 图标

中度可信度描述已自动生成

SHANGHAI UNIVERSITY

2022-2023学年夏季学期

课程报告

《**计算机硬件综合大型作业报告**》

小组序号： 4-2

项目名称： 智能交通信号灯

指导老师： 张云华

组员学号姓名：

组长： 袁浩 21122453

组员： 谢怡雯 21122481

骆星宇 21122576

曹政 21122634

吴凡 21122635

计算机工程与科学学院

报告日期 2023年 7 月 3 日

1. **实践项目意义**

通过该项目，我们将深入了解硬件设计的流程和原理，将学习如何将概念转化为具体的电路设计，包括选择适当的元件和组件，并将它们集成到一个功能完善的系统中。这将提升我们的硬件设计能力，培养我对电路和元器件的理解和运用能力。

其次，参与团队协作和测试是项目中至关重要的一部分。团队成员紧密合作，共同制定项目计划、分工和时间表。在协作过程中，将学会有效沟通、协调资源和解决问题的能力。与团队成员一起调试和测试交通信号灯系统，将学习如何识别和解决实际工程中的问题，提升解决问题的能力。

最重要的是，交通信号灯硬件项目具有现实意义。交通安全一直是社会的重要问题，而交通信号灯作为交通管理的基础设施之一，对于减少事故、提升交通效率至关重要。通过参与该项目，我们将为社会交通安全做出贡献，并在未来的职业生涯中应用我们所学到的知识和经验，为构建更安全和智能的交通系统贡献自己的力量。

1. **实践项目原理**

通过分析可以知道，系统设计要求交通控制电路能够实现上述状态机的功能，并通过LED灯显示十字路口交通灯的亮灭情况以及通过计数器表示各个路口各个方向灯的倒计时情况，如此循环下去。因此设计系统流程框架如下图1所示

|  |
| --- |
| 图1 系统流程框架图 |

如上图所示，先由系统50Mhz产生1hz的分频信号，用于状态转移和计时，依据状态产生LED控制信号。

系统时钟和当前状态去决定控制数码管倒计时，当倒计时时间确定之后就可以产生各个方向的个位和十位信号，依据系统时钟产生位选信号对数码管进行显示，再通过段选信号去显示结果。

所设计的八个状态如下所示：

我们假设直行和右转的绿灯均为27s，黄灯均为3s。

一些文字和图片的手机截图

描述已自动生成

图2 系统状态转换图

**状态1**：南北方向直行绿灯27s，其余红灯。

此时南北方向的倒计时n\_tim、s\_time从27s绿灯开始，南北左转方向的倒计时nl\_time、sl\_time30s开始；东西方向直行倒计时e\_time、w\_time从60s开始，南北左转倒计时el\_time、wl\_time从90s开始。

当南北方直行时间n\_time、s\_time递减到1时，之后被赋值为3s黄灯(0s直接跳变为3s)，南北左转方向时间nl\_time、sl\_time变为红灯倒计时3s；东西方向直行时间e\_time、w\_time也被赋值为33s红灯，东西左转方向时间el\_time、wl\_time被赋值为63s红灯。

**状态2**：南北方向直行黄灯3s，其余红灯。

到达状态2的时候，南北直行方向倒计时n\_time、s\_time被重新赋值为3s，南北左转方向nl\_time、sl\_time为3s倒计时；东西方向直行红灯倒计时w\_time、e\_time被赋值为33s，东西左转方向红灯倒计时被赋值为63s。

当南北方直行黄灯时间倒计时到1s之后南北方向直行红灯被赋值为90s，南北左转方向赋值为27s绿灯；东西直行方向为30s红灯倒计时，东西左转方向为60s倒计时

**状态3**：南北方向左转绿灯27s，其余红灯。

到达状态3时，南北方向直行红灯倒计时为90s，南北左转方向绿灯为27s；东西方向直行红灯倒计时30s，东西方向左转红灯倒计时60s。

当南北左转方向绿灯时间递减到1时，之后被赋值为3s黄灯(0s直接跳变为3s)，东西直行方向为3s红灯倒计时，东西左转方向为33s红灯倒计时；南北直行方向为63s红灯倒计时。

