

课程设计报告

**中文题目 ：** **基于数字系统的 2048 游戏**

**英文题目：****A 2048 Game Based On a Digital System**

|  |  |
| --- | --- |
| 小组成员1： | 彭一炜 |
| 专业： | 计算机科学与技术 |
| 小组成员2： | 陈奕萱 |
| 专业： | 软件工程 |
| 小组成员3： | 赵佳乐 |
| 专业： | 软件工程 |
| 学院： | 计算机科学与技术学院 |
| 指导教师： | 王跃明、洪奇军 |
| 报告日期： | 2024年01月17日 |

摘要

本次课程设计设计了一个带VGA显示的博弈小球的游戏，该游戏由两个玩家进行。开始时，屏幕中央有一个小球a，同时在屏幕上还有四个看不到的“小球”b、c、d、e。首先玩家1通过四个按钮上下左右移动位于显示屏中央的小球a，若小球碰不到隐身的小球，则玩家1可以选择继续走或者将球权交给玩家2；若碰到隐身的小球，则小球a会变大，同时玩家将球权交给玩家2，重复上述规则。当碰到最后一个隐身小球的玩家输掉比赛。

课程设计在Sword Kintex7 实验平台上进行，在Xilinx ISE12.4 开发环境中运行，采用Verilog语言进行编写，由小组成员分工合作完成。

**关键词：**　VGA，博弈，小球

目录

[第1章 绪论 4](#_Toc11975)

[第2章 ……设计原理 5](#_Toc21687)

[第3章 ……设计实现 5](#_Toc8112)

[第4章 系统测试验证与结果分析 6](#_Toc21911)

[第5章 结论与展望 6](#_Toc7459)

1. **绪论**
   1. **博弈小球游戏设计背景**

这个课程设计是一个将VGA与博弈很好地结合的游戏，排除玩家恶意死循环的情况，每一次抉择都是一次博弈。这个游戏借鉴了课程中的部分实现代码，在此基础上加上我们自己的创意和代码，最终较好地呈现了这个游戏。

这个游戏的代码量不算太多（比起较为复杂的游戏），但游戏运行成功的那一刻仍能够带给设计者成就感，还能够在一定程度上表现出设计者使用该语言和配套思想能够做到什么程度，适合使用者拿来练手和熟悉语言、思想。

进行这个课程设计希望能够检测自己的部分实验情况，学习部分课程实验外的内容，同时也希望能够自己实现一个可以实现基本功能的游戏。

* 1. **主要内容和难点**

该设计可以使用之前的实验中使用到的模块来进行辅助，其他的需要完成的内容有：

1. 设计VGA显示模块。这个模块输入25MHz的时钟，通过计数产生行扫描信号和场扫描信号，并从给定的RGB值得到输出用到的RGB值。
2. 设计隐形小球存在和运动小球的操作模块。小球的位置由初始化定义的x，y坐标确定，隐形小球的位置一旦确定便不再更改，运动小球运动方向控制由按键输入控制，而且小球在离开屏幕后能从另一方向回到屏幕中。
3. 当运动小球与隐形小球相遇后，运动小球变大，在本回合中，已被碰到的隐形小球不会再使运动小球变大，重难点在于运动小球与隐形小球相遇的判定条件的设置、运动小球变大的实现以及异性小球的二次访问无效性。
4. **博弈小球游戏设计原理**
   1. **博弈小球游戏设计相关内容**
5. VGA显示

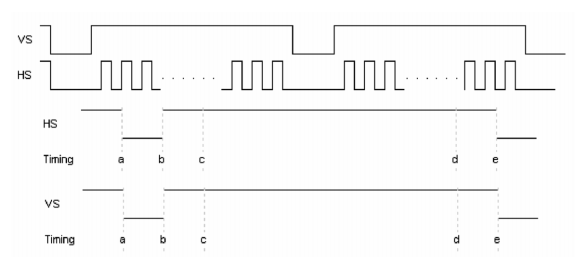
在本次作业中，我们利用了实验板中的 VGA 显示屏，通过对于 VGA 时序控制信号以及输出颜色的控制，实现了显示屏信号的分时输出。

