lab-2

```
module example_ckt(f, x1, x2, x3);
    input x1, x2, x3;
    output f;
    assign f = (x1 & x2) | (~x2 & x3);
endmodule
```

lab-2(with wire)

```
module lab_2(a,b,c,f);
input a, b, c ;
output f ;

wire g,h;

assign g = (~a & b) | (a & ~b);
assign h = g ^ c;
assign f = h & c;

endmodule
```

lab-3

Question : 3>1>2>0

```
module lab3(w,y);
input[3:0]w;
```

```
output reg[1:0]y;

always @(w)
casex(w)
4'b1xxx: y=3;
4'b0x1x: y=1;
4'b010x: y=2;
4'b0001: y=0;
endcase
endmodule
```

Lab-4 Moore

```
module moore11(Clock, Reset, w, z,y,Y);
input Clock, Reset, w;
output reg z;
output reg [1:0] y,Y;
parameter [1:0] A = 0, B = 1, C = 2;


always @(posedge Clock, posedge Reset)
begin
if (Reset == 1)
begin
y = 0;
Y=0;
end
else
begin
y=Y;
case(y)
A: if (w) Y= B;
else Y= A;
B:  if (w) Y = C;
else Y = A;
C:  if (w) Y = C;
else Y = A;
default: Y = 2'bxx;
endcase
end
end
```

```
always @(w, y)
begin
case (y)
A:  z=0;
B:  z=0;
C:  z=1;
endcase
end
endmodule
```

mealy

```
module mealy11 (Clock, Reset, w, z,y,Y);
input Clock, Reset, w;
output reg z;
output reg  y,Y;
parameter  A = 0, B = 1;

always @(posedge Clock, posedge Reset)
            begin
if (Reset == 1)
begin
y = 0;
Y=0;
z=0;
end
else
begin
y=Y;
case(y)
A: if (w)
begin
     Y= B;
     z=0;
end
else
begin
   Y= A;
     z=0;
 end
B: if (w)
begin
     Y= B;
     z=1;
     end
```

```
else
begin
    Y= A;
    z=0;
end
default: Y = 1'bx;
endcase
end
          end

endmodule
```

Example 4: Write down a Verilog HDL code for an up counter with reset operation.

```
module upcounter(rst,clk,q);

input rst,clk;
output reg[3:0]q;

always @(posedge clk, posedge rst)
      begin
            if (rst)
                  q<=0;
            else
                  q<=q+1;
      end
endmodule
```

Sure, here is the Verilog HDL code for an up counter with reset operation, modified to have start bit and end bit as 10 and 01 respectively:

```
module upcounter(rst, clk, q);
input rst, clk;
output reg[3:0] q;
always @(posedge clk, posedge rst)
begin
if (rst)
q <= 10;
else if (q == 01)
q <= 10;
else
q <= q + 1;
```

```
end
endmodule
```

Lab-4 Moore

```
module moore11(Clock, Reset, w, z,y,Y);
input Clock, Reset, w;
output reg z;
output reg [1:0] y,Y;
parameter [1:0] A = 0, B = 1, C = 2;


always @(posedge Clock, posedge Reset)
begin
if (Reset == 1)
begin
y = 0;
Y=0;
end
else
begin
y=Y;
case(y)
A: if (w) Y= B;
else Y= A;
B:  if (w) Y = C;
else Y = A;
C:  if (w) Y = C;
else Y = A;
default: Y = 2'bxx;
endcase
end
end


always @(w, y)
begin
case (y)
A:  z=0;
```

```
B:  z=0;
C:  z=1;
endcase
end
endmodule
```

mealy

```
module mealy11 (Clock, Reset, w, z,y,Y);
input Clock, Reset, w;
output reg z;
output reg  y,Y;
parameter  A = 0, B = 1;

always @(posedge Clock, posedge Reset)
            begin
if (Reset == 1)
begin
y = 0;
Y=0;
z=0;
end
else
begin
y=Y;
case(y)
A: if (w)
begin
     Y= B;
     z=0;
end
else
begin
   Y= A;
     z=0;
 end
B: if (w)
begin
     Y= B;
     z=1;
     end
else
begin
     Y= A;
     z=0;
end
default: Y = 1'bx;
endcase
```

```
end
        end

endmodule
```