Combinational Ckts - I

Comprehensive Course on Digital Logic Design 2023/2024

B V Reddy • Lesson 22 • Nov 20, 2022

NUMBER SYSTEMS

Any number is associated with **Base** (or) **Radix**

$$(734)_{10}$$

$(734)_{10}$ =

(472.15) =

A number system with base ‘ ***b***’ , will have ***b*** different digits and they are from ***0 to b -1*** .

$(421)_4$

$(243)_5$

$(851)_9$

Base (b) is always a positive integer .

In general b $\geq$ 2

Base	**Different digits**
2 (Binary)	
8(Octal)	
10 (Decimal)	
16 (Hexadecimal)	

Conversation of Number System

1. Decimal to Any Base

$$[N]_{10} \to [?]_b$$

2. Any base to Decimal

$$[N]_b \to [?]_{10}$$

3. one base to another base

$$[N_1]_{b_1} \to [?]_{b_2}$$

4. Required base = $(\, Given\ Base\,)^{integer}$

1. Decimal to Other Base

- Integer part, repeated division by the required base .
- Fractional part , repeated multiplication by the required base .

Q) $(53.75)_{10} = \quad (- - - - -)_2$

Q) $(0.15)_{10} = \quad (- - - - -)_2$

Note :

It is possible to obtain the equivalent of integer part but may not possible for fractional part .

Q) $(53.75)_{10} = \quad (-----\ -)_4$

Q) $(39.5)_{10} = \quad (-----\ -)_8$

Q) $(39.5)_{10} = \quad (-----)_{16}$

2. Any base to Decimal

$$(x_2x_1x_0\ .\ x_{-1}x_{-2}x_{-3})_b = (\qquad ? \qquad)_{10}$$

Q) $(311.30)_4 = \quad (-----)_{10}$

Q) Find the minimum decimal equivalent of $(3AB26)_x$

Q) $(137.4)_8 = \quad (- - - - -)_{10}$

Q) $(DAD)_{16} = \quad (-----\ -)_{10}$

Q) $(ECE)_{16} = \quad (-----\ -)_{10}$

Q) $(EEE)_{16} = \quad (-----\ -)_{10}$

Q) Find b if $\sqrt{(41)_b} = (5)_{10}$

3.One base to another base

$$[N]_{b_1} \rightarrow [?]_{b_2}$$

1. Convert the given number to the decimal system
2. After that convert to required base

Q) $(3)_4 = (\ ?\ )_8$

Q) $(7)_8 = (\ ?\ )_9$

Q) Find the value of x if $(193)_x = (623)_8$

Q) Find $b_1 and\ b_2$ if $(235)_{b_1} = (565)_{10} = (1065)_{b_2}$

Q) The solution to the quadratic equation $x^2 - 11x + 22 = 0$ is x =3 and x =6 , what is the base of the system

4. Required base = $(Given\ Base\ )^{integer}$

Q) $(1011010110.010110010)_2 = (-----\ -\ )_8$

Q) $(10110101 10.010110010)_2 = (-----\ -)_4$

Q) $(1011010110.010110010)_2 = (-----\ -\ )_{16}$

Q) $(2210121012.2011022)_3 = (-----)_9$

Q) $(3210332101.2210)_4 = (-----\ -)_{16}$

Q) Find the number of solutions of ‘ Y ‘ exists for $(123)_5 = (X8)_Y$

Q) Find the number of solutions of ‘ x ‘ exists for $(123)_X = (12X)_3$

Q) Find the base of the following system such that given operation is valid

$24+14 = 41$

Q) Find the base of the following system such that given operation is valid

$$\frac{66}{6} = 11$$

Q) Find the base of the following system such that given operation is valid

$$\sqrt{\mathbf{121}} = \mathbf{11}$$

Q) Find the base of the following system such that given operation is valid

$$\sqrt{\mathbf{41}} = \mathbf{5}$$

Complement Analysis

$$|N|_r$$

1. **r' s complement**

2. (r-1)'s complement

Binary (r=2)

1's complement

2's complement

Octal (r=8)

7's complement

8's complement

Decimal (r=10)

9’s complement

10’s complement

Hexadecimal (r=16)

F’s complement

16’s complement

r ’ s complement

r’ s complement of the number (N) $= r^n - N$

r ---------> Radix

n --------> number of integer digits

N ---------> given number

(r-1) ' s complement

(r-1) ' s complement of the number (N) $= r^n - r^{-m} - N$

if m=0

$r^n - N - 1$

(r's comp) − 1

r --------> Radix
n -------> number of integer digits
m -------> number of decimal digits
N --------> given number

$\boxed{r\text{'s comp} = (r-1)\text{'s comp} + 1}$

(r-1) ' s complement of the number (N) $= r^{n} - r^{-m} - N$

r' s complement of the number (N) = (r-1)'s complement + r^{-m}

If m = 0

r' s complement of the number (N) = (r-1)'s complement + 1

Q) Find the 10's complement of $(327.452)_{10}$

$n = 3$

$r = 10$

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

10's complement $= (10^3)_{10} - (327.452)_{10}$

$= 1000 - 327.452$

$= (672.548)_{10}$

Q) Find the 9's complement of $(327.452)_{10}$

$r = 10$

$(r-1)$'s complement $= r^n - r^{-m} - N$

9's complement $= (10^3)_{10} - (10^{-3})_{10} - (327.452)_{10}$

$= 1000 - 0.001 - 327.452$

$= (672.547)_{10}$

Trick

$N = (327.452)_{10}$

$$\begin{array}{r} 999.999 \\ 327.452 \\ \hline 672.547 \end{array}$$

9's complement = 672.547

$$\begin{array}{r} +1 \\ \hline \end{array}$$

10's complement = 672.548

Q) Find the 10's complement of $(784732179)_{10}$

9's complement = 215267820

10's complement = 215267821.

Q) Find the 2's complement of $(101100)_2$

$r = 2$

1's complement = 010011

2's complement = 010011

$$\begin{array}{r} 010011 \\ +\quad 1 \\ \hline 010100 \\ \hline \end{array}$$

N = 101100

2's comp = 010100

Q) Find the 2's complement and 1's complement of $(\underline{0}.0110)_2$

1's complement = 0.1001

$$\begin{array}{r} 0.1001 \\ 1 \\ \hline 0.1010 \\ \hline \end{array}$$

2's complement = 0.1010

Q) Find the 9's and 10's complement of $(52520)_{10}$

9's complement = 47479

10's complement =

$$\begin{array}{r} 47479 \\ 1 \\ \hline 47480 \\ \hline \end{array}$$

Q) Find the 9's and 10's complement of $(\underline{0}.3267)_{10}$

9's complement $= 0.6732$

10's complement $= 0.6733.$

Q) Find 1's and 2's complement of $(10100100111)_2$

1's complement = 01011011000

2's complement = 01011011001

Q) Find 8's and 9's complement of $(278421)_9$

$r = 9$

8's   9's.

9's complement $= (610468)_9$

8's complement $=$

$$\begin{array}{cccccc} 8 & 8 & 8 & 8 & 8 & 8 \\ 2 & 7 & 8 & 4 & 2 & 1 \\ \hline \end{array}$$

$$(6\;1\;0\;4\;6\;7)_9$$

Q) Find F's and 16's complement of $(792410)_{16}$

$r = 16$

F's 16's

$16 - b = 16 - 16 = 0$

F's complement $= (86DBEF)_{16}$

$$\begin{array}{r} 15 \\ 1 \\ \hline 16 \end{array}$$

16's complement $=$

$$\begin{array}{r} 86DBEF \\ 1 \\ \hline 86DBF0 \end{array}$$

$$
\begin{array}{r}
7482 \\
3898 \\
{\scriptstyle 1\,1\,1\,1} \\
\hline
11380 \\
\hline
\end{array}
$$

$\underline{b = 10}$

$10 - b = 10 - 10$

$= 0$

$18 - 10 = 8$

$13 - 10 = 3$

$11 - 10 = 1$

Q) Find 1's and 2's complement of $(11000100)_2$

1's complement = 00111011

2's complement = 00111100

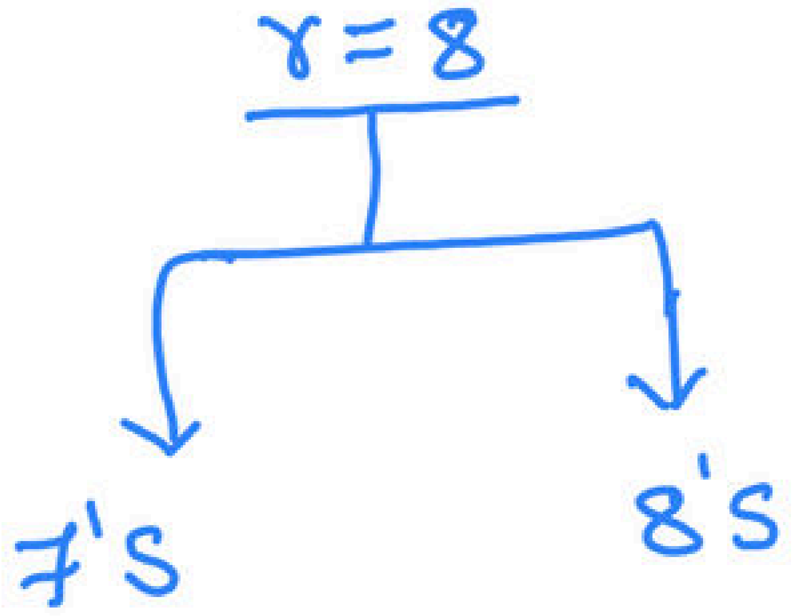
Q) Find 1's and 2's complement of $(11010.11)_2$

1's complement = 00101.00

2's complement = 00101.01

Q) Find 8's complement of $(2670)_8$

$8-8=0$

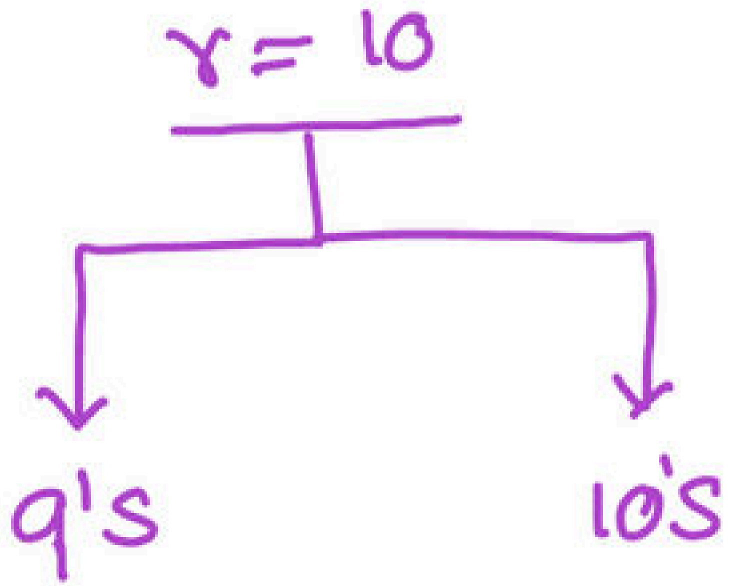


$$\begin{array}{r} 5107 \\ 11 \\ \hline 5110 \\ \hline \end{array}$$

7's complement $= (5107)_8$

8's complement $= (5110)_8$

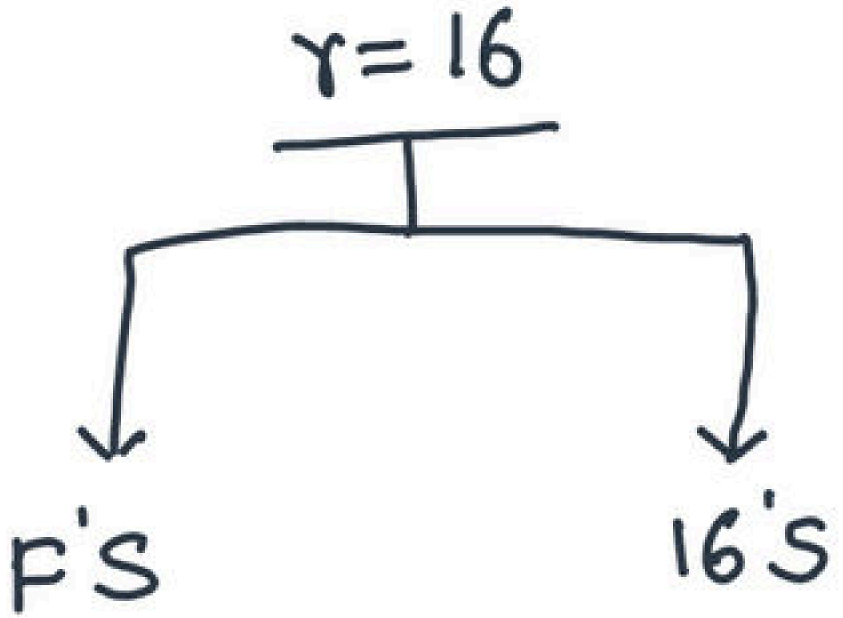
Q) Find 10's complement of $(7492)_{10}$



9's complement $= (2507)_{10}$

10's complement $= (2508)_{10}$

Q) Find 16's complement of $(9623)_{16}$



F's complement $= (69DC)_{16}$

16's complement $= (69DD)_{16}$.

Data Representation

- Unsigned number representation
  - 9 ⊕ve numbers only
- Signed number representation
  - Signed Magnitude Representation
  - 1's complement Representation
  - 2's complement Representation

Unsigned Number Representation

- Strictly applicable for positive numbers
- There is no sign bit concept

+ 5 ------> 101

- 5 -----> X

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

Range with 4 bits = 0 to 15

Range with 5 bits = 0 to 31

$\boxed{\textbf{Range with n- bits} = 0 \text{ to } 2^n - 1}$

Signed Number Representation

1. Signed magnitude representation
2. 1' s complement representation
3. 2's complement representation

Signed Magnitude representation

- ➢Valid for $\boxed{\text{both positive and negative}}$ numbers .
- ➢Sign bit concept is used .

Sign bit $= 0$, for $\oplus$Ve number

$= 1$, for $\ominus$ *ve number*

+5 =	0	1	0	1

- 5 =	1	1	0	1

+5 =	0	0	0	0	0	1	0	1
- 5 =	1	0	0	0	0	1	0	1

Decimal number	Signed Magnitude Representation (4-bits)
+0	0000
+1	0001
+2	0010
+3	0011
+4	0100
+5	0101
+6	0110
+7	0111
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

Range with 4 bits = -7 to $+7$

Range with 5- bits = -15 to $+15$

Range with n- bits = $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

1's Complement Representation

➢In this ⊕*Ve numbers* are represented as *normal binary number with MSB '0'*

+6

1's complement Representation of +6 = 0110

Representation of ⊖ *ve number*

1. Write the binary equivalent of magnitude
2. Take its 1's complement

+6 = 0110

1001

− 6 = 1001

$1001 \rightarrow$ 1's comp. rep.

$= -[0110] = -6$

$11001 \rightarrow$ 1's comp. rep.

