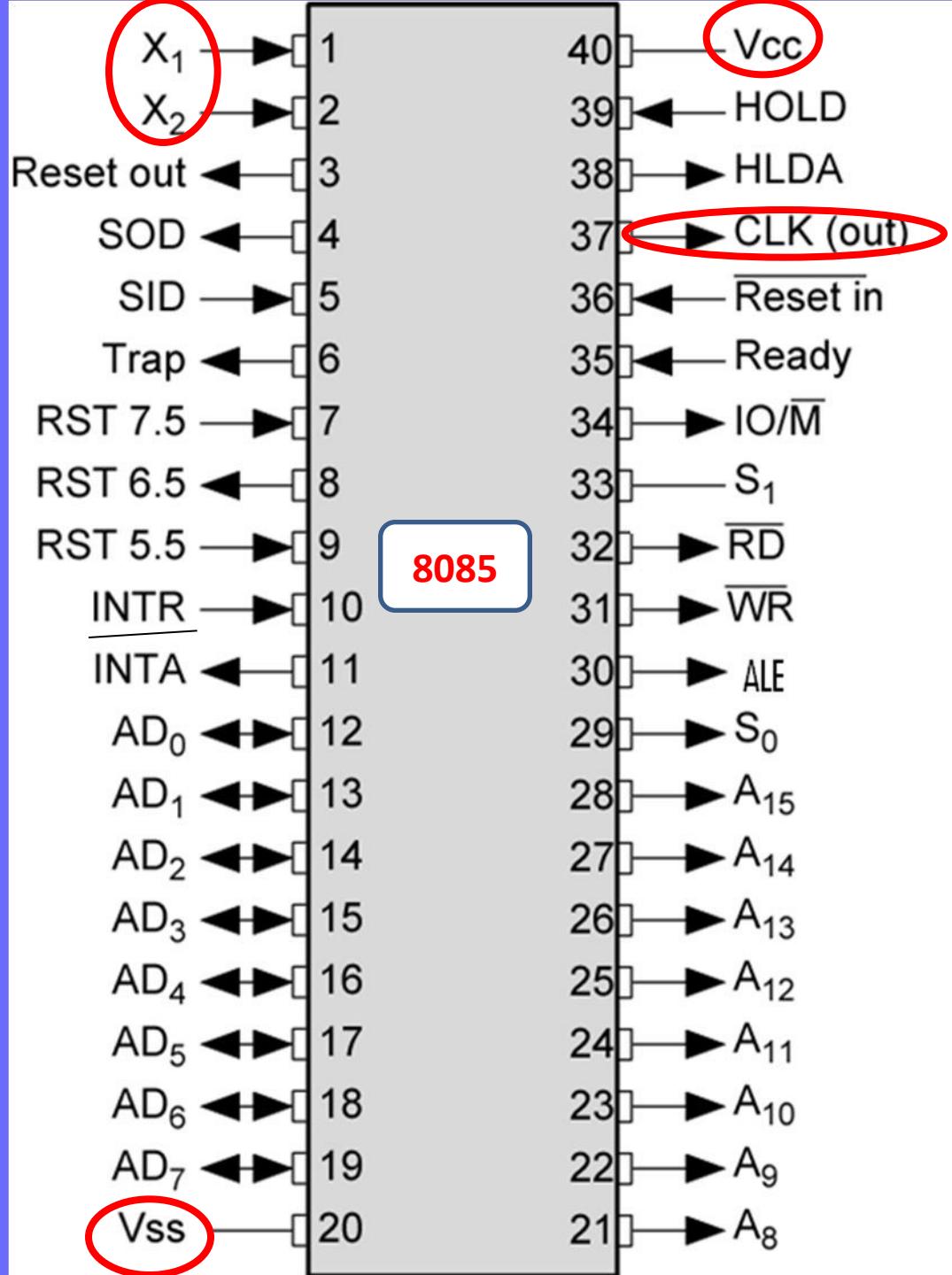
Intel 8085 Pin Configuration

X ₁	1	40	Vcc
X ₂	2	39	HOLD
Reset out	3	38	HLDA
SOD	4	37	CLK (out)
SID	5	36	Reset in
Trap	6	35	Ready
RST 7.5	7	34	IO/M
RST 6.5	8	33	S ₁
RST 5.5	9	32	RD
INTR	10	31	WR
INTA	11	30	ALE
AD ₀	12	29	S ₀
AD ₁	13	28	A ₁₅
AD ₂	14	27	A ₁₄
AD ₃	15	26	A ₁₃
AD ₄	16	25	A ₁₂
AD ₅	17	24	A ₁₁
AD ₆	18	23	A ₁₀
AD ₇	19	22	A ₉
Vss	20	21	A ₈

Classifications of Pins

8085 has 40 PIN IC

1. POWER SUPPLY & FREQUENCY Signals
2. DATA Bus & ADDRESS Bus
3. CONTROL & STATUS Signals
4. INTERRUPT Signals
5. SERIAL I/O Signals
6. DMA Signals
7. RESET Signals



Power Supply & Frequency Signals

- V_{CC} : +5 Power Supply
- V_{ss} : Ground Reference
- X1 and X2 : Determine the Clock Frequency
- CLOCK OUT : Half the crystal or Oscillator Frequency (Used as a system clock for other devices)



Data Bus & Address Bus

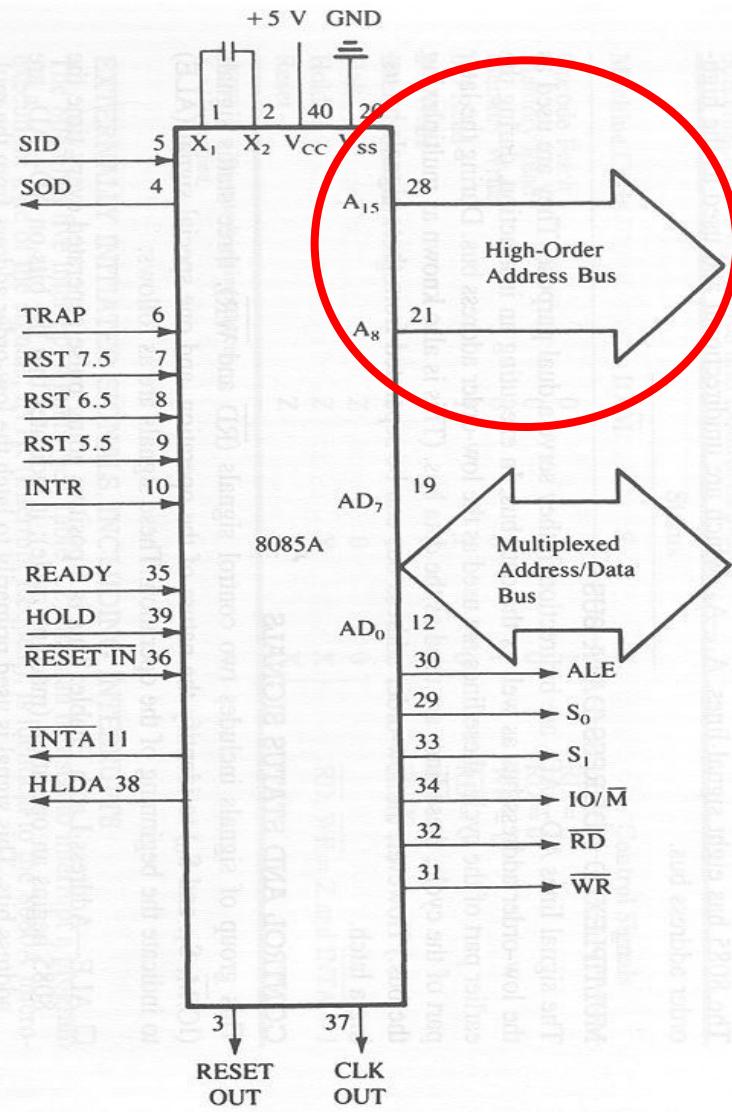
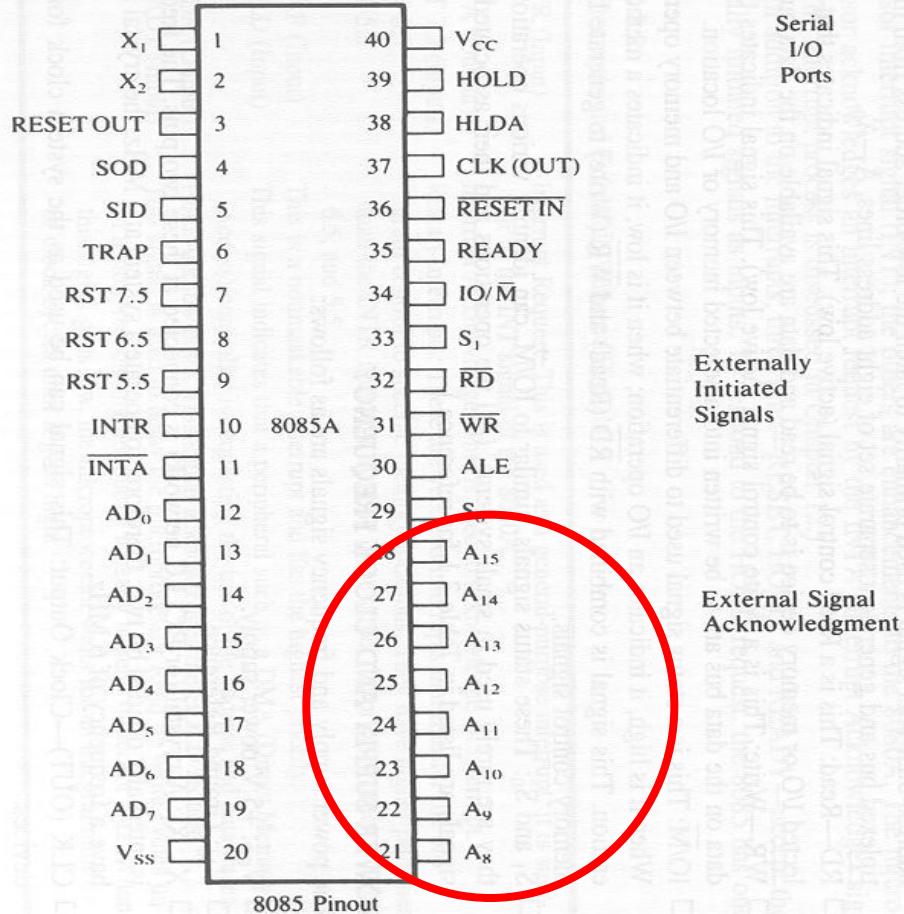
8085 µp consists of 16pins use as Address Bus & 8 pins use as Data Bus

Divide into 2 part : $A_8 - A_{15}$ (Upper)
: $AD_0 - AD_7$ (Lower)

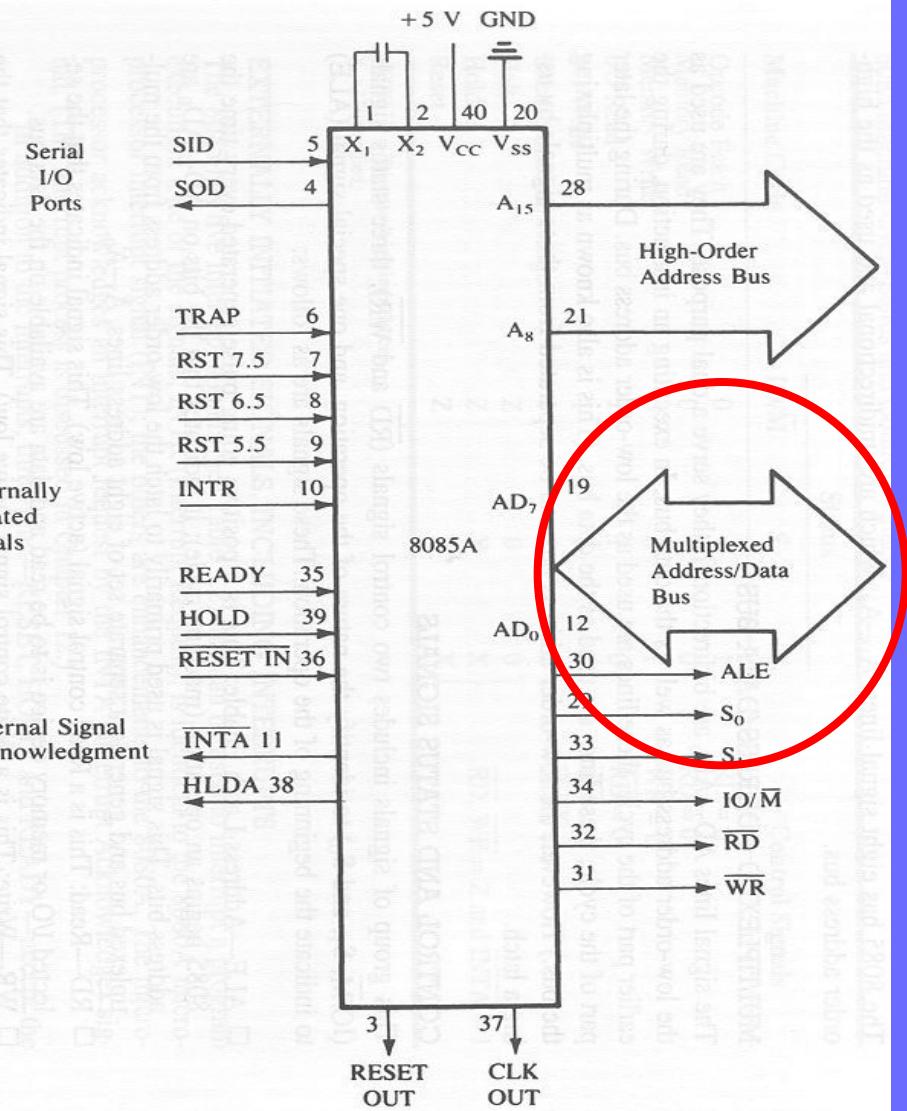
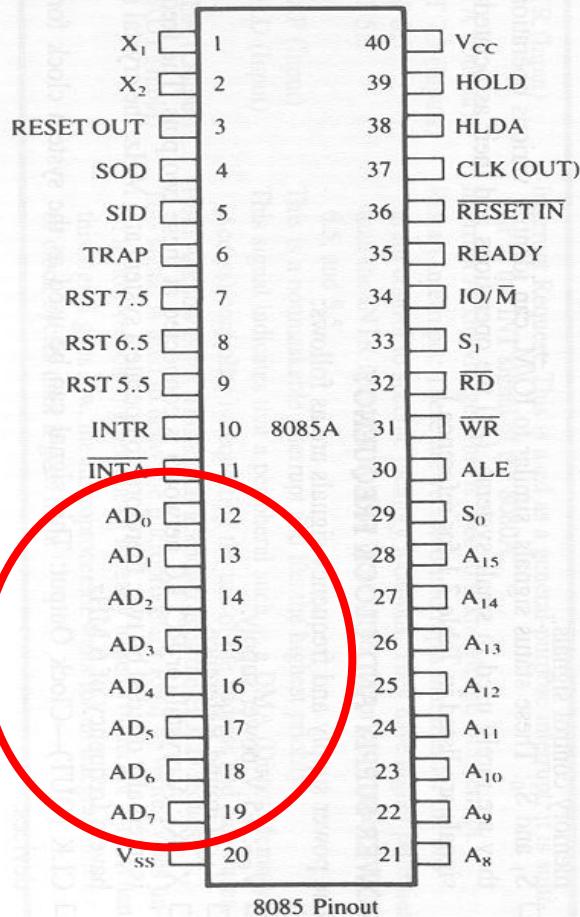
$A_8 - A_{15}$: Unidirectional, known as 'High Order Address'
 $AD_0 - AD_7$: Bidirectional and Dual purpose
(Address and Data are Multiplexed)
 $A_0 - A_7$ □ Low Order Address
 $D_0 - D_7$ □ Data Bus

The method to change from address bus to data bus known as "Bus Multiplexing" (Adv : Reduces the Number of Pins)

High -order Address Bus(8 bits)



Low -order Address Bus(8 bits) & Data Bus(8 bits)



Control & Status Signals

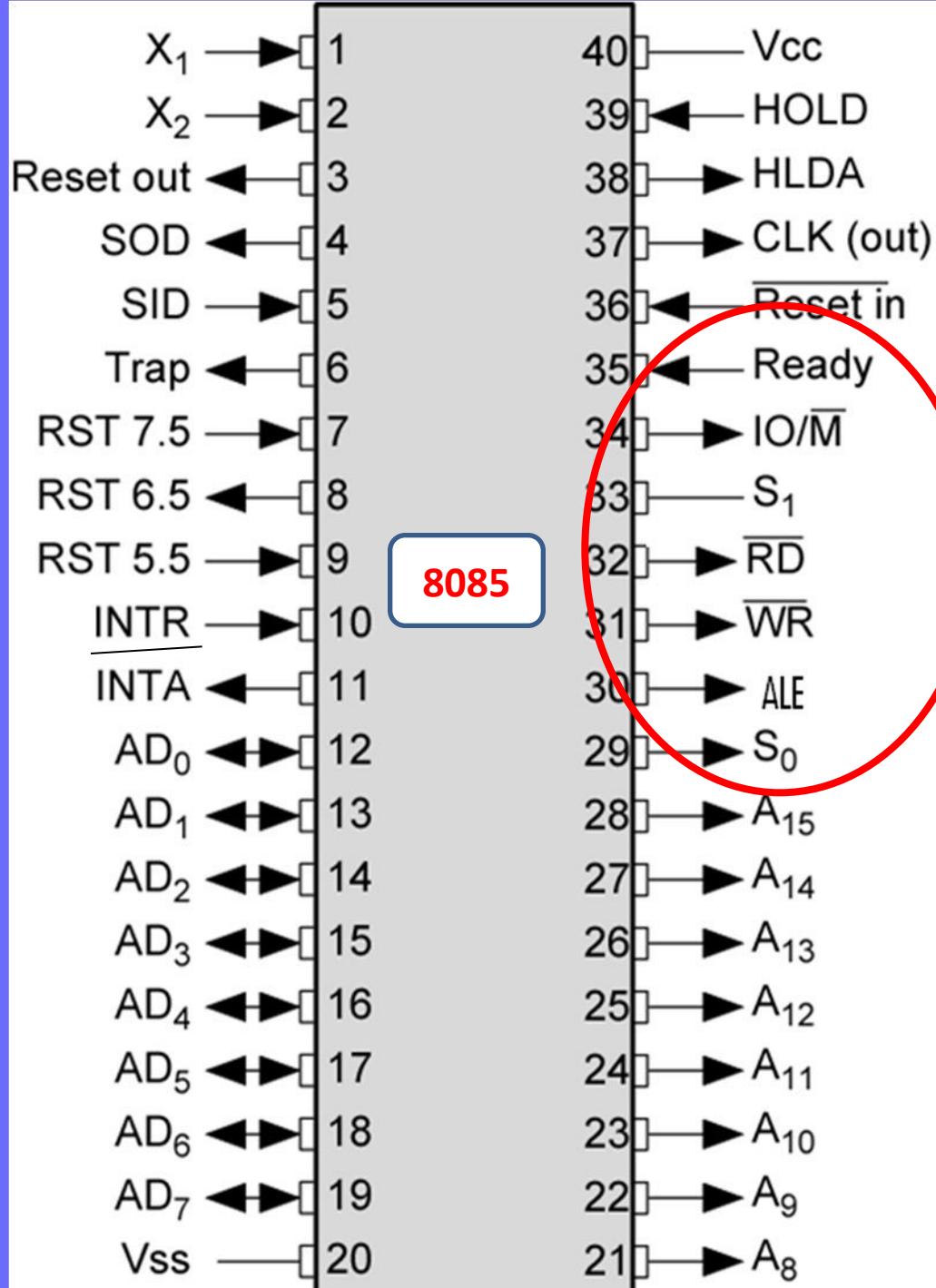
ALE : Address Latch Enable

RD & WR : Read & Write Operation

IO/M : I/O Operation or Memory Operation

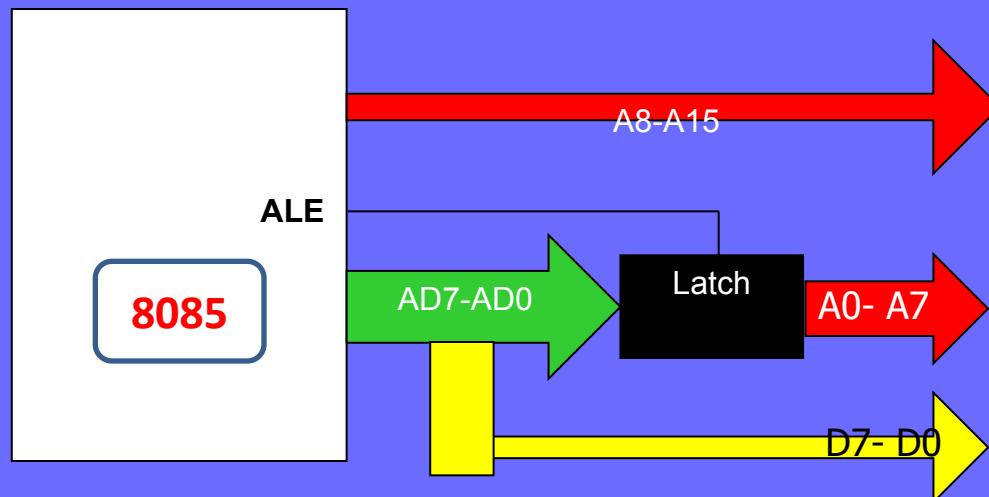
S₀ & S₁ : Machine Cycle Progress

READY : Peripheral is ready or not for Data transfer



ALE used to Demultiplex Address/Data bus

ALE -Active high output used to **latch** the lower 8 address bits $A_0 - A_7$



Control & Status Signals

- ❖ RD (Active low) □ To indicate that the I/O or memory selected is to be read and data are available on the bus
- ❖ WR (Active low) □ This is to indicate that the data available on the bus are to be written to memory or I/O
- ❖ IO/M □ To differentiate I/O or memory operations
 - '0' - indicates a memory operation
 - '1'-indicates an I/O operation
- ❖ SO & S1 □ Status signals, similar to IO/M

