

数字逻辑与数字系统设计

实验大作业报告

(2020 年)

课程名称：数字逻辑与数字系统设计

任课教师：张彦航

作业题目：电梯控制器设计

完成人：冯开来 涂鑫泊

学号：1190201215 1190201223

班级：1903007 1903007

报告日期：2020 年 12 月 13 日

报告成绩	
教师评语	

目 录

1. 设计要求	3
1.1 实验目的	3
1.2 实验要求	3
1.2.1 基本要求	3
1.2.2 附加功能	3
1.3 实验内容	3
1.4 实验环境	3
2. 工作原理及系统方框图	4
2.1 工作原理	4
2.2 系统工作流程图	4
2.3 系统逻辑图	5
3. 各部分模块具体功能及设计思路	5
3.1 综合模块	5
3.2 楼层输入模块	6
3.3 楼层比较模块	6
3.4 电梯移动模块	7
3.5 时钟模块	7
3.6 楼层控制模块	8
3.7 计时模块	8
3.8 楼层显示模块	9
4. 调试过程	9
4.1 问题一	9
4.2 问题二	10
5. 设计结论	10
5.1 功能总结	10
5.2 流程图	11
5.3 完善与反思	11
6. 设计心得与总结	11
6.1 成员 1	11
6.2 成员 2	11
参考文献	12
附录	12
附录 1	12
附录 2	13
附录 3	14

报告正文

实验题目	电梯控制器的设计		实验难度	★★★（3分）
成员	姓名	班号	学号	
1	冯开来	1903007	1190201215	
2	涂鑫泊	1903007	1190201223	
1.设计要求				
1.1 实验目的				
● 灵活运用知识实现一个电梯控制器				
1.2 实验要求:				
1.2.1基本要求 <ul style="list-style-type: none">● 电梯最少可以往返于0-9层楼.● 乘客要去的楼层数A可手动输入并显示,按取消键可清除本次输入● 可自动显示电梯运行的楼层数B● 当A>B时,电梯上升;● 可以自动显示电梯每一次启停之间的运行时间● 任何时候按下复位键,电梯回到1层				
1.2.2附加功能 <ul style="list-style-type: none">● 允许多次输入楼层,并进行存储● 拥有正确的优先级,正确处理上楼下楼顺序● 显示了上行提示,下行提示● 显示了开门提示,关门提示● 能够延长开门时间和关门时间● 避免非法输入				
1.3 实验内容:				
综合运用计数器、触发器、译码电路等方面的知识，完成从设计、选片、连线、调试、排除故障到实现一个数字系统的全过程，具体包括： <ul style="list-style-type: none">1、查阅相关资料，写出设计过程，说明系统的工作原理2、完成数字系统电路原理图的设计3、给出仿真实验结果4、完成实验演示				
1.4 实验环境:				
Vivado 集成开发环境 数字逻辑设计实验平台 EGO1 开发板				

2.工作原理及系统方框图

2.1 工作原理:

1) 楼层与电梯上下行、电梯在每一次启停之间的运行时间以及开关门状态显示

电梯刚开始处于 1 层状态,通过设定楼层的二进制开关来确定目标楼层,之后按下确认键(S1)电梯便开始运行,此时显示器第一位(DK1)显示的为当前所在楼层,第二位(DK2)为最后一次输入的目标楼层,第三位和第四位显示的是当前电梯运行时间(其中第三位为十位第四位为个位),当电梯处于关门且上升状态时,LD1 中的 0 位灯管常亮,当电梯处于开门状态时,LD2 中的 4 位灯管常亮,当电梯处于关门且下降状态时,LD1 中的 1 位灯管常亮。

2) 楼层数的输入与取消

当电梯运行时,按下取消键(S3)即可清除本次的输入并且电梯会停止在当前所显示的所在的那一层。同时我们还多加了一个可以多次输入的附加功能,即可以在九层时,输入 1,2,4 层,这时候电梯会在 1,2,4 层分别停留,并且当按下取消键(S3)时,便可以取消最后一次的层数输入,并且在误触取消键时,还可以再次点击取消键来重新得到最后一次的输入层数。

