



## Department of Computer Engineering

**CLASS: S.E. COMPUTER**

**SUBJECT: DEL**

**EXPT. NO.: 4**

**DATE:**

**TITLE : Realization of Boolean Expression for suitable combination logic using MUX 74151 / MUX 74153 & DMUX 74154/74138**

**OBJECTIVE :** To learn different techniques of designing multiplexer

**APPARATUS :** Digital-Board, GP-4Patch-Cords, IC 74151, IC 7404, IC 7432 and Required Logic gates if any.

**THEORY :**

### 1. What is multiplexer?

- Multiplexer is a digital switch which allows digital information from several sources to be routed onto a single output line. Basic multiplexer has several data inputs and a single output line.
- The selection of a particular input line is controlled by a set of selection line.
- There are  $2^n$  input lines &  $n$  is the number of selection line whose bit combinations determines which input is selected. It is —Many into One.
- Strobe: - It is used to enable/ disable the logic circuit **OR** 'E' is called as enable I/P which is generally active LOW. It is used for cascading
- MUX is a *single pole* multiple way switch.

### 2. Necessity of multiplexer

- In most of the electronic systems, digital data is available on more than one lines. It is necessary to route this data over a single line.
- It selects one of the many I/P at a time.
- Multiplexer improves the *reliability* of digital system because it reduces the number of external wire connection.



## Department of Computer Engineering

### 3. Enlist significance and advantages of Multiplexer

- It doesn't need K-map & logic simplification.
- The IC package count is minimized.
- It simplifies the logic design.
- In designing the combinational circuit
- It reduces the complexity & cost.
- To minimize number of connections in communication system where we need to handle thousands of connections. Ex. Telephone exchange.

### 4. Applications of MUX

- Data selector to select one out of many data I/P.
- In Data Acquisition system.
- In the D/A converter.

### Multiplexer Tree

- It is nothing but construction of more number of lines using less number of lines.
- It is possible to expand the range of inputs for multiplexers beyond the available range in the integrated circuits. This can be accomplished by interconnecting several multiplexers.

### **8:1 MUX:**

The block diagram of 8:1 MUX & its TT is shown. It has eight data I/P & one enable input, three select lines and one O/P.

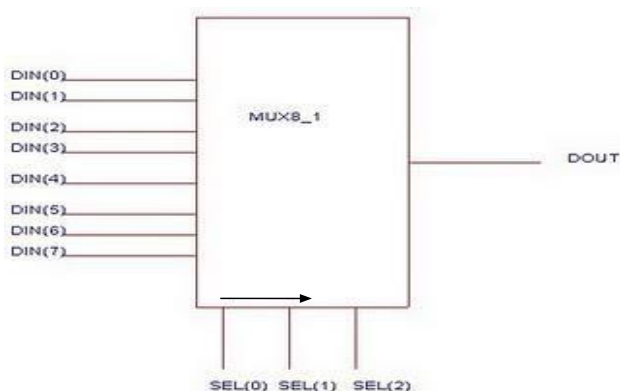
### **Operating principle:**

When the Strobe or Enable input is active low, we can select any one of eight data I/P and connect to O/P.



## Department of Computer Engineering

### IN DIAGRAM:



| SELECTION LINES |   |   | STROBE | OUTPUTS |                |
|-----------------|---|---|--------|---------|----------------|
| C               | B | A | E      | Y       | $\overline{Y}$ |
| X               | X | X | 1      | 0       | 1              |
| 0               | 0 | 0 | 0      | D0      | D0             |
| 0               | 0 | 1 | 0      | D1      | D1             |
| 0               | 1 | 0 | 0      | D2      | D2             |
| 0               | 1 | 1 | 0      | D3      | D3             |
| 1               | 0 | 0 | 0      | D4      | D4             |
| 1               | 0 | 1 | 0      | D5      | D5             |
| 1               | 1 | 0 | 0      | D6      | D6             |
| 1               | 1 | 1 | 0      | D7      | D7             |

**X = don't care condition.**

### Part 1: MUX as a function generator.

Convert the given Boolean expression into standard SOP / POS format if required and complete the logic diagram design accordingly for realization of the same.



