Boolean Algebra

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

$= A+B$

OR Operation

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

AND Operation

$0 \cdot 0$	0
$0 \cdot 1$	0
$1 \cdot 0$	0
$1 \cdot 1$	1

- $A+\bar{A}=1$
- $A+1=1$
- $A+0=A$
- $A+A+A+A+\ldots$ n terms $= A$

- $A \cdot 1 = A$
- $A \cdot 0 = 0$
- $A \cdot \bar{A} = 0$
- $A \cdot A \cdot A \ldots$ n terms $= A$

Sum of products

- In SOP, the logic circuit having logic '1' output is considered.
- In SOP, each term is known as minimal term (MIN TERM)
- In min term, all variables should be present in the form of product
- In min term, logic '1' will be represented by simple variable, but logic '0' will be represented by complemented variable

SOP & POS

SOP	POS	A	B	Y
$\bar{A}\cdot\bar{B}$ (m_0)	$A+B$ (M_0)	0	0	0
$\bar{A}\cdot B$ (m_1)	$A+\bar{B}$ (M_1)	0	1	1
$A\cdot\bar{B}$ (m_2)	$\bar{A}+B$ (M_2)	1	0	1
$\bar{A}\cdot\bar{A}$ (m_3)	$\bar{A}+\bar{B}$ (M_3)	1	1	0

minterm maxterm

SOP $\xrightarrow{0} A$, $\xrightarrow{1} \bar{A}$ } Product

POS $\xrightarrow{0} A$, $\xrightarrow{1} \bar{A}$ } Sum

In SOP, $Y = m_1 + m_2 = \sum m(1,2) = \bar{A}B + A\bar{B}$

In POS, $Y = M_0 \cdot M_3 = \prod M(0,3) = (A+B)(\bar{A}+\bar{B}) = \bar{A}B + A\bar{B}$

Product of sum

- In POS, the logic circuit having logic '0' is considered
- In POS, each term is known as maximal term (MAX TERM)
- In Max term, all variables should be present in the form of sum.
- In Max term, logic '0' will be represented by simple variable but logic '1' will be represented by complemented variable.

SOP	POS	A	B	C	Y
$\bar{A}\cdot\bar{B}\cdot\bar{C}$ (m_0)	$A+B+C$ (M_0)	0	0	0	0
$\bar{A}\cdot\bar{B}\cdot C$ (m_1)	$A+B+\bar{C}$ (M_1)	0	0	1	1
$\bar{A}\cdot B\bar{C}$ (m_2)	$A+\bar{B}+C$ (M_2)	0	1	0	1
$\bar{A}\cdot B\cdot C$ (m_3)	$A+\bar{B}+\bar{C}$ (M_3)	0	1	1	0
$A\cdot\bar{B}\cdot\bar{C}$ (m_4)	$\bar{A}+B+C$ (M_4)	1	0	0	1
$A\cdot\bar{B}\cdot C$ (m_5)	$\bar{A}+B+\bar{C}$ (M_5)	1	0	1	0
$A\cdot B\cdot\bar{C}$ (m_6)	$\bar{A}+\bar{B}+C$ (M_6)	1	1	0	0
$A\cdot B\cdot C$ (m_7)	$\bar{A}+\bar{B}+\bar{C}$ (M_7)	1	1	1	1

SOP $Y = m_1 + m_2 + m_4 + m_7 = \sum m(1,2,4,7) = A \oplus B \oplus C$

POS $Y = M_0 \cdot M_3 \cdot M_5 \cdot M_6 = \prod M(0,3,5,6) = A \oplus B \oplus C$

Q $f(A,B) = \sum m(1,2)$

$f(A,B) = \prod M(0,3)$

$\overline{f(A,B)} = \sum m(0,3)$

$\overline{f(A,B)} = \prod M(1,2)$

A	B	$f(A,B)$	$\overline{f(A,B)}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Q $f(A,B,C) = \sum m(1,2)$

$f(A,B,C) = \prod M(0,3,4,5,6,7)$

$\overline{f(A,B,C)} = \sum m(0,3,4,5,6,7)$

$\overline{f(A,B,C)} = \prod M(1,2)$

Q $f(A,B,C) = \sum m(0,1,2,6)$

$f(A,B,C) = \prod M(3,4,5,7)$

$\overline{f(A,B,C)} = \sum m(3,4,5,7)$

$\overline{f(A,B,C)} = \prod M(0,1,2,6)$

Q $f(A,B,C) = \sum m(0,1,3,6,7)$

$f(A,B,C) = \prod M(0,2,4,5,6)$

A	B	C	f(A,B,C)
0	0	0	(10) → X
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	(10) → X
1	1	1	1

$f(A,B,C) = \sum m(1,3,7) + \sum d(0,6)$

$f(A,B,C) = \prod M(2,4,5) \cdot \prod d(0,6)$

■ Standard form of SOP and POS

① $A+\bar{B}C$

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

(M1) $Y = A\cdot 1\cdot 1 + 1\cdot\bar{B}C$

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

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

$= \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(5)\\110}}{AB\bar{C}} + \underset{\substack{(6)\\101}}{A\bar{B}C} + \underset{\substack{(4)\\100}}{A\bar{B}\bar{C}} + \underset{\substack{(5)\\101}}{A\bar{B}C} + \underset{\substack{(1)\\001}}{\bar{A}\bar{B}C}$

$Y = m_1 + m_4 + m_5 + m_6 + m_7$

$= \sum m(1,4,5,6,7)$

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

$= \prod M(0,2,3)$

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

② $A\bar{B}+AC+BC$

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

$Y = A\bar{B}\cdot 1 + A\cdot 1\cdot C + 1\cdot B\cdot C$

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

$= \underset{\substack{(5)\\101}}{A\bar{B}C} + \underset{\substack{(4)\\100}}{A\bar{B}\bar{C}} + \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(5)\\101}}{A\bar{B}C} + \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(3)\\011}}{\bar{A}BC}$

$Y = m_3 + m_4 + m_5 + m_7$

$= \sum m(3,4,5,7)$

$= \prod M(0,1,2,6)$

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

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

③ $PQ + Q\bar{R} + P\bar{R}$

$Y = PQ\cdot 1 + Q\bar{R} + P\bar{R}$

$= P\cdot Q(R+\bar{R}) + (P+\bar{P})Q\bar{R} + P(Q+\bar{Q})\bar{R}$

$= \underset{\substack{(7)\\111}}{PQR} + \underset{\substack{(6)\\110}}{PQ\bar{R}} + \underset{\substack{(6)\\110}}{PQ\bar{R}} + \underset{\substack{(2)\\010}}{\bar{P}Q\bar{R}} + \underset{\substack{(6)\\110}}{PQ\bar{R}} + \underset{\substack{(4)\\100}}{P\bar{Q}\bar{R}}$

$= \Sigma m(2,4,6,7)$

$= m_2 + m_4 + m_6 + m_7$

$\Pi M(0,1,3,5)$

$= M_0 \cdot M_1 \cdot M_3 \cdot M_5$

$(P+Q+R)\cdot(P+Q+\bar{R})\cdot(P+\bar{Q}+\bar{R})\cdot(\bar{P}+Q+\bar{R})$

④ $(A+B)(A+C)$

$Y = AA + AC + AB + BC$

$Y = AB + BC + AC = AB\cdot 1 + 1\cdot BC + A\cdot 1\cdot C$

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

$= \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(6)\\110}}{AB\bar{C}} + \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(3)\\011}}{\bar{A}BC} + \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(5)\\101}}{A\bar{B}C}$

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

$= m_3 + m_5 + m_6 + m_7$

$\Pi M(0,1,2,4)$

$= M_0 \cdot M_1 \cdot M_2 \cdot M_4$

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

⑤ $(A)\cdot(\bar{B}+C)$

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

$Y = A\bar{B}\cdot 1 + A\cdot 1\cdot C$

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

$= \underset{\substack{(5)\\101}}{A\bar{B}C} + \underset{\substack{(4)\\100}}{A\bar{B}\bar{C}} + \underset{\substack{(7)\\111}}{ABC} + \underset{\substack{(5)\\101}}{A\bar{B}C}$

$Y = \Sigma m(4,5,7) = m_4 + m_5 + m_7$

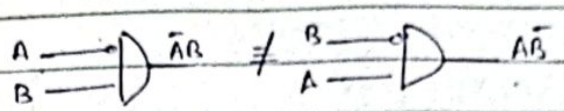
$Y = \Pi M(0,1,2,3,6)$

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

- **Commutative law**

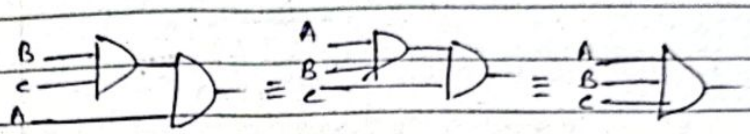
$AB = BA$

$A+B = B+A$



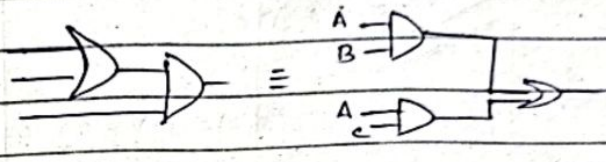
- **Associative law**

$A(BC) = (AB)C = ABC$



- **Distributive law**

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



- **Idempotent law** $A \cdot \bar{A} = 0$

- **Distributive Theorem** $(A+B)(A+C) = A + BC$

(POS) (SOP)

(POS)
$(A+B)(A+C)$
$A \cdot A + A \cdot C + AB + B \cdot C$
$A(1+B+C) + BC$
$A \cdot 1 + BC$
$A + BC$
(SOP)

- **Consensus Theorem** $AB + BC + B\bar{C} = AC + B\bar{C}$

- **Transposition Theorem** $(A+B)(\bar{A}+\bar{B}) = A\bar{B} + \bar{A}B$

- **Absorption Theorem** $(A + \bar{A}B) = A + B$

- **De-Morgan's Theorem** $\overline{(X+Y+Z+W)} = \bar{X} \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W}$

$\overline{X \cdot Y \cdot Z \cdot W} = \bar{X} + \bar{Y} + \bar{Z} + \bar{W}$

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

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

$= (0 + \bar{B}(A+\bar{A}) + \bar{B}\bar{B})(0 + B(A+\bar{A}) + BB)$

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

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

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

$= (\bar{B} + 0)(B + 0)$

$= \bar{B} \cdot B$

$= 0$

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

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

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

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

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

$= B + \bar{B}$

$= 1 = 0$

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

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

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

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

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

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

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

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

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

$= A+BC$

Q $A+\bar{A}B$

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

$= 1(A+B)$

$= A+B$

$\underset{1}{(A+B)}\underset{2}{(A+C)} = \underset{1}{A}+\underset{2}{B}\underset{3}{C}$

Q $\bar{A}+AB$

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

$= \bar{A}+B$

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

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

$= AB+C$

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

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

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

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

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

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

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

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

$= B+\bar{A}$

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

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

$= B + A\bar{B}$

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

$= B+A$

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

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

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

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

$= (A+B)C$

$= AC + BC$

- Concensus theorem
  - Three variables should be present
  - Combination of two variables should be taken in the form of SOP and POS form
  - Only one variable will be in the form of complemented and uncomplemented form.
  - Then reduce the term which has not complemented and uncomplemented form of variable.

① A, B, C	P, Q, R
② $AB + BC + AC$	$(P+Q)(Q+R)(P+R)$
③ $AB + BC + A\bar{C}$ → Redudant form	$(P+Q)(Q+R)(\bar{P}+R)$ → Redundant form
④. $BC + A\bar{C}$	$(P+Q)(\bar{P}+R)$

Q $AB + \bar{A}C + BC$ $\quad AB + \bar{A}C$

Q $AB + \bar{B}C + AC$ $\quad \bar{B}C + AC$

Q $AB + B\bar{C} + AC$ $\quad B\bar{C} + AC$

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

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

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

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

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

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

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

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

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

$= AB + \bar{A}C$

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

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

$= \bar{A}B + BC + \cancel{A\bar{A}C}^{0} + AC \cdot C$

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

$= \bar{A}B + AC$

(*) Maximum number of boolean function or truth table implemented using one variable $= 2^{2^1} = 4$

2^{2^n}

A	Y_0	Y_1	Y_2	Y_3
0	0	0	1	1
1	0	1	0	1

(*) Max. number of boolean function or truth table implemented using two variable $= 2^{2^2} = 16$

SOP	POS	A	B	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
$\bar{A}\bar{B}$	$A+B$	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$\bar{A}B$	$A+\bar{B}$	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$A\bar{B}$	$\bar{A}+B$	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
AB	$\bar{A}+\bar{B}$	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
				0	AB	$A\bar{B}$	A	$A\bar{B}$	B	$A \odot B$	$A+B$	$\bar{A}\bar{B} = \overline{A+B}$	$A \odot B$	$\bar{B}$	$A+\bar{B}$	$\bar{A}$	$\bar{A}+B$	$\bar{A}+\bar{B} = \overline{AB}$	1

⊛ How many functions of type $f(x,y,z) = f(\bar{x},\bar{y},\bar{z})$ are present can be formed from boolean functions with three variables x, y, z.

X	Y	Z	
0	0	0	a
0	0	1	b
0	1	0	c
0	1	1	d
1	0	0	e
1	0	1	d
1	1	0	a
1	1	1	b

$2^4 = 16$

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

X Y Z

0 0 0 $\Rightarrow f(0,0,0) = f(1,1,1)$

0 0 1 $\Rightarrow f(0,0,1) = f(1,1,0)$

0 1 0 $\Rightarrow f(0,1,0) = f(1,0,1)$

0 1 1 $\Rightarrow f(0,1,1) = f(1,0,0)$

Self Dual

If one time dual of any logical expression gives same expression then that expression is in self dual mode.

$$\boxed{F^D = F}$$

⊛ $F = AB + BC + AC$

$F^D = (A+B)(B+C)(A+C)$

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

$= AB + BC + AAC + ACC$

$= AB + AC + BC + AC$

$= AB + BC + AC$

$= F$

• $F = A\bar{B} + \bar{B}C + AC$

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

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

$= AA\bar{B} + AAC + A\bar{B}\bar{B} + A\bar{B}C + A\bar{B}C + ACC + \bar{B}\bar{B}C + \bar{B}CC$

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

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

$= F$

F	F^D
0	1
1	0
A	A
$\bar{A}$	$\bar{A}$

SELF DUAL $\equiv 2$

$0 \to 1$

$1 \to 0$

$A \to A$

$\bar{A} \to \bar{A}$

$B \to B$

$\bar{B} \to \bar{B}$

$AB \to A+B$

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

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

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

$A+B \to AB$

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

$A \oplus B \to A \odot B$

$A \odot B \to A \oplus B$

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

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

■ No. of Self Dual functions/expressions with n variable $= 2^{2^{n-1}}$

Q $F = A \oplus B \oplus C$

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

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

$F^D = \underbrace{(\bar{A}+\underbrace{\bar{B}+C}_{2})}_{1}\underbrace{(\bar{A}+\underbrace{B+\bar{C}}_{3})}_{1}\underbrace{(A+\underbrace{\bar{B}+\bar{C}}_{2})}_{1}+\underbrace{(A+\underbrace{B+C}_{3})}_{1}$

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

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

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

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

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

$= \underset{m_1}{\bar{A}\bar{B}C} + \underset{m_2}{\bar{A}B\bar{C}} + \underset{m_4}{A\bar{B}\bar{C}} + \underset{m_7}{ABC}$

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

$= F$

$\boxed{F^D = F}$

000
001
010
011
100
101
110
111

■ <u>Complement</u> $(Y^c, \bar{Y}, Y')$

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

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

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

$$= (\underbrace{\bar{A}+\bar{B}+\bar{C}}_{2\ 1})(\underbrace{A+\bar{B}+\bar{C}}_{3\ 1})(\bar{A}+\bar{B}+C)$$

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

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

$$= (\underbrace{\bar{B}+\bar{C}}_{1\ 2})(\underbrace{\bar{A}+\bar{C}+B}_{3\ 1})$$

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

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

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

<u>Q</u> $Y = ABC + \bar{A}BC + AB\bar{C}$

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

$$Y = BC + AB$$

$$\bar{Y} = \overline{BC+AB}$$

$$= (\underbrace{\bar{B}+\bar{C}}_{1\ 2})\cdot(\underbrace{\bar{A}+\bar{B}}_{3\ 1})$$

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

NOT ⇄ BUFFER	$Y=\bar{A}$	$\bar{Y}=A$
AND ⇄ NAND	$Y=A$	$\bar{Y}=\bar{A}$
OR ⇄ NOR	$Y=AB$	$\bar{Y}=\bar{AB}$
EX-OR ⇄ EX-NOR	$Y=\bar{AB}$	$\bar{Y}=AB$
	$Y=A+B$	$\bar{Y}=\overline{A+B}$
	$Y=\overline{A+B}$	$\bar{Y}=A+B$
	$Y=A\oplus B$	$\bar{Y}=A\odot B$
	$Y=A\odot B$	$\bar{Y}=A\oplus B$

<u>Q</u> $F = \left(\overline{\overline{(\overline{X+Y}+X)} + \overline{(\overline{X+Y}+Y)}}\right)$

$$= \overline{\overline{(\overline{X+Y}+X)}}\cdot\overline{\overline{(\overline{X+Y}+Y)}}$$

$$= (\underbrace{\overline{X+Y}}_{1}+\underbrace{X}_{2})\cdot(\underbrace{\overline{X+Y}}_{1}+\underbrace{Y}_{3})$$

$$= (\overline{X+Y} + XY)$$

$$= \bar{X}\bar{Y} + XY$$

$$= X\odot Y$$

<u>Q</u> $F = (\underbrace{P+\bar{Q}+\bar{R}}_{1\ 2})(\underbrace{P+\bar{Q}+R}_{1\ 3})(P+Q+\bar{R})$

$$= (P+\bar{Q}+\underbrace{R\bar{R}}_{=0})(P+Q+\bar{R})$$

$$= (\underbrace{P+\bar{Q}}_{1\ 2})(\underbrace{P+Q+\bar{R}}_{1\ 3})$$

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

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

Q $F = (P+\bar{Q})(P\bar{Q}+PR)(\bar{P}\bar{R}+\bar{Q})$

$= (P+\bar{Q})\,P(\bar{Q}+R)(\bar{P}\bar{R}+\bar{Q})$

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

$= (\bar{Q}+PR)P\bar{Q}$

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

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

$= P\bar{Q}(1+R)$

$= P\bar{Q}$

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

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

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

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

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

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

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

$= \overline{AB}\cdot C$

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

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

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

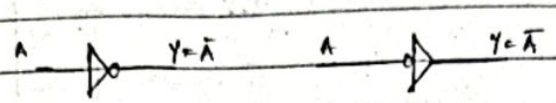
$= AC + AB + BC$

Inversion
logical multiplication
logic sum

Logic Gate Physical device which performs logic operation on one or more logical inputs and produces a single logical output

- Basic gates : NOT, AND, OR
- Universal gates: NAND, NOR
- Arithmetic gates: XOR, XNOR

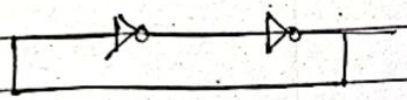
■ **NOT Gate**



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

Q

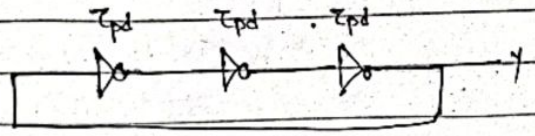
[IES 2010]
[GATE 2010]



(a) Buffer
(b) Astable multivibrator
(c) Bistable multivibrator
(d) Square wave generator.

Bistable, in which the circuit is stable in either state. It can be flipped from one state to the other by an external trigger pulse

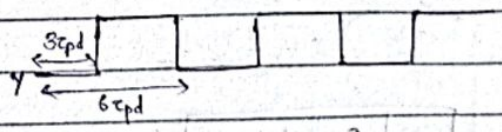
Q



$T = 6\tau_{pd}$

τ_{pd} = propagation delay
$T = 2n\tau_{pd}$
↳ No. of NOT gates

Square wave generator
Astable multivibrator
Clock generator



AND gate

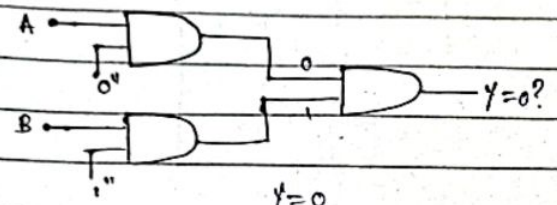
Output is high (1) when all inputs are high

A	B	Y = A·B
0	0	0
0	1	0
1	0	0
1	1	1

- Associative law : $A\cdot(B\cdot C) = (A\cdot B)\cdot C$
- Commutative law : $A\cdot B = B\cdot A$
- Enable / disable :

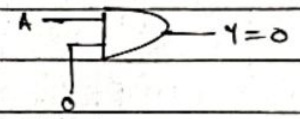
	A	B	Y
Disable o/p = 0	0	0	0
	0	1	0
Enable buffer	1	0	0
	1	1	1

Q

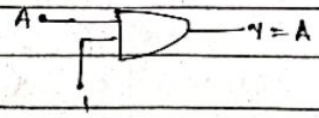


Y = 0

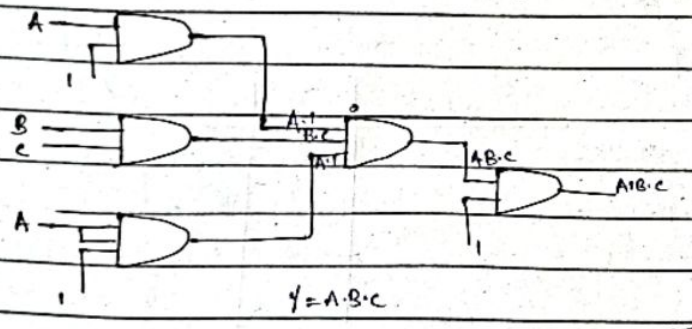
"0" → disable with o/p = 0



"1" → enable / buffer



Q

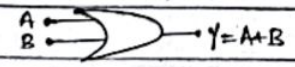


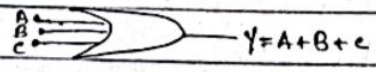
Y = A·B·C

* In TTL logic, if any i/p is open / floating, it will act as "1"
* ECL → "0"

OR gate

Output is high when any one or all inputs are high



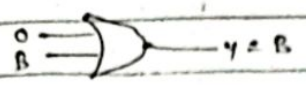


A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

- Associative law : $(A+B)+C = A+(B+C)$
- Commutative law : $A+B = B+A$
- Enable / disable :

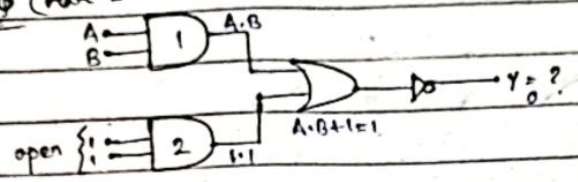
	A	B	Y
Enable buffer	0	0	0
	0	1	0
Disable o/p = 1	1	0	0
	1	1	1

"0" → disable with o/p = 0

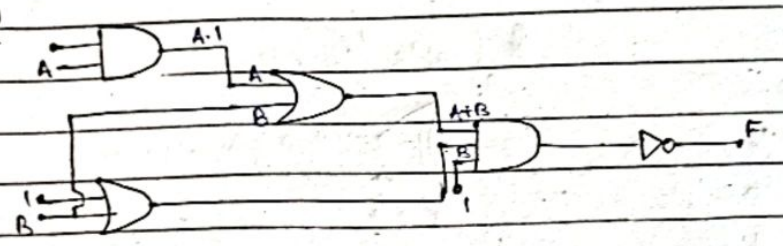


"Disable with output = 1"

Q (Trade 2004)



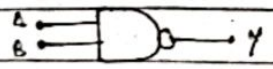
Q

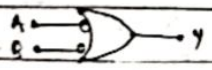


A+B.B.1

NAND Trade

Bubbled AND

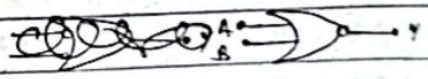




A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

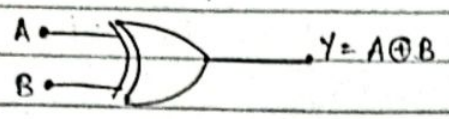
NOR gate

Bubbled OR



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

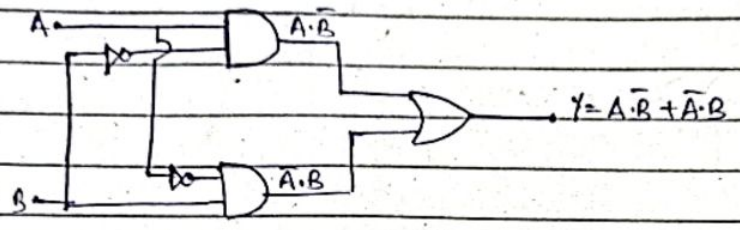
XOR gate



$Y = A \oplus B$

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

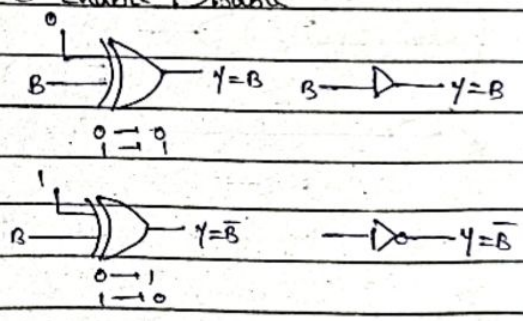
Internal circuit



Truth table

	A	B	$Y = A\bar{B} + \bar{A}B$
Enable buffer	0	0	0
	0	1	1
Enable Inverter	1	0	1
	1	1	0

Enable / Disable



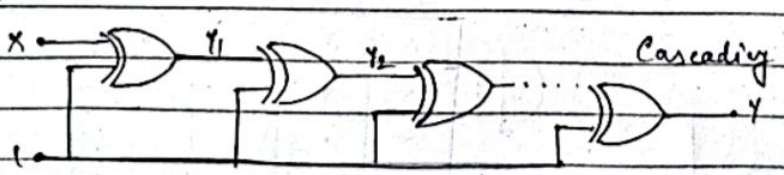
Properties

- (i) $A \oplus A = 0$
- (ii) $A \oplus \bar{A} = 1$
- (iii) $A \oplus 0 = A$
- (iv) $A \oplus 1 = \bar{A}$

- if $A \oplus B = C$, then
  $B \oplus C = A$
  $A \oplus C = B$
  $A \oplus B \oplus C = 0$

- $A \oplus A \oplus A + \cdots n = A$ for n = odd
  $A \oplus A \oplus A + \cdots n = 0$ for n = even

Q



Cascading of 20 XOR gates

$Y_1 = X \oplus 1 = \bar{X}$

$Y_2 = \bar{X} \oplus 1 = (\bar{\bar{X}}) = X$

$\cdots$

$Y_{20} = X$

Q $Y = A \oplus \bar{A} \oplus \bar{A} \oplus A \oplus A \oplus \bar{A} \oplus \bar{A} \oplus A \oplus A$

(a) $1+E$ (b) $A \oplus 0$

(b) 1 NOR B (d) A

$Y = A \oplus \bar{A} \oplus \bar{A} \oplus A \oplus A \oplus \bar{A} \oplus \bar{A} \oplus A \oplus A$

$= 1 \oplus 1 \oplus 1 \oplus 1 \oplus A$

$= A$

Q $f(A,B) = A \oplus B$

Simplified form of the func, $f(f(x \oplus y, z), w) = ?$

(a) $(x \oplus y \oplus z) \cdot w$ (b) $x \oplus y \oplus z \oplus w$

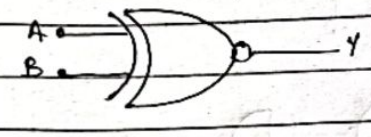
(c) $(x \oplus y \oplus z) + w$ (d) None

$= f(f(x \oplus y, z), w)$

$= f(x \oplus y \oplus z, w)$

$= x \oplus y \oplus z \oplus w$

XNOR gate



	A	B	Y
Enable Inverter	0	0	1
	0	1	0
Enable buffer	1	0	0
	1	1	1

Truth table

A	B	$Y = A \odot B = A \cdot B + \bar{A} \cdot \bar{B}$
0	0	1
0	1	0
1	0	0
1	1	1

Properties

$A \odot A = 1$

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

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

$A \odot 1 = A$

$A \odot A = 1$

$A \odot A \odot A = A$

$\vdots$

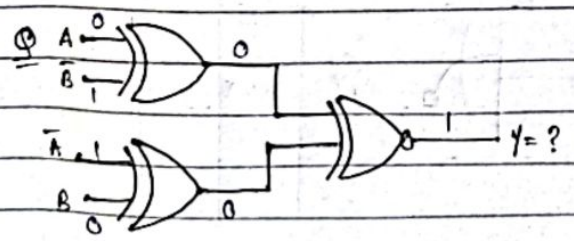
$A \odot A \odot A \odot \cdots n = A, n = \text{odd}$

$= 1, n = \text{even}$

XOR & XNOR → Compliment → Even i/p

XOR & XNOR → Same → Odd i/p

Q



$Y = 1$

Q $Y = A \odot A \odot A \odot A \odot A \odot \bar{A} \odot A$

$= A \odot \bar{A} \cdot A$ $(5\ A \odot A = 1)$

$= 0 \cdot A$

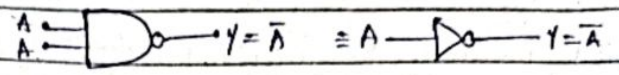
$= \bar{A}$

NAND as universal gates

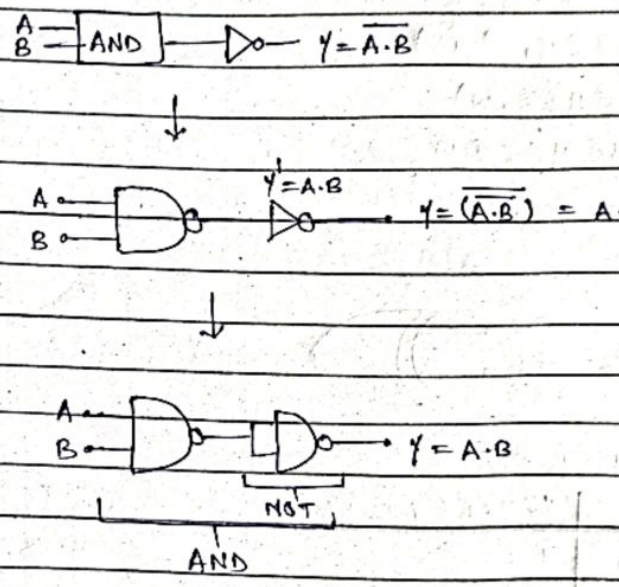
- AND, OR and NOT are sufficient to implement any digital system.
- If we convert NAND or NOR to any of these, we can say that any circuit can be implemented by NAND or NOR alone.

- NAND as NOT $\quad Y = \overline{A \cdot B} \quad Y = \overline{A \cdot A} = \overline{A}$

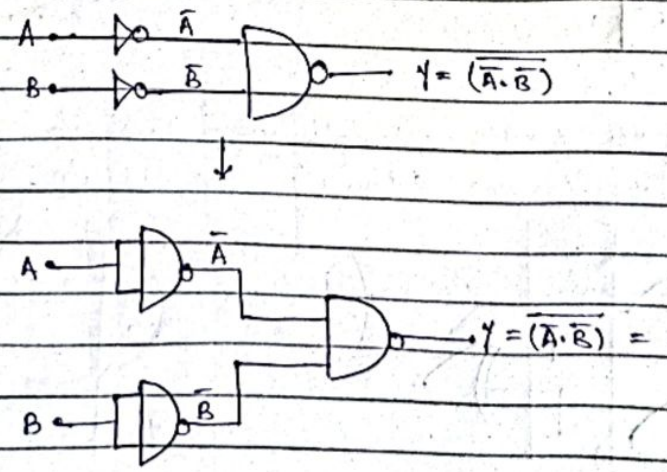
if i/p are same



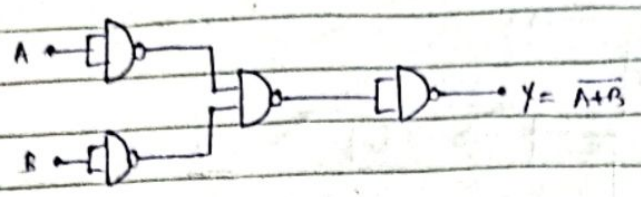
- NAND as AND
(2 Nand Gates)



- NAND as OR $\quad Y = \overline{A \cdot B} = \overline{A} + \overline{B}$
(3 Nand Gates)

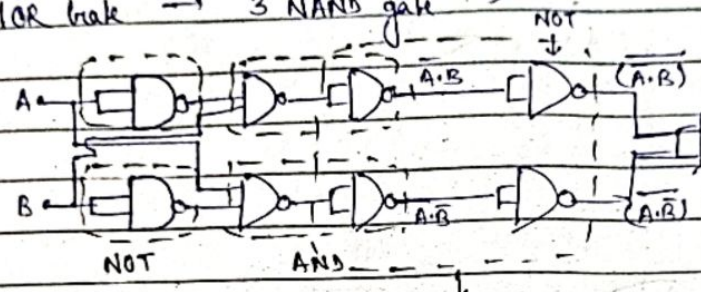


• NAND as NOR (4 Nand gates)



• NAND as XOR $Y = A\bar{B} + \bar{A}B$

2 NOT Gate → 2 NAND gate
2 AND Gate → 4 NAND gate
1 OR Gate → 3 NAND gate
} 9 NAND gate



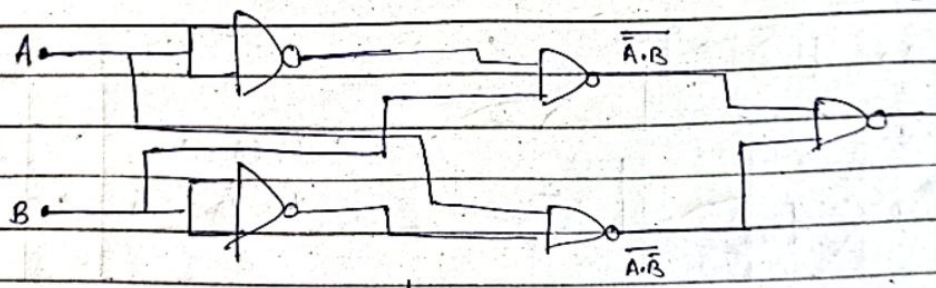
$Y = \overline{(\overline{(\overline{A \cdot B})} \cdot (\overline{A \cdot \bar{B}})}$

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

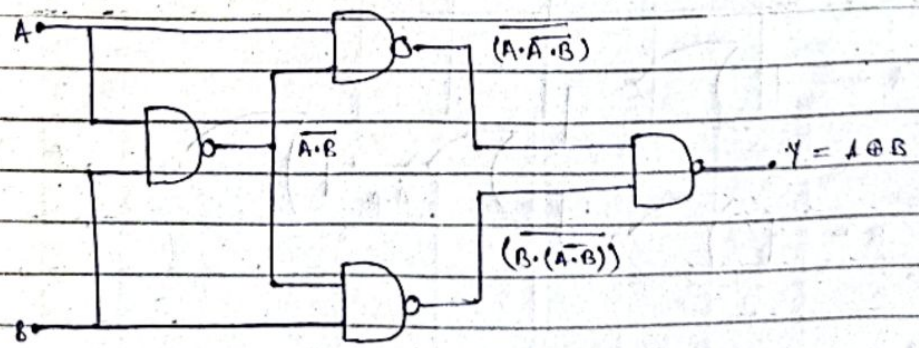
$= A \oplus B$

Not required
Can be removed

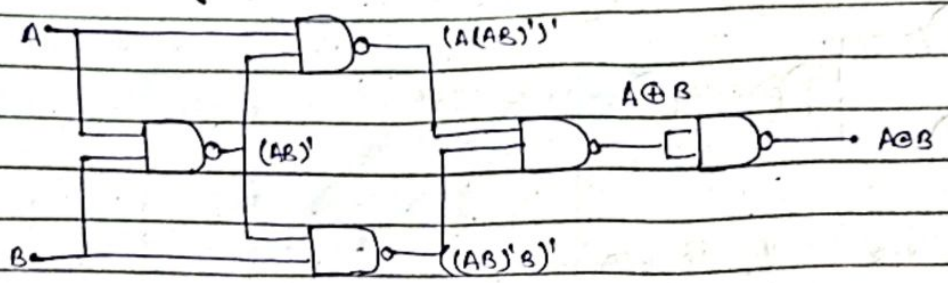
↓



↓



- NAND as XNOR gate



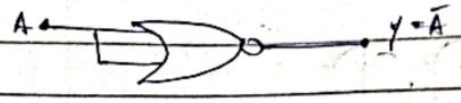
NOR as Universal gates

- NOT (1NOR)

