Digital logic Design

UNIT - I

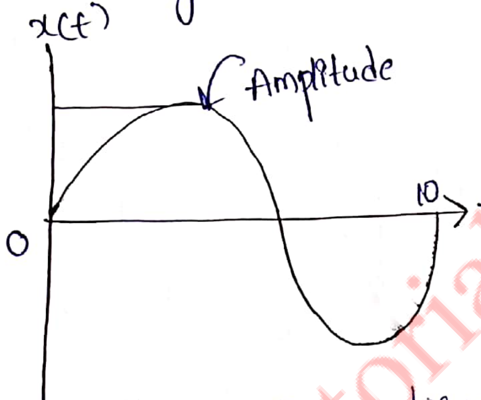
Digital system and binary numbers

Introduction:- In a real time basic signal is analog signal analog signal is converted into digital signal

Analog signal:-

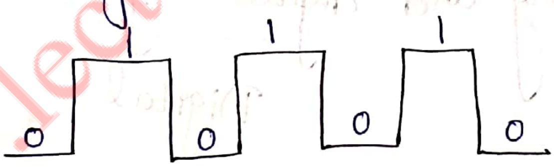
* It is a continous time signal with respect to the amplitude

Ex:- sinusoidal signal



Digital signal:- It is a discreate time signal. It's binary digits.

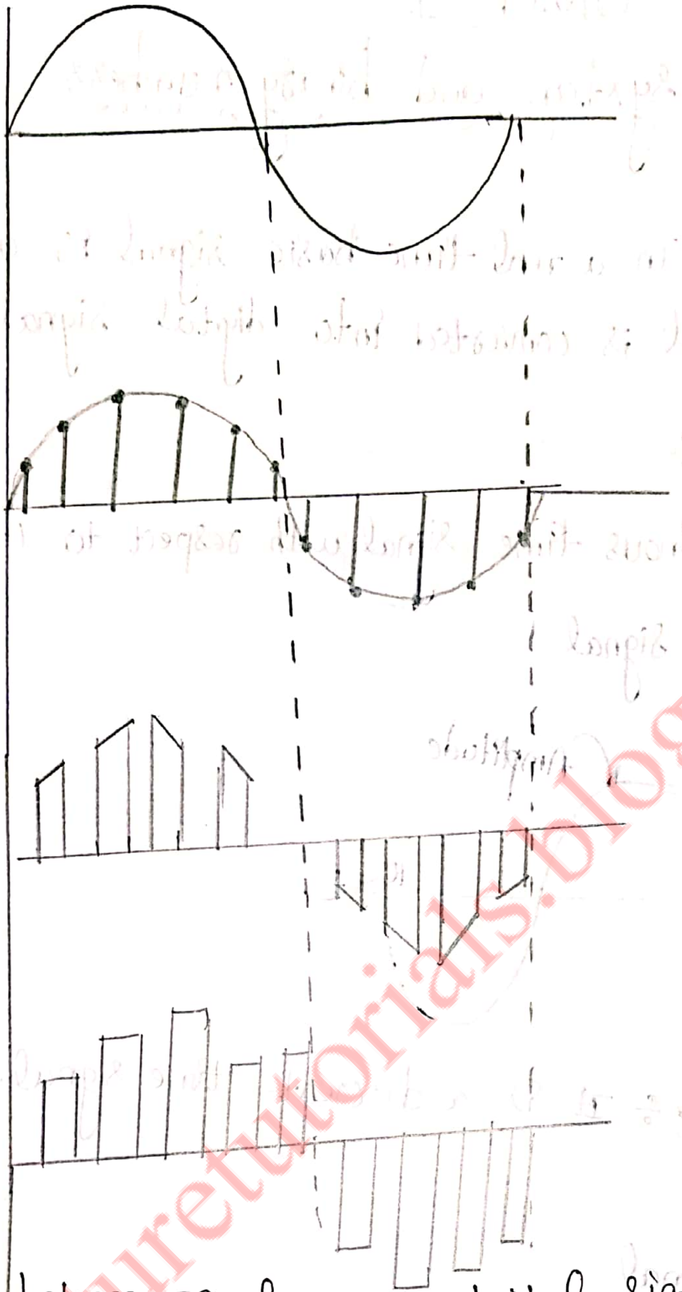
Ex:- pulse signal



Convert analog to digital signal:-

Basically analog to digital signal is conversion in a three steps.

1. sampleing
2. quatisation
3. coder



* Difference between analog and digital signal

Analog	Digital
* low accuracy	* more accuracy
* Procesur is slow	* procesure is large
* size is large	* size is small
* Expensive (or) highcost	* lowcost
* It is not upgraded	* It is easily upgard device
* more noise occur	* low noise occur

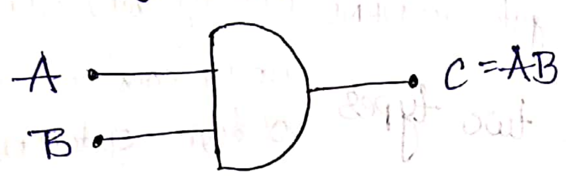
Logic gates:

Basically logic gates are three types. They are

1) AND gate
2) OR gate
3) NOT gate

AND gate:

The AND gate symbol is given below,

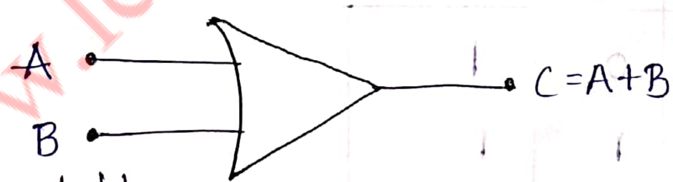


Truth table:

A	B	C=AB
0	0	0
0	1	0
1	0	0
1	1	1

OR gate:

The OR gata symbol is given below

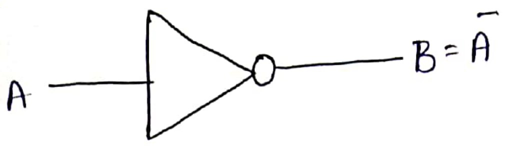


Truth table:

A	B	C = A+B
0	0	0
1	0	1
0	1	1
1	1	1

NOT gate:

NOT gate symbol is given below



truth table:

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

Universal gates:

universal gates are NAND, NOR By using NAND and NOR we can construct any type of logic gate. NAND and NOR easily constructed transistor circuits.

The universal gates are two types

1. NAND
2. NOR

1. NAND gate:

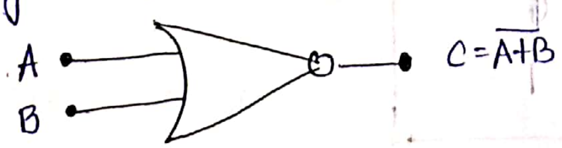
The NAND gate symbol is given below

Truth table:

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

NOR gate:

NOR gate symbol is given below



truth table:

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

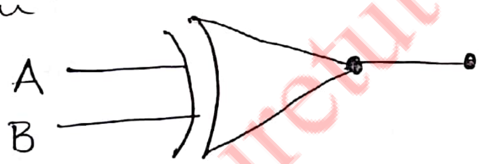
Design: Designd means to reduce the logic gates to (or) generate logic gates (or) different components is connected to the sequentional manner (or) sequence manner

Exclusive gates:

X-OR gate:

X-OR gate symbol is given below

Symbol:



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

$C = A \oplus B$

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

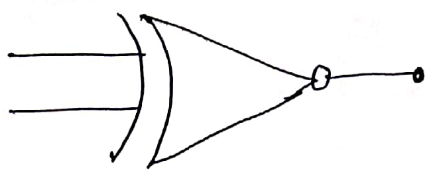
Truth table:

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

X-NOR gate:

X-NOR gate symbol is given below

Symbol:



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

$C = A \odot B$

truth table:

A	B	$C = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Feature scope of digital logic design:

The digital logic design mainly use is a real time

→ Tele communication
→ Internet of thing (IOT)
→ Cloud computing
→ body area networking
→ Information technology
→ microweb point
→ satilight & optical fibre technology
→ Instrumentation
→ Remote searching
→ signal processing
→ Image processing

UNIT-I

Digital System and Binary Numbers

Digital system or number system:

→ The basic ~~digital~~ decimal number system with its 10 digits. they are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

→ In digital communication operate with binary numbers. which is use only the digits 0s and 1s

→ The number systems are basically fourtypes they are:

i) Binary number system (2)

ii) octal number system (8)

iii) decimal number system (10)

iv) Hexadecimal number system (16)

(i) Binary number System(2):

→ The binary number system are used in digits 0s and 1s

→ The binary Position Values as a power of '2' is represented by

. . . . .	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	. . . . . .

MSB – Most significance Bit

LSB – Least Significance Bit

Ex: Represent binary number 1101.101 in power of '2'. find its decimal Equalent?

2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
1	1	0	1	.	1	0	1

$$N = 1\times2^3 + 1\times2^3 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{3}$$

$$N = 8+4+0+1+\frac{1}{2}+0+1/8$$ $[\because a^0 = 1,\ 2^0 = 1]$

$$N = (13.625)_{10}$$

Ex:-2 $(1001101.10111)_2$

2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
1	0	0	1	1	0	1	.	1	0	1	1	1

$$N = 1\times2^6 + 0\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 +$$
$$1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} \times 1\times2^{-5}$$

$$N = 64 + 8 + 4 + 2 + 0.5 + 0.125 + 0.0625 + 0.03125$$

$$N = (78.71875)_{10}$$

* The octal number system uses first Eight values are decimal number system. they are 0,1,2,3,4,5,6,7. It's base is '8'. octal position value as a power of '8' represented is given by

...	8^3	8^2	8^1	8^0	.	8^{-1}	8^{-2}	8^{-3}

MSB LSB

* Represent octal number 567 in power of 8 and find its decimal Equalent.

$[8^0 = 1]$

8^2	8^1	8^0
5	6	7

$$N = 5\times8^2 + 6\times8^1 + 7\times8^0$$

$$= 5\times64 + 6\times8 + 7\times1$$

$$N = (375)_{10}$$

* $(4271.635)_8$

8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	8^{-3}
4	2	7	1	.	6	3	5

$$N = 1\times8^0 + 7\times8^1 + 2\times8^2 + 4\times8^3 + 6\times8^{-1} + 3\times8^{-2} + 5\times8^{-3}$$

$$= 1 + 56 + 128 + 2048 + 6.125 + 0.1875 + 0.00976$$

$$= (2239.80)_{10}$$

* $(64562.1057)_8$

8^4	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	8^{-3}	8^{-4}
6	4	5	6	2	.	1	0	5	7

$$N = 6\times8^4 + 4\times8^3 + 5\times8^2 + 6\times8^1 + 2\times8^0 + 1\times8^{-1} + 0\times8^{-2} + 5\times8^{-3} + 7\times8^{-4}$$

$$= 24576 + 2048 + 320 + 48 + 2 + 0.125 + 0.00968 + 0.0017089$$

$$N = (26994.13639)_{10}$$

Decimal number system:-

→ In Decimal number system we can Express any decimal number in units, tens, Hundreds and thuasands.......

→ In Decimal number system the numbers are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

→ The Decimal position Values as a power of 10 is represented by

....	10^2	10^1	10^0	.	10^{-1}	10^{-2}	10^{-3}	.....

MSB LSB

Ex:- 6587.6

10^3	10^2	10^1	10^0		10^{-1}
6	5	8	7	.	6

$$N = 6\times10^3 + 5\times10^2 + 8\times10^1 + 7\times10^0 + 6\times10^{-1}$$

$$= 6000 + 500 + 80 + 7 + 0.6$$

$$= 6587.6$$

Hexa Decimal number system:-

→ The Hexa Decimal number system as a base of 16. It having 16 digits they are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
(A=10, B=11, C=12, D=13, E=14, F=15)

→ The Hexa decimal position Values as a power of 16 is represented by

....	16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}	....

MSB — LSB

Ex:- Represents Hexadecimal number 3FD in power of 16 and find Decimal Equilent.

16^2	16^1	16^0
3	F	D

$$N = 3\times16^2 + F\times16^1 + D\times16^0$$

$$= 3\times256 + 15\times16 + 13\times1$$

$$= (1021)_{10}$$

Ex:- FDE42A.1DB9

16^5	16^4	16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}	16^{-4}
F	D	E	4	2	A	.	1	D	B	9

$$N = 15\times16^5 + 13\times16^4 + 14\times16^3 + 4\times16^2 + 2\times16^1 + 10\times16^0 + 1\times16^{-1} + 13\times16^{-2} + 11\times16^{-3} + 9\times16^{-4}$$

$$= 16639018 + 851968 + 57344 + 1024 + 32 + 10 + 0.0625 + 0.0507 + 0.0026 + 0.00013732$$

$$= (17549396.12)_{10}$$

Relation b/w binary, decimal, Hex decimal in below tables.

Decimal	Binary 8421	Hexa Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3

Decimal	Binary 8421	HexaDecimal
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Counting in Radix (Base) r :-

Radix (Base) r	Character in set
r=2	0, 1
r=8	0,1,2,3,4,5,6,7
r=10	0,1,2,3,4,5,6,7,8,9
r=16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F

Ex:- To find Decimal values of 0 to 9 by using radix 5

Decimal	Radix		
0	00	$0\times5^1+0\times5^0$	= 0
1	01	$0\times5^1+1\times5^0$	= 1
2	02	$0\times5^1+2\times5^0$	= 2
3	03	$0\times5^1+3\times5^0$	= 3
4	04	$0\times5^1+4\times5^0$	= 4
5	10	$1\times5^1+0\times5^0$	= 5
6	11	$1\times5^1+1\times5^0$	= 6
7	12	$1\times5^1+2\times5^0$	= 7
8	13	$1\times5^1+3\times5^0$	= 8
9	14	$1\times5^1+4\times5^0$	= 9

* find the decimal Equalent of 231.23 base 4.

4^2	4^1	4^0		4^{-1}	4^{-2}
2	3	1	.	2	3

$$N = 2\times16+3\times4+1\times4+2\times4^{-1}+3\times4^{-2}$$

$$= 32+12+4+0.5+0.1875$$

$$= (48.6875)_{10}$$

Number base Conversion:-

→ The decimal, binary, octal and hexadecimal table is given below

Decimal	Binary 8421	octal	Hexa
0	0000	0	0
1	0001	1	1
2	0010	2	2

Decimal	Binary 8421	Octal	Hexadecimal
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversions:

→ Basically binary, octal, Hexdecimal conversions or 6 types

1. Binary to octal
2. octal to Binary
3. Binary to Hexa
4. Hexa to Binary
5. octal to Hexa
6. Hexa to octal

1. Binary to Octal:

→ The binary numbers are 0 and 1. The octal numbers are 0 to 7 the octal number table is given below

Decimal	Binary-421
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Ex: $(11010101 0 . 101010)_2$ to convert octal

Sol:

$$\underset{6}{110}\ \underset{5}{101}\ \underset{2}{010} \ . \ \underset{5}{101}\ \underset{2}{010}$$

$\therefore (652.52)_8$

Ex: $(100101010.0101)_2$

$$\underset{4}{100}\ \underset{5}{101}\ \underset{2}{010} . \underset{5}{101}$$

010 010 101 . 010 100

$(452.5)_8$

Ex:

(ii) Octal to Binary Conversion:

Ex: $(643.27)_8$

$$\begin{array}{cccccc} 6 & 4 & 3 & . & 2 & 7 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 110 & 100 & 011 & & 010 & 111 \end{array}$$

$\therefore (110\,100\,011 . 010\,111)_2$

(iii) Binary to Hexa conversion:

Ex: $(1101100010011011)_2$

$$\begin{array}{cccc} 1101 & 1000 & 1001 & 1011 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ D & 8 & 9 & B \end{array}$$

$(D89B)_{16}$

$\therefore (D89B)_H$

(iv) Hexa to Binary Conversion:

Ex: Convert $(3FD)_H$

$$\begin{array}{ccc} 3 & F & D \\ \downarrow & \downarrow & \downarrow \\ 0011 & 1111 & 1101 \end{array}$$

$(0011\ 1111\ 1101)_2$

Ex: $(F9B0.1D8)_H$

$$\begin{array}{cccccccc} F & 9 & B & 0 & . & 1 & \overset{13}{D} & 8 \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 1111 & 1001 & 1011 & 0000 & & 0001 & 1101 & 1000 \end{array}$$

$(1111\ 1001\ 1011\ 0000\ 0001\ 1101\ 1000)_2$

Ex÷ convert (5A89·B4)H to binary

5	A	8	9	.	B	4
↓	↓	↓	↓		↓	↓
0101	1010	1000	1001		1011	0100

$(0101\ 1010\ 1000\ 1001\ 1011\ 0100)_2$

v) Octal to Hexadecimal÷

→ The easyest way to convert octal number to Hexadecimal number in two steps.

i) convert octal number to binary form
ii) convert binary form to Hexa decimal

convert binary form to Hexa decimal equvalent.

Ex÷ convert $(615)_8$ to its Hexa decimal

$(615)_8$

(i) octal – binary

6 1 5
↓ ↓ ↓
$(110\ 001\ 101)_2$

(ii) Binary – Hexa (110001101)

0001	1000	1101
↓	↓	↓
1	8	D

∴ $(18D)_H$

Ex÷ $(7523.426)_8$

(i) octal – binary

7	5	2	3	.	4	2	6
↓	↓	↓	↓		↓	↓	↓
111	101	010	011		100	010	110

111101010011100010110

ii) Binary – Hexa

0001	1110	1010	0111	0001	0110
↓	↓	↓	↓	↓	↓
1	E	A	7	1	6

∴ $(1EA716)_H$

vi) Hexa to octal conversion:-

→ The Hexa decimal to octal conversion two steps is there

i) Convert Hexa to binary

ii) Convert binary - octal

Ex:- convert $(25B)_H$ to its octal equalent

$(25B)_H$

(i) Hexa to binary

2 5 B
↓ ↓ ↓
0010 0101 1011

(ii) binary - octal

0010 0101 1011

001 001 011 011
↓ ↓ ↓ ↓
1 1 3 3

∴ $(1133)_8$

Ex:- ② $(9DF6 \cdot C83)_H$

(i) Hexa to binary

9 D F 6 . C 8 3 → 0011
↓ ↓ ↓ ↓ ↓ ↓
1001 1101 1111 0110 1100 1000

(ii) binary - octal

1001110111110110110001000

010 011 011 111 011 011 001 000
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
2 3 3 7 3 3 1 0

∴ $(23373310)_8$

* Convert any Radix to Decimal :-

In general the number can be represented as

$$N = A_{n-1} r^{n-1} + A_{n-2} r^{n-2} + \dots + A_1 r^1 + A_0 r^0 + A_{-1} r^{-1} + A_{-2} r^{-2} + \dots + C_m r^{-m}$$

where N = number in decimal

A = digit

r = radix (or) base of a number system

n = The number of digits in the integer portion of number

m = The no. of digits in the fractional portion of number

① $(1101.1)_2$ to convert Decimal

② $(475.25)_8$ to convert Decimal

③ $(9B2.1A)_H$ to convert Decimal

④ $(3102.12)_4$ to Decimal

⑤ $(614.15)_7$ to Decimal

① $(1101.1)_2$

2^3	2^2	2^1	2^0		2^{-1}
1	1	0	1	.	1

$$N = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times{}^1\!/_2$$

$$= (13.5)_{10}$$

② $(475.25)_8$

8^2	8^1	8^0		8^{-1}	8^{-2}
4	7	5	.	2	5

$N = 4\times 8^2 + 7\times 8^1 + 5\times 8^0 + 2\times 8^{-1} + 5\times 8^{-2}$

$= (317.328)_8$

③ $(9B2.1A)_H$

16^2	16^1	16^0		16^{-1}	16^{-2}
9	B	2	.	1	A

$N = 9\times 16^2 + B\times 16^1 + 2\times 16^0 + 1\times 16^{-1} + A\times 16^{-2}$

$= (2482.101563)_{10}$

④ $(3102.12)_4$

4^3	4^2	4^1	4^0		4^{-1}	4^{-2}
3	1	0	2	.	1	2

$N = 3\times 4^3 + 1\times 4^2 + 2\times 1 + 1\times 1/4 + 2\times 1/16$

$= (210.375)_{10}$

⑤ $(614.15)_7$

7^2	7^1	7^0		7^{-1}	7^{-2}
6	1	4	.	1	5

$N = 6\times 7^2 + 1\times 7^1 + 4\times 7^0 + 1\times 1/7 + 5\times 1/49$

$N = (301.244)_{10}$

Convertion of Decimal numbers to radix numbers

→ Basically the conversions of radix numbers are two type

1. Succesive division for Integer part Convertion
2. Succesive multiplication for fractional part convertion

1. Successive division for integer part Convertion:

→ In this method we repeatedly divided the Integer part of the decimal number by "r". until ~~Coeff~~ qotient is '0'.

→ The remainder of each division becomes the numareical in the new radix.

→ The reminders are taken in the reverse order to a new radix number.

→ This means that first remainder is the least significant bit in the new radix and the last significant is the most significant bit in the new radix number.

Ex: Convert decimal number 37 to in binary Equalent

$(37)_{10}$

2	37	
2	18	1
2	9	0
2	4	1
2	2	0
	1	0

↑ LSB

MSB

$\therefore (100101)_2$

$\therefore (37)_{10} = (100101)_2$

Verification:

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

1 0 0 1 0 1

$= 1\times2^5 + 1\times2^2 + 1\times2^0$

$= 32 + 4 + 1\times1$

$= 37$

HW

1) (625) to Binary
2) (739) to "
3) (523) to octal
4) (649) "

Ex÷ convert decimal to octal

(214)

$$\begin{array}{r|l} 8 & 214 \\ \hline 8 & 26 - 6 \\ \hline & 3 - 2 \end{array}$$

$\therefore (326)_8$

Ex÷ Convert decimal number (3509) to Hexa decimal Equalent

$$\begin{array}{r|l} 16 & 3509 \\ \hline 16 & 219 - 5 \\ \hline 16 & 13 - 11 \\ \hline \end{array}$$

$\therefore$ (13 11 5)

$\therefore (DB5)_{16}$

Ex÷ convert decimal to Hexa (4559)

Ex÷ Convert decimal to Hexa (5320)

(i) (4559)

$$\begin{array}{r|l} 16 & 4559 \\ \hline 16 & 284 - 15 \\ \hline 16 & 17 - 12 \\ \hline 16 & 1 - 1 \\ \hline \end{array}$$

$\therefore (15\ 12\ 1\ 1)_{16}$

$\therefore (4559) = (FC11)_H$

(ii) (5320)

$$\begin{array}{r|l} 16 & 5320 \\ \hline 16 & 332 - 8 \\ \hline 16 & 20 - 12 \\ \hline 16 & 1 - 4 \\ \hline \end{array}$$

$\therefore$ (5320) = (8 12 4 1)

$\therefore (8C41)_H$

2. Successive multiplication for fractional part conversion

→ convert 0.8125 to Binary

fractional	Radix	Result		Recorded carry
0.8125	x 2	= 1.625	= 0.625	with carry 1
0.625	x 2	= 1.25	= 0.25	with carry 1
0.25	x 2	= 0.5	= 0.5	with carry 0
0.5	x 2	= 1.0	= 0.0	with carry 1

MSB ↓ LSB

$\therefore (0.8125)_{10} = (0.1101)_2$

→ convert 0.95 to Decimal number to its binary Equalent

fractional	Radix	Result		Recorded carry
0.95	x 2	= 1.9	= 0.9	with carry 1
0.9	x 2	= 1.8	= 0.8	with carry 1
0.8	x 2	= 1.6	= 0.6	" 1
0.6	x 2	= 1.2	= 0.2	" 0
0.2	x 2	= 0.4	= 0.8	" 0
0.4	x 2	= 0.8	= 0.8	" 0
0.8	x 2	= 1.6	= 0.6	" 1

$(0.95)_{10} = (0.1110001)_2$.

1) 625 to Binary

2 | 625
2 | 312 − 1
2 | 156 − 0
2 | 78 − 0
2 | 39 − 0
2 | 19 − 1

2 | 9 − 1
2 | 4 − 1
2 | 2 − 0
2 | 1 − 0

$\therefore$ ~~$(100011100)_2$~~ $(1001110001)_2$

Verification:

~~2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0~~ 2^0

~~1 0 0 0 1 1 1 0 0~~ 1

$N = 1\times2^8 + 1\times2^4 + 1\times2^3 + 1\times2^2$

(2) 739

2	739	
2	369	— 1
2	184	— 1
2	92	— 0
2	46	— 0
2	23	— 0
2	11	— 1
2	5	— 1
2	2	— 1
2	1	— 0

LSB ↑ MSB

$\therefore (1011100011)_2$

(3) 523

8	523	
8	65	— 3
8	8	— 1
8	1	— 0

$\therefore (1013)_8$

(4) 649

8	649	
8	81	— 1
8	10	— 1
	1	— 2

$(1211)_8$

Ex:- Convert 0.640625 decimal number to its octal Equivalent

Fraction	Radix	Result	Recorded with carry
0.640625	× 8	= 5.125 = 0.125	with carry 5
0.125	× 8	= 1.0 = 0.0	with carry 1

MSB ↓ LSB

$\therefore (0.640625)_{10} = (0.51)_8$

i) (0.925284) (ii) (0.752386)

Fraction	Radix	Result	Recorded with carry
0.925284	× 8	7.402272 = 0.402272	with carry 7
0.402272	× 8	= 3.218176 = 0.218176	with carry 3
0.2	× 8	1.6 = 0.6	with carry 1
0.6	× 8	4.8 0.8	with carry 4
0.8	× 8	6.4 0.4	with carry 6

$\therefore (0.925284)_{10} = (0.73146)_8$

ii) (0.752386)

Fraction	Radix	Result	Recording with carry
0.752386	× 8	6.019088 = 0.01	with carry 6
0.01	× 8	= 0.08	with carry 0
0.08	× 8	= 0.08 0.64	with carry 0
0.12	× 8	5.12 = 0.12	with carry 5
0.96	× 8	0.96	with carry 0
0.68	× 8	1.68	with carry 1

* Convert 0.1289062 Decimal to Hexa Decimal

Fraction	Radix	Result	Recorded with carry
0.1289062	× 16	2.0624992 = 0.0624992	with carry 2
0.0624992	× 16	0.999872 = 0.999872	with carry 9
0.999872	× 16	15.997952 = 0.997952	with carry 15
0.997952	× 16	15	

Wrong

Fraction	Radix	Result	Recorded with carry
0.1289062	× 16	2.0624992 = 0.0625	with carry 2
0.0625	× 16	= 1.0 0.0	with carry 1

MSB → LSB

$\therefore (0.1289062) = (0.21)_{16}$

(iii) (0.15638) iv) (0.29725) Hexa

0.15638 × 16 = 2.50208

0.50

0.8

* Convert decimal number 35.45 to octal number

i) $(35.45)_{10}$

(i) Integer Part

$$8 \lfloor 35$$
$$4 - 3$$

$\therefore (35)_{10} = (43)_8$

ii) fractional Part

Fraction	Radix	Result	Recorded with carry
0.45	x 8	= 3.6 = 0.6	Carry 3
0.6	x 8	= 4.8 = 0.8	Carry 4
0.8	x 8	= 6.4 = 0.4	Carry 6
0.4	x 8	= 3.2 = 0.2	Carry 3
0.2	x 8	= 1.6 = 0.6	Carry 1
0.6	x 8	= 4.8 = 0.8	Carry 4

$\therefore (0.45)_{10} = (0.346314)_8$

finally

$\therefore (35.45)_{10} = (43.346314)_8$

* (i) Convert $(22.64)_{10}$ to Hexa decimal

(ii) Convert $(24.6)_{10}$ to Binary

22.64

16 | 22
1 - 6

(i) $(22.64)_{10}$

(i) Integer part ÷ 22

16 | 22
16 | 16 — 6
1 — 0

22.64

2 | 24
2 | 12 - 0
2 | 6 - 0
2 | 3 - 1
1 - 0

(10100

$\therefore (22)_{10} = (106)_{16}$

(ii) fractional Part : 0.64

fraction	Radix	Result	Recorded carry
0.64	× 16	= 10.24 = 0.24	with carry 10
0.24	× 16	= 3.84 = 0.84	with carry 3
0.84	× 16	= 13.44 0.44	with carry 13
0.44	× 16	= 7.04 0.04	with carry 7
0.04	× 16	= 0.64 = 0.64	with carry 0
0.64	× 16	= 10.24 = 0.24	with carry 10

MSB ↓ LSB

$\therefore (0.64)_{10} = (0.10313701 0)_H$

finally

$\therefore (22.64)_{10} = (16.103137010)_H$

(q) $(24.6)_{10}$

(i) Integration part : 24

$$
\begin{array}{r|l}
2 & 24 \\
2 & 12 - 0 \\
2 & 6 - 0 \\
2 & 3 - 0 \\
 & 1 - 1
\end{array}
$$

$\therefore (24)_{10} = (11000)_2$

fraction	Radix	Result	Recorded with carry
0.6	x 2	= 1.2 = 0.2	with carry 1
0.2	x 2	= 0.4 = 0.4	with carry 0
0.4	x 2	= 0.8	with carry 0
0.8	x 2	1.6 = 0.6	with carry 1
0.6	x 2	1.2 = 0.2	with carry 1

$(0.6)_{10} = (10011)_2$

$\therefore (24.6)_{10} = (11000.10011)_2$

(iv) (0.29725) to Hexa

fraction	Radix	Result	Recorded with carry
0.29725	x 16	4.756 = 0.756	with carry 4
0.756	x 16	12.096 = 0.096	with carry 12
0.096	x 16	15.36 = 0.36	with carry 15
0.36	x 16	5.76 = 0.76	with carry 5
0.76	x 16	12.16 = 0.16	with carry 12
0.16	x 16	2.56 = 0.56	with carry 2
0.56	x 16	8.96 = 0.96	with carry 8

0.96	x 16	15.36 = 0.36	with carry 15
0.36	x 16	5.76 = 0.76	with carry 5

$\therefore (0.29725) = (4\,12\,15\,5\,12\,8\,8\,15\,5)_{16}$

(iii) (0.15638)

fraction	Radix	Result	Recorded with carry
0.15638	x 16	2.50208 = 0.502	with carry 2
0.502	x 16	8.032 = 0.032	with carry 8
0.032	x 16	0.512 = 0.512	with carry 0
0.512	x 16	8.192 = 0.192	with carry 8
0.192	x 16	3.072 = 0.072	with carry 3
0.072	x 16	1.152 = 0.152	with carry 1
0.152	x 16	2.432 = 0.432	with carry 2
0.432	x 16	6.912 = 0.912	with carry 6
0.912	x 16	14.592 0.592	with carry 14
0.592	x 16	9.472 0.9472	with carry 9
0.9472	x 16	15.1552 0.1552	with carry 15

$\therefore (0.15638)_{10} = (2\,8\,0\,8\,3\,12\,6\,14\,9\,15)_H$

$= (28083C6E9F)_H$

Complement of numbers:

(i) 1's complement representation:

→ The 1's complement of a binary number is the number that results when we change all 1's is 0's and all 0's to 1's

Ex:- find 1's complement of $(1101)_2$

$(1101)_2$

= 0010

Ex:- find 1's complement of $(10111010111)_2$

$(10111010111)_2$

010001010000

(ii) 2's complement representation:

→ The 2's complement is the binary number that results when we add 1 to the 1's complement. It can be represented as

[2's complement = 1's complement + 1]

Addition operation:

truth table

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

→ The 2's complement form is used to represent a (-ve) numbers

Ex:- find 2's complement of $(1001)_2$

$(1001)_2$

= 0110 — 1st complement

= 0 1 1 0

+1 → adding +1

0 1 1 1

Ex:- $(1010100110)_2$

0101011001 → 1st complement

0 1 0 1 0 1 1 0 0 1

+1 → adding +1

0 1 0 1 0 1 1 0 1 0

$\therefore (1010100110)_2 = (0101011010)$

1's Complement Subtraction:-

→ For subtraction of two numbers we have two cases

(i) subtraction of smaller number from larger number

L - S

(ii) subtraction of larger number from smaller number

S - L

(i) Smaller number from larger number:-

Steps:-

1. Determine the 1's complement of the smaller number.
2. Add the 1's complement of the larger number.

3. Remove the carry and adding to the Result this is called End around the carry

Ex:- Subtract ~~1010~~ $(101011)_2$ from $(111001)_2$ using the 1's Complement method

$(101011)_2$ from $(111001)_2$

(1) first u-find big number are large number

2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	1

= 43

2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	0	1

= 57

L - S

```
      1 1 1 0 0 1
      0 1 0 1 0 0   → 1st complement of 101011
   ----------------
  (1) 0 0 1 1 0 1
                +1
   ----------------
      0 0 1 1 1 0
```

End around Carry

57
43
14

∴ (001110)

Ex:- $(1101011)_2$ from $(111001)_2$

$(1101011)_2$ from (0111001)

2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	1	0	1	1

107

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	1	0	0	1

57

L - S

```
      1 1 0 1 0 1 1
      1 0 0 0 1 1 0   → 1st complement of 0111001
   ------------------
  (1) 0 1 1 0 0 0 1
```

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

107
57
50

∴ (0110010)

= 50 //

(ii) Subtraction of large number from smaller number:

Steps: S − L

1. Determine the 1's complement of the larger number
2. Add the 1's complement of the smaller number
3. Answer is in 1's complement from to set the answer in true form take the 1's complement and assign (−ve) sign to the answer.

Ex: subtract $(111001)_2$ from $(101011)_2$ using the 1's complement method

57 43
$(111001)_2$ from $(101011)_2$

1 0 1 0 1 1
0 0 0 1 1 0 ← 1's complement

1 1 0 0 0 1
0 0 1 1 1 0 → 1's complem

−14

Ex: $(1111101)_2$ from $(1101011)_2$

$2^6\,2^5\,2^4\,2^3\,2^2\,2^1\,2^0$ $\qquad$ $2^6\,2^5\,2^4\,2^3\,2^2\,2^1\,2^0$

1 1 1 1 1 0 1 $\qquad$ 1 1 0 1 0 1 1

125 $\qquad$ 107

(a) S − L

```
  1 1 0 1 0 1 1
  0 0 0 0 0 1 0  → 1's complement of (1111101)
            1
  -------------
  1 1 0 1 1 0 1
  -------------
- 0 0 1 0 0 1 0  → 1's complement
```

−18 //

(b) L − S

```
    1 1 1 1 1 0 1
    0 0 1 0 1 0 0  ← 1's
    1 1 1 1
    -------------
(1) 0 0 1 0 0 0 1
                1
    -------------
    0 0 1 0 0 1 0
```

End around carry

= 18 //

2's complement subtraction:-

(i) Subtraction of smaller number from larger number:-

→ Determine the 2's complement of a small number

→ Add the 2's complement to the large number

→ Dicaurd the carry (or) remove the carry

Ex:- Subtract $(101011)_2$ from $(111001)_2$
43 57

(i) L - S

2's complement

```
  1 0 1 0 1 1
  0 1 0 1 0 0 → 1st compl
            1
  -----------
  0 1 0 1 0 1
```

```
    1 1 1 0 0 1
(+) 0 1 0 1 0 1
  -------------
  1 0 0 1 1 1 0
```

↓ remove

(ii) S - L

2's complement

```
  1 1 1 0 0 1
  0 0 0 1 1 0 → 1's
            1
  -----------
  0 0 0 1 1 1
```

```
    1 0 1 0 1 1
    0 0 0 1 1 1
  -------------
    1 1 0 0 1 0
  -------------
    0 0 1 1 0 1
              1
  -------------
  - 0 0 1 1 1 0
```

(ii) Subtraction of larger number from smaller number:-

→ Determine the 2's complement of the large number

→ Add the 2's complement to the smaller n

→ Answer is in the 2's complement form to get the Answer in the true from to take the 2's complement add assign (-ve) sign to the Answer

Signed binary numbers:-

→ The sign magnitude formate for 8 bit signed number is given by

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0

B_7 → sign Bit; $\underbrace{B_6 \ldots B_0}_{\text{magnitude}}$

Ex:-
1) +6 = 0000 0110
2) −14 = 1000 1110
3) +24 = 0001 1000
4) −64 = 1100 0000
5) +127 = 0111 1111
6) −128 = 1111 1111

→ The maximum (+ve) number +127 = 0111 1111

Binary Airthematic

Rules for binary addition:-

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

Ex: Add $(1010)_2$ & $(0011)_2$

```
  1 0 1 0
  0 0 1 1
     1
 --------
  1 1 0 1
```

Ex: Add 28 and 15 binary

16 8 4 2 1

```
   1 1 1 0 0
   0 1 1 1 1
 1 1 1 1 1
 -----------
 1 0 1 0 1 1
32
```

28
15
43

Ex: B

Binary subtraction

A	B	Difference	Barrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Ex: subtract $(0101)_2$ from $(1011)_2$
5 11

```
  1 0 1 1
  0 1 0 1
 --------
  0 1 1 0
```

Hexa Decimal Arithmatic:

(1) $9_{16} + 3_{16} = C_{16}$

(2) $9_{16} + 7_{16} = (16-16) = 0$ carry 1

(3) $A_{16} + 8_{16} = (18-16) = 2$ carry 1

① Add $(3F8)_{16}$ and $(5B3)_{16}$ $(26-16)$

$$\begin{array}{rccc} & \overset{1}{3} & F & 8 \\ \text{Add} & 5 & B & 3 \\ \hline & (9 & A & B)_{16} \end{array}$$

② Add Hexadecimal number $(4FB)_{16}$ & $(75D)_{16}$, $(A12)_{16}$, $(C39)_{16}$

$$\begin{array}{ccc} \overset{1}{4} & \overset{2}{F} & B \\ 7 & 5 & D \\ A & 1 & 2 \\ C & 3 & 9 \\ \hline 22 & A & 3 \end{array}$$

i) $35 = (35-16) = 19$ carry 1
$= (19-16) = 3$ carry 2

ii) $26 = (26-16) = 10$ carry 1

iii) $34 = (34-16) = 18$ carry 1
$(18-16) = 2$ carry 2

Subtraction with 15's Complement

→ The 15's complement of a Hexadecimal number is formed by subtracting each digit from 15

Ex:- find 15's complement of $(A9B)_{16}$

$$\begin{array}{rccc} & 15 & 15 & 15 \\ (-) & A & 9 & B \\ \hline & 5 & 6 & 4 \end{array}$$

Ex:- Use the 15's complement method of subtraction to compute $(B02)_{16} - (98F)_{16}$

$$\begin{array}{rccc} (-) & 15 & 15 & 15 \\ & 9 & 8 & F \\ \hline & 6 & 7 & 0 \end{array}$$

first (-ve) value tho (-)ట

(+) Adding $17-16=1$

$$\begin{array}{rccc} & B & 0 & 2 \\ & 6 & 7 & 0 \\ \hline ① & 1 & 7 & 2 \\ & & & 1 \\ \hline & 1 & 7 & 3 \end{array}$$

Subtraction of 15's complement

→ find 15's complement of subtratend

→ Add to Hexadecimal numbers (1st number and 15's complement of 2nd number)

→ If carry is produced in the addition, add carry to the least significant bit of the sum, otherwise find 15's complement of the sum as a result with a (-ve) sign.

Ex: Use the 15's complement method of subtracted to Compute $(69B)_{16} - (C14)_{16}$

(22-16)=6 carry 1
24-16=8 carry 1

step 1:

	15	15	15
(-)	C	1	4
	3	E	B

step 2:

	6	9	B
(+)	3	E	B
	A	8	6

step 3:

15	15	15
- A	8	6
- 5	7	9

no carry

$\therefore (69B)_{16} - (C14)_{16} = (-579)_{16}$

Subtraction of 16's complement

→ The 15's complement of a Hexa decimal number is found by subtracting each digit from "15" and add 1.

Steps: steps for Hexadecimal subtraction using 16's complement method.

