Unit 3: Instruction Cycle

Contents:

Instruction Cycle, Machine Cycle and T-states

❑Machine Cycle of 8085 Microprocessor: op-code fetch, memory read, memory write, I/O read, I/O write, interrupt

Fetch and Execute Operation, Timing Diagram

❑Timing Diagram of MOV, MVI, IN, OUT, LDA, STA

Memory Interfacing and Generation of Chip Select Signal

Key Terms

- **T-state**
  - It is the time period of a single cycle of the clock frequency.

- **Machine Cycle**
  - The number of T-states required performing a read or a write operation either from memory or I/O.
  - A machine cycle may consist of three to six T-states.

- **Instruction Cycle**
  - It is the total number of machine cycles required to execute a complete instruction.

Machine Cycles of 8085 µP

I. **Op-code fetch cycle**

- ❑ The MP uses this cycle to take the op-code of an instruction from the memory location to processor.
- ❑ The op-code is taken from memory and transferred to instruction register for decoding and execution.
- ❑ The time required to complete this cycle is 4 to 6 T-states.

II. **Memory read cycle**

- ❑ The MP executes these cycles to read data from memory.
- ❑ The address of memory location is given by instruction.
- ❑ The time required to complete the memory read cycle is 3 T-states.

III. **Memory write cycle**

- ❑ The MP executes these cycles to write data to memory.
- ❑ The address of memory is given by instructions.
- ❑ The time required to complete the memory write cycle is 3 T-states.

Machine Cycles of 8085 µP

IV. **I/O read cycle**

- ❑The MP executes these cycles to read data from I/O devices.
- ❑The address of I/O port is given by instruction.
- ❑The time required to complete the I/O read cycle is 3 T-states.

V. **I/O write cycle**

- ❑ The MP executes these cycles to write data into I/O devices.
- ❑ The address of I/O port is given by instruction.
- ❑ The time required to complete the I/O write cycle is 3 T-states.

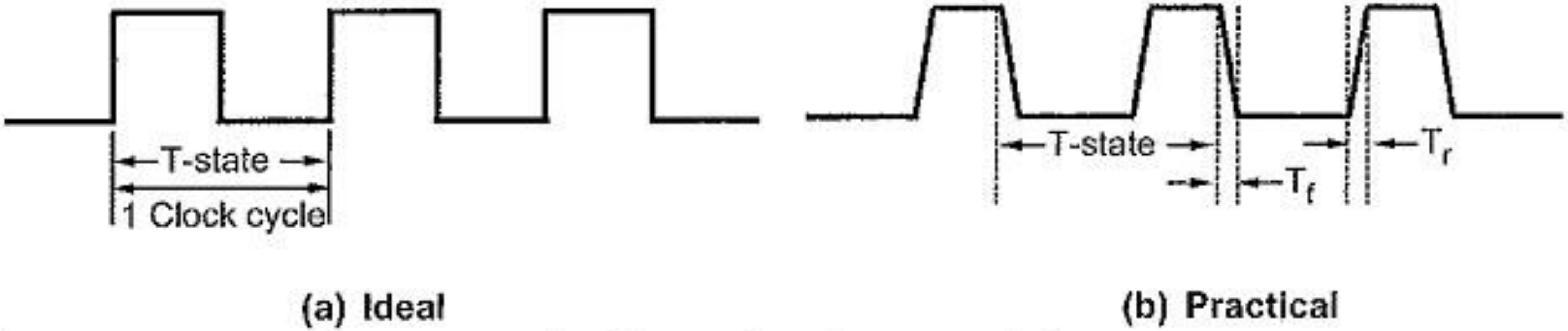
VI. **Interrupt acknowledge cycle**

- ❑ In the response to interrupt request input **INTR**, the MP executes these cycles to get information from the interrupting devices.
- ❑ The time required to complete this cycle is 3 T-states.

Clock Signal

- The 8085 divides the clock frequency provided at X1 and X2 inputs by 2, which is called **operating frequency**.
- All the operations within the Instruction Cycle of 8085 are synchronized with this operating frequency.
- Therefore in the timing diagram operating frequency clock is shown on the top and then the signals are shown with reference to operating frequency clock.
- Ideally, the clock signal should be square wave with zero rise time and fall time, as shown in the figure. But in practice, we don’t get zero rise time and fall time.
- Therefore the clock and other signals are always shown with finite rise and fall times. Fig. 1.5. shows the practical way of representing clock signal.

Clock Signal



(a) Ideal

(b) Practical

Fig. 1.5 Clock signal representation

Single Signal

- ❖Single signal is represented by a line. It may have status either **logic 0** or **logic 1** or **tri-state**. The change in the state of the signal takes finite time and hence the state change of signal is represented with finite rise time and fall time, as shown in the Fig. 1.6.

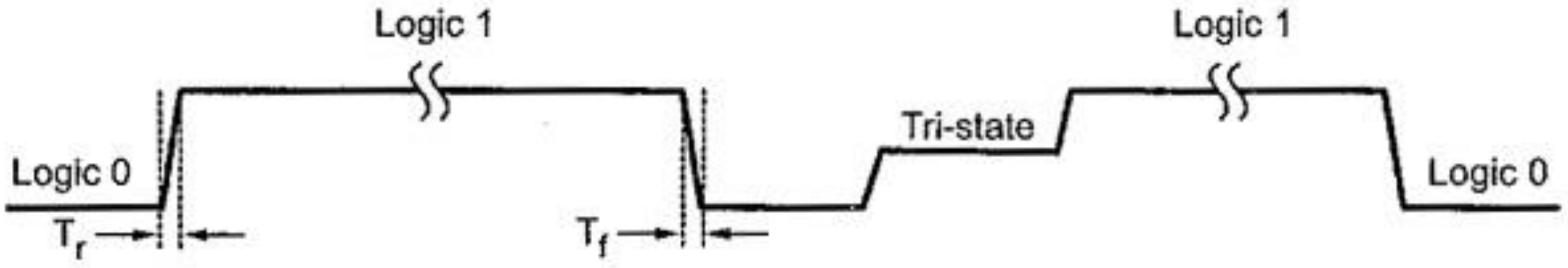


Fig. 1.6 Single signal representation

Group of Signals

- ❖**Group of signals** is also called **a bus** e.g. address bus and data bus. To avoid complications in the timing diagram these signal are grouped and shown in the form of block as shown in Fig. 1.7.

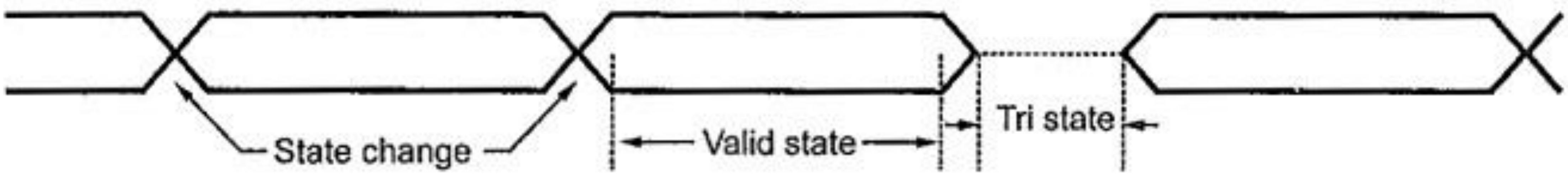


Fig. 1.7 Group of signals representation

Group of Signals

- ❖In the group representation individual state is not considered, but the group state is considered. Change in state of single signal changes the state of group.
- ❖It is represented by the cross as shown the Fig. 1.7.
- ❖The tri-state condition of the group signals is shown by dotted lines.
- ❖Two straight lines represent valid state/stable state.

Group of Signals

- In microprocessor systems, activation of signal/signals depends on the state of other signal/signals. Such situations are shown in the timing diagrams with the help of specific symbols.
- There are four possibilities :
  - **Activation of a signal with the change in state of other signal.**
  - **Activation of a signal with the Change in state of other signals.**
  - **Activation of signals with the change in state of other signal.**
  - **Activation of signals with the change in state of other signals.**

Group of Signals

❖Fig. 1.8 shows the representation of dependence of the signal/signals, in the timing diagram.

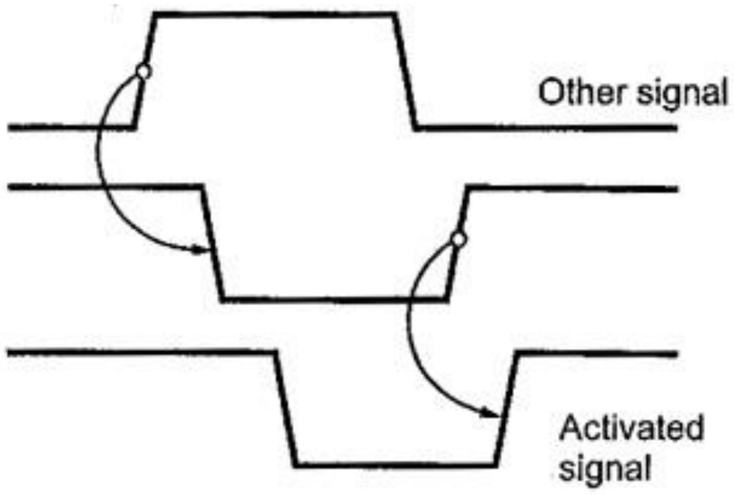


Fig. 1.8 (a) Activation of signal with the change in state of other signal

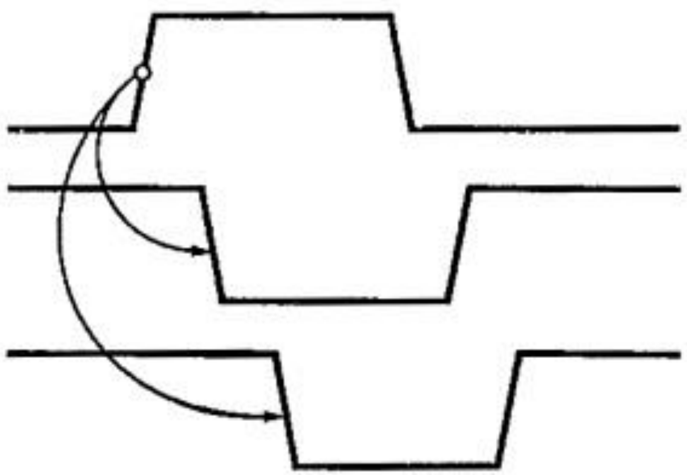


Fig. 1.8 (b) Activation of signal with the change in state of other signal

Signal Timings

- ❖In the Instruction Cycle of 8085 microprocessor, signals are activated at specific instant for specific time period.
- ❖Once we understand this, it is very easy to draw timing diagrams.
- ❖The following section explains when the signals are activated and for what period they remain in active state.

ALE (Address Latch Enable)

- This signal is active high signal.
- It is activated in the beginning of the **T1 state** of each machine cycle, except bus idle machine cycle, and it remains active in the T1 state as shown in the Fig. 1.9.

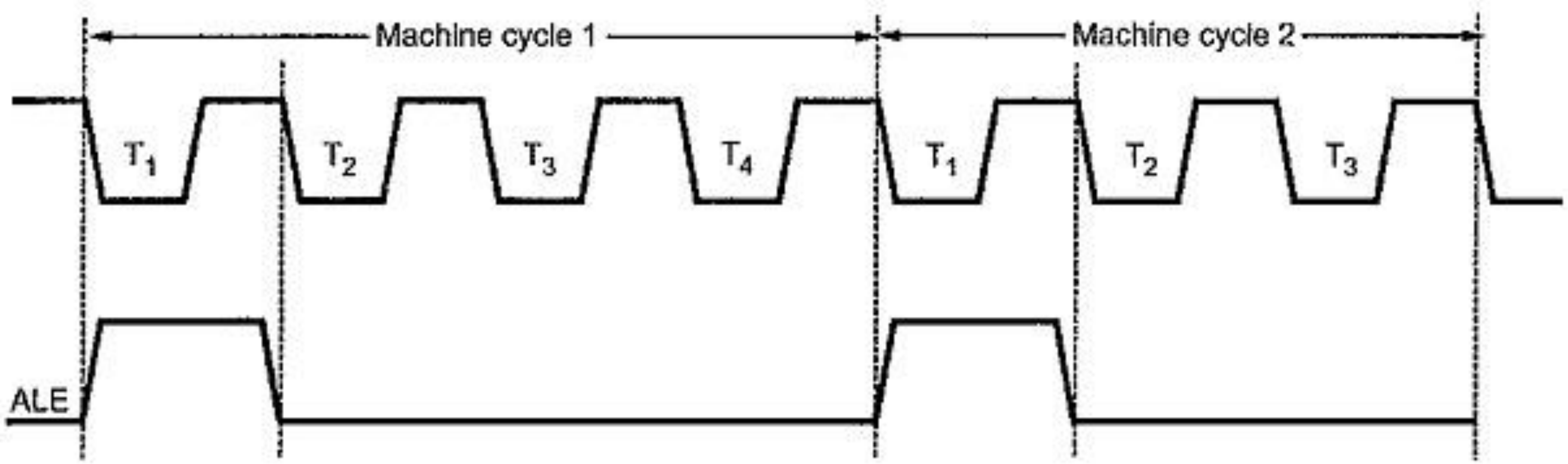


Fig. 1.9 ALE activation and its period

A_0-A7 (Lower byte address)

❖The lower byte of address is available on the multiplexed address/data bus (AD_0 – AD_7) during **T1 state** of each machine cycle, except bus idle machine cycle, as shown in Fig. 1.10.

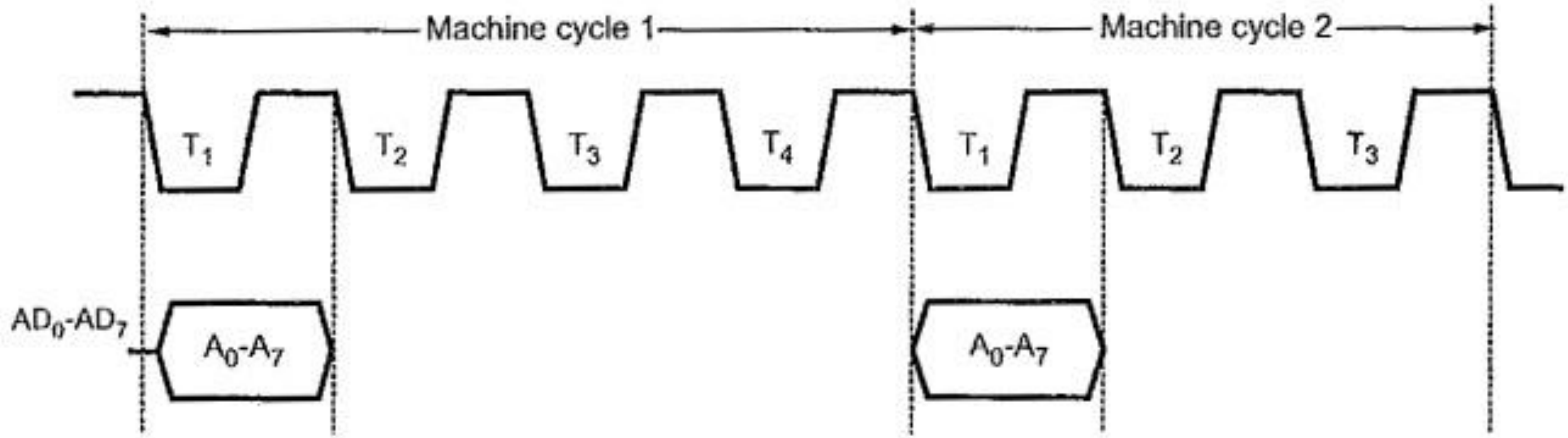


Fig. 1.10 Lower address on the multiplexed bus

D_0-D_7 (Data Bus)

- The data from memory or I/O device and from microprocessor to memory or I/O device is transferred during T2 and T3-states.
- It is important to note that in **read machine cycle**, data will appear on the data bus during the **later part of the T2-state**, as shown in the Fig. 1.11.
- Whereas in **write cycle** data will appear on the data bus at the **beginning of the T2-state**, as shown in the Fig. 1.11.

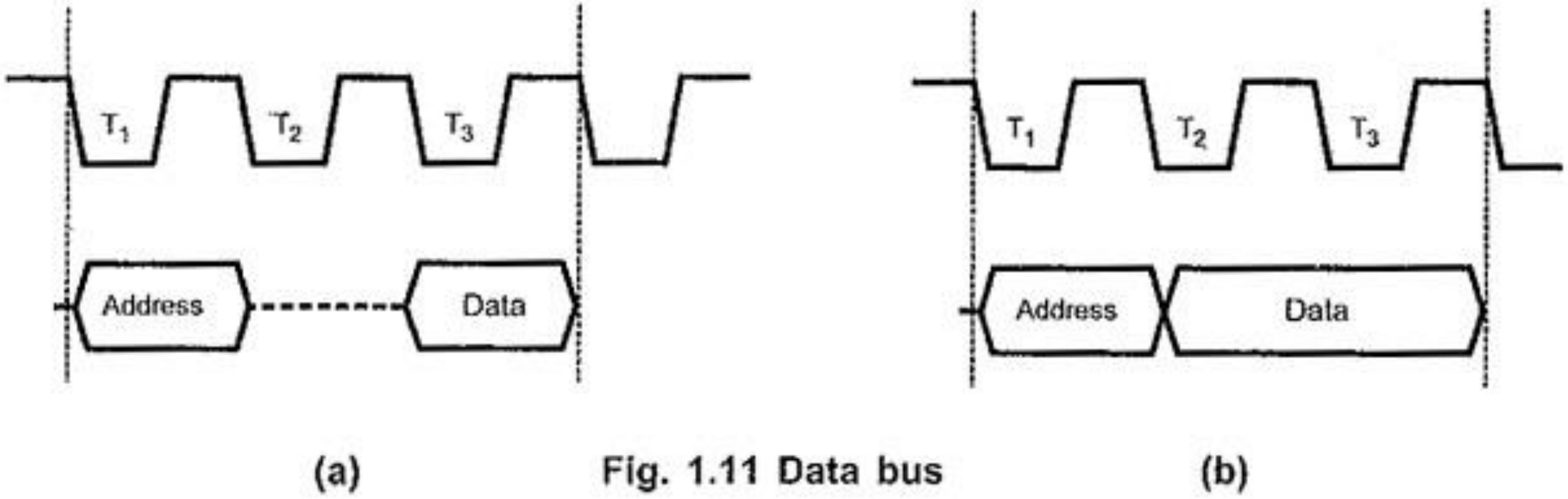


Fig. 1.11 Data bus

D_0-D_7 (Data Bus)

- ❖To **read** data from **memory** or **I/O** device it is necessary to select memory or I/O device.
- ❖After selection, device will put the data from, selected location on the data bus.
- ❖This action needs finite time. This time is referred to as '**access time**' .
- ❖In case, of write cycle, data is available in the registers of the microprocessor and it can put that data on the data bus with **zero access time**.

$A_8 – A_{15}$ (Higher byte address)

- The higher byte of address is available on the **$A_8 – A_{15}$** bus during **T1**, **T2** and **T3** - states of each machine cycle, except **bus idle** machine cycle, as shown in Fig. 1.12.

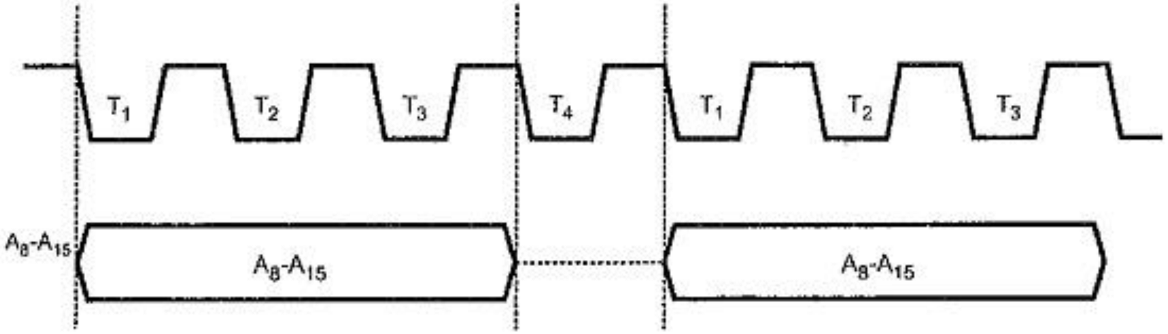


Fig. 1.12 Higher byte address on A_8–A_{15}

IO/$\overline{M}$, S_0, S_1 :

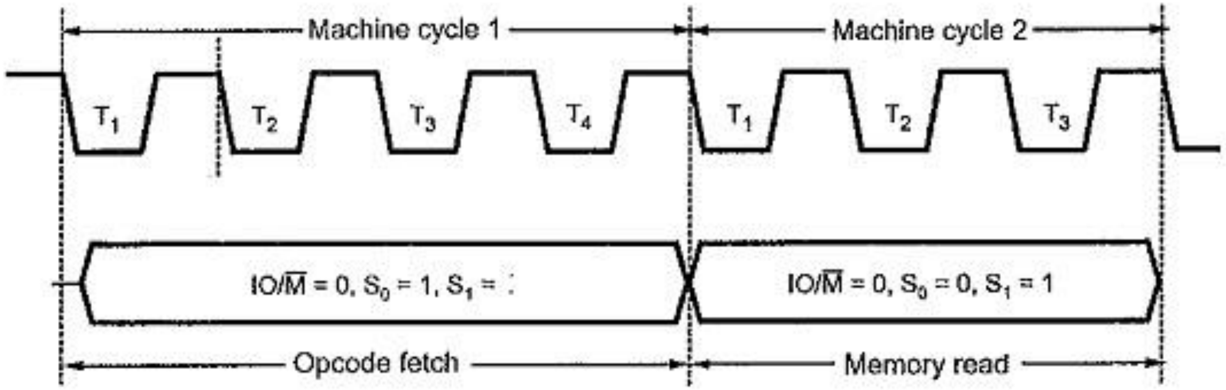

Machine cycle 1
Machine cycle 2
T1
T2
T3
T4
T1
T2
T3
IO/$\overline{M}$ = 0, S_0 = 1, S_1 = 1
IO/$\overline{M}$ = 0, S_0 = 0, S_1 = 1
Opcode fetch
Memory read

Fig. 1.13 Status signals

- These signals are called **status signals**.
- They decide **the type of machine cycle to be executed**.
- They are activated at the **beginning of T1-state** of each machine cycle and remain active till the **end of the machine cycle**.

