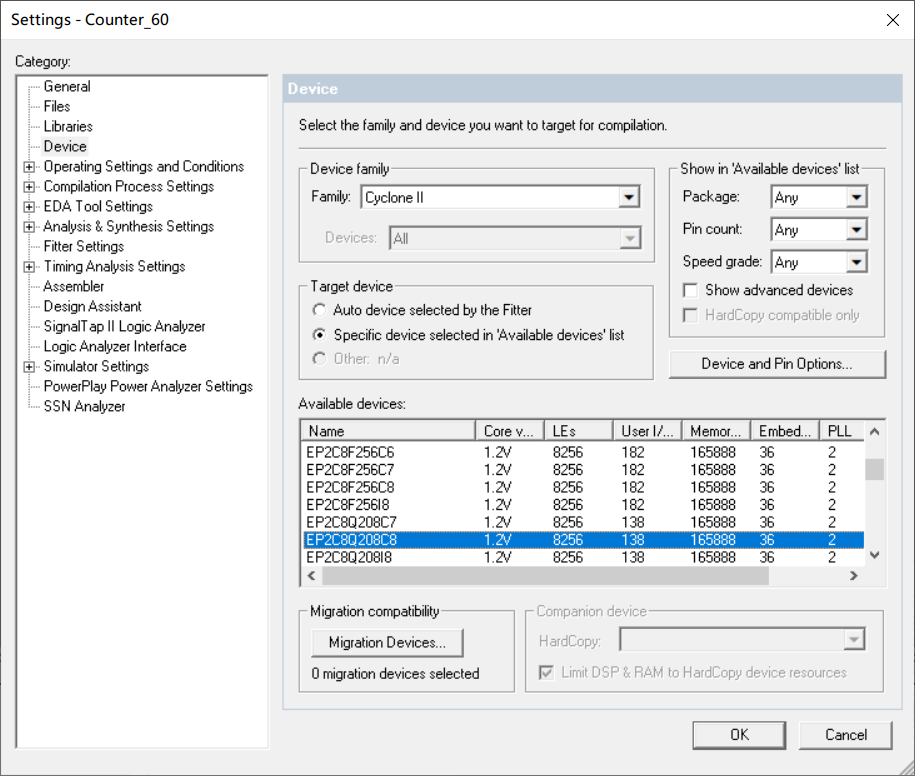
**实验七 可编程小系统设计**

1. **实验目的和要求**
2. 了解可编程数字系统设计的流程；
3. 掌握Quartus II 软件的使用方法；
4. 掌握原理图输入方式设计数字系统的方法和流程；
5. **实验原理（基础部分）**
6. **观察并记录实验箱上的FPGA型号，新建一个Project，器件选用实验箱上的FPGA：**

实验箱上的FPGA型号为Cyclone II-EP2C8Q208C8，在Quartus中的Assignments-Devices中选择对应的器件：



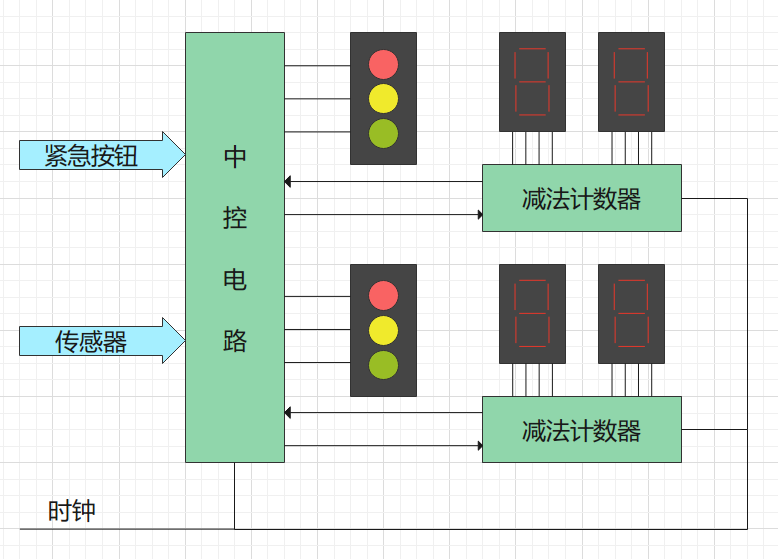
1. **设计思路：**

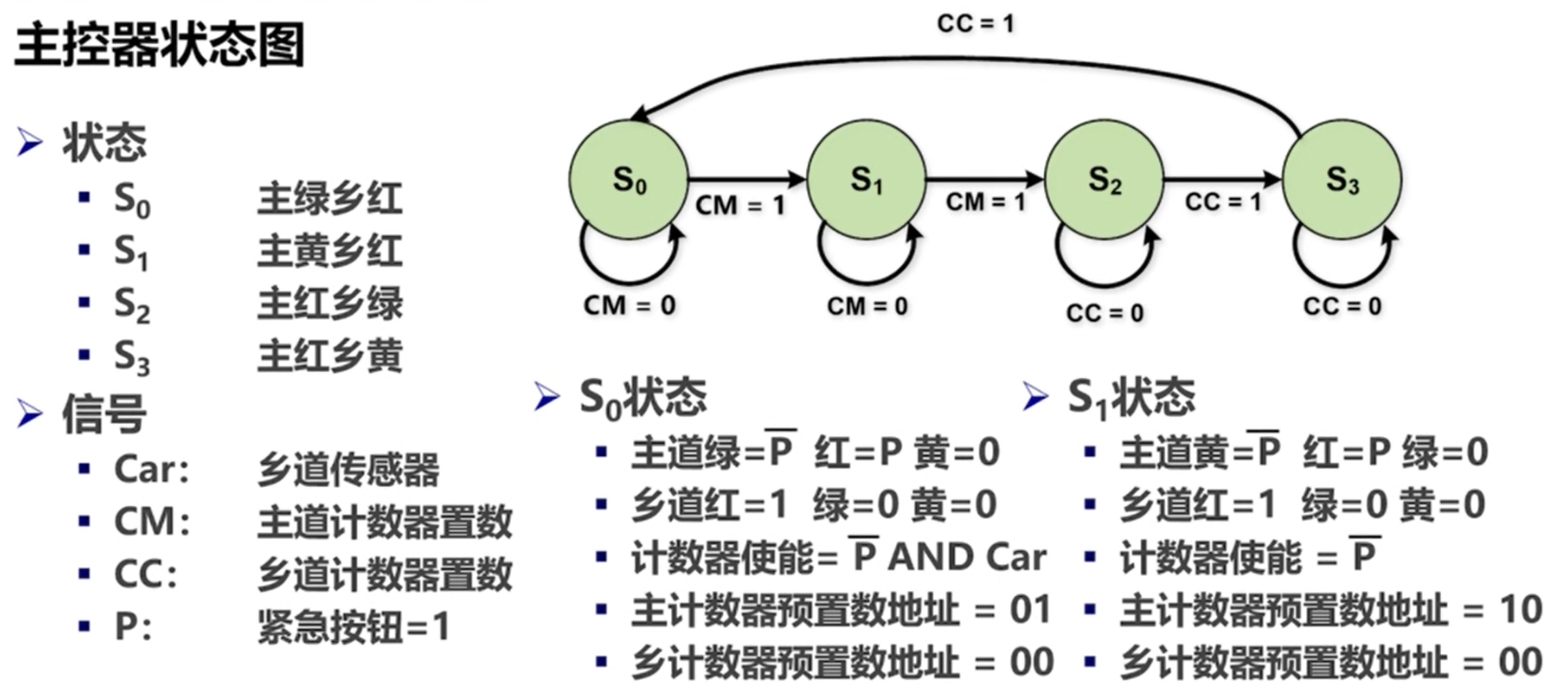
将整个系统分成以下几部分：

外部输入：紧急按钮、乡间道车辆传感器；

显示部分：用以显示倒计时的数码管、红绿黄三色灯；

中央控制部分：倒计时计数器（分为主干道和乡间道两部分），涉及外部输入信号的逻辑处理。

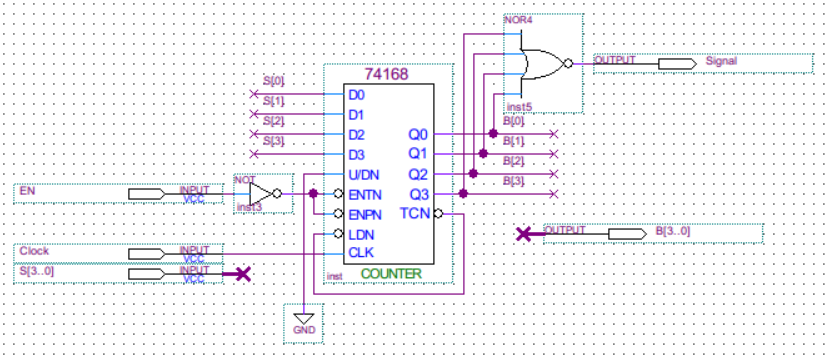




1. **新建一个原理图文件，设计一个可变模减法计数器，并用功能仿真进行验证：**
2. **设计：**

用到的主要元器件是74168，选择其减法计数模式。并使用同步置数的方式来完成计数到0后再次载入数据重新开始计数循环的操作。另外需要添加这样几个输入信号：时钟源、载入的数据（8421BCD码）、使能端EN来控制是否工作。