VGA 是 IBM 在 1987 年随推出的一种视频传输标准，具有分辨率高、显示速率快、颜色 丰富等优点，在彩色显示器领域得到了广泛的应用。

一般所使用的 VGA 显示器均为标准五输入类型，如下表所示。通过对行列同步信号进行控制，遍历输出各个像素点的颜色信息，最终输出完整图像。



信号线中 HS 与 VS 用于控制显示器的显示频率，R、G 与 B 则用于控制显示器当前像素的颜色。



上图是 VGA 的标准时序。不难看出，列同步信号与行同步信号的时序都分

为 4 段。a-b 段称为同步期，此时同步信号为低电平；b-c 段称为消隐后肩，

c-d 段称为显示期，是数据的有效区域，d-e 段称为消隐前肩，这三段的同步信

号为高电平。显示器就是根据列同步信号与行同步信号的周期，来决定显示的分

辨率、频率等参数。不同分辨率与频率对应的 VS-HS 数值，可以通过查表得到。

在实验中只需要在一个 VS-HS 完成一个周期的时间内，按从左上角到右下角的顺序输出各个像素点的颜色，就能在显示器上显示出对应的图案。需要注意的是，VGA 颜色的输出必须严格遵循时序，只能在 c-d 段输出非 0 数据，否则会对显示造成干扰。

VGA显示依赖于行扫描信号、场扫描信号和当前像素的RGB值。电子枪从左上角开始从左向右扫描，一行扫描完后，进行行消隐（使电子束不再发射，同时将其回归到下一行的最左边，此时行扫描信号为0），接着扫描下一行，当所有行扫描完后，进行场消隐（电子束回到第一行的最左边，此时场扫描信号为0），在每次指定的时钟的上升沿，输出场扫描信号、行扫描信号和对应的RGB值。

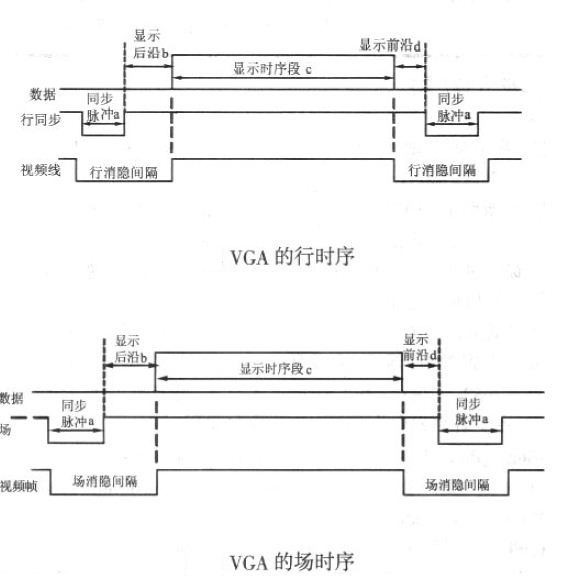


图1 VGA的行、场时序

1. 小球的移动和绘制

我们定义了小球的横纵坐标以及半径，用于表示小球的位置及形状，再通过输入的交互选择对其横纵坐标进行加减以实现小球的移动；根据小球球心与隐形小球球心距离与半径之和的关系判断两小球是否发生碰撞，若碰撞则运动小球半径增加以实现小球的变大。

* 1. **博弈小球游戏设计方案**

要设计的博弈小球游戏，输入有5个按键，分别是左、右、上、下、游戏复位，游戏复位键控制状态的改变，上下左右键控制小球的移动，这些按键输入到运动控制模块，该模块再根据状态控制模块输出的状态改变小球的位置，同时判断运动小球与隐形小球的位置关系，判定小球半径的变化与否输出。为增加游戏的趣味性，通过调节sw[12：1]来改变小球的颜色。RGB设置部分根据小球和的位置及当前半径设置RGB，VGA显示模块根据RGB的设置输出RGB，根据时钟的计数输出行、场扫描信号。

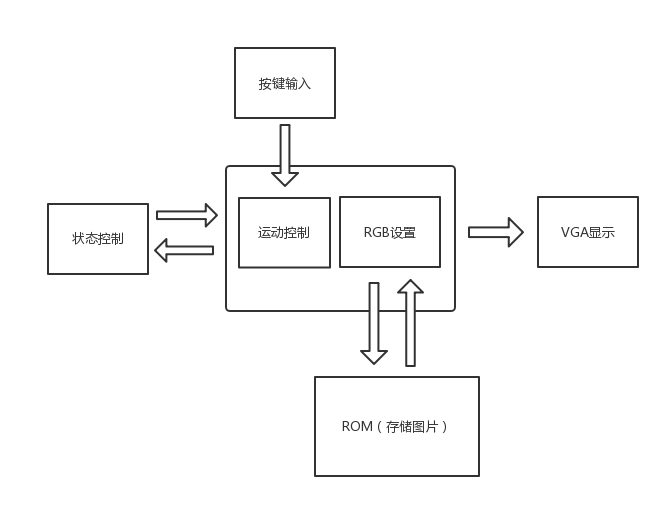


图2 程序结构框图

* 1. **系统软件设计**

软件设计中的各子模块基本如图2所示，分频模块和按键防抖动模块没有列出来，分别设计这些子模块后，在顶层模块中进行变量声明和模块调用。

（1）防抖动模块被按键输入模块调用。按键输入模块将阵列形式的按键输入作为输入，输出按键按下信号和按键键值。部分代码如下， keyX指定键值的低位， keyY指定键值的高位，若键值的两部分都有意义，ready\_raw（按键信号）为1。