3) 电梯的复位功能

任何时候按下复位键(RESET),即使已经在别的楼层且有目标楼层,电梯也会立即回到一层。

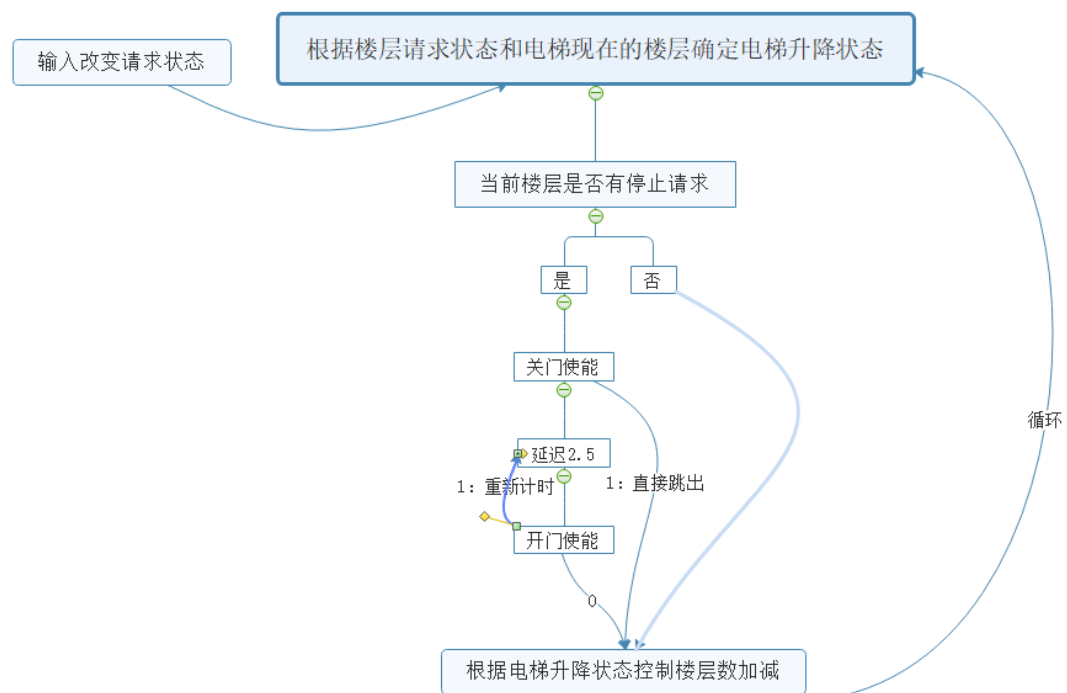
4) 电梯拥有正确的优先级

电梯拥有正确的优先级,即上升的时候会先上升到最高的楼层请求,再执行下降的请求,避免出现电梯在楼层间反复的辗转导致最高层乘客一直无法到达的情况。

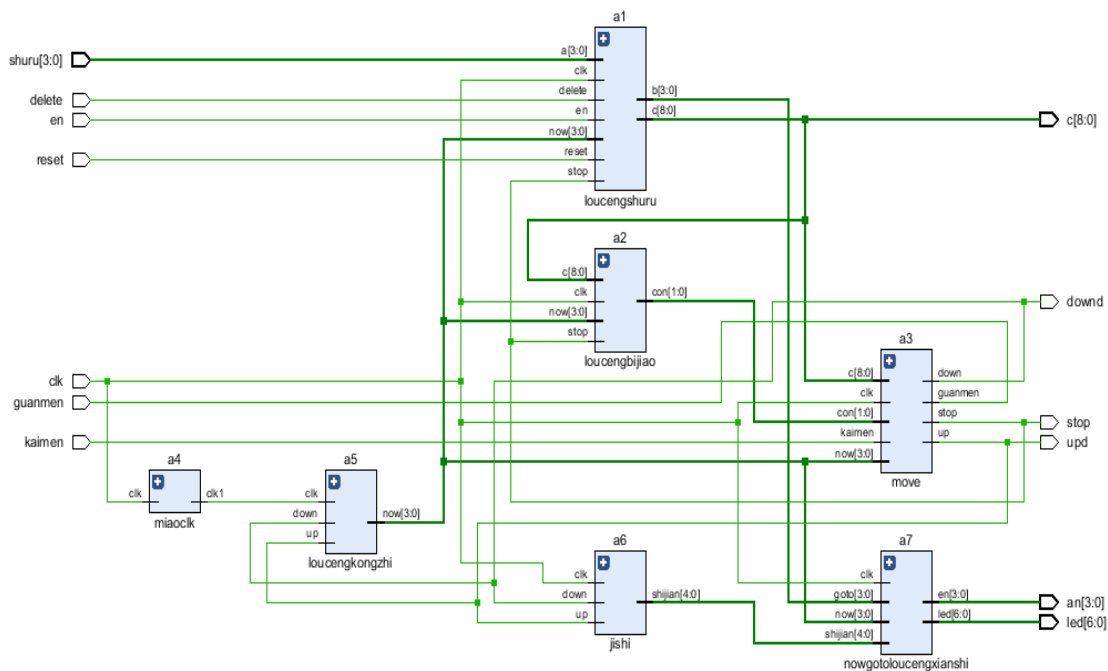
5) 能够延长或缩短电梯的开门时间(实际生活中的开关门按钮)

当电梯处于开门状态时,按延长开门键(S0)便可以延长开门时间,按关门键(S4)便可以立即关门。

2.2 系统工作流程图:



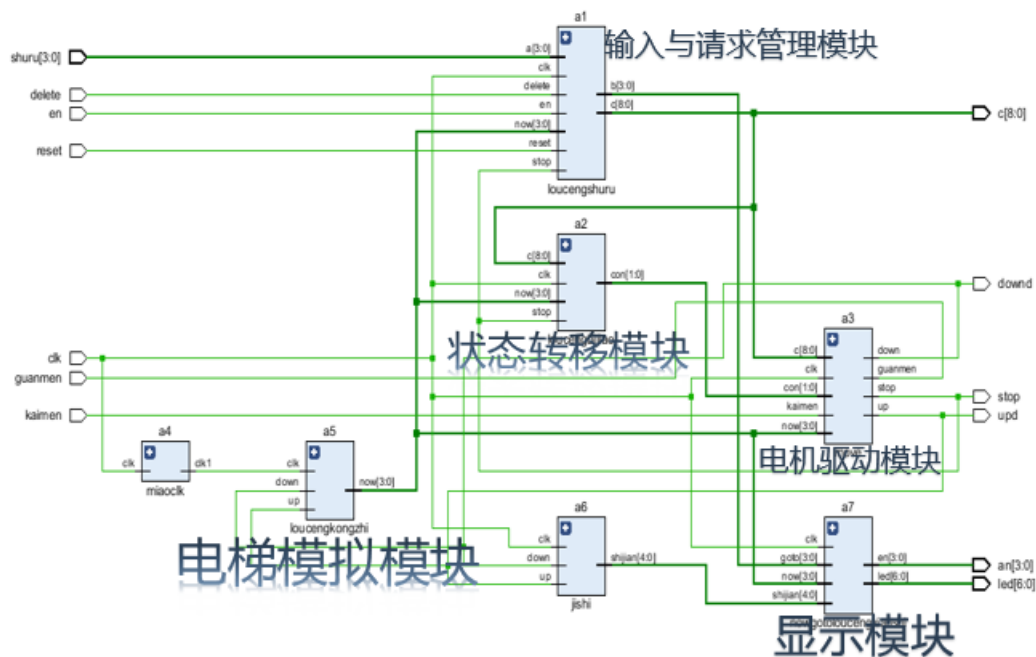
2.3 系统逻辑图



3.各部分模块具体功能及设计思路

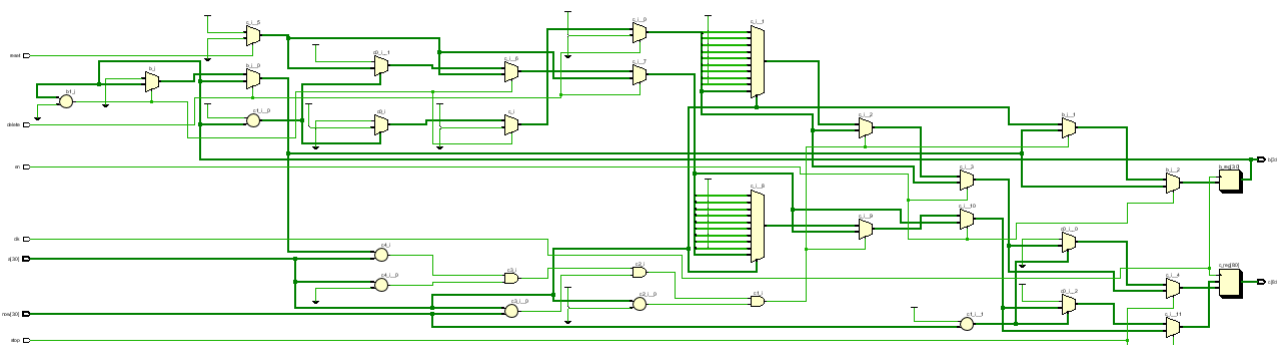
3.1 综合模块

输入目标楼层数并且按下确认键，综合模块便可以依次调用楼层输入模块、楼层比较模块、电梯移动模块、时钟模块、楼层控制模块、计时模块以及楼层显示模块，各模块会分别输出对应功能的结果。具体图示如下



3.2 楼层输入模块

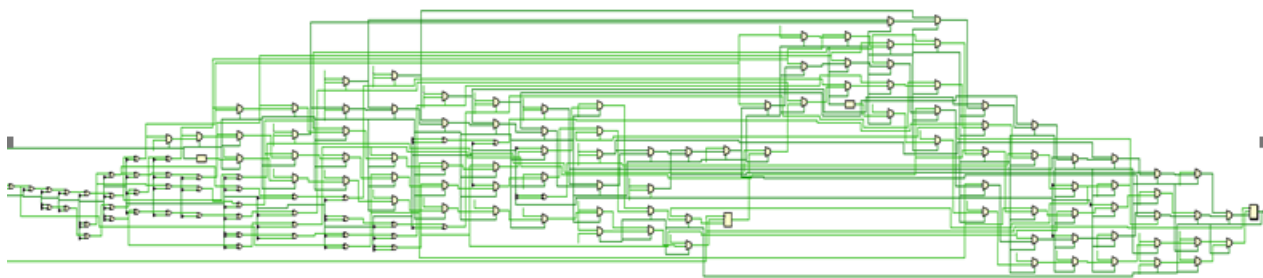
具体逻辑图如下：



输入目标楼层数、当前楼层数、删除键、时钟端、暂停键以及重置归一键。输出为对应的四位二进制数和对应各楼层是否需要停止的九位二进制数。其中九位分别对应 1~9 层，若对应位置为 1 则表示该楼层需要停止，为 0 则表示该楼层不需要停止。具体操作为当按下确认键时，将输入的四位二进制楼层数 a 赋值给 b ，将 b 进行 case 判定，若为 0001 则对应 $c【0】$ 为 1，若为 0010 则对应 $c【1】$ 为 1，以此类推，最后得到一个九位二进制数 c 来表示各楼层是否有请求。当按下删除键时，把刚才输入的楼层数由 1 变作 0，以实现取消最近一次输入的功能。当按下停止键时，另下一层（分上升和下降情况）的值变为 1，便可让电梯立即停止。当按下置 1 键时，重置 c 为 1，即让九位二进制数 c 的 0 号位（也就是表示第一层楼状态的值）变为 1，其余的全为零，以实现按下置 1 键便可以让电梯立即回到 1 层的功能

3.3 楼层比较模块

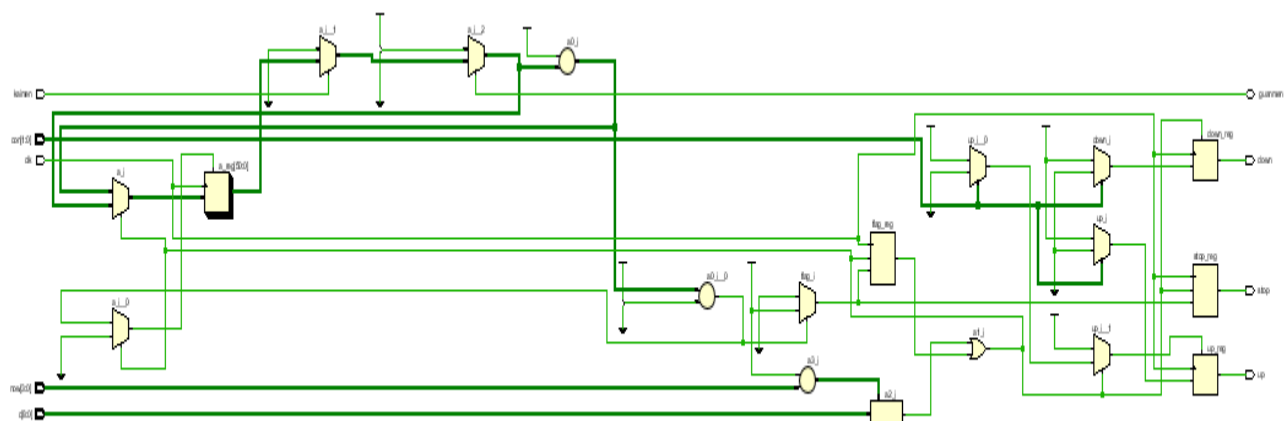
具体逻辑图如下



其中输入为当前电梯的请求状态（也就是九位二进制数），当前电梯所处楼层以及时钟端和暂停键。输出为电梯的状态（上升下降或者静止）。当电梯处于上升态时，判断当前楼层以上的楼层，若有请求，则电梯继续上升，输出上升，没有则输出为静止。当电梯处于下降态时，判断当前楼层以下的楼层，若有请求，则电梯继续下降，输出为下降，没有则为静止。当电梯处于静止态时，首先判断其上有无请求，若有则输出为上升，再判断其下有没有请求，若有则输出为下降状态。

