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，观察、分析仿真波形和RTL电路图。

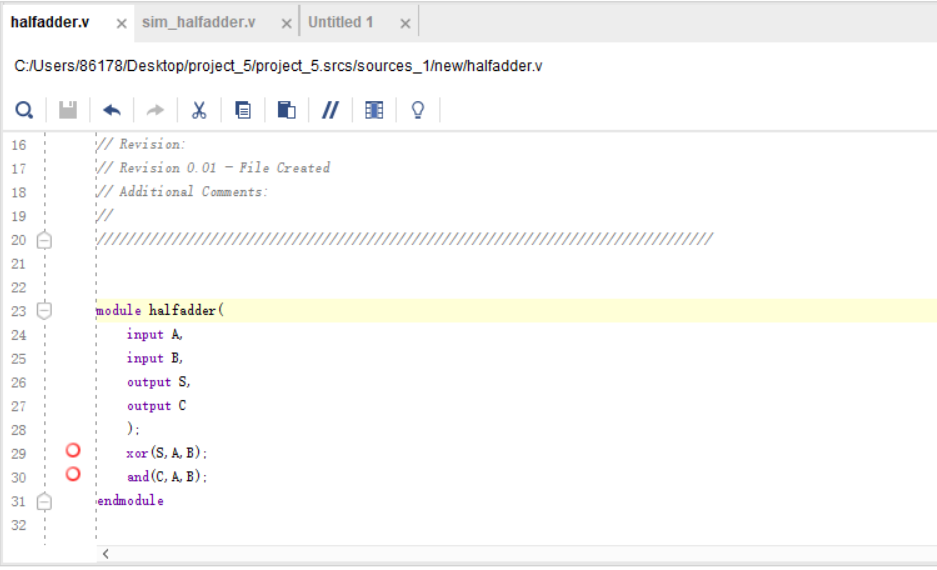
三、实验要求

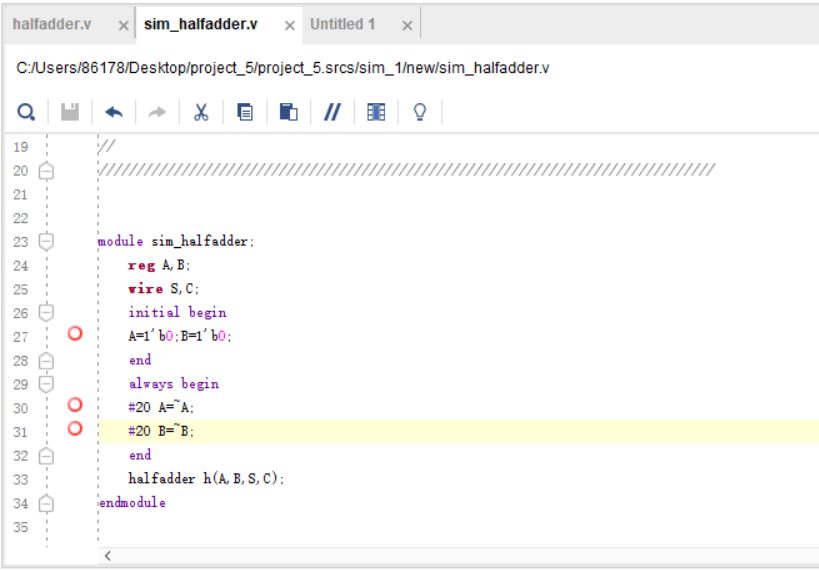
1. 画出模块的电路图（RTL分析）。
2. 分析电路的仿真波形
3. 记录设计和调试过程。

