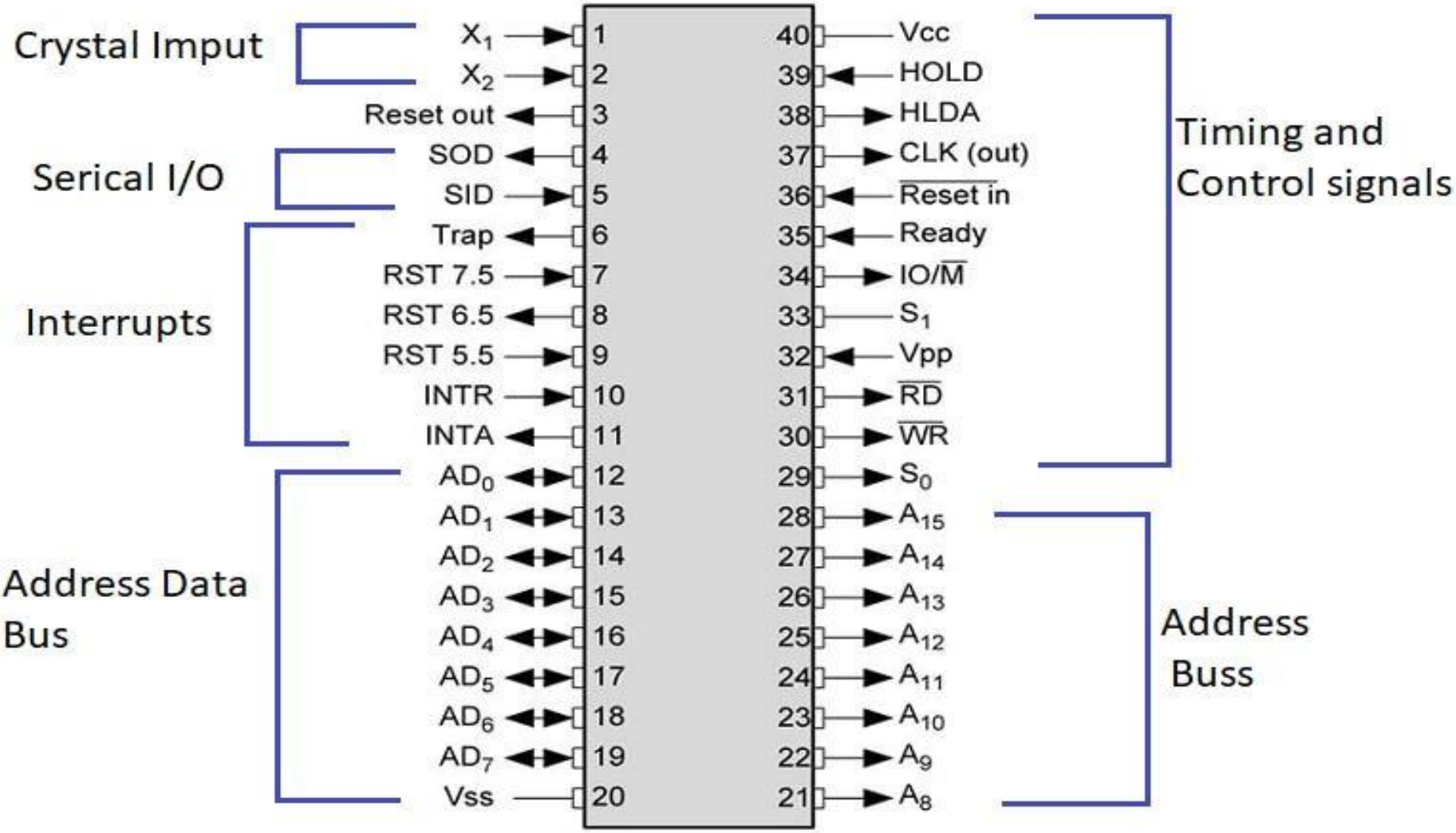
Chapter-3
8085 Microprocessor Architecture

Syllabus Content

- 8085 Microprocessor Architecture
- Address, Data And Control Buses
- 8085 Pin Functions
- Demultiplexing of Buses
- Generation Of Control Signals
- Instruction Cycle, Machine Cycles, T-States
- Memory Interfacing

8085 Pin Diagram



8085 Pin Functions

Address Bus, Data Bus & Control Bus

What is a bus ? : A set of pins, wires or signals having common function is called as **bus.**

What is system bus ? : A bus is a bundle of wires that are grouped together to serve a single purpose in the 8085 microprocessor there are three sets of communication lines that are called buses. They are the address bus, the data bus and control bus. The three buses together form the "**system bus**".

- Fig. 1.11.1 shows the 8085 bus structure.

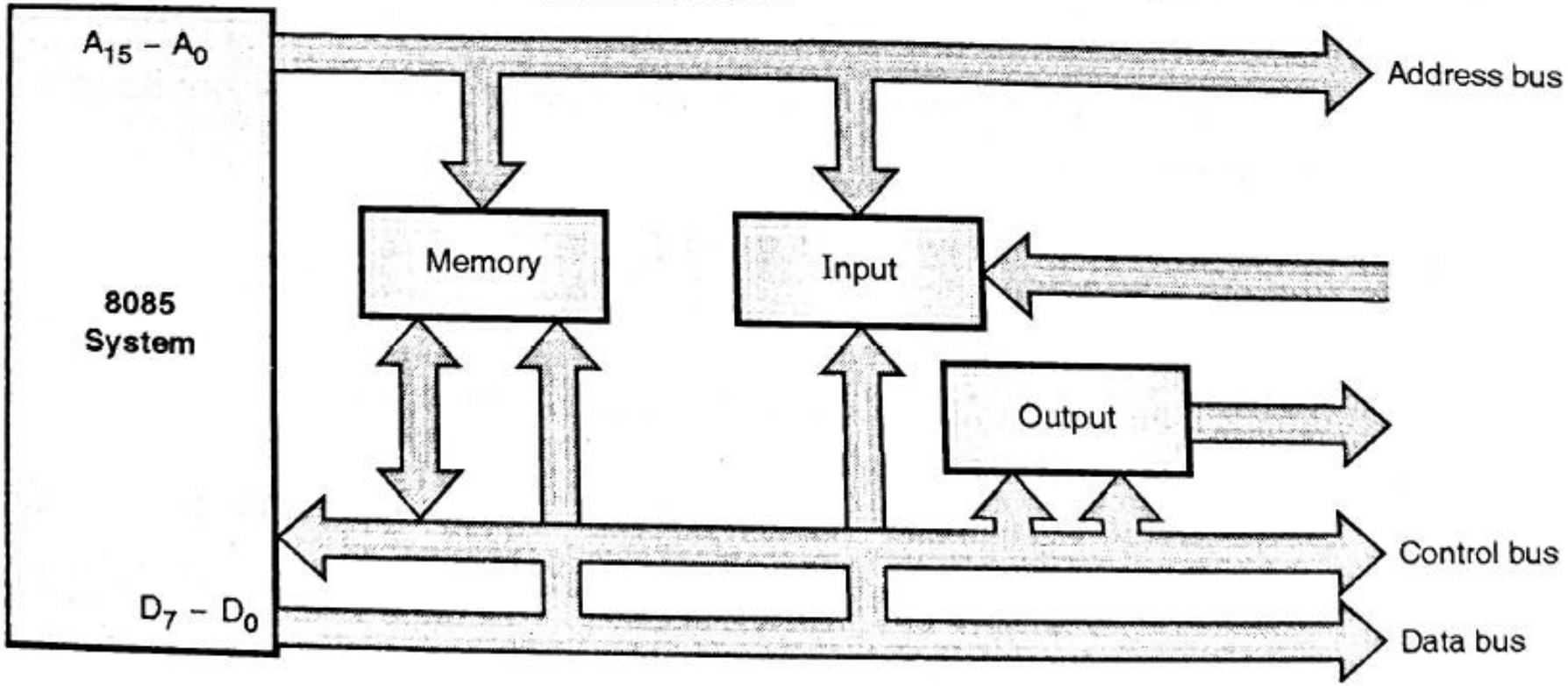


Fig. 1.11.1 : The 8085 bus structure

1.11.1 Address Bus

GTU : June 2010

Function of address bus : The bus over which the microprocessor sends out the address of a memory location or I/O location is called as the **address bus**.

- The address bus carries the address of the memory or I/O location to be read or written from.

Size of address bus : In 8085 the address bus is 16-bit ($A_0 - A_{15}$). So the microprocessor can be used to access 16 bit address and is capable of addressing $2^{16} = 65536$ i.e. 64 K memory locations.

- The address bus is unidirectional i.e. bits flow only in one direction from the microprocessor unit to memory and I/O devices.
- The address bus is also used to send the port address on the address bus. When the microprocessor reads data from or writes data to a port, it sends the port address out on the address bus.

1.11.2 Data Bus

GTU : June 2010

Size of data bus : The data bus of 8085 consists of 8 parallel lines $D_0 - D_7$.

- The data bus is a bi-directional bus. This means the data can be transferred from CPU to memory or I/O locations and viceversa.

- The number of data lines used in the data bus is equal to the size of data word being written or read.
- The data bus also connects the I/O ports and microprocessor. So the microprocessor can write data to or read data from the memory or the I/O ports.

1.11.3 Control Bus

- The 8085 microprocessor uses the control bus to provide the timing signals.
- The microprocessor sends signals on the control bus to enable the outputs of addressed memory devices or I/O port devices.
- Some of the control bus signals are follows :
  (1) Memory read. (2) Memory write.
  (3) I/O read. (4) I/O write.
- These signals are used to identify a device type with which the microprocessor intends to communicate.

