

西安电子科技大学

电子线路实验

课程实验报告

实验名称 乒乓球比赛的模拟设计

机电工程 学院 2004031 班

姓名 贾昊燊 学号 20049200517

同作者 周欣彤 纪晓慧 李思云 杜钰珑 叶双田

实验日期 2021 年 11 月 21 日

成 绩

指导教师评语：

指导教师：

_____年____月____日

实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验数据记录（或仿真及软件设计）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

一、实验目的

1. 对乒乓球比赛进行电子线路上的模拟。
2. 掌握分模块的 VHDL 语言的编写及调试技能，理解实体设计与构造体等概念。
3. 进一步加深对数字电路应用技术方面的了解与认识，熟悉数字电路系统设计、制作与调试的方法和步骤。
4. 针对电子线路课程要求，进行 VHDL 语言、调试等各环节的综合性训练，培养运用课程中所学的理论与实践紧密结合，独立地解决实际问题的能力。

二、实验所用仪器（或实验环境）

- 1、数字逻辑电路实验箱
- 2、CPLD 开发板

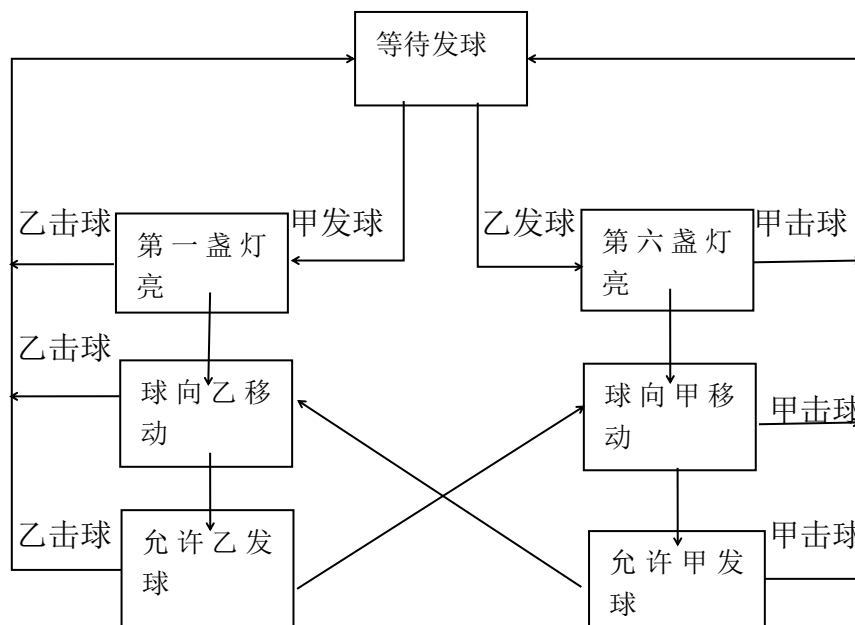
三、设计内容

1. 设计介绍及创新点说明

模拟乒乓球单打比赛系统的设计的思路是用 6 个 LED 排成一条直线，以中点为界，两边各代表参赛双方的位置，其中一只点亮的 LED 表示球的当前位置，点亮的 LED 依次从左到右，或从右到左进行移动代表乒乓球运动的轨迹。当“球”（点亮的那只 LED）运动到某方的最后一位时，此方应按下位于自己一方的按钮开关，即表示启动球拍击球，假如击中，球向相反方向运动，直至任意一方没有击中球，此时则对手记一分。设置自动记分电路，甲乙双方各用一位数码管进行记分显示，每计满 11 分为 1 局。1 局结束后计分清零，自动进入下一局。

本设计的创新点在于不仅对比赛的过程做了直观的模拟，比赛双方的比分与输赢结果也可以用数码管进行直观的显示，同时我们也设计了决定发球方向的输入端，使模拟更加贴近现实。在考虑电路模拟的不确定性上，我们也设计了重置端口，使系统在发生故障时可以重新启动。另外，在第一次设计完成后我们发现没有考虑当达到 10:10 时的情况，因此我们在局分记录模块对此进行补充改善，使其可以更符合实际规则，如以 12:10 结束本局。