TABLE 4.1

8085 Machine Cycle Status and Control Signals

Machine Cycle	Status			Control Signals
	IO/M	S₁	S₀	
Opcode Fetch	0	1	1	$\overline{RD} = 0$
Memory Read	0	1	0	$\overline{RD} = 0$
Memory Write	0	0	1	$\overline{WR} = 0$
I/O Read	1	1	0	$\overline{RD} = 0$
I/O Write	1	0	1	$\overline{WR} = 0$
Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0$
Halt	Z	0	0	
Hold	Z	X	X	\overline{RD} , $\overline{WR} = Z$ and $\overline{INTA} = 1$
Reset	Z	X	X	

NOTE: Z = Tri-state (high impedance)

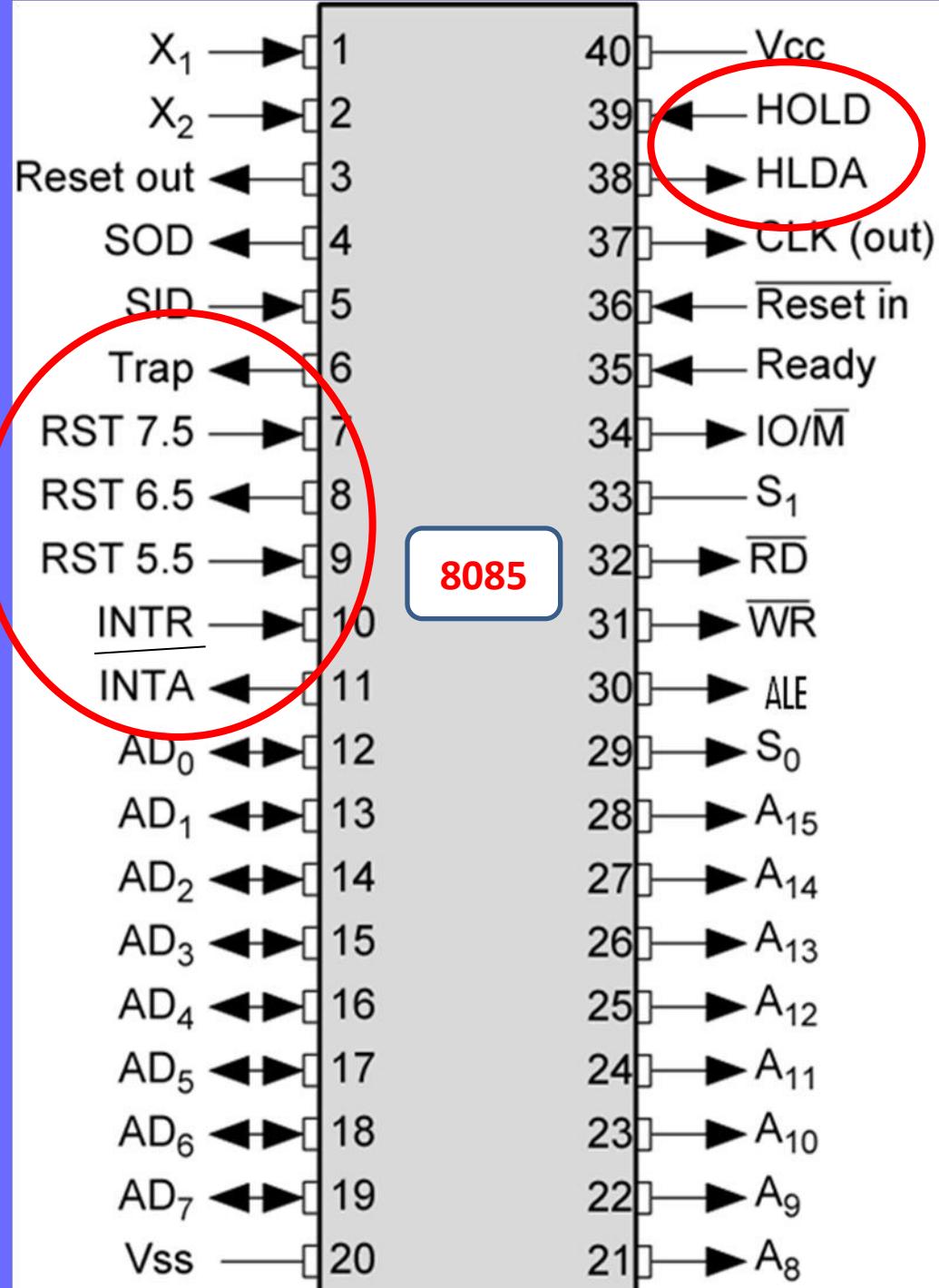
X = Unspecified

Interrupt & DMA Signals

TABLE 4.2

8085 Interrupts and Externally Initiated Signals

<input type="checkbox"/> INTR (Input)	Interrupt Request: This is used as a general-purpose interrupt; it is similar to the INT signal of the 8080A.
<input type="checkbox"/> INTA (Output)	Interrupt Acknowledge: This is used to acknowledge an interrupt.
<input type="checkbox"/> RST 7.5 (Inputs) RST 6.5 RST 5.5	Restart Interrupts: These are vectored interrupts that transfer the program control to specific memory locations. They have higher priorities than the INTR interrupt. Among these three, the priority order is 7.5, 6.5, and 5.5.
<input type="checkbox"/> TRAP (Input)	This is a nonmaskable interrupt and has the highest priority.
<input type="checkbox"/> HOLD (Input)	This signal indicates that a peripheral such as a DMA (Direct Memory Access) controller is requesting the use of the address and data buses.
<input type="checkbox"/> HLDA (Output)	Hold Acknowledge: This signal acknowledges the HOLD request.
<input type="checkbox"/> READY (Input)	This signal is used to delay the microprocessor Read or Write cycles until a slow-responding peripheral is ready to send or accept data. When this signal goes low, the microprocessor waits for an integral number of clock cycles until it goes high.



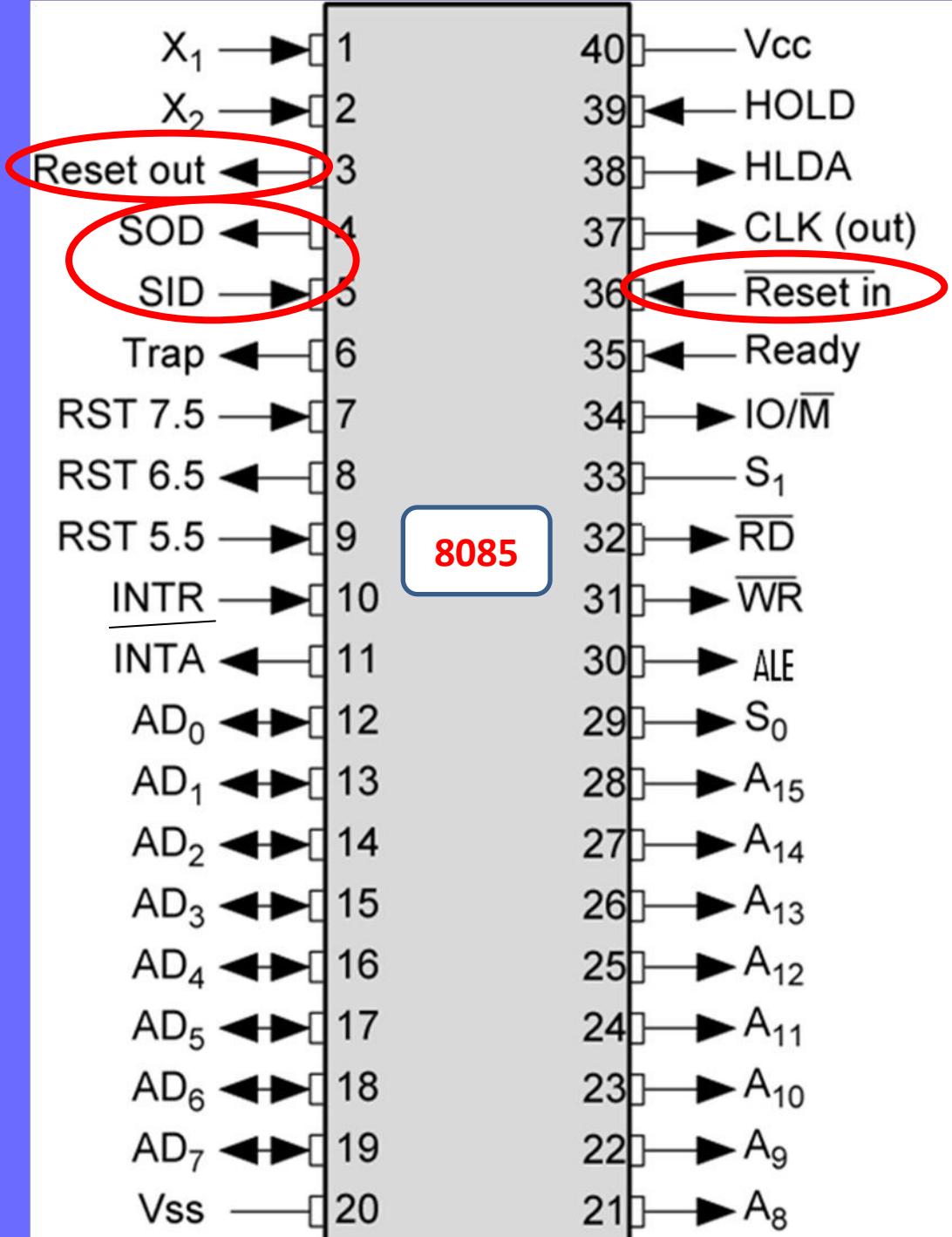
Serial I/O Control

SID (**Serial Input Data**)

- used to Receive or accept Data serially bit by bit from the external device

SOD (**Serial Output Data**)

- used to Transmit or send Data serially bit by bit to the external device



Reset Signals

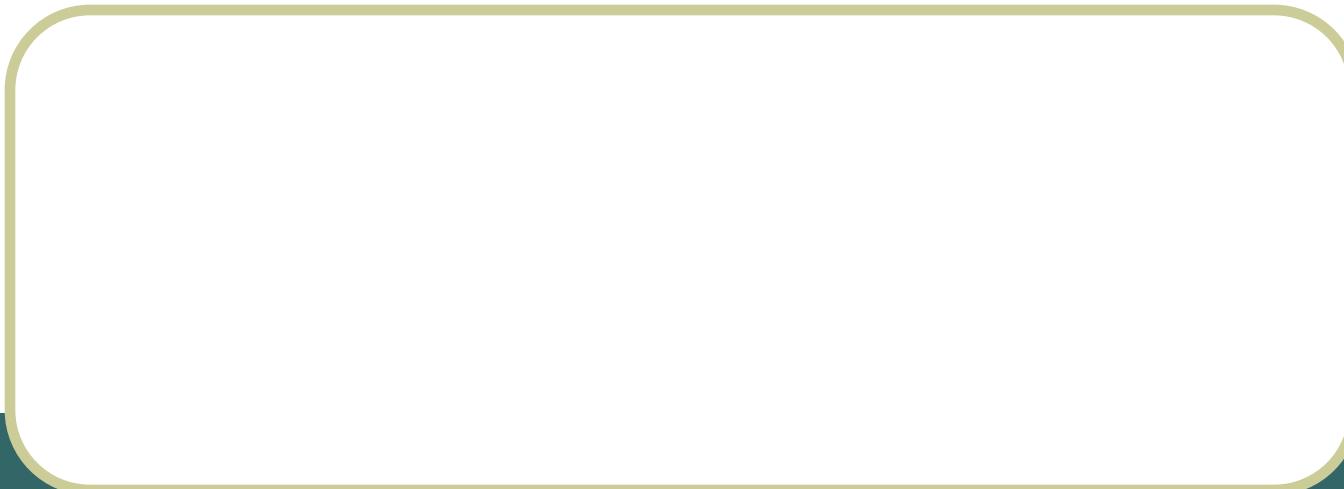
RESET IN an active low input signal

1. Set Program Counter to Zero $PC=0000H$ (μp will reset)
2. Reset interrupt & HLDA Flip-flops
3. Tri states the address, data & control bus
4. Affects the contents of internal registers randomly

RESET OUT to indicate that the μp was reset (RESET IN = 0)

- It also used to reset external devices.

TIMING DIAGRAM OF 8085



TIMING DIAGRAM

- Timing Diagram is a graphical representation.
- 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.

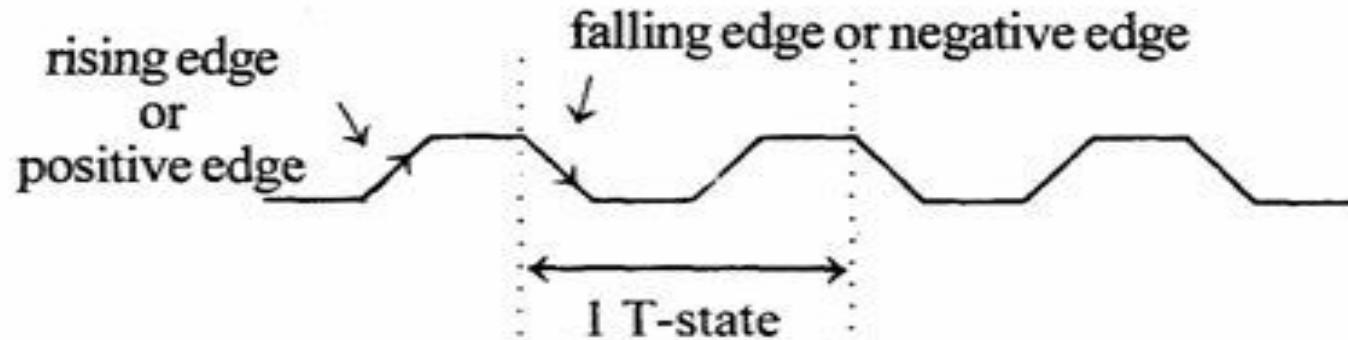
MACHINE CYCLE

- 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

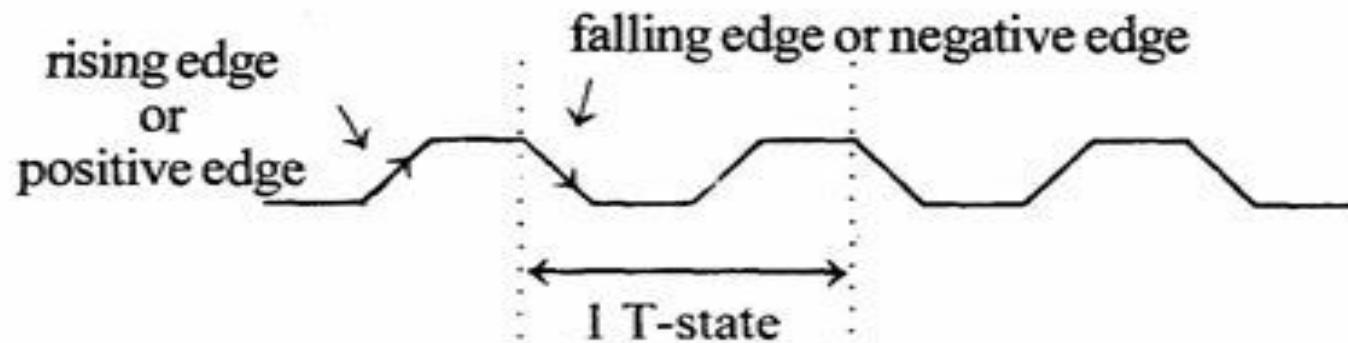
Note : Time period, $T = 1/f$; where $f = \text{Internal clock frequency}$



Clock Period

- The **clock period** time, T , is the time between Falling edges of a repetitive **clock** signal. .

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



MACHINE CYCLES OF 8085

The 8085 microprocessor has 5 basic machine cycles.

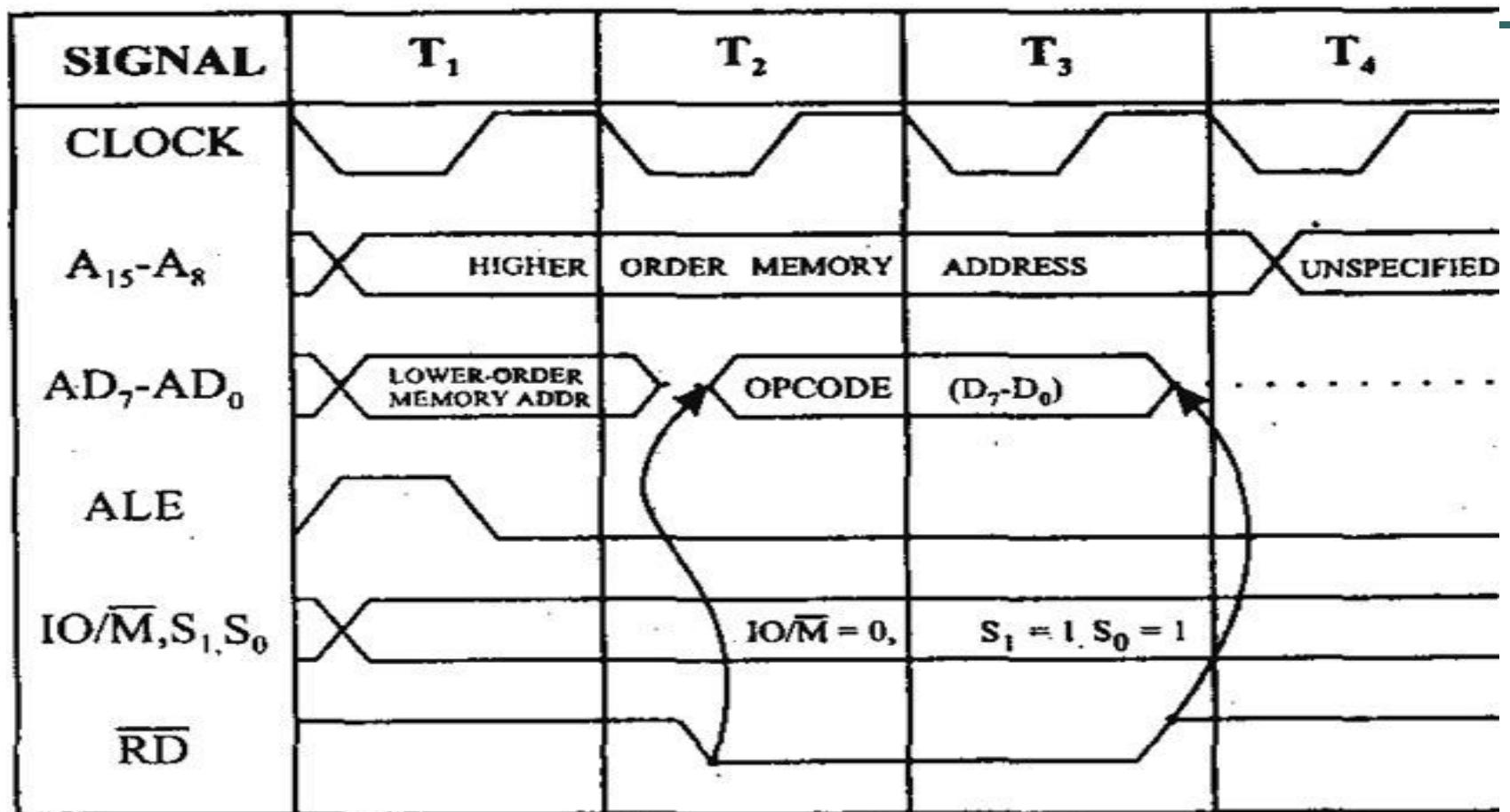
They are

1. Opcode fetch cycle (4T)
2. Memory read cycle (3T)
3. Memory write cycle (3T)
4. I/O read cycle (3 T)
5. I/O write cycle (3 T)

MACHINE CYCLES OF 8085

- The processor takes a definite time to execute the machine cycles. The time taken by the processor to execute a machine cycle is expressed in T-states.
- One T-state is equal to the time period of the internal clock signal of the processor.
- The T-state starts at the falling edge of a clock.

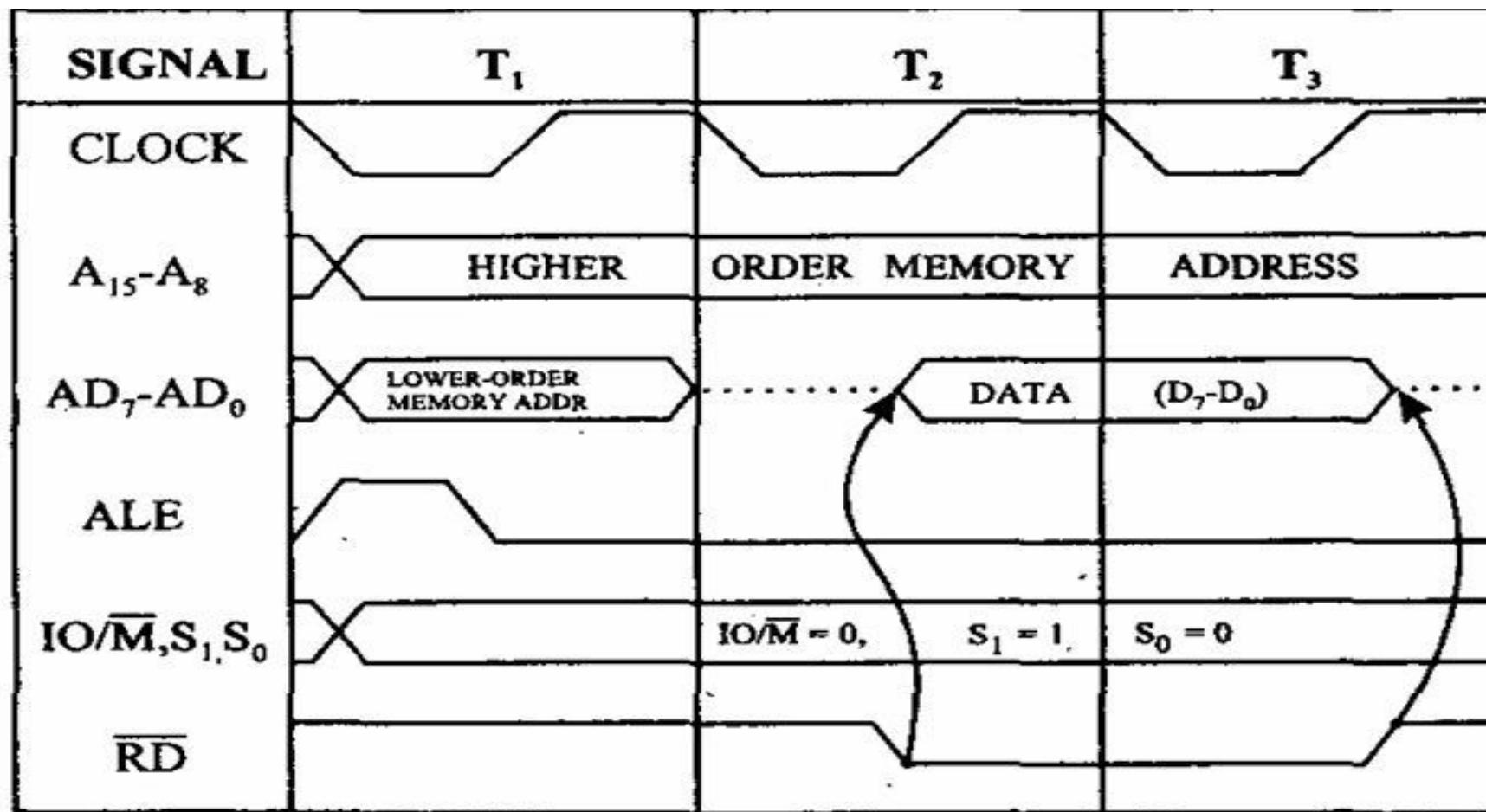
OPCODE FETCH MACHINE CYCLE OF 8085



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.

