Some Important Concepts/Problems - DEC

1) A sequential circuit with two T-flipflops, A & B, one i/p, x, and one output, y, is specified by the following next-state, output eqns.

$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$

(a) Draw the logic diagram of the circuit

(b) List the state table for the sequential ckt.

(c) Draw the corresponding state diagram.

Ans:- The characteristic eqn of 'T' flipflop

$$Q_{n+1} = Q(t+1) = T\bar{Q}_n + \bar{T}Q_n \quad \text{— (1)}$$

Consider the sequential circuit. It has 'x' flipflops A and B, one i/p 'x' and one o/p 'y'

It can be described algebraically by two i/p equations and o/p eqn as given in the question

$$T_A = Bx \quad \text{— (2)}$$

$$T_B = x \quad \text{— (3)}$$

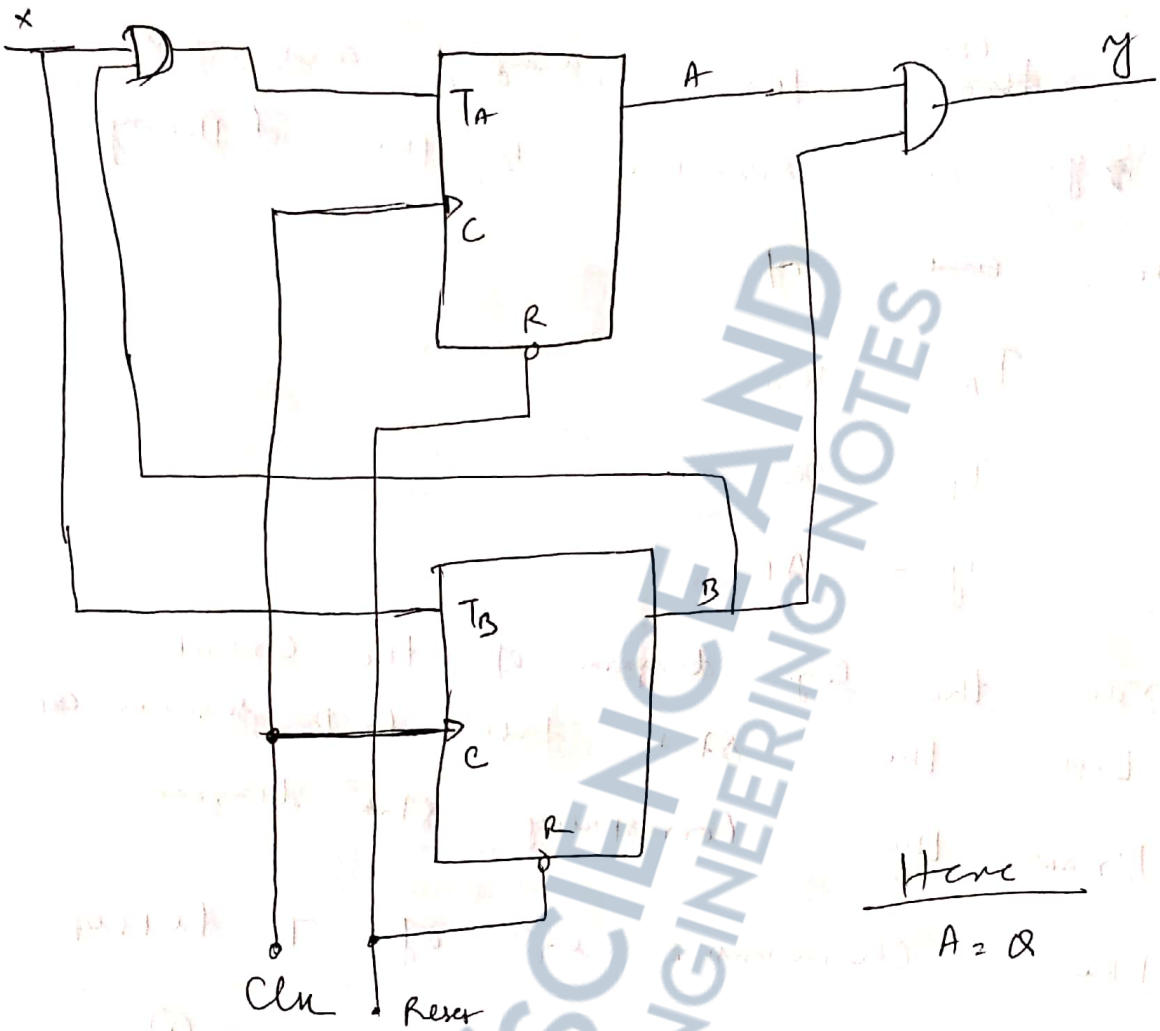
$$y = AB \quad \text{— (4)}$$

The o/p of T flip flop 'A' is given by

$$Q(t+1) = T'Q + TQ' \quad \text{—(5)}$$

(Using eqn (1))

(a)



Here

$A = Q$ (O/P)

So, ~~$A(t+1) = (Bx)' A$~~

For T_A FF eqn (5) becomes

$$A(t+1) = T_x' A + T_A A'$$

$$= (Bx)' A + (Bx) A'$$

(From eqn (2))

$$= (B' + x') A + A' Bx$$

$$A(t+1) = AB' + Ax' + A'Bx \quad \text{—(6)}$$

For T_B F/F eqn (5) becomes,

$$B(t+1) = T_B' B + T_B B' \quad \text{(From eqn (2))}$$

$$B(t+1) = x' B + x B' \quad \text{---(7)}$$

$$B(t+1) = x \oplus B$$

From eqn (6) & (7) the state table can be formed.

Ques:- When A=0, B=0, x=0

(Next state) $A(t+1) = 0 \cdot B' + 0 \cdot x' + 1 \cdot 0 \cdot 0 = 0$

$B(t+1) = x \oplus B = 0 \oplus 0 = 0$

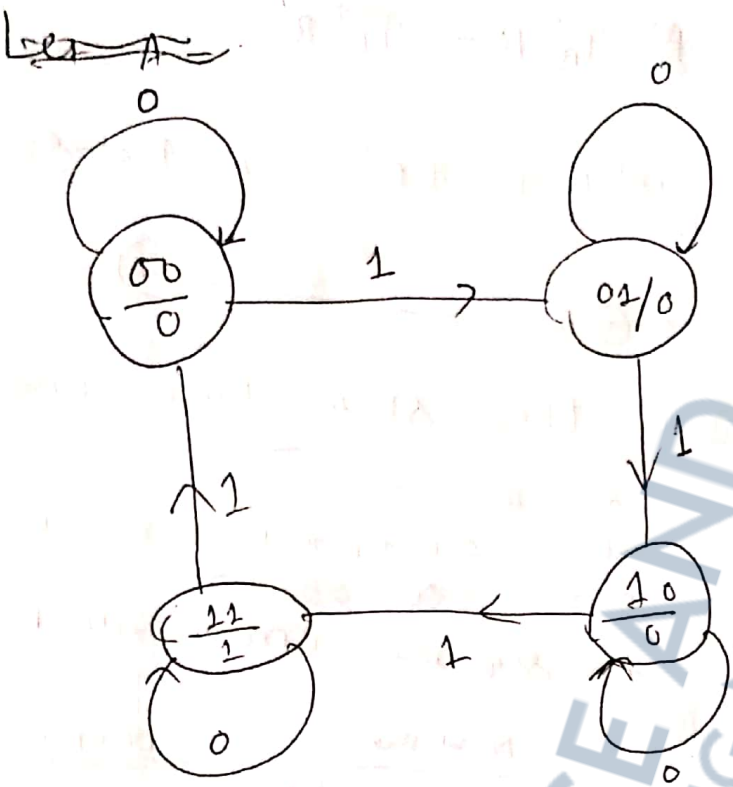
Table :- State table for sequential circuit with T F/Fs

Present State A B	Input x	Next State A	Next State B	Output y (y=AB) [Present A, B state of]
0 0	0	0	0	0
0 0	1	0	1	0
0 1	0	0	1	0
0 1	1	1	0	0
1 0	0	1	0	0
1 0	1	1	1	0
1 1	0	1	1	1
1 1	1	0	0	1

→ Since the O/P y is independent of I/P(x) and function of present states A & B only it is a Moore machine.

(∵ y = AB)

State diagram

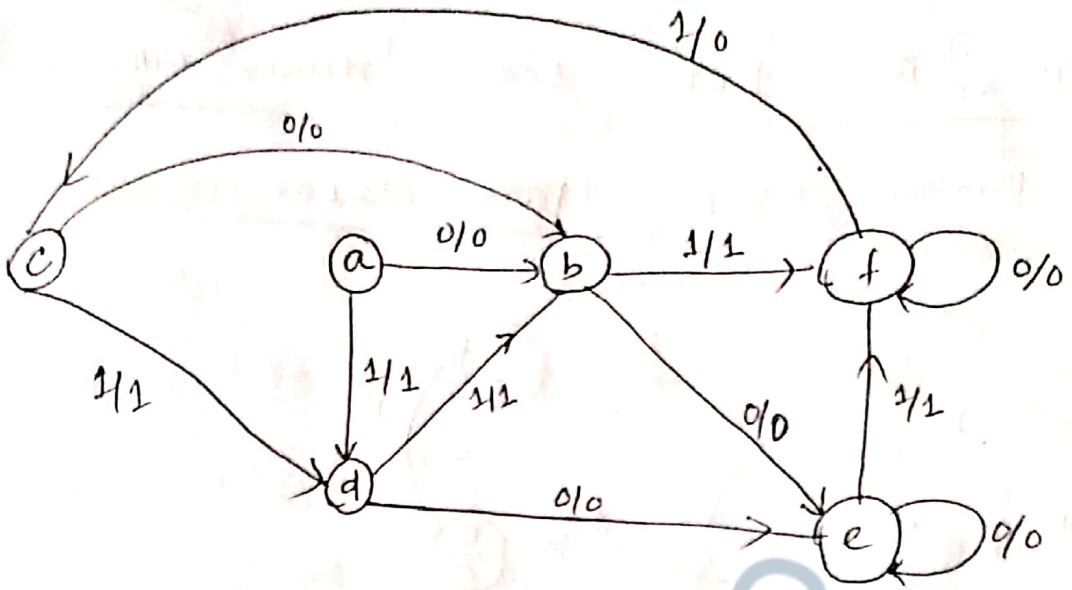


Note:- o/p is product of A & n in each circle.

2) State Reduction:-

The state reduction technique basically avoids the introduction of redundant states. The reduction in redundant states reduces the number of flipflops and logic gates, reducing the cost of the final circuit.

Two states are said to be equivalent if every possible set of inputs generate exactly the same output and the same next state. When two sets are equivalent, one of them can be removed without altering the i/p-o/p relationship. Consider the sequential circuit whose state diagram is shown in figure.



PS	NS		O/P	
	x=0	x=1	x=0	x=1
✓a	b	d	0	1
✓b	e	f	0	1
✓c	b	d	0	1
d	e	b	0	1
✓e	e	f	0	1
f	f	c	0	0

→ Since a & c P. States have same N. State & O/P, c can be removed & whenever there is c we can replace with 'a'.

→ Similarly b & e are equivalent. So 'e' can be removed and replaced by 'b'.

Reduced State Table

PS	NS x=0	NS x=1	O/P x=0	O/P x=1
a	b	d	0	1
b	b	f	0	1
d	b	b	0	1
f	f	a	0	0

3) Design a '101' sequence detector with overlapping permitted Using Moore Machine

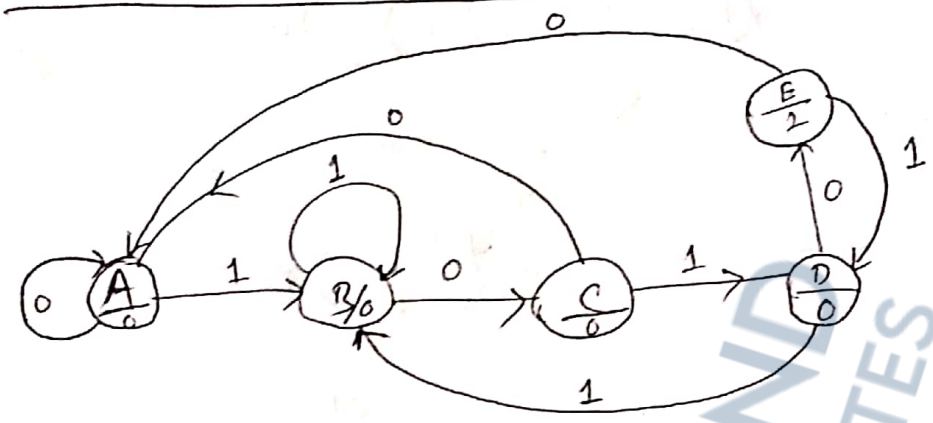


Fig 1:-

A → Nothing detected

B → '1' detected

C → '10' detected

D → '101' detected

E → '1012' detected.

