实验三 芯片的封装及应用

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74LS161计数器芯片的实现、测试及6进制计数器的实现，将程序下载到FPGA开发板进行验证；

2. 分析电路中的竞争与冒险，给出解决方案并验证；

3. 将芯片及相关模块封装为IP核，通过原理图设计实现N进制计数器，观察、分析仿真波形和电路图；

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）。

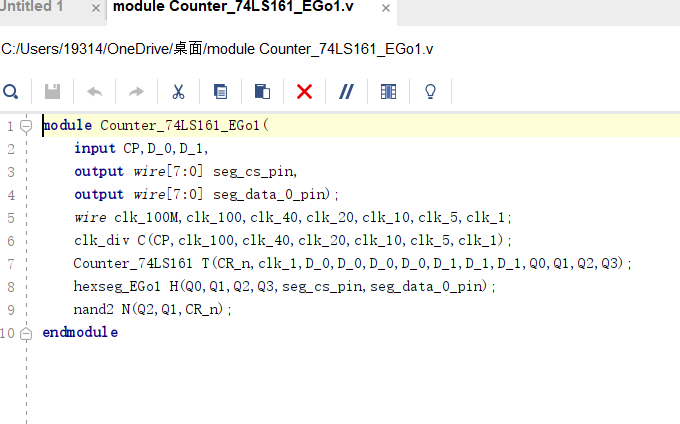
3.记录设计和调试过程。

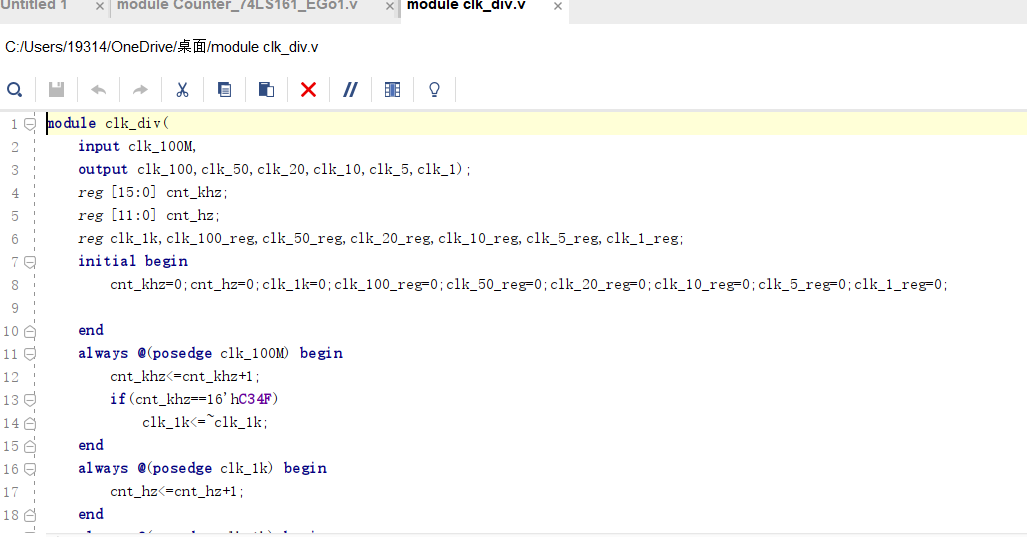
四、实验过程及结果分析

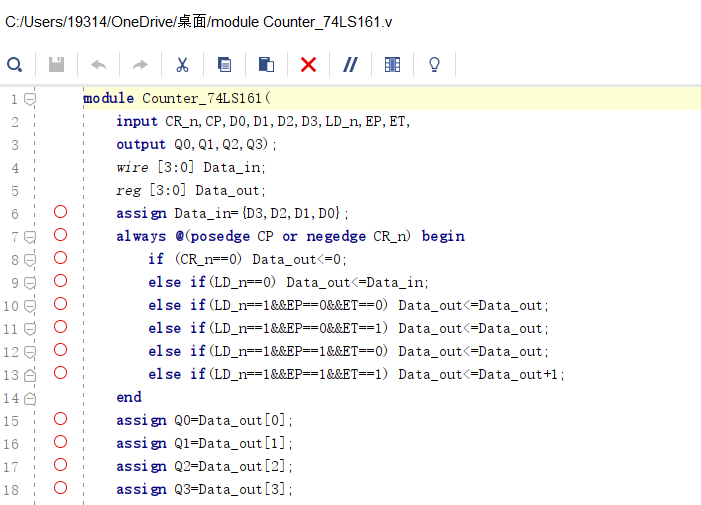
**对电路功能、设计代码、电路图、测试结果等进行简要分析说明，不能只贴截图。**

