

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.

COMPONENTS / SOFTWARE USED

Components / Software

- ICs
- Bread board, power supply, LED's, resistors, switches, connecting wires
- Software(s) Used

Specification

74151, 7404

As per requirement

Vivado 2016.1

THEORY :

Multiplexer (MUX) 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 conducted to the output based on the values of sequential lines. Since there are 'n' selection lines there will be 2^n possible combinations of zeroes and ones. So each combination will select only one data input. The multiplexer acts like a digitally controlled multistate 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

Teacher's Signature : _____

Block Diagram

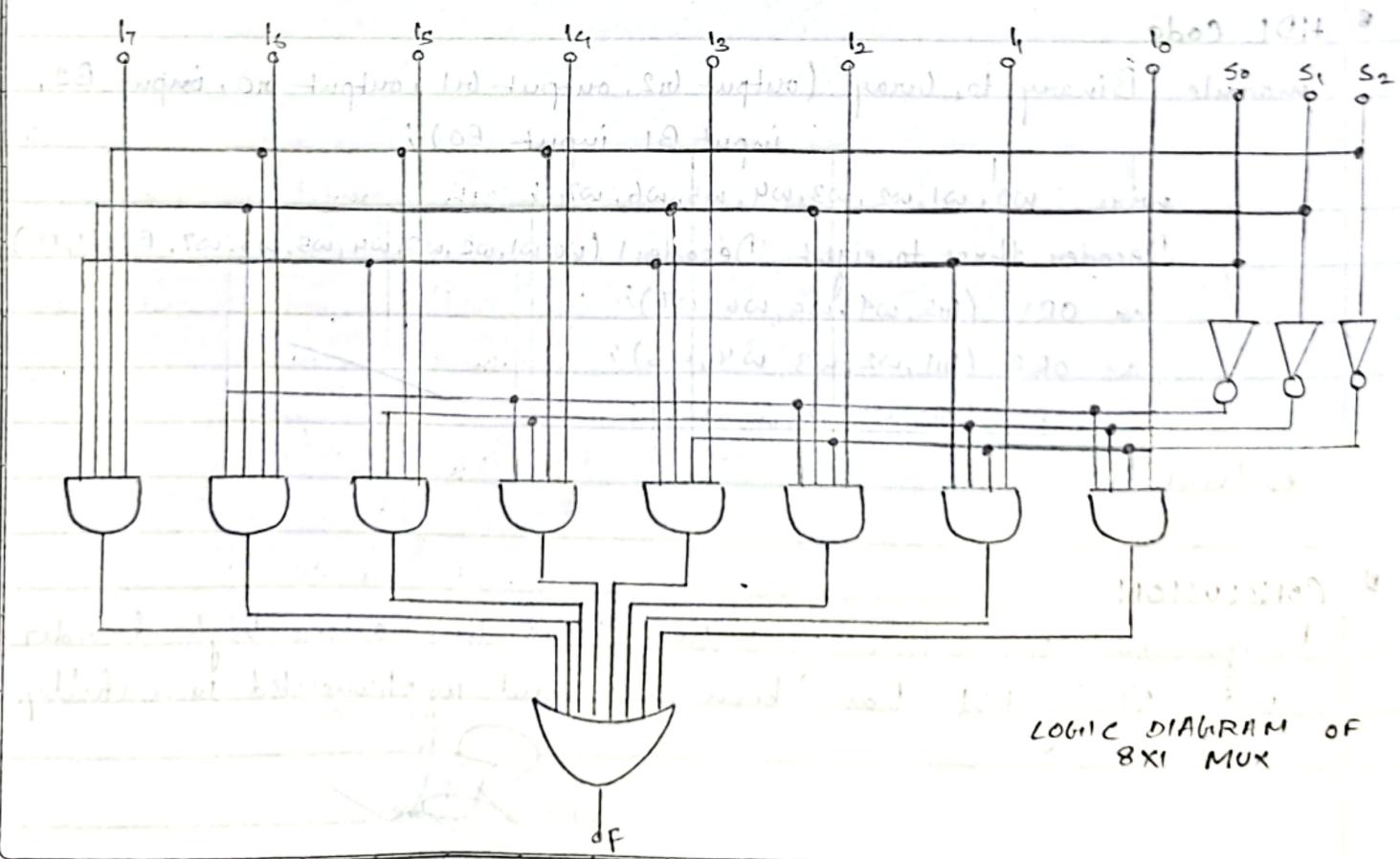
BLOCK DIAGRAM OF 8X1 MUX			SELECTING INPUTS	OUTPUT		
			S_2	S_1	S_0	F
10	11	12	0	0	0	I_0
13	14	15	1	0	0	I_1
16	17	18	1	1	0	I_2
19	20	21	0	1	0	I_3
22	23	24	0	0	1	I_4
25	26	27	1	1	1	I_5
28	29	30	0	1	0	I_6
31	32	33	1	1	1	I_7

