30 Days of RTL Coding

-By T Tushar Shenoy

Day 8

Problem Statement: : Implement a 4x1 mux using 2x1 mux using structural style of coding and to write a test bench code for the above using Repeat and Task.

Theory:

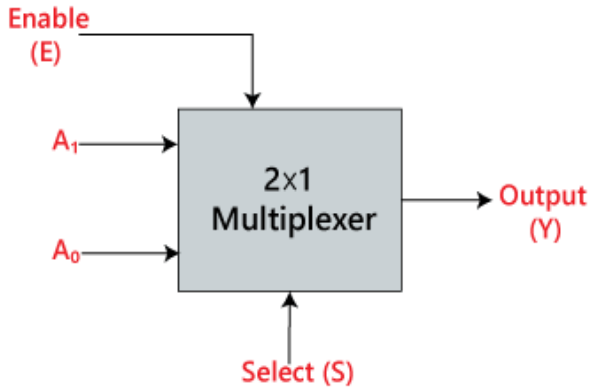
A multiplexer is a combinational circuit that has 2n input lines and a single output line. Simply, the multiplexer is a multi-input and single-output combinational circuit. The binary information is received from the input lines and directed to the output line. On the basis of the values of the selection lines, one of these data inputs will be connected to the output.

Unlike encoder and decoder, there are n selection lines and 2n input lines. So, there is a total of 2N possible combinations of inputs. A multiplexer is also treated as Mux.

2×1 Multiplexer:

In 2×1 multiplexer, there are only two inputs, i.e., A_0 and A_1, 1 selection line, i.e., S_0 and single outputs, i.e., Y. On the basis of the combination of inputs which are present at the selection line S^0, one of these 2 inputs will be connected to the output. The block diagram and the truth table of the 2×1 multiplexer are given below.

Block Diagram:



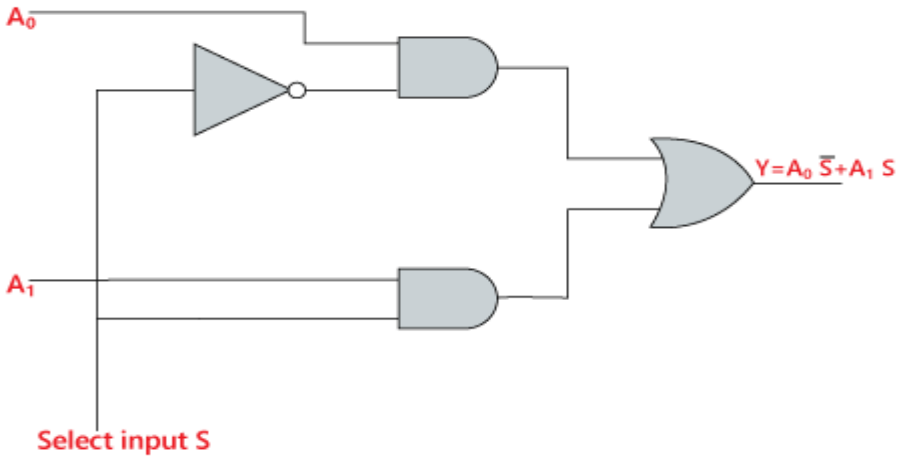
Truth Table:

INPUTS	Output
S_0	Y
0	A_0
1	A_1

The logical expression of the term Y is as follows:

$Y=S_0'.A_0+S_0.A_1$

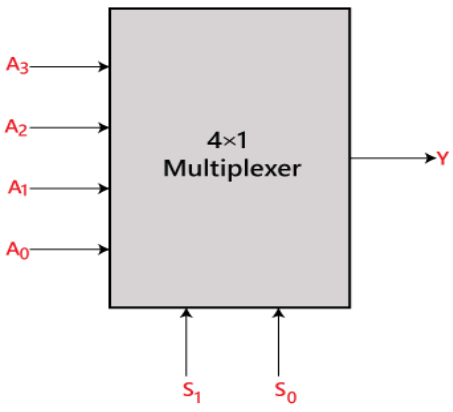
Logical circuit of the above expression is given below:



4×1 Multiplexer:

In the 4×1 multiplexer, there is a total of four inputs, i.e., A_0, A_1, A_2, and A_3, 2 selection lines, i.e., S_0 and S_1 and single output, i.e., Y. On the basis of the combination of inputs that are present at the selection lines S^0 and S_1, one of these 4 inputs are connected to the output. The block diagram and the truth table of the 4×1 multiplexer are given below.

