**数字电子技术基础实验实验报告三**

1. **实验步骤：**

1．建立工程文件，编写模块源码和测试模块,要求测试模块能对源文件进行比较全面的测试。

2．编译源码和测试模块，用测试模块对源文件进行测试，并综合仿真。得到波形图。

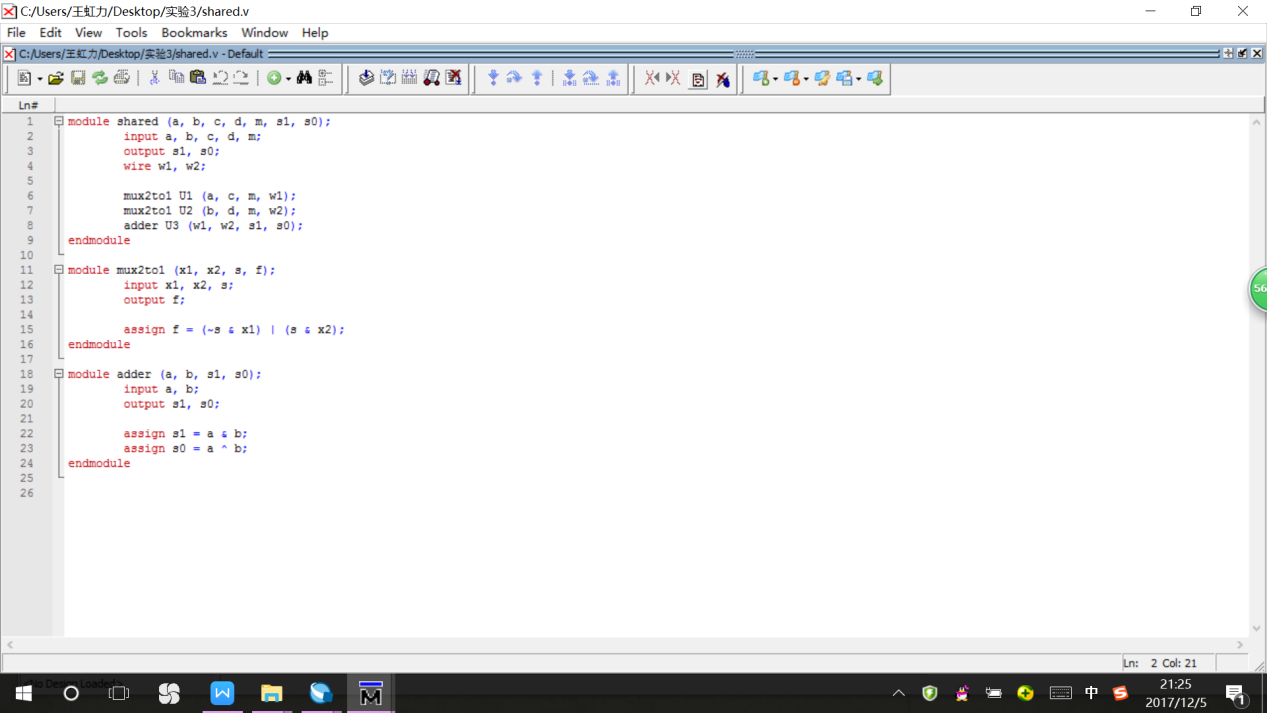
3．记录数据并完成实验报告。

4．使用QuartusII软件的RTL viewer工具查看/确认所设计的电路。

**二、Quartus II软件基本使用步骤：**

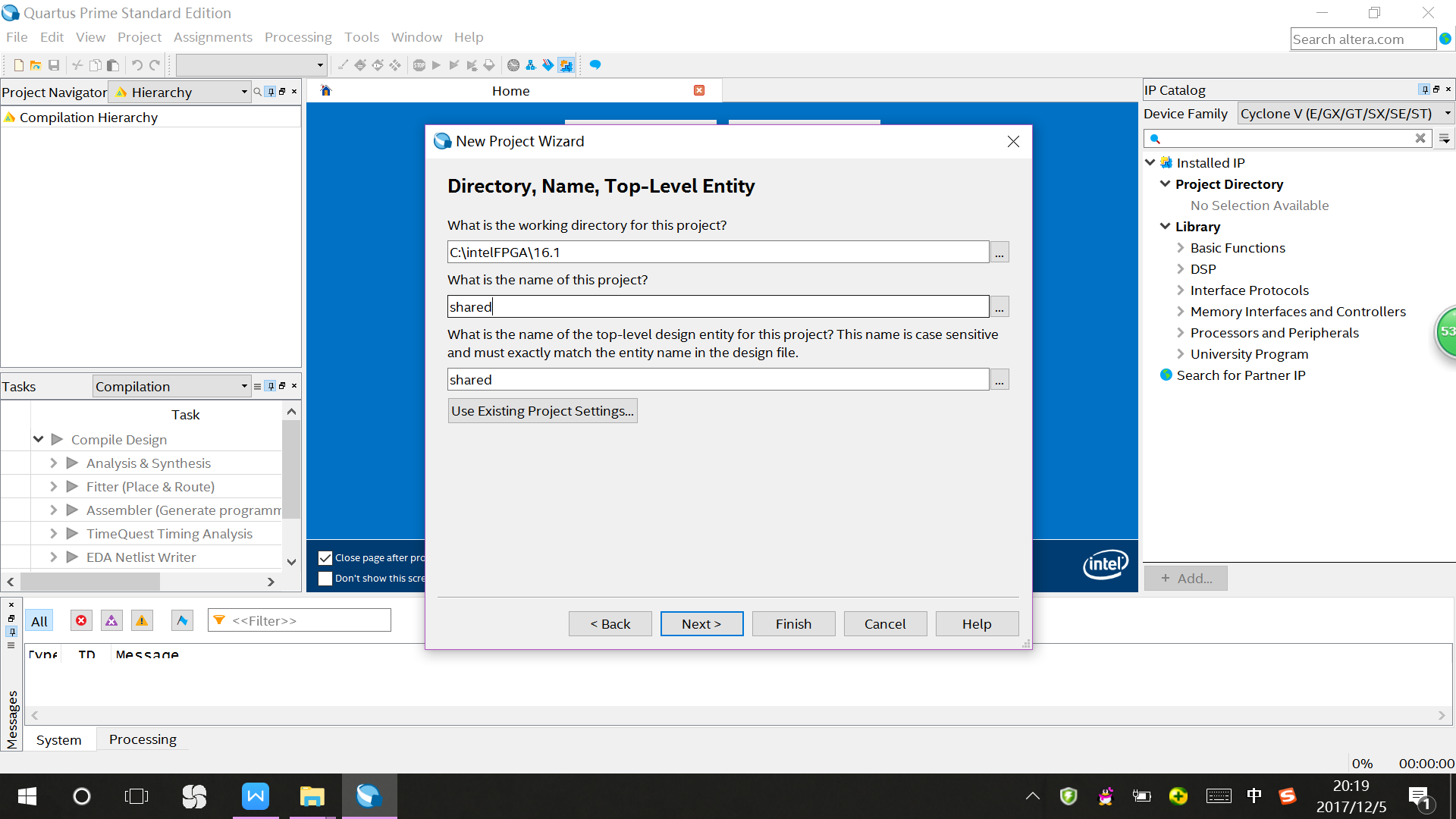
1、编码

在实例中代码已经提供，并且是正确的，我们只需对其进行modelsim仿真确认该电路设计正确。



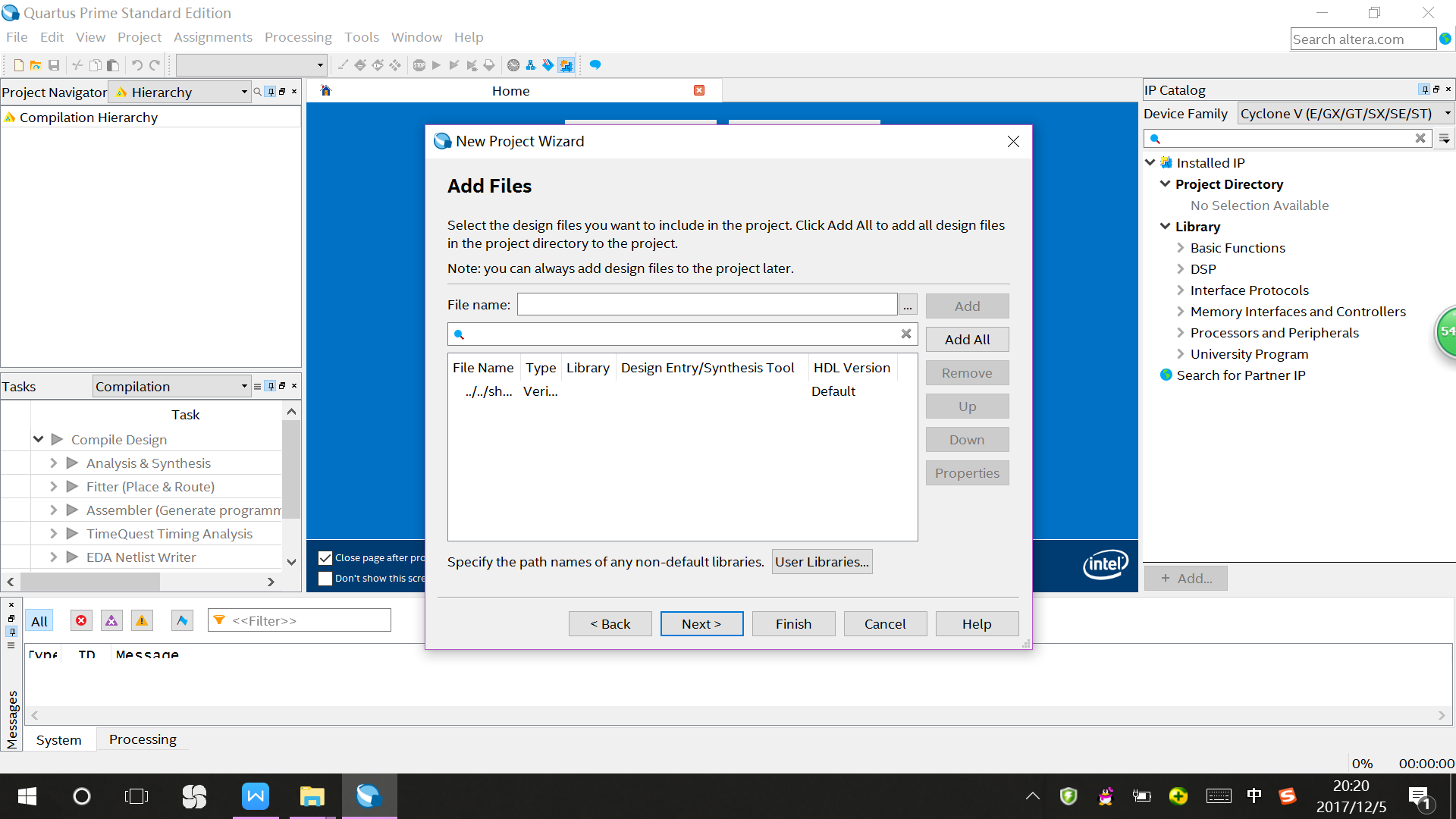
2、建立工程

首先建立一个新的工程，点击开始页面的new project wizard或者File->new project wizard。然后点击next，选择工程目录，在给工程命名的时候，需要注意的是名称要和Verilog代码中的一致，这里提供的典例代码名称是shared，则将工程的名字命名为shared。