$\overline{\mathrm{RD}}$ and $\overline{\mathrm{WR}}$

- These signals decide the direction of the data transfer.
- When **$\overline{\mathrm{RD}}$** signal is **active**, data is transmitted from memory or I/O device to the microprocessor.
- When **$\overline{\mathrm{WR}}$** signal is active, data is transmitted from microprocessor to the memory or I/O device.
- Both signals are never active at a time.
- As we know data transfer in Instruction Cycle of 8085 takes place during T2 and T3, these signals are activated during T2 and T3, as shown in the Fig. 1.14.

$\overline{RD}$ and $\overline{WR}$

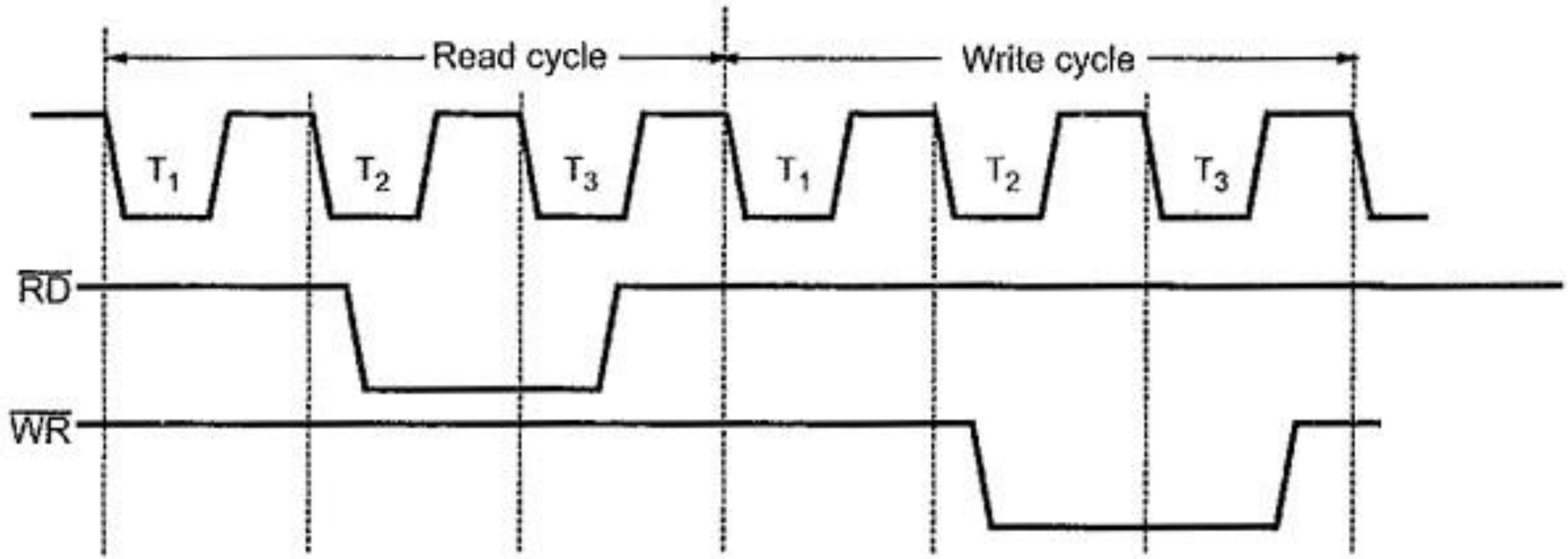


Fig. 1.14 $\overline{RD}$ and $\overline{WR}$ signals

RULES TO IDENTIFY NUMBER OF MACHINE CYCLES IN AN INSTRUCTION

- If an addressing mode is direct, immediate or implicit then No. of machine cycles = No. of bytes.
- If the addressing mode is indirect then No. of machine cycles = No. of bytes + 1. Add +1 to the No. of machine cycles if it is memory read/write operation.
- If the operand is 8-bit or 16-bit address then, No. of machine cycles = No. of bytes +1.
- These rules are applicable to 80% of the instructions of 8085.

Instruction Cycle, Machine Cycle and T-states

- ❖During normal operation, the microprocessor sequentially **fetches**, **decodes** and **executes one instruction** after **another** until a halt instruction (HLT) is executed.
- ❖The fetching, decoding and execution of a single instruction constitutes an instruction cycle, which consists of one to five read or write operations between processor and memory or input/output devices.
- ❖Each memory or I/O operation requires a particular time period, called **machine cycle**.
- ❖In other words, to move byte of data in or out of the microprocessor, a machine cycle is required.
- ❖Each machine cycle consists of 3 to 6 clock periods/cycles, **referred to as T-states**.

Instruction Cycle, Machine Cycle and T-states

❖Therefore we can say that, one Instruction Cycle of **8085** consists of **one to five machine cycles** and one machine cycle consists of **three to six T-states** i.e. **three to six** clock periods, as shown in the Fig. 1.4.

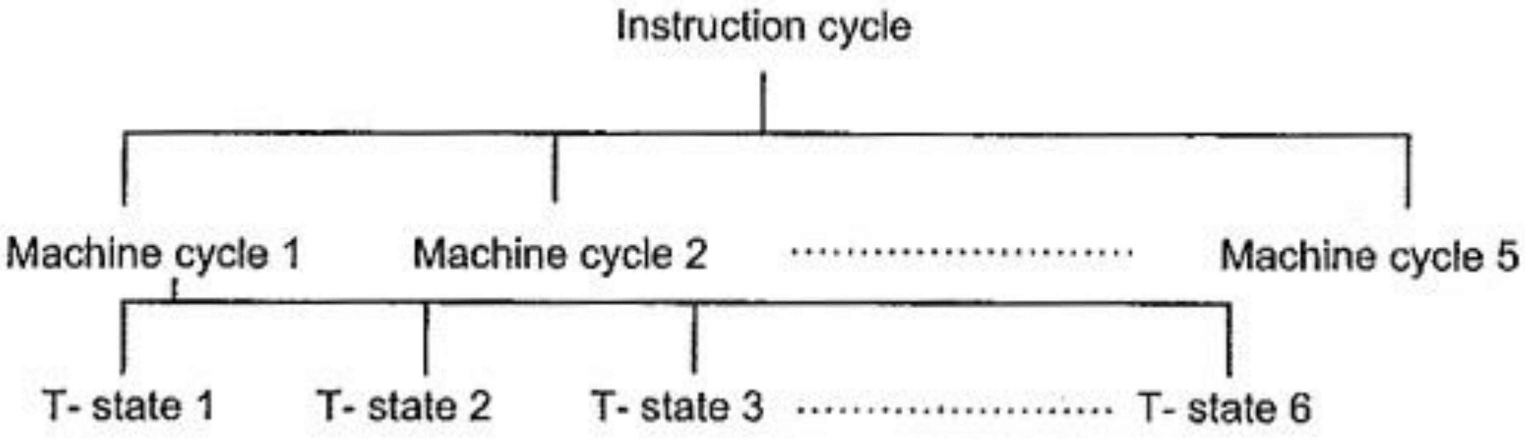
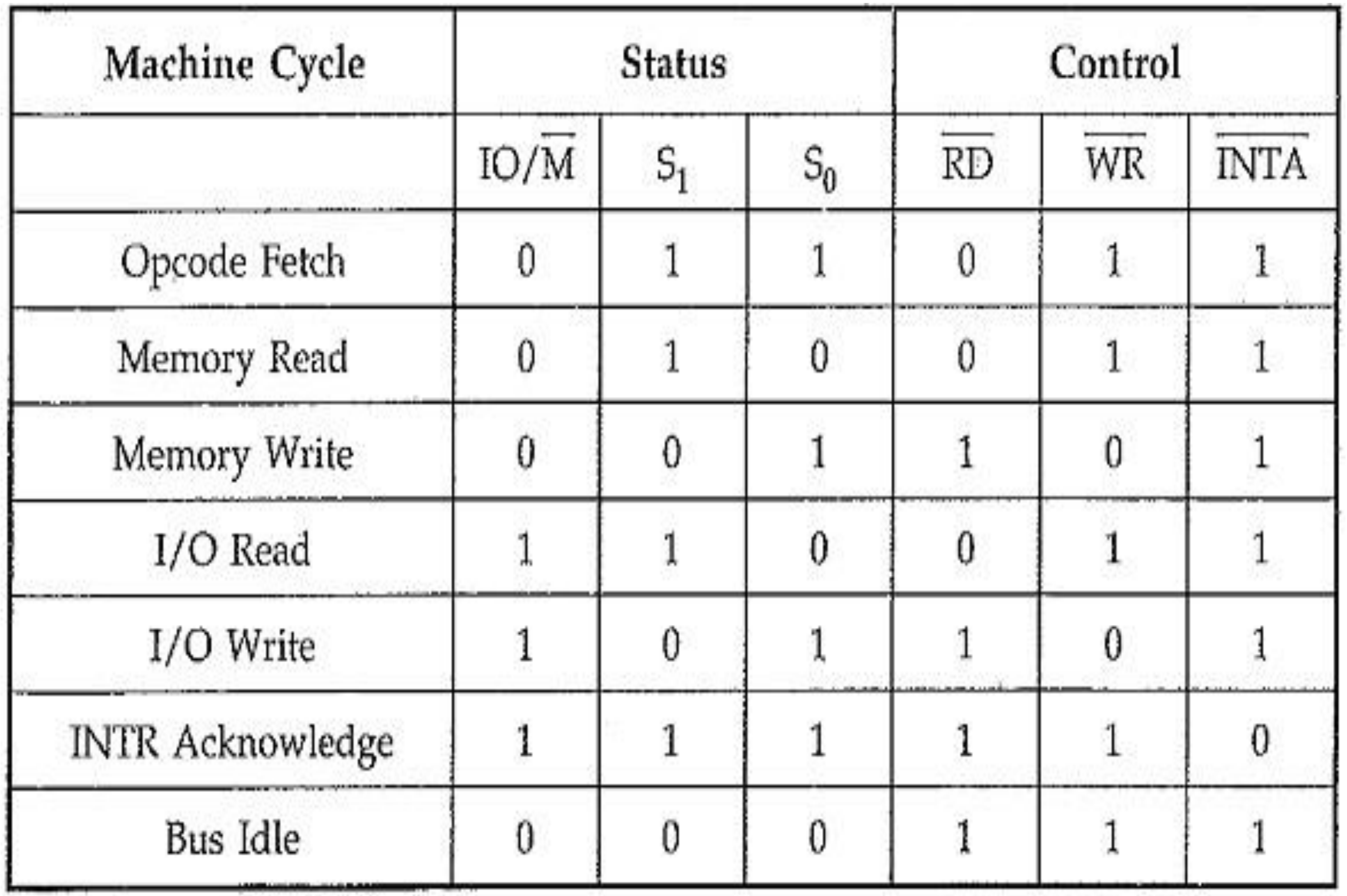


Fig. 1.4 Relation between instruction cycle, machine cycle and T-state

Instruction Cycle, Machine Cycle and T-states

- ❖There are **seven** different types of machine cycles in the 8085A.
- ❖Three status signals **IO/$\overline{\mathbf{M}}$**, **S1** and **S0** identify each type as shown in Table 1.1.
- ❖These signals are generated **at the beginning** of each machine cycle and remained valid for the duration, of the cycle.



Machine Cycle	Status			Control		
	IO/$\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
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/O Read	1	1	0	0	1	1
I/O Write	1	0	1	1	0	1
INTR Acknowledge	1	1	1	1	1	0
Bus Idle	0	0	0	1	1	1

Table 1.1 8085 machine cycles

Instruction Cycle, Machine Cycle and T-states

❖The seven Machine Cycle in 8085 Microprocessor are :

1. **Opcode Fetch Cycle**
2. **Memory Read**
3. **Memory Write**
4. **I/O Read**
5. **I/O Write**
6. **Interrupt Acknowledge**
7. **Bus Idle**

1. Opcode Fetch Cycle

- ❖The first Machine Cycle of 8085 Microprocessor of every instruction is opcode fetch cycle in which the 8085 finds the nature of the instruction to be executed.
- ❖ In this Machine Cycle in 8085, processor places the contents of the **Program Counter** on the address lines, and through the read process, reads the opcode of the instruction.
- ❖Fig. 1.15 (a) shows flow of data (opcode) from memory to the microprocessor and Fig. 1.15 (b) shows the timing diagram for Opcode Fetch Machine Cycle 8085.
- ❖The length of this cycle is not fixed. It varies from 4T states to 6T states as per the instruction.

1. Opcode Fetch Cycle

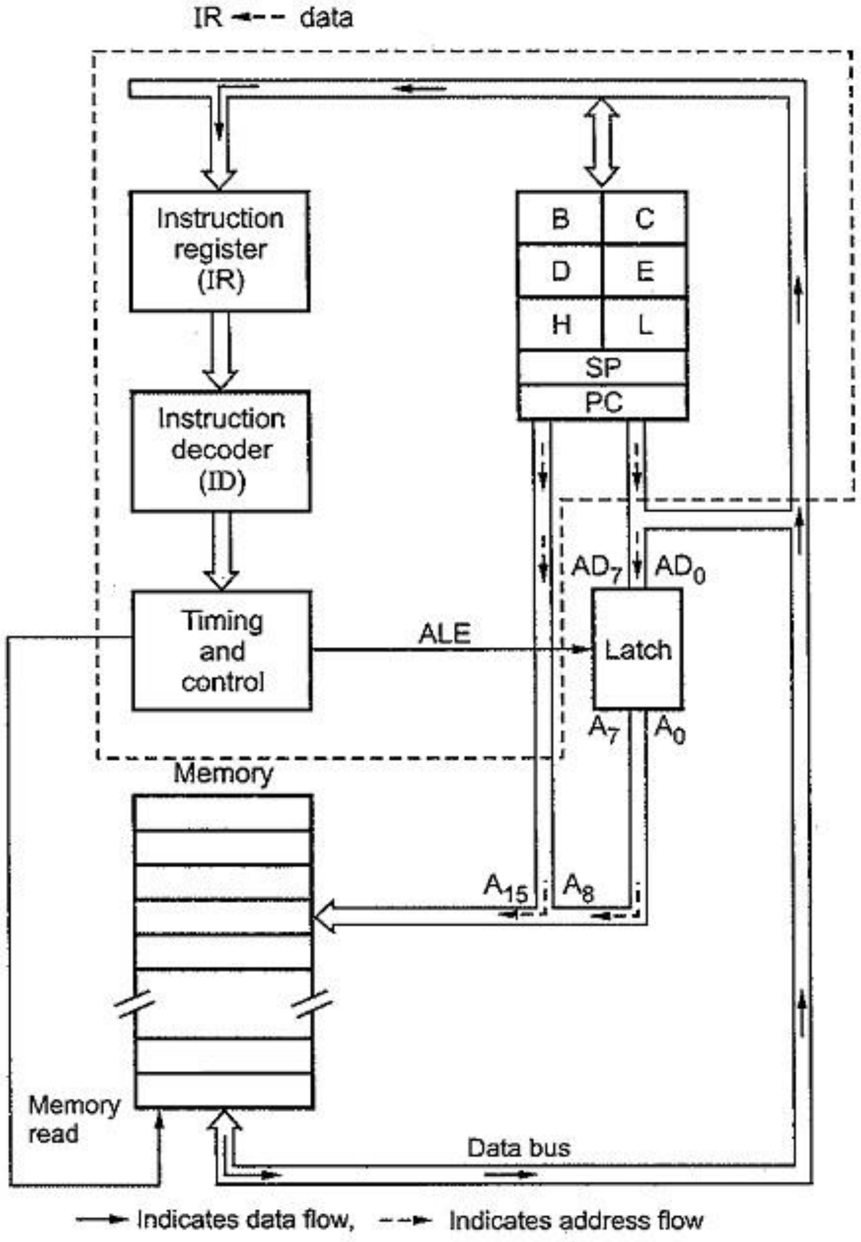


Fig. 1.15 (a) Data (opcode) flow from memory to micrprocessor

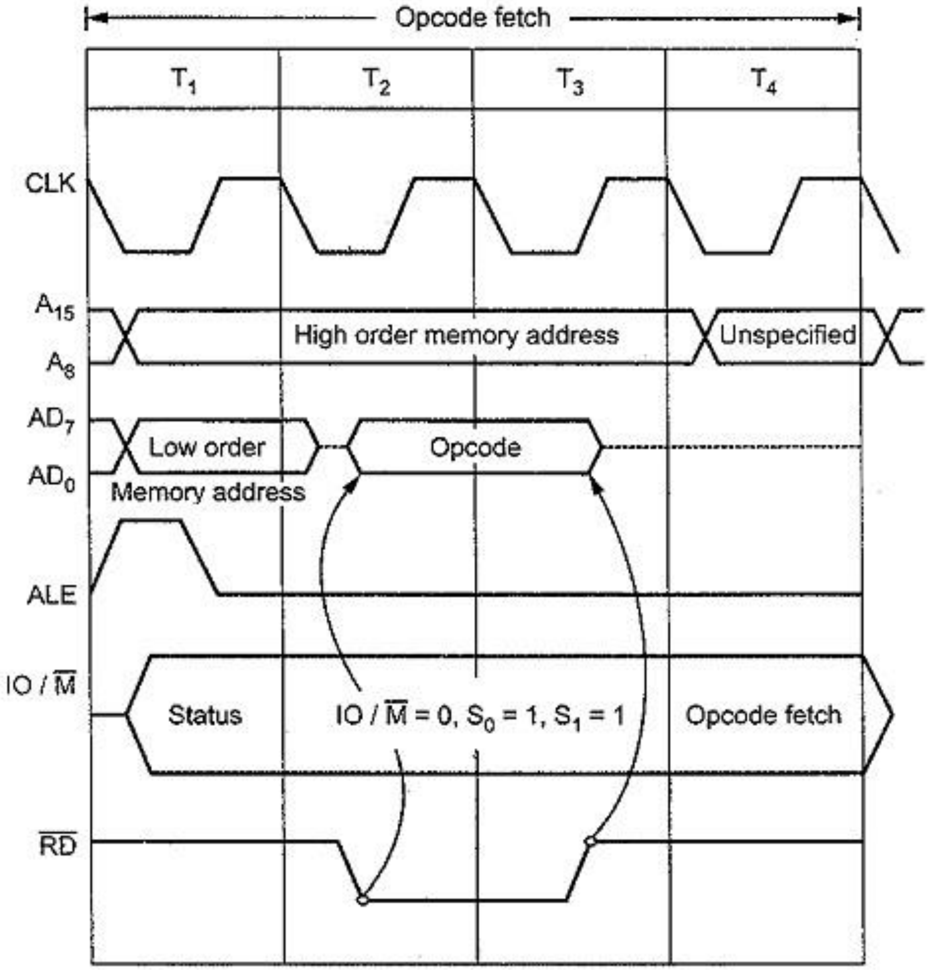


Fig. 1.15 (b) Opcode fetch machine cycle

1. Opcode Fetch Cycle

- The following section describes the opcode fetch cycle in step by step manner.

- **Step 1 : (State T_1):** In T_1 state, the 8085 places the contents of program counter on the address bus. The high-order byte of the PC is placed on the $A_8 – A_{15}$ lines. The low-order byte of the PC is placed on the $AD_0 – AD_7$ lines which stays on only during T_1. Thus microprocessor activates ALE (Address Latch Enable) which is used to latch the low-order byte of the address in external latch before it disappears.
  - In T_1, 8085 also sends status signals IO/M, S_1, and S_0. IO/M specifies whether it is a memory or I/O operation, S_1 status specifies whether it is read/write operation; S_1 and S_0 together indicates read, write, opcode fetch, machine cycle operation, or whether it is in HALT state. In opcode fetch machine cycle status signals are : IO/M = 0, S_1 = 1, S_0 = 1.

- **Step 2 : (State T_2)** In T_2, low-order address disappears from the $AD_0 – AD_7$ lines. (However $A_0 – A_7$ remain available as they were latched during T_1). In T_2, 8085 sends RD signal low to enable the addressed memory location. The memory device then places the contents of addressed memory location on the data bus ($AD_0 – AD_7$).

