**KIIT, Deemed to be University**
**School of Electronics Engineering**
**Digital System Design Laboratory [EC 29005]**

# EXPERIMENT - 4

## Aim:

Design and Simulation of 8-to-1-line Multiplexer using Verilog HDL.
Generation of 4 variable logic function using 8-to-1-line Multiplexer.

## Component/Software Used:

| Component/Software | Specification |
|---|---|
| ICs | 74151, 7404 |
| Bread Board, Power supply, LEDs, Resistors,Switches, Connecting wires | As per requirement |
| Software(s) Used | Vivado 2016.1 |

## Theory:

### Multiplexer:

Multiplexer (MUX): Multiplexer is a combinational circuit that has maximum of $(2)^n$ data inputs, '**n**'select input lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines. Since there are '**n**'selection lines, there will be $2^n$ possible combinations of zeros and ones. So, each combination will select only one data input. The multiplexer acts like a digitally controlled multi-position switch. A multiplexer is also called a **data selector** , since it selects one of many inputs and steers the binary information to the output line.

In 8-to-1-line **MUX (8X1 MUX)** there are eight**(8)** input lines and three**(3)** select lines whose bit combinations determine which input is selected. The block diagram and the function table is shown in Figure 4.1 and Table 4.1 respectively. In the Figure 4.1, $\mathbf{I_0}$ to $\mathbf{I_7}$ are eight inputs, $\mathbf{S_0}$ to $\mathbf{S_2}$ are the select input and **F** is the output of the **8X1 MUX**. The logic diagram is shown in Figure 4.2.

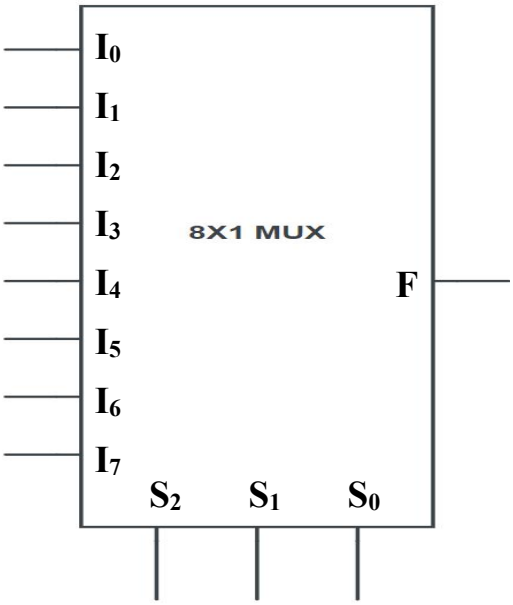


**Figure 4.1: Block diagram of 8X1 MUX**

| Select Inputs | | | Output |
|---|---|---|---|
| $S_2$ | $S_1$ | $S_0$ | F |
| 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 | $I_7$ |

