// Code design here

module mux\_4to1\_case ( input [3:0] a,

input [3:0] b,

input [3:0] c,

input [3:0] d,

input [1:0] sel,

output reg [3:0] out);

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

case (sel)

2'b00 : out <= a;

2'b01 : out <= b;

2'b10 : out <= c;

2'b11 : out <= d;

endcase

end

endmodule

// Code testbench here

module tb\_4to1\_mux;

reg [3:0] a;

reg [3:0] b;

reg [3:0] c;

reg [3:0] d;

wire [3:0] out;

reg [1:0] sel;

integer i;

mux\_4to1\_case mux0 ( .a (a),

.b (b),

.c (c),

.d (d),

.sel (sel),

.out (out));

initial begin

$monitor ("[%0t] sel=0x%0h a=0x%0h b=0x%0h c=0x%0h d=0x%0h out=0x%0h", $time, sel, a, b, c, d, out);

sel <= 0;

a <= $random;

b <= $random;

c <= $random;

d <= $random;

for (i = 1; i < 4; i=i+1) begin

#5 sel <= i;

end

#5 $finish;

end

initial begin

$dumpvars;

$dumpfile("dump.vcd");

end

endmodule