

实验四——EDA 实验平台与实验环境运用

实验报告

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼 509

实验日期： 2019 年 10 月 9 日

一、实验目的和要求

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

二、实验内容和原理

实验内容：

1. 熟悉 ISE 工具软件的运行环境与安装过程
2. 设计简单组合逻辑电路，采用图形输入逻辑功能描述，建立 FPGA 实现数字系统的 Xilinx ISE 设计管理工程，并进行编辑、调试、编译、行为仿真，时序约束、引脚指定（约束）、映射布线后时序仿真及 FPGA 编程代码下载与运行验证
3. 设计简单时序逻辑电路，采用 Verilog 代码输入逻辑功能描述，建立 FPGA 实现数字系统的 ISE 设计管理工程，并进行编辑、调试、编译、行为仿真，时序约束、引脚约束、映射布线后时序仿真及 FPGA 编程代码下载与运行验证

实验原理：

问题 1：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。

问题 2：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。

三、主要仪器设备

实验设备：

装有 ISE 14.7 的计算机	1 台
SWORD 开发板	1 套

实验材料：无

四、操作方法与实验步骤

问题一：

1. 新建楼道控制的工程文件，在此命名为 Dreamerryao_sch.ise（基本命名为 LampCtrl_sch.ise），具体步骤如下：

1、依次点击菜单 File → New Project...

