

# 浙江大学

## 本科实验报告

课程名称: 数字逻辑设计

姓 名:

学 院: 竺可桢学院

专 业: 混合班

指导教师: 董亚波

报告日期: 2025 年 3 月 23 日

# 浙江大学实验报告

课程名称： 数字逻辑设计 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：    学号：    同组学生姓名： 无

实验地点： 紫金港东四 509 室 实验日期： 2025 年 3 月 20 日

## 一、实验目的

- 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 Vivado 平台及下载实验平台物理验证

## 二、操作方法与实验步骤

### 1. 原理图设计实现 74LS138 译码器模块

实验原理为：

- 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。
- 变量译码器是一个将  $n$  个输入变为  $2n$  个最小项输出的多输出端的组合逻辑电路。 $n$  通常在 2~64 之间。

实验步骤如下：

- 在 Digital 里画原理图，命名为 D\_74LS138。
- 在 Digital 上设计测试用例，检查错误，将仿真结果写到实验报告里。
- 导出 Verilog，查看并学习 Verilog HDL 代码。
- 在 Vivado 上进行后续的仿真以及在实验板上验证。

### 2. 用 74LS138 译码器实现楼道灯控制

需要解决的问题：某三层楼房的楼梯通道共用一盏灯，每层楼都安装了一只开关并能独立控制该灯，请设计楼道灯的控制电路。（与 Lab4 任务一相同）

实验步骤如下：

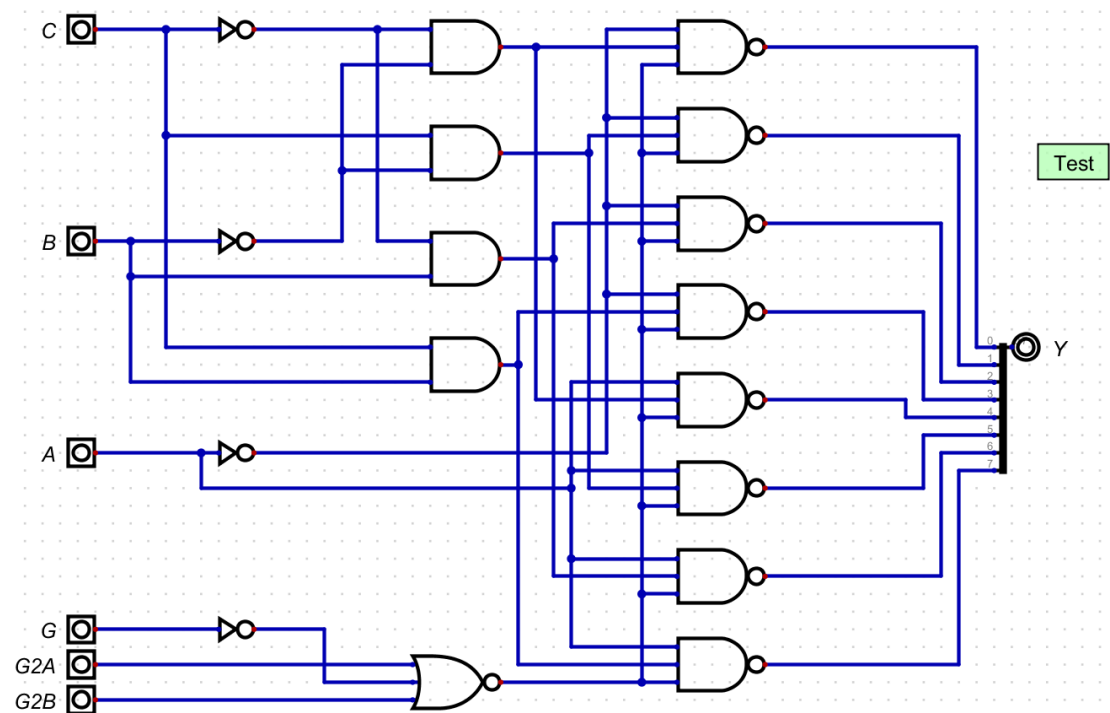
- 在 Digital 新建电路并保存为文件 LampCtrl138。
- 使用任务 1 建立的 D\_74LS138 模块和一个 4 输入与非门实现三个开关的楼道灯控制逻辑。

