湖南科技大学计算机科学与工程学院

考查课程报告

**课程名称：** 数字系统设计

**专 业：** 物联网工程

**班 级：** 一班

**姓 名：** 2305040107 曹烨（组长）

**姓 名：** 2305040114 贺兵

**授课教师：** 夏新军

|  |
| --- |
| 授课教师评语：    **成绩： 等级：**  **签名：**  **年 月 日** |

**1、题目分析**

**选购模式：**

1. 6个数码管代表**6种商品**，每种商品数量为0-9；
2. 6个按键代表对应商品的**选择按钮**，按下表示选择购买，只有该**商品数量**大于0是可以购买，为零时提示**缺货**，用一个LED灯指示；
3. 使用一组LED显示选中商品的价格，并开始**计时**（用剩下两个数码管显示倒计时），60秒内没有收到成功**付款**（扫码付款）信号，则取消该交易，恢复成选择购买前状态；
4. 用剩下按键给出成功付款信号，收到该信号，倒计时结束，并**出货**（用led等指示已出货），该**商品数量减少**。

**补货模式：**

(1) 一个机械按键控制**补货**，按下后进行一键补货。

**2、具体分工**

曹烨（组长）：负责

1. 倒计时模块，数码管显示模块
2. 补货模块，支付模块
3. 流程图的设计与绘制，PPT的制作
4. 代码规范化、注释和改进

贺兵：负责

1. 按键的消抖操作，选购模块
2. 总价显示模块，缺货提醒
3. 仿真波形的设计，PPT的制作
4. 代码规范化、注释和改进

**3、模块划分**

(1) 倒计时模块：选择100KHZ时钟clk2信号进行分频，当出现上升沿时，若按键按下且未支付，则开始计时。使用一个一秒计数器的信号进行计时，倒计时初始值为60s。

(2) 数码管显示模块：选择100KHZ的时钟clk2信号作为进程敏感信号，定义八个变量存储段选数值，段选数值通过数组进行存储。当出现上升沿时，读取对应变量的数码管段选数值，之后进行动态扫描。

(3) 选购模块：当1000HZ的时钟clk3出现上升沿时，若倒计时结束或者已支付货物，将选购数量全部清零，且停止计时。K7-K2按键按下进行消抖后，相应的选购数量+1。1个商品的选购数量由4个LED显示。

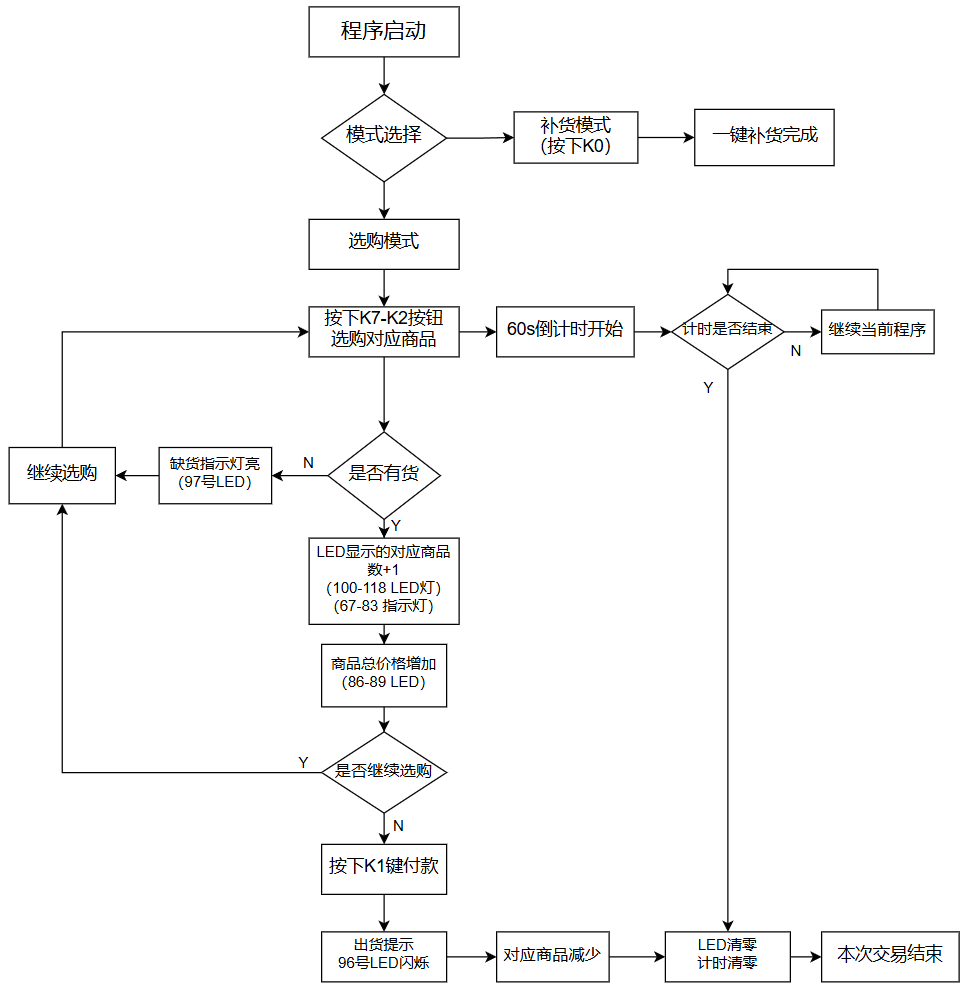
(4) 显示模块：用一个整数变量记录所有选购商品的数值，并通过四位二进制数展示出来。用多个LED灯展示选购的商品及其数量。