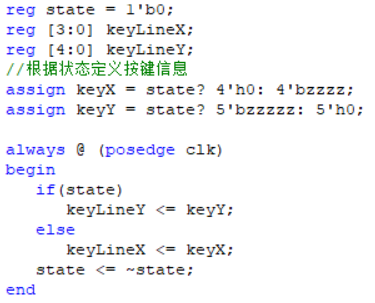


图3 根据状态定义按键信息

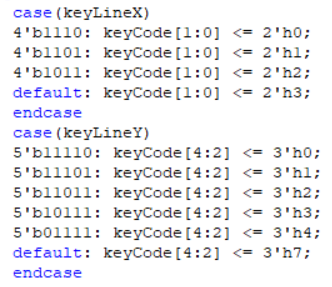


图4 获取按键信息

（2）vga扫描模块

VGA显示模块。定义两个变量来计数，当数字大于消隐信号后，数字减去消隐信号的大小就得到了对应像素在屏幕上的x，y值（以左上角为坐标原点），出于消隐信号时，对应的扫描信号设置为0，否则设置为1，且只有像素处于屏幕上时，才能指定RGB值。

分别进行行扫描和列扫描，

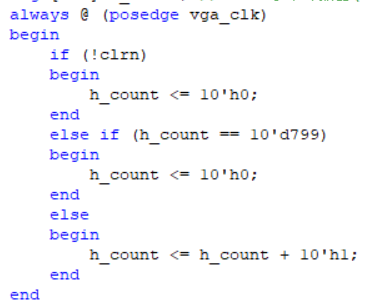


图5 行计数

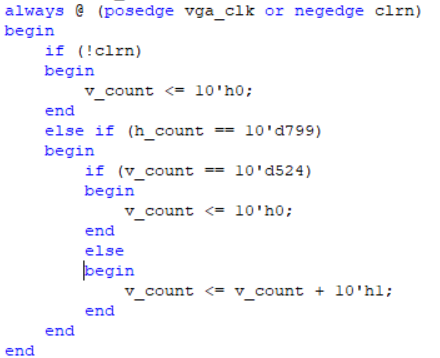


图6 列计数

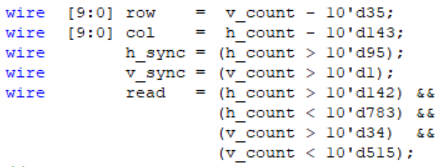


图7 判断是否大于消隐信号

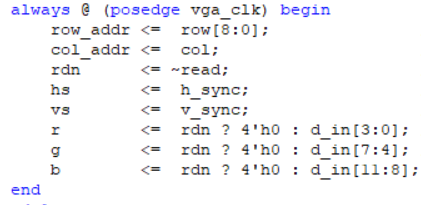


图8 指定RGB信号

（3）运动控制模块

运动控制模块输入场扫描信号VS和当前状态，若复位，则所有信号恢复到游戏开始；反之，在VS上升沿，小球根据按键输入方向进行移动。如果碰到隐形小球，运动小球的半径增大，同时定义该隐形小球触碰无效，在本回合中不会再次被碰到。

