



# 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 .....	1
1、前言 .....	1
2、VGA 简介 .....	1
3、2048 游戏介绍 .....	2
4、实验环境 .....	2
二、需求分析 .....	2
三、可行性分析 .....	3
1、器件可行性 .....	3
2、技术可行性 .....	4
四、实验设计 .....	4
1、实验器材 .....	4
2、模块设计 .....	5
3、代码分析 .....	6
4、演示视频 .....	错误!未定义书签。
五、总结及心得体会 .....	7
1、总结 .....	7
2、心得体会 .....	8
☆ 王锡淮 : .....	8
☆ 徐汤柯 : .....	8
附录 .....	9
参考资料 : .....	9
特别鸣谢 : .....	9

# 一、 实验背景

## 1、 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。

## 2、 VGA 简介

VGA(Video Graphics Array)是 IBM 在 1987 年随 PS/2 机一起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。支持热插拔，不支持音频传输。

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。

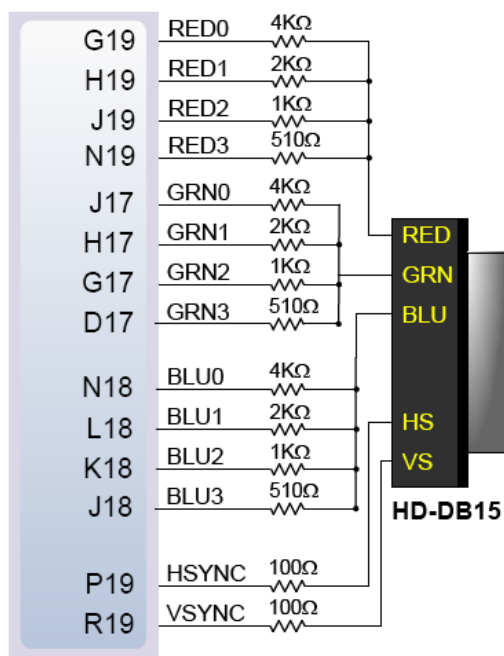
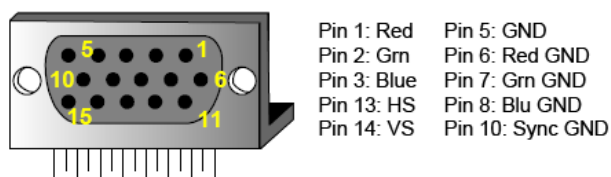
光栅扫描的路径通常为：从上到下扫过每一行，在每一行内从左到右扫描。

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。



### 3、 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，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 或者 8，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。

