

数逻实验报告 Lab5

雷远航

October 28, 2022

Abstract

实验项目: 变量译码器

1 操作方法与实验步骤

1.1 原理图设计实现 74LS138 译码器模块

1.1.1 画出原理图

用原理图的方式在 ISE 中绘制译码器

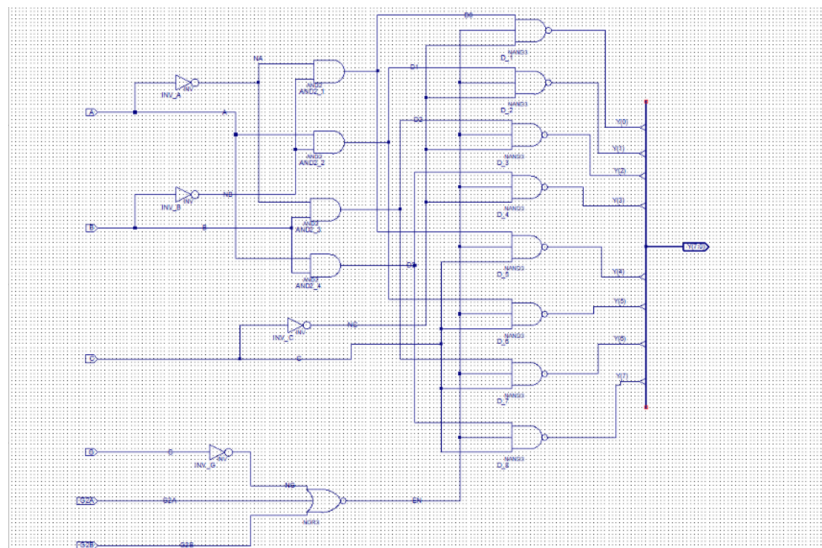


Figure 1: 原理图

1.1.2 在 ISE 中运行生成 HDL 代码

Check Design Rules 检查错误, 运行 Synthesize, 使用 View HDL Function Model 查看 HDL 代码

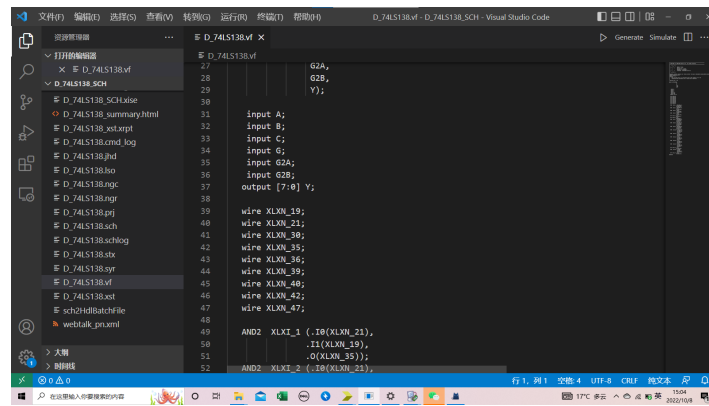


Figure 2: HDL

1.1.3 对模块进行仿真

导入测试文件对模块进行模拟仿真

```

1      `timescale 1ns / 1ps
2
3      module MyMC14495_tb();
4
5          // Inputs
6          reg D0;
7          reg D1;
8          reg D2;
9          reg D3;
10         reg LE;
11         reg point;
12
13         // Output
14         wire p;
15         wire a;
16         wire b;
17         wire c;
18         wire d;
19         wire e;
20         wire f;
21         wire g;
22
23         // Instantiate the UUT
24         MyMC14495_HDL UUT (
25             .D0(D0),
26             .D1(D1),
27             .D2(D2),

