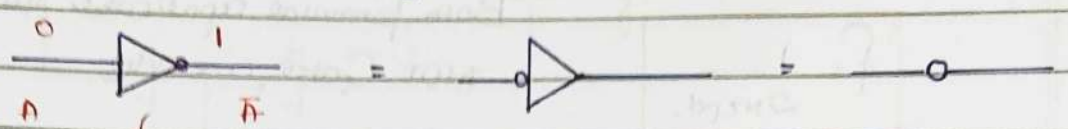
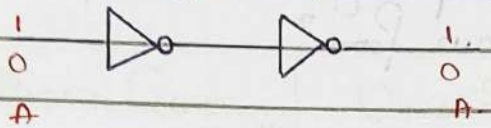
LOGIC GATES

1. NOT Gate [Inverter, Negation, Compliment logic].



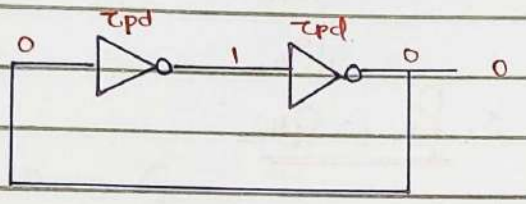
Propagation delay [τ_{pd}].

"Buffer" (for even)



No. of not gates are even - "Buffer".
No. of not gate odd - "Inverter".

[High.]



$2\tau_{pd}$ $2\tau_{pd}$ $2\tau_{pd}$ $2\tau_{pd}$

1 → 1 → 1 → 1 →

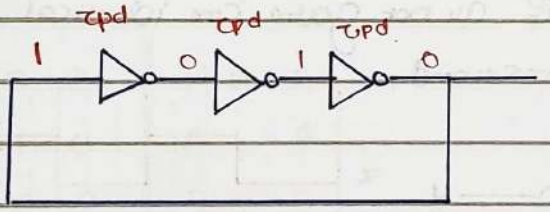
$2\tau_{pd}$ $2\tau_{pd}$ $2\tau_{pd}$

0 → 0 → 0 →

[Low]

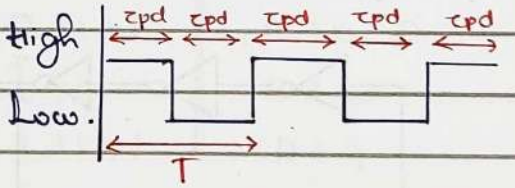
When even numbers of 'not' gate in Circuit (loop).

1. Basic memory element
2. Bistable multivibrator
3. DC Generator ($f = 0$ Hz).



τ_{pd} τ_{pd} τ_{pd} τ_{pd} τ_{pd}

1 → 0 → 1 → 0 → 1



When odd no. of NOT Gate:

1. Astable multivibrator
2. Square wave generator
3. Clock generator
4. Free running Circuit
5. Ring oscillator.

$$T = 2\tau_{pd}$$

$$f = \frac{1}{2\tau_{pd}}$$

$$f = \frac{1}{T}$$

$T = 2 \times N \times \tau_{pd}$. N = no. of not gate in loop.

$f = \frac{1}{2N\tau_{pd}}$

* Both formulas applicable when number of NOT Gate are odd.

Q1. For the circuit given below, all NOT Gate are identical to each other and having Propogation delay 10 ps. Find the frequency of generated wave form?

10^{-12}.

$N = 5$. $\tau_{pd} = 10 \times 10^{-12}$.

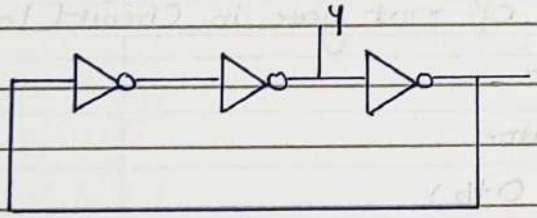
$f = \frac{1}{2N\tau_{pd}} = \frac{1}{2 \times 5 \times 10 \times 10^{-12}} = \frac{10^{12}}{10 \times 10} Hz$

(odd).

$= 10^{9} Hz$

$\therefore f = 1.0\ GHz$

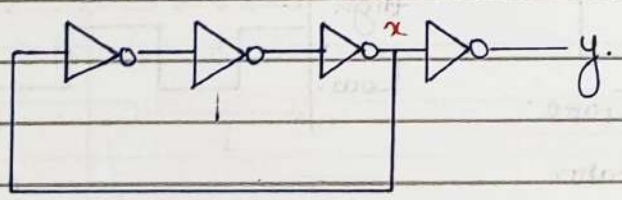
Q2 Circuit given below are called. Astable Multivibrator.

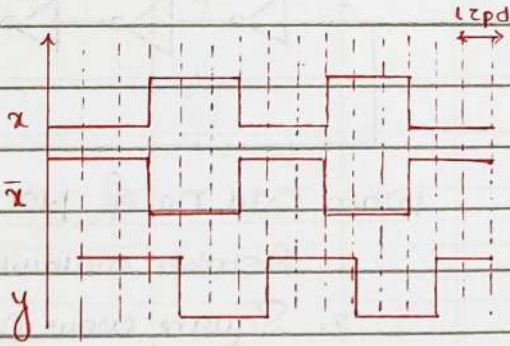


Note: always, for calculation of frequency consider the no. of NOT gate in loop.

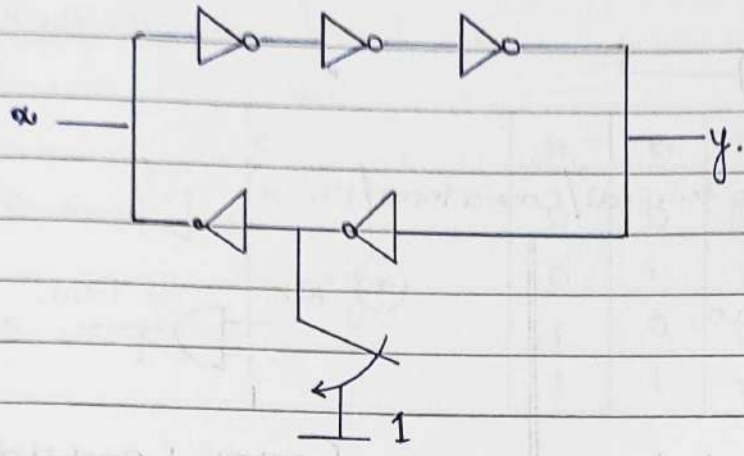
HW

Q3. Sketch the wave form of y if all not gate are identical and having Propogation delay of 1 microsecond.





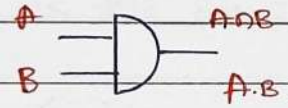
Q4. For the circuit given below x and y condition will be - x and y both toggle



Odd no of NOT gate in the loop
∴ x and y will toggle

Basic Gate	Universal Gate	Exclusive Gate
NOT	NAND	X-OR
AND	NOR	X-NOR
OR		

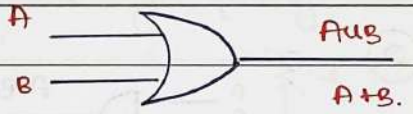
AND Gate



$A = \{1,2,3,4\}$ $B = \{2,3,4\}$
$A \cap B = \{2,3,4\}$

$0.0 = 0$	$A.A = A$
$0.1 = 0$	$A.1 = A$
$1.0 = 0$	$A.0 = 0$
$1.1 = 1$	$A.\bar{A} = 0$

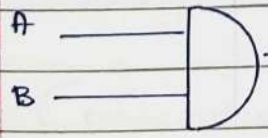
OR Gate



$A = \{1,2\}$ $B = \{3,4\}$
$A \cup B = \{1,2,3,4\}$

$0+0 = 0$	$A+A = A$
$0+1 = 1$	$A+0 = A$
$1+0 = 1$	$A+1 = 1$
$1+1 = 1$	$A+\bar{A} = 1$

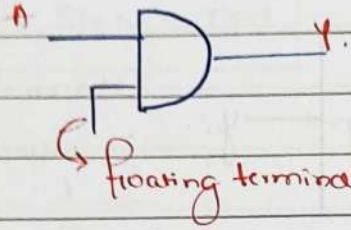
AND Gate [Intersection]

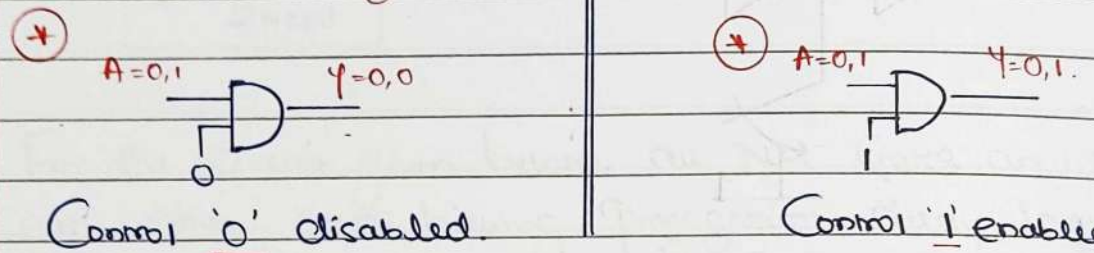


↳ Symbol.

A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

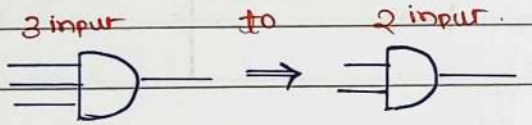
Truth Table





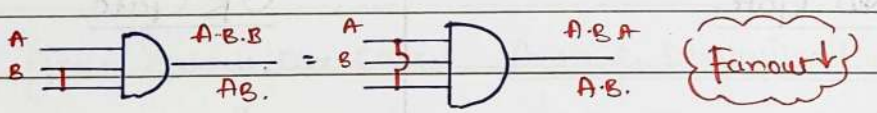
* Commutative law : $A \cdot B = B \cdot A$
* Associative law : $A \cdot B \cdot C = (A \cdot B) C$

} AND is both commutative and associative.

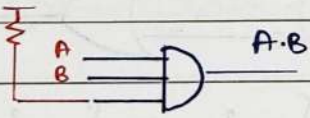


TTL = Transistor Transistor Logic.

1.

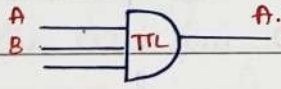


2.



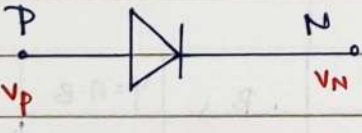
pull up arrangement. (Costly)

3.



(Noise margin ↓)

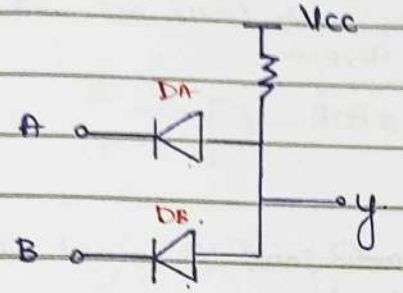
"Diode"



$V_P > V_N$ → Forward bias → Short ckt.

$V_P < V_N$ → Reverse bias → Open ckt.

Circuit diagram: [AND]



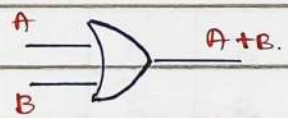
A	B	D_A	D_B	Y
0	0	FB	FB	0
0	1	FB	RB.	0
1	0	RB	FB.	0
1	1	Re	RB.	1

Note:

* Whenever logic are designed by TTL the floating terminal always works as high.
* Whenever logic are designed by ECL (Emitter Coupled logic) then floating terminal always works as low.
* Noise margin: Maximum noise added to the input which will not affect the output are called Noise margin.
* Fan out: The number of logic driven by the logic are called fan out.
* Fan in: Number of input of a logic are called fanin.
* ECL is the fastest logic among all the logic family.

OR Gate.

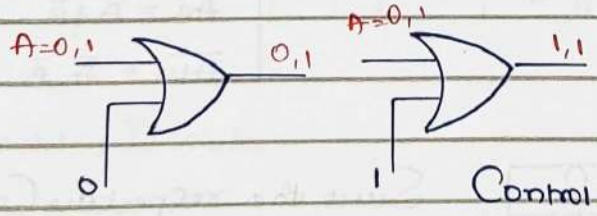
1. Symbol.



2. Truth Table.

A	B	Y
0	0	0
0	1	1
01	0	1
1	1	1

3. Enable/Disable.

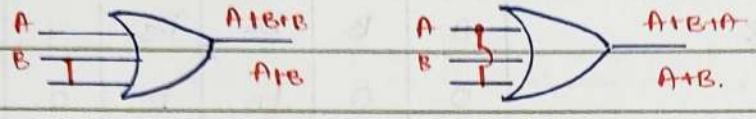


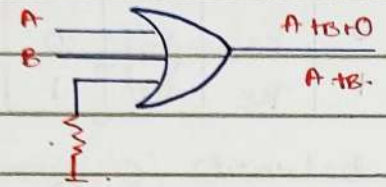
Control '0' enabled.

Control '1' disabled.

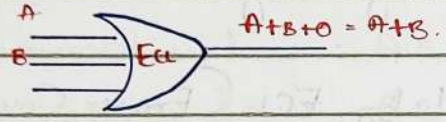
4. Commutative: $A+B = B+A$
5. Associative: $A+(B+C) = (A+B)+C$.

ECL - Emitter Coupled Logic.

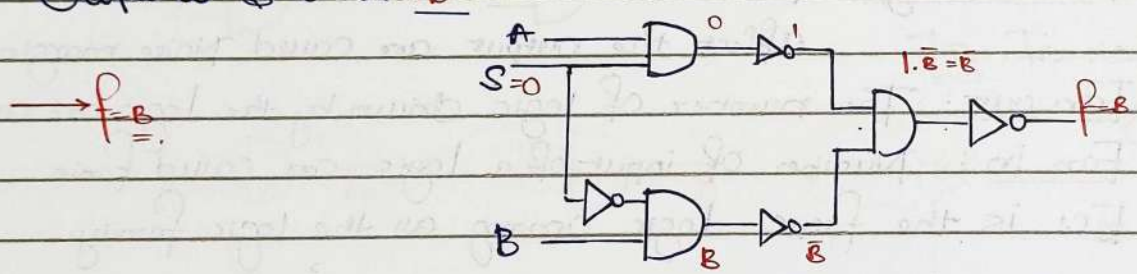
1. 

2. 

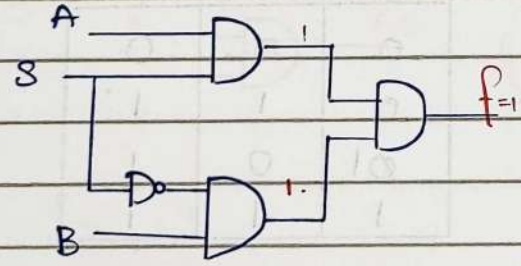
Pull down arrangement.

3. 

Q1. Output at S=0 is . B



Q2. f=1 for Combination of A, B, S.



[None of the options are correct.]

De Morgan's Law

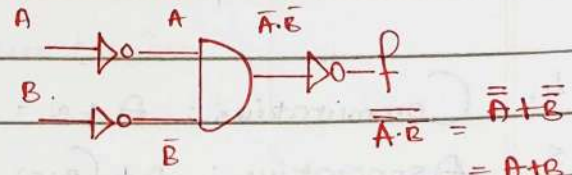
$\overline{AB} = \bar{A} + \bar{B}$

$\overline{A+B} = \bar{A} \cdot \bar{B}$.

Q3. For the given truth table.

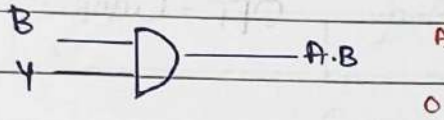
A	B	f
0	0	0
0	1	1
1	0	1
1	1	1.

Select the respective Combinational diagram

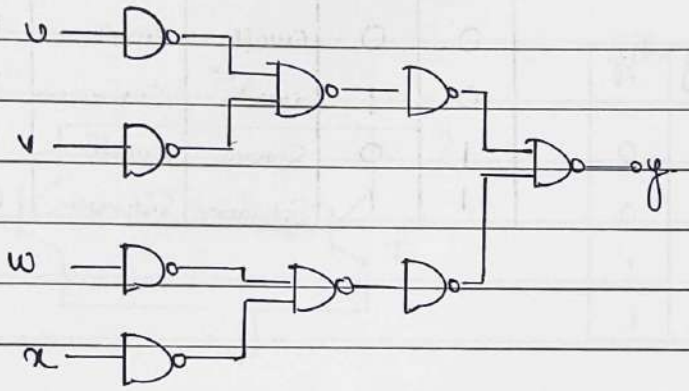


Date 9/6/23

HW Q4. A logical circuit is as shown below, which of the following circuit can be used to get the desired expression.

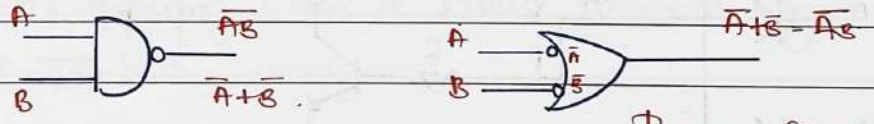


Q5. The logic circuit shown below, is equivalent to.



NAND Gate.

1. Symbol:

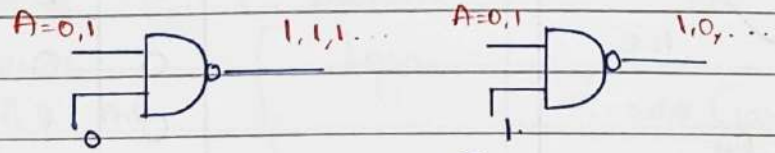


Bubbled OR = NAND

2. Truth Table.

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

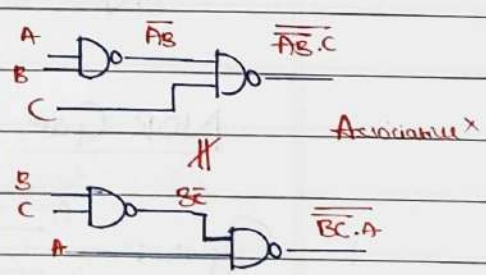
3. Enable / Disable.



Control '0' disabled. Control '1' enabled

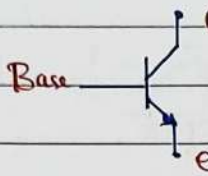
4. Commutative law. ✓

5. Associative law. ×



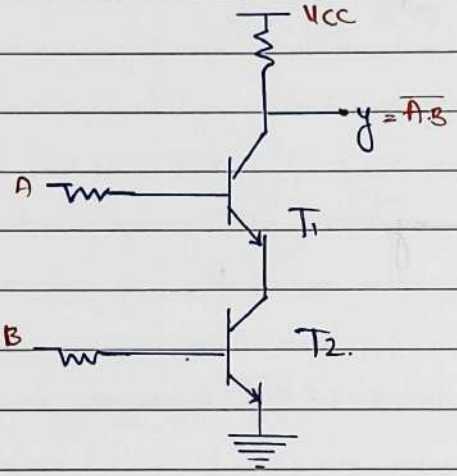
NAND follows commutative law but does not follow associative law.

Transistor



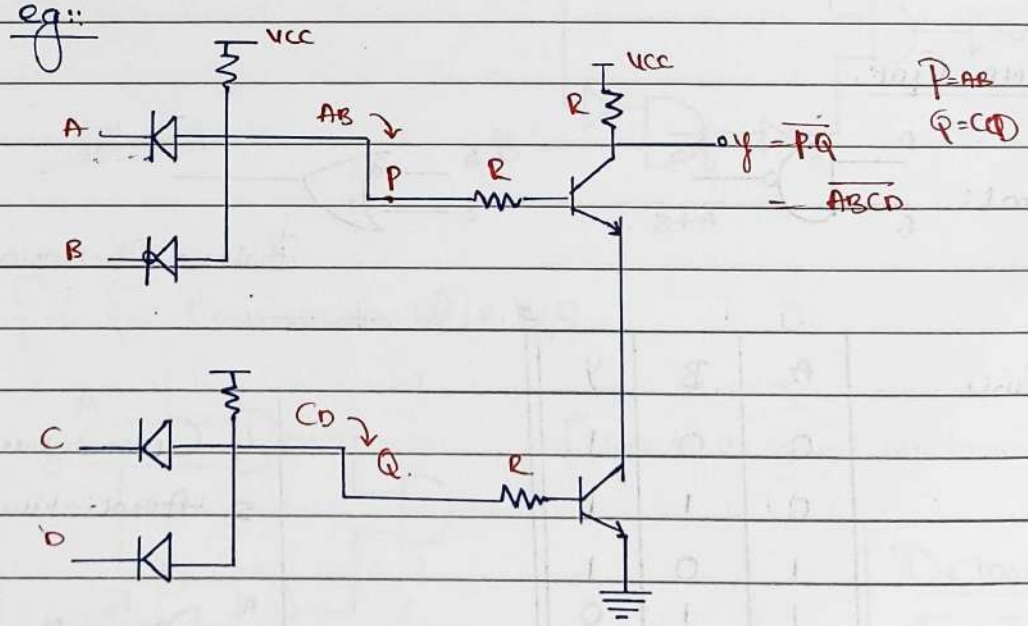
Input = 1 → ON = Saturation → Short Circuit.

Input = 0 → OFF = Cutoff → Open Circuit.



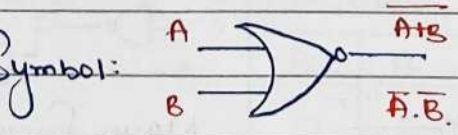
A	B	T_1	T_2	y
0	0	Cutoff	Cutoff	1
0	1	Cutoff	Saturation	1
1	0	Saturat.	Cutoff.	1
1	1	Saturation	Saturation	0

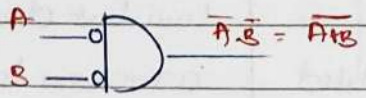
eg::



NOR Gate.

1. Symbol:



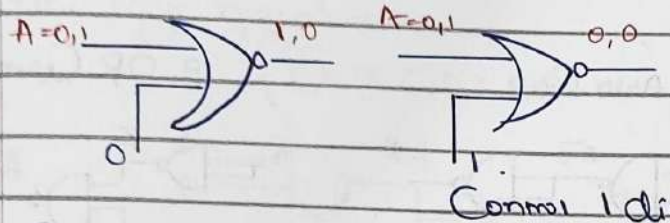


Bubbled AND = NOR

2. Truth Table:

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

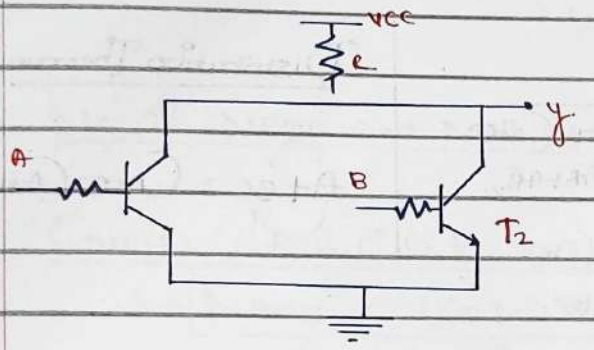
3. Enable / Disable



Control '0' enable

4. Commutative law ✓

5. Associative law. ×



A	B	T_1	T_2	y
0	0	Cutoff	Cutoff	1
0	1	Cutoff	Saturation	0
1	0	Saturation	Cutoff	0
1	1	Saturation	Saturation	0

Note: NAND, NOR are called "Universal logic".

Q1. Which of the following option is called universal logic?

Ans. Both NAND and NOR. (option given)

Q2. Which of the following is called Universal logic? [MCQ]

Ans NAND (if both are given choose NAND)

Q3. Which of the following Option(s) is/are called universal logic? [MSQ]

Ans. $(\overline{A+B})$, $(\overline{A \cdot B})$, $(A+\overline{B})$, $A \cdot \overline{B}$.

$\overline{A \cdot B}$

$\overline{A+B}$

$\overline{A}+B$

$A+\overline{B}$

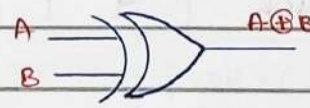
$\overline{A} \cdot B$, $A \cdot \overline{B}$

Multiplexer

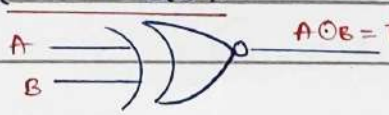
Decoder + OR logic

} Universal logic.

X-OR Gate:

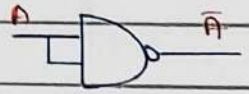


X-NOR Gate

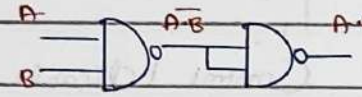


NAND as Universal logic

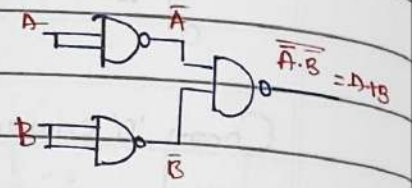
1. NOT Gate



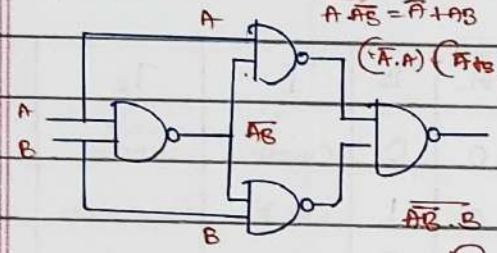
2. AND Gate



3. OR Gate



4. XOR Gate



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

$(\bar{A}\cdot A)(\bar{A}+B) = \bar{A}+B$

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

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

$A\bar{B}\cdot B$

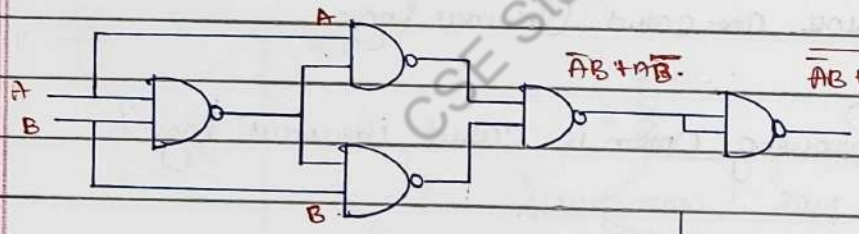
$A\bar{B}\cdot\bar{B}$ $(A+\bar{B})(\bar{B}+B)$

$= A+B$

Distributive Theorem

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

5. X-NOR Gate



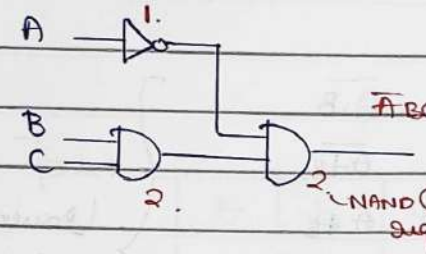
$\overline{A\bar{B}+\bar{A}B} = \bar{A}\bar{B}+AB$

$= A\odot B$

		NAND	NOR
Table of no. of NAND and NOR gate required to make the given Gates.	NOT	1	1
	AND	2	3
	OR	3	2
	X-OR	4	5
	X-NOR	5	4
	NAND	1	4
	NOR	4	1

Q1. Find the minimum no. of two input NAND Gate required to implement the logic given below.

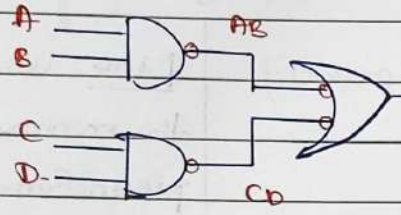
$f(A,B,C) = \bar{A}BC$



$\therefore 1+2+2 = 5$ required

Q2 Find the minimum no. of two input NAND Gate required to implement the logic given below.

$f(A, B, C, D) = AB + CD$



Create using AOI approach and make it with NAND by adding inverter ('o') to balance out!

∴ 3 NAND Gate required.

No. of NAND and NOR Gate Required:

Case (1) : A.B.$\bar{C}$.D.E....

→ NAND = $(2n-2)+k$ → NOR = $(3n-3)-k$

n = total no. of variable
k = no. of Compliment variables.

n = no. of variables
k = no. of Complement variables.

eg: $f = A.\bar{B}.C$

n = 3, k = 1.

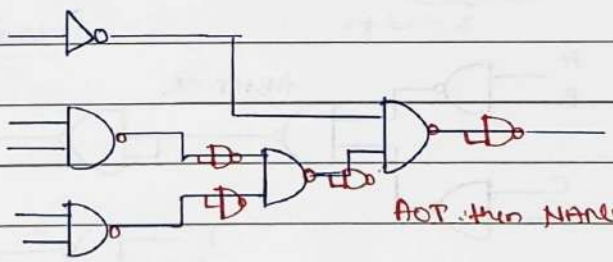
NAND = $(2n-2)+k$
= $(2\times3-2)+1 = 5$

NOR = $(3n-3)-k$
= $(3\times3-3)-1$
= $6-1 = 5$

eg: $f = \bar{A}BCDE$

n = 5, k = 1.

NAND = $(2\cdot5-2)+1$
= 9



Case (2) : A+B+$\bar{C}$+D....

NAND = $(3n-3)-k$

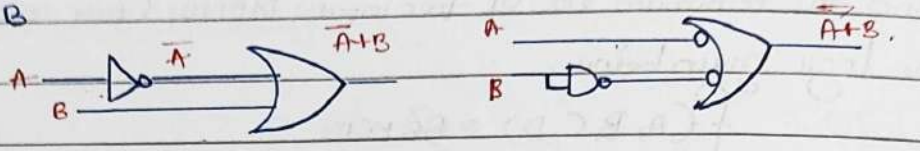
NOR = $(2n-2)+k$

eg: $f = \bar{A}+B$, n = 2 k = 1

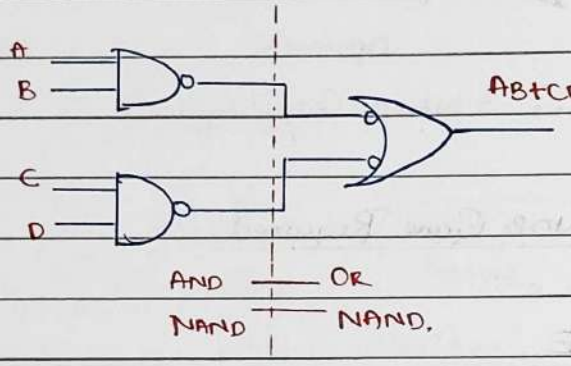
NAND = $(3n-3)-k$
= $(3\times2-3)-1 = 2$

NOR = $(2n-2)+k$
= $(2\times2-2)+1$
= 3

eg: $\bar{A}+B$

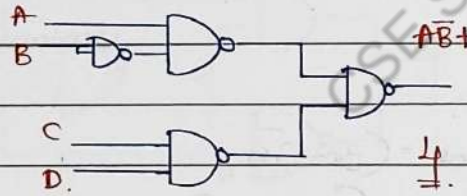


Case (3) $f = AB+CD$. [Sum of product]

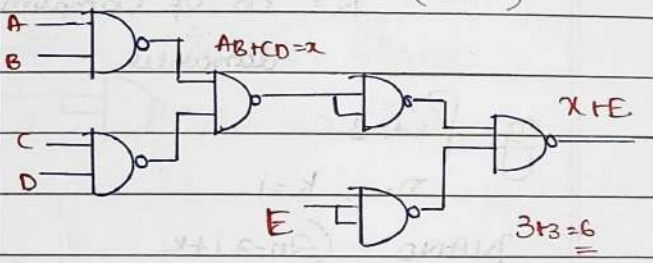


Note: Whenever in the problem the minimum number of NAND Gate are asked then write in SOP form and implement it by AOI and this is exactly equal to NAND NAND implementation.

Q. $f = A\bar{B}+CD$.

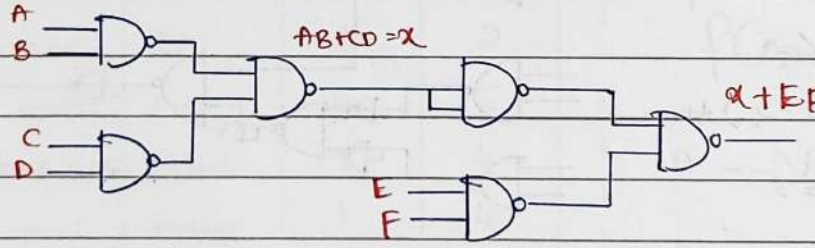


$f = AB+CD+E$. $f = x+E$ $(AB+CD)$

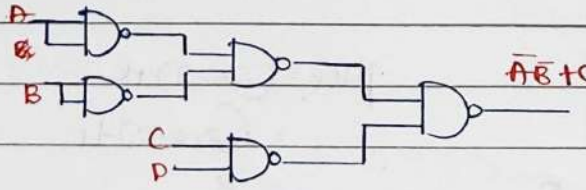


Q. $f = AB+CD+EF$

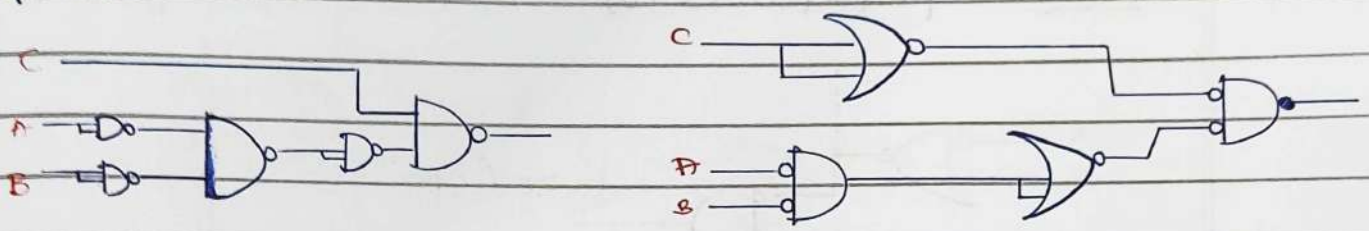
$x \rightarrow 3+3$



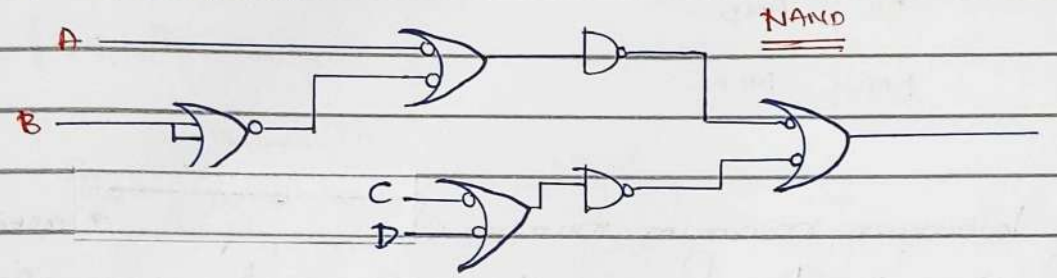
Q. $\bar{A}\bar{B}+CD$.



Q1 $f = \bar{A}\bar{B}C$.



Q.2 $f = \bar{A} + B + \bar{C} + \bar{D}$.

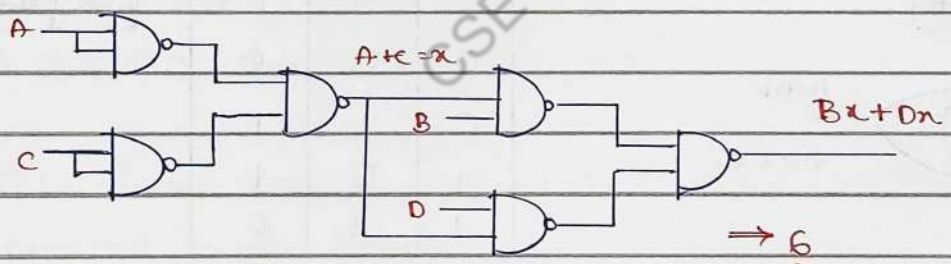


$(30-3)+x$
$= (3\times4-3)-3$
$= 6$

Q3. $f = AB + BC + CD + DA$

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

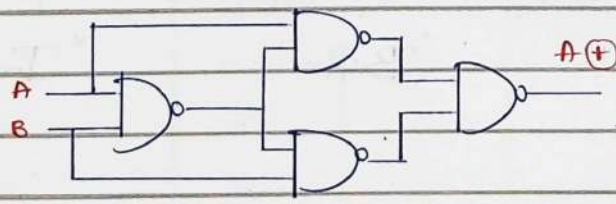
$f = Bx + Dx$, $A+C = x$.



$\Rightarrow 6$

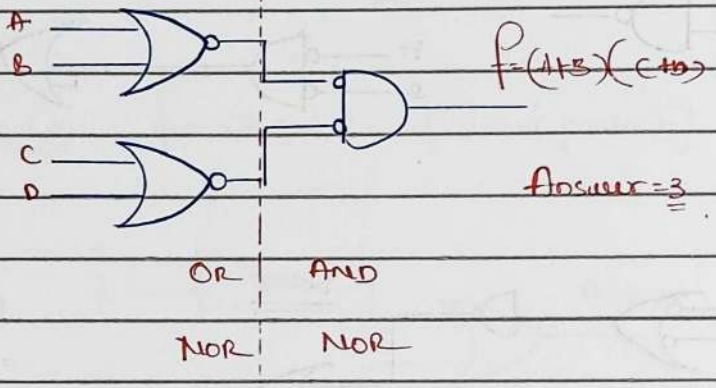
Case (4). Most Important.

Q1. $\bar{A}B + A\bar{B}$ ⟶ XOR.



Answer = 4.

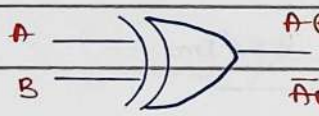
Case(5) $f = (A+B).(C+D)$ ⟶ Product of Sum form.



Answer = 3

Note: Whenever minimum number of NOR Gate are asked, write the function in POS form and implement by using AOI which is AND-OR Implementation which is exactly equal to NOR NOR implementation.

XOR Gate.



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

Truth Table

$A \oplus A = 0$

$A \oplus 0 = A$

$A \oplus 1 = \bar{A}$

$A \oplus \bar{A} = 1$

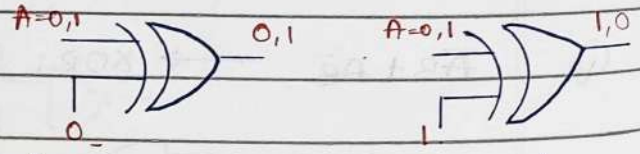
eg:: $A \oplus B = C$

$A \oplus C = B$

$B \oplus C = A$.

$A \oplus B \oplus C = 0$

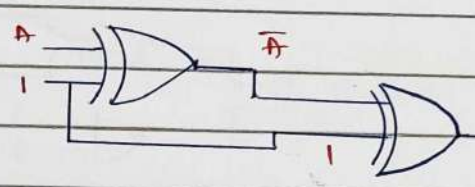
Enable/Disable



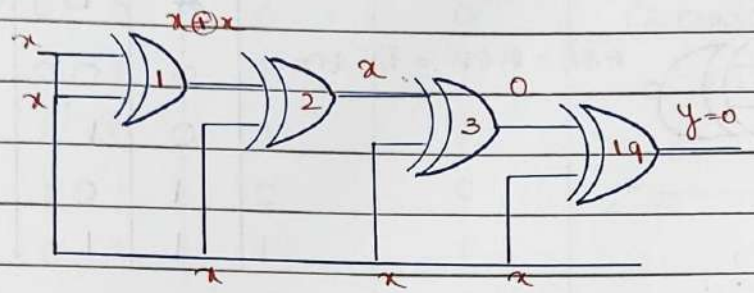
⟶ $A \oplus A = 0$ (for even)

$A \oplus A \oplus A = A$ (for odd)

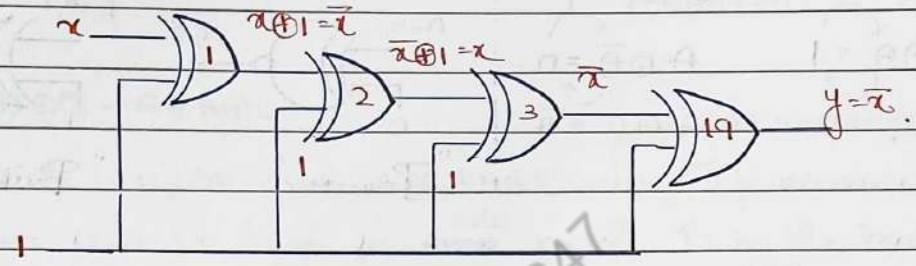
eg::



Q1. Find the output y.



Q2. Find the output y.



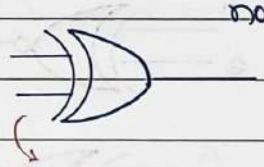
→ XOR Gate follows Commutative as well as associative law

$A \oplus B = B \oplus A$ → Commutative law

$A \oplus B \oplus C = (A \oplus B) \oplus C$ → associative law

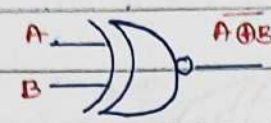
→ X-OR Gate output will be high when odd number of 1's are connected in the inputs.

ABC	A⊕B⊕C	(A⊕B)⊕C
000	0	0
001	1	1
010	1	1
011	0	0
100	1	1
101	0	0
110	0	0
111.	1.	1



X-NOR Gate:

Symbol.



A	B	A⊙B
0	0	1
0	1	0
1	0	0
1	1	1

Truth Table

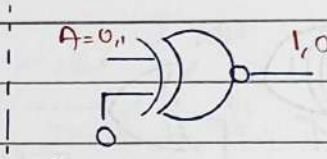
* $A = B \quad y = 1$

  $A = \bar{B} \quad y = 0$

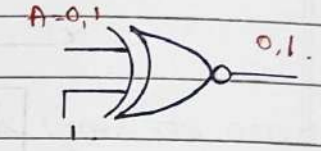
* It is also called as "even parity detector / Equivalence logic / Equal detector / Coincidence Logic".

* $A \odot A = 1$ $\quad A \odot \bar{A} = 0$

  $A \odot 1 = A$ $\quad A \odot 0 = \bar{A}$

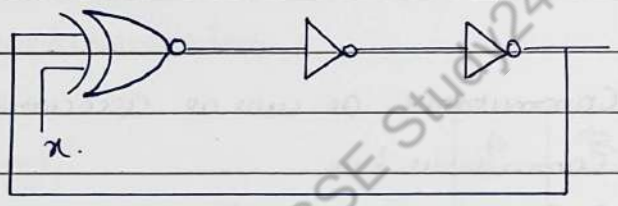


"Inverter".



"Buffer".

eg::



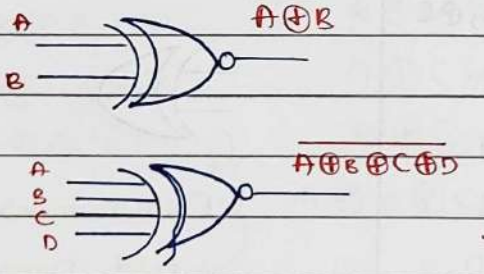
$x = 0$ - Astable multivibrator

$x = 1$ - Bistable multivibrator.

* X-NOR Gate follows commutative as well as associative law.

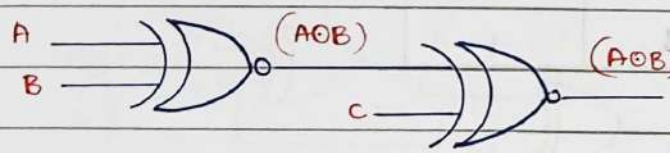
* For even number of variable and even number of inputs.

Case 1:



Output will be high for even no of inputs are high.

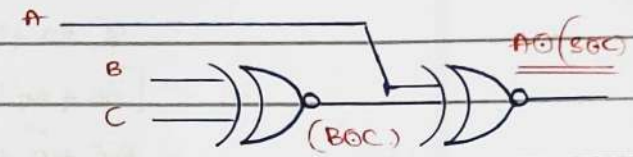
Case 2:



	A	B	C	(A⊙B)⊙C	A⊙(B⊙C)
1.	0	0	0	0	0
2.	0	0	1	1	1
3.	0	1	0	1	1
4.	0	1	1	0	0
5.	1	0	0	1	1
6.	1	0	1	0	0
7.	1	1	0	0	0
8.	1	1	1	1	1

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

↳ Associative Law.



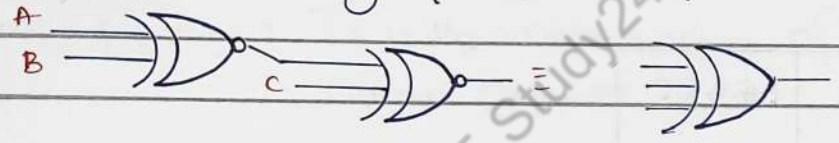
* $A \odot B = \overline{A \oplus B}$

$A \odot B \odot C \odot D = \overline{A \oplus B \oplus C \oplus D}$ } for even!

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

$(((A \odot B) \odot C) \odot D) \odot E = A \oplus B \oplus C \oplus D \oplus E$ } for odd variables

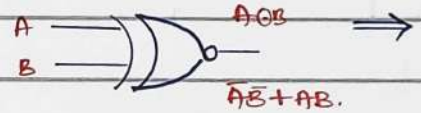
* Output will be high for odd no of 1's in the input. (for this combi.)

