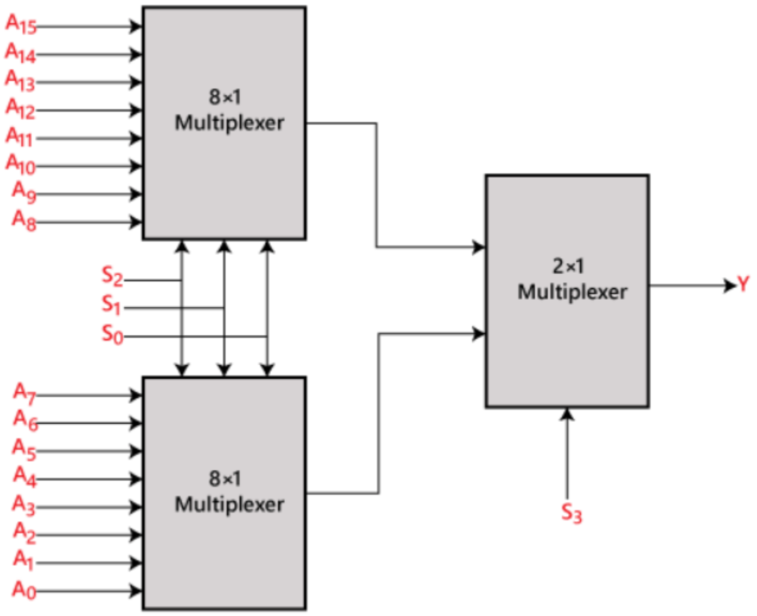
16:1 MUX USING TWO 8:1 MUX

In the 16 to 1 multiplexer, there are total of 16 inputs, i.e., A_0, A_1, …, A_{16}, 4 selection lines, i.e., S_0, S_1, S_2, and S_3 and single output, i.e., Y. On the basis of the combination of inputs that are present at the selection lines S^0, S^1, and S_2, one of these 16 inputs will be connected to the output…



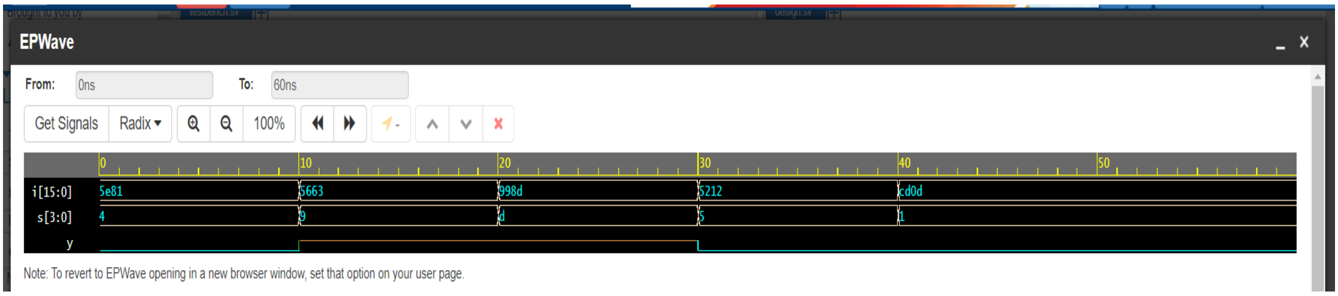
RTL CODE:

```
module MUX(input [1:0] i, input s, output reg y);
  assign y = s ? i[1] : i[0];
endmodule


module mux1(input [7:0] i, input [2:0] s, output reg y);
  always @(*)
    begin
      case(s)
    3'b000: y = i[0];
    3'b001: y = i[1];
    3'b010: y = i[2];
    3'b011: y = i[3];
    3'b100: y = i[4];
    3'b101: y = i[5];
    3'b110: y = i[6];
    3'b111: y = i[7];
   endcase
  end
endmodule

module mux_2(input [15:0] i, input [3:0] s, output reg y);
 wire [7:0] w1, w2;
 mux1 m1 (i[15:8], s[2:0], w1);
 mux1 m2 (i[7:0], s[2:0], w2);
 MUX m3 ({w1, w2}, s[3], y);
Endmodule
```

TESTBENCH

```
module testbench;
 reg [15:0]i;
 reg [3:0]s;
 wire y;
  mux_2 b(i,s,y);
 initial
```

```
  begin
   $dumpfile(".vcd");
   $dumpvars(1);
  end
 initial
  begin
   repeat(5)
    begin
     s=$random; i=$random;
     #10;
    end
  end
 initial
  begin
   #60 $finish();
  end
endmodule
```



Note: To revert to EPWave opening in a new browser window, set that option on your user page.