

浙江大学

本科实验报告

课程名称：计算机逻辑设计基础

姓 名：刘晨

学 院：计算机科学与技术学院

系：图灵 1901

专 业：计算机科学与技术

学 号：3190104666

指导教师：董亚波

2020 年 10 月 22 日

浙江大学实验报告

课程名称： 计算机逻辑设计基础 实验类型： 综合

实验项目名称： 变量译码器设计与应用

学生姓名： 刘晨 专业： 图灵班 学号： 3190104666

同组学生姓名： 指导老师： 董亚波

实验地点： 东 4-509 实验日期： 2020 年 10 月 10 日

一、实验目的和要求

掌握变量译码器的逻辑构成和逻辑功能

用变量译码器实现组合函数

采用原理图设计电路模块

进一步熟悉 ISE 平台及下载实验平台物理验证

二、实验内容和原理

实验内容：

任务 1：原理图设计实现 74LS138 译码器模块

任务 2：用 74LS138 译码器实现楼道灯控制器

实验原理：

译码器是将一种输入编码转换成另一种编码的电路，即将给定的 代码进行“翻译”并转换成指定的状态或输出信号（脉冲或电平）

译码可分为：变量译码、显示译码

变量译码一般是将一种较少位输入变为较多位输出的器件，如 $2n$ 译码和 8421BCD 码译码

显示译码主要进行 2 进制数显示成 10 进制或 16 进制数的转换， 可分为驱动 LED 和 LCD 两类

三、实验过程和数据记录

```
integer i;
  initial begin
    C = 0;
    B = 0;
    A = 0;

    G = 1;
    G2A = 0;
    G2B = 0;
    #50;

    for (i=0; i<=7;i=i+1) begin
      {C,B,A} = i;
      #50;
```

```

end
assign G = 0;
assign G2A = 0;
assign G2B = 0;
#50;

assign G = 1;
assign G2A = 1;
assign G2B = 0;
#50;

assign G = 1;
assign G2A = 0;
assign G2B = 1;
#50;

end

```

生成的波形图如下：

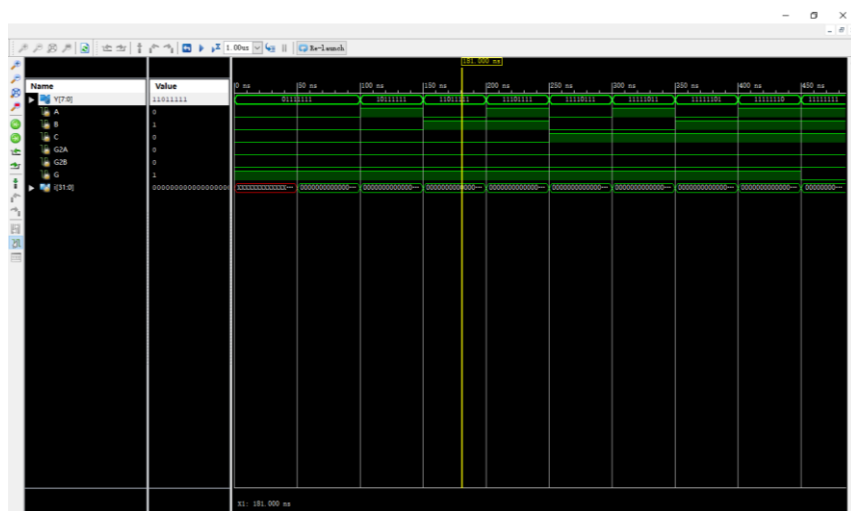


图 2 波形图

3.生成逻辑符号图

Create Schematic Symbol，系统生成 D_74LS138 模块的逻辑符号图文件，文件后缀.sym

符号图位于工程根目录

自动生成的符号可修改：可以用 Tools 菜单的 Symbol Wizard，也可以打开.sym 文件直接修改

在新工程中使用时，把.sym 和.sch 复制到对应工程目录

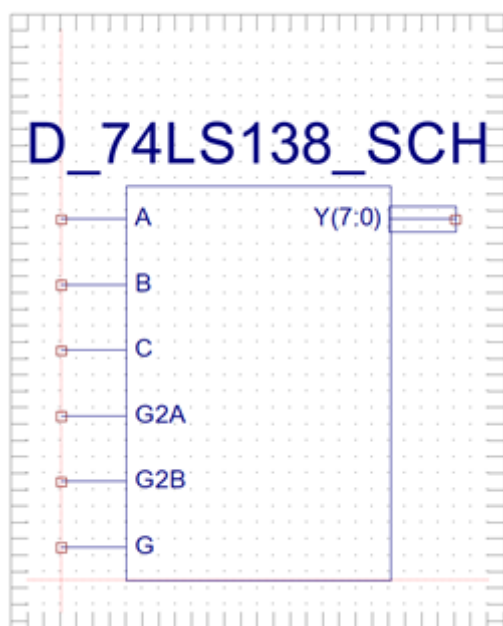


图 3 逻辑符号图

4.验证 D_74LS138

新建工程 “D_74LS138_Test”。

新建 Schematic 文件 “D_74LS138_Test”。

复制 D_74LS138.sym 和.sch 到工程目录。

在 symbols 框里的第一个元件，就是 D_74LS138。

用拨盘开关控制模块的输入，用 LED(7:0)作为模块的输出，验证模块的功能

输入			译码器输出 (低电平有效)							
使能	变量									
\overline{G} \overline{G}_{2A} \overline{G}_{2B}	CBA		Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
$\times 1 1$	$\times \times \times$		1	1	1	1	1	1	1	1
$0 \times \times$	$\times \times \times$		1	1	1	1	1	1	1	1
100	000		0	1	1	1	1	1	1	1
100	001		1	0	1	1	1	1	1	1
100	010		1	1	0	1	1	1	1	1
100	011		1	1	1	0	1	1	1	1
100	100		1	1	1	1	0	1	1	1
100	101		1	1	1	1	1	0	1	1
100	110		1	1	1	1	1	1	0	1
100	111		1	1	1	1	1	1	1	0

