

实验平台说明

特征	说明
操作系统版本和系统类型	Windows 7 64 bit enterprise edition
EDA 软件名称和版本	Quartus II 13.1 64 bit
DE2-115 开发板编号	1904615S

自查清单

特征	Part 8/Lab6	Part9/Lab6	Part/Lab	Part ?/Lab?
原理图	√	√	√	
Verilog 代码	√	√	√	
Flow Summary	√	√	×	
RTL 图	√	√	×	
状态机图	×	×	×	
工艺图	√	√	√	
仿真图	×	×	×	
SignalTapII 图	×	×	×	
硬件运行图	√	√	√	

第一部分 Part 5/Lab3

一、实验目的

1. 研究锁存器，触发器和寄存器
2. 熟悉锁存器，触发器，寄存器的设计方法。

二、实验内容

- 1.使用 LAB1 中数码管驱动程序。
- 2.使用 PART2 中 D 触发器。
- 3.写出可以储存 4 个字节的的触发器

三、实验原理

1. 以 15 个开关为输入，以 KEY1 为 CLK，KEY0 为 reset 输出映射到 8 个数码管上。。
2. 利用触发器可以暂时存储的特点，储存第一次输入的值。

四、实验步骤

1) 建立工程，并添加文件。

- 1.建立工程，选择常用文件夹，取名为 lab3。
- 2.添加 Block Diagram/Schematic 文件，取名为 lab3;
- 3.添加 Verilog HDL 文件，取名为 PART4.5
- 4.添加 VMF 文件，取名为 lab3.vmf

2) 首先理清思路：

在本次实验中，由于输出需要锁存，且为两级。即从输入到数码管的输入，到数码管的 IO。则有两种方案，1，从数码管前端锁存，需要 28 个锁存器（仅对 A 的 IO 操作即可,B 可保留）。

