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

Instruction Cycle, Machine Cycle and T-states

- **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.

Instruction Cycle, Machine Cycle and T-states

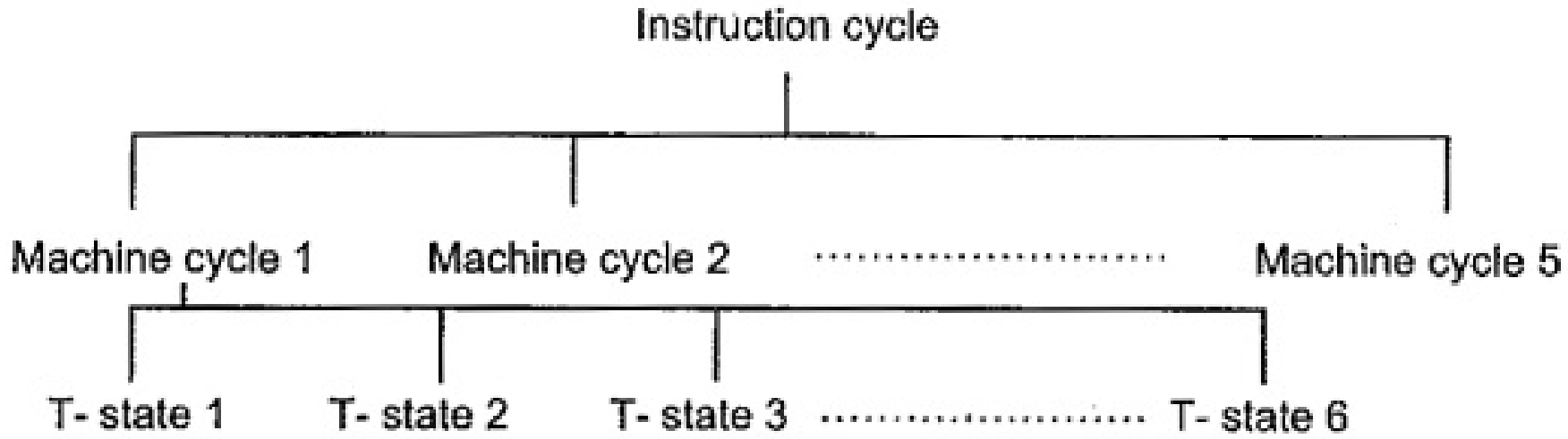


Fig. 1.4 Relation between instruction cycle, machine cycle and T-state

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.

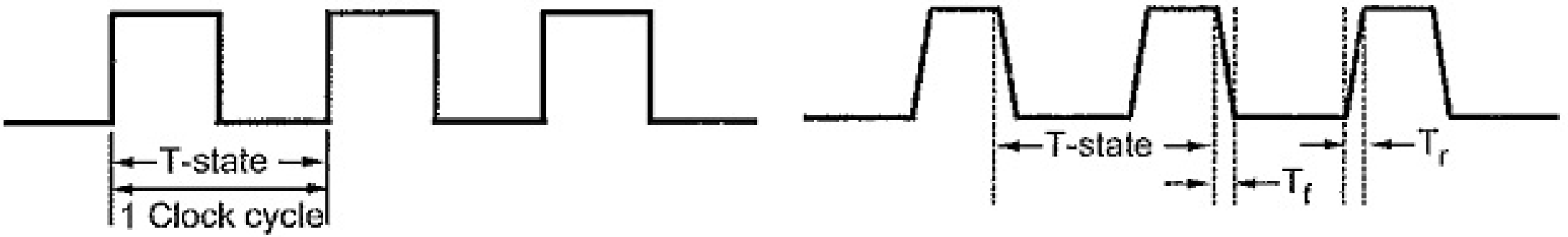
VI. Interrupt acknowledge cycle

- 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.

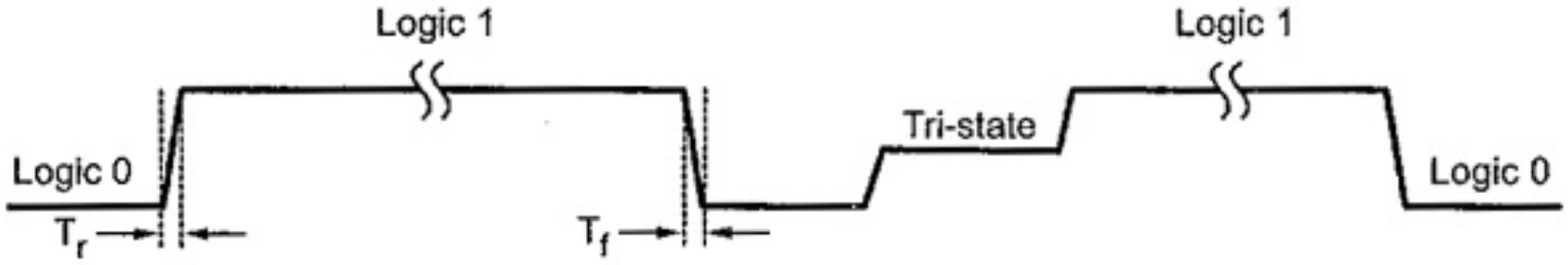


Fig. 1.6 Single signal representation

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.

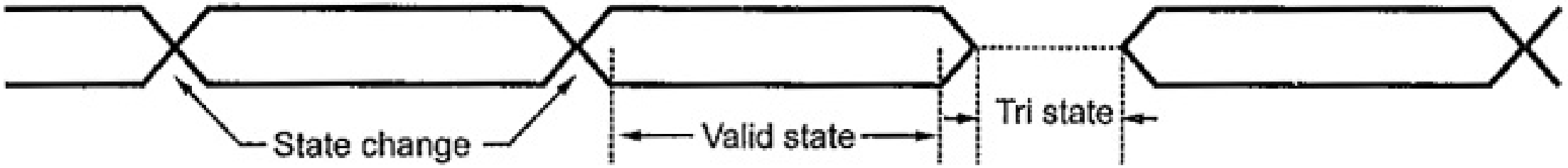


Fig. 1.7 Group of signals representation

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.

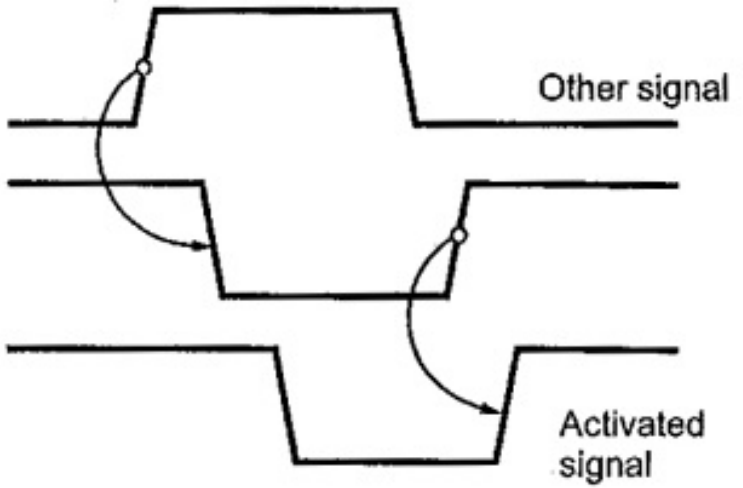


Fig. 1.8 (a) Activation of signal with the change in state of other signal

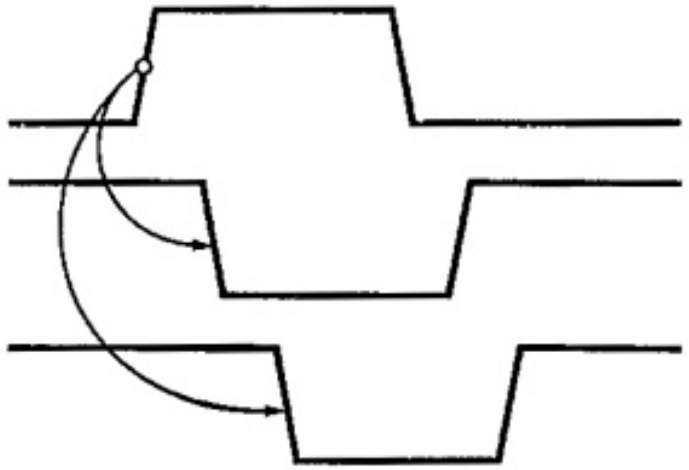


Fig. 1.8 (b) Activation of signal with the change in state of other signal

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.

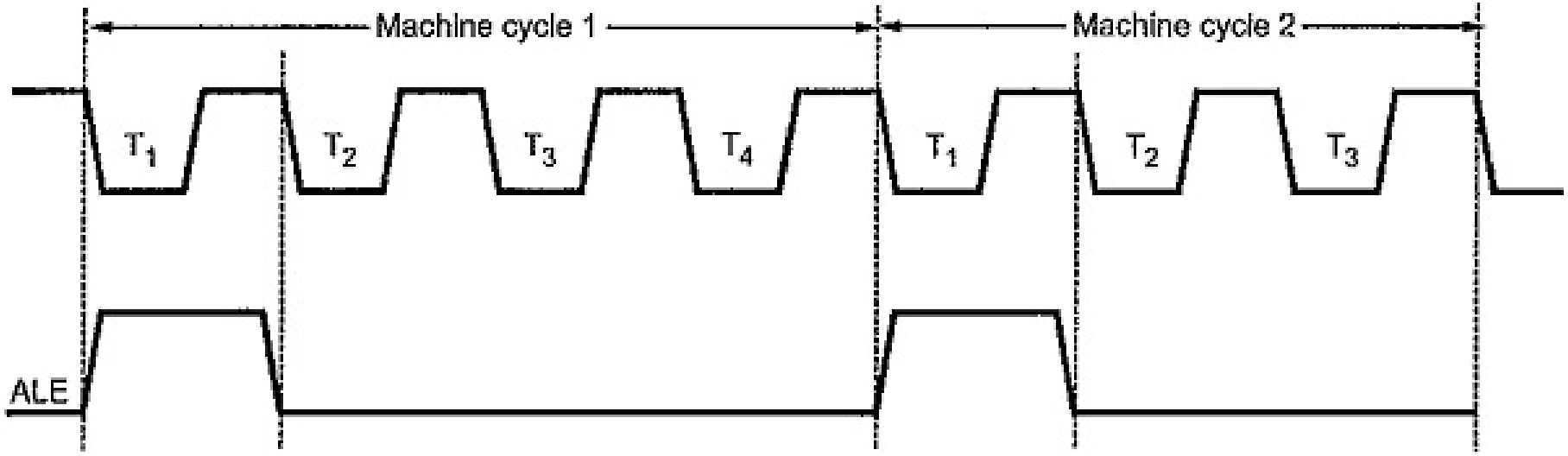


Fig. 1.9 ALE activation and its period

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.

