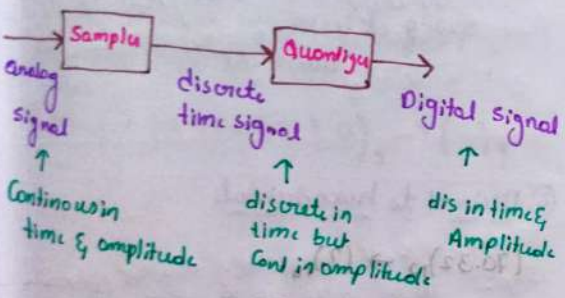
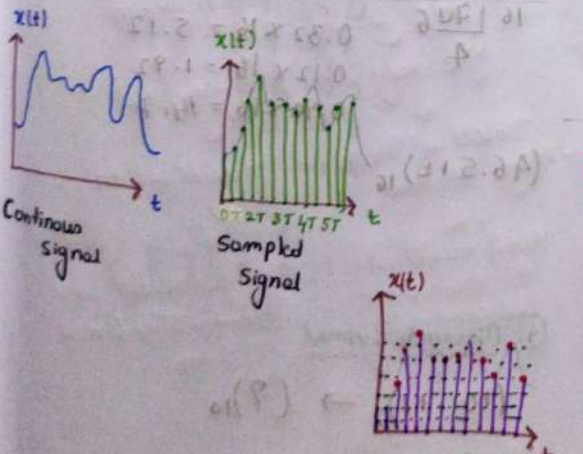
DIGITAL ELECTRONICS :- 20/02/25

- Digital has been derived from the word digit, represents the system with the finate number of possibilities
- Analog is similar to real world signal and infinate number of possibility
- Analog is more Natural, so it need to convert by data Converters.
- Conversion by digital is more powerful and required less storage, accuracy, noise resistant.
- High speed, Easy measurment, Security

How digital Signal obtained ?





We approximate value of each sample to the nearest allowed level
↳ introduce an error called as Quantization.

Δ = Step size

max Error = Δ/2.

- To minimize quantization error we must reduce Δ.
  ↳ noise margin ↓

NUMBER SYSTEM

A number system with base 'b' will have b different digits, ranging from 0 to b-1

NOTE:- • b must not Equal to 0
• b must be positive integer.

① Other base System to Non-decimal System.

Steps :-

ⓐ Convert original number to the decimal.

ⓑ Convert the decimal to the required base.

Example: $(101101)_2 \rightarrow (?)_3$

$1\times2^0+1\times2^2+1\times2^3+1\times2^5 = (45)_{10}$

3	45	0
3	15	0
3	5	2
	1	

$(1200)_3$

No of bit required formula by rounding off the number is given by

$$\log_2(N)$$

Max of n = n-1 = max of Coffiient

Group of 4 bit = nibble
8-bit = Byte
16 bit = Word
32 bit = double Word

① Decimal to binary

$(70.32)_{10} \rightarrow (?)_2$

2 | 70	0	$0.32 \times 2 = 0.64$
2 | 35	1	$0.64 \times 2 = 1.28$
2 | 17	1	$0.28 \times 2 = 0.56$
2 | 8	0	$0.56 \times 2 = 1.12$
2 | 4	0	
2 | 2	0	
1		

$(1000110.0101)_2$

② Decimal to octal

$(70.32)_{10} \rightarrow (?)_8$

8 | 70	6	$0.32 \times 8 = 2.56$
8 | 8	0	$0.56 \times 8 = 4.48$
1		

$(106.24)_8$

③ Decimal to hexadecimal.

$(70.32)_{10} \rightarrow (?)_{16}$

16 | 70	6	$0.32 \times 16 = 5.12$
4		$0.12 \times 16 = 1.92$
		$0.92 \times 16 = 14.7$

$(46.51E)_{16}$

④ Binary to decimal

$(101.11)_2 \rightarrow (?)_{10}$

$= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + \frac{1}{2} + \frac{1}{4}$

$= 4 + 0 + 1 + 0.5 + 0.25$

$= (5.0.75)_{10}$

⑤ Binary to octal

Group 3 bit and write convenient number

⑥ Binary to hexa

Group four four bit and then write its Equivalent

NOTE:- While grouping to Convert the number from binary to hexadecimal or to octal.

Real → left to right

Fraction → Right to left

Note-2:- From $(\ )_3 \rightarrow (\ )_9$

$3^2 = 9$

With this relation we need to group then and write the result

$(02\,1\,2\,1\,01.22\,10)_3 \rightarrow (\ )_9$

For base 3

$3^1\ 3^0 =$ 3 1

0 0 → 0
0 1 → 1
0 2 → 2
1 0 → 3
1 1 → 4
1 2 → 5
⋮

3 | 5 | 2 ↑
  | 1 |

hence (2771.83)

NOTE-3:- From higher to lower base

$(\ )_9 \rightarrow (\ )_3$

$(783.437) \rightarrow (?)_3$

$(21\,22\,10.11\,10\,21)_3$

Problem-1:- $(123)_5 = (x8)_y$

$25 + 10 + 3 = xy + 8$

$38 = xy + 8$

$xy = 30$

$y > x$ & $y > 8$

hence $y = 6$ & $x = 5$ ✗

$1 \times 30 = 30$
$2 \times 15 = 30$
$3 \times 10 = 30$

Problem :-2 $\frac{312}{20} = 13.1$ find no of x

$\frac{(312)_x}{(20)_x} = (13.1)_x$

$= \frac{3x^2 + 1x + 2}{2x} = 1x + 3 + x^{-1}$

$= 2x^2 + 6x + 2$ (RHS)

$x^2 - 5x = 0$

$x(x-5) = 0$

$x = 0$ ✗ $x = 5$

KB = 2^{10} Byt
MB = 2^{20} Byt
GB = 2^{30} Byt

L-2

24/02/25

Negative Number representation

- Signed
- (r-1)' Comp — 1's Comp
- 2's Complement — r's Complement

Object	Range
1) Unsigned number	$0 - 2^n - 1$
2) Signed	$-(2^{N-1}-1)$ to $(2^{N-1}-1)$
3) Unsigned no of bits	$\log_2 N$
4) 1's Complent	$-(2^{N-1}-1)$ to $(2^{N-1}-1)$
5) 2's Complemt	$-(2^{N-1})$ to $(2^{N-1}-1)$

1 → represents -Ve
0 → represents +Ve

Limitation of Sign magnitude

+0 → 0000
-0 → 1000

Since both should have same binary value but here it is different, this can be solved by Signed mag by 2's Complement

NOTE: ① 2's Complemint = 1'Complent + 1.
② Scale bit 10010 → (2's) 01110.

Advantages of 2's Complement :-

① 0 is uniquely represented
② Extended range
③ No need additional hardware while performing A-B operation.

8's Complement & 7's Complement of (3674)8

7777
3674
4103 → 7's Complement
+ 1
4104 → 8's Complement

NOTE:- Every Computer uses 2's Complement representation.

While performing addition operation try to choose the N values which has higher among them.

Trick

1 1 1
1 1 1 1
1 0 0 1 1
1 1 0 1 0
1 1 0 1 0
1 1 0 1 1
1 0 0 1 0

- if even no of 1's = 0
  odd no 1's = 1
- no of pairs = Carry

Addition of the form A - B

Step 1:- take 1's complemt of negative number
Step 2:- Add both the number

- if Carray → result will be not negative (+Ve) → Final result will be obtained by adding Carry
- Not generated → result will be -Ve. → Final result will be 1's Complemint of Sum

5
-4 } N=4
1

0101
0100
1's 1011
0 0000

0000
1
0001

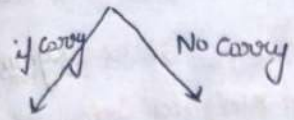
4
-5
-1

0100
0101
1's 1010
1110

No Carray
-(0001)
= -1

Substraction by 2's Complement

step1:- take 2's of neg number
step2:- Add both number



+Ve
discard the carry

negative
Find num = -(2's Complement)

Sub | Addition is of the form :-

- A
- B

Step1:- Choose Correct No of bit
step2:- Take 2's complent of both num
Step 3:- It will always generate a carry it should be discarded
step4:- Result is Equal to -(2's Complemt of final)

Addition ( )$_8$

$$(4327)_8$$
$$(3745)_8$$
$$(10274)_8$$

$(11)_8$ → $(9)_8$ Every carry no also need to Convert into Octal.
$(7)_8$ → $(7)_8$
?

2's Complemt Advantage :-

① Range is Expended
② No need Extra adder.
③ -0 & +0 ambigity has been resolved.

L-3 25/02/25

1KB = 2^{10} bytes
2^{13} bits

1MB = 2^{10} K bytes
2^{20} bytes
2^{23} bits

1GB = 2^{10} M bytes
2^{30} bytes
2^{33} bits

Binary Codes

- Weighted Codes
  - Each position has its own weight
  - BCD
    - 8421
    - 2421
    - 5211
    - 84-2-1
    - default value
- Non weighted Codes
  - No weight
  - ① Excess 3
  - ② Gray Code

ASCII → 8bit
EBCD → 9bit representation

BCD :-
① Very popular and easy to represent
② Varies from (0-9)
③ Each digit is group of 4 bit

Excess-3.

0	0011
1	0100
2	0101
:	
9	1100

- Additional feature of Excess 3 is Self Complement.

Self Complement:- Code whose 9's Complement can be obtained by replacing 0 with 1 and 1 with 0

- It is used because it has symmetry
- It is for non-weighted code

NOTE:- Self complement in BCD is applicable for non weighted code, in weighted code it is sum of digith = 9.

Eg :- 2421 = 2+4+2+1 = 9
5211 = 9
84-2-1 = 9

①

4 → 0111 (Excess-3.)

9's Complement 1000

5 ——— Excess 3

② 435 Excess 3 9's Complement

0111 0110 100
1000 1001 0111

```
  999
  435
  ---
  564
+ 333
  ---
  897
  ---
```

123 in binary

2	123	1
2	61	1
2	30	0
2	15	1
2	7	1
2	3	1
	1	

• Excess-3 is self complementing code

	8421	2421	5211	84-2-1
0	0000	0000	0000	0000
1	0001	0001	0001	
			0010	
2	0010	1000	0100	
		0010	0011	
3	0011	1001	0110	
		0010	0101	
4	0100	1010	0111	
		0100		
5	0101	1011	1000	
6	0110	1100	1001	
		0110	1010	
7	0111	1101	1100	
		0111	1011	
8	1000	1110	1101	
			1110	
9	1001	1111	1111	

Question :- What are the different ways which 5 can be represented in 2421.

Ans 2 ways { 1011
0101 }

Question :- No of unique representation of 2421

{9, 0, 1, 8}

And it is also self complement.

Question :- Unique representation of 5211

{0, 4, 5, 9}

Gray Code :- A code having a hamming distance of 1, is called gray code

Application :- 1. used in lower power device
2. used in FIFO and K. Map

Binary to gray :-

$(B_2 \oplus B_1 \oplus B_0)$

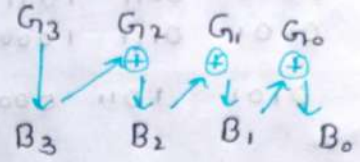
$(G_2 \quad G_1 \quad G_0)$

$G_2 = B_2$

$G_1 = B_2 \oplus B_1$

$G_0 = B_1 \oplus B_0$

Gray to Binary



BCD Addition :-

Step 1 :- Replace each BCD digit by its 4 bit binary number

Step 2 :- perform binary addition on each nibble

Step 3 :- If sum of each nibble is either > 1001 (6) or Carry is generated

Step 4 :- Correct the nibble by adding 0110.

Example 1 :-

$$\begin{array}{r} 43 \\ 739 \\ \hline 112 \end{array}$$

• Two corrections are needed. > 6

0100	0011	0010
0111	0011	1001
1011	0110	1011
0110		0110
10001	0111	0001

Hence the results are matching.

1 0 3 9
2 5 7 8
3 6 1 7

2 Connection required

0001	0000	0011	1001
0010	0101	0111	1000
0011	0101	1011	0001
		0110	0110
0011	0110	0001	0111

Hence the result is matching

Why we are adding 6 only ?

① BCD { 0 ... 9 }

4 bit possible { ... 16 } 6 remaining

② $2^4 = 16$

$\frac{10}{6}$

=

Logic Gates :

Positive logic

Vcc = 1

Gnd = 0

Negative logic

Vcc = 0

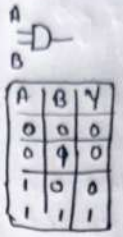
Gnd = 1

NOTE: By using duality we can convert +ve logic to negative logic

Logic gates :

- Basic: AND, OR, NOT
- Universal: NAND, NOR
- Special purpose: XOR, XNOR

AND



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

OR

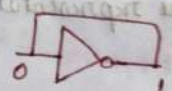
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT

A	Y
0	1
1	0

X = UnKnown → (When proper inputs are not defined)

Z = High impedence → (When input is not connected to output logic)

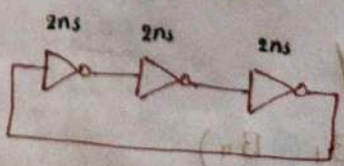


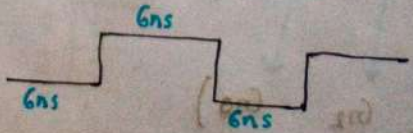
1) Tp (Time period) = 6ns

2) Frequency = $\frac{1}{Tp} = \frac{1}{6ns}$ = 166 MHz

3) Duty cycle = $\frac{Ton}{Tp} = \frac{3\ ns}{6ns} = 0.5$

50 %.



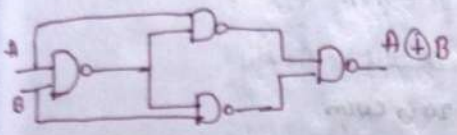


$T_p = 12ns$

$Freq = \frac{1}{12ns} = 83\ MHz$

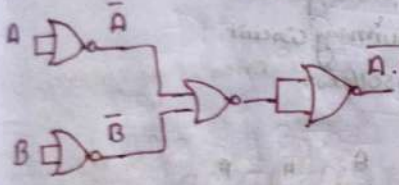
Duty cycle $= \frac{6ns}{12ns} = 0.5$ 50%

XOR Gate using NAND :-

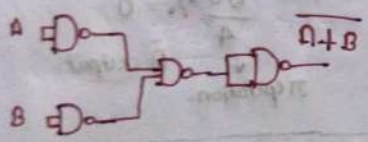


NAND Gate by NOR Gate :-

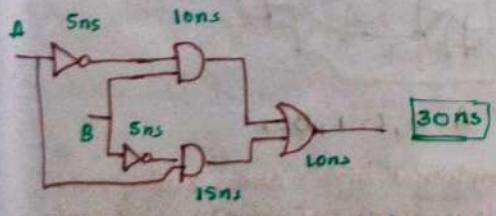
↓
- Symbol need to be used
- Equation Also.



NOR using NAND :-

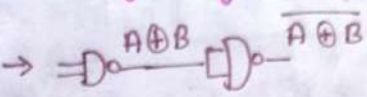


Find the total delay in the circuit

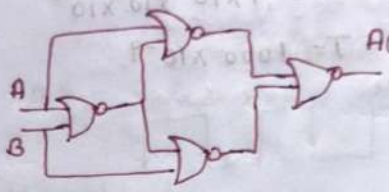


It will try to take the largest delay.

XNOR by using NAND :-



XNOR by using NOR



followed by another Not gate will give me XOR gate

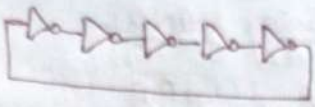
The Time period of Square Wave Generated by AMW = $T = 2nt_{pd}$

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

*** NOR Gate is used in low power devices

27/09/2025

Ring Oscillator with 5 inverter generate a waveform 1MHz. What is the delay? What is the delay of each inverter



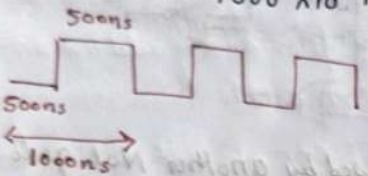
Given

$f = 1MHz$

$T = \frac{1}{f} = \frac{1}{1\times10^{6}} = 1\times10^{-6} = 1\mu s$

$1\times10^{-6}\times10^{3}\times10^{+3}$

$T = 1000\times10^{-9}$



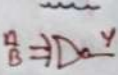
$Delay = \frac{T_{ON}}{T_P} = \frac{500\,ns}{5\,ns}$

Exor

$Y = A \oplus B$

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

EXNOR



$Y = A \odot B$

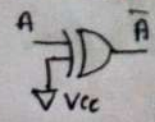
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

- Inequality
- Odd one detector
- Adder | Substractor
- Parity generator

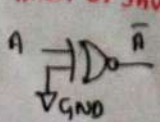
$Y = A\bar{B} + \bar{A}B$

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

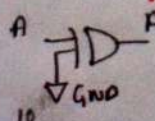
XOR as Inverter



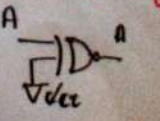
Xnor as Inverter



XOR as Buffer

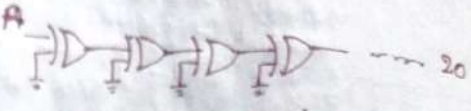


Xnor as Buffer



If even number of inverter are cascaded it will be

- Bistable multivibrator
- Basic memory element
- Used in FF/latch

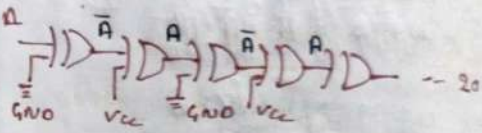


Output will be A only

by 20 is even

If odd number of inverter are cascaded It will be

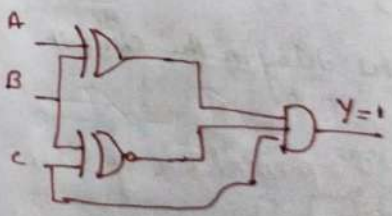
- Astable multivibrator
- Square wave generator
- Free running circuit
- Ring oscillator



$\frac{20}{4} \cdot /. = 0$

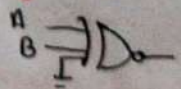
↓ repetation.

$\bar{A}$ output



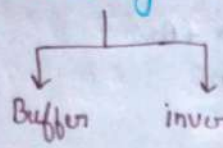
A = 0, C = 1, B = 1

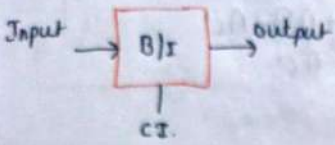
Implement higher order XOR & XNOR by lower Exor gate design XNOR gate using 2-input XNOR gate.



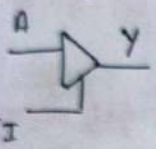
Hint whenever we are implementing higher by lower we can use Cascade technique

Tristate Logic



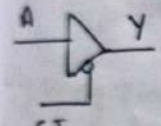


① Active high tristate Buffer.



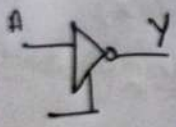
CI	A	Y	
0	X	Z	(inactive)
1	0	0	(active)
1	1	1	(active)

② Active low tristate Buffer.



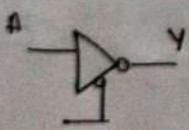
CI	A	Y	
0	0	0	active
0	1	1	active
1	0	Z	Inactive
1	1	Z	Inactive

③ Tristate Active high inverter



CI	A	Y
0	X	Z
1	0	1
1	1	0
1	0	1

④ Active low tristate inverter.



CI	A	Z
0	0	1
0	1	0
1	0	Z
1	1	Z

Logic Minimization :-

- Boolean Algebra invented by George Bool in 1854.
- Used to simplify logical solution.

Properties :

1) $A+A+A = A$
2) $A+0 = A$
3) $A+1 = 1$

$A \cdot A = A$	
$A \cdot 0 = 0$	
$A \cdot 1 = A$	
$A \cdot A' = 0$	

Dual ↗

Boolean Law :

① Dominance Law / Annulment law.

1) $A \cdot 0 = 0$
2) $A+1 = 1$.

② Identity law

1) $A \cdot 1 = A$
2) $A+0 = A$.

④ Idempotent law

1) $A \cdot A = A$
2) $A+A = A$

⑤ Complement law.

1) $A \cdot \bar{A} = 0$
2) $A+\bar{A} = 1$.

⑥ Commutative law.

1) $A \cdot B = B \cdot A$
2) $A+B = B+A$.

⑦ Double negation law.

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

⑧ Demorgan's law.

1) $\overline{A \cdot B} = \bar{A} + \bar{B}$
2) $\overline{A+B} = \bar{A} \cdot \bar{B}$

⑨ Distibutive law.

$A \cdot (B+C) = AB + AC$

$A+(B \cdot C) = (A+B)(A+C)$

Absorption law

$A + (AB) = A$

$A \cdot (A+B) = A$

Associative law

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

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

Consensus Theorem / Redundency Theorem

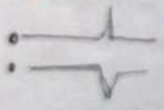
1. There must be 3 Variables
2. Each Variable must repeted twice
3. One Variable must be complemented

$Y = AB + A'C + BC$

$Y = AB + A'C$

NOTE:- Significance when ever we come across Hazard problem it can be solved by adding some redundent term to it.

- Hazard
  - Static Hazard
    - Static 0
    - Static 1
  - dynamic Hazard



- Cost will be increase but it will be getting accurate output

① $AB + A(B+C) + B(B+C)$

$= AB + AB + AC + B + BC$

$AB + AC + B + BC$

$AB + AC + B(1 + C)$

$AC + B$

② $A(A+B') + (B+A)(A+B)$

$= A$

③ $(A+C)(AD+AD) + AC + C'$

$AD + AD + ADC + ADC + AC'$

$AD + AC + C'$

④ $A(A+BC) + A(B+C')$

$A + ABC + AB + AC'$

$A + A + AC'$

A

⑤ $\overline{A \cdot \bar{A}\bar{B}} \cdot \overline{B \cdot \bar{A}\bar{B}}$

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

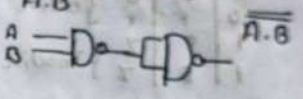
Implementation of logic Gate using NAND & NOR

SOP → Directly implement by NAND gate

POS → NOR Gate

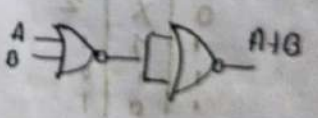
① AND Gate by NAND

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



② OR Gate by NOR

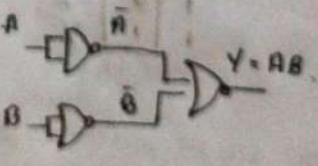
$\overline{\overline{A+B}}$



③ AND Gate by NOR

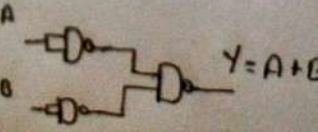
$= \overline{\overline{A \cdot B}}$

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

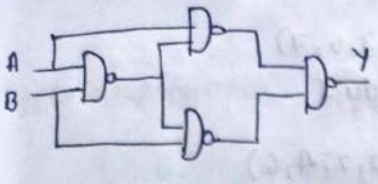


④ OR Gate by NAND

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

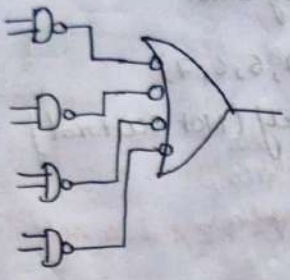


Implementation of XOR gate using Min number of NAND Gates



by NOR just replace by NOR

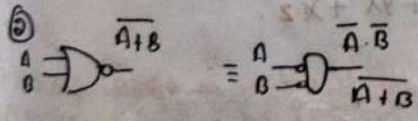
① $Y = AB + BC + CA$.

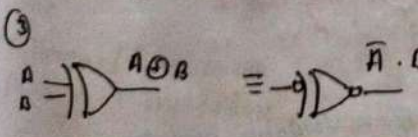


By using 2 input we can use this technique

try to implement by 2 input NAND Gate.

Alternate Symbols :-

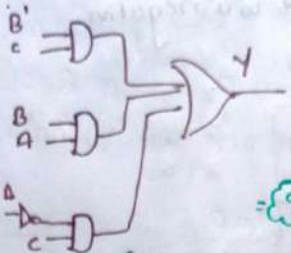




	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
XOR	4	5
XNOR	5	4
NAND	1	4
NOR	4	1

Finding circuit Efficiency after minimization :-

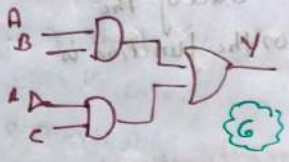
$Y = AB + Bc + \bar{A}c$



NOTE :- Cost is Equal to no of inputs. Each input has Cost of 1.

After simplification

$Y = AB + A'c$.



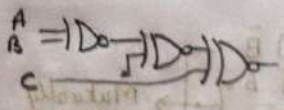
Circuit Efficiency :-

$$\frac{9-6}{9} = \frac{3}{9} = \frac{1}{3} \times 100$$

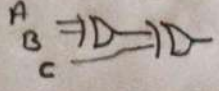
33% Efficient

***

Note :- For 3 input exor gate to impliment in 2 input exor gate we required 3 exor gate



For Exor gate no of inputs $= n-1$.



Duality Principle :

3/03/25

+ logic → Active high buffer
-v logic → Active low buffer

- Bubble indicate it is a negative logic

From	Dual
+	.
.	+
0	1
1	0

Eg :- $AB + BC + CA$

$(A+B)(B+C)(C+A)$

Eg $A.0 \rightarrow A+1$.

Self dual function : dual of the function is same as the function is called self dual

Testing of self dual function :

① Function must be "neutral"
↳ The function which has equal no of minterm and maxterm then it is called neutral.

- if n is no of variables 2^n minterm and maxterms are possible

② It must not contain mutually Exclusive terms

0 0	$\bar{A}\bar{B}$	
0 1	$\bar{A}B$	→ Mutually Exclusive pairs
1 0	$A\bar{B}$	
1 1	AB	

Check whether it is self dual or Not

(a) $f = \Sigma(1,2,3,4)$
= Non self

(b) $f = \Sigma(1,3,5,7)$
= self dual

(c) $f = \Sigma(0,2,4,6)$
= self dual

(d) $f = \Sigma(0,1,6,7)$
= Non self dual

(e) $f = \Sigma(1,2,5,6,7)$
= Non self (Not neutral)

Prove $xy + yz + zx$ is a self dual.

$xy + yz + zx$ is self dual

$(x+y)(y+z)(z+x)$

$[xy + xz + y + yz](z+x)$

$[y(x+1+z) + xz](z+x)$

$(y + xz)(z+x)$

$= yz + yx + xzz + xzx$

$= yz + yx + xz$.

⇒ dual of the function is same hence it is self dual

NOTE :- If n variable function total how many self dual function

$2^{2^{n-1}}$

Trick for demorgan's law.

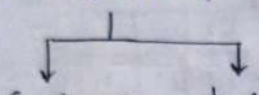
Replace
+ → ·
· → +
0 → 1
1 → 0

Example :- $(A+B)(B+C)(C+A)$

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

Complement Vs dual

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



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

dual :-

+ → ·
· → +

Simplify

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

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

K-MAP :-

Minterm

- SOP
- It is a product term produces output always one
- All variable/literals are present in True/False
- Total no of minterm $= 2^n$: n no of variable
- Minterm is rep by $m_{(sub)}$ → rep decimal value.

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

Maxterm

- Pos
- output = 0
-  A+B
- Represented by π

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

Rules of K-Map :-

1. Make a largest group.
2. Not in the diagonal form.
3. grouping will have 1 variable change.
4. We can have overlapping is allowed but no redundend group.

NOTE :- Kmap reduces the bool exp/eqn in SOM/POM

- We have grouping of 1, 2, 4, 8 ✓
  3, 5, 6, 7, 9 ✗

Grouping order :- 32 → 16 → 8 → 4 → 2

Rules :-

- group should only include cell containing ones in case of SOP.
- group should only include zeros in POS
- group should be only horizontal or Verticle.
- Group must contain 2^n cells
  $2^3, 2^2, 2, 2^0$.
- Overlapping of group is allowed.

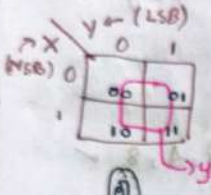
Dont care :-

helps to reduce the ckt but we try to use less.

