实验二 加法器的设计及应用

一、实验目的

1.掌握Verilog语言框架，编程及调试的方法。

2.熟悉Verilog的基本语法。

3.掌握Vivado开发平台及FPGA开发板的使用。

二、实验内容

1.完成1位半加器、1位全加器模块的实现与测试；

2.利用1位全加器实现2位全加器，并烧录到开发板进行验证；

3.实现16/32位全加器，记录、分析仿真波形和RTL电路图。

三、实验要求

1.说明电路功能，分析设计、仿真代码和电路图。

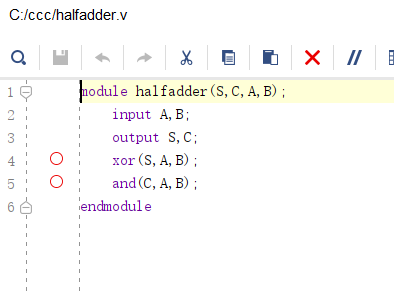
2.分析仿真波形，观察输入输出是否与预期电路功能相符（测试要全面，关注特殊情况的测试）。

3.记录设计和调试过程。

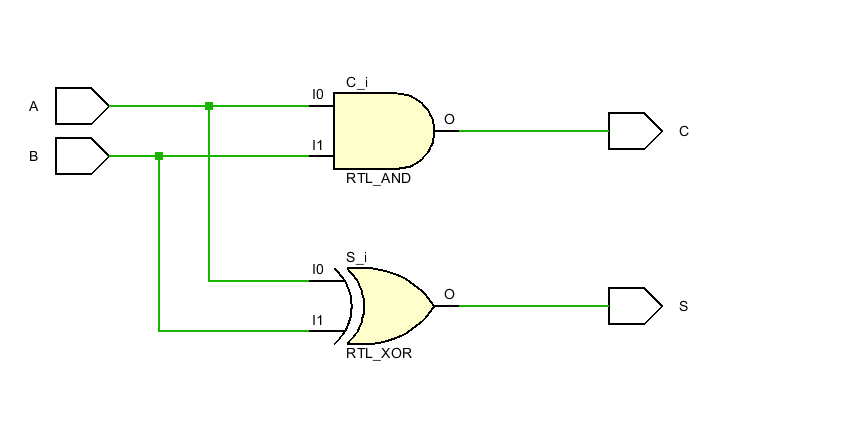
1. 实验过程及结果分析

对于实验内容1：

设计代码1：



