

SHANGHAI JIAOTONG UNIVERSITY



计算机系统结构实验报告–Lab1

姓名：刘一凡

学号：518021910609

完成时间：2020/4/16

目录

[1. 概述 3](#_Toc40977284)

[1.1 实验名称 3](#_Toc40977285)

[1.2 实验目的 3](#_Toc40977286)

[2. 实验描述 3](#_Toc40977287)

[2.1 熟悉软件与新建项目 3](#_Toc40977288)

[2.1.1整体界面 3](#_Toc40977289)

[2.1.2 流水灯项目代码 4](#_Toc40977290)

[2.2 功能仿真 5](#_Toc40977291)

[2.2.1 步骤简述 5](#_Toc40977292)

[2.2.2 激励文件代码 5](#_Toc40977293)

[2.2.3 仿真波形 6](#_Toc40977294)

[2.2.4 注意事项 7](#_Toc40977295)

[2.2.5 实验结论 8](#_Toc40977296)

[2.3 工程实现 8](#_Toc40977297)

[2.3.1 步骤简述 8](#_Toc40977298)

[2.3.2 修改后的流水灯代码 8](#_Toc40977299)

[2.3.3管脚约束文件代码 9](#_Toc40977300)

[3. 实验心得 10](#_Toc40977301)

# 1. 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1/0 Planing 添加管脚约束

5) 生成Bitstream 文件

6) 上板验证

# 2. 实验描述

## 2.1 熟悉软件与新建项目

### 2.1.1整体界面

如图 1 是Vivado2018.3 的整体界面，大致分为四个区：

1) 左侧区Flow Navigator 包含整个开发流程，像Project Settings、

Run Simulation、Run Synthesized 和Generate Bitstream 等；

2) 中间区通常显示当前工程包含的文件树结构，提供工程文件的管理；

3) 右侧区会显示工程信息、打开的编辑文件等；

4) 下部区显示各种信息状态

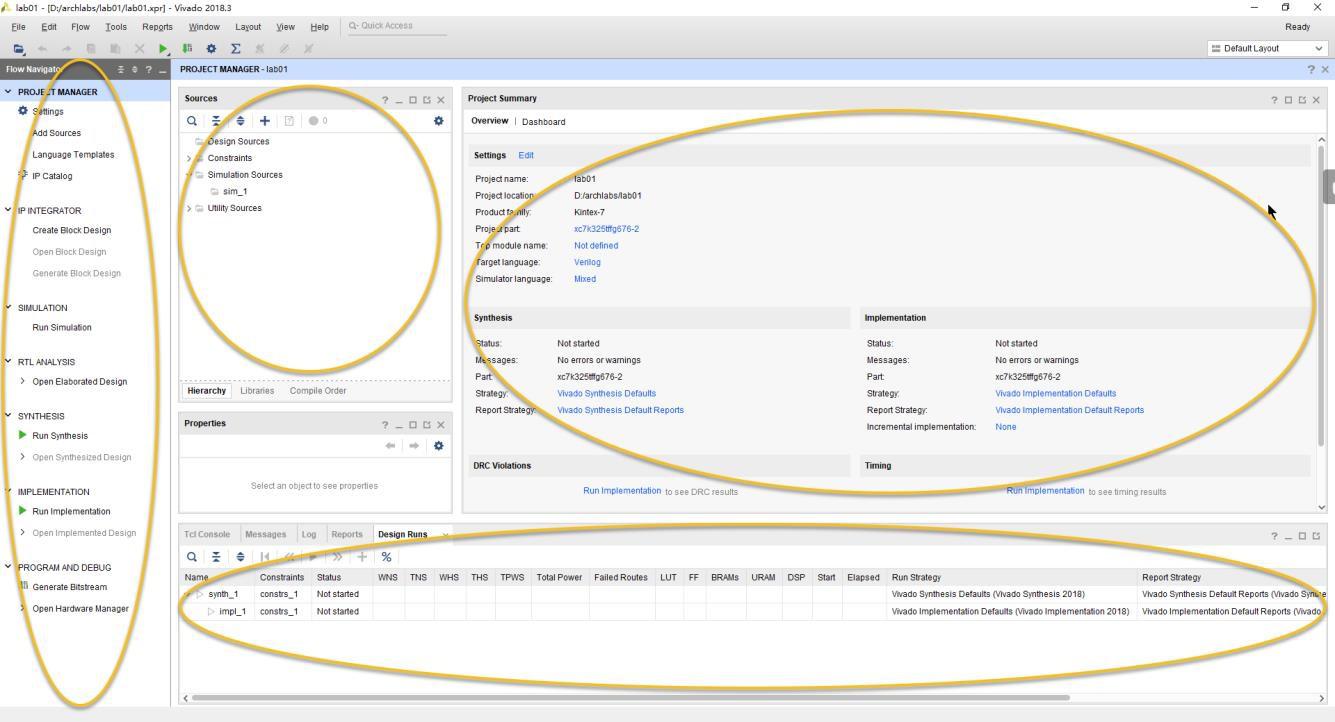


图1：Vivado的整体界面

### 2.1.2 流水灯项目代码

1. module flowing\_light(
