



原文网址<https://blog.csdn.net/qinglingLS/article/details/74278856>

# 哈爾濱工業大學

HARBIN INSTITUTE OF TECHNOLOGY

电子密码锁电路设计

课 程 名 称: 数字逻辑


学生所在院（系）: 计算机学院

学 生 所 在 专 业: 计算机科学与技术

小 组 成 员: [Redacted]

小 组 项 目: 电子密码锁电路设计

2017 年 6 月 27 日



---

## 目录

- 1、设计目的及要求
- 2、工作原理、系统方框图
- 3、各部分选定方案及电路组成、相关器件说明
- 4、调试过程
- 5、设计结论
- 6、设计心得与总结
- 7、参考文献

附录一：总体器件表及相关器件的功能表、管脚分布

附录二：总体设计图

附录三：仿真结果

附录四：小组各成员所做工作

## 1. 设计目的及要求

题目：电子密码锁的设计

### [设计要求]

- (1) 设计一个开锁密码至少为 4 位数字（或更多）的密码锁。
- (2) 当开锁按钮开关（可设置 8 位或更多，其中只有 4 位有效，其余位为虚设）的输入代码等于所设密码时启动开锁控制电路，并且用绿灯亮、红灯灭表示开锁状态。
- (3) 从第一个按钮触动后的 5 秒内若未能将锁打开，则电路自动复位并发出报警信号，同时用绿灯灭、红灯亮表示关锁状态。

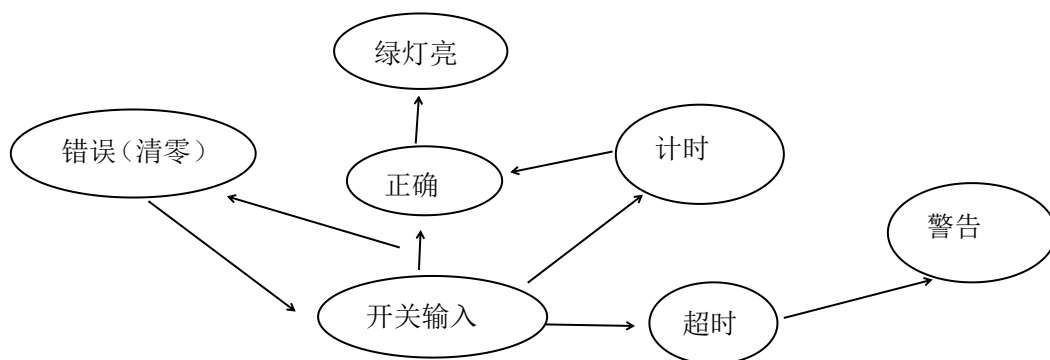
### 附加功能：

- (1) 可以设置密码，通过设置密码按钮 SP（S4）设置，且可以重新设置新密码。任何情况下按 SP 后输入都可以设置密码。
- (2) 五秒计时采用倒计时显示在开发板的七段数码管上。五秒后发出警报（绿灯亮）。
- (3) 能记录按键输入密码的次数（0-9）。
- (4) 能清零输入次数而不改变密码。

## 2. 工作原理、系统方框图

根据系统功能来分析，

- (1) **编码工作端：**将输入的 4 位密码转换成二进制编码（我们选择 8421-BCD 码）；
  - (2) **存储工作端：**初始要先设置密码，就需要将此密码保存。
  - (3) **等值比较工作端：**将之后输入的密码和存储的密码进行比较。
  - (4) **计时工作端：**体现五秒倒计时的功能
- 系统的结构功能如下图：



### 3. 各部分选定方案及电路组成、相关器件说明

下面就对以上的功能端进行一一的实现。

由于对 verilog 语言运用还不够完全熟练，我们采用分步将各个功能端所需要的器件用较简单的 verilog 语言写好后自己封装成 IP 核，这样会降低整体设计的难度和错误率。

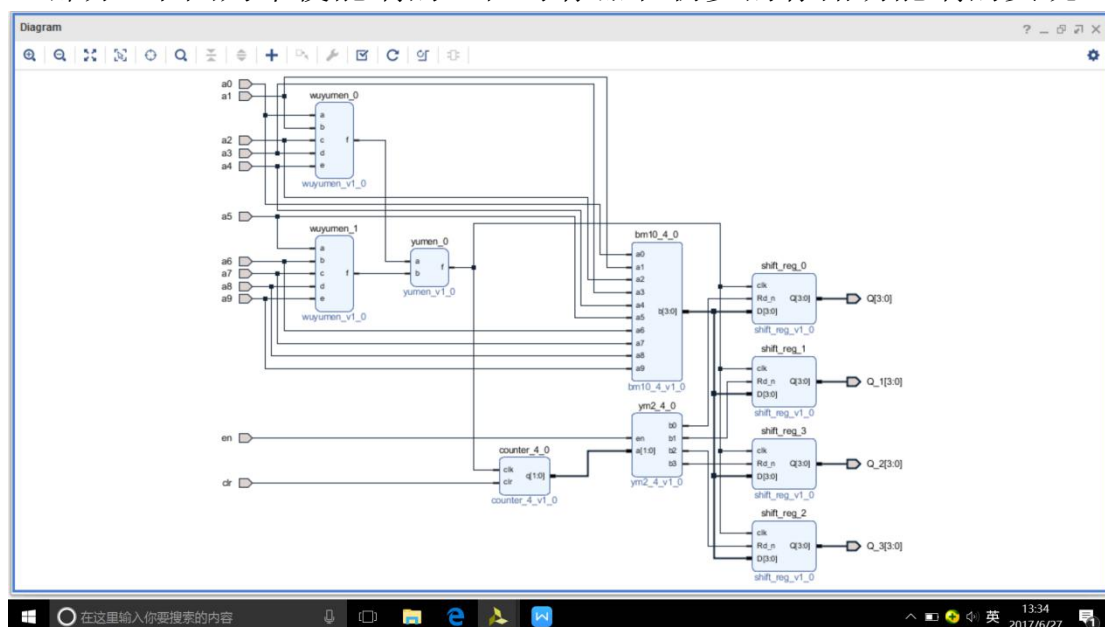
(1) **编码工作端**，最核心的功能就是将十进制数编码为二进制数，提供给其他工作端。我写的编码器的功能就是 10—4 编码器。这样就基本实现了编码功能。编码器的输入是高有效的。能够将输入的数转变为 4 位二进制码，表示从 0-9。