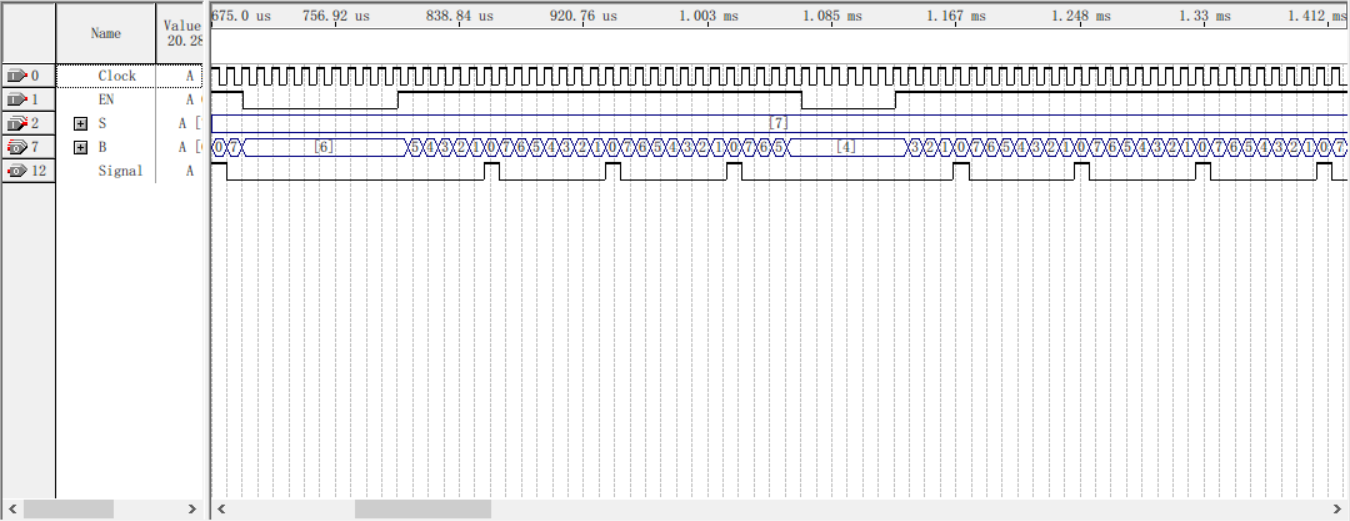
1. **逻辑电路图：**



输出信号用于两片计数器间的级联，以及向中央控制电路传递计数完成、状态切换的信号。

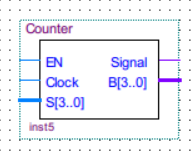
1. **使用功能仿真验证：**

创建波形文件，完成功能仿真。



可以看到，计数功能、使能信号以及循环完成的信号都正确。

1. **封装成元件：**

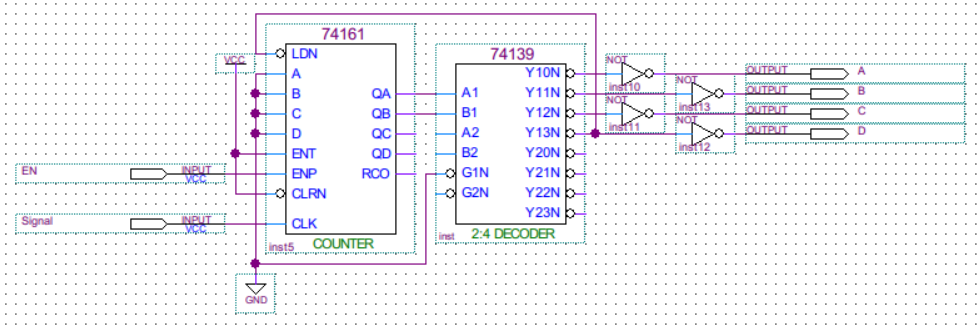


1. **新建一个原理图文件，设计一个状态切换模块，并用功能仿真进行验证：**
2. **设计：**

在基础部分实验中，两个路口的计数器基本相同，需要的模值有3、60、63这三种，对于3和60，只需要给级联起来的两个计数器设置好02、59（十进制）这两种初始值即可；而对于模63的计数器，则需要先输入62，当计数到60时，将低位计数器的模值变为9（否则会出现计数状态为62->61->60->52->51->50->42…的情况，出现错误）。

