实验二　译码器和编码器的设计

2021 年秋季学期

伊吉斯将另一面白色的帆交给领航，特别交代他在回航的时候，如果帖修斯平安归来，就将这面船帆升起来；要是事与愿违就用黑色的船帆，等于是悬挂出不幸的信号。

–《希腊罗马名人传》，普鲁塔克

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，请读者自行设计一个 8-3 优先编码器及七段数码管显示。

2.1 译码器

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。

常用的二进制译码器（如图 2–1所示）是一个有 n 路输入和 $m = 2^n$ 路输出的逻辑电路。译码器有一个使能信号 En，当 En=0 时，无论输入为什么，译码器没有任何有效值输出；当 En=1 时，输入的值决定了输出信号的值。在二进制码中，最常用的输出编码是 $m = 2^n$ 位中取 1 位编码，即任何时刻，m 位输出编码中只能有 1 位有效，其余各位都为 0，这样的二进制编码被称为独热编码（one-hot encoded），意思是那个被置为 1 的码看起来是 “热” 的，而二进制译码器输出的信号就是独热编码。

本次实验首先具体介绍 2-4 译码器和 3-8 译码器的工作原理，学习译码器的设计。

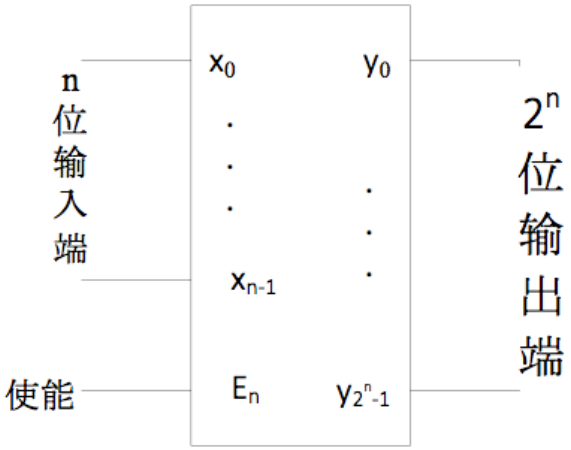
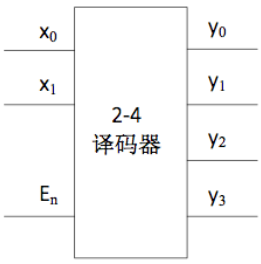


图 2-1: n 位输入，2^n 位输出的译码器

2.1.1 2-4 译码器

2-4 译码器（图 2-2所示），它的输入信号是 x_0 和 x_1，对每一个二进制输入信号 x_0 和 x_1 进行译码，译码输出是在 y_0、y_1、y_2 和 y_3 四位中选择一位使其有效。译码器的输出可以设计成高电平有效或低电平有效，在本例中采用的是高电平有效。



E_n	x_0	x_1	y_0	y_1	y_2	y_3
0	×	×	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

图 2-2: 2-4 译码器

译码器的逻辑电路图如图2-3所示。

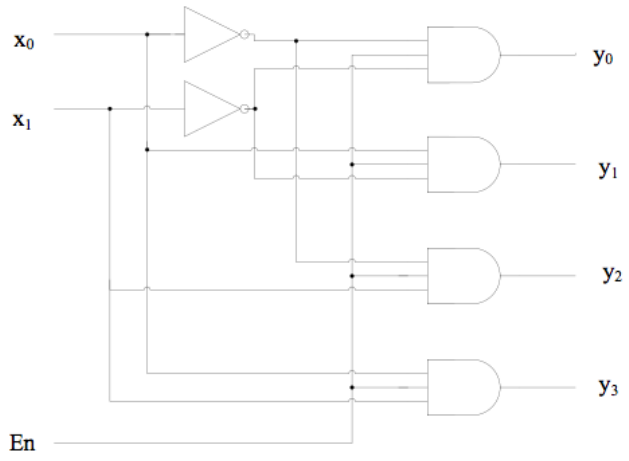


图 2-3: -4 译码器逻辑电路

可以采用硬件描述语言来实现 2-4 译码器电路。表2-1是一个带有使能端的 2-4 译码器的代码:

表 2-1: 2-4 译码器代码

```
module decode24(x,en,y);
  input  [1:0] x;
  input  en;
  output reg [3:0]y;

  always @(x or en)
    if (en)
    begin
       case (x)
             2'd0 : y = 4'b0001;
             2'd1 : y = 4'b0010;
             2'd2 : y = 4'b0100;
             2'd3 : y = 4'b1000;
       endcase
    end
    else  y = 4'b0000;

endmodule
```

程序中用到了数值字串（literal）“n’Bdd...d”，其中 n 是字串的位数，用十进制表示，这里字串的位数是 “dd...d” 这个值存放在机器中（二进制）所用的位数，而不是其用 “B” 进制表示的位数。“B” 是指定基数的单个字母，可以是 b

（二进制），o（八进制），d（十进制，可省略）和 h（十六进制）。“dd...d”是此数值字串，用 “B” 进制表示的值。

输入设计好的程序代码后，再输入激励代码，对设计进行仿真。

表 2-2: 2-4 译码器测试代码

```
`timescale 10 ns/ 1 ps
module test_decode24();
  reg  [1:0] x;
  reg  en;
  wire [3:0]y;
  decode24  i1 (
               . x (x),
               . en (en),
               .y(y) );
  initial
  begin
    en = 1'b0; x = 2'b00;  #10;
               x = 2'b01;  #10;
               x = 2'b10;  #10;
               x = 2'b11;  #10;
    en = 1'b1; x = 2'b00;  #10;
               x = 2'b01;  #10;
               x = 2'b10;  #10;
               x = 2'b11;  #10;
  end
endmodule
```

对所设计电路进行功能仿真，如图 2-4所示，分析时序图发现，结果和真值表一致。

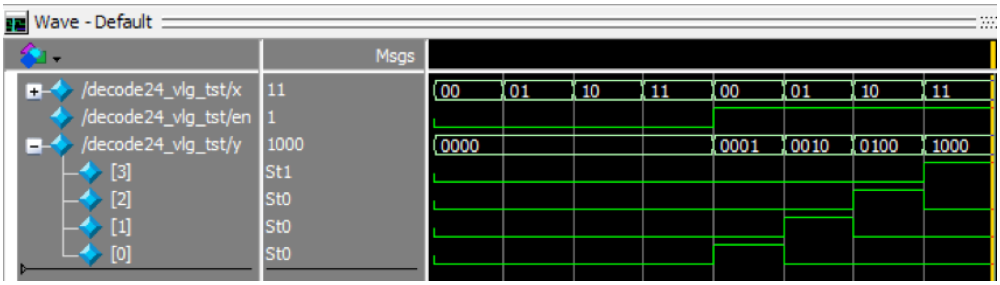


图 2-4: 2-4 译码器仿真结果

☞ for 循环语句

我们也可以利用一个 for 循环语句来实现译码器，如表 2-3实现的一个 3-8 译码器。

此代码中重复使用了 if—else 语句，请注意 if 和 else 之间的配对情况。

表 2-3: 3-8 译码器 for 循环代码

```
module decode38(x,en,y);
  input  [2:0] x;
  input  en;
  output reg [7:0]y;
  integer i;

  always @(x or en)
    if (en) begin
       for( i = 0; i <= 7; i = i+1)
           if(x == i)
                y[i] = 1;
           else
                y[i] = 0;
    end
    else
       y = 8'b00000000;

endmodule
```

另外代码中还使用了 for 循环语句。for 循环语句也是 Verilog HDL 常用的语句之一，一个 for 循环语句按照指定的次数重复执行过程赋值语句若干次。其格式为:

```
    for ( initial_assignment; condition; step_assignment )
```

初始赋值 initial_assignment 给出循环变量的初始值。condition 条件表达式指定循环在什么情况下必须结束；只要条件为真，循环中的语句就执行。而 step_assignment 给出要修改的赋值，通常为增加或减少循环变量计数。

在 Verilog HDL 代码中，for 循环的 condition 条件表达式必须为常数，不能是可改变的量。

2.2 编码器

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，如图 2-5所示。二进制编码器每次输入的 2^n 位信号中只能有一位为 1，其余均为 0（即独热码），编码器的输出端为一个二进制数，用来指示对应的哪一个位输入为 1。