3.4 电梯移动模块

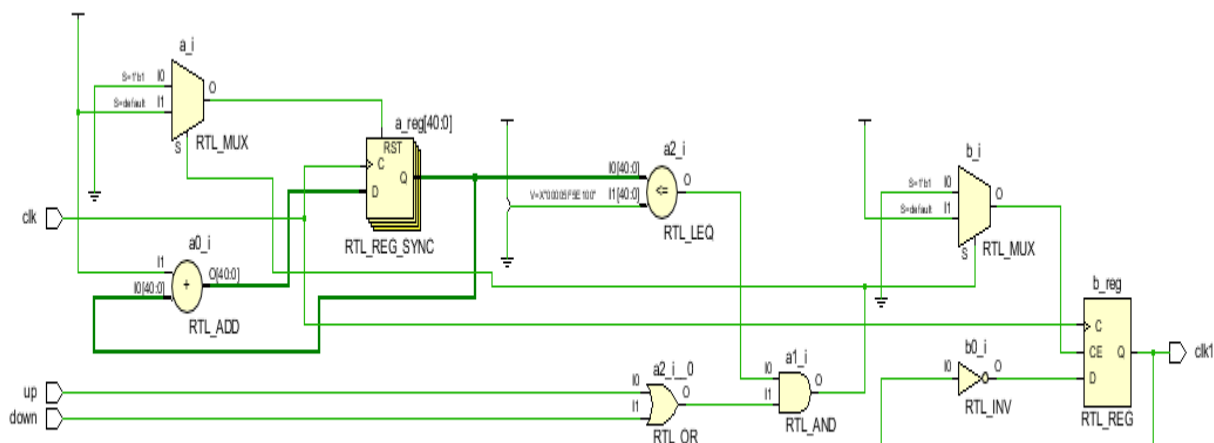
具体逻辑图如下



其中输入为当前楼层数，目标楼层数、时钟端以及上行或下行的标志（二进制数 00 表示静止，01 表示下行，10 表示下行）。输入输出为开门和关门的提示，输出为上，下或者是静止。当电梯运行时。执行关门指令，即关门的值为 1。当电梯到达目标楼层时，执行开门指令，持续一秒钟，然后自动执行关门指令。

3.5 时钟模块

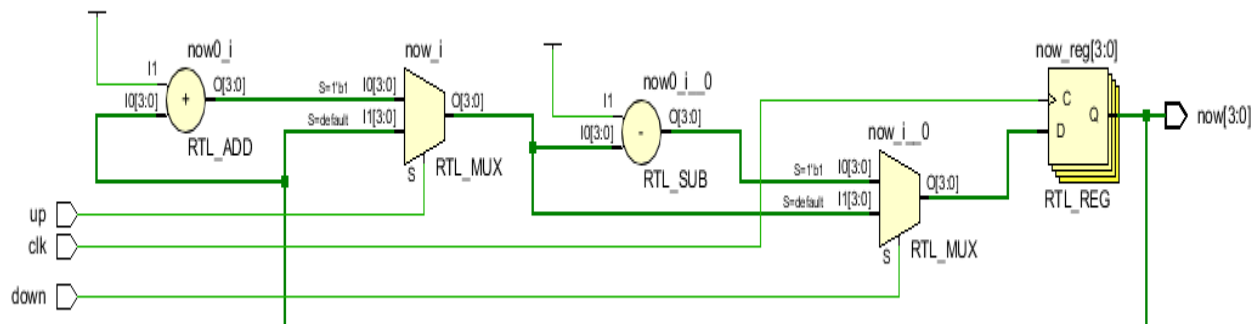
具体逻辑图如下：



输入为时钟端，上行和下行。输出为另一个时钟端。当上行（up）或下行（down）为 1 时，即可激活计时器开始进行工作。

3.6 楼层控制模块

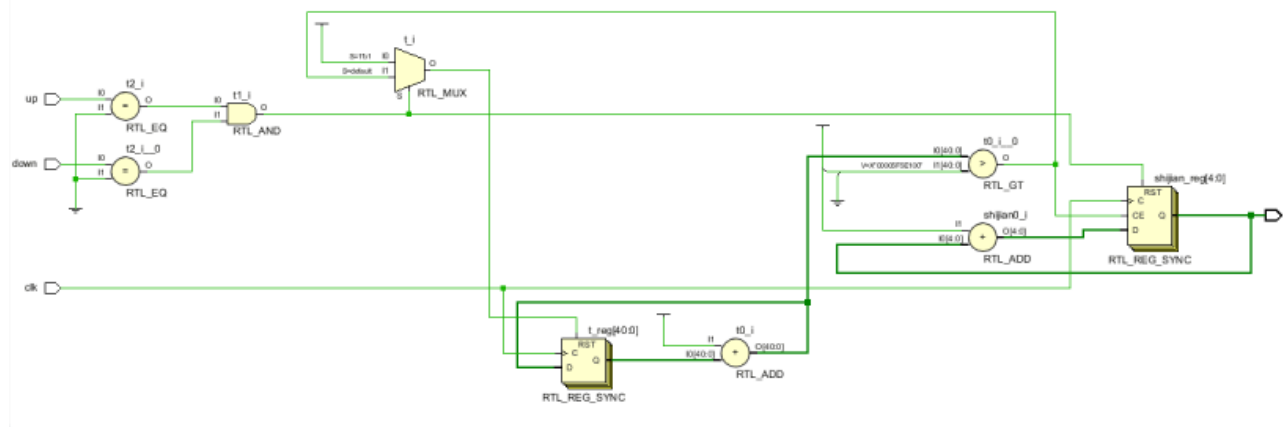
具体逻辑图如下



输入为上行信号、下行信号以及时钟端，输出为此时的电梯所处楼层（四位二进制数 **now**）。当上行信号为 1，下行信号为 0 时， $now = now + 1$ ，即下一状态为当前楼层加 1。当上行信号为 0，下行信号为 1 时， $now = now - 1$ ，即下一状态为当前楼层减 1。输出得到的 **now**。