FUNCTION TABLE OF 8X1 MUX

$$(I_0, I_1, I_2, I_3) \bar{S} = (0, 0, 1, 0) \text{ m}$$

$$(I_4, I_5, I_6, I_7) \bar{S} = (0, 0, 0, 1) \text{ m}$$

$$(I_0, I_1, I_2, I_3) S = (0, 0, 0, 1) \text{ m}$$



I_0 to I_7 are eight inputs. So S_2 are the select input and F is the output of the 8X1 MUX. The logic diagram is shown in figure 4.2.

Digital Multiplexer is available in IC74151. IC74151 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 (\bar{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.

Boolean function implementation

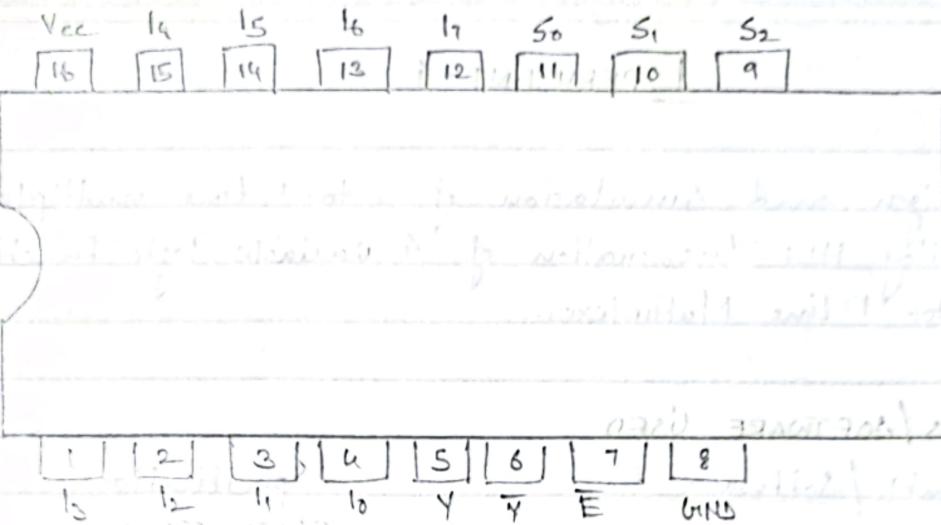
As the name indicates, MUX is many into one, which means that this circuit is associated with many inputs and a single output. Here, the minterms of a function are generated by the circuit associated with the selection inputs. MUX has n selection inputs and 2^n data inputs, one for each minterm.

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.

- ① Firstly, the boolean function is listed in a truth table.
- ② The first $(n-1)$ variables in the table are applied to the selection inputs of the MUX.

IC-74151

OS 0950 AM



PIN DIAGRAM OF IC-74151

PIN NO	SYMBOL	FUNCTION
4,3,2,1,15, 14,13,12	I ₆ to I ₁	multiplexer inputs
5	Y	multiplexer output
6	\bar{Y}	complementary multiplexer output
7	E	enable input (active LOW)
8	GND	ground (0V)
11,10,9	S ₀ , S ₁ , S ₂	select inputs
16	V _{cc}	positive supply voltage

PIN DESCRIPTION OF IC-74151

- (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 table 4.4.

