Ex-2 $F(A,B,C,D) = (0,3,7,5,4,11,15)$

Don't care ← $d(A,B,C,D) = (1,2,6)$

Sol ⇒ Total Variables = 4

Total Cells = $2^4 = 16$

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

Quad 1

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

A=0, C=0

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

+

Quad 2

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

C=1, D=1

$C \cdot D$

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

Ex ⇒

$$F(w,x,y,z) = \Sigma(1,3,7,11,15)$$

$$d(w,x,y,z) = \Sigma(0,2,5)$$

Sol ⇒

Total Variables = 4

Total cells = $2^4 = 16$

wx \ yz	00	01	11	10
00	X	1	1	X
01		X	1	
11			1	
10			1	

Quad 1

w	x	y	z
$\cancel{0}$	$\cancel{0}$	1	1
$\cancel{0}$	$\cancel{1}$	1	1
$\cancel{1}$	$\cancel{1}$	1	1
$\cancel{1}$	$\cancel{0}$	1	1

$y=1,\ z=1$

yz

Quad 2

w	x	y	z
0	$\cancel{0}$	$\cancel{0}$	1
0	$\cancel{0}$	$\cancel{1}$	1
0	$\cancel{1}$	$\cancel{0}$	1
0	$\cancel{1}$	$\cancel{1}$	1

$w=0,\ z=1$

$\bar{w}\cdot z$

$$yz + \bar{w}\cdot z$$

$$\boxed{= \bar{w}z + yz} \qquad \boxed{yz + \bar{w}z}$$

★ Don't Care Situation ⇒

→ जब किसी Boolean Expression का Output Unspecified हो या इस Output की Need नहीं हो, तो इस Condition को Don't Care कहते है।

→ इसे K-Map में X से denote किया जाता है।

→ इस Situation में Group बनाते समय सबसे बड़ा Group Prefer किया जाता है तथा कम से कम X को include किया जाना चाहिए।

	1	1	
	1	X	X

X			
X			
1	X		
1	1	1	1

Q=6. $F(A,B,C,D) = \Sigma(0,1,2,3,8,9,10,11,5,7,13,15)$

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

Octate 1

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

$D=1$

D

Octate 2

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

$B=0$

$\bar{B}$

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

Q=5 $F(A,B,C,D) = \Sigma(0,1,4,5,8,9,12,13)$

Total Variables = 4

Total Cells = $2^4 = 16$

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

Octate 1

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

C=0

Rule =

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

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

$Q=4$ $\quad F(A,B,C) = \Sigma(0,1,2,3,5,6)$

Total Variables = 3

Total cells = $2^3 = 8$

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

Quad 1

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

$A = 0$

$\bar{A}$

Quad 2

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

$B=0, C=1$

$\bar{B}.C$

Quad 3

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

$B=1, C=0$

$B.\bar{C}$

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

$\underline{Q=3}$ $F(A,B,C) = \Sigma(0,1,2,3,5)$

$= 3$ Variables

Total Cells: $2^3 = 8$

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

Quad 1

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

Quad 2

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

$\underline{A=0}$ → $\bar{A}$

$\underline{B=0}, \underline{C=1}$ → $\bar{B}$ C

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

Rule =

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

	00	01	11	10
0	0	1	3	2
1	4	5	7	6

⇒ 0, 1 cut करेंगे |

= Same Quad = multiply

= Different Quad = Add

Q=2 $\qquad F(A,B) = \Sigma(1,2)$

= 2 Variable

Total cells = $2^2 = 4$

A \ B	0	1
0		1
1		1

Pair 1

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

B = 1

$\boxed{= B}$

Q⇒1 $F(A,B) = \Sigma(0,2,3)$

$= 2$ Variables

cells $= 2^2 = 4$ cells

A \ B	0	1
0	1	
1	1	1

Pair 1

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

$B = 0$

$\bar{B}$

Pair 2

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

$1 = A$

A

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

Rule ⇒

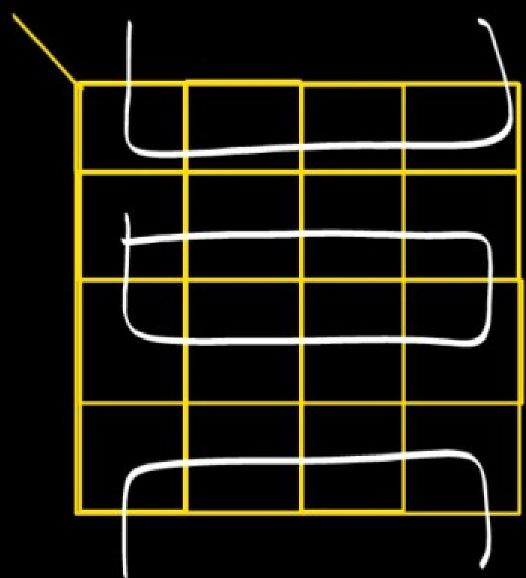
$1 = A$

$0 = \bar{A}$

A \ B		
	0	1
	3	2

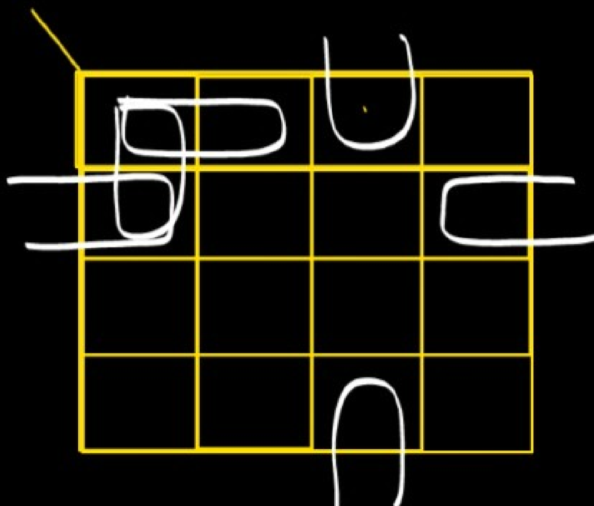
→ Same Pair में 0 & 1 को cut कर देते है।

C. Oct Group ⇒ 8 Cells का Combination

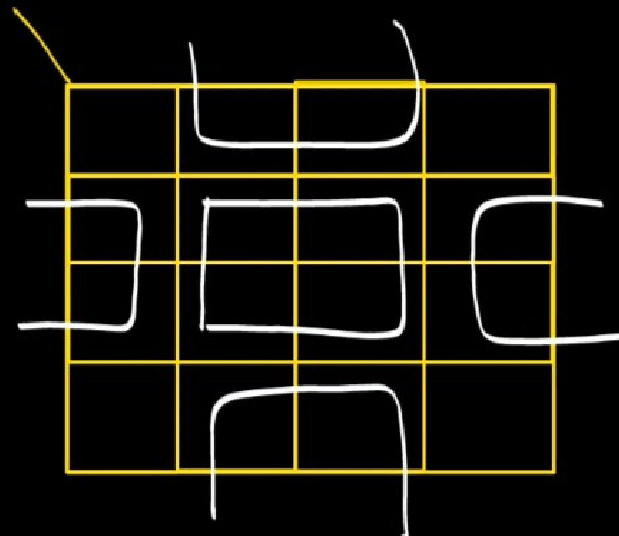


✧ Groups in K-Map ⇒

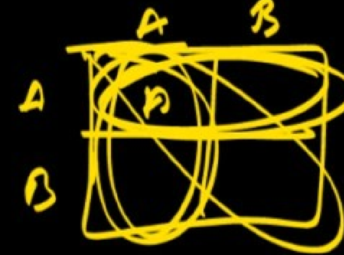
A. Pair ⇒ 2 Cells का combination



B. Quad Groups ⇒ 4 Cells का combination



☆ Rules for creating group in K Map ⇒



1. No Zeros allowed.
2. Group can be Horizontal & vertical but not diagonal.
3. Overlapping is allowed.
4. Group should be as large as possible.

$2^2 = 4$

$A \backslash B$		
	0	1
	3	2

$2^3 = 8$

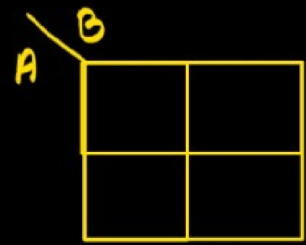
$A \backslash BC$				
	0	1	3	2
	4	5	7	6

$2^4 = 16$

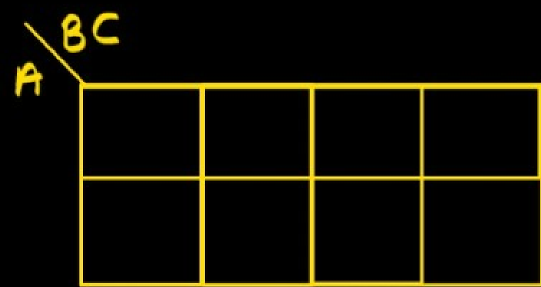
$AB \backslash CD$				
	0	1	3	2
	4	5	7	6
	12	13	15	14
	8	9	11	10

जैसे -

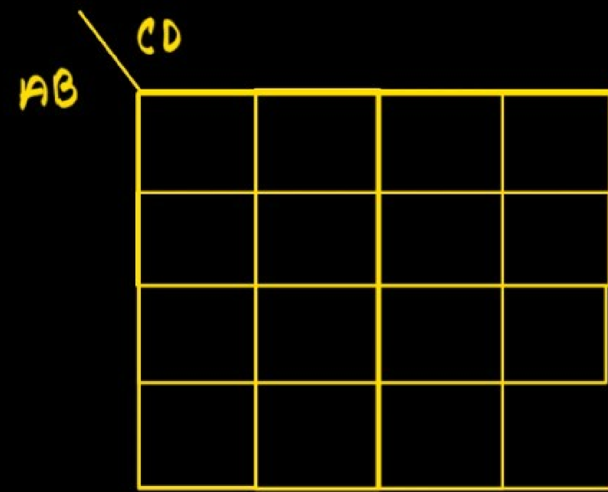
$2^2 = 4$



$2^3 = 8$



$2^4 = 16$



K-Map ⇒

- → Karnaugh Map
- → Launched by Karnaugh in 1953.
- → It is used to Simplify boolean expressions in pictorial format without using laws.
- → Formula = $\boxed{2^n}$

n = number of variables

(0 - 9)

minterm (0, 1, 3, 5, 7, 9)

maxterm (2, 4, 6, 8)

Question ⇒ Represent / Expand $F(A,B) = \bar{A} + \bar{B}$ in SOP & POS. Also find minterm & maxterm.

Sol ⇒ $F(A,B) = \bar{A} + \bar{B}$

B missing (for $\bar{A}$) A missing (for $\bar{B}$)

→ $A + \bar{A} = 1$

→ $B + \bar{B} = 1$

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

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

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

$$= 01 + 00 + 10$$

$$= m1 + m0 + m2$$

$$\boxed{= \Sigma m(0,1,2)}$$ minterm

in minterm m3 is missing, so m3 will be maxterm.

Hence $POS = \Pi m(3)$

$$= 1\ 1$$

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

Express F in Product of Sum form

	A	B	F	
m_0	0	0	0	✓
m_1	0	1	1	
m_2	1	0	0	✓
m_3	1	1	1	

* when F = 0

$0 = A$

$1 = \bar{A}$

$$F(A,B) = (0+0)\cdot(1+0)$$

$$F(A,B) = \underbrace{(A+B)\cdot(\bar{A}+B)}_{\text{MaxTerm}}$$

$$F(A,B) = m0 \cdot m2$$

$$\boxed{F(A,B) = \pi m(m_0, m_2)} \Rightarrow F(A,B) = \pi m(0.2)$$

$$(0,2)$$

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

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

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

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

$$\boxed{= A+B}$$ Minimul SOP Form

Distributive law

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

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

Standard or Canoninical form

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

(Standard / Canonical

$f(A,B)$

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

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

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

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

$$A + AB + BC + ABC$$

Express F in Sum of Product form

	A	B	F
m_0	0	0	0
m_1	0	1	1
m_2	1	0	1
m_3	1	1	1

$$\boxed{\begin{aligned} \Sigma m &= SOP \\ \Pi m &= POS \end{aligned}}$$

* When F=1

$0 = \bar{A}$

$1 = A$

$\Rightarrow 01 + 10 + 11$

$$\boxed{\bar{A}B + A\bar{B} + AB} \rightarrow \text{Standard or Canonical Form}$$

$$F(A,B) = m_1 + m_2 + m_3$$

$$\boxed{F(A,B) = \Sigma m(1+2+3)}$$

SOP

$\downarrow$

Minterm

☆ SOP & POS ⇒

SOP	POS
1. Sum of Product जैसे- $(AB)+(BC)+(AC)$	1. Product of Sum जैसे- $(A+B)\cdot(B+C)\cdot(A+C)$
2. When function $F=1$	2. When function $F=0$
3. $A \longrightarrow 1$ $\bar{A} \longrightarrow 0$ जैसे- 00 , 01 $\bar{A}\bar{B} + \bar{A}B$	3. $A \longrightarrow 0$ $\bar{A} \longrightarrow 1$ जैसे- 00 , 01 $(A+B)\cdot(A+\bar{B})$

9. IIL (Integrated Injection Logic)

10. HTL (High Threshold Logic)

11. TTL (Transister Transister Logic)

12. ECL (Emitter Coupled Logic)

13. PMOS (P-Channel Matel Oxide Semiconductor)

14. NMOS (N-Channel Matel Oxide Semiconductor)

15. CMOS (Complementry Matel Oxide Semiconductor)

16. NMOS Voltage High होने पर ON होता है व Low होने पर OFF होता है, whereas PMOS Voltage High होने पर OFF होता है व Voltage Low होने पर ON होता है तथा इन दोनो को मिलाकर CMOS बनाया जाता है।

☆ Important Points ⇒

1. DCTL (Direct Coupled Transister Logic) की मुख्य Problem Current Hogging है, जिसे दुर करने के लिए IIL (Integrated Injection Logic) का use होता है।
2. Multiple Parallal Lines में से कोई एक Line जब ज्यादा current खींचती है, तो इसे Current Hogging कहते है।
3. Noise Immunity सबसे अच्छी CMOS की होती है।
4. IIL (Integrated Injection Logic) Multi Collector Transister का use करते है, जिससे इसकी Packing Density High होती है।
5. Power Dissipation CMOS की सबसे कम होती है।
6. RTL (Resistor Transister Logic)
7. DTL (Diode Transister Logic)
8. DCTL (Direct Coupled Transister Logic)

8. Logic Swing ⇒

- → दो output voltage के बीच का अन्तर |
- → $V_H - V_L$

9. Noise Immunity ⇒

- → वह Maximum Noise जो output को change किये बिना bother / सहन की जा सके |

6. Noise Margin ⇒

→ यह Noise Voltage की Limit होती है, जो बिना किसी issue के Circuit को operate करती है।

7. Figure of Merit ⇒

→ Power Dissipation & Propogation Delay का Multiply.

