DLD : 2103

Subject:

Date:

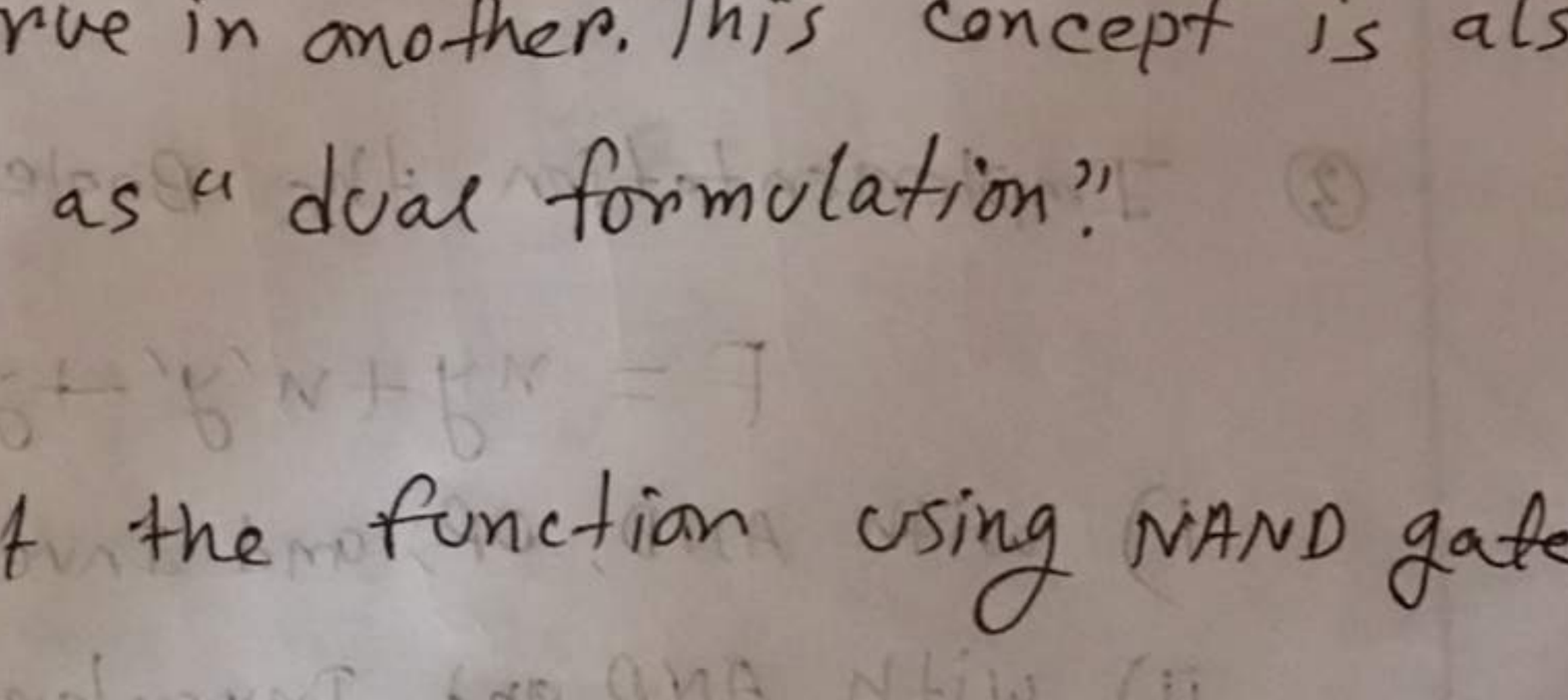
① Mention the duality principle?

The principle of duality is a kind of pervasive property of algebraic structure in which two principles or concepts are interchangeable only if all outcomes held true in one formulation are also held true in another. This concept is also referred to as "dual formulation".

② Implement the function using NAND gate?

$$F = A'B + Bc'$$

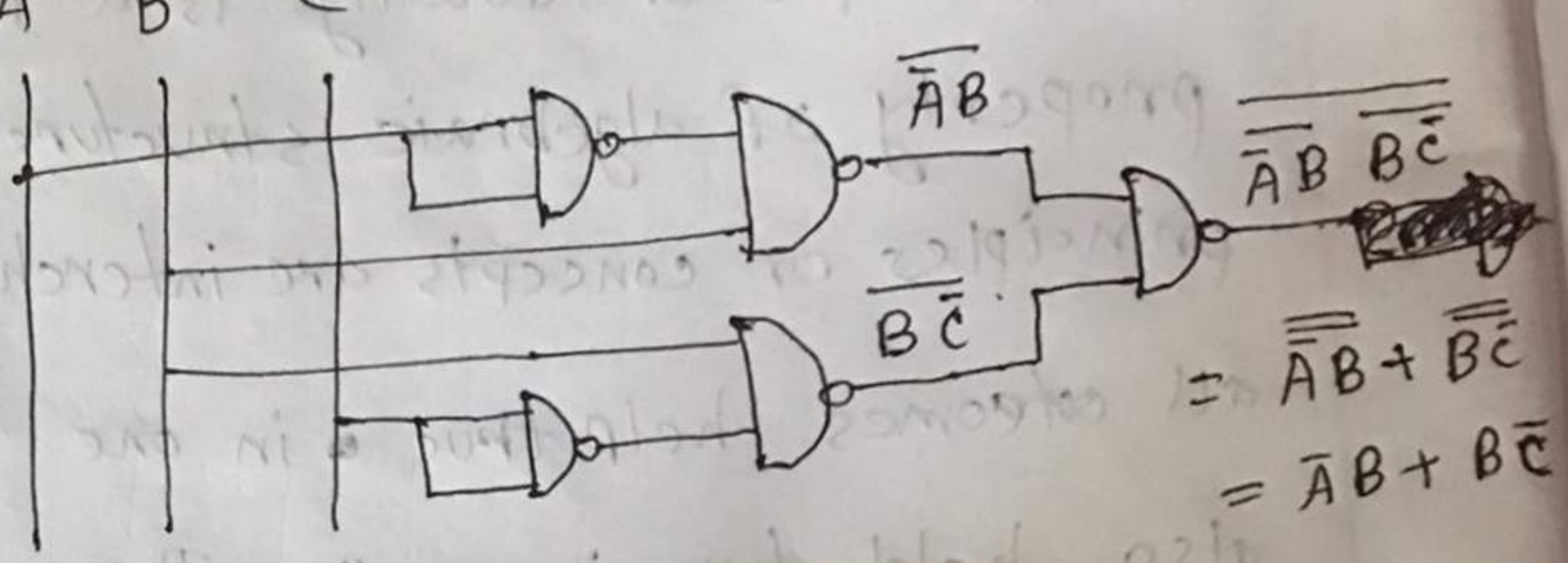
Logic gate:



Subject:

Date:

Implementation by NAND gate :



③ Implementation the Boolean function?

$$F = xy + x'y' + y'z$$

i) with AND, OR, and Inverter gate

ii) with AND and Inverterr gate

iii) with OR and " "

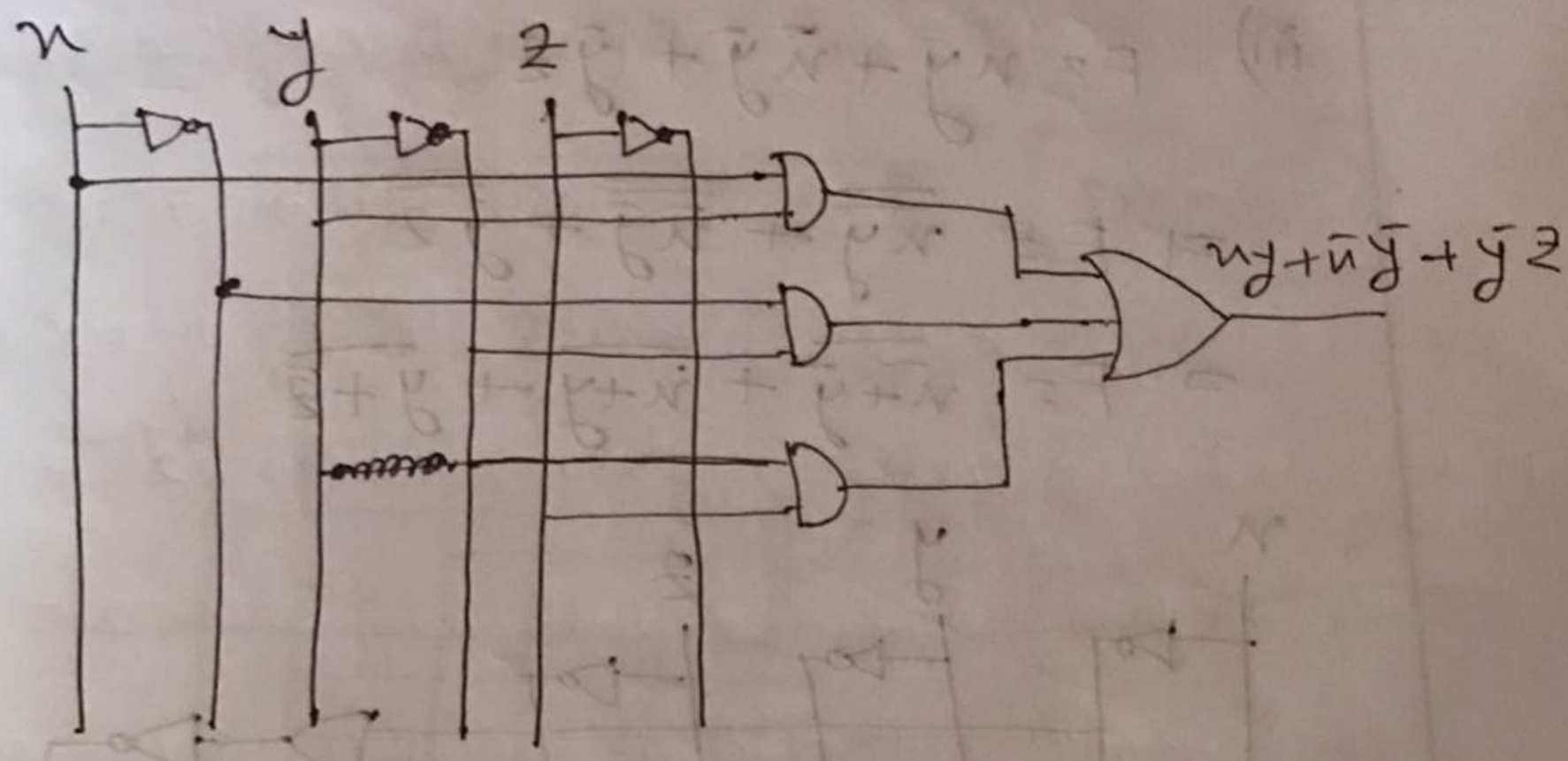
iv) with NAND " " "

v) with NOR " " "

i) $F = xy + \bar{x}\bar{y} + \bar{y}z$

Subject:

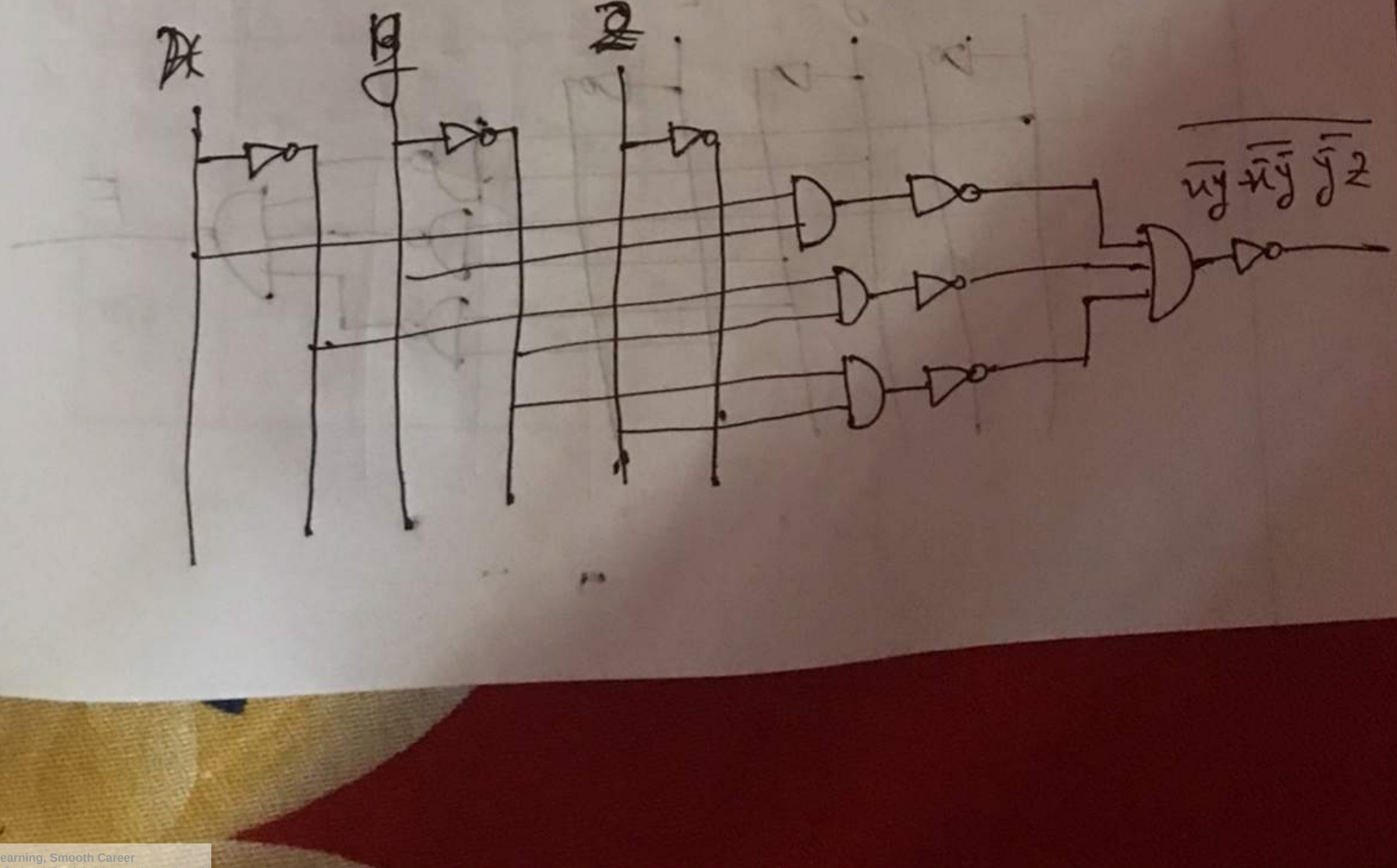
Date:



ii) $F = xy + \bar{x}\bar{y} + \bar{y}z$

$\Rightarrow \overline{F} = \overline{xy + \bar{x}\bar{y} + \bar{y}z}$

$\Rightarrow \overline{F} = \overline{xy}\ \overline{\bar{x}\bar{y}}\ \overline{\bar{y}z}$



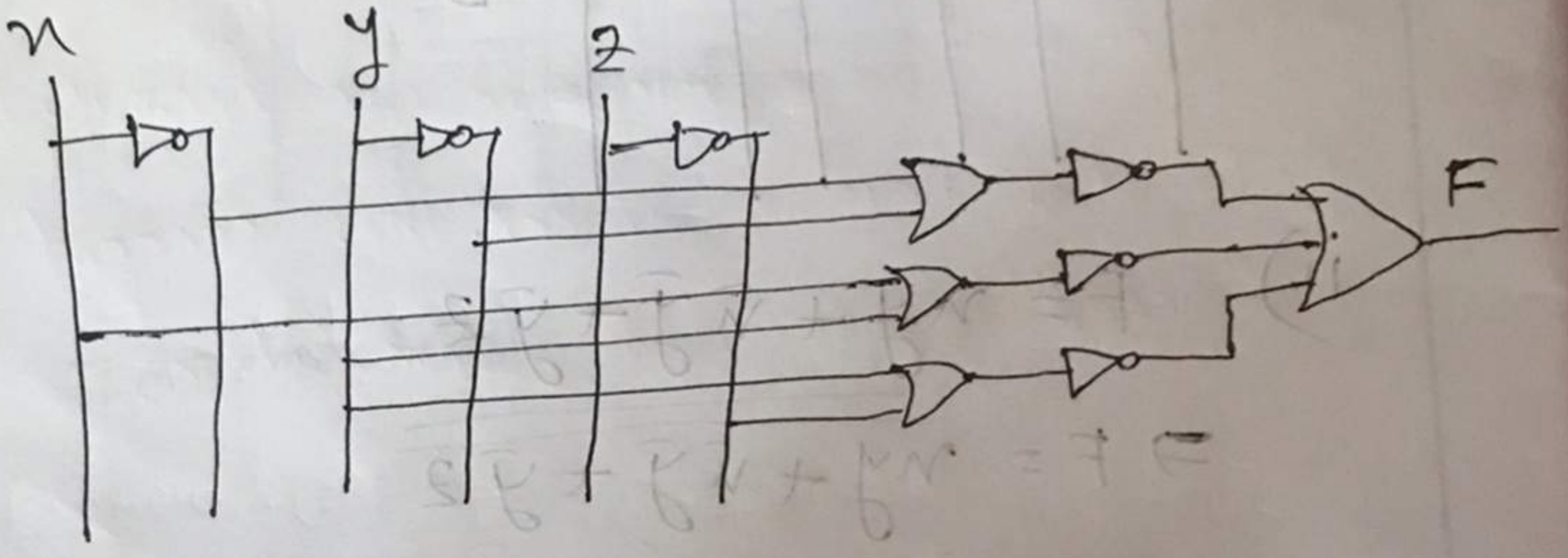
Subject:

Date:

iii) $F = xy + \bar{x}\bar{y} + \bar{y}z$

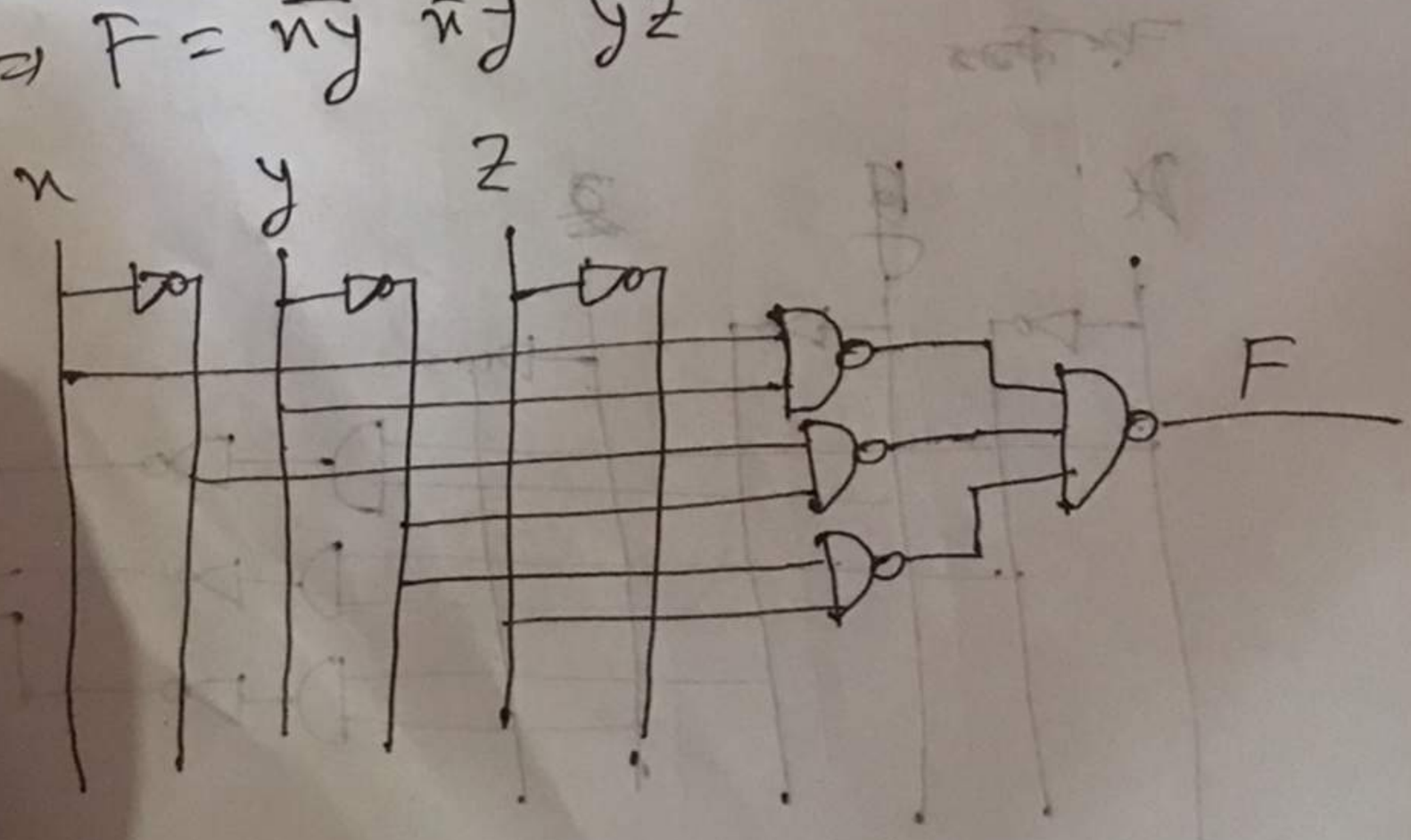
$\Rightarrow F = \overline{\overline{xy} \cdot \overline{\bar{x}\bar{y}} \cdot \overline{\bar{y}z}}$

$\Rightarrow F = \overline{\bar{x}+\bar{y}} + \overline{\bar{x}+y} + \overline{\bar{y}+\bar{z}}$



iv) $F = xy + \bar{x}\bar{y} + \bar{y}z$ [From (ii)]

$\Rightarrow F = \overline{\overline{xy} \cdot \overline{\bar{x}\bar{y}} \cdot \overline{\bar{y}z}}$

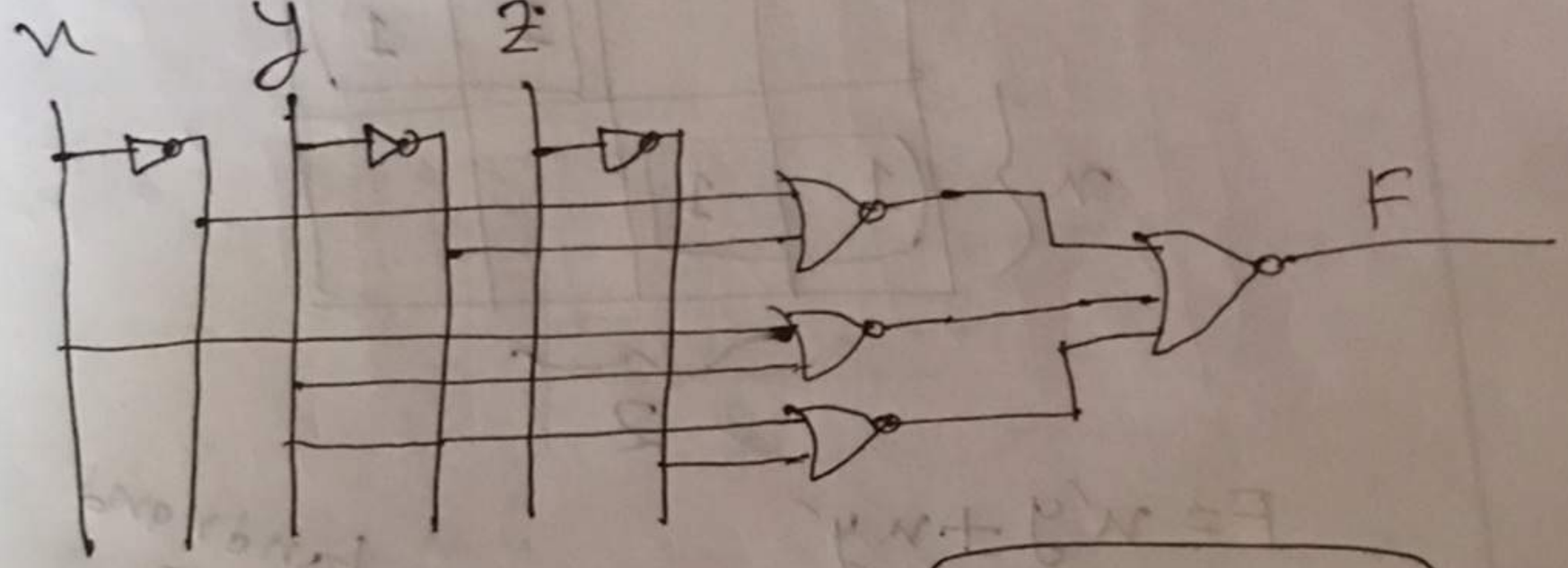


Subject:

Date:

(v) $F = xy + \bar{x}\bar{y} + yz$

$\Rightarrow F = \overline{\bar{x}+\bar{y}} + \overline{x+y} + \overline{y+\bar{z}}$ [from (iii)]



K-Map

Variable = 3

Note:

$x+0=x$

$x+1=1$

$x+x=x$

$x+\bar{x}=1$

$x.0=0$

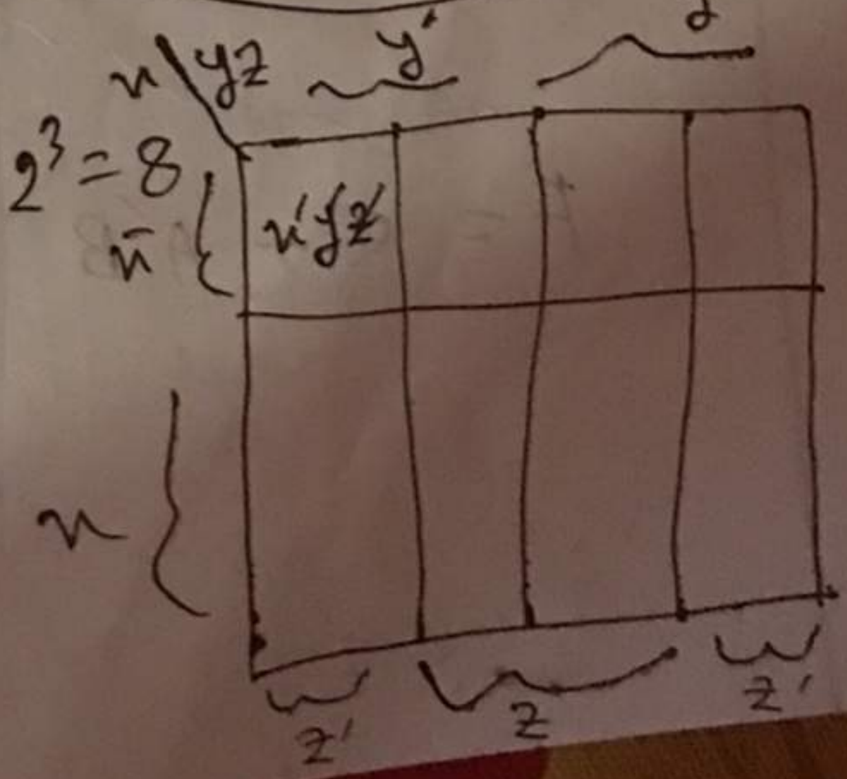
$x.1=x$

$x.x=x$

$x.\bar{x}=0$

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

$2^3 = 8$

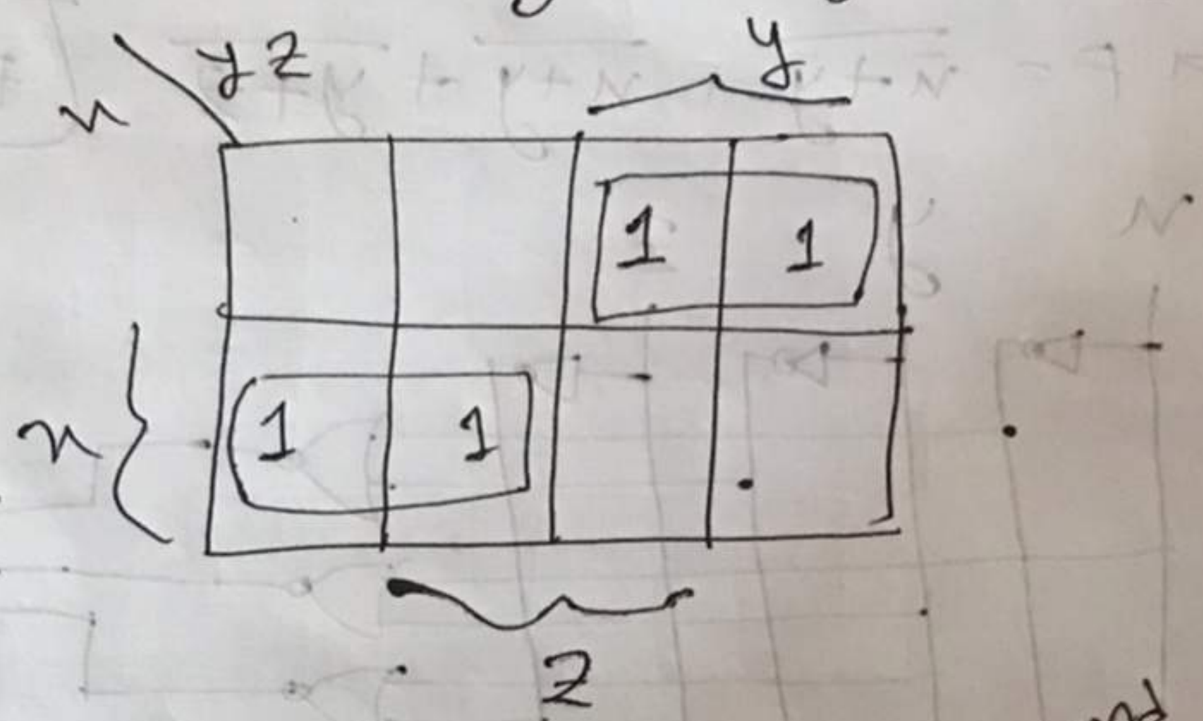


Subject:

Date:

Example: ① → standard form

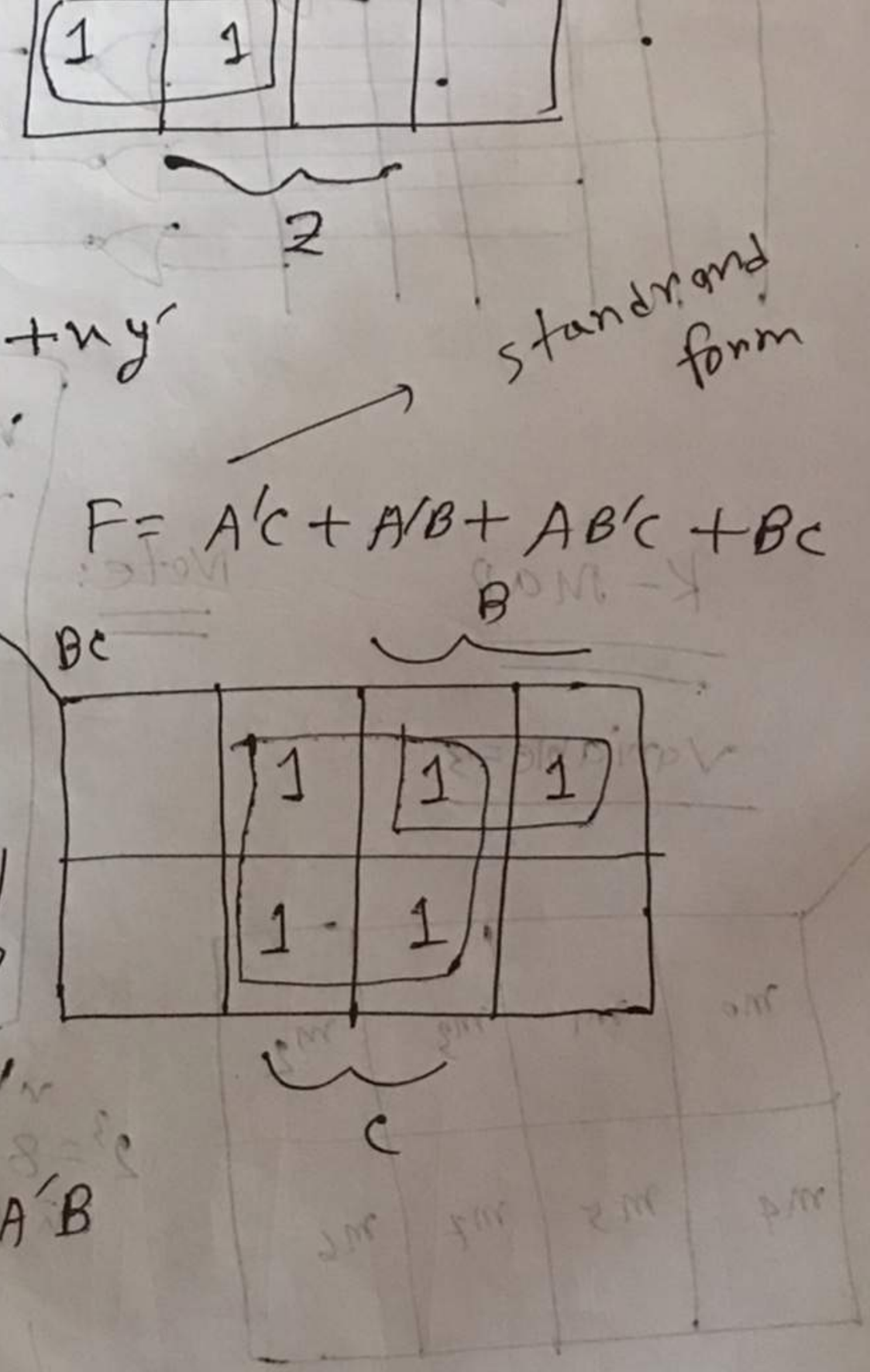
$F = x'yz + xy'z' + xy'z + xyz$



$F = x'y + xy'$ → standard form

Example: ②

$F = A'C + A'B + AB'C + BC$



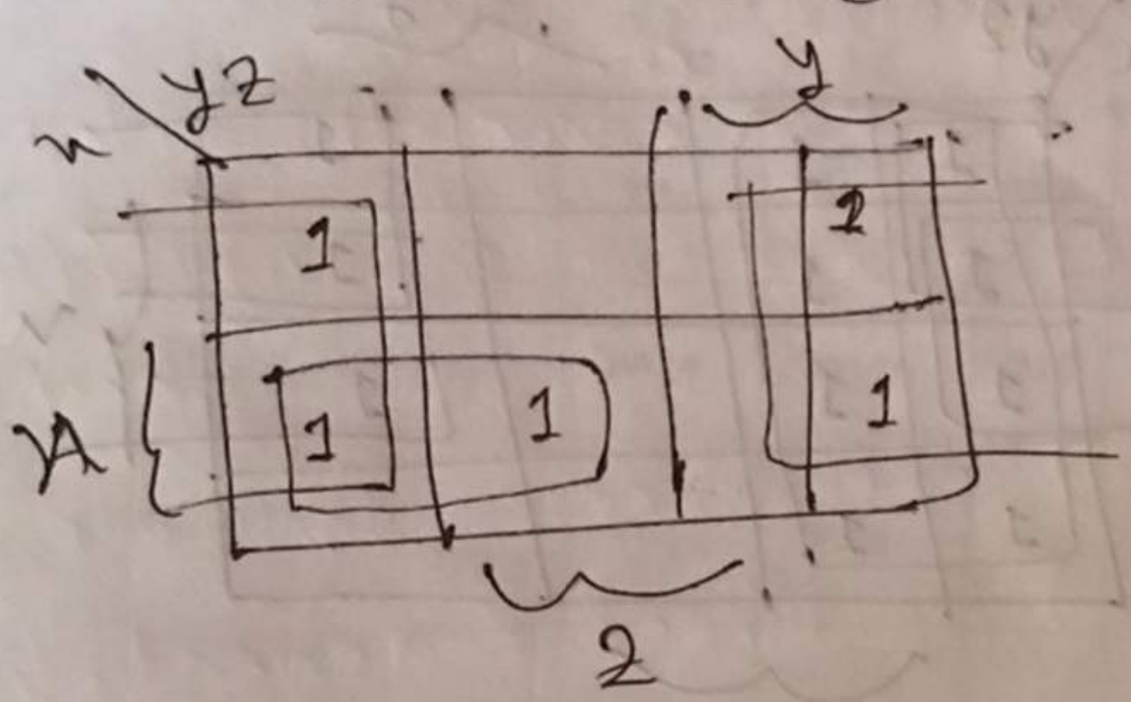
$F = C + A'B$

Subject:
Date:

Example: 3

$$F(x,y,z) = \sum(0,2,4,5,6)$$

canonical form

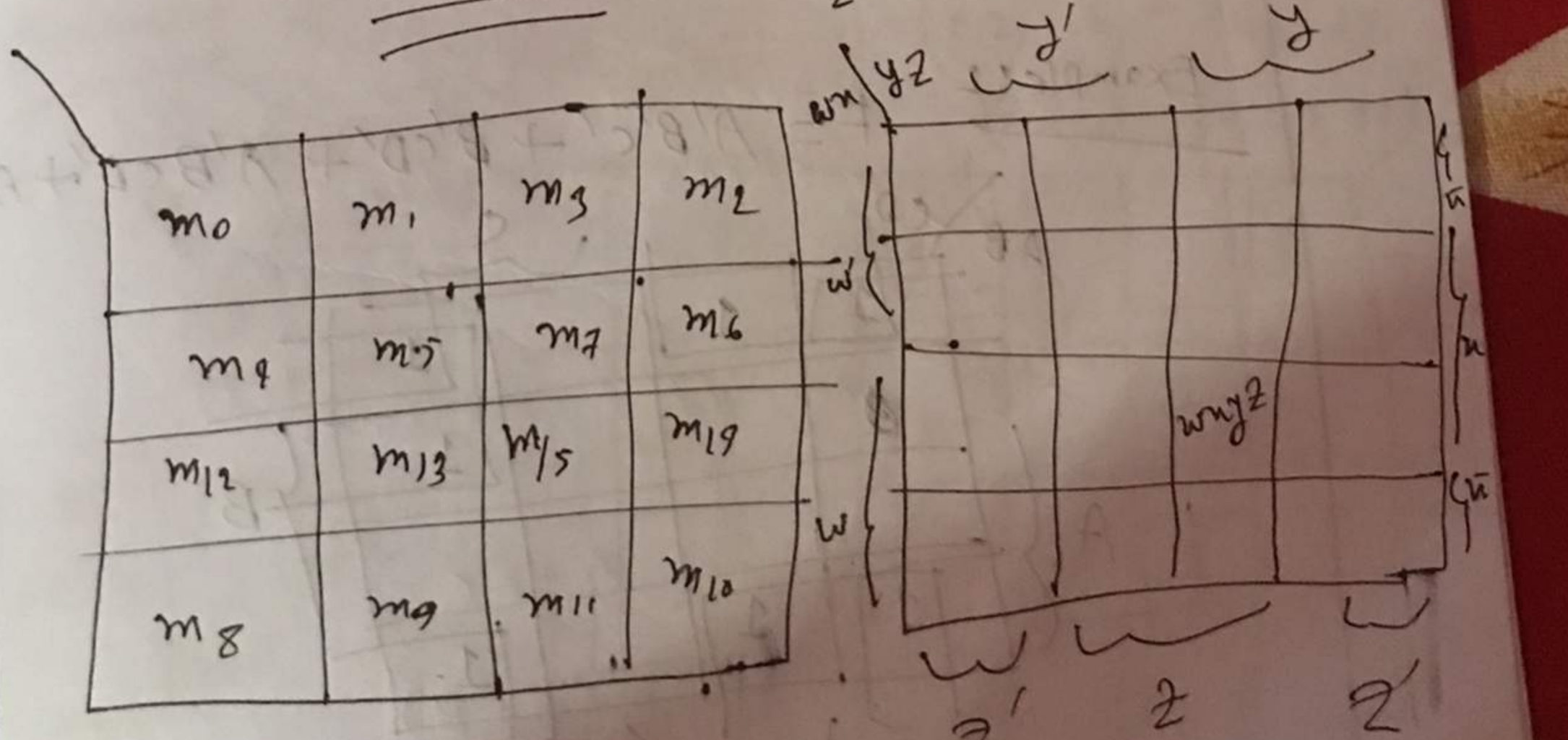


$$F = z' + xy'$$

4 variable

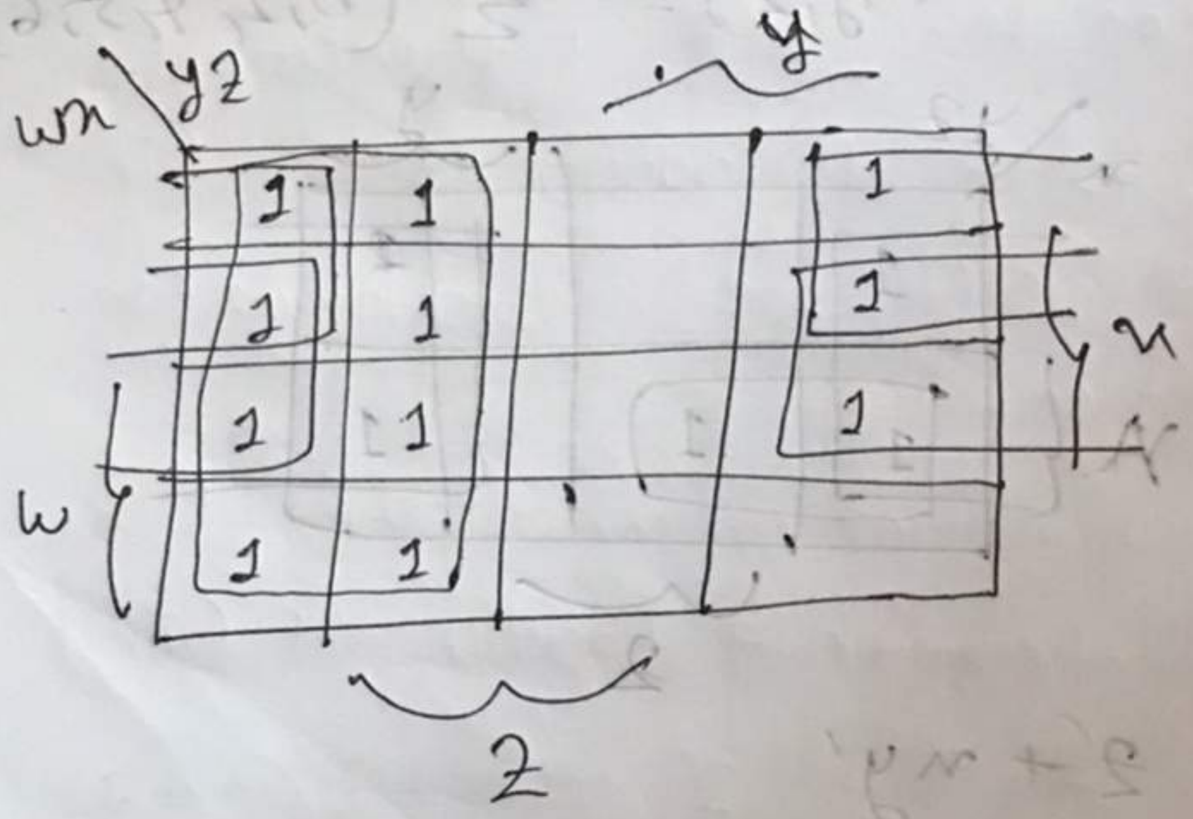
$2^4 = 16$

wxyz



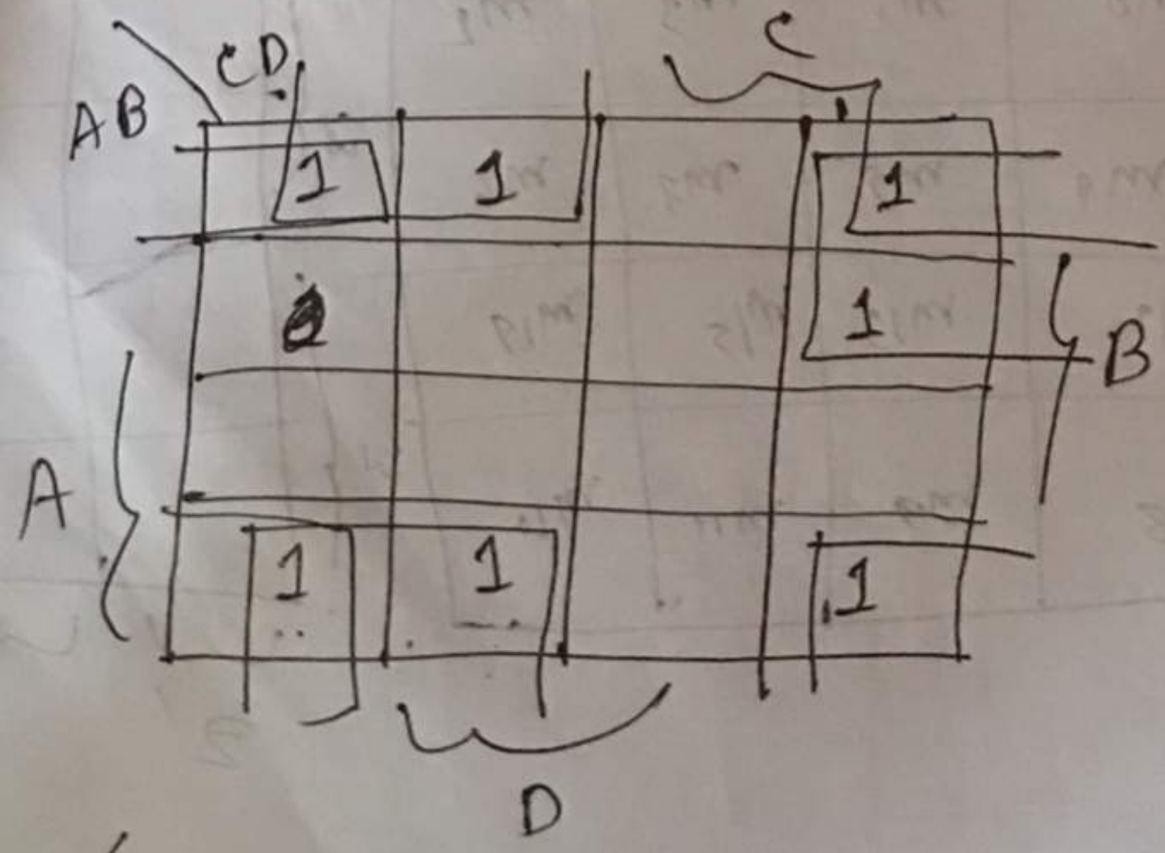
Subject:
Date:

Example: 1 $f(w,x,y,z) = \sum (0,1,2,4,5,6,8,9,12,13,14)$ ← canonical form



$F = y' + w'z' + xz'$

Example: 2 $F = A'B'C'D' + B'CD' + A'BCD' + AB'C'$



$F = B'C' + B'D' + A'CD'$

Subject:
Date:

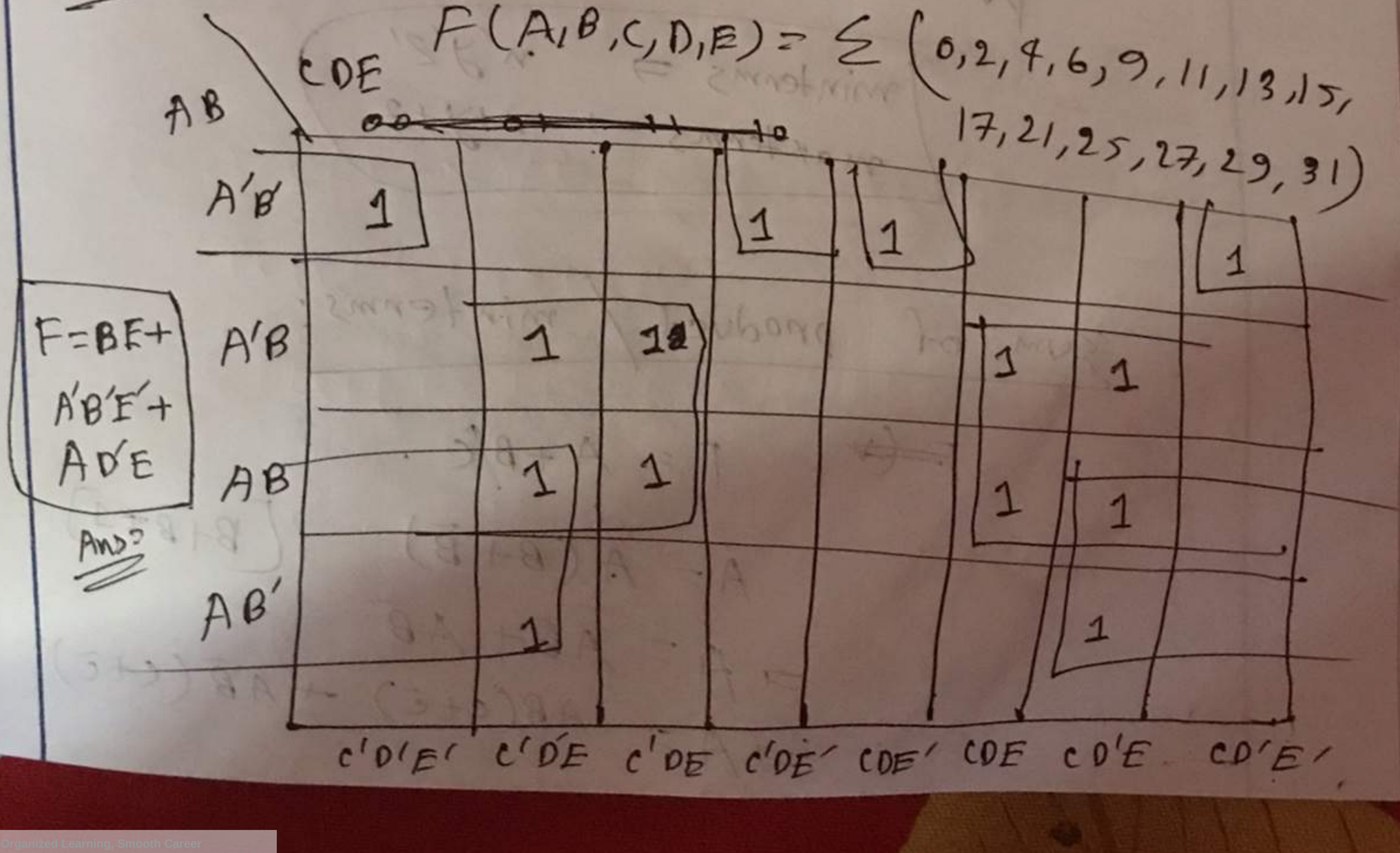
5 Variable

$A'BCDE'$ $2^5 = 32$ $A'B'C'D'E'$

AB \ CDE	$C'D'E'$ 000	$C'D'E$ 001	$C'DE$ 011	$C'DE'$ 010	CDE' 110	CDE 111	$CD'E$ 101	$CD'E'$ 100
$A'B'$ 00	$A'B'C'D'E'$ m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
$A'B$ 01	m_8	m_9	m_{11}	m_{10}	$A'BCDE'$ m_{14}	m_{15}	m_{13}	m_{12}
AB 11	m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
AB' 10	m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

Example : ①

$F(A,B,C,D,E) = \sum (0,2,4,6,9,11,13,15,17,21,25,27,29,31)$



Subject:
Date:

④ List Limitation of k-map ?

i) scalability ii) visual complexity
iii) Human error iv) Limited to Binary Logic
v) complex functions vi) Difficulty with don't cares
vii) Dependence on completeness
viii) No Direct Implementation.