Demultiplexing of Buses

- 74LS373 Latch IC is used to latch the address issued by 8085.
- It has 8 latches (D-flip flops)
- The clock pin of these flip flops are connected together and available at the output as a pin called Enable (G).

Operation :

(1) Address will appear on $AD_0 - AD_7$ lines.

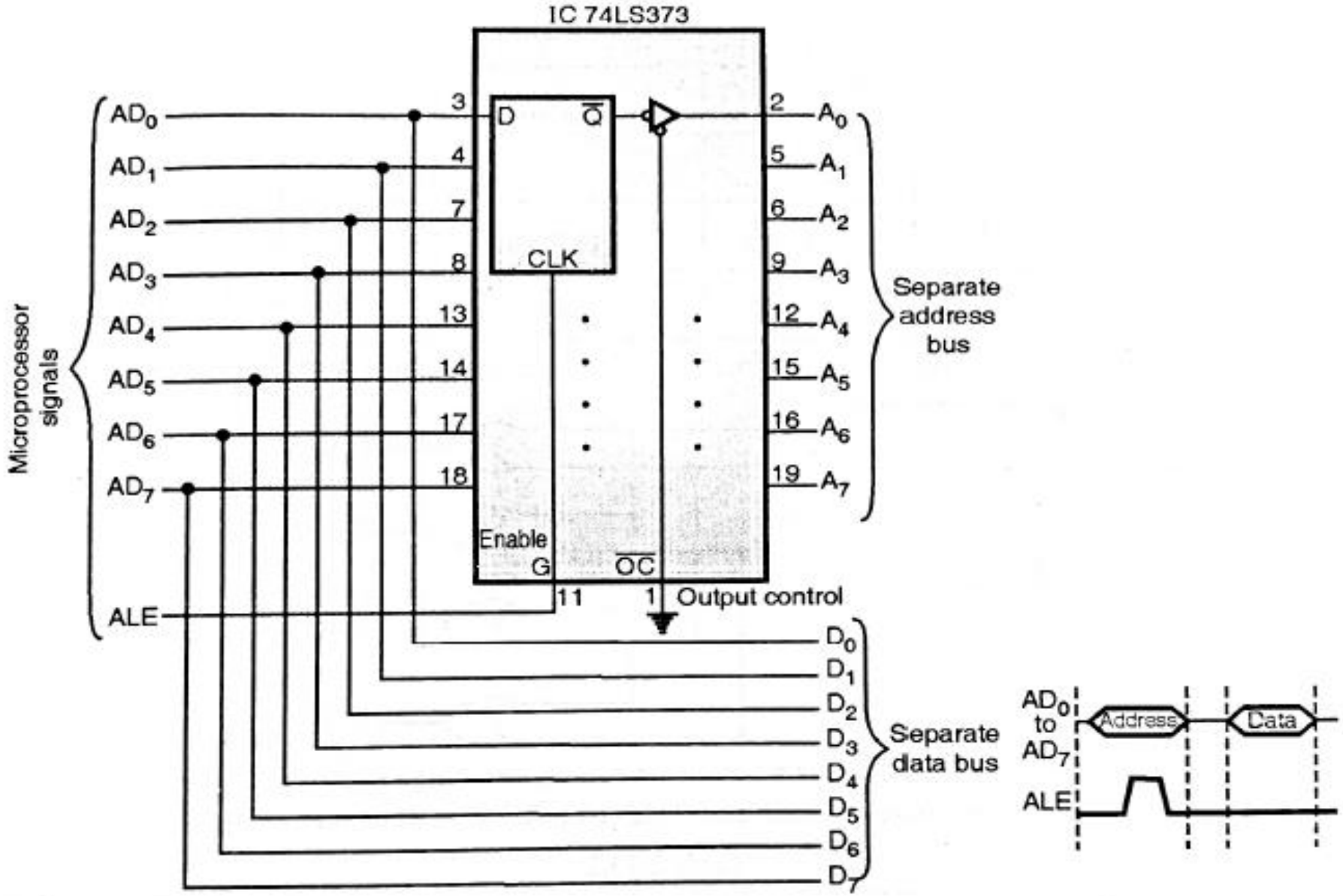
(2) ALE will go high, forcing Enable (G Pin of 74LS373) = 1. This will make 74LS373 latch *"transparent"* i.e. whatever will be input, will be output. Presently input is address ($A_7 - A_0$), therefore output is $A_7 - A_0$.

(3) Before address disappears, ALE = Enable (G) = 0. Due to this address will get latched. Even though input changes, output will not change. Thus till next cycle i.e. unless next ALE appears, status is LATCHED and will not change.

(4) $AD_0 - AD_7$, will now be used as data bus, $\therefore$ labeled as D_0 to D_7, only.

From $A_{15} - A_8$, we have PURE (not multiplexed) higher order address bus. Thus, output of latch ($A_7 - A_0$) and upper order address, in total will provide us 16 bit address ($A_{15} - A_0$).

Thus at this stage we are ready with $A_{15} - A_0$ address bus and $D_7 - D_0$ data bus.



(a) Demultiplexing address/data

(b) Waveforms

Generation of Control Signals

- In 8085 microprocessor, control signals are not directly available for reading and writing data from memory or I/O device.
- But it contains IO/ $\overline{M}$ to differentiate between memory and I/O device, $\overline{RD}$ to read data and $\overline{WR}$ to write data. By using these lines we can generate **four separate control** signals read/write for memory and read/write for I/O device.

They are,

i) $\overline{MEMR}$ (Memory Read) : To read data from memory.

ii) $\overline{MEMW}$ (Memory write) : To write data in memory.

iii) $\overline{IOR}$ (I/O Read) : To read data from I/O device.

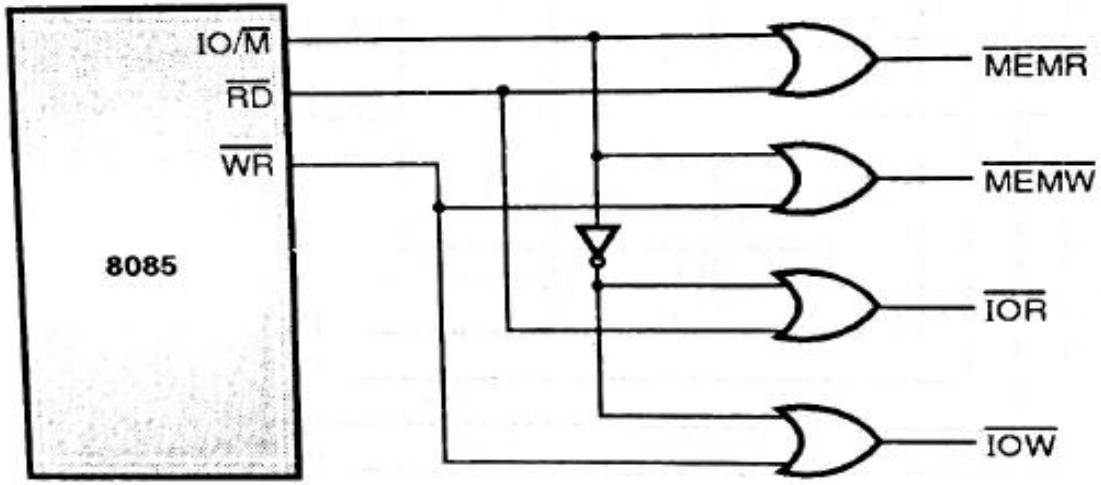
iv) $\overline{IOW}$ (I/O Write) : To write data in I/O device.

The truth table will be as shown in Table 1.13.1.

Table 1.13.1 : Truth table

IO/$\overline{M}$	$\overline{RD}$	$\overline{WR}$	Control signal
0	0	1	Memory read ($\overline{MEMR}$)
0	1	0	Memory write ($\overline{MEMW}$)
1	0	1	I/O Read ($\overline{IOR}$)
1	1	0	I/O write ($\overline{IOW}$)

From the truth table one can directly draw a multifunction logic as shown in Fig. 1.13.1.



m(1.18)**Fig. 1.13.1 : Generation of control signals**

But instead of logic gates, we can use 74LS138 (3:8 line decoder).

