

计算机系统结构实验 lab01

Flowing light

徐阳 521021910363

2023 年 3 月 15 日

目录

目录.....	2
1. 摘要.....	2
2. 实验目的.....	2
3. 原理分析.....	2
4. 功能实现.....	2
5. 结果验证.....	3
5.1 仿真模拟	3
5.2 管脚约束与上板验证	5
6. 总结与反思	7

1. 摘要

本实验中，我们实现了 FPGA 基础实验中 LED 流水灯的设计，初步了解了 Vivado 软件的使用和 Verilog 语言的基础语法，学习了如何通过 Vivado 软件平台仿真，并在 FPGA 器件上进行了验证。

2. 实验目的

- (1) 熟悉 Xilinx 逻辑设计工具 Vivado 软件开发环境；
- (2) 了解硬件描述语言 Verilog，进行简单的逻辑设计；
- (3) 理解流水灯实现原理，实现 flowing light，使用软件仿真验证正确性；
- (4) 学习使用 I/O Planning 添加管脚约束，上板验证正确性；
- (5) 熟悉系统硬件开发的基本实验流程。

3. 原理分析

实验要求实现 LED 流水灯器件，该器件在每个周期的时钟上升沿时计数器加 1，当计数器达到最大值时，通过位移操作使 8 个 LED 灯依次轮流亮灭；该器件还支持接收 reset 信号对 LED 灯进行初始化与复位。

Vivado 软件中的工程文件由源文件 (*.v)、激励文件 (*.v)、管脚约束文件 (*.xdc) 构成。

4. 功能实现

flowing_light.v 的关键代码如下，定义了两个寄存器数据类型，cnt_reg 用来计数，light_reg 用来点灯。

从仿真 0 时刻开始循环执行 always 行为语句模块，posedge 指在 clock 上升沿的时候执行该模块，always 模块并行执行。

在第一个 always 模块中，当 reset 信号为 1 时，清零重置 cnt_reg 计数器，否则正常更

新 cnt_reg。

在第二个 always 模块中，当 reset 信号为 1 时，将 LED 灯状态重置，计数器达到特定值后，重置计数器，对 light_reg 进行左移操作，实现灯的轮流亮灭。

```
29      reg [23 : 0] cnt_reg;
30      reg [7 : 0] light_reg;
31      always @ (posedge clock)
32      begin
33          if (reset)
34              cnt_reg <= 0;
35          else
36              cnt_reg <= cnt_reg + 1;
37      end
38      always @ (posedge clock)
39      begin
40          if (reset)
41              light_reg <= 8'h01;
42          else if (cnt_reg == 2'h3)
43              begin
44                  cnt_reg <= 0;
45                  if (light_reg == 8'h80)
46                      light_reg <= 8'h01;
47                  else
48                      light_reg <= (light_reg << 1);
49              end
50      end
51      assign led = light_reg;
52  endmodule
```

在仿真中，为使实验结果明显，我们将寄存器最大值减小为 2'h3，而在上板验证中为 24'hffffff。

实现 flowing_light.v 后，添加激励文件 flowing_light_tb.v 用于仿真测试，添加管脚约束文件 lab01_xdc.xdc 用于上板验证