2. input clock,
3. input reset,
4. output [7:0] led
5. );
6. reg [23:0] cnt\_reg;
7. reg [7:0] light\_reg;
9. always @ (posedge clock)
10. begin
11. **if** (reset)
12. cnt\_reg<=0;
13. **else**
14. cnt\_reg<=(cnt\_reg+1);
15. end
16. always @ (posedge clock)
17. begin
18. **if** (reset)
19. light\_reg<= 8'h01;
20. **else** **if** (cnt\_reg==24'hffffff)
21. begin
22. cnt\_reg<=0;
23. **if**(light\_reg==8'h80)
24. light\_reg<=8'h01;
25. **else**
26. light\_reg<=light\_reg<<1;
27. end
28. end
29. assign led=light\_reg;
30. endmodule

## 2.2 功能仿真

### 2.2.1 步骤简述

功能仿真是实验的模拟验证部分。本次功能仿真利用激励文件来实现流水灯项目的模拟波形，从而验证自己的代码是否正确。（注：工程文件中的流水灯代码是“工程实现”中修改后的代码，而非仿真实验中的代码，故如果需要验证仿真实验的正确性，请将上面的代码复制到相应文件中。）

### 2.2.2 激励文件代码

1. module flowing\_light\_tb(
3. );
4. reg clock;
5. reg reset;
6. wire [7:0] led;
8. flowing\_light u0(
9. .clock(clock),
10. .reset(reset),
11. .led(led));
13. parameter PERIOD=20;
15. always #(PERIOD\*2) clock=!clock;
17. initial begin
18. clock=1'b0;
19. reset=1'b0;
20. # (PERIOD\*10) reset=1'b1;
21. # (PERIOD\*5) reset=1'b0;
23. //#580; reset=1'b1
25. end
26. endmodule

