

# 计算机系统结构实验报告 Lab01

殷浩 518030910095

2020 年 6 月 11 日

## 摘要

Lab01 中主要进行 FPGA 基础实验——LED Flow Water Light，学习软件的基本使用、调试方法，掌握设计的基本原理。

## 目录

目录	1
1 实验目的	2
2 原理分析	2
2.1 基础实验	2
2.1.1 Flowing_Light	2
2.1.2 Flowing_Light_tb	2
2.2 拓展部分	2
2.2.1 Flowing_Light	2
2.2.2 Flowing_Light_tb	2
3 代码说明	3
3.1 基础实验	3
3.1.1 Flowing_Light	3
3.1.2 Flowing_light_tb	4
3.2 拓展部分	4
4 仿真结果	5
5 仿真结果	6
6 鸣谢	6

## 1 实验目的

1. 熟悉 Xilinx 逻辑设计工具 Vivado 的基本操作
2. 掌握使用 VerilogHDL 进行简单的逻辑设计
3. 使用功能仿真
4. 使用 I/O Planning 添加管脚约束
5. 生成 Bitstream 文件
6. 上板验证（暂不进行）

## 2 原理分析

### 2.1 基础实验

基础实验要求使用仿真软件编写代码，实现模拟多个 LED 的流水开关效果

#### 2.1.1 Flowing\_Light

程序的主要信号及其功能如表1所示。

信号	输入/输出	功能
clock	输入	时钟信号，用于控制变量变化
reset	输入	重置信号，用于初始化程序
[7:0]led	输出	输出电平信号，控制不同 LED 灯的状态

表 1: 程序输入输出

#### 2.1.2 Flowing\_Light\_tb

这部分代码用于测试程序是否正确，该部分解释将放在代码说明相应内容中。

### 2.2 拓展部分

基础实验中，由于仿真时间不够，只能看到一个灯亮起（即信号变成高电平），所以在拓展部份中对代码做部分修改，使得能观察到两个或以上灯的开关状态。

#### 2.2.1 Flowing\_Light

程序的主要信号及其功能和表1所示相同。只需修改部分代码即可达到目的，因此本部分解释也将放在代码说明相关章节。

#### 2.2.2 Flowing\_Light\_tb

测试代码与基础实验完全相同。

## 3 代码说明

### 3.1 基础实验

#### 3.1.1 Flowing\_Light

程序循环检测时钟信号 clock 的变化,如果在时钟上升沿收到重置信号 reset,就把计数变量 cnt\_reg 设置为 0; 否则,计数变量自增。在时钟下降沿检测计数变量,达到最大值时更换亮起的 LED。

```
1  module flowing_light(  
2      input clock,  
3      input reset,  
4      output [7:0] led  
5  );  
6  
7      reg[24:0] cnt_reg;  
8      reg[7 : 0] light_reg;  
9  
10     always @ (posedge clock)  
11         begin  
12             if (reset)  
13                 cnt_reg <= 0;  
14             else  
15                 cnt_reg <= cnt_reg + 1;  
16         end  
17  
18     always @ (posedge clock)  
19         begin  
20             if (reset)  
21                 light_reg <= 8'h01;  
22                 //if quicker to light the second led, use following  
23                 //else if (cnt_reg == 4'hf)  
24             else if (cnt_reg == 24'hffffff)  
25                 begin  
26                     if (light_reg == 8'h80)  
27                         light_reg <= 8'h01;  
28                     else  
29                         light_reg <= light_reg << 1;  
30                 end  
31             end  
32             assign led = light_reg;  
33     endmodule  
34     end
```