```

```

28         .D3(D3),
29         .LE(LE),
30         .point(point),
31         .p(p),
32         .a(a),
33         .b(b),
34         .c(c),
35         .d(d),
36         .e(e),
37         .f(f),
38         .g(g)
39     );
40     // Initialize Inputs
41     integer i;
42     initial begin
43         //$dumpfile("MyMC14495_HDL.vcd");
44         //$dumpvars(1, MyMC14495_HDL_tb);
45
46         D3 = 0;
47         D2 = 0;
48         D1 = 0;
49         D0 = 0;
50         LE = 1'b0;
51         point = 0;
52
53         for (i=0; i<=15; i=i+1) begin
54             {D3,D2,D1,D0}=i;
55             point = i;
56             #50;
57         end
58
59         #50;
60         LE = 1'b1;
61         #10;
62         end
63     endmodule

```

测试代码解释：

1. 测试时的输入部分为六个寄存器 (reg):G,G2A,G2B,C,A,B. 他们与所画原理图的输入部分相对应
2. 输出部分为 8 位的网线 (wire)Y
3. 实例化自己所画的模块 D 74LS138, 命名为 UUT

4. 在 initial 部分中对输入部分的值进行相应的改变和赋值, 从而达到测试的目的

在本 initial 块中, 设置 G,G2A 和 G2B 确保总控为开启的状态, 然后通过 for 语句遍历可以改变 A,B,C 的输入从而可以在波形图上显示

1.1.4 完成全部操作

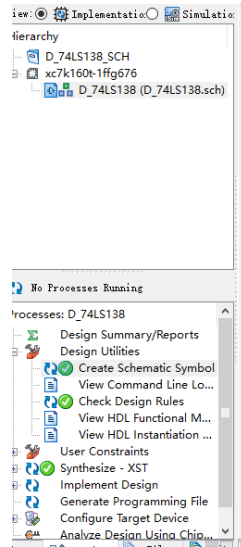


Figure 3: 验证

1.2 验证 D 74LS138

1.2.1 D 74LS138 TEST 原理图

导入第一个工程的 *.sym 与 *.vf 到第二个工程当中, 进行原理图的绘制

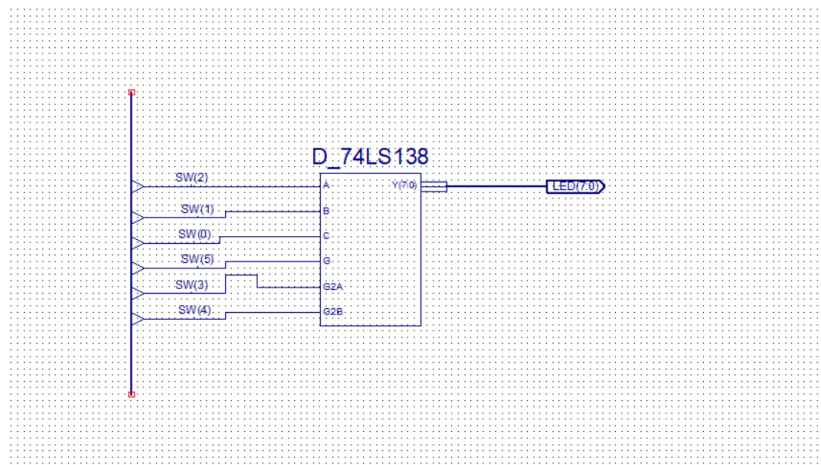


Figure 4: 原理图

1.2.2 下版验证

导入 SWORD4.ucf 文件, 将生成的 .bit 文件导入到实验版中, 进行实验结果的验证

1.3 实现楼道灯控制

1.3.1 进行原理图绘制

导入第一个工程的 *.sym 与 *.vf 到第三个工程当中, 进行原理图的绘制

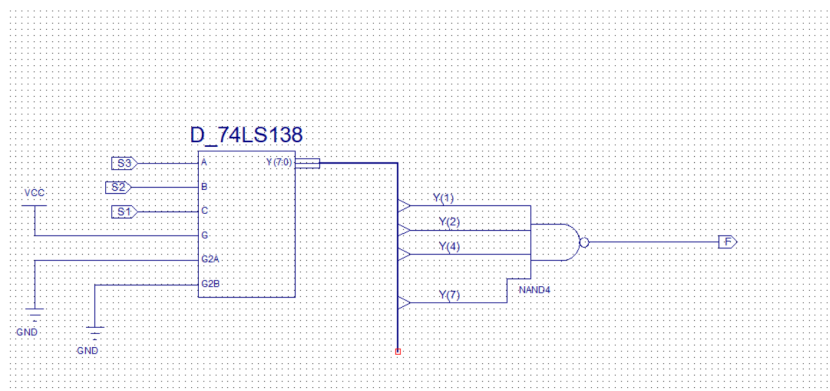


Figure 5: 原理图

1.3.2 对模块进行模拟仿真

导入仿真激励代码, 进行模拟仿真

测试文件各部分的意义已在 1.1.3 节中进行说明.

1.3.3 下版验证

导入 SWORD4.ucf 文件, 将生成的 .bit 文件导入到实验版中, 进行实验结果的验证