⑤ Sum of product / minterms
product of sum / maxterms

minterms ⇒ $x'y'z'$
maxterms ⇒ $x+y+z$.

Sum of product / minterms:

$F = A + B'C$

$A = A(B+\bar{B})$ $[B+\bar{B}=1]$

$\Rightarrow A = AB + A\bar{B}$

$\Rightarrow A = AB(C+\bar{C}) + A\bar{B}(C+\bar{C})$

Subject:
Date:

→ $A = ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C}$

⇒ $\bar{B}\bar{C} = \bar{B}\bar{C}(A + \bar{A})$

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

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

$= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + \bar{A}\bar{B}C$

$= ABC + AB\bar{C} + A\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$

$[x + x = x]$

$= m_7 + m_6 + m_5 + m_4 + m_1$

$F(A,B,C) = \sum (1,4,5,6,7)$

product of sum / max terms

$F = x'y + x'z$

$= (x + x'z)(y + x'z)$

$= (x + x')(x + z)(y + x')(y + z)$

$x + x' = x + x' + y \cdot \bar{y}$ $[\because y \cdot \bar{y} = 0]$

$= (x + x' + y)(x + x' + y')$

Subject:
Date :

~~$x+z = x+z$~~.

$$= (x+z)(y+x')(y+z)$$

$$x+z = x+z+y\cdot\bar{y} \qquad [y\cdot\bar{y}=0]$$

$$= (x+z+y)(x+z+\bar{y})$$

$$y+x' = y+x'+z\cdot\bar{z}$$

$$= (y+x'+z)(y+x'+\bar{z})$$

$$y+z = y+z+x\cdot\bar{x}$$

$$= (y+z+x)(y+z+\bar{x})$$

$$F = (x+y+z)(x+\bar{y}+z)(\bar{x}+y+z)(\bar{x}+y$$

$$(x+y+z)(\bar{x}+y+z)$$

$$= (x+y+z)(x+\bar{y}+z)(\bar{x}+y+z)(\bar{x}$$

$$= M_0\, M_2\, M_4\, M_5.$$

$$F(x,y,z) = \pi(0,2,4,5)$$

Subject:
Date:

(5) Simplify the following Boolean function D in product of sum—

~~⇒ D = BDE′ + CD′E′~~

~~⇒ D = (BDE′ + BCD′E′)~~

~~⇒ D = BDE′+~~

Solve : D = BDE′ + CD′E′

⇒ D = (B + CD′E′) (D + CD′E′) (E′ + CD′E′)

⇒ D = (B + C) (B + D′) (B + E′) (D + C) (D + D′)

(D + E′) (E′ + C) (E′ + D′) (E′ + E′)

⇒ D = (B + C) (B + D′) (B + E′) (D + C) (D + F′)

(E′ + C) (F′ + D′) · E′

$B + C = B + C + A \cdot \bar{A}$

$= (B + C + A)$

$(B + C + \bar{A})$

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

$= (B + C + A + D)$

$(B + C + A + \bar{D})$

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

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

$(B + C + A + \bar{D} + \bar{E})$

(6)

A	B	C	
1	1	0	

A B C′ ⇒ m_6

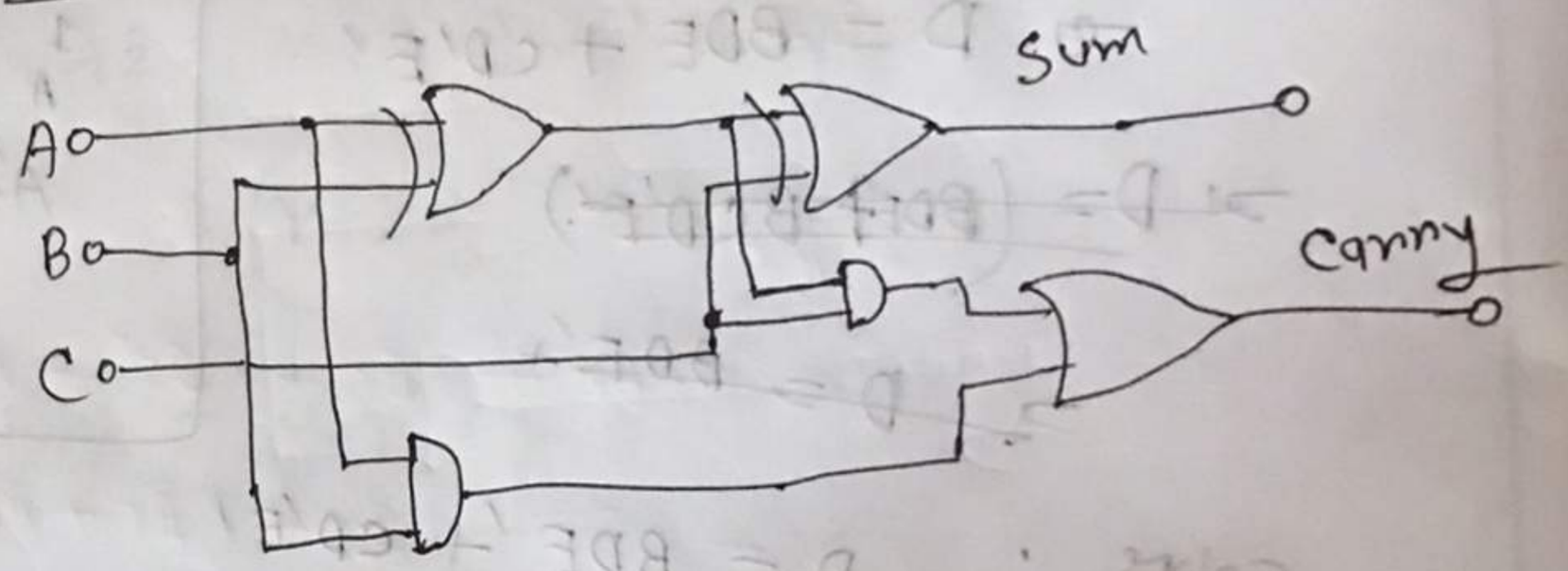
A′ + B′ + C ⇒ M_6

এভাবে চলবে।

Subject:
Date:

(6) Design a Full adder?

Circuit Diagram :-



For sum: Sum = $\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$

$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$

$= A \oplus B \oplus C$

For Carry: Carry $= \bar{A}BC + A\bar{B}C + ABC + ABC$

$= C(\bar{A}B + A\bar{B}) + AB(C + C)$

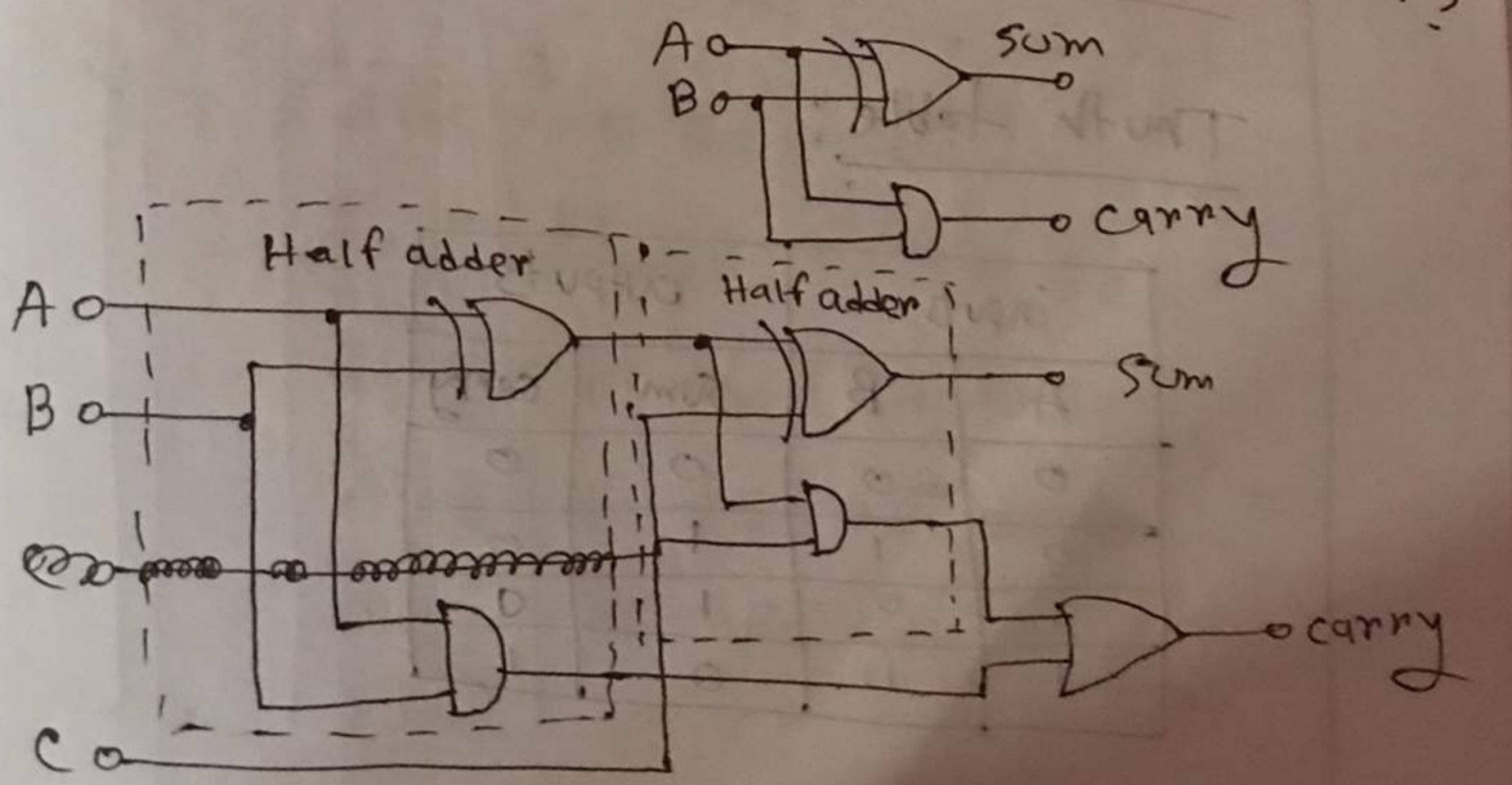
$= C(A \oplus B) + AB$

Subject:
Date:

Truth table:

inputs			outputs	
A	B	C	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

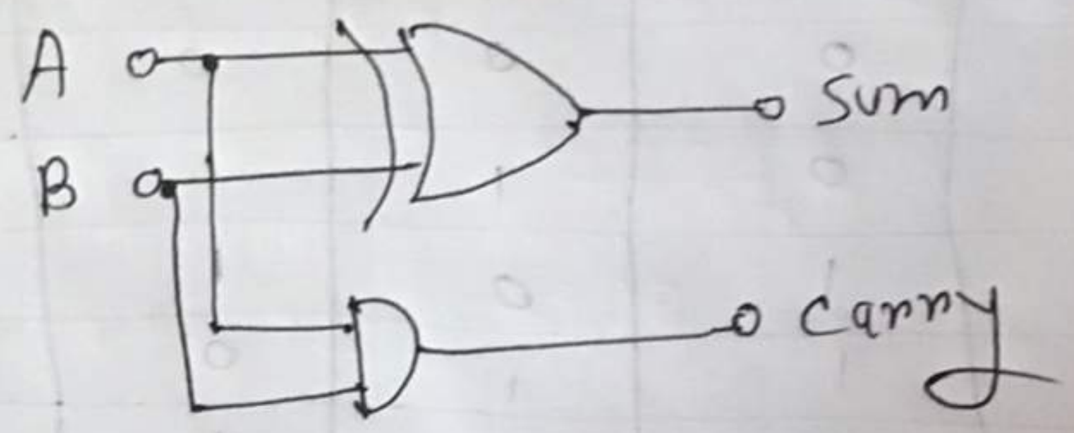
(7) Draw a FULL adder using two half adders



Subject:
Date:

④ Design Half adder ?

Circuit Diagram:



For Sum: $Sum = \bar{A}B + A\bar{B}$

$= A \oplus B$

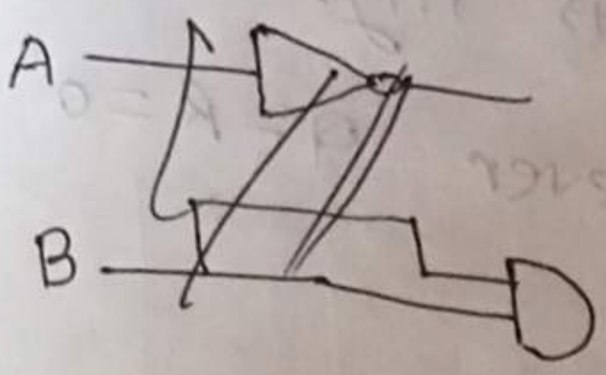
For carry: $Carry = AB$

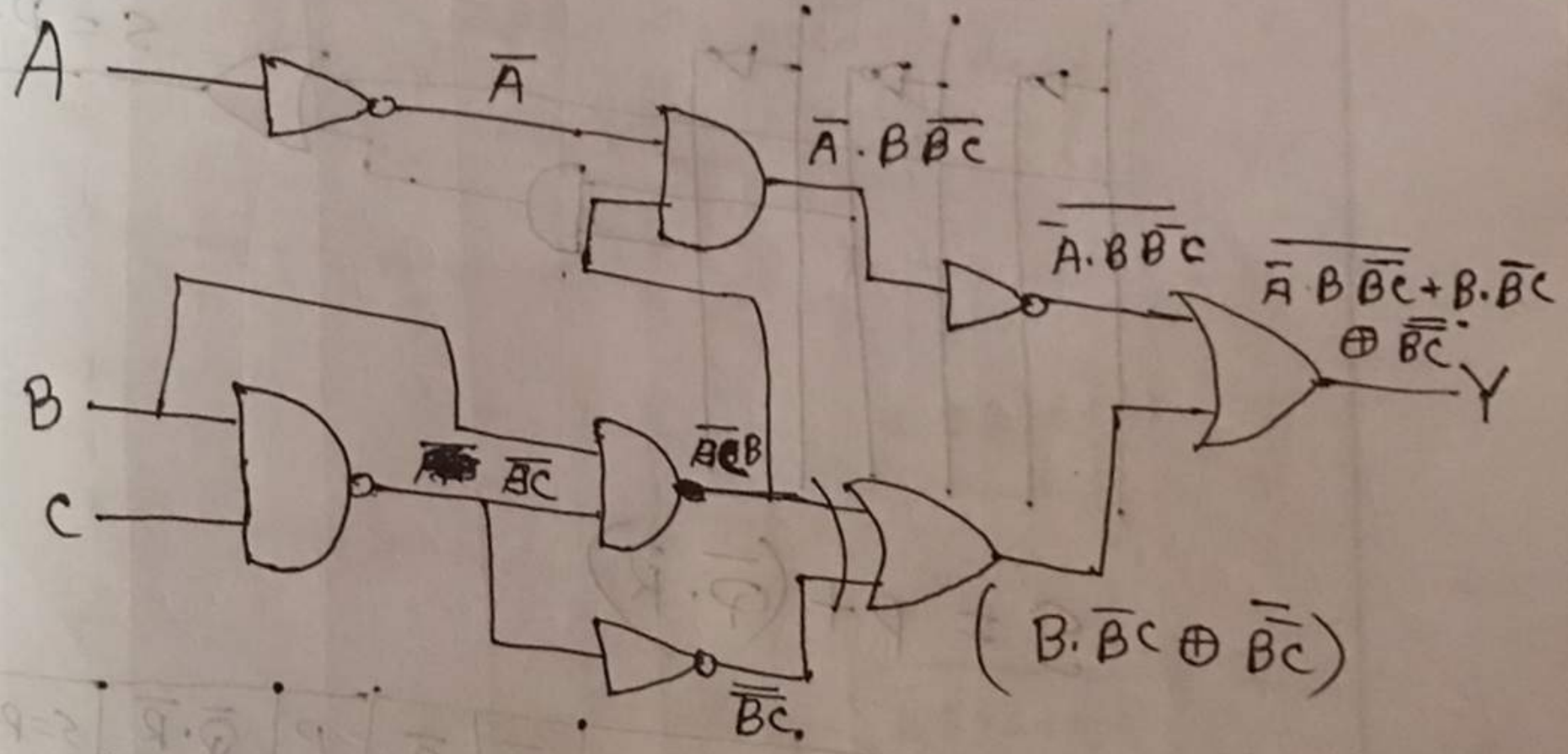
Truth table:

inputs		outputs	
A	B	Sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Subject:
Date:

(8) Simplify the logic circuit ?





$$Y = \overline{\bar{A}B\bar{B}C} + (B \cdot \bar{B}C \oplus \bar{\bar{B}C})$$

$$= \bar{\bar{A}} + \bar{B} + \bar{\bar{B}C} + \{B(\bar{B}+\bar{C}) \oplus BC\}$$

$$= A + \bar{B} + BC + \{B\bar{B} + B\bar{C} \oplus BC\}$$

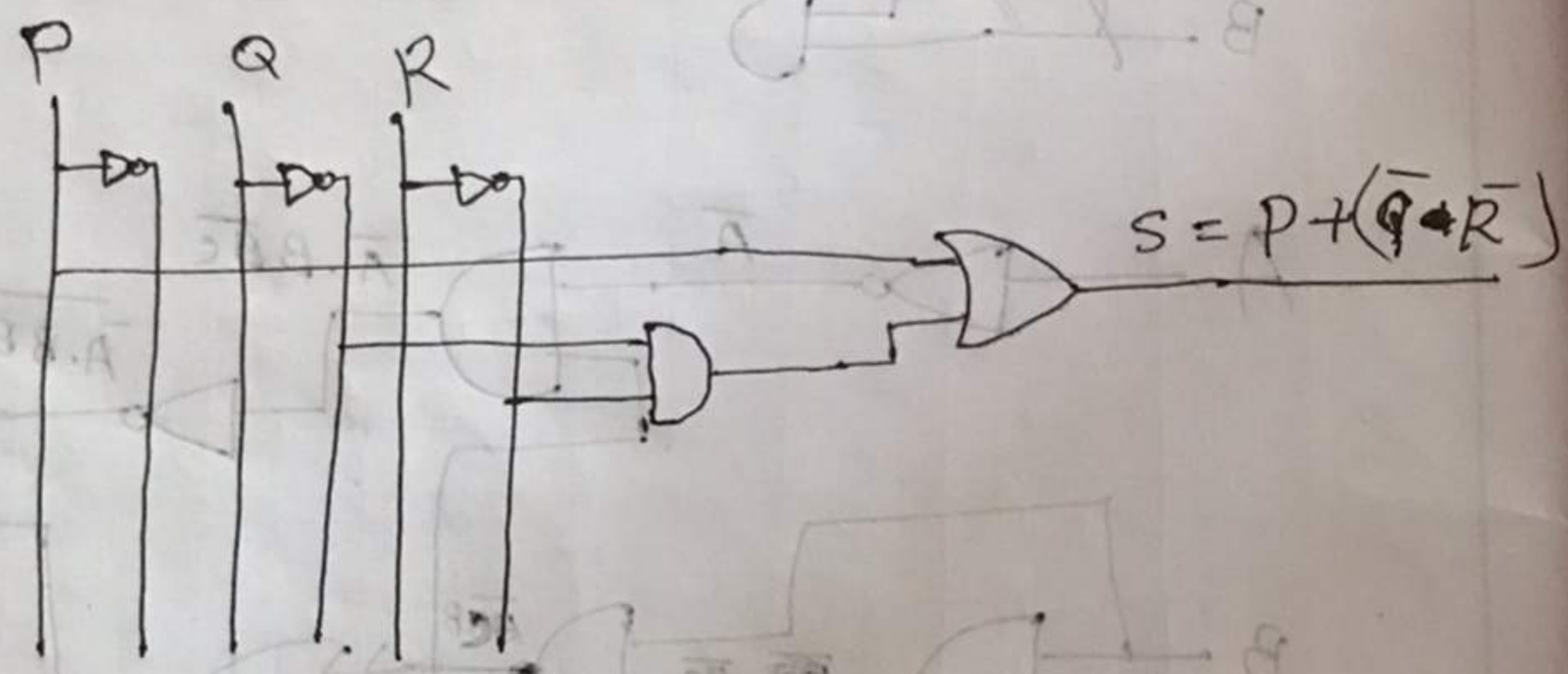
$$= A + \bar{B} + BC + (B\bar{C} \oplus \bar{B}C)$$

$$= A + \bar{B} + BC + (B\bar{C} \oplus BC)$$

Subject:

Date:

⑨ Design a logic circuit with input P, Q, R so that output S is High whenever P is "1" or whenever $Q = R = 0$

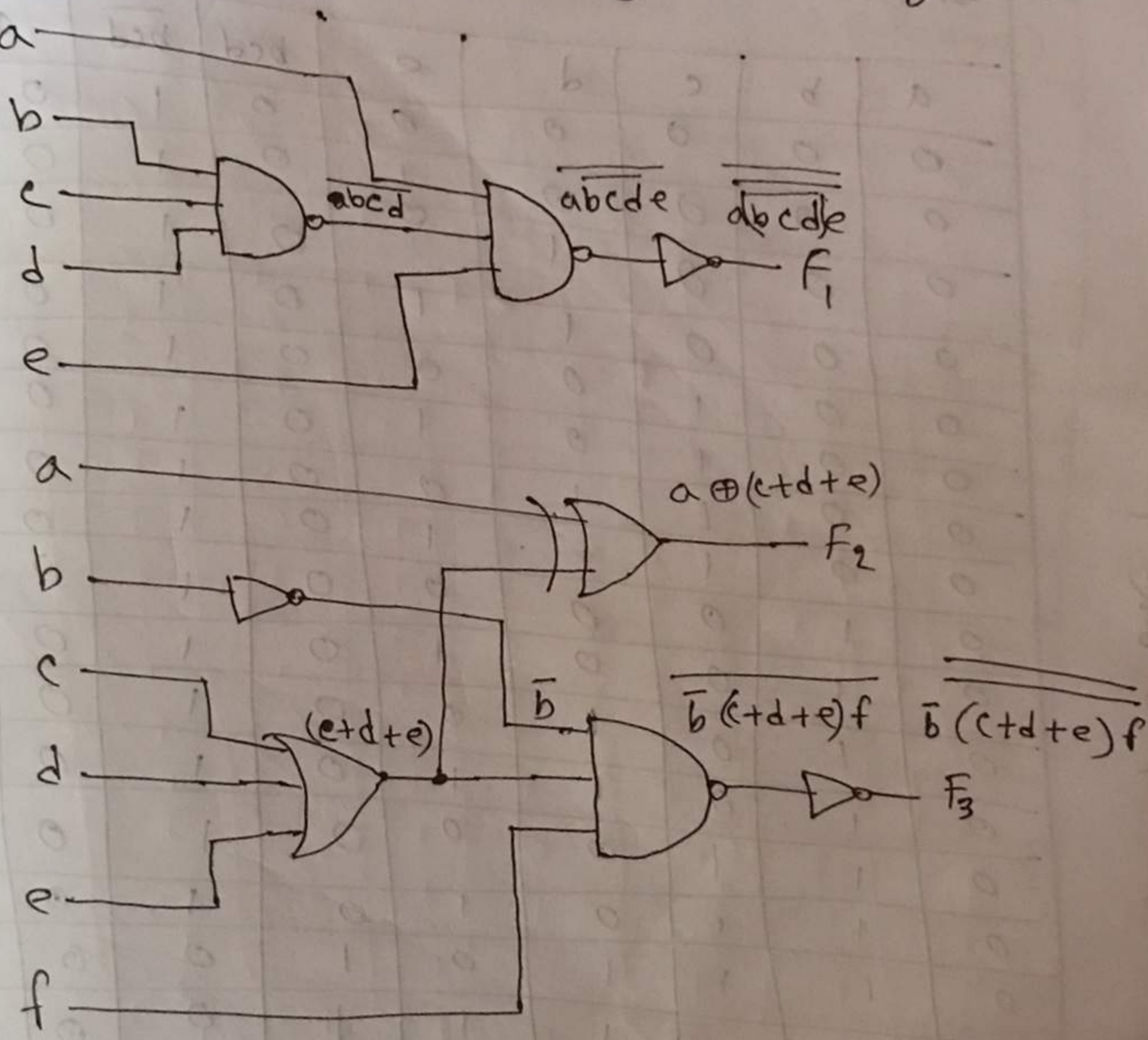


$$S = P + (\bar{Q} \cdot \bar{R})$$

	P	Q	R	$\bar{Q}$	$\bar{R}$	P	$\bar{Q} \cdot \bar{R}$	$S = P + (\bar{Q} \cdot \bar{R})$
→	0	0	0	1	1	0	1	1
	0	0	1	1	0	0	0	0
	0	1	0	0	1	0	0	0
	0	1	1	0	0	0	0	0
→	1	0	0	1	1	1	1	1
→	1	0	1	1	0	1	0	1
→	1	1	0	0	1	1	0	1
→	1	1	1	0	0	1	0	1

Subject:
Date:

(10) write Boolean expressions and construct the truth tables describing the outputs of the circuits described by the following logic diagrams?



Subject:

Date :

$F_1 = \overline{\overline{a(bcd)e}}$

$= F_1 = a\overline{bcd}e = ae\,\overline{bcd}$

a	b	c	d	e	bcd	$\overline{bcd}$	ae	$\overline{bcd}ae$
0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	1	0	0
0	0	0	1	0	0	1	0	0
0	0	0	1	1	0	1	0	0
0	0	1	0	0	0	1	0	0
0	0	1	0	1	0	1	0	0
0	0	1	1	0	0	1	0	0
0	0	1	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0
0	1	0	0	1	0	1	0	0
0	1	0	1	0	0	1	0	0
0	1	0	1	1	0	1	0	0
0	1	1	0	0	0	1	0	0
0	1	1	0	1	0	1	0	0
0	1	1	1	0	0	1	0	0
0	1	1	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0
1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	0	1	1	0	1	0	0
1	0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	0	0
1	0	1	1	0	0	1	0	0
1	0	1	1	1	0	1	0	0
1	1	0	0	0	0	1	0	0
1	1	0	0	1	0	1	0	0
1	1	0	1	0	0	1	0	0
1	1	0	1	1	0	1	0	0
1	1	1	0	0	0	1	0	0
1	1	1	0	1	0	1	0	0
1	1	1	1	0	0	1	0	0
1	1	1	1	1	0	1	1	1

32 গুণিত হবে।

Subject:
Date:

$F_2 = a \oplus (c+d+e)$

a	c	d	e	c+d+e	a ⊕ (c+d+e)
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	0

Subject:

Date:

F_{32} $\overline{\overline{\bar{b}\,(c+d+e)\,f}}$

$\Rightarrow F_3 = \bar{b}f\,(c+d+e)$

b	c	d	e	f	$\bar{b}$	$\bar{b}f$	c+d+e	$\bar{b}f(c+d+$
0	0	0	0	0	1	0	0	0
0	0	0	0	1	1	1	1	0
0	0	0	1	0	1	0	1	0
0	0	0	1	1	1	1	1	1
0	0	1	0	0	1	0	1	0
0	0	1	0	1	1	1	1	1
0	0	1	1	0	1	0	1	0
0	0	1	1	1	1	1	1	1
0	1	0	0	0	1	0	1	0
0	1	0	0	1	1	1	1	1
0	1	0	1	0	1	0	1	0
0	1	0	1	1	1	1	1	1
0	1	1	0	0	1	0	1	0
0	1	1	0	1	1	1	1	1
0	1	1	1	0	1	0	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	1	0

32 মানের টেবিল।

Subject:
Date:

(11) Show that a p tha @ positive logic NAND gate is a negative logic NOR gate and vice versa.

Definitions:

1) positive Logic: i) Logic Level '1' (true) is represented by a high voltage

ii) Logic Level '0' (false) is represented by a low voltage.

2) Negative Logic: i) Logic Level '1' (true) is represented by a low voltage.

ii) Logic Level '0' (false) is represented by a high voltage

NAND gate: $P = \overline{A \cdot B}$

NOR gate: $P = \overline{A + B}$

Subject:
Date:

NAND gate (positive)

A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

In positive
The output is high
(1) for all combinations
of inputs except when
both inputs are high
(1)

NOR gate (negative)

A	B	$\overline{A+B}$
1	1	0
1	0	0
0	1	0
0	0	1

In negative,
The output is low
(0) only when
baths inputs are
high (1).

For NAND gate in positive logic:

when both inputs are low (0), the output is high (1), and when both inputs are high (1), the output is low (0).

Subject:
Date:

For NOR gate in negative Logic:

When both inputs are high(1), the output is low (0), and when both inputs are ~~high(1)~~, low(0), the output is high (1).

Thus, we can see that a positive logic NAND gate behaves like a negative logic NOR gate and vice versa.

Extra:

AND (positive)

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

AND (negative)

A	B	$\overline{AB}$
1	1	1
1	0	1
0	1	1
0	0	0

NAND (positive)

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

NAND (Negative

A	B	$\overline{AB}$
1	1	0
1	0	0
0	1	0
0	0	1

Subject:.....

Date:.....

OR (positive)

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

OR (negative)

A	B	A+B
1	1	1
1	0	0
0	1	0
0	0	0

NOR (positive)

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

NOR (Negative)

A	B	$\overline{A+B}$
1	1	0
1	0	1
0	1	1
0	0	1

Dual Form

AB (positive) ⟶ AB (Negati
or A+B (pos

A+B (positive) → A+B (negative
or AB (positive)

যুগ থাকলে যোগ।
যোগ থাকলে যুগ।

Subject:
Date:

(12) show that the dual form of the XOR is also its complement?

Soln: $x \oplus y = x \cdot y' + x' y$

Complement $= (x \oplus y)' = (xy' + x'y)'$

$= \overline{xy' + x'y}$

$= \overline{xy'}\ \overline{x'y}$

$= (x' + y)(x + y')$

Dual of $= (x \oplus y)$ = Dual of $(xy' + x'y)$

$= (x + y')(x' + y)$

$= (x' + y)(x + y')$

$\therefore (x \oplus y)' =$ Dual of $(x \oplus y)$

Subject:

Date:

(13) Implement the following Boolean expression with Exclusive-OR and AND gate :

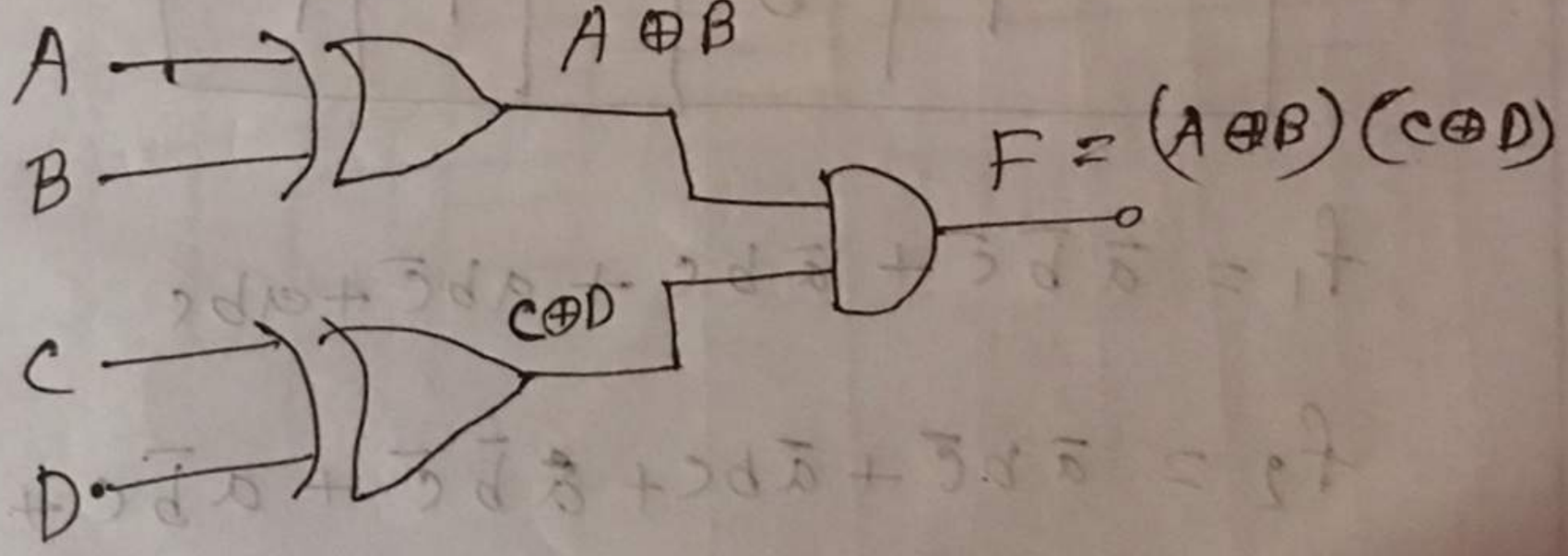
$$F = AB'CD' + A'BCD' + AB'C'D + A'BC'D$$

$$\Rightarrow F = C'D'(AB' + A'B) + C'D(AB' + A'B)$$

~~$\Rightarrow F = (A'B + AB')$~~

$$\Rightarrow F = (AB' + A'B)(CD' + C'D)$$

$$\Rightarrow F = (A \oplus B)(C \oplus D)$$



Subject:
Date:

(19) write the Boolean expression and draw the logic circuit whose outputs are by the following truth table?

f_1	f_2	a	b	c
1	0	0	0	0
0	0	0	0	1
0	1	0	1	0
1	1	0	1	1
0	1	1	0	0
0	1	1	0	1
1	1	1	1	0
1	0	1	1	1

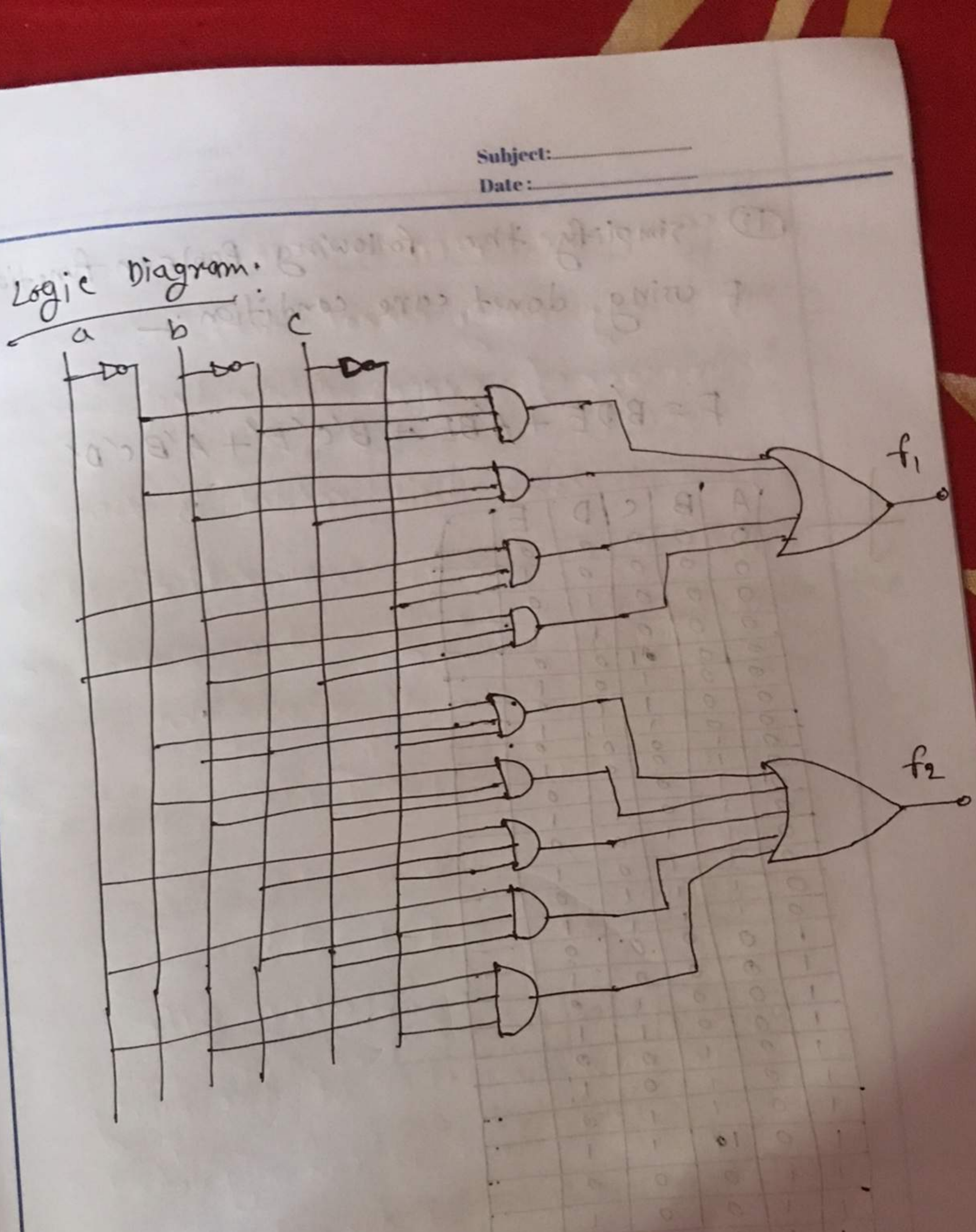
$f_1 = \bar{a}\bar{b}\bar{c} + \bar{a}bc + ab\bar{c} + abc$

$f_2 = \bar{a}b\bar{c} + \bar{a}bc + \bar{a}\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$

Subject:

Date :

Logic Diagram:

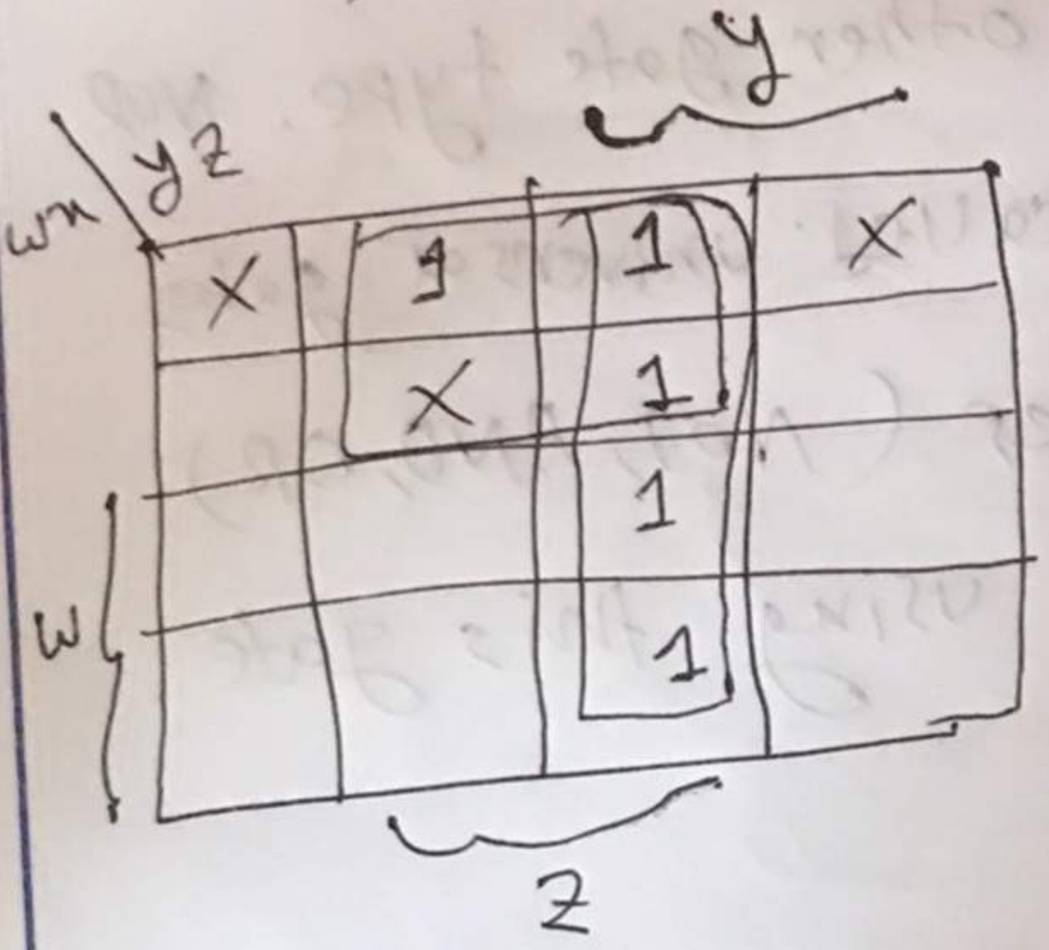


data with parity bit.