(2) **存储工作端**：要想存储 4 个 0-9 之间的数，显然每个数转换成二进制有 4 位。我们这里写了 4 个 4 位寄存器来分别存储这 4 个 4 位二进制数，我们用编码工作端的输出端来连接存储工作端的输入端。由于每次只能有一个十进制数（4 位二进制数）进入输入端，我们需要分别使能这四个寄存器按照确定顺序存在 4 个寄存器中。

我们还写了一个 4 位计数器和一个 2-4 译码器来组合实现这个功能。计数器负责选择每位数，从 1-4 位计数。

为了知道已经输入几位密码而且准确地存储，计数器的时钟端脉冲是密码输入端按下时候产生的上升沿，用来驱动计数器，计数器从 00 记到 11。计数器的输出端接到译码器的输入端，译码器的输出端分别与 4 个寄存器的使能端相连接。

这样每一个密码输入，都只能使译码器输出端的一个有效，从而将这个数存入寄存器中。（下图是存储工作端和编码工作端的组合的一部分，下图为带使能端的 4 位寄存器和初步的存储功能端的实现）。



要想使每次密码输入端按下时都产生一个脉冲，上图中我们可以用或门来实现，开始时密码输入端都为低电平，只要有一个高电平输入，输出到时钟信号就是高电平，计数器就向下计一个数。

(3)等值比较工作端的核心工作是判断输入的密码与存储工作端存储的密码是否相同。我们设计了 4 个 4 位数值比较器来实现比较的功能，当每个比较器的输出端都为 0 时，密码正确。于是我们将 4 个输出端用或门连接，当与门输出为 0 时绿灯亮。

密码每次只能输入一个，同样的需要计数器和译码器来使能编码输出端的数据输入到每个比较器中，每输入四个密码如果正确，那么密码锁就会打开。

我们新输入的密码需要暂存储在另外四个寄存器中，以便实现比较，暂存的原理与设置正确密码所用的原理基本一致，也使用计数器和译码器使能的。

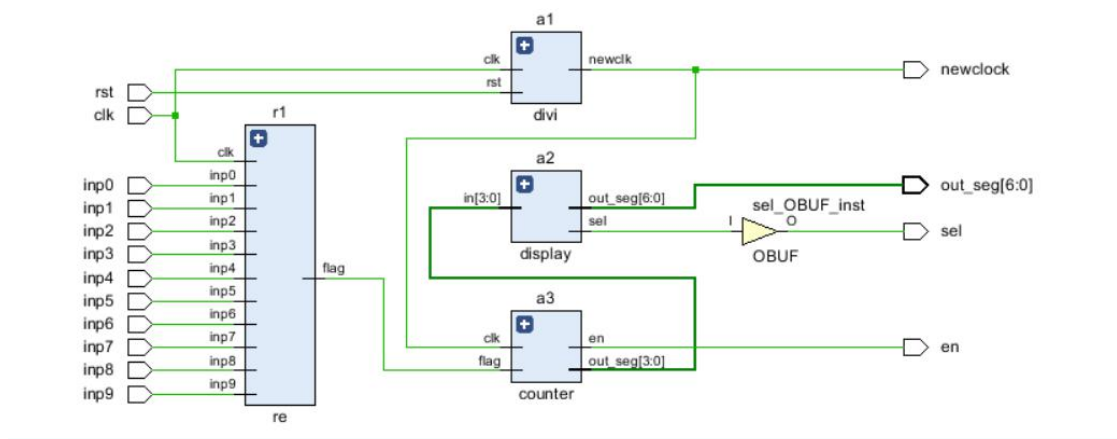
实现了输入密码的暂时存储之后，等值比较就显得很简单了，只需要将寄存器的输出端口对应连接到两个等值比较器上，并将两个比

[REDACTED]

#### (4) 倒计时工作端:

倒计时工作端包括 a. 分频器, b. 计数器, c. 显示器。能准确表示秒数, 并且能在记录 5s 后, 发出警告 (alarm 警告器亮) 且红灯亮。使等值比较器和寄存器暂停工作, 即使输入正确密码也不能再使绿灯发光。

功能：输入 0-9 数字中任何一个，计数器开始工作倒计时，在 rst 为 0 的情况下，能够输出新的时钟信号（现实秒数），输出段码，位码，使能信号。为下一个器件的工作与否提供信号。



## 4. 调试过程

实现了各工作端的功能,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:

我们设置正确密码然后再次输入密码进行比较，这时就出现了问题。设置密码没有问题，密码都存储在了 4 存储工作端中。但是我们在之后输入密码却出现了问题，我们发现密码没有正常的存储在寄存器中，也没有正常传输到比较器中，导致灯一直是亮着的，无法进行正常的锁定状态。这个问题困扰了我们很久。几乎使我们崩溃。

后来我们对计数器 ip 进行了改进，当它从 00 计数到 11 之后就锁死，而且只有清零之后才会再次可以输入密码。

还有比较器出现了失误，总线输出是两位信号，不能成功输入到

或门，导致灯的状态错误，然后改进比较器的输出为一位信号就好了。

计时工作端也出了一些差错，它不能正确计时只能在按键时计时，而且复位没有体现。会有乱码，甚至不能正确表数字，固定不动。后来重新改写了计时工作端的代码就可以完成功能了。