$= -[00110] = -6$

$11111001 \rightarrow$ 1's comp. rep.

$= -[00000110] = -6.$

0000000110 = 6

111111111111001 = -6

1's comp. rep.

0111. $\longrightarrow$ 1's comp. rep.

$= +[0111] = +7.$

+6 =

0	1	1	0

0110

-6 =

1	0	0	1

0110

1001

+6 =

0	0	0	0	0	1	1	0

Sign bit extension.

-6 =

1	1	1	1	1	0	0	1

Decimal number	1's complement Representation (4-bits)
+0	0 000
+1	0 001
+2	0 010
+3	0 011
+4	0 100
+5	0 101
+6	0 110
+7	0 111
-0	1 111
-1	1 110
-2	1 101
-3	1 100
-4	1 011
-5	1 010
-6	1 001
-7	1 000

Range with 4 bits = -7 to $+7$

Range with 5- bits = -15 to $+15$

Range with n- bits = $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

2's complement Representation

- In this ⊕**Ve numbers** are represented as ***normal binary number with MSB '0'***

$+3 = \underline{0}\,\underline{011}$

$3 = 011$

Representation of ⊖ *ve number*

1. Write the binary equivalent of magnitude
2. Take its 2's complement

$\leftarrow$

$\underline{+5 = 0101}$

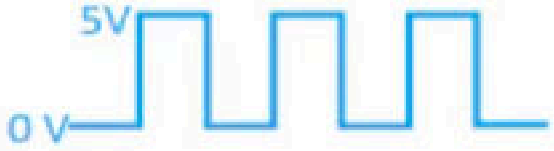
$\underline{-5} = 1011$

Positive logic system

High voltage corresponds to logic " 1"
Maximum positive value is taken as logic '1

+5V ----> logic "1"

0V ----> logic "0 "

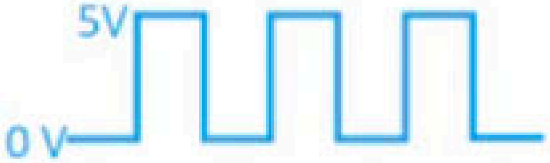


Negative logic system

High voltage corresponds to logic " 0"
Maximum positive value is taken as logic '0 '

+5V ----> logic "0"

0V ----> logic "1 "



A positive logic system is converted into negative logic system by using the concept of duality

Finding the dual of a given Boolean expression

1. $* \leftrightarrow +$
2. $0 \leftrightarrow 1$
3. Keep the variables as it is

BVREDDY

Trick to find Complement

1. Find the dual
2. Complement the each literals

OR -Operation
A+ 0 = A
1+A= 1
A+A=A
$A+\bar{A}=1$

AND-Operation
$A*1 = A$
$A*0 = 0$
$A*A = A$
$A*\bar{A} = 0$

BVREDDY

Commutative Law

A+B= B+A

$A*B = B*A$

Associative Law

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

$A*B*C = (A*B)*C = (B*C)*A = (C*A)*B$

Distribution Law (Mingle wala)

A(B+C)=AB+AC

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

D- Morgan's Law

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

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

Transposition theorem (T- 1)

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

Transposition theorem (T- 2)

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

Consensus theorem (Rajinikanth wala)

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

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

Canonical form : Each minterm (maxterms) contains all the Boolean variables

F(A, B,C)= $ABC + \bar{A}BC + AB\bar{C}$ ---->SOP

F(A, B, C) = $(A+B+C)(A+\bar{B}+C)(\bar{A}+B+\bar{C})$ --> POS

Minimal Form : The minimized form of Boolean expression

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

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

Literal : A Boolean variable either in normal form (or) complimented form is known as literal

Minterm :Each term in canonical SOP representation is known as minterm

Maxterm: Each term in canonical POS representation is known as maxterm

BVREDDY

1. Maximum possible minterms = 2^n

2. Maximum possible maxterms = 2^n

BVREDDY

3. Number of minterm's + number of maxterm's = 2^n

4. The sum of all the minterms = **ONE**

5. The product of all maxterms = **ZERO**

6. Minterm's and maxterm's of same index are **compliment** to each other

7. By using 2- Boolean variables total number of possible Boolean functions = 16

8. By using n- Boolean variables total number of possible Boolean functions = 2^{2^n}

9. By using 2- Boolean variables total number of possible Boolean functions having at most 3- minterms = $4_{C_0} + 4_{C_1} + 4_{C_2} + 4_{C_3} = 15$

10. By using 2- Boolean variables total number of possible Boolean functions having at most 3- maxterms = 15

11. By using 2- Boolean variables total number of possible Boolean functions having 3- minterms = $4_{C_3} = 4$

12. By using n- Boolean variables total number of possible Boolean functions having 2- minterms = $2^n{}_{C_2}$

13. By using 5- Boolean variables total number of possible Boolean functions having at most 3- minterms = $32_{C_0} + 32_{C_1} + 32_{C_2} + 32_{C_3}$

BVREDDY

Neutral Function :

The number of minterms = number of maxterms

Mutually exclusive terms

BVREDDY

The mutually exclusive term of m_i is m_{2^n-i-1}

Self Dual Expression

If one time dual of the Boolean expression result the same expression, then it is called as self dual expression

Eg : f = AB+BC+AC

Conditions for the given expression is self dual

1. The number of minterms = number of maxterms (Neutral Function)

   number of minterms+ number of maxterms = 2^n

   number of minterms = number of maxterms = 2^{n-1}

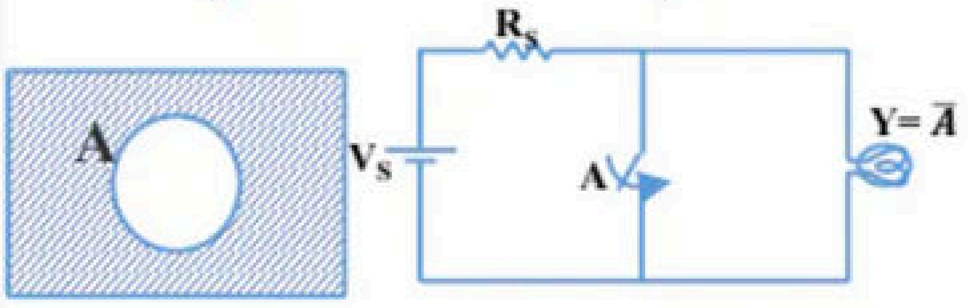
2. If m_i belongs to f, then m_{2^n-i-1} should belongs to $\bar{f}$

3. The number of self dual functions = $2^{2^{n-1}}$

NOT GATE

$Y = \bar{A}$

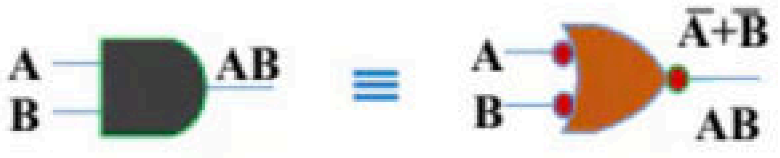
The output is the complement of the input

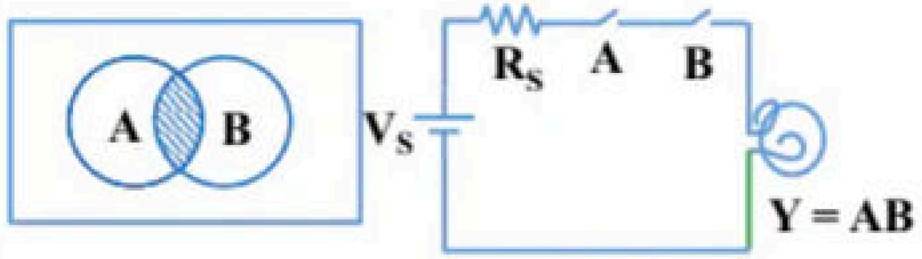


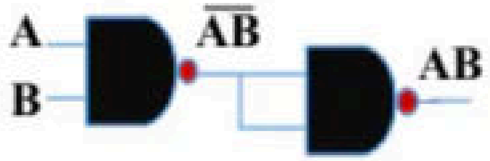
AND GATE

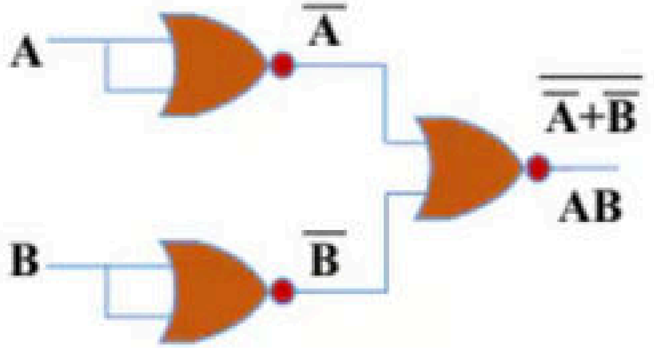
Y = AB

- Output is '0' if any one input '0'
- Y = AB = $\mathbf{\Sigma(3)} = \mathbf{\Pi(0,1,2)}$
- Enable input ⟹ 1
- Disable input ⟹ 0
- Commutative law ⟹ Obeys
- Associative law ⟹ Obeys





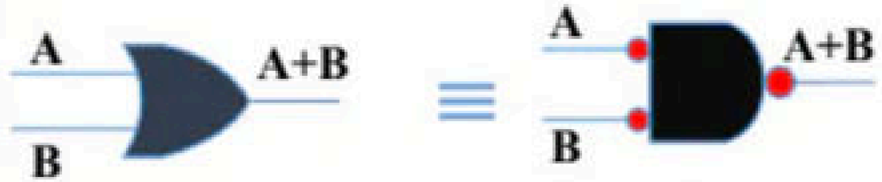


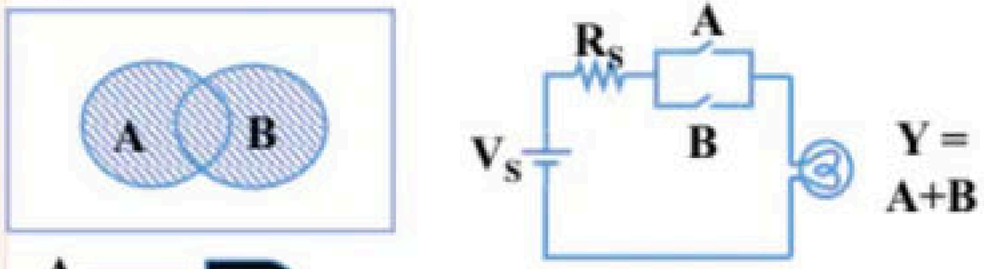


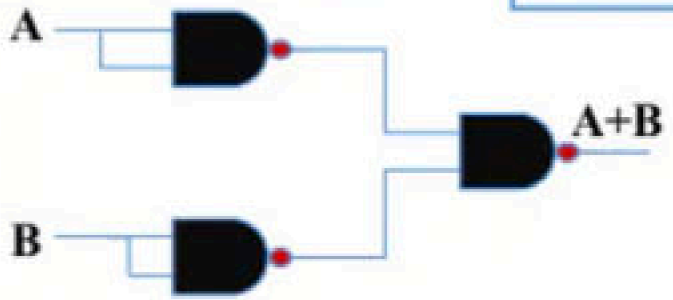
OR GATE

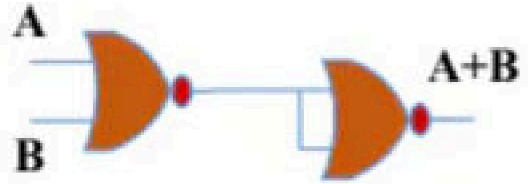
Y =A+B

- Output is '1' if anyone of the inputs are '1'
- Y = A+B = $\mathbf{\Sigma(1,2,3)} = \mathbf{\Pi(0)}$
- Enable input ⟹ 0
- Disable input ⟹ 1
- Commutative law ⟹ Obeys
- Associative law ⟹ Obeys





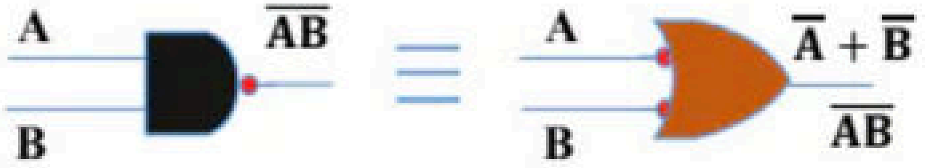


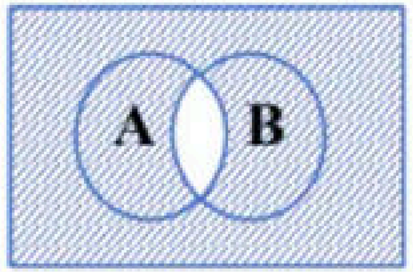


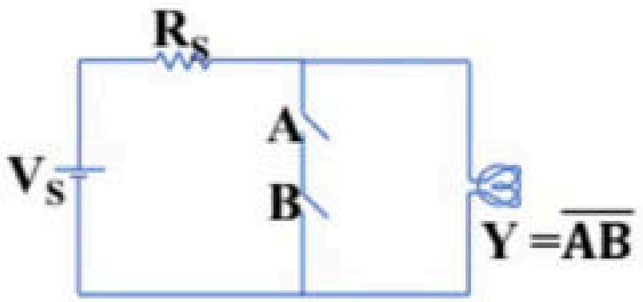
NAND GATE

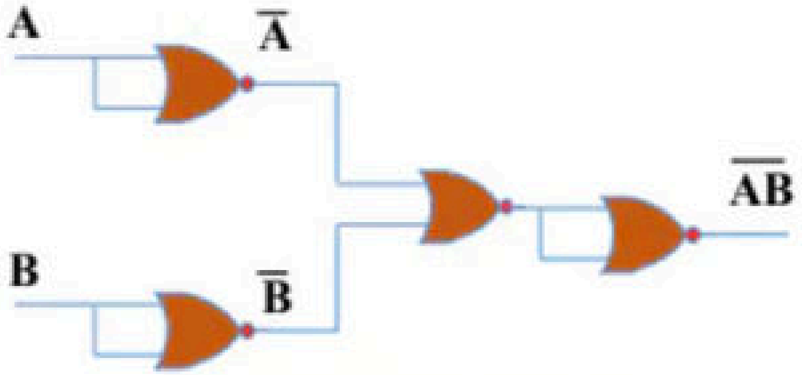
$$Y = \overline{AB}$$

- Output is '1' if any one input is '0'
- $Y = \overline{AB} = \sum(0,1,2) = \prod(3)$
- Enable input --1
- Disable input– 0
- Commutative law ---> Obeys
- Associative law ----> not Obeys