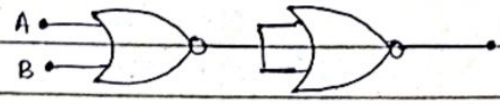
$Y = \overline{A+B}$

$Y = \overline{A+A}$

$Y = \overline{A}$

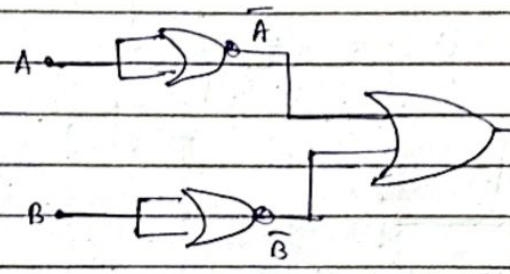


- OR (2 NOR)



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

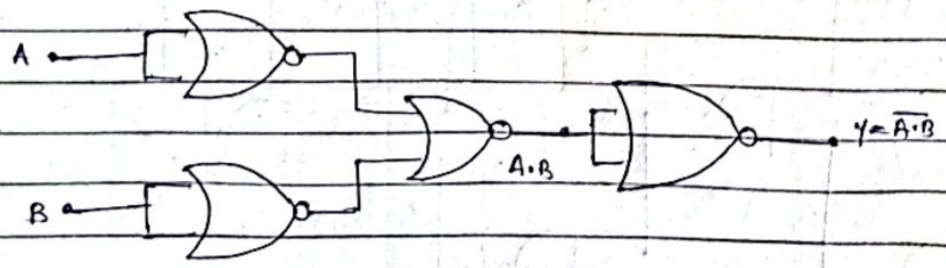
- AND (3 NOR)



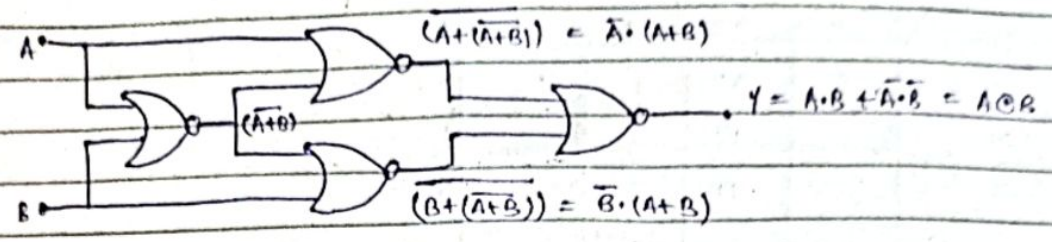
$Y = \overline{(\overline{A}+\overline{B})}$

$Y = (\overline{\overline{A}}) \cdot (\overline{\overline{B}}) = A \cdot B$

- NAND (4 NOR)

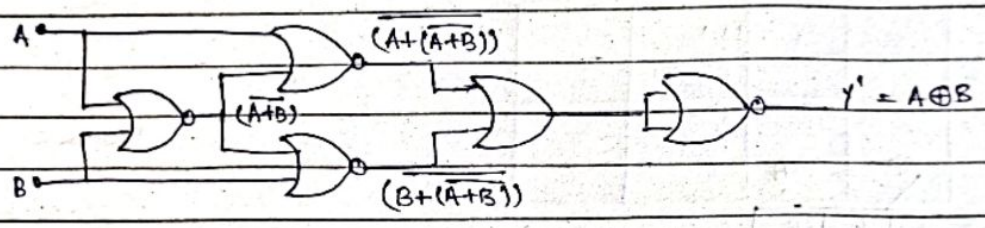


• XNOR (5 NOR)



$$Y = \left[\overline{\bar{A}(A+B) + \bar{B}(A+B)}\right]$$
$$Y = \overline{(\bar{A}\cdot(A+B))}\cdot\overline{(\bar{B}\cdot(A+B))}$$
$$Y = (A+(\overline{A+B}))\cdot(B+(\overline{A+B}))$$
$$Y = A\cdot B + (\overline{A+B})$$
$$Y = A\cdot B + \bar{A}\cdot\bar{B} \qquad \boxed{Y = A \odot B}$$

• XOR (5 NOR)



■ K'MAP [Karnaugh Map]

Based on Gray Code

Binary → B_3 B_2 B_1 B_0

Gray → G_3 G_2 G_1 G_0

Ex: Binary : 1 1 0 0 1 0

Gray : 1 0 1 0 1 1

Binary : 1 1 0 0 0 1 1

Gray : 1 0 1 0 0 1 0

• Gray Code

Decimal	Binary	Gray Code
0	00	00
1	01	01
2	10	11
3	11	10

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

f(A,B)

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	$\bar{A}\bar{B}$ 00 1	$\bar{A}B$ 01 2
A 1	$A\bar{B}$ 10 2	AB 11 3

f(A,B,C)

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

$f(A,B,C,D)$

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

Formation of groups

16 group $= 2^{4}$ → 4 variable minimise
8 group $= 2^{3}$ → 3 variable minimise
4 group $= 2^{2}$ → 2 variable minimise
2 group $= 2^{1}$ → 1 variable minimise

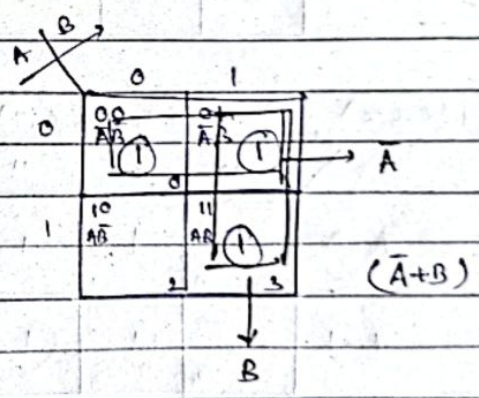
Q $f(A,B) = \bar{A}\bar{B} + \bar{A}B + AB = \sum m(0,1,3)$

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

$= \bar{A} + AB$

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

$= \bar{A}+B$



Q $f(A,B) = \bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB = \sum m(0,1,2,3)$

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	1 (0)	1 (1)
A 1	1 (2)	1 (3)

$= 1$

Q $f(A,B,C) = \sum m(0,1,3,6,7)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1 (0)	1 (1)	1 (3)	(2)
A 1	(4)	(5)	1 (7)	1 (6)

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

AB

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

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

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

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

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

Q $f(A,B,C) = \sum m(0,1,3,5,6,7)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1 (0)	1 (1)	1 (3)	(2)
A 1	(4)	1 (5)	1 (7)	1 (6)

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

Q $f(A,B,C,D) = \sum m(0,2,4,6,10,11,13,15)$

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

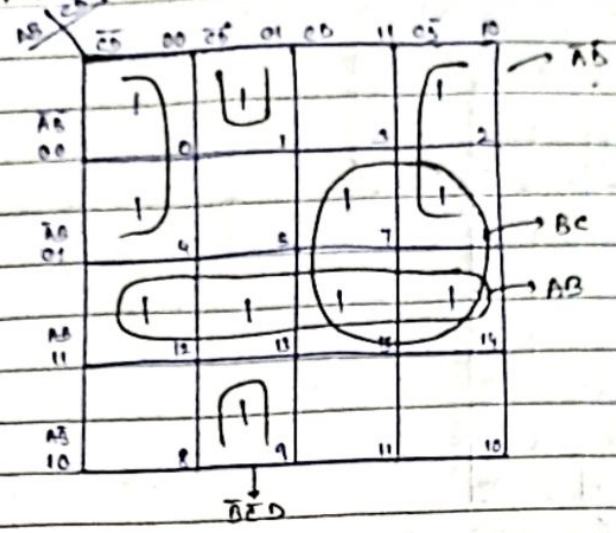
$\bar{A}\bar{D}$

ABD

$A\bar{B}C$

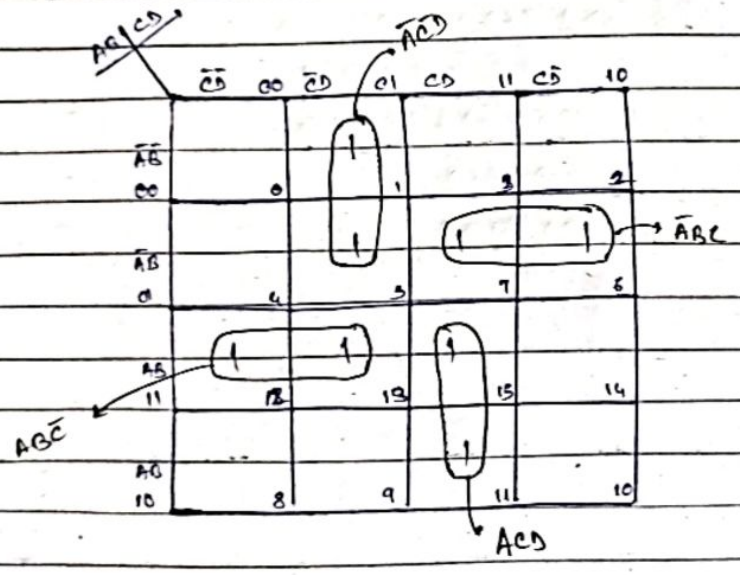
$\bar{A}\bar{D} + ABD + A\bar{B}C$

Q $f(A,B,C,D) = \sum m(0,1,2,4,6,7,9,12,13,14,15)$



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

Q $f(A,B,C,D) = \sum m(1,5,6,7,11,12,13,15)$



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

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

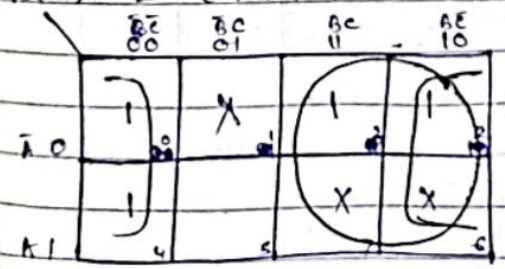
$\bar{A}BCD + \bar{A}BC\bar{D}$
$\bar{A}BC(D+\bar{D})$
$\bar{A}BC$
$AB\bar{C}\bar{D} + AB\bar{C}D$

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

Don't Care Condition.

Combination of inputs on which the output may or may not depends are called don't care condition.

Q Find the minimised Boolean expression for the function given as $f(A,B,C) = \sum m(0,2,3,4) + \sum d(1,6,7)$



$\Rightarrow B + \bar{C}$

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

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

Q Find the minimised Boolean expression for the function given as
$f(A,B,C,D) = \Sigma m(0,2,4,6,7,8,10,11,12,14,15) + \Sigma d(1,3)$

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

$\bar{D} + CD$

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

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

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

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

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

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

$\bar{D}$

$\bar{A}\bar{B}CD + \bar{A}BCD + ABCD + A\bar{B}CD$

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

$\bar{B}CD + BCD$

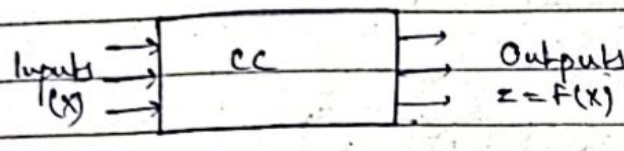
$CD(B+\bar{B})$

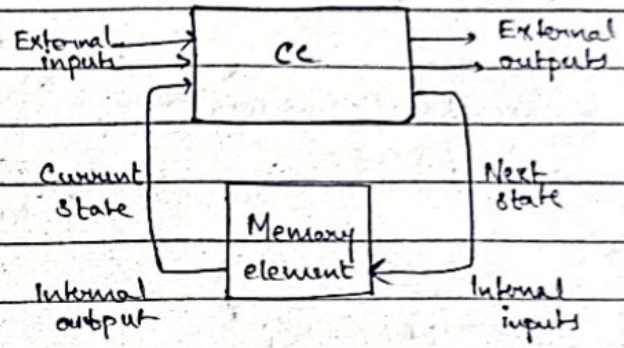
CD

Comparision b/w Combinational & Sequential Circuit

- o/p is only dependent on present i/p
- Speed is fast
- No feedback b/w input and output.
- Elementary building blocks - logic gates.

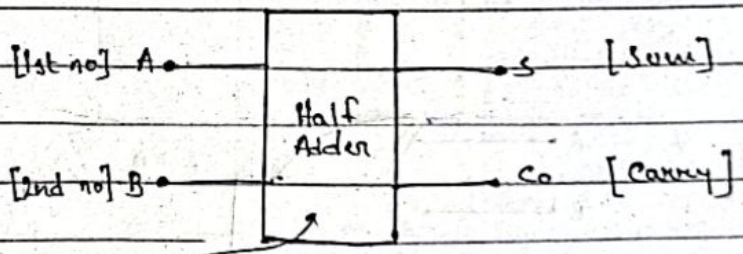
- o/p depends on the present i/p as well as previous o/p
- Speed is slow
- Feedback path exists b/w input and output
- Elementary building blocks - Flip flops.





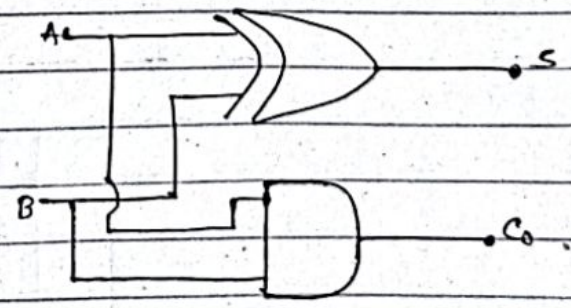
Half Adder (Combinational Circuit)

- Single bit numbers
- It doesn't carry from previous sum



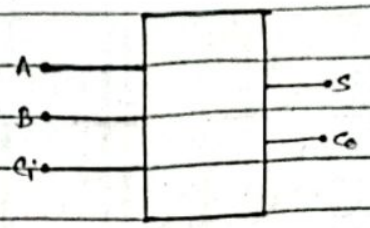
A	B	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B$$
$$C_o = A \cdot B$$



Full Adder (Combinational Circuit)

A	B	C	S	C_o
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 Adder

For S

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

Check board configuration

$S = A \oplus B \oplus C$

For C_o

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

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

$BC + AC + AB$

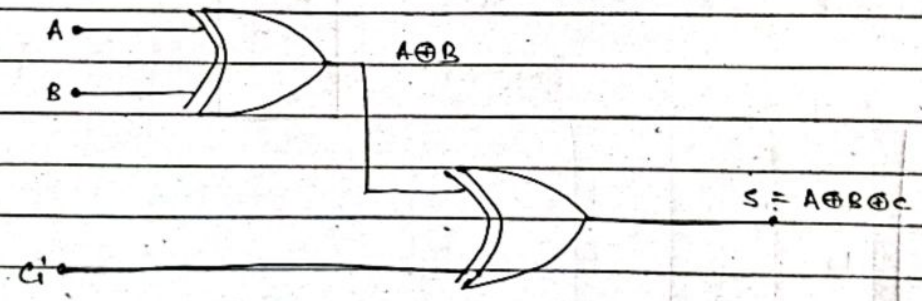
$C_o = BC_i + AB + AC_i$

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

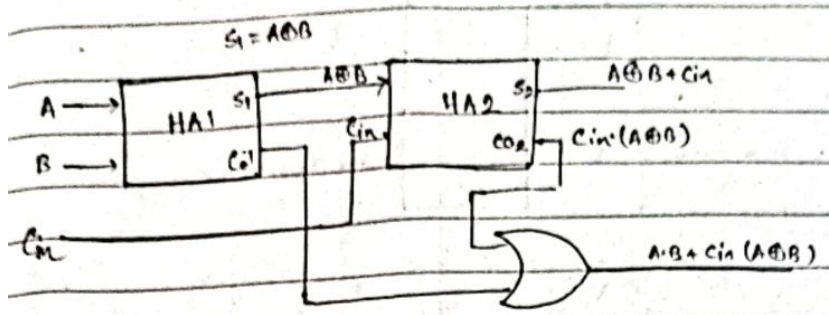
$\bar{A}BC + A\bar{B}C + ABC + AB\bar{C}$

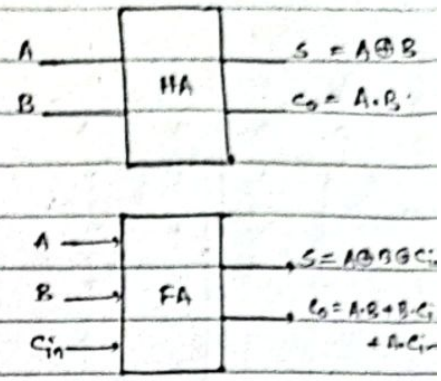
$AB + C_i(A\bar{B} + \bar{A}B)$

$AB + C_i(A \oplus B)$



Full Adder using half adder





$$
\begin{aligned}
A \cdot B + C_{in}(A \oplus B) &= A \cdot B + C_{in}(AB' + A'B) \\
&= A \cdot B + AB'C_{in} + A'BC_{in} \\
&= A(B + B'C_{in}) + A'BC_{in} \\
&= A(B + C_{in}) + A'BC_{in} \\
&= A \cdot B + A \cdot C_{in} + A'BC_{in} \\
&= A \cdot B + (A + A'B)C_{in} \\
&= A \cdot B + (A + B)C_{in} \\
&= A \cdot B + AC_{in} + BC_{in}
\end{aligned}
$$

Four bit parallel adder using full adder

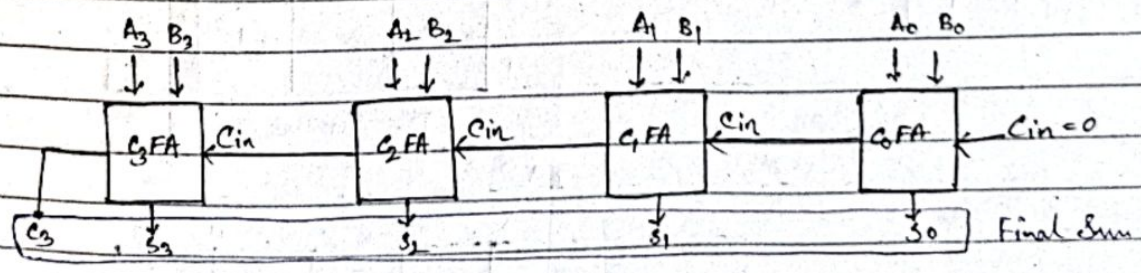
- 74HC283 → IC for 4 bit full adder

		C_{in}	C_{in}	C_{in}	C_{in}
A	= 0	A_3	A_2	A_1	A_0
B	= 0	B_3	B_2	B_1	B_0
S	= C_3	S_3	S_2	S_1	S_0
		C_3	C_2	C_1	C_0

$C_3 = 1$ or 0

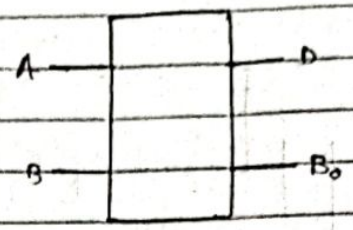
When $C_3 = 0 \rightarrow$ 4 bit sum

$C_3 = 1 \rightarrow$ 5 bit sum



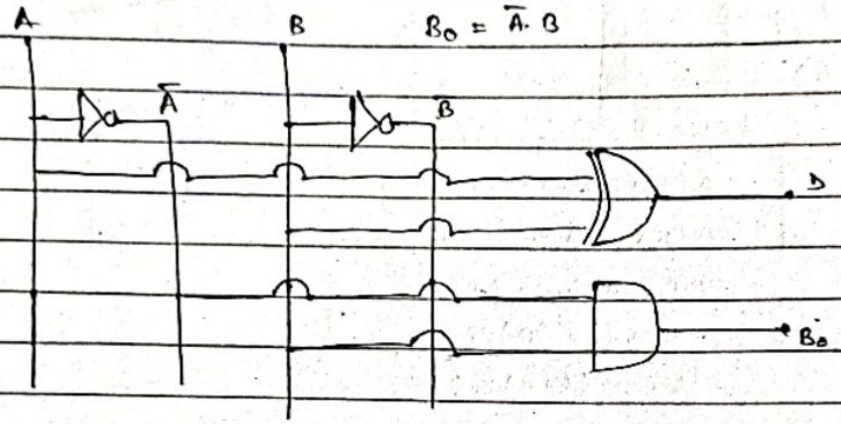
Half Subtractor

A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



$D = A \oplus B$

$B_0 = \bar{A} \cdot B$



Full Subtractor

A	B	C	B	D_0
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

Full subtractor

For D

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

$D = A \oplus B \oplus C$

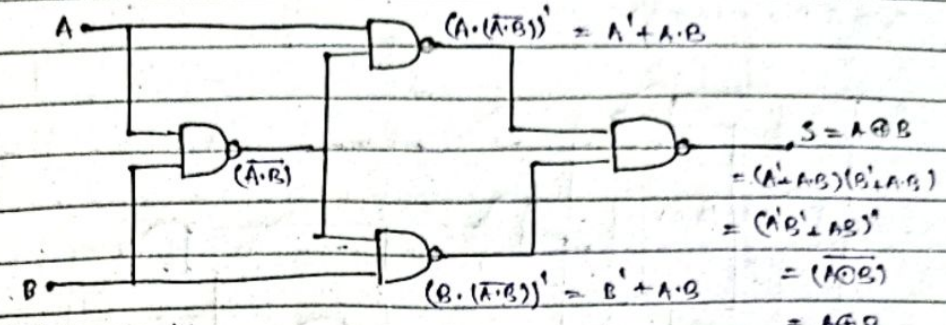
A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

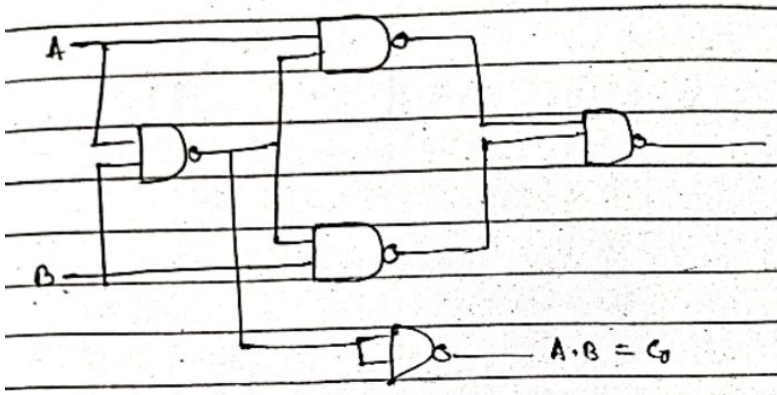
$B_0 = BC + \bar{A}C + \bar{A}B$

Half Adder using NAND Gate only.

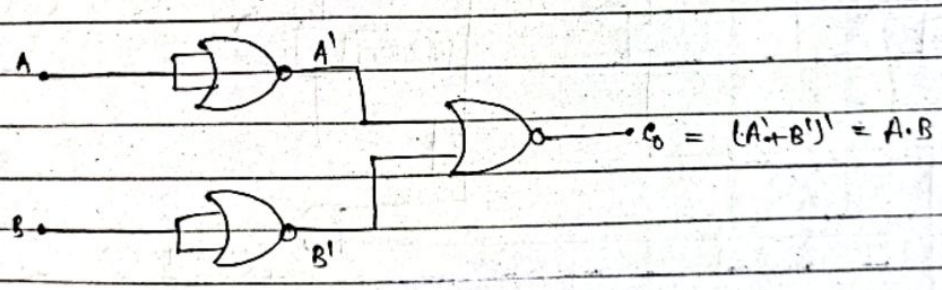
$S = A \oplus B$

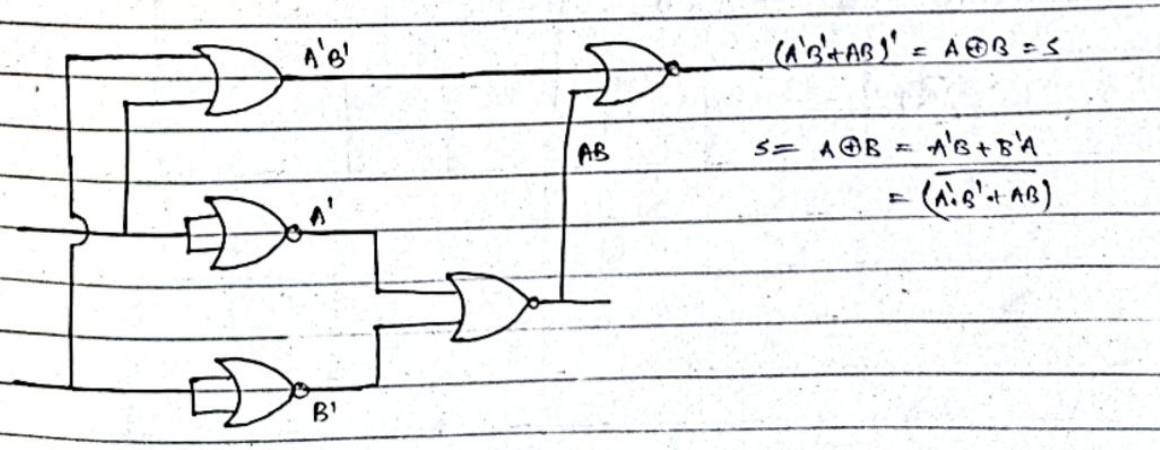
$C_0 = A \cdot B$





Half Adder using NOR gate only

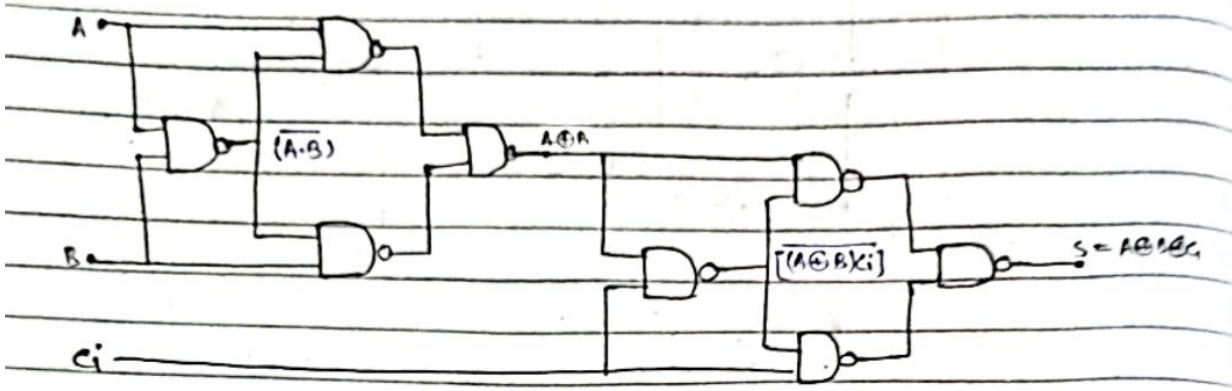


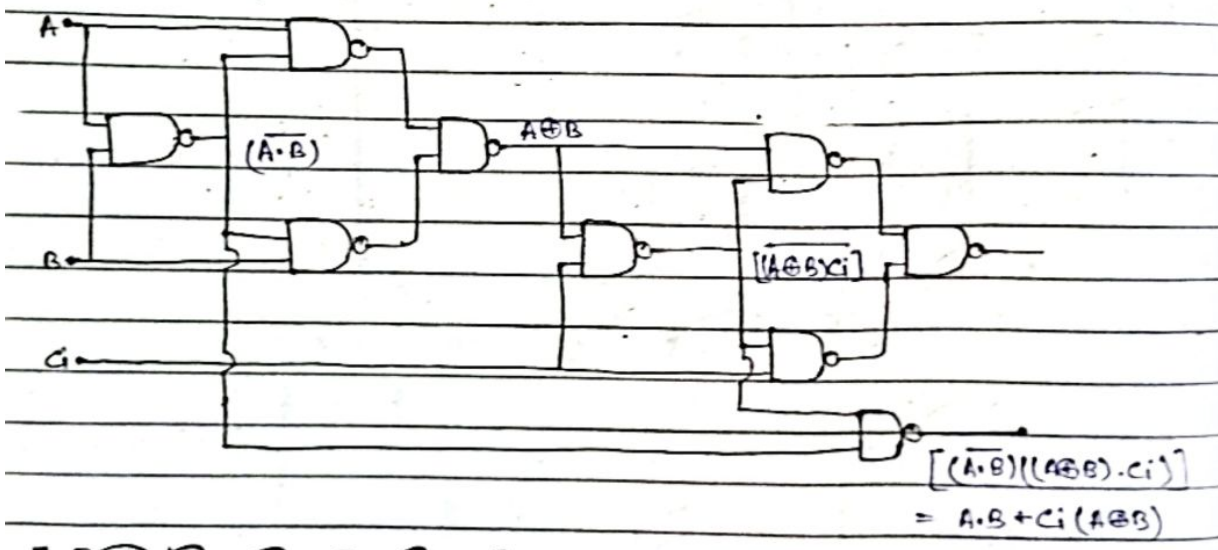


- Full Adder using NAND Gate

$S = A \oplus B \oplus C_i$

$C_o = A \cdot B + C_i(A \oplus B)$





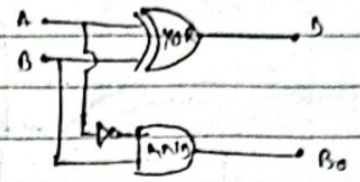
- ~~Full Adder using NOR gate~~

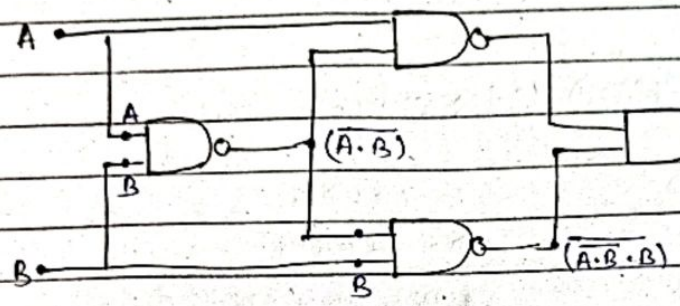
Half Subtractor using NAND Gate only

A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A \oplus B$

$B_0 = A'B$



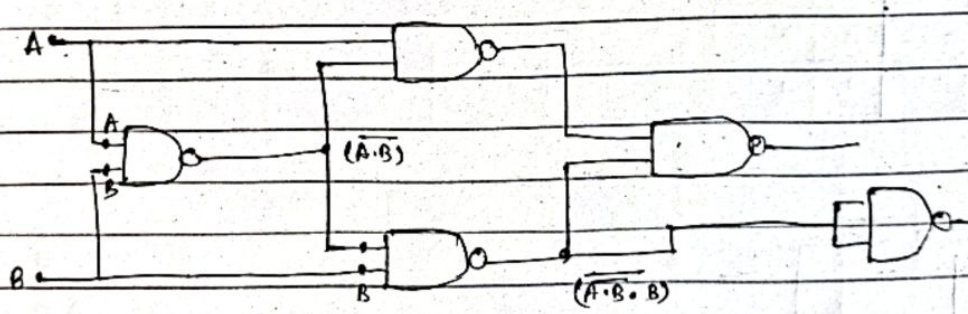


$D = A \oplus B$

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

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

$= [\bar{A}B]'$



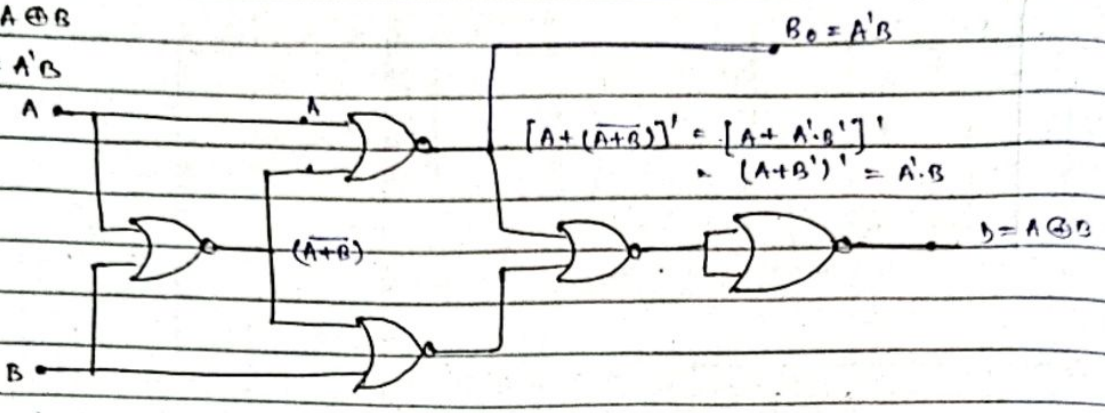
$B_0 = [(A'B)']'$

$B_0 = A'B$

Half Subtractor using NOR gate only

$D = A \oplus B$

$B_0 = A'B$



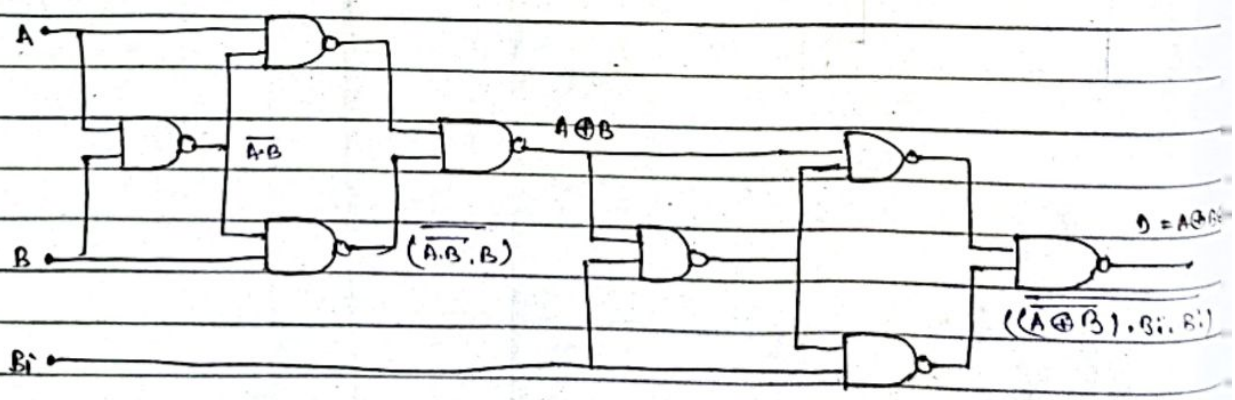
Full Subtractor using NAND gates only

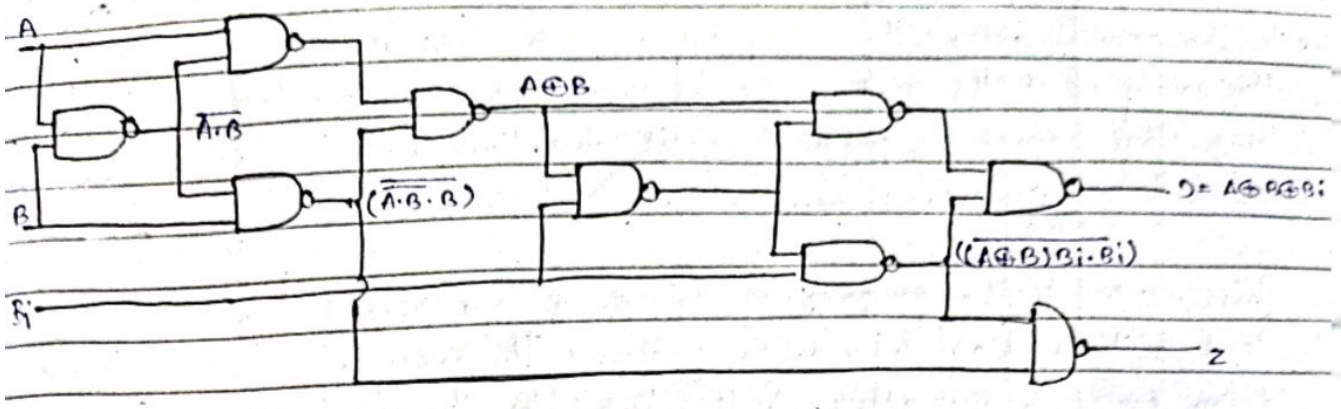
Minuend with A	Subtrahend B	B_i (lsb)	D	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$S = A \oplus B \oplus C_i$