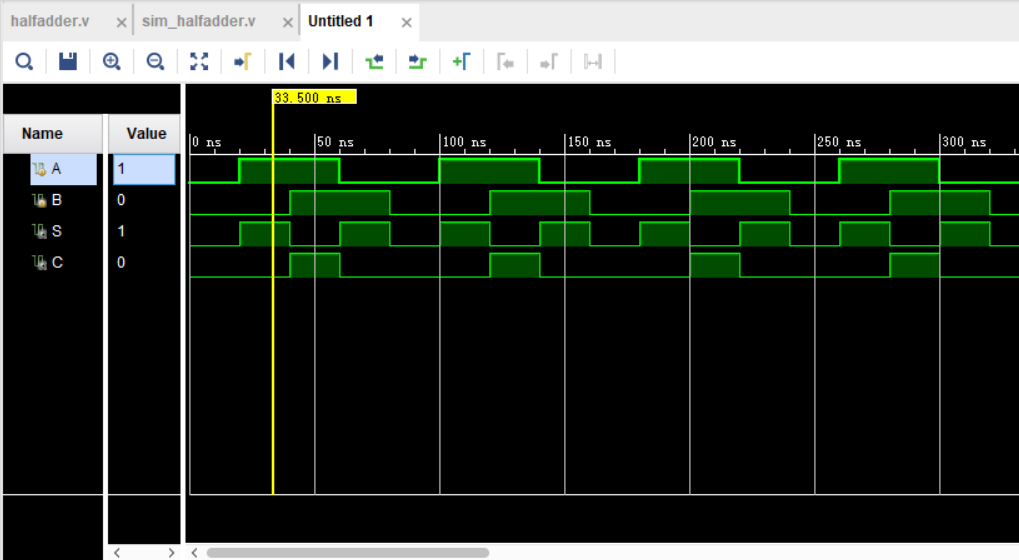
四、实验过程及结果分析

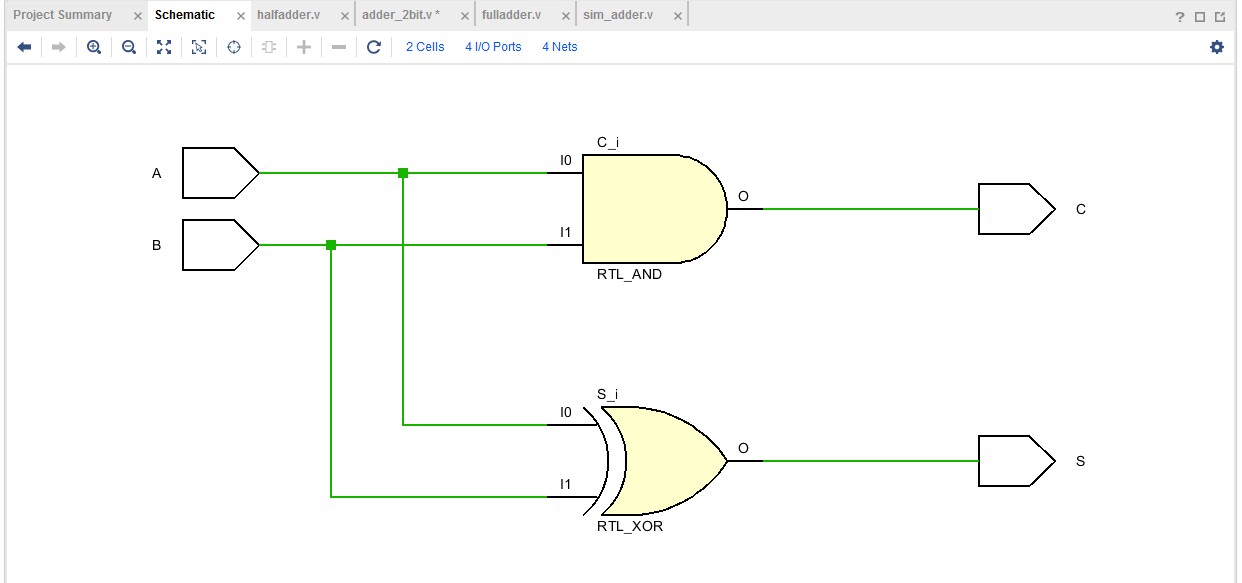
**对实验代码、仿真结果等进行分析说明，不能只贴截图。**

**1位半加器：**







****

分析：

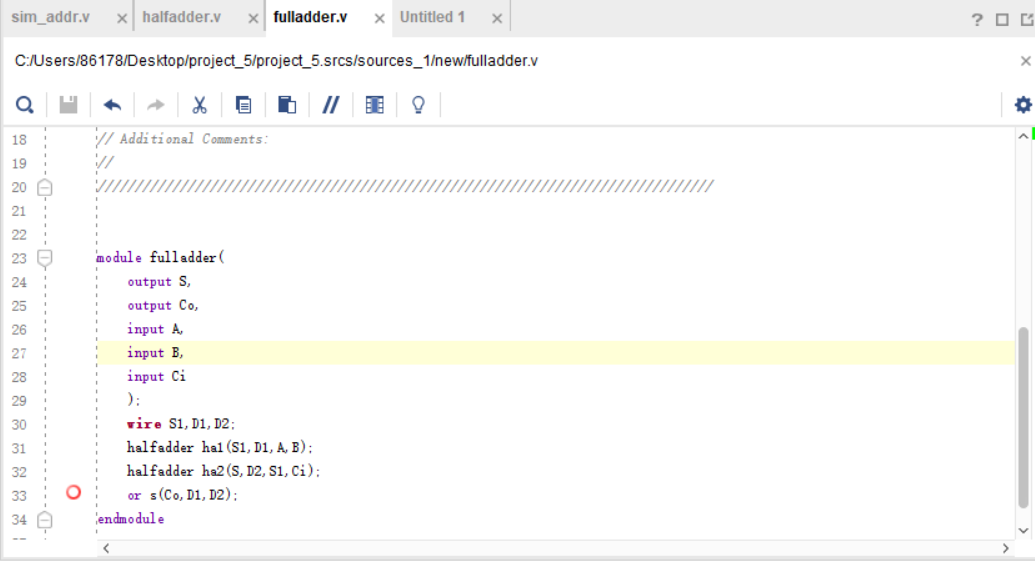
半加器的和S为输入A和B的异或，即当A和B中一个为1一个为0的时候S才为1，当A和B相同的时候，S为0

