****

**数字电路与逻辑设计**

**实验报告**

姓名：徐汤柯、王锡淮

学号：16337265、16337236

院系：数据科学与计算机学院

专业：计算机类

指导老师：陈云洽

年级班级：16级计科7班

实验名称：基于basys3实验板的2048小游戏

目录

[一、 实验背景 2](#_Toc485631029)

[1、 前言 2](#_Toc485631030)

[2、 VGA简介 2](#_Toc485631031)

[3、 2048游戏介绍 3](#_Toc485631032)

[二、 需求分析 4](#_Toc485631033)

[三、 可行性分析 4](#_Toc485631034)

[1、 器件可行性 4](#_Toc485631035)

[2、 技术可行性 4](#_Toc485631036)

[四、 实验设计 4](#_Toc485631037)

[1、 实验器材 4](#_Toc485631038)

[2、 模块设计 5](#_Toc485631039)

[3、 代码分析 6](#_Toc485631040)

[4、 源文件 7](#_Toc485631041)

[5、 演示视频 7](#_Toc485631042)

[五、 总结及心得体会 7](#_Toc485631043)

[1、 总结 7](#_Toc485631044)

[2、 心得体会 7](#_Toc485631045)

[ 王锡淮： 7](#_Toc485631046)

[ 徐汤柯： 7](#_Toc485631047)

[六、 参考资料 8](#_Toc485631048)

# 实验背景

## 前言

本小组计划使用verilog语言，在basys3 实验板上实现2048小游戏，并通过其VGA

接口将游戏显示在支持VGA的显示屏上。

## VGA简介

1987年IBM放弃了其所创立的PC机，推出了一种新结构的个人计算机系PS/2，

VGA即是作为PS/2的标准显示系统随PS/2一起推出的。VGA被制作在PS/2的主

板上，而不是一个单独的插卡。从兼容角度考虑，IBM还单独提供了供PC机使用的

VGA显示卡。

VGA采用光栅扫描电子束按固定的路径扫过整个屏幕，在扫描过程中，通过电子

束的通断强弱来控制电子束所经过的每个点是否显示或显示的颜色。

光栅扫描的路径通常为：从上到下扫过每一行，在每一行内从左到右扫描。

其扫描过程如下：

电子束从屏幕的左上角开始向右扫，当到达屏幕的右边缘时，电子束关闭(水平消

隐)，并快速返回屏幕左边缘(水平回扫)，又在下一条扫描线上开始新的一次水平扫

描。一旦所有水平扫描均告完成，电子束在屏幕的右下角结束并关闭(垂直消隐)，然

后迅速返回到屏幕左上角(垂直回扫)，开始下一次光栅扫描。

Basys 3实验板的VGA支持：

VGA视频显示部分的电路如图所示。我们所用的电阻搭的12bit(212色)电路，由

于没有采用视频专用DAC芯片，所以色彩过渡表现不是十分完美。



## 2048游戏介绍

2048游戏是一种操作及界面都比较简单的小游戏。

每次控制所有方块向同一个方向[运动](http://www.yxdown.com/tag/23/)，两个相同[数字](http://www.yxdown.com/tag/175/)的方块撞在一起之后合并成

