

RTL CHALLENGE

DAY-15 :- Verilog Code for Implementation of 8*1 Multiplexer using 2*1 Multiplexer.

Software used:- Xilinx Vivado(2023.1)

Theory:

8*1 MULTIPLEXER USING 2*1:

An 8-to-1 MUX has 8 data inputs, 3 selection lines (to select one of the 8 inputs), and a single output.

A 2-to-1 MUX has 2 data inputs, 1 selection line, and a single output.

The idea is to arrange the 2-to-1 MUXes in a hierarchical structure to achieve the functionality of an 8-to-1 MUX.

Implementation:

Group Inputs: Divide the 8 data inputs into 4 groups of 2 inputs each.

First Stage MUXes: Use 4, 2-to-1 MUXes. Each MUX will take one group of 2 inputs and a selection line. The output of each MUX will be a single data line representing the selected input from that group.

Second Stage MUX: Use another 2-to-1 MUX. This MUX will take the outputs from the first stage (4 data lines) and another 2 selection lines. These selection lines will determine which of the 4 groups (already processed by the first stage MUXes) will be passed to the final output.

Selection Lines: The logic for the selection lines in both stages becomes crucial, You'll need to assign values to these lines to select the desired input among the 8 based on the binary representation.

CODE FOR 8*1 MULTIPLEXER USING 2*1

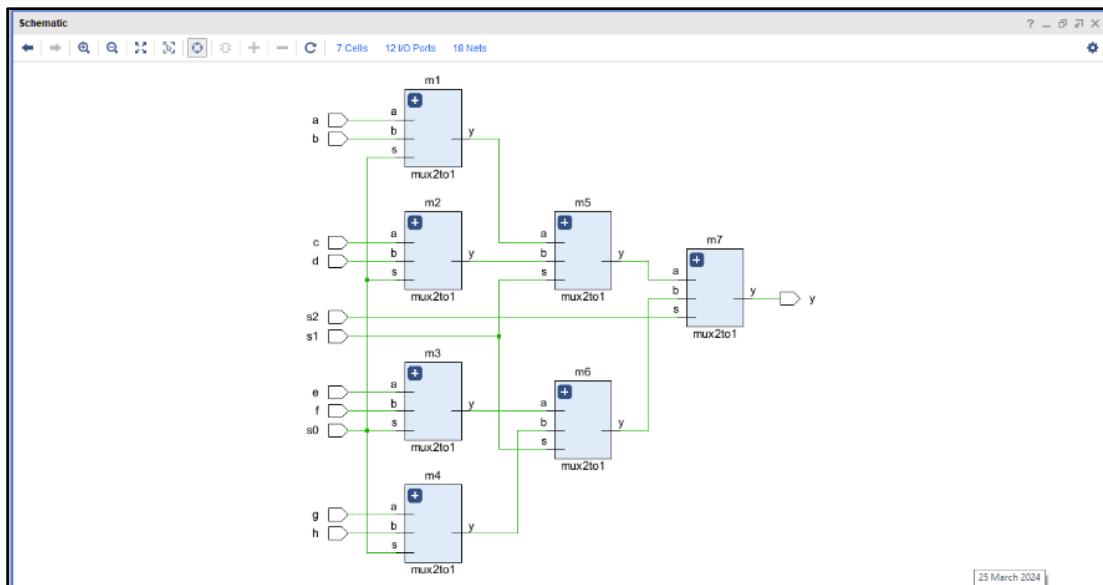
```
module mux8tolusing2to1(
    input a,b,c,d,e,f,g,h,
    input s0,s1,s2,
    output y
);
mux2to1 m1 (.a(a),.b(b),.s(s0),.y(w1));
mux2to1 m2 (.a(c),.b(d),.s(s0),.y(w2));
mux2to1 m3 (.a(e),.b(f),.s(s0),.y(w3));
mux2to1 m4 (.a(g),.b(h),.s(s0),.y(w4));
mux2to1 m5 (.a(w1),.b(w2),.s(s1),.y(w5));
mux2to1 m6 (.a(w3),.b(w4),.s(s1),.y(w6));
mux2to1 m7 (.a(w5),.b(w6),.s(s2),.y(y));

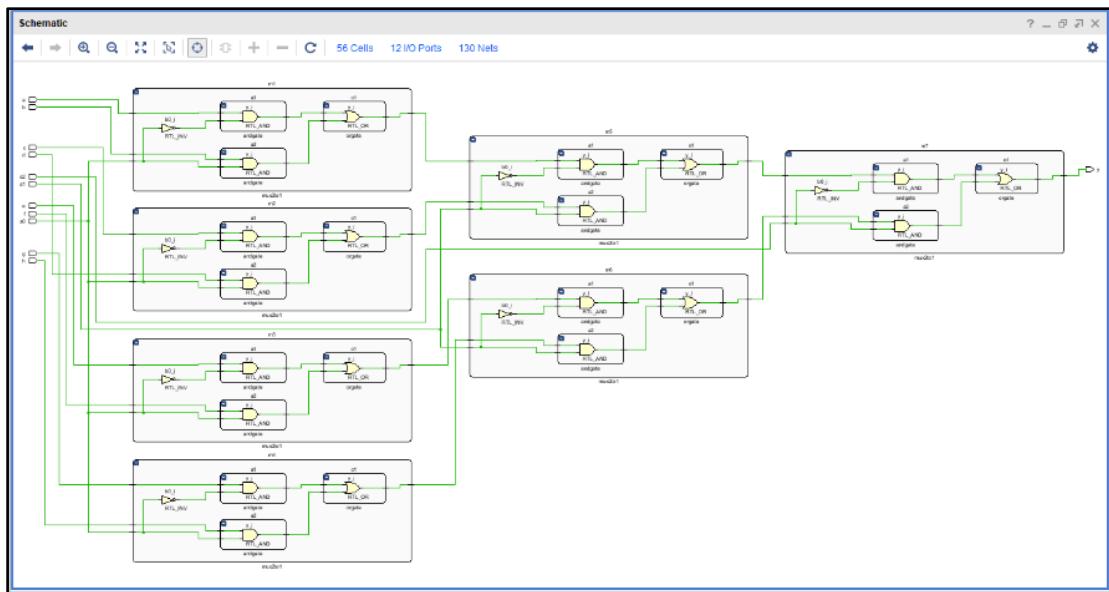
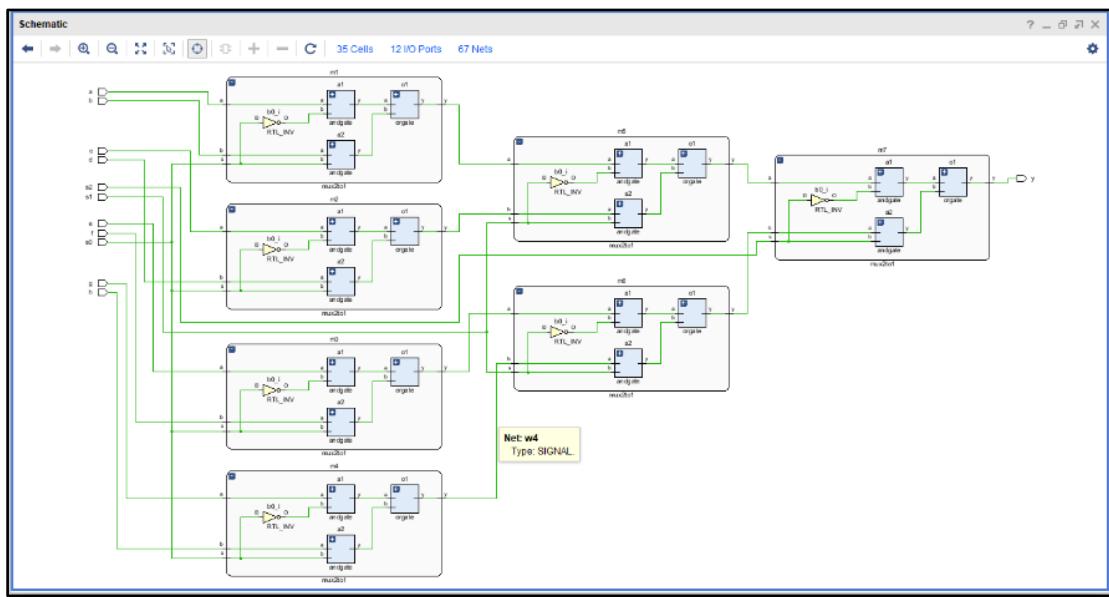
endmodule
```