2、在对话框中设置：

Project Name: Dreamerryao_sch Top-Level Source Type:Schematic

3、确认后，点击 Next 到设备属性页，设置：

Family: Kintex7 Device: XC7K160T Package: FFG676 Speed: -1

4、确认后，一直点击 Next 直到创建工程结束。

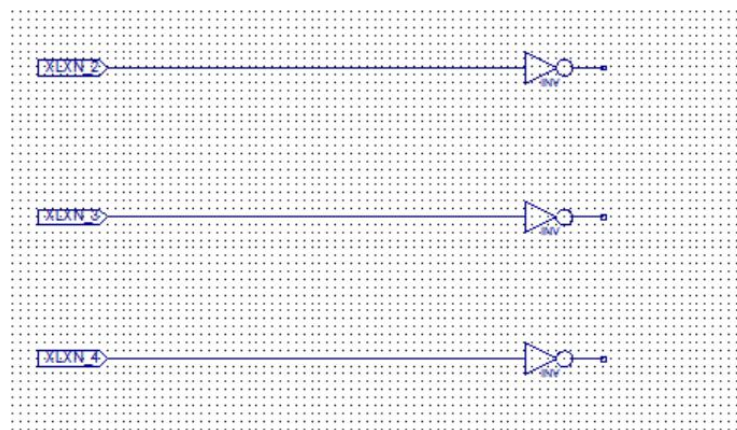
2. 创建原理图文件：Dreamerryao_lampctrl.sch

1. 在 sources 窗口右键，点击 new source, 选择 schematic 文件，命名为 Dreamerryao_lampctrl 即可。

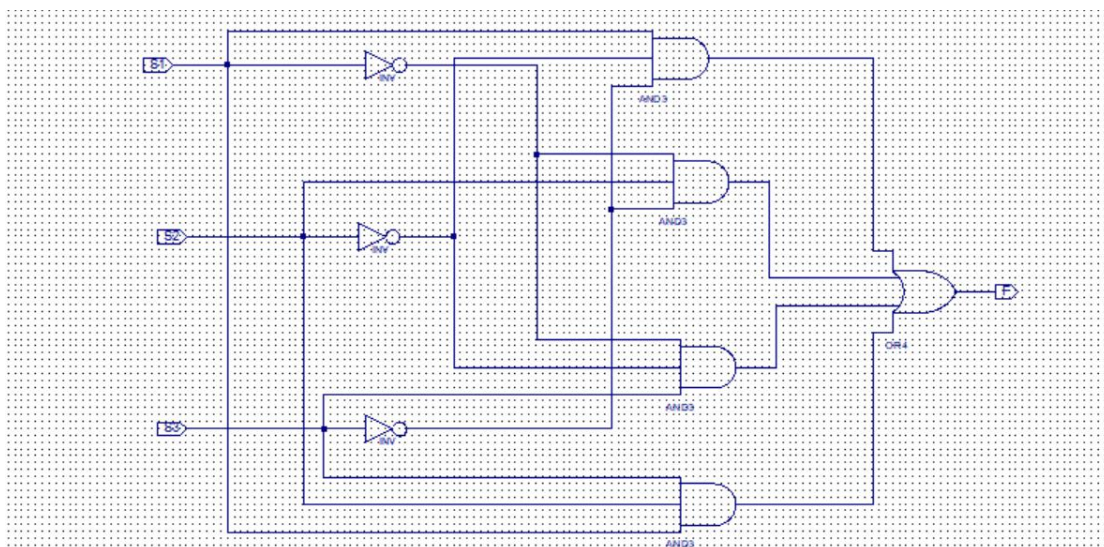
1. 在 Sources 窗口中选择 Symbols 选项卡，输入相关元器件名字，如 (inv, add3 等)，

注：此阶段需将面板先放大，否则拖入元器件过小，看不清楚。

配合 Schematic Editor 工具条输入原理图，如图



此图为 Inv 与 I/O marker 结合后产生，在此图基础上继续绘制得到图如下：



也可通过 add copy of source 将老师发给的文件直接导入（但不建议，最好自己动手）

3. 查看输入电路的硬件描述代码

在 Sources 窗口中选择 Sources for: Synthesis / Implementation, 选中 LampCtrl.sch 图标，在 Processes 窗口 Processes 选项卡中展开 Design Utilities 并双击 View HDL Functional Model，如图：

```
20 //
21 'timescale 1ns / 1ps
22
23 module Dreamerryao_lampctrl (S1,
24                               S2,
25                               S3,
26                               F);
27
28     input S1;
29     input S2;
30     input S3;
31     output F;
32
33     wire NS1;
34     wire NS2;
35     wire NS3;
36     wire S1NS2NS3;
37     wire S1S2NS3;
38     wire S2NS1NS3;
39     wire S3NS1NS2;
40
41     AND3 AND3_1 (.I0(NS3),
42                 .I1(NS2),
43                 .I2(S1),
44                 .O(S1NS2NS3));
45     AND3 AND3_2 (.I0(NS3),
46                 .I1(S2),
47                 .I2(NS1),
48                 .O(S2NS1NS3));
49     AND3 AND3_3 (.I0(S3),
50                 .I1(NS2),
51                 .I2(NS1),
52                 .O(S3NS1NS2));
53     AND3 AND3_4 (.I0(S1),
54                 .I1(S2),
55                 .I2(S3),
56                 .O(S1S2S3));
57     INV INV_1 (.I(S1),
58               .O(NS1));
59     INV INV_2 (.I(S2),
60               .O(NS2));
61     INV INV_3 (.I(S3),
62               .O(NS3));
63     OR4 OR4_1 (.I0(S1S2S3),
64                .I1(S3NS1NS2),
65                .I2(S2NS1NS3),
66                .I3(S1NS2NS3),
67                .O(F));
68 endmodule
```

