

（深圳）

实验报告

开课学期： 2019秋季

课程名称：数字逻辑设计（实验）

实验名称： 快递柜设计

实验性质： 综合设计型

实验学时： 6 地点： T2614

学生班级： 1801105

学生学号： 180110527

学生姓名： 李秋阳

评阅教师：

报告成绩：

实验与创新实践教育中心制

2019年12月

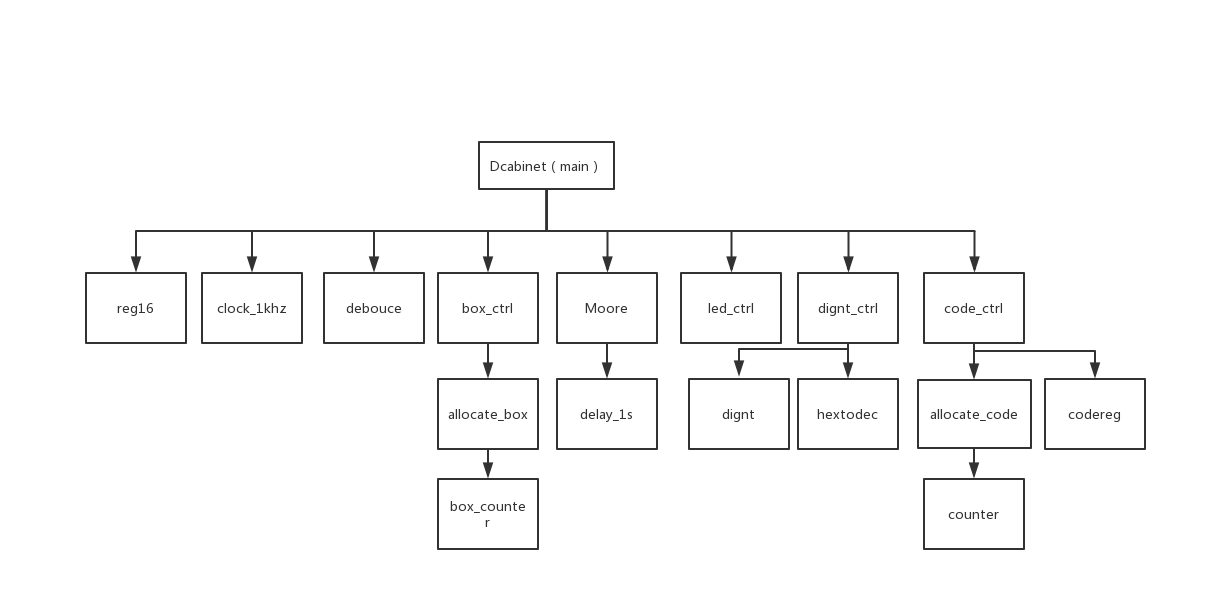
1. **实验项目**

**快递柜设计**

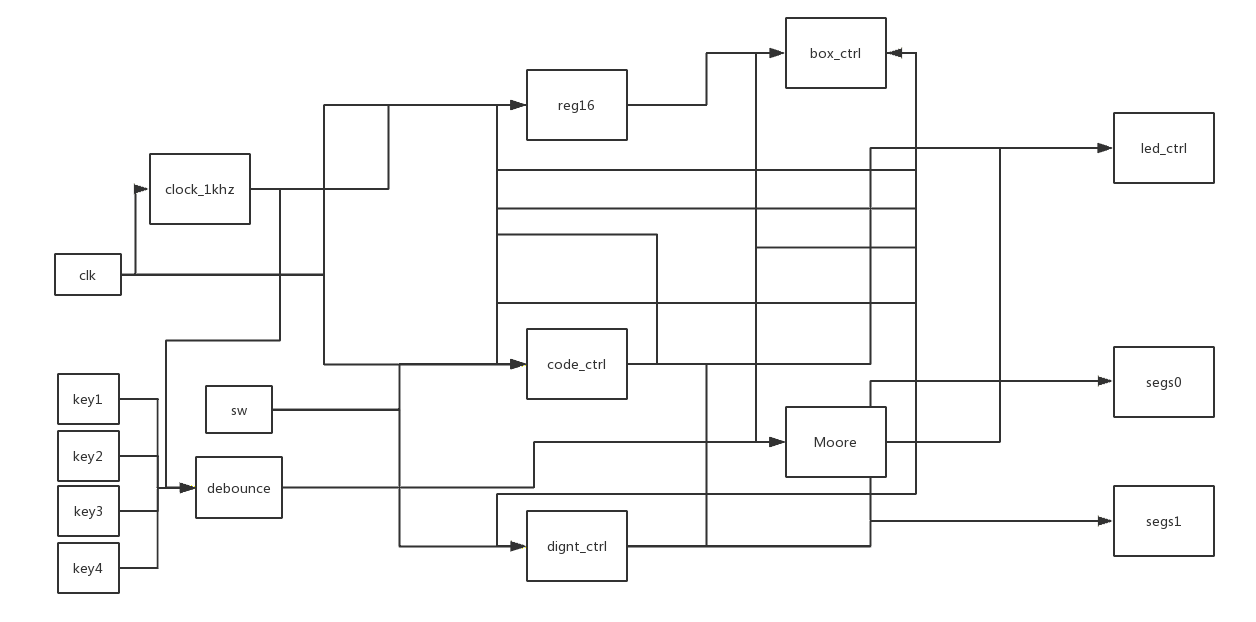
1. **系统功能详细设计及实现**
2. **系统设计描述和硬件框图**

此系统描述了简易快递柜的工作过程。系统连接后进入欢迎界面，显示空快递柜的数量（快递柜总数量为16）。选择存柜后，随机分配空箱号并显示随机生成的密码，确认后返回欢迎界面。选择取柜后，通过拨码开关输入密码，按确认后显示密码。若密码正确，箱号对应led灯闪烁；若密码错误，数码管显示“88888888”。按确认后返回欢迎界面。

系统总共分为8个模块进行设计，分别是clock\_1khz——时钟分频模块，reg16——快递箱寄存器模块，debounce——消抖模块，Moore——状态机模块，box\_ctrl——快递箱控制模块，led\_ctrl——led灯控制模块，code\_ctrl——密码控制模块，dignt\_ctrl——数码管控制模块。其下包含实现其功能的小模块。

