**实验序号及名称： 实验 一 基本逻辑门和组合电路的设计**

**一．实验目的**

1. 掌握Verilog编程设计电路，熟悉ModelSim仿真工具。

2. 掌握基本逻辑门和加法器的功能。

**二．实验工具**

ModelSim

**三．实验内容**

1. 用Verilog完成三态门的设计，并用ModelSim进行仿真。

2. 用Verilog完成4位加法器的设计，并使用ModelSim进行仿真。

**四．实验过程（包括源程序及仿真结果截图）**

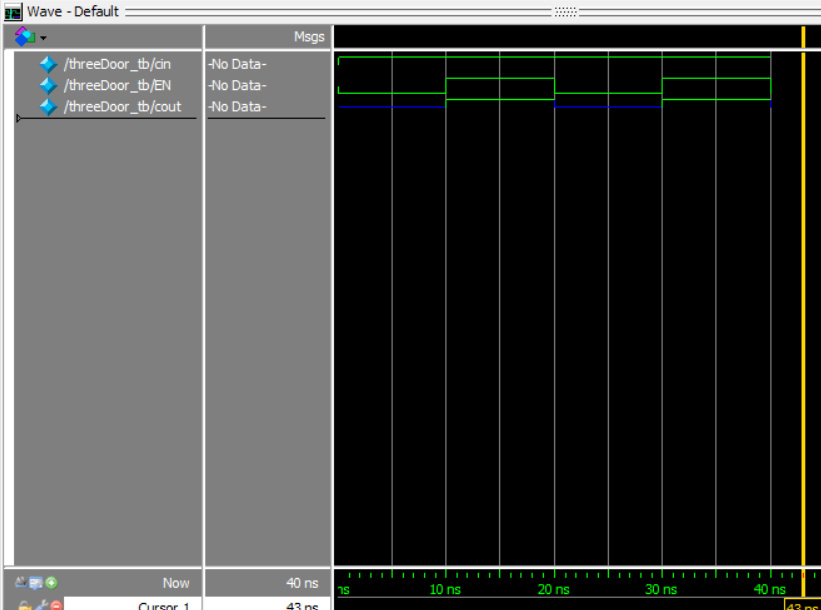
**三态门源码：**

****

**Testbench**

****

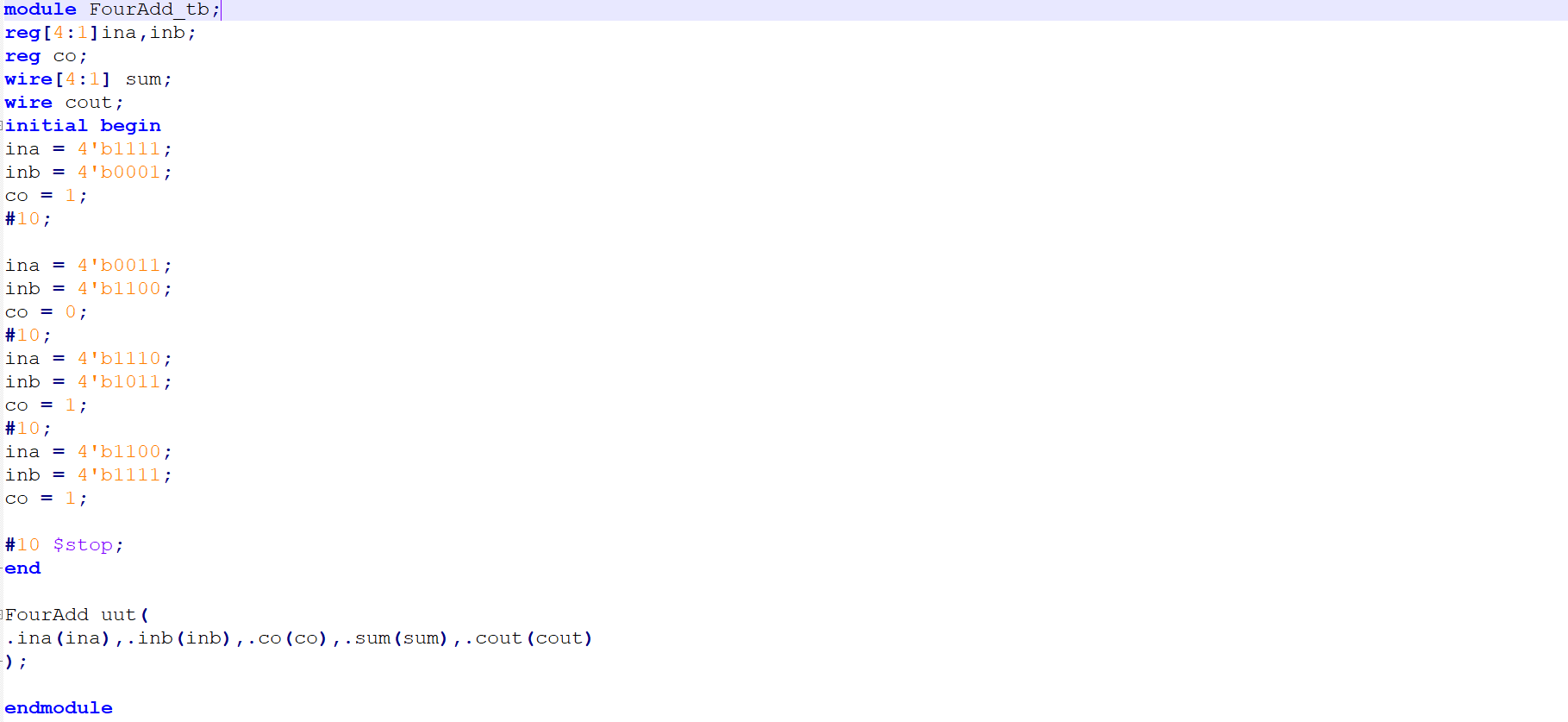
**仿真结果：**



**四位加法器源码：**



**Testbench**

****

**仿真结果：**



**四．实验感想、体会**

第一次使用Verilog语言，花了一个下午，看了老师发的华为教程还有一个ppt教程，然后取网上找了一个两小时快速入门Verilog的视频教程看，看完之后，对Verilog语言整体印象觉得还是挺简单的，就是对一些细节还是不清楚，看了ppt上的几个例子和老师的给的mux的例子后，掌握了Verilog的基本用法，所以写起来还是挺轻松的。但是testbench还是不太会写，这一个的testbench看了好多资料，感觉不太好写，最后还是参考老师那个mux的testbench来写的，出了很多次错后发现，testbench的输入都是reg类型的，不是wire类型的，然后就完成了第一个实验，总体来说学的比较久，但是写起来觉得还是很简单。