For design of Moore type of circuit, the same steps are to be followed. The state diagram and state table of a Moore type sequence detector to detect the sequence 1012 are shown in figure 1 & Table 1.

In the Mooretype state diagram the O/Ps are written inside the circle below the state name. The state diagram is drawn in normal way. The machine will be in state D if the last three bits are 101. If the next bit is a '0', the last 4 bits

will be 1010 which is a valid sequence. So the machine o/p is 1, but to utilize overlapping it cannot goto 'C' because the o/p of C is '0'. Instead it will go to a new state 'E' where output is equal to 1.

While at E, if the next bit is a '0' the last five bit becomes 10100. So the machine goes to state A to restart the detection. If the next bit is 1, the last five bits become 10101. So to utilize the last 3 bits i.e 101 it goes to state D. Once, the more type state diagram & state table are drawn, the Moore circuit can be designed following the normal procedure

State Table

PS	NS		O/P
	x=0	x=1	
A	A	B	0
B	C	B	0
C	A	D	0
D	E	B	0
E	A	D	1

4) Draw the state diagram a state table for Moore type sequence detector to detect the sequence 110 (with overlapping permitted) (8)

Ans :-

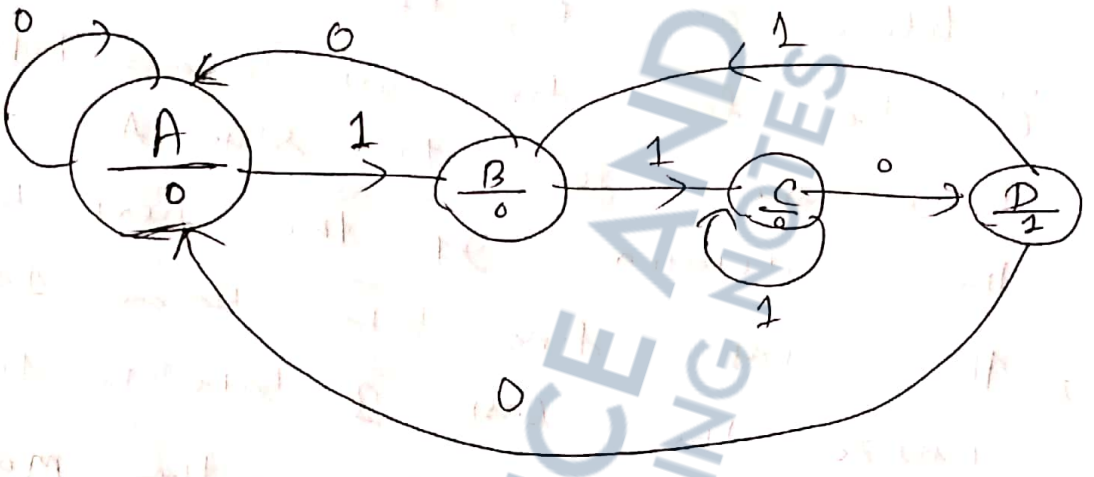


Fig 2 :-

State Table :-

PS	NS		O/P
	X=0	X=1	
A	A	B	0
B	A	C	0
C	D	C	0
D	A	B	1

The Moore type state diagram and state table of sequence detector to detect the sequence 110 is shown in fig. 2. The state diagram is drawn in the normal way. The machine is in 'C' state when the

When the last two bits received are 11. If the next bit is a '0', the last three bits become 110 which is a valid sequence, hence O/p is 1. But the machine cannot go to state A to restart the detection process because state A O/ps a '0'. So the machine goes to a new state 'D' and O/p is 1.

While at 'D' if next bit received is a '0', the last 4 bits will be 1100. So the machine goes to state A to restart the process. If the bit received is a '1', then the last four bits will be 1101. The machine goes to state 'B' to utilize the last bit.

5) The Content of a 4 bit register is 1101 Initially. The register is shifted six times to the right with serial i/p being 101101. What is the Content of register after each shift.

Ans:

Initially →

After Clock pulse

→ 0

101101

	Q0	Q1	Q2	Q3	After Clock pulse
1 →	1	1	1	0	→ 1
0 →	0	1	1	1	→ 2
1 →	1	0	1	1	→ 3
1 →	1	1	0	1	→ 4
0 →	0	1	1	0	→ 5
1 →	1	0	1	1	→ 6

Ans:- 2 0 1 2

6) The groups of bits 111 000 is parallely taken into a 6 bit shift register with initial state 101101 and the bits will be taken out serially. How many clock pulses are required for the shift register to have 000111 at its flipflop outputs.

Initial State: — 10101 ↓ Parallel in (11)

Ans :-

New Content of Six register → 111001 → One clock pulse required.

To Collect the data serially, we have to enter series of '0' (zeros)

Clock Pulse			(111001)
1	0 →	0 1 1 1 0 0	
2	0 →	0 0 1 1 1 0	for parallel storing
3	0 →	0 0 0 1 1 1	for shifting.

∴ Total clock pulse required = 1+3 = 4

∴ After 4 clock pulses, the content of F/Fs becomes 000 111.

7) An 8-bit shift register has the binary equivalent of the decimal number 86 stored in it. What are the base-10 equivalent contents of the register after the following operations have been performed? For each case, assume the same initial state given.

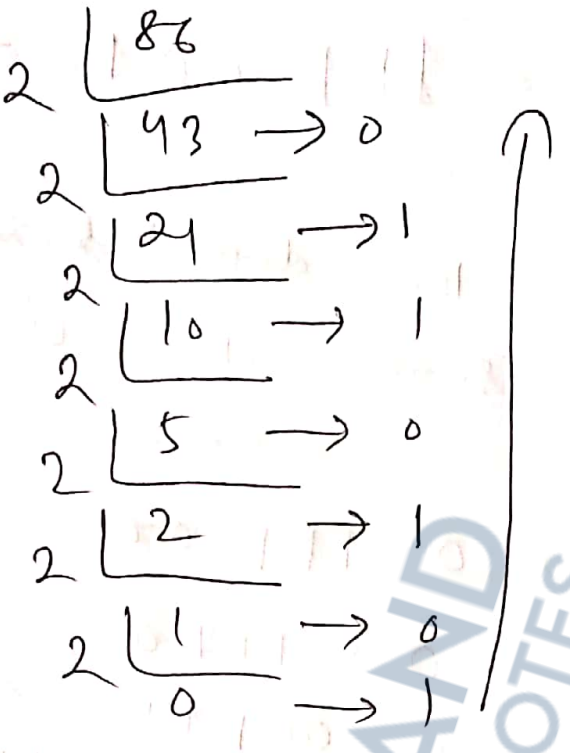
(i) Shift right 1

(ii) Shift left 1

(iii) Shift right 2

(iv) Rotate right 2

Ans :-



$(86)_{10} = (\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ )_2$ in 8 bits.

(i)

Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	1	0	1	0	1	1	0

Shift Right 1

with '0' entered at Q_0. Rest shifted one place right

$0 \rightarrow 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1 = (43)_{10}$

$2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$(32 + 8 + 2 + 1) = 43$

(ii) Shift left 1

with '0' entered at Q_7.

Rest shifted one place left

$1\ 0\ 1\ 0\ 1\ 1\ 0\ 0 \leftarrow 0$

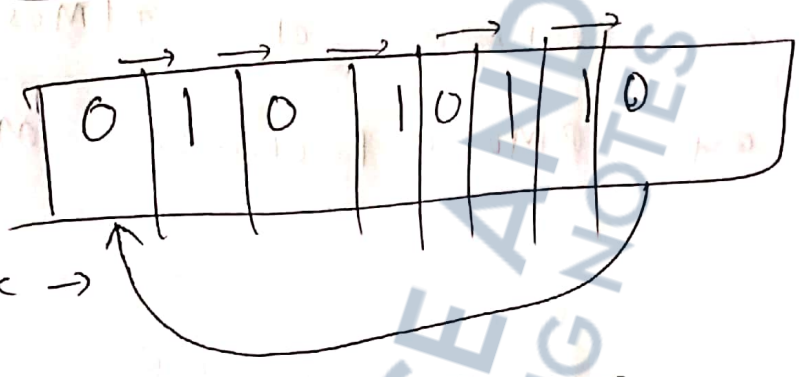
$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$(128 + 32 + 8 + 4) = (172)_{10}$

(iii) Shift Right 2

$$00\ 0\ 1\ 0\ 1\ 0\ 1 = (21)_{10}$$

(iv) Rotate Right 2



1st rotate →

After 1st Rotate → 0 0 1 0 1 0 1

2nd rotate

After 2nd rotate → 1 0 0 1 0 1 0

$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$= 128 + 16 + 4 + 1$

$= (149)_{10}$ (Ans)

CMOS Gates:-

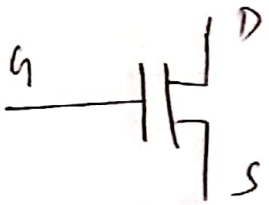
→ MOSFET → Metal Oxide Semiconductor Field Effect Transistor.

→ CMOS → Complementary Metal oxide Semiconductor FET FET

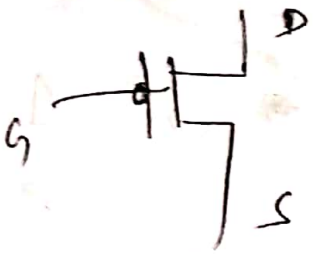
→ It is a Combination of NMOS (N-Channel MOSFET) and PMOS (P-Channel MOSFET)

Symbol

NMOS



PMOS

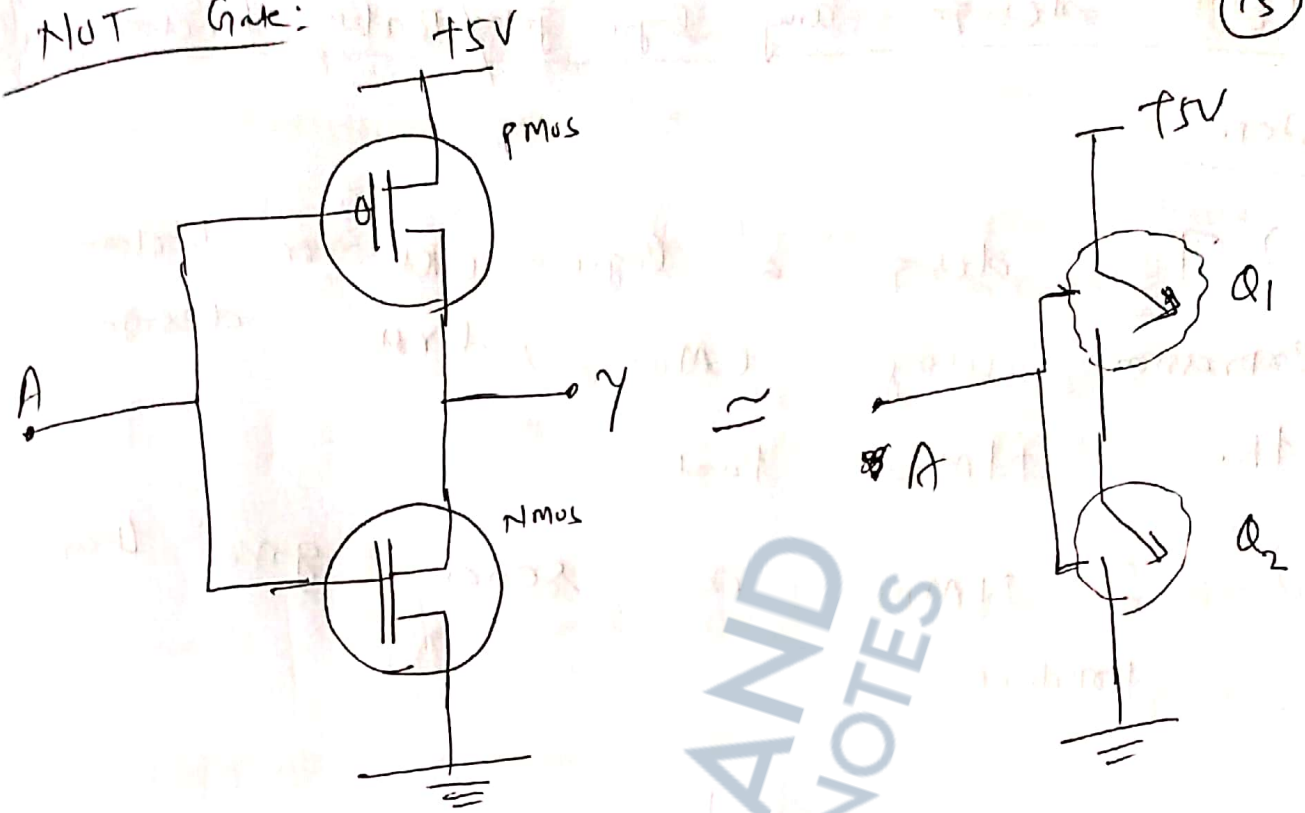


G → Gate, D → Drain, S = Source

PMOS & NMOS

Logic value	PMOS	NMOS
Logic '0'	ON	OFF
Logic '1'	OFF	ON

NOT Gate:

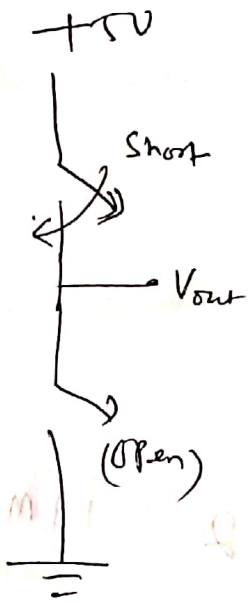


When A = 0, Q_1 = PMOS = ON,

Q_2 = NMOS = OFF

V_{out} = +5V

= logic '1'



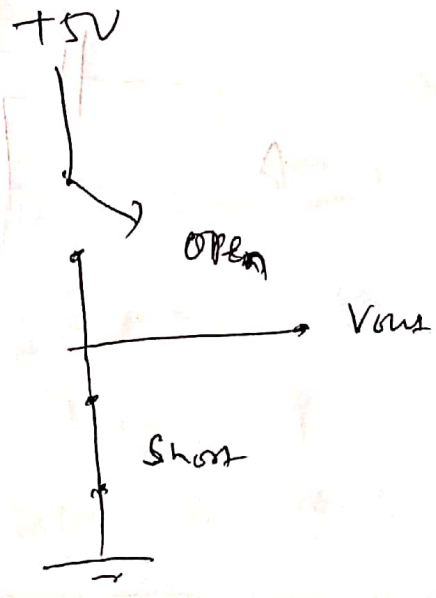
A = 0, Y = 1 (NOT gate)

When A = 1,

PMOS = OFF,

NMOS = ON,

V_{out} = 0V = logic = 0



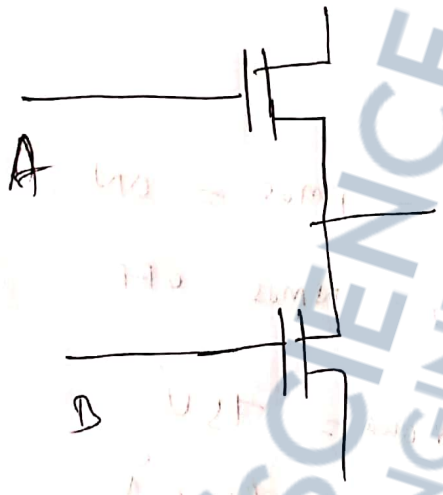
A = 1, Y = 0. NOT gate

To design any logic gate/Boolean function

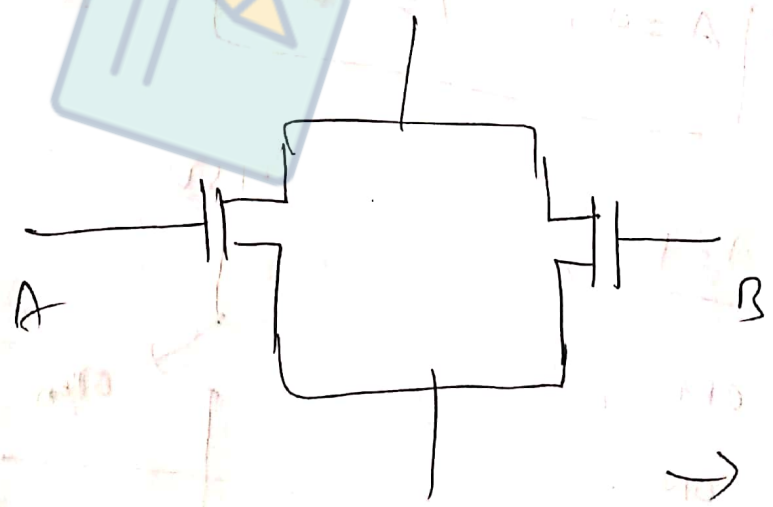
Steps

1) To design a logic ckt or boolean expression using CMOS, first design the NMOS part

2) 2 NMOS in series give their product.



2 NMOS in parallel give their sum.



3) Then design the PMOS part exactly opposite of NMOS Connection.

i.e If NMOS are in series → PMOS should be in parallel.

If " " " parallel → PMOS " " series

4) Their Combination (NMOS and PMOS) gives the O/P → Complement of design of NMOS function.

Ex:-1) Design a CMOS ckt to implement NAND gate (i.e $\overline{AB}$)

Ans:- To design a NAND gate, a AND gate is

We have to design PMOS part will be

NMOS first. Then of NMOS part.

Exactly opposite

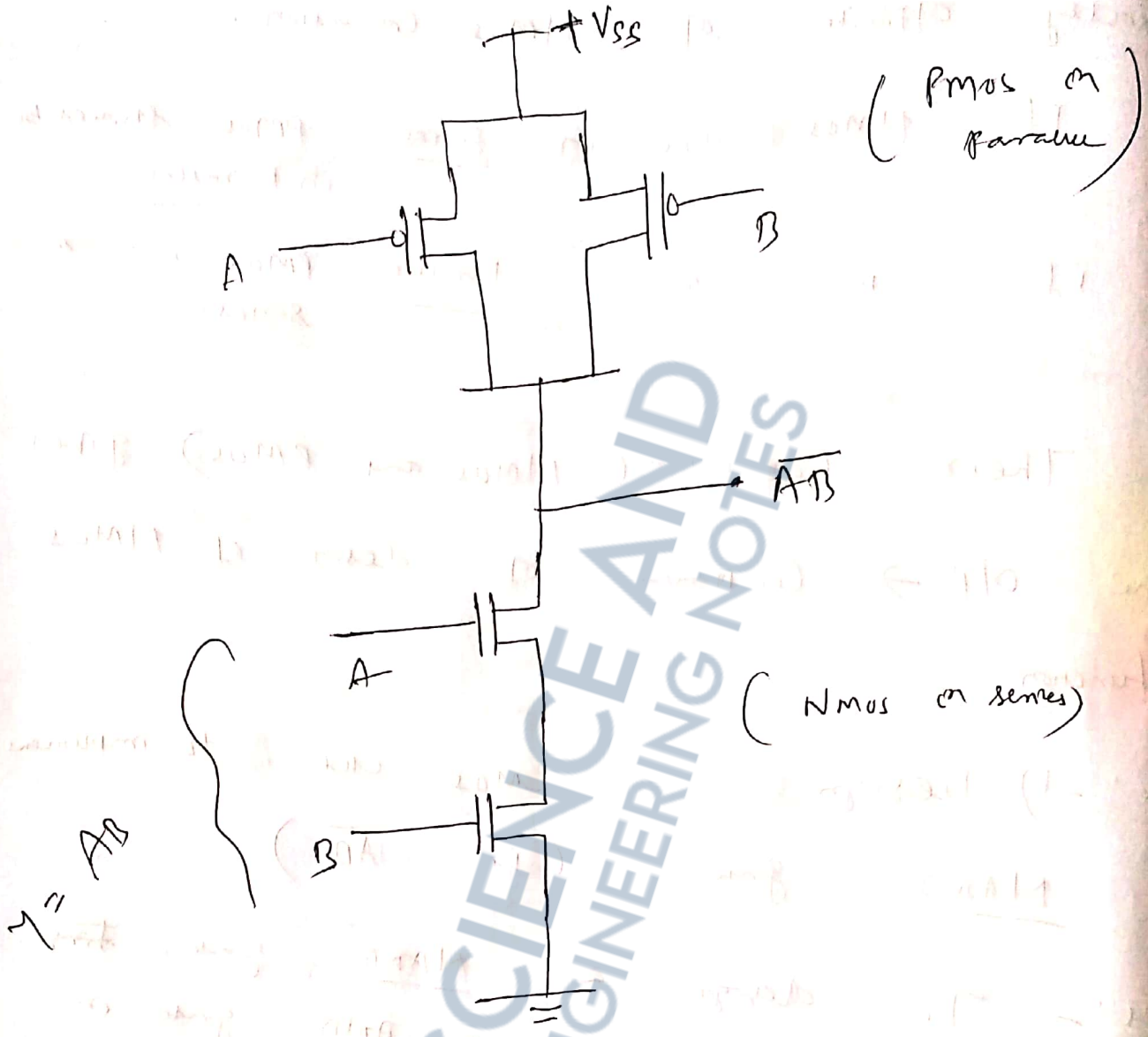
After Combining NMOS & PMOS, the

Resultant output is $\overline{AB}$ (NAND gate)

→ To design a AND gate using NMOS

2 NMOS should be in series. for PMOS they will be in parallel. The design of NAND gate

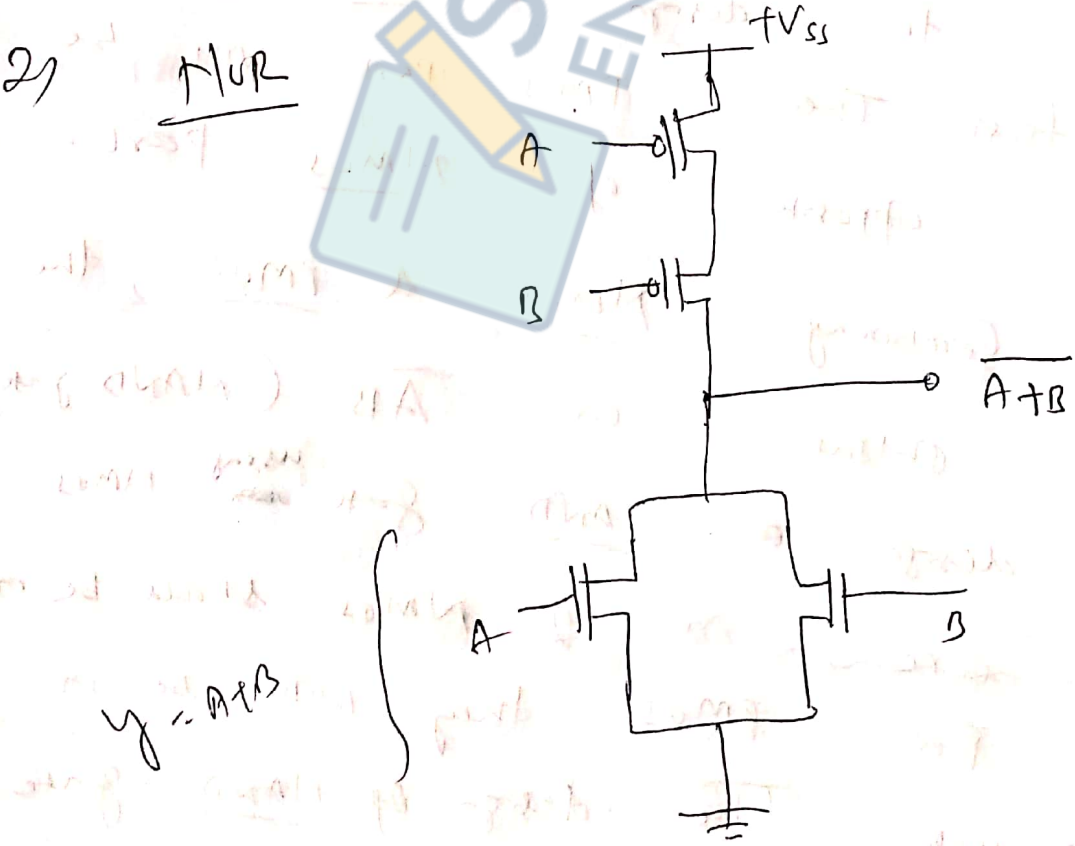
is as follows



(PMOS in parallel)