所以，每个路口的计数器需要4个状态，设计如下。

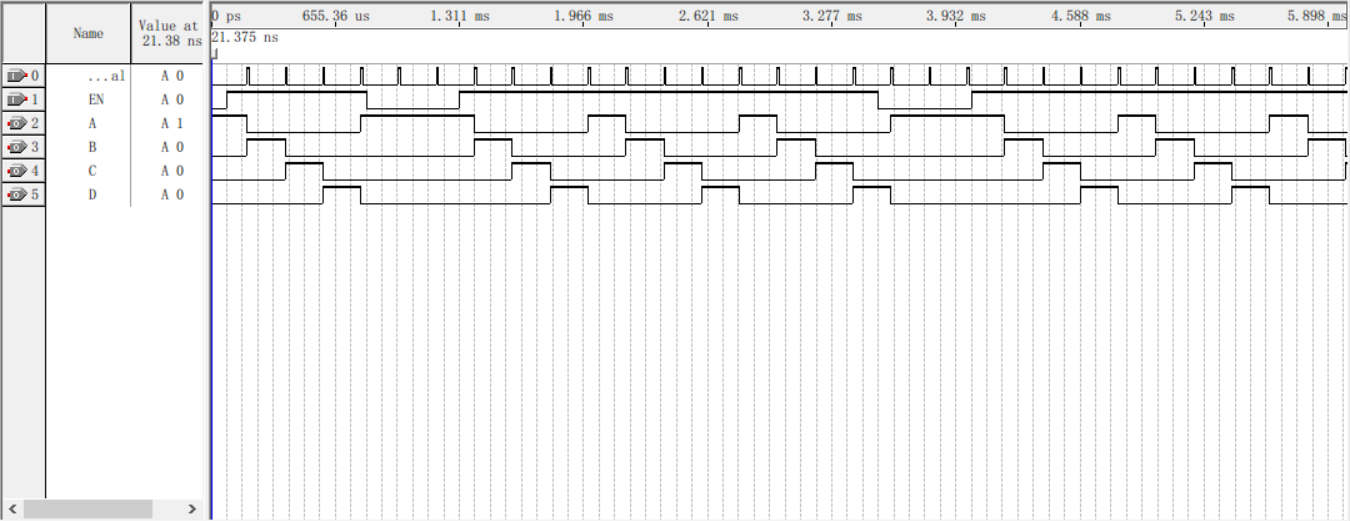
1. **逻辑电路图：**



这里使用了一片计数器74161和一片2-4译码器来实现。

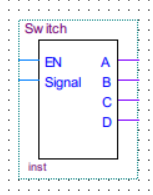
1. **使用功能仿真验证：**

创建波形文件，完成功能仿真。



可以看到状态随Signal输出的上升沿的到来而切换，且EN可以控制电路是否工作。

1. **封装成元件**

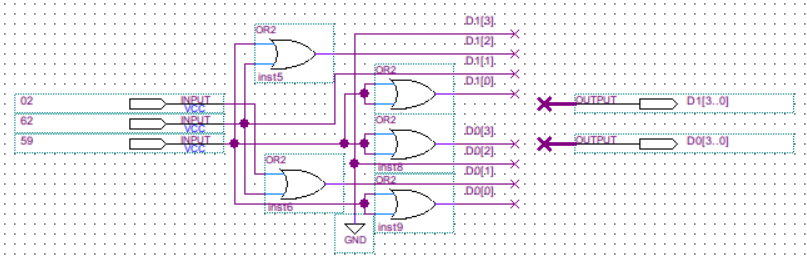


1. **新建一个原理图文件，设计一个编码模块，并用功能仿真进行验证：**
2. **设计：**

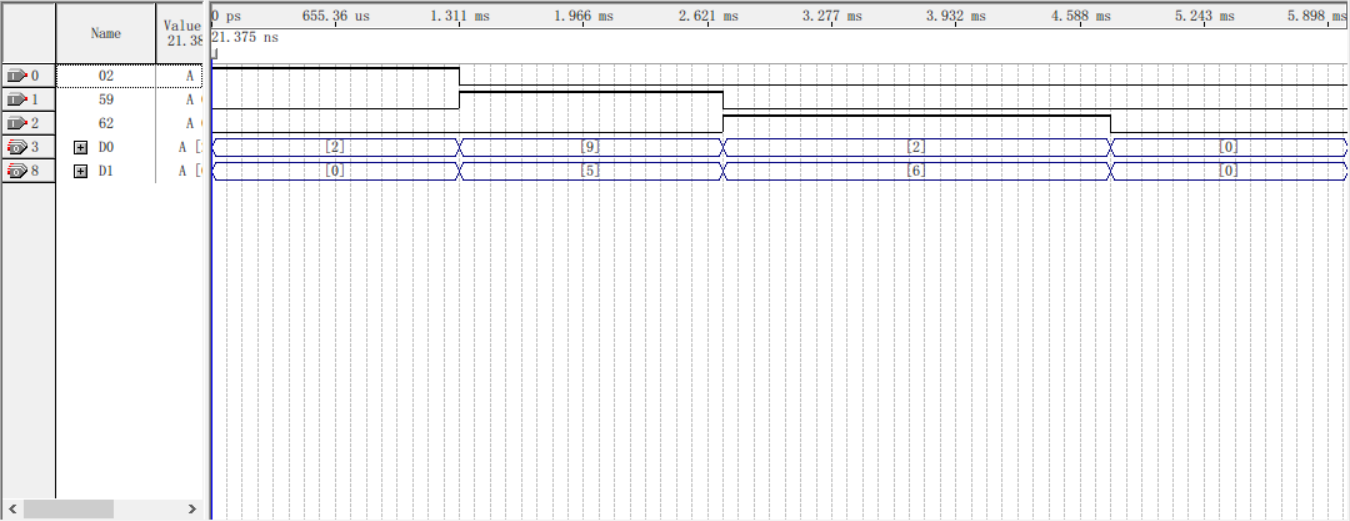
这个模块的功能是进行模值的选择，其输出端是对应的两组8421BCD码，输入给计数器。

需要用到的数值有62、59、02，对这三个数进行转码即可。

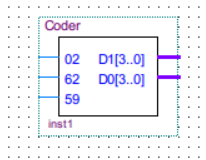
1. **逻辑电路图：**



1. **使用功能仿真验证：**



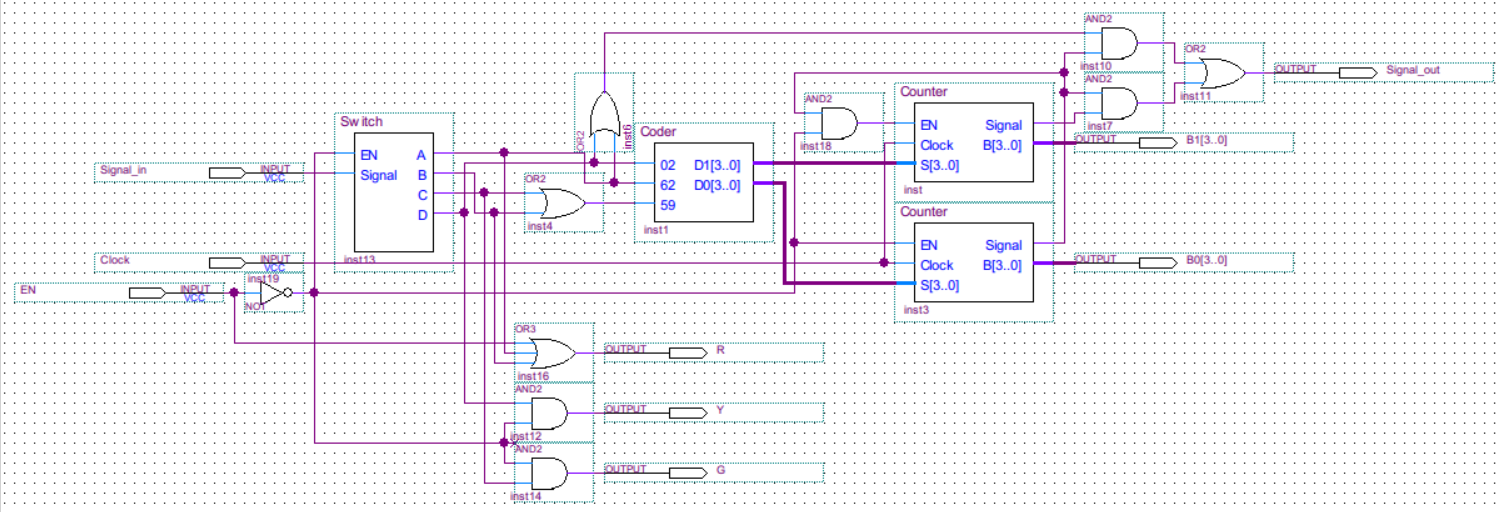
1. **封装成元件：**



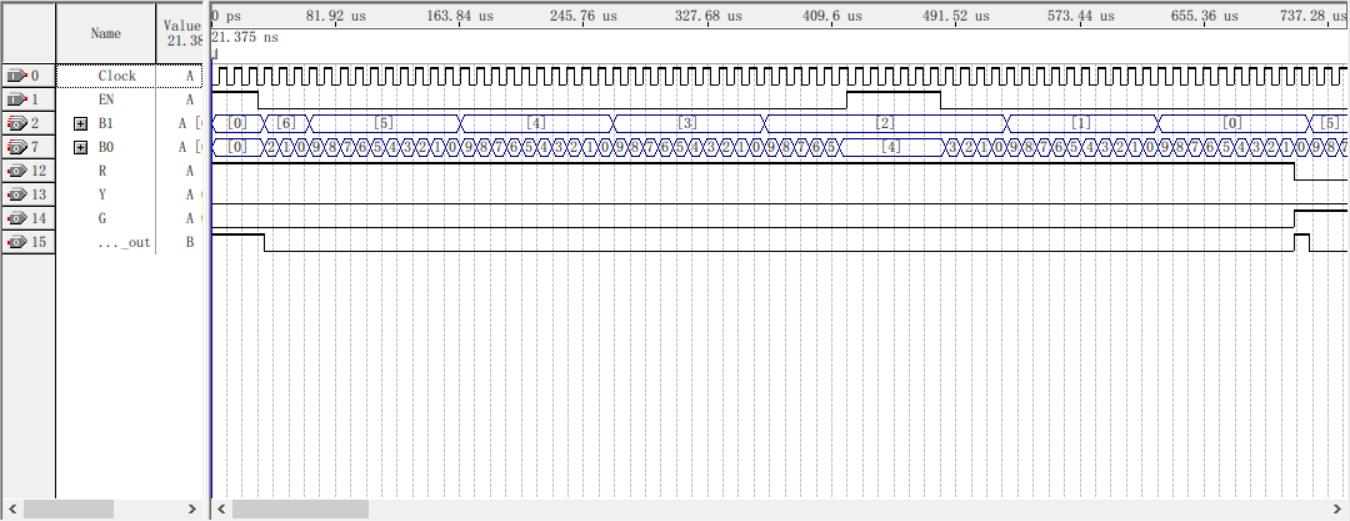
1. **新建一个原理图文件，将上述封装好的元件进行组合、连接，并用功能仿真进行验证：**
2. **设计：**