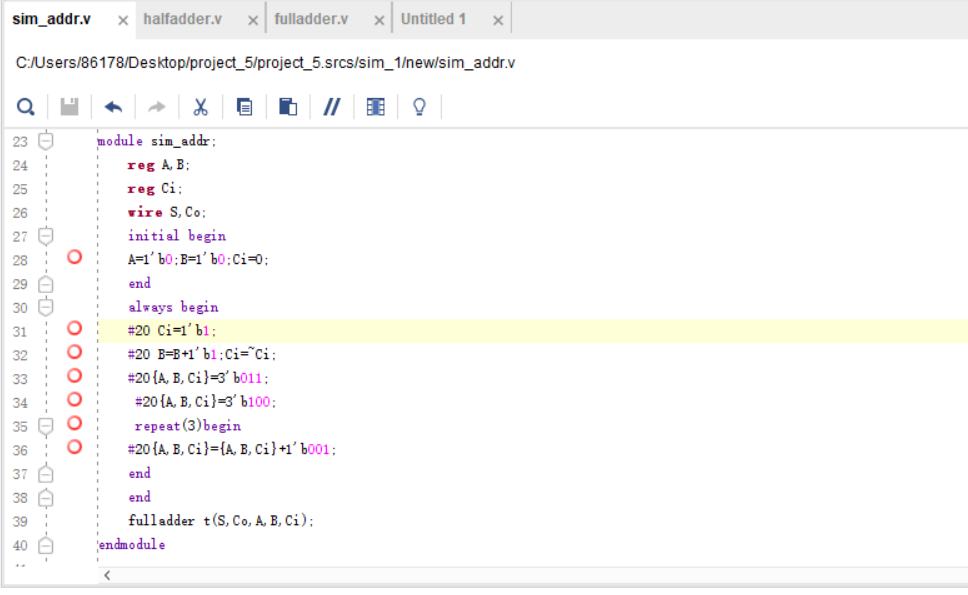
当且仅当A和B同时为1的时候进位C的值才为1，其余情况C都为0

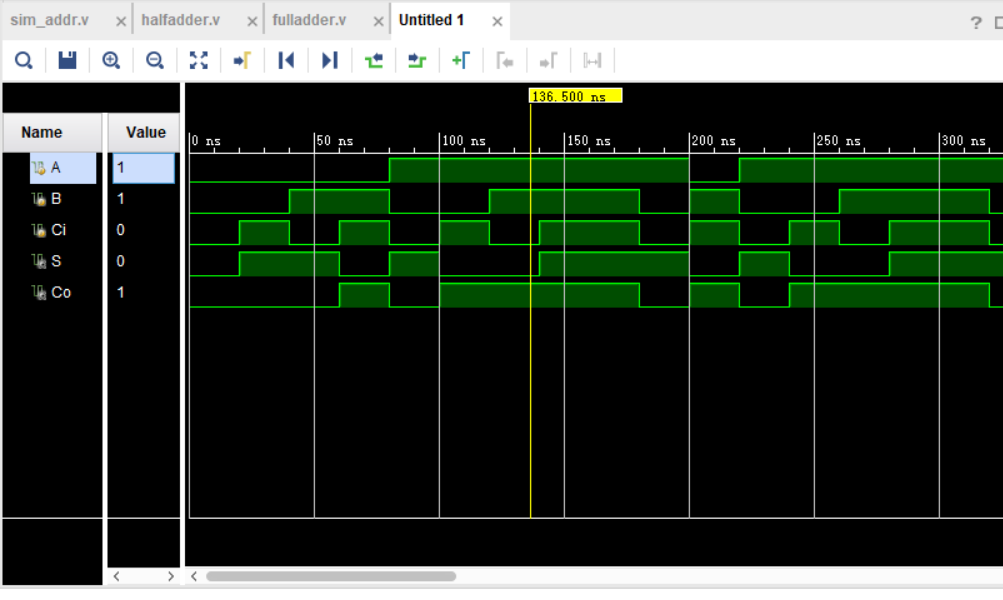
将A和B进行异或，输出结果为S,将A和B进行逻辑与操作，输出结果为C;

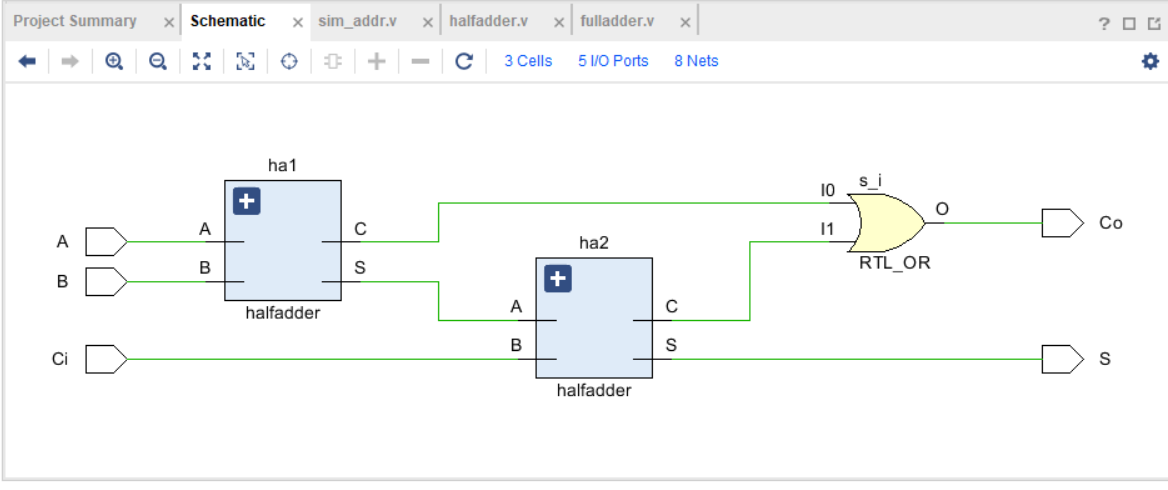
仿真时先将输入A和B初始化为0，再让A、B依次取反，不断循环，最后得到的仿真结果如预期所示

