

# Lab01 FPGA 基础实验：LED Flow Water Light

w1049

## 目录

1	实验目的	1
2	原理分析	1
2.1	Vivado 源文件分类 . . . . .	1
2.2	Vivado 开发流程 . . . . .	2
2.3	流水灯原理 . . . . .	2
3	功能实现	2
4	功能仿真	2
4.1	仿真激励文件 . . . . .	2
4.2	仿真结果 . . . . .	3
5	上板验证	3
5.1	修改代码 . . . . .	3
5.2	管脚约束 . . . . .	3
5.3	连接并下载 . . . . .	4
6	总结与反思	4

## 1 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 原理分析

### 2.1 Vivado 源文件分类

实验中使用的 Vivado 源文件（Sources）分为三种类型：

- design sources: Verilog 源文件
- simulation sources: 用于仿真的源文件
- constrains: 上板验证需要的管脚约束文件

## 2.2 Vivado 开发流程

1. 创建工程
2. 编写 Verilog 源文件，实现所需功能
3. 创建激励测试文件，进行功能仿真（Simulation），检查功能是否符合预期
4. 运行综合（Synthesis）、配置管脚约束
5. 运行实现（Implementation）、生成字节流 Bitstream（Generate Bitstream）
6. 连接开发板，下载字节流文件（Program Device）

## 2.3 流水灯原理

使用一个 8 位寄存器表示灯的亮灭情况，每隔一定时间时循环左移一位，即可实现流水灯效果。为了计时，可以再使用一个寄存器作为计数器。

## 3 功能实现

根据实验指导书，可以用以下代码进行功能仿真：

```
module flowing_light (  
    input clock,  
    input reset,  
    output [7:0] led  
);  
    reg [23:0] cnt_reg;  
    reg [7:0] light_reg;  
  
    always @(posedge clock) begin  
        if (reset)  
            cnt_reg <= 0;  
        else  
            cnt_reg <= cnt_reg + 1;  
        end  
  
    always @(posedge clock) begin  
        if (reset)  
            light_reg <= 8'h01;  
        else if (cnt_reg == 24'hffffff) begin  
            if (light_reg == 8'h80)  
                light_reg <= 8'h01;  
            else  
                light_reg <= light_reg << 1;  
            end  
        end  
  
        assign led = light_reg;  
    endmodule
```

其中两个寄存器分别用于存储计数值和存储灯的状态，两个 `always` 块分别用于计数和更新灯的状态；`assign` 使用组合逻辑把 `led` 连接到 `light_reg` 上，使前者的值时刻等于后者。

## 4 功能仿真

### 4.1 仿真激励文件

仿真与开发板上直接运行不同，没有板上的按键、开关、时钟等外部输入，因此需要自己编写激励文件，为寄存器赋初值、提供虚拟时钟等，使其尽量与实际情况接近。

## 4.2 仿真结果

运行仿真后，调整波形图的显示，可以看到波形如图 1。

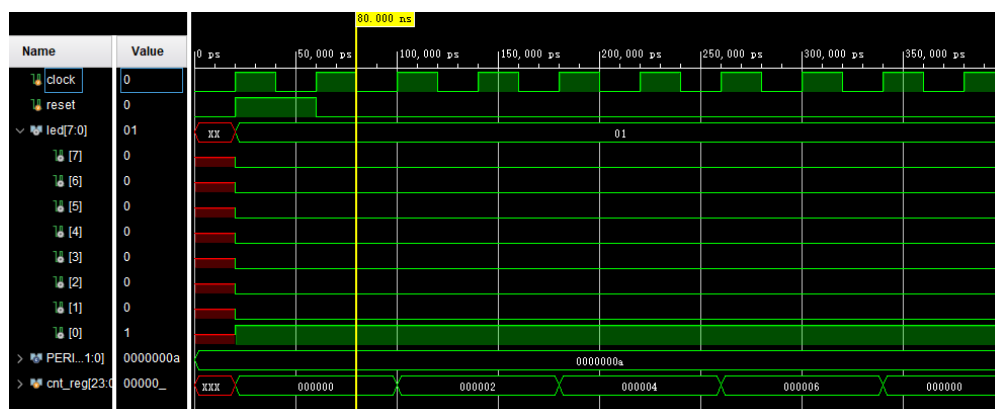


图 1: 首次仿真波形

注意到只有 led[0] 一直为 1，其他灯均为 0，这是因为仿真的时间很短，远远达不到灯切换的时刻。为了在仿真中观察到“流水效果”，可以把切换间隔改小。我在这里把 reg [23:0] cnt\_reg 改为 reg [1:0] cnt\_reg，相应地把 if (cnt\_reg == 24'hffffff) 改为 if (cnt\_reg == 2'h3)，再次运行仿真，可以看到波形如图 2。