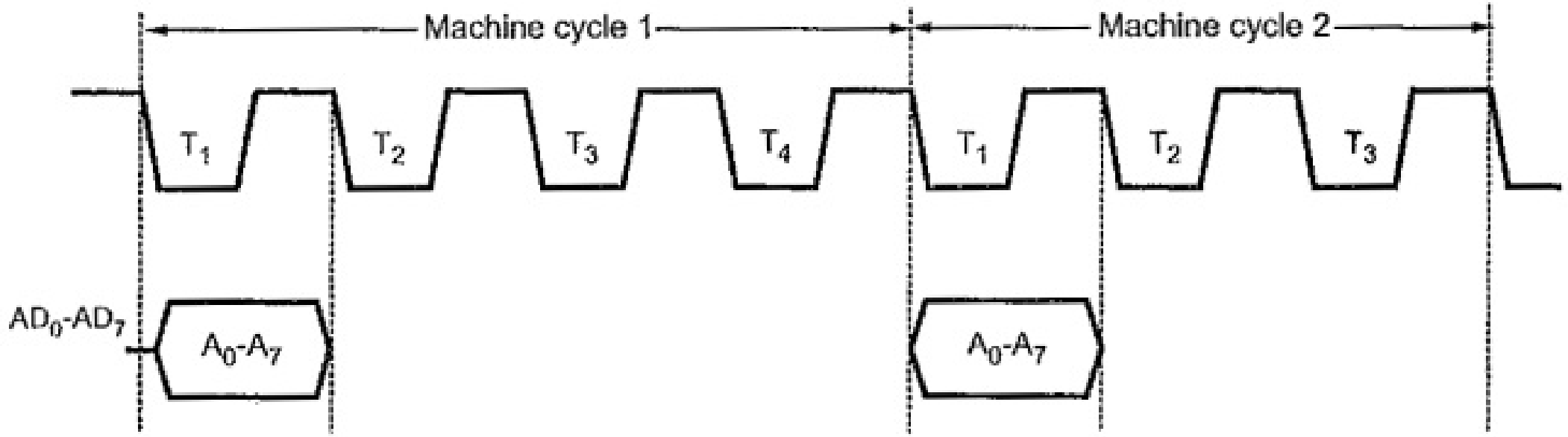
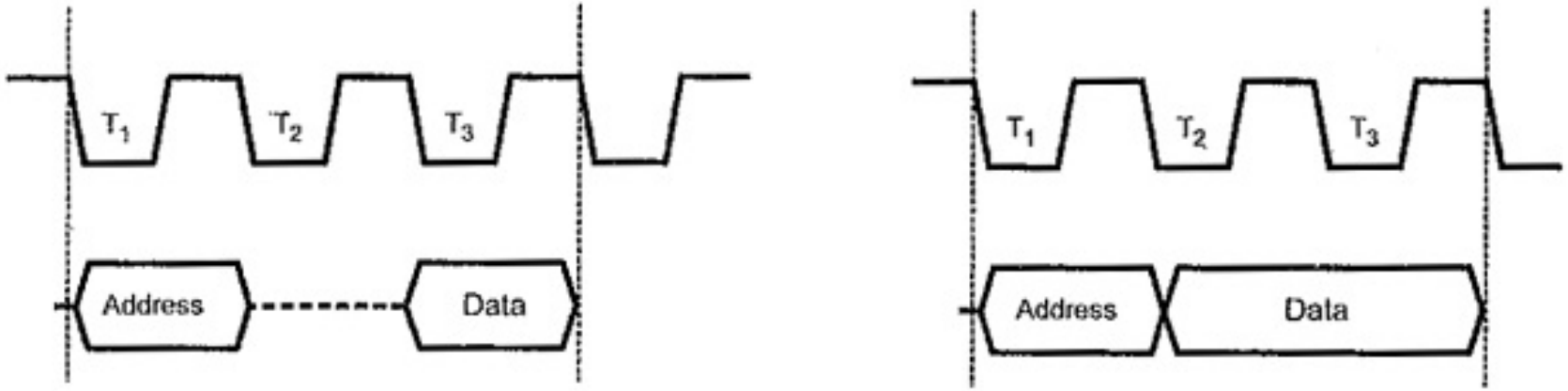


Fig. 1.10 Lower address on the multiplexed bus

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.



(a) **Fig. 1.11 Data bus** (b)

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.

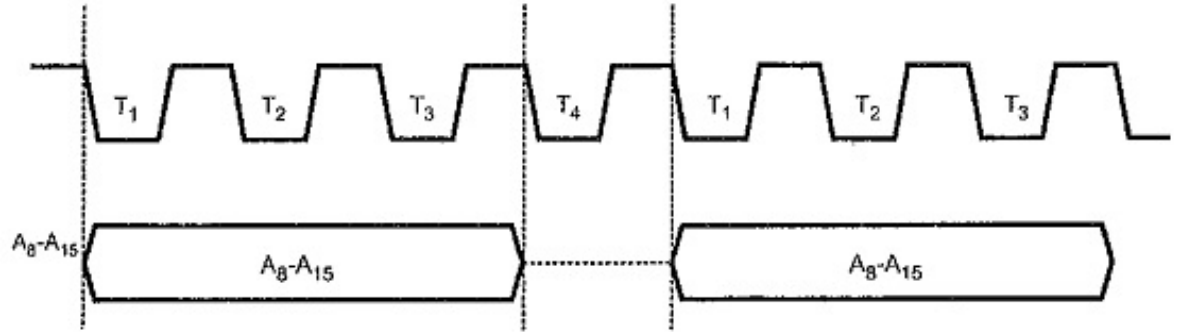


Fig. 1.12 Higher byte address on A_8–A_{15}

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

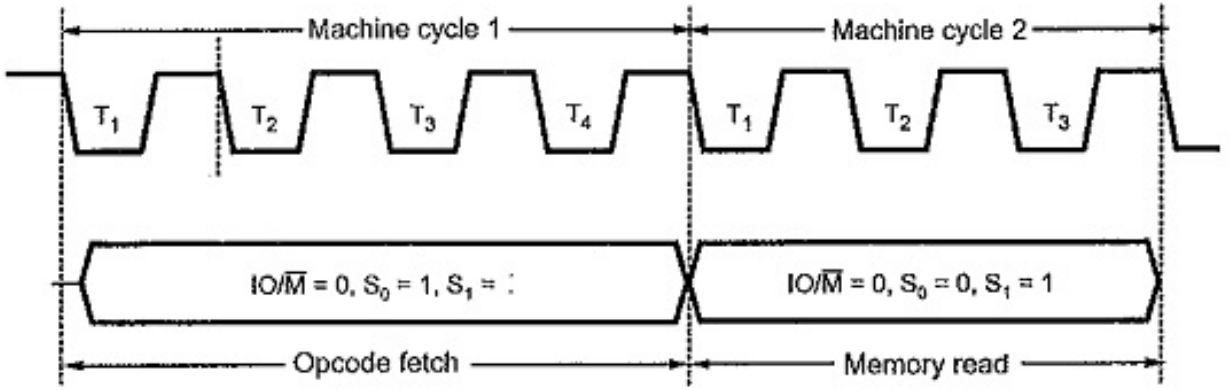


Fig. 1.13 Status signals

- 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**.

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

- These signals decide the direction of the data transfer.
- When $\overline{RD}$ signal is **active**, data is transmitted from memory or I/O device to the microprocessor.
- When $\overline{WR}$ 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 $\overline{WR}$

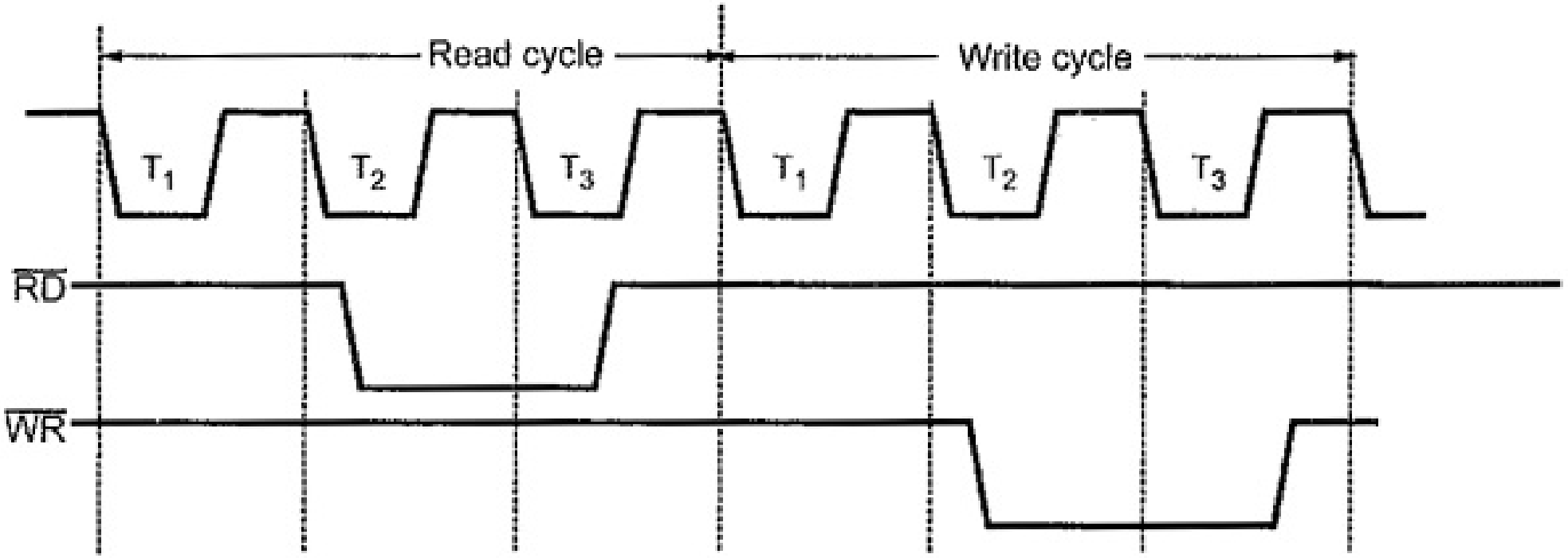


Fig. 1.14 $\overline{RD}$ and $\overline{WR}$ signals

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.