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}B\bar{C}$	$\bar{A}BC$	$A\bar{B}\bar{C}$	$A\bar{B}C$	$AB\bar{C}$	ABC
D	0	2	4	6	8	10	12	14
D	1	3	5	7	9	11	13	15
8x1 MUX Data Input	0	0	0	0	0	0	1	1

Table 4.4: Truth table for implementing boolean function with MUX

PROCEDURE

- For software implementation
 - Create a module with required number of variables and mention its input/output.
 - Write the description of the given boolean function using operators or by using the built-in primitive gates.
 - Synthesize to create RTL Schematic

Enable	INPUTS												OUTPUTS	
	Select			Data										
\bar{E}	S_2	S_1	S_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Z	\bar{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	0	X	X	X	X	X	0	1	
0	0	1	1	X	X	1	X	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	0	X	X	X	0	1	
0	1	0	1	X	X	X	X	1	X	X	X	1	0	
0	1	1	0	X	X	X	X	X	0	X	X	0	1	
0	1	1	0	X	X	X	X	X	1	X	X	1	0	
0	1	1	1	X	X	X	X	X	X	0	X	0	1	
0	1	1	1	X	X	X	X	X	X	1	X	1	0	
Φ	X	X	X	X	X	X	X	X	X	X	X	0	1	

FUNCTION TABLE OF IC-74151

- (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 stimulate the design and compare the obtained output with the corresponding truth table.
- (f) Take the screenshots of the RTL schematic and simulated waveforms.

- For hardware implementation:

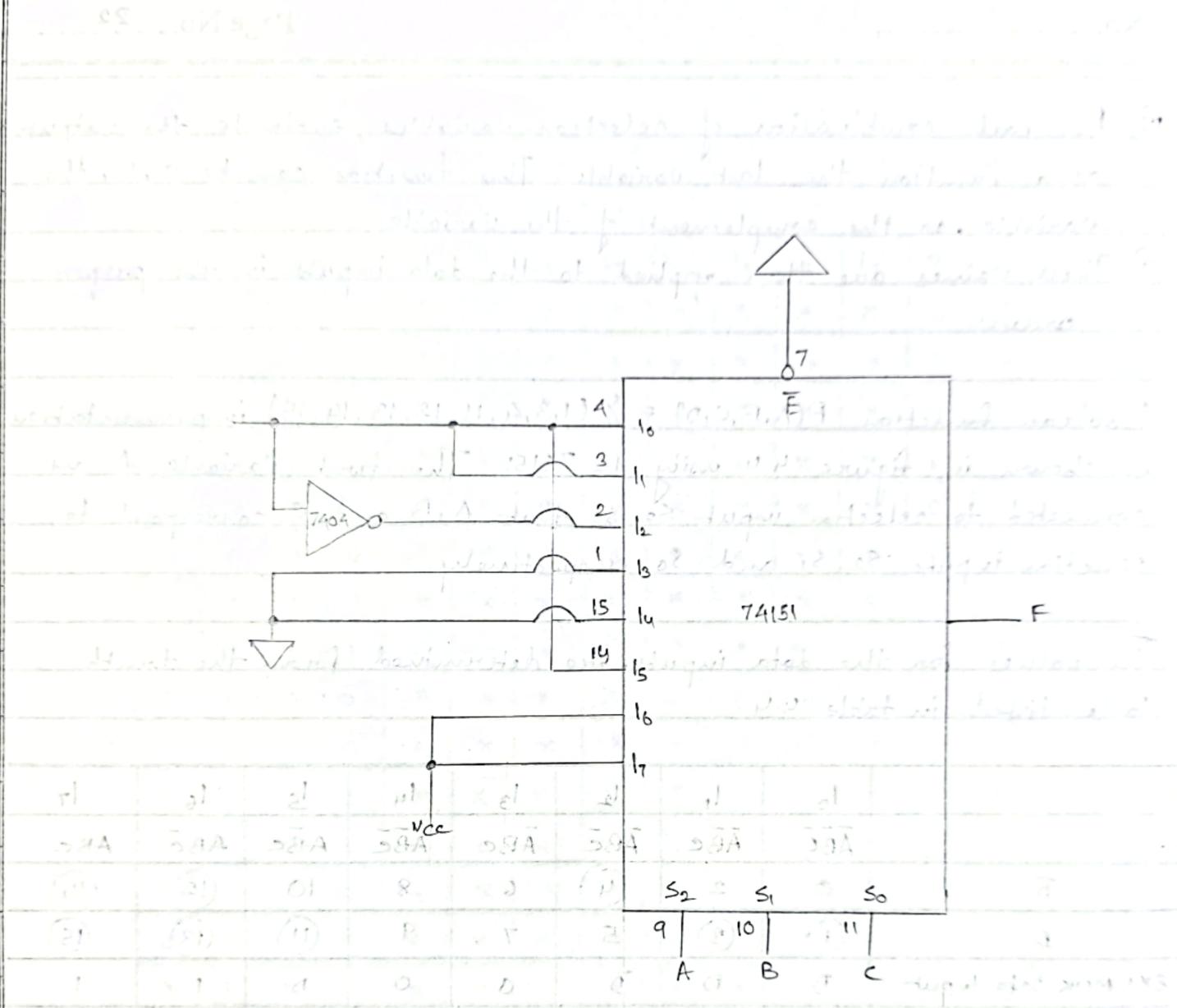
- Turn off the power of the trainer kit before constructing any circuit.
- Connect power supply (+5V DC) pin and ground pin to the respective pins of the trainer kit.
- Place the IC's properly on the bread board in the trainer kit.
- Connect VCC and GND pins of each chip to the power supply and ground bus strips on the bread board.
- Connect the input and output pins of chips to the input switches and output LEDs respectively in the trainer kit.
- Check the connections before you turn on the power.
- Apply various combinations of inputs according to truth tables and observe outputs of LED's.

