

verilog 乒乓球大作业（仿真成功）

本页仅作为文档页封面，使用时可以删除
This document is for reference only-rar21year.March

一 任务描述

设计要求

左、右双方各持一按键作为“球拍”，试验板上一行 8 只发光二极管为乒乓球台，其中那只发光的发光二极管代表乒乓球所在位置。



设计如图 1 所示的乒乓球玩耍实物图的模拟掌握器。发球方最终一位 LED 点亮，按下键表示发球，亮的灯依次向对方移动，当到达对方最终一位时 0.5s 内对方必需按下按键表示接球，接球后 LED 灯向对方移动，否则输球。

接球时，LED 没有亮到最终一位时就按下接球按键为犯规。输球或者犯规，对方加 1 分，领先加到 11 分者玩耍胜出这一局。

设计思路

考虑到玩耍的简单度，承受状态机来实现这些模式转换。用到 8 个状态，如下所示：

状态	功能
S0	复位状态，也是推断发球权的状态
S1	甲方发球
S2	球从甲方向乙方右移,假设此时乙方接球
S3	开头推断乙方是否在 0.5S 内接球
S4	乙方发球
S5	球从乙方向甲方左移,假设此时甲方接球就给乙方加一分，并将球权给乙方
S6	开头推断甲方是否在 0.5S 内接球

程序还参加了蜂鸣、用数码管显示分数和发球权等功能。

一 任务描述

设计要求

左、右双方各持一按键作为“球拍”，试验板上 8 只发光二极管为乒乓球台，其中那只发光的发光二极管代表乒乓球所在位置。



设计如图 1 所示的乒乓球玩耍实物图的模拟掌握器。发球方最终一位 LED 点亮，按下键表示发球，亮的灯依次向对方移动，当到达对方最终一位时 0.5s 内对方必需按下按键表示接球，接球后 LED 灯向对方移动，否则输球。