**1位全加器**：









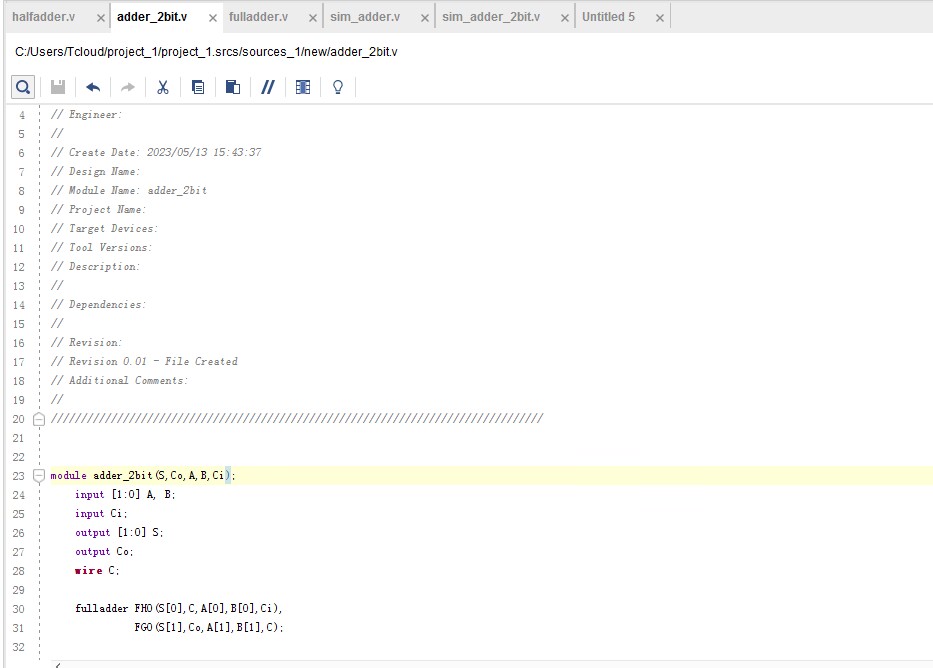
分析：

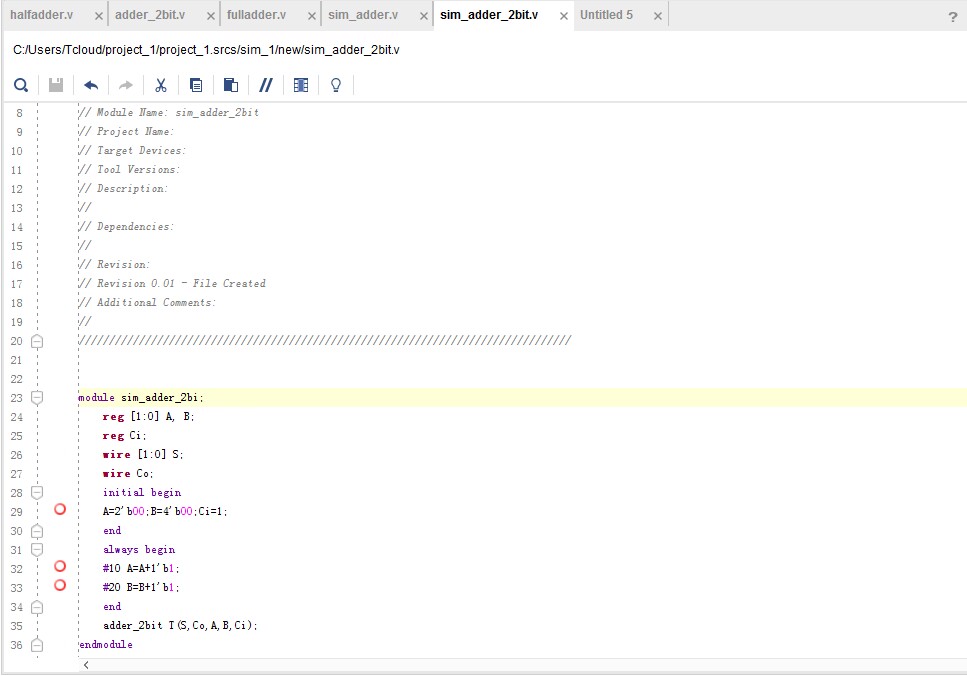
全加器的构建可以依赖半加器，输入A,B,Ci,先设置三个wire变量S1,D1,D2, 让A和B作为半加器的输入，得到和S1和进位D1,再让S1和Ci作为半加器的输入，得到的和即为要求的和S，得到的进位D2和前一个进位D1只要有一个为1，最终的进位Co就为1，所以对D2和D1进行或操作，得到的结果即为进位Co。

