

（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：

学生班级： 计科5班

学生学号： 180110528

学生姓名： 张馨予

评阅教师：

报告成绩：

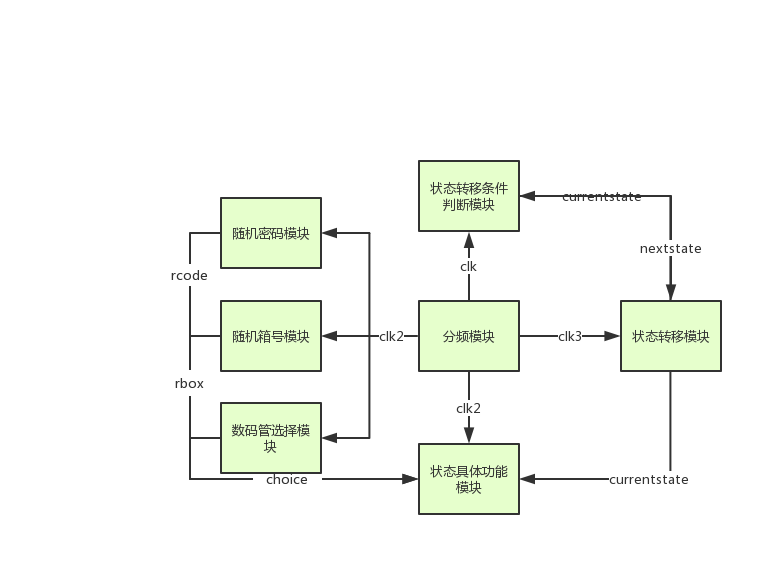
实验与创新实践教育中心制

2019年12月

1. **实验项目**

利用FPGA设计模拟一个具有16个快递箱的“快递柜”，需要实现的功能如下：

1. 欢迎界面：使用数码管前四位显示班级学号，后两位显示空箱数，使用16位LED灯代表快递箱，灯亮表示空箱。
2. 存包界面：按下存包键随机分配一个空箱，并生成一个随机密码，使用数码管前两位显示分配箱号，后四位显示分配密码，对应箱号的LED灯亮起。按下确认键表示快递柜已放入物品并关闭，回到欢迎界面。
3. 取包界面（附加题）：按下取包键进入取包界面，通过4位拨码开关从低位开始每次输入一位数，按下输入键输入并显示在数码管后四位对应的位置上，直到四位密码输入完毕，等待至少1s后判定打开成功或失败。若打开成功，数码管熄灭，对应的LED灯闪烁；若打开失败，数码管显示“88888888”。不论成功或失败，都能通过按确认键回到欢迎界面。
4. 清箱功能：按下复位键，系统还原到欢迎界面，并且将快递柜、密码清空。
5. **系统功能详细设计及实现**
6. **设计描述**