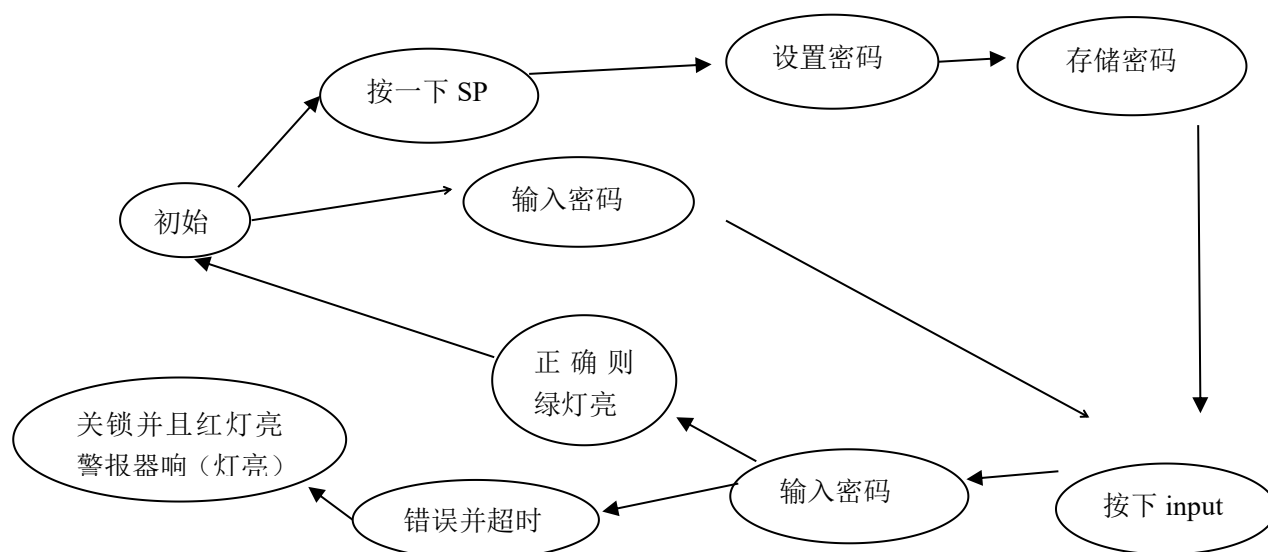
可以看到初始时没有密码，故绿灯亮，等设置好密码之后绿灯不再亮，只要输入正确的密码，绿灯就会再次点亮。至此，我们密码锁的基本功能已经实现了，能够设置存储密码，并进行判别了。

再来检验一下输入错误密码，绿灯会不会亮，可以清楚的看出，输入错误的密码后，绿灯没有亮，直到输入正确密码才亮，而且超时五秒之后电路复位，发出报警信号，绿灯灭，红灯亮。

Ip 在连接之后又出现了锁死和其他各种各样的问题，让我们很头疼，每个器件都进行了检查。

## 5. 设计结论

我们设计的电子密码锁有 10 个数字输入端，能存储 4 位密码，可以设置、修改密码，能够用绿灯亮表示密码输入正确，绿灯灭表示关锁。同时密码输入只有 2 次机会，并且只能在 5 秒之内完成输入，否则关锁。整个流程如下：



可以按照现在的结构实现更多位数的密码锁，还有密码的锁死方面考虑到了却没有来得及修正。但我们在这个过程中已经花费了大量的精力和时间。鉴于时间紧迫，我们在之前的那个错误上耗去的时间太多，我们就不再一一优化实现了。目前的这个系统还存在一些错误与不足，我们已经尽力，在设计要求之外就不再一一修改。

## 6. 设计心得与总结

这个过程中我先感觉激动，后感觉痛苦，最后觉得自己在设计过程中犯了很多错误，这些痛苦都是可以避免的。

1. 模块设计问题。每个模块我们写好了就直接用上去了，没有仿真波形，没有正确性验证。出错了就去找原来的 ip 一一改错，这些麻烦都是可以避免的。

2. 避免写仿真文件。这种偷懒的行为绝对不可取，要仿真要一个个模块地验证。

3. 全代码或者全模块。因为是两个人合作，她采用了模块写作，我采用的代码写作，最后合并起来排错的时候，因为对各自的方法不熟悉，带来了很多时间上的浪费。

4. 分工明确。因为一开始沟通的偏差，最后综合的时候，程序和想象的偏差实在太太大，倒计时器几乎需要重写。比较器也是一改再改。正确的做法应该是一次完成，检查过保证正确了就无需再改，而不是无厘头地改来改去，每次卡在一些小细节上。

5. 要静下心来。到了后期因为时间上的紧张。我感觉自己的思考能力和逻辑能力都大大下降，卡在一个地方出不来。造成自己更加紧张，更加想不出来，更加担心。这导致排错更加困难。

6. 思路清晰。每个模块，哪个端口做什么，哪个端口什么功能，从一开始就应该商量清楚，而不是后期反复修改。应该把结构，大体



---

搞清楚，自顶向下，不可以这样模糊不清，只求大概。对于每个端口每个结构，都应该很清楚。

只有理论知识是远远不够的，只有把所学的理论知识与实践结合起来，从理论中得出结论，用于实践，提高自己的实际动手能力和独立思考的能力。在设计的过程难免会遇到过各种各样的问题，同时在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固。需要耐心，信心去完成。

这次的设计总体来说还算是成功，虽然经历了一些波折，虽然还有一些想法由于时间和精力有限没有实现，但总的来说完成了基本的一些功能。

最耗时的应该就是调试时查看所有的 IP 对其改进，特别麻烦，最后整体出错导致所有工程全部重来一遍但还是没有消除错误，直到所有的 IP 都打包好了才成功了。

同时我觉得自己最大的收获是一种兴趣，一种设计意识。我找到了自己喜欢的东西，也学会了排错。虽然过程很煎熬，结果很心碎，找错很抓狂。但是我真正学到了一些东西，明白了设计一件东西需要从何做起。

## 7. 参考文献和资料

逻辑设计基础（原书第七版）

数字逻辑与数字系统设计实验指导书

<https://wenku.baidu.com/view/18eb7d1310a6f524ccbf8509.html>

<https://wenku.baidu.com/view/c2a4d314336c1eb91b375d28.html>

[http://blog.csdn.net/qq\\_23851075/article/details/52783323](http://blog.csdn.net/qq_23851075/article/details/52783323)

<https://wenku.baidu.com/view/0294cbb3bb4cf7ec4bfed01a.html>

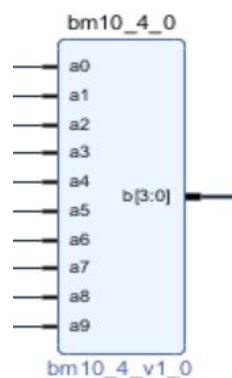
<https://www.xilinx.com/support/answers/54074.html>