**实验序号及名称： 实验 二 组合电路的设计**

**一．实验目的**

1. 掌握Verilog编程设计电路，熟悉ModelSim仿真工具。

2. 掌握译码器、数据比较器的功能。

**二．实验工具**

ModelSim

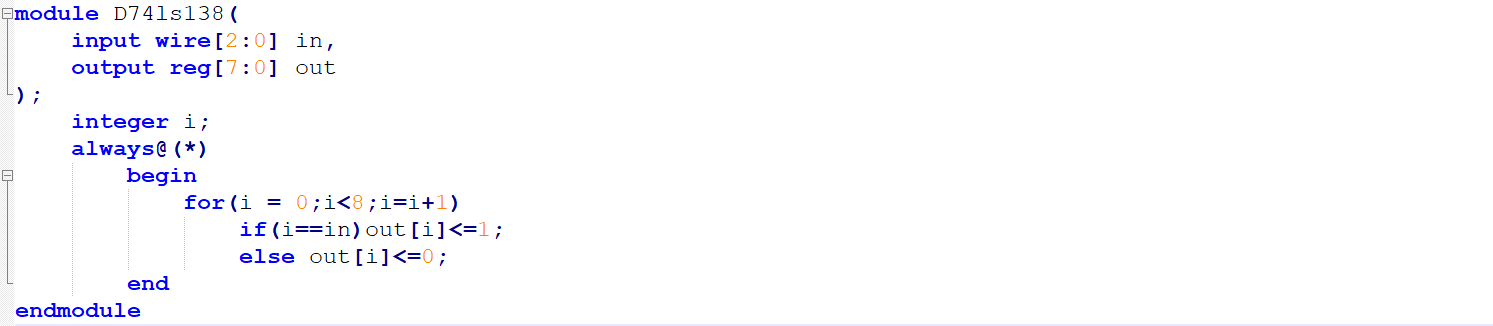
**三．实验内容**

1. 用Verilog完成3-8译码器的设计，并用ModelSim进行仿真。

2. 用Verilog完成两个4位二进制数据比较器的设计，并使用ModelSim进行仿真。

**四．实验过程（包括源程序及仿真结果截图）**

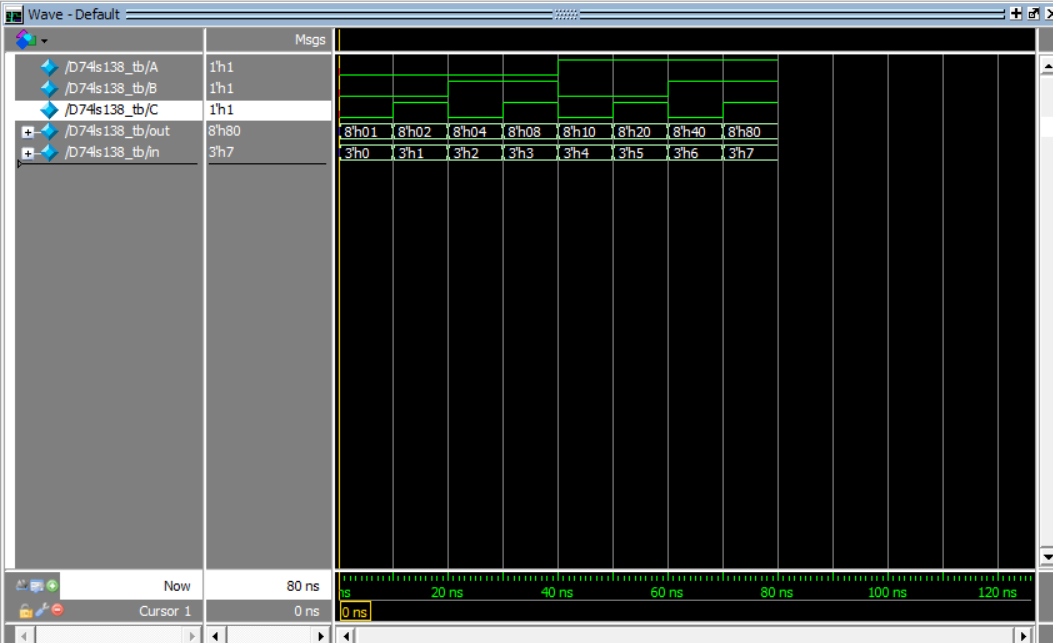
**三八译码器源码：**

****

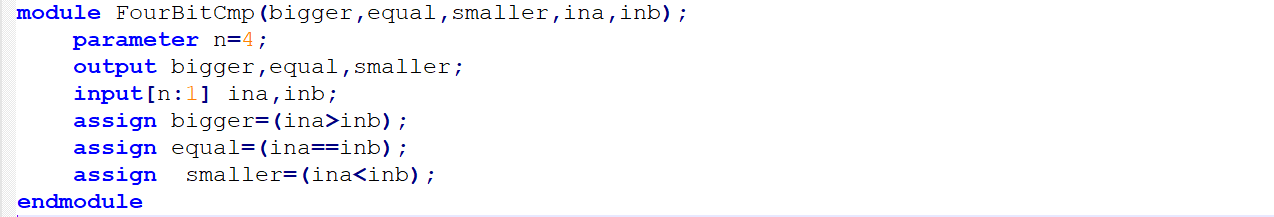