1. **状态描述及状态转换图**
2. 状态描述：

**S0：**欢迎界面；

**S1：**存放待确认；

**S2：**取出待输入第一位密码；

**S3：**等待输入第二位密码；

**S6：**等待输入第三位密码；

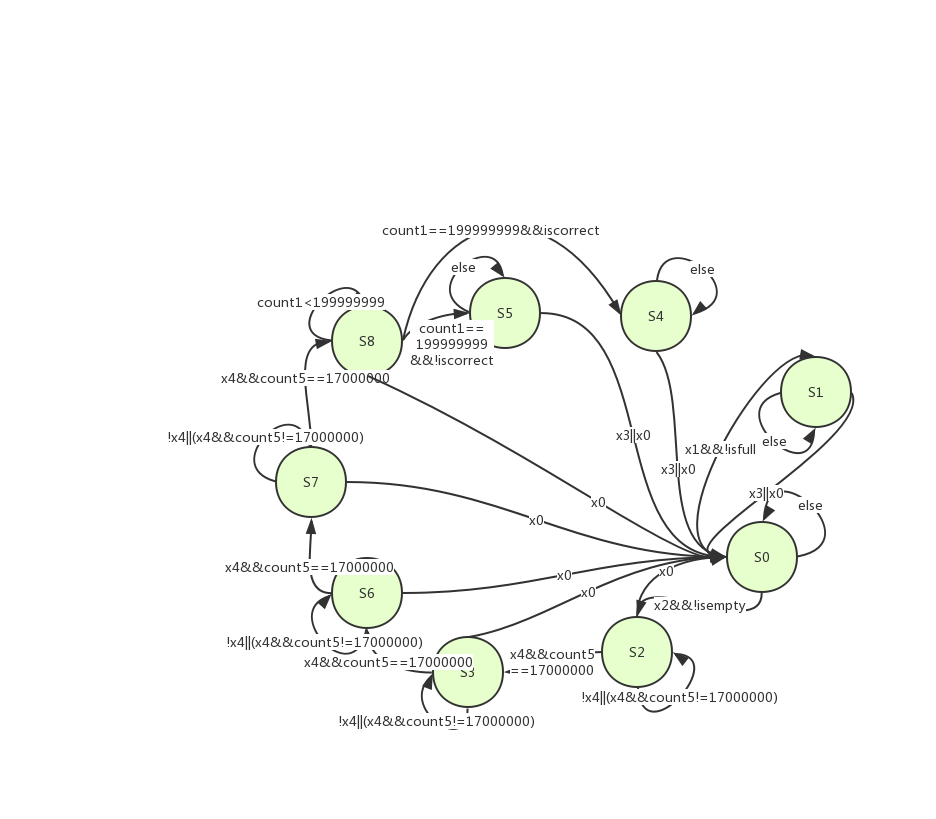
**S7：**等待输入第四位密码；

**S8：**判断等待；

**S4：**密码正确；

**S5：**密码错误；

1. 状态转换图



1. **模块描述**
2. 状态转移模块：
3. 功能：进行状态转移；
4. 变量含义：

clk3：频率为10MHz的时钟；

currentstate：当前状态；

nextstate：下一状态；

1. 状态转移条件判断模块
2. 功能：基于当前状态和输入，判断下一状态；
3. 变量含义：

clk：系统时钟；

currentstate：当前状态；

nextstate：下一状态；

x0：重置键；

x1：存包键；

x2：取包键；

x3：确认键；

x4：输入键；

count5：输入键计时，确保在取包输入密码时按下x4后发生一次状态转移；

count1：取包结果判断延时计时，确保至少等待1s后判断打开成功或者失败；

isfull：判断快递柜是否为满标志位，1代表快递柜满，0代表快递柜非满；

isempty：判断快递柜是否为空标志位，1代表快递柜空，1代表快递柜非空；

iscorrect：判断密码是否正确标志位，1代表密码正确，0代表密码错误；

1. 状态具体功能模块
2. 功能：各状态下的具体功能实现；
3. 变量含义：

clk2：10MHz的时钟；

isfull：判断快递柜是否为满标志位，1代表快递柜满，0代表快递柜非满；

isempty：判断快递柜是否为空标志位，1代表快递柜空，1代表快递柜非空；

iscorrect：判断密码是否正确标志位，1代表密码正确，0代表密码错误；

flag：分配箱子标志位，确保每次只分配一个箱子；

flag2：取物标志位，确保一次取物箱子数只加一次；

hexreg0：从左到右第1位数码管寄存位；

hexreg1：从左到右第2位数码管寄存位；

hexreg2：从左到右第3位数码管寄存位；

hexreg3：从左到右第4位数码管寄存位；

hexreg4：从左到右第5位数码管寄存位；

hexreg5：从左到右第6位数码管寄存位；

hexreg6：从左到右第7位数码管寄存位；

hexreg7：从左到右第8位数码管寄存位；

en：设置数码管亮灭情况；

len：控制数码管整体亮灭的使能位；

segs0：控制左四位数码管；

segs1：控制右四位数码管；

choice：数码管选择；

box：记录箱子的空余情况；

boxnum：记录空箱数；

boxnow：密码的遍历变量，从而判断输入的密码是否正确；

led：控制led的亮灭；

code：记录所有密码；

incode0：从右到左用户输入密码的第1位；

incode1：从右到左用户输入密码的第2位；

incode2：从右到左用户输入密码的第3位；

incode3：从右到左用户输入密码的第4位；

srcode：记录该次取包的随机密码值；

rcode：密码随机变量；

srbox：记录该次取包的随机箱号；