Introduction to the Timing Diagram

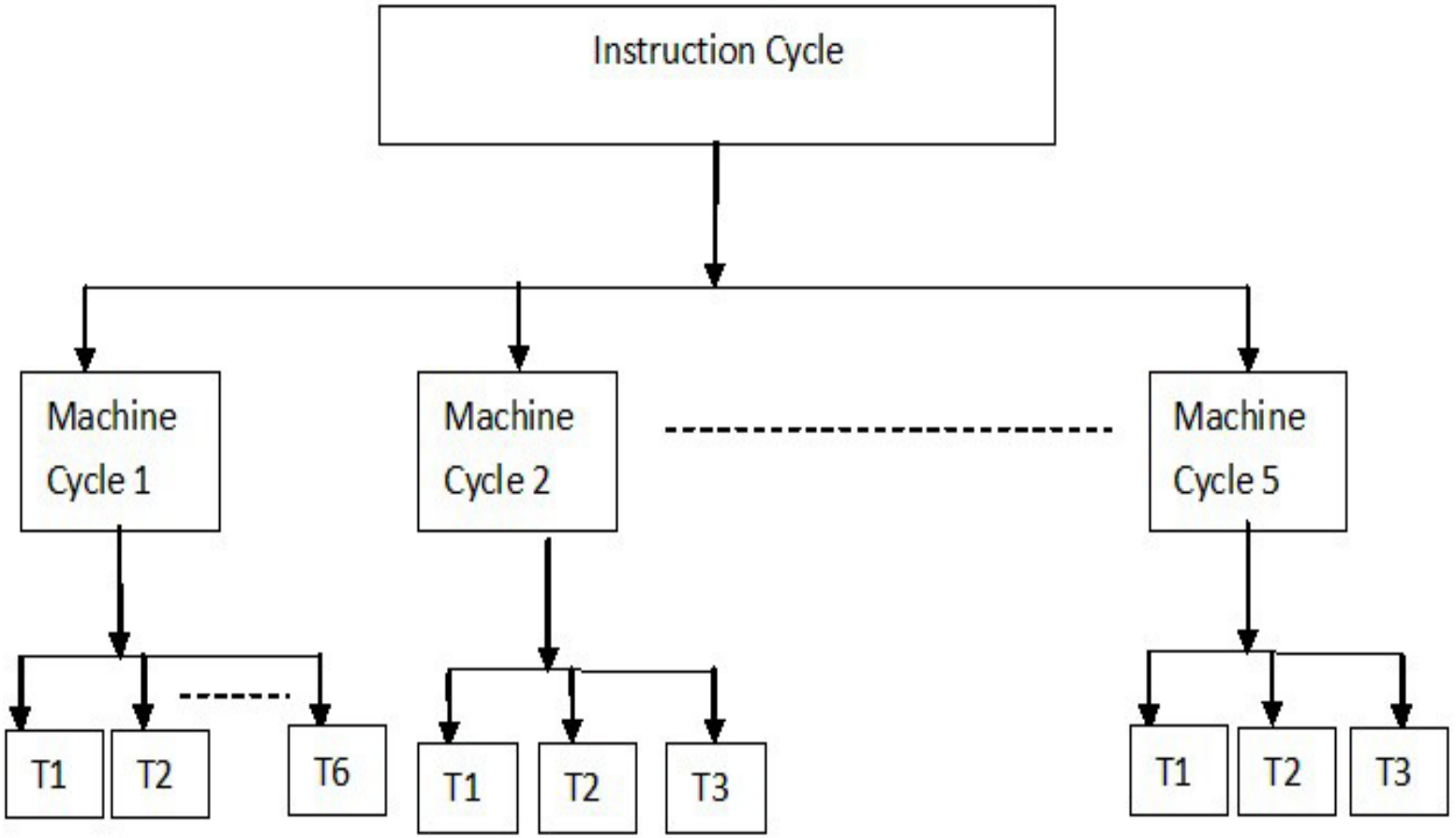
- **Timing Diagram** is a graphical representation of the instruction execution in steps with respect to the time (clock signal). It represents the execution time taken by each instruction in a graphical format. The execution time is represented in T-states. The different types of cycles used in the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

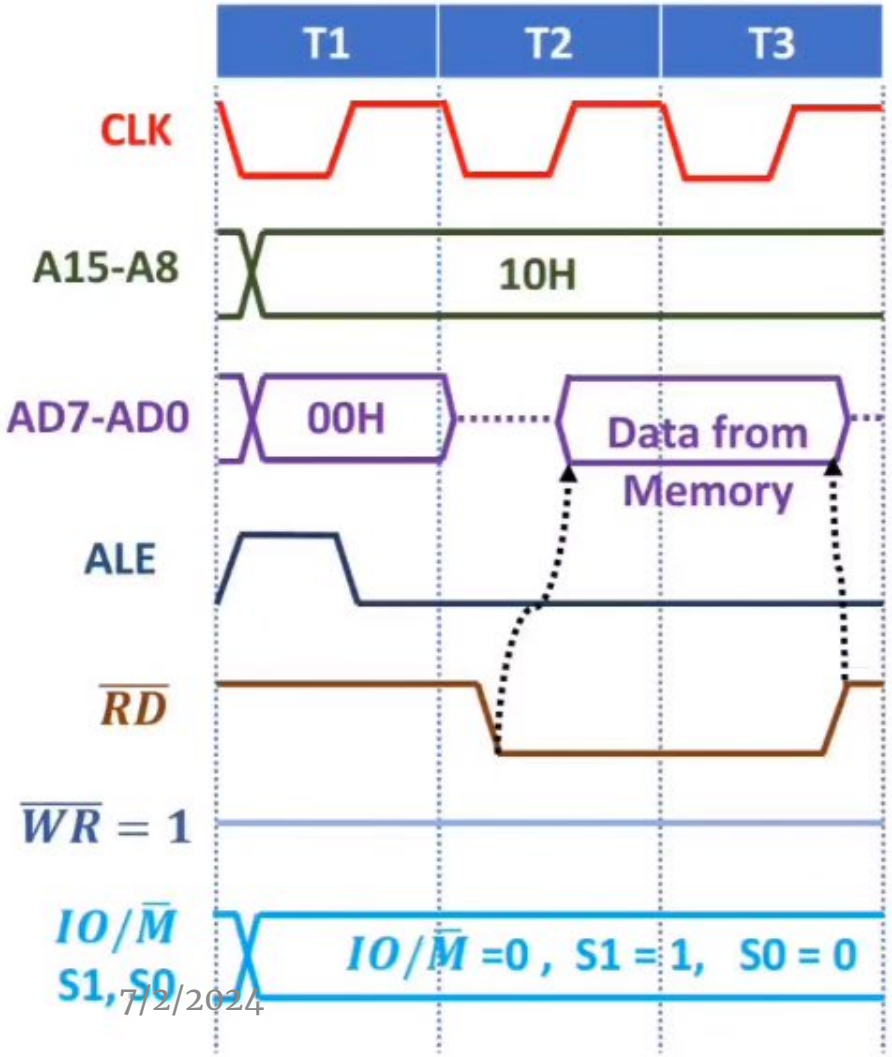
Machine Cycle	Status Signals			Control Signals		
	$IO/\overline{M}$	S1	S0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
IO Read	1	1	0	0	1	1
IO Write	1	0	1	1	0	1
INTR ACK	1	1	1	1	1	0
Bus Idle	0	0	0	1	1	1

Machine Cycles

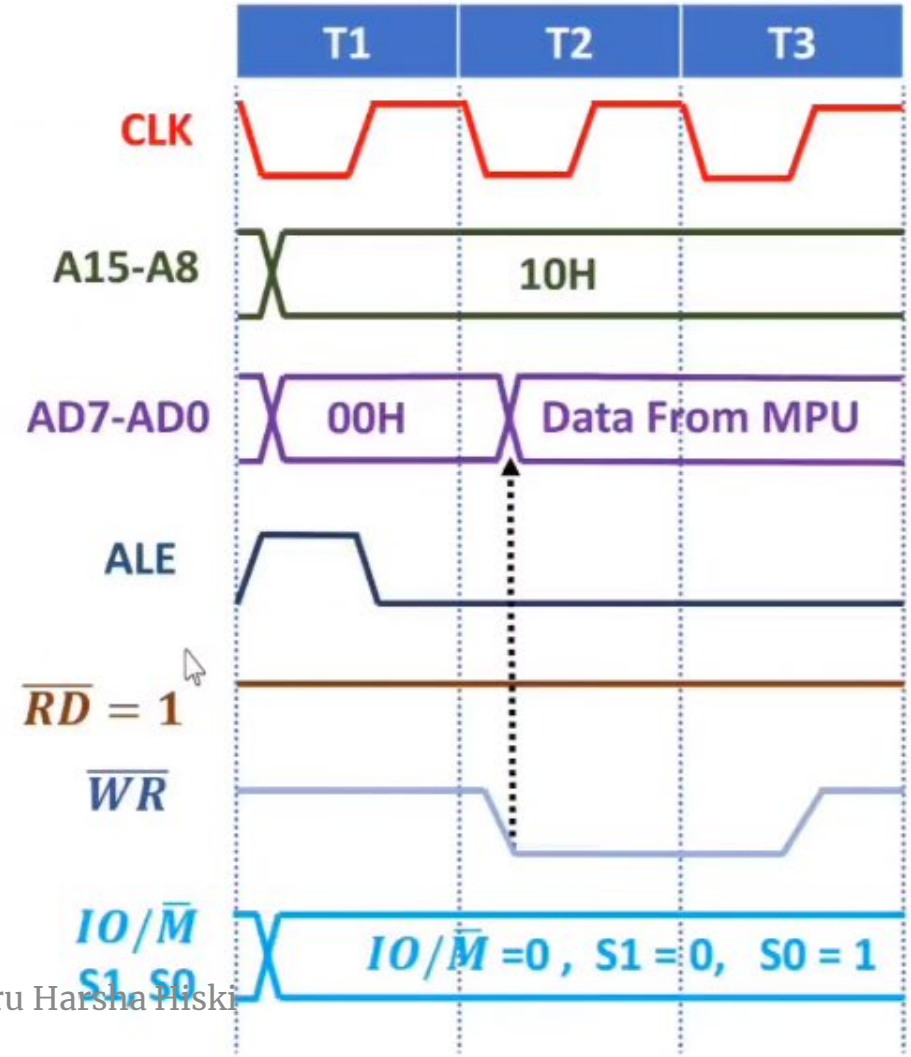
S No	Instruction	No: of machine cycles	Machine cycle - 1	Machine cycle - 2	Machine cycle - 3	Machine cycle - 4
1	**MOV A,B**	1	OF	-	-	-
2	**MVI A, 50H**	2	OF	MR	-	-
3	**LDA 5000H**	4	OF	MR	MR	MR
4	**STA 5000H**	4	OF	MR	MR	MW
5	**IN 80H**	3	OF	MR	IOR	-
6	**OUT 80H**	3	OF	MR	IOW	-

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]