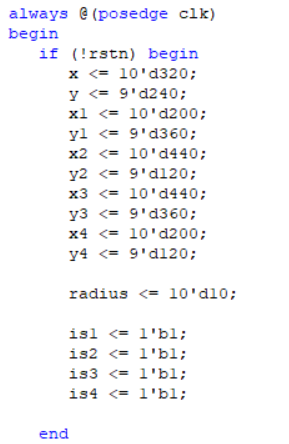


图8 初始化运动小球与隐形小球

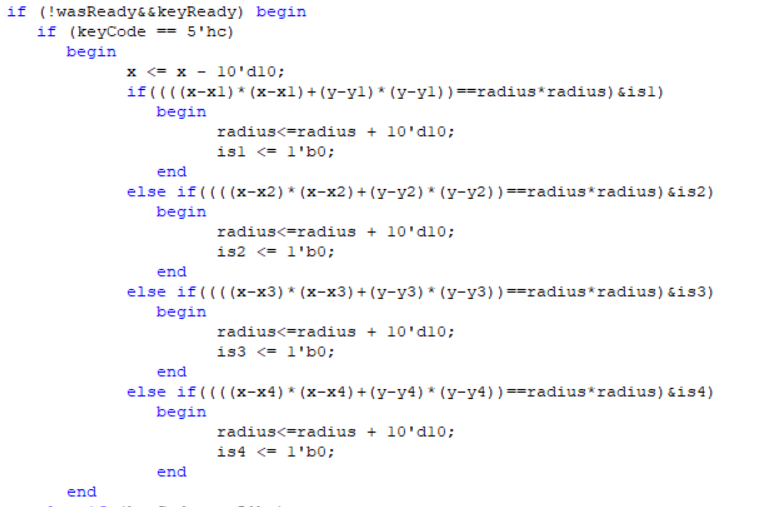


图9 小球向左运动

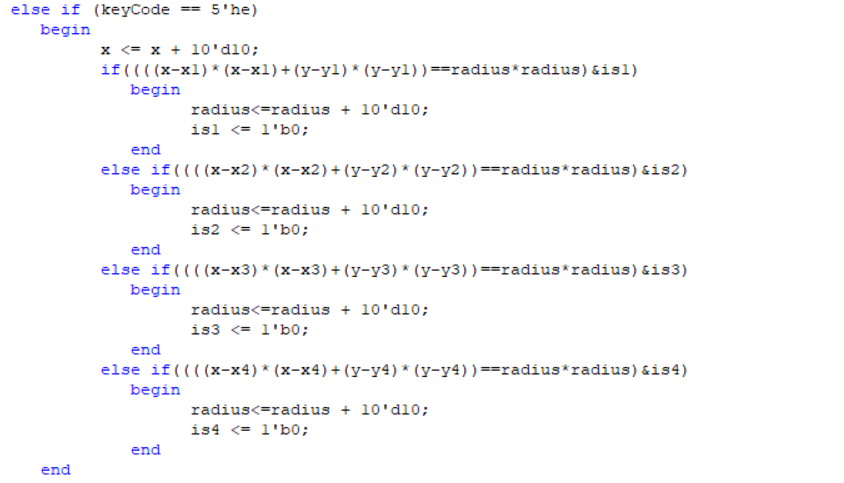


图10 小球向右运动

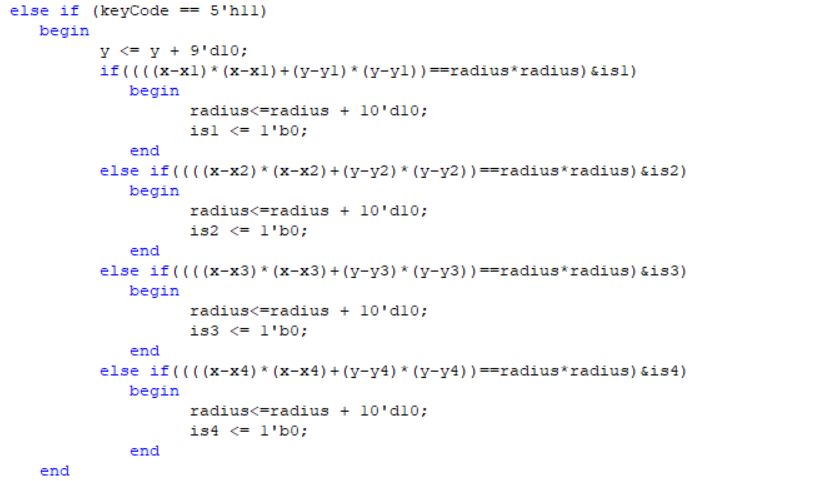


图11 小球向下运动

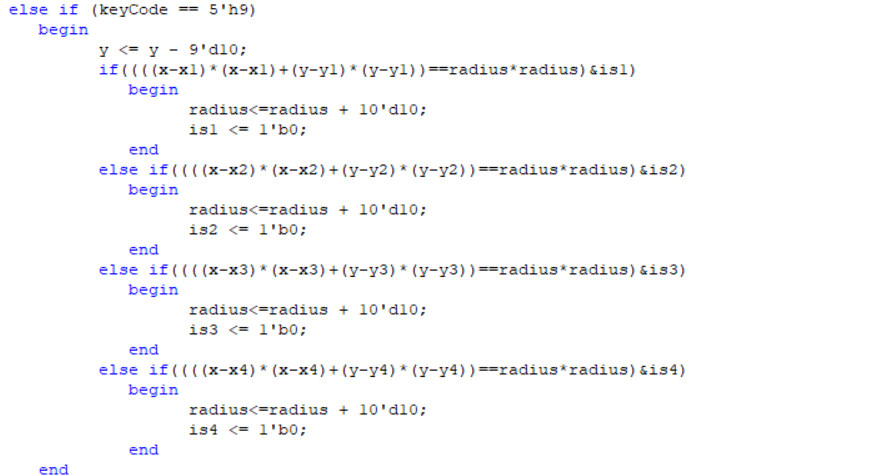
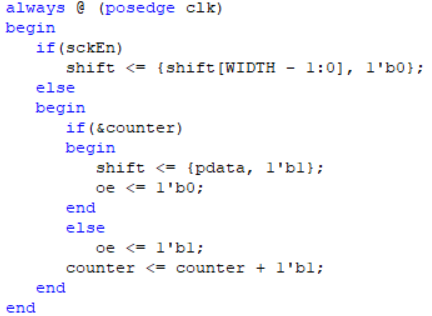