2-Variable - K-MAP

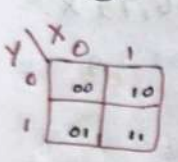
Total number of min = 4
max = 4

X	Y	Min	Max
0	0	$\bar{x}\bar{y}$	$x+y$
0	1	$\bar{x}y$	$x+\bar{y}$
1	0	$x\bar{y}$	$\bar{x}+y$
1	1	xy	$\bar{x}+\bar{y}$



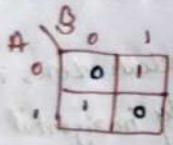
(or)

$Y \Sigma m(0,1)$
$Y = \Pi(2,3)$



$\bar{X}$ = SOP
X = POS

Eg:-

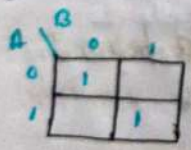


$Y = \bar{A}B + A\bar{B}$ (SOP

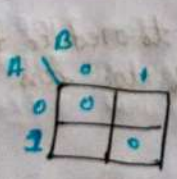
$Y = (\bar{A}+B)(A+\bar{B})$. POS)

↳ Focus on 0 term only

Q



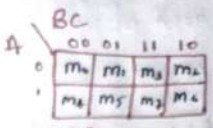
$Y = \bar{A}\bar{B} + AB$

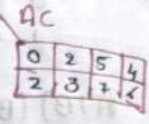


$Y = (A+B)(\bar{A}+\bar{B})$

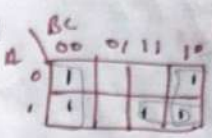
3- Variable - K Map

$2^3 = 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$





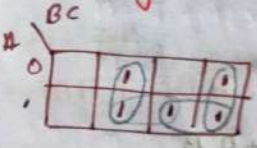
Eg



$Y = \bar{C} + AB$

Octave → 8 cells
quad → 4 cells
pair → 2 cells

Reduce $Y = \Sigma m(1,2,5,6,7)$ in SOP from using K-map.



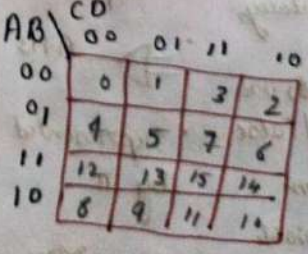
$Y = \bar{B}C + B\bar{C} + Ac$

(or)

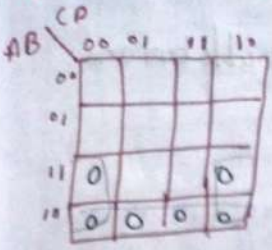
$Y = \bar{B}C + B\bar{C} + AB$

4- Variable K. map

$2^4 = 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

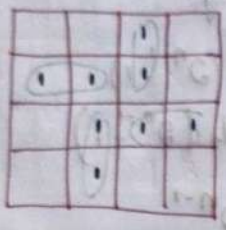


Reduced $Y(A,B,C,D) = \Sigma m(0,1,2,3,4,5,6,7,13,15)$ in SOP form using K-map



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

Note:-



In this above their is no need of quad beg it creates an Extra problem

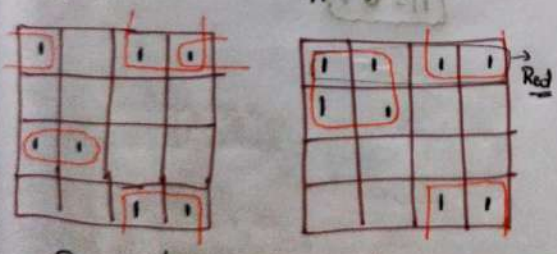
4/03/2025

5-Variable K-MAP

- Two ways:-

(a) Two K map separately for MSB and MSB(1).

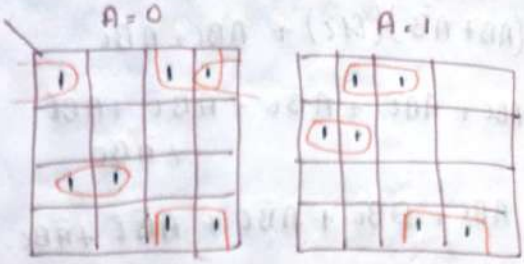
(a)



Try to form a group by Combining them

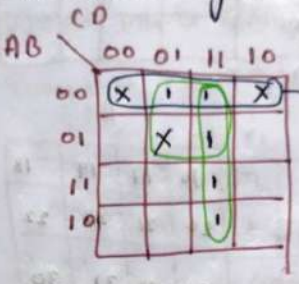
- Reduce by using K-map.

$\Sigma(0,2,3,10,11,12,13,17,19,20,21,26,27)$



$B\bar{C}D + \bar{C}D + \bar{B}\bar{C}E + BC\bar{D} + \bar{B}CD$

NOTE:- Try to form a group with less number of redundent term.



Canonical SOP :- Each literal a contain all Variable in it.

in SOP $Y = AB + BC + CA$

Step 1 :- Observe the missing literals and add $C+\bar{C}$ and apply distibution

Step 2 :- do for all the product term

Step 3 :- Remove all the replicated term Keep only one copy

Step 4 :- Represent in Sum of minterm

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

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

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

$Y = \Sigma m(3,4,5,6,7)$

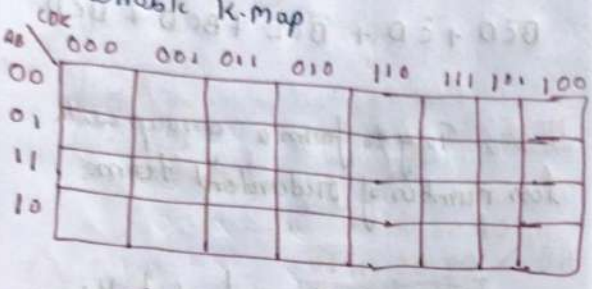
$Y = A(B+\bar{B})(C+\bar{C}) + BC(A+\bar{A})$

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

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

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

5-Variable K-map



$\bar{A}$

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

A

16	17	19	18
20	21	23	22
28	29	31	30
24	25	27	26

While grouping Consider $\bar{A}$ and A also and try to make Common grouping.

NOTE:- Standard derived results :-

- With n variables possible min & maximum terms = 2^n
- With n variables Maximum possible logical expression = 2^{2^n}
- With n variable Maximum possible self dual. $2^{2^{(n-1)}}$

Minimization by NAND & implementation

① Try to Keep double bar and then processed.

Finding the number of bits requires to implement

-64

- One bit for sign

2's $-(2^{n-1})$ to $(2^{n-1}-1)$.

1's $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

$-64 = -2^{n-1}$

$2^6 = 2^{n-1}$

$6 = n-1$

$n = 7 + 1$ (sign bit)

$n = 8$

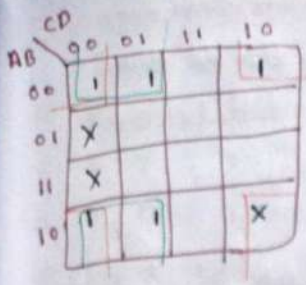
1's Complement :-

$2^6 = 2^{n-1}-1$

if $n = 7$ it is not Enough

$n = 8$ 9

5/3/2025

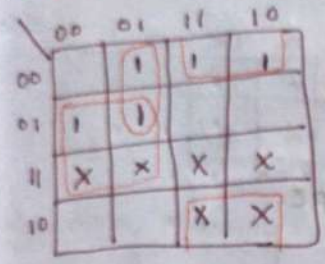


$Q_1 = (0,1,8,9)$

$Q_2 = (0,2,8,10)$

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

$Y = \Sigma_m(1,2,3,4,5) + d(10,11,12,13,14,15)$



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

Variable Entered Mapping (VEM)

1) Map the largest no of variable to lower variables

$\Sigma[0,2,3,5,6,7,8,10,11,13,14,15)$

Step-1

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

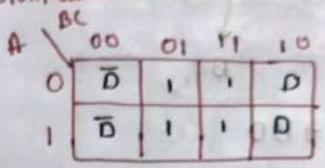
Hint: Take LSB bit as reference of all the conditions.

Step-2

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

In this process we are reducing the 4 Variable into 3 Variable.

From above Truth Table

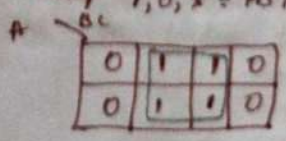


Steps involved in Solving this :-

1. Treat all the Variables as zero and get SOP (0)
2. Mark one Variable of all as 1' with previous 1's, or don't care & get SOP
3. Multiply SOP with Variable
4. Repeat step-2 & 3 until all Variables are covered
5. SOP of VEM is sum of all VEM obtained earlier.

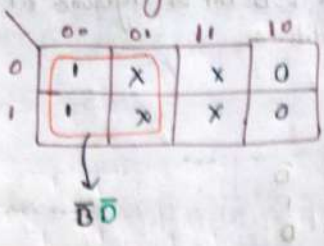
① Make D & $\bar{D} = 0$

Keep 1, 0, X = As it is

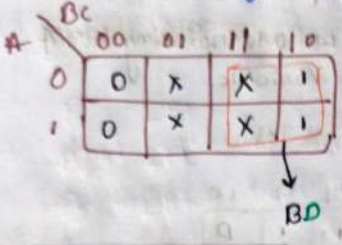


C

⑤ · D=0 $\bar{D}$=1
· Replace 1 with x
· 0,x unchanged.



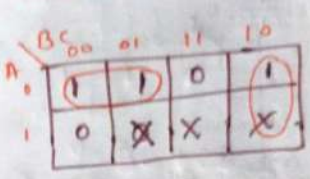
③ D=1 $\bar{D}$=0
Replace 1 with x
0/x → unchanged



$Y = C + B\bar{D} + BD$

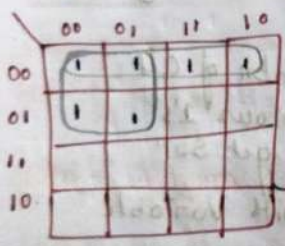
A B C	f
0 0 0	1
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	0
1 0 1	X
1 1 0	X
1 1 1	X

① Since there is no Variable it is similar to normal K-Map



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

$\Sigma(0,1,2,3,4,5) + d(10,11,12,13,14,15)$



A B C D	Y	
0 0 0 0	1	} 1
0 0 0 0	1	
0 0 0 1	1	
0 0 1 0	1	} 1
0 0 1 1	1	
0 1 0 0	1	} 1
0 1 0 1	1	
0 1 1 0	0	} 0
0 1 1 1	0	
1 0 0 0	0	} 0
1 0 0 1	0	
1 0 1 0	X	} X
1 0 1 1	X	
1 1 0 0	X	} X
1 1 0 1	X	
1 1 1 0	X	

6/3/2025

LSB as mapping Variable :-

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

Now during Mapping Variable as A :-

Trick :- Maping Variable on LSB

B	C	A	f	
0	0	0	1	1
0	0	1	1	
0	1	0	0	A
0	1	1	1	
1	0	0	0	0
1	0	1	0	
1	1	0	1	1
1	1	1	1	

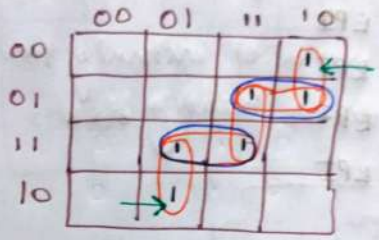
Given the minimized Expression and write the output. By Comparing the sequence by dividing the groups we can assign by seeing variables

Implicants :- All terms which are mentioned in K-map are also know as implicants

Prime implicants :- All possible pairs in a K-map are know as prime implicants

Essential prime implicants :- All possible pairs in which at least one min term, which can't be pair with any other way considered as EPI.

Redundent prime implicant :- The prime implicant whose each '1' is covered by atleast one important / Essential prime implicants



Implicants = 6 {2, 6, 7, 9, 13, 15}

Prime implicants = 5

Essential implicants = 2

Redundent prime implicants = 2

Prime implicants =

$\bar{A}CD + \bar{A}BC + BCD + ABD, AC'D$

Essential prime =

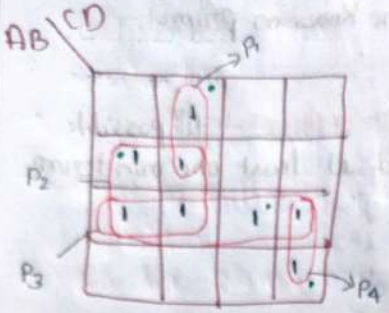
$A'CD', AC'D$

Redundent prime =

$A'BC, ABD$

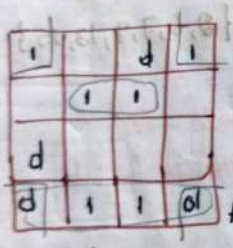
Selective prime implicant :- selecting one more extra term

① Which of the following is essential Prime implicant $Y(ABCD) = [1,4,5,10,12,13,14,15]$



P_1 EPI (Atleast one minterm which is not grouped with any other)

P_1 = EPI
P_2 = EPI
P_3 = EPI
P_4 = EPI



EPI = 3

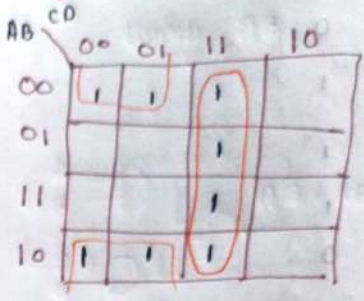
In all the group there will be many unpaired terms

NOTE:- Once the don't care is used you can use to minimize them.

Tabular Method

- used to minimize variable more than 4.

Simplify $Y(A,B,C,D) = \Sigma(0,1,3,7,8,9,11,15)$ using Tabular Method.



$Y = \bar{B}\bar{C} + CD$.

Table-1

Minterms	Variables			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
3	0	0	1	1
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
11	1	0	1	1
15	1	1	1	1

Table-2 :- Based on number of ones.

No of 1's	Minterm	Variables			
		A	B	C	D
0	0	0	0	0	0
1	1	0	0	0	1
	8	1	0	0	0
2	3	0	0	1	1
	9	1	0	0	1
3	7	0	1	1	1
	11	1	0	1	1
4	15	1	1	1	1

Table 3 :- Make a group by one bit change and change by –

Group	Match pair	Variable A	B	C	D
Group 0	(0,1)	0	0	0	–
	(0,8)	–	0	0	0
Group 1	(1,3)	0	0	–	1
	(1,9)	–	0	0	1
	(8,9)	1	0	0	–
Group 2	(3,7)	0	–	1	1
	(3,11)	–	0	1	1
	(9,11)	1	0	–	1
Group 3	(7,15)	–	1	1	1
	(11,15)	1	–	1	1

Note :- After checking put check mark

Table 4 :- Combining 4 Maching pair.

Group	Match pair	Variable A	B	C	D
Group 0	(0,1,8,9)	–	0	0	–
	(0,8,1,9)	–	0	0	–
Group 1	(1,3,9,11)	–	0	–	1
	(1,9,3,11)	–	0	–	1
Group 2	(3,7,11,15)	–	–	1	1
	(3,11,7,15)	–	–	1	1

Group	Matched Variable	Variable A	B	C	D
Group 0	(0,1,8,9)	–	0	0	–
Group 1	(1,3,9,11)	–	0	–	1
Group 2	(3,7,11,15)	–	–	1	1

Prime implicant Table :-

Prime implicant	Minterms involved	0	1	3	7	8	9	11	15
B'C'	(0,1,8,9)	X	X			X	X		
B'D	(1,3,9,11)		X	X			X	X	
CD	(3,7,11,15)			X	X			X	X
		✓				✓			✓

Choose colum with simple cross, write associated minterm

$Y = B'C' + CD$

EPI Verification :-

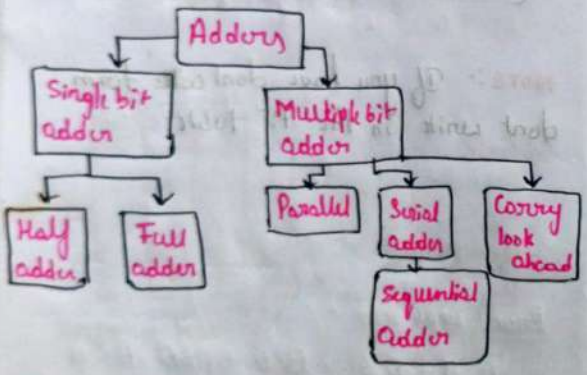
EPI	Mint	0	1	3	7	8	9	11	15
$\overline{B}\overline{C}$	(0,1,8,9)	✓	✓			✓	✓		
CD	(3,7,11,15)			✓	✓			✓	✓

NOTE :- If you have dont care term dont write in the PI table

Combinational circuits:-

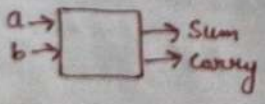
→ output depends only on present input
→ Add, Mux, demux, Encoder, decoder Parity encoder, Mux
→ Speed is fast
→ Its in design is easy
→ Used for arithmetic & Boolean operation
→ No clock, No memory, No feedback

→ Output will be at instant of time
→ Basic gates are building block



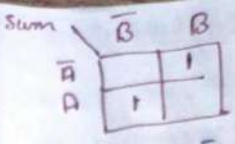
• Parallel adder = Carry ripple adder

Half adder:-

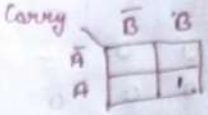


A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum = $\Sigma m(1,2)$
Carry = $\Sigma m(3)$

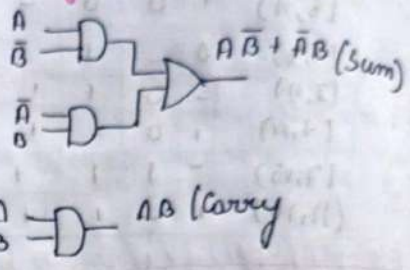


Sum = $A\bar{B} + \bar{A}B = A \oplus B$

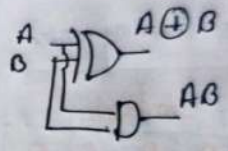


Carry = AB

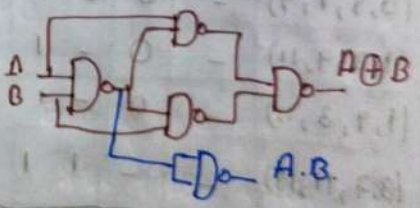
Basic gate:-



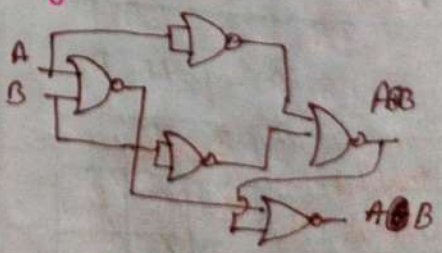
EXOR:-



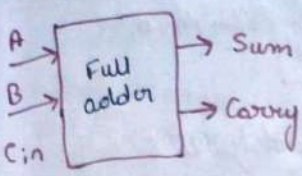
By NAND Gate:-



By NOR Gate:-



Full adder :



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

$Sum = A \oplus B \oplus Cin \quad \Sigma(1,2,4,7)$

$Carry = AB + BC + CA \quad \Sigma(3,5,6,7)$

Sum

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

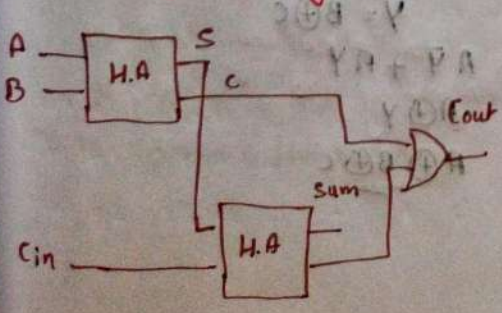
$Sum = A \oplus B \oplus C$

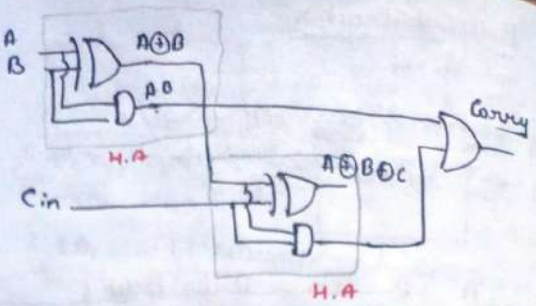
Carry

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

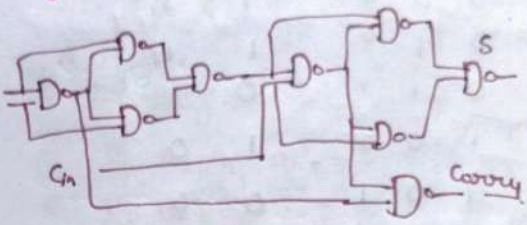
$Carry = AC + AB + BC$

Full adder by two Half adder

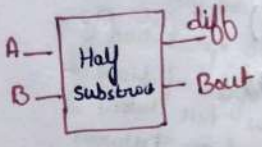




By Nand Gate :



Half Substractor :



A	B	D	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

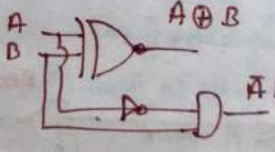
$D\Sigma(1,2)$

$Bout(1)$

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

$= A \oplus B$

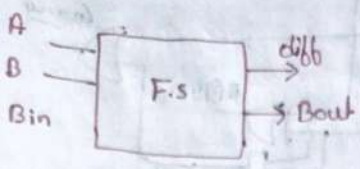
$Bout = \bar{A}B$



Steps to design Combinational circuit

1. Analyse the statement and draw a block diagram.
2. Tabulate the truth table
3. Get reduced expression from K-Map
4. Design internal ckt by specified logic gates.

Full Substractor :-



A	B	Bin	Diff	Bout
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

$Diff = \Sigma m(1,2,4,7)$

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

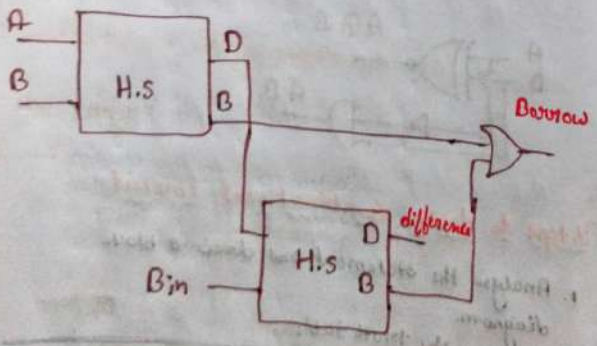
$diff = A \oplus B \oplus C$

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

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

$\bar{A}B + Bc + \bar{A}c$

Full substractor by using Half Substractor :-



Sum = $A \otimes B \otimes c$

Carry = $AB + Bc + cA$

Diff = $A \otimes B \otimes C$

Borrow = $\bar{A}B + B.C + \bar{A}c$

Implement of A+B by half adder

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

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

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

$= (A \odot B)AB + (A \oplus B)\overline{AB}$

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

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

$\bar{A}\bar{B}A^{\circ}B + AB$ — ①

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

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

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

$= \bar{A}B$ — ②

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

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

$= A\bar{B}$ — ③

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

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

$B + A\bar{B}$

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

$A+B$

Bout Equation :-

$\bar{A}\bar{B}c + \bar{A}B\bar{c} + ABc + A\bar{B}\bar{c}$

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

$A(B \odot c) + \bar{A}(B \oplus c)$

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

$Y = B \oplus c$

$A\bar{Y} + \bar{A}Y$

$A \oplus Y$

$A \oplus B \oplus c$

10/03/2085

Note: Why we are implementing the full adder by using 2 half adder.

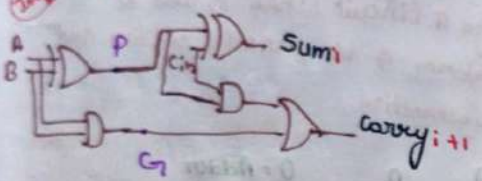
→ Based on requirement we need to implement them.

Distributive Law:

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

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

Full Substraction circuit can be used for carry full adder circuit to implement carry lookahed adder. for carry propogation and carry generation



P = Propogation = $A \oplus B$

G = Generation = $A.B$.

—:o:—

NOTE: Now with full and holf adder(sum) we can add 2 bit of 3 not more than those this can be implemented by

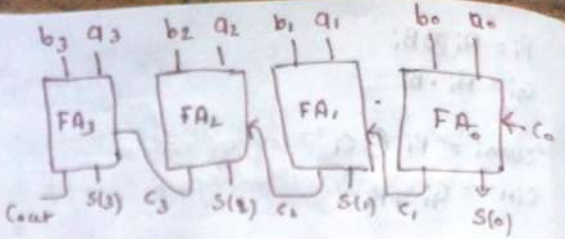
① Parallel adder (Ripple carry)

② Carry look ahed addon.

→ When we compare both carry look adder has less propogation delay

	FA	FA		
	C_3	C_2	C_1	C_0 — FA
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

→ So it required 4 full adders



→ FA_1 will be activated when C_1 will be propgated.

→ If every F.A has t_1 delay total delay will be $4t_1$.

No input carry:

A	A_3	A_2	A_1	A_0
B	B_3	B_2	B_1	B_0
	S_3	S_2	S_1	S_0
	+	+	+	
	C_2	C_1	C_0	

half adder

F.A

Full adder

→ It has large propogation delay. it need to wait for before carry mean it need to waint not working in full parallel mode.

Carry Look Ahead adder.

→ It include carry generation & carry propogation.

Involed 3 steps:-

① Obtain carry generation(G_i) & carry propogation (P_i)

② Sum at every stage

$Sum = P_i \oplus C_i$

$S_0 = P_0 \oplus C_0$

$S_1 = P_1 \oplus C_1$

$S_2 = P_2 \oplus C_2$

$S_3 = P_3 \oplus C_3$

③ Next stage carry

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

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0)$

$C_3 = G_2 + P_2 C_2$.

$C_4 = G_3 + P_3 C_3$.

$P_i = A_i \oplus B_i$

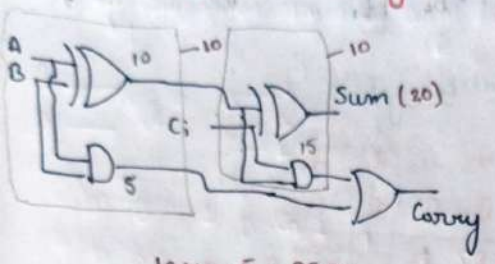
$G_i = A_i \cdot B_i$

$Sum_i = P_i \oplus C_i$

$C_{i+1} = G_i + P_i C_i$ } ④

If we consider a Full adder circuit with half adder. Each XOR = 10ns, AND = 5ns, OR = 5ns } delay

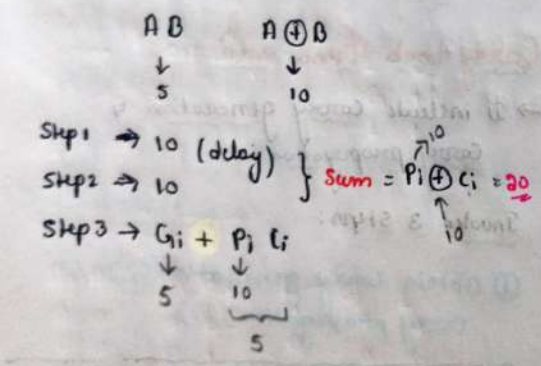
For 4 bit adder total latency.



10+10+5 = 25ns

For 4 Full adder = 4 × 25 = 100ns

NOTE: By these 4 Equation we can reduce the Carry propogation delay.



→ Carry Ripple Adder :-

- Parall adder (cascaded), Propogation delay is added to each circuitary

→ CLA :-

- Additional circuitary for C_g & C_p but latency is reduced.

A n-bit parallel adder requires

- n Full adder ⊕ (n-1) Full adder and Half adder.

- by half adder we required 2n half adder and n-gate.

→ In n-bit parallel adder, the minimum delay to produce the final result is Equal to 2 * n * tpd (where tpd is the propogation delay of Each)

→ To speed up the procss look Ahed Carry Adder is used, which is based on Carry generation & propgation Fully parallel circuit. Cost extra circuit

Design a Circuit which is able to perform 4-bit parallel addition and substraction

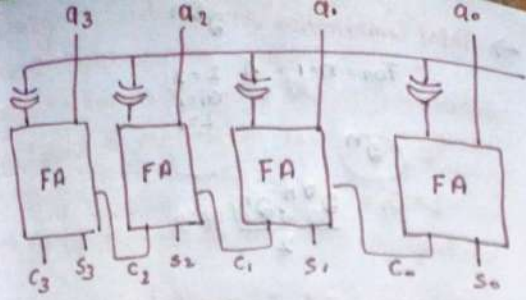
A + B	A − B	0 = Adder, 1 = substractor
		Ctrl. Logic

&

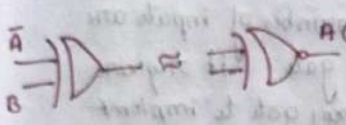
A + B	A − B (B's Complement)
no change	

A	B	Y	
0	0	0	Buffer
0	1	1	Buffer
1	0	1	Compliment
1	1	0	Compliment

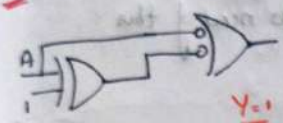
NOTE: desicion making circuit will be designed by XOR & XNOR Gates.

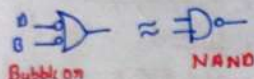


NOTE: In carryout Equation of full Substractor one need to be Complimented.



Q





How many 4-bit parallel adder would be required to add two binary number each representing decimal number up through 280?

2	280	0
2	140	0
2	70	0
2	35	0
2	17	1
2	8	0
2	4	0
2	2	0
	1	

1001 0000 0

3 required

For 1024.

