

EXPERIMENT - 03

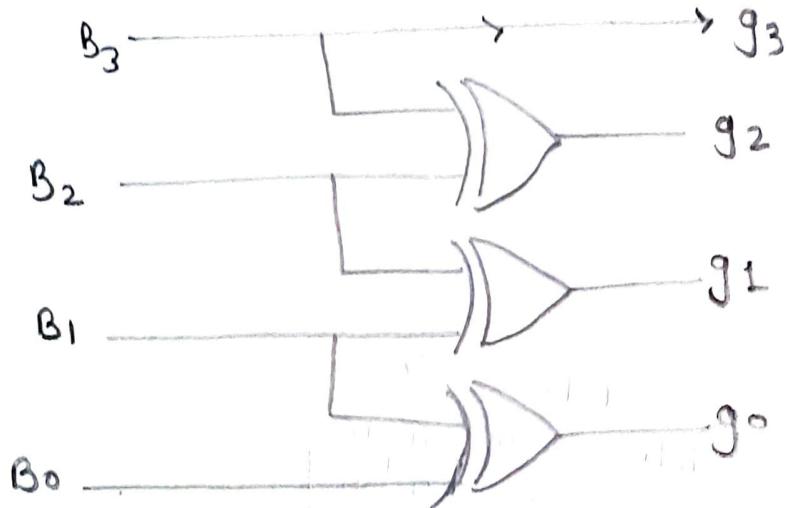
IMPLEMENTATION OF CODE CONVERTOR

Aim: To Design & Implement code converter
 Binary to Gray, Gray to Binary,
 BCD to Excess-3 & Excess-3 to BCD

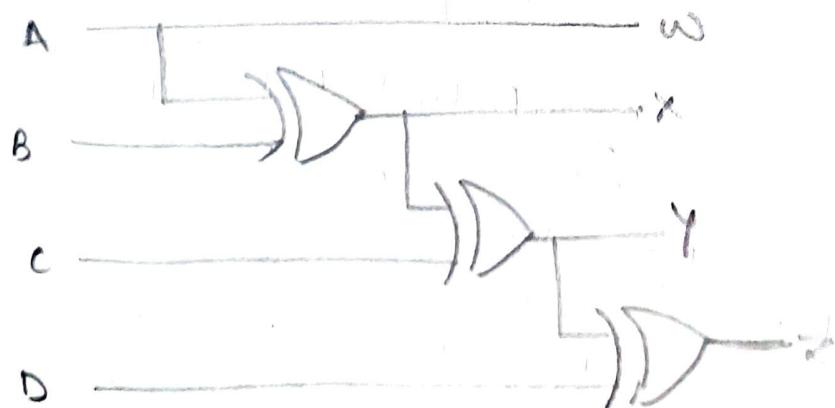
Software: Circuit verse | Multisim 14, IC7402, IC7403
 IC7432, IC7410, IC7404, IC Trains kit.

THEORY

Circuit: BINARY TO GRAY



Circuit: GRAY TO BINARY



BINARY TO GRAY

4-Bit Binary code Truth table 4-Bit -gray-code

| B_3 | B_2 | B_1 | B_0 |
|-------|-------|-------|-------|
| 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 |

| G_3 | G_2 | G_1 | G_0 |
|-------|-------|-------|-------|
| 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 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |

K-map

| | B_3B_2 | B_1B_0 | G_3 |
|----------|----------|----------|-------|
| B_3B_2 | 00 | 01 | 11 |
| 00 | 00 | 00 | 00 |
| 01 | 00 | 00 | 00 |
| 11 | 11 | 11 | 11 |
| 10 | 11 | 11 | 10 |

| | B_3B_2 | B_1B_0 | G_2 |
|----------|----------|----------|-------|
| B_3B_2 | 00 | 01 | 11 |
| 00 | 0 | 0 | 0 |
| 01 | 1 | 1 | 1 |
| 11 | 0 | 0 | 0 |
| 10 | 1 | 1 | 1 |

$$G_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2 = \boxed{B_3 \oplus B_2}$$

| q | 00 | 01 | 11 | 10 |
|-----|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

| | B_3B_2 | B_1B_0 | G_0 |
|----------|----------|----------|-------|
| B_3B_2 | 00 | 01 | 11 |
| 00 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 |
| 11 | 0 | 1 | 0 |
| 10 | 0 | 1 | 1 |

$$G_0 = B_0 \bar{B}_1 + B_1 \bar{B}_0$$

$$\boxed{G_0 = B_0 \oplus B_1}$$

G_1

| B_3B_2 | 00 | 01 | 11 | 10 |
|----------|----|----|----|----|
| 00 | 0 | 1 | 1 | 1 |
| 01 | 1 | 0 | 1 | 1 |
| 11 | 1 | 1 | 0 | 1 |
| 10 | 1 | 1 | 1 | 0 |

$$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$$

$$\boxed{G_1 = B_2 \oplus B_1}$$

GREY TO BINARY

| 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 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

G_3G_2

| B_3 | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

$$\boxed{B_3 \simeq G_3}$$

G_3G_2

| B_2 | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| 00 | | | | |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | | | |
| 10 | 1 | 1 | 1 | 1 |

$$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2$$

$$\boxed{B_2 = G_3 \oplus G_2}$$

| B ₂ | | | |
|-------------------------------|----|----|----|
| Q ₃ Q ₂ | 00 | 01 | 11 |
| 00 | 00 | 11 | 11 |
| 01 | 11 | 11 | 11 |
| 11 | 11 | 11 | 11 |
| 10 | 11 | 11 | 11 |

$$\begin{aligned}
 B_1 &= \bar{q}_3 \bar{q}_2 q_1 + \bar{q}_3 q_2 \bar{q}_1 + \\
 &\Rightarrow q_3 q_2 q_1 + q_3 \bar{q}_2 \bar{q}_1 \\
 &\Rightarrow q_3 \oplus q_2 \oplus q_1
 \end{aligned}$$

BCD to XS-3

BCD

| A | B | C | D |
|---|---|---|---|
| 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 |

| B ₁ | | | |
|-------------------------------|----|----|----|
| Q ₃ Q ₂ | 00 | 01 | 11 |
| 00 | 00 | 11 | 11 |
| 01 | 11 | 11 | 11 |
| 11 | 11 | 11 | 11 |
| 10 | 11 | 11 | 11 |

$$\begin{aligned}
 B_0 &= \bar{q}_3 \bar{q}_2 \bar{q}_1 q_0 + \bar{q}_3 \bar{q}_2 q_1 \bar{q}_0 + \\
 &\quad \bar{q}_3 q_2 \bar{q}_1 \bar{q}_0 + \bar{q}_3 q_2 q_1 q_0 + \\
 &\quad q_3 q_2 \bar{q}_1 q_0 + q_3 q_2 q_1 \bar{q}_0 + \\
 &\quad q_3 \bar{q}_2 \bar{q}_1 \bar{q}_0 + q_3 \bar{q}_1 q_1 q_0 \\
 \boxed{B_0 = q_3 \oplus q_2 \oplus q_1 \oplus q_0}
 \end{aligned}$$

XS 3

| X | Y | Z |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| X | X | X |
| X | X | X |
| X | X | X |
| X | X | X |

k-map

| AB | | CD | | | | | |
|----|----|----|----|----|----|----|----|
| 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

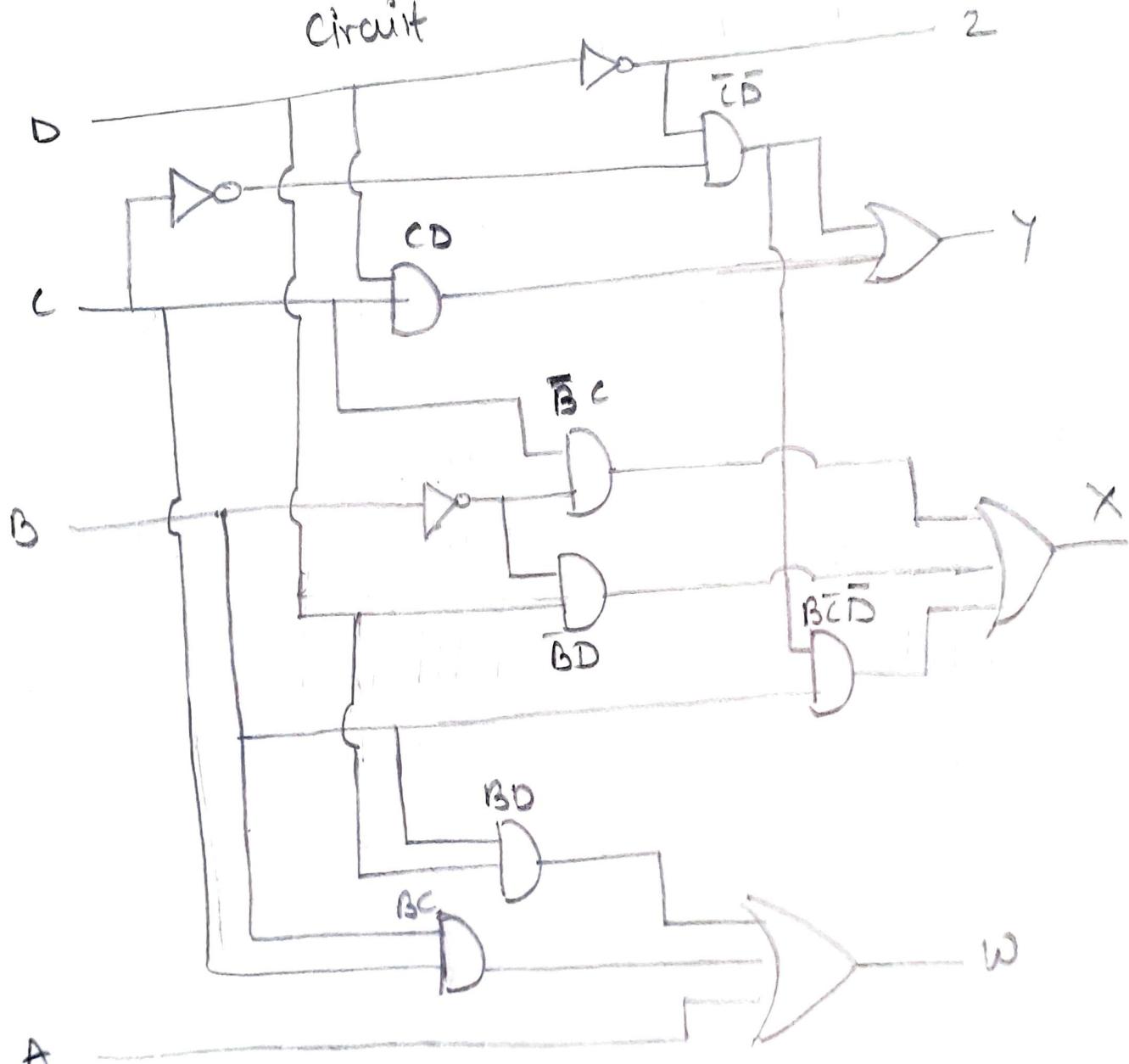
$w = A + BD + BC$

| AB | | CD | | | | | |
|----|----|----|----|----|----|----|----|
| 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| X | X | X | X | X | X | X | X |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