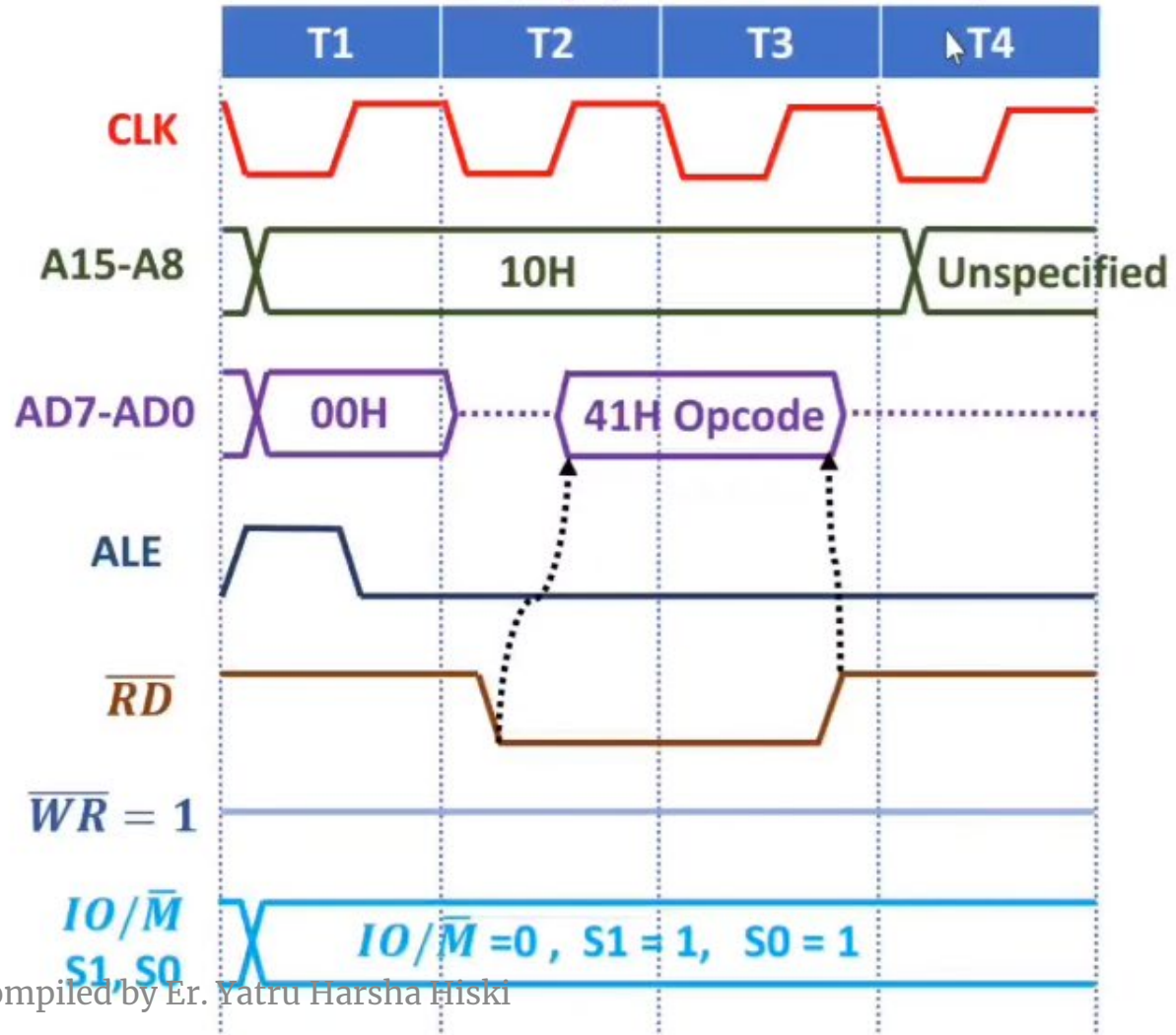
PC = 1000H
8085
C= 24H

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

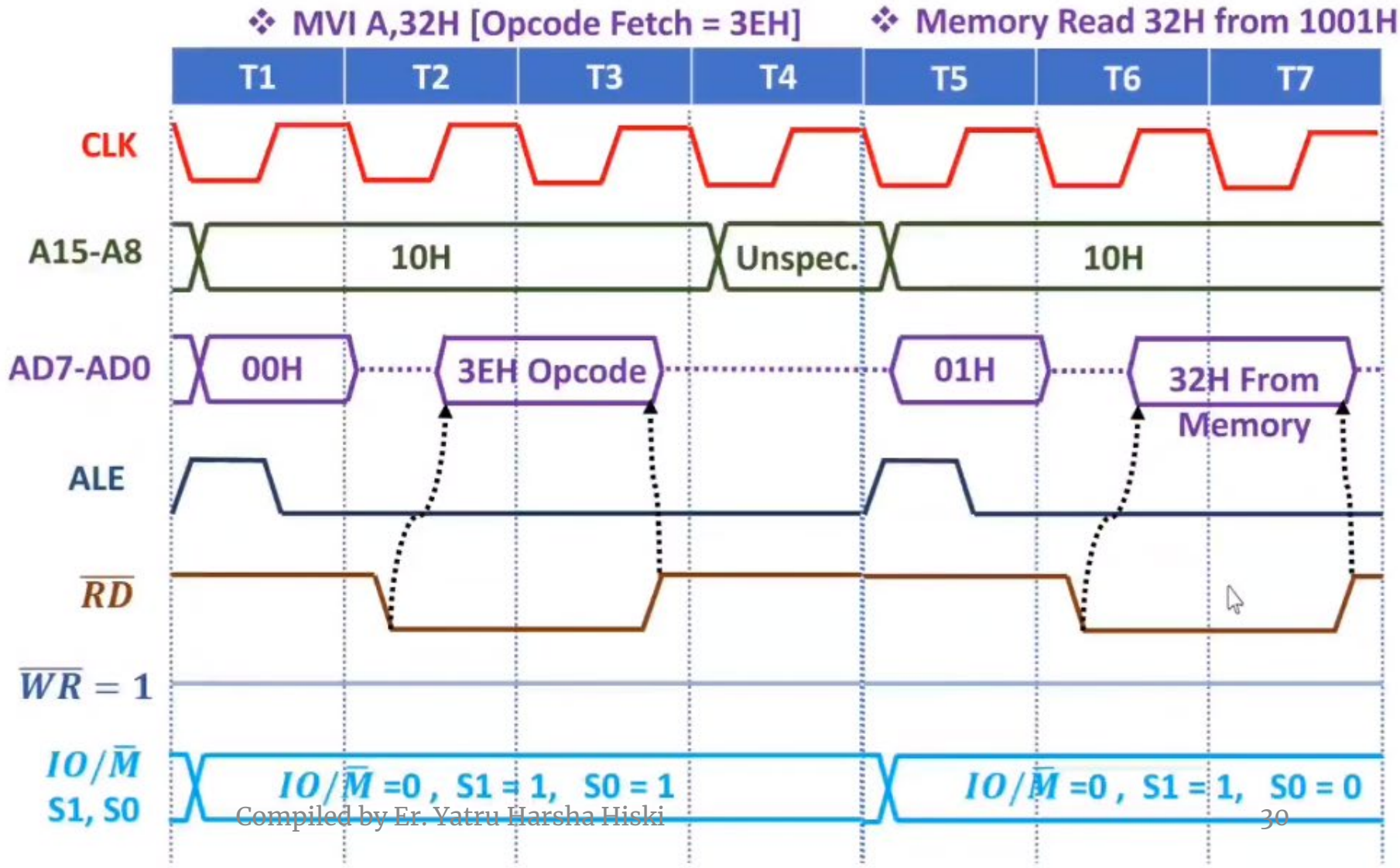
❖ MVI A,32H [Opcode =3EH]

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



Timing Diagram of LDA instruction in 8085

❖ LDA 8000H [Opcode =3AH]