→ $FOM = P_D \times t_{PD}$

→ Unit = पिको जुल

4. Power Dissipation ⇒

- → प्रत्येक Gate द्वारा use की गई Energy.
- → Unit = mW (मिलीवाट)
- → इसे P_D से denote किया जाता है।

5. Propogation Delay ⇒

- → किसी Signal के Input से Output तक transmit होने में लगा समय।
- → Unit = NanoSecond (nsec)
- → इसे t_{PD} से denote किया जाता है।

Characteristics of Logic Family ⇒

1. Speed ⇒

→ Output में change & Input के बीच Time decide करती है।

2. Fan Out ⇒

→ Load की वह संख्या जो Gate के Output को Drive कर सके।

3. Fan In ⇒

→ किसी logic gate को दिये जा सकने वाले Max Inputs.

Logic Family ⇒

- Logic Family
  - BiPolar
    - Saturated
      1. RTL
      2. DTL
      3. DCTL
      4. IIL
      5. HTL
      6. TTL
    - Unsaturated
      1. Schottky TTL
      2. ECL
  - UniPolar
    1. PMOS
    2. NMOS
    3. CMOS

B. Synchronous / Parallal Counter ⇒

- → इस Counter में सभी Flip-Flops, एक ही Clock Pulse से Trigger होते है।
- → इसमें सारे Flip-Flops Parallal connected होते हैं, इसलिए इसे Parallal Counter कहते है।
- → High Speed.
- → इसका Circuit अधिक Complex होने के कारण अधिक Hardware की Need होती है।

6. Noise Margin ⇒

→ यह Noise Voltage की Limit होती है, जो बिना किसी issue के Circuit को Operate करती है।

4. Power Dissipation ⇒

- → प्रत्येक Gate द्वारा use की गई Energy.
- → Unit = mW (मिलीवाट)
- → इसे P_D से denote किया जाता है।

5. Propogation Delay ⇒

- → किसी Signal के Input से Output तक transmit होने में लगा समय।
- → Unit = NanoSecond (nsec)
- → इसे t_{PD} से denote किया जाता है।

Characteristics of Logic Family ⇒

1. Speed ⇒

→ Output में change & Input के बीच Time decide करती है।

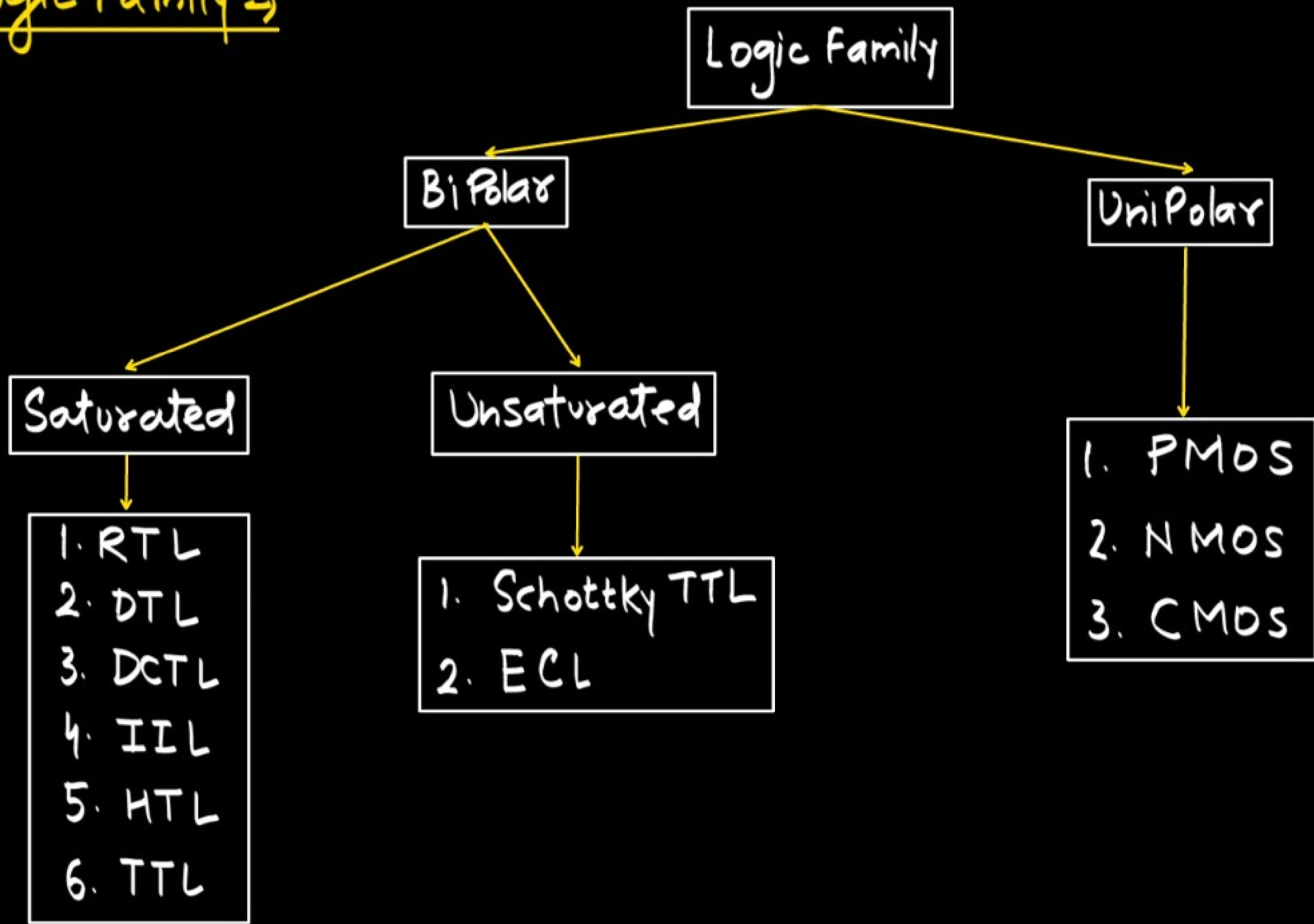
2. Fan Out ⇒

→ Load की वह संख्या जो Gate के Output को Drive कर सके।

3. Fan In ⇒

→ किसी logic gate को दिये जा सकने वाले Max Inputs.

✡ Logic Family ⇒



B. Synchronous / Parallal Counter ⇒

- → इस Counter में सभी Flip-Flops, एक ही clock Pulse से Trigger होते है।
- → इसमें सारे Flip-Flops Parallal connected होते है, इसलिए इसे Parallal Counter कहते है।
- → High Speed.
- → इसका Circuit अधिक Complex होने के कारण अधिक Hardware की Need होती है।

✲ Counter ⇒

→ Clock Pulses को Count करता है।

→ 2 प्रकार -

(A) ASynchronous / Ripple Counter

(B) synchronous / Parallal Counter

A. ASynchronous / Ripple Counter ⇒

→ Serial Counter क्योंकि इसमें Flip-Flops Serial order में Arrange होते है।

→ Speed कम।

→ इसमें केवल LSB वाले FlipFlop को ही सीधी Clock-Pulse दी जाती है।

→ इसमें LSB वाले Flip-Flop के अतिरिक्त अन्य सभी Flip-Flops अपने से पहले वाले Flip-Flop के Output से Trigger होते है।

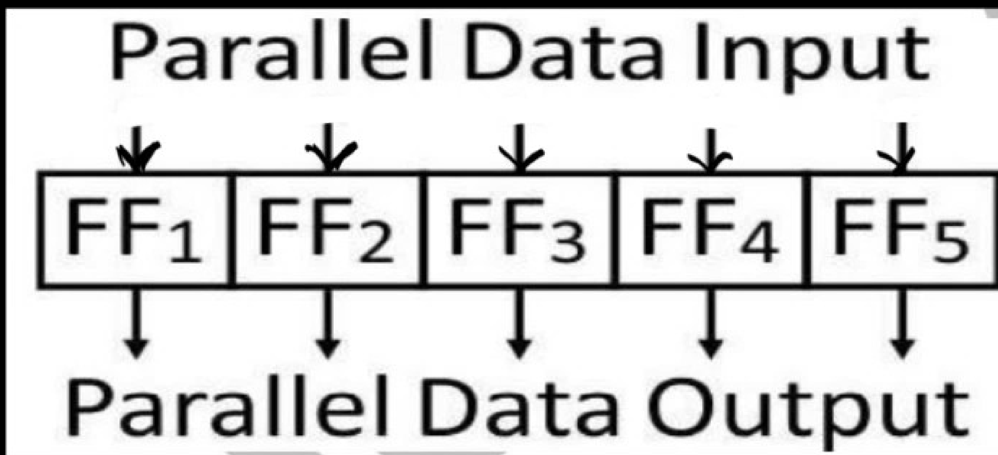
→ इसमें Flip-Flop Trigger wave of water की तरह कार्य करते है, इसलिए इसे Ripple Counter कहते है।

(V) Bi-Directional Register ⇒

- SISO, SIPO, PISO, PIPO में Present Data की Shifting Clock Pulse की Presense में केवल एक Direction में होती है।
- इस Register में Data दोनो Side / Direction में shift किया जा सकता है।
- इसमें Control Signal = 1 होने पर Data Right Side में तथा Control Signal = 0 होने पर Data Left Side में shift होता है।

(IV) PIPO ⇒

→ Parallal In Parallal Out

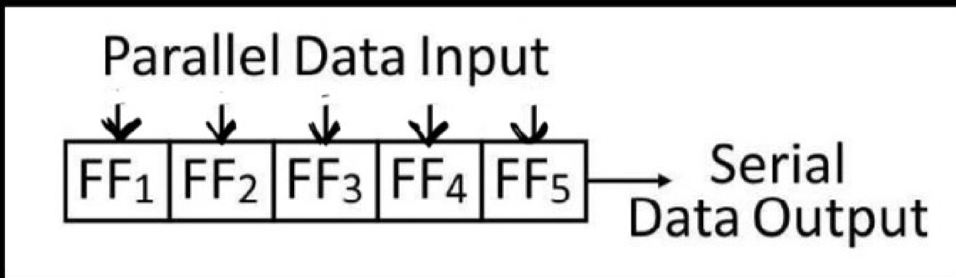


→ Fastest Register

→ यह Register Input & Output दोनो Parallal Order में ही लेता है, अर्थात् Multiple Bits at a time, इसी कारण यह Fastest होता है|

(iii) PISO ⇒

→ Parallal In Serial Out

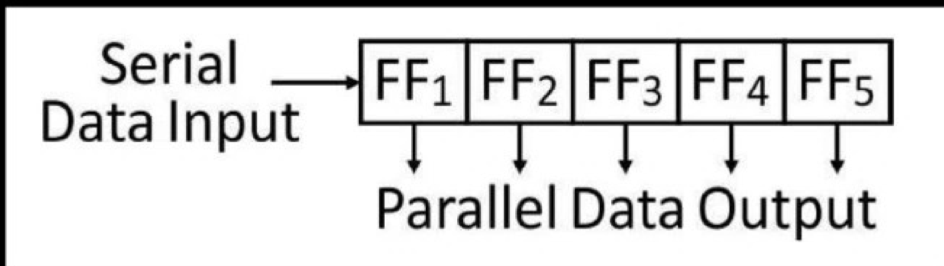


→ इस Register में Store किया जाने वाला Data Parallal Order में दिया जाता है अर्थात् Multiple Bits at a time.

→ Stored Data को Output में Serial Order में Receive किया जाता है अर्थात् One Bit at a time.

(ii) SIPO ⇒

→ Serial In Parallel Out

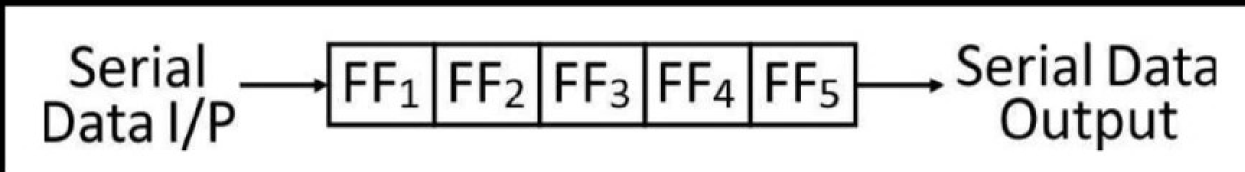


→ इस Register में store किया जाने वाला Data serial Order में दिया जाता है अर्थात One Bit at a time.

→ इस Register में stored information को Output में Parallel Order में Receive किया जाता है अर्थात Multiple Bits at a Time.

(i) SISO ⇒

→ Serial In Serial Out



→ Register में store किया जाने वाला Data Serial Order में Input किया जाता है अर्थात् One bit at a time.

→ Stored Information को Output में भी Serial Order में ही Receive किया जाता है अर्थात् One bit at a time.

→ Slow Process अर्थात् Data Input करने व Output करने में ज्यादा Time लगता है।

A. Buffer Register ⇛

- → एक General Register जो Binary Data को Temporary रूप से store करता है।
- → D Flip-Flop का use.

B. Shift Register ⇛

- → JK Flip-Flip का use.
- → यह Clock Pulse के Receive होने पर Stored Data को एक Bit आगे ले जाता है।
- → यह Multiple Flip-Flops से मिलकर बनता है, जो एक साथ Same Clock Pulse Receive करते है।
- → प्रकार -
  - (i) SISO
  - (ii) SIPO
  - (iii) PISO
  - (iv) PIPO
  - (v) Two directional Shift Register

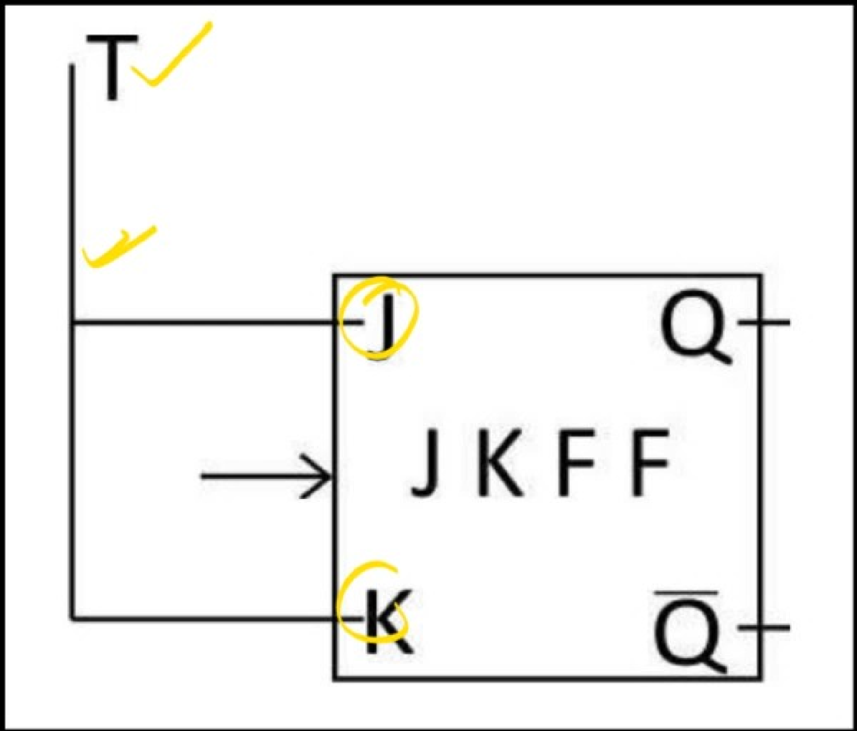
Register ⇒

- यह एक Logical Circuit होता है, जो Binary Information को Store तथा उनकी Movement करने के लिए प्रयोग में लिया जाता है।
- यह Multiple Flip-Flops से बनते है, जो आपस में Logically Connected होते है।
- N Bits को Store करने के लिए N Flip-Flops की Need होती है।
- इसमें Information को Serial या Parallel किसी भी Pattern में Store & Retrieve किया जा सकता है।
- Serial form के Data को Temporary & Parallel form के data को positional Data कहते है।
- 2 प्रकार ⇒
  - A. Buffer Register
  - B. Shift Register

