

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 15 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： EDA 实验平台与实验环境运用

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 13 日

一、实验目的

- 熟悉 Verilog HDL 语言并能用其建立基本的逻辑部件，在 Digital 软件和 Vivado 平台进行输入、编辑、调试、行为与仿真与综合后功能仿真
- 熟悉掌握 SWORD FPGA 开发平台，同时在 Vivado 平台上进行时序约束、引脚约束及映射布线后时序仿真
- 运用 Vivado 将设计验证后的代码下载到实验板上，并在实验板上验证

二、操作方法与实验步骤

1. 设计简单组合逻辑电路

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。

实验步骤如下：

- 分析楼道灯的事件行为，用组合电路实现，用拨动开关作为电路输入 S_1, S_2, S_3 ，电路输出为 F 。
- 变量赋值：开关往下为 1，往上为 0；输出灯亮为 1，灯暗为 0。
- 编写真值表，并根据真值表分析输入与输出之间的关系，并使用 Digital 软件绘制出电路图。
- 电路图调试无误后将其导出为 Verilog 代码，并在 Vivado 上进行后续的仿真以及在实验板上验证。

2. 设计简单时序逻辑电路

需要解决的问题：增加控制要求，灯打开后，延时若干秒自动关闭，请重新设计楼道灯的控制电路。

实验步骤如下：

- 分析楼道灯的事件行为，用时序电路实现，用拨动开关作为电路输入 S_1, S_2, S_3 ，电路输出为 F 。
- 变量赋值：开关往下为 1，往上为 0，开关拨上以后灯亮，开关拨下以后开

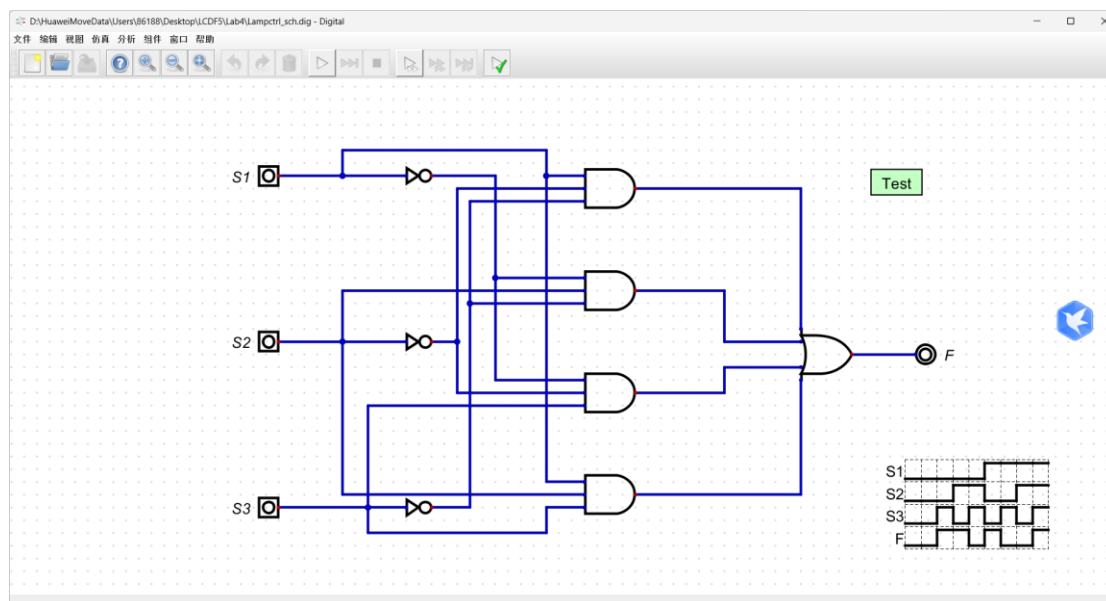
始计时，延迟 2s 以后灯灭；输出灯亮为 1，灯暗为 0。

- 编写 Verilog 代码，并在 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. 设计简单组合逻辑电路

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：

测试数据				
1	S1	S2	S3	F
2	0	0	0	0
3	0	0	1	1
4	0	1	0	1
5	0	1	1	0
6	1	0	0	1
7	1	0	1	0
8	1	1	0	0
9	1	1	1	1

