Lab No. 3:

1)

module gray2bin (BIN, GRAY);

parameter N = 4;

input [N-1: 0] GRAY;

output [N-1: 0] BIN;

reg [N-1: 0] BIN;

integer i;

always @(GRAY)

begin BIN[N-1] = GRAY[N-1];

for(i = N-2; i >= 0; i = i-1)

BIN[i] = BIN[i+1]^GRAY[i];

end

endmodule

2)

module com4bit(x,y,l,g,e);

input [3:0] x,y;

output l,g,e;

wire l1,g1,e1,l2,g2,e2;

com2bit stage0(x[0],y[0],x[1],y[1],l1,g1,e1);

com2bit stage1(x[2],y[2],x[3],y[3],l2,g2,e2);

assign e = e1 & e2;

assign g = g2|(e2&g1);

assign l = l2|(e2&l1);

endmodule

module com2bit(x0,y0,x1,y1,l,g,e);

input x0,y0,x1,y1;

output l,g,e;

wire i,j;

assign j = ~(x1^y1);

assign i = ~(x0^y0);

assign e = i&j;

assign g = (x1&~y1)|(j&x0&y0);

assign l = ~(g|e);

endmodule

3)

Part 1)

module m81(out, D, S);

input [7:0] D;

input [2:0] S;

wire [7:0] D;

wire [2:0] S;

output out;

reg out;

always@(D or S)

begin

case(S)

0: out=D[0];

1: out=D[1];

2: out=D[2];

3: out=D[3];

4: out=D[4];

5: out=D[5];

6: out=D[6];

7: out=D[7];

endcase

end

endmodule

Part 2)

module m21( D0, D1, S, Y);

input D0, D1, S;

wire D0, D1;

output Y;

reg Y;

always @(D0 or D1 or S)

begin

if(S)

Y= D1;

else

Y=D0;

end

endmodule

Part 3)

module m16to1(out, D, S);

input [15:0] D;

input [3:0] S;

output out;

reg out;

wire [15:0] D;

wire [3:0] S;

wire [1:0] x;

m8to1 m1(x[0], D[7:0], S[2:0]);

m8to1 m2(x[1], D[15:8], S[2:0]);

m2to1 m3(out, x[0], x[1], S[3]);

endmodule

module m2to1(out, D0, D1, S);

input D0, D1, S;

wire D0, D1;

output out;

reg out;

always @(D0 or D1 or S)

begin

if(S)

out= D1;

else

out=D0;

end

endmodule

module m8to1(out, D, S);

input [7:0] D;

input [2:0] S;

wire [7:0] D;

wire [2:0] S;

output out;

reg out;

always@(D or S)

begin

case(S)

0: out=D[0];

1: out=D[1];

2: out=D[2];

3: out=D[3];

4: out=D[4];

5: out=D[5];

6: out=D[6];

7: out=D[7];

endcase

end

endmodule