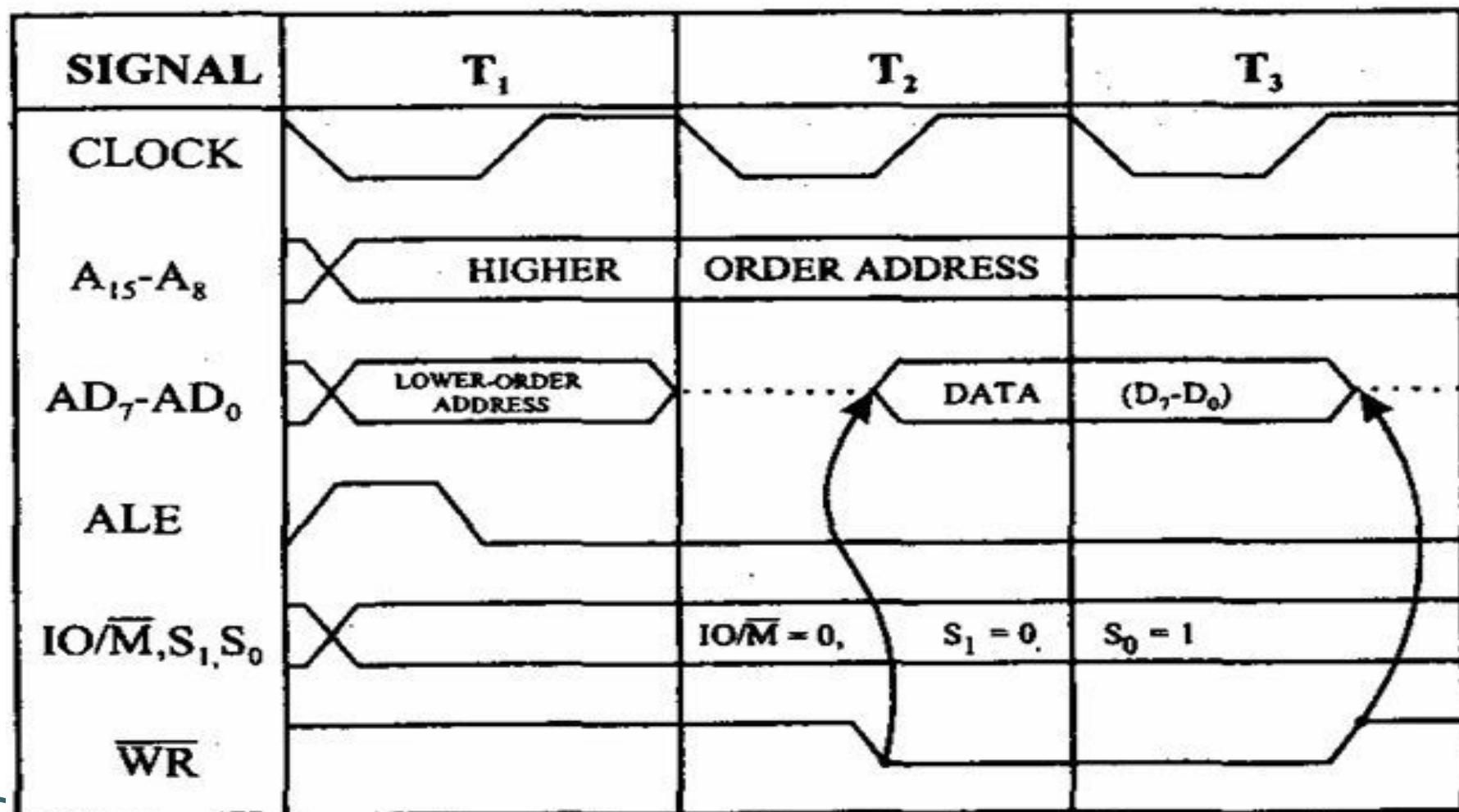
MEMORY READ MACHINE CYCLE OF 8085



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.

MEMORY WRITE MACHINE CYCLE OF 8085



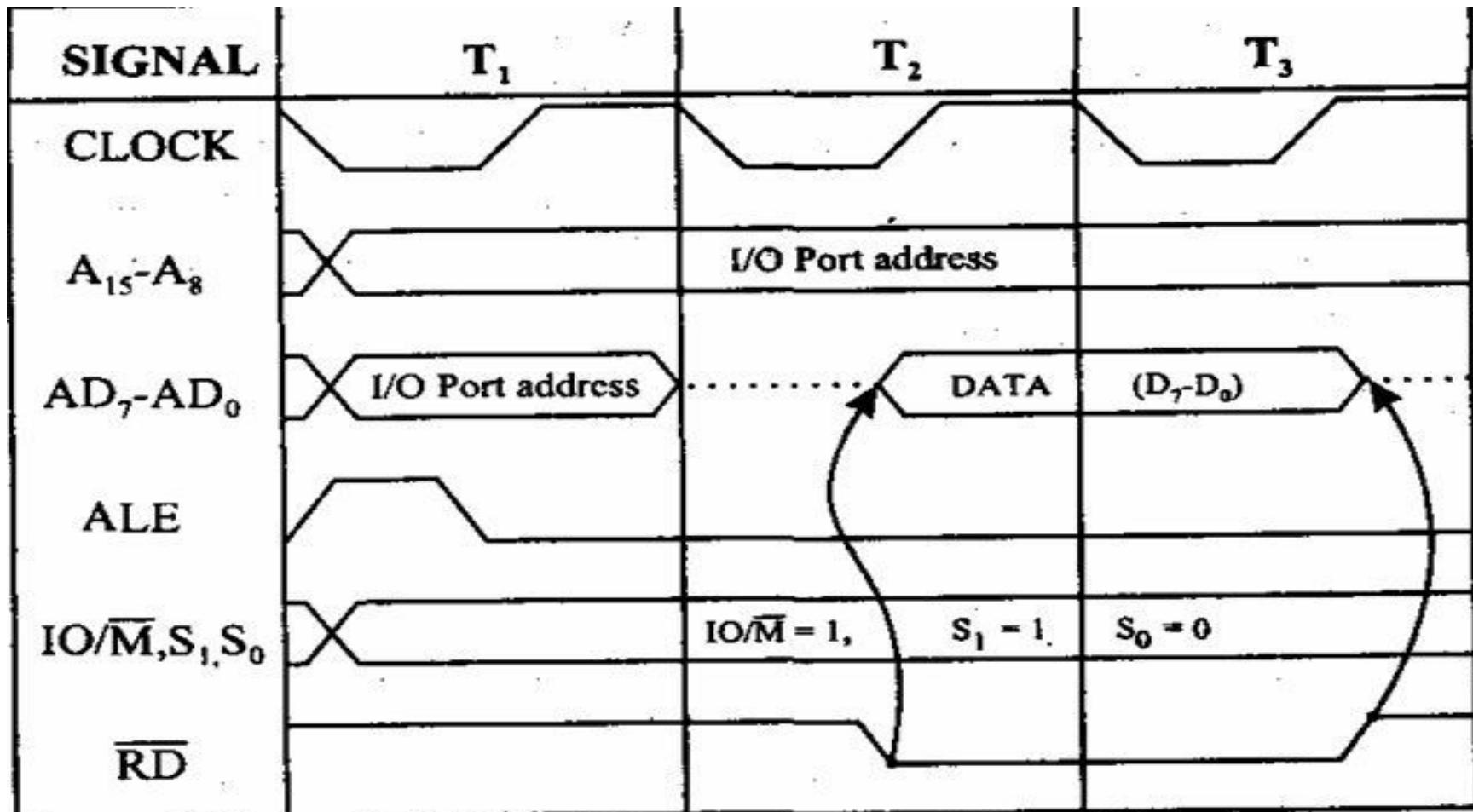
MEMORY WRITE MACHINE CYCLE OF 8085

- 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

I/O READ CYCLE OF 8085

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

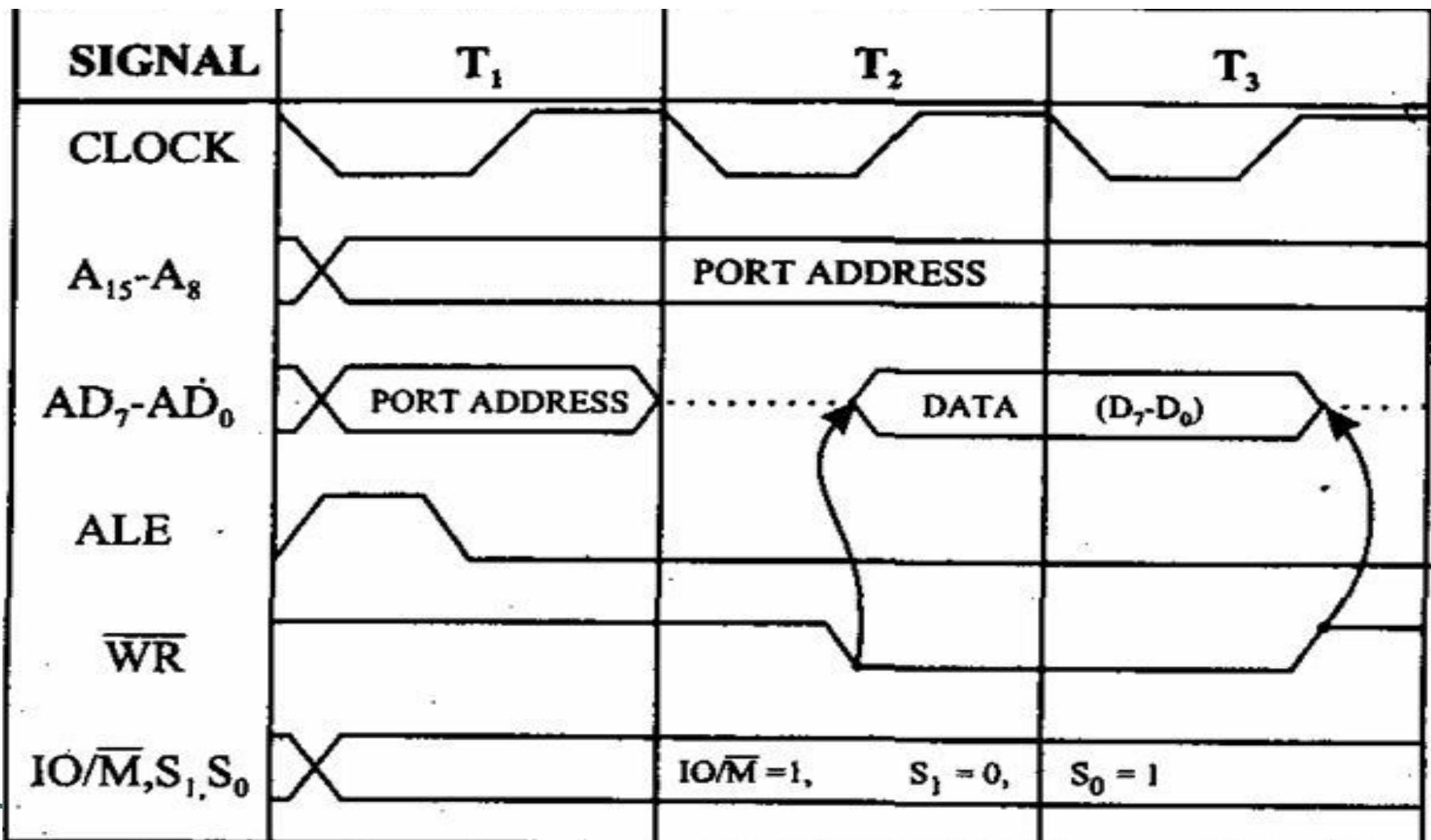
I/O READ CYCLE OF 8085



I/O WRITE CYCLE OF 8085

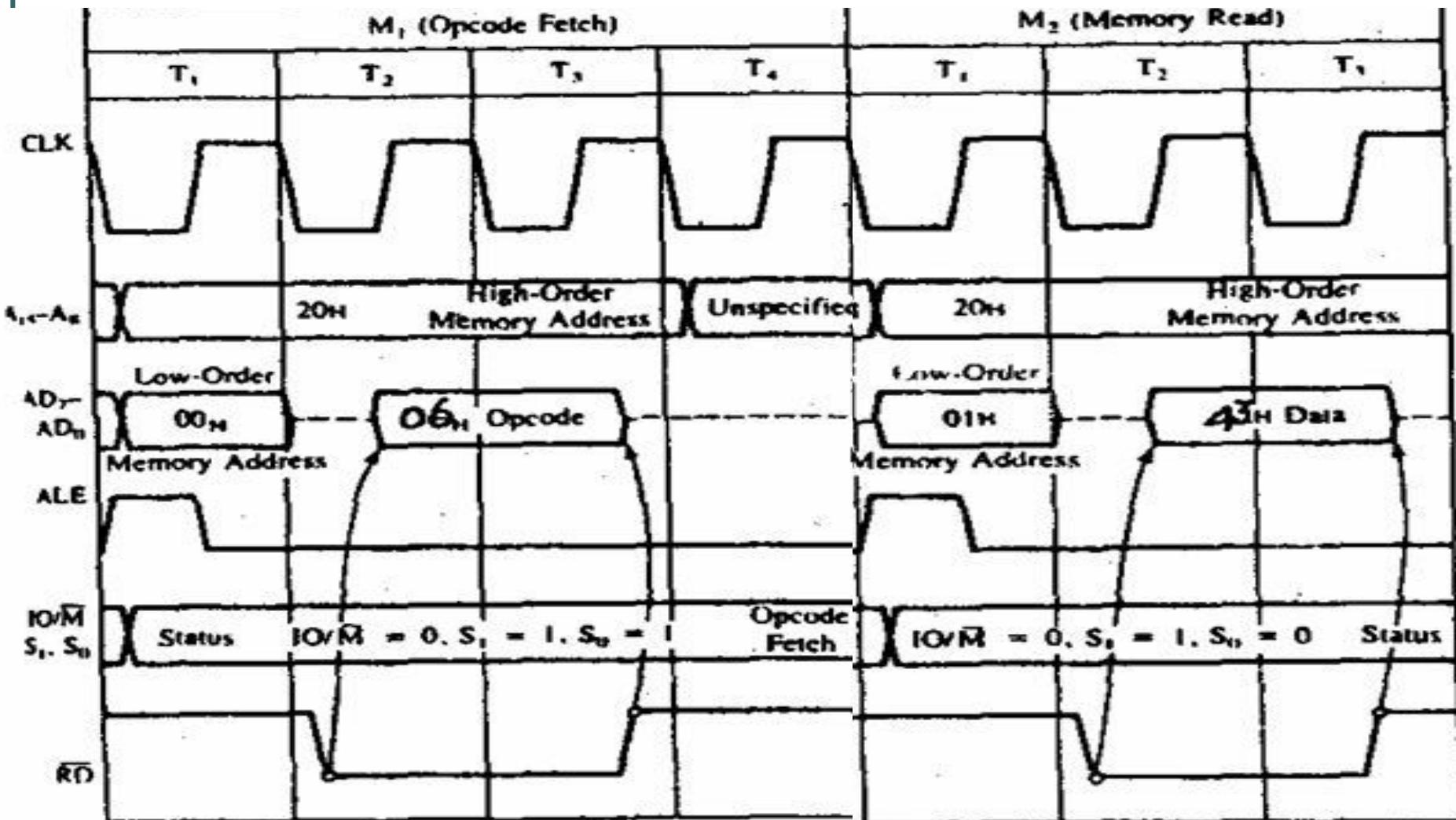
- The I/O write machine cycle is executed by the processor to write a data byte in the I/O port or to a peripheral, which is I/O, mapped in the system.
- The processor takes, 3T states to execute this machine cycle.

I/O WRITE CYCLE OF 8085



EXAMPLE INSTRUCTION :

MVI B, 43



Reference Book:

- **Microprocessors and Microcomputer-Based System Design, Author: Mohamed Rafiquzzaman,**
- **Chapter-2**
 - **2.1-2.3,2.7-2.8.**
 - 2.5(only overview) [Not included for examination]



CSE 462I

Microprocessor and Interfacing

Class-22

8085 Microprocessor :
Bus Timing Diagram

TIMING DIAGRAM

- Timing Diagram is a graphical representation.
- It represents the execution time taken by each instruction in a graphical format.
- The execution time is represented in T-states.

Instruction cycle

- **Instruction:** A command given to the microprocessor to perform an operation
- **Program :** A set of instructions given in a sequential manner to perform a particular task

The CPU fetches one instruction from memory at a time & executes it.

Instruction cycle = Fetch cycle + Execute cycle

Fetch Cycle : The steps taken by CPU to fetch the opcode from the memory

The time taken for fetch cycle is fixed.

Execute Cycle : The steps taken by CPU to fetch data & to perform the operation specified in the instruction

The time taken for execute cycle is variable which depends on the type of instruction ,i.e. 3 –byte , 2-byte & 1-byte instruction.

INSTRUCTION CYCLE

- The time required to execute an instruction is called instruction cycle.

MACHINE CYCLE

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

T - states

Microprocessor performs an operation in specific clock cycles.

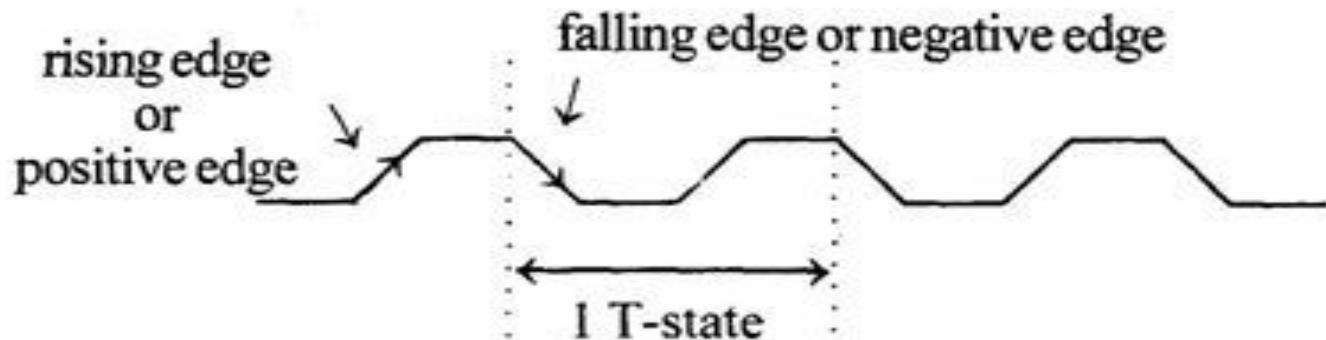
Each clock cycle is called as T –States.

The number of T – states required to perform an operation is called Machine Cycle .

Clock Period

- The **clock period** time, T , is the time between Falling edges of a repetitive **clock** signal. .

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



Machine cycle

The time required by the micro processor to complete the operation of accessing memory or I/O device .

Operations like :

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

MACHINE CYCLES OF 8085

The 8085 microprocessor has 5 basic machine cycles.

