

# 计算机系统结构实验 Lab1 实验报告

---

- 姓名：夏鸿驰
- 学号：520021910965
- 日期：2022年4月26日

## 目录

---

1. 实验概述
  - 1.1 实验名称
  - 1.2 实验目的
2. 实验步骤
  - 2.1 新建工程
  - 2.2 熟悉Vivado整体界面
  - 2.3 添加文件
  - 2.4 功能仿真
  - 2.5 工程实现
3. 实验心得
  - 3.1 实验重难点
  - 3.2 实验感想
4. 参考资料

## 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. 掌握 Xilinx 逻辑设计工具 Vivado 的基本操作
2. 掌握使用 Verilog HDL 进行简单的逻辑设计
3. 掌握功能仿真
4. 使用 I/O Planing 添加管脚约束
5. 生成 Bitstream 文件
6. 上板验证

## 2. 实验步骤

---

### 2.1 新建工程

2.1.1 启动桌面 Vivado 2018.3 开发工具

2.1.2 点击 Create Project

2.1.3 弹出 New Project 向导，由此建立一个新工程，点击 Next

2.1.4 输入工程名称 lab01，选择工程存放位置，确认勾选 Create project subdirectory 后，点击 Next

2.1.5 选择 RTL Project 工程类型，勾选 Do not specify sources at this time 在创建工程时不决定 sources 文件，点击 Next

2.1.6 选择 SWORD4.0 的 FPGA 参数:

Family 选 Kintex-7, Package 选 ffg676, Speed grade 选-2; 接着具体型号中选 xc7k325tffg676-2, 点击 Next

2.1.7 弹出新工程信息综述, 点击 Finish 结束工程创建

## 2.2 熟悉Vivado整体界面

2.2.1 观察左侧区Flow Navigator:

包含整个开发流程, 如Project Settings、Run Simulation、Run Synthesized 和 Generate Bitstream 等;

2.2.2 观察中间区:

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;

2.2.3 观察右侧区:

会显示工程信息、打开的编辑文件等;

2.2.4 观察下部区:

显示各种信息状态;

## 2.3 添加文件

2.3.1 点击左侧区 Flow Navigator 下的 Project Manager->Add Sources或中间区 Sources 的“+”号, 打开 Add Sources 对话框

2.3.2 选择第二项 Add or Create Design Sources, 从而添加或新建 Verilog HDL源文件, 点击Next

2.3.3 新建代码文件, 选择 Create File 项; (若已有代码模块文件或 IP 核文件, 可选 Add Files 以添加所需文件。)

2.3.4 弹框 Create Source File 中, 输入文件名, 如flowing\_light, 点击 OK

2.3.5 点击 Finish

2.3.6 在弹出的 Define Module 中的 I/O Port Definition, 输入设计模块所需的端口, 并设置端口方向: 若端口为总线型, 勾选 Bus 选项, 并由 MSB 和LSB 确定总线宽度; 完成后点击 OK

2.3.7 打开Sources 中的 Design Sources, 观察到新建的 flowing\_light 文件显示在其中

2.3.8 向刚刚新建的文件 (flowing\_light) 中写入添加设计代码:

```
1
2 `timescale 1ns / 1ps
3
4
5 module flowing_light(
6     input clock,
7     input reset,
8     output [7:0] led
9 );
10     reg [23 : 0] cnt_reg;
11     reg [7 : 0] light_reg;
12
13
14     always @ (posedge clock)
15         begin
16             if (reset)
17                 cnt_reg <=0;
```

```

18         else
19             cnt_reg <= cnt_reg +1;
20         end
21     always @ (posedge clock)
22     begin
23         if (reset)
24             light_reg<=8'h01;
25         else if (cnt_reg == 24'hffffff)
26             begin
27                 if (light_reg == 8'h80)
28                     light_reg <= 8'h01;
29                 else
30                     light_reg <= light_reg << 1;
31             end
32         end
33     assign led = light_reg;
34 endmodule
35

