

《数字逻辑电路》实验报告

第 2 次实验： 译码/编码器实验

姓名： 凌嘉伟

学号： 151220061

15 级 计算机科学与技术系 5 班

邮箱： 151220061@smail.nju.edu.cn

时间： 2016.9.19

一、实验目的

复习译码/编码器知识，学习 Verilog HDL 中 case 语句的使用，学习译码器的设计；学习 N4 开板上七段数码管的使用。完成编码器的设计。

译码器和编码器是数字系统中的常用电路，也是组合逻辑电路中的主要组成元件之一。本实验首先介绍常用的译码器和编码器的设计方法以及七段数码管的使用。本实验还将介绍 Verilog 语言中 for 循环的使用。最后，自行设计一个 4-16 译码器及七段数码管显示。

二、实验原理（背景知识）

译码器也是组合逻辑电路的一个重要器件，译码器是将某一输入信息转换为某一特定输出的逻辑电路，通常是多路输入/输出电路，它将 n 位的输入编码转换为 m 位的编码输出，一般情况下 $n < m$ ，输入编码和输出编码之间存在着一一对应的映射关系，每个输入编码产生唯一的一个不同于其他的输出编码。

编码器是一种与译码器功能相反的逻辑电路，编码器的输出编码比其输入编码位数少。常用的二进制编码器把来自于 2^n 条输入线的信息编码转换成 n 位二进制码。

七段 LED 数码管是一种常用的显示元件，常应用于手表、计算器等仪器中，用于显示十进制数值。七段显示器采用七段译码，一般情况下，其输入为 4 位的 BCD 码，输出为 7 位的编码，用于驱动七段显示器的不同位，以显示出不同的数字。

三、实验器材/环境

实验器材：实验室配置电脑

环境：win 7，vivado 2015.2

四、实验设计思路（验收实验）

1、4-16 译码器：

实验设计代码：

```
module decode4_16(x,en,y);
input [3:0] x;
input en;
output reg [15:0]y;
integer i;
always @(x or en)
if (en)
begin
for( i = 0; i <= 15; i = i+1)
if(x == i) y[i] = 1; else y[i] = 0;
end
else y = 16'b0000000000000000;
endmodule
```

激励代码：

```

`timescale 10 ns/ 1 ps
module test_decode416();
reg [3:0] x;
reg en;
wire [15:0]y;
decode416 i1( .x(x),.en(en),.y(y));

initial begin
en =1'b1; x =4'b0000; #10;
        x =4'b0001; #10;
        x =4'b0010; #10;
        x =4'b0011; #10;
        x =4'b0100; #10;
        x =4'b0101; #10;
        x =4'b0110; #10;
        x =4'b0111; #10;
        x =4'b1000; #10;
        x =4'b1001; #10;
        x =4'b1010; #10;
        x =4'b1011; #10;
        x =4'b1100; #10;
        x =4'b1101; #10;
        x =4'b1110; #10;
        x =4'b1111; #10;
en =1'b0; x =4'b0000; #10;
        x =4'b0001; #10;
        x =4'b0010; #10;
        x =4'b0011; #10;
        x =4'b0100; #10;
        x =4'b0101; #10;
        x =4'b0110; #10;
        x =4'b0111; #10;
        x =4'b1000; #10;
        x =4'b1001; #10;
        x =4'b1010; #10;
        x =4'b1011; #10;
        x =4'b1100; #10;
        x =4'b1101; #10;
        x =4'b1110; #10;
        x =4'b1111; #10;

        end
endmodule

```

2、一位数码管输出

实验设计代码:

```
module seg7(  
    input btnD,// 使能端  
    input[15:0] sw,// 16 位二进制输入端  
    output reg[3:0] led,// 编码输出端  
    output [7:0] sel_seg,// 数码管选择端, 选择点亮某个数码管  
    output reg[6:0] seg );// 每个七段数码管的各段控制端  
assign sel_seg =8'b11111110;  
always@(*)  
    if(btnD)begin  
        case(sw)  
            16'h0001:begin led=4'b0000; seg=7'b1000000;end  
            16'h0002:begin led=4'b0001; seg=7'b1111001;end  
            16'h0004:begin led=4'b0010; seg=7'b0100100;end  
            16'h0008:begin led=4'b0011; seg=7'b0110000;end  
            16'h0010:begin led=4'b0100; seg=7'b0011001;end  
            16'h0020:begin led=4'b0101; seg=7'b0010010;end  
            16'h0040:begin led=4'b0110; seg=7'b0000010;end  
            16'h0080:begin led=4'b0111; seg=7'b1111000;end  
            16'h0100:begin led=4'b1000; seg=7'b0000000;end  
            16'h0200:begin led=4'b1001; seg=7'b0010000;end  
            16'h0400:begin led=4'b1010; seg=7'b0001000;end  
            16'h0800:begin led=4'b1011; seg=7'b0000011;end  
            16'h1000:begin led=4'b1100; seg=7'b1000110;end  
            16'h2000:begin led=4'b1101; seg=7'b0100001;end  
            16'h4000:begin led=4'b1110; seg=7'b0000110;end  
            16'h8000:begin led=4'b1111; seg=7'b0001110;end  
            default:begin led=4'b0000; seg=7'b1111111;end  
        endcase  
    end  
    else begin led=4'b0000; seg=7'b1111111;end  
endmodule
```