**Testbench：**

****

**仿真结果：**



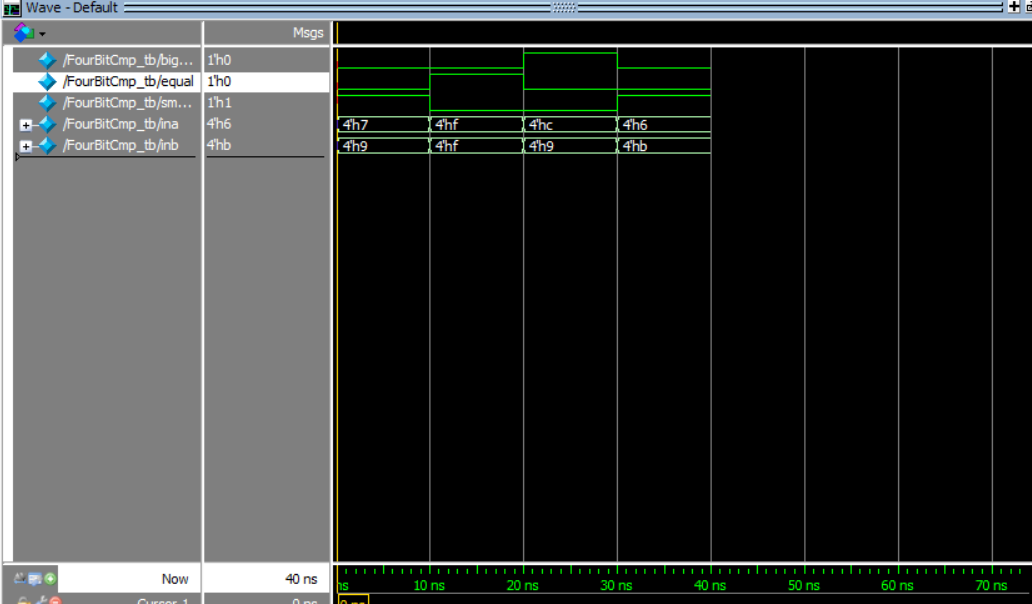
**四位比较器源码：**

****

**Testbench:**

****

**仿真结果：**



**四．实验感想、体会**

四位比较器的实现一开始觉得要一位一位进行比较，后来看了许多例子后发现直接将四位数字当作整体，当成一个数字就可以直接进行比较，非常简单。

三八译码器最初是想用case语句来列举所以的情况，但是如果位数多了以后，四位，五位，就需要很多行代码，但由于Verilog能直接将二进制的一位一位的数据与十进制整数进行比较，所以就能用一个for循环语句，将选中的输出位设为一，就能很简便的输出译码结果。但是对于testbench还是列举了八个信号，因为对于testbench还不熟悉，所以没有写简便一点的方法。

