数电实验exp02_report

数电实验 二班 杨志斌 **181860126** 实验时间：**2019年9月12日**

Email：2500223255@qq.com

实验目的：

> 实现一个 8-3 优先编码器，使用拨动开关，SW7—SW0 随机输入一个 8 位二进制值，对此 8 位二进制数进行高位优先编码成一个 3 位二进制值，并根据是否有输入增加一位输入指示位，即 8 个开关全 0 时指示位为 0，有任何一个开关为 1 时指示位为 1。将此编码结果以二进制形式显示在四个发光二极管上。再将此结果跟据七段数码管的显示进行译码，将二进制的优先编码结果以十进制的形式显示在数码管上。编码器的使能端可选实现。

实验原理：

> 编码器的实现原理比较简单，通过对输入的选择决定对应的正确输出即可。根据Guide里面的提示我们主要选择casez作为编码的主体逻辑实现部分，而使能端的实现则体现在if语句开启casez的判断，利用casez优先判断的原理，我们可以很轻松的实现优先编码。

实验环境及实验器材：

> 软件环境：Quartus (Quartus Prime 17.1) Lite Edition
>
> 硬件环境：DE10开发平台
>
> FPGA芯片，Cyclone V 5CSXFC6D6F31C6

设计代码：

```
// 编码器逻辑主体部分代码
// Created by Yang Zhibin on 2019-9-11

module encode83(x, en, y, z);
    input [7:0] x;
    input en;
    output reg [3:0] y;
    output reg [6:0] z;

    always @ (en or x) begin
```

```
        y = 4'b0000;
        z[6:0] = 7'b1111111;
        if(en) begin
            casez (x)
                8'b1zzzzzzz : begin
                    y[3:0] = 4'b1111;
                    z[6:0] = 7'b1111000;
                    end
                8'b01zzzzzz : begin
                    y[3:0] = 4'b1110;
                    z[6:0] = 7'b0000010;
                    end
                8'b001zzzzz : begin
                    y[3:0] = 4'b1101;
                    z[6:0] = 7'b0010010;
                    end
                8'b0001zzzz : begin
                    y[3:0] = 4'b1100;
                    z[6:0] = 7'b0011001;
                    end
                8'b00001zzz : begin
                    y[3:0] = 4'b1011;
                    z[6:0] = 7'b0110000;
                    end
                8'b000001zz : begin
                    y[3:0] = 4'b1010;
                    z[6:0] = 7'b0100100;
                    end
                8'b0000001z : begin
                    y[3:0] = 4'b1001;
                    z[6:0] = 7'b1111001;
                    end
                8'b00000001 : begin
                    y[3:0] = 4'b1000;
                    z[6:0] = 7'b1000000;
                    end
                default : begin
                    y[3:0] = 4'b0000;
                    z[6:0] = 7'b1111111;
                    end
            endcase
        end
        else begin
            y[3:0] = 4'b0000;
            z[6:0] = 7'b1111111;
        end
    end

endmodule

// 顶层调用代码

//  This code is generated by Terasic System Builder
```

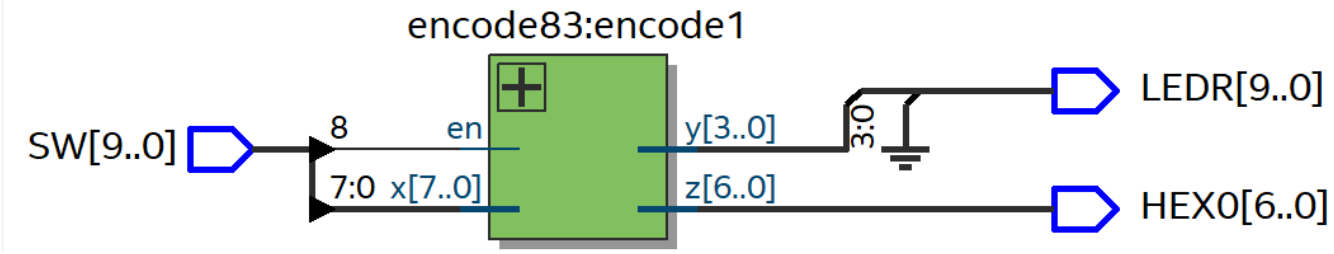
```
64
65  module exp02(
66      input             [9:0]       SW,
67
68      output            [9:0]       LEDR,
69
70      output            [6:0]       HEX0,
71      output            [6:0]       HEX1,
72      output            [6:0]       HEX2,
73      output            [6:0]       HEX3,
74      output            [6:0]       HEX4,
75      output            [6:0]       HEX5
76  );
77
78  encode83 encode1(SW[7:0], SW[8], LEDR[3:0], HEX0[6:0]);
79
80  endmodule
```

