输入端口包括clk、coin1（投入一元货币）、coin5（投入0.5元货币）、cola（选择cola）、pepsi(选择pepsi)，输出端口有paid（已投入多少钱）、needed（还需多少钱）moneyout（找零）、success（灯亮表示交易成功）、failure（灯亮表示交易失败）、showmoneyout（灯亮表示正在找零）。

1.1.3 程序设计

主控模块完整程序如下：

Library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity shouhuoji is --定义实体

port(clk: in std\_logic; --定义端口

coin1:in std\_logic; --投入1元货币

coin5:in std\_logic; --投入5毛货币

cola:in std\_logic; --选择cola

pepsi:in std\_logic; --选择可乐

paid:out std\_logic\_vector(7 downto 0); --已投入多少钱

needed:out std\_logic\_vector(7 downto 0); --还需要多少钱

success:out std\_logic; --灯亮表示出货成功

failure:out std\_logic; --交易失败

showmoneyout:out std\_logic; --正在找零钱

moneyout:out std\_logic\_vector(7 downto 0)); --找零

end shouhuoji;

architecture behav of shouhuoji is --结构体

type state\_type is (qa,qb,qe,qc,qg,qd,qf);--定义七个状态

signal current\_state :state\_type:=qa; --现在的情形

signal q:integer range 0 to 100; --整数判别区间

begin

process(clk) --进程

variable paidtemp:std\_logic\_vector(7 downto 0); --定义变量，现在付款金额

variable neededtemp:std\_logic\_vector(7 downto 0); --现在还需要多少钱

variable backmoney:std\_logic\_vector(7 downto 0); --找钱

variable pricetemp:std\_logic\_vector(7 downto 0); --物品的价格

begin

if clk'event and clk='1' then --如果事件和时钟为1

case current\_state is --那么当前的申明为：

when qa=>paidtemp:="00000000";neededtemp:="00000000";

--- =>相当于then

backmoney:="00000000";pricetemp:="00000000";q<=0;

showmoneyout<='0';moneyout<="00000000";paid<="00000000";

needed<="00000000";failure<='0';success<='0';

--都为0-------

if cola='1' or pepsi='1' then current\_state<=qb;

-----选择其中一种饮料，将qb赋值给前面---

if cola='1' then pricetemp:=pricetemp+"00001111";

---如果选择cola,则物品的价格为---

neededtemp:=pricetemp; --还需要金额为物品价格

Else

pricetemp:=pricetemp+"00010100"; --否则物品价格为--

neededtemp:=pricetemp;

end if;

end if;

paid<=paidtemp; --投入的钱为现在投入的钱

needed<=neededtemp;

when qb=>if coin1='1' or coin5='1' then

---当qb信号为投入1块或者投入5毛时--

if coin1='1'then paidtemp:=paidtemp+"00001010";

---如果选择coin,则投入的金额为---

else

paidtemp:=paidtemp+"00000101";

---如果选择pepsi---

end if;

if paidtemp>=pricetemp then backmoney:=paidtemp-pricetemp;

--如果投入的今额大于等于物品价格，则退出的金额为投入减去物品价格---

neededtemp:="00000000";current\_state<=qd;

---还需要投入的今额为0，当前形式为qd---

else neededtemp:=pricetemp-paidtemp;backmoney:="00000000";

-----否则需要的金额为价格减去投入今额，退出金额为0---

current\_state<=qc;q<=0;

end if;

paid<=paidtemp;

needed<=neededtemp;

end if;

if q<8 then q<=q+1;

if cola='1' or pepsi='1'then q<=0;

---如果cola为1或者pepsi为1 ，那么q=0---

if cola='1' then pricetemp:=pricetemp+"00001111";

neededtemp:=neededtemp+"00001111";

---如果cola为1，那么物品价格为，需要金额为。---

else

pricetemp:=pricetemp+"00010100";

neededtemp:=neededtemp+"00010100";

---否则物品价格为，需要金额为。--

end if;

paid<=paidtemp;

needed<=neededtemp;

end if;

else current\_state<=qe;q<=0; --如果当前形式为qe，q<0--

end if;

when qe=>failure<='1';

---当为qe时，出货失败---

if q<4 then q<=q+1;

else current\_state<=qa;q<=0;

end if;

when qc=>if coin1='1' or coin5='1' then

if coin1='1'then paidtemp:=paidtemp+"00001010";

else

paidtemp:=paidtemp+"00000101";

end if;

if paidtemp>=pricetemp then

backmoney:=paidtemp-pricetemp;

neededtemp:="00000000";current\_state<=qd;

else neededtemp:=pricetemp-paidtemp;backmoney:="00000000";

current\_state<=qc;

end if;

paid<=paidtemp;

needed<=neededtemp;

end if;

if coin1/='1'and coin5/='1' then

---如果投入1块和投入5毛不等于1---

if q<10 then q<=q+1;

else current\_state<=qg;

end if;

else q<=0;

end if;

when qg=>failure<='1';

showmoneyout<='1';moneyout<=paidtemp;