E. T Flip-Flop ⇒

- Toggle Flip-Flop
- J-K Flip-Flop का परिवर्तित रूप।
- इसमें दोनों Input Terminals J&K को Combined कर एक Single Terminal T का use किया जाता है।
- **Truth Table ⇒**

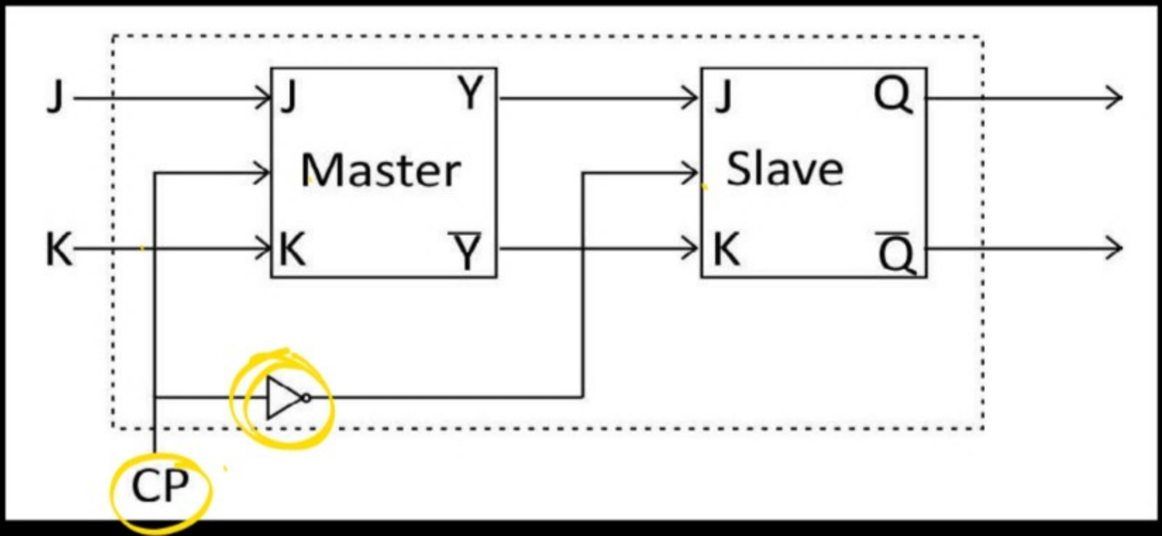
CLK	T	Q_{n+1} (Next State)
0	0	Q_n
1	0	Q_n = Hold
1	1	$\overline{Q_n}$ = Toggle



D. JK Master Slave Flip-Flop ⇒

→ इसका प्रयोग JK Flip-Flop की Race Around Condition को दुर करने में किया जाता है।

→ Master Flip-Flop Clock Pulse की Positive Edge पर Output देता है और Slave Flip-Flop Clock Pulse की Negative Edge पर Output देता है।



→ Slave Flip-Flop के Edge Trigger होने पर इसके output पर Race Around की condition नहीं बनती है।

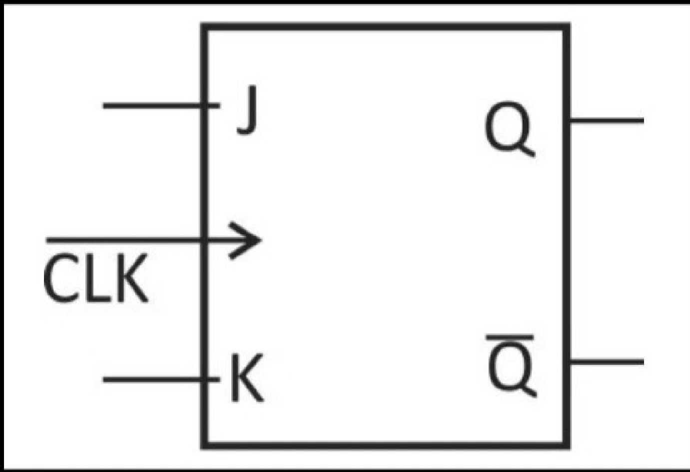
→ Maste Flip-Flop का Output तभी change होगा, जब Slave का output change होगा।

* Race Around Condition ⇒

- J = K = 1 होने पर यह Condition बनती है।
- यह Level Trigger होने पर होती है तथा इसे दुर करने के लिए Edge Trigger किया जाता है।
- JK Flip-Flop में Toggle Mode (J=K=1) में operation के समय use किया गया Clock Duration की Value यदि Flip-Flop के Propagation Delay से अधिक होने पर ($T_{PW} > T_{PD}$) Flip-Flop Multitoggling करने लगता है, इसे Race Around Condition कहते है।
- इस Condition को दुर करने के लिए Clock Duration को Flip-Flop के Propagation Delay से कम कर दिया जाता है। ($T_{PW} < T_{PD}$)
- इस Condition को Edge Triggered Flip-Flop से ही खत्म किया जा सकता है।

T_{PW} = Clock Pulse Width
T_{PD} = Propagation Delay

C. JK Flip-Flop ⇒



- → JK Flip-Flop, SR Flip-Flop की Invalid State को दुर करता है।
- → इसमें Set, Reset, No change conditions SR Flip-Flop के जैसे ही होती है।
- → इसमें J=K=1 की condition को Race Condition / Race Around कहते है।
- → Race Around condition को दुर करने के लिए Master Slave Flip-Flop का use किया जाता है।
- → JK = Jack Kilby (Jump-Key Flip Flop)
- → Truth Table ⇒

CLK	J	K	Q_{n+1}
×	0	0	Q_n (Previous State)
1	0	0	Q_n (Previous State)
1	0	1	0 (Reset)
1	1	0	1 (Set)
1	1	1	$\overline{Q_n}$ (Toggle)

B. D Flip-flop ⇒

→ D = Delay

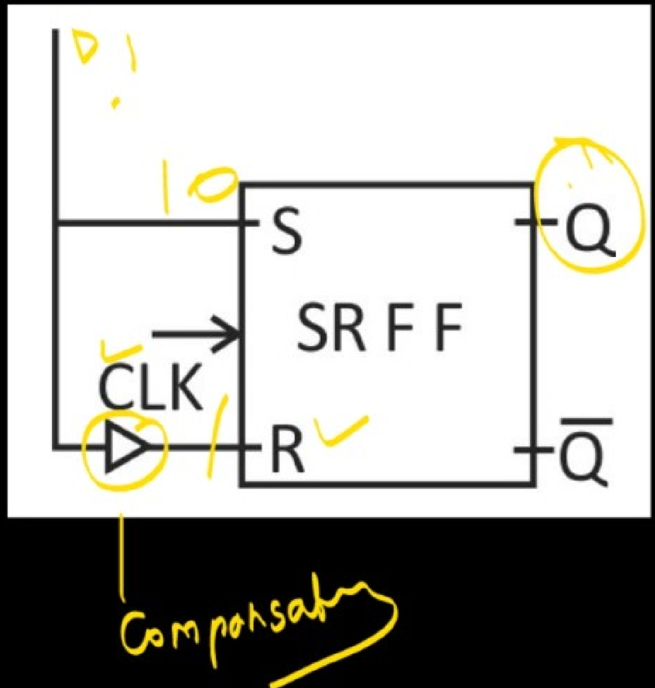
→ यह SR Flip-Flop का परिवर्तित रूप है, इसमें केवल एक Input Terminal D होता है।

→ मुख्यत: इसका use Register बनाने में किया जाता है।

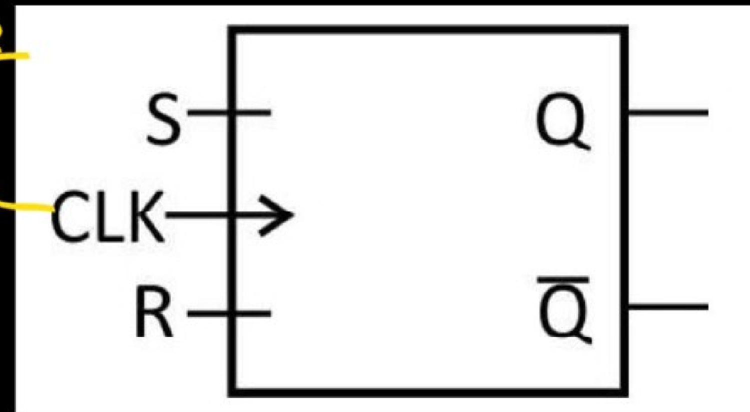
→ इसे Data Transmission Flip-Flop या Transparent latch Flip-Flop भी कहते हैं।

→ Truth Table ⇒

CLK	D	Next State (Qn+1)
0	X	last state (Qn)
1	0	0
1	1	1



A. S-R Flip-Flop ⇒



- Latch Flip-Flop
- NAND या NOR Gate से बना सकते हैं।
- NOR Gate का प्रयोग करके Active High SR Flip Flop & NAND Gate का प्रयोग करके Active Low SR Flip Flop बनाये जाते हैं।
- इसमें 2 Inputs = S(Set) & R(Reset) होते हैं।
- इसमें S=R=1 हो जाने पर Not Permitted Condition बन जाते हैं, जिसके Solution के लिए JK Flip Flop का use किया जाता है।
- इसमें S=R=0 हो जाने पर Memory State बन जाती है।
- Truth Table ⇒

S	R	Q	Action
0	0	Last Value	No Action
0	1	0	Reset
1	0	1	Set
1	1	Invalid	Not Permitted

☆ Flip-Flop ⇒

- → Logic Gate & Combinational Circuits से बनता है।
- → Data Transfer, Data Storage & Frequency Divider के रूप में use.
- → 2 Inputs & 2 outputs.
- → इसका Output Q & $\overline{Q}$ होता है।
- → 1 Bit Store
- → Types ⇒
  - A. S-R Flip Flop
  - B. D Flip-Flop
  - C. J K Flip-Flop
  - D. JK Master Slave Flip Flop
  - E. T Flip-Flop

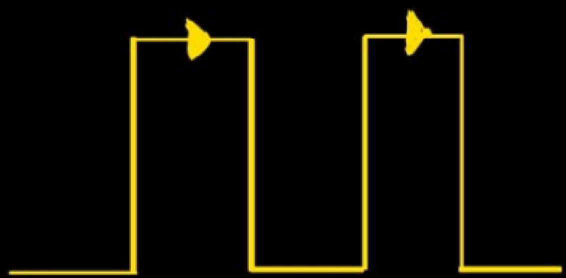
☆ Flip-Flop V/s Latches ⇒

Flip-Flop	Latch
1. Synchronous Input	1. Asynchronous Input
2. Clock Pulse का use	2. Clock Pulse का use नहीं
3. Permanant Storage	3. Temporary Storage
4. Edge Trigger की Need	4. Level Trigger की Need
5. More Power Consume	6. Lower Power Consume

B. Asynchronous Sequential Circuit ⇒

→ इसमें Memory element Unclocked FlipFlop या Time Delay Element होता है

→ Level Triggered.



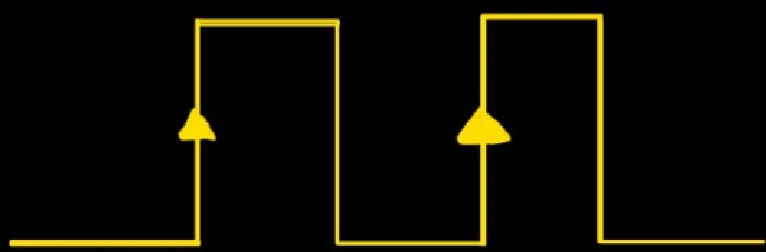
Positive level Trigger

Negative level Trigger

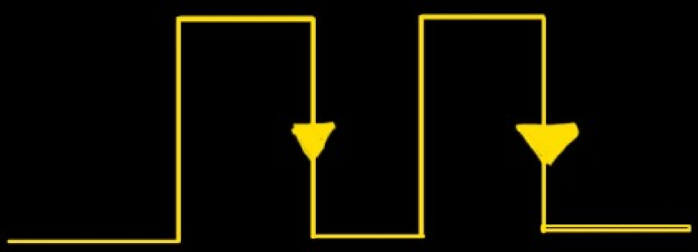
→ Clock Pulse की Absense के कारण यह Synchronous Circuit से तेज होता है

A. Synchronous Sequential Circuits ⇒

- → इसमें Memory = Clock Flip Flop होता है।
- → Edge Triggered है।



Positive Edge Trigger



Negative Edge Trigger

- → Asynchronous की तुलना में slow.

* Sequential Logic Circuit ⇒

→ इसका Output, Input और Output की Previous State पर निर्भर करता है।

→ इन Circuits को Previous state/ Information को store करने के लिए Memory की Need होती है।

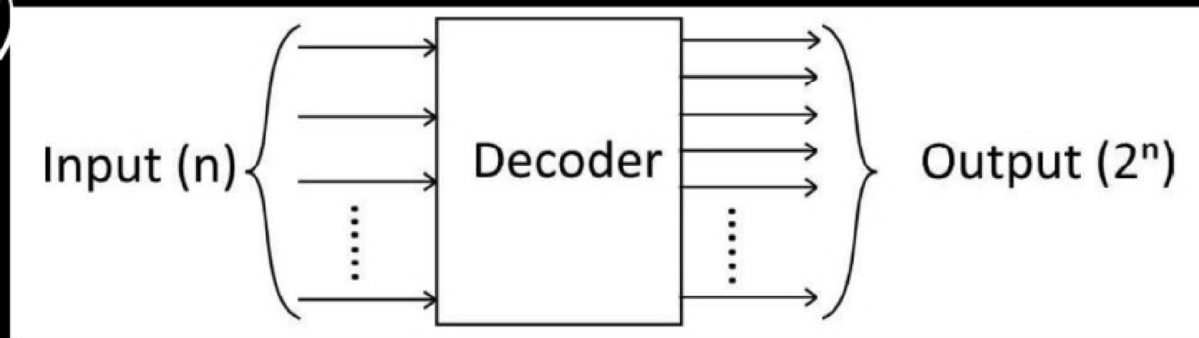
→ Slower than Combinational Circuits.