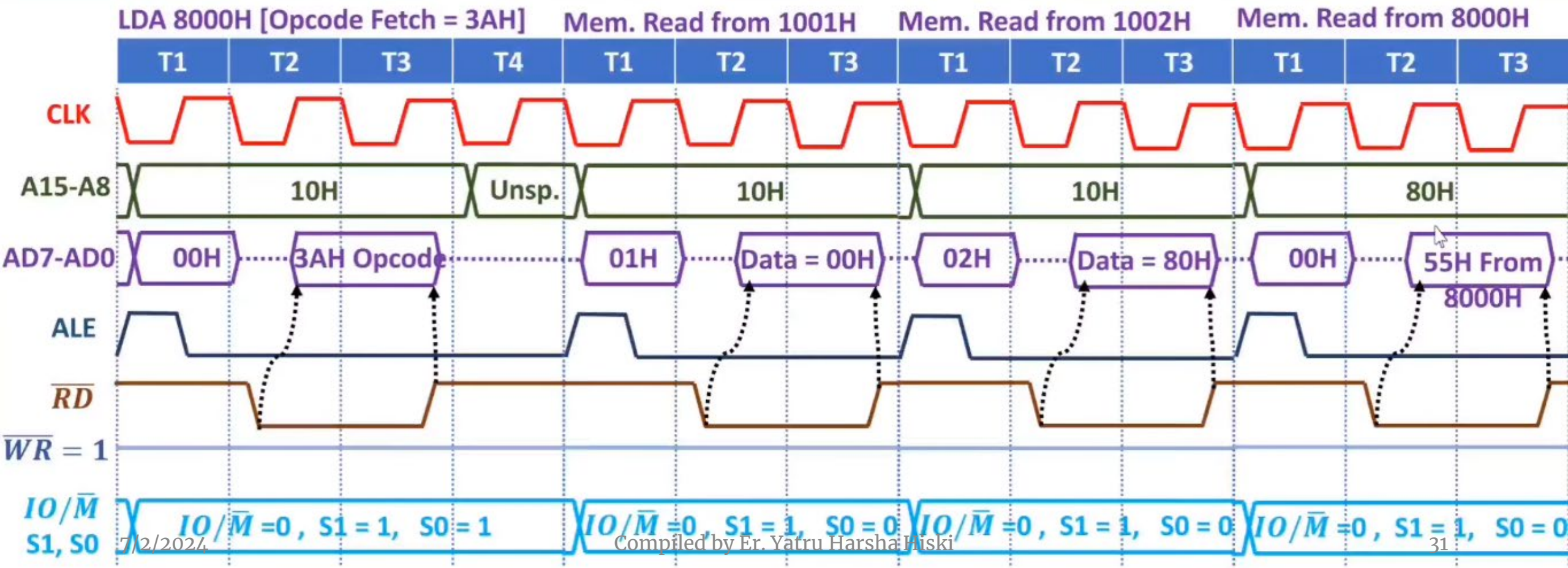
PC = 1000H
8085
A= 55H

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

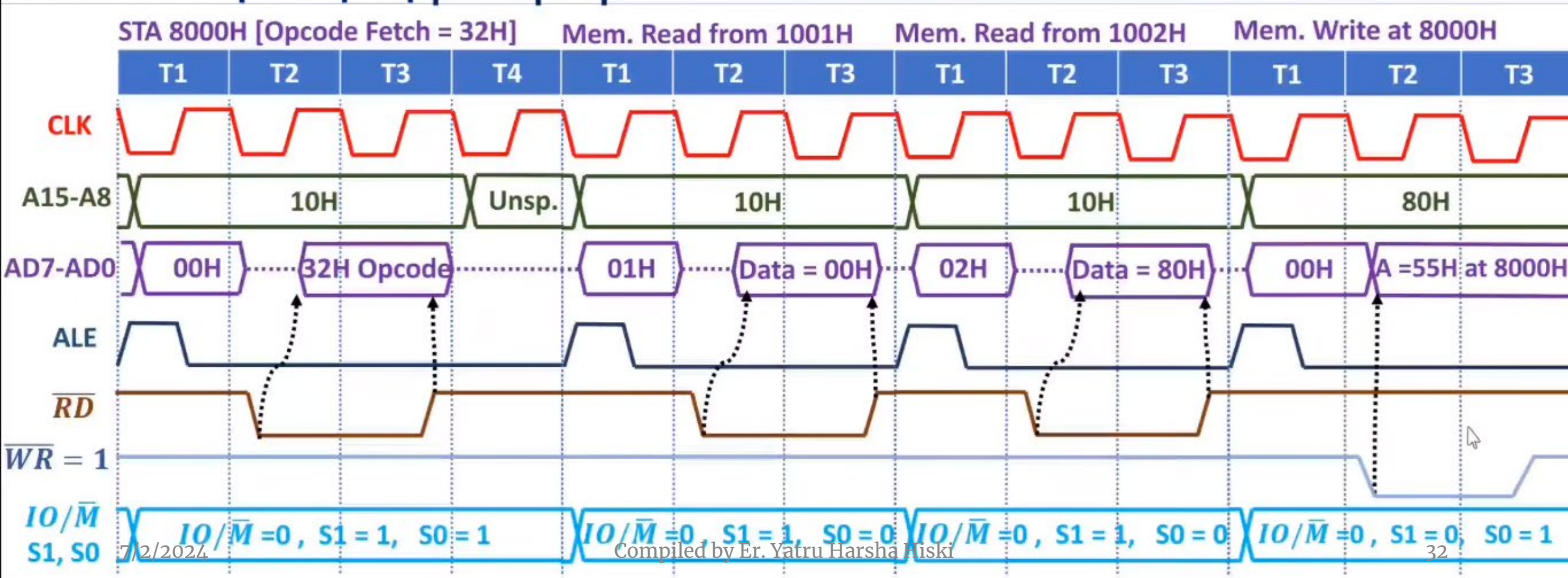
- STA 8000H [Opcode =32H]

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.



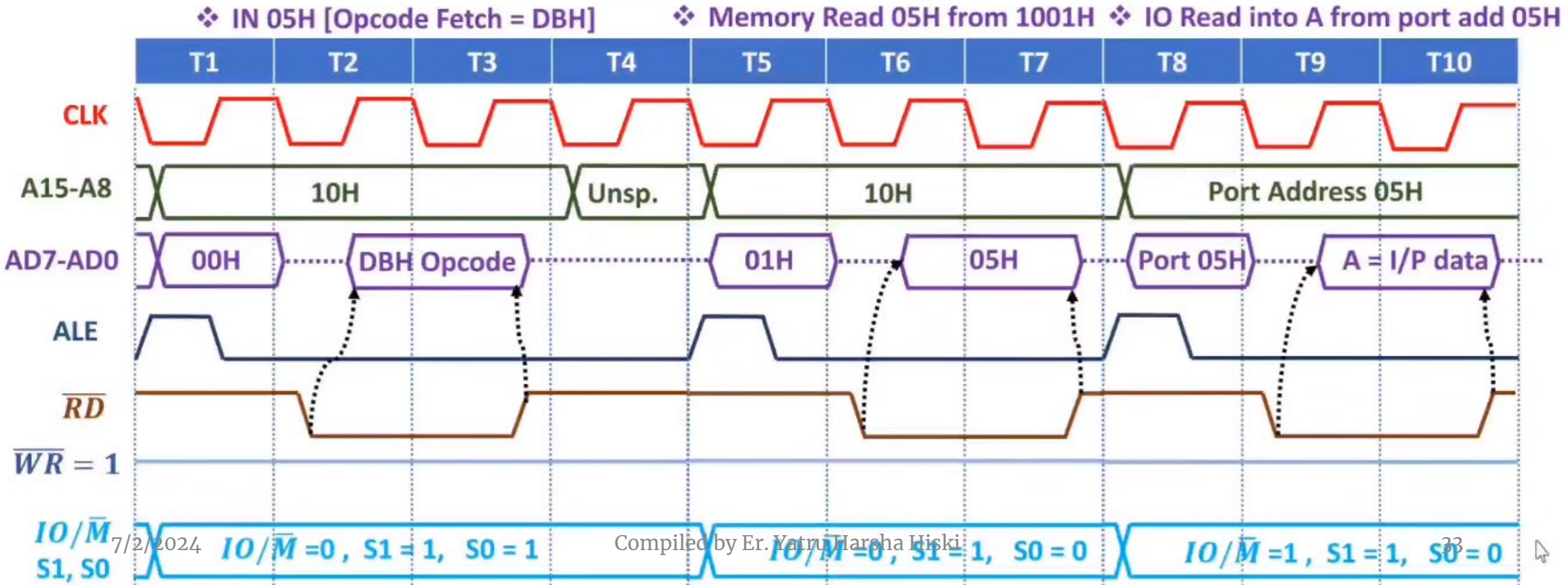
Timing Diagram of IN instruction in 8085

- 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.



Timing Diagram of OUT instruction in 8085

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

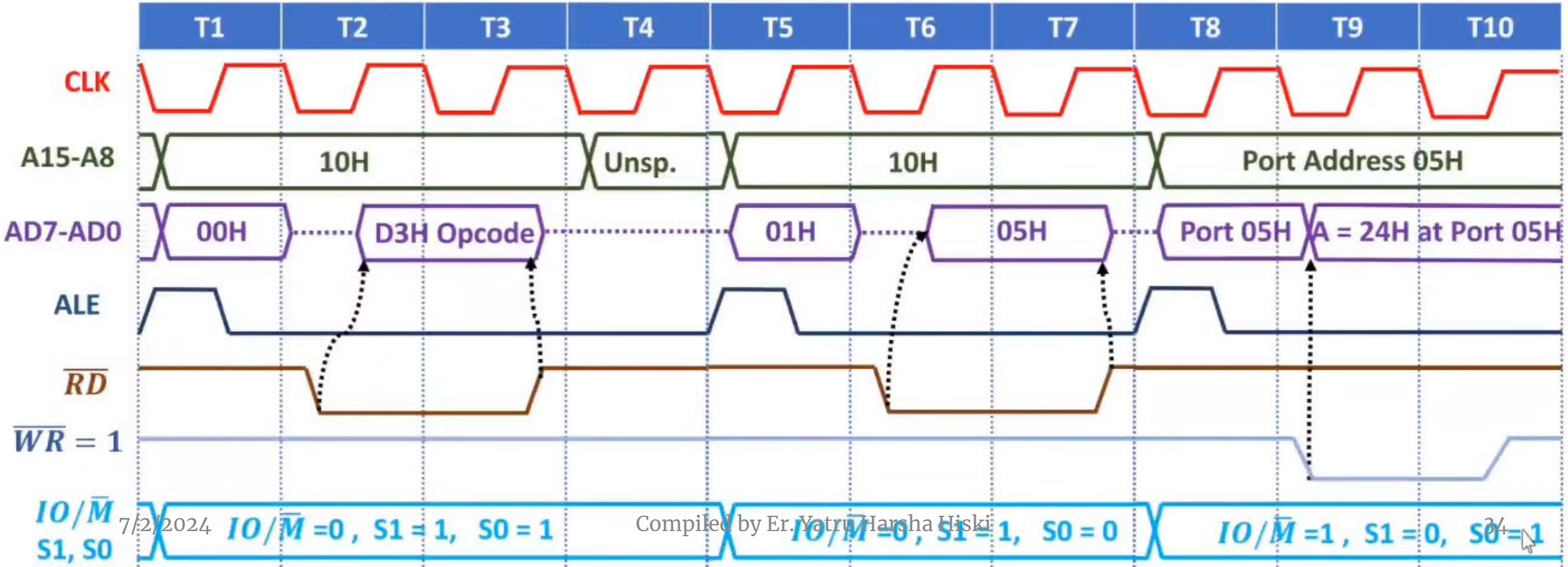
PC = 1000H
8085
A= 24H

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/$\overline{\text{M}}$**, **$\overline{\text{RD}}$**, and **$\overline{\text{WR}}$** when it wants to **read from** and **write into memory**.
- Similarly, each memory chip has signals such as **$\overline{\text{CE}}$** or **$\overline{\text{CS}}$** (**chip enable** or **chip select**), **$\overline{\text{OE}}$** or **$\overline{\text{RD}}$** (**output enable** or **read**), and **$\overline{\text{WE}}$** or **$\overline{\text{WR}}$** (**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/$\overline{M}$** , **$\overline{RD}$** and **$\overline{WR}$** signals as shown in Table 6.1.