**Table 4.1: Function table of 8X1 MUX**

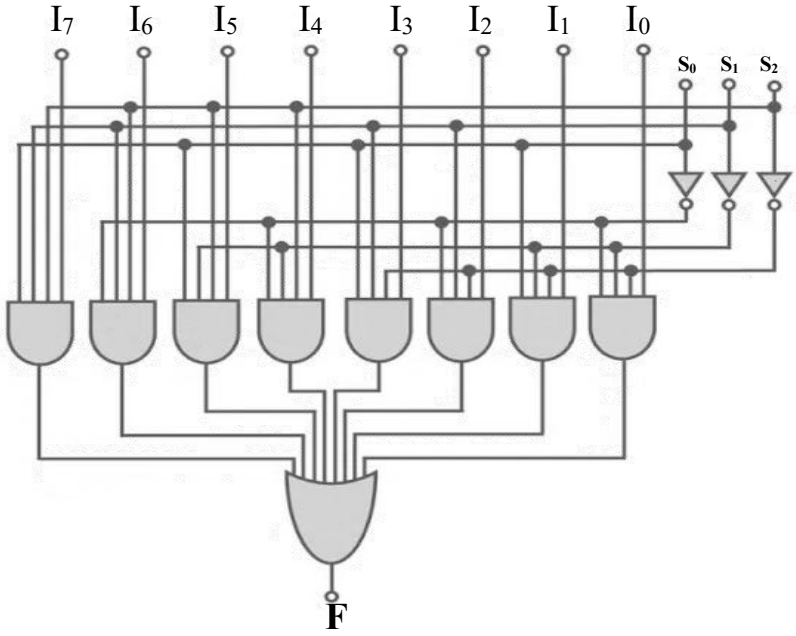


**Figure 4.2: Logic diagram of 8X1 MUX**

Digital multiplexer is available in IC 74151. IC 74151 is an 8-to-1 line (8X1 MUX) that has active high 8-data inputs and 3-selection input lines and two outputs, one active low and other active high output. The enable input ($\overline{E}$) is active low in the MUX, must be assigned logic 0 (LOW) for its normal operation. Figure 4.3 shows the pin diagram IC-74151.

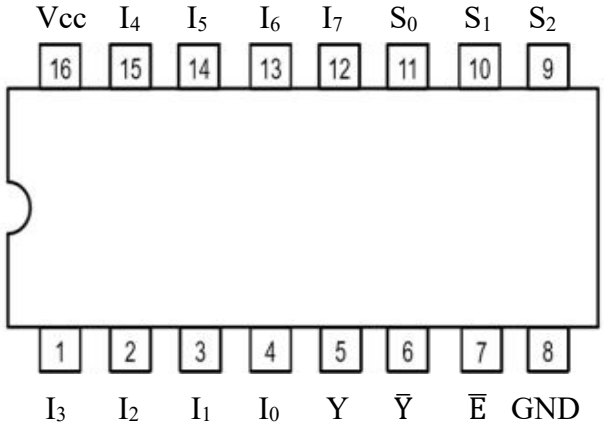


**Figure 4.3: Pin diagram IC-74151**

| PIN NO | SYMBOL | FUNCTION |
|---|---|---|
| 4, 3, 2, 1, 15, 14, 13, 12 | $I_0$ to $I_7$ | multiplexer inputs |
| 5 | Y | multiplexer output |
| 6 | $\overline{Y}$ | complementary multiplexer output |
| 7 | $\overline{E}$ | enable input (active LOW) |
| 8 | GND | ground (0 V) |
| 11, 10, 9 | S0, S1, S2 | select inputs |
| 16 | Vcc | positive supply voltage |

**Table 4.2: Pin Description of IC-74151**

| Inputs | | | | | | | | | | | | Outputs | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Enable | Select | | | Data | | | | | | | | | |
| $\overline{E}$ | $S_2$ | $S_1$ | $S_0$ | $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | Z | $\overline{Z}$ |
| 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | x | x | x | x | x | x | x | 1 | 0 |
| 0 | 0 | 0 | 1 | x | 0 | x | x | x | x | x | x | 0 | 1 |
| 0 | 0 | 0 | 1 | x | 1 | x | x | x | x | x | x | 1 | 0 |
| 0 | 0 | 1 | 0 | x | x | 0 | x | x | x | x | x | 0 | 1 |
| 0 | 0 | 1 | 0 | x | x | 1 | x | x | x | x | x | 1 | 0 |
| 0 | 0 | 1 | 1 | x | x | x | 0 | x | x | x | x | 0 | 1 |
| 0 | 0 | 1 | 1 | x | x | x | 1 | x | x | x | x | 1 | 0 |
| 0 | 1 | 0 | 0 | x | x | x | x | 0 | x | x | x | 0 | 1 |
| 0 | 1 | 0 | 0 | x | x | x | x | 1 | x | x | x | 1 | 0 |
| 0 | 1 | 0 | 1 | x | x | x | x | x | 0 | x | x | 0 | 1 |
| 0 | 1 | 0 | 1 | x | x | x | x | x | 1 | x | x | 1 | 0 |
| 0 | 1 | 1 | 0 | x | x | x | x | x | x | 0 | x | 0 | 1 |
| 0 | 1 | 1 | 0 | x | x | x | x | x | x | 1 | x | 1 | 0 |
| 0 | 1 | 1 | 1 | x | x | x | x | x | x | x | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | x | x | x | x | x | x | x | 1 | 1 | 0 |
| 1 | x | x | x | x | x | x | x | x | x | x | x | 0 | 1 |

**Table 4.3: Function table of IC-74151**

## Boolean function implementation:

As the name indicates, MUX is many into one, which means that this circuit is associated with many inputs and a single output. Here, the minterms of a function are generated by the circuit associated with the selection inputs. MUX has n-selection inputs and $2^n$- data inputs, one for each minterm.