rbox：箱号随机变量；

1. 随机箱号模块
2. 功能：生成随机箱号；
3. 变量：

clk2：频率为10MHz的时钟；

countbox：随机箱号计数器的计数变量，按照题目要求范围为[28,255]；

midbox：第一版箱号随机变量，尚未剔除非空的箱号；

rbox：箱号随机变量；

1. 随机密码模块
2. 功能：生成随机密码；
3. 变量：

clk2：频率为10MHz的时钟；

rcode：密码随机变量；

1. 分频模块
2. 功能：对系统时钟进行分频，得到需要的时钟；
3. 变量：

clk：系统时钟；

count：clk1对应的计数变量；

clk1：频率为1kHz的时钟；

count2：clk2对应的计数变量；

clk2：频率为10MHz的时钟；

count3：clk3对应的计数变量；

clk3：频率为10MHz的时钟；

1. 数码管选择模块
2. 功能：对数码管选择变量choice进行循环赋值；
3. 变量：

clk1：频率为1kHz的时钟；

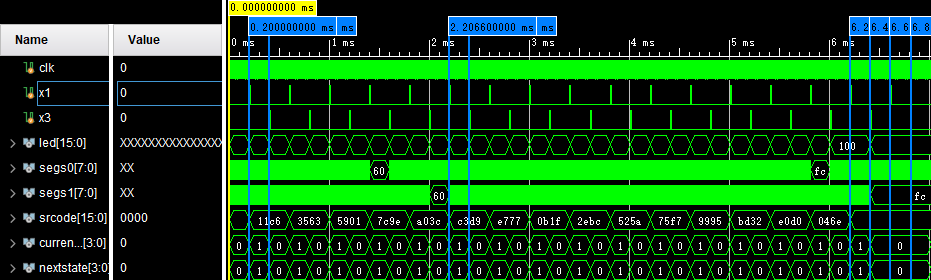
choice：数码管选择；

1. **管脚分配表**

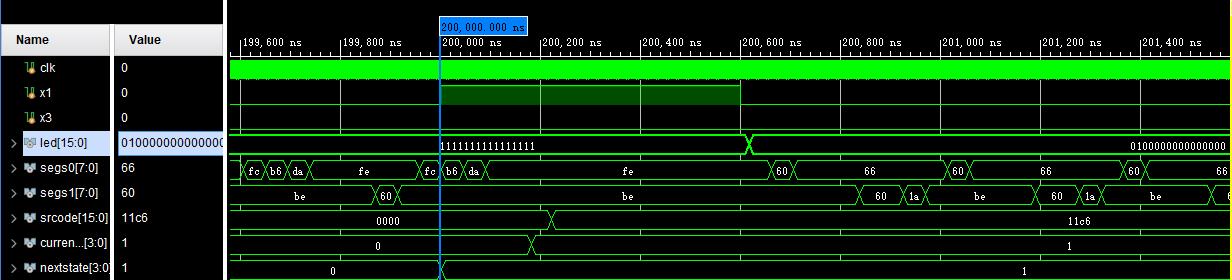
|  |  |
| --- | --- |
| 名称 | IO PIN |
| clk | P17 |
| hex0[0] | P2 |
| hex0[1] | P3 |
| hex0[2] | P4 |
| hex0[3] | P5 |
| x0 | R17 |
| x1 | R15 |
| x2 | V1 |
| x3 | U4 |
| x4 | R11 |
| segs0[0] | B4 |
| segs0[1] | A4 |
| segs0[2] | A3 |
| segs0[3] | B1 |
| segs0[4] | A1 |
| segs0[5] | B3 |
| segs0[6] | B2 |
| segs0[7] | D5 |
| segs1[0] | D4 |
| segs1[1] | E3 |
| segs1[2] | D3 |
| segs1[3] | F4 |
| segs1[4] | F3 |
| segs1[5] | E2 |
| segs1[6] | D2 |
| segs1[7] | H2 |
| len[0] | G2 |
| len[1] | C2 |
| len[2] | C1 |
| len[3] | H1 |
| len[4] | G1 |
| len[5] | F1 |
| len[6] | E1 |
| len[7] | G6 |
| led[0] | K3 |
| led[1] | M1 |
| led[2] | L1 |
| led[3] | K6 |
| led[4] | J5 |
| led[5] | H5 |
| led[6] | H6 |
| led[7] | K1 |
| led[8] | K2 |
| led[9] | J2 |
| led[10] | J3 |
| led[11] | H4 |
| led[12] | J4 |
| led[13] | G3 |
| led[14] | G4 |
| led[15] | F6 |