Block Diagram:



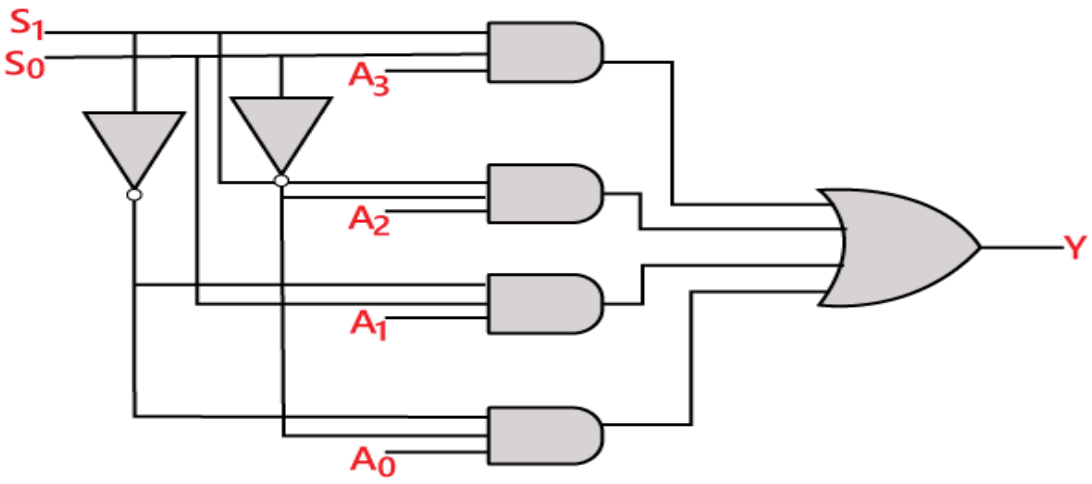
Truth Table:

INPUTS		Output
S_1	S_0	Y
0	0	A_0
0	1	A_1
1	0	A_2
1	1	A_3

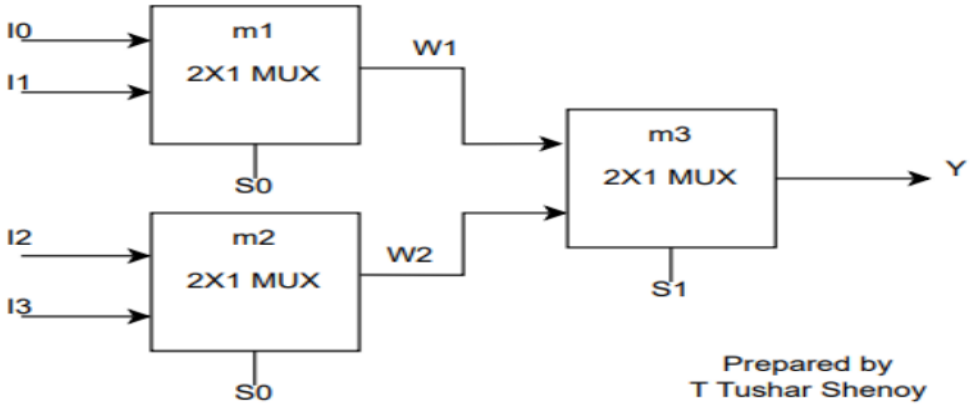
The logical expression of the term Y is as follows:

$Y=S_1' S_0' A_0+S_1' S_0 A_1+S_1 S_0' A_2+S_1 S_0 A_3$

Logical circuit of the above expression is given below:



Block Diagram



Prepared by
T Tushar Shenoy

Verilog Code:

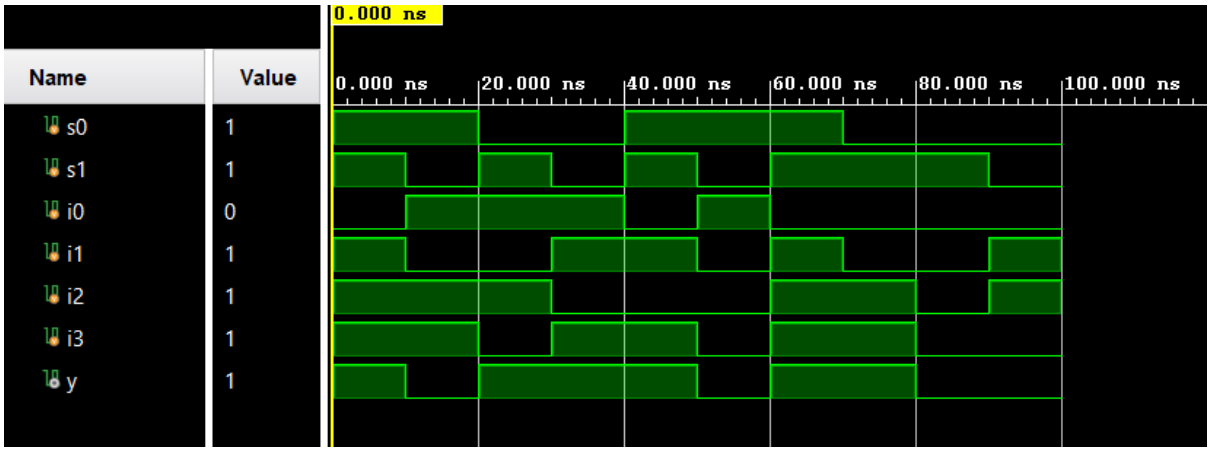
```
// Verilog Code for mux_2x1
module mux_2x1(i0,i1,s,y);
input i0,i1,s;
output y;
assign y=(~s&i0)|(s&i1);
endmodule

//Verilog Code mux_4x1
module mux_4x1(i0,i1,i2,i3,s1,s0,y);
input i0,i1,i2,i3;
input s0,s1;
output y;

wire w1,w2;

mux_2x1 m1(i0,i1,s0,w1);
mux_2x1 m2(i2,i3,s0,w2);
mux_2x1 m3(w1,w2,s1,y);

endmodule
```