- 导出 Lampctr1138.v 文件。
- 在 Vivado 上进行后续的仿真以及在实验板上验证。

### 三、实验结果与分析


#### 1. 原理图设计实现 74LS138 译码器模块


按照真值表搭建的电路如图所示：

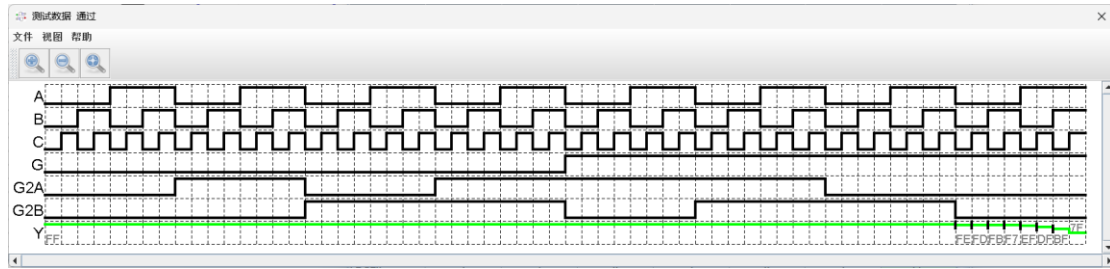


对该电路图进行正确性检验，结果如下（已加入预期结果）：

1	A	B	C	G2A	G2B	Y
2	x	x	0	x	x	0xFF
3	x	x	1	1	x	0xFF
4	x	x	1	0	1	0xFF
5	0	0	1	0	0	0xFE
6	0	0	1	0	0	0xFD
7	0	1	0	1	0	0xFB
8	0	1	1	0	0	0xF7
9	1	0	0	1	0	0xEF
10	1	0	1	0	0	0xDF
11	1	1	0	1	0	0xBF
12	1	1	1	0	0	0x7F

测试结果							
文件 视图							
							
通过							
	A	B	C	G	G2A	G2B	Y
L2;X=0	0	0	0	0	0	0	FF
L2;X=1	0	0	1	0	0	0	FF
L2;X=2	0	1	0	0	0	0	FF
L2;X=3	0	1	1	0	0	0	FF
L2;X=4	1	0	0	0	0	0	FF
L2;X=5	1	0	1	0	0	0	FF
L2;X=6	1	1	0	0	0	0	FF
L2;X=7	1	1	1	0	0	0	FF
L2;X=8	0	0	0	0	1	0	FF
L2;X=9	0	0	1	0	1	0	FF
L2;X=10	0	1	0	0	1	0	FF
L2;X=11	0	1	1	0	1	0	FF
L2;X=12	1	0	0	0	1	0	FF
L2;X=13	1	0	1	0	1	0	FF
L2;X=14	1	1	0	0	1	0	FF
L2;X=15	1	1	1	0	1	0	FF
L2;X=16	0	0	0	0	0	1	FF
L2;X=17	0	0	1	0	0	1	FF
L2;X=18	0	1	0	0	0	1	FF
L2;X=19	0	1	1	0	0	1	FF
L2;X=20	1	0	0	0	0	1	FF
L2;X=21	1	0	1	0	0	1	FF
L2;X=22	1	1	0	0	0	1	FF
L2;X=23	1	1	1	0	0	1	FF
L2;X=24	0	0	0	0	1	1	FF
L2;X=25	0	0	1	0	1	1	FF
L2;X=26	0	1	0	0	1	1	FF
L2;X=27	0	1	1	0	1	1	FF
L2;X=28	1	0	0	0	1	1	FF
L2;X=29	1	0	1	0	1	1	FF
L2;X=30	1	1	0	0	1	1	FF
L2;X=31	1	1	1	0	1	1	FF