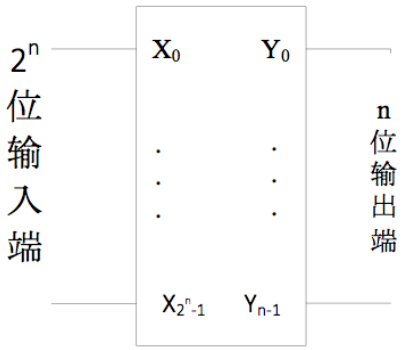


图 2-5: 2^n 位输入，n 位输出的编码器

2.2.1 4-2 编码器

4-2 编码器即为 4 位输入 2 位输出的编码器（如表 2-4所示），它的输入信号是 x_0、x_1、x_2 和 x_3，输出是 y_0 和 y_1。本例中采用独热码，每次输入中只有一位为 1，对于有两位或者两位以上为 1 的情况，则将输出置为高阻态。

表 2-4: 4-2 编码器真值表

x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

表 2-5为该 4-2 编码器的代码。

表 2-5: 4-2 编码器代码

```
module encode42(x,en,y);
  input  [3:0] x;
  input  en;
  output reg [1:0]y;

  always @(x or en)
    if (en) begin
       case (x)
          4'b0001 : y = 2'b00;
          4'b0010 : y = 2'b01;
          4'b0100 : y = 2'b10;
          4'b1000 : y = 2'b11;
          default: y = 2'bzz;
       endcase
    end
    else  y = 2'bzz;

endmodule
```

表2-6为该 4-2 编码器的测试代码。

表 2-6: 4-2 编码器测试代码

```
module test_encode4_2();
   reg en;
   reg [3:0] s;
   wire [1:0] y;

   encode42 t1(
    .en(en),
    .x(s),
    .y(y));
   initial
   begin
     en=1'b0;  s =4'b0000; #10;
               s =4'b0001; #10;
               s =4'b0010; #10;
               s =4'b0100; #10;
               s =4'b1000; #10;
     en=1'b1;  s =4'b0000; #10;
               s =4'b0001; #10;
               s =4'b0010; #10;
               s =4'b0100; #10;
               s =4'b1000; #10;
   end
endmodule
```

其仿真结果如图2-6所示:

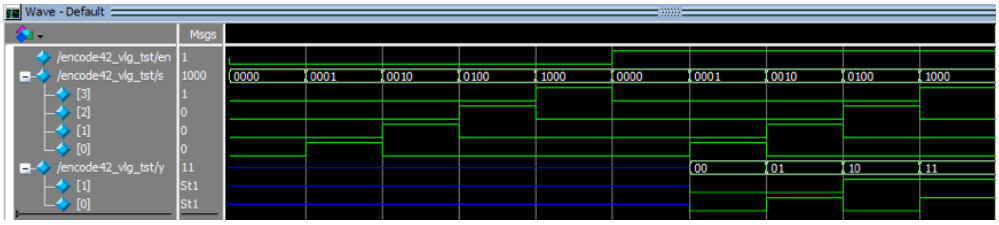


图 2-6: 4-2 编码器仿真结果

2.2.2 优先编码器

优先编码器允许同时在几个输入端有输入信号，即输入不止一个“1”，编码器按输入信号排定的优先顺序，只对同时输入的几个信号中优先权最高的一个进行编码。

我们可以利用 for 循环语句可以很方便地实现优先编码器，如表 2-7实现的一个 4-2 优先编码器。

表 2-7: 4-2 优先编码器代码

```
module encode42(x,en,y);
  input  [3:0] x;
  input  en;
  output reg [1:0]y;
  integer i;
  always @(x or en) begin
    if (en) begin
       y = 0;
       for( i = 0; i <= 3; i = i+1)
           if(x[i] == 1)  y = i;
    end
    else  y = 0;
  end
endmodule
```