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

| | CD | AB |
|---------|--|-----------------------|
| 4 | 1 0 1 0 1 0 1 0 X X X X 1 0 X X | |
| Y | | $\bar{C}\bar{D} + CD$ |
| circuit | | |

| | CD | AB | 2 |
|---------------|--|----|---|
| 2 | 1 0 0 1 1 0 0 1 X X X X 1 0 X X | | |
| $2 = \bar{B}$ | | | |



EXCISE-3 TO BCD

| ε_3 | ε_2 | ε_1 | ε_0 |
|-----------------|-----------------|-----------------|-----------------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

| B_3 | B_2 | B_1 | B_0 |
|-------|-------|-------|-------|
| X | X | X | X |
| X | 0 | 1 | 0 |
| X | 0 | 0 | 0 |
| X | 0 | 1 | 0 |
| X | X | X | X |
| X | X | X | X |
| X | X | X | X |
| X | X | X | X |

K-Map

| $\varepsilon_3\varepsilon_2$ | $\varepsilon_1\varepsilon_0$ | 0111 | 10 |
|------------------------------|------------------------------|-------|----|
| 00 | X X | 0 | X |
| 01 | 0 0 | 0 | 0 |
| 11 | (X) X | (X) X | |
| 10 | 0 0 | 0 0 | 0 |

$$B_3 = \varepsilon_3\varepsilon_2 + \varepsilon_3\varepsilon_1\varepsilon_0$$

| $\varepsilon_3\varepsilon_2$ | $\varepsilon_1\varepsilon_0$ | B ₀ |
|------------------------------|------------------------------|----------------|
| X X | (X) | |
| (1) | X X | |
| X | X X | |
| (1) | (1) | |

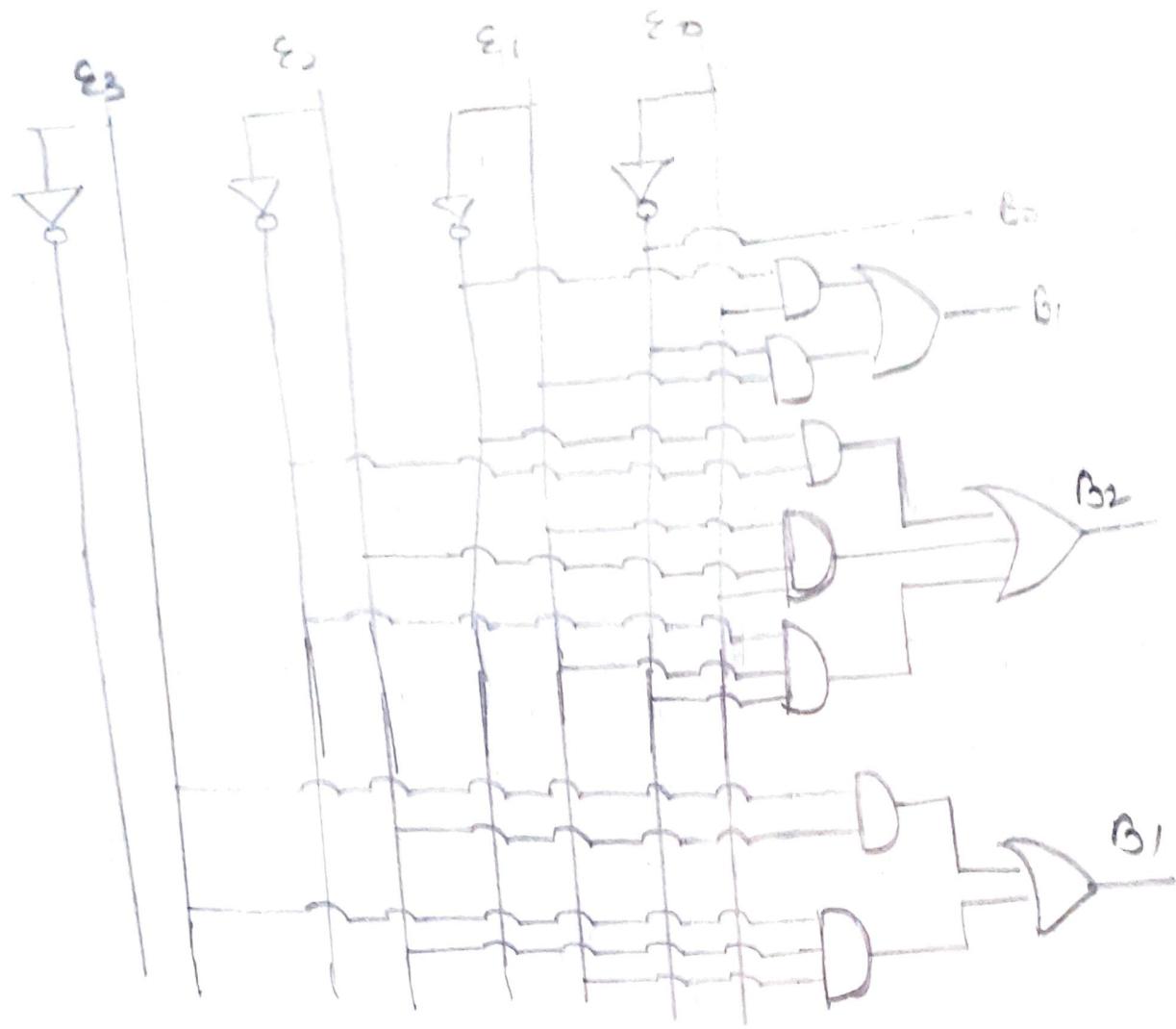
$$B_1 = \bar{\varepsilon}_2\bar{\varepsilon}_1 + \varepsilon_2\varepsilon_1\varepsilon_0 + \bar{\varepsilon}_2\varepsilon_1\bar{\varepsilon}_0$$

| $\varepsilon_3\varepsilon_2$ | $\varepsilon_1\varepsilon_0$ | B ₁ |
|------------------------------|------------------------------|----------------|
| X X | 0 | X |
| 0 1 | 0 1 | |
| 0 X | X X | |
| 0 1 | 0 1 | 1 |

$$B_1 = \bar{\varepsilon}_1\varepsilon_0 + \varepsilon_1\bar{\varepsilon}_0$$

| $\varepsilon_3\varepsilon_2$ | $\varepsilon_1\varepsilon_0$ | B ₀ |
|------------------------------|------------------------------|----------------|
| X X | 0 | X |
| 1 0 | 0 1 | |
| X X | X X | |
| 1 0 | 0 1 | 1 |

$$B_0 = \bar{\varepsilon}_0$$



Conclusion

Code connector implementation successful.

EXPERIMENT - 4

AIM: To Implement 4 Bit Binary Adder Using IC 7483,
To Implement 8 Bit Binary Adder Using IC 7483
To Implement 4 Bit BCD Adder Using IC 7483

Requirement | Software Used:

IC Trainer kit, patch cords, IC 7408 (AND), IC 7432 (OR),
IC 7483 (4 Bit Binary Adder), Multisim 14.

THEORY

IC 7483 - It is function specific or Application Specific IC (ASIC)

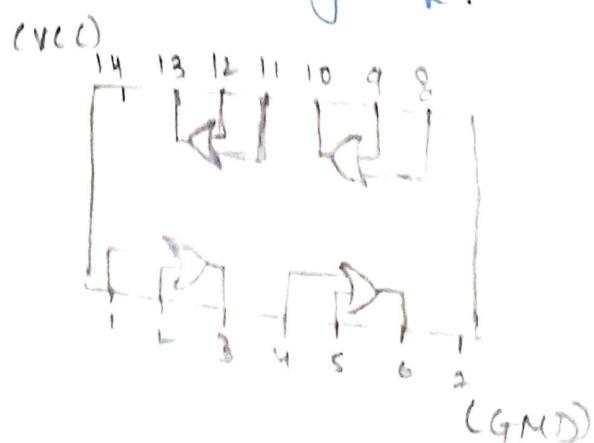
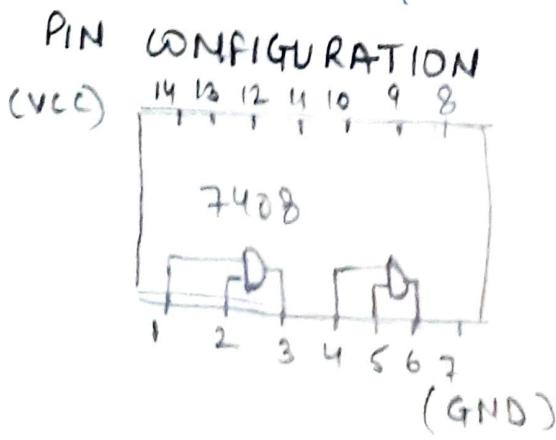
IC 7483 is a 4 Bit Binary Adder with 2 input of 4 Bits each and 1 (carry in) input. It has one output of 4 Bit & one (carry out) output.