[https://www.baidu.com/link?url=Ip1K3P4\\_E2BN1zKR7NrdoUyy\\_GKT-evWrnremCXfPdJO\\_SeRftZRHDqmOrzFqqQd\\_jFxRwepg7JhQeN5cwvc7a&wd=&eqid=943f89cd000758a300000000359493b12](https://www.baidu.com/link?url=Ip1K3P4_E2BN1zKR7NrdoUyy_GKT-evWrnremCXfPdJO_SeRftZRHDqmOrzFqqQd_jFxRwepg7JhQeN5cwvc7a&wd=&eqid=943f89cd000758a300000000359493b12)

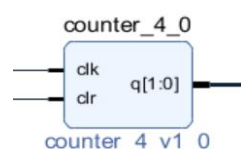
## 8. 附录

### 附录一：总体器件表及相关器件的功能表、管脚分布

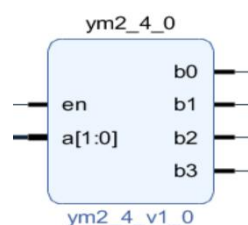
(1) 编码器，高有效，十个输入，四个输出



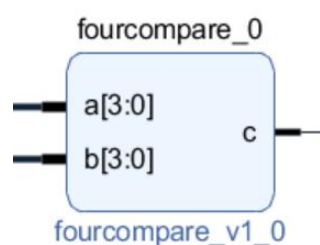
(2) 计数器，四位可清零，从 00 计到 11，除非清零，否则不会再改变数值。



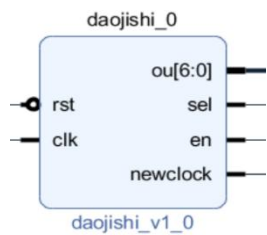
(3) 2-4 译码器，高有效输出，每次分别使能四个译码器中的一个。



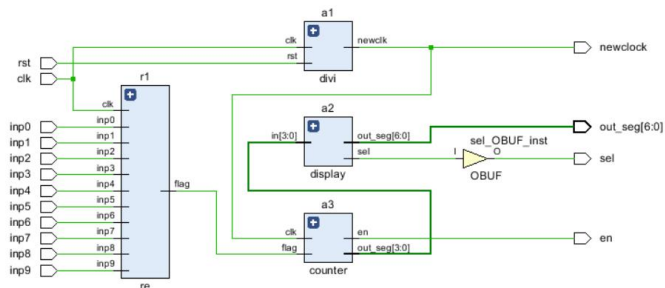
(4) 数值比较器，输入两个四位二进制数，当两个数相等时，输出为 0，给四个比较器取或非，绑定输出的灯。



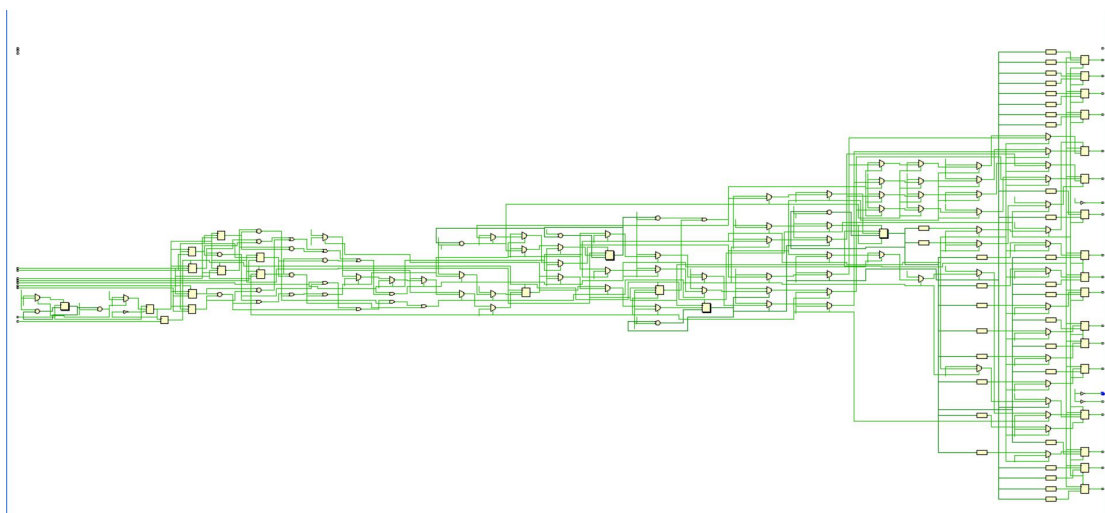
(5) 计时器，左端接受输入按键信号，时钟信号，右端输出七段数码管信号，以及新的时钟信号，复位信号。