I HDL Code

```

module MUX_eight_to_one (output F, input S2, input S1, input S0,
                        input I7, input I6, input I5, input I4, input I3, input I2,
                        input I1, input I0);
    wire W0,W1,W2,W3,W4,W5,W6,W7,W8,W9,W10;
    not NOT1 (W0,S0);
    not NOT2 (W1,S1);
    not NOT3 (W2,S2);

```



IMPLEMENTATION OF BOOLEAN FUNCTION WITH A MULTIPLEXER

Implementation of Boolean function using multiplexer

Boolean function:

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

Implementation using 74151 Multiplexer:

Inputs:

- A
- B
- C

Outputs:

- F

Address pins:

- A
- B
- C

Data pins:

- $D_0 = \overline{A}B$
- $D_1 = \overline{A}B\overline{C}$
- $D_2 = A\overline{B}\overline{C}$
- $D_3 = 0$

Enable pin:

- $E = 1$

Vcc and GND connections:

```

and AND1 (w3, 10, w2, w1, w0);
and AND2 (w4, 11, w2, w1, w0);
and AND3 (w5, 12, w2, s1, w0);
and AND4 (w6, 13, w2, s1, w0);
and AND5 (w7, 14, s2, w1, w0);
and AND6 (w8, 15, s2, w1, w0);
and AND7 (w9, 16, s2, s1, w0);
and AND8 (w10, 17, s2, s1, w0);
or OR1 (F, w3, w4, w5, w6, w7, w8, w9, w10);
end module