→ find 16's complement of subtratend

→ Add to Hexadecimal numbers (1st number & 16's complement of the 2nd number)

→ If carry is produced in the addition it is discarded or removed, otherwise find 16's complement of the sum as a result with a (-ve) sign

Ex:- find the 16's complement of $(A8C)_{16}$

$$\begin{array}{rccc} & 15 & 15 & 15 \\ - & A & 8 & C \\ \hline & 5 & 7 & 3 \\ & & & +1 \\ \hline & 5 & 7 & 4 \end{array}$$

← 15's complement

2. use the 16's Complement method of subtraction to compute $(CB2)_{16} - (972)_{16}$

Step 1:-

$$\begin{array}{rccc} & 15 & 15 & 15 \\ - & 9 & 7 & 2 \\ \hline & 6 & 8 & D \\ & & & +1 \\ \hline & 6 & 8 & E \end{array}$$

← 15's

← 16's complement

Step 2:-

$$\begin{array}{rccc} & C & B & 2 \\ + & 6 & 8 & E \\ \hline & 3 & 4 & 1 \end{array}$$

(16-16) = 1 carry
(20-16) = 4 carry
(19-16) = 3 carry 1

3. use the 16's complement method of subtraction to compute $(387)_{16} - (854)_{16}$

Step:-1

$$\begin{array}{rccc} & 15 & 15 & 15 \\ - & 8 & 5 & 4 \\ \hline & 7 & A & B \\ & & & +1 \\ \hline & 7 & A & C \end{array}$$

← 16's complement

$(19-16) = 3$ carry 1
$(22-16) = 6$ carry 1

Step 2:

$$\begin{array}{cccc} & 3 & B & 7 \\ (+) & 7 & A & C \\ \hline & B & 6 & 3 \\ \hline \end{array}$$

Step 3: If No carry

$$\begin{array}{cccc} & 15 & 15 & 15 \\ - & B & 6 & 3 \\ \hline & 4 & 9 & C \\ & & & 1 \\ \hline & -4 & 9 & D \\ \hline \end{array}$$

$\therefore (3B7)_{16} - (854)_{16} = (-49D)_{16}$

If no carry

$$\begin{array}{ccc} 15 & 15 & 15 \\ B & 6 & 3 \\ \hline 4 & 9 & C \\ & & 1 \\ \hline -4 & 9 & D \end{array}$$

Binary codes:

Codes

- Weighted
  - Binary
  - BCD (Binary coded Decimal)
    - 8421
    - 2421
    - 3321
    - 4221
    - 5211
    - 5311
    - 5421
    - 6311
    - 7421
    - $74\bar{2}\bar{1}$
    - $84\bar{2}\bar{1}$
- Non-weighted
  - Ex-3
  - Gray
  - five-bit BCD code
- Reflective
  - 2421
  - 5211
  - Ex-3
- Sequential
  - 8421
  - Ex-3
- Alpha numaric
  - ASCII
- Error Defective & correction
  1. Parity
  2. Hamming Code

Excess3 Code (Ex-3)

Decimal	Ex-3	Binary code
0	0+3=3	0011
1	4	0100
2	5	0101
3	6	0110
4	7	0111

* To find binary code and Gray code the given binary number is 1101010101

truth table of Hexa X-OR Gate

A, B → C

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

$= A \oplus B$

A	B	$C = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

binary-gray

(i) 1101010101

1011111111

$\therefore$ 1011111111

(ii) 1011101011

111001110

gray-binary

Gray code: (i) 1101010101

1 1 0 1 0 1 0 1 0 1

1 0 0 1 1 0 0 1 1 0

$\therefore$ 1001100110

(i) BCD Addition:

Ex: BCD of 58

5	8
↓	↓
0101	1000

(i) sum Equals "9" or less with Carry is "0"

Ex: Add 3 and 6 in BCD

$$\begin{array}{rcl} 3 & — & 0011 \\ +6 & — & 0110 \\ \hline 9 & & 1001 \rightarrow \text{Ans} \end{array}$$

(ii) ~~Add~~ sum > 9 with carry is "0"

Ex: Add 6 and 8

$$\begin{array}{rcl} 6 & — & 0110 \\ +8 & — & 1000 \\ \hline 14 & & 1110 \rightarrow 14 \end{array}$$

0110 ← Add 6

Carry 1

0001 0100

1 4 ⟶ BCD

(iii) sum Equals to 9 or less with Carry 1:

Ex: Add 8 + 9

$$\begin{array}{rcl} 8 & — & 1000 \\ +9 & — & 1001 \\ \hline 17 & & 000\,1\,0001 \end{array}$$

← Invalid BCD

0110

0001 0111

1 7

* Perform Each of the following decimal addition in 8421 BCD

① 24 + 18

		8421	8421
24	—	0010	0100
18	—	0001	1000
42		0011	1100
		carry	0110 ← Add 6
		0011	0010
		0100	0010 ← valid BCD
		4	2

② 48 + 58

58	—	0101	1000
48	—	0100	1000
106		1001	0000
		1010	0000 ← Add 6
			0110
		1010	0110
		10	6
		A	

③ 175 + 326

175	—	0001	0111	0101
326	—	0011	0010	0110
501		0100	1001	1011
				0110 ← Add 6
			1001	0001
			1010	

```
      1010      ← Add 6
  (1) 0110
      0000
 0100
 0101 ← Add 6
 0101   0000   0001
```

(4) 589 + 199

```
 589  ———  0101  1000  1001
 199  ———  0001  1001  1001
 ---
 788          1    (1)      (1)    1
           0110 | 0001 | 0010
              1        1
           -----------------------
           0111   0010   0010
                         0110 ← Add 6
                         ----
                         1000
           -----------------------
                  0010
                  0110 ← Add 6
                  ----
                  1000
           -----------------------
           0111   1000   1000
             7      8      8
```

BCD subtraction:

(a) subtraction with 9's complement

Digits	9's complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2

Digits	9's com
8	1
9	0

Regular subtraction

1)
$$\begin{array}{r} 8 \\ -\,2 \\ \hline 6 \end{array}$$

(2)
$$\begin{array}{r} 9 \\ -\,5 \\ \hline 4 \end{array}$$

(3)
$$\begin{array}{r} 4 \\ -\,8 \\ \hline -4 \end{array}$$

9's complement subtraction

(i)
$$\begin{array}{r} 8 \\ 7 \\ \hline (1)\ 5 \\ 1 \\ \hline 6 \end{array}$$

(9's complement of 2)

(ii)
$$\begin{array}{r} 9 \\ 4 \\ \hline (1)\ 3 \\ 1 \\ \hline 4 \end{array}$$

(iii)
$$\begin{array}{r} 4 \\ 1 \\ \hline 5 \end{array}$$

(9's complement of result no Carry indicates that the answer is negative and completed from)

* perform Each of the following decimal subtraction in 8421 BCD using 9's complement method.

(i) 79 − 26

$$\begin{array}{r} 79 \\ -\,26 \\ \hline 53 \end{array}$$

[9 − 2 = 7
6 − 2 = 3]

0111	1001	
0111	0011	← (9's complemt of 26)
1110	1100	
	0110	← Add 6
1110	(1) 0010	
1		
0110		Add 6
(1) 0101	0010	carry
	1	
0101	0011	

10's complement

(a)

$$\begin{array}{r} 8 \\ -\ 2 \\ \hline 6 \end{array}$$

$$\begin{array}{r} 8 \\ ①\ 8 \\ \hline 6 \end{array}$$

remove

→ (10's complement (10−2)=8)

(2)

$$\begin{array}{r} 9 \\ -\ 5 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 9 \\ ①\ 5 \\ \hline 4 \end{array}$$

remove

→ (10's complement (10−5)=5)

(3)

$$\begin{array}{r} 4 \\ -8 \\ \hline -4 \end{array}$$

$$\begin{array}{r} 4 \\ 2 \\ \hline 6 \end{array}$$

→ (10's complent of (10−8)=2)

Result no carry indicate that the answer is (−ve) and in the 10's complement form (or) in completed form

UNIT - 2

Concept of Boolean Algebra

Introduction:

→ In 1854 George Boolean introduced a systamatic treatment of logic and developed for this purpose of an algebraic system now called Boolean Algebra.

→ In 1938 C.E Shanon introduced a two-valued Boolean Algebra called a switching Algebra

Fundamental Postulateds of Boolean Algebra

S.NO	Postulates	Comments
1.	→ Result of each operator is either 0 or 1	$1, 0 \in B$
2.	a) $0+0=0$, $0+1=1+0=1$ b) $1\cdot 1=1$; $0\cdot 1=1\cdot 0=0$	Identity element "0" for "+" and "1" for "·"
3.	a) $(A+B)=(B+A)$ b) $(A\cdot B)=(B\cdot A)$	Commulative law
4.	a) $A\cdot(B+C)=(A\cdot B)+(A\cdot C)$ b) $A+(B\cdot C)=(A+B)\cdot(A+C)$	Distributive law
5.	a) $A+\bar{A}=1$, $0+\bar{0}=1$, $1+\bar{1}=1$ b) $A\cdot\bar{A}=0$, if $0\cdot\bar{0}=0\cdot 1=0$ $1\cdot\bar{1}=1\cdot 0=0$	Complement

$[\bar{0}=1$

$\bar{1}=0]$

Basic theorems and propertys

t.me/jntukonlinebits

Duality:-

→ The principle of Duality theorem says that starting with a boolean relation, we can derive another boolean relation by

1. Changing each "OR" sign to an "AND" sign
2. Changing each "AND" sign to an "OR" sign
3. Any 0 or 1 operating in the Expression of boolean function.

Ex:- Dual of relation $A+\bar{A}=1$ is $A\cdot\bar{A}=0$

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

$A\cdot\bar{A}=0$

principal of Duality theorem says that starting with a

Theorems:-

Laws of Boolean algebra:-

Boolean Expression	B.A law (or) Rule
⇒ $A+1=1$ $A\cdot 0=0$	Annulment
⇒ $A+0=A$ $A\cdot 1=A$	Identity
⇒ $A+A=A$ $A\cdot A=A$	Idempotent
⇒ $\bar{\bar{A}}=A$ NOT $\bar{A}$ = A	Double Nagation

Boolean Expression	B.A law (or) Rule
$\Rightarrow A+\bar{A}=1$ $A\cdot\bar{A}=0$	Complement
$\Rightarrow A+B=B+A$ $A\cdot B=B\cdot A$	Commutative law
*** $\Rightarrow \overline{A+B}=\bar{A}\cdot\bar{B}$ $\overline{AB}=\bar{A}+\bar{B}$	De-morgan

Boolean Algebra functions:

Function	Description	Expression
1.	NULL	0
2.	Identity	1
3.	INput A	A
4.	Input B	B
5.	NOT A	$\bar{A}$
6.	NOT B	$\bar{B}$
7.	A AND B (AND)	$A\cdot B$
8.	A AND NOT B	$A\cdot\bar{B}$
9.	NOT A AND B	$\bar{A}\cdot B$
10.	NOT AND (NAND)	$\overline{A\cdot B}$
11.	A OR B (OR)	$A+B$
12.	A OR NOT B	$A+\bar{B}$

Function	Description	Expression
13.	NOT A OR B	$\bar{A}+B$
14.	NOT OR (NOR)	$\overline{A+B}$
15.	Exclusive OR (X-OR)	$A\bar{B}+\bar{A}B$
16.	Exclusive NOR (X-NOR)	$AB+\bar{A}\bar{B}$

Theorems:

① a) $A+A=A$

b) $A\cdot A=A$

② a) $A+1=1$

b) $A\cdot 0=0$

③ a) $\bar{\bar{A}}=A$

④ a) $A+AB=A$

b) $A(A+B)=A$

⑤ a) $A+\bar{A}B=A+B$

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

① ⓐ Proof:

$A+A=A$

$\begin{cases} \text{if } A=0;\ 0+0=0 \\ \text{if } A=1;\ 1+1=1 \end{cases}$

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

L.H.S

$$A+A=(A+A)\cdot(1)$$
$$=(A+A)(A+\bar{A})$$
$$=AA+A\bar{A}+AA+A\bar{A}$$
$$=AA+A\bar{A}$$
$$=A(A+\bar{A})$$
$$=A\ //$$

1 ⓑ Proof:

$A\cdot A=A$

$\begin{cases} A=0 \Rightarrow 0\cdot 0=0 \\ A=1;\ 1\cdot 1=1 \end{cases}$

$[A\cdot\bar{A}=0]$

L.H.S

$$A\cdot A=A\cdot A+(0)$$
$$=A\cdot A+A\cdot\bar{A}$$
$$=A(A+\bar{A})$$
$$=A\ //$$

② ⓐ $A+1=1$

Proof ÷ $A+1 = (1)A+1$

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

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

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

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

$= A + \bar{A}$

$= 1$

$[A\cdot\bar{A}=0$

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

$A+1=1$

$\left\{\begin{array}{l} \text{if } A=0;\ 0+1=1 \\ A=1;\ 1+1=1 \end{array}\right.$

ⓑ $A\cdot 0 = 0$

proof ÷ $A\cdot 0 = A\cdot(A\cdot\bar{A})$

$= A(0)$

$= 0$

$A\cdot 0=0$

$A=0;\ 0\cdot 0=0$

$A=1;\ 1\cdot 0=0$

③ $\bar{\bar{A}} = A$

Proof ÷ if $A=0$; $\bar{\bar{0}} = \bar{1} = 0$

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

$[\bar{0}=1$

$\bar{1}=0]$

④ ⓐ $A+AB=A$

proof ÷ $A+AB = A(1+B)$

$= A(1)$

$= A$

$[\because 1+B=1]$

ⓑ $A(A+B) = A$

proof ÷ $A(A+B) = (A+AB)(A+B)$

$= AA + AB + ABA + AB\cdot B$

$= A + AB + A\cdot AB + AB$

$= A + AB + AB + AB$

$= A(1+B)+AB$

$= A+AB$

$= A(1+B)$

$= A$ //

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

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

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

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

$[A\cdot\bar{A}=0]$

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

$= 0+AB$

$= AB$ //

***

De-Morgan's theorem:

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

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

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

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

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

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

→ Consensus theoram:

→ In simplification of Boolean Expresion, an Expresion of the form $AB+\bar{A}C+BC$. The term BC is redented and can be elemin-ated to form the Equilent expresion $AB+\bar{A}C$.

→ The theoram is used for this simplification is known as Consensus theoram. and it is stated as

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

Proof: L.H.S ⇒ $AB+\bar{A}C+BC = AB+\bar{A}C+BC(1)$

$[\because A+\bar{A}=1]$

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

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

$[\because 1+C=1$
$1+B=1]$

$= AB(1+C)+\bar{A}C(1+B)$

$= AB+\bar{A}C$

* Solve the given Expresion using Consensus theorem

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

$\Rightarrow \bar{A}\bar{B}+AC+B\bar{C}+\bar{B}C(1)+AB$ $[A+\bar{A}=1]$

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

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

$\Rightarrow \bar{A}\bar{B}(1+C)+AC(1+\bar{B})+B\bar{C}+AB$ $[1+C=1$
$1+\bar{B}=1]$

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

$\Rightarrow \bar{A}\bar{B}+AC+B\bar{C}+AB(1)$

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

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

$\Rightarrow \bar{A}\bar{B}+AC(1+B)+B\bar{C}(1+A)$

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

Dual of Consensus theorem:

→ The Dual form of consensus theorem is stated as

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

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

$(0 + AC + \bar{A}B + BC)(B+C) = (0 + AC + \bar{A}B + BC)$ $[A \cdot \bar{A} = 0]$

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

$[BC + BC = BC]$

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

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

$\bar{A}B(1+C) + BC(1+A) + AC = AC + \bar{A}B + BC$

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

Boolean function (or) switching function:

→ Boolean equations are constructed by connecting the boolean constants and variables with the boolean operation.

→ This Boolean expressions are known as boolean formulas we use boolean expression to describe boolean functions

for Example: If Boolean Expresion $(A+\bar{B})C$ is used to describe the function of f, then boolean function is written as $f(A,B,C) = (A+\bar{B})C$

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

→ Let us consider the whole four variable boolean function.

Product-terms:

$f(A,B,C,D) = A + (B\bar{C}) + (A\bar{C}D)$ ← product terms; ↓ literals

Sum - Terms :-

Sum terms

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

Literals

→ The literals and terms are arranged in the forms in two

1. sum of product (sop)

2. product of sum (pos)

1. sum of products :-

→ The sum of product is also called Disjunctional normal form (or) Disjunctional normal formula

→ the words sum & product are discribed from the symbolic representation "OR" and "AND" function (+ and *)

Ex:-1 $f(A,B,C) = A\bar{B}C + A\bar{B}\bar{C}$

sum terms

products

Ex:-2 $f(P,Q,R) = \bar{P}Q + QR + RS$

sum terms

Products

2. Product of sum :-

→ The product of sum is also called conjunctional normal form (or) conjunctional normal formula.

→ A product of sum is any group of sum terms ANDed to gether.

Ex:- $f(A,B,C) = (A+B+C)\cdot(A+\bar{B}+\bar{C})$

Product

Sum

Ex: $f(p,q,r) = (p+q)\cdot(r+p)\cdot q$

Sum — product

* **Canonical form:** (Standard sop and pos forms)

→ Basically canonical forms are two types

1. standard sop (or) minterm canonical form
2. standard pos (or) maxterm canonical form

1. **standard sop:** (minterm canonical form)

→ The ssop is given by $f(A,B,C) = (A\bar{B}C) + (ABC) + (\bar{A}BC)$

Each product term is consistent all literals in either completed form or uncompleted form

2. **standard pos:** (maxterm)

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

Each sum term consists of all literals in the completed form

* **Converting expressions in standard sop (or) pos form**

Converting to standard sop form:

Steps to convert sop to standard sop

Steps:
1. find the missing literal in each product term if any
2. "AND" each product term having missing literals which term 1 formed by "OR" ing the literals and its complement
3. Expand the terms by applying Distributive law and recorded the literals in the product terms
4. Reduce the expression omitting (or) remove repeated product terms if any. because $(A+A) = A$

Ex: Convert the given Expression in standard SOP form

$$f(A,B,C) = AC + AB + BC$$

Step: 1 ~~f(A,B,C) =~~ find the missing literal in Each product

$$f(A,B,C) = AC + AB + BC$$

A literal is missing

C literal missing

B literal missing

Step: 2 AND product terms with (missing literal + complement)

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

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

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

Step 3: Expand the term and recorded the terms

Expand:

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

Recorded:

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

Step 4: omitting the repeted product term

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

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

Ex: Convert the given Expression in standard SOP form

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

Step 1: find

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

B literal missing

C literal missing

Step 2: AND product term with (missing literal + complement)

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

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

Step3:

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

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

Step4: omitting the repeated product term

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

Convert pos to standard pos form:

Step1:

1. find the missing literals in each sum term if any.
2. OR each sum term having missing literal formed by AND ing the literal and its complement
3. Expand the terms by applying distributive law and recorded the literal in the sum terms
4. Reduce the Expression by omitting repeated sum term if any

$A \cdot A = A$

Ex: convert the given expression in standard pos form

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

$[A \cdot \bar{A} = 0$
$B \cdot \bar{B} = 0$
$C \cdot \bar{C} = 0]$

Step 1: find the missing leteral in each product

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

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

→ B literal missing

→ A literal missing

→ C literal missing.

Step2: OR product term with (missing leteral + complement)

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

Step3: Expand the term and recorded the terms

Expand:

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

Recorded:

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

Step 4: omitting the repeated product term

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

Ex: Convert the given expression in standard pos forms

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

Step 1: find the missing literal in each product

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

B, C literals missing

Step 2: OR product term with (missing literal + complement)

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

Step 3: Expand the term and recorded the terms

Expand: $[A+(B+C) \cdot (B+\bar{C}) \cdot (\bar{B}+C) \cdot (\bar{B}+\bar{C})] \cdot (A+B+C)$

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

Step 4: omitting the repeated product term

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

M- Notations:- (minterms and maxterms)

Decimal NO	Binary Numbers (421)	Minterm (m) (SOP)	Maxterm (M) (POS)
0	000	$\bar{A}\bar{B}\bar{C}$ (m_0)	$A+B+C$ (M_0)
1	001	$\bar{A}\bar{B}C$ (m_1)	$A+B+\bar{C}$ (M_1)
2	010	$\bar{A}B\bar{C}$ (m_2)	$A+\bar{B}+C$ (M_2)
3	011	$\bar{A}BC$ (m_3)	$A+\bar{B}+\bar{C}$ (M_3)
4	100	$A\bar{B}\bar{C}$ (m_4)	$\bar{A}+B+C$ (M_4)
5	101	$A\bar{B}C$ (m_5)	$\bar{A}+B+\bar{C}$ (M_5)
6	110	$AB\bar{C}$ (m_6)	$\bar{A}+\bar{B}+C$ (M_6)
7	111	ABC (m_7)	$\bar{A}+\bar{B}+\bar{C}$ (M_7)

$$\boxed{\text{Minterm} = \text{Complement of maxterm}}$$

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

$= m_0 + m_1 + m_3 + m_6$

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

Ex:- $f(A,B,C) = (A+B+\bar{C}) \cdot (A+\bar{B}+\bar{C}) \cdot (\bar{A}+\bar{B}+C)$

$= M_1 \cdot M_3 \cdot M_6$

$= \pi M(1,3,6)$

Ex:- To find sum of product form to the given table

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

[ABC is a input variables]

[Sop is 1's]

A. B C	y
1 0 1	0
1 1 0	1
1 1 1	0

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

$$= m_2 + m_3 + m_6$$

$$= \Sigma m(2,3,6)$$

To find the product of sum from the given table [Product of sum ks zero's]

A+B+C	y
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	1

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

$$= M_2 \cdot M_5$$

$$= \Pi M(2,5)$$

Algebraic Simplifications:

1. $A \cdot \bar{A}C = 0$
2. $ABCD + ABD = ABD$
3. $A(A+B) = A$
4. $AB + ABC + AB(D+E) = AB$
5. $xy + xyz + xy\bar{z} + \bar{x}yz$
6. $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC$
7. $ABC + A\bar{B}C + AB\bar{C}$
8. $A + \bar{A}B + A\bar{B} = A+B$

Ex: $\overline{\overline{AB} + \bar{A} + AB}$

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

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

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

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

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

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

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

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

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

$[\bar{1} = 0]$

$\Rightarrow \overline{1+\bar{B}}$

$\Rightarrow \bar{1}$

$\Rightarrow 0$

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

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

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

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

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

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

$= \overline{1+\bar{B}} = \bar{1} = 0$

(1) $A \cdot \bar{A}C = 0$

$(A \cdot \bar{A})C = 0$

$(0)C = 0$

$A \cdot \bar{A}C = 0$

② $ABCD + ABD = ABD$ $[1+C=1]$

$\Rightarrow ABD(1+C)$

$\Rightarrow ABD$

$\Rightarrow ABCD + ABD = ABD$

③ $A(A+B) = A$ $[A \cdot A = 1]$

$\Rightarrow A \cdot A + AB$

$\Rightarrow A + AB$

$\Rightarrow A(1+B)$

$\Rightarrow A$

$A \cdot A + AB = A$

$A + AB = 0$

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

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

$A(1+B)$

$= A$

④ $AB + ABC + AB(D+E) = AB$

$\Rightarrow AB + ABC + AB(D + E)$

$\Rightarrow AB(1+C) + ABD + ABE$

$\Rightarrow AB + AB(D+E)$

$\Rightarrow AB(1+D+E)$

$\Rightarrow AB[(1+D)+E]$

$\Rightarrow AB[1+E]$

$\Rightarrow AB(1)$

$\Rightarrow AB$

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

$= xy + yz$

$xy + xy(z+\bar{z}) + \bar{x}yz$

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

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

$xy + y(x+z)$

$xy + xy + z$

$xy + xy$

$y(x+z)$

5. $xy + xyz + xy\bar{z} + \bar{x}yz = y(x+z)$

$\Rightarrow xy(1) + xyz + xy\bar{z} + \bar{x}yz$

$\Rightarrow xy(z+\bar{z}) + xyz + xy\bar{z} + \bar{x}yz$

$\Rightarrow xyz + xy\bar{z} + xyz + xy\bar{z} + \bar{x}yz$

$\Rightarrow xy(z+\bar{z}) + xy(z+\bar{z}) + \bar{x}yz$

(or)

$xy + xyz + xy\bar{z} + \bar{x}yz$

$xy(1+\bar{z}) + yz(x+\bar{x})$

$xy + yz$

$y(x+z)$ //

$$\Rightarrow \cancel{xy} + \cancel{xy} + \bar{x}yz$$

$$\Rightarrow xy + \bar{x}yz$$

$$\Rightarrow y(x+\bar{x}z) \Rightarrow y((x+\bar{x})\cdot(x+z)) \Rightarrow y(1)\cdot(x+z)$$

$$\Rightarrow y(x+z)$$

$[(x)+(\bar{x}z)$
$(x+\bar{x})\cdot(x+z)]$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$AC+AB$$

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

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

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

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

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

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

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

$\Rightarrow (1)\cdot(A+B)$

$\Rightarrow A+B$ //

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

Ex: Simplify the following three variable Expression by using Boolean algebra $Y=\Sigma m(1,3,5,7)$

Given Data $Y=\Sigma m(1,3,5,7)$

$= m_1+m_3+m_5+m_7$

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

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

$=\bar{A}C(1)+AC(1)$

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

$= C$ //

$Y=\Sigma m(1,3,5,7)$
$=\bar{A}\bar{B}C+\bar{A}BC+A\bar{B}C+ABC$
$=\bar{A}C(B+\bar{B})+AC(B+$
$=\bar{A}C+AC$
$=C(A+\bar{A})$
$=C(1)$

Ex: Simplify the following three variable Expression by using Boolean algebra $Y=\Pi M(1,3,5,7)$

Given that $Y=\Pi M(1,3,5,7)$

$= M_1\cdot M_3\cdot M_5\cdot M_7$

$= M_1\cdot M_3\cdot M_5\cdot M_7$
$=(\bar{A}+B+\bar{C})\cdot(A+\bar{B}+C)$

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

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

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

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

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

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

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

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

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

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

Ex(2) Simplify the following three Variable Expression. Convert the Expression into minterm complementary property. ~~and~~ &

$$y = \pi_M(1,3,5,7)$$

Sol: The given Expression is maxterm $y = \pi_M(1,3,5,7)$.

To convert given sequences to minterm Complementary form

$\boxed{\text{minterm} = \text{Complementary of Maxterm}}$

The minterm is given by $= \Sigma m(0,2,4,6)$

$y = \Sigma m(0,2,4,6)$ $= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$

$= m_0 + m_2 + m_4 + m_6$ $= \bar{A}\bar{C}(B+\bar{B}) + A\bar{C}(B+\bar{B})$

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

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

$= \bar{C}$

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

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

(14m) **

* ~~transform~~ Each of the following Canonical Expression into its other Canonical form its decimal notations.

(i) $f(x,y,z) = \Sigma m(1,3,5)$

(ii) $f(w,x,y,z) = \pi_M(0,2,5,6,7,8,9,11,12)$

(i) $f(x,y,z) = \Sigma m(1,3,5)$ $f(x,y,z) = \pi_M(0,2,4,6,7)$

$f(x,y,z) = \pi_M(0,2,4,6,7)$

$= M_0 \cdot M_2 \cdot M_4 \cdot M_6 \cdot M_7$ $(x+y+z) \cdot (x+\bar{y}+z) \cdot (\bar{x}+y+z) \cdot (\bar{x}+\bar{y}+z) \cdot (\bar{x}+\bar{y}+\bar{z})$

$= (\cancel{x+y+z}) \cdot \cancel{(A+\bar{B}+C)} \cdot \cancel{(\bar{A}+B+C)} \cdot \cancel{(\bar{A}+\bar{B}+C)} \cdot \cancel{(\bar{A}+\bar{B}+\bar{C})}$

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

$= xyz+x\bar{y}z+xz+x\bar{y}z+xyz+x\bar{y}z+xz+xyz+xyz+\bar{x}yz+\bar{x}yz+\bar{x}yz+$
$yz+\bar{x}yz+yz+xyz+x\bar{y}z+xz+\bar{x}\bar{y}z+\bar{x}\bar{y}z+\bar{x}\bar{y}z+\bar{x}\bar{y}z+$
$\bar{y}z+\bar{y}z+\bar{x}z+\bar{x}\bar{y}z+\bar{x}z+\bar{x}yz+yz+\bar{x}z+z\bar{y}+z)$

$= xyz+x\bar{y}z+xz+\bar{x}yz+yz+\bar{x}\bar{y}z+\bar{y}z+\bar{x}z+z$

$= xz(y+\bar{y})+yz(\bar{x}+1)+\bar{y}z(1+\bar{x})+z(x+\bar{x})+z$

$= xz+yz+\bar{y}z+z \Rightarrow x(x+y)+z(1+\bar{y})$

$= xz+xy+z$ //

(ii) $f(w,x,y,z) = \Pi M(0,2,5,6,7,8,9,11,12)$

$f(w,x,y,z) = \Pi M(0,2,5,6,7,8,9,11,12)$

$= \Sigma m(1,3,4,10,14,15)$

$= m_1+m_3+m_4+m_{10}+m_{13}+m_{14}+m_{15}$

$= \bar{w}\bar{x}\bar{y}z+\bar{w}\bar{x}yz+\bar{w}x\bar{y}\bar{z}+w\bar{x}y\bar{z}+wx\bar{y}z+$
$+wxy\bar{z}+wxyz$

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

$= \bar{w}\bar{x}z+wxz+wy\bar{z}+\bar{w}x\bar{y}\bar{z}$

Decimal NO	Binary number (842)	minterm (m)	Maxterm (M)
8	1000	$A\bar{B}\bar{C}\bar{D}$	$A+B+C+D$
9	1001	$A\bar{B}\bar{C}D$ (m9)	$\bar{A}+B+C+\bar{D}$
10	1010	$A\bar{B}C\bar{D}$ (m	$\bar{A}+B+\bar{C}+D$
11	1011	$A\bar{B}CD$	$\bar{A}+B+\bar{C}+\bar{D}$
12	1100	$AB\bar{C}\bar{D}$	$\bar{A}+\bar{B}+C+D$
13	1101	$AB\bar{C}D$	$\bar{A}+\bar{B}+C+\bar{D}$
14	1110	$ABC\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+D$
15	1111	$ABCD$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$

UNIT-III

Gate-Level Minimization

Map :-

→ The map method gives us a systematic approch for simplyfying a boolean expression. The map method first proposed by veitch and modified by Karnaugh, Hence it is known as the veitch diagram or the karnaugh map (K-map)

One-Variable, two Variable, three Variable and four Variable Maps :-

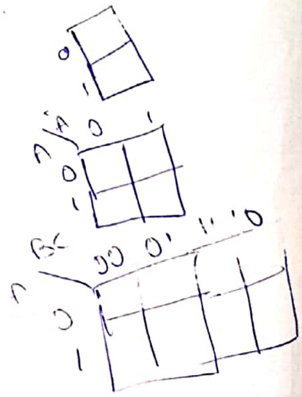
1 Variable $= 2^1 = 2$ cells

2 Variable $= 2^2 = 4$ cells

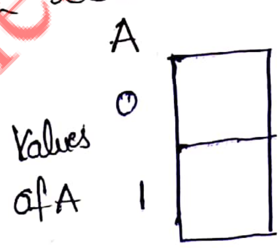
3 Variable $= 2^3 = 8$ cells

4 Variable $= 2^4 = 16$ cells

1 Var $= 2^1 = 2$ cells
2 Var $= 2^2 = 4$ cells
3 Var $= 2^3 = 8$ cells
4 Variable $2^4 = 16$ cells

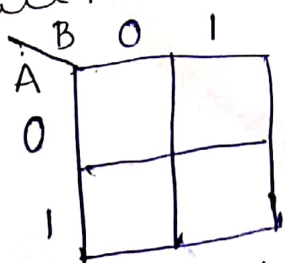


1 Variable K-Map :-



1-Variable Map (2cells)

2-Variable K-Map :-



...variable Map (4cells)

3-Variable K-map:

A \ BC	00	01	11	10
0				
1				

3-Variable K-map (8 cells)

4-Variable K-map:

AB \ CD	00	01	11	10
00				
01				
11				
10				

Values of CD
In gray code

4-Variable K-map (16 cells)

1-Variable:

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

2-Variable:

A \ B		$\bar{B}$	B
0	$\bar{A}$	$\bar{A}\bar{B}$ 0	$\bar{A}B$ 1
1	A	$A\bar{B}$ 2	AB 3

3-Variable:

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

4-Variable

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
$\bar{A}B$	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
AB	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
$A\bar{B}$	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 11	$A\bar{B}C\bar{D}$ 10

Representation of Truth table on k-map

two-Variable k-map

→ The representation of two variable truth table on k-map is given below

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

A \ B	0	1
0	0	1
1	1	0

OR

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

Three-Variable k-map :-

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

A \ BC	00	01	11	10
0	0 (0)	0 (1)	0 (3)	1 (2)
1	1 (4)	1 (5)	1 (7)	0 (6)

OR

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

Four-Variable K-map:

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

AB \ CD	00	01	11	10
00	0000: 1 (0)	0001: 0 (1)	0011: 1 (3)	0010: 0 (2)
01	0100: 1 (4)	0101: 1 (5)	0111: 1 (7)	0110: 0 (6)
11	1100: 1 (12)	1101: 0 (13)	1111: 1 (15)	1110: 1 (14)
10	1000: 0 (8)	1001: 0 (9)	1011: 0 (11)	1010: 1 (10)

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1 (0)	0 (1)	1 (3)	0 (2)
$\bar{A}B$	1 (4)	1 (5)	1 (7)	0 (6)
AB	1 (12)	0 (13)	1 (15)	1 (14)
$A\bar{B}$	0 (8)	0 (9)	0 (11)	1 (10)

Ex:- Plot Boolean Expression $Y = AB\bar{C} + ABC + \bar{A}\bar{B}C$ on the k-map

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

A \ BC	00	01	11	10
0	0 (0)	1 (1)	0 (3)	0 (2)
1	0 (4)	0 (5)	1 (7)	1 (6)

Ex:- Plot Boolean Expression $Y = \bar{A}B\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D} + A\bar{B}CD + AB\bar{C}D$ on the k-map.

$$Y = \bar{A}B\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D} + A\bar{B}CD + AB\bar{C}D$$

$$= 0100 + 1010 + 0110 + 1011 + 1101$$

$$= m_4 + m_{10} + m_6 + m_{11} + m_{13}$$

0100 + 1010 +

AB \ CD	00	01	11	10
00	0000 0	0001 1	0011 3	0010 2
01	0001 1 4	0101 5	0111 7	0110 1 6
11	1100 12	1101 1 13	1111 15	1110 14
10	1000 8	1001 9	1011 1 11	1010 1 10

$\bar{A}B\bar{C}\bar{D}$ $A\bar{B}C\bar{D}$ $AB\bar{C}D$ $A\bar{B}CD$ $A\bar{B}C\bar{D}$

Representation of standard POS on k-map

Three Variable k-map:-

A	B	C	Maxterm (M)
0	0	0	$A+B+C$ (M_0)
0	0	1	$A+B+\bar{C}$ (M_1)
0	1	0	$A+\bar{B}+C$ (M_2)
0	1	1	$A+\bar{B}+\bar{C}$ (M_3)
1	0	0	$\bar{A}+B+C$ (M_4)
1	0	1	$\bar{A}+B+\bar{C}$ (M_5)
1	1	0	$\bar{A}+\bar{B}+C$ (M_6)
1	1	1	$\bar{A}+\bar{B}+\bar{C}$ (M_7)

Foure Variable K-map:-

A	B	C	D	Maxterm (M)
0	0	0	0	$A+B+C+D$ (M_0)
0	0	0	1	$A+B+C+\bar{D}$ (M_1)
0	0	1	0	$A+B+\bar{C}+D$ (M_2)
0	0	1	1	$A+B+\bar{C}+\bar{D}$ (M_3)
0	1	0	0	$A+\bar{B}+C+D$ (M_4)
0	1	0	1	$A+\bar{B}+C+\bar{D}$ (M_5)
0	1	1	0	$A+\bar{B}+\bar{C}+D$ (M_6)
0	1	1	1	$A+\bar{B}+\bar{C}+\bar{D}$ (M_7)
1	0	0	0	$\bar{A}+B+C+D$ (M_8)
1	0	0	1	$\bar{A}+B+C+\bar{D}$ (M_9)
1	0	1	0	$\bar{A}+B+\bar{C}+D$ (M_{10})
1	0	1	1	$\bar{A}+B+\bar{C}+\bar{D}$ (M_{11})
1	1	0	0	$\bar{A}+\bar{B}+C+D$ (M_{12})
1	1	0	1	$\bar{A}+\bar{B}+C+\bar{D}$ (M_{13})
1	1	1	0	$\bar{A}+\bar{B}+\bar{C}+D$ (M_{14})
1	1	1	1	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$ (M_{15})

Ex:- plot Boolean Expression $Y=(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)(A+B+\bar{C})$ on the K-map.

$Y = M_2 \cdot M_3 \cdot M_6 \cdot M_1$

[Minterm ki "1" represent, maxter "0" represent]

A \ BC	00	01	11	10
0	000 0	001 1 0	011 3 0	010 2 0
1	100 4	101 5	111 7	110 6 0

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

$A+\bar{B}+C$

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

$A+B+\bar{C}$

Ex-3 plot Boolean Expression $y = (A+B+C+\bar{D})(A+\bar{B}+\bar{C}+D)(A+B+\bar{C}+\bar{D})$
$(\bar{A}+\bar{B}+C+\bar{D})(\bar{A}+\bar{B}+\bar{C}+D)$

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

$Y = M_1 \cdot M_6 \cdot M_3 \cdot M_{13} \cdot M_{14}$

AB \ CD	00	01	11	10
00	0000 0	0001 1 0	0011 3 0	0010 2
01	0100 4	0101 5	0111 7	0110 6 0
11	1100 12	1101 13 0	1111 15	1110 14 0
10	1000 8	1001 9	1011 11	1010 10

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

Grouping cells for simplification:

Grouping two adjacent ones (pair)

Ex: $y = \bar{A}\bar{B}C + \bar{A}BC$ $\quad [B+\bar{B}=1]$

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

$= \bar{A}C$

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

$\bar{A}C$

(or)

A	B	C
0	0	1
0	1	1

$\bar{A}C$

Ex-2 $y = \bar{A}BC + ABC = BC$

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

$= BC$

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

BC

Ex: ③ $Y = A\bar{B}\bar{C} + AB\bar{C}$

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

$= A\bar{C}$

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

$A\bar{C}$

Ex: ④ $y = \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D$

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

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

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

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

Ex: ⑤ $Y = \bar{A}\bar{B}C + \bar{A}BC + ABC$

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

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

$= \bar{A}C + BC$

Group A $\rightarrow \bar{A}C$

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

Group2 $\rightarrow BC$

[OR]

A B C

1

Grouping four adjacent ones (Quad)

Ex: ①

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

A

② $Y = A$

Ex:-2

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

CD

(b) $y = CD$

Ex:-(3)

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

BD

(3) $y = BD$

Ex:-(4)

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

$A\bar{D}$

$y = A\bar{D}$

Ex:- 5

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

$\bar{B}\bar{D}$

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

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

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

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

Ans:- $Y = \bar{B}\bar{D}$

Ex:- 6

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

Group 1 → AB

Group 3 → AC

Group 2 ⇒ AD

(A) $Y = AB + AD + AC$

Grouping Eight Adjacent ones (Octet)

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

B

A B C D
0 1 0 0
0 1 0 1
0 1 1 1
0 1 1 0
1 1 0 0
1 1 0 1
1 1 1 1
1 1 1 0
B

(a) $Y = B$

A B C D
0 0 0 1
0 1 0 1
0 0 1 1
0 1 1 1
1 1 0 1

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

D

(b) $Y = D$

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

$\bar{B}$

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

$\bar{D}$

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
1	0	0	0
1	0	0	1
1	0	1	1
1	0	1	0

$\bar{B}$

A	B	C	D
0	0	0	0
0	1	0	0
1	1	0	0
1	0	0	0
0	0	1	0
0	1	1	0
1	1	1	0
1	0	1	0

$\bar{D}$

Simplifications of SOP Expressions

→ From the above discussion we can outline generalized procedure to simplify Boolean Expressions as follows:

1. Plot the K-map and place 1s in those cells corresponding to the 1s in the truth table or sum of product expression. Place 0s in other cells.

