

数电实验EXP2

刘永鹏 191220070 计科 693901492@qq.com 2020.9.14


1. 实验目的

实现一个 8-3 优先编码器

查找 8-3 优先编码器相关原理和实现方法，实现一个 8-3 编码器，完成 8-3 编码器的设计、功能仿真和硬件实现。

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。

2. 实验原理

1. 编码器原理：根据 2^k 位输入，将其转化位10进制输出，即译码器的逆过程。
2. 高位优先原理：允许 2^k 输入中存在多个1（有效位），但只处理所有1中所在位最高的。
3. casex的应用：verilog提供了casex, casez供编程者更方便地处理未知态，高阻态。在本实验中，**低于最高位1的所有位都可视为无关态**，因此应用casex可以简化代码。
4. image-20200914185813769

3. 实验环境

1. input: [9:0]SW
2. output: [9:0]LEDR
3. output: [6:0]HEX0

使用Systembuilder自动生成，存在多余的接口。

4. 程序代码/流程图

首先是always-casex语句

```
always @ (SW)
begin
    casex(SW)
        10'bxx1xxxxxxx: begin LEDR = 10'd7; HEX0 = 7'b1111000; end
        10'bxx01xxxxxxx: begin LEDR = 10'd6; HEX0 = 7'b0000010; end
        10'bxx001xxxxxx: begin LEDR = 10'd5; HEX0 = 7'b0010010; end
        10'bxx0001xxxxx: begin LEDR = 10'd4; HEX0 = 7'b0011001; end
        10'bxx00001xxxx: begin LEDR = 10'd3; HEX0 = 7'b0110000; end
        10'bxx000001xx: begin LEDR = 10'd2; HEX0 = 7'b0100100; end
        10'bxx0000001x: begin LEDR = 10'd1; HEX0 = 7'b1111001; end
        default: begin LEDR = 10'd0; HEX0 = 7'b1000000; end
    endcase
end
```

这里使用**casex**简化代码，完成高位优先的逻辑设计。七段数码管则参考引脚图完成赋值。注意七段数码管为**低电平有效**。

输入为10位，两个无效位设为x。

然后是激励代码。

```
begin
// code executes for every event on sensitivity list
// insert code here --> begin
SW = 10'b0011111111; #10;
SW = 10'b0001111111; #10;
SW = 10'b0000111111; #10;
SW = 10'b0000011111; #10;
SW = 10'b0000001111; #10;
SW = 10'b0000000111; #10;
SW = 10'b0000000011; #10;
SW = 10'b0000000001; #10;
SW = 10'b0000000000; #10;
SW = 10'b0010000000; #10;
SW = 10'b0001000000; #10;
SW = 10'b0000100000; #10;
SW = 10'b0000010000; #10;
SW = 10'b0000001000; #10;
SW = 10'b0000000100; #10;
SW = 10'b0000000010; #10;
SW = 10'b0000000001; #10;
SW = 10'b0000000000; #10;
@eachvec;
// --> end
end
```

这里分别测试了该编码器的普通编码功能以及其优先编码功能。将两个无效位设为始终为0。

5. 实验过程

1. Systembuilder自动建立项目。
2. 查找七段数码管的引脚图。
3. 编写程序代码。
4. 编译通过后自动生成Testbench。
5. 连接Testbench, RTL仿真。
6. 仿真结果无误后烧制代码在板上再次验证。

6. 测试方法

使用testbench对结果在RTL仿真中初步测试。因为modelsim中无法直接检验七段数码管（或许有？但我在网上没有查到），因此在开发板上进一步对七段数码管的显示进行测试。

最终测试结果与预期相符。

7. 实验结果



8. 问题与解决方案

1. 七段数码管为**低电平有效**，需要特别注明。
2. 生成项目中勾选“七段数码管”项，在systembuilder建立项目时为其自动分配引脚。

9. 启示

本实验中，casex可以显著地较少代码量和bug出现的概率，在编写前应当根据verilog语言特性选择最合适的语法。

10. 意见与建议

暂无。