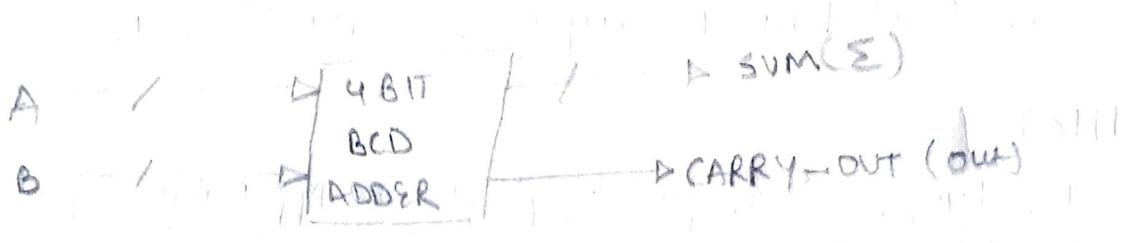
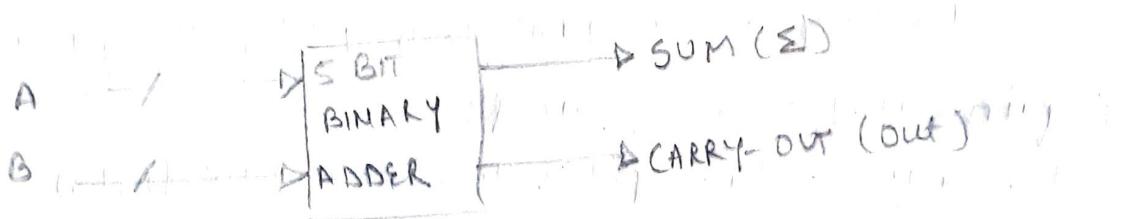
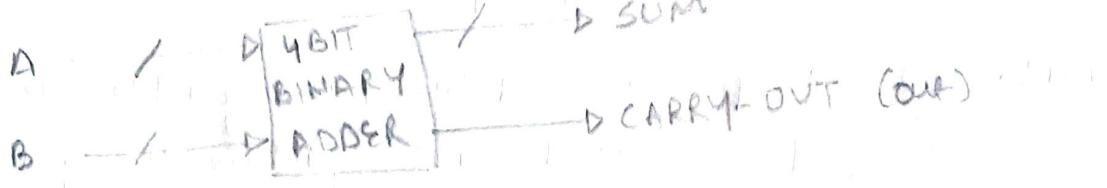
BCD Adder

BCD adder is a circuit that adds two BCD digits and produces a sum digit also in BCD

To implement BCD adder we require-

- 4 Bit Binary adder for initial addition
- logic circuit to detect sum greater than 9
- One more 4 Bit adder to add 6(0110) if sum is greater than 9 or carry is 1.



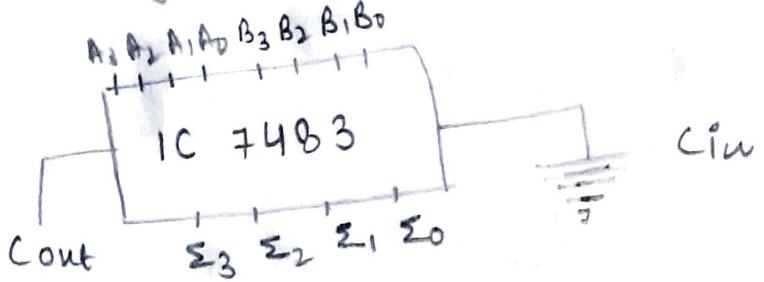


TRUTH TABLE

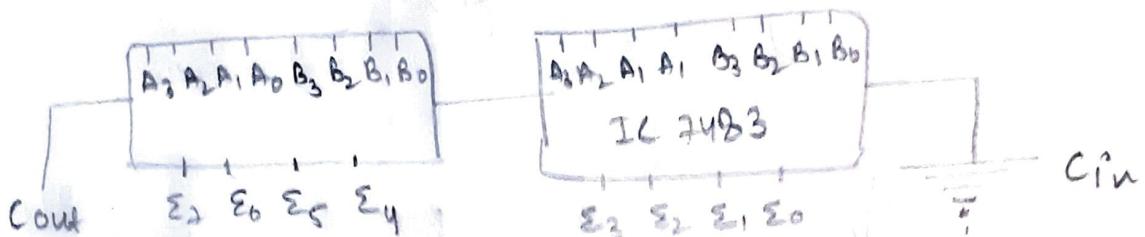
| A_3 | A_2 | A_1 | A_0 | B_3 | B_2 | B_1 | B_0 | Cout | Σ_3 | Σ_2 | Σ_1 | Σ_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|------|------------|------------|------------|------------|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 2 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

| A_3 | A_2 | A_1 | A_0 | B_3 | B_2 | B_1 | B_0 | Cout | Σ_3 | Σ_2 | Σ_1 | Σ_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|------|------------|------------|------------|------------|
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 2 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

Circuit Diagram



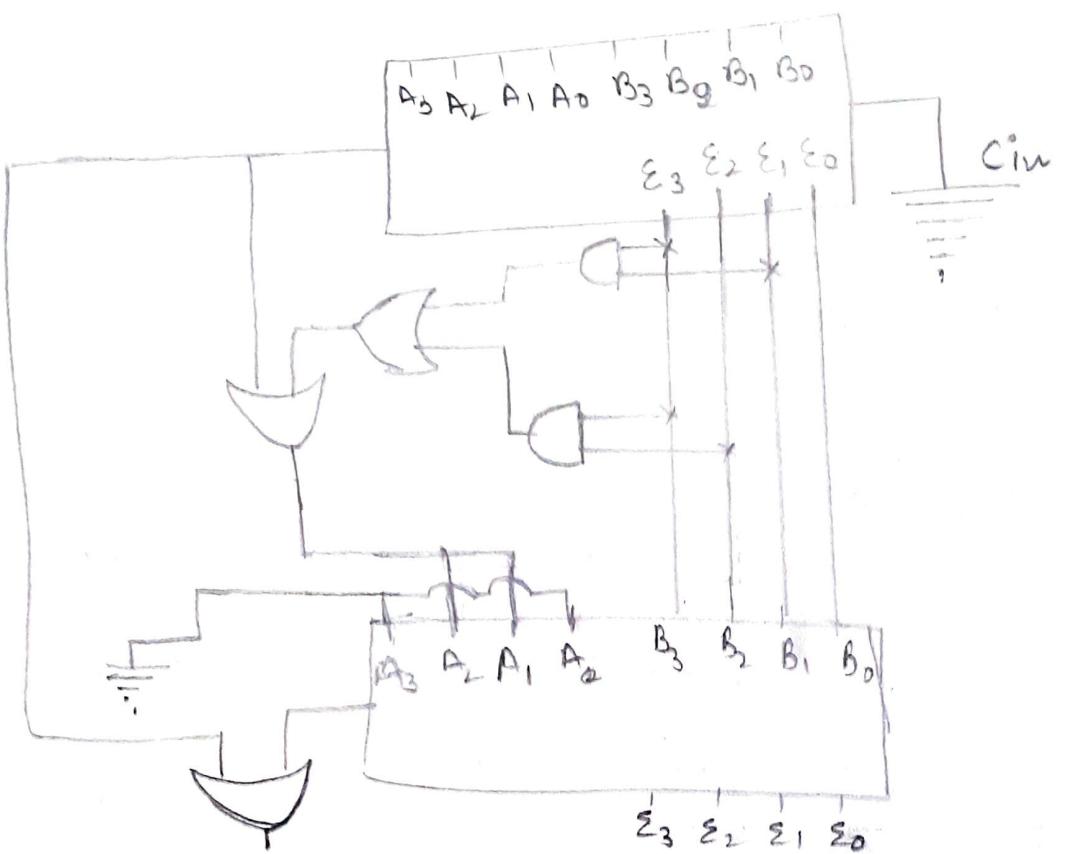
8-BIT BINARY ADDER USING IC7483



8-BIT BINARY ADDER USING IC7483

CIRCUIT DIAGRAM

4-BIT BCD ADDER USING IC7483



4-BIT BCD ADDER Truthy Table

| A ₃ | A ₂ | A ₁ | A ₀ | B ₃ | B ₂ | B ₁ | B ₀ | Cout | S ₃ | S ₂ | S ₁ | S ₀ | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------|----------------|----------------|----------------|----------------|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 4+3=7 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 5+3=8 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 9+9=18 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2+2=4 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 5+5=10 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 3+6=9 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0+2=15 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 3+8=11 | 1 | 0 | 0 | 0 | 1 |

Conclusion

In this lab we learned to build 4-bit Binary adder using IC7483, 8 Bit Binary Adder using IC7483 and 4 Bit BCD Adder using IC7483

EXPERIMENT - 5.

Multiplexers & Demultiplexers

AIM :- Implement Boolean expression using MUX 74151, 74157, and DMUX 74154.

Conversion of 8:1 MUX using 2:1 MUX 16:1, MUX using 4:1.

Hardware / Software / IC Required.

IC 74151, IC 74157, IC 74154, 74138.

Theory

Multiplexer: It is a device that has multiple input & single line output.

The select line determine which input is connected to the output, and also to increase the amount of data that can be sent over a network within certain time.

It is also called a 'data selector'.

8:1 MUX: This type of MUX takes 8 inputs and give single output, also it has 3 select lines.

2:1 MUX: This MUX takes 2 input & 1 output is given and it has a 3 sel

Boolean Expression

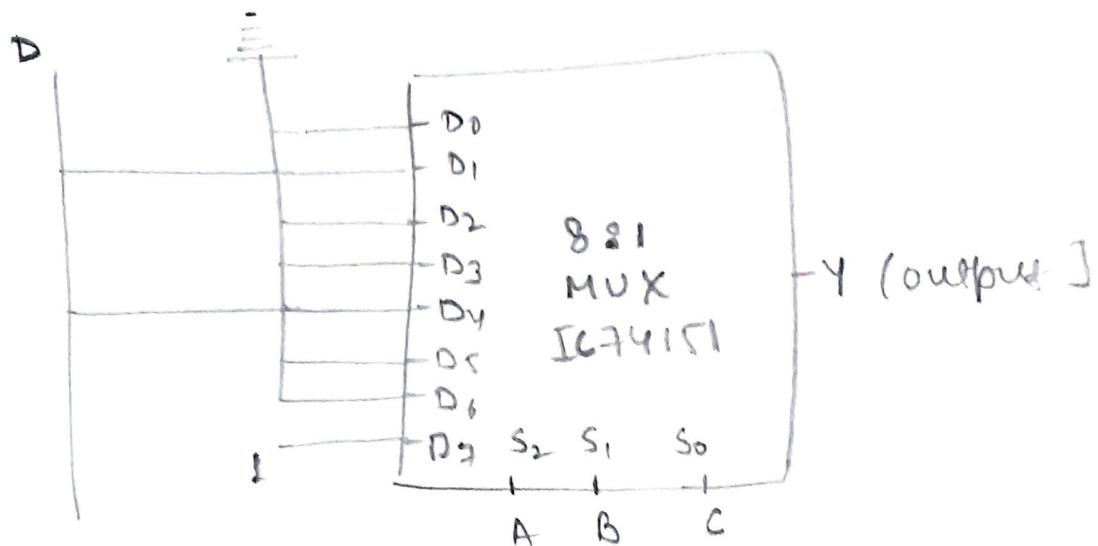
$$f(A, B, C, D) = \Sigma m(7, 9, 12, 15) \text{ using MUX IC 74151}$$

Truth table

| Decimal | Binary | output. |
|---------|--------|---------|
| 0 | 0000 | 0 |
| 1 | 0001 | 0 |
| 2 | 0010 | 0 |
| 3 | 0011 | 0 |
| 4 | 0100 | 0 |
| 5 | 0101 | 0 |
| 6 | 0110 | 0 |
| 7 | 0111 | 1 |
| 8 | 1000 | 0 |
| 9 | 1001 | 1 |
| 10 | 1010 | 0 |
| 11 | 1011 | 0 |
| 12 | 1100 | 1 |
| 13 | 1101 | 0 |
| 14 | 1110 | 0 |
| 15 | 1111 | 1 |

