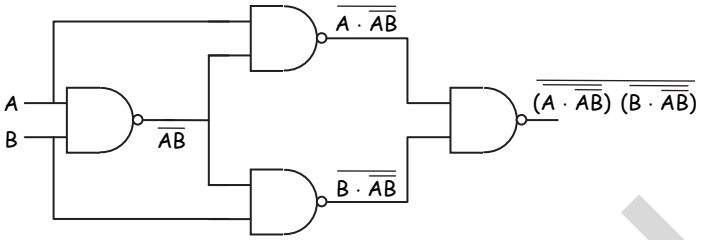
**Q.1(a) Implement Ex-OR gate using Four NAND gates only.** **[5]**

**(A)**

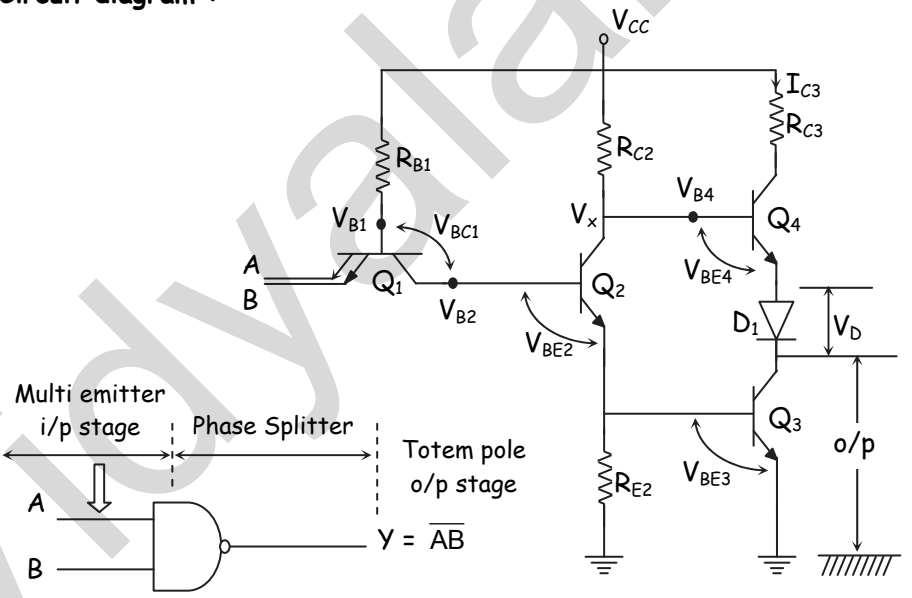



$$\overline{(A\cdot\overline{AB})}\,\overline{(B\cdot\overline{AB})} = \overline{\overline{A\cdot\overline{AB}}}+\overline{\overline{B\cdot\overline{AB}}} = A(\overline{A}+\overline{B})+B(\overline{A}+\overline{B})$$

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

**Q.1(b) Explain TTL NAND gate.** **[5]**

**(A) Circuit diagram :**




| A | B | $Y = \overline{AB}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Operation :**

**Case I :** $A = 0, B = 0$

For this case $Q_1$ works in normal mode and transistors $Q_2$ and $Q_3$ both will be OFF and voltage at $V_x$ point will be $V_{CC}$ and this is sufficient to turn ON $Q_4$ and $D_1$ (because to turn ON $Q_4$ and $D_1$ voltage $V_x = V_{B4} = V_{BE4} + V_{D1} = 0.7 + 0.7 = 1.4$ V) so $Q_4$ and $D_1$ both will be ON when output $V_0$.

$$V_0 = V_x - V_{BE4} - V_{D1}$$
$$= V_{CC} - V_{BE4} - V_{D1}$$
$$= 5 - 0.7 - 0.7$$
$$= 3.6 \text{ V}$$
$$= \text{logic 1}$$
$$V_0 = \text{logic 1}$$

**Case II** : $A = 0, B = 1$
**Case III** : $A = 1, B = 0$
**Case IV** : $A = 1, B = 1$
} same as case (I)

For this case, transistor $Q_1$ works in inverse mode due to this $Q_2$ and $Q_3$ both will be ON and voltage at $V_x = V_{CE2} + V_{CE2\,(sat)} + V_{BE3}$
$= 0.2 + 0.7$
$= 0.9$ V

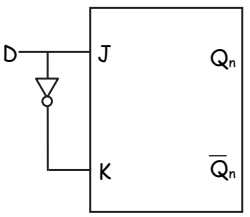
This voltage is not sufficient to turn ON $Q_4$ and $D_1$ because it is less than 1.4 V. So $Q_4$ and $D_1$ both will be OFF and output $V_0 = V_{CE3\,(sat)} = 0.2$ V = logic 0.