```

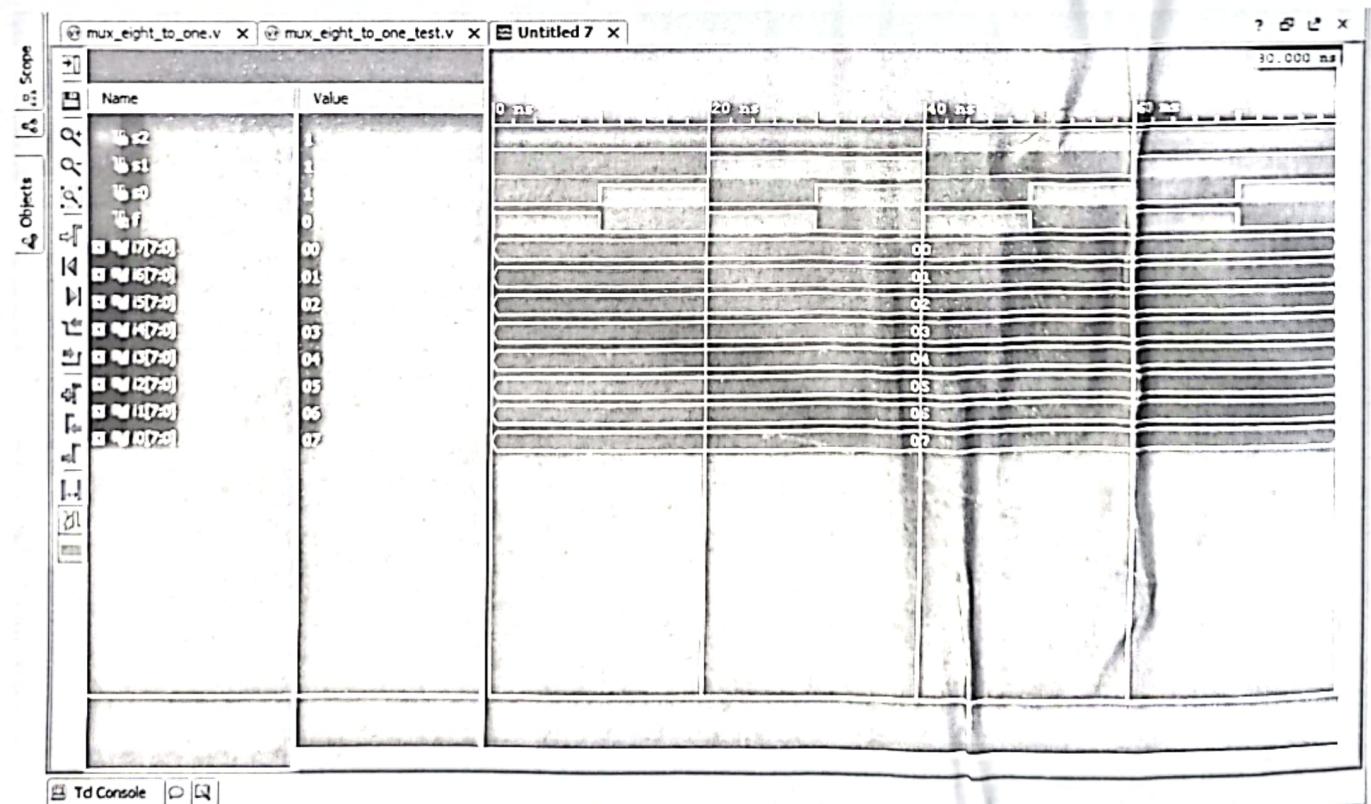
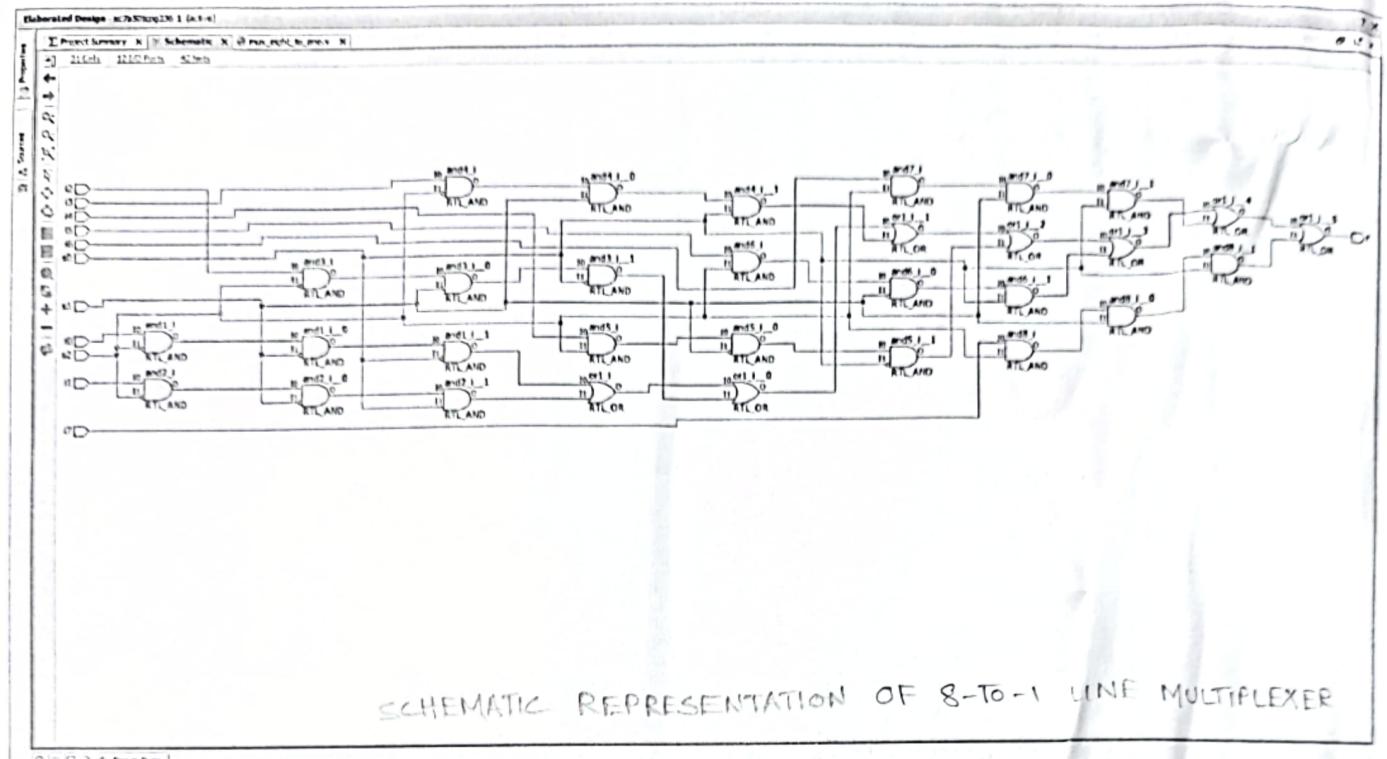
TEST BENCH CODE

```

module Check_MUX_eight_to_one();
reg s0, s1, s2;
wire F;
MUX_eight_to_one ckt(F, s2, s1, s0, 17, 16, 15, 14, 13, 12, 11, 10);
initial
begin
    s0 = 0; s1 = 0; s2 = 0;
#10 s0 = 1; s1 = 0; s2 = 0;
#10 s0 = 0; s1 = 1; s2 = 0; *
#10 s0 = 1; s1 = 1; s2 = 0; s2 = 0; s1 = 0; s0 = 0;
#10 s0 = 0; s1 = 0; s2 = 1; i7 = 8'b00000000;
#10 s0 = 1; s1 = 0; s2 = 1; i6 = 8'b00000001;
#10 s0 = 0; s1 = 1; s2 = 1; i5 = 8'b00000010;
#10 s0 = 1; s1 = 1; s2 = 1; i4 = 8'b00000011;
#10 s0 = 1; s1 = 1; s2 = 1; i3 = 8'b000000100;
#10 s0 = 1; s1 = 1; s2 = 1; i2 = 8'b000001101;
#finish; i1 = 8'b000000110;
end
endmodule

