**华中科技大学**

**《随机模拟与实验》实验报告**

|  |  |
| --- | --- |
| 实验名称： | 基于FPGA的节奏大师游戏 |
| 院（系）： | 电子信息与通信学院 |
| 专业班级： | 信卓1501 |
| 姓名： | 林俊宏、罗天琦 |
| 学号： | U201513247、U201510168 |
| 指导教师： | 杨明 |

2017 年 6 月 1 日

# 实验目的

1. 探究使用FPGA构建小型数字系统。
2. 进一步掌握EDA工具的深入使用。

# 实验步骤

1、通过EDA工具构建系统硬件部分。

1. 写入相应的SDK程序。
2. 烧录FPGA芯片，写入程序，运行。

# 实验工具

Nexys4 DDR开发板，VGA显示屏，USB键盘，Vivado HLS 2017.1开发环境，Xilinx 2017.1 SDK 开发环境。

# 实验原理

制作一个能方便实现的游戏，我们的第一想法就是通过硬件搭建一个可供游戏运行的嵌入式系统，在上面运行我们写好的游戏程序。此方法易于产品的迭代，硬件系统以及游戏软件都可以分别进一步优化升级。

## 硬件部分

### 最小系统的搭建

硬件部分我们通过使用Xilinx公司提供的MicroBlaze软核，搭建一个100MHz主频的32位处理器，并加入一系列的外设，以满足游戏运行的需求。

Vivado HLS 2017.1中软核的搭建提供了十分直观方便的以电路原理图相连接的Block Design框图，我们根据系统需求依次设置MicroBlaze的配置参数，和各IP之间系统总线带宽等参数，就可以构建出一个最小系统。

