

浙江大学

本科实验报告

课程名称： 数字逻辑设计

姓 名： 蒋奕

学 院： 计算机学院

系： 计算机系

专 业： 计算机科学与技术

学 号： 3210103803

指导教师： 马德

2022 年 10 月 10 日

浙江大学实验报告

课程名称: 数字逻辑设计 实验类型: 绘图实验

实验项目名称: EDA 实验平台与实验环境运用

学生姓名: 蒋奕 专业: 计算机科学与技术 学号: 3210103803

同组学生姓名: 任庭旭 指导老师: 马德

实验地点: 东 4-509 实验日期: 2022 年 10 月 10 日

一、实验目的:

- ①熟悉 Verilog HDL 语言并能用其建立基本的逻辑部件, 在 Xilinx ISE 平台进行输入、编辑、调试、行为与仿真与综合后功能仿真。
- ②熟悉掌握 SWORD FPGA 开发平台, 同时在 ISE 平台上进行时序约束、引脚约束及映射布线后时序仿真
- ③运用 Xilinx ISE 具将设计验证后的代码下载到实验板上, 并在实验板上验证掌握二极管导通和截止的概念

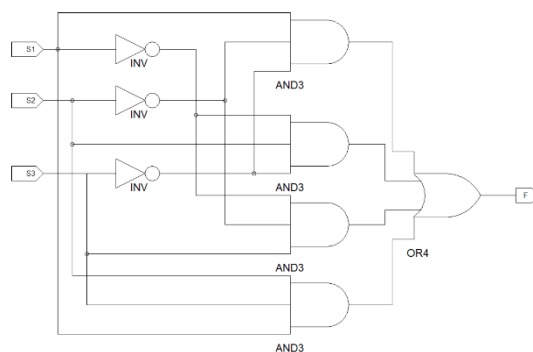
实验一: 设计简单组合逻辑电路, 采用图形输入逻辑功能描述

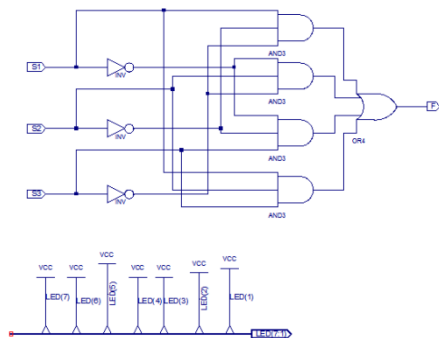
分析楼道灯的事件行为, 用组合电路实现, 用拨动开关作为电路输入 S_1, S_2, S_3 , 电路输出为 F 。变量赋值: 开关往下为 1, 往上为 0; 输出灯亮为 1, 灯暗为 0。

① 画出真值表和电路图如下

- ② 建立楼道控制的工程: LampCtrl_sch.ise, 创建原理图文件: LampCtrl.sch, 输入楼道灯控逻辑电路, 查看输入电路的硬件描述代码, 建立基准测试波形文件: LampCtrl_sim.tbw, 仿真激励输入, 建立用户时序约束并为模块的端口指定引脚分配, 建立用户时序约束并为模块的端口指定引脚分配, 设计实现并检查约束结果

S_3	S_2	S_1	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





进入 K7. ucf 编辑窗口，输入以下代码：

```
NET "S1" LOC = AA10 | IOSTANDARD = LVCMOS15;
NET "S2" LOC = AB10 | IOSTANDARD = LVCMOS15;
NET "S3" LOC = AA13 | IOSTANDARD = LVCMOS15;
NET "F" LOC = AF24 | IOSTANDARD = LVCMOS33;
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;
```

实验二：设计简单时序逻辑电路，采用 Verilog 代码输入逻辑功能描述

- ① 建立楼道控制的工程：LampCtrl_HDL.ise 高低电平通过 S1~S6 拨位开关产生，创建 Verilog 输入源文件 LampCtrl.v，输入楼道灯控逻辑电路 Verilog HDL 代码，建立基准测试波形文件：LampCtrl_sim.tbw，仿真激励输入波形为便于 仿真，LampCtrl.v 代码中计数器位数采用的是 8 位长，建立用户时序约束并为模块的端口指定引脚分配，将 LampCtrl.v 代码中计数器位数改成 28 位。
- ② 将生成 Bit 文件下载到 SWORD 实验板 在 SWORD 板上物理运行根据 I/O 约束定义和交互按钮操作和显示，板上通过按键开关，查看灯的变化是否正确，验证设计是否成功