### 4、 实验环境

编程语言：Verilog

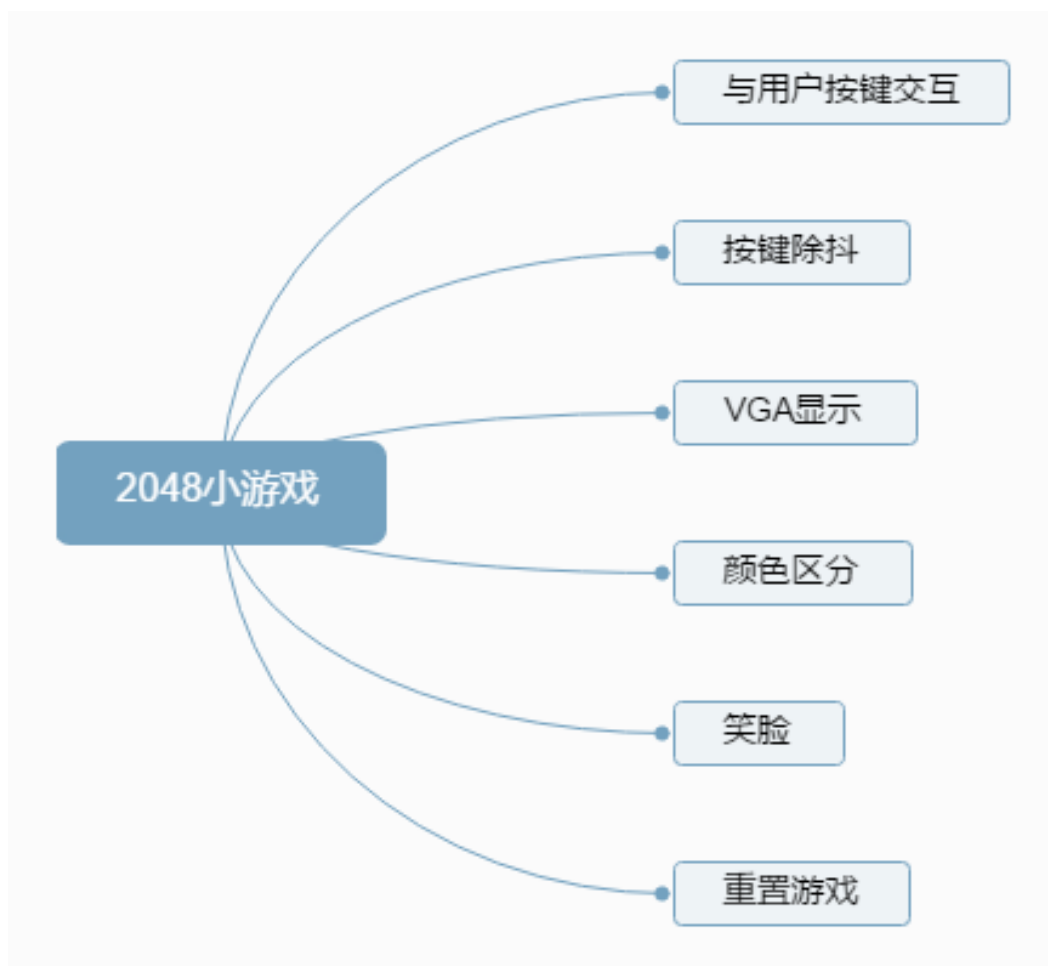
开发环境：Xilinx Vivado 2015.3

## 二、 需求分析

本次实验我们要实现 2048 游戏，游戏一共有 5 种操作。当按下相应的按键后需要程序执行相应的操作，并且防止抖动产生预期之外的情况。

在屏幕上通过 VGA 显示游戏界面，同时使操作后及时在显示屏上反映出操作的结果，并且要根据数字在不同的方格中显示出不同的颜色。

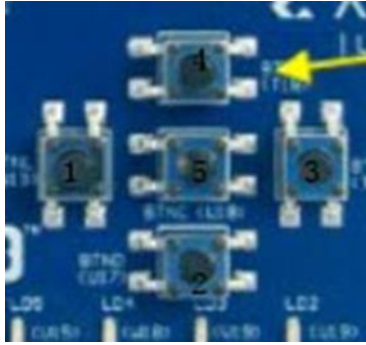
在游戏区域的下方添加一个判断游戏状态的标志，用蓝色发的笑脸代表还可以继续游戏，红色的哭脸代表游戏结束，黄色笑脸代表游戏胜利。并且在游戏结束后，（无论是胜利还是失败），都可以重置游戏状态。