2. Check the K-map for adjacent 1s and encircle those 1's which are not adjacent to any other 1s. These are called isolated 1s.

3. Check for those 1's which are adjacent to only one other 1 and encircle such pairs.

4. Check for quads and octets of adjacent 1s even if it contains some 1s that have already bee encircled. while doing this make sure that there are minimum no. of groups.

5. Combaine any pairs necessary to include any 1s that have not yet been grouped.

6. From the simplified Expresion by summing product terms of all groups.

Ex: minimize the Expresion $Y = A\bar{B}C + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$

Solution: $Y = 101 + 001 + 011 + 100 + 000$

$= m_5 + m_1 + m_3 + m_4 + m_0$

$= \Sigma m(5, 3, 4, 0)$

Step 1: The k-map for three Variables and it is ploted according to the given expression.

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

Step 2: There are no isolated 1s

Step 3: 1 in the cell 3 adjacent only to 1 in the cell 1. This pair is combained and refered to as group 1

Step 4: There is no octet, but there is a Quad cells 0, 1, 4 and 5 from a quard. This quad is combained and reffered to as a group.

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

Step 5: All 1's have alredy been grouped

Step 6: Each group generates a term in expression for y. In group 1 B variable is eliminated and in group 2 variables A and C are eliminated and we get

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

$\bar{A}C$

B

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

Ex: minimize the expression

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

$$= 0100 + 0101 + 1100 + 1101 + 1001 + 0010$$

$$= m_4 + m_5 + m_{12} + m_{13} + m_9 + m_2$$

$$= \Sigma m(4, 5, 12, 13, 9, 2)$$

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	1	3	2 (1)
$\bar{A}B$ 01	4 (1)	5 (1)	7	6
AB 11	12 (1)	13 (1)	15	14
$A\bar{B}$ 10	8	9 (1)	11	10

$\bar{A}\bar{B}C\bar{D}$ Group 3

Group 1 $B\bar{C}$

Group 2 $A\bar{C}D$

G1

A	B	C	D
0	1	0	0
0	1	0	1
1	1	0	0
1	1	0	1

$B\bar{C}$

G2

A	B	C	D
1	1	0	1
1	0	0	1

$A\bar{C}D$

G3

ABCD

0010

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

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

* Simplify the logic function specified by the truth table by using the Kanaugh map method. Y is the output variable and A, B and C are the input variables

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

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

$$= m_0 + m_3 + m_4 + m_7$$

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

A \ BC	00	01	11	10
0	0: 1	1	3: 1	2
1	4: 1	5	7: 1	6

Group 1: $\bar{B}\bar{C}$ — Group 2: BC

Group 1

A	B	C
0	0	0
1	0	0

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

Group 2

A	B	C
0	1	1
1	1	1

BC

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

Ex: Reduce the following function using K-map technique.

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

$$f(A,B,C,D) = \Sigma m(0,1,4,8,9,10)$$

$$= m_0 + m_1 + m_4 + m_8 + m_9 + m_{10}$$

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

AB \ CD	00	01	11	10
00	0 (1)	1 (1)	3	2
01	4 (1)	5	7	6
11	12	13	15	14
10	8 (1)	9 (1)	11	10 (1)

Group 2, Group 3, Group 1

Group 1

A	B	C	D
1	0	0	0
1	0	1	0

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

Group 2

A	B	C	D
0	0	0	0
0	0	0	1
1	0	0	0
1	0	0	1

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

Group 3

A	B	C	D
0	1	0	0

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

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

Product of Sum simplification:

1. plot the k-map and place 0s in those cells corresponding to the 0s in the truth table or maxterms in the product of sum expression.
2. check the k-map for adjacent 0s and Encircle those 0s which are not adjacent to any other 0s. These are called isolated 0s.
3. Check for those 0s which are adjacent to only one other 0 and Encircle such pairs.
4. check for quads and octets of adjacent 0s even if it contains some 0s that have already been Encircled. while doing this make sure that there are minimum no. of groups.
5. Combine any pairs necessary to include any 0s that have not yet been grouped.
6. From the simplified pos Expression for F by taking product of sum terms of all the groups

To get familiar with these steps we will solve some Examples.

Ex: minimize the Expression

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

$= \cancel{(001)(011)}$

$= (A+B+\bar{C}) = M_1$, $(A+\bar{B}+\bar{C}) = M_3$, $(\bar{A}+\bar{B}+\bar{C}) = M_7$

$(\bar{A}+B+C) = M_4$, $(A+B+C) = M_0$

Step 1: (a) shows the k-map for three variable and it is plotted according to given maxterms

A \ BC	00	01	11	10
0	0 (0)	0 (1)	0 (3)	(2)
1	0 (4)	(5)	0 (7)	(6)

Step 2: three are no isolated 0s

Step 3: 0 in the cell 4 is adjacent only to 0 in the cell 0 and 0 in cell 7 is adjacent only to 0 in the cell 3. These two pairs are combined and refered to as group 1 and group 2 respectively.

A \ BC	00	01	11	10
0	0 (0)	0 (1)	0 (3)	(2)
1	0 (4)	(5)	0 (7)	(6)

Step 4: There are no quads and octets

Step 5: The 0 in the cell 1 can be combined with 0 in the cell 3 to be from a pair. This pair is referred to as group 3

A \ BC	00	01	11	10
0	0 (0)	0 (1)	0 (3)	(2)
1	0 (4)	(5)	0 (7)	(6)

Step 6: In group 1 and in group 2, A is eliminated, where in group 3 variable B is eliminated and we get :

Group 1	Group 2	Group 3
A B C	A B C	A B C
0 0 0	0 1 1	0 0 1
1 0 0	1 1 1	0 1 1
$\bar{B}\bar{C}$	BC	$\bar{A}C$

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

$$\bar{\bar{Y}} = Y = \overline{\bar{B}\bar{C} + BC + \bar{A}C} \quad \text{[According to Demorgans]}$$

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

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

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

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

Ex:- Minimize the following expression in the pos form

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

Sol:- $(\bar{A}+\bar{B}+C+D) = M_{12}$, $(\bar{A}+\bar{B}+\bar{C}+D) = M_{14}$, $(\bar{A}+\bar{B}+\bar{C}+\bar{D}) = M_{15}$

$(\bar{A}+B+C+D) = M_8$, $(A+\bar{B}+\bar{C}+D) = M_6$, $(A+\bar{B}+\bar{C}+\bar{D}) = M_7$

$(A+B+C+D) = M_0$ and $(\bar{A}+\bar{B}+C+\bar{D}) = M_{13}$

AB \ CD	$C+D$ 00	$C+\bar{D}$ 01	$\bar{C}+\bar{D}$ 11	$\bar{C}+D$ 10
$A+B$ 00	0: 0	1	3	2
$A+\bar{B}$ 01	4	5	7: 0	6: 0
$\bar{A}+\bar{B}$ 11	12: 0	13: 0	15: 0	14: 0
$\bar{A}+B$ 10	8: 0	9	11	10

Step 1: shows the k-map for four variable and it plotted according to given max terms.

Step 2: There are no isolated 1's

Step 3: 0 in the cell 0 is adjacent only to in the cell this pair is combained and refferd to as group 1

Step 4: There are two quads cells 12, 13, 14 and 15 form a quad 1 and cells 6, 7, 14, 15 forms a quad 2. These two quads are reffered to as group 2 and group 3, respectively

Step 5: All 0s have already been grouped

Step 6:- In group 1, variable A is eliminated in group 2, variable C and D are eliminated and in group 3 variable A and D are eliminated.

∴ we get simplified pos expression

AB \ CD	00	01	11	10
00	0: 0	1	3	2
01	4	5	7: 0	6: 0
11	12: 0	13: 0	15: 0	14: 0
10	8: 0	9	11	10

Group 1 — Group 2 — Group 3

Group 1

A	B	C	D
0	0	0	0
1	0	0	0

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

Group 2

A	B	C	D
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0

A B

Group 3

A	B	C	D
0	1	1	1
0	1	1	0
1	1	1	1
1	1	1	0

B C

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

$$y = \bar{\bar{Y}} = \overline{\bar{B}\bar{C}\bar{D} + AB + BC}$$

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

Incompletely specified functions (Don't Care terms (or) conditions)

Ex:

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

Sol:

A \ BC	00	01	11	10
0	^{0}	^{1} 1	^{3} 1	^{2}
1	^{4}	^{5} 1	^{7} X	^{6} X

Don't Care Conditions

A \ BC	00	01	11	10
0	^{0}	^{1} 1	^{3} 1	^{2}
1	^{4}	^{5} 1	^{7} 1	^{6} 0

Group 1

$$\boxed{\therefore Y = C}$$

Group 1

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

Describing Incomplete Boolean function

Ex: (i) $f(A,B,C) = \Sigma m(0,2,4) + d(1,5)$

(ii) $f(A,B,C) = \Pi M(2,5,7) + d(1,3)$

(i) $f(A,B,C) = \Sigma m(0,2,4) + d(1,5)$

A \ BC	00	01	11	10
0	0: 1	1: X	3	2: 1
1	4: 1	5: X	7	6

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

Group 1, Group 2, Group 3

G_1

A	B	C
0	0	0
1	0	0

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

G_2

A	B	C
0	0	0
0	0	1

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

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

Group 1

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

$\bar{B}$

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

(ii) $f(A,B,C) = \Pi M(2,5,7) + d(1,3)$

A \ BC	00	01	11	10
0	0	1: X	3: X	2: 0
1	4	5: 0	7: 0	6

A \ BC	00	01	11	10
0	0	1: 0	3: 0	2: 0
1	4	5: 0	7: 0	6

Group 1, Group 2, Group 3

G_1

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

BC

G_2

A	B	C
1	0	1
1	1	1

AC

G_3

A	B	C
0	1	0

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

G_1

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

C

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

Don't care conditions in logic Design

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

$P = \Sigma m(1,2,4,7,8) + d(10,11,12,13,14,15)$

Ex: find the reduced SOP form of the following function.

$f(A,B,C,D) = \Sigma m(1,3,7,11,15) + \Sigma d(0,2,4)$

$f(A,B,C,D) = \Sigma m(1,3,7,11,15) + \Sigma d(0,2,4)$

AB \ CD	00	01	11	10
00	X (0)	1 (1)	1 (3)	X (2)
01	X (4)	(5)	1 (7)	(6)
11	(12)	(13)	1 (15)	(14)
10	(8)	(9)	1 (11)	(10)

Group2

AB \ CD	00	01	11	10
00	1	1	1	1
01	0		1	
11			1	
10			1	

Group1

G_1

A	B	C	D
0	0	1	1
0	1	1	1
1	1	1	1
1	0	1	1

CD

G_2

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0

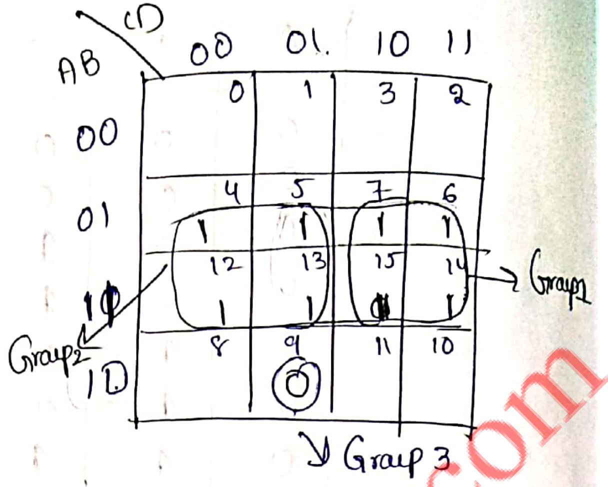
$\bar{A}\bar{B}$

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

Ex: Reduce the following function using K-map technique

$f(A,B,C,D) = \Sigma m(5,6,7,12,13) + \Sigma d(4,9,14,15)$

AB \ CD	00	01	11	10
00	0	1	3	2
01	4 X	5 1	7 1	6 1
11	12 1	13 1	15 X	14 X
10	8	9 X	11	10



G1

A	B	C	D
0	1	0	0
0	1	0	1
1	1	0	0
1	1	0	1

$B\bar{C}$

G2

A	B	C	D
0	1	1	0
0	1	1	1
1	1	1	0
1	1	1	1

BC

A	B	C	D
1	0	0	1

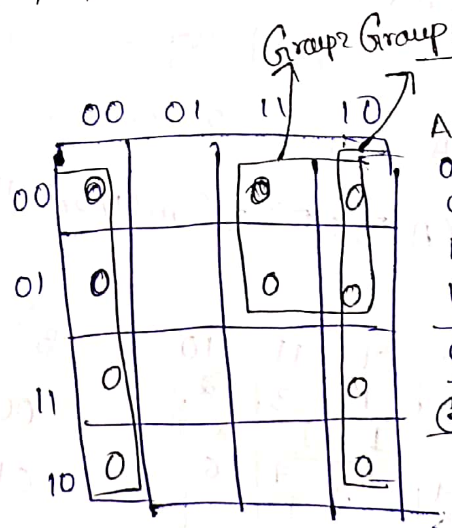
$A\bar{B}\bar{C}D$

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

Ex: Reduce the following function by using K-map technique

$f(A,B,C,D) = \Pi M(0,3,4,7,8,10,12,14) + d(2,6)$

AB \ CD	00	01	11	10
00	0 0	1	0 3	X 2
01	0 4	5	0 7	X 6
11	0 12	13	15	0 14
10	0 8	9	11	0 10



A	B	C	D
0	0	1	0
0	1	1	0
1	1	1	0
1	0	1	0
0	0	0	0
0	1	0	0
1	1	0	0
1	0	0	0

$\bar{D}$

G2

A	B	C	D
0	0	1	1
0	0	1	0
0	1	1	1
0	1	1	0

$\bar{A}C$

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

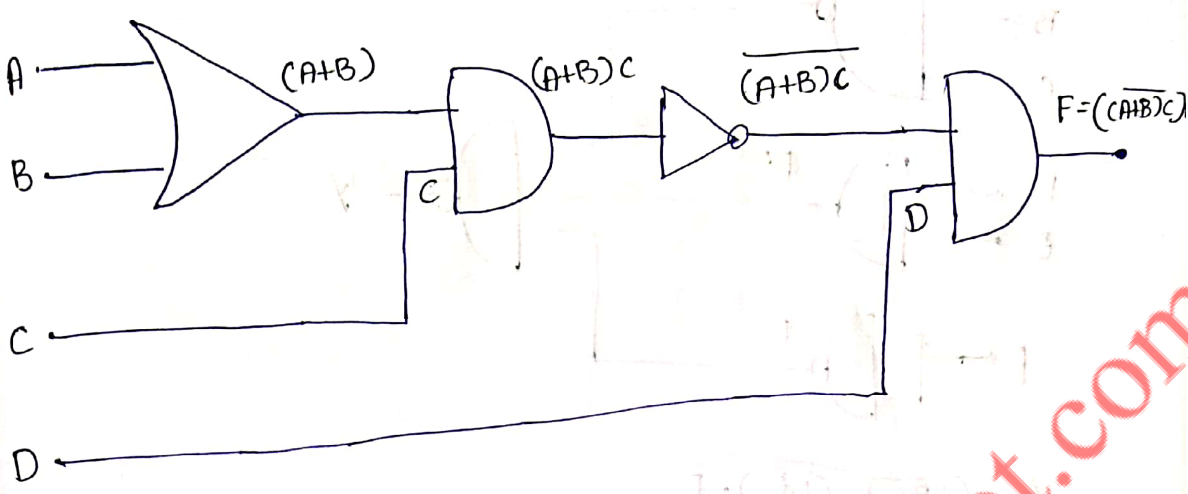
$Y = \bar{\bar{Y}} = \overline{\bar{D} + \bar{A}C}$

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

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

* ~~f(A+B, C+D) =~~

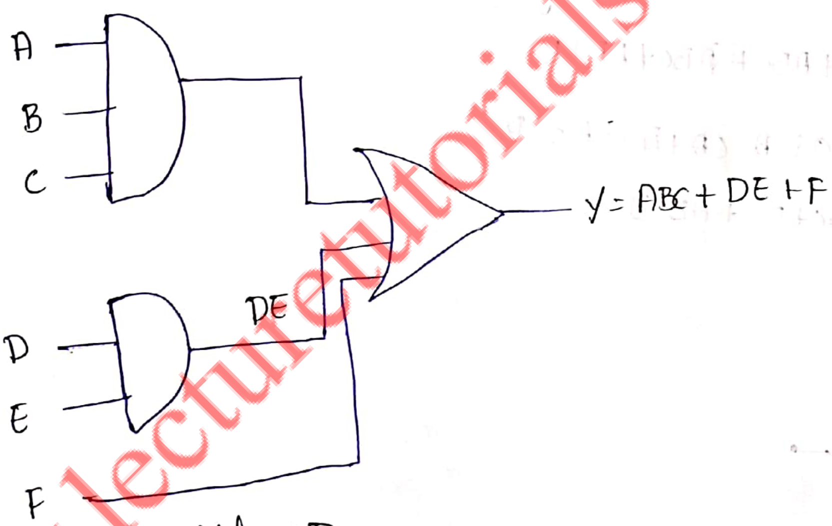
Ex:- $F = (\overline{(A+B)C})D$ to Implement logic Design?



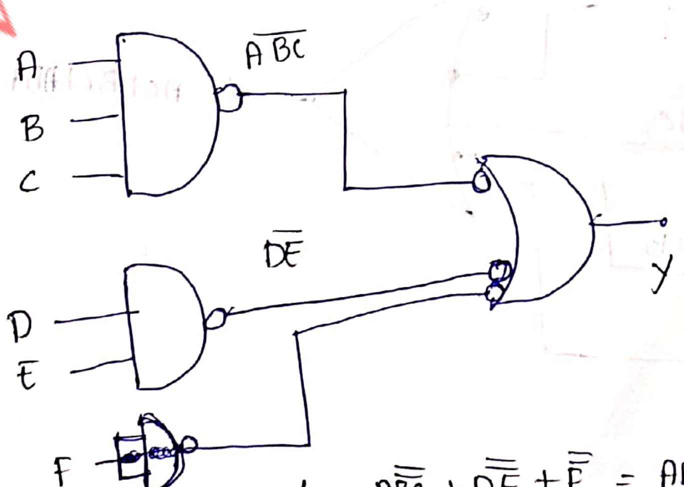
NAND to NAND Implementation:-

$y = ABC + DE + F$

(i) AND - OR

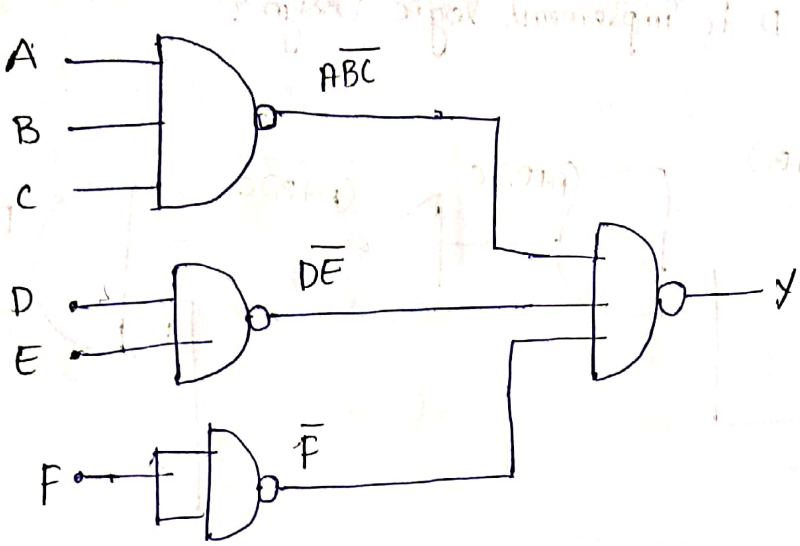


(ii) NAND - Bubble OR



$y = \overline{\overline{ABC}} + \overline{\overline{DE}} + \overline{\overline{F}} = ABC + DE + F$

(iii) NAND – NAND

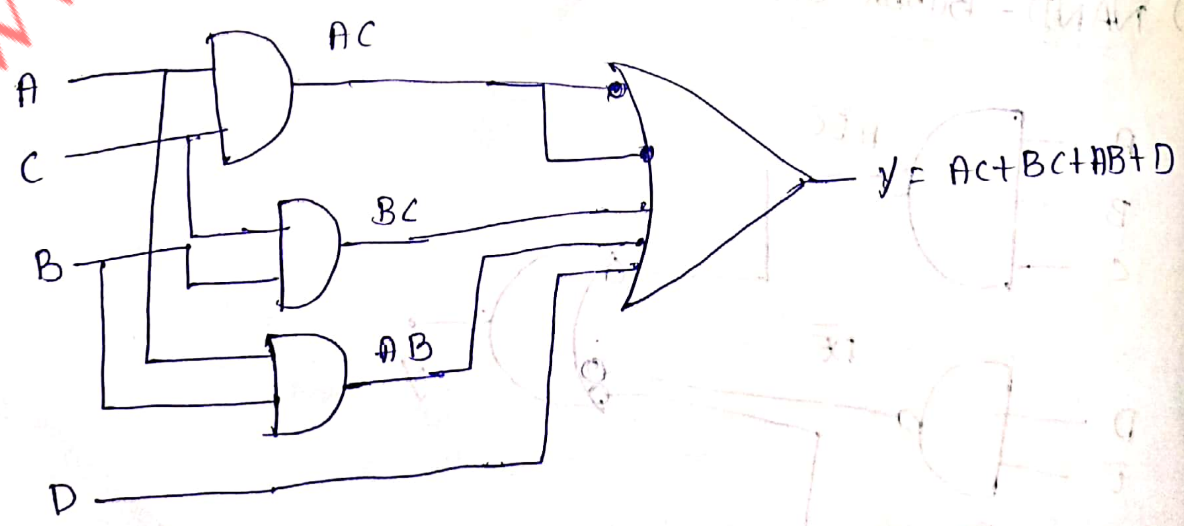


$$y = \overline{(\overline{ABC}) \cdot (\overline{DE}) \cdot \overline{F}}$$
$$= (\overline{\overline{ABC}}) + (\overline{\overline{DE}}) + \overline{\overline{F}}$$
$$= ABC + DE + F$$

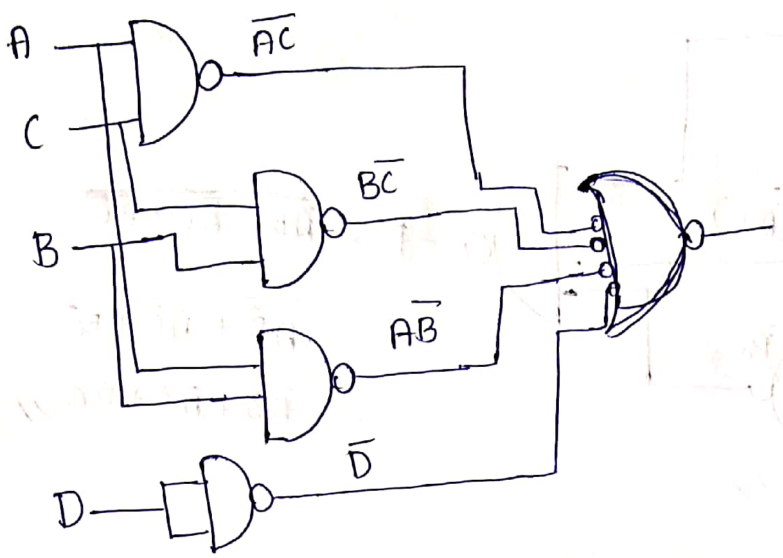
Ex: $Y = AC + ABC + \overline{A}BC + AB + D$ Implement the following Boolean function with NANA to NAND logic.

$$Y = AC + ABC + \overline{A}BC + AB + D$$
$$= AC + BC(A + \overline{A}) + AB + D$$
$$Y = AC + BC + AB + D$$

(i) AND – OR



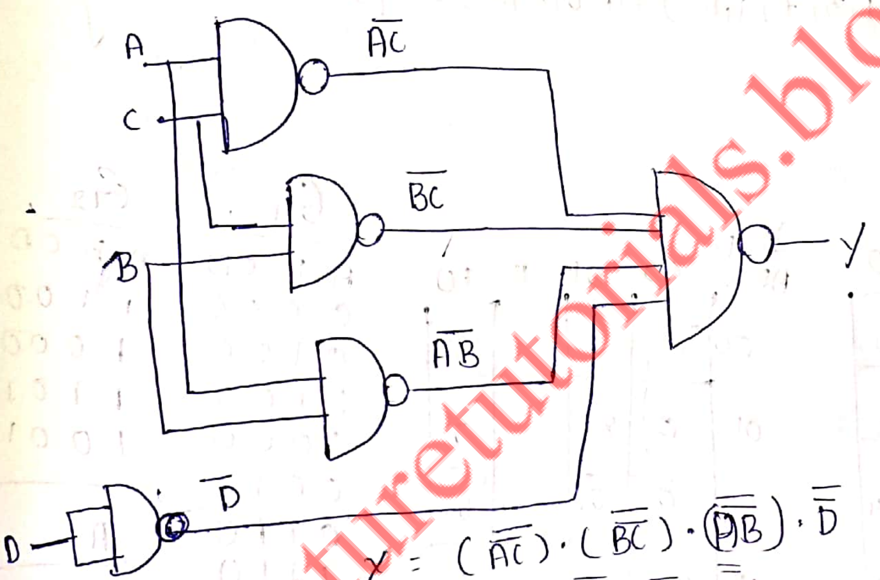
(ii) NAND - Bubble OR



$Y = \overline{\overline{AC}} + \overline{\overline{BC}} + \overline{\overline{AB}} + \overline{\overline{D}}$

$= AC + BC + AB + D$

(iii) NAND - NAND



$Y = \overline{(\overline{AC})\cdot(\overline{BC})\cdot(\overline{AB})\cdot\overline{D}}$

$= \overline{\overline{AC}} + \overline{\overline{BC}} + \overline{\overline{AB}} + \overline{\overline{D}}$

$= AC + BC + AB + D$

Ex: Implementation the following Boolean Expresion with NAND-NAND logic.

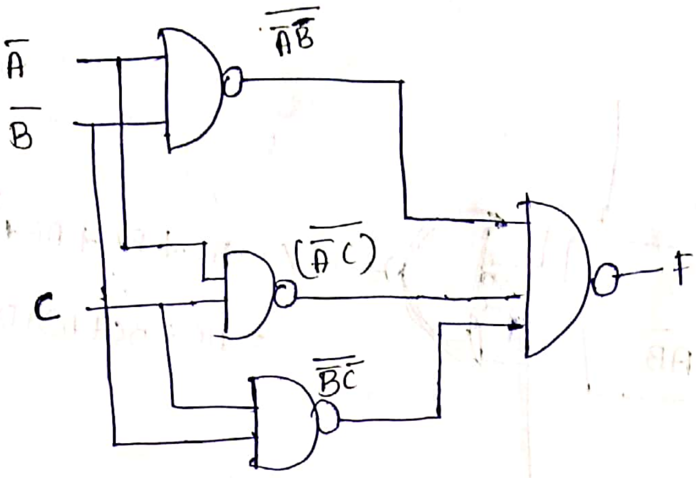
$F(A,B,C) = \Sigma m(0,1,3,5)$

A \ BC	00	01	11	10
0	1 (0)	1 (1)	1 (3)	(2)
1	(4)	1 (5)	(7)	(6)

G1, G2, G3

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

NAND - NAND



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

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

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

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

Ex:① Find the reduced pos form of the following Equation $F(A,B,C,D) = \Sigma m(1,3,7,11,15) + d(0,2,5)$ Implement and using NAND logic?

AB \ CD	00	01	11	10
00	X 0	1 1	1 3	X 2
01	0 4	X 5	1 7	0 6
11	0 12	0 13	1 15	0 14
10	0 8	0 9	1 11	0 10

AB \ CD	00	01	11	10
00	0			0
01	0	X		0
11	0	0		0
10	0	0		0

Group2 Group1

G_1

A	B	C	D
0	0	0	0
0	1	0	0
1	1	0	0
1	0	0	0
0	0	1	0
0	1	1	0
1	1	1	0
1	0	1	0

$\bar{D}$

G_2

A	B	C	D
1	1	0	0
1	0	0	0
1	1	0	1
1	0	0	1

$A\bar{C}$

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

$Y = \bar{\bar{Y}} = (\bar{\bar{D}})(\overline{A\bar{C}})$

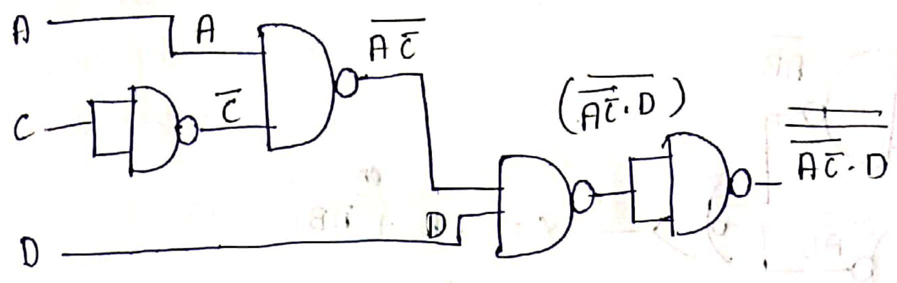
$\therefore Y = D \cdot (\bar{A} + C)$

Ex:② ~~using k-map, Determine the minimal sop Expression~~ and ~~resize the simplified Expression using~~ NAND logic?

~~$f(w,x,y,z) = \Pi M(0,2,3,7,8,$~~$9,10)$

Sol: ~~$f(w,x,y,z) = \Sigma m(1,4,5,6,11,$~~$12,13,14,15)$

$\therefore Y = D\cdot(\bar{A}+C)$



$Y = (\overline{A\bar{C}}\cdot D)$

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

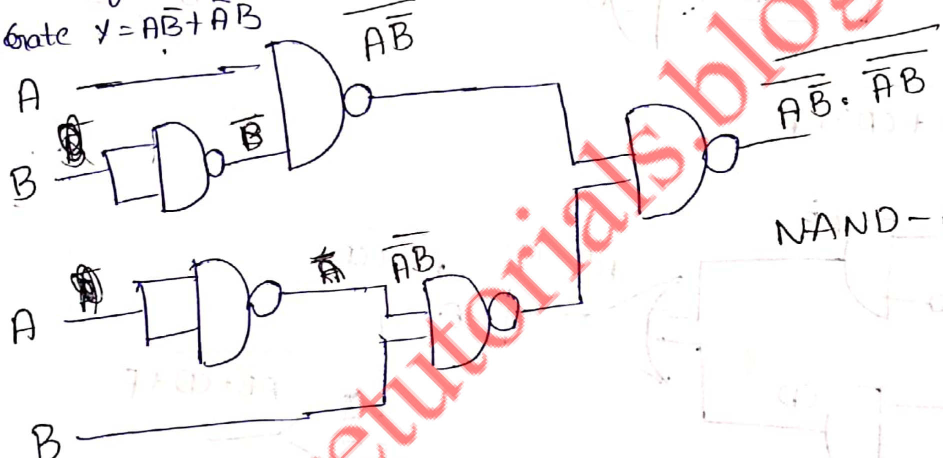
$Y = (\bar{A}+C)\cdot D$ //

Ex:- To Implement X-OR gate by using NAND-NAND logic gate

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

The X-OR gate function is two Inputs and one output. The i/p variables are A and B. The o/p variable Y

X-OR Gate $Y = A\bar{B}+\bar{A}B$

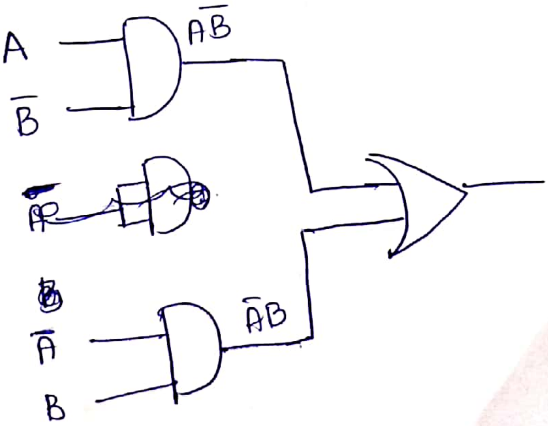


NAND-NAND

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

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

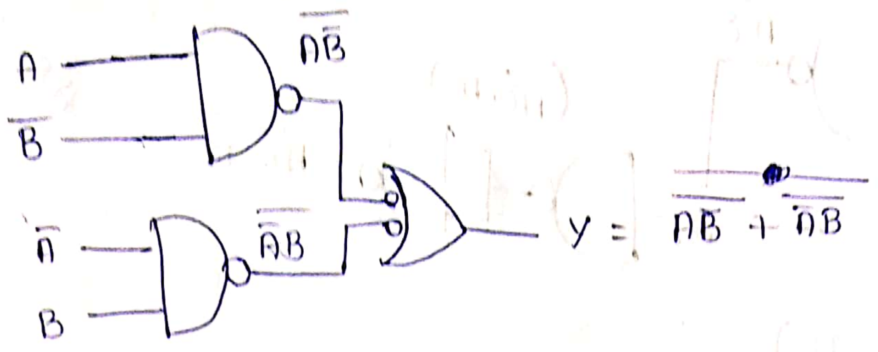
(9) AND-OR



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

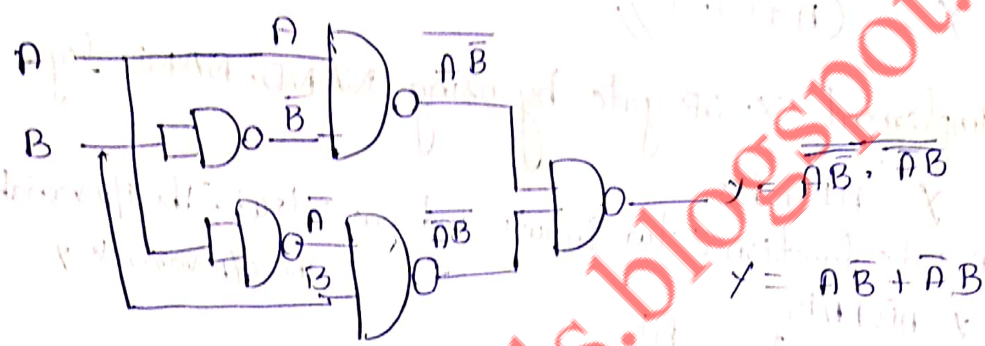
(ii) NAND - Bubbled OR

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

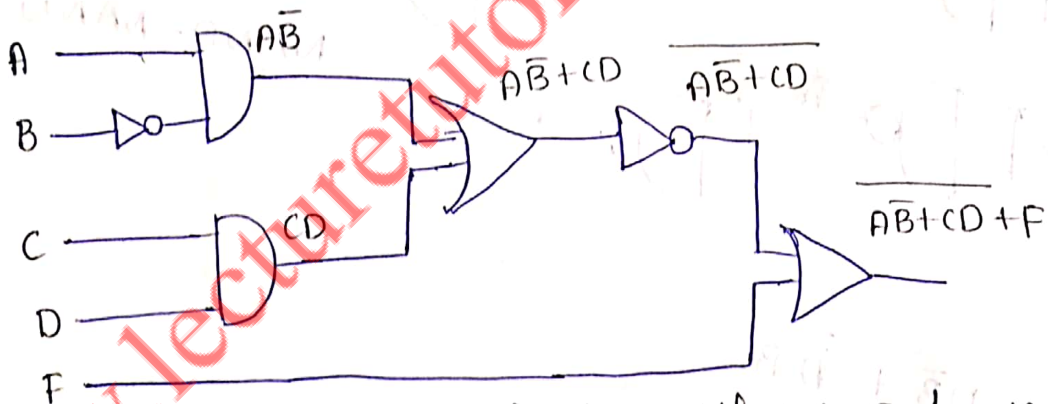


$Y = A\bar{B} + \bar{A}B$

(iii) NAND - NAND



Ex: $\overline{(A\bar{B} + CD)} + F$



Ex: Implement the following function with NAND to NAND logic

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

$Y = AC + BC + AB + D$

NOR- NOR Implementation:-

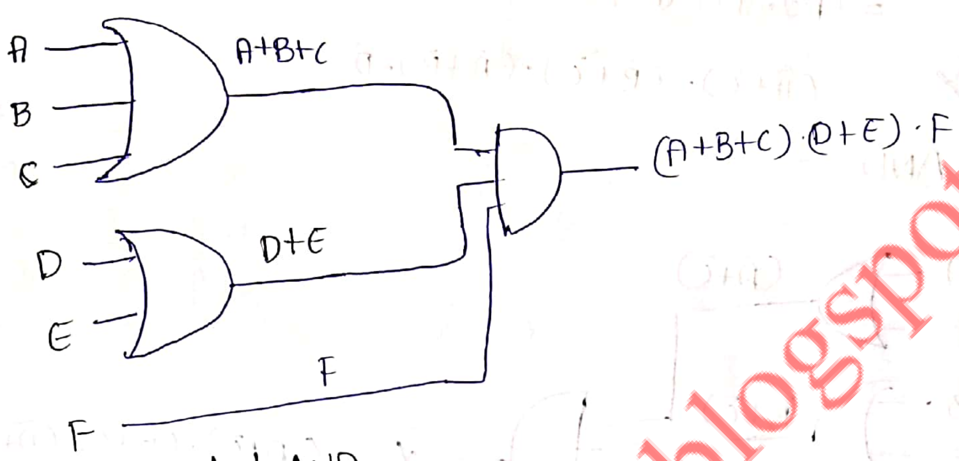
(i) OR – AND

(ii) NOR- Bubbled AND

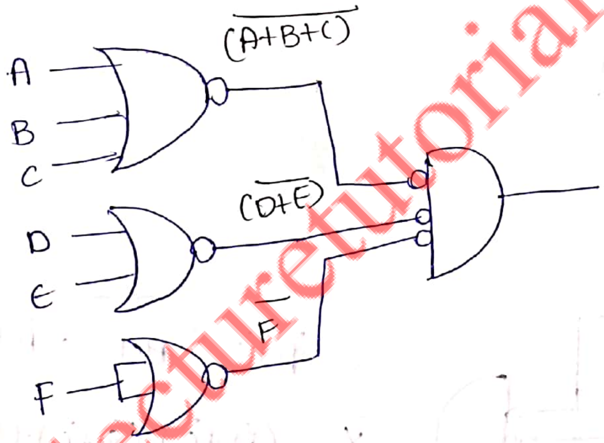
(iii) NOR- NOR

Ex:- $Y = (A+B+C)\cdot(D+E)\cdot F$

i) OR- AND



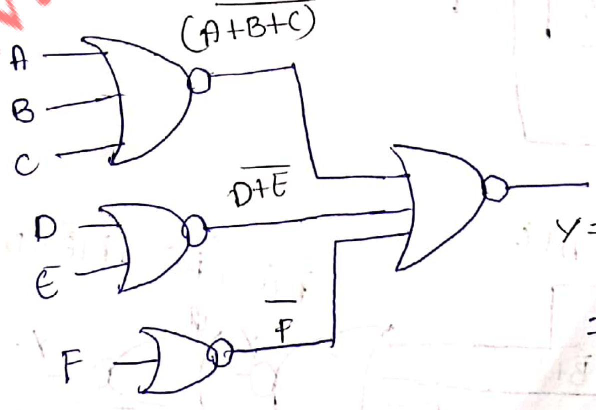
(ii) NOR- Bubbled AND



$Y = \overline{\overline{(A+B+C)}} \cdot \overline{\overline{(D+E)}} \cdot \overline{\overline{F}}$