图12 小球向上运动

（4）RGB设置模块。

调节sw[12:1]，设置小球颜色。

首先获取像素值，若当前像素为球，设置为相应的颜色；若不是，则设置为白色。



1. **弹跳小球游戏设计实现**
2. 创建工程；
3. 调用分频模块对100MHz时钟进行分频，调用按键输入模块对输入进行分析和防抖动，实现状态控制模块，vga模块，运动控制模块（其中包含RGB设置功能）。

Game

AntiJitter

keyboard

Seg7Device

vga

AntiJitter

ShiftReg

图13 模块层次图

1. **系统测试验证与结果分析**
   1. **功能测试**
2. 状态变换。

程序开始时，状态为初始化小球的位置在屏幕中央；

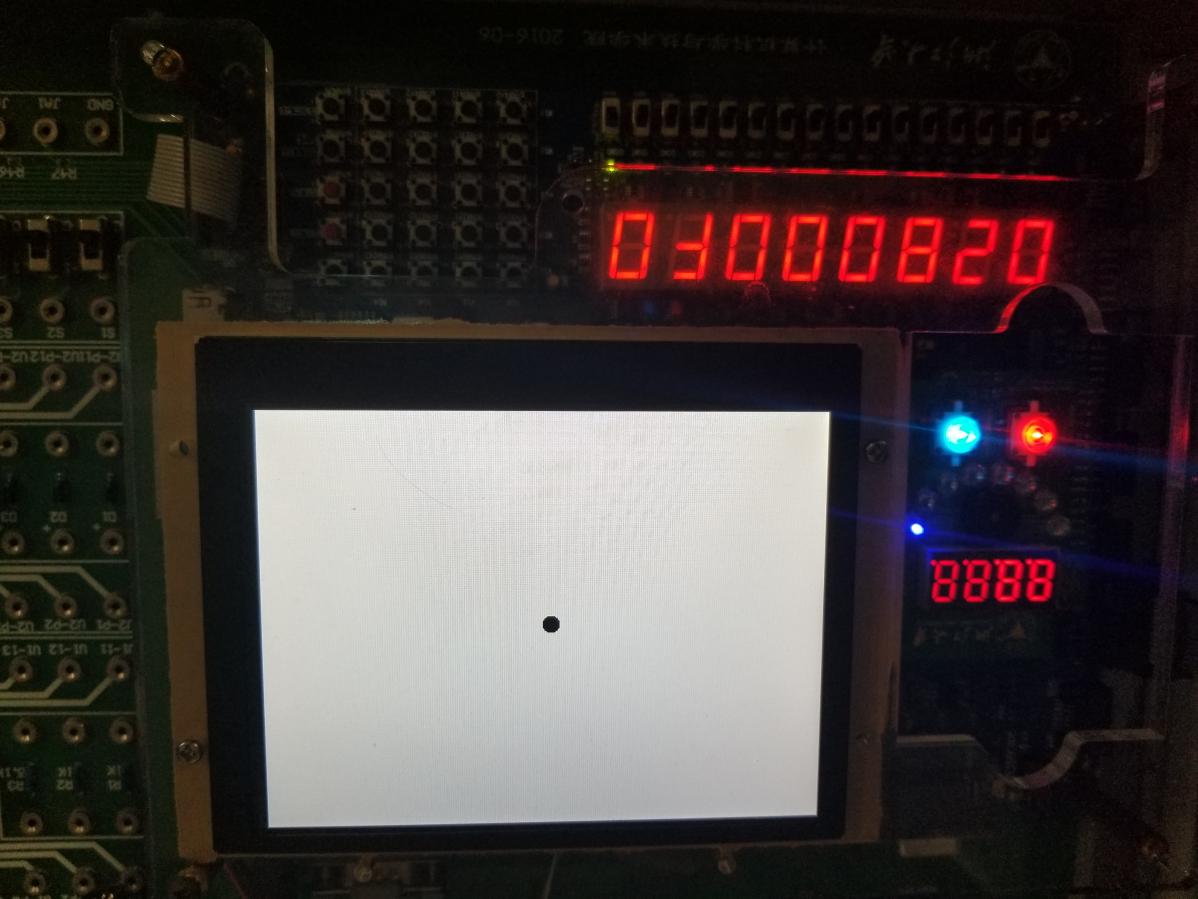


图14 游戏开始

按上下左右键控制小球移动，若运动小球未碰到隐形小球大小不变，

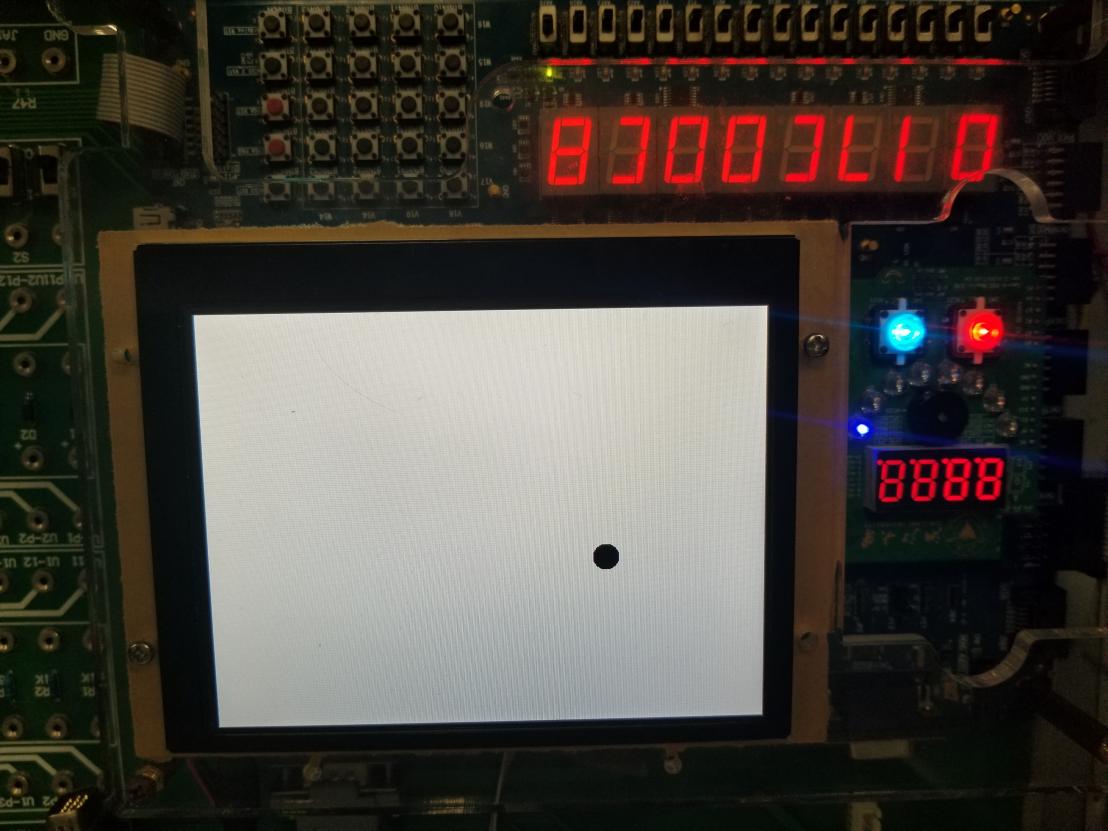


图15 运动小球未碰到隐形小球

若运动小球碰到隐形小球，则运动小球变大，且在此回合中，已碰到的隐形小球不会再引起运动小球变大；

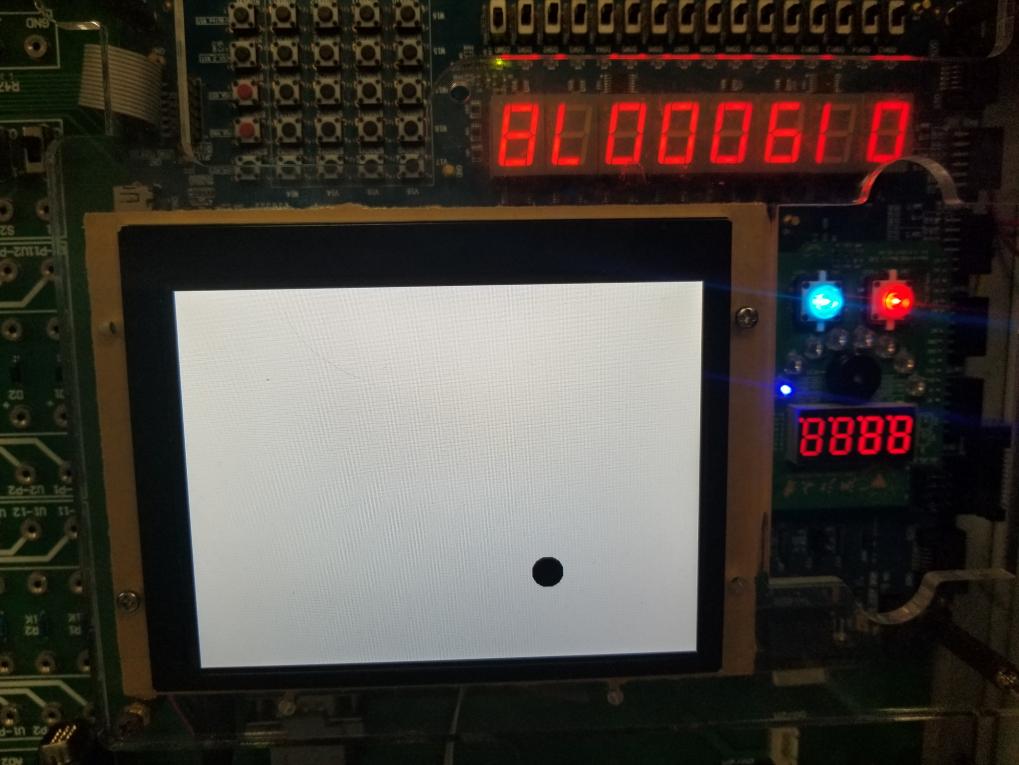


图16 碰到隐形小球

一共四个隐形小球，都触碰完后不会再使运动小球变大；

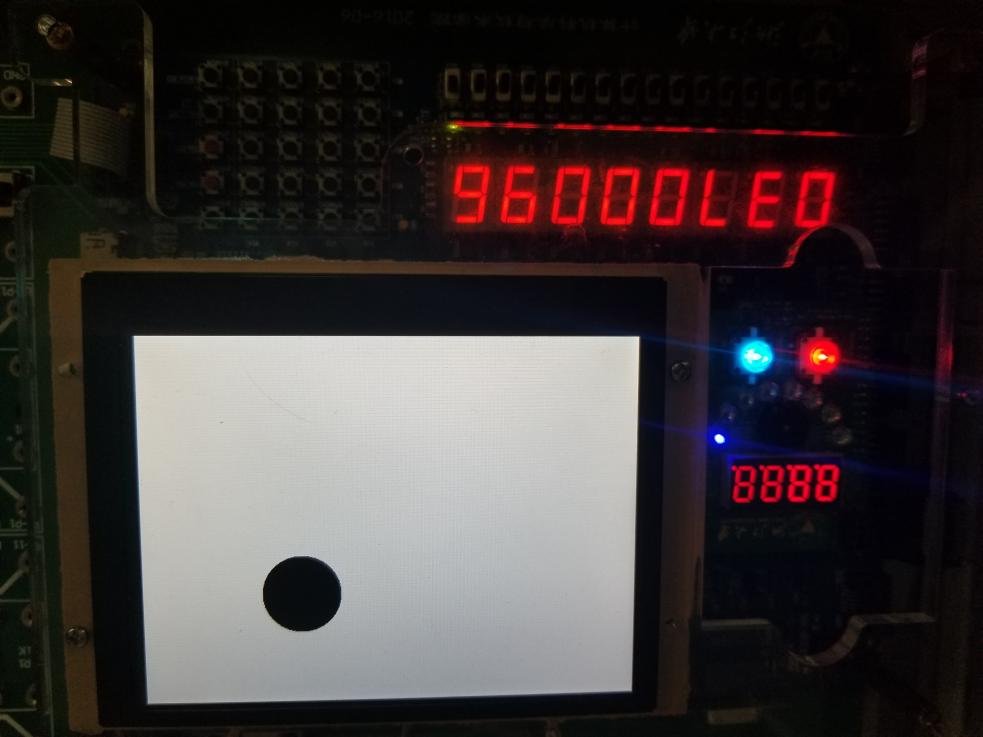


图17 隐形小球触碰完毕

按下复位键，回到游戏开始状态，运动小球回到屏幕中间，所有隐形小球重新定义。