3.7 计时模块

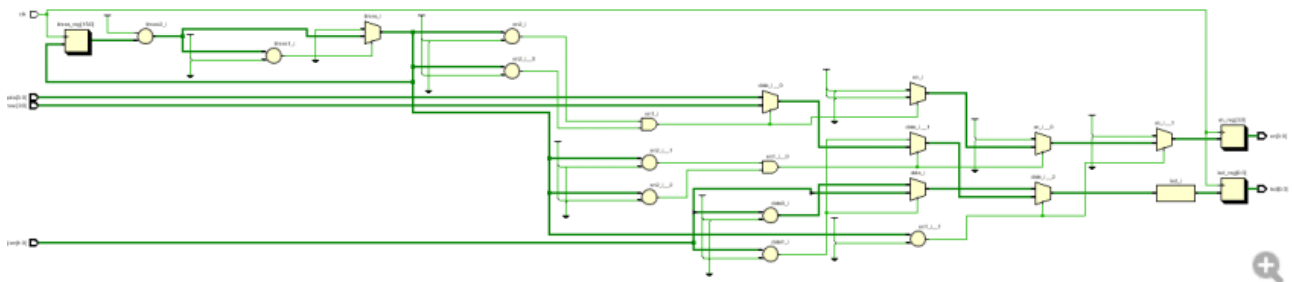
具体逻辑图



输入为上行信号、下行信号以及时钟端，输出为当前电梯所运行的时间。当处于上升状态时，当上行信号为 1，下行信号为 0 时，开始计时，当上行信号为 0，下行信号为 0 时，停止计时，时间清零。当处于下行状态时，当上行信号为 0，下行信号为 1 时，开始计时，当上行信号为 0，下行信号为 0 时，停止计时，时间清零。

3.8 楼层显示模块

具体逻辑图



输入为时钟端、时间、当前电梯所处楼层和目标楼层。输出为电梯的楼层显示和开关门显示.其中的四个使能利用视觉延迟同步实现。当时间大于 10 的时候，将时间减去 10 的倍数，作为个位，将倍数作为十位，即可显示出电梯运行所用的时间。

4.调试过程

实现了各工作端，但是当我们把它们组合到一起时就出现了问题。调试过程如下：

4.1 问题一：

情况数目特别多，每一次的判断需要融合内部，上升，下降，彼此无法分离。例如：

1.

楼层数	1	2	3	4
现态				
上升请求			1	
下降请求			1	
内部请求				1

2.

楼层数	1	2	3	4
现态				
上升请求				
下降请求			1	1
内部请求				

解决 1：

用了一堆 assign，但是结果驱动负载不足.....

解决 2：

用 if else if 书写逻辑

好处：if 和 else if 本身具有优先级，在书写的时候要注意顺序，就可以达到满意的效果，并且 Vivado 会帮助我们选择最合适的门，不怕驱动不足了。

4.2 问题二:

对于题目的要求,我们的优先级应该是:在电梯上行时,只对电梯上方的楼层请求进行检查,而在电梯下行时,只对电梯下方的楼层进行检查,例如:

电梯层数	1	2	3	4	5	6	7	8	9
				上升					
请求信号	1					1		1	

4 楼: 6 8 有请求 con10 电梯上行

6 楼: 8 有请求 con10 电梯上行

8 楼: 上方无请求 con00 电梯下行

1 楼: 下方无请求 con00 电梯静止

面临的困难: 电梯现态不同,所需要判断的请求不同,而 verilog 中,这样的循环不能被综合

解决办法: 拆分了循环,分类讨论进行判断

从 1500 行的代码简化成 300 行。

5.设计结论

5.1 我们设计的电梯控制器除了 5 个基本功能,还增加了 6 个附加功能,简单来说,设计功能如下:

1) 楼层与电梯上下行、电梯在每一次启停之间的运行时间以及开关门状态显示

电梯刚开始处于 1 层状态,通过设定楼层的二进制开关来确定目标楼层,之后按下确认键(S1)电梯便开始运行,此时显示器第一位(DK1)显示的为当前所在楼层,第二位(DK2)为最后一次输入的目标楼层,第三位和第四位显示的是当前电梯运行时间(其中第三位为十位第四位为个位),当电梯处于关门且上升状态时,LD1 中的 0 位灯管常亮,当电梯处于开门状态时,LD2 中的 4 位灯管常亮,当电梯处于关门且下降状态时,LD1 中的 1 位灯管常亮。

2) 楼层数的输入与取消

当电梯运行时,按下取消键(S3)即可清除本次的输入并且电梯会停止在当前所显示的所在的那一层。同时我们还多加了一个可以多次输入的附加功能,即可以在九层时,输入 1,2,4 层,这时候电梯会在 1,2,4 层分别停留,并且当按下取消键(S3)时,便可以取消最后一次的层数输入,并且在误触取消键时,还可以再次点击取消键来重新得到最后一次的输入层数。

3) 电梯的复位功能

任何时候按下复位键(RESET),即使已经在别的楼层且有目标楼层,电梯也会立即回到一层。

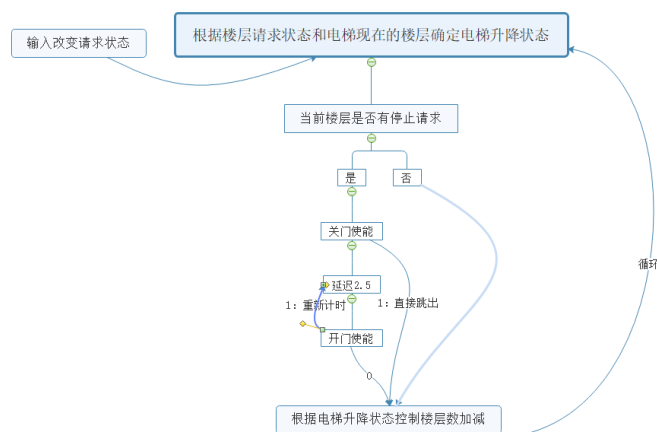
4) 电梯拥有正确的优先级

电梯拥有正确的优先级,即上升的时候会先上升到最高的楼层请求,再执行下降的请求,避免出现电梯在楼层间反复的辗转导致最高层乘客一直无法到达的情况。

5) 能够延长或缩短电梯的开门时间(实际生活中的开关门按钮)