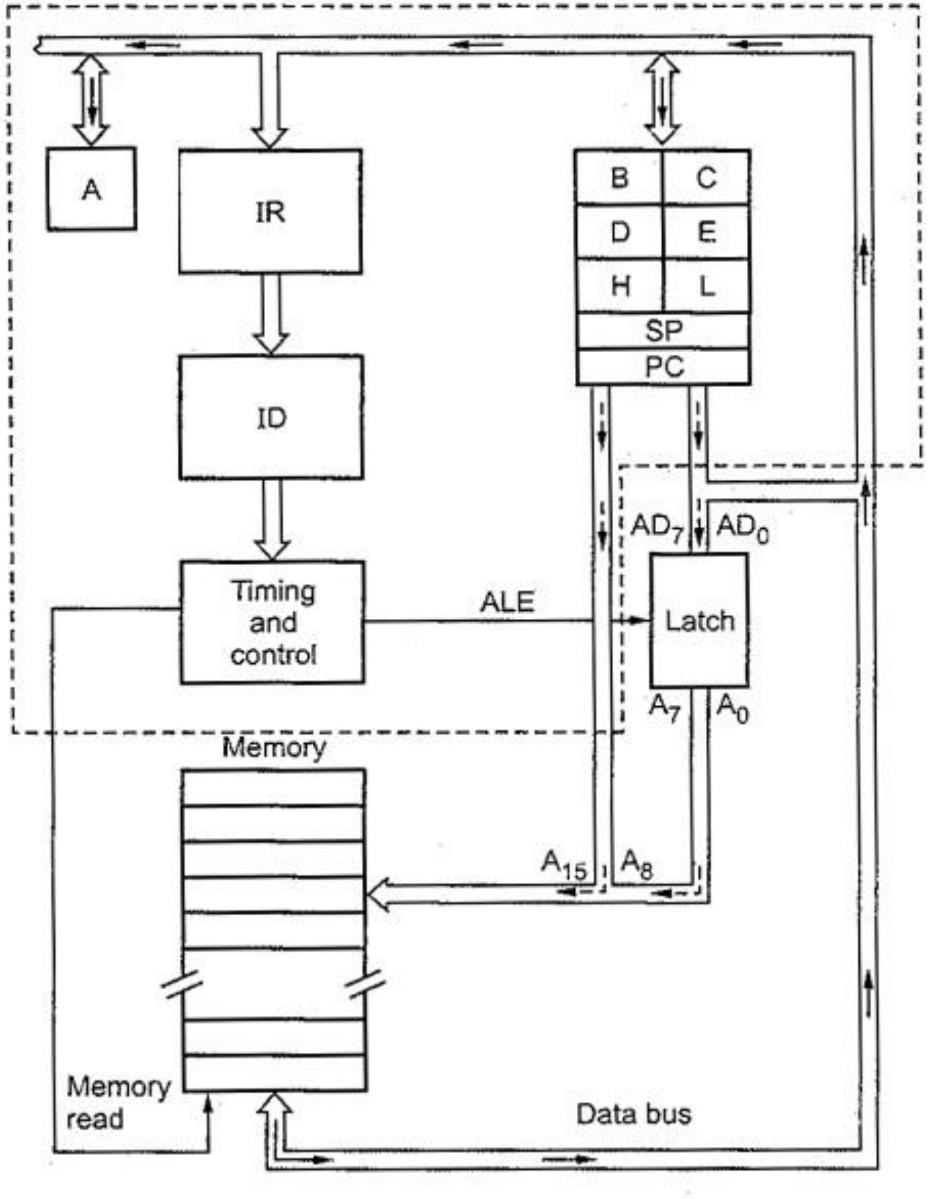
1. Opcode Fetch Cycle

- ❖**Step 3 : (State T_3)** During T_3, 8085 loads the data from the data bus in its Instruction Register and raises RD to high which disables the memory device.
- ❖**Step 4 : (State T_4)** In T_4, microprocessor decodes the opcode, and on the basis of the instruction received, it decides whether to enter state T_5 or to enter state T_1 of the next Machine Cycle of 8085 Microprocessor. One byte instructions those operate on eight bit data (8 bit operand) are executed in T_4.
  - ❑For example : MOV A, B, ANA D, ADD B, INR L, DCR C, RAL and many more.
  - ❑**Note** : For one byte instructions which operate on eight bit data, data is always available in the internal memory of 8085 i.e. registers.
- ❖**Step 5 : (State T_5 and T_6)** State T_5 and T_6, when entered, are used for internal microprocessor operations required by the instruction. During T_5 and T_6 , 8085 performs stack write, internal 16 bit; and conditional return operations depending upon the type of instruction. One byte instructions those operate on sixteen bit data (16 bit operand) are executed in T_5 and T_6. For example DCX H, PCHL, SPHL, INX H, etc.

2. Memory Read Cycle

- ❖The 8085 executes the memory read cycle to read the contents of R/W memory or ROM.
- ❖The length of this machine cycle is 3-T states ($T_1 – T_3$).
- ❖In this Machine Cycle in 8085, processor places the address on the address lines from the stack pointer, general purpose register pair or program counter, and through the read process, reads the data from the addressed memory location.
- ❖Fig. 1.16 (a) shows flow of data from memory to the microprocessor and Fig. 1.16 (b) shows the timing diagram for memory read machine cycle.
- ❖Memory read machine cycle is similar to the opcode fetch machine cycle. However, they use only states T1 to T3, and the status signal values ($IO/\overline{M}$ = 0, S_1 = 1, S_0 = 0) appropriate for memory read machine cycle are issued in T1.

2. Memory Read Cycle



→ Indicates data flow, --→ Indicates address flow

Fig. 1.16 (a) Data flow from memory to microprocessor

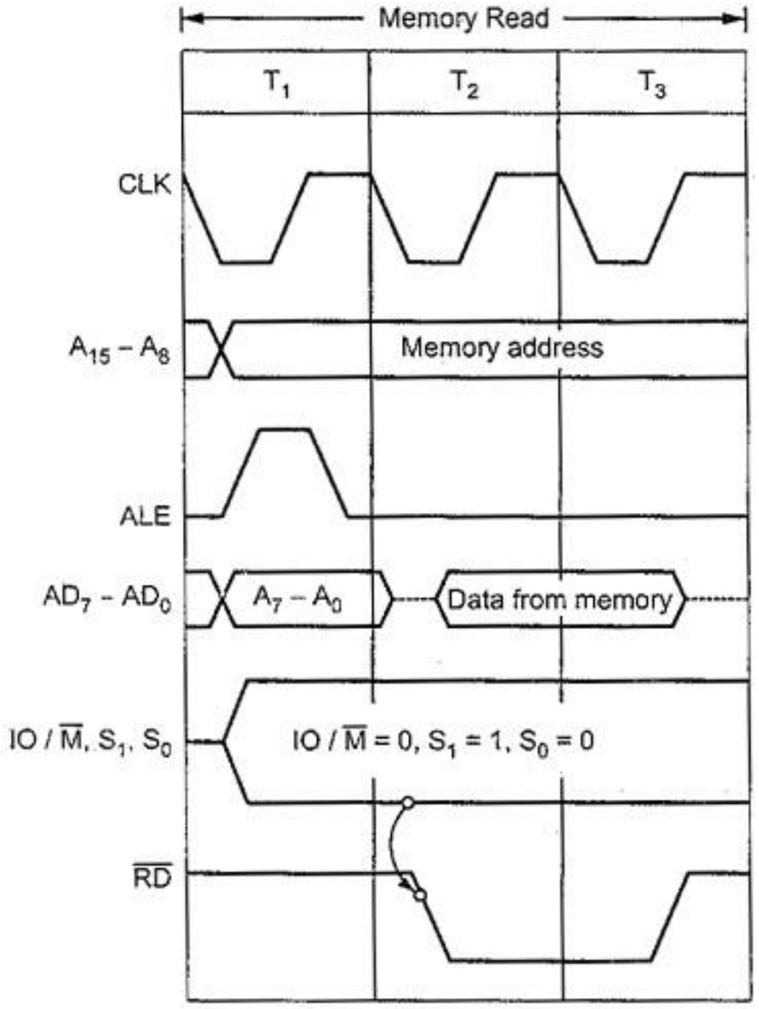


Fig. 1.16 (b) Memory read machine cycle

2. Memory Read Cycle

- ❖The following section describes the memory read machine cycle in step by step manner.
- ❖**Step 1 : (State T_1)** In T_1 state, microprocessor places the address on the address lines from stack pointer, general purpose register pair or program counter and activates ALE signal in order to latch low-order byte of address.
  - ❑During T_1, 8085 sends status signals : IO/M = 0, S_1 = 1, and S_0 = 0 for memory read machine cycle.
- ❖**Step 2 : (State T_2)** In T_2, 8085 sends RD signal low to enable the addressed memory location. The memory device then places the contents of addressed memory location on the data bus (AD_0 -AD_7).
- ❖**Step 3 : (State T_3)** During T_3, 8085 loads the data from the data bus into specified register (F, A, B, C, D, E, H, and L) and raises RD to high which disables the memory device.

3. Memory Write Cycle

❖The 8085 executes the memory write cycle to store the data into data memory or stack memory.

❖The length of this machine cycle is 3T states. ($T_1 - T_3$).

❖In this Machine Cycle of 8085 Microprocessor, processor places the address on the address lines from the stack pointer or general purpose register pair and through the write process, stores the data into the addressed memory location.

❖Fig. 1.17 shows the timing diagram for memory write machine cycle.

❖The memory write timing diagram is similar to the memory read timing diagram, except that instead of $\overline{RD}$, WR signal goes low during T2 and T3. The status signals for memory write cycle are : $IO/\overline{M} = 0$, $S_1 = 0$, $S_0 = 1$.

❖The following section describes the memory write machine cycle in step by step manner.

3. Memory Write Cycle

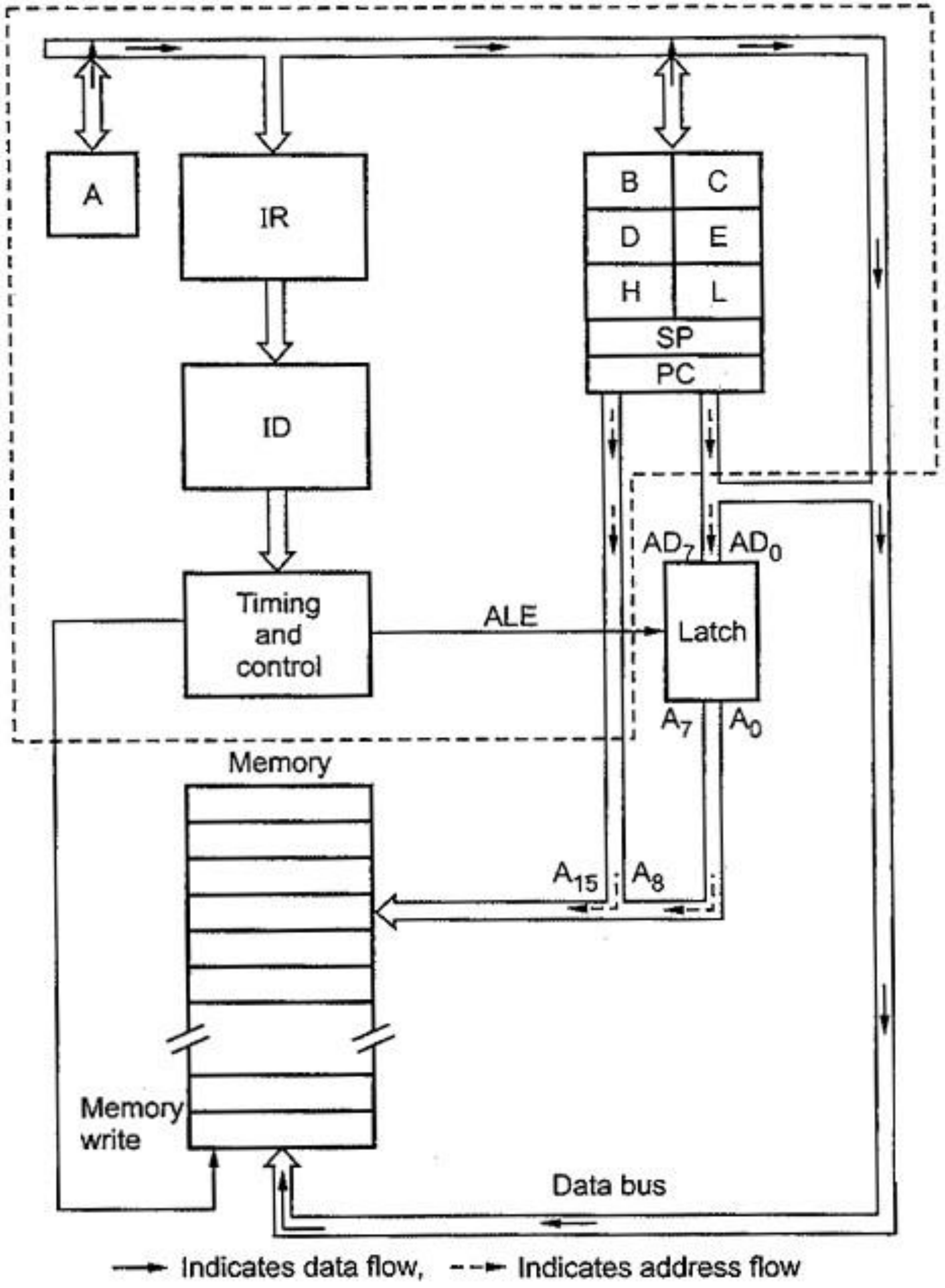


Fig. 1.17 (a) Data flow microprocessor to memory

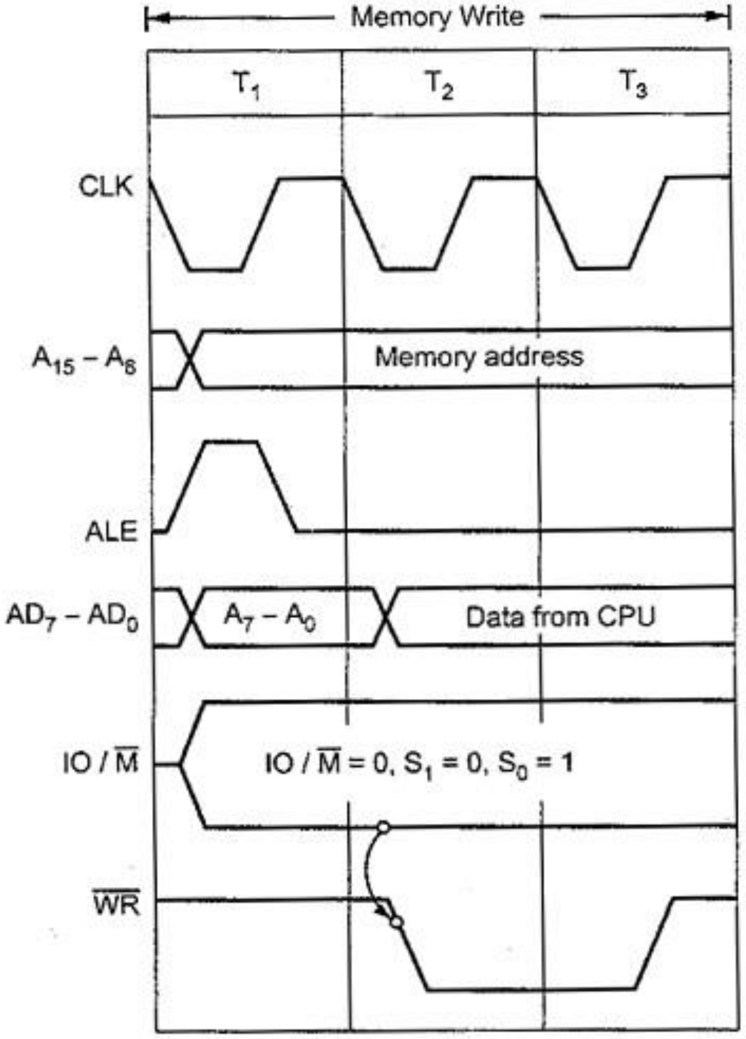


Fig. 1.17 (b) Memory write machine cycle

3. Memory Write Cycle

- **Step 1 : (State T_1)** In T_1 state, the 8085 places the address on the address lines from stack pointer or general purpose register pair and activates ALE signal in order to latch low-order byte of address. During T_1, 8085 sends status signals :
  - ❑ IO/M = 0, S_1 = 0 and S_0 = 1 for memory write machine cycle.
- **Step 2 : (State T_2)** In T_2, 8085 places data on the data bus and sends WR signal low for writing into the addressed memory location.
- **Step 3 : (State T_3)** During T_3, WR signal goes high, which disables the memory device and terminates the write operation.

4, 5. I/O Read and I/O Write cycles:

- The I/O read and I/O write machine cycles are similar to the memory read and memory write machine cycles, respectively, except that the IO/$\overline{\mathrm{M}}$ signal is high for I/O read and I/O write machine cycles.
- High IO/$\overline{\mathrm{M}}$ signal indicates that it is an I/O operation.
- Fig. 1.18 (b) and Fig. 1.19 (b) show the timing diagrams for I/O read and I/O write cycles, respectively.

4, 5. I/O Read and I/O Write cycles:

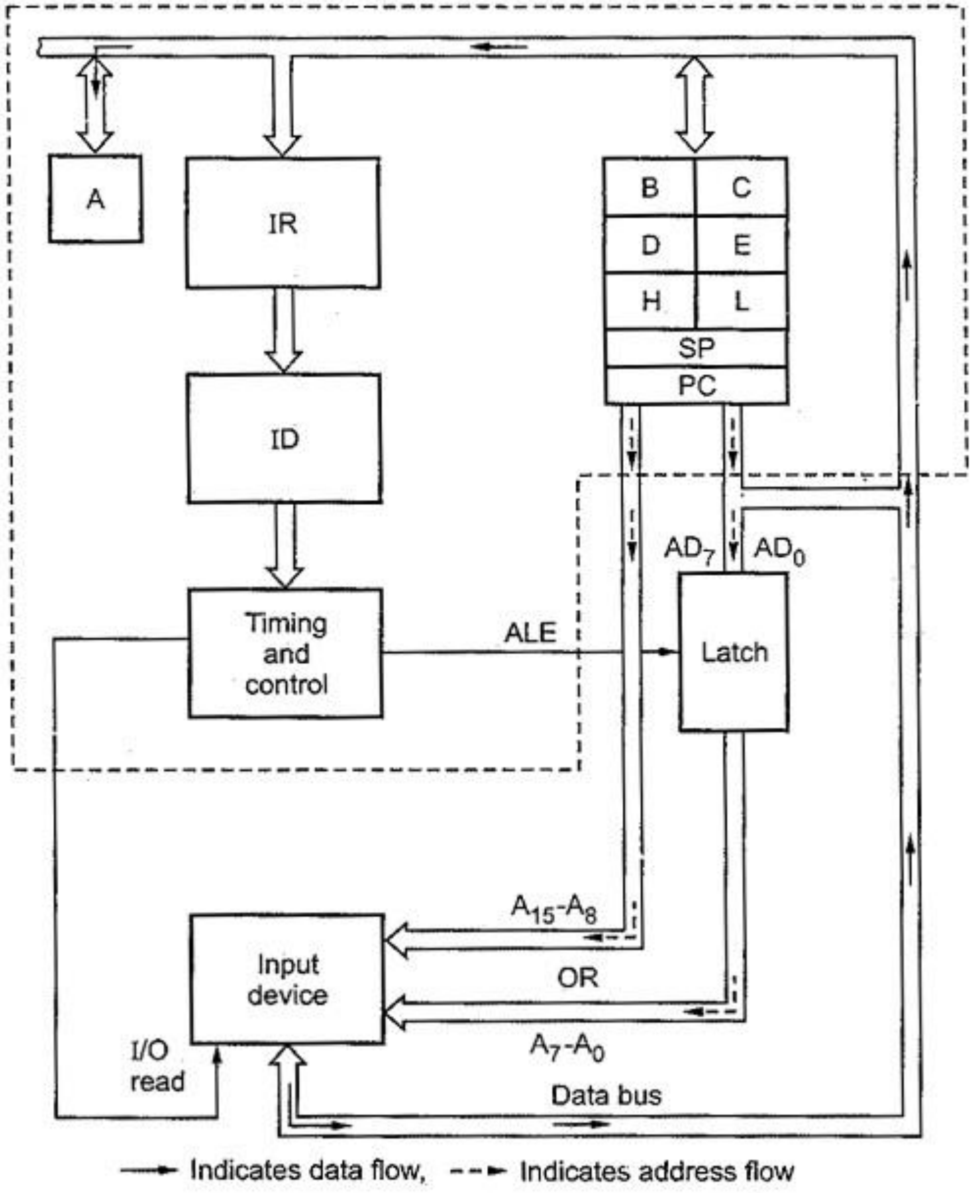


Fig. 1.18 (a) Data flow from input device to microprocessor

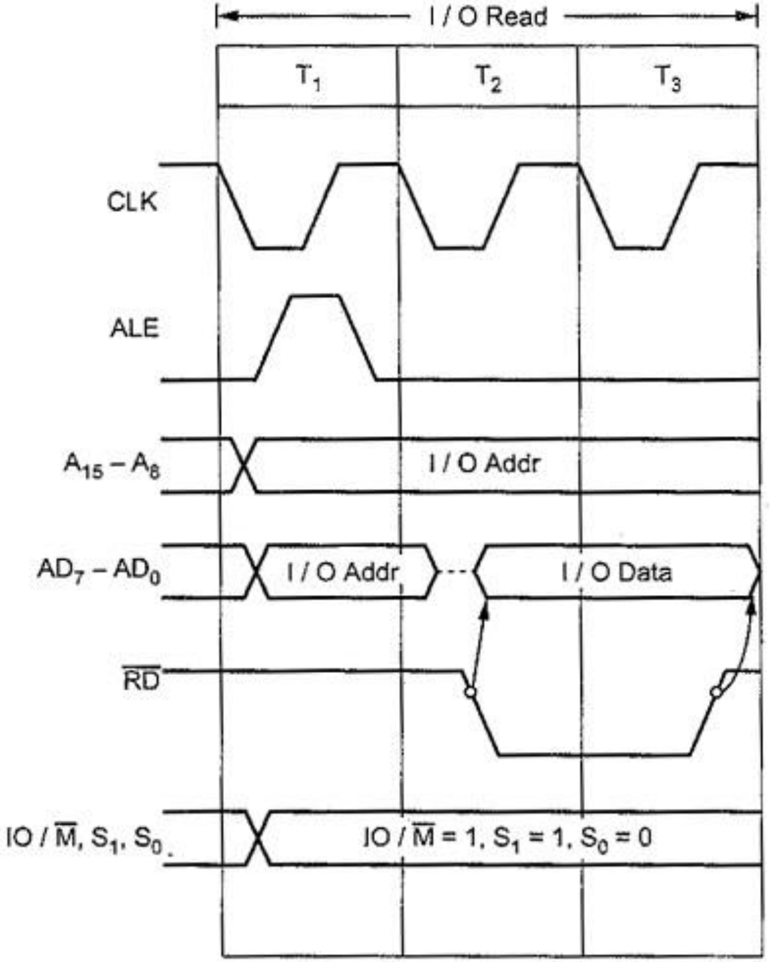


Fig. 1.18 (b) I/O read memory cycle

6. Interrupt Acknowledge Cycle

- In response to INTR signal, 8085 executes interrupt acknowledge machine cycle to read an instruction from the external device.
- Theoretically, the external device can place any instruction on the data bus in response to $\overline{\text{INTA}}$.
- However, only RST and CALL, save the PC contents (return address) before transferring control to the interrupt service routine.
- The next sections explain Interrupt Acknowledge Cycle of 8085 for RST and CALL instructions.