Extra: K-Map (Don't Care conditions)

Simplif the boolean function: $F(w,x,y,z) = \Sigma(1,3,7,11,15)$

$d(w,x,y,z) = \Sigma(0,2,5)$



$F = yz + w'z$

[Combining 1's and X's]

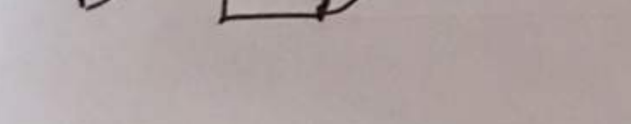
NAND gate is a universal gate?

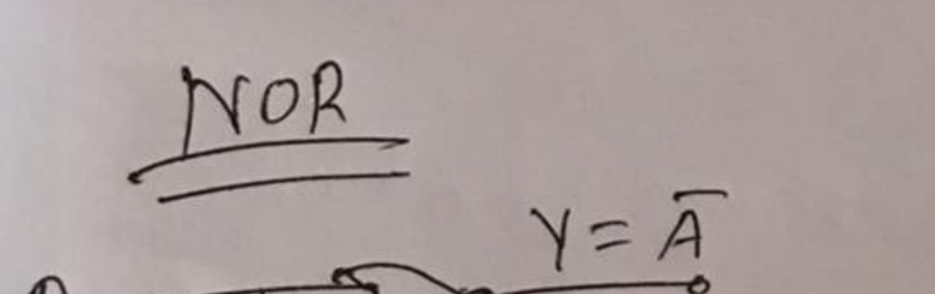
① A universal gate is a gate which can implement any Boolean function without need to use any other gate type. NOR and NAND gate is called universal gates because all the gates (NOT, AND, OR) can be created by using this gate.

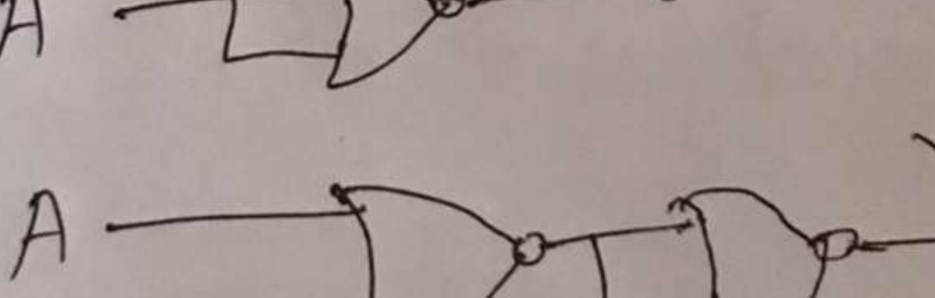
Subject:
Date:

Implement :

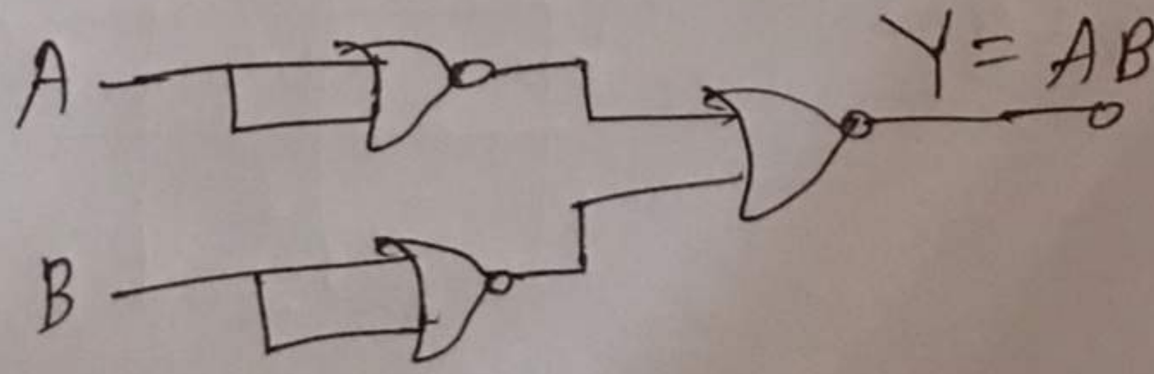
NAND





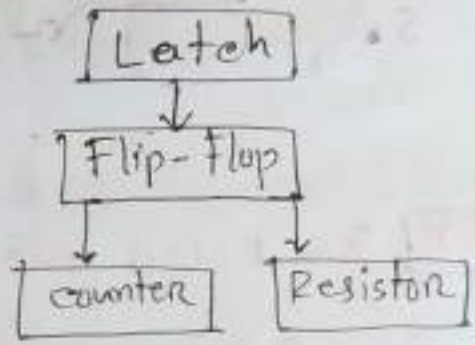


NOR



Latches (DLD)

Latch: Latches are the basic building blocks of any flip flop and they are capable of holding 1 bit unit necessary.

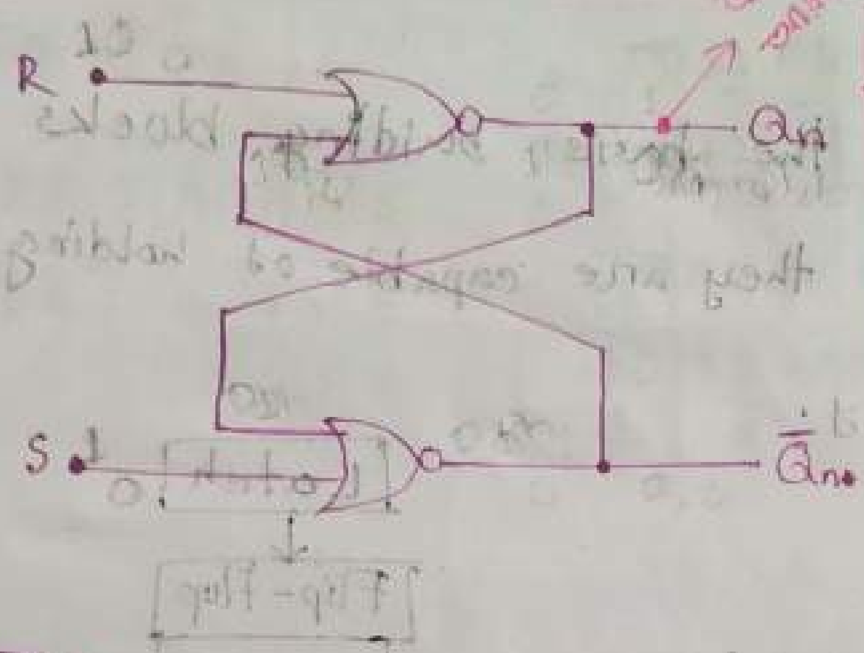


NOR Latch & NAND ~~gate~~ Latch:

The SR Latch is a circuit with two cross-coupled NOR gates / two cross-coupled NAND gates and two inputs labeled S for 'Set' and R for 'Reset', Outputs Q_n and $\overline{Q_n}$ are the complement of each other.

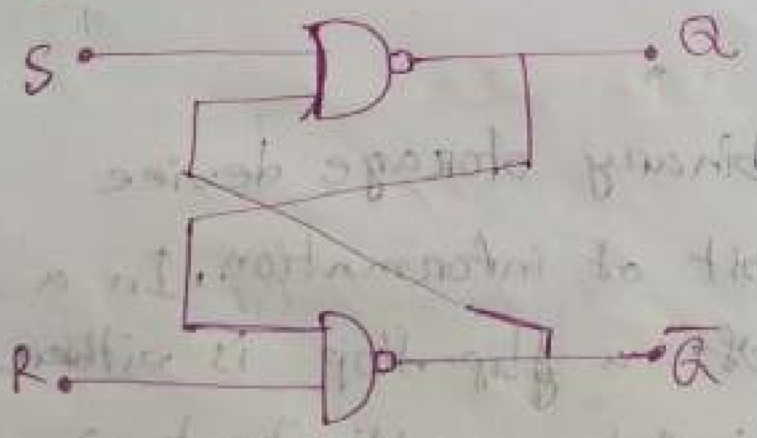
** A Flip-Flop is a binary storage device capable of storing one bit of information. In a stable state the output of a flip-flop is either 0 or 1 (it is called bi-state multi-vibrator).

NOR:



S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	×
1	1	1	×

Q_{n+1} হলো memory এর মতো যেটা input হিসাবে দিবো সেটাকে আবার output হিসাবে ফেরত দিবো। $\bar{Q}_n$, Q_n এর complement. যদি Q_n and $\bar{Q}_n$ same result output হিসেবে দিবে তাহলে বুঝতে হবে circuit কাজ করছে না। R and S control key হিসেবে কাজ করে।



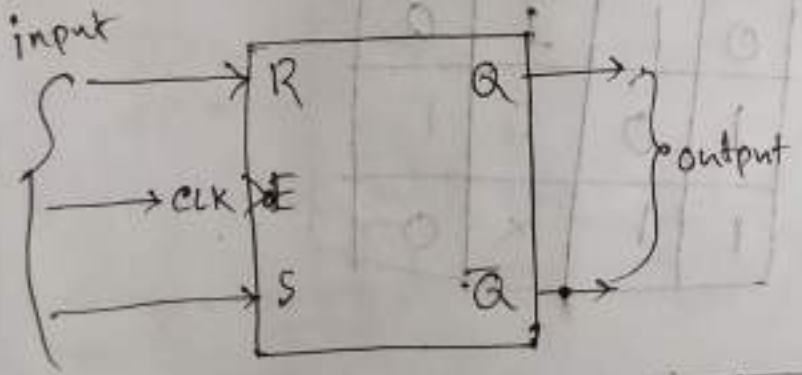
S	R	Q_n	Q_{n+1}
0	0	0	×
0	0	1	×
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

SR - flip-flop

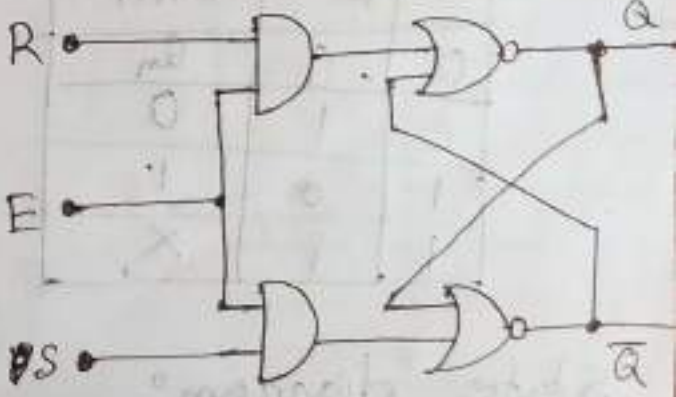
SR flip-flop: It is a flip-flop with two inputs, one is S (set) and another is R (Reset). Set basically indicates set the flip-flop which means output 1 and reset do the complement of set (output = 0).

Here a clock pulse is supplied to operate the flip flop.

Block diagram:



Circuit diagram



Truth Table:

	S	R	Q_n	Q_{n+1}
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	x
7	1	1	1	x

	sr	s'r'	s'r	sr	sr'
q		00	01	11	10
q'	00			X	1
q	01	1		X	1

characteristics equation:

$$Q_{n+1} = S + R'Q_n$$

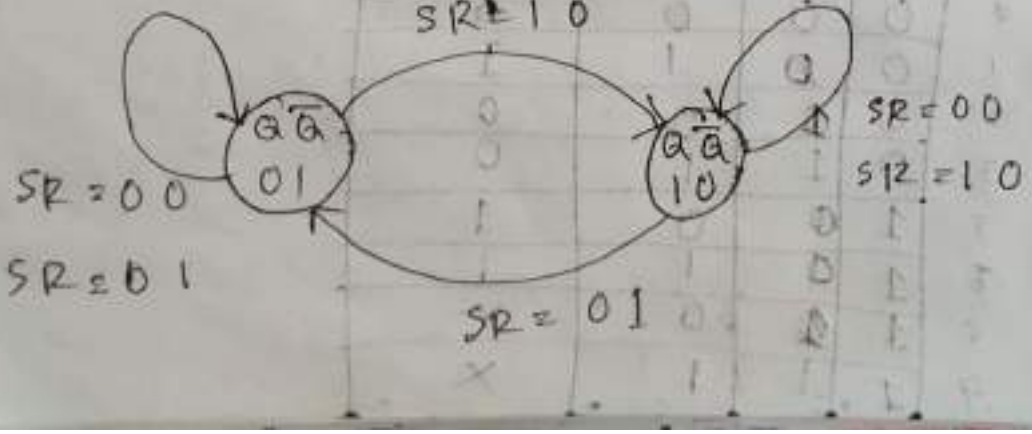
Functional Table:

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	X

Excitation table:

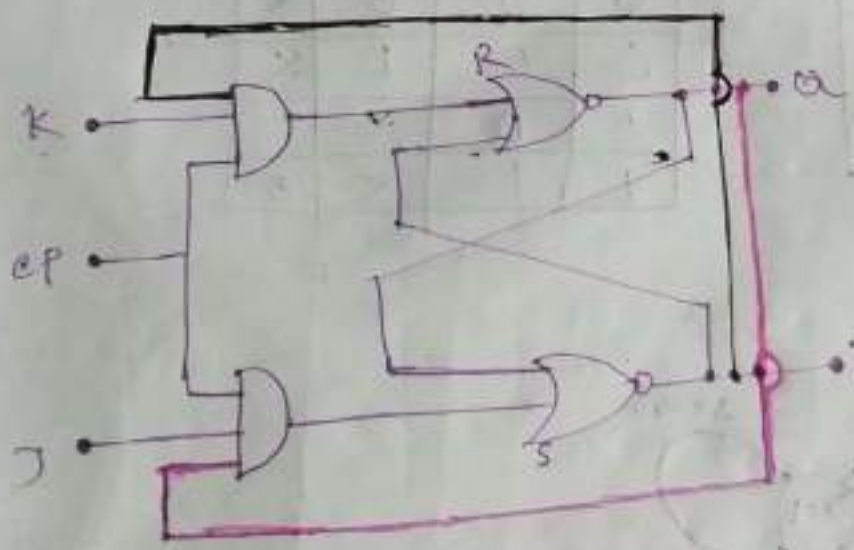
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

State diagram:



JK - flip-flop.

to solve the invalid input state $S = R = 1$ of SR flip flop we use JK flip-flop.



J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

0 output $Q_{n+1} = 1$

1 " " $Q_{n+1} = 0$

Block diagram

K-Map

	JK	J'K'	J'K	JK	JK'
q		00	01	11	10
q'	00	0	0	1	1
q	01	1	0	0	1

Characteristics Eqn:

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

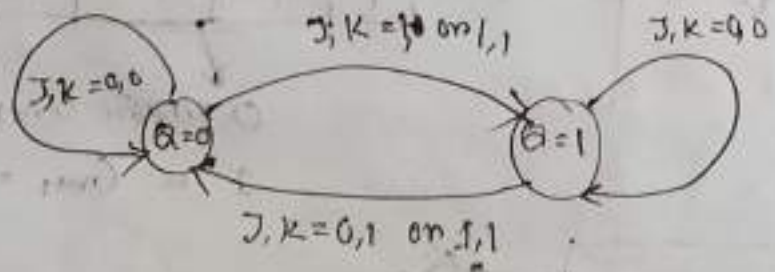
Functional table:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

Excitation table:

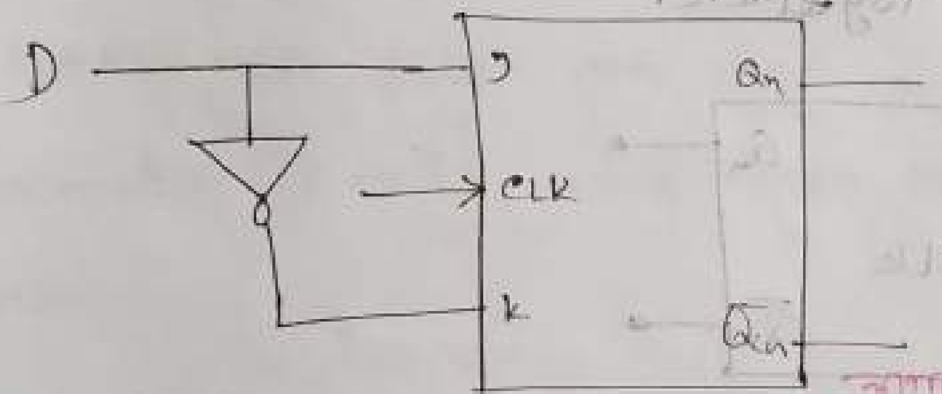
Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

State diagram:



D flip-flop

The D (Data/Delay) flip-flop, tracks the input at D and produce the same value as output.



Qn আমাদের দেওয়া থাকবে না। D এ যা input দিব Qn+1 এ তাই output পাবো।

D

J	K	Qn	Qn+1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J-K flip-flop #

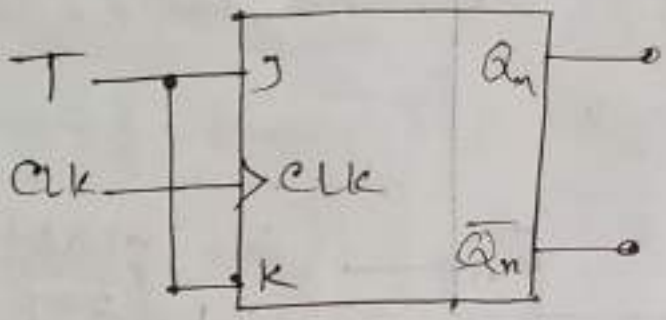
J and K এর same input বাদ যাবে

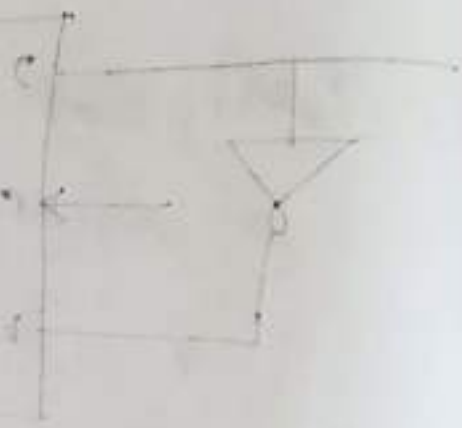
বাকি input গুলোর মধ্যে J এর মান D হিসেবে নিব।

D	Qn	Qn+1
0	0	0
0	1	0
1	0	1
1	1	1

T (toggle) flip flop

The T (toggle) flip-flop is a complement of flip-flop and can be obtained from J-K flip-flop when inputs J and K are ~~same~~ tied together.





J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J and K are different input

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

$$Q_{n+1} = T \oplus Q_n$$

1. Explain how a NAND/NOR latch store a bit. What is their limitation? (20-21, 19-20) [3]

Answer:

A NAND/NOR latch can store a single bit of information (0 or 1) using cross-coupled logic gates. In those latch the inputs are indicated by S(set) and R(reset) and outputs are indicated by Q_n and $|Q_n$.

The process of storing bit of NAND Latch:

1. When S = R = 1, the latch maintains the current state.
2. When S = 0 and R = 1, the output = 1 means set the latch.
3. When S = 1 and R = 0, the output = 0 means reset the latch.

S (set)	R (reset)	Q_n	Q_{n+1}
0	0		invalid
0	0		invalid
0	1		1
0	1		1
1	0		0
1	0		0
1	1		unchanged
1	1		unchanged

Limitations:

1. When S = R = 0, the output = invalid and latch throws error.
2. When $Q_n = |Q_n$ = same bit of value (0 or 1), the latch level-triggered and it goes to race condition.

The process of storing bit of NOR Latch:

1. When S = R = 0, the latch maintains the current state.
2. When S = 0 and R = 1, the output = 0 means reset the latch.
3. When S = 1 and R = 0, the output = 1 means set the latch.

S (set)	R (reset)	Q_n	Q_{n+1}
0	0	0	unchanged
0	0	1	unchanged
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	invalid
1	1	1	invalid

Limitations:

1. When S = R = 1, the output = invalid and latch throws error.
2. When $Q_n = |Q_n$ = same bit of value (0 or 1), the latch level-triggered and it goes to race condition.

2. What are the limitations of S-R flip flop? How can be resolved these? (20-21, 18-19) [4]

The limitations and resolutions of S-R flip-flop are given below:

Indeterminate State:

- Limitation: When both inputs (S and R) = 1, the output becomes unstable.
- Solution: Use a J-K flip-flop, which toggles instead of becoming undefined when both inputs are high.

Lack of Clock Control:

- Limitation: Basic S-R flip-flops are level-triggered and may not reliably capture input changes.
- Solution: Use a Clocked S-R flip-flop or a D flip-flop for stable, edge-triggered responses.

Glitches from Asynchronous Inputs:

- Limitation: Small timing differences can cause unstable output.
- Solution: Use a Clocked or Master-Slave flip-flop to synchronize inputs and prevent glitches.

Limited in Complex Circuits:

- Limitation: The S-R flip-flop's simplicity restricts its use in advanced circuits.
- Solution: Use more versatile flip-flops like D or J-K flip-flops in complex designs.

3. Define race-around condition in J-K flip-flop? How can you overcome the problem? Explain with appropriate figure and waveforms. (20-21, 19-20) [5]

Answer:

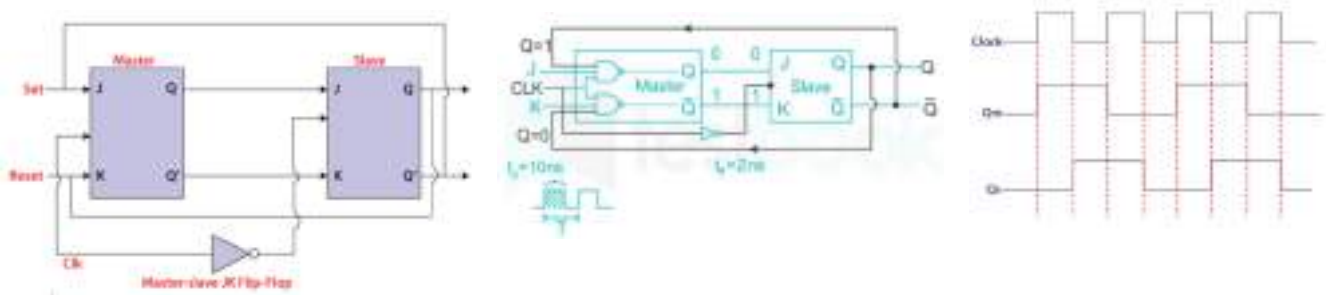
In JK Flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop unstable or uncertain. This problem is called race around condition in J-K flip-flop.

The truth table of the J-K flip-flop is given below:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	invalid

The invalid at J = K = 1 is known as the race-around condition.

To overcome this condition, a Master-slave JK filp-flop is used.



Explaination:

1. The master-slave flip flop is constructed by combining two JK flip flops.
2. These flip-flops are connected in a series configuration. In these two flip flops, the 1st flip flop work as a "master", called the master flip flop, and the 2nd work as a "slave", called the slave flip flop.
3. When the clock pulse is true, the slave flip flop will be in the isolated state, and the system's state may be affected by the J and K inputs. The "slave" remains isolated until the CP is 1. When the CP is set to 0, the master flip-flop passes the information to the slave flip-flop to obtain the output.

4. How a NAND/NOR latch store a bit? (19-20) [2]

Answer:

NAND Latch

1. Setup: Made of two cross-connected NAND gates.
2. Inputs: S(Set) and R(Reset).
3. Operation:
    - Set: S = 0 and R = 1, makes output Q = 1.
    - Reset: S = 1, R = 0, makes output Q = 0.
    - Hold: S = 1, R = 1, keeps the last value of Q, storing a bit.

NOR Latch:

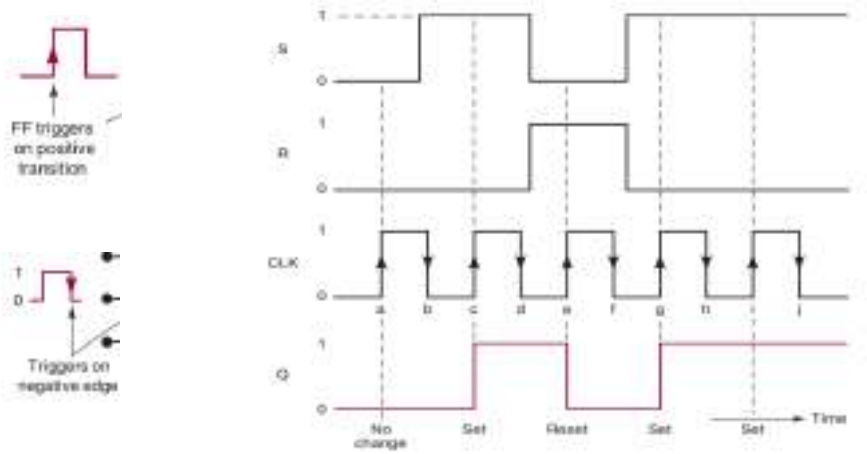
1. Setup: Made of two cross-connected NOR gates.
2. Inputs: S(Set) and R(Reset).
3. Operation:
    - Set: S = 0 and R = 1, makes output Q = 0.
    - Reset: S = 1, R = 0, makes output Q = 1.
    - Hold: S = R = 0, keeps the last value of Q, storing a bit.

5. Explain the synchronous/clocked S-R filp-flop with waveforms. (19-20) [4]

Answer:

A **synchronous/clocked S-R flip-flop** is a type of S-R (Set-Reset) flip-flop that includes a clock signal to control when the inputs S (Set) and R (Reset) can affect the output. The flip-flop changes its state only during a specific clock edge (usually the rising edge) if the clock signal is active, making it synchronous with the clock.

Waveform:



The waveforms in Figure show how a **clocked S-R flip-flop** operates:

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

1. **Initial State**: All inputs are 0, and Q is assumed to be 0.
2. **First Clock Pulse (Point a)**: With both S and R at 0, the flip-flop remains in its current state (Q = 0).
3. **Second Clock Pulse (Point c)**: S = 0 and R = 0, so the flip-flop sets Q to 1.
4. **Third Clock Pulse (Point e)**: S = 0 and R=1, so the flip-flop clears Q to 0.
5. **Fourth Clock Pulse (Point g)**: S=1 and R=0, setting Q to 1 again.
6. **Fifth Clock Pulse**: S=1 and R=0 again, so Q remains at 1 since it's already high.
7. **Invalid Condition**: When S=R=1, this creates an ambiguous state and should be avoided.

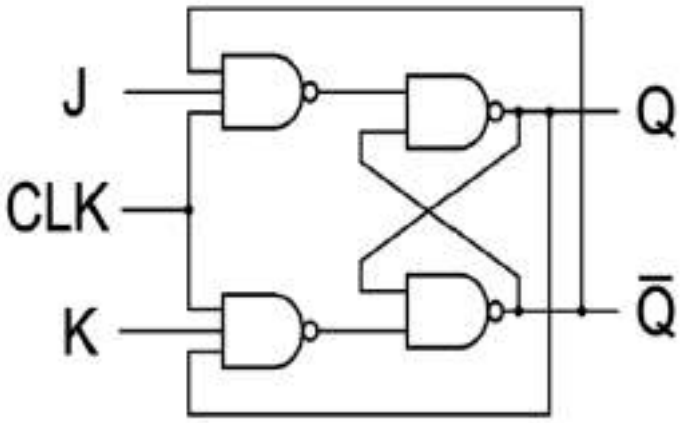
6. What is a flip flop? Explain the function of a JK flip flop with suitable circuit diagram. (18-19) [4]

Answer:

A flip-flop is a fundamental memory element in digital electronics that can store a single bit (0 or 1) of data.

J-K Flip-Flop

The **J-K flip-flop** is an improvement over the S-R flip-flop, as it eliminates the indeterminate (invalid) state that occurs when both Set (S) and Reset (R) inputs are 1. Instead, the J-K flip-flop includes feedback that allows it to toggle its output when both inputs are high.



Truth Table

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Function and Operation

The J-K flip-flop's operation depends on the values of J, K, and the clock:

When Clock is Low: The flip-flop maintains its previous state; inputs J and K have no effect.

When Clock is High (or on the rising edge, in edge-triggered designs):

- **J = 0, K = 0**: The flip-flop holds its previous state.
- **J = 0, K = 1**: The output Q is reset to 0,
- **J = 1, K = 0**: The output Q is set to 1.
- **J = 1, K = 1**: The output Q toggles (switches between 0 and 1) with each clock pulse.

Chapter 6 (Basic)

Binary Addition, Subtraction, Multiplication and divition:

addition:

$$\begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ +1 \\ \hline 11 \end{array}$$

carry (for 10)

carry (for 11)

Lets Example:

$$\begin{array}{r} 1100 \\ +\ 1110 \\ \hline 11010 \end{array}$$

carry

$$\begin{array}{r} 1101.011 \\ +\ 1100.110 \\ \hline 11010.001 \end{array}$$

carry

Subtraction:

$$\begin{array}{r} 1 \\ -1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ -0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ -0 \\ \hline 0 \end{array}$$

$0-1=1$ (borrow)

$0-10=0$ (borrow)

$1-10=1$ (borrow)

```
  11001          11000
- 10101        - 00111
--------       --------
  00100          10001
```

Multiplication:

```
   10101      স্বাভাবিক
 ×   101      গুণের পদ্ধতি
---------
   10101
  00000
+ 1010100
---------
  1101001
```

Division:

```
   11 ) 1001 ( 11      স্বাভাবিক
        - 11
        ----
        0011
          11
        ----
           0
```

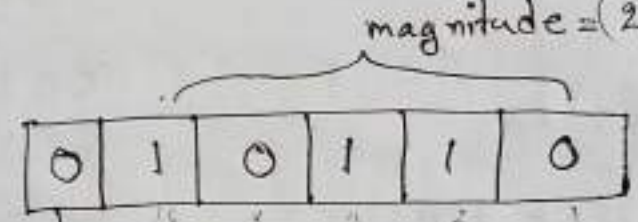
```
  100 ) 11101 ( 111
        - 100
        -----
        0110
          100          = 111
        -----
        0101           Reminder = 001
          100
        -----
        0010
```

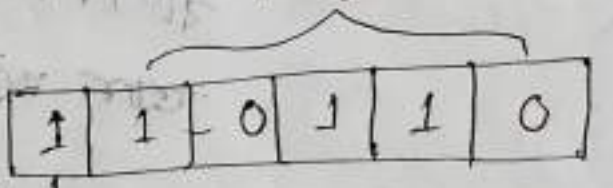
2's complement: (addition and subtraction)

Representing signed number:

In general the common convention is that a 0 in the sign bit, represents a positive number and a 1 in the sign bit repre a negative number.

Example:





magnitude = absolute value

1's complement:

The 1's complement of binary number means changing the bit 0 to 1 and 1 to 0.

```
1 0 1 0 1 1  ⟶ orginal number
↓ ↓ ↓ ↓ ↓ ↓
0 1 0 1 0 0  ⟶ 1's complement.
```

2's complement:

The 2's complement of a binary number is formed by taking 1's complement of the number, then a adding 1 to the least-significant-bit position.

Example:

$45_{10} = 101101_2$

101101 → original
010010 → 1's complement
+ 1 → adding 1 with 1's complement

010011 → 2's complement of 101101_2 or 45_{10}

→ -45_{10}

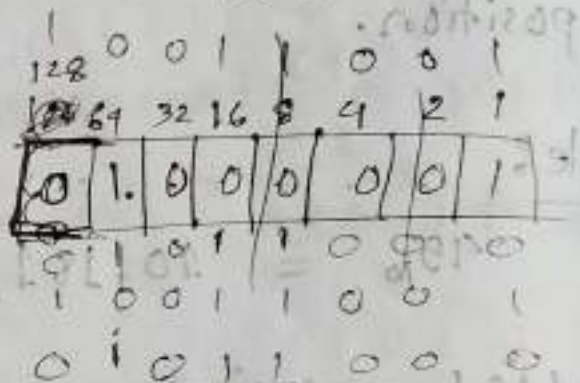
* 2's complement এর মাধ্যমে negative number কে positive এবং positive কে negative করা যায়। এক্ষেত্রে sign bit এর sign change হয়।

* Die Decimal number (127 to -128) এর জন্য 8-bit ব্যবহার করা হয়। এর বেশি হলে 16 bit.

4| Using 2's complement system perform the following operations: (19-20) [2]

(i) $-65-88$ (ii) $34+55$

128+16+8+1



(i) $-65-88$

$\Rightarrow (-65)+(-88)$

Now,

$65 \rightarrow 01000001$

$10111110 \rightarrow$ 1's complement

$+1$

$(-65) \rightarrow 10111111 \rightarrow$ 2's complement

sign bit (-)

$88 \rightarrow 01011000$

$10100111 \rightarrow$ 1's complement

$+1$

$(-88) \rightarrow 10101000 \rightarrow$ 2's complement

sign bit (-)

① -65-88

Given that,

= -65-88

or. (-65) + (-88)

65 → 00000000 01000001

11111111 10111110 → 1's complement

+ 1

(-65) → 11111111 10111111 → 2's complement

88 → 00000000 01011000

11111111 10100111 → 1's complement

+ 1

(-88) → 11111111 10101000 → 2's complement

-65 → 11111111 10111111

-88 → 11111111 10101000

(+)

-153 → [1][1]11111 01100111

carry bit

sign bit

Now the result is in overflow condition. To resolved this we need to find 2's complement

of this result again.

So,

```
1 111 1111 10011001   → overflow condition
0000 0000 1 0011000   → 1's complement
                  + 1
-------------------
0000 0000 0010011001  → 2's complement
```

~~Or~~ 10011001 ~~is in decimal~~

or, 10011001

sign bit(-)

Now convent this result to decimal which is -153 as we expected.

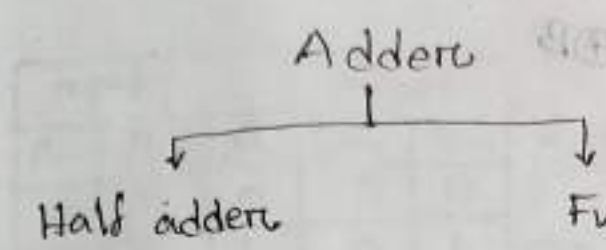
(ii) 34+55

```
+34 → 00100010
+55 → 00110111
-------------
+89 → 01011001
```

sign bit(+)

∴ The result of 34+55 in 2's complement system +89 or 01011001.

Adder



~~Adder~~

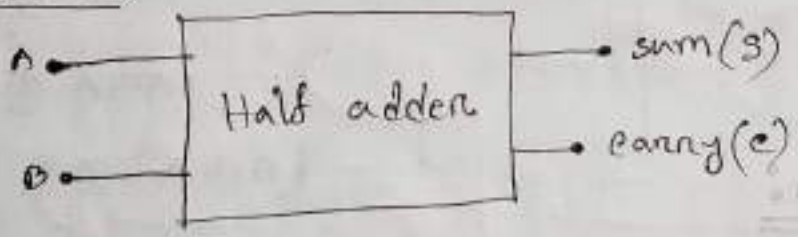
Decoder, Encoder, half adder, full adder

↳ related Q. exam এ আসতে পারে 2 2 5 step এ

বর্ণনা দিতে হবে, → (i) Defination
(ii) Block diagram
(iii) Truth Table
(iv) Boolean expression
(v) Logic gate.

(i) Half adder:

(ii) Block diagram:



(iii) Truth Table:

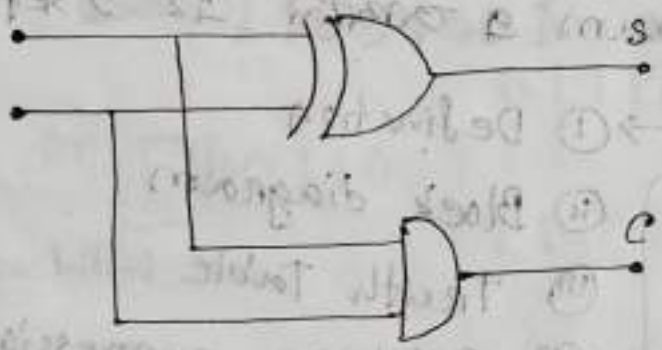
A	B	carry	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

④ Boolean expression:

$$Sum = \bar{A}B + A\bar{B} = A \oplus B$$

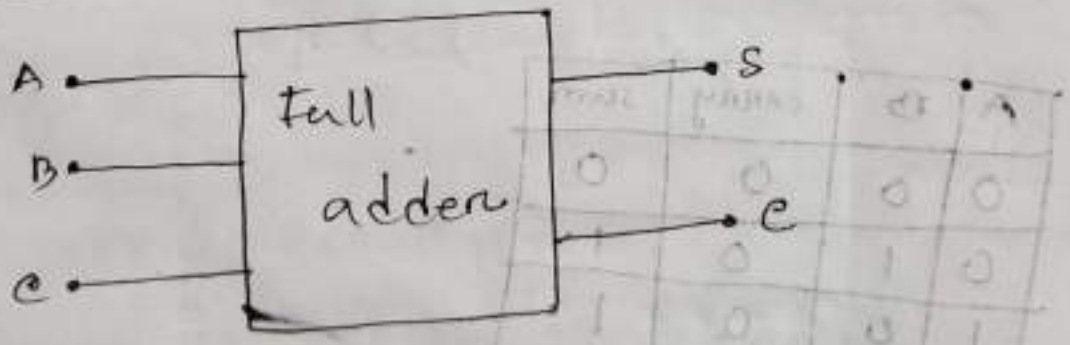
$$Carry = AB$$

⑤ Logic gate:



Full adder:

Block diagram:



Truth table:

input			Output	
A	B	C	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

** input এ যখন 2টি 1 থাকে তখন sum = 0 and carry = 1 হবে।
যখন তিনটা 1 থাকে তখন sum = 1, carry = 1 হবে।

Boolean expression:

$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$$

$$= \bar{C}(\bar{A}B + A\bar{B}) + C(AB + \bar{A}\bar{B})$$

$$= \bar{C}(A \oplus B) + C(\overline{A \oplus B})$$

$$= C \oplus (A \oplus B)$$

$\therefore$ Sum = $C \oplus (A \oplus B)$

যদি $x \oplus y = \bar{x}y + x\bar{y}$

$x = C$

$y = A \oplus B$

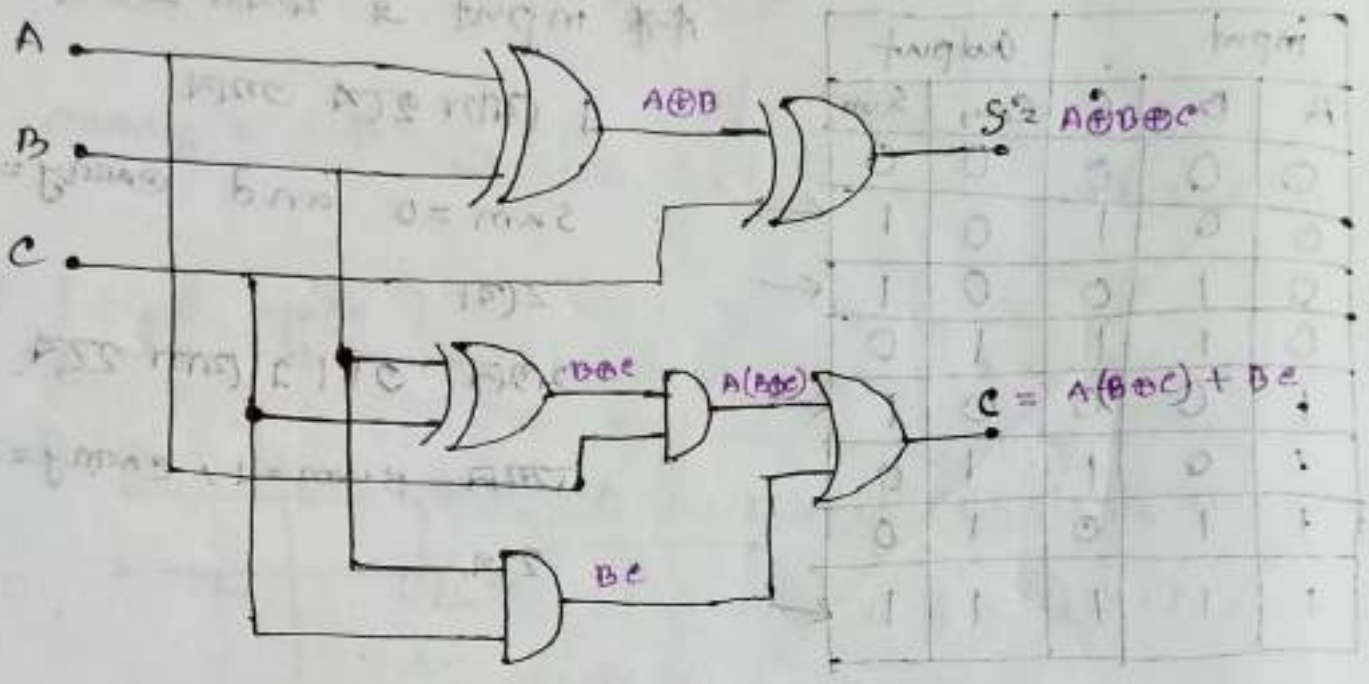
$$\text{Carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$= A\bar{B}C + AB\bar{C} + \bar{A}BC + ABC$$

$$= A(\bar{B}C + B\bar{C}) + BC(\bar{A} + A)$$

$$= A(B \oplus C) + BC$$

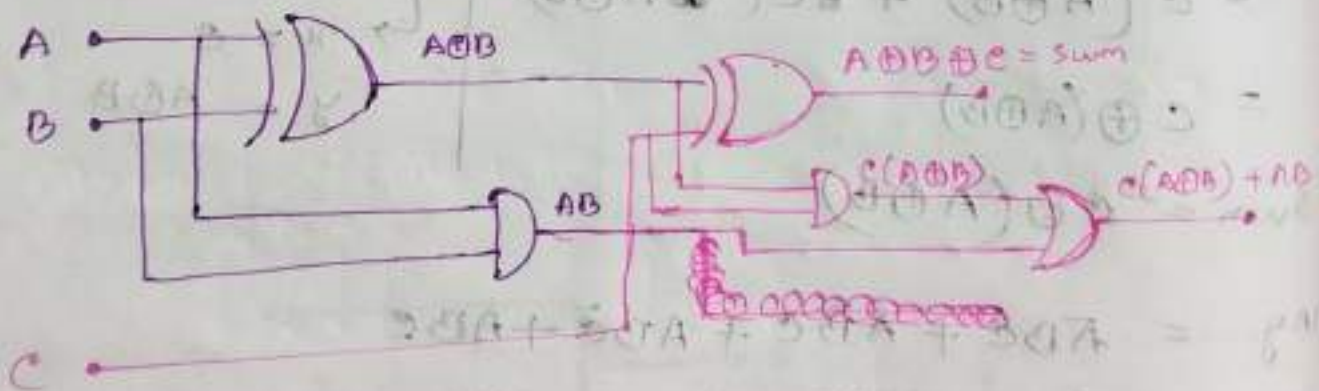
Logic gate:



Logic gate of full adder.