Table 1.15.2

IO/$\overline{M}$ (C)	$\overline{RD}$ (B)	$\overline{WR}$ (A)	Active output line	
0	0	0	0	→ Impossible condition
0	0	1	1	→ $\overline{MEMR}$
0	1	0	2	→ $\overline{MEMW}$
0	1	1	3	→ No operation
1	0	0	4	→ Impossible condition
1	0	1	5	→ $\overline{IOR}$
1	1	0	6	→ $\overline{IOW}$
1	1	1	7	→ No operation

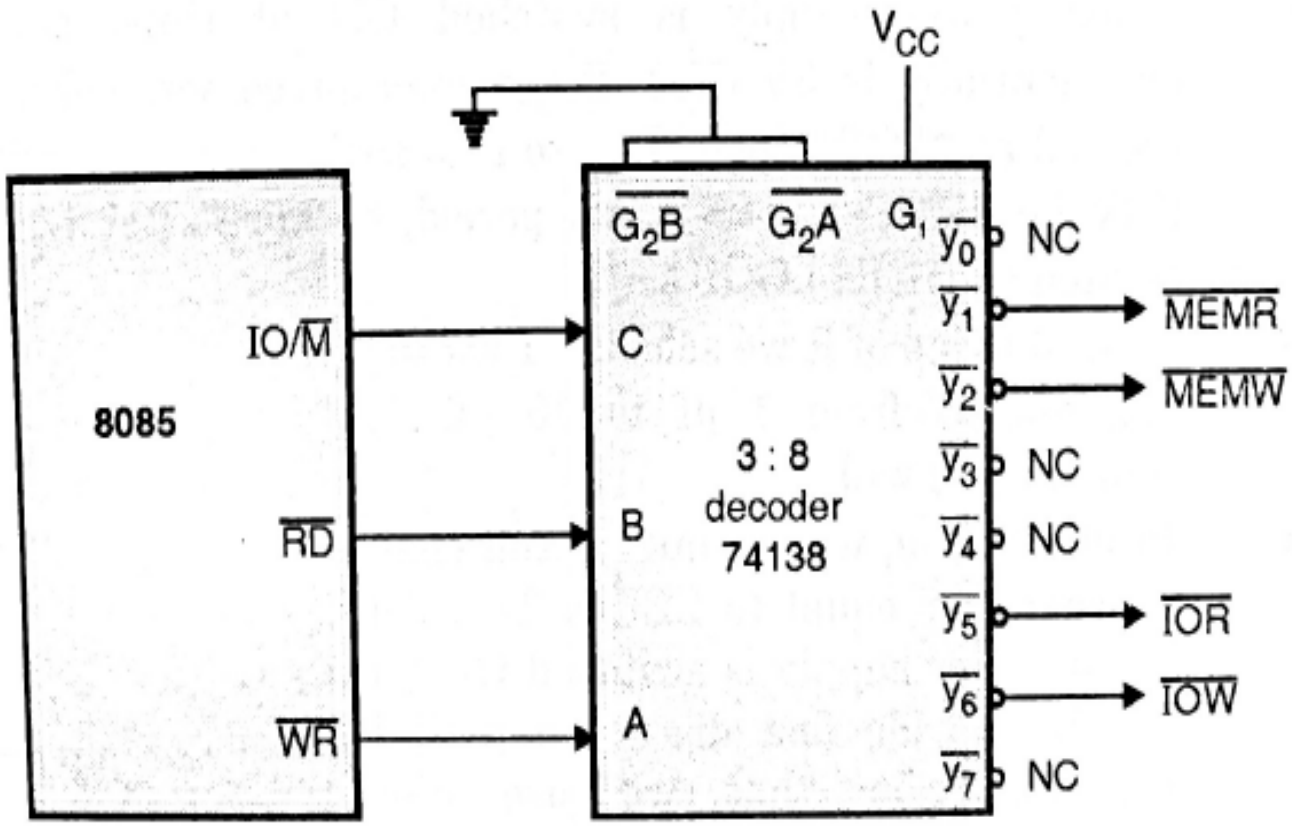
(i) When the combination on IO/$\overline{M}$ = 0, $\overline{RD}$ = 0 and $\overline{WR}$ = 1. It represents that it is a $\overline{MEMR}$ control signal. For the above combination 1st output line is active so 1st output line of 3 : 8 decoder can be used as $\overline{MEMR}$ signal.

(ii) IO/$\overline{M}$ = 0, $\overline{RD}$ = 1 and $\overline{WR}$ = 0, 2nd output line is active. It is a $\overline{MEMW}$ signal.

(iii) IO/$\overline{M}$ = 1, $\overline{RD}$ = 0 and $\overline{WR}$ = 1, 5th output line is active. It is a $\overline{IOR}$ signal.

(iv) IO/$\overline{M}$ = 1, $\overline{RD}$ = 1 and $\overline{WR}$ = 0, 6th output line is active. It is a $\overline{IOW}$ signal.

The above 4 points specify that, if we use IO/ $\overline{M}$, $\overline{RD}$ and $\overline{WR}$ as C, B, A inputs to 3 : 8 decoder respectively, the output lines 1, 2, 5 and 6 can be used as 4 control signals. There is no problem of inversion also as 74138 outputs are inverted. The 3 : 8 decoder and its connections are as shown in Fig. 1.13.2.



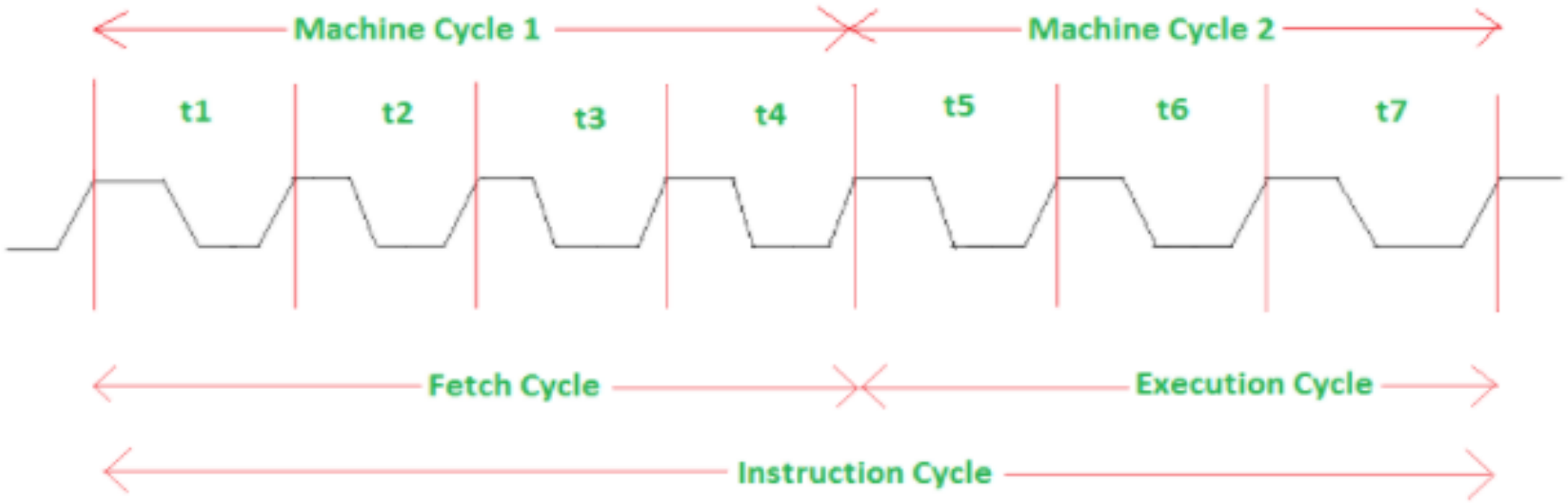
m(1.19)**Fig. 1.13.2 : Generation of control signals using decoder**

Instruction Cycle, Machine Cycle & T-State

Time required to execute and fetch an entire instruction is called *instruction cycle*. It consists:

- **Fetch cycle –** The next instruction is fetched by the address stored in program counter (PC) and then stored in the instruction register.
- **Decode instruction –** Decoder interprets the encoded instruction from instruction register.
- **Reading effective address –** The address given in instruction is read from main memory and required data is fetched. The effective address depends on direct addressing mode or indirect addressing mode.
- **Execution cycle –** consists memory read (MR), memory write (MW), input output read (IOR) and input output write (IOW).

The time required by the microprocessor to complete an operation of accessing memory or input/output devices is called *machine cycle*. One time period of frequency of microprocessor is called *t-state*. A t-state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse. Fetch cycle takes four t-states and execution cycle takes three t-states.



Instruction cycle in 8085 microprocessor

Machine Cycle in 8085 Microprocessor

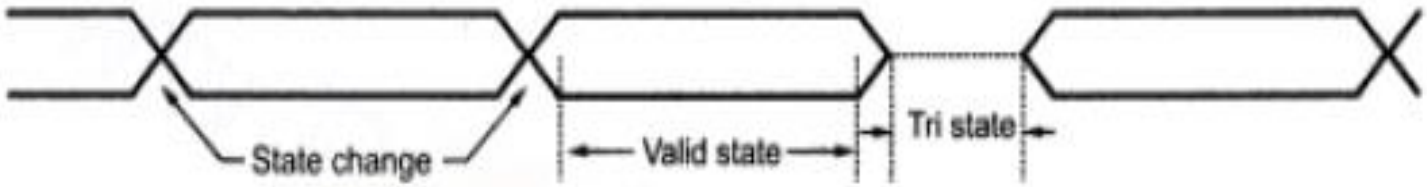
The seven Machine Cycle in 8085 Microprocessor are :