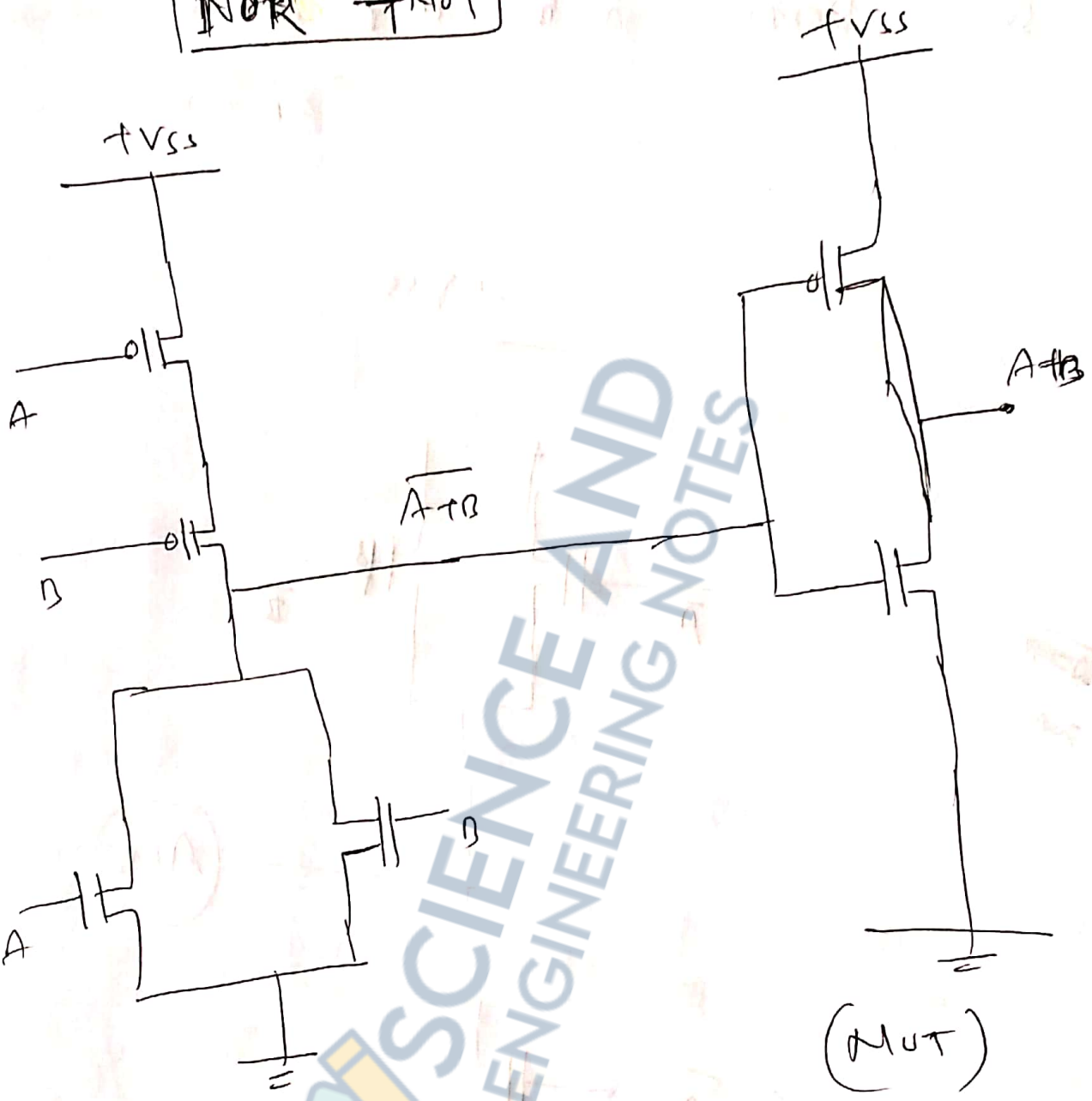
(NMOS in series)

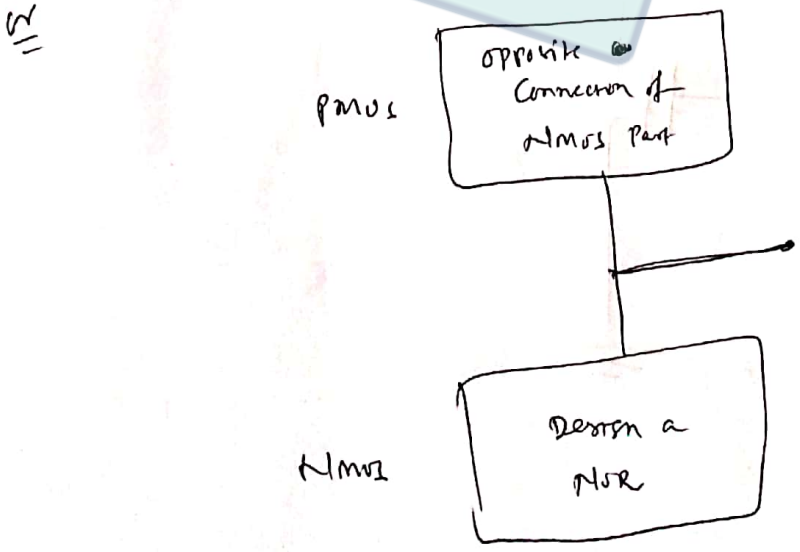
2) NOR



3) OR

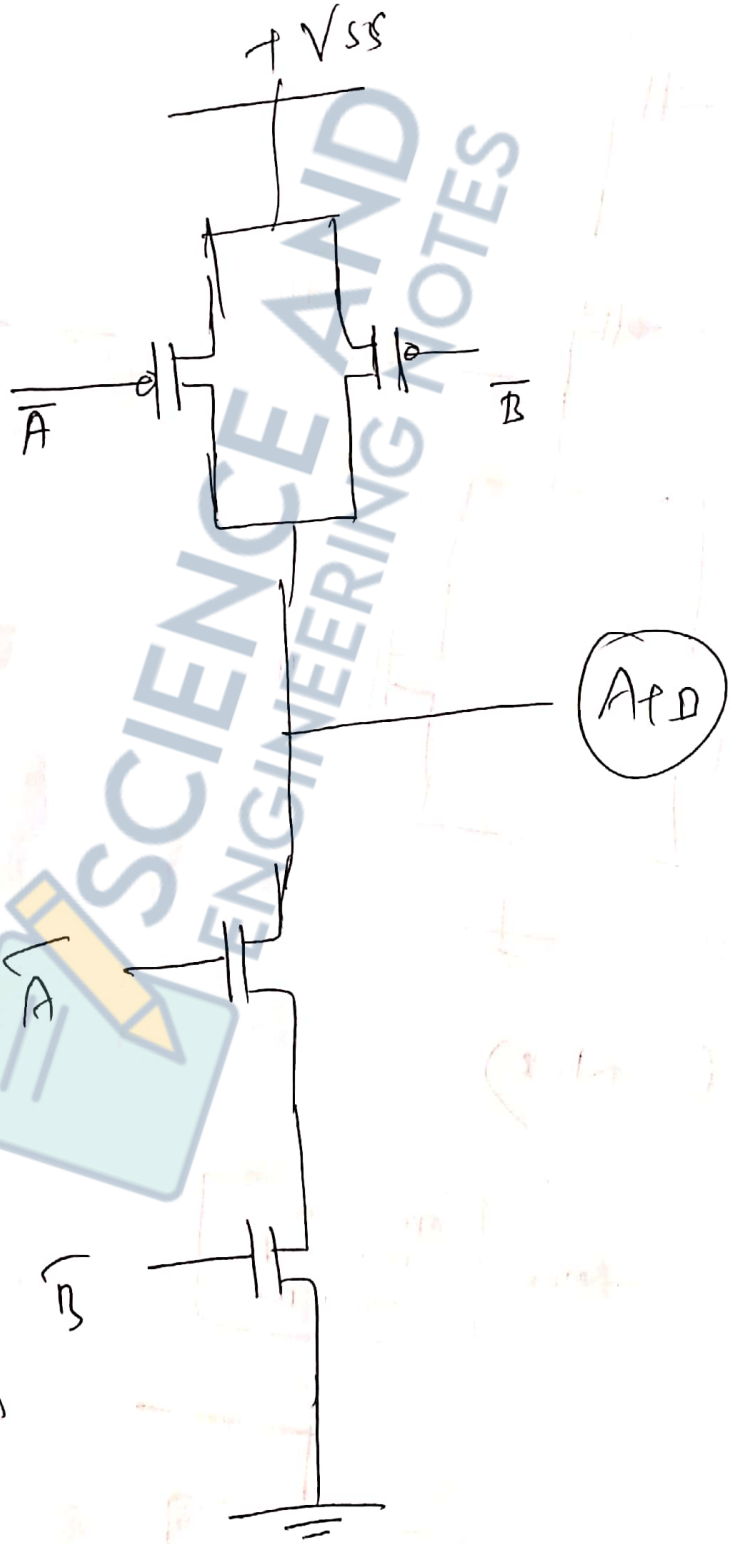
NOR GATE





NOR → $\overline{A+B} = \bar{A} \cdot \bar{B}$

So Design $\bar{A} \cdot \bar{B}$ using NMOS.



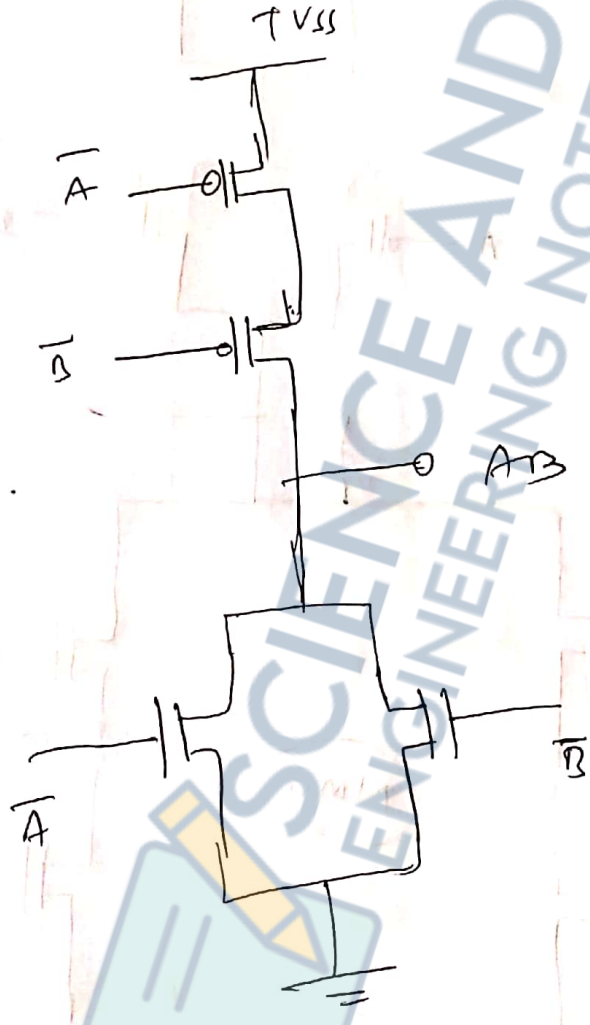
$y = \bar{A} \cdot \bar{B}$

$= \overline{A+B}$

4) AND gate

Design a NAND gate in NMOS.