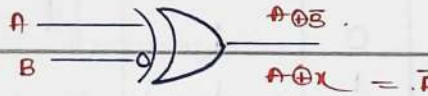


variable

* Output will be high for even no. of 1's in input

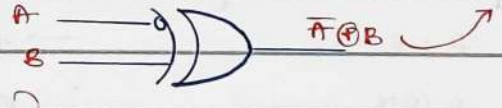
Magic





$\bar{A}B + A\bar{B}$ ← X-NOR

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



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

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

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

Same!

$A \odot B = \bar{A} \oplus B = A \oplus \bar{B} = \bar{A} \odot \bar{B} = \overline{A \oplus B}$

$A \oplus B = \bar{A} \odot B = A \odot \bar{B} = \bar{A} \oplus B = \overline{A \odot B}$

Q1 The boolean function given: $f(A,B) = A \oplus B \oplus AB$

Which statement is/are correct?

$f(A,B) = (A \oplus B) \oplus AB$

$= X \oplus AB$ $\quad A \oplus B = X = \bar{A}B + A\bar{B}$

$= \bar{X} \cdot AB + X\bar{A}\bar{B}$ $\quad \overline{A \oplus B} = \bar{X} = A \odot B = \bar{A}\bar{B} + AB$

$= [\bar{A}\bar{B} + AB] \cdot AB + [\bar{A}B + A\bar{B}] \cdot [\bar{A} + \bar{B}]$

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

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

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

$= A[B + \bar{B}] + \bar{A}B$

$= A \cdot 1 + \bar{A}B \Rightarrow A + \bar{A}B = (A + \bar{A})(A + B) = A + B$ (OR)

Ans: → It is a OR Gate

→ It requires 3 NAND Gate to implement the function.

A B	AB	A⊕B⊕C
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	1

$f = A + B$

Q2 Minimized expression will be $Y = A \oplus (A + B)$

$A + B = X$

$A \oplus X$

$\bar{A}X + A\bar{X}$

$\bar{A}[A + B] + A[\overline{A+B}]$

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

$= \bar{A}B$

→ Associative law does not follow by X-OR Gate and OR Gate.

$A \oplus (A + B)$

$(A \oplus A) \neq (A \oplus B)$

$0 \neq A \oplus B$

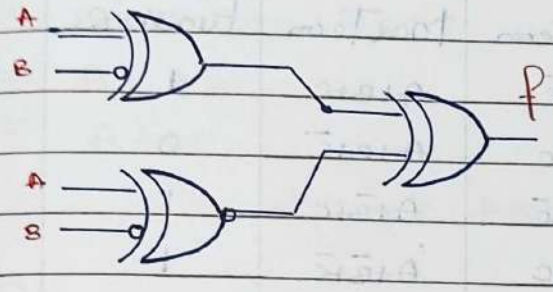
$A \oplus B$.

A B	A	(A+B) X	A⊕X
0 0	0	0	0
0 1	0	1	1
1 0	1	1	0
1 1	1	1	0

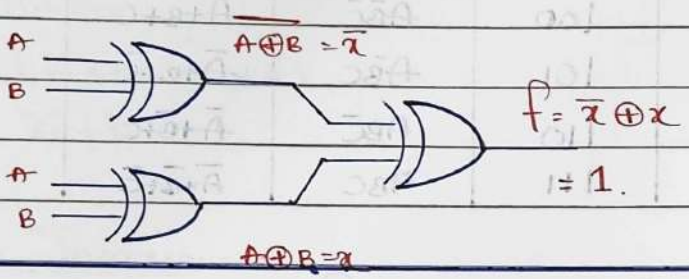
$A \oplus (A + B)$

$A \oplus X$.

eg::



f will be 1.



Laws of Boolean Algebra

Boolean function: It is the combination of inputs on which output depends.

eg:: $f(A,B) = AB + \bar{A}\bar{B}$

No. of variable = 2

No. of terms = 2, Literals = 4.

Function.

- SOP — Sum of Product (Minterm)
- POS — Product of Sum (Maxterm).

Standard Canonical Form

→ Each term should contain all the variable

eg: $f(A,B) = \bar{A}B + A\bar{B}$ ✓

$f(A,B) = A + \bar{A}B$

Decimal	ABC	MinTerm	MaxTerm	Functions
0	000	$\bar{A}\bar{B}\bar{C}$	$A+B+C$	1
1	001	$\bar{A}\bar{B}C$	$A+B+\bar{C}$	0
2	010	$\bar{A}B\bar{C}$	$A+\bar{B}+C$	1
3	100	$\bar{A}BC$	$A+\bar{B}+\bar{C}$	1
4	100	$A\bar{B}\bar{C}$	$\bar{A}+B+C$	0
5	101	$A\bar{B}C$	$\bar{A}+B+\bar{C}$	0
6	110	$AB\bar{C}$	$\bar{A}+\bar{B}+C$	0
7	111	ABC	$\bar{A}+\bar{B}+\bar{C}$	1

Standard Canonical Sop form:

$$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + ABC$$
$$= m_0 + m_2 + m_3 + m_7.$$
$$= \Sigma m(0,2,3,7)$$
$$= \Sigma(0,2,3,7)$$

Standard Canonical Pos form:

$$F(A,B,C) = (A+B+\bar{C})(\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)$$
$$= M_1 \cdot M_5 \cdot M_2 \cdot M_6$$
$$= \pi M(1,4,5,6)$$
$$= \pi(1,4,5,6).$$

Q1. No. of terms present in Standard Canonical form (Sop)?

Ans. $f(A,B,C) = A + BC$

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

$\therefore$ 5 terms are present.

(1.) Distribution Theorem.

$$(A+B)(A+C)$$
$$= A.A + A.C + AB + BC$$
$$= A + AC + AB + BC$$

$= A[1+C+B] + BC$

$= A1 + BC$

$= A + BC$

$A + BC = (A+B).(A+C)$

eg: $A + \bar{A}B$

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

$= 1(A+B)$

$= A+B.$

Q2. $f(A,B) = \bar{A} + A\bar{B}.$

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

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

(2) Concensus Theorem

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

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

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

$= AB[1+C] + \bar{A}C[1+B]$

$= AB + \bar{A}C$

$BC \rightarrow$ Redundant term

* 3 variable function
* each term consist of 2 variables.
* each variable repeated twice except one.
* One variable repeated in form of compliment.

eg: $AB + \bar{B}C + AC$

$\Rightarrow AB + \bar{B}C$ $\quad AC \rightarrow$ redundant.

(3) Transpose Theorem

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

$= A\bar{A} + AC + \bar{A}B + BC$

$= AC + \bar{A}B + BC$

$= AC + \bar{A}B.$

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

$= AC + \bar{A}B.$

(4) De Morgans Law

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

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

Q1. Find min. no of NAND Gate required to implement $f(A,B,C) = A + ABC + A\bar{B}C.$

$A[1 + BC + B\bar{C}] = A$

$\therefore$ 0 NAND Gate required

Classmate
Date 16/6/23
Page

Q2. Find the minimum no. of NAND gate required to implement the boolean function

Ans $f(A,B,C) = A + ABC + AB\bar{C}$

$= A[1 + BC + B\bar{C}]$

$= A.$

$\therefore$ 0 NAND Gate required

Q3. Minimize the expression $f(A,B) = A + A\bar{B}$.

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

$= A.$

Q4 Minimize the expression $f(A,B) = \bar{A}\bar{B} + \bar{A}B + AB$

Ans. $= \bar{A}[\bar{B} + B] + AB$

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

$= \bar{A} + B$

Q5 Minimize the expression $f(A,B) = \bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB$

Ans $= \bar{A}[\bar{B} + B] + A[\bar{B} + B]$

$= \bar{A} + A \Rightarrow 1$

n=1		n=2				
A	4	$\bar{A}\bar{B}$	$\bar{A}+\bar{B}$	A	0	16 different terms
$\bar{A}$		$\bar{A}B$	$\bar{A}+B$	$\bar{A}$	1	
1		$A\bar{B}$	$A+\bar{B}$	B	$\bar{A}B + A\bar{B}$	
0		AB	A+B	$\bar{B}$	$\bar{A}\bar{B} + AB$	
		minterms		maxterms		

"n" variables $\rightarrow 2^n$ distinct expressions can be formed

2^n minterms 2^n maxterms.

Q6. Minimize the expression $f(A,B) = \bar{A}B + A\bar{B} = A \oplus B$.

Ans Already minimized

Q7. Minimize the expression : $f(A,B) = \bar{A}B + \bar{A}C + BC$

↳ function written in wrong format.

eg:: $f(A,B,C) = \overset{001}{\bar{A}\bar{B}C} + \overset{010}{\bar{A}B\bar{C}} + \overset{100}{A\bar{B}\bar{C}} + \overset{111}{ABC}$.

$= \sum(1,2,4,7)$

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

eg:: $f(B,A) = \overset{01}{\bar{A}B} + \overset{10}{A\bar{B}} + \overset{11}{AB} = \sum m(1,2,3)$ ×

↑↑ $= B\bar{A} + \bar{B}A + BA = \sum m(1,2,3)$ ✓

↳ note the order, function should be written in same order.

Q8. Minimize the expression : $f(A,B,C) = \bar{A}\bar{B} + \bar{A}C + \bar{B}\bar{C}$

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

Q9. Minimize the expression : $f(A,B,C) = (A+B)(A+C)(\bar{B}+C)$

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

Q10. Write the functions for truth table and minimize it.

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

SOP form:

$Y = \bar{A}\bar{B} + \bar{A}B.0 + A\bar{B}.1 + AB.1$

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

$Y = \bar{A}\bar{B} + A[\bar{B}+B]$

$y = A + \bar{A}\bar{B} = (A+\bar{A})(A+\bar{B})$

$\boxed{Y = A + \bar{B}}$

POS:

$Y = (A+B+1).(A+\bar{B}+0).(\bar{A}+B+1).$

$(\bar{A}+\bar{B}+1)$

$= 1(A+\bar{B}).1.1$

$= A + \bar{B}$

Q11. Write the function for truth table and minimize it.

Ans.

A	B	Y
0	0	C
0	1	$\bar{C}$
1	0	1
1	1	1

SOP:

$Y: \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B} + AB$

Q12 Two way switch is an example of which logic?

Ans. X-OR

Q13 If we have 4 variable, then total different expressions will be?

Ans. $2^{2^n} = 2^{2^4} = 2^{16} = 2^6 \times 2^{10}$

$= 64K$.

Q14 $A + BC + \bar{A}C$ is equal to :

$A + BC + \bar{A}C$

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

$= A + C + BC$

$= A + C[1 + B]$

$= A + C$ (not in option!)

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

$= AB[1+C] + \bar{A}\bar{B}[1+\bar{C}]$

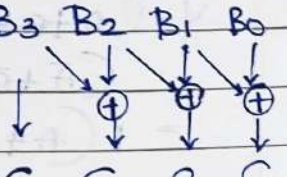
$= AB + \bar{A}\bar{B}$. ——— (Semi minimised) → $AB + \bar{A}\bar{B} + \bar{A}\bar{B}\bar{C}$

Modified Veitched Diagram.

↳ It is also known as "K-Map".

Minimization by K-Map. → Binary → B_3 B_2 B_1 B_0

* Based on graycode
* Graycode:

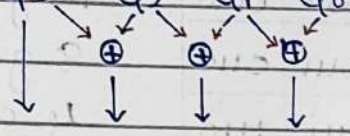


Graycode → G_3 G_2 G_1 G_0

eg: Binary: 0110101
Gray code: 0101111

eg: Binary: 111001011
Gray code: 100101110

→ Gray Code: G_3 G_2 G_1 G_0

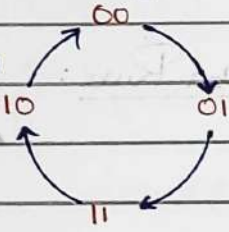


Binary: B_3 B_2 B_1 B_0.

eg: Gray code: 1101011
Binary: 1001101.

Gray Code: Gray code is the code in which Successive numbers differ by 1 bit.

Decimal	Binary	Gray Code
0	00	00
1	01	01
2	10	11
3	11	10



1. Unity Hamming distance code
2. Cyclic code
3. Reflecting code

Decimal	Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

* eg:: $f(A, B)$

MSB (Most Significant bit) LSB (Least Significant bit)

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	$\bar{A}\bar{B}$ 00 ⓪	$\bar{A}B$ 01 ①
A 1	$A\bar{B}$ 10 ②	AB 11 ③

→ $f(A, B, C)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	000 0	001 1	011 3	010 2
A 1	100 4	101 5	111 7	110 6

→ $f(A, B, C, D)$

* Group can be formed in order of 2^n.

AB \ CD	$\bar{C}\bar{D}$ (00)	$\bar{C}D$ (01)	CD (11)	$C\bar{D}$ (10)
$\bar{A}\bar{B}$ (00)	0 — 0000	1 — 0001	3 — 0011	2 — 0010
$\bar{A}B$ (01)	4 — 0100	5 — 0101	7 — 0111	6 — 0110
AB (11)	12 — 1100	13 — 1101	15 — 1111	14 — 1110
$A\bar{B}$ (10)	8 — 1000	9 — 1001	11 — 1011	10 — 1010

16 group = 2^4 → 4 variables reduce
8 group = 2^3 → 3 variables reduce
4 group = 2^2 → 2 variables reduce
2 group = 2^1 → 1 variable reduce
1 group = 2^0 → 0 variable reduce.

Rule of Minimization.

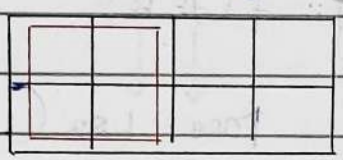
Terms will minimise

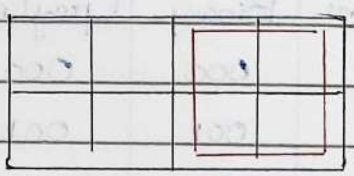
CJ Baba Rule: Kam se kam group banana hai and bade se bade group banana hai.

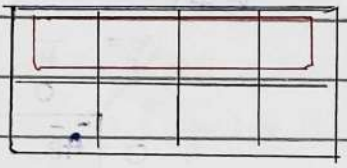
↓

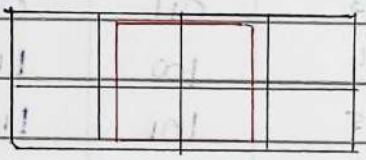
Variable will minimize.

Quad (making group)

1. 

2. 

3. 

4. 

And many Similar combinations possible

etc.

Q1. $f(A,B) = \bar{A}\bar{B} + \bar{A}B + A\bar{B} = \sum m(0,1,3)$

A \ B	$\bar{B}$ (0)	B (1)
$\bar{A}$ (0)	1	1 — $\bar{A}$
A (1)	0	1 — B

$f(A,B) = \bar{A}\bar{B}.1 + \bar{A}B.1 + A\bar{B}.0 + AB.1$

$\Rightarrow \bar{A} + B$

Q2 $f(A,B) = \bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB = \sum m(0,1,2,3)$

A \ B	$\bar{B}$ (0)	B (1)
$\bar{A}$ (0)	1	1
A (1)	1	1

$\Rightarrow 1$.

Q3. $f(A,B,C) = \sum m(0,1,3,6,7)$

A \ BC	$\bar{B}\bar{C}$ (00)	$\bar{B}C$ (01)	BC (11)	$B\bar{C}$ (10)
$\bar{A}$ (0)	1 (0)	1 (1)	1 (3)	(2)
A (1)			1 (7)	1 (6)

(4) (5)

$\bar{A}C$

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

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

AB

A \ BC	$\bar{B}\bar{C}$ (00)	$\bar{B}C$ (01)	BC (11)	$B\bar{C}$ (10)
$\bar{A}$ (0)	1	1	1	
A (1)			1	1

BC

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

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

AB

A \ BC	$\bar{B}\bar{C}$ (00)	$\bar{B}C$ (01)	BC (11)	$B\bar{C}$ (10)
$\bar{A}$ (0)	1	1	(1)	
A (1)			1	1

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

Semi minimized expression.

Question Practice Session

Q1. If $x \odot y = \bar{x} + y$ and $z = x \odot y$ Then $z \odot y$ will be:

$z \odot Y = \bar{z} + y = \overline{x \odot y} + y$

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

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

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

$= x + y$.

Q2. If $A * B = AB + \bar{A}\bar{B}$ and $C = A * B$. Then which one is correct?

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

$C = A \odot B$ $\qquad (A \odot B) \odot (A \odot B)$

$B \odot C = A$. $\qquad x \odot x = 1$.

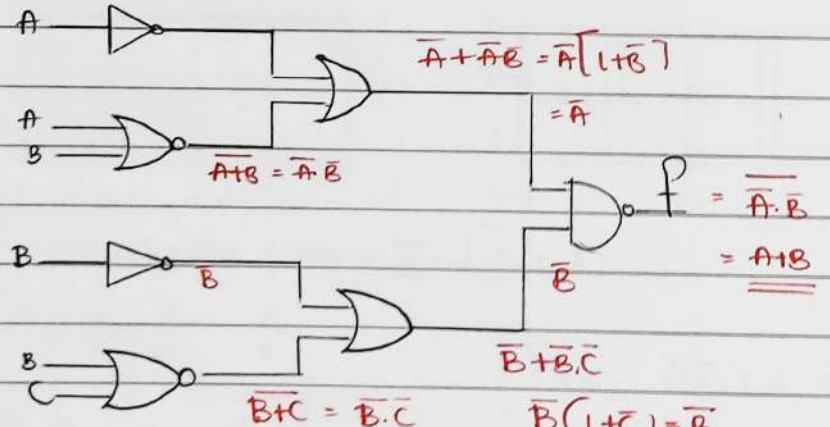
(i) $A = B * C$ ✓

(ii) $B = A * C$ ✓

(iii) $A \odot B \odot C = 1$ ✗

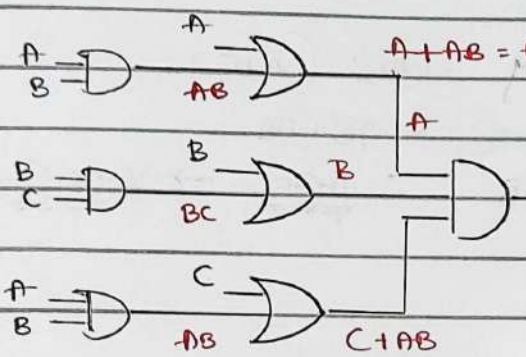
(iv) $A = B$ ✗

Q3. The output for the given logic circuit will be:



$\Rightarrow f = A + B$

Q4. Consider the given logic circuit with the input A, B and C then $f(A,B,C)$ will be.



$f(A,B,C) = A \cdot B \cdot (AB+C)$

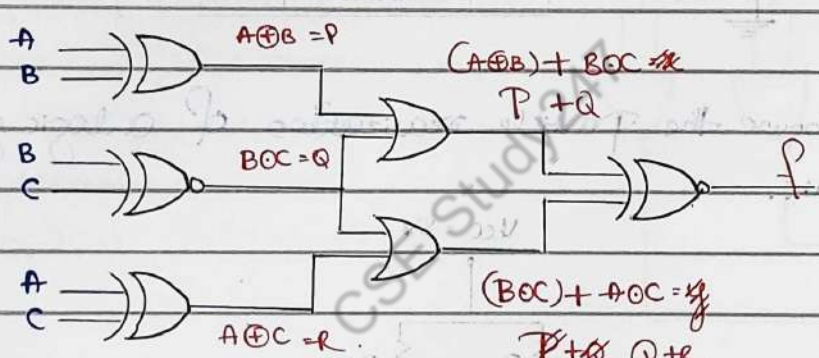
$= AB \cdot AB + ABC$

$= AB + ABC = AB(1+C)$

$= AB$

$\Rightarrow$ 

Qs. The output for the given logic circuit will be:



$f = \bar{P}\bar{Q}R + P\bar{Q}\bar{R} + PQR + Q\bar{R} + QR$

$f = \bar{P}\bar{Q}R + Q[P+\bar{R}+R] + PR$

$f = \bar{P}\bar{Q}R + Q + PR$

$= \bar{P}\bar{Q}R + Q + PR$

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

$f = \underline{P\bar{R} + PR + Q \Rightarrow (P \odot R) + Q}$

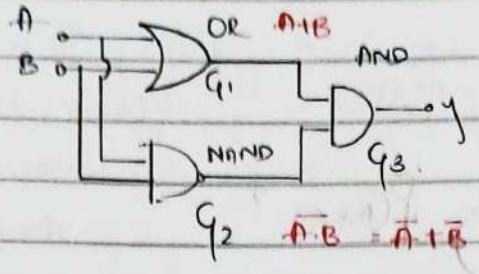
A \ BC	00	01	11	10
0	1		1	
1	1		1	

$f = \bar{B}\bar{C} + BC$

$= \underline{B \odot C}$

ABC	P A⊕B	Q B⊙C	R A⊕C	x	y	x⊙y
000	0	1	0	1	1	1
001	0	0	1	0	1	0
010	1	0	0	1	0	0
011	1	1	1	1	1	1
100	1	1	1	1	1	1
101	1	0	0	1	0	0
110	0	0	1	0	1	0
111	0	1	0	1	1	1

Q6. The following logic gate circuit equivalent to

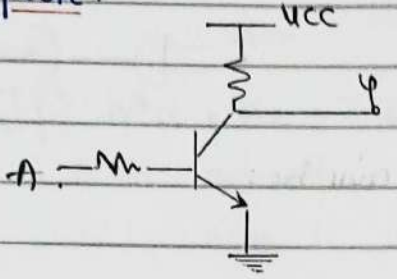


$= (A+B)\cdot(\bar{A}+\bar{B})$

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

$= A \oplus B$ → X-OR Gate

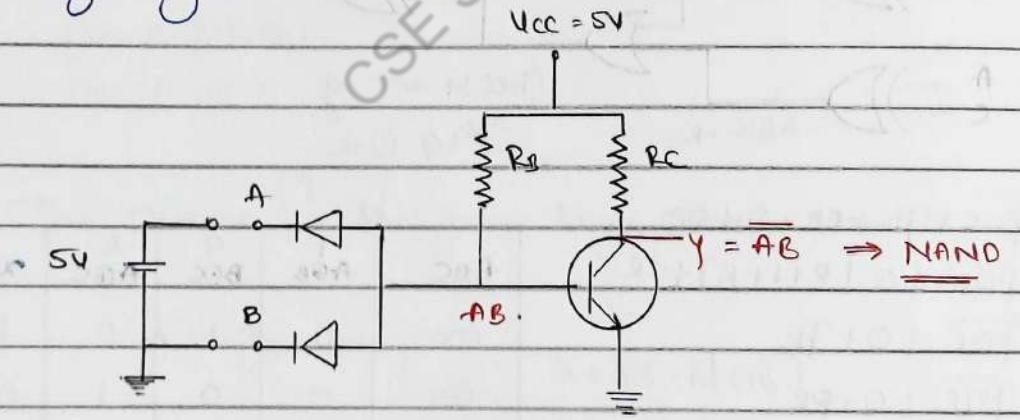
Note:



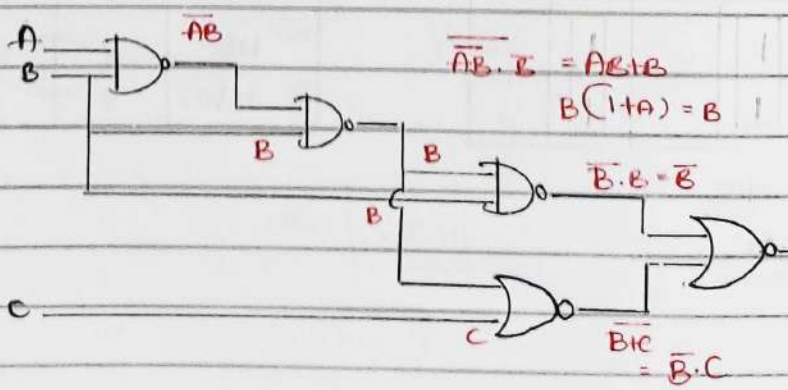
A	Transistor	y
0	cutoff	1
1	Saturation	0

} NOT Gate

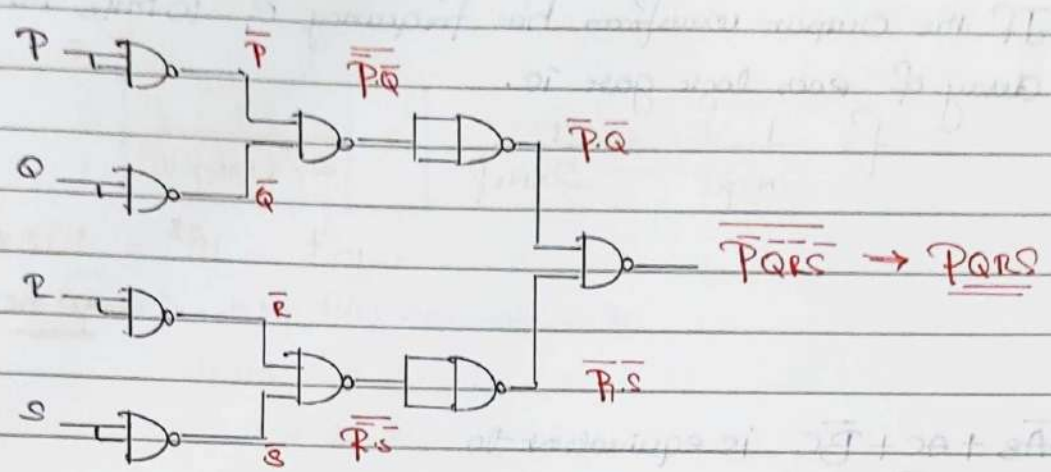
Q7. Figure shows the particle realization of a logic gate. Identify the logic gate.



Q8. For the logic circuit shown, the simplified boolean expression for the output y is.

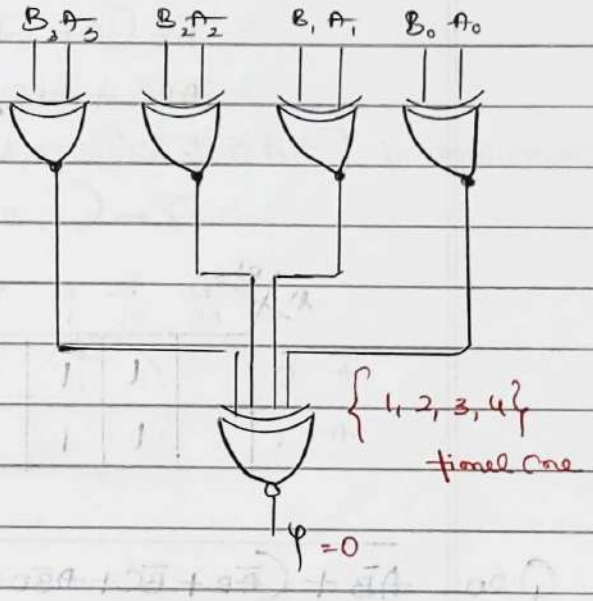


Q13. For the circuit shown in fig. the boolean expression for the output y in terms of input P, Q, R and S is.



Q14. A_3, A_2, A_1, A_0, $B_3B_2B_1B_0$ is shown in figure. To get output y=0, choose one pair of correct input numbers.

⇒ 0010, 1011

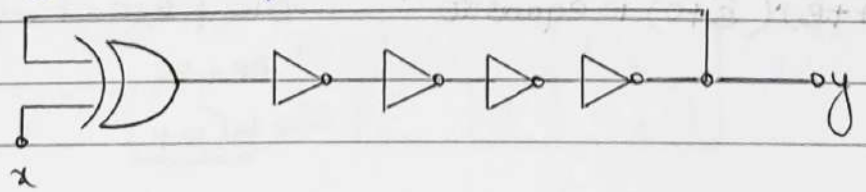


Q15. The minimum number of two input NAND gate required to implement $y = abcd$ is.

$y = abcd$

$NAND = (2n-2)+k = (2\times4-2)+0 = 6$.

Q16. All the logic gates in the circuit shown below, have equal finite propagation delay.



Q17. The Circuit can be used as a clock generator, if $x = 1$

Q18. If the output waveform has frequency of 10 MHz the propagation delay of each logic gate is.

$$f = \frac{1}{2N t_{pd}} = \frac{1}{2 \times N \times f} = \frac{1}{2 \times 5 \times 10 \times 10^6}$$

$$= \frac{10^{-7}}{10} = 10^{-8} = 10 \times 10^{-9}$$

$$= 10 \text{ ns}$$

Q19 $\bar{A}B + AC + \bar{B}C$ is equivalent to

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

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

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

$$= \sum m(2,3,5,7,1)$$

$$= \sum m(1,2,3,5,7)$$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0		1	1	1
A 1		1	1	

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

Q20 $\overline{A\bar{B} + (\bar{A}B + \bar{B}C + A\bar{B}D + A\bar{B}\bar{D})}$ is equal to

$$\overline{A\bar{B} + \bar{A}B + \bar{B}C + A\bar{B}D + A\bar{B}\bar{D}}$$

$$\overline{A\bar{B}[1 + D + \bar{D}] + \bar{A}B + \bar{B}C}$$

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

$$\overline{A\bar{B} \cdot \bar{A}B \cdot \bar{B}C}$$

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

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

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

$$\bar{A}\bar{B}B + ABB + \bar{A}\bar{B}C + AB\bar{C}$$

$$AB[1+\bar{C}] + \bar{A}\bar{B}C$$

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

Q21. $\overline{(\bar{A}+\bar{B})\cdot(\bar{B}+C)}$ is equal to : $\overline{\bar{A}+\bar{B}} + \overline{\bar{B}+C}$

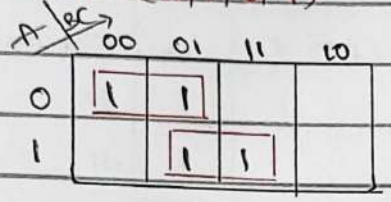
$$AB + B\bar{C}$$

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

Q22. $\bar{A}\bar{B}+AC+\bar{B}C$ is equivalent to

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

$\Sigma m(0,1,5,7)$



$\Rightarrow \bar{A}\bar{B}+AC$

Q23. $(A+B)(A+C)(A+\bar{C})$ is equivalent to

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

$= A+BC.\bar{C}$

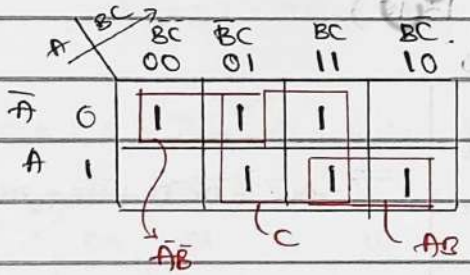
$\Rightarrow A$

HW.

Q24. A logical function is given as

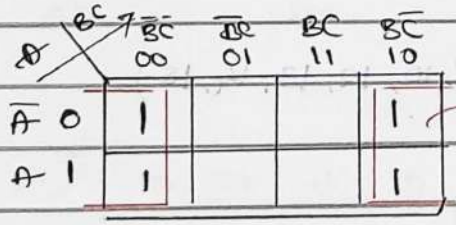
$f(A,B,C,D) = B\bar{C}[A+B\bar{C}D+\bar{B}CD+\bar{A}B\bar{C}+\bar{A}\bar{B}\bar{C}]$ is equivalent to

Q25. $f(A,B,C) = \Sigma m(0,1,3,5,6,7)$



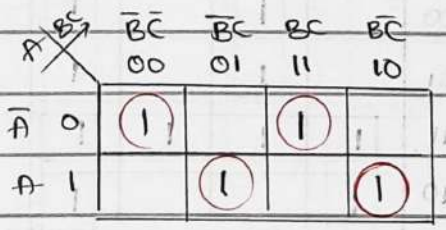
$= \bar{A}\bar{B}+A\bar{B}+C$

Q26 $f(A,B,C) = \Sigma m(0,2,4,6)$

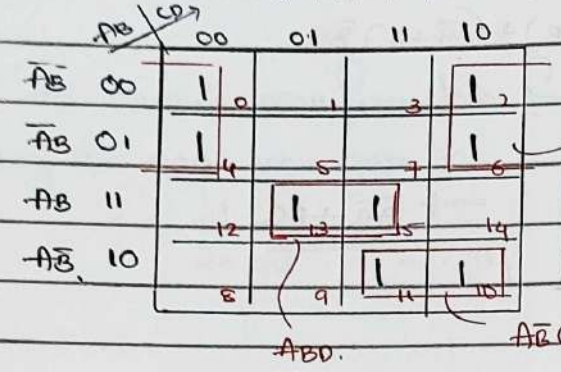


$\bar{C}$ $\Rightarrow \bar{C}$.

Q27 $f(A,B,C) = \Sigma m(0,3,5,6)$

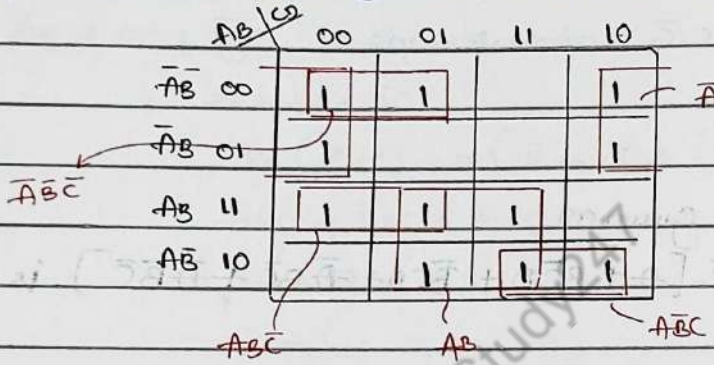


Q28 $f(A,B,C,D) = \sum(0,2,4,6,10,11,13,15)$



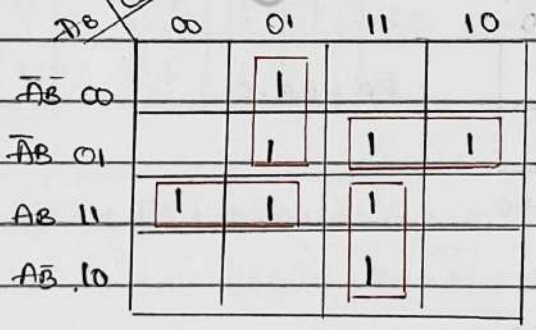
$\Rightarrow \bar{A}\bar{D} + ABD + A\bar{B}C$

Q29. $f(A,B,C,D) = \sum m(0,1,2,4,6,9,10,11,12,13,15)$



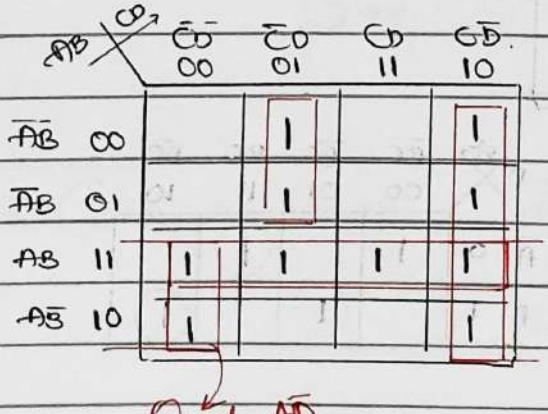
$\Rightarrow \bar{A}\bar{D} + AD + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}B\bar{C}$

Q30. $f(A,B,C,D) = \sum m(1,5,6,7,11,12,13,15)$



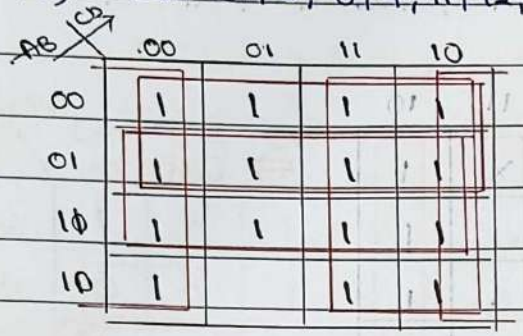
$\Rightarrow A\bar{B}C + \bar{A}C\bar{D} + \bar{A}B + ACD$

Q31. $f(A,B,C,D) = \sum m(1,2,5,6,8,10,12,13,14,15)$



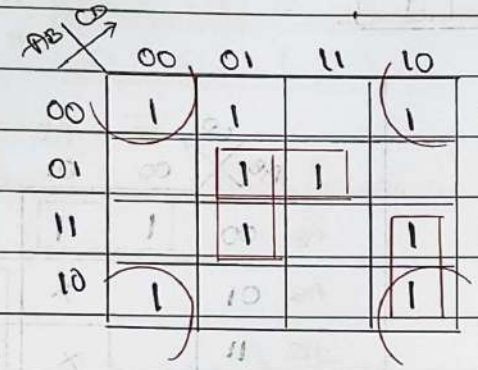
$\Rightarrow AB + C\bar{D} + A\bar{D} + \bar{A}C\bar{D}$

Q 22. $f(A,B,C,D) = \Sigma m(1,5,6,7,11,12,13,15)$



$\Rightarrow \bar{A}+B+C+\bar{D}$.

eg::



$\Rightarrow \bar{B}\bar{D} + B\bar{C}D + \bar{A}BD + AC\bar{B} + \bar{A}B\bar{C}$

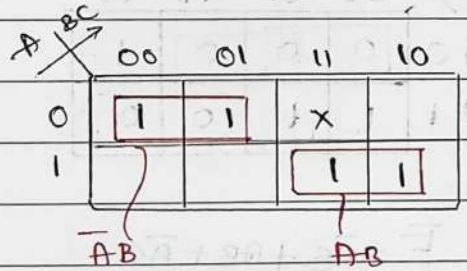
$\Rightarrow \bar{B}\bar{D} + B\bar{C}D + \bar{A}BD + AC\bar{D} + \bar{A}C\bar{D}$

Don't Care Condition:

Combination of inputs on which the output may or may not depends are called don't care conditions.

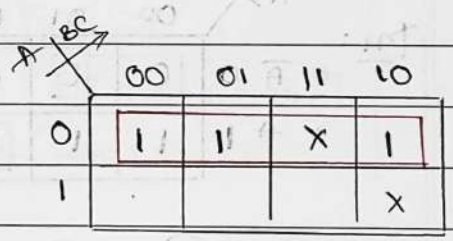
$f(A,B) = \bar{A}B + (AB)$

Q1. $f(A,B,C) = \Sigma m(0,1,6,7) + \Sigma d(3)$ ← don't care.

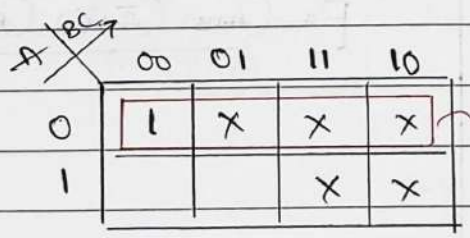


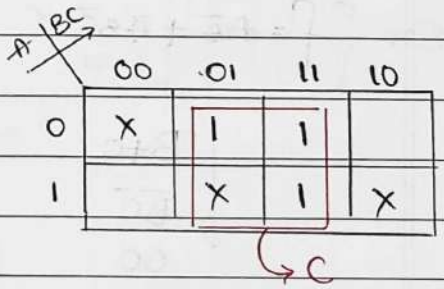
$\Rightarrow AB + \bar{A}B$

$f(A,B,C) = \Sigma m(0,1,2) + \Sigma d(3,6)$



eg::





Q2. $f(A,B,C,D) = \sum m(0,2,4,6,7,8,10,11,12,14,15) + \sum d(1,3)$.

AB \ CD	00	01	11	10
00	1	X	X	1
01	1		1	1
11	1		1	1
10	1		1	1

$\Rightarrow C + \bar{D}$.

eg::

AB \ CD	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{A}\bar{B}$ 00	1	1	X	1
$\bar{A}B$ 01			1	X
AB 11		X	1	
$A\bar{B}$ 10	X	X		1

$\Rightarrow \bar{A}\bar{B} + \bar{B}\bar{D} + BCD$.

AB \ CD	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{A}\bar{B}$ 00	1	1		
$\bar{A}B$ 01	1	X	X	1
AB 11		X	X	1
$A\bar{B}$ 10			1	

$\Rightarrow \bar{A}\bar{B}\bar{C} + BC + ACD$

Q3. $F(A,B,C) = \Pi m(0,1,3,6,7) = \sum m(2,4,5)$

M1:

A \ BC	$\bar{B}+C$ 00	$\bar{B}+\bar{C}$ 01	$B+\bar{C}$ 11	$B+C$ 10
A 0	0	0	0	1
$\bar{A}$ 1	1	1	0	0

POS $\begin{cases} F = (A+B).(\bar{A}+\bar{B}).(A+\bar{C}) \\ F = (A+B)(\bar{A}+\bar{B})(\bar{B}+\bar{C}) \end{cases}$

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

$B+C$
$\bar{B}\bar{C}$
00

M2 (Method 2)

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0	0	0	1
A 1	1	1	0	0

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

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

$F = (A+B)(\bar{A}+\bar{B}).(A+\bar{C})$ ✓

Q4. $f(A,B,C,D) = \Sigma m(0,1,3,6,7,10,11)$ minimize it in Pos form

$= \pi M(2,4,5,8,9,12,13)$

AB \ CD	C+D	C+$\bar{D}$	$\bar{C}$+$\bar{D}$	$\bar{C}$+D
	00	01	11	10
A+B 00	1	1	1	0
A+$\bar{B}$ 01	0	0	1	1
$\bar{A}$+$\bar{B}$ 11	0	0	0	0
$\bar{A}$+B 10	0	0	1	1

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

AB \ CD	C+D	C+$\bar{D}$	$\bar{C}$+$\bar{D}$	$\bar{C}$+D
	00	01	11	10
A+B 00	0	0		
A+$\bar{B}$ 01	0		0	X
$\bar{A}$+$\bar{B}$ 11		X	X	
$\bar{A}$+B 10		0	0	X

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

Variable inside the K-map.

eg:

A \ B	$\bar{B}$	B
	0	1
$\bar{A}$ 0	C	C
A 1		

$\bar{A}.C$

$\rightarrow \bar{A}\bar{B}C + \bar{A}BC$

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

$= \bar{A}C$

eg:

A \ B	$\bar{B}$	B
	0	1
$\bar{A}$ 0	1	C
A 1		

$f(A,B,C) = \bar{A}\bar{B}.1 + \bar{A}BC$

$= \bar{A}[\bar{B}+BC]$

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

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

Method 1 — Boolean Algebra

Method 4

Case 1: C=0

A \ B	$\bar{B}$	B
	0	1
$\bar{A}$ 0	1	0
A 1	0	0

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

Case 2: C=1

A \ B	$\bar{B}$	B
	0	1
$\bar{A}$ 0	1	1
A 1	0	0

$\Rightarrow \bar{A}C$

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

$\Rightarrow \bar{A}[C+\bar{B}C]$

$\Rightarrow \bar{A}[(\bar{B}+C)]$

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

Method 2: $f(A,B,C) = \bar{A}\bar{B}(\bar{C}+C) + \bar{A}BC$

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

$= \sum m(0,1,3) = \pi M(2,4,5,6,7)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1	1	1	0
A 1	0	0	0	0

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

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

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1	1	1	0
A 1	0	0	0	0

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

Method 3

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	1	C
A 1		

→ $\bar{A}\cdot(\bar{B}+C)$

Q1.

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	1	C
A 1		C

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

C=0

A \ B	0	1
0	1	
1		

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

C=1

A \ B	0	1
0	1	1
1		1

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

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

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

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

Q2.

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0		1
A 1	C	C

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

$\Rightarrow AC + \bar{A}B + BC$

A \ B	$\bar{A}$ 0	A 1
$\bar{B}$ 0		1
B 1	0	0

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

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0		1
A 1	1	1

$(A+B)C$

$\bar{A}B\bar{C} + AC + BC$

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

Implicants and Prime Implicants.

1. **Implicants**: The total number of min-terms in the boolean expression are called implicants. Or in kmap the total no. of 1 is called implicants.

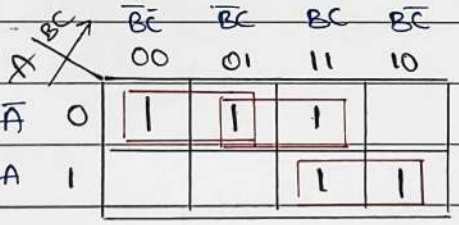
2. **Prime Implicants**: Total number of possibilities of formation of group are called prime implicants.

3. **Essential Prime Implicants / Selective PI**: If there are two answers for a boolean expression then the terms common in both the answers are called essential prime implicants.

4. **Reduced Prime Implicants**: The terms that are not essential prime implicants are called Reduced prime implicants.

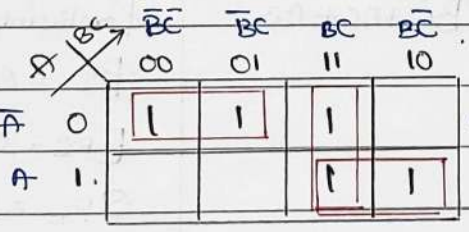
Q1. $f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC$

$= \sum_m (0,1,3,6,7)$



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

EPI.



$\Rightarrow \bar{A}\bar{B} + A\bar{B} + BC.$

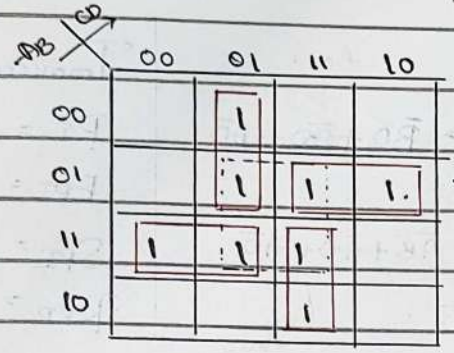
Implicants = 5
Prime Implicants = 4
Essential prime implicants = 2
SPI = 1
RPI = 1.