They are

1. Opcode fetch cycle (4T)
2. Memory read cycle (3 T)
3. Memory write cycle (3 T)
4. I/O read cycle (3 T)
5. I/O write cycle (3 T)

MPU Communication and Bus Timing

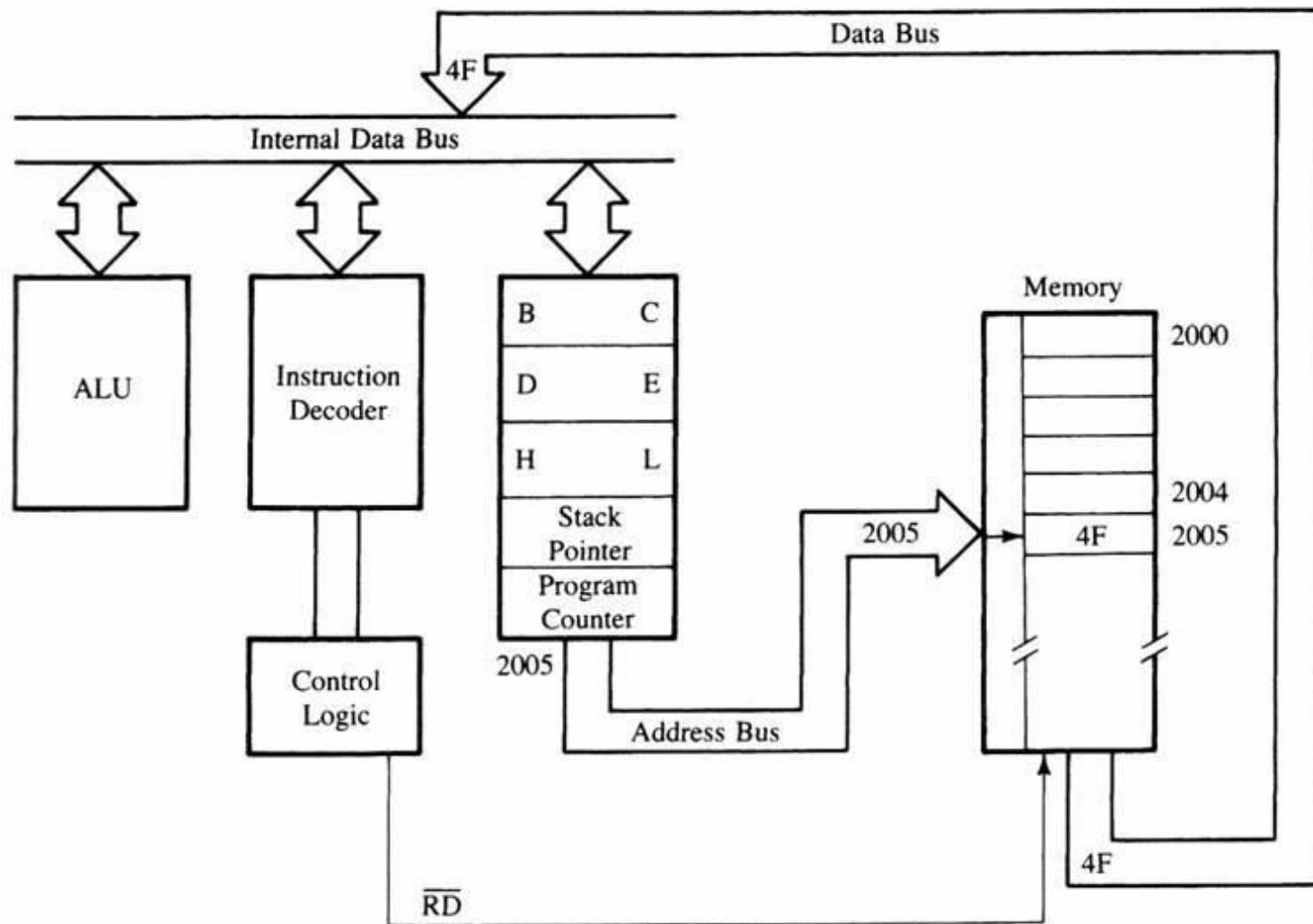


Figure 3: Moving data form memory to MPU using instruction **MOV C, A** (code machine 4FH = 0100 1111)

MPU Communication and Bus Timing

The Fetch Execute Sequence :

1. The µp placed a 16 bit memory address from PC (program counter) to address bus.

Figure 4: at T₁

- The high order address, 20H, is placed at A₁₅ – A₈.
 - the low order address, 05H, is placed at AD₇ - ADo and ALE is active high.
 - Synchronously the IO/M is in active low condition to show it is a memory operation.
2. At T₂ the active low control signal, RD, is activated so as to activate read operation; it is to indicate that the MPU is in fetch mode operation.

MPU Communication and Bus Timing

3. T3: The active low RD signal enabled the byte instruction, 4FH, to be placed on AD7 – AD0 and transferred to the MPU. While RD high, the data bus will be in high impedance mode.
4. T4: The machine code, 4FH, will then be decoded in instruction decoder. The content of accumulator (A) will then copied into C register at time state, T4.

S_0 and S_1

Pin 29 (Output) and Pin 33 (Output)

- S_0 and S_1 are called Status Pins.
- They tell the current operation which is in progress in 8085.

S_1	S_0	Operation
0	0	Halt
0	1	Write
1	0	Read
1	1	Opcode Fetch

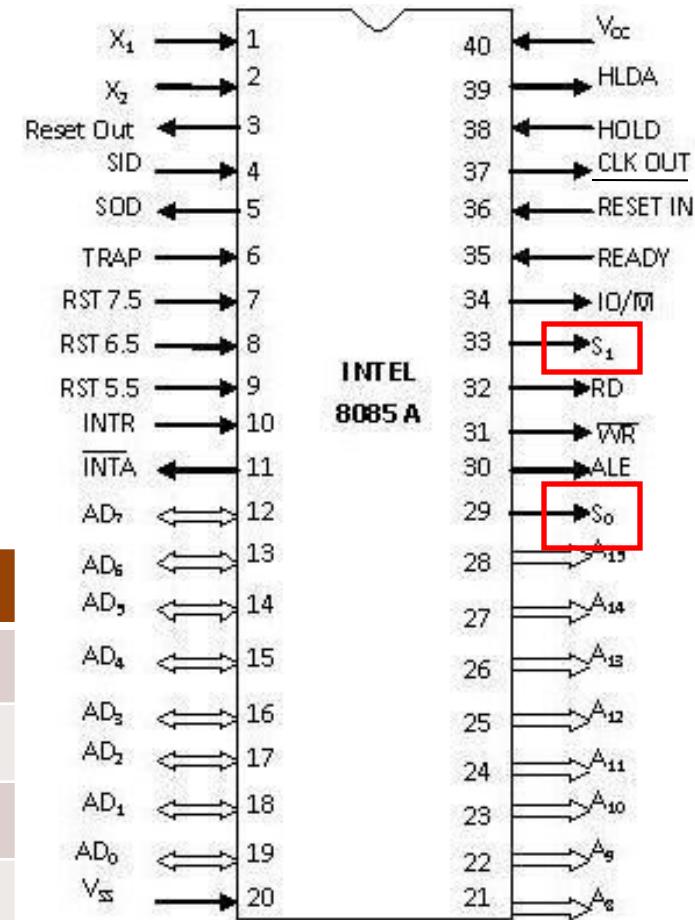


Table Showing $\text{IO}/\overline{\text{M}}$, S_0 , S_1 and Corresponding Operations

Operations	$\text{IO}/\overline{\text{M}}$	S_1	S_0
Opcode Fetch	0	1	1
Memory Read	0	1	0
Memory Write	0	0	1
I/O Read	1	1	0
I/O Write	1	0	1
Interrupt Ack.	1	1	1
Halt	High Impedance	0	0

MPU Communication and Bus Timing

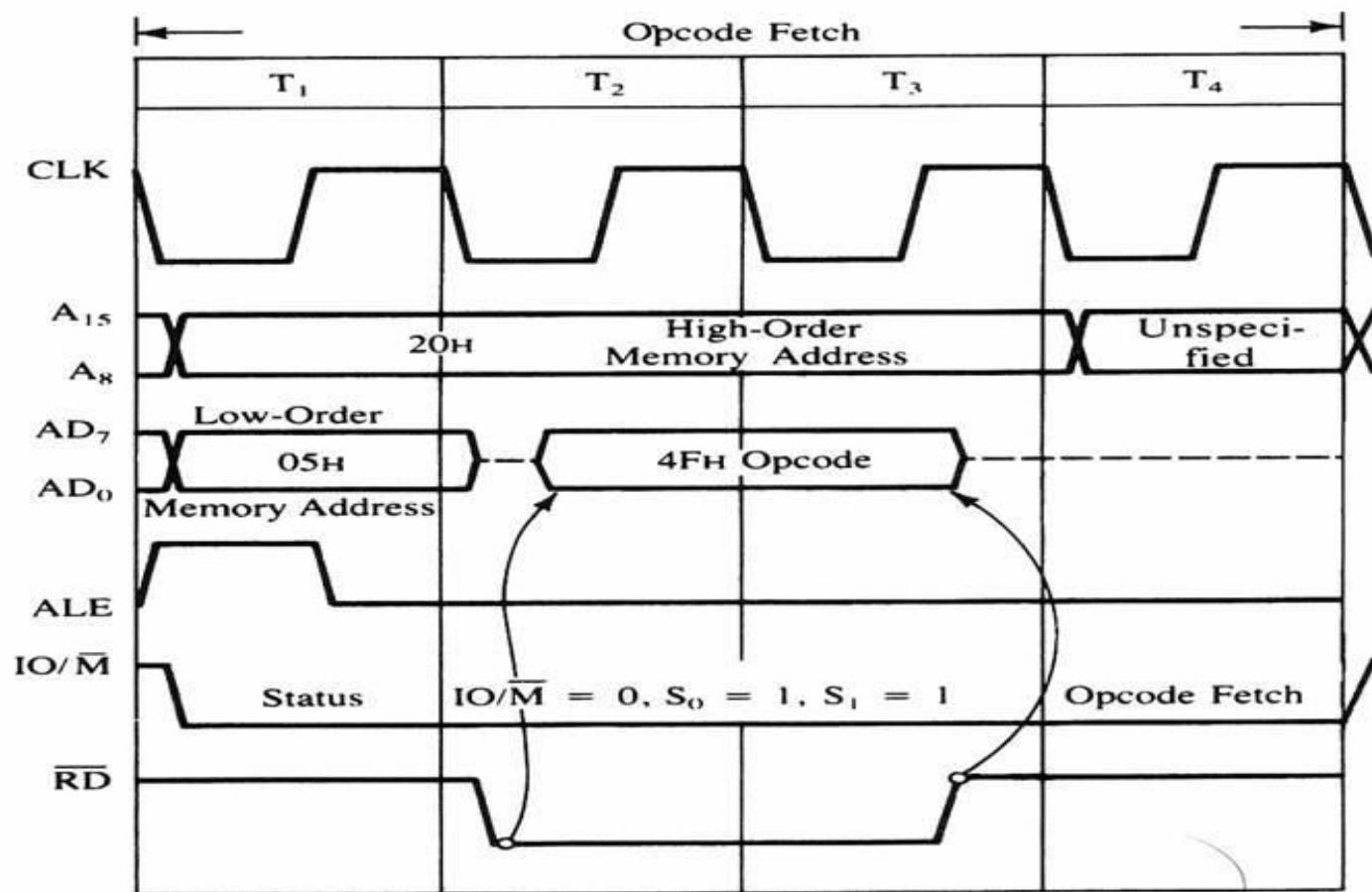
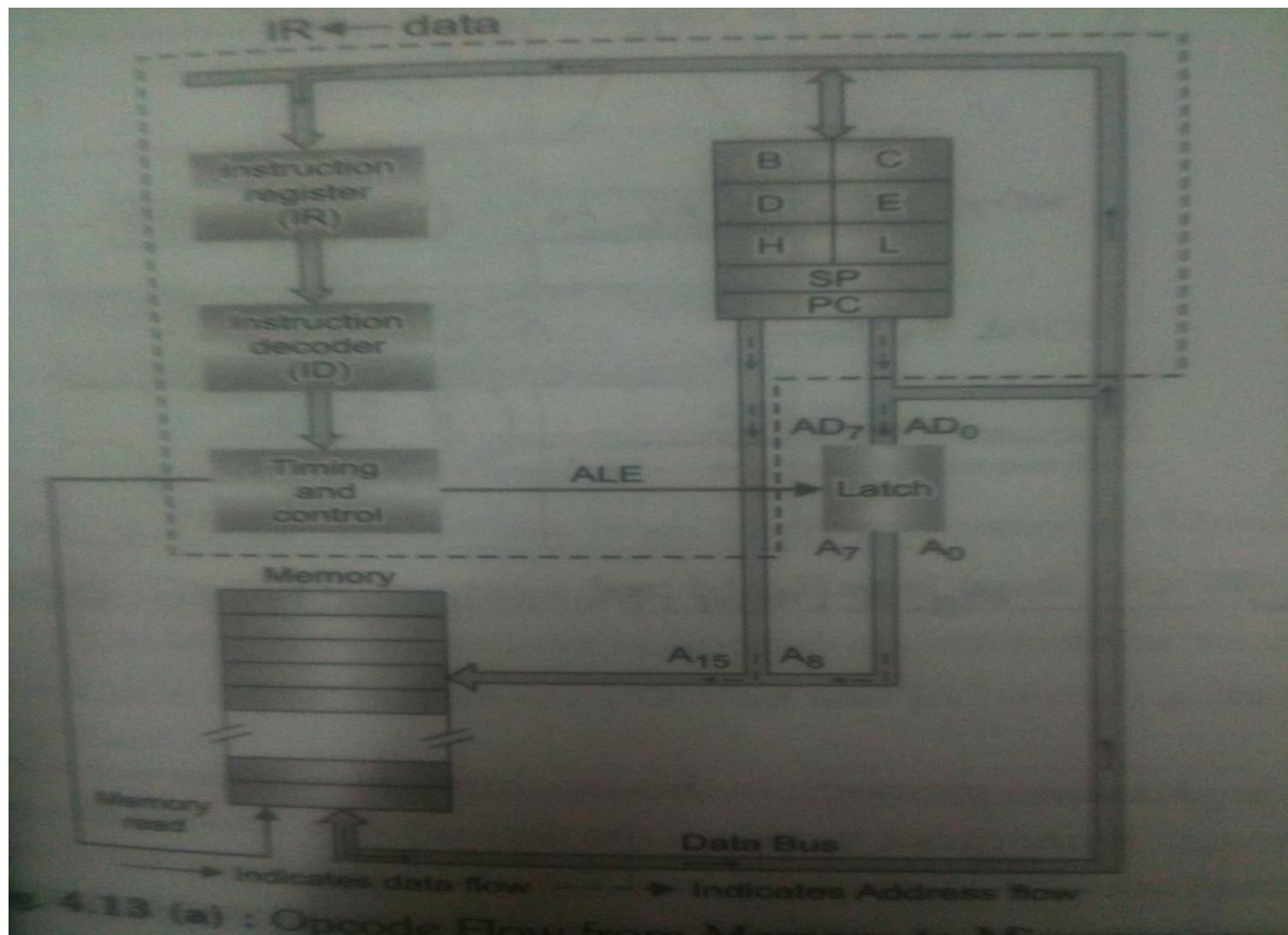
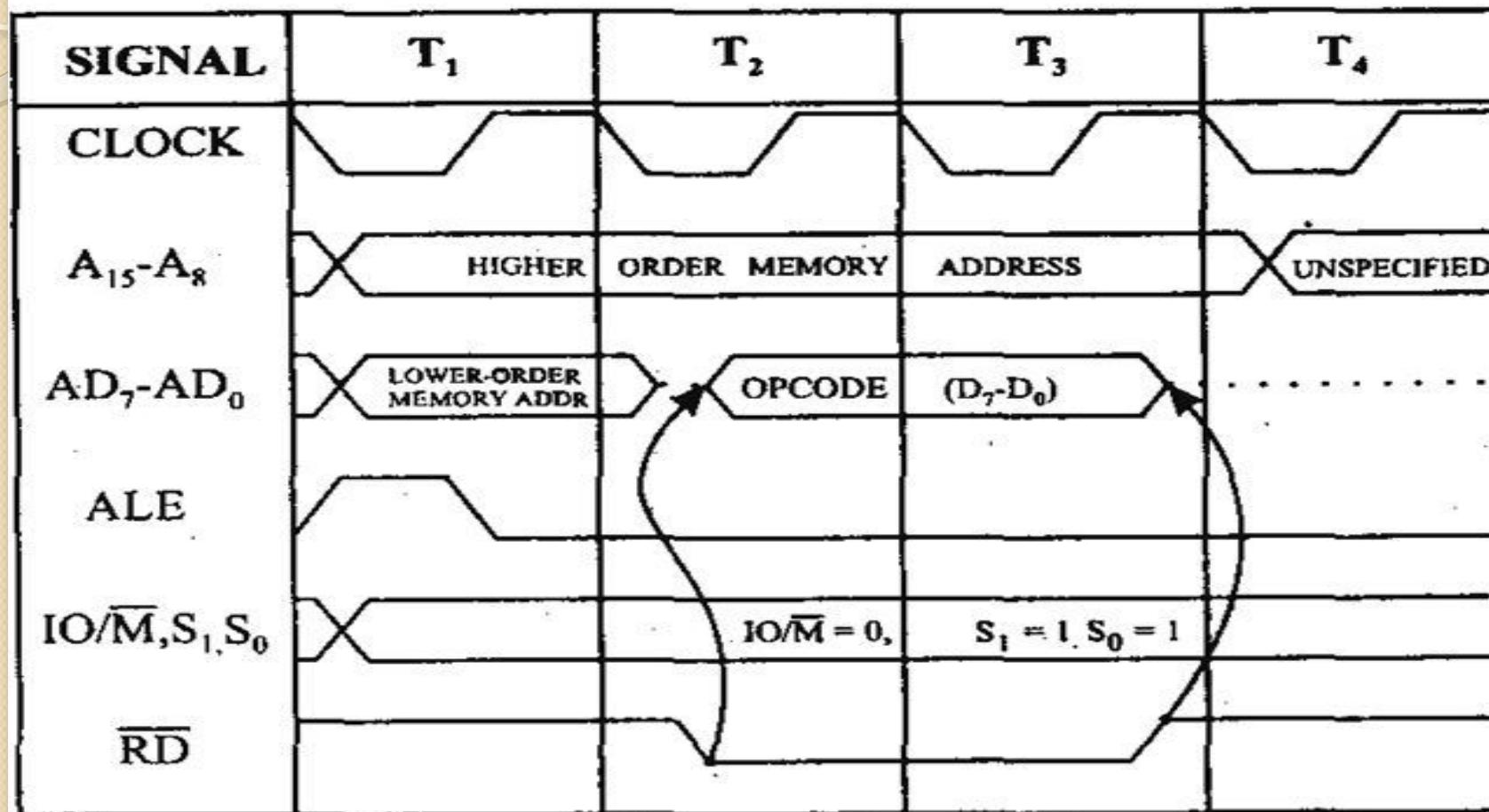


Figure 4: 8085 timing diagram for Opcode fetch cycle for **MOV C, A .**

OPCODE FETCH



OPCODE FETCH MACHINE CYCLE OF 8085



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.

