西安电子科技大学

数字电路实验 课程实验报告

实验名称 大规模集成数字电路设计

机电工程 学院 2004031 班

姓名 梁志恒 学号 20049200420

姓名 王煊 学号 20049200399

姓名 周宏宇 学号 20049200400

姓名 杨翰 学号 20049200030

姓名 张睿恒 学号 20049200095

姓名 卡米让·卡米力江 学号 20040300001

成 绩

同作者

实验日期 2021 年 11 月 21 日

指导教师评语:

指导教师：董瑞军

_____年____月__日

一、实验目的

1、熟悉大规模集成数字电路的设计方法

2、熟悉数字电路的调试

二、实验内容

自动饮料售货机

1、项目功能

（1） 识别 0.5 1.0 元两种硬币

（2） 售出 3 种不同价格的饮料，饮料价格分别为 1.5 2.0 2.5；

（3） 可以找零

（4）购买者可以选择饮料购买

(5) 卖出饮料后自动复位

（6）手动复位

2、设计要求分析

首先项目分为输入，判断与输出三个模块。输入分为商品内容选择，货币投入。判断模块为判断此时投入货币是否满足购买要求，以及是否找零。输出模块负责输出购买信号及找零信号。饮料共有三种，故设计四种状态分别为 choose00 choose01 choose10 choose11 分别代表初始，买 1.5 元饮料，买 2.0 元饮料，买 2.5 元饮料.系统内 7 种状态转化，分别为为 S0-6，分别为 s0 表示初态，s1 表示投入 5 角，s2 表示投入 1 元，s3 表示投入 1 元 5 角，s4 表示投入 2 元，s5 表示投入 2 元 5 角，s6 表示投入 3 元。例如选择 01 即 1.5 元的饮料。此时状态为 S0.输入端口输入 01 即投入五角持续三个 CP，第一个 CP 是 S0——S1 代表投入五角。第二个 CP S1——S2 代表此时已经投入一元，再经过一个 Cp S2——S3 表示投入 1.5 元此时输出为 10 即投出货物不找零，下一个 CP 时输入变为 00 状态回归 S0 并再 S0 保持。其他情况依次类推。当再初始状态 choose00 时投入硬币，则 warning 会置 1，现实意义为投币处放置挡板，硬币会被返回给用户。

3、代码实现

```
library ieee;
use ieee.std_logic_1164.all;
entity zidong is
port(clk,reset      :in std_logic;
            choose :in std_logic_vector(1 downto 0);
```

```
            state_inputs:in std_logic_vector(0 to 1);
            comb_outputs:out std_logic_vector(0 to 1);
            warning    :out std_logic);
end zidong;
architecture be of zidong is
   type fsm_st is(s0,s1,s2,s3,s4,s5,s6);
   signal current_state,next_state:fsm_st;
begin
reg:process(reset,clk)
      begin
      if reset='1'then current_state<=s0;
      elsif rising_edge(clk)then
               current_state<=next_state;
      end if;
   end process;
com:process(current_state,state_inputs,choose)
begin
   warning <= '0';
 if(choose="11") then
      warning <= '0';
     case current_state is
      when s0=>comb_outputs<="00";
        if
              state_inputs="00" then next_state<=s0;
       elsif state_inputs="01" then next_state<=s1;
       elsif state_inputs="10" then next_state<=s2;
       end if;
      when s1=>comb_outputs<="00";
        if
              state_inputs="00" then next_state<=s1;
       elsif state_inputs="01" then next_state<=s2;
       elsif state_inputs="10" then next_state<=s3;
          end if;
      when s2=>comb_outputs<="00";
        if
              state_inputs="00" then next_state<=s2;
       elsif state_inputs="01" then next_state<=s3;
       elsif state_inputs="10" then next_state<=s4;
          end if;
      when s3=>comb_outputs<="00";
        if
              state_inputs="00" then next_state<=s3;
       elsif state_inputs="01" then next_state<=s4;
       elsif state_inputs="10" then next_state<=s5;
```

```
        end if;
  when s4=>comb_outputs<="00";
     if
            state_inputs="00" then next_state<=s4;
   elsif state_inputs="01" then next_state<=s5;
   elsif state_inputs="10" then next_state<=s6;
         end if;
  when s5=>comb_outputs<="10";
     if
            state_inputs="00" then next_state<=s0;
   elsif state_inputs="01" then next_state<=s1;
   elsif state_inputs="10" then next_state<=s2;
      end if;
  when s6=>comb_outputs<="11";
     if
            state_inputs="00" then next_state<=s0;
   elsif state_inputs="01" then next_state<=s1;
   elsif state_inputs="10" then next_state<=s2;
   end if;
   end case;

elsif(choose="10") then
   warning<='0';
case current_state is
  when s0=>comb_outputs<="00";
     if
            state_inputs="00" then next_state<=s0;
   elsif state_inputs="01" then next_state<=s1;
   elsif state_inputs="10" then next_state<=s2;
   end if;
  when s1=>comb_outputs<="00";
     if
            state_inputs="00" then next_state<=s1;
   elsif state_inputs="01" then next_state<=s2;
   elsif state_inputs="10" then next_state<=s3;
      end if;
  when s2=>comb_outputs<="00";
     if
            state_inputs="00" then next_state<=s2;
   elsif state_inputs="01" then next_state<=s3;
   elsif state_inputs="10" then next_state<=s4;
      end if;
  when s3=>comb_outputs<="00";
     if
```

```
            state_inputs="00" then next_state<=s3;
    elsif state_inputs="01" then next_state<=s4;
    elsif state_inputs="10" then next_state<=s5;
        end if;
  when s4=>comb_outputs<="10";
    if
            state_inputs="00" then next_state<=s0;
    elsif state_inputs="01" then next_state<=s1;
    elsif state_inputs="10" then next_state<=s2;
        end if;
  when s5=>comb_outputs<="11";
    if
            state_inputs="00" then next_state<=s0;
    elsif state_inputs="01" then next_state<=s1;
    elsif state_inputs="10" then next_state<=s2;
      end if;
  when s6=>comb_outputs<="11";
    if
            state_inputs="00" then next_state<=s1;
    elsif state_inputs="01" then next_state<=s2;
    elsif state_inputs="10" then next_state<=s3;
      end if;
    end case;

elsif(choose="01") then
    warning<='0';
  case current_state is
  when s0=>comb_outputs<="00";
    if
            state_inputs="00" then next_state<=s0;
    elsif state_inputs="01" then next_state<=s1;
    elsif state_inputs="10" then next_state<=s2;
    end if;
  when s1=>comb_outputs<="00";
    if
            state_inputs="00" then next_state<=s1;
    elsif state_inputs="01" then next_state<=s2;
    elsif state_inputs="10" then next_state<=s3;
      end if;
  when s2=>comb_outputs<="00";
    if
            state_inputs="00" then next_state<=s2;
    elsif state_inputs="01" then next_state<=s3;
    elsif state_inputs="10" then next_state<=s4;
```