$PI\{\bar{A}\bar{B}, A\bar{B}, \bar{A}C, BC\}$

Q2. $f(A,B,C) = \sum_m (2,3,4,5,7)$

$PI\{A\bar{B}, \bar{A}B, BC, AC\}$

Q5. $f(A,B,C,D) = \sum m(1,5,6,7,11,12,13,15)$



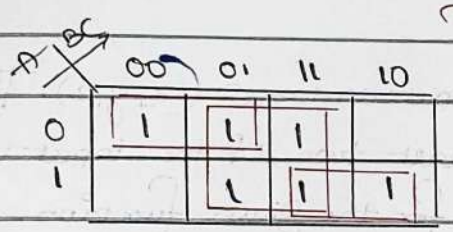
$PI\{\bar{A}\bar{C}D, A\bar{B}C, ACD, \bar{A}BC, BD\}$.

$\Rightarrow \bar{A}\bar{C}D + \bar{A}BC + ACD + A\bar{B}C$

$EPI\{\bar{A}\bar{C}D + \bar{A}BC + ACD + A\bar{B}C\}$.

Implicants = 8
PI = 5
EPI = 4
SPI = 0
RPI = 1

Q6.



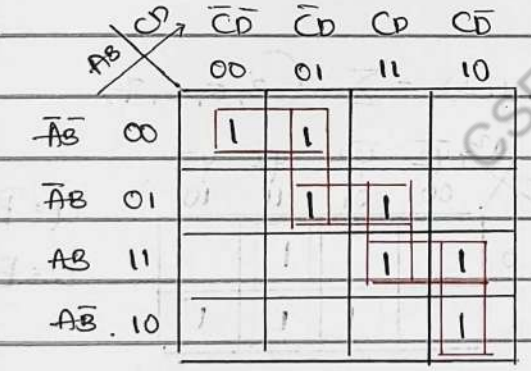
$PI\{\bar{A}\bar{B} + AB + C\}$

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

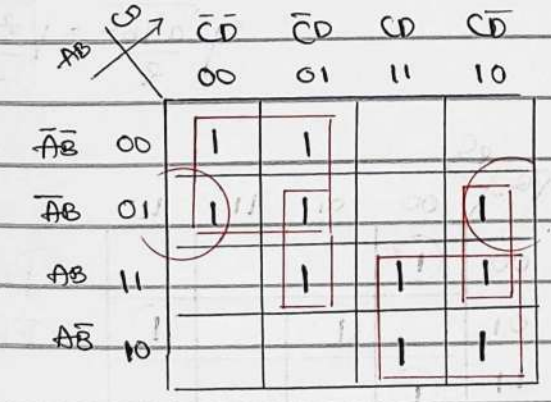
Implicants = 6
PI = 3
EPI = 3
SPI = 0 RPI = 0

HW

Q7.



Q8.



Logic

1 → 10V } +ve logic
0 → 5V }

+ve Logic
1 → High
0 → Low

-ve logic
1 → low
0 → high

1 → 5V } -ve logic
0 → 10V }

Dual

1 → 5V } +ve logic
0 → 0V }

1 → 0V } -ve logic
0 → 5V }

1 → 0V } +ve logic
0 → -5V }

1 → -10V } -ve logic
0 → -5V }

AND Gate (+ve)

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

Logic conversion ↓ Dual

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

OR

Dual ⇒ +ve logic ⟷ -ve logic

1. 1 ⟷ 0 ; 0 ⟷ 1 ; $f = A \cdot 1$
2. AND ⟷ OR ; A ⟷ A ; $f^0 = A + 0 = A$
3. . ⟷ + ; $\bar{A} \leftrightarrow \bar{A}$; ∴
4. NAND ⟷ NOR ; ∴
5. X-OR ⟷ X-NOR ; ∴
6. Buffer ⟷ Buffer
7. Inverter ⟷ Inverter.

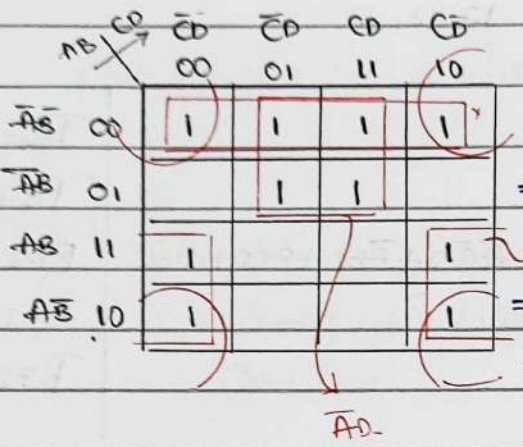
eg: $f = AB + CD + \bar{E}F$
$f^0 = (A+B) \cdot (C+D) \cdot (\bar{E}+F)$

$f = \overline{ABC + DEF}$
$f^0 = \overline{(A+B+C) \cdot (D+E+F)}$

eg: $f = AB + CD$ $f^0 = (AB+B) \cdot (C+D)$
$f^{00} = AB + CD$

$f^{00} = f$

Q9.



Ans 1

$\Rightarrow \bar{B}\bar{D} + \bar{A}\bar{D} + A\bar{D}$

$A\bar{D}$

$\Rightarrow A\bar{B} + \bar{A}D + A\bar{D}$.

Ans 2.

Implicants = 10

PI = 4

EPI = 2

SPI = 1

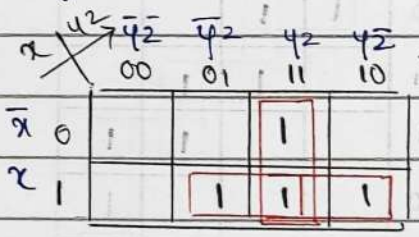
RPI = 1

Q10. Let a function F which has 3 input variables (x, y, z). The function F will be high only when atleast two of the input variables are set to high. Draw the K-map for the given function. Let the number of PI in K-map = 'a' and the no. of EPI in K-map = 'b'. Find the quadratic mean of 'a' and 'b'.

Ans:

	x	y	z	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
(3)	0	1	1	1
4	1	0	0	0
(5)	1	0	1	1
(6)	1	1	0	1
(7)	1	1	1	1

$f(x, y, z) = \sum m(3, 5, 6, 7)$



a = PI = 3

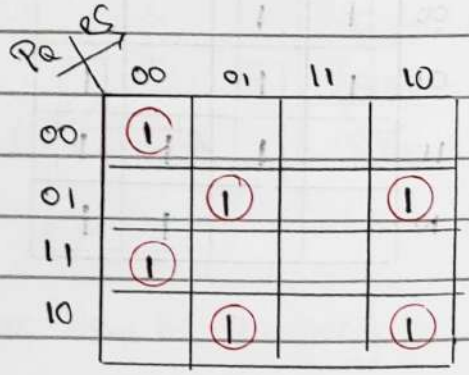
b = EPI = 3

$f = xy + yz + xz$.

Quadratic mean:

$= \sqrt{\frac{a^2+b^2}{2}} = \sqrt{\frac{3^2+3^2}{2}} = 3$

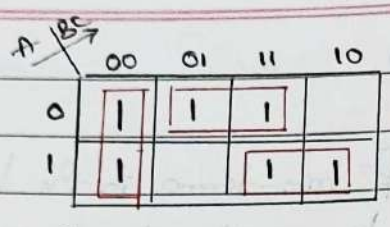
Q11. Find the number of prime implicants & essential prime implicants in the given K.



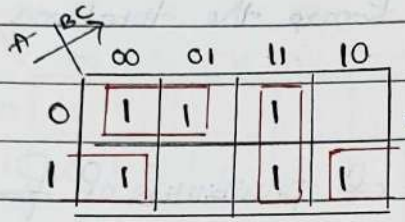
PI = 6

EPI = 6

Q3. $f(A,B,C) = \Sigma m(0,1,3,4,6,7)$.



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

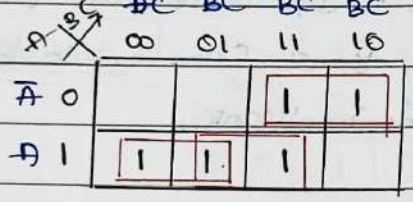


$\Rightarrow \bar{A}\bar{B} + BC + A\bar{C}$.

Implicants = 6
PI = 6
EPI = 0 (term common in both)
SPI = 3
RPI = 3.

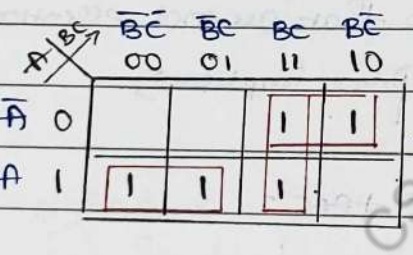
$PI\{\bar{A}\bar{B} + BC + A\bar{C} + \bar{B}\bar{C} + \bar{A}C + A\bar{B}\}$.

Q2 (continued)



$\Rightarrow A\bar{B} + \bar{A}B + AC$

EPI

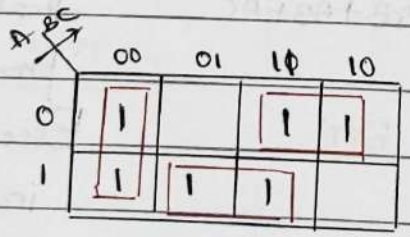


$\Rightarrow A\bar{B} + \bar{A}B + BC$

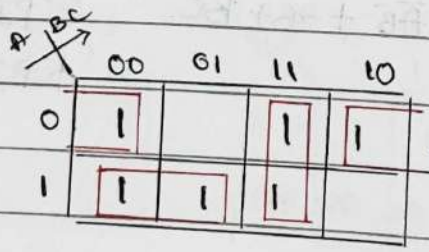
Implicants = 5
PI = 4
EPI = 2
SPI = 1
RPI = 1.

$PI\{A\bar{B} + \bar{A}B + AC + BC\}$

Q4. $f(A,B,C) = \Sigma m(0,2,3,4,5,7)$



$\Rightarrow \bar{B}\bar{C} + AC + \bar{A}B$



$\Rightarrow \bar{A}\bar{C} + BC + A\bar{B}$.

Implicants = 6
PI = 6
EPI = 0
SPI = 3
RPI = 3

$PI\{\bar{B}\bar{C} + AC + \bar{A}B + \bar{A}\bar{C} + BC + A\bar{B}\}$

eg. $f = A$

$f^D = A$

$\boxed{f^D = f}$ → Self Dual.

$n=1$ (no. of variables)		$n=2$ (no. of variables)			
A		$\bar{A}\bar{B}$	$\bar{A}+\bar{B}$	A	0
$\bar{A}$		$\bar{A}B$	$\bar{A}+B$	$\bar{A}$	1
0	Self dual = 2	$A\bar{B}$	$A+\bar{B}$	B	$\bar{A}B + A\bar{B}$
1		AB	$A+B$	$\bar{B}$	$\bar{A}\bar{B} + AB$
					Self dual = 4

* n variables → 2^{2^n} Expressions. → $2^{2^{n-1}}$ Self dual expressions.

Combinational Circuit.

* A circuit without feedback or memory are called Combinational Circuit.
* Static Circuit.

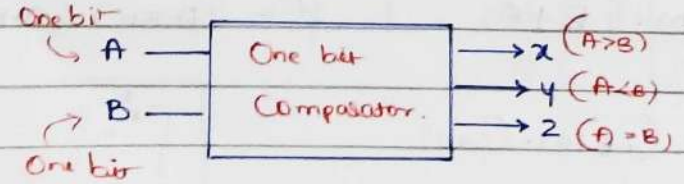
Designing of Combinational Circuit:

Step 1 : Find the number of inputs and outputs.
Step 2 : Write the truth table
Step 3 : Write the logical expression
Step 4 : Minimize the logical expressions.
Step 5: Hardware implementation.

Comparator

1. Design a one bit Comparator

Step 1:



Step 2: Truth Table

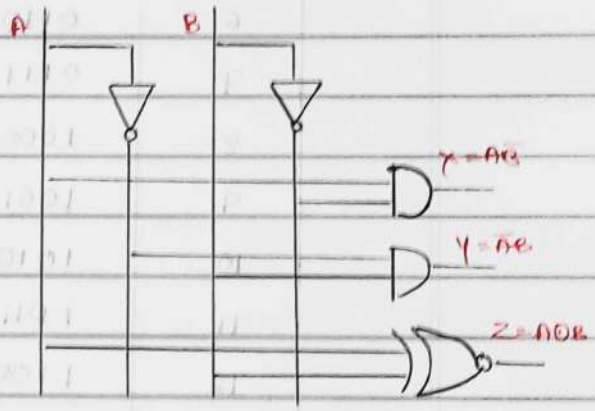
		(A>B)	(A<B)	(A=B)
A	B	x	y	z
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Step 3: Logical expression:

$x\,(A>B) = A\bar{B}$

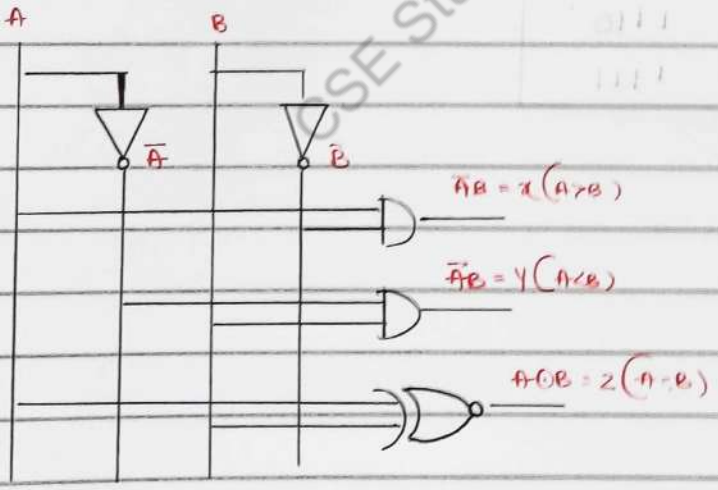
$y\,(A<B) = \bar{A}B$

$z\,(A=B) = \bar{A}\bar{B} + AB = A \odot B$

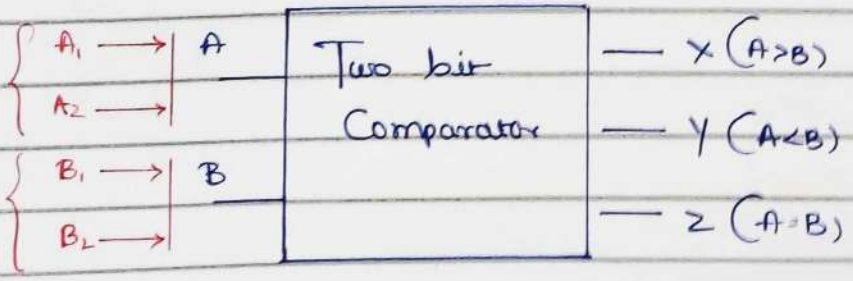


Step 4: Minimization (Already minimized)

Step 5: Hardware implementation.



Two Bit Comparator:



Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Step 2. Truth Table

A_1	A_0	B_1	B_0	X	Y	Z
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Step 3 Logical expression:

$X(A>B) = \sum m(4, 8, 9, 12, 13, 14)$

$Y(A<B) = \sum m(1, 2, 3, 6, 7, 11)$

$Z(A=B) = \sum m(0, 5, 10, 15)$

Step 4. Minimization

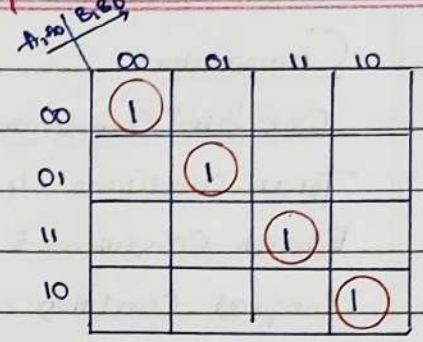
for X.

A_1A_0 \ B_1B_0	$\bar{B}_1\bar{B}_0$ (00)	$\bar{B}_1B_0$ (01)	B_1B_0 (11)	$B_1\bar{B}_0$ (10)
$\bar{A}_1\bar{A}_0$ (00)				
$\bar{A}_1A_0$ (01)	1			
A_1A_0 (11)	1	1		1
$A_1\bar{A}_0$ (10)	1	1		

for Y

A_1A_0 \ B_1B_0	$\bar{B}_1\bar{B}_0$ (00)	$\bar{B}_1B_0$ (01)	B_1B_0 (11)	$B_1\bar{B}_0$ (10)
$\bar{A}_1\bar{A}_0$ (00)		1	1	1
$\bar{A}_1A_0$ (01)			1	1
A_1A_0 (11)				
$A_1\bar{A}_0$ (10)			1	

for '2'



Semiminimised

$X(A>B) = A_1\bar{B}_1 + \bar{A}_1 A_0 \bar{B}_1\bar{B}_0 + A_1 A_0 B_1 \bar{B}_0$

$= A_1\bar{B}_1 + (\bar{A}_1\bar{B}_1 + A_1B_1) A_0\bar{B}_0$

$= A_1\bar{B}_1 + (A_1 \odot B_1) A_0\bar{B}_0$

Minimised

$X(A>B) = A_1\bar{B}_1 + A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0$

$= A_1\bar{B}_1 + (A_1 + \bar{B}_1) A_0\bar{B}_0$

$Z(A=B) = (A_1 \odot B_1) \cdot (A_0 \odot B_0)$

Semi Minimised

$Y(A<B) = \bar{A}_1B_1 + \bar{A}_1\bar{A}_0\bar{B}_1B_0 + A_1\bar{A}_0B_1B_0$

$= \bar{A}_1B_1 + (\bar{A}_1\bar{B}_1 + A_1B_1)\bar{A}_0B_0$

$= \bar{A}_1B_1 + (A_1 \odot B_1)\bar{A}_0B_0$

Minimised

$Y(A<B) = \bar{A}_1B_1 + (\bar{A}_1 + B_1)\bar{A}_0B_0$

H.W. Steps

Hardware Implementation

Comparators

One bit Comparator	Two bit combinations	Three bit comparators
Total Conditions = 4	Total condition = 16	Total combinations = 64
Equal Condition = 2	Equal Condition = 4	Equal condition = 8
Unequal condition = 2	Unequal " = 12	Unequal " = 56
Greater = less cond = 1	Greater = less " = 6.	Greater = less = 28.

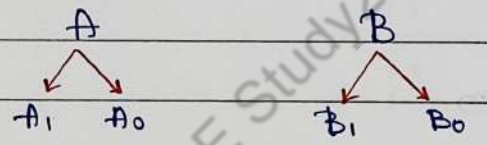
'N' bit comparators

Total condition = 2^{2n}

Equal Condition = 2^n

Unequal Condition = $2^{2n} - 2^n$.

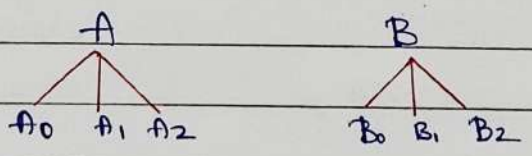
Greater = less = $(2^{2n} - 2^n)/2$.



$X(A>B) = A_1\bar{B}_1 + (A_1 \odot B_1)A_0\bar{B}_0$

$Y(A<B) = \bar{A}_1 B_1 + (A_1 \odot B_1)\bar{A}_0 B_0$

$Z(A=B) = (A_1 \odot B_1) \cdot (A_0 \odot B_0)$.

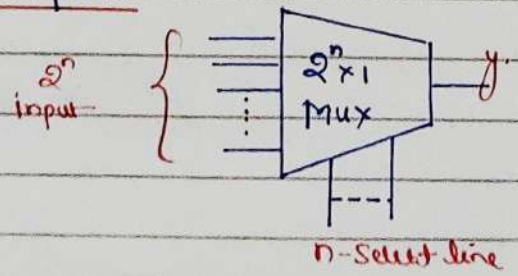


$X(A>B) = A_2\bar{B}_2 + (A_2 \odot B_2)A_1\bar{B}_1 + (A_2 \odot B_2)(A_1 \odot B_1)A_0\bar{B}_0$.

$Y(A<B) = \bar{A}_2 B_2 + (A_2 \odot B_2)\bar{A}_1 B_1 + (A_2 \odot B_2)(A_1 \odot B_1)\bar{A}_0 B_0$.

$Z(A=B) = (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$.

Multiplexer.



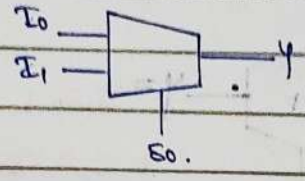
→ 2^n to 1 Converter

→ Universal logic

→ AND OR logic

Q1. Design a 2x1 Mux.

Step 1:



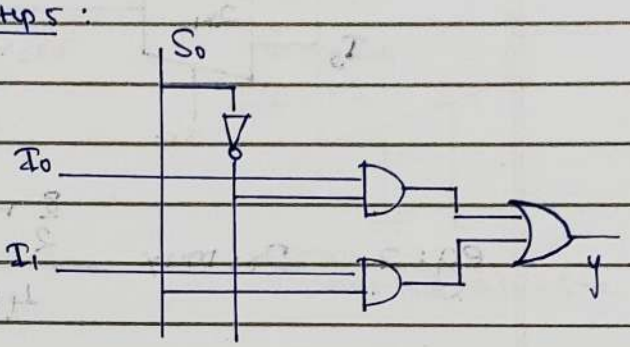
Step 2:

S_0	y
0	I_0
1	I_1

Step 3 : $y = \bar{S_0} I_0 + S_1 I_1$

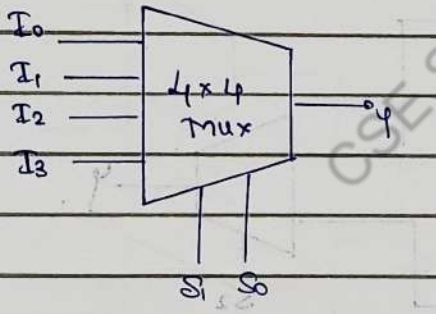
Step 4 : Already Minimized

Step 5 :



Q2. Design a 4x1 mux.

St.1



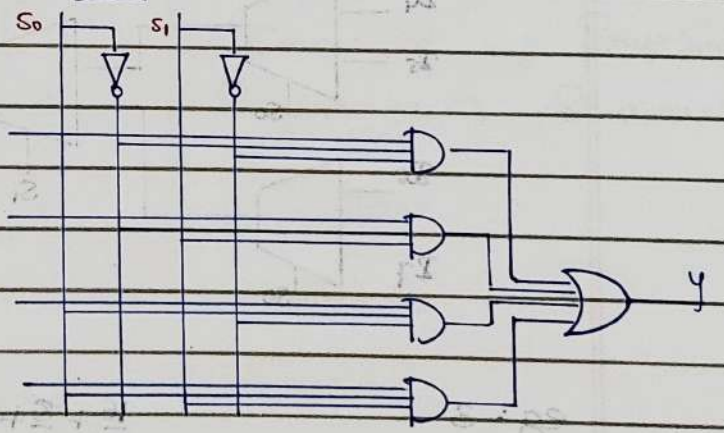
Step 2:

S_1	S_0	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Step 3 : $y = \bar{S_1}\bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$

Step 4 : No minimization

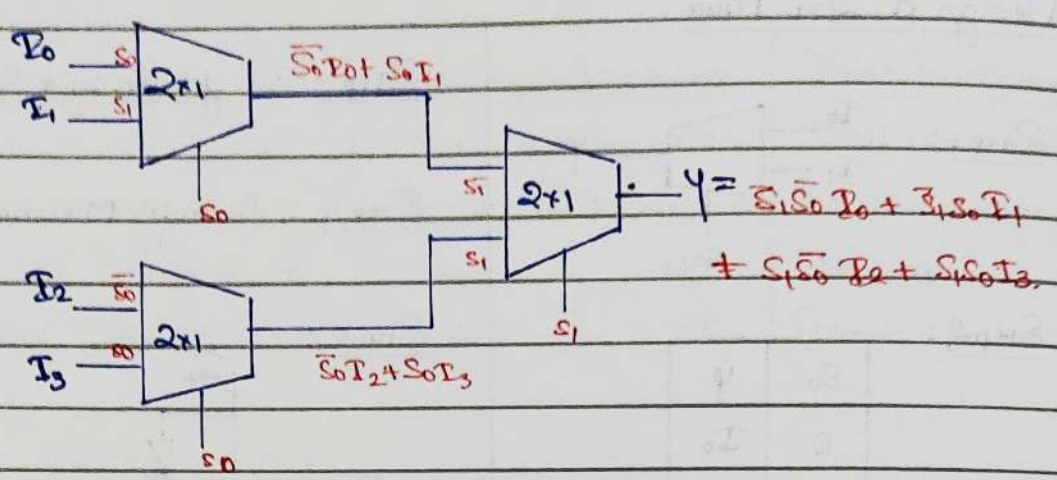
Step 5 :



Type: 1 Designing of higher order mux by using lower order mux.

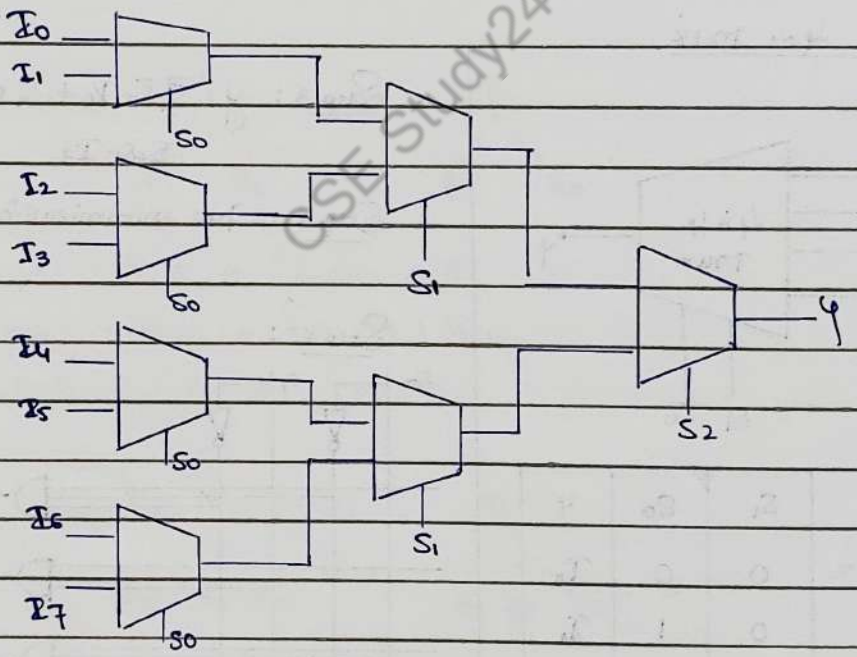
eg. 2x1 mux $\frac{2^2}{2} + \frac{2^1}{2}$ 4x1 mux

$2 + 1 = 3$.



eg: 2 2×1 mux $\xrightarrow{\frac{8}{2}+\frac{4}{2}+\frac{2}{2}}$ 8×1 mux

4+2+1 = 7.



eg: 3

2×1 mux $\xrightarrow{\frac{16}{2}+\frac{8}{2}+\frac{4}{2}+\frac{2}{2}}$ 16×1 mux

8+4+2+1 = 15

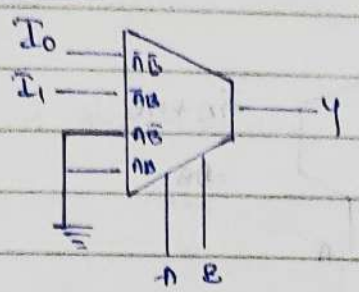
2×1 mux $\xrightarrow{63}$ 64×1 mux

2×1 mux $\xrightarrow{2^n-1}$ 2^n×1 mux

Q1. 4×1 MUX $\frac{8}{4} + \frac{8}{4}$ → 8×1 MUX

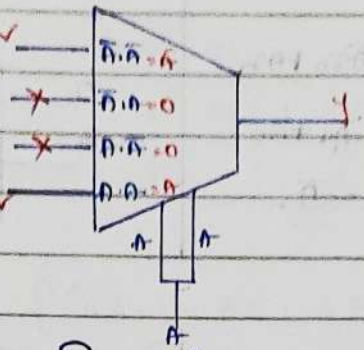
2 select line. $2+1=3$ 3 select line

Method 1



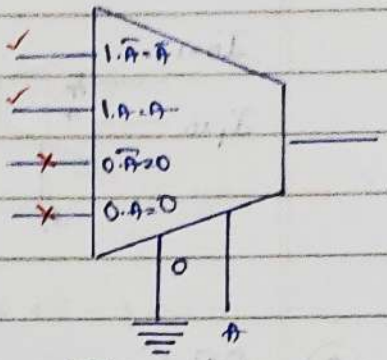
Select line = 2

Method 2



Select line = 1

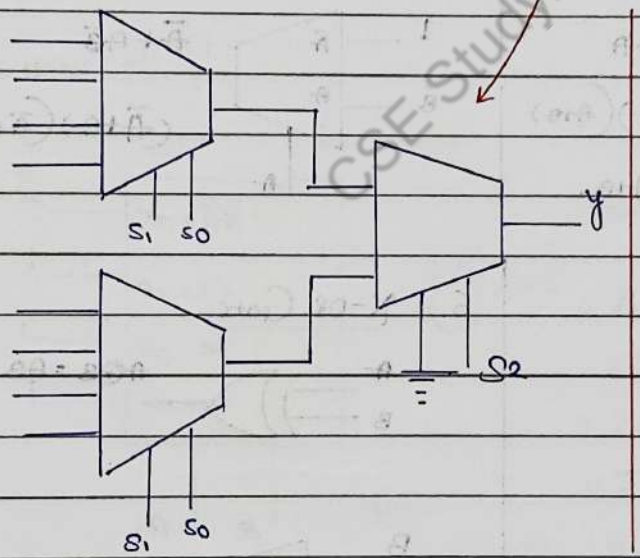
Method 3

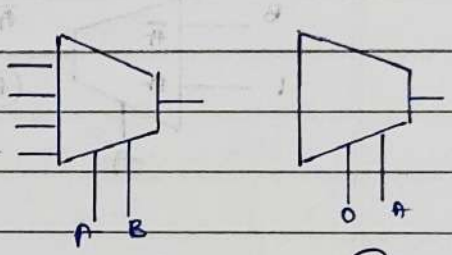


Select variable = One

4×1 MUX → 8×1 MUX

3 Select variable.

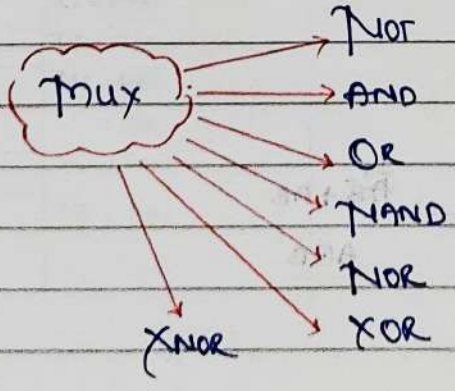


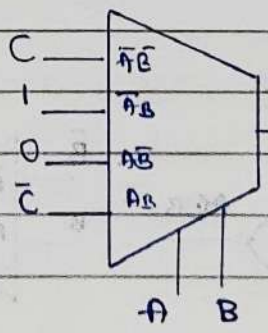


Select line = 2 Select line

Select variable = 2 Select val = 1.

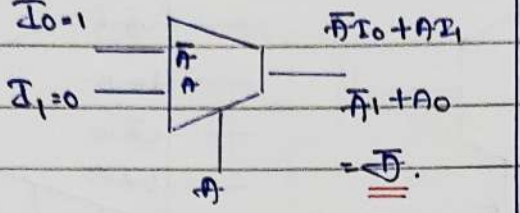
Type 2: MUX As a universal logic.



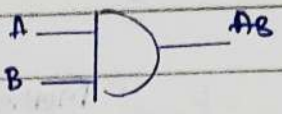


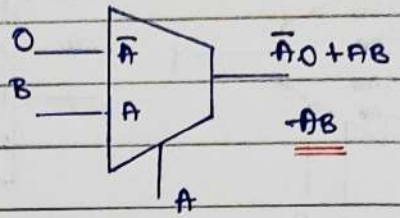
$y = \bar{A}\bar{B}C + \bar{A}B.1 + A\bar{B}\bar{C} + AB\bar{C}$.

(1.) NOR Gate

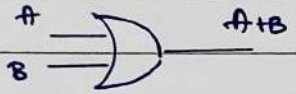


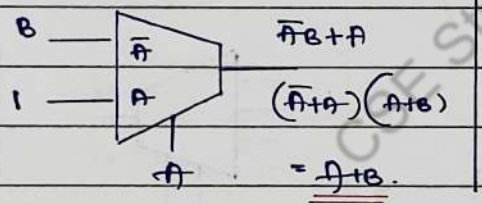
(2.) AND Gate



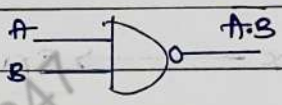


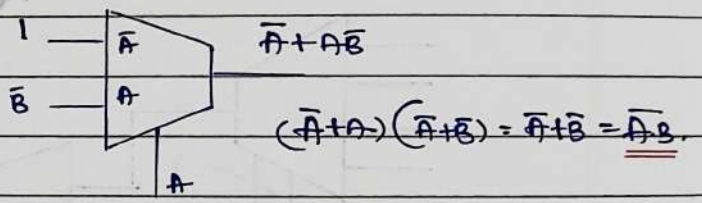
(3.) OR Gate



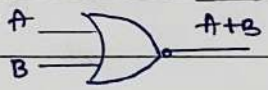


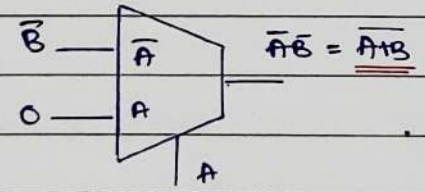
(4.) NAND Gate



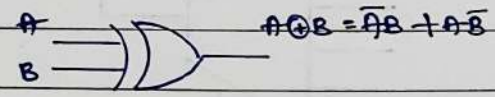


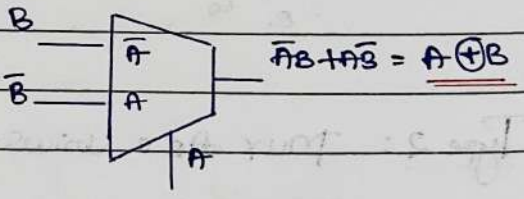
(5.) NOR Gate



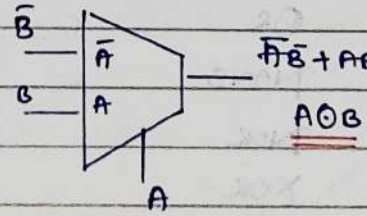


(6.) X-OR Gate





(7.) X-NOR Gate.

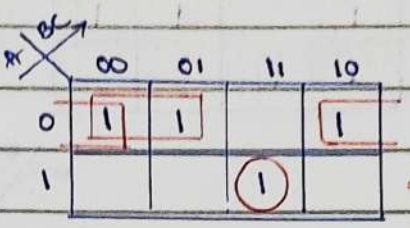


Date 29/6/23

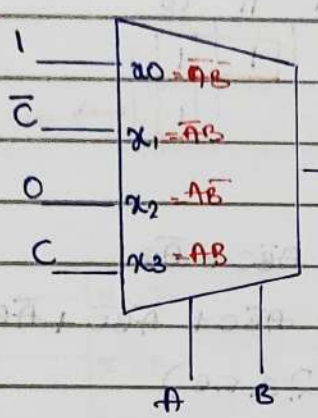
Note: To design NOT, AND, OR → One 2x1 MUX required
To design NAND, NOR, XOR, XNOR, → Two 2x1 MUX required.

Type 3 Minimization.

Q. Find the output f:

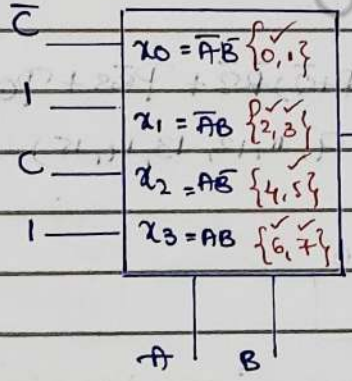


$\Rightarrow \bar{A}\bar{C} + \bar{A}\bar{B} + ABC$



$f(A,B,C) = \bar{A}\bar{B}.1 + \bar{A}B\bar{C} + A\bar{B}.0 + AB.C$
$= \bar{A}\bar{B} + \bar{A}B\bar{C} + ABC$
$= \bar{A}\bar{B}(\bar{C}+C) + \bar{A}B\bar{C} + ABC$
$= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + AB\bar{C} + ABC$
$= \sum m(0,1,2,7)$

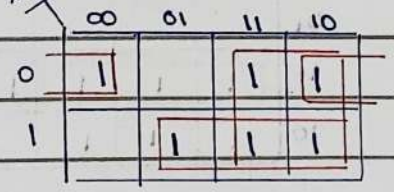
Q2. Find output of "f."



f(A,B,C)

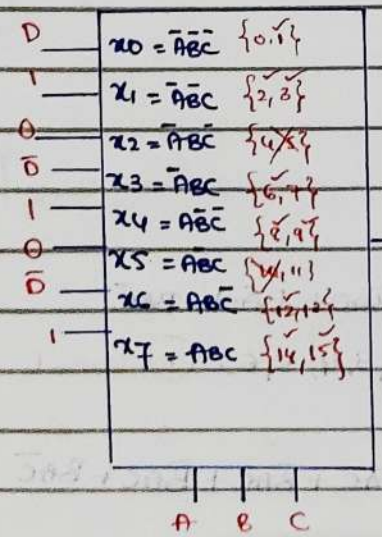
010=2 011=3 110=6 111=7
$\bar{A}B\bar{C}$ $\bar{A}BC$ $AB\bar{C}$ ABC

$= \bar{A}B\bar{C} + \bar{A}B + AB\bar{C} + AB$
$= \bar{A}B\bar{C} + \bar{A}B(\bar{C}+C) + AB\bar{C} + AB(\bar{C}+C)$
$= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + \bar{A}BC + AB\bar{C} + ABC$
$= \sum m(0,2,3,5,6,7)$



$\Rightarrow B + \bar{A}\bar{C} + AC$

Q3.



$f(A,B,C,D) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C} + A\bar{B}C\bar{D} + ABC$
$= \sum m(1,2,3,6,8,9,12,14,15)$

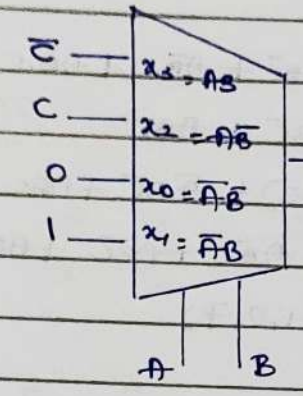
AB \ CD	00	01	11	10
00		1	1	1
01				1
11	1		1	1
10	1	1		

(spain)

Q.4 find f = ?

A \ BC	$\bar{B}\bar{C}$ (00)	$\bar{B}C$ (01)	BC (11)	$B\bar{C}$ (10)
$\bar{A}$ 0			1	1
A 1		1		1

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



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

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

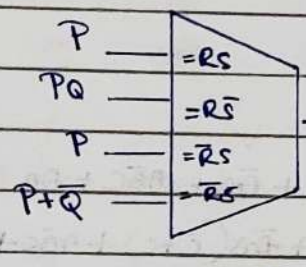
$= \sum m(2,3,5,6)$

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

$P\bar{Q}\bar{R}\bar{S} = 1000$

$PQ\bar{R}\bar{S} = 1100$

Q.5



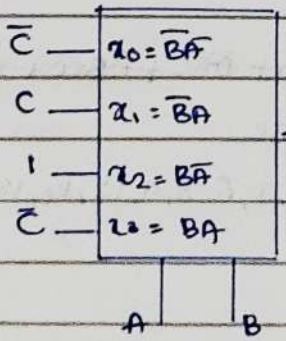
$f(P,Q,R,S) = (P+\bar{Q})\bar{R}\bar{S} + P\bar{R}S + PQR\bar{S} + PRS$

$= \sum m(0,8,9,11,12,13,14,15)$

PQ \ RS	00	01	11	10
00	1			
01				
11	1	1	1	1
10	1		1	1

$\Rightarrow PQ + PS + \bar{Q}\bar{R}\bar{S}$

Q.6.



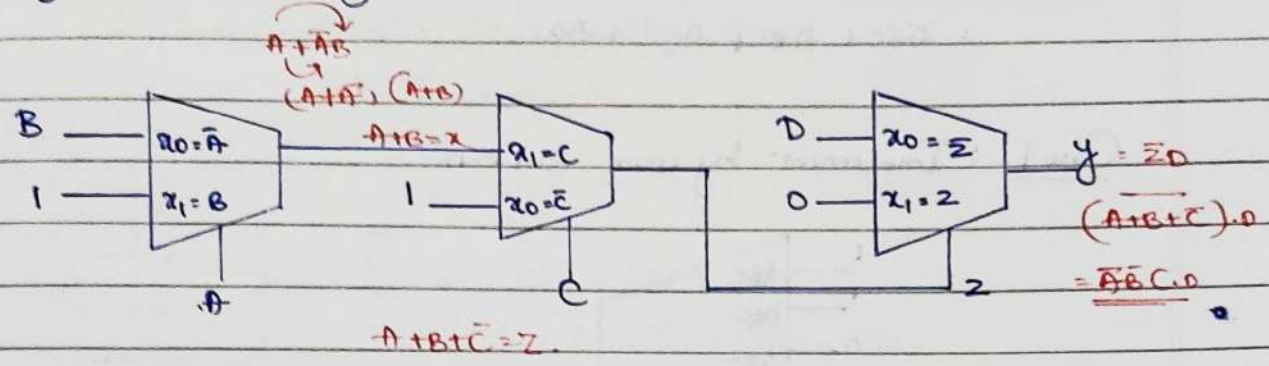
$f(A,B,C) = \bar{B}\bar{A}\bar{C} + \bar{B}AC + B\bar{A}\bar{C} + BA\bar{C}$

$= \sum m(0,3,4,5,6)$ (1) Method

$f(A,B,C) = \bar{B}\bar{A}\bar{C} + B\bar{A}C + \bar{B}A\bar{C} + \bar{B}AC + BA\bar{C}$ (2) Method

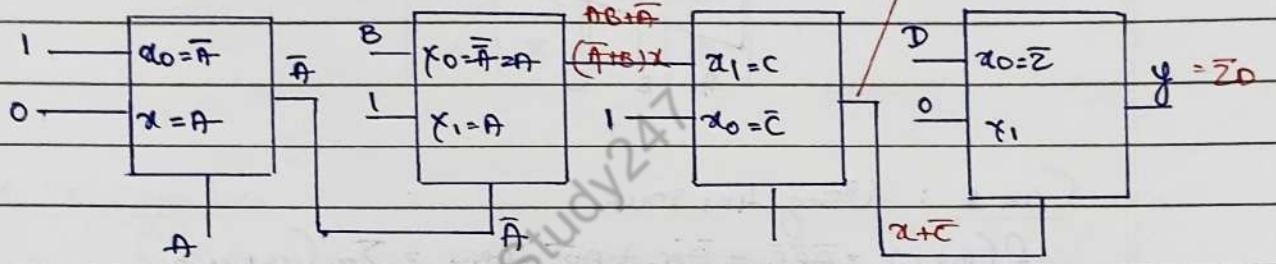
(0 5 2 3 6)

Type 4: Cascading of mux:



Q1. Find the output of y. $\bar{A}+AB = (\bar{A}+B)(\bar{A}+A)$ $= \bar{A}+B$

$xC+\bar{C} = (x+\bar{C})(C+\bar{C})$ $= x+\bar{C}$.