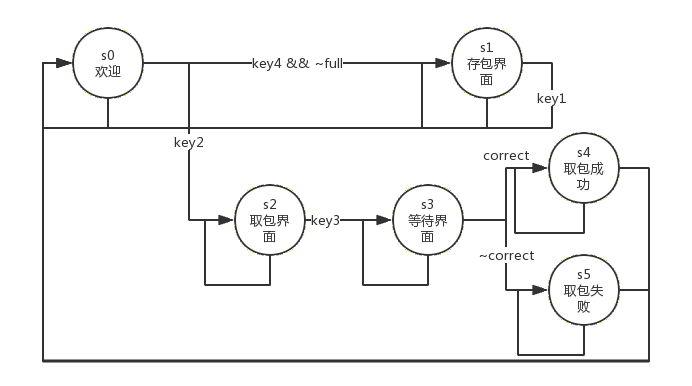
各模块关系如下：

方框图如下：



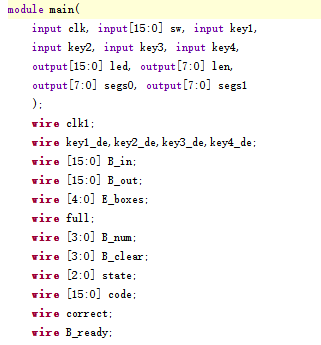
1. **状态描述及状态转换图**





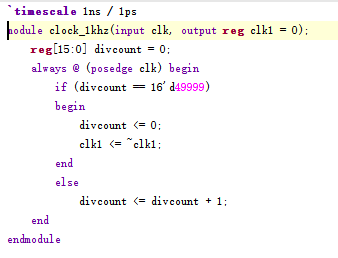
1. **模块描述**

**main 主模块**



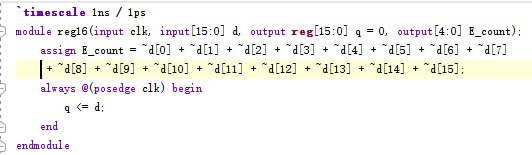
clk为时钟信号，sw为拨码开关，key1为确认，key2为取包，key3为输入，key4为存包，led为led灯，len为数码管使能开关，segs0为第一个数码管管脚，segs1为第二个数码管管脚，clk1为分频后时钟，key1\_de到key4­\_de为消抖后按键信号，B\_in为快递柜寄存器输入，B\_out为快递柜寄存器输出，full为快递柜是否满的标志，state为状态寄存器，B\_num为分配的柜号，B\_clear为要清除的柜号，code为密码，B\_ready为延时完毕的标志。