**Q.1(c) Convert JK Flip-Flop into D Flip-Flop and T Flip-Flop.** **[5]**

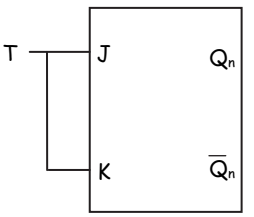
**(A)** Excitation table

| $Q_n$ | $Q_{n+1}$ | J | K | D | T |
|---|---|---|---|---|---|
| 0 | 0 | 0 | × | 0 | 0 |
| 0 | 1 | 1 | × | 1 | 1 |
| 1 | 0 | × | 1 | 0 | 1 |
| 1 | 1 | × | 0 | 1 | 0 |

1) JK to D flip-flop
$J = D, K = \overline{D}$

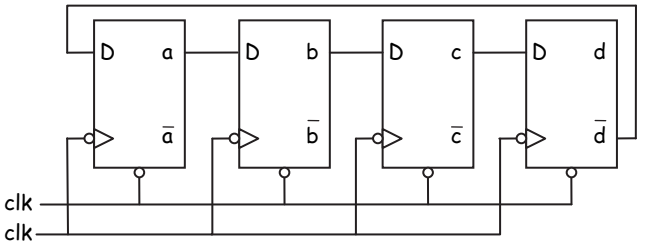



2) JK to T flip-flop
$J = T, K = T$



**Q.1(d) Explain twisted ring counter.** **[5]**

**(A)**




| a | b | c | d |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 |

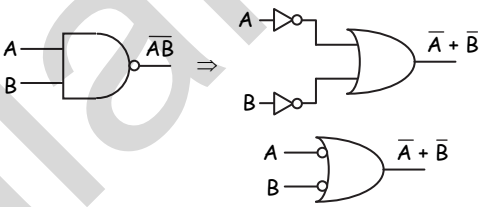
**Q.2(a) State and prove De-Morgan theorem.** **[10]**

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

Compliment of product equal to sum of their compliments.

Proof :

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

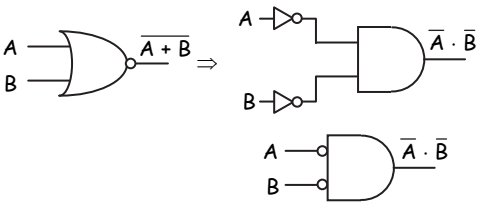



**Theorem :** $\overline{(A + B)} = \bar{A} \cdot \bar{B}$

Compliment of sum equal to product of their compliments.

Proof :

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



**Q.2(b) Reduce the following function using Quine Mc-Clusky method : [10]**

**$f(A,B,C,D) = \sum m(2,3,5,7,10,13,15)$**

**(A)** Step 1 :

| Decimal | A | B | C | d |
|---|---|---|---|---|
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 13 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 |

Step 2 :

| Group | Decimal | A | B | C | D |
|---|---|---|---|---|---|
| 1 | 2 | 0 | 0 | 1 | 0 |
| 2 | 3 | 0 | 0 | 1 | 1 |
| | 5 | 0 | 1 | 0 | 1 |
| | 10 | 1 | 0 | 1 | 0 |
| 3 | 7 | 0 | 1 | 1 | 1 |
| | 13 | 1 | 1 | 0 | 1 |
| 4 | 15 | 1 | 1 | 1 | 1 |

Step 3 :

| Group | Decimal | A | b | C | D | |
|---|---|---|---|---|---|---|
| 1 | 2,3 | 0 | 0 | 1 | X | → $P_A$ |
| | 2,10 | X | 0 | 1 | 0 | → $P_B$ |
| 2 | 3,7 | 0 | X | 1 | 1 | → $P_C$ |
| | 5,7 | 0 | 1 | X | 1 | |
| | 5,13 | X | 1 | 0 | 1 | |
| 3 | 7,15 | X | 1 | 1 | 1 | |
| | 13,15 | 1 | 1 | X | 1 | |

| Group | Decimal | a | B | C | D | |
|---|---|---|---|---|---|---|
| 1 | 5,7,13,15 | X | 1 | X | 1 | } $P_D$ |
| | 5,13,7,15 | X | 1 | X | 1 | |

**Prime Implication Table**

| | $m_2$ | $m_3$ | $m_5$ | $m_7$ | $m_{10}$ | $m_{13}$ | $m_{15}$ | Necessary |
|---|---|---|---|---|---|---|---|---|
| $P_A$ | √ | √ | | | | | | $P_A$ |
| $P_B$ | √ | | | | √ | | | $P_B$ |
| $P_C$ | | √ | | √ | | | | |
| $P_D$ | | | √ | √ | | √ | √ | $P_D$ |

