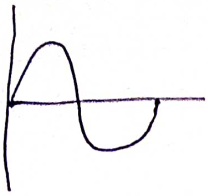
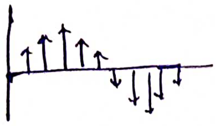
25/08/20

Digital Logic Design (86-8M)

Analog:



Digital



Advantages of Digital Compared to Analog

* Effect of noise is less.
* Digital is more efficient for data transmission.
* when data storage is required then digital is best.
* In digital data compression is possible.
* In diagital data can be encrypted. Hence data security is more.

Syllabus :

1. Binary System
2. Boolean Algebra
3. Logic gates
4. k-maps
5. Combinational Logic Circuits
6. Sequential logic circuits

Ref books:

1. Moris Mano (theory)
2. Anand kumar (problems)

1. Binary System

(i) Number System Conversions

(ii) Binary Arithmetic

(iii) Complements

(iv) Binary Codes

1. Binary System

Number system:

* Number systems are classified based on radix (r) or base.
  ↓
  i.e., no of symbols used

(i) Decimal Number System ($r=10$) $\Rightarrow$ 0, 1, 2, ... 9

(ii) Binary Number system ($r=2$) $\Rightarrow$ 0, 1

(iii) Octal Number system ($r=8$) $\Rightarrow$ 0, 1, 2 ... , 7

(iv) Hexadecimal Number System ($r=16$) $\Rightarrow$ 0, 1, 2, ..., 9, A, B, C, D, E, F

Note:

* Minimum digit of any number system is 0
* Maximum digit of any number system is $r-1$

Number System Conversion: for Integers

Decimal to Binary:

Eg: Convert given decimal number to binary

$[46]_{10} = [?]_2$

$$\begin{array}{r|l} 2 & 46 \\ \hline 2 & 23-0 \\ \hline 2 & 11-1 \\ \hline 2 & 5-1 \\ \hline 2 & 2-1 \\ \hline & 1-0 \end{array}$$

$= (101110)_2$

Decimal to Octal

Eg: $[567]_{10} = [?]_8$

$$\begin{array}{r|l} 8 & 567 \\ \hline 8 & 70 - 7 \\ \hline 8 & 8 - 6 \\ \hline & 1 - 0 \end{array}$$

$\therefore (567)_{10} = (1067)_8.$

Decimal to Hexadecimal:

Eg: $[2766]_{10} = [?]$

$$\begin{array}{r|l} 16 & 2766 \\ \hline 16 & 172 - E \\ \hline & A - C \end{array}$$

$\therefore (2766)_{10} = \cancel{(ACE)_{16}}\ (ACE)_{16}$

Decimal to Any radix:

Eg: $(531)_{10} = (?)_7$

$$\begin{array}{r|l} 7 & 531 \\ \hline 7 & 75 - 6 \\ \hline 7 & 10 - 5 \\ \hline & 1 - 3 \end{array}$$

$\therefore (531)_{10} = (1365)_7$

Number System Conversion for fractions:

Decimal to binary

Eg: $[0.25]_{10} = (?)_2$

0.25×2	$= 0.5$	0
0.5×2	$= 1.0$	1
0.0×2	$= 0$	0 — Stop with 0

i.e., $(0.25) = (0.01)_2$

Eg : $(0.95)_{10} = (?)_2$

$0.95 \times 2 = 1.9$ 1
$0.9 \times 2 = 1.8$ 1
$0.8 \times 2 = 1.6$ 1
$0.6 \times 2 = 1.2$ 1
$0.2 \times 2 = 0.4$ 0
$0.4 \times 2 = 0.8$ 0
$0.8 \times 2 = 1.6$ 1

$\therefore (0.95)_{10} = (0.11\underline{1100}\,\underline{1100}\,\underline{1100}\ldots)_2$

Mixed

Eg: $(95.75)_{10} = (?)_2$

$95.75 = 95 + 0.75$

$2 \mid 95$
$2 \mid 47 - 1$
$2 \mid 23 - 1$
$2 \mid 11 - 1$
$2 \mid 5 - 1$
$2 \mid 2 - 1$
$1 - 0$

$(95)_{10} = (1011111)_2$

~~0.75 x~~

$0.75 \times 2 = 1.5$ 1
$0.5 \times 2 = 1.0$ 1
$0.0 \times 2 = 0$

$(0.75)_{10} = (0.11)_2$

$\therefore$ ~~95.07~~ $(95.75)_{10} = (1011111.011)_2$

Binary to Decimal:

$(101110)_2 = (?)_{10}$

$1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$= 32 + 8 + 4 + 2$

$= (46)_{10}$

- stop with 0

Eg: $(1011111.11)_2 = (?)_{10}$

$(2^6+2^5+2^4+2^3+2^2+2^1)+(2^{-1}+2^{-2})$

$\dots\ 2^2\ 2^1\ 2^1\ 2^0\ .\ 2^{-1}\ 2^{-2}\ 2^{-3}\ \dots$

$95+\left(\frac{1}{2}+\frac{1}{4}\right)$

$95+(0.75)$

$=(95.75)_{10}$

Binary to Octal:

$(10110011)_2 = (?)_8$

$\underline{010}\ \underline{110}\ \underline{011}$

$2\quad 6\quad 3$

$\therefore (263)_8$

Octal to binary:

$(513.241)_8 = (?)_2$

$101\quad 001\quad 011\quad 010\quad 100\quad 001$

$\therefore (101001011.010100001)_2$

Hexadecimal to Binary ~~Table~~: & Binary to Hexa Decimal:

Eg: $(101010.101001)_2 = (?)_{16}$

$\underline{0010}\ \underline{1010}.\ \underline{1010}\ \underline{0100}$

$\therefore (2A.A4)_{16}$

Eg. $(5C3D.1F2)_{16}$

$=(\underline{0101}\,\underline{1100}\,\underline{0011}\,\underline{1101}.\underline{0001}\,\underline{1111}\,\underline{0010})_2$

Q1) $(135)_x + (144)_x = (323)_x$

Find x

$(x^2+3x+5) + (x^2+4x+4) = (3x^2+2x+3)$

$\Rightarrow \quad x^2 - 5x - 6 = 0$

$\cancel{x^2 - 2x - 3x - 6 = 0}$

$\cancel{x(x-2)} \; (x-2)(x-3) = 0$

$x =$

$x^2 - 6x + x - 6 = 0$

$\Rightarrow (x+1)(x-6) = 0$

$x = 6$ (or) $x = -1$

↳ radix is ~~not~~ never negative

$\therefore x = 6$

26/08/20

Q2) If $(11X1Y)_8 = (12C9)_{16}$, find X, Y.

$\cancel{(001\ 001\ \_\_\ 001\ \_\_)_2 = (0001\ 0010\ 1100\ 1001)_2}$

$\cancel{X = 101 = 5}$

$(001\ 001\ \_\_\ 001\ \_\_)_2 = (0001\ 0010\ 1100\ 1001)_2$

i.e., 100

$\therefore X = 011 = 3$

$Y = 001 = 1$

Q3) Find no of 1's in the binary representation of below number.

$(4 \times 4096) + (9 \times 256) + (7 \times 16) + 5$

Sol:

$= (4 \times 4096) + (9 \times 256) + (7 \times 16) + (5 \times 1)$

$= (4 \times 16^3) + (9 \times 16^2) + (7 \times 16) + (5 \times 16^0)$

$= [4975]_{16}$

$= (\underline{0100}\ \ \underline{1001}\ \underline{0111}\ \underline{0101})_2$

no of 1's = 8

(Q4) find no of 1's in binary representation of

$3 \times 512 + 7 \times 64 + 5 \times 8 + 3$

Sol:

$= (3753)_8$

$= (011\ \ 111\ \ 101\ \ 011)_2$

$\therefore$ no of 1's = 9

(Q5)

find the base or radix such that below equation holds

$$\frac{312}{20} = 13.1$$

Sol:

let x be radix

Now converting into decimal

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

$$3x^2+x+2 = 2\cancel{x}\left(\frac{x^2+3x+1}{\cancel{x}}\right) \quad (\because x \text{ can't be } 0)$$

$$3x^2+x+2 = 2x^2+6x+2$$

$$x^2-5x=0$$

$$\underline{x=5}$$

Q6 $\sqrt{(224)_r} = (13)_r$ find r

Sol :

$\sqrt{2r^2+2r+4} = r+3$

S.O.B.S

$2r^2+2r+4 = r^2+6r+9$

$\Rightarrow r^2-4r-5=0$

$r^2-5r+r-5=0$

$r=5 \quad r=-1$

$\therefore r=5$

Q7 Consider $(123)_5 = (x8)_y$. Find no of possible solns for x & y

Sol :

$25+(2\times5)+3 = xy+8$

$\Rightarrow xy=30$

WKT $x<y$ & x,y must be ~~int~~ positive integers

Also $y>8$

$\therefore x=1 \quad y=30$

$x=2 \quad y=15$

$x=3 \quad y=10$

~~$x=5 \quad y=6$~~

~~$\therefore$ 4 solutions~~

$\therefore$ 3 solutions

Q8 $(43)_x = (y3)_8$. find no of possible solns

Sol :

$4x+3 = 8y+3$

$\Rightarrow x=2y$

$x>4$

$y<8$

$x=5 \Rightarrow y=2.5$ ✗

$x=6 \Rightarrow y=3$

$x=8 \Rightarrow y=4$

$x=10 \Rightarrow y=5$

$x=12 \Rightarrow y=6$

$x=14 \Rightarrow y=7$

$\therefore$ 5 solutions

) equation

(can't be 0)

Binary Arithmetic

Binary addition

Eg: ~~add~~ find $(1011)_2 + (1010)_2$

$$
\begin{array}{cccccc}
 & & & 1 & & \\
 & & 1 & 0 & 1 & 1 \\
\text{end carry} \leftarrow & 1 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 1 & 0 & 1 \\
\hline
\end{array}
$$

Eg: $(1111)_2 + (1110)_2$

$$
\begin{array}{ccccc}
 & & 1 & 1 & \\
 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & 1 \\
\hline
\end{array}
$$

$\therefore (11101)_2$

Binary Subtraction:

Eg: $(1011)_2 - (1010)_2$

$$
\begin{array}{cccc}
1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 \\
\hline
\end{array}
$$

Ans: $(0001)_2 = 1$

Eg: $(11011)_2 - (1100)_2$

$$
\begin{array}{ccccc}
 & & 10 & & \\
0 & & \cancel{0} & 10 & \\
\cancel{1} & \cancel{1} & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 \\
\hline
\end{array}
$$

~~Take borrow~~

the number taken as borrow is equal to radix - i.e., $(10)_2$ in binary

Eg: $(11001)_2 - (1110)_2$

$$
\begin{array}{ccccc}
 & 10 & 1 & & \\
0 & \cancel{0} & \cancel{1} & 10 & \\
\cancel{1} & \cancel{1} & \cancel{0} & \cancel{0} & 1 \\
0 & 1 & 1 & 1 & 0 \\
\hline
0 & 1 & 0 & 1 & 1 \\
\hline
\end{array}
$$

$\therefore (01011)_2$

Eg : $(1000001) - (111110)_2$

$$\begin{array}{r} 1000001 \\ 0111110 \\ \hline 0000011 \end{array}$$

Eg : $(1010)_2 - (1011)_2$

This is small - large

$$\therefore \begin{array}{r} 1011 \\ 1010 \\ \hline 0001 \end{array}$$

$\therefore$ Ans : $(-0001)_2$

Octal Arithematic

Octal addition :

Eg : $(373)_8 + (264)_8$

$$\begin{array}{r} 373 \\ 264 \\ \hline 657 \end{array}$$

~~7+6 = 13~~

$(7)_8 + (6)_8 = (13)_{10}$

$= (15)_8$

$\therefore (657)_8$

Eg :

$$\begin{array}{r} 457 \\ 564 \\ \hline 1243 \end{array}$$

$\therefore (457)_8 + (564)_8 = (1243)_8$

Eg. $(345)_8 + (467)_8 = ?$

$$\begin{array}{r} 345 \\ 467 \\ \hline 1034 \end{array}$$

$\therefore (1034)_8$

...ow

...mber taken as

...o is equal to

...$(10)_2$ in binary

(Q.9) $(267)_8 + (YX6)_8 = (ZYX)_8$

Find X, Y, Z

sol:

1	1	
2	6	7
$Y_{(4)}$	$X_{(5)}$	6
7	4	5
Z	Y	X

$\therefore X=5,\ Y=4,\ Z=7$

Octal Subtraction:

Eg: $(534)_8 - (365)_8$

4	12 / 2	14
5	3	4
3	6	5
1	4	7

$\therefore (534)_8 - (365)_8 = (147)_8$

(Q10) which of the following represents $(E3)_{16}$

a) $(CE)_{16} + (A2)_{16}$

b) $(1BC)_{16} - (DE)_{16}$

c) $(2BC)_{16} - (1DE)_{16}$

d) $(200)_{16} - (11D)_{16}$

sol:

a)

C	E
A	2
	C ✗

b)

	A	1C
1	B	C
0	D	E
		E ✗

c)

	A	1C
2	B	C
1	D	E
		E ✗

d)

+10 / 1	-1	10
2	0	0
1	1	D
0	E	3

$\therefore$ opt (d)

Complements

* complements are generally applied only on negative numbers.

