

```

/////////////////////////////////////////////////////////////////
/// Behavioural Model of Decoder 4 to 8 seven segment
/////////////////////////////////////////////////////////////////
module decoder (SS,S);
output reg [7:0] SS;
input [3:0] S;
always @(S)
case (S)
0 : SS=8'b10000001;
1 : SS=8'b10011111;
2 : SS=8'b01001001;
3 : SS=8'b00001101;
4 : SS=8'b00010111;
5 : SS=8'b00100101;
6 : SS=8'b00100001;
7 : SS=8'b10001111;
8 : SS=8'b00000001;
9 : SS=8'b00000101;
10 : SS=8'b11111110;
default : SS=8'b11111111;
endcase
endmodule
/////////////////////////////////////////////////////////////////
/// Anode decoder
/////////////////////////////////////////////////////////////////
module anode_decoder(AN,cnt);
output reg [3:0] AN;
input [1:0] cnt;
always @(cnt)
case (cnt)
0 : AN=4'b1110;
1 : AN=4'b1101;
2 : AN=4'b1011;
3 : AN=4'b0111;
default : AN=4'b1111;
endcase
endmodule
/////////////////////////////////////////////////////////////////
/// CLK divider
/////////////////////////////////////////////////////////////////
module clkdiv(clk,mclk);
output clk;
input mclk;
reg [15:0] cnt;
assign clk=cnt[15];
initial cnt=0; // Don't forget!! Otherwise simulator does generate X only
always @(posedge mclk)
begin
cnt = cnt+1;
end
endmodule
/////////////////////////////////////////////////////////////////
/// Seven Segment Driver
/////////////////////////////////////////////////////////////////
module SSdriver(SS,AN,mclk,S1,S2,S3,S4);
input mclk;
input [3:0] S1,S2,S3,S4;
output reg [7:0] SS;
output reg [3:0] AN;
wire [7:0] w1,w2,w3,w4;
wire clk;

```

```

wire [3:0] w5;
reg [1:0] cnt;
clkdiv cdiv(clk,mclk);
decoder d1(w1,S1);
decoder d2(w2,S2);
decoder d3(w3,S3);
decoder d4(w4,S4);
anode_decoder d5(w5,cnt);
initial cnt=0;
always @(posedge clk)
begin
cnt=cnt+1;
end
always @(cnt)
begin
AN<=w5;
case (cnt)
2'b00 : SS<=w1;
2'b01 : SS<=w2;
2'b10 : SS<=w3;
2'b11 : SS<=w4;
endcase
end
endmodule

```