**clock\_1khz 分频模块**



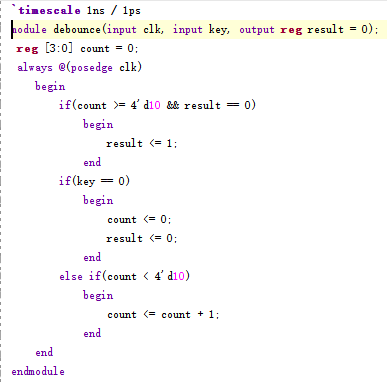
获得1khz的分频时钟

**reg16 快递柜寄存器模块**



d为输入端，q为输出端，E\_count为空箱子个数

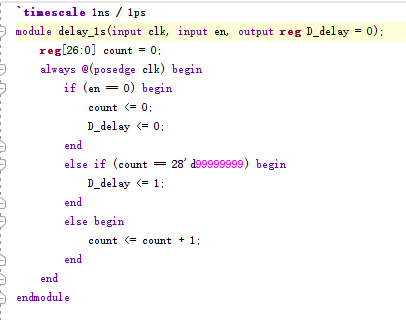
**debounce 按键消抖模块**



clk为时钟信号，key为需要消抖的按键，result为消抖后的按键信号

**Moore 状态机模块**

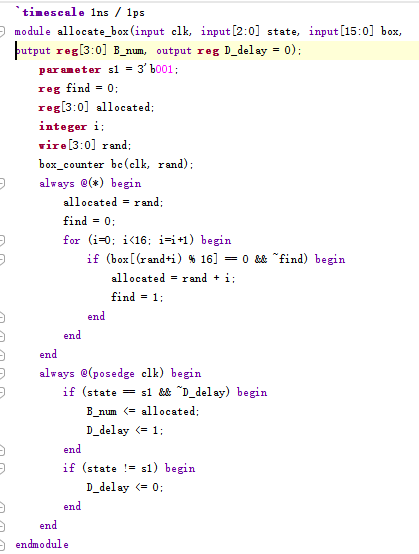
实现状态转换



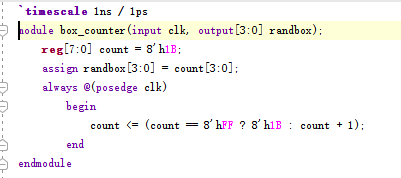
其中包含delay\_1s模块，en为使能端，D\_ready为延时完成标志

**box\_ctrl快递柜控制模块**

alloc\_ready为快递柜分配完成的标志，实现随机分配箱号，其中包含allocate\_box模块