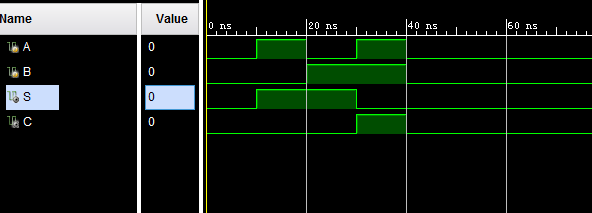
Vivado生成的逻辑电路图：



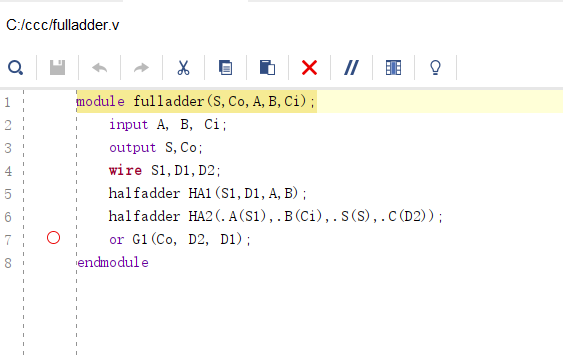
电路功能：实现了一个简单半加器的功能。

设计文件分析：输入两个一位变量A，B，通过异或和与两个逻辑门实现了半加器的功能，输出进位C和结果S。

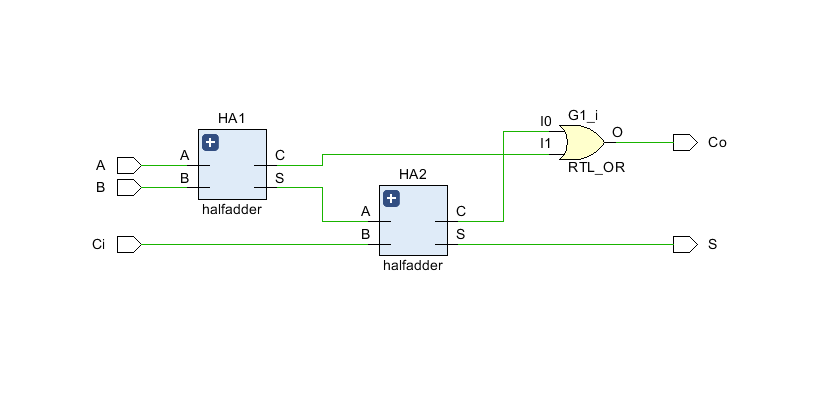
仿真结果图：



设计代码2：



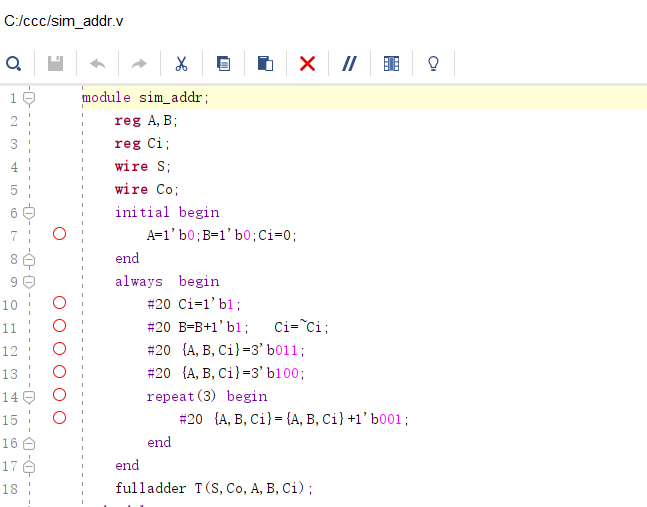
Vivado生成的逻辑电路图：



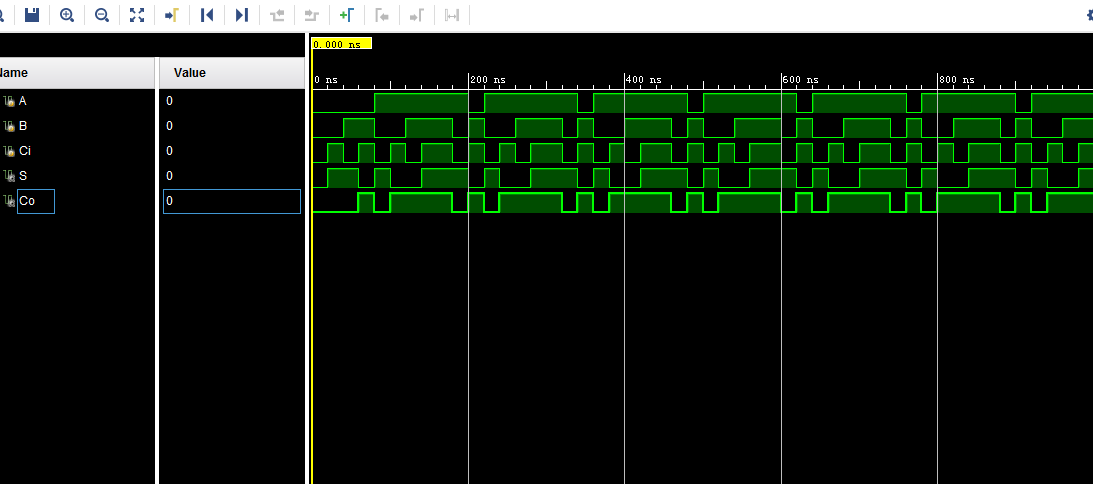
电路功能：实现了一个简单全加器的功能。

设计文件分析：输入三个一位变量A，B，Ci，通过组合两个半加器实现了全加器的功能，输出进位Co和结果S。