$\overline{AB} = \overline{A} + \overline{B}$



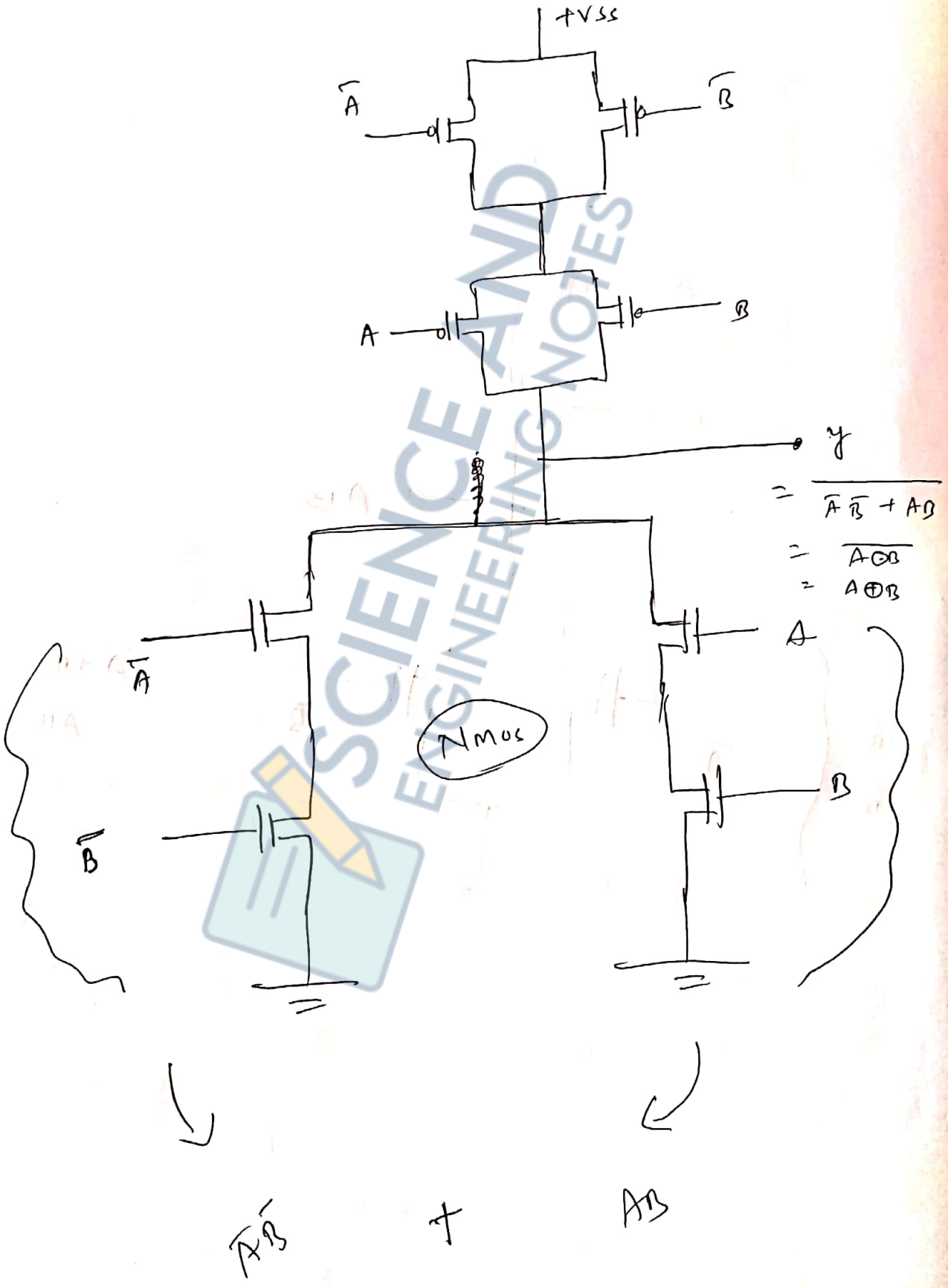
$\overline{A+B}$
$= \overline{AB}$

or $= \overline{A}+\overline{B}$
$= \overline{AB}$

5) Ex OR

To design Ex-OR, design Ex-NOR

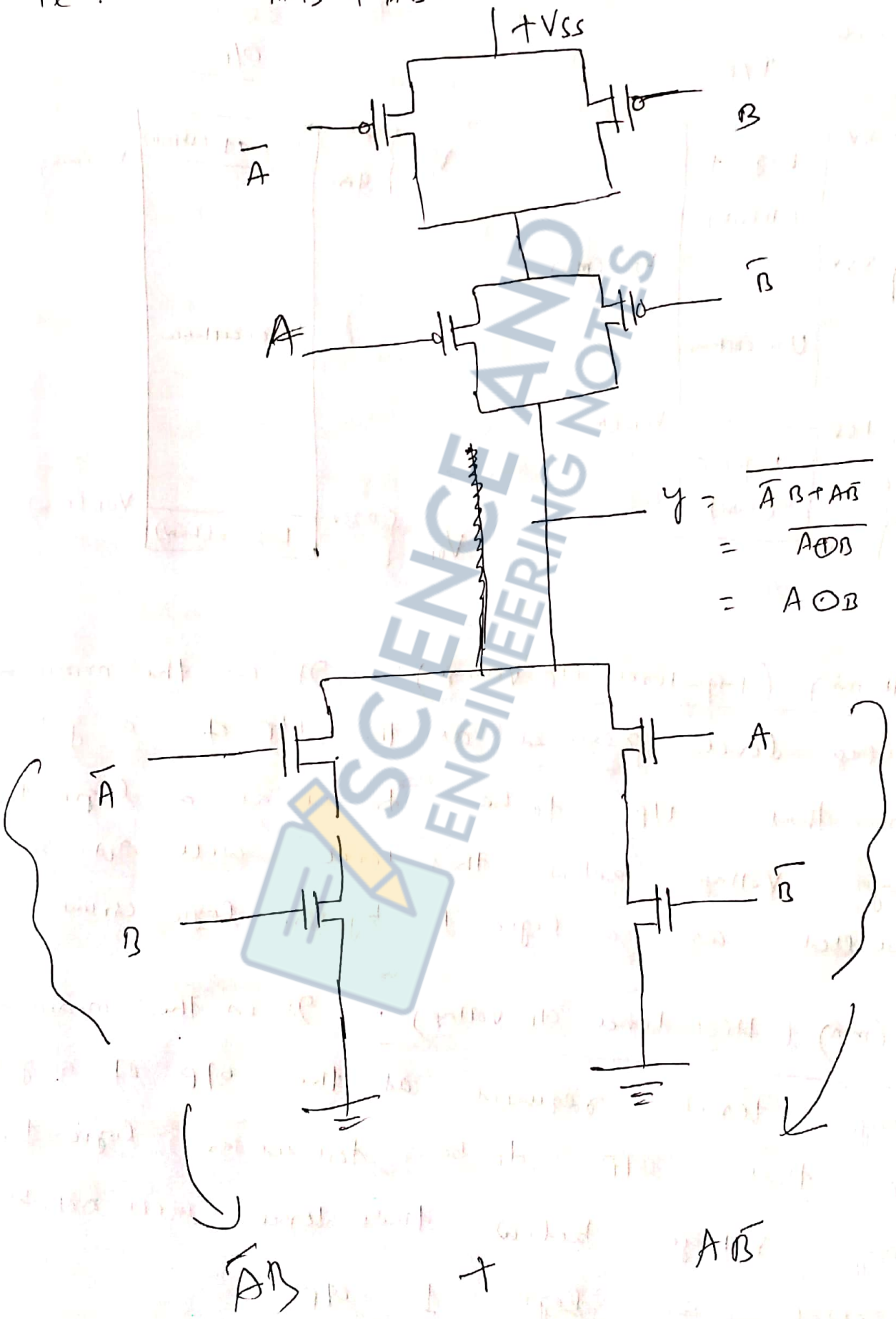
Using NMOS for $A\bar{B} + \bar{A}B$



6) **EX-NOR**

To design EX-NOR, design EX-OR using NMOS

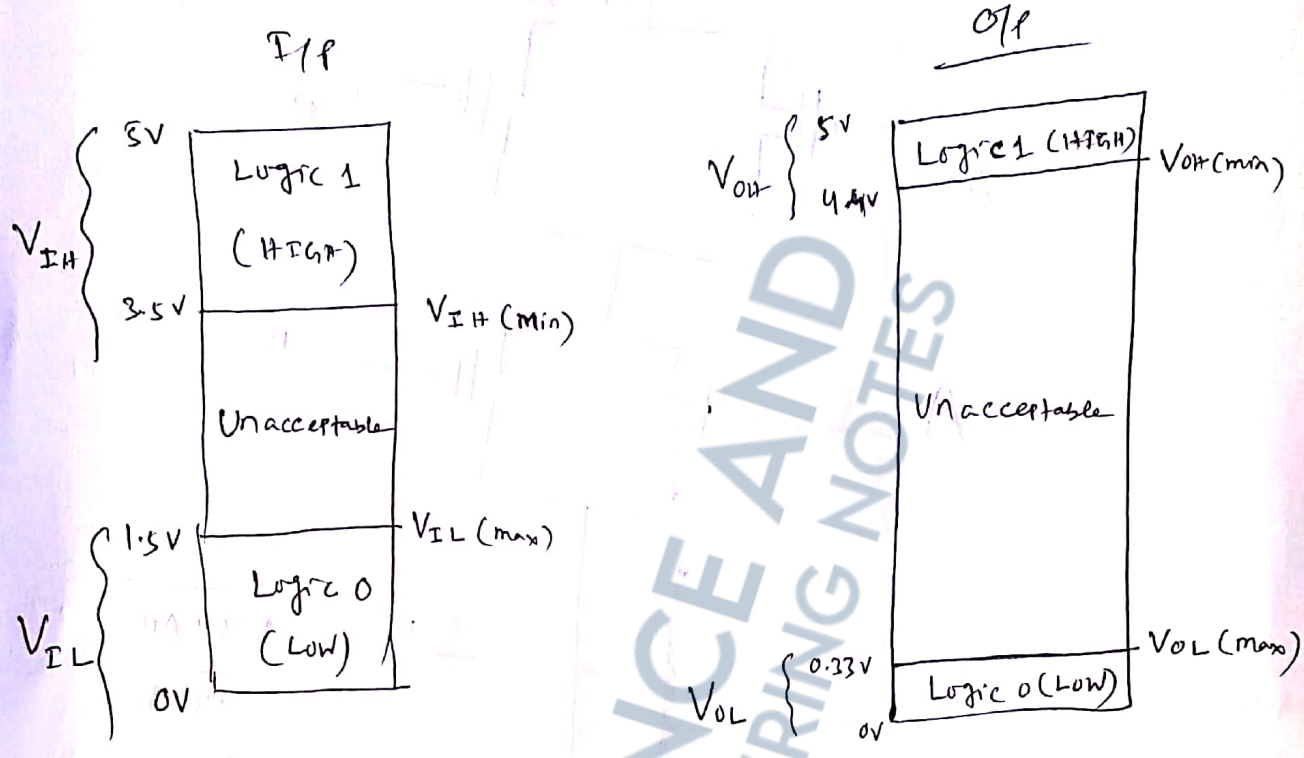
ie. $\bar{A}B + A\bar{B}$.



Basic operational Characteristics & parameters

1) Noise margin

CMOS :- (I/P & O/P logic levels for CMOS)



V_{IH}(min) (High level I/P voltage) :- It is the minimum voltage level required at the I/P of a gate for that I/P to be treated as a logic 1. Any voltage below this level will not be accepted as a logic 1 by the logic circuit.

V_{OH}(min) (High level O/P voltage) :- It is the minimum voltage level required at the O/P of a gate for that O/P to be treated as logic 1. Any voltage below this level will not be accepted as logic 1 O/P.

$V_{IL(max)}$ (Low level i/p voltage)

It is the maximum voltage level that can be treated as logic '0' at the i/p of the gate. Any voltage above this level will not be treated as logic '0' i/p by the logic gate.

$V_{OL(max)}$ (Low level o/p voltage)

It is the maximum voltage level that can be treated as logic 0 at the output of the gate. Any voltage above this level will not be treated as a logic 0 output.

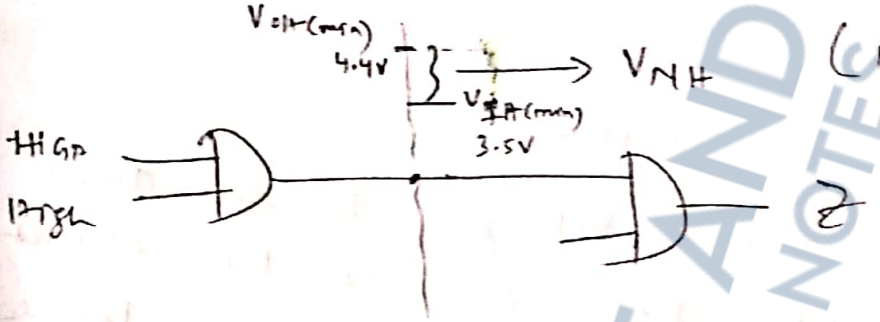
Noise Margin

Noise is unwanted voltage that is induced on electrical circuits and can present a threat to the proper operation of the circuit. In order not to be adversely affected by noise, a logic circuit must have a certain amount of noise immunity. This is the ability to tolerate a certain amount of unwanted voltage fluctuation on its inputs without changing its o/p state.

So Noise margin can be defined as

" It is the maximum noise voltage added to the i/p signal of the digital circuit that does not cause undesirable change in the circuit o/p". It is expressed in volts.

Ex:.



i.e. If 2 i/ps of AND gate are high o/p must be high. (> 4.4V)

But due to noise this o/p may drop ~~to~~ below 4.4V.

→ It can drop upto 3.5V, because for another AND it will be used as i/p. If Q drops below 3.5V it will be treated as ~~0~~ unacceptable. So the entire o/p(Z) will be ~~0~~. wrong.