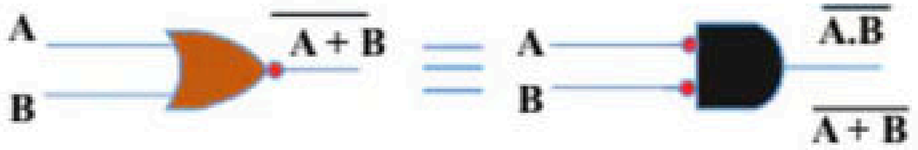


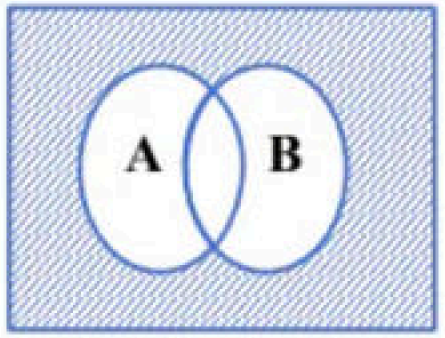


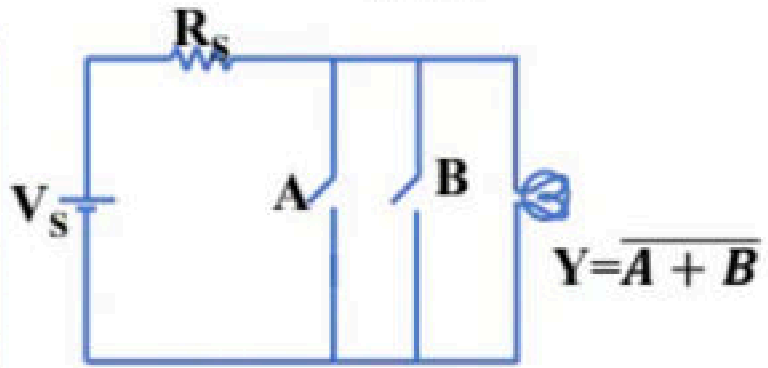
NOR- GATE

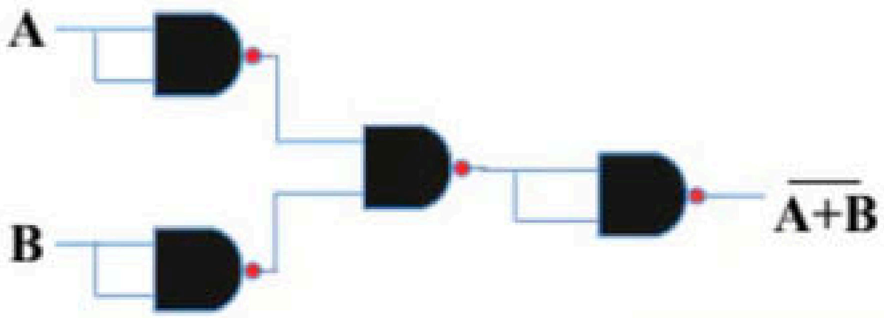
$$Y = \overline{A + B}$$

- Output is '0' if any one of the input is '1'
- $Y = \overline{A + B} = \sum(0) = \prod(1,2,3)$
- Enable input --0
- Disable input– 1
- Commutative law ---> Obeys
- Associative law ----> not Obeys



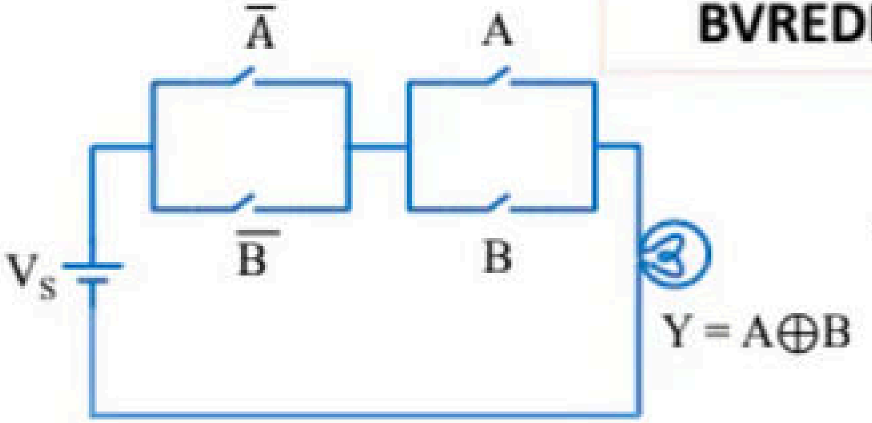


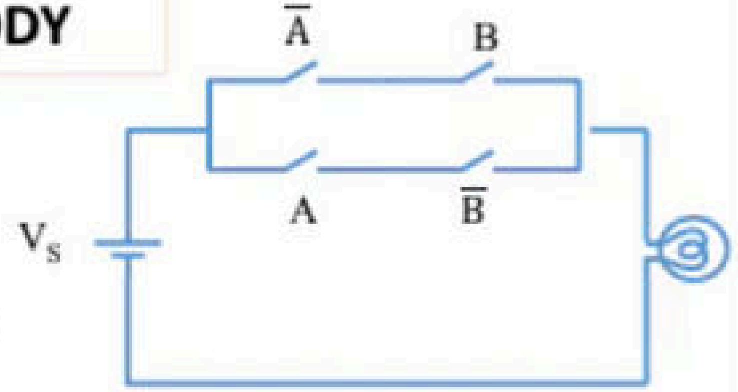


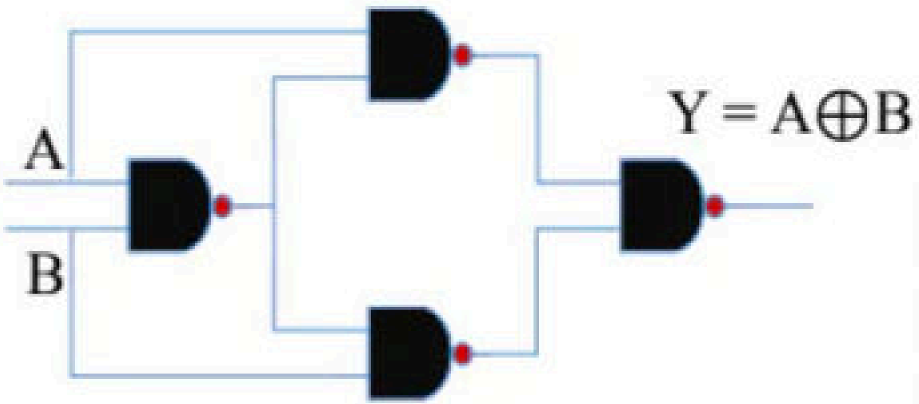


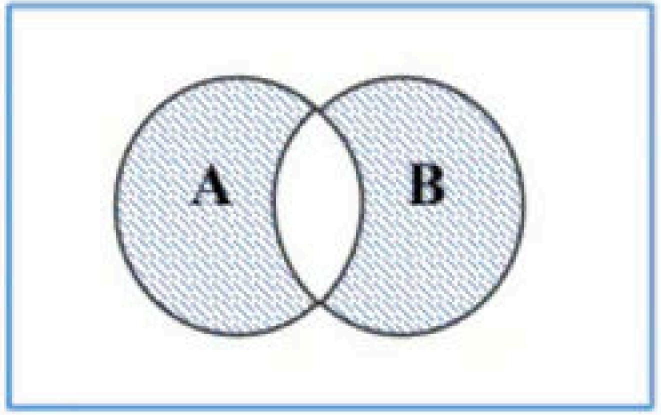
EX-OR GATE

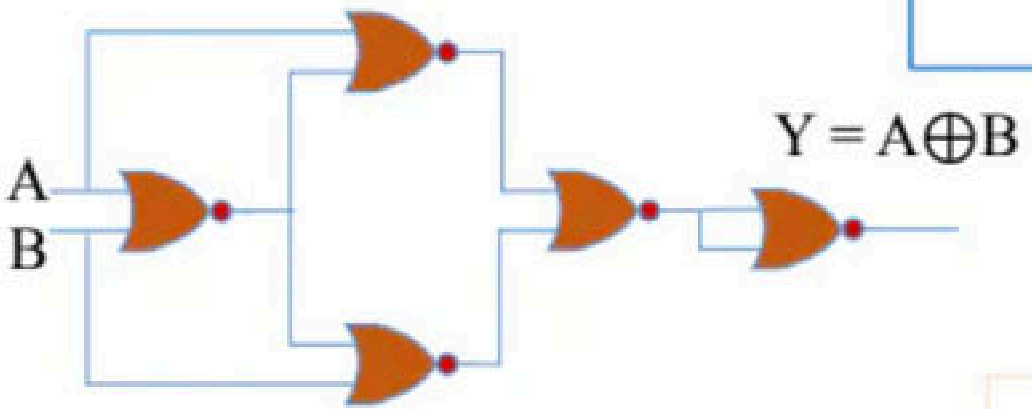
- Output is '1' for odd number of '1's in the input
- $Y = A \oplus B = \sum(1,2) = \Pi(0,3)$
- $Y = A \oplus B \oplus C = \sum(1,2,4,7)$
- $Y = A \oplus B \oplus C \oplus D = \sum(1,2,4,7,8,11,13,14)$
- Commutative law $\Longrightarrow$ Obeys
- Associative law $\Longrightarrow$ Obeys
- $A \oplus 0 = A$
- $A \oplus 1 = \overline{A}$
- $A \oplus A = 0$
- $A \oplus \overline{A} = 1$
- $A \oplus A \oplus A \oplus \ldots\ldots\ldots\ldots n \text{ times} = \begin{cases} A, & n \text{ is odd} \\ 0, & n \text{ is even} \end{cases}$
- $A \oplus \bar{A}B = A + B$
- $AB \oplus BC = B(A \oplus C)$





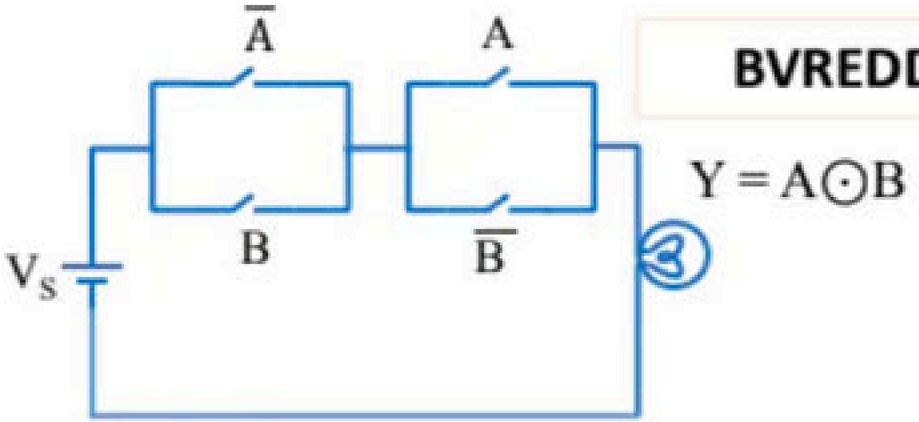


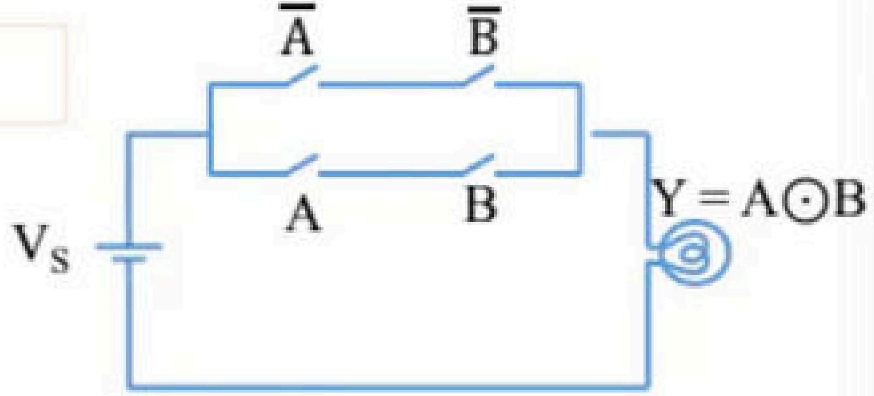


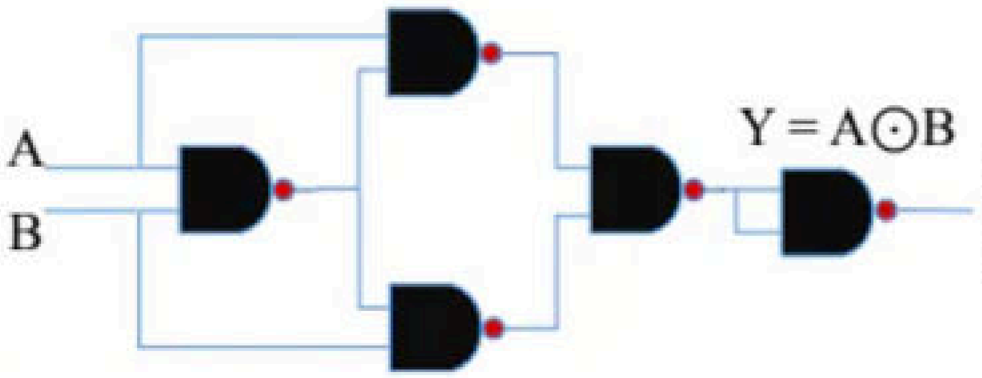


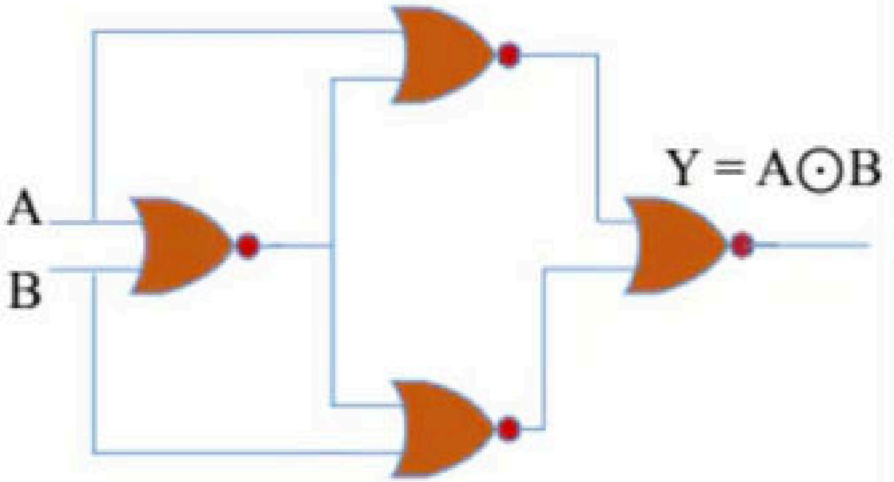
EX-NOR GATE

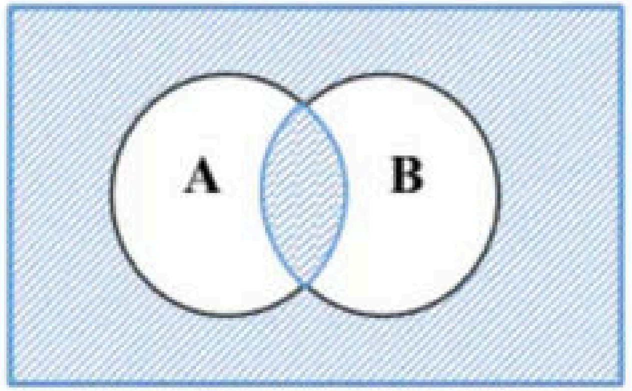
- Output is '1' for even number of '1's in the input
- $Y = A \odot B = \sum(0,3) = \Pi(1,2)$
- Commutative law $\Rightarrow$ Obeys
- Associative law $\Rightarrow$ not Obeys
- $A \odot 0 = \overline{A}$
- $A \odot 1 = A$
- $A \odot A = 1$
- $A \odot \overline{A} = 0$
- $A \odot A \odot A \odot \ldots\ldots\ldots\ldots n \text{ times} = \begin{cases} \overline{A}, & n \text{ is odd} \\ 1, & n \text{ is even} \end{cases}$
- $\overline{A \odot B} = A \oplus B$
- $A \oplus \overline{B} = A \odot B$
- $\overline{A} \oplus B = A \odot B$
- $\overline{A} \oplus \overline{B} = A \oplus B$
- $A \odot B \odot C = \sum(0,3,5,6)$
- $A \oplus B \oplus C = \sum(1,2,4,7)$
- $(A \odot B) \odot C = \sum(1,2,4,7)$
- $(A \odot C) \odot B = \sum(1,2,4,7)$
- $A \oplus B \oplus C = (A \odot B) \odot C = (A \odot C) \odot B$
- $A \odot B = \overline{A} \oplus B = A \oplus \overline{B} = \overline{A} \odot \overline{B}$
- $A \oplus B = A \odot \overline{B} = \overline{A} \odot B = \overline{A} \oplus \overline{B}$
- $\overline{A \oplus B \oplus C} = A \odot B \odot C = [A \oplus B] \odot C = A \odot [B \oplus C]$