仿真代码：



仿真结果图：



仿真代码分析：首先定义输入的A，B值以及进位Ci=0，之后

延迟20个时间单位后，将Ci置为1；

延迟20个时间单位后，B置为1，Ci翻转；

延迟20个时间单位后，三个输入分别置为1，1，0；

延迟20个时间单位后，三个输入分别置为0，0，1；

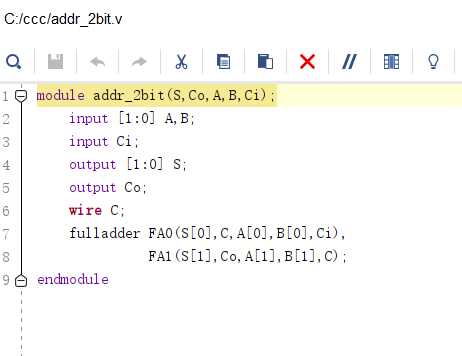
之后重复三次：将A，B，Ci拼成的寄存器+1的值赋给A，B，Ci拼成的寄存器。

最后是调用fulladder模块检查输出结果。

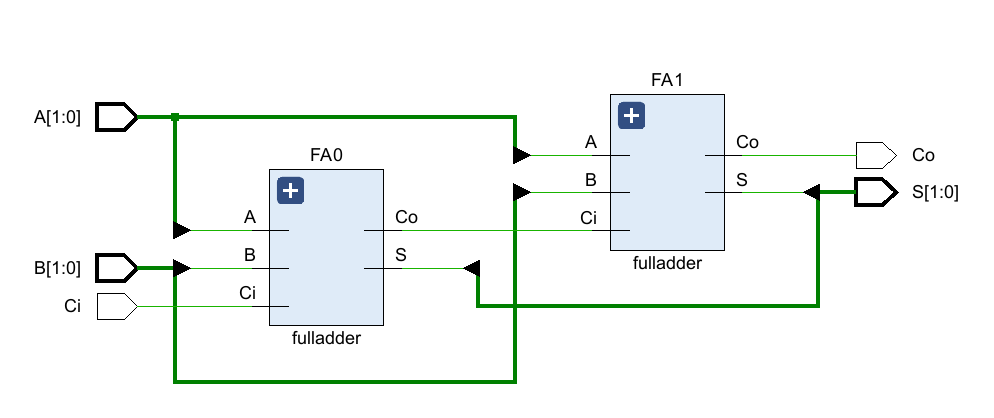
仿真结果/波形图分析：从上往下每行依次是 A，B，Ci，S，Co的波形图。可以发现，A，B，Ci 的波形图都满足上面的分析，同时，S和Co也满足全加器规则。 显然满足预定的功能。