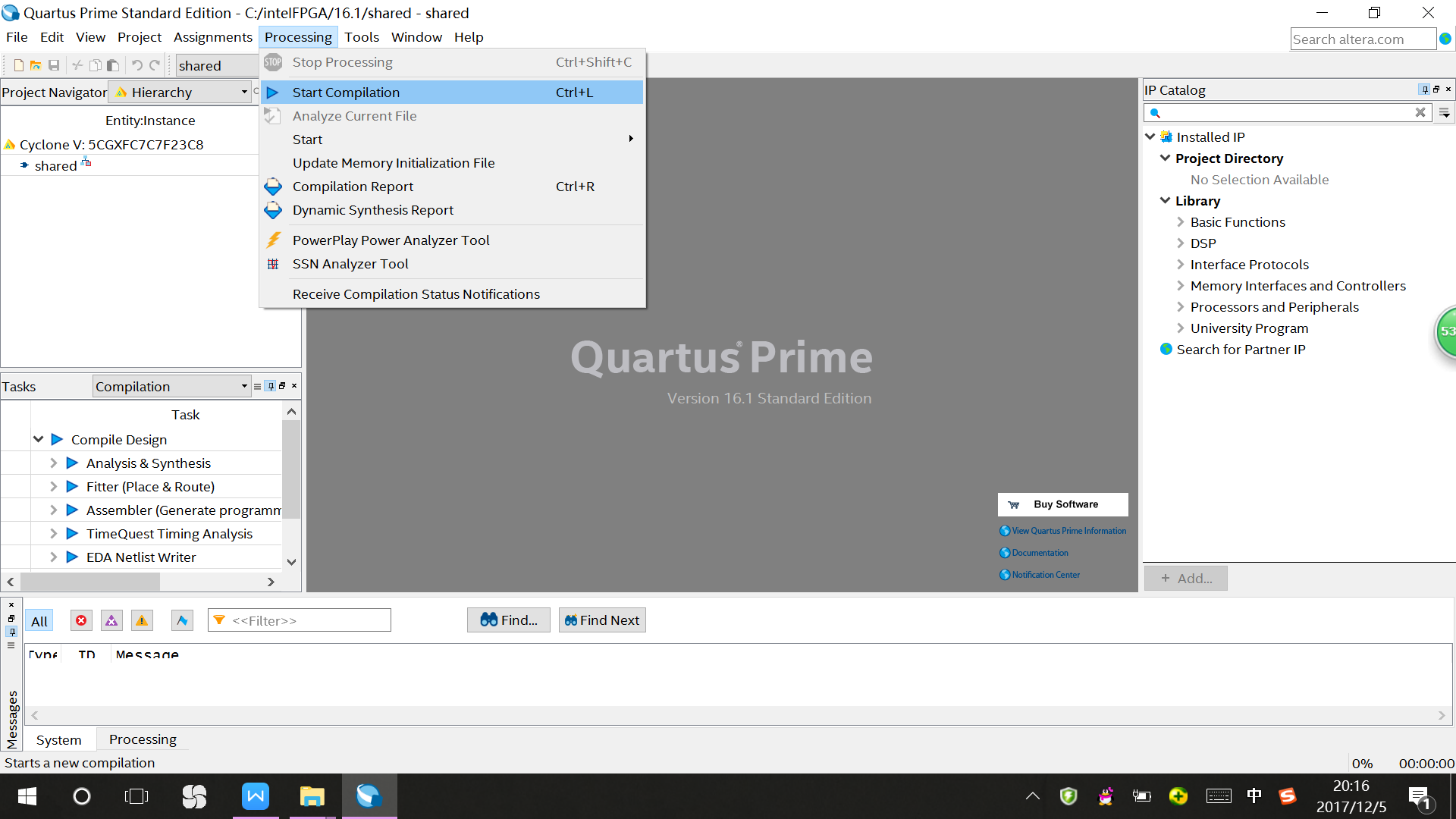
3、添加文件

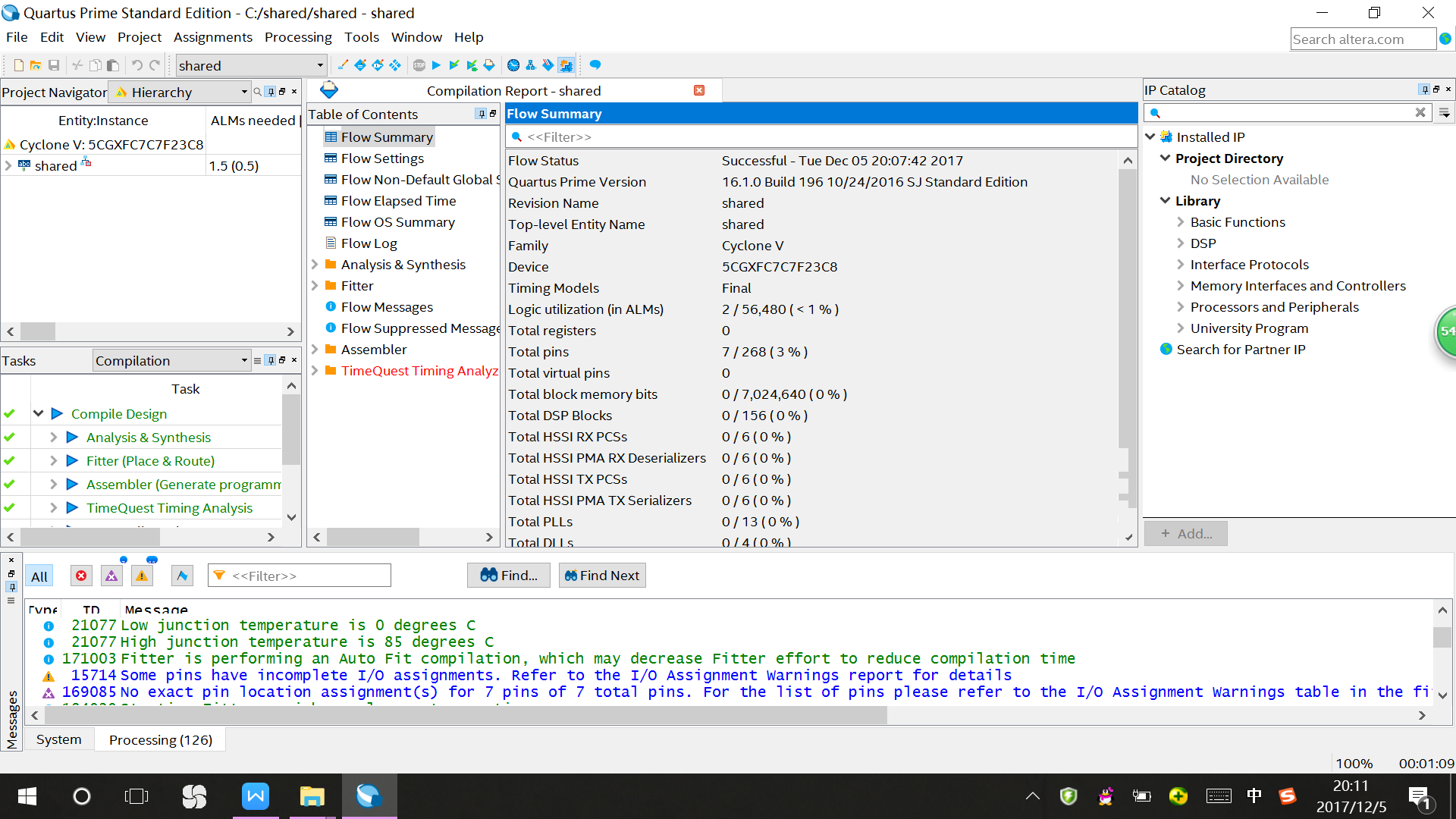
在空的工程中将已预先放置在一个文件夹中写好的正确代码shared.v，添加完毕之后，点击finish完成创建。



4、编译

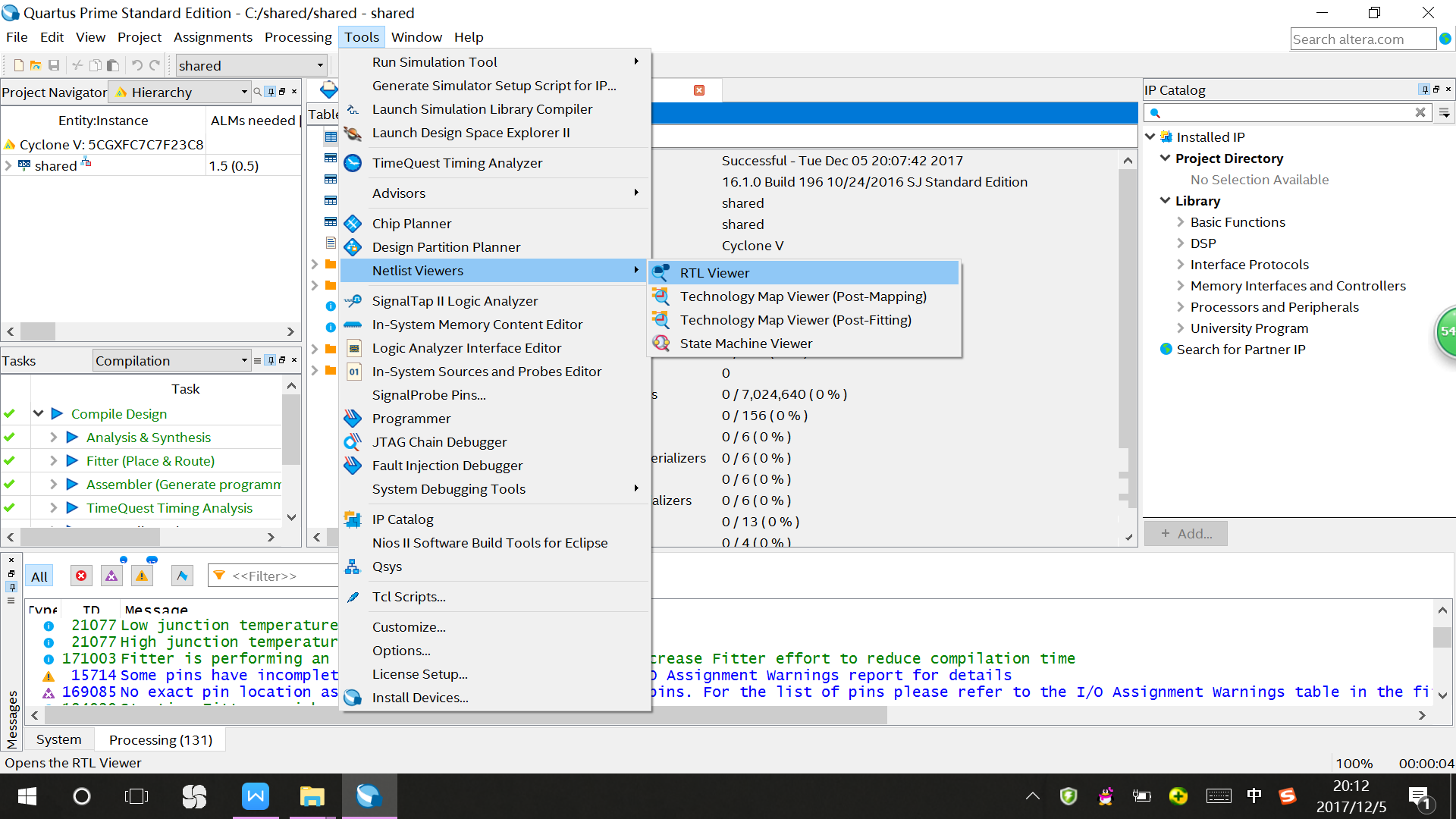
之后我们对添加文件的工程进行Start Compilation ,即编译源文件 (如有错误修改后,重新编译)。点击processing->start compilation，然后等待即可。显示编译正确之后即可进行下一步。