→ Not 10 bcy

$= 2^{10} - 1$

$= 1024 - 1$

$= 1023$

10+1 = 11 bits

3-4bit adder required

Question: $(36 \times 70)_{10}$ is 10's Complement of $(YzY0)_{10}$ then find x, y, z

9 9 9 9

Y Z Y z 0

36 x 70

9 - z = 6

9 - 6 = z

z = 3

9 - Y = 3

9 - 3 = Y

Y = 6

10 - z = 7

10 - 7 = z

z = 3

9 - Y = x

9 - 6 = x

3 = x

Q. Design a circuit with 3 inputs to Count 1's present in input Combination using full adder

Q 1's compliment addition and Substraction using parallel adder. & 2's Compliment also.

11/03/2025

Comparator Circuit:

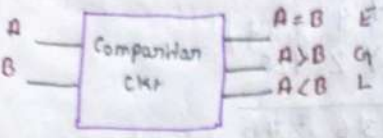
→ Total Combination = 2^{2n}

For $n=1$ = 4 (E=2, G=1, L=1)

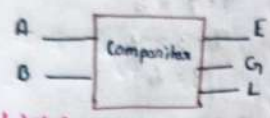
$E = 2^n$

$L = G = \frac{2^{2n} - 2^n}{2}$

D performs 3 different operation
→ greater, lessthan, Equal



→ n bit Comparitor required only 1 output



1-bit Comp:

A	B	E	G	L
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$E = \Sigma(0,3)$
$G = \Sigma(2)$
$L = \Sigma(1)$

$E = \bar{A}\bar{B} + AB = A \odot B$
$G = A\bar{B}$
$L = \bar{A}B.$

Note: 1 bit Comparitor

E	0
G	1
L	1

2-bit Comparitor:

A_1	A_0	B_1	B_0	E	G	L
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

$\Sigma E = (0,5,10,15)$
$\Sigma L = (1,2,6,3,7,11)$
$\Sigma G = (4,8,9,12,13,14).$

Note: Exnor gate is used for Comparitor is Equality.

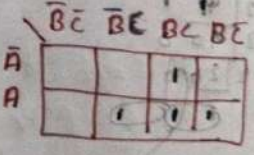
Tip: If the number of inputs are odd for exor gate it require one more exor gate to implement in cascate or not gate.

- If it is even no need of this extra gate.

Non Conventional Combinational logic circuit.

Design the Combinational ckt with 3 input to detect the majority function

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	0	1



$Y = AB + BC + CA$

$f = (3,6,5,7)$

Output is Exactly Equal to fulladder

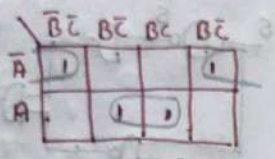
Design a Combinational ckt which has 3 input and one output and is Equal1 when no of ones is entire combination is Equal of more than two the output is one.

→ It is similar to majority output.

Q Pallindrome Sequence :-

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

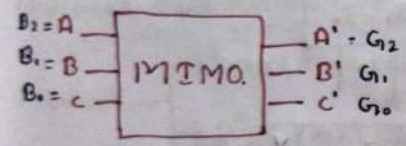
$Y = \Sigma(0, 2, 5, 7)$



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

$Y = A \odot C$

Q Design a circuit which is able to Convert 3 bit binary to gray number.

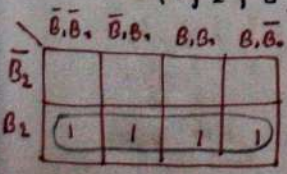


B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	0	1	0	1
1	1	1	1	0	0

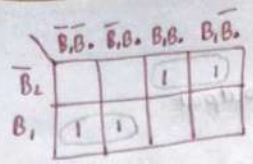
$G_2 = (4, 5, 6, 7)$

$G_1 = (2, 3, 4, 5)$

$G_0 = (1, 2, 5, 6)$



$G_2 = B_2$



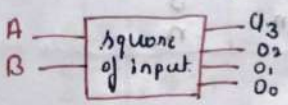
$G_2 = \bar{B}_2 B_1 + B_2 \bar{B}_1$

$G_2 = B_2 \oplus B_1$

$G_1 = B \oplus B_0$

NOTE : Gray Code low power device is used.

Design a Combinational circuit which Produces square of 2 bit number



A	B	O_3	O_2	O_1	O_0
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

Multiplier :-

①

	A_1	A_0
	B_1	B_0
	A_1B_0	A_0B_0
A_1B_1	B_1A_0	×

Resource requirement :-

→ 4 (2 input AND Gate)

→ 2 Half adder

②

		A_2	A_1	A_0
		B_2	B_1	B_0
		A_2B_0	A_1B_0	B_0A_0
	A_2B_1	A_1B_2	A_0B_1	×
A_2B_2	A_1B_2	A_0B_2	×	×
F	FH H	F.H H	H	0

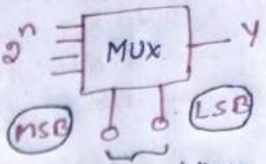
→ 9 (2 input AND Gate)

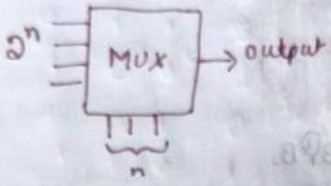
3 (Half adder)

3 (Full adder)

MULTIPLEXER :-

→ Multiple inputs single output
→ Parallel in Serial out
→ Arithmatic operator
→ Universal ckt
→ Known as MUX





→ $S = \log_2 N$. (S = select line)
(N = no of inputs)

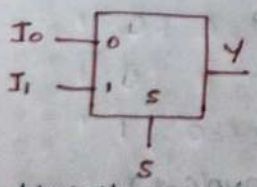
→

	inputs	select line
2:1 Mux	2	1
4:1 Mux	4	2
8:1 Mux	8	3
2^n Mux	2^n	n

Non linear design :-

	input	select line
3:1 Mux	3	2
5:1 Mux	5	3
10:1 Mux	10	4
100	100	7

① 2:1 MUX



$Y = \bar{S} I_0 + S I_1$.

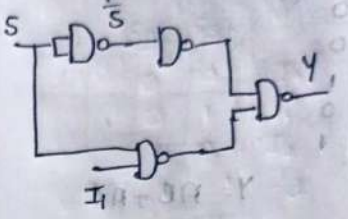
Proof of logically

I_0	I_1	S	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

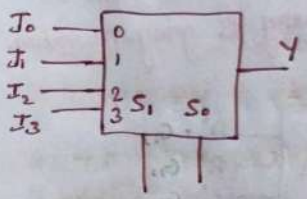
2:1 MUX using NAND

$$Y = \overline{\bar{S} I_0 + S I_1}$$

$$Y = \overline{\bar{S} I_0} \cdot \overline{S I_1}$$



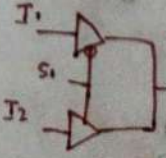
2:1 MUX



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

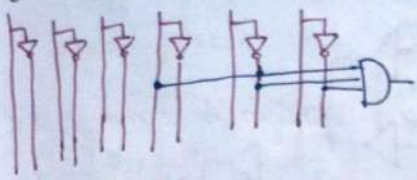
$Y = I_0 \bar{S}_0 \bar{S}_1 + I_1 \bar{S}_0 S_1 + I_2 S_0 \bar{S}_0 + I_3 S_0 S_1$

design 2:1 MUX using tristate buffer



Reconfigurable Architecture :-

I_3 I_2 I_1 I_0 S_1 S_0



	AND	OR
PLA	ON	ON
PAL	ON	X
ROM	X	P

Types of questions they may ask.

1. Impliment any boolean function circuit using Mux.
2. Impliment higher order mux using lower order mux
3. Implimentation of Analysis of Network

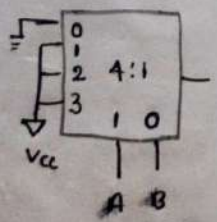
1. Implimentation of boolean Expression by Mux

Find which mux have to use

nVariable →
- → 2n-1 x 1 mux (all n inputs are select line)
- → 2^{n-1} x 1 (n-1 input → selectline, 1 input Data line)
- → 2 x 1 (n-1 → Data line, 1 input select line.

a) $y = \Sigma(1,2,3)$

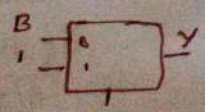
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



Or

A	B	Y	
0	0	0	$i_0 = B$
0	1	1	
1	0	1	$i_1 = 1$
1	1	1	

↓ select line, ↓ data line

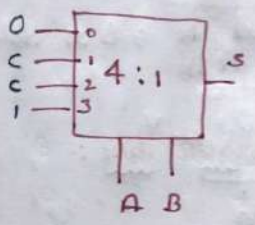


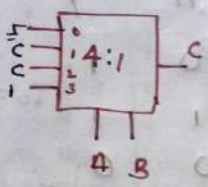
For 2 x 1 Mux = $\bar{S} I_0 + S I_1$

Implimentation of 4 x 1 Mux.

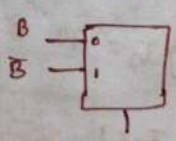
A	B	C	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S D





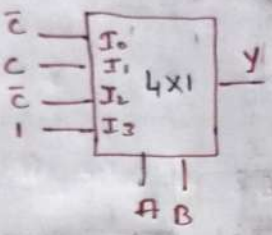
XOR Gate by 2:1 Mux



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

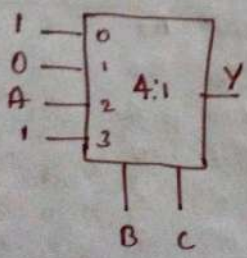
Implement $Y = \Sigma(0,3,4,6,7)$ using 4:1 Mux AB(selectline)

A	B	C	Y	
0	0	0	1	$I_0\,\bar{C}$
0	0	1	0	
0	1	0	0	$I_1\,C$
0	1	1	1	
1	0	0	1	$I_2\,\bar{C}$
1	0	1	0	
1	1	0	1	$I_3\,1$
1	1	1	1	

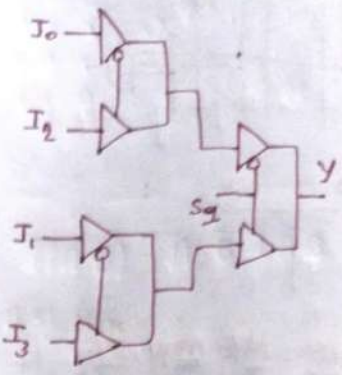


Now BC as select line :-

B	C	A	Y	
0	0	0	1	$I_0 = 1$
0	0	1	1	
0	1	0	0	$I_1 = 0$
0	1	1	0	
1	0	0	0	$I_2 = A$
1	0	1	1	
1	1	0	1	$I_3 = 1$
1	1	1	1	

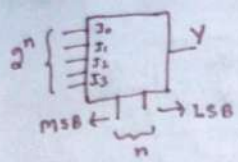


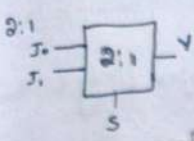
Design 4x1 Mux using tristate buffer.



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

13/03/2025

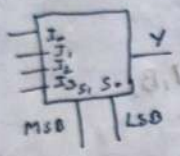




S	Y
0	I_0
1	I_1

$Y = I_0\bar{S}_0 + I_1 S_0$

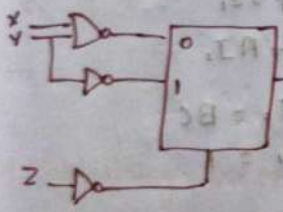
→ By using Nand gate we requires 4 Nand gates.



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_2

$Y = I_0\bar{S}_0\bar{S}_1 + I_1 S_0\bar{S}_1 + I_2\bar{S}_0 S_1 + I_3 S_0 S_1$

Eg:



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

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

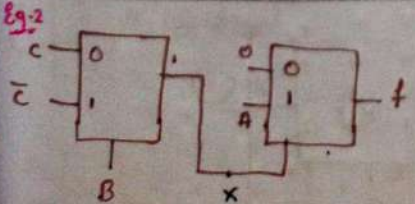
$= \bar{z}\bar{y} + \bar{x}\bar{y}z$

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

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

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

Eg:2

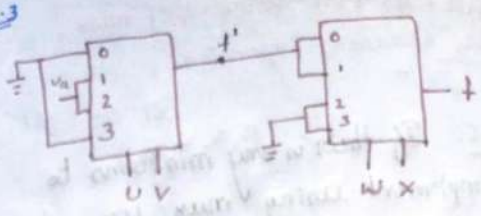


$X = \bar{B}C + B\bar{C}$

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

$f = (B \oplus C)A$

Eg:3



$f' = \bar{U}V + U\bar{V}$

$f' = U \oplus V$

$= \bar{W}\bar{X}(U \otimes V) + \bar{W}X(U \oplus V)$

$= U \oplus V(\bar{W}\bar{X} + \bar{W}X)$

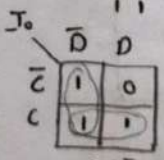
$= U \oplus V(\bar{W})$

$= (U \oplus V)\bar{W}$

Q. $Y = \Sigma(0,2,3,6,7,14,15)$ using 4:1 MUX.

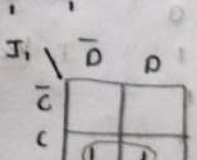
A	B	C	D	Y	
0	0	0	0	1	I_0
0	0	0	1	0	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	I_1
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	I_2
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	I_3
1	1	0	1	0	
1	1	1	0	1	
1	1	1	1	1	

I_0



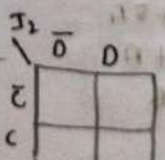
$C + \bar{D}$

I_1



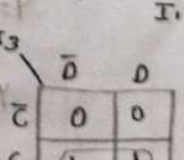
$I_1 = C$

I_2

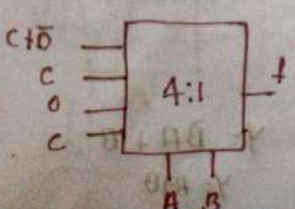


0

I_3



$I_3 = C$

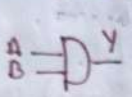


$Y = AB + BC + CA$ using 4×1 mux.
use Shannons Expansion.

NOTE: If their is any minterms to implement using mux use only tabular method.

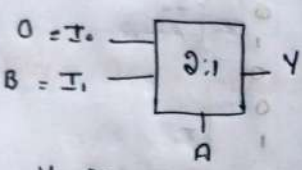
→ Shannon method only for 2:1 Mux only.

2-input AND Gate using 2:1 MUX



A	B	Y	
0	0	0	$I_0 = 0$
0	1	0	
1	0	0	$I_1 = B$
1	1	1	

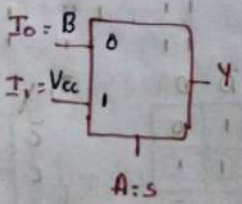
Here A = Select line & B as data line



$Y = \bar{S} I_0 + S I_1$
$= \bar{A} 0 + AB$
$Y = AB$

2:1 Mux using OR Gate.

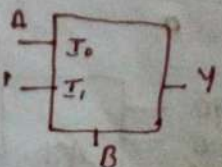
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



$Y = \bar{S} I_0 + S I_1$
$\bar{A} B + A 1$
$Y = A + B$

B → Select line
A → Data line

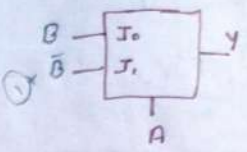
B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1



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

2 input X-OR Gate using 2:1 Mux

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



Using Shanons Expansion (only 2:1)

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

A Select line :-

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

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

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

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

By using $\bar{S} I_0 + S I_1$.

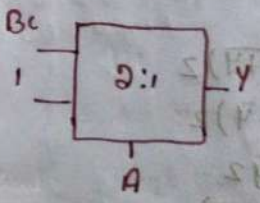
$Y = A + BC$ using 2:1 Mux

$Y = \bar{S} I_0 + S I_1$

$\Rightarrow \bar{A} I_0 + A I_1$

$f(0,B,C) = I_0 = BC$

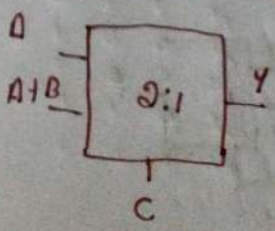
$f(1,B,C) = I_1 = 1$



C as select line :-

$f(A,B,0) = I_0 = A$

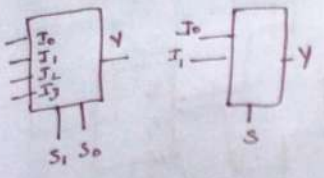
$f(A,B,1) = A + B$



Implementation of higher order Mux using lower order Mux.

For.
4:1 → 3
8:1 → 7
16:1 → 15
⋮
2^n:1 → 2^n-1 Mux.

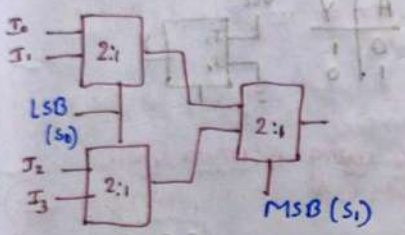
Q Impliment 4:1 Mux using 2:1 Mux



$\frac{4}{2}^{2}$ $\frac{2}{2}^{1}$

Level-1 Level-2

Note: Selection line will be same for Each level.



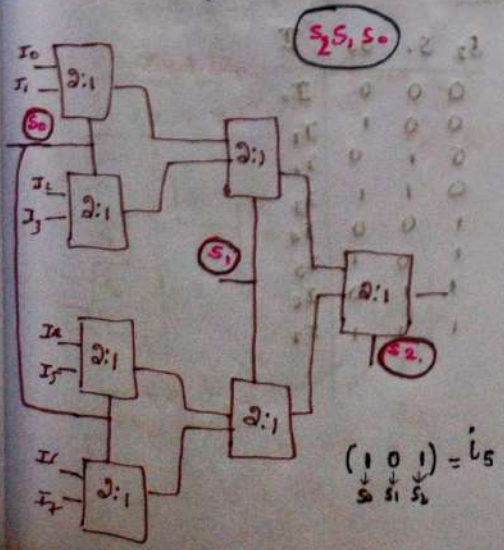
S1	S0	
1	1	= I3
1	0	= I2

8:1 Mux using 2:1 Mux

$\frac{8}{2}^{4}$ $\frac{4}{2}^{2}$ $\frac{2}{2}^{1}$

Level-1 Level-2 Level-1

7 Mux are req (2:1)

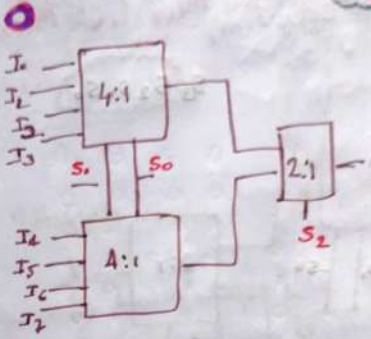


Q 8:1 Mux using 4:1 Mux & 2:1 Mux

$\frac{8}{4} = 2$ (4:1 Mux)

$\frac{2}{2} = 1$ (2:1 Mux)

$S_2 S_1 S_0$



16:1 Mux using 4:1 Mux

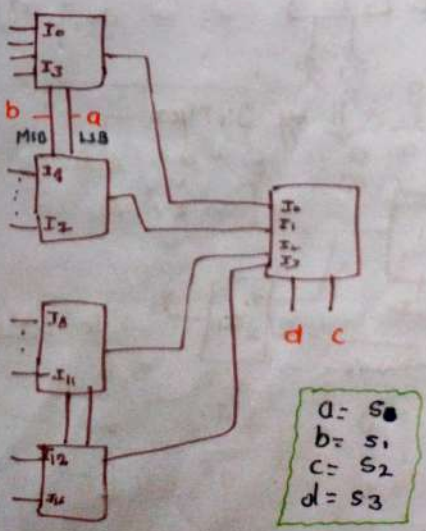
$\frac{16}{4}^{4}$ $\frac{4}{4}^{1}$

Selection lines to req = 16.

$S_3 S_2 S_1 S_0$

Level-1 (4×1) (4:1)

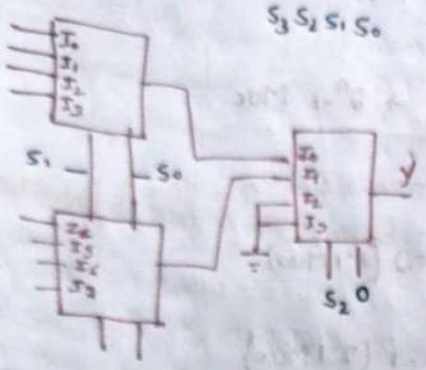
Level 0 1 (4×1)



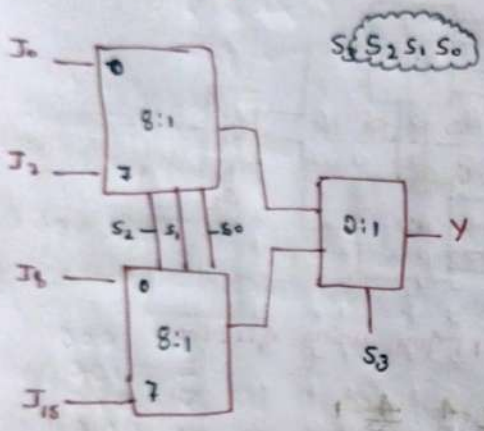
8:1 MUX using 4:1 MUX

8
4 2

$S_3 S_2 S_1 S_0$

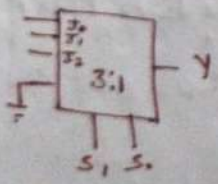


16:1 Mux using 8:1 Mux & 2:1 Mux

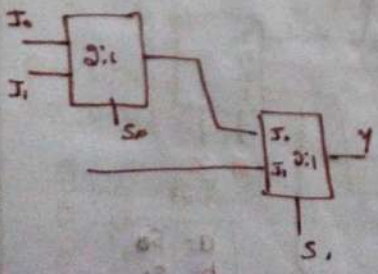


Non Conventional Multiplexer :

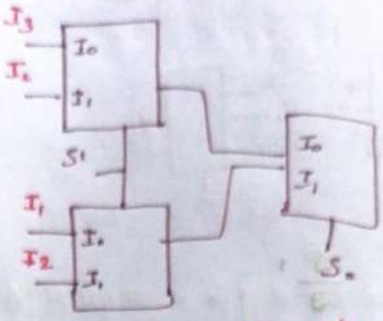
3:1 Mux using 2:1 Mux



⇓ by 2:1 Mux



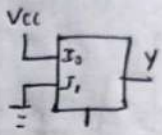
What if I change the Order of the Select line.



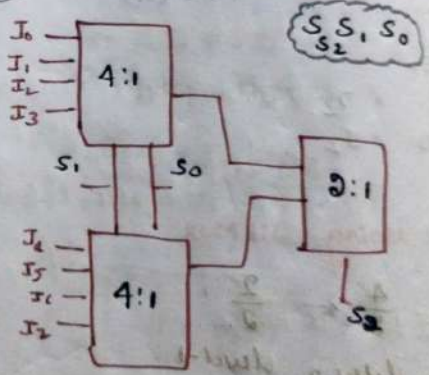
due to change in the Select line.

Inverter using 2:1 Mux

A	Y
0	1
1	0

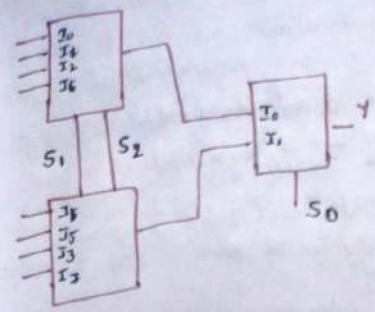


8:1 Mux using 4:1 mux & 2:1



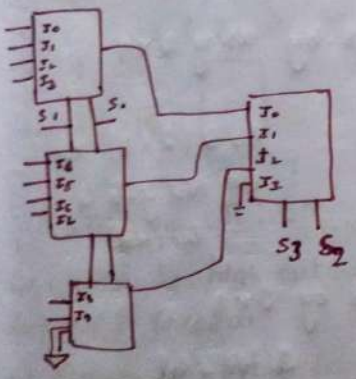
Actual

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

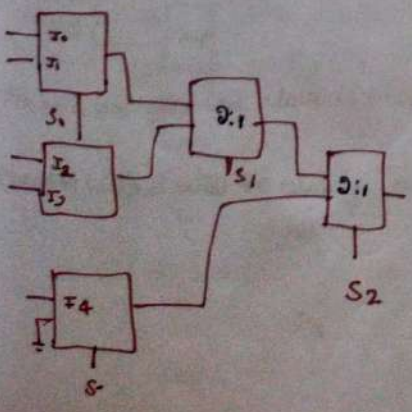


S_0	S_1	S_2	Y
0	0	0	I_0
0	0	1	I_4
0	1	0	I_2
0	1	1	I_6
1	0	0	I_1
1	0	1	I_5
1	1	0	I_3
1	1	1	I_7

Imp 10:1 Mux using 4:1 Mux



5:1 Mux using 2:1 Mux.



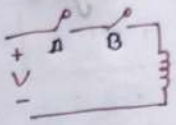
NOTE: In this below Example No need to connect the 3rd mux in the Level-1 we can directly connect.

Switching circuit.

NOT Gate :-

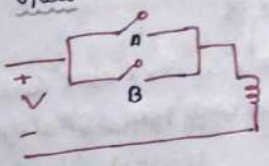
A	Y
0	1
1	0

AND Gate :-



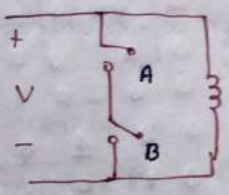
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate :-



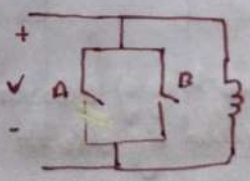
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NAND Gate :-



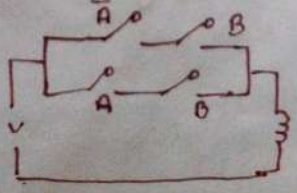
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate :-



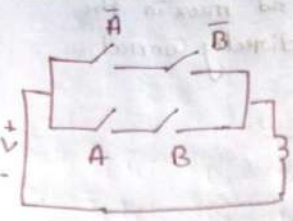
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Exor Gate :-



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

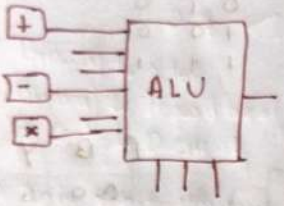
Exnor Gate :-



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

Application of MUX

① data Selector
② Serial to Parall data
③ ALU (based on select line it perform the operation)



Opcode

000	+
001	−
010	×
110	%

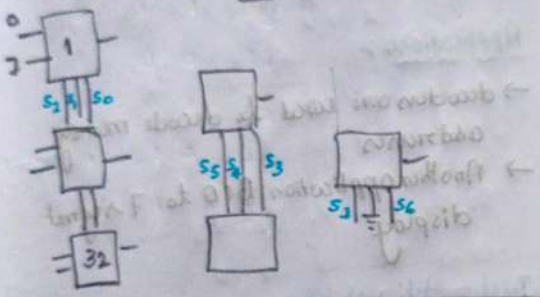
17/03/2025.

- In Mux if their is a Minterms use the VEM method.
- If their is a boolean function use the Shannon Expansion.
- The number of 2x1 mux required to implement 256x1 mux is 255.
- 256x1 mux using 8x1 mux.

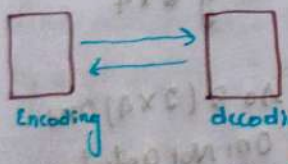
256/8 32/8 4/8

32 4 1

$S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$



Decoder :-

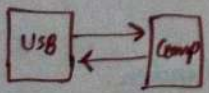


$2^n : n$ $n : 2^n$

→ decoder has high output line Compared to Encoder.

→ decoder less input lines.

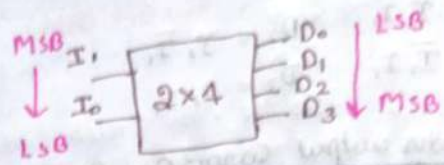
Application :-



Master is Controller but Slave is receiver

→ Decoder output will be always minterm

basic block :-



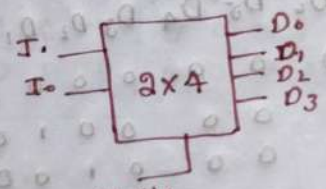
$I_0(A)$	$I_0(B)$	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$D_0 = \bar{A}\bar{B}$ $D_2 = A\bar{B}$

$D_1 = \bar{A}B$ $D_3 = AB$.

Note :- Implimentation of decoder of require of one gate gate of not.

→ We can contol the circuit by using the enable circuit.



En	I_1	I_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0
	0	1	0	0	0	0
	1	0	0	0	0	0
	1	1	0	0	0	0
1	0	0	0	0	0	1
	0	1	0	0	1	0
	1	0	0	1	0	0
	1	1	1	0	0	0

2 x 4 decoder.

$D_0 = \bar{I}_2 \bar{I}_1$ $\quad D_2 = I_1 \bar{I}_1$

$D_1 = \bar{I}_2 I_1$ $\quad D_3 = I_1 I_1$.