K-map.

| | | $B_3B_2B_1$ |
|-------------|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| \bar{B}_0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| B_0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | |
| | | D_0 | D_1 | D_2 | D_3 | D_4 | D_5 | D_6 | D_7 |

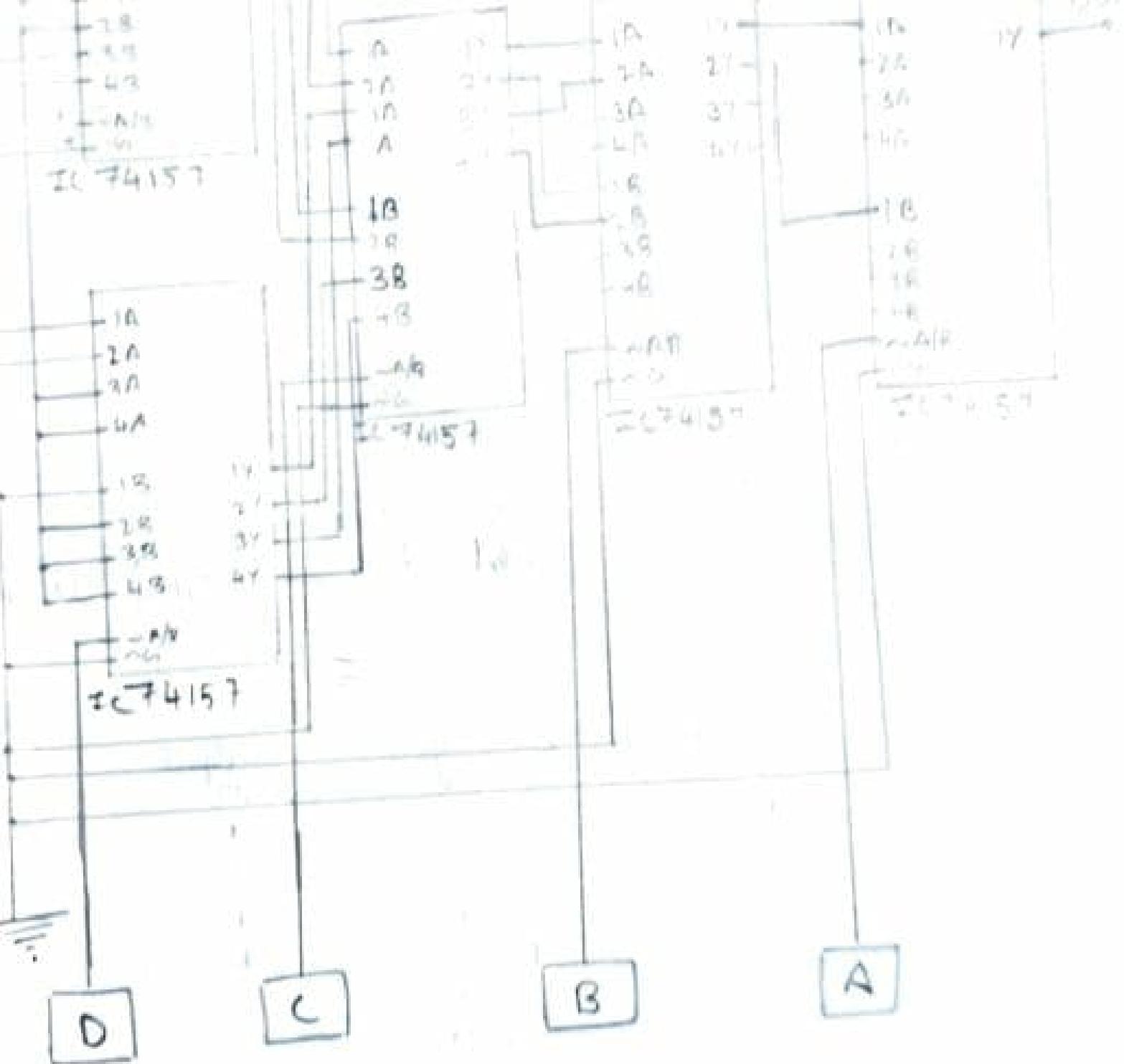


Boolean expression:

$F(A, B, C, D) = \sum m(1, 3, 4, 11, 12, 13, 14, 15)$ using
MUX IC 74151.

Truth table

| Decimal | Binary | Output |
|---------|--------|--------|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 0 |
| 3 | 0011 | 1 |
| 4 | 0100 | 1 |
| 5 | 0101 | 0 |
| 6 | 0110 | 0 |
| 7 | 0111 | 0 |
| 8 | 1000 | 0 |
| 9 | 1001 | 0 |
| 10 | 1010 | 0 |
| 11 | 1011 | 1 |
| 12 | 1100 | 1 |
| 13 | 1101 | 1 |
| 14 | 1110 | 1 |
| 15 | 1111 | 1 |



Boolean Expression

$$f(A, B, C, D) = \sum m(1, 2, 4, 11, 12, 13, 14, 15)$$

Demultiplexer :-

It is a device that has single input & multiple outputs. The select line determines which input is connected to the output and also to limit the amount of data that can be sent over a network within certain time.

1:16 Demux :-

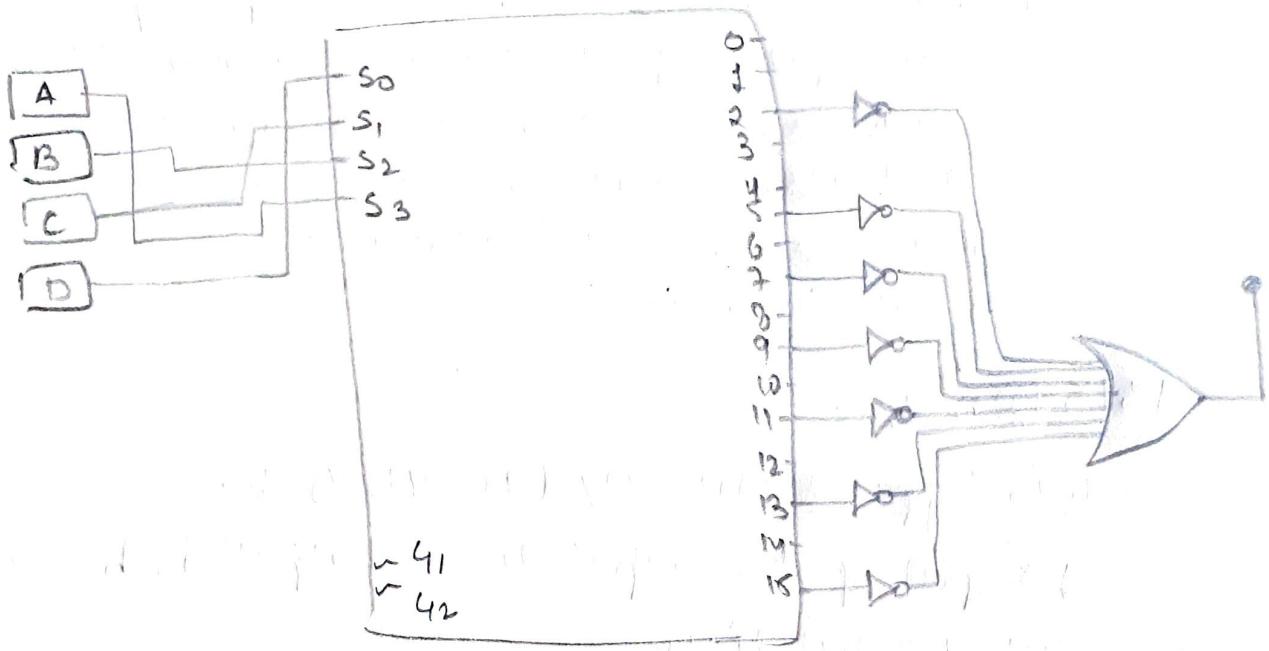
This type of Demux (IC74154) has 2 input & 16 output lines. Output is based on 4 select lines.