→ Feedback का use किया जाता है।

→ 2 प्रकार = A. Synchronous Sequential Circuit
B. Asynchronous Sequential Circuit

D. Decoder ⇒

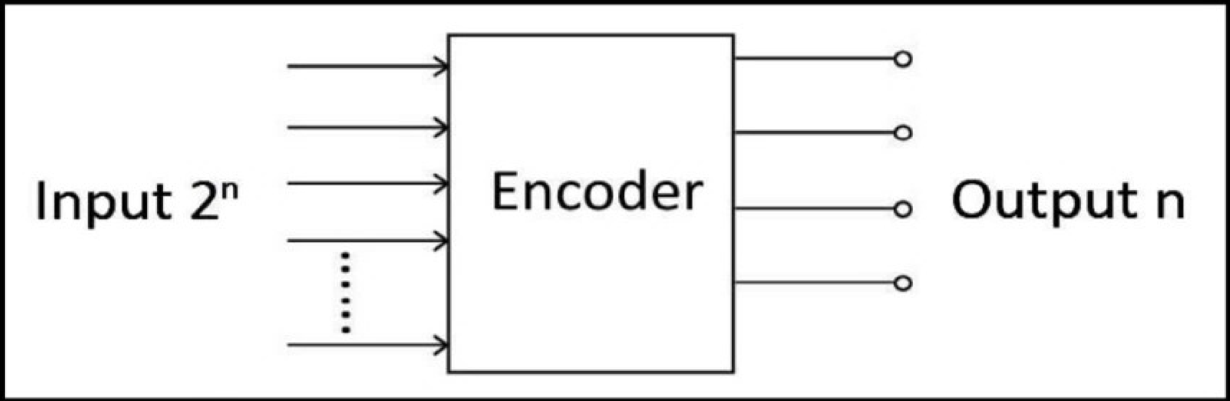
- → Encoder का Reverse operation करने में use.
- → इसमें Multiple Inputs & Multiple Outputs होते हैं, लेकिन Outputs की संख्या Inputs की संख्या से अधिक होती है।
- → Size = $n \times 2^n$ (n = Input & 2^n = output)
- → BCD Decoder भी कहते हैं।
- → Binary को Decimal में change करने में useful.



<u>Ex-</u> 3×8 Decoder, 4×16 Decoder.

C. Encoder ⇒

→ इस Circuit में Multiple Inputs & Multiple Outputs होते हैं, लेकिन Inputs की संख्या Output से अधिक होती है।



→ Size = $2^n \times n$ (2^n = Input & n = Output)

→ इसका use अन्य Codes को Binary में बदलने में भी किया जाता है।

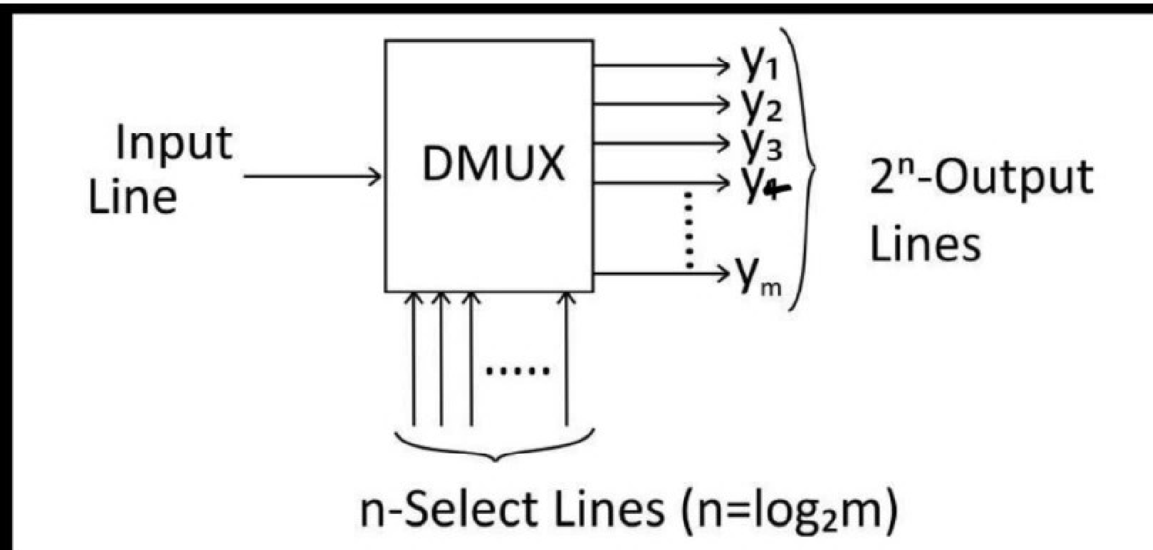
→ इसे Decimal to BCD Encoder भी कहते हैं।

Ex- Octal to Binary = 8×3 Encoder
Decimal to Binary = 10×4 Encoder
HexaDecimal to Binary = 16×4 Encoder

B. Demultiplexer ⇒

- → Works on one to Many Principle
- → Single Input & Multiple Output.
- → MUX के Reverse Operation को करने में use.
- → इसे One to Many Convertor, Serial to Parallel Convertor, Data Distributor भी कहते है।
- → Demux को Universal Gate के रूप में प्रयोग में नहीं लिया जा सकता है।

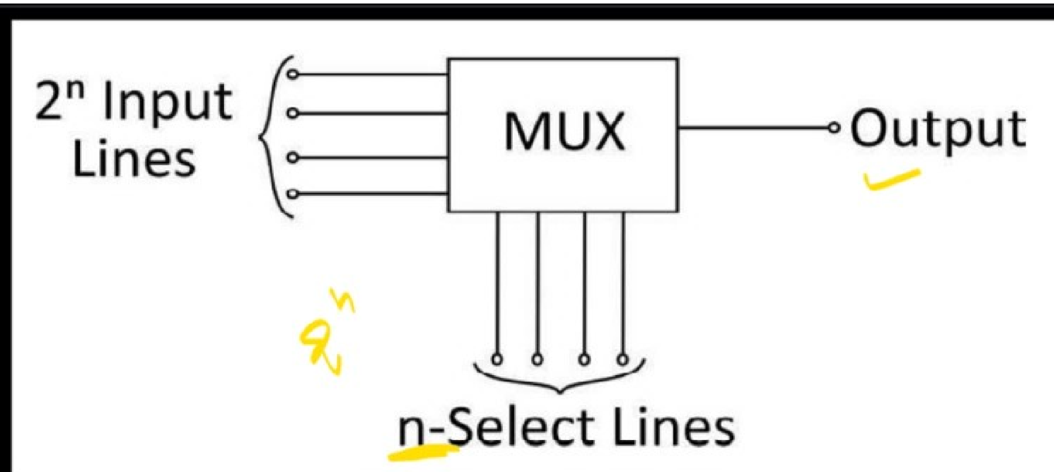
Ex- 1×2 DeMUX, 1×4 Demux, 1×8 Demux etc.



A. Multiplexer ⇒

- → Works on Many to one principle.
- → Multiple Inputs & Single Output.
- → इसमें Inputs की संख्या Select Lines पर निर्भर करती है अर्थात् n select lines होने पर Inputs की संख्या 2^n होगी।
- → इसे Data Selector भी कहते हैं, क्योंकि Multiple Inputs में से केवल एक सही Output देने के लिए use में लिया जाता है।
- → इसकी सहायता से Multiple Circuits बनाये जा सकते हैं, इसलिए इसे Universal Circuit कहते हैं।
- → इसे Data Selector, Many to One Circuit, Universal Logic Convertor, Parallal to serial Convertor, Wave form generator भी कहते हैं।

Ex = 2×1 MUX, 4×1 MUX, 8×1 MUX etc.



☆ Based on Data Transmission ⇒

A. Multiplexer

B. Demultiplexer

C. Encoder

D. Decoder

C. Comparator ⇒

→ ऐसा Circuit जो दिये गये Input के आधार पर Comparisan कर Output देता है।

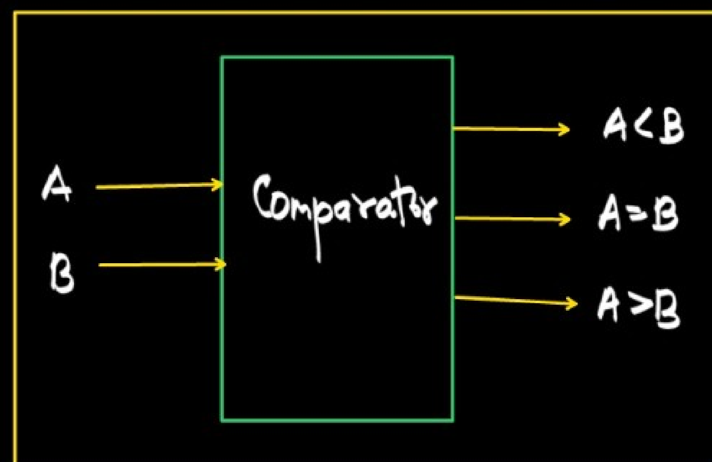
→ यह Circuit Inputs को compare निम्न Output देता है-

(i) Inputs are Same $(A==B)$

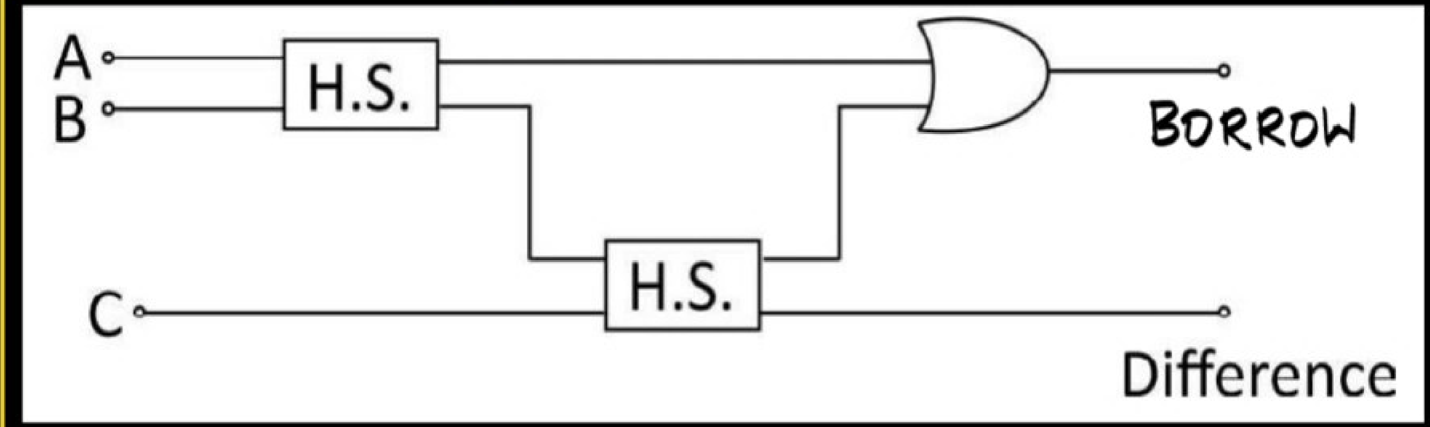
(ii) पहला Input दुसरे से बड़ा है। $(A>B)$

(iii) पहला Input दुसरे से छोटा है। $(A<B)$

Inputs	Output		
A, B	$A>B$	$A=B$	$A<B$
	$Y=1$	$Y=1$	$Y=1$

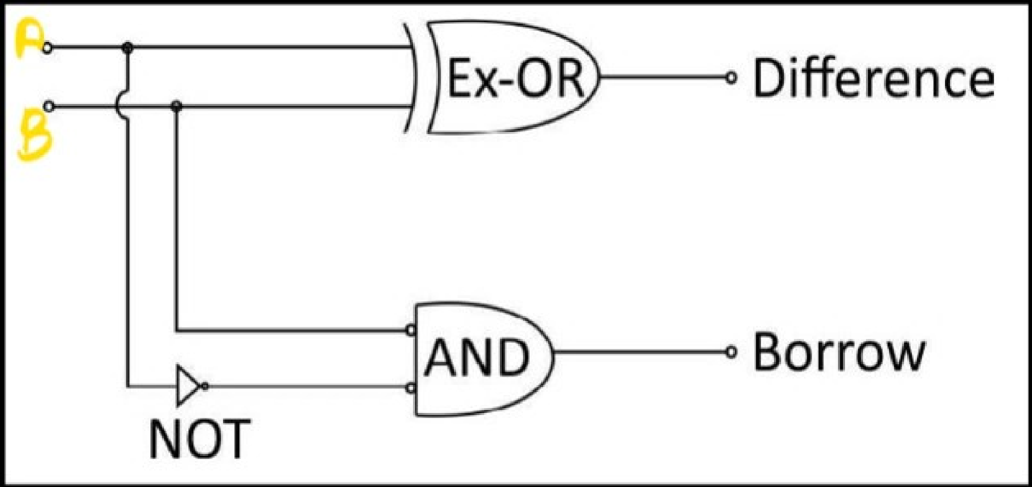


2. Full Substractor ⇒



- → 3 Inputs & 2 outputs
- → 2 Half Substractors + 1 OR Gate
- → Total 7 Gates = 2 Ex-OR, 2 AND, 2 NOT & 1 OR Gate

1. Half Substractor ⇒



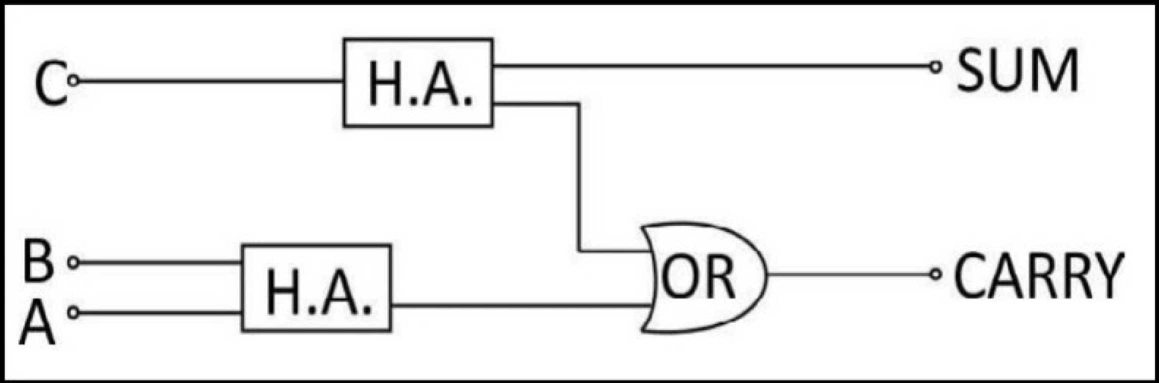
→ 2 Inputs & 2 Outputs

→ 3 Gates से निर्मित = 1 Ex-OR, 1 AND & 1 NOT

B. Substractor ⇒

- → Binary Numbers के Substraction में use.
- → 2 Output = Borrow & Difference
- → 2 प्रकार = 1. Half Substractor
  2. Full Substractor

2. Full Adder ⇒



→ तीन Inputs तथा 2 Outputs. (3 Inputs = 2 Input + 1 Carry)

→ 2 Ex-OR Gates, 2 AND Gates तथा 1 OR Gate से निर्मित। (Total 5 Gates)

→ Full Adder को 2 Half Adder & 1 OR Gate से भी बना सकते है।

→ इसका output SUM + CARRY होता है।

→ इसे बनाने के लिए 9 NAND Gates या 9 NOR Gates का use किया जाता है।

❖ Based on Arithmatic & Logical Functions ⇒

A. Adder ⇒

→ Binary Numbers का Addition करने में use.