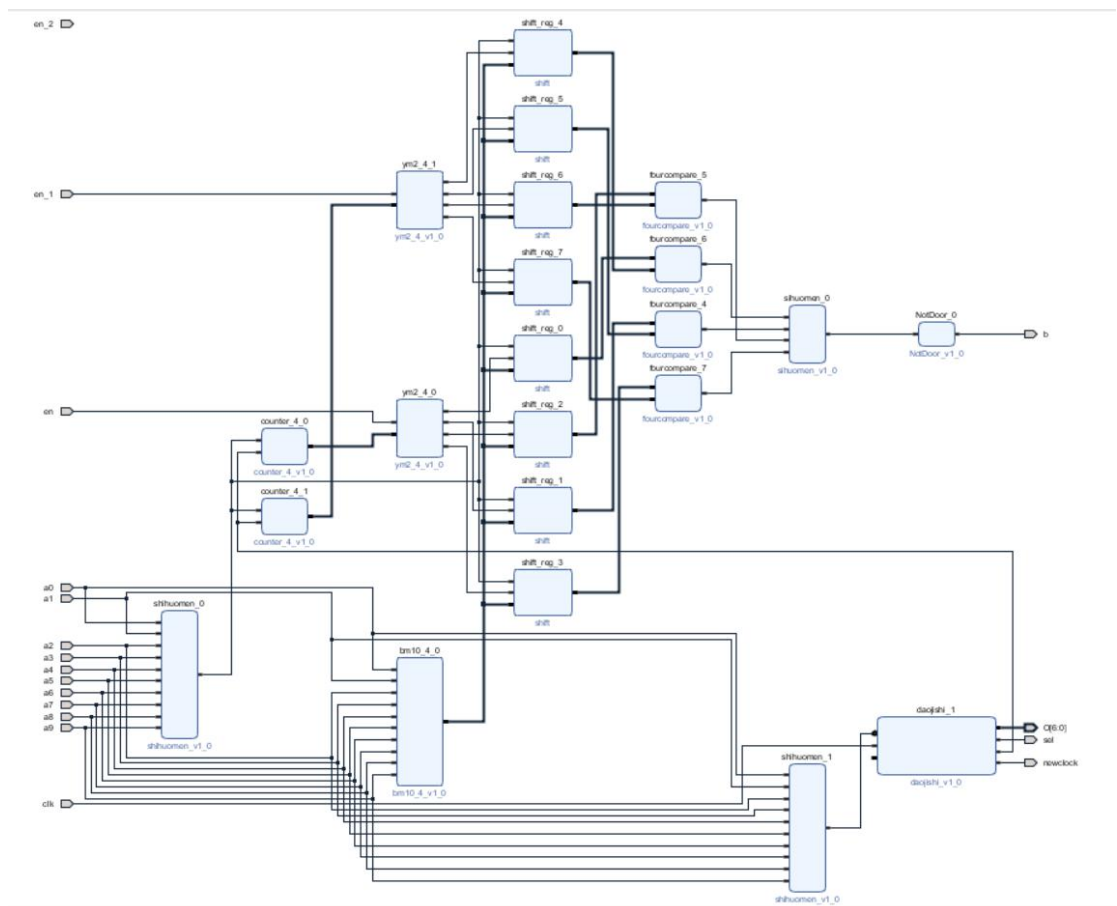
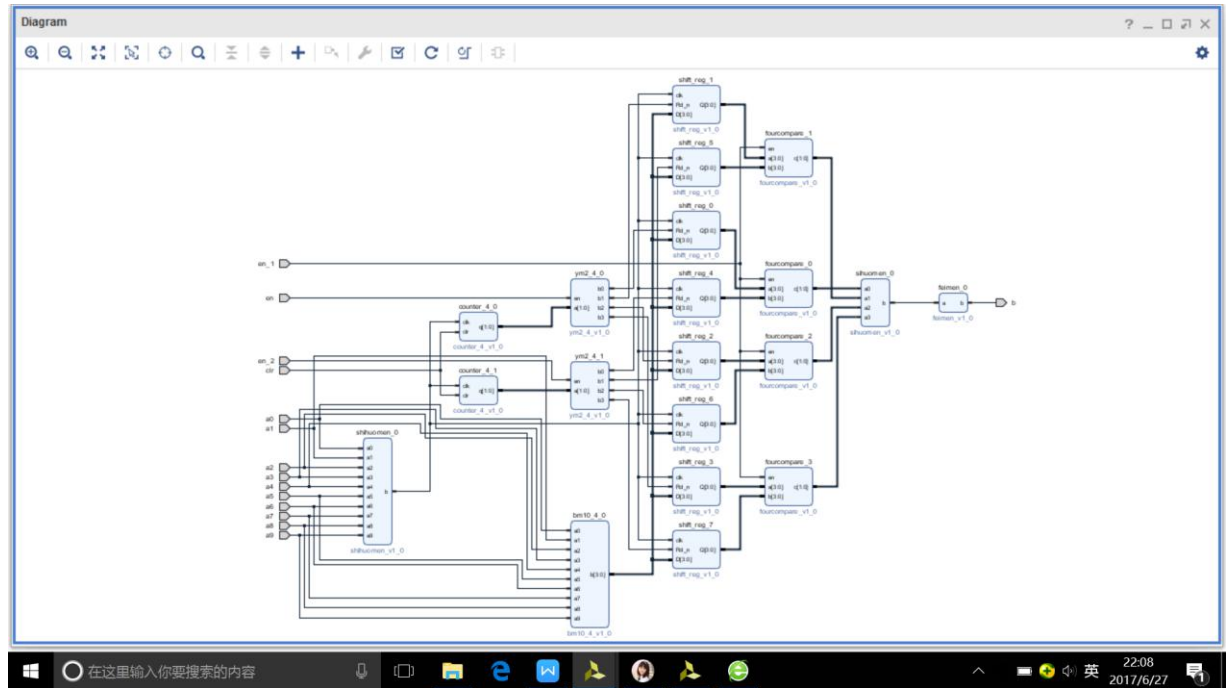