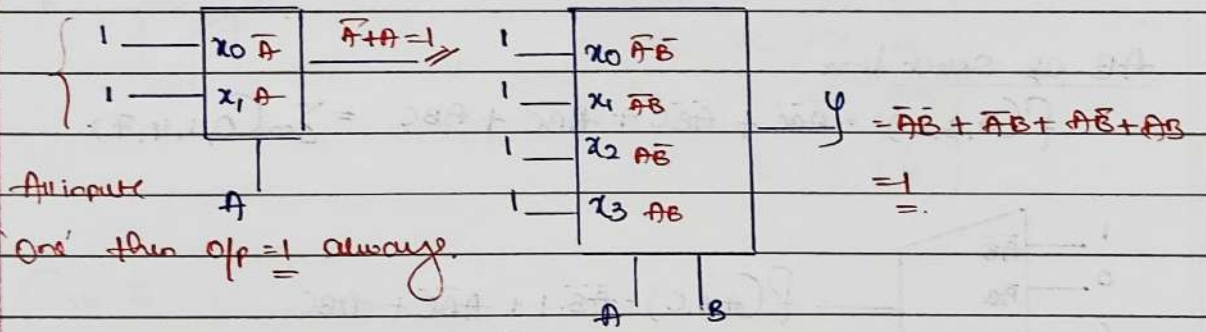
x+C

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

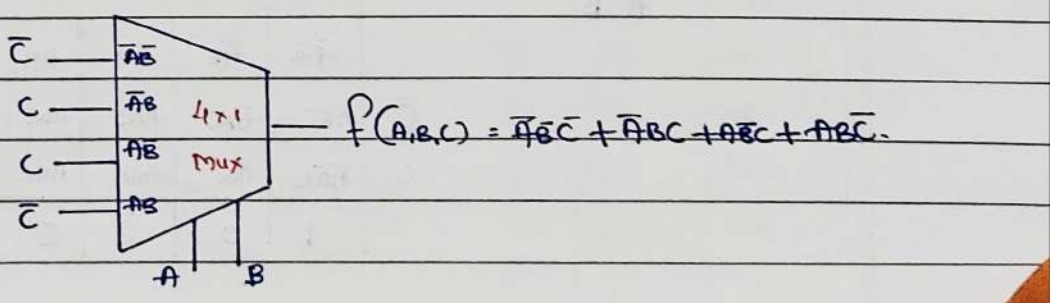
$y = \bar{z}D = (\bar{A}+B+\bar{C})D = \bar{A}\bar{B}CD$

NAND = 7

Note:



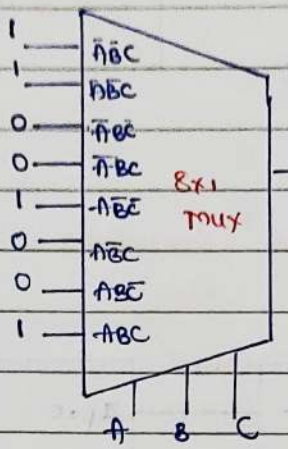
Type 5: Implementation of Function:



Ex 1. $f(A,B,C) = \sum m(0,1,4,7)$

$= \bar{A}\bar{B}C.1 + \bar{A}BC + A\bar{B}\bar{C} + ABC.$

Case I : Implement by using 8x1 mux



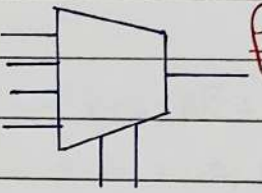
$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

$= \sum m(0,1,4,7)$

Case II : Using 4x1 mux

$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC = \sum m(0,1,4,7)$

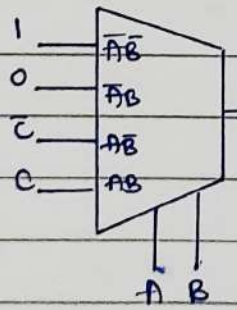
1. AB as select line
2. BC as select line
3. AC as select line



$f(A,B,C) = \sum m(0,1,4,7)$

1. AB as Select line

$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC = \sum m(0,1,4,7)$



$f(A,B,C) = \bar{A}\bar{B}.1 + A\bar{B}\bar{C} + ABC$

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

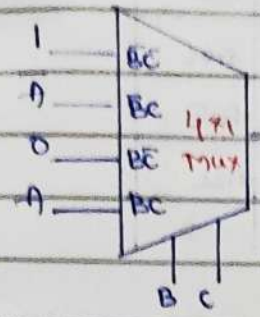
$= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
$\bar{C}$	$\bar{A}\bar{B}\bar{C}$ ⓪	$\bar{A}B\bar{C}$ 2	$A\bar{B}\bar{C}$ ④	$AB\bar{C}$ 6
C	$\bar{A}\bar{B}C$ ①	$\bar{A}BC$ 3	$A\bar{B}C$ 5	ABC ⑦
	1	0	$\bar{C}$	C

2. BC as Select Line

$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= \Sigma m(0,1,4,7)$



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

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

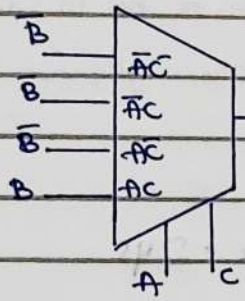
$= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$

	$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	BC
$\bar{A}$	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}B\bar{C}$ 2	$\bar{A}BC$ 3
A	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	$AB\bar{C}$ 6	ABC 7
	1	$\bar{A}$	0	A

3. AC is Select line

$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= \Sigma m(0,1,4,7)$



$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

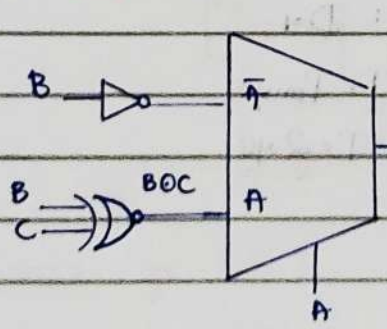
	$\bar{A}\bar{C}$	$\bar{A}C$	$A\bar{C}$	AC
$\bar{B}$	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5
B	$\bar{A}B\bar{C}$ 2	$\bar{A}BC$ 3	$AB\bar{C}$ 6	ABC 7
	$\bar{B}$	$\bar{B}$	$\bar{B}$	B

Case III: Using 2×1 Mux

(i) A as Select line

$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$

$= \Sigma m(0,1,4,7)$



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

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

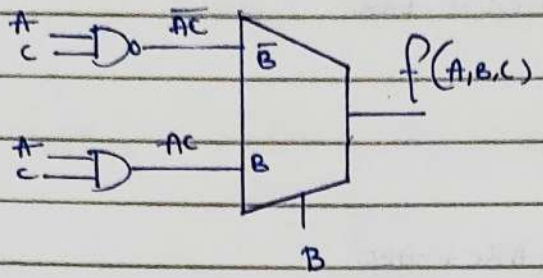
$+ABC$

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

$f(A,B,C) + ABC$

	$\bar{A}$	A
$\bar{B}\bar{C}$	$\bar{A}\bar{B}\bar{C}$ 0	$A\bar{B}\bar{C}$ 4
$\bar{B}C$	$\bar{A}\bar{B}C$ 1	$A\bar{B}C$ 5
$B\bar{C}$	$\bar{A}B\bar{C}$ 2	$AB\bar{C}$ 6
BC	$\bar{A}BC$ 3	ABC 7
	$\bar{B}$	$B \odot C$

(ii) B as Select line



$f(A,B,C) = \sum m(0,1,4,7)$

	$\bar{B}$	B
$\bar{A}\bar{C}$	$\bar{A}\bar{B}\bar{C}$ ⓪	$\bar{A}B\bar{C}$ 2
$\bar{A}C$	$\bar{A}\bar{B}C$ ①	$\bar{A}BC$ 3
$A\bar{C}$	$A\bar{B}\bar{C}$ ④	$AB\bar{C}$ 6
AC	$A\bar{B}C$ 5	ABC ⑦
	$\bar{A}C + A\bar{C}$	AC

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

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

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

(iii) C as Select line $f(A,B,C) = \sum m(0,1,4,7)$

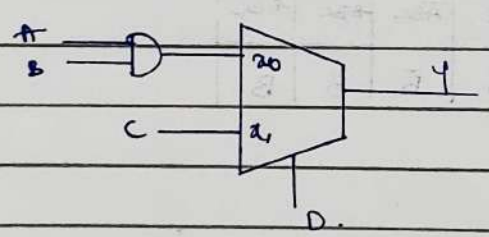
Q1. $f(A,B,C,D) = \sum m(0,1,3,5,7,9,12,15)$

1. ABD is Select line → 16x1 mux, 8x1 mux, 4x1 mux
2. ACD is Select line → 2x1 mux

(AB, AC, AD as Select line

A as Select line.

Type-6 : Delay in mux



$T_{AND} = 1\mu s$, $T_{mux} = 2\mu s$

Case 1 : D=0

$T = T_{AND} + T_{mux}$

$= 1\mu s + 2\mu s$

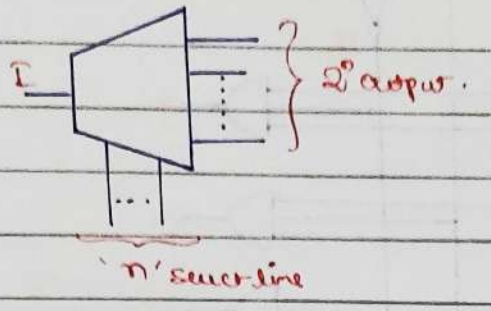
$T = 3\mu s$

Case 2 : D=1

$T = T_{mux}$

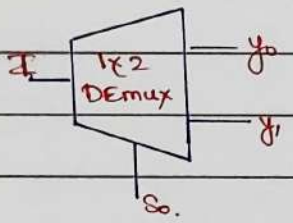
$T = 2\mu s$

DE-mux: DE-mux is called AND Logic



Q1. Design 1x2 DE-mux

Step1:



Step2:

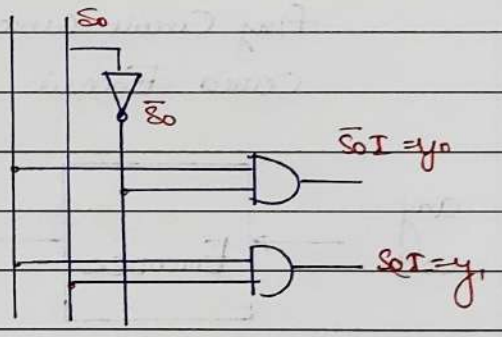
S_0	y_0	y_1
0	I	0
1	0	I

Step3: $y_0 = \bar{S}_0 I$

$y_1 = S_0 I$

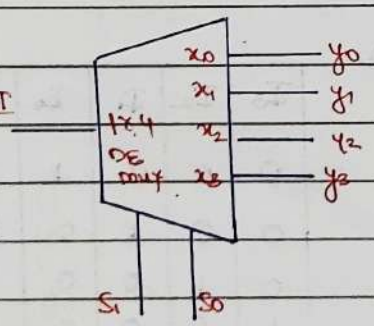
Step4:

Step5:



Q2. Design a 1x4 DEmux?

Step1:



Step2:

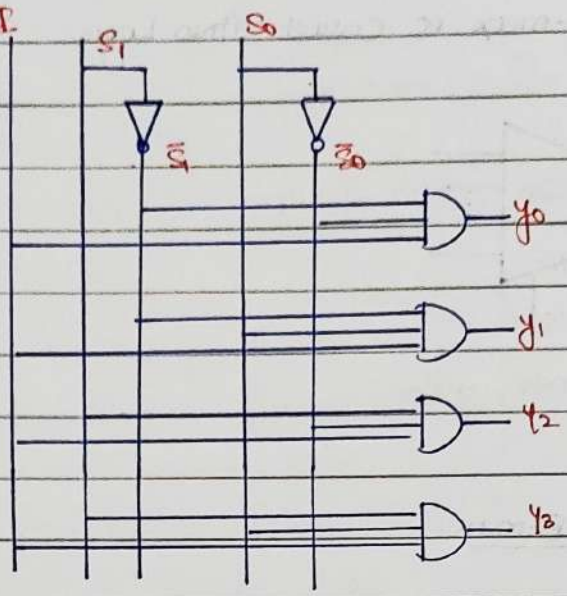
S_1	S_0	y_0	y_1	y_2	y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

Step3: $y_0 = \bar{S}_1 \bar{S}_0 I$ $y_2 = S_1 \bar{S}_0 I$

$y_1 = \bar{S}_1 S_0 I$ $y_3 = S_1 S_0 I$.

Step4:

Steps:



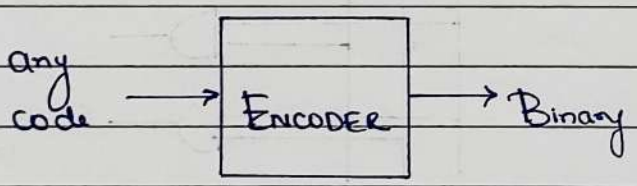
HW

Q. Design a 1x8 DE MUX

Q. Design a 1x16 DE MUX.

Encoder:

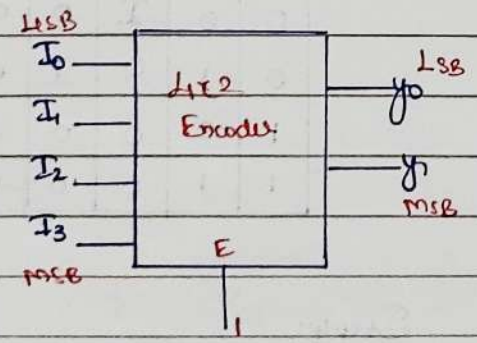
Any Circuit which is used to convert any code into Binary is called Encoder.



1> 4x2 Encoder (Quad to Binary)
2> 8x3 Encoder (Octa to Binary)
3> 16x4 Encoder (Hexa to Binary).

Q. Design a 4x2 Encoder.

Step1:



Step2:

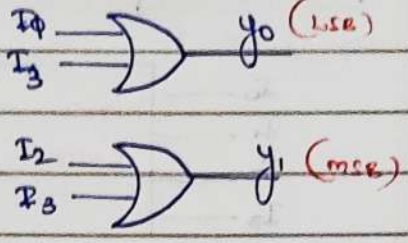
I_3	I_2	I_1	I_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Step 3: $y_0 = I_1 + I_3$

$y_1 = I_2 + I_3$

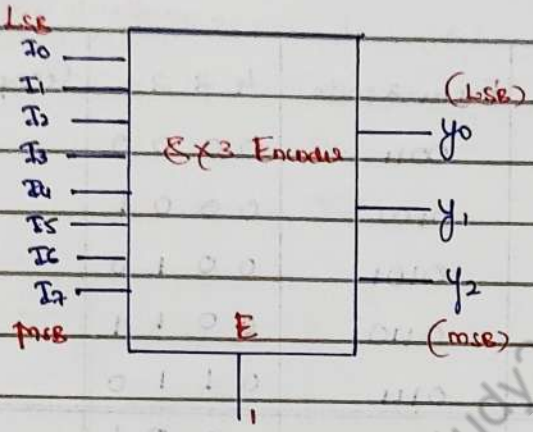
Step 4:

Step 5:



Q. Design a 8x3 Encoder.

Step 1:



HW

Q. Design a 16x4 Encoder

Step 2:

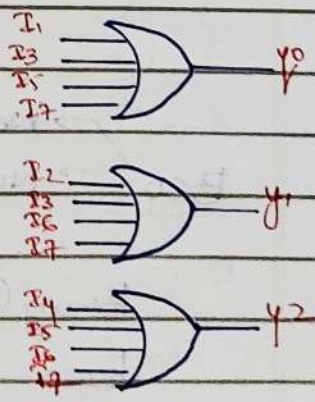
I_7	I_6	I_5	I_4	I_3	I_2	I_1	B	y_2	y_1	y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Step 3: $y_0 = I_1 + I_3 + I_5 + I_7$

$y_1 = I_2 + I_3 + I_6 + I_7$

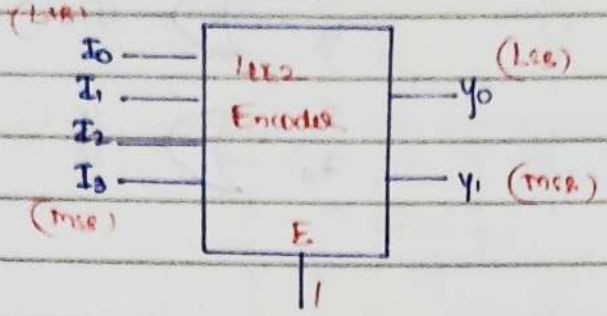
$y_2 = I_4 + I_5 + I_6 + I_7$

Step 5:



Step 4:

Note: Priority Encoder



LSB Priority Encoder.

(MSB) I_3	I_2	I_1	I_0 (LSB)	y_1	y_0
x	x	x	1	0	0
x	x	1	0	0	1
x	1	0	0	1	0
1	0	0	0	1	1

↗ weighted, Self-complimented

Decimal	BCD	Excess-3 code	4 2 2 1	Gray Code
0	0000	0011	0 0 0 0	
1	0001	0100	0 0 0 1	
2	0010	0101	0 0 1 0	
3	0011	0110	0 0 1 1	
4	0100	0111	0 1 1 0	
5	0101	1000	1 0 0 1	
6	0110	1001	1 1 0 0	
7	0111	1010	1 1 0 1	
8	1000	1011	1 1 1 0	
9	1001	1100	1 1 1 1	

Decimal: → Weighted code
→ Self Complimented

10^3 10^2 10^1 1
a_3 a_2 a_1 a_0

r → Base (radix)

Compliment
→ $(r-1)$'s
→ r's complement

	$(r-1)$'s	r's complement
$r=2$	1's	2's
$r=8$	7's	8's
$r=10$	9's	10's

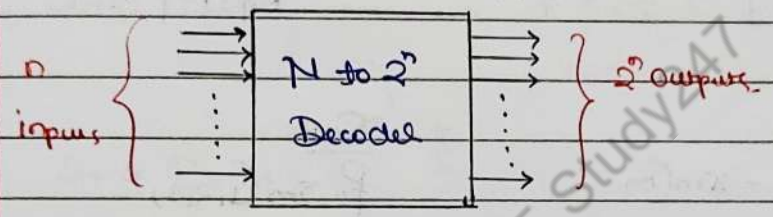
BCD → Weighted Code
→ Not a Self Complimentary Code.

→ Binary Coded Decimal
→ Each decimal numbers are represented by 4 bits.

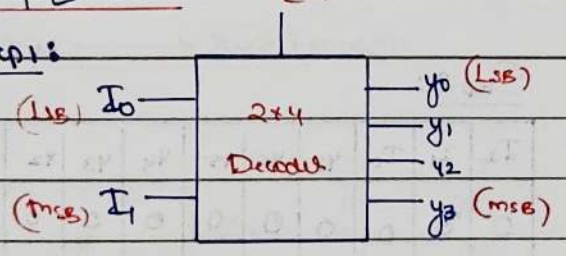
Eg: { Decimal - 0, BCD - 0000 } { Decimal - 9, BCD - 1001 } { Decimal - 29, BCD - 00101001 }

Excess - 3 code:
- It is not a weighted code
- Self complimented code

Decoder: Circuit which is used to convert Binary into any other code.
1. 2×4 Decoder
2. 3×8 Decoder
3. 4×16 Decoder.



2×4 Decoder.



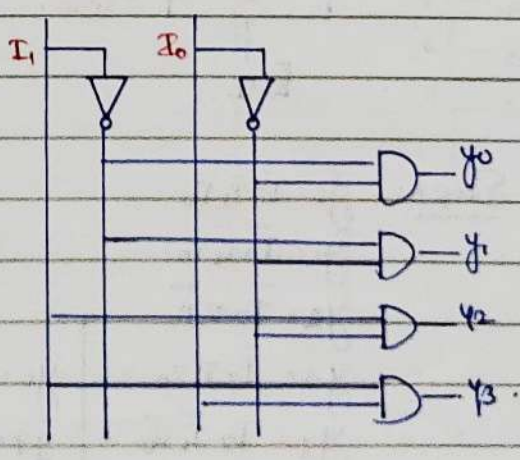
Step 2:

I_1	I_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
ϕ	1	1	0	0	0

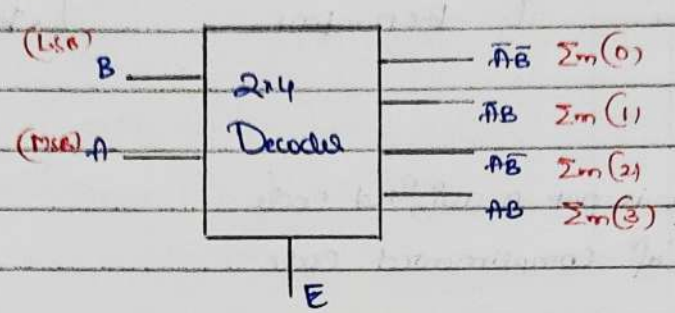
Step 3: $y_0 = \bar{I}_1 \bar{I}_0$
$y_1 = \bar{I}_1 I_0$
$y_2 = I_1 \bar{I}_0$
$y_3 = I_1 I_0$.

Step 4: minimization

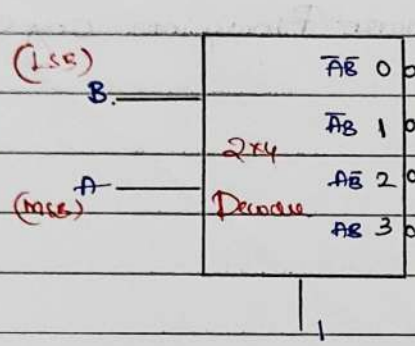
Step 4:



Example:



eg:



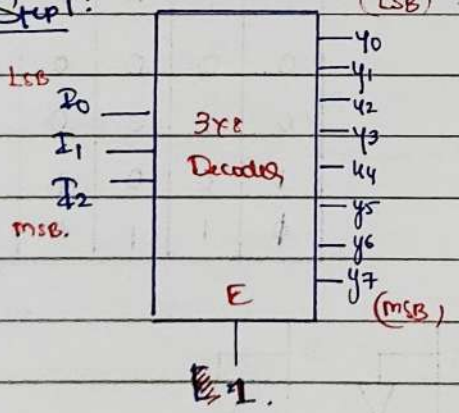
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

POS = $F = (A+B) = \pi m(0)$

SOP $F = \Sigma m(1,2,3)$

3×8 Decoder

Step 1:



Step 2:

I_2	I_1	I_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Step 3: $y_0 = \bar{I}_2\bar{I}_1\bar{I}_0$

$y_1 = \bar{I}_2\bar{I}_1 I_0$

$y_2 = \bar{I}_2 I_1 \bar{I}_0$

$y_3 = \bar{I}_2 I_1 I_0$

$y_4 = I_2\bar{I}_1\bar{I}_0$

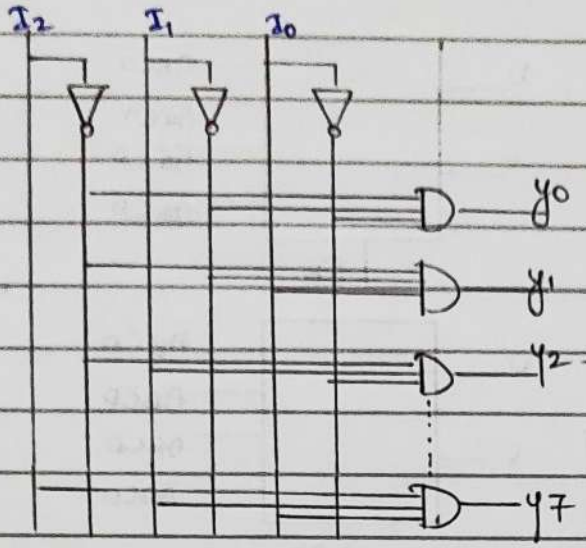
$y_5 = I_2\bar{I}_1 I_0$

$y_6 = I_2 I_1 \bar{I}_0$

$y_7 = I_2 I_1 I_0$

Step 4: Minimization.

Steps : Hardware Implementation.



H/W

Design 4x16 Decoder

$\frac{8}{4} = 2^{\frac{m}{n}}$

Q.1 2x(4) Decoder ⟶ 3x(8) Decoder

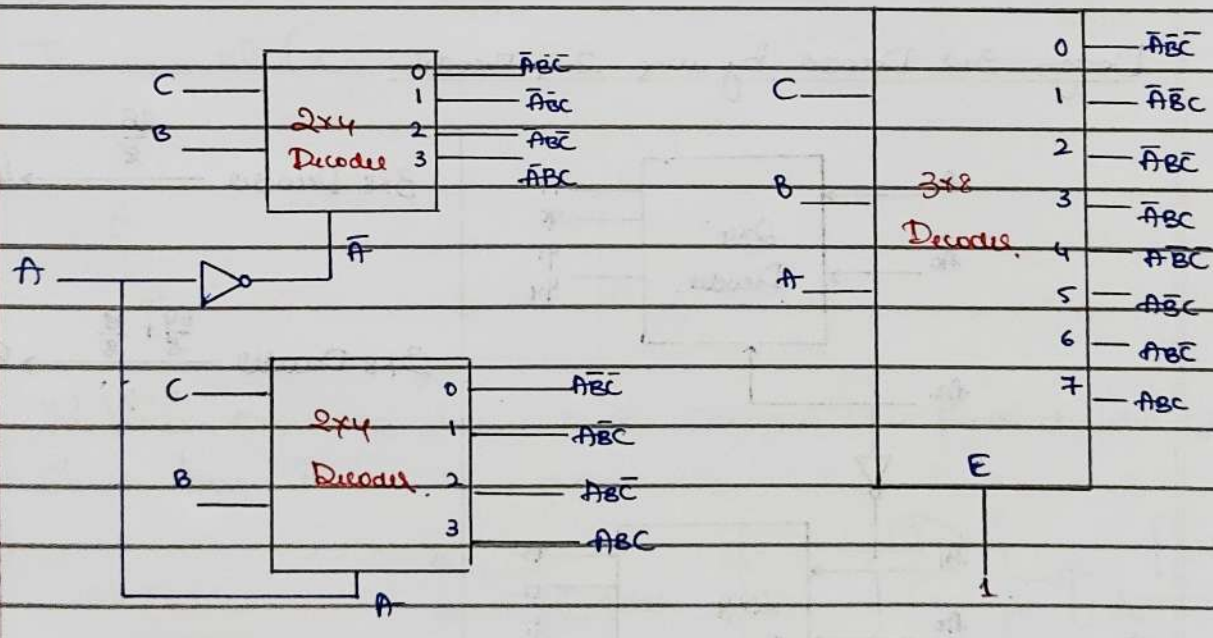
input, output

Q.2 2x4 Decoder. ⟶ 4x16 Decoder.

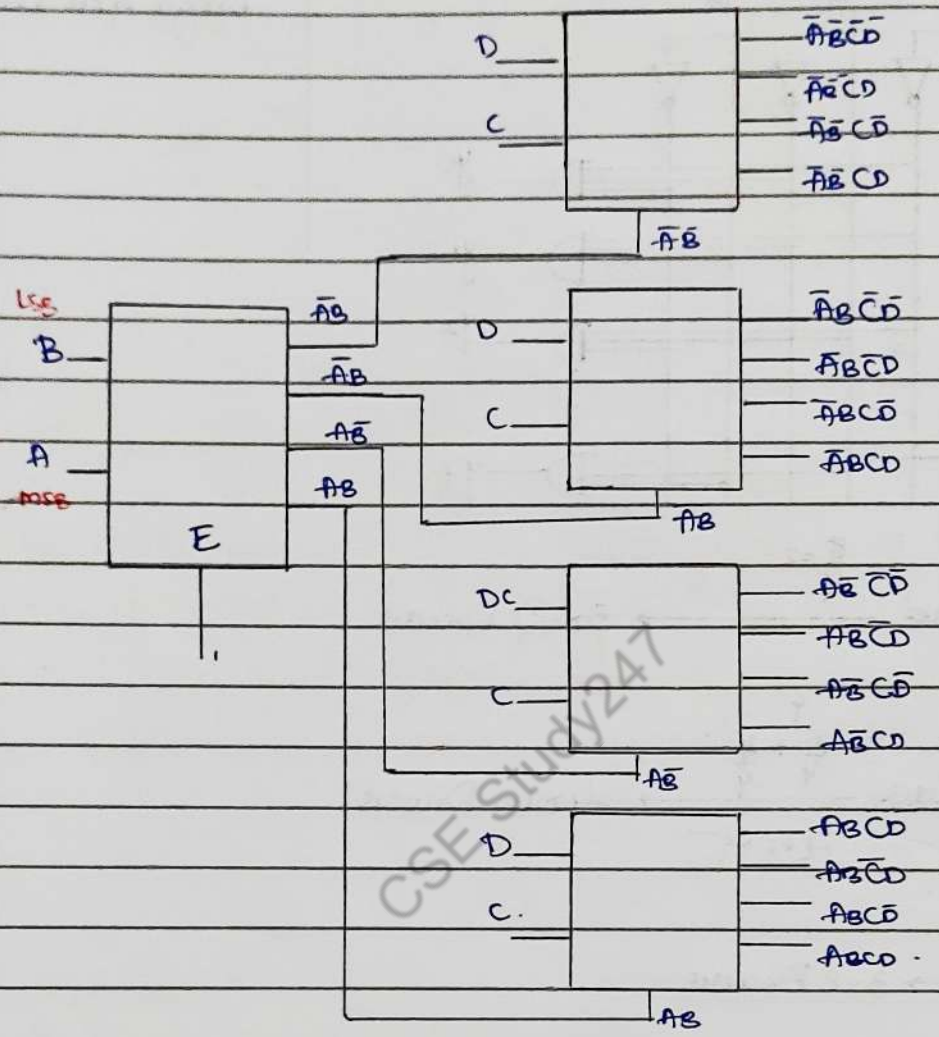
$\frac{16}{4} + \frac{1}{4}$

$4+1=5$

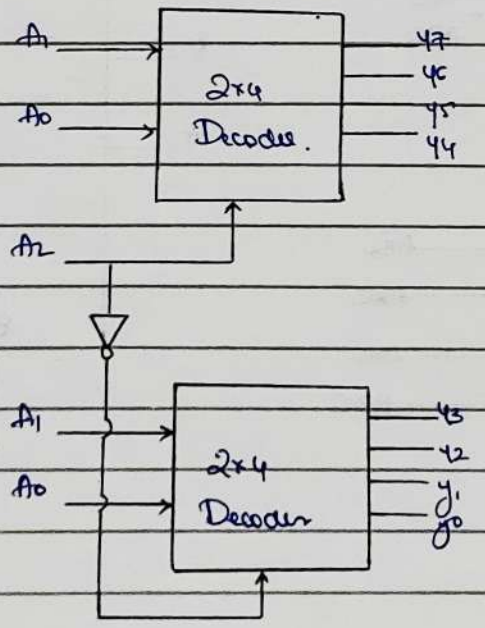
2x4 ⟶ 3x8 Decoder



2×4 Decoder ⟶ 4×16 Decoder



Design 3×8 Decoder by using 2×4 Decoder



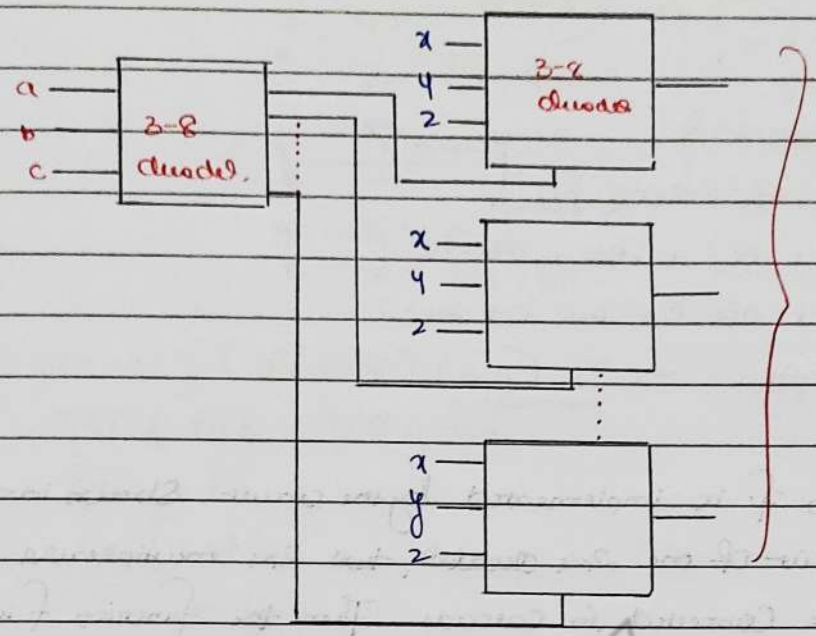
3×8 Decoder $\xrightarrow{\frac{16}{8}}$ 4×16 Decoder

$= 2$

3×8 Decoder $\xrightarrow{\frac{64}{8}+\frac{8}{8}}$ 6×64 Decoder

$8+1 = 9$

3×8 Decoder ——→ 6×64 Decoder.



Hw.
Design Binary to decimal decoder.

Q.1 A logic circuit consist of 2×4 decoder as shown in the figure. The output are as follows:

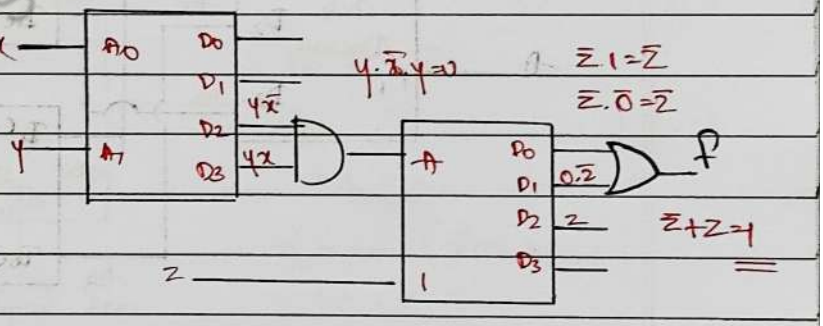
$D_0 = 1$ when $A_0 = 0, A_1 = 0$

$D_1 = 1$ when $A_0 = 1, A_1 = 0$

$D_2 = 1$ " $A_0 = 0, A_1 = 1$

$D_3 = 1$ " $A_0 = 1, A_1 = 1$.

The value of (x, y, z) is.



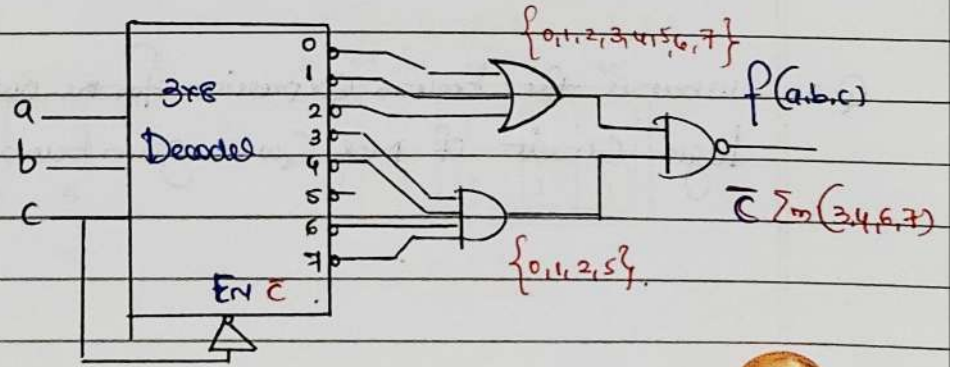
Ans : 1

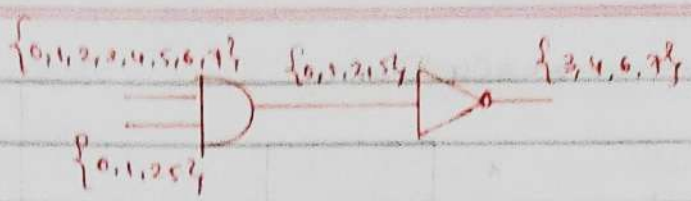
A_1	A_0	y
0	0	$D_0 = 1$
0	1	D_1
1	0	D_2
1	1	D_3.

Q2 The boolean expression $f(a,b,c)$ in its canonical form for the decoder circuit shown below is

Ans.

$\Rightarrow \sum m(4,6)$





$f = C.\Sigma m(3,4,6,7)$

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

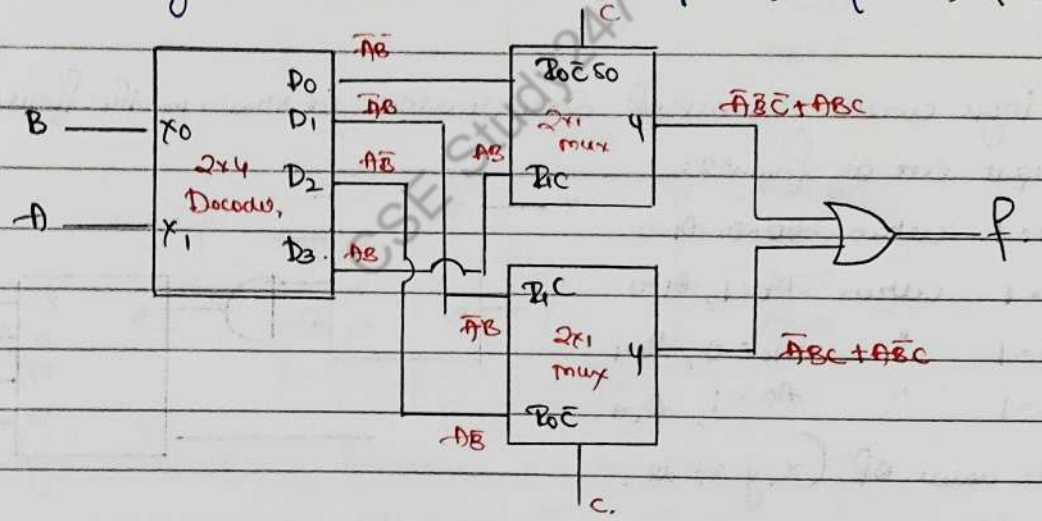
$= \bar{C}[\bar{A}BC + AB\bar{C} + A\bar{B}C + ABC]$

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

$= AB\bar{C} + A\bar{B}C \Rightarrow \Sigma m(4,6)$

Q3. A logic function 'f' is implemented by the circuit shown in the figure below. The circuit consist of one 2×4 decoder, two 2×1 multiplexer and a two input OR gate connected in cascade. Then the function f is equal to

Ans



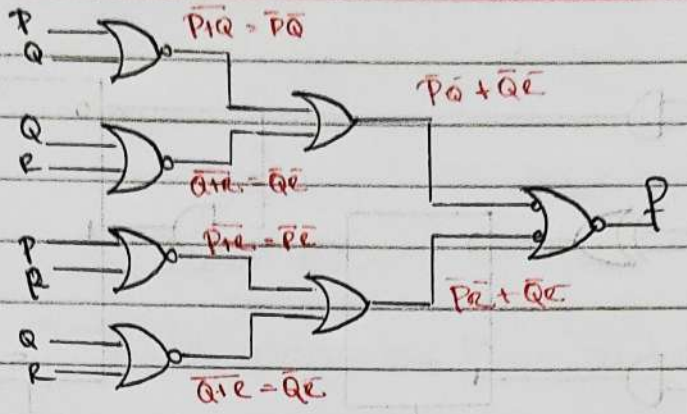
$f = \bar{A}B\bar{C} + ABC + \bar{A}BC + A\bar{B}\bar{C}$

$= \Sigma m(0,3,4,7)$

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

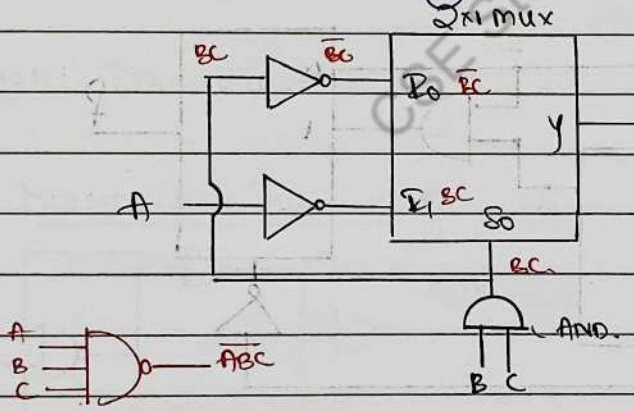
$\Rightarrow \bar{B}\bar{C} + BC = B \odot C$

Q2. What is the boolean expression for the output f of the combinational logic circuit of NOR gates given below.



$$f = [\bar{P}\bar{Q}+\bar{Q}\bar{R}]\cdot[\bar{P}\bar{R}+\bar{Q}\bar{R}]$$
$$= \bar{P}\bar{Q}\bar{R} + \bar{P}\bar{Q}\bar{R} + \bar{P}\bar{Q}\bar{R} + \bar{Q}\bar{R}$$
$$= \bar{P}\bar{Q}\bar{R} + \bar{Q}\bar{R}$$
$$= \bar{Q}\bar{R}[1+\bar{P}]$$
$$= \bar{Q}\bar{R} \Rightarrow \overline{Q+R}$$

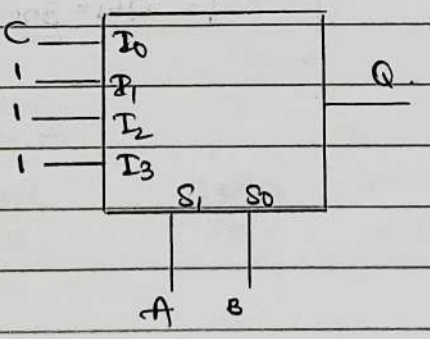
Q4. The combinational circuit given below implements which of the following



$= \bar{BC}\cdot\bar{BC} + \bar{A}BC$
$= \bar{BC} + \bar{A}BC$ $\quad \{BC = x$
$= \bar{x} + \bar{A}x$
$= (\bar{x}+\bar{A})(\bar{x}+x)$
$= \bar{BC} + \bar{A}$
$= \overline{BCA}$

$\therefore$ NAND Gate

Q5. The combinational logic circuit shown in the given figure has an output Q which is

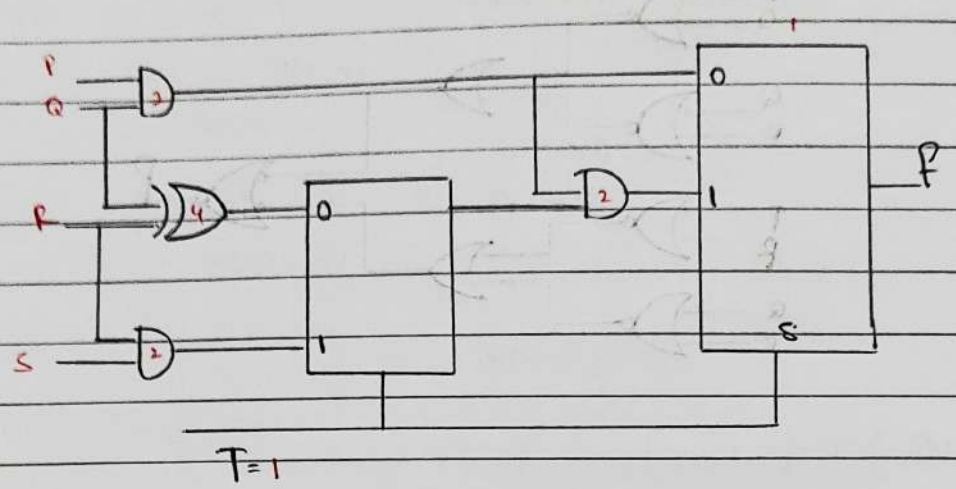


$= \bar{A}\bar{B}C + \bar{A}B + A\bar{B} + AB$
$= \sum m(1,2,3,4,5,6,7)$

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

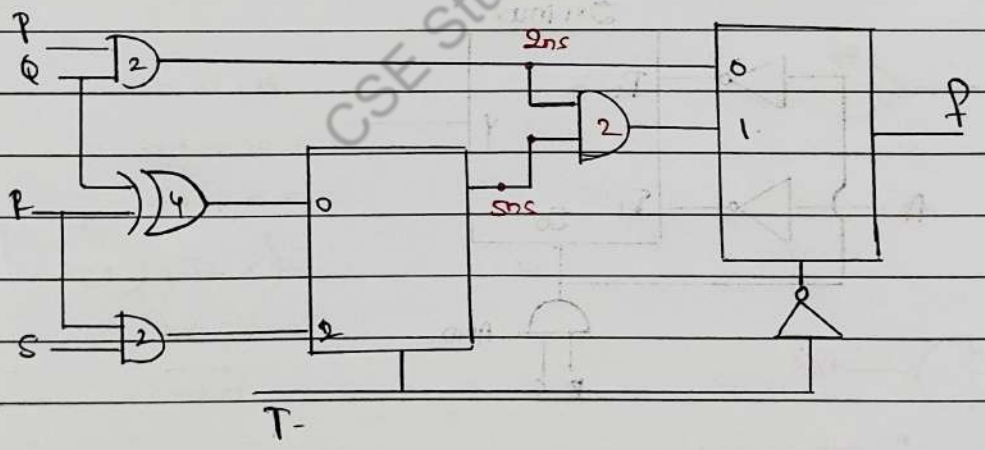
$\Rightarrow A+B+C$

Q6.



Case: T=0	Case 2 : T=1
$T = T_{AND} + T_{mux2}$	$T = T_{AND} + T_{mux} + T_{AND} + T_{mux}$
$T = 2+1 = 3ns$	$= 2+1+2+1 = 6ns$
	$\therefore T_{max} = 6ns$

Q7.



Case (1) : T=0	Case (2) : T=1
$T = T_{XOR} + T_{mux} + T_{AND} + T_{mux}$	$T = T_{AND} + T_{mux}$
$= 4+2+1+1 = 8ns$	$T = 2+1 = 3ns$

Date 4/7/2023

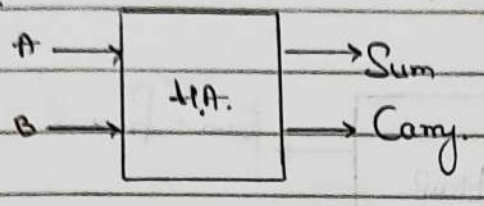
Half Adder

0	0	1	1
+0	+1	+0	+1
00	01	01	10

```
  0
 +0
 --
 00
 ↓  ↓
Carry. Sum
```

* Two bit adder are known as half adder.