Types of complements:

i) Diminished complement (or) 'r-1' complement

formula: $r^n - r^{-m} - N$

ii) Radix Complement (or) r's complement

formula: $r^n - N$. for $N \neq 0$

and 0 for $N = 0$

N is given number

r is radix

n is no. of digits in integral part

m is no of digits in fractional part

Eg:

Complements in decimal system:

* diminished complement is 9's complement
* radix complement is 10's complement

27/08/20

9's Complement:

Eg: Find 9's complement of $N = (632.54)_{10}$

r-1 complement $= r^n - r^{-m} - N$

Here $r = 10$, $n = 3$, $m = 2$, $N = 632.54$

9's complement of $10^3 - 10^{-2} - 632.54$

$= (367.45)_{10}$

$\begin{array}{ccc} +10 & & \\ -1 & & 10 \\ \cancel{2} & \cancel{0} & 0 \\ 1 & 1 & D \\ \hline & E & 3 \end{array}$

Short cut to find 9's complement:

* Subtract each digit from 9

$N = 632.54$

9's complement $= (367.45)_{10}$

> **Note:**
> * To find any r-1 complement, subtract each digit of a given number from r-1 (including rightmost zeroes)

10's complement:

Eg: Find 10's complement for $N = [632.54]_{10}$

~~for~~ r's complement $= r^n - N$

i.e., 10's complement $= 10^3 - 632.54$

$= (367.46)_{10}$

Shortcut to find 10's complement:

~~r's complement~~ =

10's complement = 9's complemt + 1
↳ (add to LSB digit)

$N = 632.54$

9's $\longrightarrow$ 367.45
$\quad +1$

10's $\longrightarrow$ 367.46

Shortcut 2 :

→ Subtract the 1st LSB non-zero digit from 10 and subtract remaining from 9. Also keep ~~make~~ remaining LSB '0' digits as '0'.

Eg:

N = 632 · 54

10's ⟶ $(367 \cdot 46)_{10}$

Eg: Find 10's complement of $(736 \cdot 520)_{10}$

N = 7 3 6 · 5 2 0

10's ⟶ 2 6 3 · 4 8 0

Note :

* To find any r's complement, subtract the rightmost non-zero digit from r and subtract remaining digits from r-1. Also keep all the rightmost 0's (if exist) as '0' only.

**Eg: Find 9's & 10's complement of $(3024 \cdot 10500)_{10}$

Sol :

9's complement ⟶ $(6975 \cdot 89499)_{10}$

10's complement ⟶ $(6975 \cdot 89500)_{10}$

Binary Complements:

1's complement:

Eg: find 1's complement of $(10101)_2$

given
10101

1's ⟶ 01010

2's Complement

Eg: find 2's complement of $(10101)_2$

sol:

1 0 1 0 1

2's → $(01011)_2$

Eg: find 2's complement of $N = (1010.10100)_2$

1010.10100

2's → 0101.01100

Eg: find 2's complement of $N = (1011.010\,1000)_2$

2's → $(0100.1011000)_2$

Octal Complements:

7's complement:

Eg: find 7's complement of $(304.520)_8$

7's ⇒ $(473.257)_8$

8's complement:

Eg: $N = (304.507000)_8$

8's → $(473.271000)_8$

* slly for 15's complement & 16's complement and any other complement, we follow the same procedure.

Signed Numbers

+ve number ⟹ MSB = 0

-ve number ⟹ MSB = 1

* Signed numbers can be represented in 3 ways:

  (i) Signed magnitude method.

  (ii) Signed 1's complement method.

  (iii) Signed 2's complement method.

(i) **Signed Magnitude method:**

Eg: Represent +5, -5 in signed mag. method

5 → 101 (unsigned representation)

$$\left.\begin{aligned} +5 &\rightarrow 0101 \\ -5 &\rightarrow 1101 \end{aligned}\right\}\text{ signed representation}$$

Range: $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

(ii) **Signed 1's complement method:**

Eg: Represent +5, -5 using 1's complement method

5 → 101 (unsigned representation)

$$\left.\begin{aligned} +5 &\rightarrow 0101 \\ -5 &\rightarrow 1010 \end{aligned}\right\}\text{ signed Representation}$$

Range: $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

* Here -ve numbers are 1's complements of their corresponding positive numbers.

(iii) Signed 2's complement method :

Eg: Represent +5, -5 in 2's complement

5 = 101

+5 = 0101

-5 = 1011

Range : (-2^{n-1}) to $(2^{n-1}-1)$

* Here -ve numbers are represented as 2's complements of their corresponding +ve numbers.

Eg: what is decimal equivalent of 1011 in 2's complement.

Sol :

given 1 0 1 1

Here MSB is 1

∴ it is -ve number

So this number is in complement format

→ finding 2's complement again will give the positive equivalent number

1011

$\xrightarrow{2's}$ 0101 → 5

∴ -5

* 2's complement representation allows sign extension.

i.e., $(1011)_2 = (11011)_2 = (111011)_2$

100

(Q11) To requ

(Q12) Minim in

(Q13) Let A complem

Sol :

A =

(Q11) To represent -17, the minimum & maximum no of bits required in 2's complement form is ______

Sol:

$17 \rightarrow 10001$

$+17 \rightarrow 010001$

$-17 \rightarrow 101111$

$\therefore$ min no of bits = 6

Also ....$1111\underline{101111}$

$\therefore$ max no of bits = ∞

(Q12) Minimum no of bits req to represent $(+32)_{10}$, $-(32)_{10}$ in 2's complement form is

Sol:

$32 \rightarrow 100000$

$+32 \rightarrow 0100000$

$-32 \rightarrow \underline{1100000}$

$\hookrightarrow$ However leftmost '1' is not needed (i.e., sign extension)

$\therefore (1100000)_2 = (100000)_2 = -(32)_{10}$

(Q13) Let A = 11111010, B = 00001010 be two 8 bit 2's complement numbers. Their product in 2's complement form is.

Sol:

$A = (11111010)_2 = (1010)_2$

$2's \rightarrow 0110$

$\therefore A = -6$

$B = (00001010)_2 = (01010)_2 \Rightarrow B = 10$

$A \times B = -60$

$+60 \rightarrow 0111100$

$-60 \rightarrow 1000100$

In 8 bits $\rightarrow$ 11000100 ($\because$ sign extension)

28/08/20

(Q14) The range of singed decimal numbers that can be represented by 6 bit 1's complement is ______

Sol :

$n = 6$

range : $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

$\Rightarrow -(2^5-1)$ to (2^5-1)

$\Rightarrow -31$ to 31

Note :

→ Both 1's & 2's Complement allow sign extension but sign magnitude method doesn't.

→ Signed magnitude & 1's complement has 2 representations for zero. But 2's complement has a unique representation.

(Q15) Find 9's complement of $(36800)_9$

Sol :

8's complement

8 8 9
36800
52100

$\therefore 52100$

Note :

10's complement of $(000.00)_{10} = (000.00)_{10}$

Binary Codes:

~~to~~ 2 types

- weighed codes
  - 8421
  - 3321
- non-weighted codes
  - Ex-3 code
  - grey code

weighted Codes:

BCD code (or) 8421 code:

* BCD:- Binary Coded Decimal

Decimal	BCD 8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

→ BCD Code is a 4-bit code.

→ BCD code is sequential code (i.e., ~~d~~ add in '1' to ~~de~~ binary form of some decimal numbers gives the next number to it.)

→ The are 6 invalid BCD codes. They are

{1010, 1011, 1100, 1101, 1110, 1111}

→ BCD & Binary are not equivalent.

→ BCD is not a self complement code.

Eg: Represent $(93)_{10}$ in binary & BCD.

Sol:

Binary:

$(1011101)_2$

BCD:

$(93)_{10}$

$(1001\ 0011)_{BCD}$

3321 Code

Decimal	3	3	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

logical complements to each other

→ For a decimal ~~number~~ digit, multiple ways may be possible to represent it under 3321. However we follow a unique a way for the purpose of making it easier.

We represent it such the

Complement (4) = 5

Complement (3) = 6

⋮

Complement (0) = 9

→ Hence we call this code as self complement code.

→ The code in which the codes of a digit and it's 9's complement are logical complements to each other is known as self complement code.

Note: (short cut)

To determine whether any given weighted code is self complement code or not,

if sum of all weights = 9, that code is self complement code (This is unidirectional)

→ 3321 is not a sequential code.

Non-weighted Codes:

Excess-3 Code:

* Here we add 0011 to the BCD (not to binary)

Decimal	Ex-3 Code
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

BCD	X's 3 Code
→ 4-bit weighted code	→ 4 bit non-weight code
→ Sequential code	→ Sequential code
→ not a self-complement code	→ Self-complement code
→ 6 invalid BCD codes {1010, 1011, 1100, 1101, 1110, 1111}	→ 6 invalid codes {0000, 0001, 0010, 1101, 1110, 1111}

Gray code

→ gray code depends on modulo sum operator ($\oplus$)

Binary to Gray Code conversion

Eg: Convert given binary number $(1011)_2$ into gray code

	b_3	b_2	b_1	b_0
Binary :	1	0	1	1
	MSB is same ↓	$\oplus$	$\oplus$	$\oplus$
gray code :	1	1	1	0
	g_3	g_2	g_1	g_0

$\therefore$ gray code of $(1011)_2$ is $(1110)_{gray}$

~~If $b_n b_{n-1} b_{n-2} \cdots b_2 b_1 b_0$~~

If $b_{n-1} b_{n-2} b_{n-3} \cdots b_2 b_1 b_0$ is a binary number and let $g_{n-1} g_{n-2} g_{n-3} \cdots g_2 g_1 g_0$ be it gray code

then

~~$g_{n-1} = g_{n-1}$~~

$$\boxed{\begin{aligned} g_{n-1} &= b_{n-1} \\ g_i &= \cancel{g}\, b_{i+1} \oplus b_i, \quad i \neq n-1 \end{aligned}}$$

Gray code to binary:

Eg: Convert given gray code (1110) to binary form

gray code : 1 1 1 0

binary form : 1 0 1 1

If $g_{n-1} g_{n-2} \cdots g_1 g_0$ is a ~~gray~~ gray code and $b_{n-1} b_{n-2} \cdots b_1 b_0$ be its binary form then

$$\boxed{\begin{aligned} b_{n-1} &= g_{n-1} \\ b_i &= b_{i+1} \oplus g_i \quad , i \neq n-1 \end{aligned}}$$

Eg: Express decimal numbers 0 to 7 in gray code using 3 bits

Decimal	binary	gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

0
1

00
01
11
10

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

Note :

From above example,

→ In gray code any two consecutive number differ by exactly one bit. Hence gray code is called unit distance code.

→ It is also called as refletive code (∵ practically it generated using mirror image)

→ It is also called as cyclic code. (∵ 3-bit gray code is generate from 2-bit graycode, 4-bit gray code is generated from 3-bit graycode.......)