## 三、 可行性分析

### 1、 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；



并且其还有 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。

## 2、 技术可行性

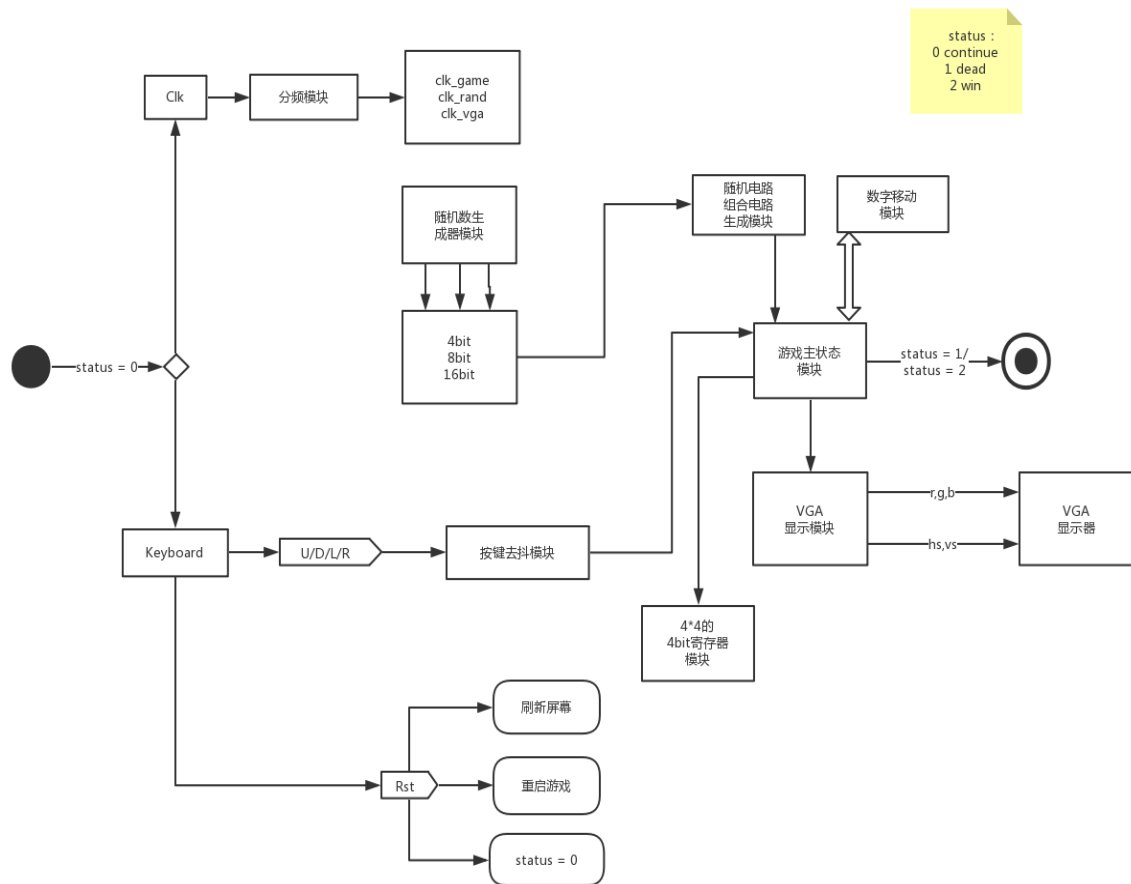
将按键对应的引脚正确分配即可实现按键交互，利用 FIFO 储存结构实现按键除抖；VGA 显示可以通过编程控制颜色与扫描信号实现，同时通过设计不同数字不同的配色，可以实现颜色的区分；与显示游戏方块同理，只要设计好笑脸显示的颜色与位置，就可以显示笑脸，加上一个哨兵变量即可实现根据游戏状态显示不同的笑脸；重置游戏可以通过重置所有变量来实现。

# 四、 实验设计

## 1、 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。

## 2、 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
```

2. 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;
```