Step 1:



Step 2:

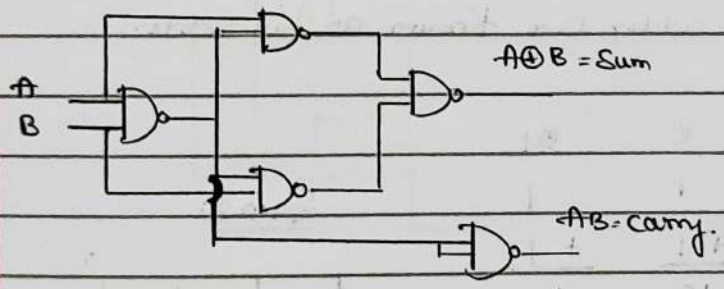
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Step 3: $Sum = A\bar{B} + \bar{A}B = A \oplus B$

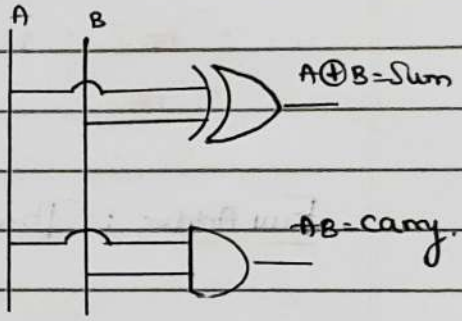
$Carry = AB$

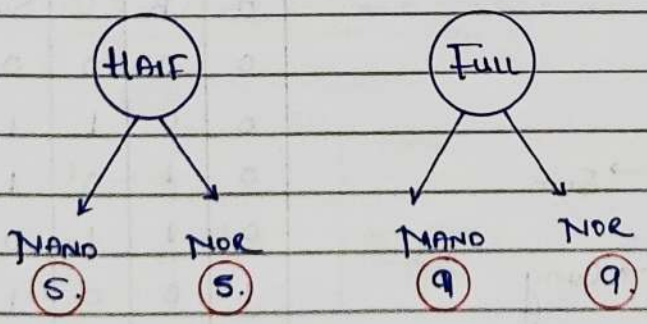
Step 4: Minimization.

NAND Gate Required = 5

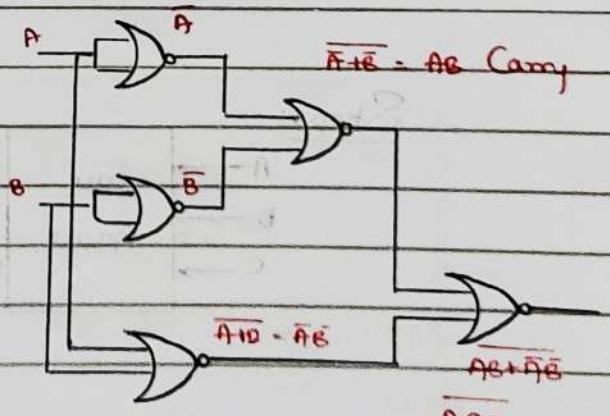


Step 5:

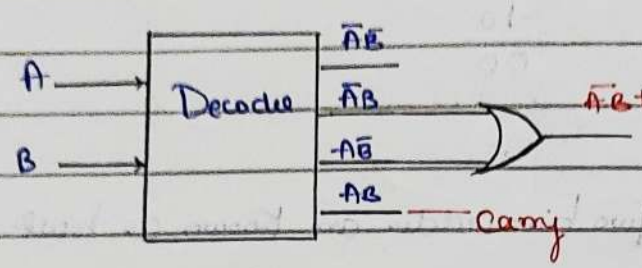




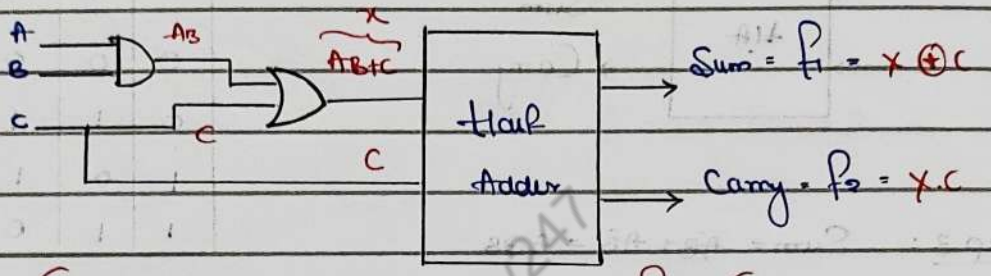
By NOR Gate



Half Adder By Using Decoder.



Q1. What is the value of f_1 and f_2?



$f_1 = (AB+C) \oplus C$

$= \overline{AB+C}.C + (AB+C)\bar{C}$

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

$= \underline{\underline{AB\bar{C}}}$

$f_2 = (AB+C)C$

$= ABC + C.C = ABC + C$

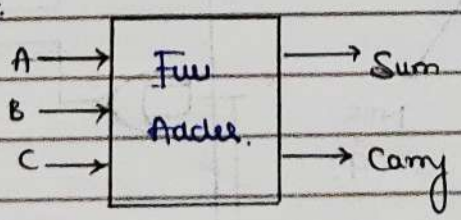
$= C.[AB+1]$

$= \underline{\underline{C.AB}}$

Full Adder : Three bit adder are known as full adder.

0	0	0	01
0	0	1	1
+0	+1	+1	+1
00	01	10	11

Step1:



Step2:

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

Step3: $\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$ (Sum).

$= \sum m(1,2,4,7)$

$= A \oplus B \oplus C$

Step4: $Carry(A,B,C) = \sum m(3,5,6,7)$

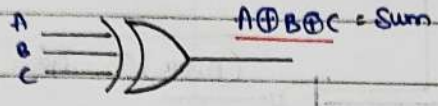
$= \bar{A}BC + AB\bar{C} + ABC + ABC$

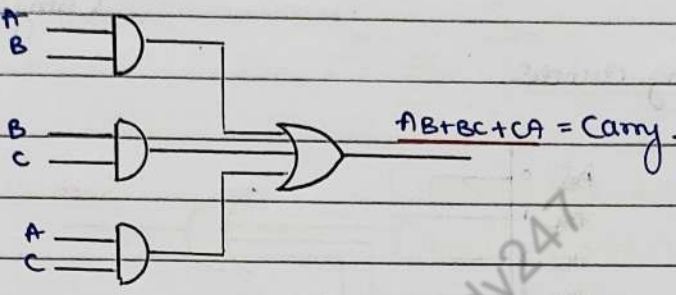
$= AB + BC + AC$ – minimised exp.

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

$= (A \oplus B)C + AB$ – Semi minimised.

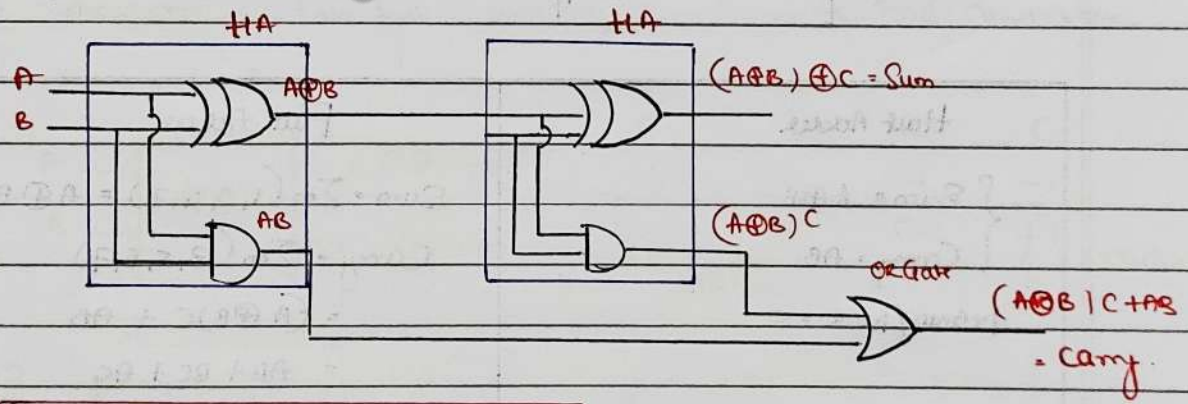
Step5:



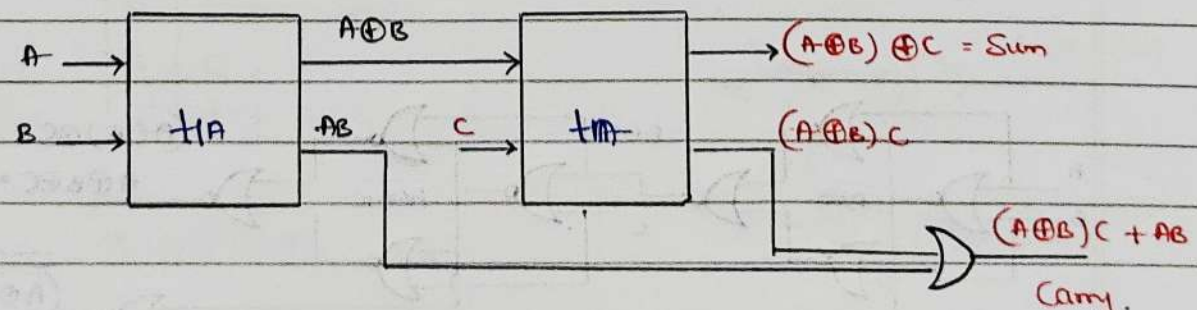


Sum = $(A \oplus B) \oplus C$

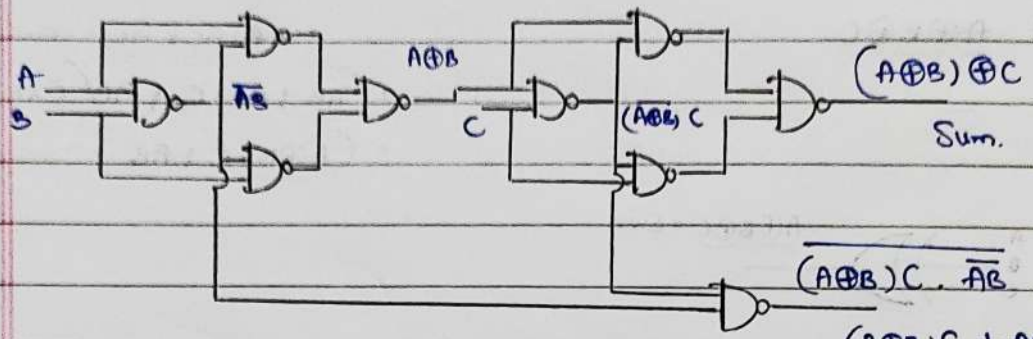
Carry = $(A \oplus B)C + AB$.



1 Full Adder = 2 H.A + 1 OR Gate



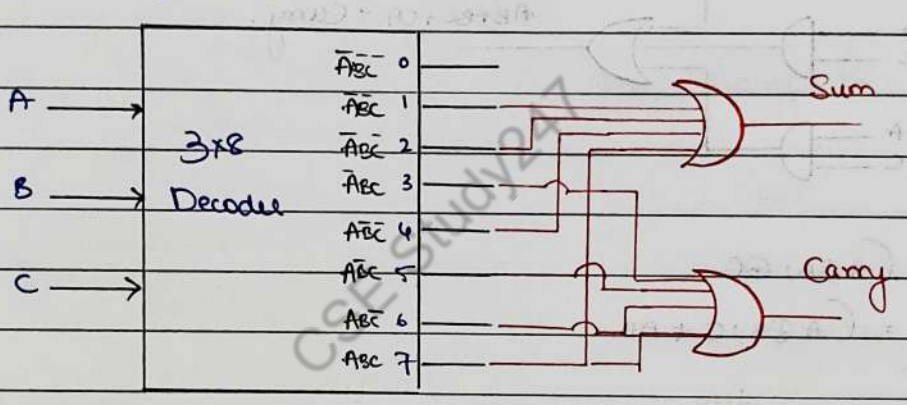
Full Adder by using NAND Gate.



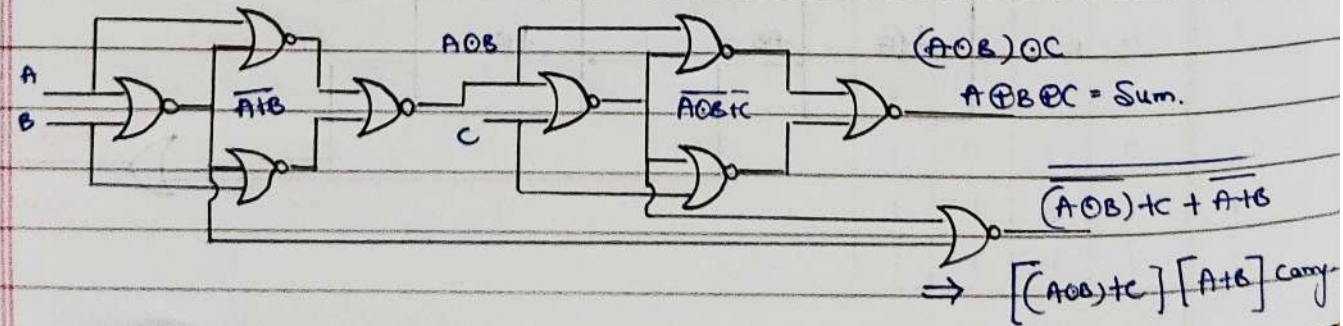
HW

Full Adder by using NOR GATE.

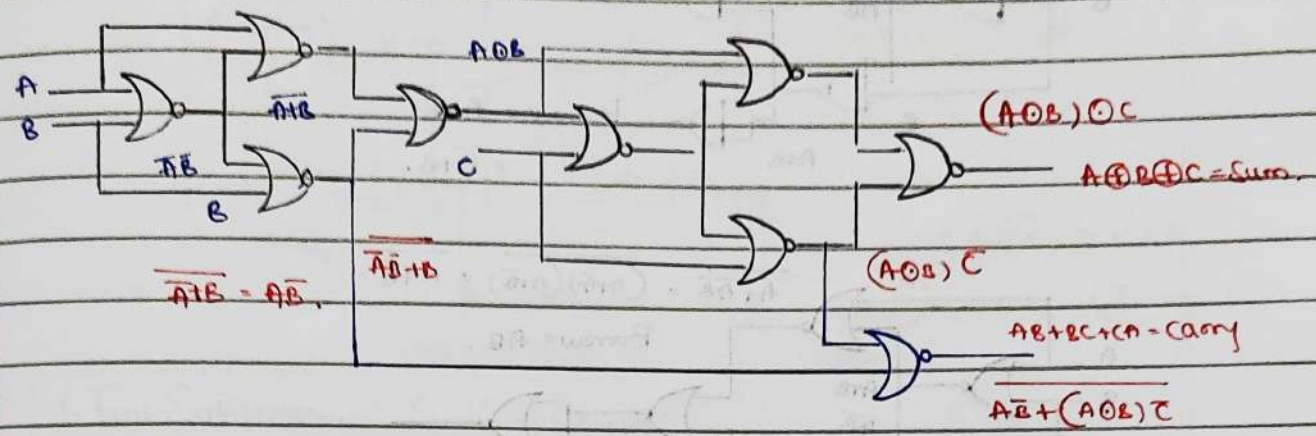
Full Adder by using decoder.



Half Adder	Full Adder
Sum = A⊕B Carry = AB NAND/NOR = 5.	$Sum = \Sigma m(1,2,4,7) = A\oplus B\oplus C$ $Carry = \Sigma m(3,5,6,7)$ $= (A\oplus B)C + AB$ $= AB + BC + AC$ NAND/NOR = 9. 1 Full = 2 HA + 1 OR



Full Subtractor



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

$$= \overline{A\bar{B}} \cdot \overline{(A \odot B)\bar{C}}$$

$$= (\bar{A}+B)\left[\overline{(A \odot B)}+C\right]$$

$$= (\bar{A}+B)\left[\bar{A}B+A\bar{B}+C\right]$$

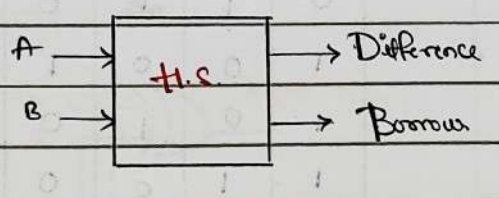
$$= \bar{A}+B+\bar{A}C+\bar{A}B+BC$$

$$\Rightarrow \bar{A}B+\bar{A}C+BC$$

↳ Borrow.

Half Subtractor : Two bit subtractor are known as half subtractor.

Step 1:



0 1 1 0

-0 -0 -1 -1

00 01 00 11

Borrow, difference, borrow, difference

Step 2:

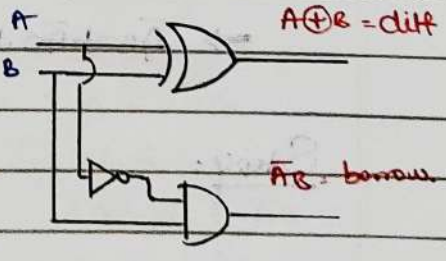
A	B	Diff.	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0.

Step 3: Diff = $A \oplus B$

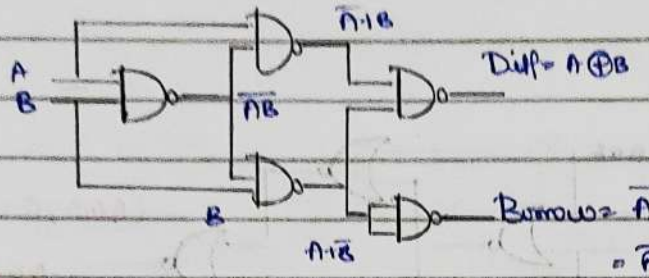
Borrow = $\bar{A}B$.

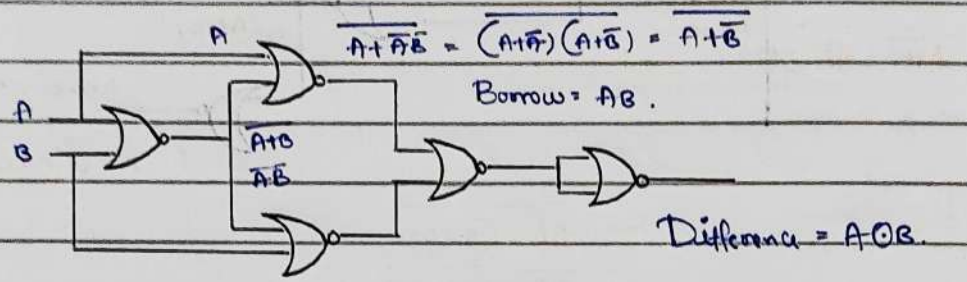
Step 4:

Step 5:



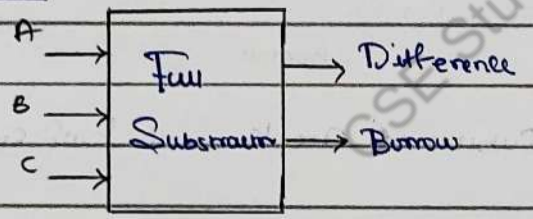
Half Subtractor using NAND Gates.





Full Subtractor.

Step 1:



Step 2:

Input			Output	
A	B	C	Diff	Bor.
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1.

Step 3: $Diff = \sum m(1,2,4,7)$

$= A \oplus B \oplus C$

$Borrow = \sum m(1,2,3,7)$

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

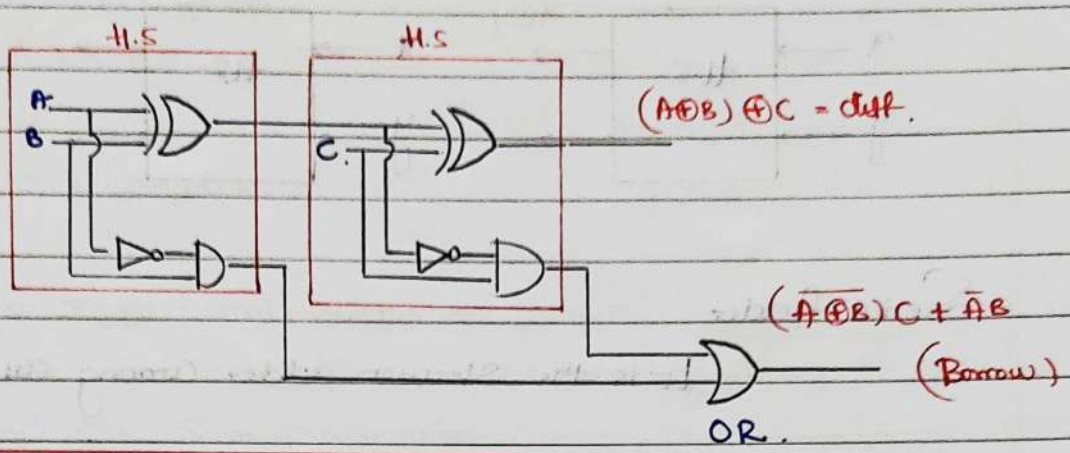
$= (\bar{A}\bar{B} + AB)C + \bar{A}B(\bar{C}+C)$

$= (\overline{A \oplus B})C + \bar{A}B$ → Semi minimized expression.

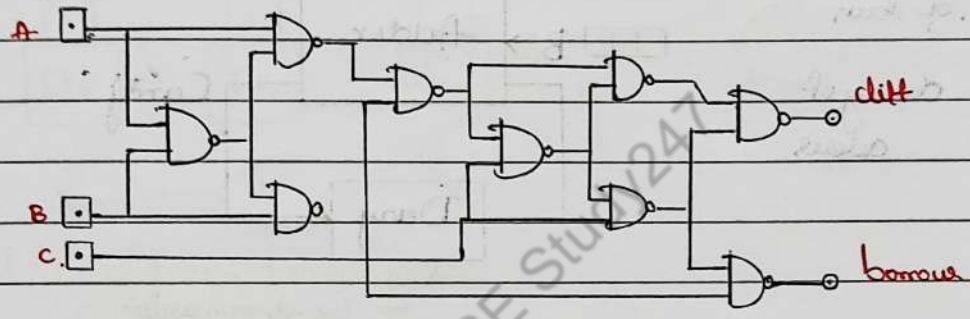
$\Rightarrow \bar{A}B + \bar{A}C + BC$

Step 4:

Steps:



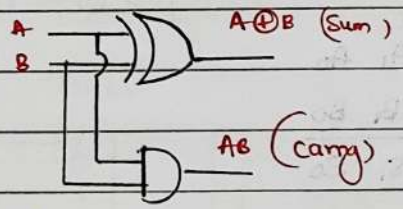
1 Full Substractor = 2 HS + Or Gate.



Half adder

Sum = A⊕B

Carry = AB

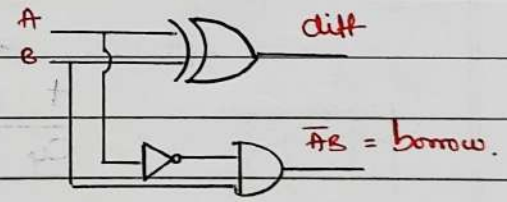


NAND } 5.
NOR }

Half Substractor

Diff = A⊕B

Borrow = $\bar{A}B$



NAND } 5
NOR }

Full Adder.

Sum = A⊕B⊕C

Carry = $\Sigma m(3,5,6,7)$

= (A⊕B)C + AB

= AB + AC + BC

NAND/NOR = 9.

Full Substractor

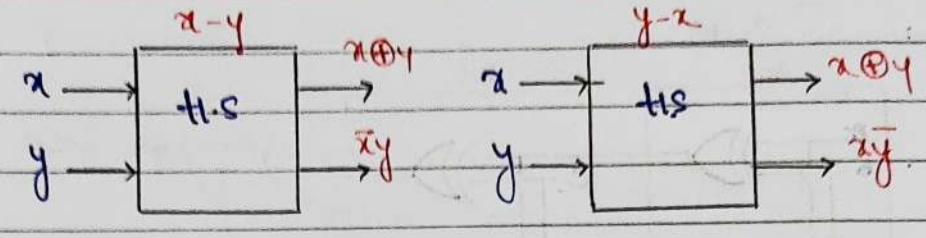
difference = A⊕B⊕C

borrow = $\Sigma m(1,2,3,7)$

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

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

NAND/NOR = 9.



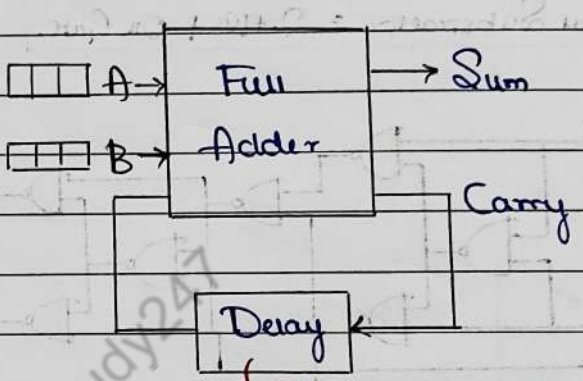
Serial Adder "One by one addition"

↳ It is the slowest adder among all the adder.

$T = n \cdot T_{delay}$

$n \rightarrow$ no. of bits

$T_{delay} \rightarrow$ delay of adder.

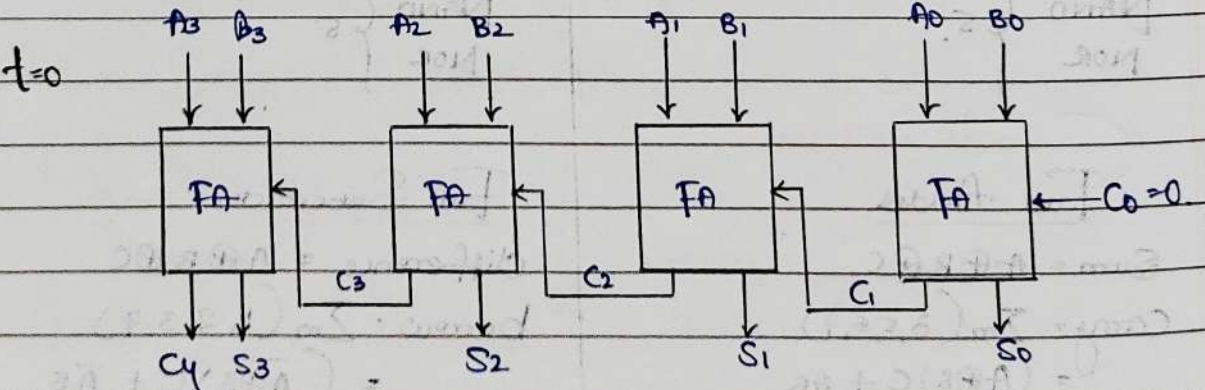


Parallel Adder (Ripple Carry Adder)

↳ Inputs are connected simultaneously

4 Bit

	C_3	C_2	C_1		
	A_3	A_2	A_1	A_0	
+	B_3	B_2	B_1	B_0	
C_4	S_3	S_2	S_1	S_0	

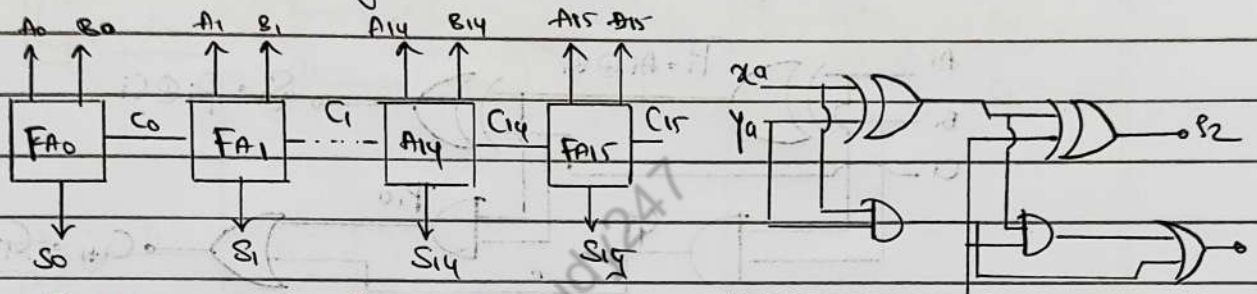


n bit Parallel adder

1. (n-1) FA + 1 HA
2. n-FA
3. (2n-1) HA + (n-1) OR.

$$T = (n-1)T_{carry} + \max(T_{sum} + T_{carry})$$

Q1. A 16-bit ripple carry adder is realised using 16 identical full adders (FA) as shown. The carry propogation delay of each FA is 12ns and the sum-propogation delay of each FA is 15ns. The worst case delay (in ns) of the 16-bit adder will be _____



$T = (n-1)T_{carry} + \max\{T_{sum}, T_{carry}\}$

$= (16-1) \times 12ns + \max\{15ns, 12ns\}$

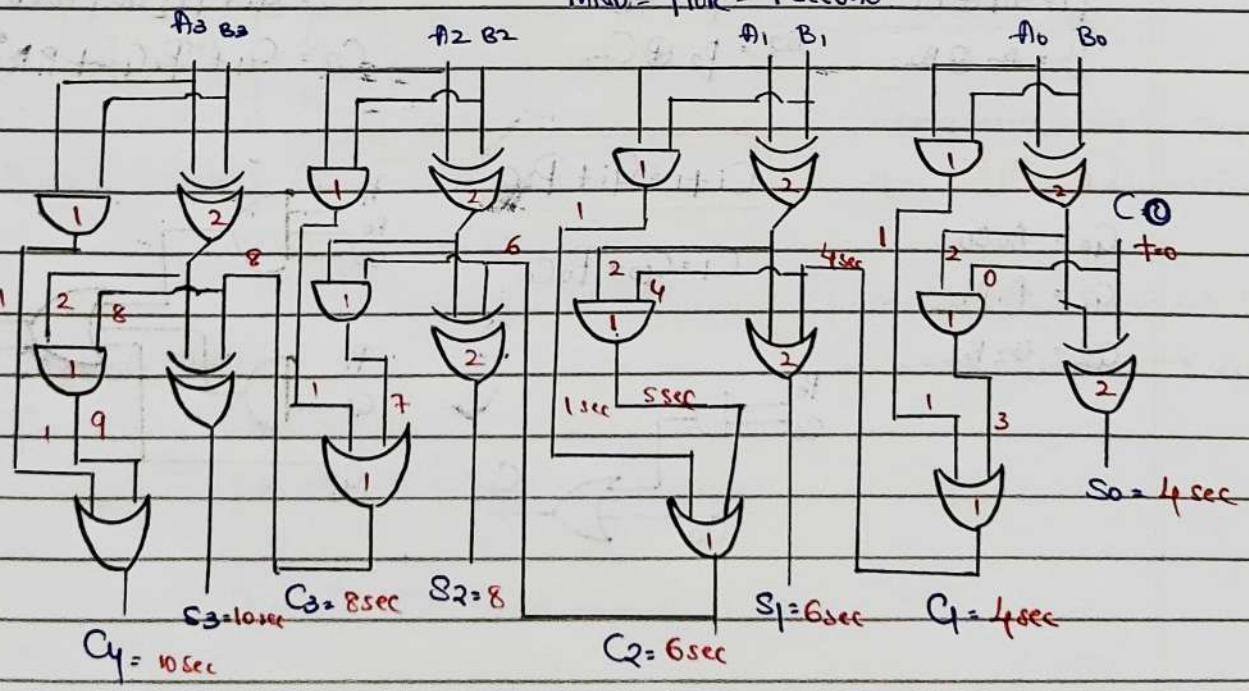
$= 15 \times 12 + 15$

$= 195\ ns$

Parallel Adder [4 bit]

$T_{XOR} = 2$ second

$T_{AND} = T_{OR} = 1$ second



$$T = (n-1)\{T_{AND} + T_{OR}\} + \max\{T_{sum}, T_{carry}\}.$$

$T = (4-1)\{1+1\} + \max\{4,4\}$

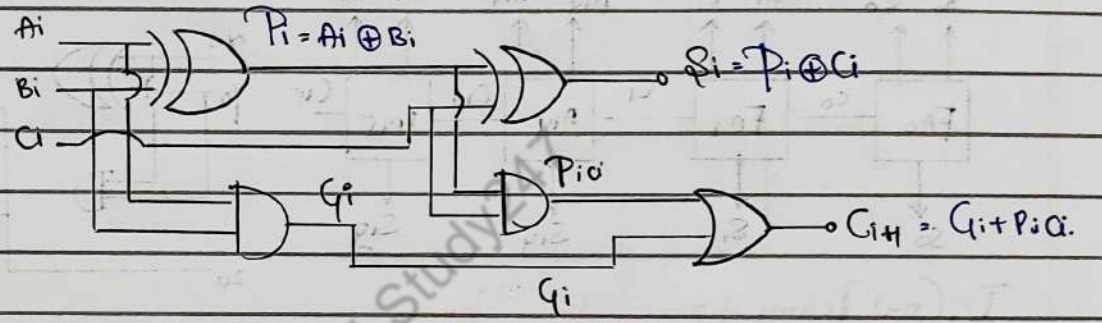
$= 3 \times 2 + 4$

$= 10\text{ns}.$

Parallel Adder, LACA

[Look Ahead Carry Adder]

↳ faster adder among all the adder



$P_i \rightarrow$ Carry Propagating term

$G_i \rightarrow$ Carry generating term

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

$P_0 = A_0 \oplus B_0$ $\quad S_0 = P_0 \oplus C_0$

$P_1 = A_1 \oplus B_1$ $\quad S_1 = P_1 \oplus C_1$

$P_2 = A_2 \oplus B_2.$ $\quad S_2 = P_2 \oplus C_2.$

$C_2 = G_1 + P_1 C_1$

$C_2 = G_1 + P_1[G_0 + P_0 C_0]$

$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0$

$G_i = A_i B_i$

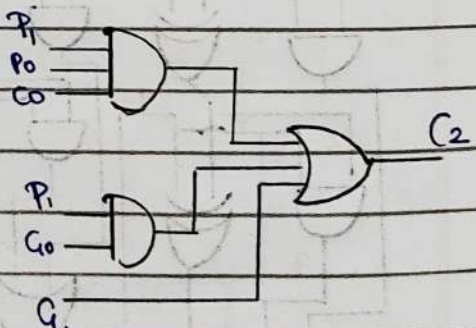
$G_0 = A_0 B_0$

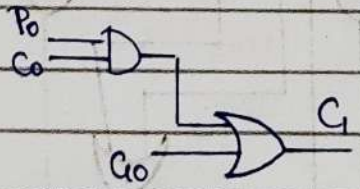
$G_1 = A_1 B_1$

$G_2 = A_2 B_2$

$$C_{i+1} = G_i + P_i C_i$$

$C_1 = G_0 + P_0 C_0.$

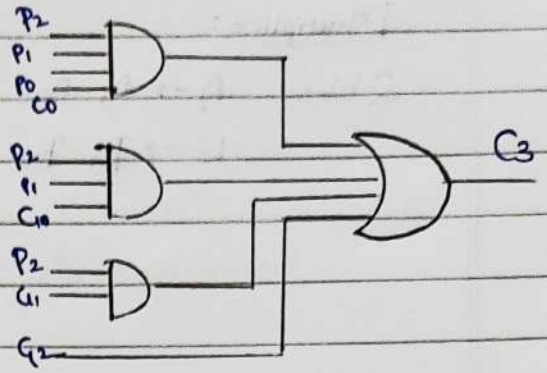




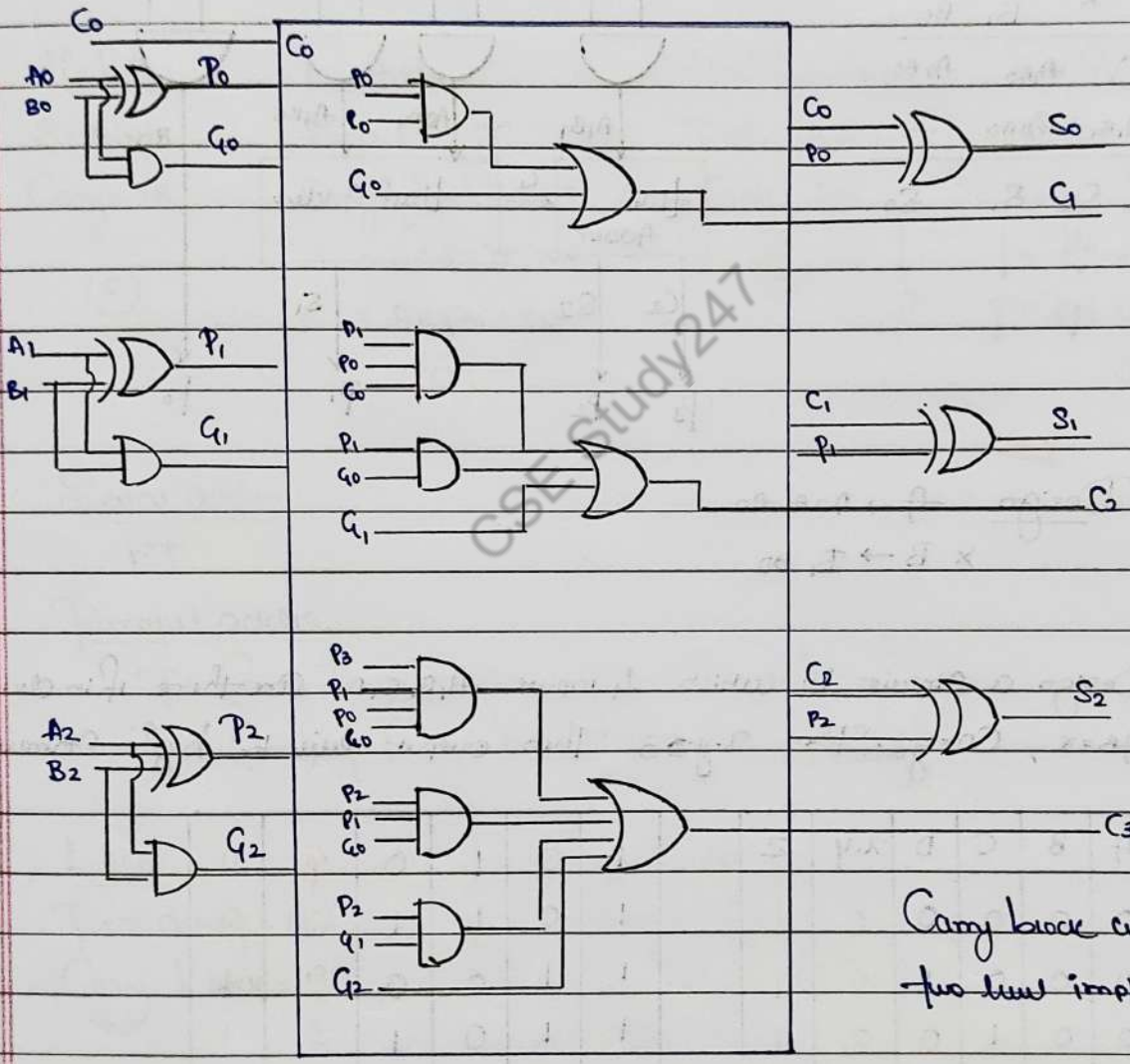
$C_3 = G_2 + P_2 C_2$

$C_3 = G_2 + P_2 [G_1 + P_1 G_0 + P_1 P_0 G_0]$

$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$.



CARRY BLOCK



Carry block can be always two level implementation.

* Delay for Carry Block ⇒ 2T
* For entire circuit ⇒ 4T

'n' bit RACA

For carry block:

No. of AND Gate = $\frac{n(n+1)}{2}$

No. of OR Gate = n.

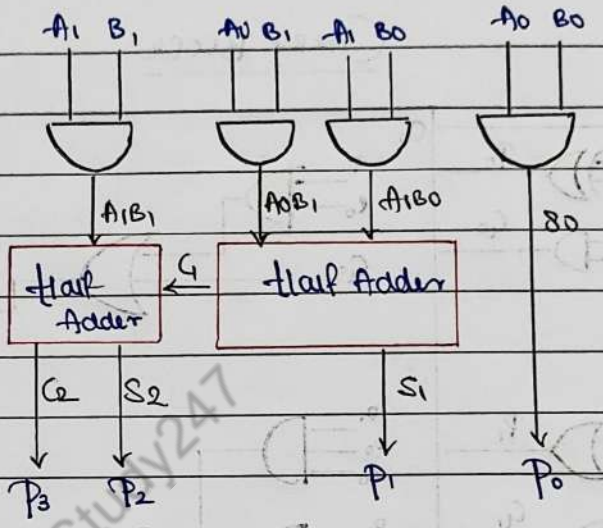
Multiplier:

2 bit. $A \rightarrow A_1\ A_0$

$B \rightarrow B_1\ B_0$

$$\begin{array}{r} 52 \\ \times\ 52 \\ \hline 2704 \\ \hline P_3\ P_2\ P_1\ P_0 \end{array}$$

$$\begin{array}{r} A_1\ A_0 \\ \times\ B_1\ B_0 \\ \hline C_1\ \ A_1B_0\ \ A_0B_0 \\ A_1B_1\ \ A_0B_0\ \ - \\ \hline C_2\ S_2\ S_1\ S_0 \\ \downarrow\ \downarrow\ \downarrow\ \downarrow \\ P_3\ P_2\ P_1\ P_0 \end{array}$$



HW

Design. $A \rightarrow A_2A_1A_0$

$\times\ B \rightarrow B_1B_0$

Q1. Design a circuit in which 4 inputs A, B, C, D. Are there if in decimal AB = x, CD = y. Then $x.y \leq 3$ Then output will be high Otherwise low.

A	B	C	D	x.y	Z
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	2	1
0	1	1	1	3	1
1	0	0	0	0	1
1	0	0	1	2	1
1	0	1	0	4	0
1	0	1	1	6	0
1	1	0	0	0	1
1	1	0	1	3	1
1	1	1	0	6	0
1	1	1	1	9	0

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1		
10	1	1		

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

$= \overline{AC}$

Combinational Circuit - { Static Circuit }

Quick Summary!

Comparator

$(A > B)\ X \Rightarrow A_1\bar{B}_1 + (A_1 + \bar{B}_1)A_0\bar{B}_0 \rightarrow$ minimised

$A_1\bar{B}_1 + (A_1 \odot B_1)A_0\bar{B}_0 \rightarrow$ Semi-minimised

MUX, DEMUX

Encoder, Decoder.

Half Adder

$S = A \oplus B$

$Carry = AB$

5.

Full Adder

$Sum = A \oplus B \oplus C$

$Carry = \sum m(3,5,6,7)$

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

$= AB + AC + BC$

9.

Half Subtractor

$Diff = A \oplus B$

$Carry = \bar{A}B$

5

Full Subtractor

$Diff = A \oplus B \oplus C$

$Carry = \sum m(1,2,3,7)$

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

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

9

Serial adder

D.T.

Parallel adder

$T = (n-1)T_{carry} + Max\{T_{sum} + T_{carry}\}$

$T = (n-1)\{T_{AND} + T_{OR}\} + Max\{T_{sum}, T_{carry}\}$.

LACA

Total delay = 47

Carry block = 27

Carry Block

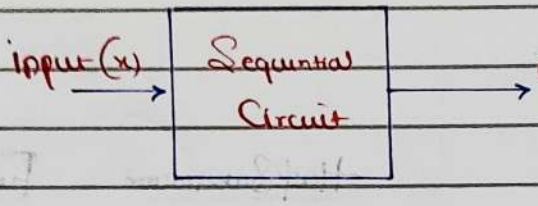
$AND = \frac{n(n+1)}{2}$

$OR = n$

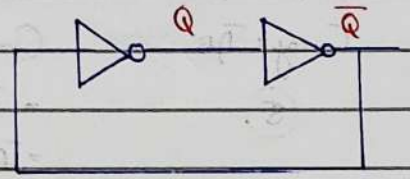
SEQUENTIAL CIRCUIT

Sequential Circuit:

* A circuit with feedback and memory are called Sequential Circuit.
* Output of the sequential circuit depends on previous output as well as present state of input.



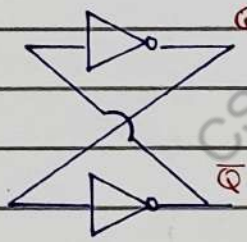
(i)



Basic Memory Element

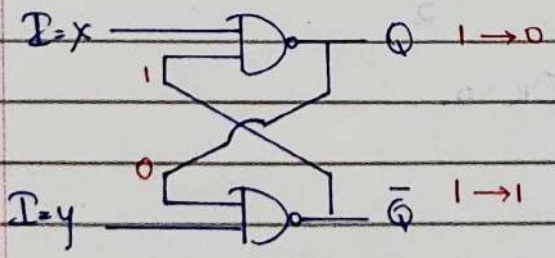
i = ii

(ii)



Latches : Basic memory element

* Latches are level triggered
* Latches have two output which is compliment of each other



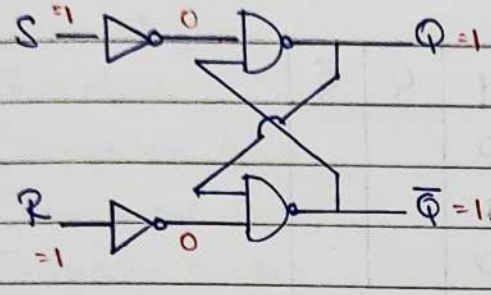
$\overline{X}$	Y	Q	$\overline{Q}$	
0	0	1	1	× Invalid
0	1	1	0	
1	0	0	1	
1	1	Q	$\overline{Q}$	→ Hold

NAND

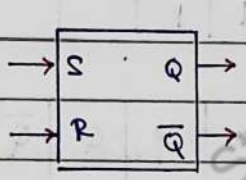
A	B	$Y=\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR: Whenever $x=y=1$ is applied and invalid condition occurs than a NAND having lower Propogation delay first change its output and other remain on it previous state are called racing problem or raising problem.

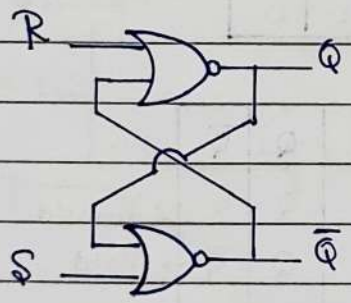
SR : Set-Reset Latch



S	R	Q	$\bar{Q}$	
0	0	Q	$\bar{Q}$	Hold
0	1	0	1	Reset
1	0	1	0	Set.
1	1	×	×	(Invalid)



S-R Latch by using NOR Gate:

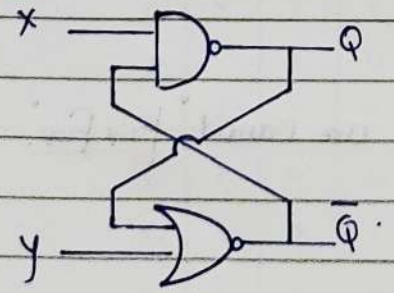


S	R	Q	$\bar{Q}$	
0	0	Q	$\bar{Q}$	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	×	×	Invalid.

