

Chapter 3: Bus structure, memory & I/O Interfacing

Bus structure:- It refers to the common path been chip & peripherals. They are group of wires to carry information b/w. The bus system can be divided into 3-functional groups.

- a) Address Bus. b) Data Bus c) Control Bus.

The Bus architecture in a microprocessor system can be shown as;

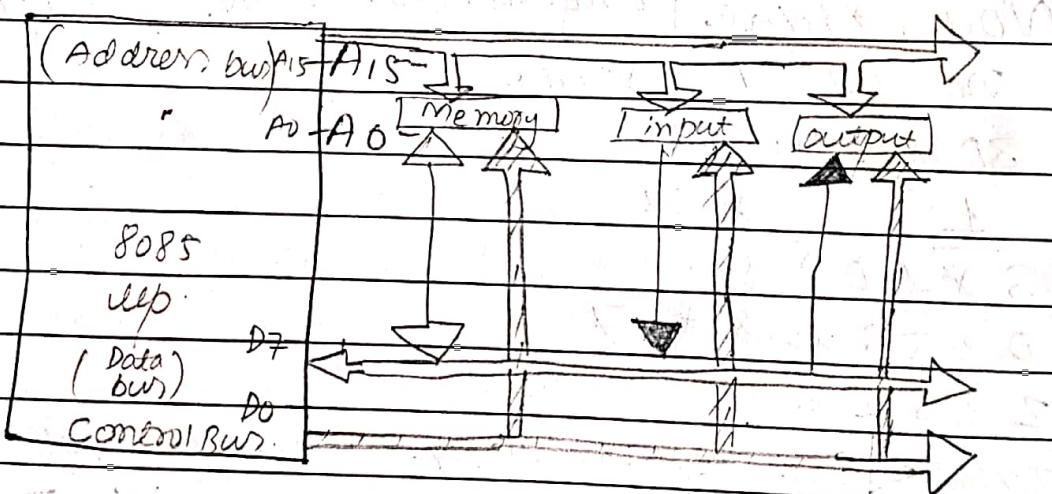


fig: Bus organization / Architecture of 8085 CPU

- a) Address Bus.

- A₀ to A₁₅ are Address Bus with group of 16-lines.
- They are unidirectional from CPU to peripheral device.
- It is used to identify a peripheral or a memory location.
- They are capable of addressing 2^{16} memory space
- = 65536/128
- = 64K memory

Data Bus

- b) → 8-bit lines ($D_0 - D_7$) are used for the flow of data.
- They are bidirectional.
- The number of lines is referred to as the width of databus.
- The width of databus is a key factor in determining the overall performance.
- The 8085 CPU contains 8-bit data bus which means it can handle $2^8 = 256$ instructions i.e. $00 \rightarrow FF$.

Control Bus

- The main job of control bus is to provide timing signals and synchronization signals.
- These are not a group of lines like address bus & data bus but individual lines that provide a pulse to indicate the CPU operation.
- CPU generates specific control signals for every operation.
- These signals are used to identify a device type with which CPU cannot communicate.
- Some of the control signals are;

- 1) Memory write, M-Read, I/O write, I/O Read.
- 2) Transfer ACK, Bus repeat, Bus Grant.
- 3) Interrupt Request, Interrupt ACK → ACKnowledge.

Synchronous & Asynchronous Buses

- i) Synchronous Bus; A synchronous bus has its events triggered by a clock. The clock transmits a regular sequence of 0's & 1's of equal duration.

A single I-O transmission is called a bus cycle.
A memory read operation in case of synchronous bus is shown as

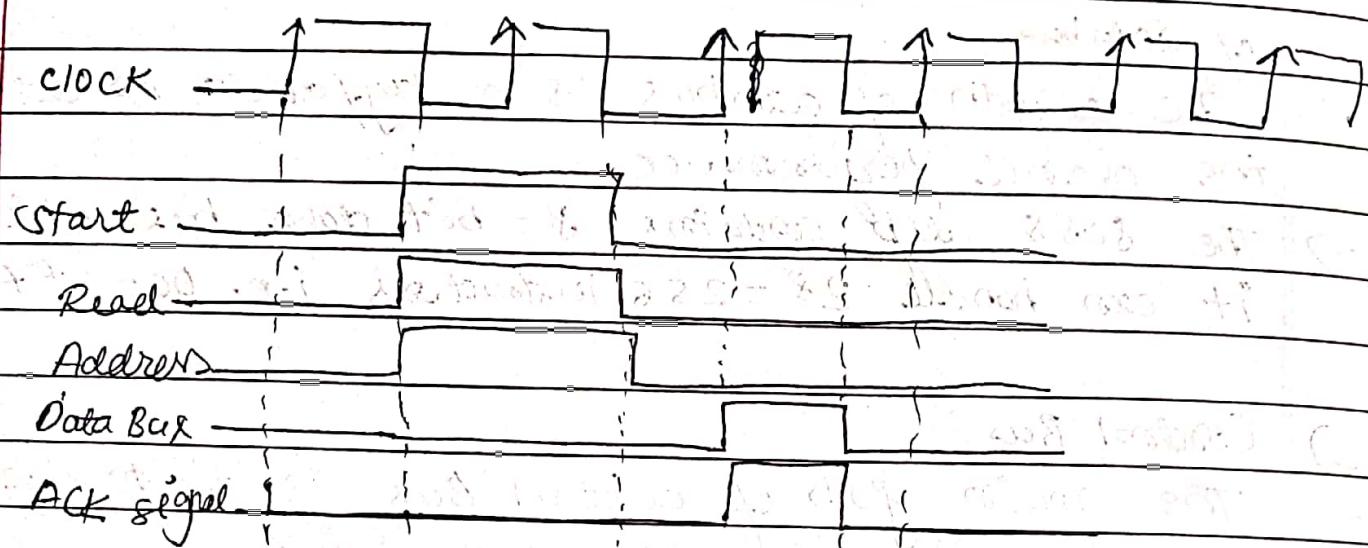


fig: Synchronous Bus on memory read operation.

