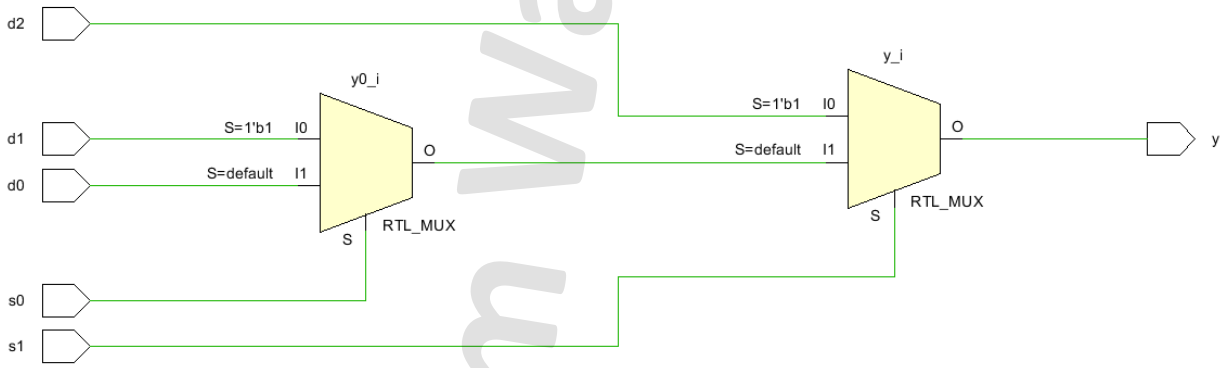
Session 2 Lab

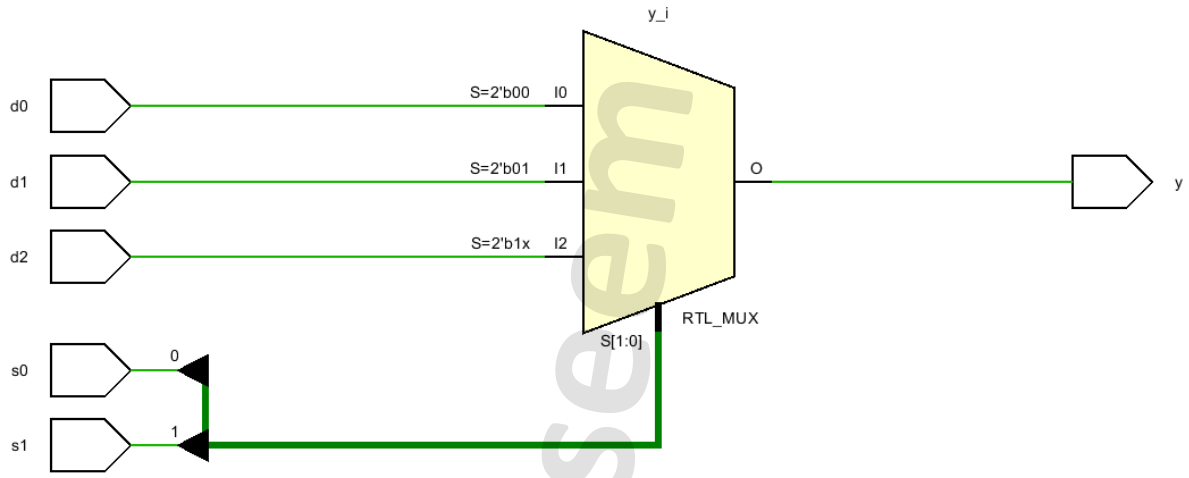
Q1

Implement 3:1 mux using dataflow and behavioral modeling then implement a testbench and run QuestaSim.

```
module mux31_2(d0,d1,d2,s0,s1,y);
  input d0,d1,d2,s0,s1;
  output y;

  assign y = (s1) ? d2 : (s0) ? d1 : d0;
endmodule
```




```
module mux31_3(d0,d1,d2,s0,s1,y);
  input d0,d1,d2,s0,s1;
  output reg y;

  always @ (d0 or d1 or d2 or s0 or s1) begin
    case ({s1,s0})
     2'b00: y = d0;
     2'b01: y = d1;
     2'b10, 2'b11: y = d2;
    endcase
  end
endmodule
```