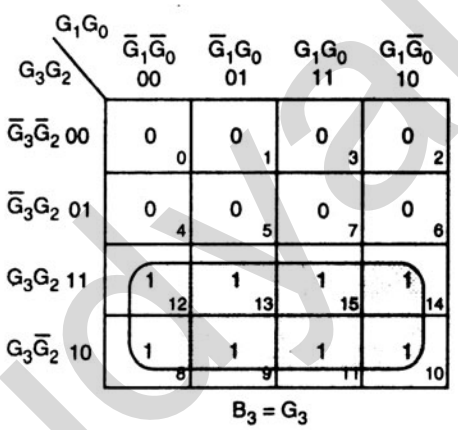
**Q.3(a) Design gray to binary code converter.** **[10]**

**(A)** Now in this case we have to do reverse i.e. convert gray codes to binary. Procedure is same as that we followed in above example.

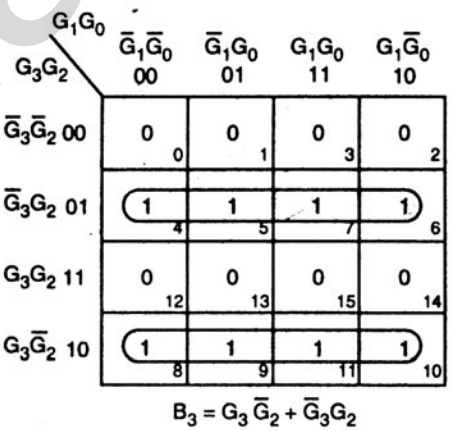
1) Truth table :

| Decimal | $G_3$ | $G_2$ | $G_1$ | $G_0$ | $B_3$ | $B_2$ | $B_1$ | $B_0$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

2) K–map :




$B_3 = G_3$




$B_3 = G_3\bar{G}_2 + \bar{G}_3G_2$

$= G_2 \oplus G_3$

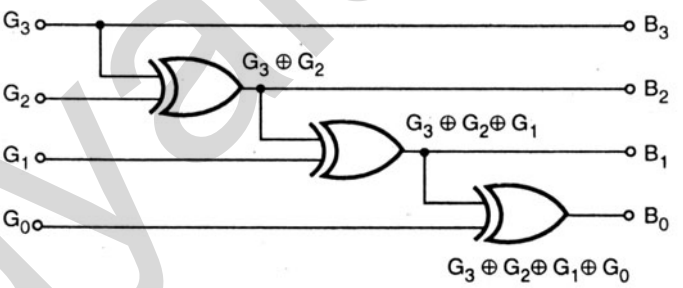
| $G_3G_2$ \ $G_1G_0$ | $\bar{G}_1\bar{G}_0$ 00 | $\bar{G}_1G_0$ 01 | $G_1G_0$ 11 | $G_1\bar{G}_0$ 10 |
|---|---|---|---|---|
| $\bar{G}_3\bar{G}_2$ 00 | 0 (0) | 0 (1) | 1 (3) | 1 (2) |
| $\bar{G}_3G_2$ 01 | 1 (4) | 1 (5) | 0 (7) | 0 (6) |
| $G_3G_2$ 11 | 0 (12) | 0 (13) | 1 (15) | 1 (14) |
| $G_3\bar{G}_2$ 10 | 1 (8) | 1 (9) | 0 (11) | 0 (10) |

$$B_1 = \bar{G}_1(\bar{G}_3 G_2 + G_3\bar{G}_2) + G_1(\bar{G}_3\bar{G}_2 + G_3 G_2)$$
$$= \bar{G}_1(G_2 \oplus G_3) + G_1\overline{(G_2 \oplus G_3)}$$
$$= G_1 \oplus G_2 \oplus G_3$$

| $G_3G_2$ \ $G_1G_0$ | $\bar{G}_1\bar{G}_0$ 00 | $\bar{G}_1G_0$ 01 | $G_1G_0$ 11 | $G_1\bar{G}_0$ 10 |
|---|---|---|---|---|
| $\bar{G}_3\bar{G}_2$ 00 | 0 (0) | 1 (1) | 0 (3) | 1 (2) |
| $\bar{G}_3G_2$ 01 | 1 (4) | 0 (5) | 1 (7) | 0 (6) |
| $G_3G_2$ 11 | 0 (12) | 1 (13) | 0 (15) | 1 (14) |
| $G_3\bar{G}_2$ 10 | 1 (8) | 0 (9) | 1 (11) | 0 (10) |