图：Dreamerryao_lampctrl.sch 对应硬件描述代码

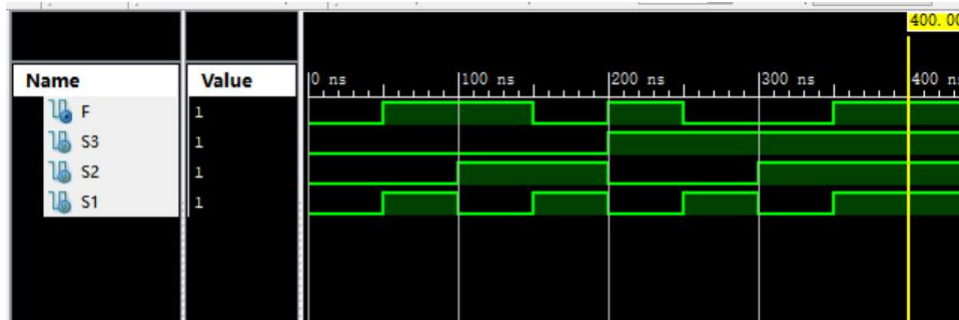
4. 建立基准测试波形文件: Dreamerryao_sim.tbw

- 1、在 Sources 窗口空白处的右键菜单中选择 New Source
- 2、在新建源文件向导中选择源类型为: Verilog Test Fixture, 输入文件名 Dreamerryao_sim, 并勾选 Add to Project
- 注: 点击 next 时会出现其对应文件, 选择 Dreamerryao_lampctrl 文件即可
- 3、点击 Finish 进入 Dreamerryao_sim.v 编辑窗口
- 4、更改 initialize inputs 下方代码, 方式一:

```
23 //  
24 // Initialize Inputs  
25 // `ifdef auto_init  
26     initial begin  
27         S3=0;S2=0; S1=0;#50;  
28         S1=1; #50;  
29         S2=1;S1=0; #50;  
30         S1=1; #50;  
31         S3=1;S2=0;S1=0; #50;  
32         S1=1; #50;  
33         S2=1;S1=0; #50;  
34         S1=1; #50;  
35     end  
36  
37 //`endif  
38 endmodule
```

注意: 将 25 行与 37 行注释掉

- 5、View 选择 Simulation 视图, Hierarchy 窗口中选择 Dreamerryao_lampctrl_Dreamerryao_lampctrl_sch_tb, Process 窗口中选择 Simulate Behavioral Model, 查看对应仿真图:



方式一对应仿真图

- 6、更改 4 中代码, 方式二:

```

24 // Initialize Inputs
25 //`ifdef auto_init
26
27 integer i;
28 initial begin
29     for(i=0;i<=8;i=i+1)begin
30         {S3,S2,S1} <= i;
31         #50;
32     end
33 end
34
35
36 //`endif
37 endmodule

```

方式二对应代码

7、重复步骤 5，查看对应仿真图：



方式二对应仿真图

5. 建立用户时序约束并为模块的端口指定引脚分配

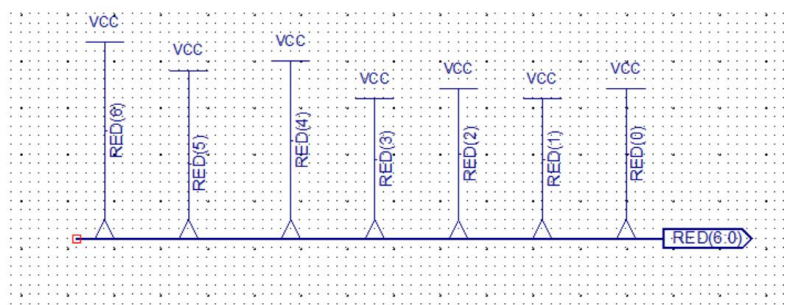
- 1、在 Sources 窗口空白处的右键菜单中选择 New Source
- 2、在新建源文件向导中选择源类型为：Implementation Constraints File，输入文件名 Dreamerryao，并勾选 Add to Project
- 3、点击 Finish 进入 Dreamerryao.ucf 编辑窗口，输入以下代码：

```

NET"S1"LOC=AA10 | IOSTANDARD=LVCOS15;#电压说明
NET"S2"LOC=AB10 | IOSTANDARD=LVCOS15;
NET"S3"LOC=AA13 | IOSTANDARD=LVCOS15;
NET"F"LOC=AF24 | IOSTANDARD=LVCOS33 ;#D8
#NET"Buzzer"LOC=AF25 | IOSTANDARD=LVCOS33 ;
#NET"LED[0]"LOC=W23 | IOSTANDARD=LVCOS33 ;#D1
#NET"LED[1]"LOC=AB26 | IOSTANDARD=LVCOS33 ;#D2
#NET"LED[2]"LOC=Y25 | IOSTANDARD=LVCOS33 ;#D3
#NET"LED[3]"LOC=AA23 | IOSTANDARD=LVCOS33 ;#D4
#NET"LED[4]"LOC=Y23 | IOSTANDARD=LVCOS33 ;#D5
#NET"LED[5]"LOC=Y22 | IOSTANDARD=LVCOS33 ;#D6
#NET"LED[6]"LOC=AE21 | IOSTANDARD=LVCOS33 ;#D7