(5) 支付模块：选择1000HZ的时钟clk3信号作为进程敏感信号，当出现上升沿时，若K1按键按下则进行消抖，出货LED灯闪烁，并减少对应的商品数量

(6) 补货模块：当按下K0按键后，若处于空闲状态（不处于选购模式中）则进行一键补货。

**4、具体实现**

流程图展示：



端口定义：

ENTITY VEM IS

PORT(

clk2,clk3,clk4 : in std\_logic;--时钟

key0,key1,key2,key3,key4,key5,buy :in std\_logic;--六个选购商品价格和付款按键

supple : in std\_logic;--补货按键

money: out std\_logic\_vector (3 downto 0);--总费用LED显示

LackGoods,OutputGoods:out std\_logic; --缺货和出货LED

sel : out STD\_LOGIC\_VECTOR(7 downto 0); -- 位选信号

SEG : out STD\_LOGIC\_VECTOR(7 downto 0); -- 段选信号

goods0,goods1,goods2,goods3,goods4,goods5 : out std\_logic\_vector(3 downto 0);

--商品数量LED

);

END ENTITY;

结构体变量定义：

architecture shop\_machine OF VEM IS

signal goodsnum0,goodsnum1,goodsnum2,goodsnum3,goodsnum4,goodsnum5:Integer range 9 downto 0 :=9;--商品数量（数码管显示）

signal g0,g1,g2,g3,g4,g5:Integer range 8 downto 0 :=0;--选中的商品数量

signal gds : Integer range 7 downto 0;--控制八位数码管的动态扫描

signal tflag:Integer range 1 downto 0 :=0;--计时标志

signal count0,count1,count2,count3, count4,count5, countb, countok:Integer :=0;

--按键计数器（用于按键消抖）

type SEG\_ARRAY is array (0 to 9) of STD\_LOGIC\_VECTOR(7 downto 0);--数码管显示表

constant SEGMENTS : SEG\_ARRAY := (

"11111100", -- 数字 0

"01100000", -- 数字 1

"11011010", -- 数字 2

"11110010", -- 数字 3

"01100110", -- 数字 4

"10110110", -- 数字 5

"10111110", -- 数字 6

"11100000", -- 数字 7

"11111110", -- 数字 8

"11110110" -- 数字 9

);

signal countdown : INTEGER range 1 to 60 := 60; -- 倒计时初始值

signal digit\_values : STD\_LOGIC\_VECTOR(7 downto 0); -- 数码管段选信号

signal one\_second\_counter : INTEGER range 0 to 99999 := 0; -- 1 秒计数器

signal mode : INTEGER RANGE 0 TO 1 := 0;

signal time\_out : INTEGER RANGE 0 TO 1 :=0;

signal ready\_pay : INTEGER RANGE 0 TO 1 :=0 ;

倒计时模块：

process(clk2)--分频逻辑

begin

if rising\_edge(clk2) then

if ready\_pay = 0 and tflag = 1 then

mode <= 0;

-- 分频至 1 Hz

if one\_second\_counter = 99999 then

one\_second\_counter <= 0;

if countdown > 0 then

countdown <= countdown - 1; -- 倒计时递减

else

mode <= 1;

countdown <= 60; -- 重新开始倒计时

end if;

else

one\_second\_counter <= one\_second\_counter + 1;

end if;

else

countdown <= 60;

mode <= 1;

if ready\_pay = 1 then

time\_out <= 1;

else

time\_out <= 0;

end if;

end if;

end if;

end process;