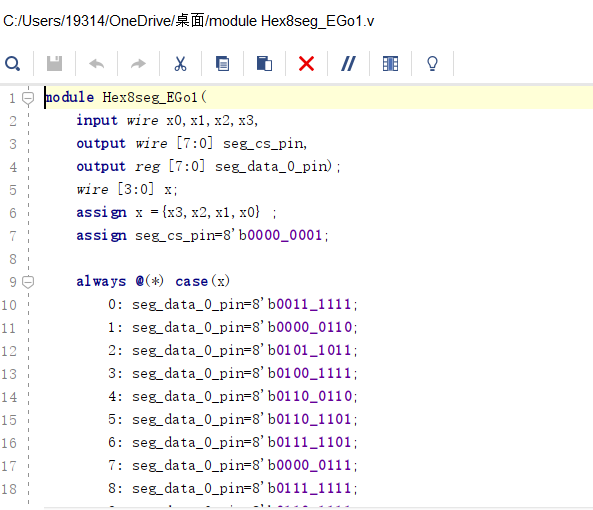
**对于实验内容1（74LS161计数器芯片）：**

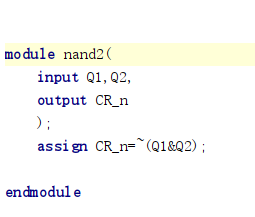
下图为74LS161计数器芯片实现的设计文件：



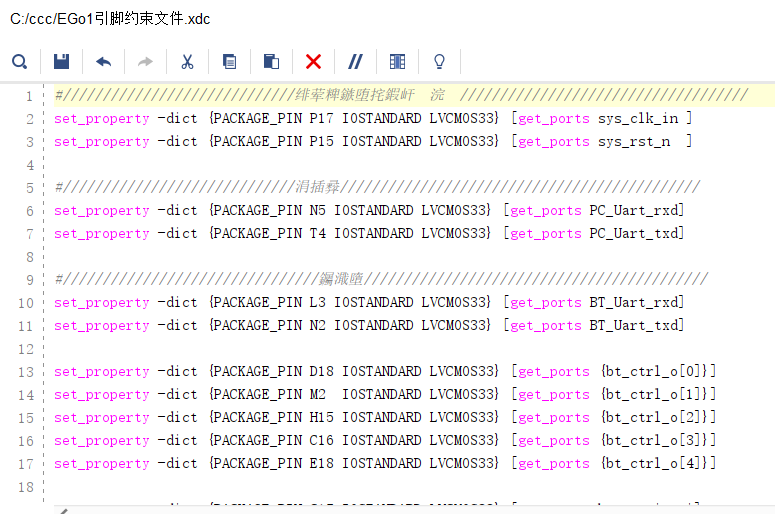




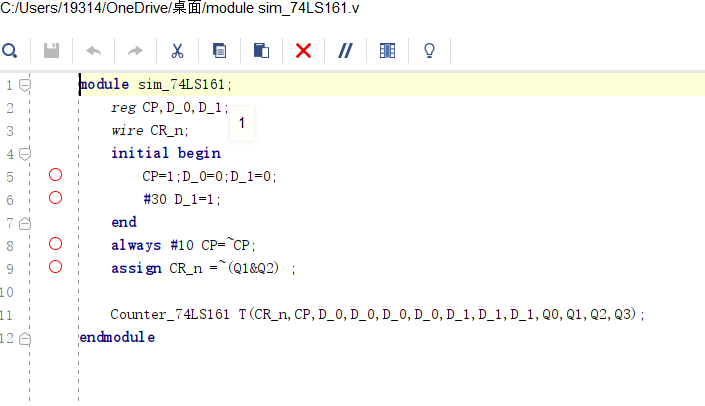