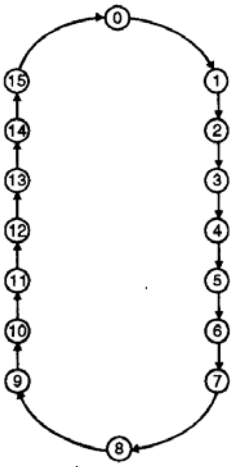
$$B_0 = \bar{G}_1\bar{G}_0(G_3\bar{G}_2 + \bar{G}_3 G_2) + \bar{G}_1 G_0(\bar{G}_3\bar{G}_2 + G_3 G_2)$$
$$+ G_1 G_0(G_3\bar{G}_2 + \bar{G}_3 G_2) + G_1\bar{G}_0(\bar{G}_3\bar{G}_2 + G_3 G_2)$$
$$= \bar{G}_1\bar{G}_0(G_3 \oplus G_2) + \bar{G}_1 G_0\overline{(G_3 \oplus G_2)}$$
$$+ G_1 G_0(G_3 \oplus G_2) + G_1\bar{G}_0\overline{(G_3 \oplus G_2)}$$
$$= (G_3 \oplus G_2) + (\bar{G}_1\bar{G}_0 + G_1 G_0) + \overline{(G_3 \oplus G_2)}(\bar{G}_1 G_0 + G_1\bar{G}_0)$$
$$= (G_3 \oplus G_2)\overline{(G_1 \oplus G_0)} + \overline{(G_3 \oplus G_2)}(G_1 \oplus G_0)$$
$$= A\bar{B} + \bar{A}B$$
$$= G_0 \oplus G_1 \oplus G_2 \oplus G_3$$

3) Circuit :



**Q.3(b) Design 4 bit synchronous converter using JK Flip-Flop. [10]**

**(A)**




| $Q_D$ | $Q_C$ | $Q_B$ | $Q_A$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |

| Present state $Q_n$ | Next state $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | × |
| 0 | 1 | 1 | × |
| 1 | 0 | × | 1 |
| 1 | 1 | × | 0 |

| Present state | | | | Next state | | | | Flip-flop input | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_D$ | $Q_C$ | $Q_B$ | $Q_A$ | $Q_{D+1}$ | $Q_{C+1}$ | $Q_{B+1}$ | $Q_{A+1}$ | $J_D$ | $K_D$ | $J_C$ | $K_C$ | $J_B$ | $K_B$ | $J_A$ | $K_A$ |
| 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 | 1 | 0 | 0 | 1 | × | 0 | 0 | × | 0 | × | 1 | × |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | × | 0 | 0 | × | 1 | × | × | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | × | 0 | 0 | × | × | 0 | 1 | × |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | × | 0 | 1 | × | × | 1 | × | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | × | 0 | × | 0 | 0 | × | 1 | × |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | × | 0 | × | 0 | 1 | × | × | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | × | 0 | × | 0 | × | 0 | 1 | × |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | × | 1 | × | 1 | × | 1 | × | 1 |

For $J_D$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | X | X | X | X |
| 10 | X | X | X | X |

$J_D = CBA$

For $K_D$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | X | X | X | X |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 |

$K_D = CB$

For $J_C$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | X | X | X | X |
| 11 | X | X | X | X |
| 10 | 0 | 0 | 1 | 0 |

$J_C = BA$

For $J_B$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | X | X |
| 01 | 0 | 1 | X | X |
| 11 | 0 | 1 | X | X |
| 10 | 0 | 1 | X | X |

$J_B = A$

For $K_C$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | 0 | 0 | 1 | 0 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | X | X | X | X |

$K_C = BA$

For $K_B$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | 1 | 0 |
| 01 | X | X | 1 | 0 |
| 11 | X | X | 1 | 0 |
| 10 | X | X | 1 | 0 |

$K_B = A$

For $J_A$

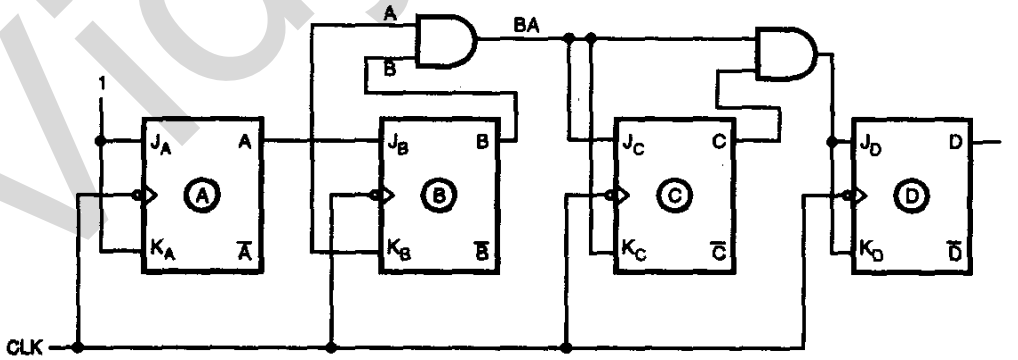
| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | X | X | 1 |
| 01 | 1 | X | X | 1 |
| 11 | 1 | X | X | 1 |
| 10 | 1 | X | X | 1 |