为他们的和，每次操作之后会在空白的方格处随机生成一个2或者4，最终得到一

个“2048”的方块就算胜利了。如果16个格子全部填满并且相邻的格子都不相同也

就是无法移动的话，那么恭喜你，gameover。每次控制所有方块向同一个方向运

动，两个相同数字的方块撞在一起之后合并成为他们的和，每次操作之后会在空白

的方格处随机生成一个2或者4，最终得到一个“2048”的方块就算胜利了。如果

16个格子全部填满并且相邻的格子都不相同也就是无法移动的话，那么恭喜你，

gameover。

# 需求分析

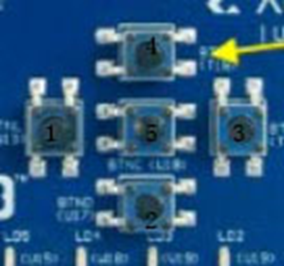
# 可行性分析

## 器件可行性

Basys 3实验板有5个按键开关，可以用1、2、3、4、5按键分别代表向左、向

下、向右、向上移动和重新开始游戏，并且其提供所需时钟信号；还有VGA接口，

可以在屏幕上显示不同的颜色，以显示游戏界面。



支持VGA的显示器能够与Basys 3 实验板的VGA输出兼容。

## 技术可行性

# 实验设计

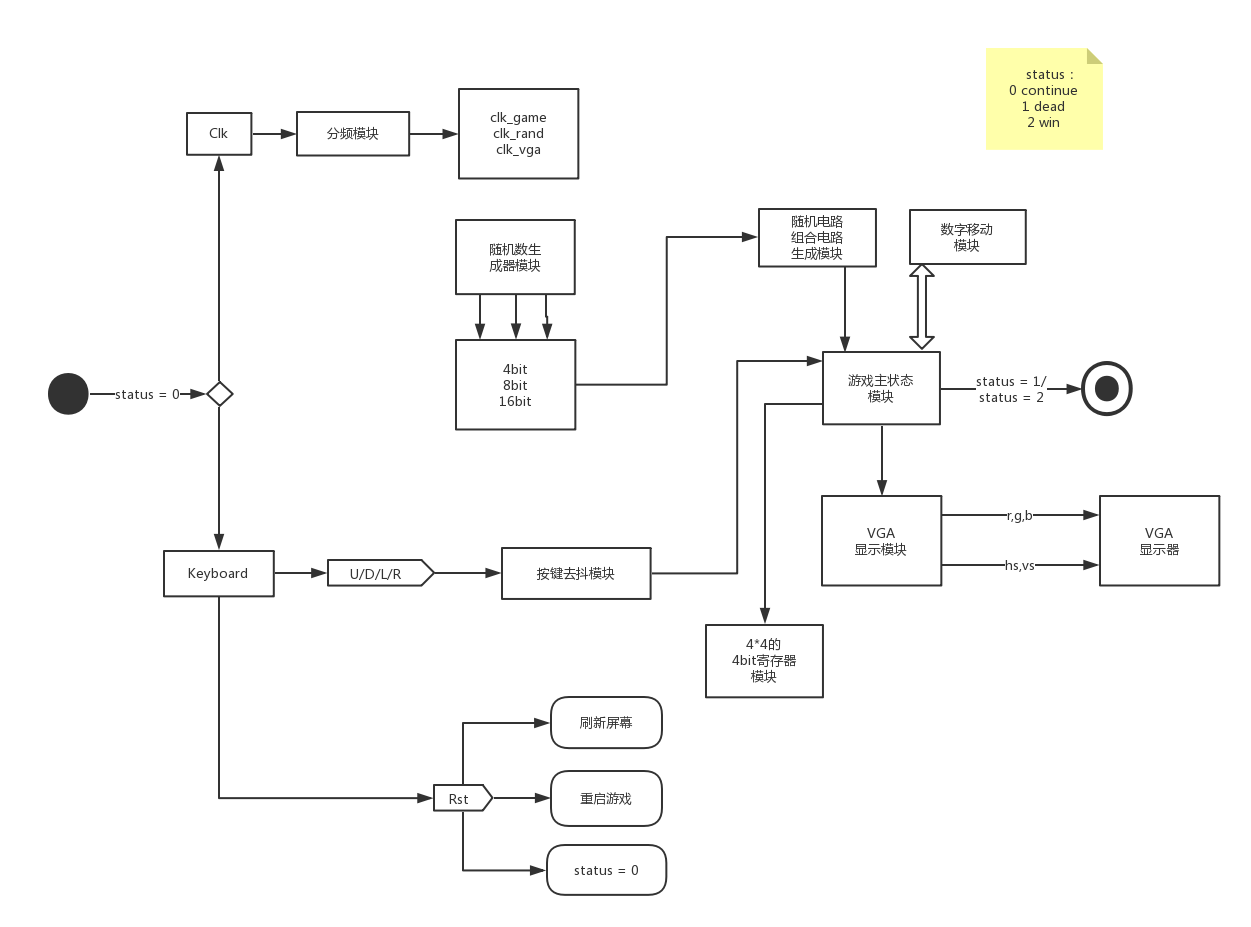
## 实验器材

Basys 3实验板一块及其连线；

采用VGA技术的显示屏一个及其连线；

装有vivado2016的电脑。

## 模块设计



（此图在电子档实验报告的doc文件夹中有）

本程序分为八个模块：

1. Game：运行游戏的主模块，决定游戏的状态，处理输入与输出的数据，调用各个模块。

module game(rst,clk,key,Hsync,Vsync,vgaRed,vgaGreen,vgaBlue);