```

## 2.4 功能仿真

2.4.1 创建激励测试文件。在中间区 Source 中右击选择 Add Source，或于左侧区 PROJEU MAHAGER 下选择 Add Source

2.4.2 在 Add Source 中选择第三项 Add or Create Simulation Source，点击 Next

2.4.3 选择 Create File 创建一个仿真激励文件，并输入激励文件名称，可以是 flowing\_light\_tb，点击 OK

2.4.4 完成之后点击 Finish，创建激励文件是不需要对外端口，再点击 OK

2.4.5 在弹出的对话框中点击 Yes

2.4.6 在 Source 区 Simulation Sources 下，打开仿真测试文件 flowing\_light\_tb。

2.4.7 在其中对要进行仿真的模块作实例化并编写激励代码如下：

```

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

```

```

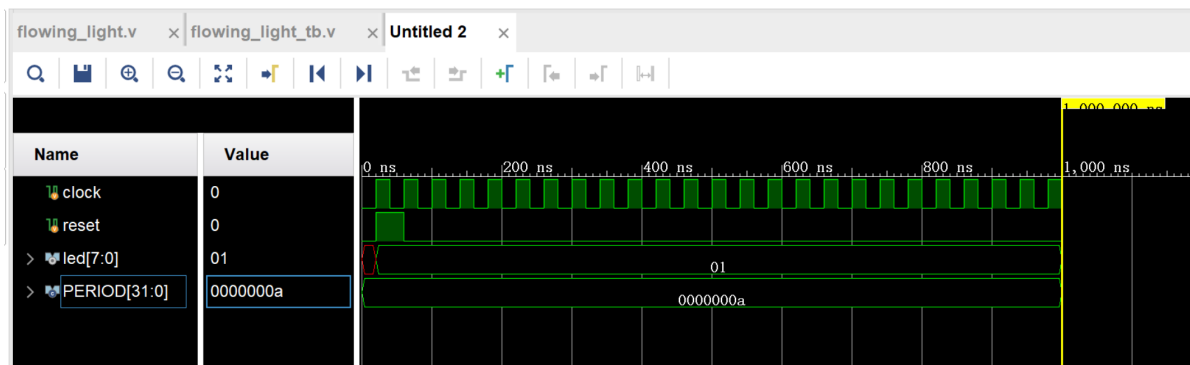
23         #(PERIOD*2) reset = 1'b1;
24         #(PERIOD*4) reset = 1'b0;
25
26         //#580; reset = 1'b1;
27     end
28
29 endmodule
30