- The CPU issues a start signal to indicate the presence of address information on the bus.
- Then it issues the memory read signal and places the memory address on the address bus.
- The address memory module recognizes the address after a delay/one clock cycle and places the data & acknowledgement signal on the bus.
- In a synchronous bus, all the devices are tied to a fixed clock rate

2) A synchronous Bus :

→ In an asynchronous bus the timing is maintained in such a way that occurrence of one event on the bus follows and depends on the occurrence of previous event.

let us consider a memory read operation.

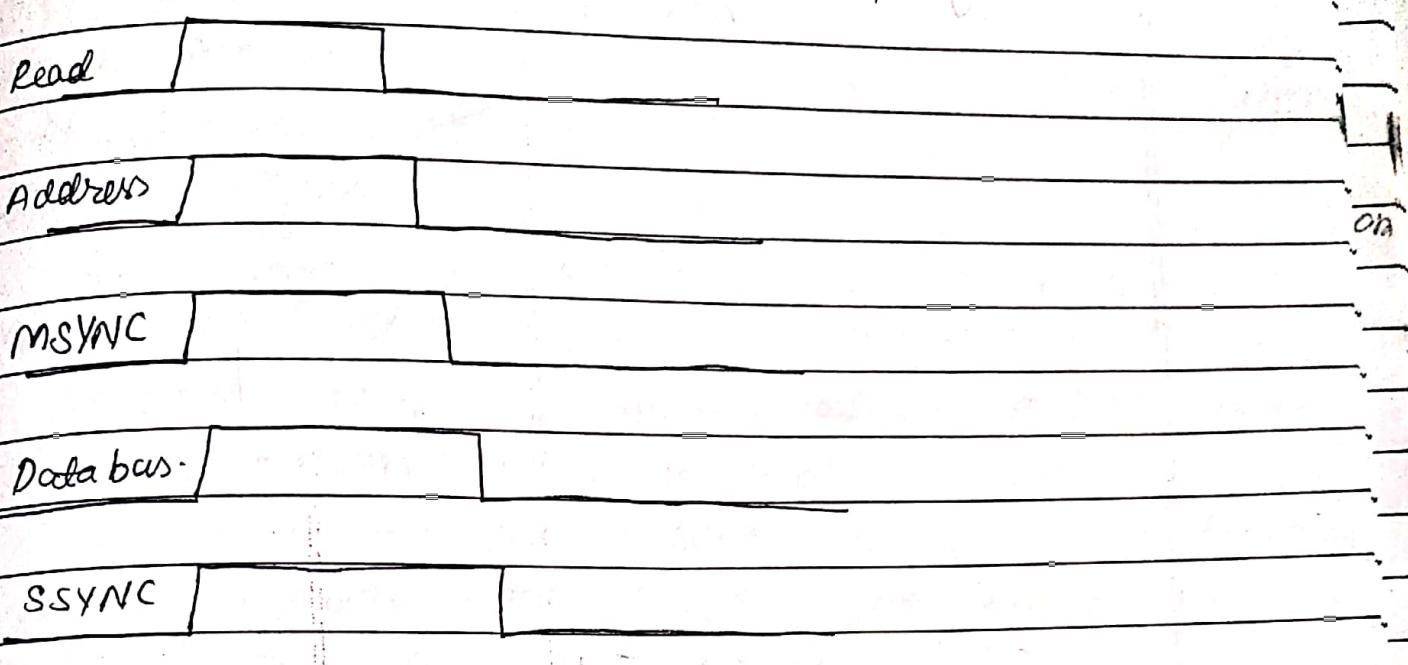
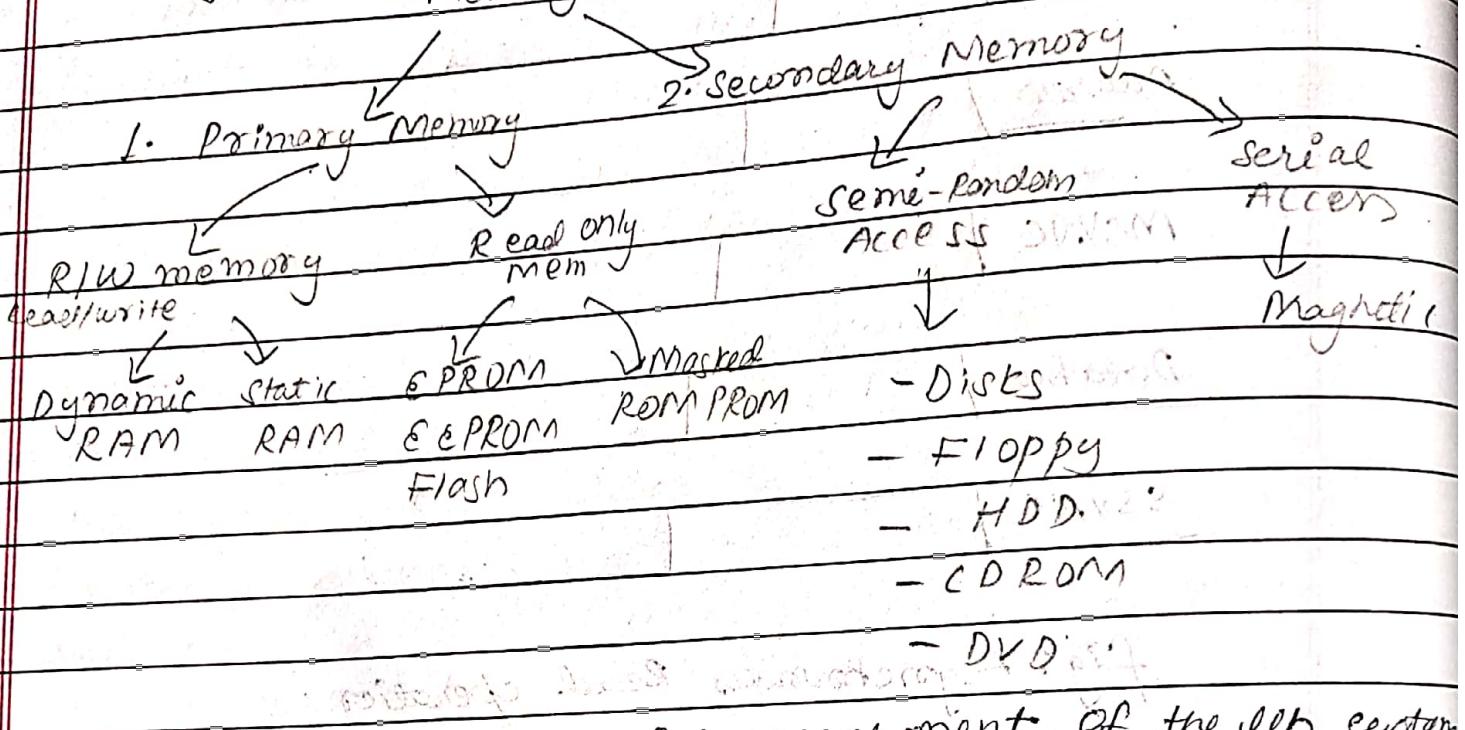