## Department of Computer Engineering

### As an example:

Function = Sum of Product (SOP)  $Y = \sum m(1, 3, 4, 5, 6, 7)$

| ELECTION LINES |   |   | STROBE | OUTPUTS |                |
|----------------|---|---|--------|---------|----------------|
| C              | B | A |        | Y       | $\overline{Y}$ |
| 0              | 0 | 0 | 0      | 0       | 1              |
| 0              | 0 | 1 | 0      | 1       | 0              |
| 0              | 1 | 0 | 0      | 1       | 0              |
| 0              | 1 | 1 | 0      | 1       | 0              |
| 1              | 0 | 0 | 0      | 1       | 0              |
| 1              | 0 | 1 | 0      | 1       | 0              |
| 1              | 1 | 0 | 0      | 1       | 0              |
| 1              | 1 | 1 | 0      | 1       | 0              |

### SOP realization Diagram

$$\text{SOP } Y = \sum m(1, 2, 3, 4, 5, 6, 7)$$

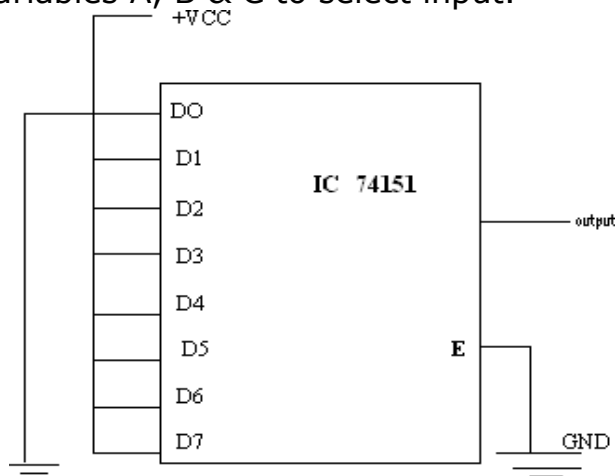
**Solution:** -Since there are 3 variables, the multiplexer have 3 select I/P should be used. Hence one 8:1 mux should be used.

Step 1: -Identify the number decimal corresponding to each minterm.

Here 1,2,3,4,5,6,7

Step 2: -Connect the data input lines 1,2,3,4,5,6,7 to logic 1(+Vcc) & remaining input line 0 to logic 0(GND)

Step 3: -Connect variables A, B & C to select input.





## Department of Computer Engineering

As an Example, Function = Product of Sum (POS)

$$Y = \prod M(0, 5, 6, 7)$$

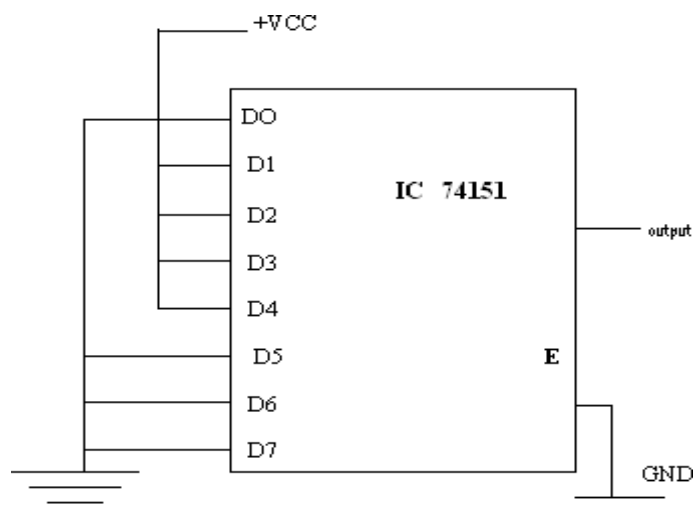
| SELECTION LINES |   |   | STROBE<br>E | OUTPUTS |                |
|-----------------|---|---|-------------|---------|----------------|
| C               | B | A |             | Y       | $\overline{Y}$ |
| 0               | 0 | 0 | 0           | 0       | 1              |
| 0               | 0 | 1 | 0           | 1       | 0              |
| 0               | 1 | 0 | 0           | 1       | 0              |
| 0               | 1 | 1 | 0           | 1       | 0              |
| 1               | 0 | 0 | 0           | 1       | 0              |
| 1               | 0 | 1 | 0           | 0       | 1              |
| 1               | 1 | 0 | 0           | 0       | 1              |
| 1               | 1 | 1 | 0           | 0       | 1              |

