```
module mux8to1(In0, In1, In2, In3, In4, In5, In6, In7, sel, out);
   input [2:0] In0, In1, In2, In3, In4, In5, In6, In7;
   input [2:0] sel;
   output [2:0] out;

wire [2:0]x,y;
mux4to1 g1 (In0, In1, In2, In3, sel[1:0], x);
mux4to1 g2 (In4, In5, In6, In7, sel[1:0], y);
mux2to1 g3 (x, y, sel[2], out1);

endmodule

module mux4to1(In0, In1, In2, In3, sel, out1);
   input [2:0] In0, In1, In2, In3;
   input [1:0] sel;
   output [2:0] out1;

reg[2:0] out1;
always@(sel)
begin
case(sel)
2'b00:out1=In0;
2'b01:out1=In1;
2'b10:out1=In2;
```

```
2'b11:out1=In3;
endcase
end
endmodule

module mux2to1(In0, In1, sel, out1);
   input [2:0] In0, In1;
   input sel;
   output [2:0] out1;

reg[2:0] out1;
always@(In0,In1,sel)
begin
case(sel)
1'b0:out1=In0;
1'b1:out1=In1;

endcase
end
endmodule



        initial begin
```

```
// Initialize Inputs
In0 = 0;
In1 = 0;
In2 = 0;
In3 = 0;
In4 = 0;
In5 = 0;
In6 = 0;
In7 = 0;
sel = 0;
#100 sel = 010;
#100 sel = 111;
#100 sel = 100;
#100 sel = 101;
#100 sel = 110;
```