**状态4**：南北方向左转黄灯3s，其余红灯。

到达状态4的时候，南北直行红灯倒计时为63s，南北左转黄灯倒计时3s；东西方向直行红灯倒计时3s，东西方向左转红灯倒计时33s。

当南北方向左转黄灯时间倒计时到1s之后南北左转红灯时间被赋值为90s(0s直接被赋值到90s)红灯，南北直行红灯倒计时60s；东西方向直行绿灯倒计时27s，东西方向左转红灯倒计时30s。

**状态5**：东西方向直行绿灯27s，其余红灯。

到达状态5时，东西方向直行绿灯27s，东西方向左转红灯倒计时30s；南北方向直行红灯倒计时60s，南北方向左转红灯倒计时90s。

当东西方向直行绿灯时间s\_time递减到1时，之后被赋值为3s黄灯(0s直接跳变为3s)，东西方向左转红灯倒计时3s；南北直行方向红灯倒计时33s；南北方向左转红灯倒计时63s。

**状态6**：东西方向直行黄灯3s，其余红灯。

到达状态6的时候，东西直行方向黄灯3s，东西方向左转红灯倒计时3s；南北方向直行红灯倒计时33s，南北方向左转红灯倒计时63s。

当东西方向直行黄灯时间倒计时到1s之后东西方向直行红灯时间被赋值为90s(0s直接被赋值到90s)红灯，东西方向左转绿灯为27s；南北方向直行红灯倒计时为30s，南北方向左转红灯倒计时为60s。

**状态7**：东西方向左转绿灯27s，其余红灯。

到达状态7时，东西方向左转绿灯为27s，东西方向直行红灯倒计时为90s；南北方向直行红灯倒计时为30s，左转方向红灯倒计时为60s。

当东西方向左转绿灯时间递减到1时，之后被赋值为3s黄灯(0s直接跳变为3s)，南北方向直行红灯倒计时为3s，南北方向左转红灯倒计时为33s；东西方向直行红灯倒计时为63s。

**状态8**：东西方向左转黄灯3s，其余红灯。

到达状态8的时候，东西方向左转黄灯3s倒计时，东西方向直行红灯倒计时为63s；南北方向直行红灯倒计时为3s，左转方向红灯倒计时为33s。

当东西方向左转黄灯时间倒计时到1s之后时间被赋值为90s(0s直接被赋值到90s)红灯，此时南北方向直行绿灯倒计时27s，左转红灯倒计时30s；东西直行方向红灯倒计时为60s。

1. **实践项目的设计**

根据设计要求和系统所要求得功能，进行了前期调研，查阅相关书籍和文献，确定本设计需要的所有器件以及模块，制定设计方案，现确定主要完成以下任务：

1. 系统时钟的设计

本设计需要提供一个50Mhz的系统时钟。

1. 控制电路的设计

本设计中需要复位功能及遇有紧急情况可人为将所有灯设为红灯，通过按钮，按照控制要求，控制key键相应输入的高低电平，继而通过程序的运行达到控制的目的。通过key按钮使所有直行灯变为黄灯。

1. 数码管显示模块的设计

本设计中需要将三种灯（红、黄、绿）的点亮时间通过数码管显示出来，由于各方向的灯的倒计时时间不相同，需要用8个数码管对各个方向进行显示。

1. LED灯显示模块设计

本设计中需要控制东西、南北两个方向的信号灯，一共24个灯，由于各个方向显示不相同，即需要控制4组信号灯。

1. **实践项目调试过程**

本部分主要做程序解释：

1. **state\_trans\_model.v**

交通灯控制模块将需要显示的时间数据连接到数码管显示模块，同时将状态信号连接到led灯控制模块。

输入信号为时钟信号sys\_clk,和系统复位信号sys\_rst\_n；

输出信号为4位state，用于控制四个方向上交通灯的亮、灭，10位的寄存器数据n\_time、e\_time、s\_time和w\_time存储北、东、南、西方向数码管要显示的时间数据。

|  |
| --- |
|  |

接下来设置基本的参数信息，比如黄灯的发光时长，红灯的发光时长，绿灯的发光时长等。WIDTH参数用于控制分频，产生1s的时钟，由于系统时钟是50Mhz，所以系统时钟的波形周期为20ns，之后基于这个20ns一个上升沿就可以计数产生分频信号：

|  |
| --- |
|  |

之后我们需要依据系统时钟产生1hz的时钟信号，我们先来产生0.5s的计数器(这里仿真变成了毫秒级)：

|  |
| --- |
|  |

其中WIDTH为25\_000\_000(仿真用25\_000为其系统频率的一半，只要计数器clk\_t小于WIDTH就不断累加，若达到这个值，或者复位信号来了，就归零，由此就产生了0.5s的计数器。依据这个0.5s的计数器，我们就可以产生1hz的时钟。

