

Digital

- nothing is missing values
- does not impact

Discrete
Values

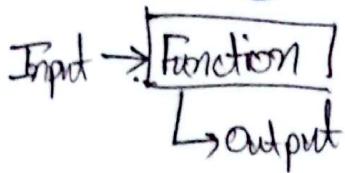
- * Combinational and Sequential
- * Half adder and Full adder

Full adder eqⁿ:

$$\begin{aligned}
 S &= \overline{A} \overline{B} C_{in} + A \overline{B} \overline{C}_{in} \wedge A \overline{B} C_{in} \wedge A B C_{in} \\
 &= \overline{A} (\overline{B} C_{in} \wedge B \overline{C}_{in}) \wedge A (\overline{B} \overline{C}_{in} \wedge B C_{in}) \\
 &= \overline{A} (\overline{B} \oplus C_{in}) \wedge A (\overline{B} \oplus C_{in}) \\
 &= A \oplus B \oplus C_{in}
 \end{aligned}$$

$$\begin{aligned}
 C_{out} &= \overline{A} B C_{in} \wedge A \overline{B} C_{in} \wedge A B C_{in} \wedge A B \overline{C}_{in} \\
 &= \overline{A} C_{in} (\overline{B} \wedge B) \wedge \overline{A} B C_{in} \\
 &= A B (C_{in} \wedge \overline{C}_{in}) \wedge C_{in} (A \oplus B) \\
 &= A B \wedge (A \oplus B) C_{in}
 \end{aligned}$$

Analog



Design

For Binary Adder,

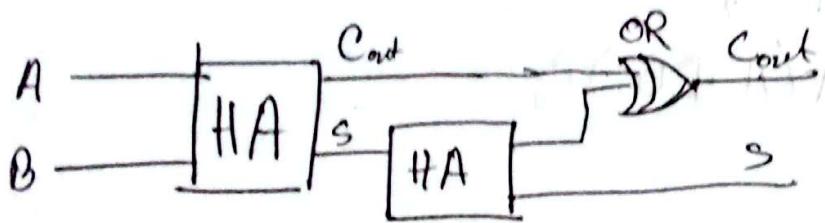
$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i B_i \wedge (A_i \oplus B_i) C_i$$

C-2, w-1

26/05/29

FA using HA



Block diagram

A

ALU

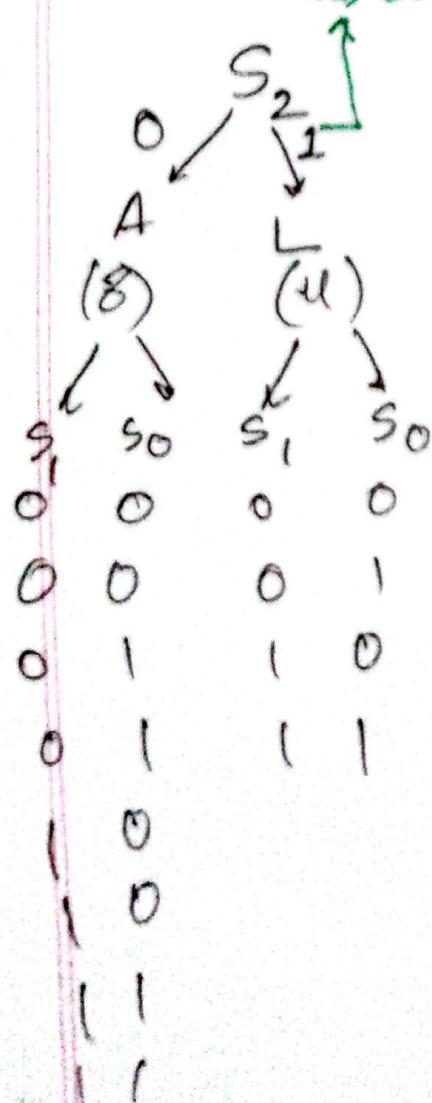
Arithmetical and Logic Unit

8 Arith + 4 Logical

$$* S_{Sub} = A + B' + 1$$

$$Sub + \text{Carry} = A + B'$$

This can change



* Q will have S_2 , S_1 , and S_0 and F
we will have to find A, B, X, Y, Z

1. Transfer

.

2. Transfer with Cout (A , $Cout = 1$)

For $A = 0010$

$$\begin{array}{r} + 1111 \\ \hline 10001 \\ + 1 \\ \hline 10010 \end{array}$$

$Cout = 1$: $A = 0010$

1

C-3, W-2

DSD Theory

02/07/25

* Watch ~~ppt~~ vid vids in classroom
lectures for quiz

Quiz - 1: 10/07/25 → can change
1.1, 1.2, 1.3, 1

Syllabus: ~~Up to vits~~

* Decoders: For n input lines

there will be $\Rightarrow 2^n$ output lines

→ Implementation using decoders

* Better to use min term than max/product term

* Find min term → use OR gates

* Encoders: For 2^n input lines

there will be n output lines

* Priority Encoders: When multiple inputs are present
a function decides priority

* $V=0$ means OFF state

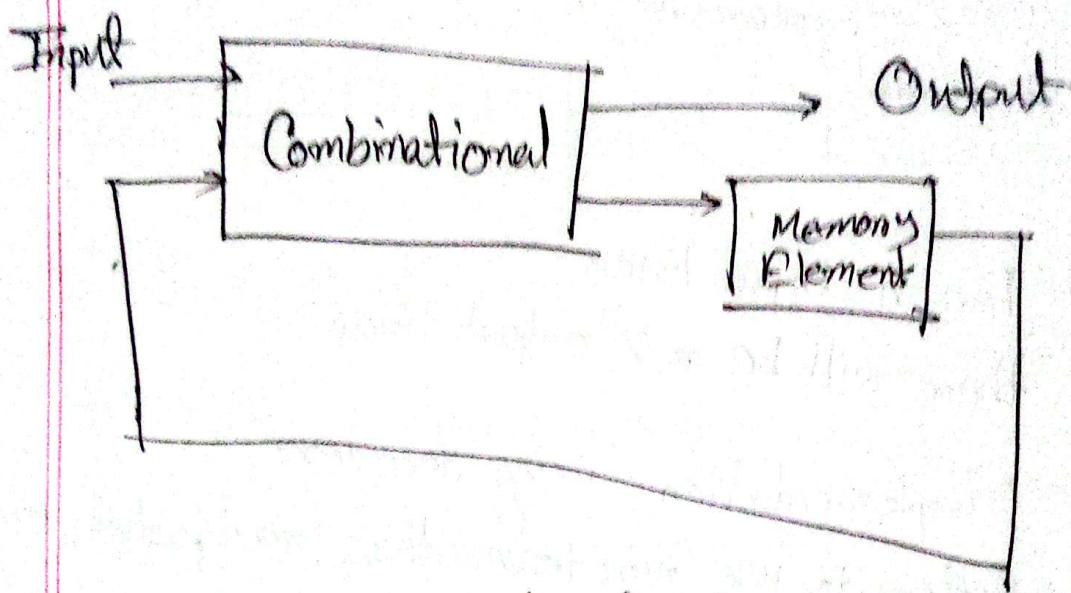
$V=1$ " ON "