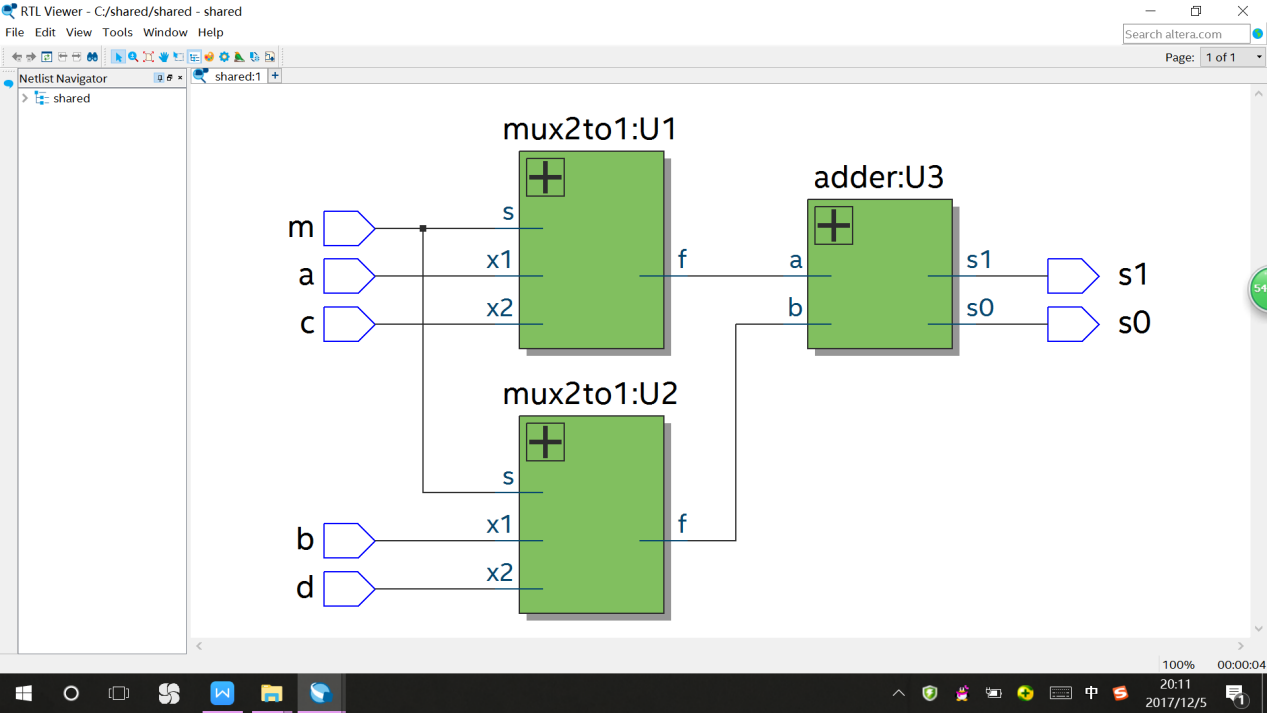


5、查看电路结构

查看电路图结构,是否和预期设计一致。点击Tools-> Netlist Viewers > RTL Viewer，即可查看对应工程中文件代码的电路结构。



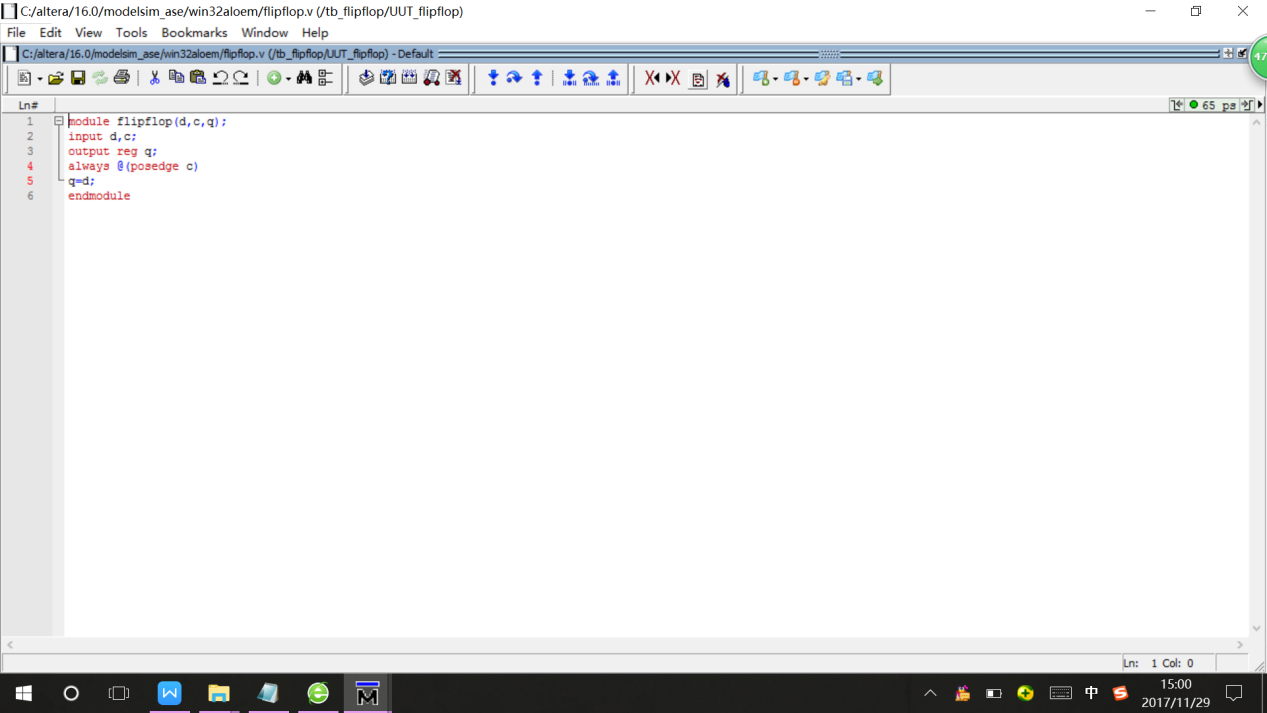
6、shared.v的电路结构如下



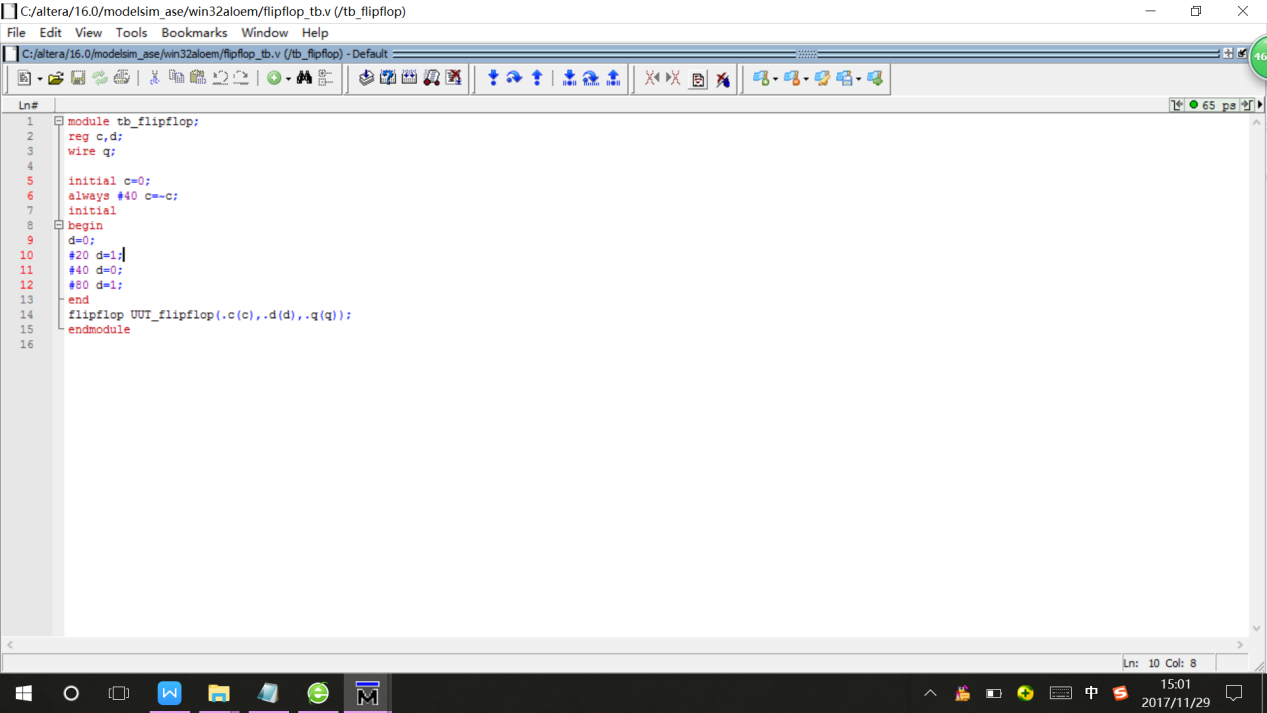
**三、实验内容：**

**1、设计一款时钟上升沿触发的D寄存器,并编写testbench**

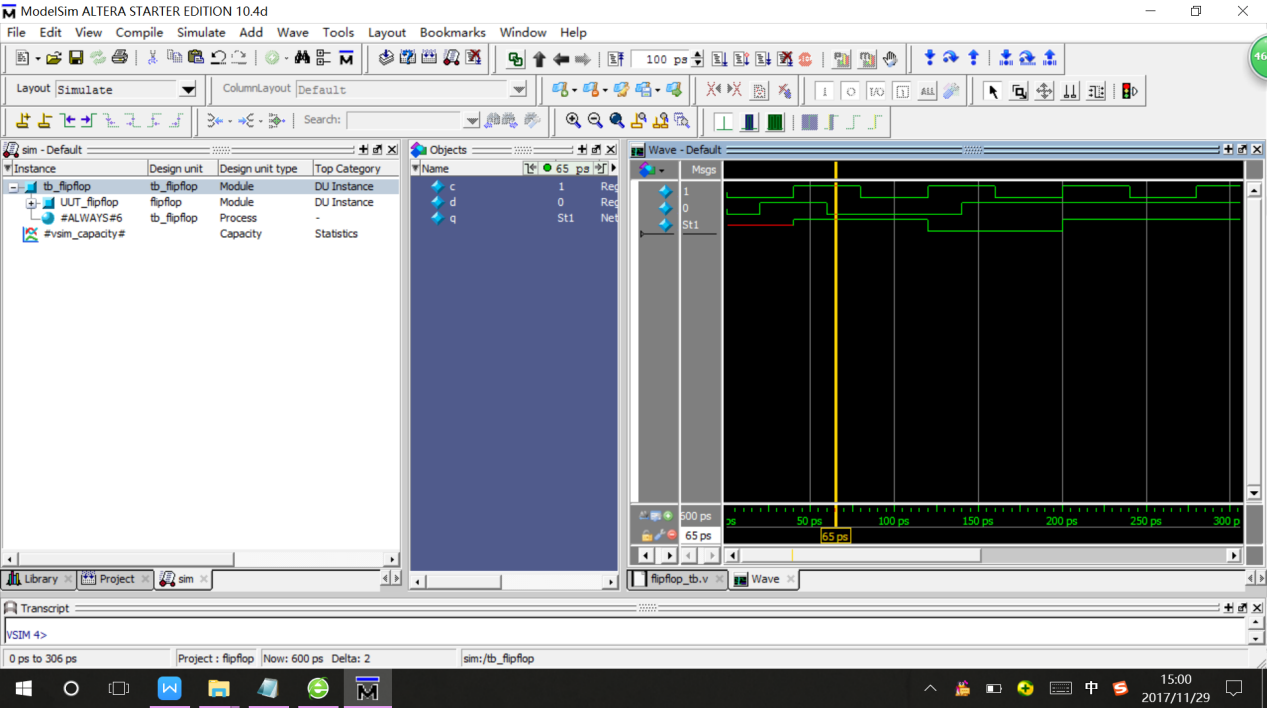