1.4 Bonous:LampCtl.v 的验证

1.4.1 自行书写 LampCtrl.v

```
1      module Decoder(  
2          input wire S1,  
3          input wire S2,  
4          input wire S3,  
5          output wire K1,  
6          output wire K2  
7      );  
8  
9      reg [2:0] test;  
10     reg temp1;  
11     reg temp2;  
12  
13     always @(*) begin  
14
```

```

15
16     test = {S3,S2,S1};
17     temp1 = 1'b0;
18     temp2 = 1'b0;
19
20     case (test)
21
22     3'b001:temp1 = 1'b1;
23     3'b111:temp2 = 1'b1;
24
25
26
27     endcase
28
29     end
30     assign K1 = temp1;
31     assign K2 = temp2;
32
33     endmodule
34
35     module OR(
36         input wire S1,
37         input wire S2,
38         output wire F
39     );
40
41     assign F = S1|S2;
42
43     endmodule
44
45     module LampCtrl(
46         input wire S1,
47         input wire S2,
48         input wire S3,
49         output wire F
50     );
51     wire K1;
52     wire K2;
53     Decoder d1(S1,S2,S3,K1,K2);
54     OR d2(K1,K2,F);
55

```

56
57
58

```
endmodule
```

代码解释

在 Decoder 模块中传入三个开关的信号 S1,S2,S3, 以及灯是否闪亮的两种情况的结果 K1,K2. 当结果输入的情况为 001 和 111 的两种情况时, 将对应的输出置为 1.

在 OR 模块中返回两个输入信号取或运算的结果

在 LampCtrl 模块中, 首先调用 Decoder 模块判断对应的输入情况, 然后调用 OR 模块获得输出的结果

1.4.2 对代码进行模拟仿真

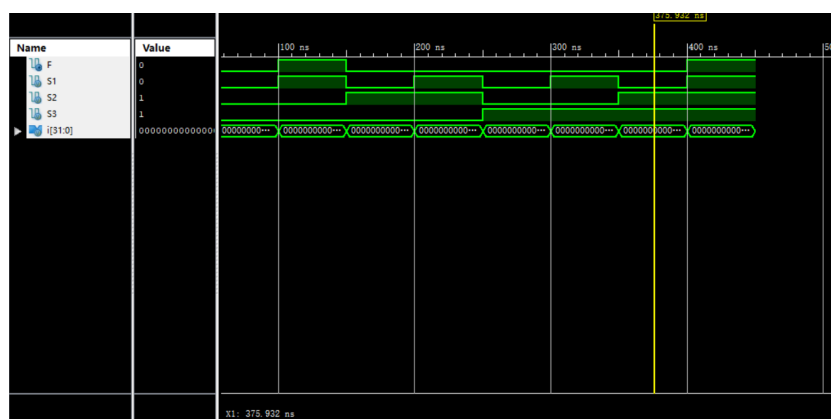


Figure 6: 模拟仿真

波形图分析

在仿真激励代码中, 输入 S3,S2,S1, 一次从 0 到 7 进行赋值, 可以发现, 当输入为 001 和 111 的两种情况时, 最终的结果才会输出为 1, 与所书写的代码的情况符合

下版验证

下版验证时可以看到, 当拨动开关的情况为上述所说的两种情况时灯才会闪亮, 其他情况并不闪亮. Bonous 部分下版验证已通过验收, 因此不再插入图片.