测试结果							
文件 视图							
							
通过							
	A	B	C	G	G2A	G2B	Y
L3;X=0	0	0	0	1	1	0	FF
L3;X=1	0	0	1	1	1	0	FF
L3;X=2	0	1	0	1	1	0	FF
L3;X=3	0	1	1	1	1	0	FF
L3;X=4	1	0	0	1	1	0	FF
L3;X=5	1	0	1	1	1	0	FF
L3;X=6	1	1	0	1	1	0	FF
L3;X=7	1	1	1	1	1	0	FF
L3;X=8	0	0	0	1	1	1	FF
L3;X=9	0	0	1	1	1	1	FF
L3;X=10	0	1	0	1	1	1	FF
L3;X=11	0	1	1	1	1	1	FF
L3;X=12	1	0	0	1	1	1	FF
L3;X=13	1	0	1	1	1	1	FF
L3;X=14	1	1	0	1	1	1	FF
L3;X=15	1	1	1	1	1	1	FF
L4;X=0	0	0	0	1	0	1	FF
L4;X=1	0	0	1	1	0	1	FF
L4;X=2	0	1	0	1	0	1	FF
L4;X=3	0	1	1	1	0	1	FF
L4;X=4	1	0	0	1	0	1	FF
L4;X=5	1	0	1	1	0	1	FF
L4;X=6	1	1	0	1	0	1	FF
L4;X=7	1	1	1	1	0	1	FF
L5	0	0	0	1	0	0	FE
L6	0	0	1	1	0	0	FD
L7	0	1	0	1	0	0	FB
L8	0	1	1	1	0	0	F7
L9	1	0	0	1	0	0	EF
L10	1	0	1	1	0	0	DF
L11	1	1	0	1	0	0	BF
L12	1	1	1	1	0	0	7F



可以观察到这 64 组输出值均符合预期，即说明该电路无误，此时将该电路图导出为 Verilog 代码：

```
/*
 * Generated by Digital. Don't modify this file!
 * Any changes will be lost if this file is regenerated.
 */

module D_74LS138 (
    input C,
    input B,
    input A,
    input G,
    input G2A,
    input G2B,
    output [7:0] Y
);
    wire s0;
    wire s1;
    wire s2;
    wire s3;
    wire s4;
    wire s5;
    wire s6;
    wire s7;
    assign s0 = ~ C;
    assign s1 = ~ B;
    assign s2 = ~ A;
    assign s3 = ~ (G2A | ~ G | G2B);
    assign s7 = (C & B);
    assign s4 = (s0 & s1);
    assign s5 = (C & s1);
    assign s6 = (s0 & B);
    assign Y[0] = ~ (s2 & s4 & s3);
    assign Y[1] = ~ (s2 & s5 & s3);
    assign Y[2] = ~ (s2 & s6 & s3);
    assign Y[3] = ~ (s2 & s7 & s3);
    assign Y[4] = ~ (A & s4 & s3);
```

```

assign Y[5] = ~ (A & s5 & s3);
assign Y[6] = ~ (A & s6 & s3);
assign Y[7] = ~ (A & s7 & s3);

```

**endmodule**

将该 Verilog 代码导入 Vivado 平台，并编写相应的仿真代码和约束文件：  
仿真代码：

```

`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2025/03/15 15:55:19
// Design Name:
// Module Name: D_74LS138_testbench
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

module D_74LS138_testbench();
    reg C;
    reg B;
    reg A;
    reg G;
    reg G2A;
    reg G2B;
    wire [7:0] Y;
    D_74LS138 D_74LS138_UUT(
        .C(C),
        .B(B),
        .A(A),
        .G(G),
        .G2A(G2A),
        .G2B(G2B),

```

```

        .Y(Y)
    );
integer i;
initial begin
    A = 0;
    B = 0;
    C = 0;

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

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

```

测试代码:

```

module D_74LS138_Test(
    input S1,
    input S2,
    input S3,
    input S4,
    input S5,
    input S6,
    output [7:0] LED
);

```

```

D_74LS138 D1 (
.A(S3),
.B(S2),
.C(S1),
.G(S4),
.G2A(S5),
.G2B(S6),
.Y(LED)
);
endmodule

```

约束文件:

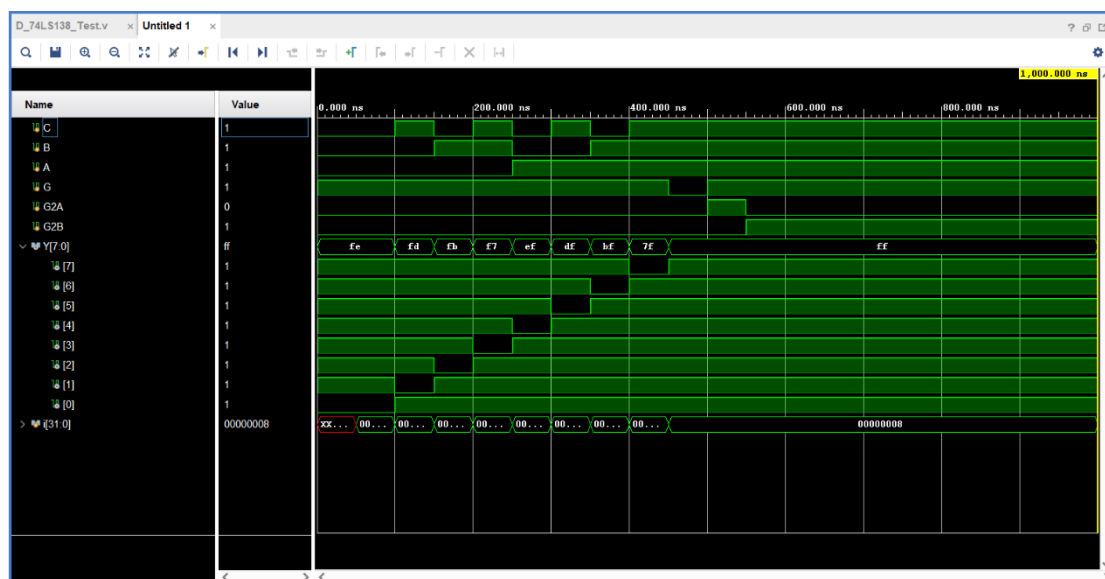
```

1  set_property PACKAGE_PIN AA10 [get_ports {S1}]
2  set_property IOSTANDARD LVCMOS15 [get_ports {S1}]
3  set_property PACKAGE_PIN AB10 [get_ports {S2}]
4  set_property IOSTANDARD LVCMOS15 [get_ports {S2}]
5  set_property PACKAGE_PIN AA13 [get_ports {S3}]
6  set_property IOSTANDARD LVCMOS15 [get_ports {S3}]
7  set_property PACKAGE_PIN AA12 [get_ports {S4}]
8  set_property IOSTANDARD LVCMOS15 [get_ports {S4}]
9  set_property PACKAGE_PIN Y13 [get_ports {S5}]
10 set_property IOSTANDARD LVCMOS15 [get_ports {S5}]
11 set_property PACKAGE_PIN Y12 [get_ports {S6}]
12 set_property IOSTANDARD LVCMOS15 [get_ports {S6}]
13 set_property PACKAGE_PIN W23 [get_ports {LED[0]}]
14 set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]
15 set_property PACKAGE_PIN AB26 [get_ports {LED[1]}]
16 set_property IOSTANDARD LVCMOS33 [get_ports {LED[1]}]
17 set_property PACKAGE_PIN Y25 [get_ports {LED[2]}]
18 set_property IOSTANDARD LVCMOS33 [get_ports {LED[2]}]
19 set_property PACKAGE_PIN AA23 [get_ports {LED[3]}]
20 set_property IOSTANDARD LVCMOS33 [get_ports {LED[3]}]
21 set_property PACKAGE_PIN Y23 [get_ports {LED[4]}]
22 set_property IOSTANDARD LVCMOS33 [get_ports {LED[4]}]
23 set_property PACKAGE_PIN Y22 [get_ports {LED[5]}]
24 set_property IOSTANDARD LVCMOS33 [get_ports {LED[5]}]
25 set_property PACKAGE_PIN AE21 [get_ports {LED[6]}]
26 set_property IOSTANDARD LVCMOS33 [get_ports {LED[6]}]
27 set_property PACKAGE_PIN AF24 [get_ports {LED[7]}]
28 set_property IOSTANDARD LVCMOS33 [get_ports {LED[7]}]

```

Vivado 平台上的仿真实验如下:



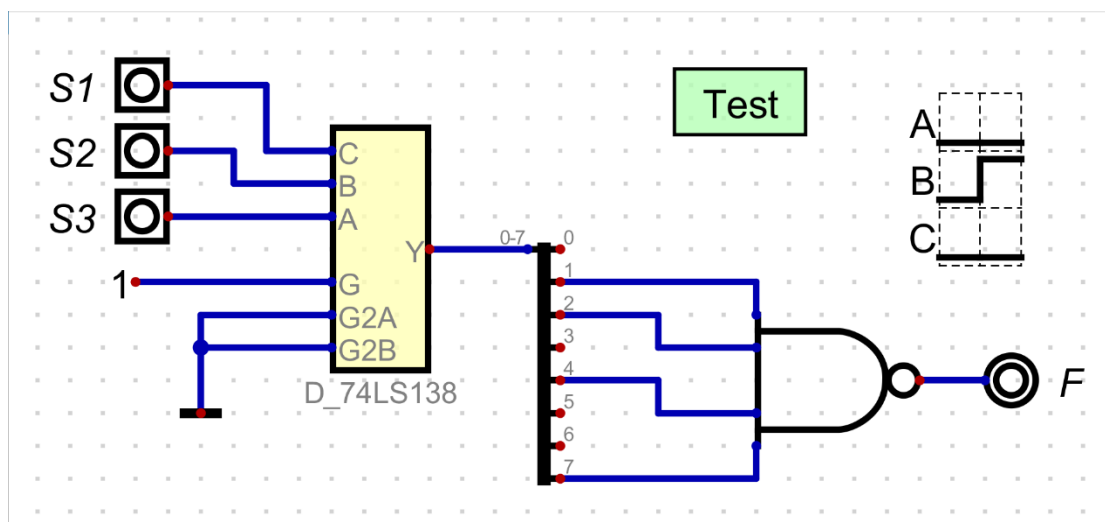


可以看到，当 G,G2A,G2B 状态正常（为 100）时，该译码器能够正常地将 000~111 的二进制码全部转化为对应的输出，证明该代码无误。

实验板上验证结果为：当 G,G2A,G2B 状态异常（不为 100）时，七盏灯全亮；状态正常时，每个二进制状态下都恰有一盏灯熄灭，与两次仿真结果一致。

## 2. 用 74LS138 译码器实现楼道灯控制

按照真值表搭建的电路如图所示：



对该电路图进行正确性检验，结果如下（已加入预期结果）：

测试数据

	S1	S2	S3	F
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

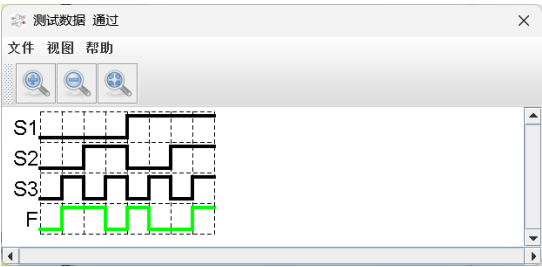
帮助 取消 运行测试用例(当前电路) 确定

测试结果

文件 视图

通过

	S1	S2	S3	F
L2	0	0	0	0
L3	0	0	1	1
L4	0	1	0	1
L5	0	1	1	0
L6	1	0	0	1
L7	1	0	1	0
L8	1	1	0	0
L9	1	1	1	1



可以观察到这 8 组输出值均符合预期，即说明该电路无误，此时将该电路图导出为 Verilog 代码：

```
/*
 * Generated by Digital. Don't modify this file!
 * Any changes will be lost if this file is regenerated.
 */

module D_74LS138 (
    input C,
    input B,
    input A,
    input G,
    input G2A,
    input G2B,
    output [7:0] Y
);
    wire s0;
    wire s1;
    wire s2;
    wire s3;
    wire s4;
    wire s5;
    wire s6;
    wire s7;
    assign s0 = ~ C;
    assign s1 = ~ B;
    assign s2 = ~ A;
    assign s3 = ~ (G2A | ~ G | G2B);
    assign s7 = (C & B);
    assign s4 = (s0 & s1);
    assign s5 = (C & s1);
    assign s6 = (s0 & B);
    assign Y[0] = ~ (s2 & s4 & s3);
    assign Y[1] = ~ (s2 & s5 & s3);
    assign Y[2] = ~ (s2 & s6 & s3);
    assign Y[3] = ~ (s2 & s7 & s3);
    assign Y[4] = ~ (A & s4 & s3);
    assign Y[5] = ~ (A & s5 & s3);
    assign Y[6] = ~ (A & s6 & s3);
    assign Y[7] = ~ (A & s7 & s3);
endmodule

module Lampctrl138 (
```

```

input S1,
input S2,
input S3,
output F
);
wire [7:0] s0;
D_74LS138 D_74LS138_i0 (
    .C( S1 ),
    .B( S2 ),
    .A( S3 ),
    .G( 1'b1 ),
    .G2A( 1'b0 ),
    .G2B( 1'b0 ),
    .Y( s0 )
);
assign F = ~ (s0[1] & s0[2] & s0[4] & s0[7]);
endmodule