接球时，LED 没有亮到最终一位时就按下接球按键为犯规。输球或者犯规，对方加 1 分，领先加到 11 分者玩耍胜出这一局。

设计思路

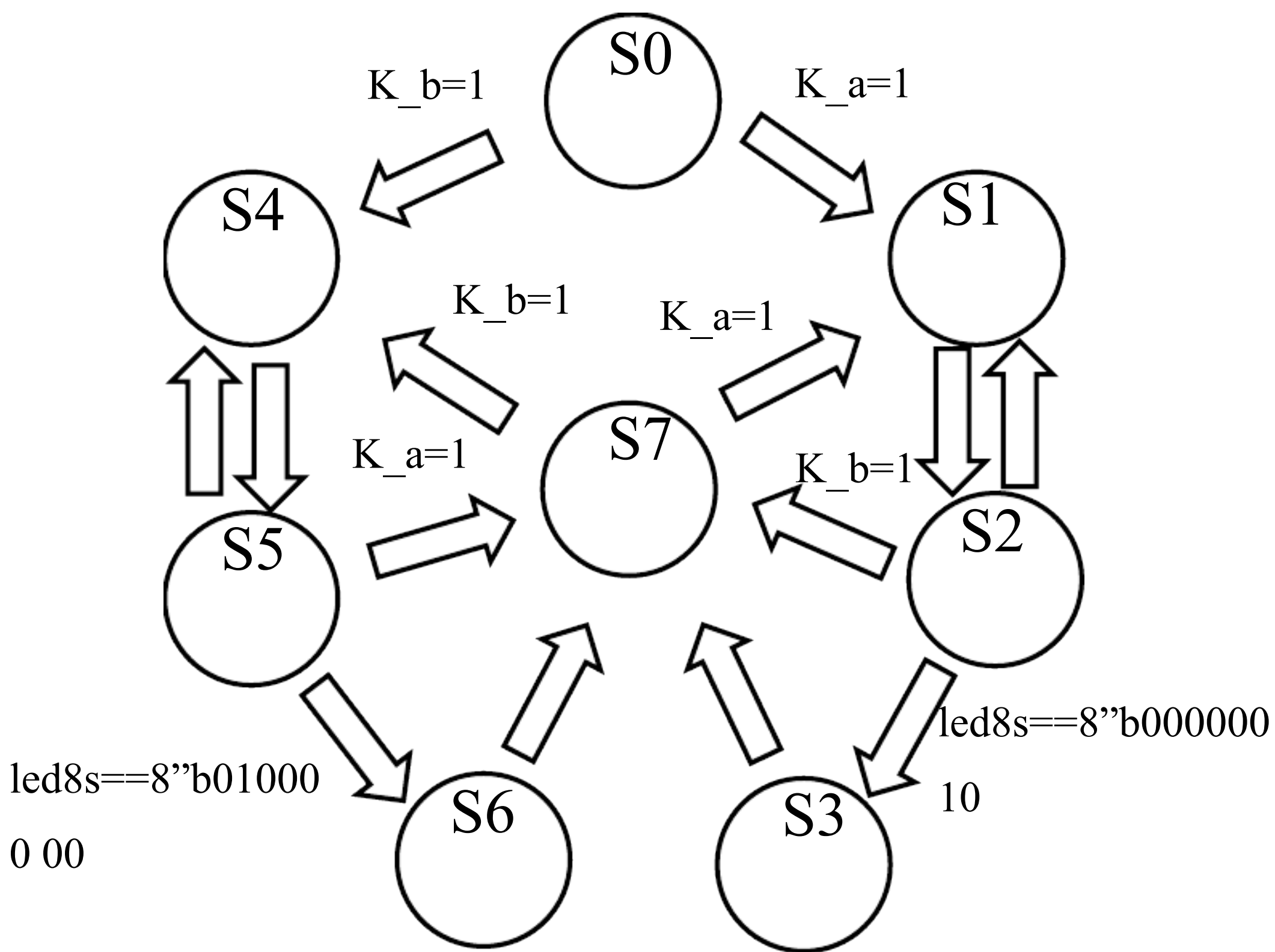
考虑到玩耍的简单度，承受状态机来实现这些模式转换。用到 8 个状态，如下所示：

状态	功能
S0	复位状态，也是推断发球权的状态
S1	甲方发球
S2	球从甲方向乙方右移,假设此时乙方接球
S3	开头推断乙方是否在 0.5S 内接球
S4	乙方发球
S5	球从乙方向甲方左移,假设此时甲方接球 就给乙方加一分，并将球权给乙方
S6	开头推断甲方是否在 0.5S 内接球

程序还参加了蜂鸣、用数码管显示分数和发球权等功能。

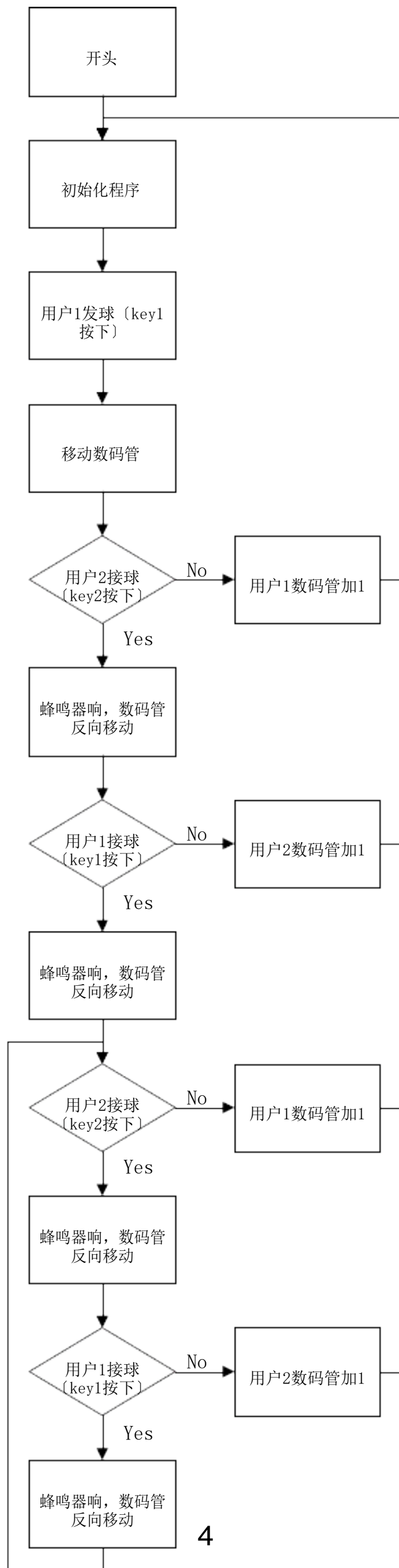
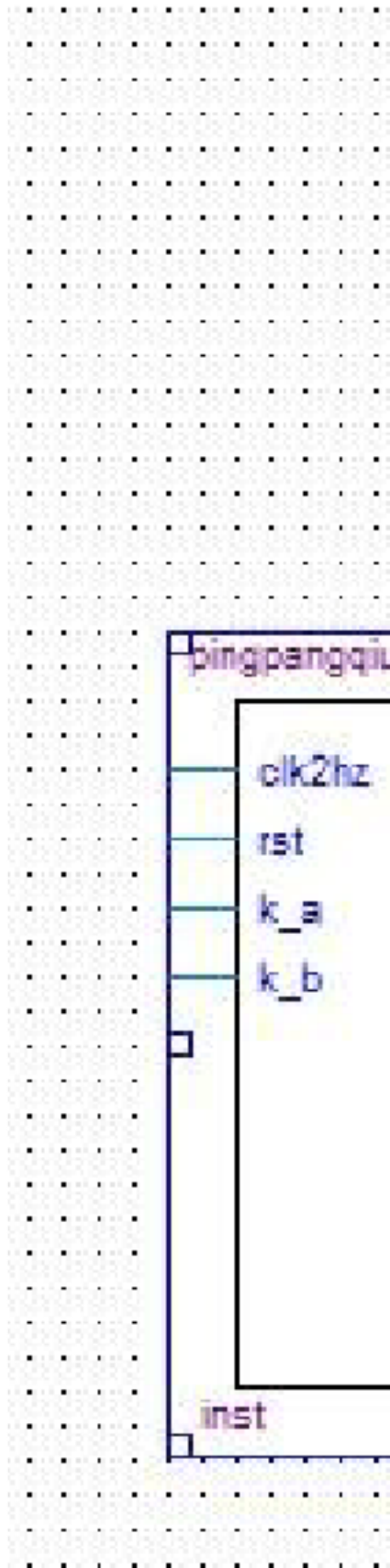
二构造框图