2 实验结果描述与分析

2.1 Part1: 译码器原理图验证

所得到的波形图结果如下:

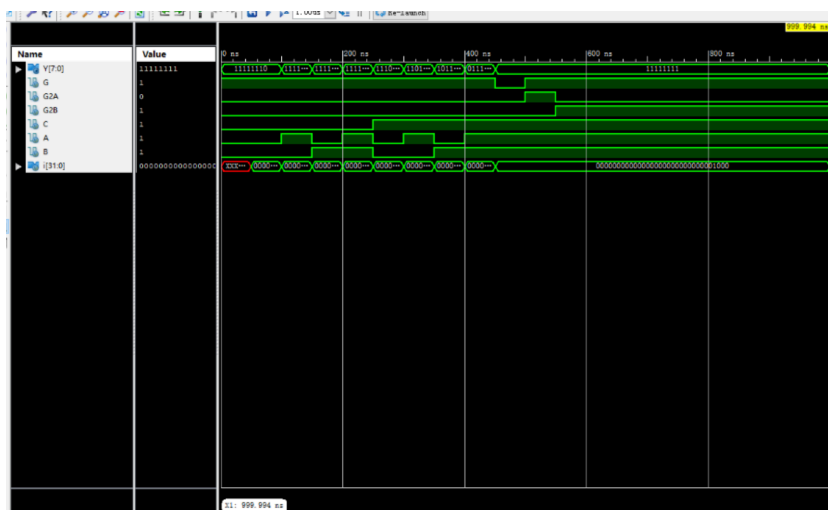


Figure 7: 原理图

波形分析:

根据测试文件, 开始的部分总控为开启状态, 所输入的 C,B,A 依次从 0 到 7 进行遍历, 因此对应的输出为遍历到的倒数第 i 个位置为 0, 其余位为 1

而后总控开关的状态为关闭, 因此输出的所有位都为 1

2.2 Part2: 译码器下版验证

下版验证的结果如下:

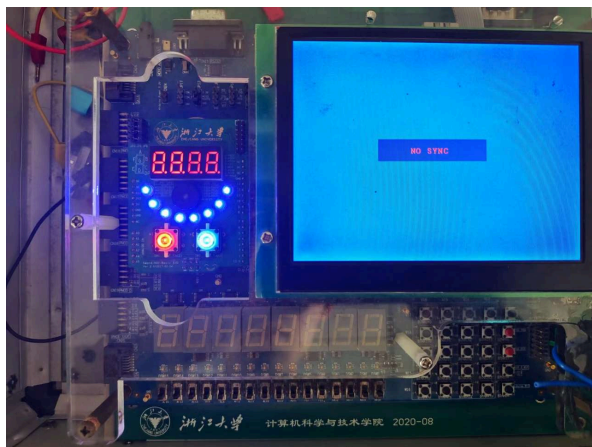


Figure 8: 下版验证

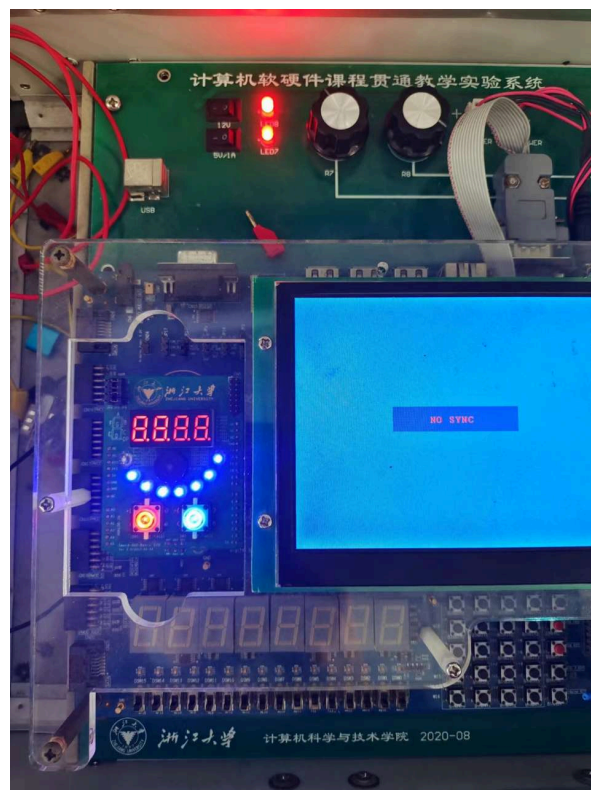


Figure 9: 下版验证

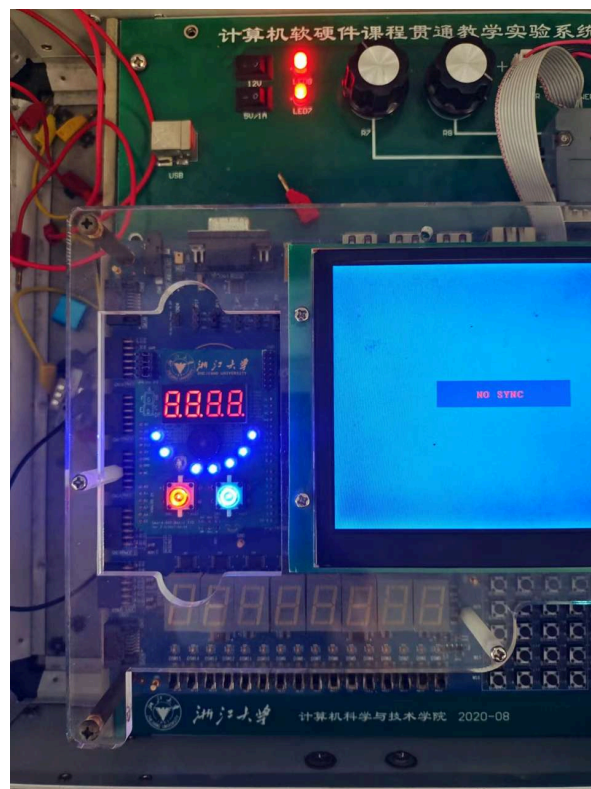


Figure 10: 下版验证

下版结果解释:

若控制总控 (enable) 的三个开关调整为关闭的状态时, 如第一张图所示, 无论后面的开关如何进行改变所有的灯都处于闪亮的状态, 如果总控的状态调整为开启的状态, 那么通过 A,B,C 输入所绑定的实验版开关输入对应的数字 (二进制), 对应的第 i 盏灯就会熄灭