(6) 倒计时，左端接受 0-9 输入信号，时钟信号，右端输出信号。

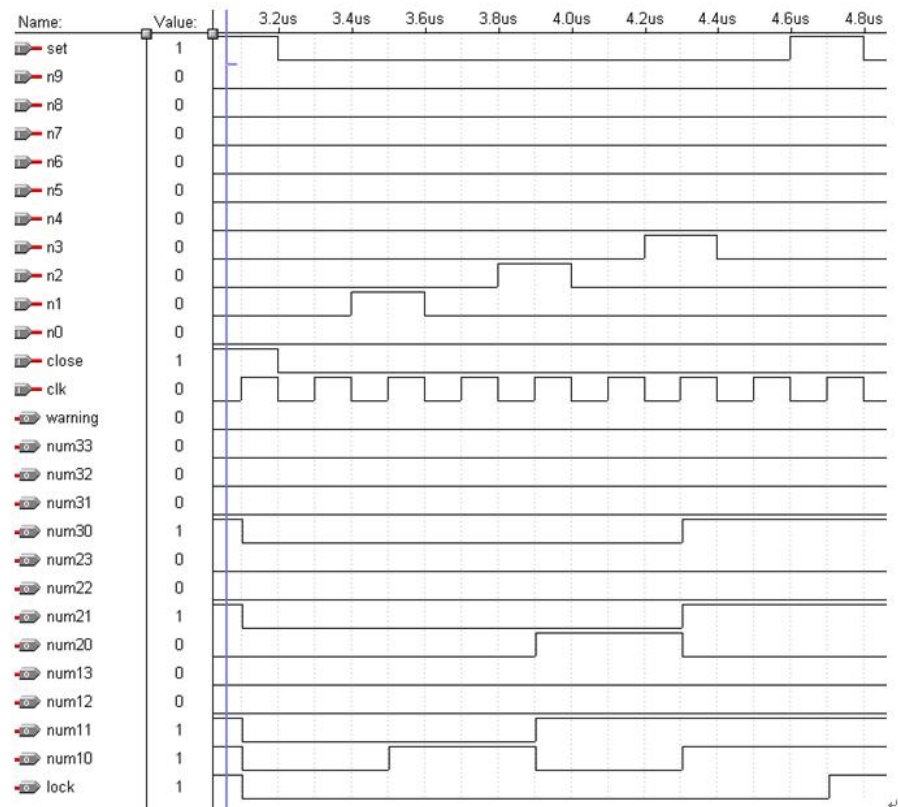


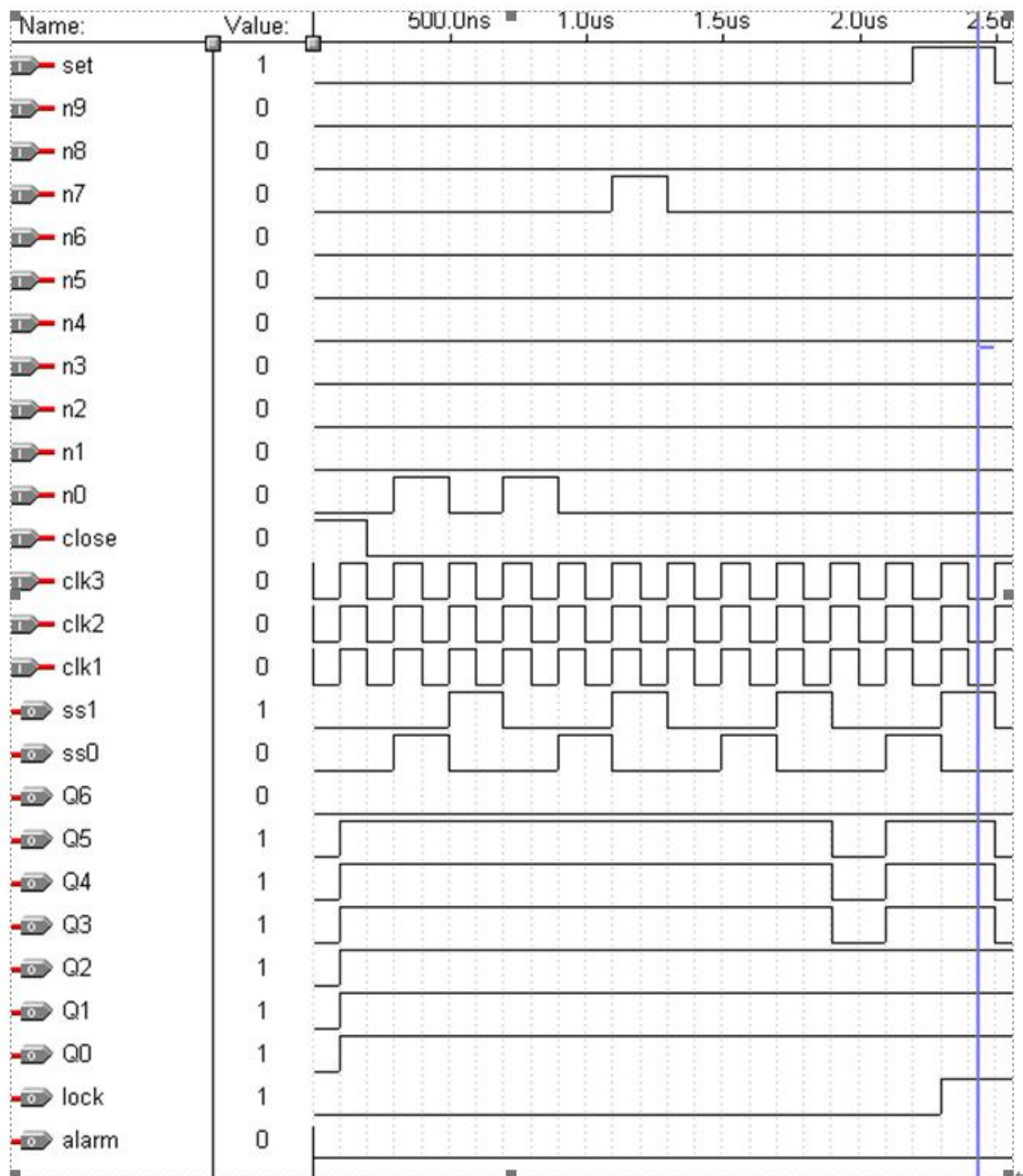