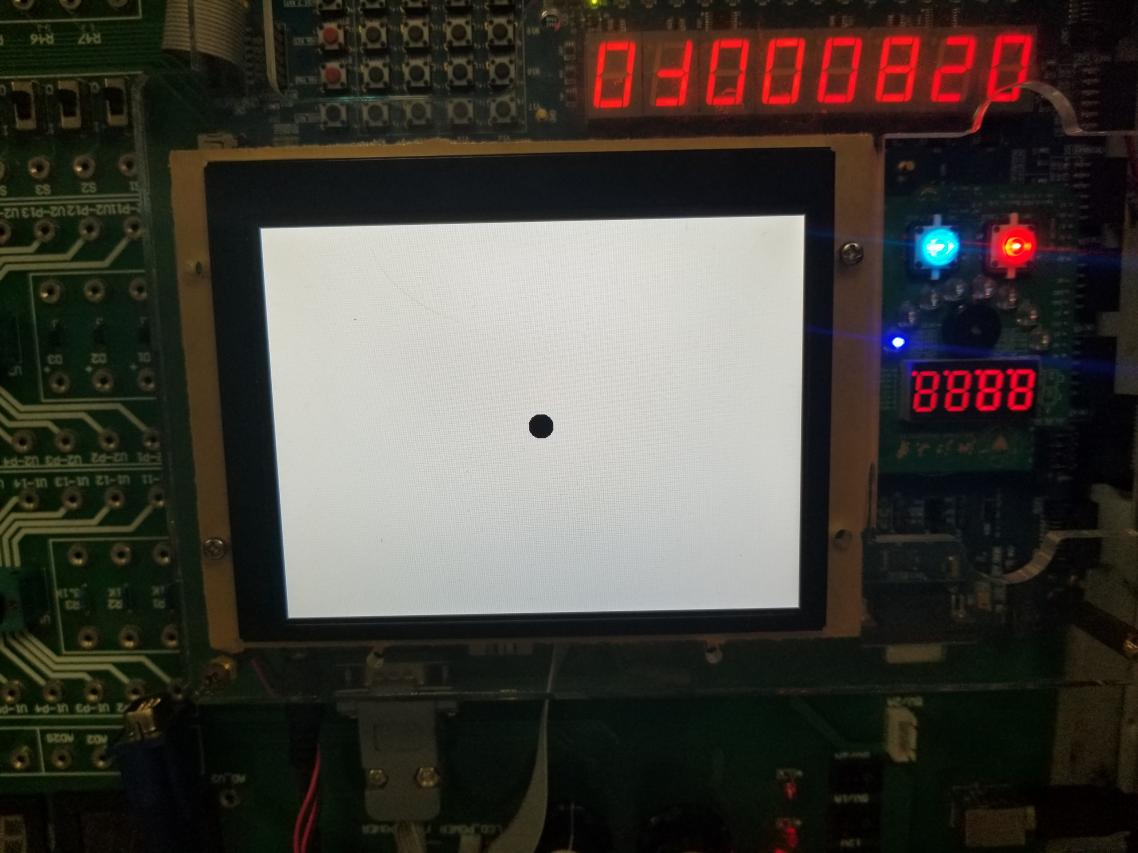


图18 复位

在整个过程中，译码器高16位显示运动小球的横坐标，低16位显示小球的纵坐标。



图19 小球的横纵坐标显示

* 1. **结果分析**

状态变换成功转换。

小球运动成功，只是有时候小球会直接穿过挡板，导致游戏结束。可能的原因是使用的时钟不同步若小球移出屏幕，会从相反一端再次移入屏幕。

横纵坐标显示成功，显示为16进制。

* 1. **系统演示与操作说明**

1. 程序初始化后，小球初始化位置，按下运动按键运动；
2. 使用按键“13”控制左，按键“15”控制右，按键“9”控制下，按键“11控制上；
3. 当运动小球碰到隐形小球后变大；
4. 当四个隐形小球都被触碰后，运动小球不会再变大，游戏结束。可按按键“rst复位进入初始化状态，此时小球重新初始化。
5. **结论与展望**

在进行课程设计时，遇到的最多的问题有两个：

1. Verilog报错，但是报错信息并不明晰，无法很快找到错误，只能通过自我检验和排除法进行Debug；
2. 显示屏显out of range，后来经过查资料发现时时钟频率错误；
3. 显示出来的程序不符合自己的要求，并出现许多莫名其妙的错误，无从下手去更改错误。
4. IP核创建时，一直显示coe文件错误，导致最后未能导入图片。

这些错误产生，很多时候是因为自己对于Verilog语言的语法不清楚，或者是认为应该正确的地方，因为时序的原因而产生许多bug。写Verilog程序时不能只顾逻辑正确，还要想到时序电路中时序可能导致的问题。