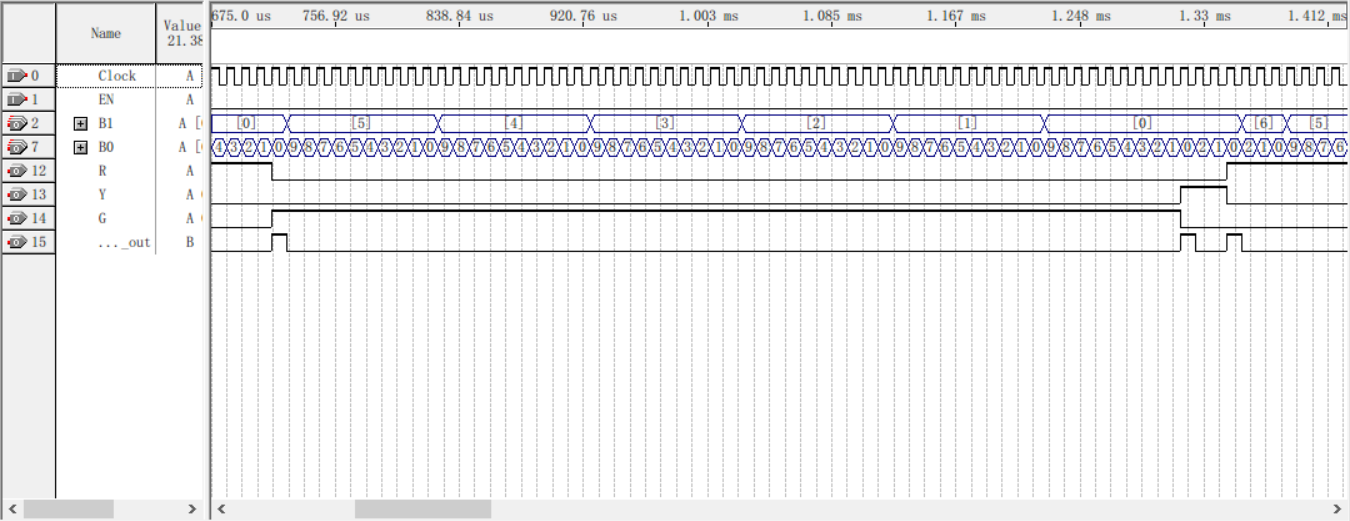
**将两片计数器进行级联，相应的信号输入到状态切换模块（异步），状态切换模块的输出连接到对应模值，通过编码器转为8421BCD码输入到计数器的置数端，以及在相应的状态下引出线来驱动红绿黄三色灯的工作。**