6. Interrupt Acknowledge Cycle

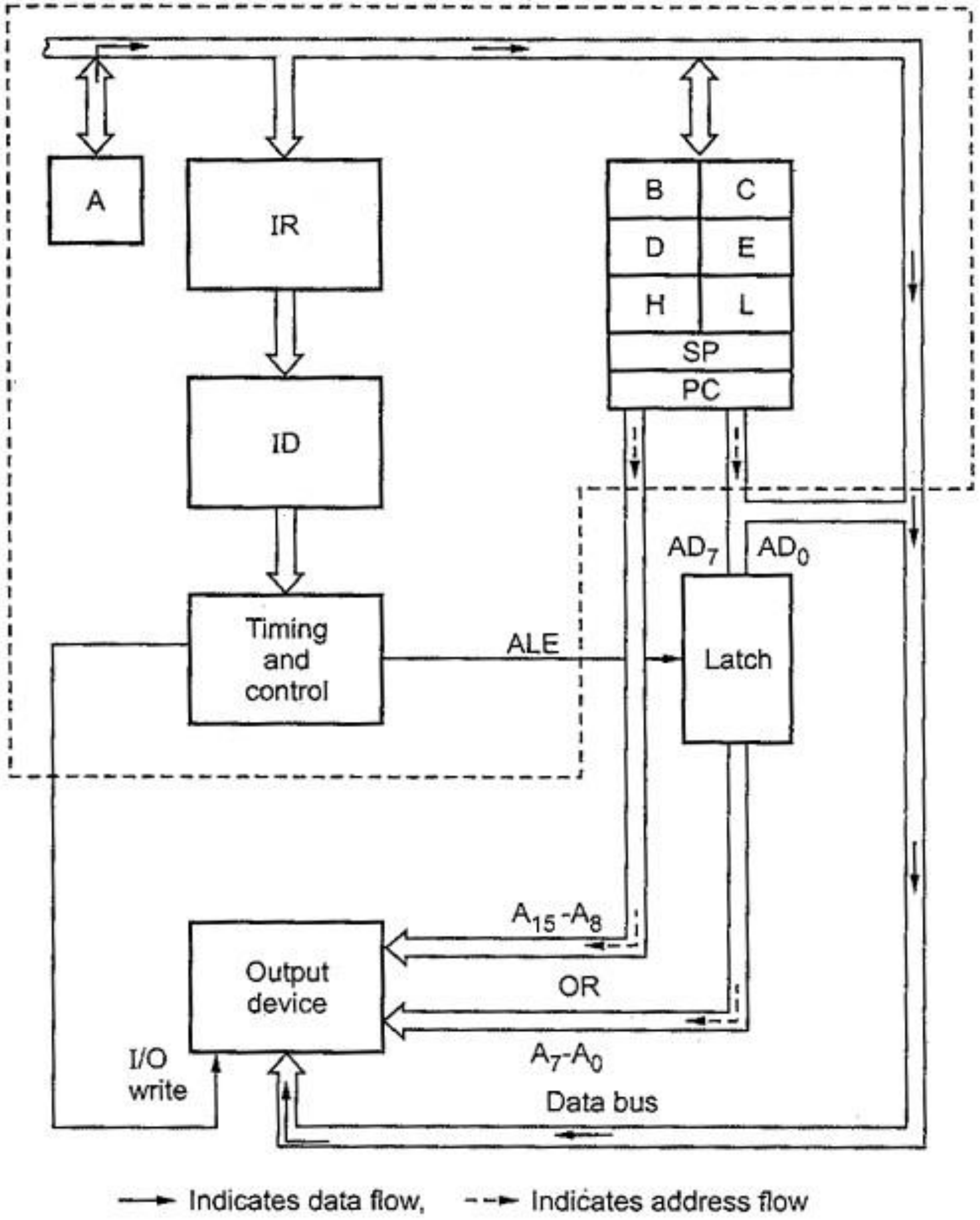


Fig. 1.19 (a) Data flow from microprocessor to output device

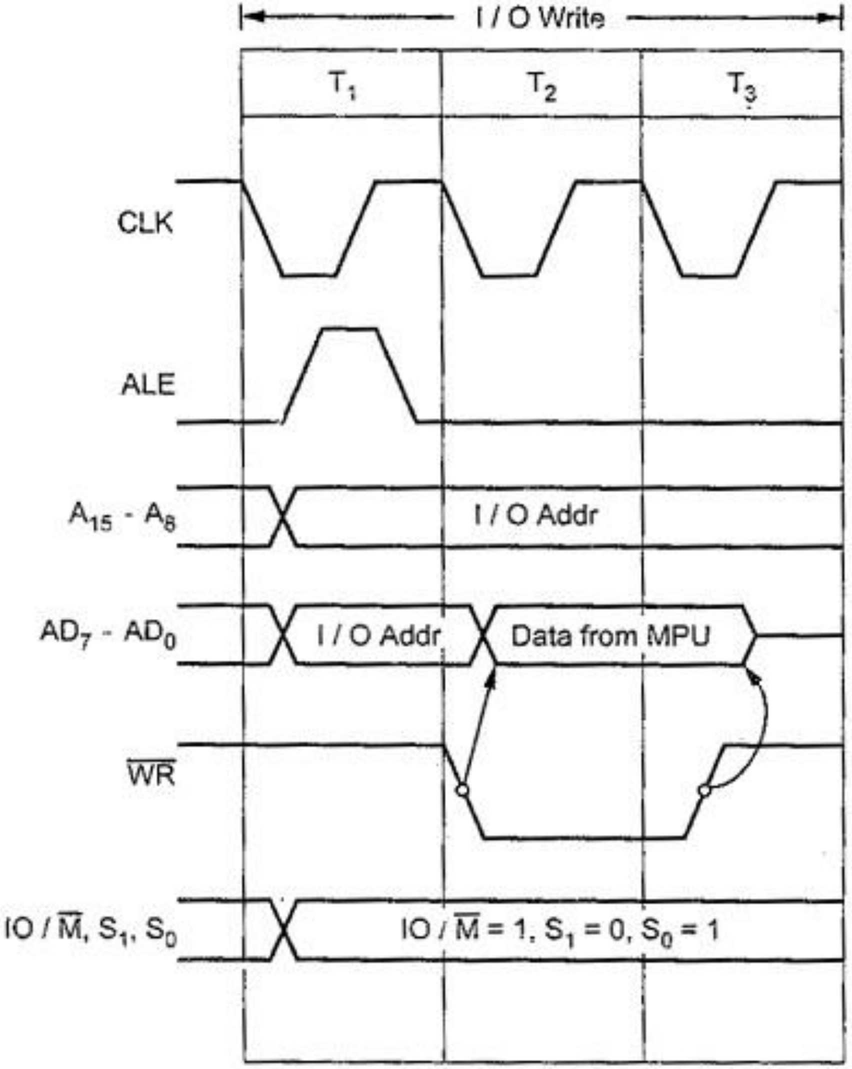


Fig. 1.19 (b) I/O write machine cycle

Interrupt Acknowledge Cycle for RST instruction

- ❖Fig. 1.20 shows the timing diagram of the interrupt acknowledge machine cycle and execution of RST instruction.

- ❖The interrupt acknowledge cycle is similar to the opcode fetch cycle, with two exceptions.
  1. **The $\overline{\text{INTA}}$ signal is activated instead of the $\overline{\text{RD}}$ signal.**
  2. **The status lines (IO/$\overline{\text{M}}$, S_0 and S_1) are 111 instead of 011.**

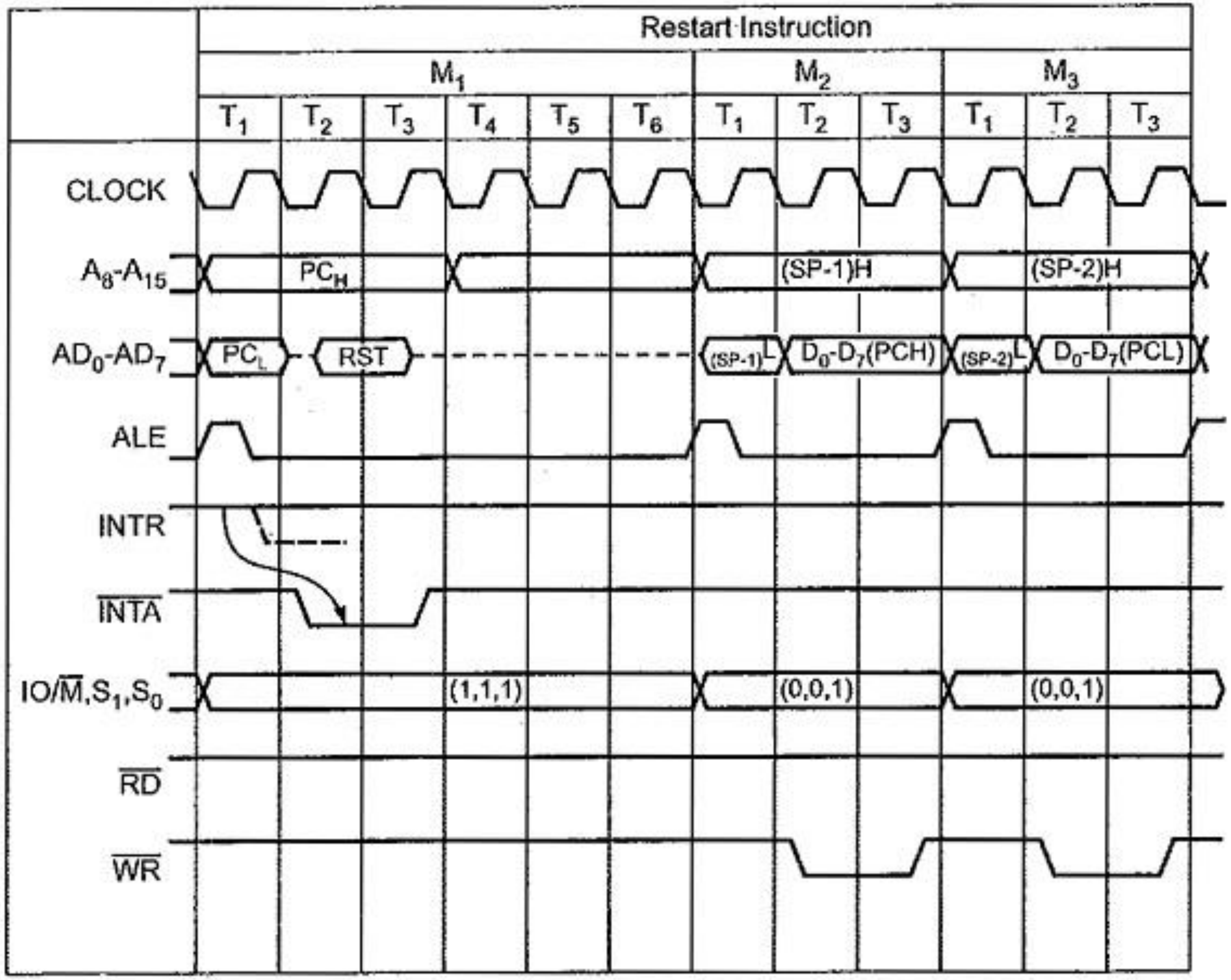


Fig. 1.20 Restart Instruction

Interrupt Acknowledge Cycle for RST instruction

- ❖ During interrupt acknowledge machine cycle (M_1), the RST is decoded, which initiates 1 byte CALL instruction to the specific vector location.
- ❖ The machine cycles M_2 and M_3 are memory write cycles that store the contents of the program counter on the stack, and then a new instruction cycle begins.

Interrupt Acknowledge Cycle for CALL instruction

- ❖Fig. 1.21 shows the timing diagram of the Interrupt Acknowledge Cycle of 8085 and execution of a CALL instruction.
- ❖For CALL instruction, it is necessary to fetch the two bytes of the CALL address through two additional interrupt acknowledge machine cycles (M_2 and M_3 in the 1.21).
- ❖The machine cycles M_4 and M_5 are memory write cycles that store the contents of the program counter on the stack, and then a new instruction cycle begins.

Interrupt Acknowledge Cycle for CALL instruction

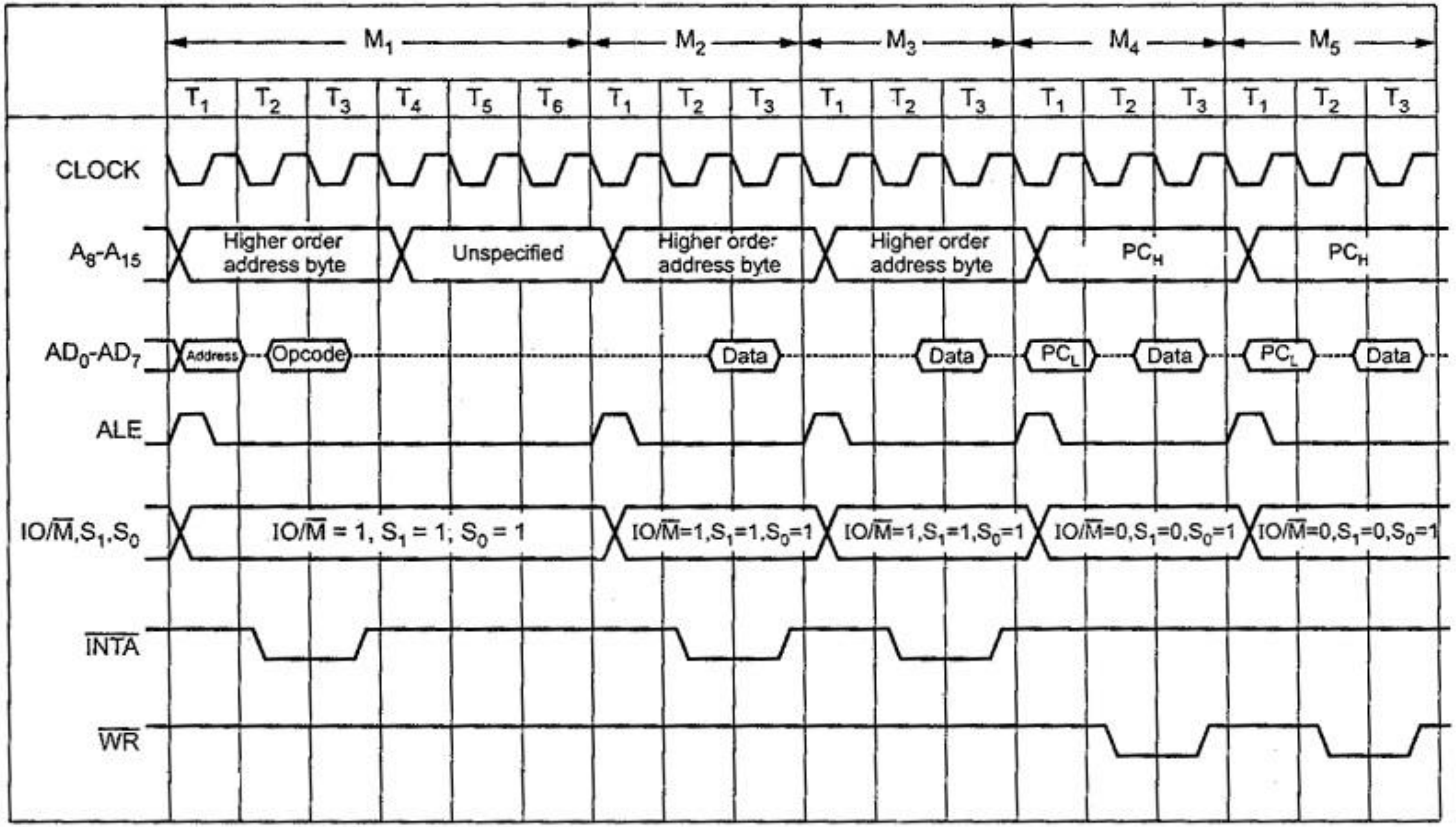


Fig. 1.21 Timing diagram of $\overline{\text{INTA}}$ machine cycle and execution of call instruction

7. Bus Idle Cycle

- ❖ There are few situations where the machine cycles are neither Read nor Write. These situations are:

1. For execution of DAD instruction (this instruction adds the contents of a specified register pair to the contents of HL register pair) ten T states are required. This means that after execution of opcode fetch machine cycle, DAD instruction requires 6 extra T-states to add 16 bit contents of a specified register pair to the contents of HL register pair. These extra T-states which are divided into two machine cycles do not involve any memory or I/O operation. These Machine Cycle in 8085 are called BUS IDLE machine cycles. Fig. 1.22 shows Bus Idle Machine Cycle of 8085 Microprocessor for DAD instruction.

7. Bus Idle Cycle

- ❖ In the case of DAD, these Bus Idle cycles are similar to memory read cycles, except RD and ALE signals are not activated.

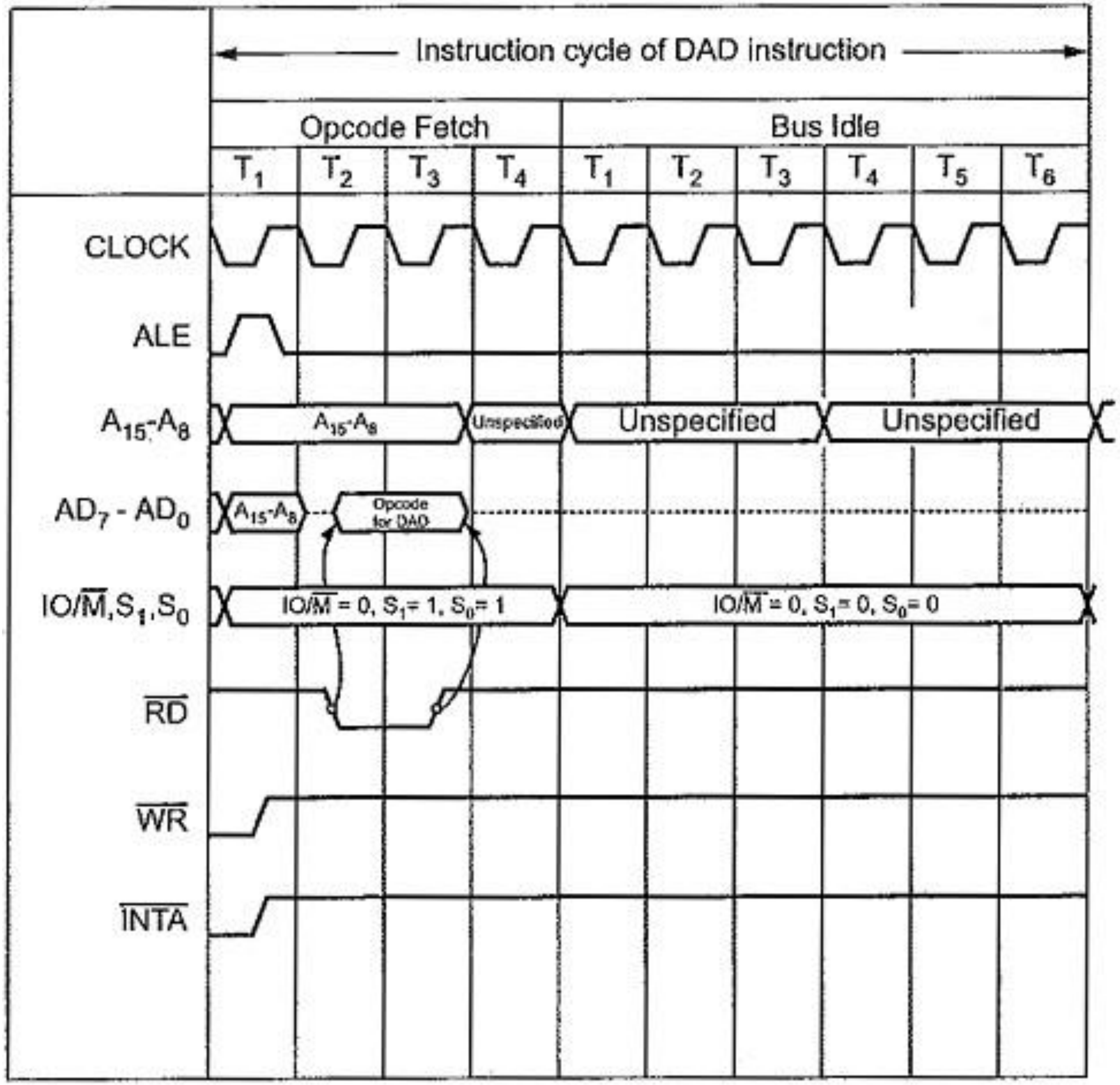


Fig. 1.22 Timing diagram for DAD instruction

7. Bus Idle Cycle

2. During internal opcode generations, for TRAP and RST interrupts, 8085 executes Bus Idle Machine Cycles. Fig. 1.23 shows the Bus Idle Machine Cycle of 8085 Microprocessor for TRAP. In response to TRAP interrupt, 8085 enters into a Bus Idle Machine Cycle during which it invokes restart instruction, stores the contents of PC onto the stack and places 0024H (Vector address of TRAP) onto the program counter.