TESTBENCH:

```
module mux8tolusing2to1_tb();
reg a,b,c,d,e,f,g,h;
reg s0,s1,s2;
wire y;
mux8tolusing2to1 uut (.a(a),.b(b),.c(c),.d(d),.e(e),.f(f),.g(g),.h(h),.s0(s0),.s1(s1),.s2(s2),.y(y));
initial
begin
    a=1;b=0;c=0;d=0;e=0;f=0;g=0;h=0;s0=0;s1=0;s2=0:#100;
    a=0;b=0;c=0;d=0;e=1;f=0;g=0;h=0;s0=0;s1=0;s2=1:#100;
    a=0;b=0;c=1;d=0;e=0;f=0;g=0;h=0;s0=0;s1=1;s2=0:#100;
    a=0;b=0;c=0;d=1;e=0;f=0;g=1;h=0;s0=0;s1=1;s2=1:#100;
    a=0;b=1;c=0;d=0;e=1;f=0;g=0;h=0;s0=1;s1=0;s2=0:#100;
    a=0;b=0;c=0;d=0;e=1;f=1;g=0;h=0;s0=1;s1=0;s2=1:#100;
    a=0;b=0;c=0;d=1;e=0;f=0;g=1;h=0;s0=1;s1=1;s2=0:#100;
    a=1;b=0;c=0;d=0;e=0;f=0;g=0;h=1;s0=1;s1=1;s2=1:#100;
end
endmodule
```

RTL SCHEMATIC:





OUTPUT WAVEFORM:

