数字逻辑与计算机组成实验

（组合逻辑电路设计）lab02:

译码器和编码器的设计

姓名：郑凯琳

学号：205220025

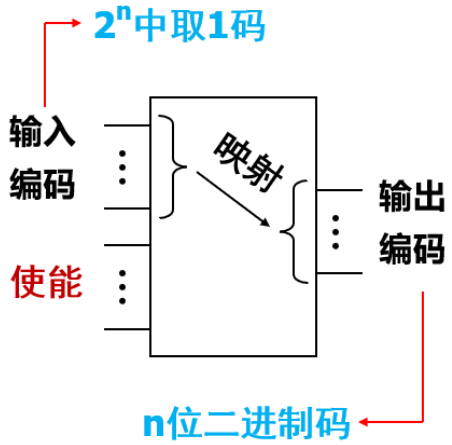
邮箱：205220025@smail.nju.edu.cn

（一） 实验目的

实现一个 8-3 优先编码器并在七段数码管上显示

（二） 实验原理

编码器 encoder：输出是输入信号的二进制编码



2^n-n：

2^n 个输入端；

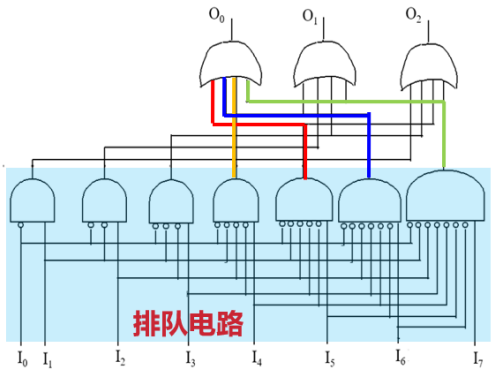
n 个输输出端

优先编码器：

- 多个输入可同时为 1，但只对优先级最高的输入进行编码输出

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	O_0	O_1	O_2
1	x	x	x	x	x	x	x	0	0	0
0	1	x	x	x	x	x	x	0	0	1
0	0	1	x	x	x	x	x	0	1	0
0	0	0	1	x	x	x	x	0	1	1
0	0	0	0	1	x	x	x	1	0	0
0	0	0	0	0	1	x	x	1	0	1
0	0	0	0	0	0	1	x	1	1	0
0	0	0	0	0	0	0	1	1	1	1

优先权编码器
的功能描述



优先权编码器逻辑电路图

（三） 实验环境/器材等

硬件器材：Nexys A7-100T 开发板

软件平台：Vivado 开发平台

（四） 实验过程

设计代码:

```
module lab02(
    input [7:0] X,
    input en,
    output reg valid,
    output [6:0] F,
    output reg [7:0] AN,
    output reg [2:0] led
    );

    reg [6:0] F;

always @ (en or X) begin
if(!en)
    begin
    F <= 7'b1111111;
    valid = 0;
    end
else if(X[7] == 1)
    begin
    F <= 7'b1111000;
    valid = 1;
    led = 3'b111;
    AN = 8'b11111110;
    end
else if(X[6] == 1)
    begin
    F <= 7'b0000010;
    valid = 1;
    led = 3'b110;
    AN = 8'b11111110;
    end
else if(X[5] == 1)
    begin
    F <= 7'b0010010;
    valid = 1;
    led = 3'b101;
    AN = 8'b11111110;
    end
else if(X[4] == 1)
    begin
    F <= 7'b0011001;
    valid = 1;
    led = 3'b100;
    AN = 8'b11111110;
    end
else if(X[3] == 1)
    begin
    F <= 7'b0110000;
    valid = 1;
    led = 3'b011;
    AN = 8'b11111110;
    end
else if(X[2] == 1)
    begin
    F <= 7'b0100100;
    valid = 1;
    led = 3'b010;
    AN = 8'b11111110;
    end
else if(X[1] == 1)
    begin
    F <= 7'b1111001;
    valid = 1;
    led = 3'b001;
    AN = 8'b11111110;
    end
else if(X[0] == 1)
    begin
    F <= 7'b1000000;
    valid = 1;
    led = 3'b000;
    AN = 8'b11111110;
    end
    else
        begin
        F <= 7'b1111111;
        valid = 0;
        AN = 8'b11111110;
        end
    end

endmodule
```