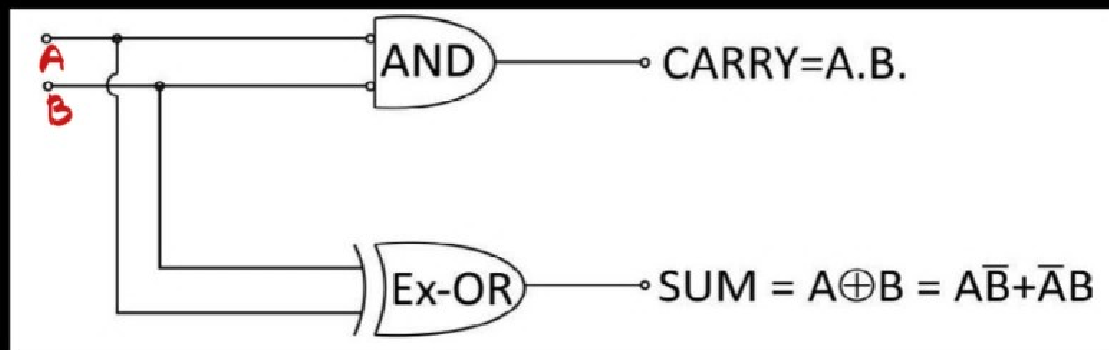
→ 2 प्रकार = 1. Half Adder
2. Full Adder

1. Half Adder →

→ 2 Input & 2 Output

→ 1 AND तथा 1 Ex-OR Gate से निर्मित।

→ AND Gate का Output = Carry तथा Ex-OR Gate का Output = Sum ($A \oplus B$)



→ Half Adder को बनाने के लिए 5 NAND Gates या 5 NOR Gates की Need होती है।

Classification of Combinational Circuits :

Combinational Circuits

- Based on Arithmatic & logical Function
  - A. Adder
  - B. Substractor
  - C. Comparator
- Based on Data Transmission
  - A. Multiplexer
  - B. De-Multiplexer
  - C. Encoder
  - D. Decoder

☆ Combinational Circuits ⇒

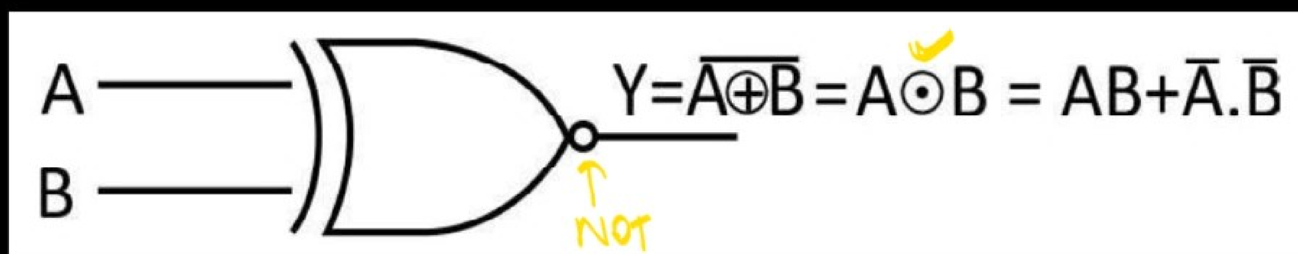
- → वह Circuit जिसका Output केवल Present Input पर ही Depend करता है।
- → No Dependancy on Feedback, Memory & Timing.
- → Logic Gates का use कर बनाते है।
- → Memory Element नहीं होता है।

LOGIC GATE	NAND की संख्या	NOR की संख्या
NOT	1	1
AND	2	3
OR	3	2
Ex-OR	4	5
Ex-NOR	5	4

B. EX-NOR Gate ⇒

- → Exclusive NOR या XNOR Gate
- → XOR + NOT Gate से निर्मित
- → इस Gate का प्रयोग दो Inputs की equality को Compare करने में किया जाता है।
- → 2 Inputs & 1 Output
- → इस Gate में Input Signals Same होने पर Output 1 तथा Input signals अलग-2 होने पर Output 0 आता है।
- → Truth Table ⇒

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

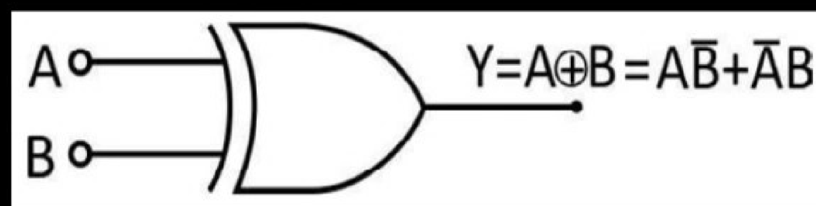


3. Special Purpose Gate ⇒

A. Ex-OR Gate ⇒

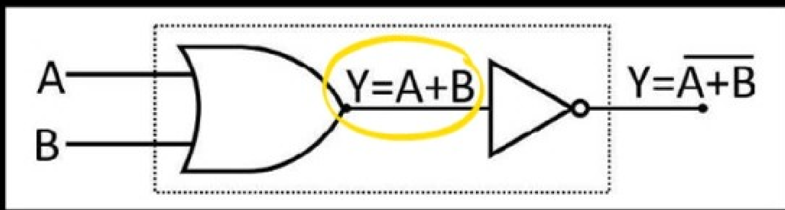
- Exclusive OR या XOR Gate
- 2 Inputs & 1 Output
- इस Gate में Input Signals यदि अलग-2 हो, तो output 1 आता है तथा Same input (0-0 या 1-1) होने पर Output 0 होता है
- Truth Table ⇒

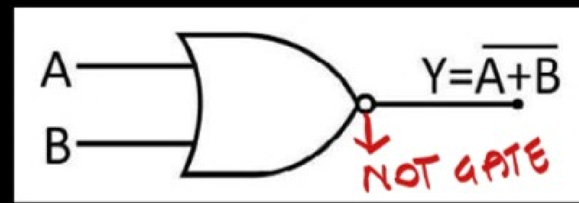
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



- Stair Case Switch

B. NOR Gate ⇒





→ OR + NOT

→ OR Gate के output पर NOT Gate लगाकर बनाते है।

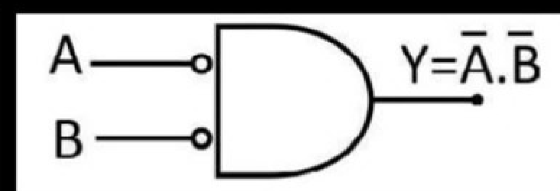
→ OR Gate के output का व्युत्क्रम (Inverse) प्राप्त करने के लिए use.

→ 2 या 2 से अधिक Inputs & 1 Output

→ Truth Table ⇒

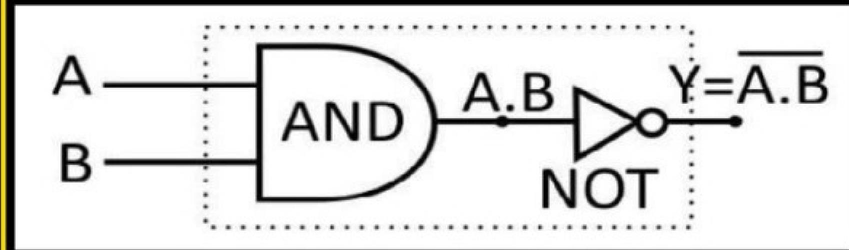
A	B	Y = A+B	Y = $\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

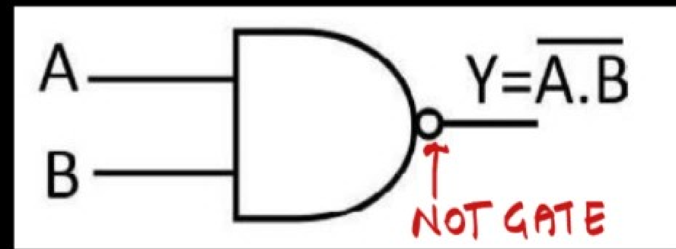
→ इस Gate में सभी Inputs 0 होने पर ही output 1 आता है, अन्यथा 0.



→ Bubble AND Gate = NOR Gate

A. NAND Gate ⇒





→ AND + NOT

→ AND Gate के Output पर NOT Gate लगाकर बनाया जाता है।

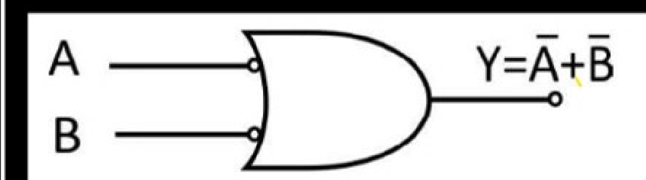
→ AND Gate के output का व्युत्क्रम (Inverse) प्राप्त करने के लिए use.

→ 2 या 2 से अधिक Inputs & 1 output.

→ इसके सभी Inputs 1 होने पर Output 0 होगा अन्यथा 1.

→ Truth Table ⇒

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

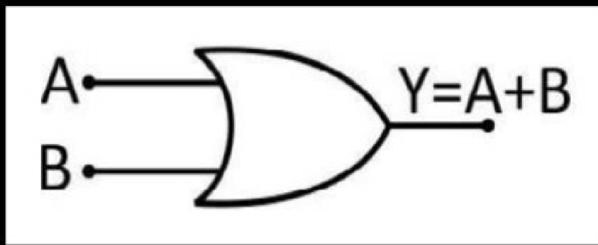


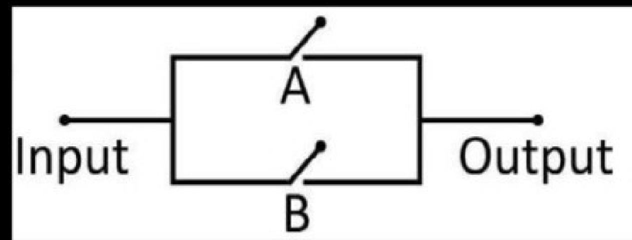
→ Bubble OR Gate = NAND Gate

2. Universal Gates ⇒

- → ऐसे Gates जो Basic Gates के सारे Functions perform करते है।
- → 2 प्रकार
  - A. NAND Gate
  - B. NOR Gate

C. OR Gate ⇒





0 = Low Voltage
1 = High Voltage

→ 2 या 2 से अधिक Input तथा 1 Output

→ इसमें किसी भी 1 Input के 1 होने से Output 1 आ जाता है।

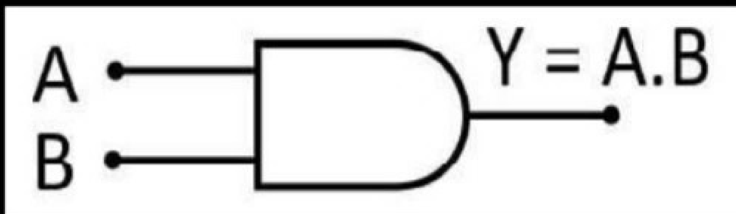
→ इसे (+) Plus Operator से Denote करते है।

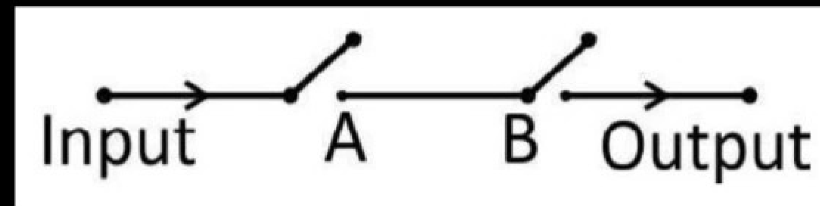
→ Truth Table ⇒

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

AND = श्रेणी क्रम (Serial)
OR = समान्तर क्रम (Parallel)

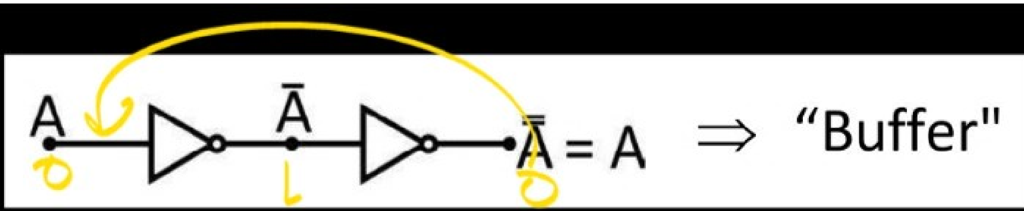
B. AND Gate ⇒



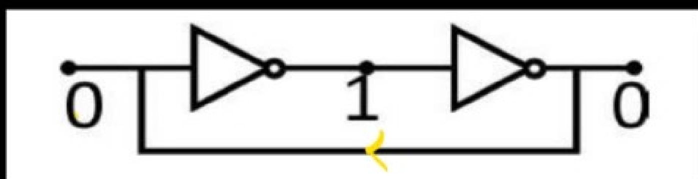


- → 2 या अधिक Inputs & Single Output
- → इसके सभी Inputs 1 होने पर ही Output 1 आता है।
- → इसे (.) Dot Operator से Denote करते हैं।
- → Truth Table ⇒

A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1



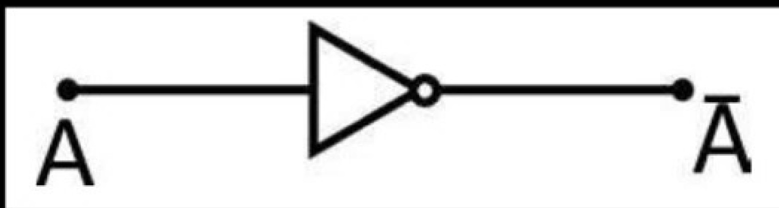
OR



→ Not Gate से बने Buffer के Output को यदि Input से जोड़ दिया जाये, तो यह Bistable Multivibrator की तरह व्यवहार करता है।

1. Basic Gates ⇒

A. NOT Gate ⇒



→ Single Input & Single Output

→ इसका Output हमेशा Input का Inverse/उल्टा होता है। Complimentor | Invertor

→ Truth Table -

A	Ā
0	1
1	0

* Logic Gates ⇒

→ ऐसा Electronic Circuit जो एक या अधिक Input Signal लेकर Output Signal देता है, उसे Logic Gate कहते है।

→ Base of Digital Electronics.