2. 工作原理



3. 系统设计

模拟系统包含两个模块：比赛过程模拟模块、比赛结果记录模块，通过顶层文件将其组成完整的模拟系统。

模拟系统输入端包括：时钟信号输入端 `clk_1` (上升沿有效)、发球输入端 `serve` (01 表示甲发球，10 表示乙发球)、两方的击球输入端 `hit1` 与 `hit2` (1 表示击球)、比赛开始输入端 `startbutton` (1 表示比赛开始)。输出端包括：控制六个 LED 的输出端口 (高电平点亮 LED)、比分输出端 `counte` 与 `countf`、局分输出端 `countc` 与 `countd`、胜者输出端 `winer`。

四、实验内容及步骤

1. 源代码

(1) pingpangqiu 子模块

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity pingpangqiu is
port(reset:in std_logic;

```

```

    clk_1:in std_logic;
    startbutton:in std_logic;
    serve:in std_logic_vector(1 downto 0);
    hit1,hit2:in std_logic;
    light:out std_logic_vector(1 to 6);
    counta,countb,counte,countf:out std_logic_vector(3 downto 0));
end pingpangqiu;

architecture zhuti of pingpangqiu is
    type states is
        (waitserve,lighton,light6on,ballmoveto2,ballmoveto1,allow1hit,allow2hit);
    signal i:integer range 0 to 6;
    signal state:states;
    signal count1,count2,count3,count4:std_logic_vector(3 downto 0);
    signal c:std_logic;
begin
    p1:process(clk_1)
    begin
        if reset='1' then
            i<=0;
            count1<="0000";
            count2<="0000";
            count3<="0000";
            count4<="0000";
        elsif clk_1'event and clk_1='1' then
            if ((count1>"1010")and(count2<(count1-1))) then
                i<=0;
                count1<="0000";
                count2<="0000";
            end if;
        end if;
    end process;
end architecture;

```

```

    count3<=count3+1;
elseif ((count2>"1010")and(count1<(count2-1))) then
    i<=0;
    count1<="0000";
    count2<="0000";
    count4<=count4+1;
elseif startbutton='0' then
    i<=0;
    count1<="0000";
    count2<="0000";
    count3<="0000";
    count4<="0000";
else
    case state is
        when waitserve=>
            case serve is
                when "01"=> i<=1; state<=light1on; c<='0';
                when "10"=> i<=6; state<=light6on; c<='0';
                when "11"=> i<=0; c<='0';
                when others=> i<=0; c<='0';
            end case;
        when light1on=>
            i<=2;
            if hit2='1' then
                i<=0;
                count1<=count1+1;
                c<='1';
                state<=waitserve;
            else

```

```

        c<=' 0' ;
        state<=ballmoveto2;
    end if;
when light6on=>
    i<=5;
    if hit1=' 1' then
        i<=0;
        count2<=count2+1;
        c<=' 1' ;
        state<=waitserve;
    else
        c<=' 0' ;
        state<=ballmoveto1;
    end if;
when ballmoveto1=>
    if hit1=' 1' then i<=0;
        count2<=count2+1;
        c<=' 1' ;
        state<=waitserve;
    elsif i=2 then i<=1;c<=' 0' ;
        state<=allow1hit;
    else i<=i-1;
    end if;
when ballmoveto2=>
    if hit2=' 1' then i<=0;
        count1<=count1+1;
        c<=' 1' ;
        state<=waitserve;
    elsif i=5 then i<=6;c<=' 0' ;

```

```

        state<=allow2hit;
    else i<=i+1;
    end if;
when allow1hit=>
    if hit1='1' then i<=2;c<='0';
        state<=ballmoveto2;
    else count2<=count2+1;
        i<=0;
        c<='1';
        state<=waitserve;
    end if;
when allow2hit=>
    if hit2='1' then i<=5;
        c<='0';
        state<=ballmovetol;
    else count1<=count1+1;
        i<=0;
        c<='1';
        state<=waitserve;
    end if;
end case;
end if;
end if;
end process p1;
counta<=count1;
countb<=count2;
counte<=count1;
countf<=count2;

```

