📘 **Day 6 Topic: Verilog codes for MUX (2:1, 4:1, 8:1) Using Cadence (Xcelium)**

🌟 First – What is a Multiplexer (MUX)?

A Multiplexer (MUX) is a digital circuit that selects one input from multiple inputs and sends it to the output based on select lines.

- 2x1 MUX → 2 inputs, 1 output, 1 select line.
- 4x1 MUX → 4 inputs, 1 output, 2 select lines.
- 8x1 MUX → 8 inputs, 1 output, 3 select lines.

General rule:
👉 To select N inputs, we need $\log_2(N)$ select lines.

1) MUX(2:1)

🔹 **1.1) Design Code**

mux2to1.v

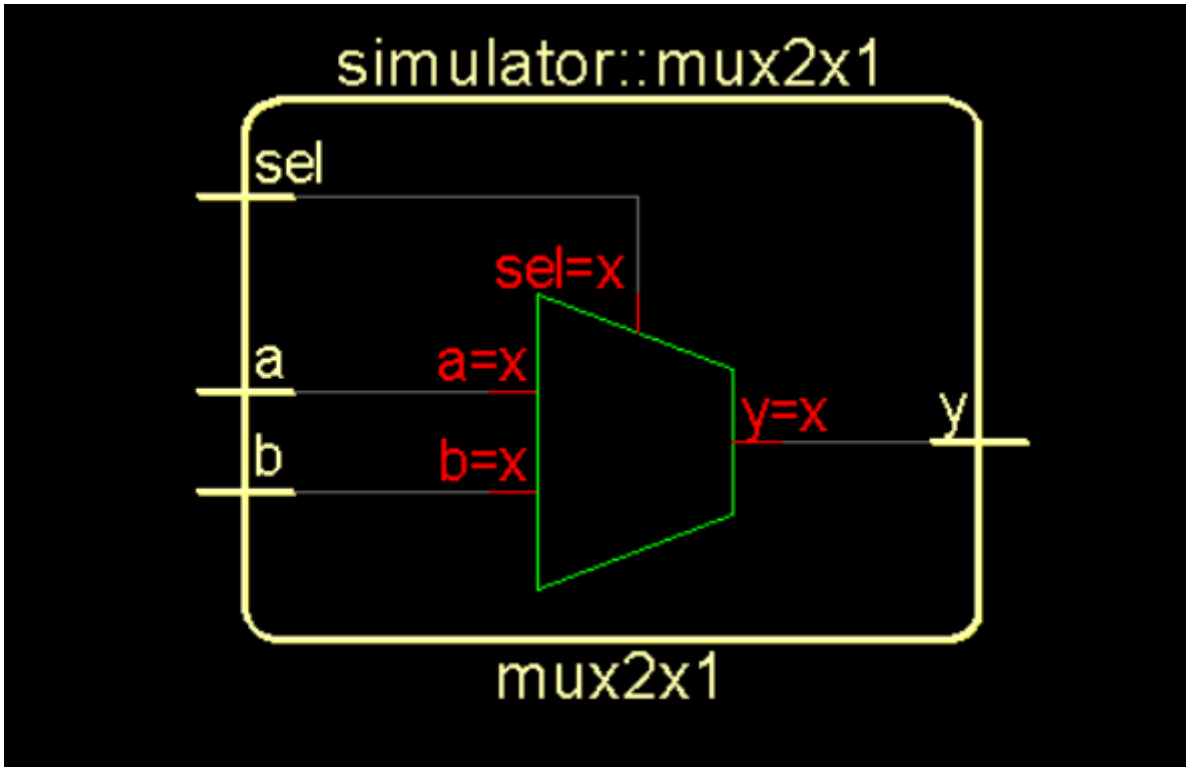
```
module mux2x1 (
    input  wire a, b, sel,
    output wire y
);
    assign y = sel ? b : a;
endmodule
module mux2x1 (
    input  wire a, b, sel,
    output wire y
);
    assign y = sel ? b : a;
endmodule
```