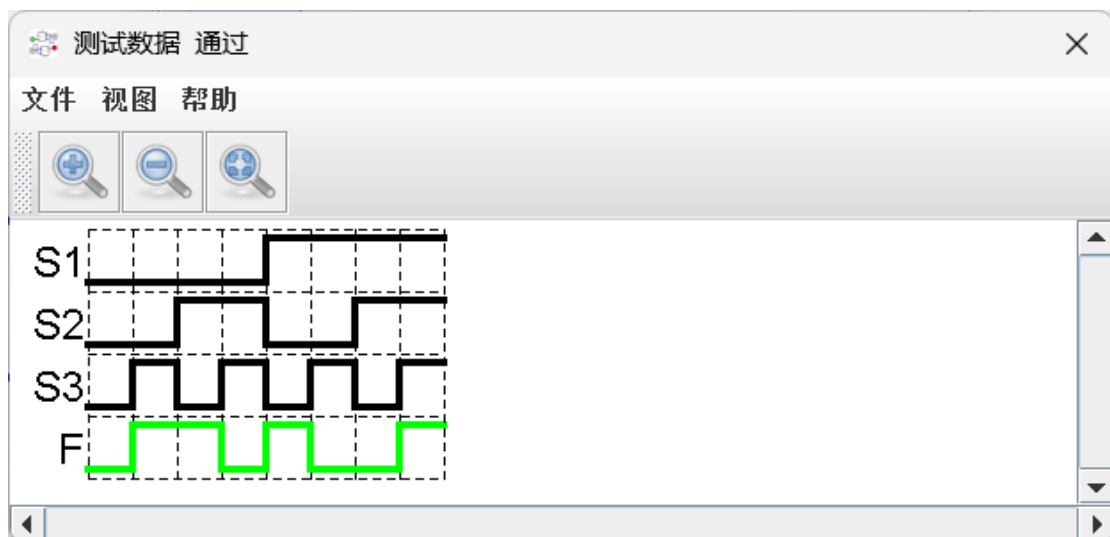
帮助 取消 运行测试用例(当前电路) 确定

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



可以观察到该输出值符合预期，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 Lampctrl_sch (
    input S1,
    input S2,
    input S3,
    output F
);

```