### 2.2.3 仿真波形

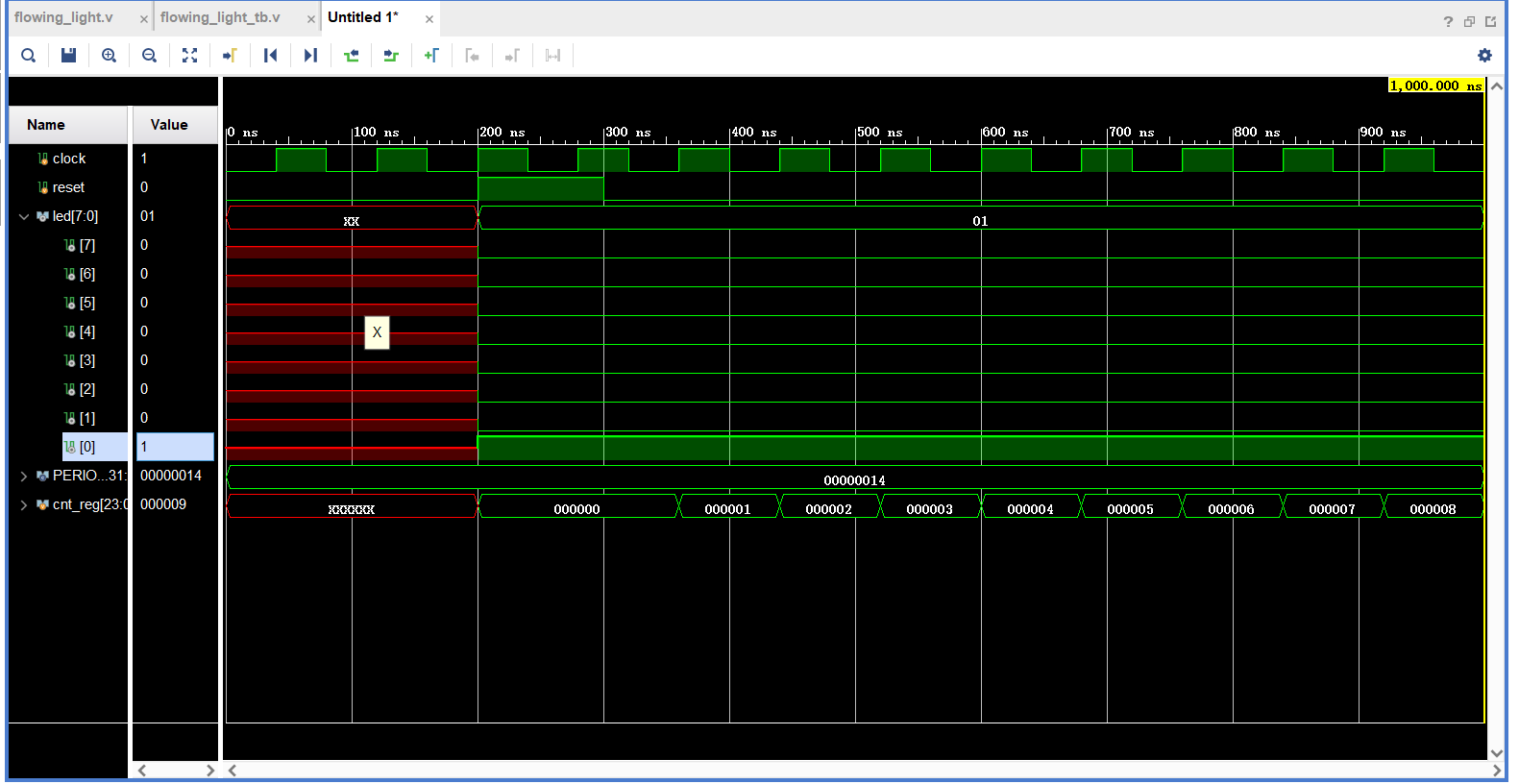


图2：流水灯的仿真波形

由于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的位数或者改变计数器计数值等参数，以便较快速达到左移条件。我修改了流水灯的实现代码如图3所示，从而得到了流水灯在较短时间内依次点亮的波形。