```

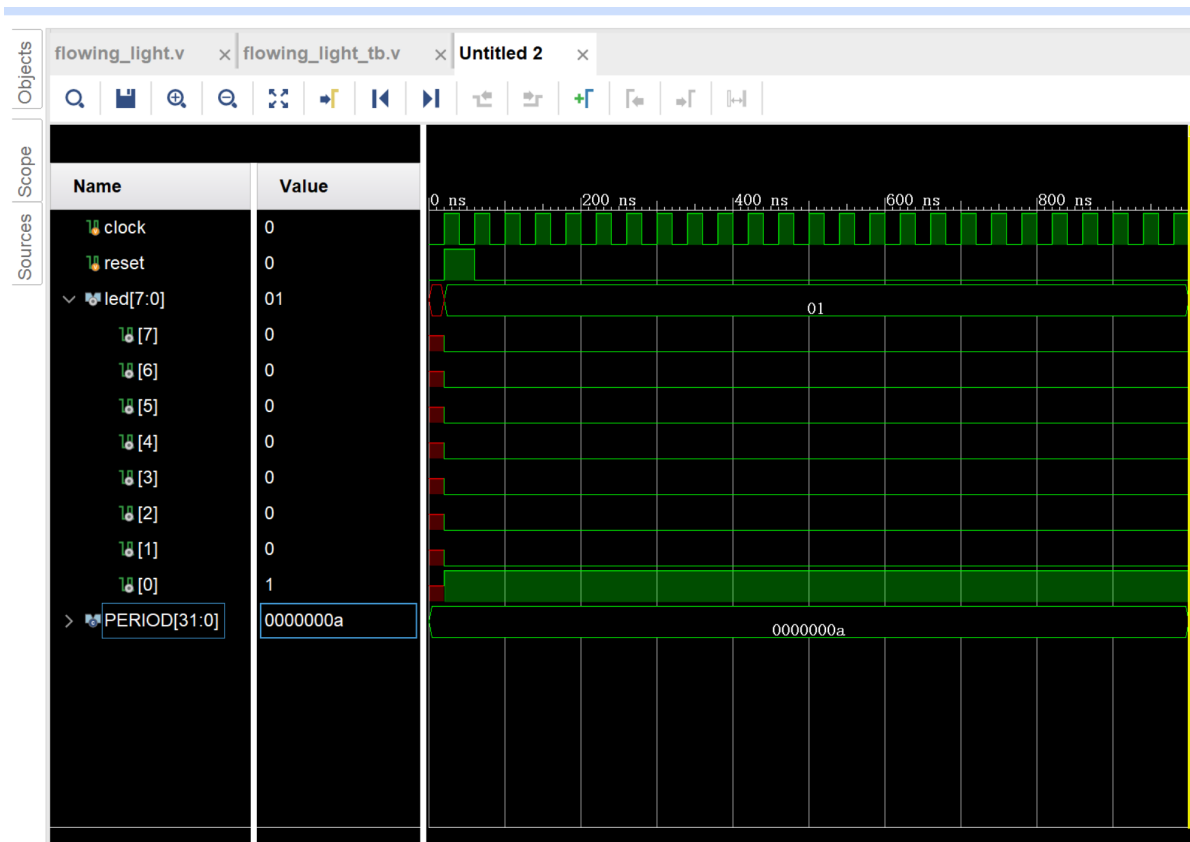
2.4.8 在左侧 Flow Navigator 中点击 Simulation 下的 Run Simulation 选项，并选择 Run Behavioral Simulation。

以下Figure1 和 Figure2 为基于以上模块代码和激励测试文件运行仿真后所得到的波形：

**Figure1:**



**Figure2:**



2.4.9 通过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，可把该信号增加到仿真波形图中。

## 2.5 工程实现

2.5.1 由于实验板板载了 200MHz 时钟振荡器，属高频时钟，做下载验证时则需用到差分时钟以更好适应工程上的需要。原 flowing\_light 代码模块需做时钟方面的修改。

flowing\_light.v 修改后如下：

```
1
2 `timescale 1ns / 1ps
3
4 module flowing_light(
5     input clock_p,
6     input clock_n,
7     input reset,
8     output [7:0] led
9 );
10 reg [23 : 0] cnt_reg;
11 reg [7 : 0] light_reg;
12
13 IBUFGDS IBUFGDS_inst (
14     .O(CLK_i),
15     .I(clock_p),
16     .IB(clock_n)
17 );
18
19 always @ (posedge CLK_i)
20     begin
21         if (!reset)
22             cnt_reg <= 0;
23         else
24             cnt_reg <= cnt_reg + 1;
25     end
26 always @ (posedge CLK_i)
27     begin
28         if (!reset)
29             light_reg <= 8'h01;
30         else if (cnt_reg == 24'hffffff)
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
40
41
```

### 2.5.2 添加管脚约束文件

有下面两种方法可添加约束文件：

- 一是利用 Vivado 的 IO planning 功能；
- 二是直接新建类型为 xdc 的约束文件，手动输入约束命令。

## 3. 实验心得

### 3.1 实验重难点

Vivado开发环境和Verilog语言对大多数人来说都很陌生，所以要完成好实验就必须尽快掌握Verilog的开发环境和语言，从而为之后的实验打下基础。

### 3.2 实验感想

这是我第一次使用Verilog这种逻辑设计和硬件描述语言。所以在最初的实验中遇到了一些困难。在此次验证实验中，我编写了Verilog代码并调试了仿真波形。这让我对Verilog和Vivado产生了浓厚的兴趣。此外还要感谢老师们在微信群中及时的答疑，帮助我解决了很多棘手的问题。

## 4. 参考资料

---

[1] 2022计算机系统结构实验指导书lab1