EX-OR GATE	EX-NOR GATE
Output is '1' for odd number of '1's in the input	Output is '1' for even number of '1's in the input
Odd number of 1's detector	Even number of 1's detector
Inequality detector	Equality detector
Anti-coincident gate	Coincident gate

	No. of NAND GATES	No . of NOR GATES
NOT	1	1
AND	2	3
OR	3	2
EX-OR	4	5
EX-NOR	5	4
NAND	1	4
NOR	4	1

- For a n- variable Boolean expression , the maximum number of literals = n
- For a n- variable K- Map if group is done by considering 2^m number of cells , then the resulting term from that group contains (n- m) number of literals .
- $8\ cells$ $- 2^3$ cells → Octet --> 3 variables eliminated
- $4\ cells$ $- 2^2$ cells → Quad ---> 2 variables eliminated
- $2\ cells$ $- 2^1$ cells → Pair ---> 1 variables eliminated

- Minimal expression may not be unique .
- The minimal expression = (All EPI's) + (Optional PI's)
- If all PI's are EPI's , then the minimal expression is unique
- The sufficient condition for a K-map to have unique solution is number of PI's = number of EPI's

K- Map

Implicant : Each minterm in canonical SOP expression is known as Implicant .

Prime Implicant is a product term , obtained by combining maximum possible cells in the K- Map. While doing so make sure that a smaller group is not completely inside a bigger group .

Essential Prime Implicant : A prime Implicant is an EPI , if and only if it contains at least one minterm which is not covered by multiple groups

All EPI's are PI's , but vice versa not true

EPI ≤ PI

Minterm mode

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

Maxterm mode

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

Number systems

- Base (b) is always a positive integer .
- In general b ≥ 0

Base	*Different digits*
2 (Binary)	0 , 1
8(Octal)	0,1,2,3,4,5,6,7
10 (Decimal)	0,1,2,3,4,5,6,7 ,8,9
16 (Hexadecimal)	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

r' s Complement

r' s Complement of the number (N) $= r^n - N$

r --------> Radix

n -------> number of integer digits

N --------> given number

(r-1) ' s Complement

(r-1) ' s Complement of the number (N) $= r^n - r^{-m} - N$

r --------> Radix

n -------> number of integer digits

m -------> number of decimal digits

N --------> given number

(r-1) ' s Complement of the number (N) $= r^n - r^{-m} - N$

r' s Complement of the number (N) = (r-1)'s complement + r^{-m}

if m= 0

r' s Complement of the number (N) = (r-1)'s complement + 1

Unsigned Number Representation

- Strictly applicable for positive numbers
- There is no sign bit concept

+ 5 ------> 101

- 5 -----> not allowed

- Range = 0 to $2^n - 1$

Signed Magnitude representation

- Valid for both positive and negative numbers .
- Sign bit concept is used .

Sign bit		Magnitude		

Sign bit = 0 , for ⊕Ve number

= 1, for ⊖ *ve number*

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

1's Compliment representation

In this ⊕Ve numbers are represented as normal binary number with MSB '0'

Representation of ⊖ *ve number*

1. Write the binary equivalent of magnitude
2. Take its 1's compliment

➢ Range = - $(2^{n-1} - 1)$ to $+(2^{n-1} - 1)$

Overflow

Over flow occurs in signed arithmetic operations if two same sign numbers are added and result exceeds with given number of bits . Overflow can be avoided by taking extra bits

1.By using carry bits

$$C_{in} \text{--------}\rightarrow \text{carry into MSB}$$
$$C_{out} \text{----}\rightarrow \text{carry out from MSB}$$

if $C_{in} \oplus C_{out} = 0$, no overflow occurs

$C_{in} \oplus C_{out} = 1$, over flow occurs

2. By using Sign Bits

X -----------> Sign bit of 1st number

Y -----------> Sign bit of 2nd number

Z-----------> Sign bit of Resultant

Over flow = $XY\overline{Z} + \overline{X}\overline{Y}Z$

2's Compliment representation

In this ⊕Ve numbers are represented as normal binary number with MSB '0'

Representation of ⊖ *ve number*

1. Write the binary equivalent of magnitude
2. Take its 2's compliment

➢ Range = - (2^{n-1}) to $+(2^{n-1} - 1)$

BCD (Binary Coded Decimal)Code

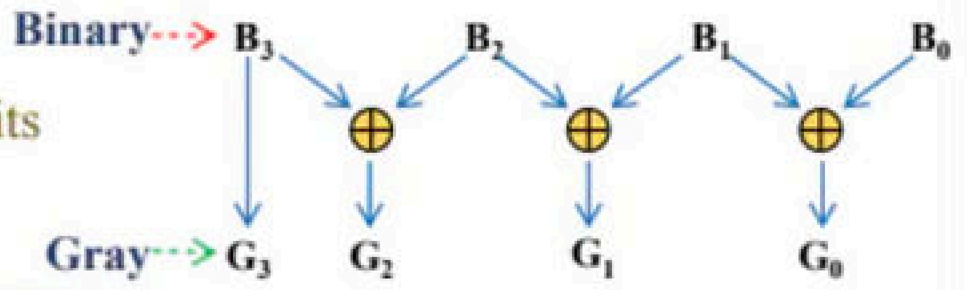
In this code each decimal number is represented by a separate group of 4- bits

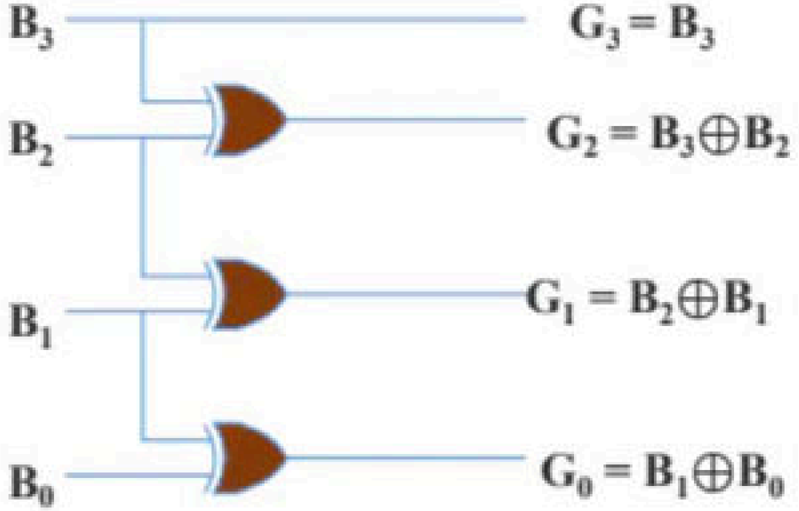
➢ It uses only 0 to 9

➢ 0 to 9 are valid BCD Code

➢ 10, 11, 12 , 13 , 14 ,15 are invalid BCD Code

➢ Coding method is very simple but it requires more number of bits .





EX-3 Code

The EX-3 code can be derived from the natural BCD code by adding 3 to each coded number

Valid EX -3 : 3 ,4,5,6,7,8,9,10,11,12

Invalid EX-3 : 0,1,2,13,14,15

Gray Code

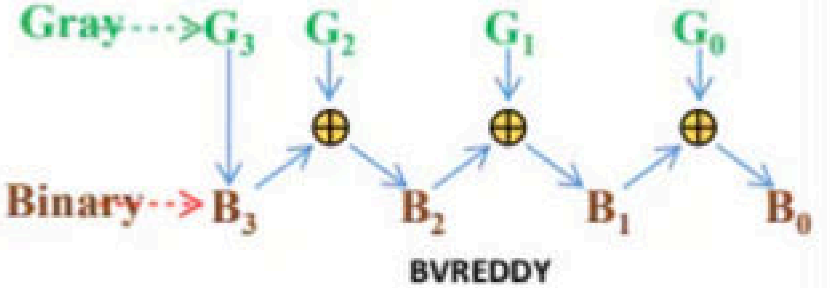
➢ Non weighted code

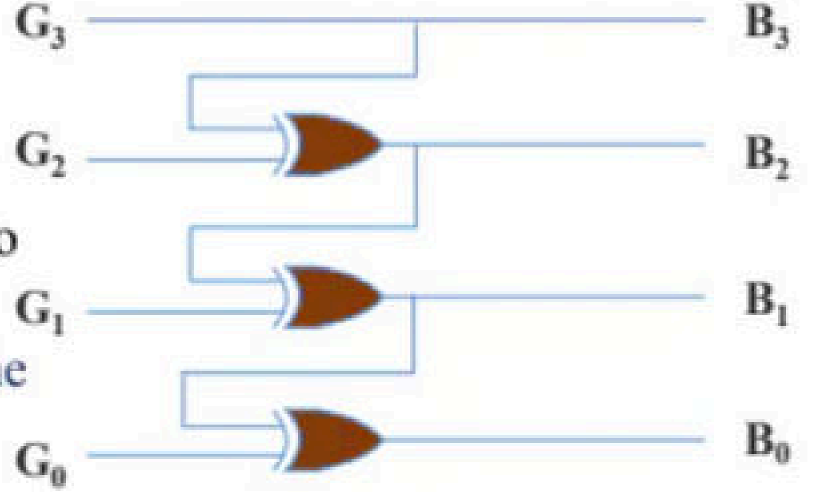
➢ Unit distance code

➢ Cyclic code

➢ Reflective code

➢ Minimum error code





SELF COMPLEMENTING CODE

A code is said to be self complementing, if the 1' complement of a number N is equal to the 9's complement of the number.

➢ For a code to be self complementing, the sum of all its weights must be 9 .

2421	5211	4311	3321	EX-3

BVREDDY

HA

1. Logical expression for Sum = $A \oplus B$
2. Logical expression for Carry = **AB**
3. Minimum number of NAND Gates = **5**
4. Minimum number of NOR Gates = **5**

FA

1. Logical expression for Sum = $A \oplus B \oplus C$
2. Logical expression for Carry =$AB+(A \oplus B)C$
3. Minimum number of NAND Gates = 9
4. Minimum number of NOR Gates = 9

HS

1. Logical expression for Difference = $A \oplus B$
2. Logical expression for Barrow=$\bar{A}B$
3. Minimum number of NAND Gates = **5**
4. Minimum number of NOR Gates =**5**

FS

1. Logical expression for Difference= $A \oplus B \oplus C$
2. Logical expression for Barrow = $\bar{A}B + (\overline{A \oplus B})C$
3. Minimum number of NAND Gates = 9
4. Minimum number of NOR Gates = **9**

Half Adder

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

Half Subtractor

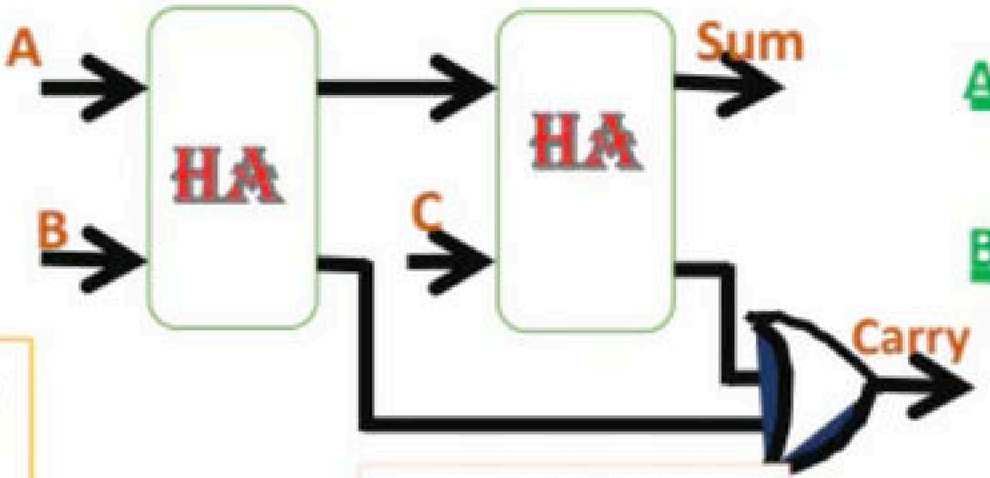
A	B	Difference	Barrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full Adder