测试代码:

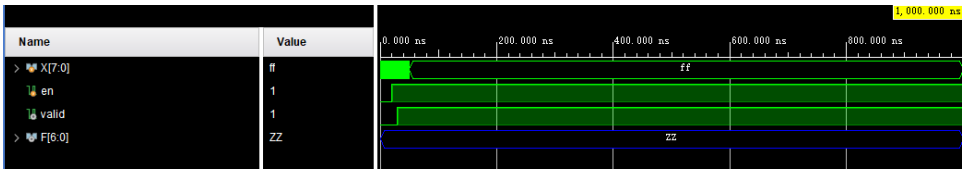
```
module lab02_test(

    );
    reg [7:0] X;
    reg en;
    wire valid;
    wire [6:0] F;

    lab02 t1(
        .X(X),
        .en(en),
        .valid(valid));

    initial
    begin
        en = 1'b0; X = 8'b00000000; #10;
                   X = 8'b11111111; #10;
        en = 1'b1; X = 8'b00000000; #10;
                   X = 8'b00001111; #10;
                   X = 8'b10000000; #10;
                   X = 8'b11111111; #10;
    end
endmodule
```

硬件实现（引脚分配）:

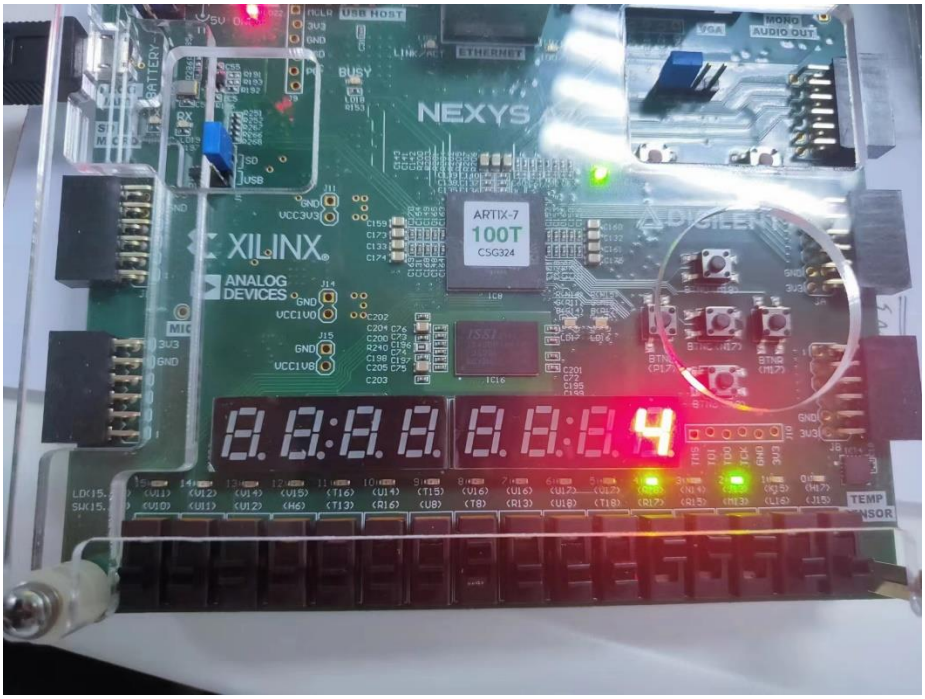
```
##Switches
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { X[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { X[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { X[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { X[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { X[4] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { X[5] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { X[6] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { X[7] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { en }]; #IO_L24N_T3_34 Sch=sw[8]
#set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports { SW[9] }]; #IO_25_34 Sch=sw[9]
#set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports { SW[10] }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
#set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports { SW[11] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
#set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports { SW[12] }]; #IO_L24P_T3_35 Sch=sw[12]
#set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { SW[13] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
#set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
#set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { SW[15] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

## LEDs
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { led[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { led[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { led[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
#set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { valid }]; #IO_L7P_T1_D09_14 Sch=led[4]
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { LED[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { LED[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
#set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { LED[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { LED[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
#set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { LED[11] }]; #IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { LED[12] }]; #IO_L16P_T2_CSI_B_14 Sch=led[12]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { LED[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
#set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { LED[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
#set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { LED[15] }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]

##7 segment display
set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { F[0] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { F[1] }]; #IO_25_14 Sch=cb
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { F[2] }]; #IO_25_15 Sch=cc
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { F[3] }]; #IO_L17P_T2_A26_15 Sch=cd
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { F[4] }]; #IO_L13P_T2_MRCC_14 Sch=ce
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { F[5] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { F[6] }]; #IO_L4P_T0_D04_14 Sch=cg
#set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { DP }]; #IO_L19N_T3_A21_VREF_15 Sch=dp
set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { AN[0] }]; #IO_L23P_T3_FOE_B_15 Sch=an[0]
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { AN[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { AN[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { AN[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { AN[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { AN[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { AN[6] }]; #IO_L23P_T3_35 Sch=an[6]
set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { AN[7] }]; #IO_L23N_T3_A02_D18_14 Sch=an[7]
```