→ decoder output corresponding to minterm.

→ We can have decoder with 2^n output

3:6 ✓

2:3 ✓

⇒ 2:5 ✗

2:4 ✓

→ Binary to octal
Binary to Hexadecimal
BCD to Decimal

→ In in nbit unused don't care combination decoder output will have 2^n output.

I_2 I_1 I_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_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

What type of questions may ask?

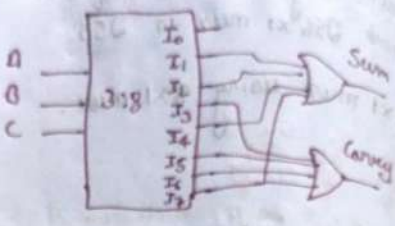
① Explain the working of decoder
② Boolean function by decoder

→ Bubble will be denoting the active low logic.

Design and impliment full adder using decoder.

Sum = $\Sigma(1, 2, 4, 7)$

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



→ decoder with enable will behave like demultiplier.

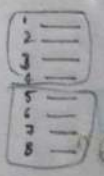
Applications :-

→ decoders are used to decode memory addresses

→ Another application BCD to 7 segment display

Implimentation of higher order decoder by using lower order decoder.

→ 3x8 by using 2x4



No 2 (2x4) and One Not gate to control.

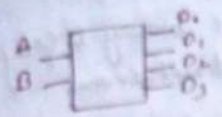
⇒ 3x8 by 2x4

8/4 2 for control.

2

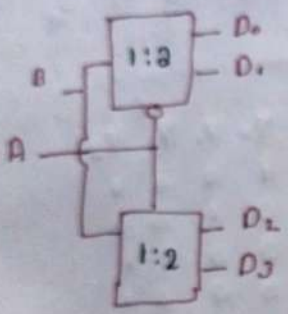
MSB → ctrl input
LSB → input of lower decoder.

2:4 decoder using 1:2 decoder

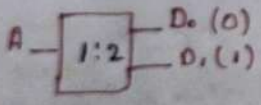


A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

We can observe here when A=0 only D_0 D_1 are active and A=1 D_2 & D_3 are active

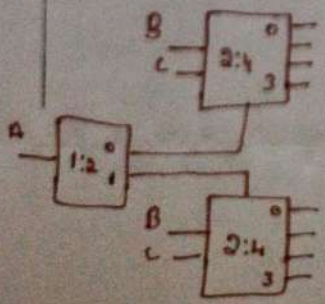


And Not can also be implimented by a other logic 1:2 decoder



3:8 decoder using 2:4 decoder

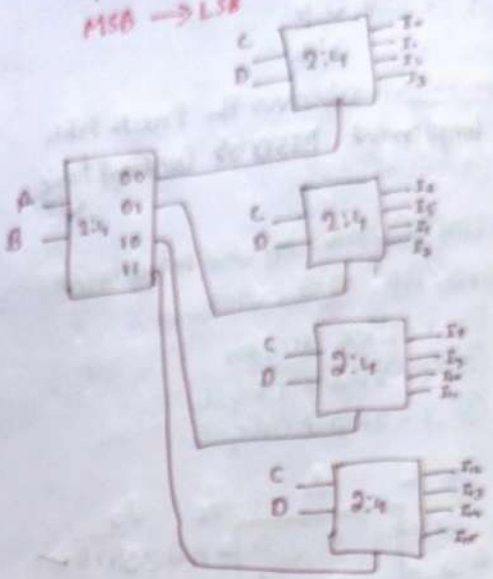
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7



4x16 decoder using ① 2x4 decoder ② 3x8 decoder & 1x2

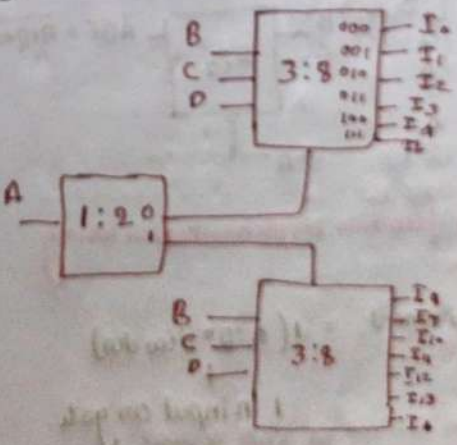
ⓐ $\frac{16}{4}$ 4 $\frac{4}{4}$ 1

MSB → LSB



A	B	C	D	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0

ⓑ

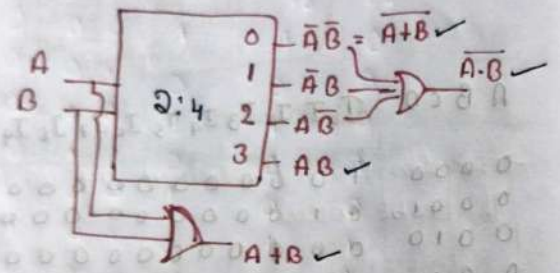


NOTE: Solve & Revise this.

① 2:4 → 1:2
② 3:8 → 1:2
2:4
③ 4:16 → 2:4
3:8

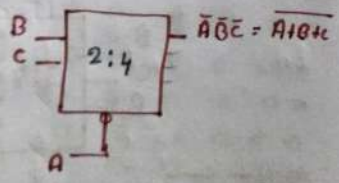
Superstrick: Observer the truth table to implement MSB of Control logic.

Q Use three resource and impliment
A+B, AB, $\overline{A.B}$, $\overline{A+B}$
① 1 - 2x4 decoder
② 1 - 2 input OR Gate
③ 1 - 3 input OR Gate.



Design and impliment 3 input Nor gate using 2:4 decoder

A	B	C	Y
0	0	0	1
0	0	1	0
⋮			0
⋮			⋮
			0

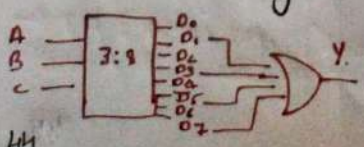


Boolean function implementation using decoder

① Requirement = 1 (n : 2^n decoder)
+
1 n input or gate.

$Y(A,B,C) = \Sigma m(1,3,5,7)$

① 3:8 decoder + or gate.

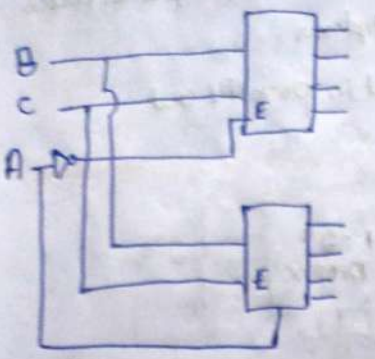


H.W

① Design a 3-bit binary to Gray Converter & implement with mux

② BCD to Excess 3 code Convert
Convert to boolean Exp & implem by in minterm

3:8 by using 2:4



19/03/2025

Application of decoder:

→ In Master Slave circuit, whenever the master want to communicate with slave.

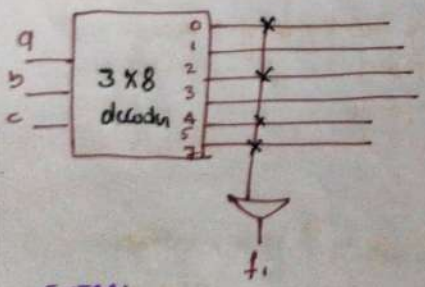
Q) Design a combinational ckt which accepts 3 inputs and provide 3 output.

a) High when a combination is Even
b) ABC are pallindrome
c) Only two or more bit are one

a	b	c	f_1	f_2	f_3
0	0	0	1	1	0
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	0	1	1

$f_1 = \Sigma(0,2,4,6)$; $f_3 = \Sigma(3,5,6,7)$

$f_2 = \Sigma(0,2,5,7)$



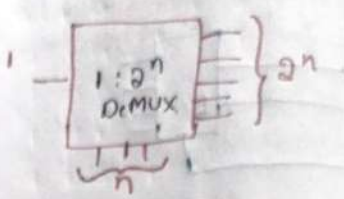
EXTRA:

→ Above circuit is ROM circuit
→ For PLD & PAL fusing & Antifusing techn
→ Mentorgraphics Dun by SIMENS
→ AMD buy Xilinx.
→ TTL & CMOS logic [7400 & 54LS]
→ Which IC is used for 7446, 7447

De-Multiplexer

→ Decoder with control input is called demultiplexer

→ Serial in parallel out.



→ 1:4 demux using 1:2 demux

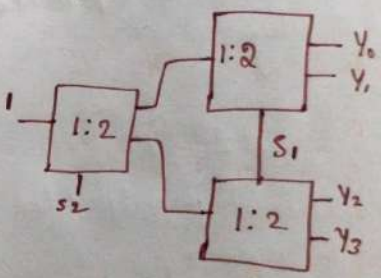
$\frac{4}{2} \quad \frac{2}{2}$

$2 + 1 = 3$

$(n-1) = (4-1) = 3$

2x4 Demux using 1x2 demux

S_2	S_1	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

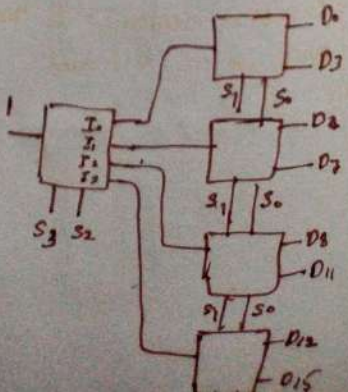


Note: In DMUX MSB → LSB

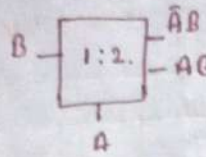
Design 1x16 Demux using 1x4 demux

$\frac{16}{4} \quad \frac{4}{4}$

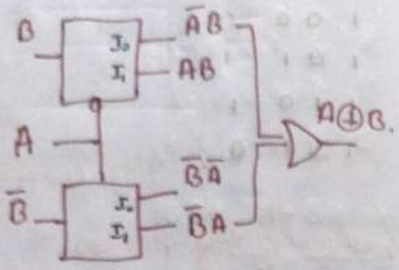
$4 + 1 = 5$



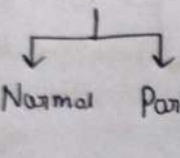
And Gate by Demux.

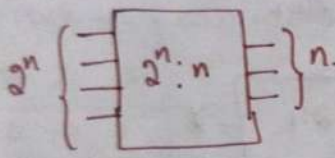


2input XOR Gate by Demux (1:2)



Encoder:- Secure the data & Compress the data this Encoder can be used.





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

Limitation: What if the more number of ones are there on single line

Priority Encoder is designed to handel multiple ones in the input sequence.

Priority is given for MSB and then LSB.

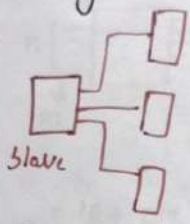
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

Priority order:- higher the subscript higher the preority

Application:-

Orbiter:- designed using priority Encoder

→ For controlling the shared resources and providing the priority.



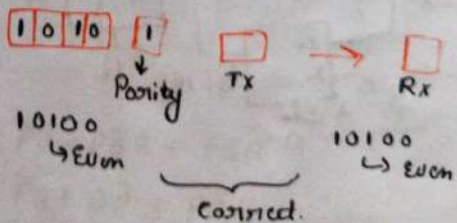
→ This may create the Starvision problem this is the limitation

→ This can be corrected by using Round Robin Algorithm.

→ Implment using Verilog

Concept of parity:-

→ Refers to no of ones in a row is called parity. And they are of 2 types
- odd parity.
- Even parity.



→ If it is odd it will be wrong

→ Parity will fail when multiple bit flip.

→ Single bit Error detection & Correction

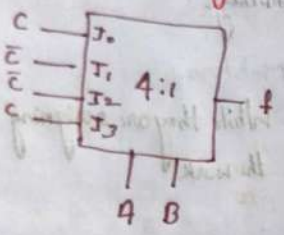
→ CRC, LFSR technique are used

Design 3-bit even parity generator ckt

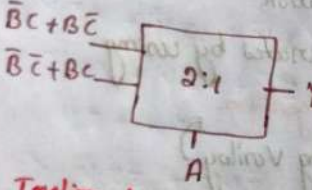
A	B	C	f_e	
0	0	0	0	C
0	0	1	1	
0	1	0	1	$\bar{C}$
0	1	1	0	
1	0	0	1	$\bar{C}$
1	0	1	0	
1	1	0	0	C
1	1	1	1	

↑ Think how to make Even

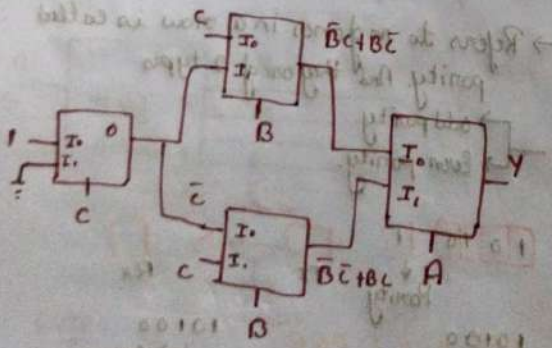
impliment this using 4:1 mux :-



by 2:1 Mux :-



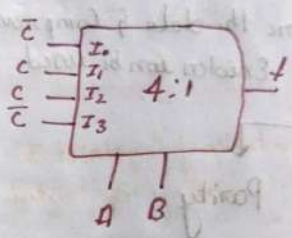
Impliment only by 2:1 Mux



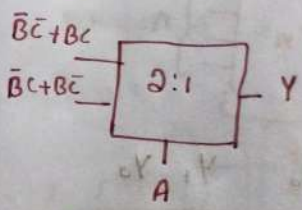
Design 3-bit odd parity generator ckt

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

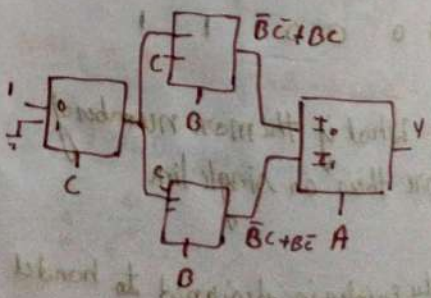
impliment using 4:1 mux



By 2:1 mux



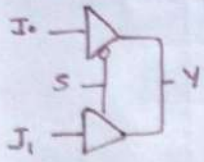
impliment only by 2:1 Mux :-



19/03/2025

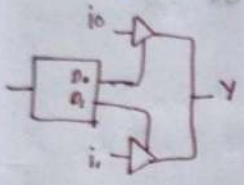
① Impliment mux using tristate buffer.

2:1



S	B_1	B_2	Y
0	Ac	Jn	J_0
1	In	A	I_1

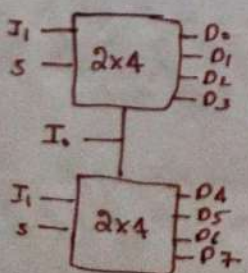
② decoder using ctrl logic



Hint:- In the implimentation of any tristate buffer replace the mux by its tristate buffer.

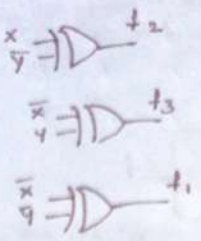
③ 2:1 Mux using 2x4 decoder

I_0	I_1	S	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



$\Sigma(3,4,4,7)$

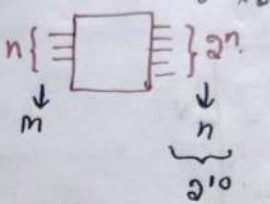
① Which of the following is not an exclusive xnor.



② If there are m input lines and n output lines for a decoder that is used to uniqily address only 1KB RAM, where data width 1byte long, what is the min value of m+n.

Size = M x n

$1KB = 1 \times 2^{10} \times 2^3$



$= 10 + 2^{10}$

$= 10 + 1024$

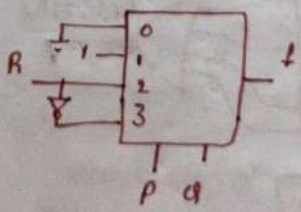
1034

③ How many 3:8 lines need with the input are need to design 6:64 line decoder with any other logic gate.

$\frac{64}{8} \quad \frac{8}{8}$

$8 + 1 = 9$

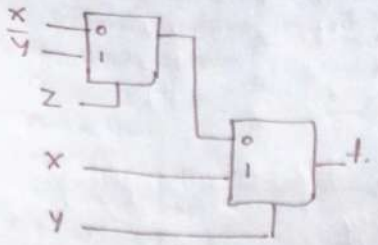
④ Output of Mux



$= \bar{P}0 + P\bar{Q}R + PQ\bar{R}$

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

⑤ Output of the Mux



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

$\bar{z}x\bar{y} + z\bar{y} + xy$

$= \bar{y}(xz + \bar{z}) + xy$

$= x + \bar{y}z$

A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

$X = \Sigma(1,2,3,4)$

$Y = \Sigma(1,2,5,6)$

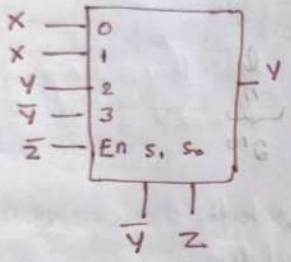
$Z = \Sigma(1,3,5,7)$

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

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

$Z = C.$

⑥



$= \bar{\bar{y}}\bar{z}x + \bar{\bar{y}}zx + \bar{y}\bar{z}y + \bar{y}zy$

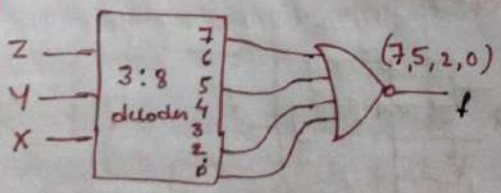
$= y\bar{z}x + yzx + \bar{y}z$

$= \bar{y}z + xy$

$\Rightarrow \bar{y}z\bar{z} + xy\bar{z}$

$f = xy\bar{z}$

⑦



$f = x \oplus z$

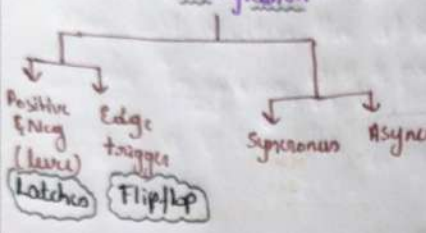
⑧ Design a Combinational circuit for generating 2's Compliment of 3 bit binary number impliment this with ⓐ 2 level AND OR
ⓑ using 4×1 Mux.

Sequential circuits :

- Are those which are depends on clock cycles and depends on present as well as past inputs to generate any output.
- Speed is slow.
- because of triggerning
- Designing is tough
- Time dependent
- Elimentary building blocks are flipflops and latches
- Mainly used in storing a data. (1-bit)
- These circuit has memory
- It is not easy to use and handel.

Examples:- latch, flipflop, Counter, shift register.

Classification



→ latches are level trigger & flipflop is Edge trigger device.

→ In single clock the latch can produce the multiple transition leads to Race Condition.

→ Flipflops are race free.

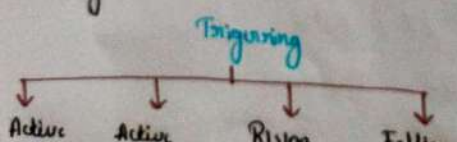
NOTE:- If the sequence are inactive refers to it will be in the previous state.

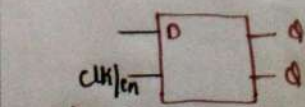
- But in Combinational the output leads to high impedence state.

→ Latches are easy to design will acts as Combinational.

→ In latch the clk is acts as a Enable signal

Triggering

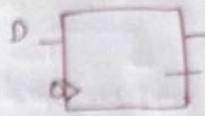




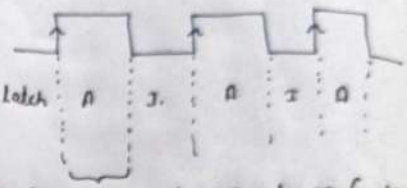
52 Active high level triggering

Active low level sensitive

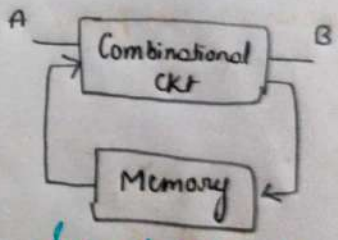
Edge triggering rising



Falling Edge triggering



- At this portion it will acts as Combinational Circuit.
- We can alter the inputs which may also leads to race around condition
- Solution is using a flip flop we can manage this



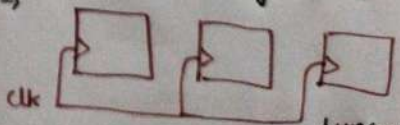
Sequential circuit.

Synchronous Vs Asynchronous :-

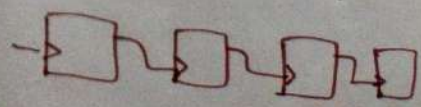
→ In Synchronous single clock to activate

→ Different clock in Asynchronous

→



Synchronous

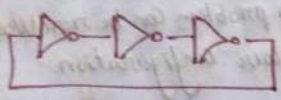


Asynchronous

→ Synchronous output will be predictable
→ Most of application we use Asynchronous ckt.
→ In now a days we are using piplining technique
→ Total propogation delay in asynchronous is high but individually very low.
→ Synchronous → Shift register
Asynchronous → Counters & frequency divider
Asynchronous logic is used in low power and high speed applications.

→ Flipflop is also called bistable multivibrator

Ringossilator:



→ Latches are more prone to noise while flip flop are less.
↳ because the latches are less parone to the noise beg of triggering

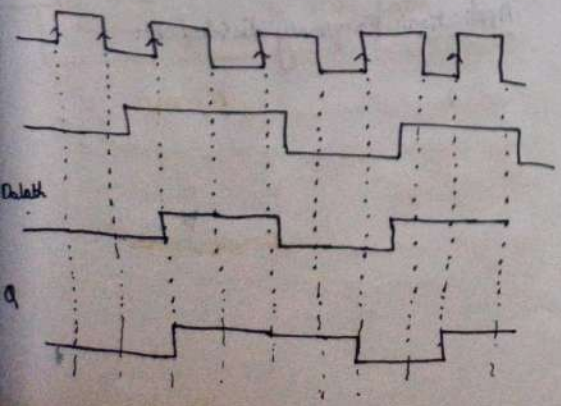
D latch and D flip flop difference:

Clk	D	Q
0	x	Q N.c
1	0	0
1	1	1

D latch

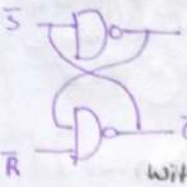
clk	D	Q
↑	0	0
↑	1	1
↑	x	Q

D-flip flop.

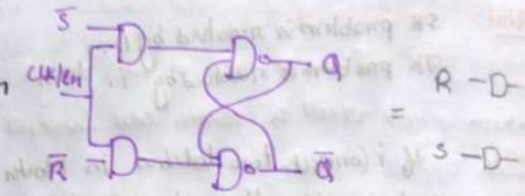


S R latch.

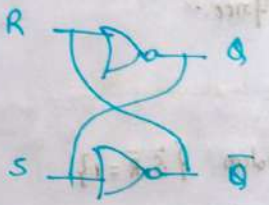
→ It is also called Set & Rst latch
→ Construct using Nand and Nor



Enable = clk.

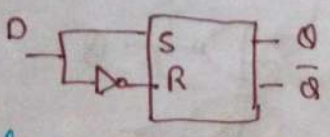


S	R	Q	
0	0	N.c	
0	1	0	Reset
1	0	1	set
1	1	x	Prohibited.

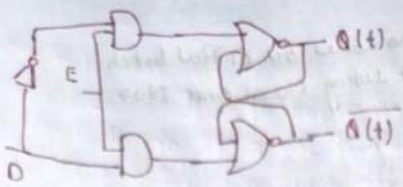


S	R	
0	0	N.c
0	1	Reset
1	0	set
1	1	Prohibited / Forbidden

limitation: Here the both the inputs are high it is creating forbidden condition.
So Now we can compliment the two inputs and give



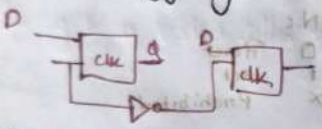
hence it is called D-flip flop. sorry it is D-latch.



En	D	Q	
0	X	Q	(Inactive)
1	0	0	
1	1	1	

Hint:- SR problem is resolved by D. JK problem is resolved by T. } latch

NOTE:- If i connect two latches in Master Slave configuration then it will be Kind of edge triggering will be produce



→ Flip flop is race free.

Nand [Forbidden]

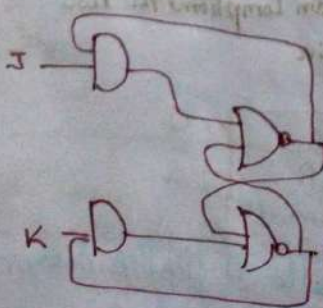
S = R = Forbidden $\{\bar{S}, \bar{R} = 1\}$

Nor [Forbidden]

S = R = Forbidden $\{S, R = 0\}$

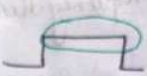
JK - Latch.

The only difference between SR Latch JK latch is that there is no output feedback towards the input in the SR latch but it is present in JK latch



J	K	Q	$\overline{Q}$	State
0	0	Q_t	$\overline{Q}_t$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	$\overline{Q}_t$	Q_t	Toggle

→ This will create Race condition at J = K = 1 only on $t_{on} > t_{pd}$



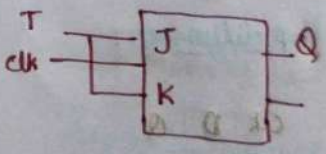
→ large ON time is necessary due to the input will change at different level leads to different outputs

Hint:- To resolve this to reduce will come down to t_{pd}. And it is so difficult but it can be reduce by master slave config of JK flip flop. (latch sorry)

→ So this problem can be resolved master slave configuration.

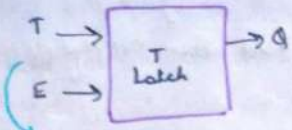
T Latch

J = K = T.



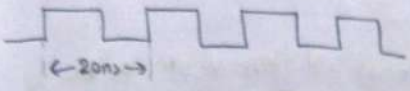
T	Q	
0	Q	N.C
1	$\overline{Q}$	T

Application:- Frequency divider. ckt



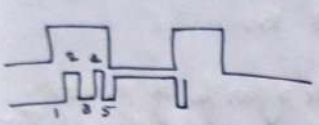
output cannot determine if latch or delay
no

Duty cycle :- Percentage of time for which clock is on.

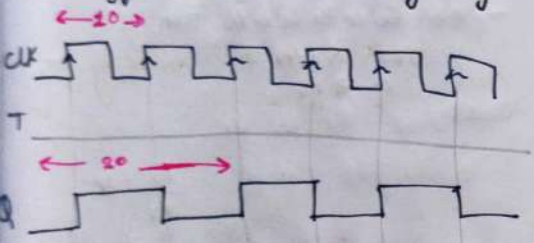


$T_{ON} = 50 \times T_P = 0.5 \times 20 = 10ns$

$T_{OFF} = 50 \times T_P = 0.5 \times 20 = 10ns$



toggle due because it is logic high

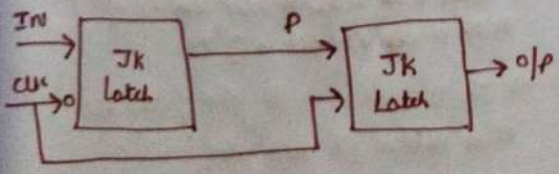


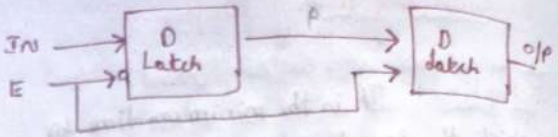
→ so it is producing the stable result.

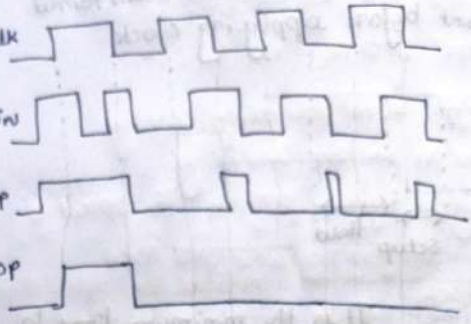
Drawback of J.K latch :-

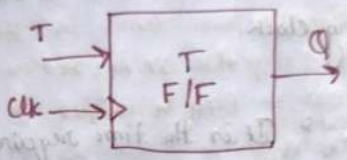
Why to avoid race around Condition

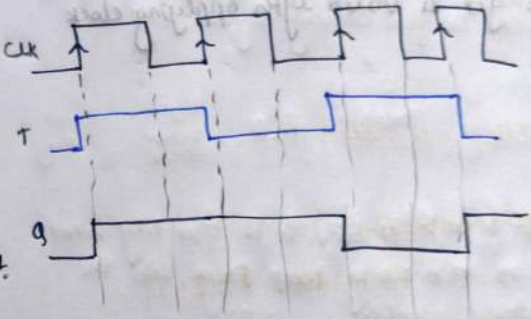
1. $T_P(latch) > T_{clk}$
2. Master Slave F.F
3. J-K always complemented to each other





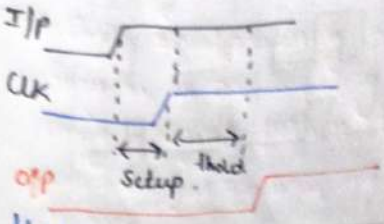






Switching times of flip flop

Setup time :- It is the minimum time for which the input must be maintained constant before applying clock



Hold time :- It is the minimum time for which input must be maintained constant after applying clock.