数码管显示模块：

process(clk2)--数码管

variable gn0,gn1,gn2,gn3,gn4,gn5,gn6,gn7 : std\_logic\_vector(7 downto 0);

begin

if rising\_edge(clk2) then

gn0:=SEGMENTS(goodsnum0);

gn1:=SEGMENTS(goodsnum1);

gn2:=SEGMENTS(goodsnum2);

gn3:=SEGMENTS(goodsnum3);

gn4:=SEGMENTS(goodsnum4);

gn5:=SEGMENTS(goodsnum5);

gn6:=SEGMENTS(countdown mod 10);

gn7:=SEGMENTS(countdown /10);

case gds is

when 0=>sel<="11111110";

when 1=>sel<="11111101";

when 2=>sel<="11111011";

when 3=>sel<="11110111";

when 4=>sel<="11101111";

when 5=>sel<="11011111";

when 6=>sel<="10111111";

when 7=>sel<="01111111";

end case;

case gds is

when 0=>digit\_values<=gn7;

when 1=>digit\_values<=gn5;

when 2=>digit\_values<=gn4;

when 3=>digit\_values<=gn3;

when 4=>digit\_values<=gn2;

when 5=>digit\_values<=gn1;

when 6=>digit\_values<=gn0;

when 7=>digit\_values<=gn6;

end case;

seg <= digit\_values;

gds<=(gds+1)mod 8 ;

end if;

end process;

选购模块：

process(key0,key1,key2,key3,key4,key5)

begin

if rising\_edge(clk3) then

if(mode = 1) then

g0<= 0;

g1<= 0;

g2<= 0;

g3<= 0;

g4<= 0;

g5<= 0;

tflag <= 0;

end if;

if(key0 = '0') then

count0 <= count0 + 1;

else count0 <= 0;

end if;

if(count0 = 100) then

if goodsnum0 >= g0 then

g0<=(g0+1) mod (goodsnum0+1);

if tflag = 0 then

tflag <=1;

end if;

end if;

end if;

if(key1 = '0') then

count1 <= count1 + 1;

else count1 <= 0;

end if;

if(count1 = 100) then

if goodsnum1 >= g1 then

g1<=(g1+1) mod (goodsnum1+1);

if tflag = 0 then

tflag <=1;

end if;

end if;

end if;

if(key2 = '0') then

count2 <= count2 + 1;

else count2 <= 0;

end if;

if(count2 = 100) then

if goodsnum2 >= g2 then

g2<=(g2+1) mod (goodsnum2+1);

if tflag = 0 then

tflag <=1;

end if;

end if;

end if;

if(key3 = '0') then

count3 <= count3 + 1;

else count3 <= 0;

end if;

if(count3 = 100) then

if goodsnum3 >= g3 then

g3<=(g3+1) mod (goodsnum3+1);

if tflag = 0 then

tflag <=1;

end if;

end if;

end if;

if(key4 = '0') then

count4 <= count4 + 1;

else count4 <= 0;

end if;

if(count4 = 100) then

if goodsnum4 >= g4 then

g4<=(g4+1) mod (goodsnum4+1);

if tflag = 0 then

tflag <=1;

end if;

end if;

end if;

if(key5 = '0') then

count5 <= count5 + 1;

else count5 <= 0;

end if;

if(count5 = 100) then

if goodsnum5 >= g5 then

g5<=(g5+1) mod (goodsnum5+1);

if tflag = 0 then

tflag <=1;

end if;

end if;

end if;

end if;

end process;

LED显示模块：

process(CLK2)--总价

variable res : INTEGER range 15 downto 0 :=0;

begin

res :=g0+g1+g2+g3+g4+g5;

money <= CONV\_STD\_LOGIC\_VECTOR(res,4);

end process;

process(clk4)--购买显示模块

begin

if rising\_edge(clk4) then

case g0 is

when 0=> goods0<="0000";

when 1=> goods0<="0001";

when 2=> goods0<="0010";

when 3=> goods0<="0011";

when 4=> goods0<="0100";

when 5=> goods0<="0101";

when 6=> goods0<="0110";

when 7=> goods0<="0111";

when 8=> goods0<="1000";

end case;

case g1 is

when 0=> goods1<="0000";

when 1=> goods1<="0001";

when 2=> goods1<="0010";

when 3=> goods1<="0011";

when 4=> goods1<="0100";

when 5=> goods1<="0101";

when 6=> goods1<="0110";

when 7=> goods1<="0111";

when 8=> goods1<="1000";

end case;

case g2 is

when 0=> goods2<="0000";

when 1=> goods2<="0001";

when 2=> goods2<="0010";

when 3=> goods2<="0011";

when 4=> goods2<="0100";

