

```

module top;
reg strobe;
reg [0:3]data;
reg [1:0]s;

wire out;

// instantiate mux
my_mux gg(.A(s[0]), .B(s[1]), .G(strobe), .C(data), .Y(out) );

// test device. line number is line in function table
initial
begin
    strobe = 1;          // line 1

    #5;
    strobe = 0;
    s = 2'b00;
    data = 4'b0xxx; // line 2
    #5;
    data = 4'b1xxx; // line 3

    #5;
    s = 2'b01;
    data = 4'bx0xx; // line 4
    #5;
    data = 4'bx1xx; // line 5

    #5;
    s = 2'b10;
    data = 4'bxx0x; // line 6
    #5;
    data = 4'bxx1x; // line 7

    #5;
    s = 2'b11;
    data = 4'bxxx0; // line 8
    #5;
    data = 4'bxxx1; // line 9

    #5;
    $finish;
end

endmodule

```

```

module my_mux(A,B,C,G,Y);
`define H 1
`define L 0

input A,B,G;
input [0:3] C;
output Y;
reg Y;

// mux has Y = L whenever G = H. Otherwise A,B selects one of
// the inputs to drive the output Y
//

always@(A or B or C or G)
    if(G == `H)
        Y = `L;
    else
        case({B,A})
            2'b00: Y = C[0];
            2'b01: Y = C[1];
            2'b10: Y = C[2];
            2'b11: Y = C[3];
        endcase

endmodule

```