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

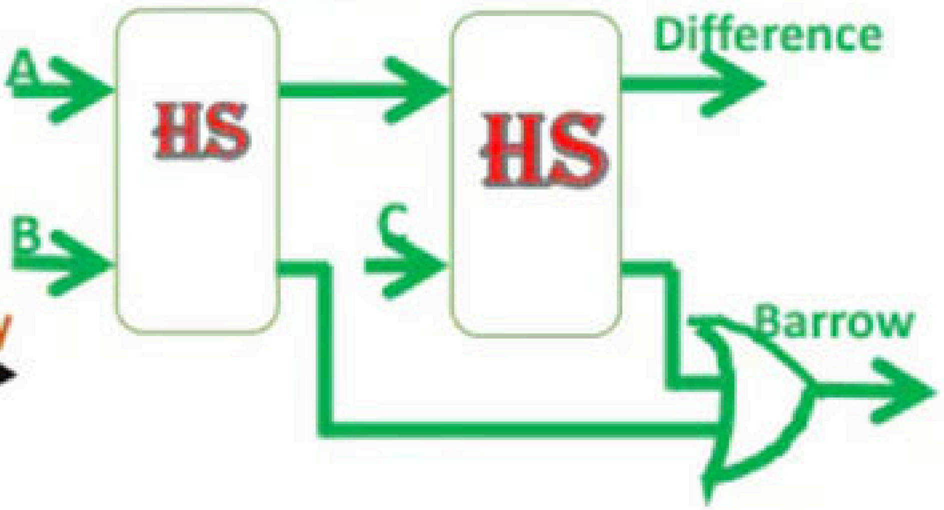
Full Subtractor

A	B	C	Difference	Barrow
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



BVREDDY

BVREDDY



Use the Code : BVREDDY

FS : A- B- C

Difference = $A \oplus B \oplus C$

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

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

FS : B- C- A

Difference = $A \oplus B \oplus C$

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

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

FS : C- A- B

Difference = $A \oplus B \oplus C$

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

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

Binary Multiplier

Number of AND gates required = $m \times n$

Number of Adders required = m+n-2

m-----------> number of bits in A

n ------------>number of bits in B

In general for n- bit Parallel Adder

Worst case Delay = (n-1)$\left((t_{pd})\, carry\right) + Max(sum\ , carry)$

For n- bit Magnitude Comparator

Total number of input combinations = 2^{2n}

Lesser than combinations = $\frac{2^{2n} - 2^n}{2}$

Greater than combinations = $\frac{2^{2n} - 2^n}{2}$

Equal combinations = 2^n

Look Ahead Carry Adder

BVREDDY

- In this adder ,the carry dependency of Ripple Carry Adder (RCA) is eliminated
- This is the fastest adder among all
- This adder have the maximum complexity

Hardware Requirements

$$\left.\begin{array}{l}\left.\begin{array}{ll} L1: & \text{n- XOR + n- AND} \\ L2: & \frac{n(n+1)}{2} - AND \\ L3: & \text{n- OR} \end{array}\right\} \text{carry} \\ \begin{array}{ll} L4: & \text{n- XOR} \end{array}\end{array}\right\} \text{sum}$$

Total number of gates for carry = $3n + \frac{n(n+1)}{2}$

Total number of gates for sum = $4n + \frac{n(n+1)}{2}$

For 3- bit magnitude comparator

BVREDDY

$Y_1(A<B) = \bar{a}_2 b_2 + (a_2 \odot b_2)\bar{a}_1 b_1 + (a_2 \odot b_2)(a_1 \odot b_1)\bar{a}_0 b_0$

$Y_2(A=B) = (a_2 \odot b_2)(a_1 \odot b_1)(a_0 \odot b_0)$

$Y_3(A>B) = a_2 \bar{b}_2 + (a_2 \odot b_2)a_1 \bar{b}_1 + (a_2 \odot b_2)(a_1 \odot b_1)a_0 \bar{b}_0$

For 4-bit Magnitude Comparator

$Y_1(A<B) = \overline{a_3}\, b_3 + (a_3 \odot b_3)(\overline{a_2}\, b_2) + (a_3 \odot b_3)(a_2 \odot b_2)(\overline{a_1}\, b_1) + (a_3 \odot b_3)(a_2 \odot b_2)(a_1 \odot b_1)(\overline{a_0}\, b_0)$

$Y_2(A=B) = (a_3 \odot b_3)\ (a_2 \odot b_2)\ (a_1 \odot b_1)\ (a_0 \odot b_0)$

$Y_3(A>B) = a_3 \overline{b_3} + (a_3 \odot b_3)(a_2\, \overline{b_2}) + (a_3 \odot b_3)(a_2 \odot b_2)(a_1 \overline{b_1}) + (a_3 \odot b_3)(a_2 \odot b_2)\ (a_1 \odot b_1)(a_0\, \overline{b_0})$

Worst delay for Carry = Max(xor ,and)+ $(t_{pd})_{and} + (t_{pd})_{or}$

Worst case delay for Sum = Max(xor ,and)+ $(t_{pd})_{and} + (t_{pd})_{or} + (t_{pd})_{xor}$

BVREDDY

Multiplexer (MUX)

BVREDDY

- Data selector
- Many to one
- Universal logic gate
- Parallel to serial converter

$2^n \times 1$

2^n ----------> number of data inputs

n -----------> number of select inputs

1 ------------> number of outputs

Demultiplexer

- One input to many output
- Data distributor
- One to many circuit

1×2^n

n---------> number of select lines

2^n --------> number of output lines

1 ----------> number of inputs

Decoder

Decoder is a multi input ,multi output logic circuit which coverts coded input into coded output , where the input and output codes are different

$n \times 2^n$

n ------------> number of inputs

2^n ------------> number of outputs

Decoder is a special case of Demultiplexer , in which the select lines or Demultiplexer are treated as input's to the decoder and input of Demultiplexer is treated as Enable input of the Decoder

Inputs ⟺ **Enable**

Select lines ⟺ **Inputs**

BVREDDY

Logic Gate	Number of MUX required
BUFFER	1
NOT	1
AND	1
OR	1
NAND	2
NOR	2
EX-OR	2
EX-NOR	2
HA	3
HS	2

Encoder

Encoder is a combinational circuit , which is used to convert

1. Octal to binary (8 × 3 encoder)
2. Decimal to Binary (10× 4 encoder)
3. Hexadecimal to Binary (16 × 4 encoder)

$2^n X\, n$

n ------------> number of outputs

2^n ------------> number of inputs

- For an Encoder at a time only one among the all inputs is high , reaming all inputs should be zero
- If multiple inputs are simultaneously high, then the output is not valid, to avoid this restriction we will go for priority encoder.

1. By using one 4 × 1 Mux → All 2 variable functions; Some but not All 3 variable functions

BVREDDY

2. By using one 4 × 1 Mux + NOT Gate → All 2 variable functions; All 3 variable functions
3. By using one 8 × 1 Mux → All 3 variable functions; Some but not All 4 variable functions
4. By using one 8 × 1 Mux + NOT Gate → All 3 variable functions; All 4 variable functions
5. n- variable function → **One 2^n × 1 MUX**; **One 2^{n-1} × 1 MUX + one NOT Gate**

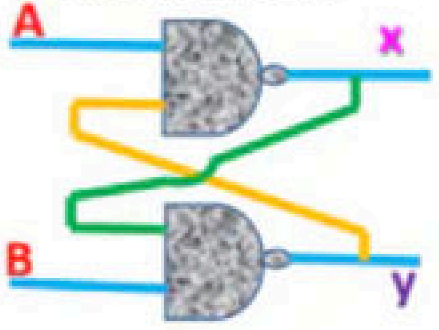
BVREDDY

Sequential Circuits

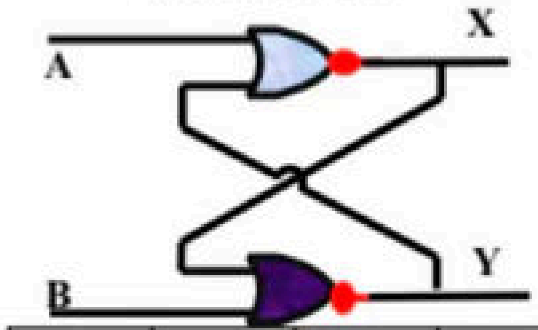
The logic circuit whose outputs at any instant of time depends on the present inputs as well as on the past outputs are called sequential circuits, in sequential circuits ,the output signals are fed back to the input side .

- Out put of combinational circuit depends on input combinations .
- Output of sequential circuits depends on input sequence.
- For unequal delay of gates also the operation is valid

NAND LATCH



NOR LATCH



A	B	X	Y
0	0	1	1
0	1	0	1
1	0	1	0
1	1	Memory	

A	B	X	Y
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Memory	

For **SR NAND** latch , if the input sequence is **00** ---------> **11** , then the following cases arises

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as critical race
- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates

For **SR NOR** latch , if the input sequence is **11** ---------> **00** , then the following cases arises

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as critical race .
- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates .

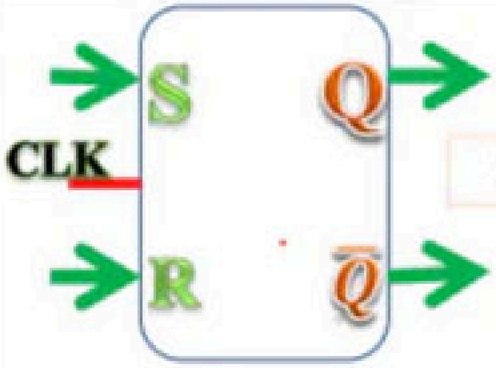
FLIP FLOP

In a latch the output changes immediately in response to external input , so to have an additional control , we are introducing a signal called " ***CLOCK*** " , whose purpose is same as Enable pin of Decoder.

Latch +Clock = Flip Flop

Latches are universally not unique and hence their truth tables are not unique .

Flip Flops are universally unique , and their truth tables are unique .



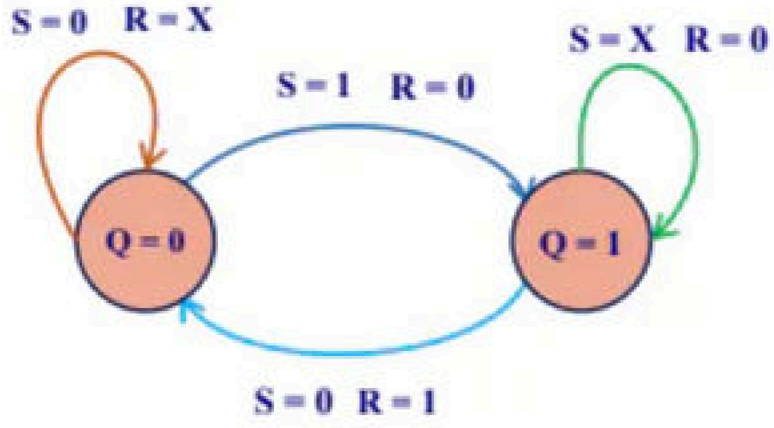
BVREDDY

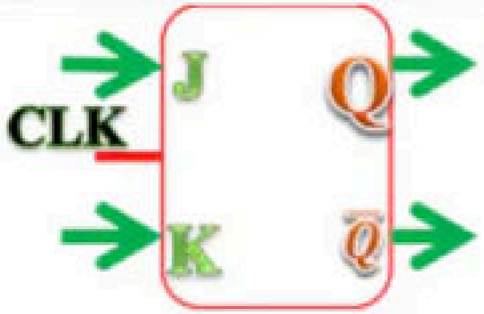
$$Q^{+} = S + \bar{R}Q$$

CLK	S	R	Q+	State
0	×	×	Q	Memory
1	0	0	Q	Memory
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	×	Invalid

CLK	S	R	Q	Q+
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	×
1	1	1	1	×

Q	Q+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



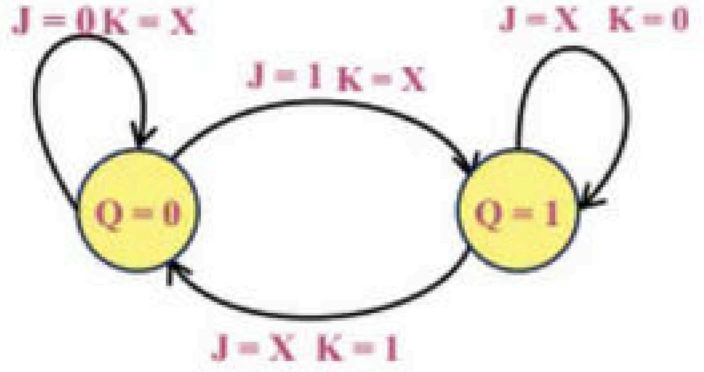


$$Q^{+} = J\bar{Q} + \bar{k}Q$$

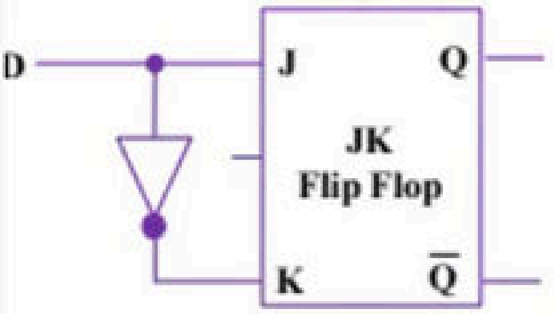
CLK	J	K	Q+	State
0	×	×	Q	Memory
1	0	0	Q	Memory
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	$\bar{Q}$	Toggle

CLK	J	K	Q	Q+
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



D Flip Flop

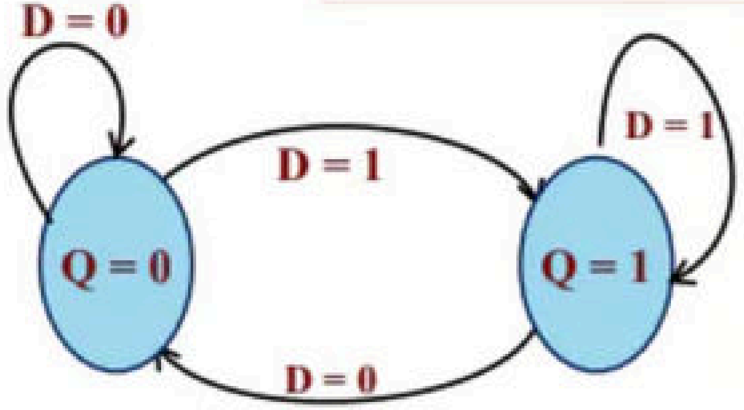


$$Q^{+} = D$$

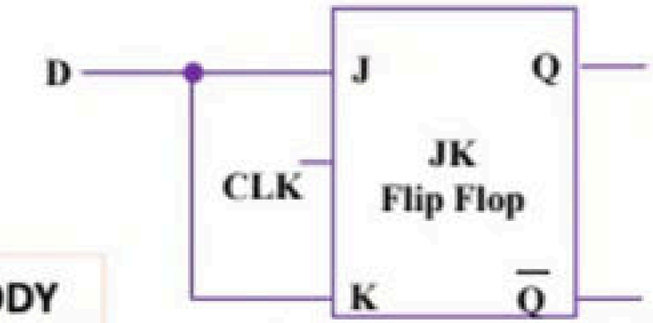
CLK	D	Q+
0	X	Hold
1	0	0
1	1	1

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1

CLK	D	Q	Q+
0	X	Q	Q
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



T Flip Flop

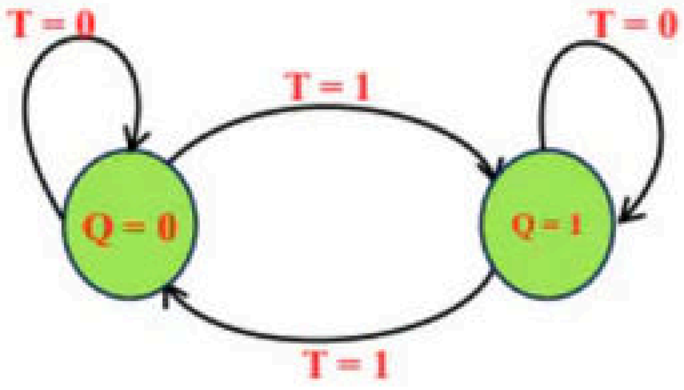


$$Q^{+} = T \oplus Q$$

CLK	T	Q+
0	X	Hold
1	0	Q
1	1	Toggle

CLK	T	Q	Q+
0	X	Q	Q
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0



Race Around Condition

The output of the FF changes to 0 →1 →0 …. Continuously at the starting of the next clock the output is uncertain , which is called as Race Around Condition (RAC)