测试激励代码:

```
module test_seg7();  
    reg[15:0] sw;  
    reg btnD;  
    wire[3:0] led;  
    wire[7:0] sel_seg;  
    wire[6:0] seg;  
    seg7 i1(.sw(sw),.btnD(btnD),.led(led),.sel_seg(sel_seg)  
        ,.seg(seg));  
    initial begin
```

```

    btnD =1'b1;
        sw =16'h0001;#10;
        sw =16'h0002;#10;
        sw =16'h0004;#10;
        sw =16'h0008;#10;
        sw =16'h0010;#10;
        sw =16'h0020;#10;
        sw =16'h0040;#10;
        sw =16'h0080;#10;
        sw =16'h0100;#10;
        sw =16'h0200;#10;
        sw =16'h0400;#10;
        sw =16'h0800;#10;
        sw =16'h1000;#10;
        sw =16'h2000;#10;
        sw =16'h4000;#10;
        sw =16'h8000;#10;
    btnD =1'b0;
        sw =16'h0001;#10;
        sw =16'h0002;#10;
        sw =16'h0004;#10;
        sw =16'h0008;#10;
        sw =16'h0010;#10;
        sw =16'h0020;#10;
        sw =16'h0040;#10;
        sw =16'h0080;#10;
        sw =16'h0100;#10;
        sw =16'h0200;#10;
        sw =16'h0400;#10;
        sw =16'h0800;#10;
        sw =16'h1000;#10;
        sw =16'h2000;#10;
        sw =16'h4000;#10;
        sw =16'h8000;#10;
    end
endmodule

```

引脚配置文件:

```

set_property PACKAGE_PIN P18 [get_ports btnD]

set_property PACKAGE_PIN J15 [get_ports sw[0]]
set_property PACKAGE_PIN L16 [get_ports sw[1]]
set_property PACKAGE_PIN M13 [get_ports sw[2]]
set_property PACKAGE_PIN R15 [get_ports sw[3]]

```

set_property PACKAGE_PIN R17 [get_ports sw[4]]
set_property PACKAGE_PIN T18 [get_ports sw[5]]
set_property PACKAGE_PIN U18 [get_ports sw[6]]
set_property PACKAGE_PIN R13 [get_ports sw[7]]
set_property PACKAGE_PIN T8 [get_ports sw[8]]
set_property PACKAGE_PIN U8 [get_ports sw[9]]
set_property PACKAGE_PIN R16 [get_ports sw[10]]
set_property PACKAGE_PIN T13 [get_ports sw[11]]
set_property PACKAGE_PIN H6 [get_ports sw[12]]
set_property PACKAGE_PIN U12 [get_ports sw[13]]
set_property PACKAGE_PIN U11 [get_ports sw[14]]
set_property PACKAGE_PIN V10 [get_ports sw[15]]

set_property PACKAGE_PIN H17 [get_ports led[0]]
set_property PACKAGE_PIN K15 [get_ports led[1]]
set_property PACKAGE_PIN J13 [get_ports led[2]]
set_property PACKAGE_PIN N14 [get_ports led[3]]

set_property PACKAGE_PIN J17 [get_ports sel_seg[0]]
set_property PACKAGE_PIN J18 [get_ports sel_seg[1]]
set_property PACKAGE_PIN T9 [get_ports sel_seg[2]]
set_property PACKAGE_PIN J14 [get_ports sel_seg[3]]
set_property PACKAGE_PIN P14 [get_ports sel_seg[4]]
set_property PACKAGE_PIN T14 [get_ports sel_seg[5]]
set_property PACKAGE_PIN K2 [get_ports sel_seg[6]]
set_property PACKAGE_PIN U13 [get_ports sel_seg[7]]

set_property PACKAGE_PIN T10 [get_ports seg[0]]
set_property PACKAGE_PIN R10 [get_ports seg[1]]
set_property PACKAGE_PIN K16 [get_ports seg[2]]
set_property PACKAGE_PIN K13 [get_ports seg[3]]
set_property PACKAGE_PIN P15 [get_ports seg[4]]
set_property PACKAGE_PIN T11 [get_ports seg[5]]
set_property PACKAGE_PIN L18 [get_ports seg[6]]