ASCII Code

* 7-bit code

0 to 9	48 to 57
A to Z	65 to 90
a to z	97 to 122

Decimal	ASCII	Binary equivalent
0	48	0110000
1	49	0110001
2	50	0110010
3	51	0110011
4	52	0110100
5	53	0110101
6	54	0110110
7	55	0110111
8	56	0111000
9	57	0111001

Boolean Algebra

Laws:

* Commutative law:

$$A + B = B + A$$
$$AB = BA$$

* Associative law:

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

* Distributive law

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

Rules on Postulates of boolean algebra:

→ $A + 0 = A$

→ $A + 1 = 1$

→ $A \cdot 0 = 0$

→ $A \cdot 1 = A$

→ $A + A = A$

→ $A \cdot A = A$

→ $A + \bar{A} = 1$

→ $A \cdot \bar{A} = 0$

→ $\overline{(\bar{A})} = A$

→ $A + AB = A$ $\quad (A + AB = A(1+B) = A)$

→ $(A+B)\cdot(A+C) = A + BC$

→ $A + \bar{A}B = A + B$

Simplify the following boolean functions

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

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

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

Variables: A, B, C....

Complements: $\bar{A}$, $\bar{B}$, $\bar{C}$...

Literals: A, $\bar{A}$, B, $\bar{B}$, ...

→ we need to minimize until we get ~~min~~ min no. of literals

for example

$\bar{A}(B+\bar{C})$ has 3 literals

Eg: $(b\bar{c}+\bar{a}d)(a\bar{b}+c\bar{d})$

Sol:

$b\bar{c}a\bar{b} + b\bar{c}c\bar{d} + \bar{a}da\bar{b} + \bar{a}dc\bar{d}$

$0 + 0 + 0 + 0$

$= \underline{0}$

Eg: Simplify

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

$= (0 + A\bar{B} + AD + \bar{A}\bar{B} + \bar{B} + \bar{B}D)(\bar{B}+C+\bar{D})$

$= \underline{A\bar{B} + A\bar{B}C} + \underline{A\bar{B}\bar{D} + A\bar{B}D} + ACD + 0 + \underline{\bar{A}\bar{B} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{D}}$
$+ \bar{B} + \bar{B}C + \bar{B}\bar{D} + \bar{B}D + \bar{B}CD + 0$

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

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

$= \bar{B} + ACD$

$\therefore$ 4 literals

21/05/20

Duality Theorem:

* $0 \leftrightarrow 1$, $\cdot \leftrightarrow +$

Eg: apply duality to the following function

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

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

Note: while applying duality, variables remain same.

Eg: Apply duality for $f = A+1$

$$f_D = A\cdot 0$$

Eg: $\bar{B}+1 = 1$

$$\Rightarrow \bar{B}\cdot 0 = 0$$

Eg: Find duality then simplify

$$f = AB+BC+CA$$

$$\Rightarrow f_D = (A+B)(B+C)(C+A)$$

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

$$f_D = BC+BA+AC+AC$$

$$f_D = AB+BC+CA$$

Here $f_D = f$

Such function is called self dual function

A function is self dual
$\Rightarrow$ exactly one element of every complementary pair is present

If $i \in$ "m" then
$2^n-1-i \in$ "M"
Such function is called self dual

Self Dual function:

A function f is called as self dual function, if $f_D == f$

Q14) By using 2-variables, no of ~~bo~~ possible boolean functions is ____

Sol :

A	B	F	
0	0	0/1	→ 2 ways
0	1	0/1	→ 2 ways
1	0	0/1	→ 2 ways
1	1	0/1	→ 2 ways

i.e., 2^4 ways = 16

Q15) By using 2-variables, no of possible self dual functions is ____

Sol :

$f = A$, $f = \bar{A}$, $f = B$, $f = \bar{B}$

are self dual functions

$\therefore 4$

~~Q16) By using 3-variables, no of possible self dual functions is ____~~

~~Sol~~:

Note :

→ Using n variables no of function possible $= 2^{(2^n)}$

→ Using n variables, no of self dual functions possible $= 2^{(2^{n-1})}$

Demorgan's theorem :

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

Slly $\overline{A+B+C+\dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \dots\dots$

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

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

...ions is

Eg: Simplify

$$f = \overline{\overline{\overline{AB}\, C} \cdot D}$$

Sol:

given $\overline{(\overline{(\overline{AB})\, C})\, D}$

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

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

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

31/08/20

ctions is

Q16. Simplify

$$f = \underbrace{B + AD + BC}_{} + \overline{B + A(C+D)}$$

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

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

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

actions is

$f = \underbrace{B + \bar{A}\bar{B}} + AD + \bar{B}\bar{C}\bar{D}$

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

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

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

$= 2^{(2^n - 1)}$

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

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

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

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

Method 2:

$f = B + AD + BC + \overline{B + A(C+D)}$

$= (B + AD) + \overline{(B + AD) + AC}$

$= (B + AD) + \overline{(B + AD)} \cdot \overline{AC}$

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

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

Complement of a Boolean function :

Eg: find complement of the given boolean function

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

Sol :

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

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

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

Eg: The simplied form of the following

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

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

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

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

$$f = C + \bar{C}\bar{A} + \bar{A}D + BD$$

$$f = C + \bar{A} + \bar{A}D + BD$$

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

Q17 Simplyfy $f = (A\bar{B}\bar{C} + C\bar{D}) + [(\bar{A}+B+C)(\bar{C}+D)]$

Sol :

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

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

now $f = x + \bar{x} = 1$

Boolean function forms

Simplified forms — SOP, POS

Standard forms — SSOP, SPOS

SOP (Sum of Products):

Eg: $f_{sop} = \underbrace{\bar{B}\bar{C} + AB}_{2\ terms}$

b/w the terms we have operator '+' (sum)

b/w the literal we have operator '.' (product)

SSOP: (Standard Sum of Product)

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

ssop is sop in which every term must have all variables' literals

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

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

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

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

→ Each term in ssop is called as minterm

It is indicated with "m".

$\cancel{f_{ssop} = A}$

Eg: let ~~for~~ $f_{ssop} = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$

i.e., $\underbrace{000}_{m_0} + \underbrace{100}_{m_4} + \underbrace{110}_{m_6} + \underbrace{111}_{m_7}$

minterms $\rightarrow$ $m_0 \quad m_4 \quad m_6 \quad m_7$

$\therefore f_{ssop} = \sum m(0,4,6,7)$

POS: (Product of Sum)

Eg:

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

terms are separated by product

within a term, literals are seperated by sum

SPOS: (Standard Product of sum)

~~Sol~~

Eg: Convert $f_{pos} = (x+y)\cdot(\bar{y}+z)$ into SPOS

~~$f_{pos} = x+$~~

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

$$= (x+y+0)(0+\bar{y}+z)$$

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

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

In SPOS each term is called maxterm.

Maxterm is indicated using "M"

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

$\underbrace{0\,0\,0}_{M_0} \quad \underbrace{0\,0\,1}_{M_1} \quad \underbrace{0\,1\,0}_{M_2} \quad \underbrace{1\,1\,0}_{M_6}$

Maxterms $\rightarrow$ $M_0 \quad M_1 \quad M_2 \quad M_6$

$\therefore f_{spos} = \prod M(0,1,2,6)$

Conversions b/w canonical form

Eg: Convert $f(A,B,C) = \Sigma m(1,3,6)$ into SPOS

↓ absent terms (0 to 7)

$f_{SPOS} = \Pi M(0,2,4,5,7)$

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

Q18 Let * be defined as

$x*y = \bar{x}+y$.. let $z = x*y$

value of $z*x$ is

Sol:

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

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

$= x$

Q19 Simplified sop form of $(P+\bar{Q}+\bar{R})(P+\bar{Q}+R)(P+Q+\bar{R})$

sol:

given $f = \Pi M(3,2,1)$

$= \Sigma m(0,4,5,6,7)$

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

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

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

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

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

Q20 If $x=1$ in the logic equation

$$\left[x+z\{\bar{y}+(\bar{z}+x\bar{y})\}\right]\{\bar{x}+\bar{z}(x+y)\}=1$$

then

a) $y=z$ b) $y=\bar{z}$ c) $z=1$ d) $z=0$

Sol:

$x=1 \Rightarrow$ $\cancel{\left[z\{\bar{y}+(\bar{z}+\bar{y})\}\right].(}$

$$(1)\{0+\bar{z}(1+y)\}=1$$

$\cancel{\bar{z}y=1}$ $\bar{z}=1$

$\cancel{\Rightarrow z=0 \ \& \ y=1}$ $\Rightarrow z=0$

Q21 Simplify $(x+y)(x+\bar{y})+\overline{(x\bar{y})+\bar{x}}$

Sol:

$$(x+y\bar{y})+\overline{x\bar{y}}\cdot x$$

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

$$x+x\bar{x}+xy= \quad x+xy=x$$

Q22 The boolean expression $xy+(\bar{x}+\bar{y})z$ is equivalent to

a) $xy\bar{z}+\bar{x}\bar{y}z$ b) $\bar{x}\bar{y}\bar{z}+xyz$

c) $(x+z)(y+z)$ d) none

Sol:

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

let $xy=p \Rightarrow \bar{p}=\bar{x}+\bar{y}$

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

$= xy+z$

01/09/20

Logic Gates:

Basic gates

(i) NOT (ii) AND (iii) OR

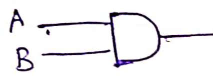
(iv) NAND (v) NOR

(vi) Ex-OR (vii) Ex-NOR

(i) NOT Gate (inverter):

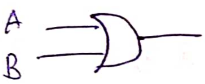
$\bar{A}$

(ii) AND Gate:



$A \cdot B$

(iii) OR Gate:



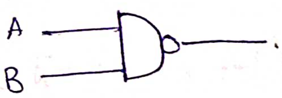
$A + B$

Note:

→ AND, OR, NOT are called basic gates (or) AOI gates

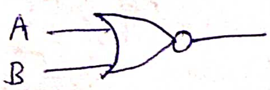
A — AND, O — OR, I — INVERTOR

(iv) NAND Gate:



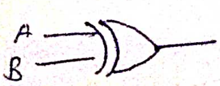
$\overline{AB} = \bar{A} + \bar{B}$

(v) NOR Gate:



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

(vi) Ex-OR: (Exclusive OR) (XOR)



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

7) Ex-NOR (XNOR) :

A, B

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

Properties of XOR logic gate :

→ $A \oplus B = A\bar{B} + \bar{A}B$

→ $A \oplus 0 = A$

→ $A \oplus 1 = \bar{A}$

→ $A \oplus A = 0$

→ $A \oplus \bar{A} = 1$

Consider

~~A⊕OP~~

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

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

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

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

i.e., $\boxed{\overline{A \oplus B} = A \odot B}$

Now consider

$$A \oplus B \oplus C = [A \oplus B] \oplus C$$

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

$$= x \cdot \bar{C} + \bar{x} C \qquad (\text{let } x = A\bar{B} + \bar{A}B)$$

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

$$\cancel{= x\bar{C} +}$$

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

$A \oplus B \oplus C$

$= (A \odot B)C$ +

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

$$\begin{array}{ccccccc} = A\bar{B}\bar{C} & + & \bar{A}B\bar{C} & + & \bar{A}\bar{B}C & + & ABC \\ 100 & & 010 & & 001 & & 111 \\ m_4 & & m_2 & & m_1 & & m_7 \end{array}$$

$\therefore A \oplus B \oplus C = \sum m(1,2,4,7)$

Now consider

$$A \odot B \odot C = [A \odot B] \odot C$$

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

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

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

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

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

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

$$\boxed{A \odot B \odot C = A \oplus B \oplus C}$$

Note:

→ $\overline{A \oplus B} = A \odot B$

→ $A \oplus B \oplus C = A \odot B \odot C$

→ $\overline{A \oplus B \oplus C \oplus D} = A \odot B \odot C \odot D$

$\vdots$

$$\boxed{\therefore x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_n = \begin{cases} \overline{x_1 \odot x_2 \odot x_3 \cdots \odot x_n}, & \text{if } n \text{ is even} \\ x_1 \odot x_2 \odot x_3 \odot \cdots \odot x_n, & \text{if } n \text{ is odd} \end{cases}}$$

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

Universal logic gates

→ NAND, NOR are called universal logic gates, cuz using either of these we can design any boolean function.