（1）flipflop的源代码：



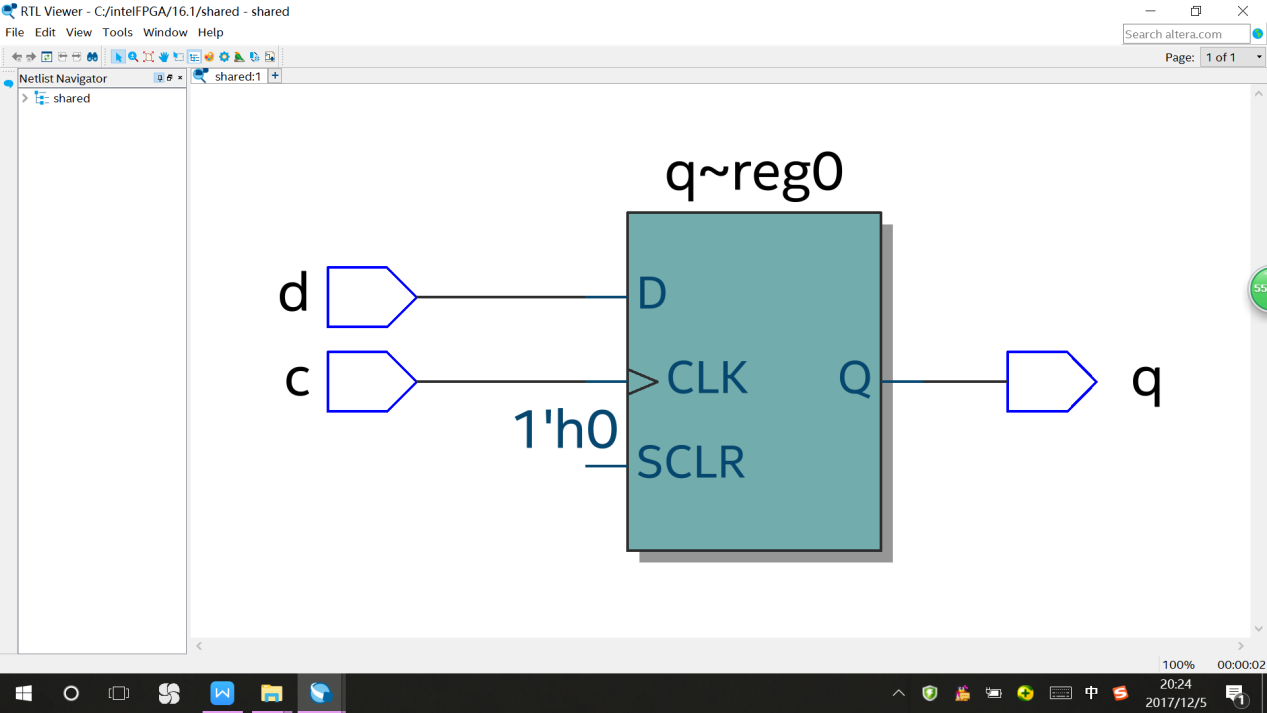
1. flipflop的测试代码：



（3）flipflop的wave图：



（4）flipflop的RTL viewer：

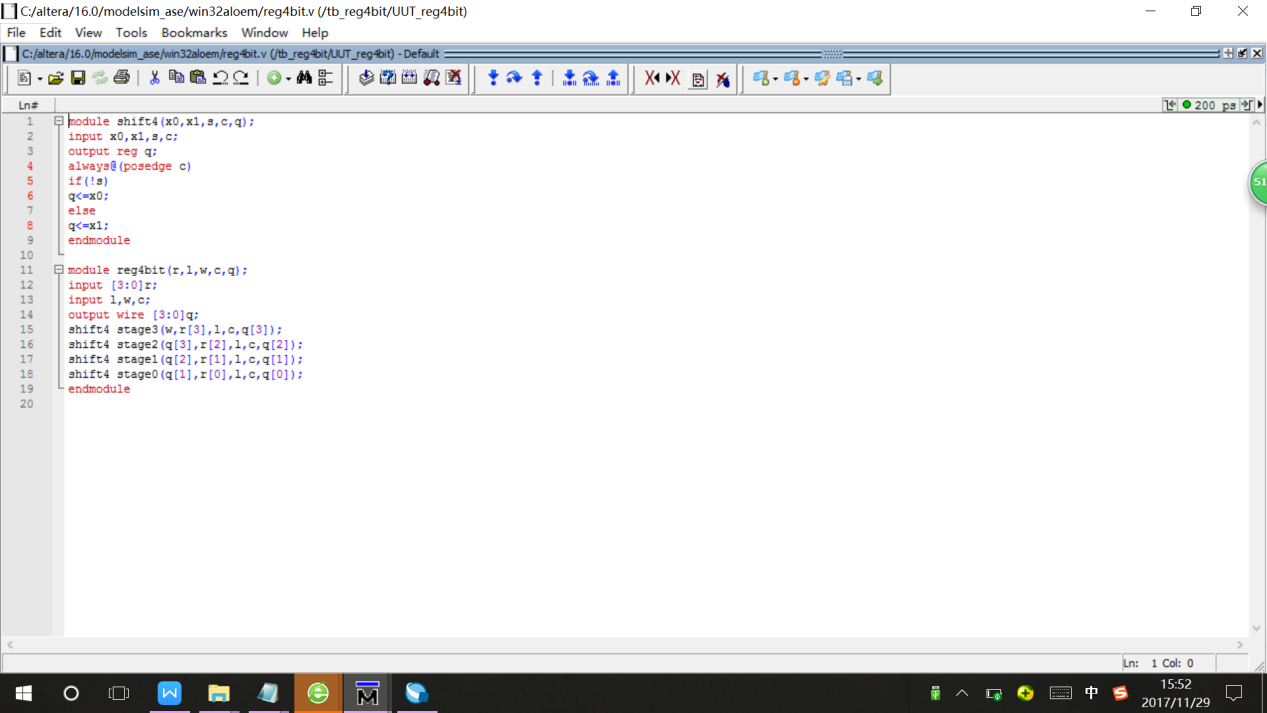


1. 简要分析：

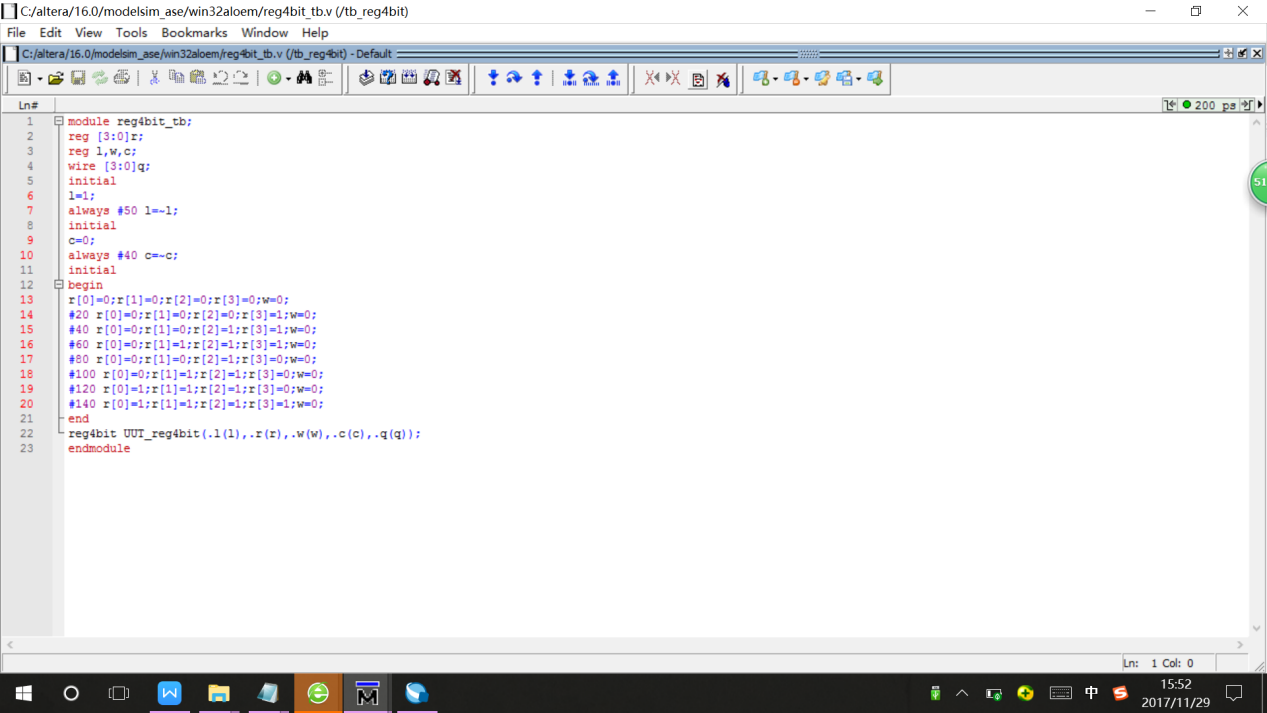
即实现当处于c上升沿且d为0的时候，输出量q突变为0，并且c处于其他状态时q不变；当处于c上升沿且d为1的时候，输出量突变为1，并且c处于其他状态时q不变。

