

Họ và tên: Đỗ Xuân Sơn

MSSV: 20133299

1. Hiển thị số trên led 7 thanh

```
module hienthiso (S,HEX);
input      [3:0] S;
output reg [0:6] HEX;
  always @ (S)
  case (S)
4'b0000 : HEX = 7'b0000001;
4'b0001 : HEX = 7'b1001111;
4'b0010 : HEX = 7'b0010010;
4'b0011 : HEX = 7'b0000110;
4'b0100 : HEX = 7'b1001100;
4'b0101 : HEX = 7'b0100100;
4'b0110 : HEX = 7'b1100000;
4'b0111 : HEX = 7'b0001111;
4'b1000 : HEX = 7'b0000000;
4'b1001 : HEX = 7'b0001100;
default : HEX = 7'b0000001;
  endcase
endmodule
```

2. Hiển thị đếm 1 số từ 0 đến 9 sau mỗi lần có xung clk1hz vào. Nếu có tín hiệu reset thì về 0.

```
module counter (clock,key_reset,out);
input key_reset,clock;
```

```
output reg [3:0] out;
always @ (negedge key_reset)
begin
if(!key_reset) out = 4'b0000;
else
    begin
        if(out==9) out <= 0;
        else
            out = out +1;
    end
end
always @(posedge clock)
begin
case(out)
    4'b0000 : HEX = 7'b0000001;
    4'b0001 : HEX = 7'b1001111;
    4'b0010 : HEX = 7'b0010010;
    4'b0011 : HEX = 7'b0000110;
    4'b0100 : HEX = 7'b1001100;
    4'b0101 : HEX = 7'b0100100;
    4'b0110 : HEX = 7'b1100000;
    4'b0111 : HEX = 7'b0001111;
    4'b1000 : HEX = 7'b0000000;
    4'b1001 : HEX = 7'b0001100;
    default : HEX = 7'b0000001;
endcase
end
endmodule
```

3. Hiển thị số đếm được trên 2 led 7 thanh từ 29 về 00...

```
module twoled(  
    input clock,  
        input reset,  
        output a,  
        output b,  
        output c,  
        output d,  
        output e,  
        output f,  
        output g,  
        output dp,  
        output [3:0]an  
    );  
  
    reg [3:0]first;  
    reg [1:0]second;  
    reg [22:0] delay;  
    wire test;  
    always @ (posedge clock or negedge reset)  
    begin  
        if (!reset)  
            delay <= 0;  
        else  
            delay <= delay + 1;  
    end  
  
    assign test = &delay;  
    always @ (posedge test or negedge reset)
```

```

begin
    if (!reset) begin
        first <= 0;
        second <= 0;
    end
    else if (first==4'd9) begin
        first <= 0;
        if (second == 2'd2)
            second <= 0;
        else
            second <= second + 1;

    end
    else
        first <= first + 1;
    end

//ghep kenh
localparam N = 18;
reg [N-1:0]count;
always @ (posedge clock or negedge reset)
begin
    if (!reset)
        count <= 0;
    else
        count <= count + 1;
end
reg [6:0]sseg;
reg [3:0]an_temp;

```

```
always @ (*)
begin
    case(count[N-1:N-2])

        2'b00 :
            begin
                sseg = first;
                an_temp = 4'b1110;
            end

        2'b01:
            begin
                sseg = second;
                an_temp = 4'b1101;
            end

        2'b10:
            begin
                sseg = 6'ha;
                an_temp = 4'b1011;
            end

        2'b11:
            begin
                sseg = 6'ha;
                an_temp = 4'b0111;
            end

    endcase
end
```

```
assign an = an_temp;
reg [6:0] sseg_temp;
always @ (*)
begin
    case(sseg)
        4'd0 : sseg_temp = 7'b1000000; //0
        4'd1 : sseg_temp = 7'b1111001; //1
        4'd2 : sseg_temp = 7'b0100100; //2
        4'd3 : sseg_temp = 7'b0110000; //3
        4'd4 : sseg_temp = 7'b0011001; //4
        4'd5 : sseg_temp = 7'b0010010; //5
        4'd6 : sseg_temp = 7'b0000010; //6
        4'd7 : sseg_temp = 7'b1111000; //7
        4'd8 : sseg_temp = 7'b0000000; //8
        4'd9 : sseg_temp = 7'b0010000; //9
        default : sseg_temp = 7'b0111111;
    endcase
end
assign {g, f, e, d, c, b, a} = sseg_temp;
assign dp = 1'b1;
endmodule
```