$J_A = 1$

For $K_A$

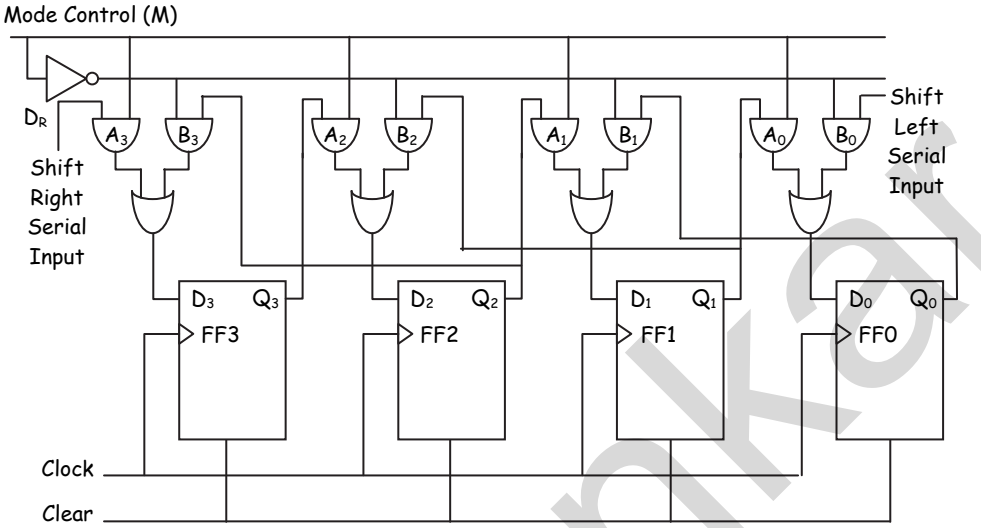
| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | X |
| 01 | X | 1 | 1 | X |
| 11 | X | 1 | 1 | X |
| 10 | X | 1 | 1 | X |

$K_A = 1$



**Q.4(a) Explain 4 bit bidirectional shift register.** **[10]**

**(A) Bi–Directional Shift Register**




When Mode Control M = 1 all the 'A' AND gates are enabled and DR is shifted to lie right, when the clock pluses are applied.

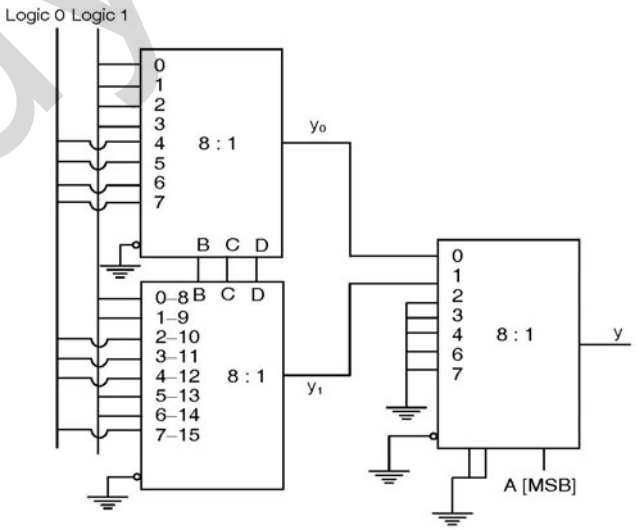
When Mode Control M =0 all the 'B' AND gates are enabled and DL is shifted to the left.

**Q.4(b) Implement following function using :** **[10]**

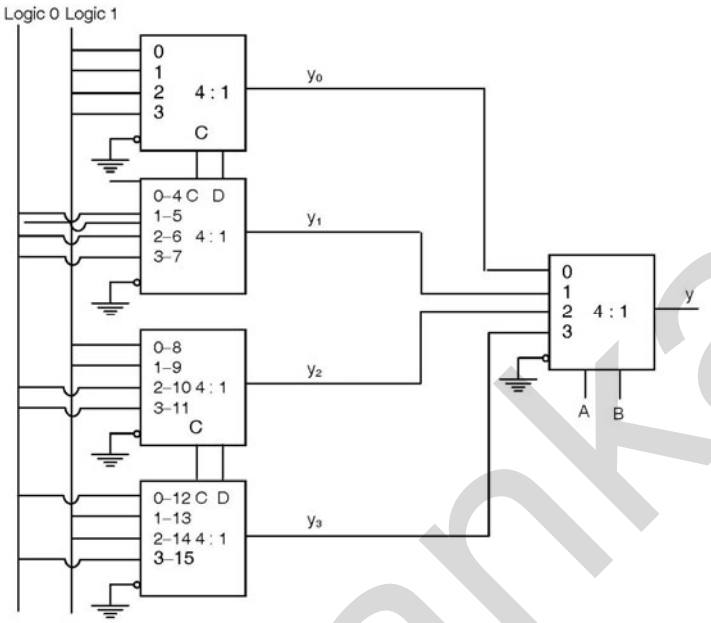
**(i) 8:1 mux, (ii) 4:1 mux, (iii) 2:1 mux**