由wave图可分析得，如黄线所示：当c=1,d=0,q=1,符合时钟上升沿d触发器的工作效果。

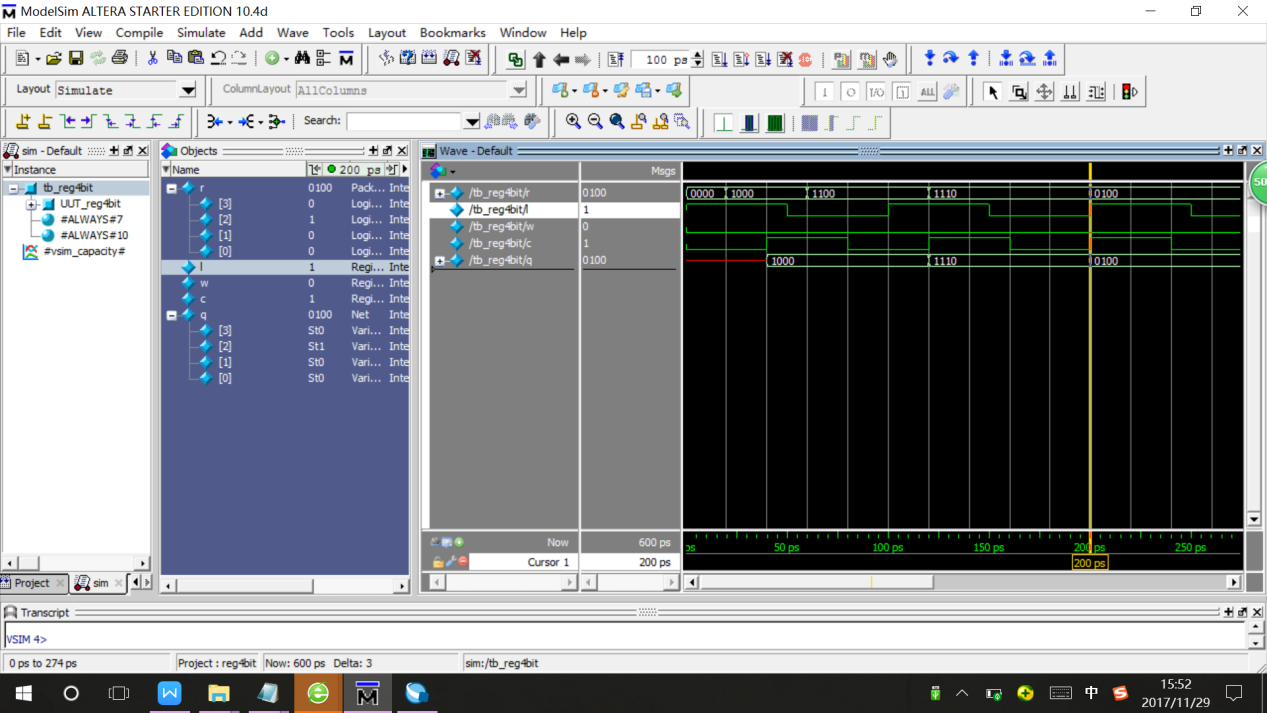
1. **设计一款4bBit具有并行加载功能的移位寄存器,并编写testbench。**
2. reg4bit的源代码：



1. reg4bit的测试代码：



1. reg4bit的wave图：



1. reg4bit的RTL viewer图：



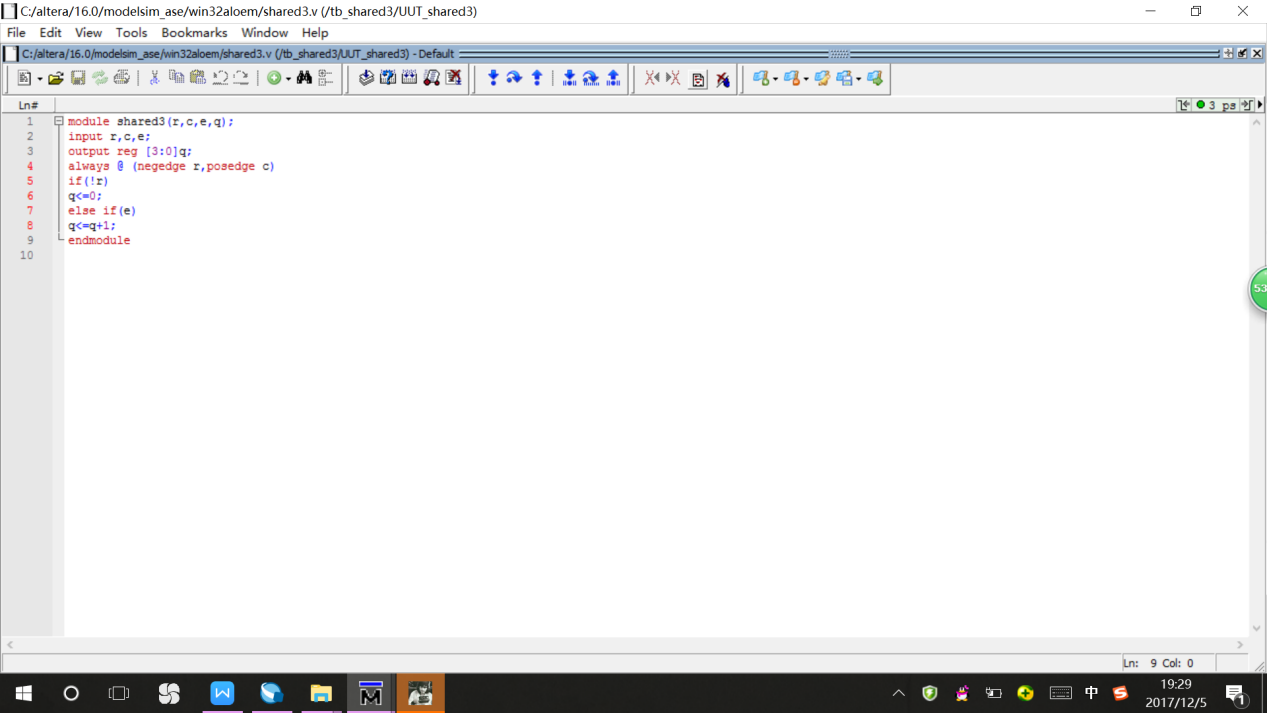
1. 简要分析：

实验中，当选择信号s为0时输入x0，寄存器处于Shift阶段，并且时钟上升沿时，进行移位操作；当选择信号s为1时输入x1，寄存器处于Load阶段，直接读入并行输入的4个数字。通过每一个多路选择器的输出量d连接到触发器的数据输入端，4组装置的级联同时，实现了并行加载和移位的功能。

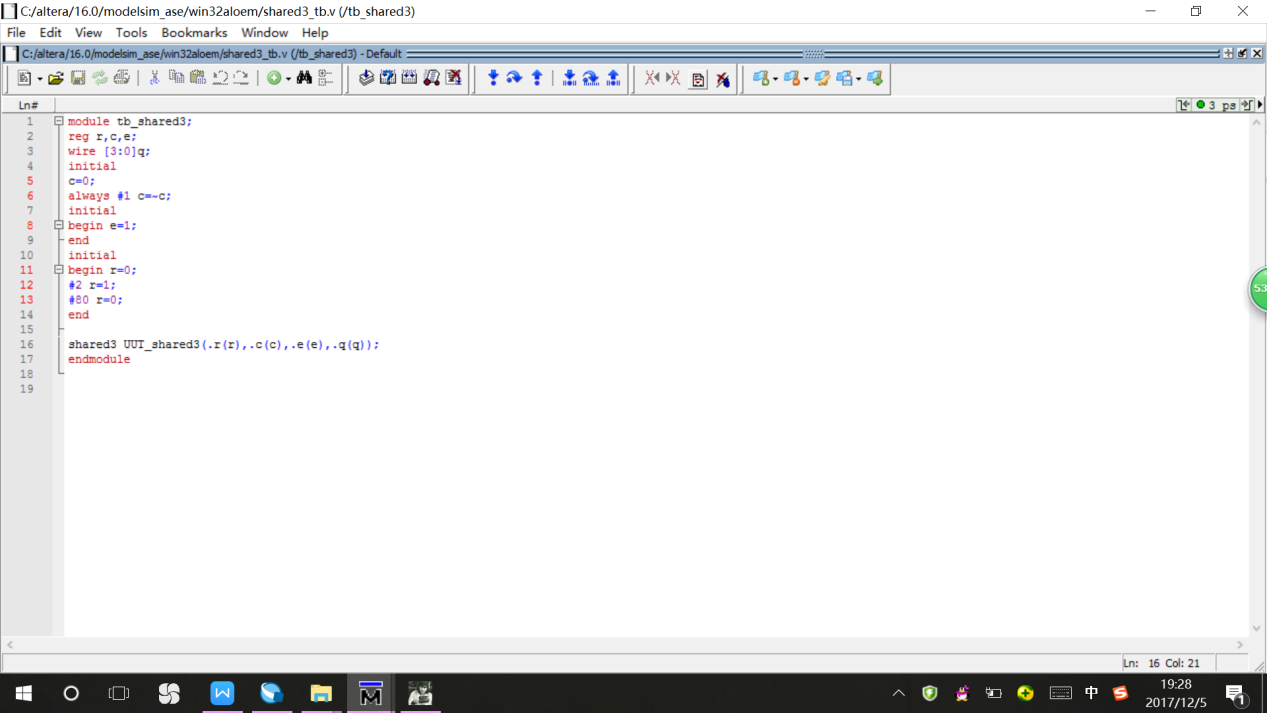
对波形图进行分析可知，实验结果完全符合4 bit具有并行加载功能的移位寄存器的工作要求。