---正在找零。找零为投入金额---

current\_state<=qf;q<=0;

success<='0';

when qd=>success<='1';

if backmoney>"00000000"then showmoneyout<='1';

moneyout<=backmoney;

----如果退出金额不为0，那么正在找零，找零为退出的金额---

end if;

current\_state<=qf;q<=0;

when qf=>if q<4 then q<=q+1;

else current\_state<=qa;q<=0;

end if;

end case;

else

end if;

end process;

end behav;

BCD译码模块完整程序如下：

Library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity code1 is

port(

b:in std\_logic\_vector(7 downto 0);

bcd0:out std\_logic\_vector(3 downto 0);

bcd1:out std\_logic\_vector(3 downto 0)

) ;

end code1;

architecture one of code1 is

begin

process(b)

begin

case b is

when"00000000"=>bcd0<="0000";bcd1<="0000";--译码“0”；

when"00000001"=>bcd0<="0001";bcd1<="0000";--译码“1”；

when"00000010"=>bcd0<="0010";bcd1<="0000";--译码“2”；

when"00000011"=>bcd0<="0011";bcd1<="0000";--译码“3”；

when"00000100"=>bcd0<="0100";bcd1<="0000";--译码“4”；

when"00000101"=>bcd0<="0101";bcd1<="0000";--译码“5”；

when"00000110"=>bcd0<="0110";bcd1<="0000";

when"00000111"=>bcd0<="0111";bcd1<="0000";

when"00001000"=>bcd0<="1000";bcd1<="0000";

when"00001001"=>bcd0<="1001";bcd1<="0000";

when"00001010"=>bcd0<="0000";bcd1<="0001";

when"00001011"=>bcd0<="0001";bcd1<="0001";

when"00001100"=>bcd0<="0010";bcd1<="0001";

when"00001101"=>bcd0<="0011";bcd1<="0001";

when"00001110"=>bcd0<="0100";bcd1<="0001";

when"00001111"=>bcd0<="0101";bcd1<="0001";

when"00010000"=>bcd0<="0110";bcd1<="0001";

when"00010001"=>bcd0<="0111";bcd1<="0001";

when"00010010"=>bcd0<="1000";bcd1<="0001";

when"00010011"=>bcd0<="1001";bcd1<="0001";

when"00010100"=>bcd0<="0000";bcd1<="0010";译码“20”

when others=>null;

end case;

end process;

end one;

顶层模块完整程序如下：

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity top is

Port( clk1:in std\_logic; ---标准逻辑位类型的输入端口---

C1,C5,P1\_5,P2:in std\_logic;

---C1,C5投入1块5毛的金额，P1\_5为cola ,P2为pepsi---

paid\_lcd0,paid\_lcd1,needed\_lcd0,needed\_lcd1,Mout\_lcd0,Mout\_lcd1:out

---已付金额、需要金额、退币金额----

std\_logic\_vector(3 downto 0);

s,f,showout :out std\_logic ); ---showout 退币指示灯---

end top;

architecture one of top is ---结构体

component shouhuoji ---配置

port ( clk:in std\_logic;

coin1:in std\_logic;

coin5:in std\_logic;

cola:in std\_logic;

pepsi:in std\_logic;

paid:out std\_logic\_vector(7 downto 0);

needed:out std\_logic\_vector(7 downto 0);

success:out std\_logic;

failure:out std\_logic;

showmoneyout:out std\_logic; ---正在找零

moneyout:out std\_logic\_vector(7 downto 0) ---找零

);

end component; ---结束配置

component code1 ---配置

port(

b:in std\_logic\_vector( 7 downto 0);

bcd0:out std\_logic\_vector(3 downto 0); --- bcd 二进制编码

bcd1:out std\_logic\_vector(3 downto 0)

);

end component;

signal p,n,mo:std\_logic\_vector( 7 downto 0);

--signal s1,s2,s3,s4,s5,s6:std\_logic\_vector(3 downto 0);

begin

u0:shouhuoji--各模块连接

port map(clk=>clk1,coin1=>C1,coin5=>C5,cola=>P1\_5,pepsi=>P2,

---端口映射，

success=>s,failure=>f,showmoneyout=>showout,paid=>p,needed=>n,moneyout=>mo);

u1:code1 port map(b=>p,bcd0=>paid\_lcd0,bcd1=>paid\_lcd1);

u2:code1 port map(b=>n,bcd0=>needed\_lcd0,bcd1=>needed\_lcd1);

u3:code1 port map(b=>mo,bcd0=>Mout\_lcd0,bcd1=>Mout\_lcd1);

end one;