计算机系统结构实验报告　实验 1

颜培深 518030910094

2020 年 5 月 17 日

摘要

本实验实现了 FPGA 基础实验的 LED Flow Water Light 器件，每个周期计数器加一，当计数器值达到最大值时 LED 灯左移一位点亮。实验通过软件仿真的方式进行结果验证。

目录

1 实验目的

本次实验给出的四个要求：

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. 理解 LED Flow Water Light 的工作原理
4. 使用功能仿真验证实现正确性

2 原理分析

本次实验内容较为简单，仅需要实现对 LED 流水灯的实现，即每间隔一小段时间实现向下一 LED 灯的点亮操作，所以我们利用一个计数器实现对时钟周期数目的记录，当计数器达到最大值后，下一次计数值将会恢复到 0，所以我们选取计数器达到最大值作为切换 LED 灯的标志信号。而 8 位 LED 灯我们利用 8 位二进制数组表示，由于每次只有一个 LED 灯处于点亮状态，所以我们每次切换只需要将这个 8 位二进制编码左移一位即可（注意：这里当二进制编码左移达到 $8'b10000000$ 的时候，此时下一次数值变换应当将其设置为 $8'b00000001$，从而实现 8 位 LED 灯的循环）。

3 功能实现

LED Flow Water Light 的功能是利用 *cnt_reg* 记录时钟周期数目，利用 *light_reg* 记录 LED 显示状态，并且初始化过程将会对计数器清零，所以计数器实现部分代码如下：

```
always @ (posedge clock) begin
    if(reset)
        cnt_reg <= 0;
    else
        cnt_reg <= cnt_reg + 1;
end
```

同样的，LED 初始化过程状态置为 1，LED 状态记录部分代码如下：

```
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
```

而由于仿真验证过程中发现由于每个周期时间较长，计数器达到最大值的次数非常少，从而导致我们的 LED 状态改变次数极少，所以我们对计数器位数以及状态改变的判定部分代码进行了修改，修改后的 LED 状态改变判定部分代码如下：

```
always @ (posedge clock) begin
    if(reset)
        light_reg <= 8'h01;
     else if (cnt_reg == 2'b11) begin
        if(light_reg == 8'h80)
             light_reg <= 8'h01;
        else
            light_reg <= light_reg << 1;
    end
end
```

4 结果验证

对于本实验中实现的 LED 流水灯模块采用软件仿真的方法进行测试，用 Verilog 语言编写测试激励文件，进行仿真，观察 LED 状态变量以及计数器的数值变化，比对是否符合预期。
仿真得到的波形如图1及图2所示：

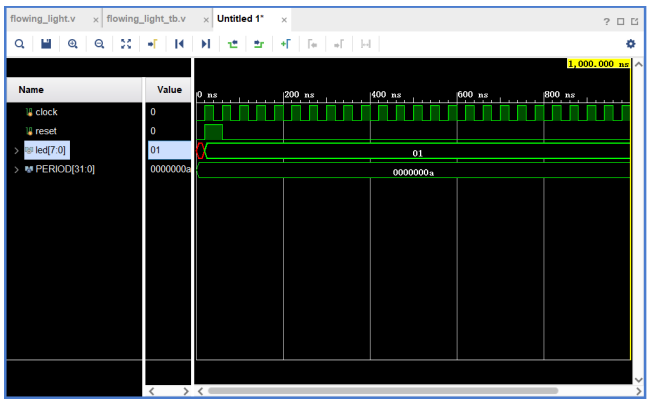

图 1: 原始计数器位数下波形图

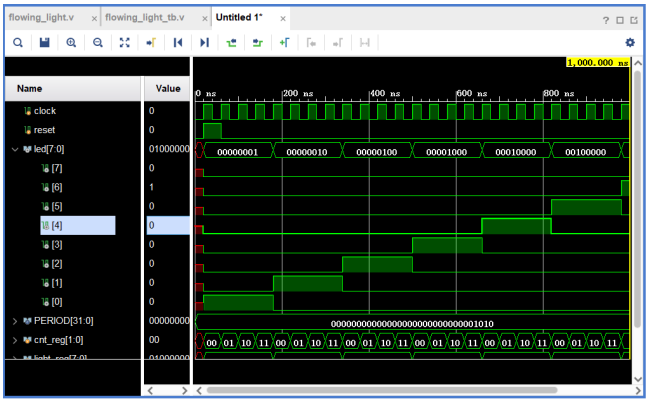

图 2: 修改计数器位数后波形图

其中红色表示输出为不定值，这是因为在仿真最开始的时候还没有进行初始化。而原始计数器位数下，计数器还没有达到最大值的时候仿真程序就已经结束运行了，所以我们减少了计数器位数之后进行的测试数值可以看出每隔一小段时间 LED 灯就向下一位切换点亮，证明本次实验验证成功。

5 总结与反思

本实验实现了 FPGA 实验中 LED Flow Water Light 器件的简单仿真，其功能实现逻辑简单，主要是为了熟悉 Verilog 的代码语法形式以及 Vivado 的文件操作以及界面功能，为后面实验中所要用到的更复杂的编程逻辑以及 debug 所需要的高级技巧奠定基础。同时，在实际操作过程中，我并没有局限于实验指导书上给出的实现形式，而是参考了 Verilog 语言指导书上的一些知识，直接通过代码的形式来为模块添加 I/O 接口，同时对于 always 语句块中的赋值操作不同的形式有了初步的了解。