* Absorption law: $A+B = A+\bar{A}B$
 $= A \cdot A + \bar{A}B$

C-4, ID - 4

11/07/25

Sequential Circuit



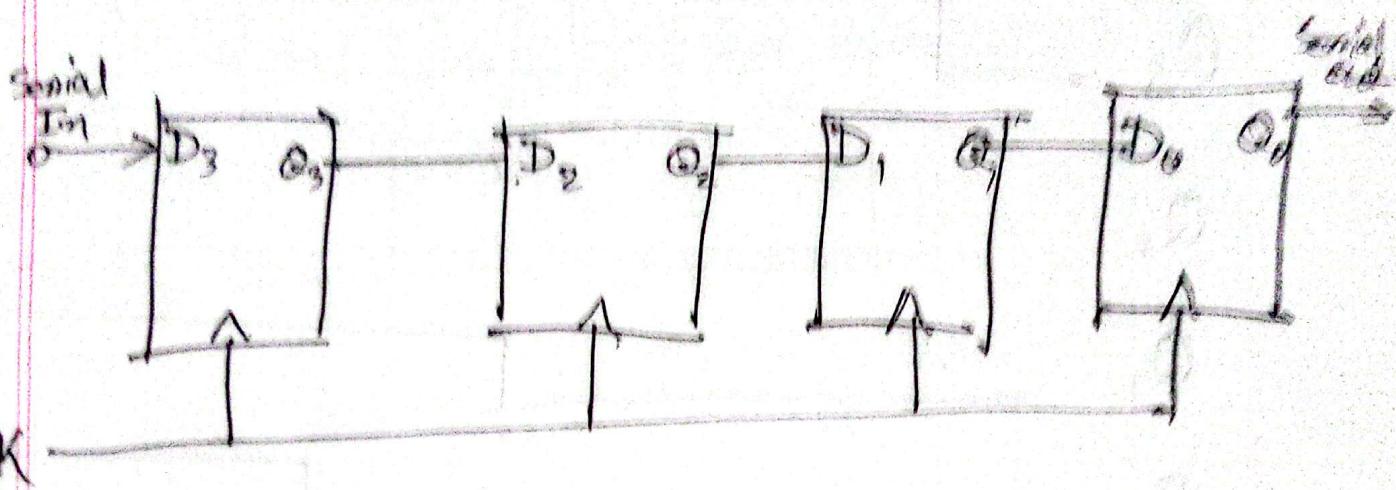
* Aynchronous: No clock

* Synchronous: Clock

* Registers: 1 bit memory cell

Synchronous

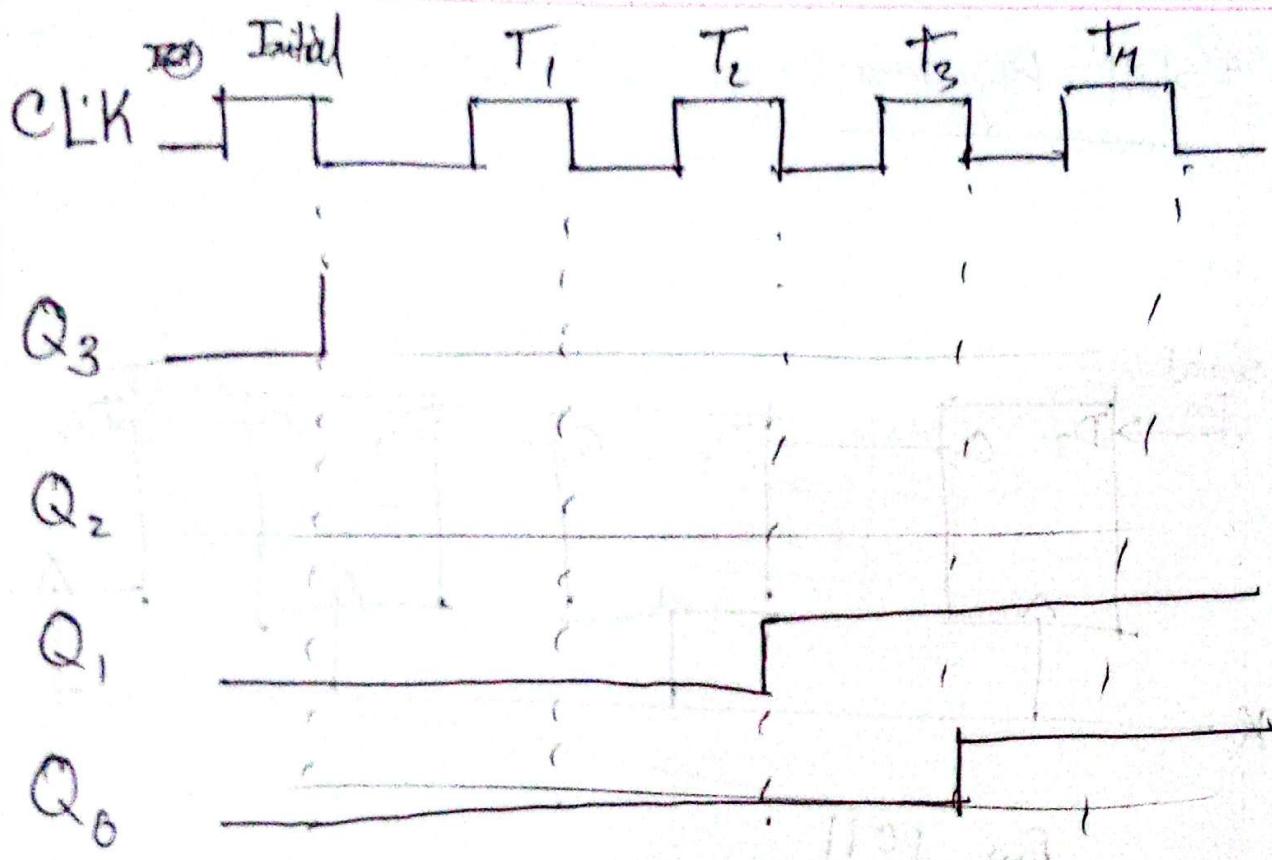
*Shift Registers:



For 1011

CLK	Q_3	Q_2	Q_1	Q_0
Init	0	0	0	0
(LSB) T_1	1	0	0	0
T_2	1	1	0	0
T_3	0	1	1	0
(MSB) T_4	1	0	1	1

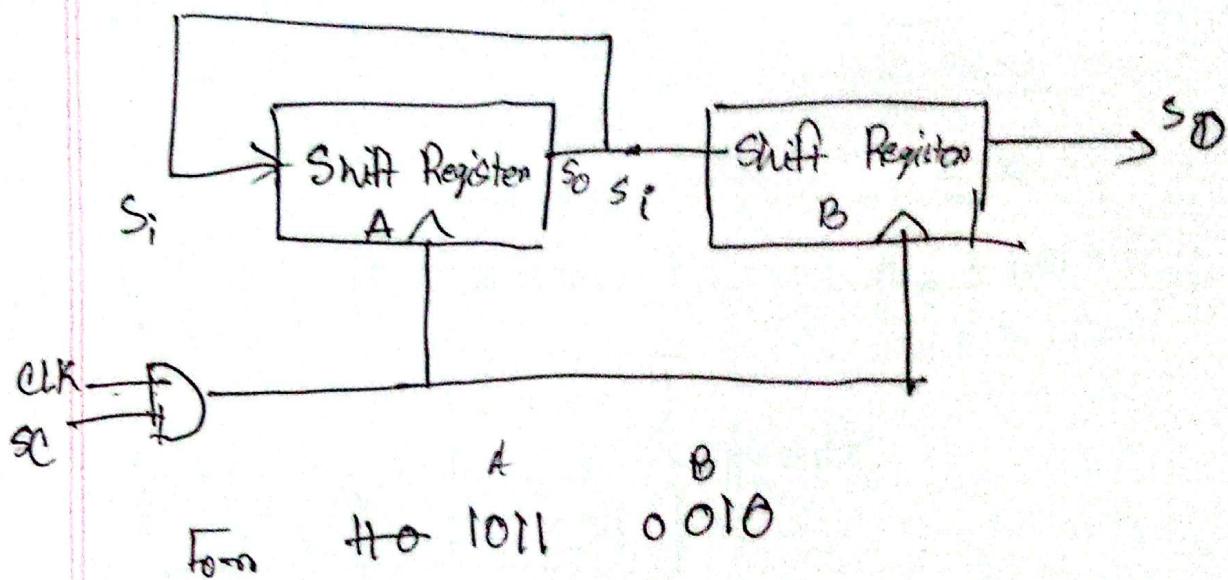
Initially its 0



* Serial Transfer: 1 bit at a time

* Parallel ... : Much n " " "

*Serial Transfer:



<u>Timing Pulse</u>	<u>S.R A</u>	<u>S.R B</u>	S_o
T_{out}	1 0 1 1	0 0 1 0	0
T_1	1 1 0 1	1 0 0 1	1
T_2	1 1 1 0	1 1 0 0	1
T_3	0 1 1 1	0 1 1 0	0
T_4	1 0 1 1	1 0 1 1	0

C-5, 10-4

16/07/25

#Find 1101×0011 using Modified Booth's Algorithm

$i+1$	i	$i-1$	Operation
0	0	0	$0 \times M$
0	0	1	$+1 \times M$
0	1	0	$+1 \times M$
0	1	1	$+2 \times M$
1	0	0	$-2 \times M$
1	0	1	$-1 \times M$
1	1	0	$-1 \times M$
1	1	1	$0 \times M$

$$m = 1101$$

$$n = 0011$$

$$-m = 0011$$

$$2m = 11010$$

$$-2m = 0110$$

Not always
needed

left shift

Operation	A	Q	Q_{-1}	SE $\frac{1}{2} \oplus Q$ $\frac{1}{2} \oplus Q_{-1}$
-1xM	0 0 0 0	0 0 1 1	0	
+1xM	0 0 1 1	0 0 1 1	0	2
R.S 1	0 0 0 1	1 0 0 1	1	1
R.S 2	0 0 0 0	1 1 0 0	1	*
+1xM	1 1 0 1			
R.S 1	1 1 0 1	1 1 0 0	0	*
R.S 2	1 1 0 0	1 1 1 0	0	*
	1 1 1 1	0 1 1 1	0	*
Final Product				

Lect - 5 (ROM & PLA)

ROM:

Size = $2^n \times m$ \rightarrow n = input
 m = output & No. of bits per word
 2^n = Total number of words
 $0 - (2^n - 1)$ = Total distinct addresses

For 256 bit ROM, word size = 8 bits

$$m = 2^n = \frac{256}{8} = 32$$

$\therefore 32 \times 8$ bit ROM

$$2^5 \times 8$$

$$\therefore n = 5$$

$$\therefore m = 8$$

$$\therefore 2^5 = 32$$

No. of bits per word = 8

\therefore Distinct Addresses = $0 \rightarrow 31$

To the ROM

Address (3 bits) word word(8 bit)

0th word \rightarrow 0 0 000 1

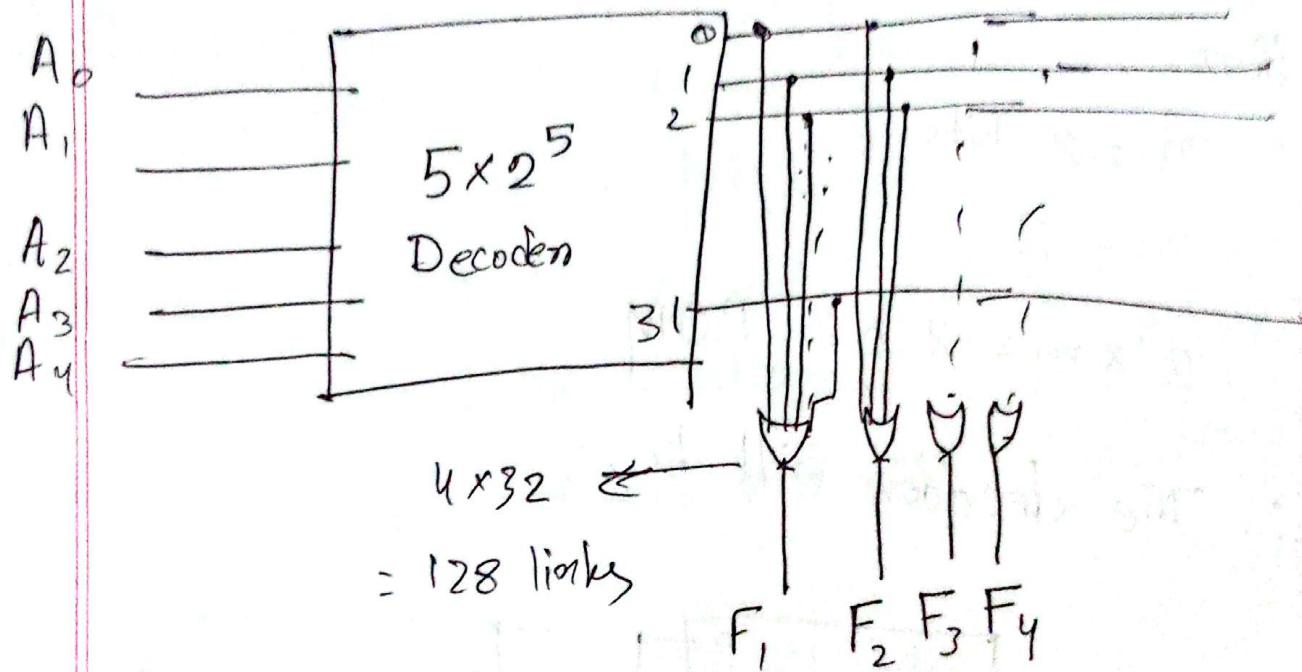
31st word \rightarrow 1 1 1 1 1

#2048 bit ROM,
word size = 4, 8 } Practice both

For a 32×4 bit ROM ($2^5 \times 4$ bit)