NOR

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

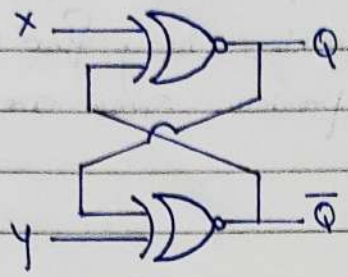
HW

① 

x	y	Q	$\bar{Q}$
0	0		
0	1		
1	0		
1	1		

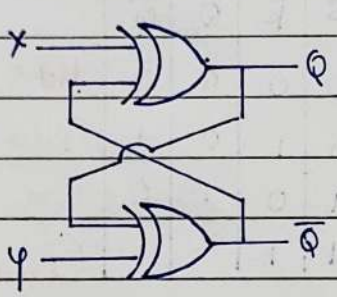
HW

(2)



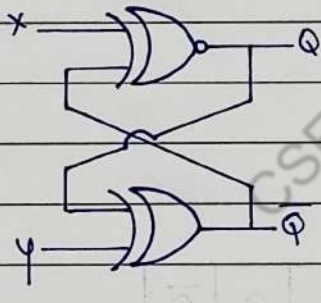
x	y	Q	$\bar{Q}$
0	0		
0	1		
1	0		
1	1		

(3)



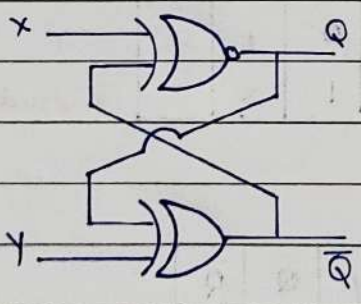
x	y	Q	$\bar{Q}$
0	0		
0	1		
1	0		
1	1		

(4)



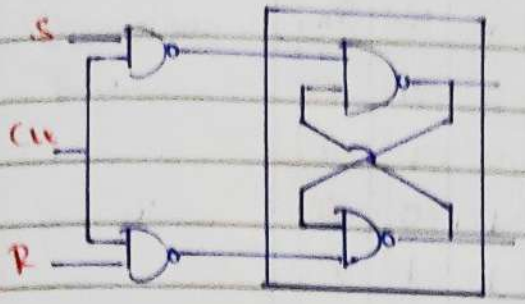
x	y	Q	$\bar{Q}$
0	0	1	0
0	1	1	0
1	0	Q	$\bar{Q}$
1	1	1	0

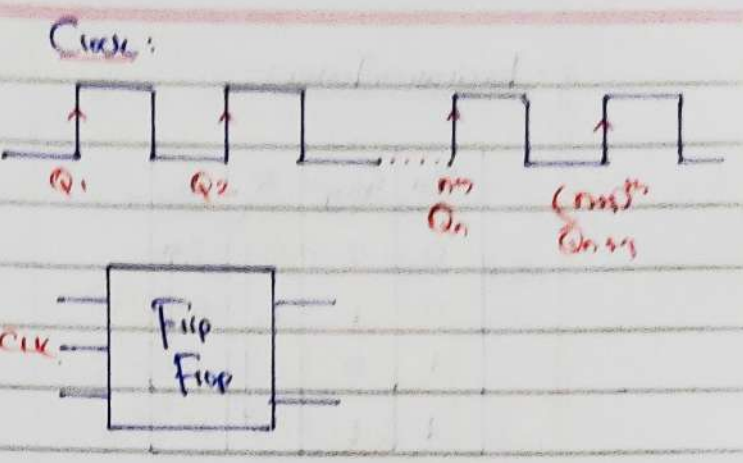
(5)



x	y	Q	$\bar{Q}$	
0	0	Q	$\bar{Q}$	Q → Invalid
0	1	Q	$\bar{Q}$	$\bar{Q}$ → Hold.
1	0	$\bar{Q}$	Q	Q → Toggle
1	1	$\bar{Q}$	$\bar{Q}$	$\bar{Q}$ → Invalid

→ Latches with control phenomena are called "Flip Flop".



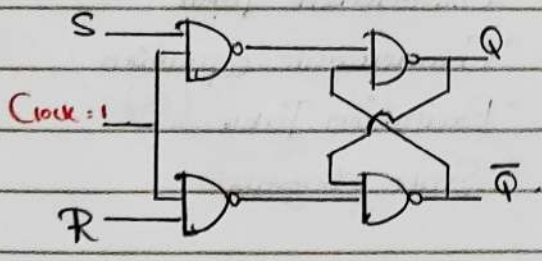


Flip Flop Steps

1. Circuit diagram
2. Truth table
3. Characteristic table
4. Characteristic Equation
5. Excitation table
6. State diagram

SR Flip Flop (Set-Reset Flip Flop)

Circuit diagram:



Truth Table:

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	
0	0	Q_n	$\bar{Q}_n$	→ Hold/Prev.
0	1	0	1	→ Reset
1	0	1	0	→ Set
1	1	x	x	→ Invalid

Characteristics 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

$Q_{n+1} = (S, R, Q_n) = \sum m(1,4,5) + \sum d(6,7)$

Characteristic Equation:

S \ Q_n	$\bar{R}\bar{Q}_n$ 00	$\bar{R}Q_n$ 01	RQ_n 11	$R\bar{Q}_n$ 10
$\bar{S}$ 0		1		
S 1	1	1	x	x

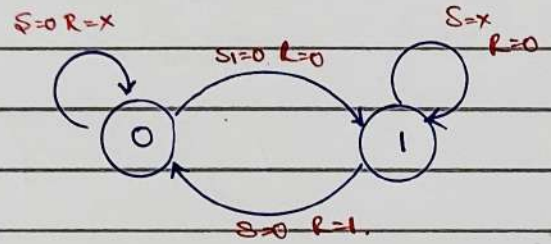
$Q_{n+1} = S + \bar{R}Q_n$

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

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	x
1	1	1	x

State Digram:



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

HW

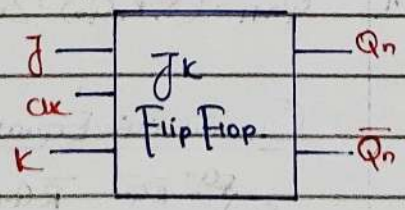
Q. Truth Table :

C	J	Q_{n+1}
0	0	$\overline{Q_n}$
0	1	Q_n
1	0	1
1	1	0

Write :
* Characteristic Table
* Characteristic equation
* Excitation Table
* State diagram.

JK Flip Flop

Symbol:



Truth Table:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Characteristic 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

$Q_{n+1}(J, K, Q_n) = \Sigma m(1,4,5,6)$

Characteristic Equation:

J \ KQn	$\bar{K}\bar{Q}_n$ 00	$\bar{K}Q_n$ 01	KQ_n 11	$K\bar{Q}_n$ 10
$\bar{J}$ 0		1		
J 1	1	1		1

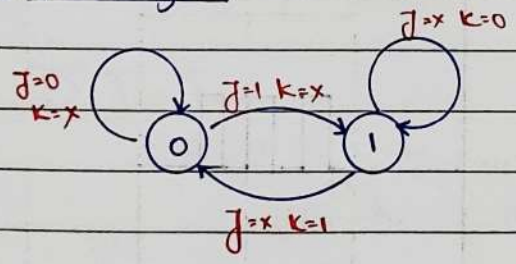
$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

Excitation Table:

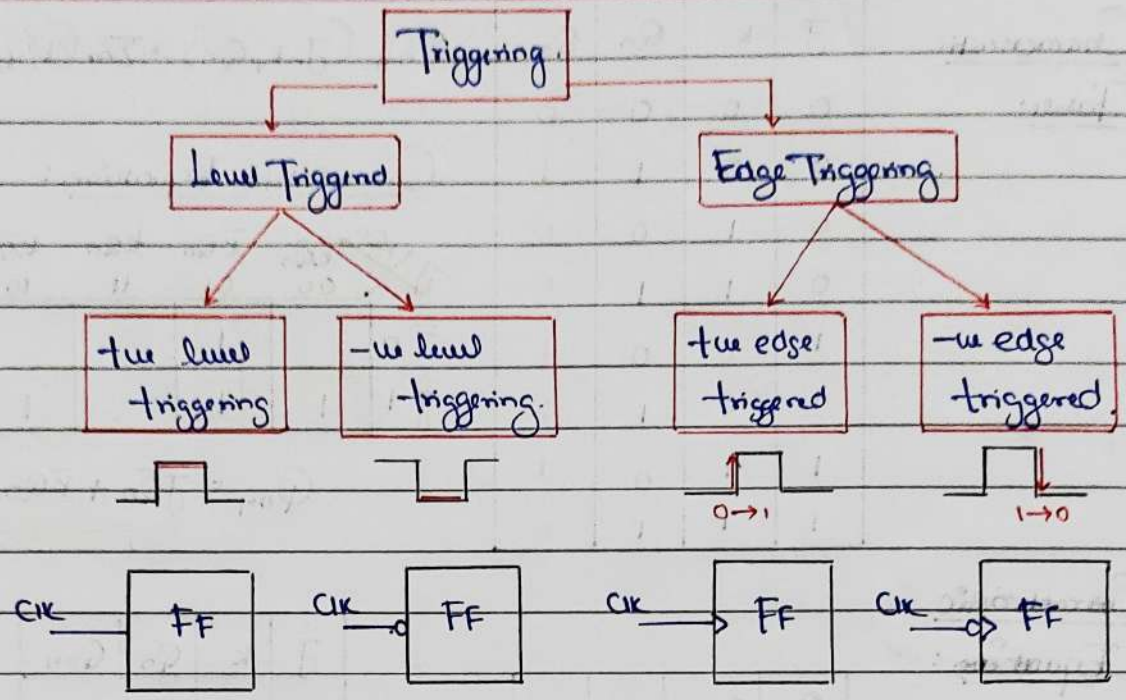
Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

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

State Diagram:

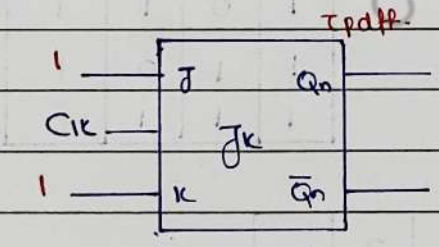


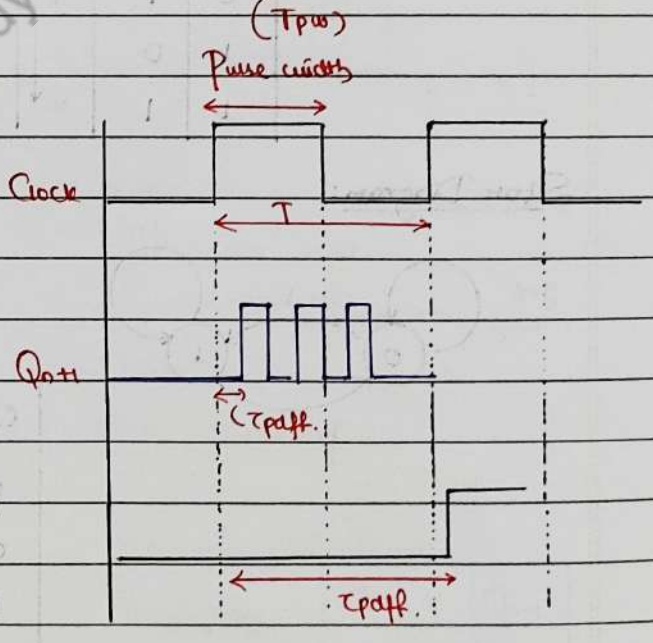
Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0



Note: level triggered JK flip flop suffers from the problem of "RACE AROUND".

Race around Problem:



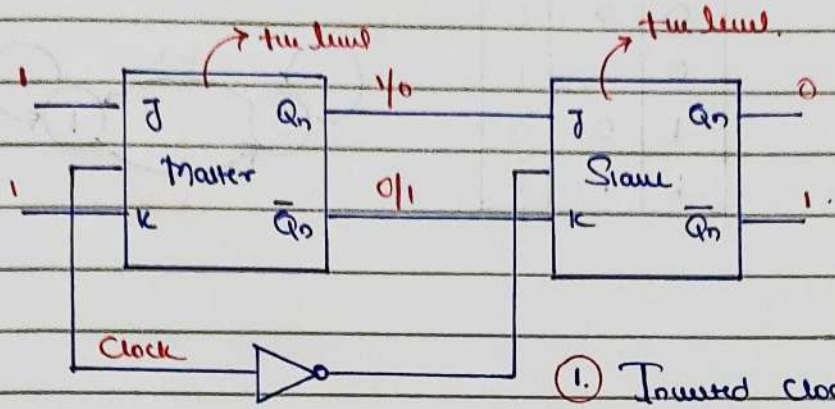


To avoid the Race around Problem.

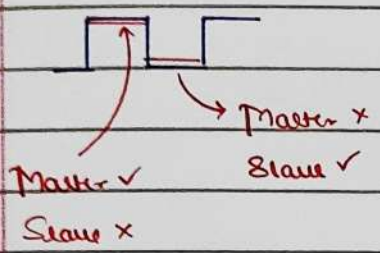
1. $T_{pw} < T_{pdff} < T_{clk}$
2. By Master Slave ff.

→ When J=K=1 is applied to the level sensitive of JK FF, then within the duration of pulse output of the FF toggle more than one times are called "Race around Problem".

Master Slave FF.



(1.) Inverted clock is applied to the slave as compared to master.



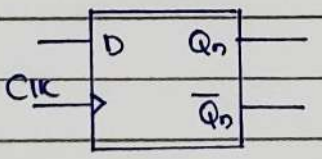
(2.) Master-Slave flip flop is used to store single bit because output is taken only from slave.

(3.) By the operation it seems that master is level triggered where as slave is negative edge triggered.

D Flip Flop.

* It is known as Delay FF or Transparent FF
* In the D FF whatever the input is applied it will directly come to the output along with the clock.

Symbol:



Characteristic Table:

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

Truth Table:

D	Q_{n+1}
0	0
1	1.

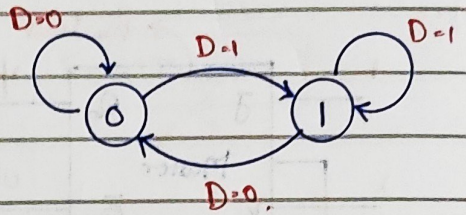
Characteristic Equation:

$Q_{n+1} = D$

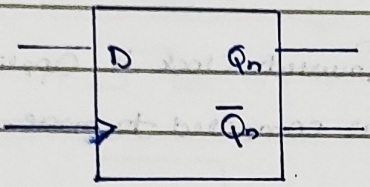
Excitation Table:

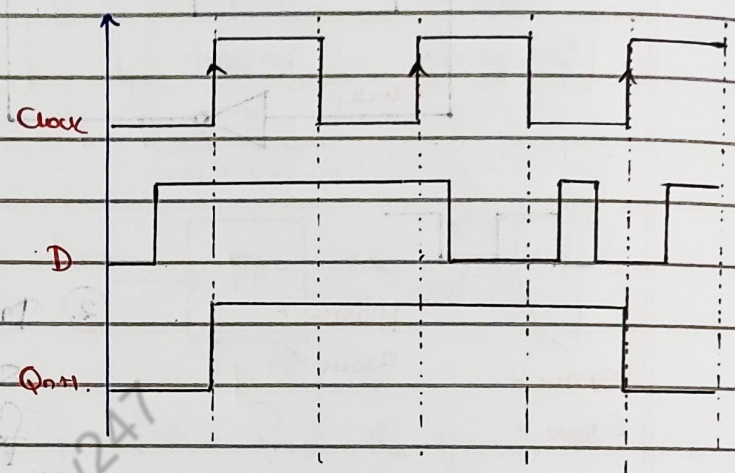
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

State Diagram:

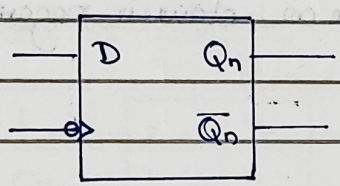


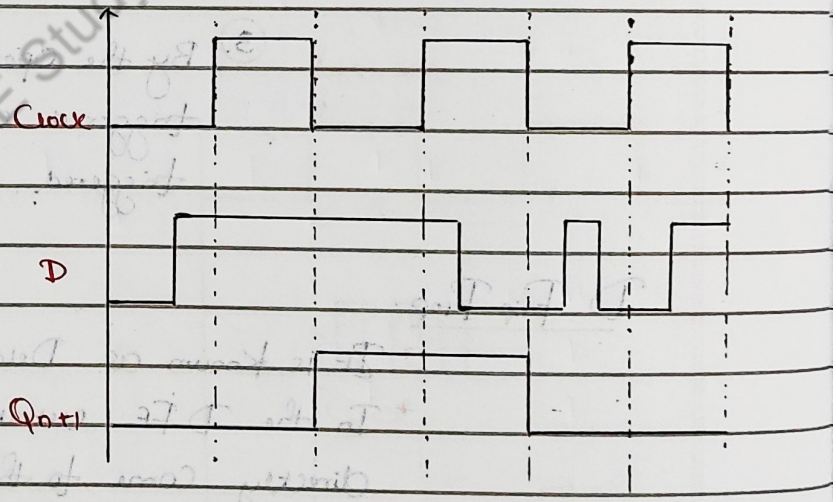
Q_1





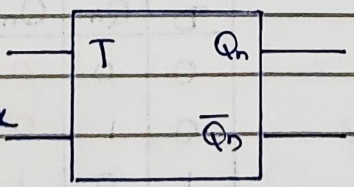
Q_2



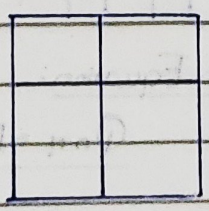


T Flip Flop (Toggle Flip Flop)

Symbol:



Truth Table:



Characteristic Table:

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

Characteristic Equations:

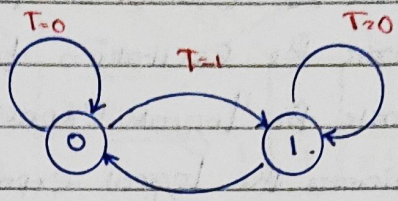
$Q_{n+1} = \overline{T}Q_n + T\overline{Q_n}$

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

Excitation Table:

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

State Diagram:



Note: J-K FF is also known as universal flip flop.

	J	K	Q_{n+1}	
SR flf	0	0	Q_n	T.FF.
	0	1	0	
	1	0	1	
	1	1	$\overline{Q_n}$	

SR FF

→ $Q_{n+1} = S + \overline{R}Q_n$

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

JK FF

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

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

D FF

→ $Q_{n+1} = D$

D	Q_{n+1}
0	0
1	1

T FF

→ $Q_{n+1} = T \oplus Q_n$

T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

Excitation Table

Q_n	Q_{n+1}	S	R	J	K	D	T
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

Designing of Flip Flop:

Step 1 : Write the characteristic table of desired FF
Step 2 : Write the excitation table of available FF
Step 3 : Write the logical expression
Step 4 : Minimize the logical expression
Step 5 : Hardware implementation.

Q3. Design JK flip flop (desired) using SR Flip Flop (available).

Step 1 / Step 2

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

Step 3: $S(J,K,Q_n) = \sum m(4,6) + \sum d(1,5)$

$R(J,K,Q_n) = \sum m(3,7) + \sum d(0,2)$

Step 4:

S ⇒

J \ KQn	00	01	11	10
0		x		
1	1	x		1

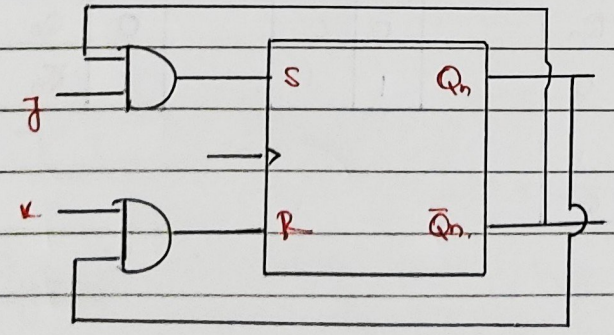
$S = J\bar{Q}_n$

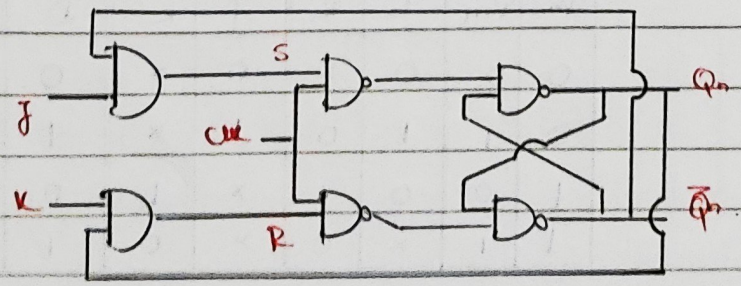
R ⇒

J \ KQn	00	01	11	10
0	x		1	x
1			1	

$R = KQ_n$

Step 5:



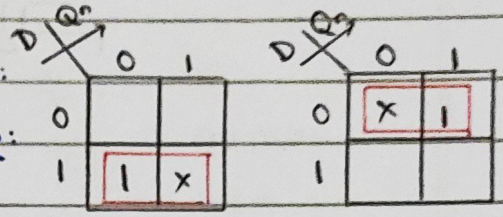


Q4. Design a D-FF by using S-R FF.

Step 1:
Step 2:

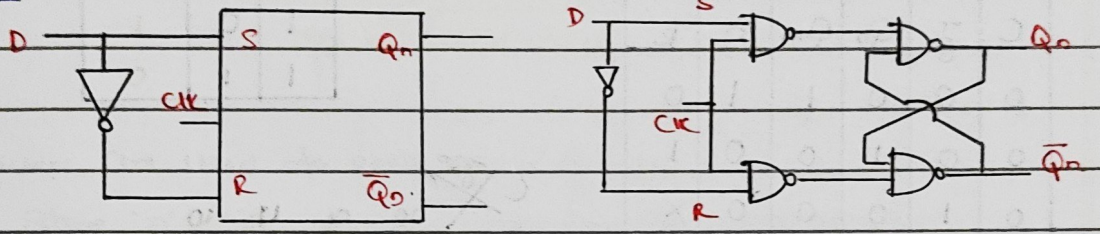
D	Q_n	Q_{n+1}	S	R
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

Step 3 & Step 4:



$S = D$ $\quad R = \bar{D}$.

Step 5:

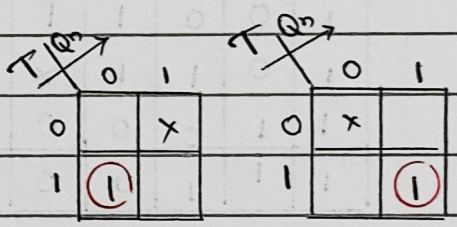


Q5 Design a T-FF using S-R Flip Flop:

T	Q_n	Q_{n+1}	S	R
0	0	0	0	x
0	1	1	x	0
1	0	1	1	0
1	1	0	0	1

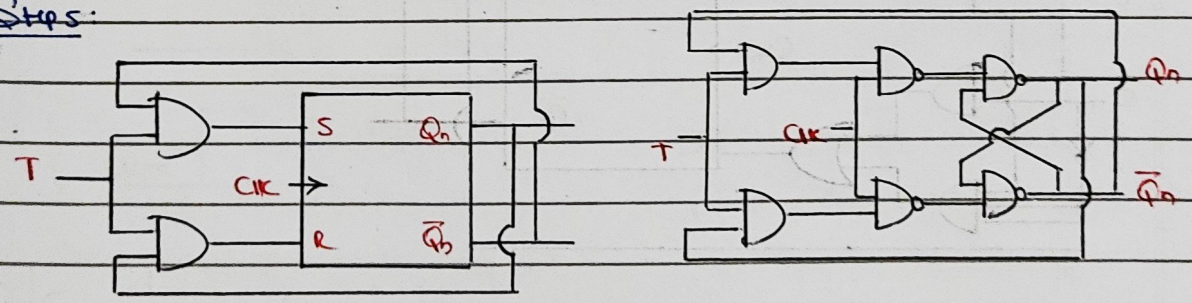
Step 1:
Step 2:

Step 3 & Step 4:



$S = T\bar{Q}_n$ $\quad R = TQ_n$.

Step 5:



Q6. Design T-FF using D-FF.

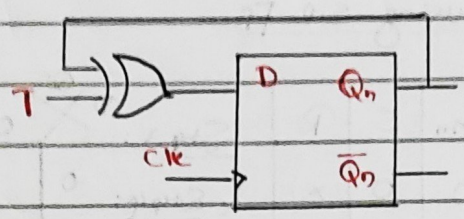
Step 1:
Step 2:

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

Step 3 & Step 4:

$D = T \oplus Q_n$

Steps:

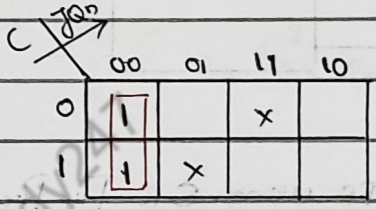


Q7. Design a CJ FF by using SR FF.

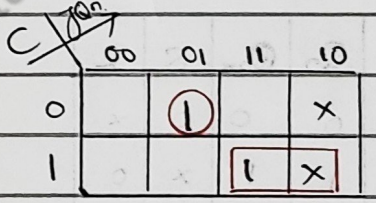
Truth Table:

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

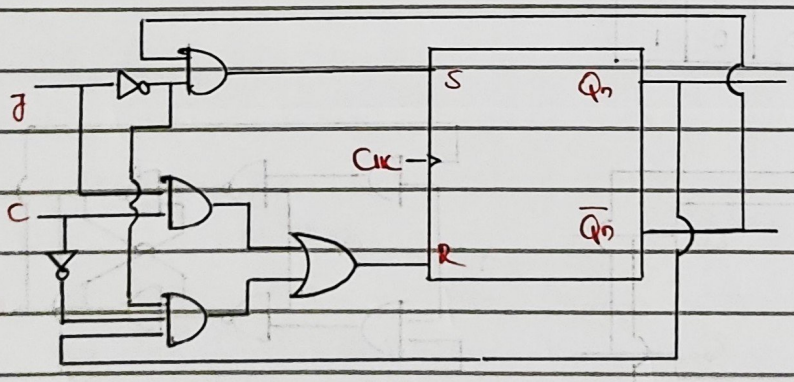
C	J	Q_n	Q_{n+1}	S	R
0	0	0	1	1	0
0	0	1	0	0	1
0	1	0	0	0	x
0	1	1	1	x	0
1	0	0	1	1	0
1	0	1	1	x	0
1	1	0	0	0	x
1	1	1	0	0	1

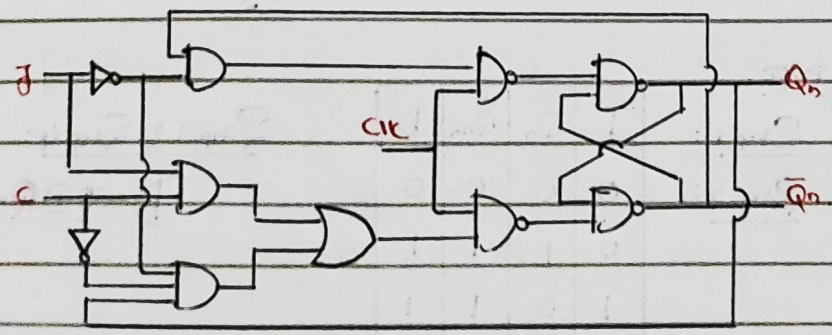


$S = \bar{J}\bar{Q}_n$



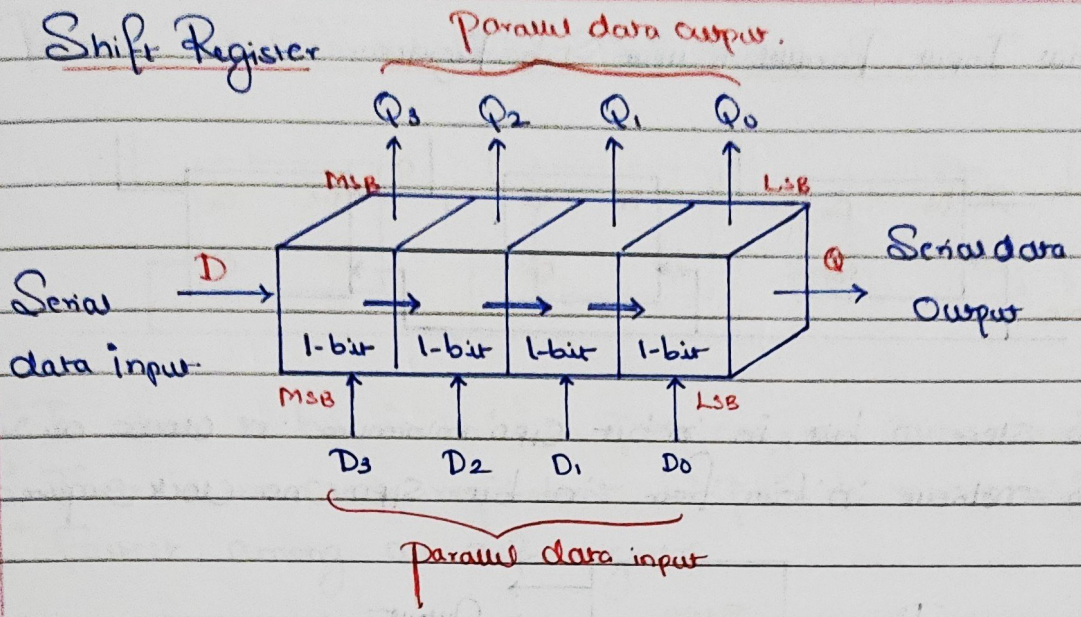
$R = \bar{C}\bar{J}Q_n + CJ$





Date 11/7/22

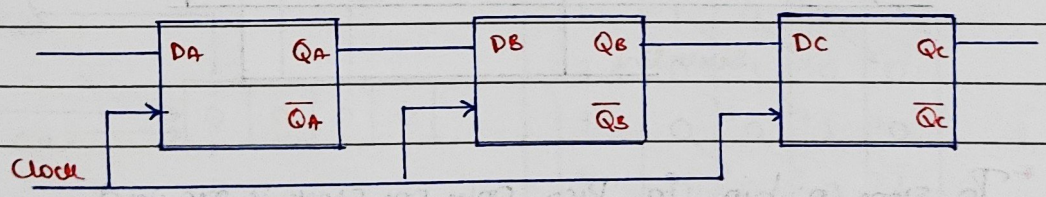
Shift Register



* Registers are used to store group of bits
* To store 'n' bits minimum 'n' flip flops are required.
* Generally D Flip Flops are used to Design registers.

1. Serial input Serial output Shift register (SISO)
2. Serial input Parallel output Shift register. (SIPO)
3. Parallel input Serial output Shift register. (PISO)
4. Parallel input Parallel Output Shift register. (PIPO)

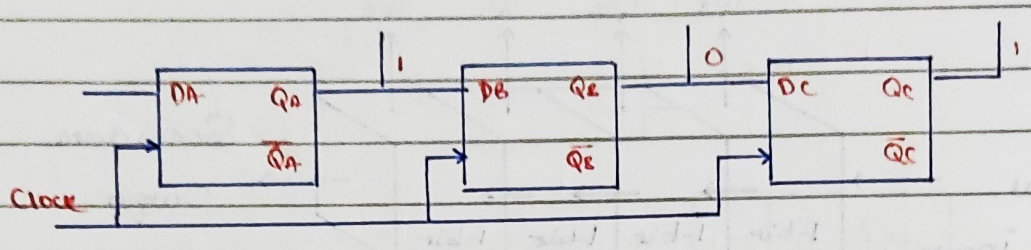
Serial Input Serial Output Shift Register.



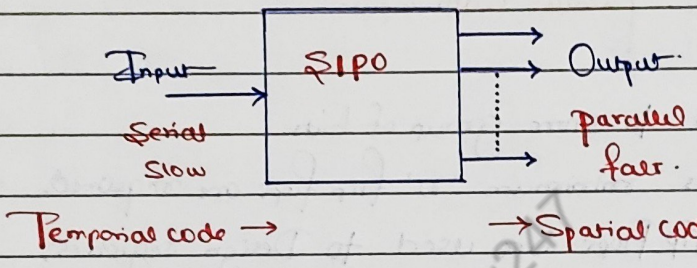
Clock	Input	Q_A	Q_B	Q_C
0	101	0	0	0
1		1	0	0
2		0	1	0
3		1	0	1

* To Store 'n' bit in SISO minimum 'n' Clocks are required
* To retrieve 'n' bits from 'n' bit SISO minimum 'n+1' Clocks are required.
* It is the slowest register among all the Shift register.

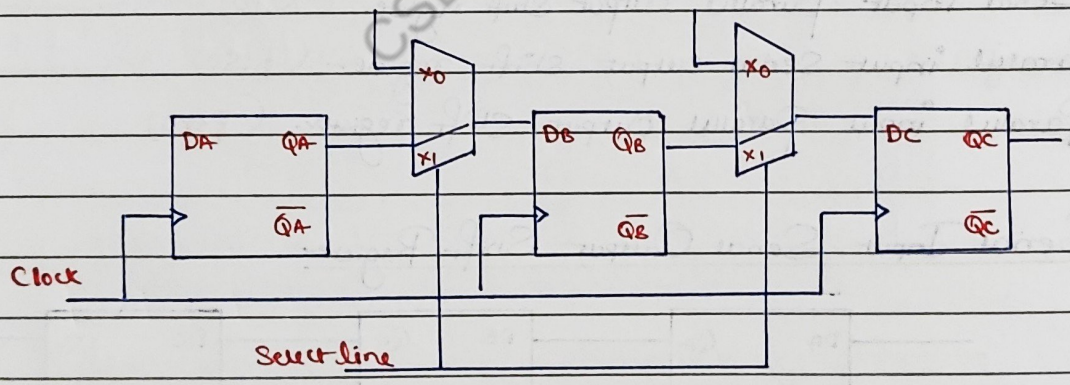
Serial Input Parallel Output Shift Register



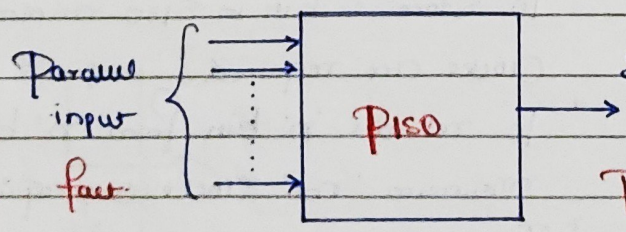
* To store 'n' bit in 'n' bit SIPO minimum 'n' clocks are required.
* To retrieve 'n' bits from 'n' bit SIPO no clock required.



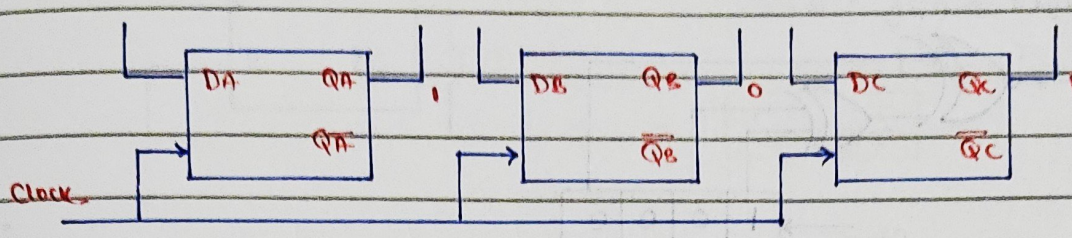
Parallel Input Serial Output Shift Register:



* To store 'n' bits in PISO only one clock is required.
* To retrieve 'n' bit from 'n' bit PISO minimum (n-1) clocks are required.



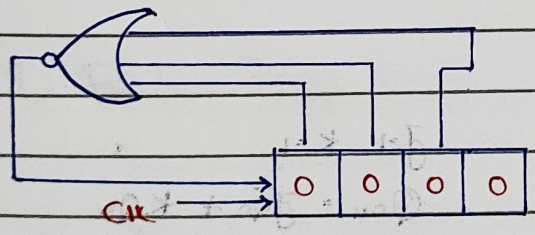
Parallel Input Parallel Output Shift Register.



* To store 'n' bit in 'n' bit PIPO only one clock is required.
* There is no clock requirement in PIPO to retrive 'n' bit.
* Fastest among all shift registers.

	Store	Retrive	Total	
SISO	n	n-1	2n-1	→ Slowest
SIPO	n	0	n	
PISO	1	n-1	n	
PIPO	1	0	1	→ Fastest.

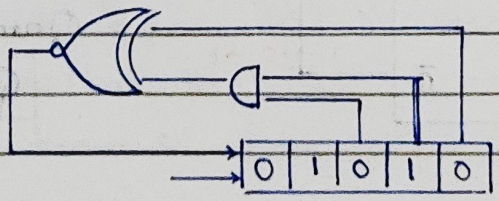
Q1 Write the state of the register given below? Assume all flip flops are reset initially



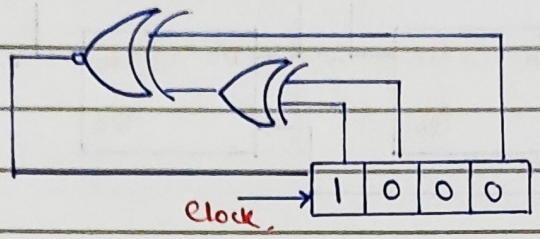
Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8.	0	0	0	1.

HW

Q2. Output after 137th clock cycle will be?



Q3. After 97th Clock the output will be?



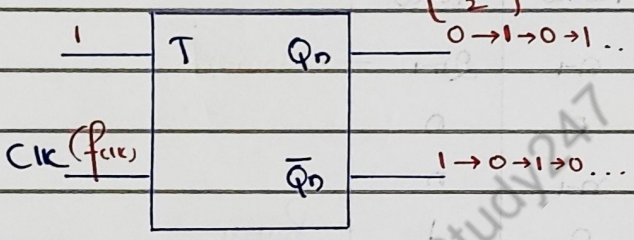
Toggle Mode of flip flops.

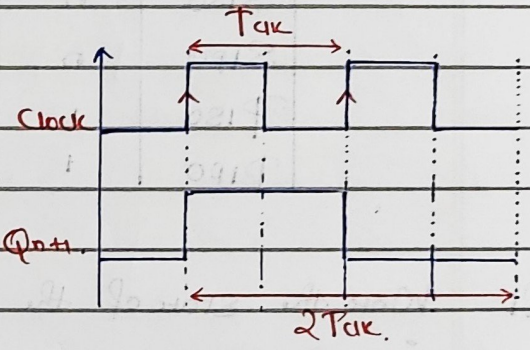
1. T Flip Flop toggle mode.

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

$Q_{n+1} = 1 \oplus Q_n = \bar{Q}_n \Rightarrow \boxed{Q_{n+1} = \bar{Q}_n}$

(Toggle Mode)



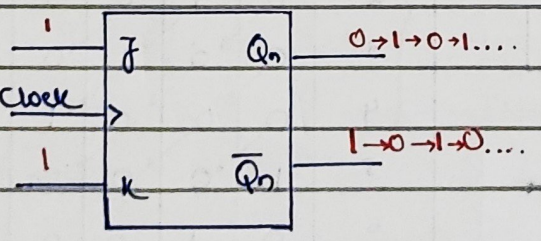


$f_{clk} = \frac{1}{T_{clk}}$

$f_{Q_{n+1}} = \frac{1}{2T_{clk}} = \frac{f_{clk}}{2}$

2. JK Flip Flop toggle mode

(i)



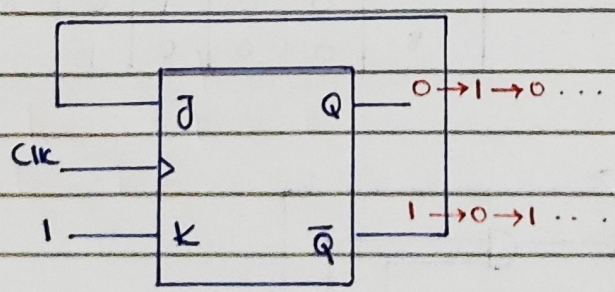
$J=1 \quad K=1$

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

$= 1\,\bar{Q}_n + 0\,Q_n$

$\boxed{Q_{n+1} = \bar{Q}_n}$

(ii)



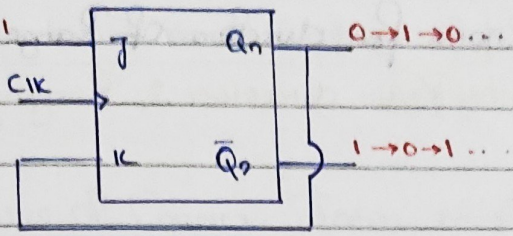
$J = \bar{Q}_n \quad K=1$

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

$Q_{n+1} = \bar{Q}_n \cdot \bar{Q}_n + 0 \cdot Q_n$

$\boxed{Q_{n+1} = \bar{Q}_n}$

(iii)



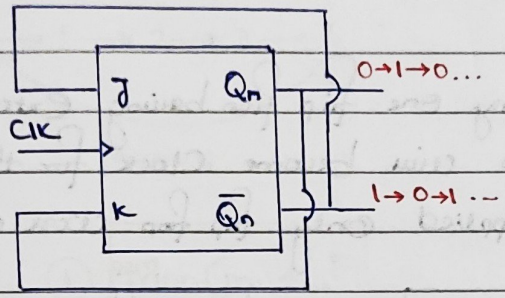
$J=1 \quad K=Q_n$

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

$= 1.\bar{Q}_n + \bar{Q}_n.Q_n$

$\boxed{Q_{n+1} = \bar{Q}_n}$

(iv)



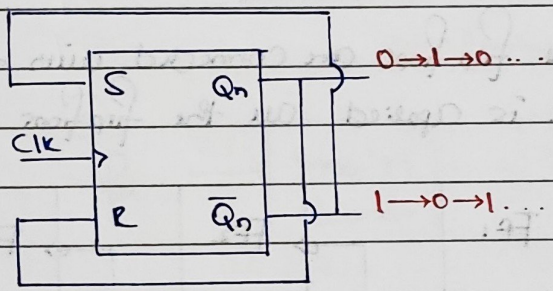
$J=\bar{Q}_n \quad K=Q_n$

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

$= \bar{Q}_n.\bar{Q}_n + \bar{Q}_n.Q_n$

$\boxed{Q_{n+1} = \bar{Q}_n}$

3 Toggle mode of SR Flip Flop.



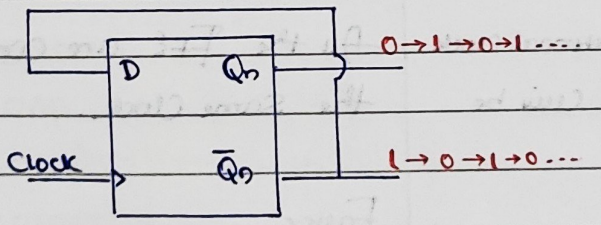
$S=\bar{Q}_n \quad R=Q_n$

$Q_{n+1} = S + \bar{R}Q_n$

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

$\boxed{Q_{n+1} = \bar{Q}_n}$

4. D-Flip Flop Toggle mode.



$D=\bar{Q}_n$

$Q_{n+1} = D$

$\boxed{Q_{n+1} = \bar{Q}_n}$

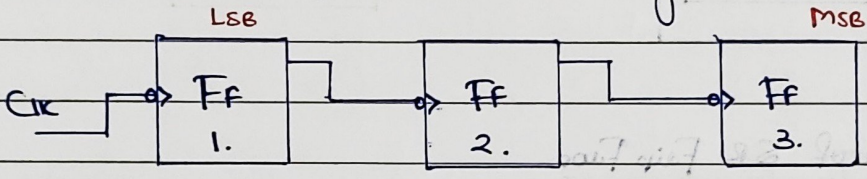
Counters:

- Counters are used to count number of clock
- Counter are used as frequency divider circuit.
- Counter are also used in ADC → (Analog to digital converter)
- Counters are also known as pulse stretcher circuit.

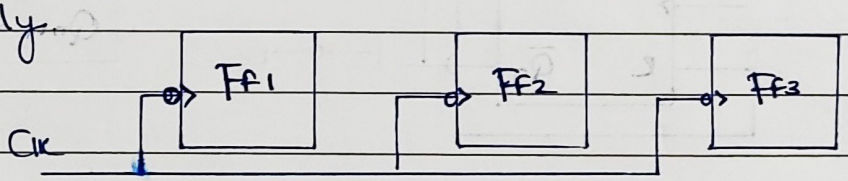
Counters also used in RADAR for direction of Range
↳ Radio direction & Ranging

Counters.
- Asynchronous Counter
- Synchronous Counters.

Asynchronous Counter : Only one flip flop having external clock and output of that flip flop will become clock for the next flip flop. So when clock applied only flip flop work at that time.



Synchronous Counter: All flip flops are connected with the same clock. Hence when the clock is applied all the flip flops work simultaneously.