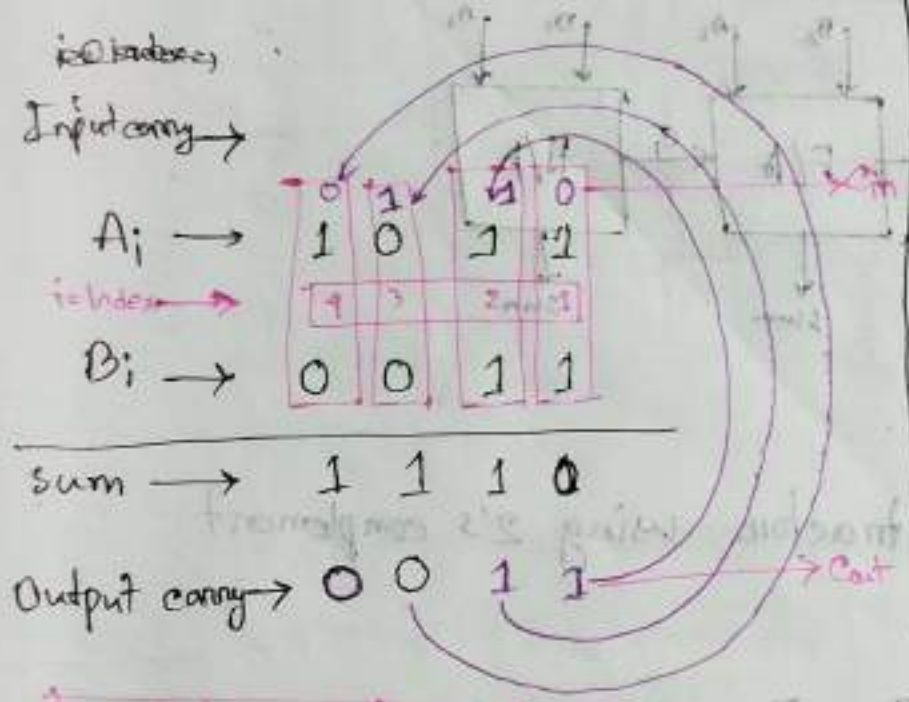
Half adder to full adder:

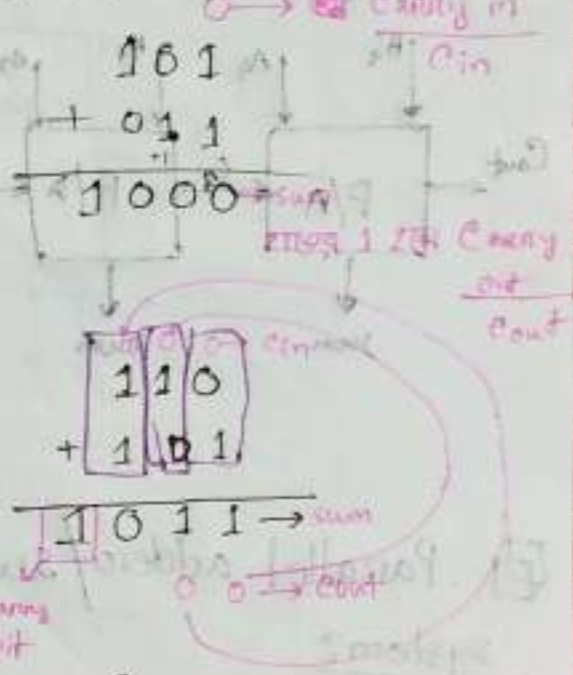
Full adder = 2 × Half adder + 1 × OR gate



Parallel adder:

Basic

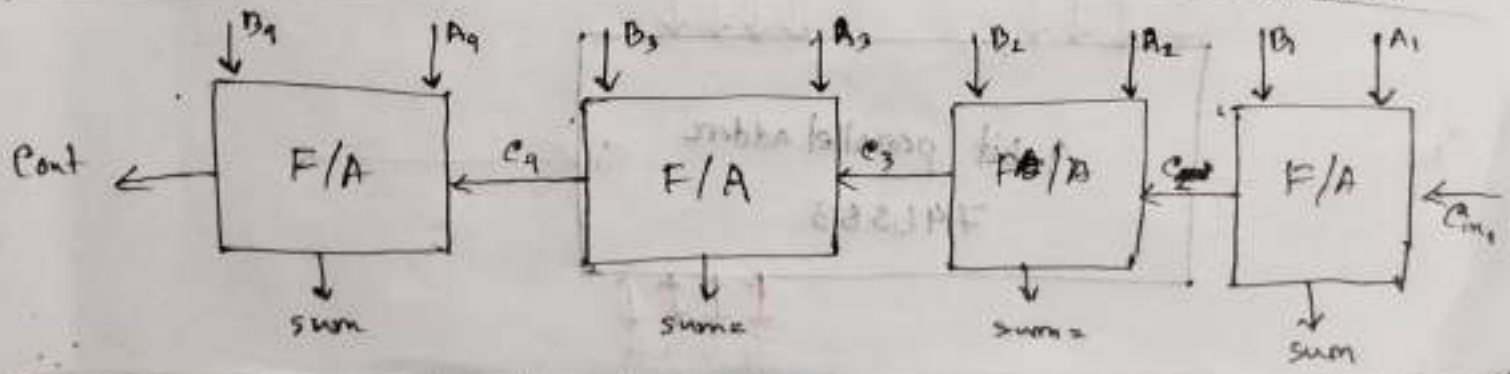




Parallel adder mechanism

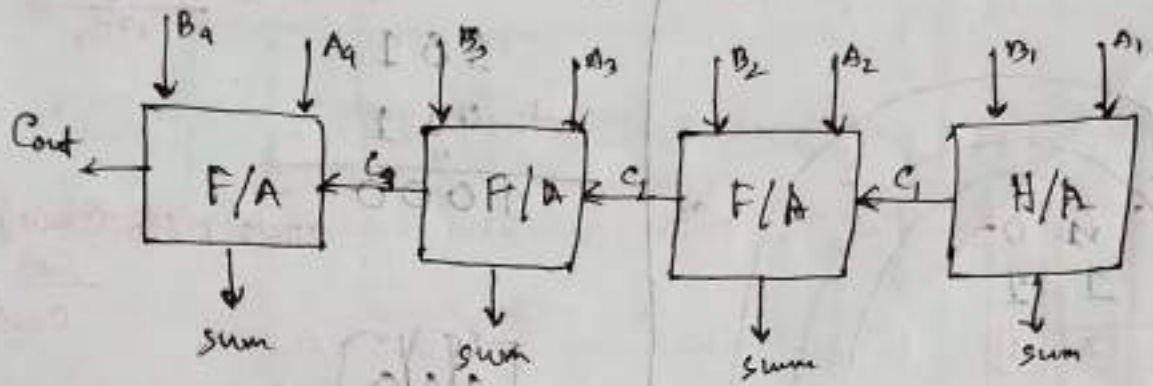
* n-bit ~~para~~ binary parallel adder এর জন্য n-bit full adder প্রয়োজন।

Block diagram: (4 bit parallel adder)



যিহেতু 2 bit এর যোগ
সহজ কিন্তু Finally
তা ৩ bit এ হয়ে।
এবং প্রয়োজনটা Parallel.
তাই এরে Parallel
~~adder~~ বলে।
addition.

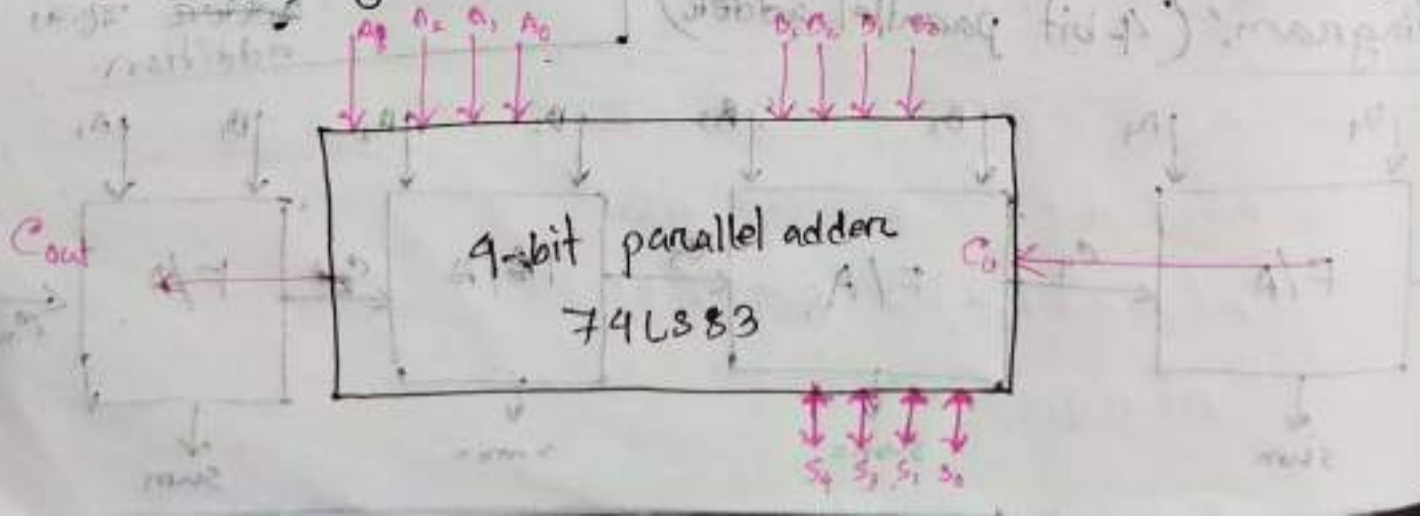
* যদি n-bit binary parallel adder H/A and F/A এর সাহায্যে গঠন করতে বলে, তাহলে সবুই প্রথম adder টি H/A দিয়ে হবে কারন প্রথমে $C_{in} = 0$ থাকে, যা দিলেও হয় না দিলেও হয়।



Parallel adder/subtractor using 2's complement system:

Understand the Ic 74LS83. It is a 4-bit parallel adder,

4-bit parallel adder এর internal mechanism → Already দেখেছি। IC এর ক্ষেত্রে একটু ভিন্ন ভাবে দেখা

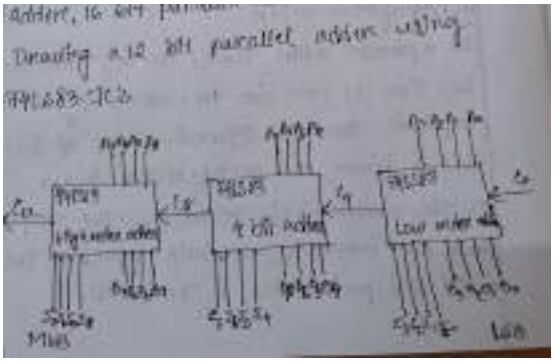


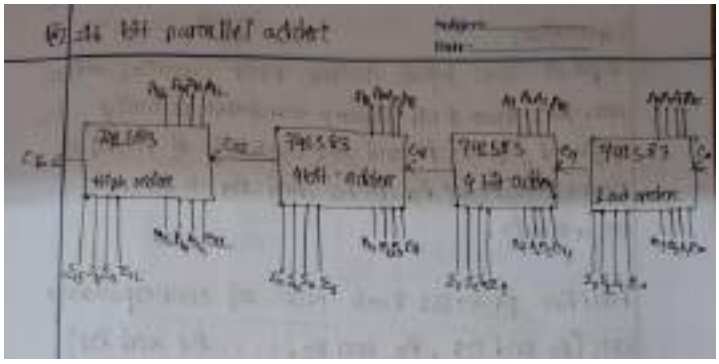
7. What is the limitation of a parallel adder? Using 74LS83 ICs draw a 16/12 bit parallel adder. (20-21, 18-19) [3]

Answer:

The main limitations of a parallel adder are:

1. **Carry Propagation Delay:** Each bit addition depends on the carry from the previous bit, which causes a delay as it moves through each stage of the adder. This delay increases with the number of bits, slowing down the adder.
2. **Complexity**: Larger adders need more gates and connections, increasing cost and size.
3. **Limited Scalability**: Due to the increased delay and complexity with each additional bit, parallel adders are not efficient for high-bit operations, such as in 32-bit or 64-bit processors.





8. Explain how 2's compliment system can facilitate arithmetic operation in digital computing. (20-21, 18-19) [3]

Answer:

The **2's complement system** is widely used in digital computing to represent negative numbers and simplify arithmetic operations, particularly subtraction. Here's how it facilitates these operations:

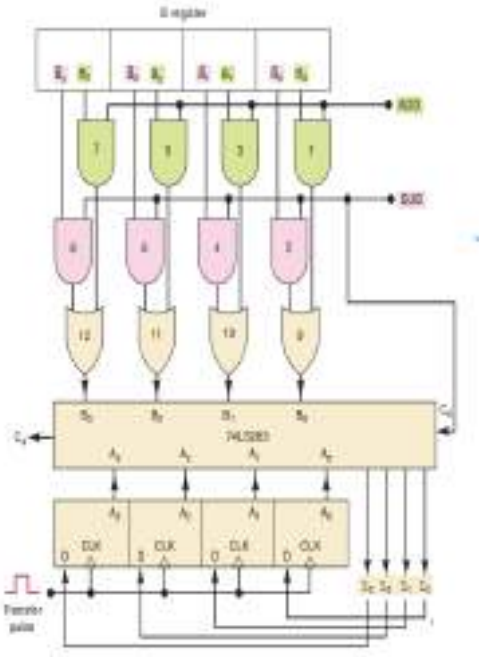
1. **Unified Addition and Subtraction**:
   - In 2's complement, subtraction can be performed as addition, simplifying the circuit design.
   - To subtract a number, simply add its 2's complement (invert all bits and add 1).
   - This means a single adder circuit can handle both addition and subtraction, making it efficient for hardware design.
2. **Single Representation of Zero**:
   - Unlike other signed number systems, 2's complement has only one representation for zero, reducing ambiguity in calculations.
3. **No Special Circuit Needed for Sign**:
   - Positive and negative numbers can be added directly without separate circuits to handle the sign, simplifying the hardware.
   - Overflow conditions are handled automatically when the sum exceeds the responsible range.
4. **Efficient Use of Bits**:
   - In a fixed number of bits, the 2's complement system maximizes the range of representable values, allowing both positive and negative values in a simple format.

9. Draw a parallel adder/subtractor using 2's complement system and explain its operation. (20-21) [6]

Answer:

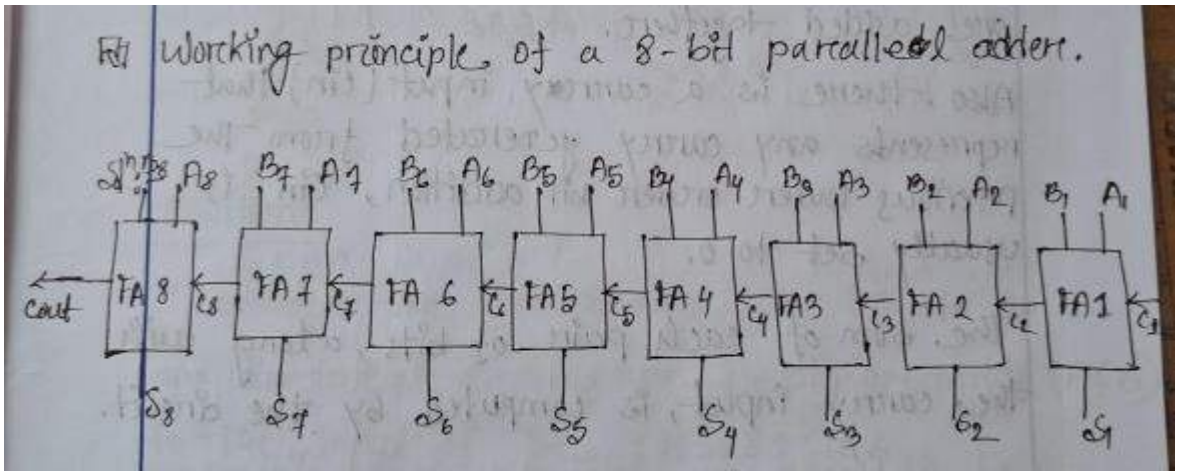
Here's a simplified description of the operation of the parallel adder/subtractor circuit:

1. **Addition Mode (ADD = 1, SUB = 0)**:
   - **Control Signal**: SUB = 0 disables certain AND gates (2, 4, 6, 8) to hold their outputs at 0, while ADD = 1 enables AND gates (1, 3, 5, 7) to pass the B_0 to B_3 values.
   - **Operation**: The B register values B_0 to B_3 pass through OR gates to the adder, where they are added to A_0 to A_3.
   - **Carry-In**: $C_0 = 0$, so no additional carry is added.
   - **Output**: The sum appears at outputs S_0 and S_3.
2. **Subtraction Mode (ADD = 0, SUB = 1)**:
   - **Control Signal**: ADD = 0 disables certain AND gates (1, 3, 5, 7), while SUB = 1 enables the other AND gates (2, 4, 6, 8) to pass B_0 and B_3 values.
   - **Operation**: The B register values B_0 to B_3 pass through OR gates to the adder, but with $C_0 = 1$, effectively converting B to its 2's complement.
   - **Output**: The adder performs A-B and the difference appears at outputs S_0 and S_3.



10. Design and explain the working principle of an 8-bit parallel adder. (19-20) [6]

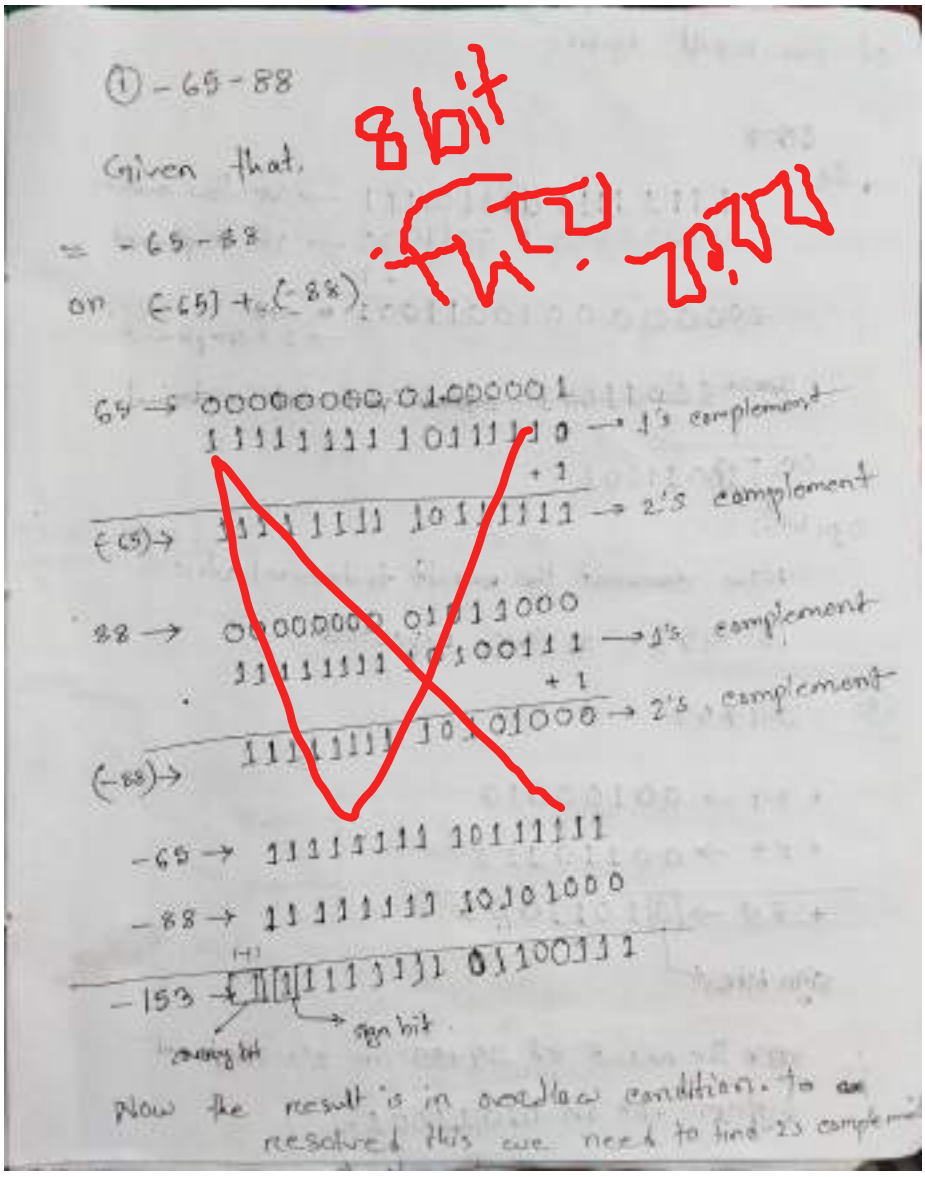
Answer:

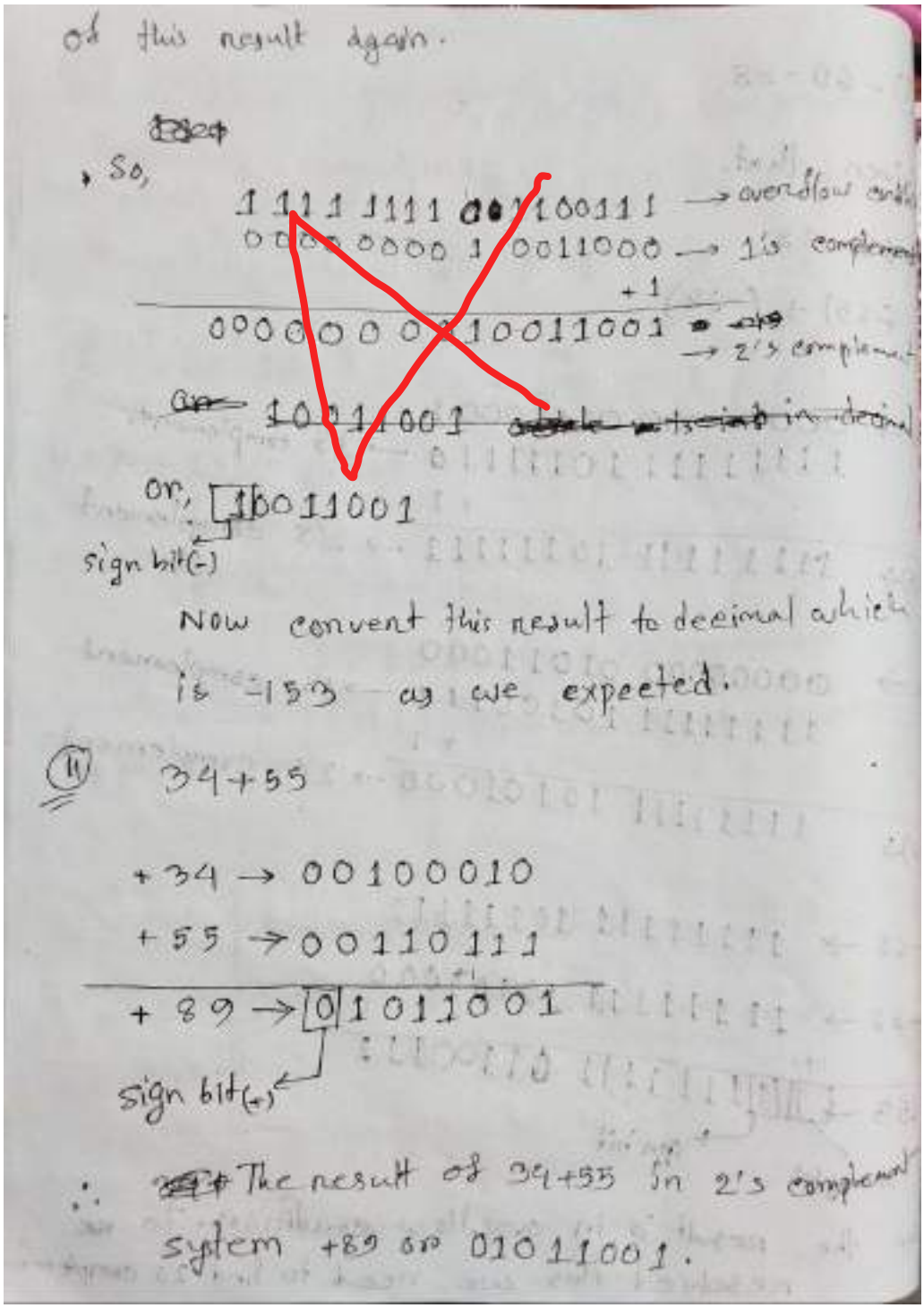


An 8-bit parallel adder is a digital circuit that performs the addition of two 8-bit binary numbers in parallel. It adds each pair of corresponding bits from the two operands simultaneously, producing an 8-bit sum output.

Inputs:
The inputs to the 8-bit parallel adder are two 8-bit binary numbers, typically referred to as A and B. Each bit of A and B is represented as A_0 to A_7 and B_0 to B_7, respectively.
Addition Process:
Each pair of corresponding bits (A_i and B_i) is added together. Additionally, a carry-in (C_{i-1}) from the previous bit addition is included in the calculation. The carry-in (C_{i-1}) is usually set to 0 for the least significant bit (LSB). The sum of each pair of bits, along with the carry input, is computed by the adder circuit.
Outputs:
The outputs of the 8-bit parallel adder are the 8-bit sum (S_0 to S_7), representing the result of the binary addition.

11. Using 2's complement system performs the following operations:(19-20) [2]
i) -65-88
ii) 34+55





of this result again.

So,

1 1 1 1 1111 00100111 → overflow bit

0000 0000 1 0011000 → 1's complement

+ 1

0000 0000 1 0011001 → 2's complement

or, 10011001

or, 1 0011001

sign bit(-)

Now convent this result to decimal which is -153 as we expected.

(iv) 34+55

+34 → 00100010

+55 → 00110111

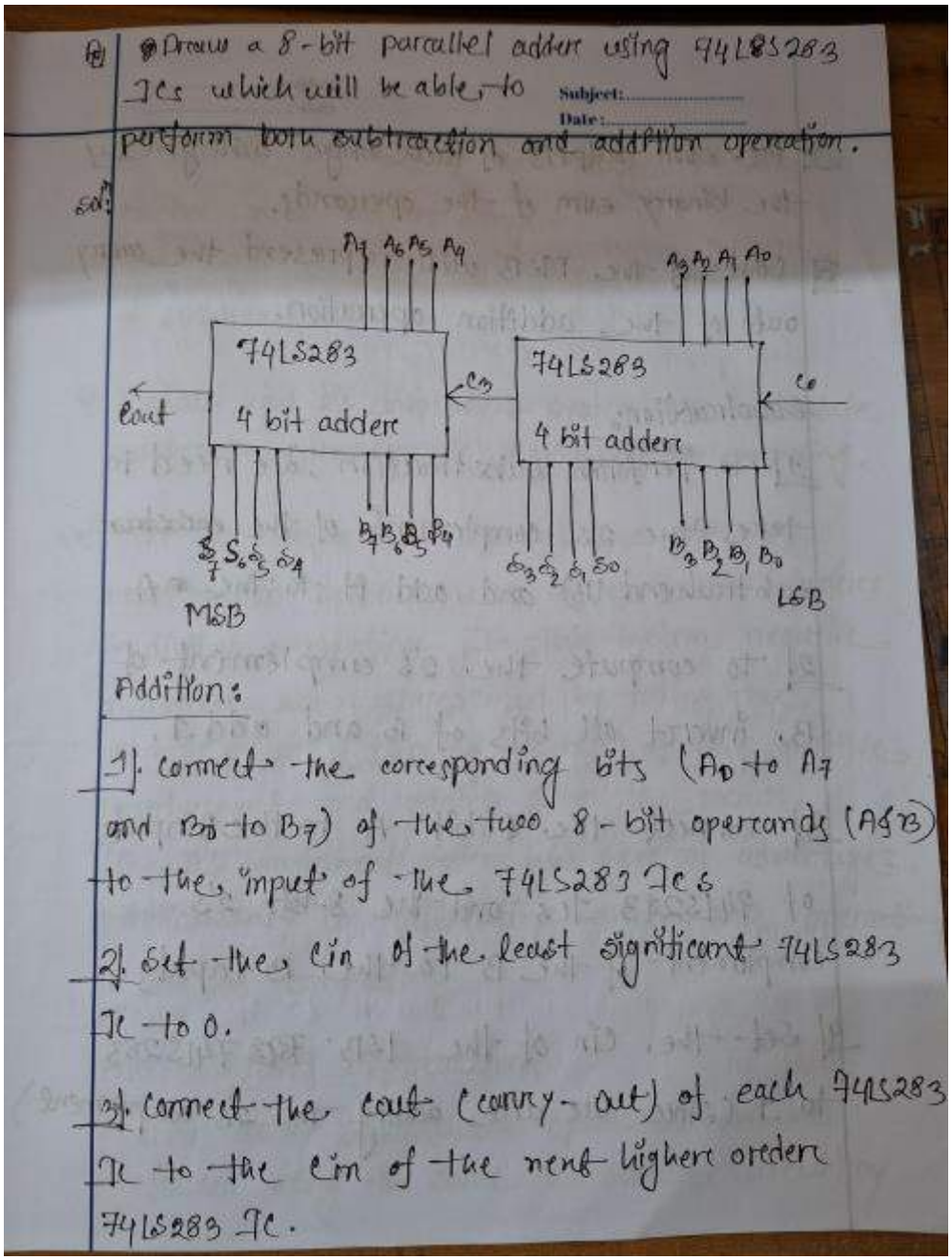
+89 → 0 1011001

sign bit(+)

∴ The result of 34+55 in 2's complement system +89 or 01011001.

12. Draw an 8-bit parallel adder using 74LS83 ICs, which will be able to perform both subtraction and addition operation. (19-20) [4]

Answer:



Subject:
Date:

4] The sum outputs of each stage will give us the binary sum of the operands.

5] Cout of the MSB will represent the carry out of the addition operation.

Subtraction:

1] To perform subtraction, we need to take the 2's complement of the subtrahend (B) and add it to the A

2] to compute the 2's complement of B, invert all bits of B and add 1.

3] connect the 8-bit A to the A inputs of 74LS283 ICs and the 8-bit 2's complement of the B to the B inputs.

4] Set the Cin of the LSB 74LS283 to 1 (since we are adding the 2's complement)

Subject:
Date:

5] Connect the Cout of each stage to the Cin of the next higher stage.

The sum outputs will give us the result of subtraction.

Chapter 7

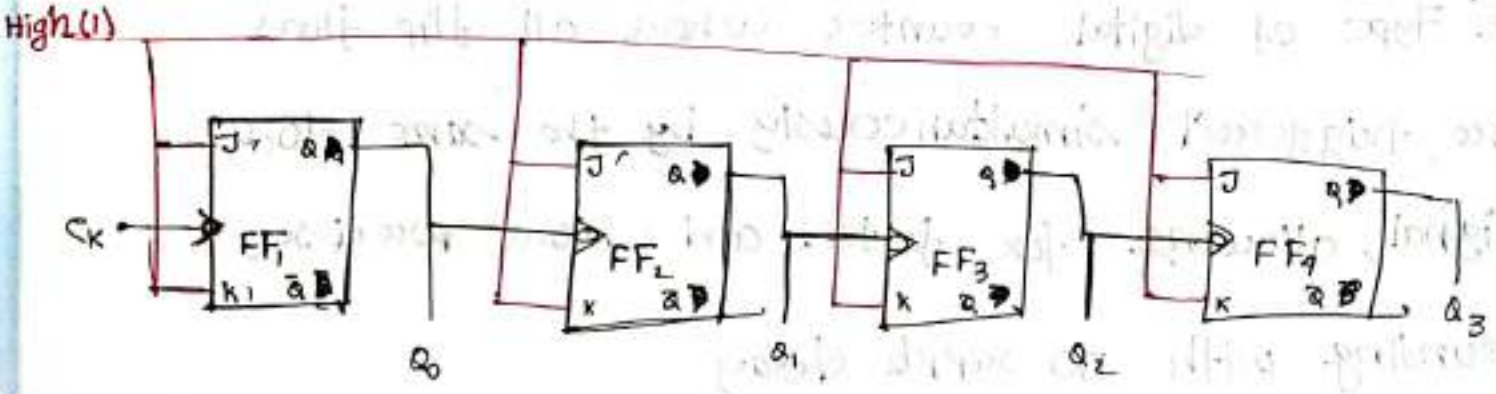
counter:-

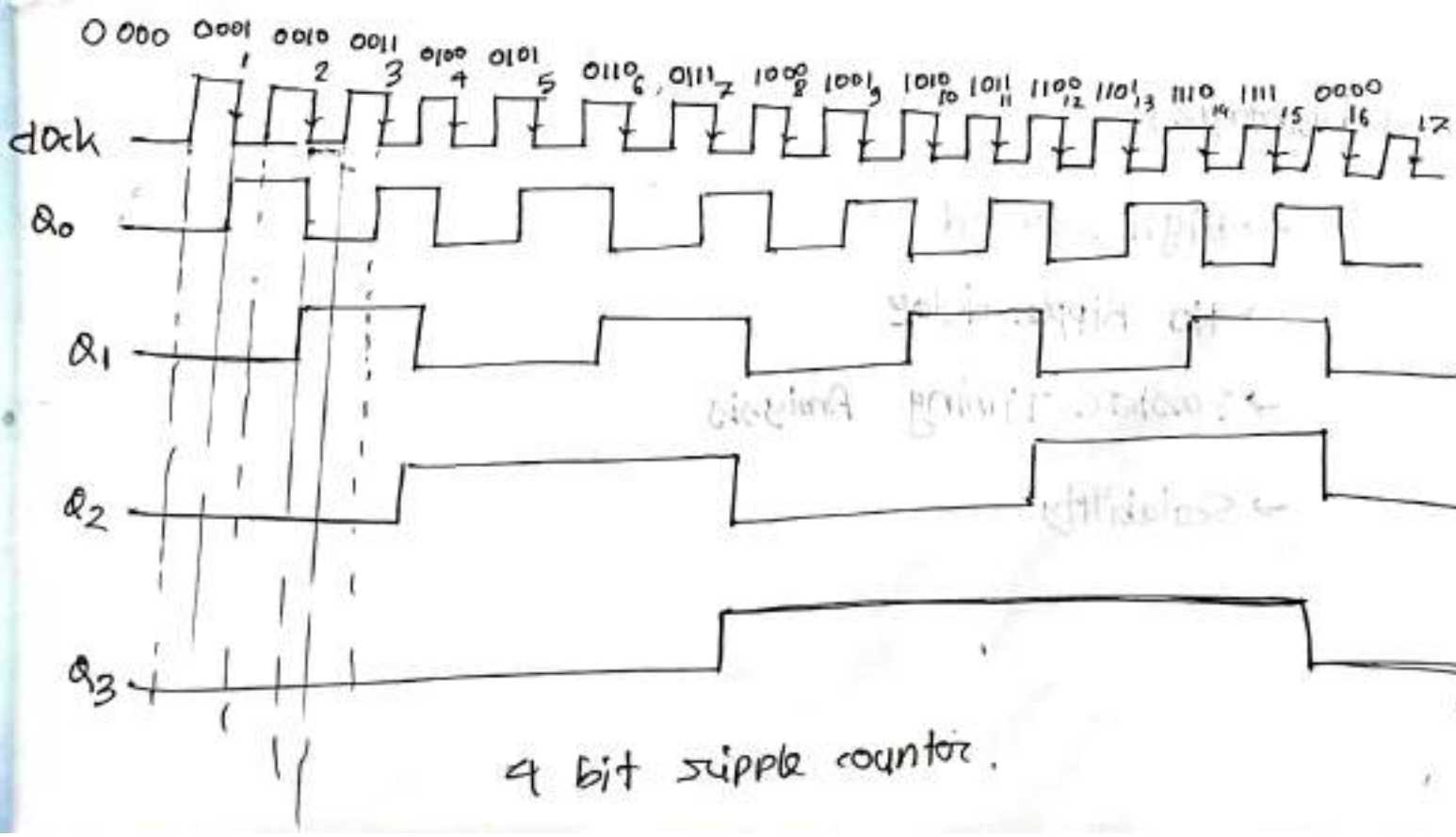
A counter is a device which stores (displays) the number of timer particular event or process has occurred, often in relationship to a clock signal.

- It is design by flip-flop (J-K)
- The main properties of counter are timing, sequencing and counting.
- counter works on two mood
  - → up counter
  - → down counter
- counter two categories
  - → Asynchronous counter (Ripple counter)
  - → synchronous counter

⊛ Asynchronous (Ripple) counter

In Asynchronous counter we don't use universal clock. only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops. We can understand it by following diagram-





4 bit ripple counter.

Definition :- An asynchronous counter is a type of digital counter where the each flip flop is triggered by the output of the previous one. It is also ripple counter.

Synchronous (parallel) counter : A synchronous counter is type of digital counter where all flip flops are triggered simultaneously by the same clock signal, allowing for faster and more precise counting with no ripple delay

advantages

- → High speed
- → No Ripple delay
- → Easier Timing Analysis
- → Scalability

disadvantages

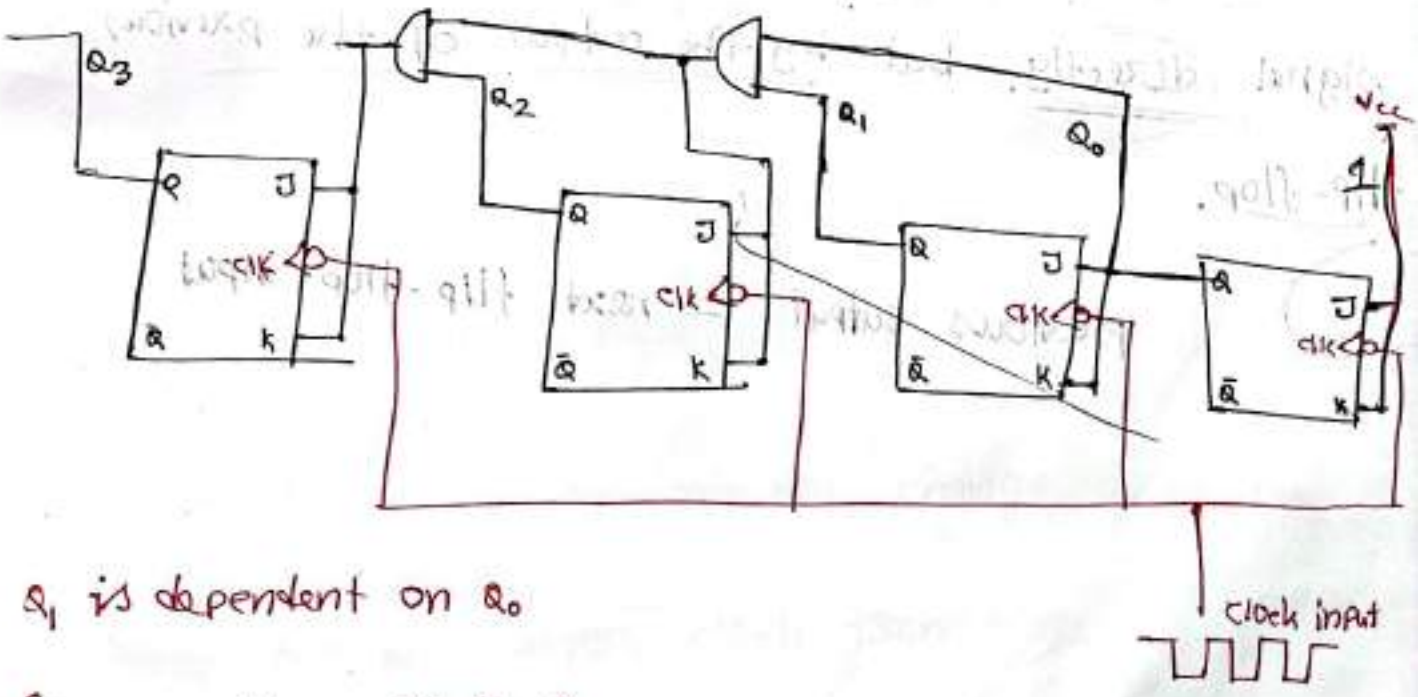
→ complexity

→ Higer cost

→ Larger circuit size

(ii) Functionality

Synchronous counter has one global clock which drives each flip-flop so ouput changes in parallel.



Q_1 is dependent on Q_0

Q_2 " " on Q_0, Q_1

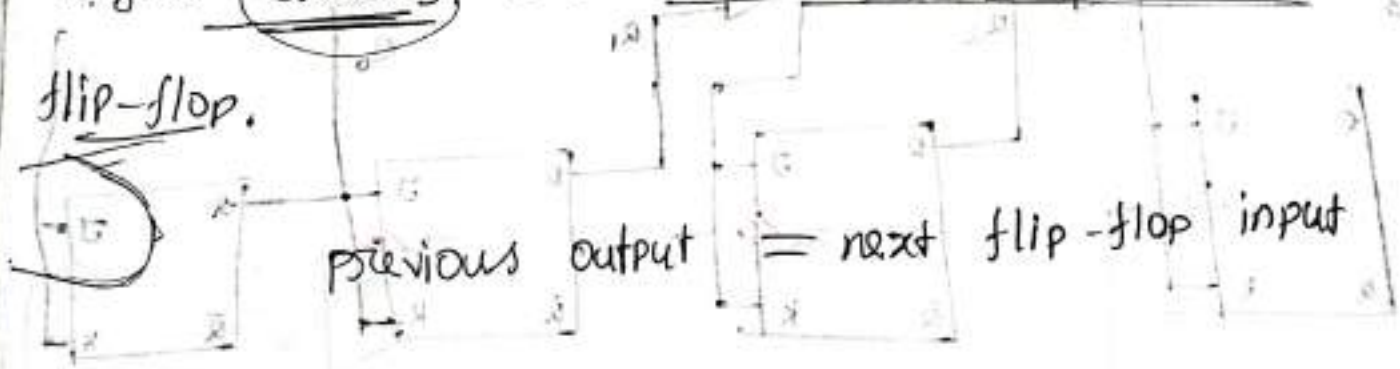
Q_3 " " on Q_0, Q_1 and Q_2

Timing diagram is same as asynchronous

⊛ Propagation Delay in Ripple counters

It refers to the time it takes for a change in the input (clock pulse) to be reflected in the output of the counter.

This delay occurs because, in a ripple counter, each flip flop is triggered not by the clock signal directly but by the output of the previous flip-flop.



each flip-flop triggered by the output of the preceding flip-flop.

→ Because of inherent propagation delay time (t_d) of each flip-flop

this means the 2nd FF_2 will not respond untill a time t_{pd} after the FF_1 active

the FF_3 will not respond untill time equl $2t_{pd}$ after the clock transition.

Overall The propagation delays of the FF accumulate so that N th Flip Flop can not change until time equal $N \times t_{pd}$ after clock transition.

max. frequency

$$f_{max} = \frac{1}{N\, t_{pd}}$$

t_{pd} = propagation delay

N ← number of FF

87002

MOD number

It represents the total count of distinct output values the counter can produce.

⇒ A counter with mod number of N will count 0 to N-1 before resetting back to 0.