fig: Asynchronous Read operation.

- The CPU places the memory read and address signal on address bus.
- After allowing these two signals to stabilize, it issues a Master synchronous signal (msync) to indicate the presence of valid address and control signals on the bus.
- The addressed memory module responds with the data and slave synchronous signal (ssync).

Memory



- memory is an essential component of the CPU system.
- used to store both instruction & data.
- Memory is made up of registers and number of bits stored in a register is called memory word.
- For 8085 CPU, the memory word is identified by an address 2^{16} = 64K of memory (0000H - FFFFH).

Memory can be classified into two main categories.

- 1) Volatile memory → contents are lost when the power is cut-off. eg:- RAM (static & Dynamic)
- 2) Non-volatile memory → retains data even when there is no power.
eg:- ROM, PROM, EEPROM, Floppy, Harddrive.

* Primary Memory; It is the storage area where all programs are executed. The CPU can directly access

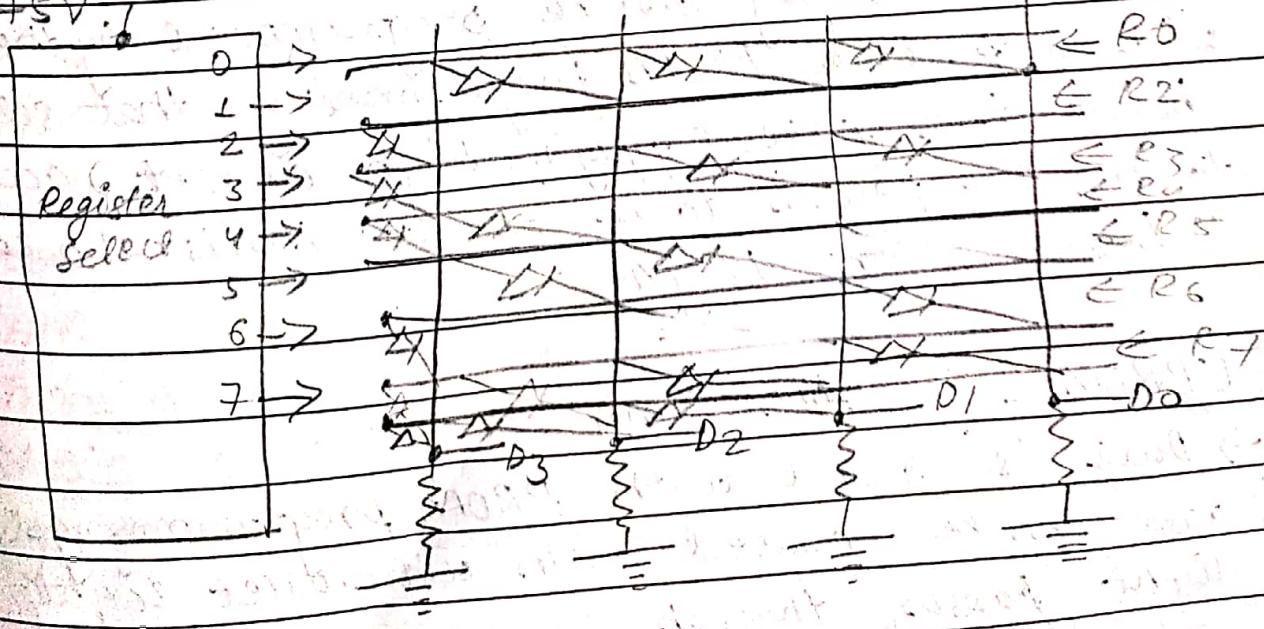
only those items/programs that are stored in primary memory.

Although the storage capacity is larger than registers but the speed is slower.

ROM \rightarrow RAM \rightarrow ROM - volatile, Read Only memory.

- stores the boot-strap loader file (essential instructions/programs to boot the computer)
- \rightarrow when the power is turned ON, the hardware of the computer sets the PC to the first address of the bootstrap loader
- \rightarrow The bootstrap loader program loads a portion of the operating system from the disk which prepares the computer for general use.