## 附录二：总体设计图





### 附录三：仿真结果





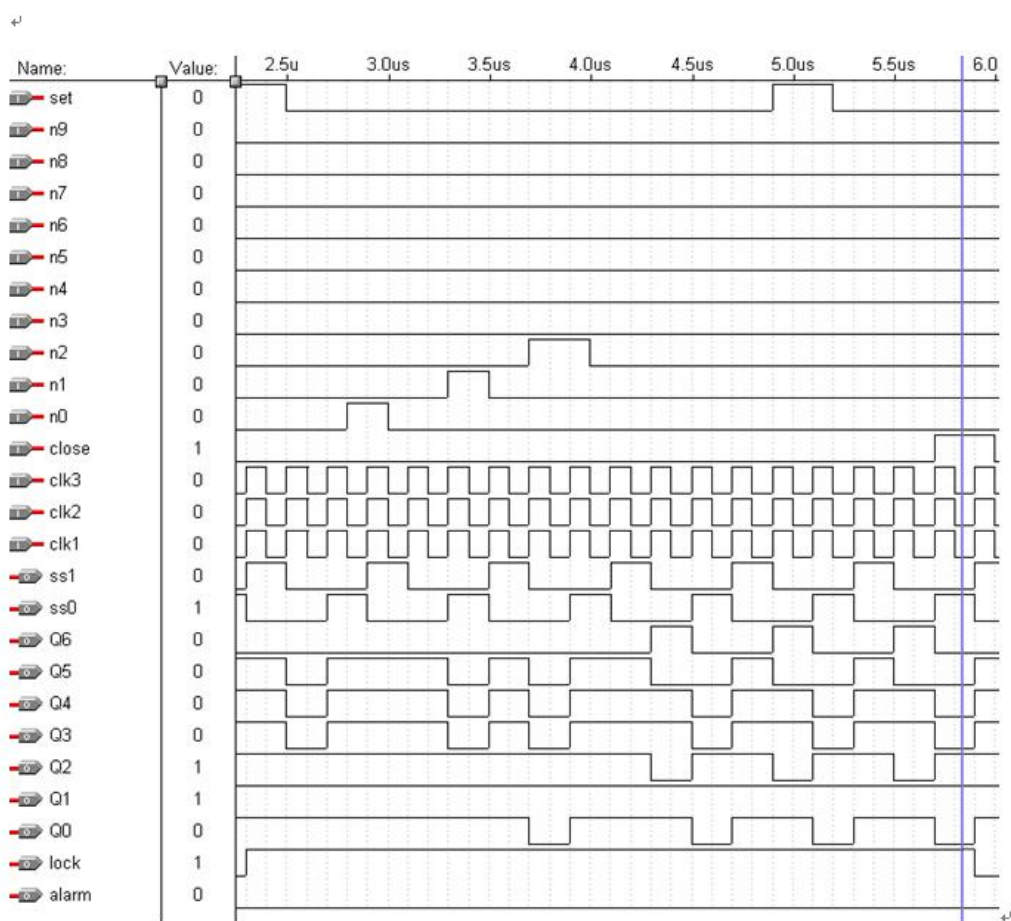
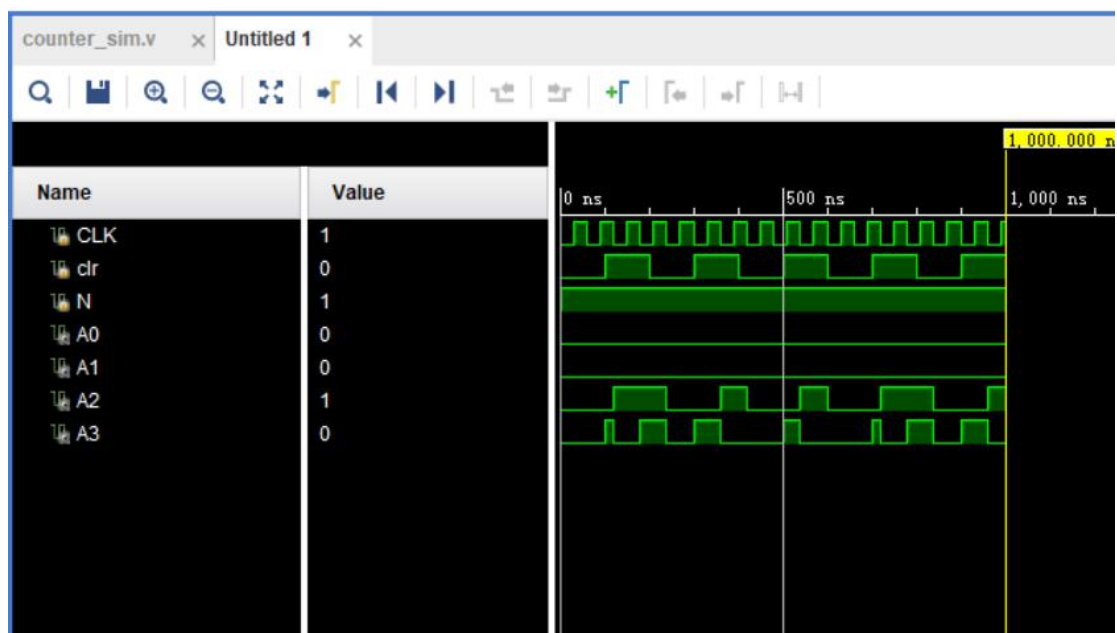