**Y = ∑m(0,1,2,3,8,9,13,14)**

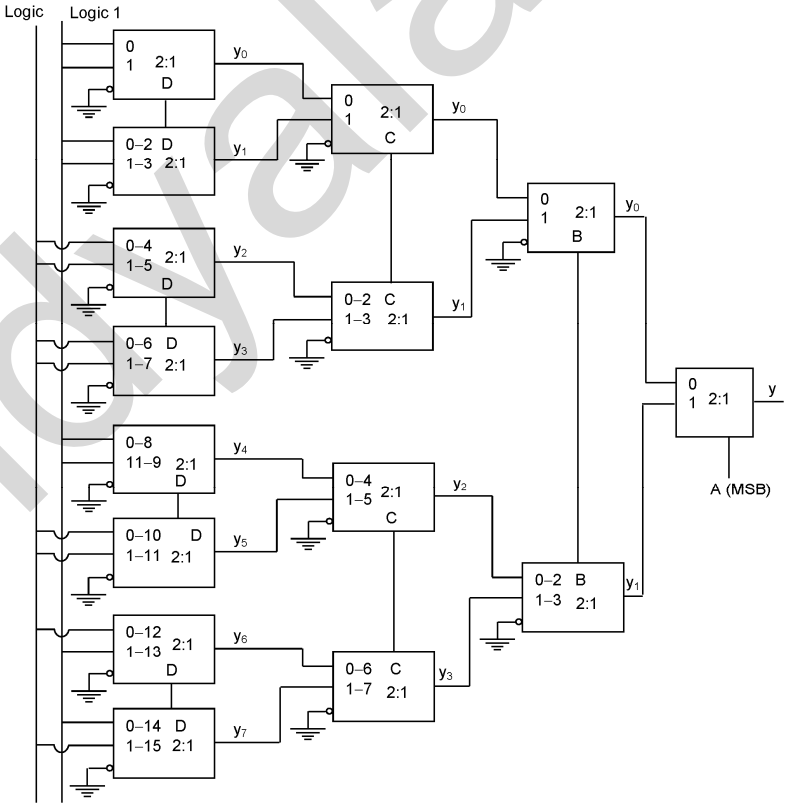
**(A) (i) Using 8 : 1**



### (ii) Using 4 : 1 Multiplexers :

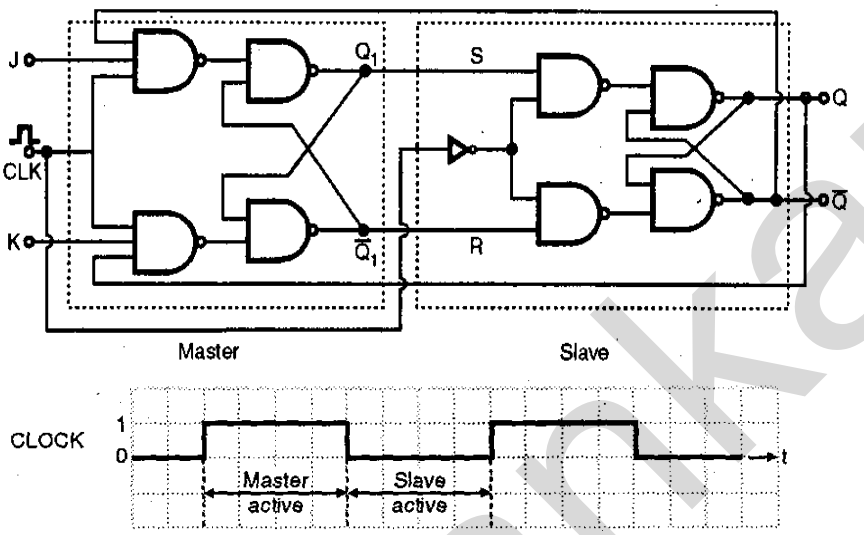



### (iii) Using 2 : 1 Multiplexers :



**Q.5(a) Draw neat circuit diagram and explain master slave JK flip-flop with NAND gates. [10]**

**(A) Master Slave JK Flip Flop**




Master is positive level triggered. But due to the presence of the inverter in the clock line, the slave will respond to the negative level. Hence when the clock = I (positive level) the master is active and the slave is inactive, whereas when clock = 0 (low level) the slave is active and the master is inactive.