when 5=> goods2<="0101";

when 6=> goods2<="0110";

when 7=> goods2<="0111";

when 8=> goods2<="1000";

end case;

case g3 is

when 0=> goods3<="0000";

when 1=> goods3<="0001";

when 2=> goods3<="0010";

when 3=> goods3<="0011";

when 4=> goods3<="0100";

when 5=> goods3<="0101";

when 6=> goods3<="0110";

when 7=> goods3<="0111";

when 8=> goods3<="1000";

end case;

case g4 is

when 0=> goods4<="0000";

when 1=> goods4<="0001";

when 2=> goods4<="0010";

when 3=> goods4<="0011";

when 4=> goods4<="0100";

when 5=> goods4<="0101";

when 6=> goods4<="0110";

when 7=> goods4<="0111";

when 8=> goods4<="1000";

end case;

case g5 is

when 0=> goods5<="0000";

when 1=> goods5<="0001";

when 2=> goods5<="0010";

when 3=> goods5<="0011";

when 4=> goods5<="0100";

when 5=> goods5<="0101";

when 6=> goods5<="0110";

when 7=> goods5<="0111";

when 8=> goods5<="1000";

end case;

end if;

end process;

process(key0,key1,key2,key3,key4,key5)--缺货情况显示

begin

if key0 = '0' and goodsnum0=0 then

LackGoods<='1';

elsif key1 = '0' and goodsnum1=0 then

LackGoods<='1';

elsif key2 = '0' and goodsnum2=0 then

LackGoods<='1';

elsif key3 = '0' and goodsnum3=0 then

LackGoods<='1';

elsif key4 = '0' and goodsnum4=0 then

LackGoods<='1';

elsif key5 = '0' and goodsnum5=0 then

LackGoods<='1';

else

LackGoods<='0';

end if;

end process;

支付模块和补货模块：

process(clk3)--准备出货

begin

if rising\_edge(clk3) then

if(buy = '0') then

countb <= countb + 1;

else

countb <= 0;

ready\_pay <= 0;

outputGoods <= '0';

end if;

if(countb = 100) then

if countdown > 0 then

ready\_pay <= 1;

outputGoods <= '1';

end if;

end if;

if ready\_pay = 1 and time\_out = 1 then

goodsnum0<=(goodsnum0-g0);

goodsnum1<=(goodsnum1-g1);

goodsnum2<=(goodsnum2-g2);

goodsnum3<=(goodsnum3-g3);

goodsnum4<=(goodsnum4-g4);

goodsnum5<=(goodsnum5-g5);

end if;

if(supple = '0' and mode = 1) then

goodsnum0 <= 9;

goodsnum1 <= 9;

goodsnum2 <= 9;

goodsnum3 <= 9;

goodsnum4 <= 9;

goodsnum5 <= 9;