\rightarrow Rom is constructed



Here, each horizontal row is a register memory location. The Register R₀ contains 3-diodes, as seen. The output of the ROM is viewed as a word; i.e. D = D₃D₂D₁D₀

- In switch position 0, a high voltage turns on the diode for R₀ register & all other diodes are off. This means that a high output appears at D₂, D₁, D₀. Therefore a word stored at memory location 0 is D = 0111. When the switch is at 1, the word stored is D = 1000.

Similarly data in other registers are stored with integrated circuits. Manufacturers stores the word at the time of fabrication. These words are permanently stored once the diode are wired in place.

* Types of ROM;

a) PROM (Programmable ROM).

→ This memory can be programmed by the user with a special PROM programming that selectively burns the fuses (applying high current) accordingly to the bit pattern to be stored. This is also called "Burning the PROM".

b) EEPROM (Erasable ROM)

→ Data is stored with PROM programming. Later data can be erased with ultraviolet light. The light passes through a quartz window onto the chip where it releases stored charges. It copies

out. the stored contents. In other words, EEPROM is called "Ultraviolet - light - erasable ROM".

c) EEPROM (Electrically Erasable ROM)

- It doesn't require UV rays to erase the contents. It can be completely erased or have certain bytes charged using electrical pulses. EEPROM is slower hence are not used in high speed circuits.

d) Flash memory: This is a modified version of EEPROM. The main difference is the erasing procedure. EEPROM can be erased at a register level whereas flash memory are erased at sector block level.

RAM (Random Access memory) / R/W memory

- It is a volatile memory and is used to store programs & data for immediate use of the processor.
- Data can be easily written into & read from RAM.

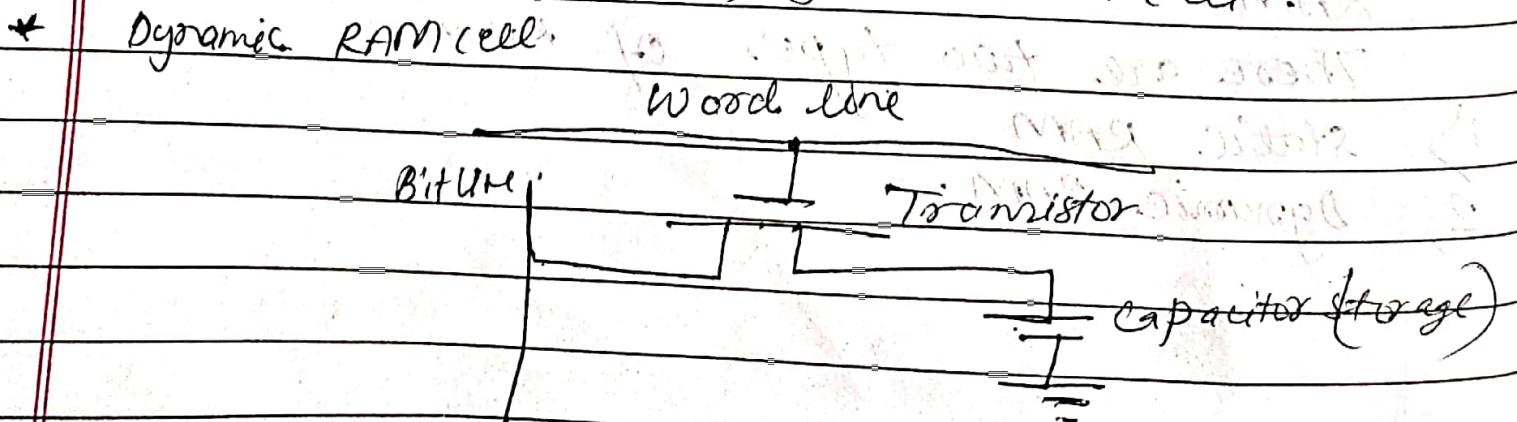
- 1) static RAM
- 2) Dynamic RAM

1) static RAM; These memory are made up of flipflops and stores the binary bits in the form of voltage. Each memory cell requires six transistors. These memory are more expensive & power consuming than dynamic RAM. They are normally used for cache memory with a fast access time of 15 to 30 nanoseconds.

2) Dynamic RAM; It is less expensive than SRAM, DRAM uses different approach to store data. Informational data are stored as charges in very small capacitors. If a charge exists in a capacitor, it is interpreted as binary 1 and the absence of charge is binary 0. since DRAM uses capacitors there is a slight chance of leakage of charge thus it needs constant refreshment through signal every few milliseconds.

* Two common types of RAM cells which uses MOS technology are:

1) Static RAM cell. 2) Dynamic RAM cell.



The above fig shows the dynamic RAM cell which consists of a MOSFET and a capacitor.

- The capacitor is used to store the data bits.
- The transistor acts as a switch.

In an ideal case the capacitor could store the charge permanently. But since the capacitor slowly loses charge in real world, it needs constant refreshment.

Read operation on Dynamic RAM cell.

- The voltage is applied in the wordline to turn on the transistor.
- Then the charge in capacitor will be available in bit line.
- Eventually the capacitor will start to lose its charge.
- Hence it is also called destructive READ operation.

Write operation

- All the bitline/data line are pre charged with some finite value (high data line).
- The MOSFET (transistor) is turned on and the capacitor is charged.
- When the data line goes low, the MOSFET turns off and the capacitor retains its charge.

Advantages of Dynamic RAM over static RAM.