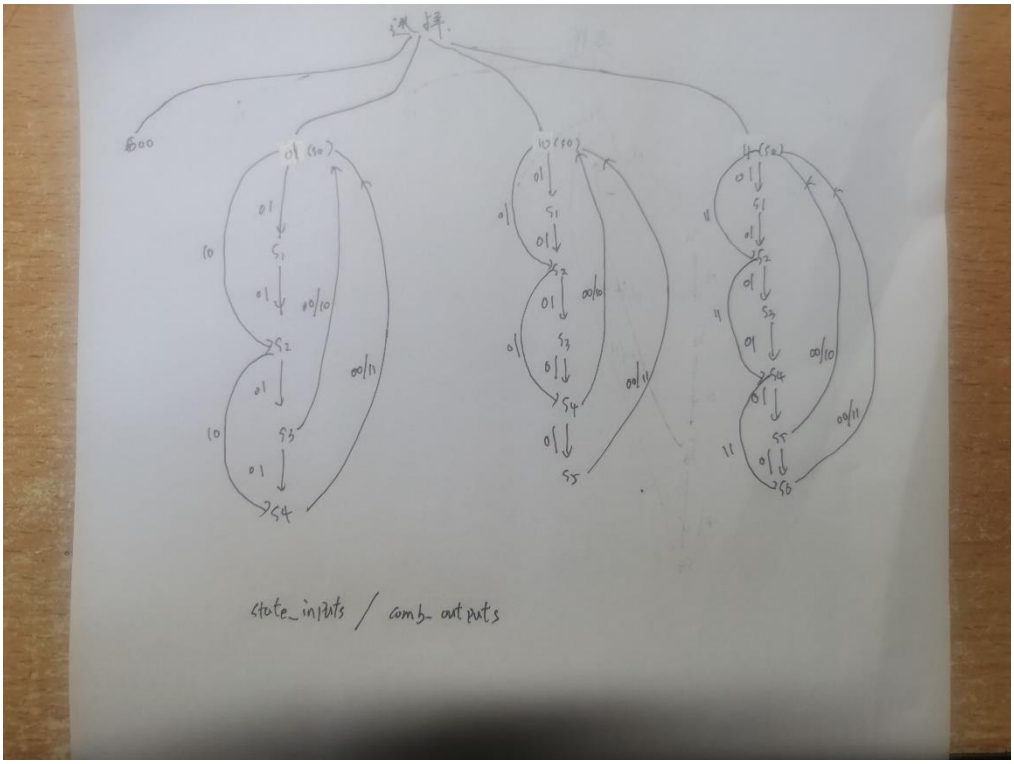
```
      end if;
    when s3=>comb_outputs<="10";
      if
            state_inputs="00" then next_state<=s0;
     elsif state_inputs="01" then next_state<=s1;
     elsif state_inputs="10" then next_state<=s2;
        end if;
    when s4=>comb_outputs<="11";
      if
            state_inputs="00" then next_state<=s0;
     elsif state_inputs="01" then next_state<=s1;
     elsif state_inputs="10" then next_state<=s2;
         end if;
    when s5=>comb_outputs<="11";
      if
            state_inputs="00" then next_state<=s1;
     elsif state_inputs="01" then next_state<=s2;
     elsif state_inputs="10" then next_state<=s3;
      end if;
    when s6=>comb_outputs<="11";
      if
            state_inputs="00" then next_state<=s2;
     elsif state_inputs="01" then next_state<=s3;
     elsif state_inputs="10" then next_state<=s4;
      end if;
     end case;
    else
      if
            state_inputs="00" then warning<='0';
                 next_state<=s0;
     elsif state_inputs="01" then warning<='1';
                 next_state<=s0;
     elsif state_inputs="10" then warning<='1';
                 next_state<=s0;
     end if;
    end if;
     end process;
     end be;
```