$$F(A, B, C, D) = \sum(2, 5, 7, 9, 11, 13, 15)$$

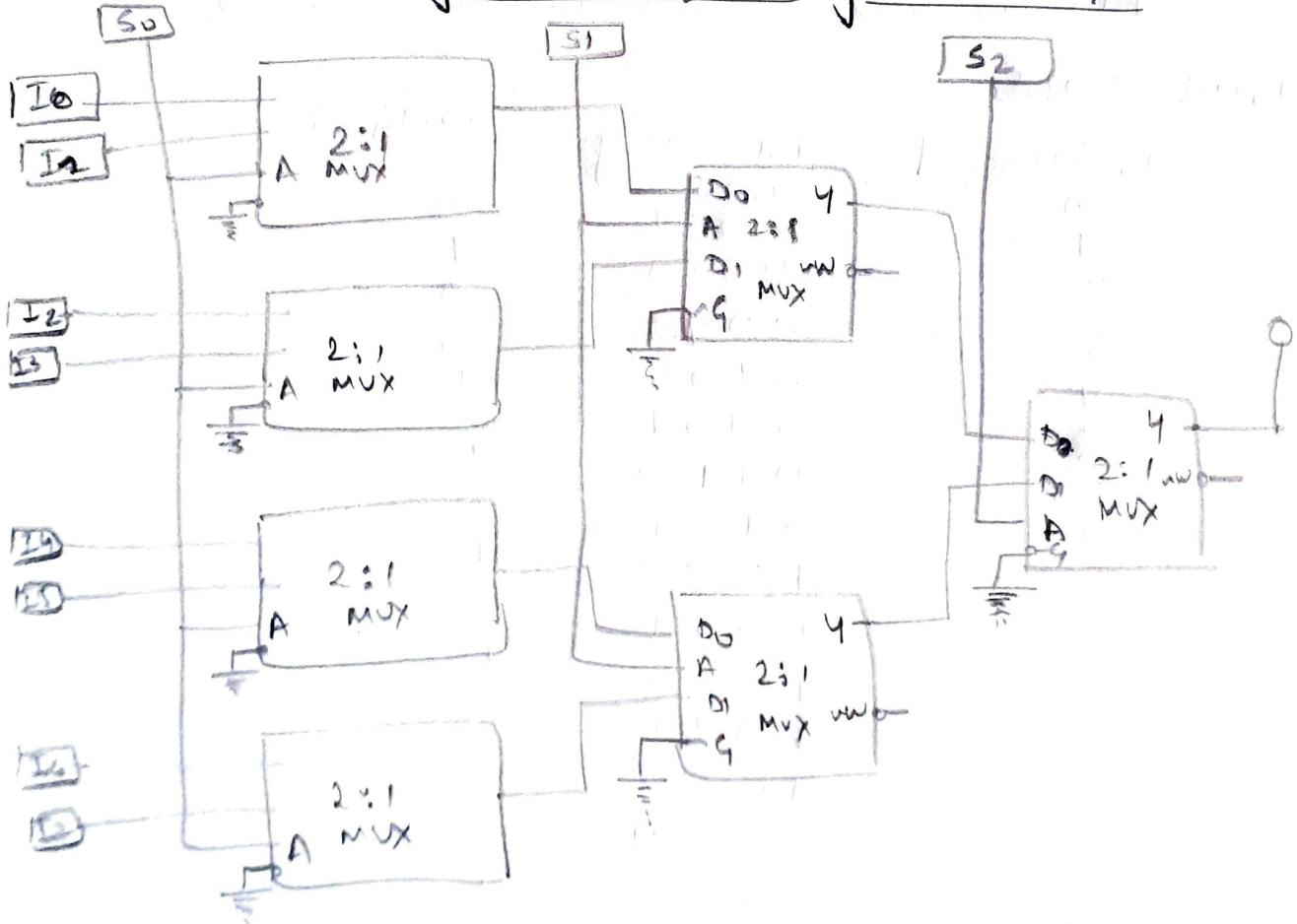
Truth table

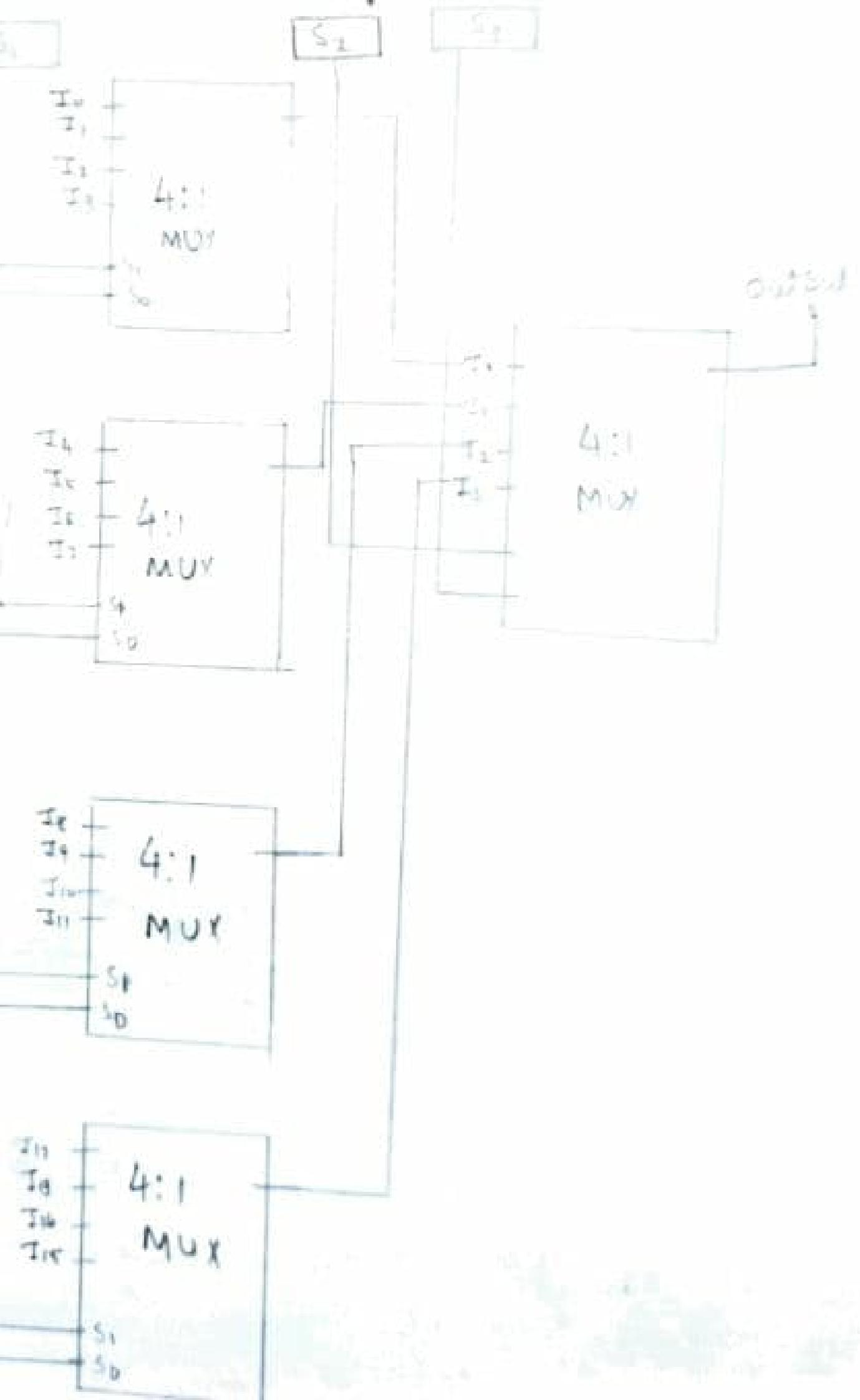
| Decimal | Binary | Output |
|---------|--------|--------|
| 0 | 0000 | 0 |
| 1 | 0001 | 0 |
| 2 | 0010 | 1 |
| 3 | 0011 | 0 |
| 4 | 0100 | 0 |
| 5 | 0101 | 0 |
| 6 | 0110 | 1 |
| 7 | 0111 | 0 |
| 8 | 1000 | 1 |
| 9 | 1001 | 0 |
| 10 | 1010 | 1 |
| 11 | 1011 | 0 |
| 12 | 1100 | 1 |
| 13 | 1101 | 0 |
| 14 | 1110 | 1 |
| 15 | 1111 | 0 |

Circuit Diagram

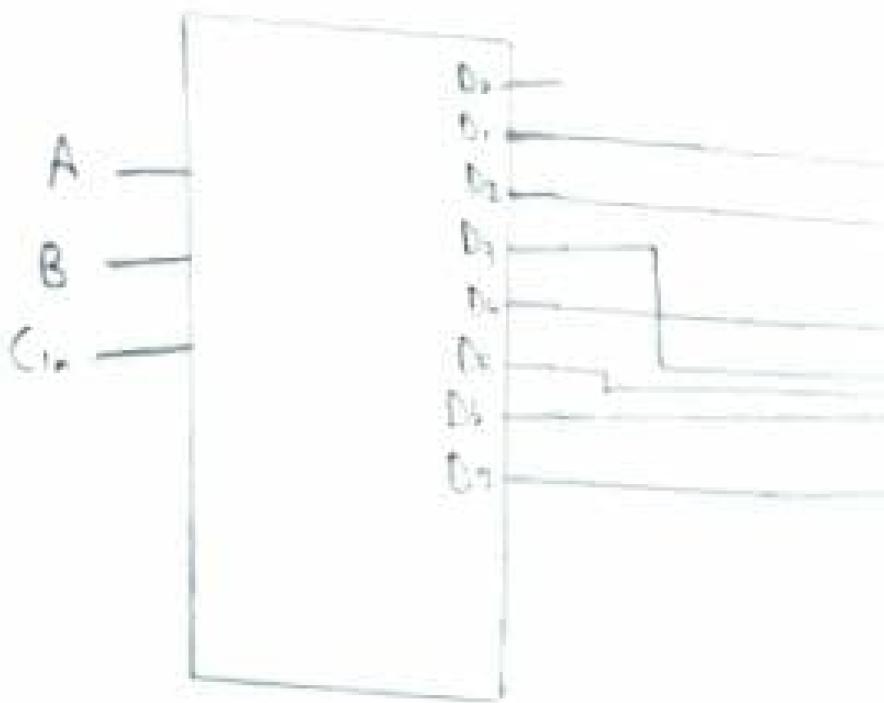


Conversion of 8:1 MUX using 2:1 MUX.





| A | B | C _{in} | Sum | Carry |
|---|---|-----------------|-----|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



Conclusion

In this lab we implemented Book 6 using MUX & DEMUX, conversion of full adder, and converting 2:1 Mux and 4:1 MUX to 16:1 MUX.

EXPERIMENT - 6

Parity generator / checker

AIM :- Design & Implement Parity generator and checker using EX-OR.

Hardware | Software | Sc required

XOR gate (IC 7486), X-NOR Gate (IC 74077), Multisim / circuit vise.

Theory

Even Parity generator

If the data input contains odd no of 1's it will produce logic 1 or high and on even count of 1's it will give low output (0)

TRUTH TABLE

| Input | | | Output |
|-------|-------|-------|--------|
| B_2 | B_1 | B_0 | P |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Circuit diagram



k-map



$$\begin{aligned}
 P = & B_2 \bar{B}_1 B_0 + \bar{B}_2 B_1 \bar{B}_0 + B_2 \bar{B}_1 \bar{B}_0 \\
 & + B_2 (B_1 B_0 + B_1 \bar{B}_0) + B_2 (\bar{B}_1 B_0 + \bar{B}_1 \bar{B}_0) \\
 & + B_2 (\oplus B_1 \oplus B_0)
 \end{aligned}$$

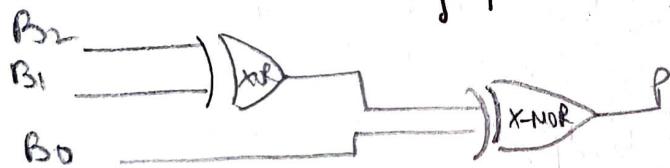
2) Odd Parity Generator

If data input has even no. of 1's output will be high (1) otherwise low (0).