7. Bus Idle Cycle

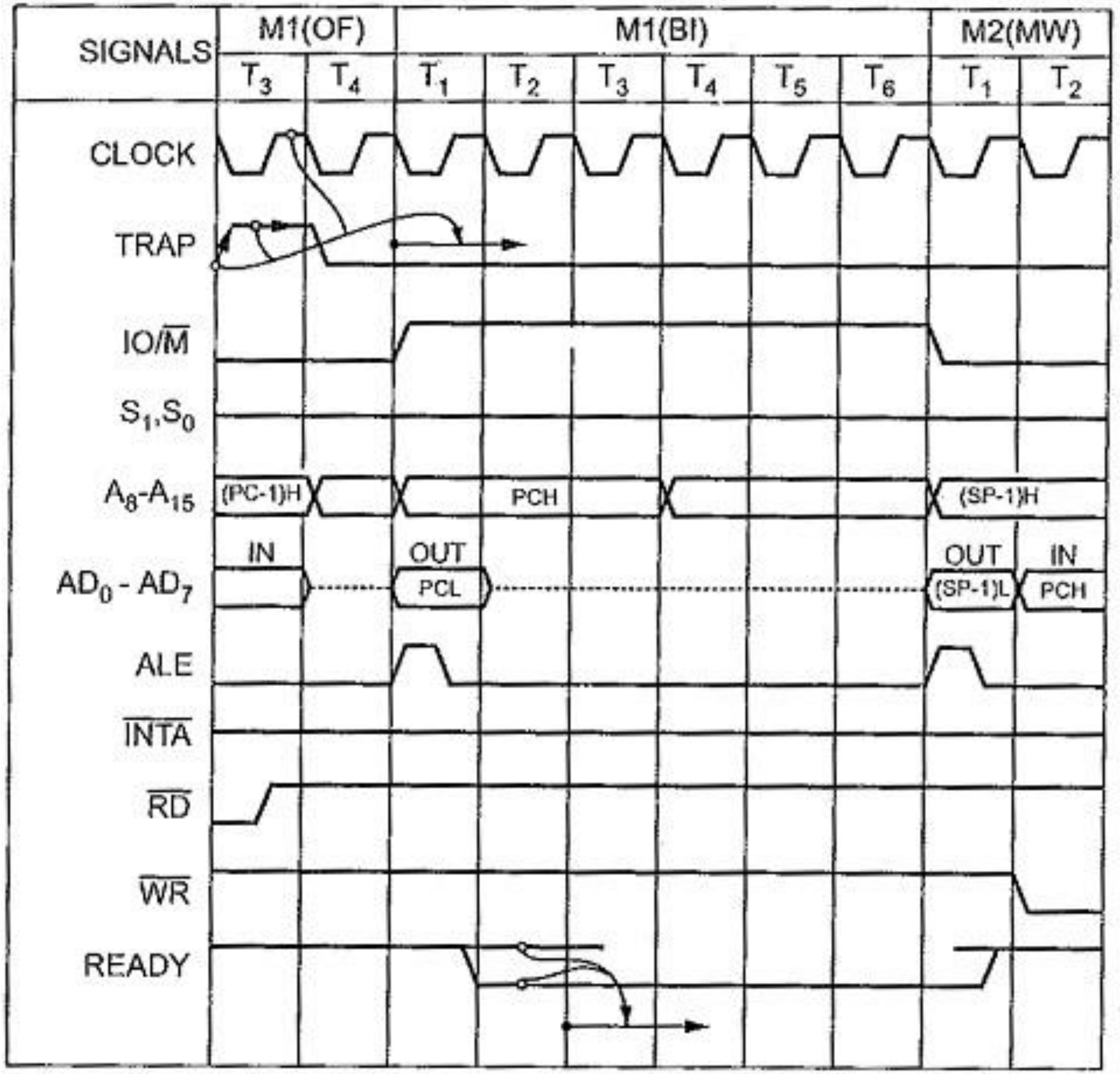


Fig. 1.23 Bus idle machine cycle for trap

7. Bus Idle Cycle

- ❖ The number of machine cycles required to fetch complete instruction depends on the instruction type :
  1. **One byte**
  2. **Two byte or**
  3. **Three byte**
- ❖ One byte instruction doesn’t require any additional machine cycle. Two byte instruction requires one additional memory read machine cycle, whereas three byte instruction requires two additional memory read machine cycles.
- ❖ The number of Machine Cycle in 8085 required to execute the instruction depends on the particular instruction. The total number of machine cycles required varies from one to five. It is possible that memory read and memory write machine cycles occur more than once in a single instruction cycle. The following examples illustrate the timing diagrams and machine cycles used for few 8085 instructions.

Concept of Wait States in 8085

- In some applications, speed of memory system and I/O system are not compatible with the microprocessor’s timings. This means that they take longer time to read/write data.
- In such situations, the microprocessor has to confirm whether a peripheral is ready to transfer data or not.
- If READY pin is high, the peripheral is ready otherwise 8085 enters wait state.
- Fig. 1.24 shows the timing diagram for memory read machine cycle with and without wait state.

Concept of Wait States in 8085

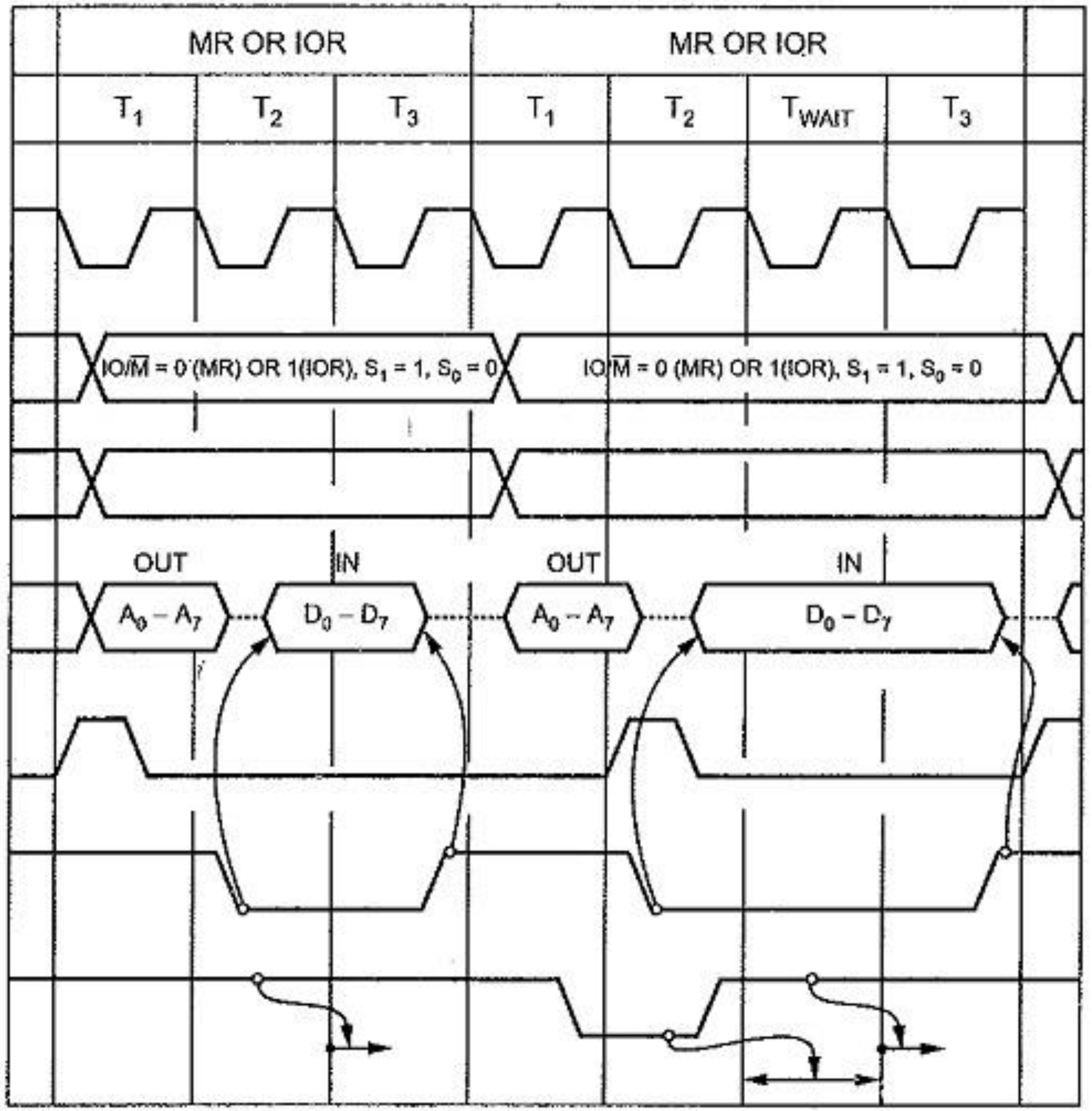


Fig. 1.24 Read machine cycle with and without wait state

Concept of Wait States in 8085

- Concept of Wait States in 8085 continue to be inserted as long as READY is low. After the wait state, 8085 continues with T3 of the machine cycle. During a wait state the contents of the address bus, the data bus, and the control bus are all held constant.
- The wait state then gives an addressed memory or I/O port an extra clock cycle time to output valid data on the data bus. This feature allows to use cheaper memory or I/O devices that have longer access times.

Introduction to the Timing Diagram

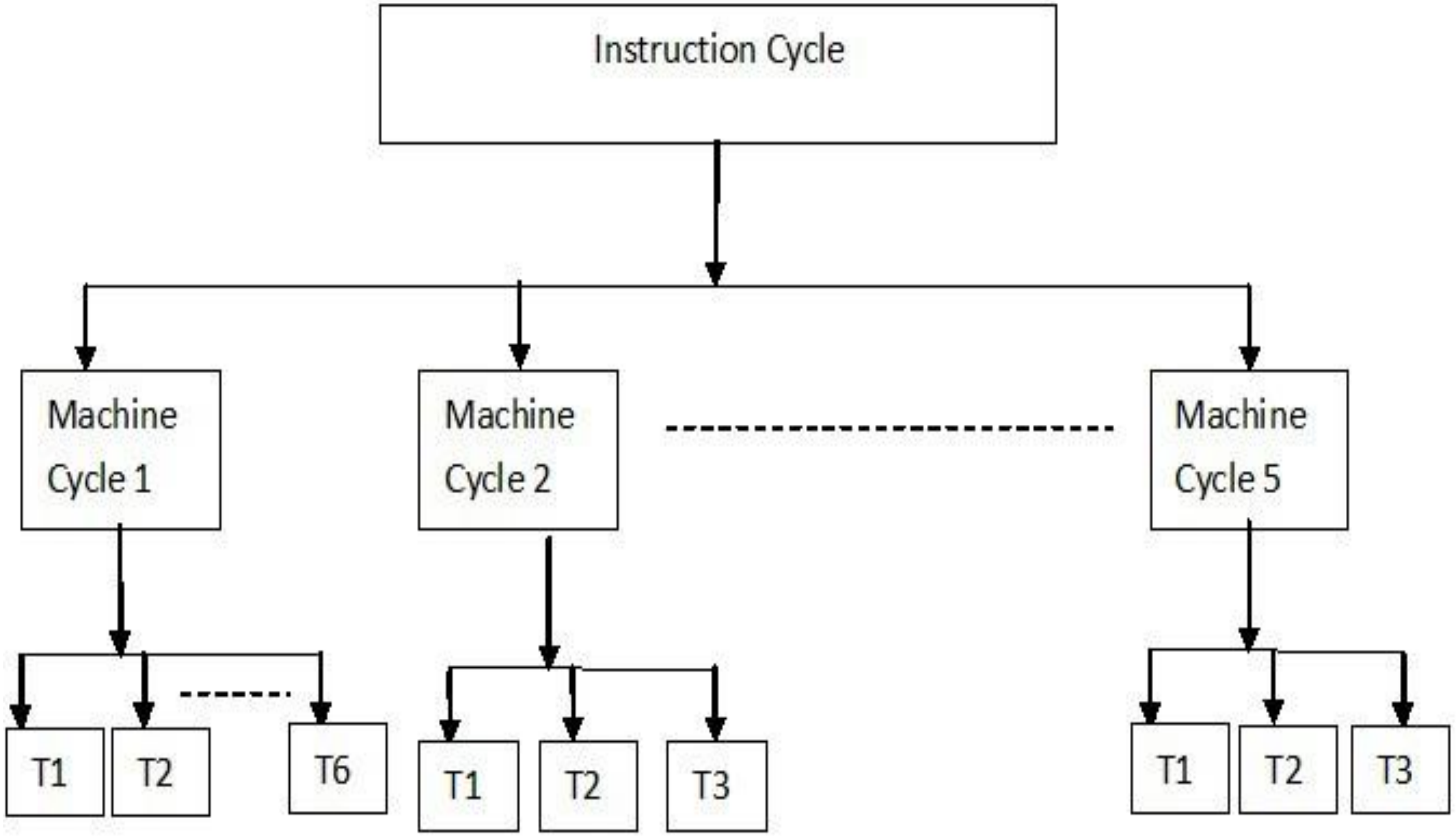
- **Timing Diagram** is a graphical representation of the instruction execution in steps with respect to the time (clock signal). It represents the execution time taken by each instruction in a graphical format. The execution time is represented in T-states. The different types of cycles used in the timing diagram representation are as follows:
- **Instruction Cycle:**

  Instruction cycle is defined as the time required completing the execution of an instruction. The 8085 µP instruction cycle consists of one to five m/c cycles or one to five operations.
- **Machine Cycle:**

  Machine cycle is defined as the time required completing the operation of accessing memory or input/output. In 8085 µP, m/c cycle may consists of three to six timing state (T - state)
- **T-State:**

  T State is defined as one subdivision of the operation performed in one clock period. These subdivisions are internal states synchronized with the system clock.

Introduction to the Timing Diagram



Machine Cycles of Microprocessor 8085

- Opcode fetch cycle (4T/6T)
- Memory read cycle (3 T)
- Memory write cycle (3 T)
- I/O read cycle (3 T)
- I/O write cycle (3 T)
- Halt state machine cycle
- Interrupt acknowledge machine cycle

Machine Cycles

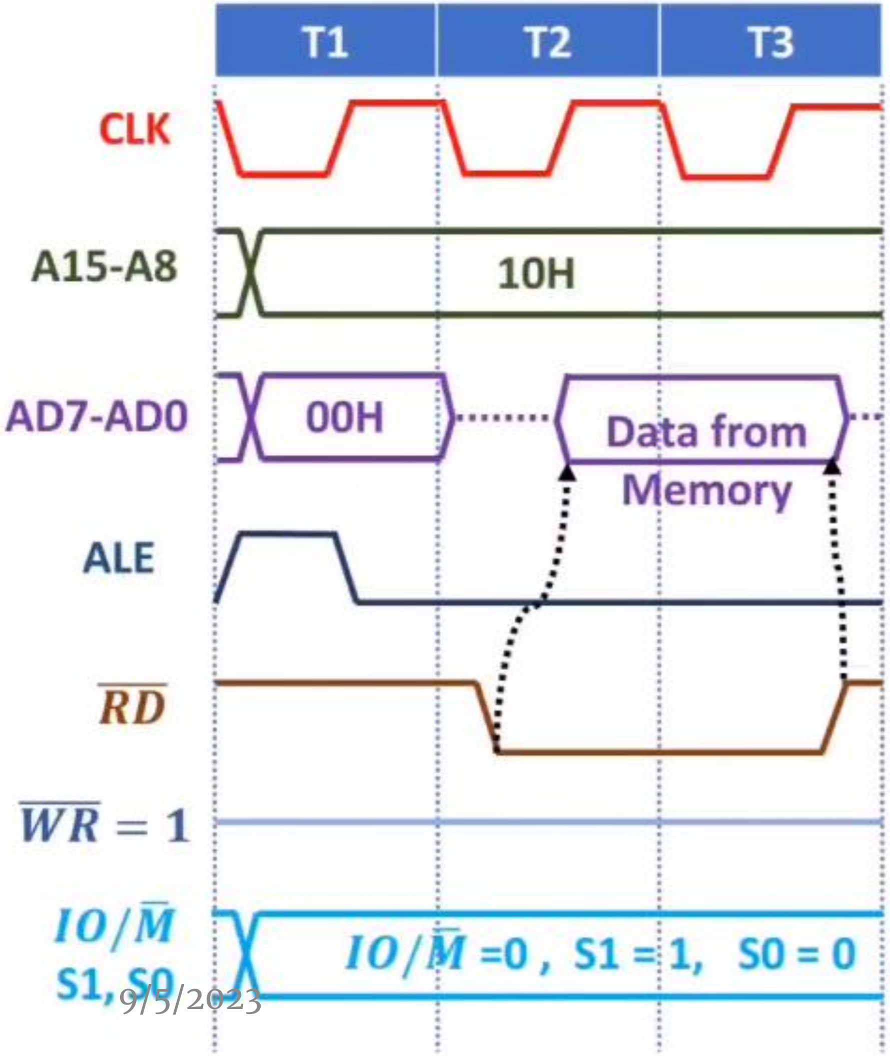
Machine Cycle	Status Signals			Control Signals		
	$IO/\overline{M}$	S1	S0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
IO Read	1	1	0	0	1	1
IO Write	1	0	1	1	0	1
INTR ACK	1	1	1	1	1	0
Bus Idle	0	0	0	1	1	1

Machine Cycles

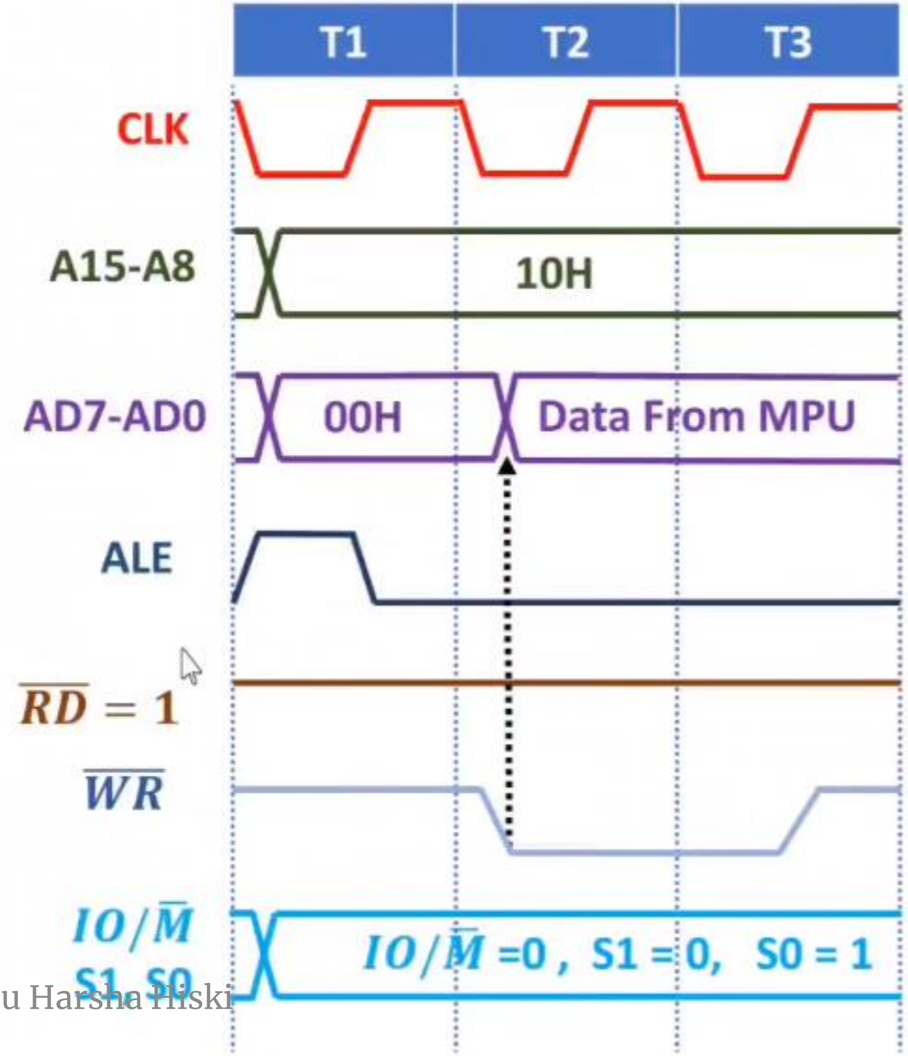
S No	Instruction	No: of machine cycles	Machine cycle - 1	Machine cycle - 2	Machine cycle - 3	Machine cycle - 4
1	**MOV A,B**	1	OF	-	-	-
2	**MVI A, 50H**	2	OF	MR	-	-
3	**LDA 5000H**	4	OF	MR	MR	MR
4	**STA 5000H**	4	OF	MR	MR	MW
5	**IN 80H**	3	OF	MR	IOR	-
6	**OUT 80H**	3	OF	MR	IOW	-

Memory Read & Memory Write Timing Diagram in 8085

❖ Memory Read Machine Cycle



❖ Memory Write Machine Cycle



Opcode Fetch Timing Diagram in 8085
Timing Diagram of MOV instruction

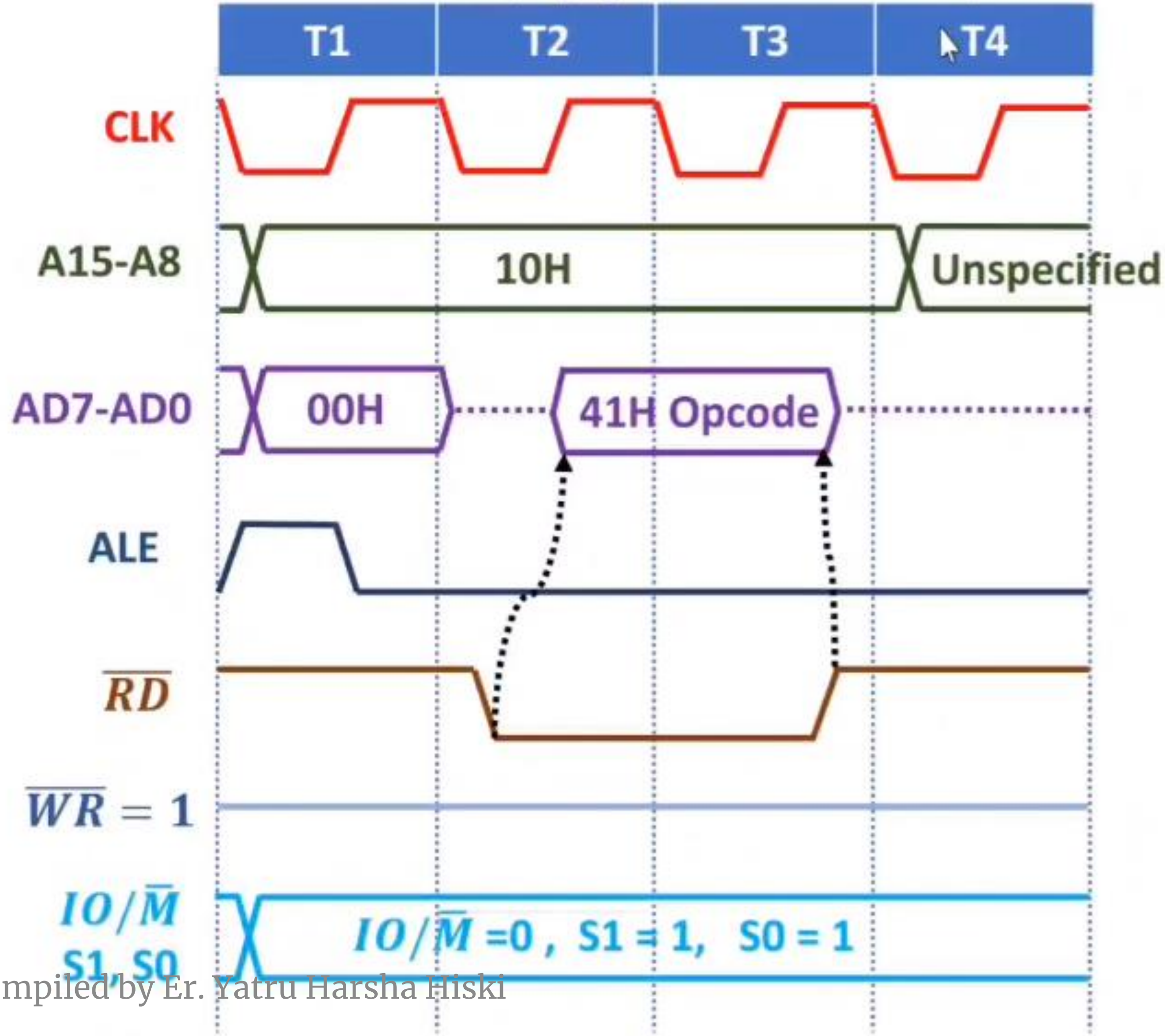
- MOV B,C [Opcode =41H]