请大家为上述代码编写一段测试代码，并对其进行仿真，查看仿真结果。

2.2.3 七段数码管

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，用于显示数值。图 2-7是数码管的原理图，数码管分为共阴极和共阳极两种类型，共阴极就是将七个 LED 的阴极连在一起，让其接低电平。这样给任何一个 LED 的另一端高电平，它便能点亮。而共阳极就是将七个 LED 的阳极连在一起，让其接高电平。这样，给任何一个 LED 的另一端低电平，它就能点亮。

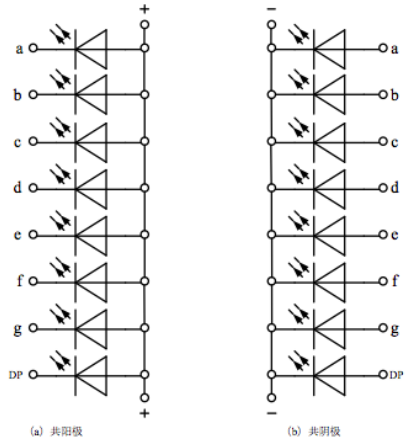


图 2-7: 数码管原理图

DE10-Standard 开发板上的数码管就是七段共阳极的。每个数码管的七段 LED 的一端连接在一个共同的阳极上，另一端和开发板上 Cyclone V SoC FPGA 的某一个一个引脚连接在一起，如图 2-8所示，如果从这个引脚输出一个逻辑 0，则此段数码管被点亮。

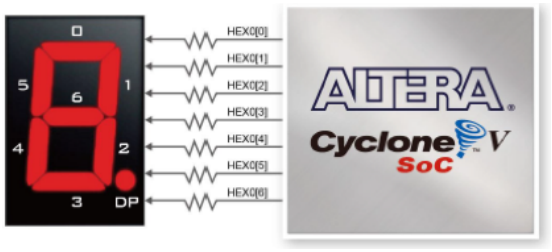


图 2-8: DE10-Standard 数码管连接

DE10-Standard 开发板共有六个数码管，各数码管和 FPGA 的引脚连接如图 2-9所示

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_W17	Seven Segment Digit 0[0]	3.3V
HEX0[1]	PIN_V18	Seven Segment Digit 0[1]	3.3V
HEX0[2]	PIN_AG17	Seven Segment Digit 0[2]	3.3V
HEX0[3]	PIN_AG16	Seven Segment Digit 0[3]	3.3V
HEX0[4]	PIN_AH17	Seven Segment Digit 0[4]	3.3V
HEX0[5]	PIN_AG18	Seven Segment Digit 0[5]	3.3V
HEX0[6]	PIN_AH18	Seven Segment Digit 0[6]	3.3V
HEX1[0]	PIN_AF16	Seven Segment Digit 1[0]	3.3V
HEX1[1]	PIN_V16	Seven Segment Digit 1[1]	3.3V
HEX1[2]	PIN_AE16	Seven Segment Digit 1[2]	3.3V
HEX1[3]	PIN_AD17	Seven Segment Digit 1[3]	3.3V
HEX1[4]	PIN_AE18	Seven Segment Digit 1[4]	3.3V
HEX1[5]	PIN_AE17	Seven Segment Digit 1[5]	3.3V
HEX1[6]	PIN_V17	Seven Segment Digit 1[6]	3.3V
HEX2[0]	PIN_AA21	Seven Segment Digit 2[0]	3.3V
HEX2[1]	PIN_AB17	Seven Segment Digit 2[1]	3.3V
HEX2[2]	PIN_AA18	Seven Segment Digit 2[2]	3.3V
HEX2[3]	PIN_Y17	Seven Segment Digit 2[3]	3.3V
HEX2[4]	PIN_Y18	Seven Segment Digit 2[4]	3.3V
HEX2[5]	PIN_AF18	Seven Segment Digit 2[5]	3.3V
HEX2[6]	PIN_W16	Seven Segment Digit 2[6]	3.3V
HEX3[0]	PIN_Y19	Seven Segment Digit 3[0]	3.3V
HEX3[1]	PIN_W19	Seven Segment Digit 3[1]	3.3V
HEX3[2]	PIN_AD19	Seven Segment Digit 3[2]	3.3V
HEX3[3]	PIN_AA20	Seven Segment Digit 3[3]	3.3V
HEX3[4]	PIN_AC20	Seven Segment Digit 3[4]	3.3V
HEX3[5]	PIN_AA19	Seven Segment Digit 3[5]	3.3V
HEX3[6]	PIN_AD20	Seven Segment Digit 3[6]	3.3V
HEX4[0]	PIN_AD21	Seven Segment Digit 4[0]	3.3V
HEX4[1]	PIN_AG22	Seven Segment Digit 4[1]	3.3V
HEX4[2]	PIN_AE22	Seven Segment Digit 4[2]	3.3V
HEX4[3]	PIN_AE23	Seven Segment Digit 4[3]	3.3V
HEX4[4]	PIN_AG23	Seven Segment Digit 4[4]	3.3V
HEX4[5]	PIN_AF23	Seven Segment Digit 4[5]	3.3V
HEX4[6]	PIN_AH22	Seven Segment Digit 4[6]	3.3V
HEX5[0]	PIN_AF21	Seven Segment Digit 5[0]	3.3V
HEX5[1]	PIN_AG21	Seven Segment Digit 5[1]	3.3V
HEX5[2]	PIN_AF20	Seven Segment Digit 5[2]	3.3V
HEX5[3]	PIN_AG20	Seven Segment Digit 5[3]	3.3V
HEX5[4]	PIN_AE19	Seven Segment Digit 5[4]	3.3V
HEX5[5]	PIN_AF19	Seven Segment Digit 5[5]	3.3V
HEX5[6]	PIN_AB21	Seven Segment Digit 5[6]	3.3V