→ example 3 bit binary counter has MOD number is $2^3 = 8$, it counts through 8 states 0 to 7,

Synchronous

(i) The operation is faster

(ii) It is also known as parallel counter

(iv) less error

(vi) Desigen is complex

Asychronous

(i) The operation is lower

(ii) It is also known as ripple counter

(iv) more error

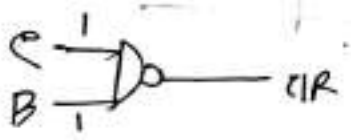
(vi) Desigen is simple

(*) counter with mod numbers $< 2^n$

Stat mod 6 counter

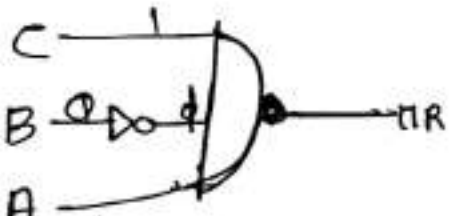
6 9 21

mod 6



count	C	B	A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

mod 5

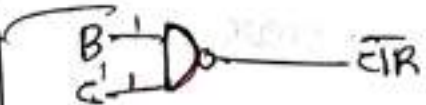


	D	C	B	A	
8	1	0	0	0	
9	1	0	0	1	

$2^4 = 16$

count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16				

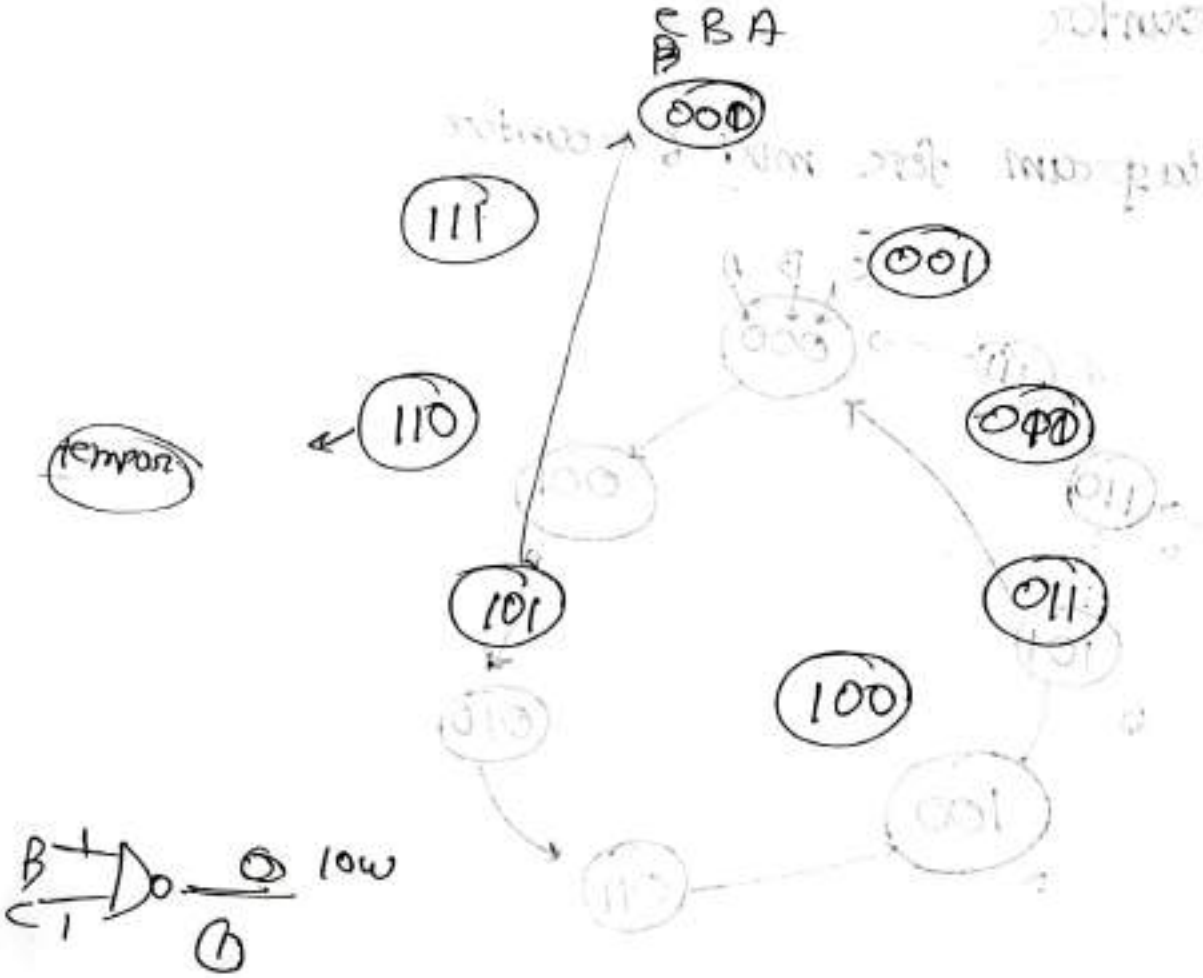
mod 6 → Temp state

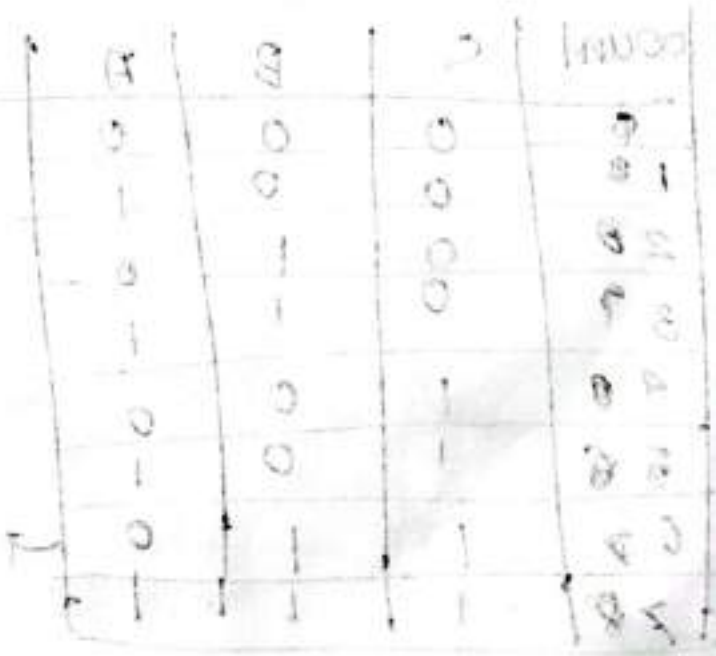


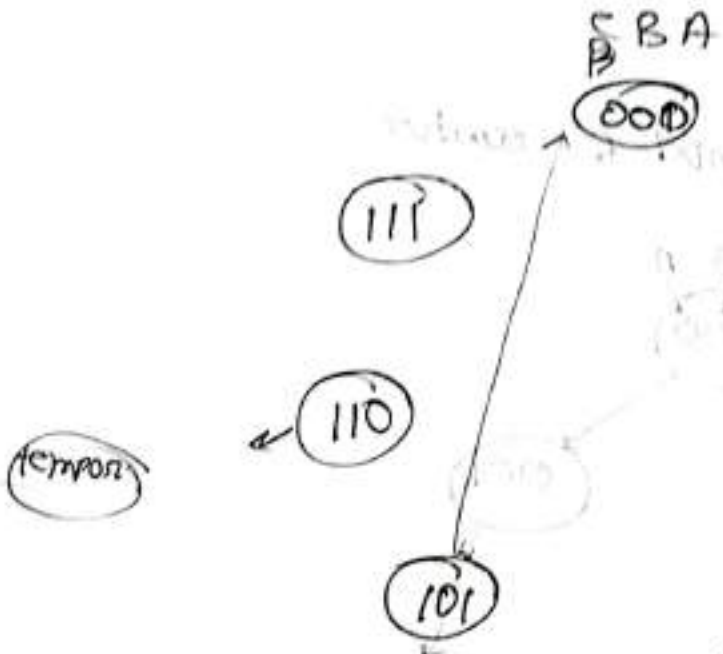
mod → 10 Temp state

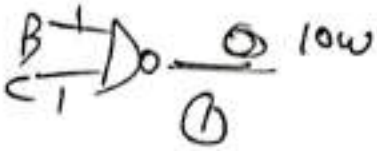
mod (13)

mod 14 → Temp state



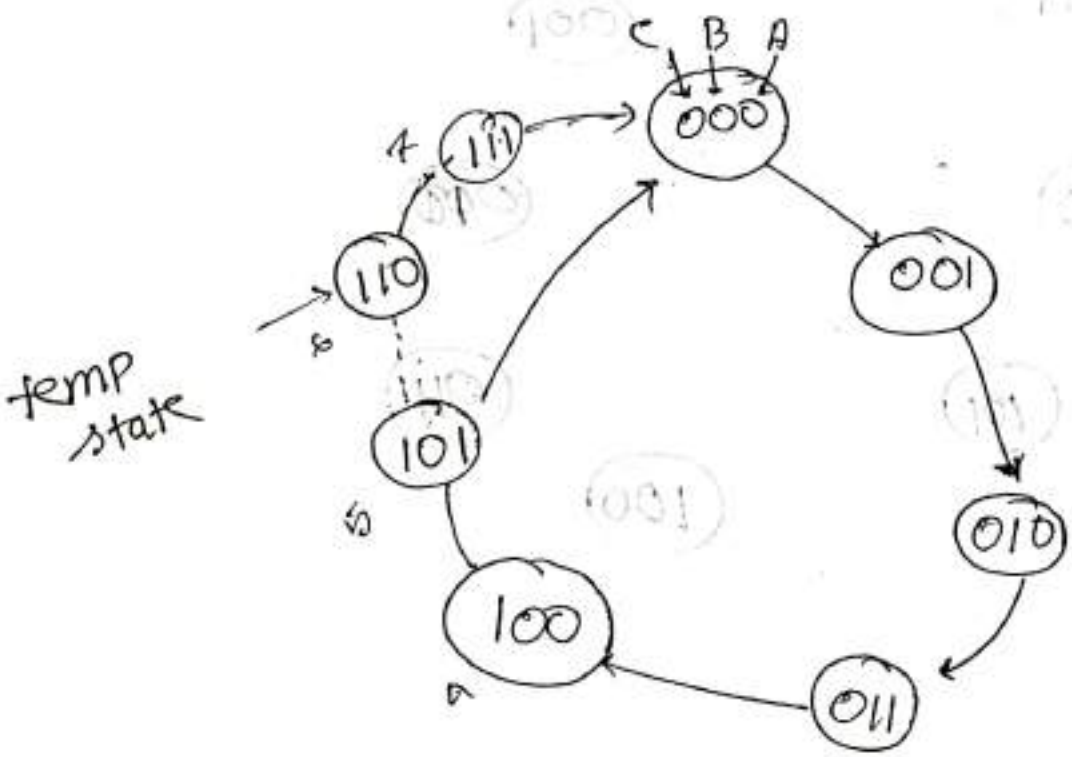






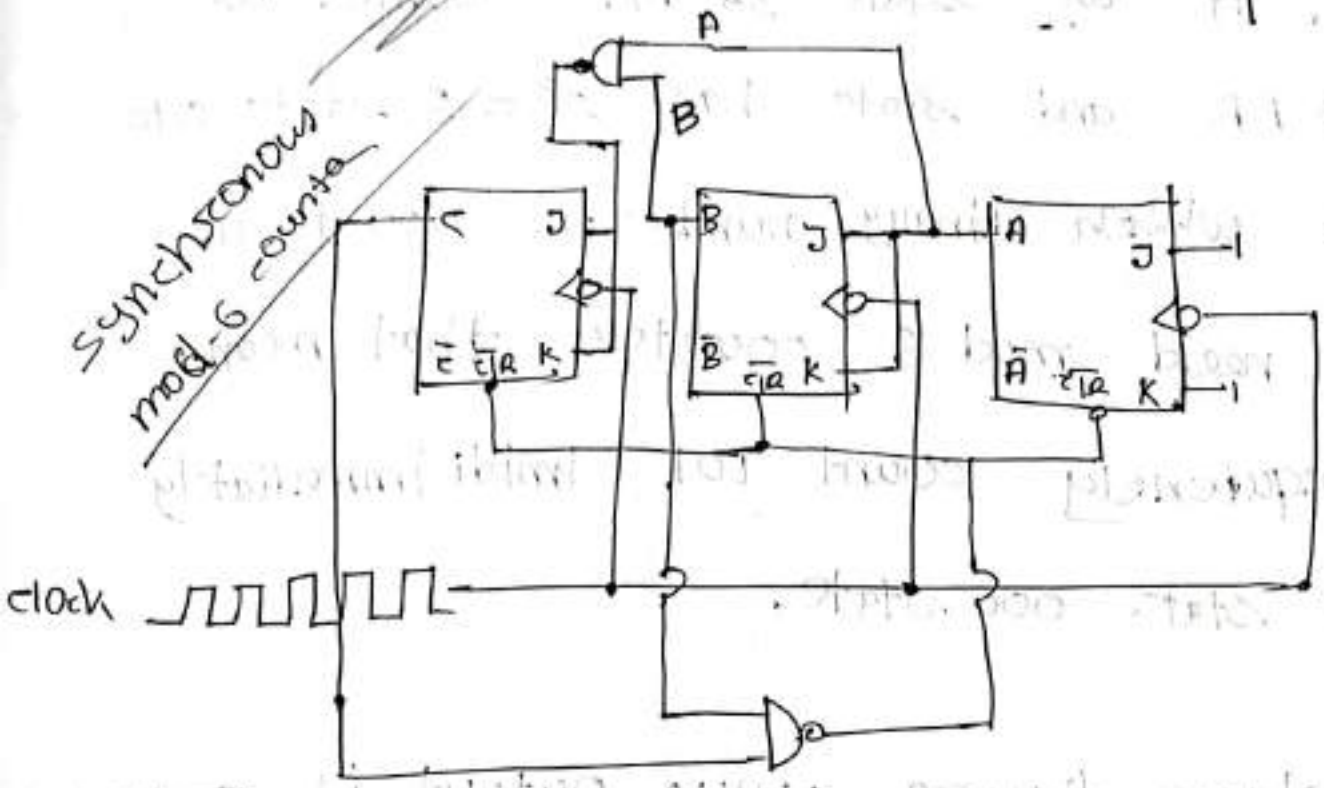
MOD 6 counter

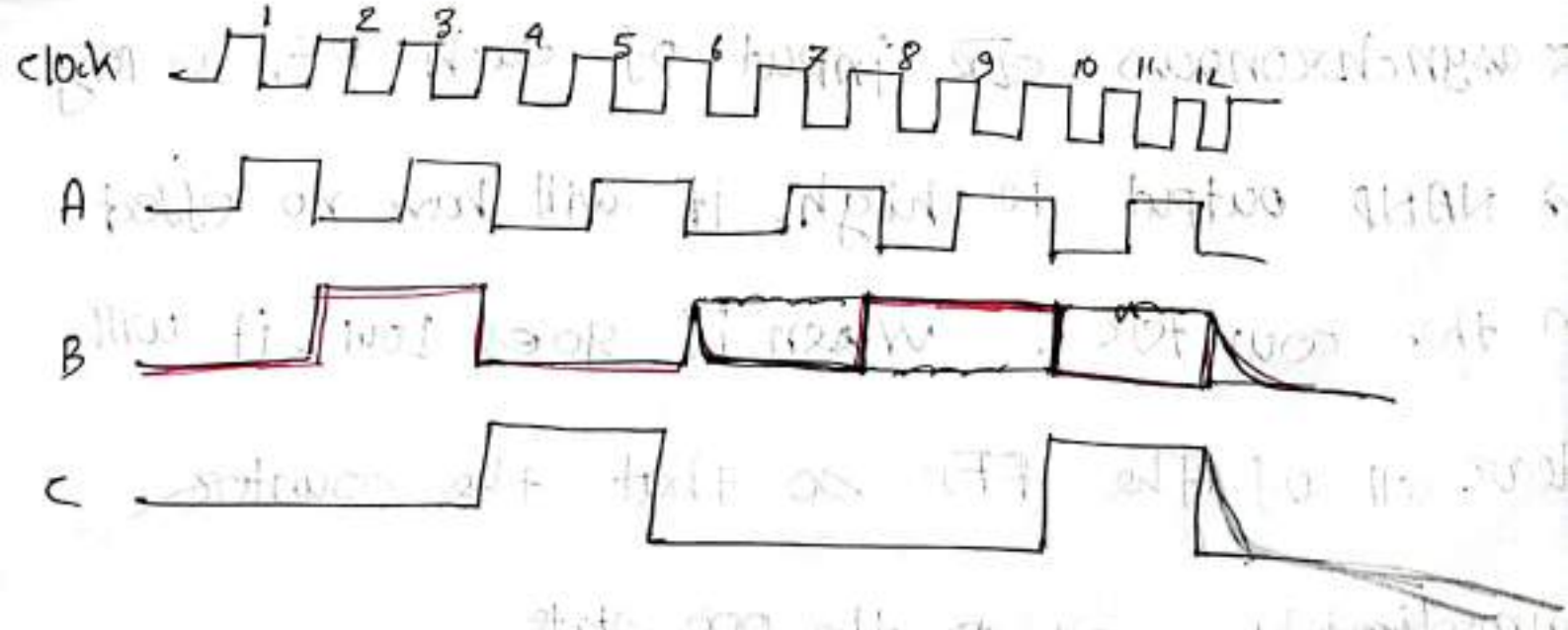
Stat diagram for mod 6 counter



count	C	B	A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Synchronous mod-6 counter





The basic counter has 2^N ~~whe~~ state whe N is FF. if we ~~state~~ build 3 bit counter we need 3 FFs and state has $2^3 = 8$ state 0 to 7 decimal which binary number is 000 to 111. But we need mod 6 counter that means after sequenely count 101. ~~imidi~~ immediakly goes to stat 000 state.

In the above figure NAND output is connecte ~~as~~ asynchronous $\overline{CLR}$ input of each FF. As long as NAND output is high, it will have no effect on the counter. When it goes low, it will clear all of the FFs so that the counter immediately goes to the 000 state.

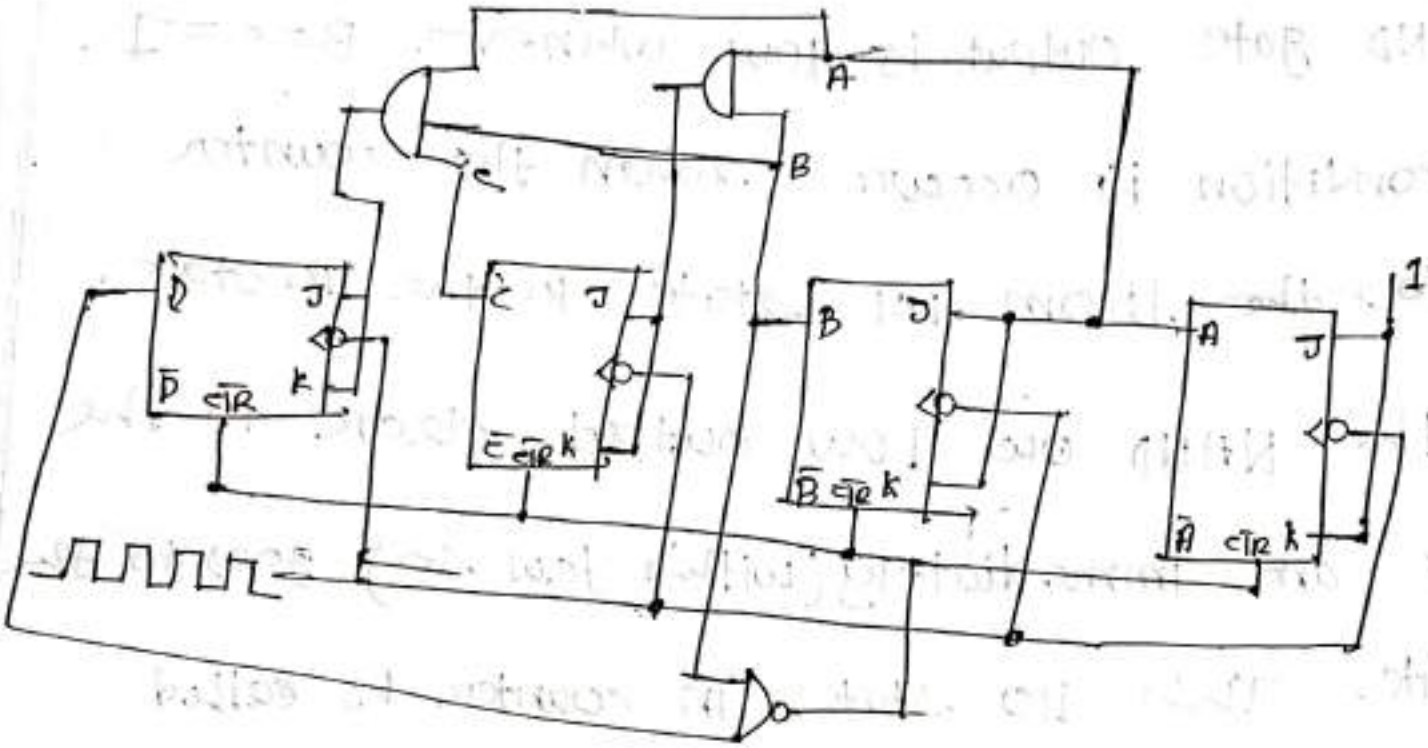
The input of NAND gate is outputs of FF B and C.

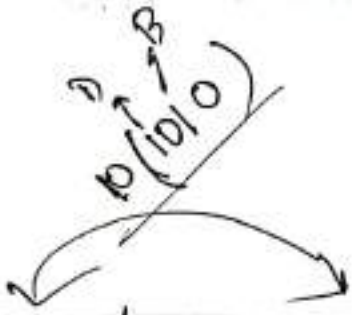
so NAND gate output is low whenever B=C=1.

This condition is occure when the counter goes ~~to the~~ from 101 state to the 110 state.

The the NAND ~~out~~ LOW output clear to the counter and immediately (within few sec) goes to the 000 state. This 110 state in counter is called temporary state it's needed to clear counter.

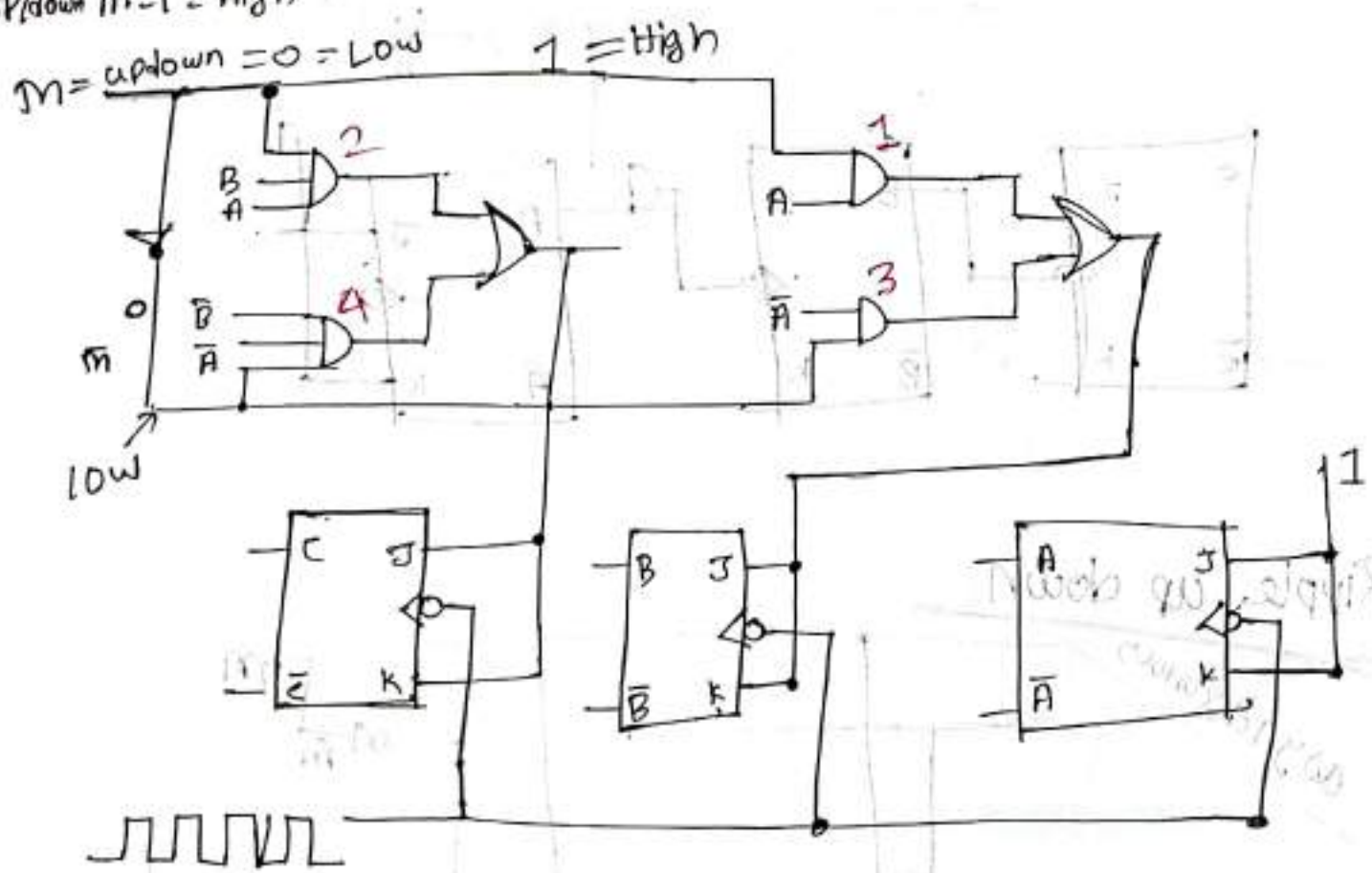
⊛ MOD (10) Decade counter.





count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

- updown m=1 = High

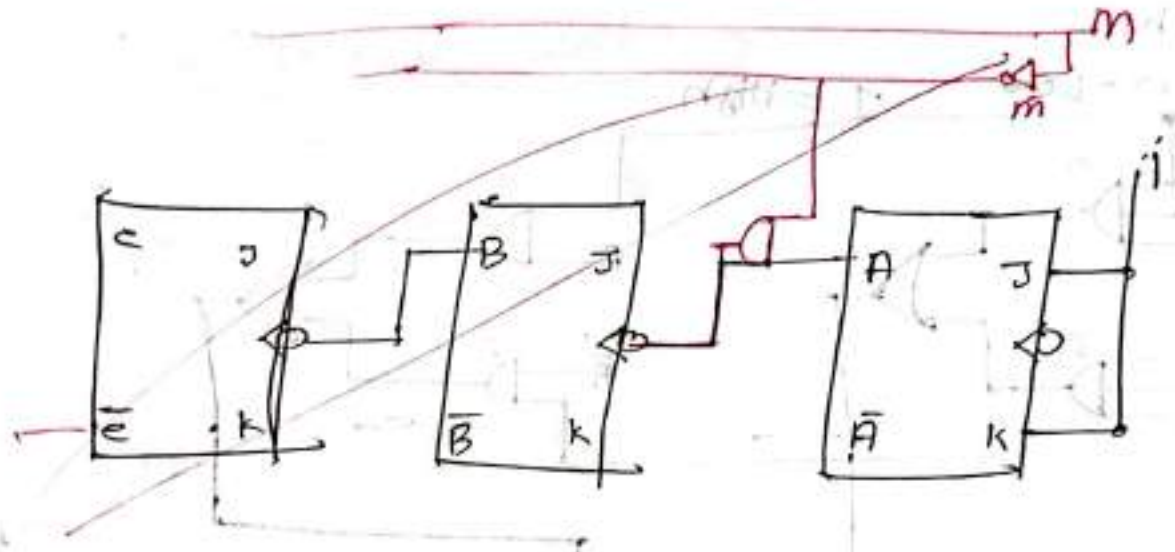


if updown is high.

→ AND gate 1, 2 enable and 3, 4 gate are disable.

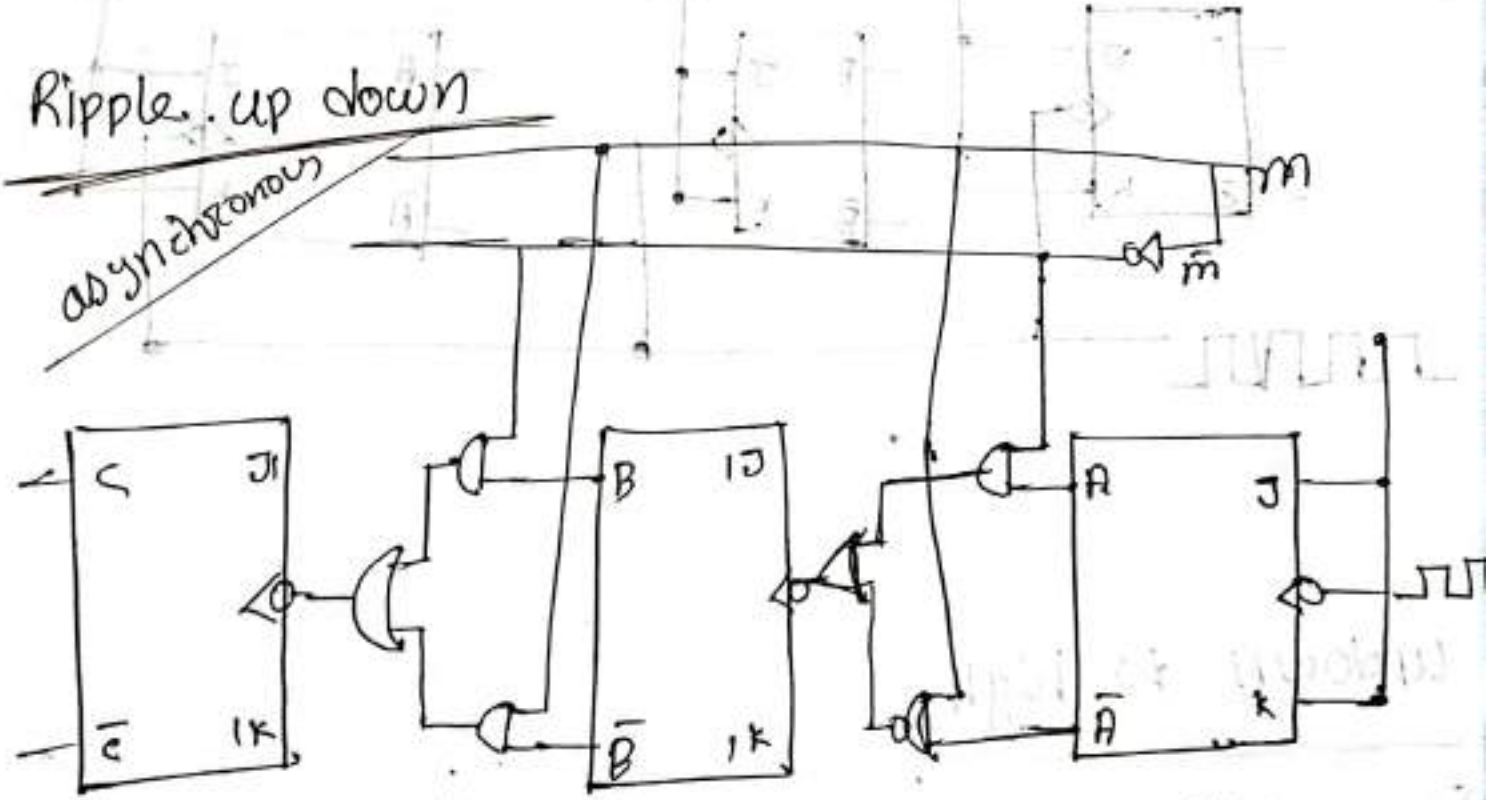
→ This allows A and B outputs gates 1 and 2 control the J and K inputs of FFs B and C

→



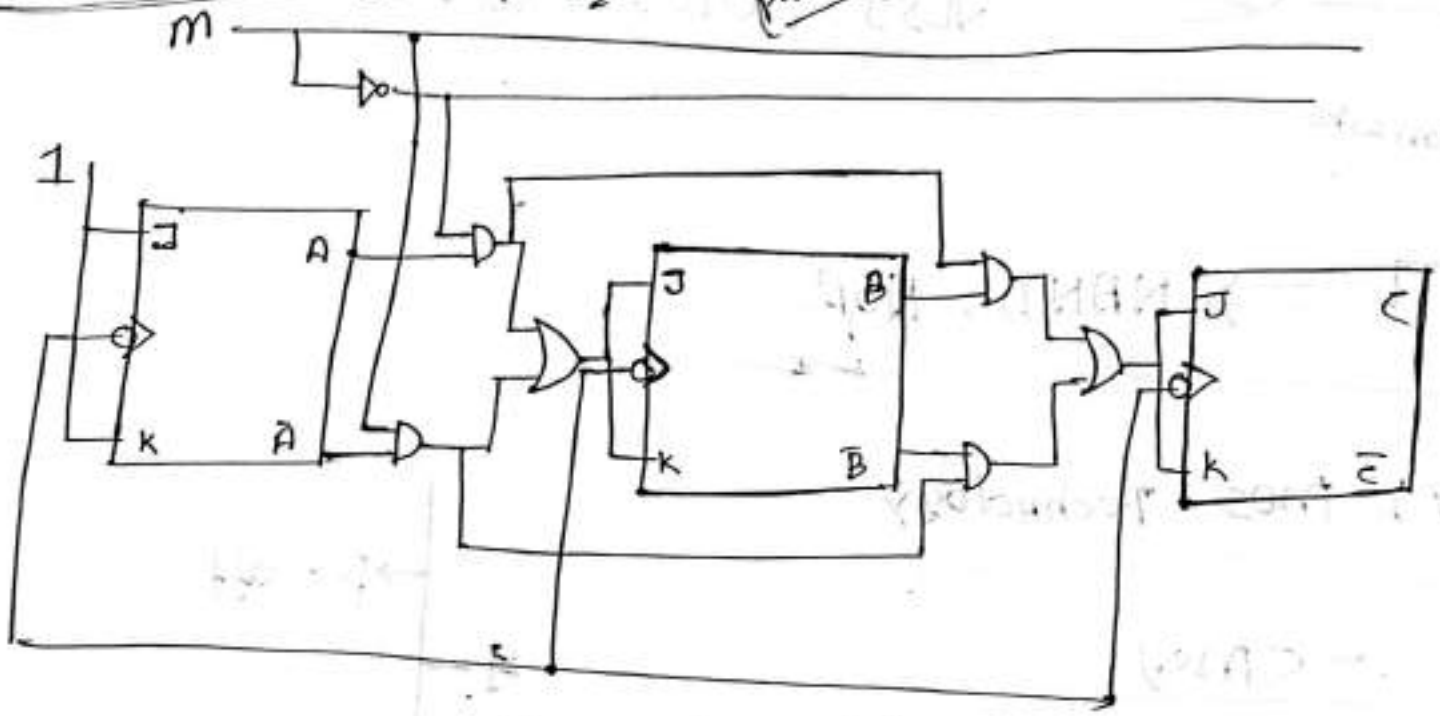
Ripple up down

asynchronous



Synchronous up down

parallel

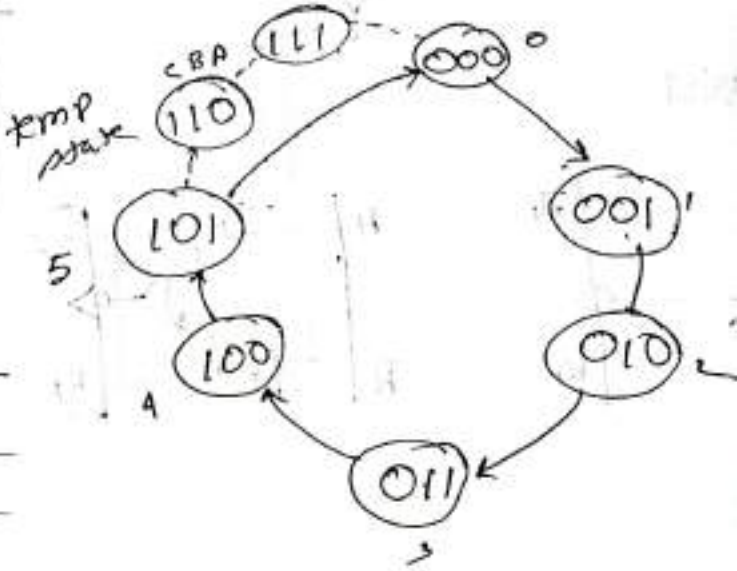


MOD-6 Asynchronous (Ripple) counter

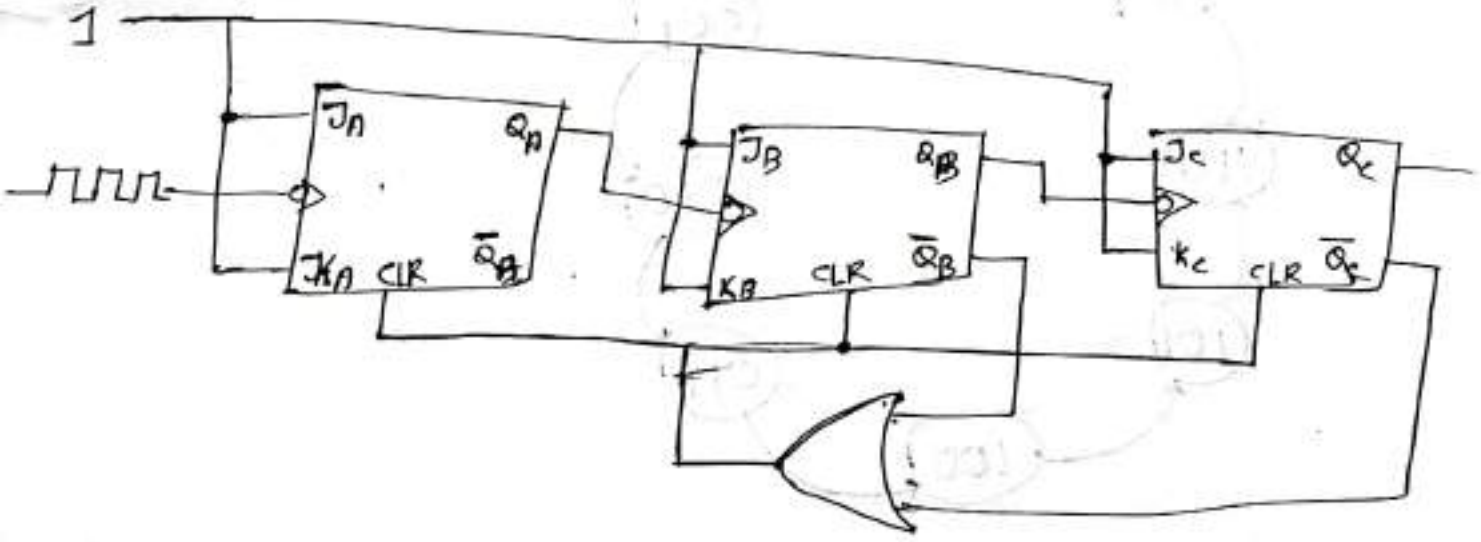
count	C	B	A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

mod 6 →

temp state

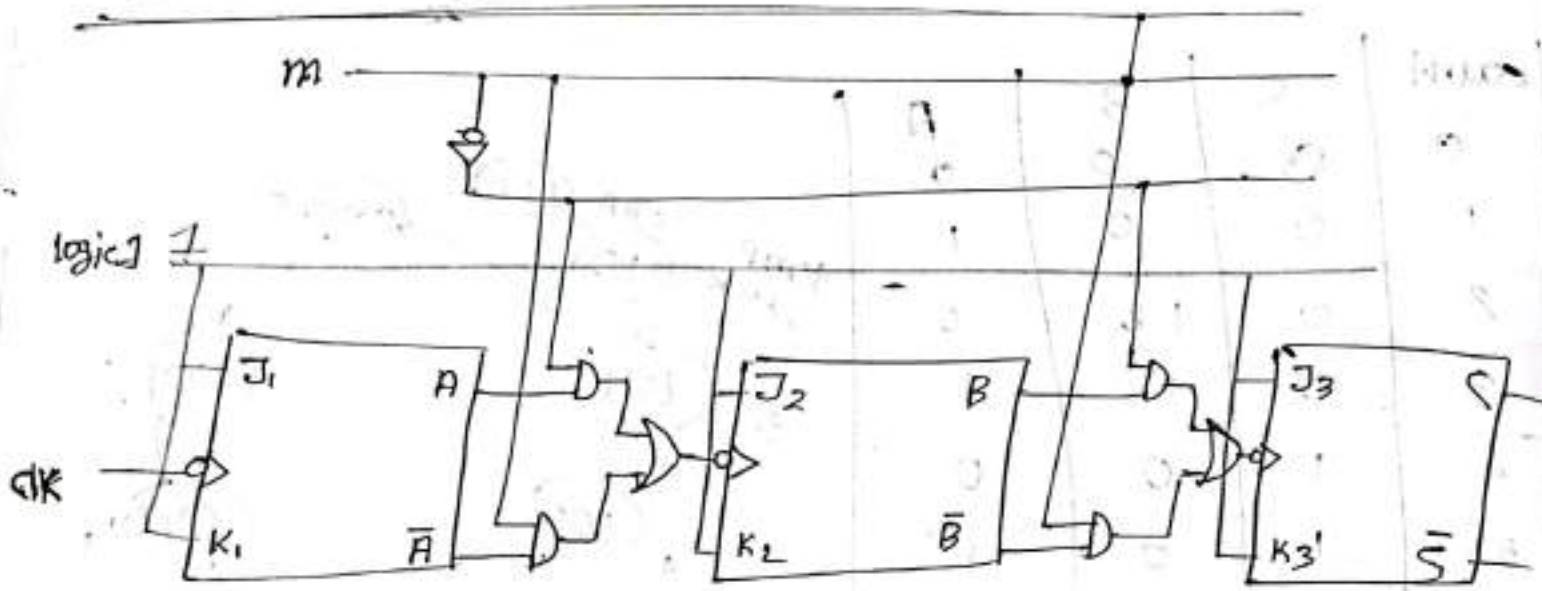


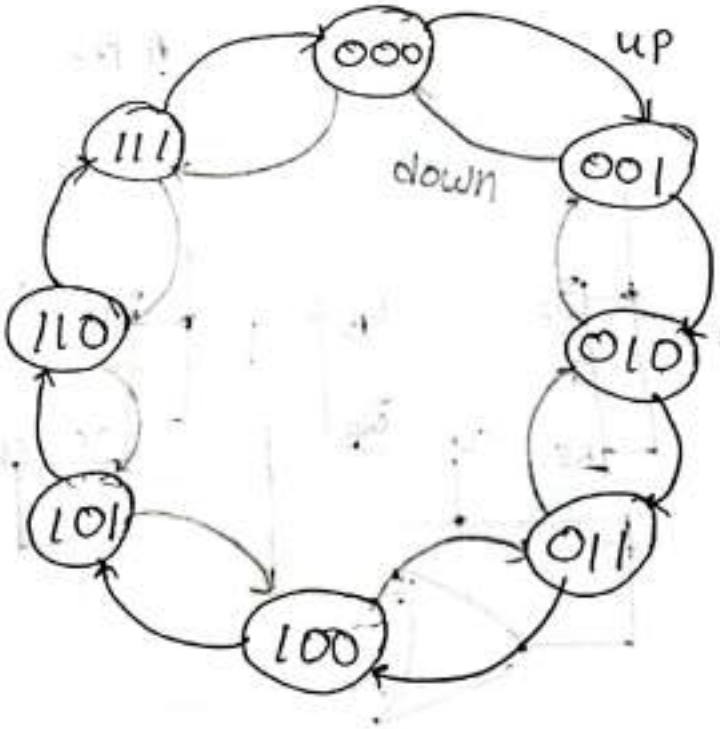
A B C



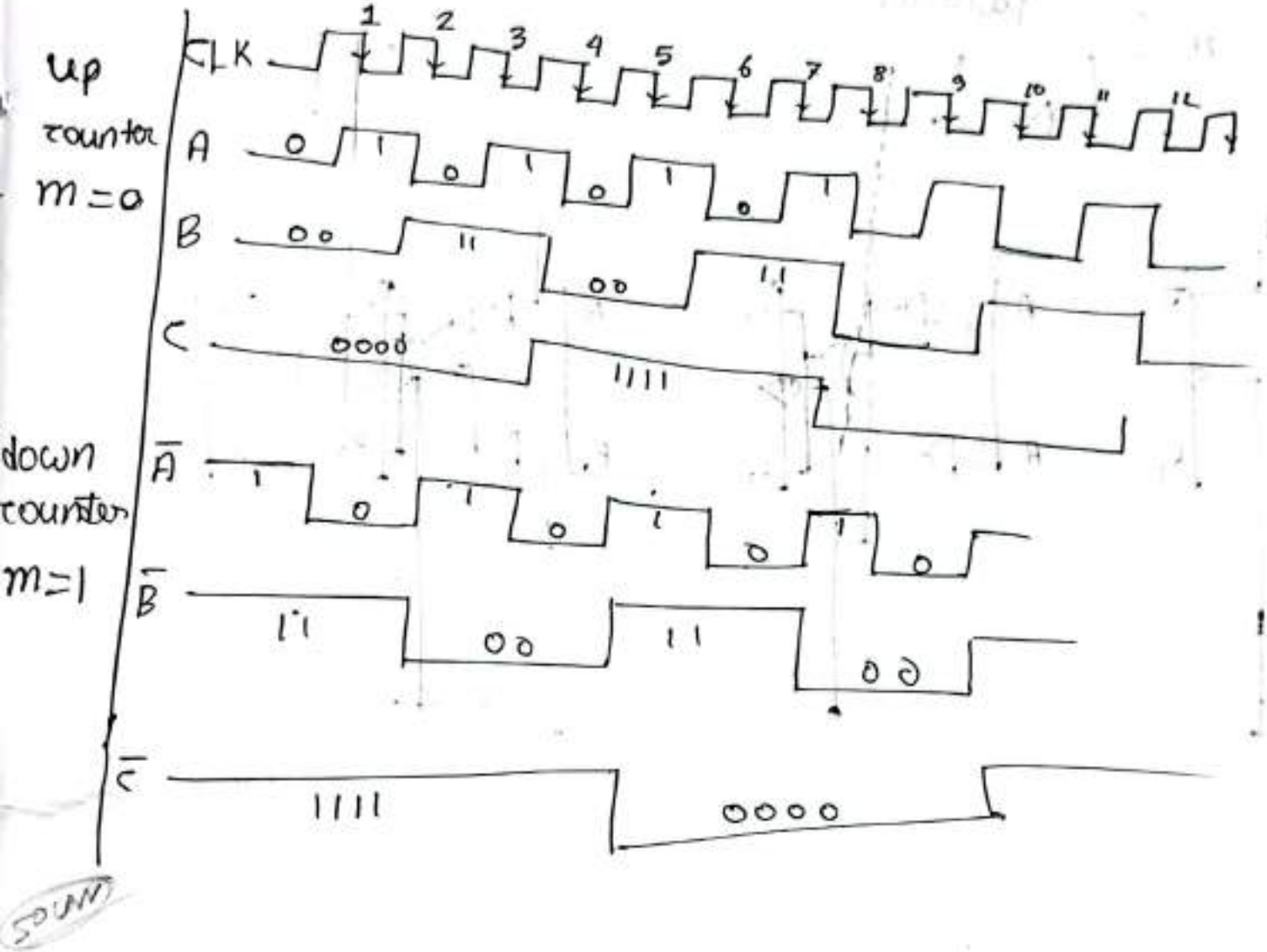
Same as another counter.