- **1.2) Test Bench Code**

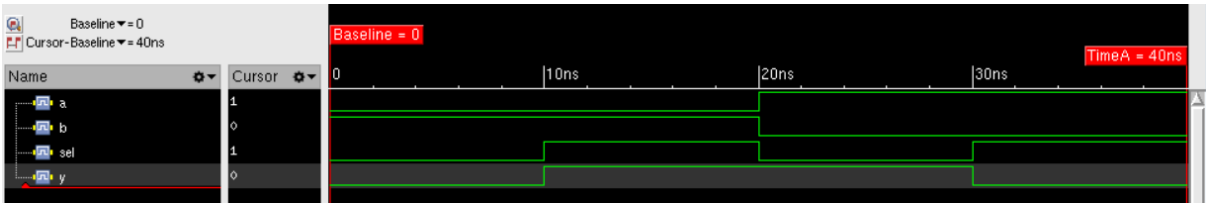
mux2to1_tb.v

```
module mux2x1_tb;
    reg a, b, sel;
    wire y;

    mux2x1 uut (a, b, sel, y);

    initial begin
        $display("a b sel | y");
        a=0; b=1; sel=0; #10 $display("%b %b  %b   | %b", a,b,sel,y);
        a=0; b=1; sel=1; #10 $display("%b %b  %b   | %b", a,b,sel,y);
        a=1; b=0; sel=0; #10 $display("%b %b  %b   | %b", a,b,sel,y);
        a=1; b=0; sel=1; #10 $display("%b %b  %b   | %b", a,b,sel,y);
        $finish;
    end
endmodule
```

- **1.3) Schematic**



- **1.4) Wave Forms**



2) MUX (4:1)

- **2.1) Design Code**

mux4to1.v

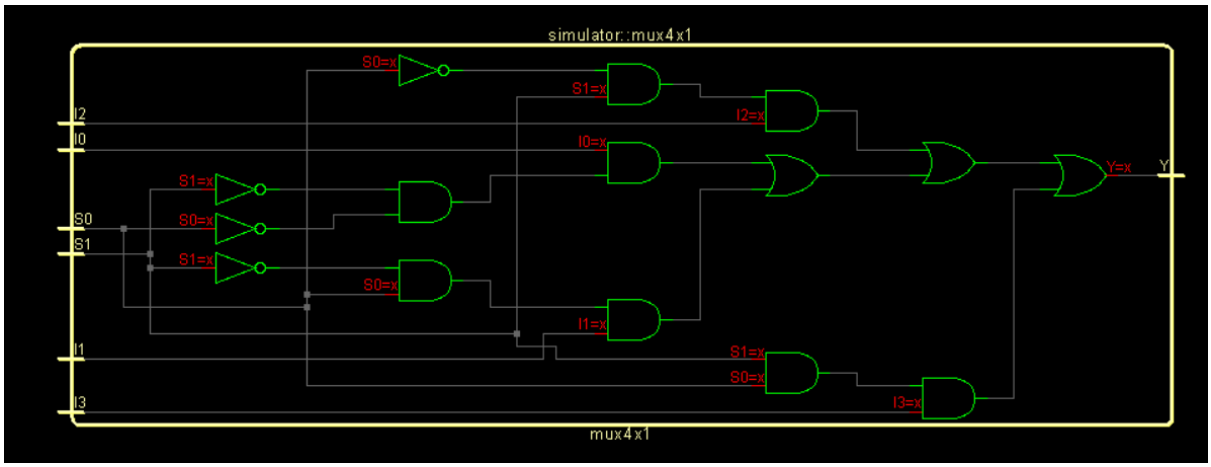
```
module mux4x1 (
    input  wire d0, d1, d2, d3,
    input  wire [1:0] sel,
    output wire y
);
    assign y = (sel == 2'b00) ? d0 :
               (sel == 2'b01) ? d1 :
               (sel == 2'b10) ? d2 : d3;
endmodule
```