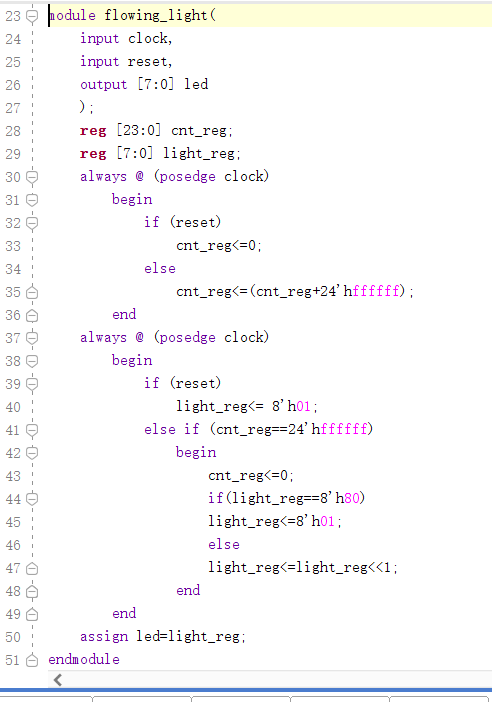


图3：流水灯依次点亮的修改后代码

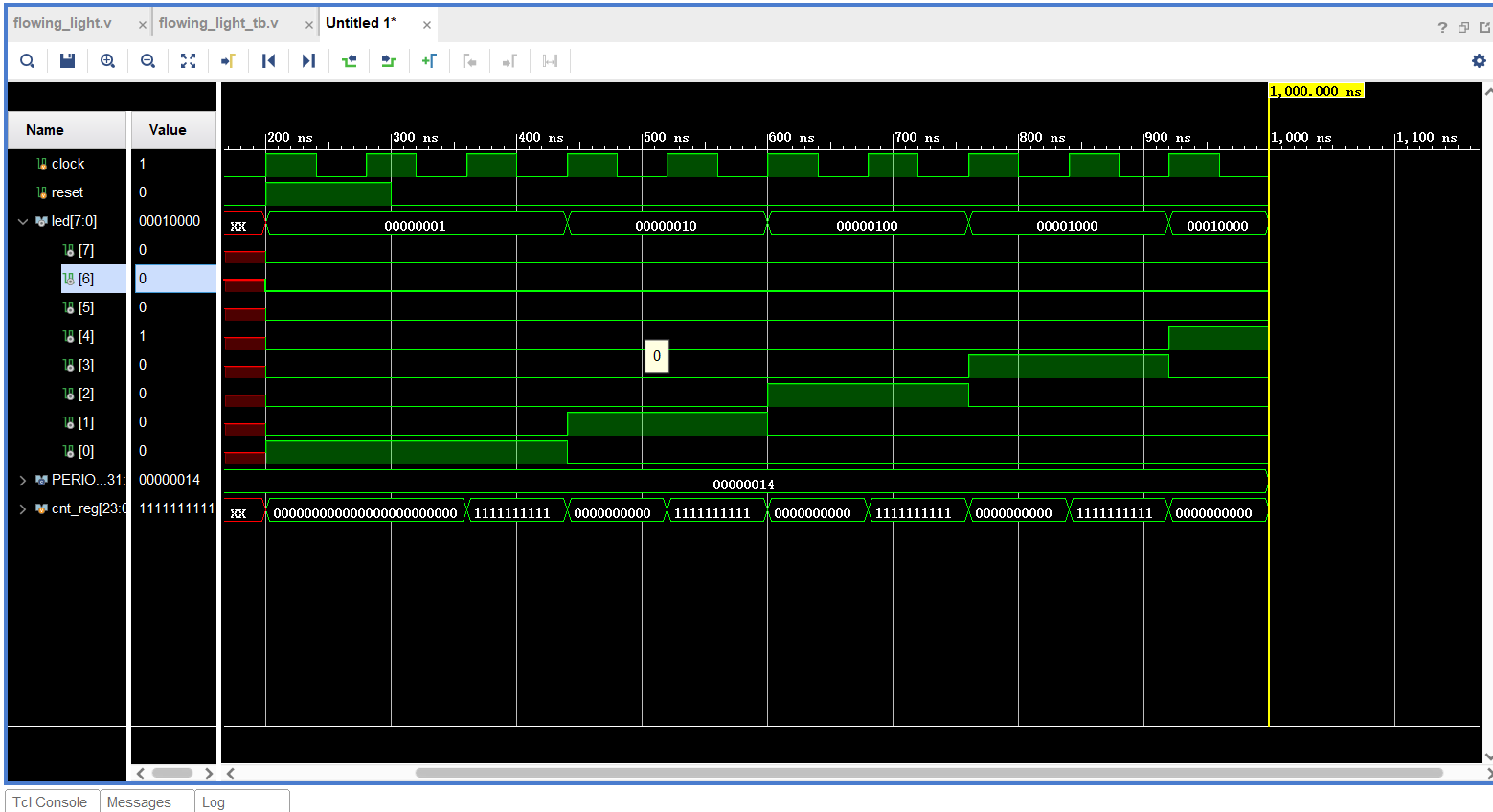


图4：流水灯短时间内依次点亮的仿真波形

### 2.2.4 注意事项

a) 通过键盘Ctrl+“-”和Ctrl+“+”可以对波形图进行缩放；

b) 对长信号而言，有时十六进制看起来更顺。选择该信号并点击右键，通过Radix菜单可以改变信号不同进制的显示方式；

c) Scopes 窗口里可选中需要查看信号的模块，选中感兴趣的信号，点击右键并选Add To Wave Window，便可把改信号增加到仿真波形图中：

d) 增加信号后需要使用Run 菜单下的Restart，重新开始或用工具条按钮实现；

e) 使用Run for…仿真指定时间长的波形；

f) 可通过选择工具栏中的工具条按钮来进行波形的仿真时间控制。工具条上分别是复位波形（即清空现有波形）、运行仿真、运行特定时长的仿真、仿真时长设置、仿真时长单位、单步运行、暂停。

### 2.2.5 实验结论

仿真波形与理论应得波形及代码完全对应，能够实现流水灯的单个点亮以及依次点亮，实验成功。

## 2.3 工程实现

### 2.3.1 步骤简述

工程实现是模拟实验的实际验证阶段，用实际的硬件完成实验项目的下载以及验证。由于线上上课无法进行实际操作，在这里仅仅做了部分代码的实现，没有进行实际的验证。

### 2.3.2 修改后的流水灯代码

1. module flowing\_light(
2. input clock\_p,
3. input clock\_n,
4. input reset,
5. output [7:0] led
6. );
7. reg [23:0] cnt\_reg;
8. reg [7:0] light\_reg;
10. IBUFGDS IBUFGDS\_inst(
11. .O(CLK\_i),
12. .I(clock\_p),
13. .IB(clock\_n)
14. );
15. always @ (posedge CLK\_i)