如下图所示，是搭建软核过程中的Block Design总系统框图。

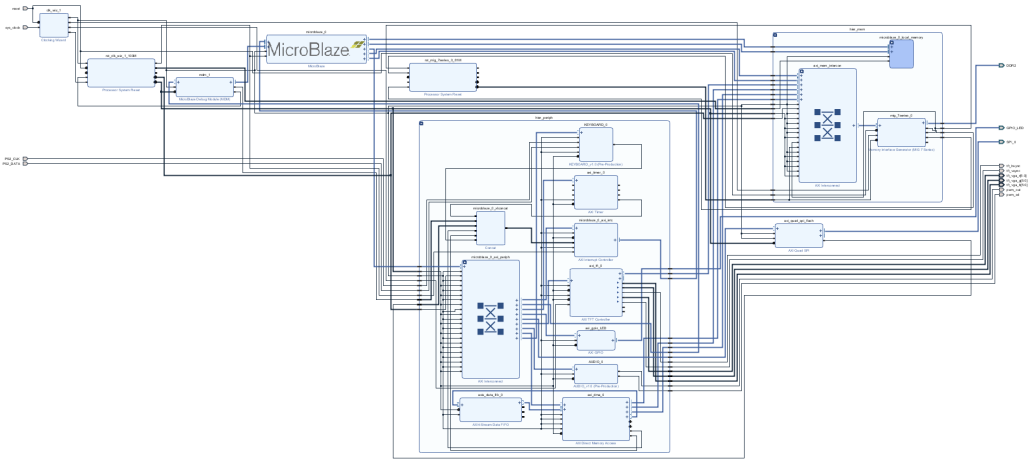


图 1. Block Design系统框图

系统中包含如下外设：

* + - Memory Interface Generator（MIG）

生成SDRAM控制接口，作为128M外部DDR2 SDRAM与MicroBlaze处理器数据交互的桥梁。

* + - AXI Interrupt Controller

中断控制器，各种外设的中断通过此控制器送往MicroBlaze的中断接口。

* + - AXI Timer

定时器，用于产生定时中断用于绘图。

* + - AXI TFT Controller

显示控制器，用于产生VGA输出信号，只需改变显存中的内容就能改变显示的图像。

* + - AXI Direct Memory Access

DMA控制器，可以不受处理器调控，自动地根据设置将数据在内存中搬运。用于图像显示的硬件加速。

* + - AXI4-Stream Data FIFO

先入先出存储器，用于DMA控制器数据的中转。

* + - KEYBOARD\_V1.0

自定义的PS/2键盘数据收发IP。

* + - AUDIO\_V1.0

自定义的音乐播放IP。

由于系统框图太过复杂，我们将其中的外设组合显示，可以得出一个系统结构更加清晰的系统框图。

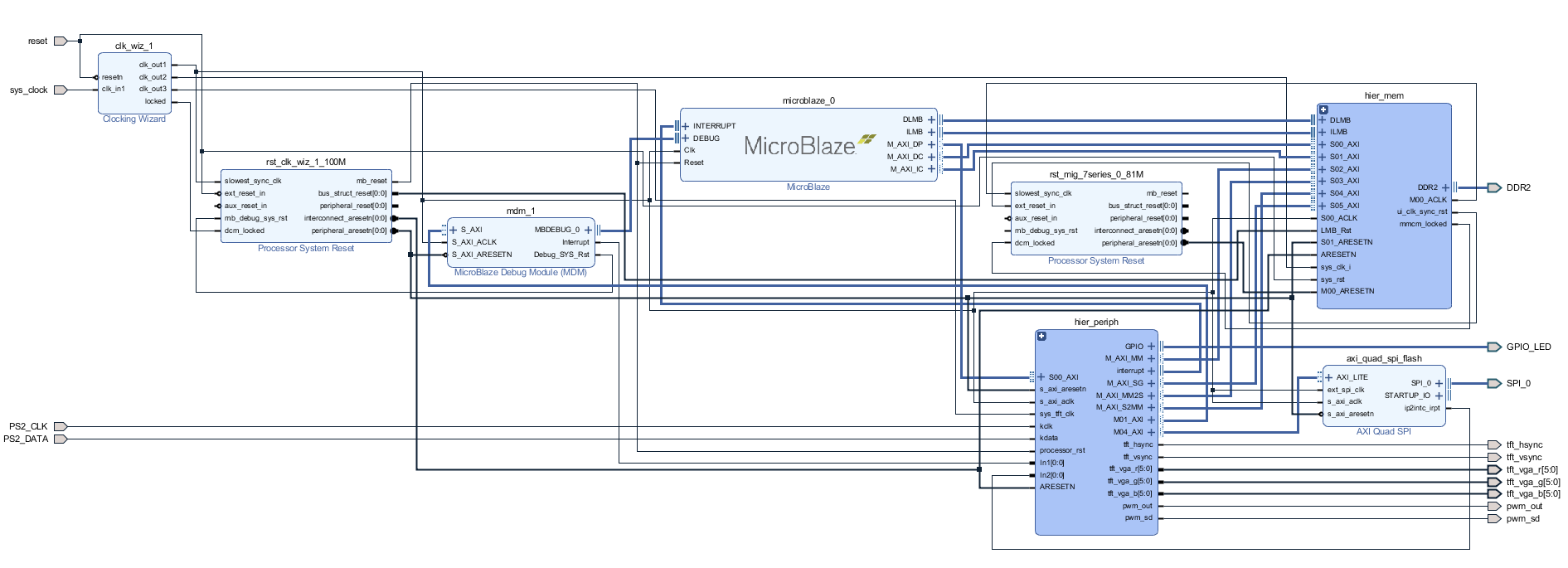


图 2. 化简后的系统框图

### 生成自定义IP

实验过程中因音频生成和PS/2键盘数据收发均没有现成可用的IP，故需由我们自行定义封装新的IP在框图中使用。

先在Vivado HLS 2017.1开发环境中，选择Tools菜单中的Create and Package New IP选项，并在新对话框中选择生成AXI4的IP，并键入用户IP核的名字，就可以点击确认，生成用户IP。