set_property IOSTANDARD LVCMOS33 [get_ports btnD]

set_property IOSTANDARD LVCMOS33 [get_ports sw[0]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[1]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[2]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[3]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[4]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[5]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[6]]

```
set_property IOSTANDARD LVCMOS33 [get_ports sw[7]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[8]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[9]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[10]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[11]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[12]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[13]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[14]]
set_property IOSTANDARD LVCMOS33 [get_ports sw[15]]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports led[0]]
set_property IOSTANDARD LVCMOS33 [get_ports led[1]]
set_property IOSTANDARD LVCMOS33 [get_ports led[2]]
set_property IOSTANDARD LVCMOS33 [get_ports led[3]]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[0]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[1]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[2]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[3]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[4]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[5]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[6]]
set_property IOSTANDARD LVCMOS33 [get_ports sel_seg[7]]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports seg[0]]
set_property IOSTANDARD LVCMOS33 [get_ports seg[1]]
set_property IOSTANDARD LVCMOS33 [get_ports seg[2]]
set_property IOSTANDARD LVCMOS33 [get_ports seg[3]]
set_property IOSTANDARD LVCMOS33 [get_ports seg[4]]
set_property IOSTANDARD LVCMOS33 [get_ports seg[5]]
set_property IOSTANDARD LVCMOS33 [get_ports seg[6]]
```

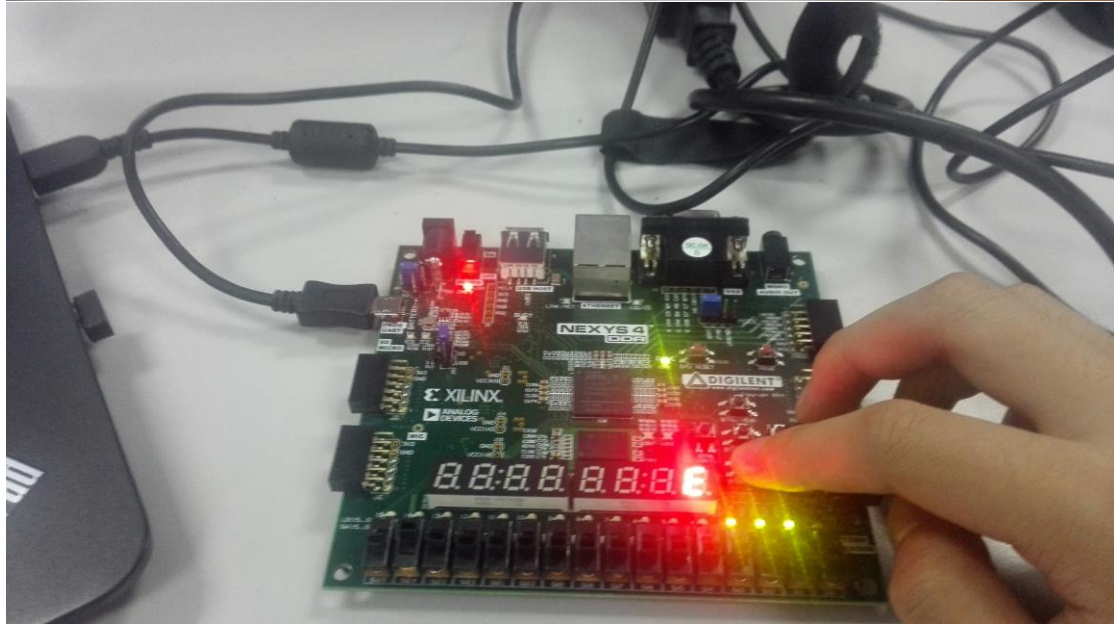
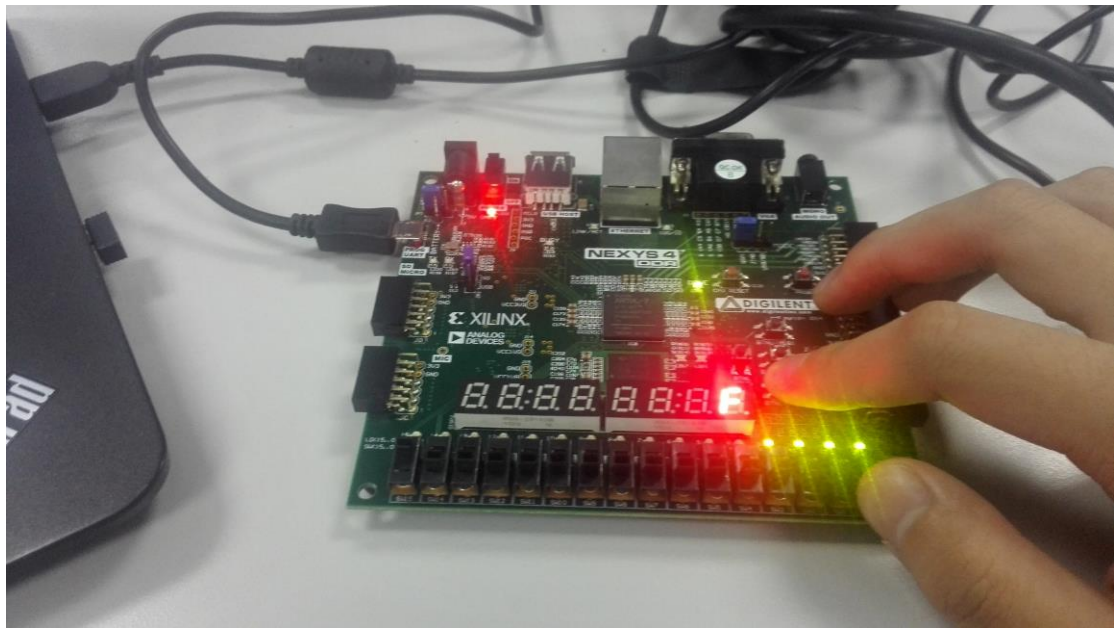
五、实验的测试序列或验证方法