POS realization exp:

1. As there are 3 i/p so use 8:1 MUX
2. Connect the given min terms to GND and else decimal numbers to logic1(+VCC).

### POS realization Diagram

$$\text{POS } Y = \prod M(0, 5, 6, 7)$$



### Part -2: Implementation of 16:1 MUX using 8:1 MUX

Use hardware reduction method and implement the given Boolean expression with the help of neat logic diagram. (**N-circle Method**)



## Department of Computer Engineering

**First Method:**  $F(A,B,C,D) = \sum m ( 2, 4, 5, 7, 10, 14 )$

|                | D0 | D1 | D2 | D3 | D4             | D5             | D6 | D7             |
|----------------|----|----|----|----|----------------|----------------|----|----------------|
| $\overline{A}$ | 0  | 1  | 2  | 3  | 4              | 5              | 6  | 7              |
| A              | 8  | 9  | 10 | 11 | 12             | 13             | 14 | 15             |
| Input to Mux   | 0  | 0  | 1  | 0  | $\overline{A}$ | $\overline{A}$ | A  | $\overline{A}$ |

- \*Bold and red marks represent the minterms
- Consider B,C,D as a select line
- Use NOT gate to obtain A and complement of it.

### Solution:-

Step 1:- Apply B, C, D to select I/P & design

table(Implementation table). Step 2: Encircle those min terms which are present in output.

Step 3: If the min terms in a column are not circled then apply logic 0. Step 4: If the min terms in a column are circled then apply logic 1.

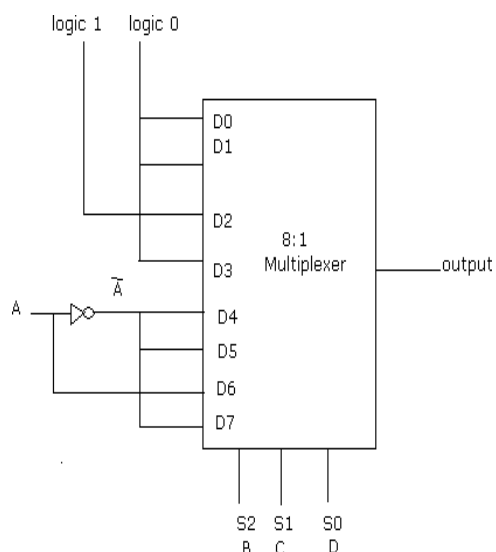
Step 5: If only min term in 2<sup>nd</sup> row is encircled then  $\overline{A}$  should be applied to that data input. Hence apply  $\overline{A}$  to D6.

Step 6: If only min term in 1<sup>st</sup> row is encircled then  $A$  should be applied to that data input. Hence apply  $A$  to D4, D5, D7.