```

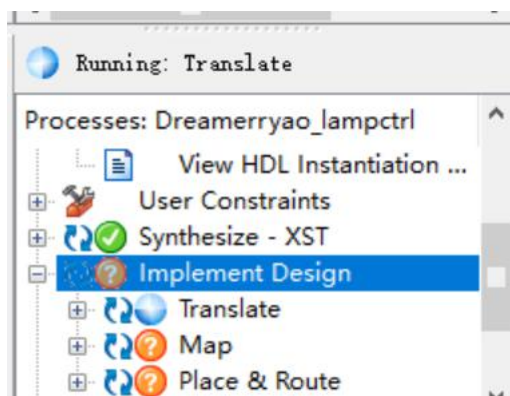
并且加入以下原理图：



总线图

6. 设计实现并检查约束结果

1、在 Sources 窗口中选择 Synthesis/Implementation，选中 Dreamerryao_lampctrl;在 Processes 窗口下选择 Implement Design，进行物理转换、平面布图、映射、物理布线等 FPGA 目标格式实现文件生成。



正在进行 translate 过程

最后在设计摘要文档中有如下结果：

	Pin Number	Signal Name	Pin Usage	Pin Name	Direction	IO Standard	IO Bank Number	Drive (mA)	Slew Rate	Termination
1	A1			GND						
2	A2			GND						
3	A3		OPAD	MGTXTXN3_116	UNUSED					
4	A4		OPAD	MGTXTXP3_116	UNUSED					
5	A5			GND						
6	A6			GND						
7	A7			GND						
8	A8		IOB...	IO_L9N_T1_DQS_16	UNUSED		16			
9	A9		IOB...	IO_L9P_T1_DQS_16	UNUSED		16			
10	A10		IOB...	IO_L22N_T3_16	UNUSED		16			
11	A11			VCCO_16			16			
12	A12		IOB...	IO_L24N_T3_16	UNUSED		16			
13	A13		IOB...	IO_L24P_T3_16	UNUSED		16			
14	A14		IOB...	IO_L21N_T3_DQS_16	UNUSED		16			
15	A15		IOB...	IO_L23N_T3_16	UNUSED		16			
16	A16			GND						
17	A17		IOB...	IO_L3N_T0_DQS_AD1N_15	UNUSED		15			
18	A18		IOB...	IO_L2P_T0_AD8P_15	UNUSED		15			
19	A19		IOB...	IO_L2N_T0_AD8N_15	UNUSED		15			
20	A20		IOB...	IO_L8N_T1_D12_14	UNUSED		14			
21	A21			VCCO_14			14			
22	A22		IOB...	IO_L2N_T0_D03_14	UNUSED		14			
23	A23		IOB...	IO_L4P_T0_D04_14	UNUSED		14			
24	A24		IOB...	IO_L4N_T0_D05_14	UNUSED		14			