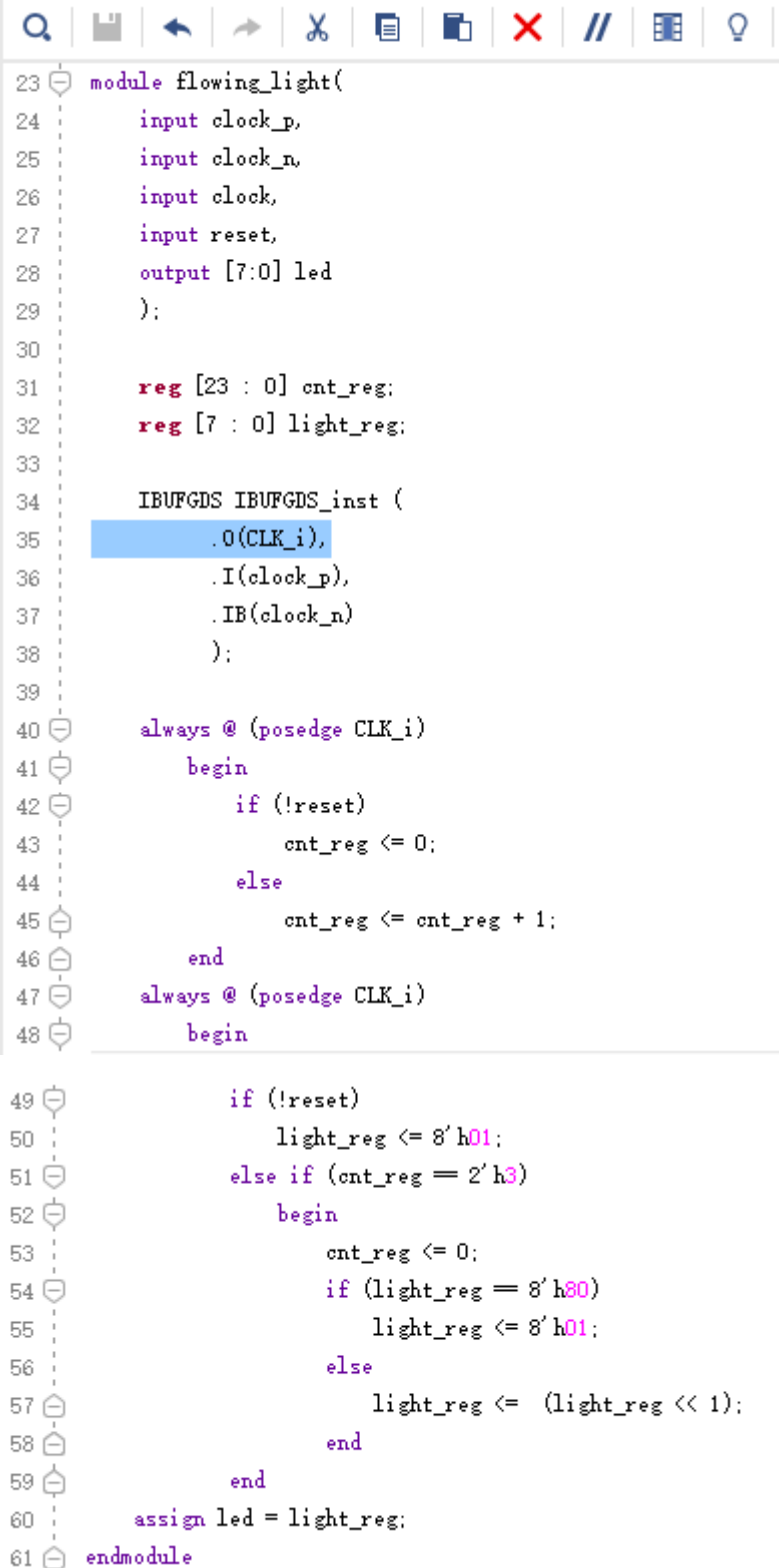
5. 结果验证

5.1 仿真模拟

我们按实验指导书编写激励文件如下，可以看出时钟周期是 20 个时间单位。

5.2 管脚约束与上板验证

上板验证时，按实验指导书，代码需要做细节修改如下，添加了拆分时钟



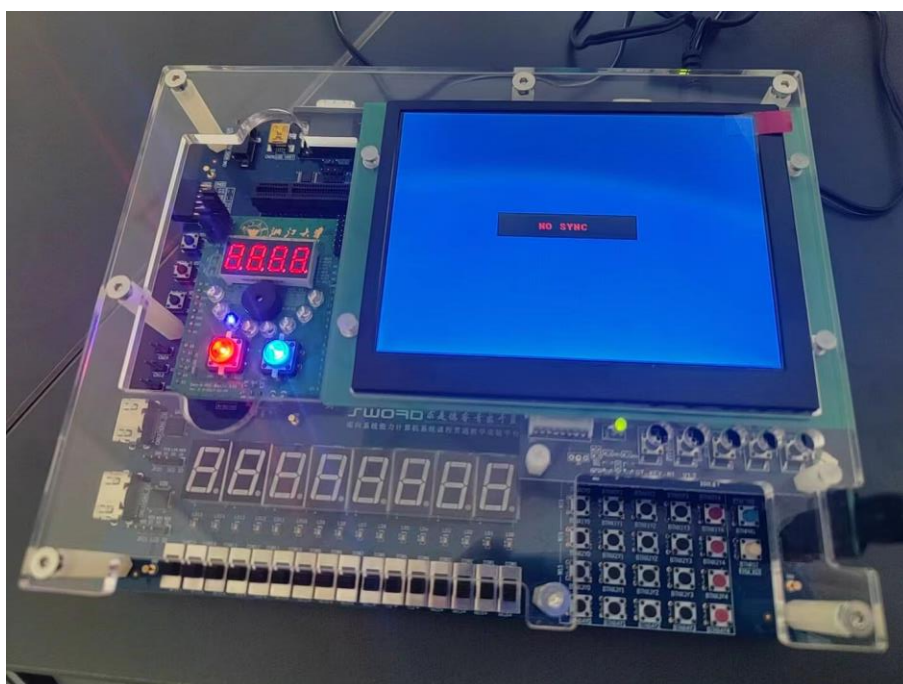
```
23 module flowing_light(  
24     input clock_p,  
25     input clock_n,  
26     input clock,  
27     input reset,  
28     output [7:0] led  
29 );  
30  
31 reg [23 : 0] cnt_reg;  
32 reg [7 : 0] light_reg;  
33  
34 IBUFGDS IBUFGDS_inst (  
35     .O(CLK_i),  
36     .I(clock_p),  
37     .IB(clock_n)  
38 );  
39  
40 always @ (posedge CLK_i)  
41 begin  
42     if (!reset)  
43         cnt_reg <= 0;  
44     else  
45         cnt_reg <= cnt_reg + 1;  
46 end  
47 always @ (posedge CLK_i)  
48 begin  
49     if (!reset)  
50         light_reg <= 8'h01;  
51     else if (cnt_reg == 2'h3)  
52     begin  
53         cnt_reg <= 0;  
54         if (light_reg == 8'h80)  
55             light_reg <= 8'h01;  
56         else  
57             light_reg <= (light_reg << 1);  
58         end  
59     end  
60     assign led = light_reg;  
61 endmodule
```

接着添加管脚约束文件，可以使用 Vivado 的 I/O Planning 功能，也可手动添加管脚约束 xdc 文件，本次实验中，手动添加管脚约束文件如下，

```
Project Summary x flowing_light.v x flowing_light_tb.v x lab01_xdc.xdc x
E:/archlabs/lab01/lab01.srcs/constrs_1/new/lab01_xdc.xdc

1 set_property PACKAGE_PIN W23 [get_ports {led[7]}]
