

计算机系统结构实验 Lab01: flowing light

黄正翔 520021910014

摘要

在 Lab01 中，我通过 Verilog 语言实现了 flowing_light 功能，学习了如何新建 module，与 simulation 文件，初步理解了 Vivado 的语法、项目流程、仿真方法与调试手段，收获颇丰。

目录

计算机系统结构实验 Lab01: flowing light.....	1
黄正翔 520021910014.....	1
摘要	1
1. 实验目的.....	2
2. 原理分析.....	2
2.1 Vivado 工程的基本组成	2
2.2 flowing light 的原理	2
3. 功能实现.....	2
4. 结果验证.....	3
4.1 测试用激励文件.....	3
4.2 reset 与基本逻辑的测试	3
4.3 调整控制逻辑以观察移位.....	4
5. 管脚约束.....	5
6. 总结与反思.....	5

1. 实验目的

- (1) 通过基础实验熟悉 Xilinx 逻辑设计工具 Vivado 开发环境;
- (2) 了解硬件描述语言 Verilog HDL 描述功能行为的逻辑;
- (3) 通过仿真检验电路设计是否预期;
- (4) 学习使用 I/O Planing 添加管脚约束;
- (5) 实现 flowing light 的功能;
- (6) 熟悉系统硬件开发的基本实验流程。

2. 原理分析

2.1 Vivado 工程的基本组成

- (1) design source .v 文件
- (2) simulation source .v 文件
- (3) constraints .xdc 文件 (上板验证所需的管脚约束文件)

2.2 flowing light 的原理

Flowing light 要求在一段时间内, 8 个 LED 灯依次轮流亮灭, 最后一个 LED 熄灭后, 第一个 LED 循环亮起。这个功能可以使用移位来实现控制。当然, 实验指导书中也给出了相应的实现。我的代码如下:

```
37 @ always @ (posedge clock)
38 @ begin
39 @     if(reset)
40 @         light_reg <= 8'h01;
41 @     else if(cnt_reg == 24'hffffff)
42 @         begin
43 @             if(light_reg == 8'h80)
44 @                 light_reg <= 8'h01;
45 @             else
46 @                 light_reg <= light_reg << 1;
47 @             end
48 @         end
49 @     end
```

可以看到控制 LED 的 light_reg 在每过 384 个时钟周期进行一次移位。

3. 功能实现

本实验比较简单, 基于上述的原理容易实现 flowing light 的功能。

在实现 flowing_light.v 后, 生成 flowing_light_tb.v 的激励文件用以仿真测试, 生成 lab01_xdc.xdc 的管脚约束用以练习。

4. 结果验证

4.1 测试用激励文件

首先，我按照实验导书上的要求，编写激励文件。

```
flowing_light u0(  
    .clock(clock),  
    .reset(reset),  
    .led(led));  
  
parameter PERIOD = 10;  
always #(PERIOD*2) clock = !clock;  
  
initial begin  
    clock = 1'b0;  
    reset = 1'b0;  
    #(PERIOD*2) reset = 1'b1;  
    #(PERIOD*4) reset = 1'b0;  
  
    //55ns reset = 1'h1.
```