摘要过程的结果

双击 Boundary Scan 弹出下载编辑窗口

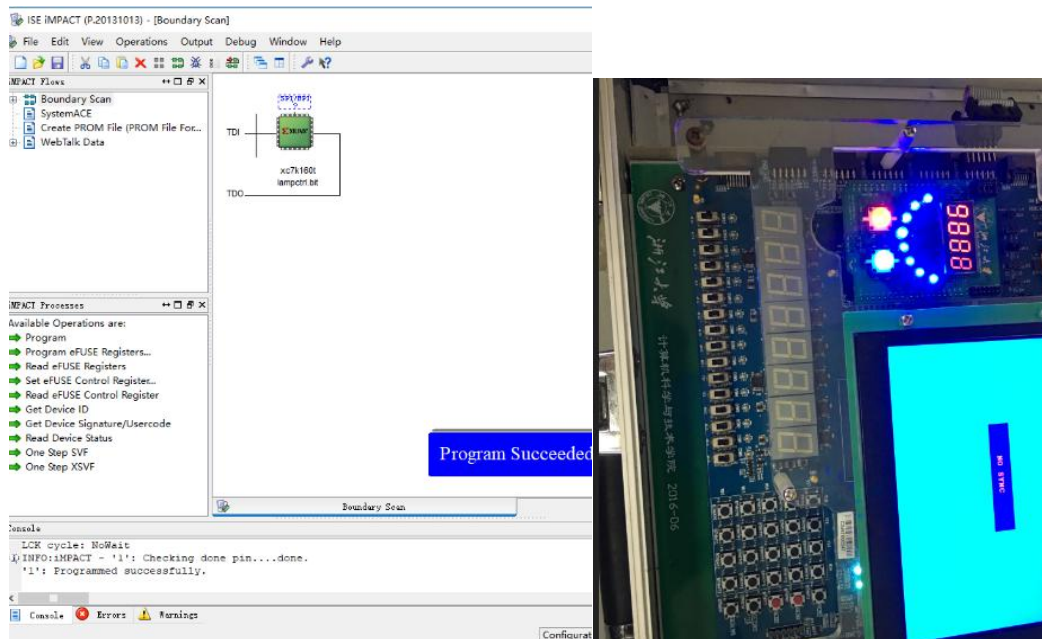
鼠标右键选择 Initialize Chain，系统自动查找已连接在电脑上的开发

平台 JTAG 下载链

接下来出现 Assign Configuration Files 对话框。这时从文件列表中选择 Dreamerryao_lampctrl.bit 文件，将会为 JTAG chain 上的 xc7k160t 设备指定配置文件；在弹出的 Attach SPI or PRI PROM 对话框弹出，点击 NO 按钮；在弹出的“Device Programming Property 对话框，选择 OK 按钮即可。

右键点击 xc7k160t 设备图标，选择菜单项 Program 后即可对硬件设备进行下载编程

通过验证:发现满足要求



问题二:

1. 建立楼道控制的工程: Dreamerryao_lampctrl_HDL.ise:

- 1、依次点击菜单 File → New Project...
- 2、在对话框中设置如下:
Project Name: Dreamerryao_lampctrl_HDL
Top-Level Source Type: HDL
- 3、确认后，点击 Next 到设备属性页，设置:
Family: Kintex7
Device: XC7K160T
Package: FFG676
Speed: -1

4、确认后，一直点击 Next 直到创建工程结束

2. 输入楼道灯控逻辑电路 Verilog HDL 代码

- 1、仿照问题一的步骤 2 建立 lampctrl 文件，注意文件类型选择为

verilog module

- 2、在源代码编辑器，输入代码，代码如下图
- 3、检查输入代码的语法规则，
并排除输入错误

```
////////////////////////////////////////  
module lampctrl(input wire clk,  
    input wire S1,  
    input wire S2,  
    input wire S3,  
    output wire F  
);  
  
    parameter C_NUM = 28;  
    parameter C_MAX = 28'hFFF_FFFF;  
  
    reg [C_NUM-1:0] count;  
    wire [C_NUM-1:0] c_next;  
  
    initial begin //初始化  
        count = C_MAX;  
    end  
  
    assign w=S1^S2^S3;  
  
    assign F = ((count < C_MAX) ? 1'b1 : 1'b0);  
  
    always@(posedge clk)  
    begin
```