PC = 1000H
8085
C= 24H

Add	Data
1000H	41H

- MOV B,C executions needs only opcode fetch machine cycle.
- So, After instruction, B = 24H

- MOV B,C [Opcode Fetch = 41H]



Timing Diagram of MVI instruction in 8085

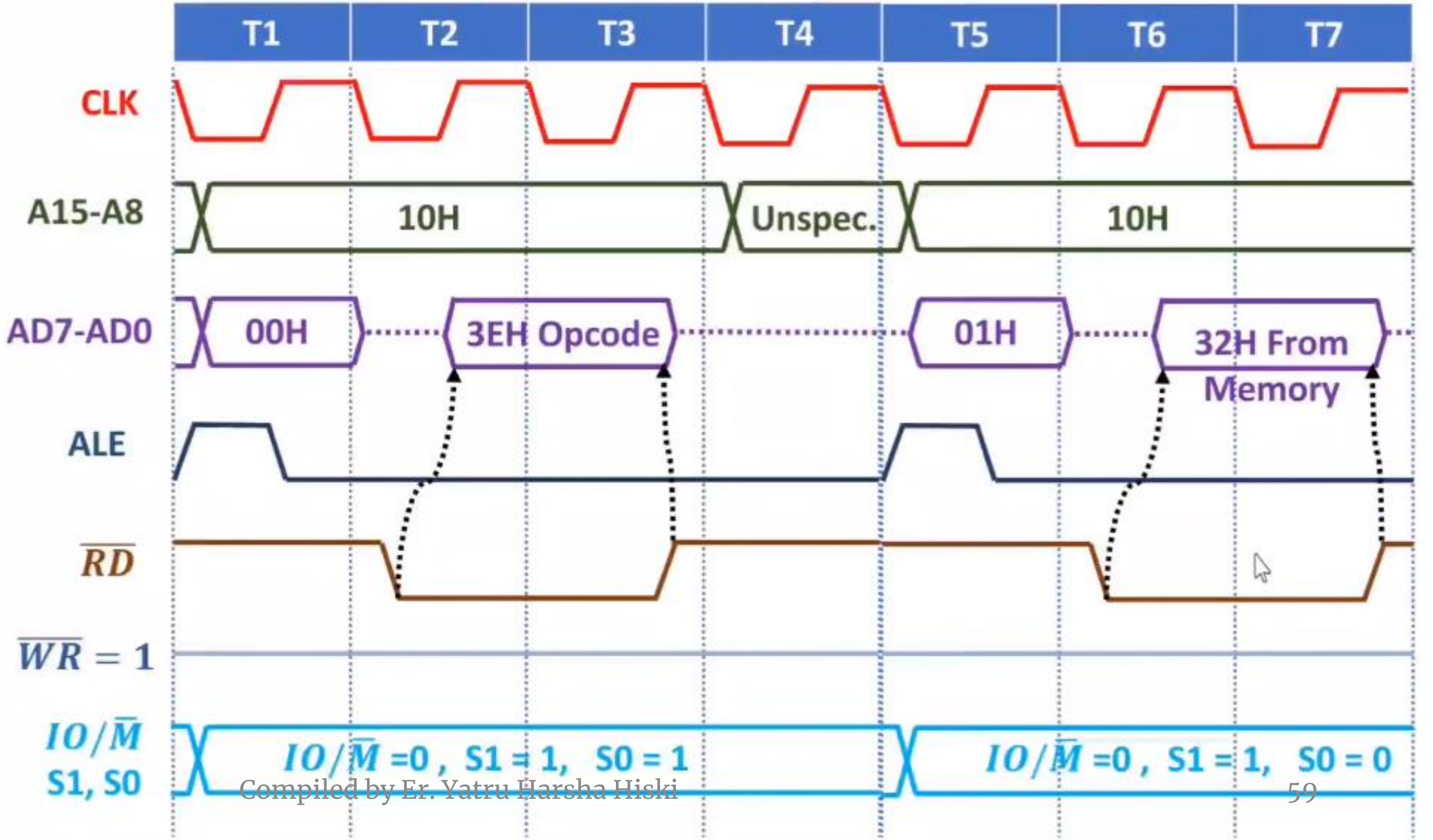
- MVI A,32H [Opcode =3EH]

PC = 1000H
8085
A= 24H

Add	Data
1000H	3EH
1001H	32H

- MVI A,32H executions needs opcode fetch & Memory Read machine cycle.
- So, After instruction
  A = 32H

- MVI A,32H [Opcode Fetch = 3EH]
- Memory Read 32H from 1001H



Timing Diagram of LDA instruction in 8085

- LDA 8000H [Opcode =3AH]

PC = 1000H

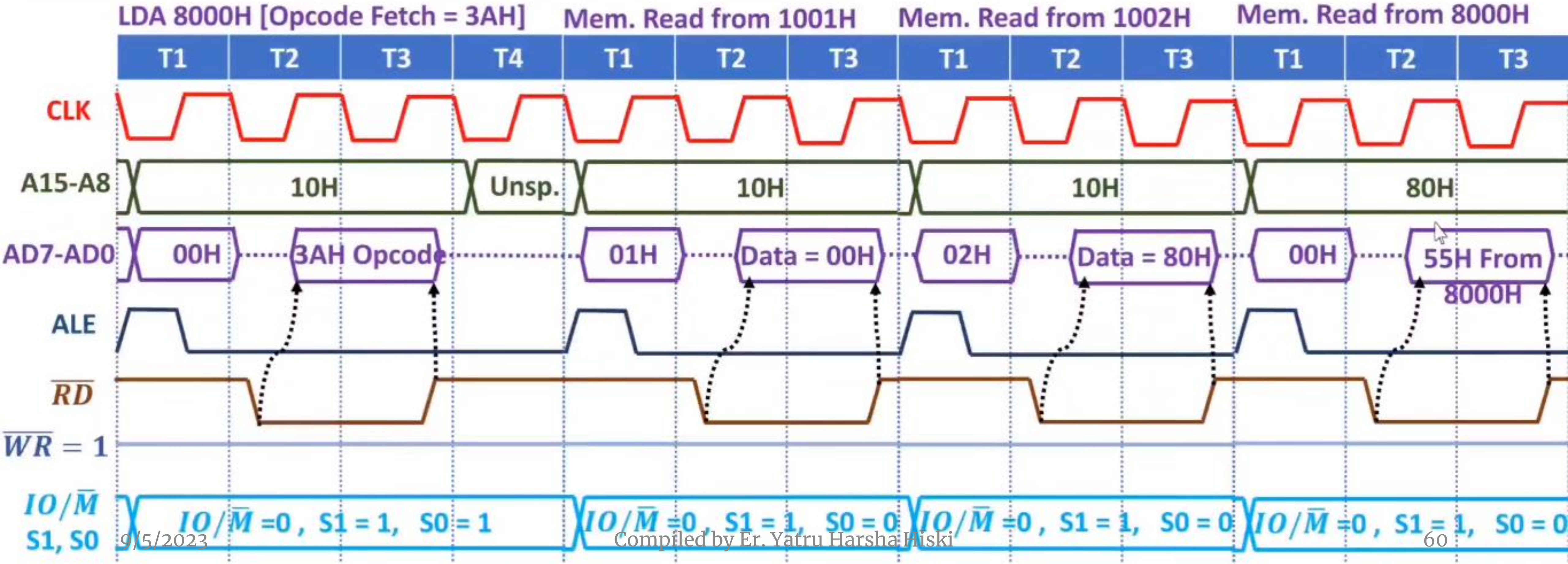
8085

A= 55H

Add	Data
1000H	3AH
1001H	00H
1002H	80H

Add	Data
8000H	55H

- LDA 8000H executions needs opcode fetch and Three Memory Read machine cycles.
- So, After instruction, Accumulator A = 55H will get copied from 8000H Address.



Timing Diagram of STA instruction in 8085

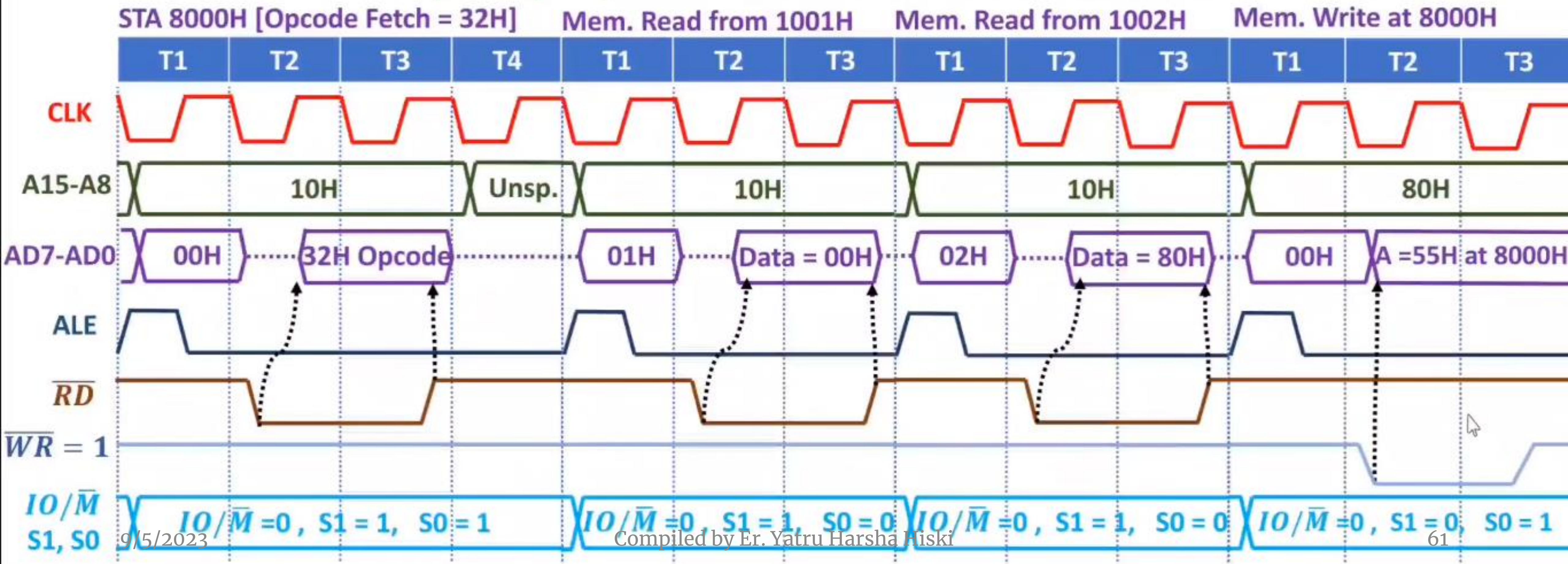
- ❖ STA 8000H [Opcode =32H]

PC = 1000H
8085
A= 55H

Add	Data
1000H	32H
1001H	00H
1002H	80H

Add	Data
8000H	55H

- ❖ STA 8000H executions needs opcode fetch, Memory Read, Memory Read & Memory write machine cycles.
- ❖ So, After instruction, Accumulator A = 55H will get copied to 8000H Address.



Timing Diagram of IN instruction in 8085

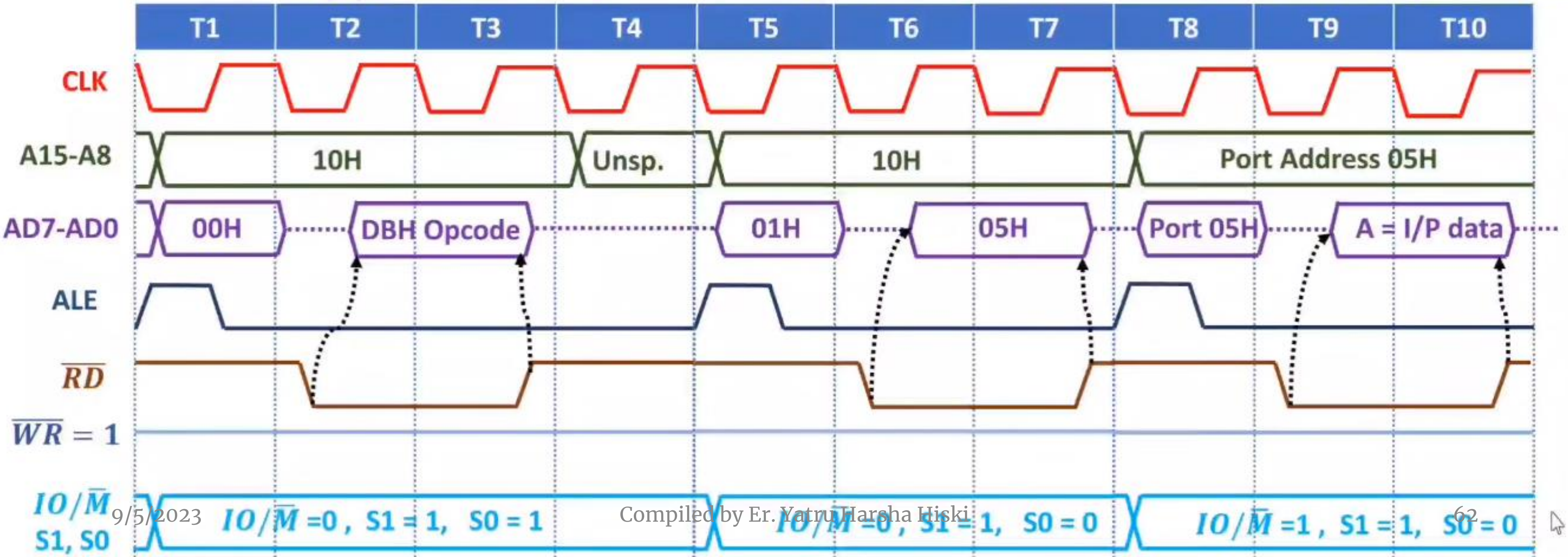
- IN (8 bit Port Address = 05H) [Opcode =DBH]

PC = 1000H
8085
A= 00H

Add	Data
1000H	DBH
1001H	05H

- IN 05H executions needs opcode fetch, Memory Read & IO Read machine cycle.
- So, After instruction, Accumulator will get copied from 05H port Address.

- IN 05H [Opcode Fetch = DBH]
- Memory Read 05H from 1001H
- IO Read into A from port add 05H



Timing Diagram of OUT instruction in 8085

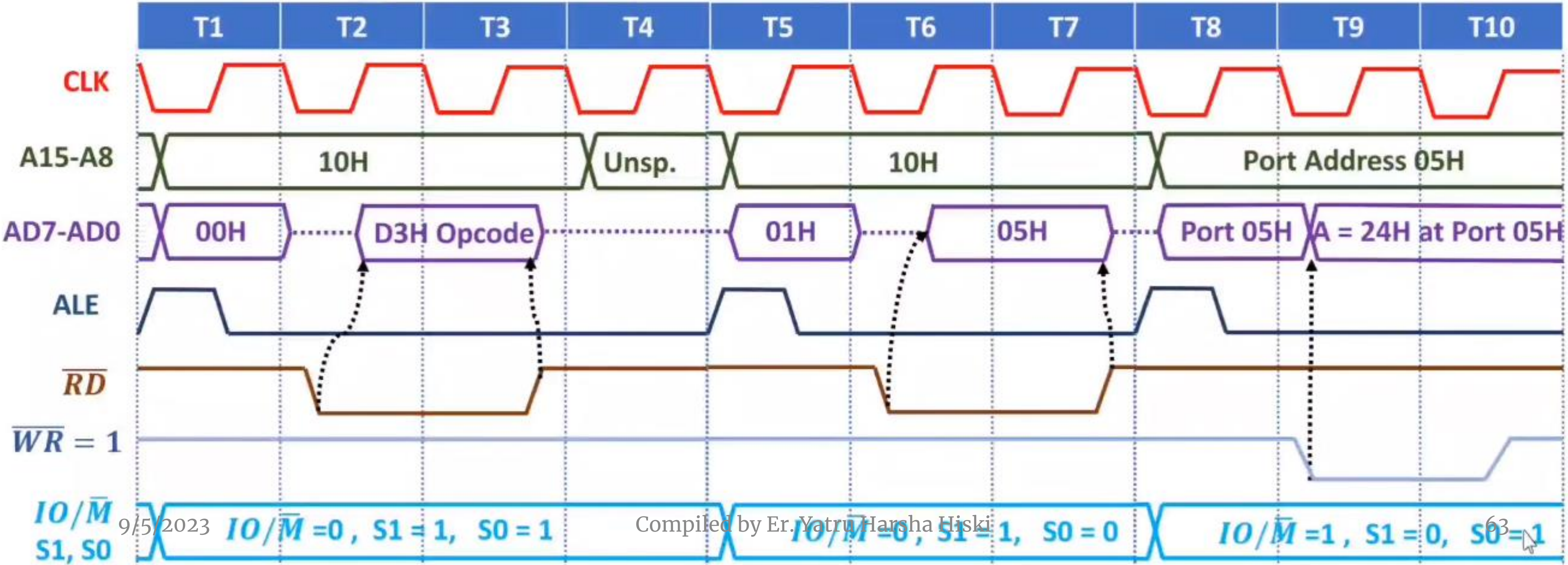
- OUT (8 bit Port Address = 05H) [Opcode =D3H]

PC = 1000H
8085
A= 24H

Add	Data
1000H	D3H
1001H	05H

- OUT 05H executions needs opcode fetch, Memory Read & IO Write machine cycle.
- So, After instruction, A = 24H will get display at 05H port Address.

- OUT 05H [Opcode Fetch = D3H]
- Memory Read 05H from 1001H
- IO Write A at 05H Output Port



Memory Interfacing with 8085 µP

- ❖The programs and data that are executed by the microprocessor have to be stored in ROM/EPROM and RAM, which are basically semiconductor memory chips.
- ❖The programs and data that are stored in ROM/EPROM are not erased even when the power supply to the chip is removed. Hence, ROM/EPROM is called non-volatile memory. It can be used to store **permanent programs** (e.g., **monitor program**, **also known as system start-up program**) and **data** (e.g., **look-up table**), which are necessary in microprocessor-based systems.
- ❖The difference between ROM and EPROM is that a ROM chip can be **programmed only once**, whereas an EPROM chip can be **programmed many times** after erasing the previously stored contents. The contents of an EPROM chip can be erased by passing UV rays for a few minutes through the quartz window situated at the top of the chip.

Memory Interfacing with 8085 µP

- ❖In a RAM, stored programs and data are erased when the **power supply** to the chip is **removed**. Hence, RAM is called **volatile memory**.
- ❖Programs and data that are modified often are stored in the RAM. Examples of such programs and data include programs written during **software development** for a microprocessor-based system, programs written when one is learning assembly language programming, and data entered while testing these programs.
- ❖ RAM is also used to store data that are variable in nature. For example, data to be used by the programmer for arithmetic and logic operations can be stored in the RAM.
- ❖Data can only be read from a ROM or EPROM, whereas it can be written into and read from a RAM.

Memory Interfacing with 8085 µP

- ❖Input and output devices, which are interfaced to the 8085, are essential in any microprocessor-based system. They can be interfaced using two schemes- **I/O mapped I/O** and **memory-mapped I/O**.
- ❖In the **I/O-mapped I/O** scheme, the **I/O devices** are treated differently from memory.
- ❖In the **memory-mapped I/O** scheme, each I/O device is assumed to be a memory location.
- ❖This chapter discusses the interfacing of the following:
  - i. EPROM and RAM chips with the 8085-using address decoders made of either logic gates or decoder ICs (e.g., 74LS138)
  - ii. I/O devices with the 8085-using **I/O-mapped I/O** and **memory-mapped I/O** schemes

Interfacing Memory Chips With 8085

- Since the 8085 has **16 address lines** (A0-A15), a maximum of **64KB** (= 2^{16} bytes) of memory locations can be interfaced with it.
- The memory address space of the 8085 (i.e., the range of memory addresses that can be addressed by the 8085) takes values from **0000H** to **FFFFH** when represented in hexadecimal form.
- While executing a program, the 8085 microprocessor needs to access the memory regularly to read instructions and data and to store results.
- The 8085 initiates a set of signals such as **$IO/\overline{M}$**, **$\overline{RD}$**, and **$\overline{WR}$** when it wants to **read from** and **write into memory**.
- Similarly, each memory chip has signals such as **$\overline{CE}$** or **$\overline{CS}$** (**chip enable** or **chip select**), **$\overline{OE}$** or **$\overline{RD}$** (**output enable** or **read**), and **$\overline{WE}$** or **$\overline{WR}$** (**write enable** or **write**) associated with it.
- The memory interfacing circuit must match the processor’s signals to the memory chip’s signals.

Generation of Control Signals

- When the 8085 wants to read from and write into memory, it activates the **IO/$\overline{\mathbf{M}}$** , $\overline{\mathbf{RD}}$ and $\overline{\mathbf{WR}}$ signals as shown in Table 6.1.