```

将该 Verilog 代码导入 Vivado 平台，并编写相应的仿真代码和约束文件：  
仿真代码：

```

`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2025/03/15 16:49:27
// Design Name:
// Module Name: Lampctrl138_testbench
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
module Lampctrl138_testbench();
    reg S1;
    reg S2;

```

```

reg S3;
wire F;
Lampctrl138 Lampctrl138_UUT (
    .S1(S1),
    .S2(S2),
    .S3(S3),
    .F(F)
);
initial begin
    S1 = 0;
    S2 = 0;
    S3 = 0;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 1;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 0;
    S3 = 1;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 1;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 0;
    S3 = 0;
end

endmodule

```

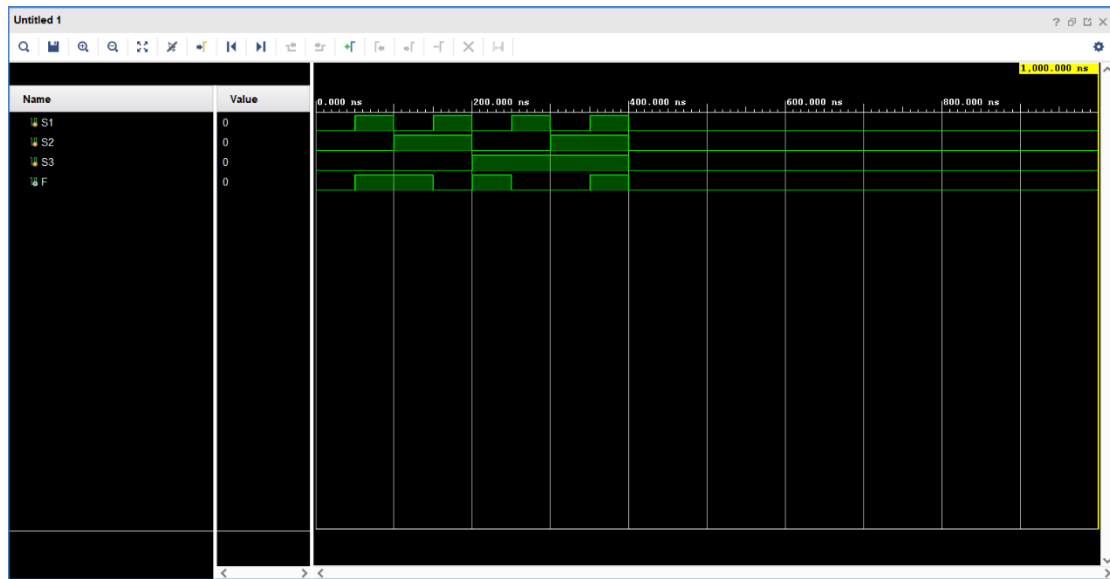
约束文件：

```

1 | set_property PACKAGE_PIN AA10 [get_ports {S1}]
2 | set_property IOSTANDARD LVCMOS15 [get_ports {S1}]
3 | set_property PACKAGE_PIN AB10 [get_ports {S2}]
4 | set_property IOSTANDARD LVCMOS15 [get_ports {S2}]
5 | set_property PACKAGE_PIN AA13 [get_ports {S3}]
6 | set_property IOSTANDARD LVCMOS15 [get_ports {S3}]
7 | set_property PACKAGE_PIN AF24 [get_ports {F}]
8 | set_property IOSTANDARD LVCMOS33 [get_ports {F}]

```

Vivado 平台上的仿真实验如下：



可以看到，当 S1,S2,S3 由奇数个为 1 时，F 值为 1，否则 F 值为 0，证明该代码无误。

实验板上验证结果为：当有奇数个开关闭合时，小灯亮，否则不亮，与两次仿真结果一致。

注：忘记贴上板结果了，很可能是扣分点！

## 四、讨论、心得

通过这一次实验，我对 Digital 软件和 Vivado 平台的熟悉程度进一步加深。在本次实验中，有了上一次实验的经验，效率高了很多，很快就能完成实验。