4、flow summary

Flow Summary

Flow Status	Successful - Sun Nov 21 09:01:25 2021
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	autosell
Top-level Entity Name	autosell
Family	MAX7000S
Device	EPM7128SLC84-15
Timing Models	Final
Met timing requirements	Yes
Total macrocells	25 / 128 (20 %)
Total pins	13 / 68 (19 %)

五、状态转移

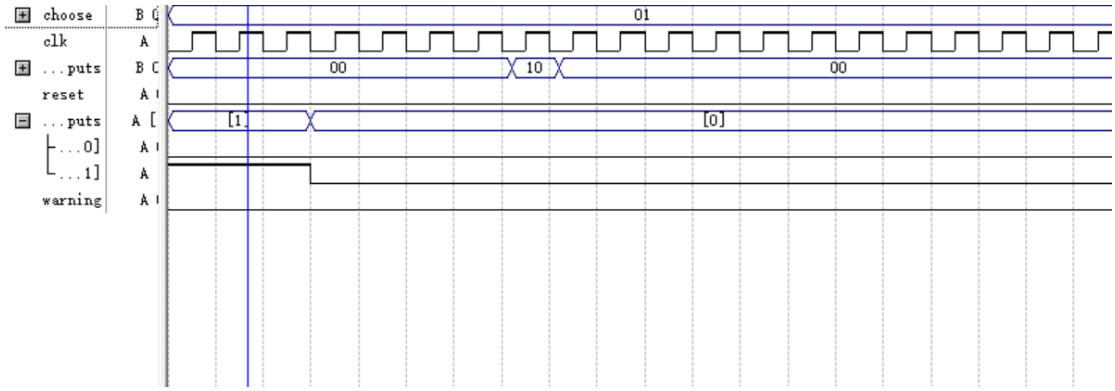


六、波形仿真

图中 choose 表示选择的饮料 clk 为时钟信号 上 PUT 为输出结果，下 PUT 为投入钱数

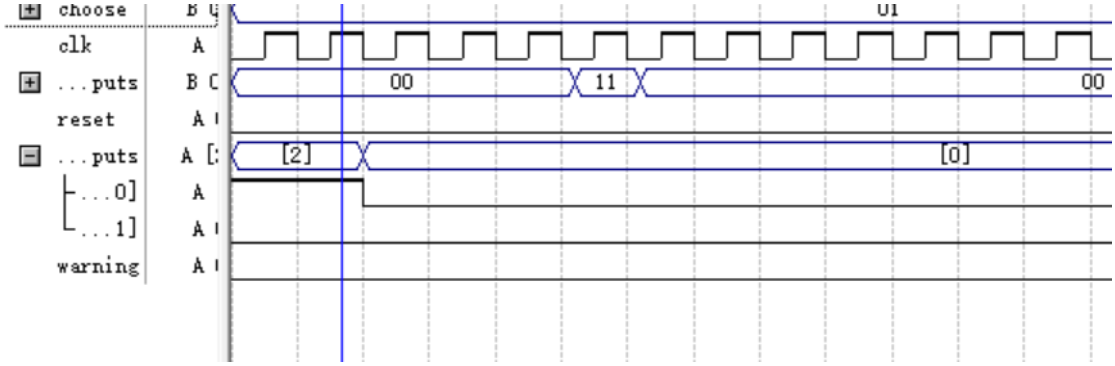
基本功能测试

（1）、



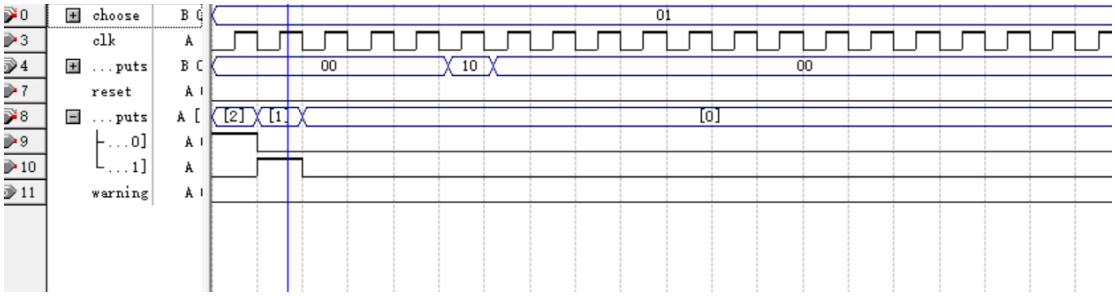
输入三个 01 即投入 1.5 元。选择 01 即 1.5 元饮料。输出 10，表示放出一瓶饮料并找回 0 元