MEMORY READ MACHINE CYCLE

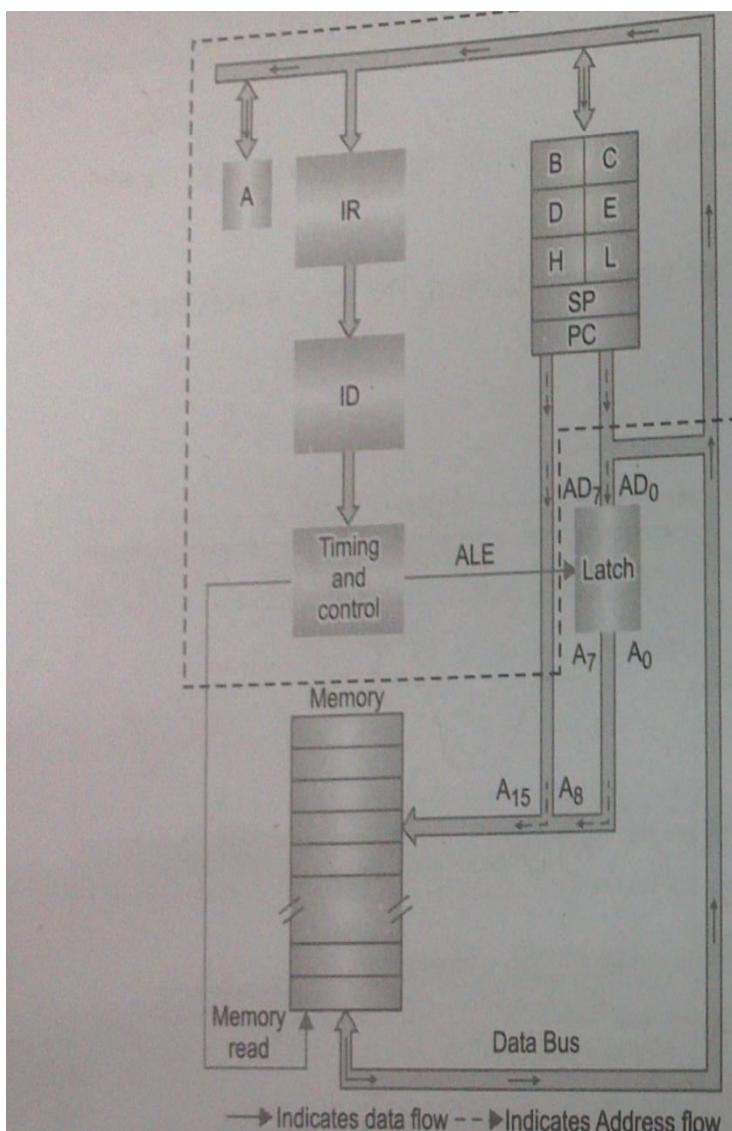


Fig. 4.14(a) : Data Flow from Memory to Microprocessor

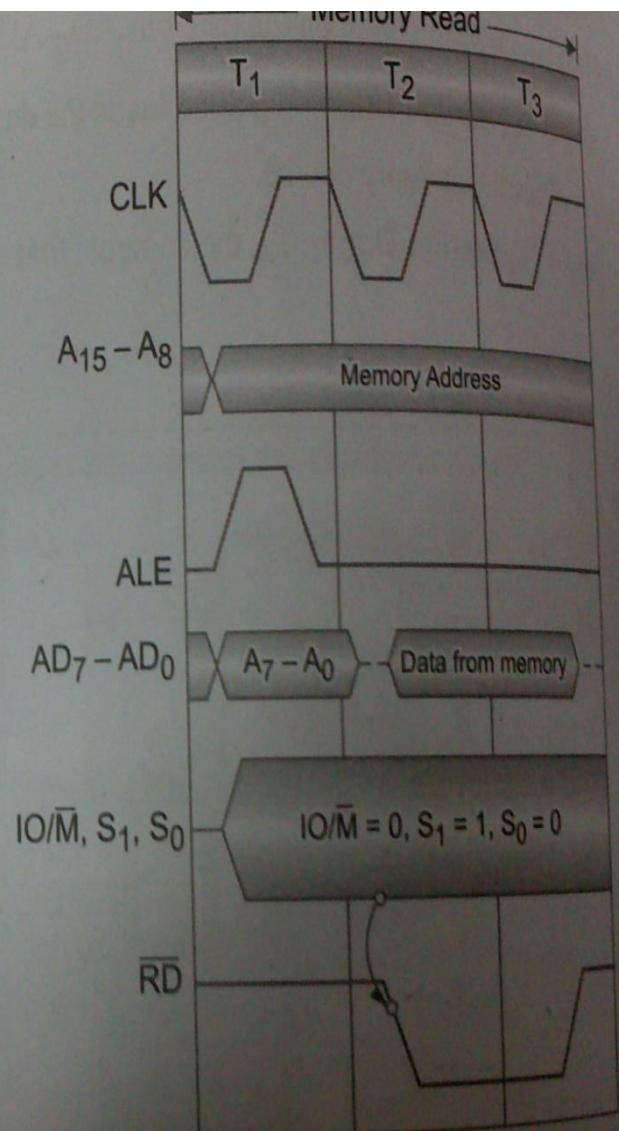
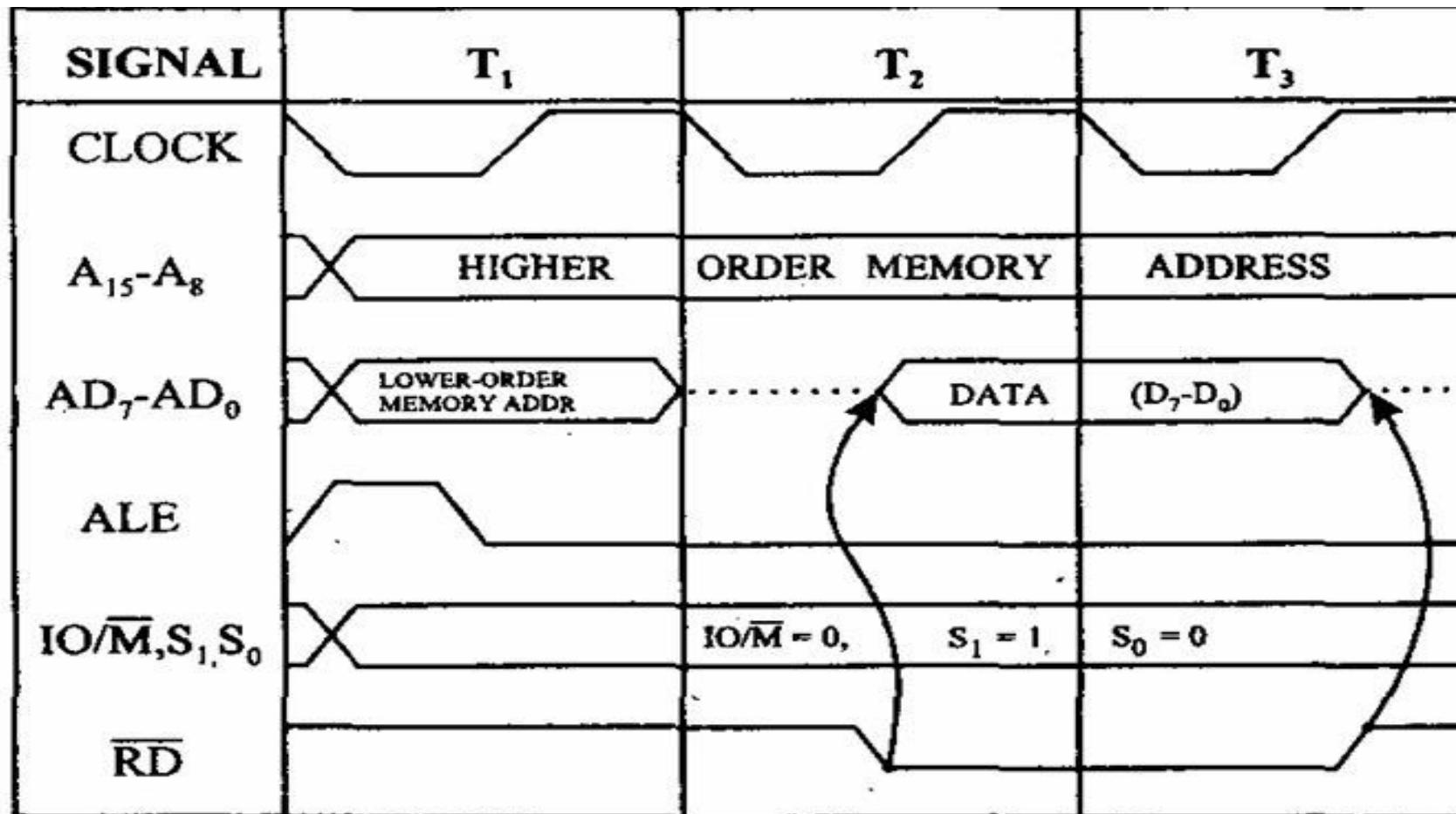


Fig. 4.14(b) : Memory Read Machine Cycle

MEMORY READ MACHINE CYCLE OF 8085



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.

Memory Read: The flow of data from the memory to the microprocessor.

Step 1 (T1) : microprocessor places the address on the address lines from program counter & activates ALE signal to multiplex the low order address.

It also sends status signals $\text{IO/M} = 0$, $S_1 = 1$, $S_0 = 0$ for memory read operation.

Step 2 (T2) : SE

MEMORY WRITE MACHINE CYCLE

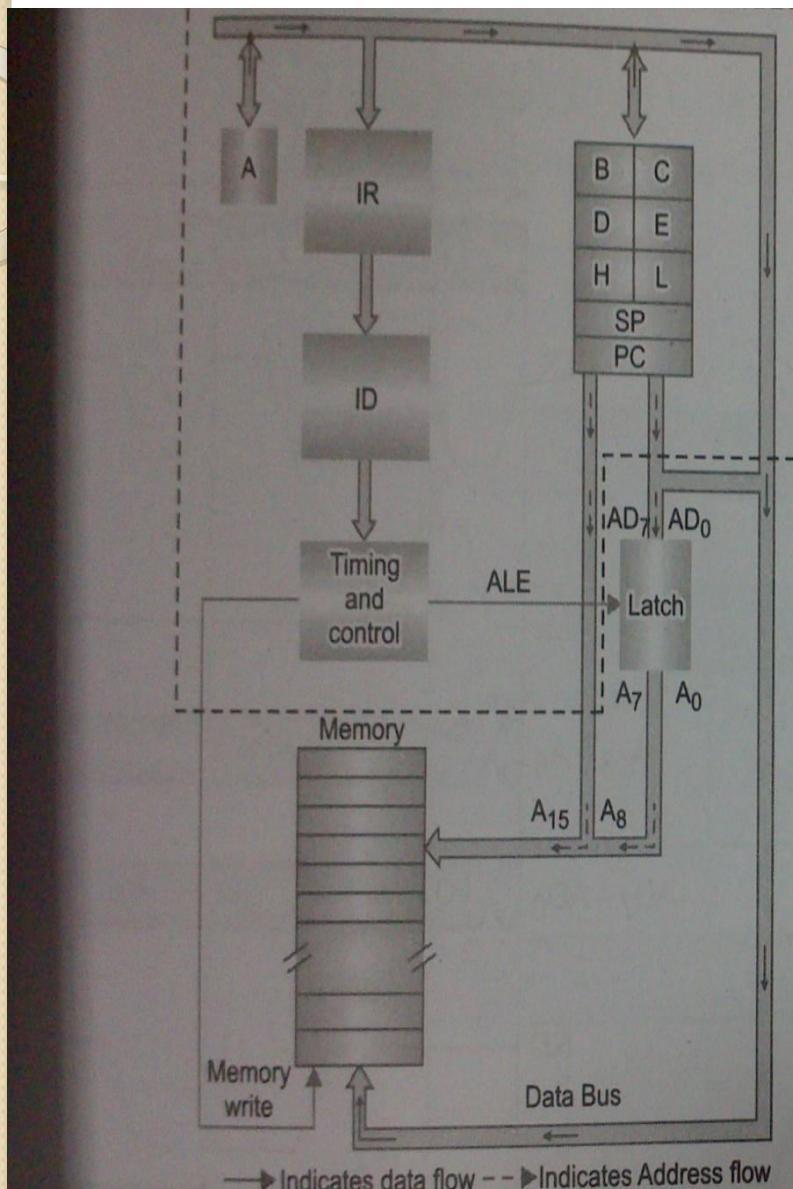


Fig. 4.15(a) : Data Flow from Microprocessor to Memory

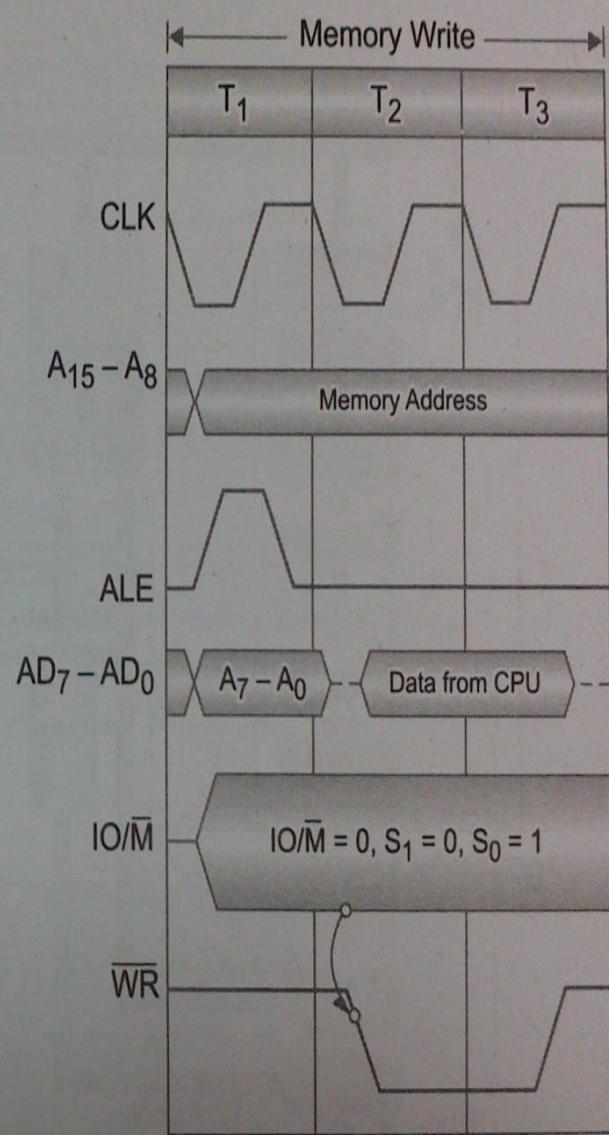
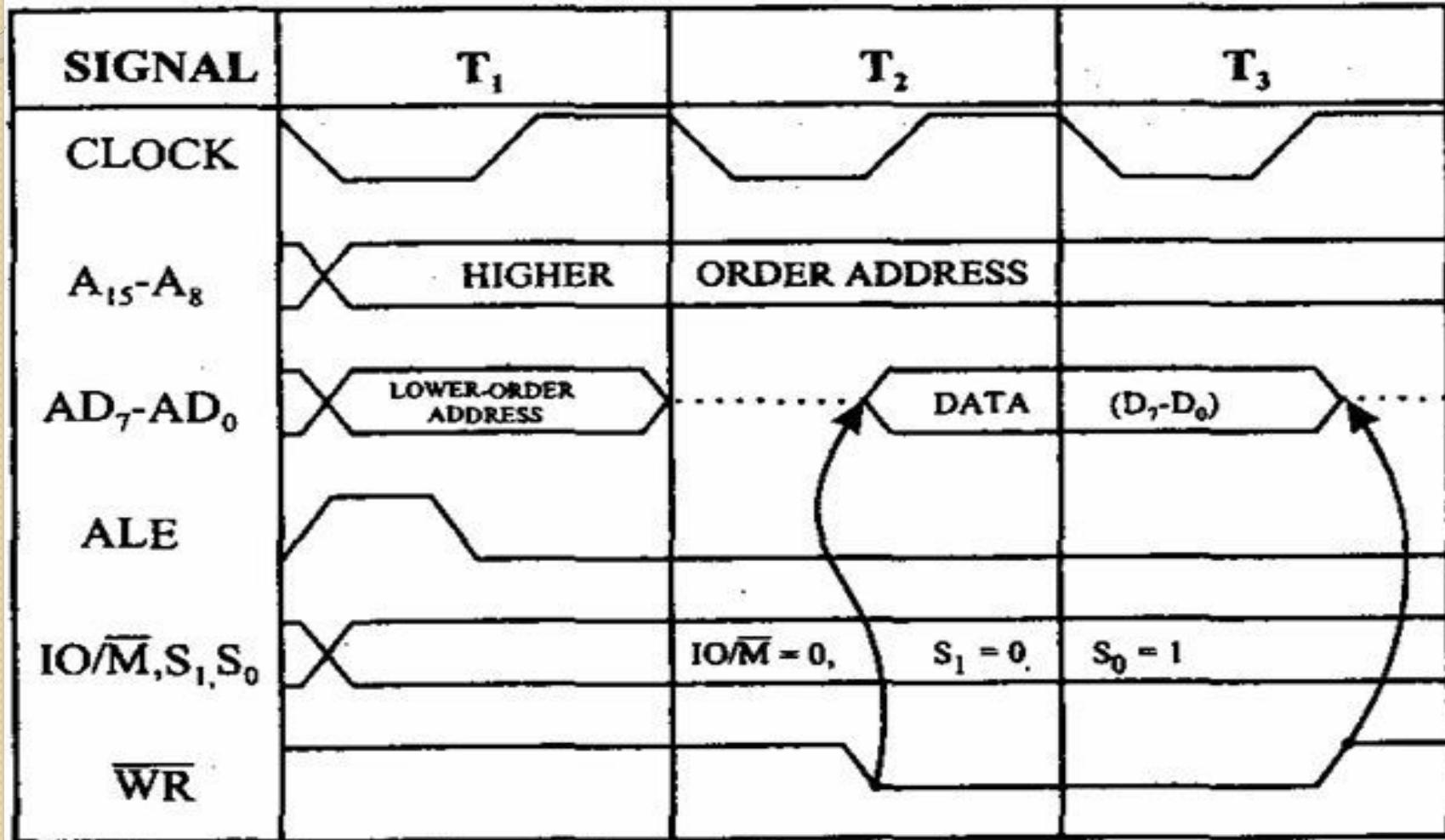


Fig. 4.15(b) : Memory Write Machine Cycle

MEMORY WRITE MACHINE CYCLE OF 8085



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

I/O READ MACHINE CYCLE

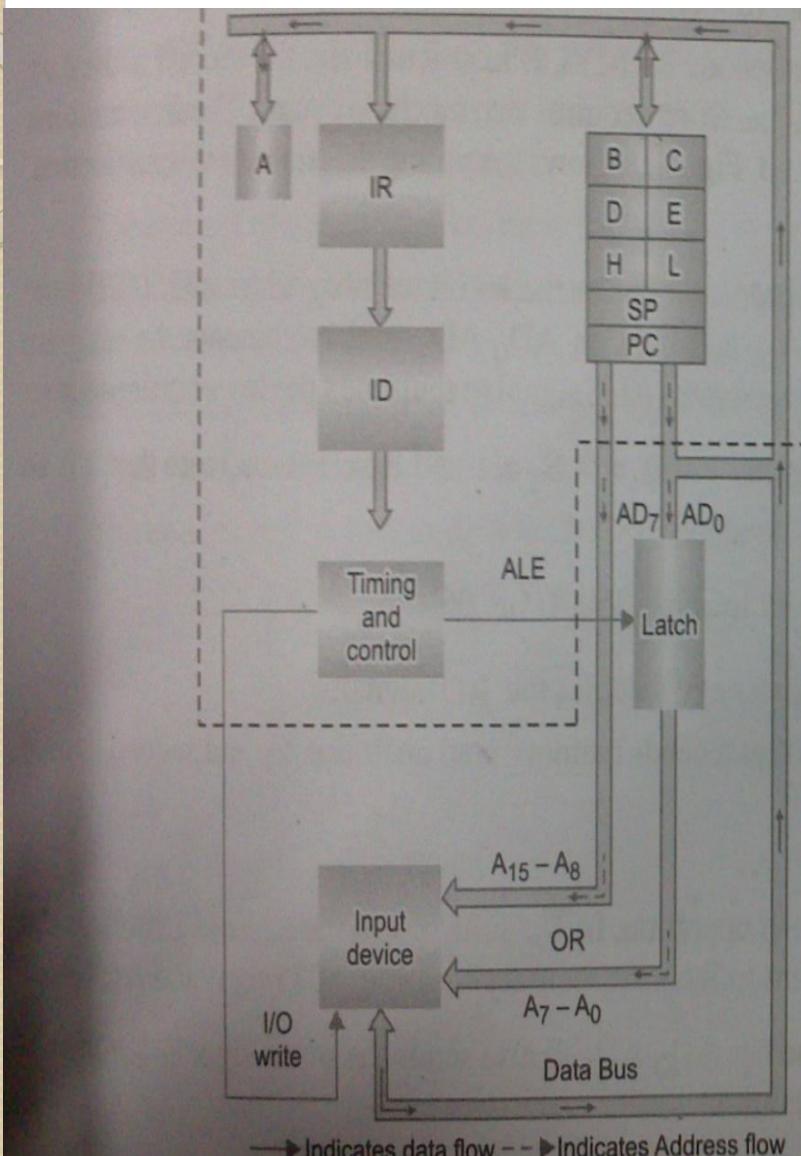


Fig. 4.16(a) : Data Flow from Input Device to Microprocessor

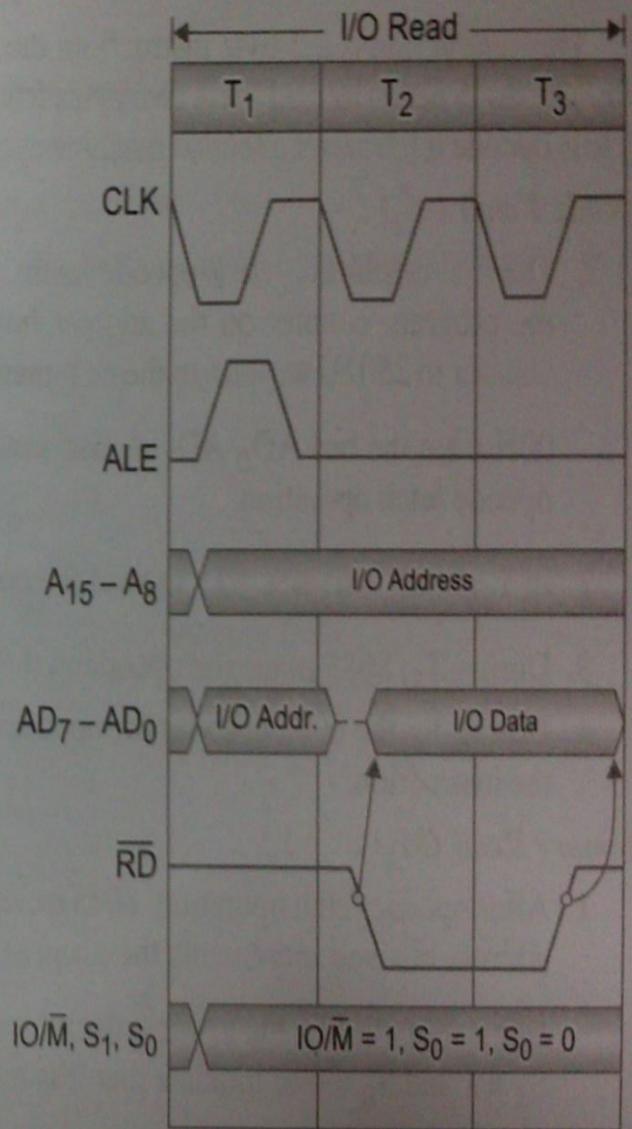
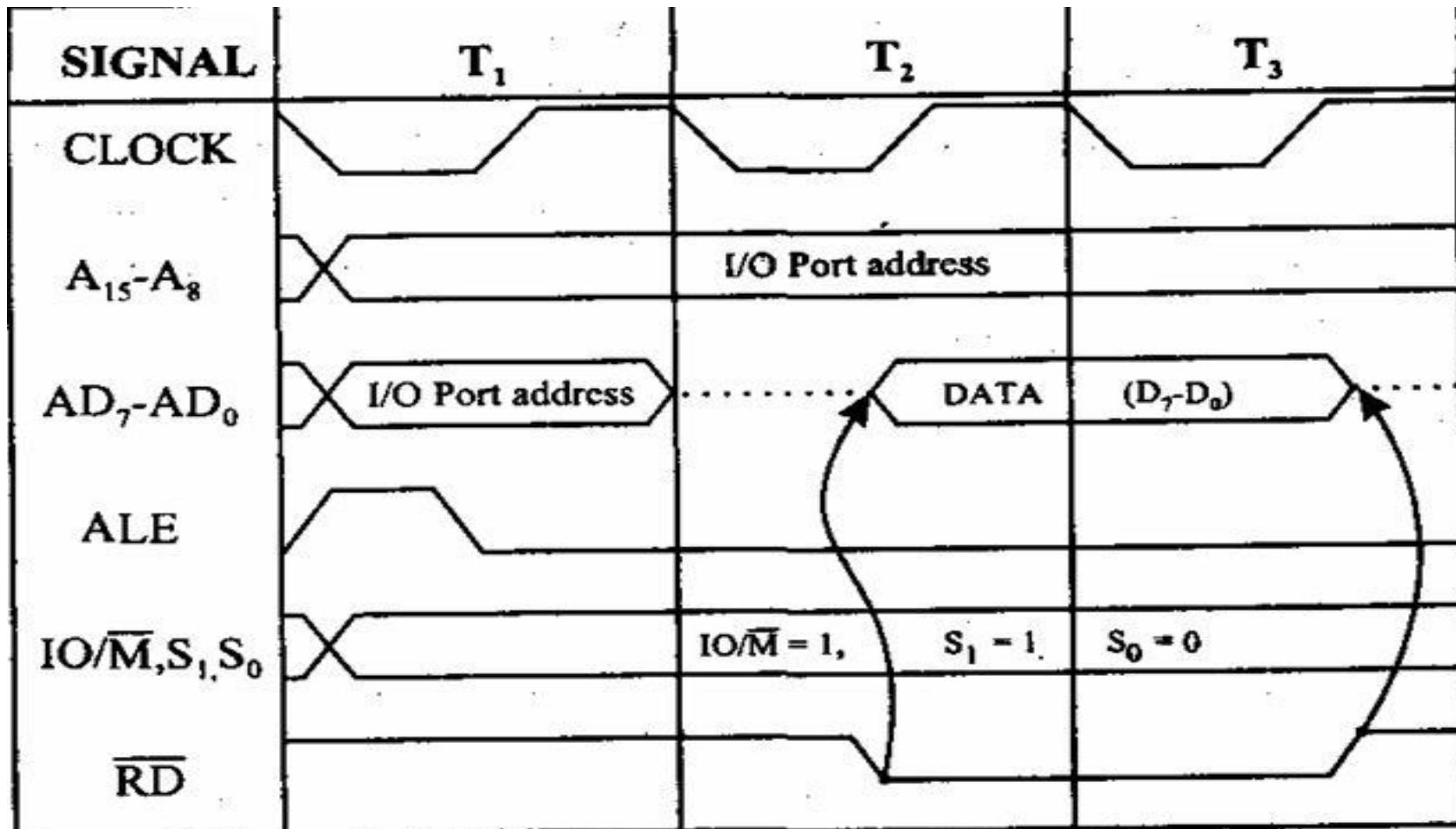