有了1hz的时钟之后就可以改变状态机，切换交通信号灯工作的8个状态。在开始状态切换之前有一个复位信号，状态复位到状态1，计时为27s。

|  |
| --- |
|  |

**状态1**：北方向绿灯亮27s，东方向红灯30s，南方向红灯60s，西方向红灯90s。当倒计时到1s时，之后被赋值为3s黄灯，此时东方向被赋值为3s但显示依旧是红灯，南方向是33s红灯，西方向是63s红灯。

|  |
| --- |
|  |

**状态2**：到达状态2的时候，北方向倒计时n\_time被重新赋值为2s，东方倒计时e\_time也被重新赋值为2s，南方倒计时s\_time被赋值为32s，西方倒计时被赋值为62s。

当北方时间倒计时到1s之后北方时间被赋值为90s(0s直接被赋值到90s)红灯，东方时间e\_time被赋值为27s绿灯，南方时间s\_time被赋值为为30s红灯，西方时间被赋值为60s红灯。

|  |
| --- |
|  |

**状态3**：东方向绿灯27s，其余红灯，此时led输出为24'b100\_100\_100\_100\_010\_100\_010\_100;。同理可得状态3的代码：

|  |
| --- |
|  |

**状态4**：东方向黄灯3s，其余红灯。

|  |
| --- |
|  |

**状态5**：南方向绿灯27s，其余红灯。

|  |
| --- |
|  |

**状态6**：南方向黄灯3s，其余红灯。

|  |
| --- |
|  |

**状态7**：西方向绿灯27s，其余红灯。

|  |
| --- |
|  |

**状态8**：西方向黄灯3s，其余红灯。

|  |
| --- |
|  |

经过上述分析可以知道：交通灯状态转移控制模块（state\_trans\_model）：交通灯控制模块是本次实验的核心代码，这个模块控制信号灯的状态转换，将实时的状态信号state[3:0]输出给led灯控制模块（led\_module）,同时将北、东、南、西方向的实时时间数据n\_time[9:0]、e\_time[9:0]、s\_time[9:0]和w\_time[9:0]输出给数码管显示模块(bit\_seg\_module)。然后数码管显示模块和led灯控制模块驱动交通信号灯外设工作。

1. **led\_module.v**

在这个模块里主要是接收状态信息然后使北、东、南、西方向的红黄绿LED灯使能。所以输入就是系统时钟，复位信号和交通灯的状态等，输出是led灯的使能信号。

|  |
| --- |
|  |

led寄存器从高到低分别驱动：北、东、南、西向**红绿黄**灯。

整个实现为：

|  |
| --- |
|  |

led灯控制模块（led\_module）：根据接收到的实时状态信号state[3:0]，驱动北、东、南、西方向的led发光。

1. **bit\_seg\_module.v**

数码管显示模块（bit\_seg\_module）：接收交通灯控制模块传递过来的北、东、南、西方向的实时时间数据n\_time[9:0]、e\_time[9:0]、s\_time[9:0]和w\_time[9:0]，并以此驱动对应的数码管，将数据显示出来。主要就是产生南北方向和东西方向倒计时的十位和个位的8421信号来控制数码管显示相应的数字。

所以数码管显示模块的输入为系统时钟信号，系统复位信号，东西方向数码管要显示的数值，南北方向数码管要显示的数值。输出就是南北方向和东西方向倒计时的十位和个位的8421信号。