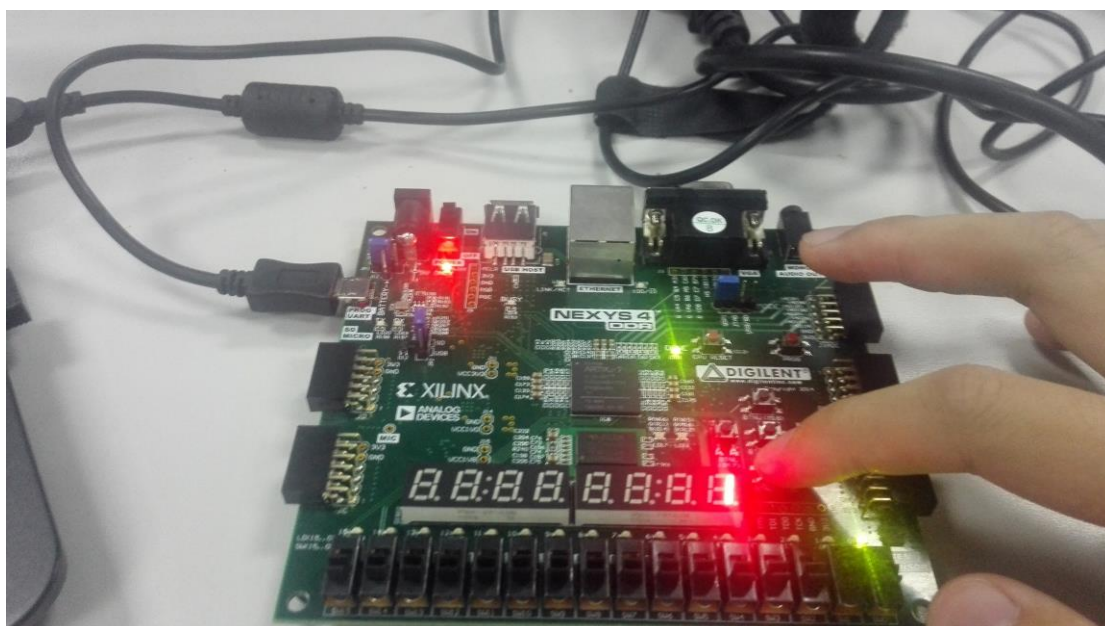
按下开关，启动使能端，即可看到相应的七段数码显示器上显示器的结果和 LED 灯的显示结果

六、实验过程（主要指验收实验）

设计实验流程，编写设计源代码，激励测试代码和引脚配置文件，将开发平台连接至电脑上进行操作

照片如下：





七、实验结果、结论等

本次实验实现了一位数码管的输出,并运用了 for 循环语句、case 语句设计相应功能的器件,熟悉了常用的译码器和编码器的设计方法以及七段数码管的使用。学习了译码器和编码器相关知识,提高了知识水平。

八、实验中遇到的问题及解决方案

遇到的问题: 相应的引脚配置位置未知

解决方案: 本科教学支撑平台已经上传了有关资料, 自行查阅即可

九、实验的启示/意见和建议

- 1、意见建议: 实验难度偏高, 实验时间安排较为紧凑
- 2、可以实现多个数码管的显示
- 3、可以实现比赛计分的显示器, 并实际应用

十、实验时间

- 1 预习时间约 1 小时—1.5 小时
- 2 实验课后花费时间约 1 小时。