Propagation delay :- It is the time required to get a valid after applying clock

Data Converters:-

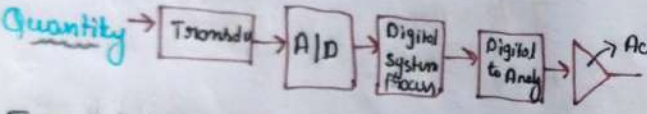
→ The purpose is to convert analog to digital and digital to analog.

→ Digital signals are defined for all interval of time. - NO

→ Digital signal processing is very Easy so we are converting analog-to digital, perform the operation whichever you required and send it back.

→ Processing refers to Converting signal into required form.

→ Transmitting signal is analog and ag for processing we need to convert into digital and again for the reciver we need to give analog signal only.



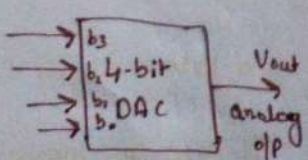
Transducer :- It converts the quantity what we are measuring temp, pressure, into the Electrical quantity.

Actuator :- To control physical variable

Digital to Analog Converter :-

→ We are giving input in the form of binary. And we are getting analog output for each inputs.

→ output in the form of voltage (clock signal it is in the form of voltage



→ In every DAC the resolution is fix stepsize

Resolution | Step size :-

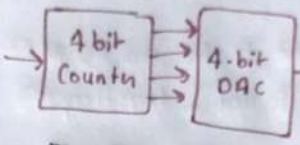
Increment in Vout when binary i/p increases

0000 → V

0001 → V'

Stepsize = V' - V.

" It is called pseudo analog " because we will not get the between values here.



max^m Vout of full scale.

O/p = 15V

→ Full scale refers to the maximum value of the output of the system

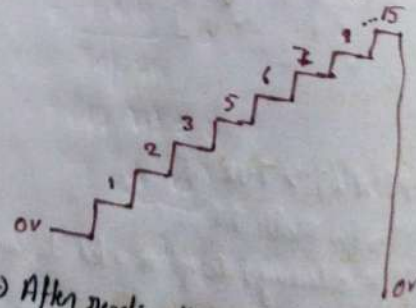
→ 4 bit Counter counts from

0000 → 1111

0 → 15

Resolution = $\frac{\text{Max out}}{\text{no of Counter (in)}} = \frac{15}{15} = 1V$

→ Since the resolution is one



→ After reaching this maximum 15 if I/p reset it will reach to zero.

→ The output is in the form of Staircase it is called 'Pseudo analog'.

→ As a input increases by 1 DAC output will change according to resolution of stepsize.

→ gap b/w two steps are called stepsize.

→ The output of the DAC will be always multiple of Resolution. Not all values are possible. Only certain values are possible at the output, which are not perfectly continous

→ If I reduce the size we can we can make it analog.

→ By increasing the number of bit we can reduce the stepsize of Resolution.

→ To make the output more analog we keep the Full scale output fix but increase number of bits, so that resoultion reduces.

Eg:- If there is 5 bit, maximum input

11111

$$\frac{15}{3R} = \frac{0.5}{}$$

Parameters of DAC :-

① Resolution :- It is the smallest change that occur in output due to change in digital input.

ⓐ 0010 → V
0011 → V'

Resolution = (V'-V).

ⓑ 0000 → 0
0001 → h

Resolution = h-0 = h.

→ The output when all bits are zero and LSB is one, represents the resolution of the DAC.

Resolution = $\frac{\text{Step size (h)}}{\text{Full Scale Value}} \times 100\%$

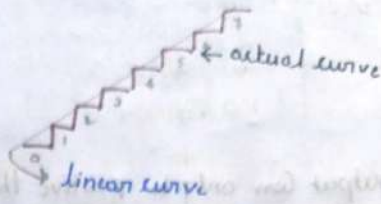
in previous eg

$= \frac{1}{15\ (\text{total no of steps})} \times 100\%$

$= \frac{1}{\text{no of steps}}$

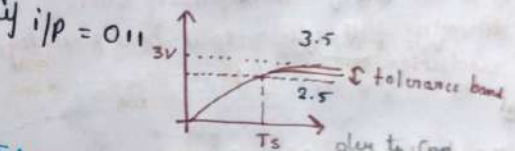
$= \frac{1}{2^n - 1} \times 100\%$ (n = no of bits)

Accuracy :- The accuracy of DAC is specified in terms of full scale error, linearity error, expressed as percentage of full scale output.



→ Full scale error is the deviation of the actual output from the linear curve. If we written with % is called accuracy.

Settling time :- The time taken by the output to settle within the tolerance band around the steady state value is called settling time



Tolerance band :- The nearest value of the required output is called tolerance, so that our work can be done. And time req to reach of that point is called settling time.

Tolerance band = ± 1/2 LSB
= ± h/2

h = 5

Offset Voltage :- The output of DAC at 0 input is called as offset voltage mean if we apply a zero voltage but we are getting nonzero because of nonlinearity is called offset voltage.

→ Ideally offset voltage should be zero

Monotonicity :- If the output increases with increase in binary input then DAC is said to monotonous.

Unipolar and bipolar DAC :-

- In unipolar the sign remain same either +ve or -ve.
- But if DAC will produce both the type is called bipolar.

Eg in sign magnitude

0 001 = 1

1 001 = -1

If the output can only be positive then DAC is said to be Unipolar. In such DAC unsigned binary nuber should be input

If output can be both +ve & -ve then it is said to be bipolar DAC. And such DAC we use signed binary numbers.

A 6-bit DAC on a step size of 50mv determine full scale output and percentage resolution.

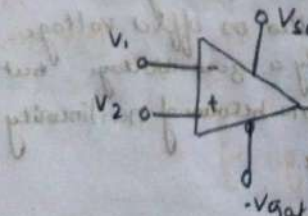
Sol

- no of steps = $2^6 - 1 = 63$.
- full scale o/p = 0.05×63
  $= 3.15V$.

% resolution = $\frac{0.05}{3.15} \times 100$

$= \frac{100}{63} = 1.587\%$

Virtual ground :-



$V_{out} = A_d \cdot V_d$

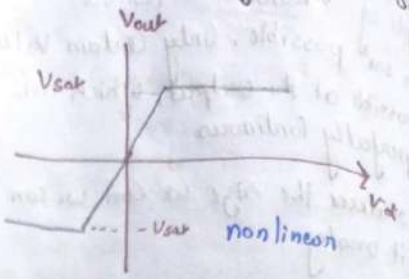
$= A_d (V_2 - V_1)$

A_d - gain

V_d = differential input (+ve - ve terminal)

Open loop gain = A_d = very high

Rule :- Vout can't go beyond V_{sat}



→ For small input output changes into saturation

→ In Opamp output should be propotional to input.

→ To make that into Equal & propotional, increase the linear region

→ linear region extended by -ve feedback o/p connected to negative terminal of input.

→ $V_{out} = A_d (V_2 - V_1)$

$V_2 - V_1 = V_d = \frac{V_{out}}{A_d}$

V_{out} = finite $A_d \to \infty$

$= \frac{\text{Finite}}{\text{Infinite}} = 0$

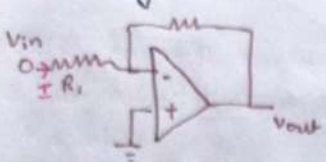
$= (V_2 - V_1) = 0$

$V_1 = V_2$

By using the -ve feedback and make the voltage same is called Virtual ground.

The Amplifier made by Opamp of two type ① Inverting
② Non Inverting.

Inverting: The output will be inverted & negative.



By Virtual ground $V_- = V_+ = 0$

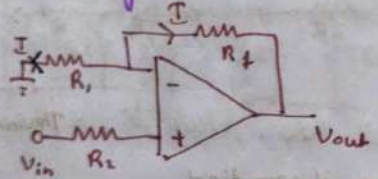
$$I = \frac{V_{in} - 0}{R_1}$$

opamp has zero input current.

$$I = \frac{0 - V_o}{R_f} = \frac{V_{in}}{R_1}$$

$$V_o = -\frac{R_f}{R_1} V_{in}$$

Non inverting:



Virtual ground $V_- = V_+$

R_2 has no Voltage drop

$$V_+ = V_{in} = V_-$$

$$I = \frac{V_{in} - 0}{R_1} = \frac{V_{in}}{R_1}$$

$$I = \frac{V_{out} - V_{in}}{R_f} = \frac{V_{in}}{R_1}$$

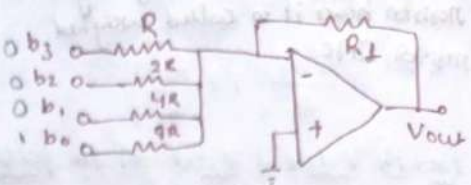
$$V_{out} = V_{in} + V_{in} R_f / R_1$$

$$V_{out} = V_{in}\left(1 + \frac{R_f}{R_1}\right)$$

gain

① Weighted Resistor DAC.
② R-2R ladder Network.

① Weighted Resistor DAC.



If $b_n = 1$, $b_n = V_{ref}$
$b_n = 0$ $b_n = 0$

$$b_n = b_n V_{ref}$$

→ Inverting amplifier.
→ The output can be obtained by superposition.

→ $$V_{out} = -\frac{R_f}{R}(b_3 V_{ref}) - \frac{R_f}{2R}(b_2 V_{ref}) - \frac{R_f}{4R}(b_1 V_{ref}) - \frac{R_f}{8R}(b_0 V_{ref})$$

$$= -\frac{R_f}{8R} V_{ref}(2^3 b_3 + 2^2 b_2 + 2^1 b_1 + b_0)$$

$$= \underbrace{\frac{-R_f}{2^{n-1} R}}_{\text{resolution}} V_{ref} \times (\text{decimal value of i/p})$$

→ To get resolution LSB = 1 ; others = 0

→ Amplifier output = input × gain

So output we will get by only b_0

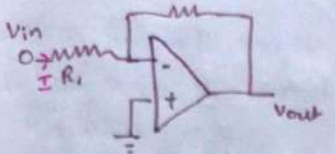
$$V_{out} = -\frac{R_f}{8R} V_{ref} = \text{resolution}$$

$$= V_{out} = \text{Resolution} \times \text{decimal value of i/p.}$$

→ They are in a fixed ratio

The Amplifier made by Opamp of two type ① Inverting ② Non Inverting.

Inverting: The output will be inverted or negative.



By Virtual ground $V_- = V_+ = 0$

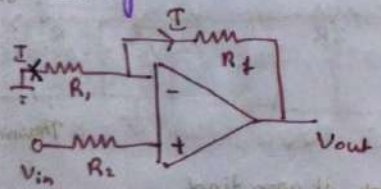
$$I = \frac{V_{in} - 0}{R_1}$$

opamp has zero input current.

$$I = \frac{0 - V_o}{R_f} = \frac{V_{in}}{R_1}$$

$$\boxed{V_o = -\frac{R_f}{R_1} V_{in}}$$

Non inverting:



Virtual ground $V_- = V_+$

R_2 has no Voltage drop

$$V_+ = V_{in} = V_-$$

$$I = \frac{V_{in} - 0}{R_1} = \frac{V_{in}}{R_1}$$

$$I = \frac{V_{out} - V_{in}}{R_f} = \frac{V_{in}}{R_1}$$

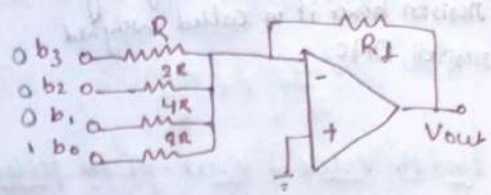
$$V_{out} = V_{in} + V_{in} R_f / R_1$$

$$\boxed{V_{out} = V_{in}\left(1 + \frac{R_f}{R_1}\right)}$$

gain

① Weighted Resistor DAC.
② R-2R ladder Network.

① Weighted Resistor DAC.



If $b_n = 1$, $b_n = V_{ref}$

$b_n = 0$ $b_n = 0$

$$\boxed{b_n = b_n V_{ref}}$$

→ Inverting amplifier.

→ The output can be obtained by superposition.

→ $$V_{out} = -\frac{R_f}{R}(b_3 V_{ref}) - \frac{R_f}{2R}(b_2 V_{ref}) - \frac{R_f}{4R}(b_1 V_{ref}) - \frac{R_f}{8R}(b_0 V_{ref})$$

$$= -\frac{R_f}{8R} V_{ref}(2^3 b_3 + 2^2 b_2 + 2^1 b_1 + b_0)$$

$$= \underbrace{\frac{-R_f}{2^{n-1} R}}_{\text{resolution}} V_{ref} \times (\text{decimal value of i/p})$$

→ To get resolution LSB = 1 ; others = 0

→ Amplifier output = input × gain

So output we will get by only b_0

$$V_{out} = -\frac{R_f}{8R} V_{ref} = \text{resolution}$$

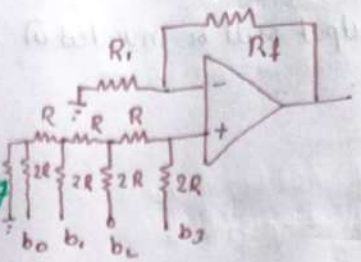
$$= V_{out} = \text{resolution} \times \text{decimal value of i/p.}$$

→ They are in a fixed ratio

Note: higest weighed bit has minimum resistance, minimum bit has maximum resistance, so the output depends on the weights of the resistor hence it is called weighted resistor DAC

R-2R ladder network

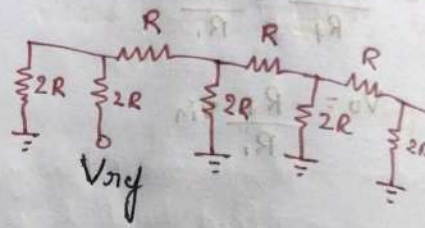
i) Non inverting configuration

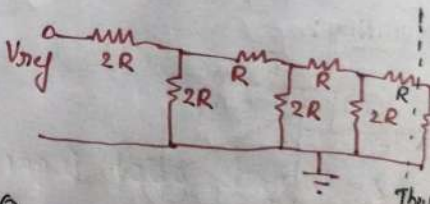


→ MSB = the bit near to opamp
LSB = far from opamp

If $b_i = 1$ $b_i = V_{ref}$
$b_i = 0$ $b_i = 0$

Case i: input 0001

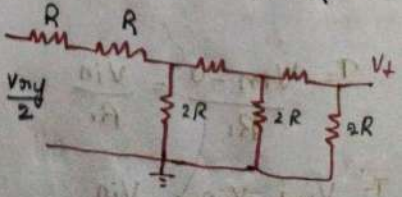




① make it open. find
② Req by series & parallel

Voltage divide $V_{th} = V_{ref} \times \frac{2R}{4R} = \frac{V_{ref}}{2}$

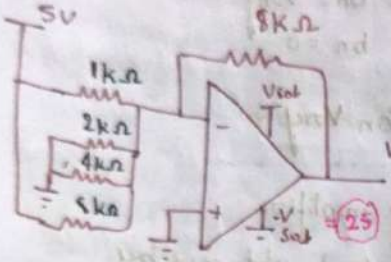
$R_{th} = 2R \parallel 2R = R$



$V_{th} = \frac{V_{ref}}{2} \times \frac{2R}{4R} = \frac{V_{ref}}{4}$

$R_{th} = 2R \parallel 2R = R$

Find the value of Vout for the following circuit.



→ We are using superposition theorem
→ It is inverting amplifier

$$V_o = -\frac{R_f}{R_i} V_{in}$$

$$V_{out} = -\frac{8}{1} \times 5 - \frac{8}{8} \times 5$$

$$= -45V$$

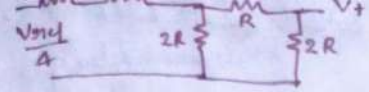
but $V_{sat} = 25$

Opamp output can never exceed V_{sat} limit -25 to 25

So output = -25V

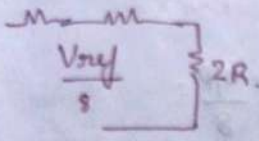
Resolution: Making only LSB bit has logic high, Others to zero.

For any input output will be taken by Resolution x decimal value of input.



$V_{Th} = \frac{V_{ref}}{4} \times \frac{2R}{4R} = \frac{V_{ref}}{8}$

$R_{th} = 2R \,||\, 2R = R$



$V_t = \frac{V_{ref}}{8} \times \frac{2R}{2R+R+R}$

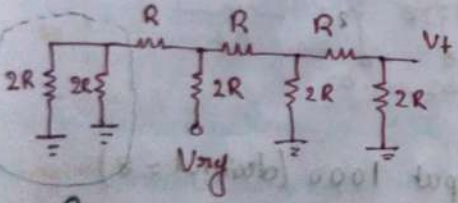
$= \frac{V_{ref}}{16}$

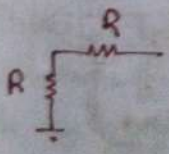
if D001

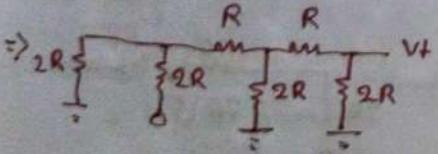
$$V_{out} = \frac{V_{ref}}{16}\left(1 + \frac{R_f}{R_1}\right)$$

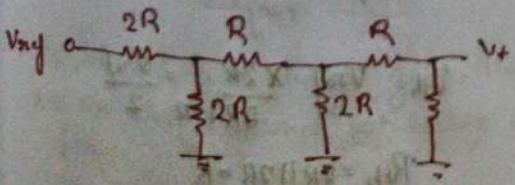
(resolution)

Case-2 input 0010



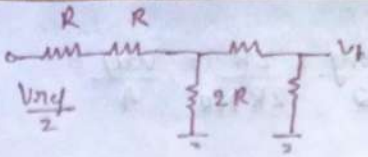






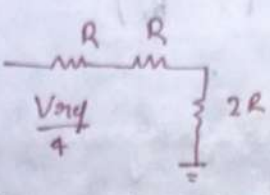
$V_{Th} = V_{ref} \times \frac{2R}{2R+2R} = \frac{V_{ref}}{2}$

$R_{Th} = 2R \,||\, 2R = R$



$V_{Th} = \frac{V_{ref}}{2} \frac{2R}{2R+2R} = \frac{V_{ref}}{4}$

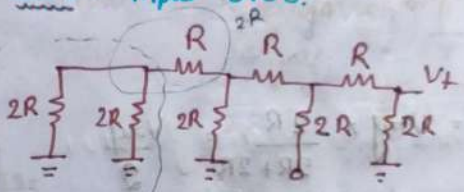
$R_{Th} = 2R \,||\, 2R = R$

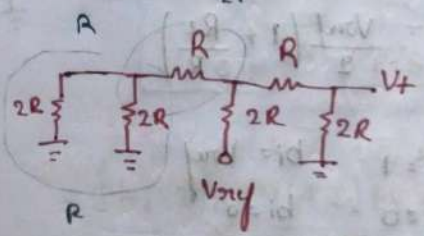


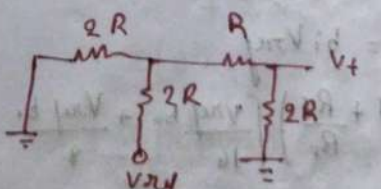
$V_{Th} = \frac{V_{ref}}{4} \frac{2R}{2R+2R} = \frac{V_{ref}}{8}$

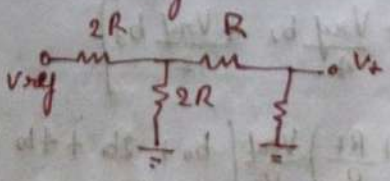
$$V_{out} = \frac{V_{ref}}{8}\left(1 + \frac{R_f}{R_1}\right)$$

Case 3 : input 0100.



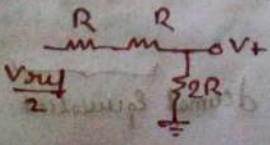






$V_{Th} = V_{ref} \times \frac{2R}{2R+2R} = \frac{V_{ref}}{2}$

$R_{Th} = 2R \,||\, 2R = R$

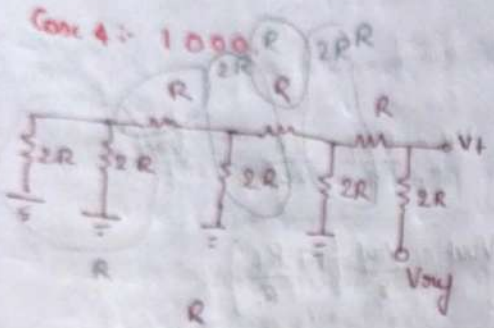


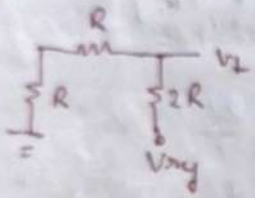
$$V_t = \frac{V_{ref}}{2} \times \frac{2R}{2R+2R} = \frac{V_{ref}}{4}$$

i/p 0100

$$V_{out} = \frac{V_{ref}}{4}\left(1 + \frac{R_f}{R_1}\right)$$

Case 4:- 1000





$$V_t = V_{ref} \times \frac{2R}{2R+2R} = \frac{V_{ref}}{2}$$

$$V_{out} = \frac{V_{ref}}{2}\left(1 + \frac{R_f}{R_1}\right)$$

If $b_i = 1 \quad b_i = V_{ref}$

$b_i = 0 \quad b_i = 0$

$$b_i = b_i V_{ref}$$

$$V_{out} = \left(1 + \frac{R_f}{R_1}\right)\left[\frac{V_{ref}}{16} b_0 + \frac{V_{ref}}{8} b_1 + \frac{V_{ref}}{4} b_2 + \frac{V_{ref}}{2} b_3\right]$$

$$V_{out} = \left(1 + \frac{R_f}{R_1}\right)\frac{V_{ref}}{16}\left[b_0 + 2b_1 + 4b_2 + 8b_3\right]$$

Resolution

$2^3\ 2^2\ 2^1\ 2^0$

$b_3\ b_2\ b_1\ b_0$

$8b_3 + 4b_2 + 2b_1 + b_0$

decimal Equivalent

Hint:- output of DAC = resolution × decimal value of i/p

→ Finding output at MSB is very easy.

if i/p = 1000

o/p = resolution × 8.

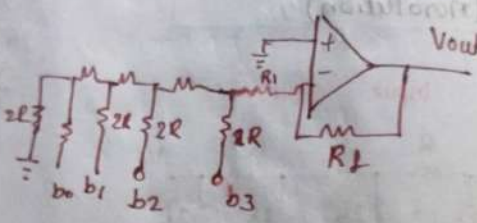
$$\text{resolution} = \frac{o/p}{8} =$$

for i/p 1000

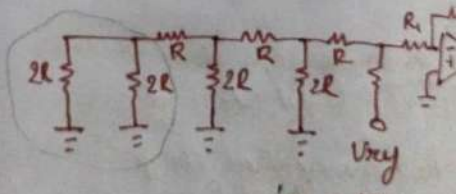
$$V_{out} = \frac{V_{ref}}{2}\left(1 + \frac{R_f}{R_1}\right)$$

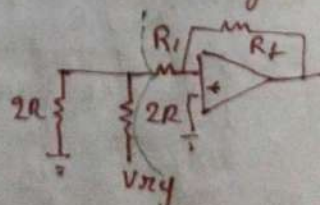
$$= \frac{V_{ref}}{16}\left(1 + \frac{R_f}{R_1}\right)$$

Inverting Configuration :-



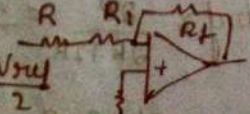
input 1000 (decimal = 8)





$$V_{th} = V_{ref} \times \frac{2R}{2R+2R} = \frac{V_{ref}}{2}$$

$$R_{th} = 2R \parallel 2R = R$$



$$\frac{V_{ref}}{2}\left(\frac{-R_f}{R+R_1}\right)$$

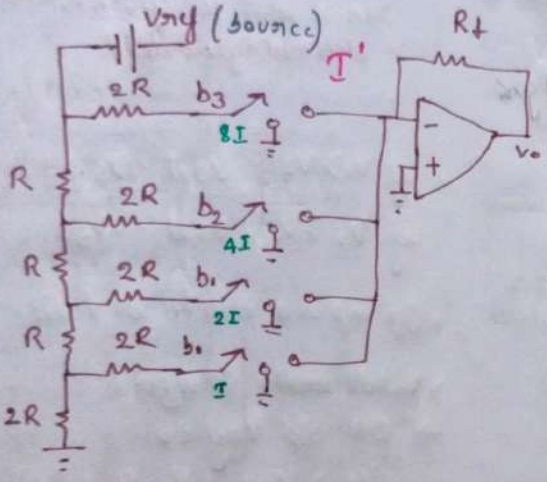
Vout = resolution × 8

$$\text{Resolution} = \frac{V_{ref}}{16}\left(\frac{-R_f}{R+R_1}\right)$$

General formula.

$$V_{out} = \frac{V_{ref}}{16}\left(\frac{-R_f}{R+R_1}\right)\left[2^3 b_3 + 2^2 b_2 + 2^1 b_1 + b_0\right]$$

③ Current Switched R2R ladder Network.

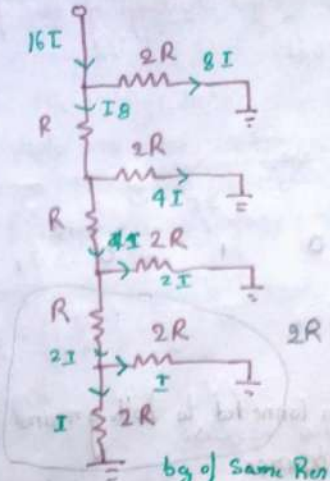


The bit nearest to Vref is MSB.
And the bit nearest to ground is LSB.

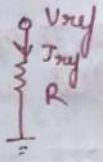
Before to make input one we are connecting with Vref, but here we need to connect for opamp

-Ve feedback
-Virtual ground
$V - V_f = 0$.

→ Switch voltage is always zero, here we can change current but voltage is fix



So the current is varying every where hence it is called, Current Switch



$$I_{ref} = \frac{V_{ref}}{R} = 16I$$

$$I = \frac{V_{ref}}{16R}$$

I' is.

$$I' = (8I)b_3 + (4I)b_2 + (2I)b_1 + Ib_0$$

$$= I\left(2^3 b_3 + 2^2 b_2 + 2^1 b_1 + 2^0 b_0\right)$$

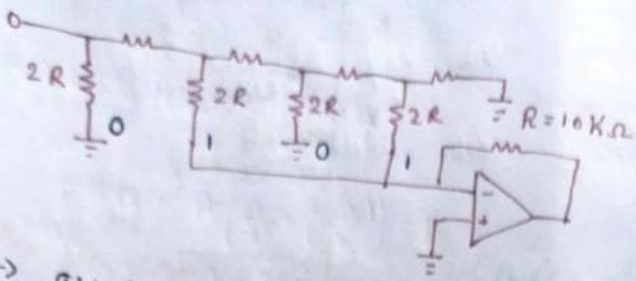
$$= I\,(\text{decimal Equivalent})$$

$$I' = \frac{0 - V_o}{R_f}$$

$$V_o = -I' R_f$$

$$V_o = -\frac{R_f}{16R} V_{ref}\,(\text{decimal Equivalent})$$

Find the output voltage of the following DAC circuit.



→ It is connected to both ground & also with the source.

→ Resistance connected with Vref or ground is called Voltage switched.

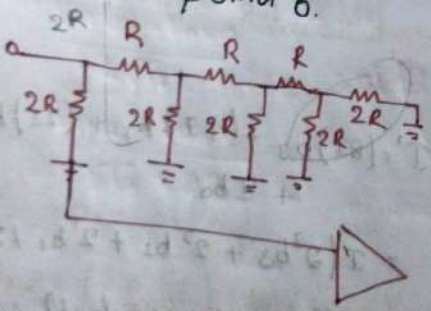
→ Resistor connected with the opamp or gnd called current switched.

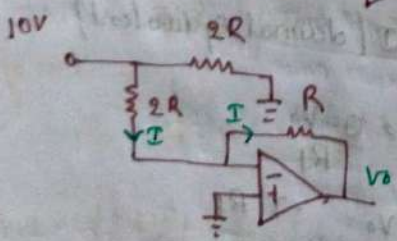
→ So it is connected to opamp & ground it is called current switched.

So we need to find output for 0101.

sol^n

make LSB=1 & other 0.