|  |
| --- |
|  |

因为要显示数据，所以需要计算出十位和个位数据先：

|  |
| --- |
|  |

设置0.1ms的计数来切换数码管点亮的四个状态：

|  |
| --- |
|  |

之后就按照北、东、南、西方向数码管的十位和个位数值去驱动数码管：

|  |
| --- |
|  |

之后依据数字num去驱动数码管显示：

|  |
| --- |
|  |

除此之外还写了顶层模块traffic\_led.v和测试文件test.v。

1. **仿真结果**
2. **系统时钟显示**：

|  |
| --- |
| 图3 系统时钟仿真结果 |

1. **状态跳变以及倒计时过程**：

|  |
| --- |
| 图4 状态1跳变到状态3过程图 |

如上图所示倒计时北方向倒计时27s之后能够顺利转换到黄灯3s。Led的输出信号也与之前的状态分析的led输出信号一致对应。

|  |
| --- |
| 图5 状态3跳变到状态4、5、6、7仿真结果图 |

之后看一下状态8回到状态1的仿真结果：

|  |
| --- |
| 图6 状态8跳变到状态1仿真结果图 |

可以看到整个状态转移以及倒计时都非常正确。

1. **复位和紧急情况按钮**：

|  |
| --- |
| 图7 复位信号以及紧急情况仿真结果测试图 |

从仿真结果可以看到，复位之后能够从北方向重新开始倒计时，led输出第二位为北方向绿灯。Key变为0之后为遇到紧急情况，交通灯则变为全绿灯。

1. **紧急情况通行**

|  |
| --- |
| 图8 南北方向通行结果测试图 |
| 图9 东西方向通行结果测试图 |

从结果可以看到，当Key为1110时，只有南北方向通行，其他全为红灯；当Key为1101时，只有东西方向通行，其他全为红灯。

1. **夜间模式**

|  |
| --- |
| 图10 夜间模式仿真结果测试图 |

从结果可以看到，当Key为1011时，只有黄灯工作，处于夜间模式；

1. **位选信号和段选信号**

|  |
| --- |
| 图11 复位选扫描信号和段选信号仿真结果测试图 |

1. **系统整体仿真结果**

|  |
| --- |
| 图12 系统整体仿真结果 |

1. **大型作业的心得与收获**

**袁浩：**

在这次项目中，我负责了LED状态机的编写以及与团队成员一起进行协调测试的任务。通过这一经历，我积累了许多宝贵的经验和收获，不仅在代码逻辑编写方面有所提升，也在团队合作和协调测试的能力上有所成长。

首先，LED状态机的编写对于整个系统的功能实现和状态控制起着至关重要的作用。LED的不同状态和模式在产品展示和交互中起着关键的角色。在设计状态机时，我注重了代码的可读性和可维护性。我使用了清晰的命名和注释，结构化的代码组织方式，以及合适的设计模式，以便于团队成员理解和修改。我将状态机分解为八个不同的状态，并定义了状态之间的转换条件和动作。通过这样的设计，我成功实现了预期的LED状态和模式切换。

其次，在与团队成员一起进行协调测试的过程中，我学到了许多宝贵的经验。团队合作和协调是确保系统功能和质量的关键。我们密切配合，共同制定测试计划和方案，并积极解决出现的问题。我积极参与团队讨论，与其他成员分享我的代码逻辑和设计思路，也从其他成员那里学到了许多有益的建议和优化技巧。通过合作测试，我们及时发现了一些状态机中的逻辑错误和边界条件问题，并通过迭代和调试解决了这些问题。

在协调测试的过程中，我注意到了一些常见的问题，例如状态转换不正确、状态持续时间错误或动作执行异常等。通过仔细的观察和分析，我发现这些问题可能是由于状态机设计不完善、状态转换条件错误或动作代码有误等原因引起的。我积极与团队成员合作，一起进行代码审查和调试，以定位和解决这些问题。这个过程不仅提高了我对状态机设计和代码逻辑的理解，也锻炼了我的问题解决能力和调试技巧。

通过这次项目，我深刻体会到了LED状态机编写和团队协调测试的重要性和挑战。我学会了如何将复杂的功能拆分为可管理的状态和转换条件，以及如何与团队成员合作并有效解决问题。这次经历不仅丰富了我的编程技能，也提高了我的团队合作和沟通能力。我相信这些经验和收获将对我未来的项目和职业发展产生积极的影响。作为一名开发人员，我将继续保持对代码逻辑编写和团队协调测试的热情，并不断学习和提升自己。我希望在未来的项目中能够更加深入地理解状态机设计和代码优化，并能够运用新的技术和方法来改进产品的性能和用户体验。我也将继续加强与团队成员的合作，共同追求卓越的项目成果。通过持续的努力和学习，我相信我可以在LED模块设计和团队协调测试领域取得更大的成就。总而言之，通过参与LED和数码管设计项目，我获得了宝贵的经验和收获。这次经历不仅提升了我的技术能力，也锻炼了我的团队合作和问题解决能力。