**3、设计一款4 bit带复位功能的计数器,并编写test bench和仿真波形,请包括以下测试点:电路是否能复位，是否能按预期增加计数，是否按预定溢出。**

（1）shared3的源代码：

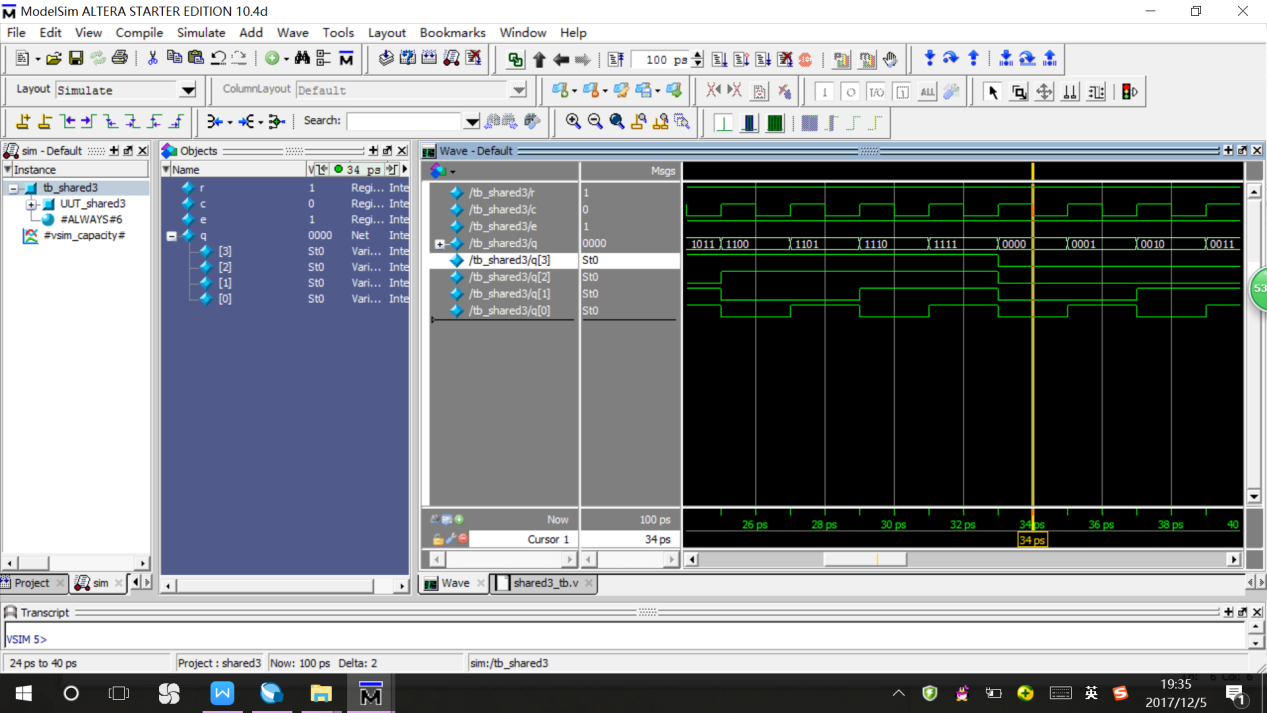


（2）shared3的测试代码：

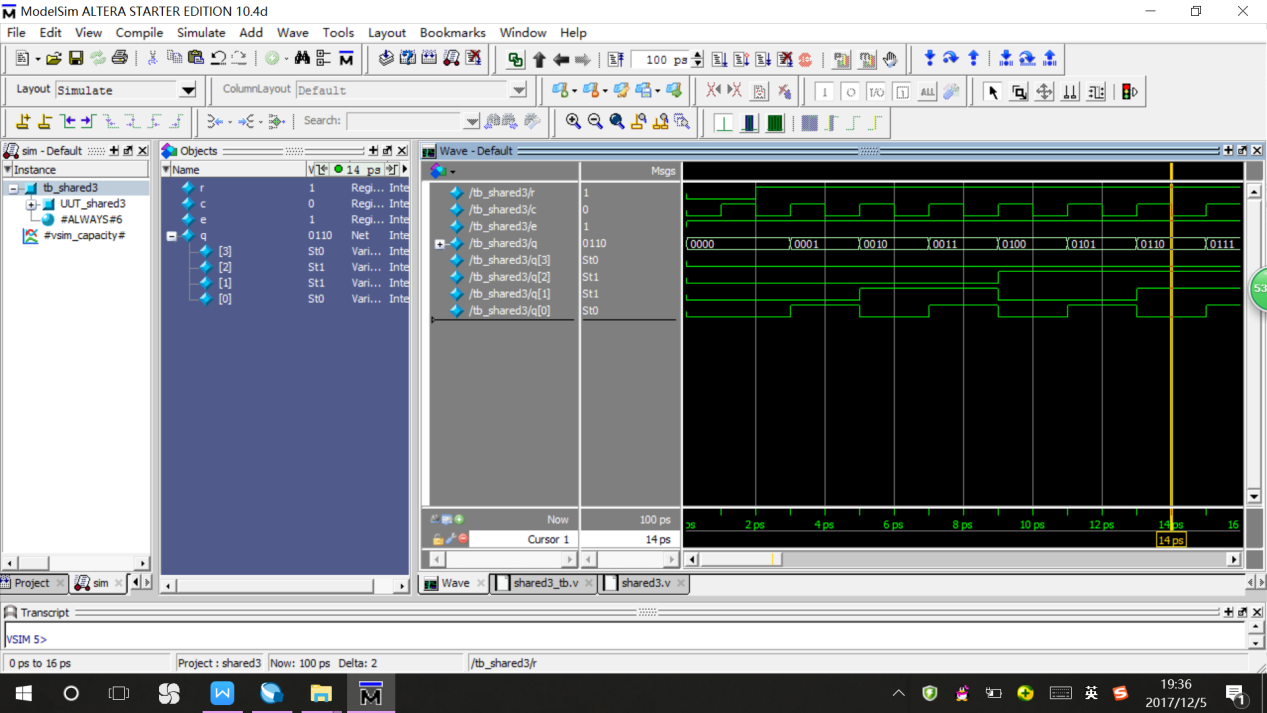


1. shared3的wave图即测试点：

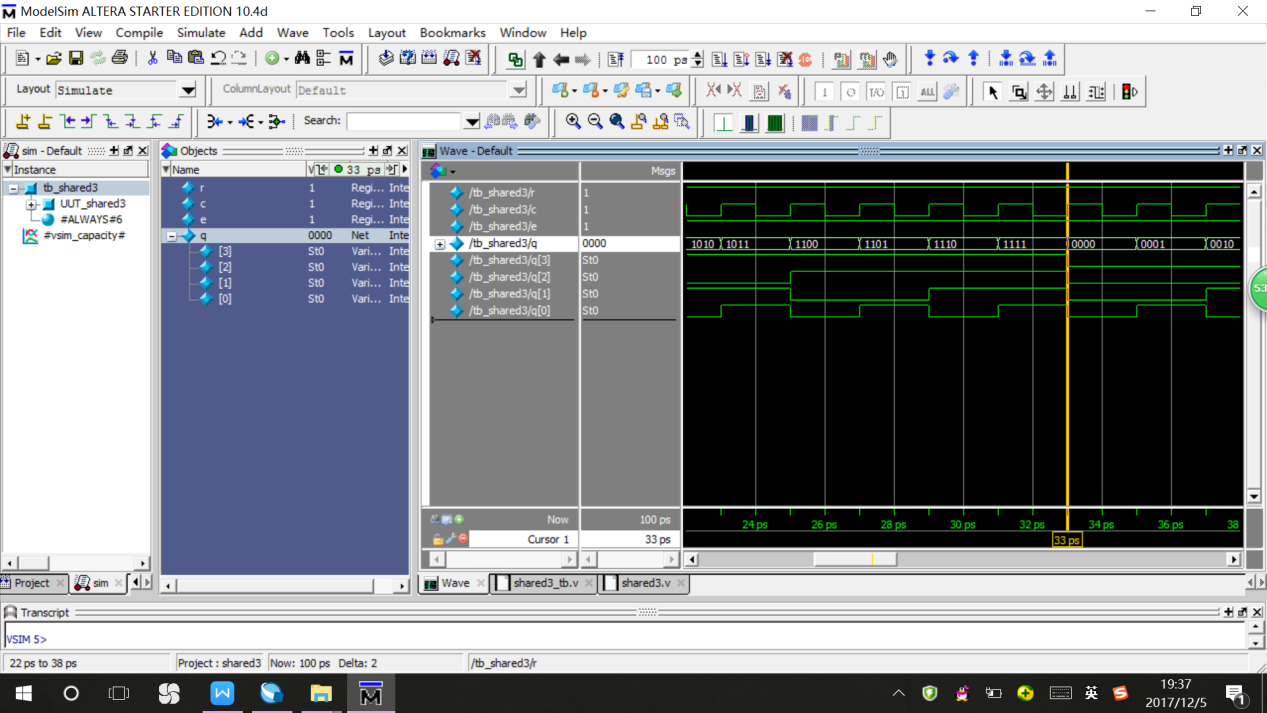
•复位功能：



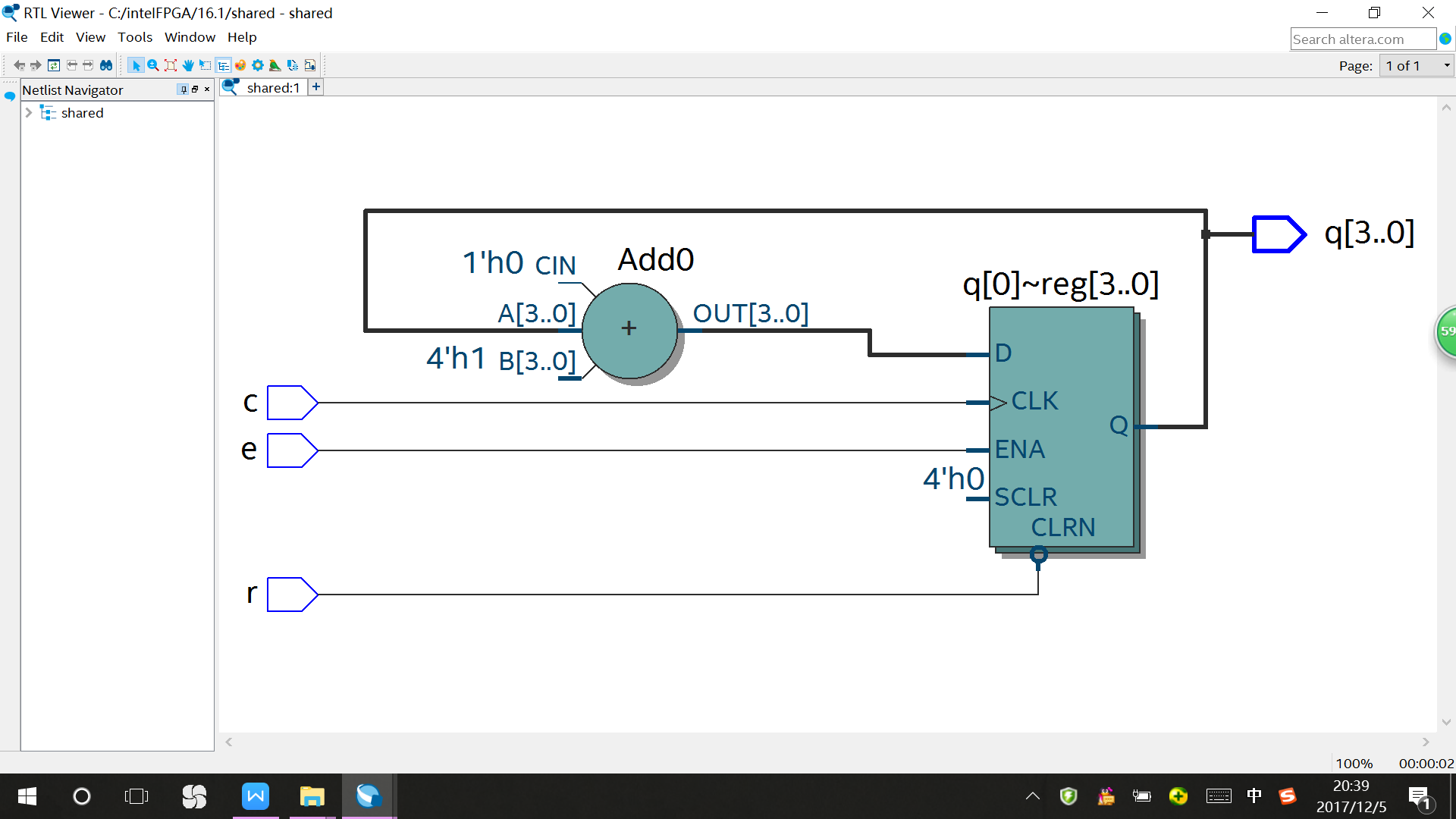
•预定计数功能：



•预定溢出功能：