图 4 功能图

5.下载验证

建立 K7.ucf 文件，代码如下：

```
NET "S1" LOC = AA10 | IOSTANDARD = LVCMOS15;  
NET "S2" LOC = AB10 | IOSTANDARD = LVCMOS15;  
NET "S3" LOC = AA13 | IOSTANDARD = LVCMOS15;  
NET "S4" LOC = AA12 | IOSTANDARD = LVCMOS15;  
NET "S5" LOC = Y13 | IOSTANDARD = LVCMOS15;  
NET "S6" LOC = Y12 | IOSTANDARD = LVCMOS15;  
NET "LED[0]" LOC = W23 | IOSTANDARD = LVCMOS33;  
NET "LED[1]" LOC = AB26 | IOSTANDARD = LVCMOS33;  
NET "LED[2]" LOC = Y25 | IOSTANDARD = LVCMOS33;  
NET "LED[3]" LOC = AA23 | IOSTANDARD = LVCMOS33;  
NET "LED[4]" LOC = Y23 | IOSTANDARD = LVCMOS33;  
NET "LED[5]" LOC = Y22 | IOSTANDARD = LVCMOS33;  
NET "LED[6]" LOC = AE21 | IOSTANDARD = LVCMOS33;  
NET "LED[7]" LOC = AF24 | IOSTANDARD = LVCMOS33;
```

根据真值表，操作实验板，验证功能。

最终译码器输出如功能图（图 4）所示。

任务 2：实现楼道灯控制

1.建立文件

新建工程 LampCtrl138。

复制 D_74LS138.sym 和.sch 文件到工程目录。

在 symbols 框里的第一个元件，就是 D_74LS138。

根据前面原理，用原理图方式输入。

1 用 VCC，0 用 GND。

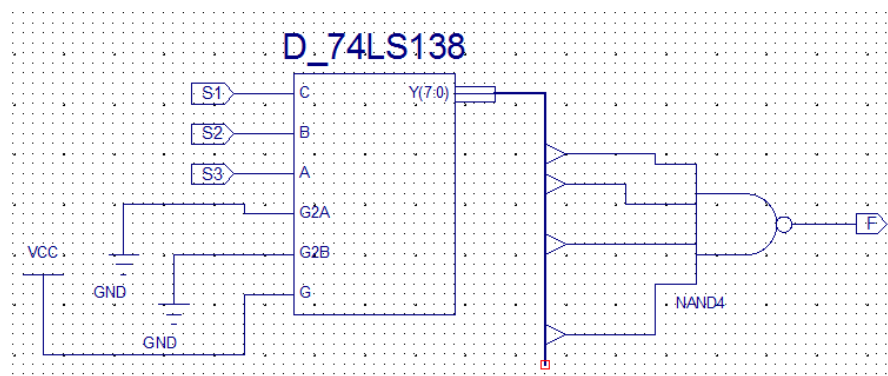


图 5 原理图-2

2.仿真

```
module LampCtrl_sim;
    // Inputs
    reg clk;
    reg S1;
    reg S2;
    reg S3;
    // Outputs
    wire F;

    // Instantiate the Unit Under Test (UUT)
    LampCtrl uut (
        .clk(clk),
        .S1(S1),
        .S2(S2),
        .S3(S3),
        .F(F)
    );
    initial begin
        // Initialize Inputs
        clk = 0;
        S1 = 0;S2 = 0;S3 = 0;

        #600 S1 = 1;
        #20 S1 = 0;
        #6000 S2 = 1;
        #20 S2 = 0;
        #6000 S3 = 1;
        #20 S3 = 0;
    end

    always begin
        #10 clk = 0;
        #10 clk = 1;
    end
endmodule
```

以上为仿真代码。

3.下载验证

```
NET "S1" LOC = AA10 | IOSTANDARD = LVCMOS15;
NET "S2" LOC = AB10 | IOSTANDARD = LVCMOS15;
NET "S3" LOC = AA13 | IOSTANDARD = LVCMOS15;
```

```
NET "F" LOC = AF24 | IOSTANDARD = LVCMOS33;
```

以上为 k7.ucf 文件的内容。

四、实验结果分析

D_74LS138 的输出：

输入		译码器输出 (低电平有效)								
使能	变量									
GG _{2A} G _{2B}	CBA	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	
×11	×××	1	1	1	1	1	1	1	1	
0××	×××	1	1	1	1	1	1	1	1	
100	000	0	1	1	1	1	1	1	1	
100	001	1	0	1	1	1	1	1	1	
100	010	1	1	0	1	1	1	1	1	
100	011	1	1	1	0	1	1	1	1	
100	100	1	1	1	1	0	1	1	1	
100	101	1	1	1	1	1	0	1	1	
100	110	1	1	1	1	1	1	0	1	
100	111	1	1	1	1	1	1	1	0	

当 G，G2A，G2B 分别为 0 1 1 时，随着 S1，S2，S3 的变化，译码器的输出，即灯的亮和暗如上表所示。

开关的输入			楼道灯的输出
S ₁	S ₂	S ₃	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1