**实验序号及名称： 实验 三 触发器**

**一．实验目的**

1. 掌握Verilog编程设计电路，熟悉ModelSim仿真工具。

2. 掌握D触发器、JK的功能。

**二．实验工具**

ModelSim

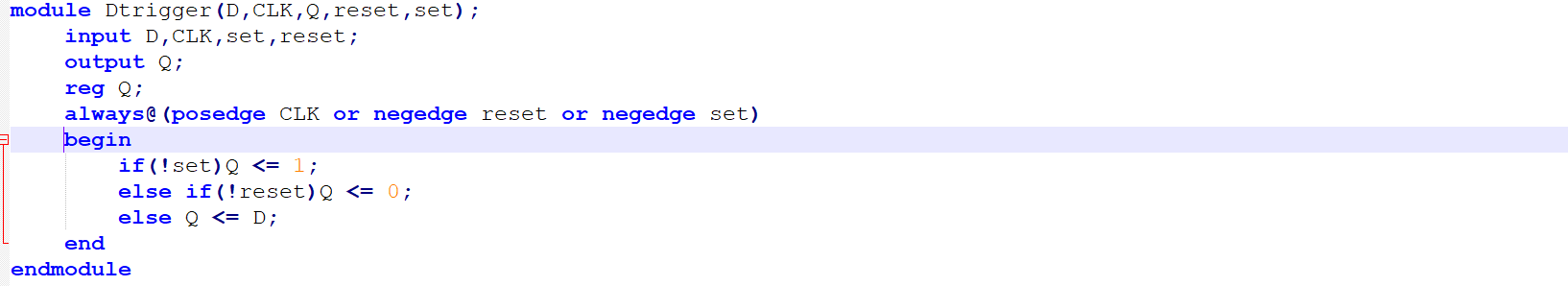
**三．实验内容**

1. 用Verilog完成具有异步清零和异步置1的D触发器的设计，并用ModelSim进行仿真。

2. 用Verilog完成具有同步清零和同步置1的JK触发器的设计，使用ModelSim进行仿真。

**四．实验过程（包括源程序及仿真结果截图）**

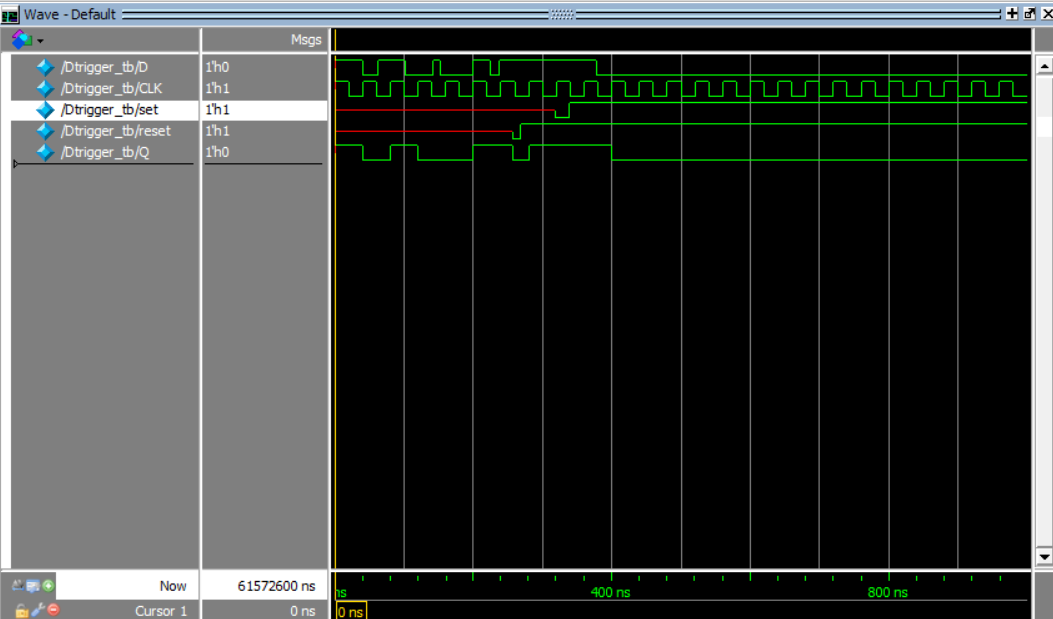