TRUTH TABLE

| Input | | | Output |
|-------|-------|-------|--------|
| B_2 | B_1 | B_0 | P |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Circuit (Odd Parity generator)



K-Map

| $B_2\bar{B}_1$ | \bar{B}_2B_1 | $\bar{B}_2\bar{B}_1$ | B_2B_1 | $B_2\bar{B}_0$ |
|----------------|----------------|----------------------|----------|----------------|
| B_0 | (1) | 0 | (1) | 0 |
| | 0 | (1) | 0 | (1) |

$$\begin{aligned}
 P &= \bar{B}_2\bar{B}_1B_0 + \bar{B}_2B_1B_0 + \bar{B}_2\bar{B}_1B_0 + B_2\bar{B}_1B_0 \\
 &= B_2(\bar{B}_1B_0 + B_1B_0) + B_2(\bar{B}_1B_0 + B_1\bar{B}_0) \\
 &\Rightarrow (B_2 \oplus B_1) \odot B_0
 \end{aligned}$$

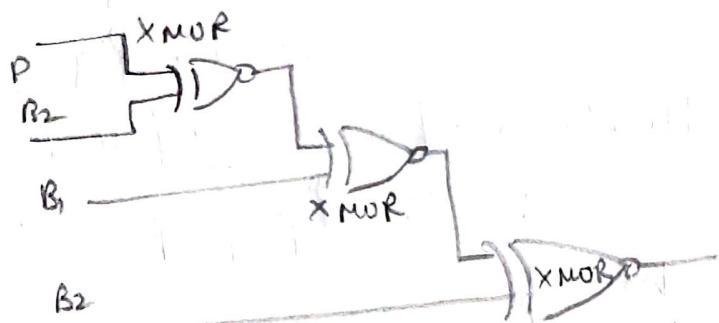
Even Parity checker.

It will produce a logic 1 at its output if data word contains even no. of 1's otherwise it'll give 0 i.e. of odd no. of 1's

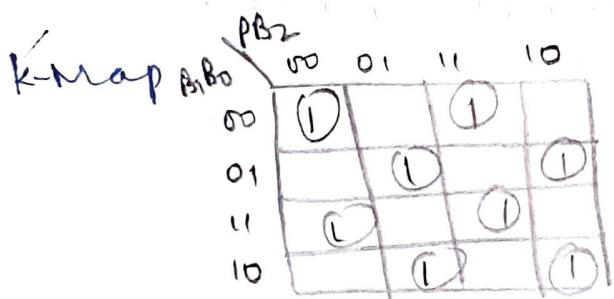
Truth Table

| Input | | | | Output |
|-------|----------------|----------------|----------------|--------|
| P | B ₂ | B ₁ | B ₀ | E |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Circuit diagram.



Even Parity checker



$$E = P \text{XOR} B_2 \text{XOR} B_1 \text{XOR} B_0$$

Conclusion

Successfully designed and implemented Parity generator & checker.

EXPERIMENT - 7

Magnitude Comparitor

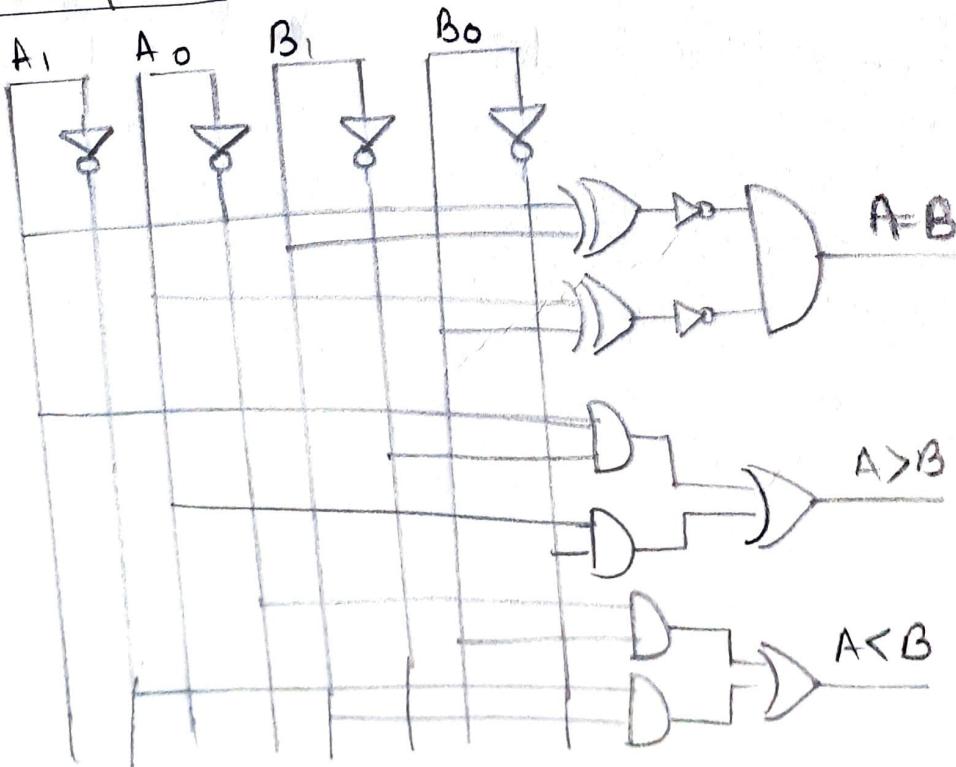
Aim: Design & implement 2, 3, 4 Magnitude comparitor using logic gates 4-Bit Bit magnitude comparators using IC 7485

Software used → circuit vise (multisim)

Theory The comparison of two no. is an operator that determine one no. in greater than, less than (or) equal to other no.

The outcome of the comparitor is specified by three binary value that indicate whether $A > B$, $A = B$ or $A < B$

LOGIC DIAGRAM



| $A_1 A_0$ | $B_1 B_0$ | 00 | 01 | 11 | 10 |
|-----------|-----------|----|----|----|----|
| 00 | 00 | 1 | | | |
| 01 | 01 | | 1 | 1 | |
| 11 | 11 | 1 | 1 | 1 | |
| 10 | 10 | | 1 | 1 | 1 |

$$A > B = A_0 \bar{B}_0 B_1 + A_1 \bar{B}_1 + \\ A_1 A_0 \bar{B}_0$$

| $A_1 A_0$ | $B_1 B_0$ | 00 | 01 | 11 | 10 |
|-----------|-----------|----|----|----|----|
| 00 | 00 | 1 | | | |
| 01 | 01 | | 1 | 1 | |
| 11 | 11 | 1 | 1 | 1 | |
| 10 | 10 | | 1 | 1 | 1 |

$$A < B = \bar{A}_1 \bar{A}_0 B_0 B_1 + \bar{A}_0 B_0 B_1 \\ + \bar{A}_1 B_1$$

| $A_1 A_0$ | $B_1 B_0$ | 00 | 01 | 11 | 10 |
|-----------|-----------|----|----|----|----|
| 00 | 00 | 1 | | | |
| 01 | 01 | | 1 | 1 | |
| 11 | 11 | 1 | 1 | 1 | |
| 10 | 10 | | 1 | 1 | 1 |

$$A = B = (A_0 B_0) (A_{10} B_1)$$

Truth Table

| A_1 | A_0 | B_1 | B_0 |
|-------|-------|-------|-------|
| 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 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |

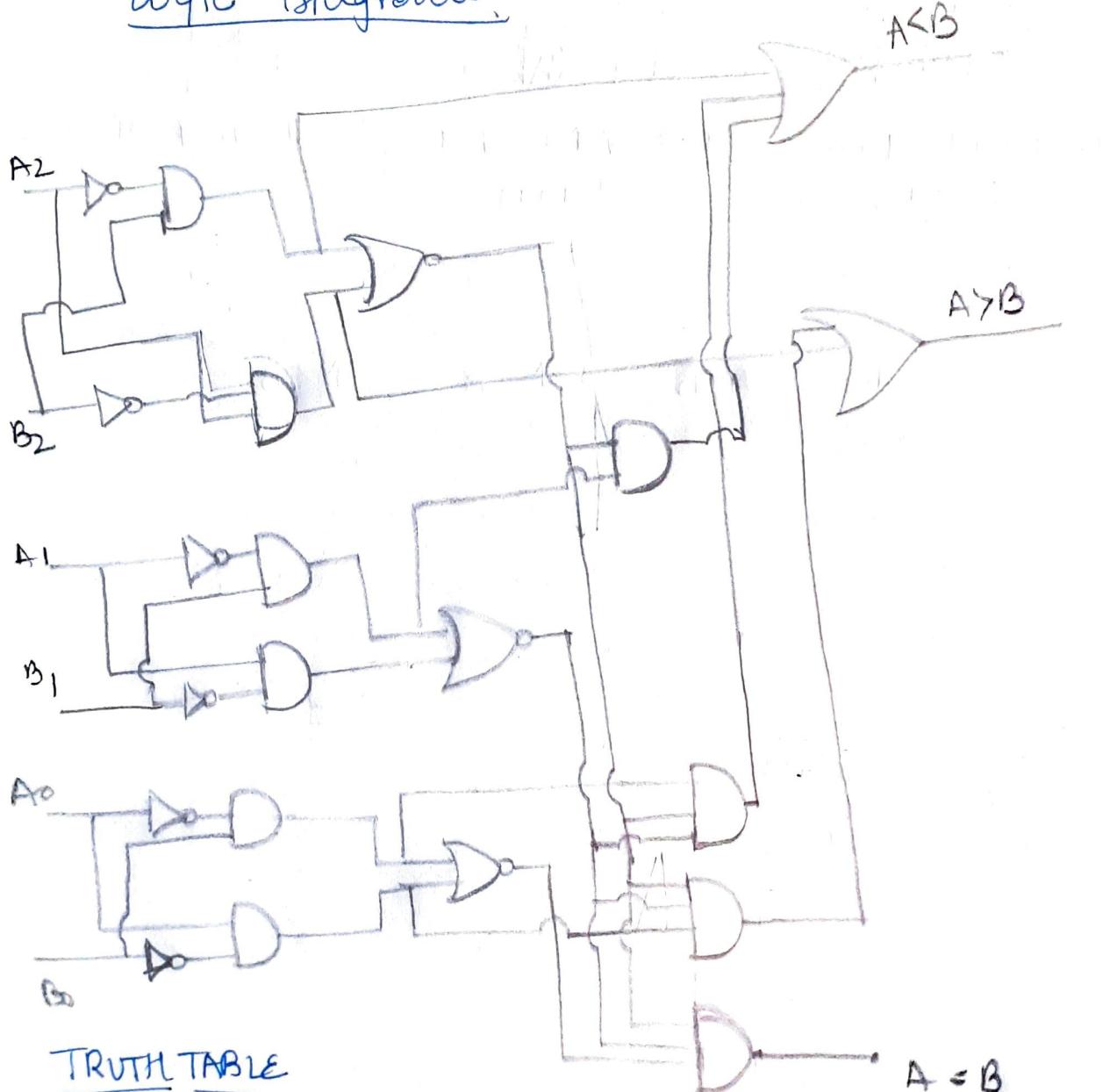
| $A > B$ | 0000000 | 1000000 |
|---------|---------|---------|
| 0 | 0000000 | 1000000 |
| 1 | 1000000 | 1100000 |
| 2 | 1100000 | 1110000 |
| 3 | 1110000 | 1111000 |

| $A = B$ | 1000000 | 0100000 |
|---------|---------|---------|
| 1 | 1000000 | 0100000 |
| 0 | 0100000 | 0010000 |
| 1 | 0010000 | 0001000 |
| 0 | 0001000 | 0000100 |

| $A < B$ | 0 | 1 | 2 | 3 |
|---------|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 1 | 1 | 1 | 0 |