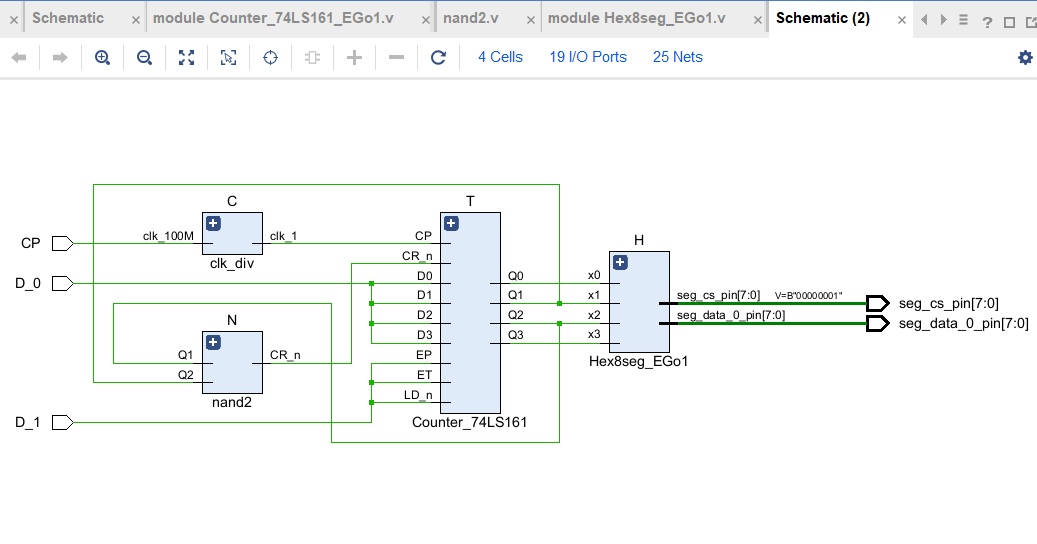
下图为约束文件：



下图为仿真文件：



Vivado生成的逻辑电路图：



电路功能：实现了一个74LS161计数器芯片的基本功能，实现6进制计数器的功能。

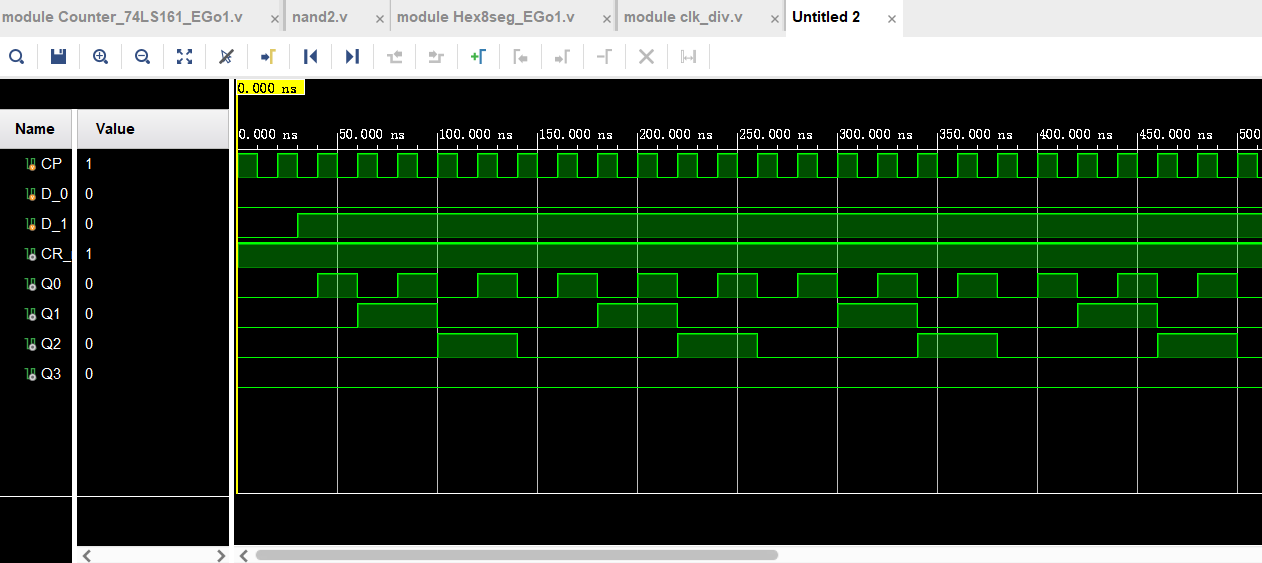
设计文件分析：clk\_div模块是时钟分频模块，用来分频并记录每个输出时钟的状态；