Synchronous ~~Ripple~~ up/down counter



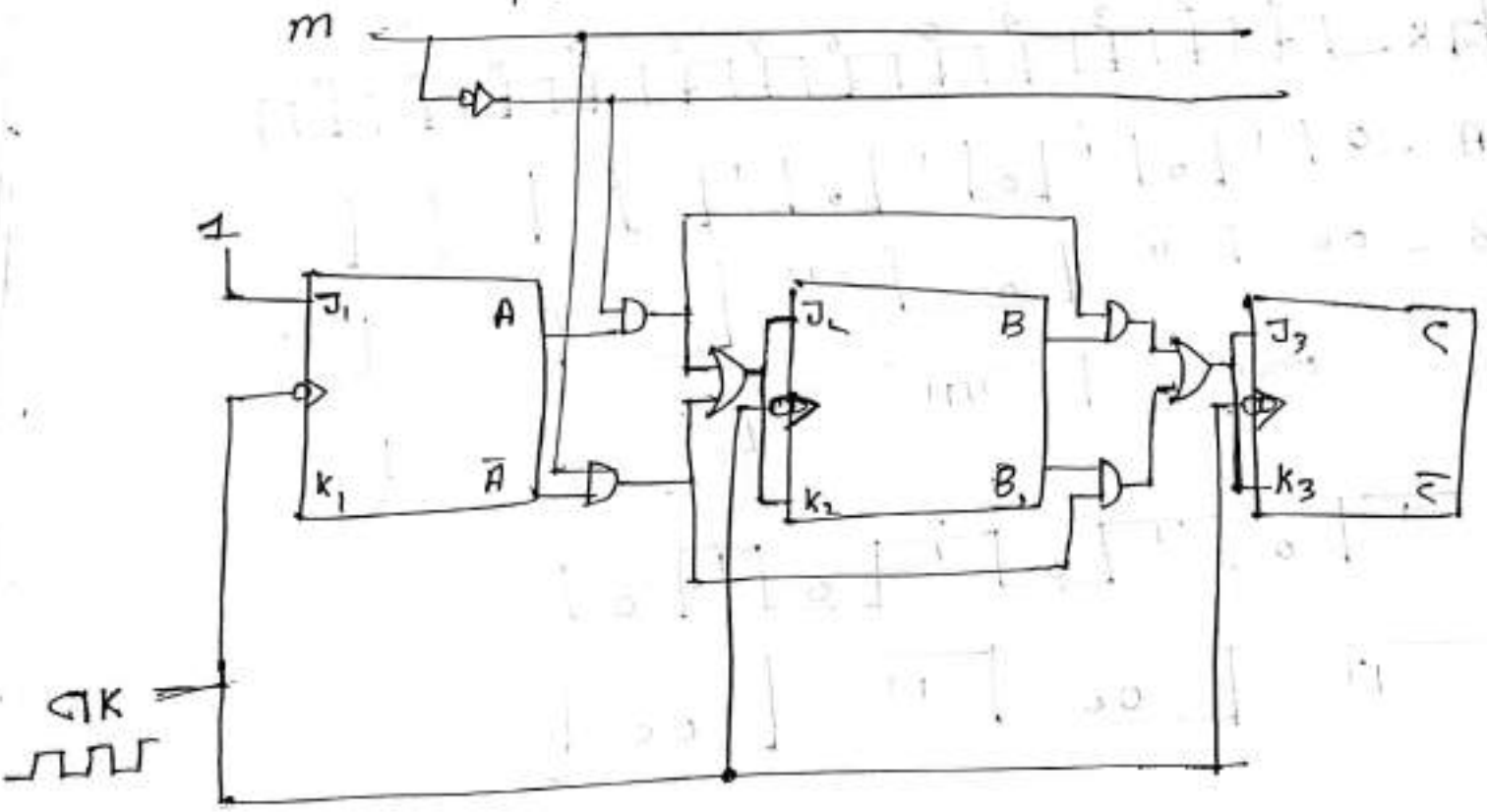


Timing diagram

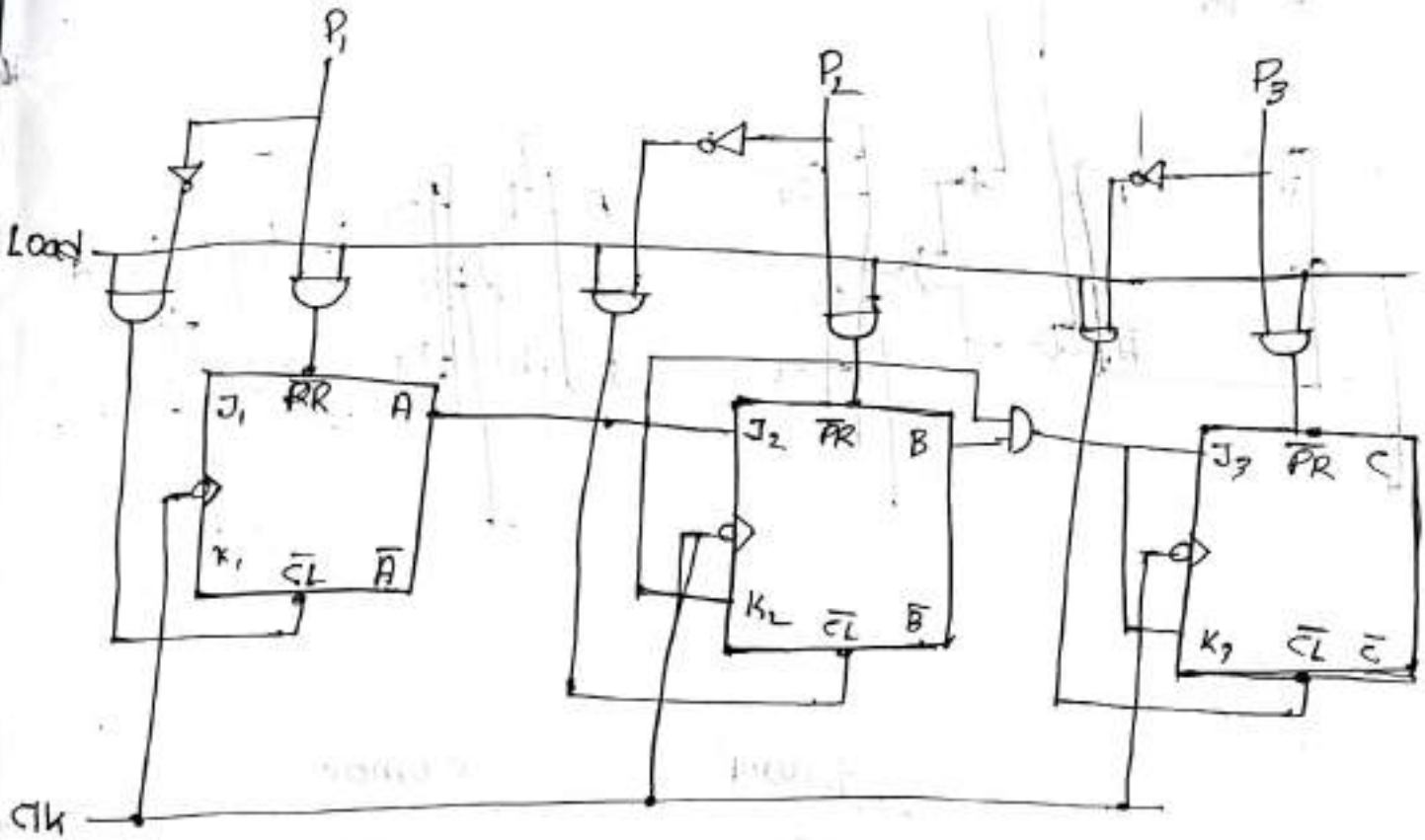


: synchronous / up and down counter

parallel

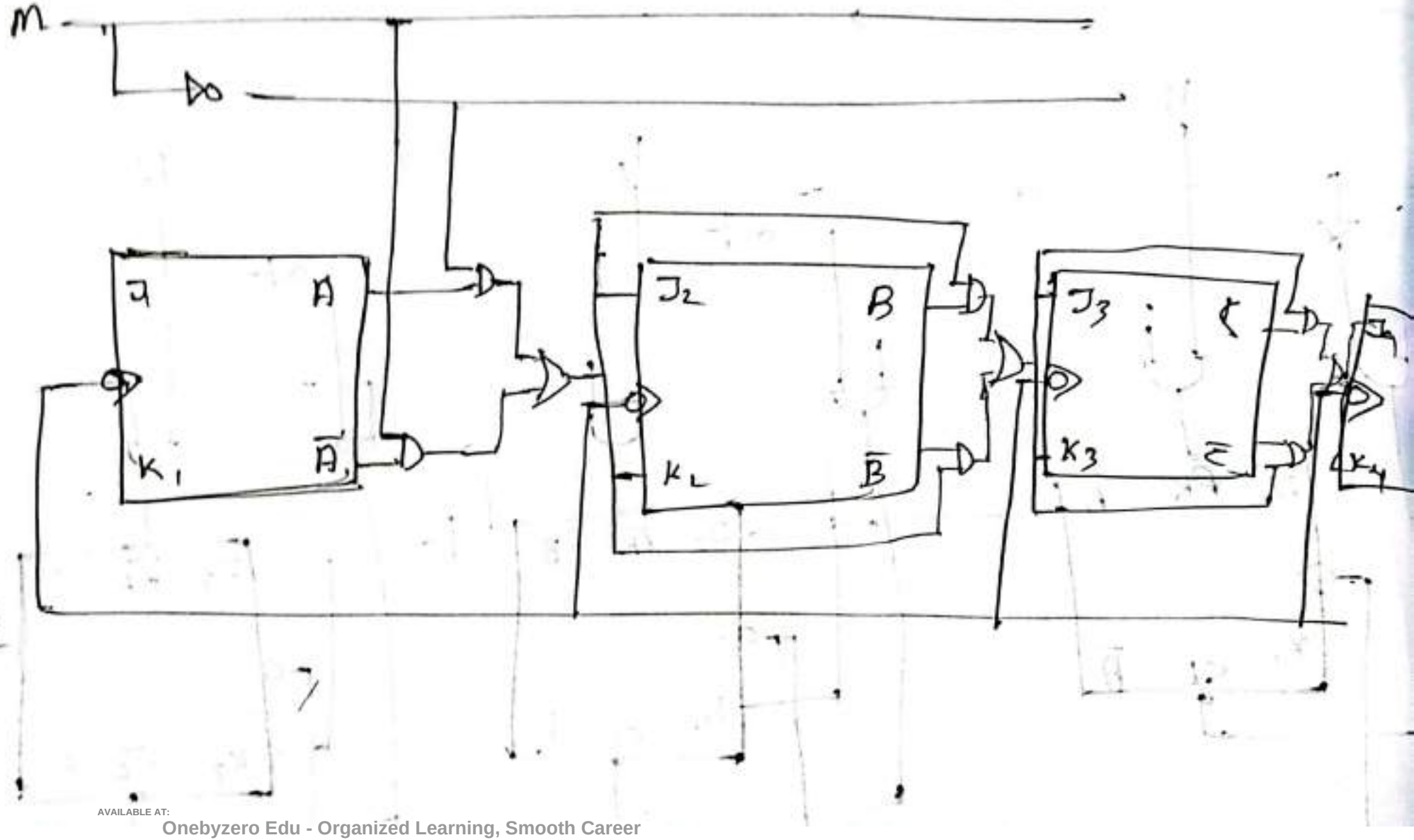


Presettable counter



Synchronous counter with asynchronous parallel load

Synchronous up/down mod 16 counter:—



Explain the working principle of a Synchronous up down Mod-8 counter with appropriate diagram.

A Synchronous up/down Mod-8 counter counts in both up and down directions based on a control signal. Here is an explanation of its working principle:

Control Signal (Up/down): This determines the direction of counting. When the control signal is HIGH (Up/down = 1) the counter counts up and when the control signal is LOW (Up/down = 0) the counter counts down.

AND (gates): It controls which signals (normal or inverted) are fed to the J and K inputs of the flip-flops (FFs).

- When Up > Down = 1 : Normal outputs are fed through enabled AND gates, allowing upward counting.
- When Up > Down = 0 : Inverted outputs are fed through enabled AND gates, enabling downward counting.

State Transition: The counter moves between state on the negative edge of the clock pulse. The direction of the state transition is based on the control input.

Mod-8: Since it is a Mod-8 counter, it can count from 0 to 7 and then warps around.

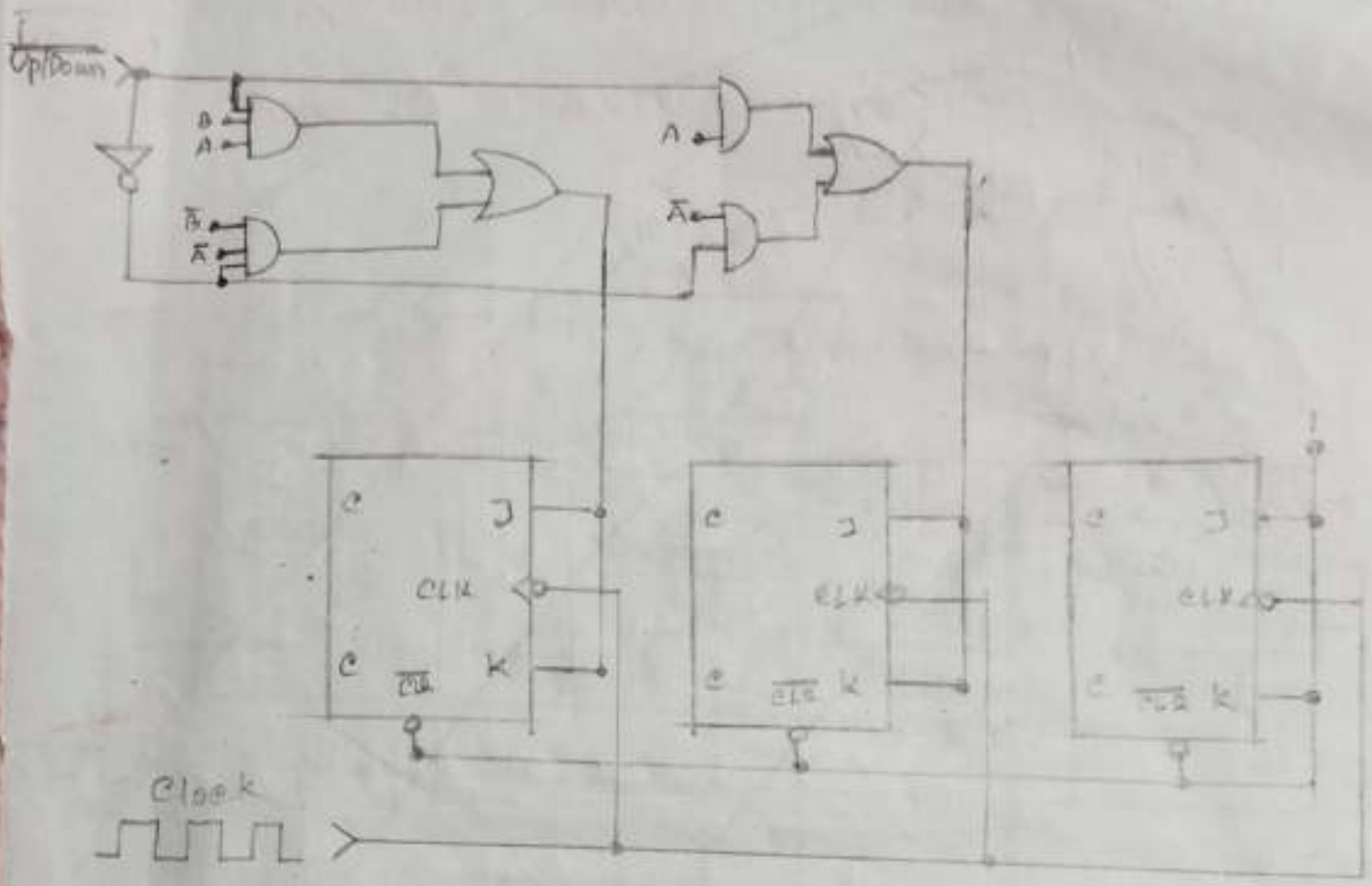
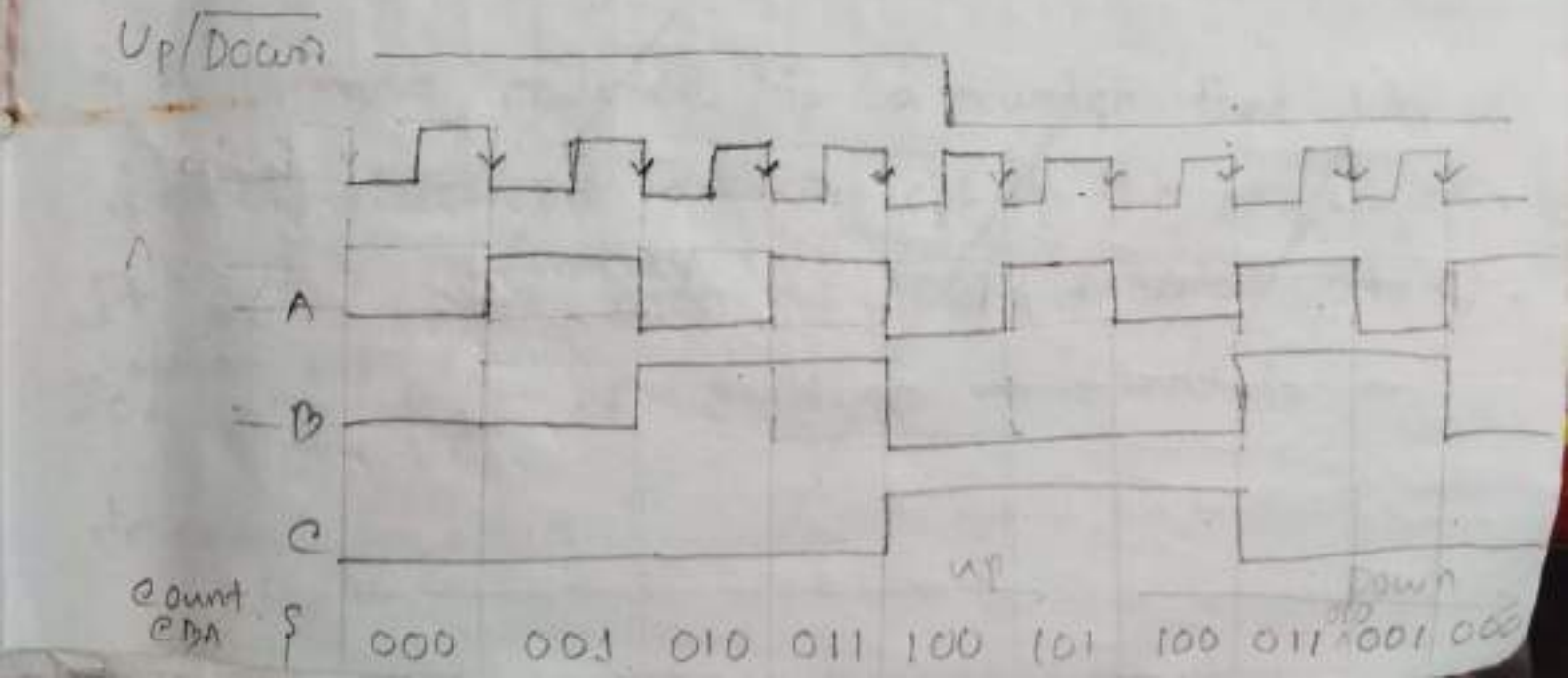
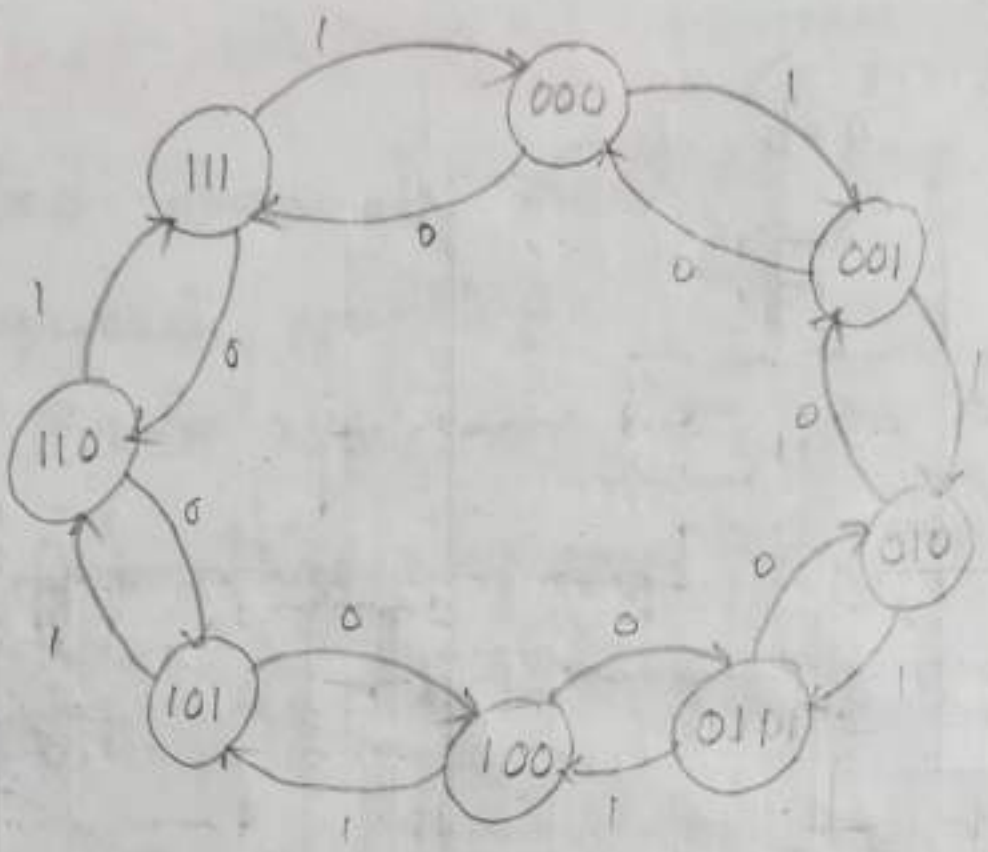


Fig: Schematic synchronous Mod-8 up/down counter





state diagram

G1152

15235

R_1 = Decentralization / hash

R_2 = Volatility / valuation

[4]. Explain 3-bit/4-bit asynchronous (ripple) counter.

A 3-bit/4-bit asynchronous (ripple) counter operates using four flip flops (A, B, C, D) each represents one bit of a binary number. The D flip-flop is MSB and the A flip-flop is LSB.

3-bit counters can count 000 to 111 (decimal 0 to 7)
4-bit " " 0000 to 1111 " 0 to 15)

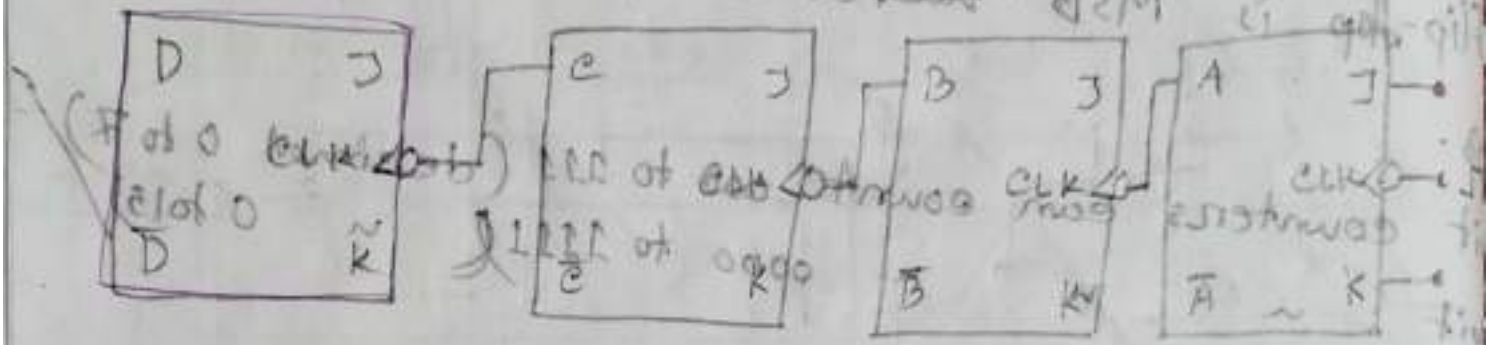
Cloking Mechanism:

① The clock pulses are applied only to flip-flop A. This toggles (change state) on every negative edge (HIGH to LOW transition) of the clock.

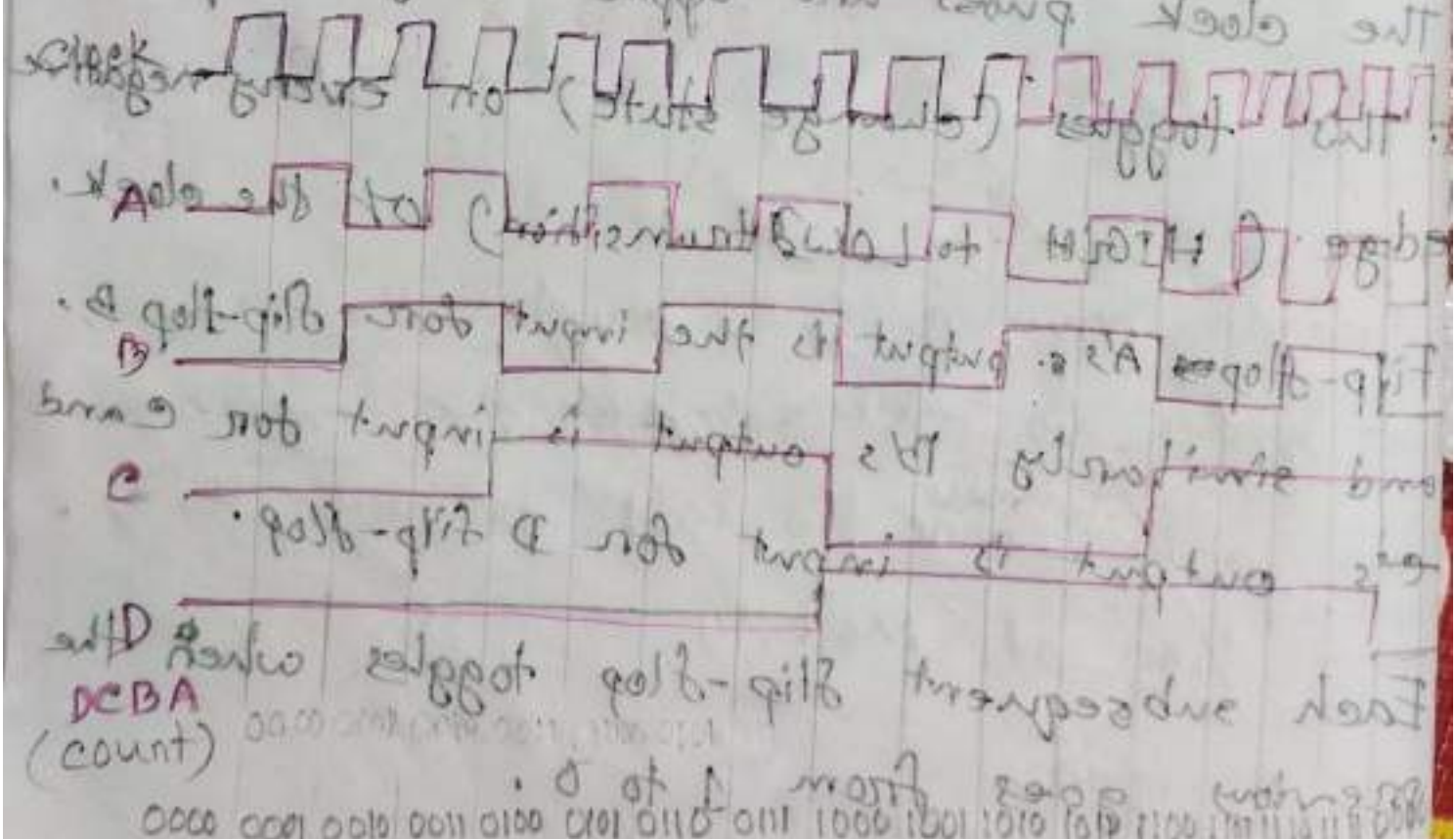
② Flip-flop A's output is the input for flip-flop B. and similarly B's output is input for C and C's output is input for D flip-flop.

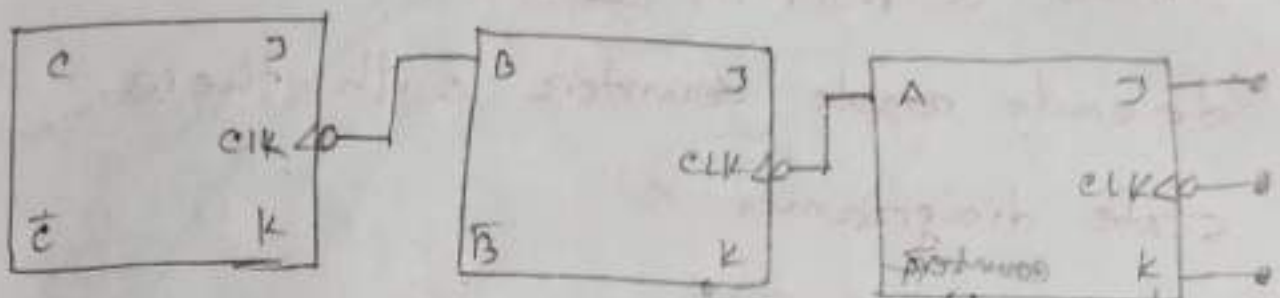
③ Each subsequent flip-flop toggles when the previous goes from 1 to 0.

(9) After reaching 1111 (15) the next clock pulse brings the counter back to 0000. It happens when complete 8/16 clock pulse.

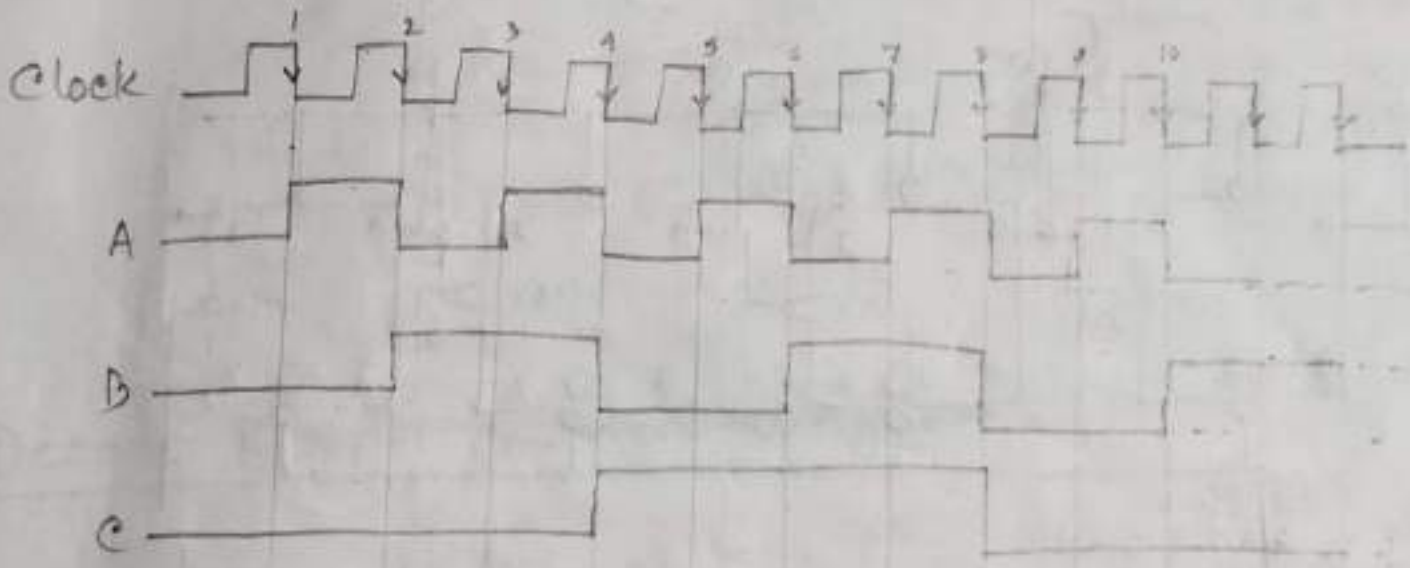


All J and K input assumed to be 1.





All inputs J and K are assumed to be 1

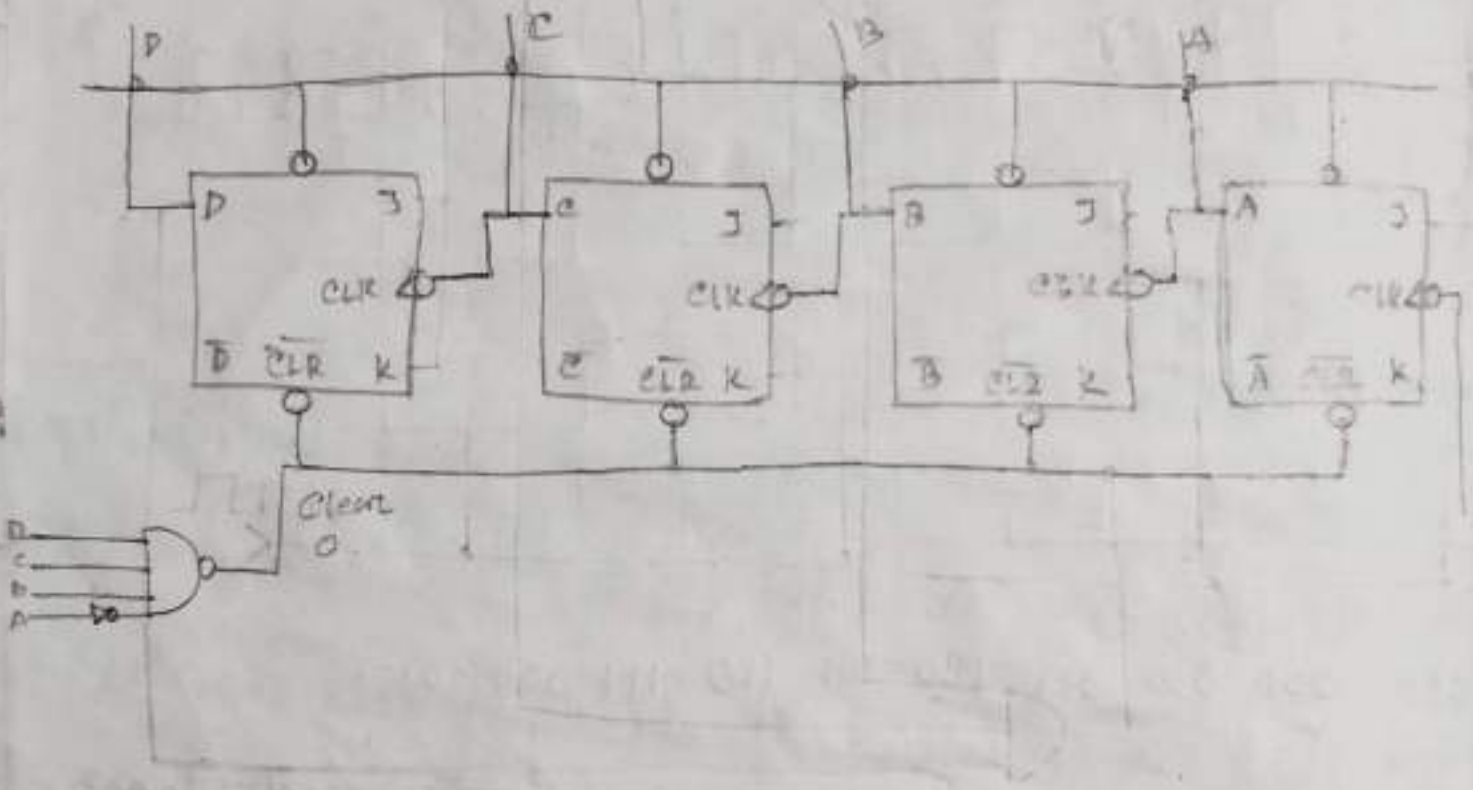


C B A (count)	000	001	010	011	100	101	110	111	000	001

→ recycle to 000

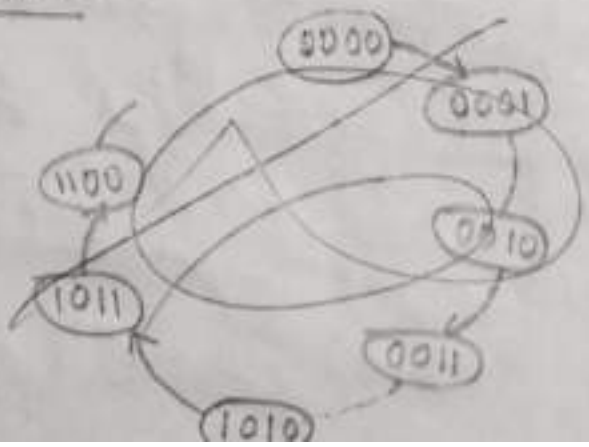
[14] Draw a Mod-14 ~~counter~~ and a decade ripple counter with their state diagram.

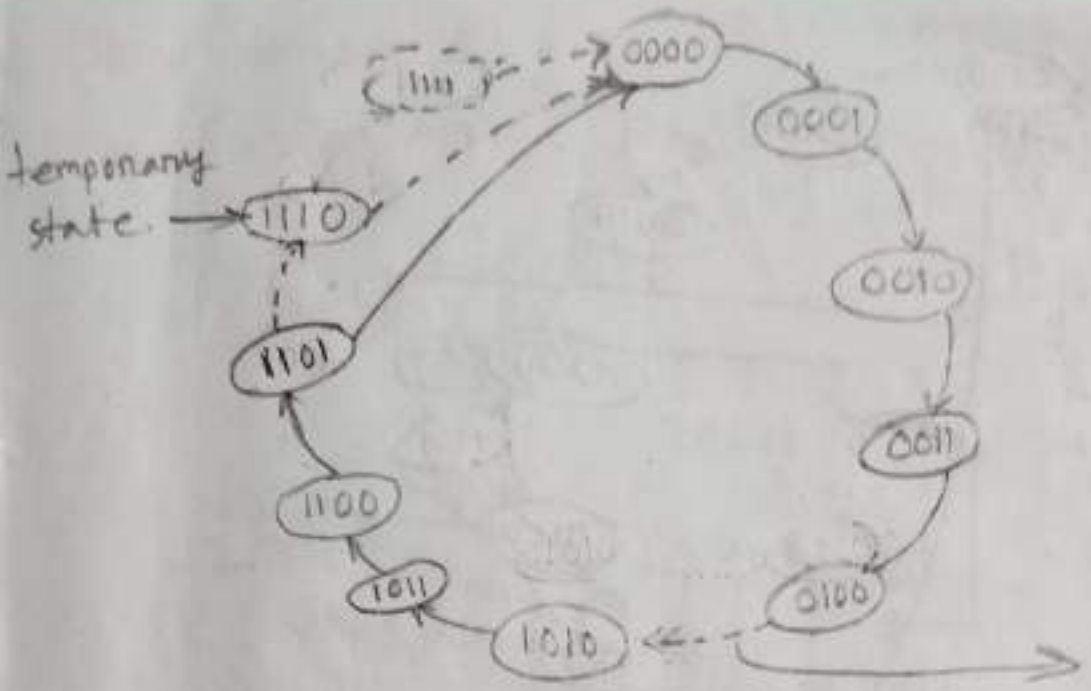
Mod-14 counter diagram:



Mod-14

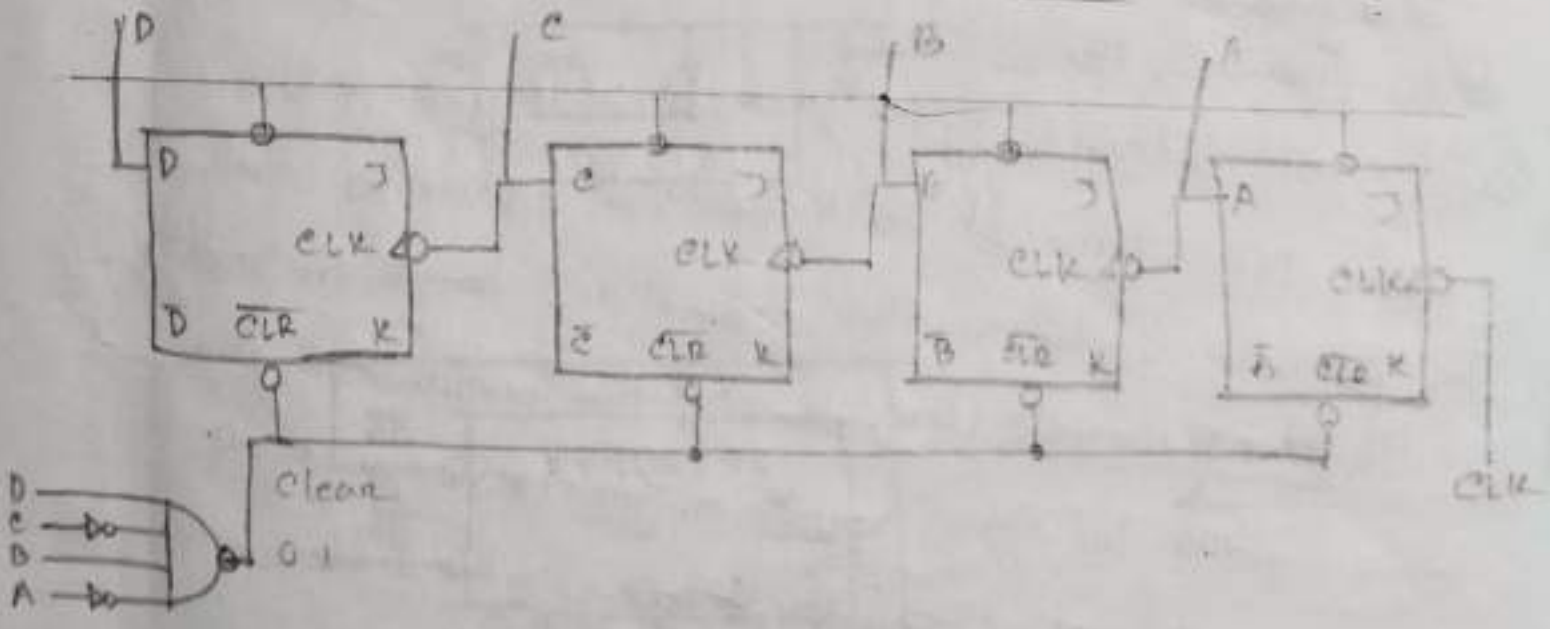
State transition Diagram:





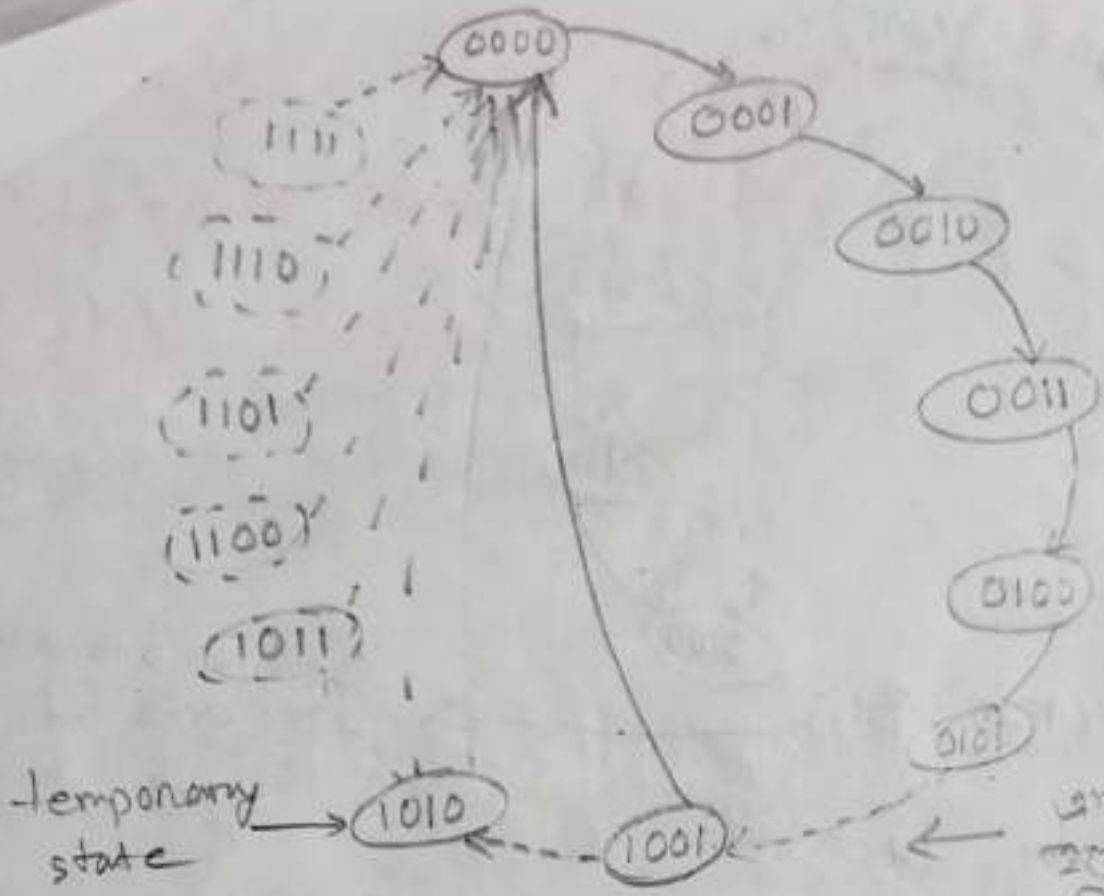
State transition diagram of Mod-14

Decade Ripple counter diagram:



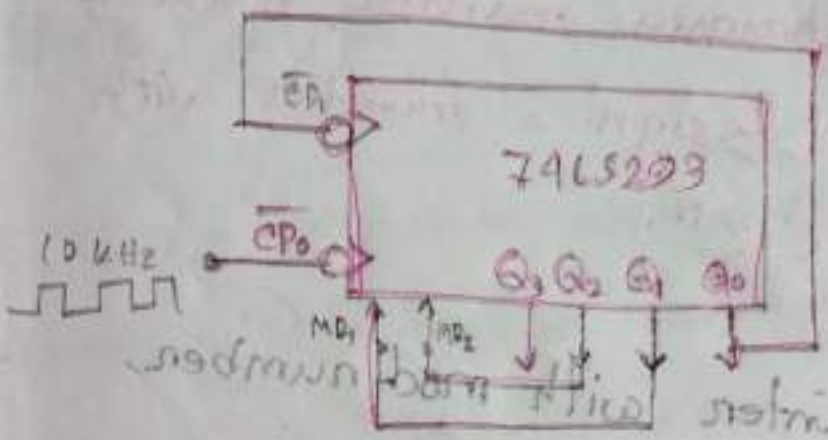
Decade counter can counts binary 0000 to 1001.