For implementing a Boolean function of n-variables with a MUX that has (n - 1)selection inputs. The first (n-1) variables of the function are connected to the selection inputs and the remaining single variable are used for the data inputs. For implementing any Boolean function of n-variables with a MUX with (n-1) selection inputs and ($2^{n-1}$) data inputs, the following steps are used:

I. Firstly, the Boolean function is listed in a truth table.

II. The first (n-1) variables in the table are applied to the selection inputs of the MUX.

III. For each combination of selection variables, evaluate the output as a function of the last variable.This function can be 0, 1, the variable, or the complement of the variable.

IV. These values are then applied to the data inputs in the proper order.

Boolean function $F(A, B, C, D) = \sum(1, 3, 4, 11, 12, 13, 14, 15)$ implementation is shown in Figure 4.4 using IC 74151. The first variable A is connected to selection input $S_2$ so that A , B, and C correspond to selection inputs $S_2$, $S_1$, and $S_0$,respectively.

The values for the data inputs are determined from the truth table listed in the Table 4.4.

| | $I_0$ $\overline{A}\overline{B}\overline{C}$ | $I_1$ $\overline{A}\overline{B}C$ | $I_2$ $\overline{A}B\overline{C}$ | $I_3$ $\overline{A}BC$ | $I_4$ $A\overline{B}\overline{C}$ | $I_5$ $A\overline{B}C$ | $I_6$ $AB\overline{C}$ | $I_7$ $ABC$ |
|---|---|---|---|---|---|---|---|---|
| $\overline{D}$ | 0 | 2 | (4) | 6 | 8 | 10 | (12) | (14) |
| $D$ | (1) | (3) | 5 | 7 | 9 | (11) | (13) | (15) |
| 8x1 MUX data input | D | D | $\overline{D}$ | 0 | 0 | D | 1 | 1 |

