**计算机系统结构试验** **Lab01: Flowing light**

姓名：N/A

摘要

在Lab01中，我使用Verilog语言成功实现了Flowing light功能。通过本次实验，我学会了如何创建module和simulation文件，并初步理解了Vivado的语法、项目流程、仿真方法和调试技巧。这次实验给我带来了很多收获。

目录

[摘要 1](#_Toc165149306)

[**1.** 实验目的 2](#_Toc165149307)

[**2.** 原理分析 2](#_Toc165149308)

[2.1 Vivado工程的基本组成 2](#_Toc165149309)

[2.2 Flowing light的原理 2](#_Toc165149310)

[**3.** 功能实现 3](#_Toc165149311)

[**4.** 结果验证 3](#_Toc165149312)

[4.1测试用激励文件 3](#_Toc165149313)

[4.3 调整控制逻辑以观察移位 4](#_Toc165149314)

[**5.** 管脚约束 5](#_Toc165149315)

[**6.** 总结与反思 6](#_Toc165149316)

**1.** 实验目的

（1）通过基础实验熟悉Xilinx逻辑设计工具Vivado开发环境；

（2）了解硬件描述语言Verilog HDL描述功能行为的逻辑；

（3）通过仿真检验电路设计是否预期；

（4）学习使用I/O Planning添加管脚约束；

（5）实现Flowing light的功能；

（6）熟悉系统硬件开发的基本实验流程。

**2.** 原理分析

2.1 Vivado工程的基本组成

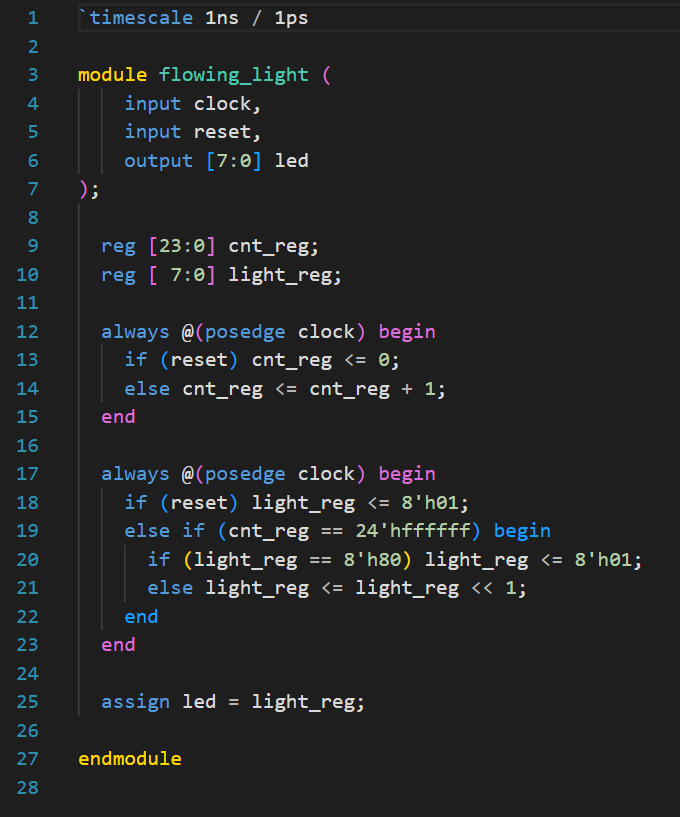
（1）design source.v文件

（2）simulation source.v文件

（3）constraints.xdc文件（上板验证所需的管脚约束文件）

2.2 Flowing light的原理

Flowing light要求在一段时间内，8个LED灯依次轮流亮灭，最后一个LED熄灭后，第一个LED循环亮起。这个功能可以使用移位来实现控制，每位对应一个LED灯。每次到达计数值时，light\_reg中的值循环左移一位。