$$\therefore n = 5$$

$$m = 4$$



* Wires can be both ~~on~~ fuse or broken

$$\# F_1 = \sum(0, 1, 2)$$

$$F_2 = \sum(3)$$

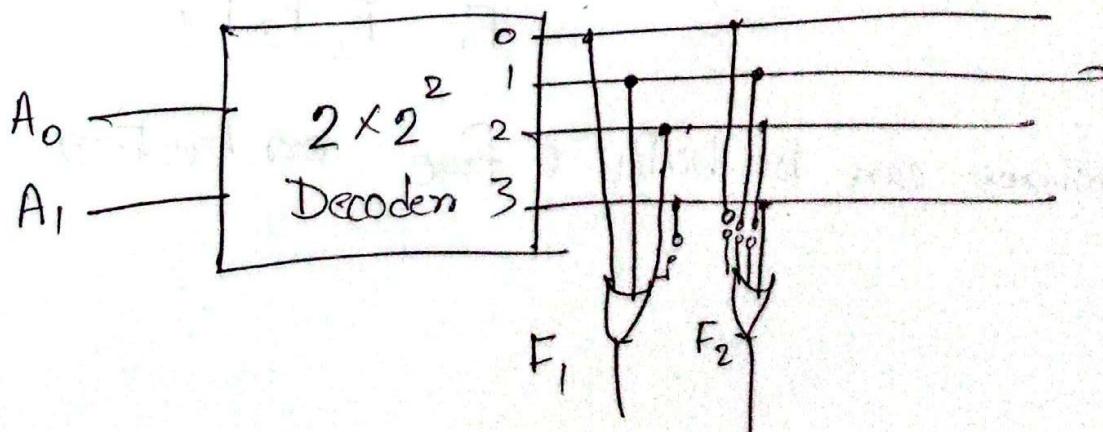
Hence,

$$n = 2 \text{ bits}$$

$$m = 2$$

$$\therefore 2^n \times m = 2^2 \times 2 \text{ ROM}$$

\therefore The decoders will be,



C-6, W-5

23/07/25

#ROM that squares

inp : 3

$$\text{Output} : (111)_2^2 = (110001)_2$$

↓
6 bits

+ But, last bit of inp = last bit of inp out

- and second bit is always 0

∴ Output is 4 bits that needs to calculated

#SelF: 3 bit cube, 2 bit cube
2 bit square

Types of ROMS

* Maskable: ROMs can be Masked

→ changing during ~~writing~~
Binary Operation

* Programmable: Can be changed using Hardware Program

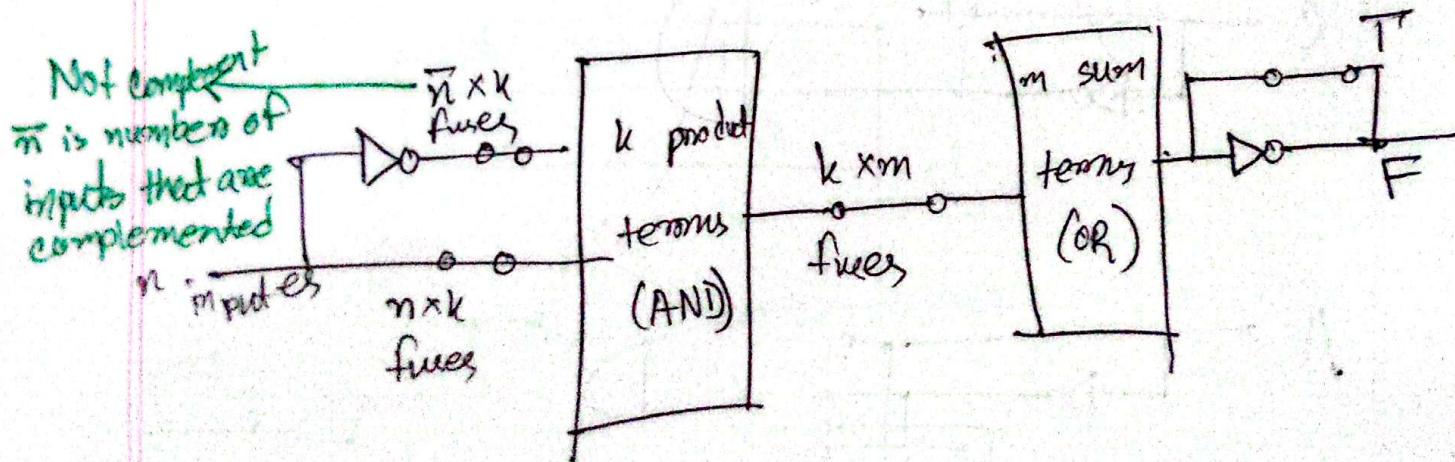
PLA

* Programmable Logic Array

→ Contains fixed architecture that contains programmable AND followed by n and programmable OR

* ROM decodes all lines
but PLA only uses the minterms required

↓
Partial Decoding

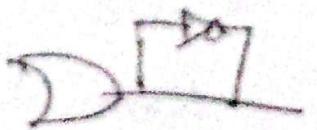
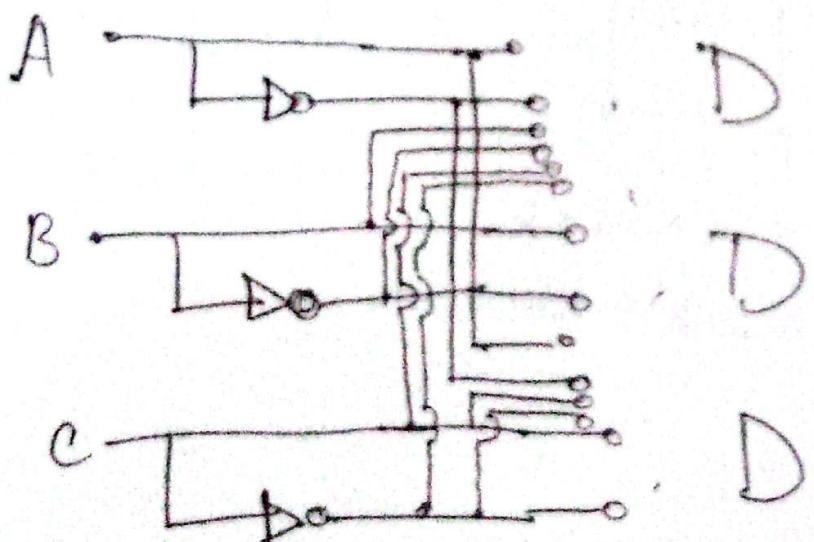
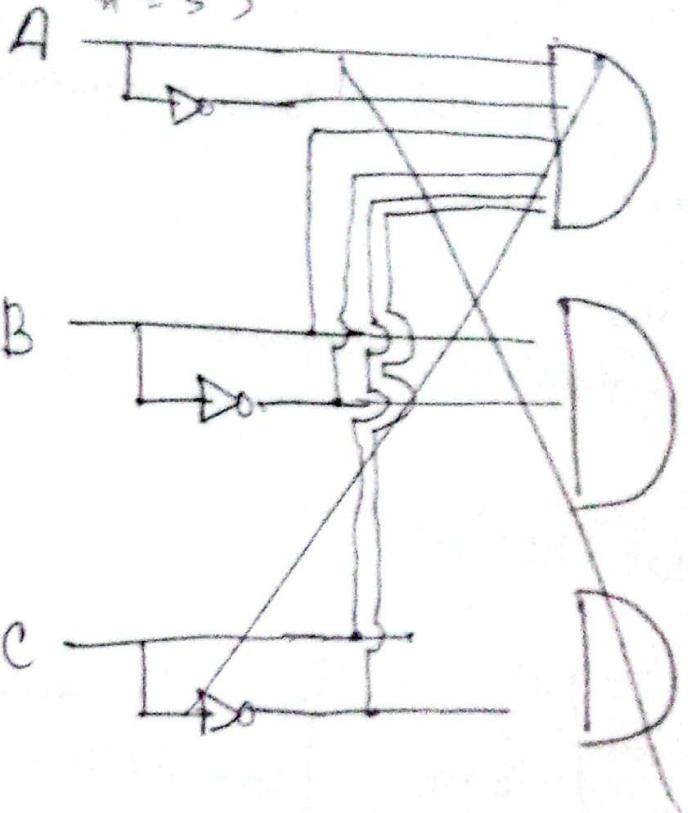