2.输入到数码管输入的前端，需要 16 个锁存器。

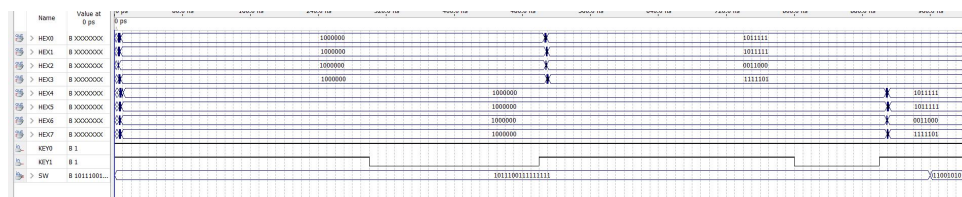
由于理解上的失误，我先尝试了第一种比较麻烦的方式，做出了比较玄学的结果。

3) 以 Verilog 代码表示。

4) 编译并将引脚对应。

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved
HEX0[6]	Output	PIN_H22	6	B6_N0	PIN_H22	2.5 V (default)	
HEX0[5]	Output	PIN_J22	6	B6_N0	PIN_J22	2.5 V (default)	
HEX0[4]	Output	PIN_L25	6	B6_N1	PIN_L25	2.5 V (default)	
HEX0[3]	Output	PIN_L26	6	B6_N1	PIN_L26	2.5 V (default)	
HEX0[2]	Output	PIN_E17	7	B7_N2	PIN_E17	2.5 V (default)	8m
HEX0[1]	Output	PIN_F22	7	B7_N0	PIN_F22	2.5 V (default)	8m
HEX0[0]	Output	PIN_G18	7	B7_N2	PIN_G18	2.5 V (default)	8m
HEX1[6]	Output	PIN_U24	5	B5_N0	PIN_U24	2.5 V (default)	8m
HEX1[5]	Output	PIN_U23	5	B5_N1	PIN_U23	2.5 V (default)	8m
HEX1[4]	Output	PIN_W25	5	B5_N1	PIN_W25	2.5 V (default)	8m
HEX1[3]	Output	PIN_W22	5	B5_N0	PIN_W22	2.5 V (default)	8m
HEX1[2]	Output	PIN_W21	5	B5_N1	PIN_W21	2.5 V (default)	8m
HEX1[1]	Output	PIN_V22	5	B5_N0	PIN_V22	2.5 V (default)	8m
HEX1[0]	Output	PIN_M24	6	B6_N2	PIN_M24	2.5 V (default)	8m
HEX2[6]	Output	PIN_W28	5	B5_N1	PIN_W28	2.5 V (default)	8m
HEX2[5]	Output	PIN_W27	5	B5_N1	PIN_W27	2.5 V (default)	8m
HEX2[4]	Output	PIN_Y26	5	B5_N1	PIN_Y26	2.5 V (default)	8m
HEX2[3]	Output	PIN_W26	5	B5_N1	PIN_W26	2.5 V (default)	8m
HEX2[2]	Output	PIN_V25	5	B5_N1	PIN_V25	2.5 V (default)	8m
HEX2[1]	Output	PIN_AA26	5	B5_N1	PIN_AA26	2.5 V (default)	8m
HEX2[0]	Output	PIN_AA25	5	B5_N1	PIN_AA25	2.5 V (default)	8m
HEX3[6]	Output	PIN_Y19	4	B4_N0	PIN_Y19	2.5 V (default)	8m
HEX3[5]	Output	PIN_AF23	4	B4_N0	PIN_AF23	2.5 V (default)	8m
HEX3[4]	Output	PIN_AD24	4	B4_N0	PIN_AD24	2.5 V (default)	8m
HEX3[3]	Output	PIN_A21	4	B4_N0	PIN_A21	2.5 V (default)	8m
HEX3[2]	Output	PIN_AB20	4	B4_N0	PIN_AB20	2.5 V (default)	8m
HEX3[1]	Output	PIN_U21	5	B5_N0	PIN_U21	2.5 V (default)	8m
HEX3[0]	Output	PIN_V21	5	B5_N1	PIN_V21	2.5 V (default)	
HEX4[6]	Output	PIN_AE18	4	B4_N2	PIN_AE18	2.5 V (default)	
HEX4[5]	Output	PIN_AF19	4	B4_N1	PIN_AF19	2.5 V (default)	
HEX4[4]	Output	PIN_AE19	4	B4_N1	PIN_AE19	2.5 V (default)	
HEX4[3]	Output	PIN_AH21	4	B4_N2	PIN_AH21	2.5 V (default)	8m
HEX4[2]	Output	PIN_AG21	4	B4_N2	PIN_AG21	2.5 V (default)	8m
HEX4[1]	Output	PIN_AA19	4	B4_N0	PIN_AA19	2.5 V (default)	8m
HEX4[0]	Output	PIN_AB19	4	B4_N0	PIN_AB19	2.5 V (default)	8m
HEX5[6]	Output	PIN_AH18	4	B4_N2	PIN_AH18	2.5 V (default)	8m
HEX5[5]	Output	PIN_AF18	4	B4_N1	PIN_AF18	2.5 V (default)	8m
HEX5[4]	Output	PIN_AG19	4	B4_N2	PIN_AG19	2.5 V (default)	8m
HEX5[3]	Output	PIN_AH19	4	B4_N2	PIN_AH19	2.5 V (default)	8m
HEX5[2]	Output	PIN_AB18	4	B4_N0	PIN_AB18	2.5 V (default)	8m
HEX5[1]	Output	PIN_AC18	4	B4_N1	PIN_AC18	2.5 V (default)	8m
HEX5[0]	Output	PIN_AD18	4	B4_N1	PIN_AD18	2.5 V (default)	8m
HEX6[6]	Output	PIN_AC17	4	B4_N2	PIN_AC17	2.5 V (default)	8m
HEX6[5]	Output	PIN_AA15	4	B4_N2	PIN_AA15	2.5 V (default)	8m
HEX6[4]	Output	PIN_AB15	4	B4_N2	PIN_AB15	2.5 V (default)	8m
HEX6[3]	Output	PIN_AB17	4	B4_N1	PIN_AB17	2.5 V (default)	8m
HEX6[2]	Output	PIN_AA16	4	B4_N2	PIN_AA16	2.5 V (default)	8m
HEX6[1]	Output	PIN_AB16	4	B4_N2	PIN_AB16	2.5 V (default)	8m
HEX6[0]	Output	PIN_AA17	4	B4_N1	PIN_AA17	2.5 V (default)	8m
HEX7[6]	Output	PIN_AA14	3	B3_N0	PIN_AA14	2.5 V (default)	8m
HEX7[5]	Output	PIN_AG18	4	B4_N2	PIN_AG18	2.5 V (default)	8m
HEX7[4]	Output	PIN_AF17	4	B4_N2	PIN_AF17	2.5 V (default)	8m
HEX7[3]	Output	PIN_AH17	4	B4_N2	PIN_AH17	2.5 V (default)	8m
HEX7[2]	Output	PIN_AG17	4	B4_N2	PIN_AG17	2.5 V (default)	8m
HEX7[1]	Output	PIN_AE17	4	B4_N2	PIN_AE17	2.5 V (default)	8m
HEX7[0]	Output	PIN_AD17	4	B4_N2	PIN_AD17	2.5 V (default)	8m
KEY0	Input	PIN_M23	6	B6_N2	PIN_M23	2.5 V (default)	8m
KEY1	Input	PIN_M21	6	B6_N1	PIN_M21	2.5 V (default)	8m
SW[15]	Input	PIN_AA22	5	B5_N2	PIN_AA22	2.5 V (default)	8m
SW[14]	Input	PIN_AA23	5	B5_N2	PIN_AA23	2.5 V (default)	8m
SW[13]	Input	PIN_AA24	5	B5_N2	PIN_AA24	2.5 V (default)	8m
SW[12]	Input	PIN_AB23	5	B5_N2	PIN_AB23	2.5 V (default)	8m
SW[11]	Input	PIN_AB24	5	B5_N2	PIN_AB24	2.5 V (default)	8m
SW[10]	Input	PIN_AC24	5	B5_N2	PIN_AC24	2.5 V (default)	8m
SW[9]	Input	PIN_AB25	5	B5_N1	PIN_AB25	2.5 V (default)	8m
SW[8]	Input	PIN_AC25	5	B5_N2	PIN_AC25	2.5 V (default)	8m
SW[7]	Input	PIN_AB26	5	B5_N1	PIN_AB26	2.5 V (default)	8m
SW[6]	Input	PIN_AD26	5	B5_N2	PIN_AD26	2.5 V (default)	8m
SW[5]	Input	PIN_AC26	5	B5_N2	PIN_AC26	2.5 V (default)	8m
SW[4]	Input	PIN_AB27	5	B5_N1	PIN_AB27	2.5 V (default)	8m
SW[3]	Input	PIN_AD27	5	B5_N2	PIN_AD27	2.5 V (default)	8m
SW[2]	Input	PIN_AC27	5	B5_N2	PIN_AC27	2.5 V (default)	8m
SW[1]	Input	PIN_AC28	5	B5_N2	PIN_AC28	2.5 V (default)	8m
SW[0]	Input	PIN_AB28	5	B5_N1	PIN_AB28	2.5 V (default)	8m
KEY2	Unknown	PIN_N21	6	B6_N2		2.5 V (default)	8m