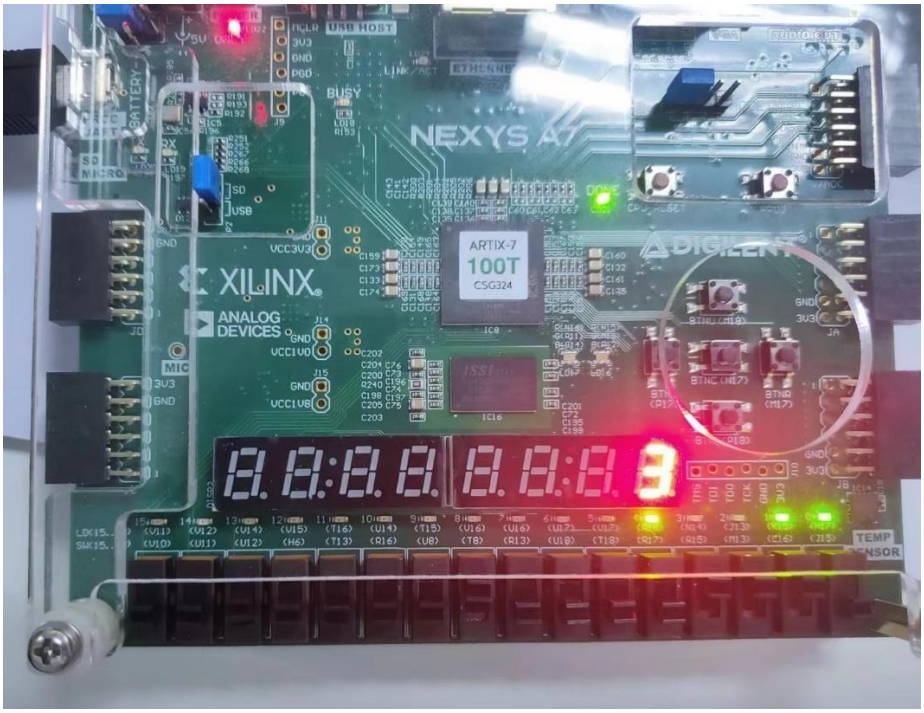
（五） 实验结果

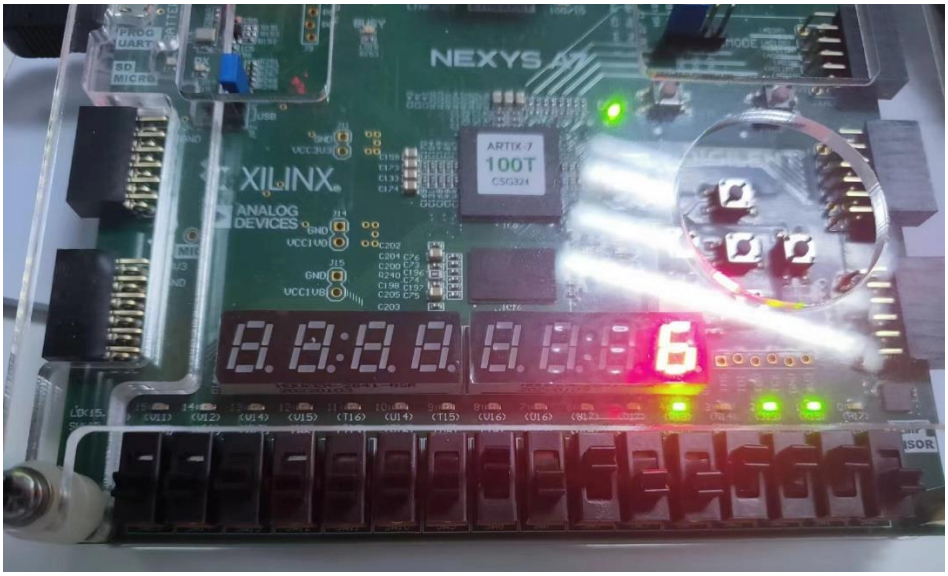
仿真结果:


Name
Value
X[7:0]
ff
en
1
valid
1
F[6:0]
ZZ
1,000.000 ns
0.000 ns
200.000 ns
400.000 ns
600.000 ns
800.000 ns
ff
ZZ

硬件实现:


NEXYS
XILINX
ANALOG DEVICES
ARTIX-7
100T
CSG324
DIGILENT
TEMP SENSOR


NEXYS A7
XILINX
ANALOG DEVICES
ARTIX-7
100T
CSG324
DIGILENT
TEMP SENSOR



（六） 实验中遇到的问题及解决方法

问题 1：不理解 reg 和 wire

解决办法 1：询问老师和助教

reg – 中间变量；→ 设计代码：输出，测试代码：输入

wire – 连线（物理）→ 设计代码：输出

问题 2：编写 testbench 时在 initial 语句块里写 wire 变量报错

不大理解测试代码如何编写

解决办法 2：上网查询资料

1. 简单测试文件 Verilog Testbench

SV/Verilog Testbench

```
module tb ();
  reg a, b, c;
  wire y;

  design dut (.a(a), .b(b), .c(c), y(y));

 // apply input sequence

  initial
    begin

      a = 0; b = 0; c = 0 ; # 10;
      a = 0; b = 0; c = 1 ; # 10;
      a = 0; b = 1; c = 0 ; # 10;