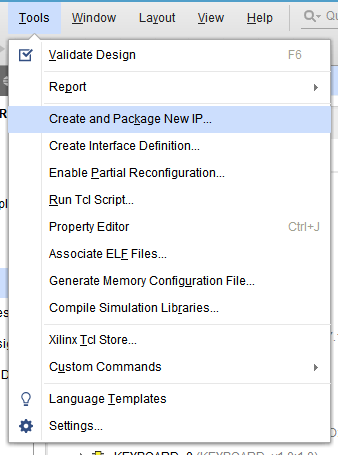


图 3. 选择生成新的用户IP

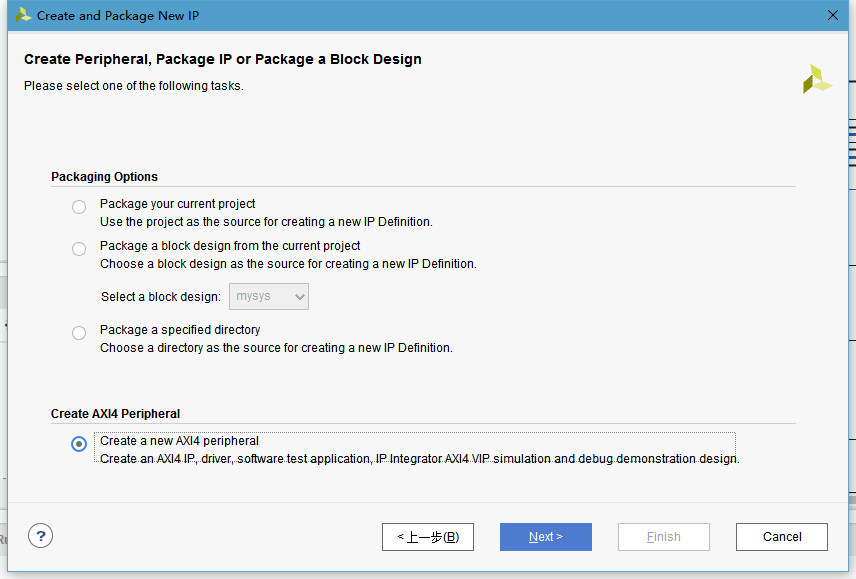


图 4. 选择生成AXI4接口的IP

新生成的IP会在IP Catalog中展示出来，我们需要现在就对这个IP核进行Verilog代码的编辑，以使它能满足我们接发数据的需求。右击KEYBOARD\_V1.0，并在菜单中选择Edit in IP Packager，打开新的Vivado界面。这个界面和原来的Vivado界面有几分相像，只是多了一个Package IP的标签页。