**Case I** : Clock = X, J = K = 0

(i) For clock = 1, the master is active, slave inactive. As J = K = 0. Therefore, output of master, i.e., $Q_1$ and $\bar{Q}_1$ will not change. Hence the S and R inputs to the slave will remain unchanged.

(ii) As soon as clock = 0, the slave becomes active and master is inactive. But since the S and R inputs have not changed, the slave outputs will also remain unchanged.

∴ The outputs won't change if J = K = 0.

**Case II** : Clock = ⎍, J = K = 0

This condition has been already discussed in case f.

**Case III** : Clock = ⎍, J = 0 and K = 1.

- Clock = 1 : Master active, slave inactive.
- Therefore, outputs of the master become $Q_1 = 1$ and $\bar{Q}_1 = 0$. That means S = 0 and R = 1.
- Clock = 0 : Slave active, master inactive.

- Therefore, outputs of the slave become Q = 0 and $\bar{Q}$ = 1.
- Again if clock = 1 : Master active, slave inactive.
  Therefore, even with the changed outputs Q = 0 and $\bar{Q}$ = 1 feedback to master, its outputs will $Q_1$ = 0 and $\bar{Q}_1$ = 1. That means S = 0 and R = 1.
- Hence with clock = 0 and slave becoming active, the outputs of slave will remain Q = 0 and $\bar{Q}$ = 1.
- Thus we get a stable output from the Master Slave.

**Case IV :** Clock = ⎍, J = 1, K = 0

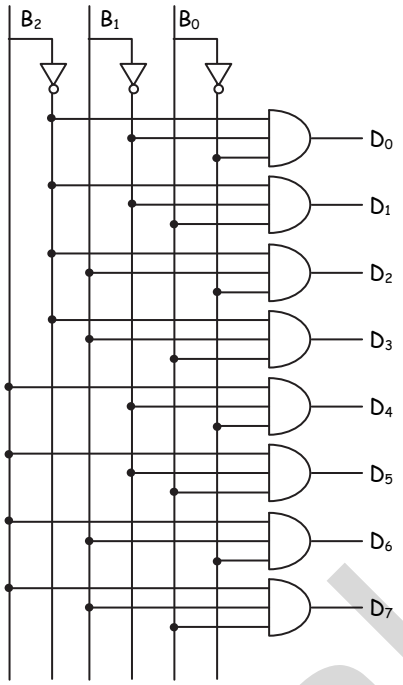
- Clock = 1 : Master active, slave inactive.
  Therefore, outputs of master become $Q_1$ = 0 and $\bar{Q}_1$ = 0, i.e., S = 1, R = 0.
- Clock = 0 : Master inactive, slave active.
  Therefore, outputs of slave become Q = 1 and $\bar{Q}$ = 0.
- Again if clock = 1 then it can be shown that the outputs of the slave are stabilized to Q = 1 and $\bar{Q}$ = 1.

**Case V :** Clock = ⎍, J = 1, K = 1.

- Clock = 1 : Master active, slave inactive.
  Therefore, outputs of master will toggle. So S and R also will be inverted.
- Clock = 0 : Master inactive, slave active.
  Therefore, outputs of the slave will toggle.
- These changed output are returned back to the master inputs.
- But since clock = 0, the master is still inactive. So it does not respond to these changed outputs.
- This avoids the multiple toggling which leads to the race around condition. Thus the master slave flip flop will avoid the race around condition.

**Q.5(b) Design 3:8 decoder using basic logic gates.** **[10]**

**(A)** Consider $B_2$, $B_1$, $B_0$ are binary inputs.
$D_0 - D_7$ are Decimal outputs.




**Functional Table :**

| $B_2$ | $B_1$ | $B_0$ | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Q.6(a) Explain features of VHDL.** **[6]**

**(A)**

- VHDL has powerful constructs
- VHDL language supports hierarchy (i.e. modelled using a set of interconnected components)
- VHDL is not case sensitive
- VHDL supports both synchronous and asynchronous timing models.
- Concurrency timing and clocking can be modeled using VHDL
- VHDL is target independent
- VHDL supports design library
- VHDL has flexible design methodologies i.e. TOP DOWN, BOTTOM UP, MIXED
- The logical behavior and timing behavior of the design can be modeled using VHDL.
- VHDL is not technology specific i.e. VHDL is not dependent on the specific manufacturer i.e. XILINX or LATTICE.
- VHDL's technology specific feature allows to specify components from various vendors
- VHDL also allows the user to specify his own data type and component
- VHDL is publicly available and has no proprietary.