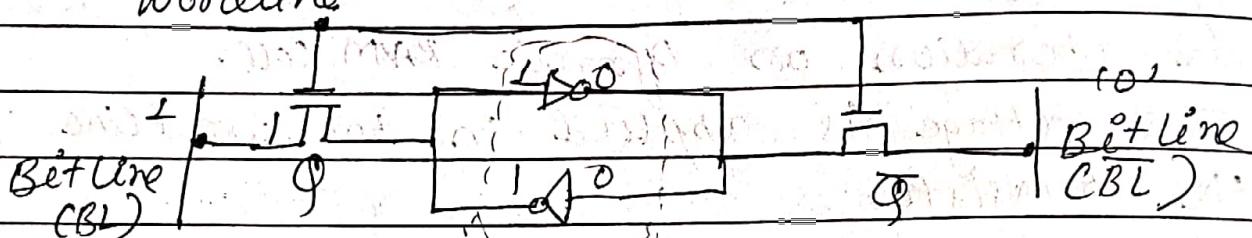
- Simple.
- low power consumption.
- large memory can be constructed at low cost per-bit.

Disadvantages of DRAM

- Needs constant refreshment.
- Process of refreshment requires additional memory.
- Slower than SRAM.

Static RAM cell

Wordline



→ cross-coupled inverters

- To turn ON the transistor, voltage is applied to wordline.
- The voltage available ' ϕ ' will be at BL and $\bar{\phi}$ will be at \bar{BL} .
- Two bit lines are used / required to increase the speed of Read & write operation.

* READ Operation

- Transistors ϕ and $\bar{\phi}$ are activated which activates Bit line.

→ The stored data appears on the bit line / data line

* WRITE operation

- The two transistors are turned ON via WL high.
- The data lines / bit lines are forced into the state $BL(\text{high}) \& \bar{BL}(\text{low})$.

When the signals are removed Q transistor will continue to hold its state.

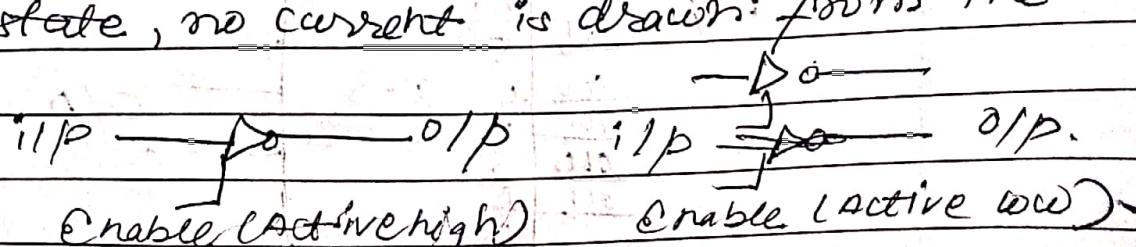
The forced state is self sustaining & stable.

The stored data will be held by flipflop until power is removed.

Interfacing Devices:

- There are semiconductor device/chips that are connected to the peripherals of I/O system.
 - Commonly used interfacing device are tri-state buffers, encoders and decoders and latches.
 - Buffers are the logic circuits that amplifies the current or power. It is primarily used to increase the driving capability of a logic circuit.
- i/p o/p
↓
Buffer.

- Tri-state buffer; Generally, a logic device has two states, logic 1 and logic 0. But a tri-state buffer has a third state i.e. high impedance state.
- The device has third state (high impedance) called enable (EN).
- When this line is activated, the tri-state buffer functions as same as ordinary logic device but when the line is deactivated, the logic device goes into a high-impedance state.
- In this state, no current is drawn from the system.

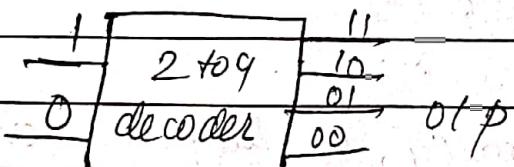


→ Tristate logic is critical to proper functioning of microcomputer system.

* **Decoder:** The decoder is a logic circuit that identifies each combination of the signal present at its input.

→ If the input to the decoder has n lines, the decoder will have 2^n output lines - eg; if $n=2$, the no. of output lines is $2^2 = 4$ i.e. 00, 01, 10, 11. Each of these lines are identified by selector lines (0-3). If 11 is input, the output line 3 will be selected (logic 1) and others will remain 0.

→ Various types of decoders are available eg: 3 to 8, 2-4, 4-1 etc.

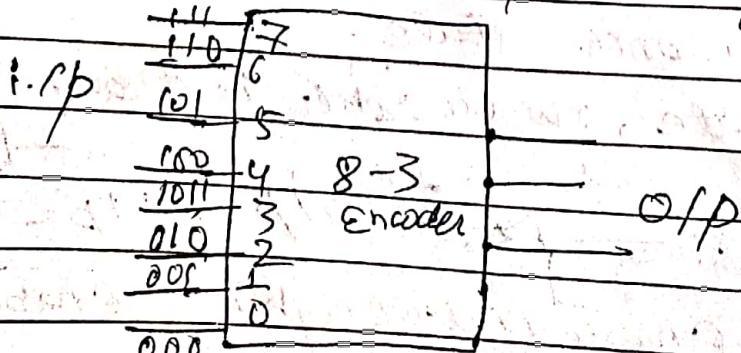


Encoder

fig 2-to-4 decoder.

* **Encoder:**

→ An encoder is a logic circuit that provides the appropriate code as o/p for each input signal.
→ It is the reverse process of decoding.



- If O goes low, the output is 000, when input line 5 goes low, the output is 101.
- * Latches; when the dep. cont'l sends an output, data are available on data bus only for few milliseconds so, a latch is used to hold the data for time being
- Latches are flip-flops; D - flipflop, R - flipflop, J - K flipflops.

Internal structure of memory

→ The internal structure of a 8×8 memory is shown below. Every memory unit will have similar type of structure.