All logic gates using NAND, NOR :

i) NOT :

using NAND

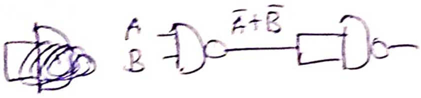
using NOR

ii) AND :

NAND :

$\Rightarrow AB = \overline{\overline{AB}}$

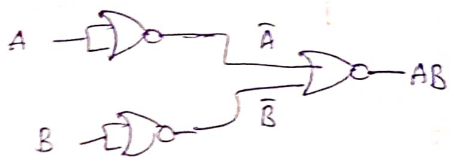
$= \overline{(\overline{A}+\overline{B})}$



NOR

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

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

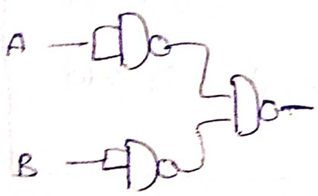


iii) OR :

NAND

Req: $A+B$

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

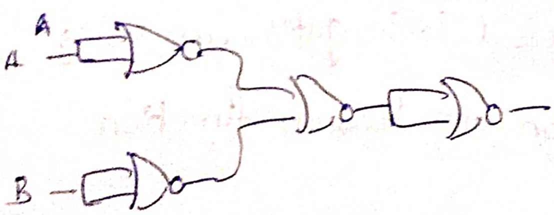


NOR

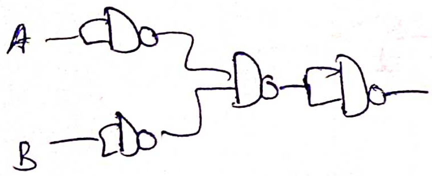
Req : $A+B$

avail : $\overline{A+B}$.

iv) NAND with NOR :

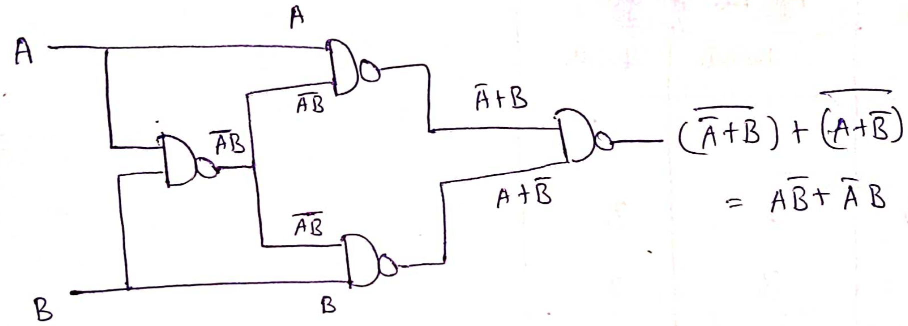


(v) NOR with NAND :

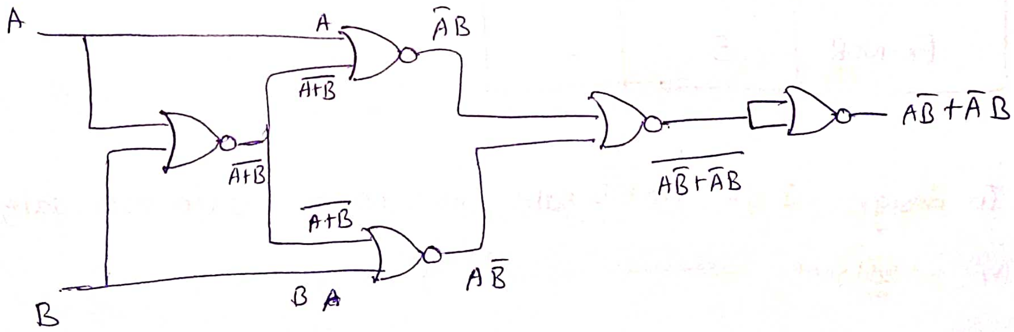


(vi) EX-OR

using NAND

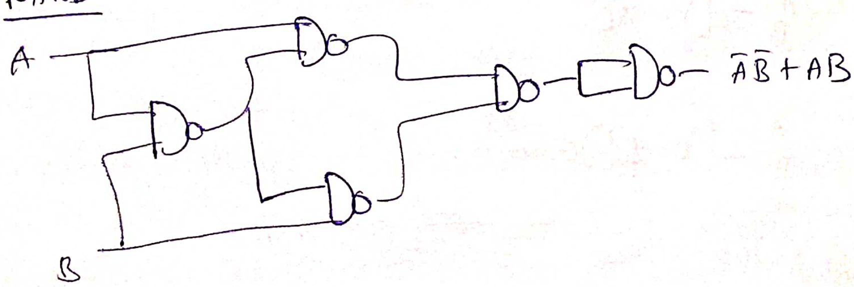


using NOR :

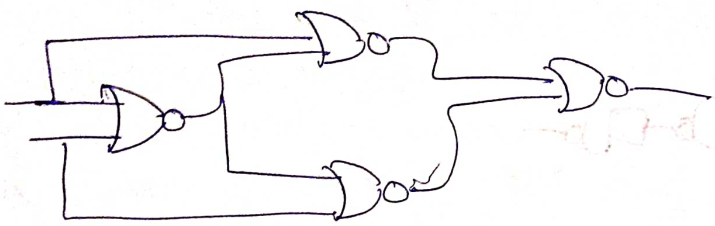


(vii) Ex-NOR

NAND



NOR :



The below table show min no of NAND & NOR gates req for implementing various gates

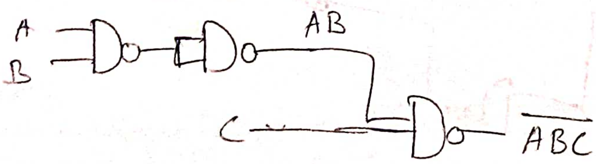
gate	min No of gates	
	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
NAND	1	4
NOR	4	1
Ex-OR	4	5
Ex-NOR	5	4

Q23 To design 3 i/p NAND gate, min no of 2 i/p nand gates required?

sol:

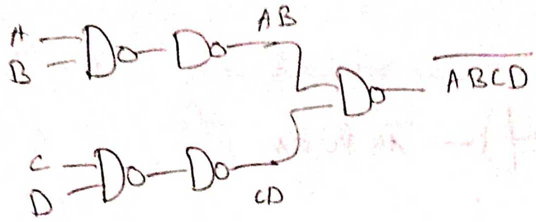
req: $\overline{ABC} = \bar{A}+\bar{B}+\bar{C}$

$= \overline{AB}+\bar{C}$



$\therefore 3$

(24) To design 4 i/p . min no of 2 ip NAND gate



$\therefore$ 5 gates

Note:

To design "n" i/p NAND gate min no of 2 i/p nand gates required is $2n-3$

Note:

To design "n" i/p NOR gate min no of 2 i/p NOR gates required is $2n-3$

Design any logic functions using universal gates

NAND:

→ To design with NAND the given function should be in SOP form.

Procedure:

Eg: Min no of NAND gates req to implement

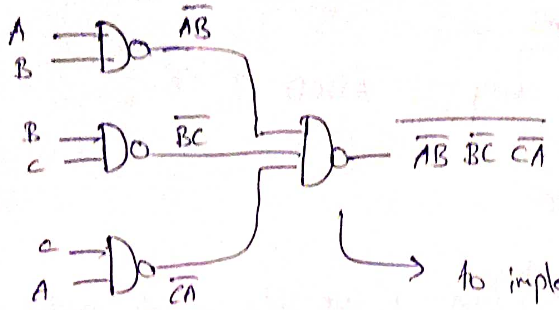
$$f = AB + BC + CA$$

$$f = \overline{\overline{f}} = \overline{\overline{AB + BC + CA}}$$

~~$f = \overline{(A+B)}$~~

$$f = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{CA}}$$

$f = \overline{\overline{AB}\;\overline{BC}\;\overline{CA}}$



to implement this 3 i/p nand gate we need 3 2 i/p nand gates.

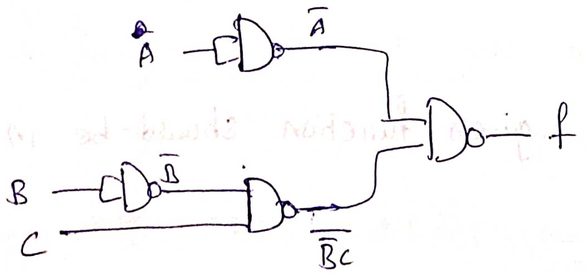
∴ Min no of NAND gates req to implement given function is 6 NAND gates.

Eg: How many min no of 2 i/p NAND gates are req to design to implement $f = A + \bar{B}C$

Sol:

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

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



NOR:

To design with NOR, the given function should be in POS form.

Eg: Find min no of NOR gates req to design it to

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

Sol:

NOR :

→ To design with NOR gate, the function must be in POS form

~~$f = A+BC$~~

Eg: Find min no of NOR gates req for $f = A+BC$

Sol:

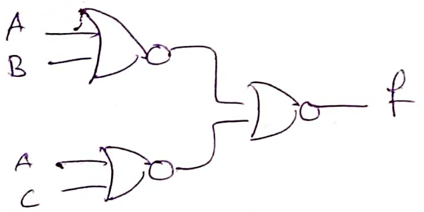
$f = A+BC$

this ~~SOP~~ is SOP

$\therefore f = (A+B)(A+C)$

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

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



$\therefore$ 3 2 i/p NOR gates

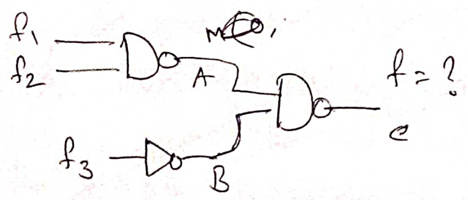
(Q25) The boolean functions are defined as

$f_1(A,B,C) = \Sigma m(0,1,3,5,6)$

$f_2(A,B,C) = \Sigma m(4,6,7)$

$f_3(A,B,C) = \Sigma m(1,4,5,7)$

to the following circuit.



Sol:

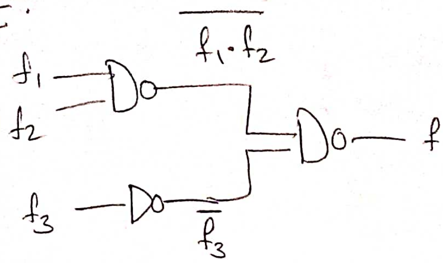
$A = \Sigma m(0,1,2,3,4,5,7)$

$B = \Sigma m(0,2,3,6)$

$\Rightarrow f = \Sigma m(1,4,5,6,7)$

e in POS form 128

Method 2 :



$$f = \overline{\overline{f_1 f_2} \cdot \overline{f_3}} = \overline{\overline{f_1 f_2}} + \overline{\overline{f_3}}$$

$$= f_1 f_2 + f_3$$

intersection union

$$f_1 \cap f_2 = \Sigma m(6)$$

$$(f_1 \cap f_2) \cup f_3 = \Sigma m(1,4,5,6,7)$$

Q26) what happens when XORed with itself n times as shown below

$$B \oplus (B \oplus (B \oplus (B \oplus \ldots \text{n times}$$

a) complements when "n" is even

b) complement when "n" is odd

c) remains unchanged when n is even

d) remains unchanged when n is odd

Sol :

$0 \oplus 0 = 0$

$1 \oplus 1 = 0$

a d

$0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$ b d

$1 \oplus 1 \oplus 1 = 1 \oplus 0 = 1$

$0 \oplus 0 \oplus 0 \oplus 0 = 0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$

$1 \oplus 1 \oplus 1 \oplus 1 = 0 \oplus 1 \oplus 1 = 1 \oplus 1 = 0$

∴ ~~for a from above opt~~ (d)

Here we need to count no of XOR operations (not variables) ∴ opt (c)

130

Q27 $f(P,Q) = \underbrace{[1 \oplus P]}_{\downarrow \; \bar{P}} \oplus \underbrace{[P \odot Q] \oplus [P \oplus Q]}_{\downarrow \; 0} \oplus \underbrace{[Q \oplus 0]}_{\downarrow \; Q}$