**谢怡雯：**

在这次夏季硬件大作业中，我们小组负责的是十字路口交通信号灯系统的设计。我在其中主要是负责控制器模块的代码编写以及与团队小组成员一起进行协调测试。在这一次完成项目的过程中，我学会了许多关于硬件设计和系统设计及开发方面相关的知识，相对于以前，在开发思维上有了大幅度的提升。以下是我在这次学习经历中，所收获的知识和经验以及总结。

首先，控制器模块在整个十字路口交通信号灯系统中有着举足轻重的地位，它关系着整个系统如何执行的大轮廓。在设计之初，我最先去考察并了解了十字路口信号灯改变的情况，包括直行与左转的两种情况分别在东西向马路和南北向马路，以及包含了红绿黄三种灯的改变。例如最开始选择的南北向直行绿灯，在绿灯过后进行了三秒黄灯之后就变为南北向左转绿灯，以此类推，将状态分为了八种状况，并用state变量进行了记录。

其次，我深入的了解了vorilog编程语言，并用其完成了控制器模块的编写。在写代码的过程中，在每个重点代码旁我都进行了注释，并且注重了代码格式，以此增加代码的可读性来方便组员间相互读代码。

接着，在进行上机调试并与团队组员进行协调交流时，我也从中收获了很多宝贵的经验。在一开始我们就碰到了一个问题，我们所属用quartus版本与机房的实验箱不相匹配，因此无法将我们的代码下载到实验箱上。小组成员们一起集思广益，进行了讨论和一次次的尝试后也是顺利解决了这个问题。但是在代码顺利下载到上面之后，又出现了一个新的问题，数码管上的数字并未改变。开始的时候，我们以为是实验箱本身的设备问题，但在经过了一系列问题的排查之后，我们发现了是实验箱上的时钟频率与我们的设置的频率有区别的原因，并即使的对其进行调整。最后在问题都顺利解决后，我们的十字路口交通信号灯系统也是顺利的执行实现了。

在本次的整个项目中，我深刻地体会到了团队之间合作与沟通的重要性。最开始在讨论了大致设计思想之后，每个人就是主要完成自己的部分，但是作为一个系统项目设计，各个部分又是相互联系的，因此在一些代码的设计，变量的命名等时，都会规范格式，用尽量清晰的命名以及一些注释，让代码更具有可读性，并且还会提前与组内成员进行清晰的沟通，以免两个人某些方面想法不同而到最后做了无用功。

并且我还熟练掌握了QuartusⅡ软件的使用，以及对vorilog编程语言的深入学习。QuartusⅡ是一个非常实用的可视化工具，提供了丰富的功能和工具，用于设计、模拟、优化和实现电路。也感谢这次大作业给我的许多宝贵经验，让我能运用贯穿于未来的学习或者工作中。

**骆星宇：**

在这次夏季硬件大作业中，我们小组负责十字路口交通灯系统的设计。我在其中负责系统时钟的设计和波形图的绘制以及traffic\_led.v的编写。通过这个项目，我学到了许多关于硬件设计和系统开发方面的知识和技能。以下是我对这次学习经历的小结，总结了我在这些方面的收获和经验。

首先，时钟设计是整个系统的关键部分之一。一个稳定可靠的时钟信号对于协调各个模块的工作至关重要。我学会了选择合适的时钟源，并设计时钟分频器来生成系统需要的各种时序信号。在机箱上只有1Mhz,100khz等时钟频率，而我们需要1hz的时钟来进行交通灯状态的转化和黄灯的闪烁，故进行时钟信号的转化十分重要。一开始，我们小组就是由于时钟信号的转化设计不当，导致所有的灯的状态均不发生变化，甚至怀疑是否是硬件问题。最终我们小组设计出了三个时钟，一个用于监测交通灯状态的变化，一个用于进行黄灯的闪烁，一个用于数码管显示的刷新间隔，并且很好的进行了代码的复用，只需要修改state\_count一个参数就可以完成不同频率时钟信号的生成。