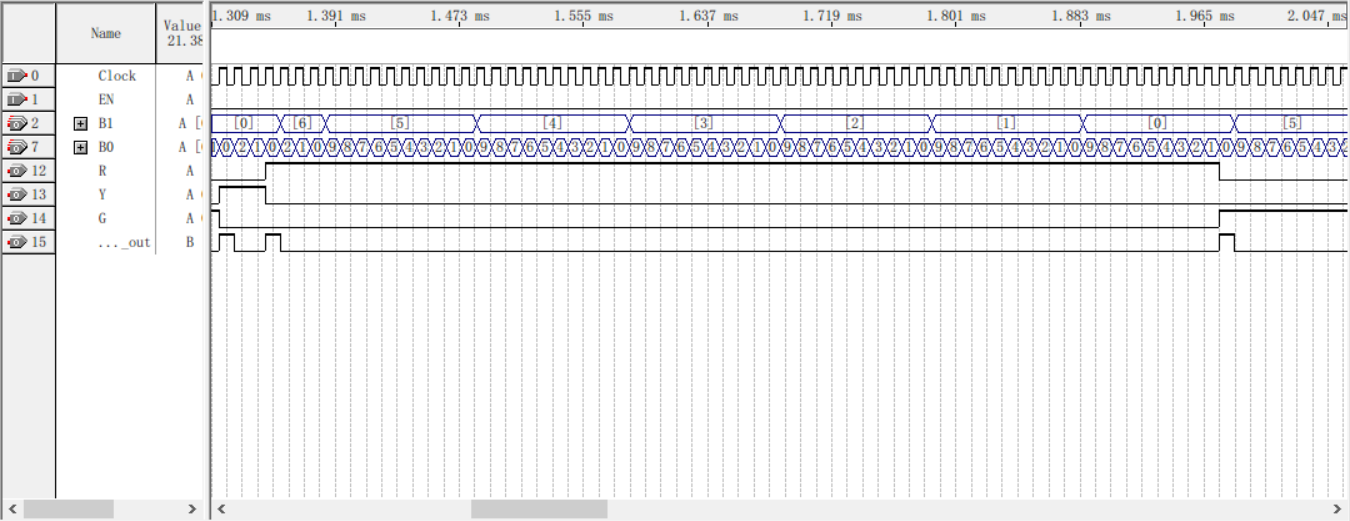
1. **逻辑电路图：**



1. **使用功能仿真验证：**

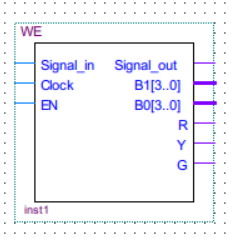




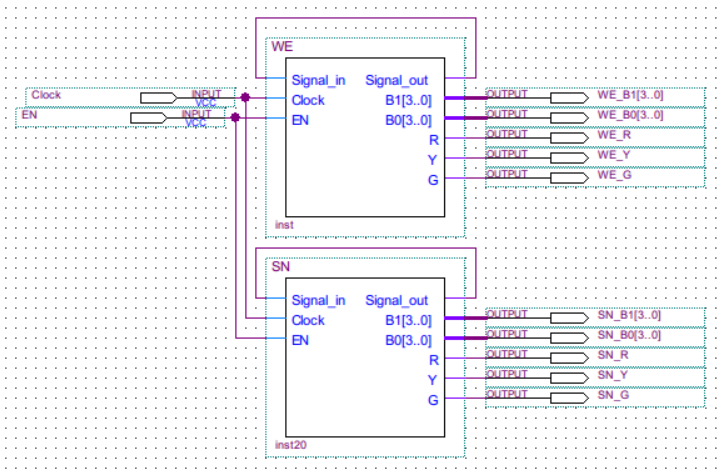


运行结果无误。

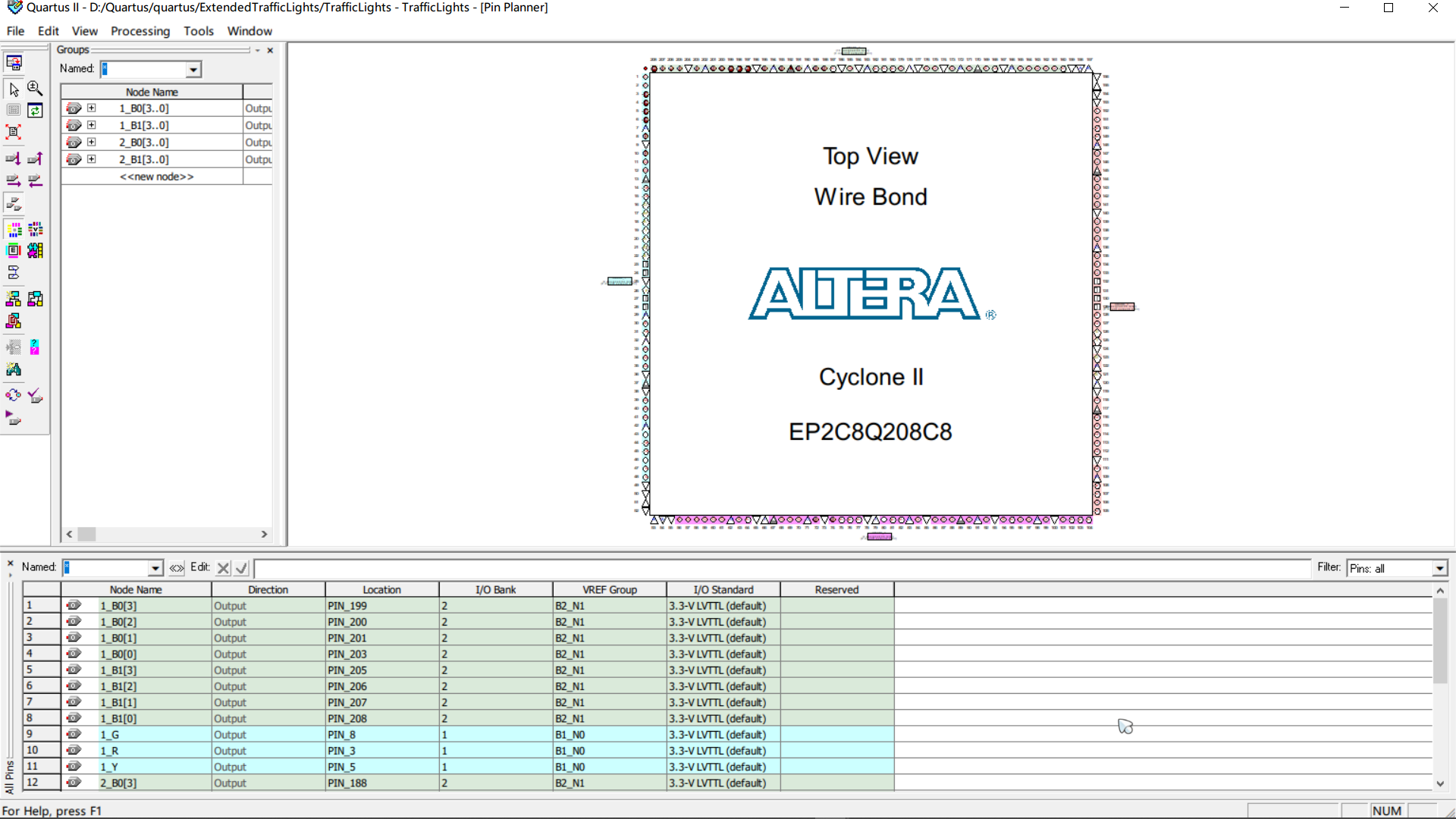
1. **封装成元件：**

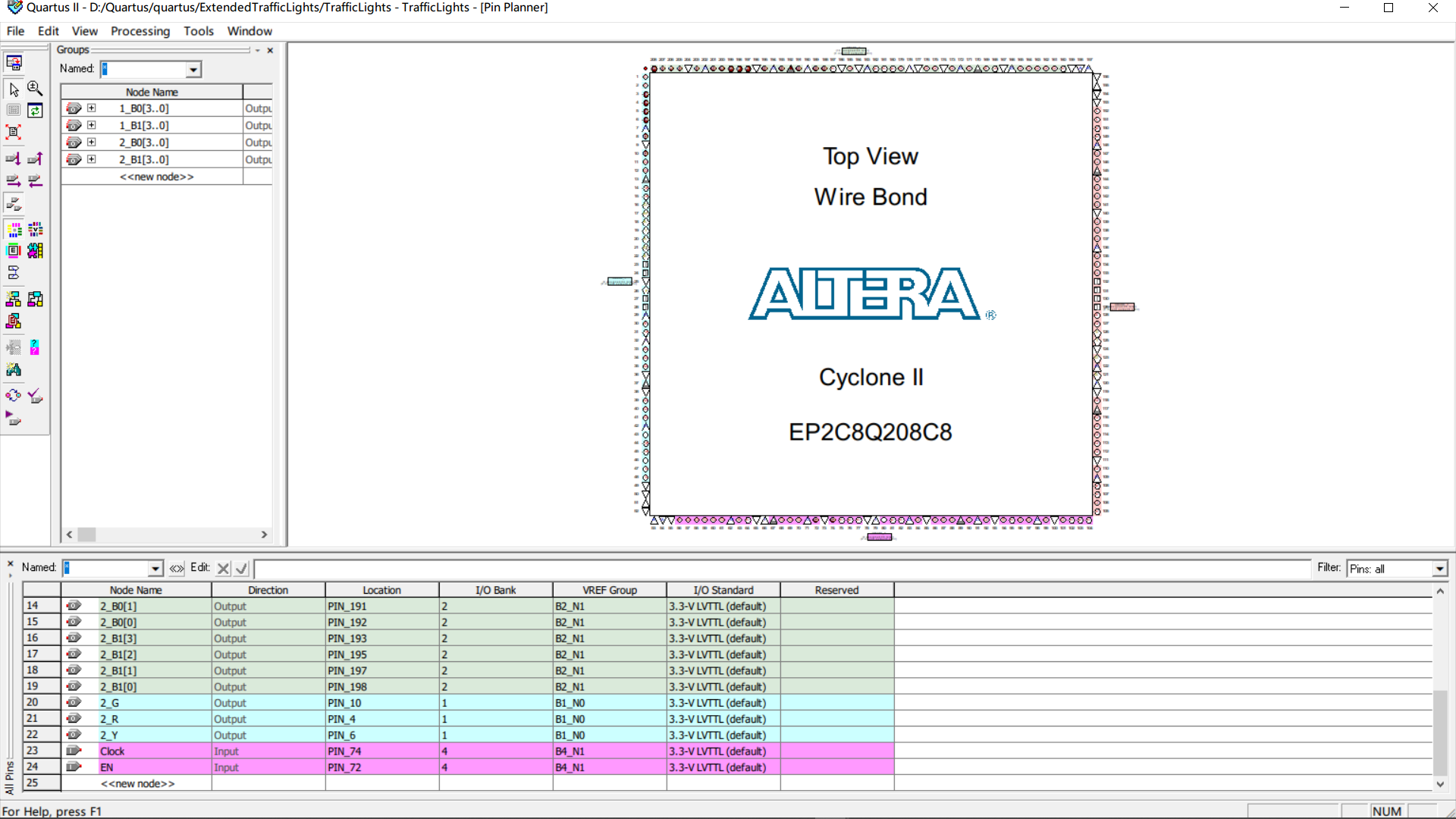


1. **组合，构成交通灯的完整系统：**
2. **逻辑电路图：**



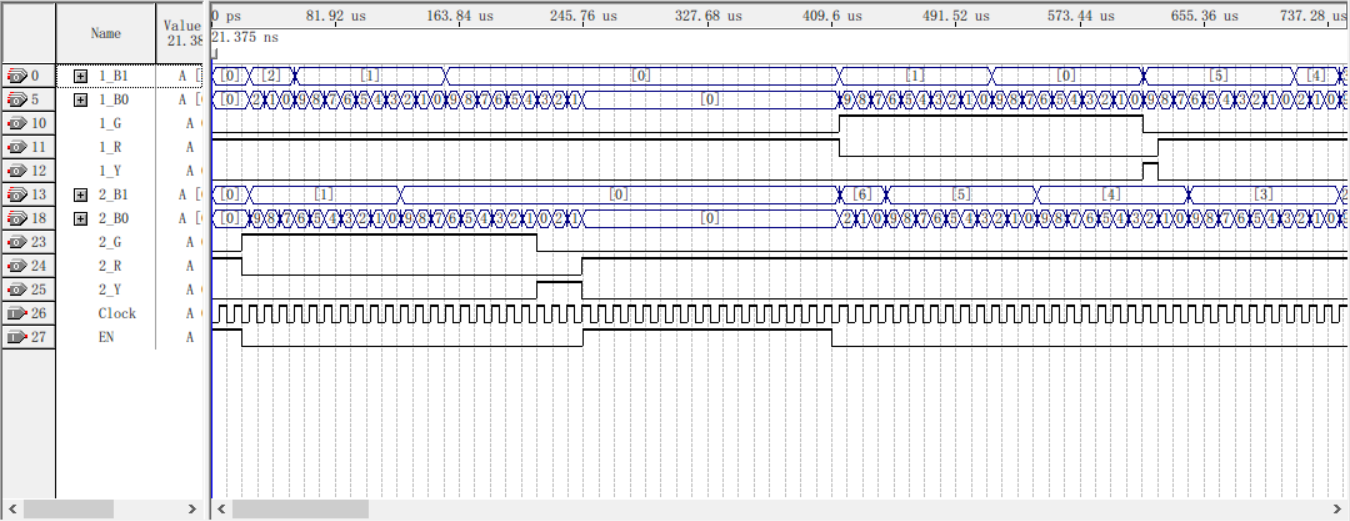
1. **分配管脚：**