$\Rightarrow \bar{P} \oplus 0 \oplus Q$

$\Rightarrow \bar{P} \oplus Q$

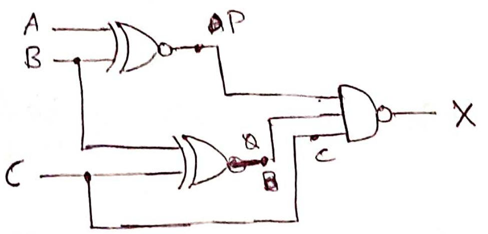
$= \bar{P}\bar{Q} + \bar{\bar{P}} Q$

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

$= P \odot Q$

02/09/20

Q28 For the logic circuit shown in figure



The required condition (A, B, C) to make the output $X=0$ is

a) 1, 1, 1 b) 1, 0, 1 c) 0, 1, 1 d) 0, 0, 1

Sol :

$P = A \odot B \quad Q = B \oplus C$

$X = \overline{PQC}$

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

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

$X = 0 \Rightarrow$

$\Rightarrow A \oplus B = 0$ && $B \odot C = 0$ && $\bar{C} = 0$

$\bar{C} = 0 \downarrow C = 1$

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

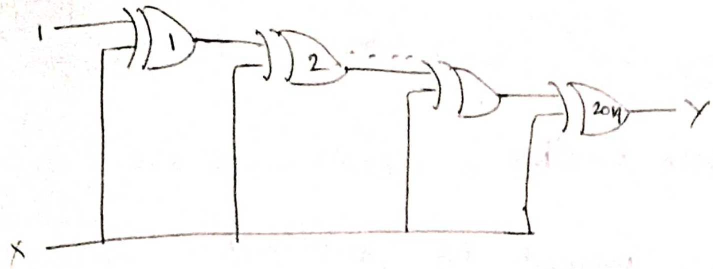
$\Rightarrow B = 0$

$A \oplus 0 = 0 \leftarrow B = 0$

$A = 0$

$\therefore A = 0 \; B = 0, C = 1 \qquad \therefore$ opt (d)

Q29



Then ~~d~~ $Y = ?$

Sol :

$OP_1 = X \oplus 1 = \bar{X}$

$OP_2 = \bar{X} \oplus X = 1$

$OP_3 = 1 \oplus X = \bar{X}$

$OP_4 = \bar{X} \oplus X = 1$

$\therefore OP_{odd} = \bar{X}, \quad OP_{odd} = 1$

$\therefore$ ~~opt~~ Ans : $\bar{X}$

X=0 is

Q30 which of the following is self dual

a) $F(A,B,C) = \Sigma m(2,3,4,6)$

b) $F(A,B,C) = \Sigma m(1,2,6,7)$

c) $F(A,B,C) = \Sigma m(0,4,5,7)$

d) $F(A,B,C) = \Sigma m(3,5,6,7)$

Sol :

self dual $\Longleftrightarrow [\forall i \in "m" \Rightarrow 2^n - 1 - i \in "M"]$

~~$\therefore$ opt a~~ $(0,7)\ (1,6)\ (2,5)\ (3,4)$

One of these elements is present in self dual function

$\therefore$ opt d

K-Maps (karnough maps)

* used for simplifying boolean function

* It is graphical & systematic method for simplification

* k-maps consists of cells.

  n-variable kmap contains 2^n cells.

* each cell represents a minter (SSOP) or maxterm (SPOS)

method of grouping in k-map:

* 1st group highest possible no of cells
* group of 2 terms is called a pair
* group of 4 terms is called a Quad
* group of 8 terms is called a ~~octate~~ octet

k-map ~~Simplicat~~ Simplification for SSOP:

SSOP ⟶ k-map ⟶ SOP

3-variable k-map:

~~gray codes (for unit distance)~~

* 8 cells

MSB ← A

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

alternate way

AB → MSB

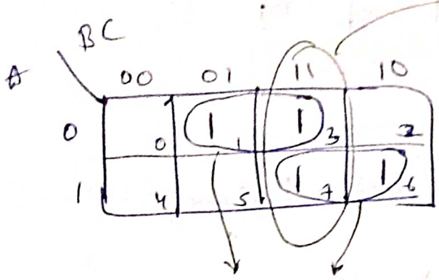
LSB ← C

C \ AB	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

Eg: Simplify following boolean function using k-map

$$f(A,B,C) = \underset{m_1}{\bar{A}\bar{B}C} + \underset{m_3}{\bar{A}BC} + \underset{m_6}{AB\bar{C}} + \underset{m_7}{ABC} \hookrightarrow \text{ssop}$$

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



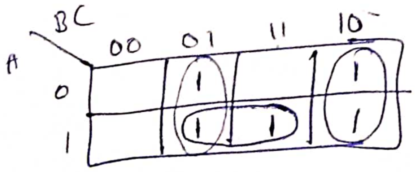
→ BC this is redundant group

∴ In a group atleast one term should be independent

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

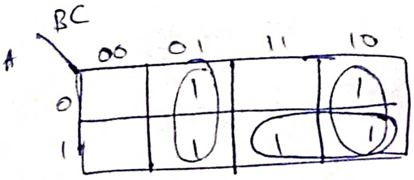
$$\therefore f(A,B,C) = \bar{A}C + AB$$

Eg: $f(A,B,C) = \sum m(1,2,5,6,7)$



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

Method 2:



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

Both are simplified since no of literals are same.

Note:

→ In k-map, simplified expressions need not to be unique.

4-variable K-map:

MSB ← AB \ CD → LSB

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

Eg: Simplify $f(A,B,C,D) = \Sigma m(0,2,4,6,12,14)$

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

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

(Q31) For the boolean function

$f(A,B,C,D) = \Sigma m(1,5,6,7,11,12,13,15)$ the minimal SOP is

a) $f = \bar{B}\bar{C}D + \bar{A}BC + BD$

b) $f = \bar{A}\bar{C}D + \bar{A}BC + ACD + AB\bar{C} + BD$

c) $f = \bar{A}\bar{C}D + \bar{A}BC + AB\bar{C} + \bar{A}CD + BD$

d) none

Sol:

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

This is not considered, Cuz it has no independent term.

$\therefore f = AB\bar{C} + \bar{A}\bar{C}D + \bar{A}BC + ACD$

$\therefore$ opt (d)

none

Don't Care:

→ when we don't care about an o/p, we consider it don't care

Eg: Simpify the following boolean function

$$F(A,B,C) = \Sigma m(0,1,3,7) + \Sigma d(2,4,5)$$

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

$\therefore$

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

* group minterms is compulsory, but grouping of all don't cares is not needed.

K-Map Simplification for SPOS:

SPOS → KMAP → POS

* Each cell indicates Max term.

Eg: $f(A,B,C) = \Pi M(0,3,4,7)$

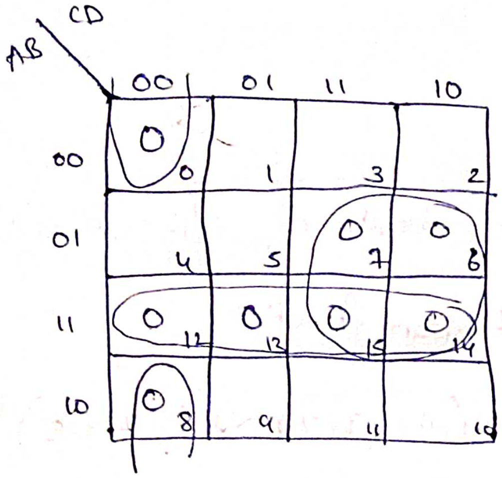
A \ BC	00	01	11	10
0	0_{0}	$_{1}$	0_{3}	$_{2}$
1	0_{4}	$_{5}$	0_{7}	$_{6}$

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

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

Eg: Simplify $f(A,B,C,D) = \Pi M(0,6,7,8,12,13,14,15)$

Sol:



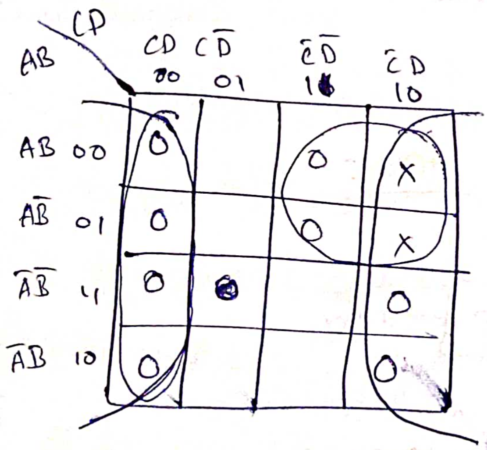
$$f(A,B,C,D) = (B+C+D)(\bar{A}+\bar{B})(\bar{B}+\bar{C})$$

03/09/20

K-Map for SPOS with Don't care:

Eg: Simplify

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



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

Note:

for n variable k-map, group 2^k terms gives a ~~literal of~~ term with n-k no of literals.

Consensus theorem :

$$\boxed{xy + \bar{x}z + yz = xy + \bar{x}z}$$

Proof :

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

Special Cases in K-Map:

Case (i):

$$f_1(A,B,C) = \sum m(1,3,6,7)$$

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

$$f_1 = \bar{A}C + AB$$

Case (ii)

$$f_2(A,B,C) = \Pi M(0,2,4,5)$$

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

$$f_2 = (A+C)(\bar{A}+B) = \underbrace{A\bar{A}}_{0} + AB + \bar{A}C + BC$$

$$f_2 = AB + \bar{A}C \quad (\because \text{Consensus theorem})$$

Case (iii)

$f_3(A,B,C) = \pi M(1,3,6,7)$

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

$f_3 = (A+\bar{C})(\bar{A}+\bar{B})$

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

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

$f_3 = A\bar{B} + \bar{A}\bar{C}$ ($\because$ consensus theorem)

from case(i) & case (ii)

$f_1 = f_2$

Consider

$f_1 * f_3 = (\bar{A}C + AB)(A\bar{B} + \bar{A}\bar{C})$

$= 0 + 0 + 0 + 0$

$= 0$

Product of two function is zero iff they are complements to each other.

$\therefore f_1 = \bar{f_3}$

$\therefore \Sigma m(1,3,6,7) = \overline{\pi M(1,3,6,7)}$ (Same terms $\Rightarrow$ complement)

$\Sigma m(1,3,6,7) = \pi M(0,2,4,5)$ (absent terms $\Rightarrow$ equal)

XOR & XNOR

A B	$A \oplus B$	$A \odot B$
0 0	0	1
0 1	1	0
1 0	1	0
1 1	0	1

XOR

O/P = 1, if no of 1's is odd

Hence XOR is called odd logic gate.

XNOR :

O/P = 1, if no of 1's is even

Hence XNOR is called even logic gate

Note :

- 2 i/p XNOR is called equality detector or coincident ~~dector~~ detector
- 2 i/p XOR is called inequality detector or anti-coincident detector

Prime Implicants (PI)

Implicant : It is a set of all adjacent minterms.

i.e., Group of all possible combinations of octet, quad, pair etc.

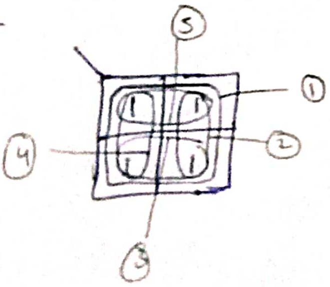
(Here we don't need to follow k-map redundant rules)

Prime implicant :

It is implicant which is not a subset of another implicant.

(no need to follow redundant property)

Case(i)



Here ② is subset of ①

③ " " " ①

④ " " " ①

⑤ " " " ①

∴ ① is the only prime implicant, but not ② ③ ④ ⑤

Case (i) :

→ Prime implicants are classified into 2 types

i) Essential Prime Implicant

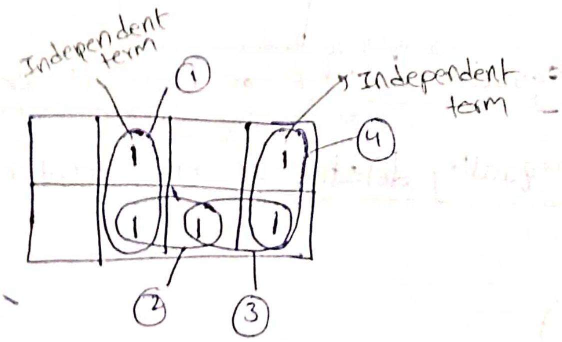
ii) Non-essential prime implicant

(iii) Selective PI (iv) Redundant PI

Essential Prime Implicant :

It is a PI which contains ~~ats~~ atleast one independent minterm from other prime implicants.