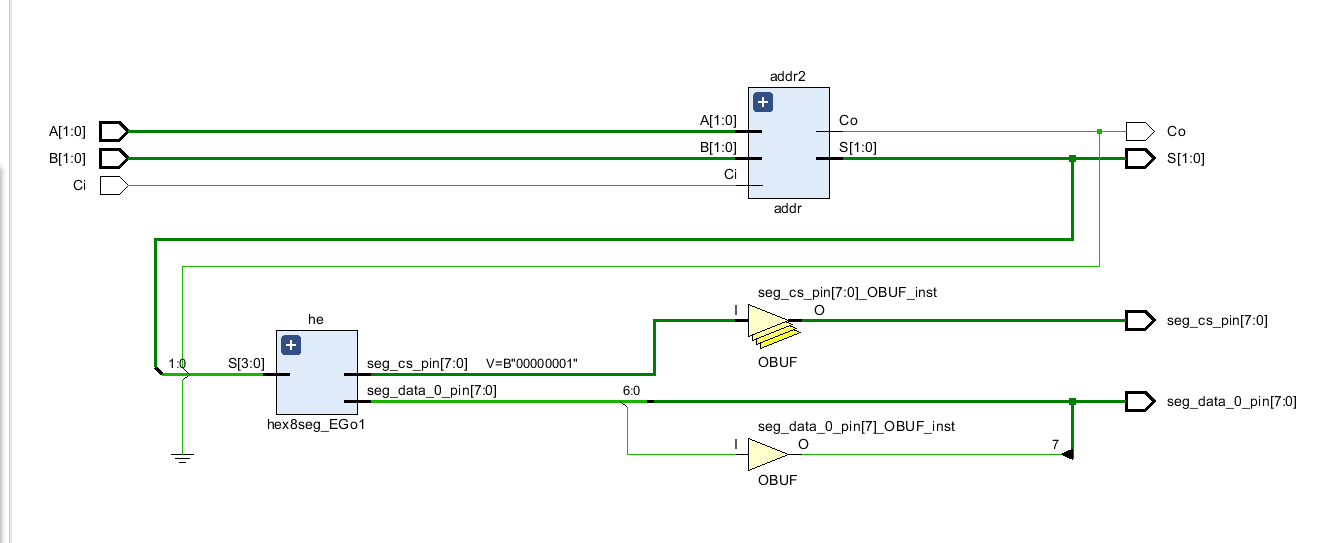
对于实验内容2：

设计代码：

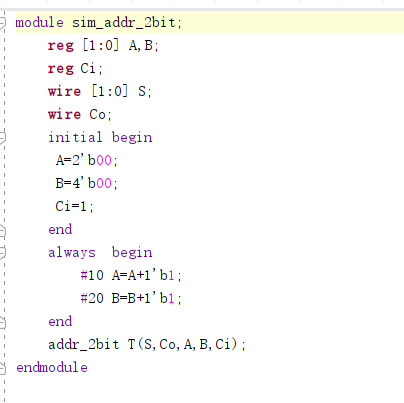


Vivado生成的逻辑电路图：

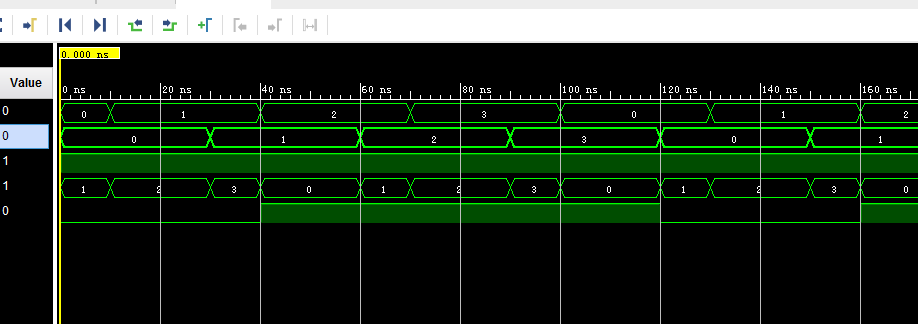




仿真代码：