$D = A \oplus B \oplus B_i$

$B_o = \overline{A}B + \overline{A}B_i + BB_i$



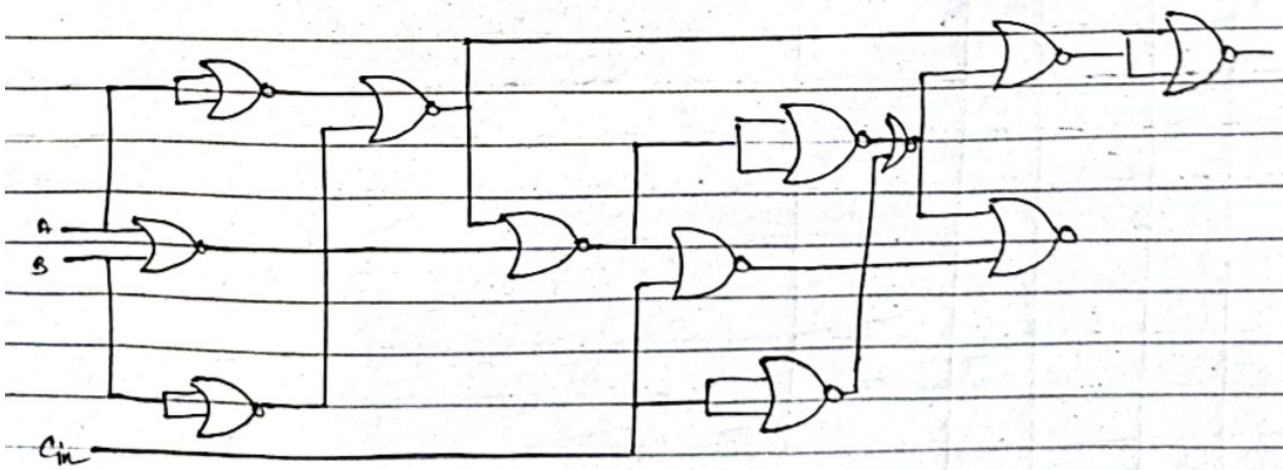


$$\overline{\overline{A \cdot B} \cdot B} = \overline{\overline{A \cdot B}} + \overline{B}$$
$$= AB + \overline{B}$$
$$= A + \overline{B}$$

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

$$Z = \overline{(A + \overline{B}) \cdot (A \oplus B + \overline{Bi})}$$
$$Z = \overline{(A + \overline{B})} + \overline{(A \oplus B + Bi)}$$
$$Z = \overline{A} \cdot B + \overline{A \oplus B} \cdot Bi$$
$$Z = \overline{A} \cdot B + A \odot B \cdot Bi$$
$$Z = \overline{A} B + (AB + \overline{A}\overline{B}) \cdot Bi = \overline{A} B + ABBi + \overline{A}\overline{B} Bi$$

- Full Adder using two input NOR gate

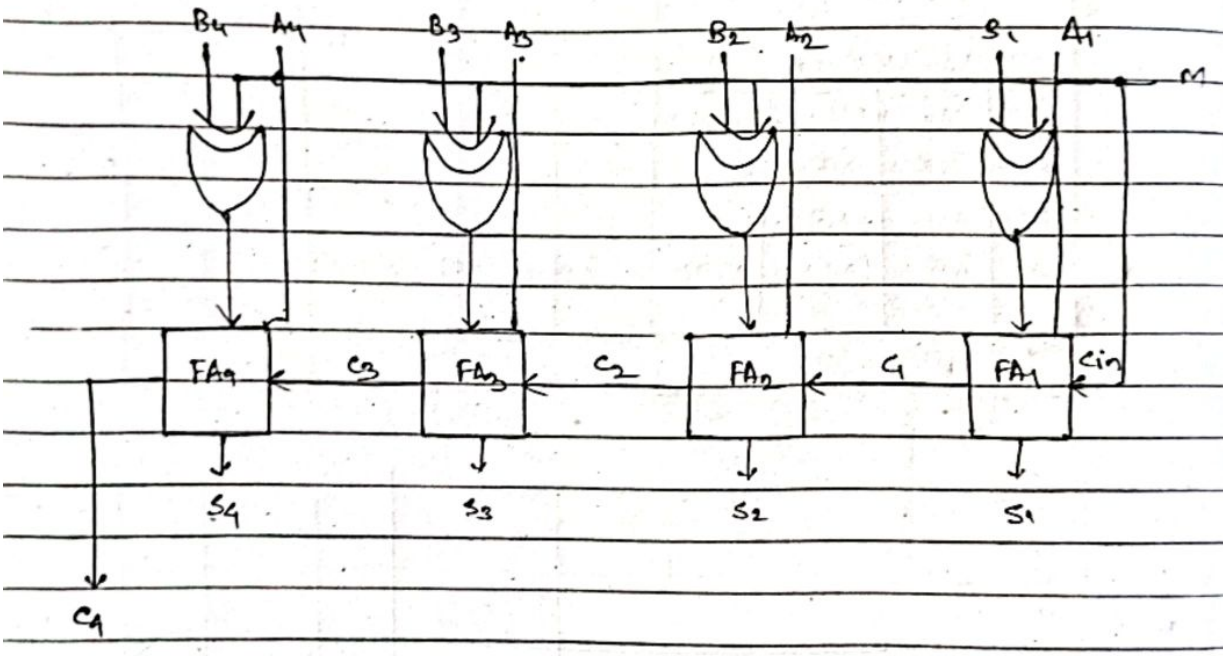


Ripple Carry Adder

In the parallel adder, the carry out of each stage is connected to the carry in of the next stage. The sum and carry out bits of any stage cannot be produced, until some time after the carry in of that stage occurs.

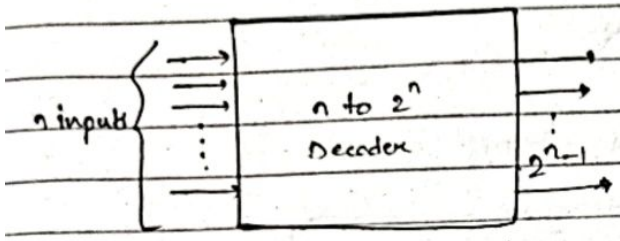
This is due to the propagation delays in logic circuitry, which lead to time delay in addition process. The carry propagation delay for each full adder is the time b/w the application of the carry in and the occurance of the carry out.

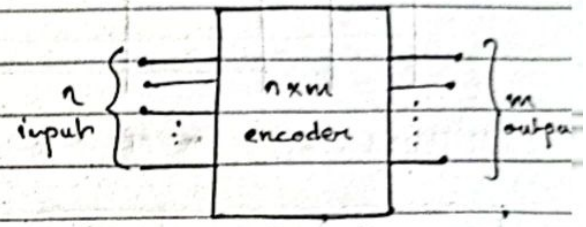
BINARY ADDER SUBTRACTOR



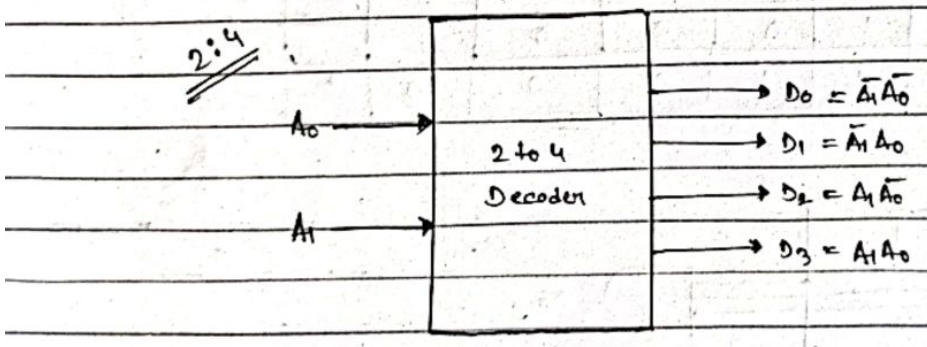
ENCODERS & DECODERS

- Combinational circuits
- Fall under Medium Scale Integrated (MSI) Circuits
- Encoders have 'n' input and 'm' output
- Function of decoder is opposite to encoder





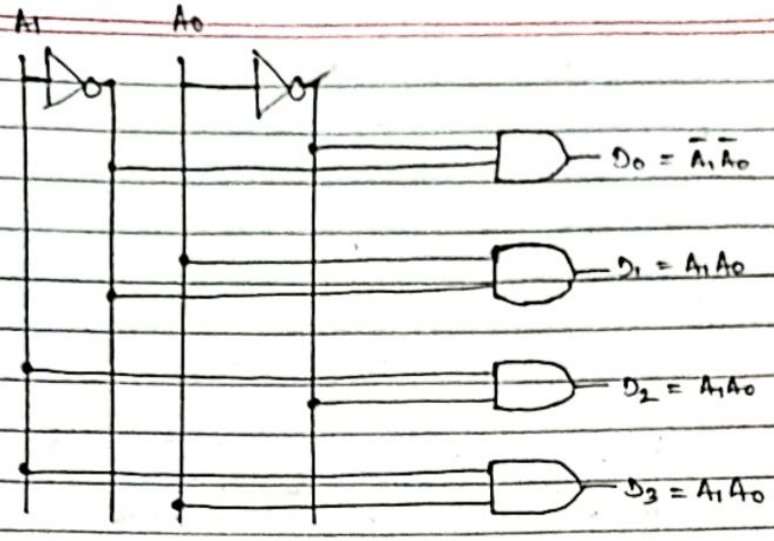
Thus a decoder has n inputs and 2^n outputs. Each of the 2^n outputs corresponds to one of the possible 2^n input combinations



2-to-4 Decoder without enable

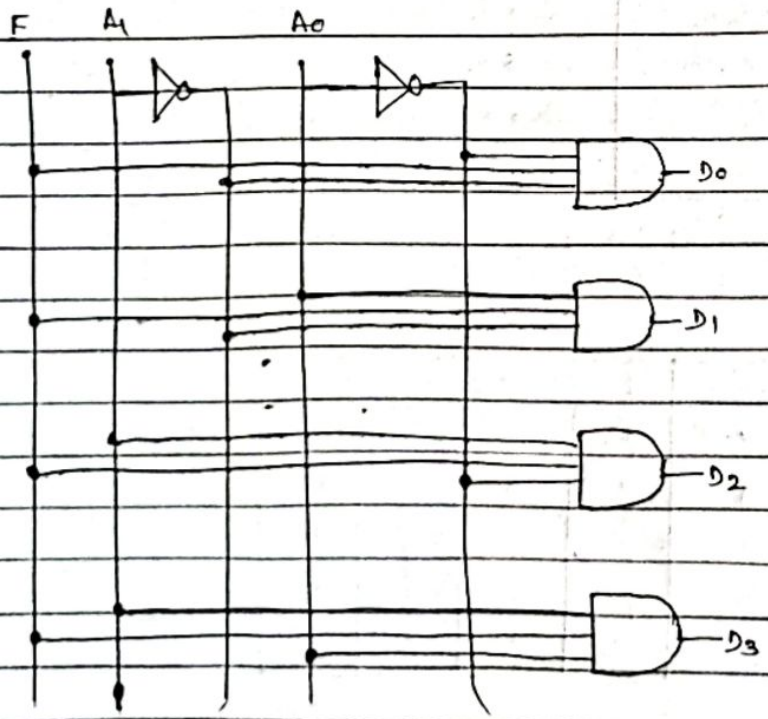
Decimal	Input		Output			
	A_1	A_0	D_0	D_1	D_2	D_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

Truth table for 2-to-4 decoder



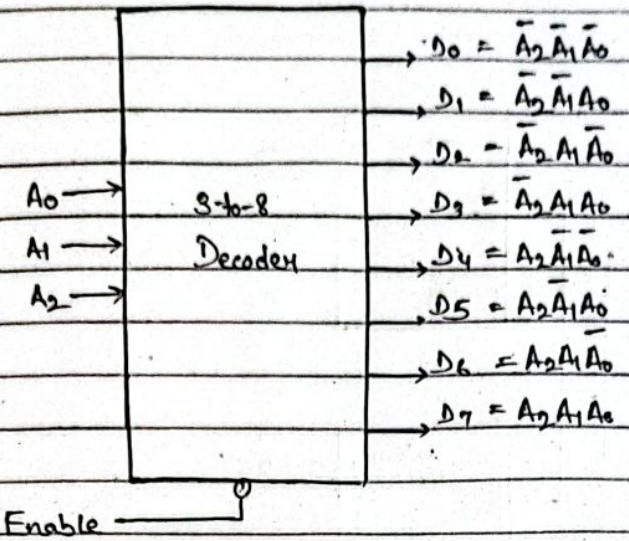
IMPLEMENTATION OF 2-TO-4 DECODER WITHOUT ENABLE

Decimal value	Enable	Inputs		Outputs			
	E	A_1	A_0	D_0	D_1	D_2	D_3
	0	X	X	0	0	0	0
0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
2	1	1	0	0	0	1	
3	1	1	1	0	0	0	1

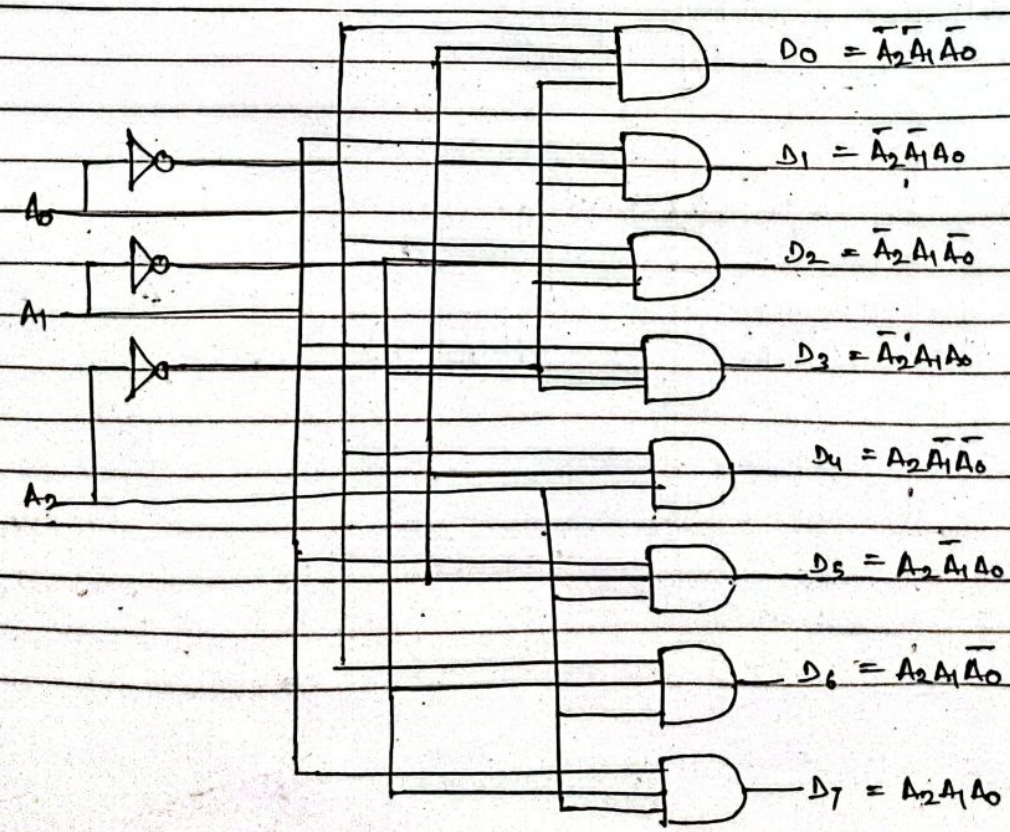


Implemmutation of 2-to-4 decoder with enable

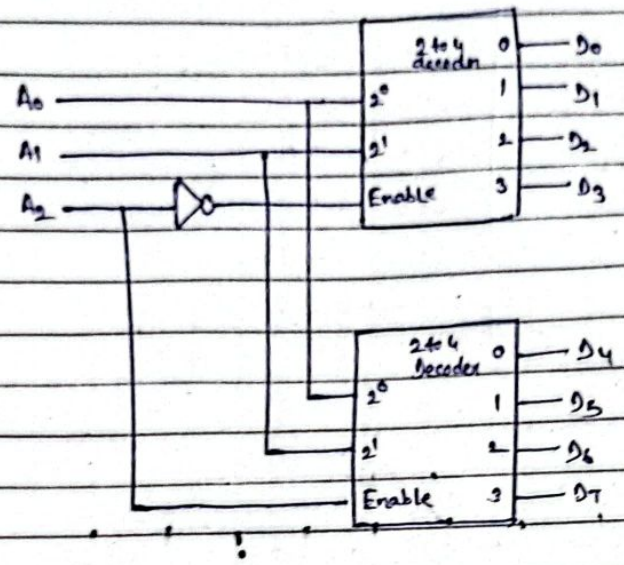
3:8



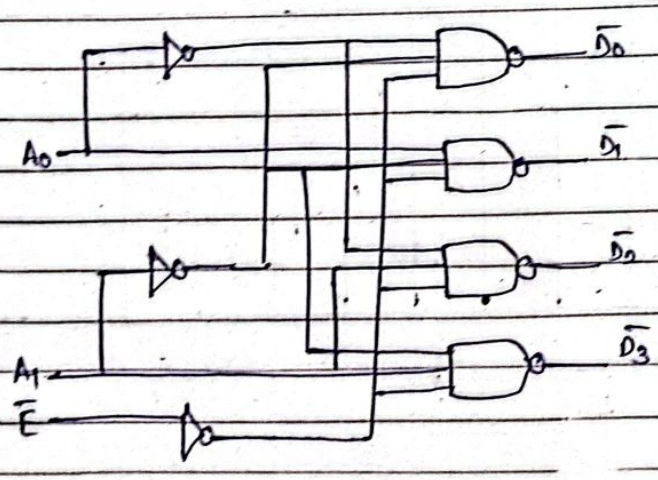
Decimal	INPUTS			OUTPUTS							
Code	A_2	A_1	A_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1



- 3 to 8 decoder using two 2 to 4 decoders



- 2 to 4 decoder with enable constructed with NAND gates



- 2 to 4 decoder with enable using NAND gates

Decimal value	Enable E'	Inputs A_1	A_0	Outputs D_0'	D_1'	D_2'	D_3'
	1	x	x	1	1	1	1
0	0	0	0	0	1	1	1
1	0	0	1	1	0	1	1
2	0	1	1	1	1	0	1
3	0	1	1	1	1	1	0

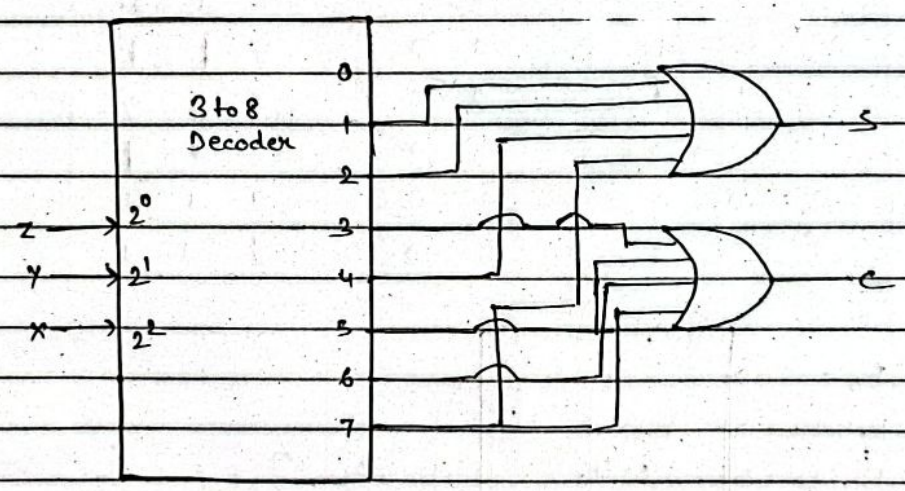
$D_0 = \overline{E\bar{A_1}\bar{A_0}}$ $\quad \bar{D_1} = \overline{E\bar{A_1}A_0}$

$\bar{D_2} = \overline{EA_1\bar{A_0}}$ $\quad \bar{D_3} = \overline{EA_1A_0}$

Decoder implementation of full adder

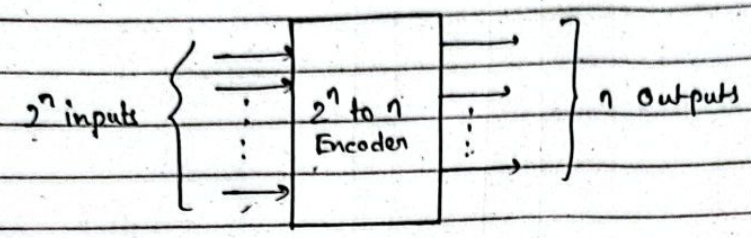
Decimal Value	Input			Output	
	x	y	z	s	c
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$$s(x,y,z) = \sum m(1,2,4,7)$$
$$c(x,y,z) = \sum m(3,5,6,7)$$



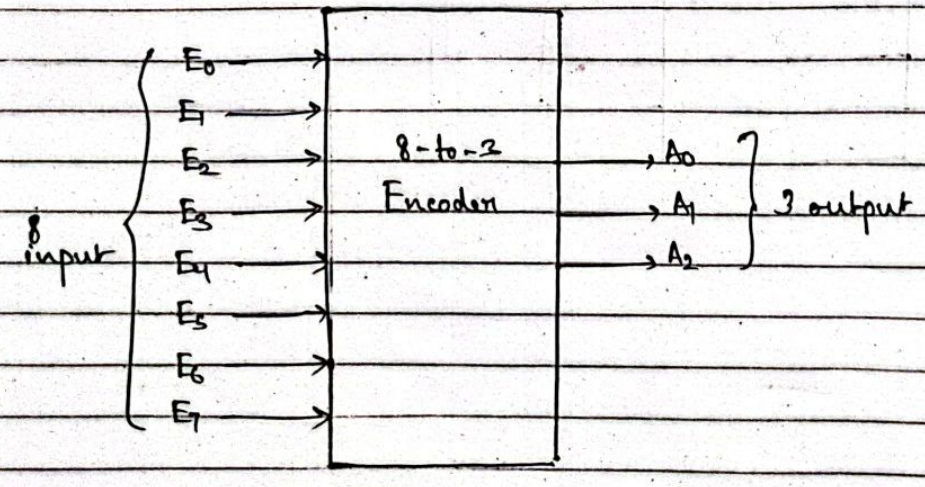
ENCODER

- An encoder performs the inverse operation of a decoder
- 2^n inputs, n outputs
- Only one input can be logic 1 at any given time (active input). All other inputs must be 0's.



Octal to binary encoder

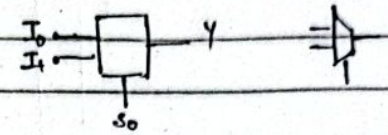
Inputs								Outputs			Decimal Code
E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	A_2	A_1	A_0	
0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0	2
0	0	0	0	1	0	0	0	0	1	1	3
0	0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	0	0	0	0	1	0	1	5
0	1	0	0	0	0	0	0	1	1	0	6
1	0	0	0	0	0	0	0	1	1	1	7



Multiplexers

- It is a combinational circuit that selects binary information from one of many input lines and directs it to the o/p line.

- It is simply a data selector



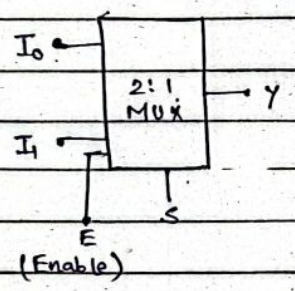
- Advantages
  1. Reduces no. of wires
  2. Reduces circuit complexity & cost
  3. Implementation of various circuit using MUX

$n = 2^m$ → No. of select lines

n → no. of i/p

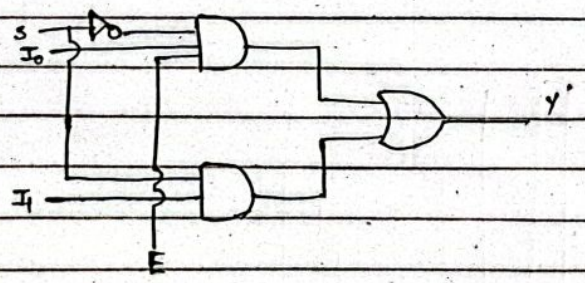
$\boxed{m = \log_2 n}$

- 2:1 MUX

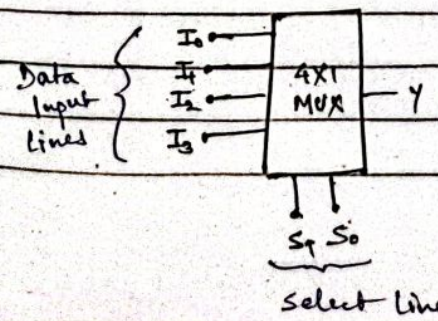


E	S	Y
0	x	0
1	0	I_0
1	1	I_1

$$Y = E\bar{S}I_0 + ESI_1$$
$$Y = E(\bar{S}I_0 + SI_1)$$



- 4X1 Multiplexer



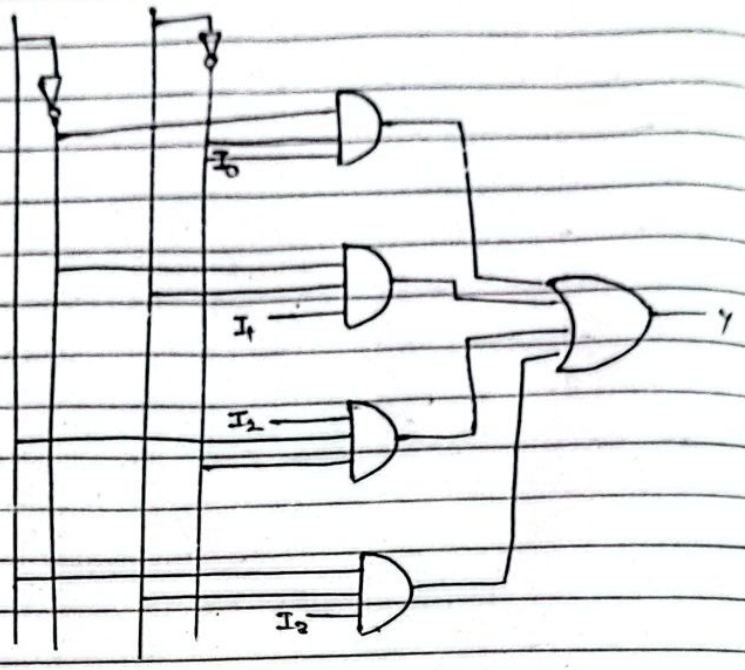
$n = 4$

$m = \log_2 n$

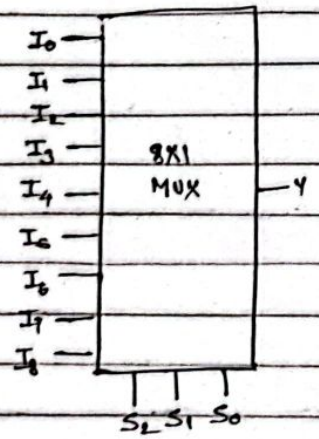
$m = \log_2 4 = \log_2 2^2 = 2\log_2 2 = 2 \times 1 = 2$

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

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



- **8X1 Multiplexer**



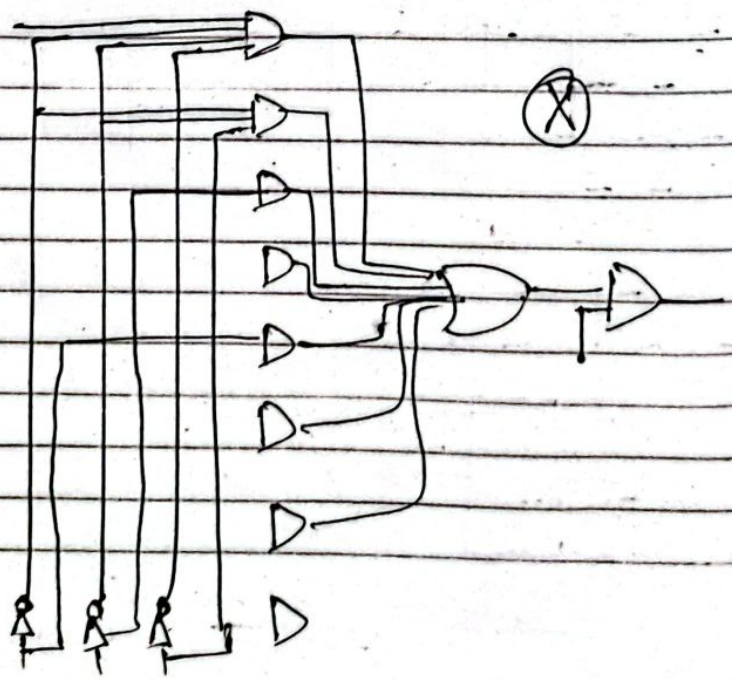
$n = 8 = 2^3$

$m = \log_2 n$

$m = \log_2 2^3 = 3$

$$Y = \bar{S}_2\bar{S}_1\bar{S}_0 I_0 + \bar{S}_2\bar{S}_1 S_0 I_1 + \bar{S}_2 S_1\bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3 + S_2\bar{S}_1\bar{S}_0 I_4 + S_2\bar{S}_1 S_0 I_5 + S_2 S_1\bar{S}_0 I_6 + S_2 S_1 S_0 I_7$$

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

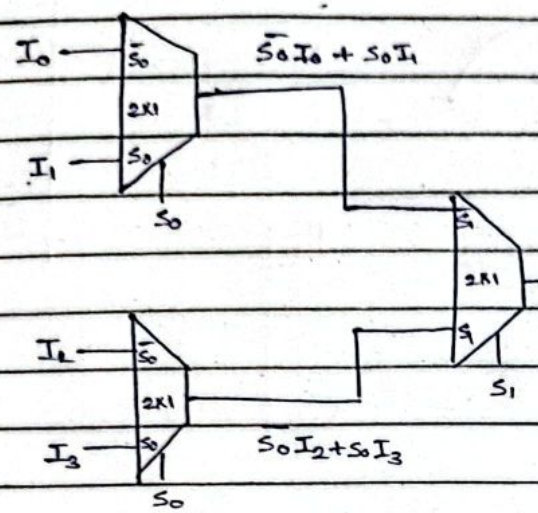


Implement 4X1 MUX using 2X1 MUX

$n = 4$

$4/2 = 2$
$2/2 = 1$ } $2+1 = 3 \rightarrow$ 2X1 MUX

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

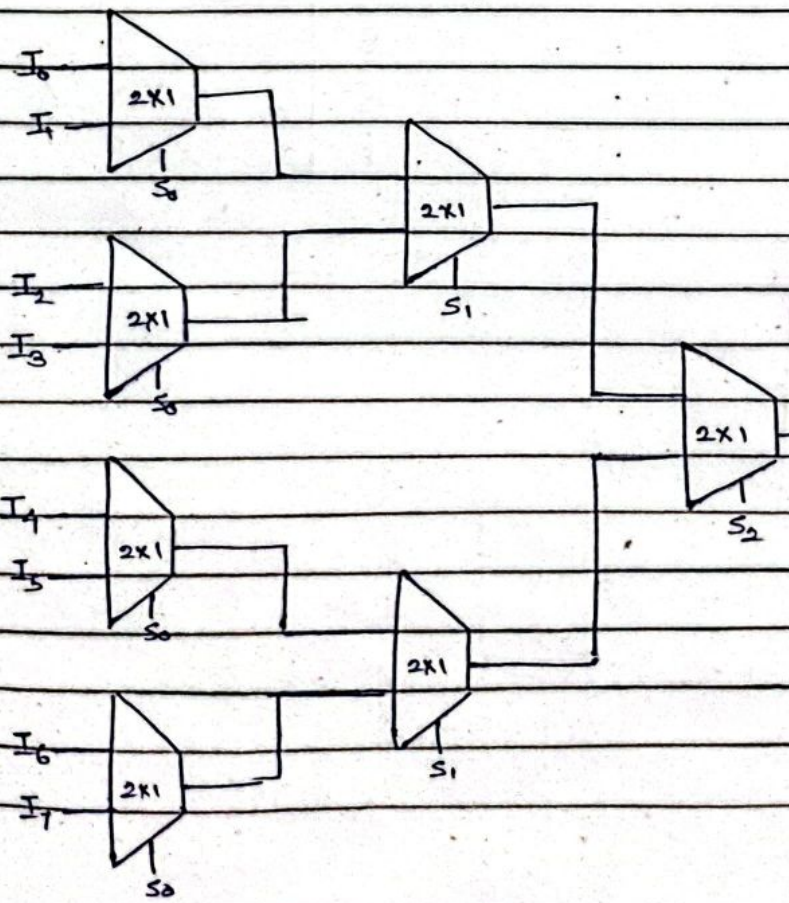


$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$

Implement 8X1 MUX using 2X1 MUX

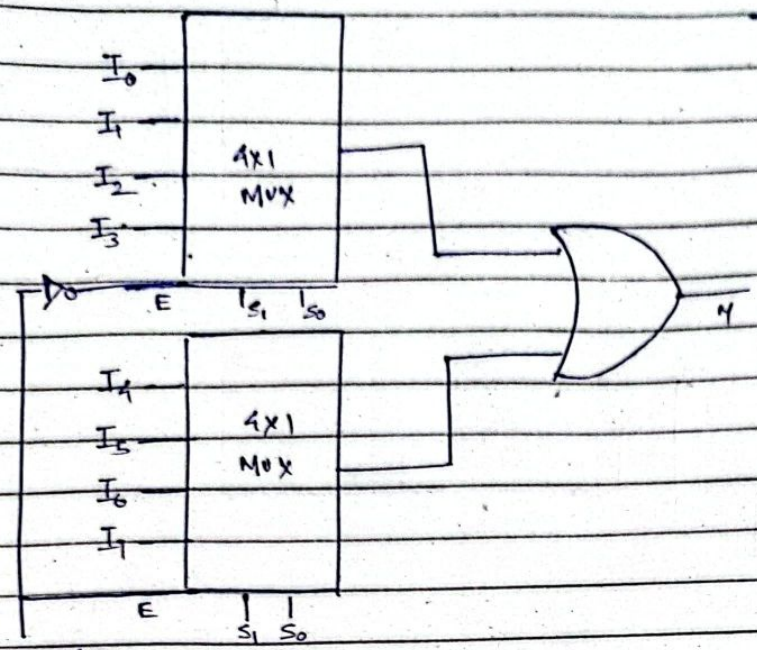
$8/2 = 4$
$4/2 = 2$
$2/2 = 1$ } $4+2+1 = 7 \rightarrow$ 2X1 MUX

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



$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_2 \bar{S}_1 S_0 I_1$
$+ \bar{S}_2 \bar{S}_1 \bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3$
$+ S_2 \bar{S}_1 \bar{S}_0 I_4 + S_2 \bar{S}_1 S_0 I_5$
$+ S_2 S_1 \bar{S}_0 I_6 + S_2 S_1 S_0 I_7$

8X1 MUX using 4X1 MUX



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

Signal

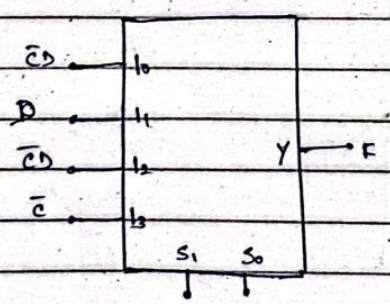
Implementation of boolean function using multiplexers

- Implement $F(A,B,C,D) = \Sigma m(1,4,5,7,9,12,13)$ using 4X1 MUX

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10	
$\bar{A}\bar{B}$ 00		1			$\bar{C}D = I_0$
$\bar{A}B$ 01	1	1	1		$\bar{C}+D = I_1$
AB 11	1	1			$\bar{C} = I_3$
$A\bar{B}$ 10		1			$\bar{C}D = I_4$

S_1	S_0	Y
0	0	$I_0 = \bar{C}D$
0	1	$I_1 = D$
1	0	$I_2 = \bar{C}D$
1	1	$I_3 = \bar{C}$

$S_0 = B$ $S_1 = A$



1 bit full adder using multiplexer

A	B	Cin	S	Co
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

For S

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

For Co

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

Design table for Sum

	BC_0	BC_1	BC_2	BC_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
Input to Mux	A	$\bar{A}$	$\bar{A}$	A