**D触发器源码（上升沿触发）：**

****

**Testbench:**

****

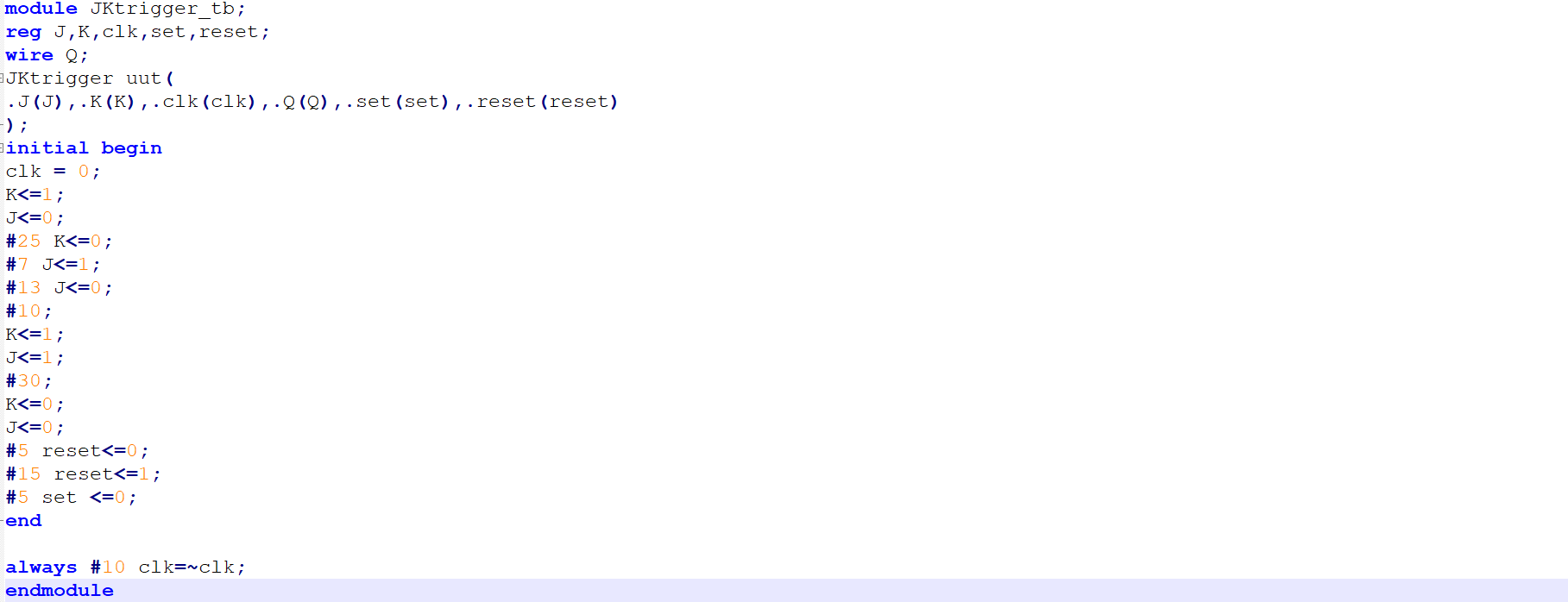
**仿真结果：**



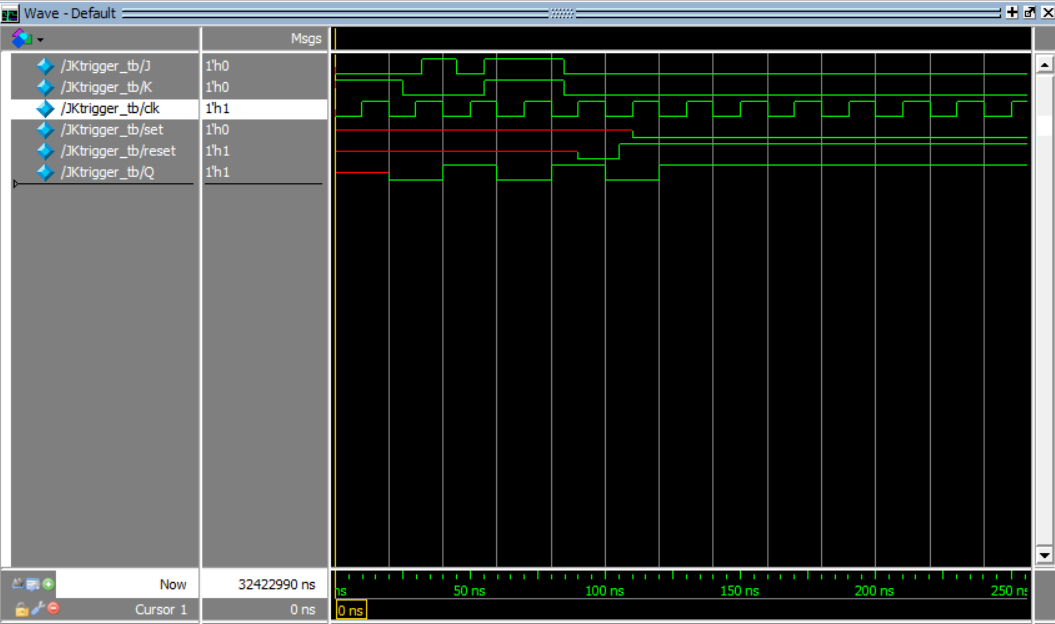
**JK触发器源码（下降沿触发）：**

****

**Testbench：**

****

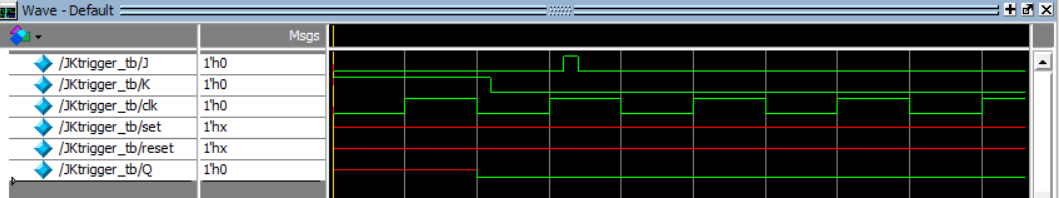
仿真结果：



**四．实验感想、体会**

**D触发器比较简单，用always接受三个敏感信号，然后对每个信号进行分析即可，但是由于if语句的先后顺序，在我的电路了会先检测set信号的变化，因为set信号在if语句里，最先执行。**

**JK触发器我最先想到的是一次翻转现象，在cp=1期间，如果Q­n=0，若J变化，则Qn+1会变成1，我进行了模拟仿