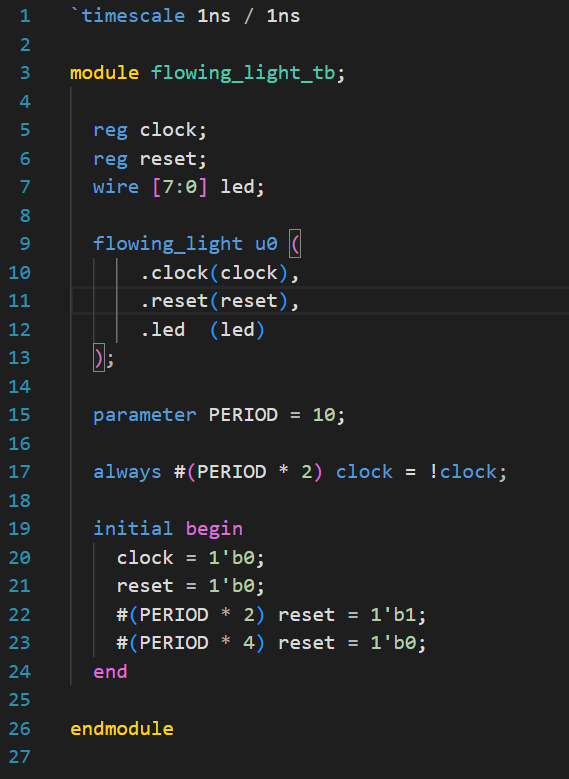
**3.** 功能实现

本实验比较简单，基于上述的原理容易实现flowing light的功能。在实现flowing\_light.v后，生成flowing\_light\_tb.v的激励文件用以仿真测试，生成flowing\_light\_xdc.xdc 的管脚约束用以练习。