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

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

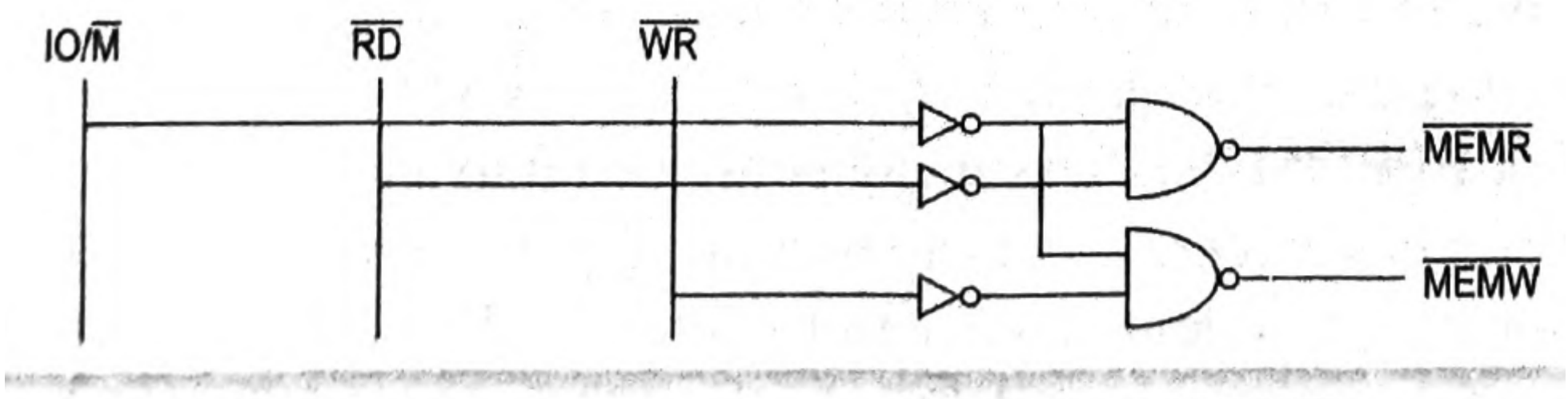


Fig. 6.1 Circuit used to generate $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ signals

Generation of Control Signals

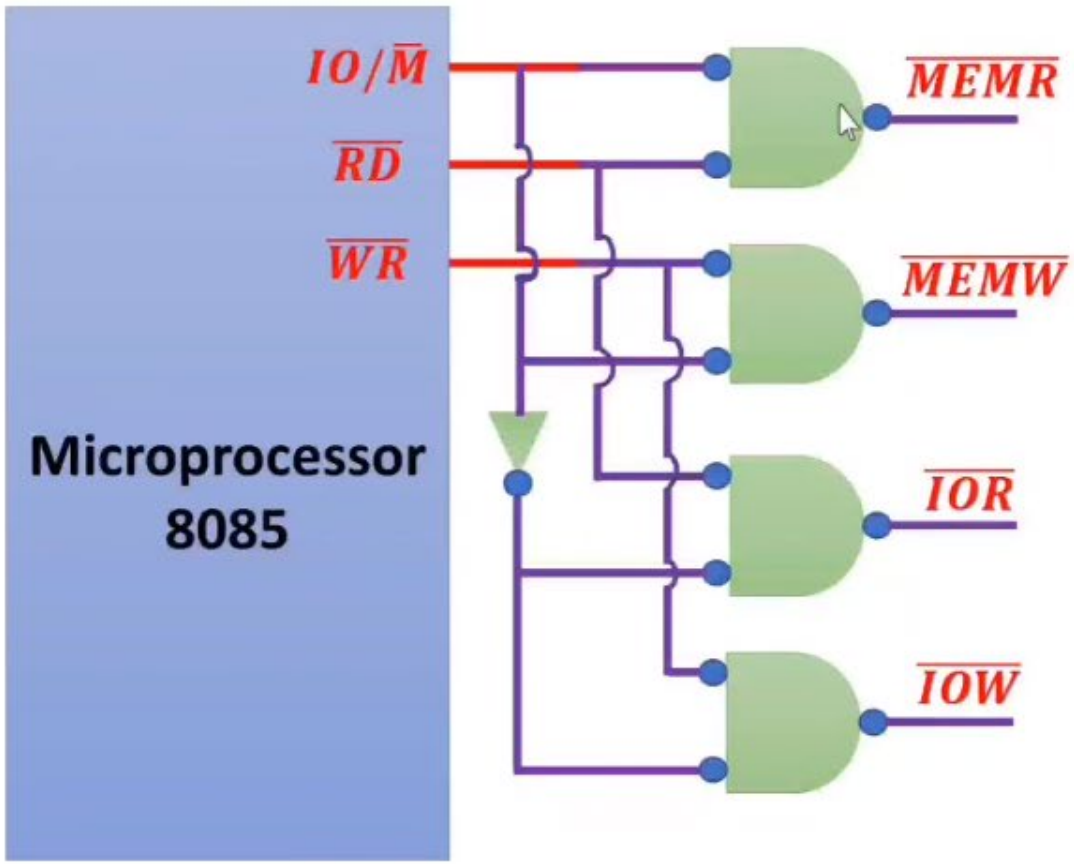
- When the **IO/$\overline{M}$** signal is high, both memory control signals are deactivated (i.e., in logic high state) irrespective of the status of the **$\overline{RD}$** and **$\overline{WR}$** signals. This is shown in the third row of Table 6.2.

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

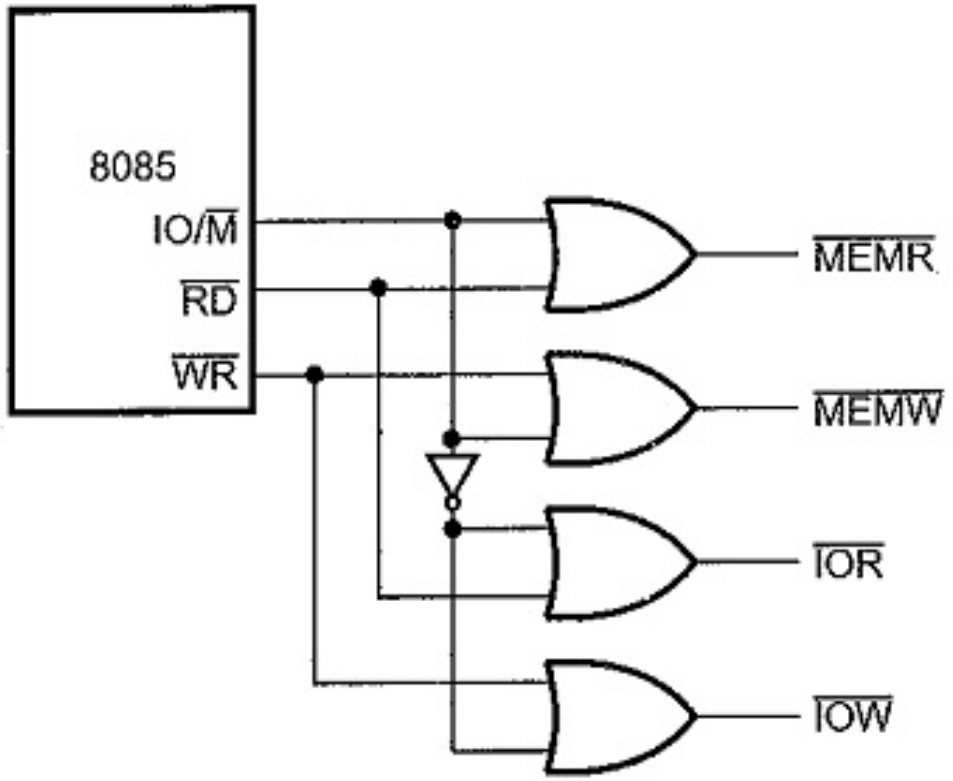


Fig. 4.8 Generation of MEMR, MEMW, IOR and IOW signals

Generation of Control Signals

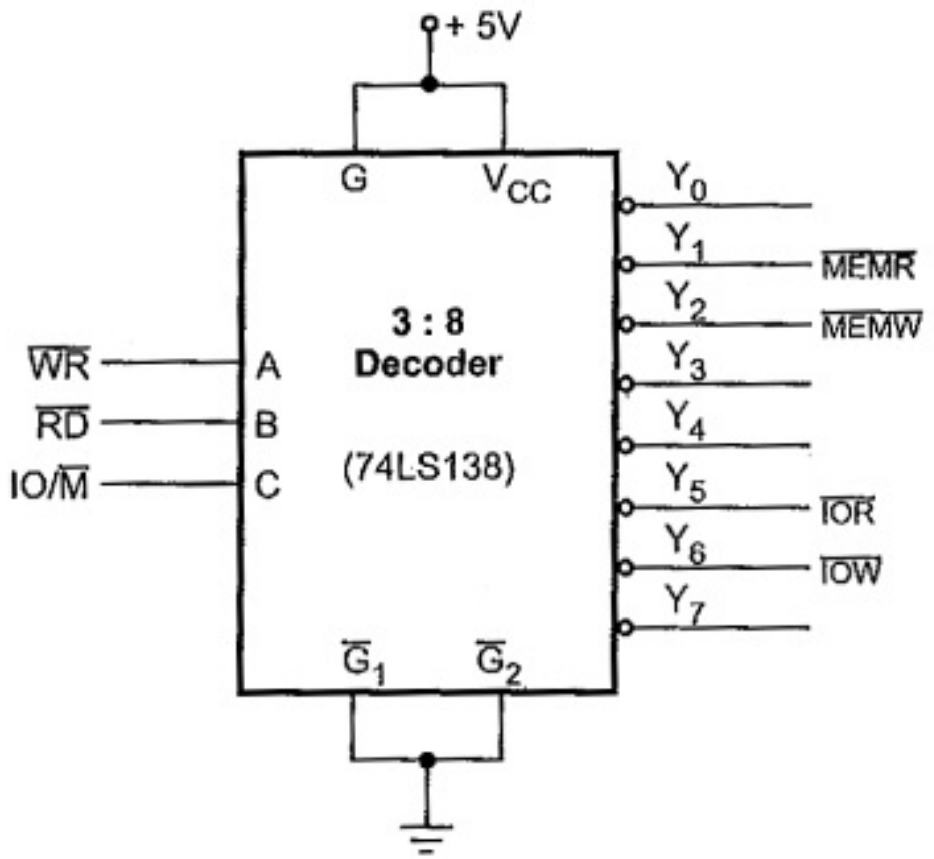
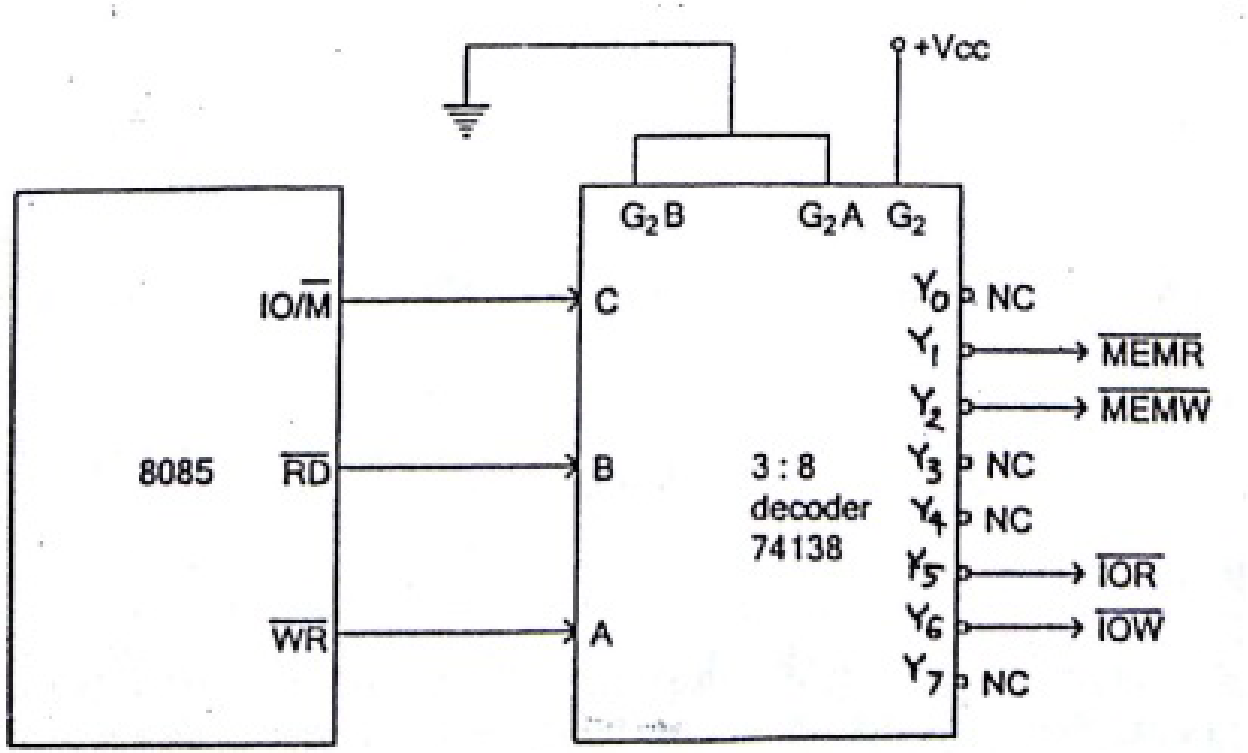


