

计算机系统结构试验 Lab01

518030910022 杨涵章

目录

- 1 实验概述: 2
 - 1.1 实验名称: 2
 - 1.2 实验目的: 2
 - 1.3 实验步骤: 2
- 2 实验描述: 3
 - 2.1 原理描述 3
 - 2.2 模块代码: 3
- 3 模拟仿真: 4
 - 3.1 仿真代码: 4
 - 3.2 仿真结果 5
- 4 总结与反思: 6

1 实验概述：

1.1 实验名称：

FPGA 基础实验：LED Flow Water Light

1.2 实验目的：

- 1) 熟悉 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

1.3 实验步骤：

- 1) 按照基本流程创建文件即可
- 2) 输入给出的示例代码和激励代码
- 3) 程序运行，调整视图获得实验波形截图
- 4) 对程序进行更改，以获得试验周期内两盏 LED 灯亮的截图

2 实验描述：

2.1 原理描述

本实验是计算机系统结构实验课程的第一个实验,目的是让同学慢慢熟悉模拟仿真的集成开发环境和掌握基本的 Verilog 语言进行编程。实验一设置时钟周期, 在时钟的上升沿进行技术操作, 在时钟的下降沿对计数寄存器进行判断处理, 计数值足够边进行进位, 改变 led 寄存器的值, 完成 led 值的变化。

实验要求实现 led 交替变亮。由于时钟周期较短, 在短时间内想要 led 值发生进位需要改变计数寄存器的位数, 使得 led 的交替能够提前发生。因此我改变了计数器的值使之更快达到左移条件。

2.2 模块代码：

代码基本与实验手册给出的代码一样, 程序主体为两个 always 结构, 在时钟上升沿时进行操作。第一部分为计数器增长和 reset 设置, 第二部分为计数器增长移位判断。但由于本仿真运行周期不够, 我选择了改变计数器的位数使得能够快速到达左移条件, 改动见图。下为模块代码。

```
11 module flowing_light(  
12     input clock,  
13     input reset,  
14     output [7:0] led  
15 );  
16  
17     reg [23:0] cnt_reg;           cnt_reg[3:0] cnt_reg;  
18     reg [7:0] light_reg;  
19     always @ (posedge clock)  
20     begin  
21         if(reset)  
22             cnt_reg<=0;  
23         else  
24             cnt_reg<=cnt_reg+1;  
25     end  
26     always @ (posedge clock)  
27     begin  
28         if(reset)  
29             light_reg<=8'h01;  
30         else if (cnt_reg==24'hffffff)           cnt_reg==4'hf  
31         begin  
32             if(light_reg==8'h80)  
33                 light_reg<=8'h01;  
34             else  
35                 light_reg<=light_reg<<1;  
36         end  
37     end  
38     assign led =light_reg;  
39 endmodule
```

3 模拟仿真：

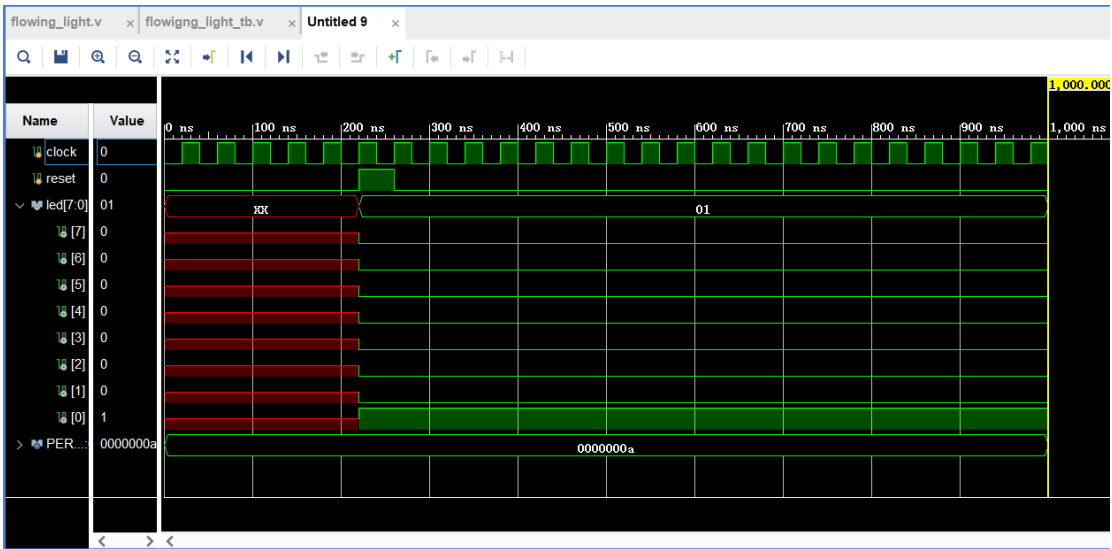
3.1 仿真代码：

```
1 module flowigng_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     #200;  
21     #(PERIOD*2) reset = 1'b1;  
22     #(PERIOD*4) reset = 1'b0;  
23  
24     //$580; reset = 1'b1;  
25 end  
26 endmodule
```

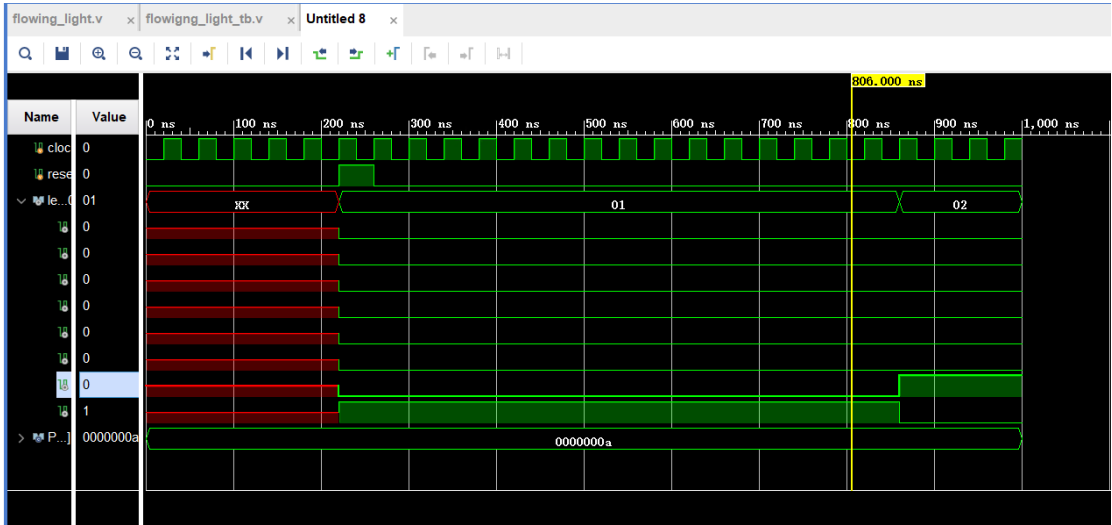
仿真代码同样与实验手册给出代码基本一致，但为了得到和实验手册相同的波形，我添加了#200 的延时，以得到相同的实验波形。

3.2 仿真结果

经过调试运行，得到以下两张波形运行截图，图一为未做更改，按照实验手册给出的代码运行所得到的结果，可以看到在 reset 之后，LED[0]被点亮。图二为更改计数器位数之后的运行波形，可以看到在运行周期内，LED[0]和 LED[1]相继被点亮，实现了实验手册的要求。



图表 1.基本运行波形



图表 2 两盏 LED 交替波形

4 总结与反思：

第一次实验主要是为了让我们了解仿真软件的基本使用, 以及 Verilog 语言的基本语法, 按照实验手册按部就班的操作即可, 没有什么难度。通过这次实验, 我对于使用软件进行仿真的基本流程有了一定的认知, 也为后续的仿真实验打下了一定的基础。

在实验结束后我便自主进行了 Verilog 语言的学习, 扩展了语法知识。本实验要求创新的一点便是使得 LED 能够交替变亮, 实验手册中给出了具体实施方法, 按照方法对应更改即可。实验结果较好, 对程序进行适当更改之后能够得到对应的波形, 自主更改代码的部分也能效果较好的波形。

总体而言算是较为完整的实验, 但遗憾未能进行上板验证。在实验过后仍需进一步学习 Verilog 语法, 以便后续实验能够顺利展开。