```

p2:process(i)
begin
case i is

when(6)=>light<= "100000";
when(5)=>light<= "010000";
when(4)=>light<= "001000";
when(3)=>light<="000100" ;
when(2)=>light<="000010";
when(1)=>light<="000001" ;
when others=>light<="000000";
end case;
end process p2;
end architecture zhuti;

```

(2) sum 子模块

```

library ieee;

use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity sum is
    port( reset:in std_logic;
          clk_1:in std_logic;
          counta:in std_logic_vector(3 downto 0);
          countb:in std_logic_vector(3 downto 0);
          countc,countd,winer,counte,countf:out std_logic_vector(3
downto 0));
    end sum;
architecture two of sum is

    signal count3,count4:std_logic_vector(3 downto 0):="0000";

```



```

    signal win:std_logic_vector(3 downto 0):="0000";
begin
    process(clk_1)
begin
    if clk_1'event and clk_1='1' then
        if reset='1' then count3<="0000";count4<="0000";
        elsif ((counta>"1010")and(countb<(counta-1))) then
            count3<=count3+1;
            elsif ((countb>"1010")and(counta<(countb-1))) then
                count4<=count4+1;
            end if;
        if ((counta>"1001")and (countb>"1001")) then
            if counta=(countb+2) then count3<=count3+1;
            elsif countb=(counta+2) then count4<=count4+1;
            end if;
        end if;
    end if;
countc<=count3;countd<=count4;
end process;
    process(clk_1)
begin
    if clk_1'event and clk_1='1' then
        if count3="0011" then win<="1010";
        elsif count4="0011" then win<="1011";
        else win<="0000";
        end if;
    end if;
    winer<=win;
end process;

```

```
end two;
```

(3) 顶层文件 c

```
library ieee;
```

```
    use ieee.std_logic_1164.all;
```

```
    use ieee.std_logic_arith.all;
```

```
entity c is
```

```
port (
```

```
    reset:in std_logic;
```

```
    clk_1:in std_logic;
```

```
    startbutton:in std_logic;
```

```
    serve:in std_logic_vector(1 downto 0);
```

```
    hit1,hit2:in std_logic;
```

```
    light:out std_logic_vector(1 to 6);
```

```
    counte,countf:out std_logic_vector(3 downto 0);
```

```
    countc,countd,winer:out std_logic_vector(3 downto 0));
```

```
end entity c;
```

```
architecture behav of c is
```

```
    signal temp1: std_logic_vector(3 downto 0);
```

```
    signal temp2: std_logic_vector(3 downto 0);
```

```
    component pingpangqiu
```

```
    port(reset:in std_logic;
```

```
        clk_1:in std_logic;
```

```
        startbutton:in std_logic;
```

```
        serve:in std_logic_vector(1 downto 0);
```

```
        hit1,hit2:in std_logic;
```

```
        light:out std_logic_vector(1 to 6);
```

```
        counte,countf:out std_logic_vector(3 downto 0);
```

```
        counta,countb:out std_logic_vector(3 downto 0));
```

```

        end component;
component sum
port( reset:in std_logic;
      clk_1:in std_logic;
      counta:in std_logic_vector(3 downto 0);
      countb:in std_logic_vector(3 downto 0);
      countc,countd,winer:out std_logic_vector(3 downto 0));
end component;

begin

    U0:pingpangqiu port
map( counta=>temp1, countb=>temp2, reset=>reset, clk_1=>clk_1, startbutton=
>startbutton,

serve=>serve, hit1=>hit1, hit2=>hit2, light=>light, counte=>counte, countf=
>countf);

    U1:sum port
map(counta=>temp1, countb=>temp2, reset=>reset, clk_1=>clk_1, winer=>winer,
countc=>countc, countd=>countd);