$= (A+B+C)\,(D+E)\cdot F$

(iii) NOR- NOR



$Y = \overline{\overline{(A+B+C)} + \overline{(D+E)} + \overline{F}}$

$= (A+B+C)\cdot(D+E)\cdot F$

$[\because \overline{A+B} = \overline{A}\cdot\overline{B}$

$\overline{\overline{A}} = A]$

Ex:- Implement the following Boolean function with NOR-NOR logic

$$Y = AC + BC + AB + D$$

The given Boolean Expression

$$Y = AC+BC+AB+D$$

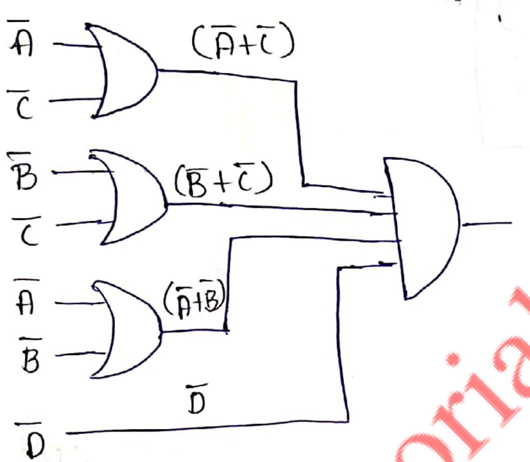
Assumption, By using Duality theorem

$$\overline{Y} = \overline{AC+BC+AB+D}$$

$$= (\overline{AC})\cdot(\overline{BC})\cdot(\overline{AB})\,\overline{D}$$

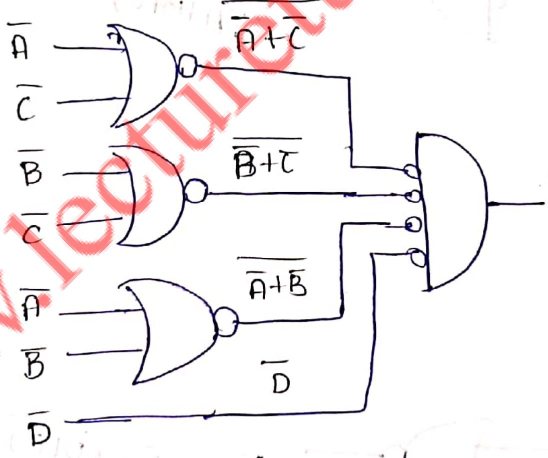
$$\overline{Y} = (\bar{A}+\bar{C})\cdot(\bar{B}+\bar{C})\cdot(\bar{A}+\bar{B})\cdot\bar{D}$$

(i) OR-AND



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

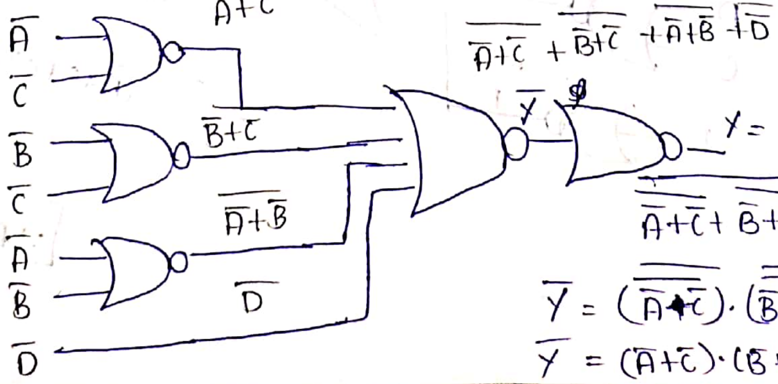
(ii) NOR- Bubbled-AND



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

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

(iii) NOR-NOR



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

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

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

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

$$\overline{\overline{Y}} = \overline{(\bar{A}+\bar{C}).(\bar{B}+\bar{C}).(\bar{A}+B).\bar{D}} \qquad [\because \overline{\bar{A}+\bar{B}} = \bar{\bar{A}}\cdot\bar{\bar{B}} = AB]$$

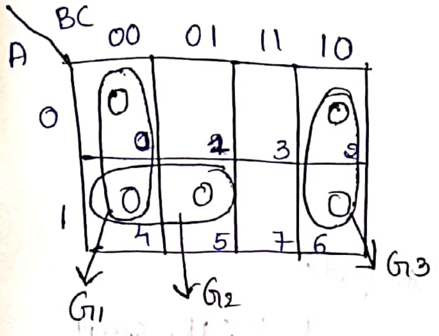
$$Y = AC+BC+AB+D$$

Ex:- Implement the following Boolean function with NOR-NOR logic

$$f(A,B,C) = \Pi_M(0,2,4,5,6)$$

Ex:- Implement the X-NOR gate by using only NOR gate $(AB+\bar{A}\bar{B})$

Ex:-① $f(A,B,C) = \Pi_M(0,2,4,5,6)$



G1

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

G2

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

G3

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

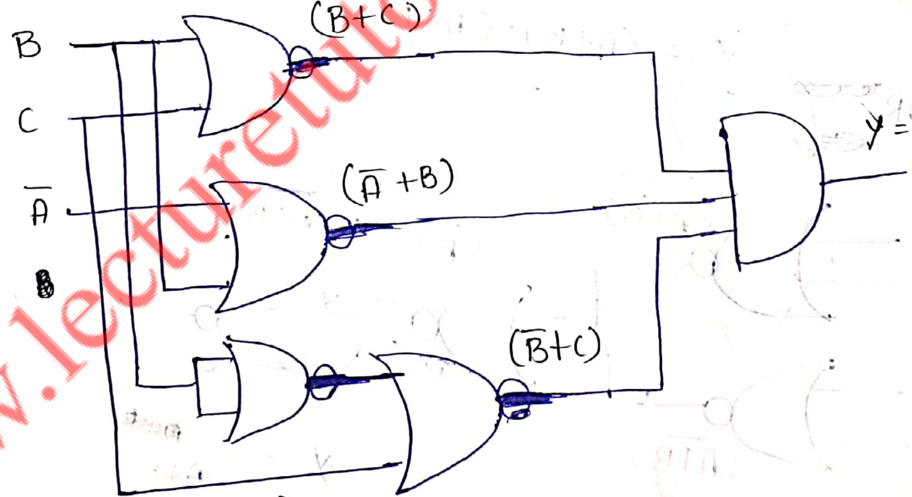
$$\bar{Y} = \bar{B}\bar{C} + A\bar{B} + B\bar{C}$$

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

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

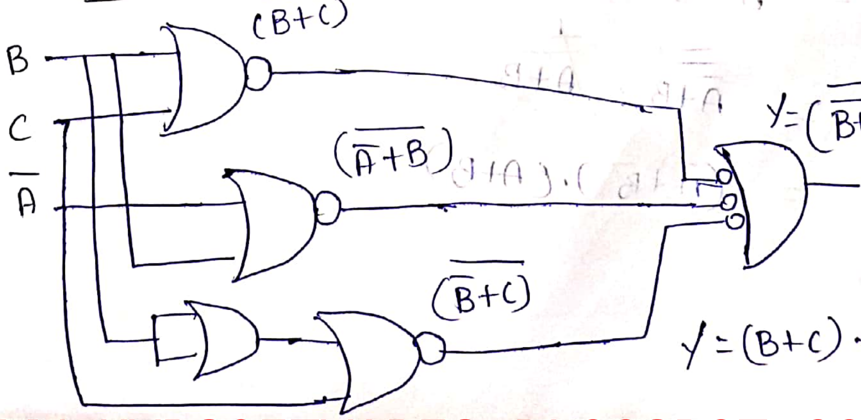
$$Y = (B+C).(\bar{A}+B).(\bar{B}+C)$$

(i) OR - AND



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

(ii) NOR - Bubbled AND

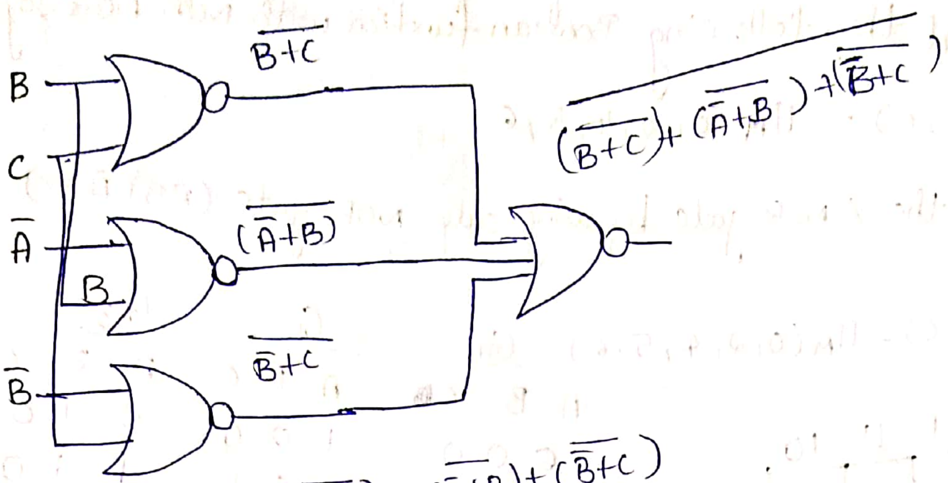


$Y = (\overline{\overline{B+C}})\cdot(\overline{\overline{\bar{A}+B}})\cdot(\overline{\overline{\bar{B}+C}})$

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

(iii) NOR - NOR

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



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

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

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

② Implement the X-NOR gate by using only NOR gate. The X-OR gate function is two inputs and one output. The input variables are A and B and the output variable is y

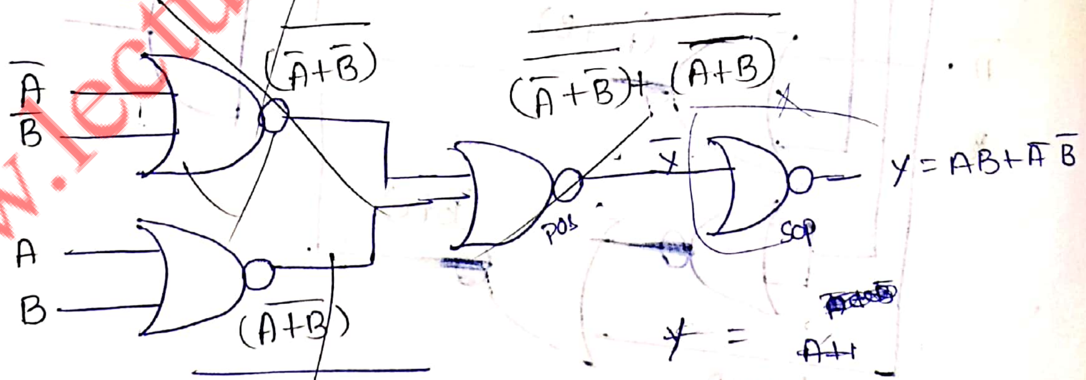
$y = AB + \bar{A}\bar{B}$

NOR - NOR

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

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

NOR - NOR



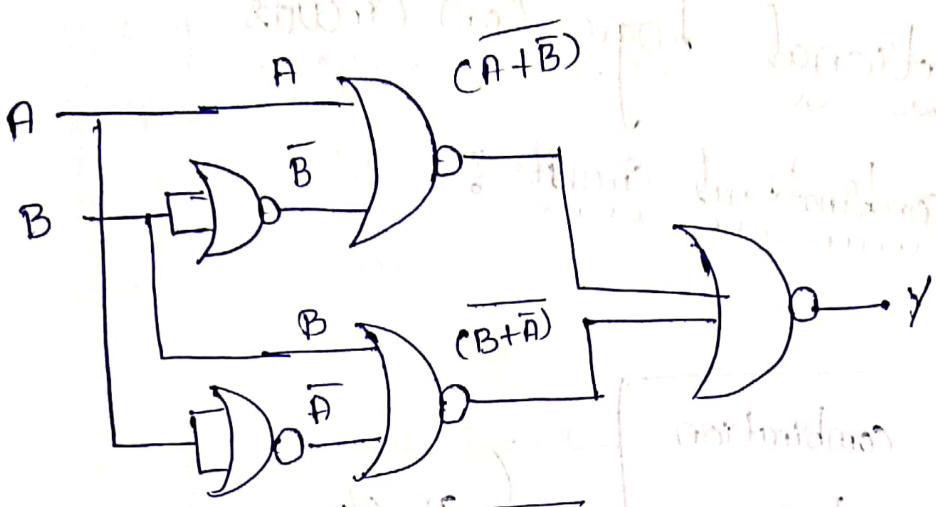
$y = \overline{A+}$

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

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

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

Ex: Implement the x-



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

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

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

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

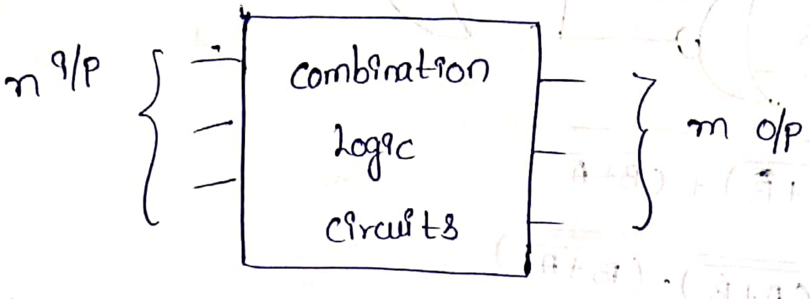
$$= AB + \overline{A}\,\overline{B}$$

$$\left[\because B\overline{B} = 0,\quad A\overline{A} = 0 \right]$$

UNIT-4

Combinational Logic (or) Circuits

Block diagram of combinational circuits:



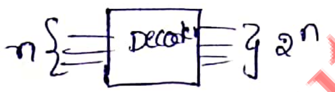
Topics:

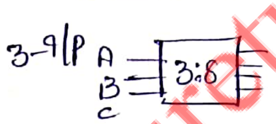
1. Half-Adder — 2-i/p — 2 o/p
2. Half-sub — 2-i/p — 2 o/p
3. Full-adder — 3-i/p — 2 o/p
4. Full-sub — 3-i/p — 2 o/p
5. Decoder — n-i/p — 2^n o/p

Ex: 2-i/p - 2^2 = 4 o/p

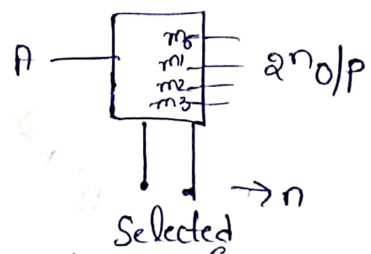
Ex: 3-i/p - 2^3 = 8 o/p

3:8 Decoder





6. Encode — 2^n i/p — n o/p



n: 2^n - o/p

7. multiplex - MUX - n selection lines 2^n

n: 2^n - i/p

8. De-multiplex - De-Mux
9. priority Encode

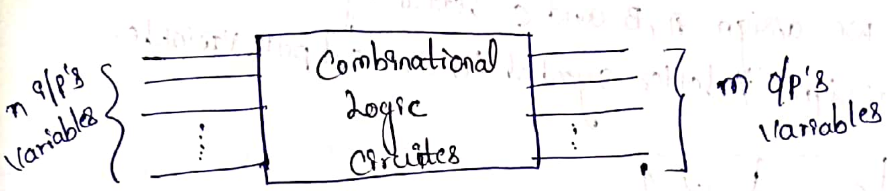
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Combinational Circuits

→ When logic gates are connected together to produce a specified output for certain specified combinations of input variables, with no storage involved. The remaining & resulting circuit is called as combinational logic circuit.

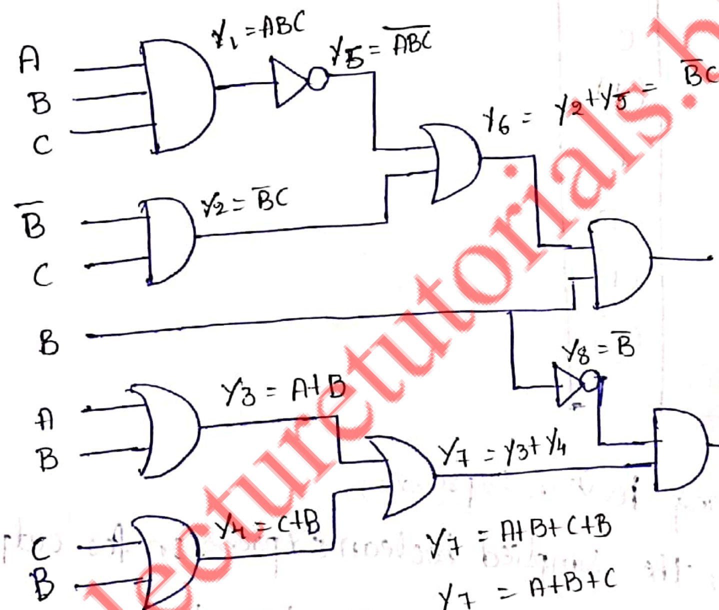
→ In combinational logic the output variables are at all times dependence on the combination of input variables.

→ The Block diagram of input via combination circuit is given below



Analysis Processor:-

Ex:- To obtain the boolean function for output of the given circuit



$F_1 = Y_6 \cdot B = (\overline{B}C + \overline{ABC})B$

$= B\overline{B}C + B(\overline{ABC})$

$= 0 + B(\overline{A} + \overline{B} + \overline{C})$

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

$F_2 = Y_8 \cdot Y_7$

$Y_7 = A+B+C+B$

$Y_7 = A+B+C$

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

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

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

$\therefore F_1 = B(\overline{A} + \overline{C})$, $F_2 = \overline{B}(A+C)$

ABC	$\overline{B}$	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	F_1	F_2
0 0 0	1	0	0	0	0	1	1	0	1	0	0
0 0 1	1	0	1	0	1	1	1	1	1	0	1
0 1 0	0	0	0	1	1	1	1	1	0	1	0
0 1 1	0	0	0	1	1	1	1	1	0	1	0
1 0 0	1	0	0	1	0	1	1	1	1	0	1
1 0 1	1	0	1	1	1	1	1	1	1	0	1
1 1 0	0	0	0	1	1	1	1	1	0	1	0
1 1 1	0	1	0	1	1	0	0	1	0	0	0

Design procedure:

Example

* Design a combinational logic circuit with three input variables that will produce a logic one output when more than one input variables are logic one.

Step 1: Derive the truth table for given statement

The given problem specifies that there are three input variables and only one output variable. we assign A, B and C letter symbols to three input variables and assign "Y" letter symbol to one output variable.

Ste

A	B	C	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

Step 2: obtain simplifying boolean Expression

Now we can obtain the simplied boolean Expression for output for out variable Y for using K-map simplification

	00	01	11	10
0	0	1	1 (3)	2
1	4	1 (5)	1 (7)	1 (6)

G1, G2, G3

G1

A	B	C
0	1	1
1	1	1

BC

G2

A	B	C
1	0	1
1	1	1

AC

G3

A	B	C
1	1	1
1	1	0

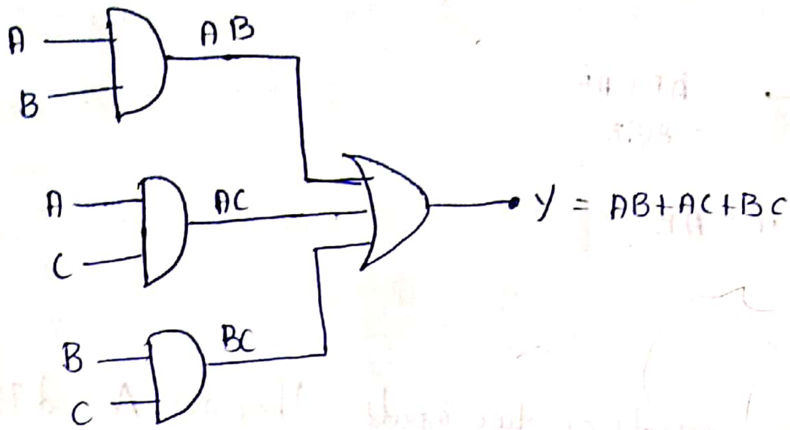
AB

$Y = BC + AC + AB$

Step 3: Draw the logic diagram

we are going various combinational circute to the above boolean expression

Logic Diagram



Design of Adders

→ Digital computers perform various Arthamatic operation is the addition of two binary digits. This simple addition consists of four elementary operations, namely

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=(10)_2$

) Half Adder

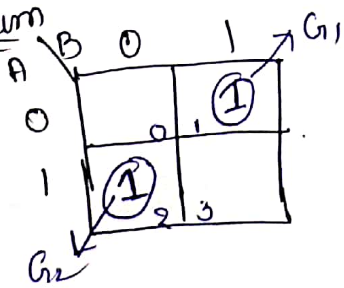
→ The Half adder consists of two input variables and two output variables ~~to~~ we assign A and B are input variables and sum (S) and carry (C).

A, B = i/p variables
S, C = o/p variables

Truth table

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

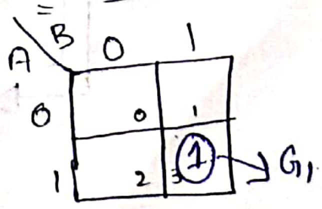
K-map for Sum



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

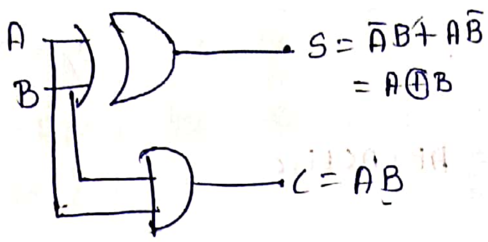
$= A \oplus B$

K-map for carry



$C = AB$

logic Diagram for S



(ii) Half - subtraction :-

→ The Half subtraction consists of two inputs they are A and B and two outputs they are difference (D) and barrow (B)

→ The truth table

A	B	D (Difference)	B (Barrow)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map for D

A \ B	0	1
0	0	1 1
1	1 2	3

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

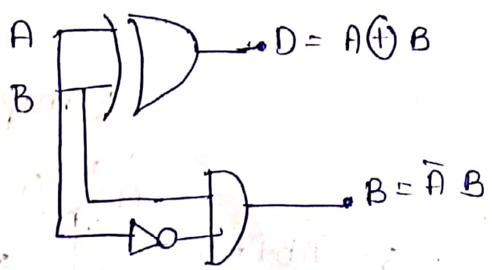
$= A \oplus B$

K-map for B

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	0	1 1
A 1	2	3

$B = \bar{A}B$

Logic Diagram



(iii) Full-Adder :-

→ In full-Adder consists of three inputs they are A, B, C and outputs are sum and carry

A	B	C	S (sum)	C (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

K-map for S

$S = A \oplus B \oplus C$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	(0)	1 (1)	(3)	1 (2)
A 1	1 (4)	(5)	1 (7)	(6)

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

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

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

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

$S = A \oplus B \oplus C$

K-map for C

$C = AB + BC + CA$

A \ BC	00	01	11	10
0	(0)	(1)	1 (3)	(2)
1	(4)	1 (5)	1 (7)	1 (6)

G_1, G_2, G_3

G_1	G_2	G_3
0 1 1	1 1 1	1 0 1
1 1 1	1 1 0	1 1 1
BC	AB	AC

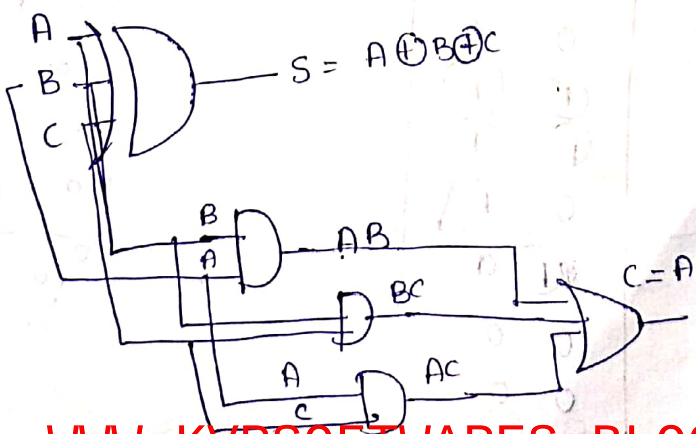
$C = AB + BC + CA$

(or)

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

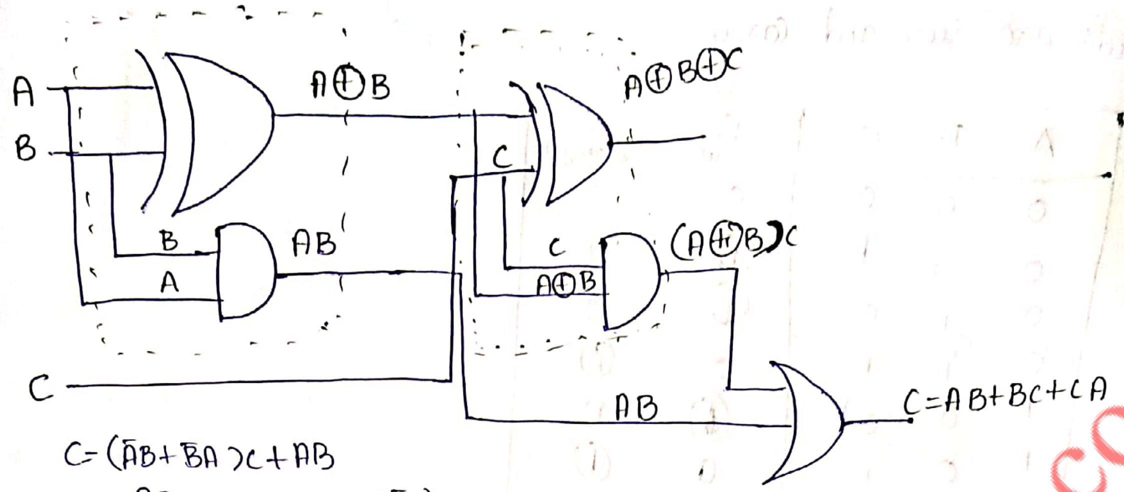
$= \cancel{C(\bar{A}B + A\bar{B}) +}$

Logic Diagram :-



(m)

** To Design full Adder by using two Half adders and OR gate:



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

$= AB(1+C)+C(\bar{A}B+A\bar{B})$

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

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

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

$= AB(1+C)+BC+A\bar{B}C$

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

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

$\boxed{C=AB+BC+AC}$

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

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

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

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

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

$= \cancel{CA}$

Subtraction operations:

A	B		D	B
0	-0	=	0	0
0	-1	=	1	1
1	-0	=	1	0
1	-1	=	0	0

Full-subtraction

→ It consists of three inputs two outputs they are A, B, C is the inputs D, B is the out puts

A	B	C	D	B
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

0 − 1

11 00

0
11

01
01
01

10 01
−0 −0
10

10
00

K-map for D

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0	1 (1)	3	1 (2)
A 1	1 (4)	5	1 (7)	6

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

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

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

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

$$= A \oplus B \oplus C$$

K-map for B

A \ BC	00	01	11	10
0	0	1 (1)	1 (3)	1 (2)
1	4	5	1 (7)	6

G_1, G_2, G_3

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

G_1:

A	B	C
0	0	1
0	1	1

$\bar{A}C$

G_2:

0	1	1
0	1	0

$\bar{A}B$

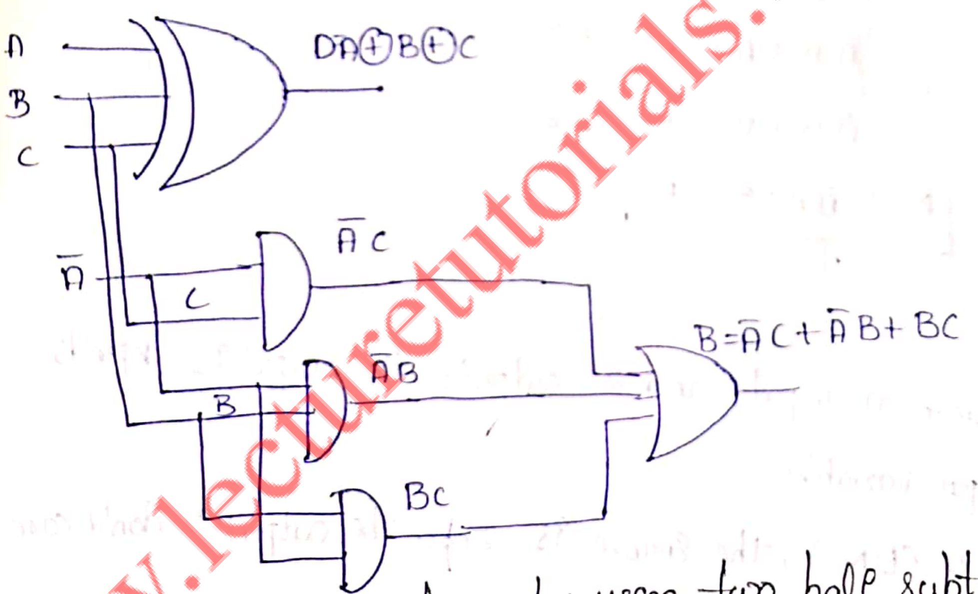
G_3:

0	1	1
1	1	1

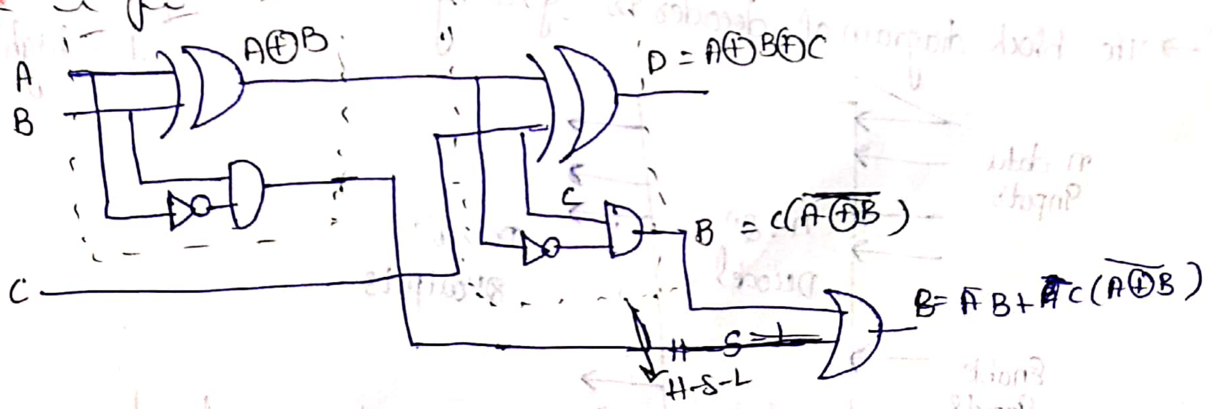
BC

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

logic diagram



To Design full subtraction by using two half subtraction and one OR gate



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

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

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

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

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

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

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

$= \bar{A}B + C\bar{A}\bar{B} + ABC$ $\qquad [\because 1+C=1]$

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

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

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

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

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

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

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

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

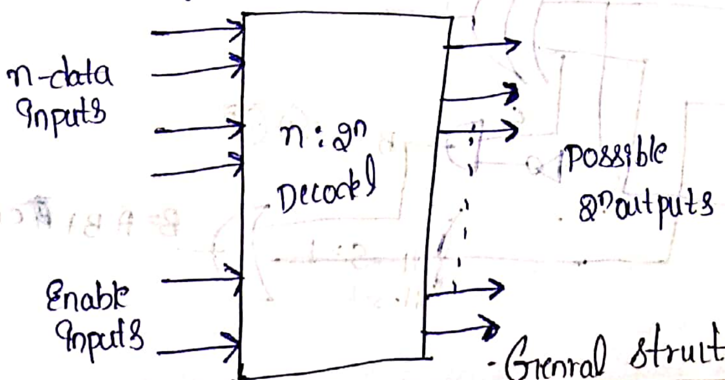
Decoder

→ The Decoder n inputs and 2^n outputs the output is depends upon Enable input variable.

→ The Enable is (EN), the Enable is zero the output is don't care condition

→ The Enable is '1' the output is on state

→ The block diagram of decoder is given by $\qquad$ [0 – Low, 1 – high]



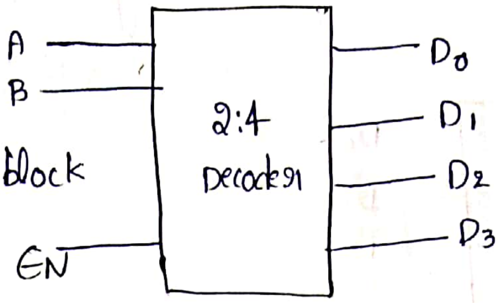
General structure of decoder

Binary Decoder:-

2:4 Decoder:-

Truth table:-

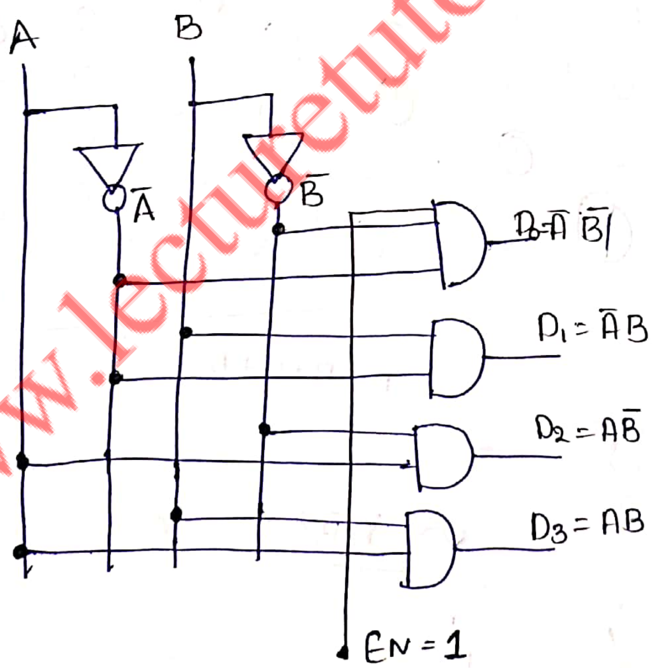
The 2:4 Decoder block diagram is given by



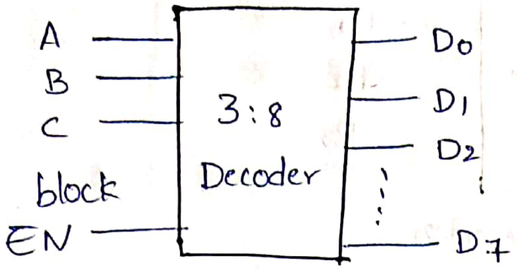
2:4 Decoder

Truth table:-

Inputs			Outputs			
EN	A	B	D3	D2	D1	D0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



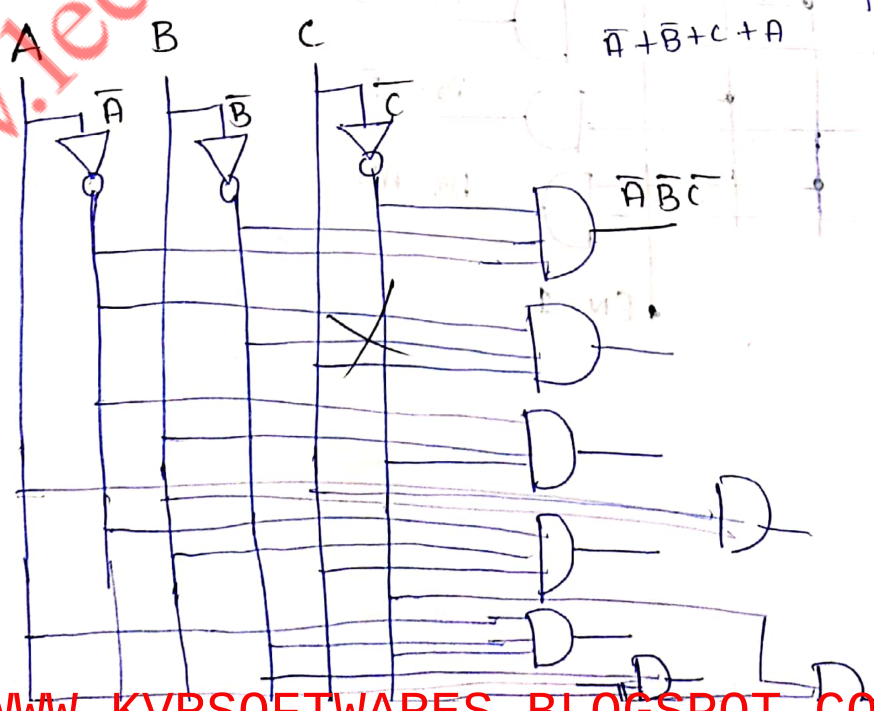
3:8 Decoder :- The 3:8 Decoder is given by



Truth table

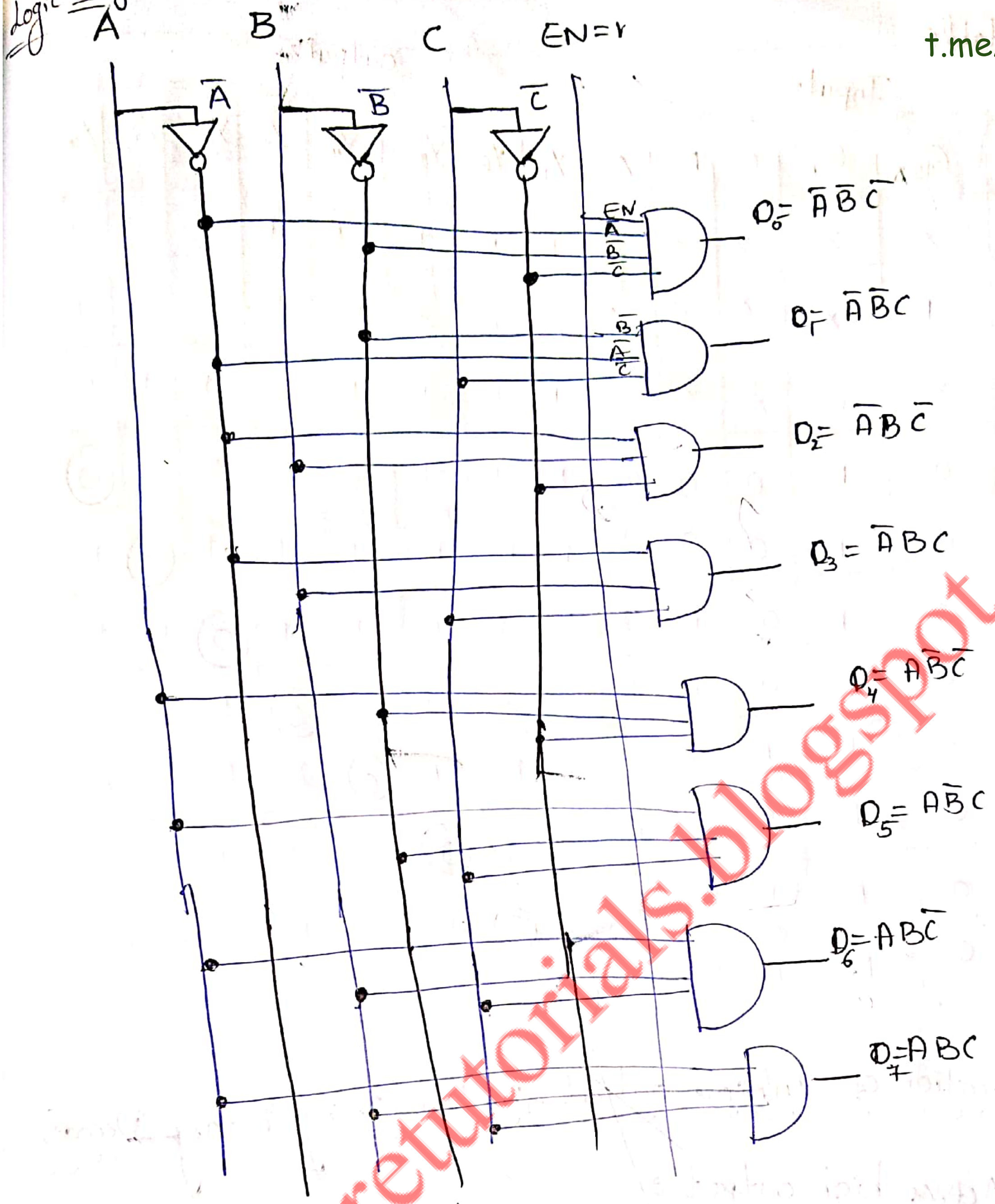
Input				output							
EN	A	B	C	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

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

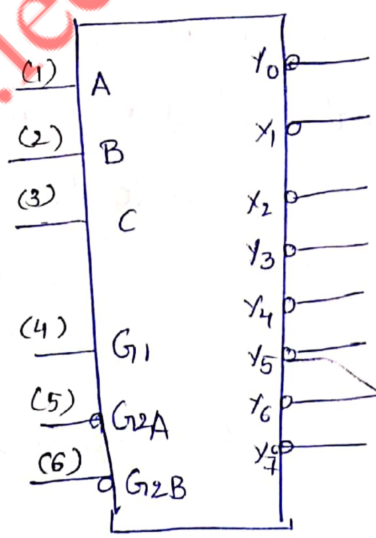


t.me/jntukonlinebits

Logic diagram:



The 74 x 138 3 to 8 Decoder



logic symbol

Truth table

Inputs						Outputs							
G_{2B}	G_{2A}	G_1	C	B	A	$\overline{Y_7}$	$\overline{Y_6}$	$\overline{Y_5}$	$\overline{Y_4}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	0
0	0	1	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	0	1	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	0	1	1	1	0	1	1	1	1
0	0	1	1	0	1	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	0	1	1	1	1	1	1
0	0	1	1	1	1	0	1	1	1	1	1	1	1

Realization of Multiple output function using Binary Decoder

For Active high output:

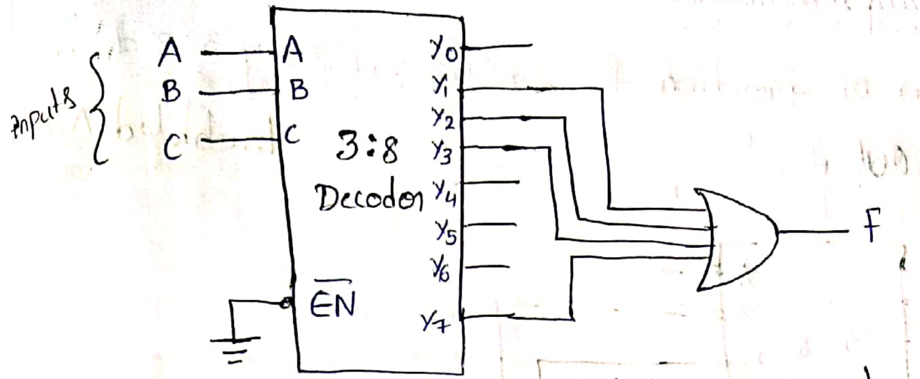
Sop function implementation

When decoder output is active high, it generates minterms (product terms) for input variables i.e it makes selected output logic 1. In such case to implement Sop function we have to take sum of selected product terms genrated by decoder. This can be implemented by ORing the selected decoder outputs, as shown in the fig 5.62. The fig 5.62 shows the implementation of function $f = \Sigma M(1,2,3,7)$ using 3:8 decoder with active high outputs

0 – low – off state

1 – high – on

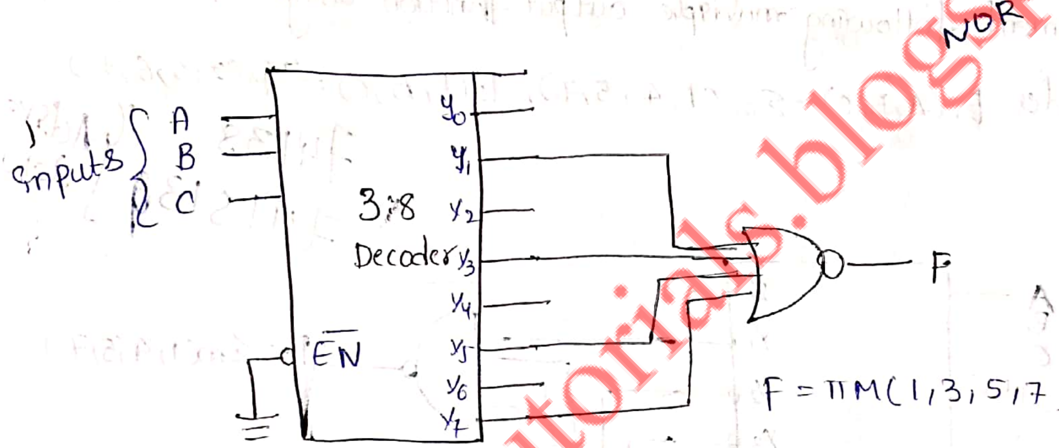
OR t.me/jntukonlinebits



Single output function implementation using decoder and gate

POS Function Implementation

The implementation of a function $f = \Pi M(1,3,5,7)$ using 3:8 decoder with active high outputs.