仿真代码及其设计思路：

```
1   `timescale 10 ns/ 1 ps
2   module exp02_vlg_tst();
3   reg eachvec;
4   reg [9:0] SW;
5   wire [6:0]  HEX0;
6   wire [9:0]  LEDR;
7   exp02 i1 (
8       .HEX0(HEX0),
9       .LEDR(LEDR),
10      .SW(SW)
11  );
12  initial
13  begin
14
15      SW[8] = 1'b0; SW[7:0] = 8'b01101000; #10;
16                    SW[7:0] = 8'b01001101; #10;
17      SW[8] = 1'b1; SW[7:0] = 8'b10101001; #10;
18                    SW[7:0] = 8'b01010010; #10;
19                    SW[7:0] = 8'b00101010; #10;
20                    SW[7:0] = 8'b00011011; #10;
21                    SW[7:0] = 8'b00001011; #10;
22                    SW[7:0] = 8'b00000101; #10;
23                    SW[7:0] = 8'b00000010; #10;
24                    SW[7:0] = 8'b00000001; #10;
25                    SW[7:0] = 8'b00000000; #10;
26  end
```

> 设计思路：对于这个编码器的测试情况来说，只存在八种有效情况，分别对应输入有效位0~7，其他情况均会被优先编码规则所覆盖，所以我们在仿真测试代码中考虑使能端为0时（即SW[8] = 0时）输入任意的两种情况，输出结果，和使能端为1，有效输入分别为0~7的八种情况，由于实验要求设置一位信号位，所以我们在测试时，额外设置一次输入全为0（即没有输入）的情况体现信号位为0

实验过程与步骤：

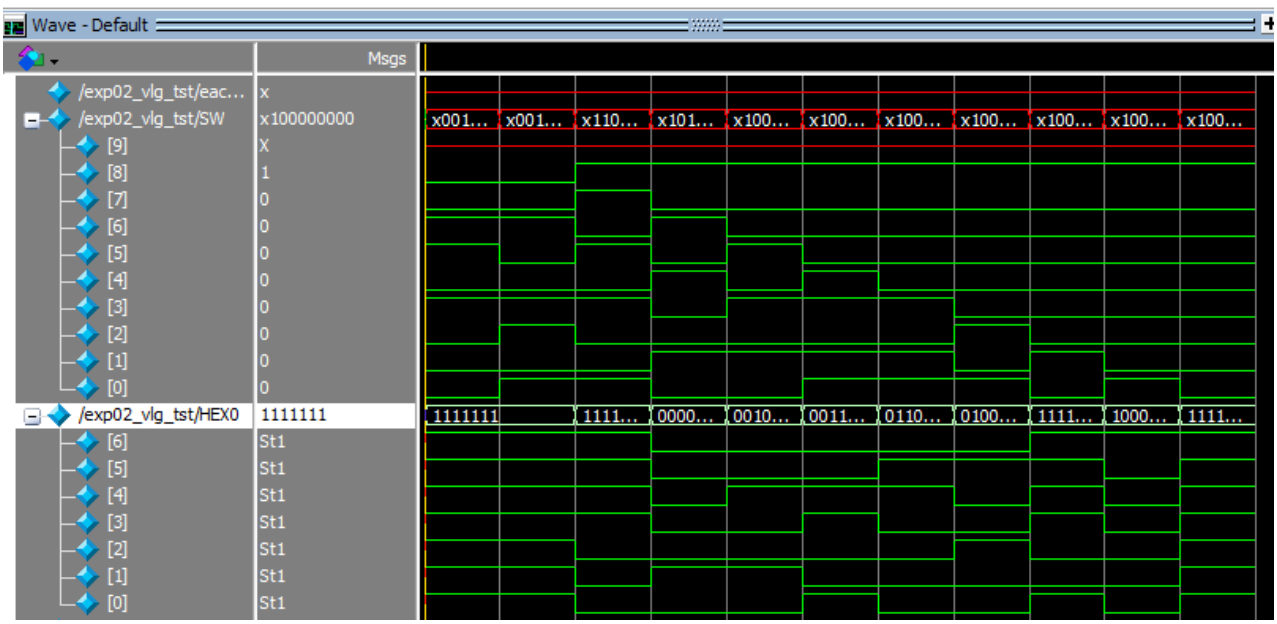


> 如上图，即为利用设计好的83优先编码器连接输入开关和LED灯，七段数码管的过程。八三编码器的内部实现，由于逻辑上使用的是casez语句，所以线路较长，图片略过。

