**计算机系统结构试验** **Lab02: 4-bit Adder**

姓名：N/A

摘要

在Lab02中，我使用Verilog语言成功实现了4-bit Adder功能。通过本次实验，我学会了如何在项目中编写多个module的组合逻辑，并进一步理解了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](#_Toc165238210)

[**1.** 实验目的 2](#_Toc165238211)

[**2.** 原理分析 2](#_Toc165238212)

[2.1 Vivado工程的基本组成 2](#_Toc165238213)

[2.2 adder\_1bit的原理 2](#_Toc165238214)

[2.3 adder\_4bits的原理 2](#_Toc165238215)

[**3.** 功能实现 3](#_Toc165238216)

[**4.** 结果验证 4](#_Toc165238217)

[4.1测试用激励文件 4](#_Toc165238218)

[4.2 加法逻辑的测试 5](#_Toc165238219)

[**5.** 管脚约束 5](#_Toc165238220)

[**6.** 总结与反思 6](#_Toc165238221)

**1.** 实验目的

（1）掌握Xilinx逻辑设计工具Vivado的基本操作；

（2）掌握Verilog HDL进行简单的逻辑设计；

（3）使用功能仿真；

（4）约束文件的使用和直接写法；

（5）生成bitstream文件；

（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）adder\_1bit.v文件

（2）adder\_4bits.v文件

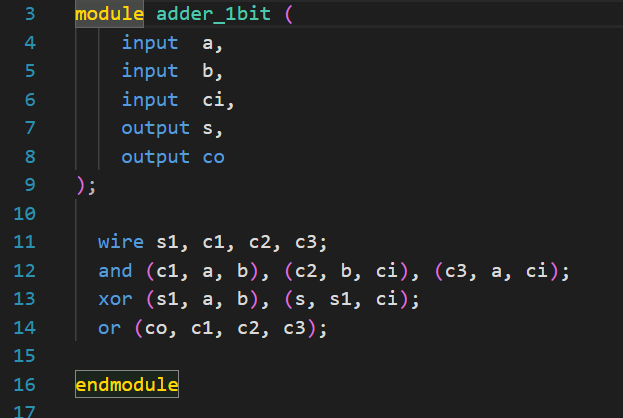
（3）Top.v文件

（4）adder\_4bits\_tb.v激励文件

（5）lab02\_xdc.xdc管脚约束文件

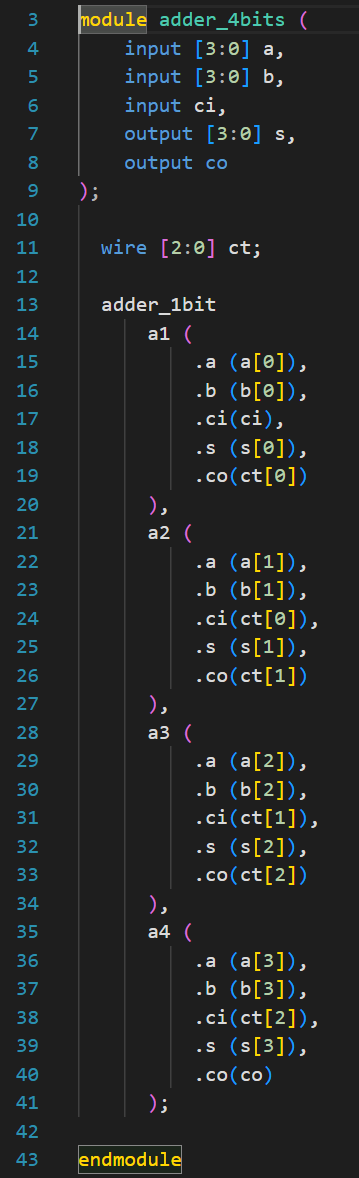
2.2 adder\_1bit的原理

adder\_1bit是一个一位全加器，输入为两个一位操作数a、b以及一位进位输入ci；输出为一位加法结果s与一位进位输出co。根据全加器逻辑，当a、b、ci中至少两个为1时，co为1；当a、b、ci有奇数个1时，s为1。代码如下：