Table 4.4: Truth table for implementing Boolean function with MUX

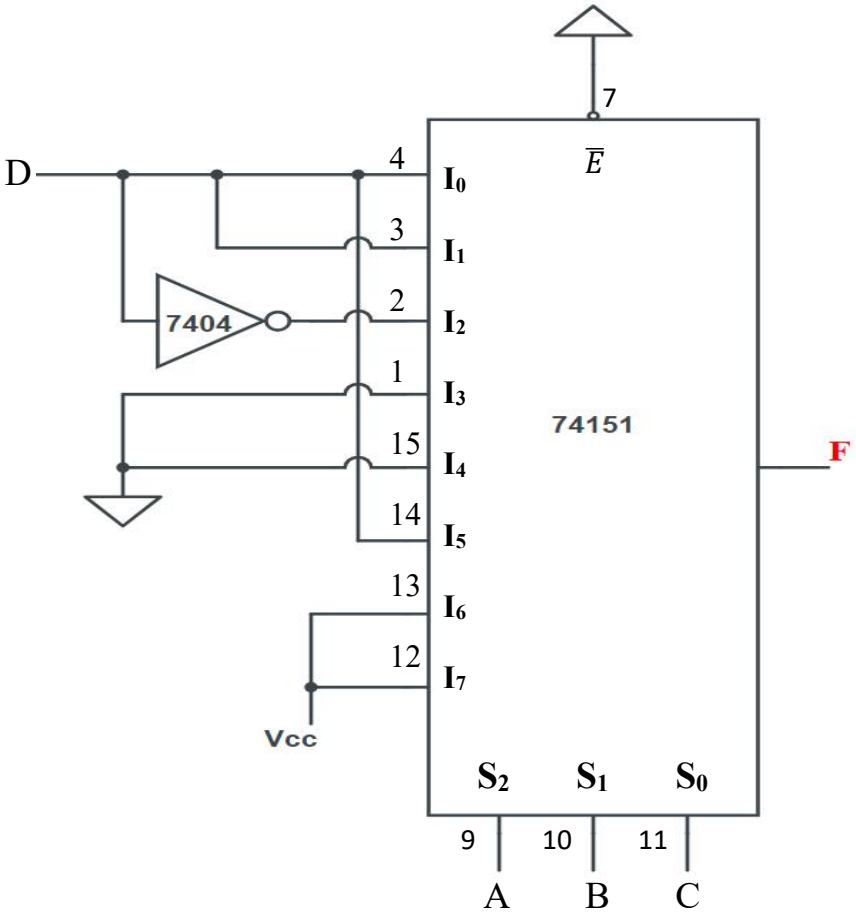


**Figure 4.4: Implemention of Boolean function with a multiplexer**

## Procedure

### For Software Simulation:

a) Create a module with required number of variables and mention it's input/output.
b) Write the description of given Boolean function using operators or by using the built in primitive gates.
c) Synthesize to create RTL Schematic.
d) Create another module referred as test bench to verify the functionality and to obtain the waveforms of input and output.
e) Follow the steps required to simulate the design and compare the obtained output with the corresponding truth table.
f) Take the screenshots of the RTL schematic and simulated waveforms.

**Note:** Students need to write the Verilog HDL code by their own for which they can refer Appendix - A if required.

### For Hardware implementation:

a) Turn off the power of the Trainer Kit before constructing any circuit.
b) Connect **power supply (+ 5 V DC)** pin and **ground** pin to the respective pins of the trainer kit.
c) Place the ICs properly on the bread board in the Trainer Kit.
d) Connect VCC and GND pins of each chip to the power supply and ground bus strips on the bread board.
e) Connect the input and output pins of chips to the input switches and output LEDs respectively in the Trainer Kit.
f) Check the connections before you turn on the power.
g) Apply various combinations of inputs according to truth tables and observe outputs of LEDs.

## Observation:

To be written by students

**Design Problem :**

Design and Simulation of 4X1 multiplexer by 2X1 multiplexers using Verilog HDL. Hardware implementation of 4X1 multiplexer using 2X1 multiplexers only.

**Solution:**

4x1 Multiplexer has four data inputs $I_3$, $I_2$, $I_1$ & $I_0$, two selection lines $S_1$ & $S_0$ and one output F. The block diagram of 4x1 Multiplexer is shown in the Figure 4.5 below.

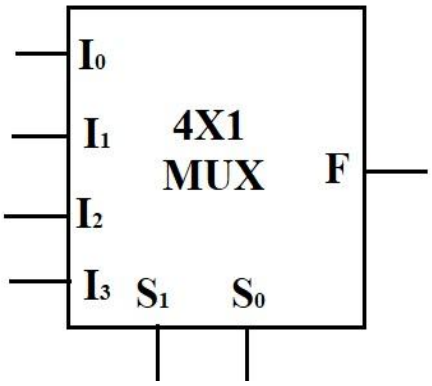


**Figure 4.5 : Block diagram of 4 X 1 MUX**

| Select inputs | | Output |
|---|---|---|
| $S_1$ | $S_0$ | F |
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |

**Table 4.5: Function table of 4 X 1 MUX**

$$F = (\overline{S_1}\,\overline{S_0}\ I_0) + (\overline{S_1}\,S_0\,I_1) + (\overline{S_0}\,S_1\,I_2) + (S_1 S_0\,I_3)$$