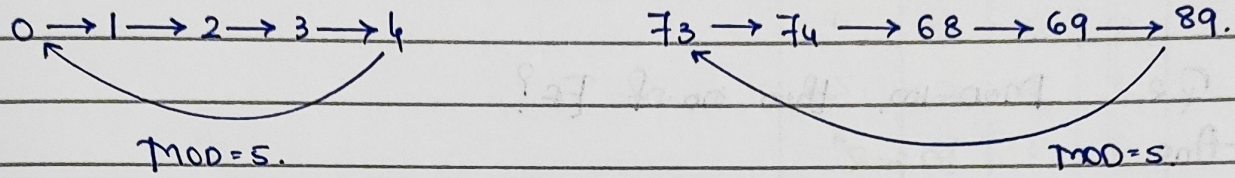


	Asynchronous Counter	Synchronous Counter.
1.	Only one FF having external clock and output of that FF will be clock for next FF.	All the FFs are connected with the same clock.
2.	Slower.	Faster
3.	Only increasing and decreasing Counting possible.	All types of countings are possible.
4.	There is transition error.	No transition error
5.	Eg: Ripple counter	Eg: Ring counter, Johnson Counter.

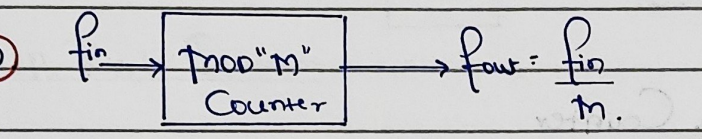
Maximum no. of States = 2^n.

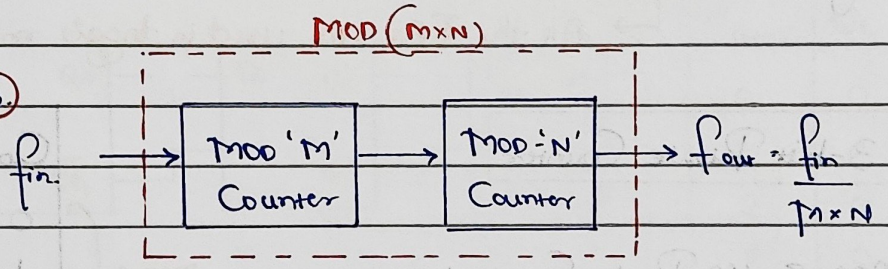
↳ n : no. of FFs.

Modulus of Counter: Total no. of States used by the counter One Cand mod of the counter.



Note: ① $MOD(m) \leq 2^n$.

② 

③ 

BCD Counter:

0 → 0000
1 → 0001
2 → 0010
3 → 0011
4 → 0100
5 → 0101
6 → 0110
7 → 0111
8 → 1000
9 → 1001

MOD → 10

$m \leq 2^n$

$n \geq \log_2 m$

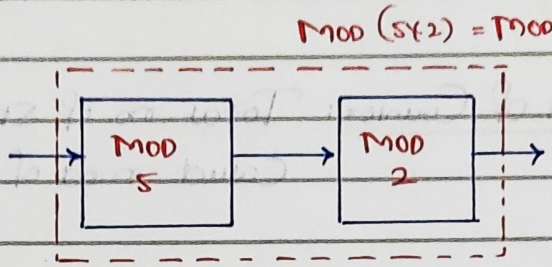
$n \geq \log_2 10$

$n \geq 3 \ldots$

$n = 4$

Q1. If mode-5 counter is cascaded with mod-2 counter, then it will become?

Ans. MOD-10 counter.



Q2. MOD-100, then no. of FF?

Ans $N \leq 2^n$

$n \geq \log_2 N$

$\geq \log_2 100$

$\geq 6.\ldots$

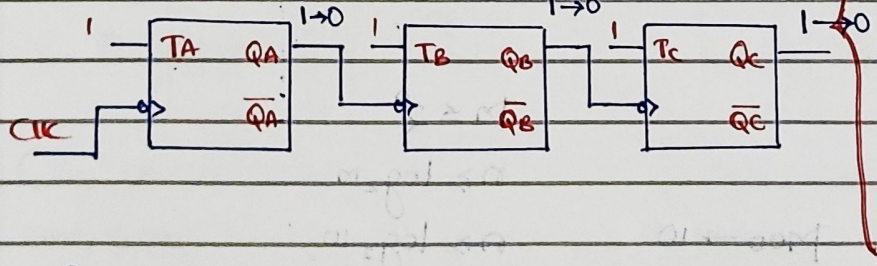
$n \geq 7$

Asynchronous Counter

→ All the FFs are used in toggle mode.

3-bit Ripple Counter

MOD-8 Up Ripple Counter:



Clock	Q_C	Q_B	Q_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
8	0	0	0
9	0	0	1
10	0	1	0

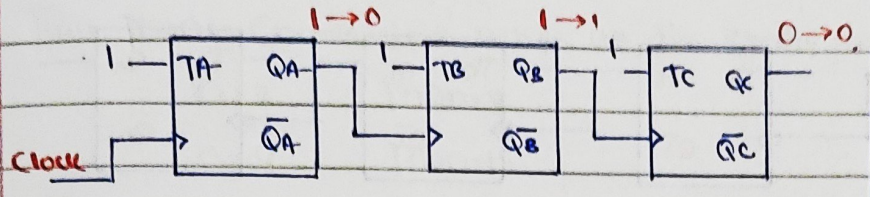
MOD-8

→ Q_A will toggle when -ve edge of the external clock appears.

→ Q_B will toggle when Q_A goes from 1 to 0

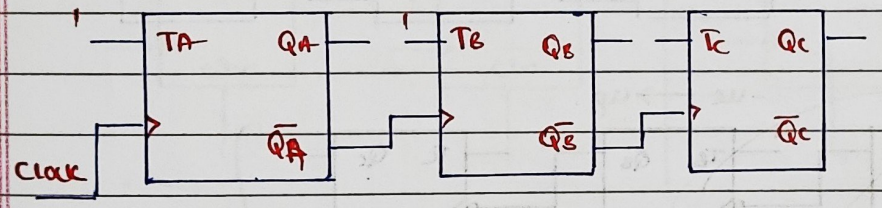
MOD-8 Down Ripple Counter:

111 → 110 → 101 → 100 → 011 → 010 → 001 → 000



Clock	Q_C	Q_B	Q_A
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0
9.	1	1	1.

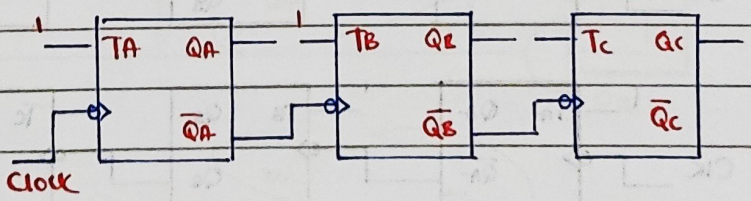
MOD - 8 UP Ripple Counter.



Clock	Q_C	Q_B	Q_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
8	0	0	0

MOD-8 DOWN Ripple Counter.

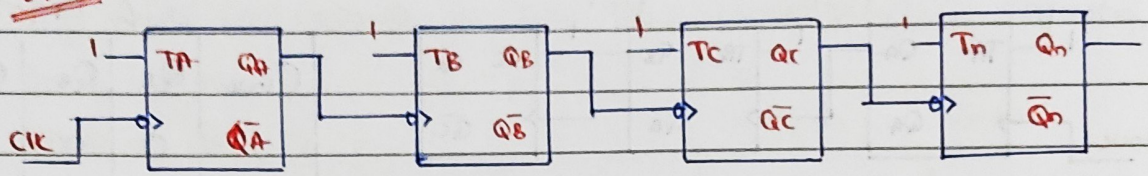
Clock	Q_C	Q_B	Q_A
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	6	0	1
8	0	0	0



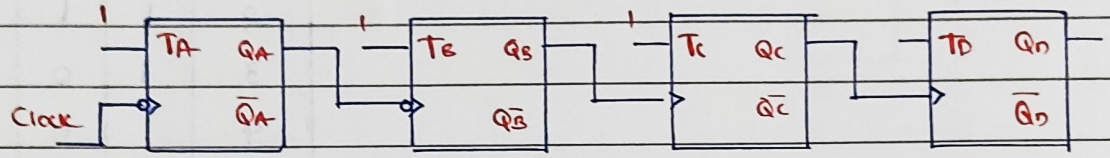
classmate
Date 14/7/23
Page

HW

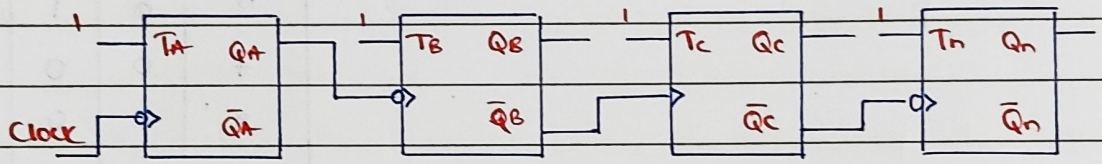
Q1.



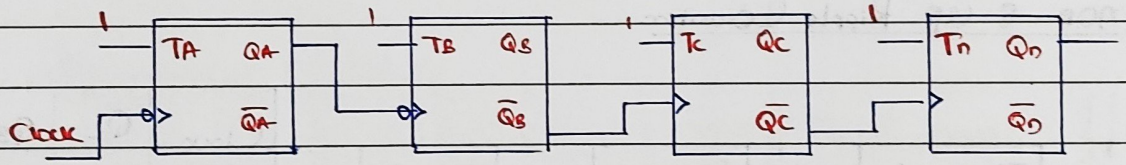
Q2.

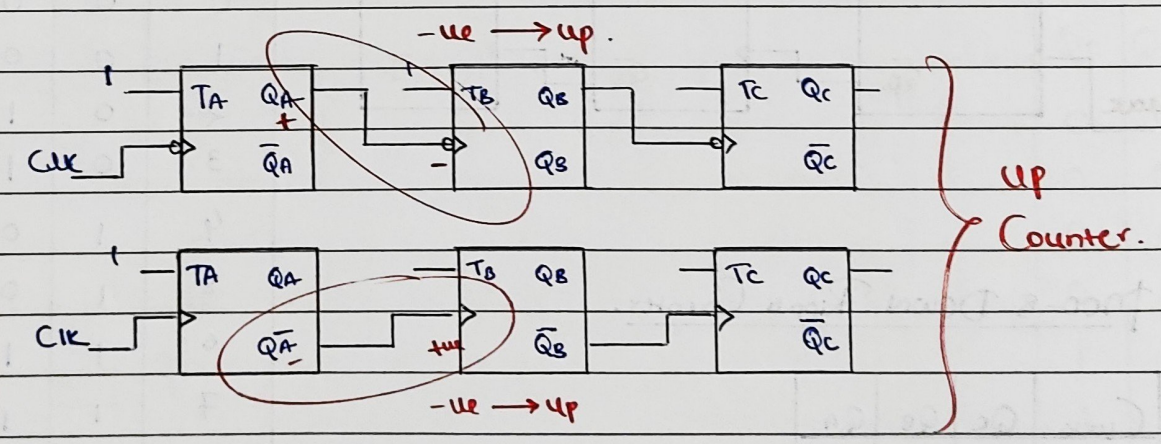


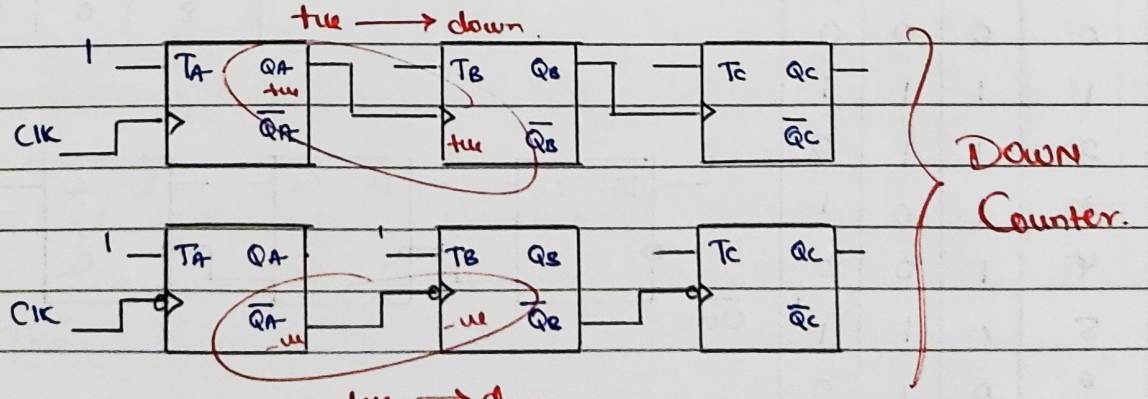
Q3.



Q4.







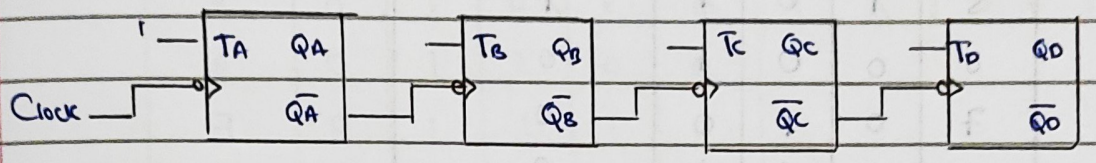
→ Full Mode Counter → when all the states are used by the counter.

n = 3 MOD = 8

n = 4 MOD = 16

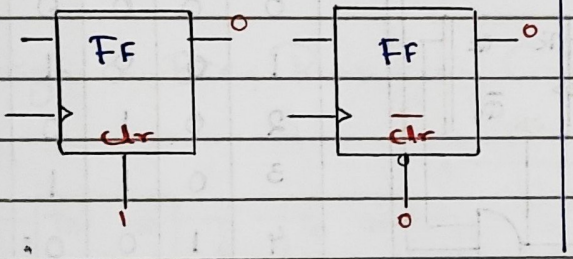
n = 5 MOD = 32

Q1. Design a MOD 16 down counter in which $\overline{Q}$ is taken as Clock.

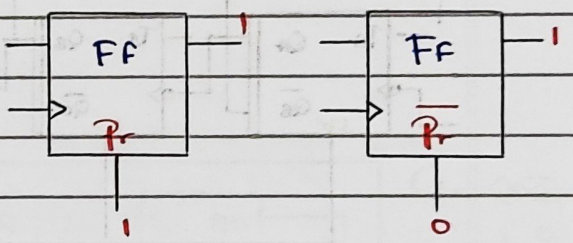


→ Feedback reduces the number of States:

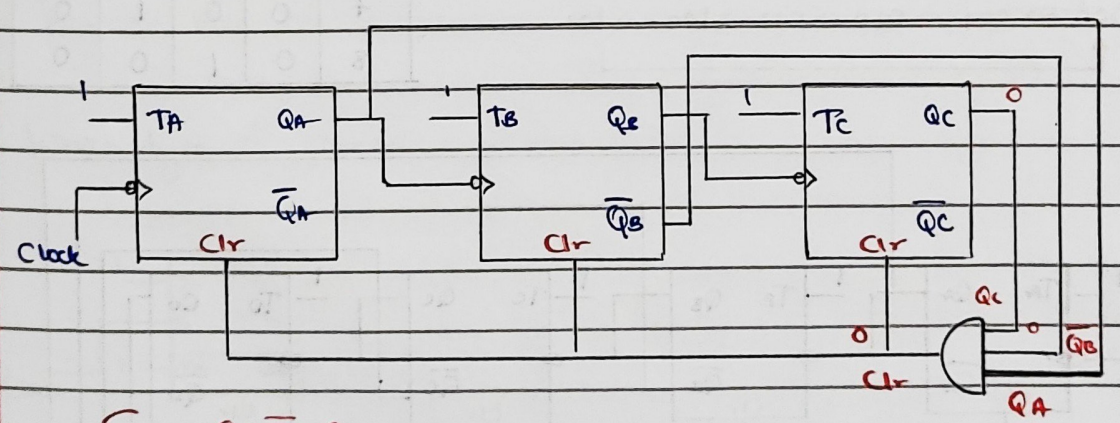
RESET — Clear.



PRESET



Feedback Reduces the Number of States.

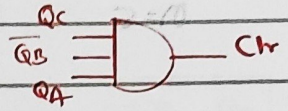


$Clr = Q_C \overline{Q_B} Q_A$

$= 1\ 0\ 1$

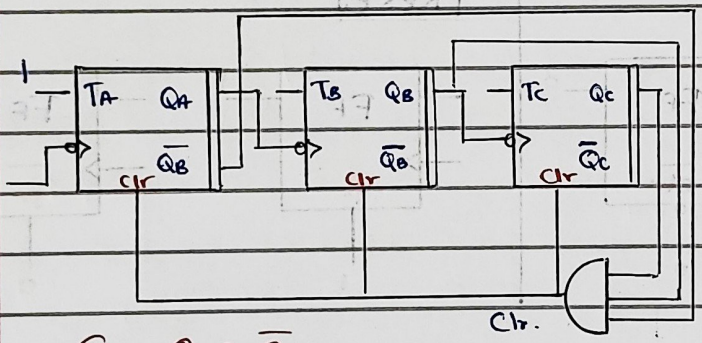
$= (5)$

Clock	Q_C	Q_B	Q_A	$CLR = Q_C\overline{Q_B}Q_A$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	↗0	↗0	↗0	↗0
6	0	0	1	0
7	0	1	0	0
8	0	1	1	0



MOD-6 UP Ripple Counter

$Clr = Q_C Q_B \overline{Q_A}$



$Clr = Q_C Q_B \overline{Q_A}$

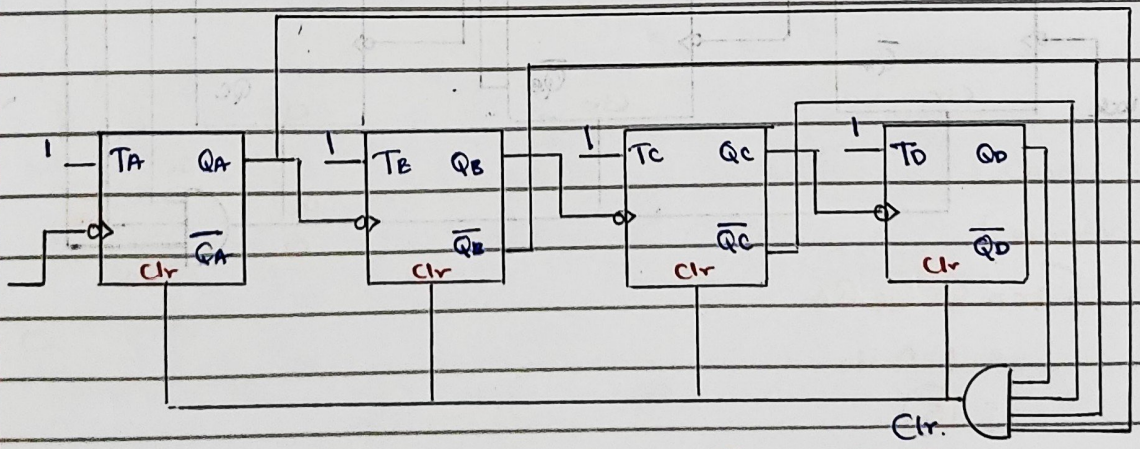
$Q_C Q_B \overline{Q_A}$

1 1 0 ⇒ 6

000 → 001 → 010 → 011 → 100 → 101

Clock	Q_C	Q_B	Q_A	Clr
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	↗0	↗0	↗0	↗0
7	0	0	1	0
8	0	1	0	0

Q.2



MOD-9 UP Ripple Counter.

$Clr = Q_D \overline{Q_C} \overline{Q_B} Q_A$

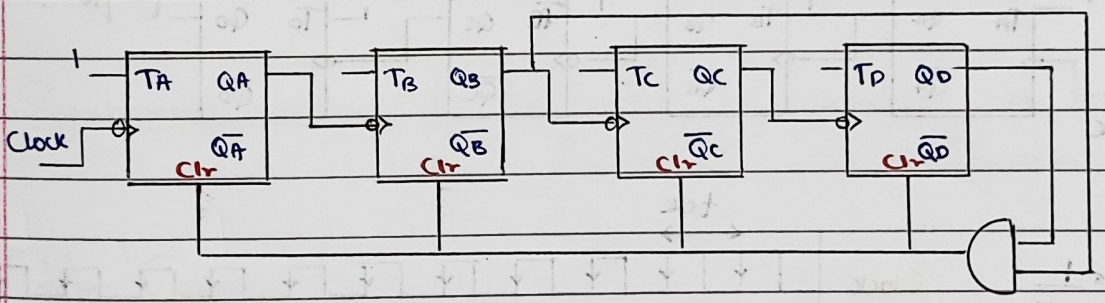
1 0 0 1 → (9)

Clock	Q_D	Q_C	Q_B	Q_A	Clr
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	↗0	↗0	↗0	↗0	↗0
10	0	0	0	1	
11	0	0	1	0	

$Clr = Q_D \bar{Q_C} Q_B Q_A$

1 0 0 1 ⇒ (9)

Q.3 Design a BCD Ripple Carry Counter.

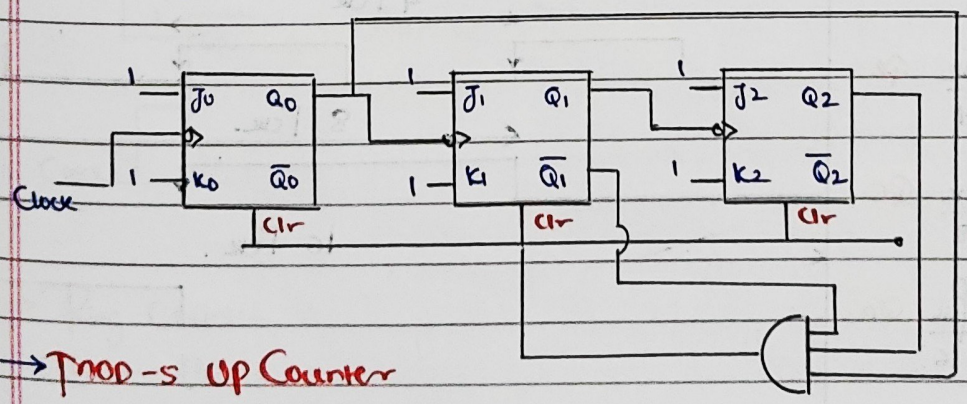


$Clr = Q_D \bar{Q_C} Q_B \bar{Q_A}$

1 0 1 0

⇒ (10)

Q4. Which type of counter is shown below?



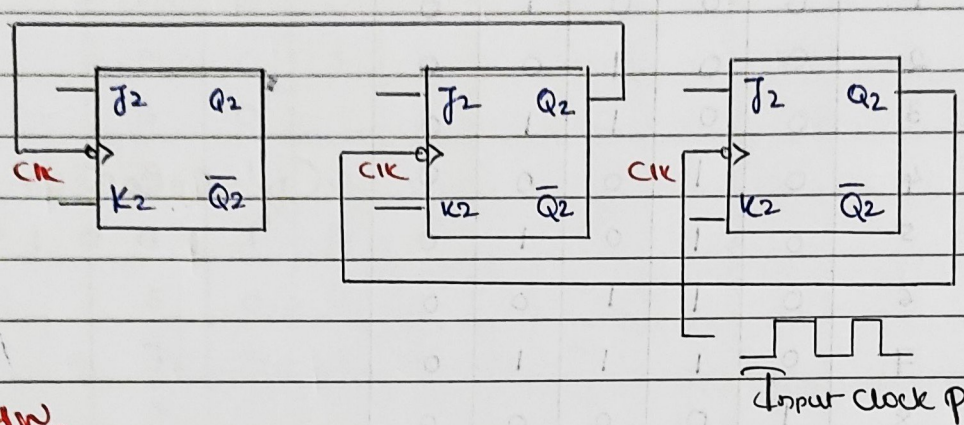
$Clr = Q_2 \bar{Q_1} Q_0$

1 0 1 → (5)

→ Mod-5 Up Counter

HW

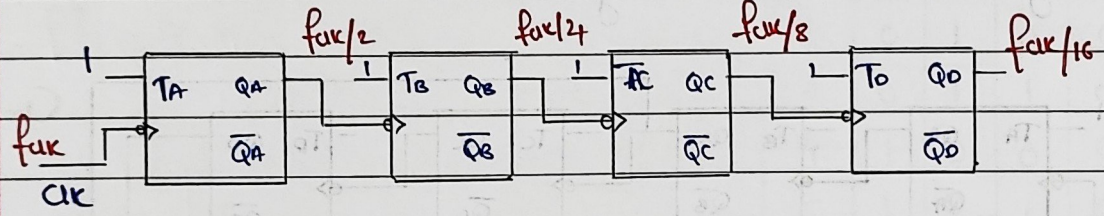
Qs. Consider the following clock. If Counter starts at 000, what will be count after 13 clock pulses?

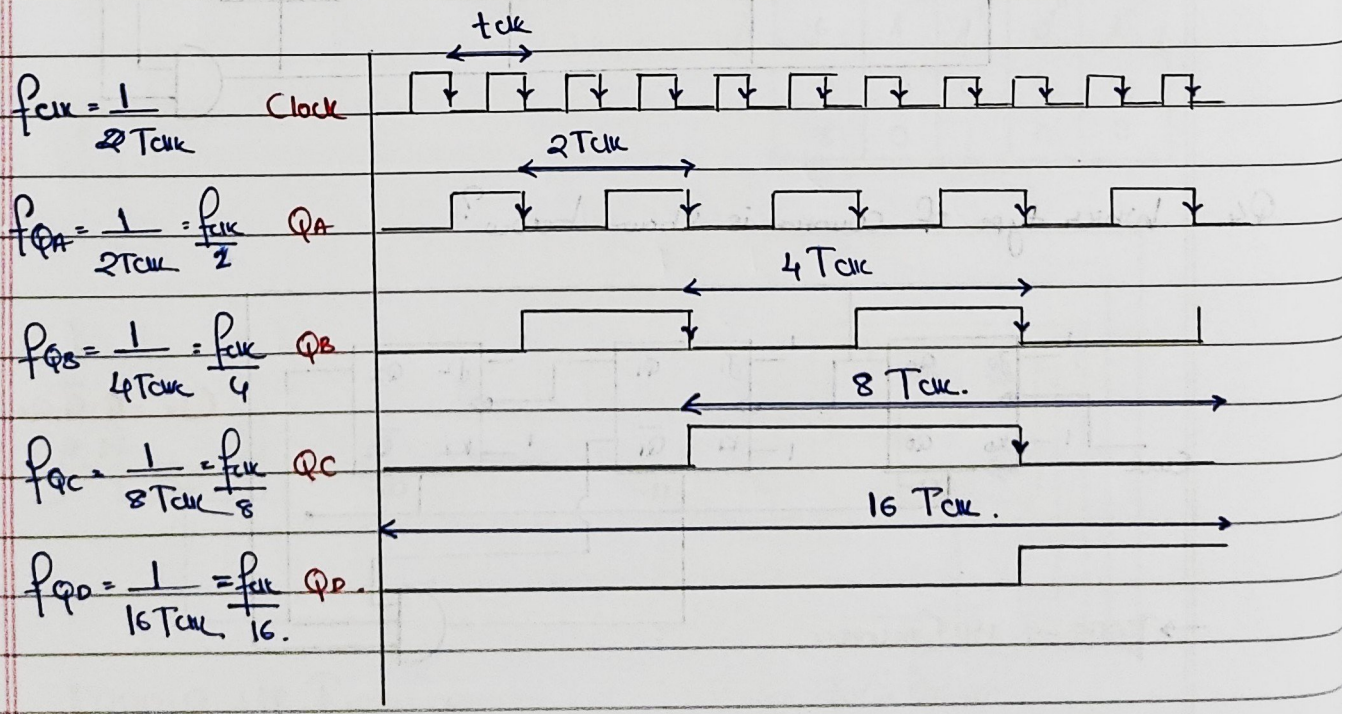


HW

Q6 Design MOD-13, MOD-21, MOD-27, MOD-14 (All UP) Ripple Counter.

Asynchronous [Ripple Counter].





$f_{clk} = \frac{1}{2T_{clk}}$ Clock

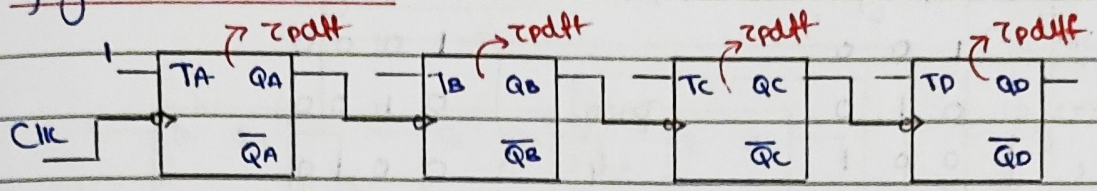
$f_{QA} = \frac{1}{2T_{clk}} = \frac{f_{clk}}{2}$ QA

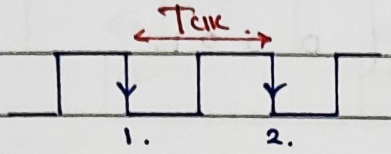
$f_{QB} = \frac{1}{4T_{clk}} = \frac{f_{clk}}{4}$ QB

$f_{QC} = \frac{1}{8T_{clk}} = \frac{f_{clk}}{8}$ QC

$f_{QD} = \frac{1}{16T_{clk}} = \frac{f_{clk}}{16}$ QD.

Asynchronous Counter.





$$T_{CLK} \geq 4\,\tau_{pdff}.$$

'N' bit Asynchronous counter

$$T_{CLK} \geq n.\tau_{pdff}.$$

$$(f_{CLK})_{max} = \frac{1}{n.\tau_{pdff}.}$$

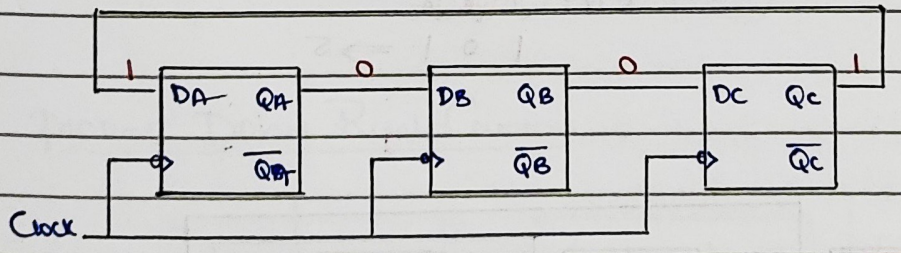
$$\frac{1}{T_{CLK}} \leq \frac{1}{n.\tau_{pdff}.}$$

$$f_{CLK} \leq \frac{1}{n.\tau_{pdff}.}$$

Synchronous Counter : All the FlipFlops are Synchronized with Same Clock
eg: Ring Counter, Jhonson Counter.

Ring Counter

→ It is also a SISO shift register in the form of Ring.



Clock	Q_A	Q_B	Q_C
0	0	0	0
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0

→ Ring Counter is not a self starting counter, to Start the ring Counter we have to place '1' at MSB then that '1' will rotate among all the Flip Flops.

3 bit Ring Counter

MOD = 3

1 0 0
0 1 0
0 0 1

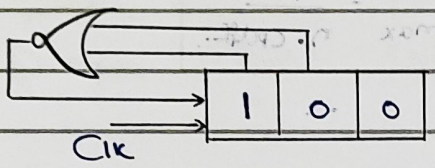
4 bit Ring Counter

MOD = 4

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

'N' bit Ring Counter
⇒ MOD = N

Self Starting Ring Counter

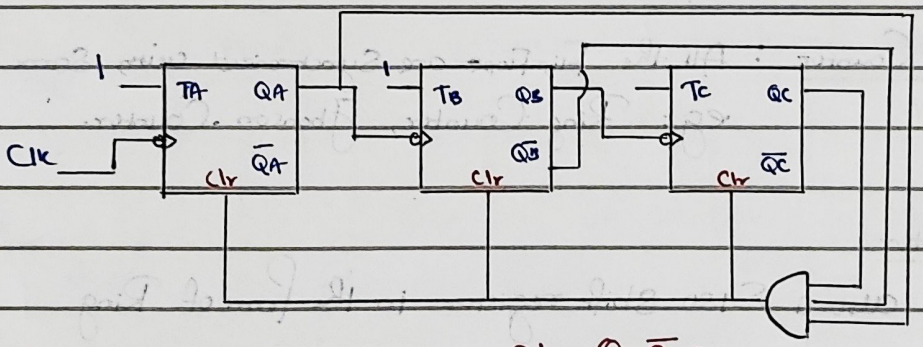


Clock	Q_A	Q_B	Q_C
0	0	0	0
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0

MOD = 3

{HW Assume of previous Class.}

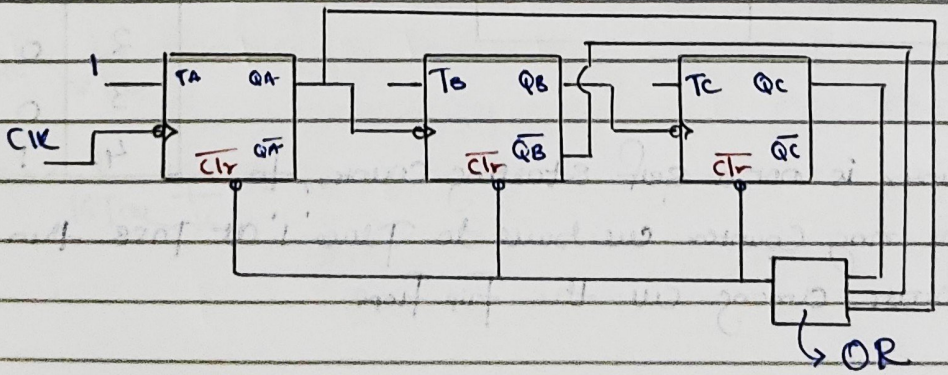
MOD-5 UP



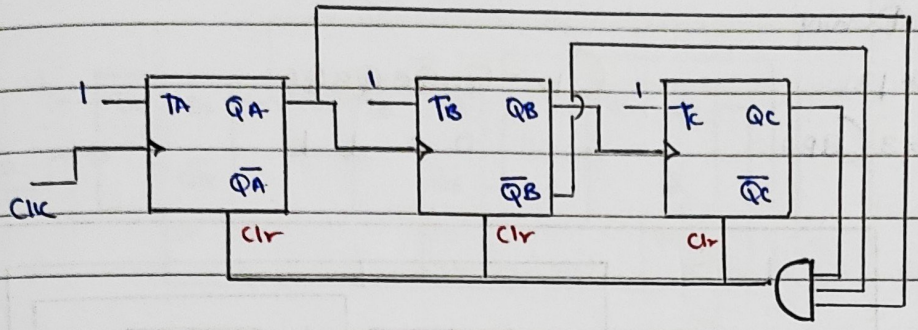
$Clr = Q_C \overline{Q_B} Q_A$

1 0 1 ⇒ 5

MOD-2 UP



MOD-3 DOWN Ripple Counter



CLK	Qc	QB	QA	Clr
0	0	0	0	0
1	1	1	1	0
2	1	1	0	0
3	✗	✗	✗	✗
4	1	1	1	0
5	1	1	0	0
6	✗	✗	✗	✗

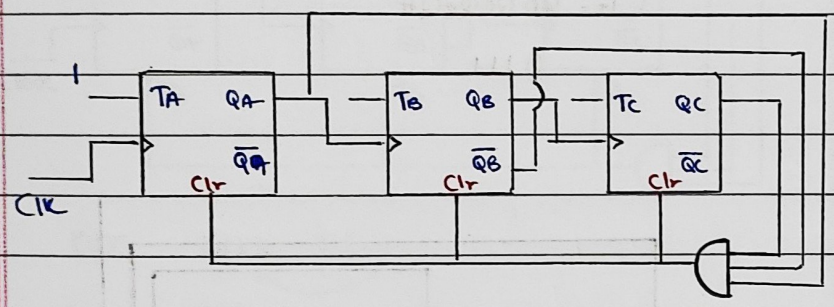
$111 \rightarrow 110 \rightarrow 000$ $\quad Clr = Q_C\bar{Q}_B Q_A$

Down = Full-up

= 8-5

= 3

MOD-4 DOWN Ripple Counter



Clock	Qc	QB	QA	Clr
0	0	0	0	0
1	1	1	1	0
2	1	1	0	0
3	1	0	1	0
4	✗	✗	✗	✗

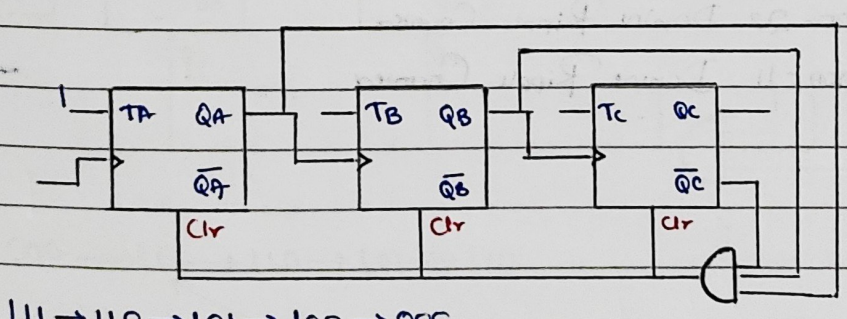
$111 \rightarrow 110 \rightarrow 101 \rightarrow 000$ $\quad Clr = Q_C\bar{Q}_B\bar{Q}_A$

100

Down = Full-up

= 8-4 → (4)

MOD-5 DOWN Ripple Counter



CLK	Qc	QB	QA	Clr
0	0	0	0	0
1	1	1	1	0
2	1	1	0	0
3	1	0	1	0
4	1	0	0	0
5	✗	✗	✗	1

$111 \rightarrow 110 \rightarrow 101 \rightarrow 100 \rightarrow 000$

Down = Full-up

(5) = 8-3

$\bar{Q}_C Q_B Q_A$

011 → 3.

Q1. Design a MOD-13 DOWN Ripple Counter.

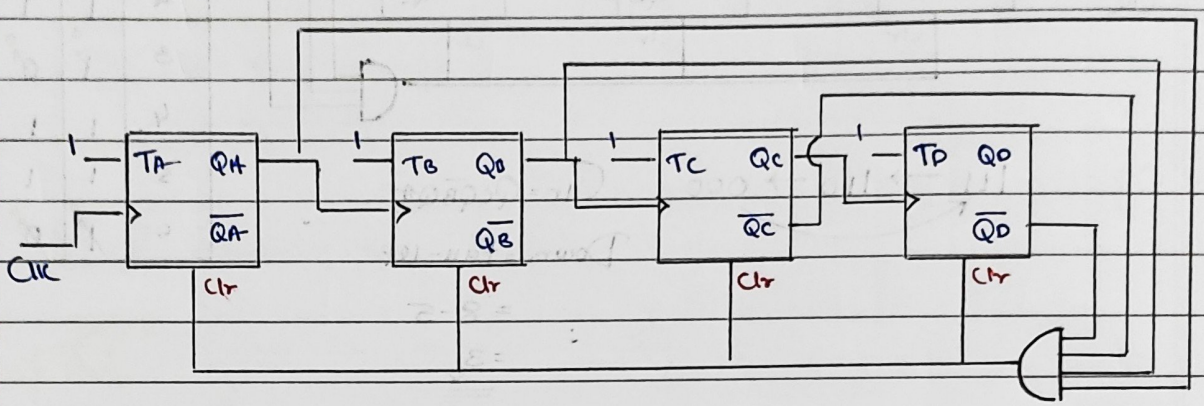
Full - UP = DOWN

16 - UP = 13

16 - 13 = 3 (UP)

$Clr = \overline{Q_D}\overline{Q_C}Q_BQ_A$

0 0 1 1



Q2. Design a MOD-9 DOWN Ripple Counter.

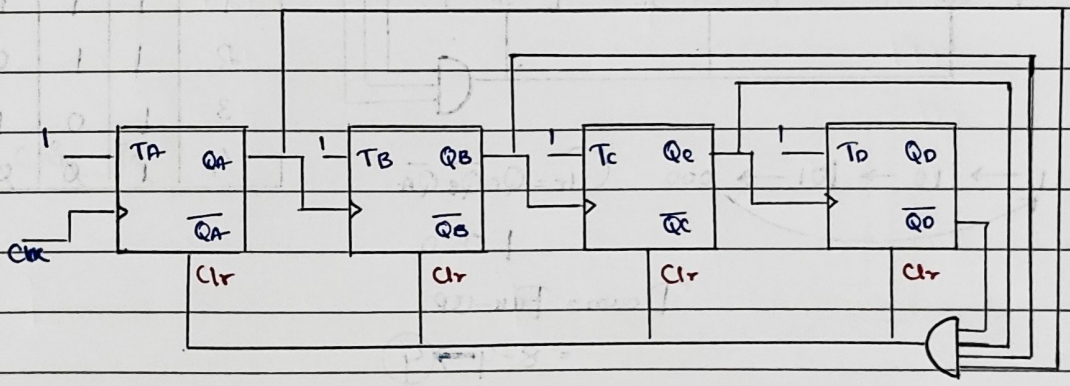
Full - UP = DOWN

16 - UP = 9

UP = 16 - 9 = 7.

$Clr = \overline{Q_D}Q_CQ_BQ_A$

0 1 1 1

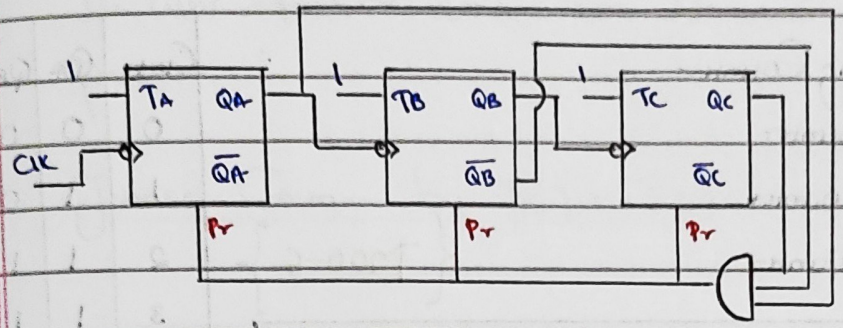


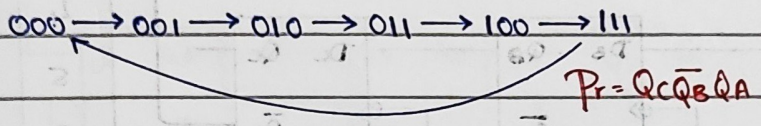
Q3. Design a MOD-19 DOWN Ripple Counter.

Q4. Design a MOD-23 DOWN Ripple Counter.

Q5. Design a MOD-11 DOWN Ripple Counter.

MOD-6

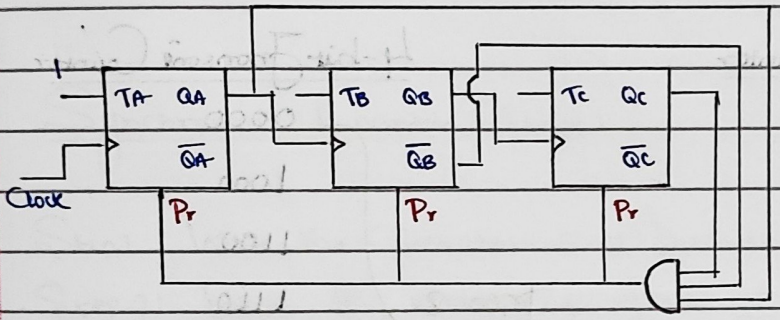


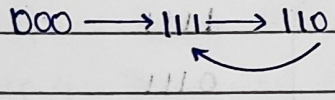


$P_r = Q_C\bar{Q}_BQ_A$

Clock	Q_C	Q_B	Q_A	P_r
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	$\not{1}'$	$\not{0}'$	$\not{1}'$	$\not{1}'$
6	0	0	0	0
7	0	0	1	0

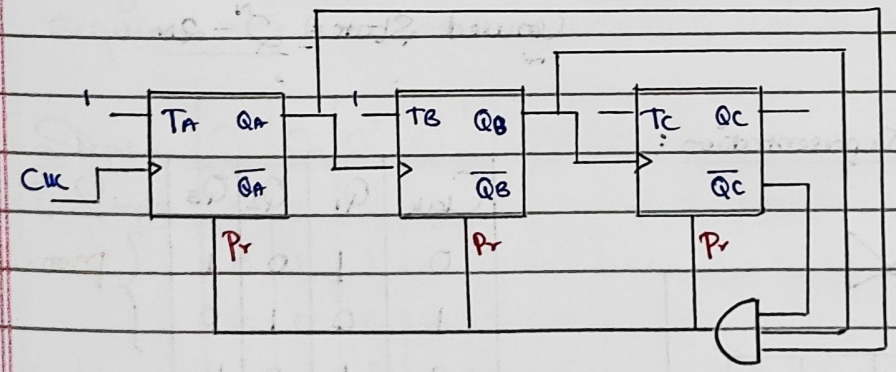
MOD-2

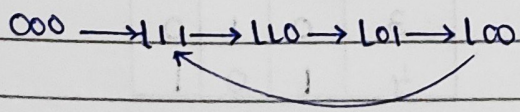




Clock	Q_C	Q_B	Q_A	P_r
0	0	0	0	0
1	1	1	1	0
2	1	1	0	0
3	$\not{1}'$	$\not{0}'$	$\not{1}'$	$\not{1}^0$
4	1	1	0	0
5	$\not{1}'$	$\not{0}'$	$\not{1}'$	$\not{1}^0$
6	1	1	0	0
7	$\not{1}'$	$\not{0}'$	$\not{1}'$	1^0

MOD 4



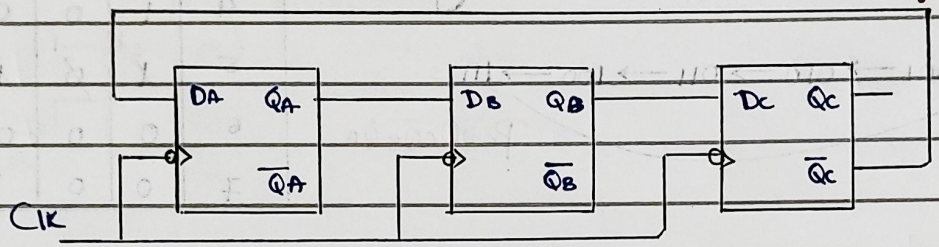


Clk	Q_C	Q_B	Q_A	P_r
0	0	0	0	0
1	1	1	1	0
2	1	1	0	0
3	1	0	1	0
4	1	0	0	0
5	$\not{0}'$	$\not{1}'$	$\not{1}'$	$\not{1}^0$
6	1	1	0	0
7	1	0	1	0

Johnson Counter

- Twisted Ring Counter
- Creeping Counter
- Mobius Counter
- Walking Counter

{Mod-6}



Clock	Q_A	Q_B	Q_C
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0
7	1	0	0

3 bit Johnson Counter

Mod 6:
000
100
110
111
011
001

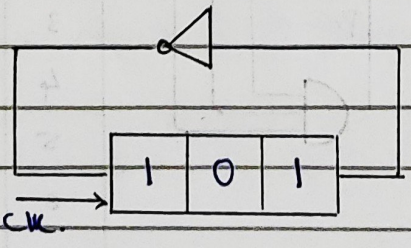
4-bit Johnson Counter

Mod-8:
0000
1000
1100
1110
1111
0111
0011
0001

→ N bit Johnson Counter ⇒ Mod (used State) = 2N

Unused State = $2^N - 2N$

Symbolic Representation:



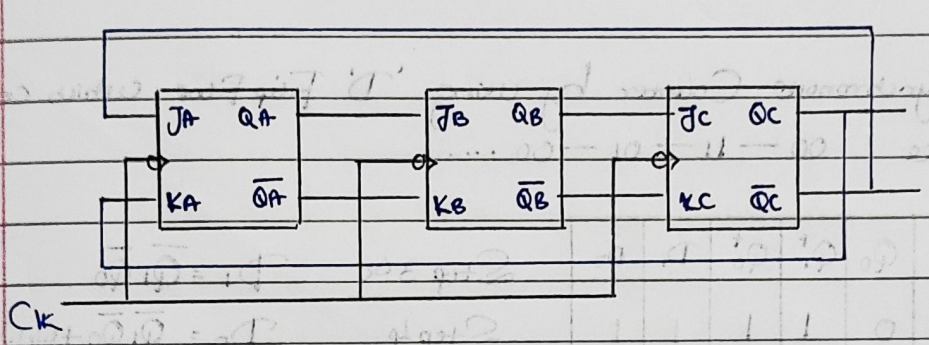
Lockout Problem!