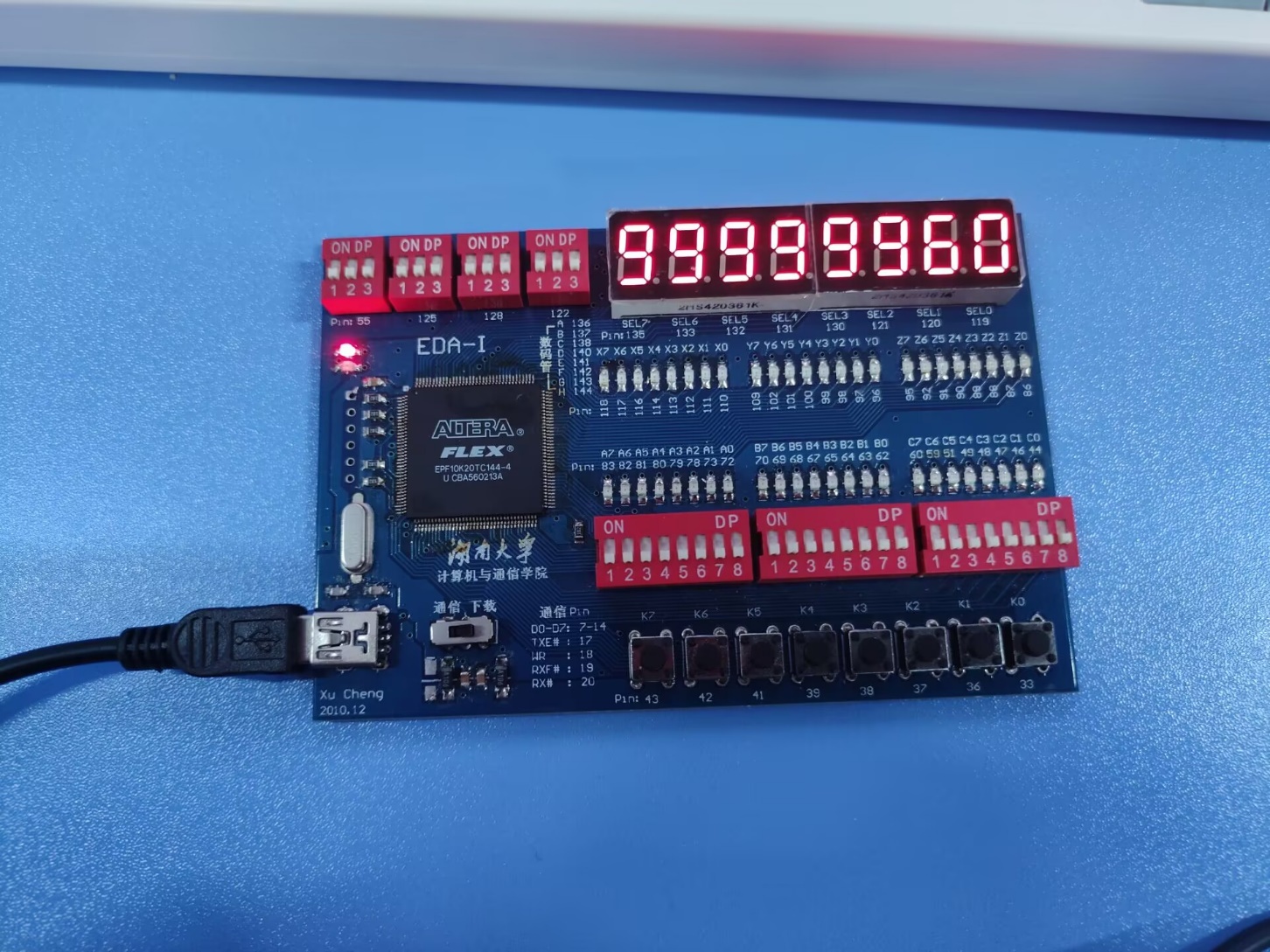
end if;

end if;

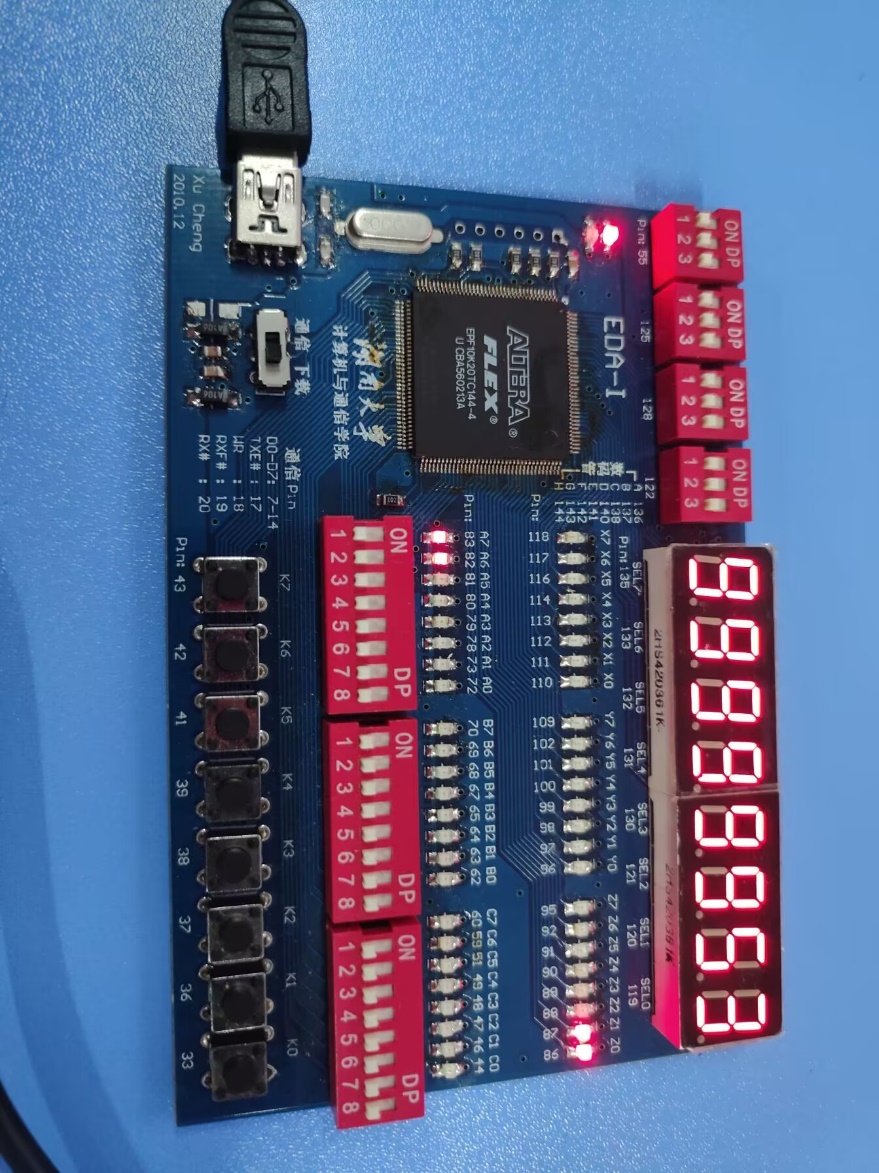
end process;