仿真结果图：



电路功能分析：实现了两位全加器的功能。

设计文件分析：输入两位变量A，B，和一位变量Ci，输出进位C和结果S。

仿真代码分析：首先初始化输入的A，B值=00以及进位Ci=1，之后

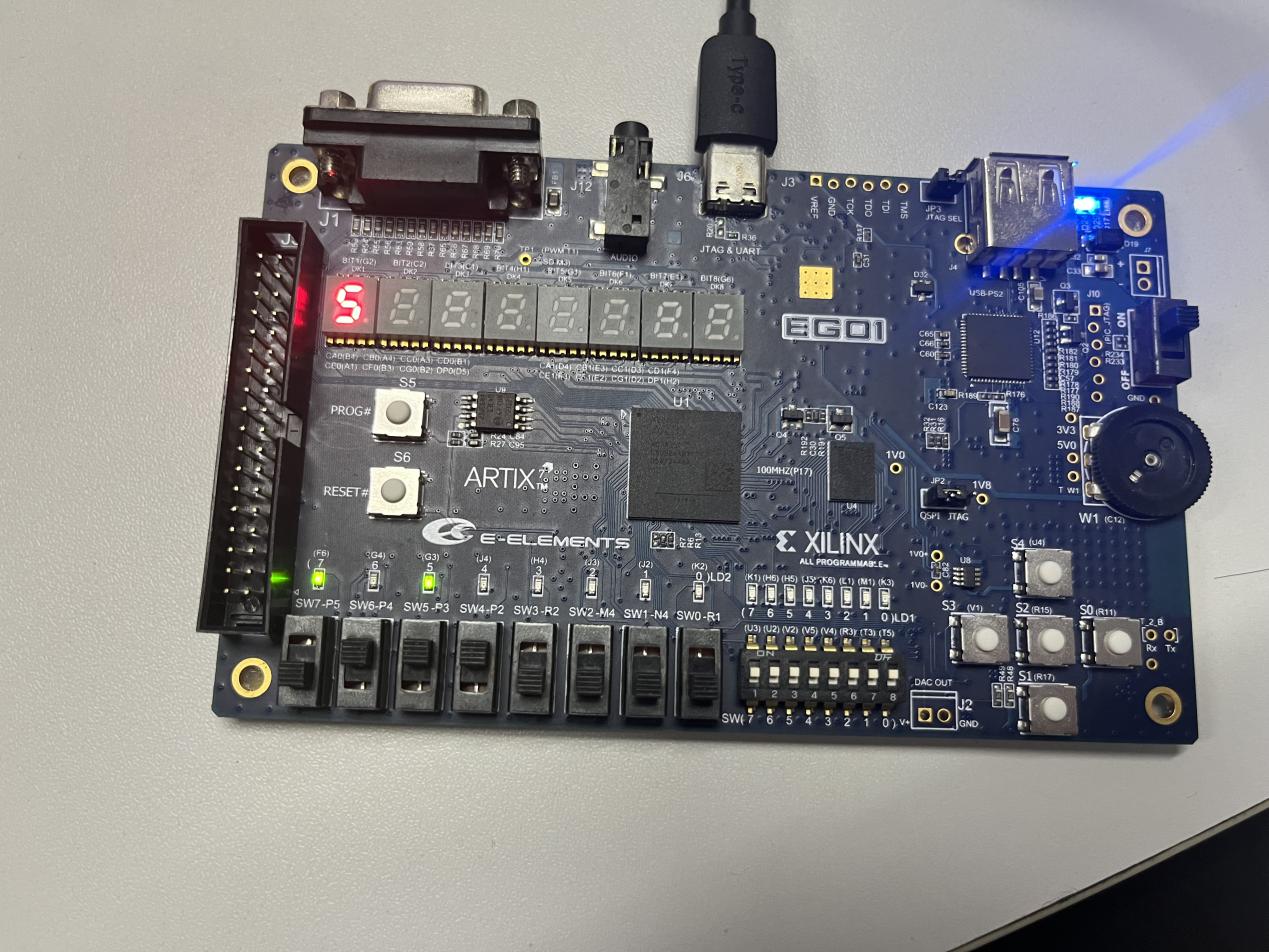
延迟10个时间单位，A的值加1；