（4）shared3的RTL viewer图：



1. 简要分析：

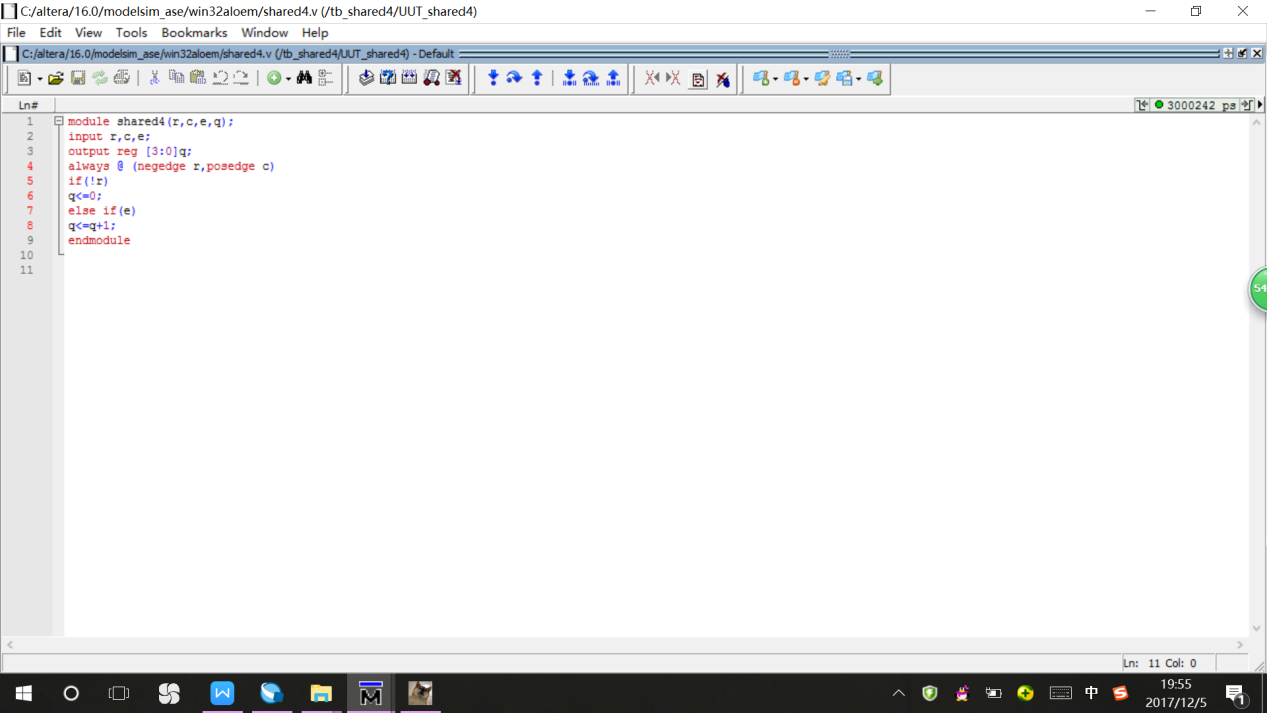
题目要求设计带有复位功能的计数器，故此根据书中计数器代码部分又重新加入了复位信号r和使能信号e。用4位数组q表示计数器的输出结果。通过使用if-else语句，实现了对判断进行复位操作还是计数器工作，即仅当e=1，计数器才会在时钟信号c的上升沿工作。

测试代码中设置了时钟信号每1单位取反，又利用了r信号在80单位时间后测试复位功能是否良好。

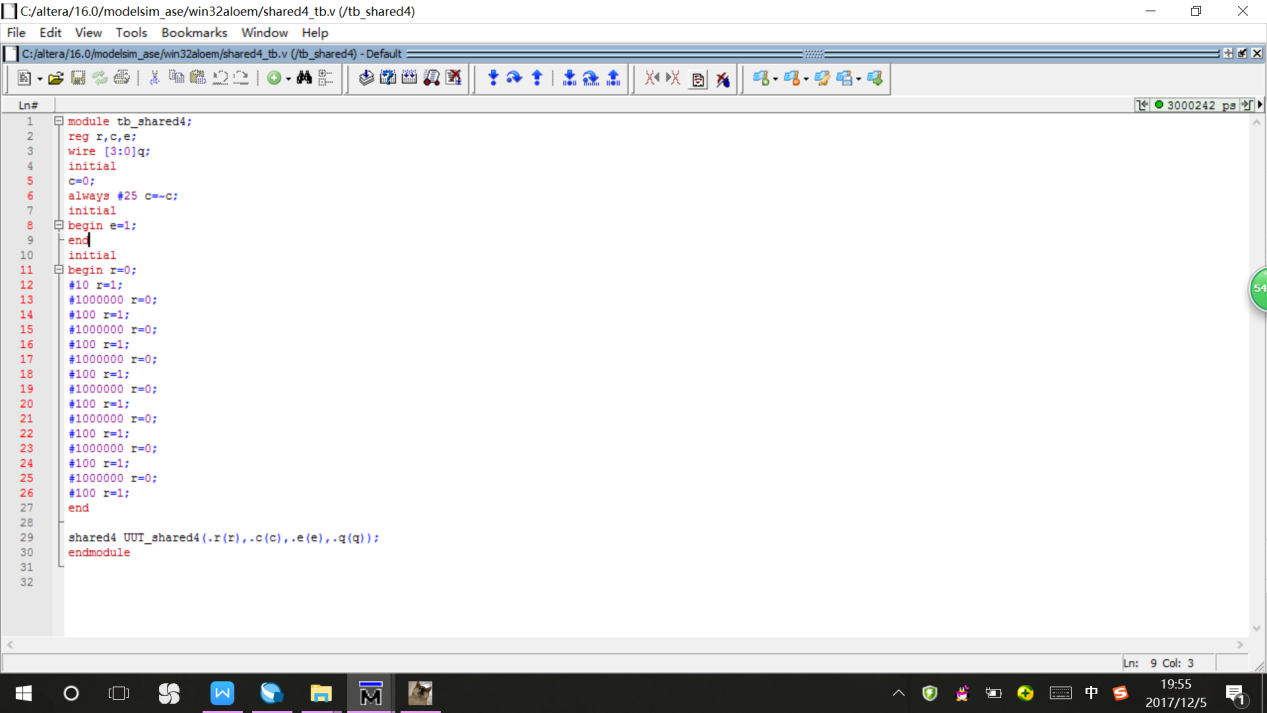
正如wave图中所显示，该代码设计实现了复位功能，计数功能，并解决了预定溢出的问题。