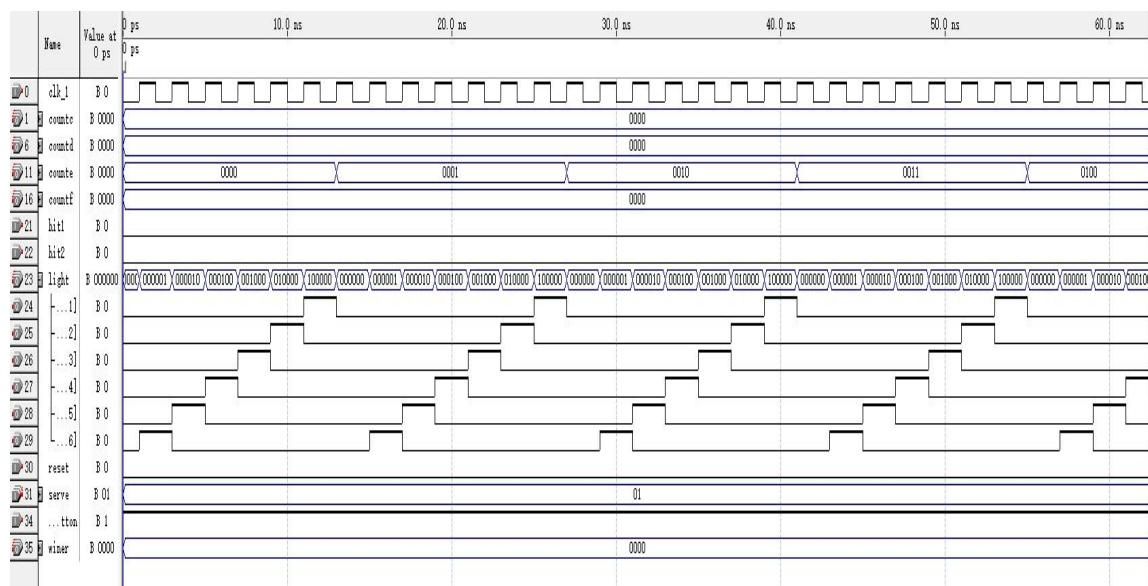
    END;

```

2. flow summary 截图

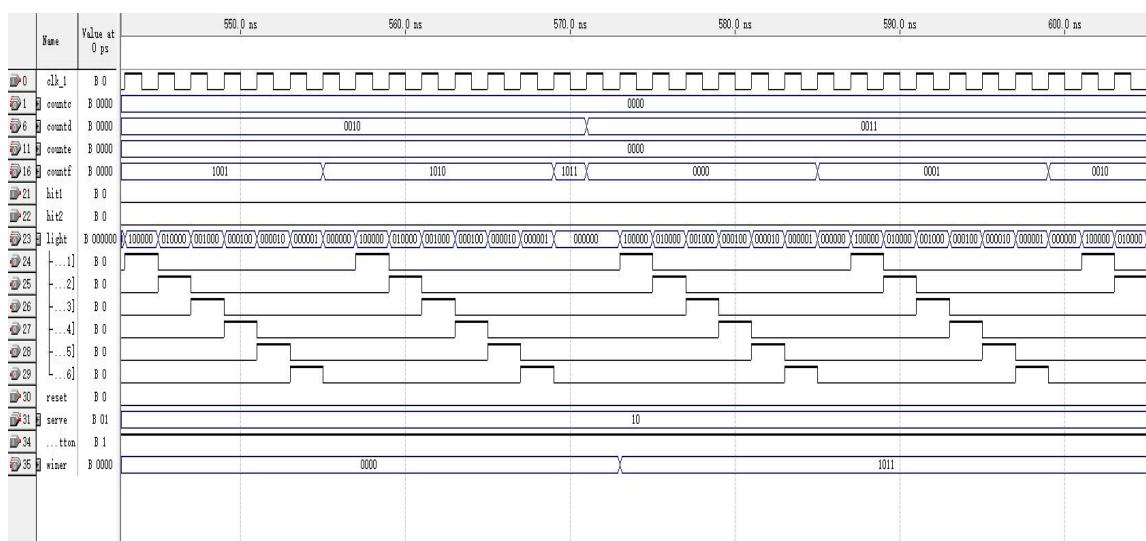
Flow Status	Successful - Sat Nov 20 21:55:09 2021
Quartus II Version	9.0 Build 132 02/25/2009 SJ Full Version
Revision Name	c
Top-level Entity Name	c
Family	MAX7000S
Device	EPM7128SLC84-15
Timing Models	Final
Met timing requirements	Yes
Total macrocells	73 / 128 (57 %)
Total pins	37 / 68 (54 %)

3. 波形仿真及解释说明

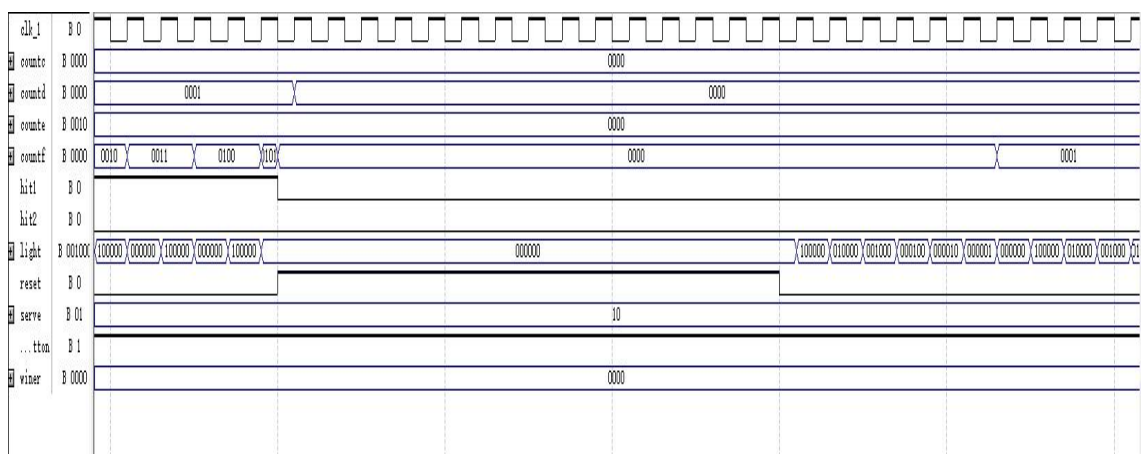


波形仿真显示端口包括 clk_1、countc、countd、counte、countf、hit1、hit2、light、reset、serve、startbutton、winner，证实系统结构完整。