So a margin is given or a

(ii) tolerance is given.

i.e

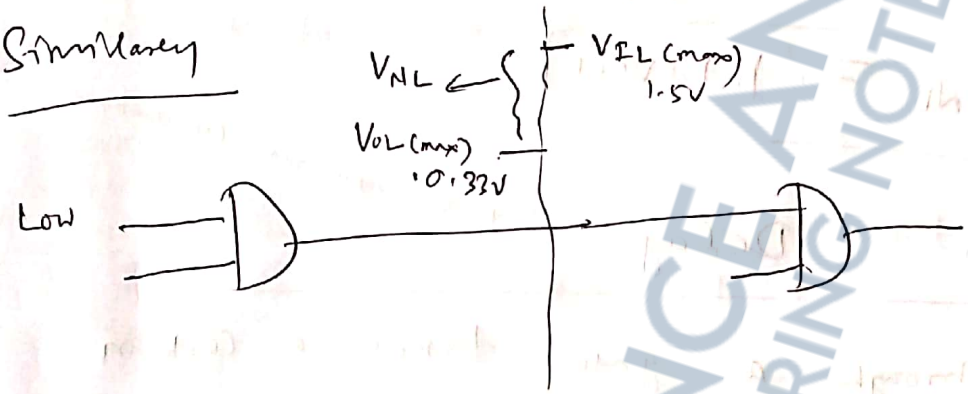
$$V_{NH} = V_{OH(min)} - V_{IH(min)} \quad \text{— (1)}$$

↙

High level noise margin

ex:- $V_{NH} = 4.4 - 3.5 = 0.9V$.

Similarly



$$V_{NL} = V_{IL(max)} - V_{OL(max)} \quad \text{— (2)}$$

The O/P of AND should be Zero as one 'Low' I/P is given. It should be (0V – 0.33V). But if due to noise it exceeds 0.33V, it can go up to max^m 1.5V, not beyond that. Because for second AND gate, $V_{IL(max)} = 1.5V$, if the second AND gate treat exceeds 1.5V, this as unacceptable. The entire O/P goes wrong.

So the low level noise margin is defined as

$$V_{HL} = V_{IL(max)} - V_{OL(max)}$$

Ex:- $V_{NL} = 1.5V - 0.33V$

$= 1.17$

$V_{NL} = 1.17$ Volt

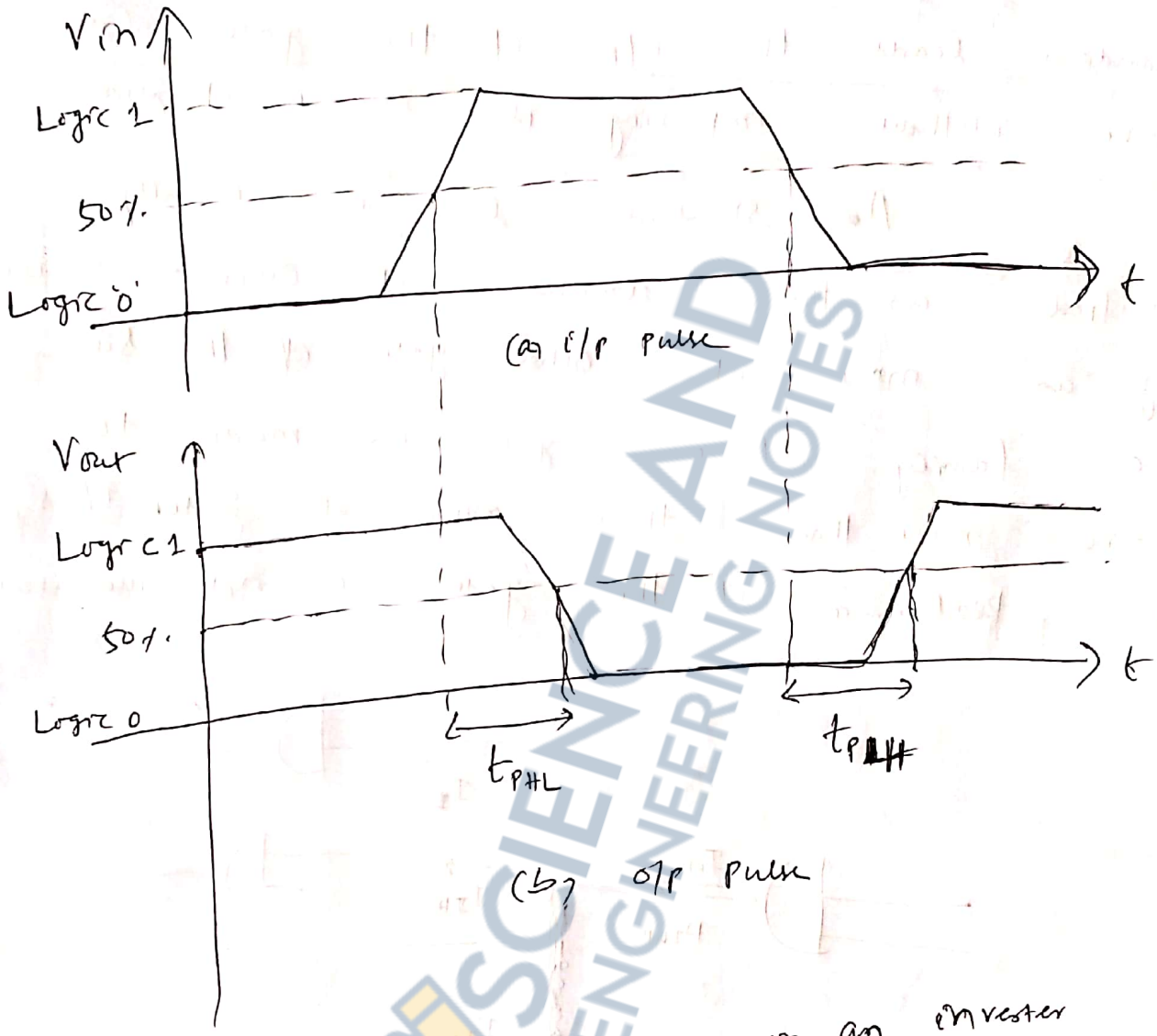
2) Propagation Delay:

A pulse through a gate takes a certain amount of time to propagate from i/p to o/p. This interval of time is known as propagation delay of the gate. It is the average transition delay time t_{pd}, expressed by

$$t_{pd} = \frac{t_{PLH} + t_{PHL}}{2}$$

Where t_{PLH} is the signal delay time when the o/p goes from logic 0 to a logic 1 state.

and t_{PHL} is the signal delay time when o/p goes from a logic 1 to logic '0' state.



Prg:- Propagation delay in an inverter

3) Fan-in :- → m { ○ — fan-in = M

Fan-in of a logic gate is defined as the number of inputs that can be connected to a gate without any degradation in the normal operation.

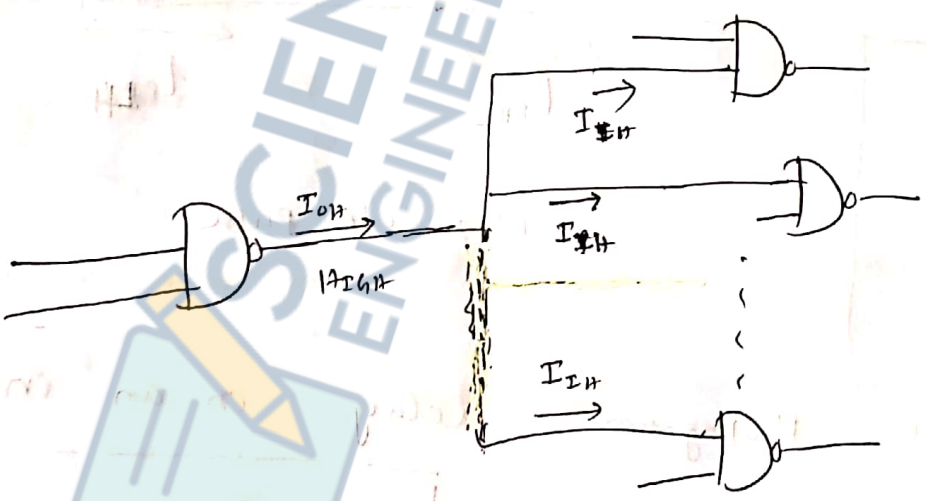
Or Fan-in of a logic circuit is defined as the number of i/ps that the gate is designed to handle.

Fan-out

Also called

The fan-out (loading factor) of a logic gate is defined as the maximum number of standard loads the o/p of the gate can drive without impairing its normal operation.

A standard load is usually specified as the amount of current needed by an input of other gate of the same IC family. If a gate is made to drive more than this number of gates i/ps, the performance of the gate is not guaranteed.



(a) Current sourcing in HIGH state

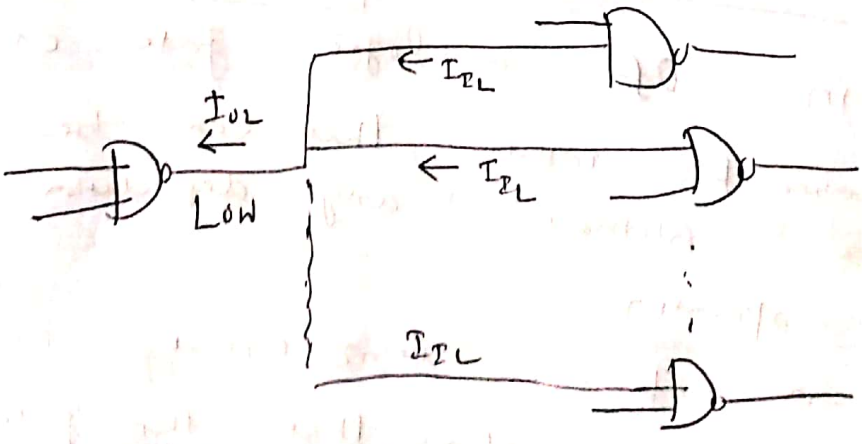


Fig: 1 (b) Current sinking in LOW state

Fan-out may be HIGH state fan-out i.e the fan-out of the gate when its o/p is Logic 1, or it may be LOW state fan-out i.e the fan-out of the gate when o/p is logic 0.

The smaller of these two numbers is taken as the actual fan-out. The driving gate sinks current when it is in LOW state & sources current when it is in HIGH state

$$\text{High state fan-out} = \frac{I_{OH(max)}}{I_{IH}}$$

Where $I_{OH(max)}$ is the maximum current that the driver gate can source when it is in 1 state. and I_{IH} is the current drawn by each driven gate from driver gate.

$$\text{LOW STATE fan-out} = \frac{I_{OL(max)}}{I_{IL}}$$

Where $I_{OL(max)}$ is the maximum current that the driver gate can sink when its o/p is a logic 0 and I_{IL} is the current drawn from each driven gate by the driver gate. Figure 1 depicts the current sourcing and current sinking actions for the TTL 7400 NAND gate.

Q) 1) Which of the following can be the name of an entity?

(i) NAND ✗ → Keyword

(ii) Nand_gate ✓

(iii) Nand gate → Space is there between 2 words

(iv) 1NAND → Cannot start a entity with numeric number.

(v) @NAND → ✗ Cannot start with special character

2) Write the correct syntax for entity declaration?

Entity entity_name

Port (signal_names : signal_modes;
signal_names : signal_modes)

End entity;

Ans: Entity entity_name is

Port (signal_names : signal_modes;
signal_names : signal_modes);

End ~~entity~~ entity_name;

3) Sensitivity list of a process contains

(a) Constants
(b) Signals
(c) Variables
(d) Literals

Ans: (b) signals

Ex:- Process Statement for 4:1 MUX

(Behavioral Modelling)

Process (I, S) → signals.

I = i/p (I0, I1, I2, I3)

S → Selection (S0, S1)

4) Difference between Concurrent & Sequential Statements :-

Concurrent Statements are defined outside Architecture.

Sequential Statements are defined outside process.

Concurrent Statements :-

→ Simple Signal assignment Statement

→ Conditional Signal " "

→ Selected " "

```
Architecture
  begin
    → Concurrent Statements
  end architecture
```

Sequential Statements

→ VHDL process

→ Sequential Signal assignment Statement

→ Variable assignment Statement

→ If Statement

→ Case Statement

→ Simple for loop Statement

```
Architecture
  Process (...)
  begin
    Sequential Statement
    Sequential Statement
  end process
end architecture
```

3) Design a asynchronous counter which counts the sequence 1, 3, 5, 7 ---- 1, 3, 5, 7 ---- & so on.

Ans: We require 3 F/Fs which can count upto 111. We have to take preset & clear switches on to account for design.

For 1, 3, 5, 7, the preset & clear should be clocked operation.

(i) i.e $\overline{PRE} = 1$, & $\overline{CLR} = 1$

(ii) After '7', CT will by default go to 000. But we have to forcefully move CT to 001.