* Size of PLA = $n \times m \times k$

Num of Links = $2nk + km + m$

$3 \times 3 \times 2$

$$\begin{cases} n = 3 \\ m = 2 \\ \bar{n} = 3 \end{cases}$$



$$F_1 = \Sigma(4, 5, 7), F_2 = \Sigma(3, 5, 7)$$

	BC	00	01	11	10
A	0	0	0	0	0
	1	1	1	1	0

$$= A' + BC'$$

	BC	00	01	11	10
A	0	0	0	0	0
	1	0	0	1	0

$$= A'C' + A'B'$$

Count of common terms

$$\therefore F_1' F_2' = 0$$

$$F_1' F_2 = 0$$

$$F_1 F_2' = 0$$

$$\boxed{F_1 F_2 = 1}$$

Max count

Hence,

$$\left. \begin{array}{l} F_1 = AB' + AC \\ F_2 = AC + BC \end{array} \right\} \begin{array}{l} T = \text{True (bypasses Invert)} \\ F = \text{False (complemented)} \end{array}$$

Program Table:

Product Term	Inputs			Outputs	
	A	B	C	F_1	F_2
AB'	1	1	0	1	-
AC	2	1	-	1	1
BC	3	-	1	-	1

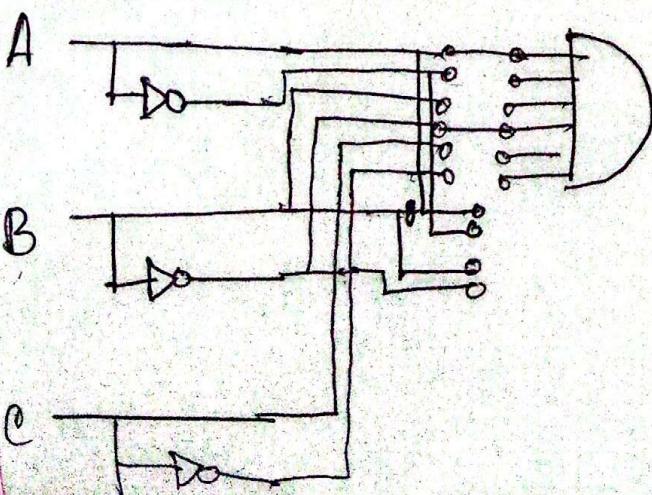
+ + $\xrightarrow{T/F}$ F_1 and F_2 are T
 F_1' and F_2' are F

Hence,

input, $n = 3$

product term, $k = 3$

output, $m = 2$



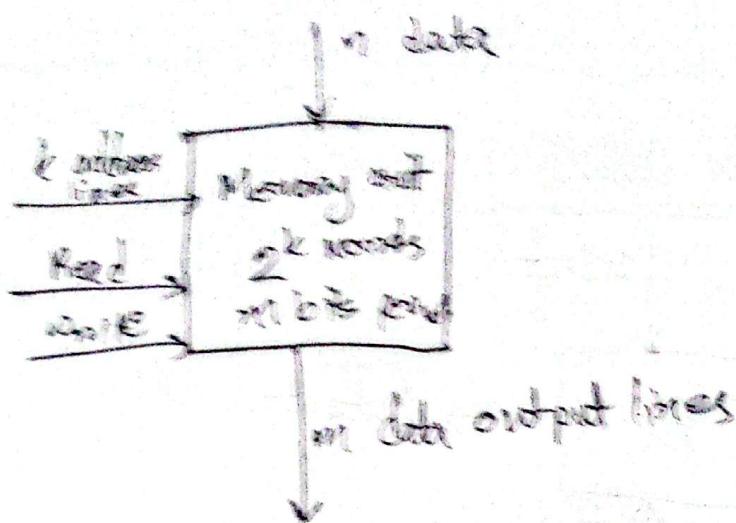
[Check slide]

C-7, 10-5

14/07/23

RAM (cont-6)

→ we randomly access it



$2^k \times m$ RAM size.

For $2^{10} \times 16$ RAM,

$$\text{Size} = 2^{10} \times 16 = 1024 \times 16 \text{ bits}$$

$$= 16384 \text{ bits}$$

$$= \frac{16384}{8} \text{ bytes}$$

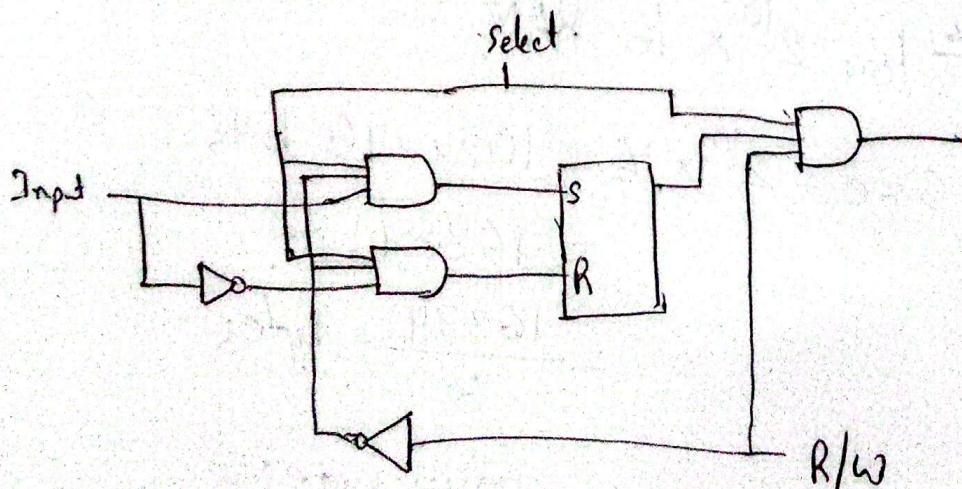
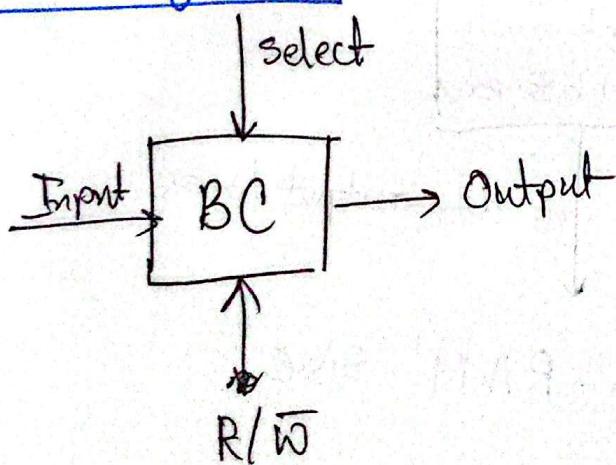
$$= 2048 \text{ bytes}$$