### 3.1.2 Flowing\_light\_tb

调试程序实例化一个 flowing\_light 对象，固定周期改变时钟信号。

```
1  module flowing_light_tb(
2      );
3
4      reg clock;
5      reg reset;
6      wire [7:0] led;
7
8      flowing_light u0(
9          .clock(clock),
10         .reset(reset),
11         .led(led));
12
13     parameter PERIOD = 10;
14
15     always #(PERIOD*2) clock = !clock;
16
17     initial begin
18         clock = 1'b0;
19         reset = 1'b0;
20         #(PERIOD*2) reset = 1'b1;
21         #(PERIOD*4) reset = 1'b0;
22         // #580; reset = 1'b1;
23     end
24 endmodule
```

## 3.2 拓展部分

为了实现观察两个及以上的灯状态的变化，可以将代码中的计数变量 cnt\_reg 位数缩小，使其在更小的周期变化数达到最大值，从而控制灯状态的变化，修改后的代码如下：

```
1  reg[4:0] cnt_reg
2  reg[7 : 0] light_reg;
3
4  always @ (posedge clock)
5      begin
6          if (reset)
7              cnt_reg <= 0;
8          else
9              cnt_reg <= cnt_reg + 1;
10     end
```

```

11
12     always @ (posedge clock)
13         begin
14             if (reset)
15                 light_reg <= 8'h01;
16             else if (cnt_reg == 4'hf)
17                 begin
18                     if (light_reg == 8'h80)
19                         light_reg <= 8'h01;
20                     else
21                         light_reg <= light_reg << 1;
22                 end
23             end
24     assign led = light_reg;