仿真激励输入波形

为便于仿真，LampCtrl.v 代码中计数器位数采用的是 8 位长

```
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

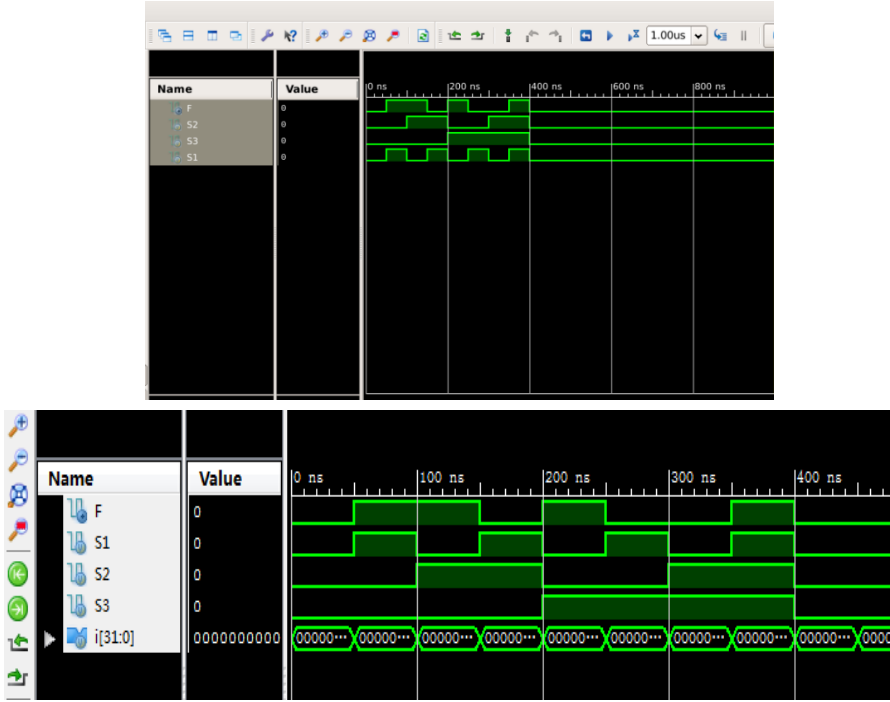
always begin
    #10 clk = 0;
    #10 clk = 1;
end
endmodule
```

建立引脚约束文件 k7. ucf ，输入代码如下

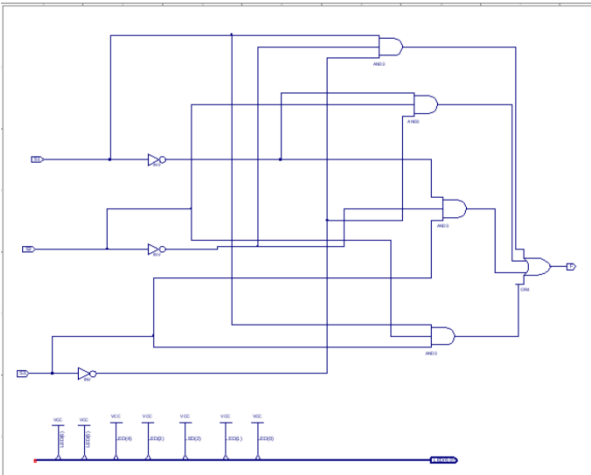
```
NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
NET "S1" LOC = AA10 | IOSTANDARD = LVCMOS15;
NET "S2" LOC = AB10 | IOSTANDARD = LVCMOS15;
NET "S3" LOC = AA13 | IOSTANDARD = LVCMOS15;
NET "F" LOC = AF24 | IOSTANDARD = LVCMOS33;
NET "LED[0]" LOC = W23 | IOSTANDARD = LVCMOS33;
NET "LED[1]" LOC = AB26 | 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;
```