其次，我还学习并绘制波形图来验证和调试我们小组的设计。波形图是一种重要的工具，可以帮助我理解系统中不同信号的时序关系，并发现潜在的问题和bug。波形图可以捕获和分析系统中各个模块的信号波形，并与我们小组设计预期的波形进行比较。是验证设计正确性和debug的重要工具，在能够去学院机房进行实际操作之前，波形图的正确性是我们判断我们小组设计是否正确的唯一标准。对于其它组员编写的module，我都使用波形图对其进行了分析，并与其讨论哪里出现了与预期不符的波形，最终编写出正确的代码。也正因如此，我对每个模块的代码都很熟悉，最终负责了traffic\_led.v这一总体调度模块的编写。这样的调试过程使我更加熟悉了硬件调试的方法和技巧，并提高了我的问题解决能力。

在整个项目中，我也学到了团队合作和沟通的重要性。虽然我在时钟设计和波形图绘制方面由我负责，但我也需要与其他团队成员密切合作，以确保我们的设计能够无缝地集成在一起。最终我们所有的成员一同进行了代码的调试以及硬件电路的链接，将预期的设计在硬件层面成功实现。通过与其他成员的合作，我学会了如何有效地分工合作、协调沟通，并互相借鉴和补充彼此的专业知识。

同时，我此次对于QuartusII 软件的使用也更加的熟悉，Quartus II提供了丰富的功能和工具，可以满足各种复杂的FPGA设计需求。它支持多种编程语言（如VHDL和Verilog）和设计方法，具备高级综合、逻辑综合、时序分析等功能，可以高效地进行设计、验证和优化，Quartus II提供了多种调试和验证工具，如逻辑分析器、波形查看器、时序分析器等，帮助设计师分析和验证设计的正确性和性能。这些工具可以帮助设计师快速定位和解决问题，并优化设计的功能和性能。

总之，这次夏季硬件大作业给了我宝贵的学习机会，让我在时钟设计和波形图绘制等方面获得了实践经验。通过这个项目，我不仅加深了对硬件设计原理和方法的理解，还培养了解决问题和团队合作的能力。这将对我未来的学习和职业发展产生积极的影响，我期待能够将所学的知识和经验应用到更多的实际项目中，不断提升自己的技能水平。

**曹政：**

在这次夏季硬件大型作业中，我参与了十字路口交通灯设计的实验。通过这次实验，我获得了许多宝贵的经验和收获，以下是我对本次实验的小结和感想：

首先，这次实验让我深入了解了数字逻辑电路设计的原理和方法。在设计和绘制电路图的过程中，我学会了如何选择适当的逻辑门和触发器等元件，并进行合理的逻辑关系和时序控制。通过对数字逻辑电路的设计和实现，我对数字逻辑电路的原理和工作方式有了更深刻的理解。

其次，通过使用Quartus软件进行电路图设计和模拟，我掌握了一种强大的工具。Quartus软件提供了丰富的功能和模拟环境，使得电路图的设计和模拟变得更加简洁和高效。通过对电路图进行模拟，我能够对电路的正确性和稳定性进行验证，同时也可以在模拟中发现并解决一些问题。

本次实验我负责的是测试文件的编写，用于控制交通灯的状态和显示。先是定义了一些输入和输出的wire信号，包括四个方向的状态剩余时间数据（n\_time, e\_time, s\_time, w\_time）、灯光状态（state），以及一些其他信号（nl\_time, el\_time, sl\_time, wl\_time）。然后定义了一些初始化的信号，包括时钟（sys\_clk）、复位信号（sys\_rst\_n）和按键输入（key）。并且实例化了一个状态转换模块（u0\_state\_trans\_model），用于根据倒计时时间和输入信号来更新交通灯的状态。实例化了一个数码管显示模块（u1\_bit\_seg\_module），根据交通灯的状态和计时信号来显示剩余时间。实例化了一个LED灯控制模块（u2\_led\_module），根据交通灯的状态来控制LED灯的点亮。整个模块的功能是根据倒计时时间和输入信号来控制交通灯的状态，并将状态以LED灯和数码管的形式进行显示。

接着，在进行实验箱接线和调试的过程中，我锻炼了动手实践和解决问题的能力。这个过程需要仔细对照电路图进行接线，并确保每个元件的连接准确无误。在调试的过程中，我发现了一些问题，如信号冲突和时序错误等，通过不断地排查和调整，我成功地解决了这些问题，并使交通灯系统正常工作。

