`timescale 1ns / 1ps module mux8\_1( output out, input [7:0] I, input [2:0] S ); wire w1, w2; UDP\_mux4 mux0 (w1, S[1], S[0], I[0], I[1], I[2], I[3]); UDP\_mux4 mux1 (w2, S[1], S[0], I[4], I[5], I[6], I[7]); UDP\_mux2 mux3 (out, w1, w2, S[2]); endmodule primitive UDP\_mux2 (out1, in0, in1, s2); output out1; input in0, in1, s2; table // in0 in1 s2 : out1 0 ? 0 : 0; 1 ? 0 : 1; ? 0 1 : 0; ? 1 1 : 1; //0 0 ? : 0; //1 1 ? : 1; endtable endprimitive primitive UDP\_mux4 (y, sel1, sel0, in0, in1, in2, in3); output y; input in0, in1, in2, in3, sel0, sel1; table //sel1 sel0 in0 in1 in2 in3 : out 0 0 0 ? ? ? : 0; 0 0 1 ? ? ? : 1; 0 1 ? 0 ? ? : 0; 0 1 ? 1 ? ? : 1; 1 0 ? ? 0 ? : 0; 1 0 ? ? 1 ? : 1; 1 1 ? ? ? 0 : 0; 1 1 ? ? ? 1 : 1; endtable endprimitive /////////////////////////////////////////////////////////////////////////// //Question 1: 8 to 1 mux test bench `timescale 1ns / 1ps module mux8\_1\_tb(); reg [7:0] I; reg [2:0] S; wire out; mux8\_1 uut ( .out(out), .I(I), .S(S) ); initial $monitor ("I=%b, S=%b, out%b", I, S, out); initial begin #0 I = 8'b10110001; S = 3'b001; #10 I = 8'b00110101; S = 3'b101; #10 I = 8'b00110001; S = 3'b111; #10 I = 8'b00111101; S = 3'b010; #10 $stop; end endmodule