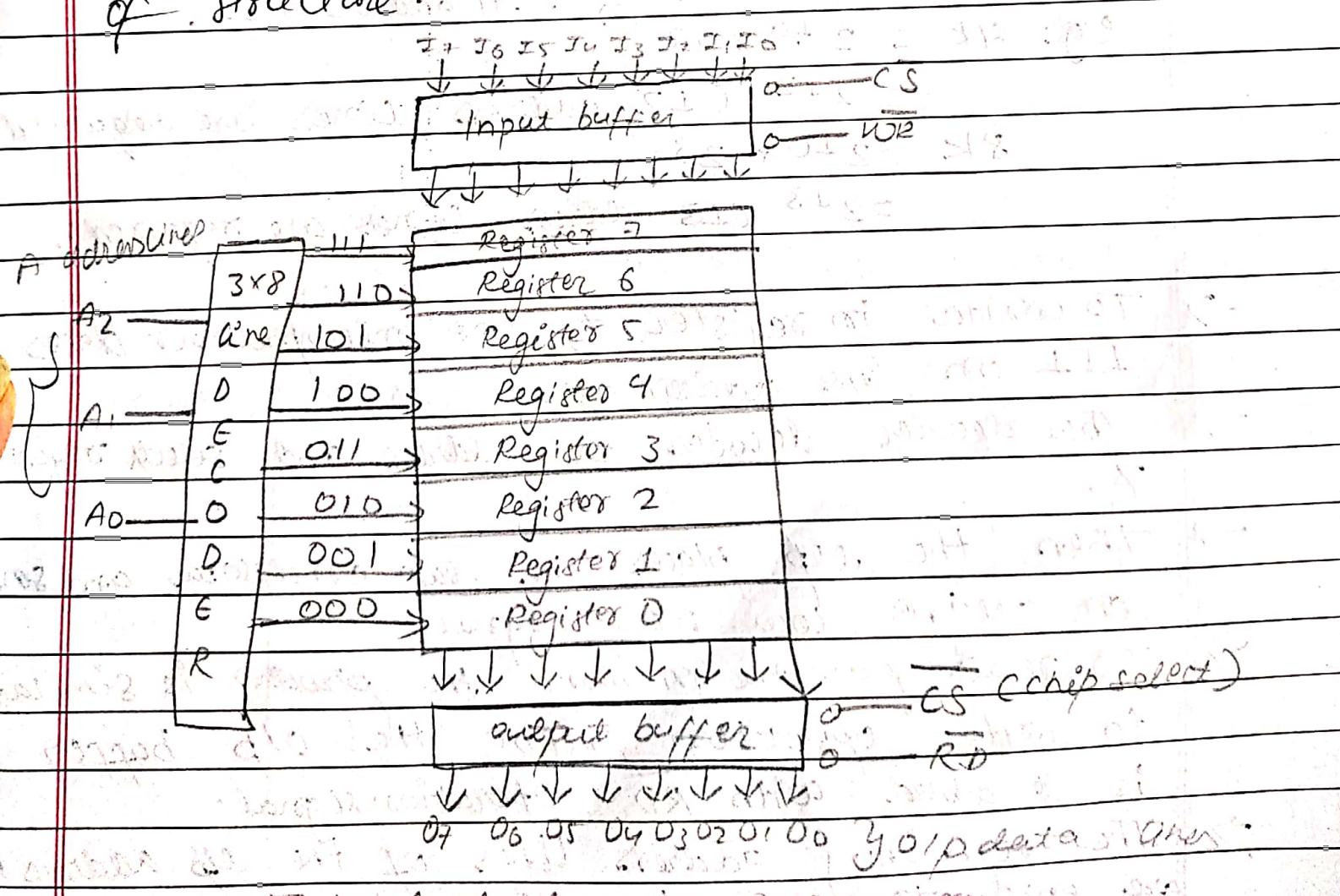


fig: Internal structure of 8×8 memory

- The internal structure of memory consists of address decoder, I/P & O/P buffer, registers, address lines, data lines, \overline{RD} , \overline{WR} , \overline{CS} and control lines.
- The no. of address lines will be determined by the memory capacity.
- The no. of data lines will be determined by memory size.
- For example: the memory capacity $1K \times 8$ will have 10 address lines & 8 data lines.
 $i.e. 2^{10} = 1K(1024)$
- ∴ We need 10 address lines.
- Eg: $2K \times 4$; 4 data lines.
 Address lines.
 $2^{10} \times 2^1 = 2^{11}$; $\therefore 11$ address lines.
- Eg: $4K = 2^{10} \times 2^2$
 $= 2^{12}$ (12 address lines are required).
- $8K = 2^{10} \times 2^3$
 $= 2^{13}$ (13 address lines are required).
- To write in register 7 , the microprocessor places 111 on the address line.
- The decoder decodes the address and select register 7 .
- Then the I/P places the data on databus and sends an active low \overline{WR} signal.
- To read from the memory, the process is similar to write operation except the O/P buffer is enabled with \overline{RD} Operation signal.
- The remaining address lines of the I/P address bus are used to select the chip (\overline{CS}).

Design an address decoding CKT to interface
2KB RAM, 4KB ROM & 8KB RAM with 8085 CPU.

No. of Address lines available in 8085 CPU is
(A₀ - A₁₅) 16 address lines.

For 2KB RAM 1

$$\text{size} = 2 \text{KB}$$

$$= 2^{10} \times 2^1 = 2^{11}$$

∴ The no. of address lines required is 11 (A₀ - A₁₀)

For ROM = 4KB

$$= 2^{10} \times 2^2 = 2^{12}$$

∴ No. of address lines required is 12 (A₀ - A₁₁)

For RAM 2

$$\text{size} = 8 \text{KB}$$

$$= 2^{10} \times 2^3 = 2^{13}$$

∴ no. of address lines required is 13 (A₀ - A₁₂).
selection lines RAM 2 has used up to L2 so it can't be used by other

Memory	Address	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
RAM 1	ST: 0000H	X	0	0	X	X	0	0	0	0	0	0	0	0	0	0	0
	END: 0FFFH	X	0	0	X	X	1	1	1	1	1	1	1	1	1	1	1
ROM	ST: 2000H	X	0	1	X	0	0	0	0	0	0	0	0	0	0	0	0
	END: 2FFFH	X	0	1	X	1	1	1	1	1	1	1	1	1	1	1	1
RAM 2	ST: 4000H	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	END: 5FFFH	X	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Selection lines
Grounded

Timing Diagram

classmate

Date _____

Page _____

DD width
MOV A, B // ADD B.