最后，这次实验让我体验到了团队合作的重要性。在实验中，我与队友共同协作，分工合作，共同完成了这个项目。通过与队友的合作，我学到了沟通、协调和分工合作的重要性，也增强了我的团队合作能力。

总的来说，这次十字路口交通灯设计的实验为我提供了一个很好的学习机会。通过实践，我不仅深入了解了数字逻辑电路设计的原理和方法，还获得了使用Quartus软件进行电路设计和模拟的技能。我还通过实验箱接线和调试的过程锻炼了动手能力，并体验到了团队合作的重要性。这次实验为我今后的学习和工作打下了坚实的基础，让我更加自信和深入地探索计算机硬件领域的知识。

**吴凡：**

在这次夏季硬件大作业中，我们小组负责十字路口的红绿灯设计，我们通过使用Verilog系统进行编码，并成功在实验箱中调试运行。通过此次实验，我收获了许多计算机硬件设计、编程等实验的经验，并且更加深入理解了“数字逻辑”、“计算机组成原理”两门课上我所学到的知识。下面是我个人对本次实验项目的小结，详细记录了我在本次活动中获得的经验与收获，我主要负责的是代码总体逻辑框架的设计以及数码管显示的代码的编写，故接下来我将从这两个方面来记录。

首先，我对交通灯的运行方式进行了思考，确定了若想要用硬件实现这一功能，需要定义灯亮模块和数码管显示模块。并且两个模块要有一定的转换关系。这也就是我们代码中的bit\_seg和light模块，并且需要一个状态机来控制两者的转换，然后进行系统时钟的设计，确保控制二者的转换是准确的。这样之后我们小组的研究就有了一个清晰的方向，大家共同根据框架进行具体代码的设计，最后成功实现其功能。

其次，我进行了数码管模块的具体设计，也就是bit\_seg module的设计。数码管是本次实验任务的关键一环，数码管设计的正确代表了整个实验电路设计的正确性，其与交通灯之间的复杂的转换是本次实验的难点，并且如果数码管的设计正确，可以通过数码管的示数来进行代码的各种调试等等。因此，我详细地进行了交通灯所可能涉及的8个状态的数码管示数计算并进行模拟，最终得出了8个状态之间循环转换状态的转移方程式，将其体现在代码中，并且根据时钟的设计来模拟每一秒数码管示数的变化过程，从而实现了最终的功能。我在实验中将代码简单化，并且为了提高其可读性特意加了许多的注释，让同一小组的其他成员能够清晰地理解我的思路，并且方便他们的设计。

此外，中间也碰到了许多的问题，比如实验箱的设备不适配我们Quartus II的版本、接线管损坏导致功能无法完全实现、下载到实验箱上之后程序明明是正确的但无法在实验箱上体现的问题，我们群策群力将其一一解决，最终近乎完美地完成了实验的任务，极大地锻炼了我的问题解决能力和思考能力，并让我对Quartus II软件的运用也更加熟悉，巩固并提高了我在两门硬件实验课中学习到的能力。其提供了丰富的功能和工具，满足了我们复杂的FPGA设计需求。并且我们熟练运用了其提供的多种调试和验证工具，如逻辑分析器、波形查看器、时序分析器等，帮助我们分析和验证设计的正确性和性能。

在本次实验中，我极大地锻炼了团队合作交流的能力，我们小组每一个人都发挥了他们的特长，并且起到了至关重要的作用。虽然总体设计和数码管功能方面由我负责，但我也需要与其他团队成员密切合作，以确保我们的设计能够无缝地集成在一起。最终我们所有的成员一同进行了代码的调试以及硬件电路的链接，将预期的设计在硬件层面成功实现。通过与其他成员的合作，我学会了如何有效地分工合作，并互相借鉴和补充彼此的专业知识。

总之，这次硬件大作业给了我宝贵的学习与实践机会，让我在FPGA功能的设计方面积累了更多的经验。期待我在这个项目中的所学所获能够在之后的学习、工作和生活中有所裨益，并且希望学校能够多多开展像这样的实践教学项目，让同学们锻炼自主研究问题的能力，同时巩固并提高相关的计算机科学知识。