      a = 0; b = 1; c = 1 ; # 10;
      a = 1; b = 0; c = 0 ; # 10;
      a = 1; b = 0; c = 1 ; # 10;
      a = 1; b = 1; c = 0 ; # 10;
      a = 1; b = 1; c = 1 ; # 10;

    end
endmodule
```

- 被测模块的输入信号定义成 reg 类型，输出信号定义成 wire 类型。

- **initial**：通过 initial 块构造输入信号的波形，同一 initial 块内部的语句是串行执行的，多个 initial 块之间并发执行。
- **测试输入信号（所以只编写输入信号）；输出信号是通过仿真波形来观察。**

2. 自检测试文件

SV/Verilog Testbench

```
module tb2();

  reg a, b, c;
  wire y;

design dut (.a(a), .b(b), .c(c), y(y));

  initial
    begin
      a = 0; b = 0; c = 0; #10;
      if (y ! == 1) $display ("000 failed)");
      c = 1; #10;
      if (y ! == 0) $display ("001 failed)");
      b = 1; c = 0;  #10
      if (y ! == 0) $display ("010  failed);
      end
endmodule
```

- 对输出结果进行判断，并打印信息。
- **$display**
- 比 1 更好：* 结果可视可读 * 在更重要的设计（如 32 位处理器）实现时效果更显著 * 节省检查的时间和费用