$BC_0 = A$ $BC_1 = \bar{A}$

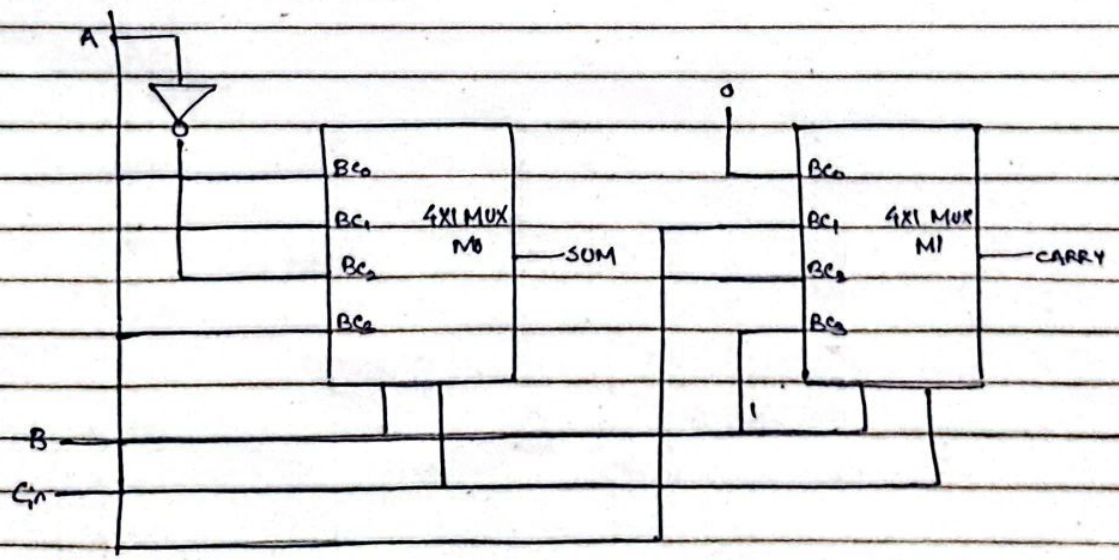
$BC_2 = \bar{A}$ $BC_3 = A$

Design table for Carry

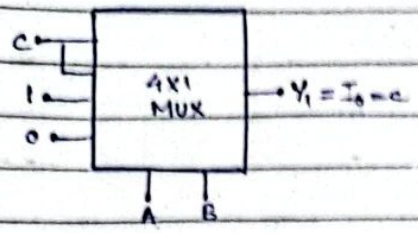
	BC_0	BC_1	BC_2	BC_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
INPUT TO MUX	0	A	A	1

$BC_0 = 0$ $BC_1 = A$

$BC_2 = A$ $BC_3 = 1$



Logical Expression from MUX

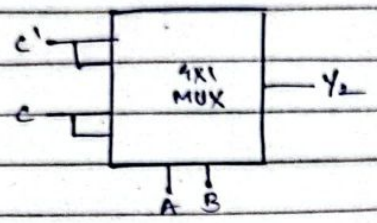
- 

$Y_1 = A'B'C + A'BC + AB'\cdot 1 + \cancel{A\cdot B\cdot 0}$

$Y_1 = A'B'C + A'BC + A\cdot B'$

$Y_1 = A'C(B'+B) + AB'$

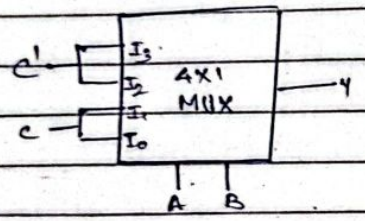
$\boxed{Y = A'C + AB'}$

- 

	A	B	Y_2
I_0	0	0	C'
I_1	0	1	C'
I_2	1	0	C
I_3	1	1	C

$Y_2 = A'C' + A\cdot C$

$Y_2 = A \odot C$

- 

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

$Y = A'B'C + A'BC + AB'C' + ABC'$

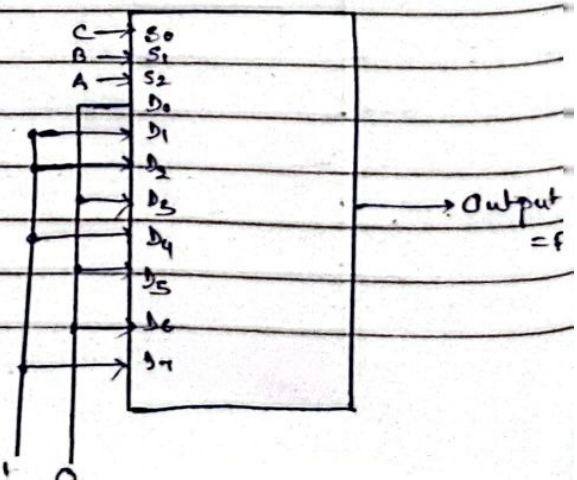
$Y = A'C(B'+B) + AC'(B'+B)$

$Y = A'C + AC'$

$Y = A \oplus C$

- Implement the function $F = A \oplus B \oplus C$ using multiplexer

S_2	S_1	S_0	$F = A \oplus B \oplus C$
A	B	C	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

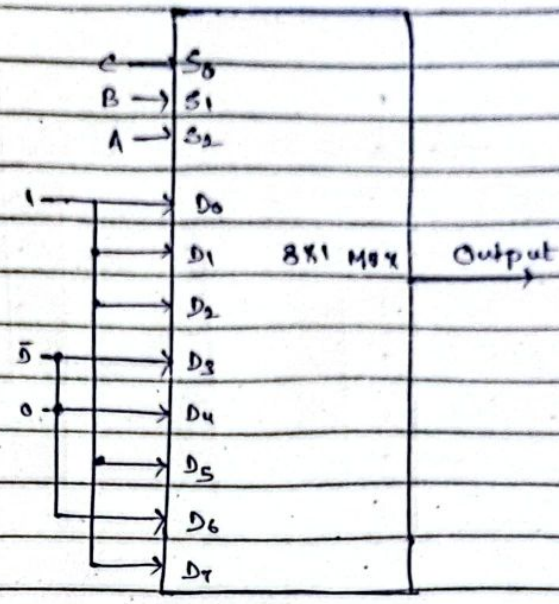


• Use a multiplexer having three data select inputs to implement the logic for a function given below. Also realise the same using a 16:1 MUX

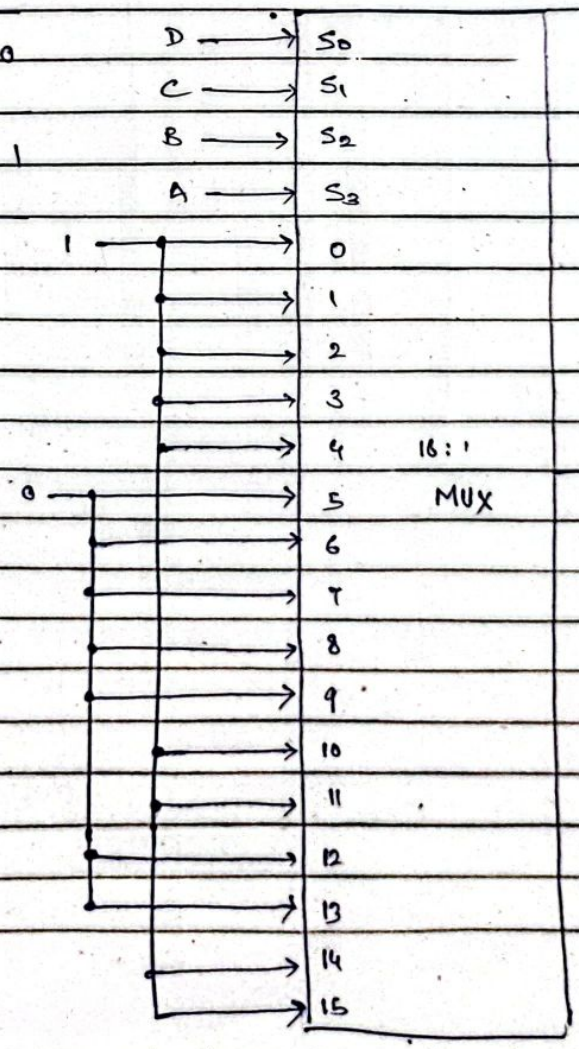
$$F = \Sigma m(0,1,2,3,4,10,11,14,15)$$

S_2	S_1	S_0			
A	B	C	D	F	
0	0	0	0	1	F=1
0	0	0	1	1	
0	0	1	0	1	F=1
0	0	1	1	1	
0	1	0	0	1	$F=\bar{D}$
0	1	0	1	0	
0	1	1	0	0	F=0
0	1	1	1	0	
1	0	0	0	0	F=0
1	0	0	1	0	
1	0	1	0	1	F=1
1	0	1	1	1	
1	1	0	0	0	F=0
1	1	0	1	0	
1	1	1	0	1	F=1
1	1	1	1	1	

(a) Truth Table



(b) Logic diagram of 8X1 MUX



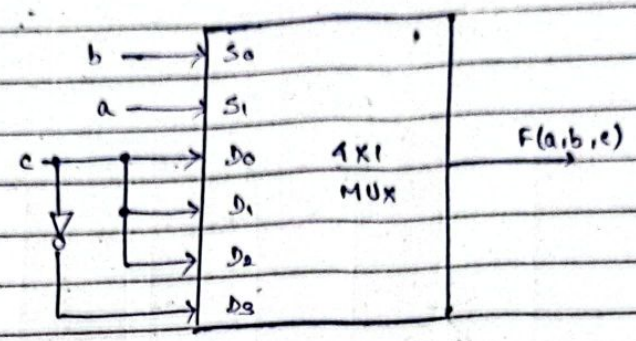
(c) logic diagram of 16X1 MUX

- Implement the following function using a MUX

$F(a,b,c) = \sum m(1,3,5,6)$

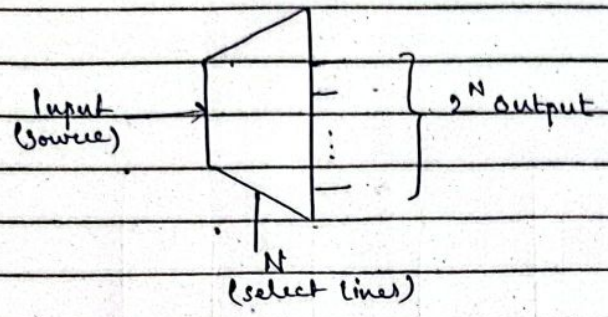
Choose a and b as select inputs

S_1	S_0		
a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



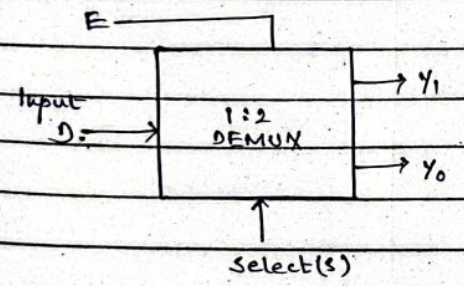
Demultiplexer

- Performs reverse operation of a multiplexer

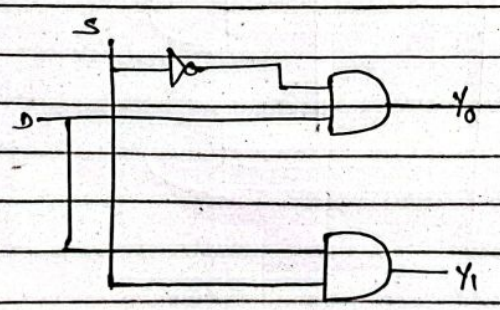


- **1 to 2 demultiplexer**

A 1 to 2 demultiplexer consists of one input line, two output lines and one select line. The signal on the select line helps to switch the input to one of the two outputs.



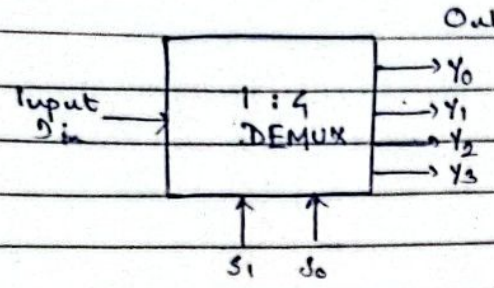
Select	Input	Output	
S	D	Y_1	Y_0
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0



- **1 to 4 Demultiplexer**

The 1 to 4 demultiplexer consists of one inputs, four outputs and two control lines to make selections.

It is also called 2 to 4 demultiplexer which means that two select lines and 4 output lines.



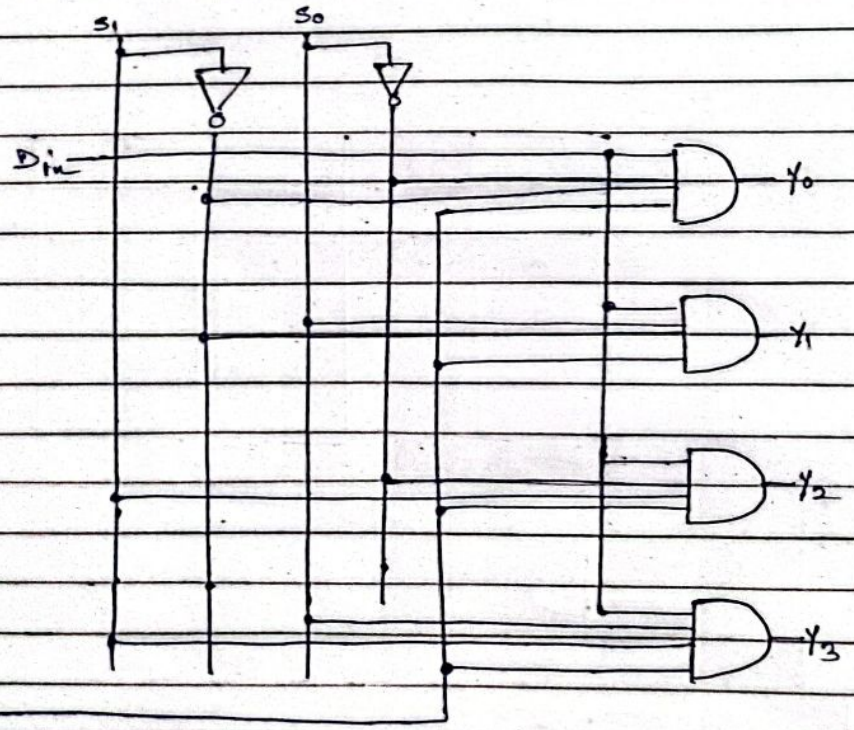
$Y_0 = \bar{S}_1 \bar{S}_0 D$

$Y_1 = \bar{S}_1 S_0 D$

$Y_2 = S_1 \bar{S}_0 D$

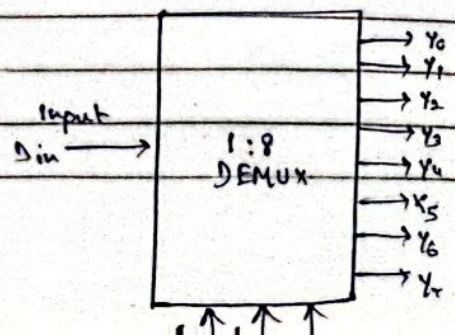
$Y_3 = S_1 S_0 D$

Data Input	Select Inputs		Outputs			
D	S_1	S_0	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0



- <u>1 to 8 Demultiplexer</u>

A 1 to 8 demultiplexer consists of one input line, 8 output lines and 3 select lines.



Data Input	Select Inputs			Outputs							
D	S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

Applications of Demultiplexers

i) Synchronous data transmission systems
ii) Boolean function implementation
iii) Data acquisition systems
iv) Combinational circuit design.

Multiplexer v/s Demultiplexer

Multiplexer	Demultiplexer
- Data selector	- Data distributor
- It is a digital switch	- It is a digital circuit
- It has n data input	- Single data input
- Single data output	- n data outputs
- Used at transmitter end	- Used at receiver end.

Decoder v/s Demultiplexer

	Decoder	Demultiplexer
Input/Output	• n input lines 2^n output lines	• n select lines 2^n output lines
Inverse of	• Encoder	• Multiplexer
Select lines	• Not contains	• Contains
Implementation	• Changing the format of instruction in the machine specific language	• Used as routing device to route the data coming from one signal into multiple signals.

■ Multiplexer v/s Decoder

• MUX accepts several inputs and allow only one data output.	• It takes n input binary code and convert it into corresponding outputs
• Used in data routing and waveform generation.	• Application of decoder is in decimal to BCD encoder
• Converts unary code into binary code.	• Converts binary code to unary code.

Flip Flops

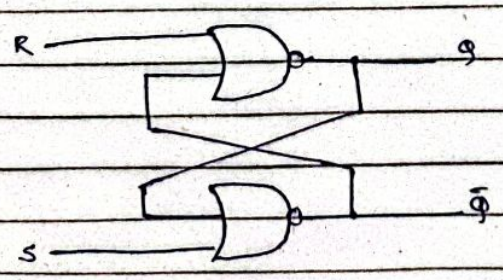
- Made up of logic gates which in turn made up of transistors.
- Binary storage device because they can store binary data (0 or 1)
- Edge sensitive / Edge triggered i.e. they are sensitive to the transmission rather than the duration or width of clock signal

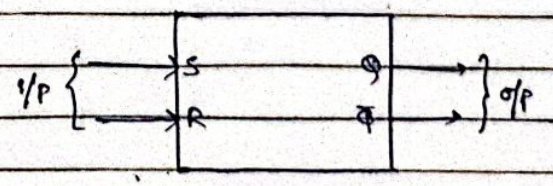
Types of flip flop
- → SR Flip Flop
- → JK Flip Flop
- → D Flip Flop
- → T Flip Flop

- **Latch:** Latch is used for certain flip flops or the simplest type of flip flop is called an SR latch. It refers to non clocked flip flops because these flip flops latch on to 1 or a 0 immediately upon receiving the input pulse called SET or RESET

Latch	Flip Flop
• Do not require clock signal	• They have clock signals
• A latch is an asynchronous device	• Flip Flop is a synchronous device.
• Level sensitive device (level triggering involved)	• Edge sensitive device (Edge triggering involved)
• Faster operation	• Slow operation due to clock
• Less power requirement.	• More power requirement

NOR Gate SR Latch (Active high SR latch)

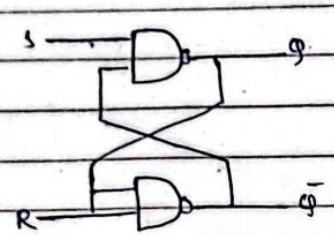




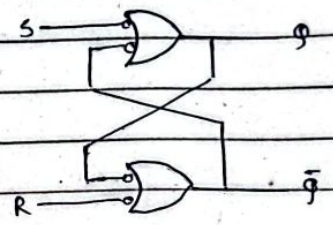
(logic diagram)

S	R	Q_n	Q_{n+1}	State
0	0	0	0	No change (NC)
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	X	Indeterminate (invalid)
1	1	1	X	

■ SR Latch (Active low SR Latch)



(using NAND Gates)

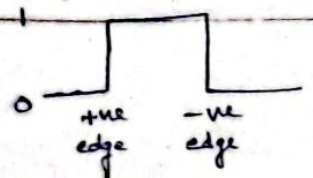


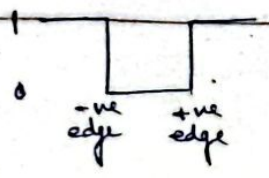
(using OR Gates)

S	R	Q_n	Q_{n+1}	State
0	0	0	X	Indeterminate
0	0	1	X	
0	1	0	1	Set
0	1	1	1	
1	0	0	0	Reset
1	0	1	0	
1	1	0	0	No change
1	1	1	1	

■ Clock pulse transition

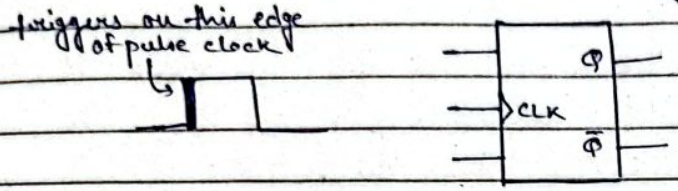
The movement of trigger pulse is always from 0 to 1 and then 1 to 0 of a signal. Thus, it takes two transitions in a single signal. When it moves from 0 to 1, it is called positive transition and when it moves from 1 to 0, it is called negative transition.



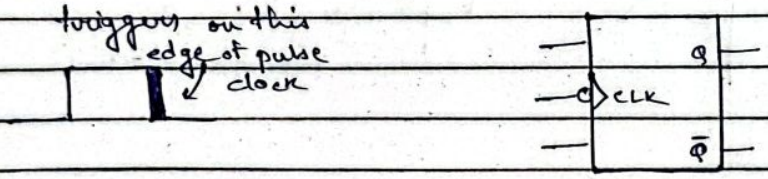


• Edge triggering : In a sequential circuit, if the output changes when the signal transmits from a high level to a low level or from a low level to a high level, it is called edge triggering.

- Positive edge triggering
When a FF is required to respond at a LOW to HIGH transition state, positive edge triggering is used. It is mainly identified from the clock input lead along with a triangle

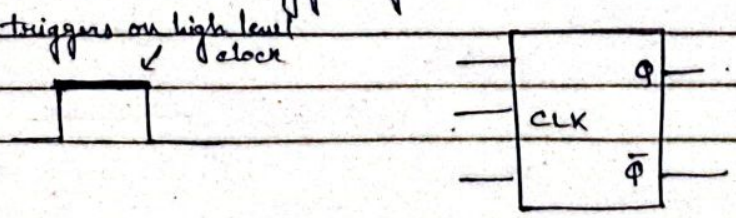


- Negative edge triggering
When a FF is required to respond at a HIGH to LOW transition state, negative edge triggering method is used. It is mainly identified from the clock input lead along with a low state indicator and a triangle.



• Level triggering : In a sequential circuit, if the output changes ~~when~~ during high voltage or low voltage period, it is called level triggering. In other words, output changes during either high voltage or low voltage period - not during the edges

- High level triggering
When a FF is required to respond at its high state, a HIGH level triggering method is used.

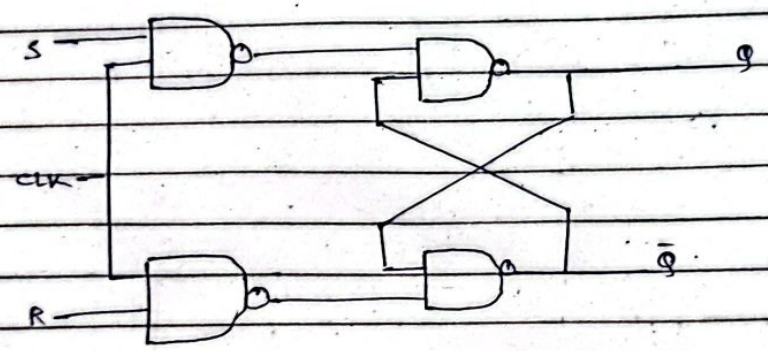


- Low level triggering

When a FF is required to respond at its LOW state, a LOW level triggering method is used. It is mainly identified from the clock input lead along with a low state indicator bubble.

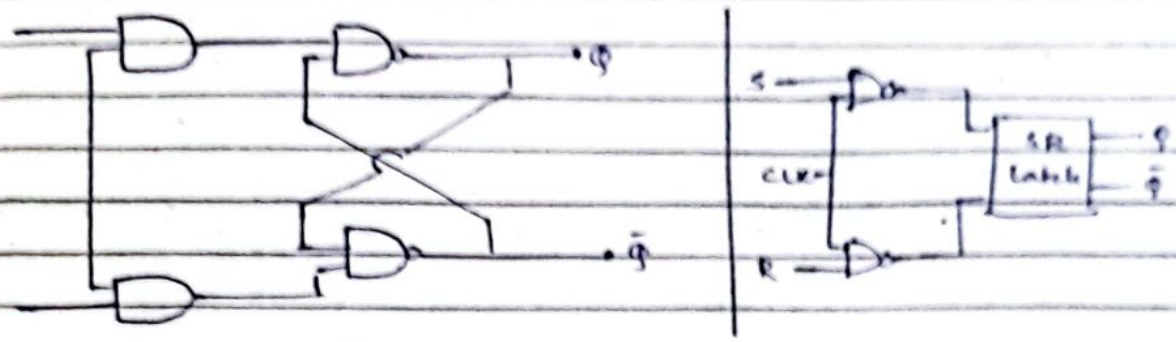
SR Flip Flop

SR flip flop works during the transition of clock pulse either from low to high or from high to low i.e: it can be either positive edge triggered or negative edge triggered.



Truth Table

INPUTS			OUTPUTS	STATE
CLK	S	R	Q	
X	0	0	No change	Previous
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	–	Forbidden



SR latch with NAND Truth Table

S^*	R^*	Q	$\bar{Q}$
0	0	Not used (Invalid)	
0	1	1	0
1	0	0	1
1	1	No change (As before)	

$S^* = \overline{(S.CLK)} = \bar{S} + \overline{CLK}$

$R^* = \overline{(R.CLK)} = \bar{R} + \overline{CLK}$

When CLK = 0

$S^* = \bar{S} + 1 \Rightarrow S^* = 1$

$R^* = \bar{R} + 1 \Rightarrow R^* = 1$

When CLK = 1

$S^* = \bar{S} + 0 \Rightarrow S^* = \bar{S}$

$R^* = \bar{R} + 0 \Rightarrow R^* = \bar{R}$

CLK	S	R	Q	$\bar{Q}$
0	X	X	Memory (As before)	
1	0	0	Memory (" ")	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid state	

$S^* = \bar{S}$, $R^* = \bar{R}$

$\left.\begin{matrix} S = 0 \\ R = 0 \end{matrix}\right\}$ $\left.\begin{matrix} S^* = 1 \\ R^* = 1 \end{matrix}\right\}$

$\left.\begin{matrix} S = 1 \\ R = 0 \end{matrix}\right\}$ $\begin{matrix} S^* = 0 \\ R^* = 1 \end{matrix}$

(*) Truth table for SR Flip Flop

CLK	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

$Q_{n+1} \rightarrow$ Next State

$Q_n \rightarrow$ Present State

Characteristic table of SR FF

Q_{n+1}	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Invalid (X)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Invalid (X)

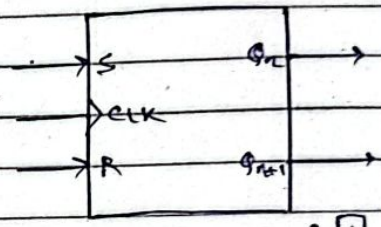
$\longrightarrow Q_{n+1} = Q_n$ (Memory)

Q_n \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$Q_{n+1} = S + Q_n \bar{R}$

Excitation table of SR FF

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

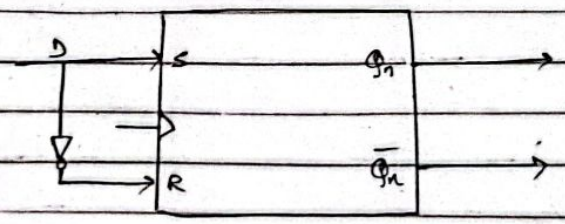


$S = Q_{n+1}$

$R = \overline{Q_{n+1}}$

D Flip Flop

- Most commonly used flip flop
- D stands for data



Truth table of D flip flop

CLK	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

Characteristic table of D flip flop

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

D = 0 means S = 0, R = 1

D = 1 means S = 1, R = 0

Now check the SR FF TT for Q_{n+1} value

$$\boxed{Q_{n+1} = D}$$

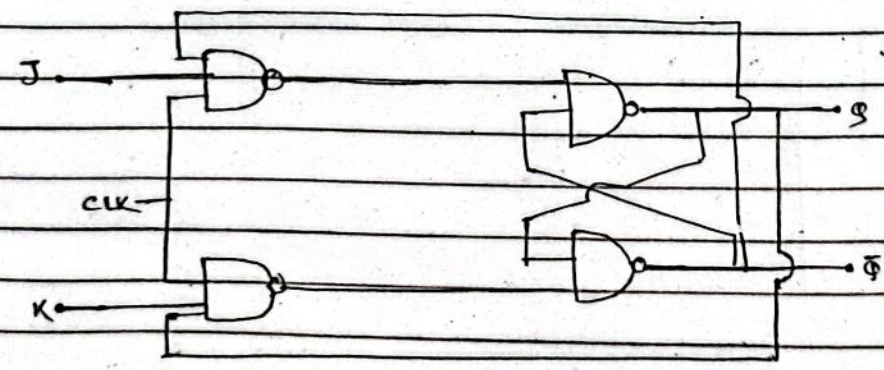
Excitation table of D flip flop

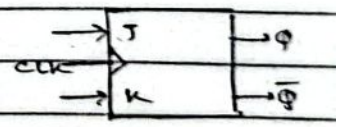
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Q_n \ D	0	1
0	0	1
1	0	1

JK Flip Flop

JK flip flop was introduced to end the error of SR flip flop (invalid state condition where both S=R=1)





CLK = 0 ⟶ $Q, \bar{Q}$ = Previous Input (Memory)

CLK = 1 ⟶ J = 0, K = 0 ⟶ $Q, \bar{Q}$ ⟶ ↑

CLK = 1, J = 1, K = 0 ⟶ Q = 1, $\bar{Q}$ = 0

CLK = 1, J = 0, K = 1 ⟶ Q = 0, $\bar{Q}$ = 1

CLK = 1, J = 1, K = 1 ⟶ $Q_{n+1} = \bar{Q}_n$ (complement of previous input) (toggle)

Truth table of JK Flip Flop

CLK	J	K	Q_{n+1}	
0	X	X	Q_n	(Memory → No change)
1	0	0	Q_n	
1	0	1	0	(Reset)
1	1	0	1	(set)
1	1	1	$\bar{Q}_n$	(Toggle)

Characteristic table of JK Flip Flop

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	$\bar{0} = 1$
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	$\bar{1} = 0$

Q_n \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$Q_{n+1} = \bar{Q}_n J + Q_n \bar{K}$

Excitation table of JK Flip Flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

For J

Q_n \ Q_{n+1}	0	1
0	0	1
1	X	X

$J = Q_{n+1}$

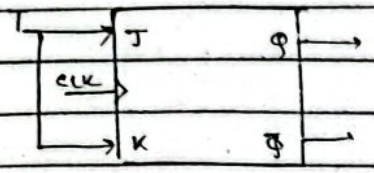
For K

Q_n \ Q_{n+1}	0	1
0	X	X
1	1	0

$K = \bar{Q}_{n+1}$

T Flip Flop

When only the toggling action is needed, we use T flip flop



When $T=0, J=0, K=1$

$T=1, J=1, K=1$

Truth table of T flip flop

clk	T	Q_{n+1}
0	x	Q_n (memory)
1	0	Q_n (")
1	1	$\bar{Q}_n$ (toggle)

Characteristic table

Q_n	T	Q_{n+1}
0	0	0
0	1	$\bar{0} = 1$
1	0	1
1	1	$\bar{1} = 0$

→ Memory

→ Toggle

Q_n \ T	0	1
0	0	1
1	0	1

Excitation table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

$T = \bar{Q}_n Q_{n+1} + Q_n \bar{Q}_{n+1}$

Flip Flop Conversion

Steps:

(i) Identify available and required flip flop

(ii) Make characteristic table for required flip flop.

(iii) Make excitation table for available flip flop.

(iv) Write boolean expression for available flip flop.

(v) Draw the circuit.

① JK Flip Flop to D Flip Flop Conversion

Available FF = JK

Required FF = D

- Characteristic table for D flip flop

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

$$\boxed{Q_{n+1} = D}$$

- Excitation table for JK flip flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Q_n	D	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

Writing Boolean Expression for available FF (JK)

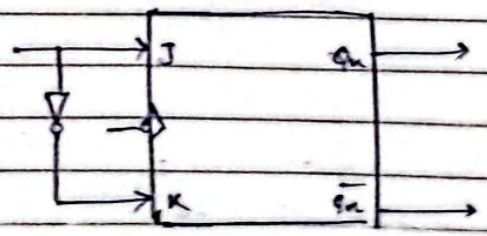
Q_n \ D	0	1
0	0	1
1	X	X

$J = D$

Q_n \ D		
	X	X
	1	0

$K = \bar{D}$

Circuit Diagram



② T Flip Flop to D Flip Flop Conversion

Available FF = T

Required FF = D

- Characteristic table for D flip flop

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

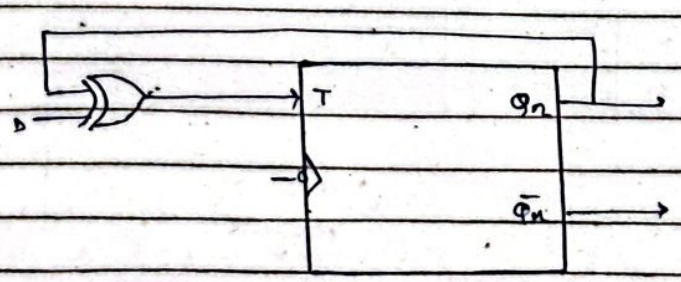
- Characteristic table for T flip flop

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Q_n	D	Q_{n+1}	T
0	0	0	0
0	1	1	1
1	0	0	1
0	1	1	0

Q_n \ D	0	1
0	0	1
1	1	0

$$T = \bar{Q}_n D + Q_n \bar{D} = D \oplus Q_n$$



③ SR Flip flop to JK flip flop conversion

Available FF = SR

Required FF = JK

- Characteristics table for JK Flip Flop

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

- Excitation table of SR flip flop

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Q_n	J	K	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

Q_n \ JK	00	01	11	10
0	0	0	1	1
1	X	0	0	X

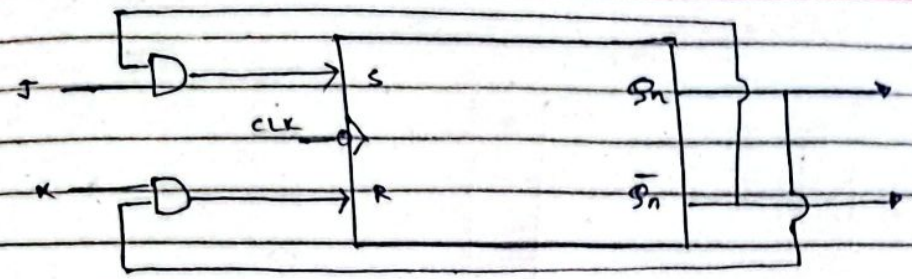
$S = \bar{Q}_n J$

For S

Q_n \ JK	00	01	11	10
0	X	X	0	0
1	0	1	1	0

$R = Q_n K$

For R



(4) SR Flip Flop to T flip flop conversion

Available FF = SR

Required FF = T

- Characteristic table for T flip flop

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

- Excitation table for SR flip flop

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

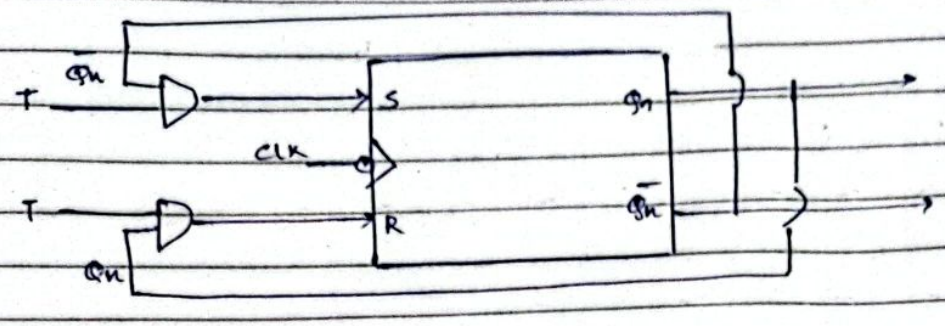
Q_n	T	Q_{n+1}	S	R
0	0	0	0	X
0	1	1	1	0
1	0	1	X	0
1	1	0	0	1

$Q_n \backslash T$	0	1
0	0	1
1	X	0

$S = \bar{Q}_n T$

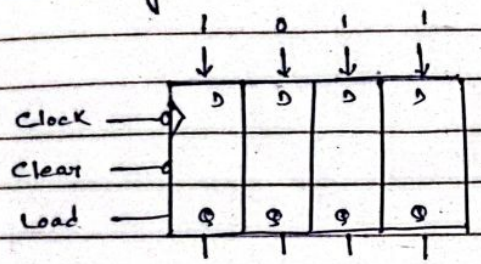
$Q_n \backslash T$		
	X	0
	0	1

$R = Q_n T$



REGISTER

Flip flop is used to store single bit binary data (either 0 or 1). To increase the storage capacity we have to use a group of flip flop. This group of flip flops is known as Register. The n bit register consist of n number of flip flops and is capable of storing n bit number.