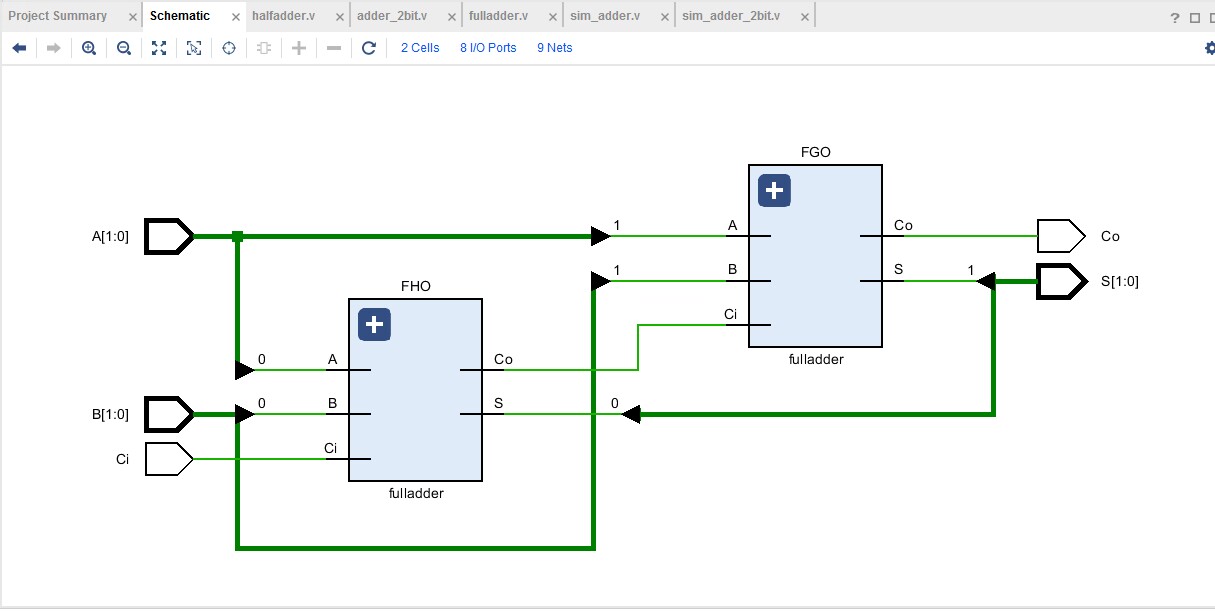
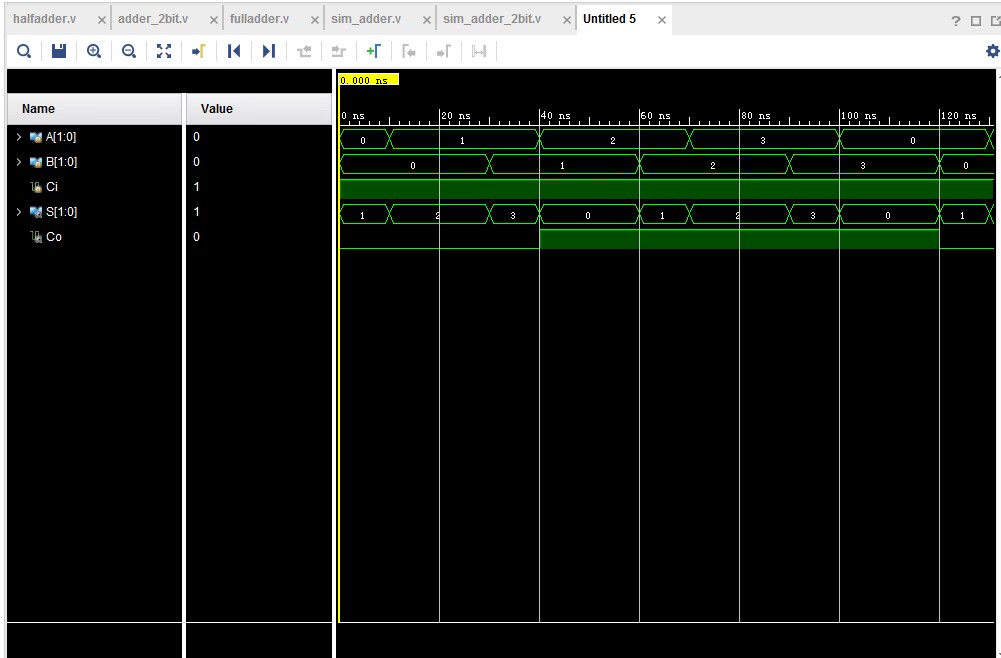
仿真时先将A,B,Ci全部初始化为0，进入循环，隔20ns后将Ci置1，再隔20ns后将B进行+1操作，同时将Ci置0，再隔20ns后将A,B,Ci分别置为0，1，1，再隔20ns后将A,B,Ci分别置为1，0，0，再进入一个小循环，在循环内部每隔20ns对A,B,Ci构成的3位二进制数进行+1操作，重复进行三次后退出小循环，进入下一个大循环.