1. **调试报告**
2. **仿真界面截图及数据分析**

下图为仿真调试的总况，共有七个标定点，分别标注了第一次存包确认、第六次存包确认、第十六次存包确认和第十七次存包；



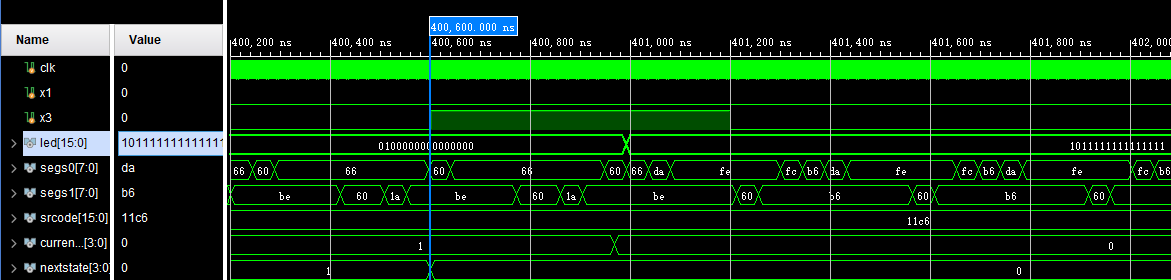
下图为第一次存包，



按下存包键前，led=”1111\_1111\_1111\_1111”, segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”60”(1) --- “be”(6),currentstate=0;//初始状态快递柜全空、数码管左四位显示班级学号0528，右两位显示空箱数16

按下存包键后，led=”0100\_0000\_0000\_0000”,segs0=”60”(1)---“66”(4),segs1=”60”(1)---”60”(1)—“1a”(c)---“be”(6),srcode=”11c6”,currentstate=1;//分配第14号箱、数码管左两位显示箱号，右四位显示密码11c6

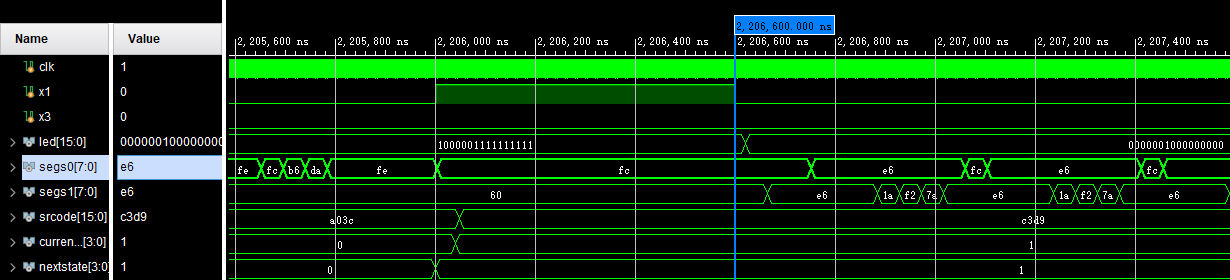
下图为第一次确认，



按下确认键前，led=”0100\_0000\_0000\_0000”,segs0=”60”(1)---“66”(4),segs1=”60”(1)---”60”(1)—“1a”(c)---“be”(6),srcode=”11c6”,currentstate=1;

按下确认键后，led=”1011\_1111\_1111\_1111”,segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”60”(1) --- “b6”(5),currentstate=0;//第14号箱非空，空箱数变为15

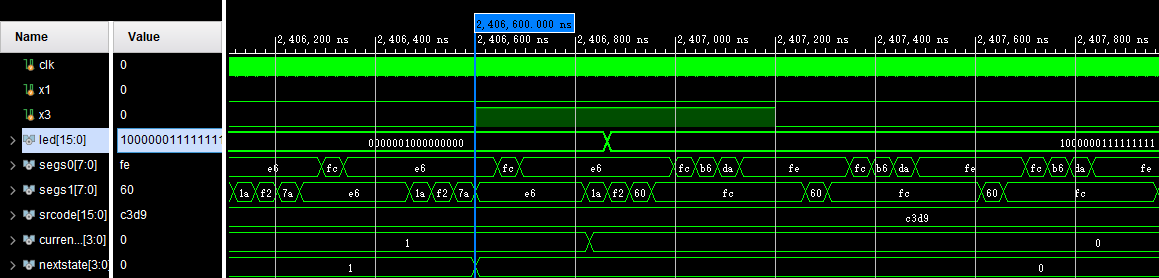
下图为第六次存包，



按下存包键前，led=”1000\_0011\_1111\_1111”, segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”60”(1) --- “60”(1),currentstate=0；//已存过五次包，空箱数11,