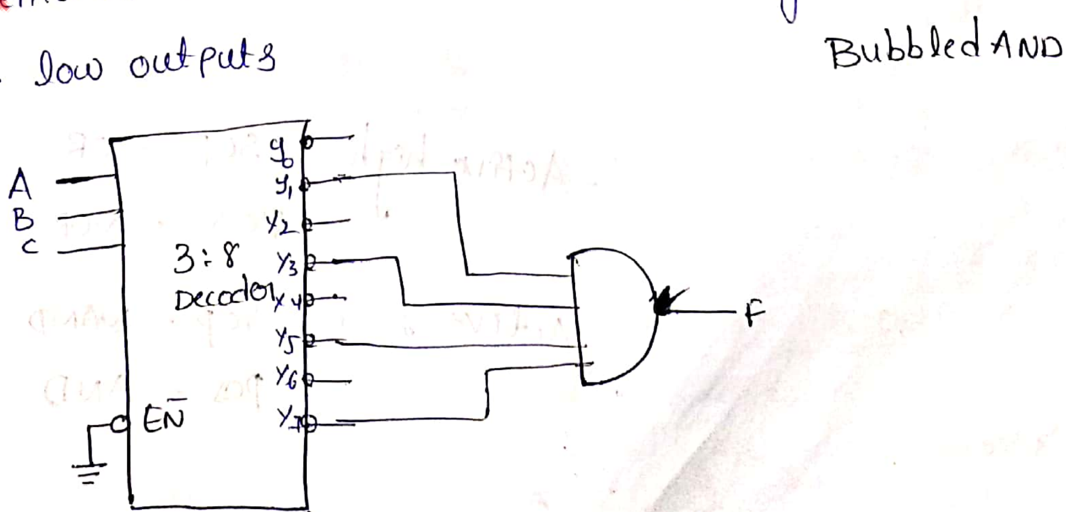
$F = \Pi M(1,3,5,7)$

Implementation of POS function using decoder

For Active low output

POS: function implementation

The implementation of function $f = \Pi M(1,3,5,7)$ using 3:8 decoder with active low outputs

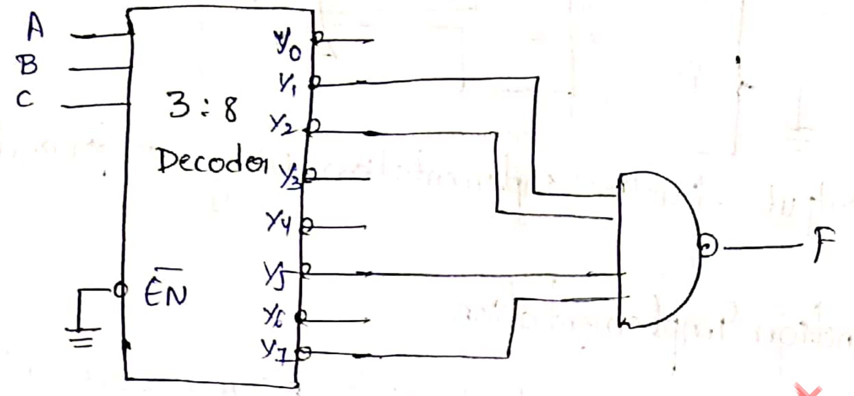


Implementation of POS function using decoder

Sop function Implementation

The implementation of function $f = \Sigma m(1,2,5,7)$ using 3:8 decoder with active low outputs

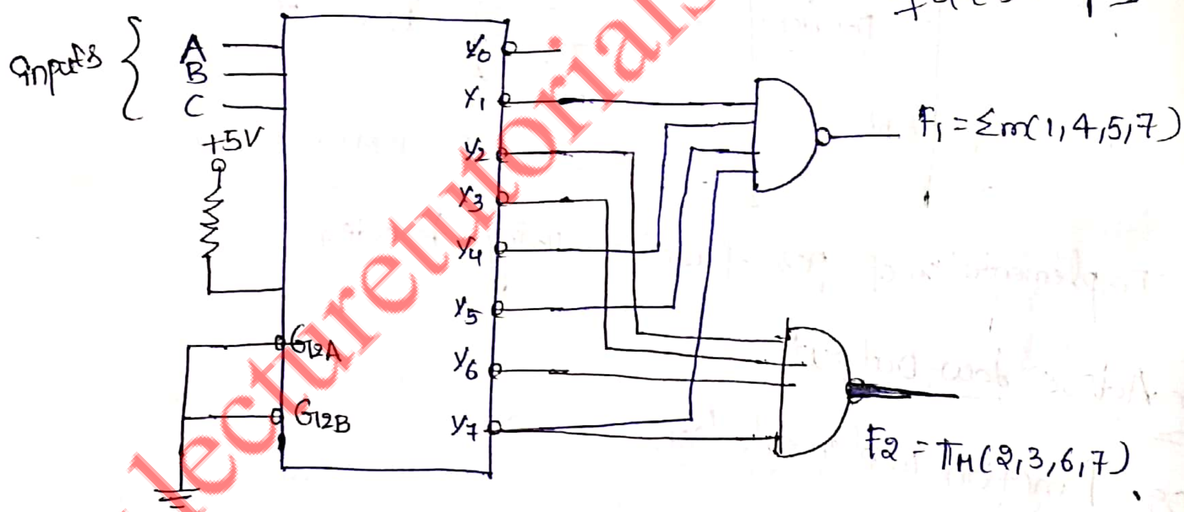
Bubbled NAND Gate



Implementation of sop function using decoder

Ex:- Implement following multiple output function using 74LS138 and External gates. $F_1(A,B,C) = \Sigma m(1,4,5,7)$, $F_2(A,B,C) = \Pi M(2,3,6,7)$

74138
74LS138 } Active low



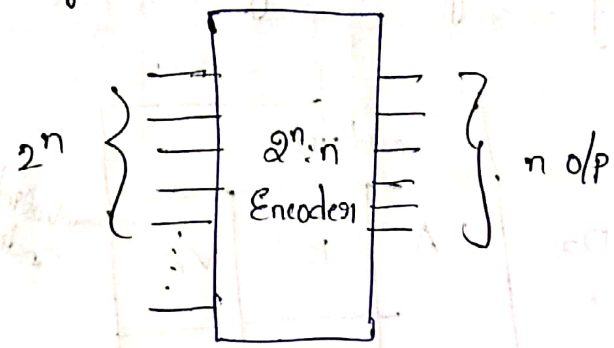
Active high — SOP — OR
POS — NOR

Active Low — SOP — NAND
POS — AND

Encoders: (2^n i/p – n o/p)

→ The encoder it consists of 2^n inputs and n outputs

Block diagram:



Ex: 4

Octal - Binary Encoder

truth-table.

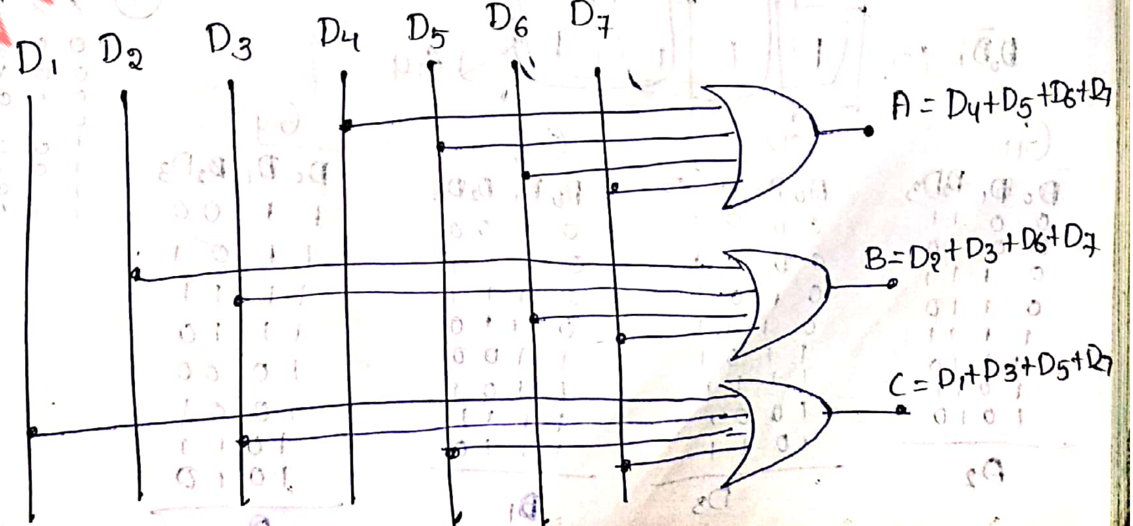
→ The truth table of octal to binary convertor

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$A = D_4 + D_5 + D_6 + D_7$

$B = D_2 + D_3 + D_6 + D_7$

$C = D_1 + D_3 + D_5 + D_7$



Priority Encoder:

A priority encoder is an encoder circuit that includes the priority function. In priority encoder, if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

truth table:

Inputs				outputs		
D_0	D_1	D_2	D_3	A	B	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

V → valid indicator

four variable k-map - "V"

D_0D_1 \ D_2D_3	00 ($\bar{D}_2\bar{D}_3$)	01 ($\bar{D}_2D_3$)	11 (D_2D_3)	10 ($D_2\bar{D}_3$)
$\bar{D}_0\bar{D}_1$ 00		1	1	1
$\bar{D}_0D_1$ 01	1	1	1	1
D_0D_1 11	1	1	1	1
$D_0\bar{D}_1$ 10	1	1	1	1

G_1, G_2, G_3, G_4

(i) 1 0 0 0

(ii) x 1 0 0
0 1 0 0
1 1 0 0

(iii) x x 1 0
0 0 1 0
0 1 1 0
1 0 1 0
1 1 1 0

(iv) x x x 1
0 0 0 1
0 0 1 1
0 1 0 1
0 1 1 1
1 0 0 1
1 0 1 1
1 1 0 1
1 1 1 1

G_1

D_0	D_1	D_2	D_3
0	0	1	1
0	0	1	0
0	1	1	1
0	1	1	0
1	1	1	1
1	1	1	0
1	0	1	1
1	0	1	0

D_2

G_2

D_0	D_1	D_2	D_3
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	1	0	1
1	1	1	1
1	0	0	1
1	0	1	1

D_3

G_3

D_0	D_1	D_2	D_3
0	1	0	0
0	1	0	1
0	1	1	1
0	1	1	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0

D_1

G_4

D_0	D_1	D_2	D_3
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	0	0
1	0	0	1
1	0	1	1
1	0	1	0

D_0

$V = D_0 + D_1 + D_2 + D_3$

k-map for "A"

$D_0D_1 \backslash D_2D_3$	00	01	11	10
00		1	1	1
01		1	1	1
11		1	1	1
10		1	1	1

G_1: $D_0 D_1 D_2 D_3$	G_2: $D_0 D_1 D_2 D_3$
0011	0001
0010	0011
0111	0101
0110	0111
1111	1101
1110	1111
1011	1001
1010	1011
D_2	D_3

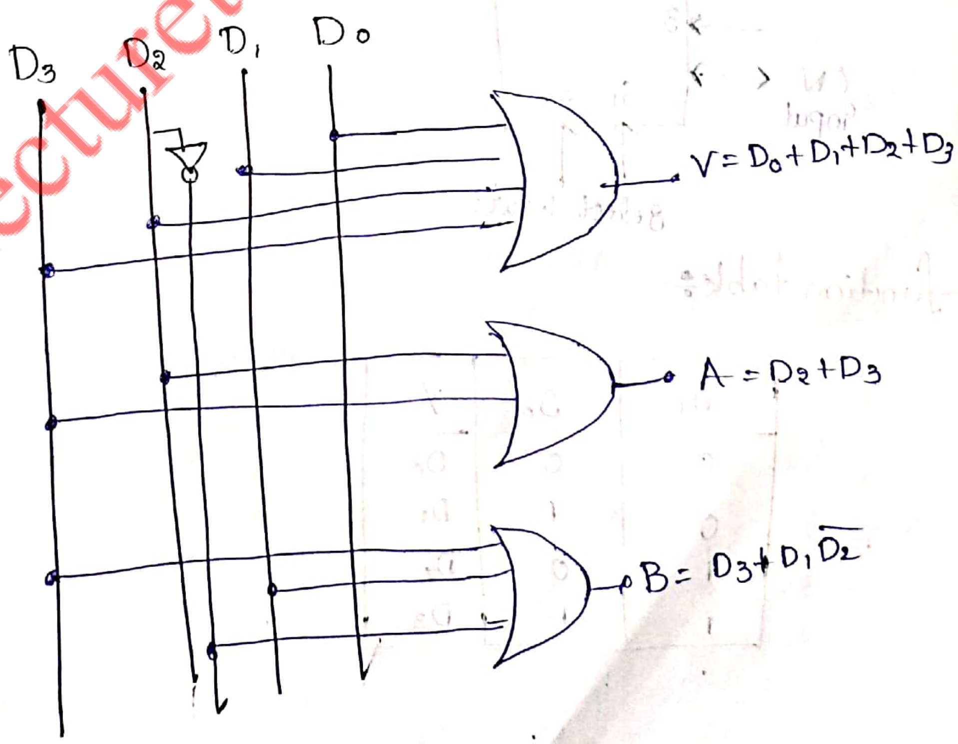
$A = D_2 + D_3$

k-map for "B"

$D_0D_1 \backslash D_2D_3$	00	01	11	10
00		1	1	
01	1	1	1	
11	1	1	1	
10		1	1	

G_1: $D_0 D_1 D_2 D_3$	G_2: $D_0 D_1 D_2 D_3$
0001	0100
0011	0101
0101	1100
0111	1101
1101	
1111	
1001	
1011	
D_3	$D_1\overline{D_2}$

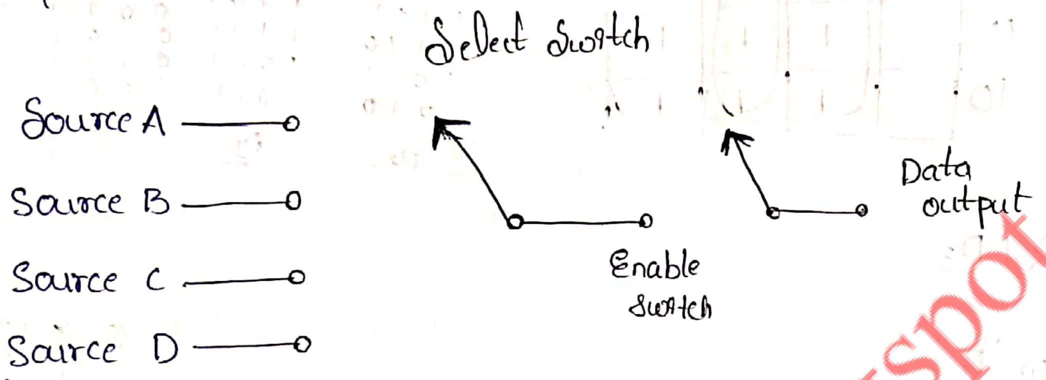
$B = D_3 + D_1\overline{D_2}$



Multiplexers (2^n inputs – n selection lines – 1 output)

Multiplexer is a digital switch. It allows digital information from several sources to be routed onto a single output line.

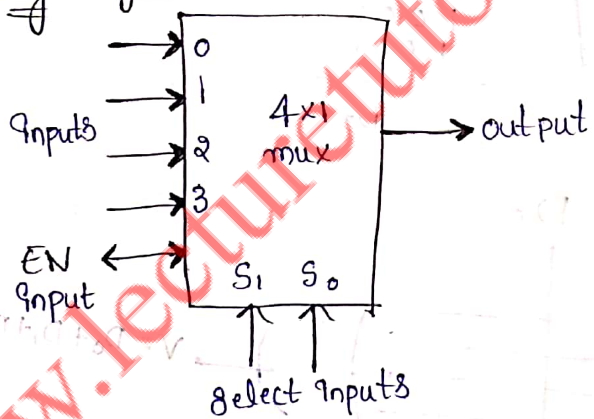
It consists of 2^n input lines and n selection lines and only one output.



Analog selector switch

4 to 1 line multiplexer:

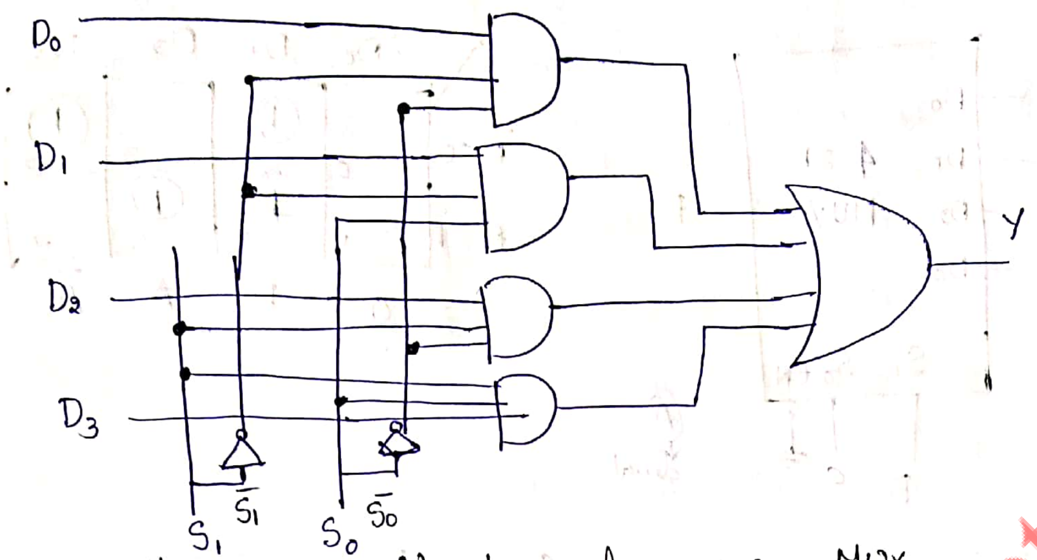
Logic symbol:



Function table:

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

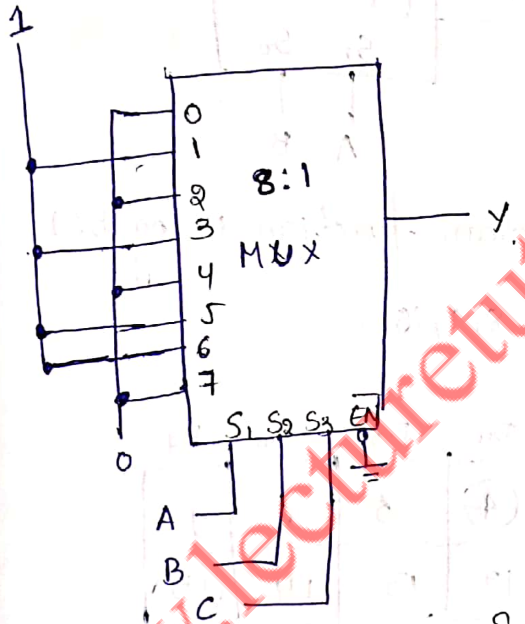
Logic diagram:



Implementation of combinational logic using MUX

Ex: Implement the following Boolean function using 8:1 multiplexer

$F(A,B,C) = \Sigma m(1,3,5,6)$



Boolean function ~~usin~~ implemention using MUX

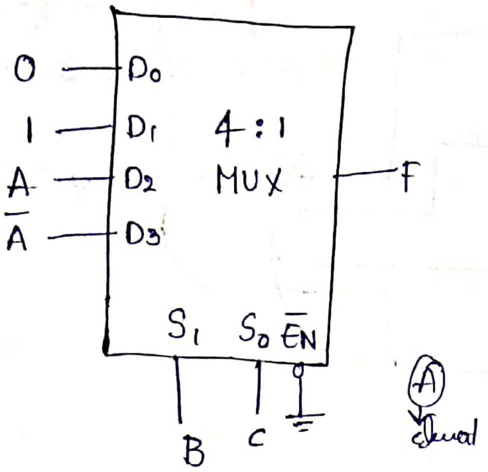
Ex: ② Implemention the following Boolean function 4:1 multiplexer

$F(A,B,C) = \Sigma m(1,3,5,6)$

truth table

minterm	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

multiplexer Implementation



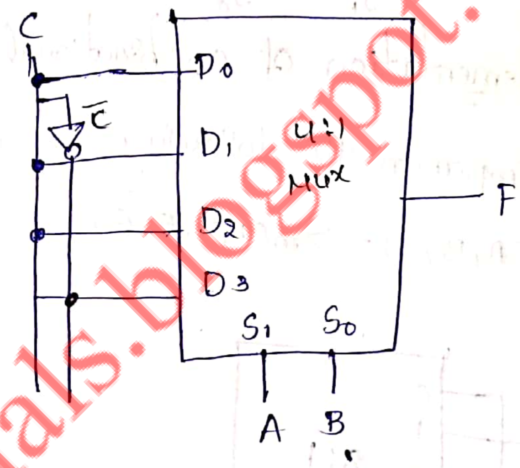
Implementation table

$F = \Sigma m(1,3,5,6)$

	D_0	D_1	D_2	D_3
$\bar{A}=0$	0	1	2	3
$A=1$	4	5	6	7
	0	1	A	$\bar{A}$

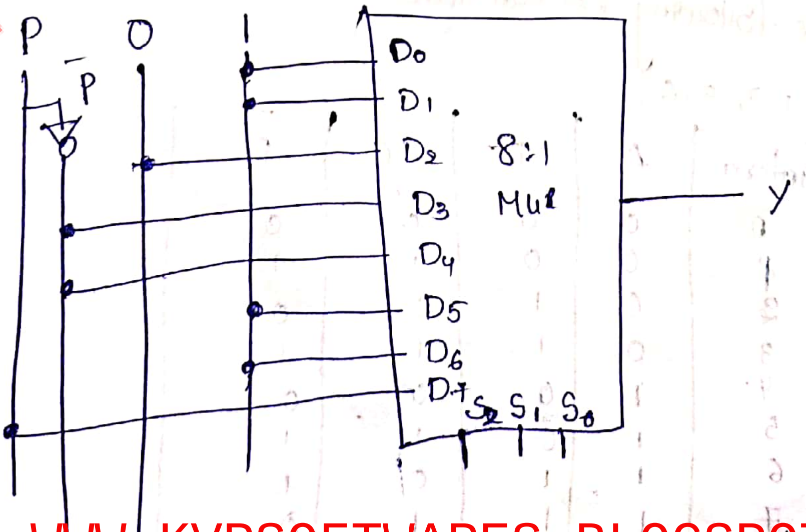
Case :- 2

	$\bar{C}$	C
C	0	1
C	2	3
C	4	5
$\bar{C}$	6	7



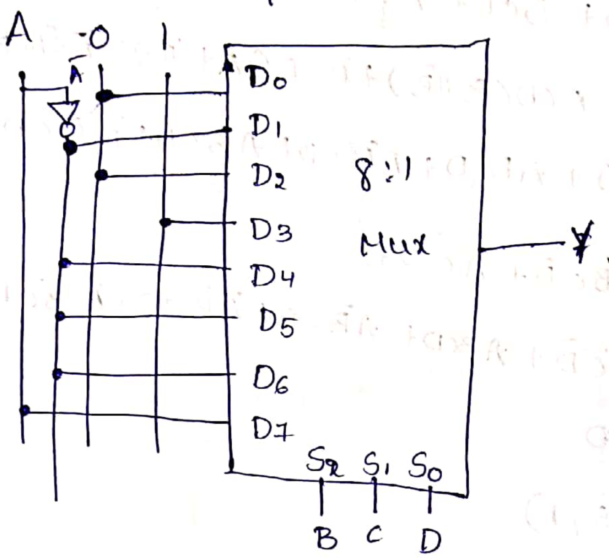
Ex :- Implement the following Boolean function using 8:1 Mux. $F(P,Q,R,S) = \Sigma m(0,1,3,4,8,9,15)$

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{P}$	0	1	2	3	4	5	6	7
P	8	9	10	11	12	13	14	15
	1	1	0	$\bar{P}$	$\bar{P}$	0	0	P



	D0	D1	D2	D3	D4	D5	D6	D7
	0	1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15
	0	$\bar{A}$	0	1	$\bar{A}$	$\bar{A}$	$\bar{A}$	A

Implementation table



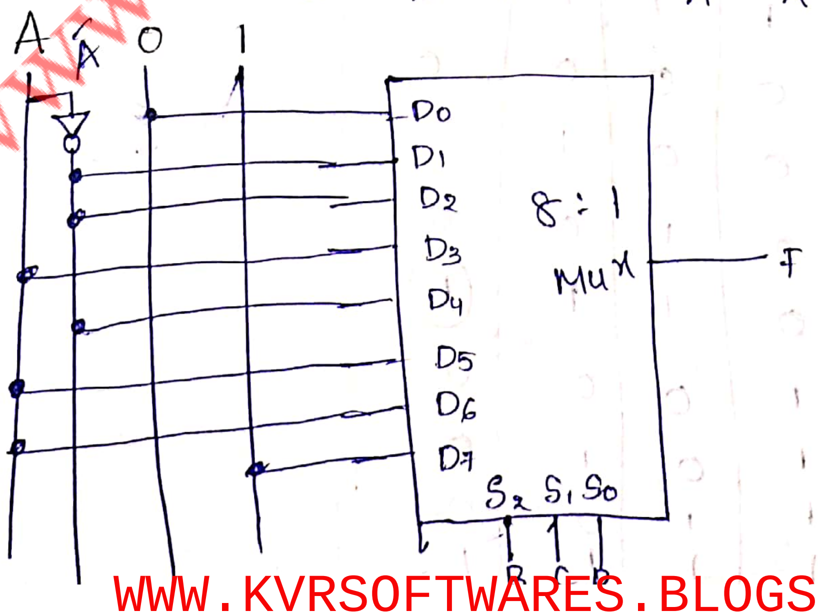
Multiplexer Implementation

* Implement the following Boolean function using 8:1 multiplexer

$F(A,B,C,D) = \prod M(0,3,8,5,9,10,12,14)$

Here, instead of minterms, maxterms are specified. Thus, we have to circle maxterms which are not included in the Boolean function shown below

	D0	D1	D2	D3	D4	D5	D6	D7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	0	$\bar{A}$	$\bar{A}$	A	$\bar{A}$	A	$\bar{A}$	1



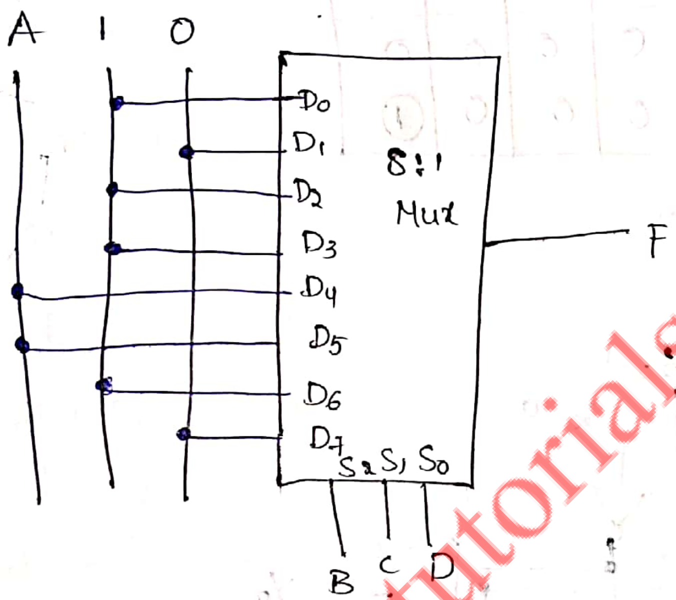
Ex: Implement the following Boolean function with 8:1 Multiplexer

$F(A,B,C,D) = \Sigma m(0,2,6,10,11,12,13) + d(3,8,14)$

Sol:

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	0	1	1	A	A	1	0

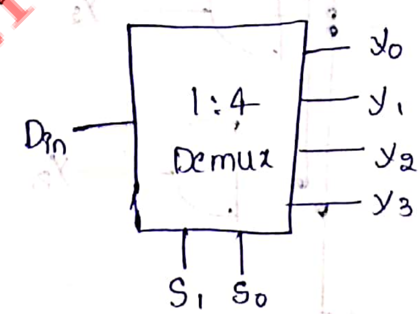
Here don't cares are treated 1s



Demultiplexer (1 Input - n selection lines - 2^n outputs)

1:4 Demux:

Logic symbol:



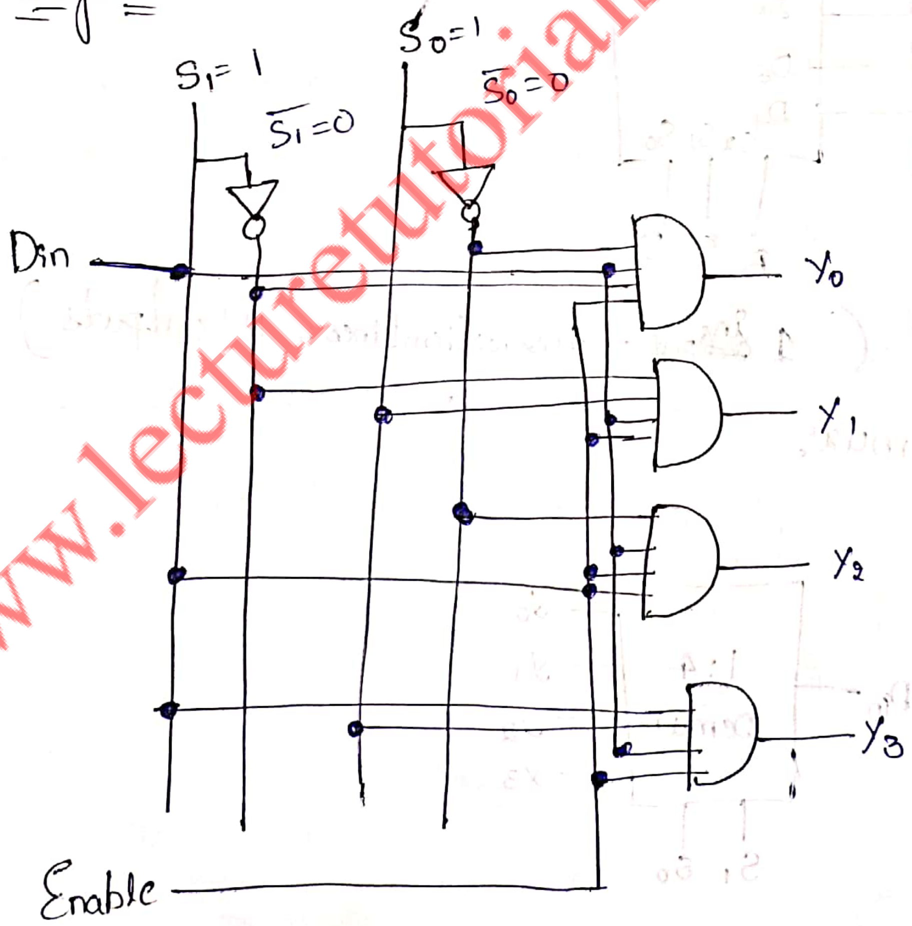
Enable (0) - Inputs are Don't care
outputs are zero

truth table:

Enable	S_1	S_0	Din	y_0	y_1	y_2	y_3
0	x	x	x	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1

1:4 demultiplexer

Logic diagram



Ex: Implement the full subtractor by using demultiplexer

A	B	D	B
0 − 0		0	0
0 − 1		1	1
1 − 0		1	0
1 − 1		0	0

Sol:

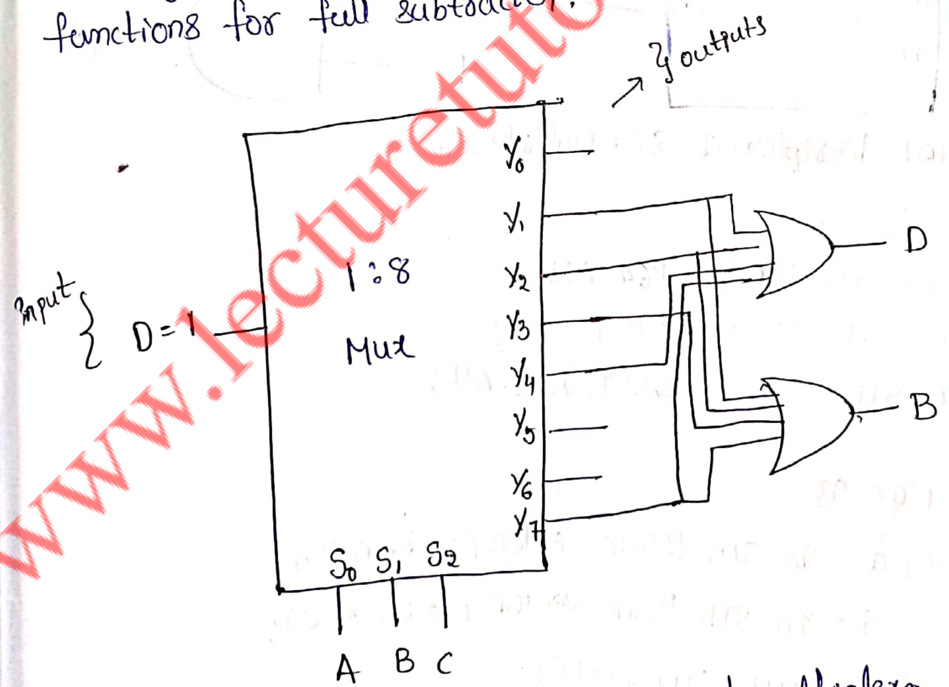
A	B	C	D Difference	B Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

truth table of full subtractor

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

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

With Din input 1, demultiplexer gives minterms at the output So by logically ORing required minterms we can implement Boolean functions for full subtractor.