2.3 Part3: 灯光控制验证

2.3.1 波形图

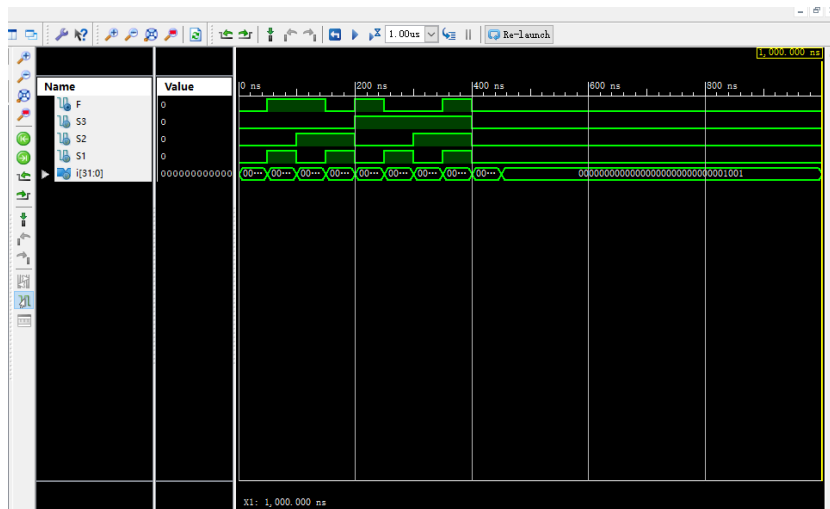


Figure 11: 波形图

波形解释: 由测试文件可以, 对应的输入从 0 到 7 遍历, 在波形图所示的八个对应段中如果有奇数个输入为 1 时, 那么灯光闪亮, 对应的输出 F 为 1, 通过波形图可以看出当输入为 $\{1,1,1\}, \{1,0,0\}, \{0,1,0\}, \{0,0,1\}$ 时, 对应的输出 F 为 1

2.3.2 下版验证

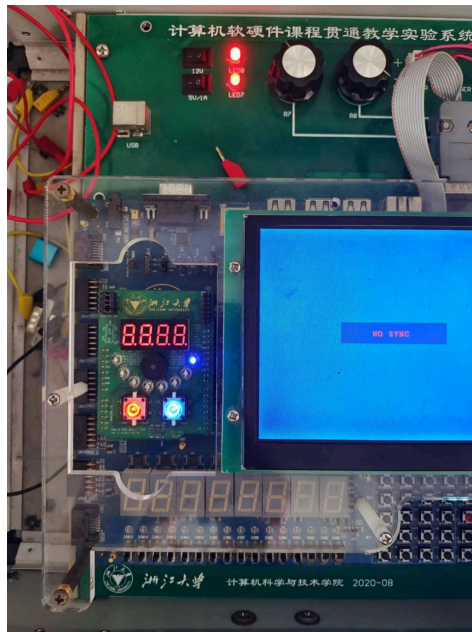


Figure 12: 下版验证

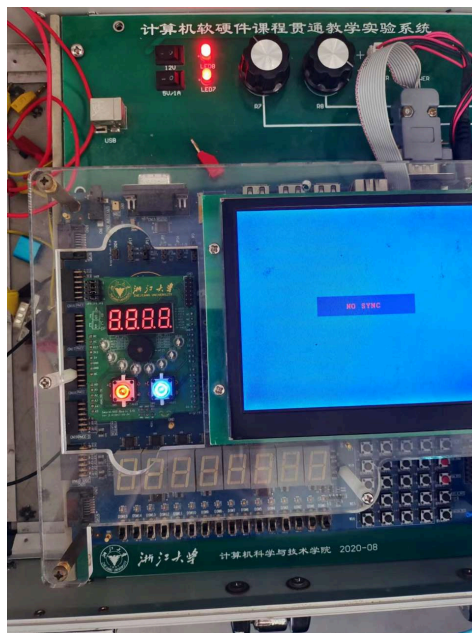


Figure 13: 下版验证

下版现象解释:

通过下版测试, 当实验板被绑定的三个开关中, 有奇数个开关被向上拨动时, 可以看到, 此时实验板所对应的灯会闪亮如果被拨动的开关数目为偶数个, 那么灯就不会闪亮, 这与所实现的灯光控制的结构图是相符合的, 当输入为 $\{1,1,1\}$, $\{1,0,0\}$, $\{0,1,0\}$, $\{0,0,1\}$ 时, 对应的输出 F 为 1, 因此, 对应的灯会闪亮, 这是与预期相符合的

3 讨论与心得

本次实验遇到的问题有:(1) 用自己的电脑连接 sword 实验版时存在连接不上的问题, 在更换了几个实验版后才找到可以连接的实验板. (2) 在实验时由于对应的引脚约束的顺序开始时有一定问题, 导致下版的结果有一定的问题, 在修改引脚约束之后才解决了问题.

4 Bonous

Bonous 部分见 1.4 节