Decade counter can counts binary 0000 to 1001. For this any Mod-10 counter is to be a decade counter



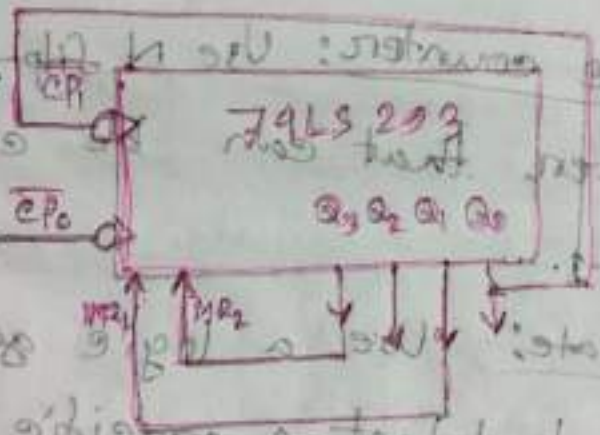
state transition of decade (Mod-10) Ripple counter diagram

How to wire the 74LS293/74293 as a Mod-6 counter? / Mod-10 / Mod-14.

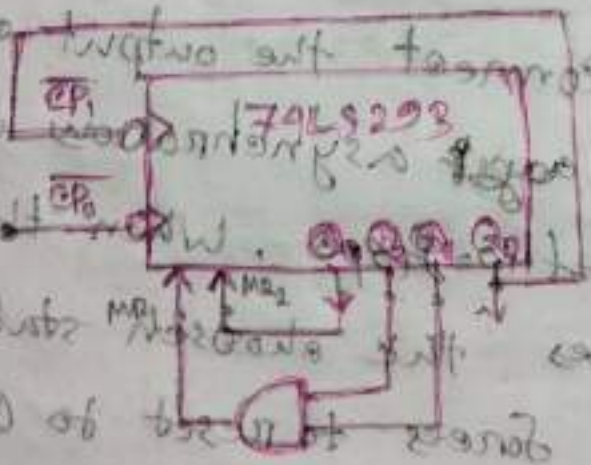


6 = 0110

Mod 6



10 = 1010



14 = 1110

What is Mod number? What is ripple counter? What is decade counter? What is synchronous counter? What is the method to design a counter with mod no number $< 2^N$.

To design a counter with mod number, follow these steps:

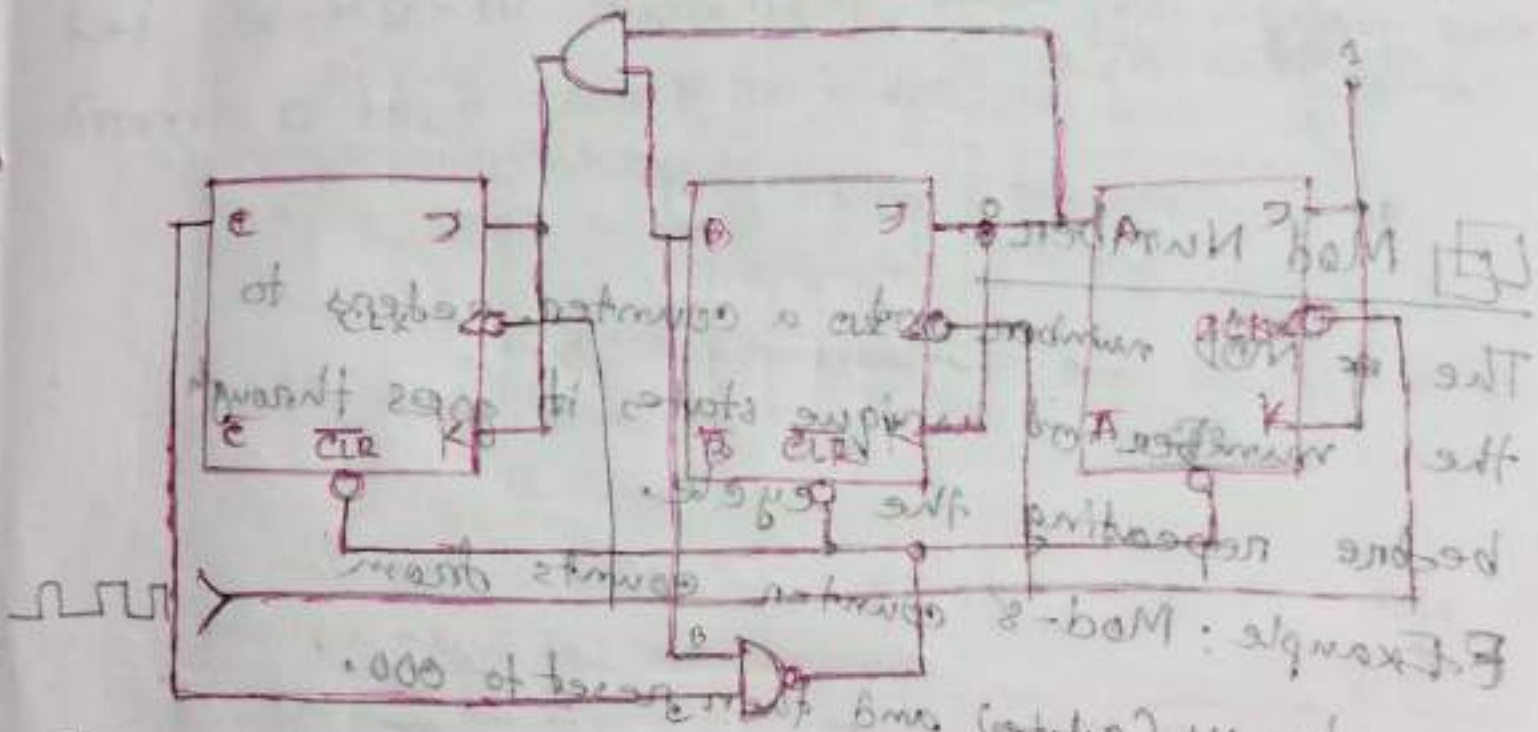
① Start with a basic counter: Use N flip-flops to create a counter that can be count from 0 to $2^n - 1$.

② Add a logic gate: Use a logic gate (like NAND gate) to detect a specific state in count where the counter back to reset.

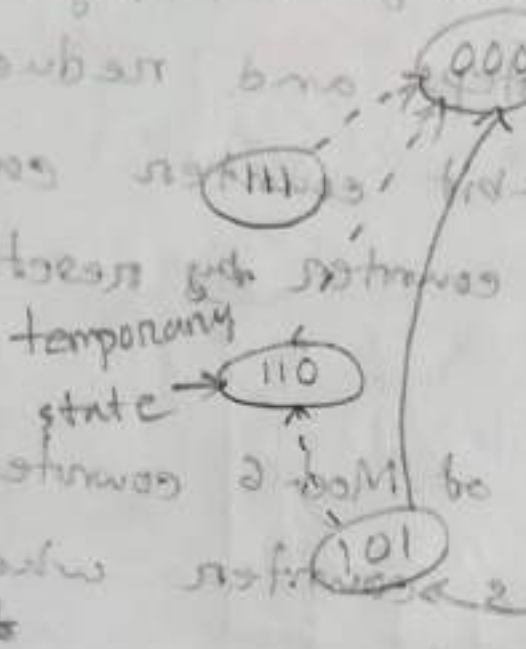
③ Trigger reset: Connect the output of the logic gate to the asynchronous clear (CLR) inputs of the flip-flops. When the counter reaches the choosen state, the logic gate forces to reset to 000.

2. Skip certain states: By resetting early, the counter skips next steps and reducing the Mod number. Example 3-bit counter can be modified as Mod-6 counter by resetting at state 110.

Here is a diagram of Mod-6 counter produced by clearing a MOD-8 counter when it count of six (110).



B	A
0	0
0	1
1	0
1	1
0	0
1	0
1	1
0	0



temporary state needed to clear counter

Mod Number:

The MOD number of a counter refers to the number of unique states it goes through before repeating the cycle.

Example: Mod-8 counter counts from 000 to 111 (8 states) and then reset to 000.

Ripple Counter:

A ripple counter is an asynchronous counter where the clock signal applied only to the first flip-flops (FF). Each subsequent flip-flops clock is driven by the output of

previous flip-flop.

Synchronous counter:

A synchronous counter is a counter where all flip-flops are triggered simultaneously by the same clock signal.

Decade Counter:

A decade counter is a type of counter that has a MOD-10 sequence, meaning that it counts from 0 to 9 (0000 to 1001) and then resets to 0000.

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Chapter - 8

Digital IC

- A chip is a small piece of semiconductor material
- Using advance photographic, miniature circuits can be produced on the surface of the chip.
- The components (resistors, diode, transistor) are an integral part of the chip.

Level of Integration:

Digital IC's can be classified according to the number of gates on a chip, this is referred as level of integration.

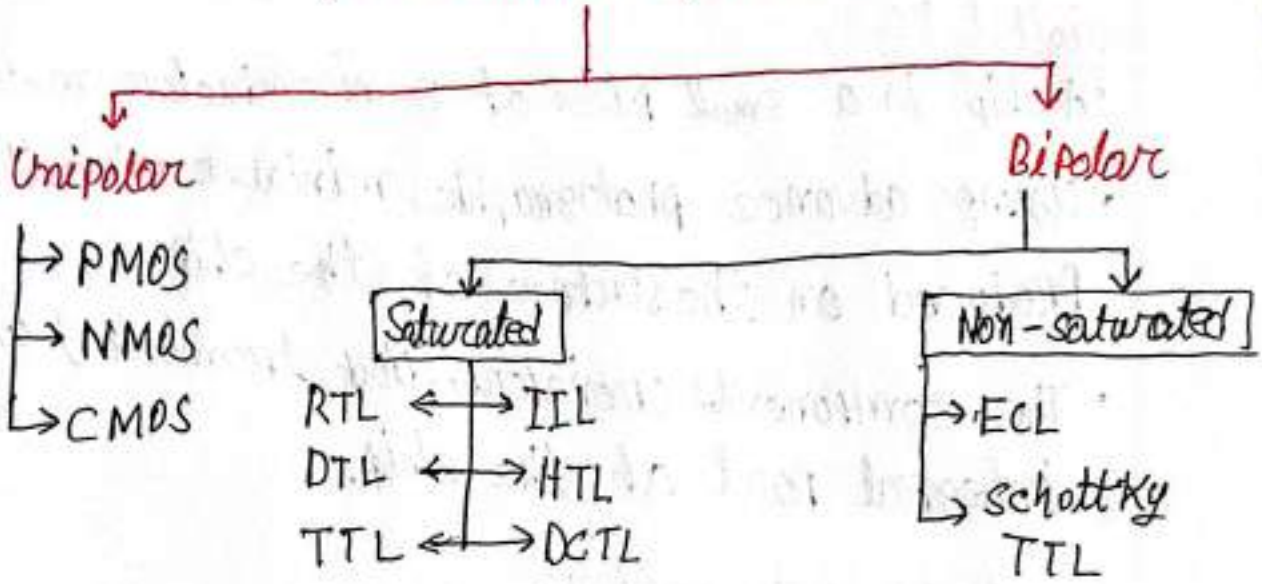
IC classification	Number of gates on a chip
Small scale integration (SSI)	Less than 12
Medium " " (MSI)	12 to 99
Large " " (LSI)	100 to 999
very large scale " (VLSI)	1000 to 9999
Ultra " " (ULSI)	10,000 or more

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Semiconductor Digital IC's



Logic Family

→ Unipolar Logic Family :

↳ In unipolar logic families, unipolar devices are the key element.

↳ MOSFET (Metal oxide semiconductor Field Effect Transistor) is a unipolar devices, in which the current flows because of only one type of charge carriers

GOOD LUCK

TOPIC NAME : _____ DAY _____

TIME : DATE : / /

→ Bipolar Logic Family :

↳ Transistors and diodes are bipolar devices, in which the current flown because of both the charge carriers (electrons and holes).

↳ In bipolar logic families, transistors and diodes are used as key element.

- PMOS : P- channel MOSFET family
- NMOS : N- " " "
- CMOS : Complementary " "
- RTL : Resistor transistor logic
- DCTL : Direct coupled " "
- IIL : integrated injection logic
- DTL : Diode transistor logic
- HTL : High threshold logic
- TTL : Transistor transistor logic
- ECL : Emitter coupled logic

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Characteristics of IC's

1. Propagation Delay (speed)
2. Power dinipation
3. figure of merit
4. Fan-in / Fan-out
5. current and voltage parameters
6. Noise Immunity
7. operating Temperature

1. Propagation Delay (speed)

- The propagation delay is defined as "the time required to change the output from one logic ~~sate~~ state to other logic sate after input is applied."
- Manufacturer specifies the speed of digital family in terms of propagation delay.
- practically propagation delay is calculated by measuring the two propagation delays.

1. t_{PHL}: Delay required to change from ~~a~~ high to low sate

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

2. t_{PLH}: Delay required to change from low to heigh

- Higher is the propagation delay. lower is the speed.

2. Power Dissipation:

- It is indirectly related with propagation delay.
- Low power dissipation is desirable because it generates less heat and therefore it avoids cooling, power supply and cost problem.
- Low power dissipation means it should operate with minimum voltage within minimum current.
- The required current can be adjusted by increasing the resistance but it increases the time because $T = R \times C$
- For example, a gate takes 2mA current and it is operated with 5V supply then its power dissipation per gate is calculated as

power dissipation $= 2mA \times 5V = 10mW$

GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

3. Figure of Merit

- This characteristics indicates the total performance or quality of a family.
- If resistance of the circuit is increased to reduce the current or power dissipation then the propagation delay increases.
- In opposite way, if we reduce the propagation delay to make it faster by reducing the resistance then more current is drawn by the gate which increases the power dissipation
- It is defined as "the product of propagation delay and power dissipation" of the gate
- Figure of merit is measured in terms of "pJ"

$$\text{figure of merit} = F.M = t_p \times P_D$$
$$= 1 \text{ns} \times 1 \text{mW}$$
$$= 1 \text{pJ}$$

- The lowest figure of merit indicates the best performance

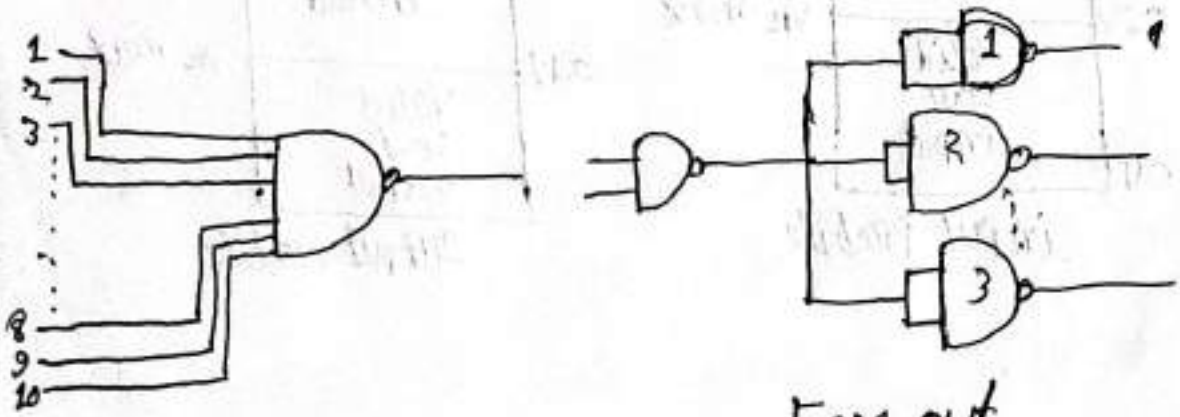
TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

4. Fan In - Fan out

The capacity of driving the loads of a gate is expressed in terms of fan out.

- Fan in is the "number of inputs to a gate".
- Fan out is also known as loading factor.
- E.g. in case of TTL fan in and fan out is 10 that is maximum 10 TTL gates can be connected to the output terminal of a TTL gate.



Fan-in

Fan out

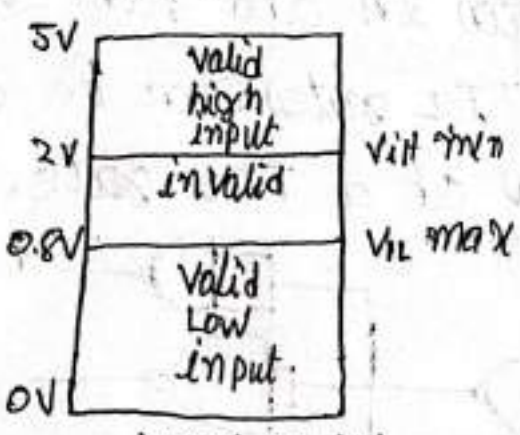
~~5. Manufacturer Specifics~~.

TOPIC NAME : _____ DAY : _____

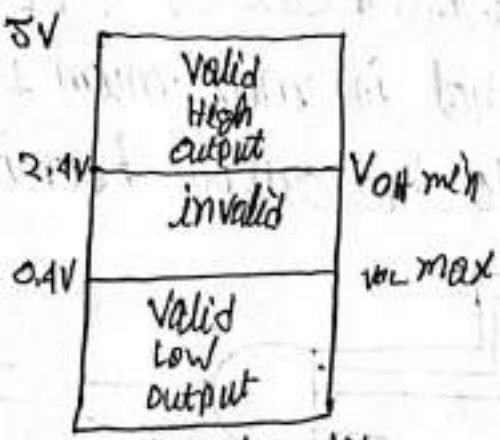
TIME : DATE : / /

5. Current and voltage parameters

- Manufacturer specifies these parameters in the data book, which are useful in designing of digital circuits.
- This is also known as worst case current and voltage parameters.



input Profile



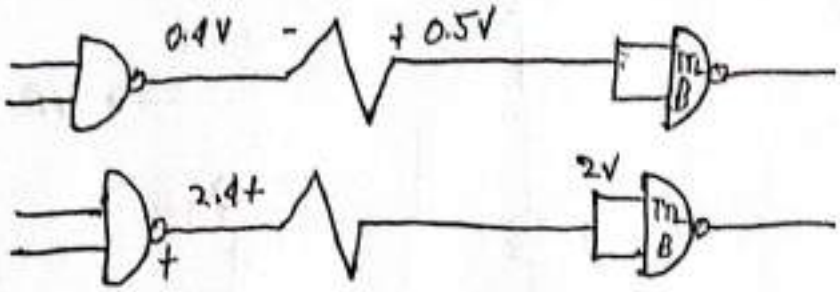
Output Profile

TOPIC NAME : _____ DAY _____

TIME : DATE / /

6. Noise Immunity (Noise Margin)

- The maximum noise voltage a device can withstand without making a false change in output
- The noise may be an external signal generated by vehicle ignition, signal generator.



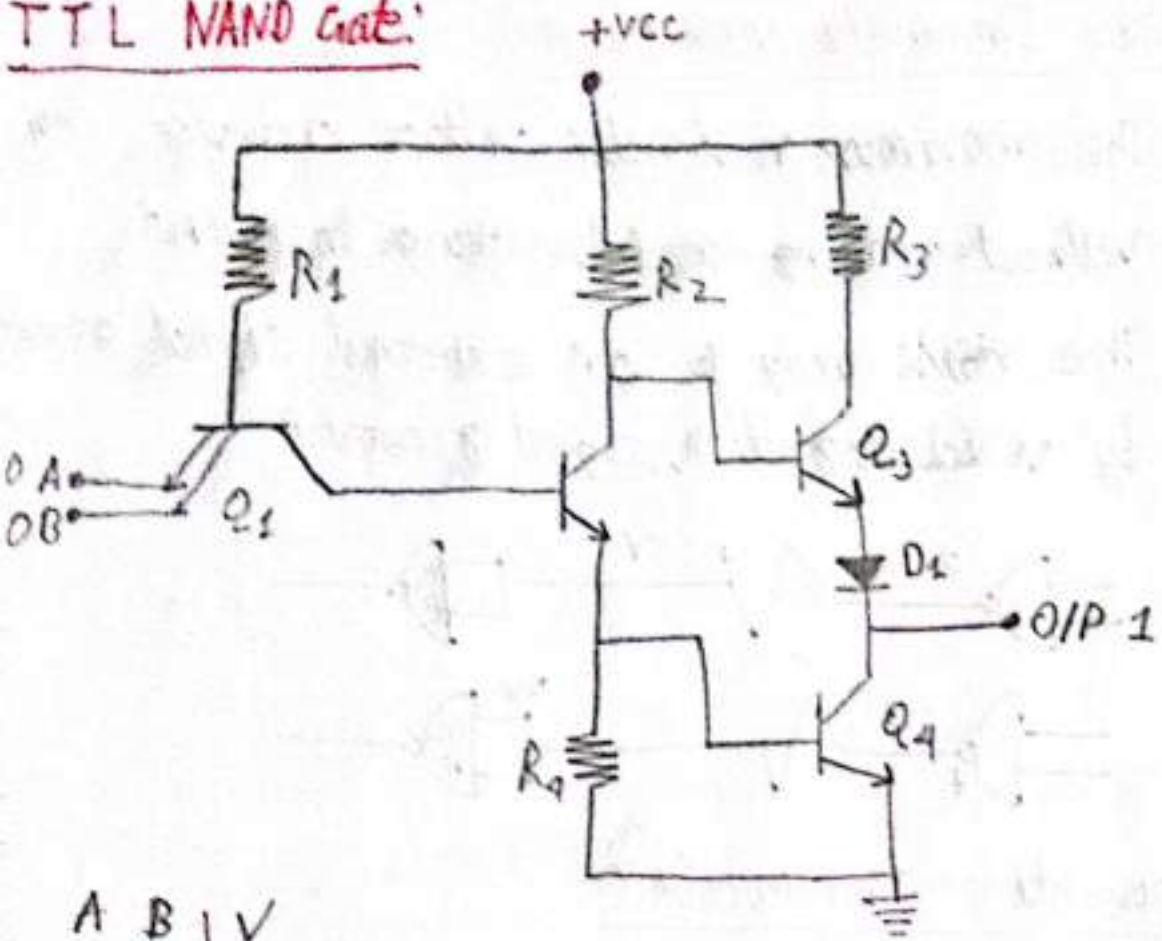
7. Operating Temparature

- Manufacturer specifies the temparature range in which the device operates properly.
- For industrial and consumer application the temparature range is 0 to 70°C
- For military applications it is 55 to 125°C
- In case of, TTL, 74XX series it for industrial and consumer applications while 54XX series is for military application.

TOPIC NAME _____ DAY _____

TIME _____ DATE :

TTL NAND Gate:

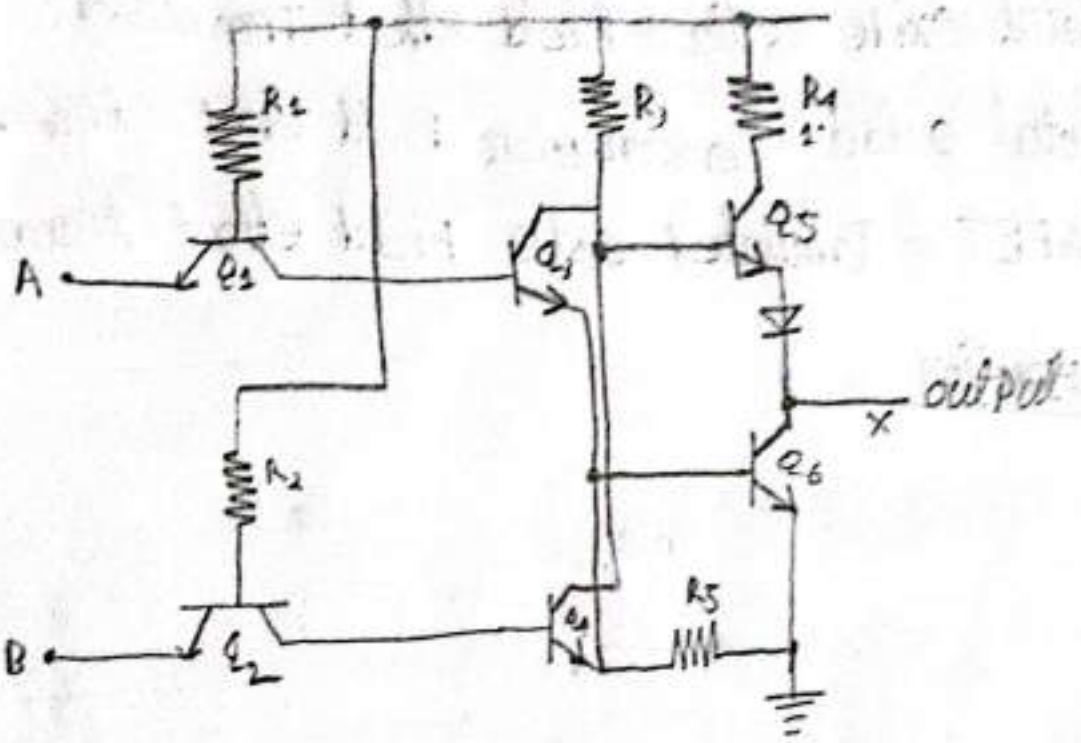


A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

TOPIC NAME : _____ DAY _____

TIME : DATE / /

TTL NOR Gate:



Q_5 = ON = output = High(1)

Q_6 = ON = output = Low(0)

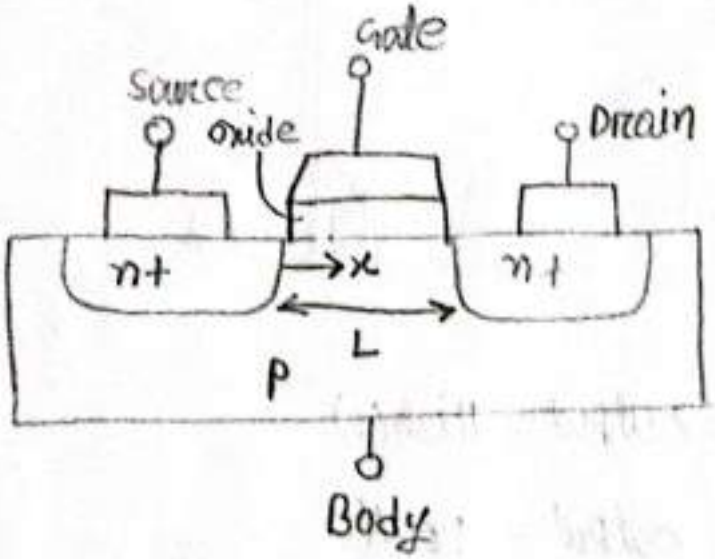
A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Y
0	0	ON	ON	OFF	OFF	ON	OFF	1
0	1	ON	OFF	OFF	ON	OFF	ON	0
1	0	OFF	ON	ON	OFF	OFF	ON	0
1	1	OFF	OFF	ON	ON	OFF	ON	0

TOPIC NAME ..... DAY .....
TIME ..... DATE ..... / ..... / .....

~~Mos.fet~~
MOSFET

- Metal oxide silicon Field effect Transistor
- Metal oxide Semiconductor Field effect transistor
- IGFET = Insulated Gate Field effect transistor

Construction



- Four-terminal device having source(S), gate(G), drain(D) and body(B) terminals.
- An oxide layer is deposited on the substrate & the gate terminal is connected to it.
- This oxide layer acts as an insulator

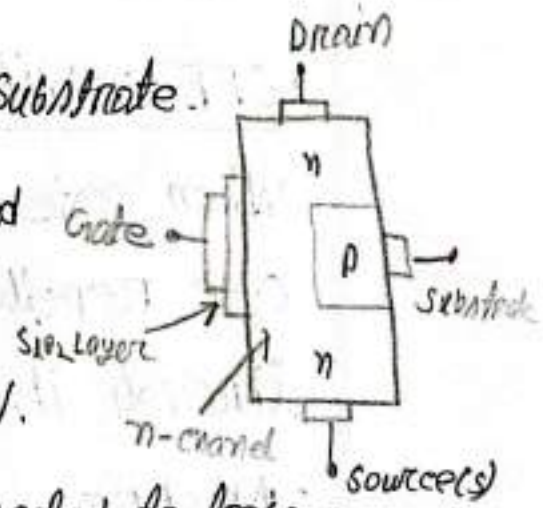
TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

- In the construction of MOSFET, a lightly doped substrate, is diffused with a heavily doped region.
- Depending upon the substrate used, they are called as P-type and N-type MOSFETs.

N channel MOSFET

- N channel bar
- P type material is doped as a substrate.
- ON condition: When gate is connected to logic 1, electrons flow from source to drain & NMOS turns ON.



- OFF condition: When gate is connected to logic 0, electrons are repelled & can't flow from source and drain, current becomes zero.
- MOSFET is in cutoff region.

"GOOD LUCK"

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

P channel MOSFET

- P channel bar.
- N type meterial is doped as a substrate.
- **ON condition:** When gate is connected to logic 0, holes attracted towards gate & then flow drain.
- MOSFET is in saturation region.
- **OFF condition:**

When gate is connected to logic 1, high voltage holes are repelled by gate hence current can't flow through it.

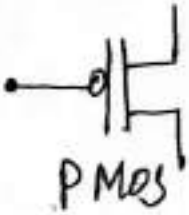
- MOSFET is in cutoff region.

Remember

Switching condition:

Input	N channel	P channel
$V_{in} = 0$	OFF	ON
$V_{in} = 1$	ON	OFF

NMos



PMos

TOPIC NAME : CMOS

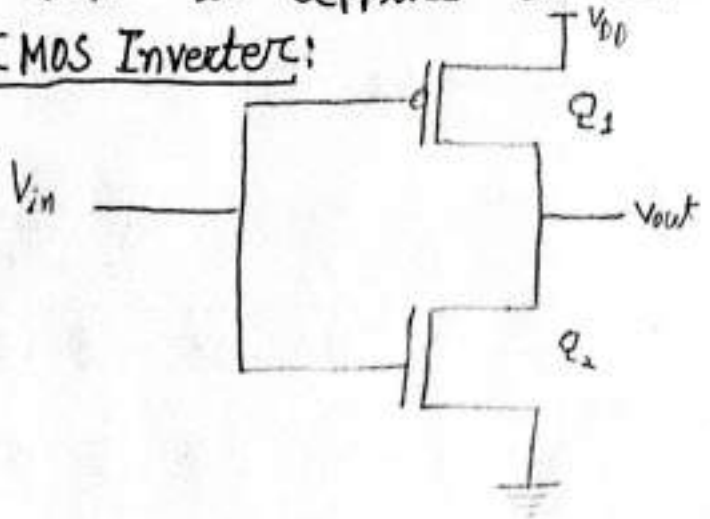
DAY :

TIME : DATE : / /

CMOS Logic

- P channel and N channel MOSFET are connected in series
- output is taken from common point of Drain.
- Input is applied at common point of Gate.

CMOS Inverter:



V_{in}	Q_1	Q_2	V_{out}
0	on	off	1
1	off	on	0

Q_1 = PMOS (on = 0)

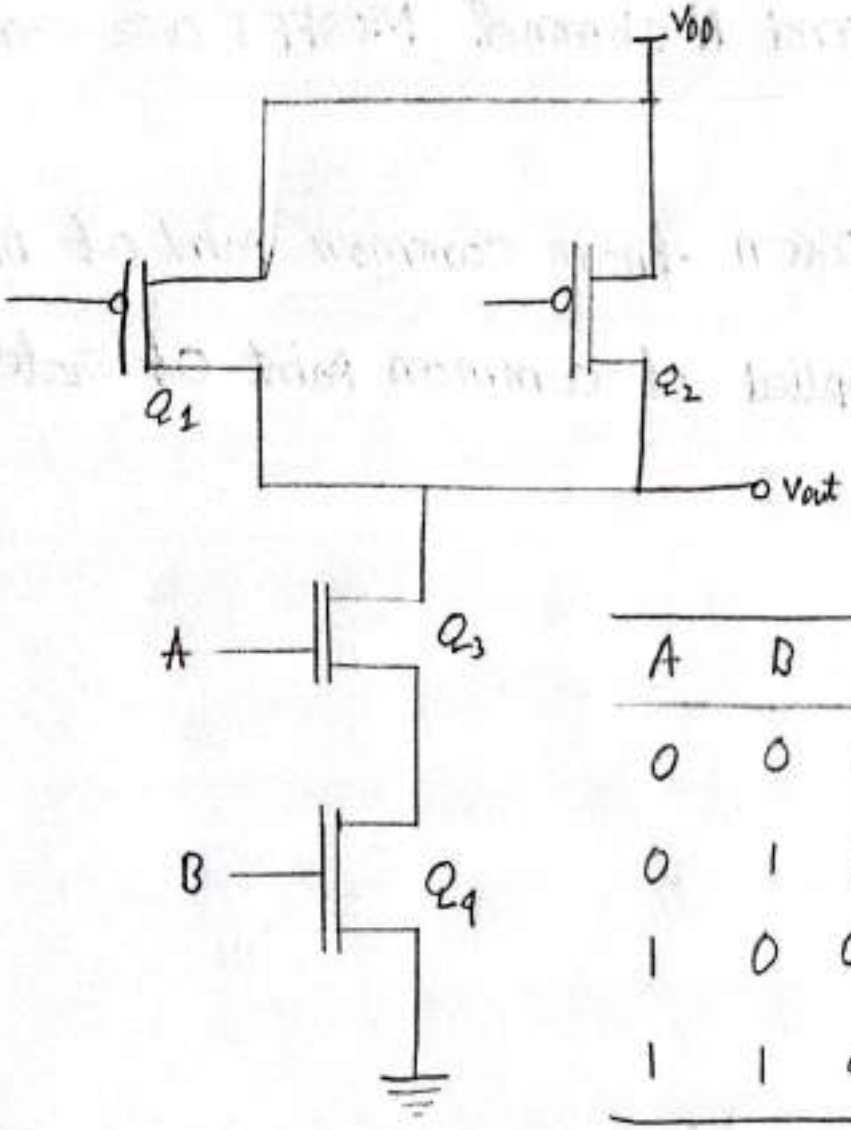
Q_2 = NMOS (on = 1)

CMOS NAND

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

CMOS NAND



A	B	Q_1	Q_2	Q_3	Q_4	Vout
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	1

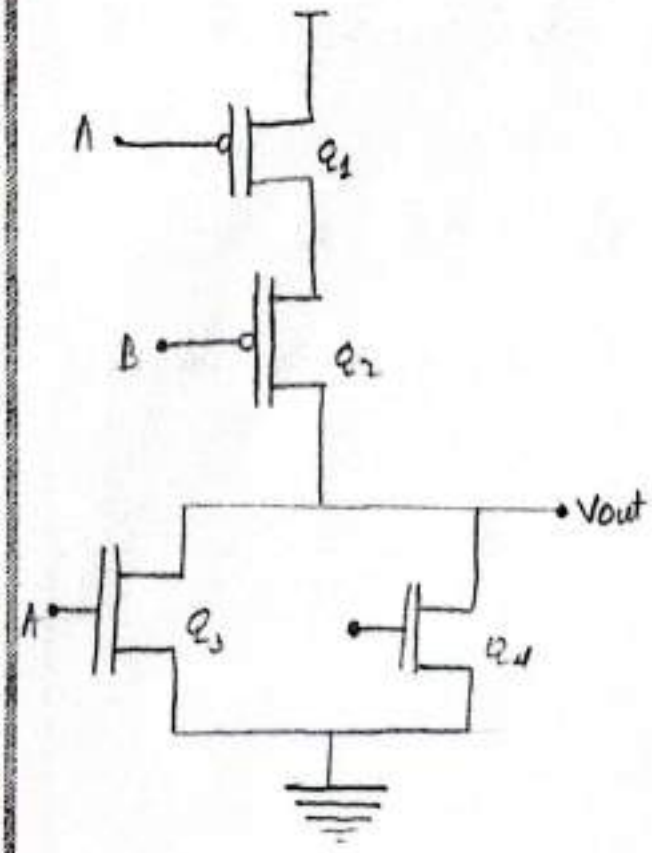
Q_1, Q_2 = PMOS (on = 0)

Q_3, Q_4 = NMOS (on = 1)

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

CMOS NOR



A	B	Q_1	Q_2	Q_3	Q_4	V_{out}
0	0	on	on	off	off	1
0	1	on	off	off	on	0
1	0	off	on	on	off	0
1	1	off	off	on	on	0

Q_1, Q_2 = PMOS (on = 0)

Q_3, Q_4 = NMOS (on = 1)

GOOD LUCK™

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF BARISAL

Final Examination

Course Title: Digital Logic Design

Course Code: CSE-2103

2nd Year 1st Semester

Session: 2020-21 (Admission Session 2019-20)

Time: 3 hour **Marks:** 60

Answer any five Questions from the followings.

1. a) Implement the Boolean function [3]

$$F = xy + x'y' + y'z$$

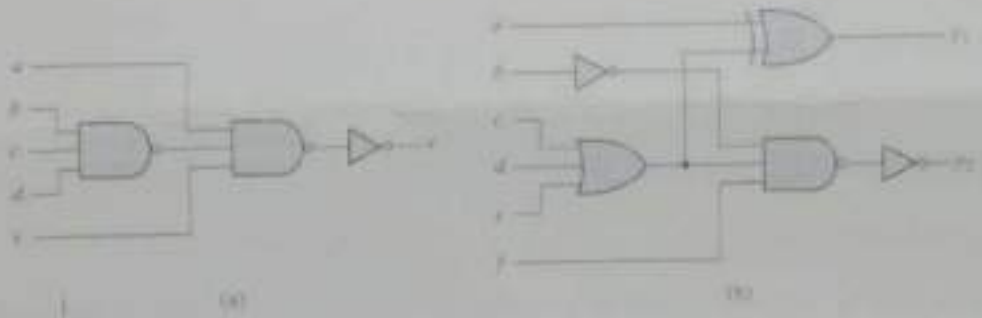
i. With AND, OR, and inverter gates,

ii. With NOR and inverter gates,

iii. With NAND and inverter gates.

b) Show that a positive logic NAND gate is a negative logic NOR gate and vice versa. [3]

c) Write Boolean expressions and construct the truth tables describing the outputs of the circuits described by the following logic diagrams: [6]



2. a) Express the following function as a sum of minterms and as a product of maxrerms: [2]

$$F(A,B,C,D) = B'D + A'D + BD$$

b) Simplify the following Boolean functions, using *Karnaugh* maps: [6]

$$F(w,x,y,z) = \sum(2,3,12,13,14,15)$$

$$F(A,B,C,D) = \sum(0,2,4,5,6,7,8,10,13,15)$$

c) Implement the following Boolean function F, together with the don't-care conditions d, using no more than two NOR gates: [4]

$$F(A,B,C,D) = \sum(2,4,6,10,12)$$

$$d(A,B,C,D) = \sum(0,8,9,13)$$

Assume that both the normal and complement inputs are available.

3. a) Explain how a NAND/NOR latch store a bit. Mention their limitations. [3]

b) What is the limitation of S-R flip-flop? Explain how the limitation can be resolved. [4]

c) Define race-around condition in J-K flip-flop. How can you overcome the problem? Explain with an appropriate figure and waveforms. [5]

4. a) What is the limitation of a parallel adder? Using 74LS83 ICs, draw a 16 bit parallel Adder. [3]

b) Explain how 2's complement system can facilitate arithmetic operations in digital computing. [3]

c) Draw a parallel adder/subtractor using 2's-complement system and explain its operations. [6]

5. a) Draw a Mod-14 and a Decade ripple counter with their state transition diagram. [4]

b) Show how to wire the 74293 IC as a MOD-6 counter. [2]

c) Explain the operation of a Synchronous up/down MOD-8 counter with an appropriate diagram. [6]

6. a) Write short notes on the followings: [5]

I) Fan-Out

II) Noise Immunity

III) Propagation delay

IV) Characteristics of TTL logic gate.

b) Design and explain the working principle of a TTL NOR gate. [5]

c) A certain TTL IC output is rated at $I_{OH}(max) = 800\ \mu A$ and $I_{OL} = 48$ mA. Express the IC's fan-out in terms of unit loads. [consider 1UL = 40 μA in the HIGH state and 1.6 mA in the LOW state] [2]

7. a) Design a 1 of 8 decoder using logic gates. [4]

b) Use 74LS138 ICs to design a 1 of 32 decoder. [5]

c) Draw the internal diagram of a 8:1 encoder. [3]

8. a) What is Multiplexer? Design an 8-input MUX and explain its operation. [4]

b) How can you use a 74138 IC as a DEMUX? Explain with diagram. [4]

c) Draw the logic diagram for the 7442 BCD to decimal decoder. [4]

6. a) Write short notes on the followings: [5]

I) Fan-Out

II) Noise immunity

III) Propagation delay

IV) Characteristics of TTL logic gate.

b) Design and expiain the working principie of a TTL NOR gate. [5]

c) A certain TTL IC output is rated at $I_{OH}(max) = 800\ \mu A$ and $I_{OL} = 48$ mA. Express the IC's fan-out in terms of unit loads. [consider 1UL = 40 μA in the HIGH state and 1.6 mA in the LOW state] [2]

I) Fan-Out

- **Definition**: Fan-out is the maximum number of inputs that a digital logic gate output can drive without degradation in performance.
- **Explanation**: It represents the load capacity of a gate; higher fan-out means it can connect to more subsequent gates.
- **Importance**: Essential for ensuring signal integrity across connected components in a circuit.

ii) Noise Immunity

- **Definition**: Noise immunity is a measure of a circuit's ability to tolerate external electrical noise without malfunctioning.
- **Explanation**: High noise immunity ensures that signals remain unaffected by minor fluctuations or interference in voltage.
- **Importance**: Critical for reliable performance in environments with electrical interference, as it prevents data corruption and errors.

iii) Propagation Delay

- **Definition**: Propagation delay is the time taken for a signal to pass from the input to the output of a gate.
- **Explanation**: It measures the speed of a gate and affects the overall speed of digital circuits.
- **Importance**: Lower propagation delays allow for faster data processing in digital systems.

IV) Characteristics of TTL Logic Gate

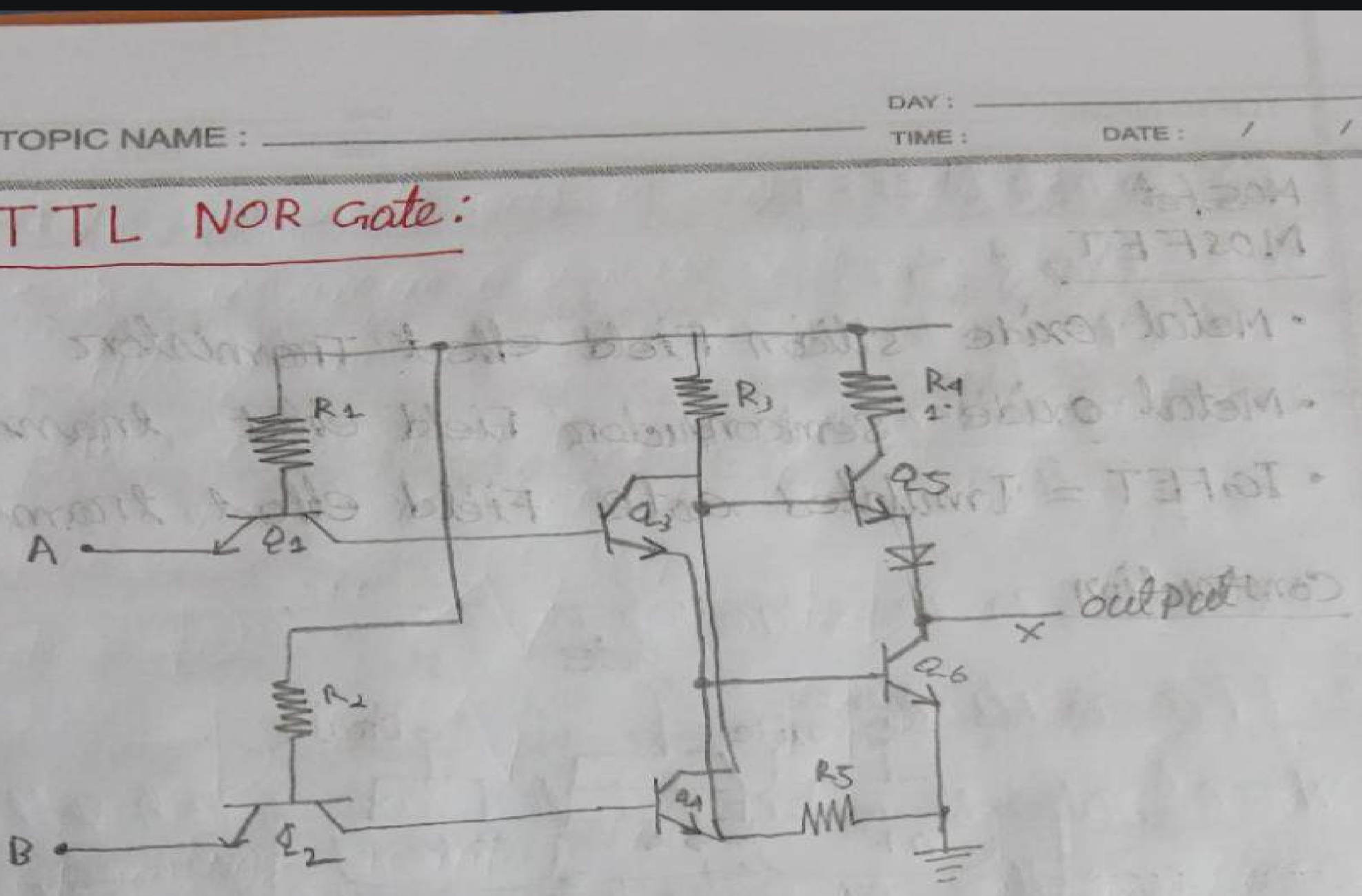
- **Speed**: TTL (Transistor-Transistor Logic) gates are relatively fast, with moderate propagation delay.
- **Power Consumption**: TTL circuits consume more power than some other logic families, like CMOS, but less than older logic types.
- **Noise Immunity**: TTL has moderate noise immunity, making it suitable for most general-purpose applications.
- **Fan-Out**: TTL logic gates usually have a fan-out of 10, meaning they can drive up to 10 standard TTL inputs.
- **Voltage Levels**: TTL operates typically at a 5V power supply with a distinct logic 0 (0V) and logic 1 (~5V).

b)

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

TTL NOR Gate:



Q_5 = ON = output = High (1)

Q_6 = ON = output = Low (0)

A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Y
0	0	ON	ON	OFF	OFF	ON	OFF	1
0	1	ON	OFF	OFF	ON	OFF	ON	0
1	0	OFF	ON	ON	OFF	OFF	ON	0
1	1	OFF	OFF	ON	ON	OFF	ON	0

A TTL NOR gate outputs a low (0) when any input is high (1) and outputs a high (1) only when all inputs are low (0).