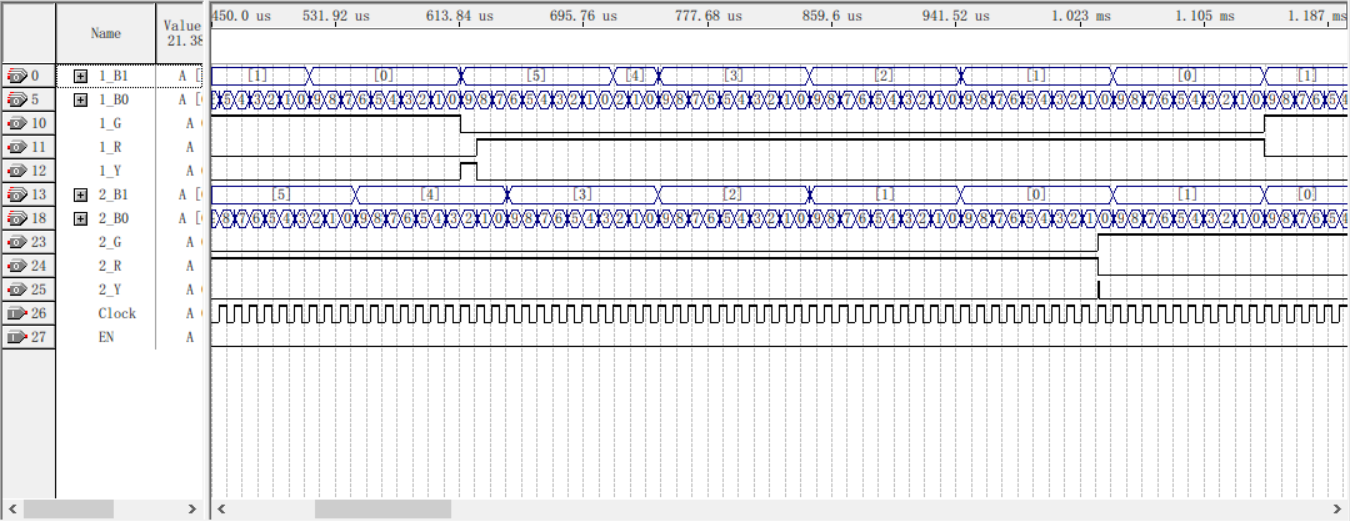




1. **使用时序仿真进行验证：**

时序仿真结果：





运行结果正确。

1. **下载到实验箱，连接时钟进行实物验证：**

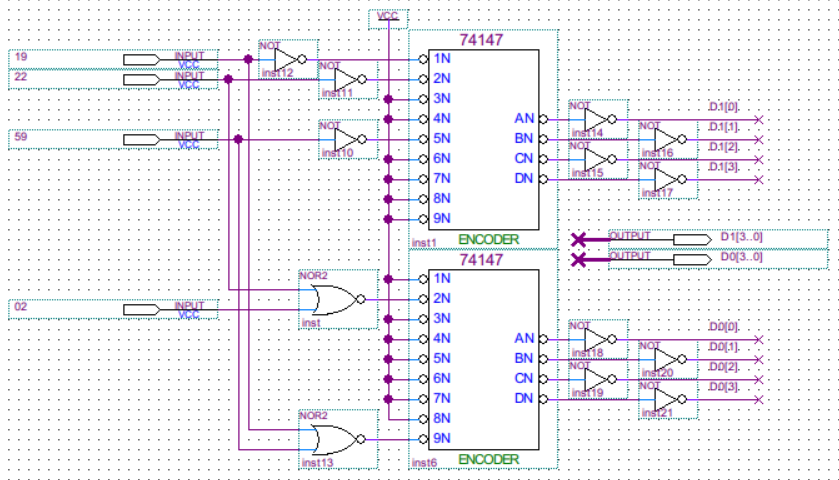
见下“实验记录”部分。

1. **实验原理（扩展部分）**
2. **设计思路：**

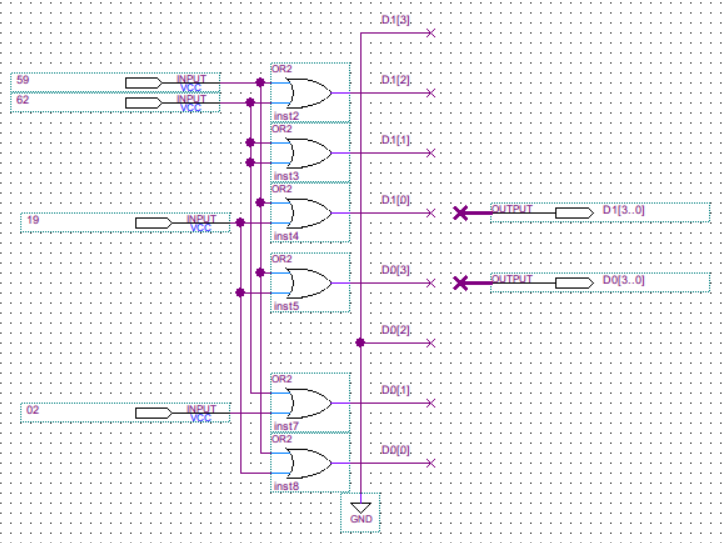
大体上与基础部分相同，需要更改的是计数器的模值和状态切换电路的设计和连接情况，并需要另外添加一个乡间道的车辆传感器输入，来控制计数器是否保持在固定状态或进行计数。

1. **逻辑电路图：**

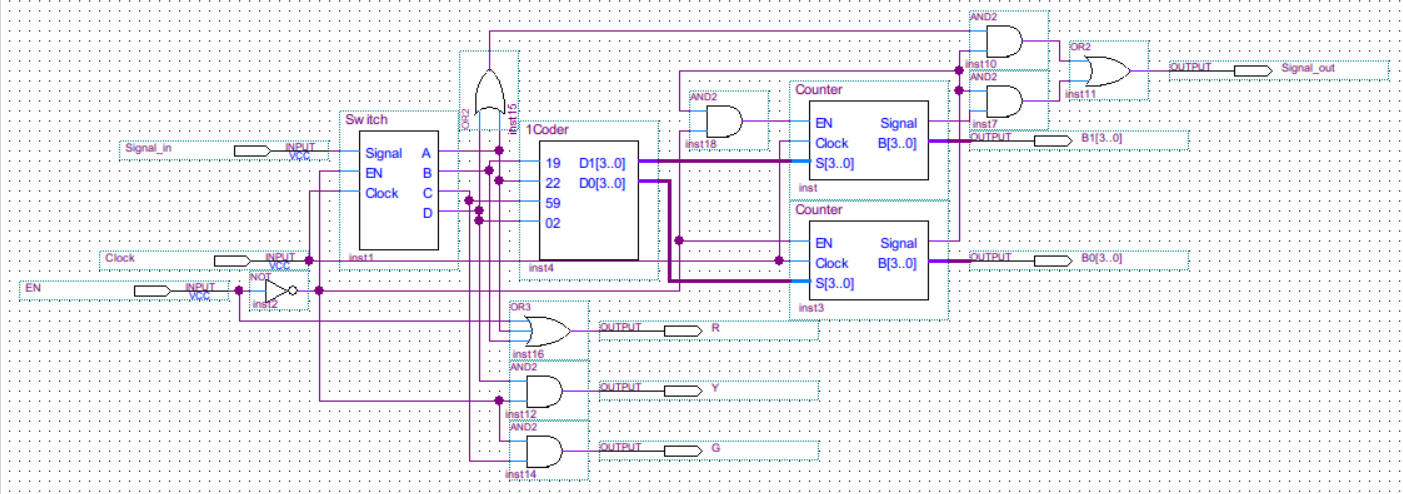
这里只展示与前面部分有较大改动的部分:

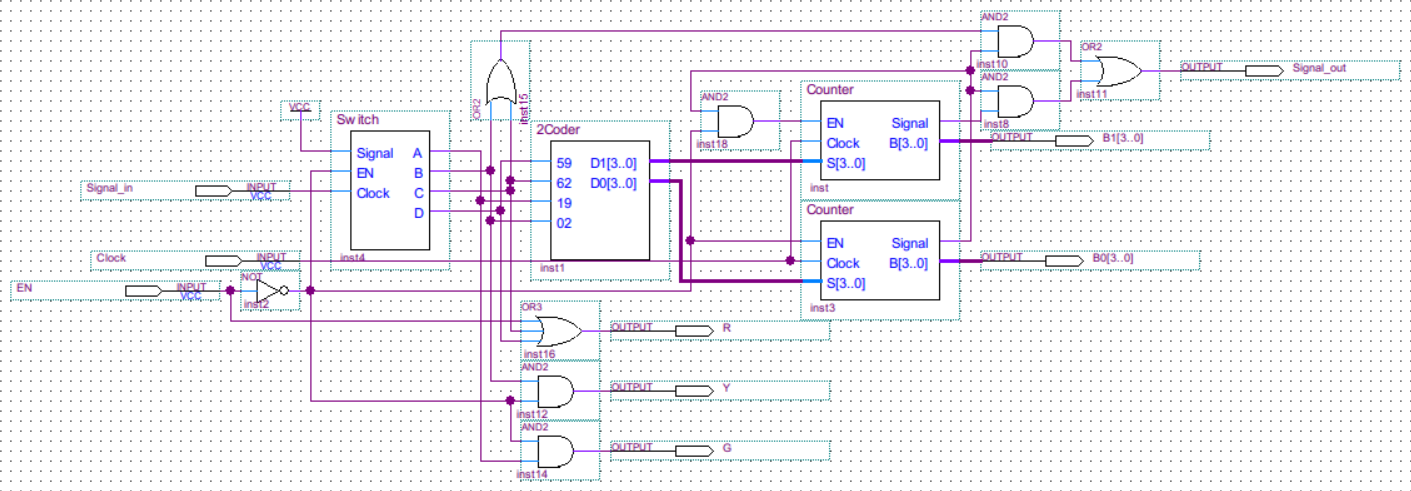


(主干道的数据选择模块，提供59、22、19、02四个数据)