从图中看到 startbutton 端为 1，比赛正常进行，serve 端为“01”，表示甲发球，因此 light 从 6 开始点亮依次至 1，证实发球选择功能与乒乓球轨迹显示功能正常工作；进而，当乒乓球轨迹达到 1 端时 hit2 为 0 表示乙未击球，因此甲得分，counte 由 0000 变为 0001，证实计分功能局部正常工作，此后乙均未在正确位置击球，比分按规则顺序变化，证实计分功能整体正常工作。



由图中可以看出当双方比分为 11:0 时，此时局分由 2:0 变为 3:0，证实局分记录功能正常工作；winner 端由 0000 变为 1011 表示乙方胜利，证实比赛结果显示功能正常工作。

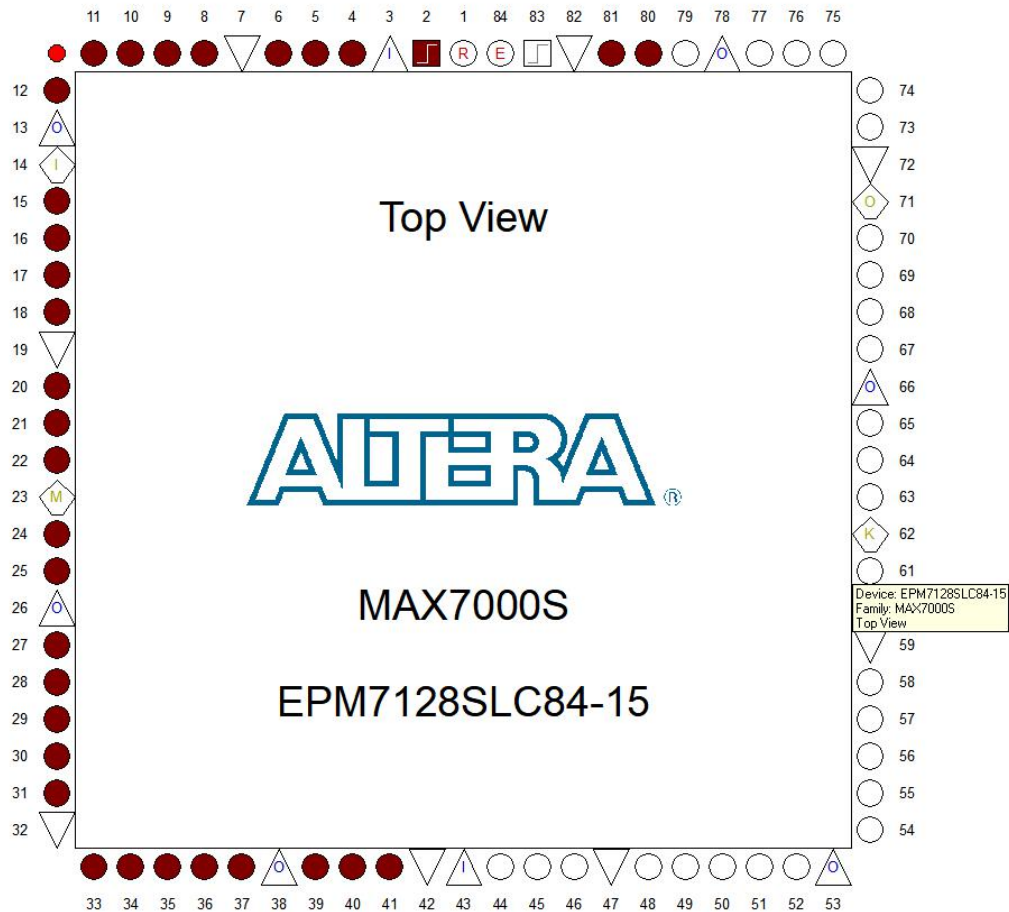


从图中看到当 reset 为 1 时，比赛重置，当 reset 由 1 变 0 时，比赛重新开始，证实比赛重置功能正常工作。

从图中看到当球到达 1 位置时，hit2 为 1 表示乙击球，此时，乙在正确位置击球，乒乓球由乙端向甲端移动，证实击打功能正常工作，也证实系统模拟乒乓球轨迹功能正确。

综上，模拟乒乓球比赛的系统正常工作，系统设计正确。

4. 锁定引脚



锁定引脚后 flow summary 截图：