Hex8seg\_EGo1模块是用来实现数码管显示的模块；

Counter\_74LS161模块是用来实现74LS161主要功能的模块；

Counter\_74LS161EGO1是上层文件，将所有模块拼接用来实现完整的功能。

仿真结果图：



仿真代码分析：

先定义reg类型变量CP和D\_0,D\_1,然后初始化CP=1，D\_0和D\_1为0；

延迟30个时间单位后，将D\_1设为1；

之后每延迟10个时间单位，翻转一次CP；

取CR\_n为Q1和Q2的与非；

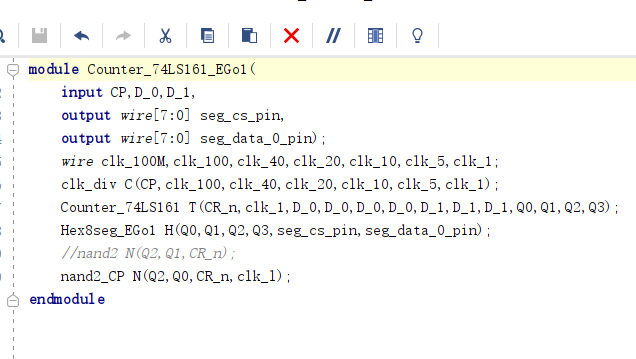
应用Counter\_74LS161检测输出结果。

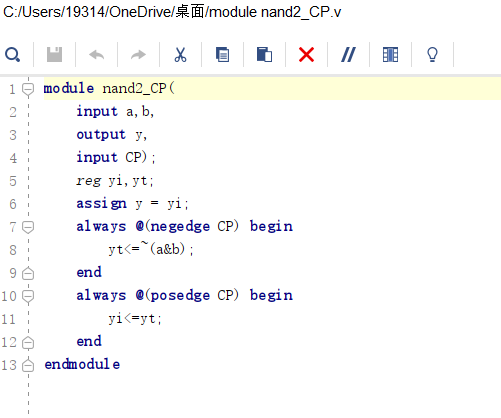
仿真结果/波形图分析：从上往下每行依次是 CP,D\_0,D\_1,CR\_n,Q0,Q1,Q2,Q3的波形图。可以发现，输入信号的波形图都满足上面的分析，同时，CR\_n也满足74LS61的计算规则规则。 显然满足预定的功能。