1. **如何设计一个定时器,(如：时钟频率20M,定时为1秒)并编写testbench（在第3题的基础上完成）**

（1）shared4的源代码：

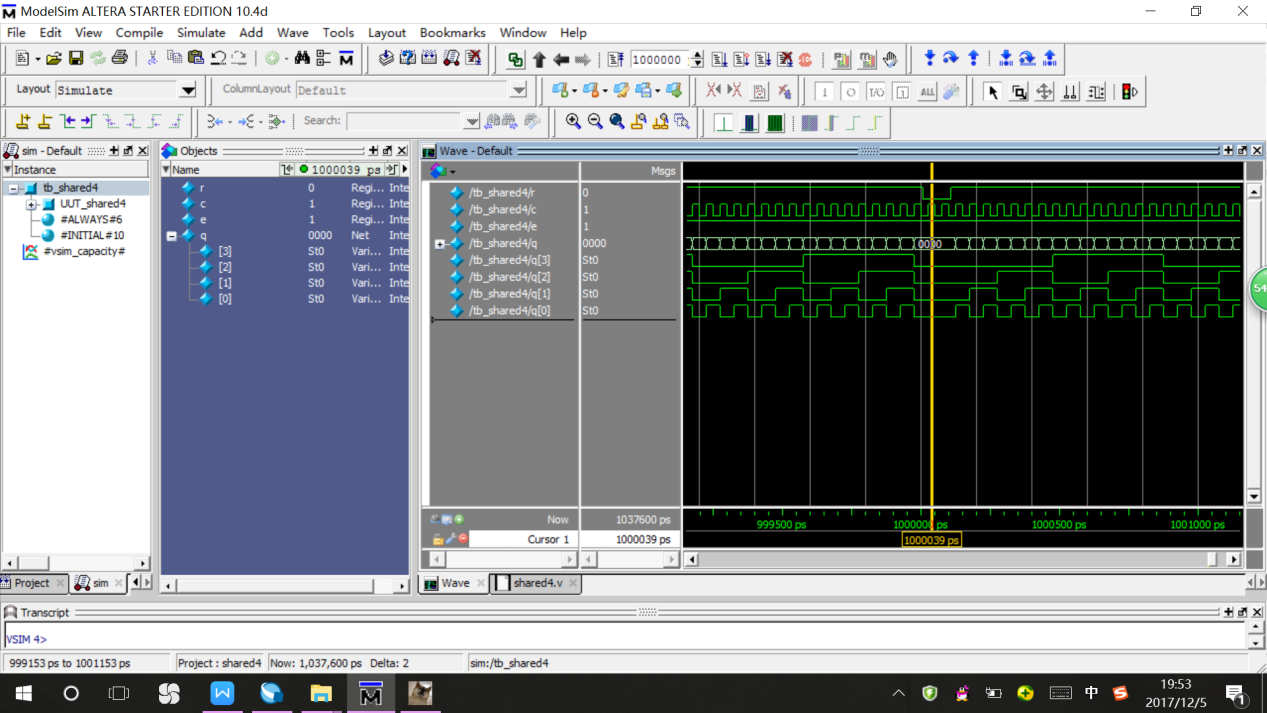


（2）shared4的测试代码：

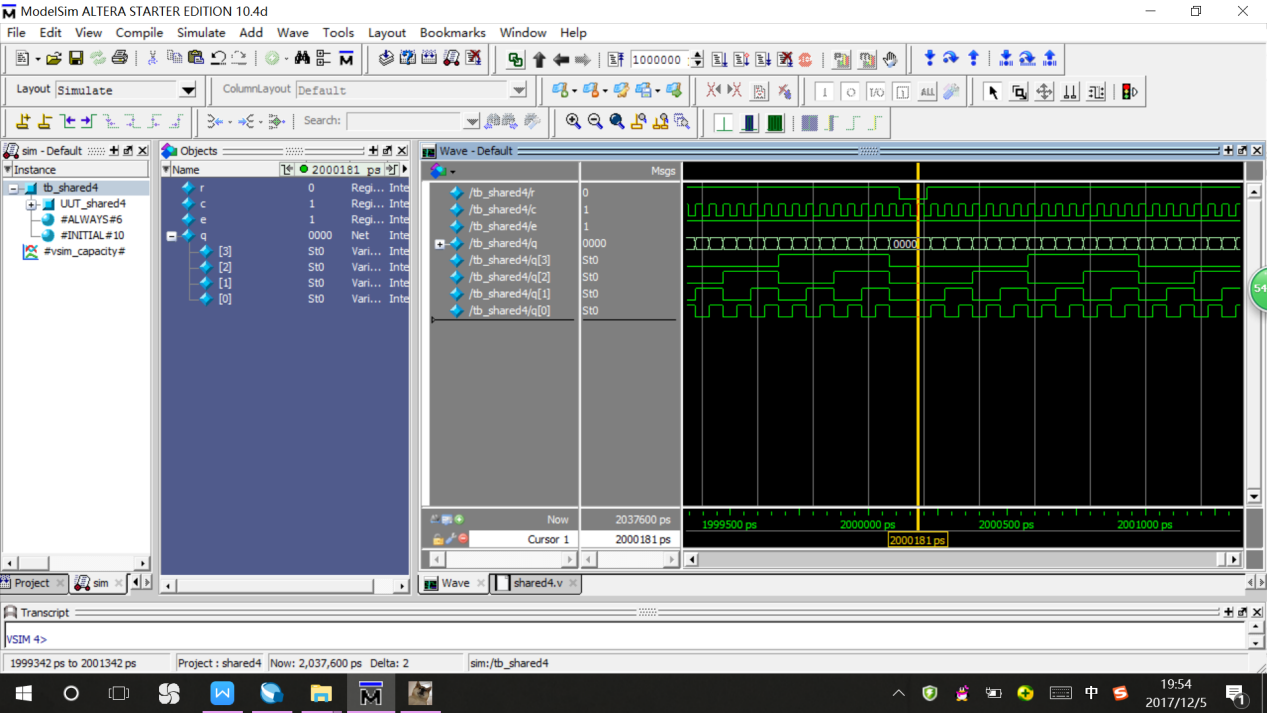


（3）shared4的wave图：

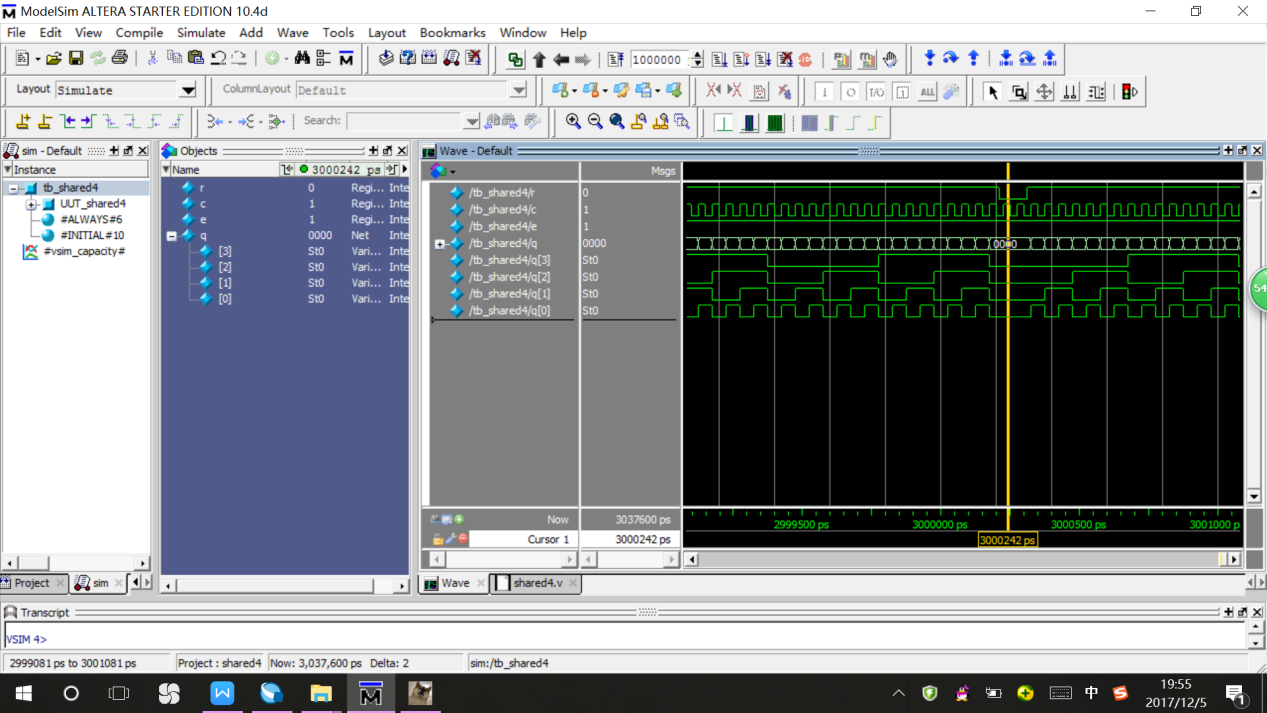
•第一次跳变：



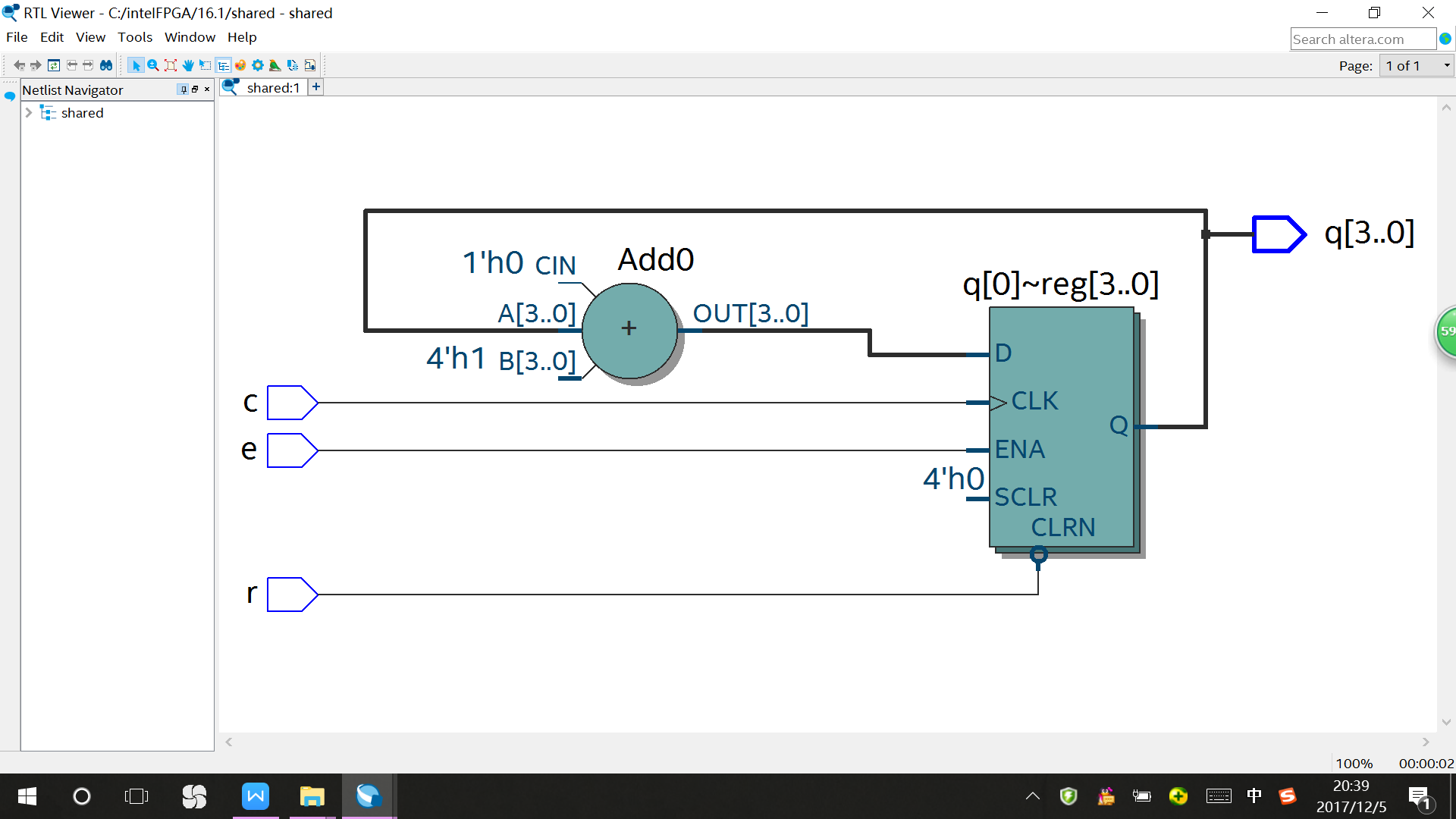
•第二次跳变：



•第三次跳变：



（4）shared4的RTL viewer图：



1. 简要分析：

根据题意将时钟频率设置为了20M，并且定时为1ms。在第3题的基础上，在变量含义并没有改变，进一步通过将测试代码中的复位信号r改变成1ms时间间隔，就实现了定时器每隔1ms变化一次，又迅速还原的特征。

在实验中设置了8组r信号实现了定时器跳变八次，如上wave图中所呈现的前三次跳变，都实现了题意中的功能，证明代码设计正确。

1. 实验收获和心得：

首先本次实验的目的为：

1．建立工程文件，编写模块源码和测试模块,要求测试模块能对源文件进行比较全面的测试。

2．编译源码和测试模块，用测试模块对源文件进行测试，并综合仿真。得到波形图。

3．记录数据并完成实验报告。

4. 使用QuartusII软件的RTL viewer工具查看/确认所设计的电路。

在了解了实验的步骤和目的后开始了本次试验，首先我安装了QuartusII软件，并对已给出的样例进行了初步的分析验证。

在安装和使用QuartusII软件的过程中我遇见了很多小问题，比如在初试QuartusII软件时，我没有正确地建立工程，通过询问同学，我明白了问题的所在，具体解决方案就是讲需要用RTL viewer查看的源代码预先放置在一个新建的文件夹中，然后再添加已有文件到空的工程中，就可以顺利地进行接下来的操作了。

在编写代码和仿真的时候，我的经验总的来说有以下两点：第一，在always的应用方面，测试代码对变量初始化的时候，不能在一个initial下同时写出两个always，否则编译不能通过会出现语法错误，这也是一个知识漏洞和细节问题，提醒我在今后的编写程序的过程中药多注意细节问题。第二，关于shift移位寄存器，注意考虑与已知的之前的电路状态有关系，在编写实验内容的过程中我正确理解了Shift移位过程的含义。

我掌握了如何综合Verilog语言进行时序逻辑设计的使用，学会了测试模块的编写、综合和不同层次的仿真，并安装了QuartusII软件，利用该软件对典例和实验内容的实验电路进行了检查和分析。我认为在以后编写代码和实验的过程中应该进一步注意细节问题，并且做到实验与理论的相结合，一个很小的理论知识漏洞就有可能导致实验出现各方面的麻烦，我也会将这些想法应用到以后的学习和实践中。