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

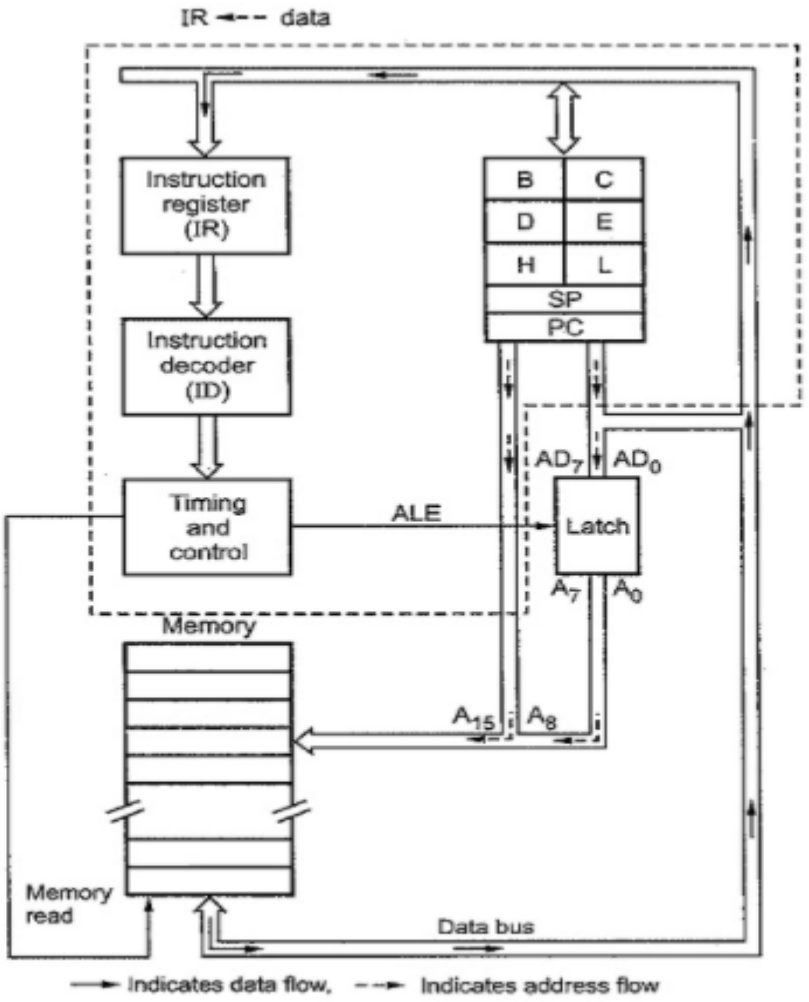
Machine Cycle	Status			No. of Machine cycles	Control
	IO/$\overline{M}$	S1	S0		
Opcode Fetch	0	1	1	4	$\overline{RD}$ =0
Memory Read	0	1	0	3	$\overline{RD}$ =0
Memory Write	0	0	1	3	$\overline{WR}$ =0
I/O Read	1	1	0	3	$\overline{RD}$ =0
I/O Write	1	0	1	3	$\overline{WR}$ =0
INTR Acknowledge	1	1	1	3	$\overline{INTA}$ =0
Bus Idle	0	0	0	3	—

(a) Ideal

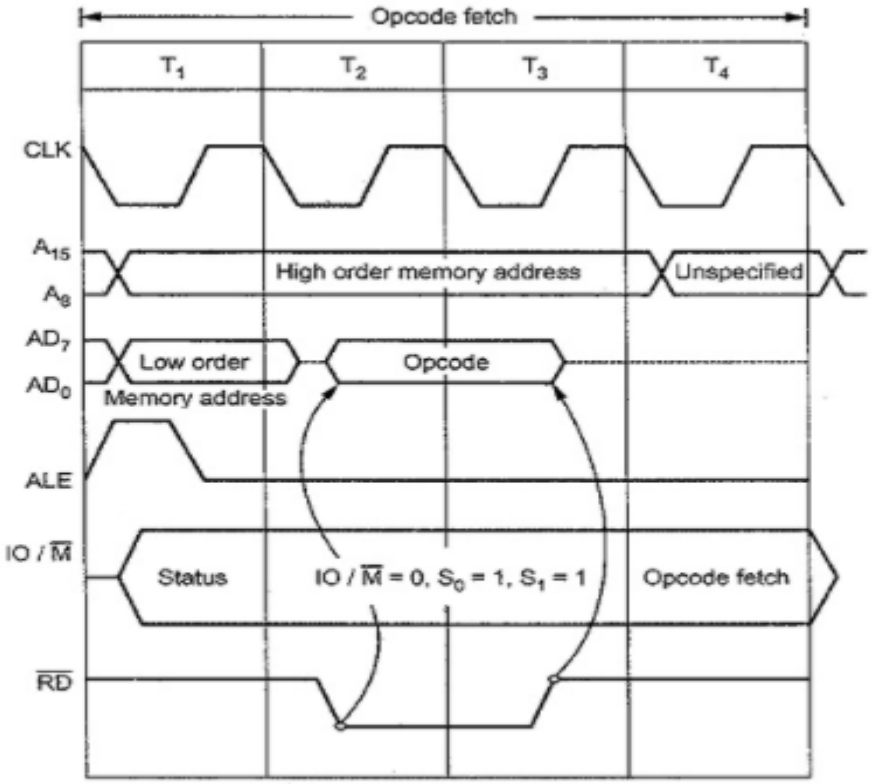
(b) Practical



Opcode Fetch Cycle



(a) Data (opcode) flow from memory to micrprocessor



(b) Opcode fetch machine 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. (a) shows flow of data (opcode) from memory to the microprocessor and Fig.(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. The following section describes the opcode fetch cycle in step by step manner.

Step 1 : (State T_1):

- 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)

- 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).

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.

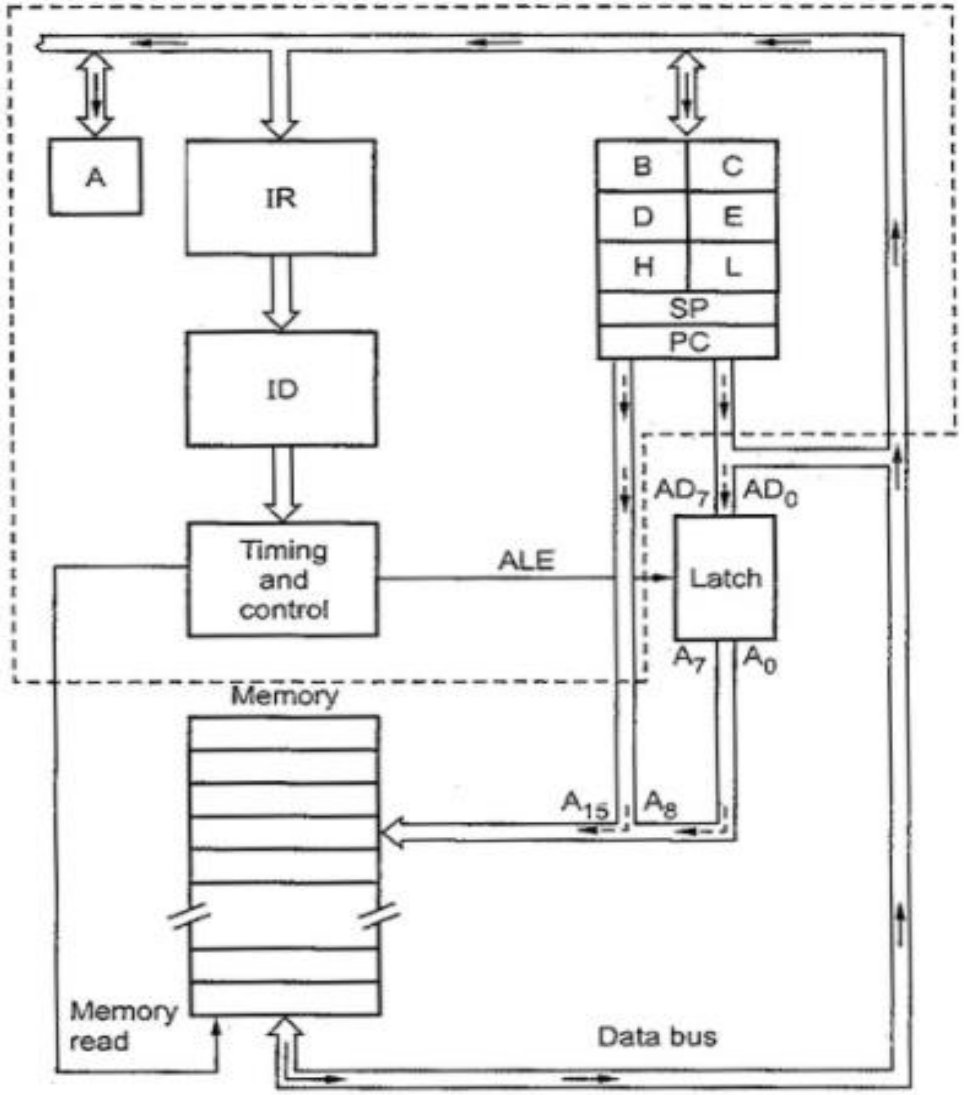
Step 4 : (State T_4)

- 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.

S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/$\overline{M}$ = 0, S1 =1 and S0 = 1. These status signals do not change throughout the OF machine cycle.
4	T_2	The microprocessor makes the $\overline{RD}$ line LOW to enable memory read and increments the Program Counter.
5		The contents on D7 – D0 (i.e. the Opcode) are placed on the address / data bus.
6	T_3	The microprocessor transfers the Opcode on the address / data bus to Instruction Register (IR).
7		The microprocessor makes the $\overline{RD}$ line HIGH to disable memory read.
8	T_4	The microprocessor decodes the instruction.

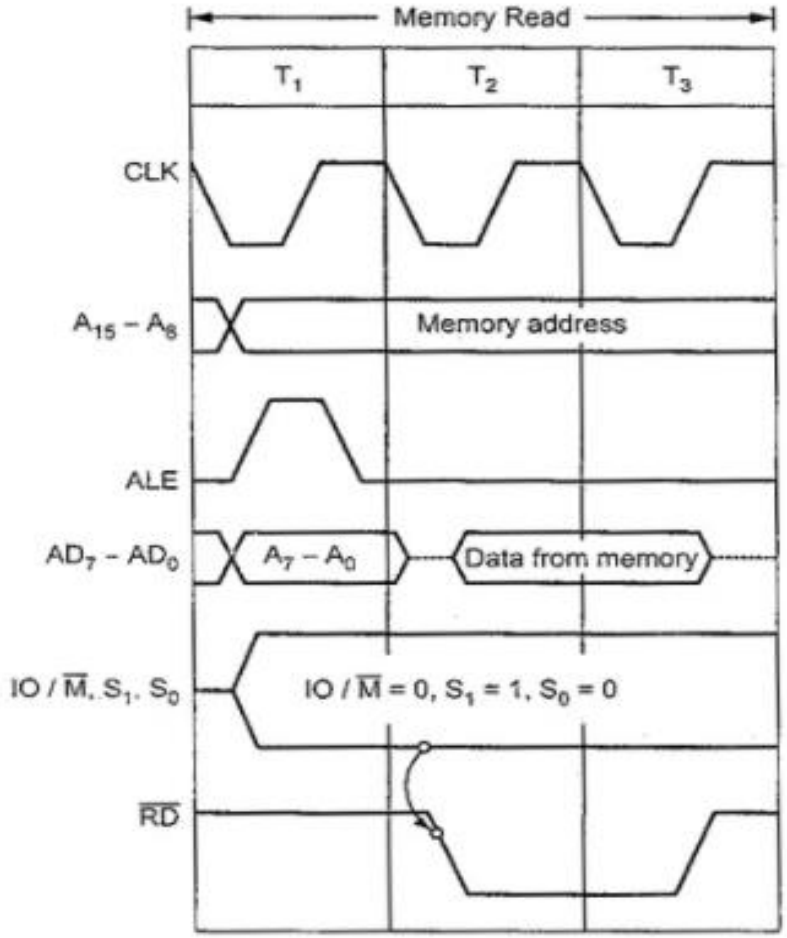
Opcode Fetch Cycle

Memory Read Cycle



→ Indicates data flow, --→ Indicates address flow

(a) Data flow from memory to microprocessor



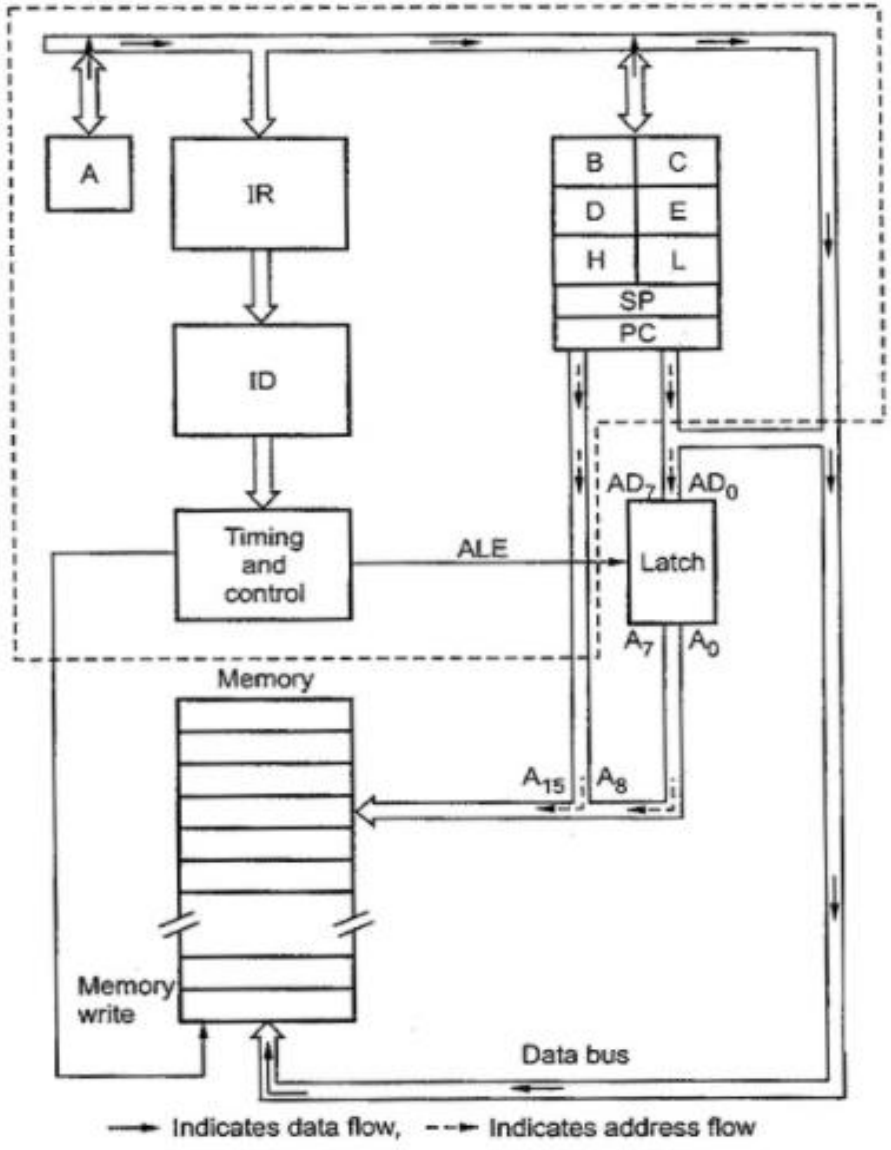
(b) Memory read machine cycle

- 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.(a) shows flow of data from memory to the microprocessor and Fig.(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 T_1 to T_3, and the status signal values (IO/M = 0, S_1 = 1, S_0 = 0) appropriate for memory read machine cycle are issued in T_1.

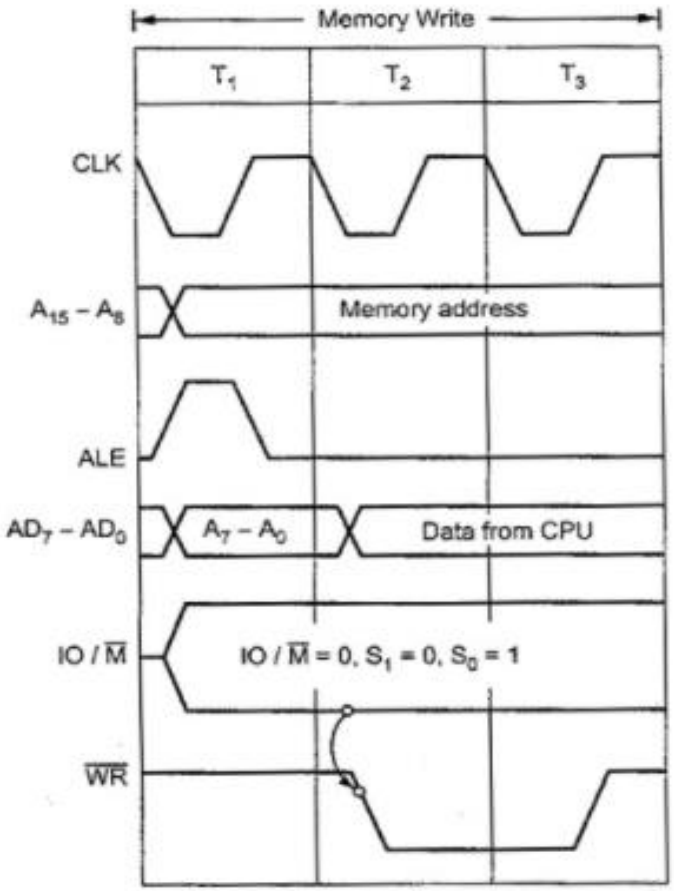
S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/*M*' = 0, S1 =1 and S0 = 0. These status signals do not change throughout the memory read machine cycle.
4	T_2	The microprocessor makes the RD' line LOW to enable memory read and increments the Program Counter.
5		The contents on D7 – D0 (i.e. the data) are placed on the address / data bus.
6	T_3	The data loaded on the address / data bus is moved to the microprocessor.
7		The microprocessor makes the RD' line HIGH to disable the memory read operation.

Memory Read Cycle

Memory Write Cycle



(a) Data flow microprocessor to memory



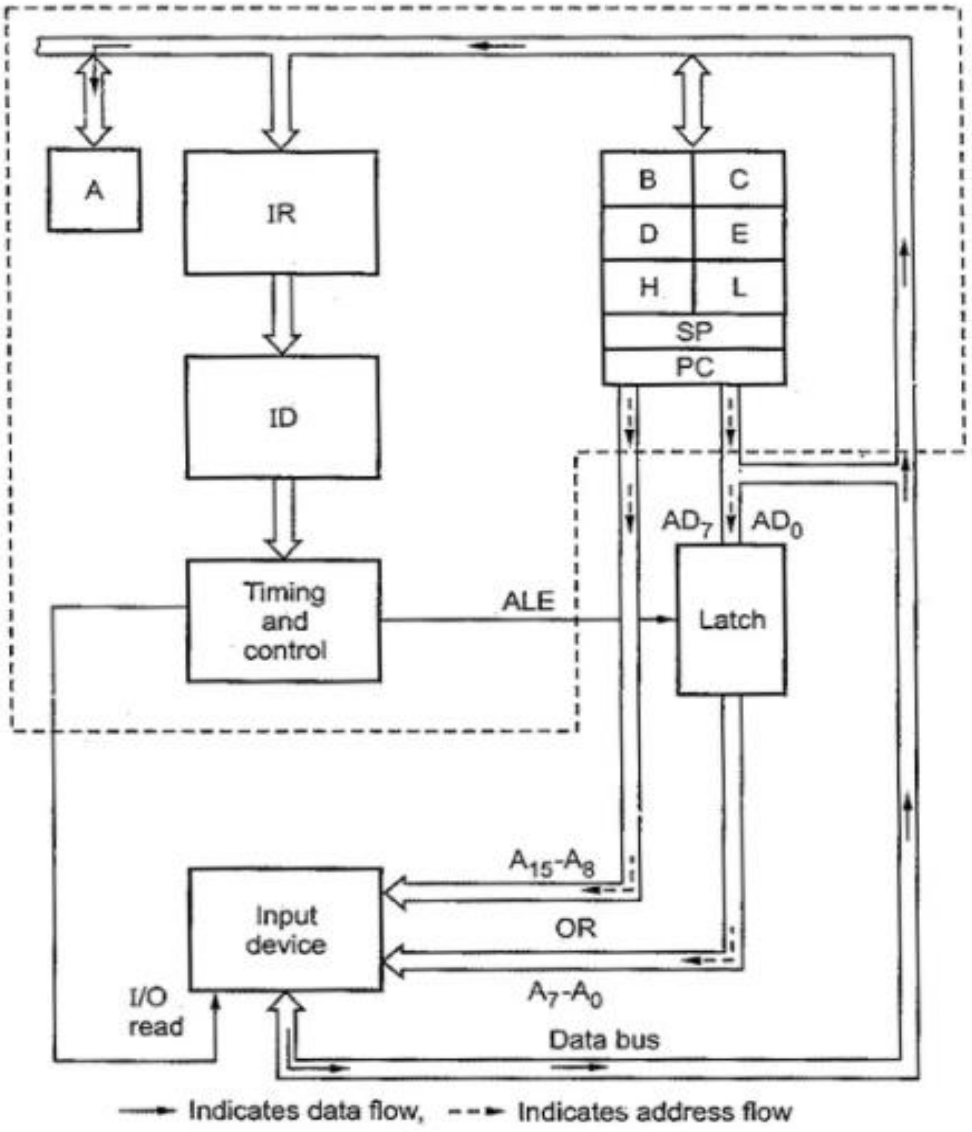
(b) Memory write machine cycle

- 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. 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 RD, WR signal goes low during T_2 and T_3.
- The status signals for memory write cycle are : IO/M = 0, S_1 = 0, S_0 = 1.
- The following section describes the memory write machine cycle in step by step manner.

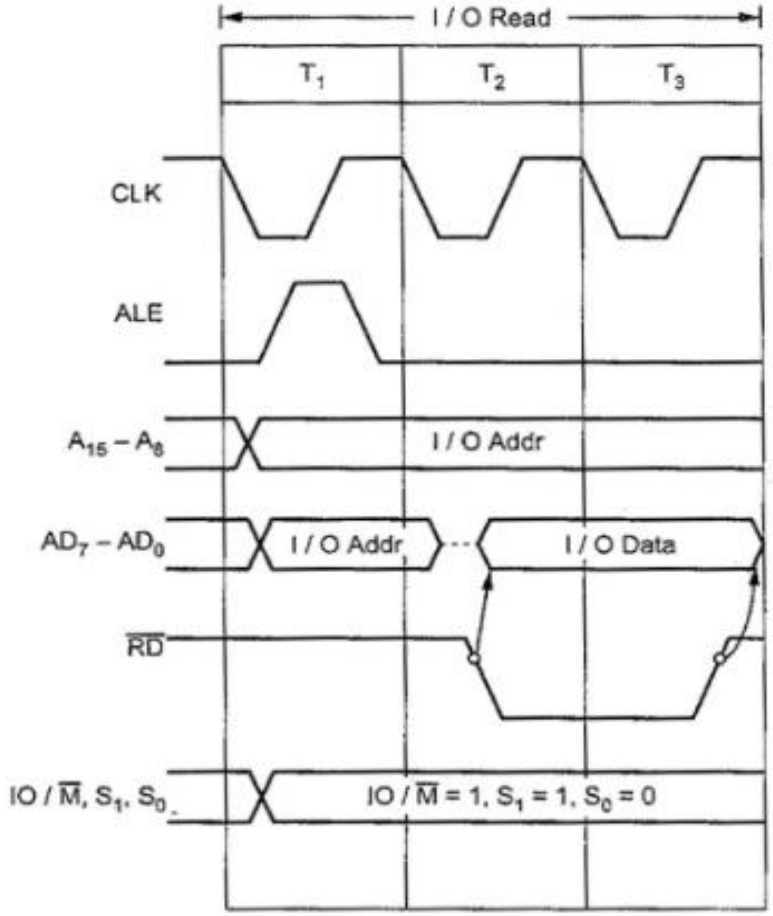
S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/M' = 0, S1 =0 and S0 = 1. These status signals do not change throughout the memory write machine cycle.
4	T_2	The microprocessor makes the WR' line LOW to enable memory write.
5		The contents of the specified register are placed on the address / data bus.
6	T_3	The data placed on the address / data bus is transferred to the specified memory location.
7		The microprocessor makes the WR' line HIGH to disable the memory write operation.

I/O Read and I/O Write Cycle

- 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 10/M signal is high for I/O read and I/O write machine cycles.
- High IO/M signal indicates that it is an I/O operation.
- Fig. 1 and Fig. 2 show the timing diagrams for I/O read and I/O write cycles, respectively.



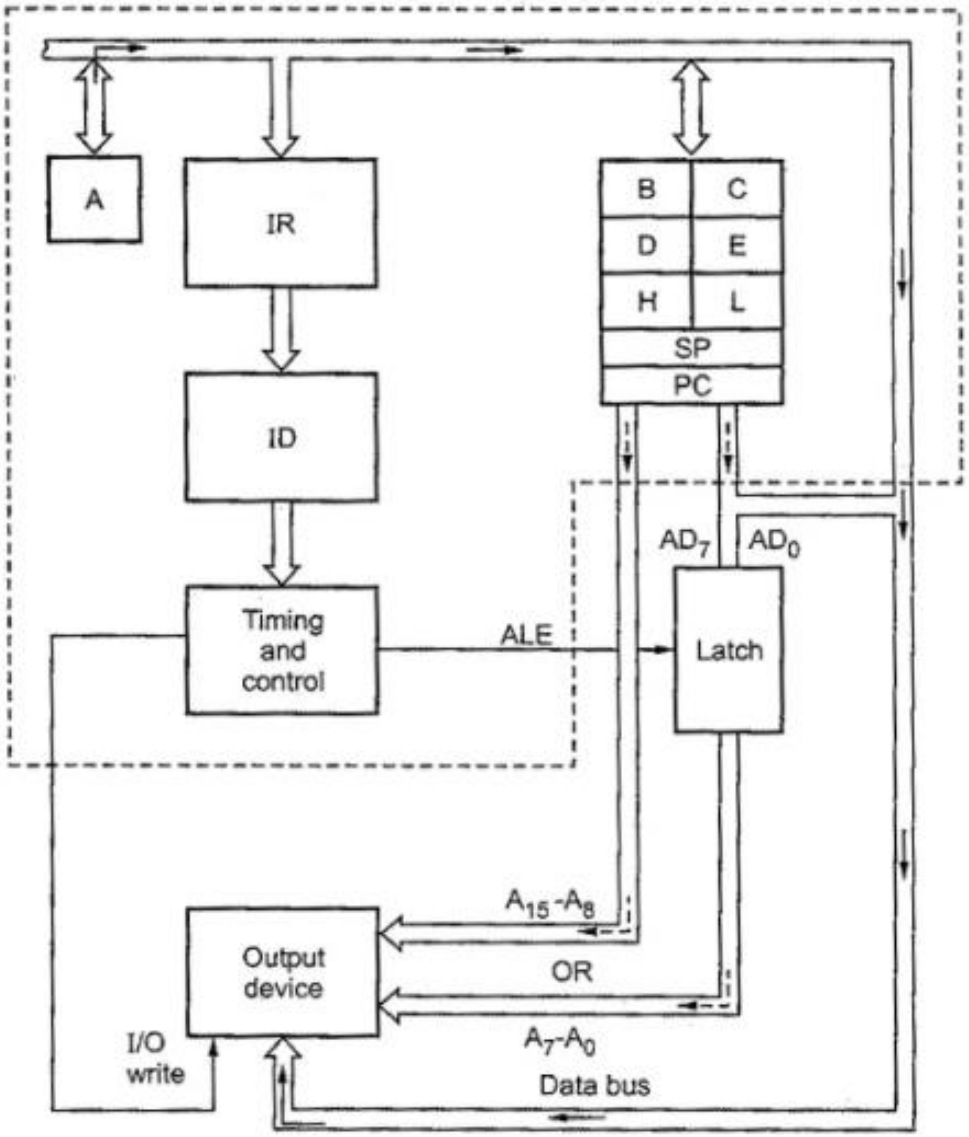
(a) Data flow from input device to microprocessor