```

当然，也可以通过修改 PERIOD 的值，使时钟信号变化周期减小达到目的。

## 4 仿真结果

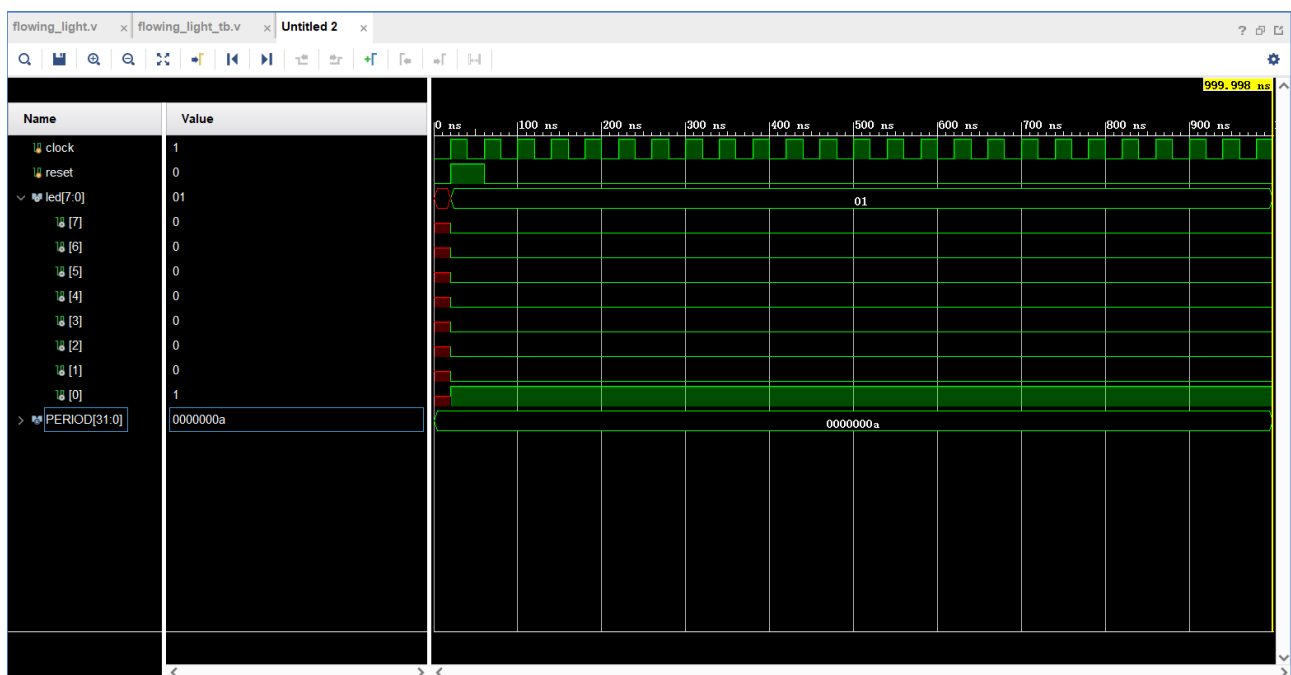


图 1: 基础实验部分仿真全波形

## 5 仿真结果

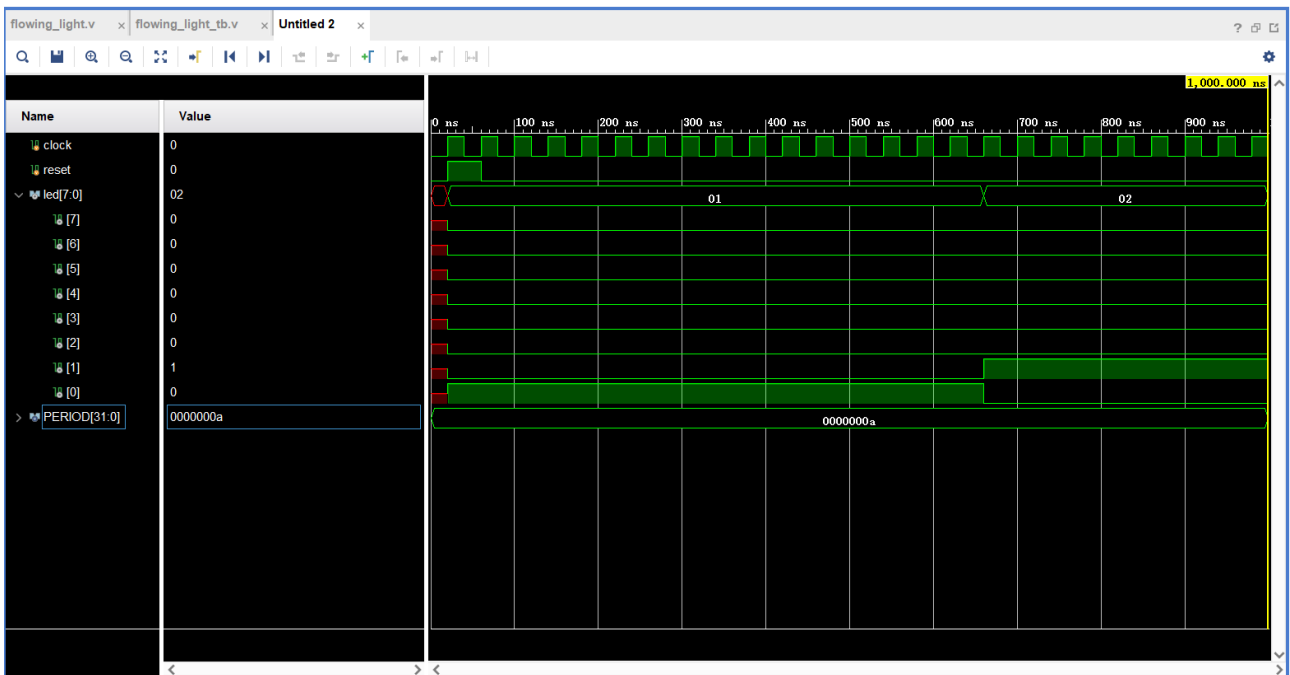


图 2: 拓展部分代码修改后全波形

可以看到基础部分与实验文档中波形相同，拓展部分成功观察到 LED[1] 信号变为高电平，说明代码正确，功能正常，实验成功。

## 6 鸣谢

感谢实验过程中，指导老师给予的帮助，以及我校远程桌面的优质资源！