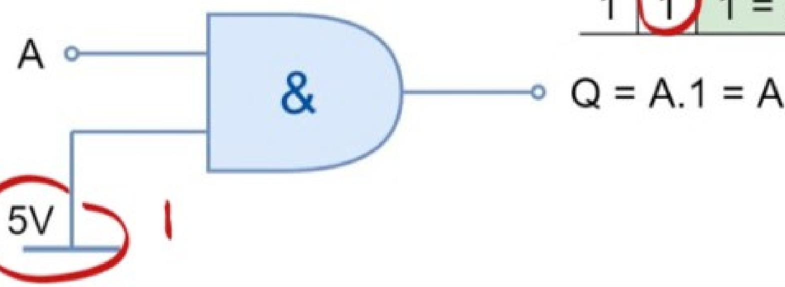
→ 3 प्रकार -

1. Basic Gates = AND, OR, NOT
2. Universal Gates = NAND, NOR
3. Special Purpose Gate = EX-OR, EX-NOR

Identity Law

$A.1 = A$

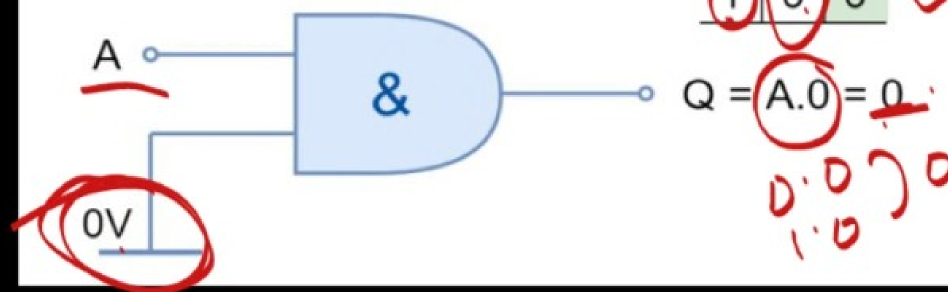
A	B	Q
0	1	0 = A
1	1	1 = A



Annulment Law

$A.0 = 0$

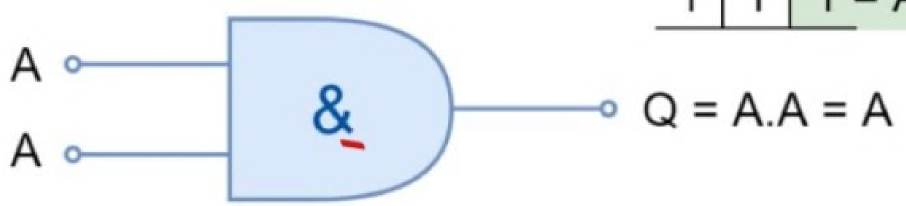
A	B	Q
0	0	0
1	0	0



Idempotent Law

$A.A = A$

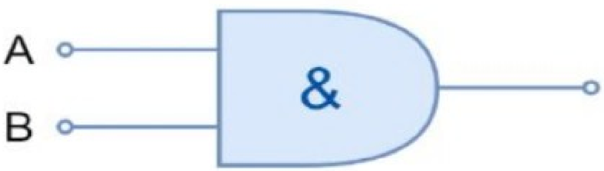
A	A	Q
0	0	0 = A
1	1	1 = A



Commutative Law

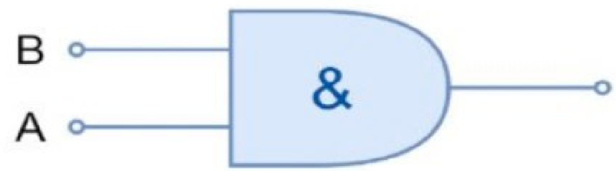
A.B = A.B

A	B	Q	B	A	Q
0	1	0	1	0	0
1	0	0	0	1	0



Q = A.B

$\frac{A \cdot B}{B \ A}$

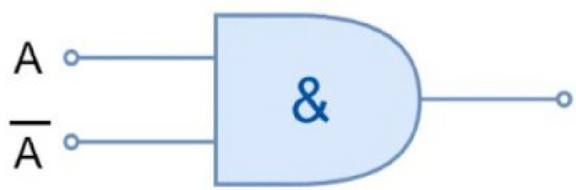


Q = B.A

Complement Law

$A.\overline{A} = 0$

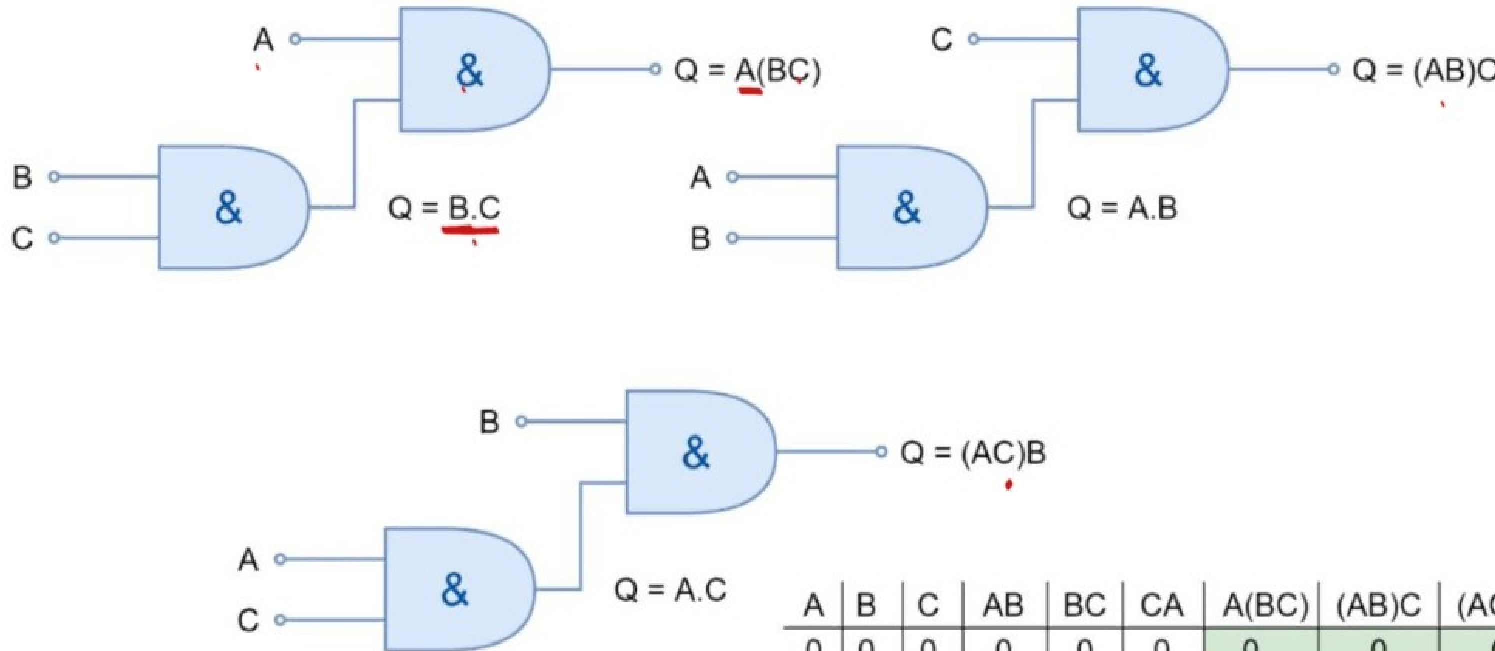
A	$\overline{A}$	Q
0	1	0
1	0	0



$Q = A.\overline{A} = 0$

Associative Law

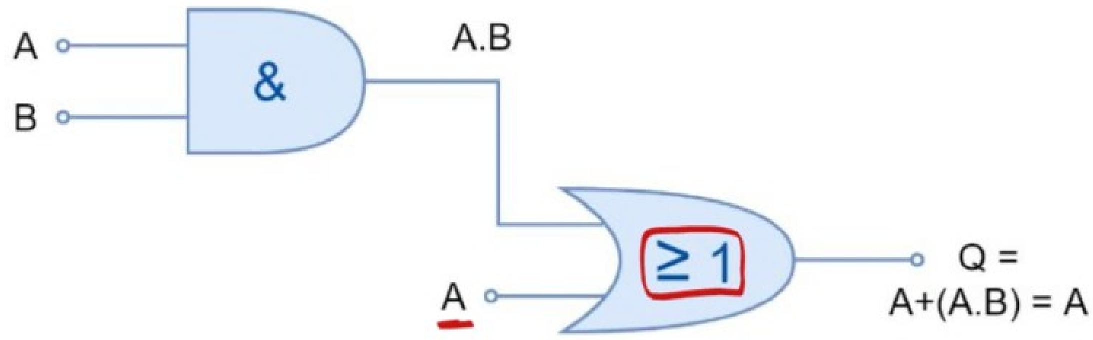
A(BC) = (AB)C = (AC)B = ABC



A	B	C	AB	BC	CA	A(BC)	(AB)C	(AC)B
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1

Absorption Law

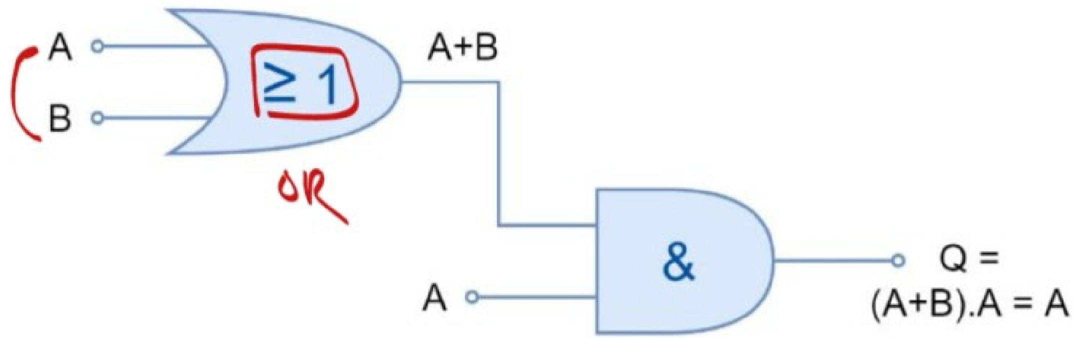
$$A+(A.B) = (A.1) + (A.B) = A(1+B) = A$$



A	B	A.B	Q
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Absorption Law

$$A(A+B) = (A+0).(A+B) = A+(0.B) = A$$



A	B	A+B	Q
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

3. Boolean Algebric Theorems ⇒

Name	AND form	OR form
Identity law	$1.A=A$	$0 + A = A$
Null law	$0.A=0$	$1 + A = 1$
Idempotent law	$A.A=A$	$A + A = A$
Inverse law	$A.\overline{A}=0$	$A + \overline{A} = 1$
Commutative law	$A.B=B.A$	$A + B = B + A$
Associative law	$(AB)C=A(BC)$	$(A+B)+C=A+(B+C)$
Distributive law	$A+BC = (A+B)(A+C)$	$A(B+C)=AB+AC$
Absorption law	$A(A+B) = A$	$A+AB=A$
De Morgan's law	$\overline{A.B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}.\overline{B}$

✲ Important Theorems ⇒

1. Duality ⇒

AND ⟷ OR

0 ⟷ 1

1 ⟷ 0

2. D-Morgan ⇒

First law ⇒ $(A \cdot B)' \longrightarrow A' + B'$ Compliment of Product = Addition of Compliments
(NAND GATE)

Second law ⇒ $(A+B)' \longrightarrow A' \cdot B'$ Compliment of Addition = Product of Compliment
(NOR GATE)

4. Alphanumerical Codes ⇒

- → ऐसा Code जिसमें Alphabet, Numbers तथा Puntuation Marks का use किया जाता है।
  जैसे- ASCII & EBCDIC.
- → Normal ASCII 7 Bit Code होता है, जबकि Extended ASCII 8 Bit Code होता है।
- → ASCII - American Standard Code for Information Interchange
- → EBCDIC - Extended Binary Coded Decimal Interchange Code
- → EBCDIC - 8 Bit Code

Ex-3 (11011) Gray = Binary

MSB	1	1	0	1	1
MSB	1	0	0	1	0

$$\boxed{=(10010)_2}$$

Ex-4 (10111) Gray ⇒ Binary

MSB	1	0	1	1	1
MSB	1	1	0	1	0

$$\boxed{=(11010)_2}$$

Ex-1 $(10010)_2$ = Gray Code

$$
\begin{array}{lccccc}
\text{MSB} & 1 & 0 & 0 & 1 & 0 \\
\text{MSB} & 1 & 1 & 0 & 1 & 1
\end{array}
$$

$$\boxed{= 11011}$$

Ex-2 $(11010)_2$ = Gray Code

$$
\begin{array}{lccccc}
\text{MSB} & 1 & 1 & 0 & 1 & 0 \\
\text{MSB} & 1 & 0 & 1 & 1 & 1
\end{array}
$$

$$\boxed{= 10111}$$

Avoid CARRY

3. Gray Code ⇒

- → Non weighted code
- → Minimum Change Code because इसमें केवल 1 Bit का change होता है।
- → Reflected Code → where Successive Bit has only 1 Bit Difference
- → Unit Distance code
- → Minimum Error Code
- → Ex-OR का use.

Ex-2 (215) का Excess-3 Code ?

$$\begin{array}{ccc} 2 & 1 & 5 \\ +3 & +3 & +3 \\ \hline 5 & 4 & 8 \\ \downarrow & \downarrow & \downarrow \\ 0101 & 0100 & 1000 \end{array}$$

$$\boxed{\text{Excess-3} \Rightarrow 010101001000}$$

Ex-1 $(732)_{10}$ का BCD & Excess-3 Code ज्ञात कीजिए।

$$\begin{array}{ccc} 7 & 3 & 2 \\ \downarrow & \downarrow & \downarrow \\ 0111 & 0011 & 0010 \end{array}$$

$$\boxed{BCD = 011100110010}$$

$$\begin{array}{ccc} {}^{11} & & \\ 0111 & 0011 & 0010 \\ 0011 & 0011 & 0011 \\ \hline 1010 & 0110 & 0101 \end{array}$$

$$\boxed{\text{Excess-3} = 101001100101}$$

$$\begin{array}{ccc} 7 & 3 & 2 \\ +3 & +3 & +3 \\ \hline 10 & 6 & 5 \\ \downarrow & \downarrow & \downarrow \\ 1010 & 0110 & 0101 \end{array}$$

$$\boxed{\text{Excess-3} \Rightarrow 101001100101}$$

2. Excess-3 Code ⇒

- → 4 Bit का Code.
- → Non weighted.
- → इस Code को Binary में 3 Add कर (0011) प्राप्त किया जाता है
- → Self Complimentory Code.

Binary	Excess-3
0 – 0000	0011
1 – 0001	0100
2 – 0010	0101
3 – 0011	0110
4 – 0100	0111
5 – 0101	1000
6 – 0110	1001
7 – 0111	1010
8 – 1000	1011
9 – 1001	1100

$\boxed{+0011}$

$$\left.\begin{array}{l}
0011 \rightarrow \text{Compliment} - 1100\ (0\ \&\ 9) \\
0100 \rightarrow \text{Compliment} \cdot 1011\ (1\ \&\ 8) \\
0101 \rightarrow \text{Compliment} - 1010\ (2\ \&\ 7) \\
0110 \rightarrow \text{Compliment} - 1001\ (3\ \&\ 6) \\
0111 \rightarrow \text{Compliment} = 1000\ (4\ \&\ 5)
\end{array}\right\}\ \text{Self Compliment}$$

☆ Weighted Code V/S Non Weighted Code ⇒

Weighted Code	Non Weighted Code
1. ऐसा Code जिसकी प्रत्येक Digit का एक Fixed weight हो। Ex – BCD, 2421 Code.	1. ऐसा Code जिसकी प्रत्येक Digit का कोई fixed weight नहीं होता है। Ex – Excess-3, Gray Code.