input rst,clk;

input [3:0] key;//origin input

output Hsync,Vsync;//VGA

output [3:0] vgaRed,vgaGreen,vgaBlue;//VGA

1. Clkdiv：分频模块，将主频率分为clk\_game, clk\_rand, clk\_vga。

module clkdiv(rst,clk,clk\_game,clk\_rand,clk\_vga);

output wire clk\_game,clk\_rand,clk\_vga;

input wire clk,rst;

1. Seg：数字以log形式保存，即0~2048记为0~11，该模块将数字转换为决定颜色的288位数组。

module seg(din,dout);

input [3:0] din;

output reg [0:287] dout;

1. Vga：决定输出的颜色。

module vga(rst,dclk,db,r,g,b,hs,vs,x,y);

input dclk,rst;

input [2:0] db;

output r;

output g;

output b;

output reg hs,vs;

output reg [9:0] x,y;

1. Shake：使用一个32位fifo数组来按键去抖。

module shake(rst,clk,din,dout);

input rst,clk,din;

output reg dout;

1. Random：生成4位，8位，16位的随机数。

module random(rst,clk,ran2,ran3,ran4);

input rst,clk;

output reg [7:0] ran2;

output reg [11:0] ran3;

output reg [15:0] ran4;

1. Gen：在按下按键对应的位置生成一列或一行随机数与三个空位。

module gen(rst,clk,i3,i2,i1,i0,o3,o2,o1,o0);

input rst,clk;

input [3:0] i3,i2,i1,i0;

output reg [3:0] o3,o2,o1,o0;

1. Move：根据按键移动与合并16个数字。

module move(i3,i2,i1,i0,o3,o2,o1,o0);

input [3:0] i3,i2,i1,i0;

output reg [3:0] o3,o2,o1,o0;

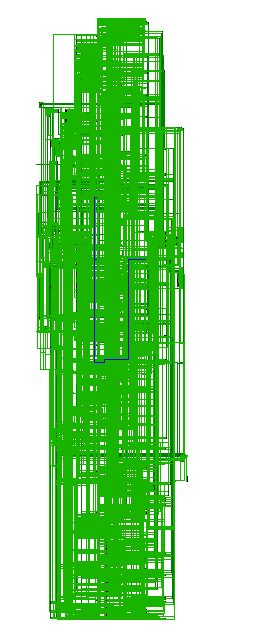
## 代码分析

代码的重点分析：

1. game模块共有三个always语句，其中第一个always语句是处理按键的输入与游戏的改变输出，即数字的大小及位置的改变；第二个always语句是处理屏幕颜色的变化，其将640\*480分辨率的显示屏等分为160\*120个格子；第三个always语句处理笑脸随游戏状态的改变。
2. vga模块通过控制屏幕的颜色与水平和垂直信号来控制屏幕的显示，并且其将当前扫描的位置通过x，y输出到game模块。
3. shake模块通过一个32位的FIFO数组处理按键输入与硬件内部频率的差异，实现去抖。
4. seg模块根据输入（即数字的大小），决定配色方案，配色方案由一个288位的数组决定，每一个配色即为3位r, g, b，配色来自所参考的开源项目。
5. move根据按键的输入移动数字并将它们合并，由此产生一个空行或空列。
6. gen模块在move模块产生的空行或空列里产生一个随机数和3个空位，该随机数由random模块产生。

## 逻辑电路图

见电子档 doc文件夹 circuit.pdf



## 源文件

电子档 src文件夹内为verilog源代码；

2048game文件夹内为完整项目。

## 演示视频

见电子档 演示视频.mp4

# 总结及心得体会

## 总结

## 心得体会

### 王锡淮：

在这次的实验进行的过程中，

### 徐汤柯：

# 参考资料

1. 《Verilog经典教程》 夏宇闻 著
2. 《VGA实用编程技术》罗建军 著 清华大学出版社
3. <http://www.digilent.com.cn> 开源项目 基于Basys3的2048小游戏项目的DIY动手指南