We are bound to follow the clock, its internally connected to all the 4 flip flops and is applied at the same time to all the 4 ffs.

$f = 1\ MHz$

$T = 1/f\ sec = 1\ \mu sec$

After $1\ \mu sec$ the 4 bit no. stored in the register will change. So, if we want to store that combination for more time, there's a thing called 'Load'.

Load is generally two types
- → Synchronous (clock ↑ and load ↑)
- → Asynchronous (only load ↑)

Data formats and classification of registers

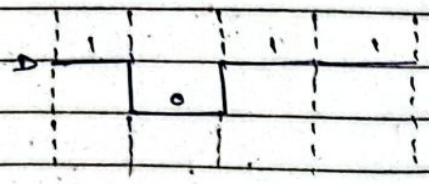
Data can be entered in two forms

① <u>Serial form</u> : (Temporal Code)

One bit at a time

Suppose we want to store 1011

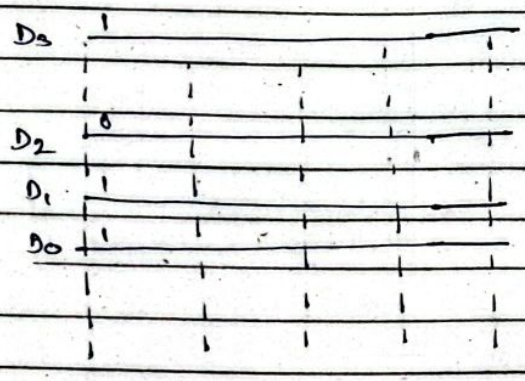
D = 1011

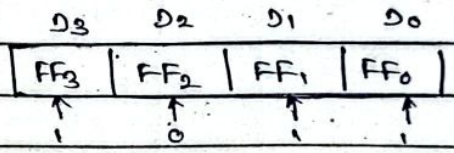


② <u>Parallel form</u> : (Special Code)

All bits at a time

Taking the same 4 bit no. 1011





■ <u>Classification of Registers</u>

(i) <u>Depending on input and output</u>

i) Serial Input Serial Output (SISO)

ii) Serial Input Parallel Output (SIPO)

iii) Parallel Input Serial Output (PISO)

iv) Parallel Input Parallel Output (PIPO)

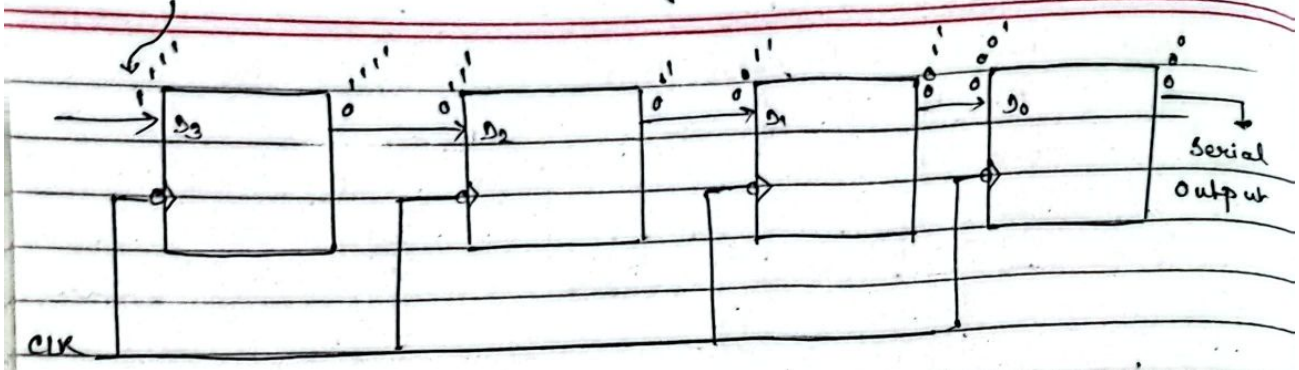
(ii) <u>Depending on applications</u>

i) Shift Register

ii) Storage Register

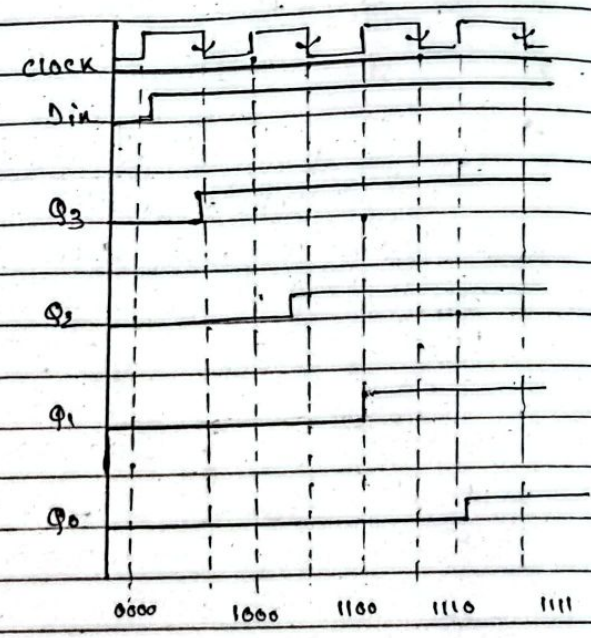
Shift Registers (Serial In Serial Out)

SISO



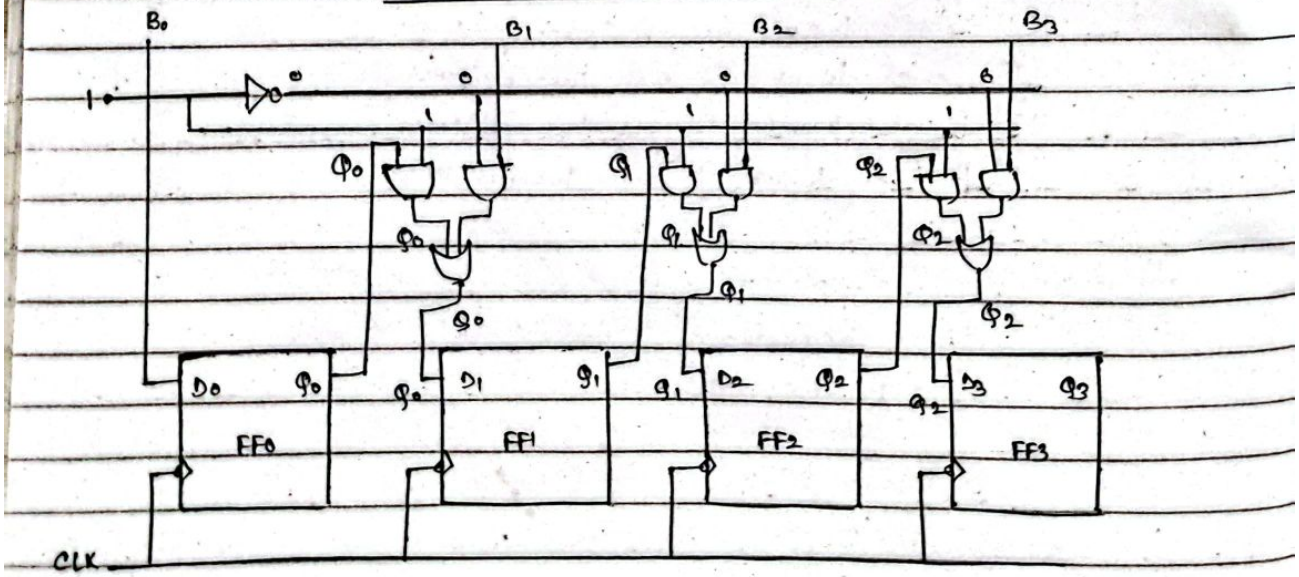
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

CLK	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1



⊛ SISO is more efficient in case of long distance data transmission.

(Parallel In Serial Out)



o/p of FF0 → Q_0 → Acting as i/p to FF1
o/p of FF1 → Q_1 → Acting as i/p to FF2
o/p of FF2 → Q_2 → Acting as i/p to FF3
previously stored data in FF3 → given as o/p → Q_3

Now check for a sequence 1011

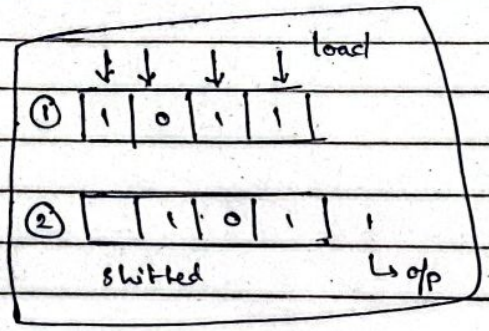
1 → B_0	1 → B_2
0 → B_1	1 → B_3

In first load case,

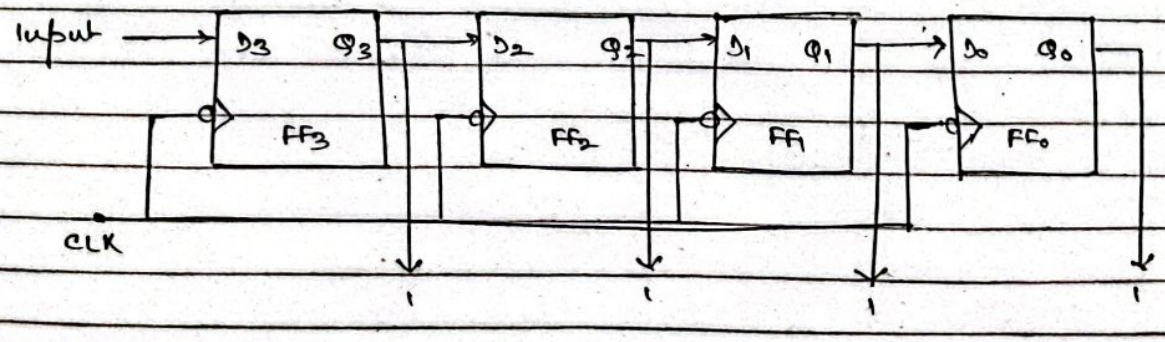
FF0 → 1	FF2 → 1
FF1 → 0	FF3 → 1

In 2nd case,

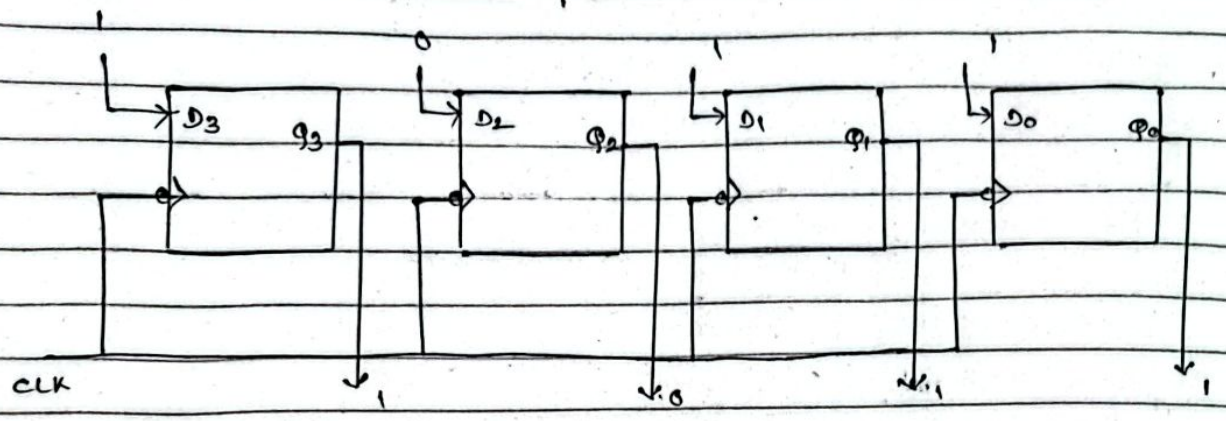
FF0 →	FF2 → 0
FF1 → 1	FF3 → 1
	o/p → 1



Serial In Parallel Out (SIPO)



Parallel Input Parallel Output (PIPO)



PIPO → Storage Register
Buffer register

$$\text{CLK}=1 \begin{cases} D=0 \Rightarrow Q_{n+1}=0 \\ D=1 \Rightarrow Q_{n+1}=1 \end{cases}$$

$$\text{CLK}=0 \Rightarrow D=X \Rightarrow Q_{n+1}=Q_n$$

Whatever we enter as the i/p in the flip flops, we get the exact value as the o/p from the flip flops. Thats why we call PIPO as 'Buffer Register'.

In PIPO, we require only one clock pulse to store the entire data which is more efficient than serial input where the no. of clock pulse is equal to the no. of bits we want to store.

4 Bit Bidirectional Shift Register

A Bidirectional shift register is one in which the data can be shifted either left or right. It can be implemented by using gating logic that enables the transfer of a data bit from one stage to the next stage to the right or to the left, depending on the level of control line.

$(3)_{10} \rightarrow 11$

$(6)_{10} \rightarrow 110$

$(3)_{10} \xrightarrow{\times 2} (6)_{10}$

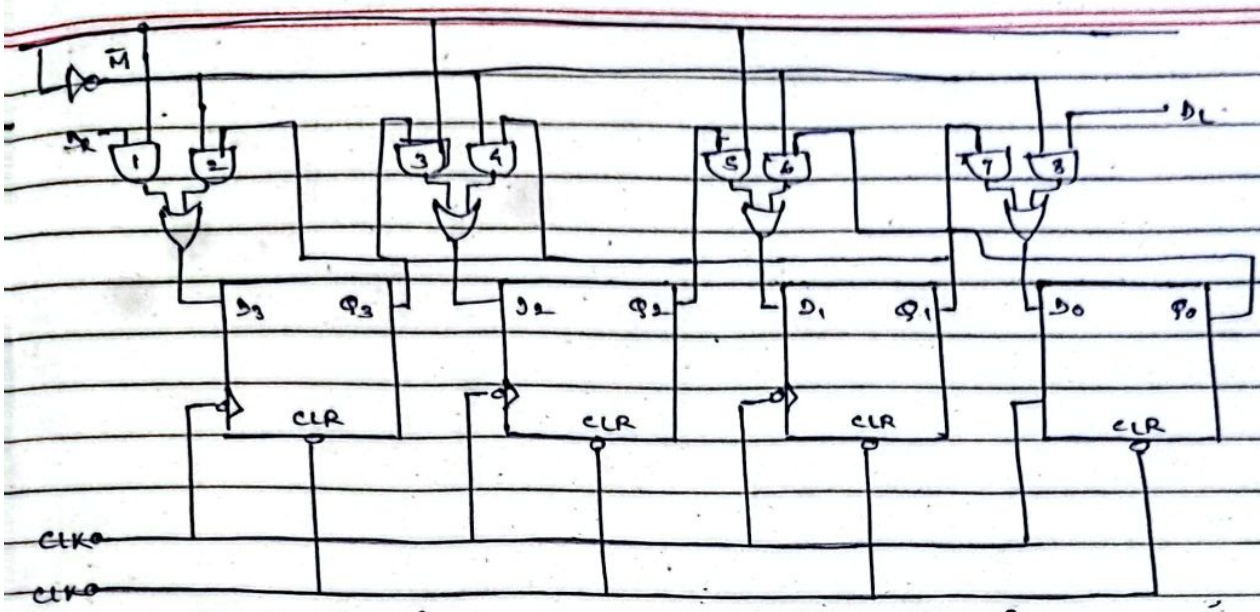
$(11)_2 \longrightarrow (110)_2$

$(6)_{10} \xrightarrow{\div 2} (3)_{10}$

$(110)_2 \longrightarrow (11)_2$

Shifting one bit to the left we've to multiply the (decimal × 2)

Shifting right → ÷ 2



Check $M=1$, $\bar{M}=0$ } Shift right operation

Check $M=0$, $\bar{M}=1$ } Shift left operation

Counter

- Counters are used to count number of clock
- Counters are used as Frequency Divider Circuit.
- Counters are also used in ADC
- Counters are also known as pulse stretcher circuit.
- They are used in RADAR for detection of range.

Types of counter

i) Asynchronous Counter: Only one flip flop having external clock and output of that flip flop will become clock for the next flip flop. So when clock applied only one flip flop work at that time.

ii) Synchronous Counter: All flip flops are connected with same clock. Hence when clock is applied all flip flops work simultaneously

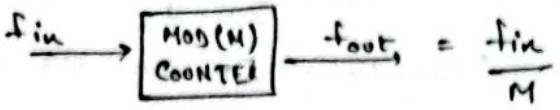
Synchronous Counter	Asynchronous Counter
① All the FFs are connected to the same clock	① Only one FF having external clock and o/p of that FF will be clock for the next FF.
② Faster	② Slower
③ All types of counting are possible.	③ Only increasing and decreasing counting are possible
④ No, transition error	④ Transition error can occur
⑤ Ex: Ring Counter, Johnson Counter	⑤ Ex: Ripple Counter

- Max. number of states $= 2^n$
  where n = total no. of flip flops.

- Total number of states used by the counter are called mod of the counter
  ↳ MODULUS of counter

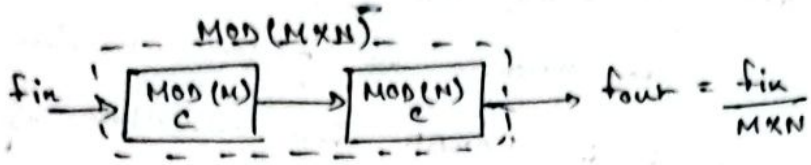
Note : 1. MOD (M) $\leq 2^n$

2.



$$f_{out} = \frac{f_{in}}{M}$$

3.



$$f_{out} = \frac{f_{in}}{M \times N}$$

Q BCD Counter

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001

MOD (10)

$M \leq 2^n$

$n \geq \log_2 M$

$n \geq 3.\text{something}$

$\boxed{n \approx 4}$

Q If mode 5 counter is cascaded with mode 2 counter, then it will become ?

(A) Mod 10 counter

(B) BCD Counter

(C) Both (A) and (B)

(D) None

Q MOD - 100 , No. of FFs = ?

$M \leq 2^n$

$n \geq \log_2 M$

$\geq \log_2 100$

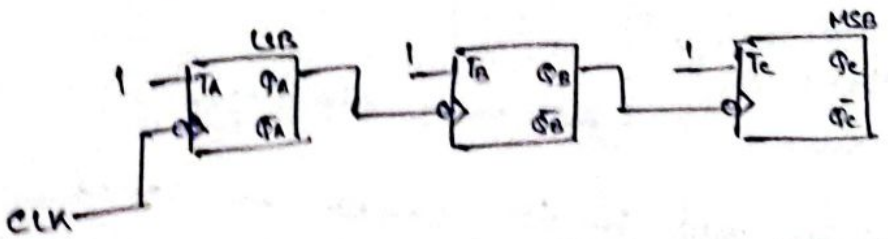
$\geq 6.\text{something}$

$\boxed{n \approx 7}$

Asynchronous Counter

↳ All the flip flops are used in toggle mode

3 bit ripple counter



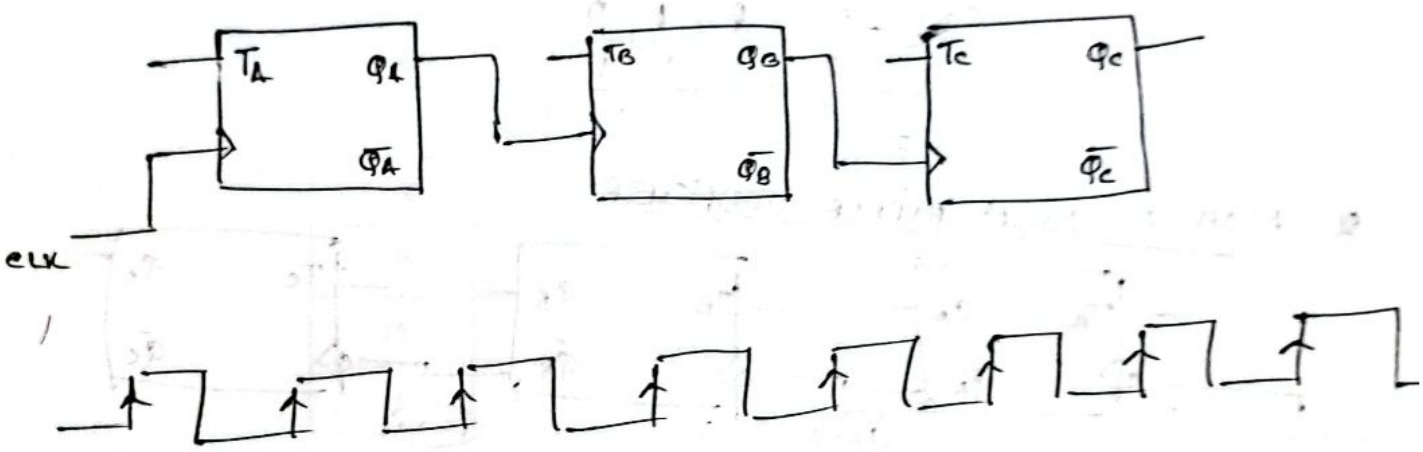
- Q_A will toggle when -ve edge at external clk appear
- Q_B will toggle when Q_A goes from 1 to 0
- Q_C will toggle when Q_B goes from 1 to 0

MOD 8 UP RIPPLE COUNTER

CLOCK	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0
9	0	0	1
10	0	1	0

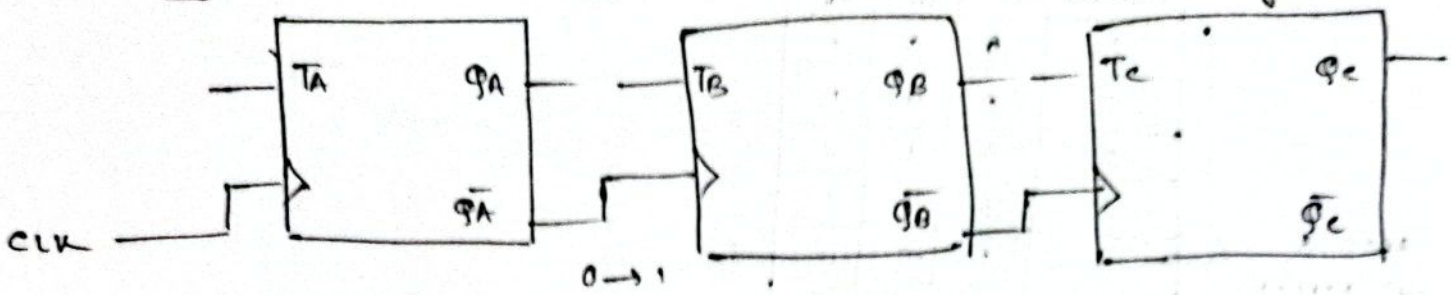
MOD 8 DOWN RIPPLE COUNTER

Positive Edge triggered



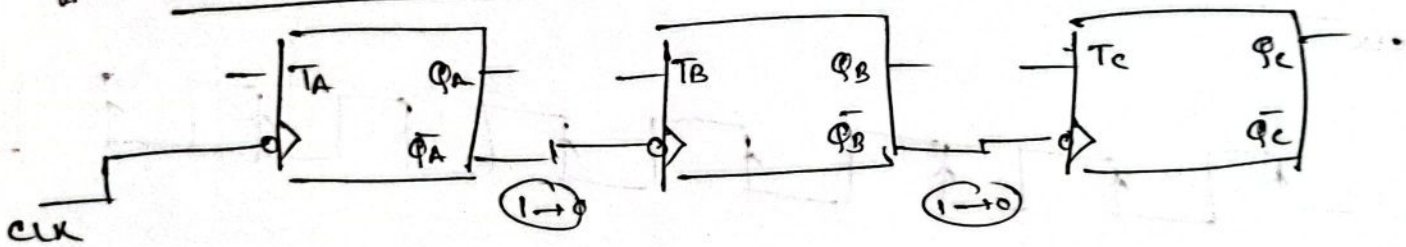
CLOCK	Q_C	Q_B	Q_A
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0
9	1	1	1
10	1	1	0

MOD 8 UP RIPPLE COUNTER . . . . Positive Edge.



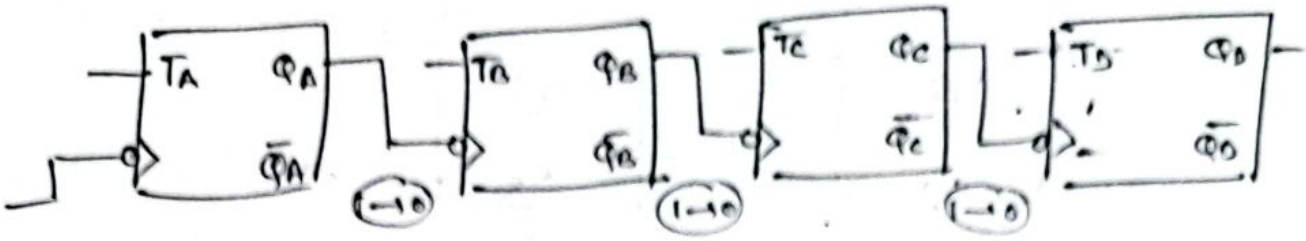
CLK	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	0
8	1	1	1
9			

MOD 8 DOWN RIPPLE COUNTER



CLK	Q_C	Q_B	Q_A
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

MOD (16) DOWN COUNTER



CLK	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

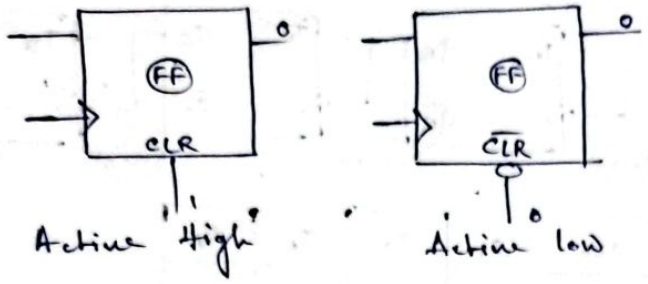
+ve → down
-ve → up

Full mode counter ⇒ when all states are used by the counter

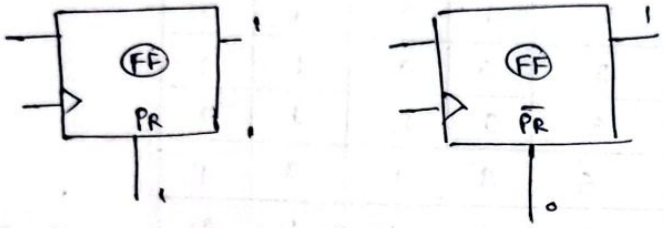
$n = 3$ MOD = 8
$n = 4$ MOD = 16
$n = 5$ MOD = 32

- Feedback reduces the number of states

RESET / CLEAR

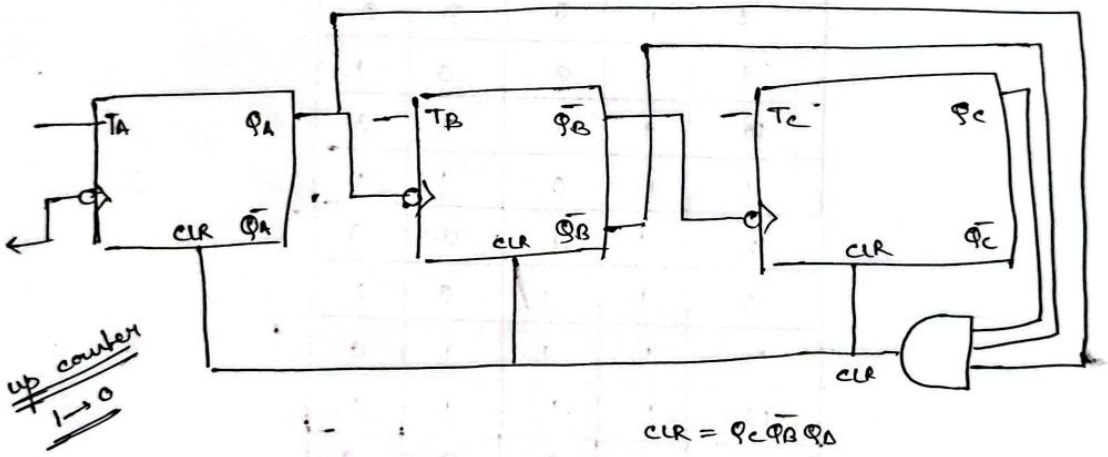


PRESET



Feedback Reduces the number of states

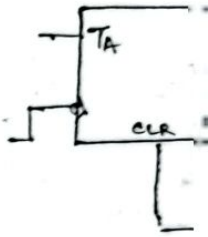
(Mod 5 Up Ripple Counter)



$CLR = Q_C \overline{Q_B} Q_A$

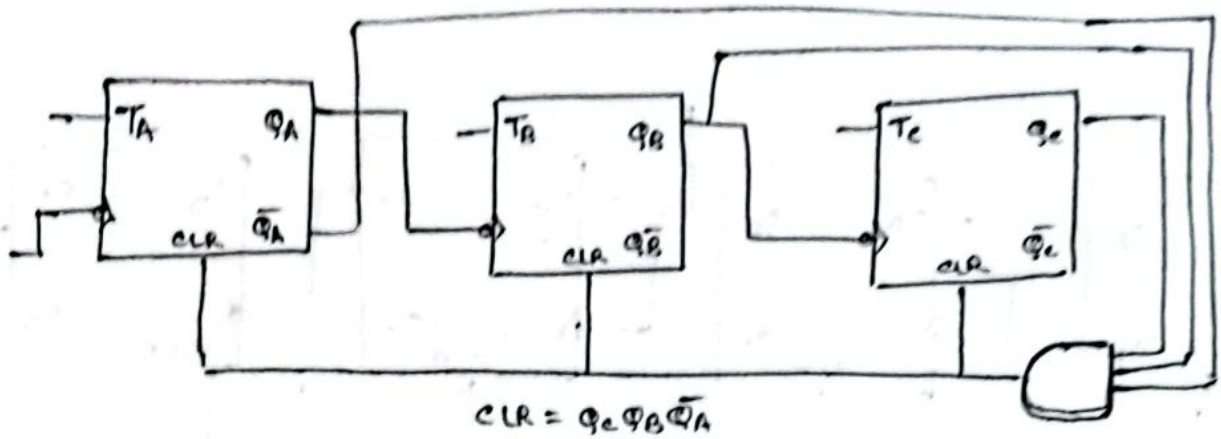
CLK	Q_C	Q_B	Q_A	$CLR = Q_C \overline{Q_B} Q_A$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0
8	0	0	0	1

MOD 6



MOD 9

MOD 6 UP RIPPLE COUNTER

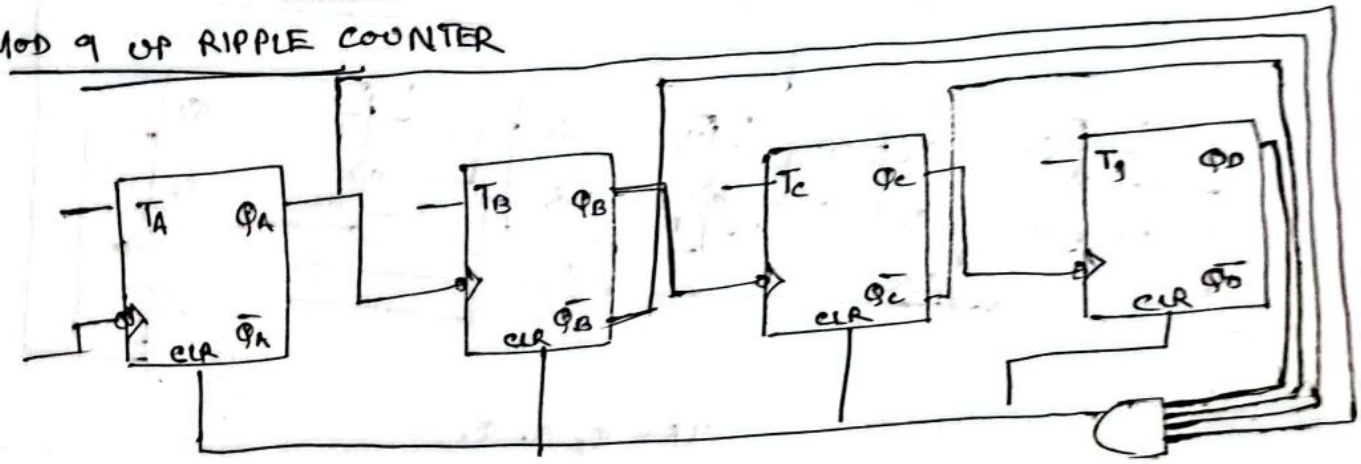


$CLR = Q_C Q_B \bar{Q}_A$

000 → 001 → 010 → 011 → 100 → 101

CLK	Q_C	Q_B	Q_A	CLR
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	~~1~~ 0	~~1~~ 0	~~0~~ 0	~~1~~ 0
7	0	0	1	0
8	0	1	0	0

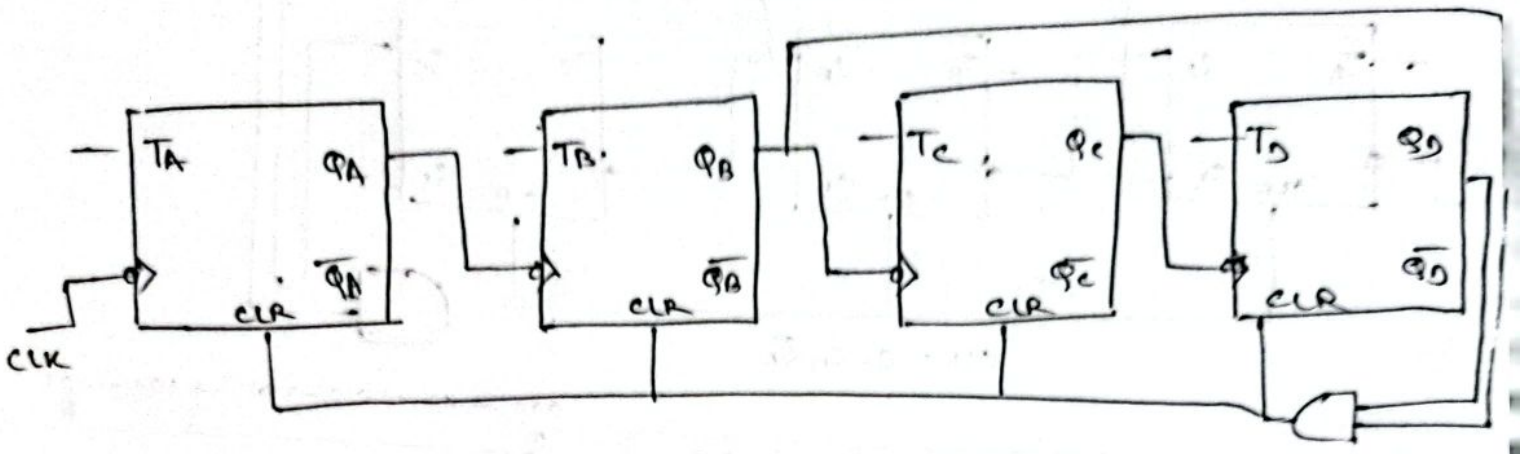
MOD 9 UP RIPPLE COUNTER



CLK	Q_D	Q_C	Q_B	Q_A	CLR
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0

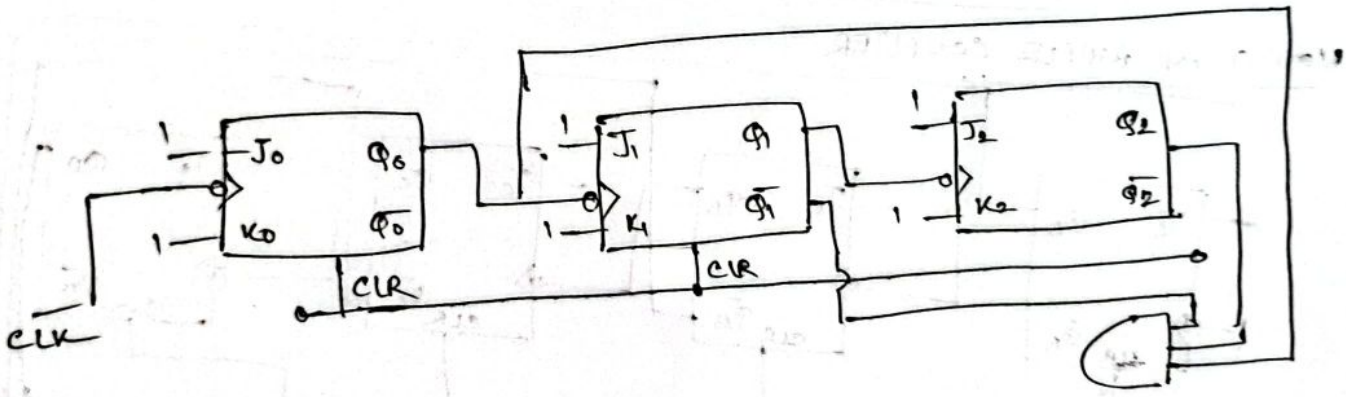
CLK	Q_D	Q_C	Q_B	Q_A	CLR
9	~~1~~ 0	~~0~~ 0	~~0~~ 0	~~1~~ 0	~~1~~ 0
10	0	0	0	1	0
11	0	0	1	0	0

Design a BCD Ripple Counter



$$CLR = Q_D \overline{Q_C} Q_B \overline{Q_A}$$
$$= 1\ 0\ 1\ 0$$

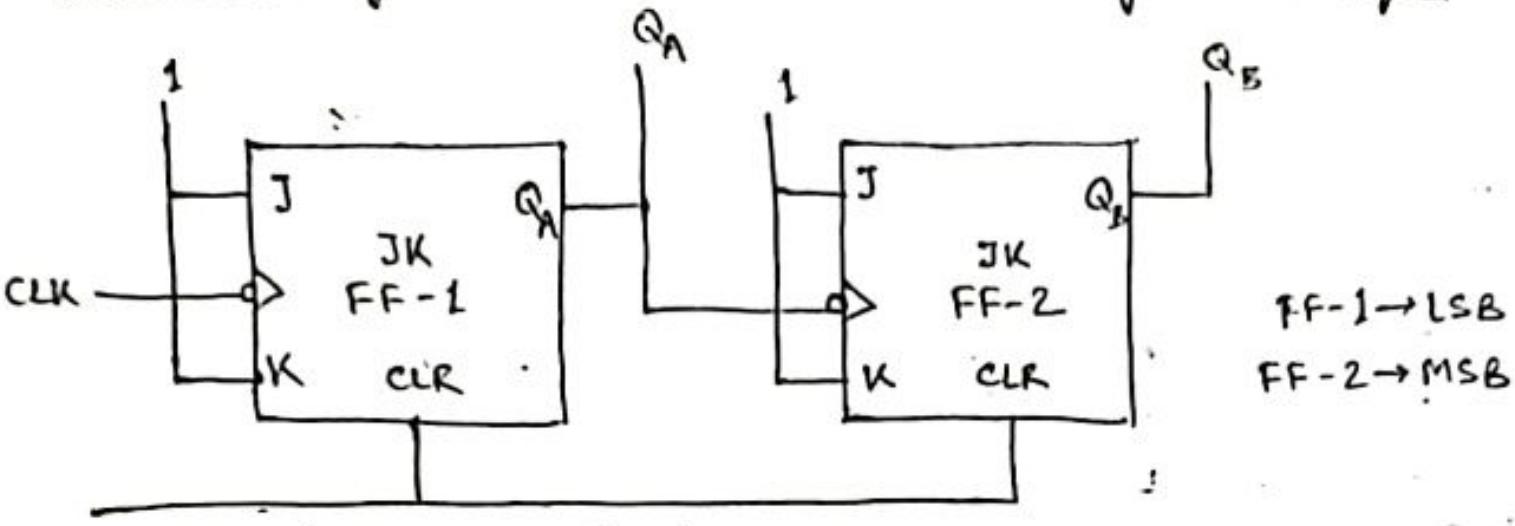
Q Which type of counter is shown below ?