Case (ii) :

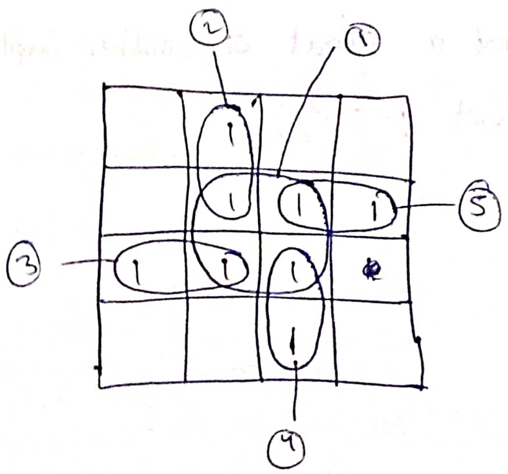


PIs ⇒ ①, ②, ③, ④

EPIs ⇒ ① ④

NEPIs ⇒ ② ③

Case (iii) :



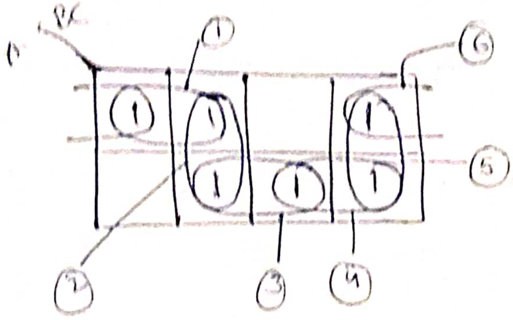
PIs ⇒ ① ② ③ ④ ⑤

EPIs ⇒ ② ③ ④ ⑤

NEPIs ⇒ ①

Find no of PI, EPI

$$f(A,B,C) = \Sigma m(0,1,2,5,6,7)$$



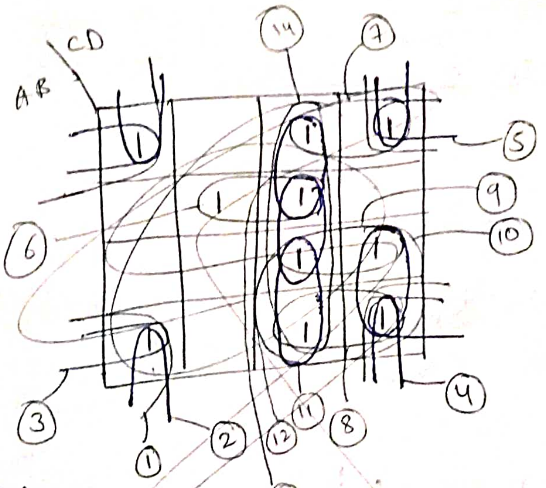
no of PIs = 6

no of EPIs = 0

2) Find PI, EPI for function

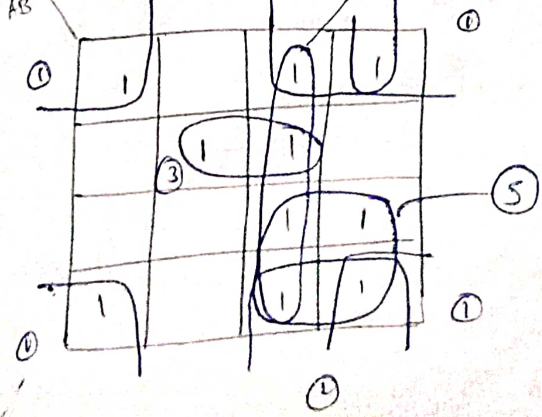
$$f(A,B,C,D) = \Sigma m(0,2,3,5,7,8,10,11,14,15)$$

Sol:



∴ ~~no of PIs~~ =

PIs => 1 6 7 10 14



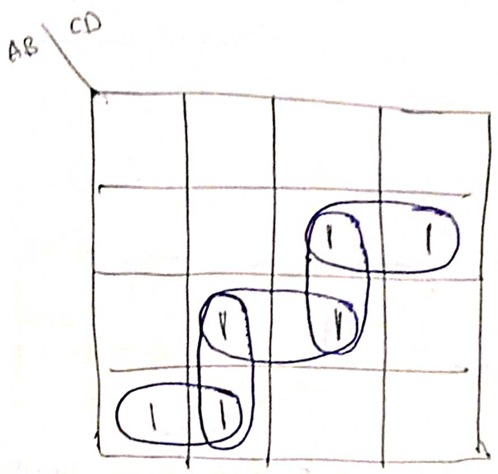
$\therefore$ no of PIs = 5

no of EPIs = 3 (i.e., [illegible] 1 2 5)

Q33 Find no of PIs, EPIs for

$f(A,B,C,D) = \Sigma m(6,7,8,9,13,15)$

Sol:



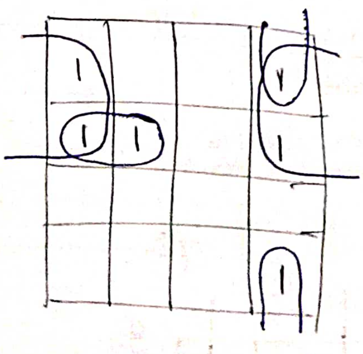
$\therefore$ no of PIs = 5

no of EPIs = 2

Q34 Find no of PIs of function

$f(w,x,y,z) = \Sigma m(0,2,4,5,6,10)$

Sol:



$\therefore$ no of PIs = 3

Combinational Logic Circuits

→ The circuit in which present o/p depends on present input but not previous data is called as combinational logic circuit.

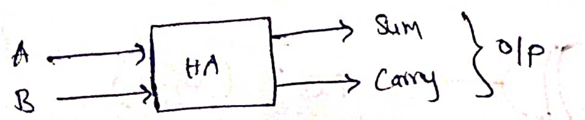
→ Adders, subtractors, Magnitude comparator, Encoder, Decoder, Multiplexer & Demultiplexers.

04/09/20

Adders & Subtractors

Half Adders (HA):

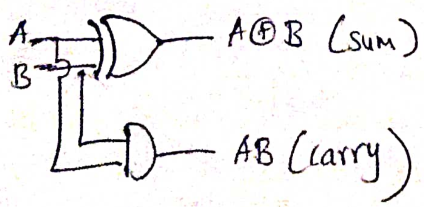
→ It is a combinational logic circuit that contains 2 i/ps & 2 o/ps.



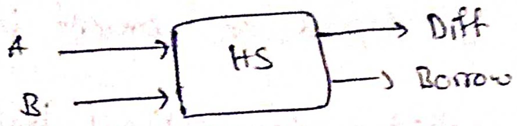
truth table

Inputs		o/p	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\boxed{\begin{aligned} Sum &= A \oplus B \\ Carry &= AB \end{aligned}}$$



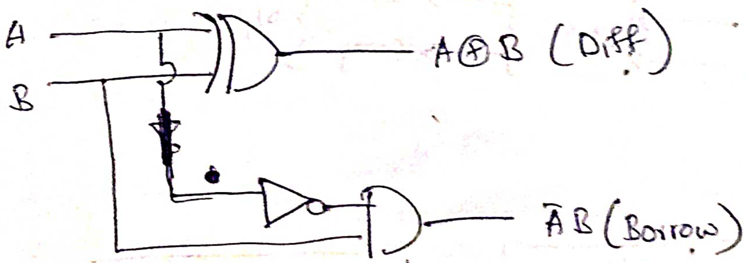
Half Subtractor (HS)



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

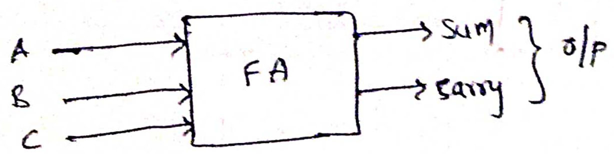
$\therefore$ Difference $= A \oplus B$

Borrow $= \bar{A} B$



Full Adder:

→ 3 i/ps & 2 o/ps



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

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

$FA_{carry} = \sum m(3,5,6,7)$

sum

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

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

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

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

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

$$\boxed{FSum = A \oplus B \oplus C}$$

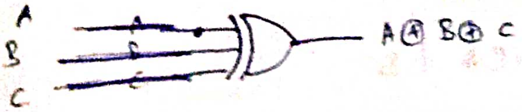
carry

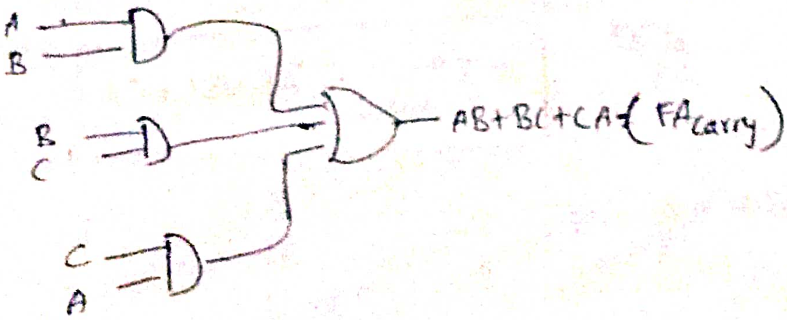
A \ BC	00	01	11	10
0			1	
1		1	1	1

$$FA_{carry} = BC + AC + AB$$

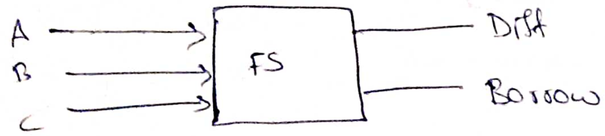
$$\boxed{FA_{carry} = AB + BC + CA}$$

<u>logic diagram :</u>





Full Subtractor:



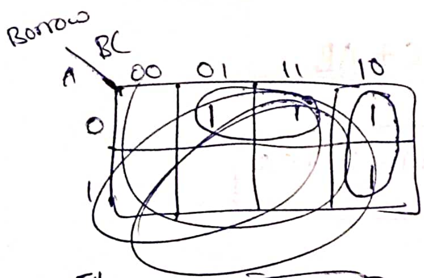
A B C	Diff	B
0 0 0	0	0
0 0 1	1	1
0 1 0	1	1
0 1 1	0	1
1 0 0	1	0
1 0 1	0	0
1 1 0	0	0
1 1 1	1	1

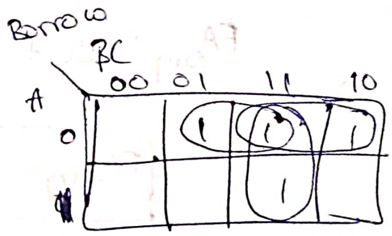
$(A-B)-C$

$\therefore FS_{Diff} = \Sigma m(1,2,4,7)$

$$\boxed{\therefore FS_{diff} = A \oplus B \oplus C}$$

$FS_{Borrow} = \Sigma m(1,2,3,7)$



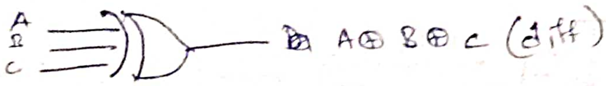


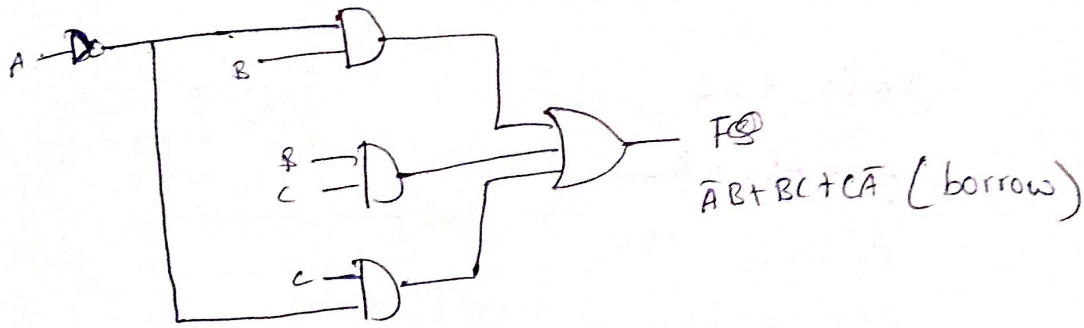
~~$FA_{borrow} = \bar{A}C + BC$~~

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

$$\boxed{\therefore FS_{borrow} = \bar{A}B + BC + C\bar{A}}$$

Logic diagram





Note :