Clock	Q_1	Q_2	Q_3
0	1	0	1
1	0	1	0
2	1	0	1
3	0	1	0
4	1	0	1

{Mod-2}

Lockout Problem: Whenever Johnson Counter enter its unused State, it will trap or lock into unused state, are called lockout problem.

Johnson Counter by Jk Flip Flop



Synchronous Counter Design:

Step 1: Write the Previous and Present State
Step 2: Write the excitation table of flip flop
Step 3: Write the logical expression
Step 4: Minimise the logical expression
Step 5: Hardware implementation

Q1. Design a Synchronous Counter using "T" Flip Flop which count the sequence $\Longrightarrow 0-3-1-2-0\ldots$ $\{00-11-01-10\rightarrow 00\ldots\}$

↳ 2 FFs are required.

Step 1 & Step 2

Q_1	Q_0	Q^+	Q_0^+	T_1	T_0
0	0	1	1	1	1
0	1	1	0	1	ϕ
1	0	0	0	1	0
1	1	0	1	1	0

Step 3 & Step 4:

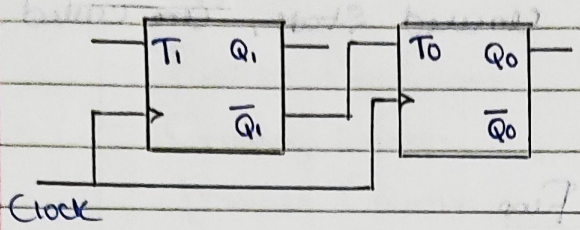
$T_1 = 1$

$T_0 = \overline{Q_1}\,\overline{Q_0} + \overline{Q_1}Q_0$

$T_0 = \overline{Q_1}\left[\overline{Q_0} + Q_0\right]$

$T_0 = \overline{Q_1}$

Step 5:



Justification:

Clock	Q_1	Q_0
0	0	0
1	1	1
2	0	1
3	1	0

Q2. Design a Synchronous Counter by using 'D' Flip Flop which count the Sequence 00 — 11 — 01 — 00 ....

Method 1

Step 1 & Step 2

Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0
0	0	1	1	1	1
1	1	0	1	0	1
0	1	0	0	0	0

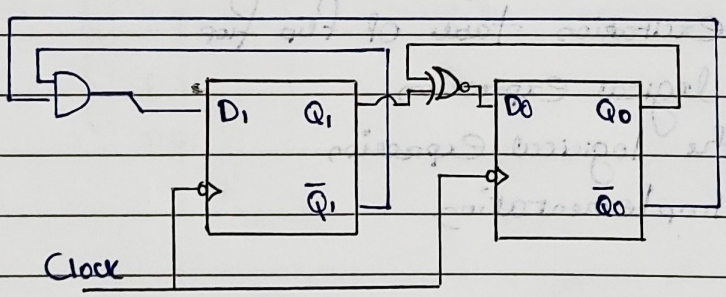
Step 3 & Step 4

$D_1 = \bar{Q}_1 \bar{Q}_0$

$D_0 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0$

$D_0 = Q_1 \odot Q_0$

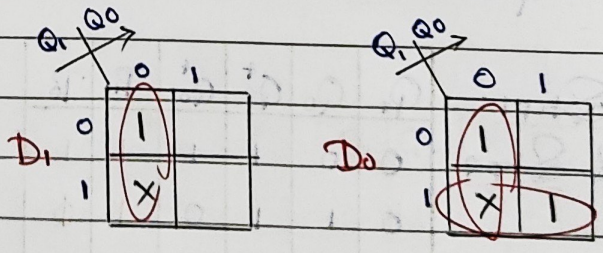
Step 5:



Clock	Q_1	Q_0
0	0	0
1	1	1
2	0	1
3	0	0

Method 2

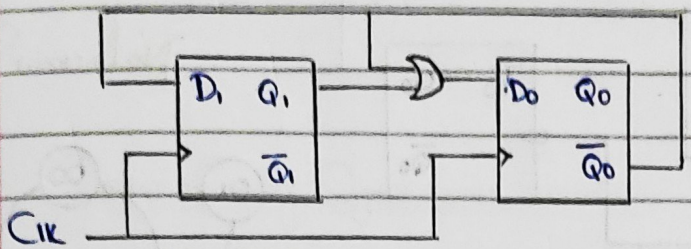
Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0
0	0	1	1	1	1
0	1	0	0	0	0
1	0	x	x	x	x
1	1	0	1	0	1



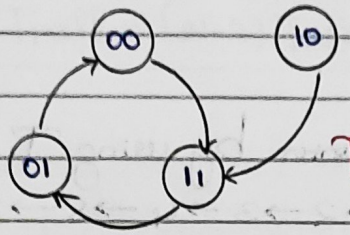
$D_1 = \bar{Q}_0$

$D_0 = Q_1 + \bar{Q}_0$

Date 23/7/23



Clock	Q_1	Q_0
0	0	0
1	1	1
2	0	1
3	0	0
4	1	1
5	0	1.



No Lockout!

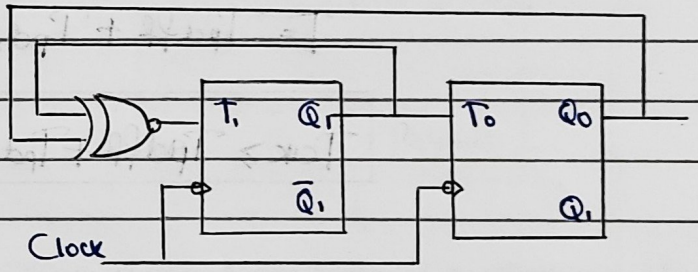
HW

Q3. Design a Synchronous Counter by using T-FF which count:
000 — 001 — 010 — 011 — 100 — 101 — 110 — 111 — 000 ....

Q4. Design a Synchronous Counter by using T-flip flop which count the Sequence 0 → 2 → 3 → 0 ...
00 → 10 → 11 → 00 2bit → 2 flip flops are required.

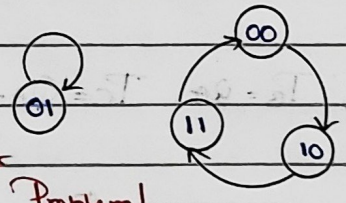
Method 1

Q_1	Q_0	Q_1^+	Q_0^+	T_1	T_0
0	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	1.



$T_1 = Q_1 \oplus Q_0$

$T_0 = Q_1$

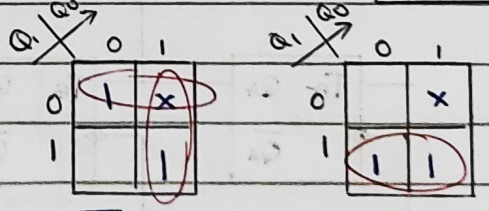


Lockout Problem!

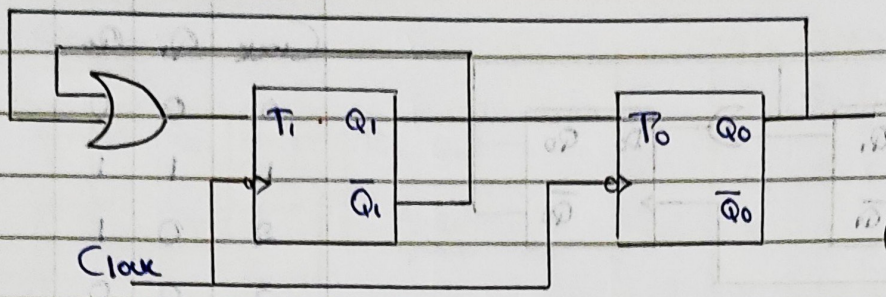
Clock	Q_1	Q_0
0	0	0
1	1	0
2	1	1
3	0	0

Method 2

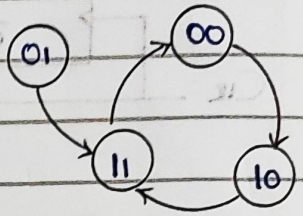
Q_1	Q_0	Q_1^+	Q_0^+	T_1	T_0
0	0	1	0	1	0
0	1	x	x	x	x
1	0	1	1	0	1
1	1	0	0	1	1



$T_1 = \overline{Q_1} + Q_0$ $T_0 = Q_1$



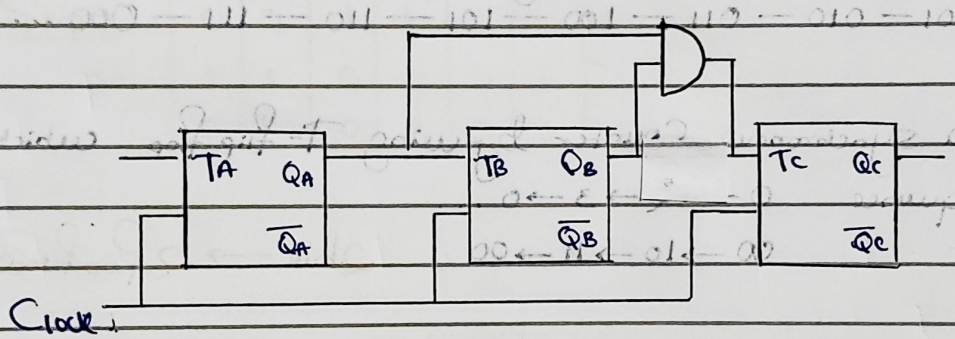
No Lockout Problem!



Q1. Design a Synchronous Counter by using T Flip Flop which count the sequence $0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 0$.

Three bit → Three Flip Flops Required

→ $T_A = 1$, $T_B = Q_A$, $T_C = Q_A Q_B$



$T = T_{pdff} + T_{pdAND}$

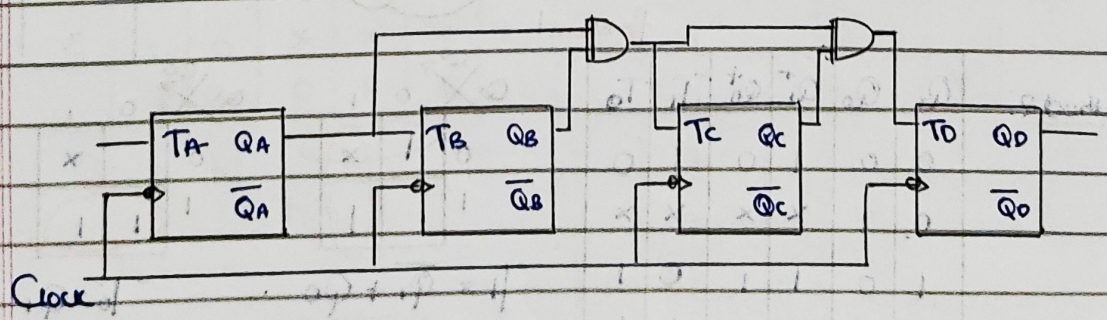
$\frac{1}{T_{CLK}} \leq \frac{1}{T_{pdff} + T_{pdAND}}$

$T_{CLK} \geq T_{pdff} + T_{pdAND}$.

$f_{CLK} \leq \frac{1}{T_{pdff} + T_{pdAND}}$

4 bit

$T_A = 1$, $T_B = Q_A$, $T_C = Q_A \cdot Q_B$, $T_D = Q_A \cdot Q_B \cdot Q_C$.



$$T_{CLK} \geq T_{pdff} + 2 \cdot T_{pdAND}$$

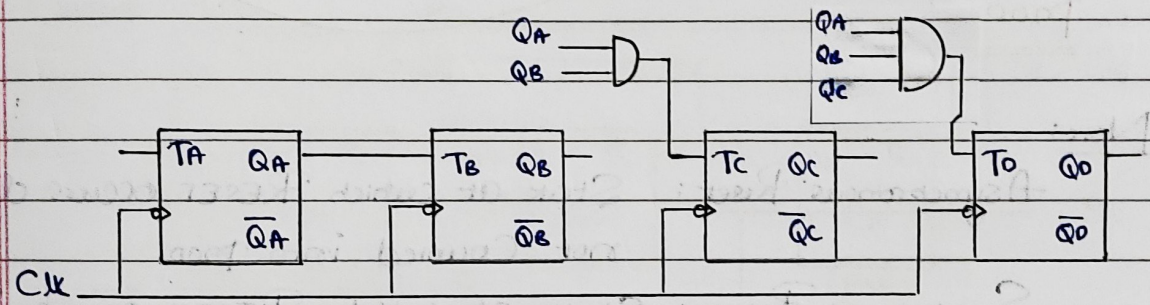
$$f_{CLK} \leq \frac{1}{T_{pdff} + 2\,T_{pdAND}}$$

Series Carry Synchronous Counter

$$T_{CLK} \geq T_{pdff} + (N-2) \cdot T_{pdAND}$$

$$f_{CLK} \leq \frac{1}{T_{pdff} + (N-2)\,T_{pdAND}}$$

Parallel Carry Synchronous Counter:



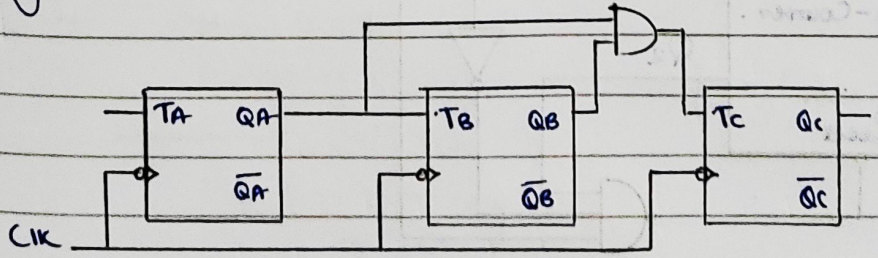
$$T_{CLK} \geq T_{pdff} + T_{pdAND}$$

$$f_{CLK} \leq \frac{1}{T_{pdff} + T_{pdAND}}$$

Q1. Minimum no. of flip flops required to construct BCD Counter is

Ans. 0000 } 4 bits → Four Flip Flops.
1001 }

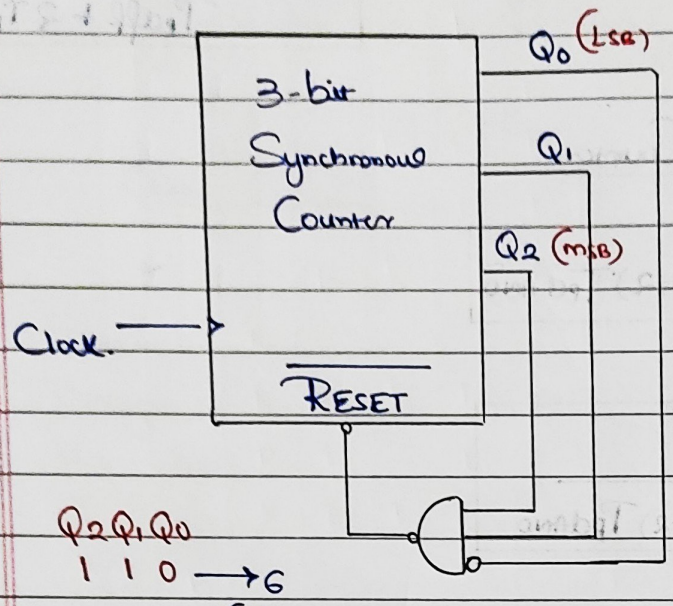
Q2. Minimum no. of Flip Flops required to design counter state given below $0 \to 0 \to 1 \to 1 \to 2 \to 2 \to 3 \to 3 \to 0 \to 0 \ldots$



3 bit Synchronous up Counter

⇒ 3 Flip Flops Required

Q3. What is the mod of the counter?



	Q_2	Q_1	Q_0
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	0	0	0

Mod-7

$Q_2Q_1Q_0$
1 1 0 → 6
(Reset)

0 → 1 → 2 → 3 → 4 → 5 → 6

∴ Mod = 7

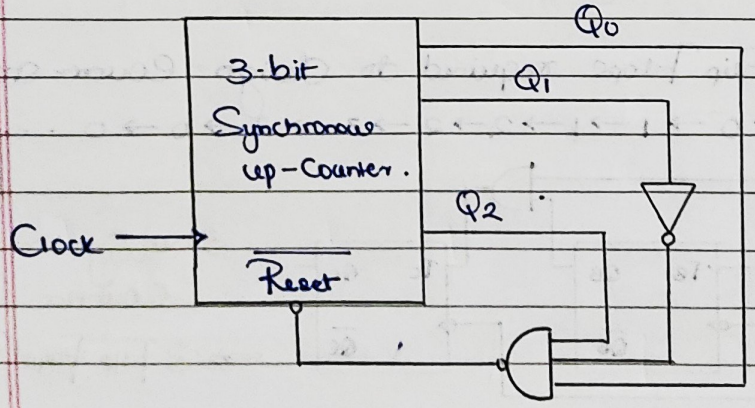
Note:

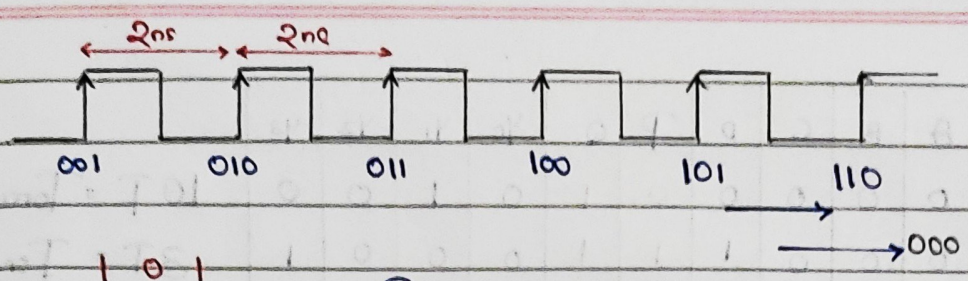
Asynchronous Reset: State at which RESET occurs does not Counted into Mod.

Synchronous Reset: State at which RESET occurs is counted into Mod.

Q4. Consider the below Circuit:

The delay of NAND, NOT, gate is 3ns, 1ns respectively and that of the Counter is assumed to be zero. If the clock frequency is 500 MHz, then the Counter behaves as:





$Clr = Q_2\bar{Q}_1Q_0$ (1 0 1)

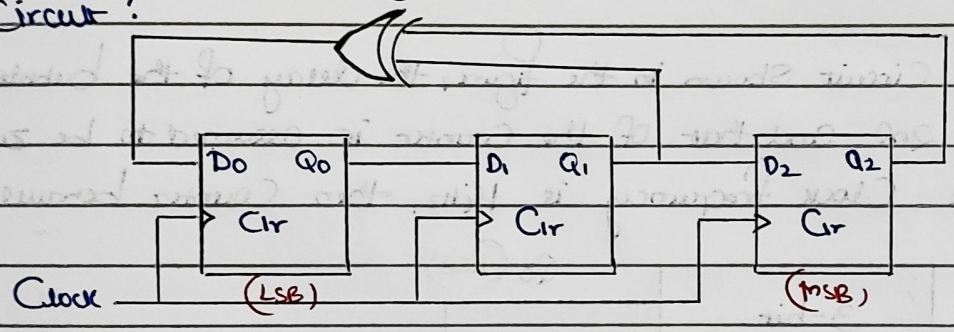
$\therefore$ Mod = 7

$f_{CLK} = 500 \times 10^6$ Hz

$T_{CLK} = \frac{1}{500 \times 10^6}$ Sec $= \frac{1000}{500} \times 10^{-9}$

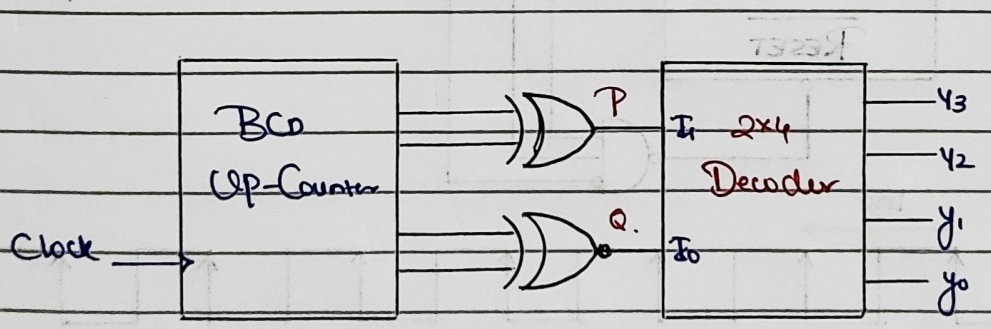
$= 2ns$

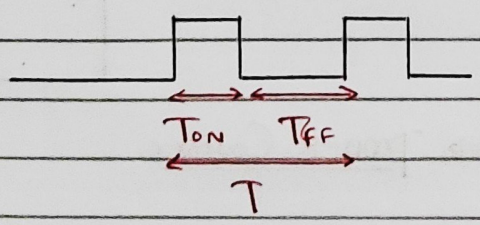
Q5. Consider the Circuit below with initial state $Q_0 = 1$, $Q_1 = Q_2 = 0$. The state of the Circuit is given by the value of $4Q_2 + 2Q_1 + Q_0$. Which of the following is the Correct State Sequence of the Circuit?



Sequence : $1 \to 2 \to 5 \to 3 \to 7 \to 6 \to 4$

Q6. Consider the Circuit given below. The duty cycle of Y_2 is





Duty Cycle $= \frac{T_{ON}}{T_{ON}+T_{OFF}} \times 100$

A	B	C	D	P	Q	Y_0	Y_1	Y_2	Y_3
0	0	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1
0	0	1	0	1	1	0	0	0	1
0	0	1	1	0	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	1	1	0	0	0	1	0
0	1	1	0	1	0	0	0	1	0
0	1	1	1	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0	1	0

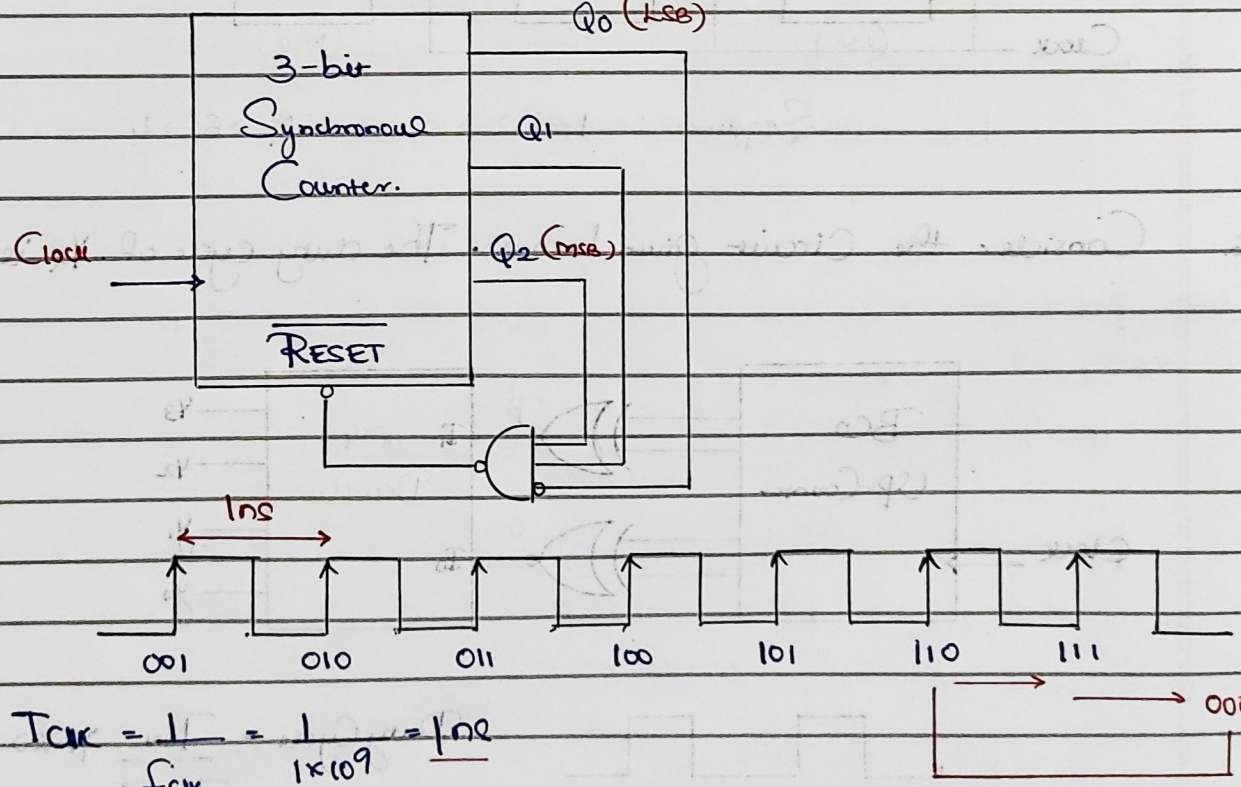
10T = Total time

3T = T_{ON}

$D = \frac{3T}{10T} \times 100$

$= 30\%$

Q7. For the circuit shown in the figure, the delay of the bubbled NAND gate is 2ns and that of the counter is assumed to be zero. If the clock frequency is 1GHz, then counter behaves as a:

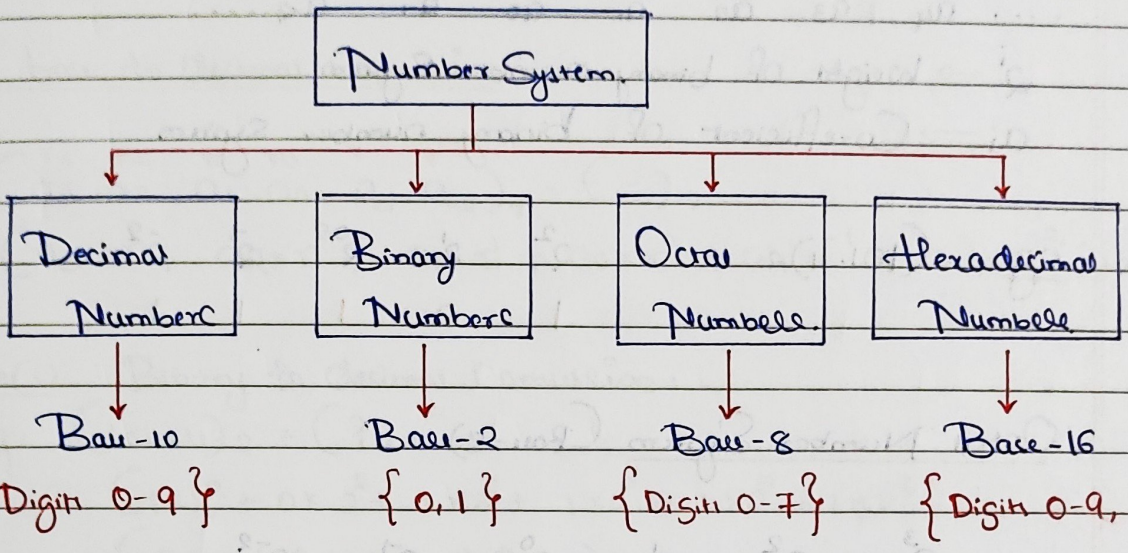


$T_{CK} = \frac{1}{f_{CK}} = \frac{1}{1 \times 10^9} = 1ns$

$\Rightarrow$ Mod-8 Counter

NUMBER SYSTEM

Base (Radix)
- Total number of digits used in the System.



Decimal Number System:

$$\ldots\ldots 10^{4} \quad 10^{3} \quad 10^{2} \quad 10^{1} \quad 10^{0} \quad 10^{-1} \quad 10^{-2} \quad 10^{-3} \ldots$$

$$\ldots\ldots a_4 \quad a_3 \quad a_2 \quad a_1 \quad a_0 \quad a_{-1} \quad a_{-2} \quad a_{-3} \ldots\ldots$$

$a_i \rightarrow$ Co-efficient of decimal number System.

$10^{i} \rightarrow$ Weight of decimal number system.

eg: $(501.23)_{10}$

10^2	10^1	10^0	10^{-1}	10^{-2}
5	0	1	2	3

Base	Digits
2	0 1
3	0 1 2
4	0 1 2 3
⋮	⋮
14	0 1 2 3 4 5 6 7 8 9 A B C D
15	0 1 2 3 4 5 6 7 8 9 A B C D E
16	0 1 2 3 4 5 6 7 8 9 A B C D E F

Binary Number System (Base = 2)

$\ldots\ 2^4\ \ 2^3\ \ 2^2\ \ 2^1\ \ 2^0\ \ 2^{-1}\ \ 2^{-2}\ \ldots$

$\ldots\ a_4\ \ a_3\ \ a_2\ \ a_1\ \ a_0\ \ a_{-1}\ \ a_{-2}\ \ldots$

$2^i \rightarrow$ Weight of binary number System.

$a_i \rightarrow$ Co-efficient of binary number System.

eg: $(101.11)_2$

2^2	2^1	2^0	2^{-1}	2^{-2}
1	0	1	1	1

Octal Number System (Base = 8)

$\ldots\ 8^3\ \ 8^2\ \ 8^1\ \ 8^0\ \ 8^{-1}\ \ 8^{-2}\ \ldots$

$\ldots\ a_3\ \ a_2\ \ a_1\ \ a_0\ \ a_{-1}\ \ a_{-2}\ \ldots$

$8^i \rightarrow$ Weight of Octal number System

$a_i \rightarrow$ Co-efficient of Octal number System $\{0-7\}$.

eg: $(720.64)_8$

8^2	8^1	8^0	8^{-1}	8^{-2}
7	2	0	6	4

Hexadecimal Number System (Base = 16)

$\ldots\ 16^3\ \ 16^2\ \ 16^1\ \ 16^0\ \ 16^{-1}\ \ 16^{-2}\ \ldots$

$\ldots\ a_3\ \ a_2\ \ a_1\ \ a_0\ \ a_{-1}\ \ a^{-2}\ \ldots$

$16^i \rightarrow$ Weight of Hexadecimal number System

$a_i \rightarrow$ Co-efficient of Hexadecimal number System $\{0-9, A-F\}$.

eg: $(A2C.F)_{16}$

16^2	16^1	16^0	16^{-1}
A	2	C	F

In base Conversion there are two key points:

1. Any base to decimal conversion.
2. Decimal to any other base conversion.

Any base to decimal Conversion:

$$(a_3\ a_2\ a_1\ a_0 . a_{-1}\ a_{-2})_r = (\ )_{10}$$

$$(a_3 \times r^3 + a_2 \times r^2 + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + a_{-2} \times r^{-2})_{10}$$

Case(1) : Binary to decimal Conversion.

eg: $(1011.11)_2 = (?)_{10}$

$= (1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2})$

$= (8 + 0 + 2 + 1 + 0.5 + 0.25)$

$= (11.75)_{10}$

Case (2) : Octal to Decimal Conversion

eg: $(721.4)_8 = (?)_{10}$

$= (7\times8^2 + 2\times8^1 + 1\times8^0 + 4\times8^{-1})$

$= 448 + 16 + 1 + 0.5$

$= (465.5)_{10}$.

Case (3) : HexaDecimal to Decimal Conversion.

eg: $(A2B.C)_{16} = (?)_{10}$

$= (A\times16^2 + 2\times16^1 + B\times16^0 + C\times16^{-1})$

$= (10\times256 + 2\times16 + 11\times1 + 12\times16^{-1})$

$= (2603.75)_{10}$.

Case (4) : Base 5 to Decimal Conversion

eg: $(432.22)_5 = (?)_{10}$

$= (4\times5^2 + 3\times5^1 + 2\times5^0 + 2\times5^{-1} + 2\times5^{-2})$

$= (100 + 15 + 2 + 0.4 + 0.08)$

$= (117.48)_{10}$.

Q1. $(62C) = (?)_{10}$

By minimum base Convert into decimal:

$(62C)$

↳ 13, 14, 15 , minimum = 13.

$= (6\times13^2 + 2\times13^1 + C\times13^0)$

$= (1052)_{10}$.

Decimal to any other base Conversion:

r	a_3 a_2 a_1 a_0	Remainder.
		b_0
		b_1
		b_2
		b_3

$a_3\ a_2\ a_1\ a_0 . a_{-1}\ a_{-2}$

Before decimal — After decimal

$0 . a_{-1}a_{-2}a_{-3} \times r = x_0 . x_{-1}x_{-2}$ → x_0

$0 . x_{-1}x_{-2} \times r = x_1 . x_{-3}x_{-4}$ → x_1

$0 . x_{-3}x_{-4} \times r = x_2 . x_{-5}x_{-6}$ → x_2

$(a_3a_2a_1a_0 . a_{-1}a_{-2}a_{-3})_{10} = (b_3b_2b_1b_0 . x_0x_1x_2)_r$

Case (1): Decimal to Binary Base Conversion:

eg: $(19.75)_{10} = (?)_2$

After decimal

2	19	1
2	9	1
2	4	0
2	2	0
	1	1

$0.75 \times 2 = 1.5$ → 1

$0.5 \times 2 = 1.0$ → 1

$(19.75)_{10} = (10011.11)_2$

Case (2): Decimal to Octal base Conversion:

eg: $(210.23)_{10} = (?)_8$

8	210	2
8	26	2
	3	3

$0.23 \times 8 = 1.84$ → 1
$0.84 \times 8 = 6.72$ → 6
$0.72 \times 8 = 5.76$ → 5

$(210.23)_{10} = (322.165)_8$

Case (3) : Decimal to Hexadecimal base Conversion.

eg: $(1228.55)_{10} = (?)_{16}$

16 | 1228	12 (C)	↑
16 | 76	12 (C)	↑
4	4	

$0.55 \times 16 = 8.8$ → 8
$0.8 \times 16 = 12.8$ → 12(C) ↓

$(1288.55)_{10} = (4CC.8C)_{16}$

Some Special Case

Case (1) : Binary to Octal base Conversion

eg: $(1011\,0111)_2 = (?)_8$

Octal = base 8 $\Rightarrow 8 = 2^3$. (∵ every 3 bit/digit of binary represent octal).

010 110 111
2 6 7 Hence : $(1011\,0111)_2 = (267)_8$.

Case (2) : Binary to Hexadecimal base conversion.

eg: $(1011011)_2 = (?)_{16}$

Hexadecimal = base 16 = 2^4 (∵ every 4 digits of binary → 1 digit of hexadecimal).

0101 1011
5 11(B) Hence $(1011011)_2 = (5B)_{16}$

Q2. $(330123)_4 = (?)_8$

Method 1 : $(330123)_4 = (\quad)_{10} - (\quad)_8$

Method 2 : $(330123)_4 - (\quad)_2 - (\quad)_8$

$4 = 2^2$.

$(\overline{111}\ \overline{100}\ \overline{011}\ \overline{011}) = (\quad)_8$

↳ 2^3

$(7433)_8$

BCD (Binary Coded Decimal) :

In this each digit is represented by its four-bit binary equivalent. It is also called as natural BCD or 8421-code and it is weighted code.

Excess-3 Code: This is a non weighted binary code used for decimal digits. Its code assignment is obtained from the corresponding value of BCD after the addition of 3.

BCO (Binary Coded Octal): In this each digit of the octal number is represented by its three-bit binary equivalent.

BCH (Binary Coded Hexadecimal): In this each digit of the hexa decimal number is represented by its four bit binary equivalent.

Decimal	BCD 8421	Excess-3	Octal	BCO	Hexadecimal	BCH
0	0000	0011	0	000	0	0000
1	0001	0100	1	001	1	0001
2	0010	0101	2	010	2	0010
3	0011	0110	3	011	3	0011
4	0100	0111	4	100	4	0100
5	0101	1000	5	101	5	0101
6	0110	1001	6	110	6	0110
7	0111	1010	7	111	7	0111
8	1000	1011			8	1000
9	1001	1100			9	1001
					A	1010
					B	1011
					C	1100
					D	1101
					E	1110
					F	1111

Q1. Find the base which satisfy the equation $\left[\left(\frac{39}{3}\right) = 13\right]_x$

Ans $\frac{(39)}{(3)_x} = (13)_x$

$\frac{3x+9}{3} = x+3$

$3x+9 = 3x+9$ → This equation satisfies for all values of x

$\therefore$ Any base > 9.

Q2 Find the base value which can satisfy the following two equations simultaneously

(i) $2+3=5$

(ii) $2\times4=10$

Ans

(i) $2_x + (3)_x = (5)_x$

$2\times x^0 + 3\times x^0 \neq (5\times x^0)$

$2+3=5$

$x > 5$

$x \geq 6$

(ii) $2\times4=10$

$(2)_x \times (4)_x = (10)_x$

$(2\times x^0)_x \times (4\times x^0) = (1\times x^1 + 0\times x^0)$

$2\times4 = x+0$

$x = 8$

$\therefore x = 8$ will satisfy both.

Q3 Find the number of solutions of x and y to satisfy $(43)_8 = (xy)_y$

Ans $y > x$

$(43)_8 = (xy)_y$

$(4\times8^1 + 3\times8^0)_{10} = (x.y + 0.xy^0)_{10}$

$35 = x.y$

$\therefore$ No. of solutions $= 2$

x	y	
35	1	
1	35	✓
5	7	✓
7	5.	

Q4. Find the number of solutions of x & y to satisfy $(123)_5 = (x8)_y$

Ans $y > 8$

$y > x$

$(123)_5 = (x8)_y$

$(1\times5^2 + 2\times5^1 + 3\times5^0) = xy^1 + 8xy^0$

$25+10+3 = xy+8$

$xy = 30$

$\therefore$ No of solutions $= 3$

x	y	
1	30	✓
2	15	✓
3	10	✓
5	6	

Q5. Result of addition of $34+43$ performed on minimum base is stored in an 8-bit register. The content will be

Ans $(34)_5 + (43)_5$

$(3\times5^1+4\times5^0)+(1\times5^1+3\times5^0)$

$= (42)_{10} \Longrightarrow 00101010$

Q6 $7\times512+3\times256+6\times128+5\times32+3\times16+3$

If the number will be written in binary then number is present will be?

Ans $2^0=1$ — 1 $\quad (4+2+1)2^9+(2+1)2^8+(4+2)2^7+(1+1)2^5+(3+1)2^4+2+1$

$2^1=10$ — 1 $\quad 2^{11}+2^{10}+2^9+2^9+2^8+2^9+2^8+2^7+2^5+2^5+2^4+2^1+2^0$

$2^2=100$ — 1 $\quad 2^{10}+2^9+2^{10}+2^7+2^6+2^4+2^1+2^0$

$2^3=1000$ — 1 $\quad 2^{10}+2^{10}+2^7+2^6+2^4+2^1+2^0$

$\Rightarrow \underline{\underline{7}}$

Magnitude Representation

- Unsigned
- Signed
- Compliment
  - 1's Compliment
  - 2's Compliment

	Unsigned	Signed	1's Compliment	2's Compliment
+5	101	0101	0101	0101
-5	x	1101	1010	1011

$r=10$

$r-1$ (9's Compliment) $\quad r$ (10's compliment)

eg. 732 find 9's Compliment

$$\begin{array}{r} 999 \\ 732 \\ \hline 267 \end{array}$$

Unsigned Representation

3 bit Number

Range : 0 to 7.

0 to 2^3-1.

$(n=3)$.

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Signed Representation :

4 bit number : $\{2^{4-1}-1\}$ to $\{2^{4-1}-1\}$.

Signed Number	Equivalence decimal
0000	+0
0001	+1
0010	+2
0011	+3
0100	+4
0101	+5
0110	+6
0111	+7

Signed Number	Equivalence decimal
1000	-0
1001	-1
1010	-2
1011	-3
1100	-4
1101	-5
1110	-6
1111	-7

1's Compliment Representation :

4 bit number : -7 to +7.

'n' bit

$-\{2^{n-1}-1\}$ to $+\{2^{n-1}-1\}$

1's Compliment	Equivale. Decimal
0000	+0
0001	+1
0010	+2
0011	+3
0100	+4
0101	+5
0110	+6
0111	+7

1's Compliment	Equivalence dec.
1000	-7
1001	-6
1010	-5
1011	-4
1100	-3
1101	-2
1110	-1
1111	-0

Q1. 001101. ⟶ +13

Q2. 110010 ⟶ -13

001101 ⟶ +13

2's Compliment Representation. 'n' bit

4 Bit Number : -8 to +7. Range : $-\{2^{n-1}\}$ to $+\{2^{n-1}-1\}$

2s Compli.	Equivalence dec.
0000	+0
0001	+1
0010	+2
0011	+3
0100	+4
0101	+5
0110	+6
0111	+7

2s Compli.	Equivalence dec.
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

Q1. Convert the number given in 1's Compliment into Equivalence Decimal.

1. 01101 → 13
2. 001101 → 13
3. 0000 1101 → 13
4. 1100 10 → -13
5. 11110010 → -13
6. 11101010 → -21
7. 0000 → +0
8. 1111 → -0

Q2. Convert the number given in 2's Compliment into Equivalence Decimal?

1. 01101 → +13
2. 001101 → +13
3. 0000 1101 → +13
4. 1100 10 → -14.
5. 11110010 → -14
6. 11101010 → -22
7. 0000 → 0
8. 1111 → -1

Q3. F's Compliment of $(2BFD)_{hex}$ is D402

FFFF
2BFD
D402

Q4. A number is expressed in binary two's Compliment as 10011. Its decimal equivalent value is -13

Q5. The greatest negative number, which can be stored in a computer that has 8-bit word length and uses 2's compliment arithmetic is.

Ans $-\{2^{n-1}\} = -2^{8-1} = -128$

Q6 What is the resultant of $(C4)_{16} - (7B)_{16}$?

Ans

Q7. The number of bytes required to represent the decimal number 185637 in packed BCD (Binary Coded Decimal Form) is ____

Q8. 11001, 1001, and 111001 correspond to the 2's compliment representation of cubics of the following sets of numbers?

Ans -7, -7 and -7 respectively.

Q9. P, Q, and R are the decimal integers corresponding to the 4 bit binary number 1100 considered in signed magnitude, 1's compliment and 2's compliment representation respectively. The 6-bit 2's compliment representation of (P+Q+R) is 110101

Ans. P = -4

Q = -3, R = -4 P+Q+R = -11

001011 ← 2's → 110101

Q10 What is the possible base of the given number system. $\sqrt{41} = 5$.

Ans $(\sqrt{41})_x = (5)_x$ $x > 5$

$\sqrt{4x+1} = 5$

$4x+1 = 25$

$4x = 24$ $\therefore x = 6$

Q11. -13 in 2's compliment will be -?

+13 = 01101

-13 = 10011

In 2's compliment: 110011

Q12. Let us consider the following equation in a 6-bit binary number System $X = A + BA$ is given A. Point as $(001010)_2$ in 1's Compliment binary number System, B is given as $(111010)_2$ in signed number System what would be x in 2's Compliment number System?

Ans $A = 10$ $\quad X = A + BA$

$B = -26$ $\quad = 10 + (-26) \times 10$

$= -250$

01111010 : 250

10000110 : -250

Q13. -24 in 2's Compliment form is 11101000

00011000 → +24

11101000 — -24