真，发现进行仿真并不会出现一次变化，所以模拟仿真应该并不是真正把电路组成现实中的电路，也克服了像一次翻转这样的问题。**



**实验序号及名称： 实验 四 计数器**

**一．实验目的**

1. 掌握Verilog编程设计电路，熟悉ModelSim仿真工具。

2. 掌握计数器的功能

**二．实验工具**

**ModelSim**

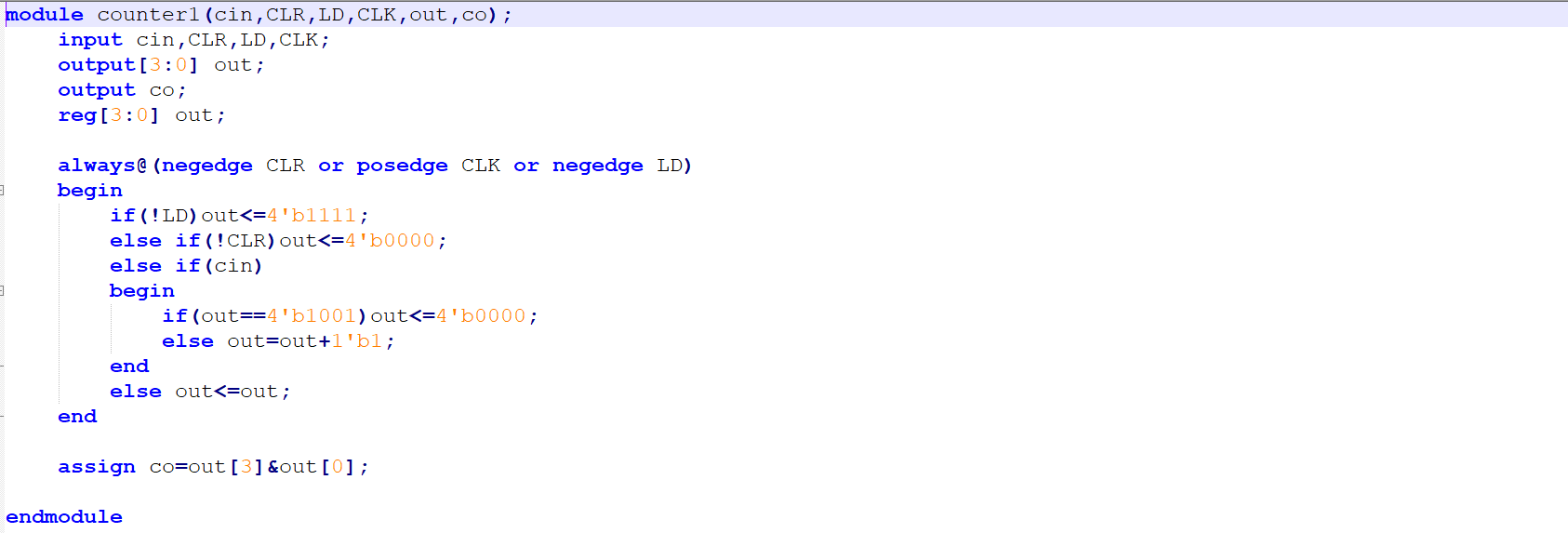
**三．实验内容**

1. 用Verilog完成具有异步清零和异步置1的模10计数器的设计，并用ModelSim进行仿真。

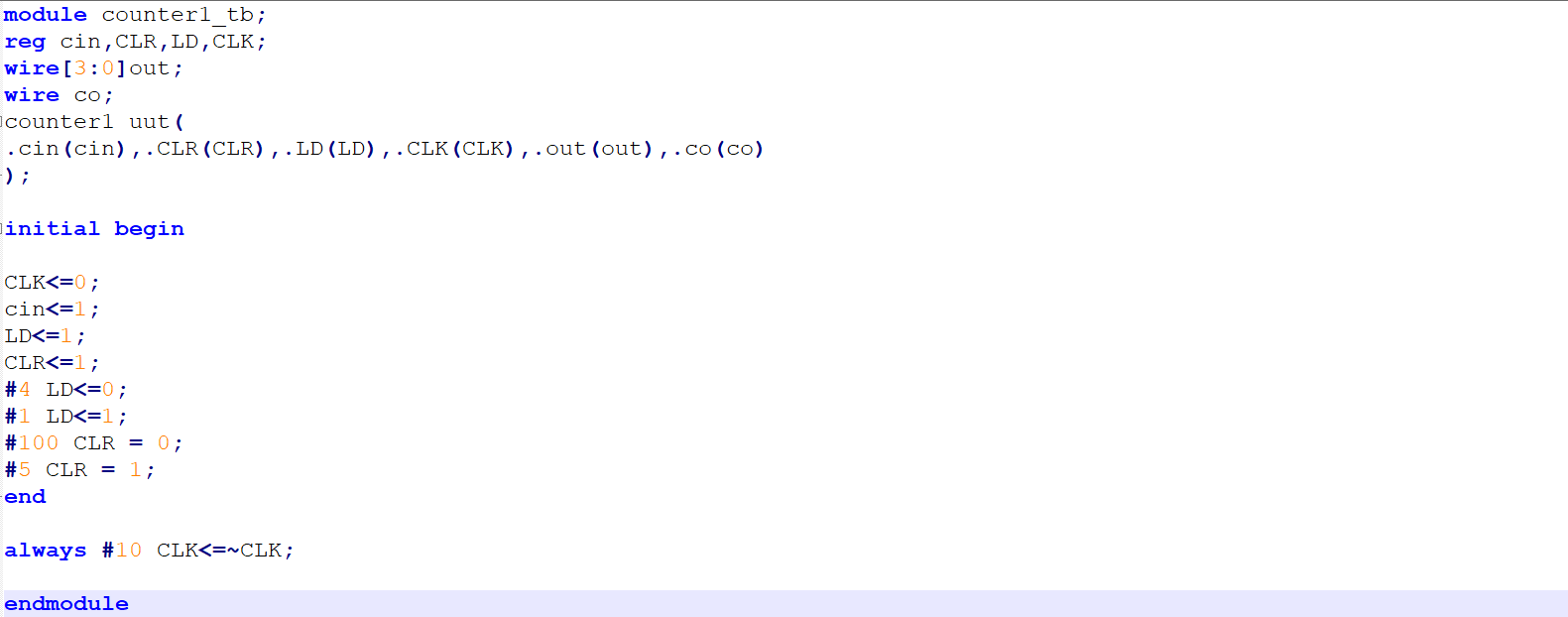
2. 用Verilog完成具有同步清零和同步置1的模16计数器的设计，使用ModelSim进行仿真。