可以看到时钟周期是 40 个时间单位。

4.2 reset 与基本逻辑的测试

首先，我进行仿真，以完成 figure 1 的要求，我的仿真结果如图 1，图 2 所示：

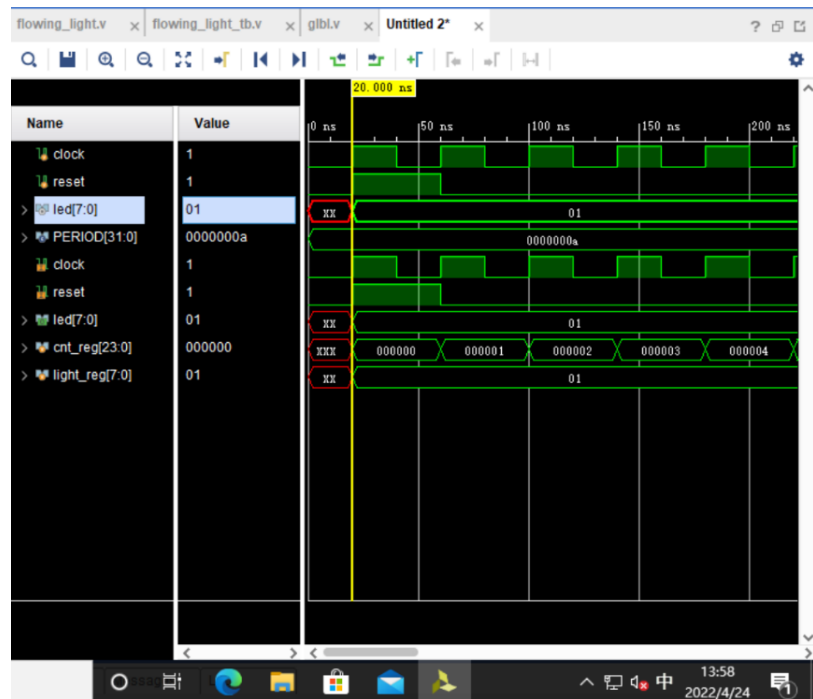


图 1 报告要求的 Figure 1

在图 1 中可以看到 reset, cnt_reg, light_reg 功能正常。

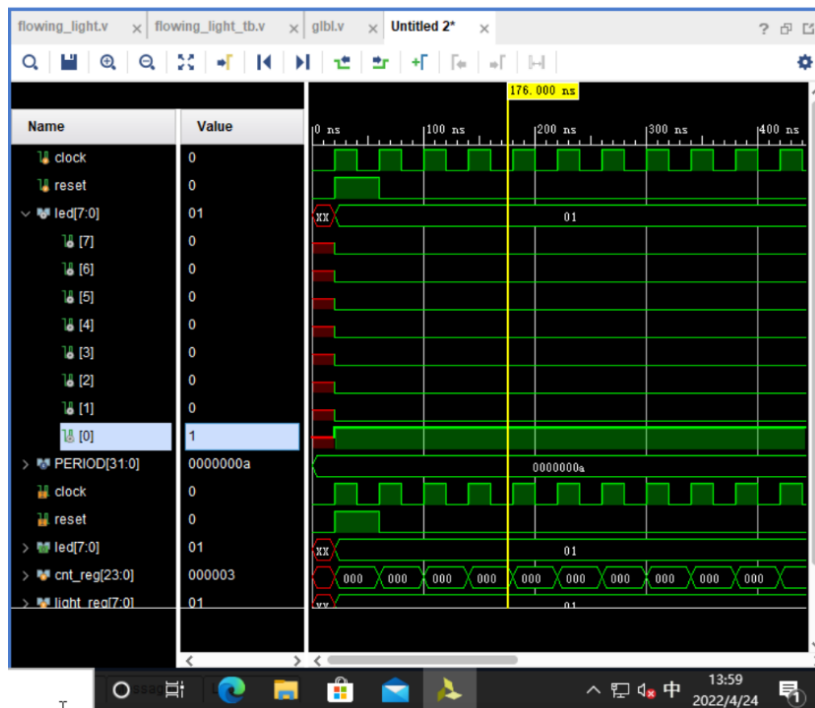


图 2 报告要求的 Figure 2

在图 2 中可以看到 led 每一位的情况。

本仿真运行周期不够，计数器并没加到 24 位全是 1 而波形显示早已结束。我们可以通过改变计数器计数值的控制逻辑，以便较快速达到左移条件，结果记录与 4.3 中。

4.3 调整控制逻辑以观察移位

```

39  if(reset)
40      light_reg <= 8'h01;
41  else if(cnt_reg == 4'hf)
42      begin
43          if(light_reg == 8'h80)
44              light_reg <= 8'h01;
45          else
46              light_reg <= light_reg << 1;
47          end
48      end

```

我将计数逻辑改为 `cnt_reg==1111` 时移位，即经过 16 个时钟周期就移位，在仿真模拟中观察到了移位，如图 3 所示：

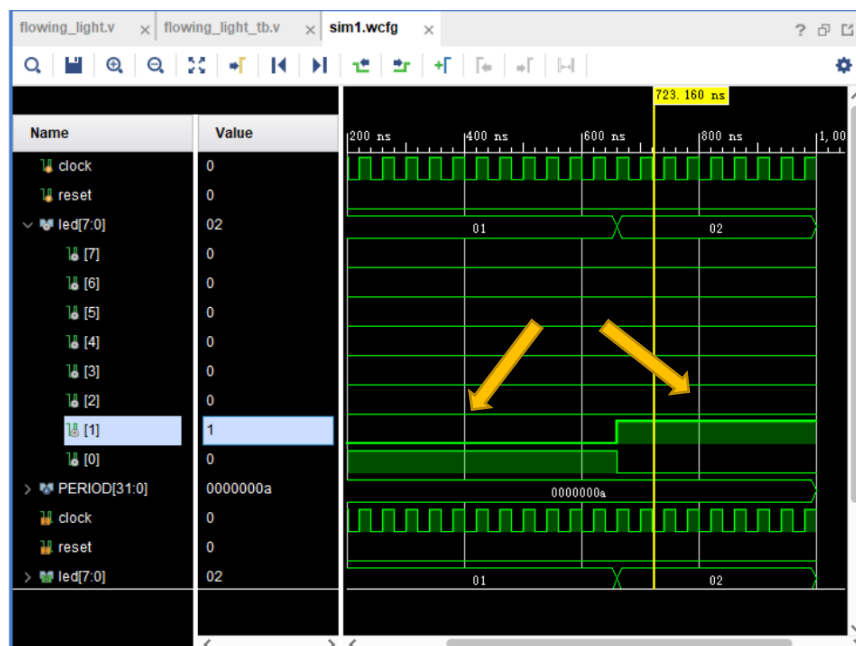


图 3 箭头所指展示了移位

这样，我通过改变计数器的控制逻辑在仿真中观察到了移位。

5. 管脚约束

我将管脚约束写在了 lab01_xdc.xdc 文件中。我也学到了如何简单地在控制台上设置管脚约束。在此就不多赘述。

6. 总结与反思

我之前在其他课程中接触过 Vivado 的开发环境，因此上手并不困难，也没有遇到什么障碍。本次实验不仅很好地为我回顾了 Verilog HDL 的基本语法以及项目开发流程，更为我纠正了一些过去对于 always 用法的错误理解，并且强化了我的仿真能力，真是太好了！感谢黄老师的课上答疑以及课程组准备的指导书。