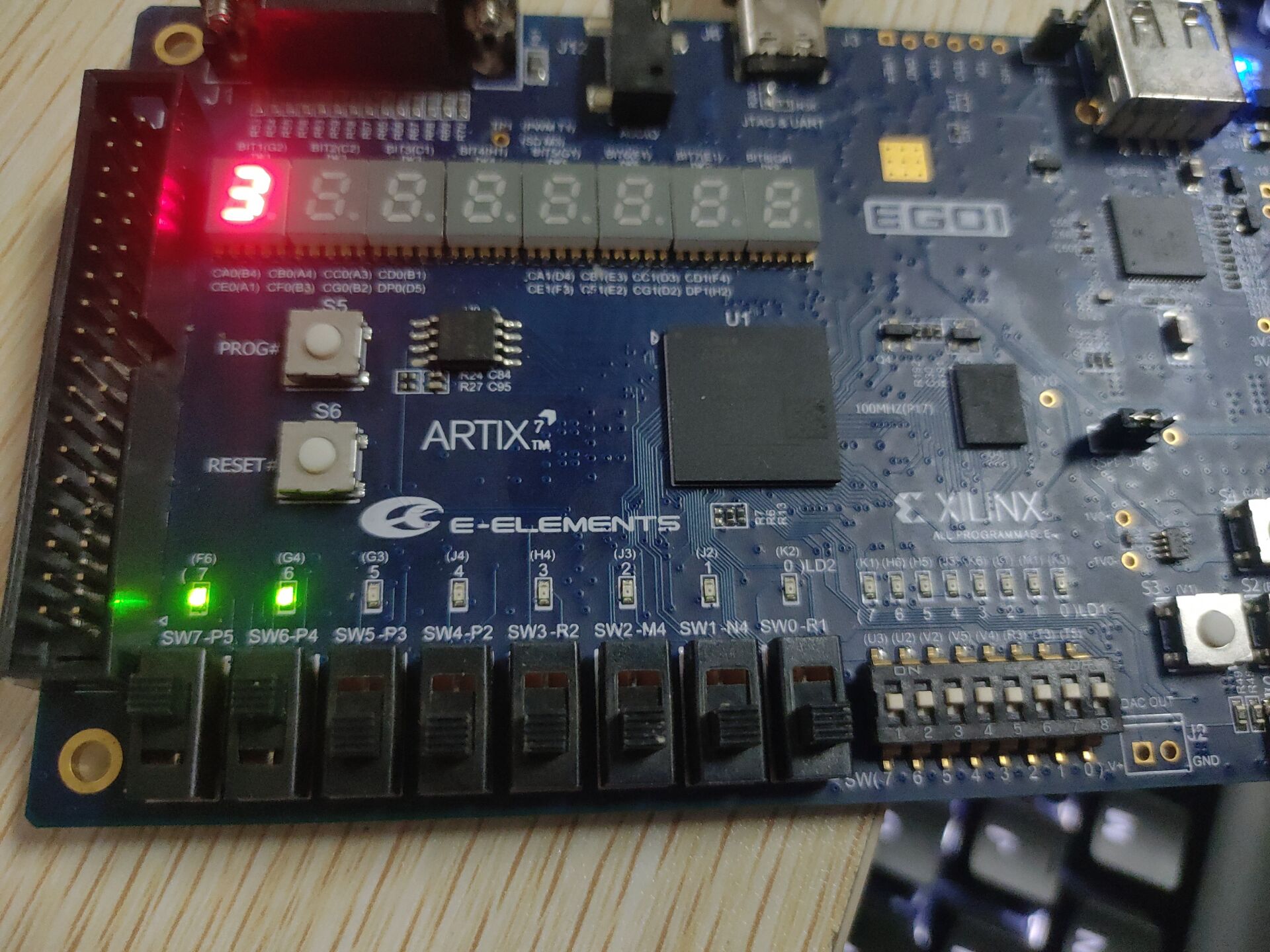
我首先依照半加器的特性编写了半加器 halfadder，然后在编写全加器 的程序中创建了两个半加器并以特定的方式连接。最后我通过这种分层编写的方式实现了一 个功能完备的全加器。 在测试代码中，我给编写的全加器送入不同的 0 或 1 输入，观察全加器的输出以判断全加器是否搭建正确，最后得到的仿真图形与预期一致。

