

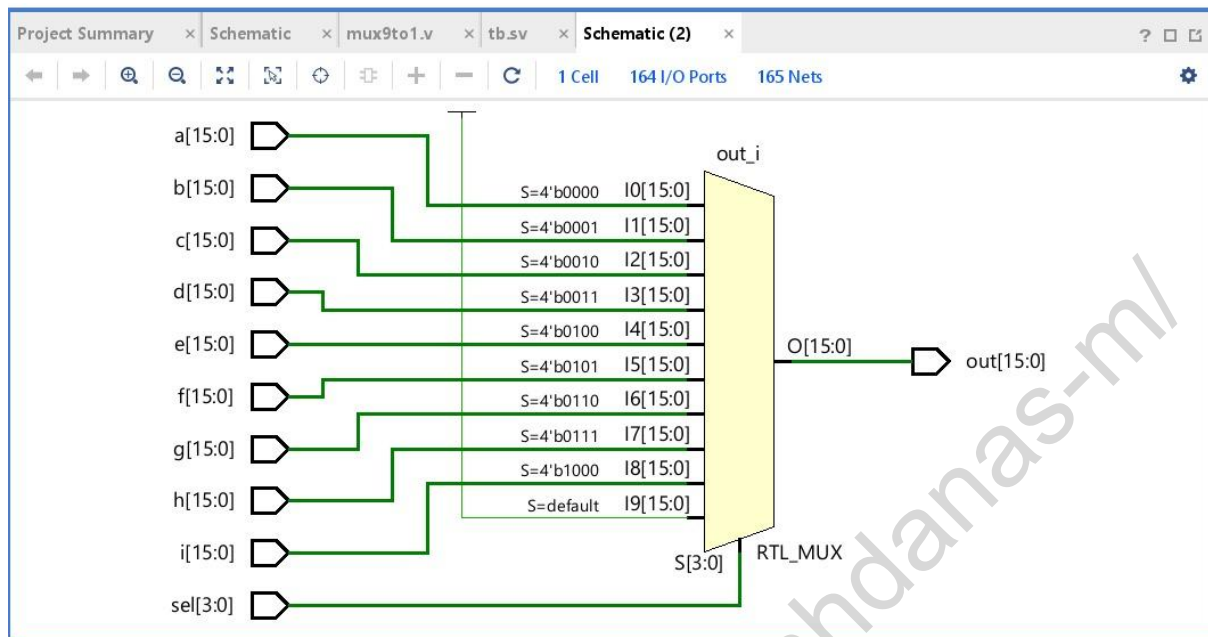
Problem

#Day10: Create a 16-bit wide, 9-to-1 multiplexer. sel=0 chooses a, sel=1 chooses b, etc. For the unused cases (sel=9 to 15), set all output bits to '1'.

Design

```
module mux #(parameter Bitwidth=16)( input [Bitwidth-1:0]
a,b,c,d,e,f,g,h,i,input [3:0]sel,
    output reg [Bitwidth-1:0] out);
    always @(*) begin
        case(sel)
            4'd0:out=a;
            4'd1:out=b;
            4'd2:out=c;
            4'd3:out=d;
            4'd4:out=e;
            4'd5:out=f;
            4'd6:out=g;
            4'd7:out=h;
            4'd8:out=i;
            default:out=16'hffff;
        endcase
    end
endmodule
```

Circuit



Testbench

```
module tb();
    reg [15:0] a,b,c,d,e,f,g,h,i; reg [3:0] sel;
    wire [15:0] out;

    mux a1(a,b,c,d,e,f,g,h,i,sel,out);
    initial begin
    repeat(10) begin
    a=$random;
    b=$random;
    sel=$random;
    c=$random;
    d=$random;
    e=$random;
    f=$random;
    g=$random;
    h=$random;
    i=$random;
    #50;
    end
    end
endmodule
```

Waveform