✱ Decimal to BCD Conversion ⇒

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

EX-1 $(943)_{10}$ = BCD

2	943	1
2	471	1
2	235	1
2	117	1
2	58	0
2	29	1
2	14	0
2	7	1
2	3	1
	1 →	1

$(1110101111)_2$ → BINARY

✱✱✱

9	4	3
1001	0100	0011

$\boxed{BCD = 100101000011}$

1. BCD Code ⇒

- → Binary Coded Decimal
- → 8421 Code
- → 4 Bits का Binary Code.
- → इसके द्वारा 0-9 तक की Decimal संख्याओं को Binary में Denote किया जाता है।
- → Weighted Code
- → Self Complimentory Code नहीं।
- → दो Decimal संख्याओं का BCD Addition यदि 9 से कम है, तो वह सही Addition होगा।
- → यदि BCD Addition 9 से अधिक हो जाये तो उसमें 6 (0110) Add कर सही BCD Addition ज्ञात करेंगे।

1. BCD Code ⇒

- → Binary Coded Decimal
- → 8421 Code
- → 4 Bits का Binary Code.
- → इसके द्वारा 0-9 तक की Decimal संख्याओं को Binary में Denote किया जाता है।
- → Weighted Code
- → Self Complimentory Code नहीं।
- → दो Decimal संख्याओं का BCD Addition यदि 9 से कम है, तो वह सही Addition होगा।
- → यदि BCD Addition 9 से अधिक हो जाये तो उसमें 6 (0110) Add कर सही BCD Addition ज्ञात करेंगे।

✰ Binary Codes ⇒

Codes

Binary

1. BCD
2. Excess-3
3. Gray Code

Non-Binary

1. Morse Code

Alphanumerical

1. ASCII (7Bit + 1 Parity)
2. EBCDIL (8Bit)
3. Internal Code (8Bit)

G. 8's Compliment ⇒

→ केवल Octal संख्या |

→ दी गई संख्या को 7 से Minus कर । Add कर दिया जाता है

Ex- 2107 का 8's Compliment ?

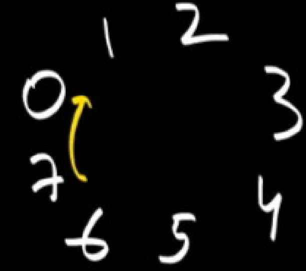
$$\Rightarrow \begin{array}{r} 7777 \\ -\ 2107 \\ \hline 5670 \\ \hline \end{array}$$

$$= \begin{array}{r} 5670 \\ +1 \\ \hline 5671 \\ \hline \end{array}$$

Ex-2 5670 का 8's Compliment ?

$\boxed{0-7}$

$$\begin{array}{r} 7777 \\ -\ 5670 \\ \hline 2107 \\ +1 \\ \hline 2110 \\ \hline \end{array}$$



F. 2's Compliment →

→ किसी भी Binary संख्या के 1's compliment में 1 Add करने से 2's compliment प्राप्त होता है।

Ex = (1001101) का 2's compliment.

$$
\begin{array}{ll}
1001101 & \\
0110010 & \text{1's Compliment} \\
\hline
\end{array}
$$

$$
\begin{array}{rl}
0110010 & \\
+1 & \text{2's Compliment} \\
\hline
\boxed{(0110011)} &
\end{array}
$$

E. 1's Compliment ⇒

→ दी गई Binary संख्या की प्रत्येक Bit को एक से घटाने पर 1's Compliment मिलता है।

OR

→ 0 का 1 & 1 का 0 करने से 1's Compliment मिलता है।

→ Inversion of Number.

Ex - 1001101 का 1's Compliment -

↓

$\boxed{(0110010)}$ ✓

D. Sign Magnitude Representation ⇒

$B_5 \quad \underbrace{B_4 \quad B_3 \quad B_2 \quad B_1}_{\text{Magnitudes}}$

$\downarrow$

Sign

0 — +ve

1 — -ve

C. Binary Multiplication ⇒

Rule ⇒

$$\boxed{\begin{array}{l} 0 \times 0 = 0 \\ 0 \times 1 = 0 \\ 1 \times 0 = 0 \\ 1 \times 1 = 1 \end{array}}$$

64 32 16 8 4 2 1

1000001

$1101 \to 13$

$101 \to 5$

65

Ex-1 101×11

$$\begin{array}{r} 1\;0\;1 \\ \times\;1\;1 \\ \hline 1\;0\;1 \\ 1\;0\;1\;\times \\ \hline 1\;1\;1\;1 \end{array}$$

$101 \text{ — } 5$, $11 \text{ — } 3$, 15

15

$\boxed{= (1111)_2}$

Ex-2 1101×101

$$\begin{array}{r} 1\;1\;0\;1 \\ 1\;0\;1 \\ \hline 1\;1\;0\;1 \\ 0\;0\;0\;0\;\times \\ 1\;1\;0\;1\;\times\;\times \\ \hline 1\;0\;0\;0\;0\;0\;1 \end{array}$$

$\boxed{= (1000001)_2}$

EX-5

111100.11 − 11011.01

$$\begin{array}{r} 111100.11 \\ -\ 011011.01 \\ \hline 100001.10 \end{array}$$

11011.01

16 8 2 1

$$\begin{array}{r} 60.75 \\ 27.25 \\ \hline 33.50 \end{array}$$

33.50

$\boxed{100001.10}$

Ex-3 11011 − 01111

$$
\begin{array}{r}
0\;1\;10\;10\;1\;1 \\
-\;0\;1\;1\;1\;1 \\
\hline
0\;1\;1\;0\;0 \\
\hline
\end{array}
$$

$$\boxed{=(01100)_2}$$

Ex-4 11100.11 − 1011.01

$$
\begin{array}{r}
1\;1\;1\;10\;10.\;11 \\
-\;0\;1\;0\;1\;1.\;01 \\
\hline
1\;0\;0\;0\;1.\;1\;0 \\
\hline
\end{array}
$$

$$\boxed{=(10001.10)}$$

$$
\begin{array}{r}
28.75 \\
11.25 \\
\hline
17.50 \\
\hline
\end{array}
$$

B. Binary Substraction ⇒

Rule ⇒

$$\begin{array}{|c|}\hline 0 - 0 = 0 \\ 1 - 1 = 0 \\ 1 - 0 = 1 \\ 10 - 1 = 1 \\ \hline \text{Next Digit से} \\ 1 \text{ Borrow} \\ \hline \end{array}$$

Ex-1 $(1011)_2 - (101)_2$

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1 \\ - & 0 & 1 & 0 & 1 \\ \hline & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

$$\boxed{= (0110)}$$

$1011 \to 11$

$0101 \to 5$

$\boxed{6 \to 0110}$

Ex-2 $(110101)_2 - (100010)_2$

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

$$\boxed{= (010011)_2}$$

Ex-3   110101 + 100010

$$
\begin{array}{r}
110101 \\
+\ 100010 \\
\hline
1010111 \\
\hline
\end{array}
$$

$$\boxed{= (1010111)_2}$$

Ex-4   1011.01 + 1100.11

$$
\begin{array}{r}
1011.01 \\
+\ 1100.11 \\
\hline
11000.00 \\
\hline
\end{array}
$$

$$\boxed{= (11000.00)_2}$$

Ex-5   111.01 + 1011.11

$$
\begin{array}{r}
111.01 \\
+\ 1011.11 \\
\hline
10011.00 \\
\hline
\end{array}
$$

$$\boxed{= (10011.00)_2}$$

✱ Binary Operations ⇒

A. Binary Addition ⇒

Rule-

$$\boxed{\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0 \text{ (carry 1)} \end{aligned}}$$

Ex-1 $(1011)_2 + (011)_2 = ?$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\ \ 0\,1\,1 \\ \hline 1\,1\,1\,0 \end{array}$$

$$\boxed{= (1110)_2}$$

$1011 \rightarrow 11$

$011 \rightarrow 3$

$\boxed{14 \rightarrow 1110}$ ✓

Ex-2 $(10110)_2 + (11011)_2 = ?$

$$\begin{array}{r} 1\,0\,1\,1\,0 \\ +\ 1\,1\,0\,1\,1 \\ \hline 1\,1\,0\,0\,0\,1 \end{array}$$

$$\boxed{= (110001)_2}$$

24. Octal Fraction to Hexadecimal ⇒

→ Octal to Hexadecimal Concept.

Ex-1 $(1257.5)_8 = (?)_{16}$

$$\begin{array}{ccccc} 1 & 2 & 5 & 7 & . & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow \\ 001 & 010 & 101 & 111 & & 101 \end{array}$$

$$\underbrace{0010}_{2}\,\underbrace{1010}_{10 = A}\,\underbrace{1111}_{15 = F}\,.\,\underbrace{1010}_{10 = A}$$

$$\boxed{= (2AF.A)_{16}}$$

23. Hexadecimal Fraction to Octal ⇒

→ Hexadecimal to Octal Concept.

Octal $2^{3} = 8$

Hexa $2^{4} = 16$

Ex-1 $(2AF.A)_{16} = (?)_8$

2	A	F	.	A
	10	15		10
↓	↓	↓		↓
0010	1010	1111		1010

001 | 010 | 101 | 111 . 101 | 0

↓ ↓ ↓ ↓ ↓

1 2 5 7 5

$$\boxed{=(1257.5)_8}$$

22. Octal to Hexadecimal ⇒

→ दी गई Octal संख्या के Equivalant 3-3 Bits की Binary निकालकर उसे Right to left 4-4 Digits का Pair बनाकर Hexadecimal संख्या प्राप्त करते है।

Ex-1 $(4332)_8 = (?)_{16}$

4	3	3	2
↓	↓	↓	↓
100	011	011	010

←

1000 | 1101 | 1010

↓ ↓ ↓

8 | 13 = D | 10 = A

$\boxed{=(8DA)_{16}}$

Ex-2 $(377)_8 = (?)_{16}$

3	7	7
↓	↓	↓
011	111	111

←

0000 | 1111 | 1111

↓ ↓ ↓

0 | 15 = F | 15 = F

$\boxed{=(FF)_{16}}$

21. Hexadecimal to Octal →

→ दी गई Hexadecimal संख्या के Equivalant 4-4 Bits को Binary निकालकर उसके Right to left 3-3 Digit के Pair बनाकर उसके समकक्ष Octal Digits लिख दिये जाते है।

Ex-1 $(8DA)_{16} = (?)_8$

$8 \rightarrow 1000$

$D = 13 \rightarrow 1101$

$A = 10 \rightarrow 1010$

$\leftarrow$ 100011011010

$100 \; 011 \; 011 \; 010$

$4 \quad 3 \quad 3 \quad 2$

$\boxed{= (4332)_8}$

Ex-2 $(FF)_{16} = (?)_8$

$F = 15 \rightarrow 1111$

$F = 15 \rightarrow 1111$

$\leftarrow$ 11111111

$011 \; 111 \; 111$

$3 \quad 7 \quad 7$

$\boxed{= (377)_8}$

20. Decimal Fraction to Hexadecimal ⇒

→ Decimal to Hexadecimal Same Concept.

→ Decimal fraction Part को 16 से multiply कर Integer को अलग कर Top to Bottom Read करते है

Ex-1 $(2026.625)_{10} = (?)_{16}$

16	2026	10
16	126	14
	7	→ 7

$= 7\ \underset{E}{\underline{14}}\ \underset{A}{\underline{10}}$

$= (7EA)_{16}$

$0.625 \times 16 = 10.\underline{000}$ → $\underset{A}{\underline{10}}$

$$\boxed{(2026.625)_{10} = (7EA.A)_{16}}$$

19. Hexadecimal Fraction to Decimal ⇒

→ Hexadecimal to Decimal Same Concept

→ Hexadecimal Fraction Digits को Left to Right $16^{-1}, 16^{-2} \ldots$ से Multiply कर Add कर लेता है।

Ex-1 $(7EA.A)_{16} = (?)_{10}$

7	+	E (14)	+	A (10)	.	A (10)
16^2 = 256		16^1 = 16		16^0 = 1		16^{-1} = 0.0625

$$1792 + 224 + 10 \,.\, 0.625$$

$$\boxed{= (2026.625)_{10}}$$

18. Decimal to Hexadecimal ⇒

→ दी गई Decimal संख्या में 16 का भाग देकर Remainder को नीचे से ऊपर Read कर Equivalant Hexadecimal संख्या ज्ञात की जाती है।

Ex-1 $(1449)_{10} = (?)_{16}$

16	1449	9
16	90	10
	5	→ 5

5 $\underset{A}{\underline{10}}$ 9

$\boxed{=(5A9)_{16}}$

Ex-2 $(2046)_{10} = (?)_{16}$

16	2046	14
16	127	15
	7	→ 7

$= 7\ \underset{F}{\underline{15}}\ \underset{E}{\underline{14}}$

$\boxed{=(7FE)_{16}}$

17. Hexadecimal to Decimal ⇒

→ दी गई Hexadecimal संख्या के प्रत्येक Digit को Right से Left की ओर $16^0, 16^1 \ldots$ से Multiply कर प्राप्त संख्याओं को Add कर दिया जाता है।

Ex-1 $(5A9)_{16} = (?)_{10}$

$$5 + \underset{10}{A} + 9$$

$$\underset{256}{16^2} \quad \underset{16}{16^1} \quad \underset{1}{16^0}$$

$$1280 + 160 + 9$$

$$\boxed{= (1449)_{10}}$$

Ex-2 $(7FE)_{16} = (?)_{10}$

$$7 + \underset{15}{F} + \underset{14}{E}$$

$$\underset{256}{16^2} \quad \underset{16}{16^1} \quad \underset{1}{16^0}$$

$$1792 + 240 + 14$$

$$\boxed{= (2046)_{10}}$$

16. Binary Fraction to Hexadecimal →

→ Binary integers को Right to left 4-4 Bits के Pair बनाकर उनके Equivalant Hexadecimal Digit लिख दिया जाता है, तथा last Pair में 4 Bits नहीं होने पर Additional 0 लिख देते है।

→ Binary float Digits को left to Right 4-4 Digits के Pair बनाकर उनके Equivalant Hexadecimal Digit लिख दी जाती है, यदि 4 Digits नहीं है तो Right मे Additional 0 लिखते है।

Ex-1 $(101110010.11)_2 = (?)_{16}$

$$\underbrace{1}_{0001 \to 1}\ \underbrace{0111}_{7}\ \underbrace{0010}_{2}\ .\ \underbrace{11}_{1100 \to 12 \to C}$$

$$\boxed{= (172.C)_{16}}$$

15. Hexadecimal Fraction to Binary ⇒

→ Hexadecimal to Binary Concept.