0000; 3E (Opcode) Operand
MVI A, 32 H.
0000; MVI A//00 (UT)
0001; 32 (3T)

Instructions.

- ① 1-byte instrn
- ② 2-byte instrn
- ③ 3-byte instrn

LDA 16-bit add add, STA C050
M-read: 3T
0000; 32
M-write: 3T
0001; 50
I/O Read: 3T
0002; C0
I/O write: 3T

STA \rightarrow m.write. STA $\frac{0250}{16} = 13$ (000; STA(4T))
STAX \rightarrow except this all are m.read. LDA $\frac{0250}{16} = 13$ (001; 3D(3T))
0002; C0(3T)

Mov A, B
Let us assume mov A, B lies in memory location C050; Mov A, B has only 4 T-states.

Opcode Fetch:

	T ₁	T ₂	T ₃	T ₄	
CLK					
A ₁₅	X				
Adres ₁₅ A ₈₋₂₅	X				
Adres ₁₅ A ₈₋₂₅ m ₁₅₋₀					
ADD1 Add ₁₅	C. O. A				
ADD1 Add ₁₅	50, 1				
ADD1 Add ₁₅					
RD					
WR					

Higher address (0-11)
Unspecified
A₁₅ A₈₋₂₅ m₁₅₋₀
Opcode mov, A
ALE
RD
WR

Active bus is m.read.
Kam nahi ka high state
Kam kula low state ka
of compilation

MOV *classmate*
Data
Page

~~I_{O/M}~~

~~S_{0, S₁}~~

~~S_{0, S₁}~~

~~I_{O/M} = 0. S_{0, S₁}~~ = 1

Machine cycle.