- **2.2) Test Bench Code**

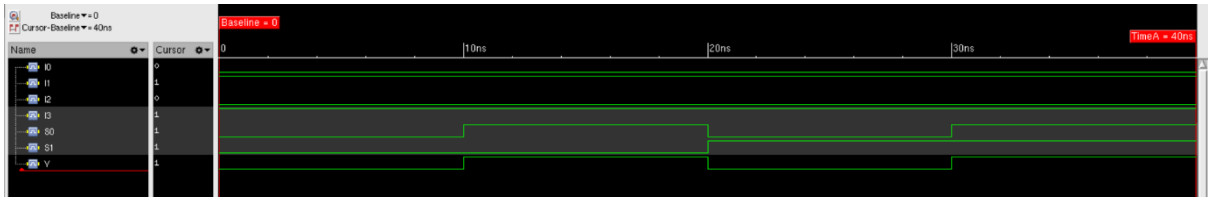
mux4to1_tb.v

```
module mux4x1_tb;
    reg d0, d1, d2, d3;
    reg [1:0] sel;
    wire y;

    mux4x1 uut (d0, d1, d2, d3, sel, y);

    initial begin
        $display("sel d0 d1 d2 d3 | y");
        d0=0; d1=1; d2=0; d3=1;
        sel=2'b00; #10 $display("%b   %b  %b  %b  %b | %b", sel,d0,d1,d2,d3,y);
        sel=2'b01; #10 $display("%b   %b  %b  %b  %b | %b", sel,d0,d1,d2,d3,y);
        sel=2'b10; #10 $display("%b   %b  %b  %b  %b | %b", sel,d0,d1,d2,d3,y);
        sel=2'b11; #10 $display("%b   %b  %b  %b  %b | %b", sel,d0,d1,d2,d3,y);
        $finish;
    end
endmodule
```

- **2.3) Schematic**



- **2.4) Wave Form**



3) MUX (8:1)

- **3.1) Design Code**

mux_8_1.v

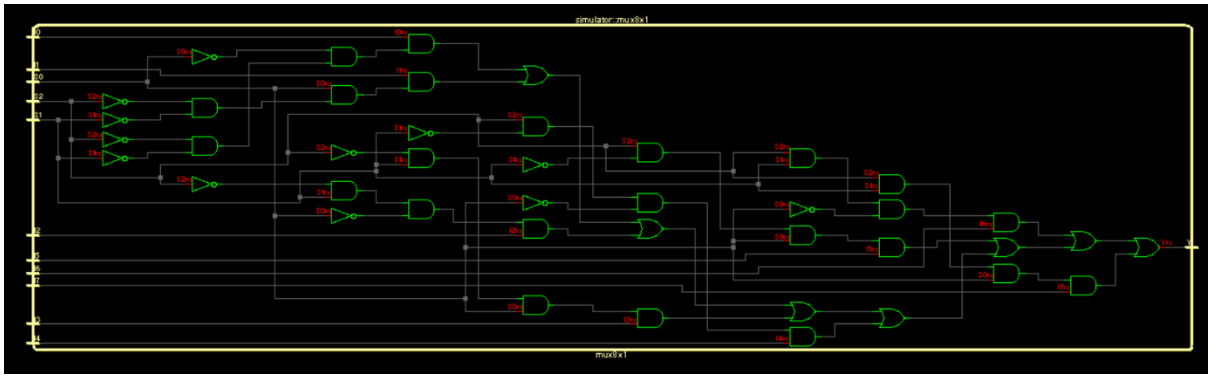
```
module mux8x1 (
    input I0, I1, I2, I3, I4, I5, I6, I7,  // data inputs
    input S0, S1, S2,                      // select lines
    output Y                               // output
);

assign Y = (~S2 & ~S1 & ~S0 & I0) |
           (~S2 & ~S1 &  S0 & I1) |
           (~S2 &  S1 & ~S0 & I2) |
           (~S2 &  S1 &  S0 & I3) |
           ( S2 & ~S1 & ~S0 & I4) |
           ( S2 & ~S1 &  S0 & I5) |
           ( S2 &  S1 & ~S0 & I6) |
           ( S2 &  S1 &  S0 & I7);

endmodule
```