5) 仿真



结果正确

五、实验代码

第一种

```
module decoder_TEST

    (input[3:0] C,

    output reg[6:0] HEX);

    always @(C)

    begin

        case(C)

            4'b0000 : HEX = 7'b1000000;

            4'b0001 : HEX = 7'b1001111;

            4'b0010 : HEX = 7'b0100100;

            4'b0011 : HEX = 7'b0110000;

            4'b0100 : HEX = 7'b0011001;

            4'b0101 : HEX = 7'b0010010;

            4'b0110 : HEX = 7'b0000010;

            4'b0111 : HEX = 7'b1111000;

            4'b1000 : HEX = 7'b0000000;

            4'b1001 : HEX = 7'b0011000;

            4'b1010 : HEX = 7'b1111110;

            4'b1011 : HEX = 7'b1111101;

            4'b1100 : HEX = 7'b1111011;

            4'b1101 : HEX = 7'b1110111;

            4'b1110 : HEX = 7'b1101111;

            4'b1111 : HEX = 7'b1011111;

            default: HEX = 7'b1111111;

        endcase

    end

endmodule
```

```

module D_trigger_test

    (input Clk,R,S,D,

    output reg Q);

    always @(posedge Clk or posedge R or posedge S)

    begin

        if(R) Q<=1'b0;

        else if(S) Q<=1'b1;

        else Q<=D;

    end

endmodule

```

```

module PART5

    (input KEY0,KEY1,

    input [15:0]SW,

    output [6:0] HEX7,HEX6,HEX5,HEX4,HEX3,HEX2,HEX1,HEX0);

    reg [3:0]DEC3,DEC2,DEC1,DEC0;

    wire [3:0]DEC7,DEC6,DEC5,DEC4;

    integer i,n;

    function D_trigger;

        input Clk,R,D;

        begin

            if(!R) D_trigger=1'b0;

            else D_trigger=D;

        end

    endfunction

    always@(posedge KEY1)

```

```

begin

for(i=0;i<=3;i=i+1)

    DEC0[i] = D_trigger (KEY1,KEY0,SW[i]);

for(i=0;i<=3;i=i+1)

    DEC1[i] = D_trigger (KEY1,KEY0,SW[i+4]);

for(i=0;i<=3;i=i+1)

    DEC2[i] = D_trigger (KEY1,KEY0,SW[i+8]);

for(i=0;i<=3;i=i+1)

    DEC3[i] = D_trigger (KEY1,KEY0,SW[i+12]);


end


assign      DEC4 = SW[3:0];
assign      DEC5 = SW[7:4];
assign      DEC6 = SW[11:8];
assign      DEC7 = SW[15:12];


decoder_TEST m1 (DEC0,HEX0);
decoder_TEST m2 (DEC1,HEX1);
decoder_TEST m3 (DEC2,HEX2);
decoder_TEST m4 (DEC3,HEX3);
decoder_TEST m5 (DEC4,HEX4);
decoder_TEST m6 (DEC5,HEX5);
decoder_TEST m7 (DEC6,HEX6);
decoder_TEST m8 (DEC7,HEX7);

endmodule