HA — Sum : $A \oplus B$
HA — Carry : AB

HS — diff : $A \oplus B$
HS — borrow : $\bar{A}B$

FA — Sum : $A \oplus B \oplus C$
FA — Carry : $AB + BC + CA$

FS — Diff : $A \oplus B \oplus C$
FS — borrow : $\bar{A}B + BC + C\bar{A}$

FA using HA :

(535) To design number of half adders ; basic gates required to design full adder respectively are ______

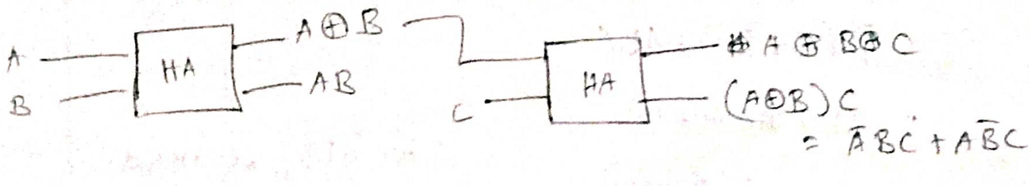
Sol :

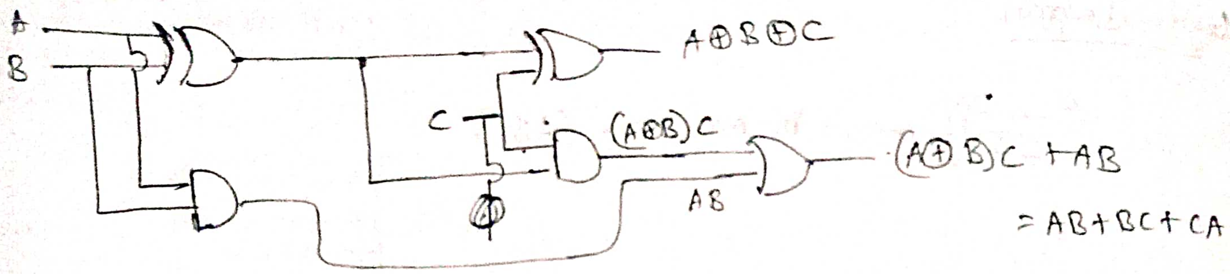
$FA_{sum} : A \oplus B \oplus C$

$FA_{carry} = AB + BC + CA$

$HA_{sum} = A \oplus B$

$HA_{carry} = AB$





$(A \oplus B)C + AB$

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

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

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

$AB + AC + \bar{A}BC$

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

$AC + B(A + C)$

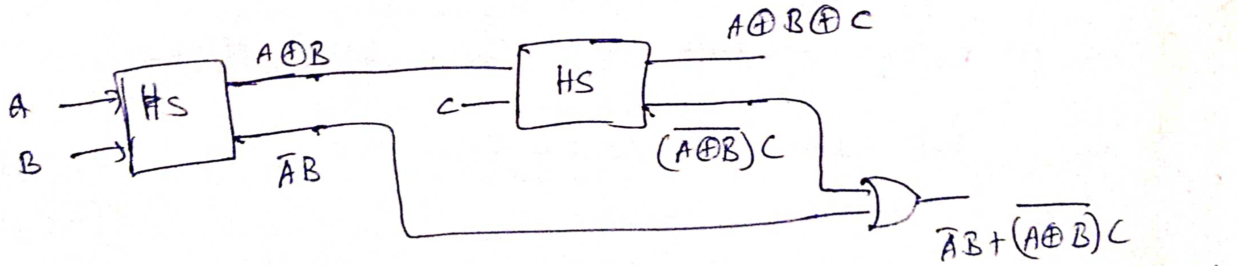
$\cancel{BC + AB}$

$= AC + AB + BC$

$\therefore (A \oplus B)C + AB = AB + BC + AC$

$\therefore$ 2 HA and 1 basic gate (OR)

Full Subtractor using half subtractor



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

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

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

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

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

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

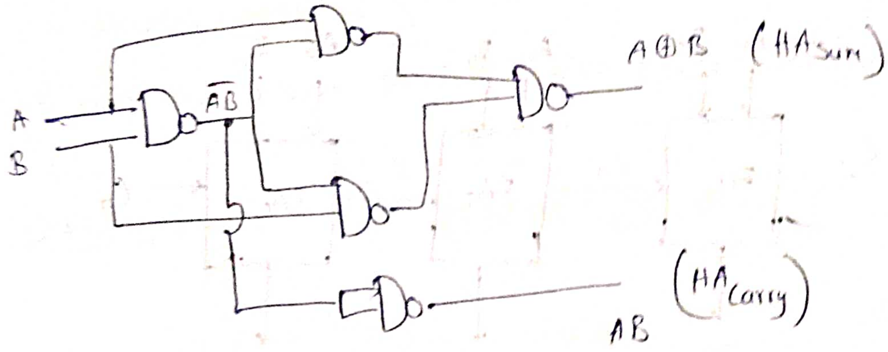
$= \bar{A}B + C(\bar{A} + AB)$

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

$\therefore$ For FS, we need 2 HS & 1 OR

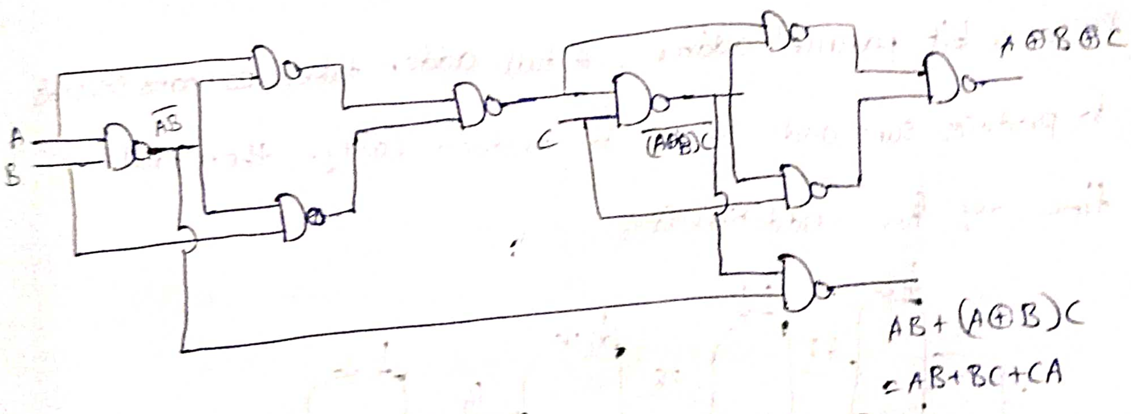
Design of Adders, Subtractors using universal logic gates

HA using NAND



→ For HA min no of NAND gates required = 5

FA using NAND



∴ 9 NAND gates for FA.

Note:

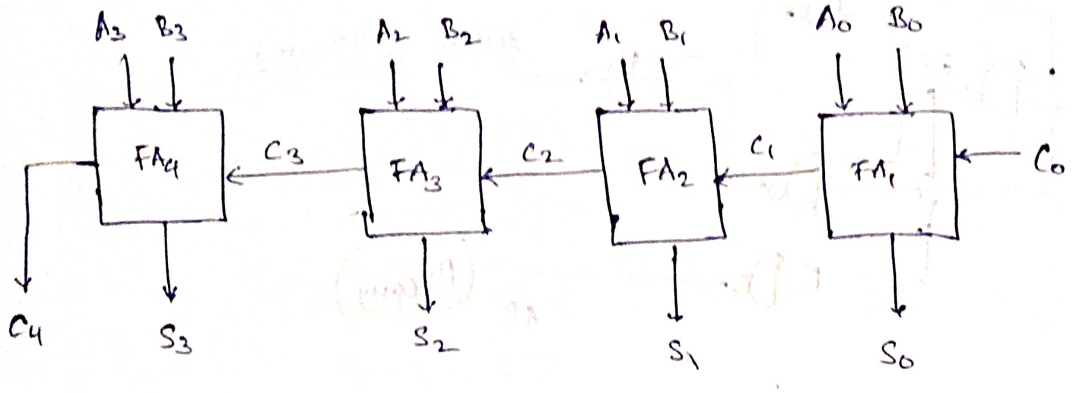
	NAND	NOR
HA	5	5
HS	5	5
FA	9	9
FS	9	9

15/10/20

150

Binary Adder or Parallel Adder:

Consider 4 bit parallel adder as shown below

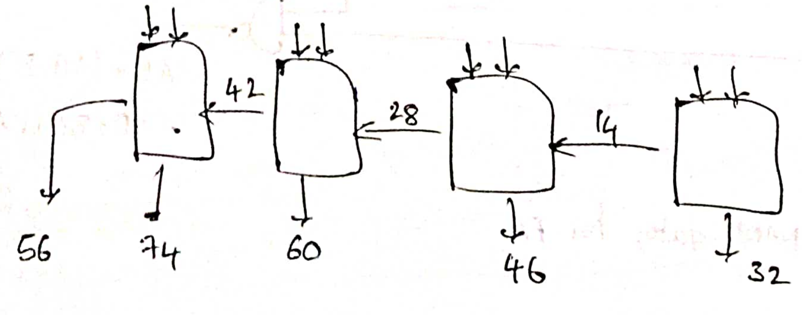


(Q36) To design 17 bit parallel adder, find min no of HA, basic gates req.

Ans: 33 HA, 16 OR gates

∴ 33, 16

(Q37) In a 4 bit parallel adder, A full adder takes 32 nano seconds to produce sum and 14 ns to produce carry. then total time req for addition is



∴ 74 ns

Note:

For an n-bit parallel adder, time req to produce sum

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

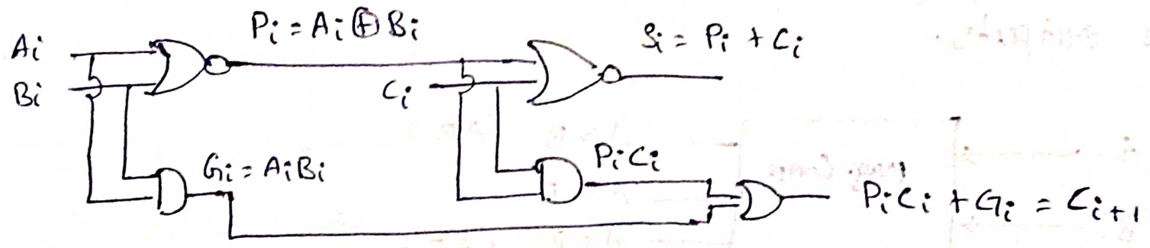
disadvantage:

→ In binary adder carry at any stage depends on previous carries. Hence it takes more time for addition.

→ It is slowest adder.

→ we can overcome above problem using carry lookahead adder.

Carry Lookahead Adder (CLA)



$P_i \rightarrow$ Carry Propagate

$G_i \rightarrow$ Carry Generate.

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

$i=0 \Rightarrow C_1 = G_0 + P_0 C_0$

$i=1 \Rightarrow C_2 = G_1 + P_1 C_1$

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

$= G_1 + P_1 G_0 + P_1 P_0 C_0$

$i=2 \Rightarrow C_3 = G_2 + P_2 C_2$

$= G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0)$

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

$i=3 \Rightarrow C_4 = G_3 + P_3 C_3$

$= G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0)$

$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

→ At any stage carry look ahead adder is independent of previous carries. Hence CLA is faster than binary adder.

→ But, internal circuit of CLA consists ~~of~~ many AND gates, OR gates

for 'n' bit CLA,

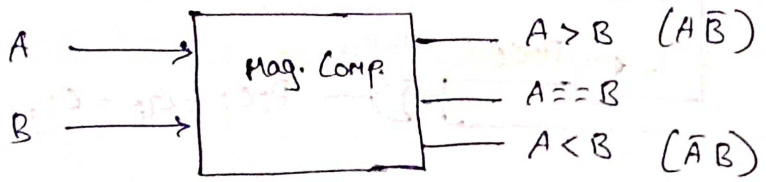
no of AND gates = $\frac{n(n+1)}{2}$

no of OR gates = n

} This is valid when all G_i & P_i are available and AND, OR can take any no of inputs

Magnitude Comparator:

It is a combinatitions circuit which consists of 2 inputs and 3 outputs.