按下存包键后，led=”0000\_0010\_0000\_0000”,segs0=”fc”(0)---“e6”(9),segs1=”1a”(c)---”f2”(3)—“7a”(d)---“e6”(9),srcode=”c3d9” ,currentstate=1;//分配第9号箱，密码c3d9

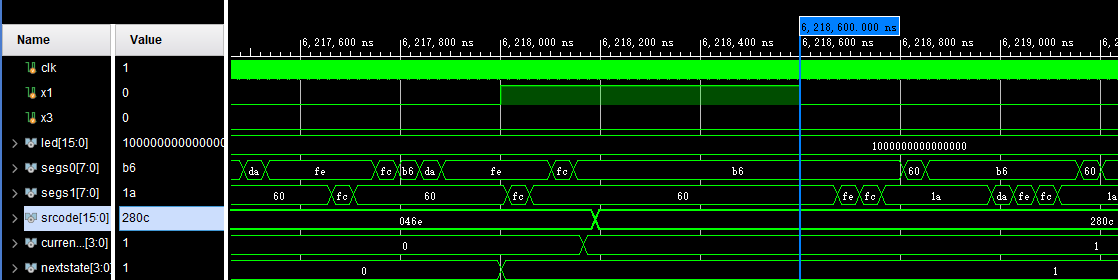
下图为第六次确认，



按下确认键前，led=”0000\_0010\_0000\_0000”,segs0=”fc”(0)---“e6”(9),segs1=”1a”(c)---”f2”(3)—“7a”(d)---“e6”(9),srcode=”c3d9”,currentstate=0;

按下确认键后，led=”1000\_0001\_1111\_1111”,segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”60”(1) --- “fc”(0) ,currentstate=0;//第9号箱非空，空箱数变成10

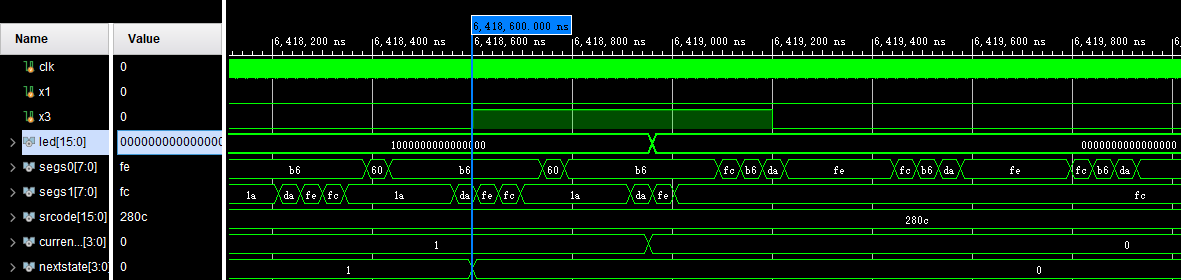
下图为第十六次存包，



按下存包键前，led=”1000\_0000\_0000\_0000”, segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”fc”(0) --- “60”(1),currentstate=0；//已存过十五次包，空箱数1

按下存包键后，led=”1000\_0000\_0000\_0000”,segs0=”60”(1)---“b6”(5),segs1=”da”(2)---”fe”(8)—“fc”(0)---“1a”(c),srcode=”280c” ,currentstate=1;//分配第15号箱，密码280c

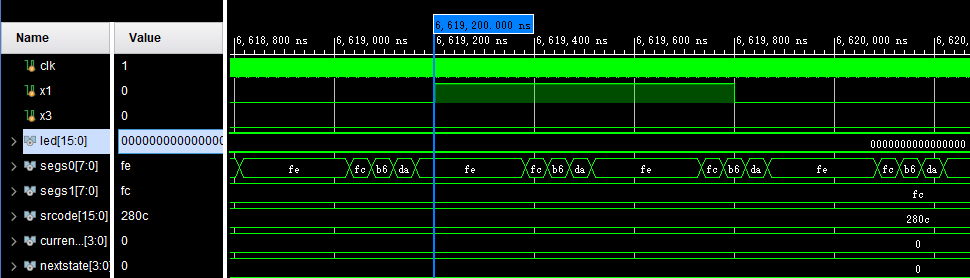
下图为第十六次确认，



按下确认键前，led=”1000\_0000\_0000\_0000”,segs0=”60”(1)---“b6”(5),segs1=”da”(2)---”fe”(8)—“fc”(0)---“1a”(c),srcode=”280c” ,currentstate=1;