```

    assign F = ((S1 & ~ S2 & ~ S3) | (~ S1 & S2 & ~ S3) | (~ S1 & ~ S2 &
S3) | (S1 & S2 & 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：

```

`timescale 1ns / 1ps
module Lampctrl_sch_testbench();
//2、定义信号类似
//与设计源文件对应，对应规则：一般输入信号定义为reg，输出信号定义为wire

    // Inputs
    reg S1;
    reg S2;
    reg S3;

// Output
    wire F;

//3、例化设计源文件
//注意第一个名字为设计源文件名，第二个满足源文件命名规则即可，这里为了方便起见，命名源文件名_UUT。

// Instantiate the UUT
    Lampctrl_sch Lampctrl_sch_UUT (
        .S1(S1),
        .S2(S2),
        .S3(S3),
        .F(F)
    );
//4、添加激励（测试条件）

// Initialize Inputs
initial begin
//5、测试条件代码是顺序执行的
    S1 = 0;
    S2 = 0;
    S3 = 0;
    #50 S1 = 1;
//6、#50 表示延迟 50ns
    #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

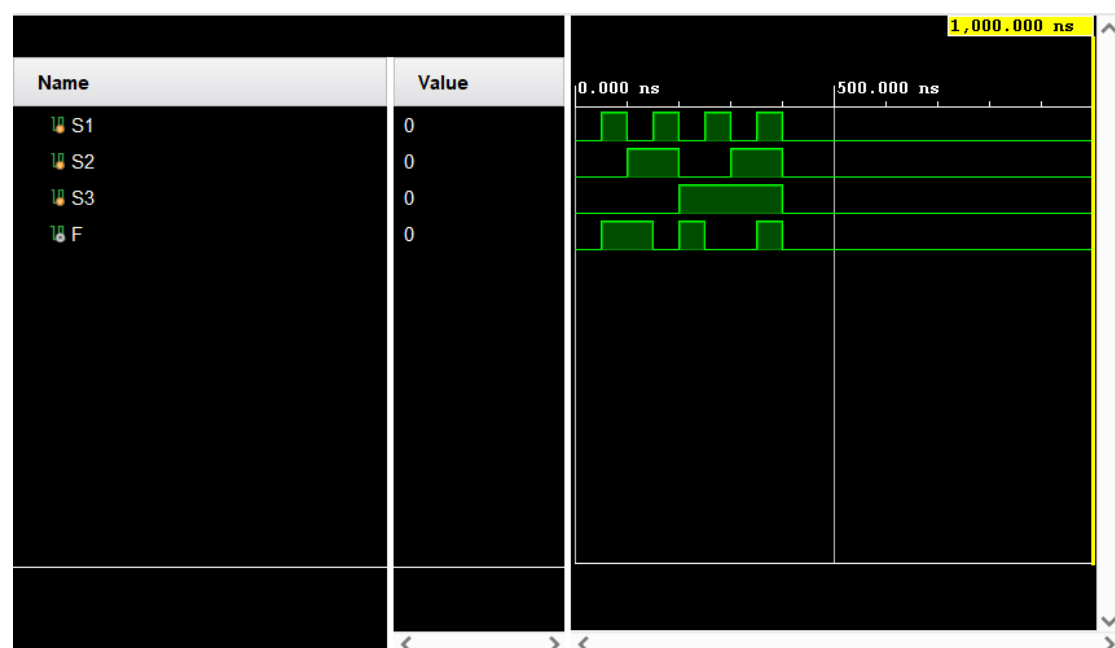
```

```

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：



其波形图与 Digital 软件上的一致。

实验板上验证结果为：当有 1 个或 3 个开关处于闭合状态时，灯亮；否则灯灭。与两次仿真结果均一致。

2. 设计简单时序逻辑电路

自己编写的 Verilog 代码、仿真代码、约束文件如下：

自己编写的 Verilog 代码：

```
`timescale 1ns / 1ps
module Lampctrl_HDL(
    input wire clk,
    input wire S1,
    input wire S2,
    input wire S3,
    output wire F
);

    parameter C_NUM = 8;
    parameter C_MAX = 8'hFF;

    reg [C_NUM-1:0] count;
    wire [C_NUM-1:0] c_next;
    wire w;

    initial begin //初始化
        count = C_MAX;
    end
    //button pressed
    assign w=S1||S2||S3;

    //Lamp Logic
    assign F = (count < C_MAX) ? 1'b1 : 1'b0;

    //count Logic
    always@(posedge clk)
    begin
        if(w == 1'b1)
            count = 0;
        else if(count < C_MAX)
            count = c_next;
    end
    //next Logic
    assign c_next = count + 8'b1;

endmodule
```

仿真代码：

```
module Lampctrl_HDL_testbench;
    // Inputs
    reg clk;
```

```

reg S1;
reg S2;
reg S3;
// Outputs
wire F;