3.楼道控制电路代码的综合

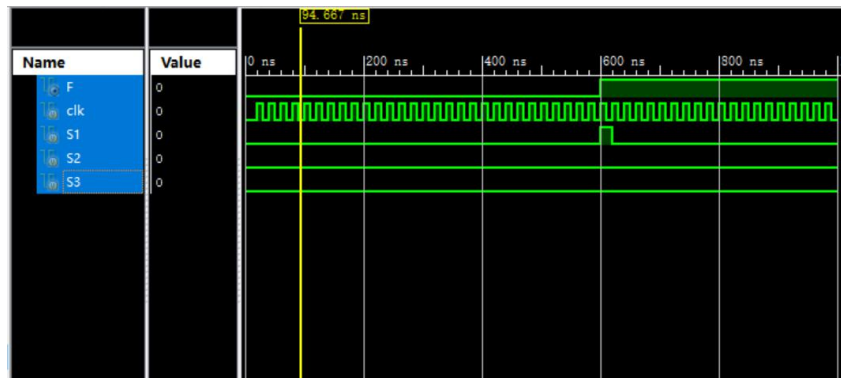
- 1、在 Sources 窗口选中文件 LampCtrl.v;
- 2、在 Processes 窗口运行 Synthesis XST → View RTL Schematic
- 3、检查综合的电路结构是否与设计目标一致;
4. 仿照问题一步骤 4，进行建立基准测试波形文件 lampctrl_sim
 - 1、在 Sources 窗口空白处的右键菜单中选择 New Source
 - 2、在新建源文件向导中选择源类型为：Verilog Test Fixture，输入文件名 LampCtrl_sim，并勾选 Add to Project
 - 3、点击 Finish 进入 LampCtrl_sim.v 编辑窗口，修改代码如下：


```

43 //
44
45 initial begin
46     // Initialize Inputs
47     clk = 0;
48     S1 = 0; S2 = 0; S3 = 0;
49
50     #600 S1 = 1;
51     #20 S1 = 0;
52     #6000 S2 = 1;
53     #20 S2 = 0;
54     #6000 S3 = 1;
55     #20 S3 = 0;
56
57 end
58
59 endmodule

```

4、得到的对应仿真图：



5. 仿照问题一步骤 5 进行建立用户时序约束并为模块的端口指定引脚分配
注意：此时需要将 LampCtrl.v 中计数器位数改为 28 位，以适应实验室板子

```

parameter C_NUM = 28;
parameter C_MAX = 28'hFFF_FFFF;
.ucf 文件配置代码如下：
NET"clk"LOC = AC18 | IOSTANDARD=LVC MOS18 ;
NET"S1"LOC = AA10 | IOSTANDARD=LVC MOS15;
NET"S2"LOC = AB10 | IOSTANDARD=LVC MOS15;
NET"S3"LOC = AA13 | IOSTANDARD=LVC MOS15;
NET"F"LOC = AF24 | IOSTANDARD=LVC MOS33 ;#D8

```

6. 仿照问题一步骤 6 进行下载到 sword 板上

-> Synthesize - XST, -> Implement design, -> Generate Programming File, 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。

五、实验结果与分析

参照上述实验步骤，建立自己的 Program, 上述步骤介绍用图即为本地 program 中得到。

六、讨论、心得

- 1.熟悉了 ISE 平台的基本使用方法，便于以后使用 ISE 平台进行作业
- 2.通过 HDL 与 SCH 两种方式，都进行了演练，花费时间较长
- 3.对整体流程都有了大致的了解，颇有成就感。