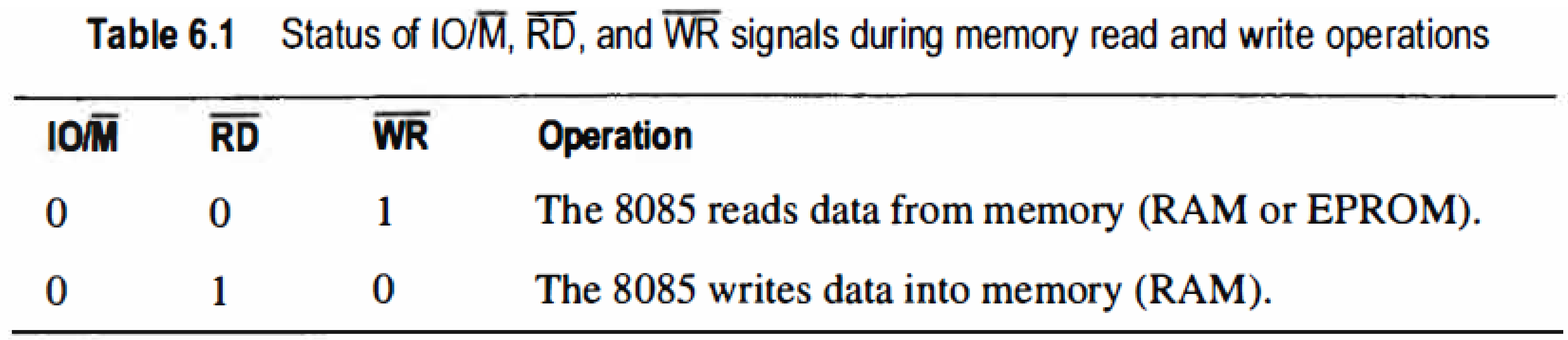


Table 6.1 Status of IO/$\overline{\text{M}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ signals during memory read and write operations

IO/$\overline{\text{M}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	Operation
0	0	1	The 8085 reads data from memory (RAM or EPROM).
0	1	0	The 8085 writes data into memory (RAM).

- Using the **IO/$\overline{\mathbf{M}}$** , $\overline{\mathbf{RD}}$ and $\overline{\mathbf{WR}}$ signals, two control signals $\overline{\mathbf{MEMR}}$ (memory read) and $\overline{\mathbf{MEMW}}$ (memory write) are generated, such that only $\overline{\mathbf{MEMR}}$ is in logic 0 when data is being read from the **RAM** or **EPROM** and only $\overline{\mathbf{MEMW}}$ is in logic 0 when data is being written into the **RAM.**

Generation of Control Signals

- Figure 6.1 shows the circuit used for generation of $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ signals.

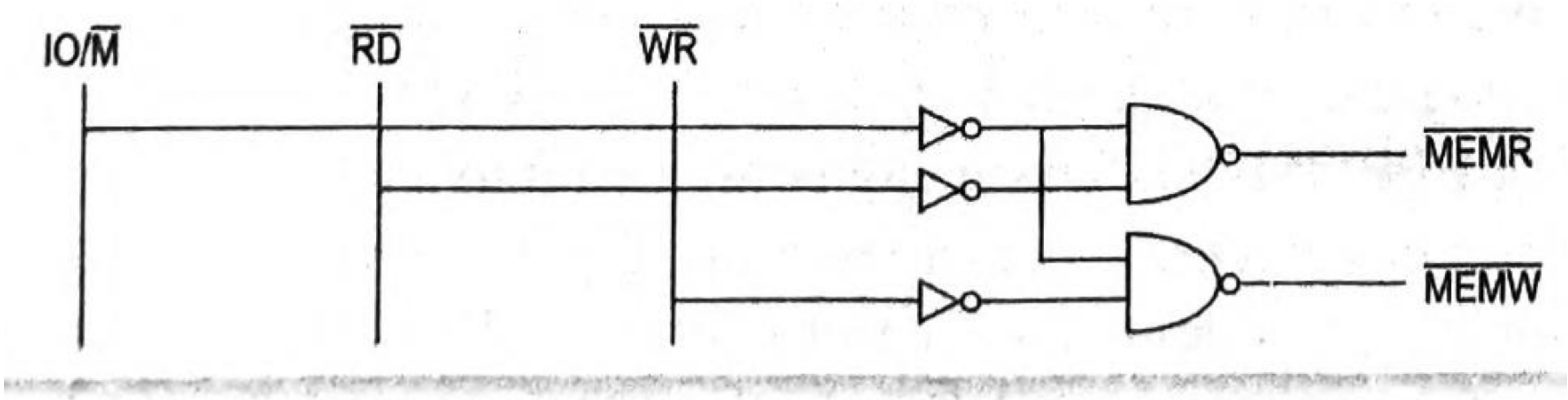


Fig. 6.1 Circuit used to generate $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ signals

Generation of Control Signals

❖ When the $\mathbf{IO/\overline{M}}$ signal is high, both memory control signals are deactivated (i.e., in logic high state) irrespective of the status of the $\mathbf{\overline{RD}}$ and $\mathbf{\overline{WR}}$ signals. This is shown in the third row of Table 6.2.

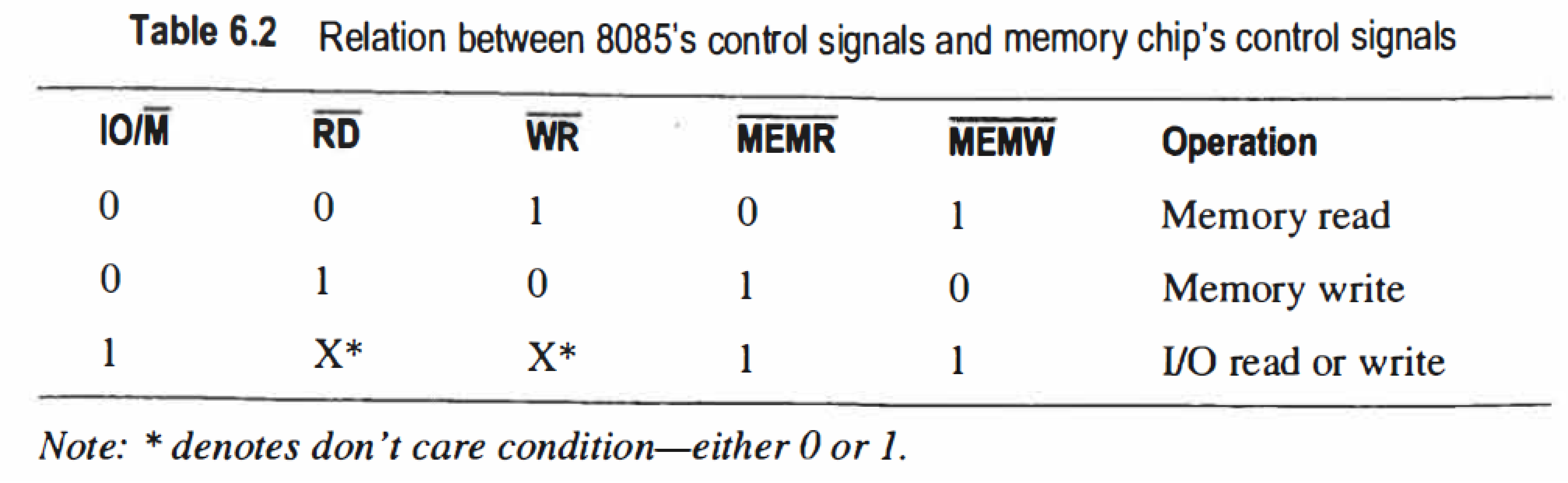
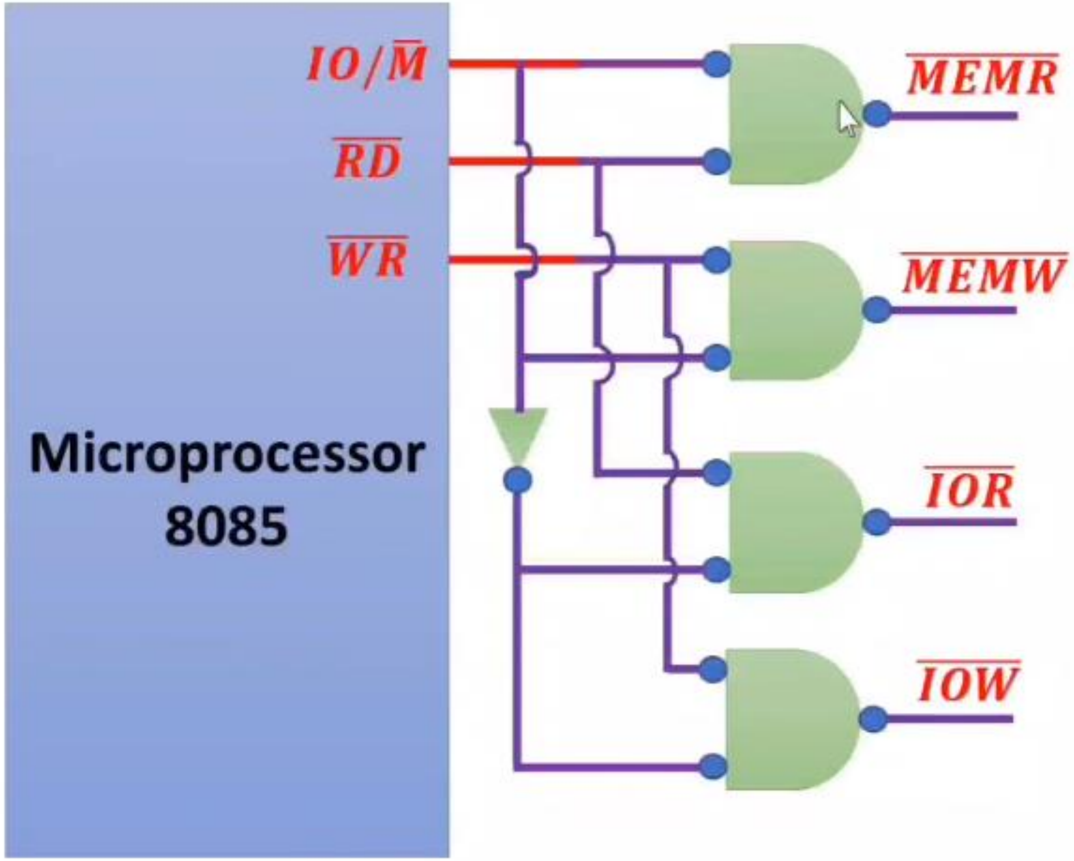


Table 6.2 Relation between 8085's control signals and memory chip's control signals

$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$	$\overline{MEMR}$	$\overline{MEMW}$	Operation
0	0	1	0	1	Memory read
0	1	0	1	0	Memory write
1	X*	X*	1	1	I/O read or write

*Note: * denotes don't care condition—either 0 or 1.*

Generation of Control Signals in 8085



- Using $IO/\overline{M}$, $\overline{RD}$ and $\overline{WR}$ we can generate four control signals: Memory Read, Memory Write, IO Read and IO Write.
- $IO/\overline{M}$ = Input-Output or Memory
  - If it is logic ‘1’, IO operations should be done by 8085.
  - If it is logic ‘0’, Memory operations should be done by 8085.
- $\overline{RD}$ = Read
  - It is Active Low signal which indicates read operation to be performed by 8085 over data lines.
  - This read operation may be there with memory or IO devices.
- $\overline{WR}$ = Write
  - It is Active Low signal which indicates write operation to be performed by 8085 over data lines.
  - This write operation may be there with memory or IO devices.

$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$	Control Signals
0	0	1	Memory Read - $\overline{MEMR}$
0	1	0	Memory Write - $\overline{MEMW}$
1	0	1	Input Output Read - $\overline{IOR}$
1	1	0	Input Output Write - $\overline{IOW}$

Generation of Control Signals

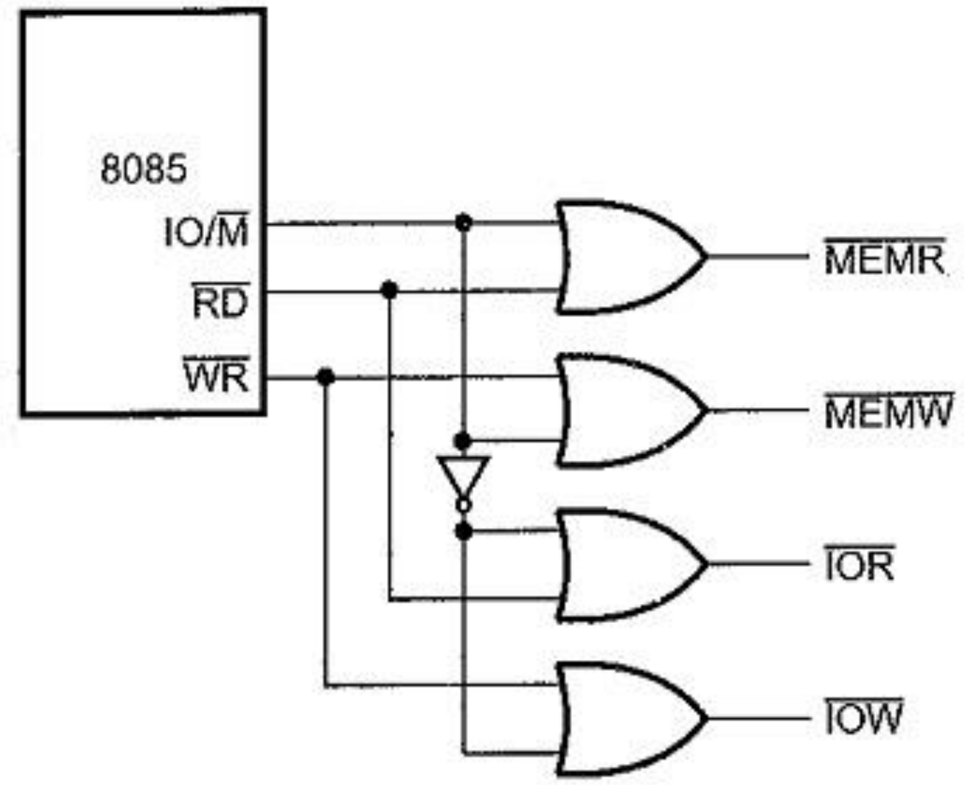


Fig. 4.8 Generation of MEMR, MEMW, IOR and IOW signals

Generation of Control Signals

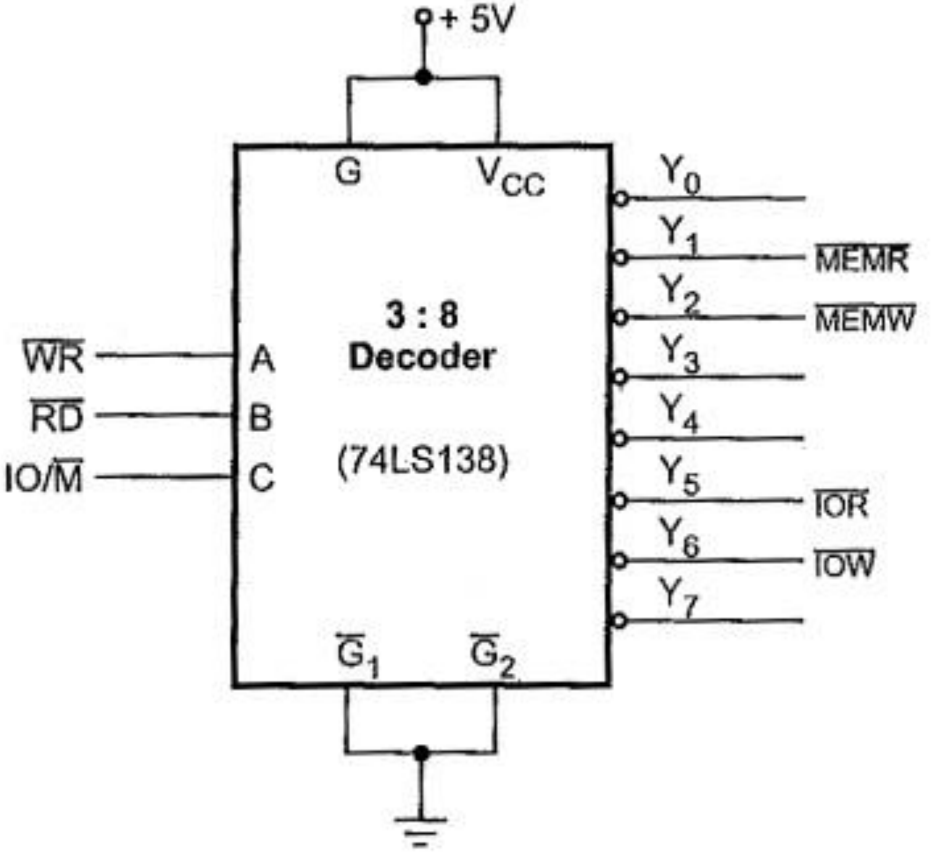
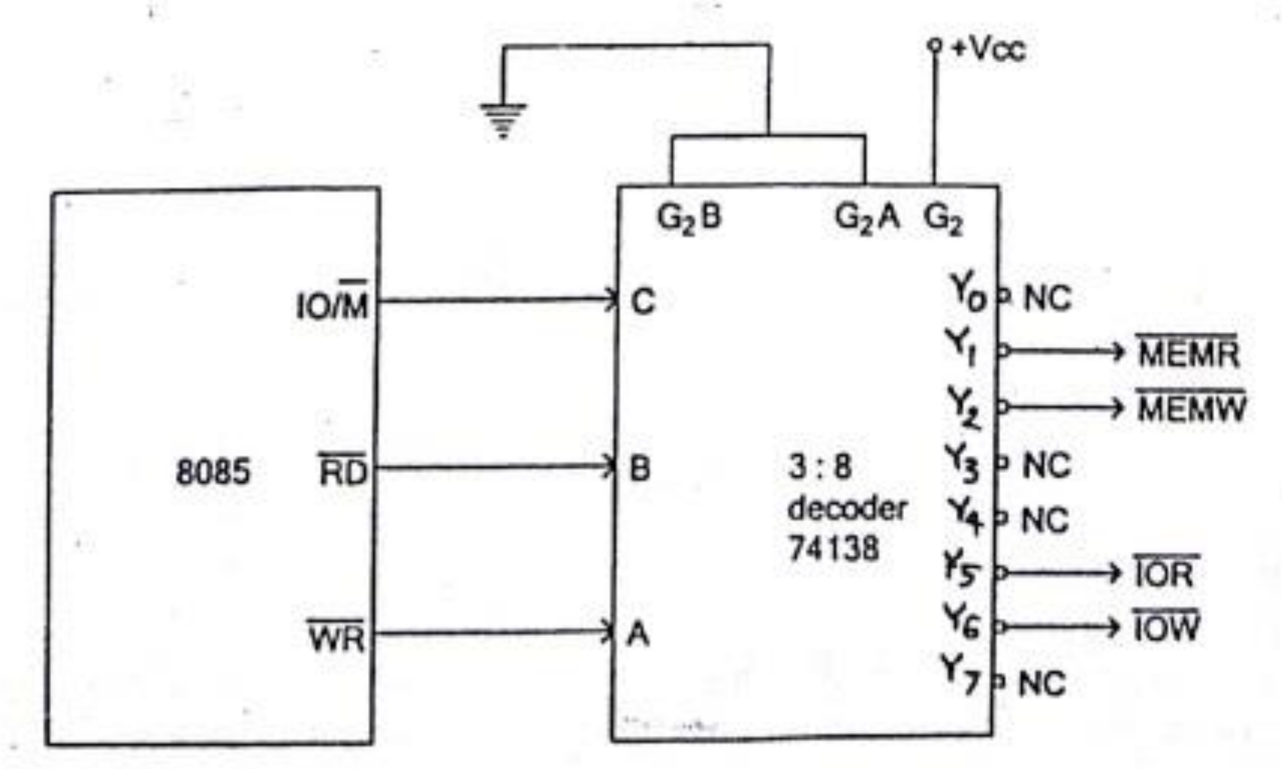


Fig. 4.9 Generation of control signals using 3 : 8 decoder

Generation of Control Signals



Memory Interfacing in Microprocessor 8085

Draw the interfacing of a 4K EPROM having a starting address 2000H with 8085 microprocessor. Use demultiplexed address/data lines and 3-to-8 decoder (74LS138).

→ In memory Interfacing four types of Signals are req^{ed}

1) Address lines.
2) Data lines.
3) Control lines.
4) Chip Select.

→ 4K EPROM

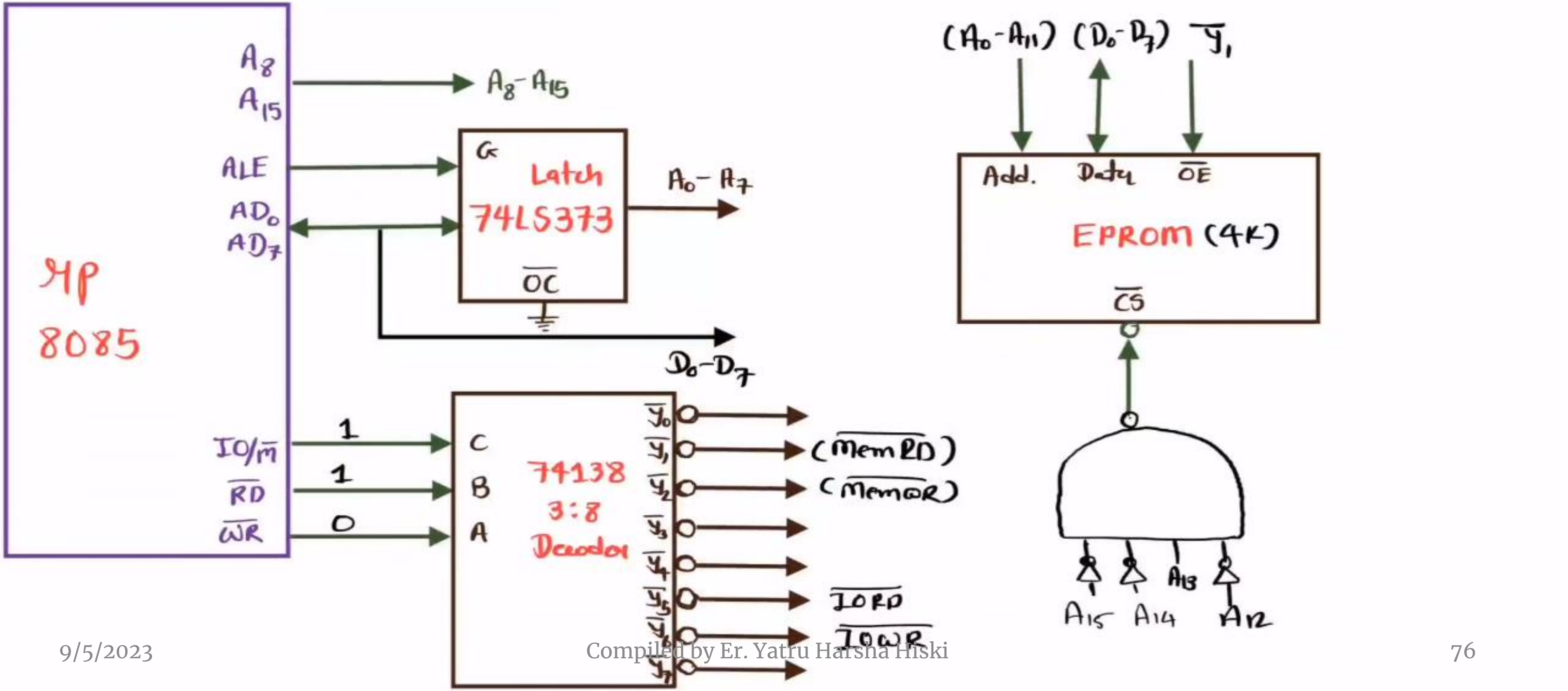
- Address lines

$$\Rightarrow 4K = 4 \times K = 2^{2} \times 2^{10} = 2^{12}$$

- So, 12 Address lines are req^{ed}.
- Data lines = 8
- Control signal = Memory Read.

