

实验名称：实验二 译码器和编码器的设计

姓名：张涵之

学号：191220154

班级：周一 5-6

邮箱：191220154@smail.nju.edu.cn

实验时间：2020/9/16

2.2.2 优先编码器

```
module exp2_1(
    //////////////// SW ////////////////
    input      [4:0] SW,
    //////////////// LED ////////////////
    output     [1:0] LEDR
);

//=====
// REG/WIRE declarations
//=====

encode42 code(SW[4:1],SW[0],LEDR[1:0]);

//=====
// structural coding
//=====

endmodule

initial
begin
    // code that executes only once
    // insert code here --> begin
    SW[0]=1'b0; SW[4:1] =4'b0011; #10;
    SW[4:1] =4'b0101; #10;
    SW[4:1] =4'b1010; #10;
    SW[4:1] =4'b1100; #10;
    SW[4:1] =4'b1111; #10;
    SW[0]=1'b1; SW[4:1] =4'b0011; #10;
    SW[4:1] =4'b0101; #10;
    SW[4:1] =4'b1010; #10;
    SW[4:1] =4'b1100; #10;
    SW[4:1] =4'b1111; #10;
    // --> end
    // $display("Running testbench");
end
```

00110	01010	10100	11000	11110	00111	01011	10101	11001	11111
00					01	10	11		

实验目的：实现一个 8-3 优先编码器。

实验原理：使用拨动开关，SW0 为使能端，SW8—SW1 随机输入一个 8 位二进制值。

对此 8 位二进制数采用 for 循环进行高位优先编码，生成一个 3 位二进制值。

显示在 LEDR3—LEDR1 上。

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，显示在 LEDR0 上。

再采用 case 语句根据七段数码管的显示进行译码得到一个 7 位二进制值，

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。

实验环境/器材：实验箱一个，笔记本电脑一台。

程序代码或流程图：

8 位二进制值—高位优先译码—>3 位二进制值—译码—>七段数码管数字显示。

```

module exp2_2(
    // SW
    input [8:0] SW,
    // LED
    output [3:0] LEDR,
    // Seg7
    output [6:0] HEX0
);

// =====
// REG/WIRE declarations
// =====

encode83 code(SW[8:1],SW[0],LEDR[3:1],LEDR[0],HEX0);

module encode83(x,en,y,s,hex);
    input [7:0]x;
    input en;
    output reg [2:0]y;
    output reg s;
    output reg [6:0]hex;
    integer i;

    always @(x or en) begin
        if (en) begin
            y = 0;
            for( i = 0; i <= 7; i = i+1)
                if(x[i] == 1) y = i;
            end
        else y = 0;
        if (en == 0 | x == 0) s = 0;
        else s = 1;
        case (y)
            3'b000 : hex = 7'b1000000;
            3'b001 : hex = 7'b1111001;
            3'b010 : hex = 7'b0100100;
            3'b011 : hex = 7'b0110000;
            3'b100 : hex = 7'b0011001;
            3'b101 : hex = 7'b0010010;
            3'b110 : hex = 7'b0000010;
            3'b111 : hex = 7'b1111000;
            default: hex = 7'b1111111;
        endcase
    end
endmodule

```

实验步骤/过程:

对 8 位二进制数采用 for 循环进行高位优先编码，生成一个 3 位二进制值。

对 3 位二进制数采用 case 语句进行译码，生成一个 7 位二进制值。

通过观察七段数码管对不同数字的显示，设计不同情况下的 7 位二进制值。

测试方法:

```

initial
begin
    // code that executes only once
    // insert code here --> begin
    SW[0]=1'b0; SW[8:1]=8'b00000000; #10;
    SW[8:1]=8'b00000011; #10;
    SW[8:1]=8'b00001111; #10;
    SW[8:1]=8'b00111011; #10;
    SW[8:1]=8'b10000000; #10;
    SW[0]=1'b1; SW[8:1]=8'b00000000; #10;
    SW[8:1]=8'b00000001; #10;
    SW[8:1]=8'b00000011; #10;
    SW[8:1]=8'b00000101; #10;
    SW[8:1]=8'b00001111; #10;
    SW[8:1]=8'b00010010; #10;
    SW[8:1]=8'b00111011; #10;
    SW[8:1]=8'b01100000; #10;
    SW[8:1]=8'b10000000; #10;

    // --> end
    // $display("Running testbench");
end

```

使能端分为 0,1 两种情况，X0~X7 分别覆盖了 8 种不同的优先级。

实验结果：

00000000	000000110	000011110	001110110	100000000	000000001	000000011
1000000						
0000						0001

000000111	000001011	000011111	000100101	001110111	011000001	100000001
1111001	0100100	0110000	0011001	0010010	0000010	1111000
0011	0101	0111	1001	1011	1101	1111

EN 输入为 0 时，无论 X0~X7 输入什么，S==0，Y==0，七段数码管显示 0。

EN 输入为 1 时，X0~X7 输入与 S 和 Y 的输出符合预期。

经接入实验箱检验，七段数码管的数字显示也符合预期。

实验中遇到的问题及解决办法：

本次实验非常顺利，没有遇到任何问题。

实验得到的启示：好玩。

意见和建议：无。

思考：了解 casex 和 casez 语句的使用，思考如何用 casex 语句设计优先编码器。

casez：出现在表达式的值为 z 的位被认为是无关位，不进行比较。

casex：出现在表达式的值为 z 和 x 的位都被认为是无关位，不进行比较。

语句的优先级同 case 语句，条件分支靠前的优先级高。

对程序代码（encode83）进行如下修改：

```

always @(x or en) begin
    if (en) begin
        //y = 0;
        //for( i = 0; i <= 7; i = i+1)
        // if(x[i] == 1) y = i;
        casex (x)
            8'b00000000 : y = 0;
            8'b00000001? : y = 1;
            8'b0000001?? : y = 2;
            8'b000001??? : y = 3;
            8'b00001???? : y = 4;
            8'b0001????? : y = 5;
            8'b001?????? : y = 6;
            8'b01??????? : y = 7;
            default      : y = 0;
        endcase
    end
end

```

再次进行仿真和实体实验，结果不变：

000...	000...	000...	001...	100...	000...	000...	000...	000...	000...	000...	001...	011...	100...
1000000							111...	010...	011...	001...	001...	000...	111...
0000						0001	0011	0101	0111	1001	1011	1101	1111