$I = \frac{V_{ref}}{2R} = \frac{0 - V_o}{R}$, $V_o = -\frac{V_{ref}}{2} = \frac{-10}{2}$

$= -5V$

V_o = res x decimal value

-5 = res $\times 8$

res = $\frac{-5}{8}$

decimal value of 0101 = 5

$V_o = -\frac{5}{8} \times 5 = -\frac{25}{8}$

$= -3.125V$

Rule:- MSB=1, Baki bit=0 find V_o. (from V_o we can find resolution). from that find resolution.

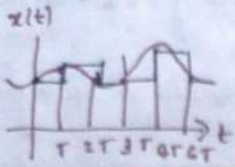
After finding resolution v_o = res x with the given bit decimal Equivalent.

Analog to Digital converter :-

Step-1 :- Sampling

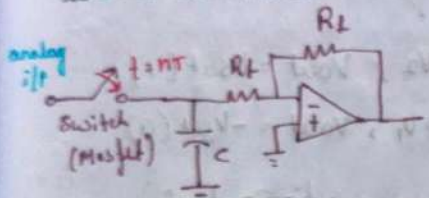
Step 2 :- Quantization

Sampling :-



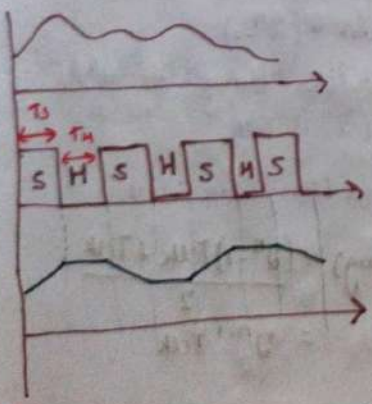
- it is a process in which we collect certain samples of a signal at a particular time instant
- If the samples are periodically placed in time then it is called uniform sampling.

Sample & Hold circuit



at t=0 S closed
t=T S closed
t=2T S closed
} momentarily (not closed at all time)

When swith is open $i=0$ $\frac{dv}{dt}=0$ V_c = Constant (hold)



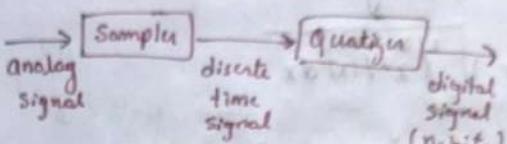
Acquisition time :- is the time during which capacitor acquire the charge supplied by the source. Ts

Aperture time :- When the switch is turned off the current flow don't stop instantly and during this interval capacitor acquires a charge from the source and this time is called as aperture time

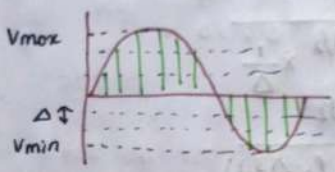
③ **Voltage droop :-** due to leakage in the capacitor the voltage across the capacitor during hold time does not remain constant and decrease gradually is called voltage droop

The leakage in the capacitor should as less as possible.

Quantization :-



no of Combination = 2^n.
no of levels = 2^n.



Δ = step size (resolution)

$\Delta = \frac{V_{max} - V_{min}}{L}$

→ closing to the nearest voltage

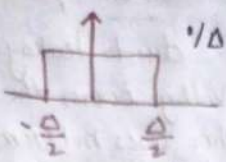
Levels
2
1.5
1.0
0.5
0
-0.5
-1
-1.5

if we get 1.642 → 1.5
1.976 → 2

- Quantization is a process of rounding of the sampled values to the nearest level
- By this rounding it may create an Error called **Quantization Error**.

1.642 → 1.5 Error = 0.142
1.976 → 2 Error = -0.024.
0.75 → 1 Error = -0.25
→ 0.5 Error = 0.25

maximum quantization Error = $\frac{\Delta}{2}$
min Error always lies in b/w.



Probability (total) = $\int_{-\infty}^{\infty} f(x)\,dx$

$$\int_{-\Delta/2}^{\Delta/2} A\,dx = A\left(\frac{\Delta}{2} - \left(-\frac{\Delta}{2}\right)\right) = 1$$

rms value of noise

$$= \sqrt{\int_{-\infty}^{\infty} x^2 f(x)\,dx}$$

$$= \sqrt{\int_{-\Delta/2}^{\Delta/2} x^2 \frac{1}{\Delta}\,dx}$$

noise $= \sqrt{\left(\frac{x^3}{3}\right)_{-\Delta/2}^{\Delta/2} \times \frac{1}{\Delta}}$

$$\sqrt{\frac{1}{3\Delta}\left(\frac{\Delta^3}{8} - \left(-\frac{\Delta^3}{8}\right)\right)}$$

$$= \sqrt{\frac{\Delta^2}{12}} = \frac{\Delta}{\sqrt{12}}$$

$$= \frac{V_{max} - V_{min}}{L\sqrt{12}}$$

$$= \frac{V_{max} - V_{min}}{\sqrt{12} \times 2^n}$$

Signal to noise ratio:-

$$SNR = 20\log_{10} \frac{\text{rms signal}}{\text{rms noise}}$$

$$= 20\log_{10} \frac{(V_{max} - V_{min})/2\sqrt{2}}{(V_{max} - V_{min})\sqrt{2} \times 2^n}$$

$$= 20\log_{10} \sqrt{6} \times 2^{n-1}$$

$$SNR = 20\log\sqrt{6} + 20\log\log 2^{n-1}$$

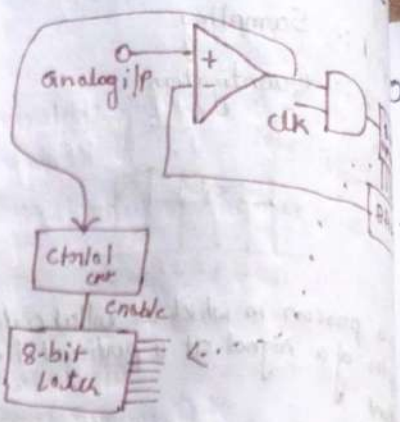
$$= 17.76 + 6.02n - 6.02$$

$$\boxed{SNR = 1.76 + 6.02n)\ dB}$$

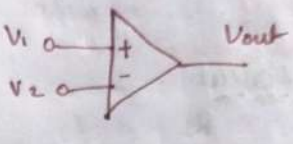
n: no of bits

CS

Counter type ADC. / ramp ADC



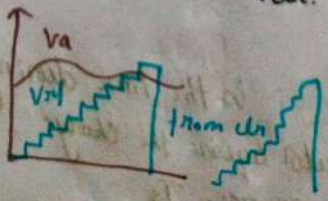
→ Opamp is used as Comparator.



- if $V_1 > V_2$, $V_{out} = V_{sat}(1)$
  if $V_2 > V_1$, $V_{out} = -V_{sat}(0)$
- Opamp is a non linear device

→ Initially Counter o/p is zero.
→ DAC o/p $V_{ref} = 0$
$V_a > V_{ref}$, $V_1 = V_{sat}(1)$
$X = 1 \cdot clk = clk$.

- Max^m Count $= (2^n - 1)$
- Maximum time taken for Conversion $= (2^n - 1)\,T_{clk}$.
- Minimum time for Con $= T_{clk}$
- $T_{conv}(avg) = \frac{(2^n - 1)T_{clk} + T_{clk}}{2}$
$= 2^{n-1} T_{clk}$.



It will take long time from initial to it will reach the maximum

Counter Keeps on Counting untill the Output of DAC exceeds the analog input.

The drawback is that we have to start the conversion from zero beg the Counter will not after the Conversion is Complete

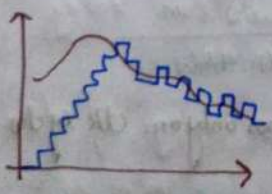
The Conversion time of Counter type ADC is directly proportional to the Analog input.

$T_{conv} \propto V_a$

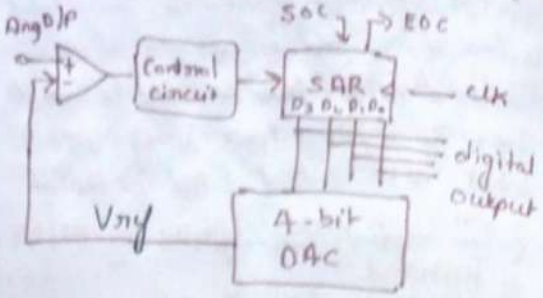
$$\frac{V_{a2}}{V_a} = \frac{T_2}{T_1}$$

Advancement of this is Tracking type ADC.

- Since Conversion in Counter type ADC always starts from zero time taken is high. but this type is accurately track, and follow analog input.



Successive Approximation Register (SAR)



SOC Start of Conversion.
EOC End of Conversion.

SAR

$V_a > V_{ref}$ SAR will set current bit at clock pulse

$V_a < V_{ref}$ SAR will reset previous bit at set current bit at clock pulse

SAR type ADC will continue to work untill all the bits have been accessed one by one, and once it happen, it will make EOC. signal high to indicate the Conversion has Ended.

resolution of DAC = 1V

$V_a = 10.2V$

$t = t_{clk} \cdot$ SAR set MSB.

SAR = 1000

$V_{ref} = 1 \times 8 = 8V$

$V_a > V_{ref}$

at $t = 2T_{clk}$; SAR 1100

$V_{ref} = 1 \times 12 = 12V$

$V_a < V_{ref}$

at $t = 3T_{clk}$ SAR 1010

$V_{ref} = 1 \times 10 = 10V$

$V_a = 10.2V > V_{ref}$

$t = 4T_{ref}$ SAR 1011 → final.

The Conversion time of SAR type ADC is fixed (n × Tclk)

A 8bit SAR type ADC is used to Convert 3.5 V to digital output Vref = 5V. What will be output of ADC at the end of 3rd clock pulse after the Conversion if initiated.

Va = 3.5V [analog input]
Vref = 5V
8-bit ADC

resolution of DAC = $\frac{5}{2^8 - 1}$

$= \frac{5}{255} = \frac{1}{51}$.

Tclk → o/p 1000 0000

Vref = $\frac{1}{51} \times 128$

$= 2.45V < Va$

so present bit one dont touch previous bit

o/p = 1100 0000] 2 Tclk

$\frac{128+64}{51} = \frac{192}{51} = 3.7 > Va$

3Tclk make precous bit one and next bit one

o/p 1010 0000

For final output (Trick)

→ decimal Equal of input

$= \frac{3.5}{1/51}$

$= 3.5 \times 51$

$= 178.5$

→ round off to nearest odd integer

$= 179$.

→ Convert to binary

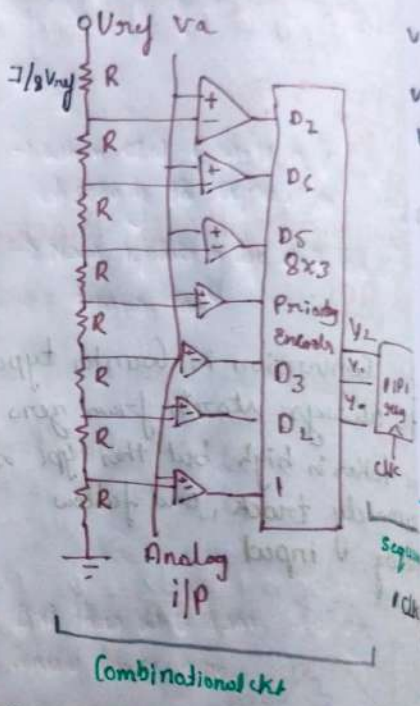
$(10110011)_2$

→ In output of SAR LSB is always high

→ decimal value is always odd

Flash type ADC :-

→ Fastest ADC
→ doesn't require DAC



→ It requires only one clk cycle.

Tconv = Tck

$I = \frac{Vref}{8}$

$V_7 = -\frac{Vref}{8R} R$

$= +Vref$

$7/8\ Vref$

$V_6 = 6/8\ Vref$

$V_5 = 5/8\ Vref$

$V_4 = 4/8\ Vref$

$V_a = \frac{2}{8} V_{ref}$

V_a	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
$< \frac{V_{ref}}{8}$	0	0	0	0	0	0	0	1	→ 000
$\frac{1}{8} < V_a < \frac{2}{8}$	0	0	0	0	0	0	1	1	→ 001
	0	0	0	0	0	1	1	1	→ 010
$\frac{2}{8} < V_a < \frac{3}{8}$	0	0	0	0	1	1	1	1	→ 011
	0	0	0	1	1	1	1	1	→ 100
$\frac{3}{8} < V_a < \frac{4}{8}$	0	0	1	1	1	1	1	1	→ 101
$\frac{4}{8} V_a < \frac{5}{8}$	0	1	1	1	1	1	1	1	→ 110
$\frac{5}{8} V_a < \frac{6}{8}$	1	1	1	1	1	1	1	1	→ 111
$\frac{6}{8} < V_a < \frac{7}{8}$									
$V_a > 7/8$									

$V_- > V_+ \quad V_o = 0$

$V_- < V_+ \quad V_o = 1$

no of resistance $= 2^n$

no of Comparitor $= 2^n - 1$

no of Encoder $= 1$

↳ order $(2^n : n)$

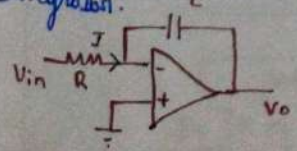
→ If there is no priority Encoder the Conversion time = 0.

Points :-

① If PIPO register is not present then no clock pulses is required and time required for conversion of propogation delay is almost zero.

Dual slope integration type ADC :-

Integration.



By Virtual ground.

$V_+ = V_- = 0$

$I = \frac{V_{in} - 0}{R} = \frac{V_{in}}{R}$

$C \frac{d}{dt} (0 - V_o) = I$

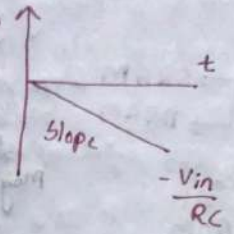
$-C \frac{dv_o}{dt} = \frac{V_{in}}{R}$

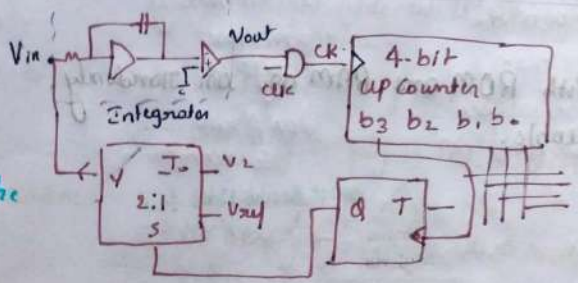
$\frac{dV_o}{dt} = -\frac{V_{in}}{RC}$

$V_o = -\frac{1}{RC} \int V_{in}\, dt$

if V_{in} (constant)

$V_o = -\frac{V_{in}}{RC} t$

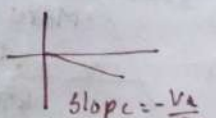




Initially Counter & FF are clear

$S = 0$, $V_{in} = V_a$

$V_{out} = -\frac{V_a}{RC} t$



Since $V_{out} < 0$

$V_+ > V_-$ V_1 = logic 1

Counter increases count until reach a

Count stops when $V_{out} \to 0^+$

at $t = 0$ $V_{out} = -\frac{V_a}{RC} t = 0$

at $= 15\, T_{clk}$

$V_{out} = -\frac{V_a t}{RC} < 0$

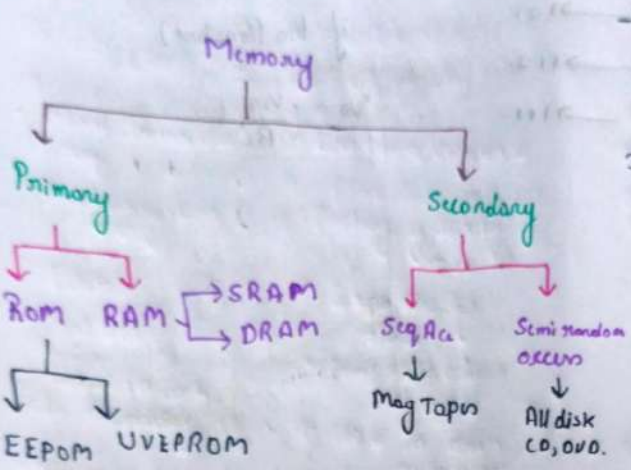
V_1 logic 1 Counter ↑

at $t = 16\, T_{clk} = T_1$

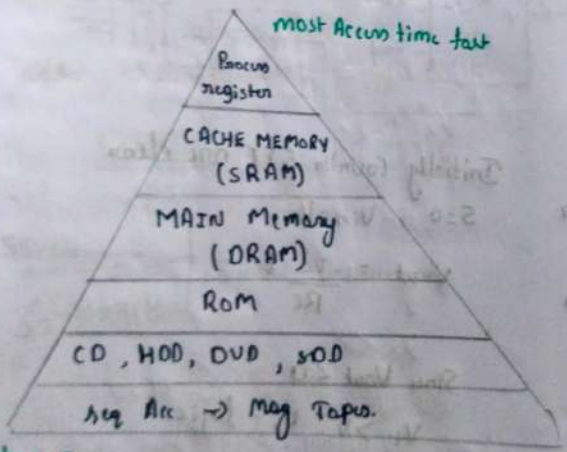
$V_{out} = -\frac{V_a}{RC} T_1$

out starts to increase due to -Vref and when it try to cross 0. the output of comparitor become low and clock of ckt Enter the counter, so the count stop.

Semiconductors Memories



→ Both ROM and RAM are are Randomly accessible.



ROM	RAM
• Read only memory	• Read & Write memory
• Random Access	• Random Access
• Non Volatile	• Volatile memory
• used to store	• used to Stor temp data
• Permanent data like system prog. OS	

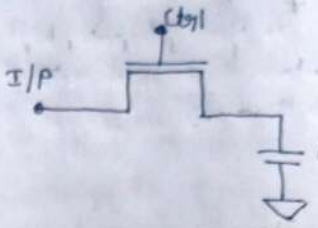
EEPROM — ↓ Electrically Erasable

UVPROM — ↓ ultra violet Erasable.

D-RAM :-

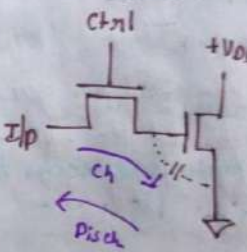
→ Main work to charge & discharge the capacitor

→ By charging store 1 by discharge store 0



→ When ctrl = 1 will helps to the discharge the capacitor.

→ Now replacing cap with another transistor.

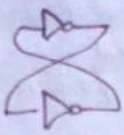


→ But this will some leakage Conductance here.

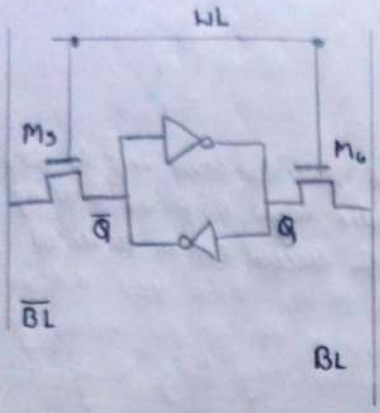
→ So periodic charge (refresh) is

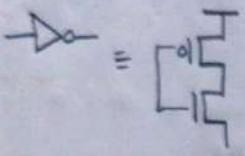
→ Compare to SRAM it is Slow.

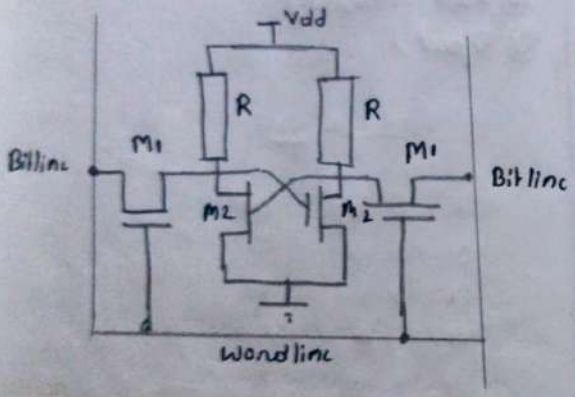
SRAM



→ Basic memory Element is flipflop or latch



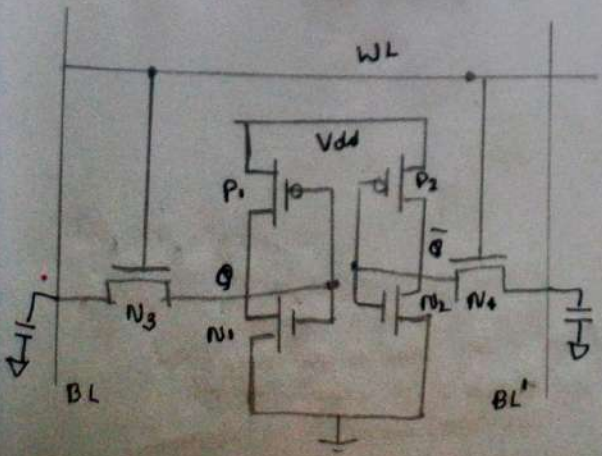




→ It is called 4T SRAM.

→ Above is a NMOS version.

→ If we replace by CMOS. it become 6T SRAM.



6T SRAM CELL

→ This has 2 operation one is the read & another is to write.

Read Operation :-

- Now we need to read the data and the data is already there
- Assume $Q = 0$ & $\bar{Q} = 1$
- WL = 1 {Activate}
- And we have 2 capacitor it is called precharged capacitance

OFF	ON	initial
ON	OFF	

→ Capacitor already charged called Precharged.

→ N_3 = ON. → capacitor is completely discharge

→ But Right side will (Cap) will remain same.

BL $\overline{BL} = 1$ } initially

→ Sensing amplifier will detect.

→ N_1 stronger than the N_3

→ N_3 strong resistance to ON than N_1.

→ For proper operation N_1 on resistance < N_3 on resistance.

$$R_{N_1} < R_{N_3}$$

$$\frac{1}{\left(\frac{W}{L}\right)_1} < \frac{1}{\left(\frac{W}{L}\right)_3}$$

Right operation :-

→ We need to write the data

→ Initially $Q = 0$ $\bar{Q} = 1$.

OFF	ON
ON	OFF.

→ Write $Q = 1$; $\bar{Q} = 0$

$BL = 1$; $\overline{BL} = 0$.

→ WL = 1 (Fix)

char → discharge

$Q = 1$ $\bar{Q} = 0$

N_3 OFF

N_1 OFF

$RP_2 > R_{TV4}$

$R_1 > RN_2$

Diffrena b/w SRAM And DRAM :-

SRAM	DRAM
• More transistor req	• Less trans required
• Packaging Less	• Packaging desity more
• Faster	• Comparitively slow
• Costlier	• Comparitively chop
• Used in cach memory	• Used in main memory.
• No periodic refresh	• periodic refresh
• More power Consumption	dur power Consumption
• Data is stored in FF	• Data Stored in Changing & discharing of capacitar.
• Volatile memory	• Volatile
• Random Access	• Random access

20/03/2025

→ Race condition : when the output changes multiple time one clock.
→ It is impossible to predict the correct output.

SR Latch F.F

En/clk	S R	Q	$\bar{Q}$	
A	0 0	0	0	(Latch State)
A	0 1	0	1	(Reset)
A	1 0	1	0	(Set)
A	1 1	0	0	(Invalid, prohibited)
X	X X	Q	Q^+	(Previous State)

To avoid this
S = D
R = $\bar{D}$

D- Latch / F.F

Clk / En	D	Q	Q^+
A	0	0	1
Ac	1	1	0
In	X	Q	$\bar{Q}$

J-K Latch FF

Clk/En	J	K	Q	Q^+	
Inactive	X	X	Q	$\bar{Q}$	
Active	0	0	Q	Q^+	
A	0	1	0	1	
A	1	0	1	0	
A	1	1	$\bar{Q}$	Q	(Toggel)

J = K = T

T Latch FF

Clk	T	Q	$\bar{Q}$
A	0	Q	Q^+
A	1	$\bar{Q}$	Q
I	X	Q	Q^+

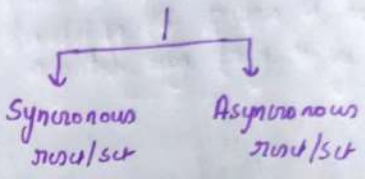
D latch → Shift register.
→ Memory.
→ Counters.

T flipflop for frequency division ckt.

Till now we only assume the intial condition in the groups with low & high, it is not a correct way in the hardware design, what to do? And how to solve this problem

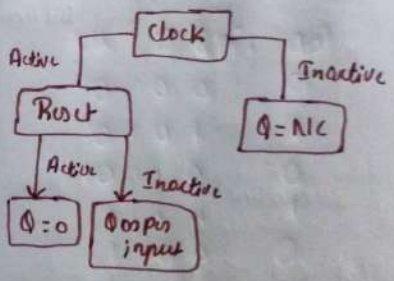
→ By using control inputs
① Reset = 0
② clear = 0
③ Preset = 1
④ Set = 1

- And this control inputs uses only negative logic, so power consumption of the circuit can be reduced.



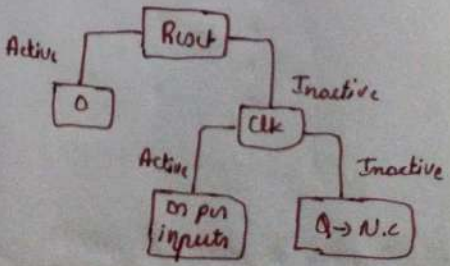
① Clk
⑤ Reset } Priority
③ Input

Synchronous (reset)

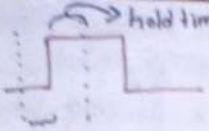


Asynchronous (reset)

→ Not depend on clock



Most of the devices are asynchronous ctrl inputs.



setup time.

D is the condition in which we dont know where it will go.

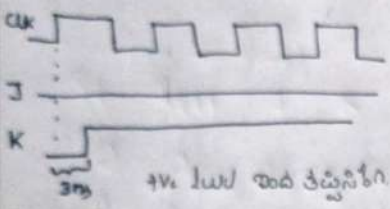
graph

JK

J	K	Q
0	0	N.C
0	1	Reset
1	0	Set
1	1	Toggle

→ Race around Condition.

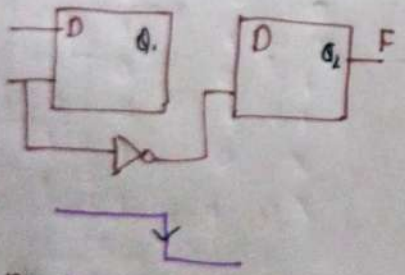
① To avoid this problem we need to increase the propogation delay.



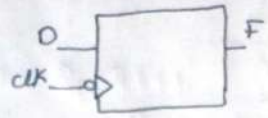
t_{pd} = 3ns

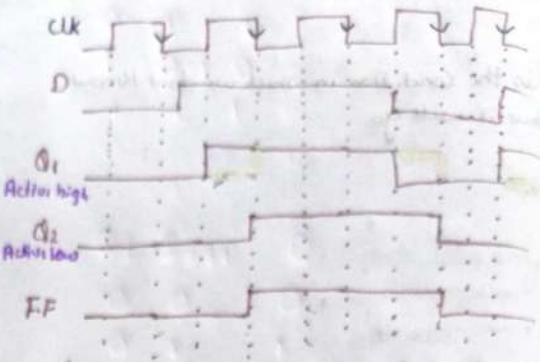
② Master slave configuration :-

- Only one of them is either master or slave
- Equalent to real implimentation by using Flip flop
- Edge triggering is also done by this master slave in the flip flop.



Negative Edge trigering.
Two latch combine to form a flip flop.

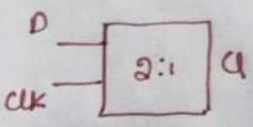




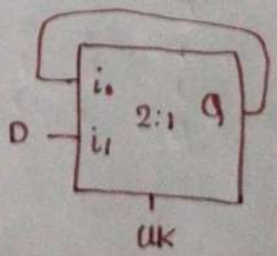
Hence the Q_2 = FF output at negative Edge hence it is the proper circuit for negative Edge trigerring

D. Latch using 2 x 1 Mux

- We need Expression of truth table to impliment this



clk	D	Q_p^-	q_{pt}	
0	0	0	0	N.C Q
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	Active D
1	0	1	0	
1	1	0	1	
1	1	1	1	



- observe the feedback connectivity by if sequential it is compulsory.
- This table is called characteristic table, tells the next state output based on clock and the previous state

. Characteristic equation is the next state Equation in terms of inputs of latch and flipflop and previous state.

→ Ignore the clock in the characteristic tabel.

Characteristic tabel for D-F-F

D	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	1

When clock is active.

	$\bar{Q}$	Q
$\bar{D}$		
D	1	1

$Q_{t+1} = D$

Characteristic Equation.