Fig. 4.16(b) : I/O Read Machine Cycle

I/O READ CYCLE OF 8085

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

I/O READ CYCLE OF 8085



I/O WRITE MACHINE CYCLE

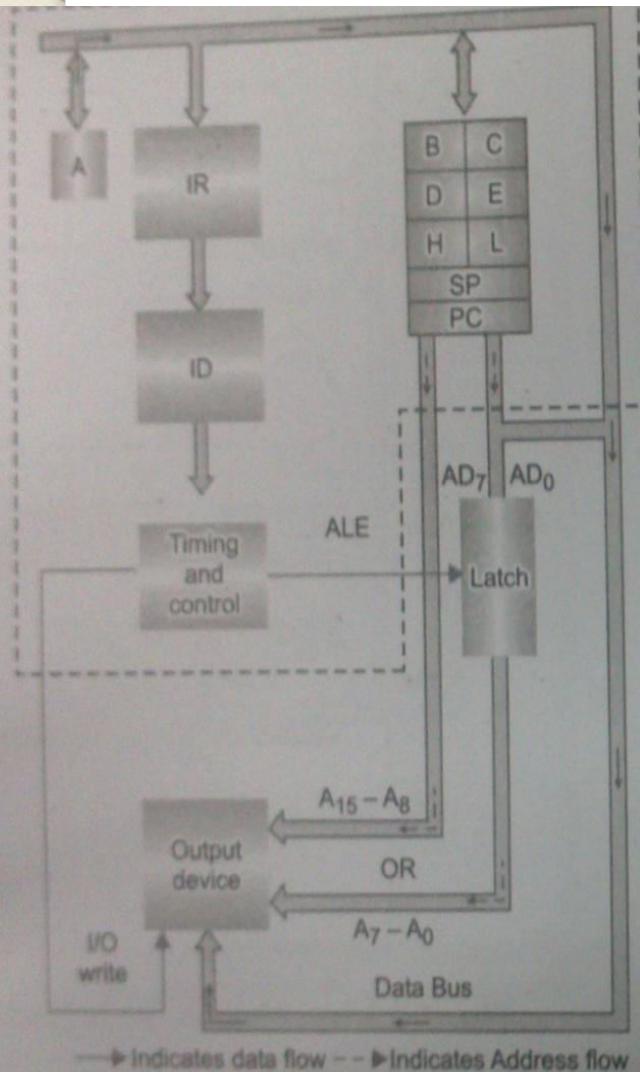


Fig. 4.17(a) : Data Flow from Microprocessor to Output Device

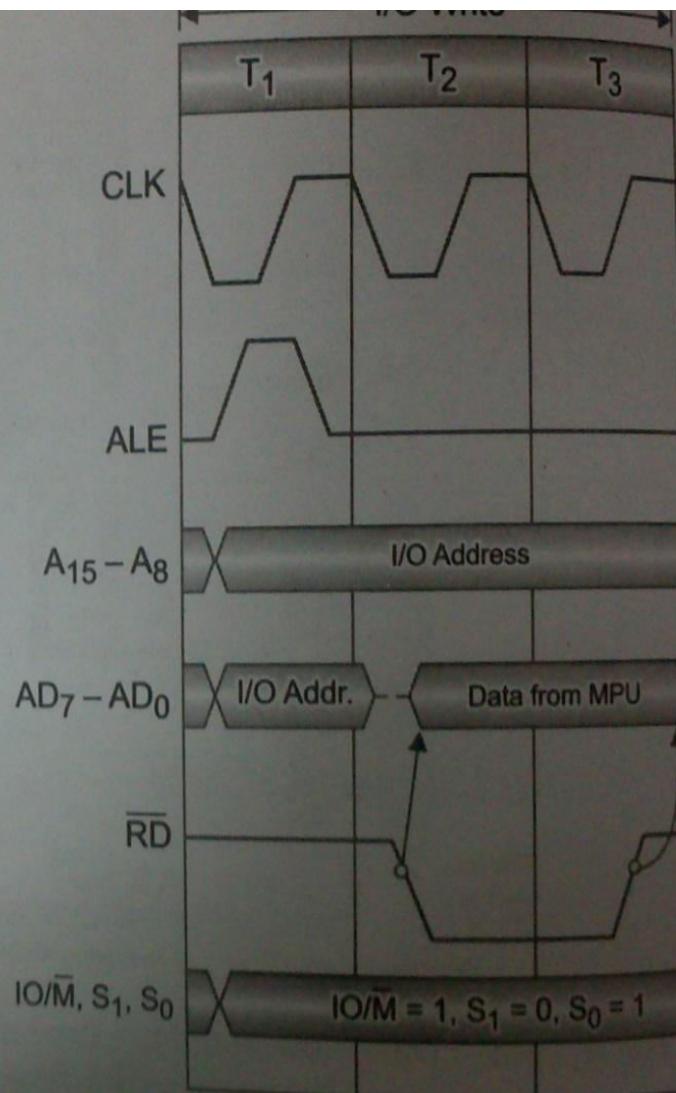
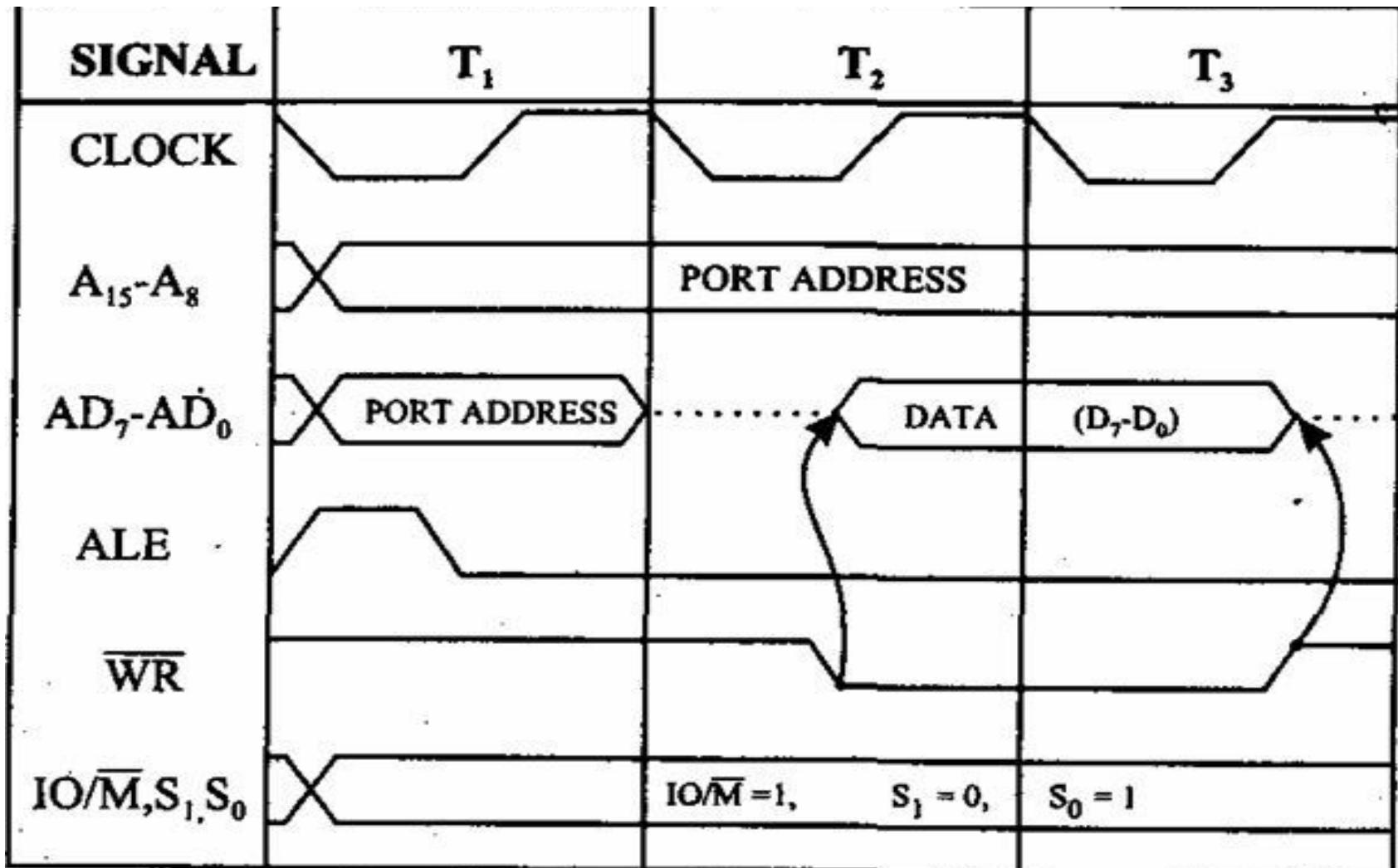


Fig. 4.17(b) : I/O Write Machine Cycle

I/O WRITE CYCLE OF 8085

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

I/O WRITE CYCLE OF 8085



Instruction Set

Instruction Format

Timing Diagram

Instruction:

A000h MOV A,B

Corresponding Coding:

A000h 78

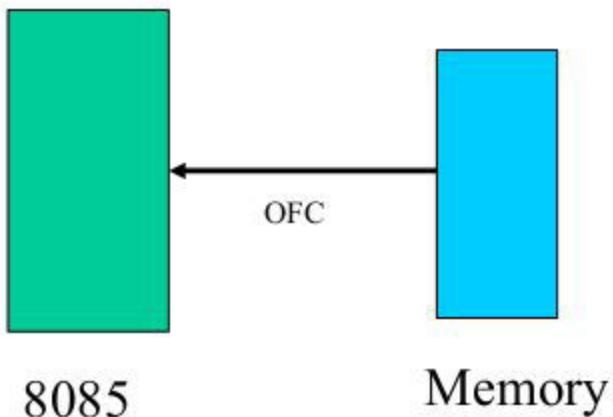
Timing Diagram

Instruction:

A000h MOV A,B

Corresponding Coding:

A000h 78



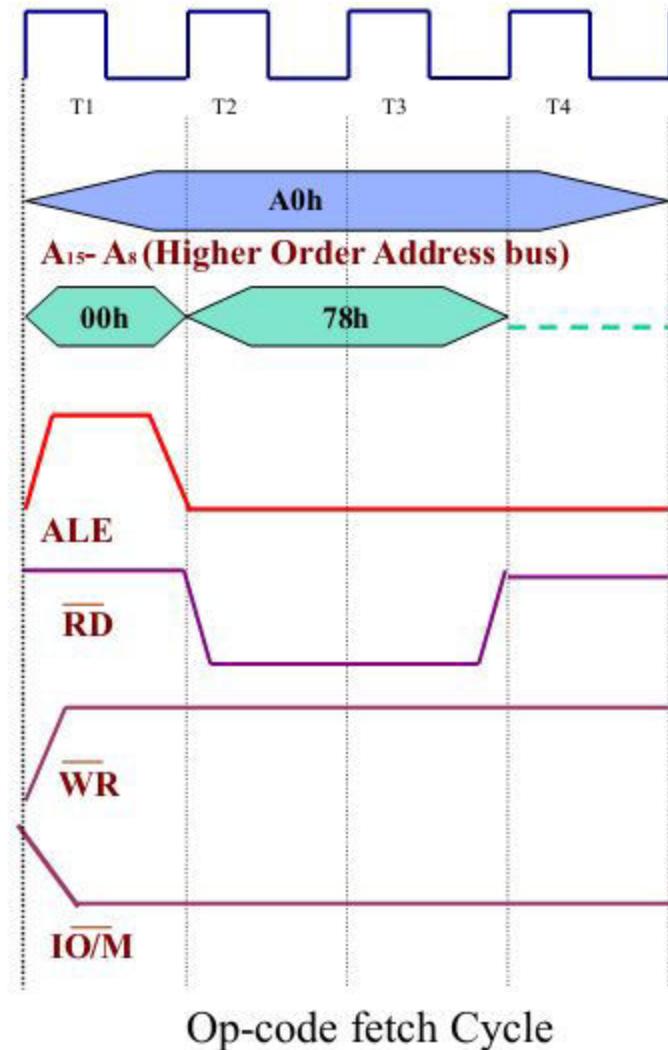
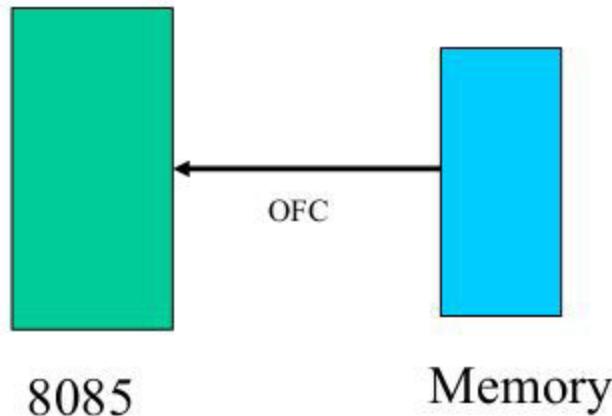
Timing Diagram

Instruction:

A000h MOV A,B

Corresponding Coding:

A000h 78



Timing Diagram

Instruction:

A000h MVI A,45h

Corresponding Coding:

A000h 3E

A001h 45

Timing Diagram

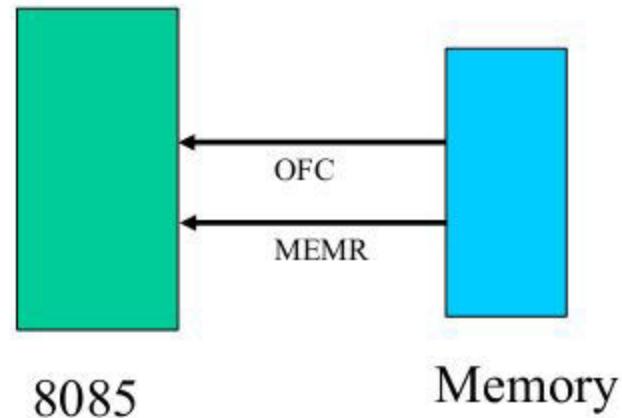
Instruction:

A000h MVI A,45h

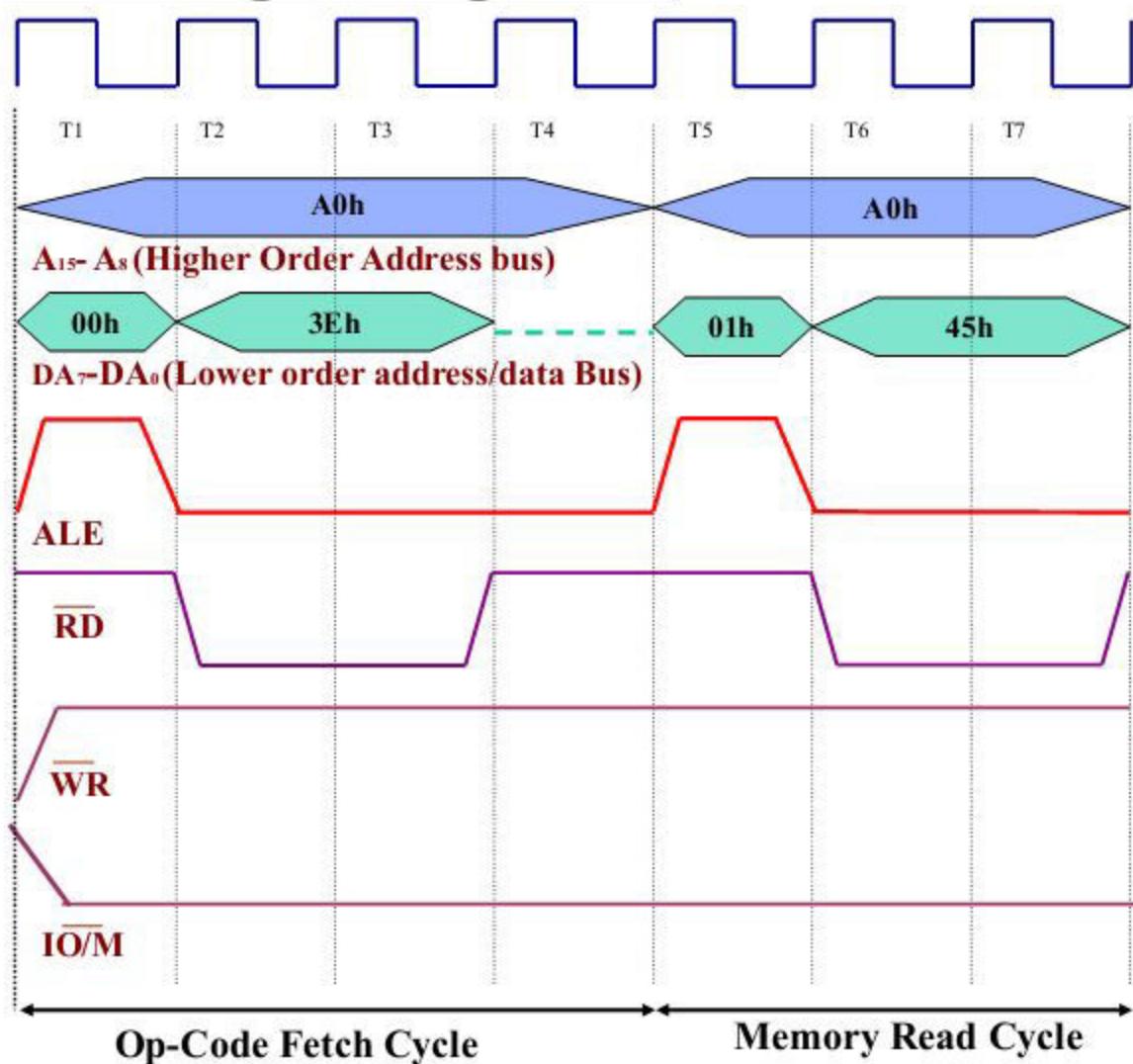
Corresponding Coding:

A000h 3E

A001h 45



Timing Diagram



Instruction:

A000h MVI A,45h

Corresponding Coding:

A000h 3E

A001h 45

Timing Diagram

Instruction:

A000h LXI A,FO45h

Corresponding Coding:

A000h 21

A001h 45

A002h F0

Timing Diagram

Instruction: **LXI H, Data**

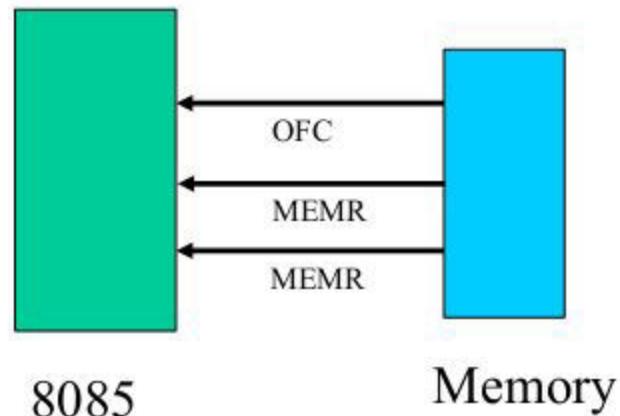
A000h LXI A,FO45h

Corresponding Coding:

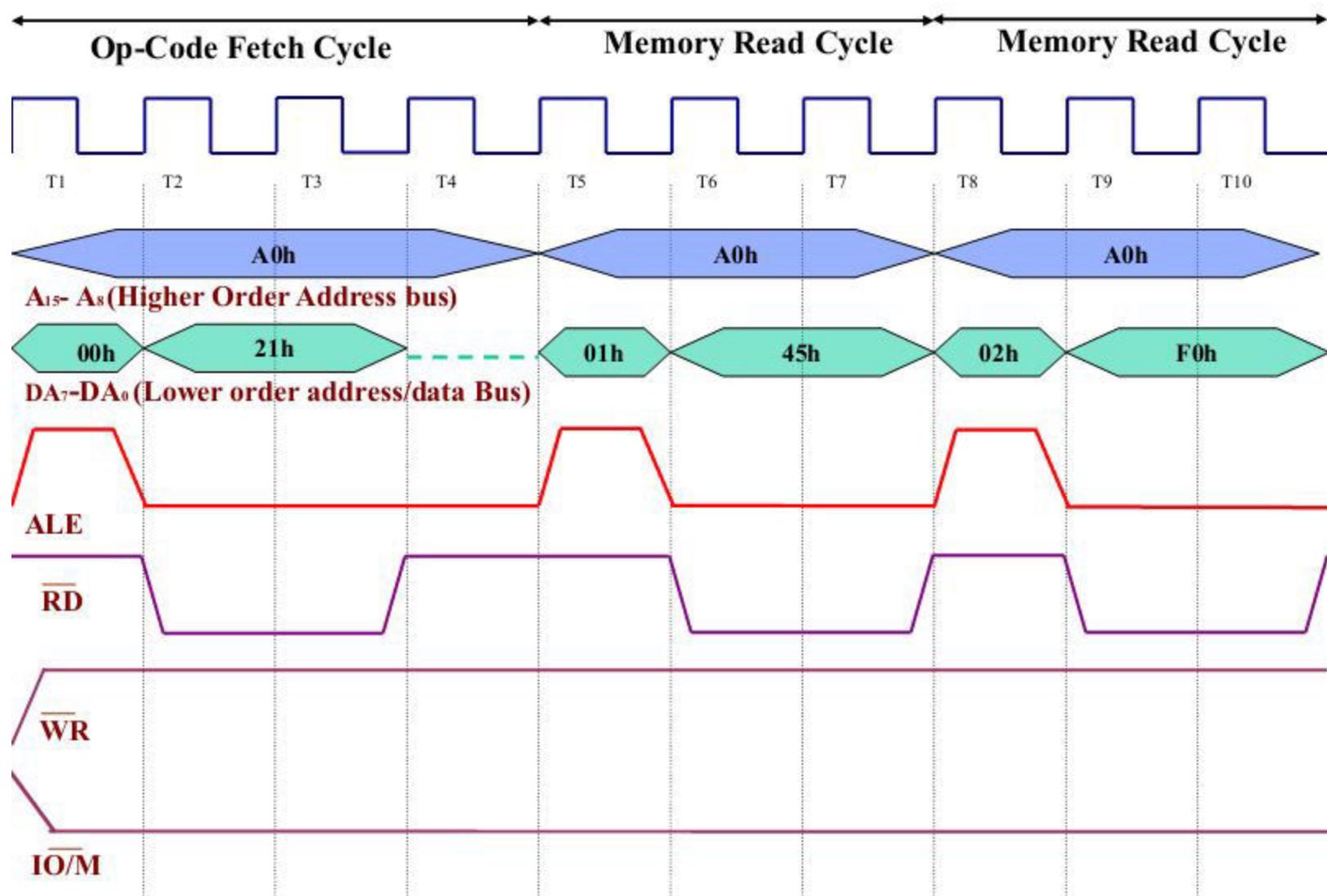
A000h 21

A001h 45

A002h F0



Timing Diagram



Timing Diagram

Instruction:

A000h MOV A,M

Corresponding Coding:

A000h 7E

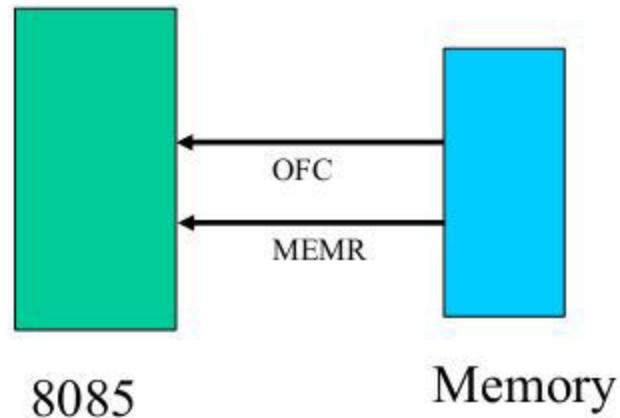
Timing Diagram

Instruction:

A000h MOV A,M

Corresponding Coding:

A000h 7E



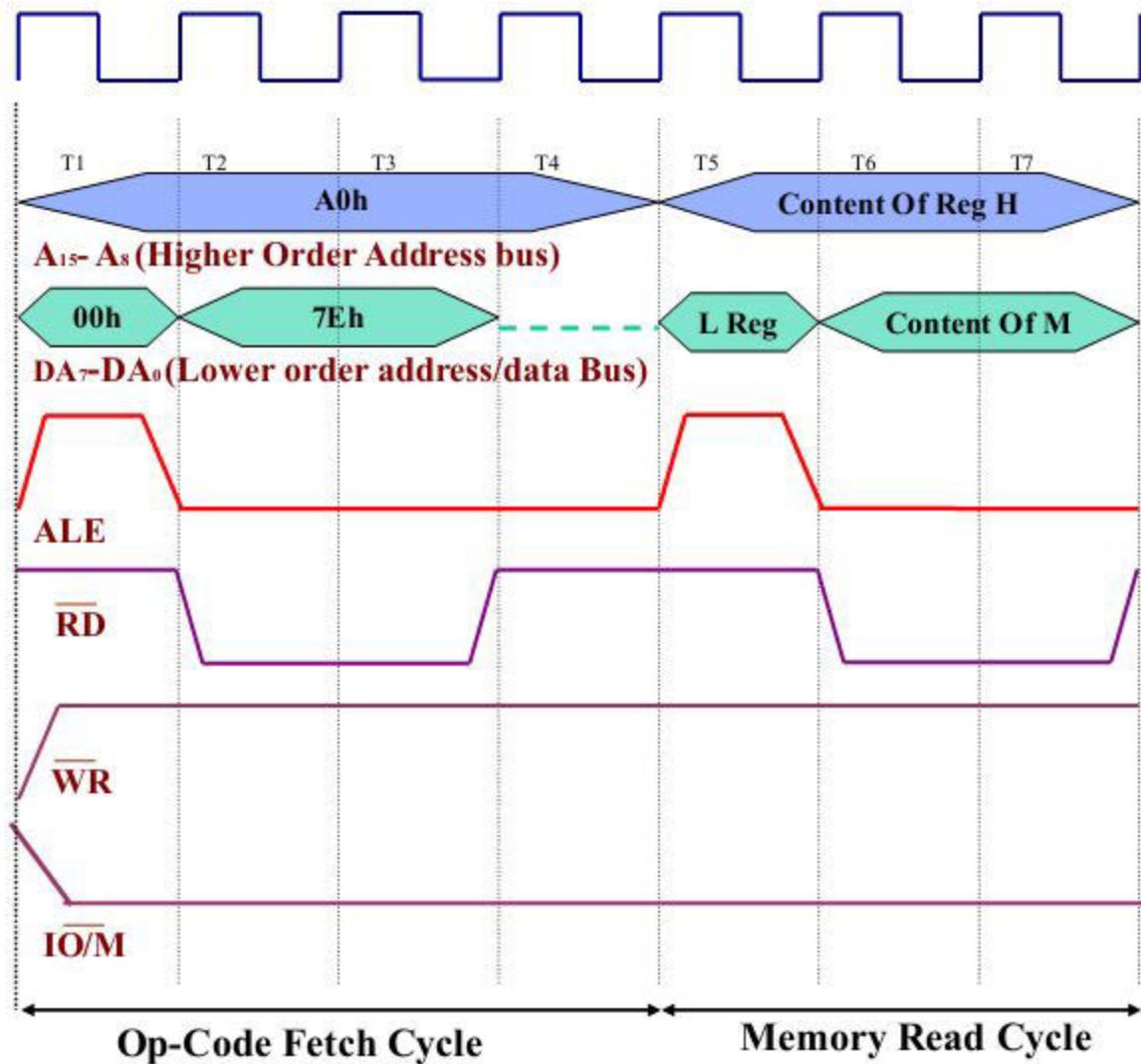
Timing Diagram

Instruction:

A000h MOV A,M

Corresponding Coding:

A000h 7E



Timing Diagram

Instruction:

A000h MOV M,A

Corresponding Coding:

A000h 77

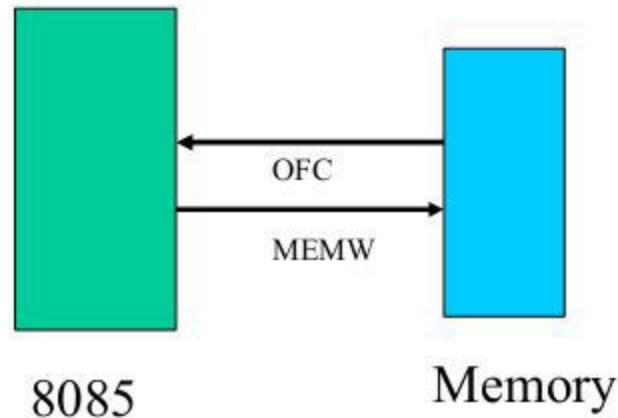
Timing Diagram

Instruction:

A000h MOV M,A

Corresponding Coding:

A000h 77



Timing Diagram

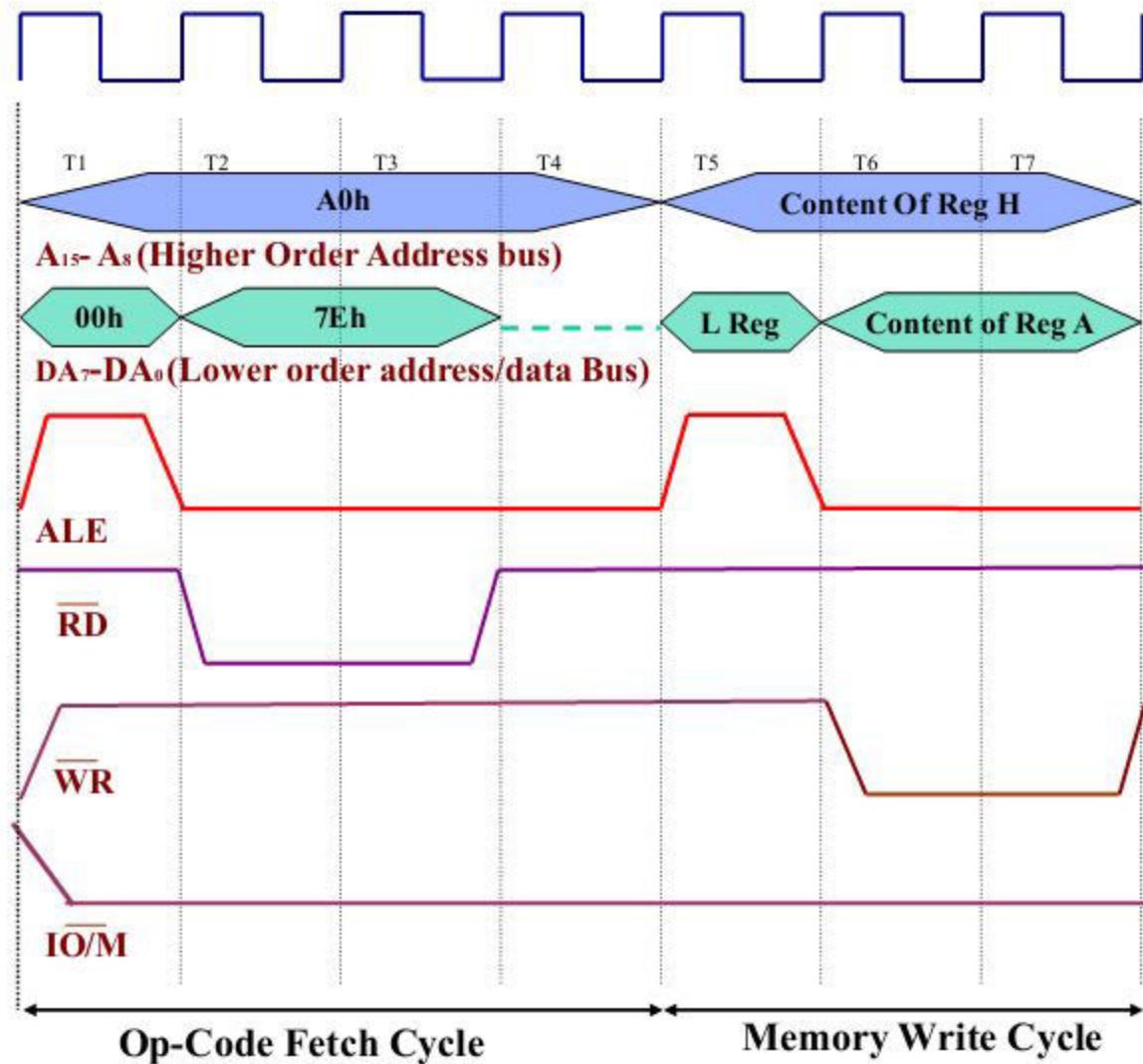
Instruction:

A000h MOV M,A

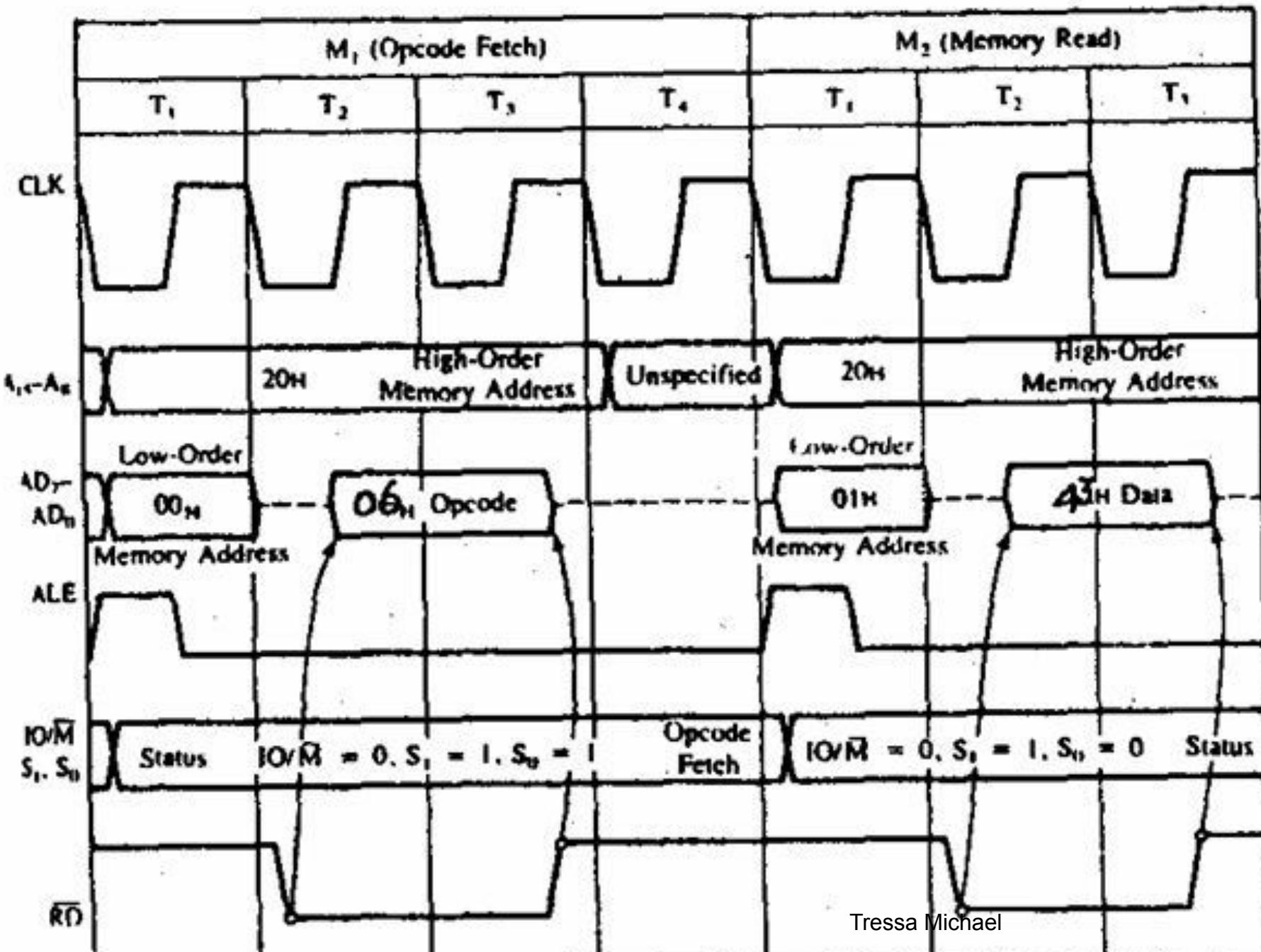
Corresponding Coding:

A000h

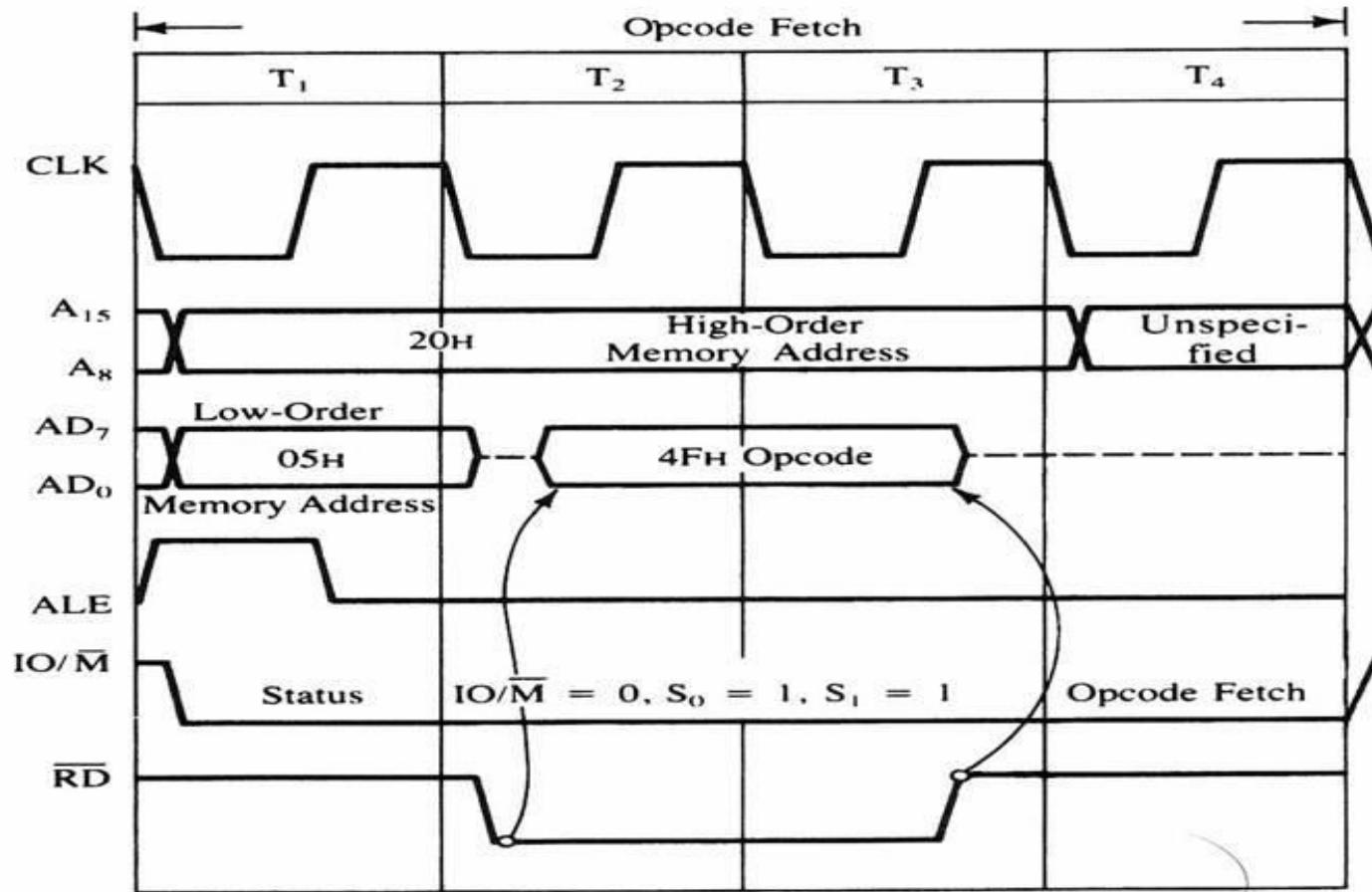
77



MVI B, data



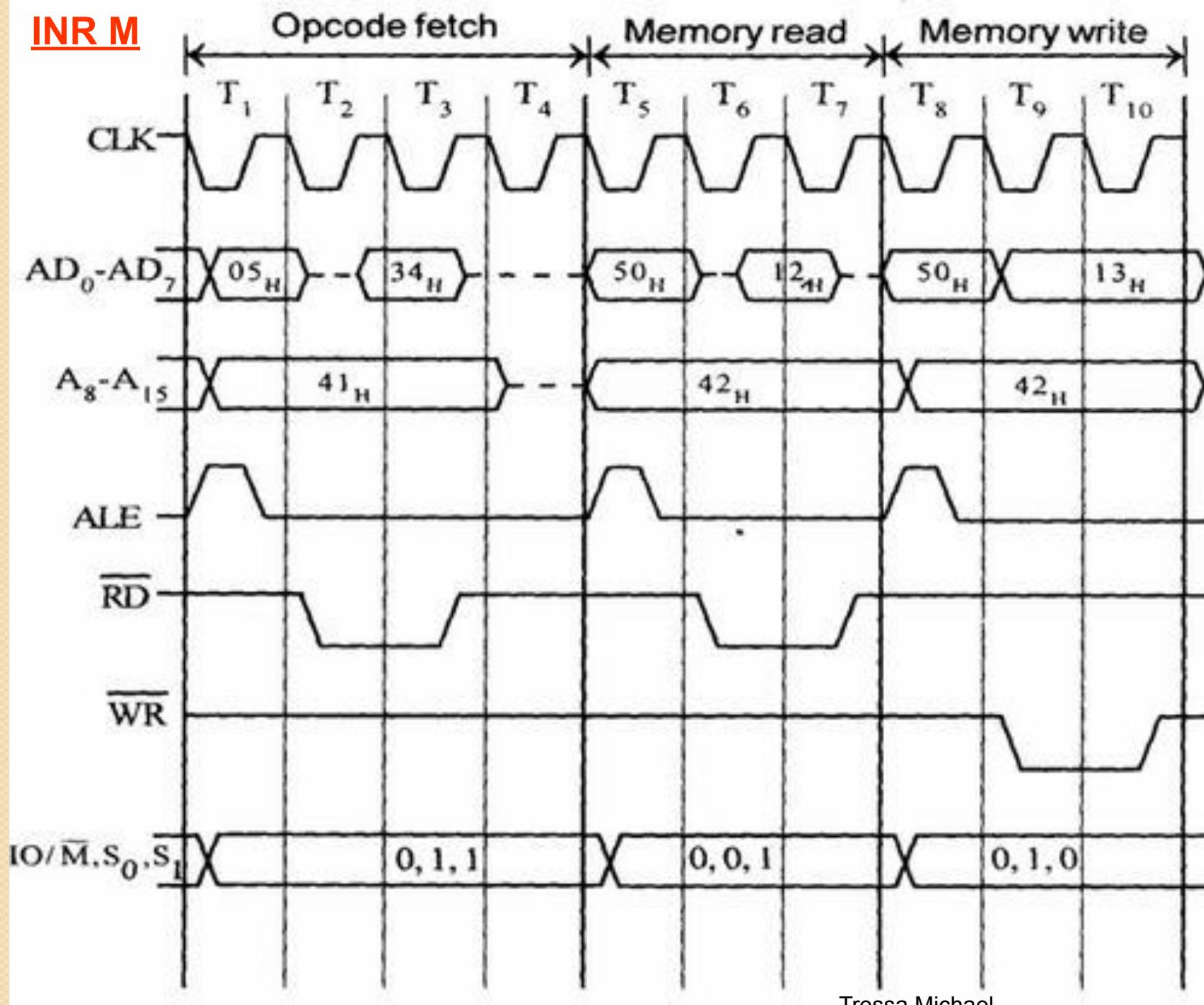
8085 timing diagram for Opcode fetch cycle for MOV C, A .