Q_2	Q_1	Q_0
0	0	0
↓	↓	↓
0	0	1

$\overline{PRE_2} = 1$, $\overline{CLR_2} = 0$ } So that $Q_2 \to 0$

$\overline{PRE_1} = 1$, $\overline{CLR_1} = 0$ } So that $Q_1 \to 0$

$\overline{PRE_0} = 0$, $\overline{CLR_0} = 1$ } So that $Q_0 \to 1$

(iii) For rest cases, i.e 2, 4, 6, ~~we don't care~~ ~~because they will never occur~~.

When 2 comes → Q_2 will step to 3

4 " → " " " 5

6 " → " " " 7

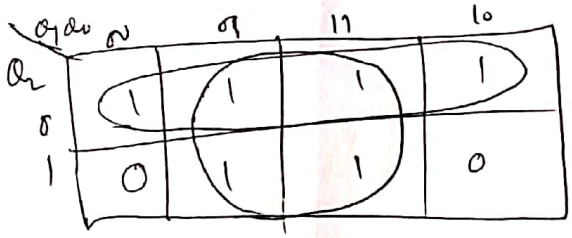
3	5	7
0 1 0	1 0 0	1 1 0
↓ ↓ ↓	↓	↓
0 1 1	1 0 1	1 1 1
$\overline{PRE_2} = 1$, $\overline{CLR_2} = 0$	$\overline{PRE_2} = 0$, $\overline{CLR_2} = 1$	$\overline{PRE_2} = 0$, $\overline{CLR_2} = 1$
$\overline{PRE_1} = 0$, $\overline{CLR_1} = 1$	$\overline{PRE_1} = 1$, $\overline{CLR_1} = 0$	$\overline{PRE_1} = 0$, $\overline{CLR_1} = 1$
$\overline{PRE_0} = 0$, $\overline{CLR_0} = 1$	$\overline{PRE_0} = 0$, $\overline{CLR_0} = 1$	$\overline{PRE_0} = 0$, $\overline{CLR_0} = 1$

(35)

Q_2	Q_1	Q_0	$\overline{PRE2}$	$\overline{CLR2}$	$\overline{PRE1}$	$\overline{CLR1}$	$\overline{PRE0}$	$\overline{CLR0}$
0	0	0 →	1	0	1	0	0	1
0	0	1 →	1	1	1	1	1	1
0	1	0 →	1	0	0	1	0	1
0	1	1 →	1	1	1	1	1	1
1	0	0 →	0	1	1	0	0	1
1	0	1 →	1	1	1	1	1	1
1	1	0 →	0	1	0	1	0	1
1	1	1 →	1	1	1	1	1	1

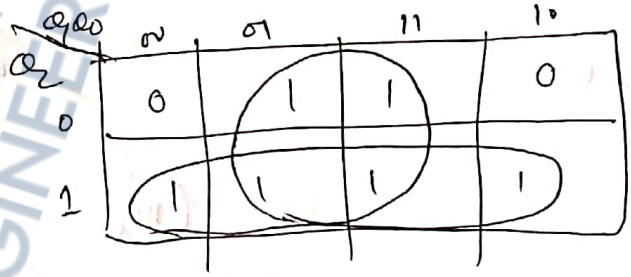
Solving K-Map

$\overline{PRE2}$



$\overline{PRE2} = \overline{Q_2} + Q_0$

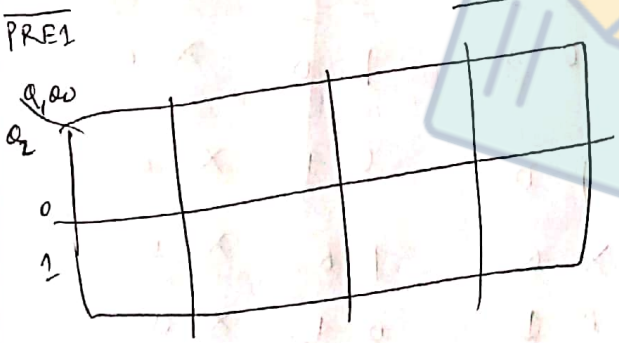
$\overline{CLR2}$



$\overline{CLR2} = Q_2 + Q_0$

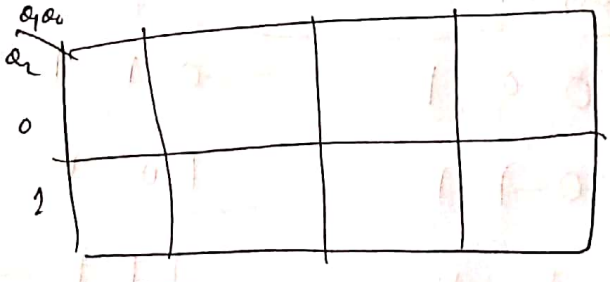
Similarly,

$\overline{PRE1}$



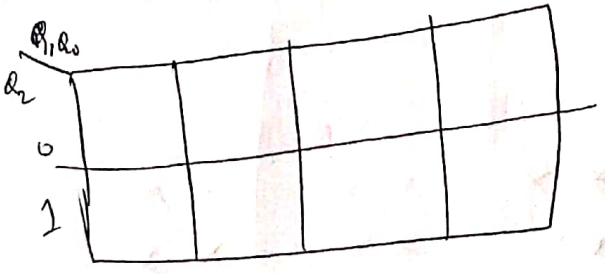
$\overline{PRE1} =$

$\overline{CLR1}$



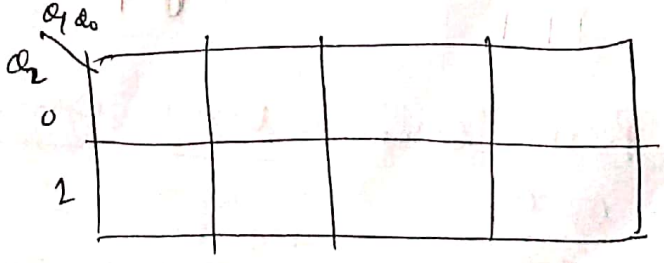
$\overline{CLR1} =$

$\overline{PRE0}$



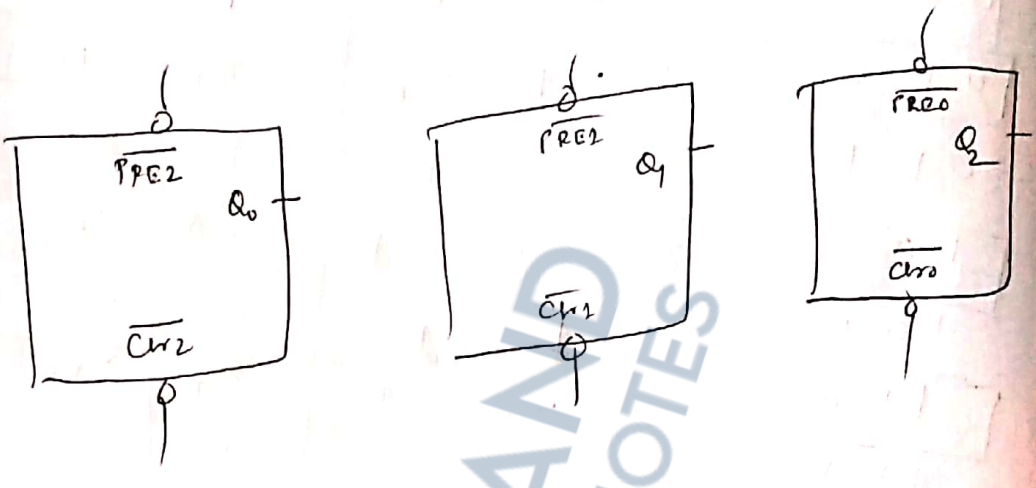
$\overline{PRE0} =$

$\overline{Clr0}$



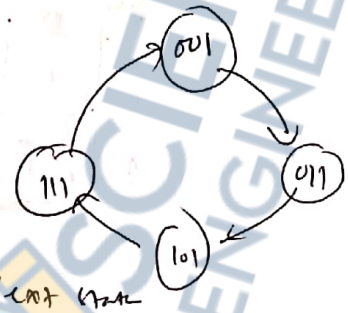
$\overline{Clr0} =$

Then Design the CKT accordingly



6) Design a Synchronous Counter that counts 1, 3, 5, 7, — — 1, 3, 5, 7 — —

Ans:

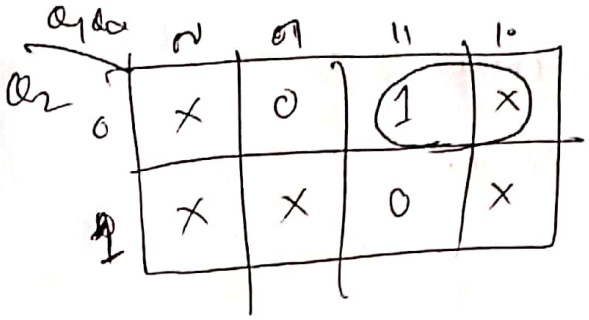


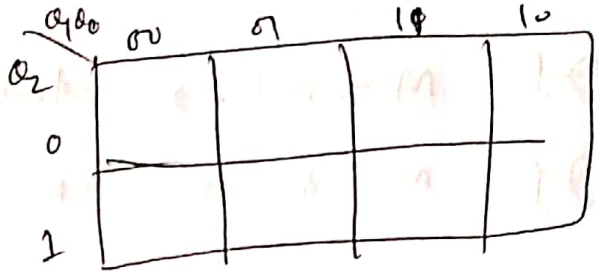
Present State			Next State			Required excitation					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	1	0	1	1	0	x	1	0	x	0
0	1	1	1	0	1	1	0	0	1	x	0
1	0	1	1	1	1	x	0	1	0	x	0
1	1	1	0	0	1	0	1	0	1	x	0

Rest cases Don't Care.

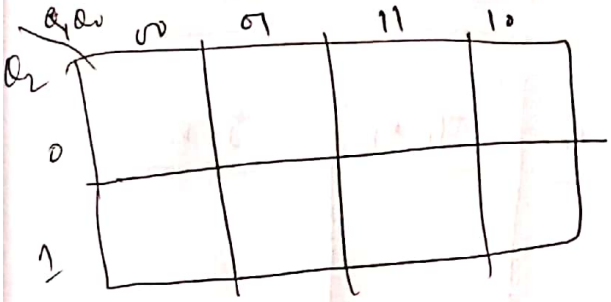
000	x	x	x	x
010	x	x	x	x
100	x	x	x	x
110	x	x	x	x

J_2 (In terms of Present state $Q_2 Q_1 Q_0$) K_2

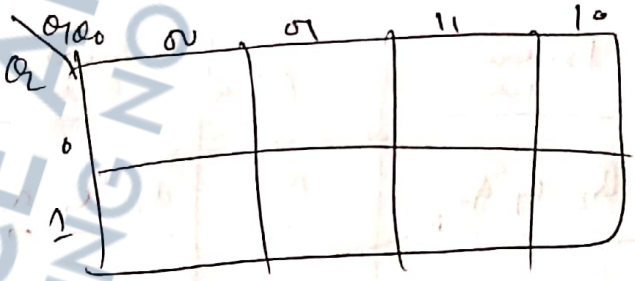




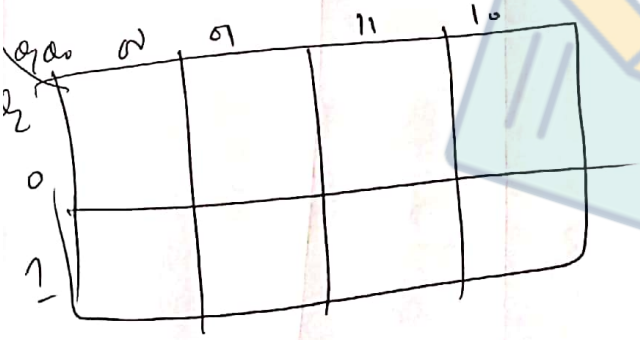
J_1



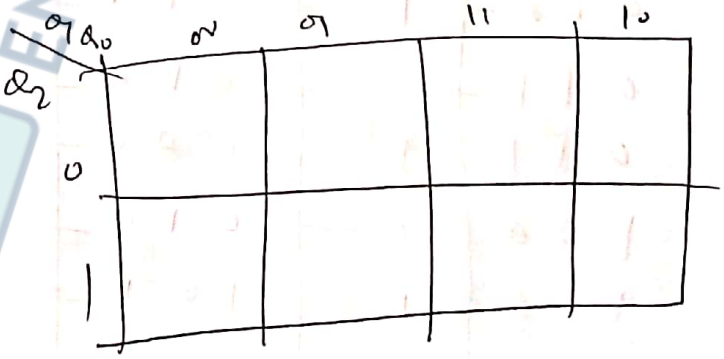
K_1



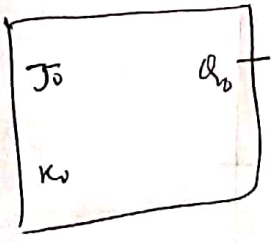
J_0

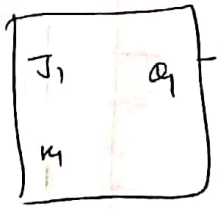


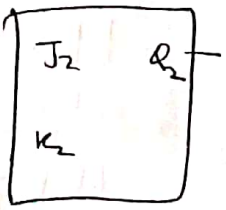
K_0



Design the ckt







11 Design a Synchronous 3 bit up-down Counter

We have a mode signal M

If M = 0 → down Counter

If M = 1 → up Counter

If M=0, 000 → 111

If M=1, 000 → 001

Present State			Mode	Next State			Required excitations					
Q_2	Q_1	Q_0	M	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	1	1	1						
0	0	0	1	0	0	1						
0	0	1	0	0	0	0						
0	0	1	1	0	1	0						
0	1	0	0	0	0	1						
0	1	0	1	0	1	1						
0	1	1	0	0	1	0						
0	1	1	1	1	0	0						
1	0	0	0	0	1	1						
1	0	0	1	1	0	1						
1	0	1	0	1	0	0						
1	0	1	1	1	1	0						
1	1	0	0	1	0	1						
1	1	0	1	1	1	1						
1	1	1	0	1	1	0						
1	1	1	1	0	0	0						