Characteristic tabel for T. F.F

T	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

	$\bar{Q}_t$	Q_t
$\bar{T}$	0	1
T	1	0

$Q_{t+1} = T \oplus Q_t$

Characteristic tabel for J-K F.F

	J	K	Q_t	Q_{t+1}
Memory	0	0	0	0
	0	0	1	1
Reset = 0	0	1	0	0
	0	1	1	0
Set = 1	1	0	0	1
	1	0	1	1
Toggle	1	1	0	1
	1	1	1	0

	$\bar{K}\bar{Q}$	$\bar{K}Q$	KQ	$K\bar{Q}$
$\bar{J}$		1		
J	1	1		1

$Q_{t+1} = \bar{K}Q + J\bar{Q}$

Characteristic tabel for SR F.F.

S	R	Q	Q_{t+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

	$\bar{R}\bar{Q}$	$\bar{R}Q$	RQ	$R\bar{Q}$
$\bar{S}$		1		
S	1	1	x	x

$Q_{t+1} = S + \bar{R}Q$

Latch F.F	C.E
D	$Q_{t+1} = D$
T	$Q_{t+1} = T \oplus Q_t$
J.K	$Q_{t+1} = \bar{K}Q + J\bar{Q}$
S.R	$Q_{t+1} = S + \bar{R}Q$

What is the use of Excitation tabel?

Based on the present and the next state we can find the output of the combination.

Why nand ?

① Low power
② high fanout
③ less time

. In Flash CMOS NAND is used.

J	K	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J.K Excitation tabel :-

Q_t	Q_{t+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

S R Excitation tabel :-

S	R	Q_t	Q_{t+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

Q_t	Q_{t+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

D- Flip flop :-

D_t	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

↑ = ↑

T- Flip flop.

T	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Conversion of one Flip flop to Another

Step 1: Identify what is available & what is required.

Available → X.
required → Y.

Step 2: Characteristic table for the req flip flop interms I/p Q_t & Q_{t+1}.

Step 3: by using above characteristic table findout the inputs for available flipflop

Step 4: Solve the K-map get the expression of available inputs in terms of required flip flop.

Step 5: Design the circuit as per the expression obtained.

Convert D flip flop into T flip flop :-

Step 1: Available = D
required = T

Step 2: Characteristic table of T flip flop

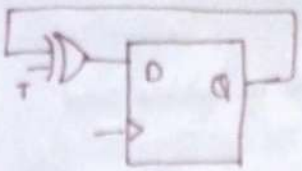
T	Q_t	Q_{t+1}	D
0	0	0	
0	1	1	
1	0	1	
1	1	0	

Step 3: → neglect this

T	Q_t	Q_{t+1}	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

	$\bar{Q}_t$	Q_t
$\bar{T}$	0	1
T	1	0

$D = T \oplus Q_t$

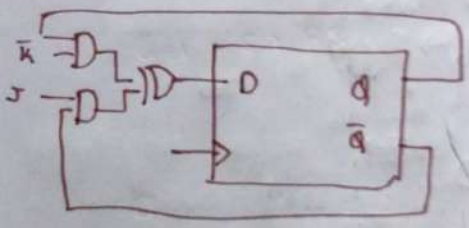


Convert D flip flop into J-K flip flop

J	K	Q	Q_{t+1}	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

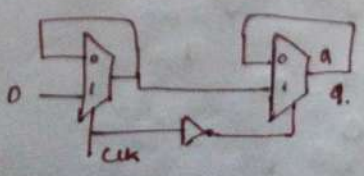
	$\bar{K}\bar{Q}$	$\bar{K}Q$	KQ	$K\bar{Q}$
$\bar{J}$		1		
J	1	1		1

$Q_{t+1} = J\bar{Q} + \bar{K}Q$



NOTE: For any implimentation of 2:1 Mux To will be given for the Q and I_2 will be its characteristic equation.

→ Two cascaded latches with opposite Clock will acted on the flip flop



→ Two Master slave configuration will detect the both the edges main 4 Master slave

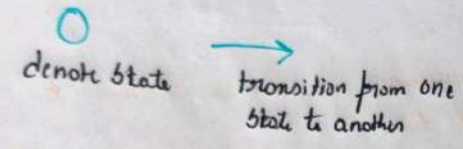
25/03/2025

→ Two Stable State in latches and flipflop.

Q	Qt	T	
0	0	0	
0	1	1	Change
1	0	1	
1	1	0	

State tabel :- It denots the transition from one state to another. And have a format and denotes, what is present state, next and exist state.

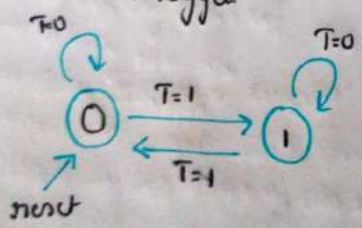
→ One flip flop has only 2 Status.



State diagram / tabel for flip flop :-

T F.F :-

0 → N.c
1 → Togget



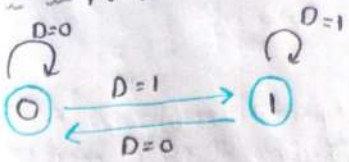
- Active low reset = 0
- Active high reset = 1.

② No change ↻

→ Above diagram is for Excitation tabel

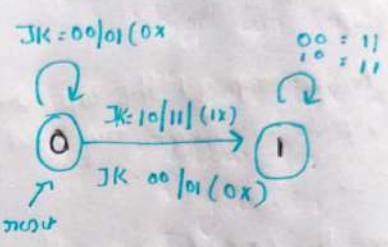
PS	T	NS
0	0	0
0	1	1
1	0	1
1	1	0

D - Flip flop :-



D	Qt	Qt+1
0	0	0
0	1	1
1	0	0
1	1	1

JK Flip flop



J	K	Qt
0	0	0
0	1	0
1	0	1
1	1	1

→ Set & preset are same it makes all the Values to 111.

Applications of sequential circuits

→ Flip flop for 1bit storage.
→ Used for data storage and transfer
→ Counters
→ Clock division
→ used as registers
→ used in PLD's, CPLD and FPGA's

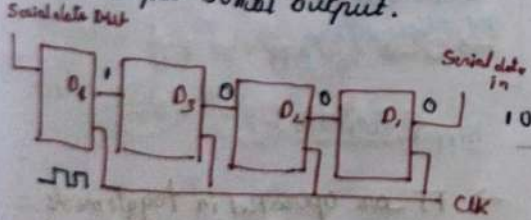
Registers :-

① For data storage.
② For data transfer.

→ They are group of flip flop
→ n bit n flip flop
→ Registers are designed by Synchronous circuit.
→ Types of Register :
① Shift register.
② Storage register.
→ We always use D flip flop in shift registers.
→ Combo SISO
PIPO
SIPO
PISO

Serial in Serial out.

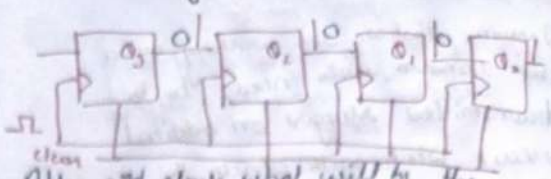
→ Serial input Serial output.



$Q_3 Q_2 Q_1 Q_0$	$Q_3 Q_2 Q_1 Q_0$
	1 0 0 0
1	0 0 0 1
1 0	0 0 1 0
1 0 0	0 1 0 1
1 0 0 0	1 0 1 1

Here we need 8 clock pulses to get the data out.

→ From LSB feeding 1011. Serial in parallel out
→ If MSB feeding 1101.



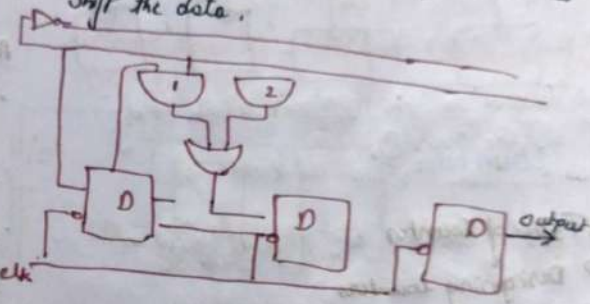
After 3rd clock what will be the Output

1 0 0 0 1st
1 1 0 0 2nd
0 1 1 0 3rd

→ Here we can take out the data from one clock pulse
→ Here for feeding we 4 clock but to receive one clock is enough

Parallel in Serial out :-

→ One additional ckt will be present to shift the data.

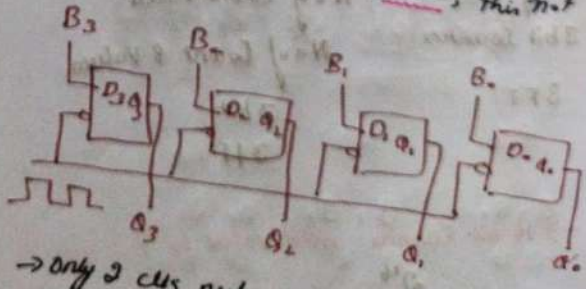


→ It will control the operation if it load condition not allowed to shift if it shift not allowed to load.

$\bar{S} + S.Q$

Parallel in Parallel out :-

→ Now time SISO, PISO, SIPO, this not used

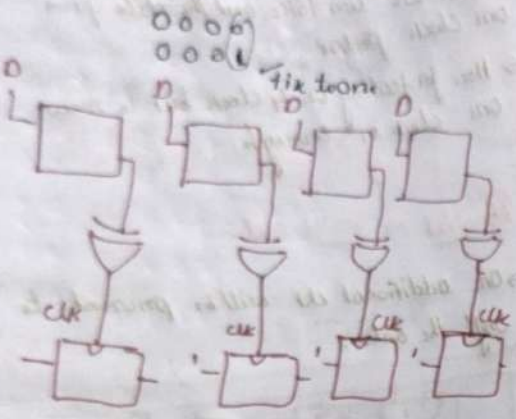


→ Only 1 clk pulse is enough

Universal Shift registers:-

AMD problem

Assume that we are working on a 8 bit random data needs to be transmitted suggest an additional circuit which will observe the obtained outputs and procedure to indicate high signal output, when minimum signal time all bits are toggle.



Counters :-

→ Type of Counter
→ Designing Counters
→ Timing Analysis
→ Clock division (T.F.F)
→
→ Used to count no of clock pulses

Resource Requirement.

- No of bits
  - 3 bit Counter
  - 3 F.F
- No of Count value
  - No of Count & Value
    - 3 bit
    - 3 F.F

→ 16 bit Counter mod value is ?

2^{16}

We require 16 flipflop.

→ Mod 64

→ F.F = 6
→ Bits = 6

} Min Count = 0, Max = 63

→ We can design termination Counter by using reset the value

→ It is also cascaded chain of T.F.
→ No of states is known as mod no

Types of Counter :-

Asynchronous :- Different clock for each flip

- Ripple Counter

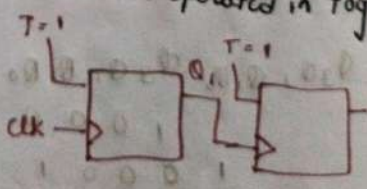
Synchronous :- Same clock to each flip

- MOD Counter
- Ring Counter
- Johnson Counter.

Asynchronous	Synchronous
• Circuit is Simple	• Circuit is Complex
• Tclk >= n tpd	• Tclk >= tpd
• Decoding error due to propagation delay	• No decoding error by of same clock
• State skipping is not allowed / Ring	• It is allowed

Ripple Counter :-

→ FF are operated in toggle mode



→ clk is applied to LSB
→ f = f/2n or f/N
→ For n bit ripple counter, time period of clock

$T_{clk} >= n\, t_{pdff}$

→ $F_{clk} = \frac{1}{n\,t_{pd}}$

t_{pd} = 10ns ; F_{clk} = ? n = 3

$= \frac{1}{(3)(10\times10^{-9})}$

$= \frac{100\times10^{6}}{3000}$

= 33.3 MHz

→ $T_{clk} \geq 3 \times 10$

≥ 30ns

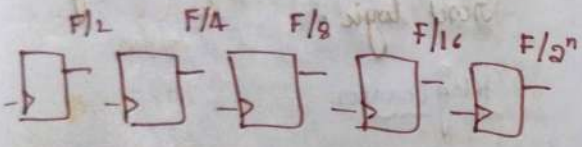
4 bit ripple counter T_{pd} = 50ns

- $T_{clk} \geq n T_{pd}$

= 200ns

- $F_{max} = \frac{1}{n T_{pd}}$

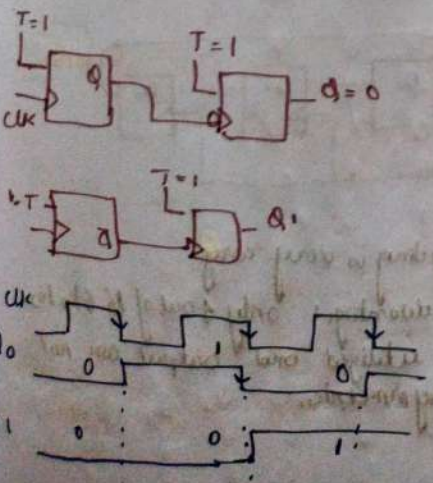
$= \frac{1}{200\times10^{-9}} = 5$ MHz



Possible in Ripple counter for up and down counting:-

① Negative edge trigg F.F & Q'on clock

② Postive edge trigger FF & Q'on clock



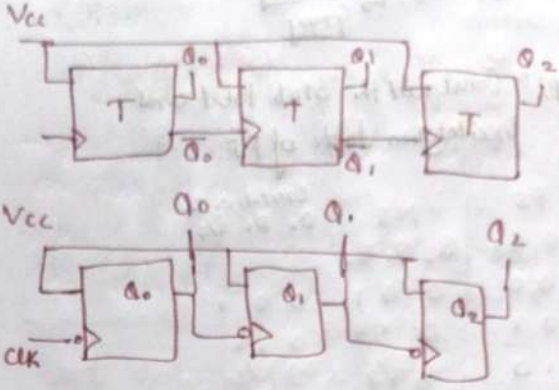
000 } upcount
001

down:-

① Negative Edge trigger FF & Q'on clock

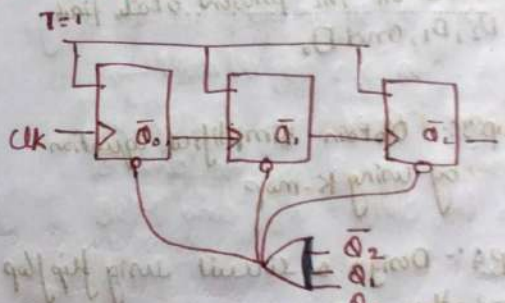
② postive Edge trigger FF Q on clock

Q design a 3 bit up counter using Asynchronous.



Truncation Concept.

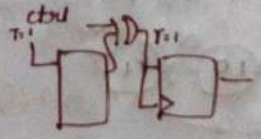
Mod 5 counter should be truncated from 000 to 100.



→ Both up and down counter can be implemented by using exor gate

up/down 1 → up

up/down 0 → down



26/05/2025

Synchronous Counter design :-

Step 1 :- Identify the required number of FF number of inputs and outputs.

mod 16 = 4 bit

4 bit → Binary
→ Gray

Step 2 :- Construct the state table and excitation table of FF. (state diagram)

PS	NS	Excitation D
$Q_2\ Q_1\ Q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	0 1 1	
0 1 1	1 0 0	
1 0 0	1 0 1	
1 0 1	1 1 0	

→ Based on the present state find D_2, D_1, and D_0.

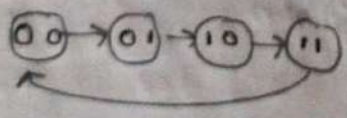
Step 3 :- Obtain simplified equation by using K-map

Step 4 :- Design a circuit using flip/flop and other gate corresponding to minimized expression

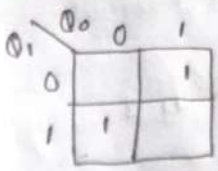
2 bit up counter using DFF.

① FF? 2 FF

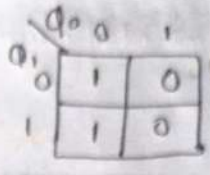
②



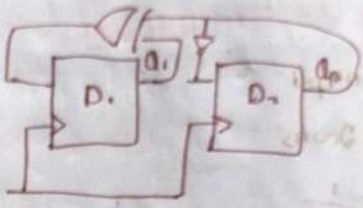
PS	NS	Excitation
$Q_1\ Q_0$	$Q_1\ Q_0$	$D_1\ D_0$
0 0	0 1	0 1
0 1	1 0	1 0
1 0	1 1	1 1
1 1	0 0	0 0



$D_1 = Q_1 \oplus Q_0$



$D_0 = \overline{Q_0}$

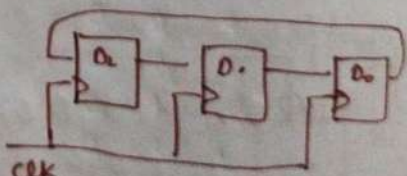


NOTE :- In Asynchronous we can implement just we need to think next logic.

Ring Counter.

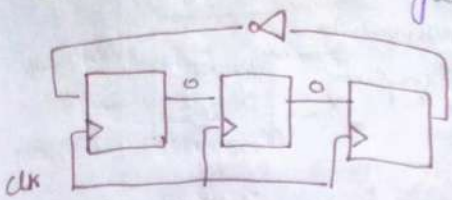
→ Type of shift register (D flip)
→ For n bit ring counter it has n states. (Lim)
→ $2^n - n$ {unused states}

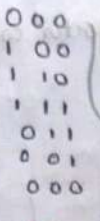
100 → 010 → 001



→ Decoding is very easy
→ disadvantage :- only 4 out of 8 are utilized and output are asymmetric.

Johnson Counter / Twisted ring Counter.





→ For 3 bit Johnson has 6 (2n)
n bit 2n states
$2^n - 2n$ {unused States}

6-bit ring Counter — Twisted.

No of States = 6 ; 12

No of unused = 64-6 = 58 ; 64-12 = 52

States = 100000 →
010000 →
001000
000100
000010
000001

000000
100000
110000
111000
111100
111110
111111
011111
001111
000111
000011
000001
000000

→ No of flips indicates the power consumption in one hot encoding the flip will be less.

Mod 10 counter

No of flip flop

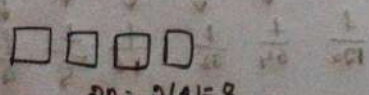
Binary → 4 $\log_2 N$

Gray → 4 $\log_2 N$

on hot → 10 N

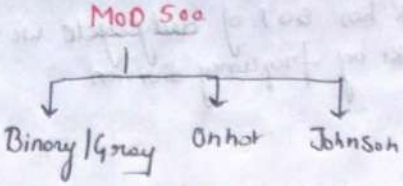
Johnson → 8.5 N/2.

Eg



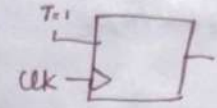
2n = 2(4) = 8
bw hn 10 2(5) = 10

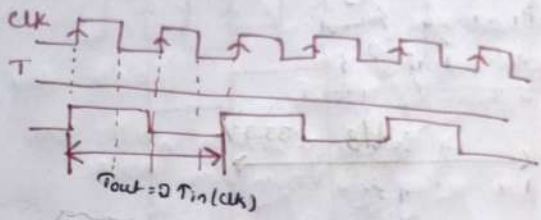
Mod 500



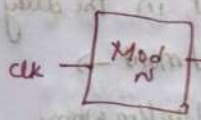
Frequency division:

→ T flipflop is majorly used.

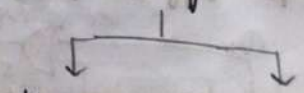


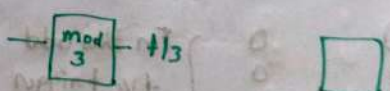


- Tout = 2 × T_{IN}
- fout = f/2



Frequency division

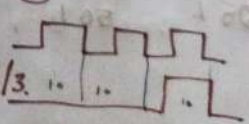




① 00 → 01 → 10

② Q_2 Q_1
0 0
0 1
1 0
0 0
0 1
1 0
MSB

③



Problem here is duty cycle = $\frac{T_{ON}}{T_{OFF}}$

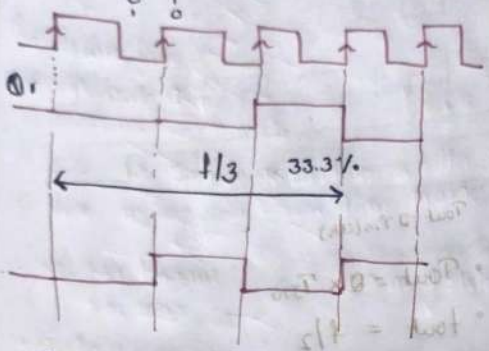
$\frac{10}{30}$

default clock has 50% of duty cycle we need to make by frequency division.

MOD 3:-

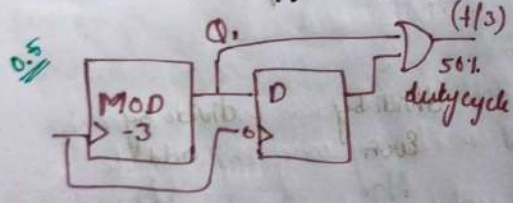
00 → 01 → 10

Q_1	Q_0
0	0
0	1
1	0



Imp → doing the adjustment in the duty cycle for 0.5 adjustment →

Add negedge trigg flipflop whose input is MSB counter and OR in the output of dff and MSB

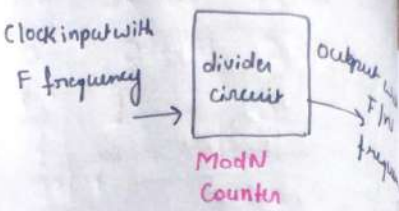


Eg →

0	0
0	0
0	1
1	1

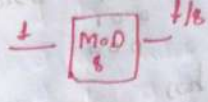
For this add +ve trigger d flip flop

25% 50%.



division factor can be 2, 3, .. N.

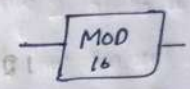
MOD-8 :-



Q_2 Q_1 Q_0

↓ ↓ ↓

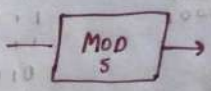
F/8 F/4 F/2



Q_3 Q_2 Q_1 Q_0

↓ ↓ ↓ ↓

F/16 F/8 F/4 F/2



Q_2 Q_1 Q_0

↓ ↓ ↓

F/5 F/4 F/2

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

change

F/120

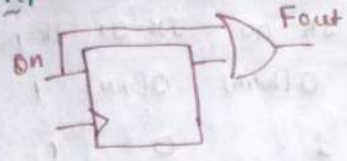
Fix

Q_6 Q_5 Q_4 Q_3 Q_2 Q_1 Q_0

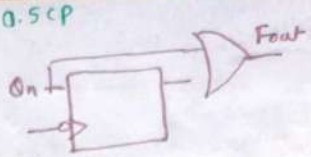
↓ ↓ ↓ ↓ ↓ ↓ ↓

$\frac{f}{120}$ $\frac{f}{64}$ $\frac{f}{32}$ $\frac{f}{16}$ $\frac{f}{8}$ $\frac{f}{4}$ $\frac{f}{2}$

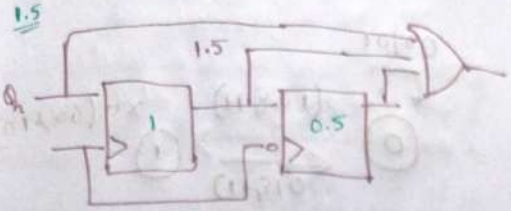
1cp



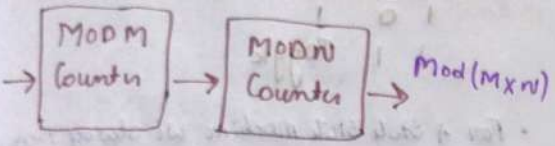
0.5cp

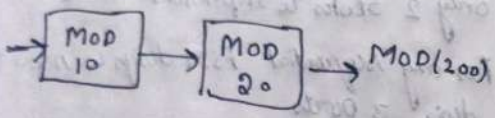


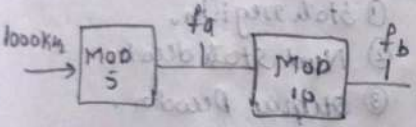
1.5



Cascaded Counter:-





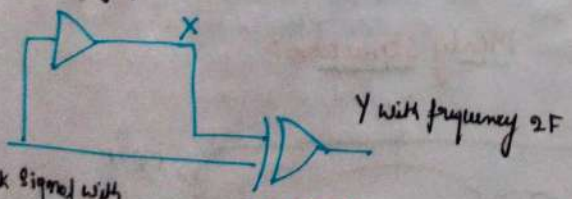


$$f_a = \frac{100KHz}{5} = 20KHz$$

$$f_b = \frac{20KHz}{10} = 2KHz$$

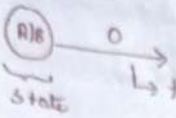
Frequency Multiplier

Buffer with delay of t/4

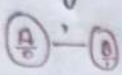


Note :- Waveform

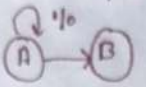
Finate State Machine :-



Mare State
① Output dependson State only



Mealy State
② output depends upon state input



↳ It will give Stable output (additional circuitary) by the output dependson only state.

NOTE:- We cant say which one is better some time mealy one some time moore.

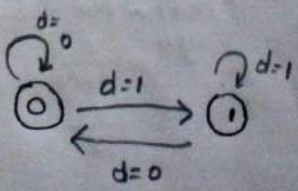
- No of states will be more
- Output depends only on present state
- Output is slow
- Output will be stable

- Less no of states
- output dependson present state and input
- We will get fast result
- output affected by the noise.

D-Flipflop :-

D	On	On+1
0	0	0
0	1	0
1	0	1
1	1	1

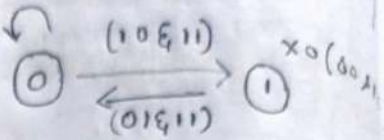
PS	NS D=0	NS D=1
0	0	1
1	0	1



90

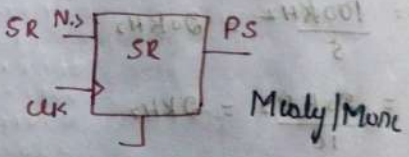
NS

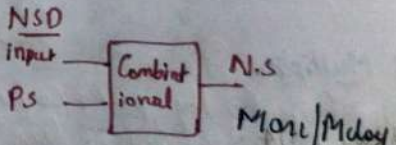
PS	JK=00	JK=01	JK=11
0	0 (latch)	0 (reset)	1
1	1	0	1



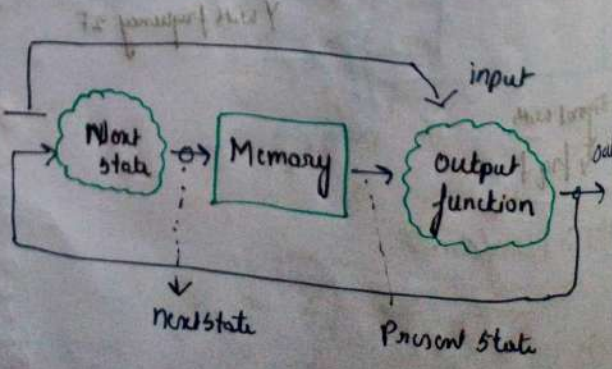
J	K	
0	0	Latch
0	1	0
1	0	1
1	1	Toggle

- For 4 State State machine we always only 2 States to represent.
- For any sequential FSM design we need design 3 parts
  ① State register.
  ② Next State decoder.
  ③ Output Decoder.



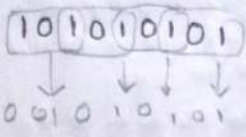


Mealy Structure :-



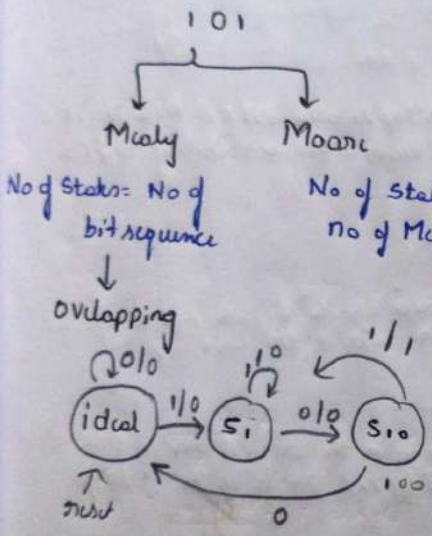
Q. detect the sequence 101

→ overlapping.
→ nonoverlapping.



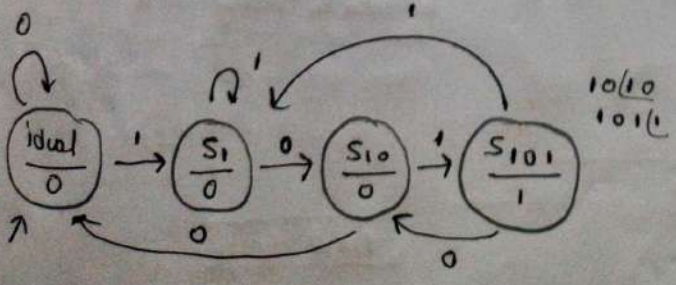
→ non overlapping last one will not be considered if the sequence is detected it will always go to ideal condition.