**2位全加器：**









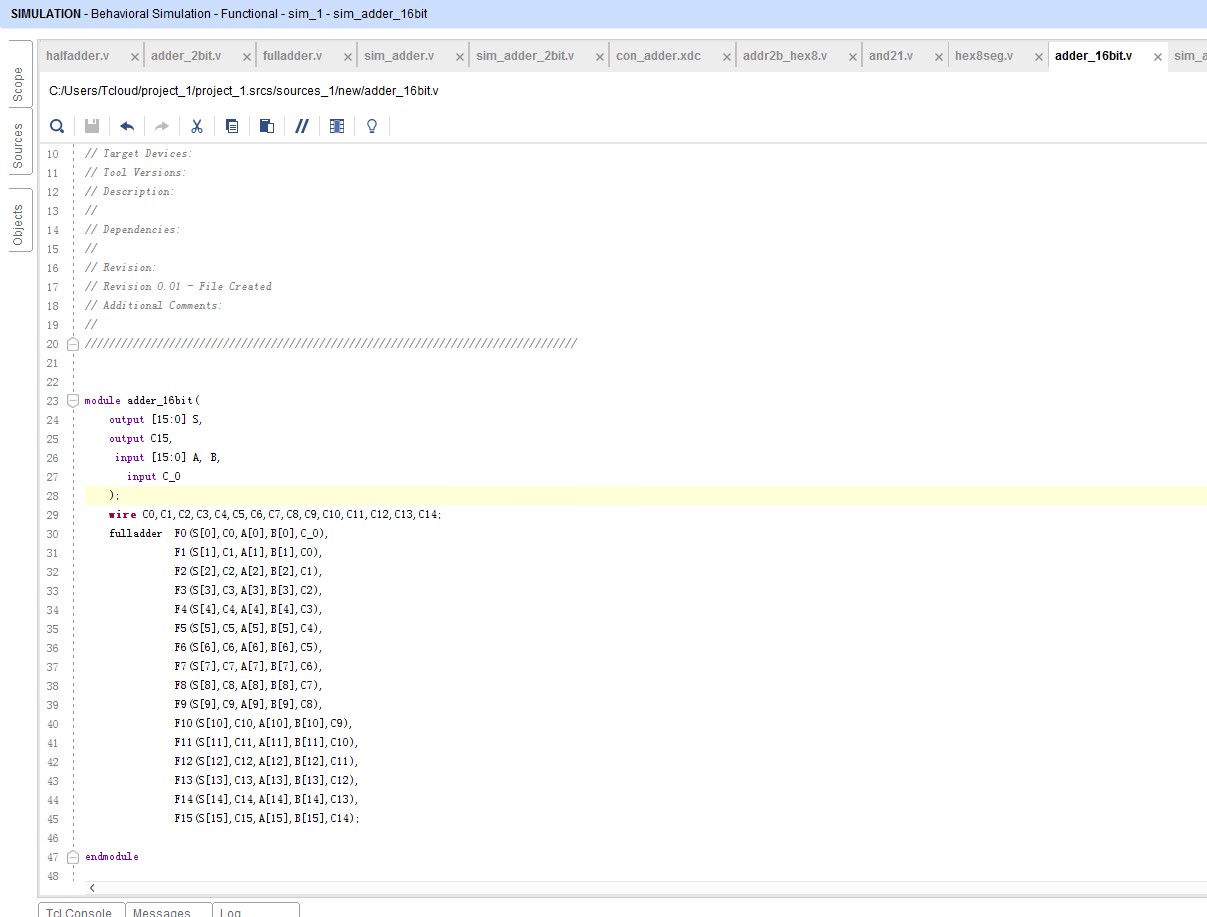
分析：

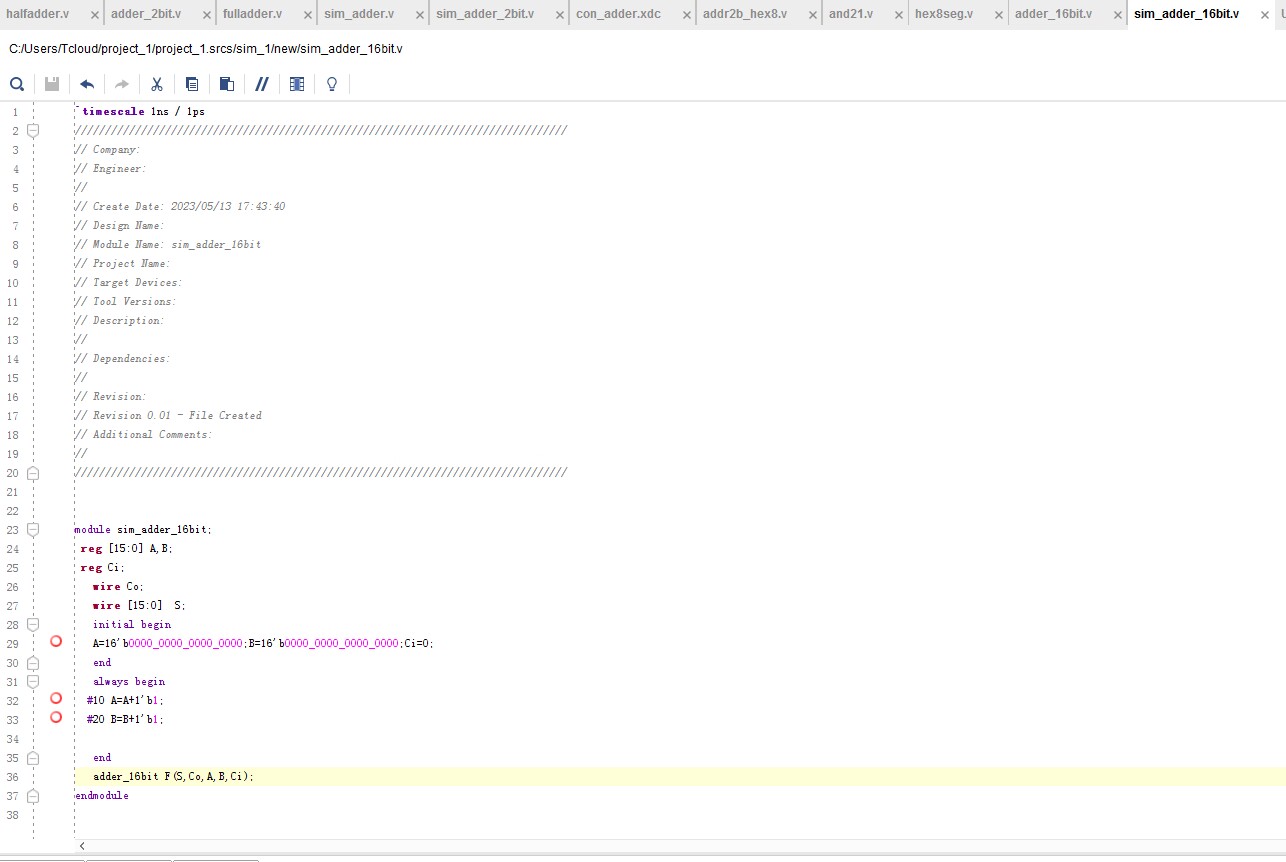
二位全加器可以由一位全加器构成，输入2位二进制数A,B和1位二进制数进位Ci,设置wire类型变量C，让A[0]，B[0]和Ci做为一位全加器的输入，得到的和为S[0],进位为C，再让A[1]，B[1]和C做为一位全加器的输入，得到的和为S[1],进位为最终的进位Co,S[0]和S[1]共同构成最终的和S。