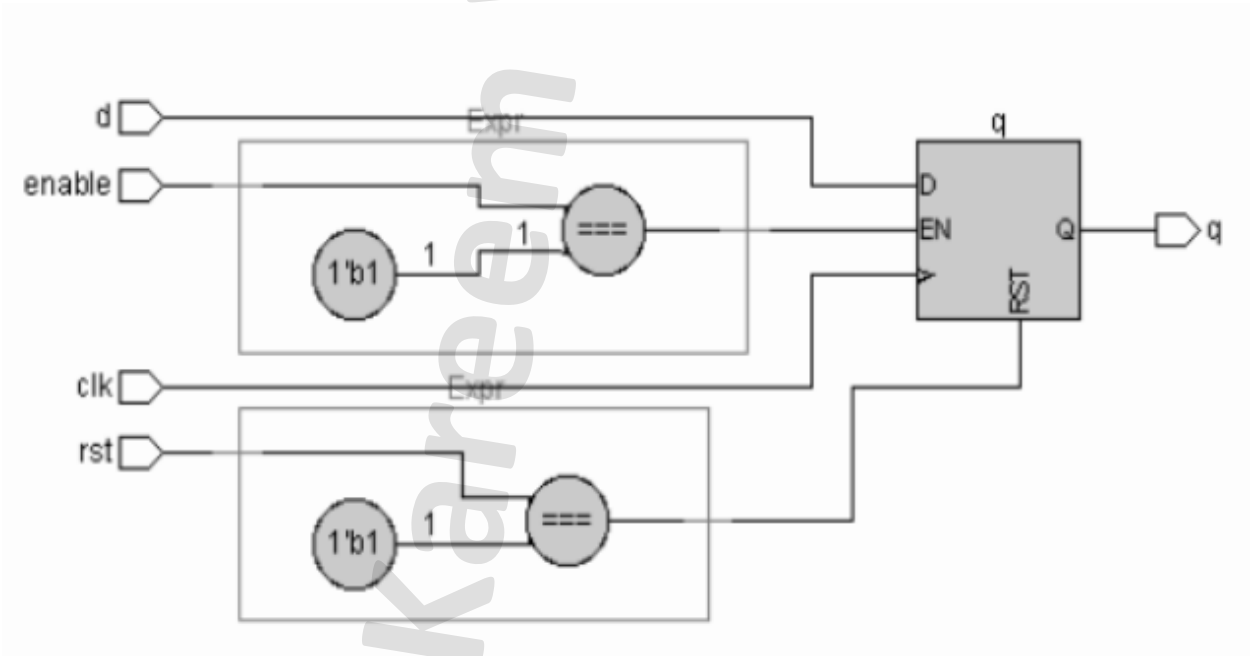



```
module mux31_tb();
  reg d0,d1,d2,s0,s1;
  wire y_dut2, y_dut3;

  mux31_2 dut2(d0,d1,d2,s0,s1,y_dut2);
  mux31_3 dut3(d0,d1,d2,s0,s1,y_dut3);
  integer i;

  initial begin
    for (i = 0; i < 100; i = i+1) begin
      d0 = $random;
      d1 = $random;
      d2 = $random;
      s0 = $random;
      s1 = $random;
      #10;
      if (y_dut2 != y_dut3) begin
        $display("Error - Incorrect mux31 output");
        $stop;
      end
    end
    $stop;
  end

  //Test monitor and results
  initial begin
    $monitor("d0 = %b, d1 = %b, d2 = %b, s0 = %b, s1 = %b, y_dut2 = %b, y_dut3 = %b ", d0, d1, d2, s0, s1, y_dut2, y_dut3);
  end
endmodule
```

Q2

Implement a DFF with clock enable and PRE control signal. Compile the design only without simulation.

```
module dff_en_pre(E, D, Q, clk, PRE);
  input E, D, clk, PRE;
  output reg Q;
  always @(posedge clk) begin
    if (~PRE)
      Q <= 1'b1;
    else if (E)
      Q <= D;
  end
endmodule
```



Common mistake

```
module dff_en_pre(E, D, Q, clk, PRE);
 input E, D, clk, PRE;
 output reg Q;
 always @(posedge clk) begin
  if (~PRE)
   Q <= 1'b1;
  else if (E)
   Q <= D;
  else
   Q <= Q;
 end
endmodule
```