Testbench Code:

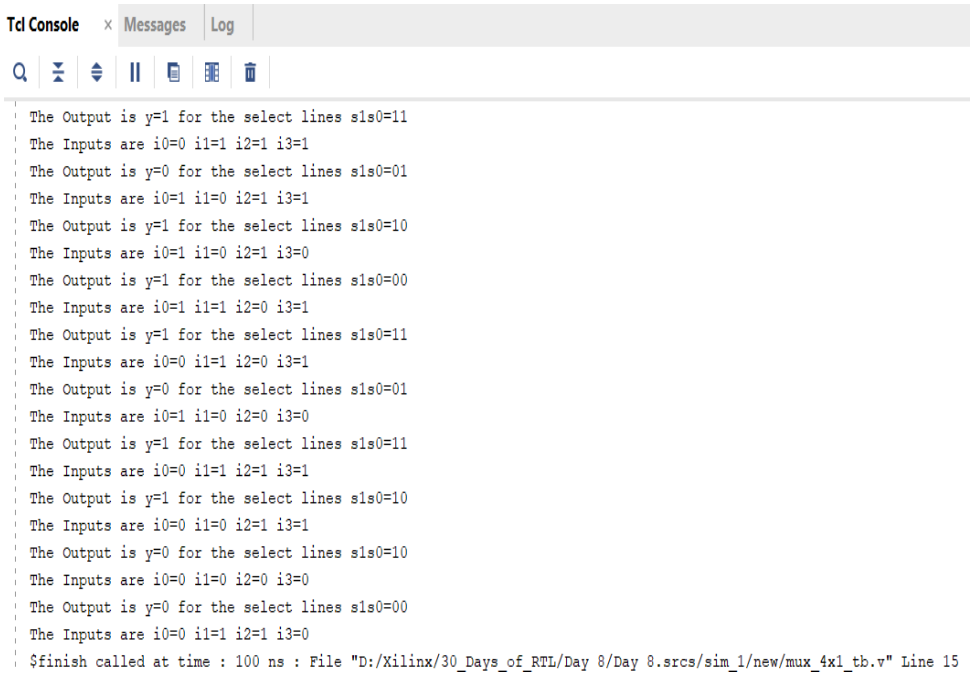
```
//Verilog Testbench Code for mux_4x1
module mux_4x1_tb;
reg s0,s1,i0,i1,i2,i3;
wire y;

mux_4x1 dut(.i0(i0),.i1(i1),.i2(i2),.i3(i3),.s1(s1),.s0(s0),.y(y));

initial begin
repeat(10)
   begin
       stimulus();
        #5 $display("The Output is y=%b for the select lines s1s0=%b%b \nThe Inputs are i0=%b i1=%b i2=%b i3=%b",y,s1,s0,i0,i1,i2,i3);

 end
 $finish;
end

task stimulus;
       begin

               i0=$random;
               i1=$random;
               i2=$random;
               i3=$random;
```

```
        s1=$random;
        s0=$random;

      #5;
    end
  endtask
endmodule
```

Simulation Output:



Console Output:



Schematics:

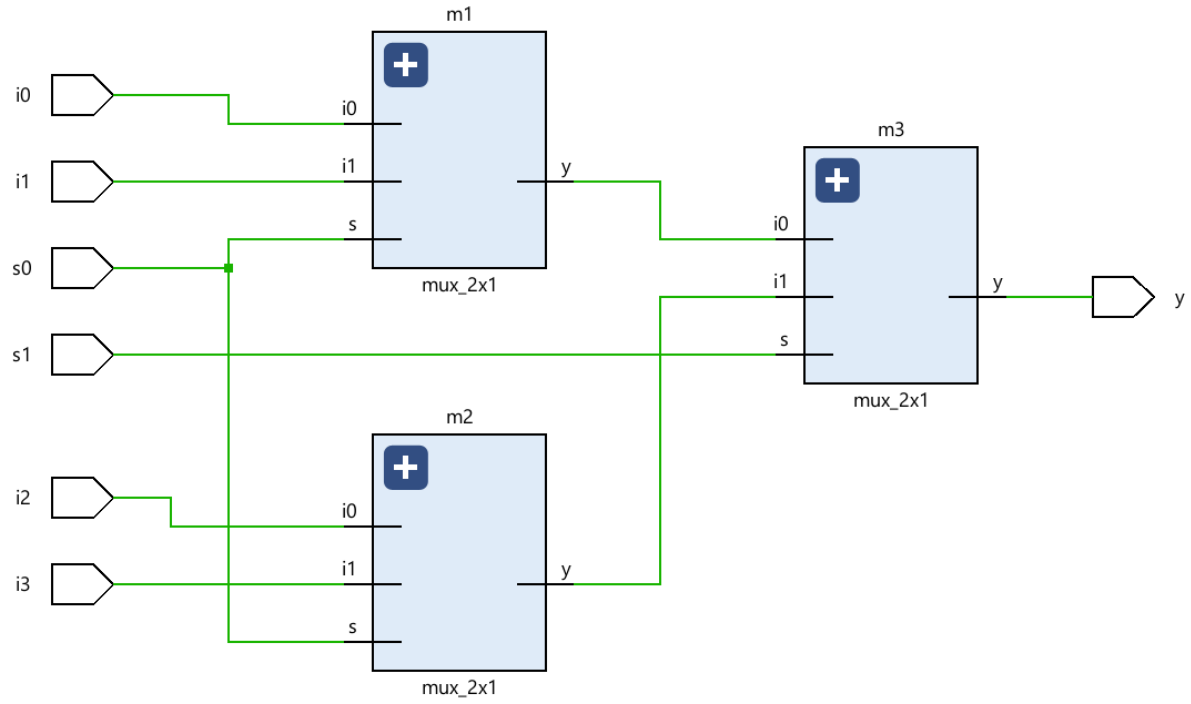


FIG: Schematic of 4x1 Mux using 2x1 Mux (i)

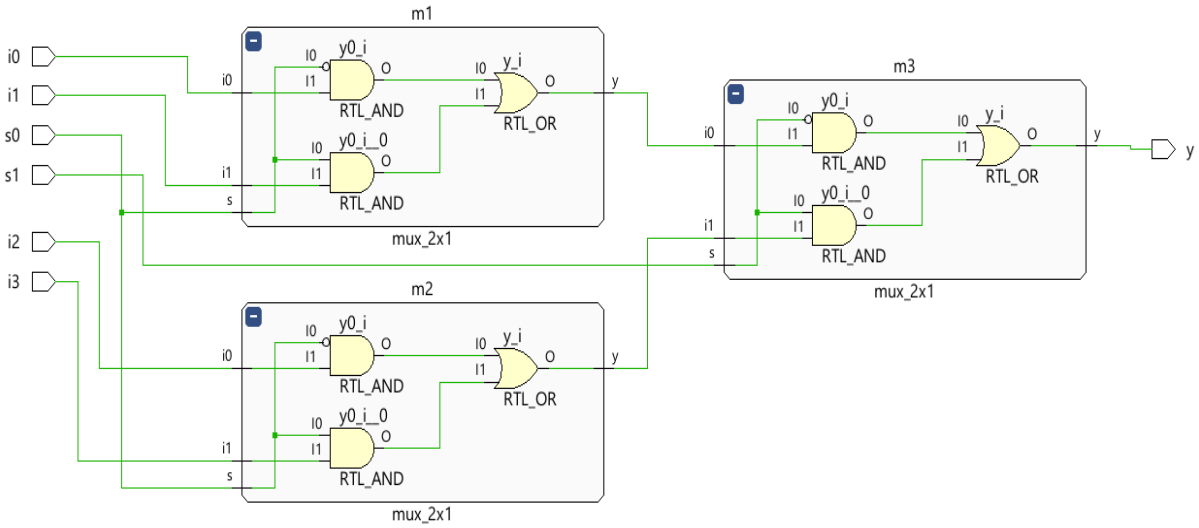


FIG: Schematic of 4x1 Mux using 2x1 Mux (ii)

GitHub Repository URL: - https://github.com/tusharshenoy/RTL-Day-8-4x1Mux-using-2x1Mux