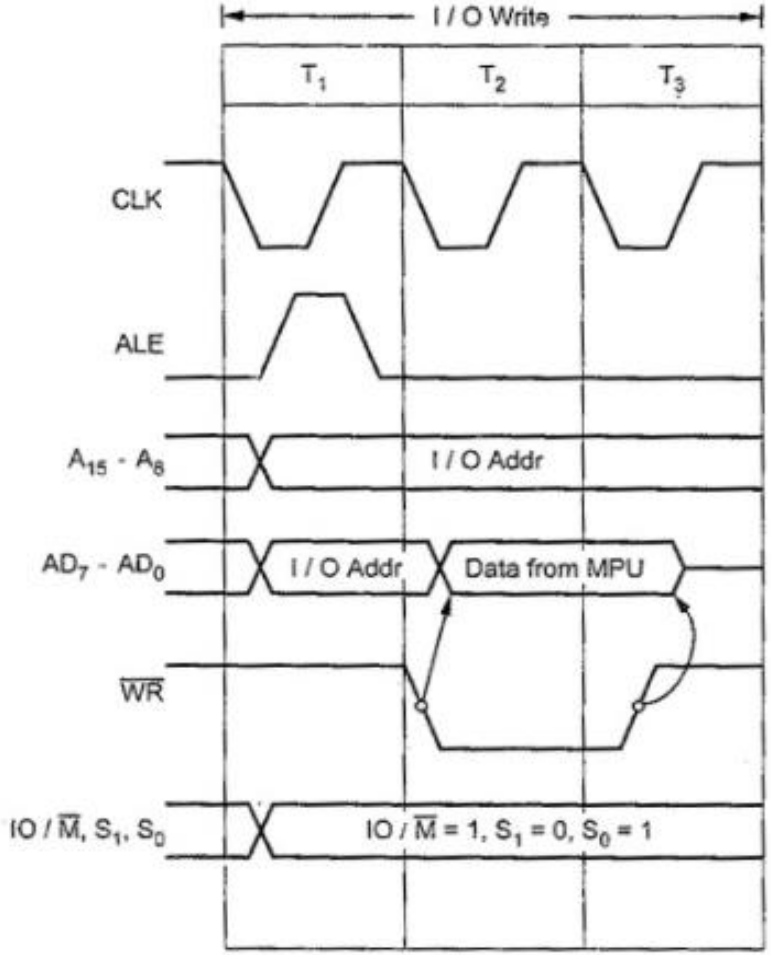
(b) I/O read memory cycle

The steps in I/O Read machine cycle are given in table.

S. No	T state	Operation
1	T_1	The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/$\overline{M}$ = 0, S1 =1 and S0 = 0. These status signals do not change throughout the I/O read machine cycle.
4	T_2	The microprocessor makes the $\overline{RD}$ line LOW to enable I/O read.
5		The contents on D7 – D0 (i.e. the data) are placed on the address / data bus.
6	T_3	The data loaded on the address / data bus is moved to the microprocessor ie., to the accumulator.
7		The microprocessor makes the $\overline{RD}$ line HIGH to disable the I/O read operation.



(a) Data flow from microprocessor to output device



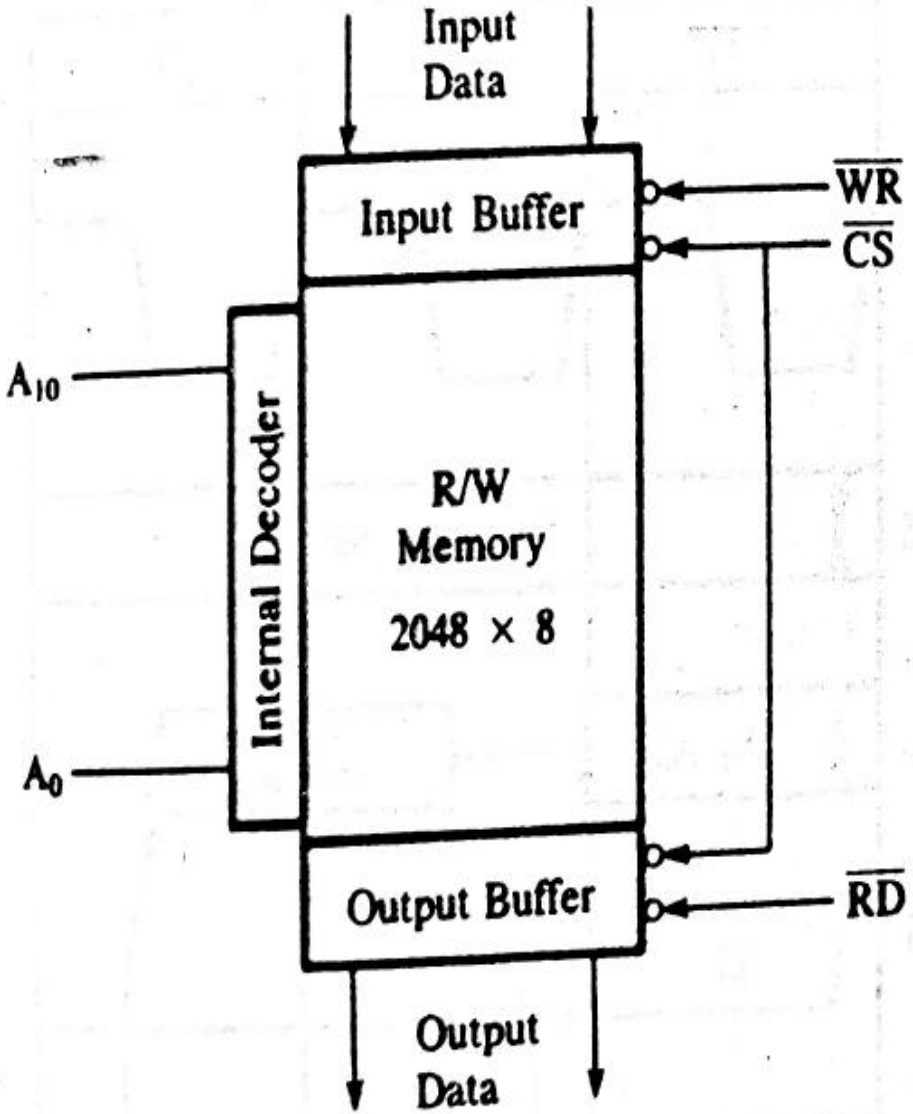
(b) I/O write machine cycle

The steps in I/O Read machine cycle are given in table.

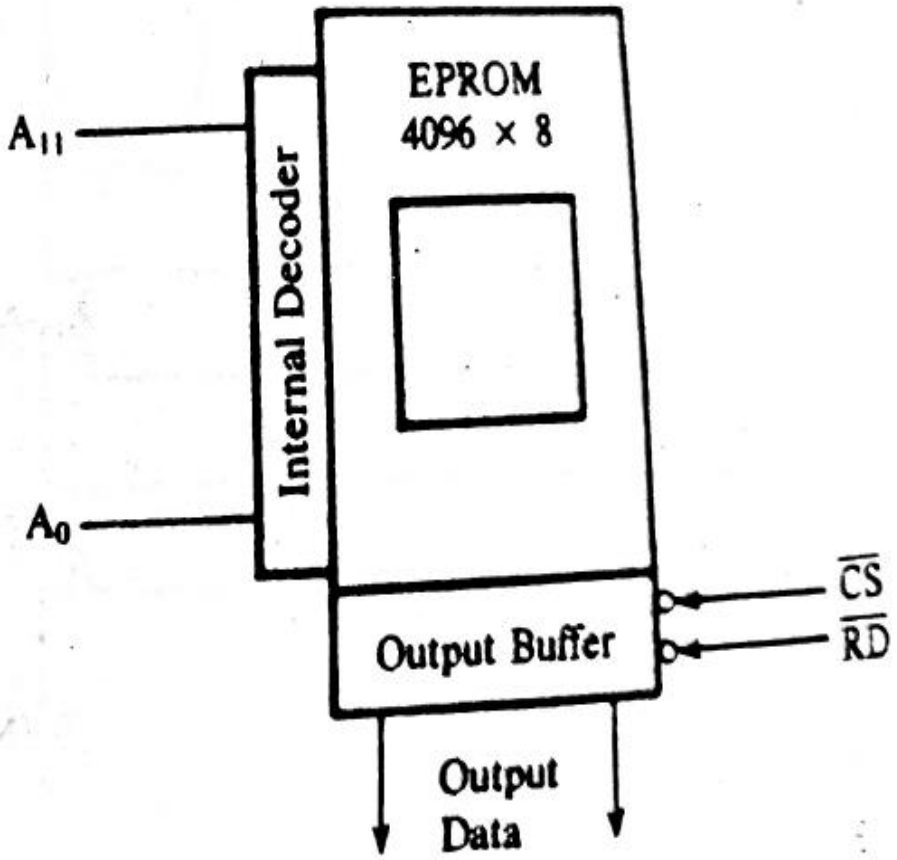
S. No	T state	Operation
1	T_1	The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/M' = 0, S1 =0 and S0 = 1. These status signals do not change throughout the I/O write machine cycle.
4	T_2	The microprocessor makes the WR' line LOW to enable I/O write.
5		The contents of the Accumulator are placed on the address / data bus.
6	T_3	The data placed on the address / data bus is transferred to the specified I/O port.
7		The microprocessor makes the WR' line HIGH to disable the I/O write operation

Memory Interfacing

Read/Write memory (R/WM) is a group of registers to store binary information. Figure 4.11(a) shows a typical R/W memory chip; it has 2048 registers and each register can store eight bits indicated by eight input and eight output data lines.* The chip has 11 address lines A_{10}–A_0, one Chip Select ($\overline{CS}$), and two control lines: Read ($\overline{RD}$) to enable the output buffer and Write ($\overline{WR}$) to enable the input buffer. Figure 4.11(a) also shows the internal decoder to decode the address lines. Figure 4.11(b) shows the logic diagram of a typical EPROM (Erasable Programmable Read-Only Memory) with 4096 (4K) registers. It has 12 address lines A_{11}–A_0, one Chip Select ($\overline{CS}$), and one Read control signal. This chip must be programmed (written into) before it can be used as a read-only memory. Figure 4.11(b) also shows a quartz window on the chip that is used to expose the chip to ultraviolet rays for erasing the program. Once the chip is programmed, the window is covered with opaque tape to avoid accidental erasing. For interfacing the R/W memory, Figure 4.11(a), and the EPROM, Figure 4.11(b), the process is similar; the only difference is that the EPROM does not require the $\overline{WR}$ signal.