```
Flow Status           Successful - Sat Nov 20 23:08:52 2021
Quartus II Version    9.0 Build 132 02/25/2009 SJ Full Version
Revision Name         c
Top-level Entity Name c
Family                MAX7000S
Device                EPM7128SLC84-15
Timing Models         Final
Met timing requirements Yes
Total macrocells      87 / 128 ( 68 % )
Total pins            37 / 68 ( 54 % )
```

五、实验问题及心得

通过本次设计实验，小组成员掌握熟悉了 VHDL 语言设计方法。本次小组分工分为三部分：1. 查找设计流程图，提供设计仿真思路 2. 分模块设计，编译仿真 3. 设计顶层文件，分析出现的问题。为提高团队有效性，我们集中时间共同完成设计，及时完成对问题的反馈。

元器件的挑选方面：优先选择项目要求的元件类型在选择同一种元件时，考虑元件的价格性能比，还需考虑选择节能性，功耗小的元件。

电路的设计思路方面：在本次设计过程中出现了宏单元超出限制个数的问题，造成程序编译不成功。这启示我们设计实验内容时要提前了解限制实验设计的因素，进而再实现想要实现的功能。设计电路时先确定电路需要几部分，再分步设计每一部分电路，最终将各部分电路按要求连接在一起，实现基本功能后再做调整与修改，最终设计出一个较为完美的电路。

实验中，各组员要明确自己的任务，这样才能提高工作效率；遇到问题时需要冷静思考，不能自乱阵脚，理智分析问题并解决问题；实验中也要注意积累经验，不断提升自己。

六、自评成绩

20049200517 贾昊燊 100

20049200505 周欣彤 74

20049200359 叶双田 74

20049200104 李思云 74

20049200427 纪晓慧 74

20049200424 杜钰珑 74