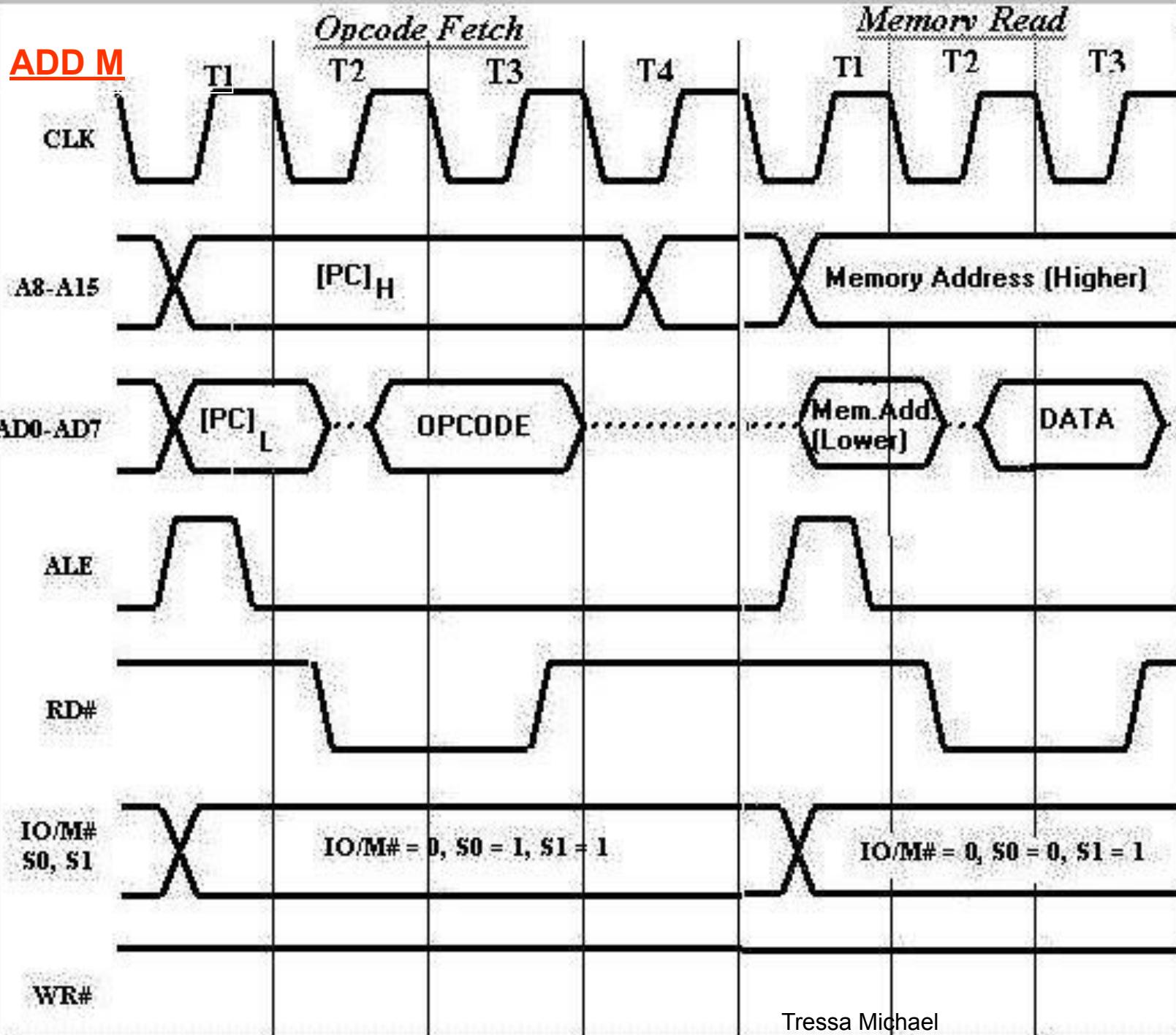


8085 timing diagram for Opcode fetch cycle for MOV C, A .

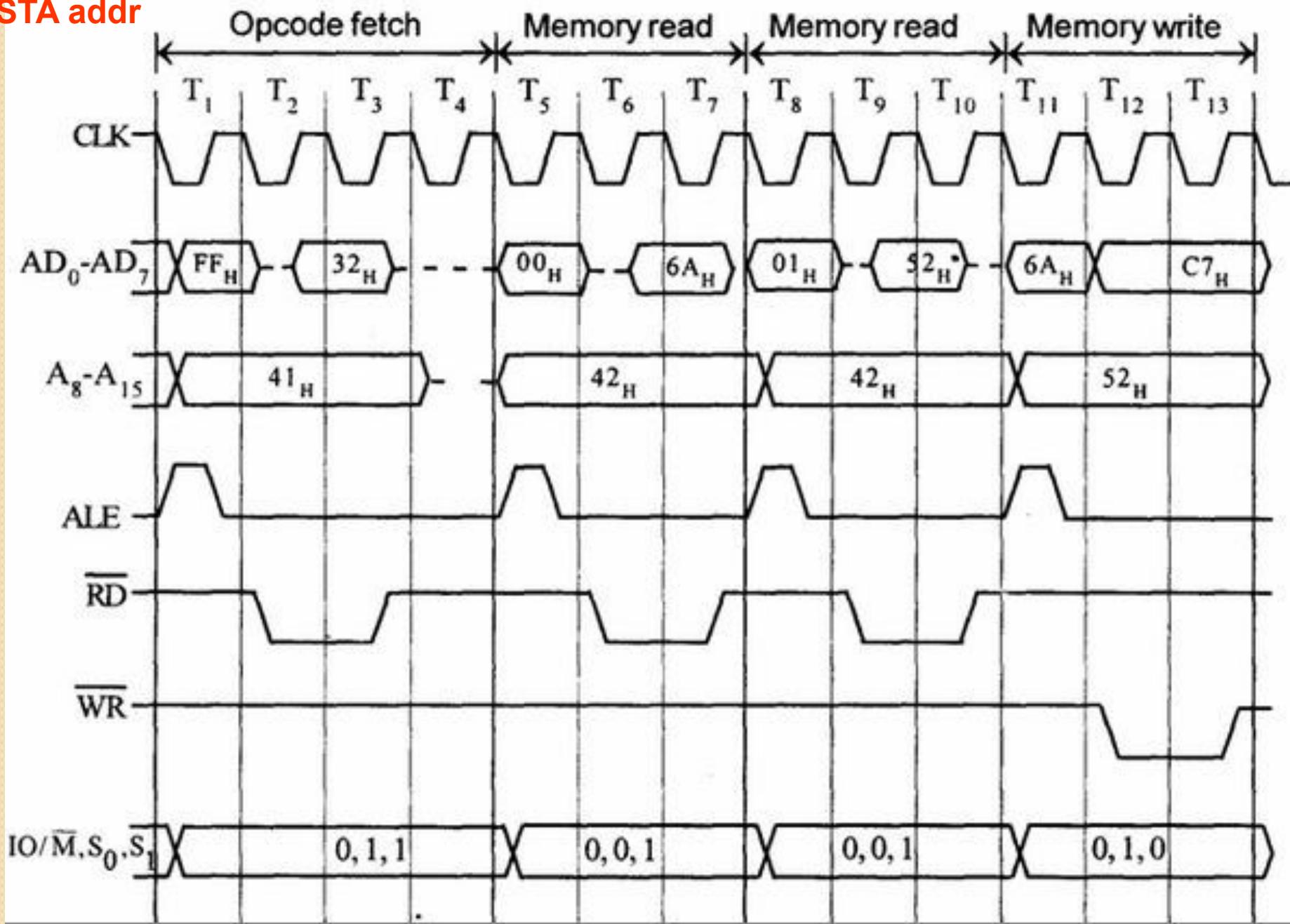
Tressa Michael

INR M

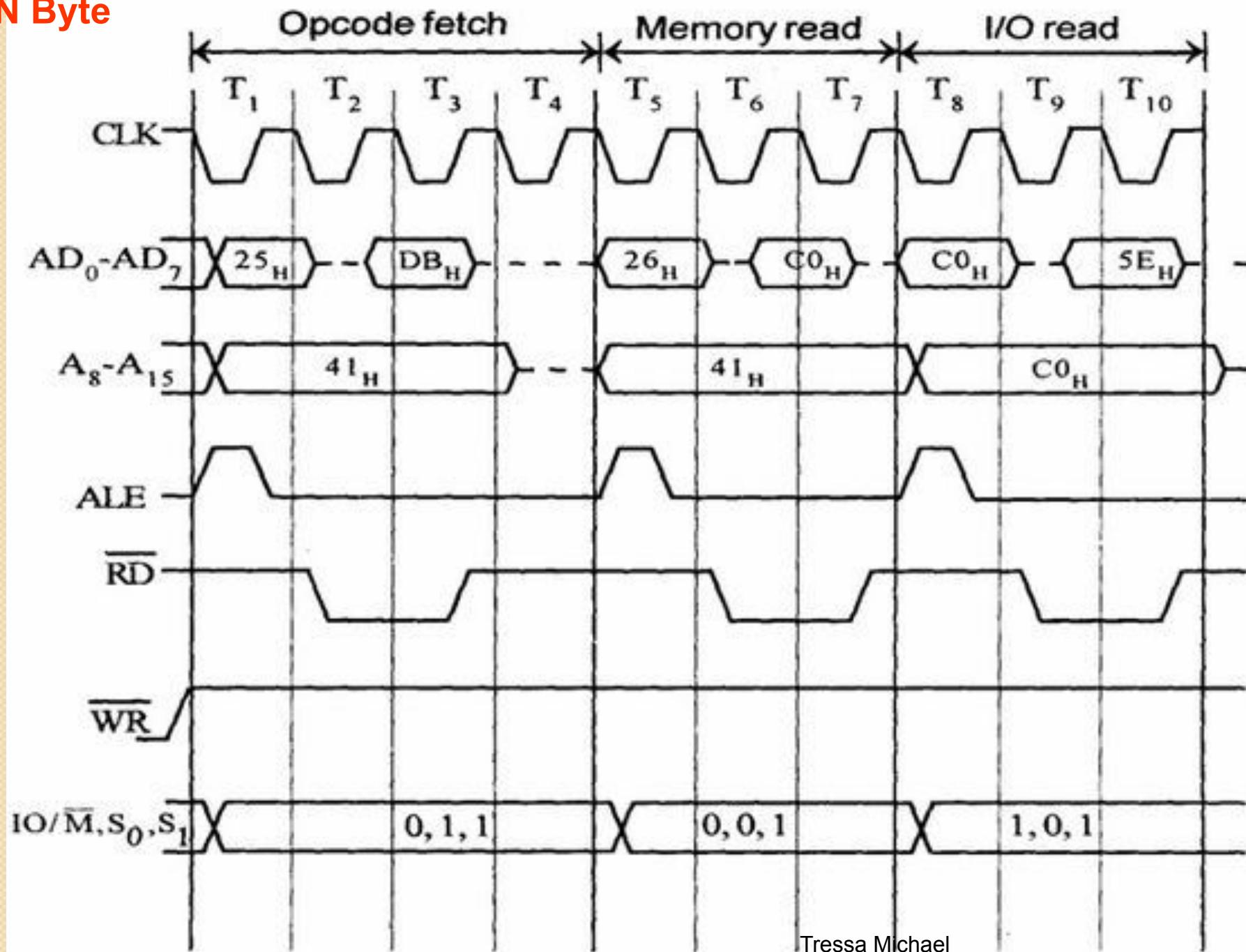


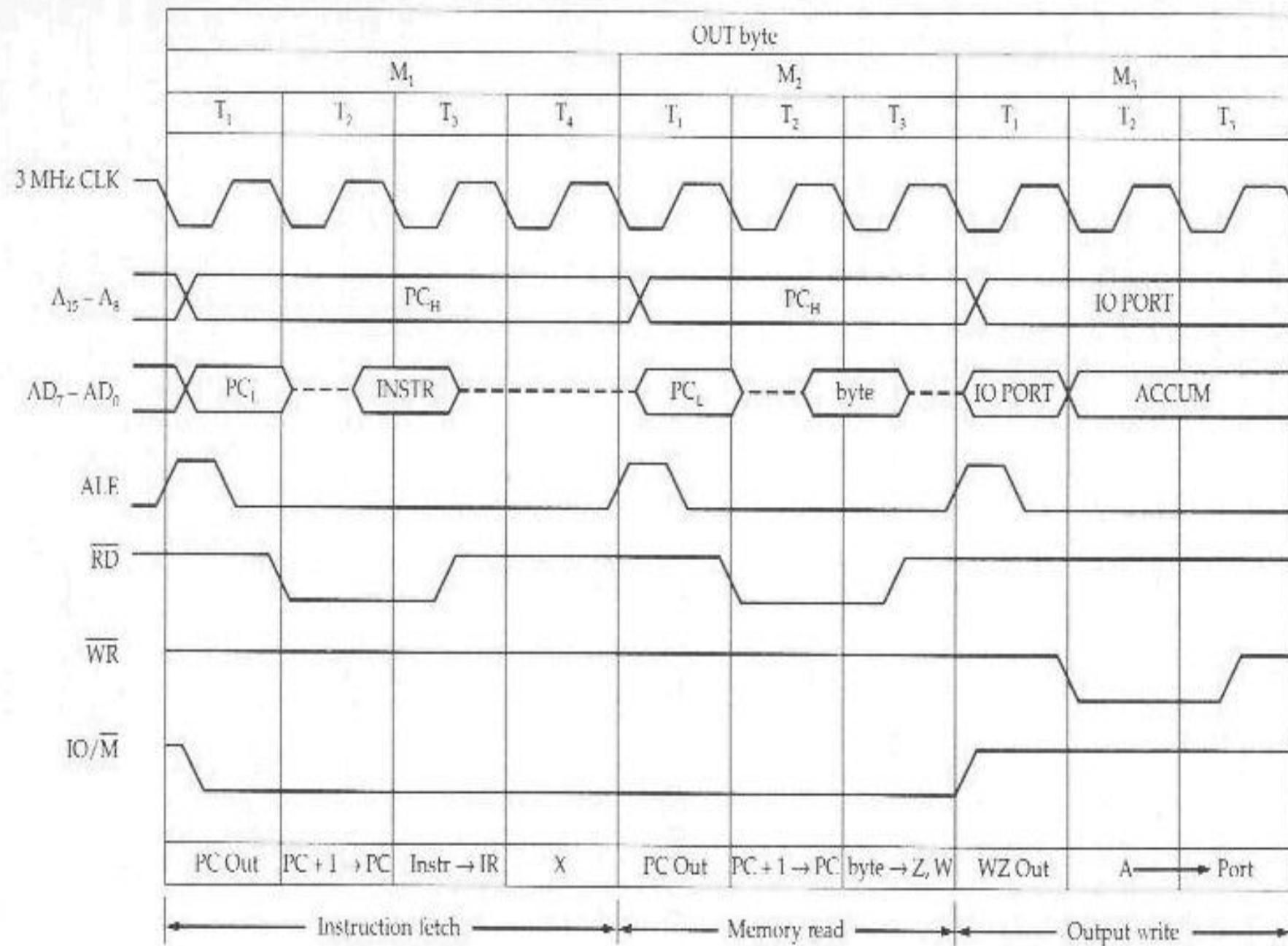


STA addr

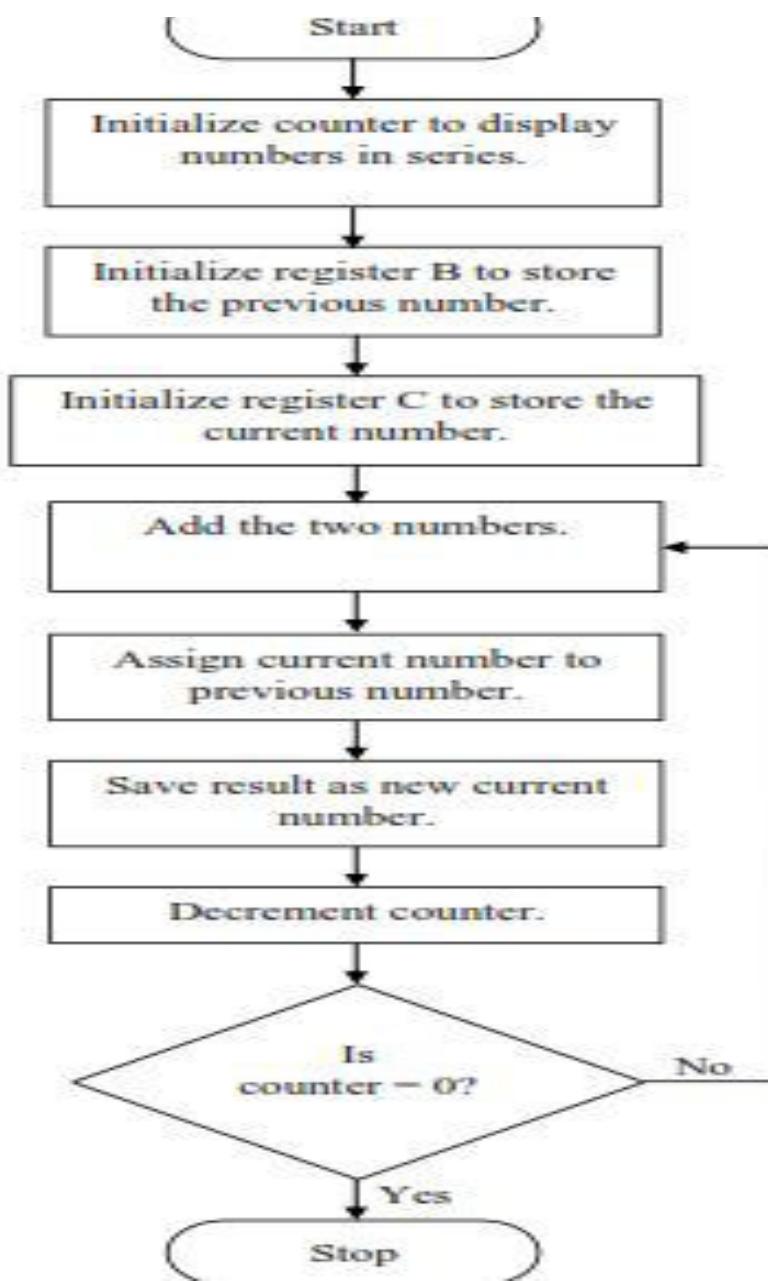


IN Byte





: Generation of Fibonacci series.



Address	Mnemonics	Operand	Opcode	Remarks
2000	MVI	D, 08H	16	Initialize counter to display numbers in series.
2001			08	Immediate value 07H.
2002	MVI	B, 00H	06	Initialize reg. B to store previous number.
2003			00	Immediate value 00H.
2004	MVI	C, 01H	0E	Initialize reg. C to store current number.
2005			01	Immediate value 01H.
2006	LXI	H, 3000H	21	Initialize H-L pair to point to memory.
2007			00	Lower-order of 3000H.
2008			30	Higher-order of 3000H.
2009	MOV	M, B	70	Move 00H from reg. B to memory.
200A	INX	H	23	Increment H-L pair.
200B	MOV	M, C	71	Move 01H from reg. C to memory.
200C	MOV	A, B	78	Move previous number from reg. B to reg. A.
200D	ADD	C	81	Add the two numbers.
200E	MOV	B, C	41	Assign current number to previous number.
200F	MOV	C, A	4F	Save result as new current number.
2010	INX	H	23	Increment H-L pair.
2011	MOV	M, A	77	Move number from reg. A to memory.
2012	DCR	D	15	Decrement counter.
2013	JNZ	200DH	C2	Jump to address 200DH if counter is not zero.
2014			0D	Lower-order of 200DH.
2015			20	Higher-order of 200DH.
2016	HLT		76	Halt.

Explanation:

- This program generates the Fibonacci series. The Fibonacci series is:
0 1 1 2 3 5 8 13 21 34
- In hexadecimal, it will be:
00 01 01 02 03 05 08 0D 15 22
- The first two numbers of the series are 0 and 1. The third number is computed as $0 + 1 = 1$, fourth number is $1 + 1 = 2$, fifth number is $1 + 2 = 3$ and so on.
- The count is initialized in register D to display the numbers in series.
- Initialize register B to first number 00H and register C to second number 01H.
- Initialize H-L pair to point to memory location 3000H.
- Move the first two numbers from registers B and C to memory locations 3000H and 3001H.
- Add the two numbers and store the result as first number.
- Increment H-L pair and move the result from accumulator to memory location.
- The next term is then computed by making the result equal to previous number.
- The process is repeated until all the numbers are calculated.

Output:**After Execution:**

3000H:	00H
3001H:	01H
3002H:	01H
3003H:	02H
3004H:	03H
3005H:	05H
3006H:	08H
3007H:	0DH
3008H:	15H
3009H:	22H

Reference Book:

- **Microprocessors and Microcomputer-Based System Design, Author: Mohamed Rafiquzzaman,**
- **Chapter-2**
 - **2.7-2.8.**
 - 2.5(only overview)