Memory Mapping

Memory Chip	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Memory Address
EPROM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 H
4K	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	2FFF H



Memory Interfacing in Microprocessor 8085

Interface 4K EPROM and 16K RAM with 8085 processor. Write address range for both the memory chips and also show the address decoding logic.

⇒ In Memory Interfacing four types of Signals are req.^{ed}

1] Address lines.

2] Data lines.

3] Control lines.

4] Chip Select.

→ For 4K EPROM

- Address lines
  $\Rightarrow 4K = 4 \times K = 2^2 \times 2^{10} = 2^{12}$
- So total 12 Address lines are req^d (A₀ - A₁₁)
- Data lines = 8 ($D_0 - D_7$)
- Control lines = Memory Read.

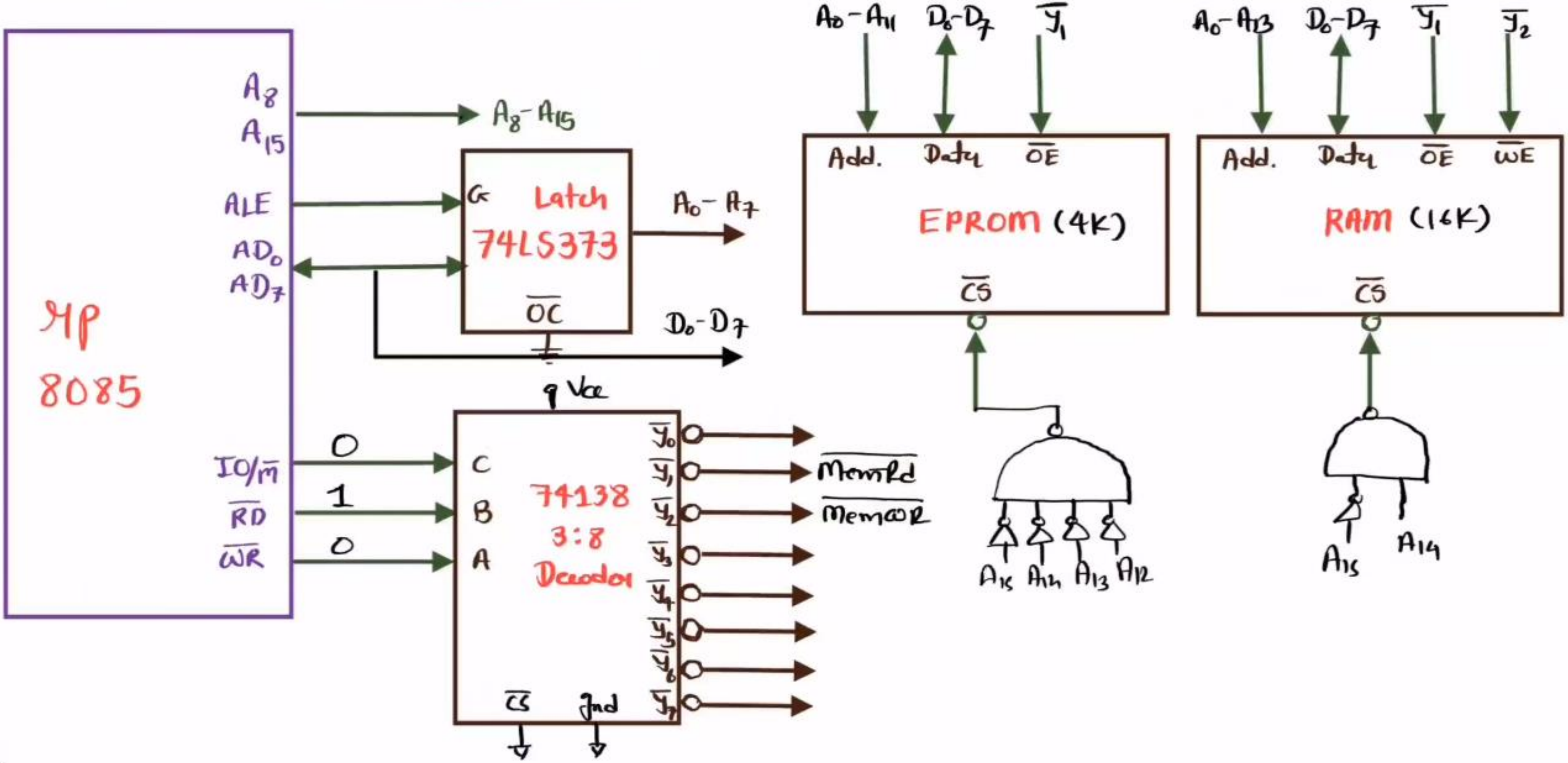
→ 16K RAM

- Address lines
  $\Rightarrow 16K = 2^4 \times 2^{10} = 2^{14}$
- Address lines = 14 ($A_0 - A_{13}$)
- Data lines = 8 ($D_0 - D_7$)
- Control lines = Memory Read & Memory Write

Memory Mapping

Chip Select

Memory Chip	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Memory Address
EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 H
4K	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF H
RAM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 H
16K	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFF H



µP
8085
A8
A15
A8 - A15
ALE
AD0
AD7
G
Latch
74LS373
OC
A0 - A7
D0 - D7
Vcc
IO/M̄
RD̄
WR̄
0
1
0
C
B
A
74138
3:8
Decoder
CS̄
Gnd
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
MemRd̄
MemWR̄
A0 - A11
D0 - D7
Ȳ1
Add.
Data
OE
EPROM (4K)
CS̄
A15
A14
A13
A12
A0 - A13
D0 - D7
Ȳ1
Ȳ2
Add.
Data
OE
WE
RAM (16K)
CS̄
A15
A14

Memory Interfacing in Microprocessor 8085

Interface 8KB EPROM and 16KB RAM with 8085 processor using the chips of 4KB of EPROM and 8KB of RAM. Show decoding of IC with 74LS138 Decoder.

↝ 8KB EPROM

↝ No of chips = 2 Chips of 4KB of EPROM.

↝ Address lines

$\Rightarrow 4K = 2^2 \times 2^{10} = 2^{12}$

↝ So, total 12 Address lines req.^{ed} [A_0-A_{11}]

↝ Data lines = 8 [D_0-D_7]

↝ Control signal = Memory Read.

↝ 16 KB RAM

↝ No of chips = 2 chips of 8KB RAM

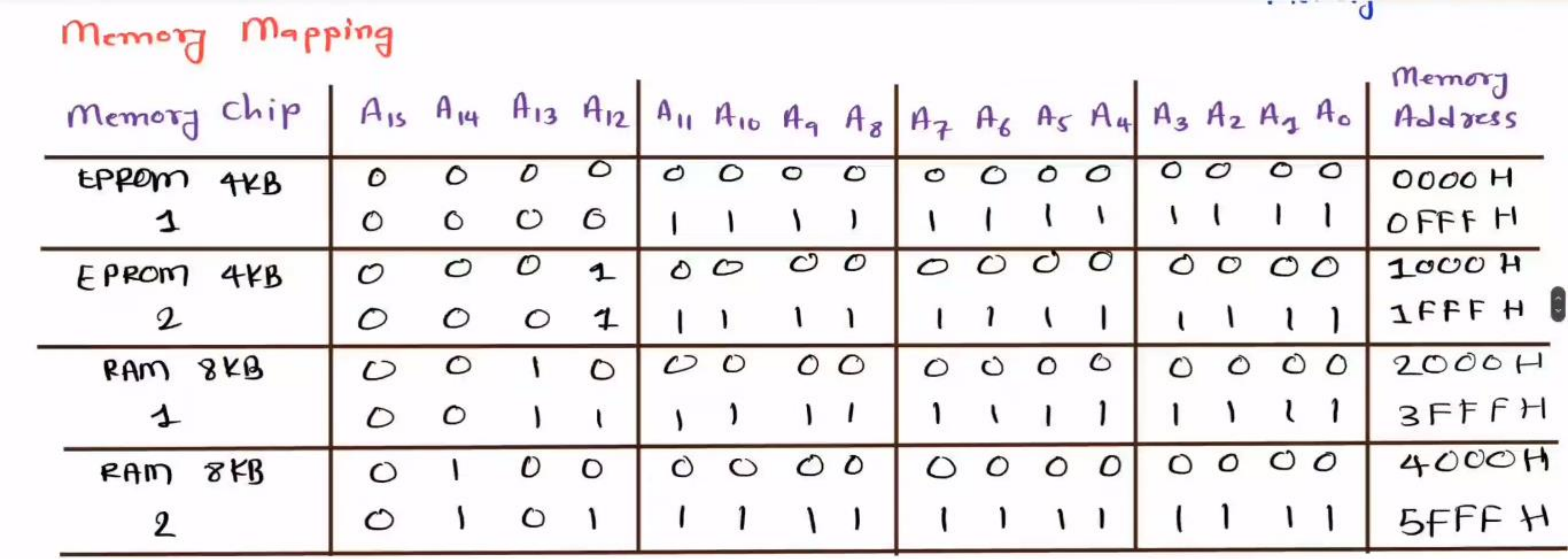
↝ Address lines

$\Rightarrow 8K = 2^3 \times 2^{10} = 2^{13}$

↝ So, total 13 Address lines. [A_0-A_{12}]

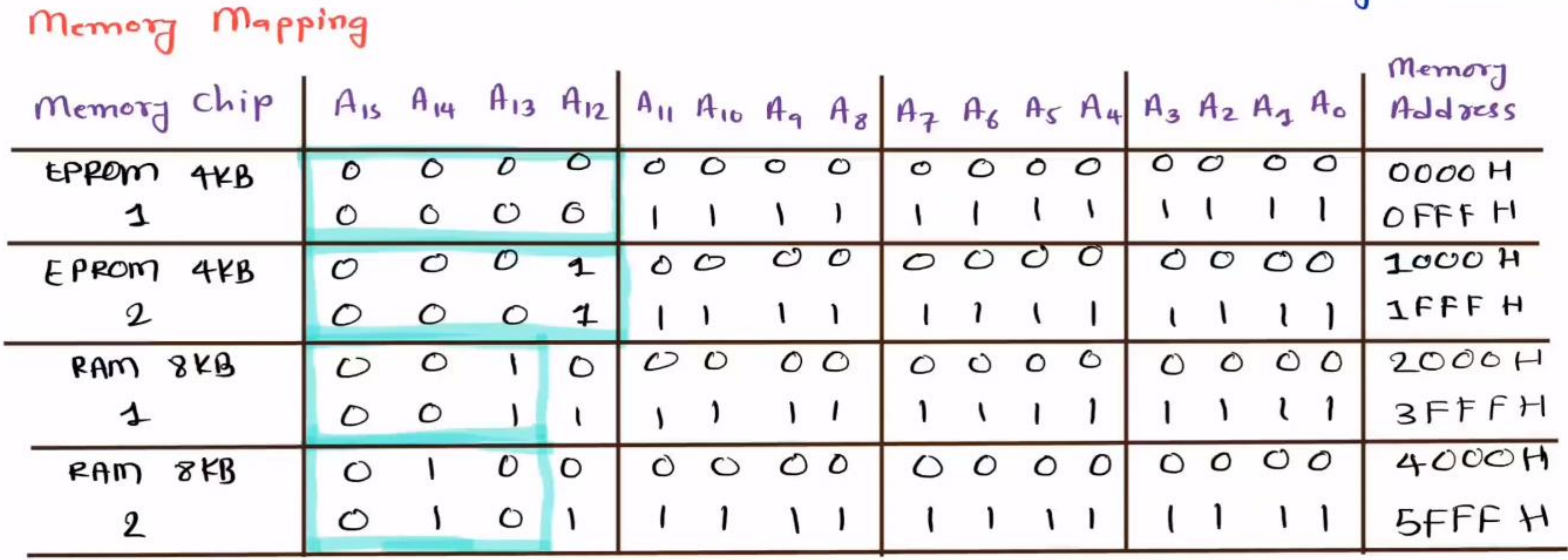
↝ Data lines = 8 [D_0-D_7]

↝ Control signal = Memory Read,
Memory write.



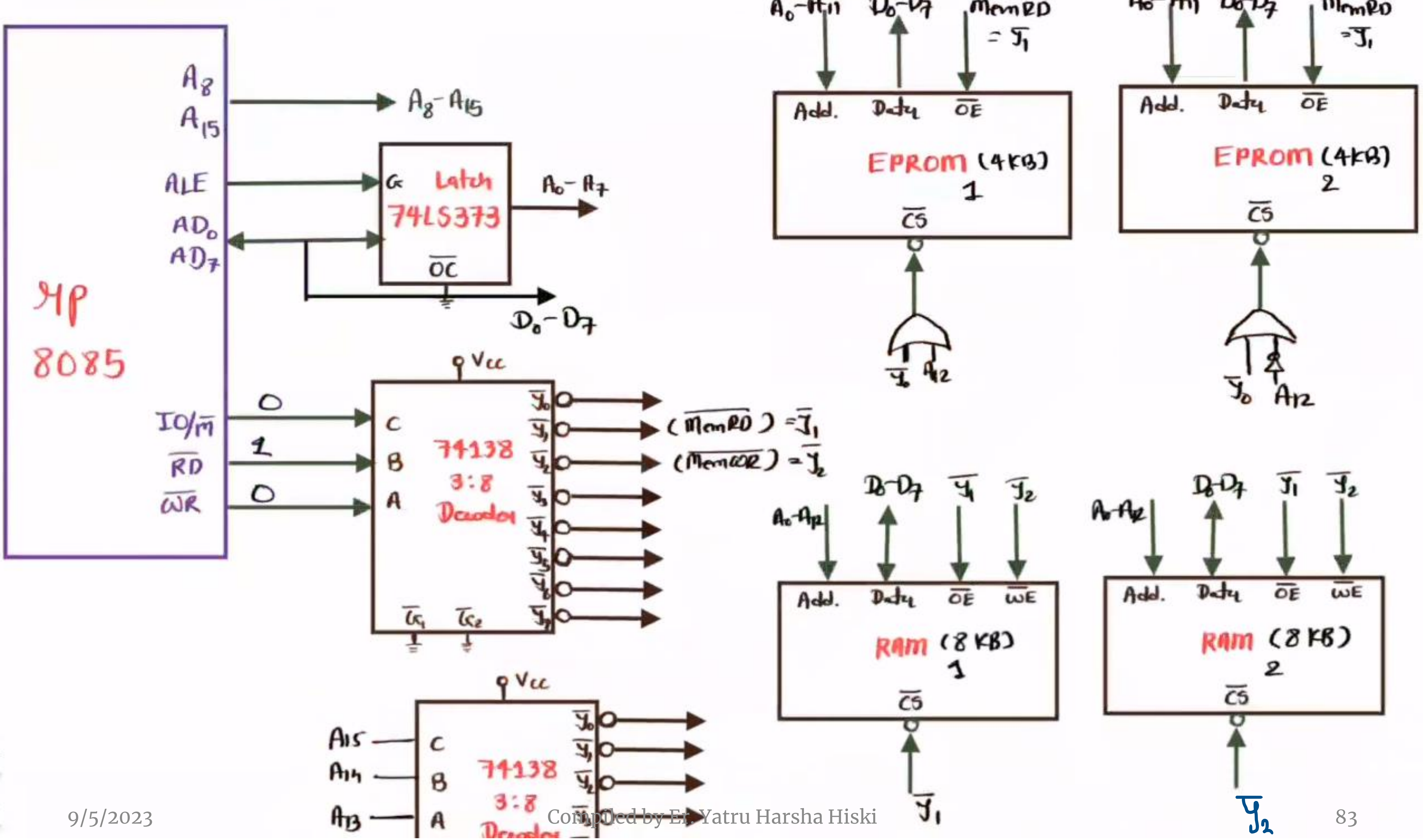
Memory Mapping

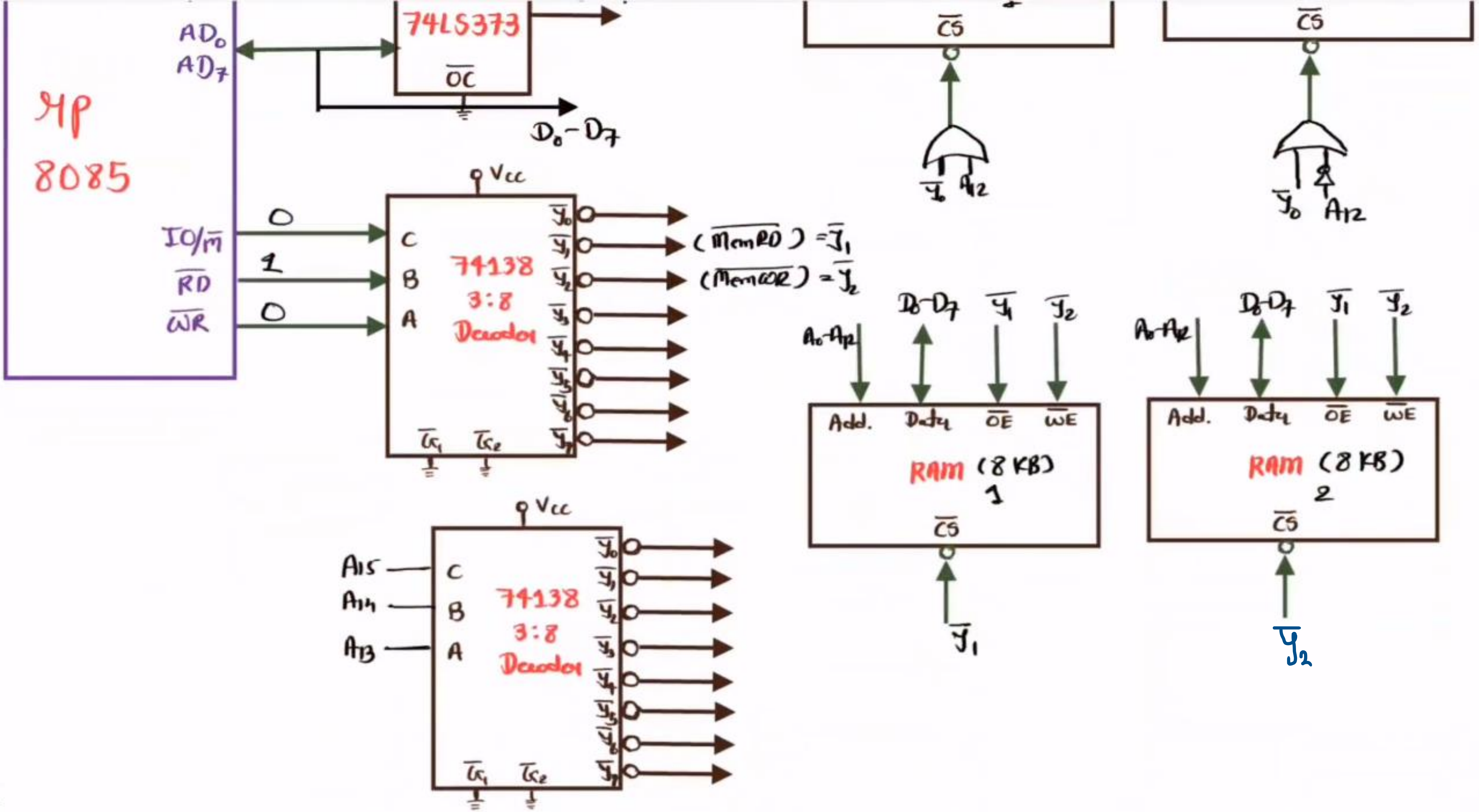
Memory Chip	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Memory Address
EPROM 4KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 H
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF H
EPROM 4KB	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000 H
2	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF H
RAM 8KB	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 H
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF H
RAM 8KB	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 H
2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF H



Memory Mapping

Memory Chip	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Memory Address
EPROM 4KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 H
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF H
EPROM 4KB	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000 H
2	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF H
RAM 8KB	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 H
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF H
RAM 8KB	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 H
2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF H


μP 8085
A8
A15
A8-A15
ALE
AD0
AD7
G
Latch 74LS373
A0-A7
OC
D0-D7
IO/M̄
RD̄
WR̄
0
1
0
Vcc
C
B
A
74138 3:8 Decoder
Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7
G1 G2
(MemRD) = Y1
(MemWR) = Y2
A0-A11
D0-D7
MemRD = Y1
Add.
Data
OE
EPROM (4KB) 1
CS
Y0
A12
EPROM (4KB) 2
A0-A12
D0-D7
Y1
Y2
WE
RAM (8KB) 1
RAM (8KB) 2
Vcc
A15
A14
A13
74138 3:8 Decoder
Y0 Y1 Y2 Y3
Y1
Y2


AD0
AD7
μP
8085
IO/$\overline{M}$
$\overline{RD}$
$\overline{WR}$
74LS373
$\overline{OC}$
D0-D7
Vcc
0
1
0
C
B
A
74138
3:8
Decoder
$\overline{Y_0}$
$\overline{Y_1}$
$\overline{Y_2}$
$\overline{Y_3}$
$\overline{Y_4}$
$\overline{Y_5}$
$\overline{Y_6}$
$\overline{Y_7}$
$\overline{G_1}$
$\overline{G_2}$
($\overline{MemRD}$) = $\overline{Y_1}$
($\overline{MemWR}$) = $\overline{Y_2}$
A15
A14
A13
$\overline{CS}$
$\overline{Y_0}$
A12
A0-A12
D0-D7
$\overline{Y_1}$
$\overline{Y_2}$
Add.
Data
$\overline{OE}$
$\overline{WE}$
RAM (8 KB)
1
2

END OF UNIT 3

THANK YOU