仿真时先将A,B初始化为0，再将Ci初始化为1，随后进入循环，隔10ns后将A进行+1操作，再隔20ns后将B进行+1操作，如此循环。在代码编写中，我创建了两个之前编写好的全加器并使它们的输入端分别接入输入数据 的高位和低位，然后我将这两个全加器的进位端相连，最后我将这两个全加器的输出合成为 2 位输出。这样一个二位全加器就搭建完成了。 在测试代码中，我给编写的二位全加器送入不同的 0 或 1输入，观察全加器的输出以判断全加器是否搭建正确最后得到的仿真图如预期所示。

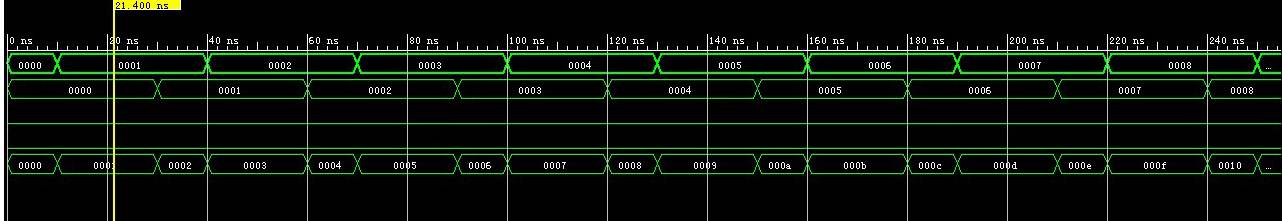
将代码烧录到开发板中，最后示例A为2b’11,B为0，得到的结果3也符合预期结果

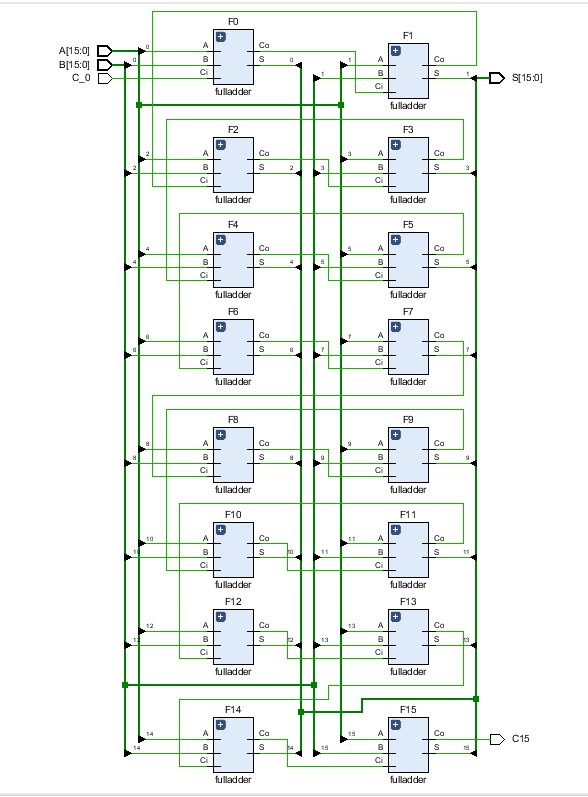
**16位全加器：**











分析：

16位全加器的构建思路与2位全加器的构建思路相同，都是利用两个相加数中处于相同位置的低位做为1位全加器的输入，得到的和做为最终和的低位，得到的进位做为下一次高位相加时的Ci,最终迭代得到S和Co。输入16位二进制数A和B还有1位二进制数Ci,设置若干wire变量Ck(k=0,1,..15),用这些变量存储每一次一位全加器的输出进位，将A[0],B[0]和Ci做为一位全加器的输入，得到的和为S[0],得到的进位为C0，再将A[1],B[1]和C0做为一位全加器的输入，得到的和为S[1],得到的进位为C1,以此类推如此迭代下去，最终A[15]，B[15]和C14做为一位全加器的输入，得到的和为S[15],得到的进位为C15，S[0],S[1],S[2]…S[15]这16个数共同构成最终的和S，C15即为最终的进位Co.

仿真时先将A,B,Ci全部初始化为0，然后进入循环，隔10ns对A进行+1操作，再隔20ns对B进行+1操作，再将各输入量输出量用十六进制表示，按照类似的方法将 16 个全加器并联即可完成编写，最后得到的仿真图如预期所示。

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。**

写完代码后急匆匆进行仿真，结果报错显示仿真文件实例化模块时出错，最后对着设计文件的输入输出相对顺序修改了仿真文件实例化模块时各输入输出的相对位置

在2位全加器烧录过程中误以为约束文件只要找到并修改ppt中展示的那一部分即可，最后发现是只需要保留ppt中展示的那一部分，其余部分全部注释掉或者删掉

心得体会: 在这次的实验中，我体验到了数字电路如何通过层层封装，从最简单的半加器构建出复杂的16位全加器的过程。本次实验我吸取了上次实验的经验和教训，在电路搭建和调试的过程中并未遇到太大的错误。烧录过程要等很久，而且很难调试，出一次错对心态有一定影响，不过最后结果出来后自己很是高兴，看到开发板的所有结果都符合预期很是开心。