**四．实验过程（包括源程序及仿真结果截图）**

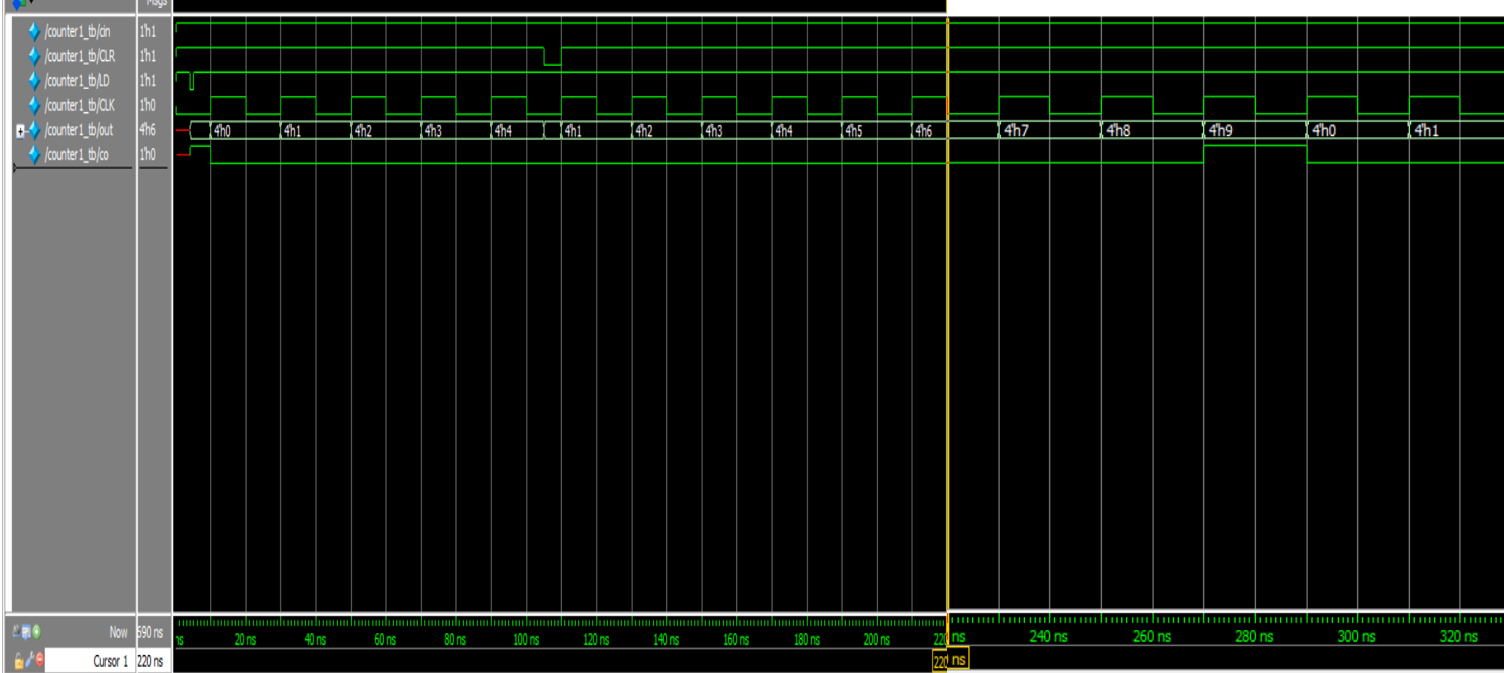
**模10计数器源码：**

****

**Testbench：**

****

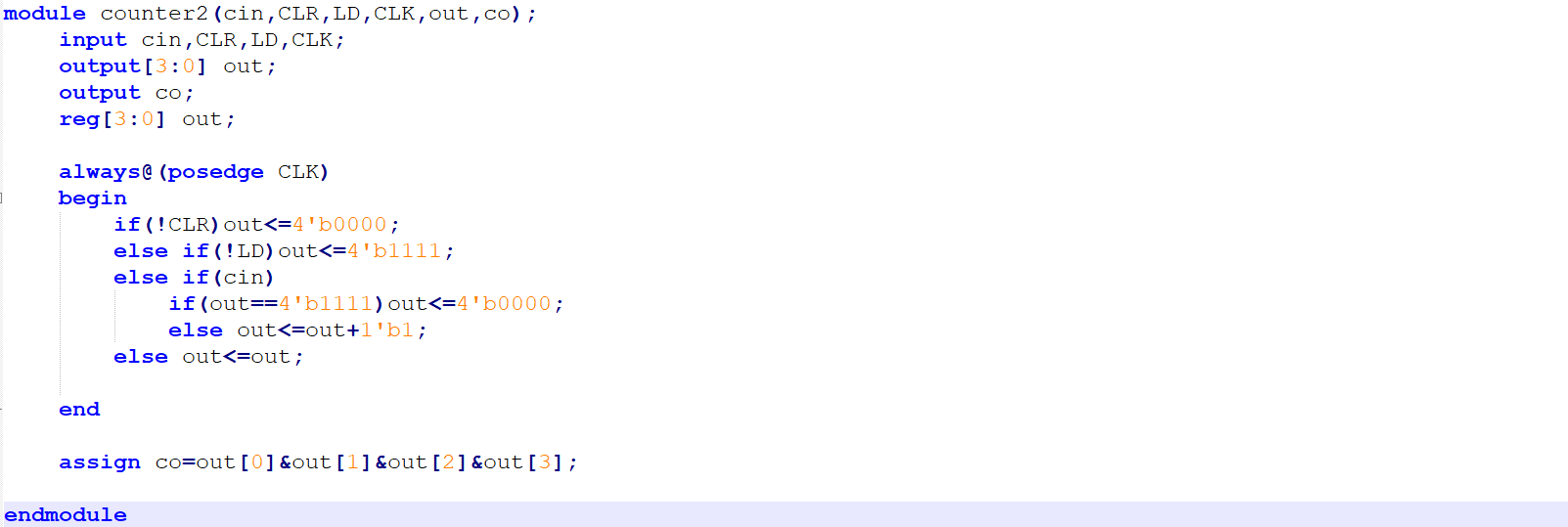
**仿真结果：**



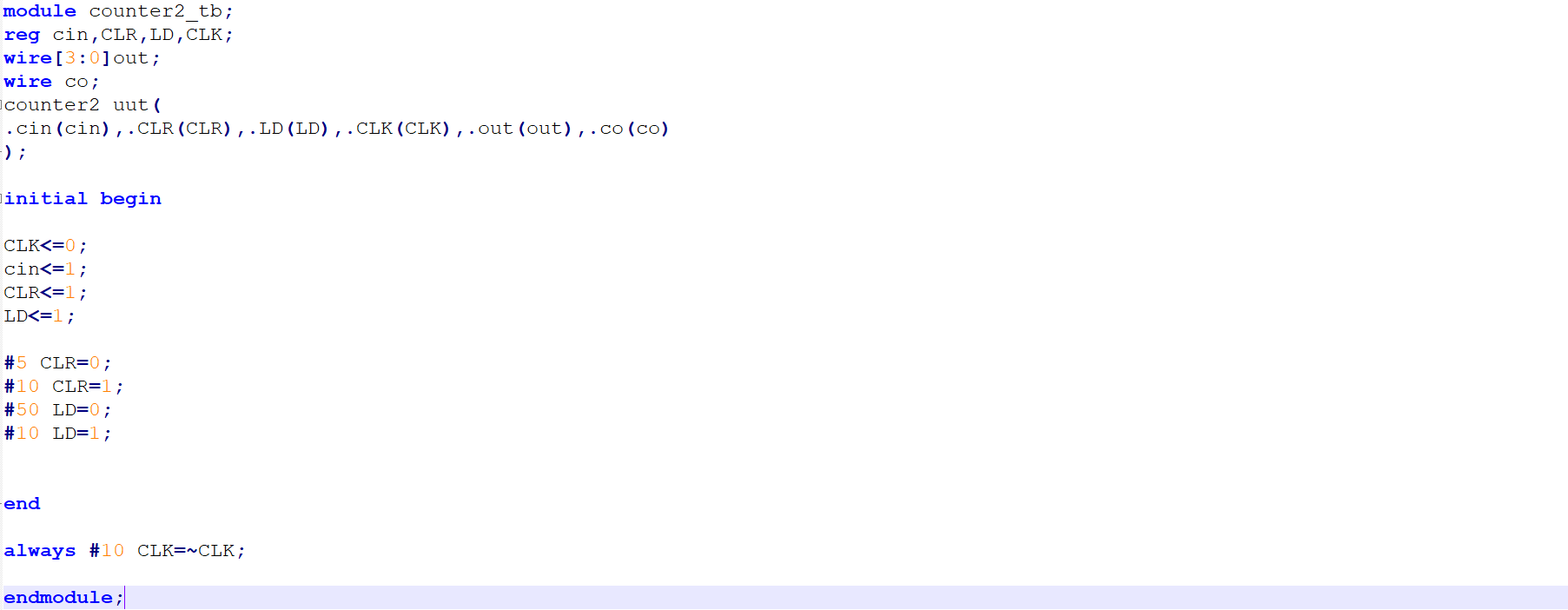
**说明：**

由于如果输出的计数结果out不初始化，就无法进行计数，所以在第一个上升沿前，将LD置为0，异步把out置为4’hf，也就是四个触发器都置为1，然后在4’h4后异步置为0（**图中110nm左右**）。随后计了十个数，在4’h9看到co变为一，下个周期out变回了0，co也变回0（**图中290nm左右**）。

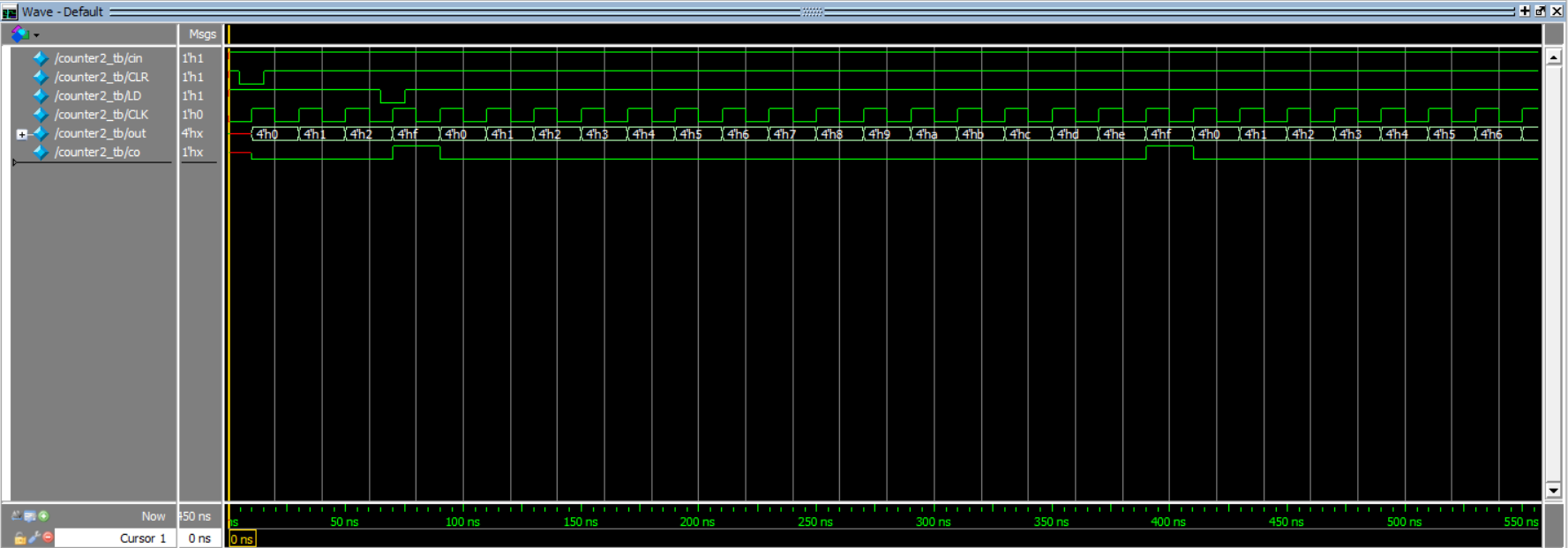
**模十六计数器源码：**

****

**Testbench：**

****

**仿真结果：**