**4.** 结果验证

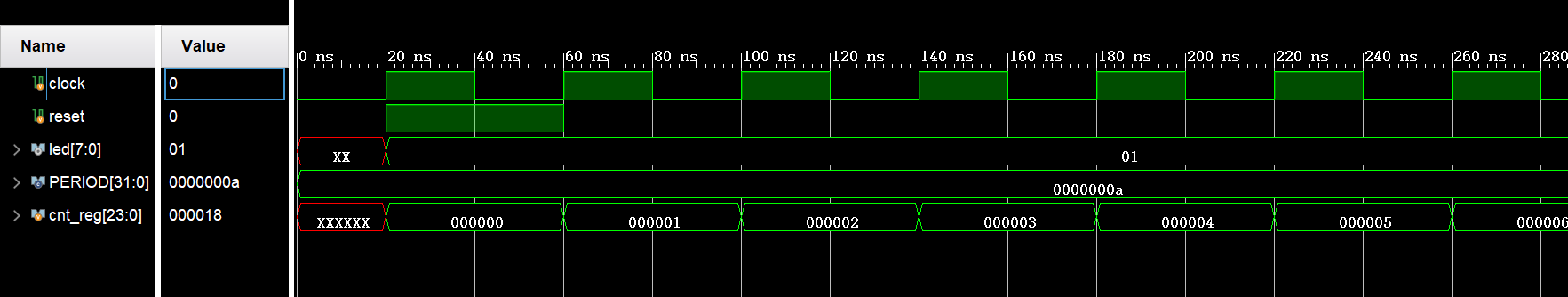
4.1测试用激励文件

首先，按照实验导书上的要求，编写激励文件。设置时钟周期为40ns，并设置各输入初值。代码如下：

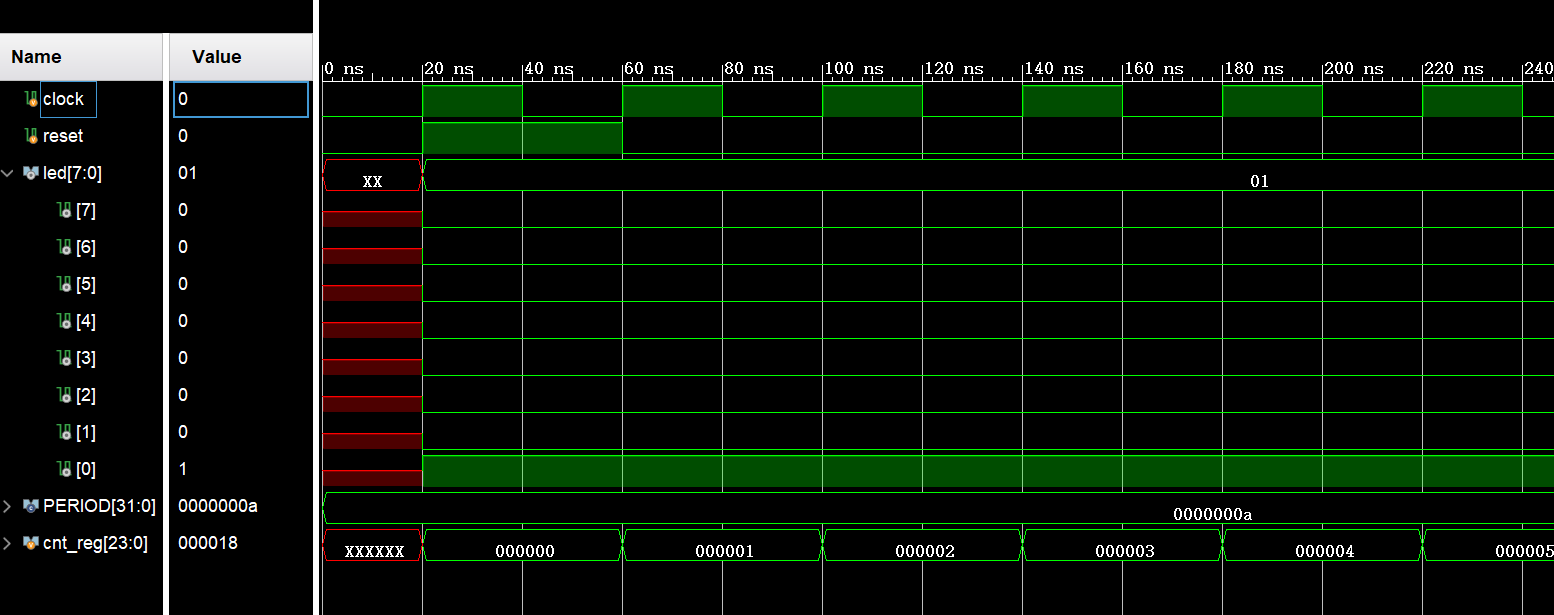


4.2 reset与基本逻辑的测试

首先进行仿真，结果如下所示：



上图中可以看到reset，cnt\_reg，light\_reg功能正常。

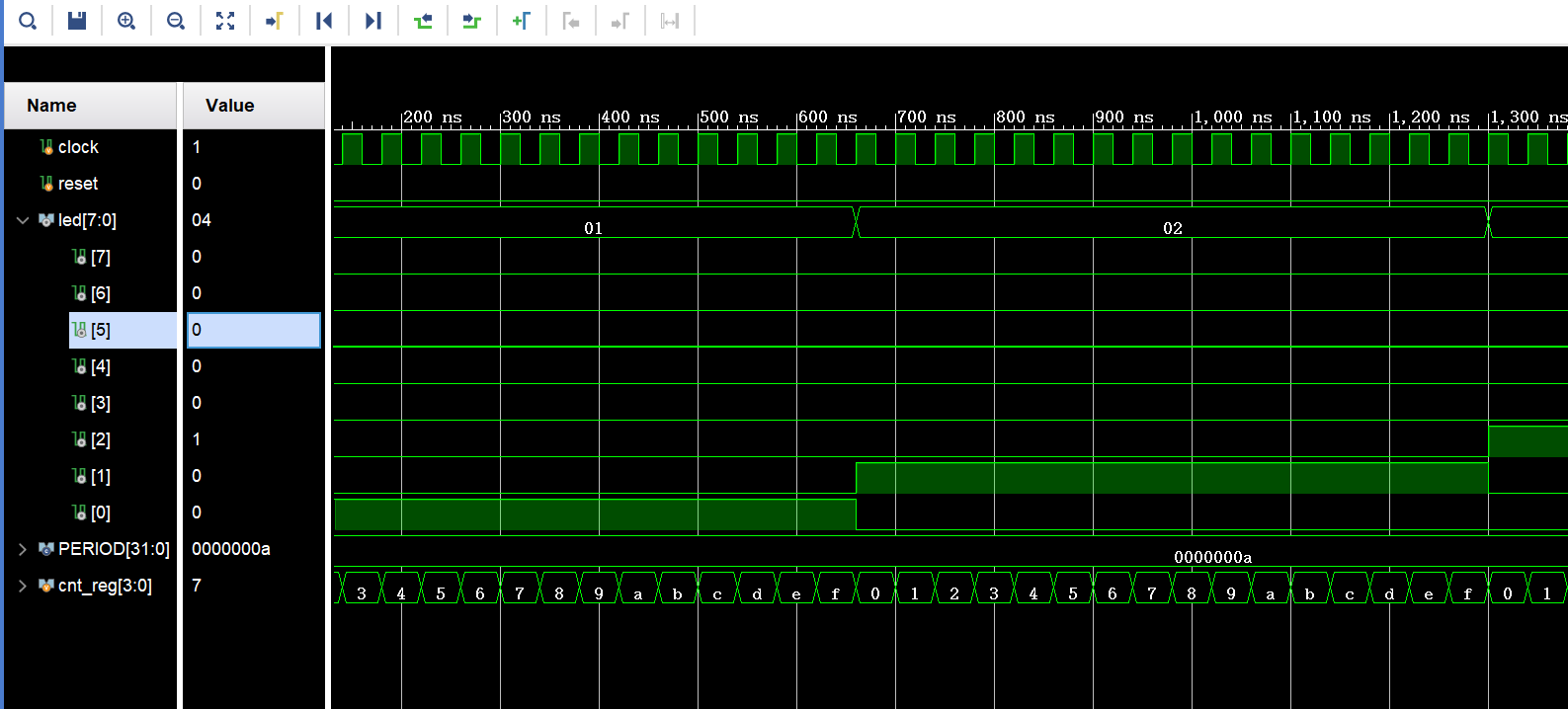


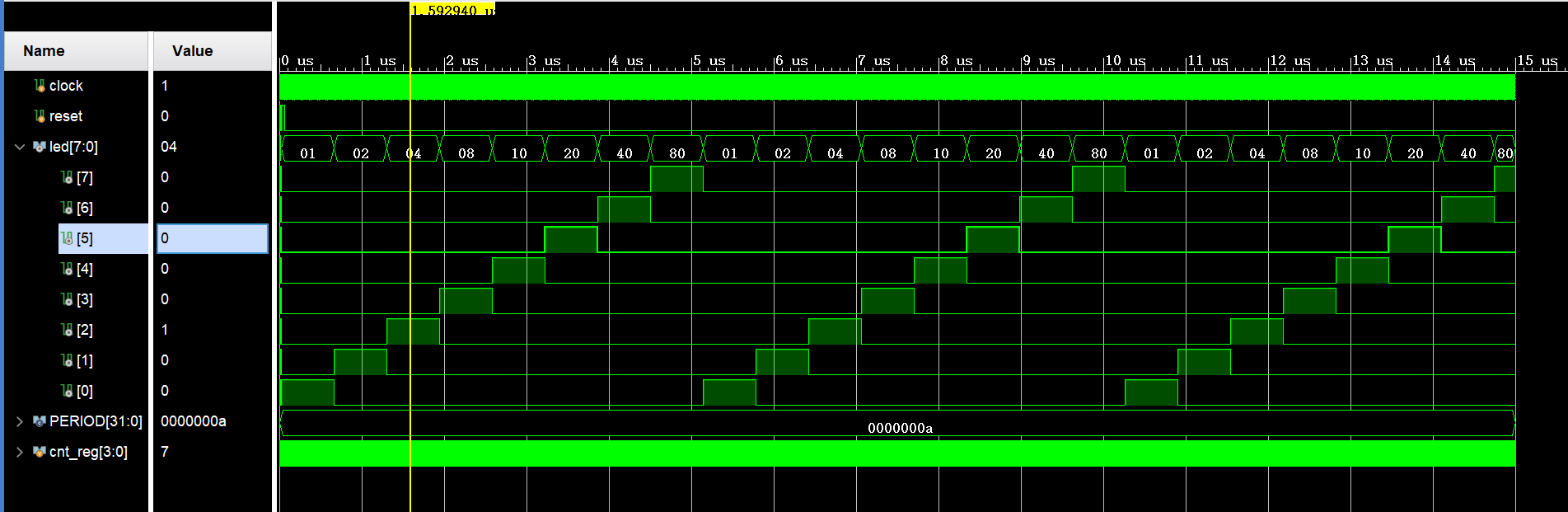