$$= 2 \text{ kB}$$

Read / Write :

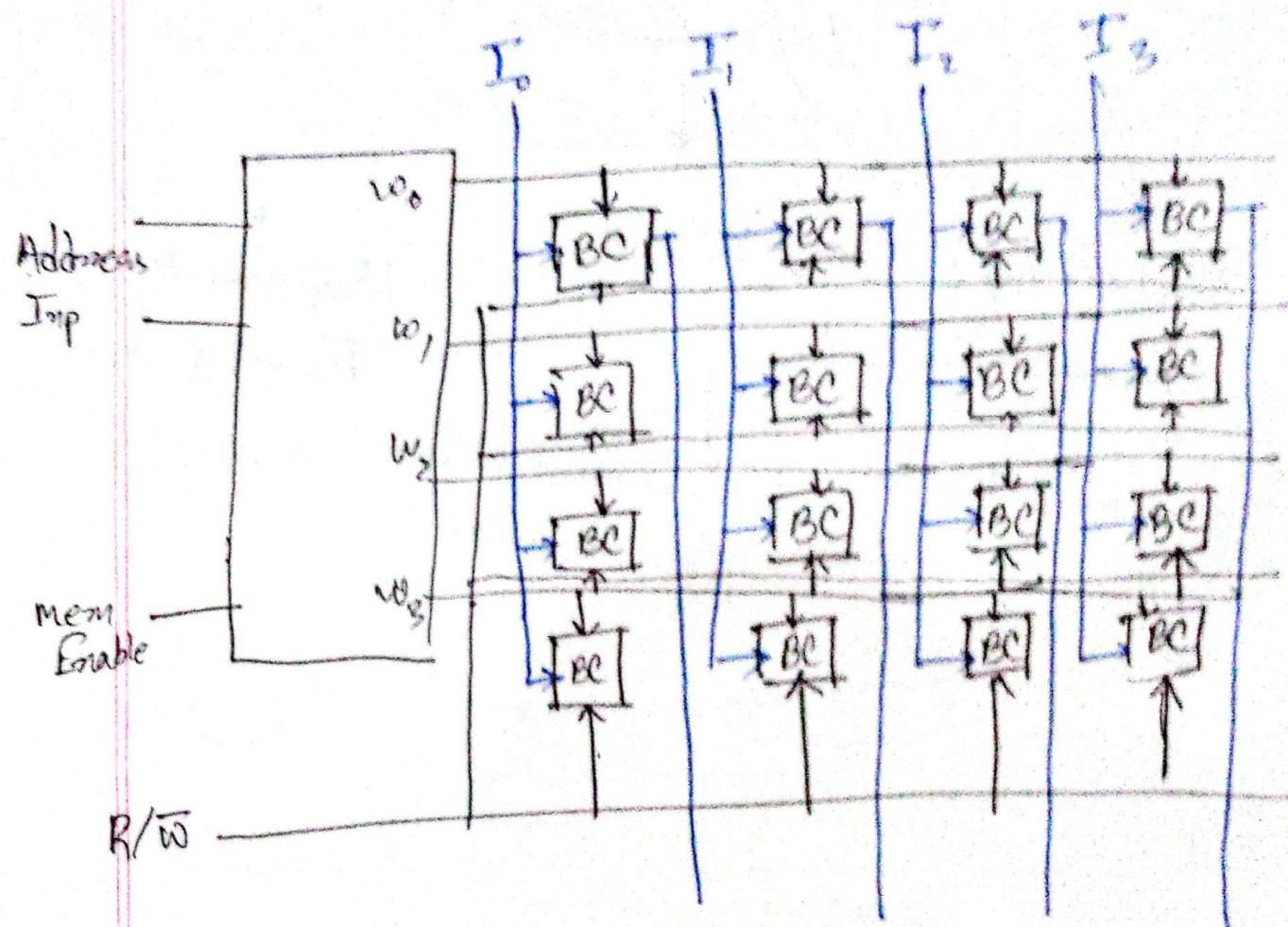
& SRAM and [Check Slide]
DRAM

RAM Memory Cell:



4×4 RAM = $2^2 \times 4$; $k = 2 \rightarrow$ input

$m = 4 \rightarrow$ output



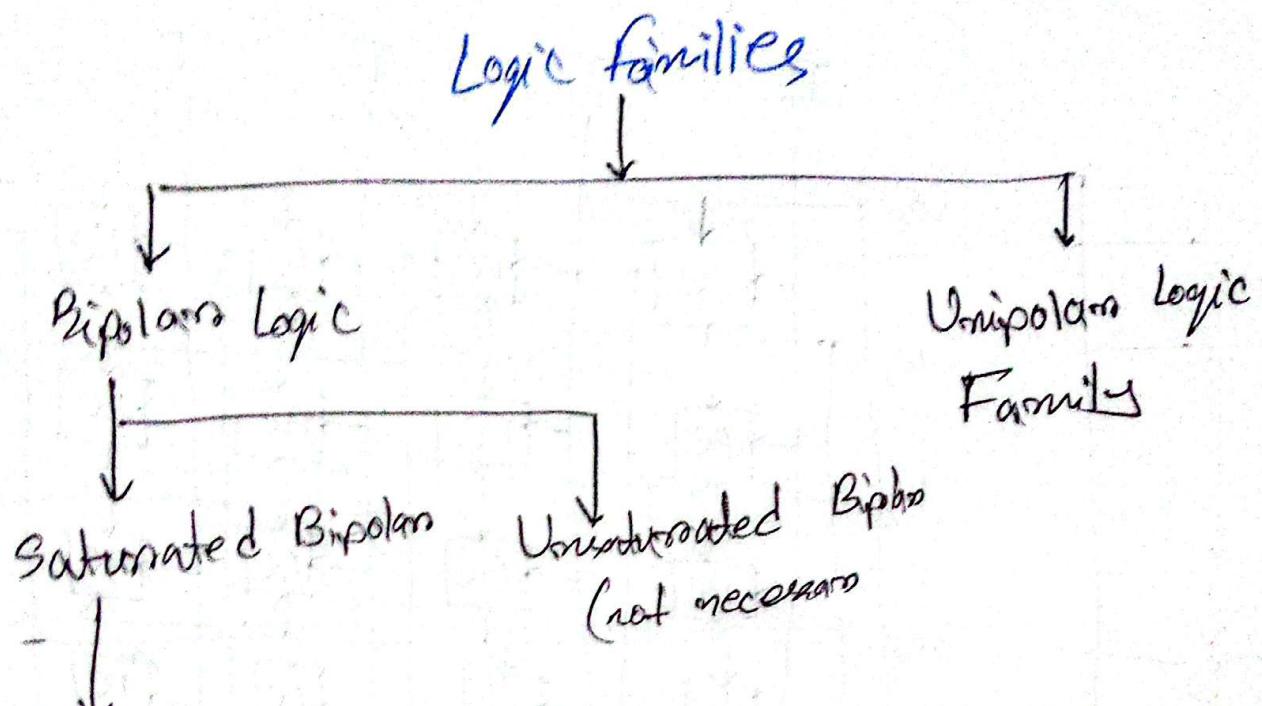
16×2

4×3

C-8, W-6

28/07/25

Lect - 2 (?)



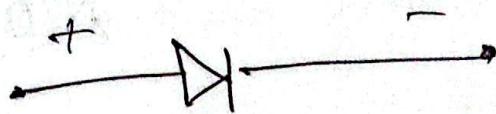
* Logic System:

Pos : High (4 V)
Low (0.2 V)

Neg : High (0.2 V)
Low (4 V)

* Diode Logic

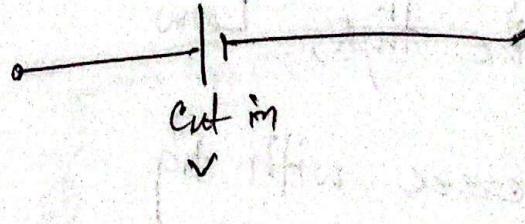
Unidirectional current flowing



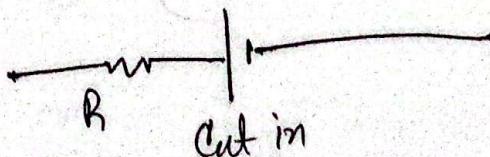
Ideal :



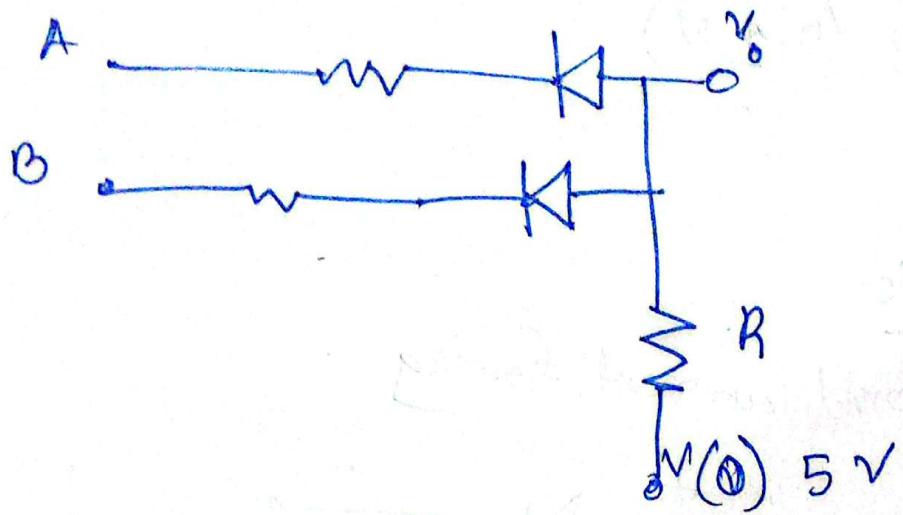
Constant :



Piecewise :