RAC occurs in any FF if the following conditions satisfies

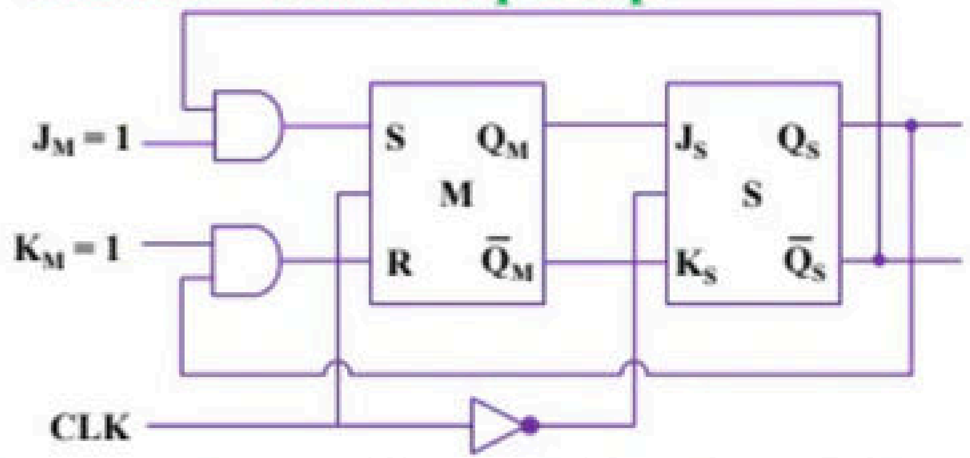
1. If the FFs are operated in level triggering
2. if (tpd) < (Tclk)on ,
3. If the FFs are operated in Toggle mode

If the above 3 conditions satisfies simultaneously then there is a continuous race in the output of the FF between 0 and 1 to reach the next state , who will be the winner of the race in not certain , that depends on tpd and (Tclk) on .

Remedy

1. (Tclk)on < (tpd) < T
2. By using Edge triggered FF
3. By using Master Slave FF

Master – Slave Flip Flop



1. In case of Master Slave configuration , Master is applied with input clock and Slave is applied with inverted clock , so out of two FFs at a time only one of the FF respond and other will not respond . As a result, Many times toggling in a single clock cycle has been converted to one time toggle , hence *RAC is avoided* .
2. In Master Slave configuration , command signal is generated by master FF and the response of the command signal is given by slave FF
3. Master slave FF can store 1 – bit of data

CONVERSATION of FLIP FLOP

JK to SR

$J = S$

$K = R$

SR to JK

$S = J\overline{Q}$

$R = KQ$

D to SR

$D = S+\overline{R}Q$

T to SR

$T = S\overline{Q}+RQ$

JK to D

$J = D$

$K = \overline{D}$

SR to D

$S = D$

$R = \overline{D}$

D to JK

$D = J\overline{Q}+\overline{K}Q$

T to JK

$T = J\overline{Q}+KQ$

JK to T

$J = T$

$K = T$

SR to T

$S = T\overline{Q}$

$R = TQ$

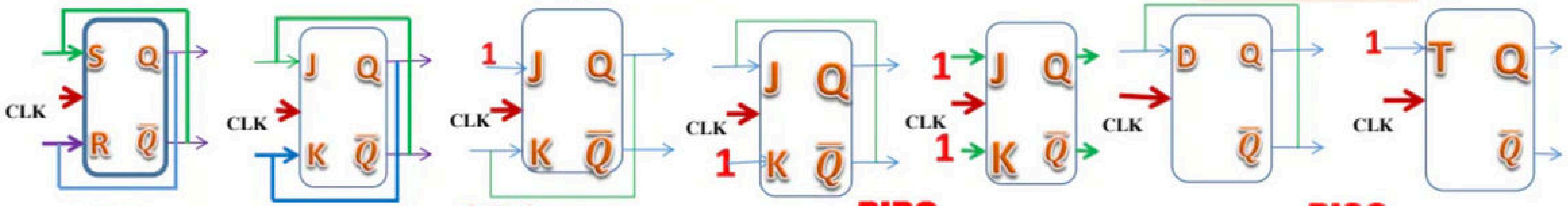
D to T

$D = T \oplus Q$

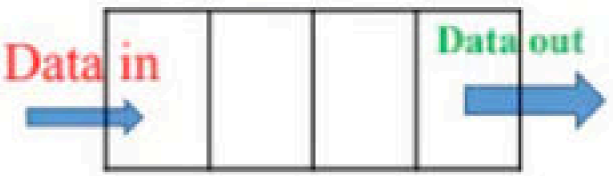
T to D

$T = D \oplus Q$

Toggle Modes



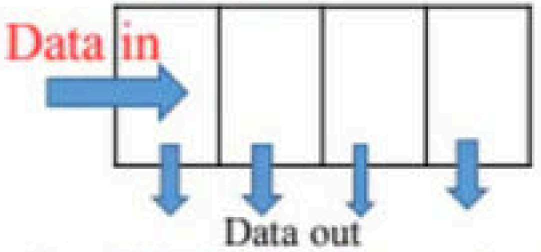
SISO



- SISO Configuration has only
  1- input
  1- output
- For SISO configuration

for storing = (n) CP
for retrieving = (n-1) CP
Total number clock pulses = 2n-1

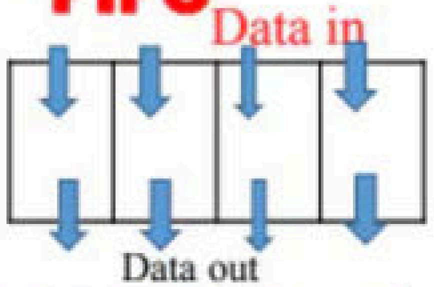
SIPO



- SIPO Configuration has only
  1- input
  4- output
- For SIPO configuration

for storing = (n) CP
for retrieving = 0 CP
Total number clock pulses = n

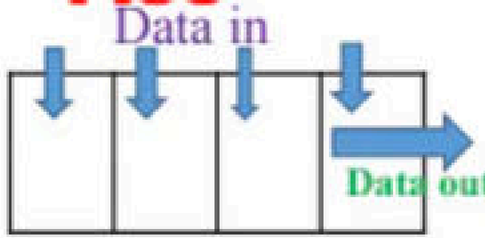
PIPO



- PIPO Configuration has only
  4- input
  4- output
- For PIPO configuration

for storing = 1 CP
for retrieving = 0 CP
Total number clock pulses = 1

PISO



- PISO Configuration has only
  4- input
  1- output
- For PISO configuration

for storing = 1 CP
for retrieving = (n-1)CP
Total number clock pulses = n

Counters

State of a Counter : Any possible output of a counter is known as its state , for a n – bit counter the maximum possible states are 2^n

The states which are counted by the counter are called as *valid states* , and the states which are not counted (skipped) by the counter are called as invalid states .

Modulus of a Counter : The minimum number of clocks needed to get the counting pattern repeats is called as Modulus of a counter

Design equation of a counter

$$2^n \geq N$$
$$n \geq \log_2 N$$

n----> number of Flip Flops
N-----> MOD no. of a counter

ASYNCHRONOUS COUNTER

BVREDDY

- Different FFs are applied with different clocks
- For only one FF external clock is applied ,which is LSB and output of one FF will acts as clock to next FFs
- FFs are operated in toggle mode
- Fixed counting sequence
  1. up counter
  2. down counter

ये वक्त भी गुजर जाएगा

This time will also pass

- ⊖ve Edge trigger and Q as a clock --------> Up counter
- ⊖ve Edge trigger and $\bar{Q}$ as a clock --------> Down counter
- ⊕ve Edge trigger and Q as a clock --------> Down counter
- ⊕ve Edge trigger and $\bar{Q}$as a clock --------> Up counter
- The disadvantages of the ripple counter is that transition states are present due to delay of the FF (Decoding errors) .
- If only one FF changes its state ,then no transition states will be present , if more than one FF changes its states than transition states present.

BVREDDY

- To avoid decoding errors strobe signal is used .
- Strobe signal is kept low for 3tpd , for 3- bit counter , so that transition states are not reflected, and after 3tpd strobe signal is made high .
- If delay each FF is t_{pd} , then

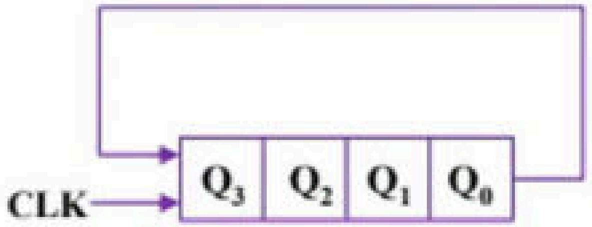
$$T_{CLK} \geq n\, t_{pd}$$

$$f_{CLK} \leq \frac{1}{t_{pd}}$$

RING COUNTER

- Ring counter is a synchronous counter , it is a shift register in which last FF output is connected to the first FF input .
- In ring counter only one FF output is logic '1 ' and it will rotate with clock .
- Ring counter performs right shift operation .

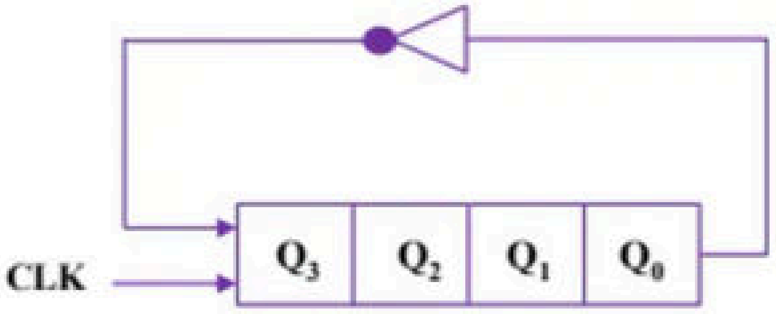
BVREDDY



- Decoding logic of ring counter is simple and does not require any external logic circuit
- If all the outputs of FFs initially zero , then the Ring counter does not start .
- If more than one FF outputs' are high initially, then the ring counter enters into unused state and never come out of unused state , this is called as **Lock out problem** .

JOHNSON RING COUNTER

BVREDDY



Johnson Ring counter

Twisted Ring counter

Switch tail counter

Walking Counter

Creeping counter

Mobies counter

Ring counter	Johnson ring counter
1. Mod No = n	1. Mod No = 2n
2. Number of used states= n Number of unused states = $2^n - n$	2. Number of used states= 2n Number of unused states =$2^{2n} - n$
3.Time period of each FF = n(T_{CLK})	3.Time period of each FF = 2n(T_{CLK})
4. Frequency of each FF = $\frac{f_{clk}}{n}$	4. Frequency of each FF = $\frac{f_{clk}}{2n}$
5. Suffer from lock out problem	5. Suffer from lock out problem
6.Decoding logic is simple	6.Decoding logic requires AND and NOR gates

FINITE STATE MACHINE

Synchronous Sequential circuits are also called as Finite State Machine (FSM)

There are two types of FSMs

1. Mealy State Machine

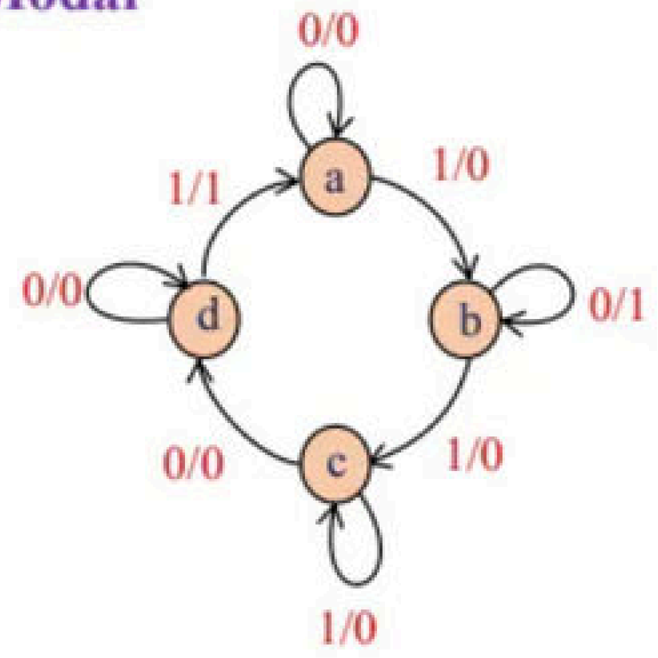
- The output of Mealy State Machine is a function of present state as well as present input
- to detect n – bit sequence by using Mealy modal n number of states are required

2. Moore State Machine

- The output of Moore State Machine is a function of present state only
- To detect **n – bit sequence** by using Moore modal **(n+1)** number of states are required

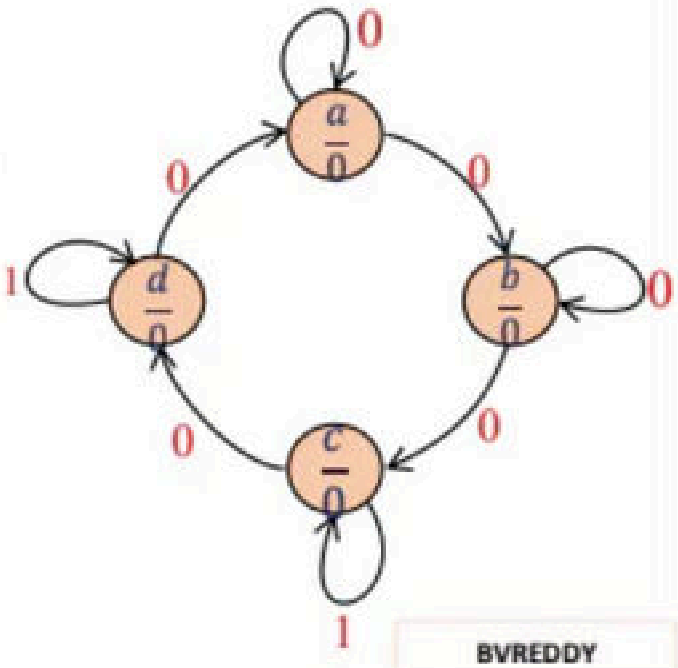
Mealy Modal

Present state	NS , O/P	
	X =0	X= 1
a	a , 0	b , 0
b	b, 1	c,0
c	d, 0	c , 0
d	d, 0	a , 1