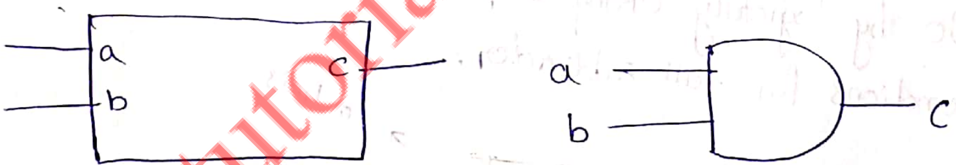
Full subtractor using 1:8 demultiplexer

HDL programming :- (Hardware discription language)

Ex:- write the HDL discription of AND gate

```
Library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.STD_LOGIC_ARITH.ALL;
Use IEEE.STD_LOGIC_UNSIGNED.ALL;

Entity andgate is
    port (a,b :in STD_LOGIC;
          c :out STD_LOGIC);
end andgate;
architecture Behavioral of andgate is
begin
c<=a and b;
End Behavioral;
```

Output:-



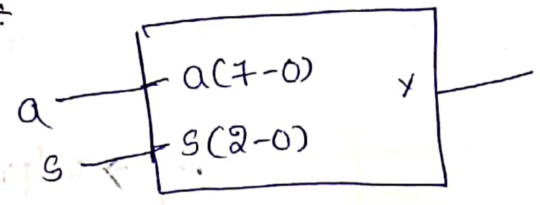
Ex:- write the HDL discription of 8:1 Multiplexer

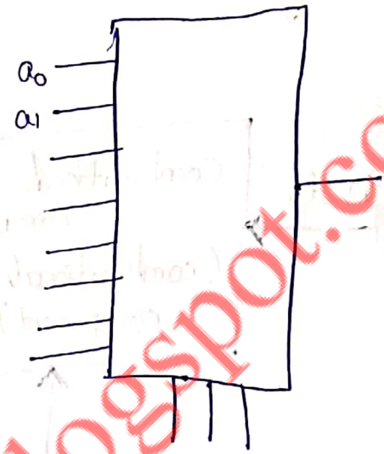
```
Library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.STD_LOGIC_ARITH.ALL;
Use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Mux is
    port (a : in STD LOGIC VECTOR (7 down 0);
          S : in STD LOGIC VECTOR (2 downto 0);
          Y : out STD LOGIC);

End MUX;
architecture Behavioral of Mux is
begin
Y <= a(0) when S = "000" else
     a(1) when S = "001" else
```

a(2) when s = "010" else
a(3) when s = "011" else
a(4) when s = "100" else
a(5) when s = "101" else
a(6) when s = "110" else
a(7) when s = "111" ;

End Behavioral;

Output:





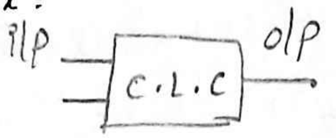
Date
5/8/19

Unit-V

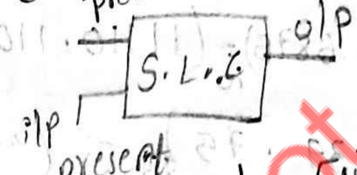
Synchronous Sequential Logic:

Logic Circuit (ckt)
- Combinational logical ckt
- Sequential logic ckt

Ex:

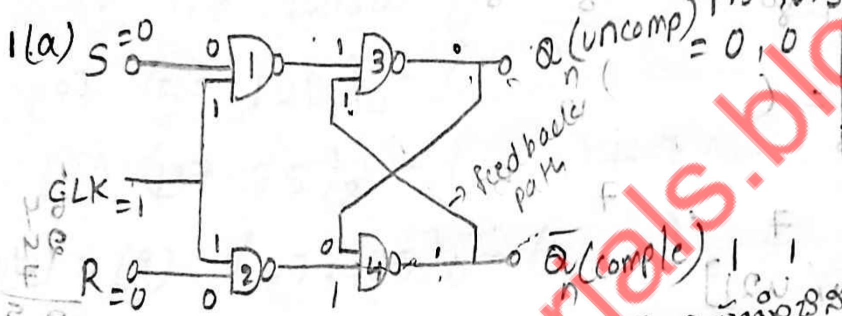


Ex:



Basic Construction of S-R Flip Flop (NAND)

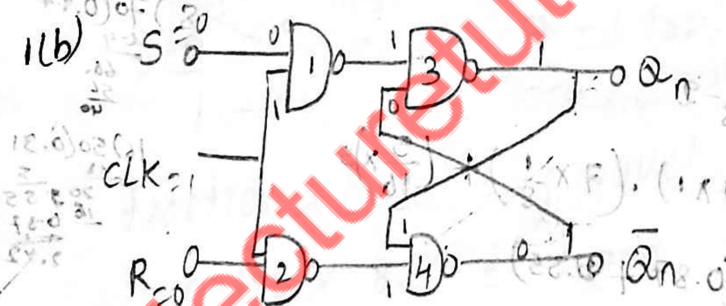
1(a)



	Q_n P.S	Q_{n+1} N.S
Q	0	0
$\bar{Q}$	1	1

$\bar{Q}_n=1$ వస్తుంది
Q=0 ఉండటం వల్ల $\bar{Q}=1$ వస్తుంది
$\bar{Q}=1$ ఉండటం వల్ల Q=0 వస్తుంది

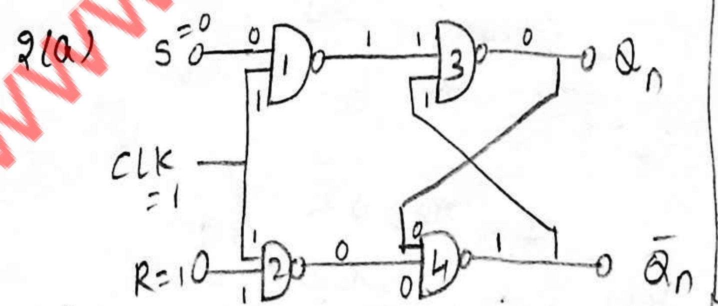
1(b)



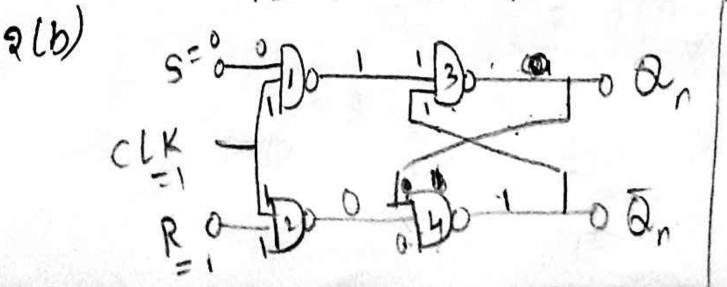
Q_n	1	1
$\bar{Q}_n$	0	1

1(a) & 1(b) are no change condition $P.S = N.S \Rightarrow Q_n = Q_{n+1}$

2(a)



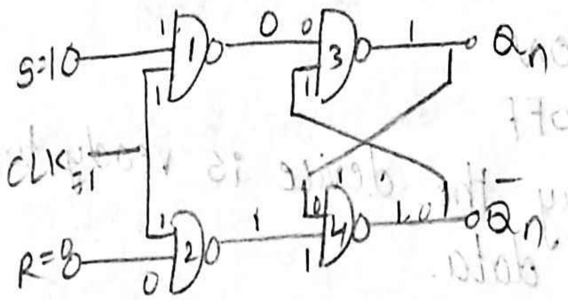
2(b)



	P.S	N.S
2(a) Q_n	0	0
2(a) $\bar{Q}_n$	1	1
2(b) Q_n	1	0
2(b) $\bar{Q}_n$	1	1

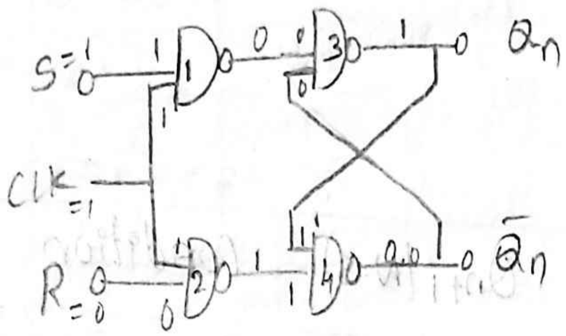
Reset condition (output=0)

3)



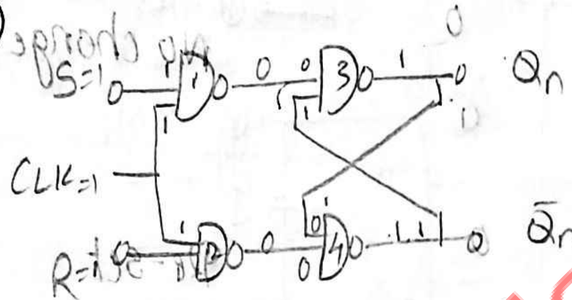
P.S	N.S
0	1
1	0

Set state (output=1)



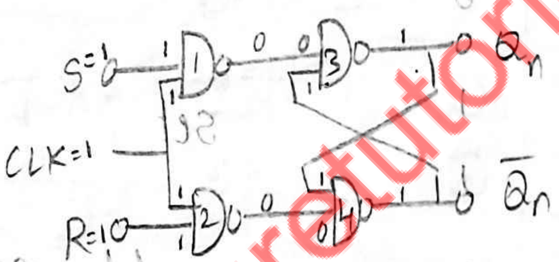
1	1
0	0

4)



P.S	N.S
0	1
1	1

Indeterminate condition



1	1
0	0
1	1

Flip flop:

→ Flip flop is a single bit storage devices

→ Types of flip flops are four

* S-R flip flop [Set-Reset]
* J-K flip flop
* T flip flop
* D flip flop

NAND

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

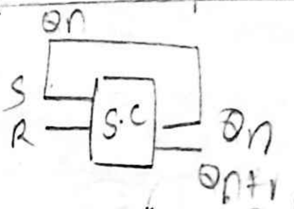
Note:

1. when CLK = 1 it is ON
2. when CLK = 0 it is OFF
3. CLK is used to say the device is ready to accept the input or data.
4. PS → present state
5. N.S → Next state

Date 8/8/19

Truth Table

	SNO	S	R	Q_n (P.S)	Q_{n+1} (N.S)	Condition
0	1	0	0	0	0	No change
1		0	0	1	1	
2	2	0	1	0	0	Re-set
3		0	1	1	0	
4	3.	1	0	0	1	set
5		1	0	1	1	
6	4.	1	1	0	X	Indeterminent
7		1	1	1	X	



Characteristic Eq^n

SR Q_n ides → Q_{n+1}

Row . col

Quard (6, 4, 7, 5)

$$(SR + S\bar{R})\cdot(\bar{Q}_n + Q_n)$$

$$S(R+\bar{R})$$

$$S$$

pair (1, 5)

$$(\bar{S}\bar{R} + S\bar{R})Q_n$$

$$\bar{R}(\bar{S}+S)Q_n$$

$$\bar{R}Q_n$$

SR \\ Q_n	0	1
$\bar{S}\bar{R}$ 00	0	1 (1)
$\bar{S}R$ 01	2	3
SR 11	X 6	X 7
$S\bar{R}$ 10	1 4	1 5

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

Minimized Truth Table

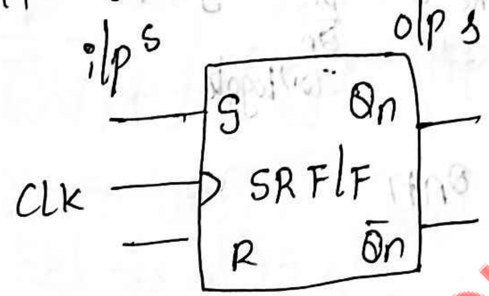
i/ps S	R	Q_{n+1} =
0	0	Q_n
0	1	0
1	0	1
1	1	X

Excitation Table

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

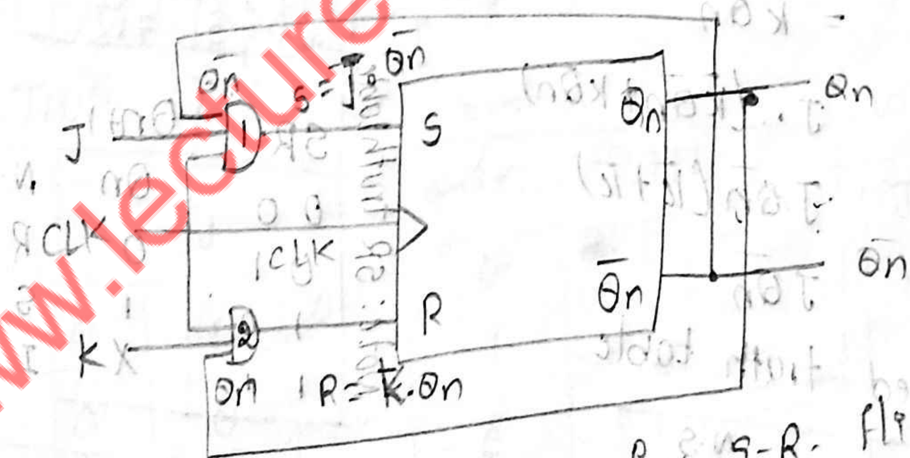
Date 9/8/19

Symbol



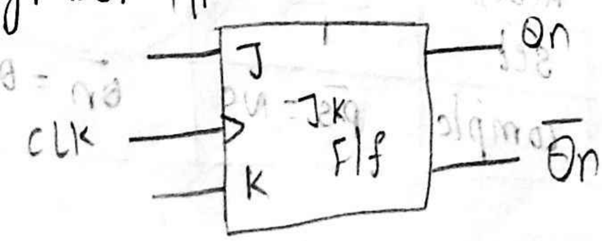
J-K- flip flop

AND Gate



Circuit Diagram for S-R flip flop

Symbol



Truth Table

excitation table

	J	K	S	R	P.S $Q_n = \bar{Q}_n$		N.S Q_{n+1}	C	$S = J\bar{Q}_n$	$R = KQ_n$
0	0	0	0	0	0	1	0	① N.C	$S = 0\cdot1 = 0$	$R = 0\cdot0 = 0$
1	0	0	0	0	1	0	1	④	$S = 0\cdot0 = 0$	$R = 0\cdot1 = 0$
2	0	1	0	0	0	1 N.C	0	① R.S	$S = 0\cdot1 = 0$	$R = 1\cdot0 = 0$
3	0	1	0	1	1	0 R.S	0	③	$S = 0\cdot0 = 0$	$R = 1\cdot1 = 1$
4	1	0	1	0	0	1 Set	1	② S	$S = 1\cdot1 = 1$	$R = 0\cdot0 = 0$
5	1	0	0	0	1	0 NC	1	④	$S = 1\cdot0 = 0$	$R = 0\cdot1 = 0$
6	1	1	1	0	0	1 set	1	② comple	$S = 1\cdot1 = 1$	$R = 1\cdot0 = 0$
7	1	1	0	1	1	0 Re	0	③ Q_n (or) toggle	$S = 1\cdot0 = 0$	$R = 1\cdot1 = 1$

old SR flip table, border conditions present

characteristic Equation

JKQ_n i/ps $\rightarrow Q_{n+1}$

J \ KQ_n	$\bar{K}\bar{Q}_n$ 00	$\bar{K}Q_n$ 01	KQ_n 11	$K\bar{Q}_n$ 10
$\bar{J}$ 0	0	1 (1)	3	2
J 1	1 (4)	1 (5)	7	1 (6)

pair(1, 5) rows. cols

$(\bar{J}+J)\cdot\bar{K}Q_n$

$= \bar{K}Q_n$

pair(4, 6) $J\cdot(\bar{K}\bar{Q}_n + K\bar{Q}_n)$

$J\bar{Q}_n(\bar{K}+K)$

$J\bar{Q}_n$

Note: SR truthtable

SR	Q_{n+1}	
0 0	Q_n	N.C
0 1	0	R
1 0	1	S
1 1	X	I

Minimized truth table

J	K	N.S Q_{n+1} =			
0	0	Q_n	N.C	P.S = N.S	$Q_n = Q_{n+1}$
0	1	0	Reset	0	
1	0	1	set	1	
1	1	$\bar{Q}_n$	comple	P.S = NS	$\bar{Q}_n = Q_{n+1}$

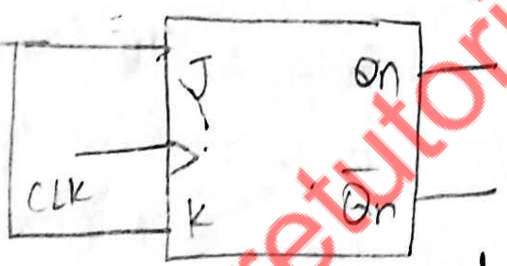
Exitation Table

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

T F/f / Toggle flip flop

T = 0

Date 10/8/19



T flip flop

i/p	o/p
1	2
T	$Q_n, \bar{Q}_n$

It has single input and we get two outputs

Truth table

Assumption

T	J	K	Q_n	Q_{n+1}	
0	0	0	0	0	→ (1) N.C
0	0	0	1	1	
1	1	1	0	1	→ (2) comp
1	1	1	1	0	

using J-K f/f

J	K	Q_{n+1}	
0	0	Q_n	NC
0	1	0	R
1	0	1	S
1	1	$\bar{Q}_n$	comp

Minimized truth table

i/p T	N.S Q_{n+1}
0	0
0	1
1	0
1	T

i/p T	N.S Q_{n+1}	
0	Q_n	N.C
1	$\bar{Q}_n$	comp

Excitation table

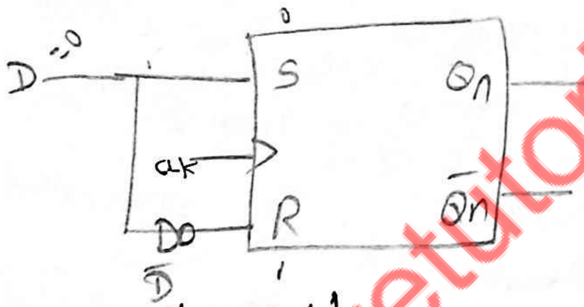
i/p Q_n	Q_{n+1}	o/p T
0	0	0
0	1	1
1	0	1
1	1	0

characteristic equation

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

T \ Q_n	$\bar{Q}_n$ 0	Q_n 1
$\bar{T}$ 0	0	1
T 1	1	3

D flip Flop (or) Delay flip flop



This D flip flop can be constructed by using S-R flip flop

Truth Table

D i/p	S=D	R=$\bar{D}$	(P.S) Q_n	(N.S) Q_{n+1}	
0	0	1	0	0	Reset
0	0	1	1	0	
1	1	0	0	1	set
1	1	0	1	1	

S.R. f/f Truth table

S	R	Qn+1	
0	0	Qn	N.C
0	1	0	R
1	0	1	S
1	1	X	inde

Minimized truth table

i/p D	N.S Qn+1	
0	0	R
1	1	S

Exitation table

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

characteristic equation

$Q_{n+1} = D$

	D	Qn	Qn+1
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	1

D \ Qn	0	1
$\bar{D}$ 0	0	1
D 1	1 (2)	1 (3)

Date 13/8/19

Conversion of flip flops

1. Convert S-R flip flop to D-flip flop (or) construct D flip flop using S-R flip flop.

Solu: 1. which flip flop is going to construct take the truth table of that flip flop

2. which flip flop is going to use for the construction take exitation table of the flip flop

	i/p D	P.S Qn	N.S Qn+1	condition	o/p S R
0	0	0	0	Reset	0 X
1	0	1	0		0 1
2	1	0	1	set	1 0
3	1	1	1		X 0

Truth table of D flip

Qn	Qn+1	S R
0	0	0 X
0	1	1 0
1	0	0 1
1	1	X 0

exitation table of SR flip flop

S-R flip flop → D flip flop

S K-map

D \ Qn	$\bar{Q}_n$ 0	Q_n 1
$\bar{D}$ 0	0	1
D 1	1 2	X 3

R- K-map

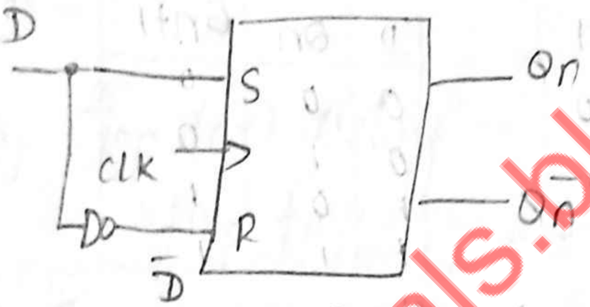
D \ Qn	$\bar{Q}_n$ 0	Q_n 1
$\bar{D}$ 0	X 0	1 1
D 1	2	3

$S = D(\bar{Q}_n + Q_n)$

$S = D$

$R = \bar{D}(\bar{Q}_n + Q_n)$

$= \bar{D}$



D-flip flop

2. Convert S-R flip flop to J-K flip flop

present (S R) flip flop → JK flip flop (construct)

Exitation table — Truth table

Truth table of J-K flip flop — Exitation table of S-R flip flop

	J	K	present Q_n	Next s Q_{n+1}	condition	S R
0	0	0	0	0	N.C	0 X
1	0	0	1	1		X 0
2	0	1	0	0	Reset	0 X
3	0	1	1	0		0 1
4	1	0	0	1	set	1 0
5	1	0	1	1		X 0
6	1	1	0	1	comp	1 0
7	1	1	1	0		0 1

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

$S = \sum_m (4,6) + d(1,5)$ $R = \sum_m (3,7)$ table/37

S K-map

J \ KQ_n	$\bar{K}\bar{Q}_n$ 00	$\bar{K}Q_n$ 01	KQ_n 11	$K\bar{Q}_n$ 10
$\bar{J}$ 0	0	X 1	3	2
J 1	1 4	X 5	7	1 6

R K-map

J \ KQ_n	$\bar{K}\bar{Q}_n$ 00	$\bar{K}Q_n$ 01	KQ_n 11	$K\bar{Q}_n$ 10
$\bar{J}$ 0	X 0	1	1 3	X 2
J 1	4	5	1 7	6

pair (4, 6)

$= J(\bar{K}\bar{Q}_n + K\bar{Q}_n)$

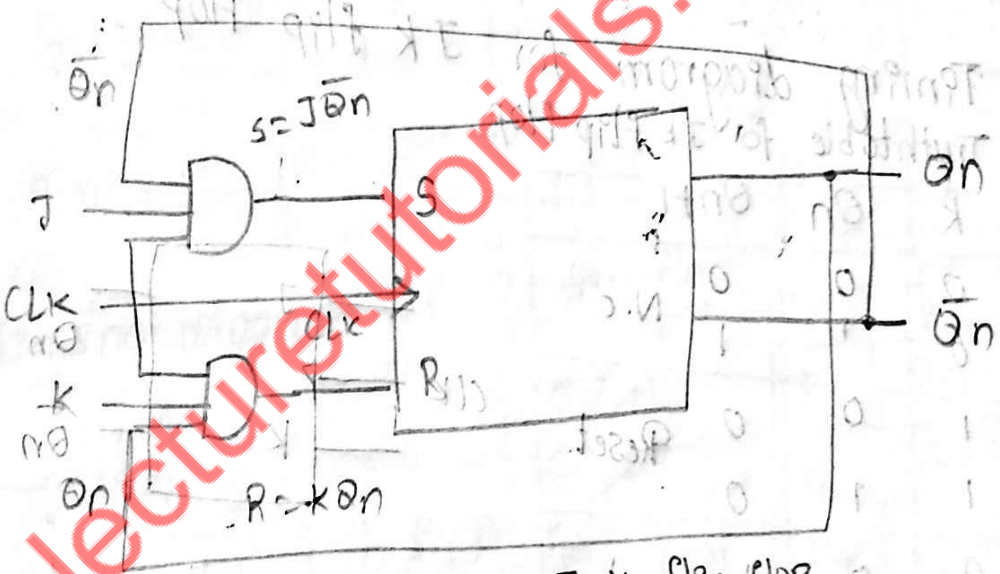
$= J\bar{Q}_n(\bar{K}+K)$

$S = J\bar{Q}_n$

pair (3, 7)

$= (\bar{J}+J) Q_n K$

$R = KQ_n$

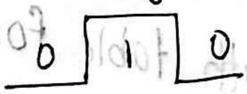


J-K flip flop

Timing Diagrams

⟹ S-R flip flop (or) Edge trigger flip flop

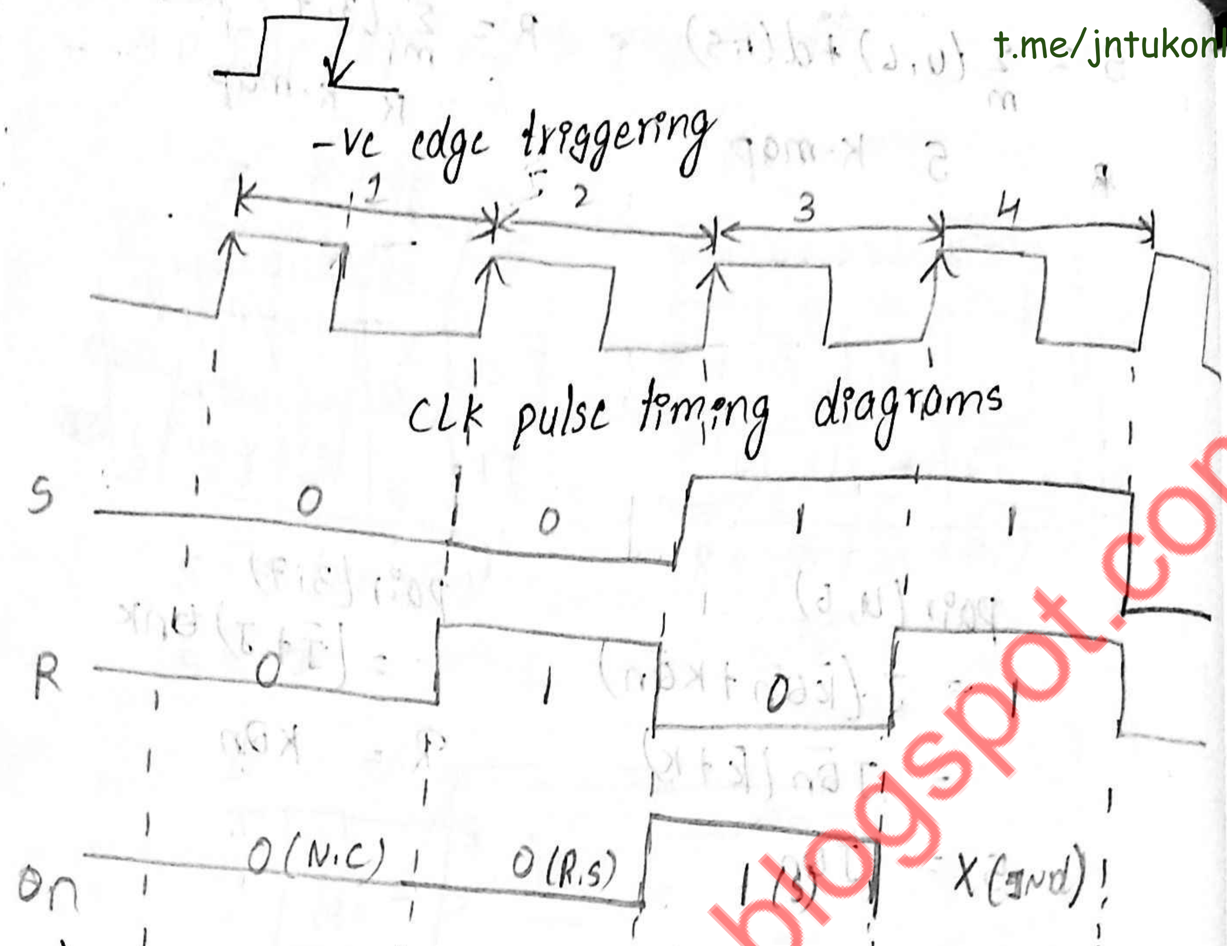
start CLK



Minimized truth table S	R	Q_{n+1}	
0	0	Q_n	N.C
0	1	0	R.S
1	0	1	S
1	1	X	iD

OFF ⟶ ON (triggering)

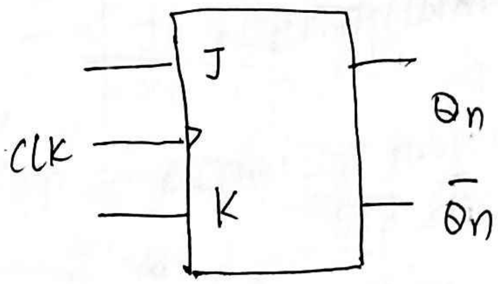
↑ +ve edge triggering



Date 14/8/19 Timing diagram for JK flip flop

Truth table for JK flip flop

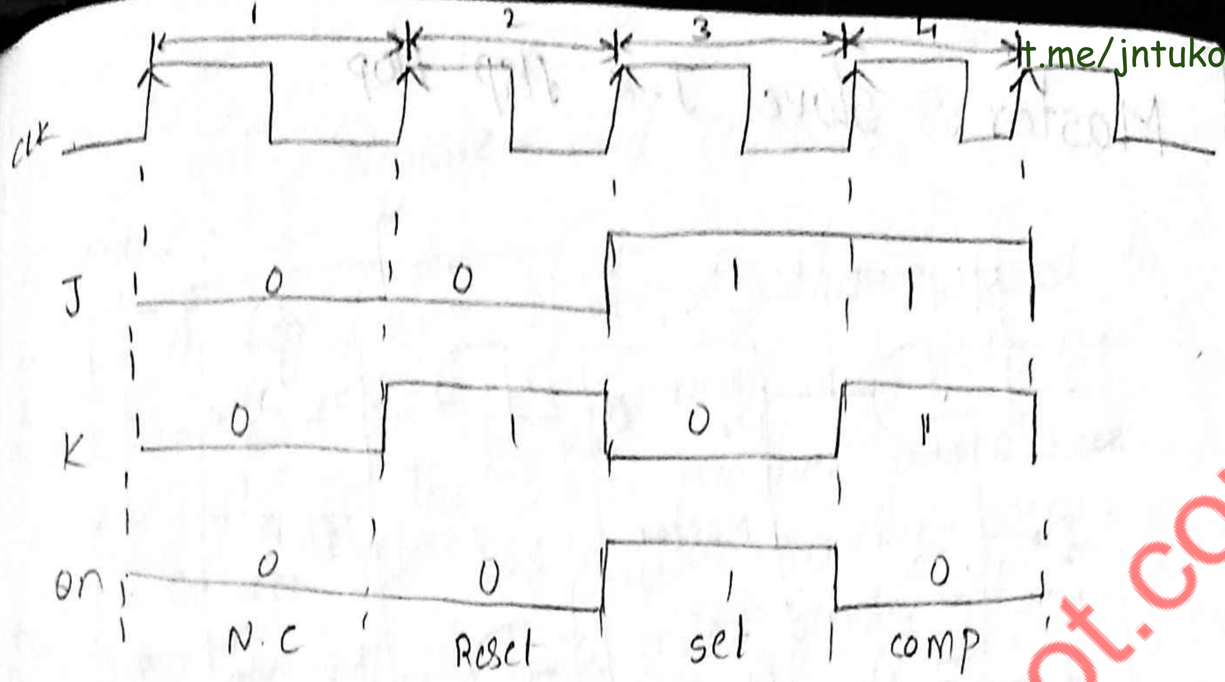
J	K	Qn	Qn+1	
0	0	0	0	N.C
0	0	1	1	
0	1	0	0	Reset.
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	1	comple
1	1	1	0	



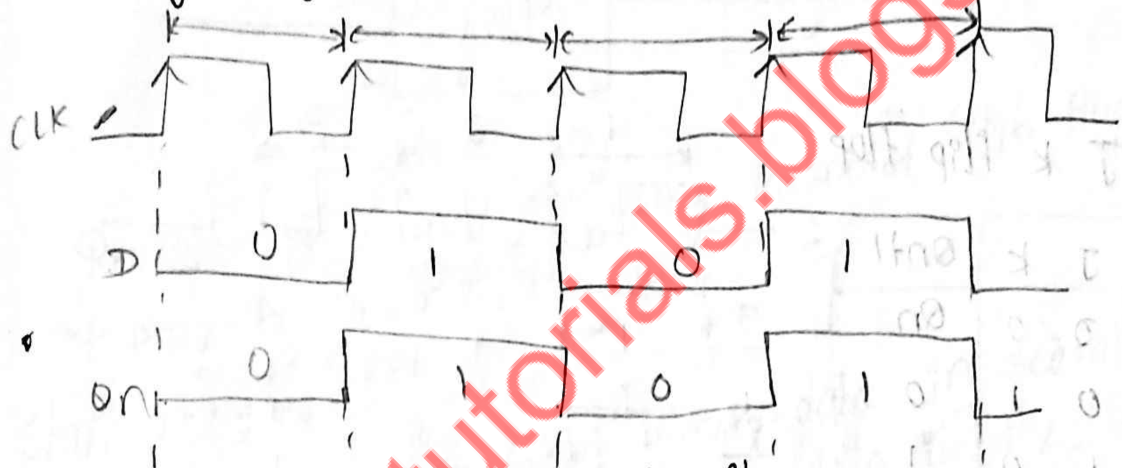
Minimized truth table for D-flip flop

D	Qn	Qn+1	
0	0	0	Reset
0	1	0	
1	0	1	set
1	1	1	

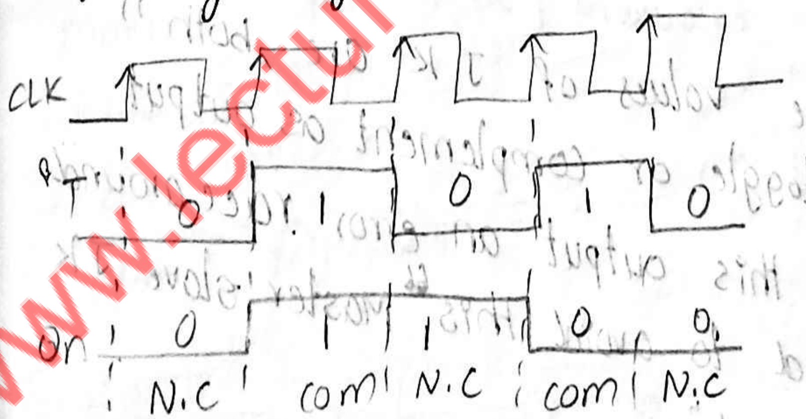
$Q_{n+1} = D$



Timing diagram for D flip flop



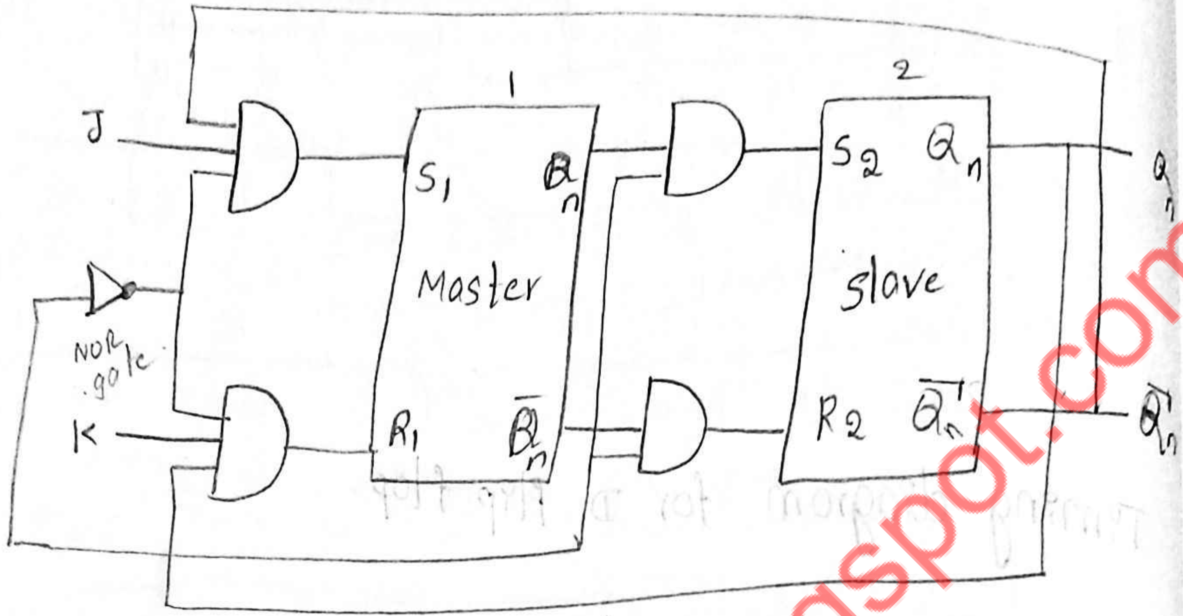
Timing diagram for T flip flop



	P.S	N.S	
T	Qn	Qn+1	
0	0	0	N.C
0	1	1	
1	0	1	comp
1	1	0	

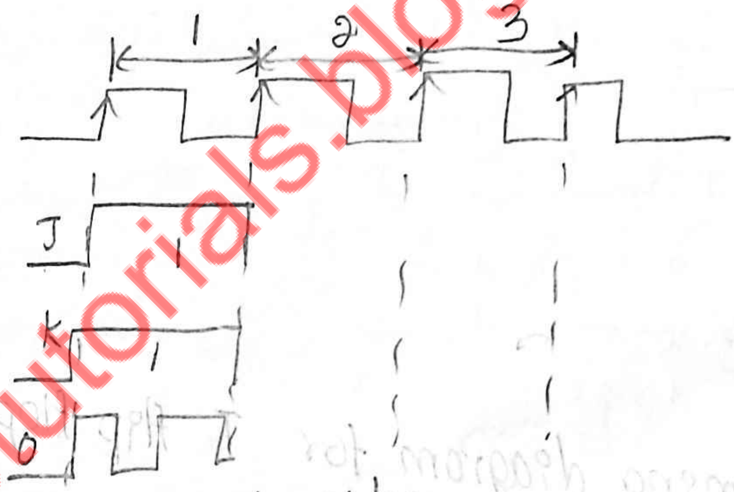
minimized truth table

Master's slave J-K flip flop



J k Flip flop

J	k	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	$\bar{Q}n$



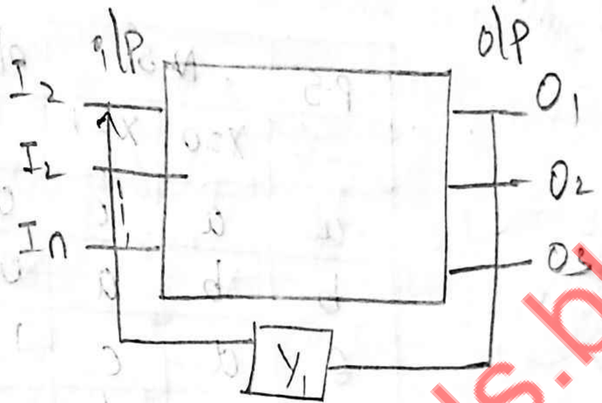
Race around problem

Here when the values of J K are both "1" we get the toggle or complement as output while occurring this output an error race around problem occurred to avoid this "Master slave JK flip flop is used".

Unit - V

Finite State Machine

Finite state machine is a model which is used to describe the synchronous sequential machine. It is a machine with fixed no. of states $i_1, i_2 --- i_n$ = input variables; $o_1, o_2, o_3 --- o_n$ = out put variables $y_1, y_2, y_3 --- y_n$ are state variables.