3-Bit Magnitude Comparator

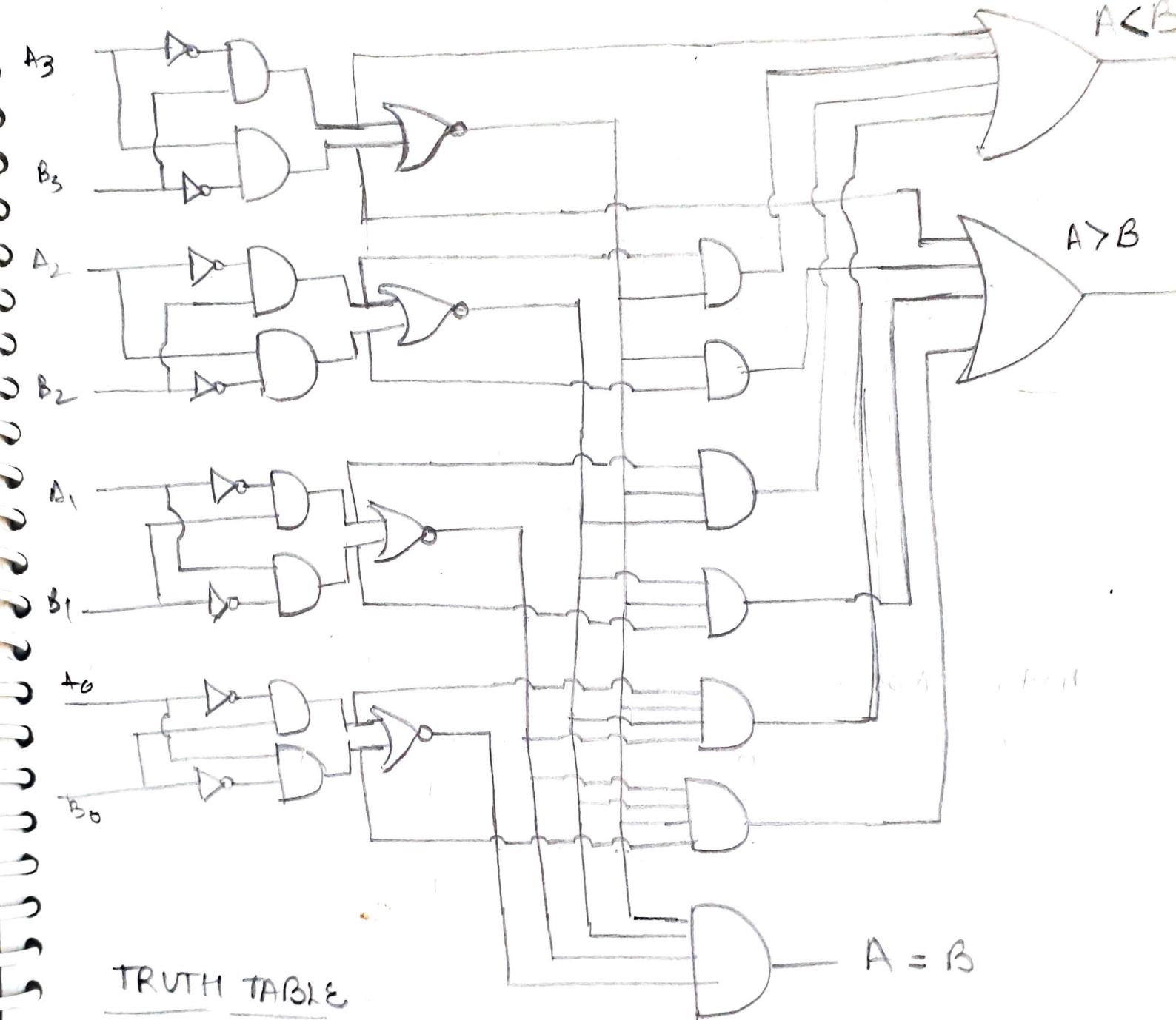
Logic Diagram.



TRUTH TABLE

| A_2 | A_1 | A_0 | B_2 | B_1 | B_0 | $A < B$ | $A = B$ | $A > B$ |
|-------|-------|-------|-------|-------|-------|---------|---------|---------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

4-BIT Magnitude Comparator



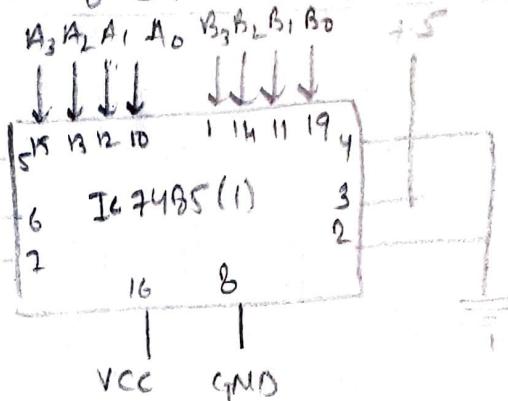
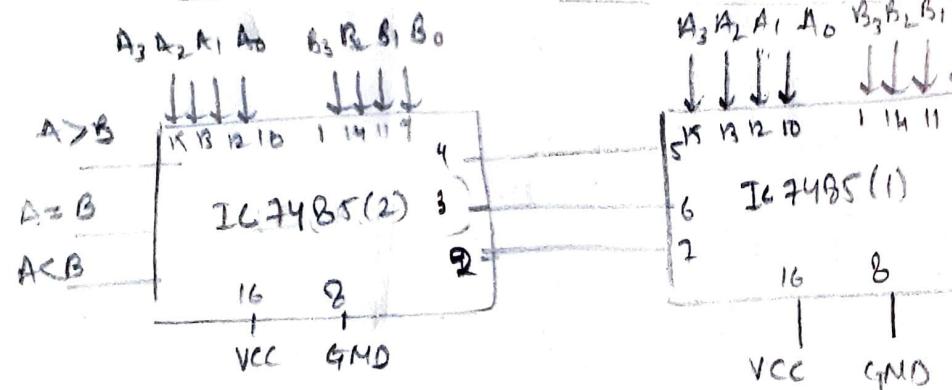
TRUTH TABLE

| A_3 | A_2 | A_1 | A_0 | B_3 | B_2 | B_1 | B_0 | $A < B$ | $A = B$ | $A > B$ |
|-------|-------|-------|-------|-------|-------|-------|-------|---------|---------|---------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

PIN DIAGRAM

| | | | |
|---------------|---|----|-----|
| B_3 | 1 | 16 | VCC |
| 1 ($A > B$) | 2 | 15 | A3 |
| 1 ($A = B$) | 3 | 14 | B2 |
| 1 ($A > B$) | 4 | 13 | A2 |
| ($A > B$) | 5 | 12 | A1 |
| ($A = B$) | 6 | 11 | B1 |
| ($A < B$) | 7 | 10 | A0 |
| GND | 9 | 9 | B0 |

LOGIC DIAGRAM FOR 8 BIT - MAGNITUDE COMPARATOR



TRUTH TABLE

| A | B | $A > B$ | $A = B$ | $A < B$ |
|---------|---------|---------|---------|---------|
| 0 0 0 0 | 0 0 0 0 | 0 | 1 | 0 |
| 0 0 0 1 | 0 0 0 1 | 1 | 0 | 0 |
| 0 0 0 0 | 0 0 0 0 | 0 | 1 | 0 |
| 0 0 0 0 | 0 0 0 0 | 0 | 0 | 1 |
| 0 0 0 0 | 0 1 0 1 | 0 | 0 | 0 |
| 0 1 0 1 | 0 0 0 0 | 1 | 0 | 0 |
| 1 1 1 1 | 1 1 1 1 | 0 | 1 | 0 |

CONCLUSION

Thus the design & implementation of magnitude comparator were done.

EXPERIMENT - 8

FLIP-FLOP CONVERSIONS

AIM: Implementation of conversion of

- i) SR to JK
- ii) JK to SR
- iii) JK to T
- iv) JK to D
- v) SR to D
- vi) D to SR
- vii) D to JK

Software used Circuit Vuise / multism

Theory flip-flop

flip-flop is a sequential circuit driven by clock input either by positive edge or by -ve edge. It is binary storage device capable of storing one of information.

The flip-flop are of various types. Their characteristics can be easily seen from their truth table.

PRESENT & CLEAR

When the power is turn on, the state of flip flop is uncertain. It may come to set ($Q=1$) or reset ($Q=0$). In many application, it is necessary to initially set or reset the flip-flop. Such initial state of flip-flop can be accomplished by using the direct or asynchronous inputs of flip-flops. These inputs are Present (\bar{P}) and Clear (\bar{C}). They can be applied anytime b/w clock pulse and are not in synchronism with clock.

SR to JK flip-flop

Logic Diagram [SR to JK Flip-Flop]



Truth Table

| J | K | Q_p | Q_n | S | R |
|---|---|-------|-------|---|---|
| 0 | 0 | 0 | 0 | 0 | X |
| 0 | 0 | 1 | 1 | X | 0 |
| 0 | 1 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | X | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

$\{ 0, 1 \}$ Toggle

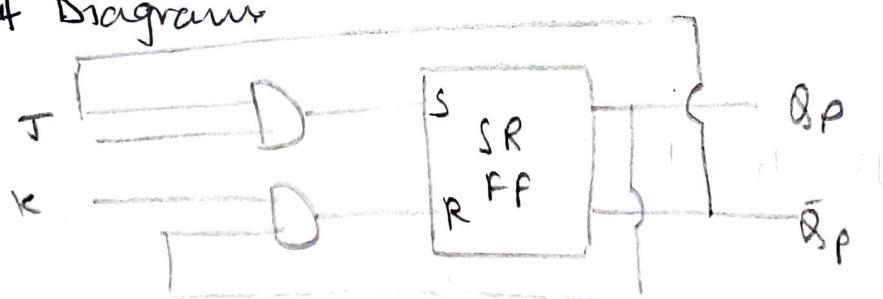
| JK | | QP | |
|----|---|----|----|
| J | K | 00 | 01 |
| 0 | 0 | 0 | X |
| 0 | 1 | 0 | 0 |
| 1 | 0 | X | 0 |
| 1 | 1 | 0 | 1 |

$S = J\bar{Q}_p$

| JK | | QP | |
|----|---|----|----|
| J | K | 00 | 01 |
| 0 | 0 | X | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 |