box为快递箱寄存器输出，其中包含随机生成箱号的计数器模块box\_counter

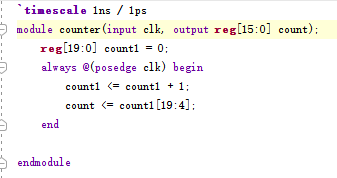


**led\_ctrl led灯控制模块**

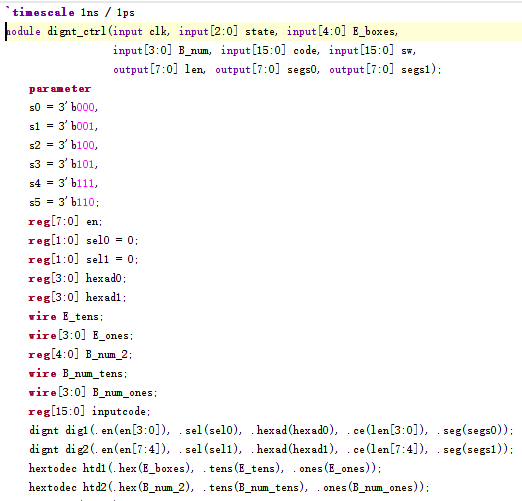
根据分配的箱号确定led的亮灭，起到类似译码器的作用

**code\_ctrl 密码控制模块**

作为随机生成密码的寄存器，randcode为随机生成的密码，input为输入的密码，box\_to\_write为写入的密码，box\_matched为随机分配的快递柜号，其下的随机分配模块及计数器与box\_ctrl类似。



**dignt\_ctrl 数码管控制模块**

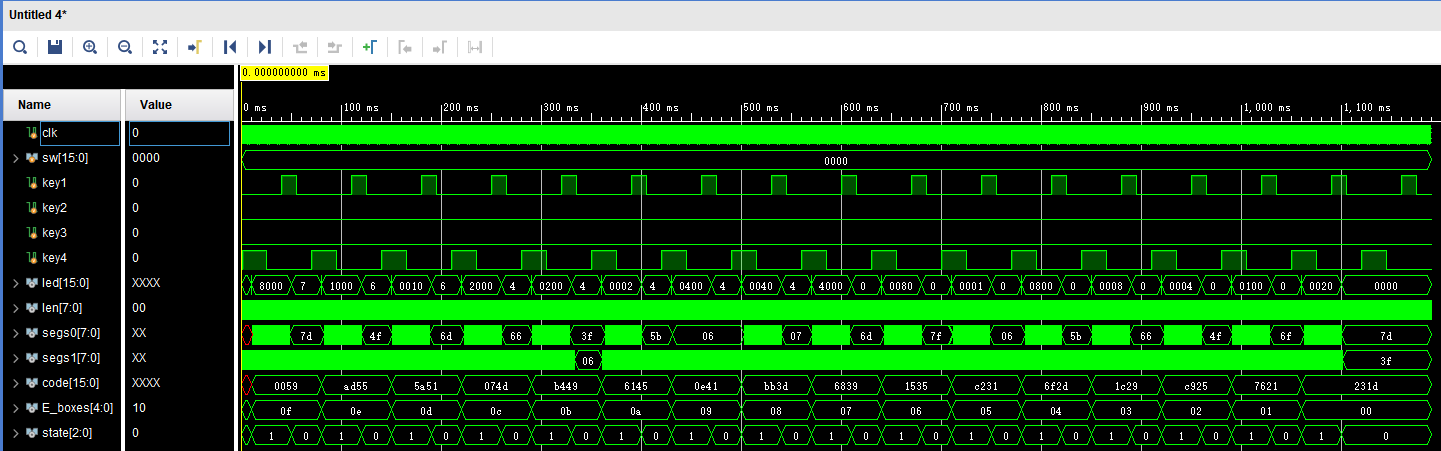


E­\_tens为空快递柜数十进制表示的十位，­E\_ones为空快递柜数十进制表示的个位，B\_num\_2为快递柜号，B\_tens为快递柜数十进制表示的十位，B\_ones为快递柜数十进制表示的个位