## Department of Computer Engineering

### **16:1 MUX using 8:1 MUX Diagram**



**Second method:**  $F(A, B, C, D) = \sum m(2, 4, 6, 7, 9, 10, 11, 12, 15)$

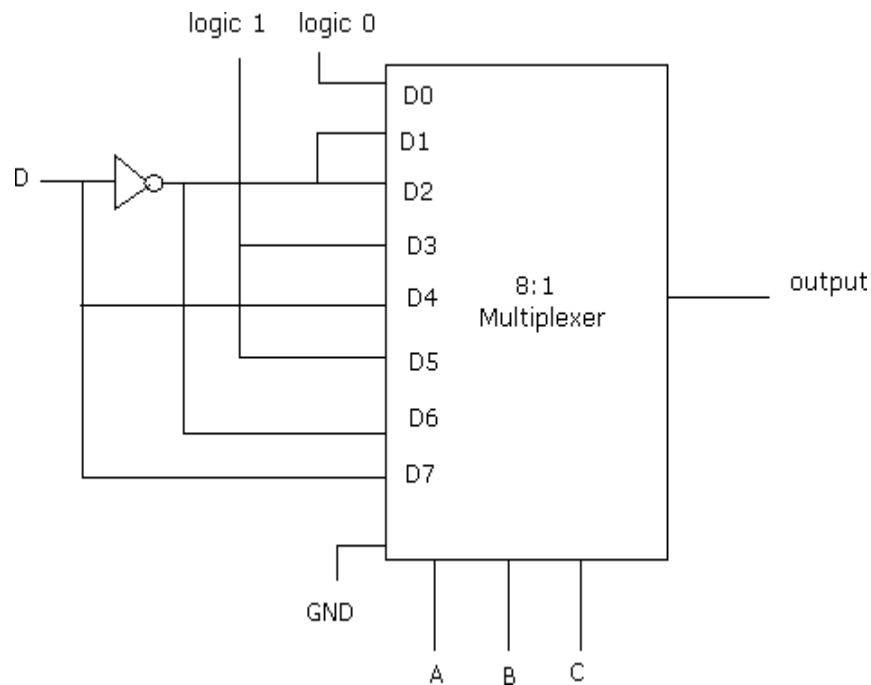
1. Make a combination of pair according to same Values of A, B and C
2. Check the output values with respect to value of D.

| A | B | C | D | output | output |
|---|---|---|---|--------|--------|
| 0 | 0 | 0 | 0 | 0      | 0      |
| 0 | 0 | 0 | 1 | 0      |        |
| 0 | 0 | 1 | 0 | 1      | D'     |
| 0 | 0 | 1 | 1 | 0      |        |
| 0 | 1 | 0 | 0 | 1      | D'     |
| 0 | 1 | 0 | 1 | 0      |        |
| 0 | 1 | 1 | 0 | 1      | 1      |
| 0 | 1 | 1 | 1 | 1      |        |
| 1 | 0 | 0 | 0 | 0      | D      |
| 1 | 0 | 0 | 1 | 1      |        |
| 1 | 0 | 1 | 0 | 1      | 1      |
| 1 | 0 | 1 | 1 | 1      |        |
| 1 | 1 | 0 | 0 | 1      | D'     |
| 1 | 1 | 0 | 1 | 0      |        |
| 1 | 1 | 1 | 0 | 0      | D      |
| 1 | 1 | 1 | 1 | 1      |        |



## Department of Computer Engineering

### **16:1 MUX using 8:1 MUX Diagram**: Second Method



### **Part-3 Implementation of 16:1 MUX using two 8:1 MUX (Cascading Method)**

$$F(A,B,C,D) = \sum m(2, 4, 5, 7, 10, 14)$$

#### **Solution:-**

Step 1: Connect S2, S1, S0 select lines of two 8:1 MUX parallel where as MSB select input is used for enabling MUX.  $\overline{S3}$

Step 2: S3 is connected directly to the enable (E) to mux-2 whereas is connect to enable input of mux-1

Step 3: The output of two MUX is OR to get final output.