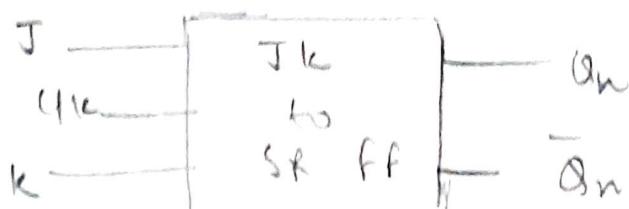
$R = KQ_p$

Circuit Diagram



[JK - to SR flip flop]

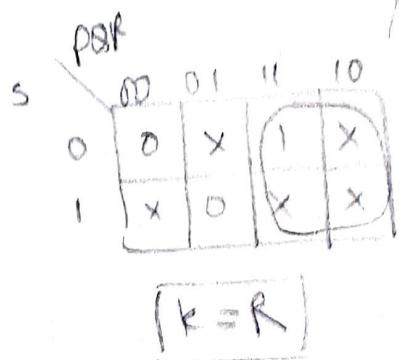
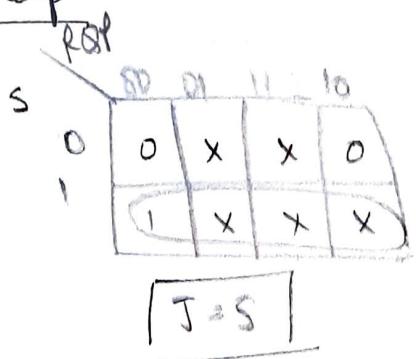
Logic Diagram



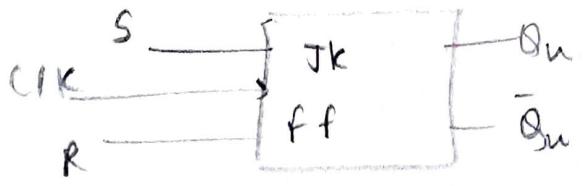
Truth Table

| S | R | Q_p | Q_n | J | K |
|---|---|-------|-------|---|---|
| 0 | 0 | 0 | 0 | 0 | X |
| 0 | 0 | 1 | 1 | X | 0 |
| 0 | 1 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 | X | 1 |
| 1 | 0 | 0 | 1 | 1 | X |
| 1 | 0 | 1 | 1 | X | 0 |
| 1 | 1 | 0 | X | X | X |
| 1 | 1 | 1 | X | X | X |

K-map



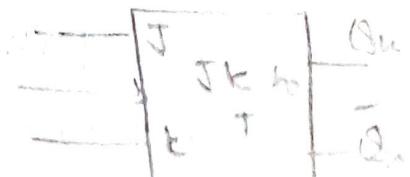
Circuit Diagram



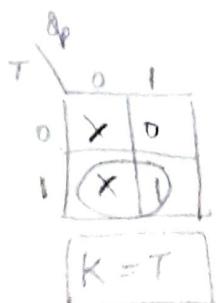
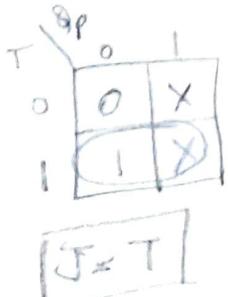
$[JK \text{ to } T]$

Truth Table

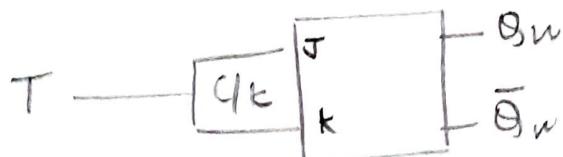
| I | Q_p | Q_n | J | K |
|---|-------|-------|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | X | 0 |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | X | 1 |



K-map



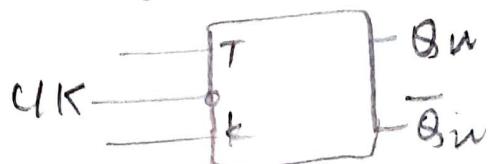
Circuit Diagram



JK to T flip-flop

[JK to D flip-flop.]

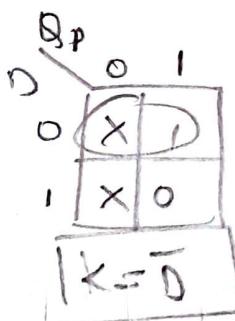
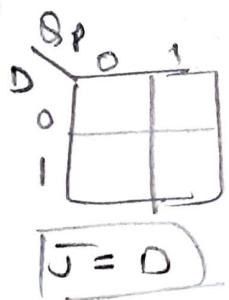
Logic Diagram



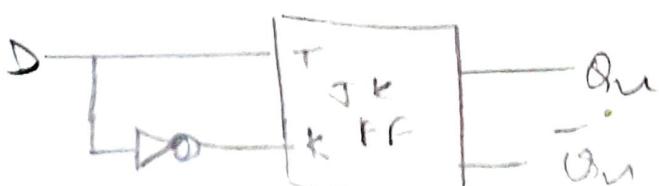
Truth table.

| D | Q_p | Q_n | J | K |
|---|-------|-------|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 1 | 0 | X | 1 |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 1 | X | 0 |

K-map.

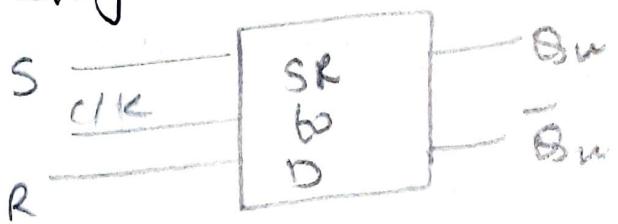


Circuit - Diagram.



SR to D.

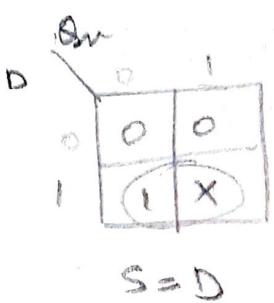
Logic diagram:



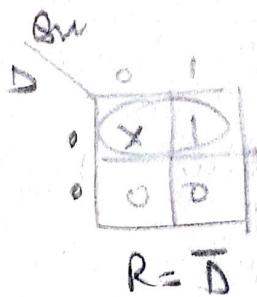
Truth table:

| D | Q_p | Q_u | S | R |
|---|-------|-------|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | X | 0 |

K-map:

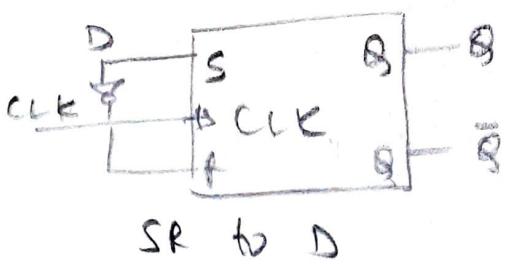


$$S = D$$



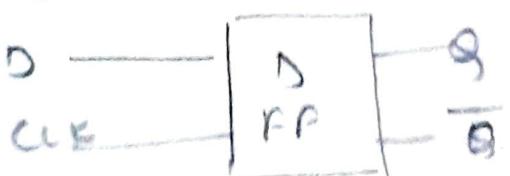
$$R = \bar{D}$$

Circuit diagram:



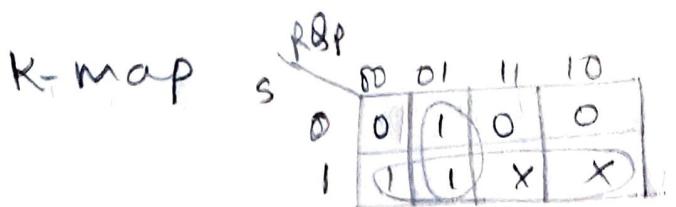
D to SR flip-flop

Logic diagram:

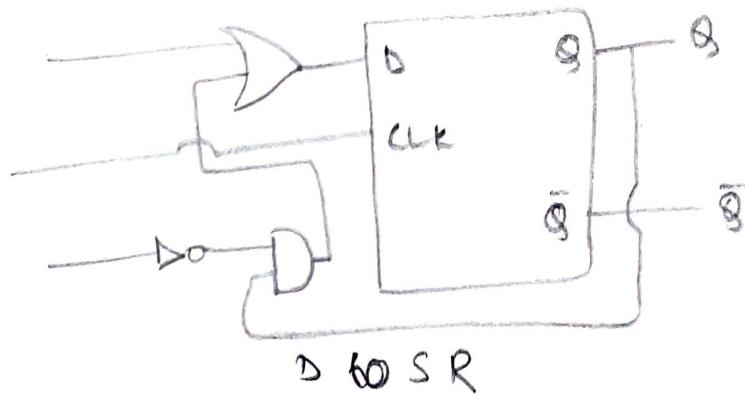


Truth table

| S | R | \bar{Q}_P | \bar{Q}_N | D |
|---|---|-------------|-------------|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | - |
| 1 | 0 | 1 | - | X |
| 1 | 1 | - | Invalid | X |
| 1 | 1 | - | Invalid | X |



$$D = S + \bar{R}\bar{Q}_P$$



D to JK

Logic Diagram



Truth table

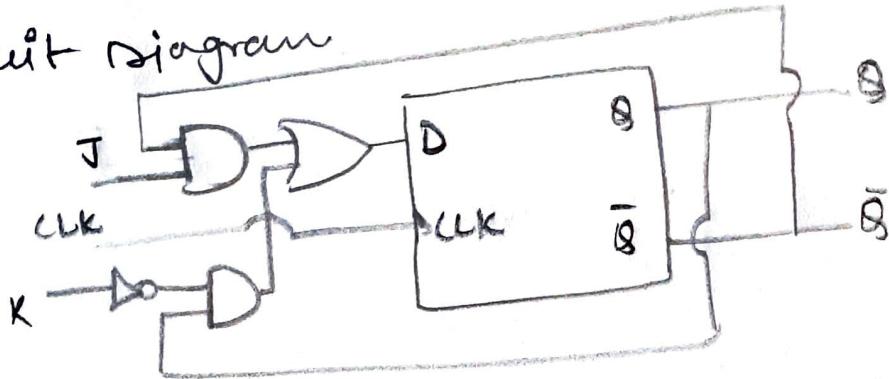
| J | K | Q_P | \bar{Q}_N | D |
|---|---|-------|-------------|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

K-map \bar{Q}_n

| J | 0 | 01 | 11 | 10 |
|---|---|----|----|----|
| 0 | 0 | 1 | | |
| 1 | 1 | 0 | 1 | |
| | | | | |

$$D = \bar{J}\bar{Q}_n + \bar{K}Q_n$$

Circuit diagram



Conclusion

In this lab we learned and implement conversion
 SR to JK, JK to SR, JK to T, JK to D
 SR to D, D to SR and D to JK.