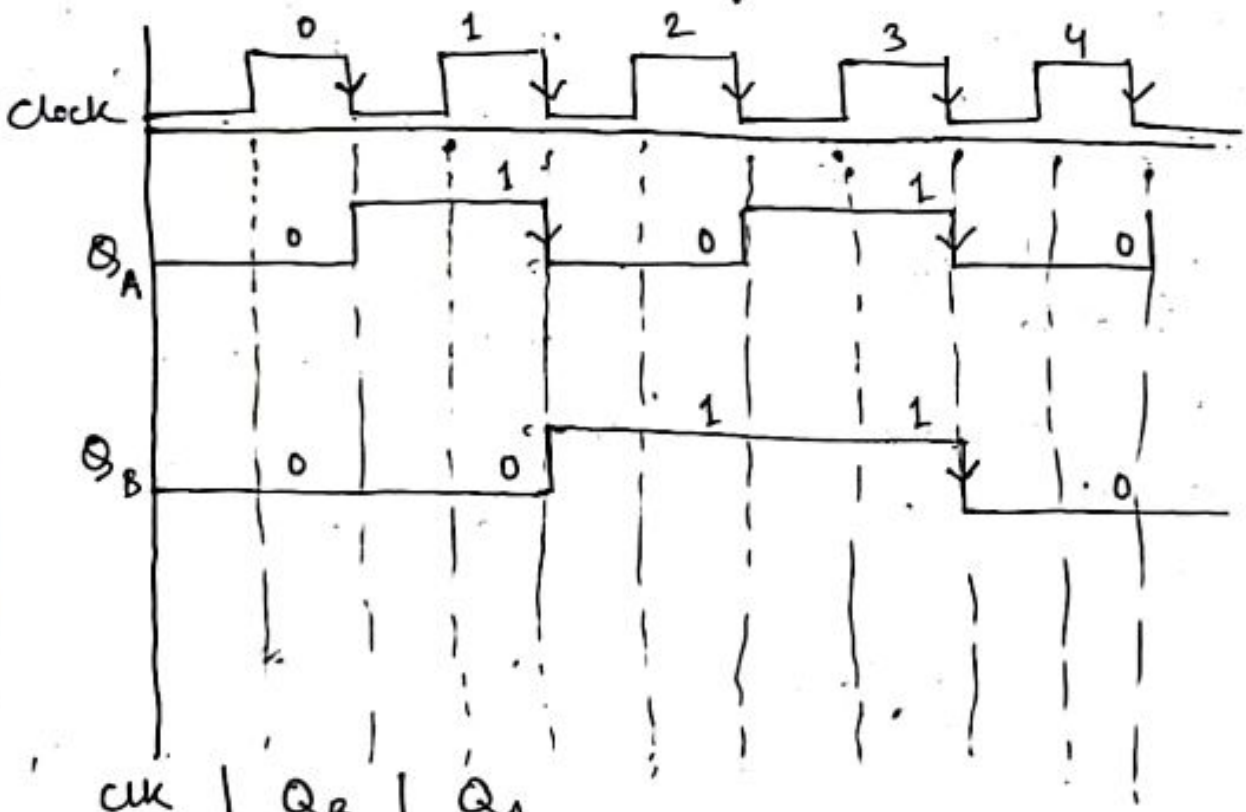
$$CLR = Q_2 \overline{Q_1} Q_0$$

Mod 5 UP Counter

2 bit Binary Up Ripple Counter Using -ve Edge



At the every clock pulse the output of the FF should toggle, in that case the next FF will be able to receive the clock signal.

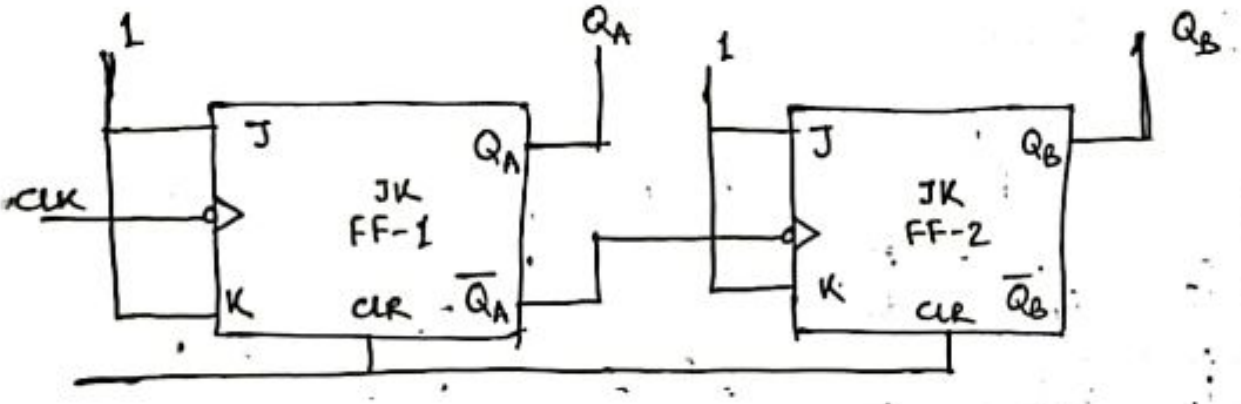


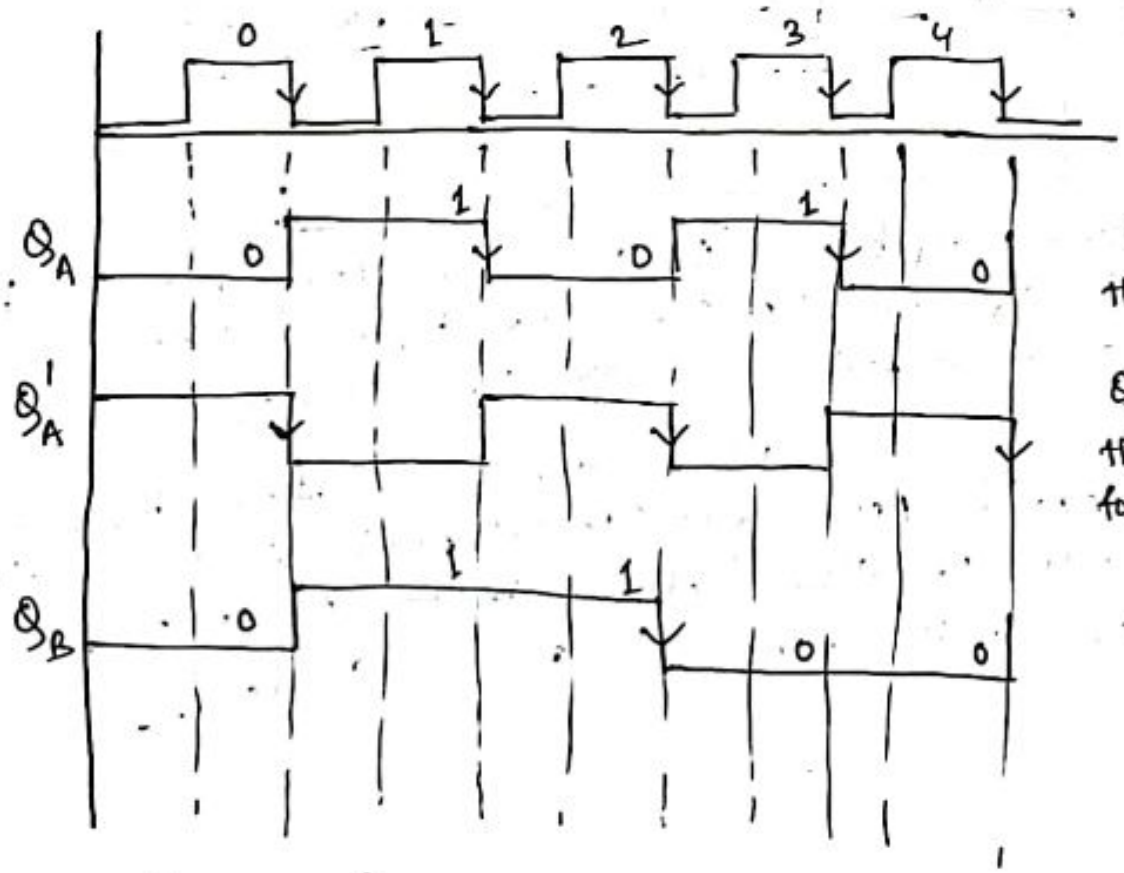
CLK	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0

4 clock pulses

Here the up counter is counting from 0 to 3 and after 4 clock pulses it'll repeat again

2 bit Binary Down Ripple Counter Using -ve Edge





Q_A' will be the opposite of Q_A

Q_A' output is the clock pulse for JK-FF-2

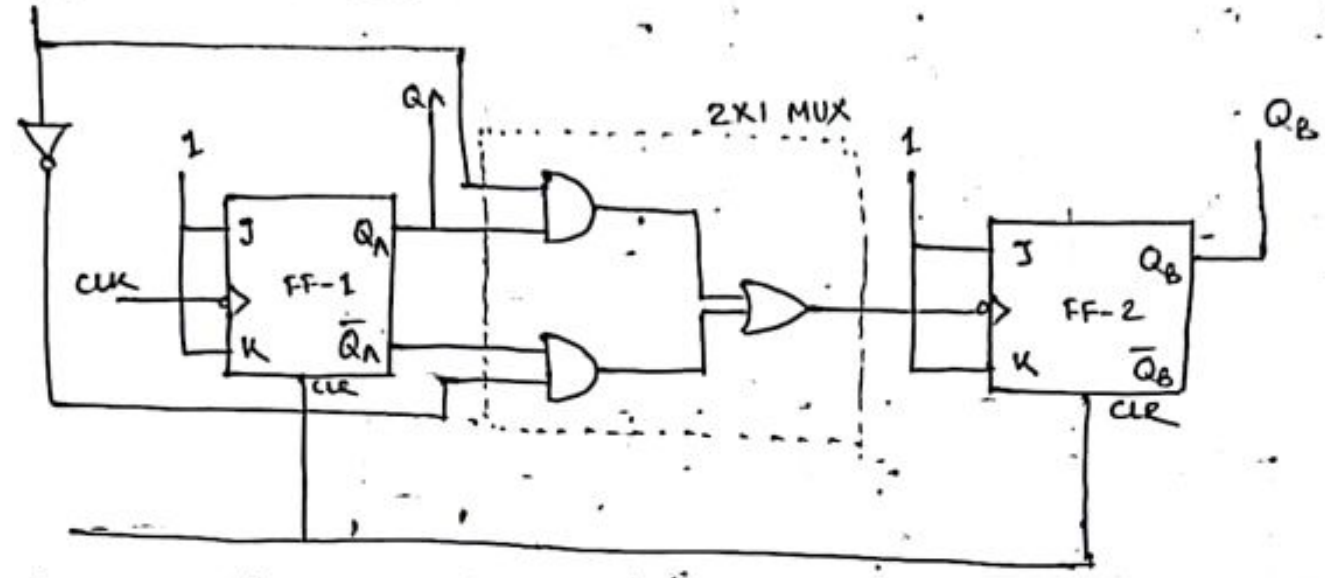
CLK	Q_B	Q_A	
0	0	0	(00)→0
1	1	1	(11)→3
2	1	0	(10)→2
3	0	1	(01)→1
4	0	0	(00)→0

5 clock pulses

Here the down counter is counting from 3 to 0 and after 5 clock pulses it'll start counting again (repeat)

2 bit Binary Up/ Down Ripple Counter Using -ve Edge

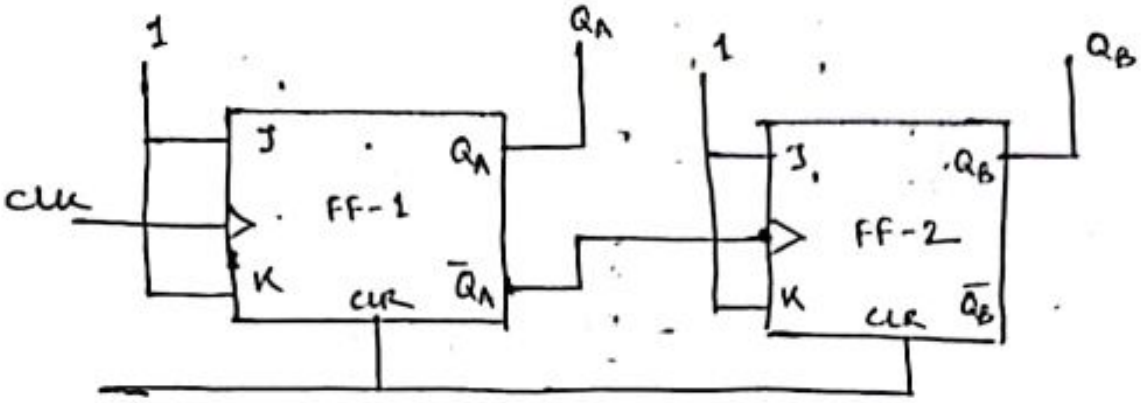
$U/\bar{D} \rightarrow$ Select Input

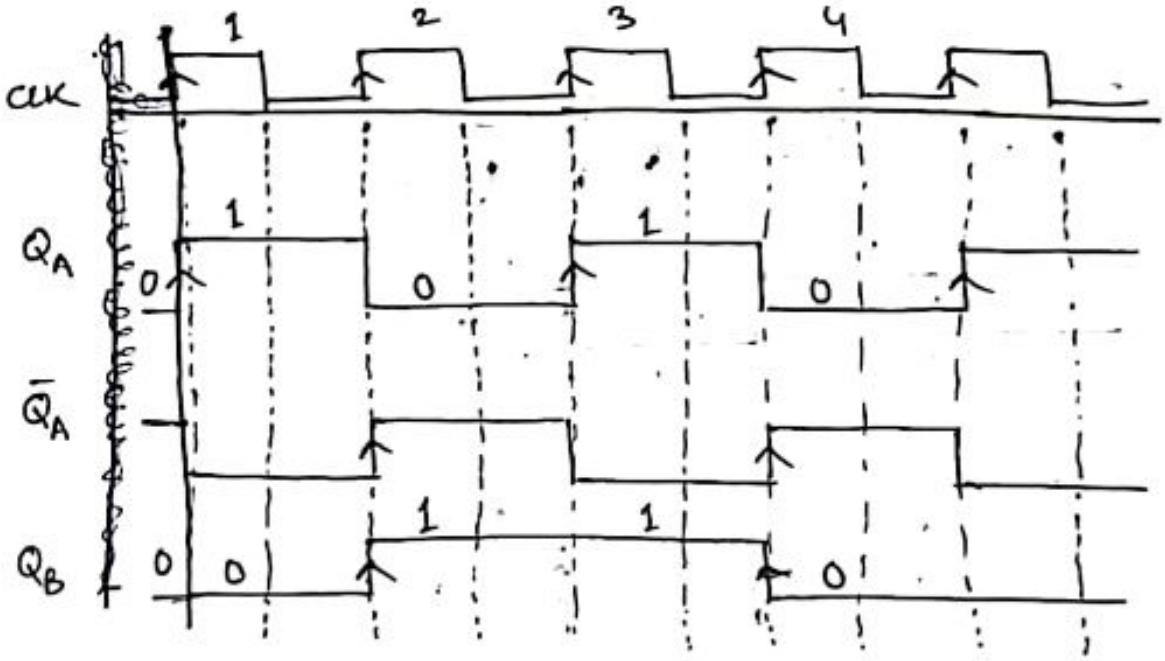


When selection input is '1' the first AND gate will be selected and Q_A will be the clock input for FF-2. And the counter will work as Up Counter.

When the selection input is '0' the second AND gate (In 2X1 MUX) will be selected and $\bar{Q}_A$ will be the clock input for FF-2. And the counter will work as Down Counter.

2 bit Binary Up Ripple Counter Using +ve Edge



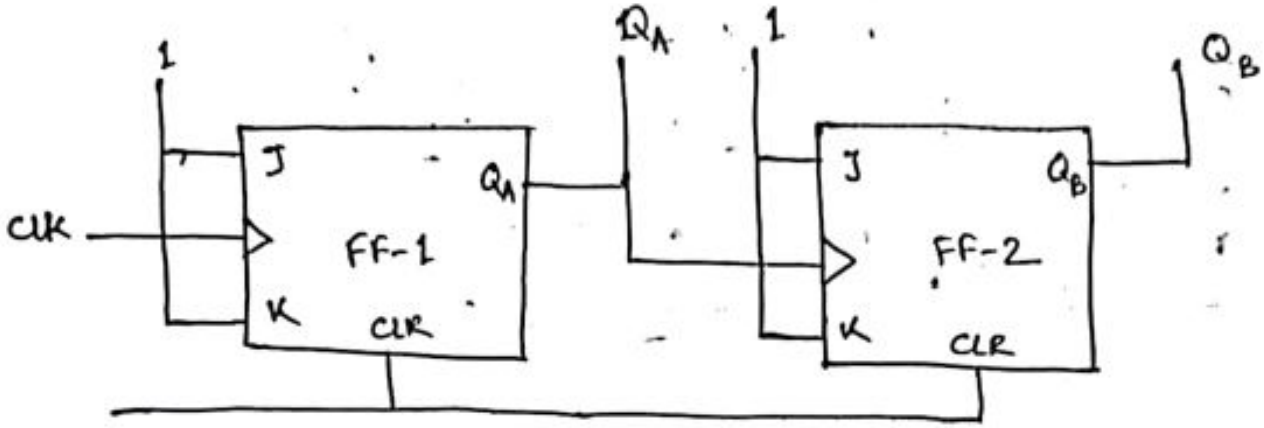


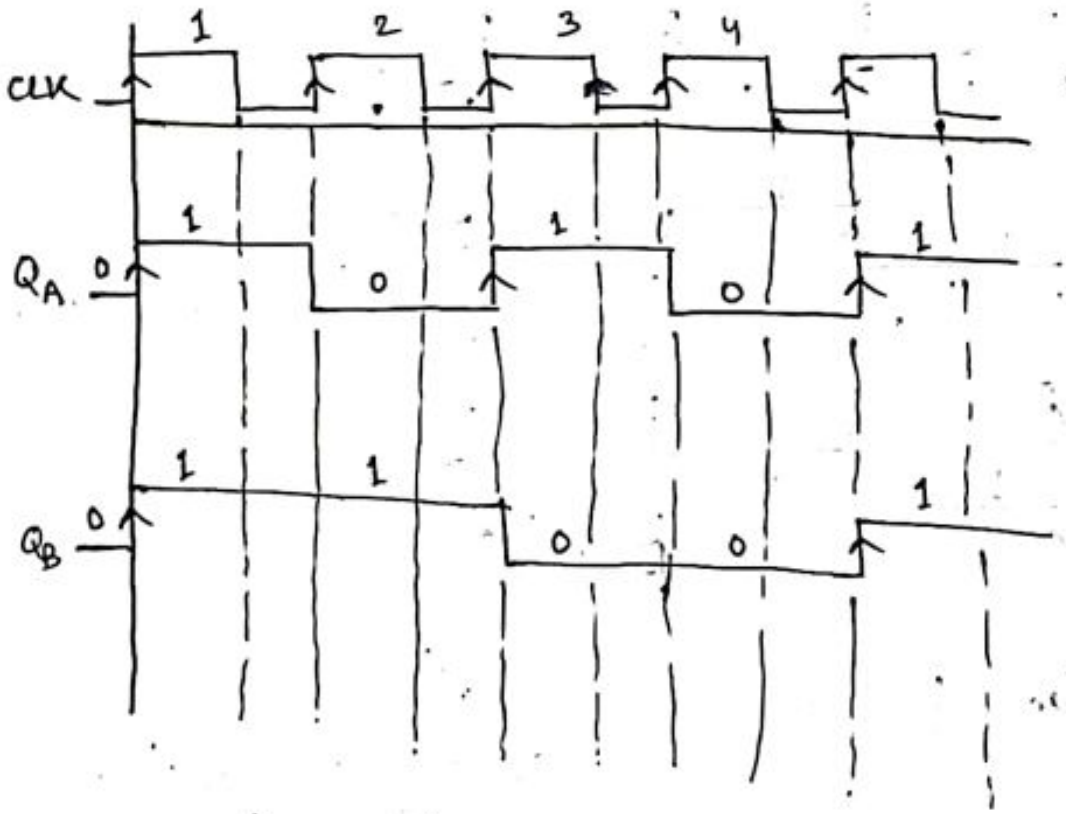
$\bar{Q}_A$ will be opposite of Q_A

CLK	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0

Counting from 0 to 3 using +ve edge and then repeating....

2 bit Binary Down Ripple Counter Using +ve Edge

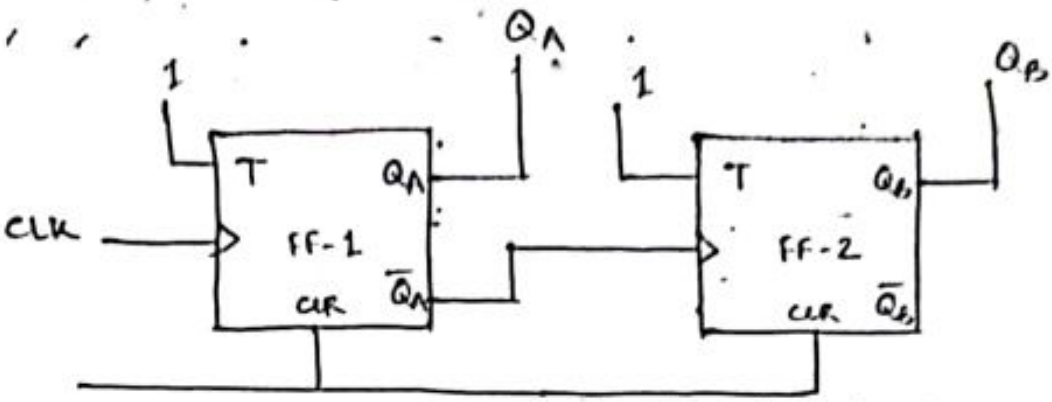




CLK	Q_B	Q_A	
0	0	0	
1	1	1	(11) → 3
2	1	0	
3	0	1	
4	0	0	(00) → 3
5	1	1	

Counting from 3 to 0 and then repeating....

2 bit Binary Up ~~Down~~ Ripple Counter Using +ve Edge (T FF)



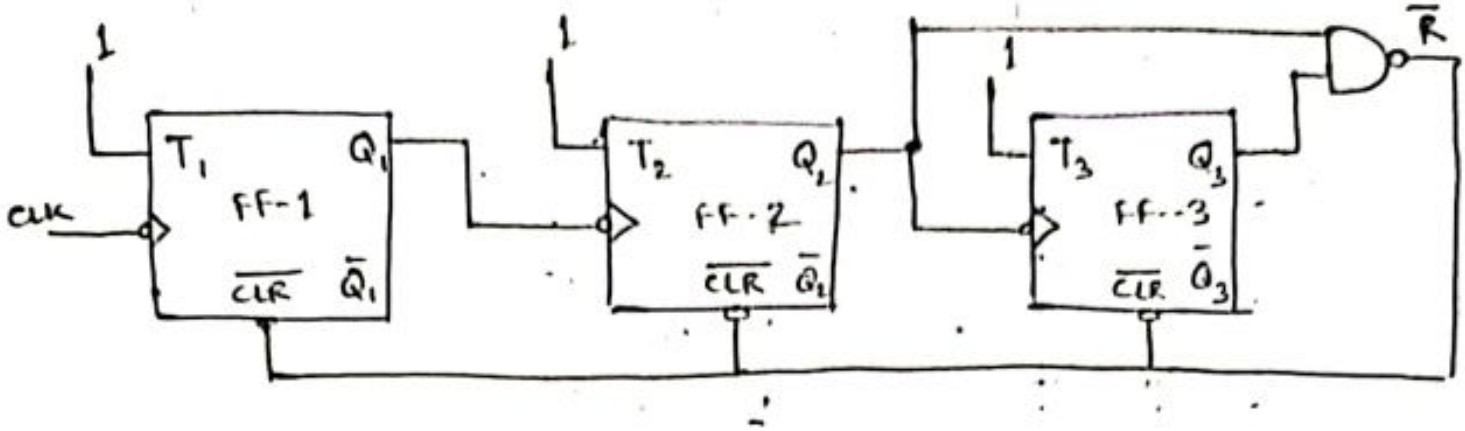
Design of a Mod-6 Counter (Asynchronous) Using T FFs

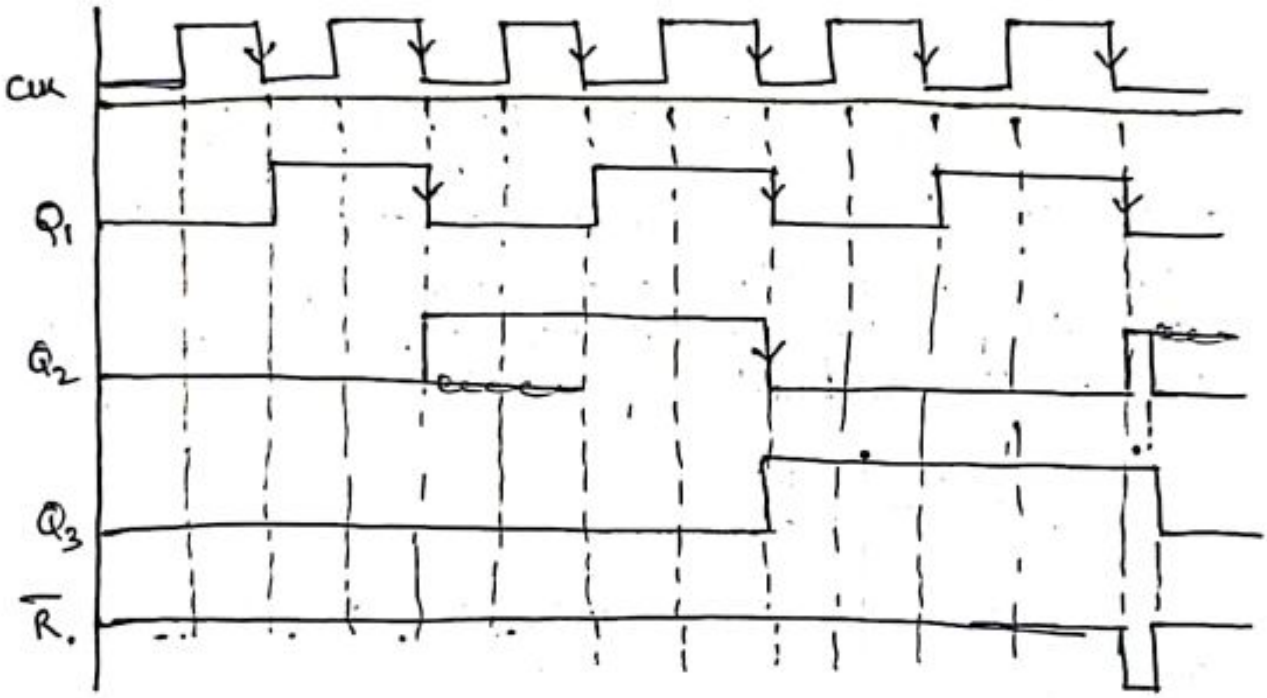
A mod 6 counter has six stable states 000, 001, 010, 011, 100, 101. When the sixth clock pulse is applied the counter temporarily goes to 110 but immediately resets to 000 bcz of the feedback provided.

CLK	Q_3	Q_2	Q_1	R
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
	↓	↓	↓	
	0	0	0	0
7	0	0	1	0

Q_3 \ Q_2Q_1	00	01	11	10
0				
1			X	X

$$R = Q_3Q_2Q_1 + Q_3Q_2\bar{Q}_1$$
$$= Q_3Q_2(Q_1 + \bar{Q}_1)$$
$$\therefore R = Q_3Q_2$$





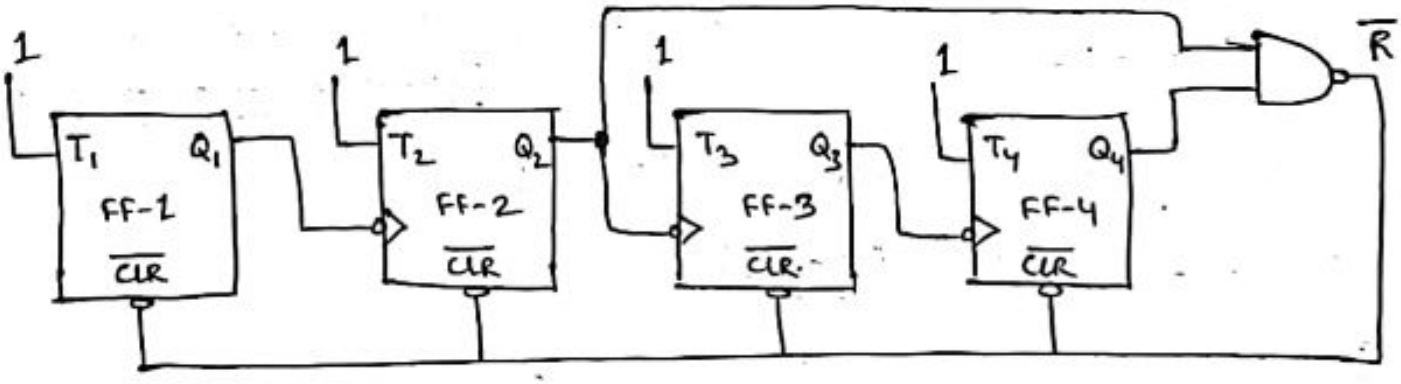
Design of a Mod 10 Asynchronous Counter Using T FFs

A mod 10 counter is a decade counter. It is also called a BCD counter or a divisible by 10 counter. It requires four FFs. There are 16 possible states out of which there are ten are valid and the remaining six are invalid. The counter has ten stable states from 0000 to 1001 which counts from 0 to 9. After ten clock pulses the counter goes to 1010 temporarily but bcz of the provided feedback it resets to the 0000 state.

CLK	Q_4	Q_3	Q_2	Q_1	R
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	0	0	0	0	1 → (Reset state)

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01				
11	X	X	X	X
10			X	X

$R = Q_4Q_2$



Effects of Propagation Delay of Ripple Counters

→ (Read from PDF)

Synchronus Counters

1. Define the no of bits or no of flip-flops required for the design.
2. Design the state diagram.
3. Choose the type of flip-flop. (If not mentioned then try to choose → D/T FF)
4. Design the excitation table or present state - next state table.
5. Find the minimal expression, using K-map.
6. Design the logic circuit.