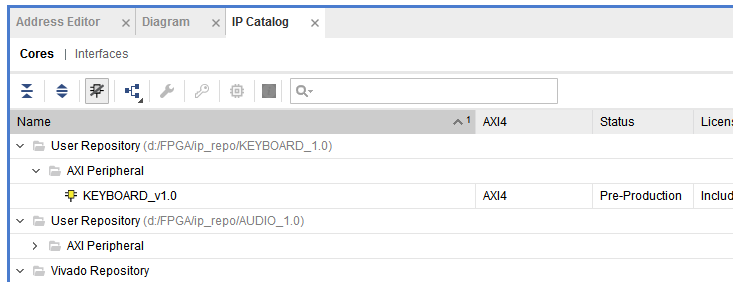


图 5. IP Catalog中新生成的IP

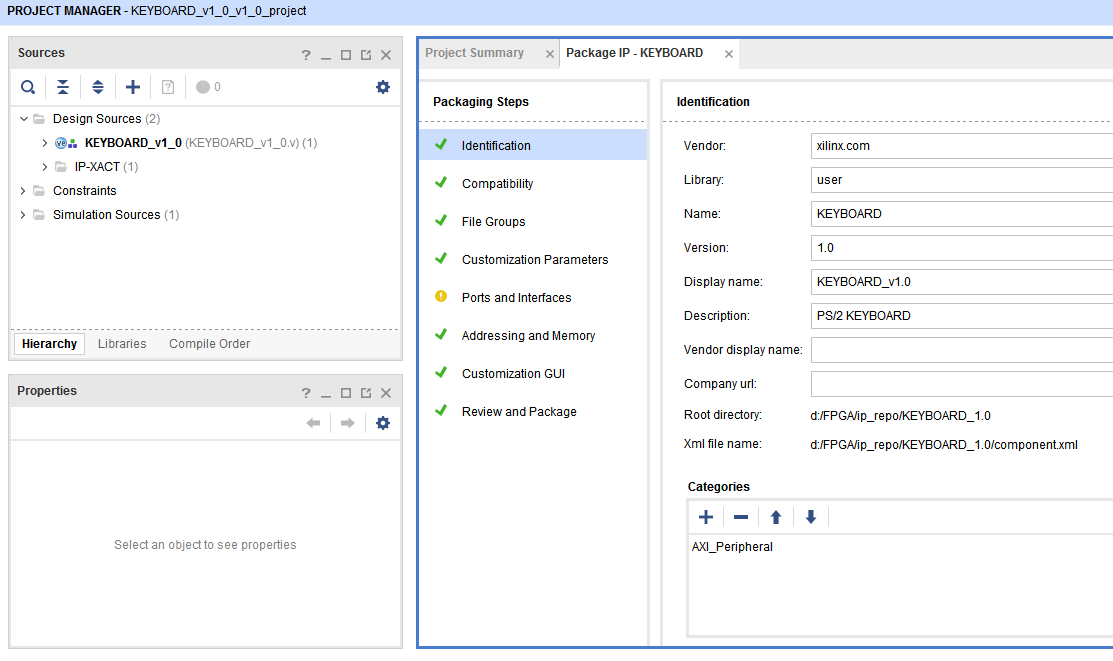


图 6. IP Packager界面

在左侧的资源结构树中打开底层Verilog文件，添加对键盘数据接收的底层代码，并在新建IP核时自动生成的AXI4通信协议的Verilog文件中添加相应的改写slv\_reg0的代码，方便我们的处理器能通过KEYBOARD外设的地址直接读得键盘按下的键值。添加完成后，在Package IP标签页上依次更新文件并重新封装IP，就可以退出界面，完成用户IP的自定义了。

### 逻辑综合和生成比特流文件

回到最初的Vivado界面，点击系统框图上方的Validate Design，可以让系统将框图检查一遍是否有问题，并且生效该Block Design。当弹出一切正常的对话框，我们就可以生成HDL Wrapper，并自己建立新的Verilog文件，调用已经例化好的最小系统模块了。此处可以添加新的用户逻辑，但本系统目前不需要添加新的逻辑，我们只添加了一系列端口，并添加约束，就可以生成比特流文件了。

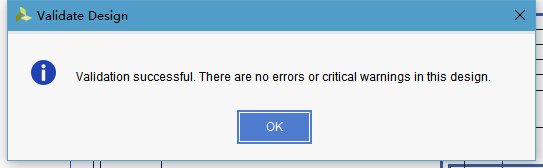


图 7. 生效框图成功

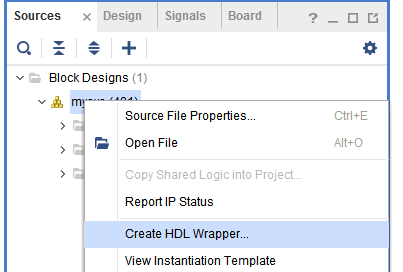


图 8. 生成HDL Wrapper

由于本次综合的系统规模相对较大，大概需要30~40分钟可以完成整个综合流程。在综合完毕以后，可以查看综合完毕的结果和报告。从内部资源占用图可以看出本系统只使用了整个芯片一半的资源，这意味着，我们甚至可以在Nexys4 DDR的板子上构建一个双核系统。