当电梯处于开门状态时,按延长开门键(S0)便可以延长开门时间,按关门键(S4)便可以立即关门。

5.2 整体流程如下：



5.3 完善与反思

其实可以完成更多的功能，真正实现现实中的电梯，功能可以继续完善，但是在整个过程中已经花费了大量的精力和时间。时间紧迫，我们也想早点完成早点复习，所以就不再一一优化了。目前的这个系统还存在一些错误和不足，但我觉得我们已经尽力，以后有时间再继续修改吧。

6.设计心得与总结

成员 1

整个过程我经历了感动，痛苦，兴奋种种过程，在设计的过程中也犯了很多错误，现在稍作反思和总结。

- 1.模块设计和仿真文件。最开始的时候模块写好就直接用上去，主要还是想早点完成工作，但是缺少了仿真波形，没有正确性验证。出错了还要找原来的 ip，非常麻烦。
- 2.平心静气。在找错误和调试的过程中，有了错误难免会心烦意乱，心一旦着急，更容易犯其他低级错误，这样周而复始的犯错找错，如果心无法平静下来，很难继续工作下去。
- 3.分工问题。一开始沟通有所欠缺导致最后合并的时候，程序和想象的偏差实在太大，计时功能几乎要重写，一改再改，体现了写代码牵一发而动全身的本质问题。最后又因为心态问题卡在了一些小问题，严重影响了工程进度。

成员 2

在工程的建立、检测与总结的过程中，我收获颇丰，既有书写代码时候找到一些小技巧的开心，也有团队合作之后收获到比知识更加重要的友谊的快乐。

1.书写代码的惊奇发现

在研究电梯上行下行的判断时，本来想用循环直接解决问题，却发现电梯的现状不同，所需要的请求也不同，而在 Verilog 中，这样的循环却无法被综合，导致我们的工程设计遇到了瓶颈。后来经过思考以及对教材网络的查询，终于找到了解决方案，也就是拆分循环，通过分类讨论进行判断，使得电梯的每一次运行、开门和关门都能准确无误的实现。

2.团队合作的友谊更加珍贵

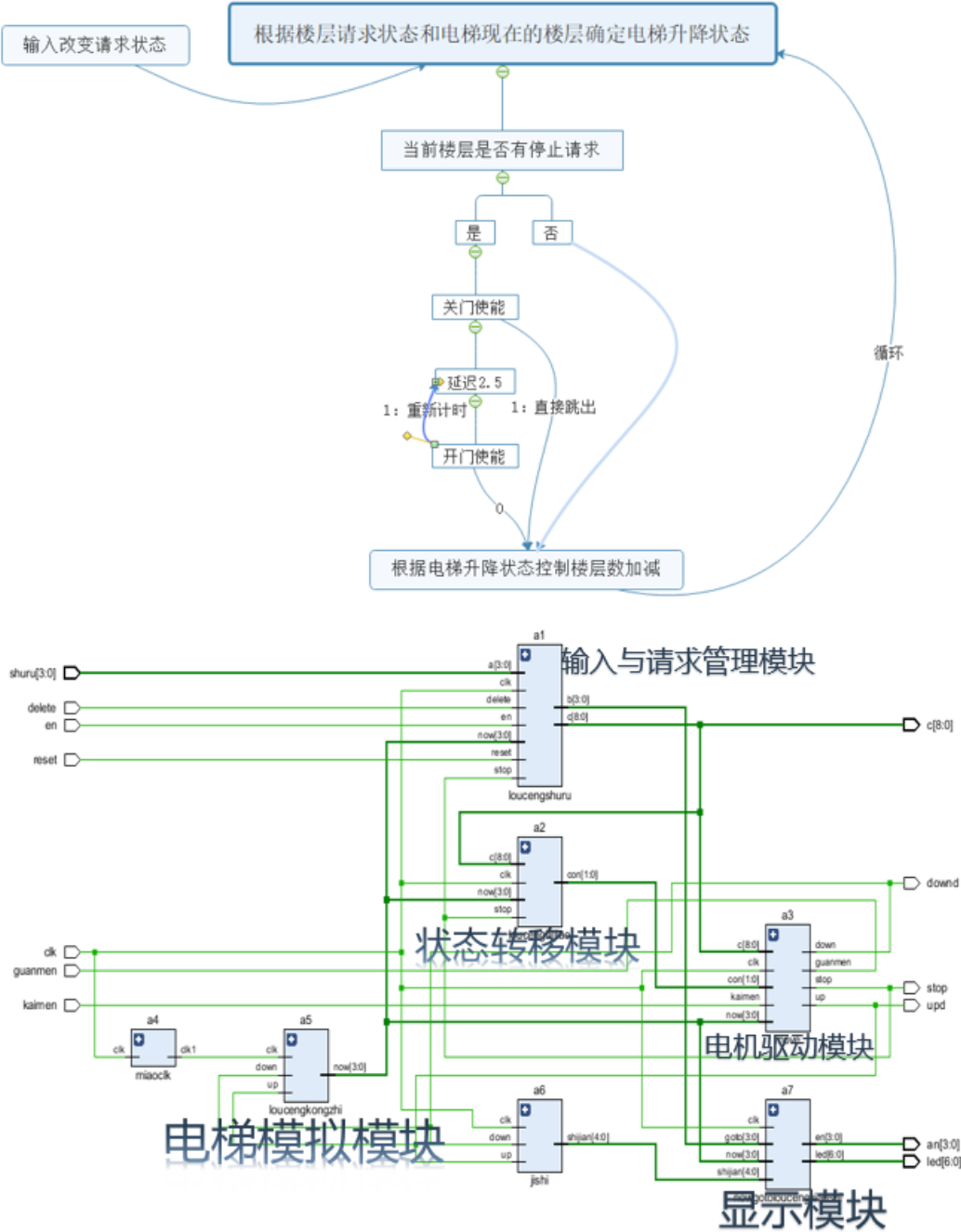
在设计研究大作业时，我们两个首先一起确定了解决该问题的大体思路并将该问题分解成了一些模块，分工处理。本来程序运行得很好，却因为最后合并时的一些语法和程序设计原因导致不能综合使用，每次修改有都要花费好大的力气，最后也无法实现双方代码的融合。最后只能重新来过，一起书写，一起讨论研究。好在有之前编写的基础，后面的程序建立也循序渐进地完成。在整个工程进行的过程中，我们也吵过，也互相埋怨过，最后还是一起熬着夜赶进度。到最后收工时两个人也都很高兴，为我们能够解决这一问题感到骄傲和自豪。在这个过程中，我觉得收获的友谊，远比知识来得更重要。