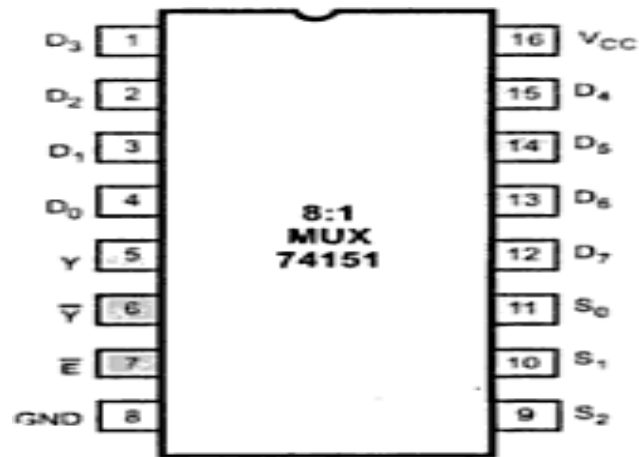
## Department of Computer Engineering

### ***Truth table: - PART\_3***

| Select line    |                |                |                | Output         |                 | Final Output    |
|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|
| S <sub>3</sub> | S <sub>2</sub> | S <sub>1</sub> | S <sub>0</sub> | Y <sub>1</sub> | Y <sub>2</sub>  | Y               |
| 0              | 0              | 0              | 0              | D <sub>0</sub> | --              | D <sub>0</sub>  |
| 0              | 0              | 0              | 1              | D <sub>1</sub> | --              | D <sub>1</sub>  |
| 0              | 0              | 1              | 0              | D <sub>2</sub> | --              | D <sub>2</sub>  |
| 0              | 0              | 1              | 1              | D <sub>3</sub> | --              | D <sub>3</sub>  |
| 0              | 1              | 0              | 0              | D <sub>4</sub> | --              | D <sub>4</sub>  |
| 0              | 1              | 0              | 1              | D <sub>5</sub> | --              | D <sub>5</sub>  |
| 0              | 1              | 1              | 0              | D <sub>6</sub> | --              | D <sub>6</sub>  |
| 1              | 1              | 1              | 1              | D <sub>7</sub> | --              | D <sub>7</sub>  |
| 1              | 0              | 0              | 0              | --             | D <sub>8</sub>  | D <sub>8</sub>  |
| 1              | 0              | 0              | 1              | --             | D <sub>9</sub>  | D <sub>9</sub>  |
| 1              | 0              | 1              | 0              | --             | D <sub>10</sub> | D <sub>10</sub> |
| 1              | 0              | 1              | 1              | --             | D <sub>11</sub> | D <sub>11</sub> |
| 1              | 1              | 0              | 0              | --             | D <sub>12</sub> | D <sub>12</sub> |
| 1              | 1              | 0              | 1              | --             | D <sub>13</sub> | D <sub>13</sub> |
| 1              | 1              | 1              | 0              | --             | D <sub>14</sub> | D <sub>14</sub> |
| 1              | 1              | 1              | 1              | --             | D <sub>15</sub> | D <sub>15</sub> |