2 set_property PACKAGE_PIN AB26 [get_ports {led[6]}]
3 set_property PACKAGE_PIN Y25 [get_ports {led[5]}]
4 set_property PACKAGE_PIN AA23 [get_ports {led[4]}]
5 set_property PACKAGE_PIN Y23 [get_ports {led[3]}]
6 set_property PACKAGE_PIN Y22 [get_ports {led[2]}]
7 set_property PACKAGE_PIN AE21 [get_ports {led[1]}]
8 set_property PACKAGE_PIN AF24 [get_ports {led[0]}]
9 set_property PACKAGE_PIN AC18 [get_ports clock_p]
10 set_property PACKAGE_PIN W13 [get_ports reset]
11 set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
12 set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
13 set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
14 set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
15 set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
16 set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
17 set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
18 set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
19 set_property IOSTANDARD LVDS [get_ports clock_p]
20 set_property IOSTANDARD LVDS [get_ports clock_n]
21 set_property IOSTANDARD LVCMOS18 [get_ports reset]
```

下载代码至 FPGA 烧写，成功实现 LED 流水灯功能。



6. 总结与反思

本次实验是我接触 FPGA、Vivado 的第一节课，作为一个初学者，我在这次实验中对 Xilinx 逻辑设计工具 Vivado 软件开发环境和 Verilog 语言有了一个初步的认识，基本掌握了 Verilog 语言的基础语法。同时我还学习了在 Vivado 平台仿真验证并上板实现的过程。感谢老师与助教提供的教学、指导答疑以及实验资源，感谢各位同学在实验过程中提供的帮助。