Bits vs States

2 bit → 4
3 bit → 8
4 bit → 16

PS Q_n	NS Q_{n+1}	J	K
0	0	0	X
1	0	X	1
0	1	1	X
1	1	X	0

Excitation Table of JK FF

• 3 bit Synchronus Up Counter :

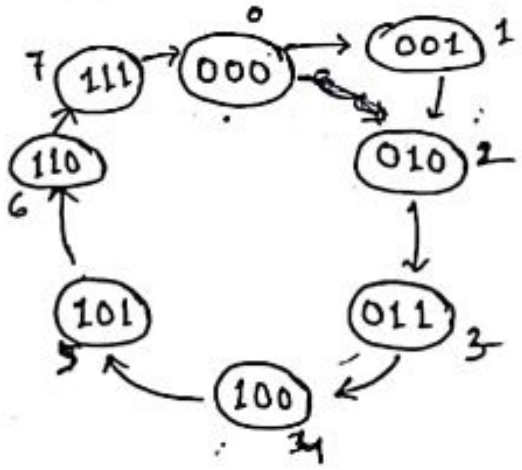
3 bit → 3 FFs (Step-I)

Step-II

0, 1, 2, ..., 7

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Choosing JK FF (Step-3)



State Diagram

(Step-4)

PS			NS			Required Excitations					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	1	1	1	x	0	x	0	1	x
1	1	1	0	0	0	x	1	x	1	x	1

Q_3 \ Q_2Q_1	00	01	11	10
0			1	
1	x	x	x	x

$J_3 = Q_2Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	x	x	x	x
1			1	

$K_3 = Q_2Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0		1	x	x
1		1	x	x

$J_2 = Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	x	x	1	
1	x	x	1	

$K_2 = Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	1	x	x	1
1	1	x	x	1

$J_1 = 1$

Q_3 \ Q_2Q_1	00	01	11	10
	x	1	1	x
	x	1	1	x

$K_1 = 1$

$J_1 = K_1 = 1$

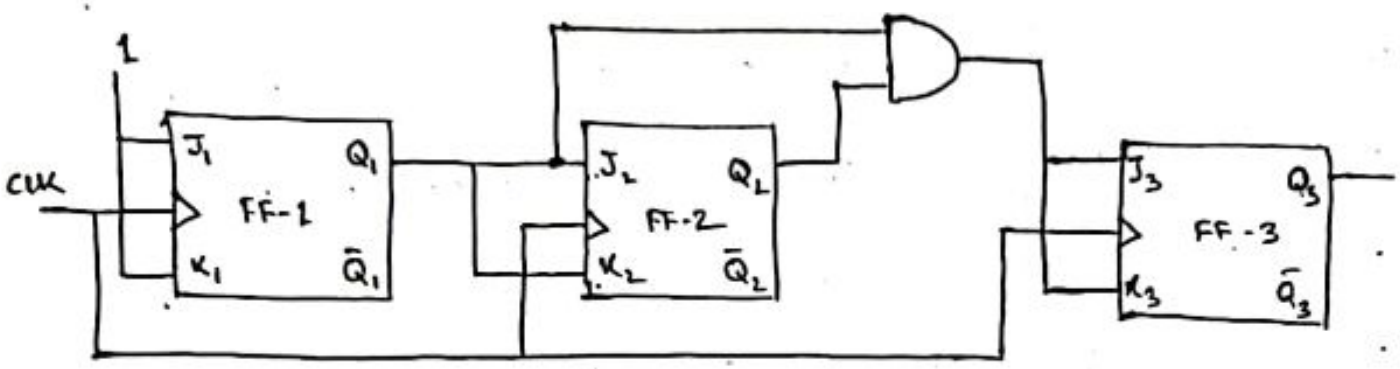
$J_2 = K_2 = Q_1$

$J_3 = K_3 = Q_2 Q_1$

FF-1 → LSB

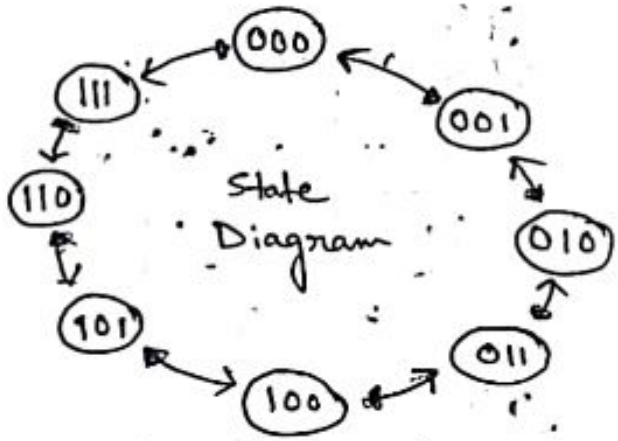
FF-3 → MSB

Step-5



- **3 bit Synchronus Down Counter**

3 bit → 3 FFs



7, 6, 5, ... 0 →

Q_n →	Q_{n+1}	J	K
0	0	0	X
1	0	X	1
0	1	1	X
1	1	X	0

Excitation Table of JK FF

PS			NS			Required Excitations Transitions					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
1	1	1	1	1	0	x	0	x	0	x	1
1	1	0	1	0	1	x	0	x	1	1	x
1	0	1	1	0	0	x	0	0	x	x	1
1	0	0	0	1	1	1	1	1	x	1	x
0	1	1	0	1	0	0	x	x	0	x	1
0	1	0	0	0	1	0	x	x	1	1	x
0	0	1	0	0	0	0	x	0	x	x	1
0	0	0	1	1	1	1	1	1	x	1	x

Q_3 \ Q_2Q_1	00	01	11	10
0	1			
1	x	x	x	x

$J_3 = \bar{Q}_2\bar{Q}_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	x	x	x	x
1	1			

$K_3 = \bar{Q}_2 \cdot \bar{Q}_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	1		x	x
1	1		x	x

$J_2 = \bar{Q}_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	x	x		1
1	x	x		1

$K_2 = \bar{Q}_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	1	x	x	1
1	1	x	x	1

$J_1 = 1$

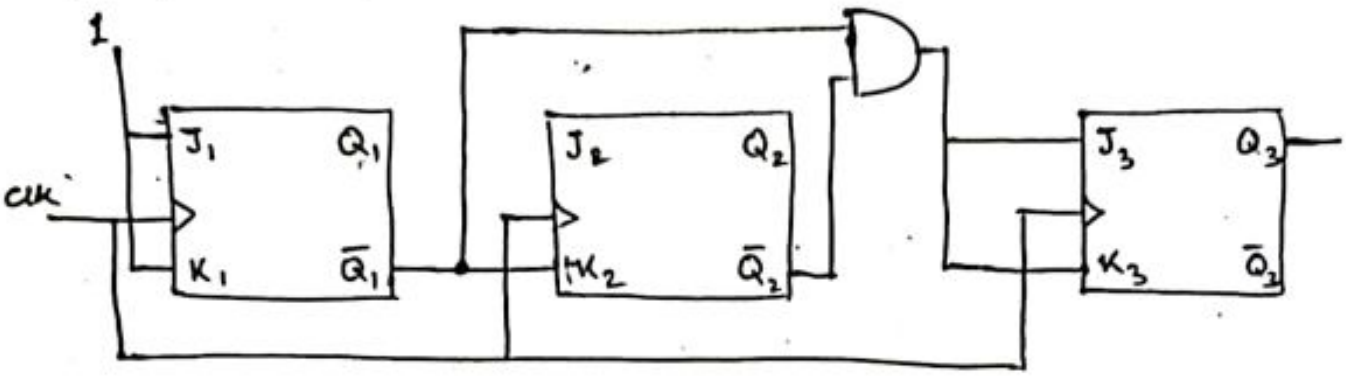
Q_3 \ Q_2Q_1	00	01	11	10
0	1	x	x	1
1	1	x	x	1

$K_1 = 1$

$J_1 = K_1 = 1$

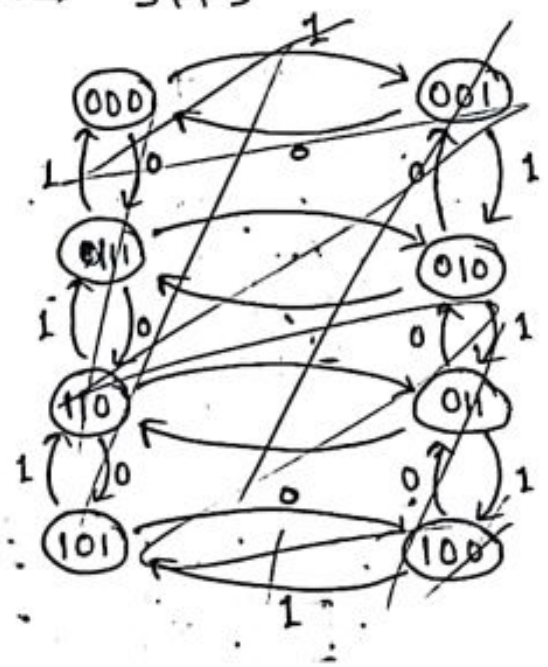
$J_2 = K_2 = \bar{Q}_1$

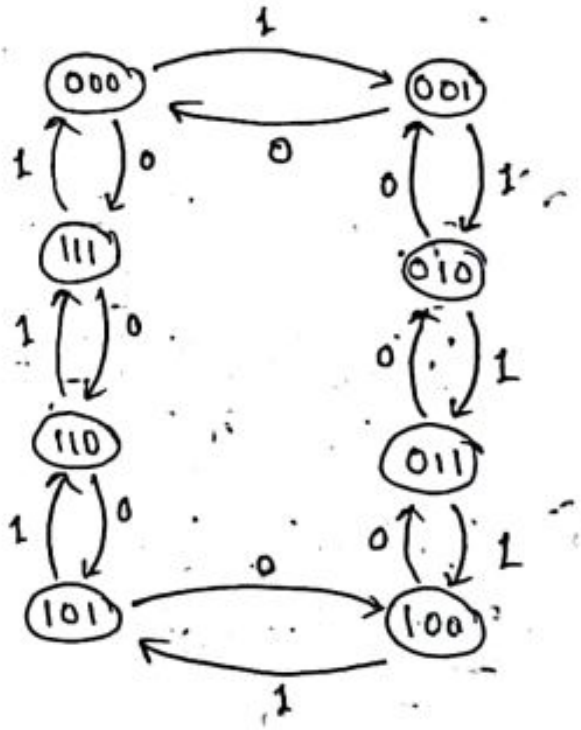
$J_3 = K_3 = \bar{Q}_2 . \bar{Q}_1$



- **Synchronus 3 bit Up/Down Counter**

3 bit → 3 FFS





PS			Mode	NS			Required excitations					
Q_3	Q_2	Q_1	M	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	1	1	1	1	X	1	X	1	X
0	0	0	1	0	0	1	0	X	0	X	1	X
0	0	1	0	0	0	0	0	X	0	X	X	1
0	0	1	1	0	1	0	0	X	1	X	X	1
0	1	0	0	0	0	1	0	X	X	1	1	X
0	1	0	1	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	0	X	X	0	X	1
0	1	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	1	1	X	1	1	X	1	X
1	0	0	1	1	0	1	X	0	0	X	1	X
1	0	1	0	1	0	0	X	0	0	X	X	1
1	0	1	1	1	1	0	X	0	1	X	X	1
1	1	0	0	1	0	1	X	0	X	1	1	X
1	1	0	1	1	1	1	X	0	X	0	1	X
1	1	1	0	1	1	0	X	0	X	0	X	1
1	1	1	1	0	0	0	X	1	X	1	X	1

Q_3Q_2 \ Q_1M	00	01	11	10
00	1			
01			1	
11	x	x	x	x
10	x	x	x	x

$$J_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$$

Q_3Q_2 \ Q_1M	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11			1	
10	1			

$$K_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$$

Q_3Q_2 \ Q_1M	00	01	11	10
00	1		1	
01	x	x	x	x
11	x	x	x	x
10	1		1	

$$J_2 = \bar{Q}_1\bar{M} + Q_1M$$

Q_3Q_2 \ Q_1M	00	01	11	10
00	x	x	x	x
01	1		1	
11	1		1	
10	x	x	x	x

$$K_2 = \bar{Q}_1\bar{M} + Q_1M$$

Q_3Q_2 \ Q_1M	00	01	11	10
00	1	1	x	x
01	1	1	x	x
11	1	1	x	x
10	1	1	x	x

$$J_1 = 1$$

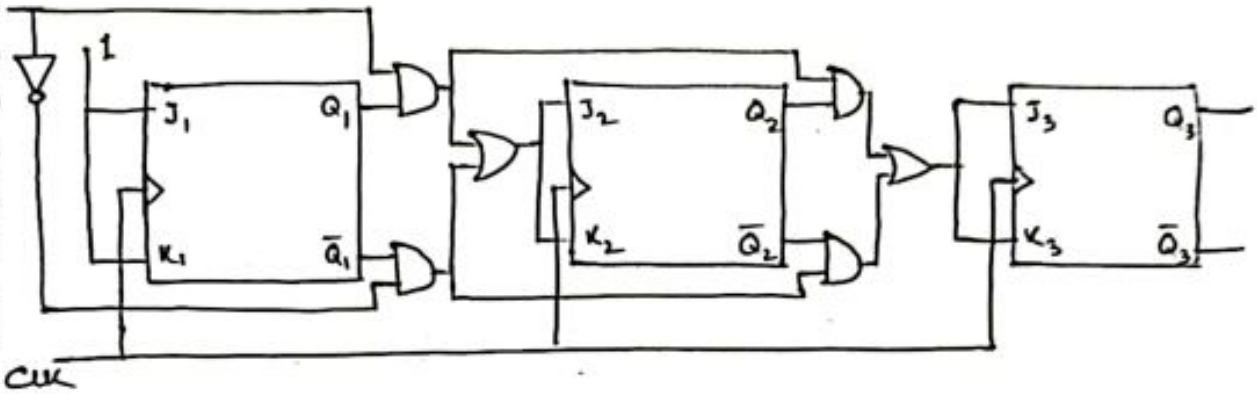
Q_3Q_2 \ Q_1M	00	01	11	10
00	x	x	1	1
01	x	x	1	1
11	x	x	1	1
10	x	x	1	1

$$K_1 = 1$$

$J_1 = K_1 = 1$

$J_2 = K_2 = \bar{Q}_1\bar{M} + Q_1M$

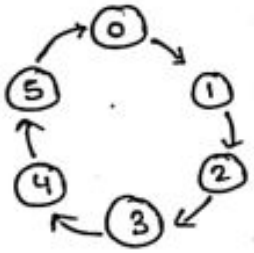
$J_3 = K_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$



Mod Counter

N = Number given

N = 6

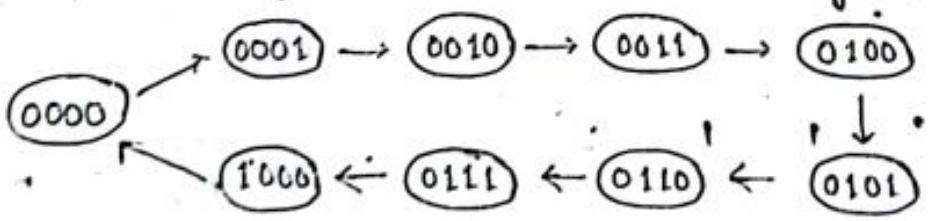


N = 12 → 0 to 11

N = 14 → 0 to 13

$N \leq 2^n$ n = No of FFs

• Modulo 9 Synchronus Counter Using T FFs



N = 9 → 0 to 8

Required FFs → 4 FF bcz of 4 bits

Upto 1000 states are valid and the remaining (1001, 1010, ..., 1111) 7 states out of 16 states are invalid.

S_n	S_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table of T FF

PS				NS				Required Excitations			
Q_4	Q_3	Q_2	Q_1	Q_4	Q_3	Q_2	Q_1	T_4	T_3	T_2	T_1
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0	0	0

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01			1	
11	X	X	X	X
10	1	X	X	X

$T_4 = Q_4 + Q_3 Q_2 Q_1$

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00			1	
01			1	
11	X	X	X	X
10		X	X	X

$T_3 = Q_2 Q_1$

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00		1	1	
01		1	1	
11	X	X	X	X
10		X	X	X

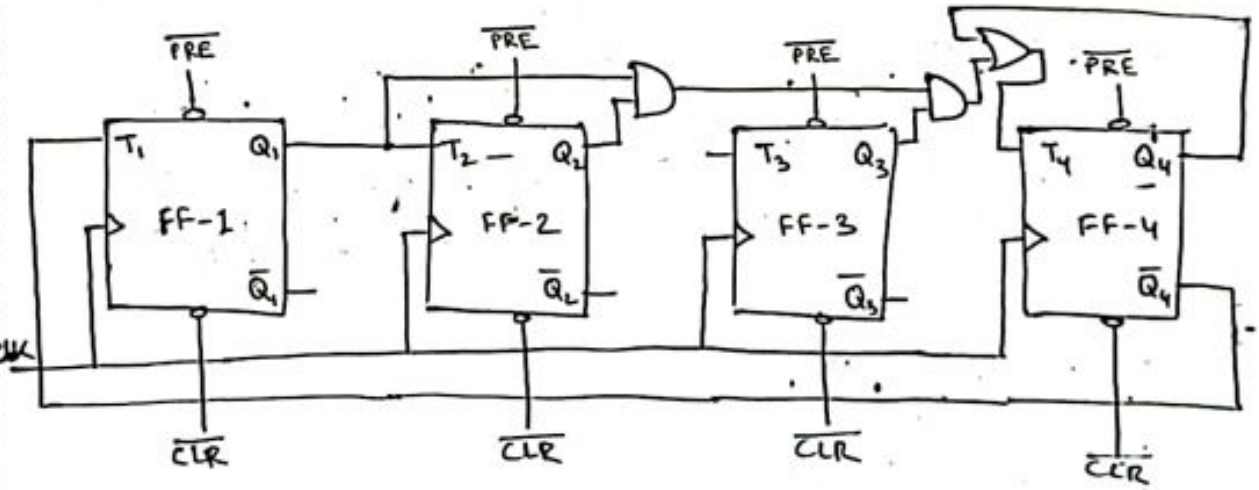
$T_2 = Q_1$

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10		X	X	X

$T_1 = \overline{Q_4}$

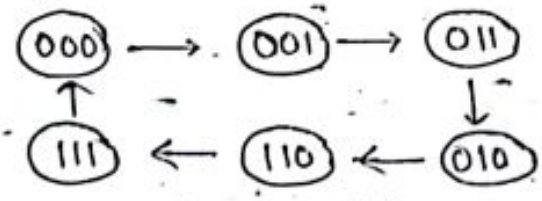
$T_1 = \overline{Q_4}$ $\quad$ $T_3 = Q_2Q_1$

$T_2 = Q_1$ $\quad$ $T_4 = Q_4 + Q_3Q_2Q_1$



- Synchronus Modulo 6 Gray Code Counter (0, 1, 3, 2, 6, 7

Required FFs → 3 → bcz of 3 bits → Total 8 states



The remaining two states (101, 100) are invalid

6 are valid states

PS			NS			Required Excitations		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	T_3	T_2	T_1
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	0	1
0	1	0	1	1	0	1	0	0
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Q_3 \ Q_2Q_1	00	01	11	10
0				1
1	X	X	1	

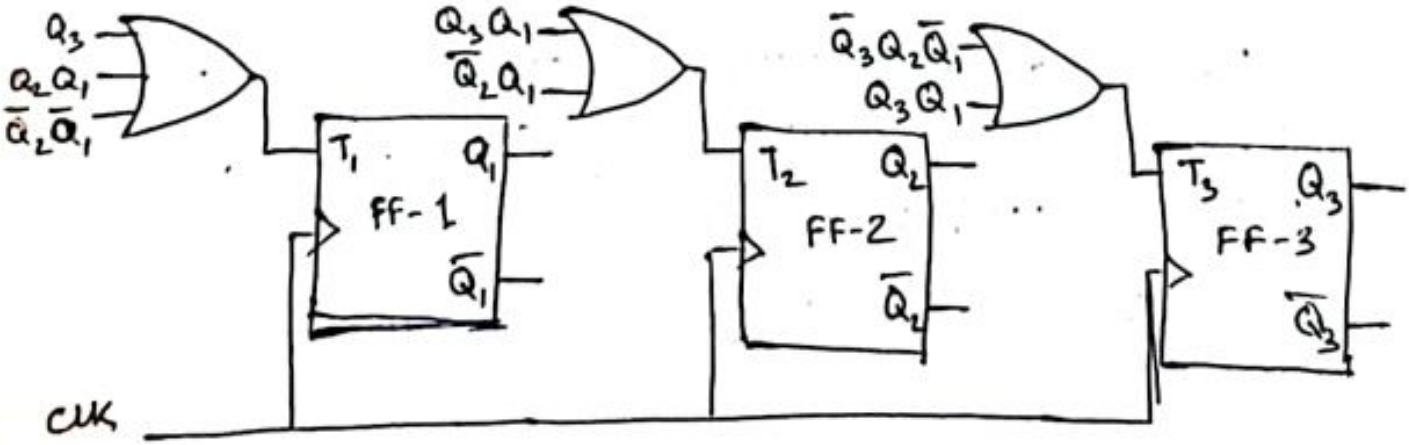
$T_3 = Q_3Q_1 + \bar{Q}_3Q_2\bar{Q}_1$

Q_3 \ Q_2Q_1	00	01	11	10
0		1		
1	X	X	1	

$T_2 = Q_3Q_1 + \bar{Q}_2Q_1$

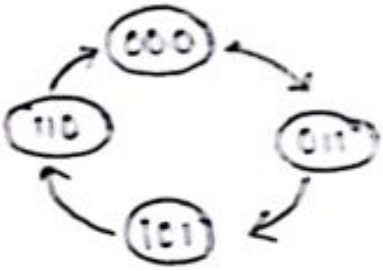
Q_3 \ Q_2Q_1	00	01	11	10
0	1		1	
1	X	X	1	1

$T_1 = Q_3 + Q_2Q_1 + \bar{Q}_2\bar{Q}_1$



- Design a type T counter that goes through states 0, 3, 5, 6, 0, . . .

Ans Required FFs → 3



PS			NS			Required Excitations		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	T_3	T_2	T_1
0	0	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	0
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0

Q_3 \ Q_2Q_1	00	01	11	10
0		x	1	x
1	x		x	1

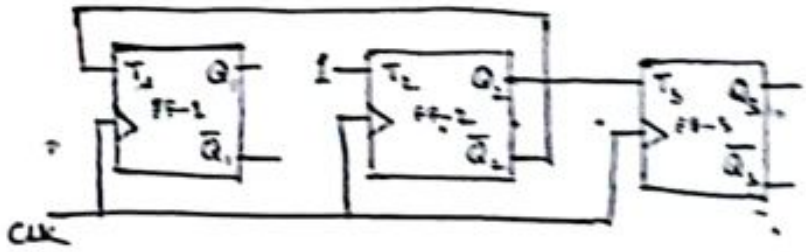
$T_3 = Q_2$

Q_3 \ Q_2Q_1	00	01	11	10
0	1	x	1	x
1	x	1	x	1

$T_2 = 1$

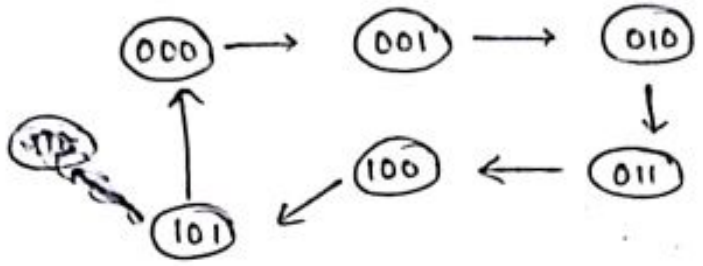
Q_3 \ Q_2Q_1	00	01	11	10
0	1	x		x
1	x	1	x	

$T_1 = \overline{Q_2}$



- Synchronous Mod 6 Counter Using JK FF

Required FFs → 3



PS			NS			Required Excitations					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

Q_3 \ Q_2Q_1	00	01	11	10
0			1	
1	X	X	X	X

$J_3 = Q_2Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	X	X
1		1	X	X

$K_3 = Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0		1	X	
1			X	X

$J_2 = \bar{Q}_3Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	1	
1	X	X	X	X

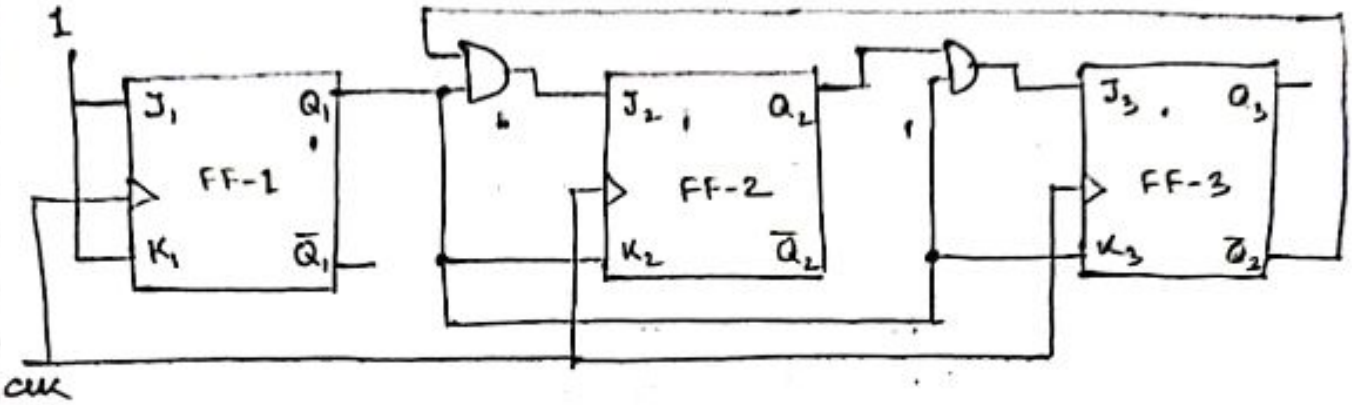
$K_2 = Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	1	X	X	1
1	1	X	X	X

$J_1 = 1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	1	1	X
1	X	1	X	X

$K_1 = 1$

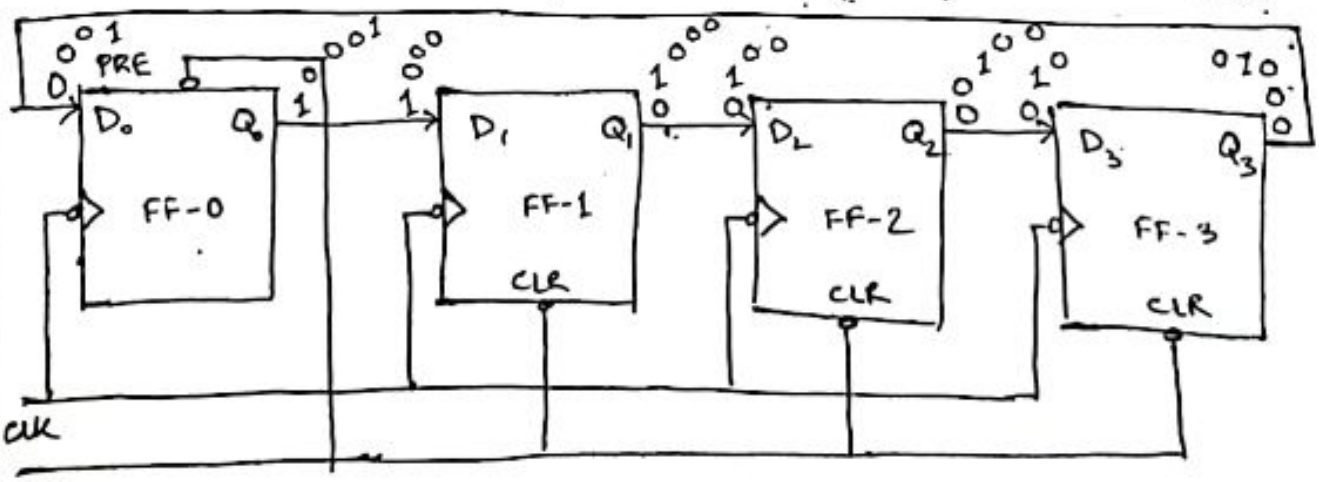


Shift Register Counter

a) Ring Counter / Basic Ring Counter / Simple Ring Counter

- Ring counter is a typical application of Shift Register.
- The only change is the output of last ff is connected to the input of first ff.
- In case of ring counter,

No of states = No of FFs used.



ORI
↳ Over Riding Input

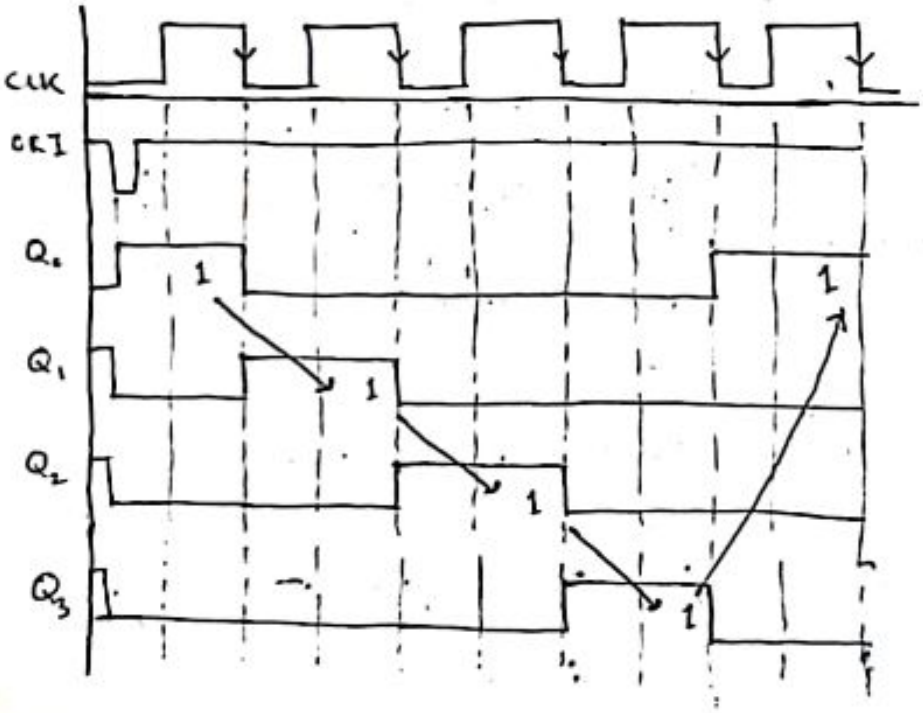
PRE = 0, Q = 1
CLR = 0, Q = 0

ORI	CLK	Q_0	Q_1	Q_2	Q_3
low	X	1	0	0	0
High 1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

When ORI is low,

PRE = 0 , Q = 1

CLR Q = 0 , Q = 0



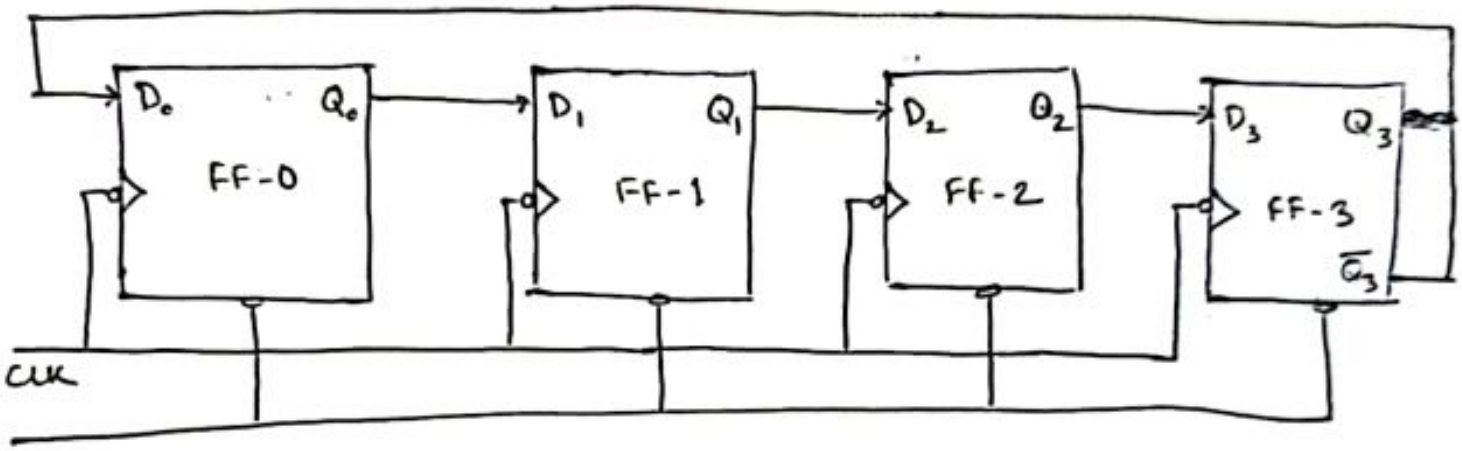
Forming a Ring

b) Twisted / Johnson / Switch Tail Ring Counter

- Johnson's Counter is much much better than the Ring Counter because,

1) No of states = 2 x No of FFs

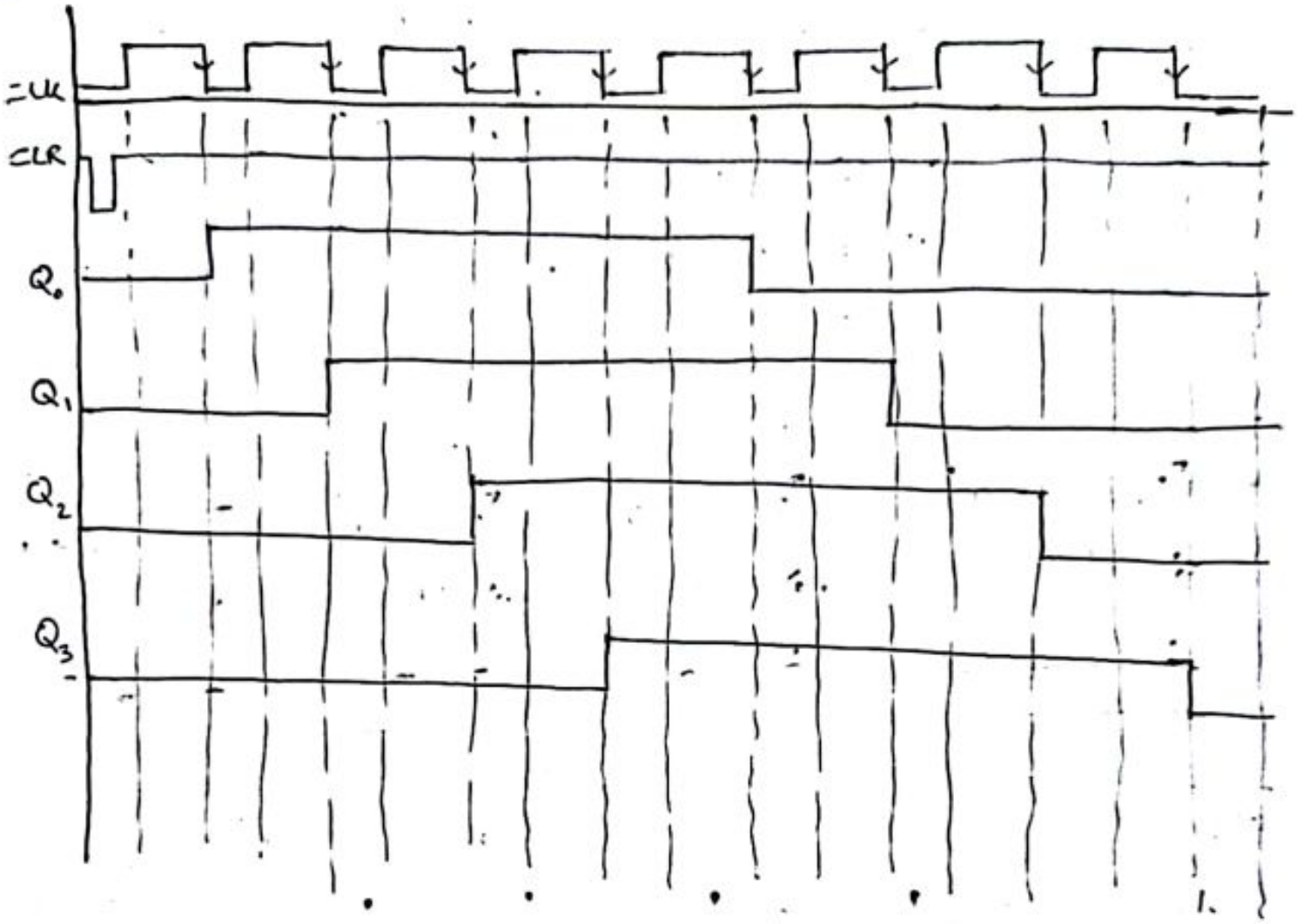
(ex: For 4 FFs we'll get 8 states)



CLR = 0 , Q = 0

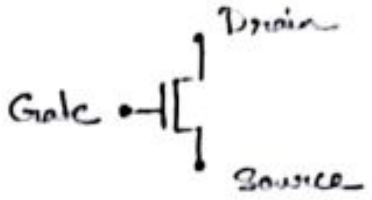
⊔ → low
1 → High

CLR	CLK	Q_0	Q_1	Q_2	Q_3
⊔	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1
1	↓	0	0	0	1
1	↓	0	0	0	0


CLK
CLR
Q0
Q1
Q2
Q3

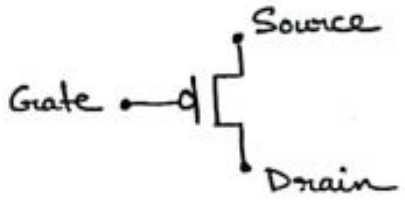
CMOS

NMOS :



N-channel MOSFET contains N type source and Drain diffused on an P type substrate. In NMOS the majority of carriers are electrons. When a high voltage is applied to the gate the NMOS will conduct. And for low voltage, NMOS will not conduct. NMOS is considered to be faster than PMOS.

PMOS :



P-channel MOSFET consists of P type source and Drai. diffused on an N type substrate. The majority of carriers are holes. When a high voltage is applie. to the gate, the PMOS will not conduct. When a low voltage is applied the PMOS will conduct.

CMOS (Complementary Metal Oxide Semiconductor)

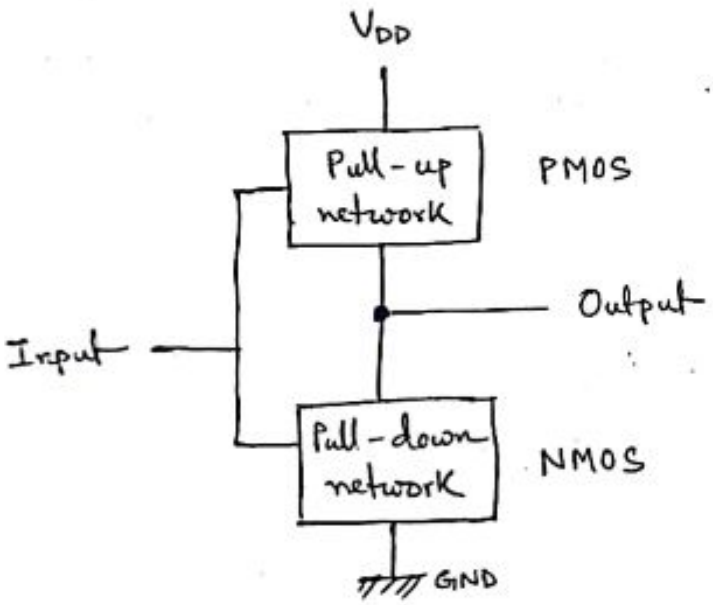
In CMOS technology both P and N type transistors are used to design logic functions.