opcode Fetch

Memory Read

memory write

I/O Read

I/O write

~~S₀~~ ~~S₁~~

~~1~~ ~~1~~ ~~0~~

~~0~~ ~~1~~ ~~1~~

~~1~~ ~~0~~ ~~0~~

~~0~~ ~~1~~ ~~1~~

~~1~~ ~~0~~ ~~1~~

~~I_{O/M}~~

~~0~~

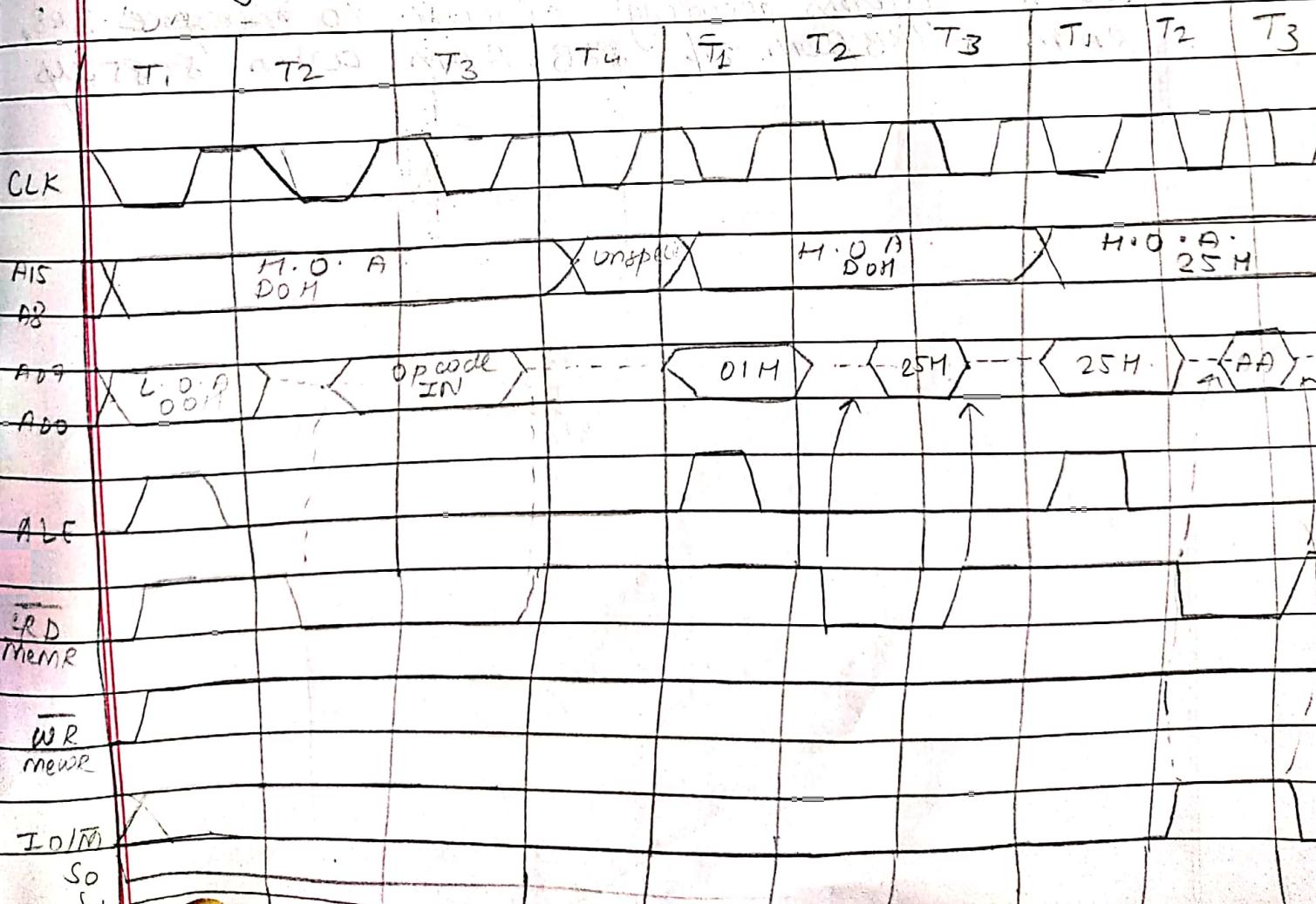
~~0~~

~~0~~

~~1~~

~~1~~

Timing diagram for IN 25H → 2 byte



* Design an address decoding circuit to interface 4KB RAM1, 4KB ROM & 4KB RAM2 with starting address 9000H.

⇒ Available address lines is $(A_0 - A_{15})$ 16 address lines.

⇒ For 4KB RAM1,

$$\text{Size} = 4\text{KB}$$

$$= 2^{10} \times 2^2 = 2^{12}$$

∴ No. of address lines required is 12 ($A_0 - A_{11}$)

⇒ For 4KB ROM $= 2^{10} \times 2^2 = 2^{12}$

∴ No. of address lines required is 12 ($A_0 - A_{11}$)

⇒ For 4KB RAM2,

$$\text{Size} = 4\text{KB}$$

$$= 2^{10} \times 2^2 = 2^{12}$$

∴ No. of address lines required is 12 ($A_0 - A_{11}$)

selection lines:

Memory	Address	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
RAM1	Start: 9000H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	End: 9FFFFH	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
ROM	Start: 5000H	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	End: 5FFFFH	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RAM2	Start: 6000H	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	End: 6FFFFH	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

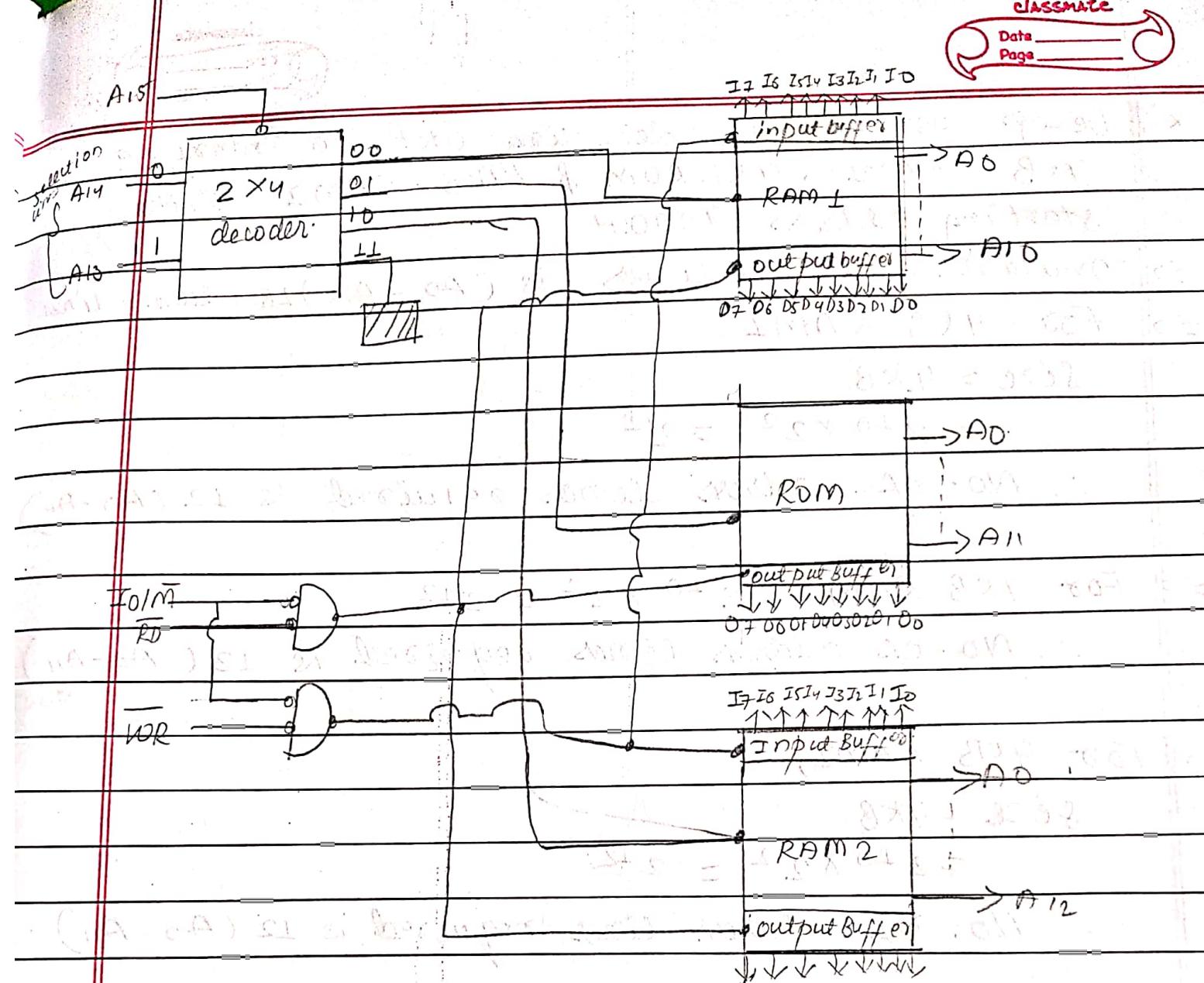


Fig: Address Decoding circuit