```

Teacher's Signature : _____



8-TO-1 LINE MULTIPLEXER WAVEFORM

DESIGN PROBLEM

4X1 MUX using the 2X1 MUX

```
module two_to_one_mux (output F, input S0, input I1, input I0);
    wire W0, W1, W2;
    nand NAND1 (W0, S0, S0);
    nand NAND2 (W1, W0, I0);
    nand NAND3 (W2, S0, I1);
    nand NAND4 (F, W1, W2);
```

Simulation of 4X1 MUX using 2X1 MUX

```
module four_to_one_mux (output F, input S1, input S0, input I3, input I2,
                        input I1, input I0);
    wire W1, W2;
    two_to_one_mux Mux1 (W1, S0, I1, I0);
    two_to_one_mux Mux2 (W2, S0, I1, I0);
    two_to_one_mux Mux3 (F, S1, W2, W1);
endmodule
```

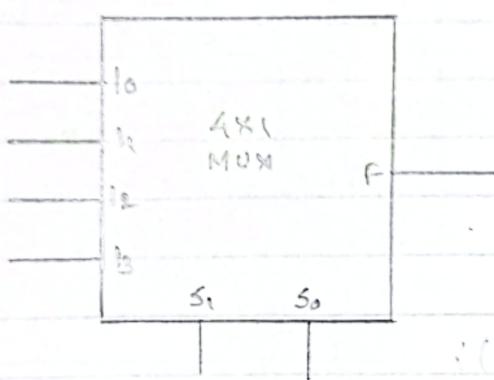
CONCLUSION

Design and simulation of 8-to-1 line multiplexer using Verilog HDL has been done and implemented successfully.

Solution:

~~4X1 Multiplexer~~ has four data inputs I_3, I_2, I_1 , and I_0 , two selection lines S_1 & S_0 and one output F . The block diagram of 4X1 Multiplexer is shown in the figure alongside

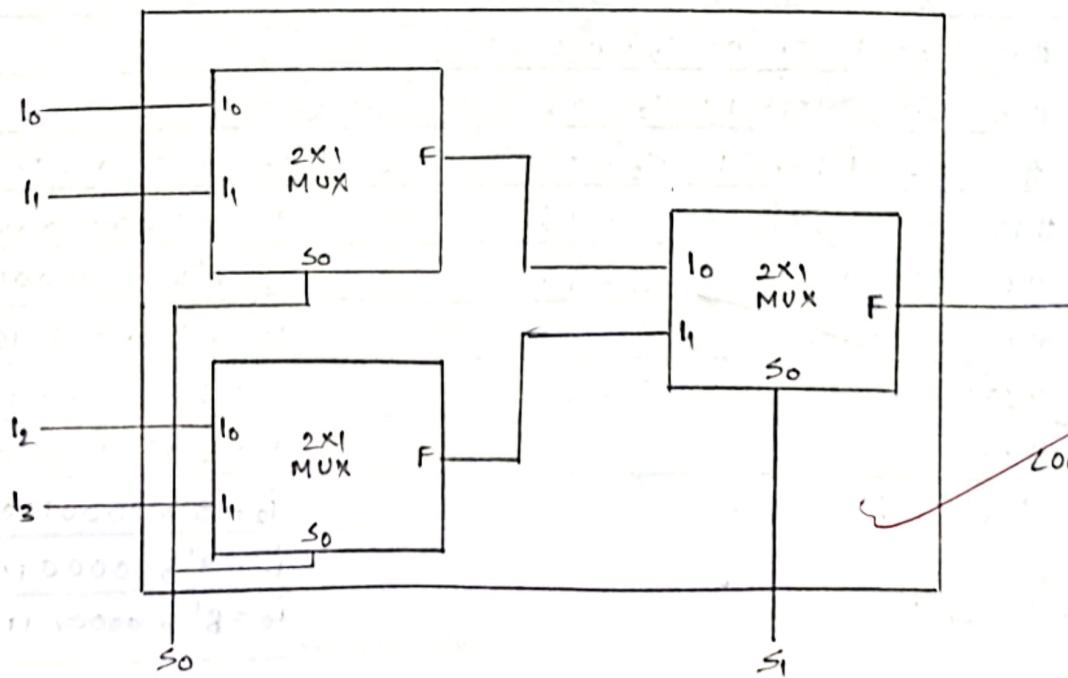
$$F = (\bar{S}_1 \bar{S}_0 I_0) + (\bar{S}_1 S_0 I_1) + (S_0 \bar{S}_1 I_2) + (S_1 S_0 I_3)$$