- **3.2) Test bench Code**

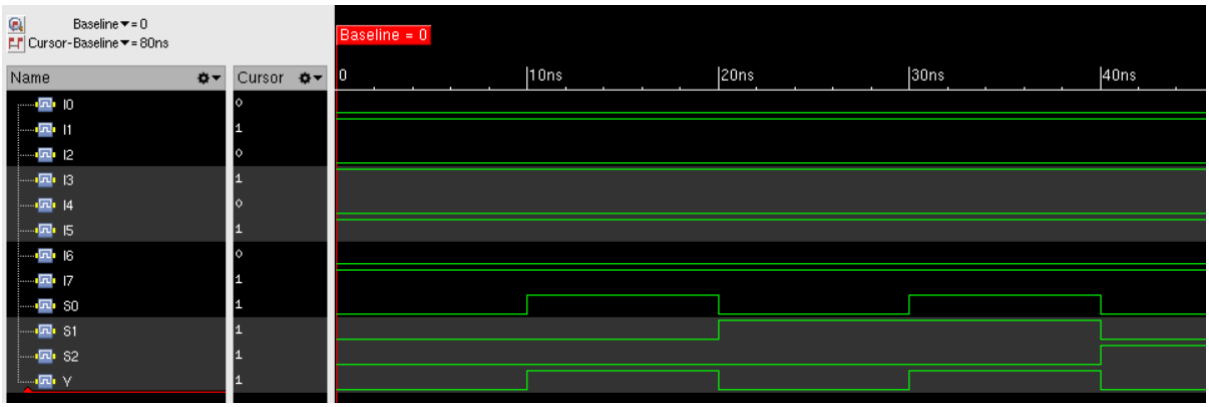
tb.v

```
`timescale 1ns/1ps

module tb_mux8x1;
    reg I0,I1,I2,I3,I4,I5,I6,I7;   // inputs
    reg S0,S1,S2;                  // select lines
    wire Y;                        // output

    // DUT instantiation
    mux8x1 uut (
        .I0(I0), .I1(I1), .I2(I2), .I3(I3),
        .I4(I4), .I5(I5), .I6(I6), .I7(I7),
        .S0(S0), .S1(S1), .S2(S2),
        .Y(Y)
    );

    initial begin
        // Set input values
        I0=0; I1=1; I2=0; I3=1; I4=0; I5=1; I6=0; I7=1;

        // Apply select combinations
        {S2,S1,S0} = 3'b000; #10;  // Expect Y=I0=0
        {S2,S1,S0} = 3'b001; #10;  // Expect Y=I1=1
        {S2,S1,S0} = 3'b010; #10;  // Expect Y=I2=0
        {S2,S1,S0} = 3'b011; #10;  // Expect Y=I3=1
        {S2,S1,S0} = 3'b100; #10;  // Expect Y=I4=0
        {S2,S1,S0} = 3'b101; #10;  // Expect Y=I5=1
        {S2,S1,S0} = 3'b110; #10;  // Expect Y=I6=0
        {S2,S1,S0} = 3'b111; #10;  // Expect Y=I7=1

        $stop;
    end
endmodule
```

🔹 3.3) Schematic



🔹 3.4) Wave Form



✅ Key Points to Remember

- **2x1 MUX → selects between 2 inputs using 1 select line.**
- **4x1 MUX → selects between 4 inputs using 2 select lines.**
- **8x1 MUX → selects between 8 inputs using 3 select lines.**
- **Formula always follows: Y = sum of (select conditions × inputs).**