→ दशमलव के बाद भी Hexadecimal Digit के समकक्ष 4 Bit Binary लिखते है

<u>Ex-1</u> $(172.5)_{16} = (?)_2$

$$
\begin{array}{cccccc}
1 & 7 & 2 & . & 5 \\
\downarrow & \downarrow & \downarrow & & \downarrow \\
0001 & 0111 & 0010 & . & 0101
\end{array}
$$

$$\boxed{= (101110010.0101)_2}$$

14. Binary to Hexadecimal ⇒

→ दी गई Binary संख्या के Right to left 4-4 Bits के Pair बनाकर उनके Equivalant Hexadecimal Digit लिख दी जाती है।

→ यदि Last Pair 4 Digits का नहीं बनता है, तो multiple 0 Add कर लिए जाते है।

Ex-1 $(11110100010)_2 = (?)_{16}$

$$\underbrace{111}\ \underbrace{1010}\ \underbrace{0010}$$

$$\begin{array}{ccc} 0111 & 1010 & 0010 \\ \downarrow & \downarrow & \downarrow \\ 7 & \frac{10}{A} & 2 \end{array}$$

$$\boxed{= (7A2)_{16}}$$

Ex-2 $(110101111)_2 = (?)_{16}$

$$\underline{1}\ \underline{1010}\ \underline{1111}$$

$$\begin{array}{ccc} 0001 & 1010 & 1111 \\ \downarrow & \downarrow & \downarrow \\ 1 & \frac{10}{A} & \frac{15}{F} \end{array}$$

$$\boxed{= (1AF)_{16}}$$

<u>EX-1</u> $(7A2)_{16} = (?)_2$

$$\begin{array}{ccc} 7 & \underline{A} & 2 \\ & 10 & \\ \downarrow & \downarrow & \downarrow \\ 0111 & 1010 & 0010 \end{array}$$

$$\boxed{= (11110100010)_2}$$

<u>EX-2</u> $(1AF)_{16} = (?)_2$

$$\begin{array}{ccc} 1 & \underline{A} & \underline{F} \\ & 10 & 15 \\ \downarrow & \downarrow & \downarrow \\ 0001 & 1010 & 1111 \end{array}$$

$$\boxed{= (110101111)_2}$$

13. Hexadecimal to Binary ⇒

→ दी गई Hexadecimal संख्या के प्रत्येक Digit के समकक्ष 4 Bit की Binary संख्या लिखते हैं।

Hexadecimal Digit	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10 (A)	1010

Hexa Digit	Binary
11 (B)	1011
12 (C)	1100
13 (D)	1101
14 (E)	1110
15 (F)	1111

12. Decimal Fraction to Octal ⇒

→ Decimal to Octal Same Concept

→ दशमलव के बाद की Decimal संख्या को 8 से multiply करेगे तथा Integer को अलग लेकर ऊपर से नीचे की ओर Read करेगे।

Ex-1 $(167.6875)_{10} = (?)_8$

8	167	7
8	20	4
	2	2

$= (247)_8$

$0.6875 \times 8 = 5.5000 \rightarrow 5$

$0.5000 \times 8 = 4.0000 \rightarrow 4$

$= (0.54)_8$

$= (167.6875)_{10} = (247.54)_8$

11. Octal Fraction to Decimal ⇒

→ Octal to Decimal वाला Same Concept

→ दशमलव के बाद वाली संख्या को Left to Right $8^{-1}, 8^{-2} \ldots$ से Multiply कर प्राप्त Digits का योग करेंगे।

$8^{-1} = 0.125$

$8^{-2} = 0.015625$

$8^{-3} = 0.001953125$

Ex-1 $(247.54)_8 = (?)_{10}$

$$\begin{array}{ccccccc} 2 & + 4 & + 7 & . & 5 & + 4 \\ 8^2 & 8^1 & 8^0 & & 8^{-1} & 8^{-2} \\ \hline 64 & 8 & 1 & & 0.125 & 0.015625 \end{array}$$

$128 + 32 + 7 \;.\; 0.625 + 0.0625$

$\boxed{(167.6875)_{10}}$

10. Decimal to Octal ⇒

→ दी गई Decimal संख्या में 8 का भाग देकर Remainder को नीचे से ऊपर Read किया जाता है।

Ex-1 $(3859)_{10} = (?)_8$

8	3859	3
8	482	2
8	60	4
	7 →	7

$= (7423)_8$

Ex-2 $(175)_{10} = (?)_8$

8	175	7
8	21	5
	2 →	2

$= (257)_8$

9. Octal to Decimal →

→ दिये गये Octal Number की प्रत्येक Digit को Right से Left की ओर $8^0, 8^1, 8^2 \ldots$ से Multiply कर प्राप्त संख्याओं को Add कर लेते है।

Ex-1 $(7423)_8 = (?)_{10}$

$$\begin{array}{cccc} 7 & + 4 & + 2 & + 3 \\ 8^3 & 8^2 & 8^1 & 8^0 \\ 512 & 64 & 8 & 1 \\ \hline \end{array}$$

$3584 + 256 + 16 + 3$

$\boxed{= (3859)_{10}}$

Ex-2 $(257)_8 = (?)_{10}$

$$\begin{array}{ccc} 2 & + 5 & + 7 \\ 8^2 & 8^1 & 8^0 \\ 64 & 8 & 1 \\ \hline \end{array}$$

$128 + 40 + 7$

$\boxed{= (175)_{10}}$

Ex-2 $(1001101.1001)_2 = (?)_8$

$$\underline{1}\;\underline{001}\;\underline{101}\;.\;\underline{100}\;\underline{1}$$

$$\begin{array}{ccccccc} 001 & 001 & 101 & . & 100 & 100 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 1 & 1 & 5 & & 4 & 4 \end{array}$$

$$\boxed{= (115.44)_8}$$

8. Binary Fraction to Octal ⇒

- Same method as Binary to Octal.
- दशमलव के बाद की संख्या Left से Right की ओर Read कर 3-3 Digits के Pair बनाकर Octal Digit लिखेंगे।
- यदि सबसे Last में 3 Digit पुरे नहीं हो, तो Extra 0 Add कर दिये जाते है।

Ex-1 $(10101111001.0111)_2 = (?)_8$

$10101111001 . 0111$

$010 \quad 101 \quad 111 \quad 001 \quad . \quad 011 \quad 100$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \quad \downarrow \quad \downarrow$

$2 \quad 5 \quad 7 \quad 1 \quad . \quad 3 \quad 4$

$$\boxed{= (2571.34)_8}$$

7. Octal Fraction to Binary ⇒

→ Same method as Octal to Binary.

Ex-1 $(165.27)_8 = (?)_2$

$$\begin{array}{cccccc} 1 & 6 & 5 & . & 2 & 7 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 001 & 110 & 101 & & 010 & 111 \end{array}$$

$$\boxed{= (1110101 \cdot 010111)_2}$$

Ex-2. $(267.15)_8 = (?)_2$

$$\begin{array}{cccccc} 2 & 6 & 7 & . & 1 & 5 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 010 & 110 & 111 & & 001 & 101 \end{array}$$

$$\boxed{= (10110111 \cdot 001101)_2}$$

6. Binary to Octal ⇒

→ दी गई Binary संख्या को Right से Left की ओर Read कर 3-3 Digits के Pair बनाकर उसके समकक्ष Octal Digit लिख देते है।

→ यदि last में 3 Digit का Pair नहीं बनता है, तो उतने ही 0 Add कर देते है।

Ex-1 $(101011)_2 = (?)_8$

101011

101	011
↓	↓
5	3

$\boxed{=(53)_8}$

Ex-2 $(1011001)_2 = (?)_8$

1011001

001	011	001
↓	↓	↓
1	3	1

$\boxed{=(131)_8}$

Ex-1 $(53)_8 = (?)_2$

$$\begin{array}{cc} 5 & 3 \\ \downarrow & \downarrow \\ 101 & 011 \end{array}$$

$$\boxed{= (101011)_2}$$

Ex-2 $(642)_8 = (?)_2$

$$\begin{array}{ccc} 6 & 4 & 2 \\ \downarrow & \downarrow & \downarrow \\ 110 & 100 & 010 \end{array}$$

$$\boxed{= (110100010)_2}$$

5. Octal to Binary ⇒

→ प्रत्येक Octal Digit के समकक्ष 3 Bit की Binary लिखते है।

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

1. $(125.525)_{10} = (?)_2$

64 32 16 8 4 2 1

1 1 1 1 1 0 1

$$\begin{array}{r} 64 \\ 32 \\ \hline 96 \\ 16 \\ \hline 112 \\ 8 \\ \hline 120 \\ 4 \end{array}$$

$0.525 \times 2 = 1.050$ 1

$0.050 \times 2 = 0.100$ 0

$0.100 \times 2 = 0.200$ 0

$0.200 \times 2 = 0.400$ 0

$0.400 \times 2 = 0.800$ 0

$0.800 \times 2 = 1.600$ 1

$0.600 \times 2 = 1.200$ 1

1000011

Recursive

Ans

1. 1111101.1000011

2. 189

3. 8.0625

4. 1011110111

$(10111101)_2 =$

128 $\cancel{64}$ 32 16 8 4 $\cancel{2}$ 1

9

1. $(125.525)_{10} = (?)_2$

2. $(10111101)_2 = (?)_{10}$

3. $(1000.0001)_2 = (?)_{10}$

4. $(759)_{10} = (?)_2$

Ex - 2 $(111011 \cdot 1101)_2 = (?)_{10}$

$$1+1+1+\cancel{0}+1+1$$
$$32 \quad 16 \quad 8 \quad \cancel{4} \quad 2 \quad 1$$

$$32+16+8+2+1 \Rightarrow (59)_{10}$$

$$1 + 1 + \cancel{0} + 1$$
$$0.50 + 0.25 + \cancel{0.125} + 0.0625$$

$$0.50+0.25+0.625 = (0.8125)_{10}$$

$$\boxed{=(59.8125)_{10}}$$

4. Binary Fraction to Decimal ⇒

→ दी गई Binary संख्या को Left से Right $2^{-1}, 2^{-2}$ ..... से Multiply कर सभी को Add कर लेते हैं।

Ex.1 $(0.10101)_2 = (?)_{10}$

$$1 + 0 + 1 + 0 + 1$$

$$0.50 + \cancel{0.25} + 0.125 + \cancel{0.0625} + 0.3125$$

$$0.50 + 0.125 + 0.03125 = \boxed{(0.65625)_{10}}$$

$$\boxed{\begin{aligned} 2^{-1} &= 0.50 \\ 2^{-2} &= 0.25 \\ 2^{-3} &= 0.125 \\ 2^{-4} &= 0.0625 \\ 2^{-5} &= 0.03125 \end{aligned}}$$

3. Binary to Decimal ⇒

→ दी गई Binary संख्या को LSB से MSB की ओर $2^0, 2^1, 2^2 \ldots$ से Multiply कर Add करते है।

Ex-1 $(11001)_2 = (?)_{10}$

$$
\begin{array}{ccccccccc}
1 & + & 1 & + & \cancel{0} & + & \cancel{0} & + & 1 \\
2^4 & & 2^3 & & \cancel{2^2} & & \cancel{2^1} & & 2^0 \\
16 & & 8 & & \cancel{4} & & \cancel{2} & & 1
\end{array}
$$

$16+8+1 = \boxed{(25)_{10}}$

Ex-2 $(1111101)_2 = (?)_{10}$

$$
\begin{array}{ccccccccccccc}
1 & + & 1 & + & 1 & + & 1 & + & 1 & + & \cancel{0} & + & 1 \\
64 & & 32 & & 16 & & 8 & & 4 & & \cancel{2} & & 1
\end{array}
$$

$64+32+16+8+4+1 = \boxed{(125)_{10}}$

Ex-2 $(59.8125)_{10} = (?)_2$

Step-1

2	59	1
2	29	1
2	14	0
2	7	1
2	3	1
	1	1

$= (111011)_2$

OR

32	16	8	4	2	1
(1	1	1	0	1	1)$_2$

Step-2 $(0.8125)_{10} = (?)_2$

$\rightarrow 0.8125 \times 2 = 1.6250 = 1$

$\rightarrow 0.6250 \times 2 = 1.2500 = 1$

$\rightarrow 0.2500 \times 2 = 0.5000 = 0$

$\rightarrow 0.5000 \times 2 = 1.0000 = 1$

$= (0.1101)_2$

$(59.8125)_{10} =$

$(111011.1101)_2$ ✓

2. Decimal Fraction to Binary ⇒

- Decimal संख्या को 2 से Multiply करते हैं तथा प्राप्त integer का अलग से लिखकर ऊपर से नीचे की ओर Read करते हैं।
- Multiplication will carry on till number becomes 0.

Ex. 1 $(0.65625)_{10} = (?)_2$

$\rightarrow 0.65625 \times 2 = 1.31250 = 1$

$\rightarrow 0.31250 \times 2 = 0.62500 = 0$

$\rightarrow 0.62500 \times 2 = 1.25000 = 1$

$\rightarrow 0.25000 \times 2 = 0.50000 = 0$

$\rightarrow 0.50000 \times 2 = 1.00000 = 1$

$$\boxed{= (0.10101)_2}$$

→ Decimal को Binary में Convert करने के लिए $2^0, 2^1 \cdots$ का Use भी किया जा सकता है।

1024	512	256	128	64	32	16	8	4	2	1

Ex = $(25)_{10} = (?)_2$

$$\begin{array}{ccccc} 16 + & 8 + & 4 + & 2 + & 1 \\ (1 & 1 & 0 & 0 & 1)_2 \end{array}$$

$(125)_{10} = (?)_2$

$$\begin{array}{ccccccc} 64 + & 32 + & 16 + & 8 + & 4 + & 2 + & 1 \\ (1 & 1 & 1 & 1 & 1 & 0 & 1)_2 \end{array}$$

✲ Number System Conversion ⇒

1. Decimal to Binary ⇒

→ दी गई Decimal संख्या में 2 का भाग देकर Remainder को नीचे से ऊपर Read कर Binary संख्या प्राप्त की जाती है।

Ex-1 $(25)_{10} = (?)_2$

2	25	1
2	12	0
2	6	0
2	3	1
	1 →	1

↑

$$= (11001)_2$$

Ex-2 $(125)_{10} = (?)_2$

2	125	1
2	62	0
2	31	1
2	15	1
2	7	1
2	3	1
	1 →	1

↑

$$= (1111101)_2$$

✱ Base & their groups of Number System ⇒

Number System	Base	Base set
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexa-Decimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 - A, 11 - B, 12 - C, 13 - D, 14 - E, 15 - F

Number System	Min & Max No. of 4 Digit
Binary	Min = 0000 (Common in All) Max = 1111
Octal	Max = 7777
Decimal	Max = 9999
Hexa Decimal	Max = FFFF

C. Decimal Number System ⇒

- Base - 10
- Total 10 Digits (0 - 9) are used.
- इस Number System में LSB से MSB की ओर जाने से प्रत्येक Digit 10 गुना बढ़ती है।

D. HexaDecimal Number System ⇒

- Base - 16
- Total 16 Digits are used.
- Alphanumerical Number System क्योंकि इसमें 0 - 9 तक Digits तथा A-F तक Alphabet का Use होता है।

  जैसे- 10 - A   12 - C   14 - E
  11 - B   13 - D   15 - F
- Method to group 4 Binary Digits.

जैसे -

6 5 4 3 2 1

6 → MSB

1 → LSB

A. Binary Number System ⇒

- Base - 2
- Only 2 Digits (0 & 1) are used.
- Digital Computers work on Binary only.

B. Octal Number System ⇒

- Base - 8
- 8 Digits (0-7) are used.
- Method to group 3 Binary Digits.

"Digital Electronics"

✲ Digital Number System ⇒

→ Digital System द्वारा Calculations करने तथा Numbers को Represent करने में use.

→ 4 Types ⇒ A. Binary Number System
B. Octal Number System
C. Decimal Number System
D. HexaDecimal Number System

→ Number System को Represent करने के लिए $(N)_b$ का use किया जाता है।
N = Number
b = Base / Radix

→ किसी भी Number की Rightmost Digit = LSB (Least Significant Bit / Digit) तथा Leftmost Digit = MSB (Most Significant Bit / Digit) कहते हैं।