Solving

(Solve in terms of present state Q_2, Q_1, Q_0 & M)

$$J_2 = \bar{Q}_1 \bar{Q}_0 \bar{M} + Q_1 Q_0 M$$

$$K_2 = \bar{Q}_1 \bar{Q}_0 \bar{M} + Q_1 Q_0 M$$

$$J_1 = \bar{Q}_0 \bar{M} + Q_0 M$$

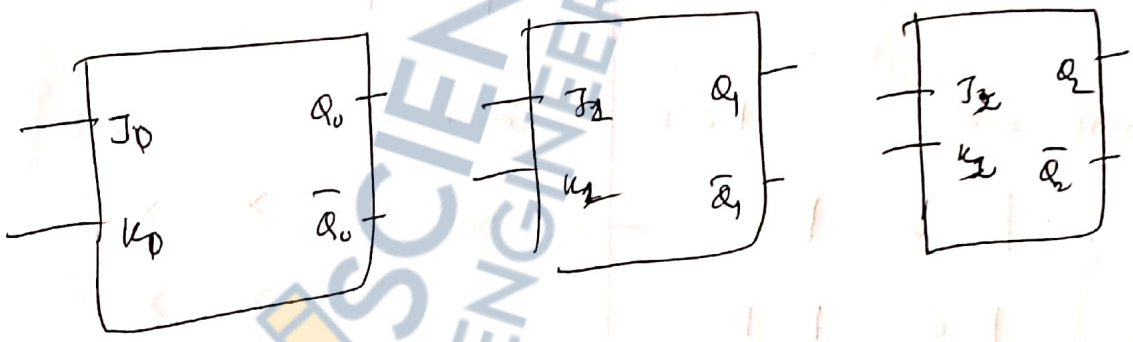
$$K_1 = \bar{Q}_0 \bar{M} + Q_0 M$$

$$J_0 = 1$$

$$K_0 = 1$$

Then design the ckt

M



8) Design 4 bit BCD to Excess-3 Code Converter

Let BCD Code $B_4\ B_3\ B_2\ B_1$

Excess-3 Code $X_4\ X_3\ X_2\ X_0$

(Add '3' with BCD number)

For 0000 → 0011

(BCD) (excess 3 code)

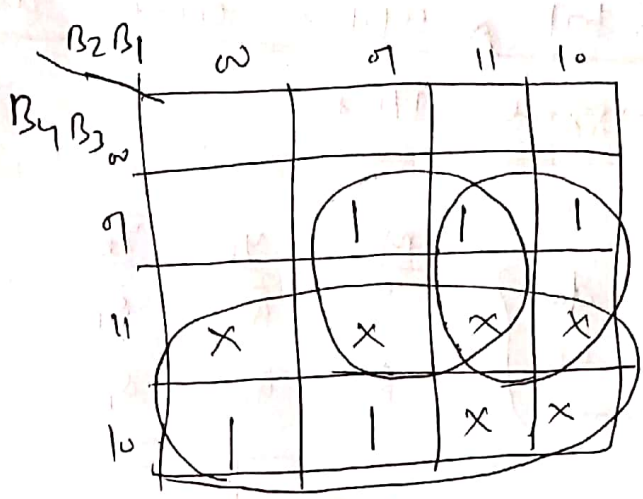
B_4	B_3	B_2	B_1	X_4	X_3	X_2	X_1
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

Rest are don't care.

Invalid BCD

Solving the K-Map for X_4, X_3, X_2, X_1 in terms of B_4, B_3, B_2, B_1, we have

X_4



$$X_4 = \cancel{B_4 \bar{B}_3} + B_4 + B_3 B_1 + B_3 B_2$$

$$\therefore X_4 = B_4 + B_3 B_2 + B_3 B_1$$

Similarly

$$X_3 = B_3 \bar{B}_2 \bar{B}_1 + \bar{B}_3 B_1 + \bar{B}_3 B_2$$

$$X_2 = \overline{B_2}\,\overline{B_1} + B_2 B_1$$

$$X_1 = \overline{B_1}$$

The circuit can be designed for X_4, X_3, X_3, and X_1 in terms of B_4, B_3, B_2 and B_1.

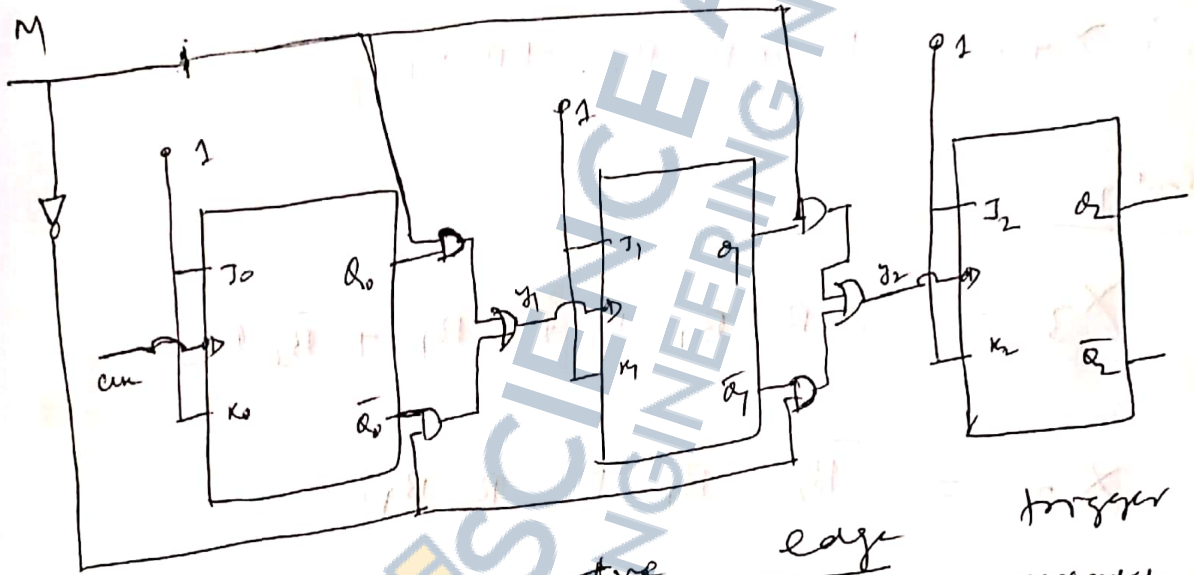
P.T.O

9) Design a 3 bit Asynchronous updown counter using -ve edge triggered F/F's

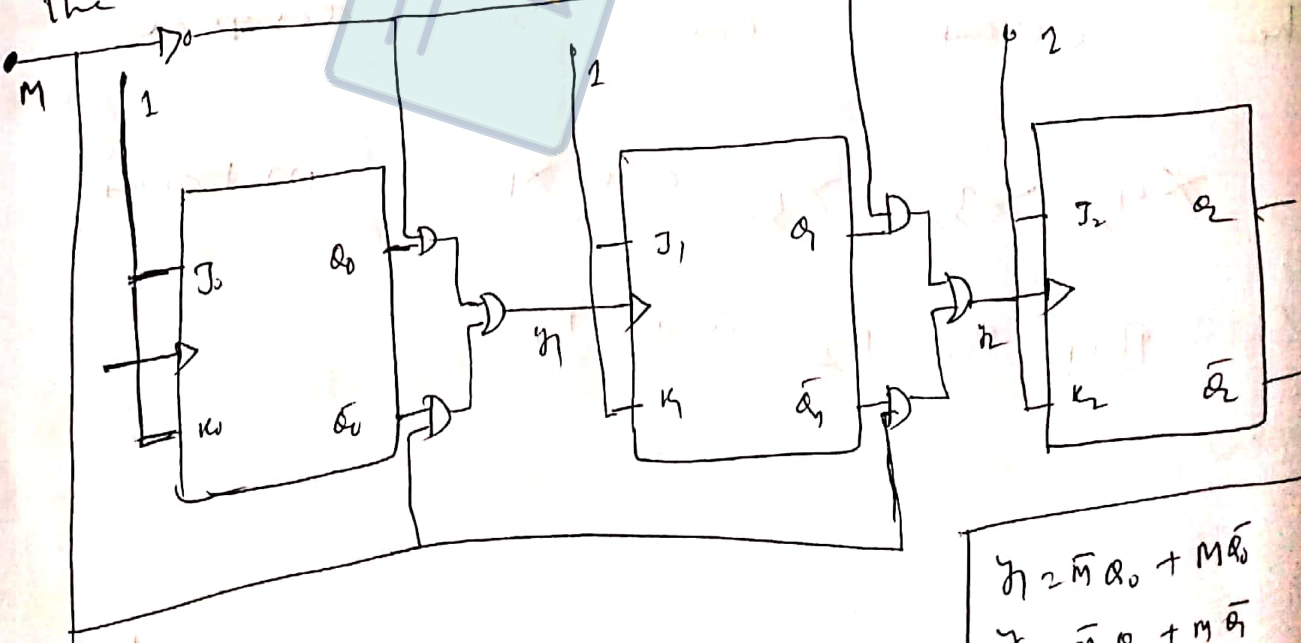
Ans: If M = 0, down counter; If M = 1, up counter

M	y_1	y_2
0	$\bar{Q}_0$	$\bar{Q}_1$
1	Q_0	Q_1

$$y_1 = \bar{M}\bar{Q}_0 + MQ_0$$
$$y_2 = \bar{M}\bar{Q}_1 + MQ_1$$



Similarly using +ve edge trigger F/F the connection will be exactly reversed.



$$y_1 = \bar{M}Q_0 + M\bar{Q}_0$$
$$y_2 = \bar{M}Q_1 + M\bar{Q}_1$$

10) Design a combinational ckt to produce 2's complement of a 4-bit binary number.

Ans:. Let the 4-bit I/P Binary Number represented by ABCD & 4 bit O/P 2's complement of that number be EFGH.

I/P					O/P			
A	B	C	D	2's Complement →	E	F	G	H
0	0	0	0		0	0	0	0
0	0	0	1		1	1	1	1
0	0	1	0		1	1	1	0
0	0	1	1		1	1	0	1
0	1	0	0		1	1	0	0
0	1	0	1		1	0	1	1
0	1	1	0		1	0	1	0
0	1	1	1		1	0	0	1
1	0	0	0		1	0	0	0
1	0	0	1		0	1	1	1
1	0	1	0		0	1	1	0
1	0	1	1		0	1	0	1
1	1	0	0		0	1	0	0
1	1	0	1		0	0	1	1
1	1	1	0		0	0	1	0
1	1	1	1		0	0	0	1

Solving the K-Map for ; E, F, G, H in terms of A, B, C, D, we have

$$E = A\bar{B}\bar{C}\bar{D} + \bar{A}B + \bar{A}D + \bar{A}C$$

$$F = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C$$

$$G = \bar{C}D + C\bar{D}$$

$$H = D$$

→ Now we can design the circuit with I/P A, B, C, D & O/P E, F, G, H.