（2）



输入两个 10 即投入 2 元。选择 01 即 1.5 元饮料。输出 10，表示放出一瓶饮料并找回 0.5 元

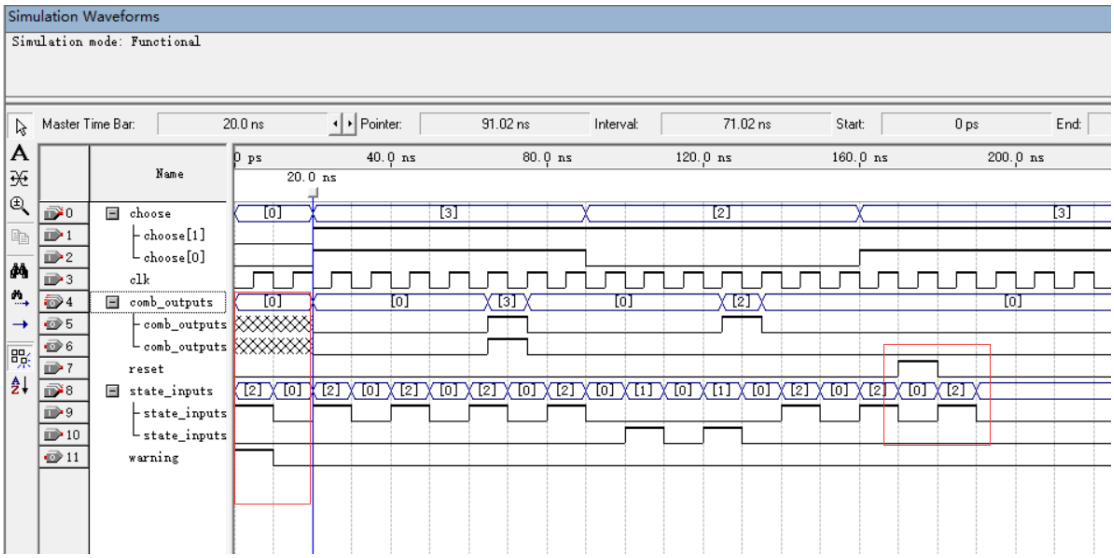
（3）



输入一个 10 一个 01 即投入 1.5 元。选择 01 即 1.5 元饮料。输出 10，表示放出一瓶饮料并找回 0 元

功能联合测试

（4）



当初始状态（出场状态）投币时，warning 置 1，表示投币处挡板下置，硬币会被返回给用户。切换至三档（2.5 元），投入 3 个 1 元硬币，后 comb_outputs 为 11，表示出一瓶饮料，且找回 5 角钱。再投 1 枚硬币后，切换至二档（2.0 元），再投入两枚 5 角硬币，comb_outputs 置 10，表示只出一瓶饮料。第二个方框处，reset 置 1，使售卖机内状态清 0。

七、成员分工及自评

姓名	学号	任务	评分
梁志恒	20049200420	程序编写及调试	100
王煊	20049200399	程序调试及汇报	79
周宏宇	20049200400	仿真波形设计及汇报	79
张睿恒	20049200095	文档编写	79
杨翰	20049200030	程序纠错及修改	79
卡米让	20040300001	小组讨论提供选题及讨论	64