Mealy Moreyson machine / sequential finite state machine

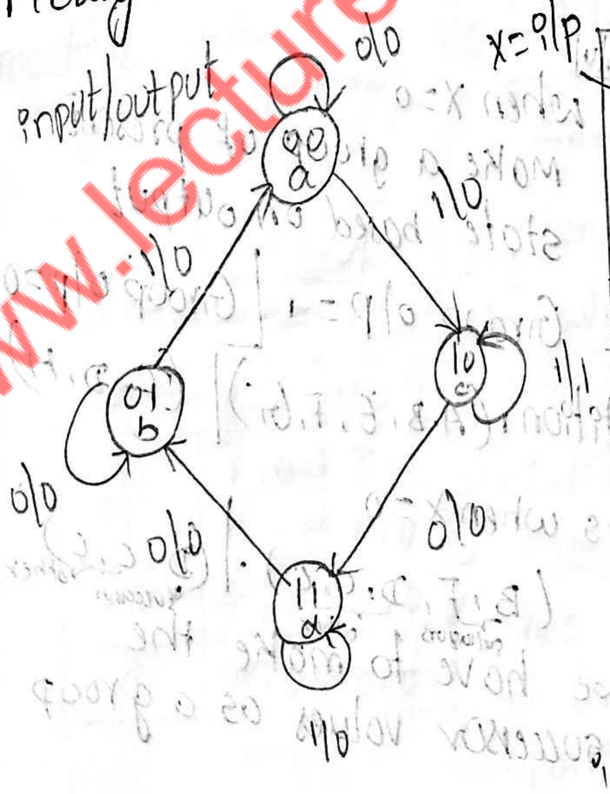
Date 17/9/19

State Diagrams

State diagram is the pictorial representation of states, present state & Next state

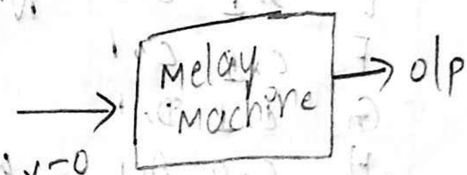
Melay Machine

State table



x = i/p

P.S	N.S x=0	N.S x=1	O/P x=0	O/P x=1
00	00	10	0	0
01	01	00	0	0
10	11	10	0	1
11	01	11	0	0



x=0 i/p (or) 1

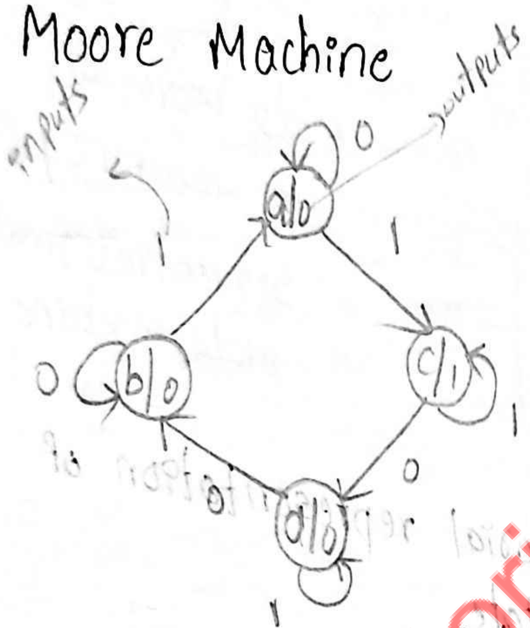
P.S → N.S

Excitation table

P.S	N.S x=0	output	N.S x=1	o/p
00	00	0	10	0
01	01	0	00	0
10	11	0	10	1
11	01	0	11	0

Note
If they didn't mention any model it is melay other wise moore

Moore Machine



State table

P.S	N.S x=0	N.S x=1	o/p
a	a	c	0
b	b	a	0
c	d	c	1
d	b	d	0

What are the conditions to minimize the state table find the minimized state table for the given table find below.

Given

P.S	N.S, Z x=0	N.S, Z x=1
A	B, 1	H, 1
B	F, 1	D, 1
C	D, 0	E, 1
D	C, 0	F, 1
E	D, 1	C, 1
F	C, 1	C, 1
G	C, 1	D, 1
H	C, 0	A, 1

N.S Z N.S Z
o/p

solu)
when x=0
Make a group of present state based on output

Group o/p = 1 | Group o/p = 0

Partition 1 (A, B, E, F, G) | (C, D, H)

N.S when x=0

(B, F, D, C, C) successors other | (D, C, C) successor other

we have to make the successor values as a group partition 2

(A, B) (E, F, G) (c) (DH) [∵ B is successor of A]

F is a successor of B

HD are successor of AB

E is not successor of c

N.S when X = 1

(HD) (cCD) (E) (FA)

$\underbrace{HD}_{\text{successor}}$ $\underbrace{(cCD)(E)(FA)}_{\text{other successors}}$

partition 3

(AB) (EFG) (c) (DH) (CD) (H)

minimized state table

A

B (x x neglect)

E = F = G

C = D

H x neglect

P.S	NS, Z	
	X=0	X=1
A	B, 1	H, 1
B	F, 1	D, 1
C	D, 0	E, 1
E	D, 1	C, 1
H	C, 0	A, 1

Date 19/9/2019

Derive a circuit that realizes the finite state machine defined by the state assignment table below using J-K flip flop.

when X = 0

Group o/p = 1 Group o/p = 0

P.S	NS, Z	
	X=0	X=1
A	B, 0	E, 0
B	E, 0	D, 0
C	D, 1	A, 0
D	C, 1	E, 0
E	B, 0	D, 0

partition 1 (C, D) (A, B, E)

N.S when X=0

(D, C) successor / not successor (B, E, B) successor

partition 2

(D) (D)(C) (AB) E

N.S when X=1

(A, E) (E, D, D) successor

partition 3

D, C (AB) E

Minimized truth table

P.S	N.S, Z	
	X=0	X=1
A	B, 0	E, 0
C	D, 1	A, 0
D	C, 1	E, 0
E	B, 0	D, 0

Implement the finite state machine using J-K flip flops [The given state table is reduced state table for the above reduced table]

Soln:

i/p	X	P.S		N.S		o/p Z	J_P	K_P	J_Q	K_Q
			PQ		$P_{t+1}Q_{t+1}$		P		Q	
0	0	A	00	B	01	0	0	X	1	X
1	0	B	01	B	01	0	0	X	X	0
2	0	C	10	D	11	1	X	0	1	X
3	0	D	11	C	10	1	X	0	X	1
4	1	A	00	B	01	0	0	X	1	X
5	1	B	01	D	11	0	1	X	X	0
6	1	C	10	A	00	0	X	1	0	X
7	1	D	11	B	01	0	X	1	X	0

Given minimized table

P.S	N.S, Z	
	X=0	X=1
A	B, 0	B, 0
B	B, 0	D, 0
C	D, 1	A, 0
D	C, 1	B, 0

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

Excitation table of J K flipflop

x \ pq	00	01	11	10
0	1	X	X	1
1	1	X	X	

$J_q = \bar{x} + \bar{p}$

x \ pq	00	01	11	10
0	X		1	X
1	X			X

$K_q = \bar{x} \cdot p$

x \ pq	00	01	11	10
0			X	X
1		1	X	X

$J_p = xq$

x \ pq	00	01	11	10
0	X	X		
1	X	X	1	1

$K_p = x$

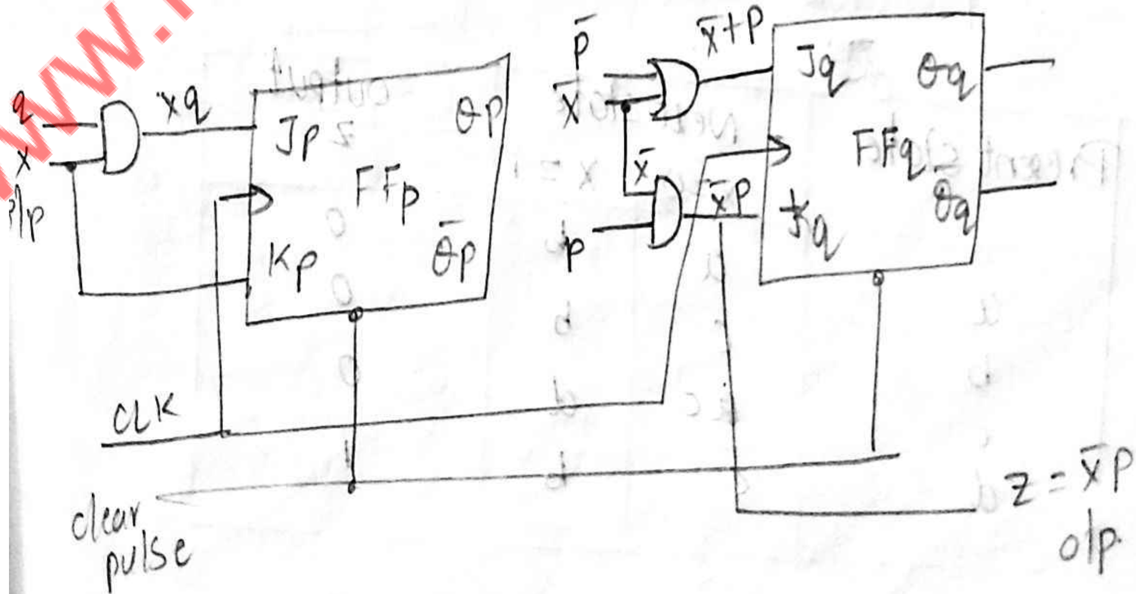
Here x, p, q are inputs.

o/p : z

x \ pq	00	01	11	10
0			1	1
1				

$z = \bar{x}p$

logic circuit



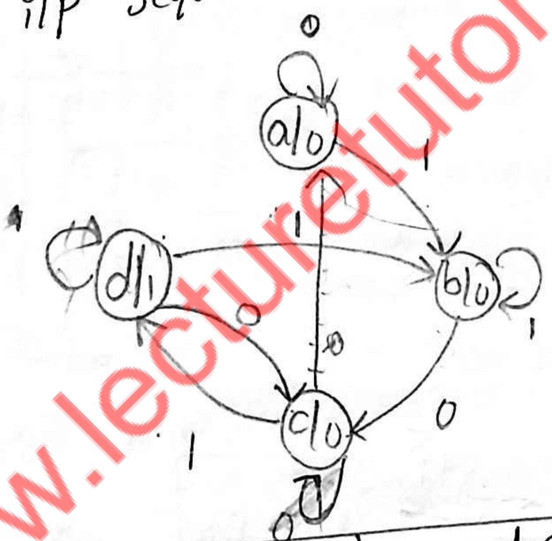
1. If the simple example explain the difference between melay and Moore type machine (unit-V)
2. Draw the diagram of Moore type FSM (finite state machine) for serial adder (unit-V)
3. Draw the diagram of melay type state machine for serial adder and explain its operation (unit V)
4. what are the capabilities and limitations of finite state machine explain (unit V)

Assignment

Date 20/9/19 Sequence Detector

1. Design a Moore type sequence detector to detect a serial input sequence of 101

o/p sequence $\overset{S_1 S_2 S_3}{101}$

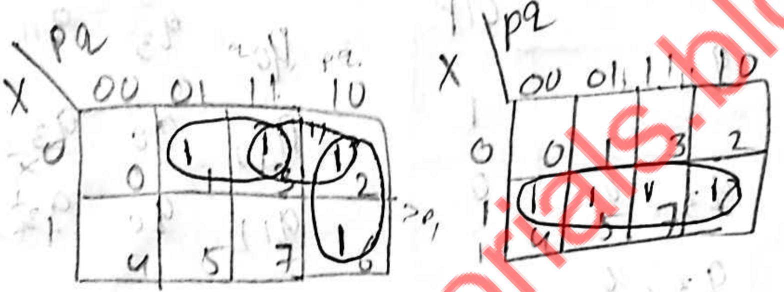


Excitation table of d F/F

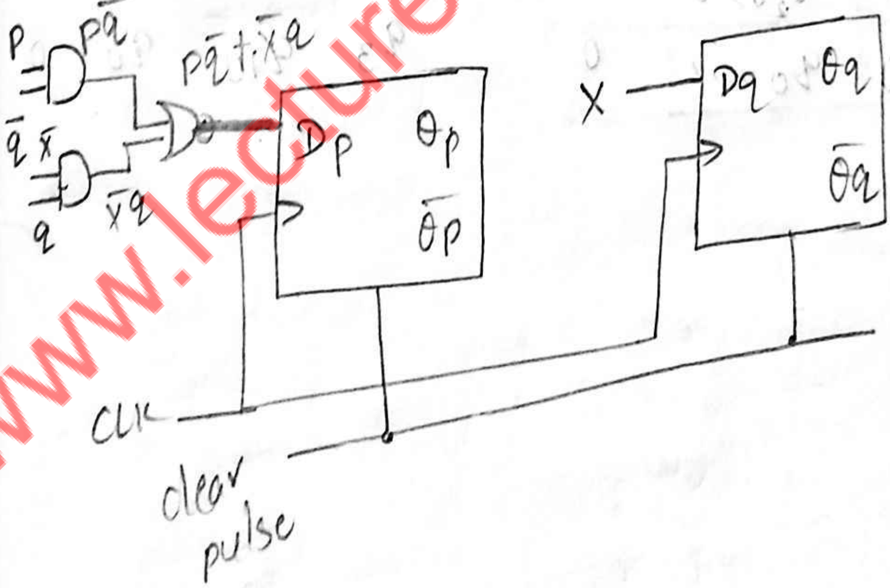
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Present state	Next state x=0	Next state x=1	output z
a	a	b	0
b	c	b	0
c	a c	d	0
d	c	b	1

	x i/p		P.S p q		N.S p_{t+1} q_{t+1}	Dp	Dq
0	0	a	0 0	a	0 0	0	0
1	0	b	0 1	c	1 0	1	0
2	0	c	1 0	c	1 0	1	0
3	0	d	1 1	c	1 0	1	0
4	1	a	0 0	b	0 1	0	1
5	1	b	0 1	b	0 1	0	1
6	1	c	1 0	d	1 1	1	1
7	1	d	1 1	b	0 1	0	1



$Dp = p\bar{q} + \bar{x}q$ $\qquad Dq = x.$



Date 21/9/19

Melay to Moore Machine

P.S	N.S , z x=0	x=1
q_0	$q_3, 0$	$q_1, 1$
q_1	$q_0, 1$	$q_3, 0$
q_2	$q_2, 1$	$q_2, 0$
q_3	$q_1, 0$	$q_0, 1$

State table

$q_0, 1$ | $q_1, 1$ | $q_{11}, 1$

$q_0, 1$ | $q_1, 0$ | $q_{10}, 0$

No change

$q_3, 0$ $q_2, 1$ | $q_{21}, 1$

$q_3, 0$ $q_2, 0$ | $q_{20}, 0$

No-change

P.S	N.S , z x=0	x=1	o/p
q_0	$q_3, 0$	$q_{11}, 1$	1
q_{10}	$q_0, 1$	$q_3, 0$	0
q_{11}	$q_0, 1$	$q_3, 0$	1
q_{20}	$q_{21}, 1$	$q_{20}, 0$	0
q_{21}	$q_{21}, 1$	$q_{20}, 0$	1
q_3	$q_{10}, 0$	$q_0, 1$	0

Moore Machine

P.S	x=0	x=1	o/p
q_0	q_3	q_{11}	1
q_{10}	q_0	q_3	0
q_{11}	q_0	q_3	1
q_{20}	q_{21}	q_{20}	0
q_{21}	q_{21}	q_{20}	1
q_3	q_{10}	q_0	0

Date 16/8/19

Unit - VI
Registers and Counters

Register

A register is a device which is used to store the data in binary format and shifts the binary bits in the serial manner and also we can collect the information in the same binary format. It is called bit by bit.

→ store the binary data

Register

temporary

Register is formed by using the no.of flip flops it may be D or T flip flop.

shift register

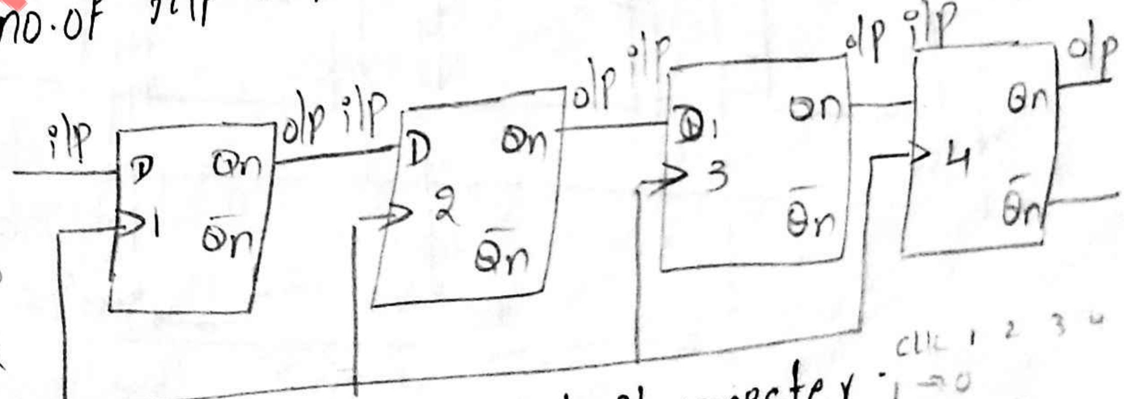
It shifts the given binary data in serial manner and we can collect the data in the same binary manner

4 bit register

no.of flip flops = 4 ; we may use "D" flip flop

1) serial i/p

ex: 1010

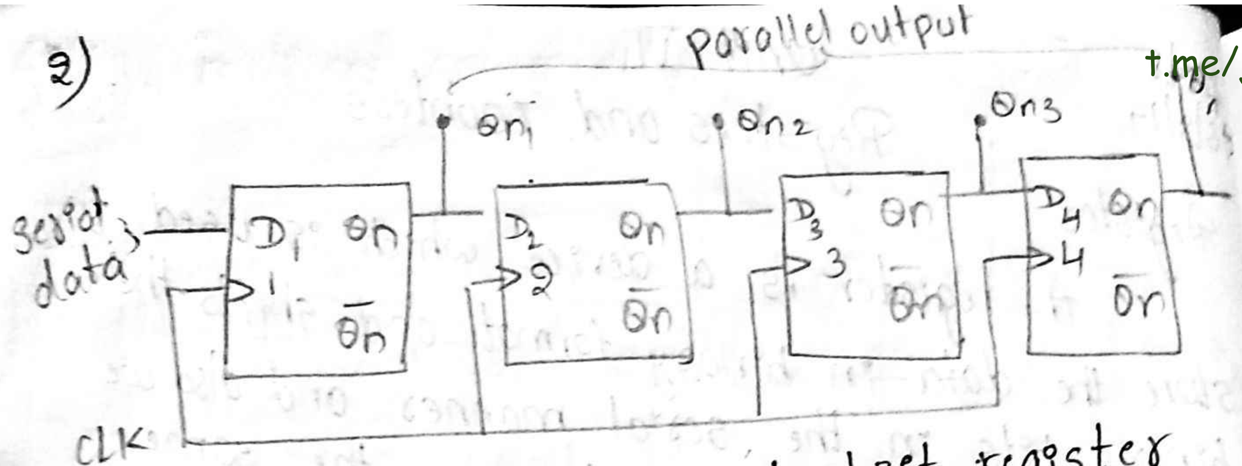


clk

Serial in - serialout shift register

∴ in means = entry of data

2)



Serial - in - parallel-out shift register

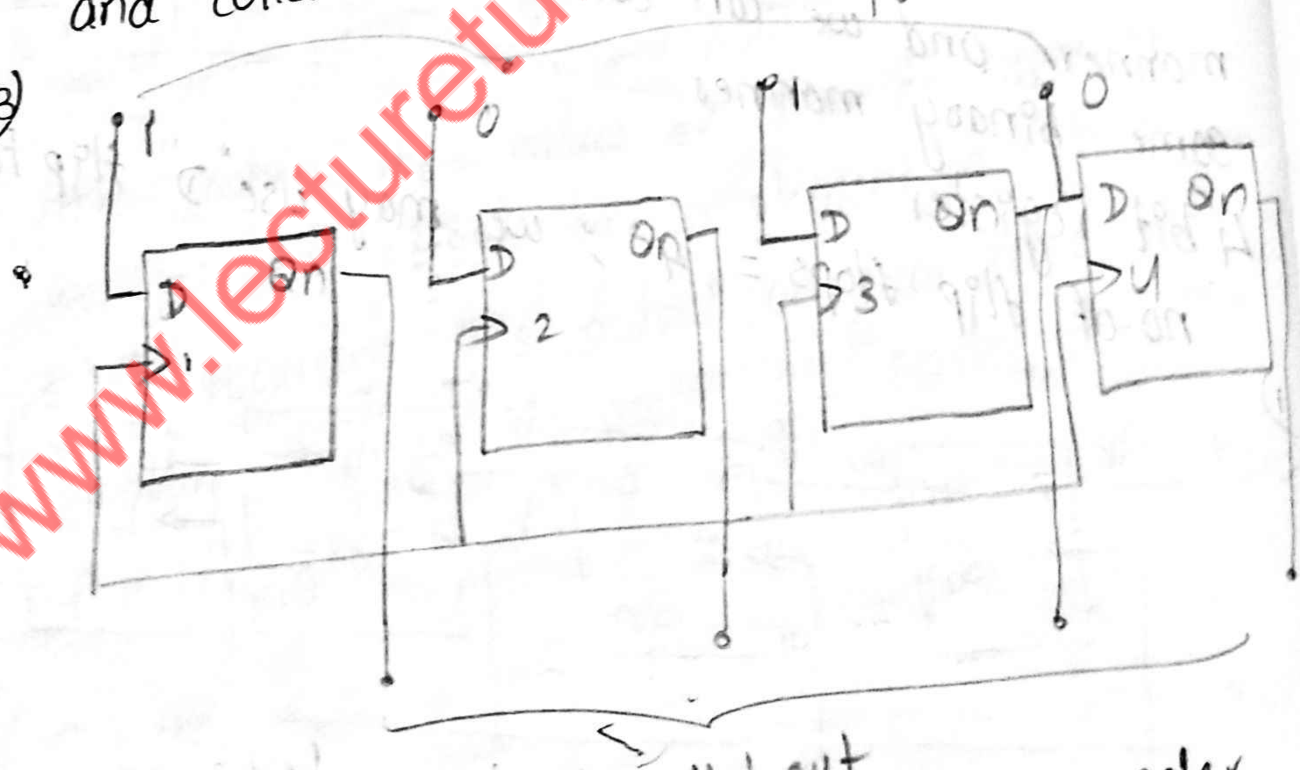
4 3 2 1
10 10

1010 →

CLK	ops across f/f			
	D_1	D_2	D_3	D_4
0	–	–	–	–
1	0	–	–	–
2	1	0	–	–
3	0	1	0	–
4	1	0	1	0

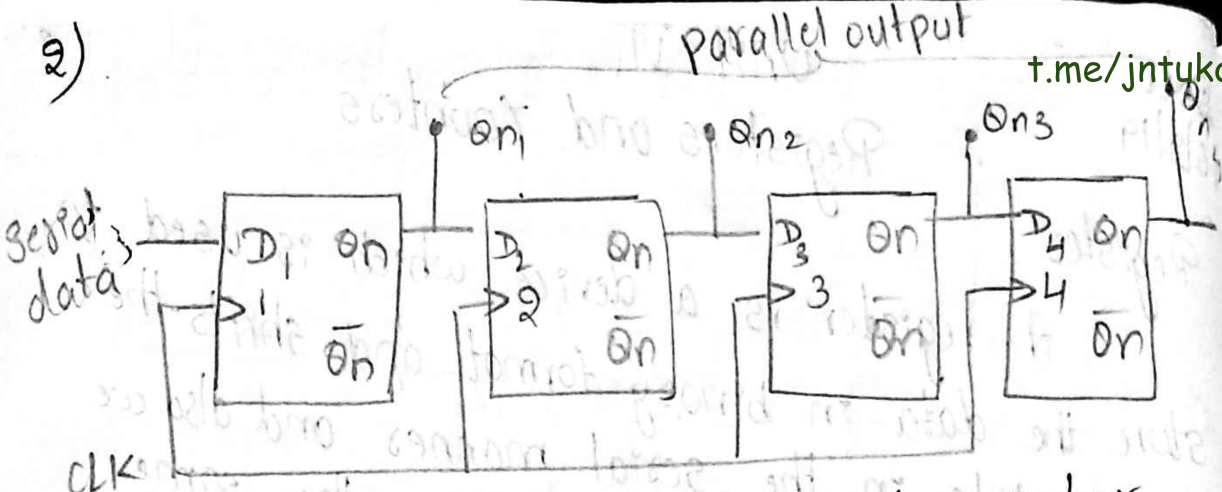
In this register the data entered in serial and collected in parallel out

3)



Parallel - in - parallel - out shift register

2)



Serial - in - parallel - out shift register

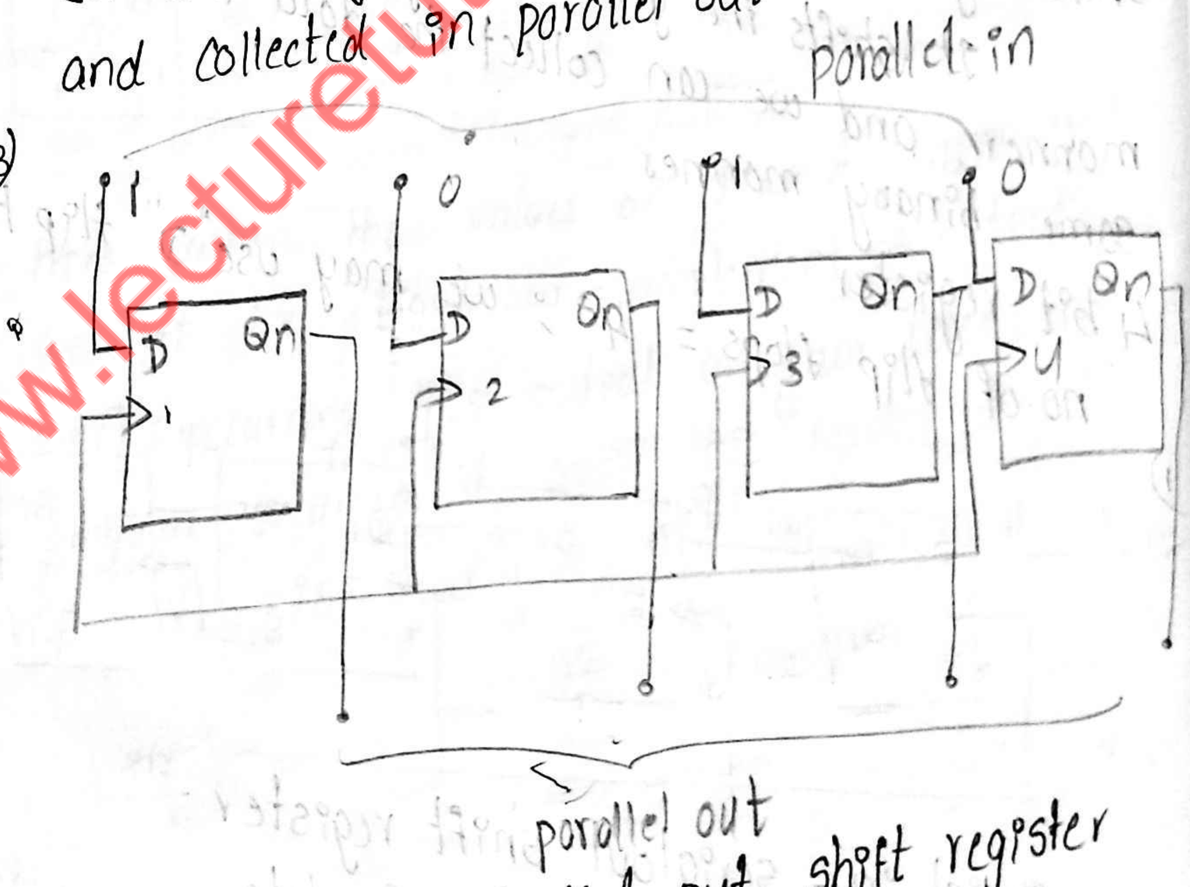
	ops across f/f			
CLK	D1	D2	D3	D4
0	–	–	–	–
1	0	–	–	–
2	1	0	–	–
3	0	1	0	–
4	1	0	1	0

4 3 2 1
1 0 1 0

1010 →

In this register the data entered in serial and collected in parallel out

3)

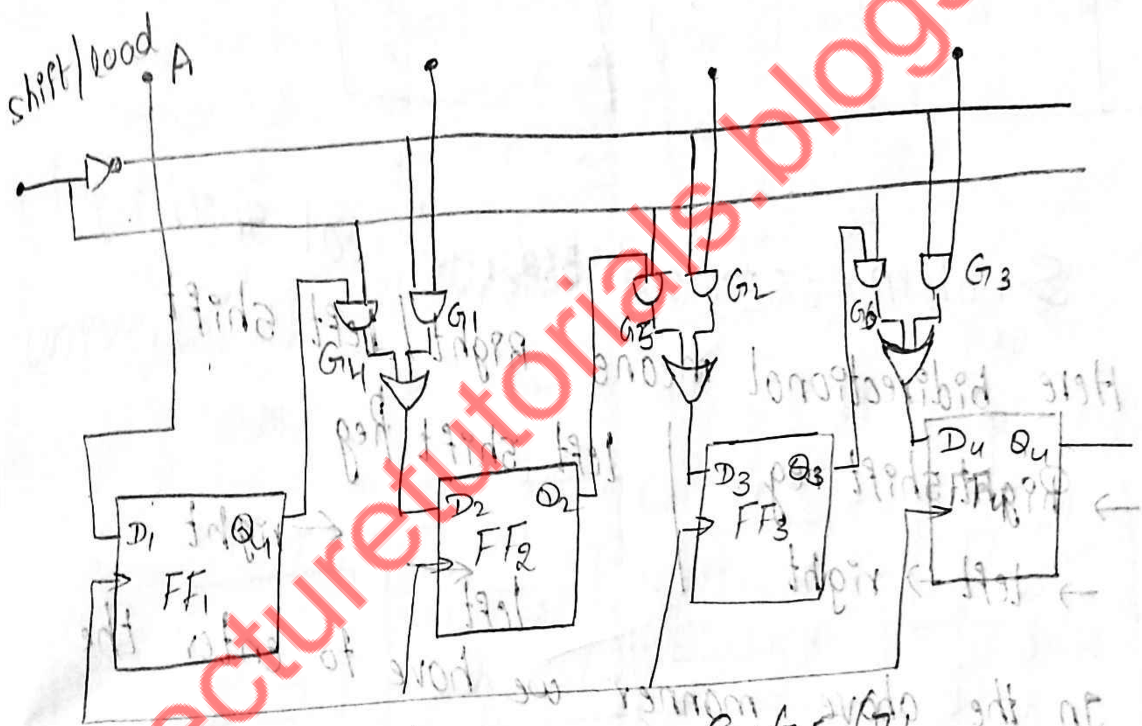


Parallel - in - parallel - out shift register

CLK	D_1	D_2	D_3	D_4
0	–	–	–	–
1	1	0	1	0
2	Q_1	Q_2	Q_3	Q_4
	1	0	1	0

Note: The input in parallel and collected the output parallel

4) Parallel-in serial out shift register



$G_1 G_2 G_3$ $\quad$ $G_4 G_5 G_6$

shift/$\overline{load}$ = 1 $\quad$ 0 0 0

Disable $\quad$ enable

load

shift/$\overline{load}$ = 0 $\quad$ shift 1 1 1 $\quad$ 0 0 0

G_1 G_2 G_3

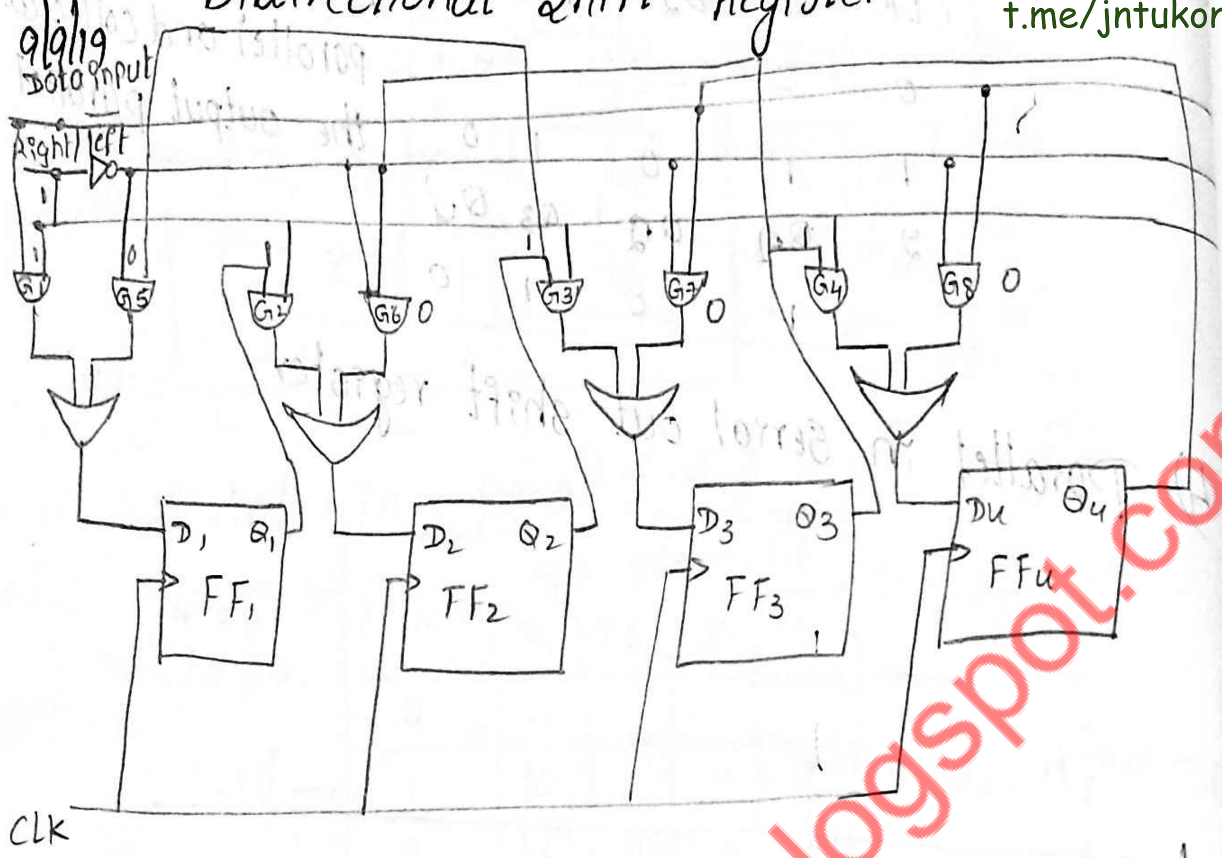
(out) ← A B C D

G_4 G_5 G_6

Q_1 Q_2 Q_3

Date 9/9/19

Bidirectional Shift Register (D- flip flop)



Serial-in - serial out Bidirectional shift register

Here bidirectional means Right / left shift

→ Right shift reg | left shift Reg

→ left → right | ← right left

In the above manner we have to enter the data according to arrow 'marks'

① Right / left = 1

G1	G2	G3	G4	G5	G6	G7	G8
1	1	1	1	0	0	0	0
enabled state				disabled state			

Example

Data input 1111 left

CLK	D1	D2	D3	D4	
				0	→ initial state
0	0	0	0	0	→ (out)
1	→ 1	→ 0	→ 0	→ 0	
2	→ 1	→ 1	→ 0	→ 0	
3	→ 1	→ 1	→ 1	→ 0	
4	→ 1	→ 1	→ 1	→ 1	

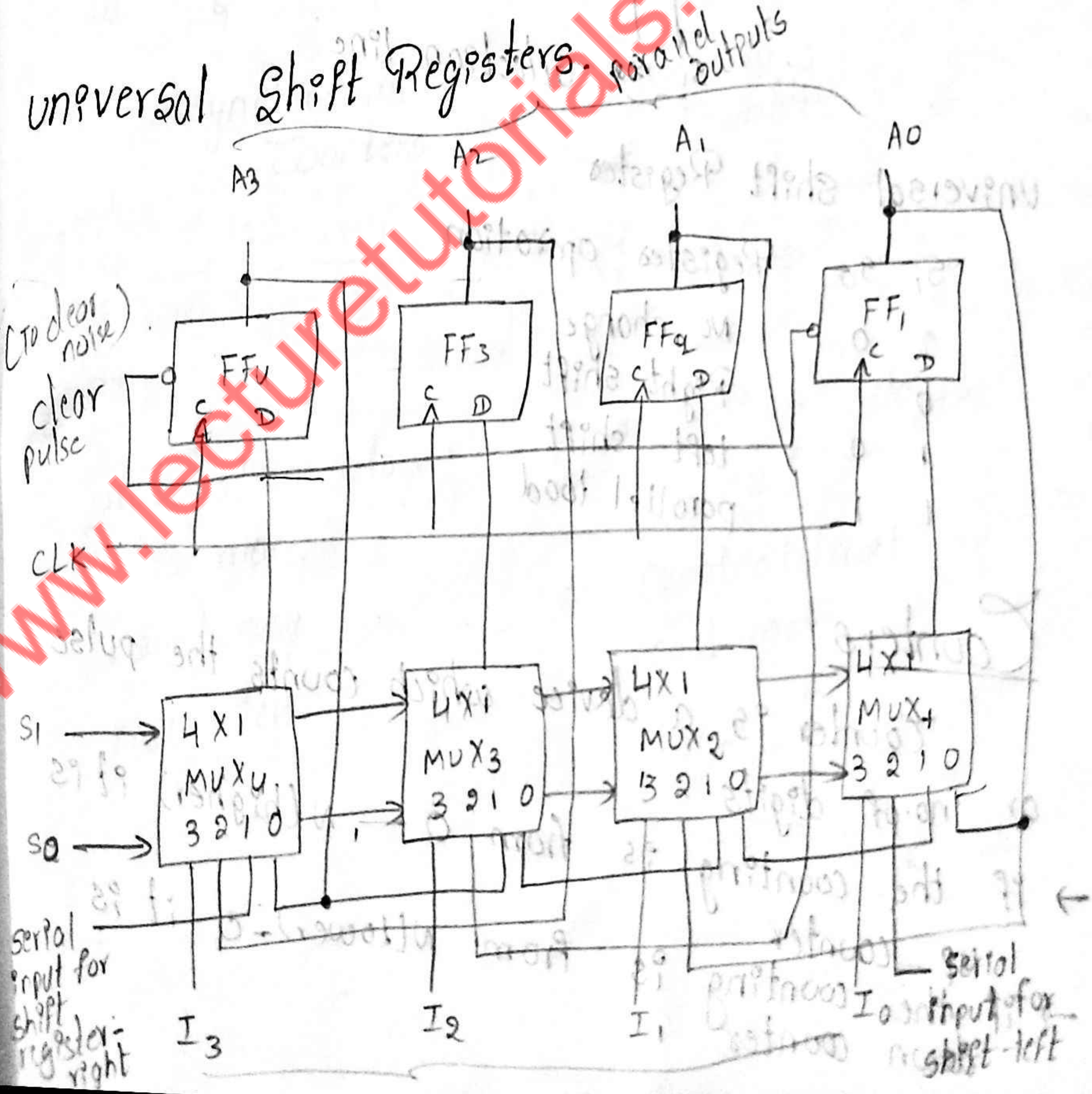
② right/$\overline{left}$ = 0

G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
0	0	0	0	1	1	1	1
disabled				enabled			

example

CLK	G_5 D_1		G_6 D_2		G_7 D_3		G_8 D_4	
0	0	←	0	←	0	←	0	
1	0	←	0	←	0	←	1	← data input right 1111
2	0	←	0	←	1	←	1	
3	0	←	1	←	1	←	1	
4	1	←	1	←	1	←	1	

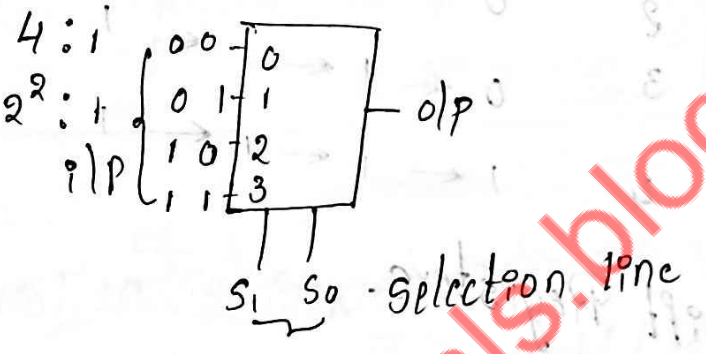
universal Shift Registers



* Enter the data in serial and collecting the data in parallel called universal shift registers.
* universal shift register consists of 'D' flip flops and multiplexers
* Multiplexer : which has 2^n inputs and 1 output is called multiplexer

$2^n : 1$ where n = no.of selection line.
i/p o/p

Ex:- $4:1$
$2^2 : 1$



00 – 0
01 – 1
11 – 2
10 – 3

Universal shift Register

S_1	S_0	Register operation
0	0	No change
0	1	Right shift
1	0	left shift
1	1	parallel load