上图中可以看到led每一位的情况。

本仿真运行周期不够，计数器并没加到0xffffff而波形显示早已结束。因此更改计数器的目标值，以便较快速达到左移条件，结果记录于4.3中。

4.3 调整控制逻辑以观察移位

我将计数逻辑改为cnt\_reg==0xf 时移位，即经过 16 个时钟周期就移位，并相应把cnt\_reg改为4位。在仿真模拟中观察到了移位，如下图所示：

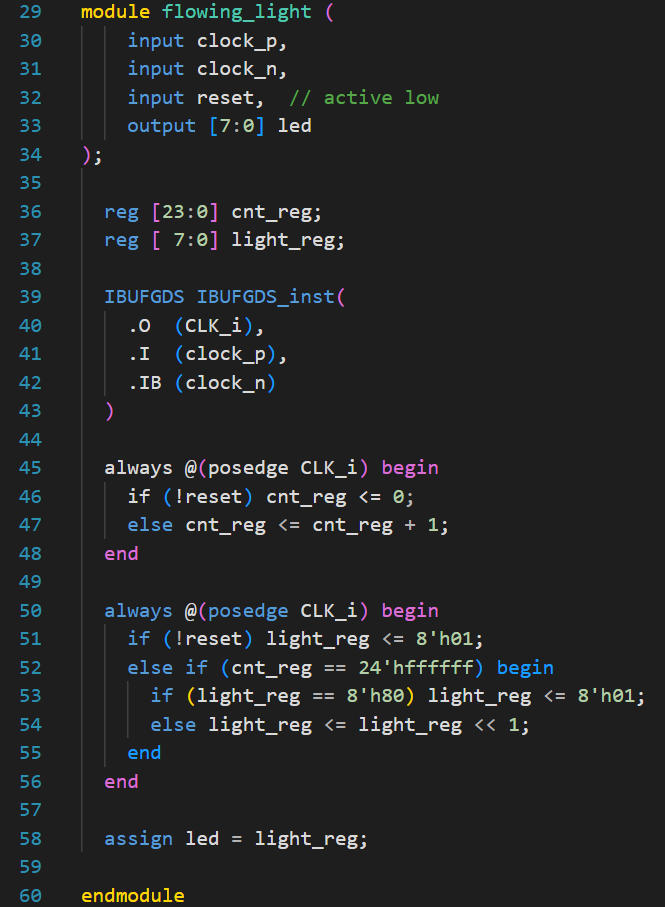




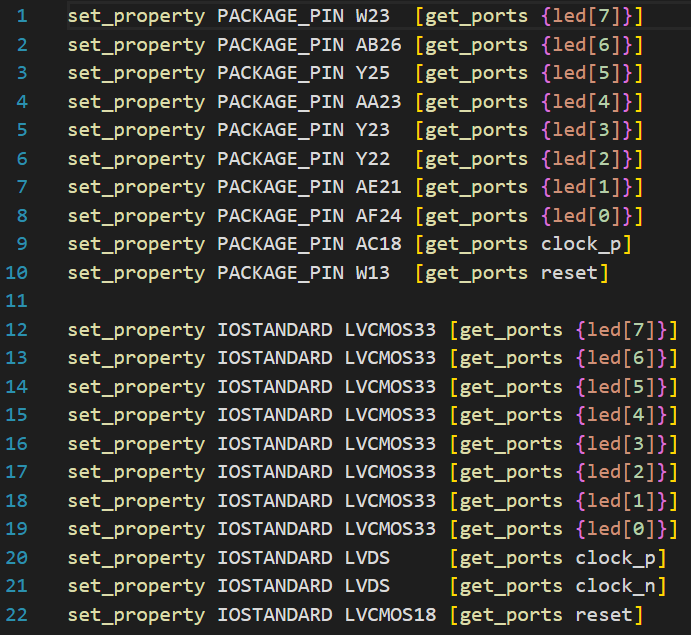
这样，我通过改变计数器的控制逻辑在仿真中观察到了移位。

**5.** 管脚约束

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。修改代码如下：



同时设置管脚约束如下：



**6.** 总结与反思

在工科创中我曾使用过Vivado通过这次实验，我再次熟悉了Vivado的开发环境，因此上手并不困难。在这个实验中，我不仅复习了 Verilog HDL 的基本语法和项目开发流程，还学习了仿真激励文件的写法。

我要感谢课程组为我们准备的详细指导书。在接下来的学习中，我计划进一步巩固 Verilog HDL 的知识，尝试更复杂的项目，并探索其他开发工具和技术，以提升我的硬件设计能力。