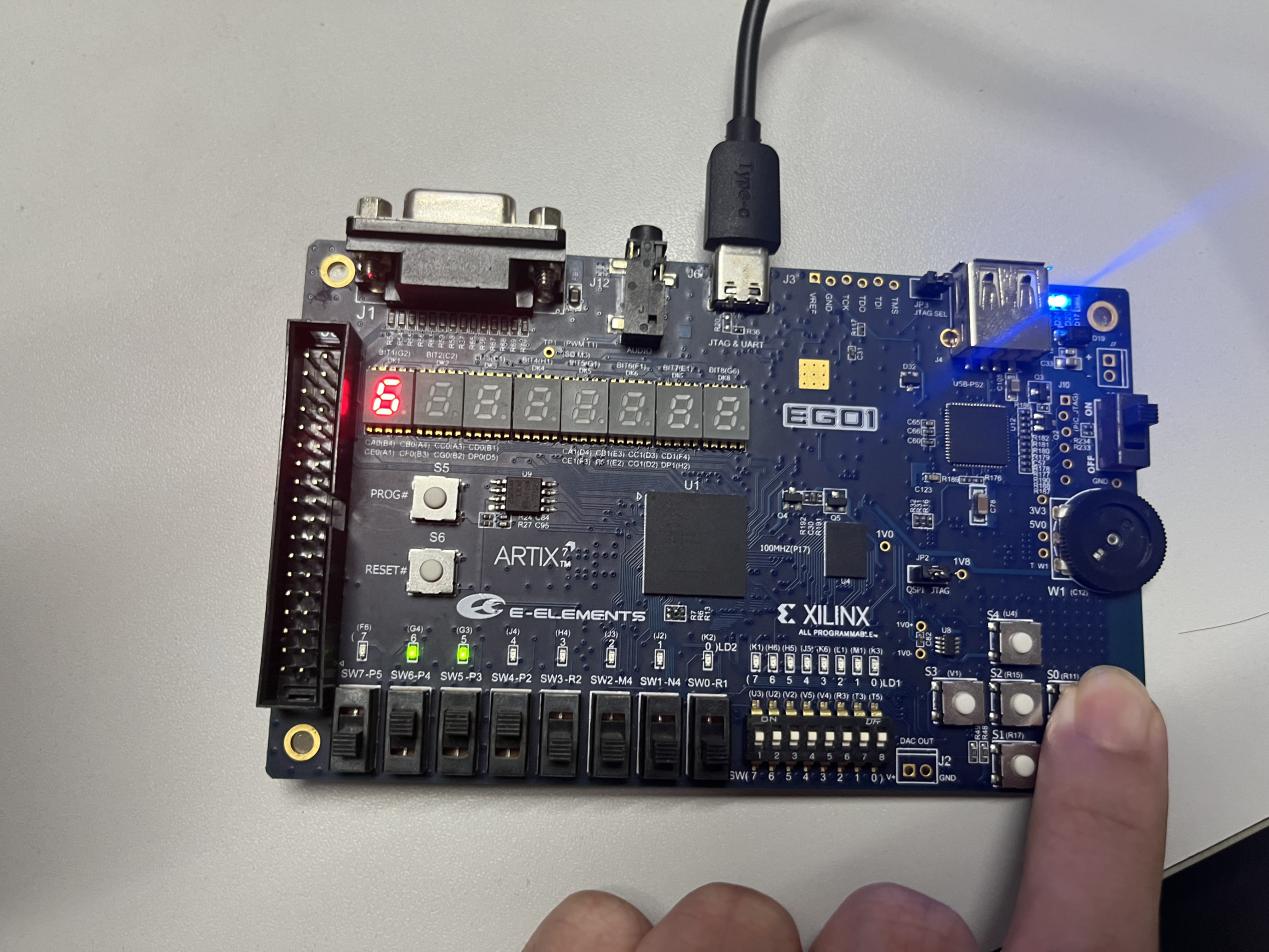
延迟20个时间单位，B的值加1；

之后调用addr\_2bit模块检测输出结果。

仿真结果/波形图分析：从上往下每行依次是 A，B，Ci，S，Co的波形图。可以发现，A，B，Ci 的波形图都满足上面的分析，同时，S和Co也满足全加器规则。 显然满足预定的功能。