Working Principle:

1. The gate uses a multi-emitter transistor for the inputs. If any input is high, this transistor conducts and keeps the output low (0).
2. When all inputs are low, no current flows in the input transistor, allowing the output to go high (1).

This structure provides the NOR logic, where the output is only high if all inputs are low.

c) Given data

$I_{OH(max)} = 800\,\mu A$

$I_{OL(max)} = 48\,mA$

In the high state, 1 UL = 40 μA

" " Low " , 1 UL = 1.6 mA

$$\text{Fan-out}_{(high)} = \frac{I_{OH}}{1\,UL} = \frac{800}{40} = 20\,UL$$

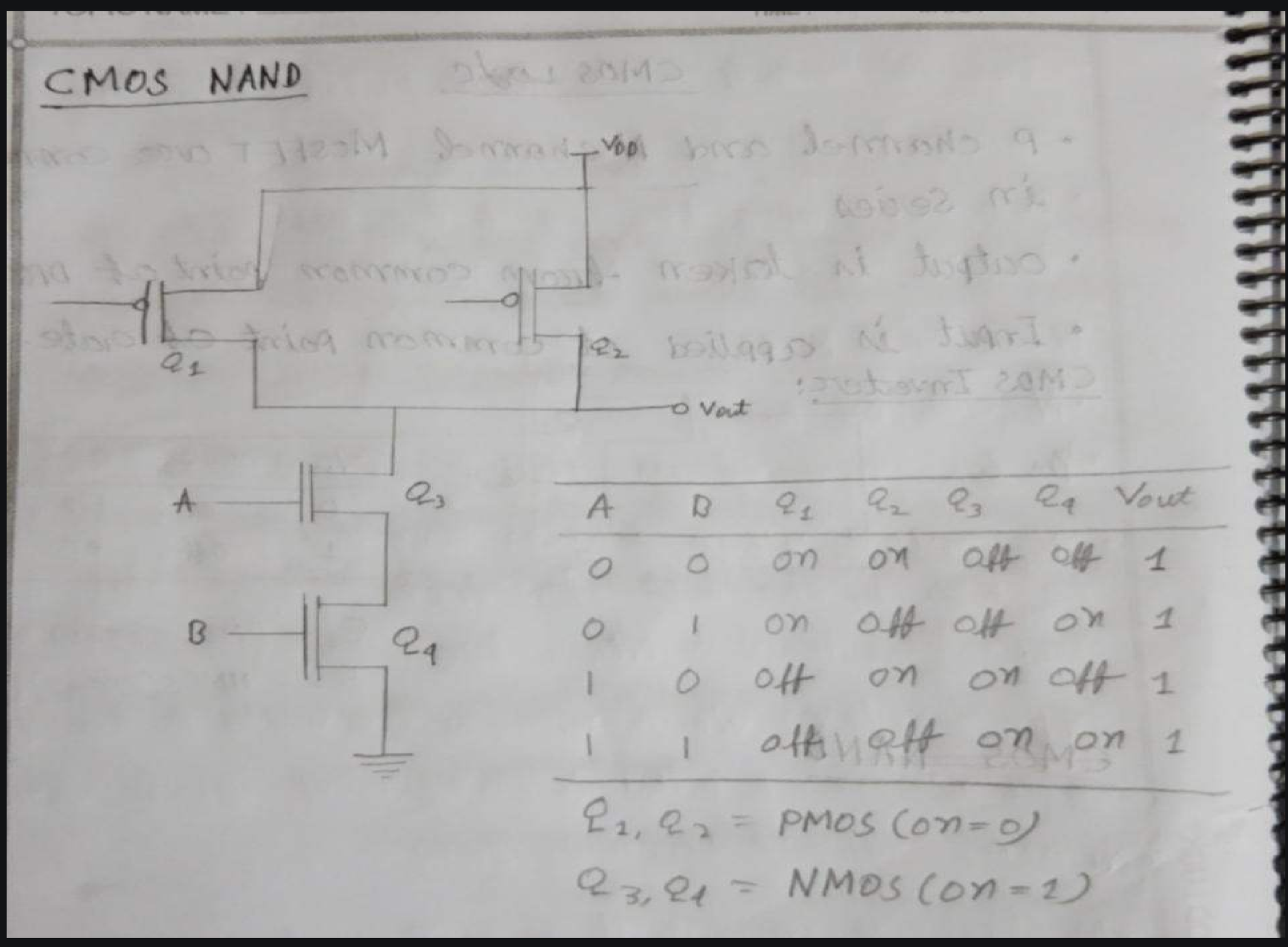
$$\text{Fan-out}_{(low)} = \frac{I_{OL}}{1\,UL} = \frac{48}{1.6} = 30\,UL$$

6. a) Explain the working principle of CMOS NAND and NOR gates.

   b) Draw the circuit diagrams of TTL NAND and NOR gates.

   c) Compare various characteristics of TTL and CMOS logic gate.

a)



A	B	Q_1	Q_2	Q_3	Q_4	Vout
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	1

Q_1, Q_2 = PMOS (on = 0)

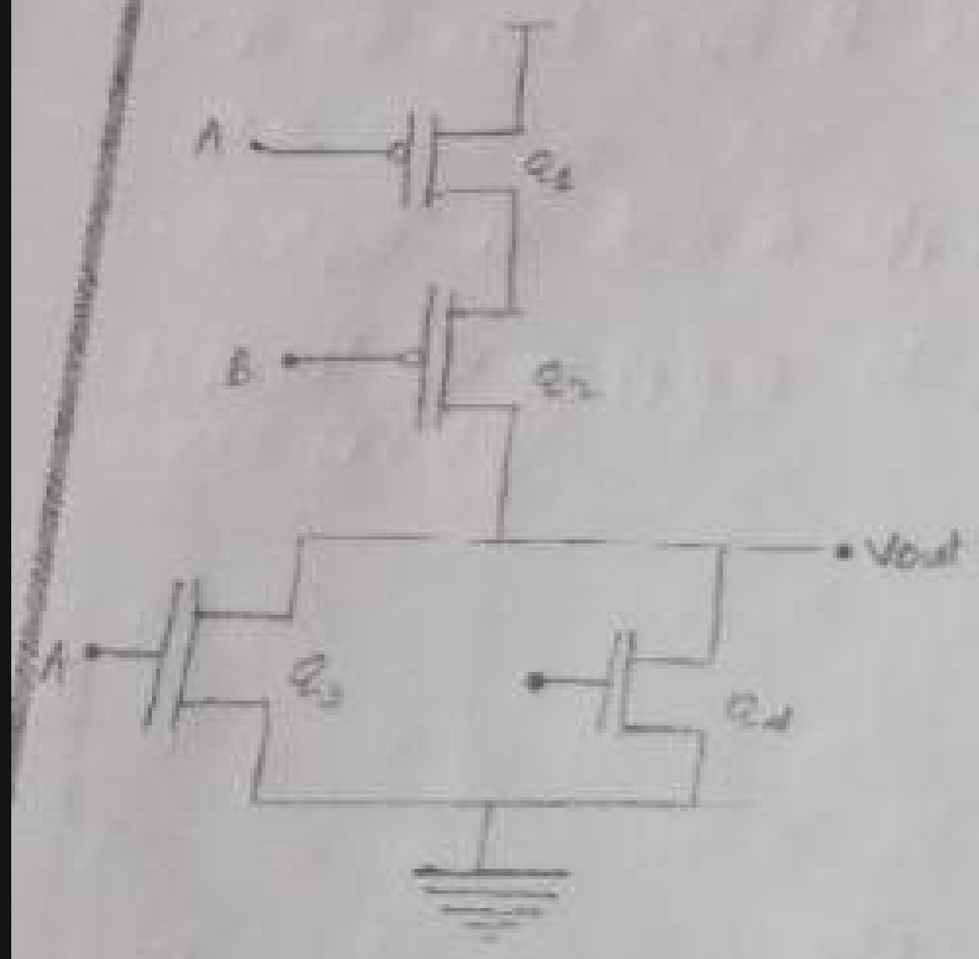
Q_3, Q_4 = NMOS (on = 1)

Working Principle:

- **When both inputs (A and B) are high (logic 1)**:
  - Both nMOS transistors conduct, creating a direct path to ground, pulling the output to low (logic 0).
  - Both pMOS transistors are off, so no current flows from V_DD to the output.
- **When either or both inputs are low (logic 0)**:
  - At least one pMOS transistor is on, connecting the output to V_DD, so the output becomes high (logic 1).
  - At least one nMOS transistor is off, breaking the path to ground.

TOPIC NAME : DAY TIME DATE

CMOS NOR



A	B	Q_1	Q_2	Q_3	Q_4	V_{out}
0	0	on	on	off	off	1
0	1	on	off	off	on	0
1	0	off	on	on	off	0
1	1	off	off	on	on	0

Q_1, Q_2 - PMOS (on = 0)

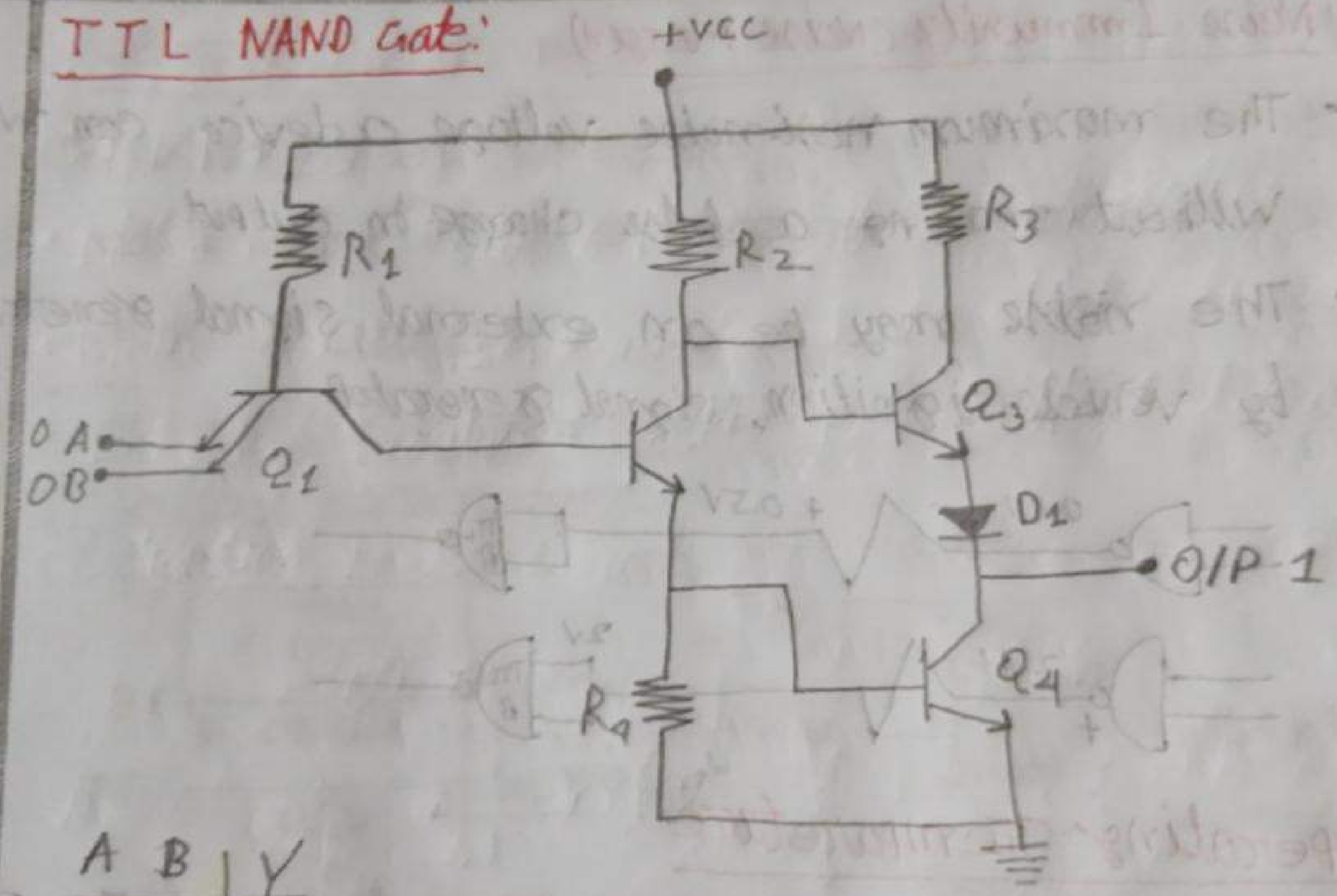
Q_3, Q_4 = NMOS (on = 1)

Working Principle:

- **When both inputs (A and B) are low (logic 0)**:
  - Both pMOS transistors conduct, creating a direct path from V_DD to the output, pulling it to high (logic 1).
  - Both nMOS transistors are off, so no current flows to ground.
- **When either or both inputs are high (logic 1)**:
  - At least one nMOS transistor conducts, creating a path to ground, so the output is pulled low (logic 0).
  - At least one pMOS transistor is off, breaking the path to V_DD.

b)

TTL NAND Gate:

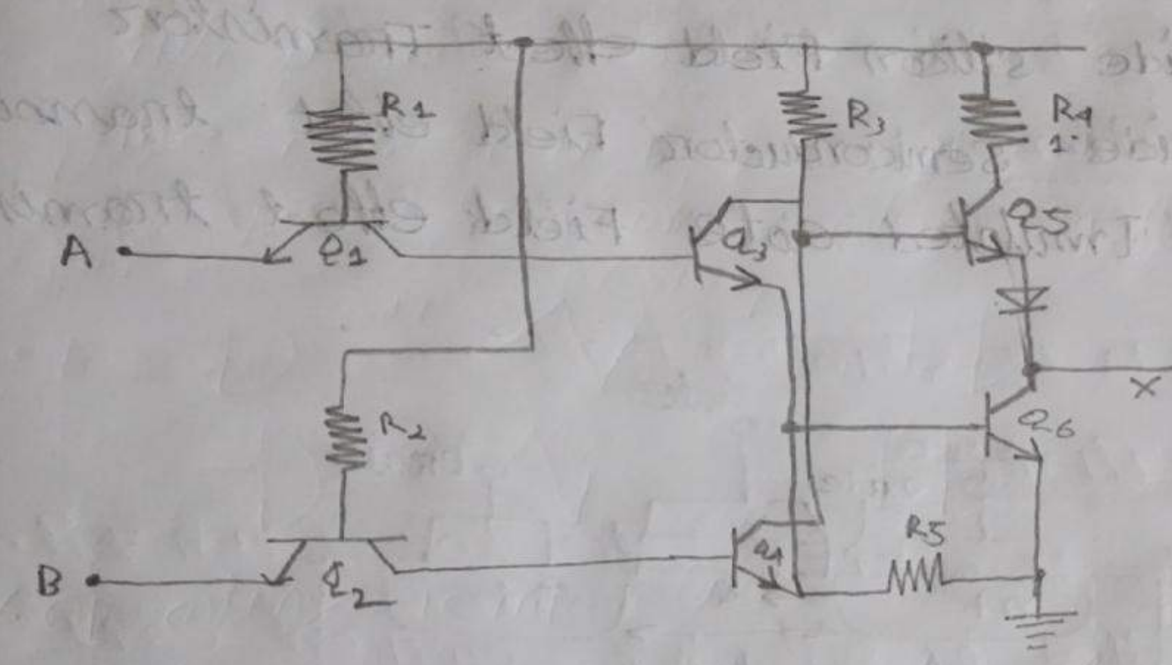


A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

TTL NOR Gate:



Q_5 = ON = output = High (1)

Q_6 = ON = output = Low (0)

A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Y
0	0	ON	ON	OFF	OFF	ON	OFF	1
0	1	ON	OFF	OFF	ON	OFF	ON	0
1	0	OFF	ON	ON	OFF	OFF	ON	0
1	1	OFF	OFF	ON	ON	OFF	ON	0

GOOD LUCK™

এটা Apply করা হেতে পারে NAND গেট মতো বসে

c)

TTL (Transistor-Transistor Logic) and CMOS (Complementary Metal-Oxide-Semiconductor) logic gates differ mainly in power usage, speed, and noise resistance. TTL has higher power consumption and faster switching but limited noise immunity and fan-out. CMOS uses less power, has higher noise immunity, and can work over a wider voltage range, making it ideal for battery-powered and complex digital circuits.

Characteristic	TTL	CMOS
Power Consumption	High	Low
Switching Speed	Fast	Initially slower but now fast
Fan-Out Capability	Limited (~10)	High
Noise Immunity	Lower	Higher
Input Impedance	Low	Very High
Supply Voltage Range	Typically 5V	3V-15V (varies)
Cost	Higher for high density	Lower, scalable
Temperature Stability	Moderate	Better due to low power

↳ প্রশ্নগুলো mandatory না সে বাড়ী মনে থাকে লেখবি

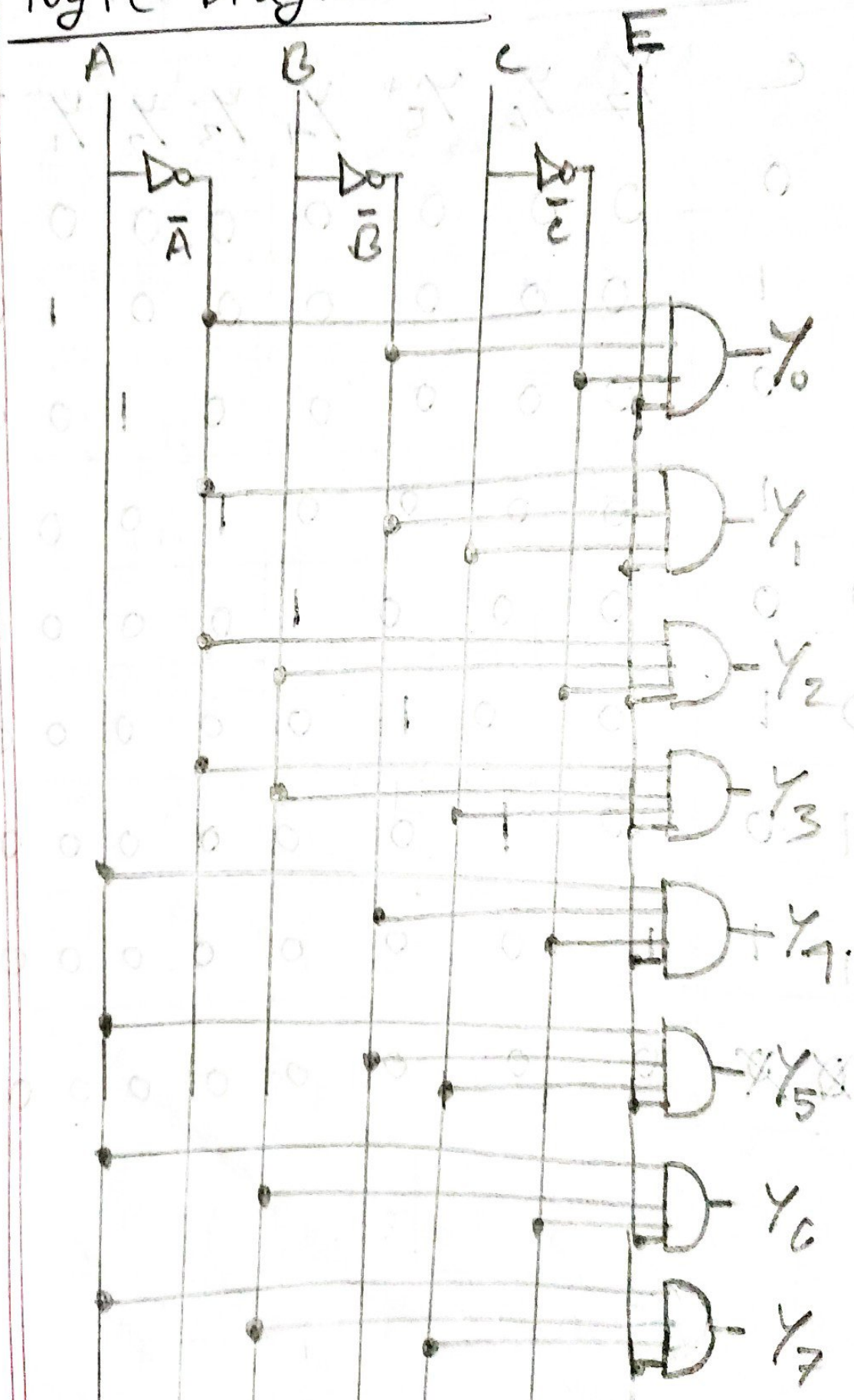
DLD

20-21

Truth table: 7(a)

E	A	B	C	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	
1	0	0	0	0	0	0	0	0	0	0	1	$Y_0 = \bar{A}\bar{B}\bar{C}$
1	0	0	1	0	0	0	0	0	0	1	0	$Y_1 = \bar{A}\bar{B}C$
1	0	1	0	0	0	0	0	0	1	0	0	$Y_2 = \bar{A}B\bar{C}$
1	0	1	1	0	0	0	0	1	0	0	0	$Y_3 = \bar{A}BC$
1	1	0	0	0	0	0	1	0	0	0	0	$Y_4 = A\bar{B}\bar{C}$
1	1	0	1	0	0	1	0	0	0	0	0	$Y_5 = A\bar{B}C$
1	1	1	0	0	1	0	0	0	0	0	0	$Y_6 = AB\bar{C}$
1	1	1	1	1	0	0	0	0	0	0	0	$Y_7 = ABC$
0	X	X	X	0	0	0	0	0	0	0	0	

logic Diagram:



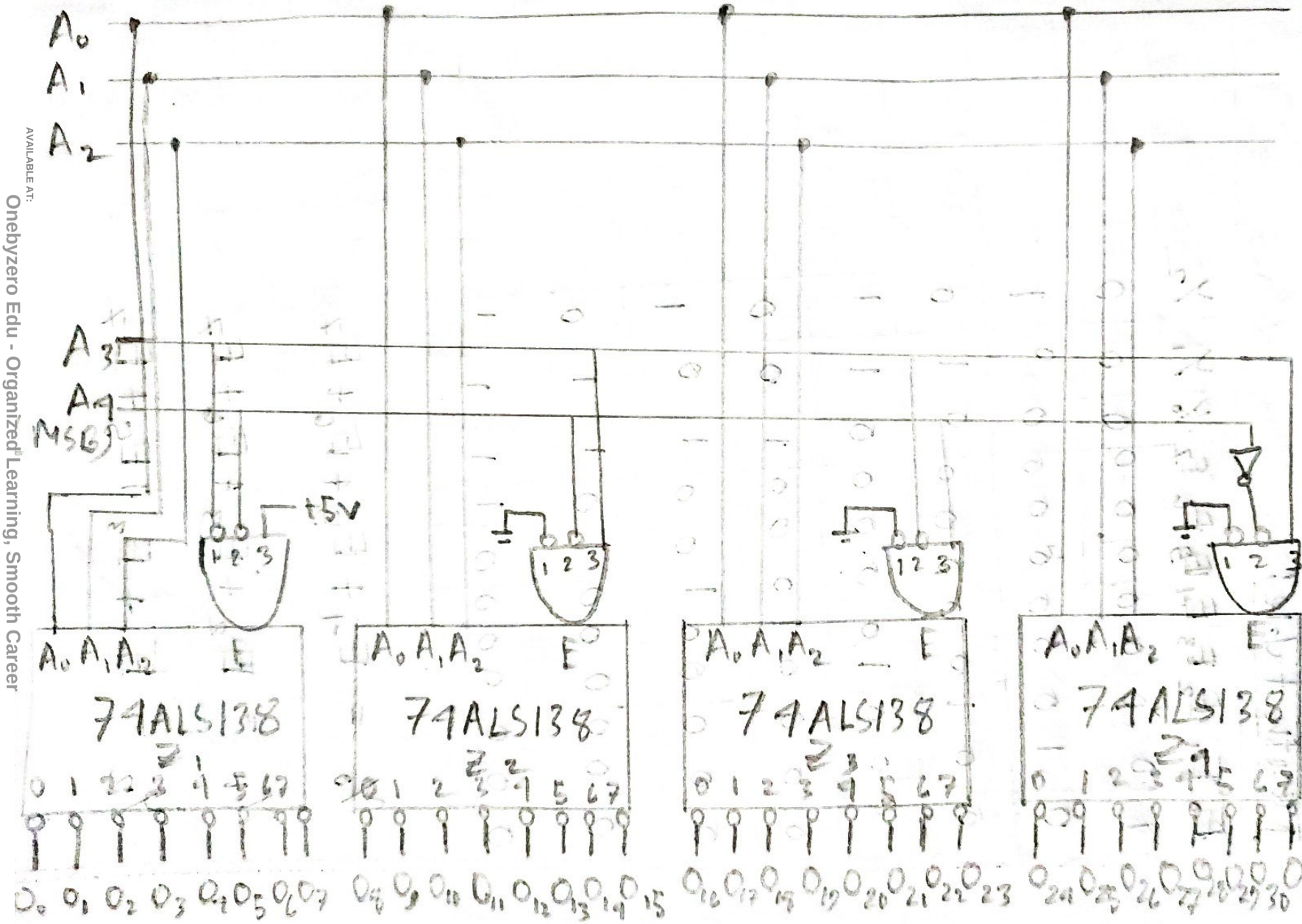


Fig: 1 of 32 Decoder.

7(c)

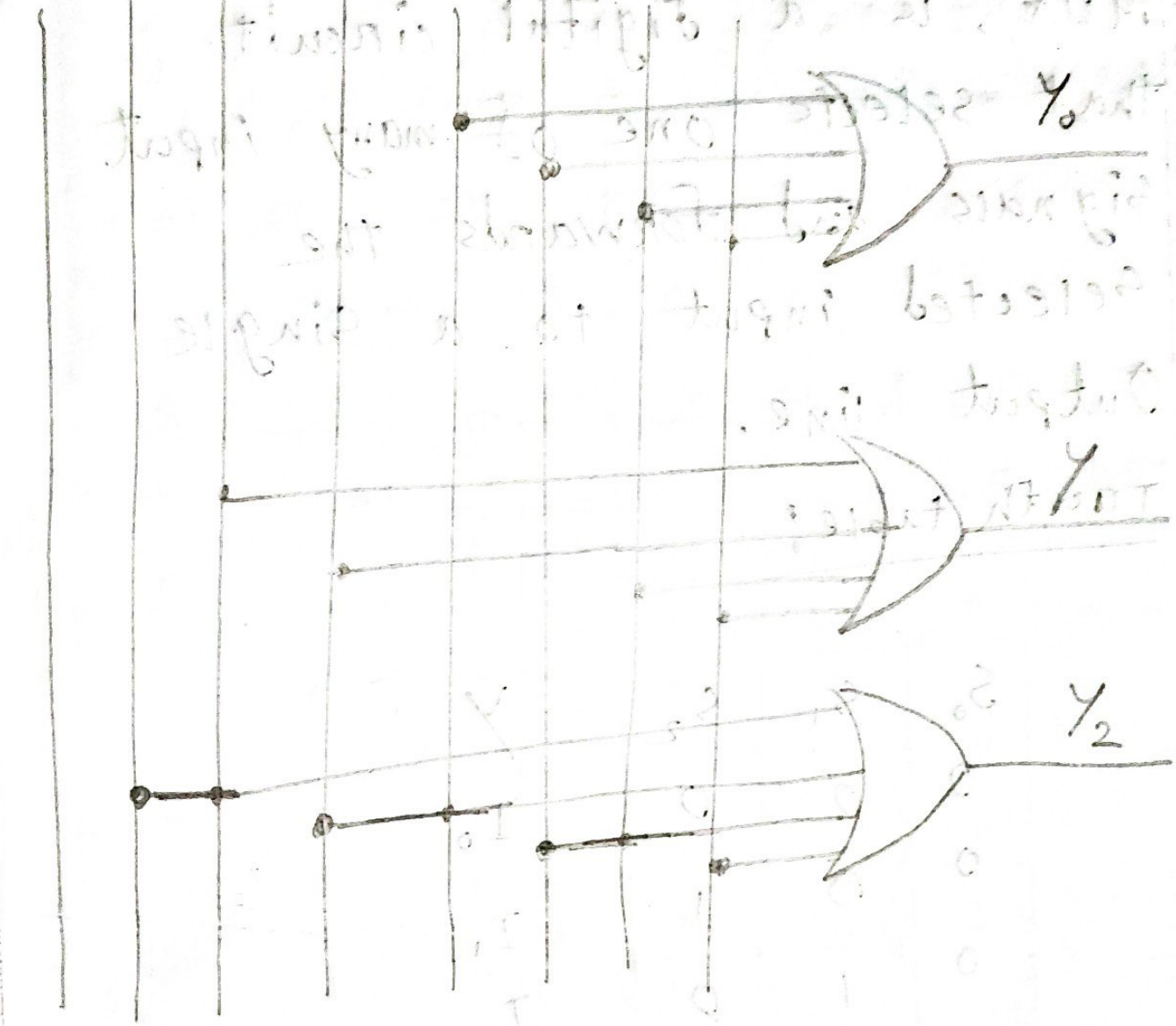
Truth table:

E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	Y_0	Y_1	Y_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$Y_0 = E_4 + E_5 + E_6 + E_7$$

$$Y_1 = E_2 + E_3 + E_6 + E_7$$

$$Y_2 = E_1 + E_3 + E_5 + E_7$$



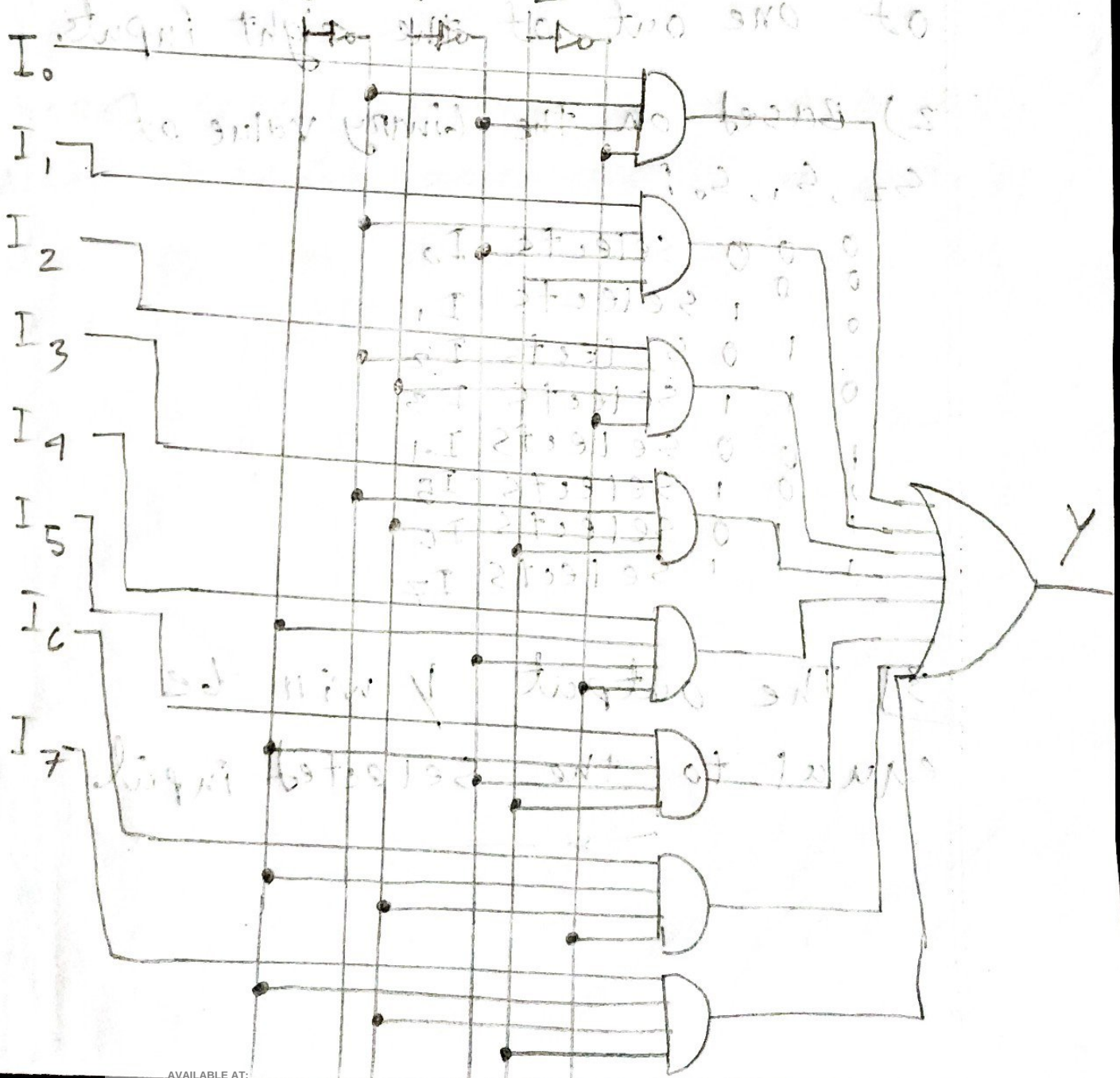
8(a)

A Multiplexer, also known as MUX, is a digital circuit that selects one of many input signals and forwards the selected input to a single output line.

Truth table:

S_0	S_1	S_2	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

$$Y = \bar{S}_0 \bar{S}_1 \bar{S}_2 I_0 + \bar{S}_0 \bar{S}_1 S_2 I_1 + \bar{S}_0 S_1 \bar{S}_2 I_2 + \bar{S}_0 S_1 S_2 I_3$$
$$+ S_0 \bar{S}_1 \bar{S}_2 I_4 + S_0 \bar{S}_1 S_2 I_5 + S_0 S_1 \bar{S}_2 I_6 + S_0 S_1 S_2 I_7$$



Operation of an 8 input Mux:

1) The three select lines S_0, S_1, S_2 create a 3-bit binary number, allowing the selection of one out of the eight inputs.

2) Based on the binary value of S_2, S_1, S_0:

0	0	0	selects I_0
0	0	1	selects I_1
0	1	0	selects I_2
0	1	1	selects I_3
1	0	0	selects I_4
1	0	1	selects I_5
1	1	0	selects I_6
1	1	1	selects I_7

3) The output Y will be equal to the selected input.

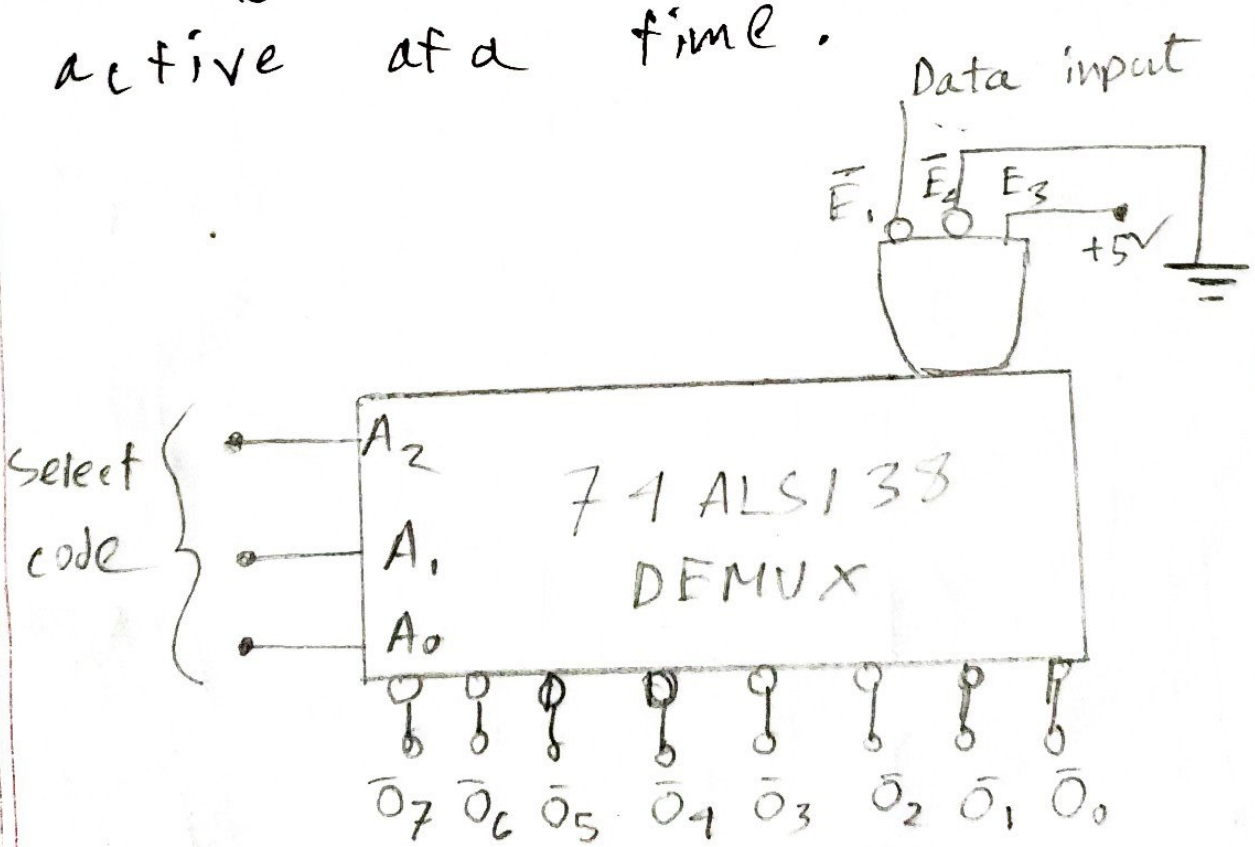
—∞—

8(b)

Using 74138 IC as a DEMUX.

Data input: The signal that needs to be demultiplexed is applied at on of the enable inputs, specifically E_3 in this case.

here, selection inputs are A_2, A_1, A_0. And outputs are O_0 to O_7. Only one of the output lines will be active at a time.

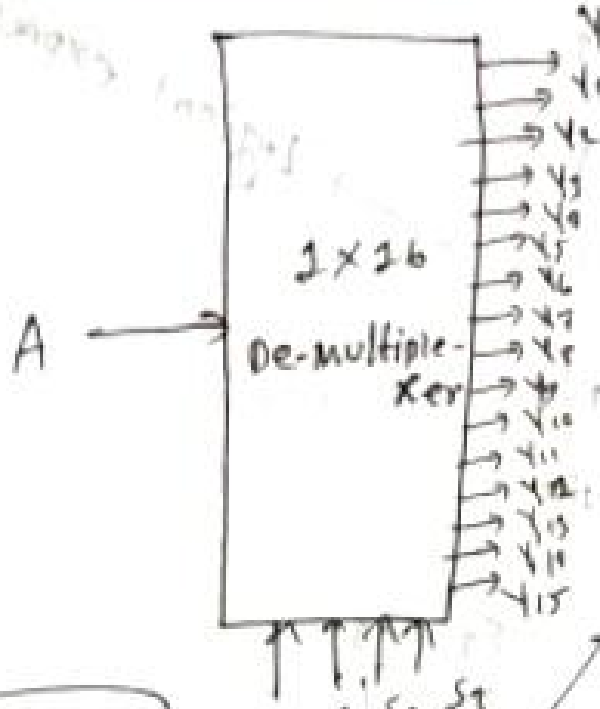


(35) With appropriate diagram explain the working principle of a 1 line to 16 line Demultiplexer.

A 1-to-16 De-multiplexer has one input (A), 4 selection lines (S0, S1, S2, S3), and 16 outputs (Y0 to Y15). Based on the combination of selection lines, the input is routed to one of the 16 outputs, while all others are inactive (low).

Working principle:

- The 4 selection lines create 16 combinations ($2^4 = 16$), each corresponding to one of the outputs being active.
- The input signal A is routed to the selected output based on the combination of S0 to S3.



Block Diagram

Truth table

Inputs

Outputs

S3	S2	S1	S0	Y15	Y14	Y13	Y12	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$Y_0 = A\bar{S}_0\bar{S}_1\bar{S}_2\bar{S}_3$

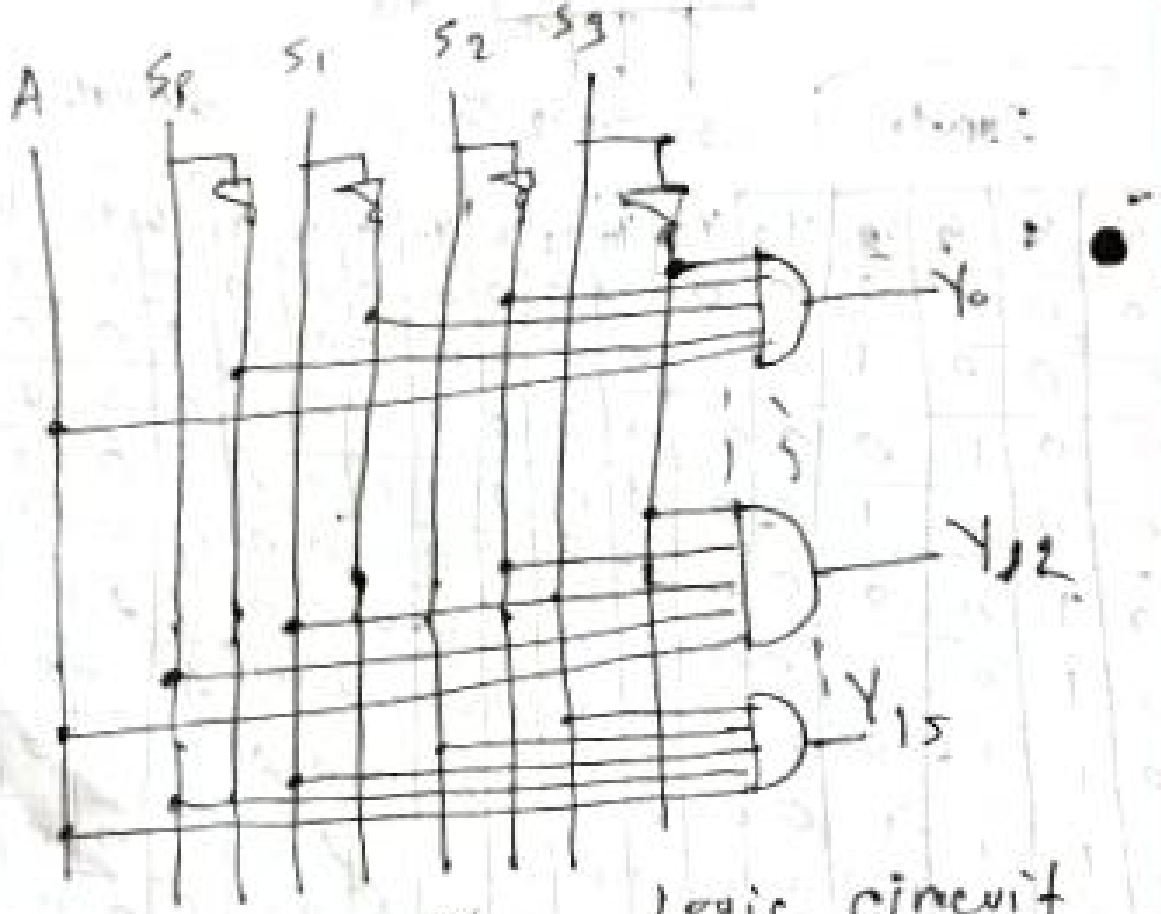
$Y_2 = A\bar{S}_0\bar{S}_1S_2\bar{S}_3$

$Y_8 = AS_0\bar{S}_1\bar{S}_2\bar{S}_3$

$Y_{12} = AS_0S_1\bar{S}_2\bar{S}_3$

$Y_{15} = AS_0S_1S_2S_3$

→ Logical expression



Logic circuit

(36) Draw the diagram and truth table of an octal to Binary encoder?

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

truth table

$A_0 = D_1 + D_3 + D_5 + D_7$

$A_1 = D_2 + D_3 + D_6 + D_7$

$A_2 = D_4 + D_5 + D_6 + D_7$

Logic Diagram