16. begin
17. **if** (!reset)
18. cnt\_reg<=0;
19. **else**
20. cnt\_reg<=(cnt\_reg+1);
21. end
22. always @ (posedge CLK\_i)
23. begin
24. **if** (!reset)
25. light\_reg<= 8'h01;
26. **else** **if** (cnt\_reg==24'hffffff)
27. begin
28. cnt\_reg<=0;
29. **if**(light\_reg==8'h80)
30. light\_reg<=8'h01;
31. **else**
32. light\_reg<=light\_reg<<1;
33. end
34. end
35. assign led=light\_reg;
36. endmodule

### 2.3.3管脚约束文件代码

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\_n]
20. set\_property IOSTANDARD LVCMOS18 [get\_ports reset]

# 3. 实验心得

本次实验是第一个实验，主要目的是让我们熟悉Vivado的使用以及熟悉开发环境。这次实验是验证实验，按部就班地照着实验书操作就可以获得相关的结果。在实验过程中，我逐步了解了本次实验使用的软件的界面和操作，知道了如何建立激励文件以及运行仿真程序，还了解了基本的Verilog语言。在实验书详尽的指导下，我从零开始，逐步学会了Vivado的基本使用流程，写出了流水灯的相关代码。

在绘制仿真波形的过程中，我通过调节周期等相关参数，得到了不同的波形，这让我更深入地了解了代码的基本逻辑，为以后的实验打下了基础。

这次实验中，我也遇到了一些问题，例如不清楚endmodule需要放在代码的最后一行而导致编译报错，感谢老师和同学的细心指导，让我较快地找到了问题来源并顺利完成了实验。由于课程是在线上课，没有能够上板验证，如果能够及时返校上课希望能补上上板验证的环节，完善整个实验。