Fig. 4.9 Generation of control signals using 3 : 8 decoder

Generation of Control Signals



Address Decoding

- Consider that we have a microprocessor interfaced to both I/O device and also a memory chip. Now how to select between the two devices according to the requirement?
- For this purpose an address decoding circuit is used. An address decoding circuit aids in selecting the required I/O device or a memory chip.
- Let us discuss the concept of Memory interfacing and I/O interfacing here:
- The processor communicates with all the parts interconnected in the system through a common address and data bus. As a result of this, only one device can transmit data at a time and others can only receive that data. If more than one device attempt to send data through the bus at the same time, the proper communication among the devices does not become possible because the data sent by them gets garbled. To avoid this situation, ensuring that the proper device gets addressed at proper time, the technique called "address decoding" is used.

Address Decoding

- Generally, there are two common methods for mapping address of these devices. They are:
  - **I/O Mapped I/O:** It can address 2^8=256 bytes if mapped in I/O mode. They are used to read 8-bit data from or write 8-bit data to selected device. In this, the I/O device is addressed with 8-bit address.
  - **Memory Mapped I/O:** In this mode, the I/O devices are addressed with 16-bit address. The total addressing capability of the processor 8085 in this mode is 2^{16} = 65536 bytes = 64 KB.
- In both modes described above, depending on the addresses that are allocated to the device, the address decoding are categorized in the following two groups:
  - **Unique Address Decoding:** If all of the address lines available on that mapping mode are used for address decoding, then that decoding is called unique address decoding.
  - **Non-unique Address Decoding:** If all of the address lines available on that mapping mode are not used for address decoding, then that decoding is called non unique address decoding.

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

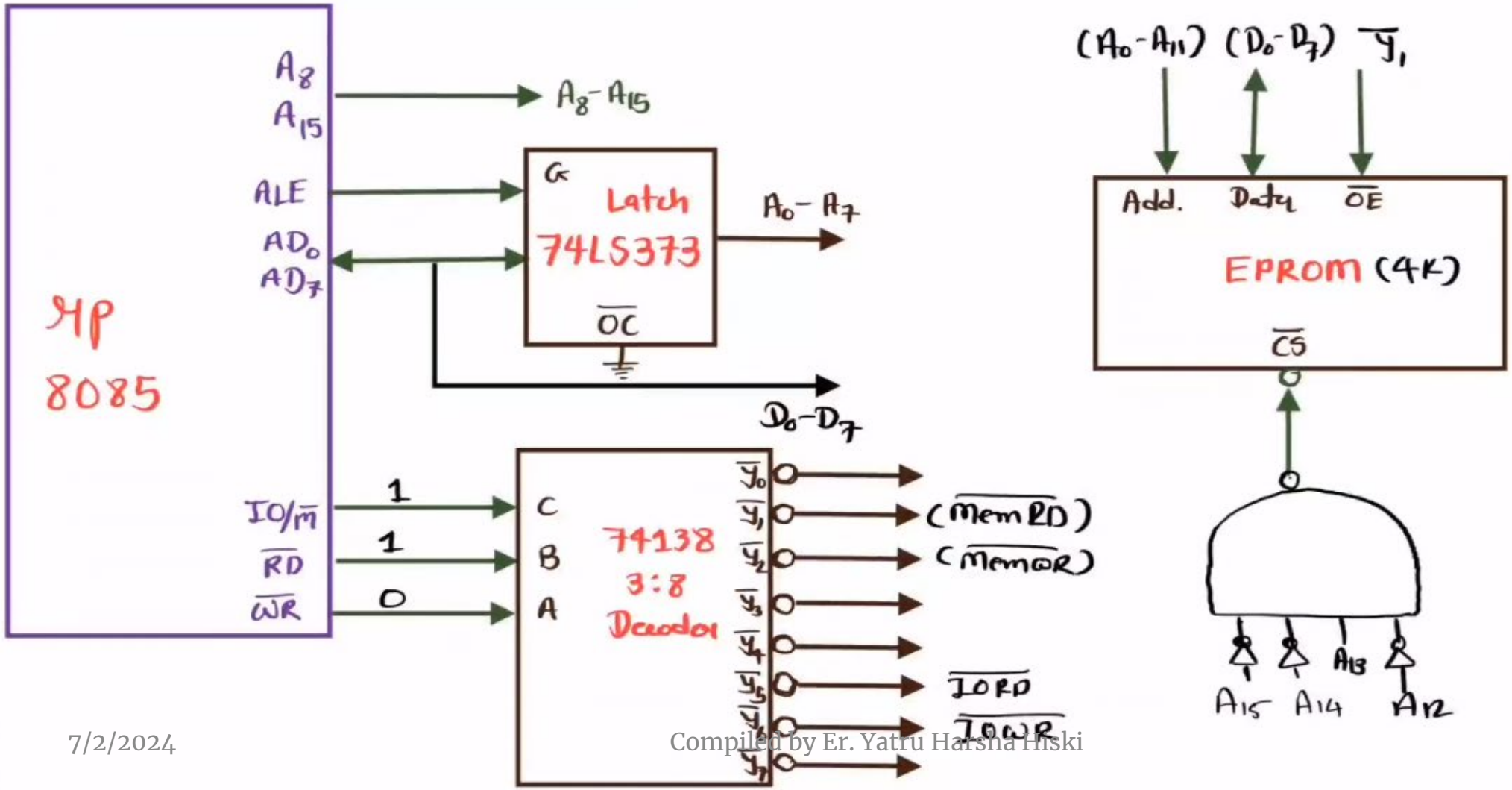
1] Address lines.
2] Data lines.
3] Control lines.
4] Chip select.

↝ 4K EPROM

- Address lines
  ⇒) $4K = 4 \times K = 2^2 \times 2^{10} = 2^{12}$
- 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.d

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

↝ For 4K EPROM

- Address lines
  ⇒ $4K = 4 \times K = 2^2 \times 2^{10} = 2^{12}$
- So total 12 Address lines are req^{d} (A_0-A_{11})
- Data lines = 8 (D_0-D_7)
- Control lines = Memory Read.

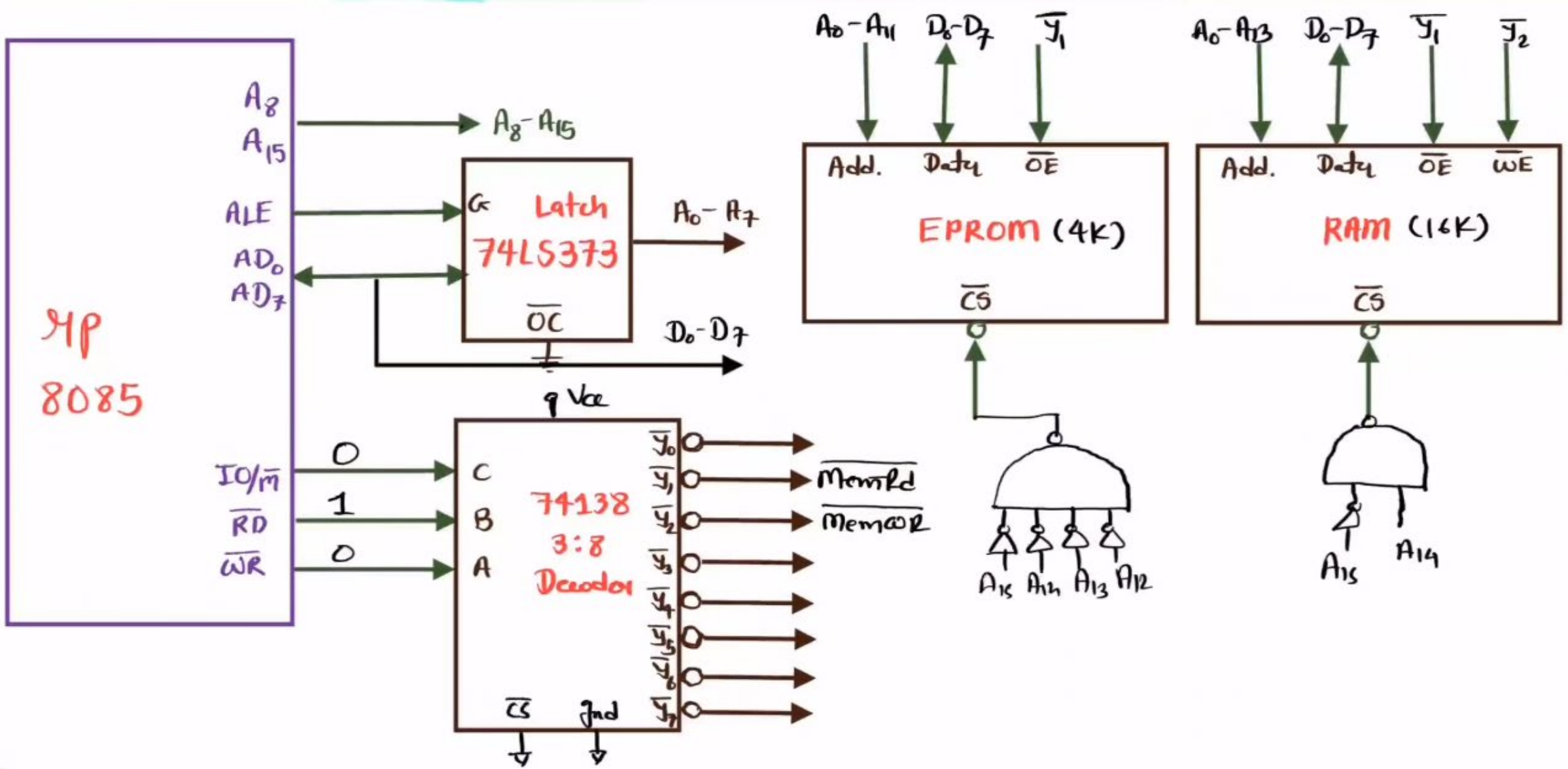
↝ 16K RAM

- Address lines
  ⇒ $16K = 2^4 \times 2^{10} = 2^{14}$
- 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


8085
A8
A15
ALE
AD0
AD7
IO/M
RD
WR
A8-A15
Latch
74LS373
G
OC
A0-A7
D0-D7
Vcc
0
1
0
C
B
A
74138
3:8
Decoder
CS
Gnd
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
MemRd
MemWR
A0-A11
D0-D7
Y1
Add.
Data
OE
EPROM (4K)
CS
A15
A14
A13
A12
A0-A13
D0-D7
Y1
Y2
Add.
Data
OE
WE
RAM (16K)
CS
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.