图 1.3 设置密码的波形



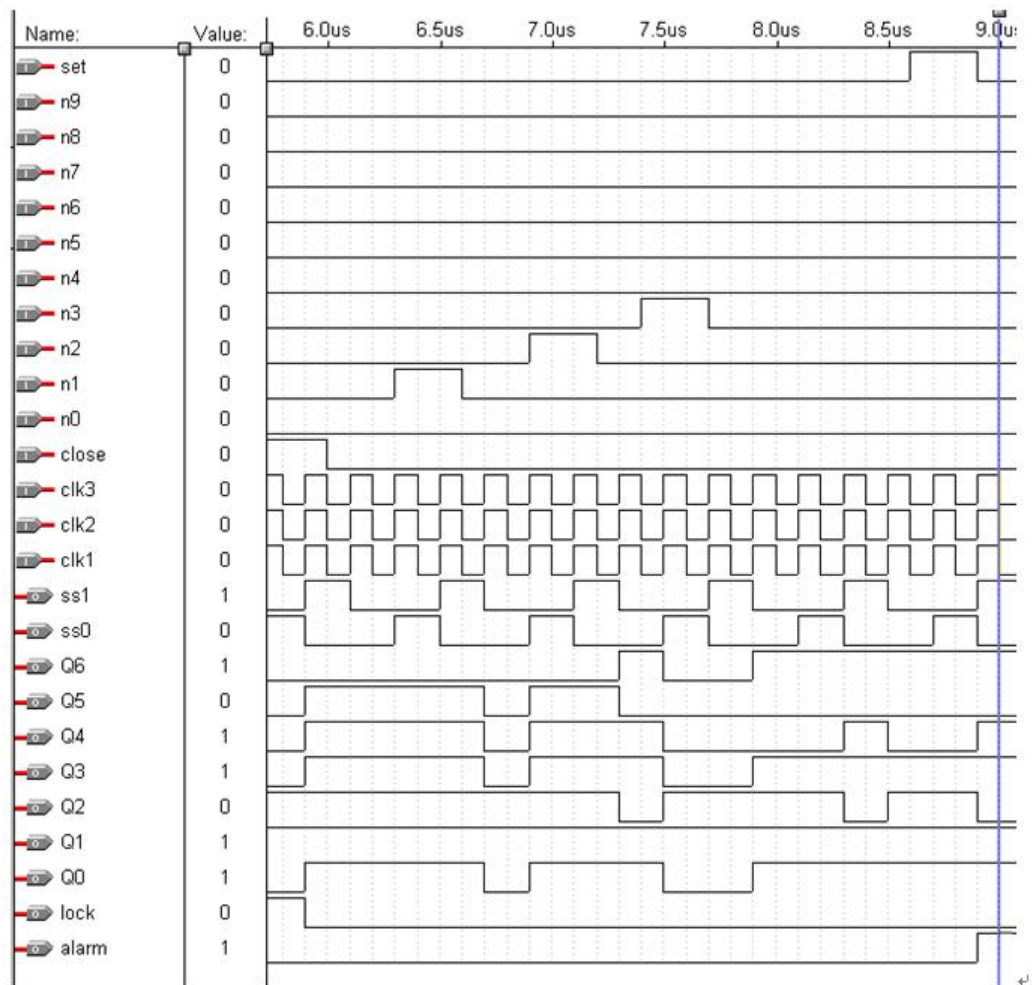
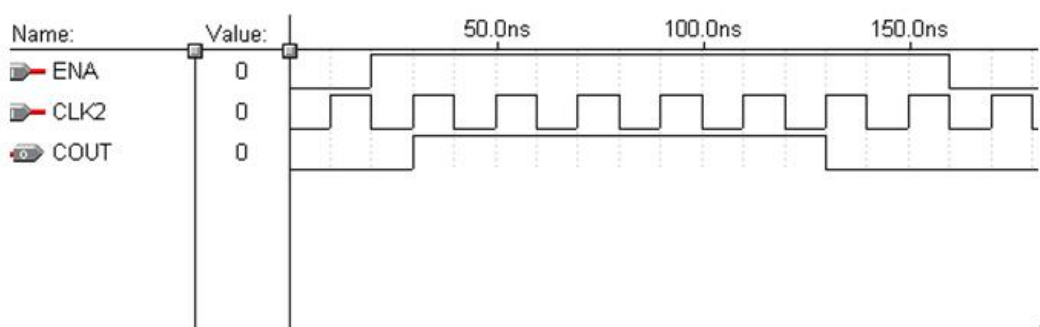


图 1.4 开锁失败的波形





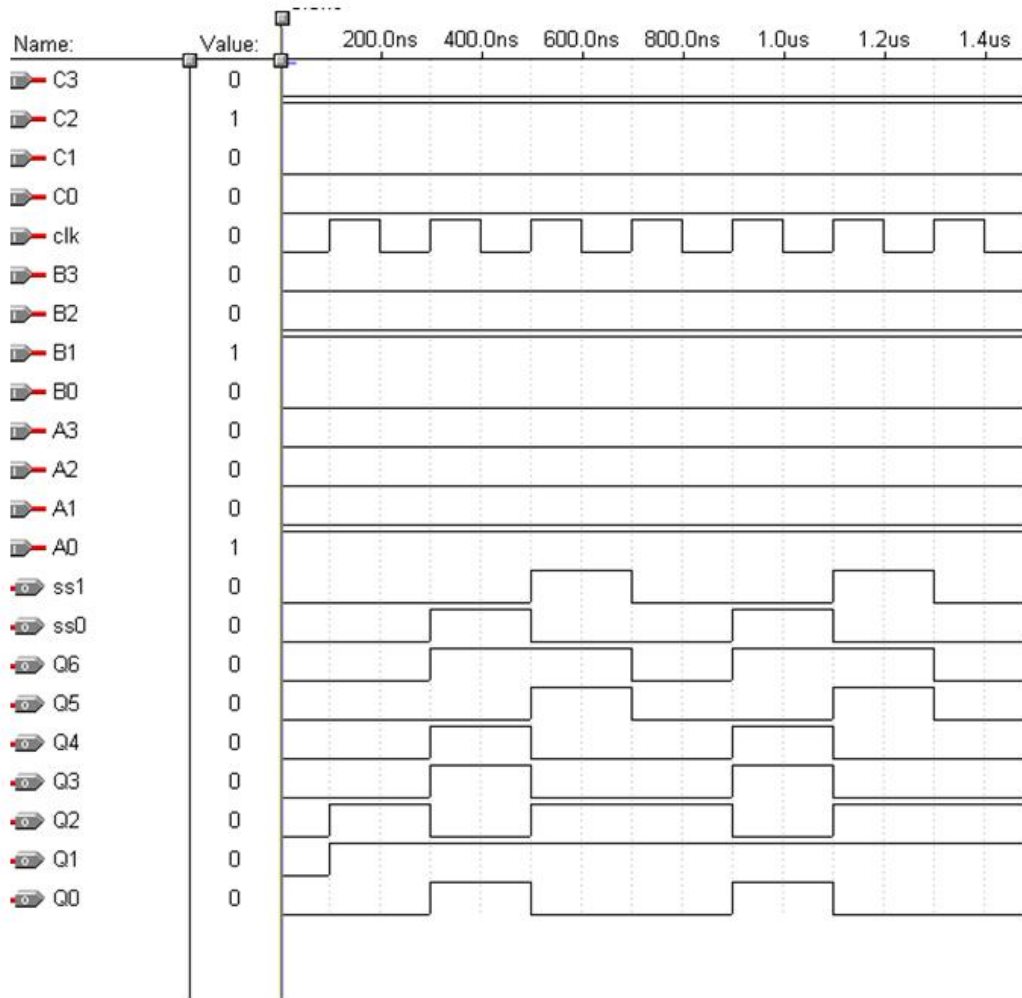


图 1.6 显示模块功能波形

#### 附录四：

小组各成员所做工作说明（设计者排序），每个成员对最终方案的贡献（哪个地方是谁设计的）

工作量差不多。当初决定模块写的时候就是平均分配了。没有谁先谁后。

计时（显示器，计数器，分频器），记录按键次数

编码器，存储器，比较器；