图 2–9: DE10-Standard 数码管引脚

数字 1~9 及十六进制的 a~f 在数码管上的显示方式如图 2-10所示。

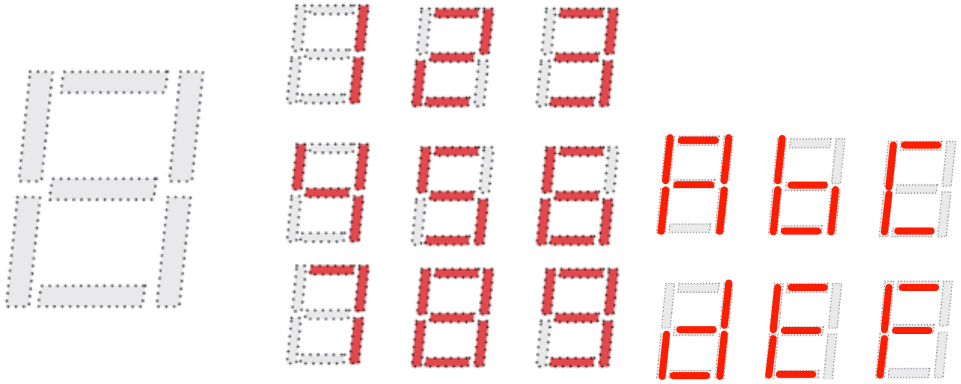


图 2-10: 七段数码管显示方式

以 HEX0 为例，如果要在此数码管上显示数字 “1”，则只要在和 HEX0[1] 以及 HEX0[2] 两段 LED 相连的 FPGA 引脚 PIN_V18 和 PIN_AG17 端输出逻辑 “0”，其他输出 “1” 即可。

请自行设计数码管的对应编码。

2.3 实验验收内容

2.3.1 上板实验

实现一个 8-3 优先编码器并在七段数码管上显示

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 输入一个 8 位二进制数，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。

例：我们从 SW7—SW0 输入 00001110，因为我们设计的是一个高位优先的优先编码器，从左（高位）开始，第一位为 1 的是第 3 号位，那么优先编

码器的编码二进制结果就为 011，将这个值显示在发光二极管上，并且指示位也同时置为 1。再对这个数值跟据七段数码管的显示进行译码，此时应显示为 “3”，用 HEX0 显示，所以 HEX0[6:0] 应该译码为 0110000（注意高低位顺序），那么在七段数码管上就会显示 “3” 这个字符。

思考：查阅相关资料，了解 casex 和 casez 语句的使用，思考如何用用 casex 语句来完成优先编码器的设计?

2.3.2 在线测试

必做 在线验证优先编码器和数码管显示代码

可选 格雷码