**5、运行结果截图**

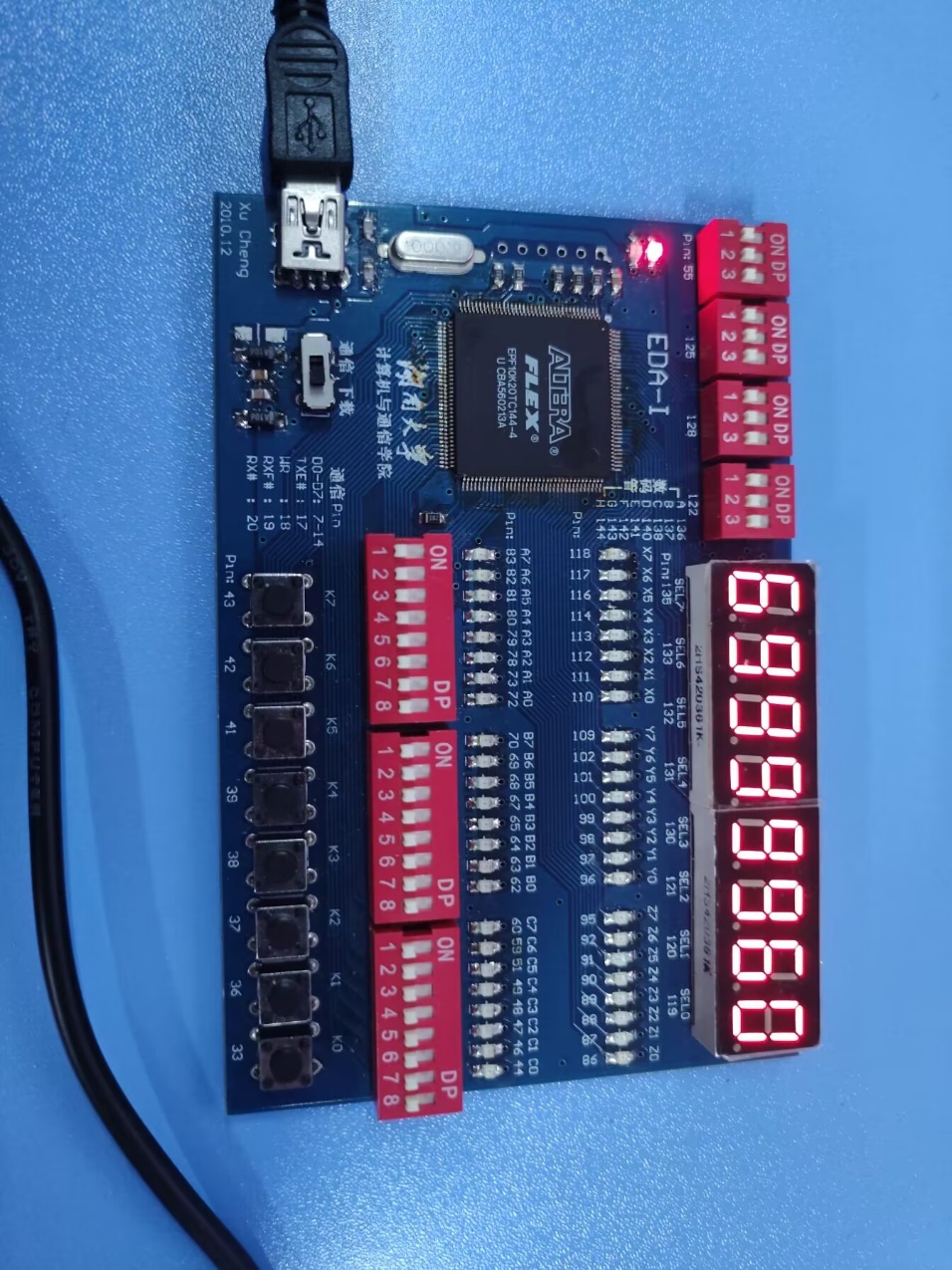
开始：



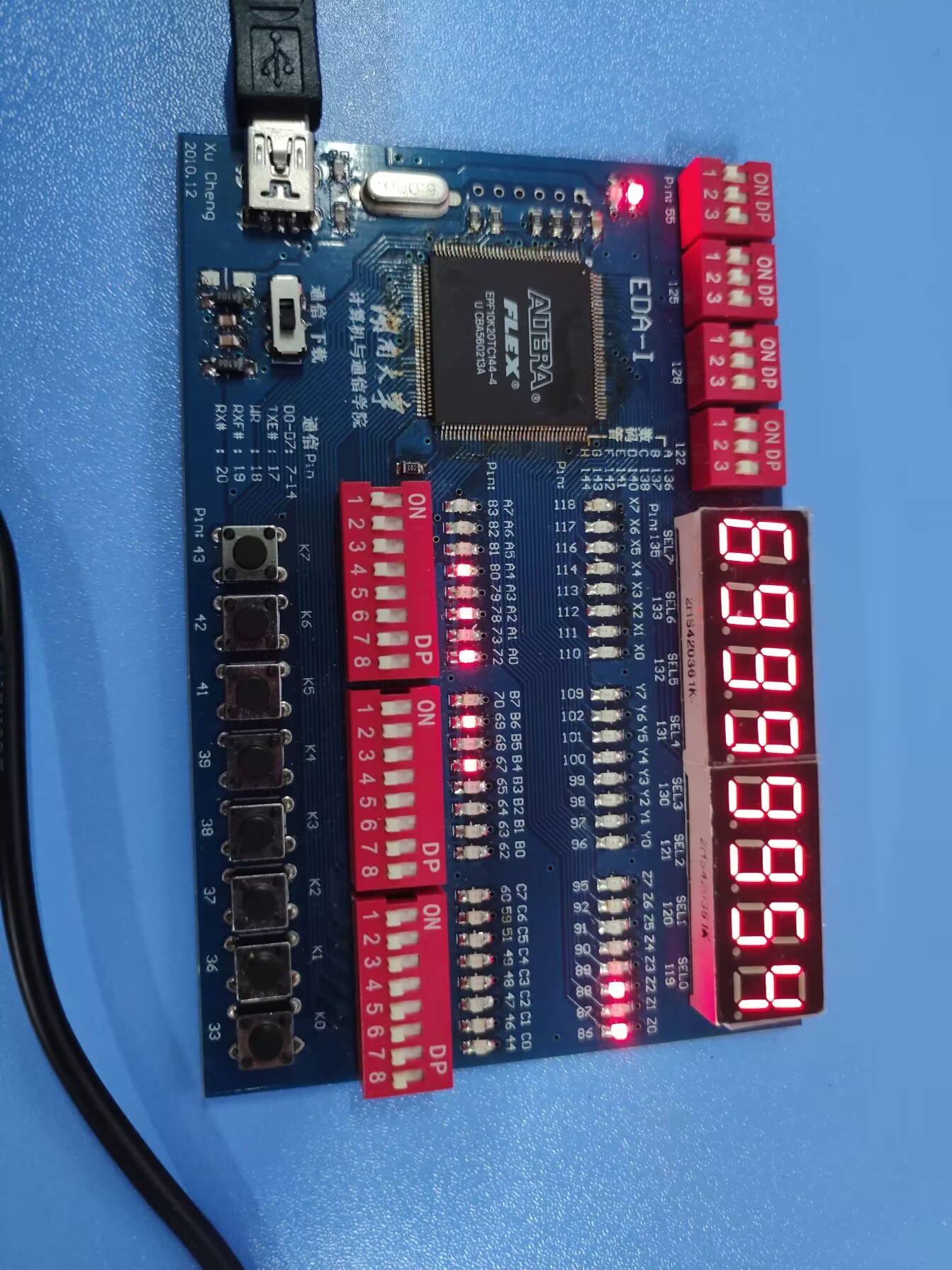
购买3个1号商品：



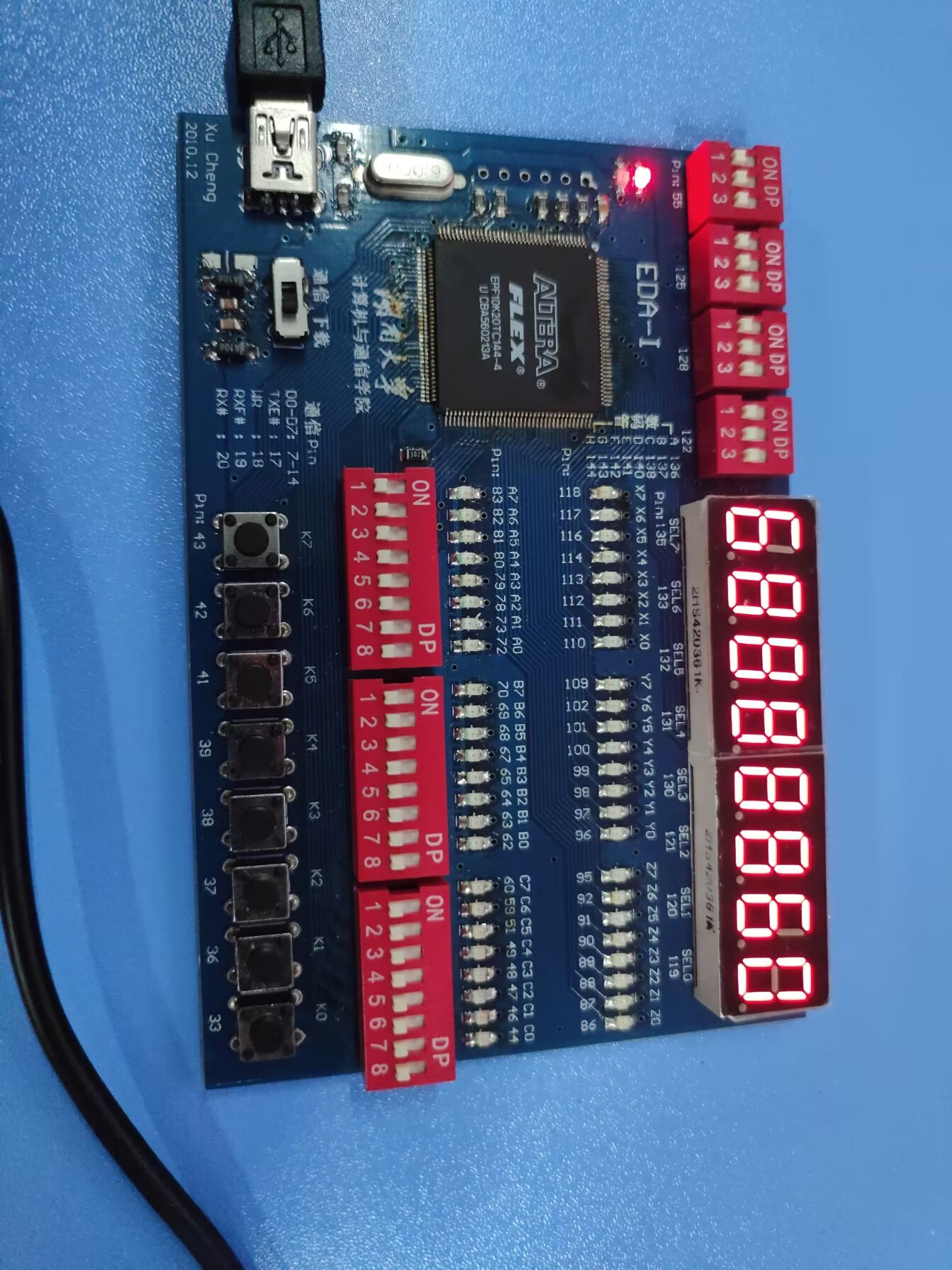
付款后：



购买1个2,3,4,5,6号商品：



付款后：



结果分析：

1. 选购按键通过消抖后可以准确计算选购数量，并通过对应的LED灯进行显示。
2. 倒计时结束后选购数量、总价清零重置。
3. 在选购模式中，一键补货按键无效，倒计时结束或付款后才可以使用。

**6、心得体会**

在此次自动售货机的EDA设计中，我们对以下的知识点有了深刻的了解：

1. 数码管的动态刷新。
2. 使用VHDL语言对机械按键进行消抖。
3. 进程与进程之间是并行的，进程内部是顺序执行的。
4. LED的使用。

易错点总结：

1. 结构体中定义的信号只能在一个进程里进行修改，但可以在多个进程中进行判断。
2. 数码管的动态刷新，取模的用处。
3. 由于VHDL语言是严格类型语言，所以赋值时要注意变量类型。
4. 变量定义需要限定具体的范围，且尽量先赋初值。
5. 时序冲突和资源分配。