decoder3to8.v > ...

```v
module decoder3to8 (
        input en,
        input [2:0] in,
        output reg [7:0] out
    );

always @(*) begin
        if (en) begin
            out = 8'b0;

            case (in)
                  3'b000: out[0]=1'b1;
                  3'b001: out[1]=1'b1;
                  3'b010: out[2]=1'b1;
                  3'b011: out[3]=1'b1;
                  3'b100: out[4]=1'b1;
                  3'b101: out[5]=1'b1;
                  3'b110: out[6]=1'b1;
                  3'b111: out[7]=1'b1;
                  default: out=8'd0;
              endcase
        end
        else begin
            out = 8'b11111111;
        end
    end

    endmodule
```

Figure 1 design

tb.v

```v
    module decoder_tb;
    wire [7:0] out;
    reg [2:0] in;
    reg en;
    integer i;
      decoder3to8 dut(en, in, out);

    initial begin
      $monitor( "en=%b, in=%b, out=%b ", en, in, out);
       for ( i=0; i<16; i=i+1)
            begin
               {en,in}  = i;
                #1;
            end
    end
    endmodule
```

Figure 2 testbench

```
PS D:\Engineering\External learning\Courses\IC Design\Chipions\Assignments> iverilog -o decoder .\decoder3to8.v .\tb.v
PS D:\Engineering\External learning\Courses\IC Design\Chipions\Assignments> vvp .\decoder
en=0, in=000, out=11111111
en=0, in=001, out=11111111
en=0, in=010, out=11111111
en=0, in=011, out=11111111
en=0, in=100, out=11111111
en=0, in=101, out=11111111
en=0, in=110, out=11111111
en=0, in=111, out=11111111
en=1, in=000, out=00000001
en=1, in=001, out=00000010
en=1, in=010, out=00000100
en=1, in=011, out=00001000
en=1, in=100, out=00010000
en=1, in=101, out=00100000
en=1, in=110, out=01000000
en=1, in=111, out=10000000
```

Figure 3 output

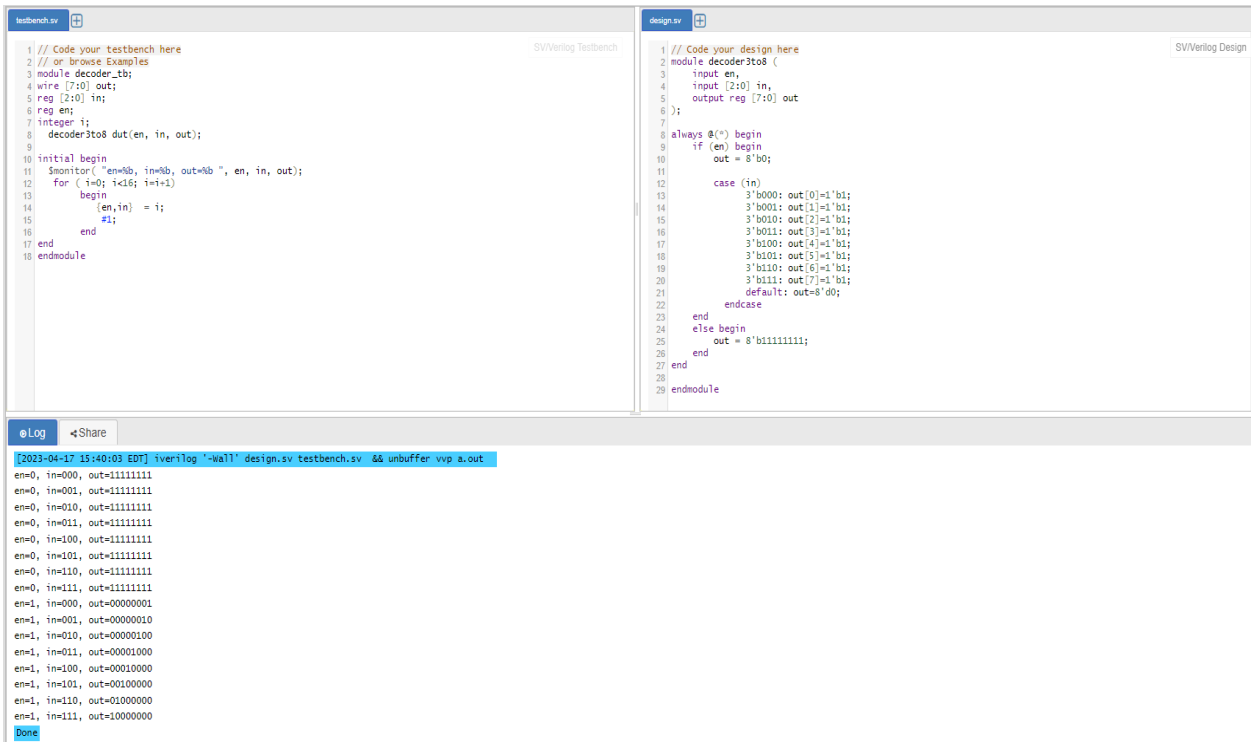

Figure 4 EasyEDA

EasyEDA link: https://www.edaplayground.com/x/KXTx