→ In n bit mag. Comp. no of combinations = $2^n \cdot 2^n = 2^{2n}$

→ In n bit mag. comp no of i/p for which

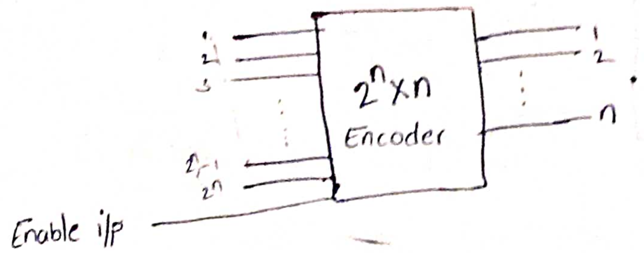
* A = B is 2^n
* A > B is $\frac{2^{2n} - 2^n}{2}$
* A < B is $\frac{2^{2n} - 2^n}{2}$

~~Code Converters~~:

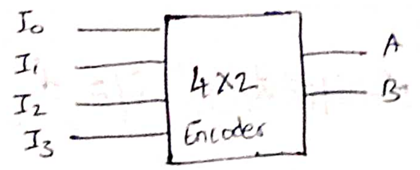
Encoders & Decoders

Encoder :

It is a combinational logic circuit which consists of 2^n inputs and n outputs.



Eg:



→ Encoder is used to reduce no of data lines in the O/P.

→ It converts decimal or (any number system) to binary.

Design 4x2 encoder

I/P				O/P	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

In encoder at a time one i/p should be active.

Disadvantage:

→ If more than one i/p are active then encoder provides invalid output.

→ we can ~~over o~~ overcome above problem using priority encoder.

4×2 Priority Encoder:

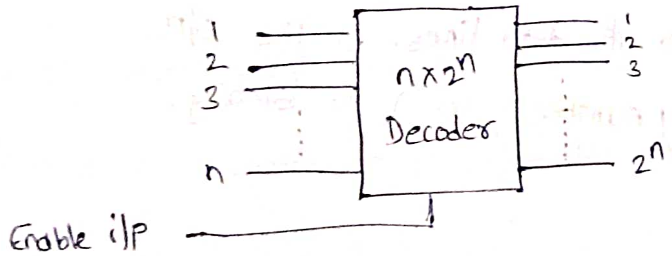
I/P				O/P	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

I_k has higher priority than I_j $\forall$ $j<k$

Decoder:

→ It is inverse of encoder.

→ Defn: Decode is a CLC with ~~2^n i/p &~~ 'n' i/p & 2^n o/p.



enable i/P:

Enable i/P is used to control the operation of encoder or decoder.

Types of enable i/p:

(i) Active high enable i/P:- '0' – disable; '1' – Enable

(ii) Active Low enable i/P – '0' – Enable; '1' – Disable

Decoder converts binary to decimal (or any other number system)

Based on o/p nature, there are 2 types of decoders

i) Active high o/p decoder:

If active high o/p decoder is disabled all o/p bits are forced to 0.

(ii) Active low o/p decoder:

If active low o/p decoder is disabled, then all o/p bits are forced to 1.

Design 2×4 Decoder:

i) Active high o/p decoder with ~~wit~~ active high enable I/P:

(ii) Active high o/p decode with active low enable I/P.

(iii) Active low o/p decoder with active high enable I/P

(iv) Active low o/p decoder with active low enable I/P

Active high o/p decoder with active high EI:

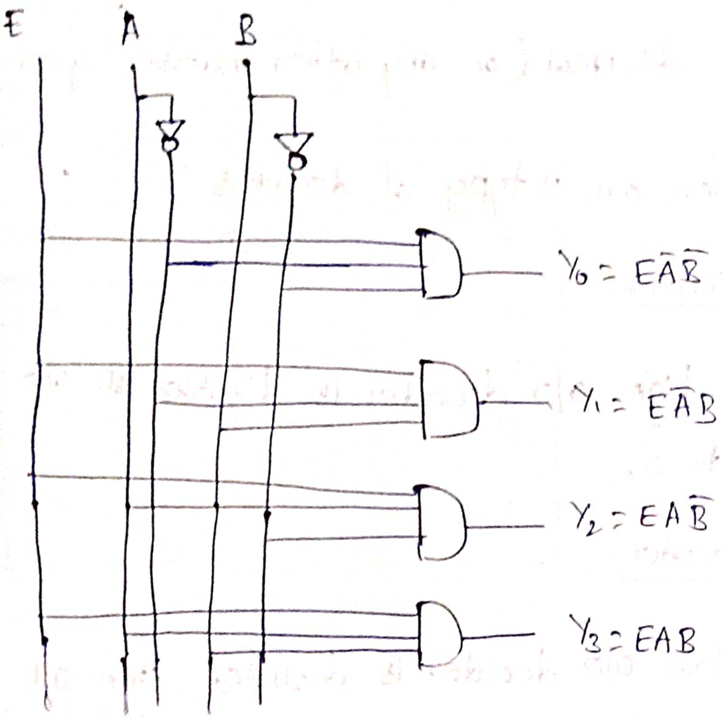
I/P			O/P			
EI	A	B	Y_0	Y_1	Y_2	Y_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

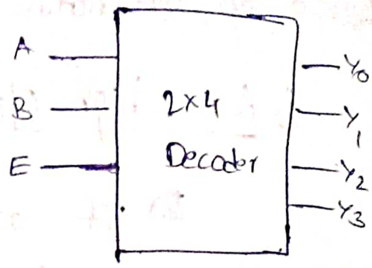
$\therefore Y_0 = E\bar{A}\bar{B}$

$Y_1 = E\bar{A}B$

$Y_2 = EA\bar{B}$

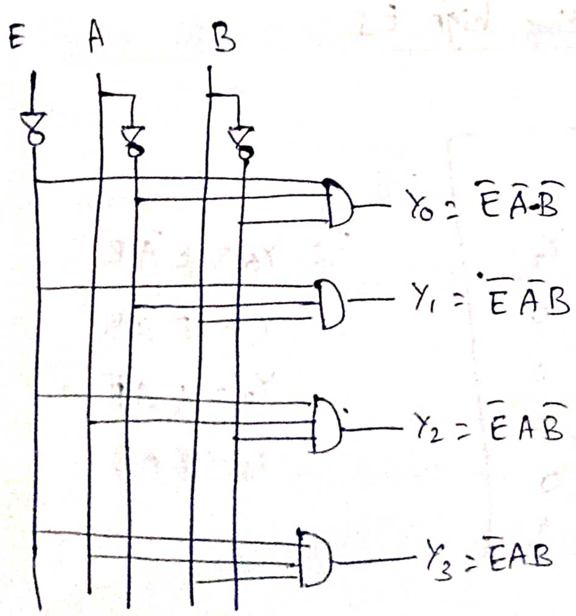
$Y_3 = EAB$

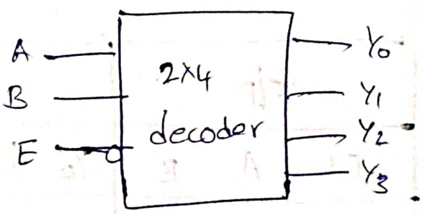




Active high o/p decoder with active low EI :

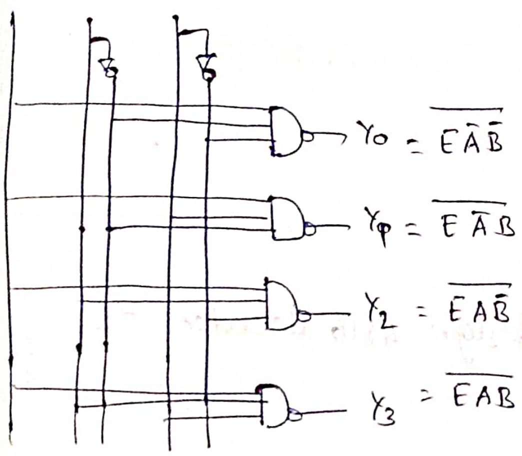
I/P			O/P			
E	A	B	Y_0	Y_1	Y_2	Y_3
1	X	X	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

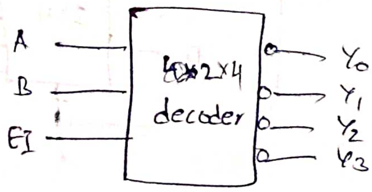




Active low o/p decoder with active high EI :

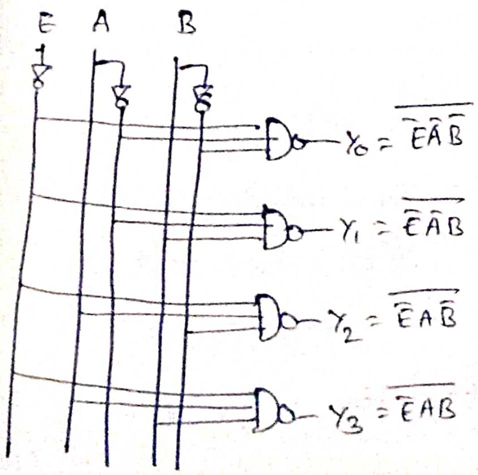
I/P			O/P			
E	A	B	Y_0	Y_1	Y_2	Y_3
0	X	X	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	0

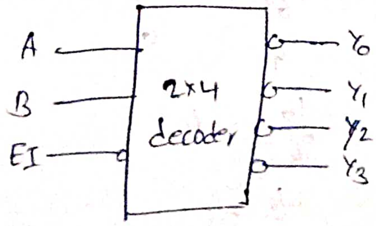




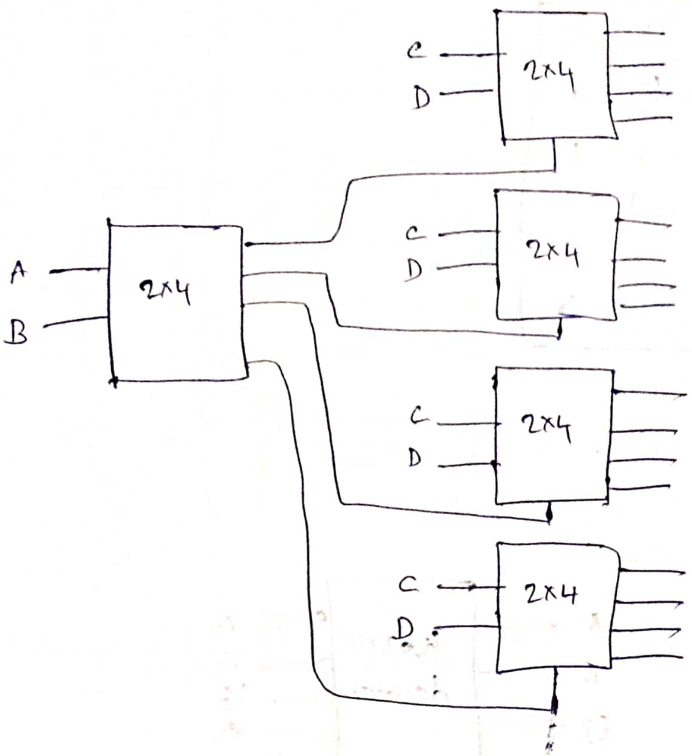
Active low o/p decoder with active active low EI

I/P			O/P			
E	A	B	Y_0	Y_1	Y_2	Y_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0





Design 4x16 decoder using 2x4 decoders.



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

∴ no of 2x4 decoder req. to design 4x16 decoder = 5.

Shortcut :

→ Recursively divide no of o/p lines req with given no of o/p lines until we obtain 1.

Eg : Find no of 2x4 decoders req to design 10x1024

$$\Rightarrow \frac{1024}{4} = 256$$

$$\frac{256}{4} = 64$$

$$\frac{64}{4} = 16$$

$$\frac{16}{4} = 4$$

$$\frac{4}{4} = 1$$

∴ no of 2x4 decoders req = 1+4+16+64+256

= 341