BLOCK DIAGRAM OF 4X1 MUX

Select Inputs		Output F
S1	S0	I0
0	0	I0
0	1	I1
1	0	I2
1	1	I3

FUNCTION TABLE OF 4X1 MUX



LOGIC DIAGRAM OF
4X1 MUX USING
2X1 MUXES

The hardware implementation of the 1x1 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 IA and IB) and one output pin (for the first MUX: output YY), 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.

Conclusion

Design and simulation of 8-to-1 line multiplexer using Verilog HDL has been done and implemented successfully.

Teacher's Signature : _____

PP
14/9/23

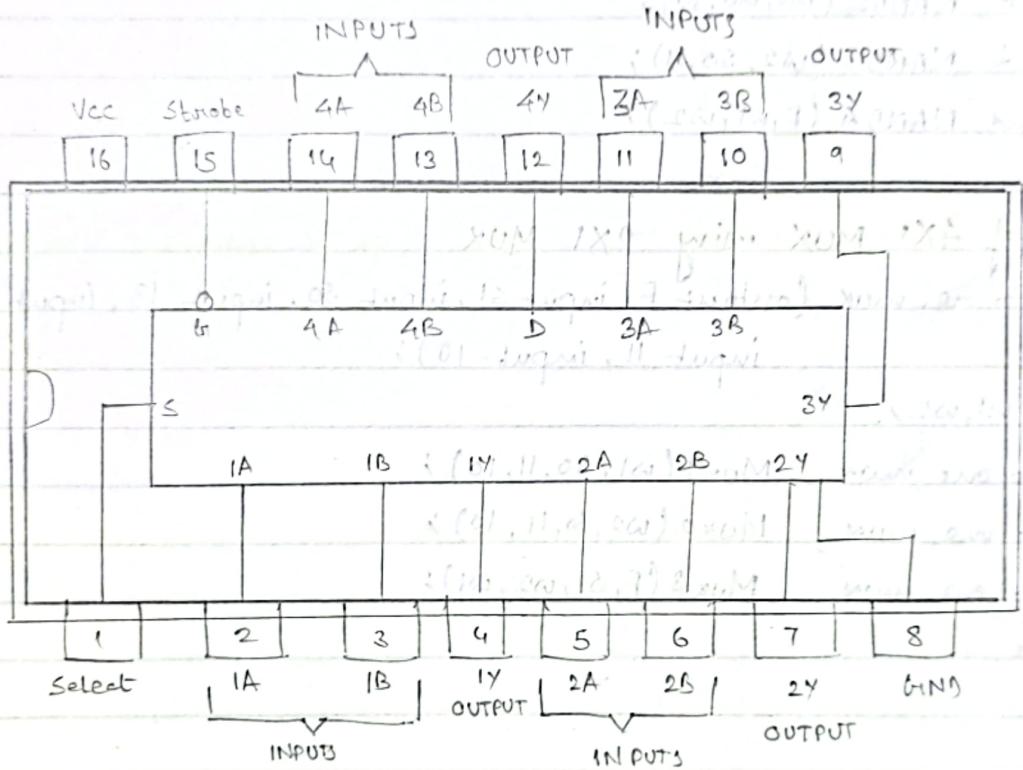
IC74157 Pinout

Y0A Y0B etc give 2x16 bit

16bit digital bus and integrated logic circuit based on CMOS technology

16bit 2x16bit MUX base

Integrating circuit to get



PIN DIAGRAM OF QUAD 2X16 MUX IC74157