// Instantiate the Unit Under Test (UUT)
Lampctrl_HDL 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

```

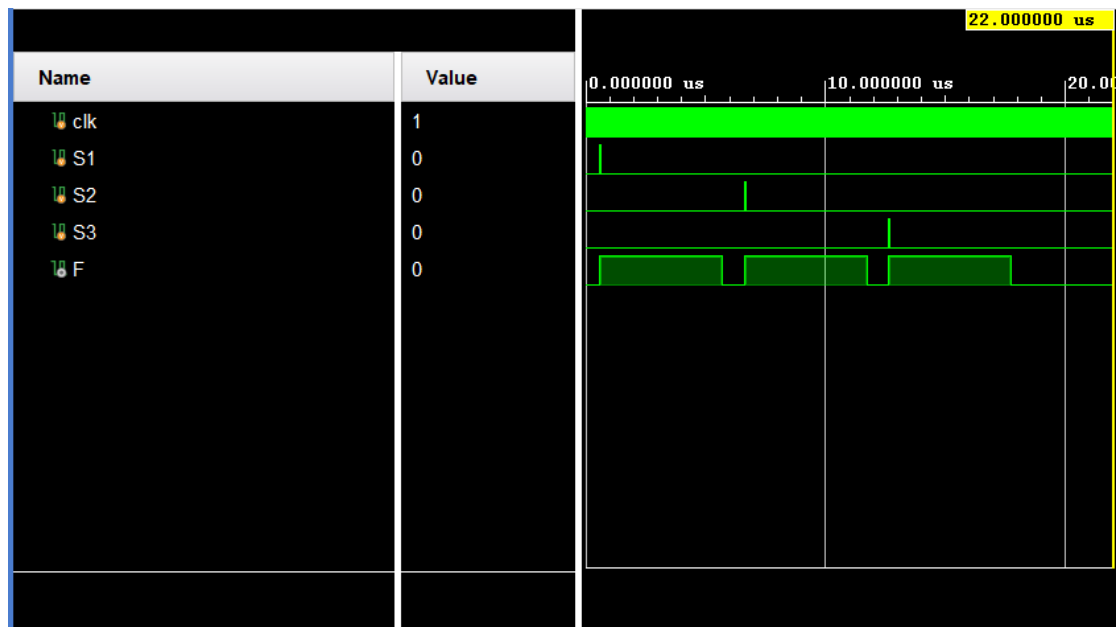
约束文件：


```

1 create_clock -name clk100MHZ -period 10.0 [get_ports {clk}]
2 set_property PACKAGE_PIN AC18 [get_ports {clk}]
3 set_property IOSTANDARD LVCMOS18 [get_ports {clk}]
4 set_property PACKAGE_PIN AA10 [get_ports {S1}]
5 set_property IOSTANDARD LVCMOS15 [get_ports {S1}]
6 set_property PACKAGE_PIN AB10 [get_ports {S2}]
7 set_property IOSTANDARD LVCMOS15 [get_ports {S2}]
8 set_property PACKAGE_PIN AA13 [get_ports {S3}]
9 set_property IOSTANDARD LVCMOS15 [get_ports {S3}]
10 set_property PACKAGE_PIN W23 [get_ports {LED[0]}]
11 set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]
12 set_property PACKAGE_PIN AB26 [get_ports {LED[1]}]
13 set_property IOSTANDARD LVCMOS33 [get_ports {LED[1]}]
14 set_property PACKAGE_PIN Y25 [get_ports {LED[2]}]
15 set_property IOSTANDARD LVCMOS33 [get_ports {LED[2]}]
16 set_property PACKAGE_PIN AA23 [get_ports {LED[3]}]
17 set_property IOSTANDARD LVCMOS33 [get_ports {LED[3]}]
18 set_property PACKAGE_PIN Y23 [get_ports {LED[4]}]
19 set_property IOSTANDARD LVCMOS33 [get_ports {LED[4]}]
20 set_property PACKAGE_PIN Y22 [get_ports {LED[5]}]
21 set_property IOSTANDARD LVCMOS33 [get_ports {LED[5]}]
22 set_property PACKAGE_PIN AE21 [get_ports {LED[6]}]
23 set_property IOSTANDARD LVCMOS33 [get_ports {LED[6]}]
24 set_property PACKAGE_PIN AF24 [get_ports {F}]
25 set_property IOSTANDARD LVCMOS33 [get_ports {F}]

```

Vivado 平台上的仿真实验结果如下：



可以看到，灯确实延时熄灭了，说明编写的 Verilog 代码是正确的。延长亮灯时间后在实验板上验证，实验板上的验证结果为：当人快速拨动开关（闭合后马上断开）时，灯并不会迅速熄灭，而是过一段时间后才熄灭，这与仿真实验的结果是一致的。

注：忘记贴板上板结果了，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 平台更加熟悉了，也学会了利用电路图转为 Verilog 代码的方法以及 Verilog 代码的基本语法。在本次实验中，由于我的计算机名是中文，导致在导出 bit 文件的时候遇到了阻碍，在当时反复调试都无法调试正确，因此我在实验课上最后用的是其他同学的电脑来完成的。回去后我修改了计算机名，现在可以正常使用了。