**说明：**在第一个时钟上升沿前，将CLR置为0，在时钟上升沿时，输出out被初始化为0，随后开始计数，cin一直有效，也就是每个上升沿都计数，在4’h2后将LD置为0（**图中75nm左右**），也就是四个触发器的输出值都置位1，out在下一个上升沿时就被置为了4’hf，随后开始正常计数，在4’hf时进位co被置为1（**图中400ns左右**），下一个周期输出out重新回到0，co回到0。

**四．实验感想、体会**

在看实验要求的时候，不知道异步置1和同步置1是什么意思，问了老师才知道是把每个触发器的输出Q都置为1，也就是把输出值置为1111。通过这个实验也让我对计数器加深了印象，理解了细节。一开始不知道怎么做，以为需要通过置数和清零来实现模十和模十六，后来仔细想了一下才反应过来，这个清零端和置1端是为了让用该计数器组成其他模值的计数器才用的到的。所以只需要实现一个清零端，和一个置1端就好了，所以实现还是挺简单的。

通过这四个实验，我觉得用计算机语言来描述数字电路更方便，如果是用实际的物件来实现一个计数器，要列出状态转移方程，还需要编码，连接多个触发器才能实现一个计数器，而通过电脑模拟，就只需要写出电路的逻辑，就能用计算机自动模拟，非常方便。