→ But in overlapping it will check wether it need to go for intermediate state or not.



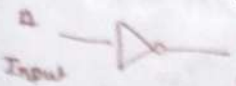
- In non overlaping if the sequence is detected it will go for initial condition
- in overlapping 1̲0̲1 → 10 it will check by removing one by one msB bit

Moore 101



Static timing Analysis:-

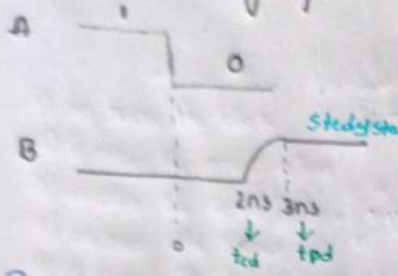
① Contamination delay and Propagation delay



t_c = 2ns

t_{pd} = 3ns

at t=0; A is changed from 1 to 0

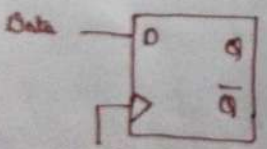


Propagation delay:- The time required to reach the Steady state.

Contamination delay:- B will start going to reach the Steady state. till that

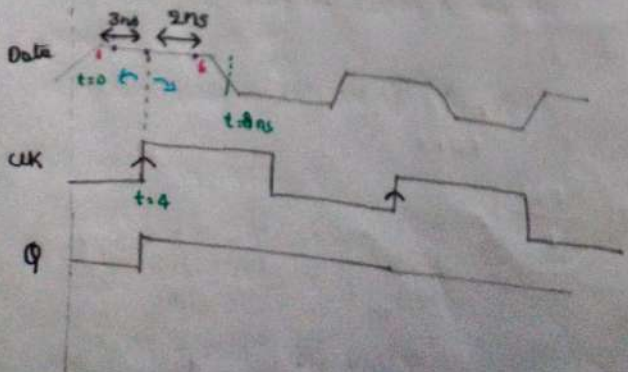
this capacitor will not charge instantly it will take some time to charge.

Setup and Hold time:-



given t_{setup} = 3ns

t_{hold} = 2ns

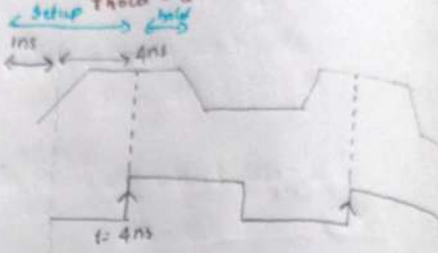


Q is the actual precedence that we will but for sampling we need to see back ahead.

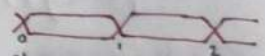
→ So since they given the setup time as 3 ns make it transfer if it is constant for 3ns and also for 2 ns.

→ Mean the data should be constant for 3 ns and also for the next 2 ns., No Voilation.

② If t_{setup} = 5ns.
t_{hold} = 2ns.



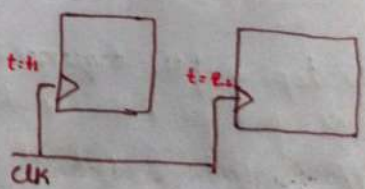
→ Setup voilation but not hold voil it leads to meta stability condit



above graph refers to the data may be one @ ya it will be constant till 0 → 1 → 2.

Clock Skew :-

Difference in the arrival time of sam at different points in the desig



Paracitic refers to the wire that will h resistance and capacitance which will affect the performance when many syste Connected to a single clock.

$\tau = RC$

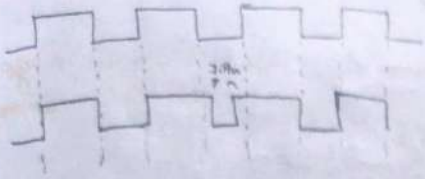
Capacitance & Time

[Assuming $t_2 > t_1$]

$t_{skew} = t_2 - t_1$

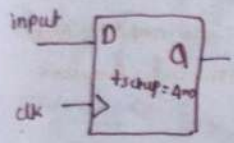
Clock Jitter :- Here the starting and reaching time remain same, but the time period will get altered.

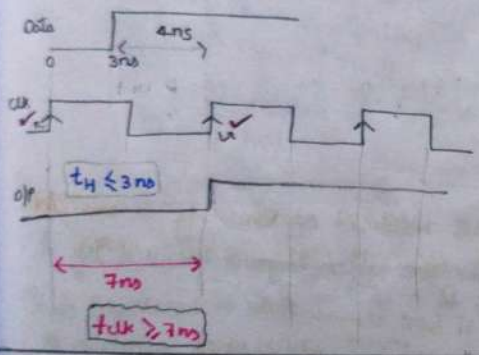
ideal clock :-

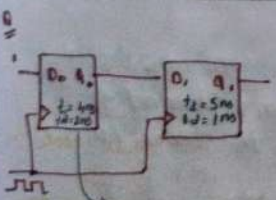


→ clock to Q delay / Propogation delay

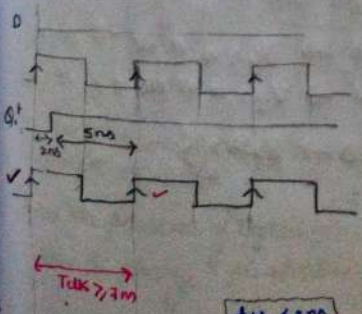
Q Input is changed from 0 to 1 at 3ns, and remain constant for proper operation ; what should be min clock time period? What should be the maximum hold time of flip flop?



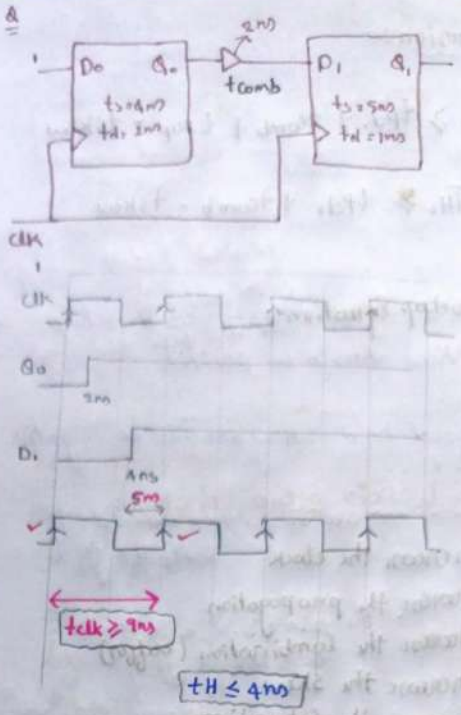




No setup & hold violation.



$t_{H_2} \leq 2ns$

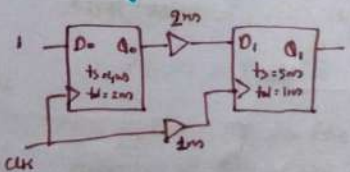


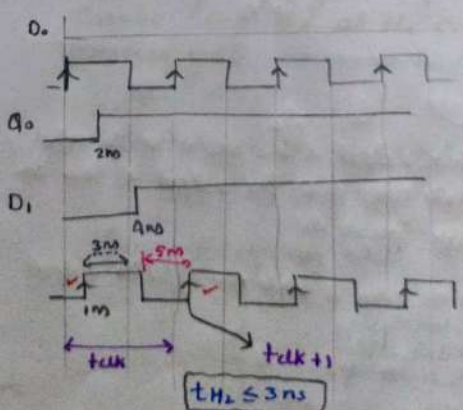
Conclusion :-

$$T_{clk} \geq t_{pd_1} + t_{comb} + t_{setup}$$

$$t_{H_2} \leq t_{p_1} + t_{comb}$$

Introducing clock skew :-





$t_{H_2} \leq 3ns$

$T_{clk} \geq 8ns$

$T_{clk+1} \geq 4 + 5ns$

$\geq 9-1ns$

$T_{clk} \geq 8ns$

Conclusion :-

$T_{clk} \geq t_{pd.} + t_{comb} + t_{sup} - t_{skew}$

$T_{HL} \geq t_{pd.} + t_{comb} - t_{skew}$

For setup Equation :- For good setup the skew should be positive

For good hold skew should be negative

Rules to avoid Setup Voilation :-

1) increase the clock
2) decrese the propogation
3) decrese the combination (buffer)
4) increase the skew
5) decrease the setup time

Rules to avoid the hold time :-

1) Propogation delay should be high
2) Combination delay should be high
3) Thold will be less
4) Tskew should be less

Logic families :-

In all the digital Electronic concepts we use the gates. Now we need to see the internal parts.

Evolution (logic family)

1. RTL
2. DCTL {Direct Coupled transistor logic}
3. I²L {Integrated injection Logic}
4. DTL {Diode transistor logic}
5. HTL {High threshold logic}
6. TTL {Transistor transistor logic}
7. ECL {Emitter Coupled logic}

MOS
- PMOS
- NMOS
- CMOS {Imp}

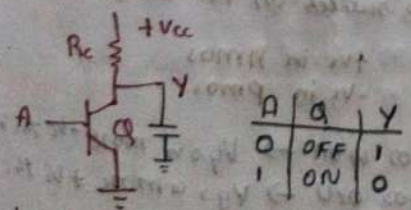
→ Based on the specification we need to choose the logic family.

Specification :-

1) Prop delay
2) Power dissipation
3) Figure of Merit
4) Fanout
5) Noise Margin

Propagation delay :- It is the time required to get a valid output after applying input. It is expressed in ~~milliseconds~~ micro and it should be as minimum as possible.

- It depends on Storage time of Transistor & RC time Constant.

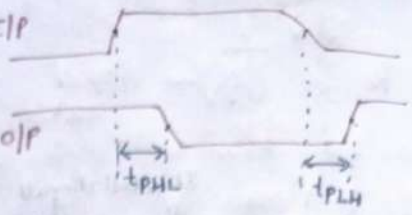


A	Q	Y
0	OFF	1
1	ON	0

↳ it should in cut off & saturation to operate as switch.

Since the presence of Capacitor & Resistor it will take some time to charge & discharge.

How to calculate propogation delay



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

Power dissipation :- It is the average power dissipated by one gate. It is represented by milli watt.

Average power :- When gate voltage is logic 1 and logic 0. is called average power.

$$P_d = V_{cc} I_c \quad (Avg)$$

$$I_{cAvg} = \frac{I_{cH} + I_{cL}}{2}$$

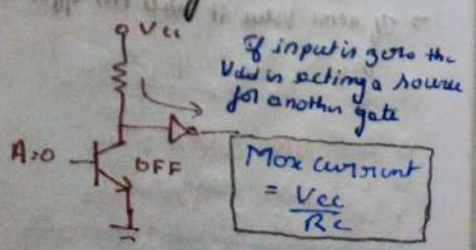
Figure of Merit :- It is a product of t_{pd} & P_d. It should be minimum as possible. It is expressed in Pico joules

$$FOM = t_{pd} \cdot P_d \quad (PJ)$$

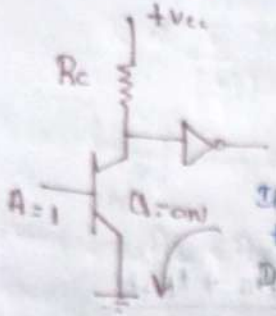
Fan out :- It is the maximum number of logic gate inputs that can be connect at the output of another gate. It must be as high as possible.

It depends on current sourcing & sinking.

Current sourcing :-



Current sinking :



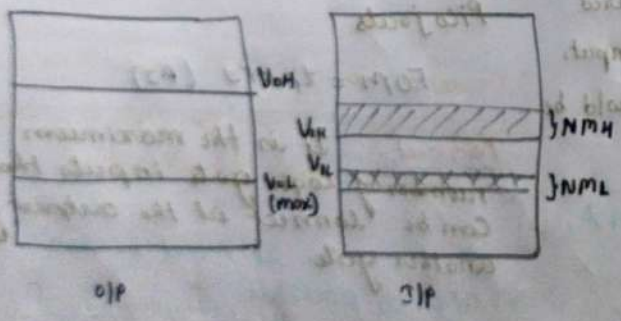
Fanout = min (FoH, FoL)

OH, OL { output low and output high }

FoH = 8
FoL = 5
Fo min {8,5} both should be satisfy ☺

- Final output should be as max as possible

Noise Margin :- how much noise you can add to your system, so still the system must be safe. Which is added at the input and that must not affect the output.



VoL : when output is low and it is connected to next gate, mean the output is less the maximum possibility

→ If some value it will not affect the output

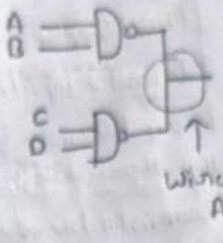
$V_{OH} > V_{IH} > V_{IL} > V_{OL}$

$NM_H = V_{OH} - V_{IH}$

$NM_L = V_{IL} - V_{OL}$

$N.M = \min(NM_H, NM_L)$

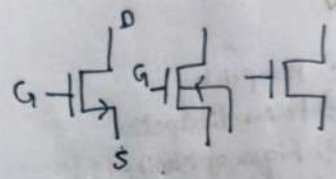
Wired Logic :- When two gate outputs are connected with a wire the function will acts like with gate.



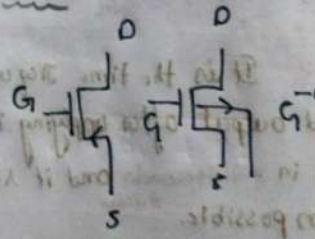
Except in ECL in all other logic families it is wired AND in ECL

In CMOS we avoid wired logic

NMOS :-



PMOS :-



Threshold Voltage :- minimum voltage required to form a channel or to switch on the transistor

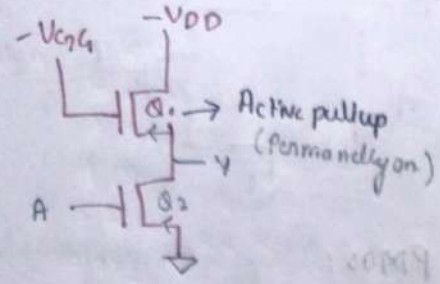
+Vt in nmos.
-Vt in pmos.

Pmos is on → V_{gs} is more $-V_{th}$
Nmos on → V_{gs} is more $+V_{th}$

$V_{gs} = 0$ → Both off.

Pmos:

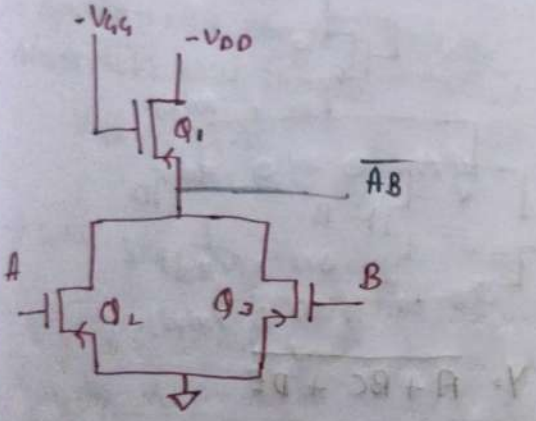
Logic 0 = -VDD
Logic 1 = 0V
} +Ve logic.



A	Q2	Y
0 (-VDD)	ON	1 (0V)
1 (0V)	OFF	0 (VDD)

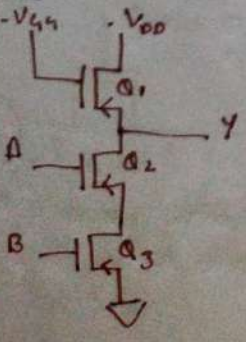
Inverter

NAND:-



A B	Q2 Q3	Y
0 0	ON ON	(0V) 1
0 1	ON OFF	(0V) 1
1 0	OFF ON	(0V) 1
1 1	OFF OFF	(-VDD) 0

NOR:-

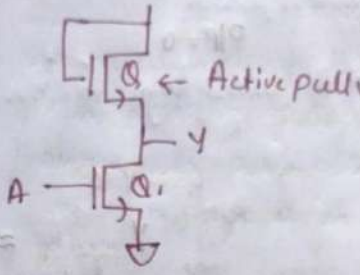


A B	Q2 Q3	Y
0 0	ON ON	(0V) 1
0 1	ON OFF	(-VDD) 0
1 0	OFF ON	(-VDD) 0
1 1	OFF OFF	-(VDD) 0

Specification:-

t_{pd} = 300ns
Pd = 0.2 MW
FoM = 60 PJ
NM = 1.5 volts
Fanout = 15

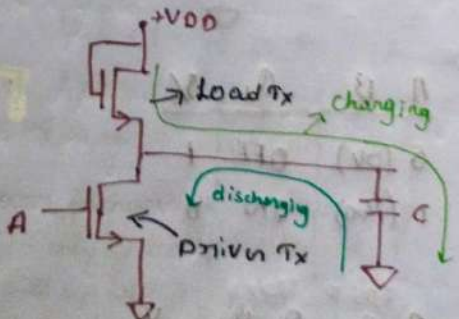
Nmos:- logic 0 :- 0V
logic 1 = +VDD
} +Ve logic.



A	Q1	Y
0 (0V)	OFF	(VDD) 1
1 (VDD)	ON	(0V) 0

In Nmos Inverter charging time is always greater than discharging time. which of the following is the reason for it?

a) Aspect ratio of load Tx more than driver Tx
b) Aspect ratio of Driven Tx more than load Tx
c) Charging takes more time than disch
d) None



$T_C = R_L \cdot C$
$T_D = R_D \cdot C$

R_L = ON resistance of Q_L
R_D = ON resistance of Q_D

Note ★★★

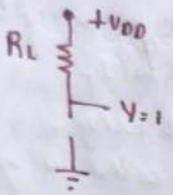
In Nmos design $R_L > R_D$

$R \propto \frac{1}{(\frac{W}{L})}$ → A spect ratio

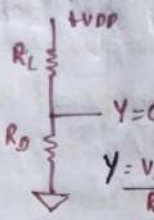
$A_L < A_D$

Analysis:

O/P = 1



O/P = 0

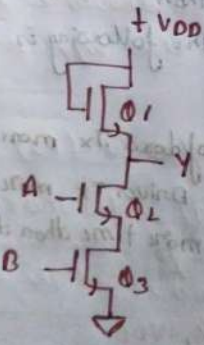


$Y = \frac{V_{DD} R_D}{R_L + R_D} \approx 0$

In O/P is 0 condition if 1V is applied in output only 0.5 will be divided.

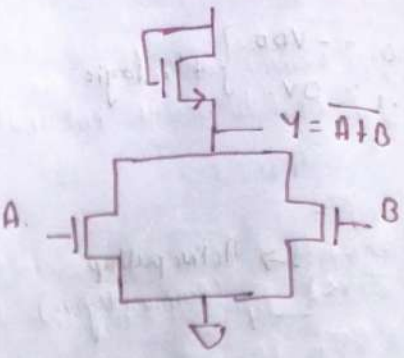
So to avoid this $R_D << R_L$

NAND:



A	B	Q_1	Q_2
0 (0V)		OFF	1
1 (V_{DD})		ON	0

A	B	Q_1	Q_2	Y
0	0	OFF	OFF	1
0	1	OFF	ON	1
1	0	ON	OFF	1
1	1	ON	ON	0

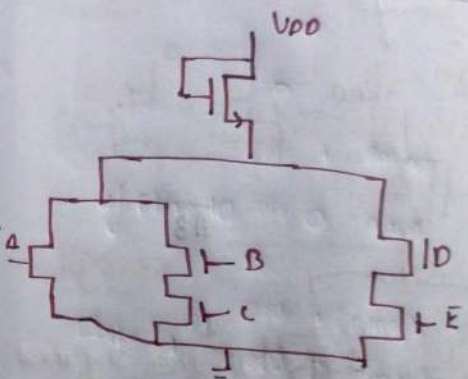


PMOS:

+ve → Series → NOR
Parallel → NAND

NMOS

+ve → Series → NAND
Parallel → NOR.



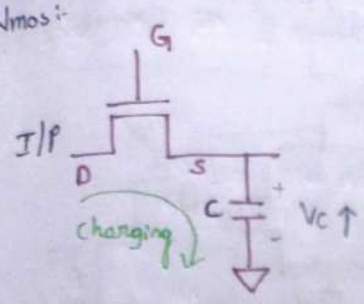
$Y = \overline{A + BC + DE}$

Specifications:-

t_{pd} : 250ns
Pd : 1mW
FOM : 250PJ
N.M : 0.15
Fanout = 10.

Pass transistor logic :-

Nmos :-



→ input will come when gate = 1.

→ if I/P = 1 G = 1 (VDD)

When $V_{gs} > V_T$ = ON.

$V_g = V_{DD}$, $V_s = 0$ } $V_{gs} = V_{DD}$

ON.

So capacitor will charge, voltage across the Vc will increase

$V_{GS} = V_G - V_S$

$= V_{DD} - V_c > V_T$

→ When $V_{DD} = V_c$ → 0 OFF

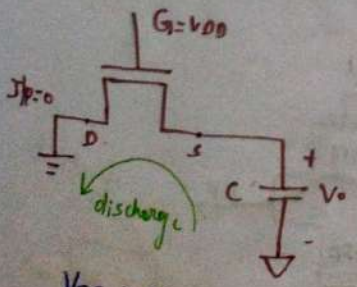
$= V_{DD} - V_T > V_c$

$V_o = V_c < V_{DD} - V_T$

↑

Weak - 1.

So if i pass Vdd input it will not able to give the output Vdd in.



$V_{gs} > V_T$

$V_g - V_s > V_T$

$V_{DD} - V_o > V_T$ → $V_{DD} - 0 > V_T$ → ON.

So the transitor will on till it is complitly discharge Strong 0.

Conclusion :-

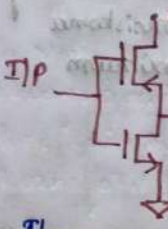
Nmos is Strong in '0'
Weak in '1'

Pmos is Strong in '1'
Weak in '0'.

CMOS :-

→ In cmos we will always avoid floating input. because its i/p Capacitance is very high. It can Easly affected by External magnetic field

Inverter :-



dogic - 0, 0
dogic - 1, 3 - 18 V.

I/P = 0 $V_{GS} = 0 - V_{DD}$ (Pmos) ✓

$V_{GS} = 0 - 0$ (nmos) ×

O/P = 1

I/P = 1 $V_{GS} = 1 - V_{DD} = 0$ (Pmos) ×

$V_{GS} = 1 - 0 = 1$ (nmos) ✓

O/P = 0

Advantage :- Compare with the inverter of the nand & only by nmos & pmos the active pull up will goes none of the current complex due to R_L.

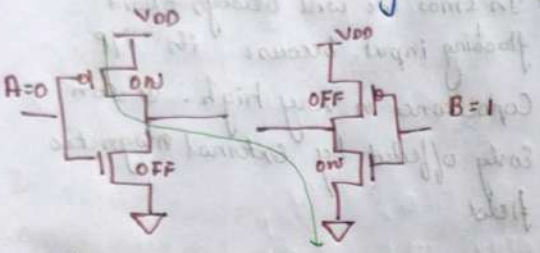
Power Efficient, delay will be Stedruced. Static power dissipation is very low in cmos.

at any given time only one of them is on static power dissipation is low.

Dynamic power consumption:-

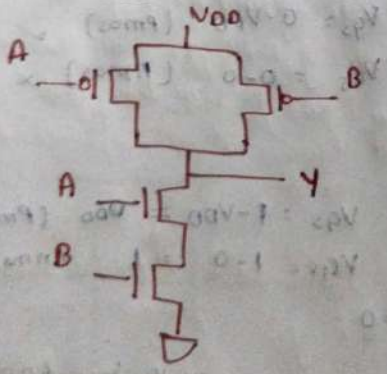
It is a type of power consumption in which when the transition of pmos & nmos from Low to high and from high to low. At some point both of them are on will leads to maximum power consumption is called dynamic power.

NOTE:- Avoid the wired logic in CMOS



if they connected means we are providing the wired logic with least resistance then it will consume maximum current.

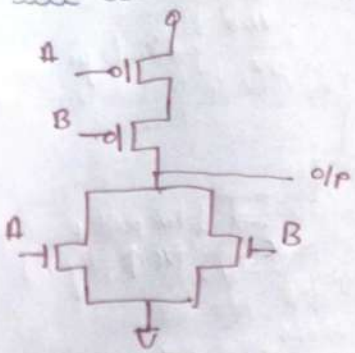
CMOS NAND



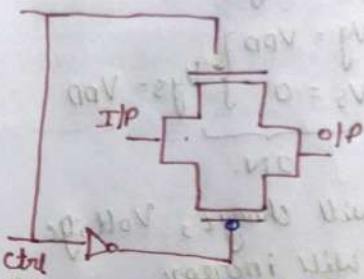
Specifications

tpd = 70ns
Pd = 0.01 mW
Fom = 0.7PJ
N.M = 0.35 VDD
Fanout = 50.

CMOS NOR:-



Pass transistors:-



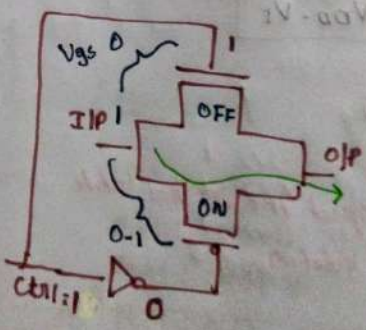
Case-1 OFF 1 (nmos)

if Ctrl = 0 OFF 0 (Pmos)

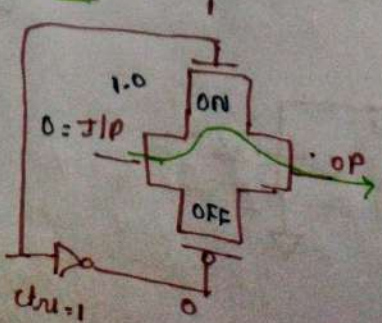
Case 2

Ctrl = 1 ON 1 (nmos)

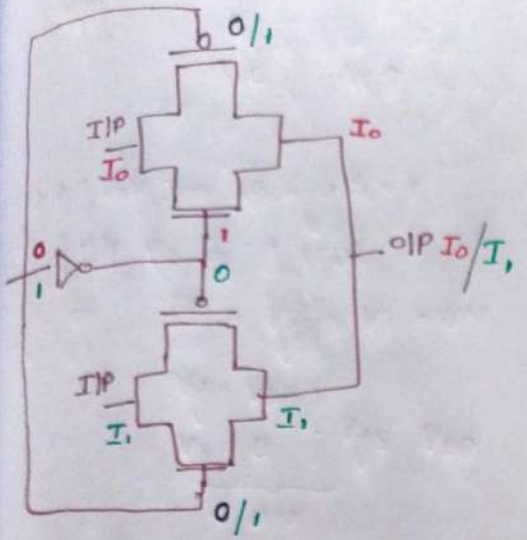
ON 0 (Pmos)



Case 3



Whenever we want to implement any ckt with less no of gates we can use this transmission gates.



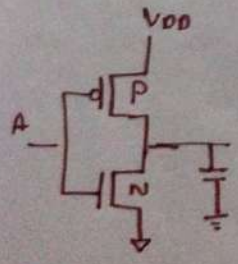
2:1 Mux = 4 + 2 = 6 Transistor

For normal implimentation.

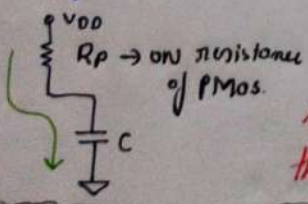
$Y = \bar{S} I_0 + S I_1$

2 AND = 2 × 6 = 12
1 OR = 1 × 6 = 6
1 INV 1 × 2 = 2
20 Trans

Inverter:-

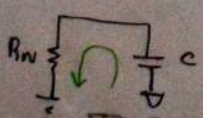


Charging



$T_c = R_P C$

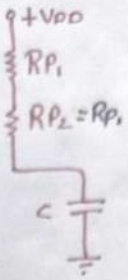
discharge



$T_D = R_N C$

For Nof Gate:-

Charging:-



$T_c = 2R_{P_1} \cdot C$

in worst case $\leq R_P C$.

$2R_{P_1} \leq R_P$

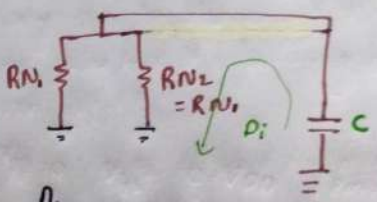
$R_{P_1} \leq \frac{R_P}{2}$

Pmos → R half → width must be double.

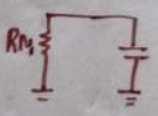
discharging:-

Both N_1 N_2 → ON. (AB)

Any one of N_1, N_2 → ON.



Any one ON.



$T_D = R_{N_1} \cdot C$

Worst $R_{N_1} C \leq R_N C$

→ No change

Hence no change in nmos but change the width in pmos

Problem:- Series → Multiply, Parall → add. } nmos

Series → add, Parallel → mul } Pmos

→ Add bar at the last.