Input
Data
Input Buffer
WR
CS
A10
Internal Decoder
R/W
Memory
2048 × 8
A0
Output Buffer
RD
Output
Data

(a)


A11
Internal Decoder
EPROM
4096 × 8
A0
CS
Output Buffer
RD
Output
Data

(b)

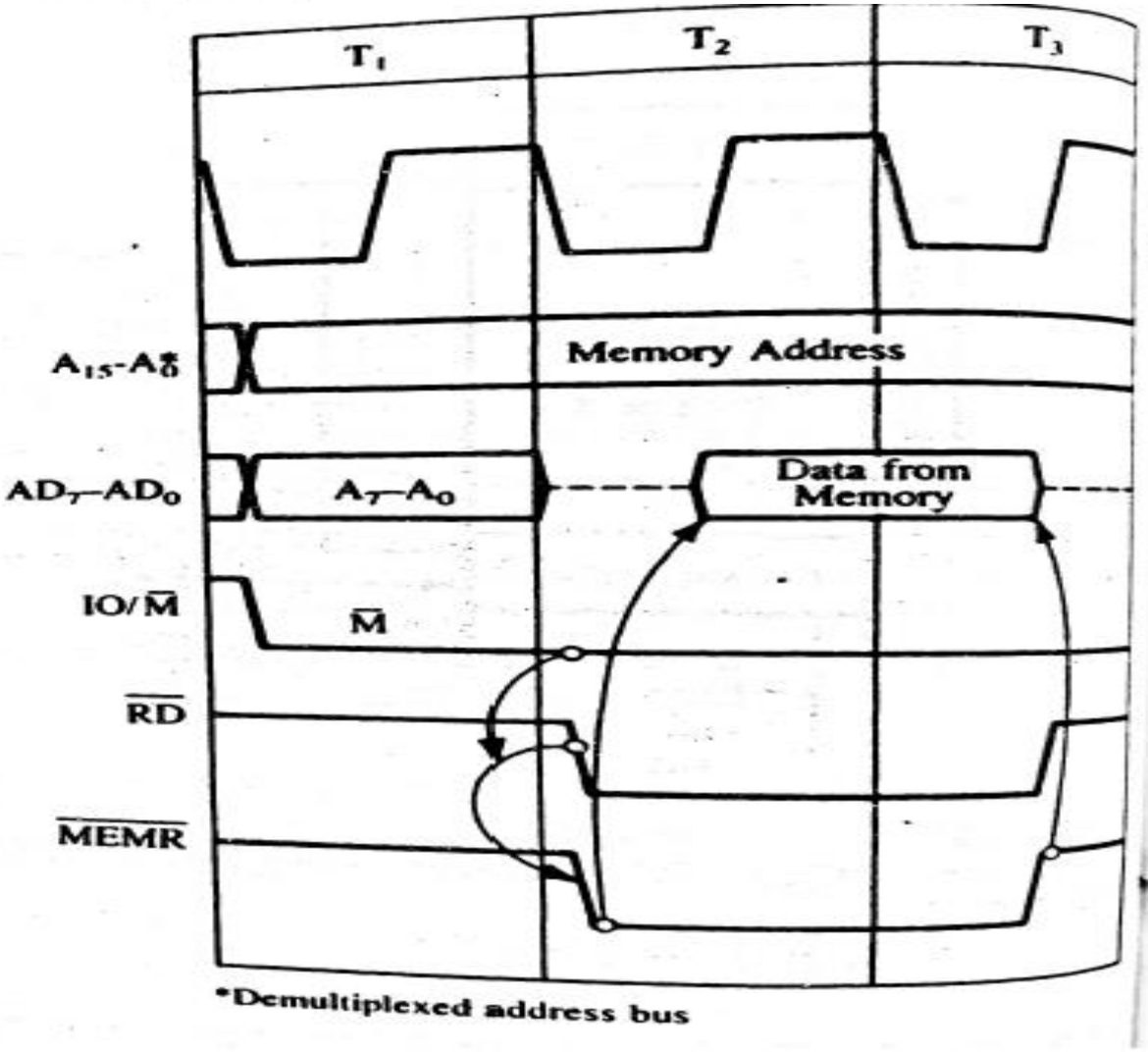
Concept of Memory Interfacing

The primary function of memory interfacing is that the microprocessor should be able to read from and write into a given register of a memory chip. Recall from Chapter 3 that to perform these operations, the microprocessor should

1. be able to select the chip.
2. identify the register.
3. enable the appropriate buffer.

Let us examine the timing diagram of the Memory Read operation (Figure 4.12) to understand how the 8085 can read from memory. Figure 4.12 is the M_2 cycle of Figure 4.10 except that the address bus is demultiplexed. We could also use the M_1 cycle to illustrate these interfacing concepts.

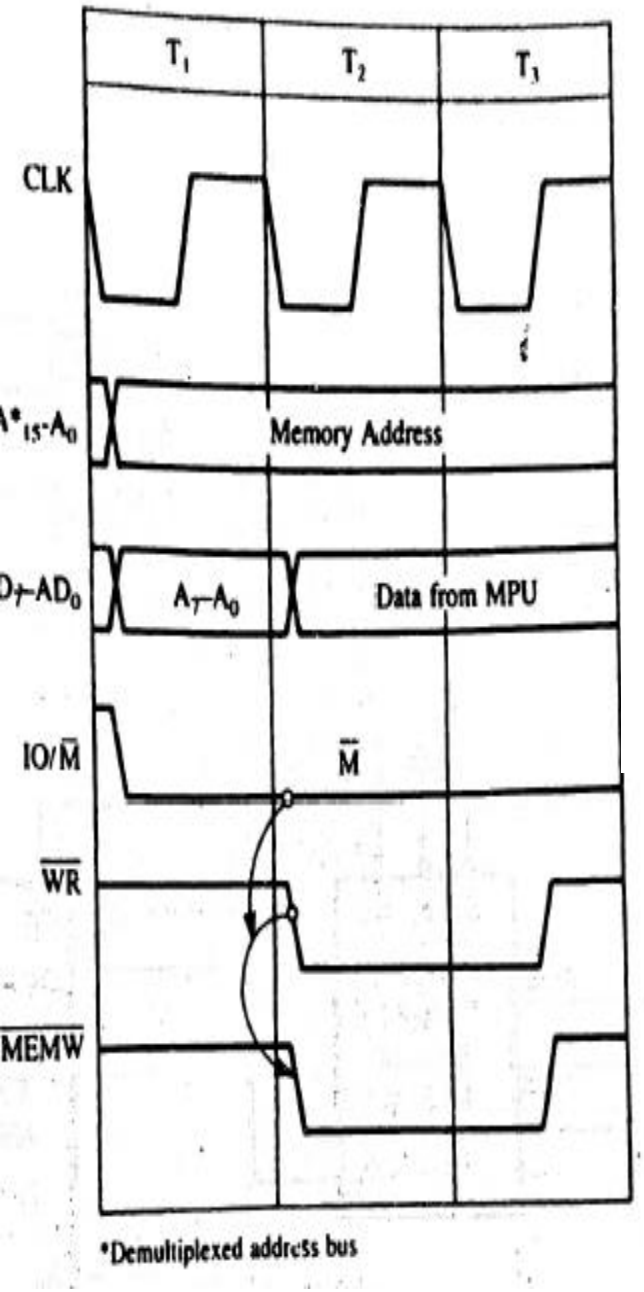
1. The 8085 places a 16-bit address on the address bus, and with this address only one register should be selected. For the memory chip in Figure 4.11(a), only 11 address lines are required to identify 2048 registers. Therefore, we can connect the low-order address lines A_{10}–A_0 of the 8085 address bus to the memory chip. The internal decoder of the memory chip will identify and select the register for the EPROM, Figure 4.11(a).



2. The remaining 8085 address lines (A_{15}–A_{11}) should be decoded to generate a Chip Select ($\overline{CS}$) signal unique to that combination of address logic (illustrated in Examples 4.3 and 4.4).
3. The 8085 provides two signals—IO/$\overline{M}$ and $\overline{RD}$—to indicate that it is a memory read operation. The IO/$\overline{M}$ and $\overline{RD}$ can be combined to generate the $\overline{MEMR}$ (Memory Read) control signal that can be used to enable the output buffer by connecting to the memory signal $\overline{RD}$.
4. Figure 4.12 also shows that memory places the data byte from the addressed register during T_2, and that is read by the microprocessor before the end of T_3.

To write into a register, the microprocessor performs similar steps as it reads from a register. Figure 4.13 shows the Memory Write cycle. In the Write operation, the 8085 places the address and data and asserts the IO/$\overline{M}$ signal. After allowing sufficient time for data to become stable, it asserts the Write ($\overline{WR}$) signal. The IO/$\overline{M}$ and $\overline{WR}$ signals can be combined to generate the $\overline{MEMW}$ control signal that enables the input buffer of the memory chip and stores the byte in the selected memory register.

To interface memory with the microprocessor, we can summarize the above steps as follows:



1. Connect the required address lines of the address bus to the address lines of the memory chip.
2. Decode the remaining address lines of the address bus to generate the Chip Select signal, as discussed in the next section (4.3.3), and connect the signal to select the chip.
3. Generate control signals $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ by combining $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals with $\text{IO}/\overline{\text{M}}$, and use them to enable appropriate buffers.

Any Question???

Thank you…