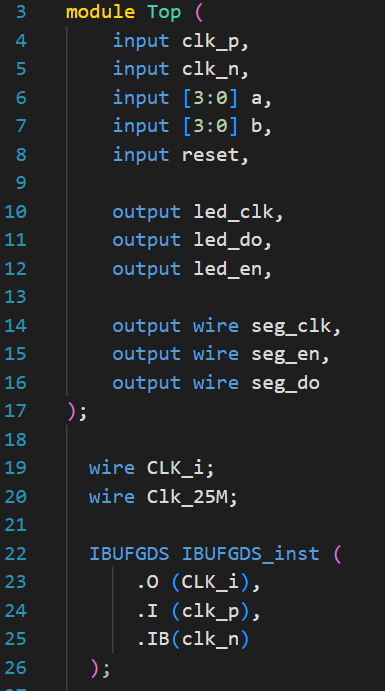
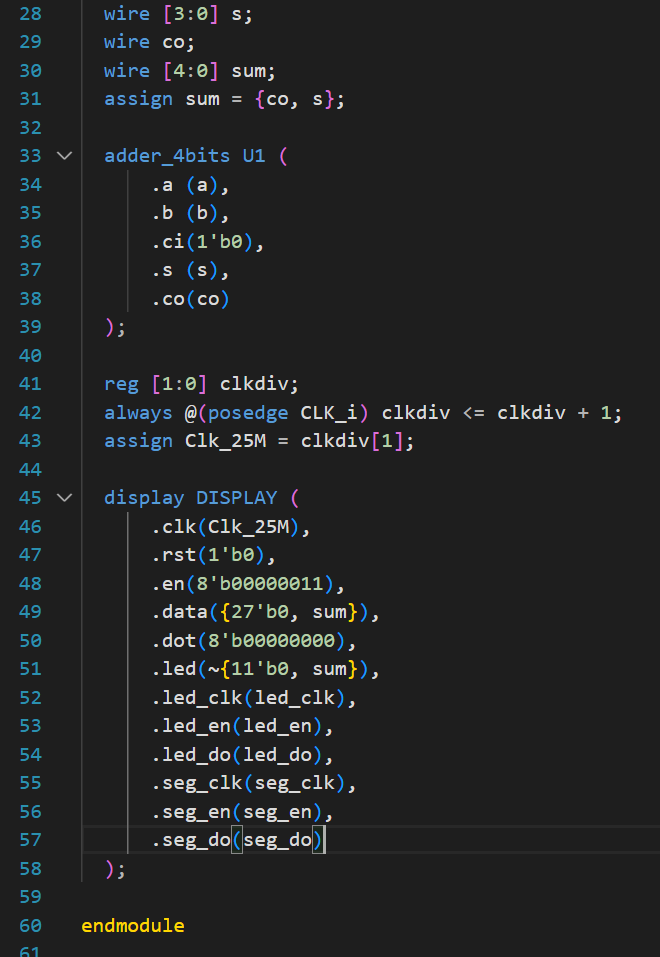
2.3 adder\_4bits的原理

adder\_4bits由4个adder\_1bit组合而成。整个模块输入为两个4位操作数a、b以及一位进位输入ci；输出为4位加法结果与一位进位输出co。模块中添加了ct用来连接一位全加器之间的进位。低位全加器的进位输出是高位全加器的进位输入，最低位全加器的进位输入为ci，最高位全加器的进位输出为co。其余输入输出与a、b、s对应连接。代码如下：



**3.** 功能实现

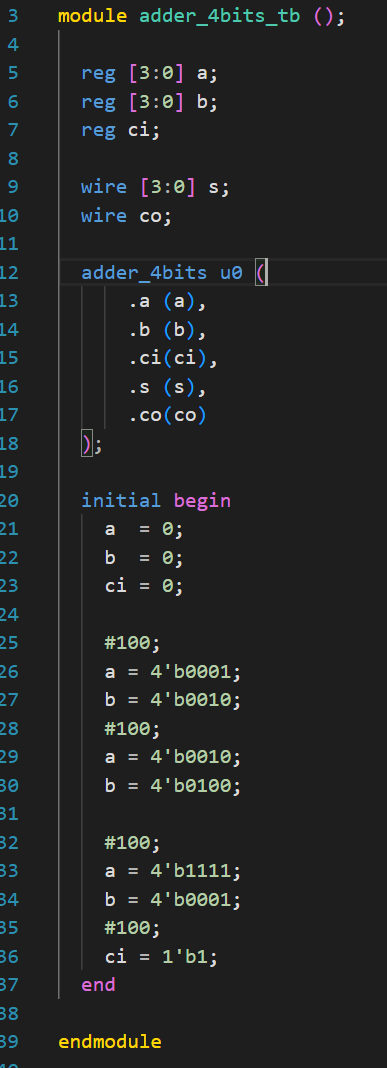
为了使用已实现的4位全加器，拟用实验板上的8个Switch对应二组4位二进制输入，用4个LED发光二极管对应输出，并用2个七段数码管显示运行结果。故本实验需要用到display.v这个七段数码管SEGMENT和LED发光二极管显示模块（实验室提供display核，以网表文件形式给出）。编写Top模块代码如下：