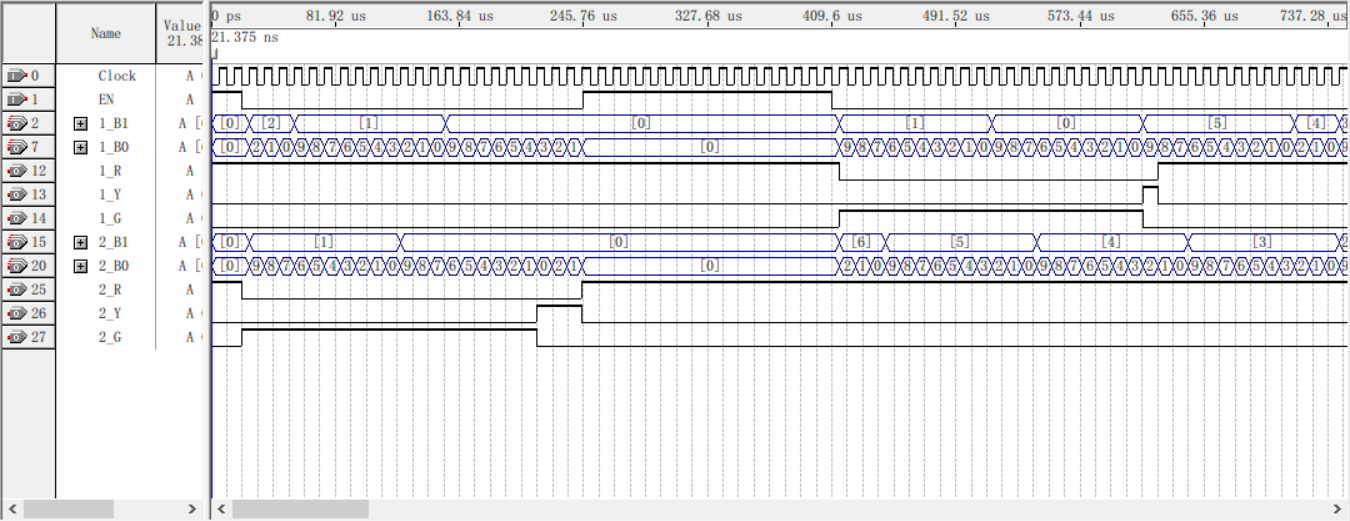
（乡间道的与基础部分相同，直接挪用）

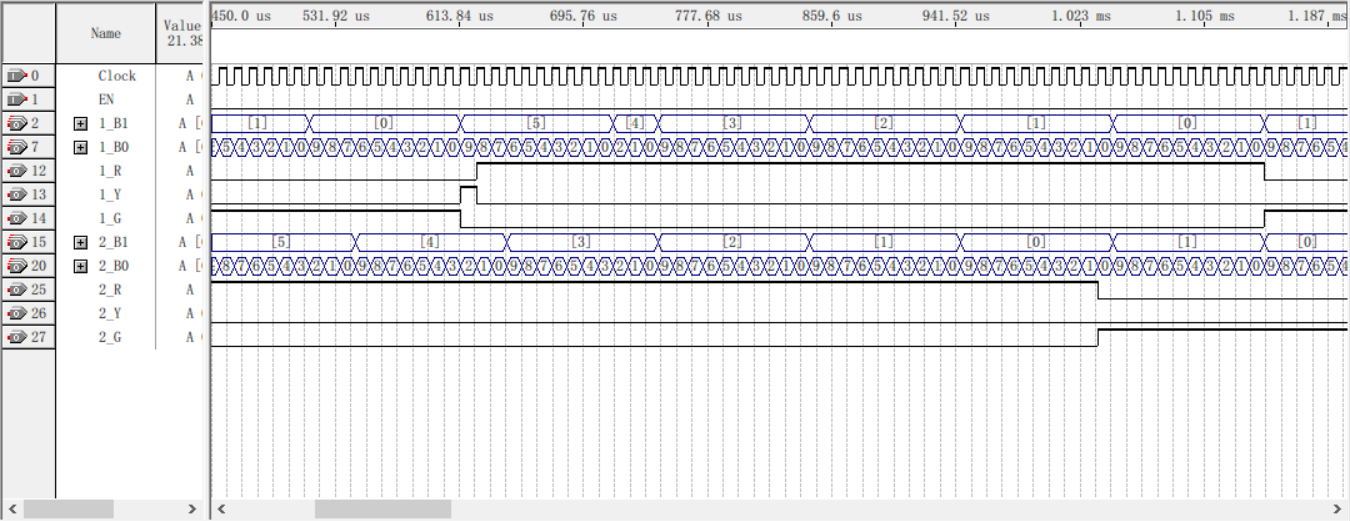


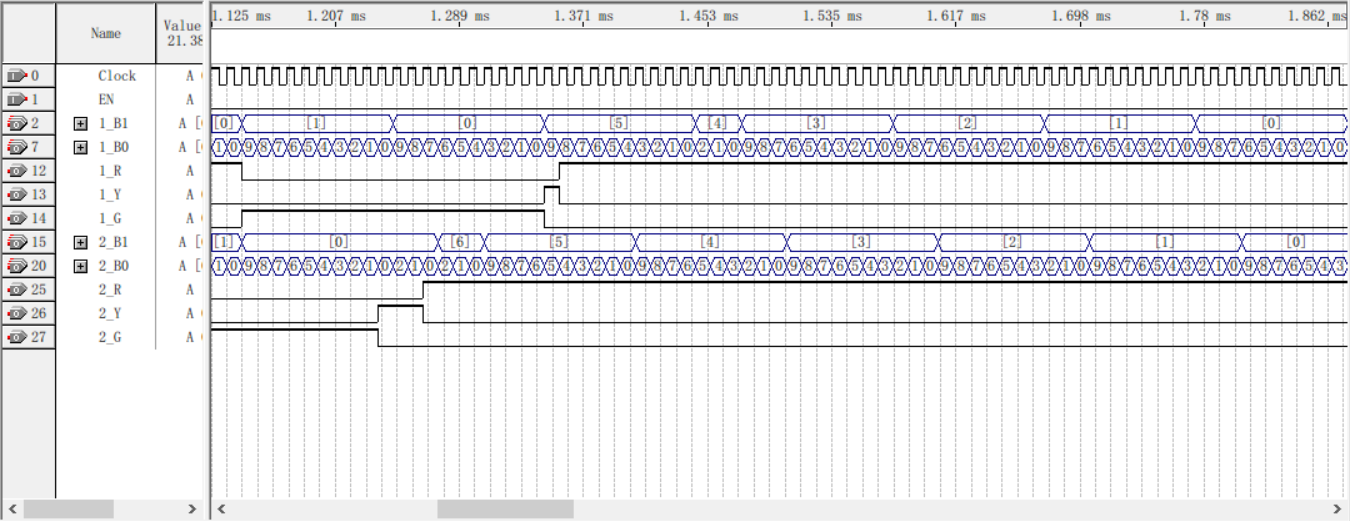


（两个路口的电路在连接方式上有改动）

1. **仿真结果：**







功能仿真运行结果计数部分正确，信号灯部分会提前一个时钟周期变化。但时序仿真会出现死循环的情况，或是状态提前跳变，导致计数与指示灯状态不匹配。

1. **下载到实验箱，连接时钟进行实物验证：**

在试验箱上运行时，会出现和时序仿真时一样的状况，在最初的一个周期正常，两次状态转换之后就会出现不正确的循环。

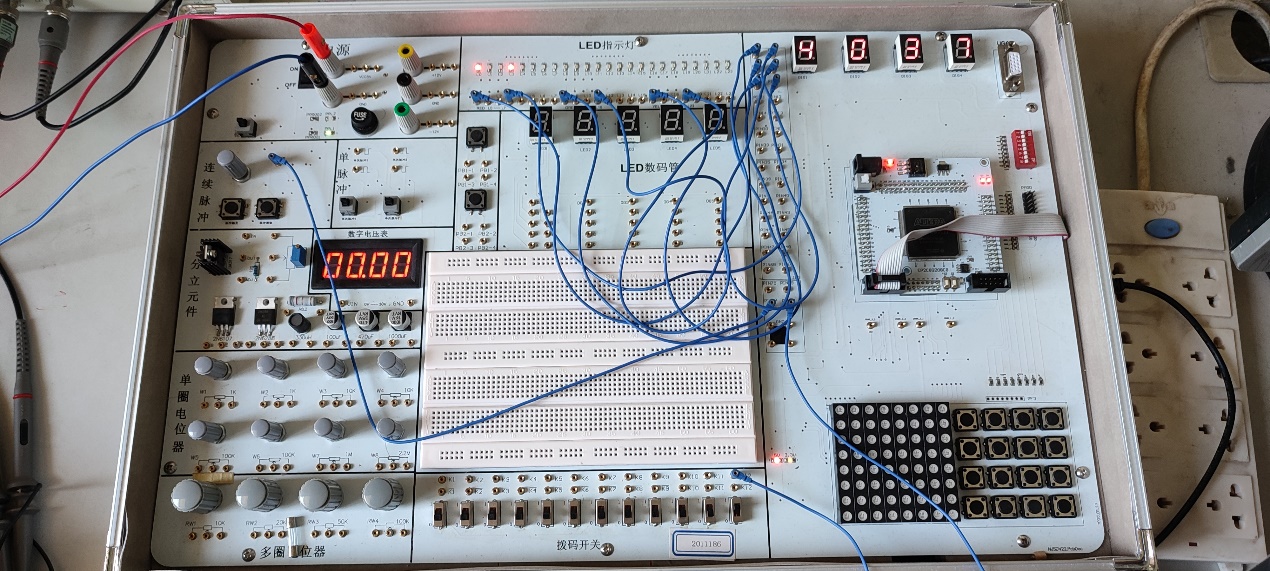
1. **出错原因分析：**

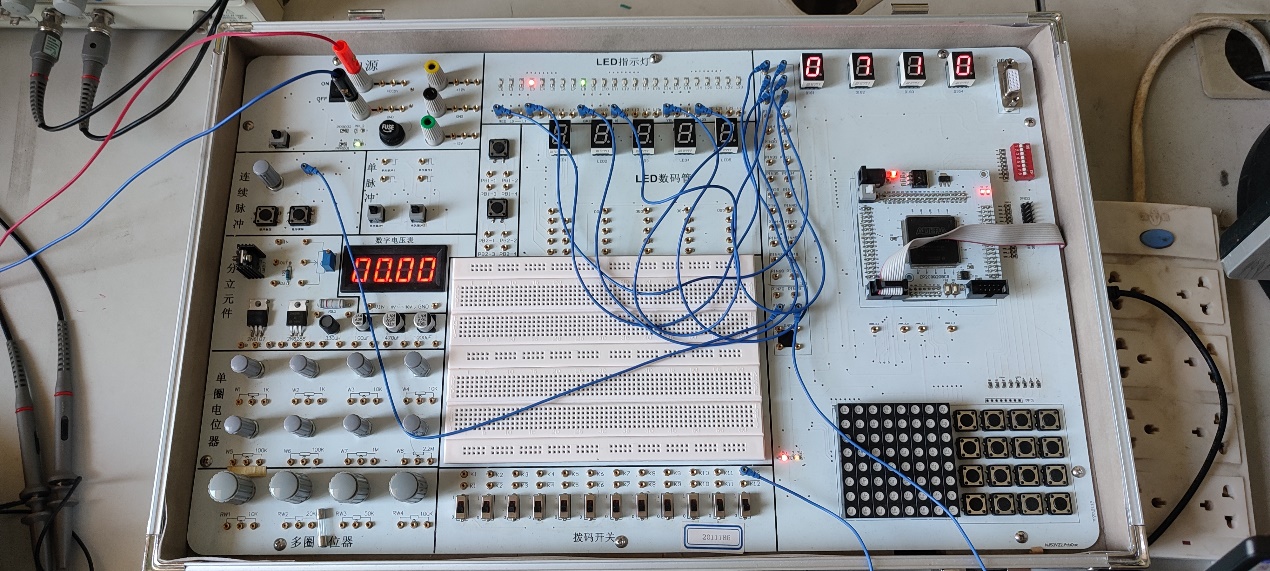
在状态切换电路的设计部分，是选用的异步的方式，即不连接时钟源，而是接受计数器的相应计数结束信号，当计数器完成一个计数周期后，输出一次高电平，状转换电路检测到信号的上升沿，完成一次状态转换。但实际运行时，或者时序仿真时，需要考虑到不同器件之间的延时情况，存在竞争与险象，在时序仿真结果中可观察到在计数器计数期间（未完成一个周期时），在计数器的输出端出现了尖峰，推测是由于这些尖峰导致状态转换提前进行，导致计数器发生错乱，以及信号指示灯亮起的时机不正确。

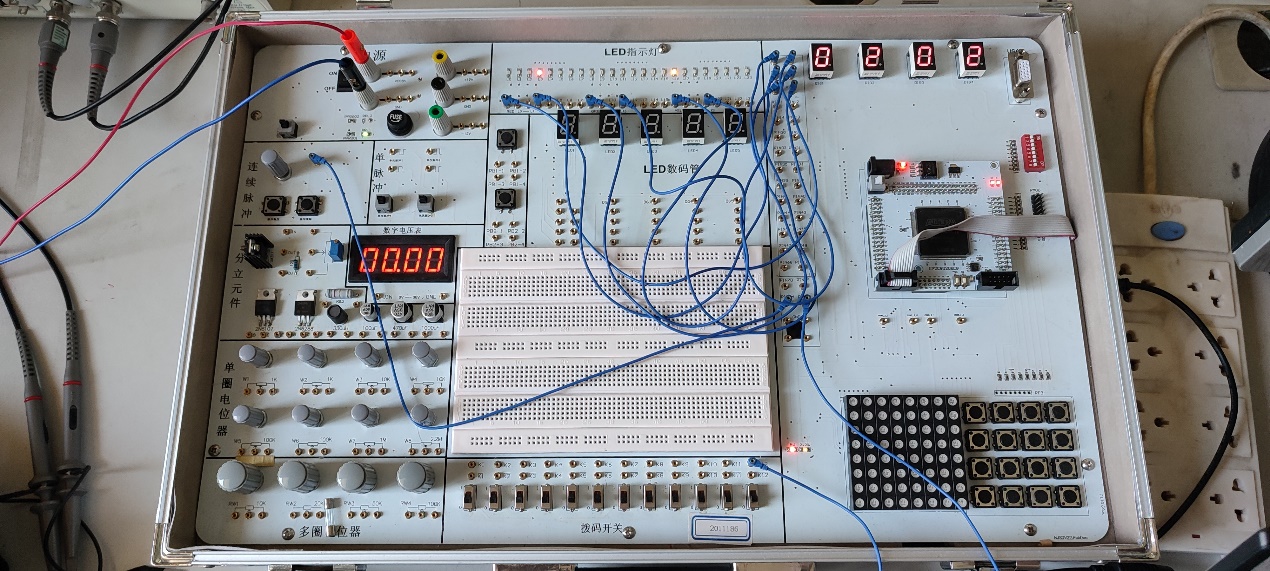
1. **实验仪器**

稳压电源、万用表、实验箱，FPGA: Cyclone II - EP2C8Q208C8.

1. **实验记录**

****

****

****