1 状态转移图



2 程序流程图

3 系统模块



三系统程序

```
module
pingpangok(state,clk2hz,rst,k_a,k_b,led8s,led7s1,led7s2,led7s3,led7s4,led7s5,speaker
);
//clk2hz指一个2Hz的时钟输入,rst异步复位,k_a和k_b是竞赛双方按键,speaker是
蜂鸣器
//led8s是一排8个发光二极管代表球场,led7s1~4是4个显示分数的数码管,led7s5
是显示发球权的数码管
input clk2hz,rst,k_a,k_b;
output [7:0] led8s;
output [6:0] led7s1,led7s2,led7s3,led7s4,led7s5;
output speaker;
output [2:0] state;
reg [7:0] led8s;
reg [6:0] led7s1,led7s2,led7s3,led7s4,led7s5;
reg [3:0] score_a,score_b;
reg [2:0] state;
reg speaker;
parameter s0=3'b000,//复位状态,也是推断发球权的状态;
          s1=3'b001,//甲方发球;
          s2=3'b010,//球从甲方向乙方右移,假设此时乙方接球就给甲方
          加一分,并将球权给甲方;
          s3=3'b011,//开头推断乙方是否在0.5S内接球
          ; s4=3'b100,//乙方发球;
          s5=3'b101,//球从乙方向甲方左移,假设此时甲方接球就给乙方
          加一分,并将球权给乙方;
          s6=3'b110;//开头推断甲方是否在0.5S内接球
          ;s7=3'b111;//用于推断接球后发球;

//_____玩耍规律掌握核心_____
-----
always @(posedge clk2hz) begin
    case(state)
        s7: begin
            if(k_a)
                state<=s1; //甲发球;
            else if(k_b)
```

```

        state<=s4;//乙发球;
    else if(speaker)
        speaker=1'b0;
    else
        speaker=1'b0;
    end
    s0: begin
        led7s5<=7'b0111111;//第五个数码管显示数字
        led8s<=7'b00000000;
        score_a<=0;
        score_b<=0;//给甲乙双方的分数复位;
        if(k_a)
            begin
                speaker<=1'b1;//由于按键k_a要持续按着保持高电位"1",蜂鸣器鸣叫是提示可以松手了;
                state<=s7;
            end
        else if(k_b)
            begin
                speaker<=1'b1;
                ;state<=s7;
            end
        end
    else
        state<=s0;
    end
    s1: begin
        led7s5<=7'b0000110;//第五个数码管显示数字
        if(led8s==8'b00000000)
            led8s<=8'b10000000;//点亮的灯
            熄灭在最右边,即甲方发球;
        state<=s2;
        if(speaker)
            speaker<=1'b0;//将刚刚鸣叫的蜂鸣器
            关闭;
        end
    s2: begin
        if(led8s==8'b00000010)
            begin

```