Moore Modal

Present state	Next State		Output
	X =0	X=1	
a	a	b	0
b	b	c	0
c	d	c	0
d	a	d	1



+6 =	0	1	1	0

- 6 =	1	0	1	0

+6 =	0	0	0	0	0	1	1	0

- 6 =	1	1	1	1	1	0	1	0

Decimal number	2's complement Representation
+0	
+1	
+2	
+3	
+4	
+5	
+6	
+7	
-0	
-1	
-2	
-3	
-4	
-5	
-6	
-7	
-8	

Range with 4 bits =

Range with 5- bits =

Range with n- bits =

Q) Find the Decimal equivalent of the **unsigned number representation** given below

a) 01101

b) 11101

Q) Find the Decimal equivalent of the **Signed magnitude representation** given below

a) 01101 b) 11101

Q) Find the Decimal equivalent of the **1's complement representation** given below

a) 01101 b) 11101

Q) Find the Decimal equivalent of the **2's complement representation** given below

a) 01101

b) 11101

Q) **1's complement representation** of the numbers given below , find the decimal equivalents

a) 01001

b)001001

c)0001001

d)11001

e)111001

f)1111001

Q) **2's complement representation** of the numbers given below , find the decimal equivalents

a) 01001

b)001001

c)0001001

d)11001

e)111001

f)1111001

Q) Find the **2's complement representation** of the following

$-2 =$

$-4=$

$-8=$

$-16=$

$-2^{n}=$

Note :

The minimum number of bits required for - 2^n , using 2's complement representation = ---------

Q) A number in 4- bit 2's complement is $x_3x_2x_1x_0$,this number when stored using 8- bits will be

a) 0000 $x_3x_2x_1x_0$
b)1111 $x_3x_2x_1x_0$
c) $x_3x_3x_3x_3x_3x_2x_1x_0$
d) $\overline{x_3}\ \ \overline{x_2}\ \ \overline{x_1}\ \ \overline{x_0}\ x_3x_2x_1x_0$

Q. How many one's are present in the binary representation of $(8 \times 4096) + (4 \times 256) + (9 \times 16) + 5$

(a) 6 (b) 5 (c) 3 (d) 4

Binary Subtraction using 1's complement

1. Represent the given numbers in the 1's complement form.
2. Add the two numbers.
3. If carry is generated ,then the result is positive and in the true form , add carry to the LSB to get the final answer .
4. If carry is not generated , then the result is negative , and in the 1's complement form . To get final answer take 1's complement of the result.

Q)Perform the following operation for the given numbers using 1's complement form .

a) 8 - 4

b)4 - 8

Binary Subtraction using 2's complement

1. Represent the given numbers in the 2's complement form
2. Add the two numbers
3. If carry is generated ,ignore it .
4. If MSB is 0, then the result is positive and in the true form .
5. If MSB is 1, then the result is negative and is in 2's complement form .
   (whether there is a carry (or) no carry does not matter)

Q. Perform the following operation for the given numbers using 2’s complement form .

46 - 14

Q. Perform the following operation for the given numbers using 2's complement form .

$$-75+26$$

Q) Simplify the following using 2's complement form

$9 + 4$

Q) Simplify the following using 2’s complement form

9- 4

Q) Simplify the following using 2’s complement form

$-9 + 4$

Q) Simplify the following using 2's complement form

$-9 - 4$

Q) Simplify the following using 2’s complement form

$$9 + 8$$

Q) Simplify the following using 2's complement form

$-9 - 8$

Overflow

Over flow occurs in signed arithmetic operations if two same sign numbers are added and result exceeds with given number of bits.

Over flow can be detected by using 2- methods

1. by using carry bits

2. by using sign bit

1. By using carry bits

$$C_{in} \dashrightarrow carry\ into\ MSB$$

$$C_{out} \dashrightarrow carry\ out\ from\ MSB$$

if $\quad C_{in} \oplus C_{out} =$

$C_{in} \oplus C_{out} =$

2. By using sign bit

X ----------> Sign bit of 1st number

Y -----------> Sign bit of 2nd number

Z-----------> Sign bit of Resultant

Over flow =

NOTE :

to avoid the overflow , increase the number of bits .

Q) Let x be the sign bit of N_1, y be the sign bit of N_2, and z be the sign bit of $N_1 + N_2$, then the condition for overflow .

a) $x \neq y \neq z$ b) $x \neq y = z$

c) $x = y \neq z$ d) $x = y = z$

Q. Let R1 and R2 be two 4-bit registers that store number in 2's complement form , for operation R1 + R2 , which of the following values of R1 and R2 gives overflow.

a) R1= 1100 , R2 = 1010

b) R1= 1001 , R2 = 1111

c) R1= 1011 , R2 = 1110

d) R1= 0011 , R2 = 0100

Q) Two numbers represented in signed 2's complement form as P = 11101101 , Q = 11100110 , if Q is subtracted from P , then the value obtained in signed 2's complement from is

a) 100000111 b) 00000111 c) 11111001 d)111111001

BINARY CODES

Numeric Codes

1. BCD Code
2. Excess-3 Code
3. Gray Code
4. Self-complementing code

1. BCD (Binary Coded Decimal) Code :

In this code each decimal number is represented by a separate group of 4- bits.

$(2\ 3\ 4\ 5\ )_{10} =$

- It uses only 0 to 9
- 0 to 9 are valid BCD Code
- 10, 11, 12,13,14,15 are invalid BCD Code
- Coding method is very simple but it requires more number of bits .

Eg. of BCD Codes

8 4 2 1
2 4 2 1
3 3 2 1
4 2 2 1
5 2 1 1
5 3 1 1
5 4 2 1
6 3 1 1
7 4 2 1
7 4 $\bar{\mathbf{2}}$ $\bar{\mathbf{1}}$
8 4 $\bar{\mathbf{2}}$ $\bar{\mathbf{1}}$

BCD Addition

1. Express the given numbers in BCD form

2. Add the corresponding digits of the decimal numbers of each group .

3. If there is no carry and the sum term is valid code , no correction is needed

4. If there is a carry out of one group to next group , (or) if the sum term is an invalid BCD code , then add 6_{10} (0110) to the sum term of that group and the resulting carry is added to the next group .

Q) Perform the following using BCD addition

$$25+13$$

Q) Perform the following using BCD addition

679.6+536.8

Q. When two BCD numbers are added, under what conditions a correction factor of 6 is added to a 4-bit nibble

a) When the nibble value is one of 1010, 1011, 1100, 1101, 1110, or 111
b) When there is a carry out of the nibble to the next higher significant nibble
c) When a final carry is generated
d) When the nibble value is one of 0001, 0010, 0100, 1000,

BCD Subtraction

1. Express the given numbers in BCD form
2. Subtract the corresponding digits of the decimal numbers of each group .
3. If there is no barrow no correction is needed.
4. If there is a barrow from the next group ,or if the difference term is an invalid BCD code then 6_{10} (0110) is subtracted from the difference term of that group .

Q) Perform the following using BCD subtraction

38-15

Q) Perform the following using BCD subtraction

206.7-147.8

EXCESS-3 CODE

The EX-3 code can be derived from the natural BCD code by adding 3 to each coded number.

Valid EX -3 :

Invalid EX-3 :

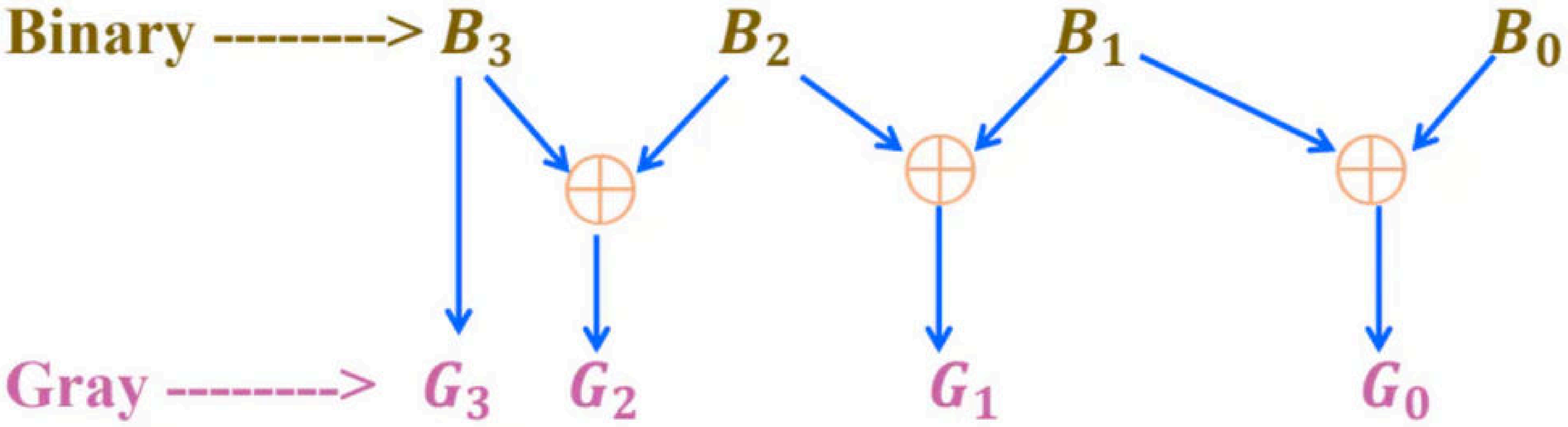
Gray Code

Gray code is a non-weighted code, successive decimal numbers are differ by only one bit .

- Non- weighted code
- Unit distance code
- Cyclic code
- Reflective code
- Minimum distance code

Decimal	1- bit Gray code	2- bit Gray code	3- bit Gray code
0			
1			
2			
3			
4			
5			
6			
7			

Binary to Gray Code



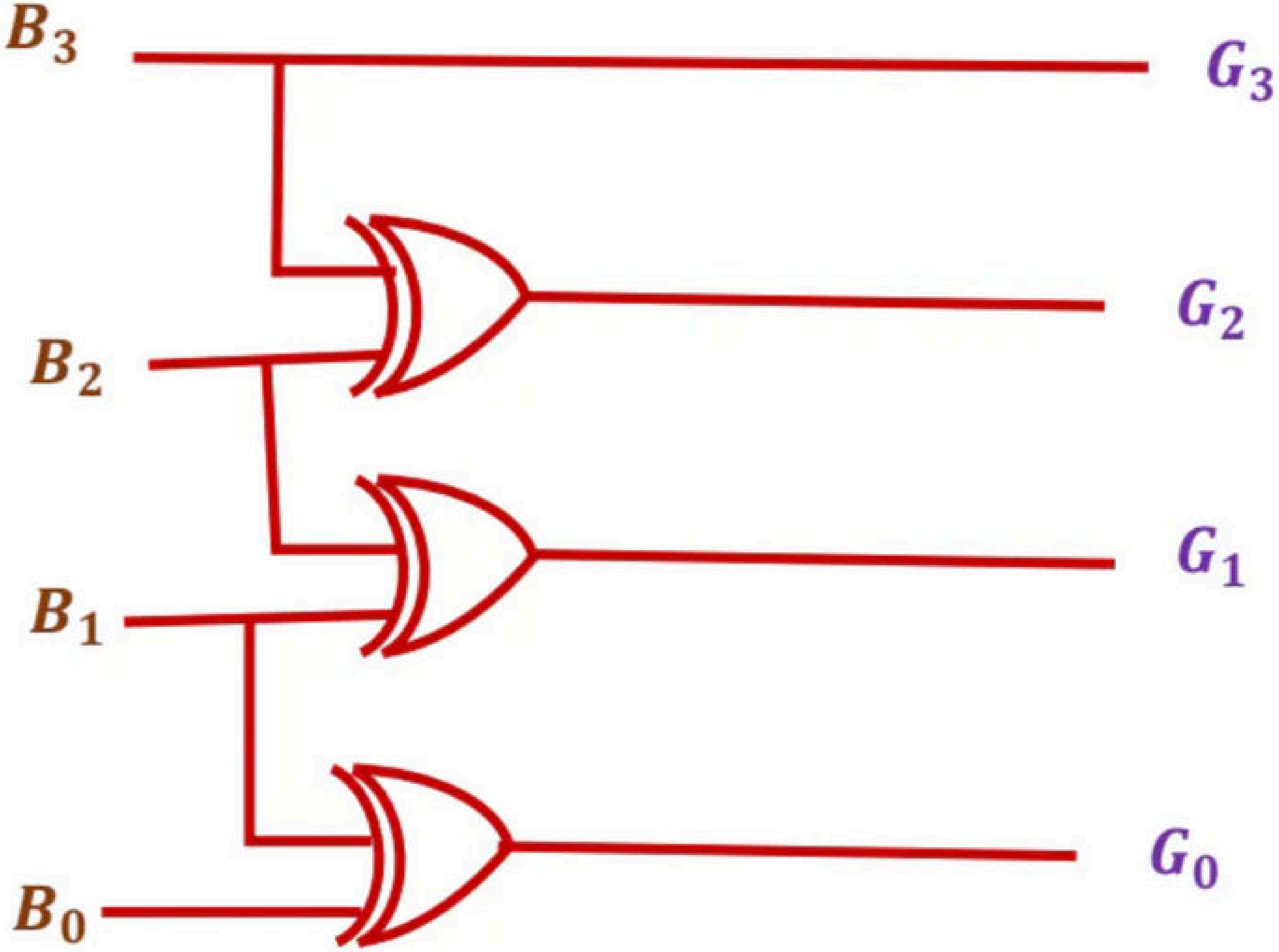
Q) Find the Gray code of the following

1 1 0 0 1 0

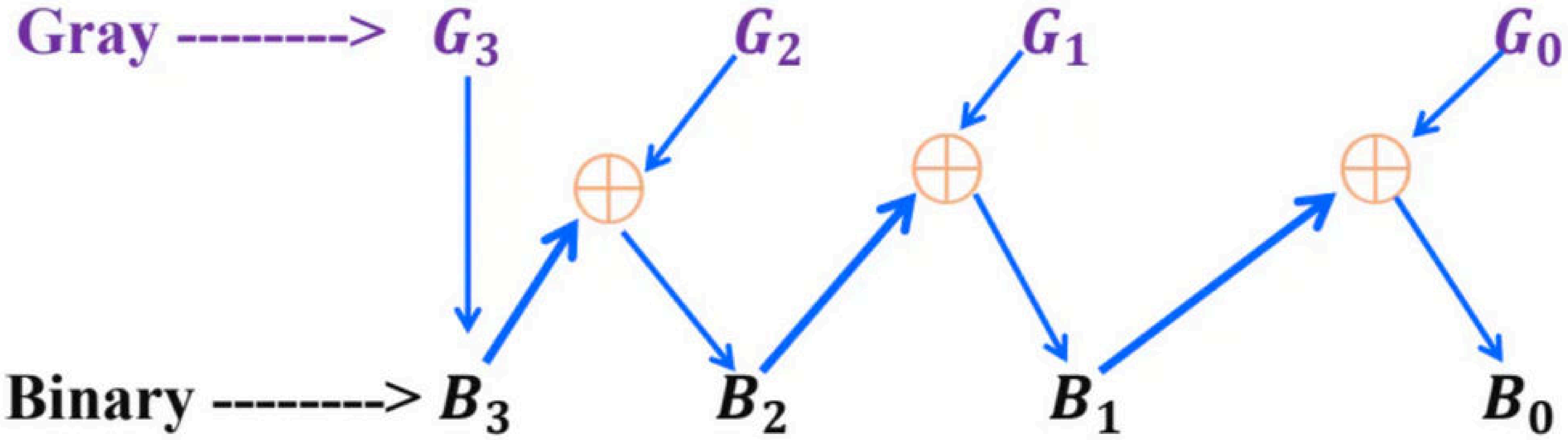
Q) Find the Gray code of the following

1 1 1 0 0 1 1 0

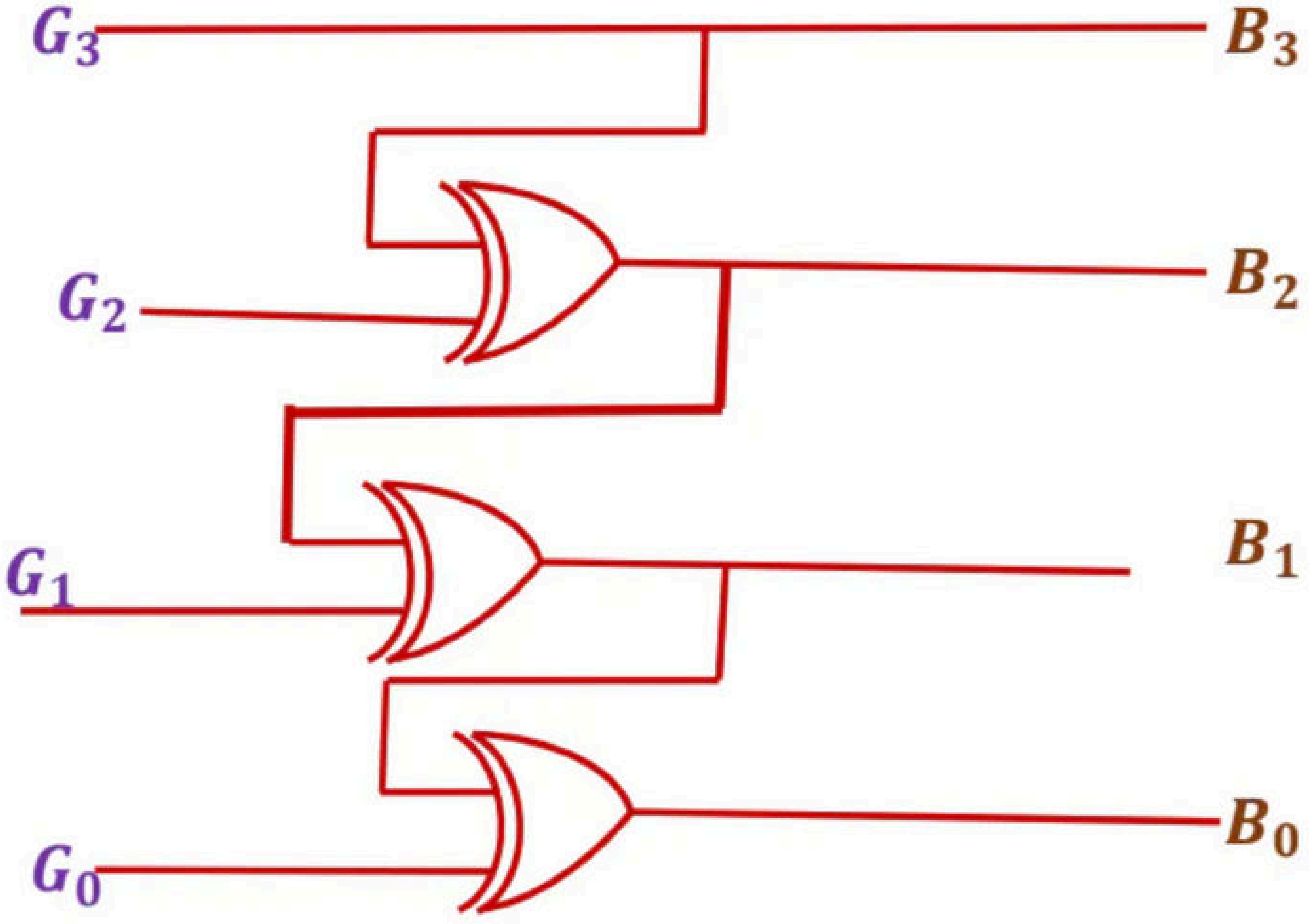
Logic gate



Gray to Binary Code



Gray Code to Binary Code



Q) Find the binary code of the following

1 0 1 0 1 1

Q) Find the binary code of the following

1 1 1 0 0 1 1 0

Self Complementing Codes

- A code is said to be self complementing, if the 1' complement of a number N is equal to the 9's complement of the number
- For a code to be self complementing, the sum of all its weights must be 9 .

Eg. of Self Complementing Codes

2 4 2 1
5 2 1 1
4 3 1 1
3 3 2 1
XS-3

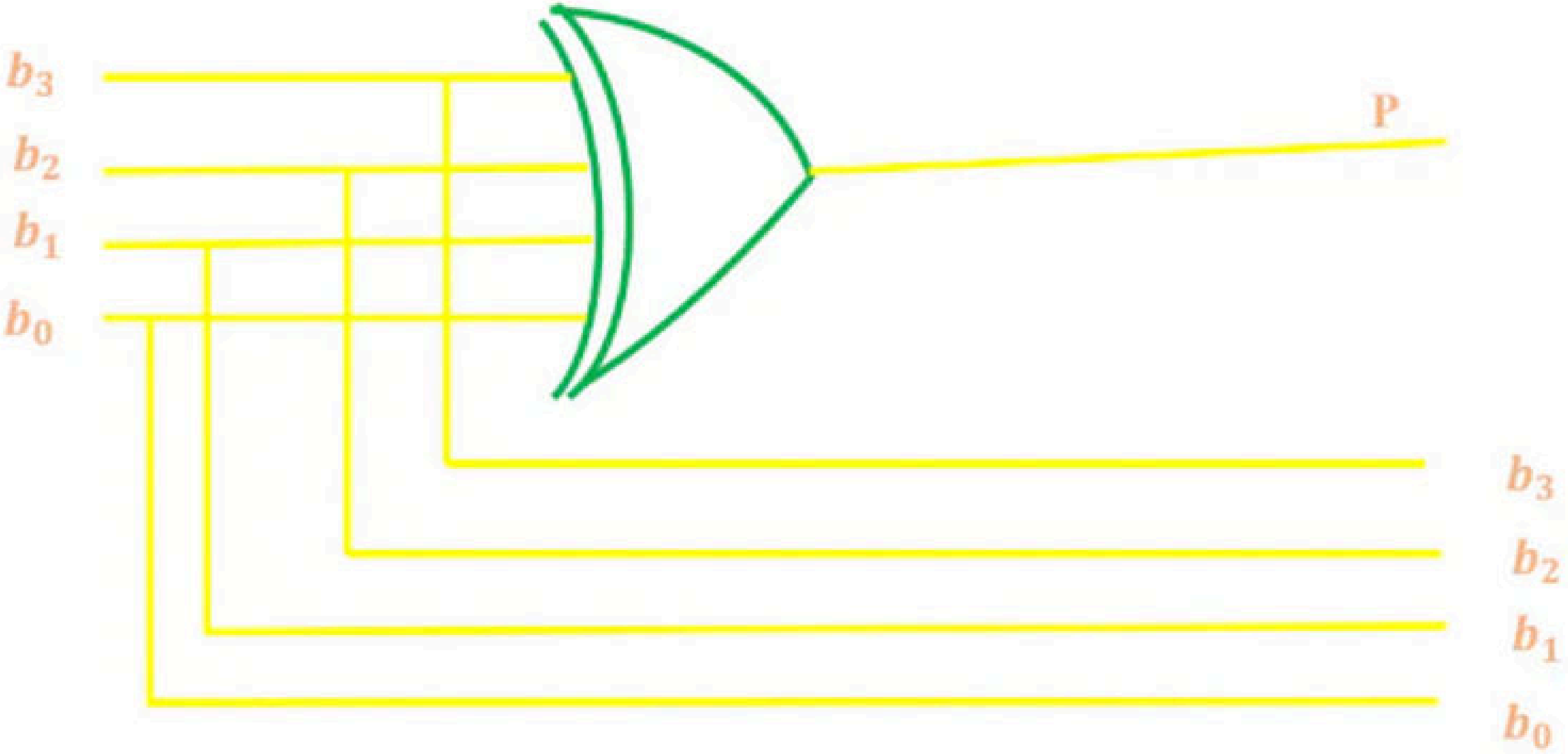
PARITY BIT

A parity bit is used for the purpose of detecting errors during transmission of binary information . A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver in called a parity checker .

Even parity

In case of even parity , the added parity bit will make the total number of 1's is an even number .

3- bit message	**Message with even parity**	
	message	**Parity**

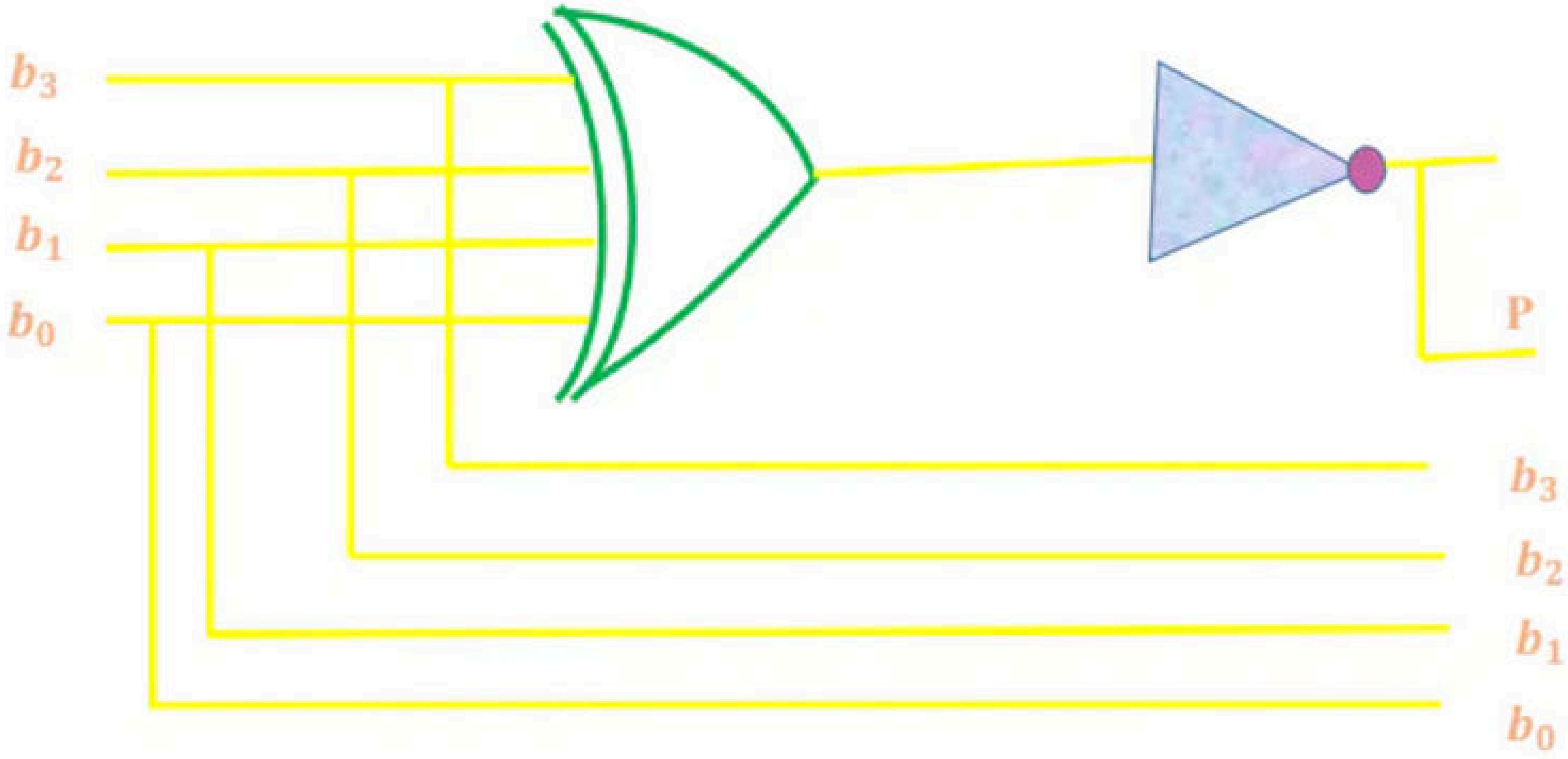


b_3
b_2
b_1
b_0
P
b_3
b_2
b_1
b_0

Odd parity

In case of odd parity , the added parity bit will make the total number of 1's is an odd number .

3- bit message	Message with odd parity	
	message	Parity



b_3
b_2
b_1
b_0
P
b_3
b_2
b_1
b_0

Q. P, Q and R are the decimal integers corresponding to the 4- bit binary number 1100 considering in signed magnitude,1's complement and 2's complement representations, respectively . The 6- bit 2's complement representation of (P+Q+R) is……..

a) 111101
b) 110101
c) 110010
d) 111001

Q. Which of the following represents ‘$E3_{16}$‘?

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

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

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

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

Q. A new Binary Coded Pentary (BCP) number systems is proposed in which every digit of a base -5 number is represented by its corresponding 3-bit binary code. For example, the base-5 number 24 will be represented by its BCP code 010100 . In this numbering system, the BCP code 100010011001 corresponds to the following number in base -5 system

a)423
b)1324
c)2201
d)4231

Q. Consider the addition of numbers with different bases

$$(X)_7 + (Y)_8 + (W)_{10} + (Z)_5 = (K)_9$$

If X=36 , Y = 67 ,W=98 and K =241 then Z is …..

a)34
b)15
c)68
d)25

Q. Let x_1 be the maximum value that can be represented in signed 2's complement form using 6 -bits.

Let x_2 be the minimum value that can be represented in signed 1's complement form using 5 bits.

Let x_3 be the minimum value that can be represented in signed 2's complement form using 7 bits.

Then find the value of $x_1 + \frac{x_2}{2} + 2x_3$

(a) -180.5
(b) -104.5
(c) 108.5
(d) 130.5

Q. The 4-bit binary number 1110 represents

(a) $(-1)_{10}$ in signed magnitude system and $(-2)_{10}$ in signed 2's complement system
(b) $(-1)_{10}$ in signed 1's complement system and $(-2)_{10}$ in signed 2's complement system
(c) $(-2)_{10}$ in signed magnitude system and $(-1)_{10}$ in signed 2's complement system
(d) $(-6)_{10}$ in signed magnitude system and $(-1)_{10}$ in signed 2's complement system

Q. A number N of base r is represented as $(N)_r$ let $(10_{16})^3 = {(X)_{10}}^2$

a)64

b)15

c)22

d)19