**4.** 结果验证

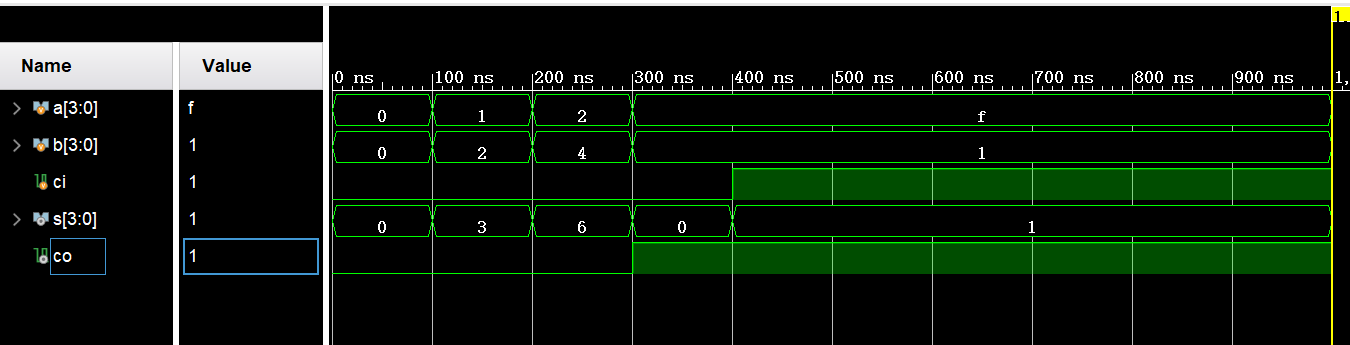
4.1测试用激励文件

首先，按照实验导书上的要求，编写激励文件。设置各输入初值。代码如下：



4.2 加法逻辑的测试

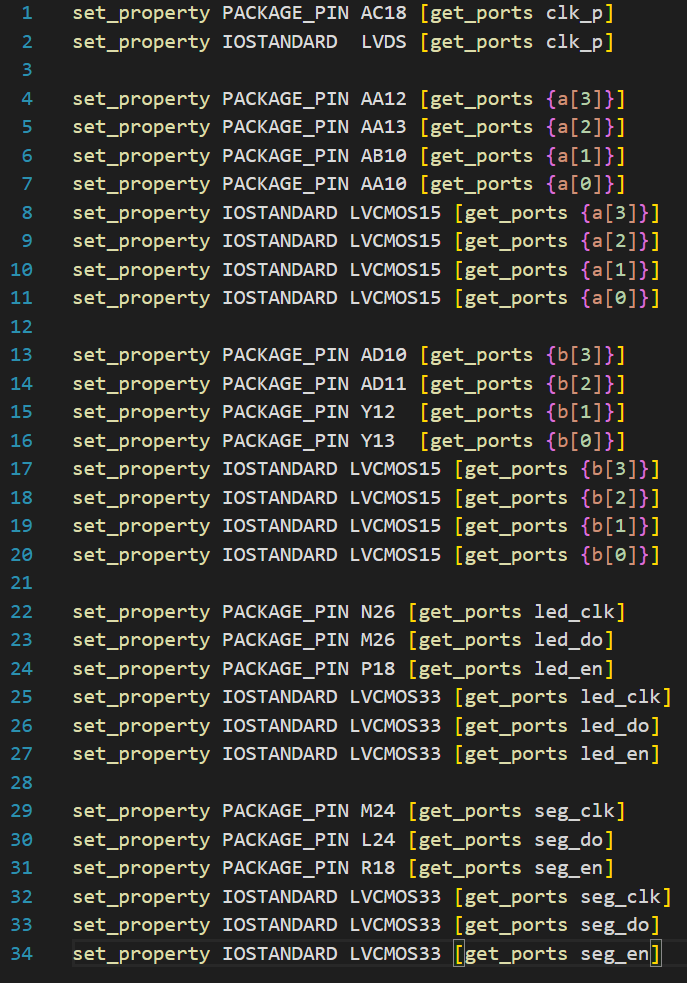
接下来进行仿真，结果如下所示：



上图中可以看到，{co，s}= a + b + ci，加法功能正常。

**5.** 管脚约束

根据实验指导书编写管脚约束文件如下：



**6.** 总结与反思

在Lab02中，我再次熟悉了Vivado的开发环境，并且对Verilog HDL的基本语法有了进一步的复习。此外，我还学习了如何使用module模块化编写组合逻辑，如何使用begin-end块编写时序激励文件。

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，我计划进一步学习 Verilog HDL 的知识，尝试设计通用的模块，提高代码复用性，并学习管脚约束文件的写法。