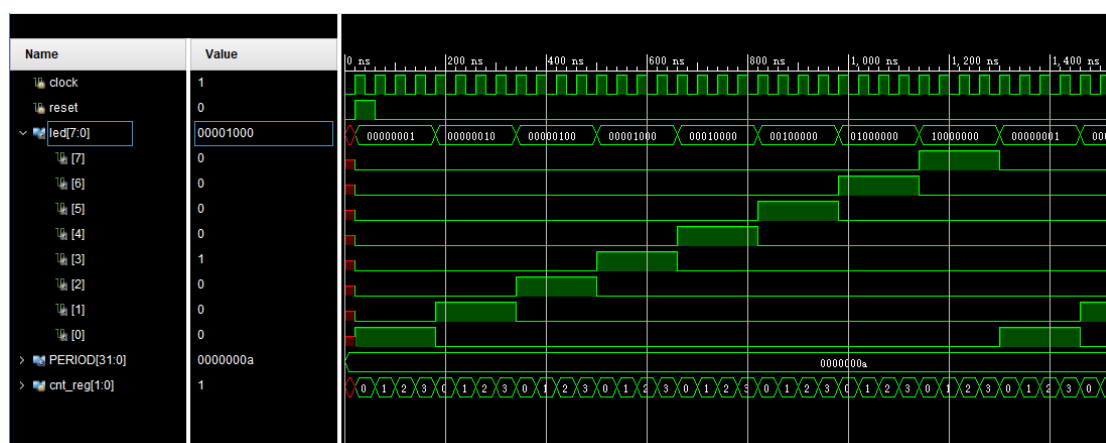


图 2: 流水灯仿真波形

## 5 上板验证

### 5.1 修改代码

在上板前，要按照实验指导书的要求修改功能仿真的代码，以适应实际工程。注意将之前仿真时修改的 cnt\_reg 相关代码改回去。

### 5.2 管脚约束

管脚约束将代码中的 input、output 变量对应到开发板上的实际管脚。可以直接新建约束文件，复制实验指导书中的内容；也可以自己在 I/O Planing 中按手册添加约束。

### 5.3 连接并下载

代码的综合、实现等要花费比较长的时间。连接开发板并下载后，就可以在板子上看到灯的流水效果了（图3）。

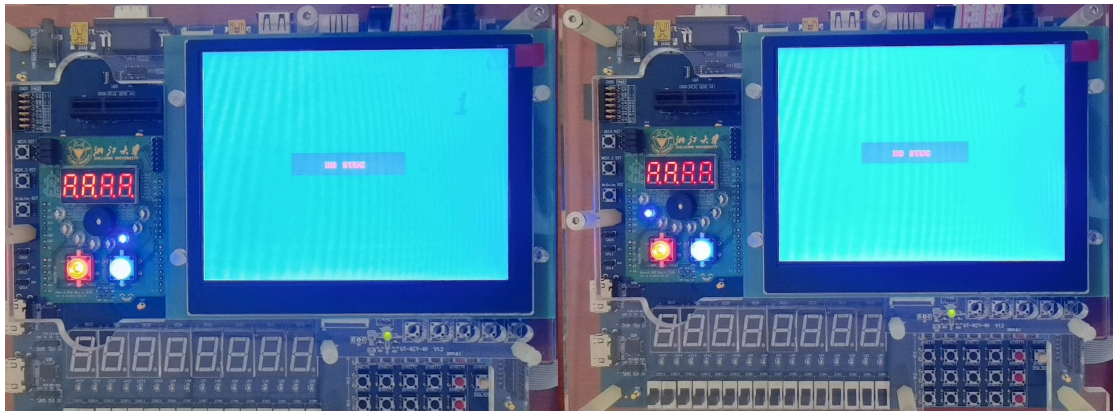


图 3: 流水灯实际效果

## 6 总结与反思

我之前在其他课程中接触过 Verilog 和 Vivado，因此对开发流程和 Verilog 语法都比较熟悉，在实验中也没有遇到什么困难。但我以前没有用过仿真，而是直接下载到板子上查看效果，这是非常低效的；这次实验让我了解了仿真功能的使用和波形图的观察方法。