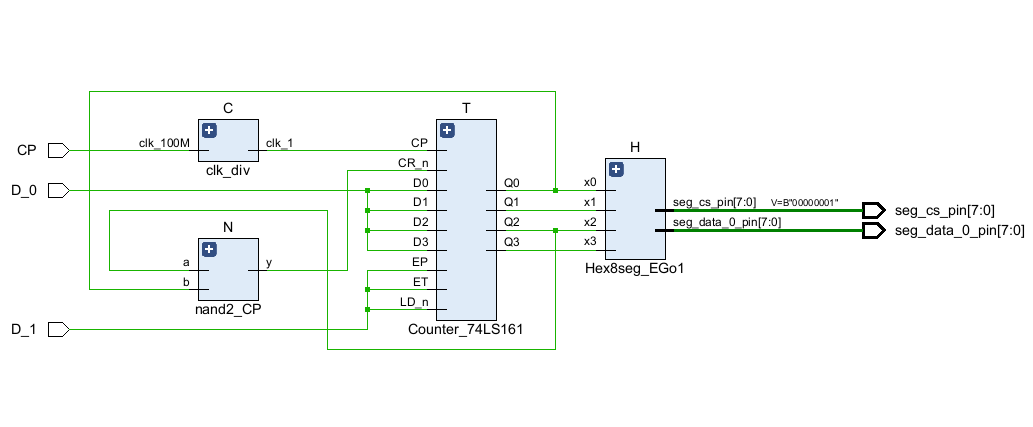
**对于实验内容1（6进制计数器）：**

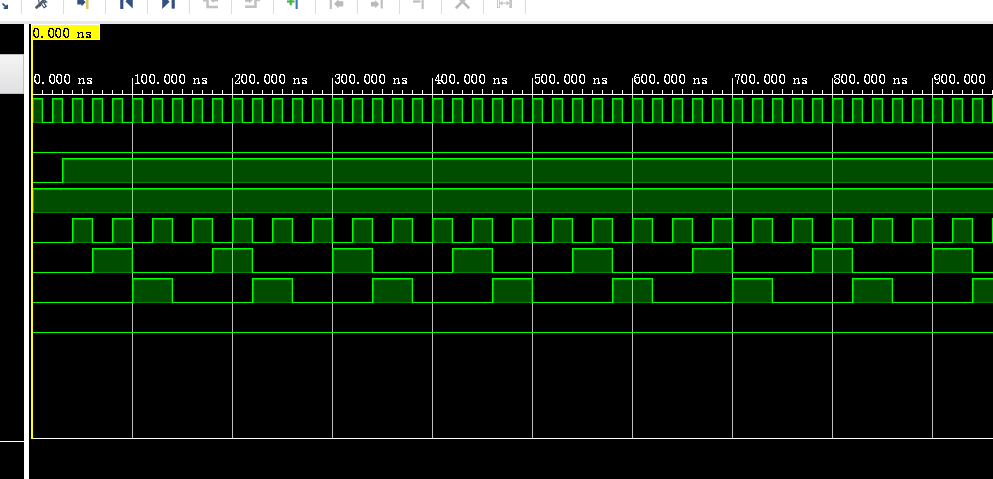
只要对原文件稍作修改即可。

以下是修改过的文件：

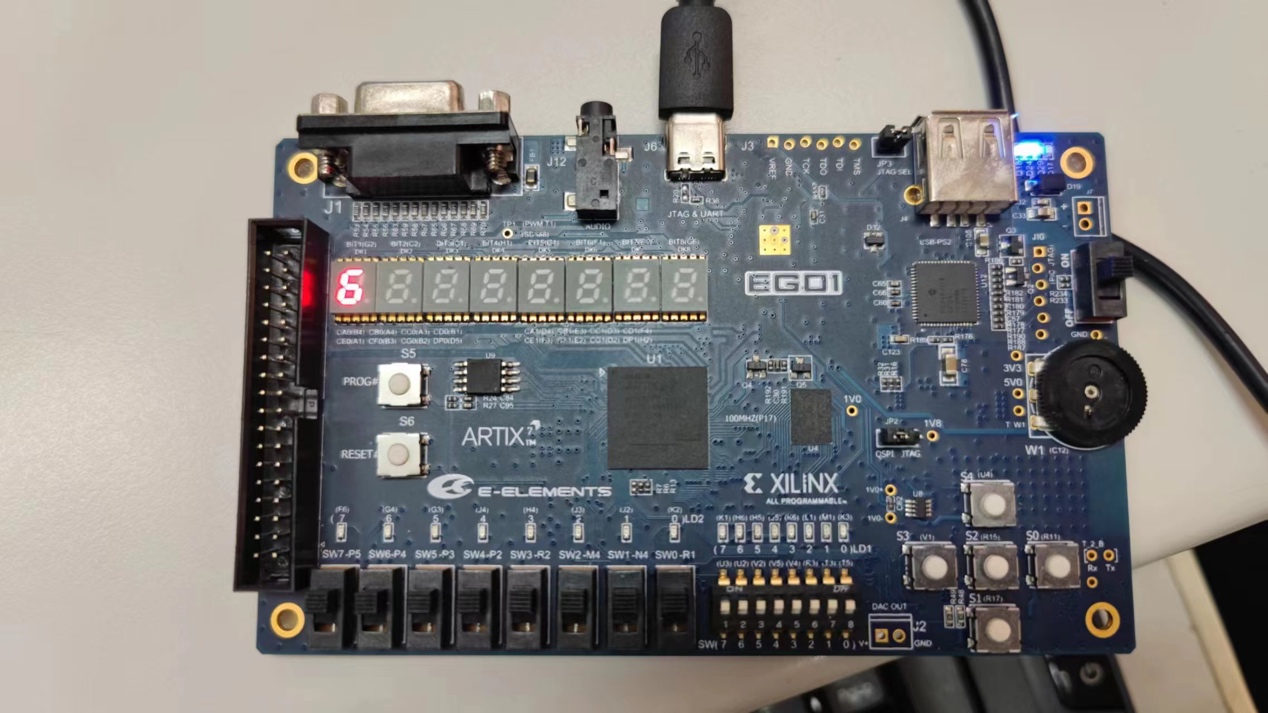








**烧录到开发板进行的验证：**

****

**对于实验内容2：**