实验结果及仿真截图：

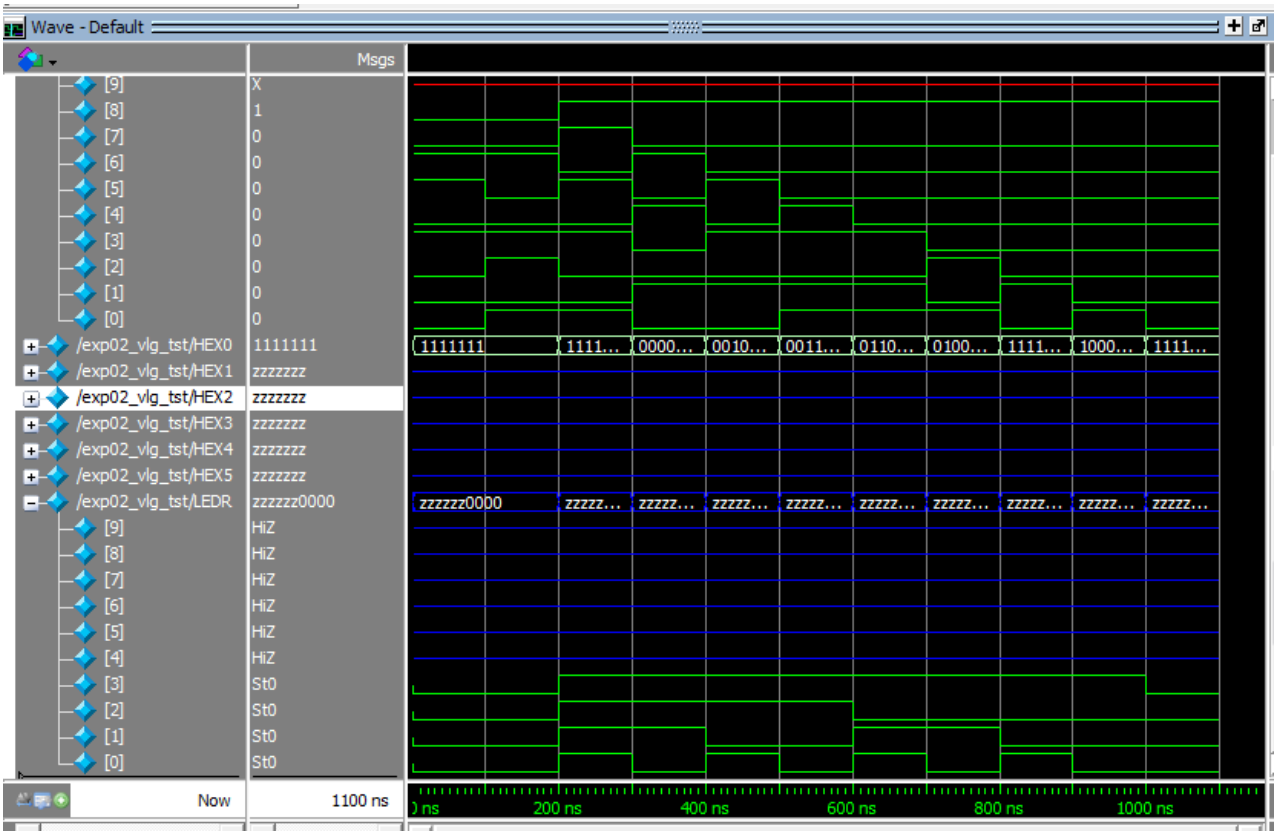
由于结果较长，所以将数码管和LED灯的结果分开绘制：

- 开关与数码管：



> 可以看到使能端为0 的前两段和使能端为1的后面九段，结果是正确的，图像不是很明显，验收时可以看出数码管的数字显示更加直观。

- 开关与LED灯：



> 可以看到，随着优先编码的递减（从使能端为1时开始讨论），LED灯（LEDR [2：0] ）从111递减到000，这期间信号灯一直是1（因为一直有输入），而当输入的SW均为0时，LED灯依然是000，但是信号灯变为0表示没有输入。

总仿真波形图和电路板的实际测试来看，本次实验是成功的，并且完成了使能端的选做任务。

实验遇到问题及解决：

> 初次尝试的时候使用了for循环的方式，来建立83编码器的主体逻辑部分，但是测试的时候没有成功，所有信号一直是1，由于我忘记保留失败的文件了，直接把那个文件删除重写，所以现在只能猜测一下原因：
>
> - 可能是对波形图没有使用zomm full，导致看到的只是局部直线（这种可能性不大，因为我应该可以反映出来）
> - 语法逻辑的实现不符合优先译码器的逻辑实现，可能导致测试时输入输出有问题（有这种可能，但是for循环的部分我是改编自Guide手册，所以出问题的可能性也比较小
> - 因为源文件丢失，所以也给我提醒，即使是错误的文件也有保留的价值，可以用来复盘，总结失误的原因，所以下次再出现这种情况，我一定不会再犯这样的错误
>
> 在仿真测试部分遇到很诡异的情况，无论如何调试，都报错有undefined veriables： sw，但是我的调试代码是由simulation自动生成的，怎么可能没有定义呢，后来我看了很久才发现，原来是大小写的问题，这编辑器的字体对SW和sw的区分太不明显了，于是我又百度修改了合适的字体。

以上就是全部问题及解决，实际上，第一个问题的解决是依赖于修改了代码实现。

启发和建议：

> 在数电实验的过程中要心平气和，不能毛毛躁躁，不能因为写了好久就越来越不想写，绝大多数时候出了问题都是人为原因导致的，机器是诚实的。其实不光数电实验如此，其他的编程逻辑设计都是这样，要对自己有自信，但不能盲目的认为自己写的东西没有一点问题，要勇于修改，勇于发现自己的问题。

实验报告完成于 2019年9月12日，学生：杨志斌