二、实验数据记录和处理

实验一：设计简单组合逻辑电路，采用图形输入逻辑功能描述



仿真波形如上图所示



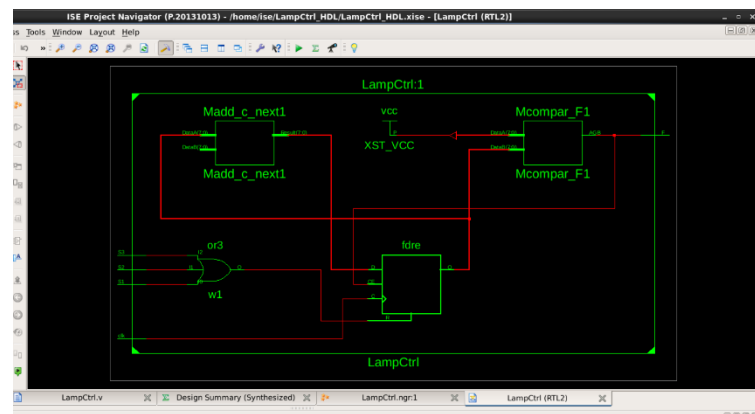
自主作图如上图所示

Design Overview	Pin Number	Signal Name	Pin Usage	Pin Name	Direction	ID Standard	ID Bank	Drive Number	Drive Rate	Termination	IDB Delay	Voltage	Constraint
Summary	1	A1	GPIO										
IOB Prop...	2	A2	GPIO										
Module L...	3	A3	GPIO	IOB335	UNUSED								
Timing C...	4	A4	GPIO	IOB335	UNUSED								
IOB Prop...	5	A5	GPIO										
Module L...	6	A6	GPIO										
Timing C...	7	A7	GPIO										
IOB Prop...	8	A8	GPIO	IOB335	UNUSED								
Module L...	9	A9	GPIO	IOB335	UNUSED								
Timing C...	10	A10	GPIO	IOB335	UNUSED								
IOB Prop...	11	A11	GPIO	IOB335	UNUSED								
Module L...	12	A12	GPIO	IOB335	UNUSED								
Timing C...	13	A13	GPIO	IOB335	UNUSED								
IOB Prop...	14	A14	GPIO	IOB335	UNUSED								
Module L...	15	A15	GPIO	IOB335	UNUSED								
Timing C...	16	A16	GPIO										
IOB Prop...	17	A17	GPIO	IOB335	UNUSED								
Module L...	18	A18	GPIO	IOB335	UNUSED								
Timing C...	19	A19	GPIO	IOB335	UNUSED								
IOB Prop...	20	A20	GPIO	IOB335	UNUSED								
Module L...	21	A21	GPIO	IOB335	UNUSED								
Timing C...	22	A22	GPIO	IOB335	UNUSED								
IOB Prop...	23	A23	GPIO	IOB335	UNUSED								
Module L...	24	A24	GPIO	IOB335	UNUSED								
Timing C...	25	A25	GPIO	IOB335	UNUSED								
IOB Prop...	26	A26	GPIO	IOB335	UNUSED								
Module L...	27	A27	GPIO	IOB335	UNUSED								
Timing C...	28	A28	GPIO	IOB335	UNUSED								
IOB Prop...	29	A29	GPIO	IOB335	UNUSED								
Module L...	30	A30	GPIO	IOB335	UNUSED								
Timing C...	31	A31	GPIO	IOB335	UNUSED								
IOB Prop...	32	A32	GPIO	IOB335	UNUSED								
Module L...	33	A33	GPIO	IOB335	UNUSED								
Timing C...	34	A34	GPIO	IOB335	UNUSED								
IOB Prop...	35	A35	GPIO	IOB335	UNUSED								
Module L...	36	A36	GPIO	IOB335	UNUSED								
Timing C...	37	A37	GPIO	IOB335	UNUSED								
IOB Prop...	38	A38	GPIO	IOB335	UNUSED								
Module L...	39	A39	GPIO	IOB335	UNUSED								
Timing C...	40	A40	GPIO	IOB335	UNUSED								
IOB Prop...	41	A41	GPIO	IOB335	UNUSED								
Module L...	42	A42	GPIO	IOB335	UNUSED								
Timing C...	43	A43	GPIO	IOB335	UNUSED								
IOB Prop...	44	A44	GPIO	IOB335	UNUSED								
Module L...	45	A45	GPIO	IOB335	UNUSED								
Timing C...	46	A46	GPIO	IOB335	UNUSED								
IOB Prop...	47	A47	GPIO	IOB335	UNUSED								
Module L...	48	A48	GPIO	IOB335	UNUSED								
Timing C...	49	A49	GPIO	IOB335	UNUSED								
IOB Prop...	50	A50	GPIO	IOB335	UNUSED								
Module L...	51	A51	GPIO	IOB335	UNUSED								
Timing C...	52	A52	GPIO	IOB335	UNUSED								
IOB Prop...	53	A53	GPIO	IOB335	UNUSED								
Module L...	54	A54	GPIO	IOB335	UNUSED								
Timing C...	55	A55	GPIO	IOB335	UNUSED								
IOB Prop...	56	A56	GPIO	IOB335	UNUSED								
Module L...	57	A57	GPIO	IOB335	UNUSED								
Timing C...	58	A58	GPIO	IOB335	UNUSED								
IOB Prop...	59	A59	GPIO	IOB335	UNUSED								
Module L...	60	A60	GPIO	IOB335	UNUSED								
Timing C...	61	A61	GPIO	IOB335	UNUSED								
IOB Prop...	62	A62	GPIO	IOB335	UNUSED								
Module L...	63	A63	GPIO	IOB335	UNUSED								
Timing C...	64	A64	GPIO	IOB335	UNUSED								
IOB Prop...	65	A65	GPIO	IOB335	UNUSED								
Module L...	66	A66	GPIO	IOB335	UNUSED								
Timing C...	67	A67	GPIO	IOB335	UNUSED								
IOB Prop...	68	A68	GPIO	IOB335	UNUSED								
Module L...	69	A69	GPIO	IOB335	UNUSED								
Timing C...	70	A70	GPIO	IOB335	UNUSED								
IOB Prop...	71	A71	GPIO	IOB335	UNUSED								
Module L...	72	A72	GPIO	IOB335	UNUSED								
Timing C...	73	A73	GPIO	IOB335	UNUSED								
IOB Prop...	74	A74	GPIO	IOB335	UNUSED								
Module L...	75	A75	GPIO	IOB335	UNUSED								
Timing C...	76	A76	GPIO	IOB335	UNUSED								
IOB Prop...	77	A77	GPIO	IOB335	UNUSED								
Module L...	78	A78	GPIO	IOB335	UNUSED								
Timing C...	79	A79	GPIO	IOB335	UNUSED								
IOB Prop...	80	A80	GPIO	IOB335	UNUSED								
Module L...	81	A81	GPIO	IOB335	UNUSED								
Timing C...	82	A82	GPIO	IOB335	UNUSED								
IOB Prop...	83	A83	GPIO	IOB335	UNUSED								
Module L...	84	A84	GPIO	IOB335	UNUSED								
Timing C...	85	A85	GPIO	IOB335	UNUSED								
IOB Prop...	86	A86	GPIO	IOB335	UNUSED								
Module L...	87	A87	GPIO	IOB335	UNUSED								
Timing C...	88	A88	GPIO	IOB335	UNUSED								
IOB Prop...	89	A89	GPIO	IOB335	UNUSED								
Module L...	90	A90	GPIO	IOB335	UNUSED								
Timing C...	91	A91	GPIO	IOB335	UNUSED								
IOB Prop...	92	A92	GPIO	IOB335	UNUSED								
Module L...	93	A93	GPIO	IOB335	UNUSED								
Timing C...	94	A94	GPIO	IOB335	UNUSED								
IOB Prop...	95	A95	GPIO	IOB335	UNUSED								
Module L...	96	A96	GPIO	IOB335	UNUSED								
Timing C...	97	A97	GPIO	IOB335	UNUSED								
IOB Prop...	98	A98	GPIO	IOB335	UNUSED								
Module L...	99	A99	GPIO	IOB335	UNUSED								
Timing C...	100	A100	GPIO	IOB335	UNUSED								