Prove the circuit works like OR gate in
Negative logic



Solⁿ: i) OR - T.T

ii) Define High, Low

iii) All cases with fig

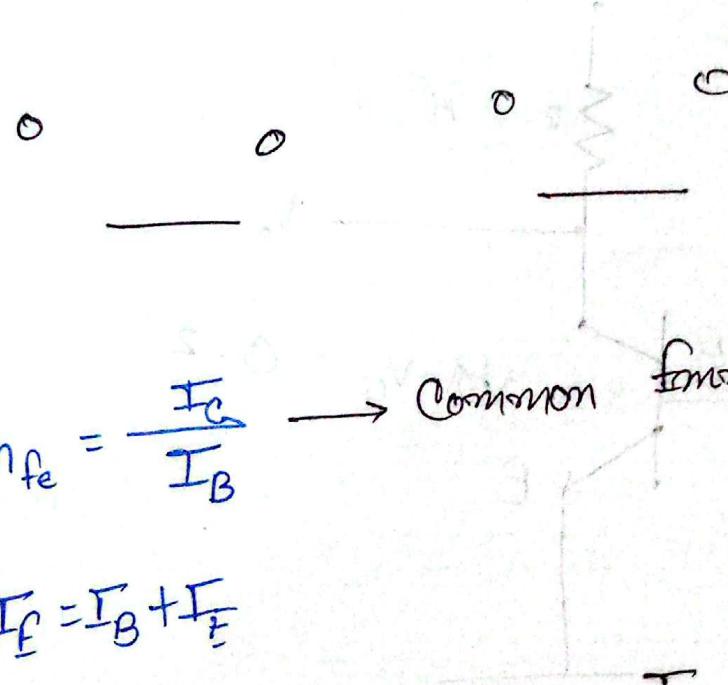
C-9, w-6

30/07/2025

Transistor

* Switch and Amplifiers

* PNP and NPN



$$+ h_{fe} = \frac{I_c}{I_B}$$

$$I_F = I_B + I_E$$

~~RP + RE Act:~~ $I_E \rightarrow I_E \propto I_B$

* At Saturation: $I_c \leq h_f * I_B$

$$V_{CE(\text{sat})} = 0.2 \text{ V}$$

$$V_{BE(\text{sat})} = 0.8 \text{ V}$$

* At cut in: $V_{BE(\text{cut})} = 0.5 \text{ V}$

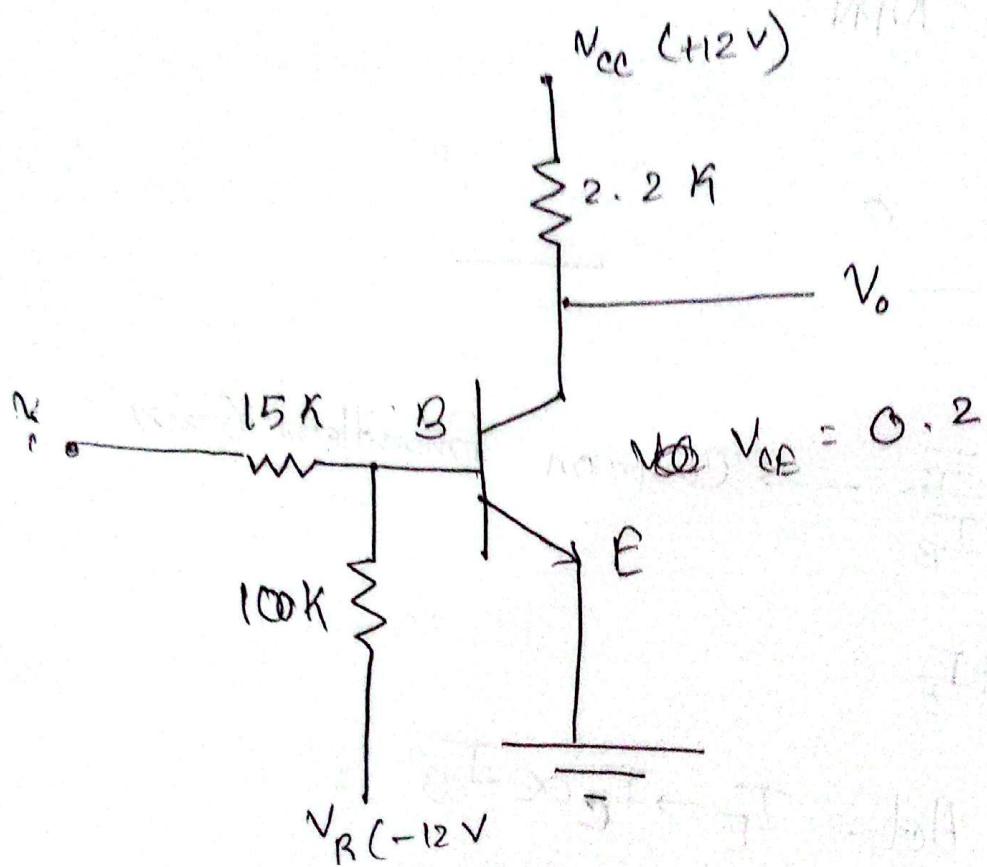
* At cut off: $V_{BE(\text{cutoff})} = 0 \text{ V}$

Q1

Transistorized NOT Gate: (a)

Prove its a " " "

for what value of h_{fe}



When $V_i = 12 \text{ V}$

①

$$\therefore I_1 = \frac{V_B - V_i}{15} = \frac{-0.8 + 12}{15}$$
$$= 0.75 \text{ mA}$$

, When $\therefore I_2 = \frac{0.8 - (-12)}{100}$

$$= 0.128 \text{ mA}$$

$$I_B = I_1 - I_2 = 0.622 \text{ mA}$$

$$I_C = \frac{12 - 0.2}{2 \cdot 2} = 5.36 \text{ mA}$$

E. 10, M. C

3/07/25

MOSFET

- * if channel \rightarrow Depletion
- else \rightarrow Enhancement

NMOS in enhancement:

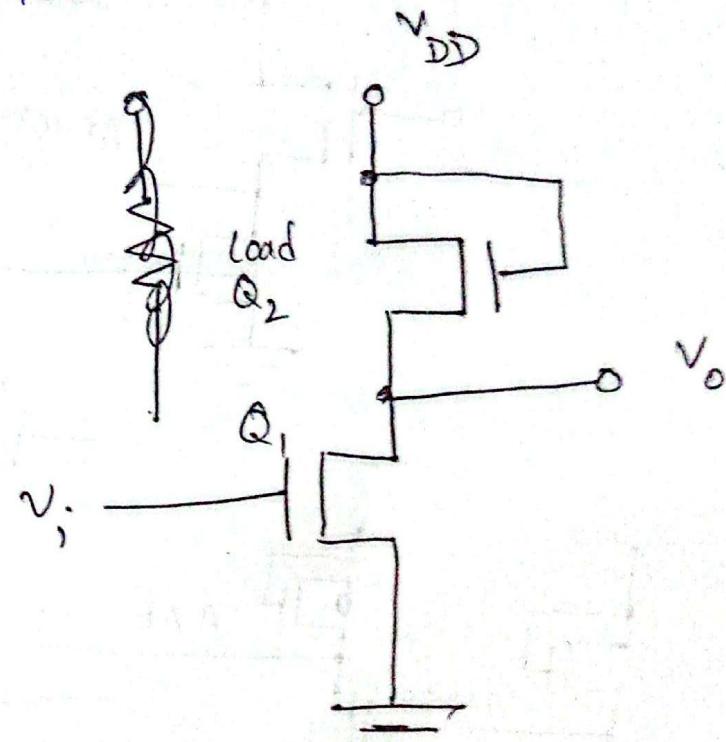
- if high on G then D, S short
- low " G " D, S open

PMOS in enhancement:

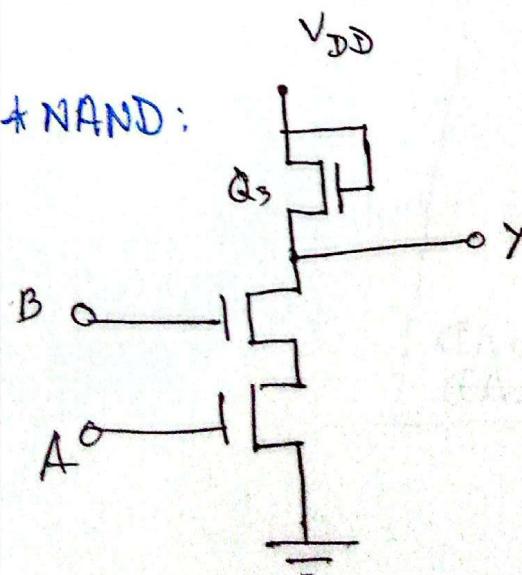
- if high on G then D, S open
- low on G " D, S short

Logic Gate Using MOSFET.

* NOT using NMOS



* NAND:



CMOS: Mixing Multiple PMOS and NMOS

Quiz-2: 04/08/23

3, 2, 4, 5, 6

Interface

* Process of connecting device for exchanging information

* Isolated I/O → IN [Address]
OUT [Address]

* Memory Mapped I/O → LOAD
STORE

* Parallel Transfer Method:

- Simple I/O
- Simple Stroke I/O
- Single-handshake I/O
- Double-handshake I/O

→ Simple I/O

- Assumes its always on ready
- Limitations: Device may not always be ready
 - ↳ Assumptions are its limitation

→ Simple strobe

- Data is present only a certain time
- Strobe signal notifies about data presence
- Limitations: CPU remains busy, Receiver may be already ^{not} available, Sender doesn't know when to send next data, extra data frame fees
- Better of to form local cache

→ Single Handshake

- ^{Receiver} uses ACK signal to notify reception

- Limitations: Sender still does not consider whether the receiver is ready to receive, multiple devices can cause confusion

→ Double Handshake:

- Strobe: Data is sent when strobe is getting turned off