**Boolean Expression of 4X1 MUX**

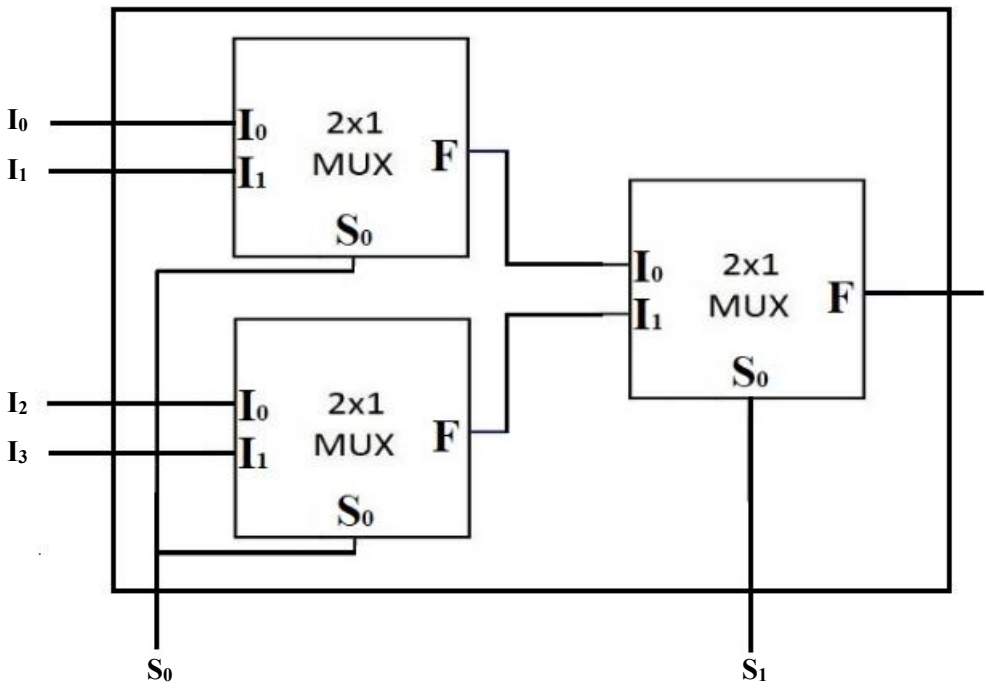


**Figure 4.6: Logic diagram of 4 X 1 MUX using 2 X 1 MUXs**

The hardware implementation of the 4 x1 MUX using 2x1 MUXs is done with the Quad 2x1 MUX IC 74157.

## Quad 2x1 MUX IC 74157:

It has four similar multiplexers inside it, and hence it is called Quad Package 2-Input Multiplexer. Each has two input pins (for the first MUX: input 1A and 1B) and one output pin (for the first MUX: output 1Y), which forms a 2X1 Multiplexer. It selects four bits of data from two sources under the control of common data select input when the enable input is active low.

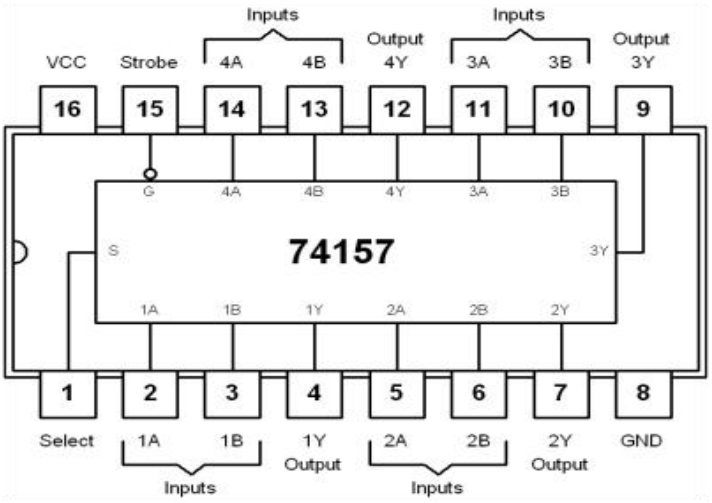


**Figure 4.7: Pin diagram of quad 2X1 MUX IC 74157**

## Conclusion:

To be written by students.

## Sample viva-voice questions

1. Explain the principle of multiplexer.
2. Draw a circuit diagram of 4× 1 multiplexer.
3. What are the advantages of multiplexer?
4. What are the disadvantages of multiplexer?
5. Make the truth table of 4× 1 multiplexer.
6. Define de-multiplexer.
7. How can a decoder be used as a de-multiplexer?
8. What is multiplexer tree?
9. What is the application of de-multiplexer?
10. What is the difference between multiplexer and de-multiplexer?