通过对输出结果的检测，并通过与非门进行判断（只有在到达计数为6的时候才会出现中间两位都是1，检测到中间两位都是1后，通过与非门输出低电平，传入清零端，完成一次循环），可以调控74LS161的输出顺序，使得它成为一个六进制计数器。

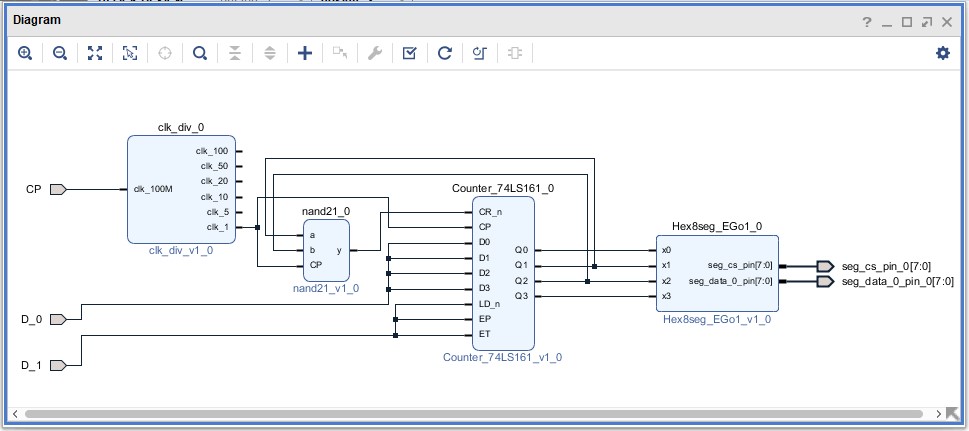
但是，由于不同路径上的信号通过多级电路可能会有延迟，也就是出现了竞争，导致输出端出现一个我们不想要的结果，也就是冒险。