Advantages of CMOS :

i) High speed ii) Low power dissipation

iii) High noise margins in both states

iv) Can be operated over a wide range of source and input voltages.



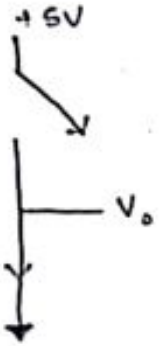
In CMOS, a collection of N-type mosfets are arrange in a pull down network b/w the output and the low voltage power supply rail (Vss or often ground).

A collection of P type mosfets are arranged in a pull up network b/w the output and the higher output voltage power supply rail (V_{dd}).

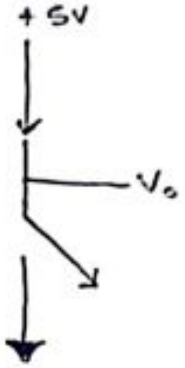
When PMOS is ON, NMOS is OFF and Vice Versa.

When a high voltage is given at input terminal of the inverter, the PMOS becomes an open circuit and NMOS switched OFF, so the output will be pulled down to V_{SS}.

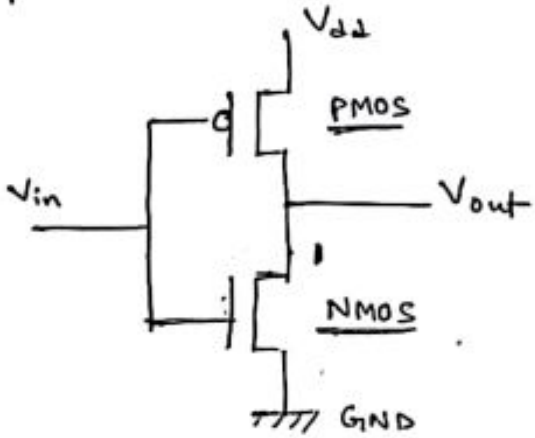
When a low voltage is given at input terminal of the inverter, the NMOS switched OFF and PMOS switched ON. So the output becomes V_{dd} or the circuit is pulled up to V_{dd}.



For high voltage in V_{in}



For low voltage in V_{in}



CMOS Inverter

When,

$$\left.\begin{array}{l} V_{in} = 0 \longrightarrow V_{out} = 1 \\ V_{in} = 1 \longrightarrow V_{out} = 0 \end{array}\right\} \text{Working as an inverter}$$

Design different logic gates using CMOS :

(i) For multiplication (AND) form, the NMOS will be in Series.

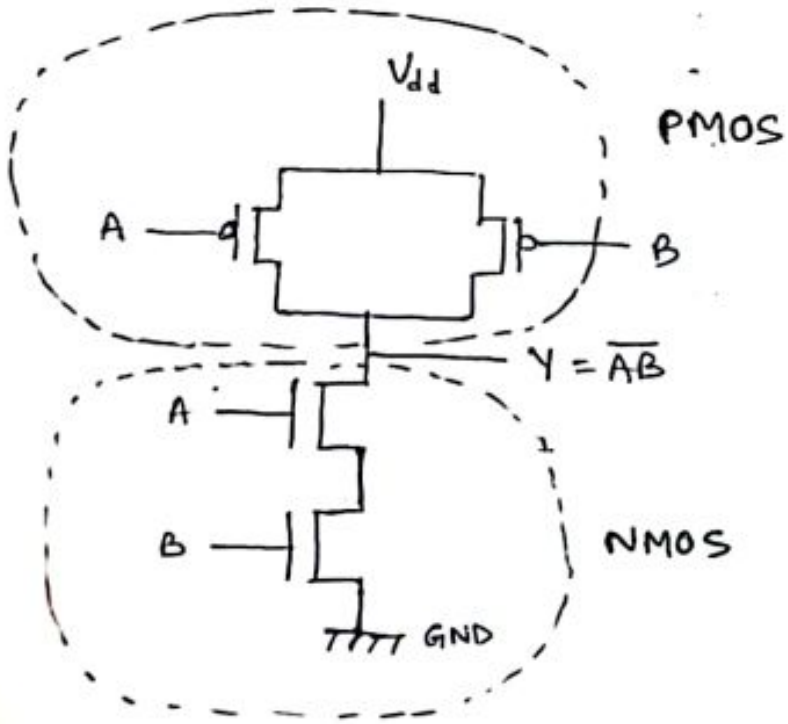
(ii) For OR (Addition) form, the NMOS will be in Parallel.

(iii) Bubble exists in PMOS only, not in NMOS.

(iv) From any boolean exp. at first draw the NMOS part then complement it for PMOS. For ex: If two inputs A and B are in Series for NMOS, then for PMOS the inputs will be in Parallel.

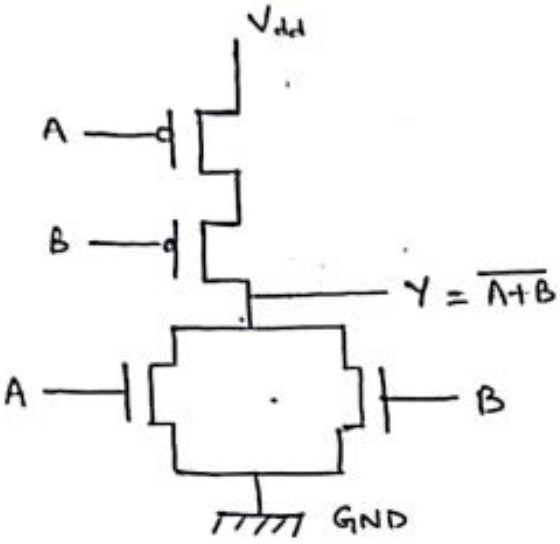
NAND gate :

$$Y = \overline{AB}$$

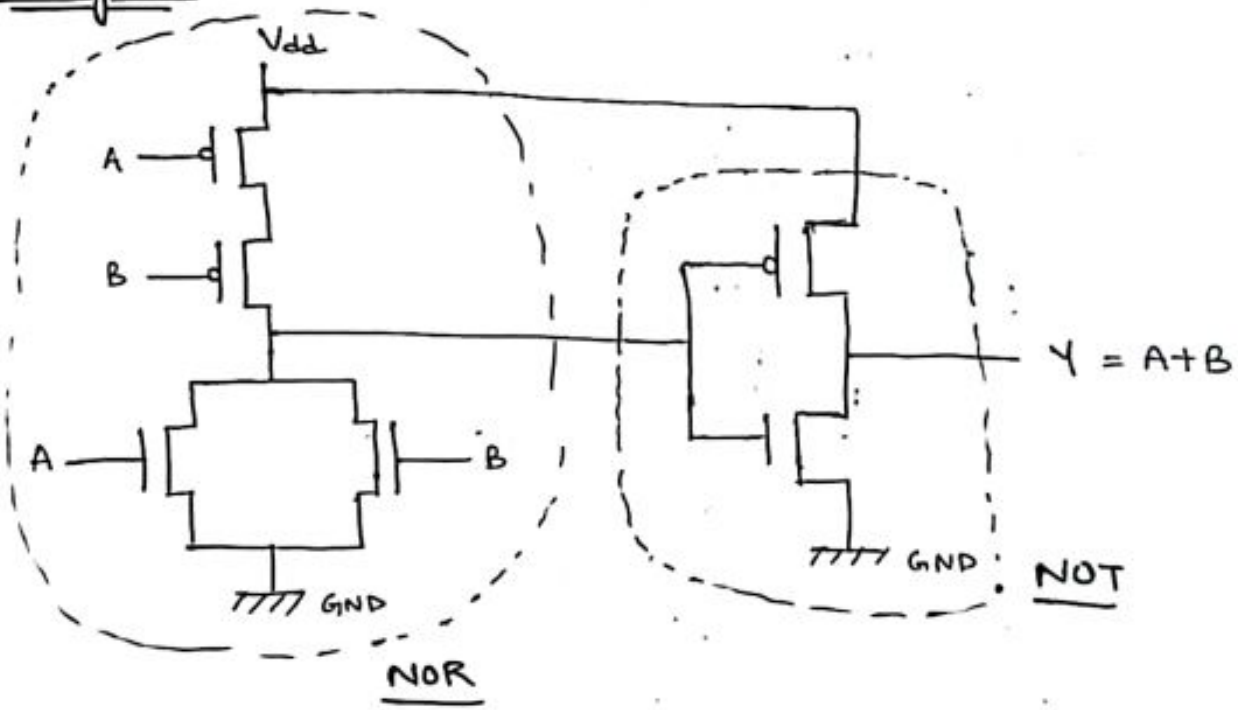


NOR gate :

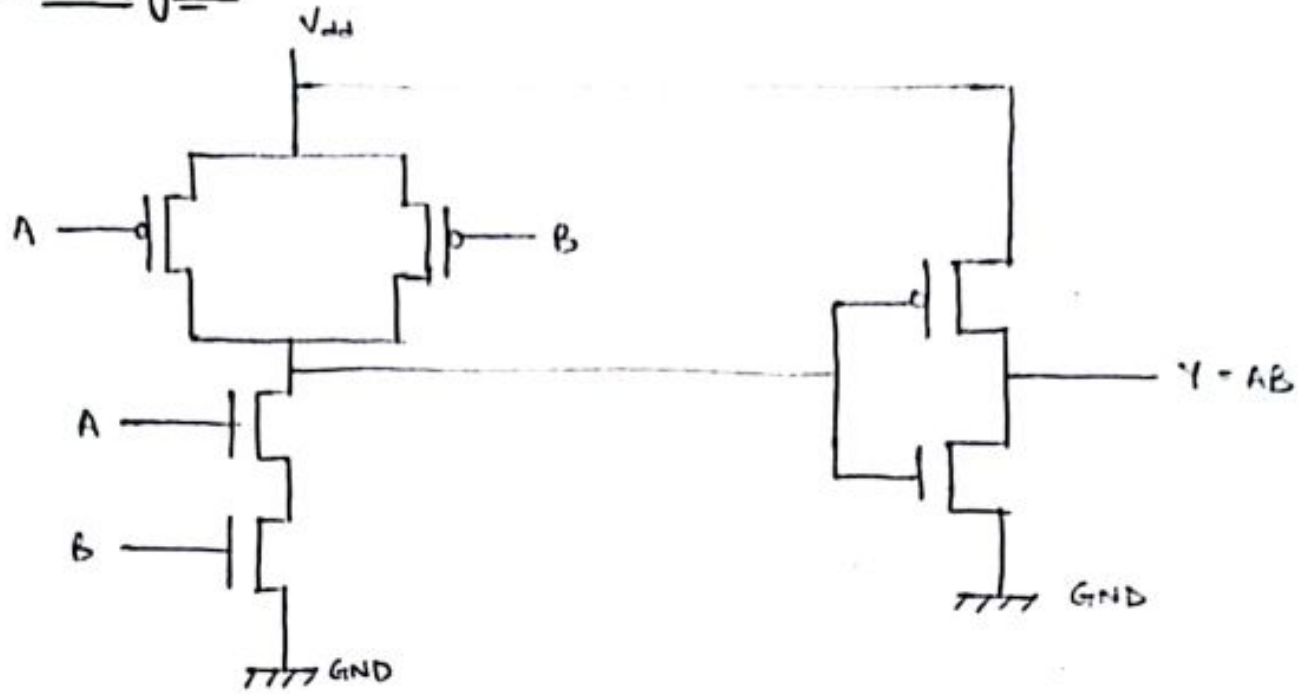
$$Y = \overline{A+B}$$



OR gate :

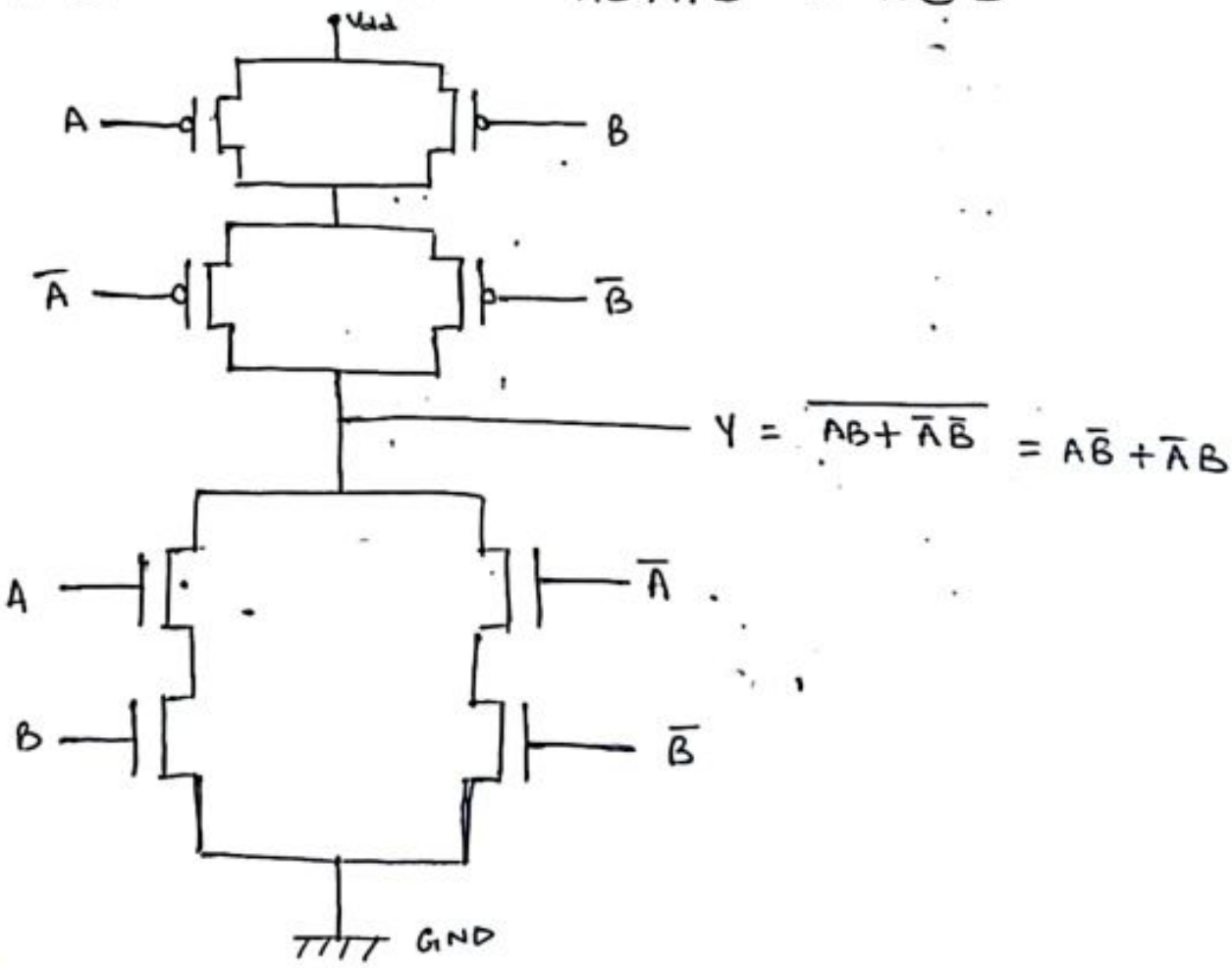


AND gate :



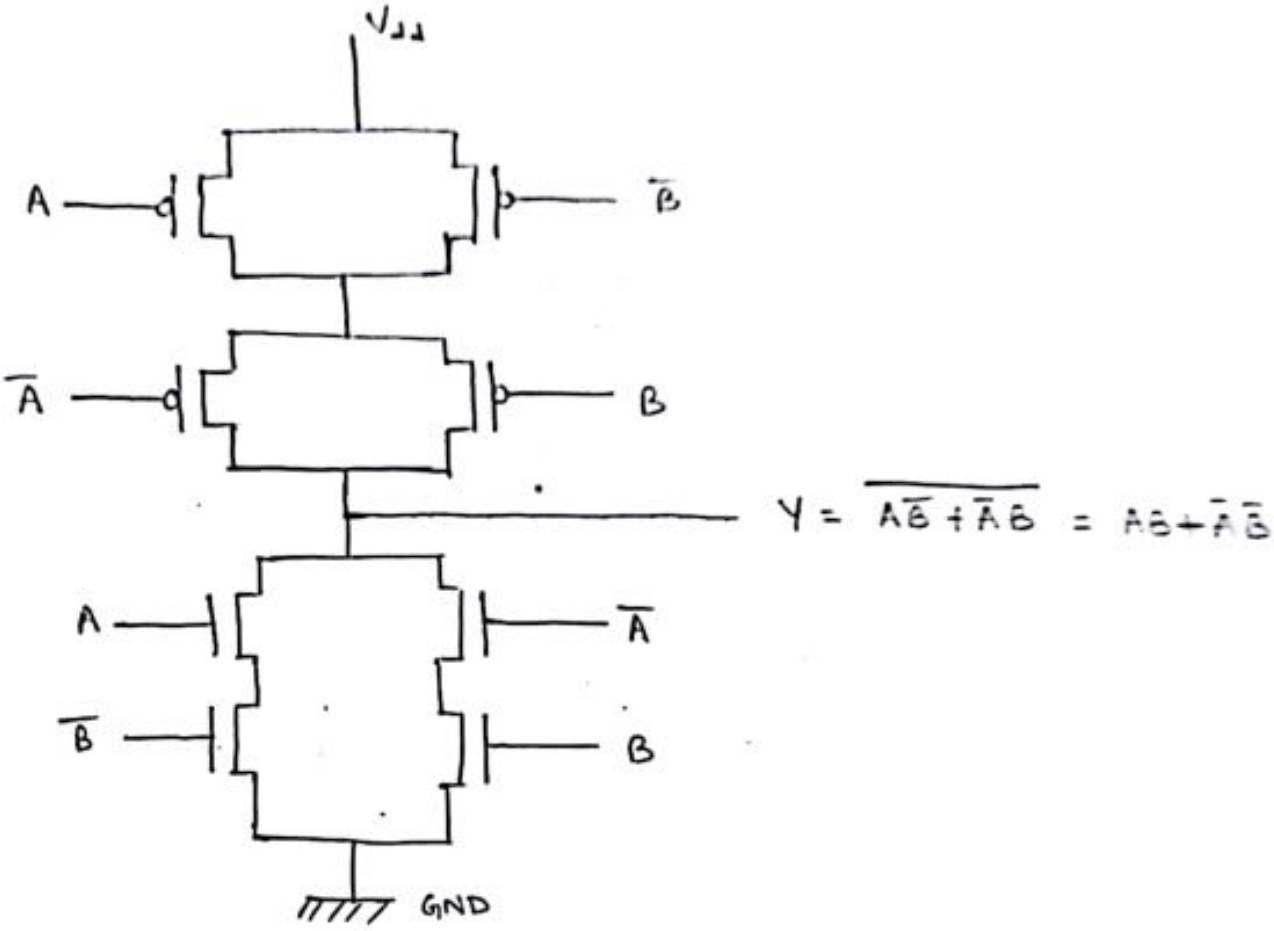
XOR gate :

$$Y = A \oplus B = A\bar{B} + \bar{A}B = \overline{AB + \bar{A}\bar{B}} = \overline{A \odot B}$$



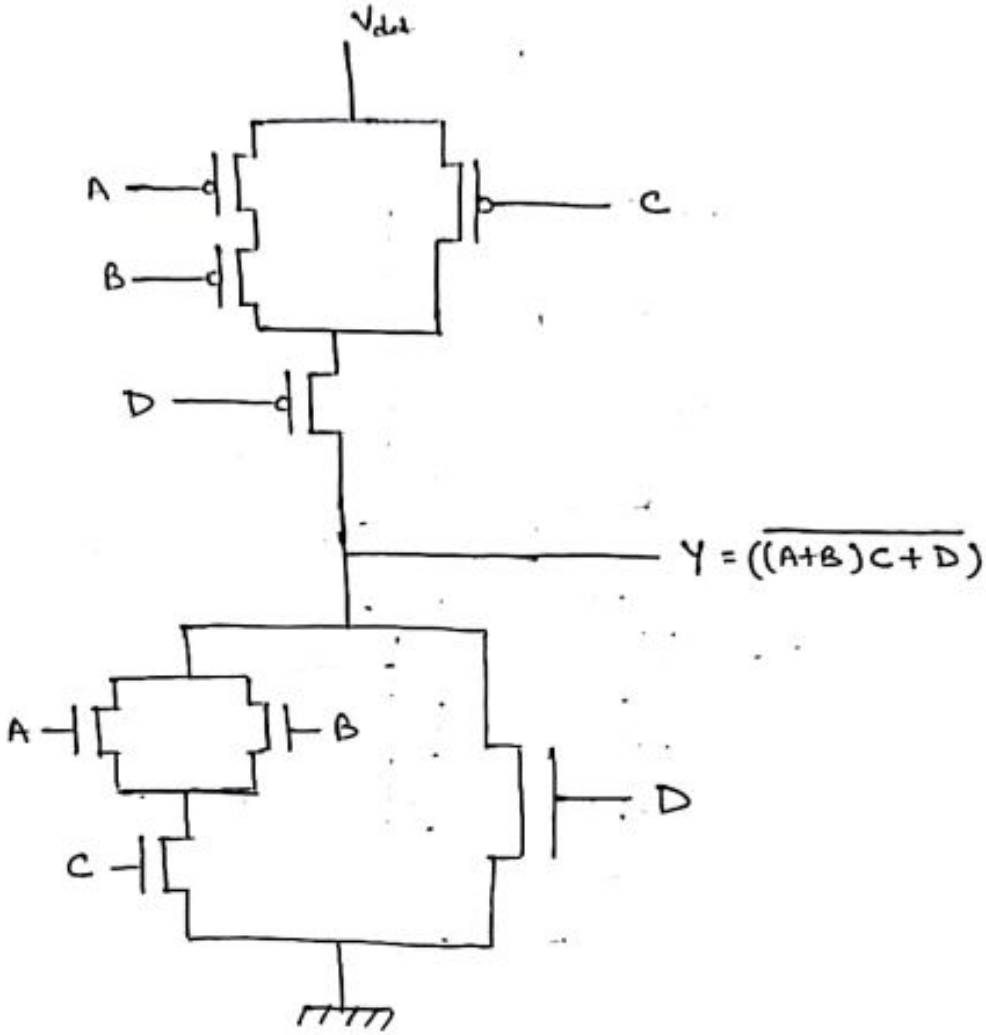
XNOR gate :

$$Y = \overline{A\bar{B} + \bar{A}B} = \overline{A \oplus B} = A \odot B = AB + \bar{A}\bar{B}$$

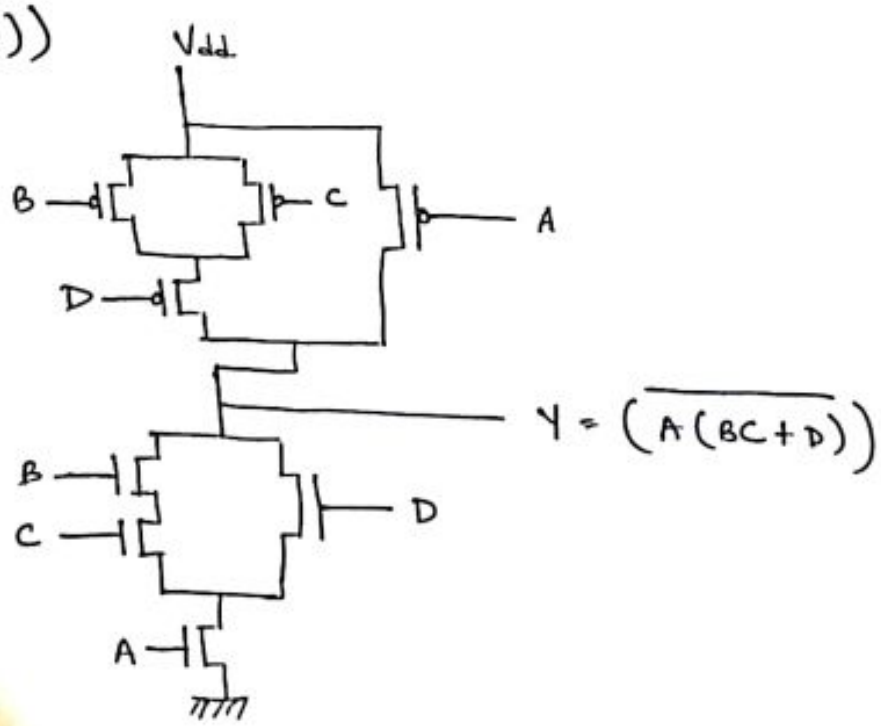


Design of General Boolean Circuits Using CMOS :

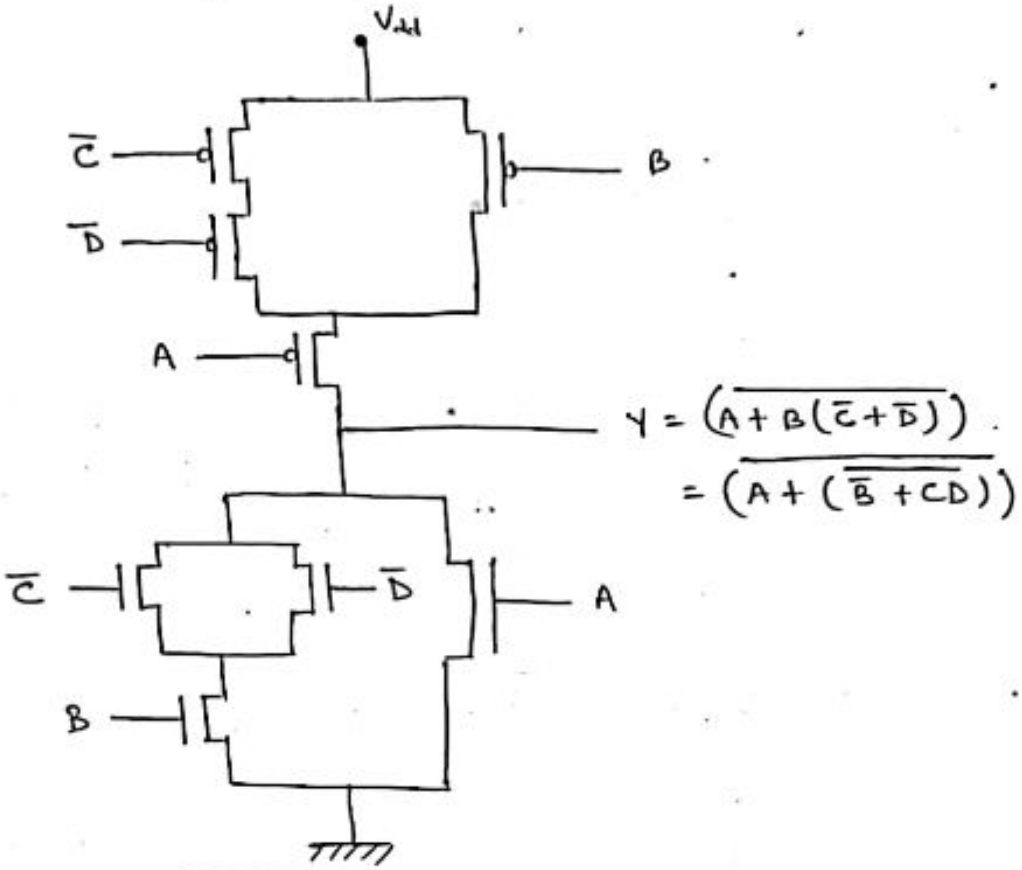
① $\overline{((A+B)C+D)}$



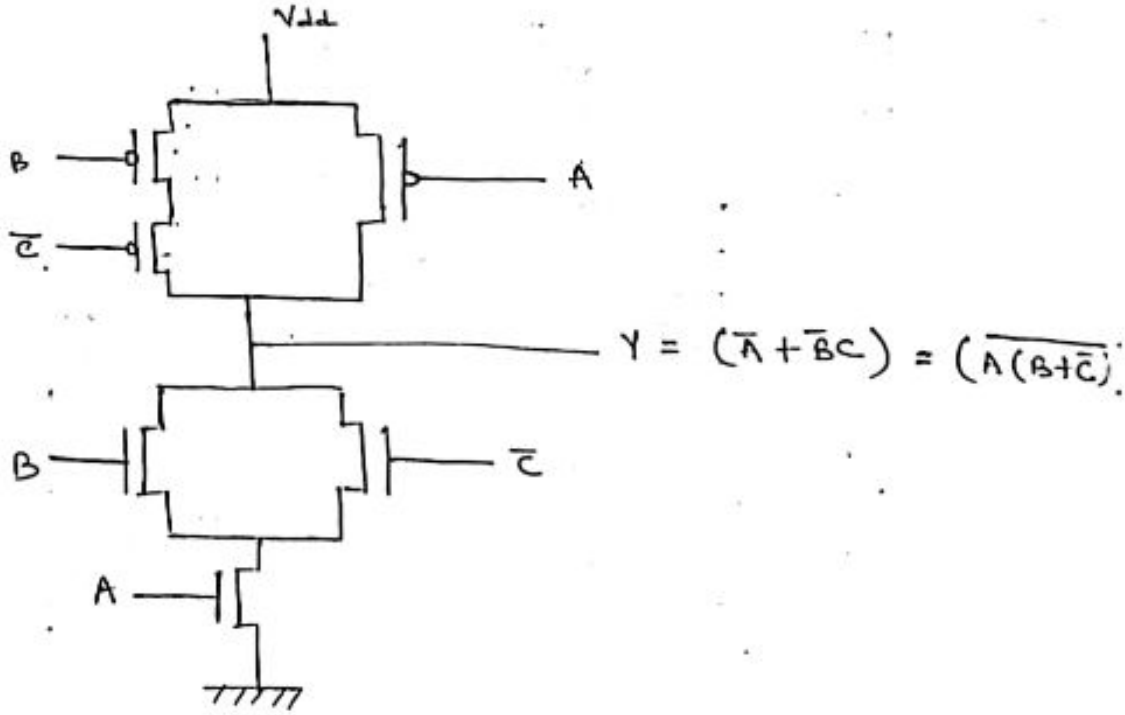
② $\overline{(A(BC+D))}$



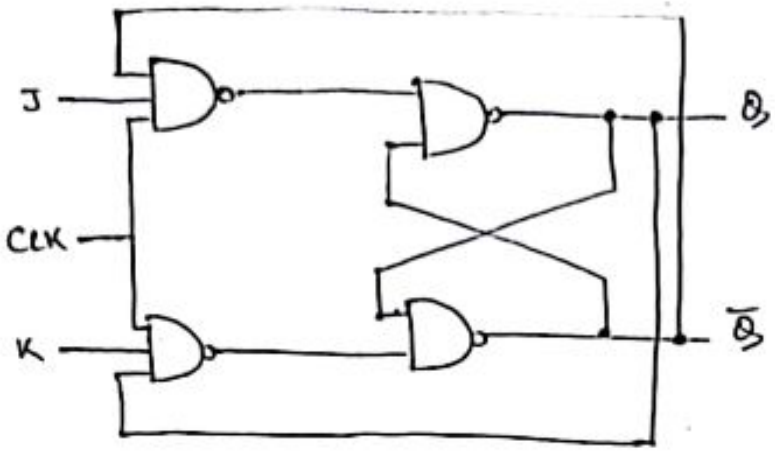
③ $\overline{(A+(\overline{B+CD}))} = (\overline{A+B(\overline{CD})}) = (\overline{A+B(\overline{C}+\overline{D})})$



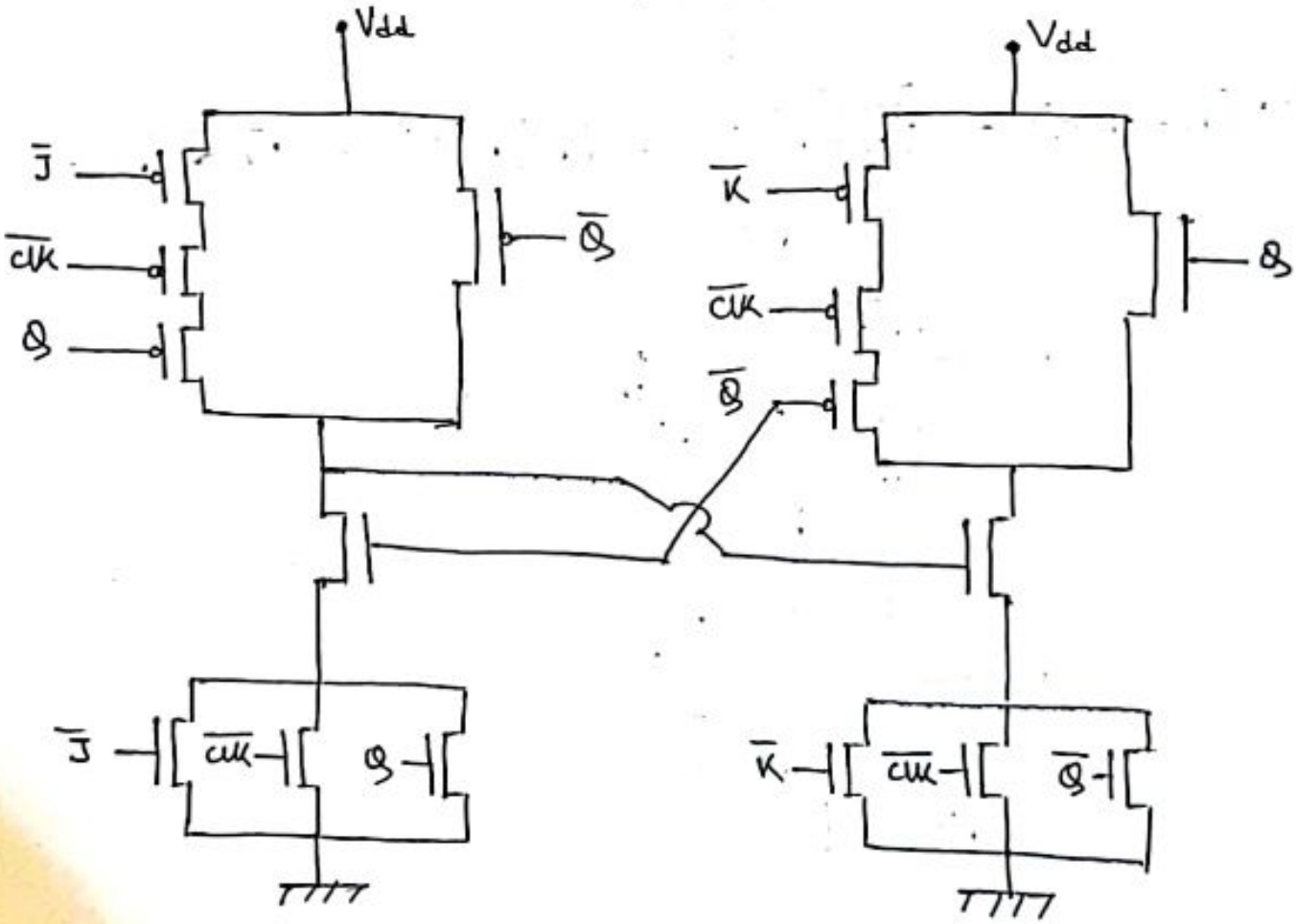
④ $(\overline{A}+\overline{B}C) = \overline{\overline{(\overline{A}+\overline{B}C)}} = (\overline{A(\overline{\overline{B}C})}) = (\overline{A.(B+\overline{C})})$



JK FF Using CMOS Logic



CLK	J	K	Q_{n+1}	$\overline{Q}_{n+1}$
0	X	X	Q_n	$\overline{Q}_n$
1	0	0	Q_n	$\overline{Q}_n$
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid	

$$Q = \overline{\overline{J.CLK.\overline{Q}}.\overline{Q}} = \overline{(\overline{J}+\overline{CLK}+Q).\overline{Q}}$$

$$Q = \overline{\overline{K.CLK.Q}.Q} = \overline{(\overline{K}+CLK+\overline{Q}).Q}$$