控制数码管显示的时序以及数码管的使能

1. **管脚分配表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **信号** | **管脚** | **信号** | **管脚** | **信号** | **管脚** | **信号** | **管脚** |
| sw[15] | P5 | sw[7] | U3 | led[15] | K3 | led[7] | K2 |
| sw[14] | P4 | sw[6] | U2 | led[14] | M1 | led[6] | J2 |
| sw[13] | P3 | sw[5] | V2 | led[13] | L1 | led[5] | J3 |
| sw[12] | P2 | sw[4] | V5 | led[12] | K6 | led[4] | H4 |
| sw[11] | R2 | sw[3] | V4 | led[11] | J5 | led[3] | J4 |
| sw[10] | M4 | sw[2] | R3 | led[10] | H5 | led[2] | G3 |
| sw[9] | N4 | sw[1] | T3 | led[9] | H6 | led[1] | G4 |
| sw[8] | R1 | sw[0] | T5 | led[8] | K1 | led[0] | F6 |
| len[7] | G6 | len[3] | H1 | segs0[7] | D5 | segs0[3] | B1 |
| len[6] | E1 | len[2] | C1 | segs0[6] | B2 | segs0[2] | A3 |
| len[5] | F1 | len[1] | C2 | segs0[5] | B3 | segs0[1] | A4 |
| len[4] | G1 | len[0] | H1 | segs0[4] | A1 | segs0[0] | B4 |
| segs1[7] | H2 | segs1[3] | F4 | key1 | R11 |  |  |
| segs1[6] | D2 | segs1[2] | D3 | key2 | R17 |  |  |
| segs1[5] | E2 | segs1[1] | E3 | key3 | V1 |  |  |
| segs1[4] | F3 | segs1[0] | D4 | key4 | U4 |  |  |

1. **调试报告**
2. 仿真界面截图及仿真数据分析

led[15:0]为led灯状态，key1为取包信号，key4为存包信号，code[15:0]为随机生成的四位十六进制密码，state[2:0]为主模块状态机的状态，E\_boxes[4:0]为剩余的快递柜数量。

可以看到，随着key4，key1信号的交替输入，状态机状态在0和1之间切换，每次状态变为1时E\_boxes中的数值减一，生成新的随机密码，led灯状态随之改变。直到key4、key1信号发出16次以后，E\_boxes中的数值减为0，不再生成新的密码，状态机状态不再跳转。

1. 出现的问题、分析原因及解决方案

在设计完成后发现仿真和上板都显示状态机状态没有跳转，经检查发现Moore状态机模块逻辑设计存在问题，是由于对三段式状态机理解不深，组合逻辑与时序逻辑混乱，在严格按照三段式状态机设计逻辑对代码进行修正后现象基本正确。多次尝试后发现随机生成的密码第一位一直为“1”，检查后发现是因为先给第一位赋了初值，没有受计数器影响，修正后现象完全正确。

1. **总结及实验课程感想**

痛苦的修炼终于结束了。Verilog这样的硬件描述语言带来比数据结构更深痛苦的原因不在于它有多难，而是你以为自己的代码完全没有问题，仿真结果也完全正确的时候，各种各样的连接错误、现象错误会接二连三地摧残你的心智。而且由于要综合、实现、生成比特流文件，一次调试需要的时间往往很长，一个小问题找不到原因可能就是好几个小时的调试。最后一次大实验体验还算不错，在机房连续9个小时的奋战后顺利通关。虽然过程很痛苦，但是完成后的快感是难以言喻的。

本门课程的实验不仅让我对于如今基于硬件编程有了一定的了解，也让我学到了模块化编程的一些知识，收获是显而易见。老师和助教们对课程的精心设计和实验中的悉心指导也给了我很大的帮助，非常感谢！