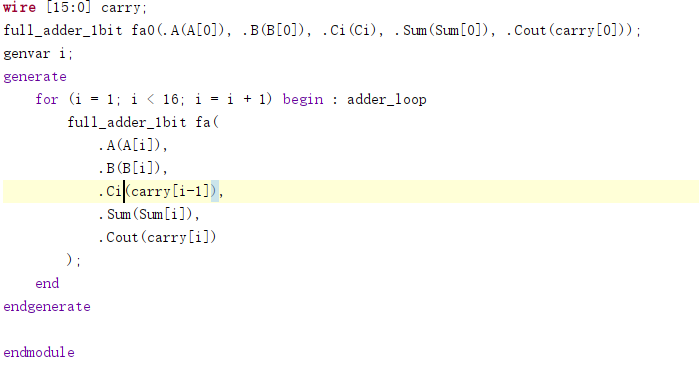
烧录到开发板进行的验证：



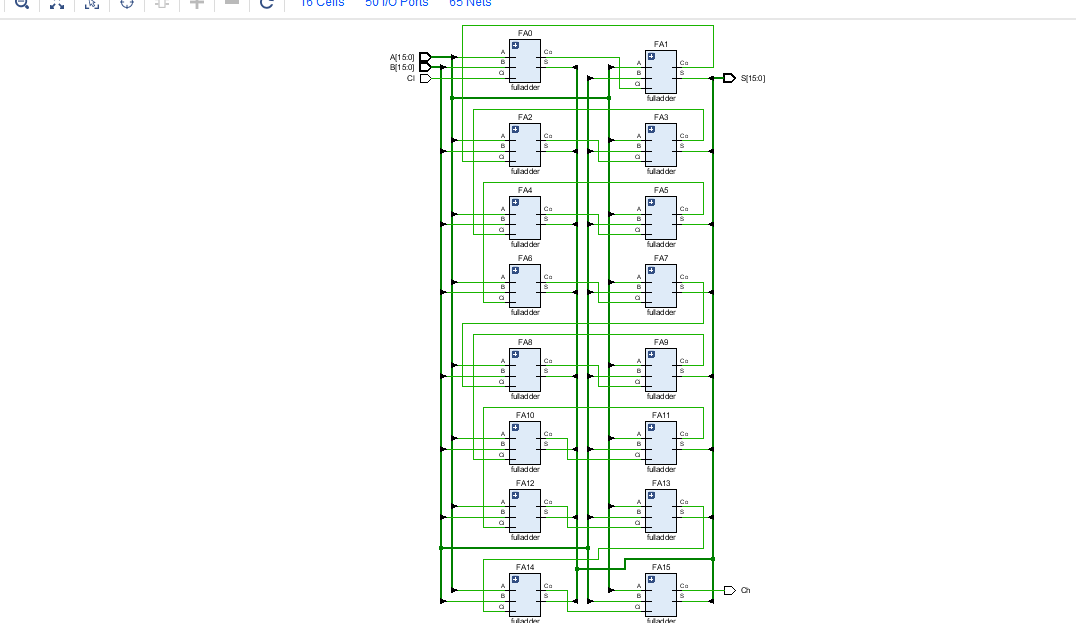


对于实验内容3：

设计文件：



Vivado生成的逻辑电路图：



电路功能分析：实现了16位全加器的功能。

设计文件分析：输入16位变量A，B，和一位变量Ci，输出进位C和结果S。

**对电路功能、设计代码、电路图、测试结果等进行简要分析说明，不能只贴截图。**

五、调试和心得体会

**记录实验过程中遇到的问题和调试解决过程，心得体会等。**

调试中遇到的困难和解决过程：

困难：无法识别接入的开发板。

解决：调整远程桌面设置。

心得体会和收获：

1. 理解了半加器和全加器的工作原理和设计过程。
2. 自主实现了16位全加器的功能。
3. 在开发板上完成两位全加器的烧录和验证。