实验五—七段管显示器

姓名：张云策 学号：3200105787 学院：计算机科学与技术学院

课程名称：计算机系统 1 同组学生姓名：/

实验时间：2021 .4.2021 实验地点：紫金港东 4-509 指导老师：吴磊

一、 实验目的和要求

1. 理解 constraint 文件。
2. 掌握时钟分频、译码器、多路选择器、时序电路并组合成七段管显示器。
3. 完成七段管模块的编写。

二、 实验内容和原理

2.1 实验内容

设计实现七段管译码器的闪烁显示。

2.2 设计模块

1. 时钟分频器
2. Nexys A7-100T 七段管
3. constraint 文件

三、 主要仪器设备

说明：Vivodo

四、 操作方法与实验步骤

4.1 操作方法

分别拨动四个开关，将其视作二进制数后转化为十六进制数闪烁显示。

4.2 实验步骤

1. 添加开关控制：

根据文件 nexy4_a7_pin.xdc，根据板子标注，分别定义四个接口。

```
# See LICENSE.Cambridge for license details.
set_property IOSTANDARD LVCMOS33 [get_ports sys_clk]
set_property PACKAGE_PIN E3 [get_ports sys_clk]


# Reset active high ST4.1 User button South
set_property IOSTANDARD LVCMOS33 [get_ports rstn]
set_property PACKAGE_PIN C12 [get_ports rstn]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[3]}]
set_property PACKAGE_PIN R15 [get_ports {switch[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[2]}]
set_property PACKAGE_PIN M13 [get_ports {switch[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[1]}]
set_property PACKAGE_PIN L16 [get_ports {switch[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[0]}]
set_property PACKAGE_PIN J15 [get_ports {switch[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[4]}]
set_property PACKAGE_PIN R17 [get_ports {switch[4]}]

```

2. 降低闪烁速度

原文件设置的 flash_clk=clkn[25]，显示大致约 0.5s 一次，将 flash_clk 设置为 clkn 的 23 位后，频率为之前的一半。故实现目标。

```
wire [31:0]clkn;
reg  [31:0] data_src;
// counter for clock_div
counter clk_div(
    .clk(sys_clk),      // clock
    .rstn(rstn),        // RESET Low Enable
    .clkn(clkn)        // clock number [32 bits]
);
assign flash_clk = clkn[24];
// - - - - - - - - - - This part should be changed to your own code
always @(posedge sys_clk) begin
```

3. 七段管的译码器

依照 F 显示，可知：1 为低电平，0 为高电平，故设置不同输出图像为不同值。

```
);
//------This part should be changed to your own decoder.------
always @(*)
begin
 case (hex)
        4'h0:  shape <= 8'b11000000;
        4'h1:  shape <= 8'b11111001;
        4'h2:  shape <= 8'b10100100;
        4'h3:  shape <= 8'b10110000;
        4'h4:  shape <= 8'b10011001;
        4'h5:  shape <= 8'b10010010;
        4'h6:  shape <= 8'b10000010;
        4'h7:  shape <= 8'b11111000;
        4'h8:  shape <= 8'b10000000;
        4'h9:  shape <= 8'b10010000;
        4'ha:  shape <= 8'b10001000;
        4'hb:  shape <= 8'b10000011;
        4'hc:  shape <= 8'b11000110;
        4'hd:  shape <= 8'b10100001;
        4'he:  shape <= 8'b10000110;
        4'hf:  shape <= 8'b10001110;
      default: shape <= 8'b11111111;
    endcase
end
```

4. 七段管扩展：

实现五个输入值，依照之前步骤，设置第五个输入”rstn”,扩展 0~9 位输出为 0~F 输出，并且设置第五位为使能端，使得七段管闪烁。

```
5'b01010:data_src<=32'hAAAAAAAA;
5'b01011:data_src<=32'hBBBBBBBB;
5'b01100:data_src<=32'hCCCCCCCC;
5'b01101:data_src<=32'hDDDDDDDD;
5'b01110:data_src<=32'hEEEEEEEE;
5'b01111:data_src<=32'hFFFFFFFF;
```

```
4'ha:  shape <= 8'b10001000;
4'hb:  shape <= 8'b10000011;
4'hc:  shape <= 8'b11000110;
4'hd:  shape <= 8'b10100001;
4'he:  shape <= 8'b10000110;
4'hf:  shape <= 8'b10001110;
default: shape <= 8'b11111111;
```

四、实验结果与分析

仿真激励代码即赋值 switch 从 00000~11111；分别判断不同输入与输出

P2S 分析：

首先，文件定义时钟，重置，显示形状数组 shape 定义 num_csn，num_an

```
module P2S(
    input  wire        clk,
    input  wire        rstn,
    input  wire [63:0] shape,
    output reg  [7:0]  num_csn,
    output wire [7:0]  num_an
);
```

定义 state，存放 num_an 状态，之后判断状态，赋值 shape，使得在不同情况下显示不同图像，并且重新赋值 state。

如果 rstn 为 0，则赋值 num_csn，并赋值 state 为 0；实现全暗功能。

```
always @ (posedge clk) begin
    if (rstn) begin
        case (state)
            8'b11111110: begin num_csn <= shape[15: 8]; state <= 8'b11111101; end
            8'b11111101: begin num_csn <= shape[23:16]; state <= 8'b11111011; end
            8'b11111011: begin num_csn <= shape[31:24]; state <= 8'b11110111; end
            8'b11110111: begin num_csn <= shape[39:32]; state <= 8'b11101111; end
            8'b11101111: begin num_csn <= shape[47:40]; state <= 8'b11011111; end
            8'b11011111: begin num_csn <= shape[55:48]; state <= 8'b10111111; end
            8'b10111111: begin num_csn <= shape[63:56]; state <= 8'b01111111; end
            8'b01111111: begin num_csn <= shape[ 7: 0]; state <= 8'b11111110; end
            default:     begin num_csn <= shape[ 7: 0]; state <= 8'b11111110; end
        endcase
    end else begin
        num_csn <= 8'b11000000;
        state <= 8'b00000000;
    end
end
```

五、 讨论、心得

说明：没有简单，非常困难（/(ㄒ o ㄒ)/~~）