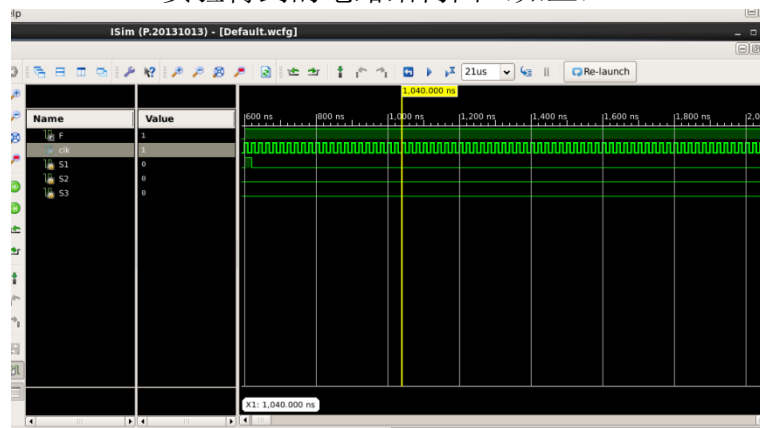
Report 如上图所示

注：灯泡的变化情况见附件“实验 1 结果.mp4”

实验二：设计简单时序逻辑电路，采用 Verilog 代码输入逻辑功能描述



实验得到的电路结构图（如上）



21 μ s 仿真波形如上图所示

注：灯泡的变化情况见附件“实验 2 结果. mp4”

三、实验结果与分析

实验测量结果基本和预测结果一样。可能因为实验设备老旧的原因使得一些结果反复测量都与预期值有一定偏差。

四、讨论、心得（选填）

实验过程中我体会到要认真对待每一次操作，正如我多次因为不小心而操作不熟练使得实验进程受阻。

附录：

下载时 Verilog 代码：

```
`timescale 1ns / 1ps
module LampCtrl(input wire clk, input wire S1, input wire S2, input wire S3, output LED[6:0], output wire F);
    parameter C_NUM = 28;
    parameter C_MAX = 28'hFFFFFF;
    reg [C_NUM-1:0] count;
    wire [C_NUM-1:0] c_next;
    initial begin//初始化
        count = C_MAX;
    end //need reset
    //button pressed
    assign LED = 7'b0000000;
    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
```