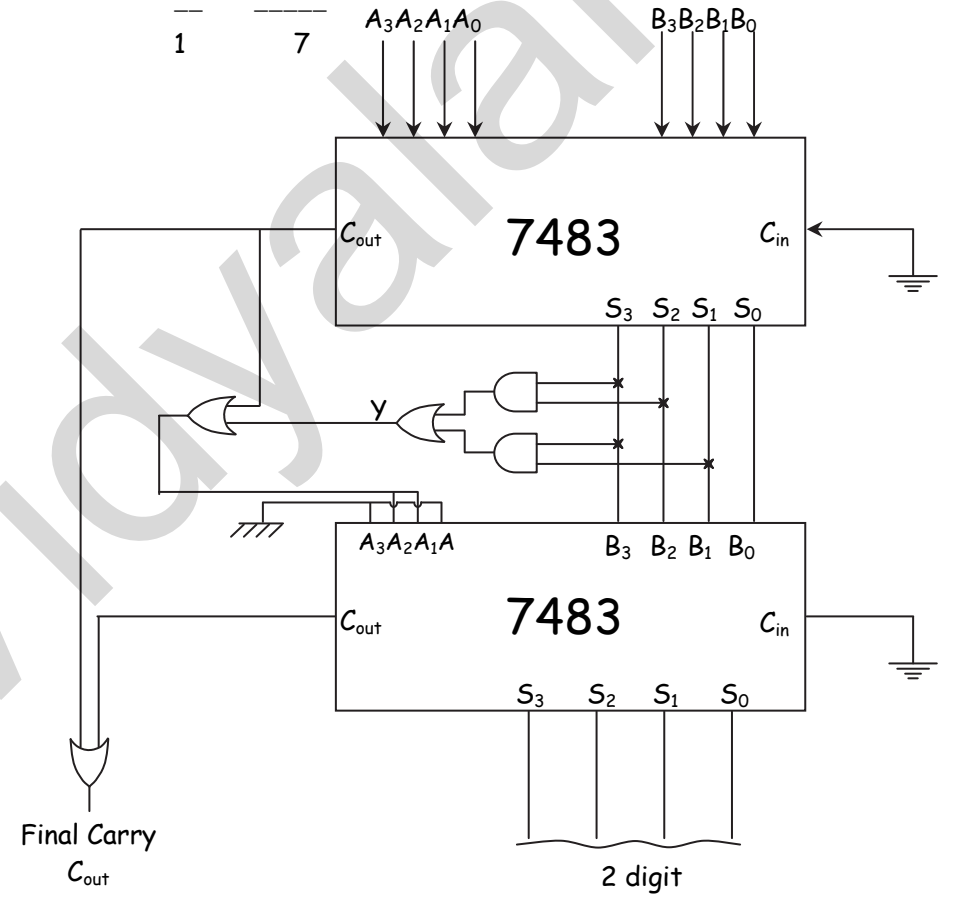
**Q.6(b) Explain BCD adder with suitable example.** **[7]**

**(A)** 1.

```
        6   →  0110
+ 5     →  0101
===        ----
 11        1011
           0110   →  add 6 if result is invalid.
          ----
          1 0001
          - ----
          1    1
```

2.

```
        8   →   1000
+9      →  + 1001
===        -----
17        1 0001
        + 0110     →  add 6
          -----
          1 0111
          -- -----
          1    7
```



**Q.6(c) Compare FPGA and CPLD. [7]**

**(A)**

| | CPLDs | FPGAs |
|---|---|---|
| Architecture | Large, wide fan-in blocks of AND-OR logic | Array of small logic blocks surrounded by I/O |
| Applications | Bus interfaces<br>Complex state machines<br>Fast memory interfaces<br>Wide decoders<br>PAL-device integration | Logic consolidation<br>Board integration<br>Replace obsolete devices<br>Simple state machines<br>Complex controllers/interfaces |
| Key Attributes | Fast pin-to-pin performance<br>Predictable timing<br>Easy to use | Very high density<br>Lots of I/Os and flip-flops<br>Generally lower power<br>SRAM devices are reprogrammable |
| Gate Capacity | 300-6,000 gates | 800-100,000 gates |
| Design Timing | Fixed, PAL-like<br>Very fast pin-to-pin performance | Application dependent<br>Very high shift frequencies |
| Number of I/Os | 30-200 | 50-400 |
| Number of Flip-flops | 30-200 | 100-5,500 |
| Process Technology | EPROM<br>EEPROM<br>FLASH | SRAM<br>Anti-fuse<br>EEPROM |
| In-System Programmable | Some EEPROM- and FLASH-based devices | SRAM-based devices and some EEPROM-based devices |
| One-Time Programmable (OTP) | EPROM devices in plastic packages. Some EEPROM- and FLASH-based devices | All anti-fuse-based devices |
| Power Consumption | 0.5-2.0W static<br>0.5-4.0W dynamic | Very low static<br>Dynamic consumption is application dependent, 0.1-2W typical |

❑ ❑ ❑ ❑ ❑