按下确认键后，led=”0000\_0000\_0000\_0000”,segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”fc”(0) --- “fc”(0) ,currentstate=0;//快递柜已满

下图为第十七次存包，



按下存包键前，led=”0 000\_0000\_0000\_0000”, segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”fc”(0) --- “fc”(0),currentstate=0；//快递柜已满

按下存包键后，led=”0 000\_0000\_0000\_0000”, segs0=”fc”(0) --- “b6”(5) --- “da”(2) --- “fe”(8),segs1=”fc”(0) --- “fc”(0),currentstate=0；//不能再存包

1. **出现的问题、分析原因及解决方案**
2. 问题：初始状态为存包界面；

分析原因：时序错误，导致状态混乱；

解决方案：对触发条件进行调整。

1. 问题：取包成功后，led无法闪烁；

分析原因：通过观察仿真图后，发现频闪频率设置错误，要么过快使得肉眼无法分辨出闪烁，要么过慢使得有限时间内观察不到亮起；

解决方案：调整频闪频率。

1. 问题：在做附加题的过程中，发现每当应跳转到等待判断状态S8时，总会跳转到初始状态S0；

分析原因：状态变量位数错误，使得S8的最高位被舍去；

解决方案：调整状态变量位数。

1. **总结及实验课程感想**
2. 综合实验总结

本次综合实验对我而言很有难度，耗费了非常多时间，难点如下：

1. 要求对之前所学的知识充分掌握，并能一定程度的灵活运用，从而实现题目要求的功能；
2. 要求对硬件语言的特性有所理解，尤其是时序问题；

复习过前几次实验内容、并认真检查书写代码逻辑后，尽管花了较长时间，但能够解决第一个难点，但因为存在时序问题，因此“我的FPGA有了自己的想法”。起初我所用的触发条件并非同步，且存在多种触发方式，因此我将其均改为了系统时钟触发的同步时序，但仍然得不到正确的结果。在花费了大量时间，仔细观察仿真图和大胆尝试后，我调整了部分模块的触发时钟频率，奇迹般的解决了时序问题……

客观来讲，附加题部分不是很难，但我仍然出了状态转移的问题，花费两个小时后终于找到了问题所在：我将状态数量增加到了9个，但是却没有修改状态变量的位数！因此就产生了每次跳转到S8（1000）实际上都跳转到了S0（0000）的神奇现象。非常无语。

综上，我认为导致我这次实验完成得如此艰难的主要原因：

1. 没有模块意识，因此使得代码逻辑不够清晰；
2. 没有模块意识，因此可能产生了难以发现的时序问题；
3. 基础不够扎实、逻辑不够严谨，因此会犯一些非常智障的错误。
4. 课程感想

虽然课程本身很有难度，我花费了若干个在机房怀疑自己怀疑人生怀疑FPGA坏掉了的日日夜夜，还收获了在机房刷完夜吃个早饭回去补觉的人生体验。但我确实还是挺喜欢这门课程的：最后看着手上终于正常了的FPGA着实有成就感。总而言之，它帮助我加深了对数字逻辑理论知识的理解，了解了FPGA开发的过程，同时树立了信心——我其实对实验课颇有恐惧，因为我心态很容易爆炸解决问题能力也不是很强。但不管怎么样，我又成功熬过了一个实验课。数据结构实验、大学物理实验、软件开发实验、硬件开发实验、汇编语言实验、数字逻辑实验……每一个都带给我不甚美妙的心灵体验，但我到目前为止都还没被过程淘汰。我也逐渐收获了一个经验：不必太紧张，只要付出了努力，事情不会像自己想象的那么难、结果不会像自己想象的那么坏。尽管对未来的实验仍然存在恐惧，但是我会努力尝试从更积极、自信的角度去看待。期望有一天能真正掌握、享受它们。

最后，非常感谢老师助教提供耐心且有效的帮助！