

# lab8

---

2018.11.23

PB17030889 吉志远

## lab8\_1\_1

代码

```
`timescale 1ns / 1ps

module signal(
    input clk_in, //verilog 100MHz板载时钟源
    input rst_n,
    input Enable,
    output pulse
);
    wire clk_out;
    wire locked;
    //调用IP core,100MHz clk_in, 5MHz out->clk
    clk_wiz_0 inst ( .clk_in1(clk_in), .clk_out1(clk_out), .reset(rst_n), .locked(locked));

    reg [21:0] cnt;
    reg pulse;
    always@(posedge clk_out)
    begin
        if(rst_n)
            cnt<=10'd0;
        if(Enable)
            cnt<=cnt+10'd1;
    end
    always@(posedge clk_out )
    begin
        if(rst_n)
            pulse<=10'd0;
        else if(Enable)
            if(cnt==10'd2499999)
                pulse <= ~pulse;
    end
end
endmodule
```

## lab8\_1\_2

代码

```
module display(
    input clk_in,
    input rst_n,
```

```

input [3:0] bcd,
output reg [7:0] an,//AN
output reg [6:0] seg//7段数管
);
reg [6:0] a1,a2;//十位和个位
always @ (*)
begin
    if(bcd>9)begin a1=7'b1111001;end
    else begin a1 = 7'b1000000; end
end

always @ (*)
begin
    case(bcd)
        0 : a2 <= 7'b1000000;
        1 : a2 <= 7'b1111001;
        2 : a2 <= 7'b0100100;
        3 : a2 <= 7'b0110000;
        4 : a2 <= 7'b0011001;
        5 : a2 <= 7'b0010010;
        6 : a2 <= 7'b0000010;
        7 : a2 <= 7'b1111000;
        8 : a2 <= 7'b0000000;
        9 : a2 <= 7'b0010000;
        10 : a2 <= 7'b1000000;
        11 : a2 <= 7'b1111001;
        12 : a2 <= 7'b0100100;
        13 : a2 <= 7'b0110000;
        14 : a2 <= 7'b0011001;
        15 : a2 <= 7'b0010010;
        default: a2 <= 7'b0000000;

    endcase
end
wire clk_out;
wire locked;

clk_wiz_0 inst (.clk_in1(clk_in),.clk_out1(clk_out),.reset(rst_n),.locked(locked));
//IP例化
parameter update_interval = 5000;
//f=1kHz
reg cursel;
integer cnt;

always @(posedge clk_out)
begin
    cnt <= 1 + cnt;
    if(cnt == update_interval)
    begin
        cnt <= 0;//归0
        cursel <= cursel +1;
    end
end
always @(*)

```

```

begin
    an = 8'b11111111;
    case(cursel)
        1'b0:begin seg = a1;an=8'b111111101;end
        1'b1:begin seg = a2;an=8'b111111110;end
    endcase
end
endmodule

```

## lab8\_2\_1

代码

```

`timescale 1ns / 1ps

module top(
    input  clk,
    input  rst,
    input  sclr,
    input  enable,
    output [7:0]  an,
    output [6:0]  seg
);

    wire  clk_5m;
    wire  rst_n;
    wire  cout;
    reg   pulse_1hz;
    wire  [3:0]  tens_dout;
    wire  [3:0]  units_dout;
    reg   [7:0]  seg_sel;
    reg   [3:0]  seg_din;
    reg   [22:0] cnt;
    clk_wiz_0  clk_wiz_0(
        .clk_in1    (clk),
        .clk_out1   (clk_5m),
        .reset      (rst),
        .locked     (rst_n)
    );

    c_counter_binary_0 units(
        .CLK        (clk_5m),
        .CE         (pulse_1hz & enable),
        .SCLR       (sclr),
        .THRESH0    (cout),
        .Q          (units_dout)
    );

    c_counter_binary_0 tens(
        .CLK        (clk_5m),
        .SCLR       (sclr),
        .CE         (cout & pulse_1hz & enable),
        .THRESH0    (),

```

```

.Q          (tens_dout)
);

bcdto7seg   ctr(
.x          (seg_din),
.sel        (seg_sel),
.an         (an),
.seg        (seg)
);

always@(posedge clk_5m or negedge rst_n)
begin
    if(~rst_n)
        cnt <= 23'h0;
    else if(cnt<23'd5000000)
        cnt <= cnt + 1;
    else
        cnt <= 23'h0;
end

always@(posedge clk_5m or negedge rst_n)
begin
    if(~rst_n)
        pulse_1hz <= 1'b0;
    else if(cnt==23'd4999999)
        pulse_1hz <= 1'b1;
    else
        pulse_1hz <= 1'b0;
end

always@(posedge clk_5m or negedge rst_n)
begin
    if(~rst_n)
    begin
        seg_sel <= 8'b0;
        seg_din <= 4'h0;
    end
    else case(cnt[15:14])
        2'b00:
        begin
            seg_sel <= 8'b1111_1110;
            seg_din <= units_dout;
        end
        2'b01:
        begin
            seg_sel <= 8'b1111_1101;
            seg_din <= tens_dout;
        end
    endcase
end
endmodule

```