```

第二种/**

* @file PART5.v

```

* @brief      lab3 PART5

* @author      Fu Zhengnan

* @date      11.20

* @copyright    Fu Zhengnan. All rights reserved.

*/

/**

* @brief      Drive digital tube.

* @param      NULL.

* @note      null.

* @retval     OUTput is the IOs of decoder.

*/

module decoder_TEST

    (input[3:0] C,
    output reg[6:0] HEX);

always @(C)
begin
    case(C)
        4'b0000 : HEX = 7'b1000000;
        4'b0001 : HEX = 7'b1001111;
        4'b0010 : HEX = 7'b0100100;
        4'b0011 : HEX = 7'b0110000;
        4'b0100 : HEX = 7'b0011001;
        4'b0101 : HEX = 7'b0010010;
        4'b0110 : HEX = 7'b0000010;
        4'b0111 : HEX = 7'b1111000;
        4'b1000 : HEX = 7'b0000000;
        4'b1001 : HEX = 7'b0011000;
        4'b1010 : HEX = 7'b1111110;
    endcase
end

```

```

        4'b1011 : HEX = 7'b1111101;

        4'b1100 : HEX = 7'b1111011;

        4'b1101 : HEX = 7'b1110111;

        4'b1110 : HEX = 7'b1101111;

        4'b1111 : HEX = 7'b1011111;

        default: HEX = 7'b1111111;

    endcase

end

endmodule

/**
 * @brief D flip-flop.
 * @param NULL.
 * @note NULL.
 * @retval The output is the D of D flip-flop.
 */

module D_trigger_test
    (input Clk,R,D,
     output reg Q);

    always @(posedge Clk or negedge R)

    begin

        if(!R) Q<=1'b0;

        else Q<=D;

    end

endmodule

/**
 * @brief Enter 16 signals and latch.
 * @param NULL.
 * @note NULL.

```



```

* @retval output is the latch signals.

*/

module D_trigger_HEX

    (input Clk,R,

     input [15:0] D,

     output [15:0]Q);

    D_trigger_test m1 (Clk,R,D[0],Q[0]);

    D_trigger_test m2 (Clk,R,D[1],Q[1]);

    D_trigger_test m3 (Clk,R,D[2],Q[2]);

    D_trigger_test m4 (Clk,R,D[3],Q[3]);

    D_trigger_test m5 (Clk,R,D[4],Q[4]);

    D_trigger_test m6 (Clk,R,D[5],Q[5]);

    D_trigger_test m7 (Clk,R,D[6],Q[6]);

    D_trigger_test m8 (Clk,R,D[7],Q[7]);

    D_trigger_test m9 (Clk,R,D[8],Q[8]);

    D_trigger_test m10 (Clk,R,D[9],Q[9]);

    D_trigger_test m11 (Clk,R,D[10],Q[10]);

    D_trigger_test m12 (Clk,R,D[11],Q[11]);

    D_trigger_test m13 (Clk,R,D[12],Q[12]);

    D_trigger_test m14 (Clk,R,D[13],Q[13]);

    D_trigger_test m15 (Clk,R,D[14],Q[14]);

    D_trigger_test m16 (Clk,R,D[15],Q[15]);

endmodule

/**

* @brief PART5 的整体函数.

* @param NULL.

* @note NULL.

* @retval 八个数码管.

```

*/

module PART5

(input KEY0,KEY1,

input [15:0]SW,

output [6:0] HEX7,HEX6,HEX5,HEX4,HEX3,HEX2,HEX1,HEX0);

wire [15:0] TEMP1,TEMP2;

D_trigger_HEX n1 (KEY1,KEY0,SW,TEMP1);

D_trigger_HEX n2 (KEY1,KEY0,TEMP1,TEMP2);

decoder_TEST m1 (TEMP1[3:0],HEX0);

decoder_TEST m2 (TEMP1[7:4],HEX1);

decoder_TEST m3 (TEMP1[11:8],HEX2);

decoder_TEST m4 (TEMP1[15:12],HEX3);

decoder_TEST m5 (TEMP2[3:0],HEX4);

decoder_TEST m6 (TEMP2[7:4],HEX5);

decoder_TEST m7 (TEMP2[11:8],HEX6);

decoder_TEST m8 (TEMP2[15:12],HEX7);

endmodule

六 实验体会

有时候达到一个目标可以有多种方法，本实验第一次走了弯路，导致做好的比较晚，最后问了别的组的思路才找到简单方法。