⇝ 8 KB 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.^{d} $[A_0 - A_{11}]$

⇝ 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 $[D_0 - D_7]$

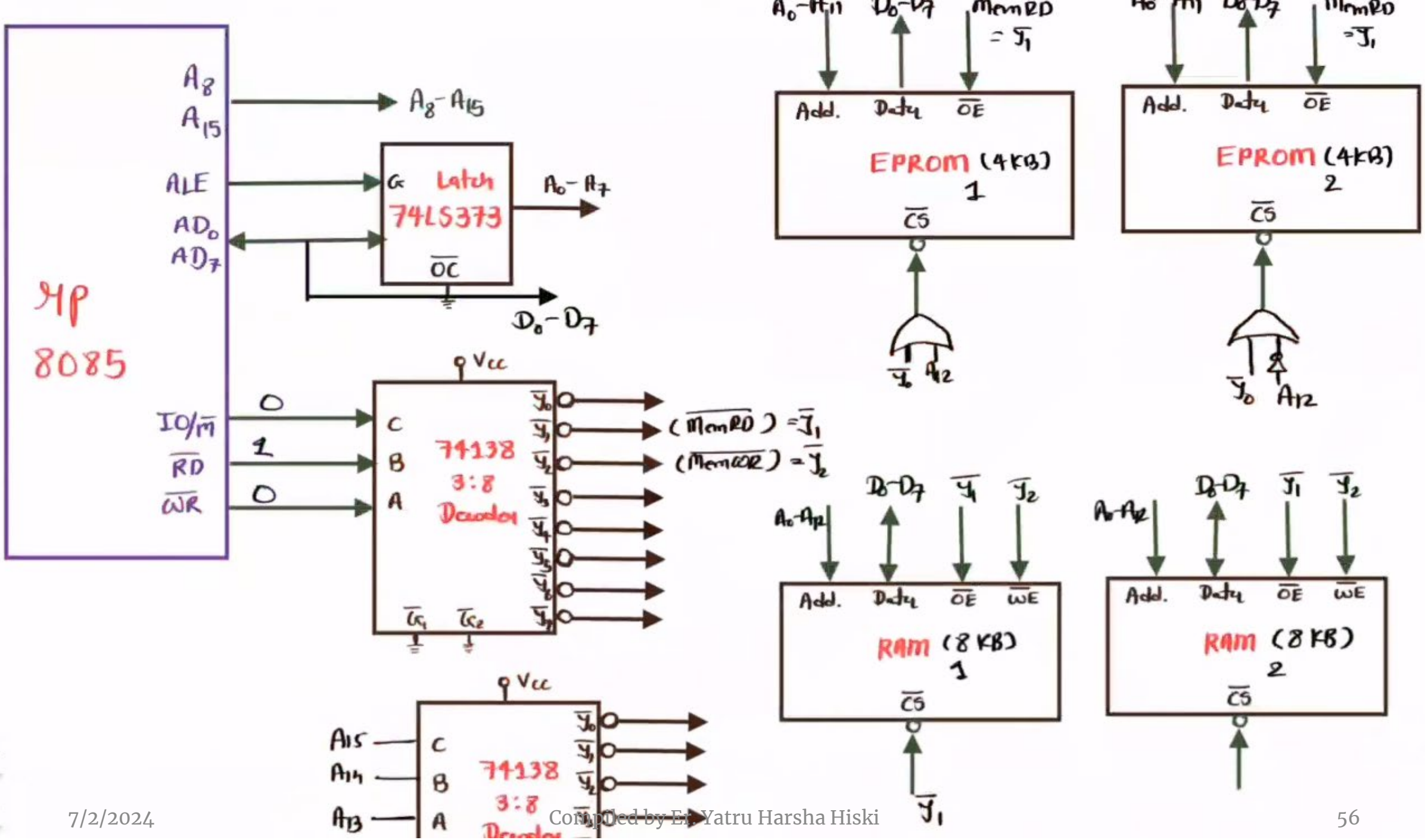
⇝ Control signal = Memory Read, Memory write.

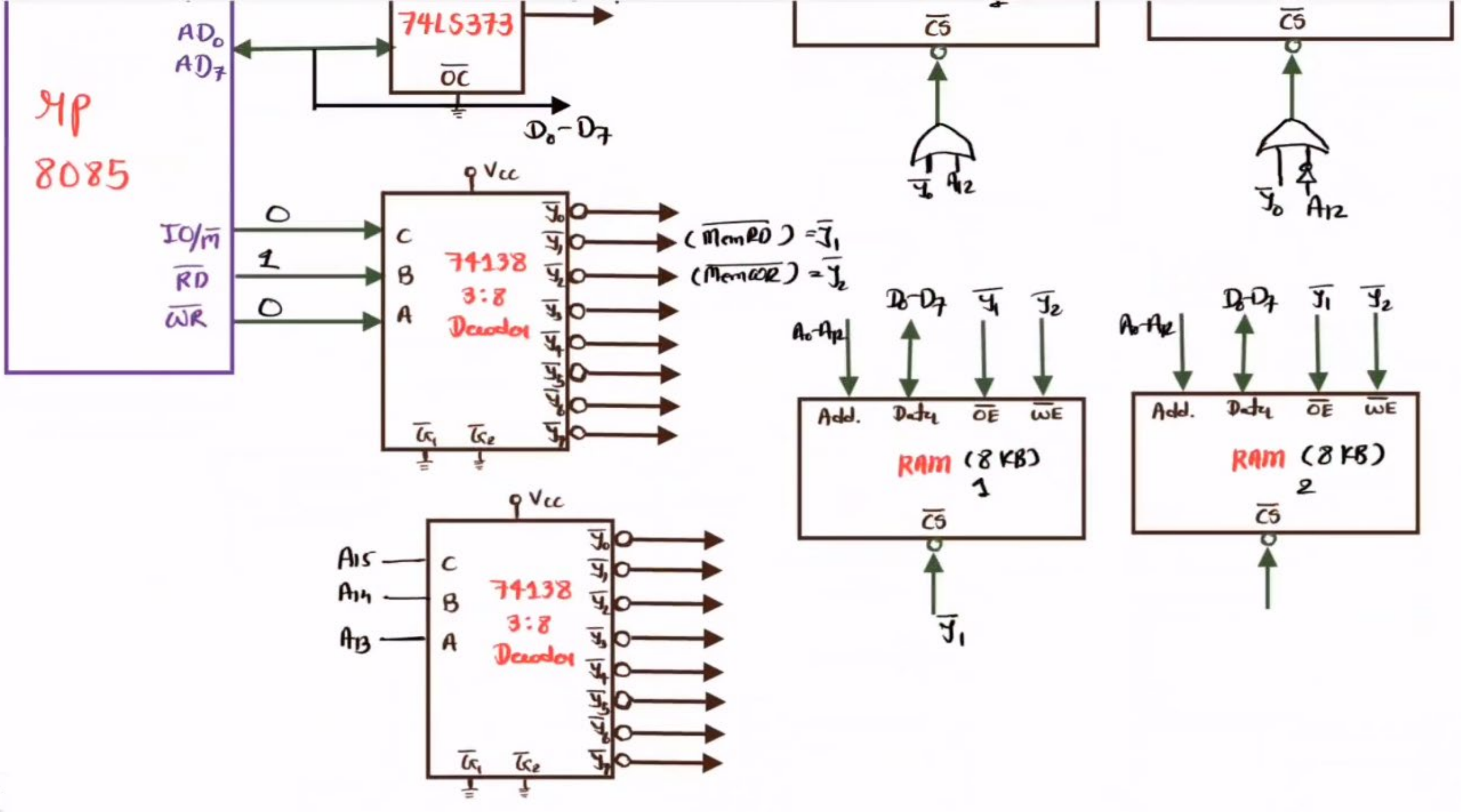
Memory Mapping

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 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 H
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF H
EPROM 4KB 2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000 H
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF H
RAM 8KB 1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 H
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF H
RAM 8KB 2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 H
	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 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 H
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF H
EPROM 4KB 2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000 H
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF H
RAM 8KB 1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 H
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF H
RAM 8KB 2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 H
	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF H


8085
A_8
A_{15}
ALE
AD_0
AD_7
IO/M
RD
WR
A_8-A_{15}
G Latch 74LS373 OC
A_0-A_7
D_0-D_7
Vcc
C
B
A
74138 3:8 Decoder
$\overline{G_1}$ $\overline{G_{2A}}$ $\overline{G_{2B}}$
$\overline{Y_0}$ $\overline{Y_1}$ $\overline{Y_2}$ $\overline{Y_3}$ $\overline{Y_4}$ $\overline{Y_5}$ $\overline{Y_6}$ $\overline{Y_7}$
$(\overline{MemRD}) = \overline{Y_1}$
$(\overline{MemWR}) = \overline{Y_2}$
A_{15}
A_{14}
A_{13}
A_0-A_{11}
D_0-D_7
$\overline{MemRD} = \overline{Y_1}$
Add. Data $\overline{OE}$
EPROM (4KB) 1
$\overline{CS}$
$\overline{Y_0}$ A_{12}
EPROM (4KB) 2
A_0-A_{12}
$\overline{WE}$
RAM (8KB) 1
RAM (8KB) 2


AD0
AD7
74LS373
OC
D0-D7
8085
IO/M
RD
WR
0
1
0
C
B
A
74138
3:8
Decoder
Vcc
G1
G2A
G2B
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
(MemRD) = Y1
(MemWR) = Y2
A15
A14
A13
A0-A12
D0-D7
Add.
Data
OE
WE
RAM (8 KB)
1
2
CS

END OF UNIT 3

THANK YOU