Counters

Counter is a device which counts the pulse or no.of digits

→ if the counting is from 0 — N(higher) it is up counter

→ if the counting is from N(lower) – 0 it is down counter

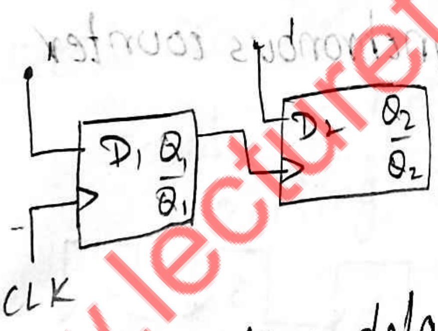
Counters are of two types
→ synchronus counters
→ Asynchronus counters

CLK	decimal count
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9

Asynchronus counters
In asynchronous counters, the output of first flip flop is given to the CLK as second flip flop and propagation delay is accured

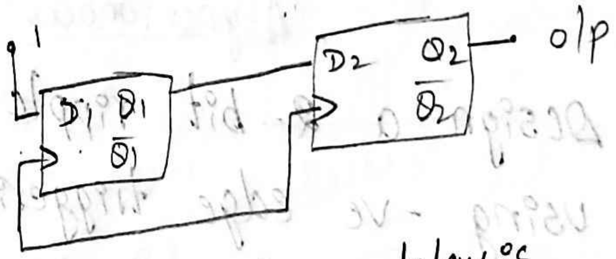
Synchronous counters
In synchronous counters, the CLK pulse is same and no propagation delay

Asynchronous Counters



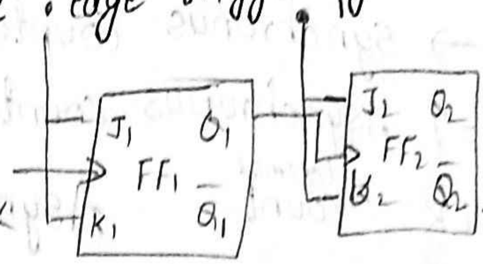
propagation delay is more
count easy
cost less

Synchronous counters



propagation delay is less
count difficult
cost more

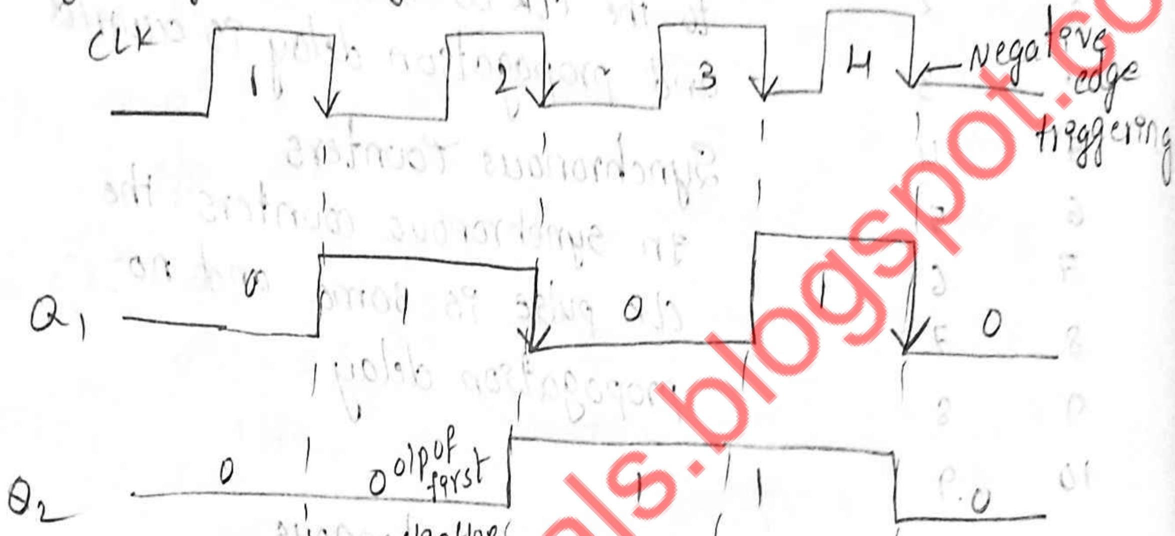
1. Design a two bit ripple up-counter using J-K flip flop by negative edge triggering



no.of bits $n = 2$

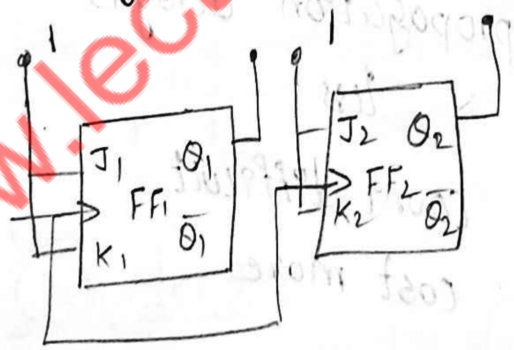
no.of flip flops = 2 (J K F/F)

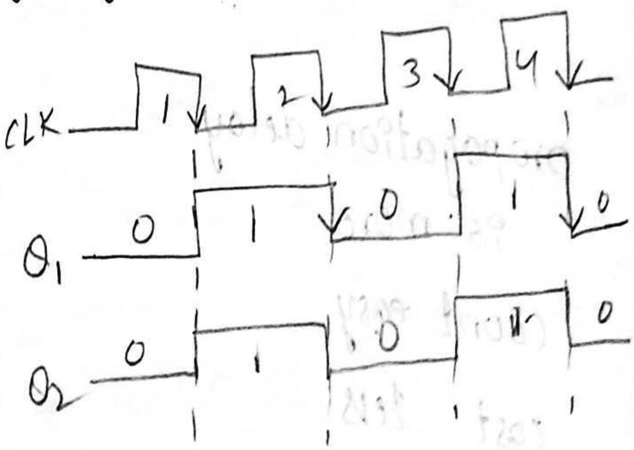
By using '-ve' edge triggering



Asynchronous counter

2. Design a 2-bit ripple up-synchronous counter using -ve edge triggering





n = no.of bits = 2

no.of F/F = 2 (J K F/F)

synchronous counter by -ve edge triggering

Design of Synchronous Counters

It is represented as bit counter or divide by ~~ten~~ counter or modulo ~~ten~~ countor or mod ~~ten~~ counter.

Design a decade counter or mod 10 counter using T flip flop.

Counters:

n bit counter
Modulo-N counter
MOD-N counter
Divide by N counter

N = no. of counter.

$N = 0$ to $N-1$

Ex: $N = 10$

MOD = 10 counter

$N = 10$ (0-9)

counting sequence

no. of f/fs are n.

Decade Counter

Count the numbers = $N = 10$

Counting sequence $(0-(N-1))$
$(0-9)$

no. of f/f required to construct the counter

n = no. of f/fs

$2^n \geq N$

$n = 4$

$n=0 \quad 2^0 \ngeq 10$

$n=1 \quad 2^1 \ngeq 10$

$n=2 \quad 2^2 \ngeq 10$

$n=3 \quad 2^3 \ngeq 10$

$n=4 \quad 2^4 \geq 10$

CLK	Dc no		present state Q_D	Q_C	Q_B	Q_A		Next state Q_{D+1}	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_D	T_C	T_B	T_A
1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
2	1	1	0	0	0	1	2	0	0	1	0	0	0	1	1
3	2	2	0	0	1	0	3	0	0	1	1	0	0	0	1
4	3	3	0	0	1	1	4	0	1	0	0	0	1	1	1
5	4	4	0	1	0	0	5	0	1	0	1	0	0	0	1
6	5	5	0	1	0	1	6	0	1	1	0	0	0	1	1
7	6	6	0	1	1	0	7	0	1	1	1	0	0	0	1
8	7	7	0	1	1	1	8	1	0	0	0	1	1	1	1
9	8	8	1	0	0	0	9	1	0	0	1	0	0	0	1
10	9	9	1	0	0	1	0	0	0	0	0	1	0	0	1

Excitation table of T F/F

P.S Q_n	N.S Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

T_A

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	1_0	1_1	1_3	1_2
01	1_4	1_5	1_7	1_6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_8	1_9	X_{11}	X_{10}

double octent

$T_A = 1$

T_B

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1_1	1_3	2
01	4	1_5	1_7	6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	8	9	X_{11}	X_{10}

quard

$$\text{quard} = (\bar{Q}_D \bar{Q}_C + \bar{Q}_D Q_C)(Q_B Q_A + \bar{Q}_B Q_A)$$

$$= \bar{Q}_D (\bar{Q}_C + Q_C) \cdot Q_A (Q_B + \bar{Q}_B)$$

$$= \bar{Q}_D Q_A$$

T_C

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	1_3	2
01	4	5	1_7	6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	8	9	X_{11}	10

$T_C = Q_B Q_A$

T_D

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	3	2
01	4	5	1_7	6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	8	1_9	X_{11}	X_{10}

quard $= Q_D Q_A$

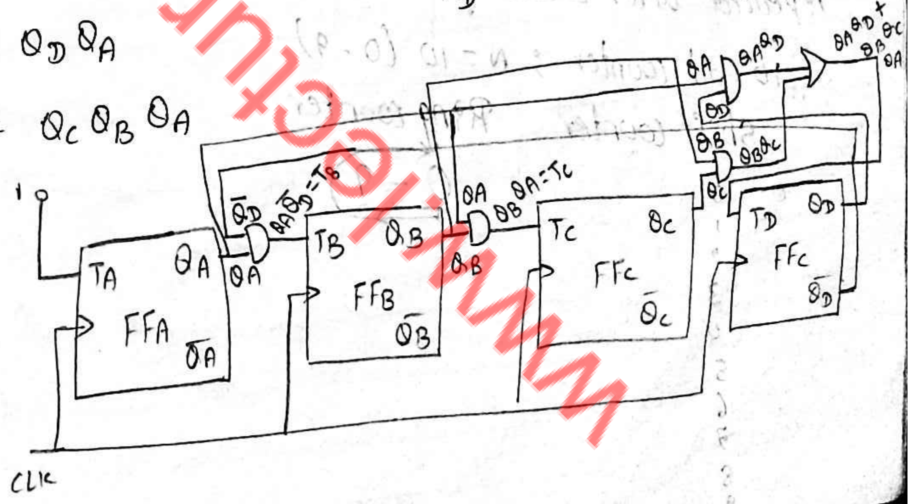
Pair $= Q_C Q_B Q_A$

$T_A = 1$

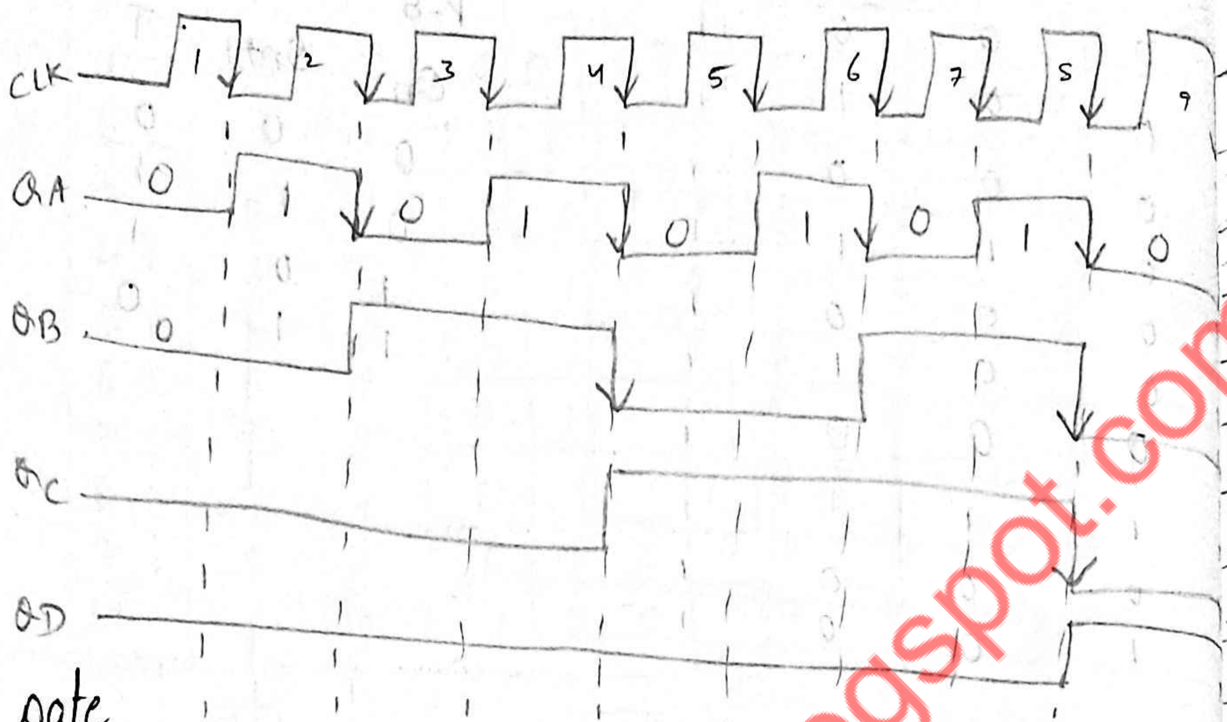
$T_B = \bar{Q}_D Q_A$

$T_C = Q_B Q_A$

$T_D = Q_D Q_A + Q_C Q_B Q_A$



Timing Diagram



Date
12/9/19

Design a 10 bit ring and shift counter with D flip flops

10 bit shift and ring counter

It is called shift counter because it shifts the value from present to next

It is called ring counter when the cycle is repeated after counting the maximum value

10 bit counter ; N = 10 (0 - 9)

shift counter — Ring counter

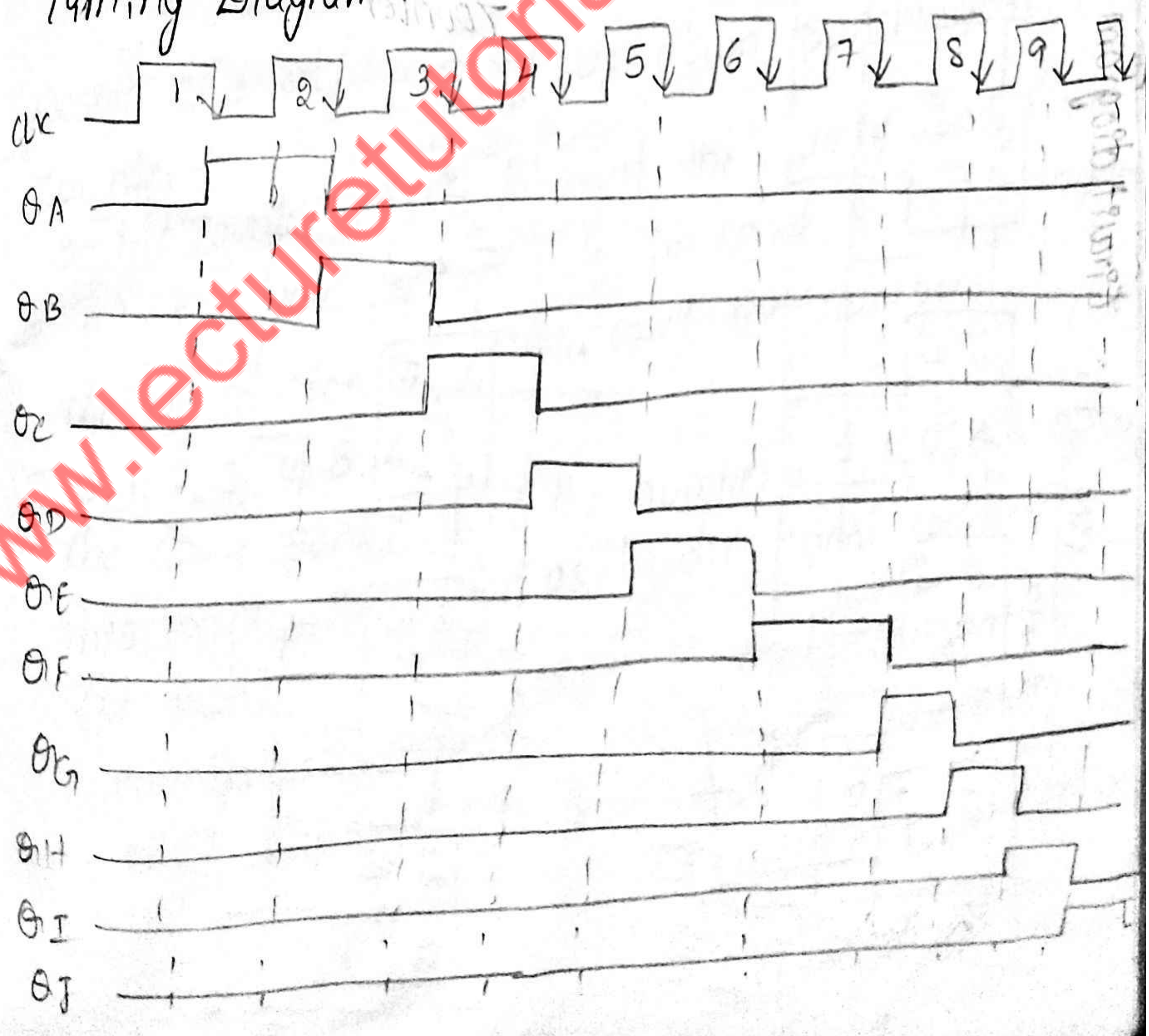
0 – 0
1
2
3
4
5
6
7
8
9

0 – 9

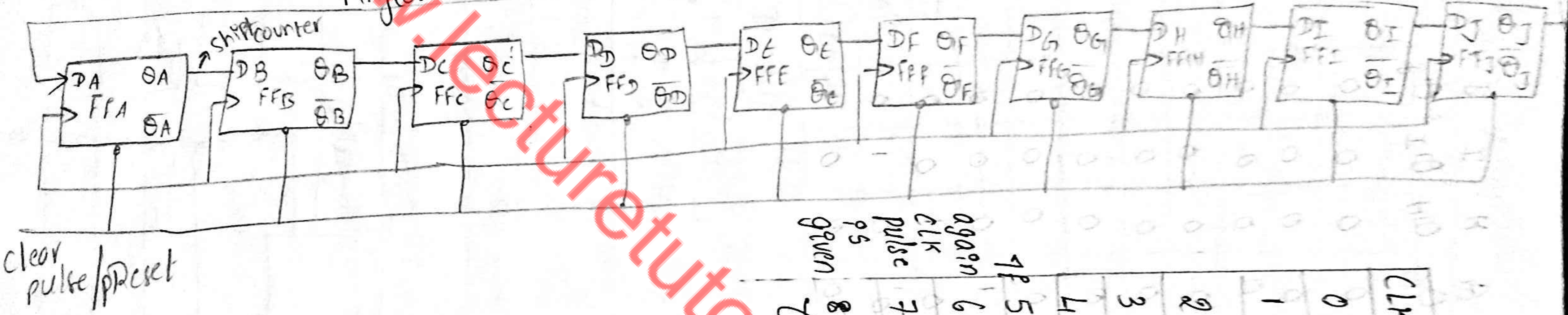
Truth table

CLK	Decimal number	A Q_A	B Q_B	C Q_C	D Q_D	E Q_E	F Q_F	G Q_G	H Q_H	I Q_I	J Q_J
0		0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0
3	2	0	0	1	0	0	0	0	0	0	0
4	3	0	0	0	1	0	0	0	0	0	0
5	4	0	0	0	0	1	0	0	0	0	0
6	5	0	0	0	0	0	1	0	0	0	0
7	6	0	0	0	0	0	0	1	0	0	0
8	7	0	0	0	0	0	0	0	1	0	0
9	8	0	0	0	0	0	0	0	0	1	0
10	9	0	0	0	0	0	0	0	0	0	1

Timing Diagram



Circuit diagram for shift and ring counter

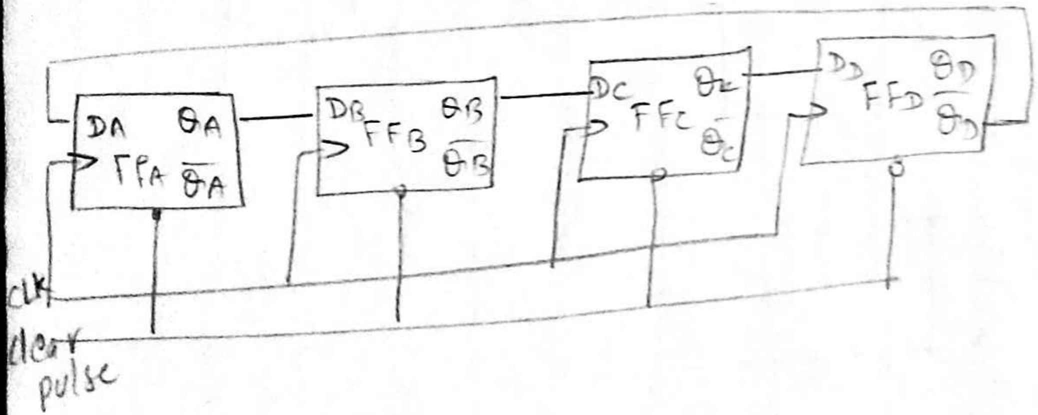


Twisted ring counter

CLK	Q_A	Q_B	Q_C	Q_D	$\bar{Q}_D$
0	0	0	0	0	1
1	1	0	0	0	1
2	1	1	0	0	1
3	1	1	1	0	1
4	1	1	1	1	0
5	0	1	1	1	0
6	0	0	1	1	0
7	0	0	0	1	0
8					

If again clk pulse is given

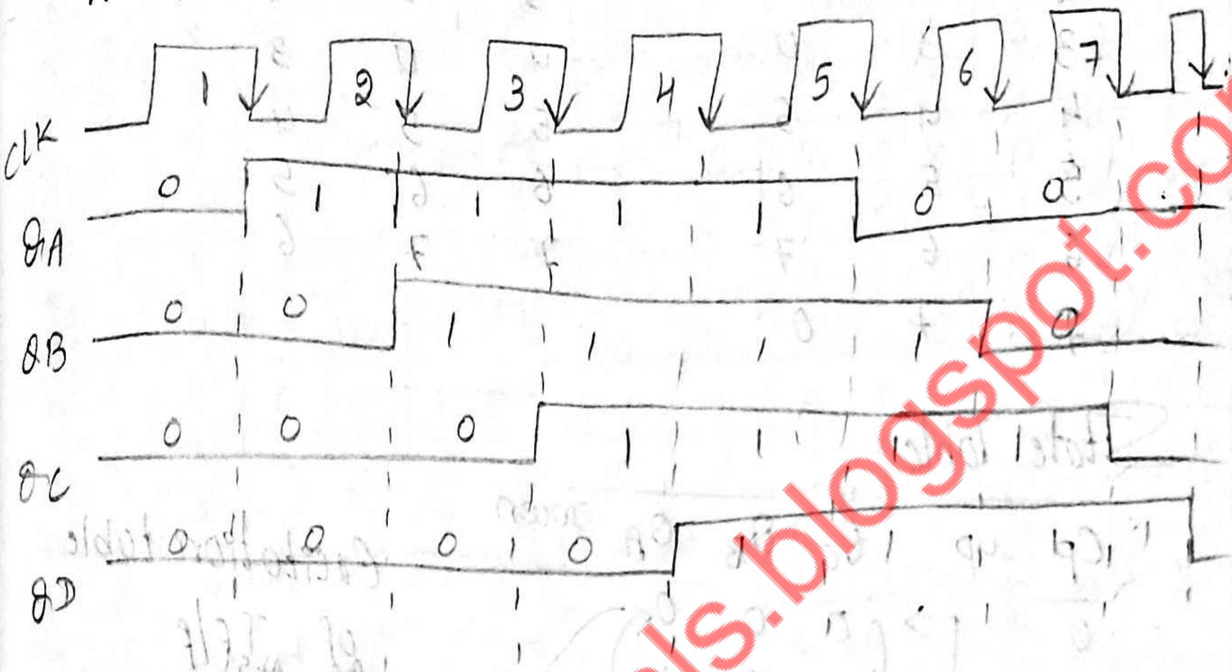
Circuit diagram for Johnson counter



Design a 4-bit Johnson counter or a 4-bit shift and twisted ring counter with D flip flops

4 bit Johnson counter / 4 bit and twisted ring counter

$N = 4\ (0-3)$ $[4+4 = 8\ (0-7)]$



Timing Diagram

Design a 3-bit [down-5] up-down synchronous counter

3-bit counter enable to count the numbers as $2^n = N$

$2^3 = 8$ you can be able to count 8 i.e., (0-7)

The up-counter can able to count the numbers 0, 1, 2, 3, 4, 5, 6, 7, 0..

The down counter count the numbers 7, 6, 5, 4, 3, 2, 1, 0

This counter is known as bi-directional counter (up-down)

counting = N. sequence

no.of flip flops = n = 3

Mode signal

up/$\overline{\text{DOWN}}$ = 1 → 1 0 ON OFF	UP counter
up/$\overline{\text{down}}$ = 0 → 0 1 OFF ON	down counter

Date 13/9/19

Up Counter

	P.S	N.S
0	0	1
1	1	2
2	2	3
3	3	4
4	4	5
5	5	6
6	6	7
7	7	0

Down Counter

	P.S	N.S
0	0	7
1	1	0
2	2	1
3	3	2
4	4	3
5	5	4
6	6	5
7	7	6

State Table

Cp	up	Q_C	Q_B	Q_A	Down
0		0	0	0	
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

Excitation table of T F/F

Q_n (P.S)	Q_{n+1} (N.S)	T
0	0	0
0	1	1
1	0	1
1	1	0

For K-map

$$T_A = \sum_m (0, 1, 2, \ldots 15)$$

$$T_B = \sum_m (0, 2, 4, 6, 9, 11, 13, 15)$$

$$T_C = \sum_m (0, 4, 11, 15)$$

Excitation table

	$U/\overline{D}$		present state Q_C	Q_B	Q_A	no. used	Next state Q_{C+1}	Q_{B+1}	Q_{A+1}		T_C	T_B	T_A
0	0	0	0	0	0	0	1	1	1	7	1	1	1
1	0	1	0	0	1	1	0	0	0	0	0	0	1
2	0	2	0	1	0	2	0	0	1	1	0	1	1
3	0	3	0	1	1	3	0	1	0	2	0	0	1
4	0	4	1	0	0	4	0	1	1	3	9	1	1
5	0	5	1	0	1	5	1	0	0	4	0	0	1
6	0	6	1	1	0	6	1	0	1	5	0	1	1
7	0	7	1	1	1	7	1	1	0	6	0	0	1
8	1	0	0	0	0	8	0	0	1	1	0	0	1
9	1	1	0	0	1	9	0	1	0	2	0	9	1
10	1	2	0	1	0	10	0	1	1	3	0	0	1
11	1	3	0	1	1	11	1	0	0	4	1	9	1
12	1	4	1	0	0	12	1	0	1	5	0	0	1
13	1	5	1	0	1	13	1	1	0	6	0	9	1
14	1	6	1	1	0	14	1	1	1	7	0	0	1
15	1	7	1	1	1	15	0	0	0	0	1	9	

T_A

$U/\overline{D}\,Q_C \backslash Q_BQ_A$	00	01	11	10
00	1_0	1_1	1_3	1_2
01	1_4	1_5	1_7	1_6
11	1_{12}	1_{13}	1_{15}	1_{14}
10	1_8	1_9	1_{11}	1_{10}

T_B

$U/\overline{D}\,Q_C \backslash Q_BQ_A$	00	01	11	10
00	1_0	$_1$	$_3$	1_2
01	1_4	$_5$	$_7$	1_6
11	$_{12}$	1_{13}	1_{15}	$_{14}$
10	$_8$	1_9	1_{11}	$_{10}$

T_C

$U/\overline{D}\,Q_C \backslash Q_BQ_A$	00	01	11	10
00	1_0	$_1$	$_3$	$_2$
01	1_4	$_5$	$_7$	$_6$
11	$_{12}$	$_{13}$	1_{15}	$_{14}$
10	$_8$	$_9$	1_{11}	$_{10}$

$T_A = 1$

$q_1 \rightarrow U/D\,Q_A$

$q_2 \rightarrow \overline{U/D}\,\overline{Q_A}$

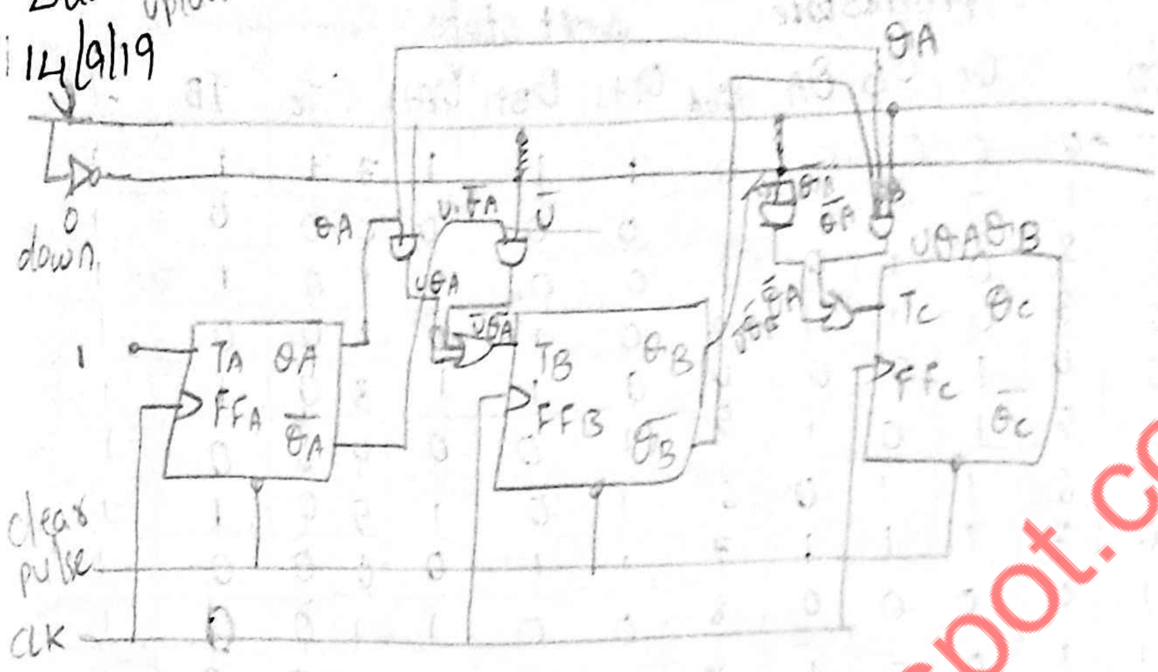
$p_1 \rightarrow \overline{U/D}\,\overline{Q_B}\,\overline{Q_A}$

$p_2 \rightarrow U/D\,Q_B\,Q_A$

$T_B = U/D\,Q_A + \overline{U/D}\,\overline{Q_A}$

$T_C = \overline{U/D}\,\overline{Q_B}\,\overline{Q_A} + U/D\,Q_B\,Q_A$

Date
14/9/19



logic diagram

Design divide by "6" counter using T flip flop

Mod 6 counter

Divide by 6 counter (or) MOD 6 - counter

no. of counts = 6 , counting range $0-(N-1)$ $(0-5)$

n = no. of flip flops $2^n \geq N$

$n=1$ $2^1 \ngeq 6$

$n=2$ $2^2 \ngeq 6$

$n=3$ $2^3 \geq 6$

No. of flip flops = 3

No. of bits = no. of flip flops = 3

Note

MOD - 6 (0-5) ; $n=3$ (0-7) 6, 7 are don't care

- 5 (0-4) ; $n=3$ (0-7) (5, 6, 7) don't care

P.S	N.S
0	1
1	2
2	3
3	4
4	5
5	6

CLK	D.no	Q_C (P.S)	Q_B (P.S)	Q_A (P.S)		Q_{C+1} (N.S)	Q_{B+1} (N.S)	Q_{A+1} (N.S)	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1	0	0	1
1	1	0	0	1	2	0	1	0	0	1	1
2	2	0	1	0	3	0	1	1	0	0	1
3	3	0	1	1	4	1	0	0	1	1	1
4	4	1	0	0	5	1	0	1	0	0	1
5	5	1	0	1	0	0	0	0	1	0	1
6	6	X	X	X							
7	7	X	X	X							

Excitation table of T f/f

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

$T_A = \sum m(0,1,2,3,4,5)$

$T_B = \sum m(1,3)$

$T_C = \sum m(3,5)$

T_A

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	1 (0)	1 (1)	1 (3)	1 (2)
1	1 (4)	1 (5)	X (7)	X (6)

$T_A = 1$

T_B

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	(0)	1 (1)	1 (3)	(2)
1	(4)	(5)	X (7)	X (6)

$T_B = \bar{Q}_C \cdot Q_A$

T_C

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	(0)	(1)	1 (3)	(2)
1	(4)	1 (5)	X (7)	X (6)

$\to P_2$

P_1

$T_C = \cancel{\bar{Q}_C \bar{Q}_B Q_A +}$

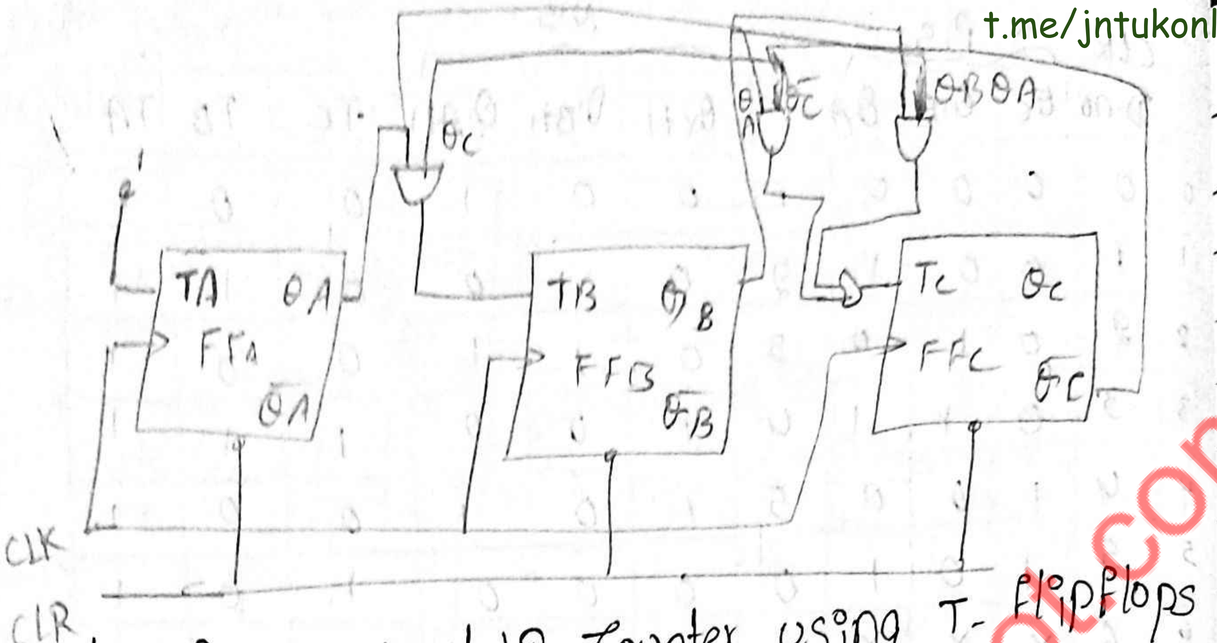
$\cancel{\bar{Q}_C Q_B Q_A}$

$P_1(\bar{Q}_C Q_A) + P_2($

$(\bar{Q}_C + Q_C)(Q_B Q_A)$

$\bar{Q}_C Q_A + Q_B Q_A$

$T_C = P_2 + P_1 \to (\bar{Q}_C Q_A + Q_B Q_A)$



Date 16/9/19 Design Mod-12 Counter using T-flipflops

The total no of counts $N = 12$

Counting sequence - 0 to $N-1$

0 to $(12-1)$

0 to 11 (12, 13, 14, 15)

don't cares

No.of flip flops - n $\quad 2^n \geq N$

n	$2^n \geq N$
1	$2^1 \ngeq 12$
2	$2^2 \ngeq 12$
3	$2^3 \ngeq 12$
4	$2^4 \geq 12$

T flip flop

P.S Q_n	N.S Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

No.of flip flops = No.of binary bits

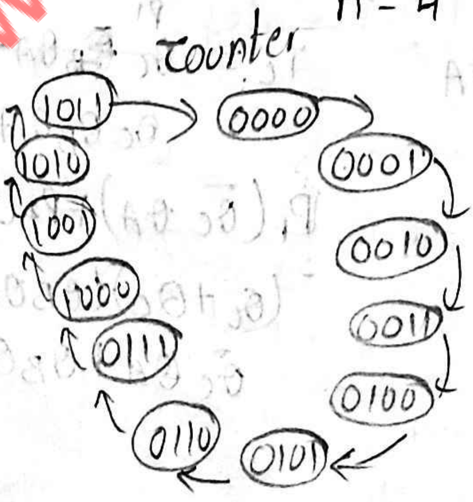
$n = 4$; $0 - (2^n - 1)$

$0 - (2^4 - 1)$

$0 - (16 - 1)$

$0 - 15$

Counter



P.S	N.S
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	0

D.NO	P.S. Q_D (8)	Q_C (4)	Q_B (2)	Q_A (1)		N.S. Q_{D+1} (8)	Q_{C+1} (4)	Q_{B+1} (2)	Q_{A+1} (1)	T_D	T_C	T_B	T_A
										T flipflop			
0	0	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	1	2	0	0	1	0	0	0	1	1
2	0	0	1	0	3	0	0	1	1	0	0	0	1
3	0	0	1	1	4	0	1	0	0	0	1	1	1
4	0	1	0	0	5	0	1	0	1	0	0	0	1
5	0	1	0	1	6	0	1	1	0	0	0	1	1
6	0	1	1	0	7	0	1	1	1	0	0	0	1
7	0	1	1	1	8	1	0	0	0	1	1	1	1
8	1	0	0	0	9	1	0	0	1	0	0	0	1
9	1	0	0	1	10	1	0	1	0	0	0	1	1
10	1	0	1	0	11	1	0	1	1	0	0	0	1
11	1	0	1	1	0	0	0	0	0	1	0	1	1

$T_A = \sum_m (0,1,2, \ldots 11)$

$T_B = \sum_m (1,3,5,7,9,11)$

$T_C = \sum_m (3,7)$; $T_D = \sum_m (7,11)$

T_A

$Q_DQ_C \backslash Q_BQ_A$	00	01	11	10
00	1_0	1_1	1_3	1_2
01	1_4	1_5	1_7	1_6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_8	1_9	1_{11}	1_{10}

octent (1)

$T_A = 1$

T_B

$Q_DQ_C \backslash Q_BQ_A$	$\bar{Q}_B\bar{Q}_A$ 00	$\bar{Q}_BQ_A$ 01	Q_BQ_A 11	$Q_B\bar{Q}_A$ 10
$\bar{Q}_D\bar{Q}_C$ 00	0	1_1	1_3	2
$\bar{Q}_DQ_C$ 01	4	1_5	1_7	6
Q_DQ_C 11	X_{12}	X_{13}	X_{15}	X_{14}
$Q_D\bar{Q}_C$ 10	8	1_9	1_{11}	10

octent $T_B = Q_A$

T_C

$Q_DQ_C \backslash Q_BQ_A$	00	01	11	10
00	0	1	1_3	2
01	4	5	1_7	6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	8	9	11	10

Pair(3,7) ; $T_C = \bar{Q}_D Q_B Q_A$

T_D

$Q_DQ_C \backslash Q_BQ_A$	00	01	11	10
00	0	1	3	2
01	4	5	1_7	6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	8	9	1_{11}	10

pair(7,5) ; pair(15,11)

$T_D = Q_C Q_B Q_A + Q_D Q_B Q_A$

$T_A = 1$; $T_B = Q_A$; $T_C = \overline{Q}_D Q_B Q_A$; $T_D = Q_C Q_B Q_A +$
$Q_D Q_B Q_A$