3. 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;
```

4. 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;
```
5. 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;
```
6. 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;
```
7. Gen : 在按下按键对应的位置生成一行或一列随机数与三个空位。  

```
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;
```
8. 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;
```

### 3、 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，数字的合并；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，决定每个格子的颜色，即将游戏界面显示在屏幕上；第三个 always 语句处理笑脸随游戏状态的改变，游戏状态有 status 表示，0 代表游戏继续，显示平淡脸；1 代表游戏失败，显示哭脸；2 代表游戏胜利，显示笑脸。
2. vga 模块能够输出屏幕的颜色变量，控制水平和垂直扫描信号来控制屏幕的显示，并且其将当前扫描的位置参数通过 x, y 变量输出到 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，实现按键去抖。



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 2 或 4 或 8 和 3 个空位，该随机数由 random 模块产生。

## 五、 总结及心得体会

### 1、 总结

#### 1) 概述

在本实验项目中，我们实现了一个简简单单的 2048 小游戏，其能够与 Basys 3 板的按键交互，并将结果通过 VGA 扫描显示在屏幕上，有较强的游戏性和趣味性。

#### 2) 问题与改进

在这次的实验进行的过程中，遇到了较多问题，以致于直到实验项目完成时，仍有一些问题的存在，比如：

- a. 在方块中间有莫名其妙的白线，导致方块有些难看，但是没有找到确定的原因。
- b. 不懂 vga 的颜色显示原理，seg 模块中的配色方案是从开源项目中照搬过来的，自己也不会更改那些方块与数字的颜色，只会更改白色和黑色。
- c. 随机数模块写的十分复杂，期待能够在随机的基础上进行改进，使该模块看起来要简练一些。

#### 3) 收获

- a. 知道了在 verilog 语言里宏定义的定义方法与使用方法与 C 语言里的较大区别，也在这上面犯过不少错误，造成了一定的困扰。在 verilog 中，宏定义的格式为：  
``define [name of the macro] [value of the macro]`  
宏使用的格式为：  
``[name of the macro]` 一定不能漏了反引号！
- b. 学会了用对数形式保存 2 的次幂来简化工作，节省空间。
- c. 对于 VGA 的显示原理有了一些了解，看起来复杂，其实只要把握了频率，水平和垂直扫描信号，颜色，就能够在屏幕上显示出内容。
- d. 对于这种复杂的程序设计，我们又一次体会到了功能模块化带来的好处，本程序的八个模块之间除了主模块之外几乎相互独立，极大的方便了我们的调试与团队协作。
- e. 体会了一次游戏开发的大部分流程，得到了满足感与成就感，也提高了自信心，这一次的实验项目为我们今后的学习与实践提供了宝贵经验。

## 2、 心得体会

### ✧ 王锡淮：

在选题的时候，很庆幸我们没有选择当时的另一个备选题目：贪吃蛇，否则我们可能不能够按时完成项目。

在选择了这样一个题目以后，我感受到了大学里面“自学”的真正意义：自由自主、自力更生、自强不息的学习。为什么我这样说呢？因为在学习如何做出这个 2048 游戏的过程中，我的心情是从开心到兴奋，从兴奋到平淡，从平淡到疲倦，从疲倦到逼自己去学习，再到后来的茅塞顿开，恍然大悟，最后坚定地把项目做完了。由此我感受到了大学的魅力，大学的精彩之处。

在做项目的过程中最大的感受就是宏定义的坑太深了，一个不小心就会掉下去，因为在 verilog 里面宏定义的使用偏偏一定要加上一个反引号，最开始的时候，我们是一个反引号都没有加的，于是乎在 vivado 里面综合的时候，发生了奇怪的错误：综合失败了，但是偏偏没有显示错误信息。后来汤柯发现在 status 的信息里面隐含了错误发生的位置，所以我们才发现了这个错误。

还有就是我们在将整个游戏界面放到屏幕中央的时候，反复的计算这个屏幕的大小比例，算了半天才知道原来这种写法是将屏幕 160\*120 等分，最终找到了中央的位置。

然后是关于配色的感想，一开始看到网上那个开源项目的 seg 模块我是崩溃的，完全无法理解那 288 位的数组是在干什么，当然最后逐行分析代码后才发现那是一个一个代表颜色的 bit，虽然还是不知道为什么这几个 bit 会显示红色，那几个 bit 会显示蓝色，但是至少知道了原理，也知道了 3'b000 是黑色，3'b111 是白色。

还要感谢神队友汤柯强大的代码理解能力与准确的直觉为我指明方向。

最后，我想说老师常常提到的那句话：“我都想不到我能做到。”

### ✧ 徐汤柯：

这次的项目本想直接做那个红绿灯的设计，但是看了半天觉得可能少了些器材，所以就直接放弃了。后来我在网上找了很多 Verilog 的题目，发现很多很高端的项目，甚至有人用这个写出了一个小霸王游戏。我和搭档觉得游戏这个想法很不错，便在网找了些 Verilog 能做的游戏，贪吃蛇和 2048。后来找了很多资料，也测试了一些时间，发现并不能完全掌握 VGA 的显示和数字之间的关系，便选择了有显示模块参考的 2048 游戏。在不断测试代码的过程中发现了很多问题，比如显示时是有是有的彩线，按键去抖的方法，显示游戏区域的乱码等问题，我们几乎都是查找资料无果后果断采用最笨的办法，直接修改数据并测试结果，虽然比较耗时间，但最终还是做出来了。在完成这个项目的过程中，学会了很多东西，也对 Verilog 更加熟悉。

## 附录

### 1、源代码：

见 电子档 src 文件夹

### 2、逻辑电路图：

见 电子档 doc 文件夹中的 circuit.pdf

### 3、演示视频：

见 电子档演示视频.mp4

## 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

## 特别鸣谢：

感想王永锋同学借出显示屏供我们进行无穷无尽的调试显示内容。