//testbench

module mux4to1tb;

reg a, b, c, d;

reg [1:0] s;

wire out;

mux4to1 uut(.a(a), .b(b), .c(c), .d(d), .s(s), .out(out));

initial begin

$dumpfile("mux4to1.vcd");

$dumpvars(1);

end

initial begin

s = 2'b00; a=1; b=0; c=0; d=0;

#10;

s = 2'b01; a=0; b=1; c=0; d=0;

#10;

s = 2'b10; a=0; b=0; c=1; d=0;

#10;

s = 2'b11; a=0; b=0; c=0; d=1;

#10;

$display("out = %b", out);

$finish;

end

endmodule

module mux4to1tb;

reg a, b, c, d;

reg [1:0] s;

wire out;

mux4to1 uut(.a(a), .b(b), .c(c), .d(d), .s(s), .out(out));

initial begin

$dumpfile("mux4to1.vcd");

$dumpvars(1);

end

initial begin

s = 2'b00; a=1; b=0; c=0; d=0;

#10;

s = 2'b01; a=0; b=1; c=0; d=0;

#10;

s = 2'b10; a=0; b=0; c=1; d=0;

#10;

s = 2'b11; a=0; b=0; c=0; d=1;

#10;

$display("out = %b", out);

$finish;

end

endmodule

//design

module mux4to1(a, b, c, d, s,out );

input a, b, c, d;

input [1:0] s;

output out;

reg out;

always @(a or b or c or d or s) begin

case (s)

2'b00: out = a;

2'b01: out = b;

2'b10: out = c;

2'b11: out = d;

endcase

end

endmodule