***Truth Table for 16:1 MUX using two 8:1 MUX***

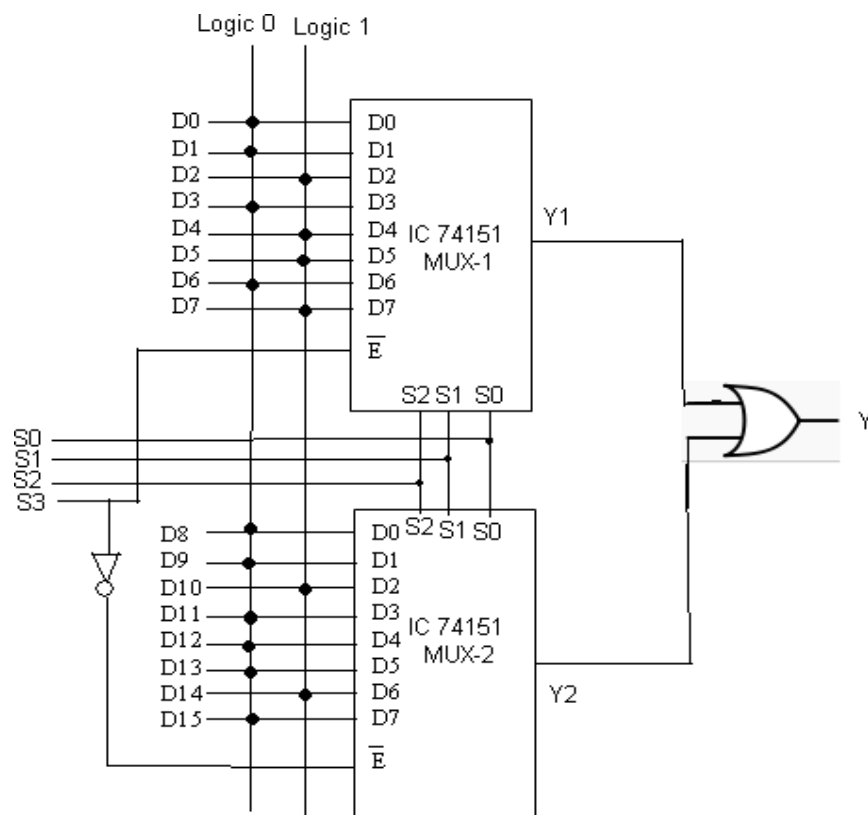
**Department of Computer Engineering**



$\overline{Y}$  : Complemented output  
 $\overline{E}$  : Active low enable input

## Pin Diagram IC74151 8:1 mux

**Multiplexer Tree according to given equation: -**





## Department of Computer Engineering

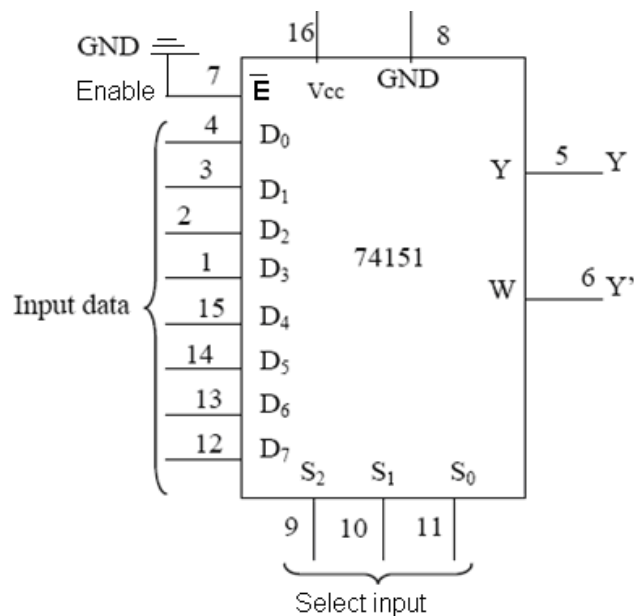
### **Outcome:**

Multiplexer is used as a data selector to select one out of many data inputs. It is used for simplification of logic design. It is used to design combinational circuit. Use of multiplexer minimizes no. of connections.

### **FAQ:**

#### ***Q1. Enlist applications of MUX***

1. MUX is used as data selector.
2. It is used to design combinational circuit.
3. Less number of wires required which reduces complexity
4. There is no need to design k-map
5. We design equation using truth table.





## Department of Computer Engineering

### **2. Define the terms Encoder and Decoder**

→ Encoders are used to encode given digital number into different numbering format. like decimal to BCD Encoder, Octal to Binary.

→ Decoders are used to decode a coded binary word like BCD to seven segment decoders. Thus, encoder and decoder are application specific logic develop, we cannot use any type of input for any encoder and decoder.

### **REFERENCE:**

- 1. R.P. Jain "Modern Digital Electronics" TMH 4<sup>th</sup> Edition**
- 2. D.Leach, Malvino, Saha," Digital Principles and Applications", TMH**