参考文献

无。

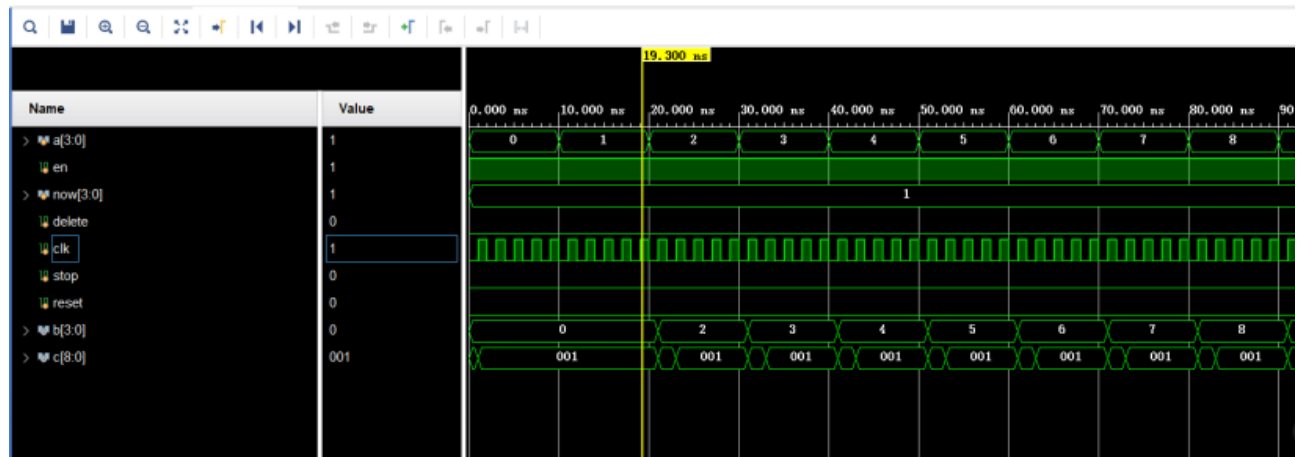
附录

附录 1：总体设计图

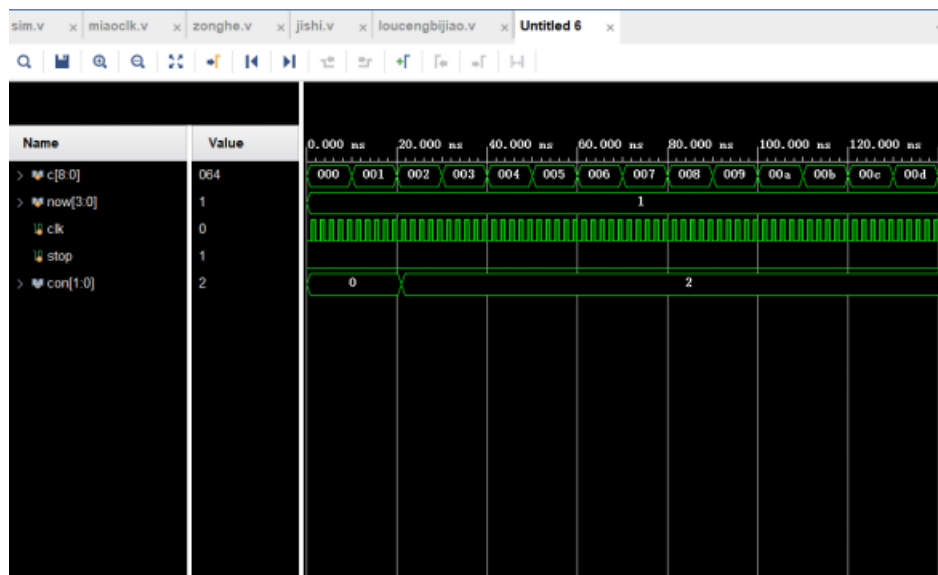


附录 2：各模块仿真截图

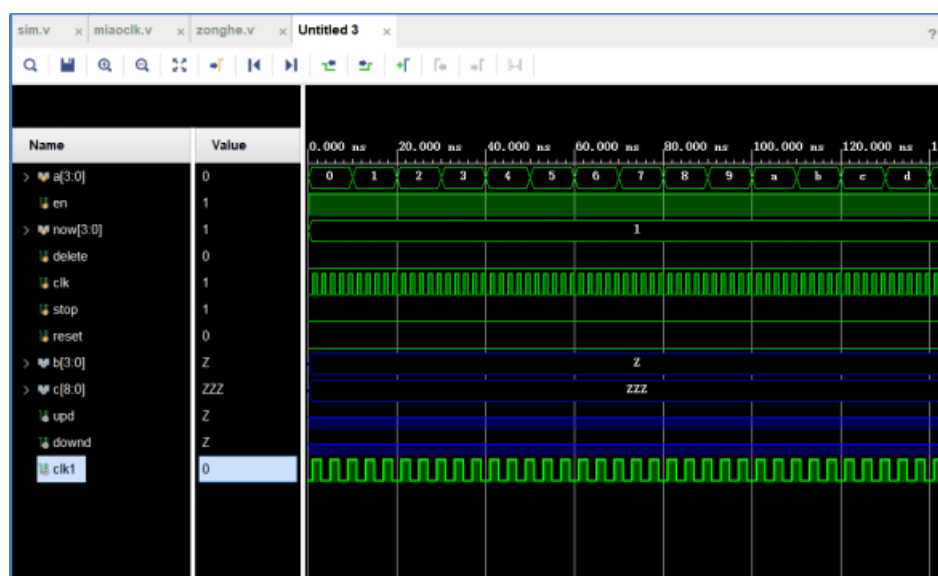
1) 楼层输入：



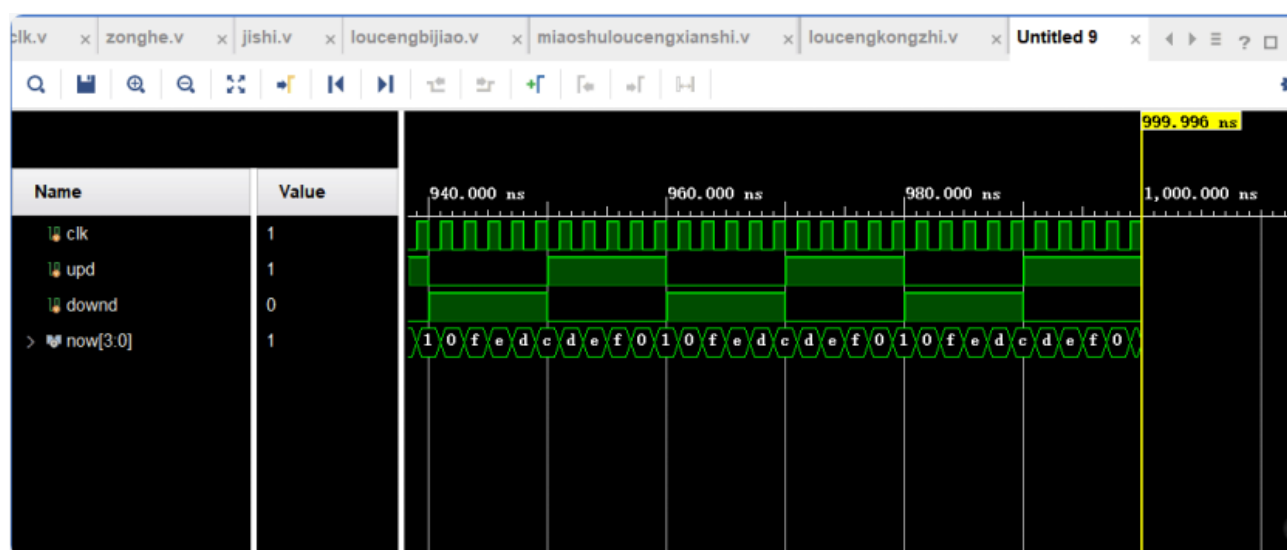
2) 楼层比较



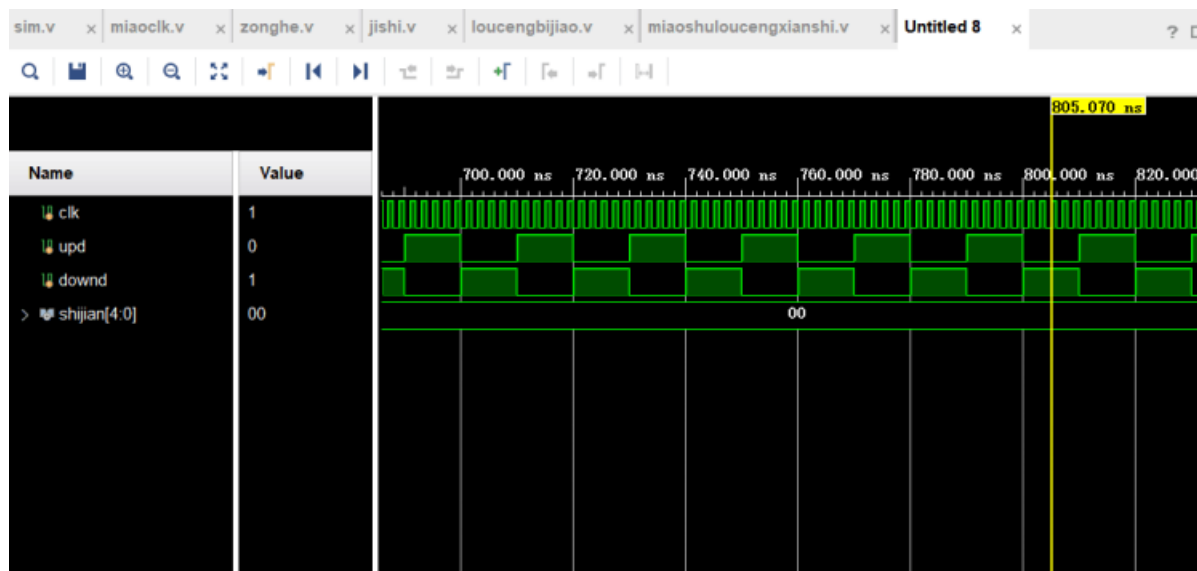
3) 时钟



4) 楼层控制



5) 计时模块



附录 3：各成员分工

成员 1 冯开来：

负责其中时钟模块、计时模块、电梯移动模块的代码书写。

负责报告 1 设计要求、2 工作原理及系统方框图、4 调试过程、5 设计结论、6 设计心得与总结成员一部分的书写

成员 2 涂鑫泊：

负责其中楼层的输入、控制、比较、显示和综合部分的代码书写。

负责报告 1 设计要求、2 工作原理及系统方框图、3、各部分模块功能及具体思路、6 设计心得与总结成员二部分的书写