在这个逻辑图中，由于通过电路门需要时间，可能导致信号到达T模块的时间出现偏差，从而导致了一个不正确的结果。

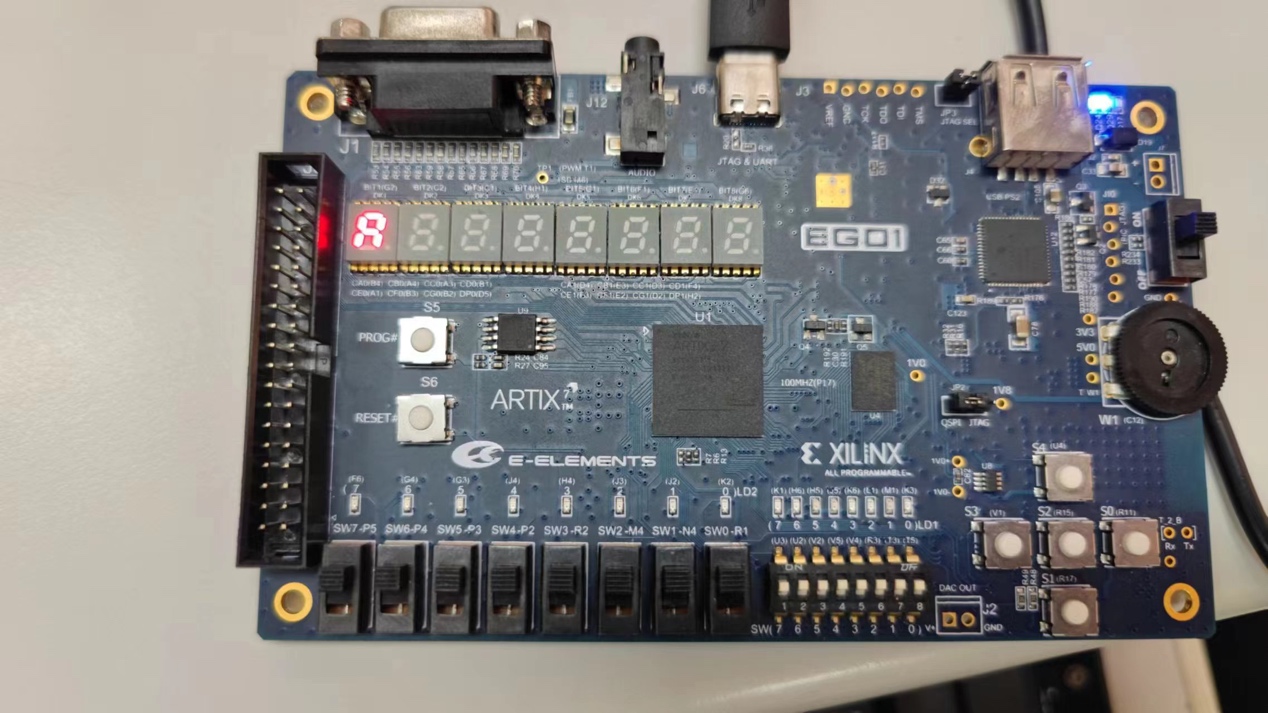
可能的解决方案是，可以对从与非门到返回清零信号的部分电路进行修改，比如将同步的时钟信号也作为输入，只有时钟信号有效的时候，才会清零或者继续循环，从而解决冒险与竞争的问题。

**对于实验内容3（这里选择实现10进制计数器）：**

**逻辑电路图如下：**

****

**烧录到开发板的验证：**

****

电路功能：实现了一个74LS161计数器芯片的基本功能，并利用74LS161实现10进制计数器的功能。

设计文件分析：其设计文件和6进制计数器的设计文件几乎一样，不再展示了。这里只是将6进制文件中，检测输出的与非门和置零的触发条件做了修改，当检测到10的时候触发与非门的清零条件，重置并完成1次循环。

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。**

1. 在调试生成schematic的时候，在修改源文件后，出现一直报错的问题，最后发现需要先关闭后台的仿真程序，然后reload源文件，再进行RTL操作，就得到了正确的图像。
2. 完成本次实验后，我对74LS161芯片和任意模计数器的原理和实现有了更深的认识。