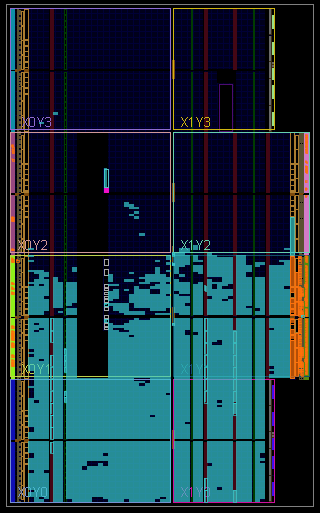


图 9. 内部资源占用图

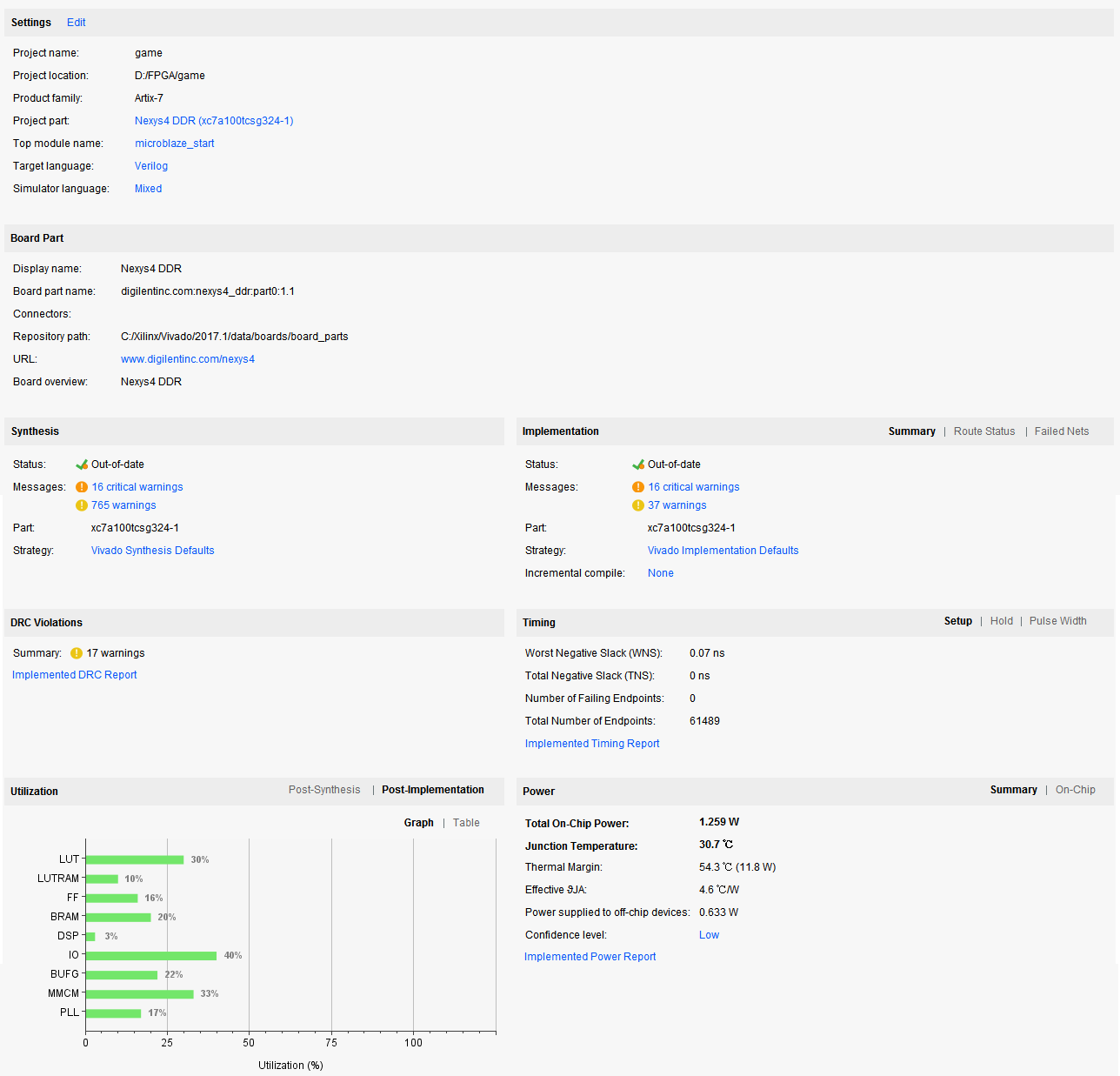


图 10. 最终生成的系统报告

在报告中有详细的各种资源的占用情况，还有系统具体的运行功耗。实际烧录比特流文件后，发现芯片温升比较高，通过USB电流电压表监测功耗与报告中描述非常接近，但是报告中温升估计误差较大，所以如果能加上散热片工作，系统会的性能竟会有更好的提升。

## 软件部分

软件部分我们完成了一个节奏大师的游戏，这个游戏的主要内容为点击随音乐生成的滑块，在滑块到达屏幕下方时刚好击中则得分

### 音乐预处理

首先需要根据选择的音乐生成一段表示滑块顺序的数组，通过控制滑块生成的频率和速度调整游戏难度。为了更好的游戏体验，击中滑块的节奏应和音乐的重音对应。为此我先对音乐信号进行处理，将重音位置提取出来，然后据此生成滑块数组。

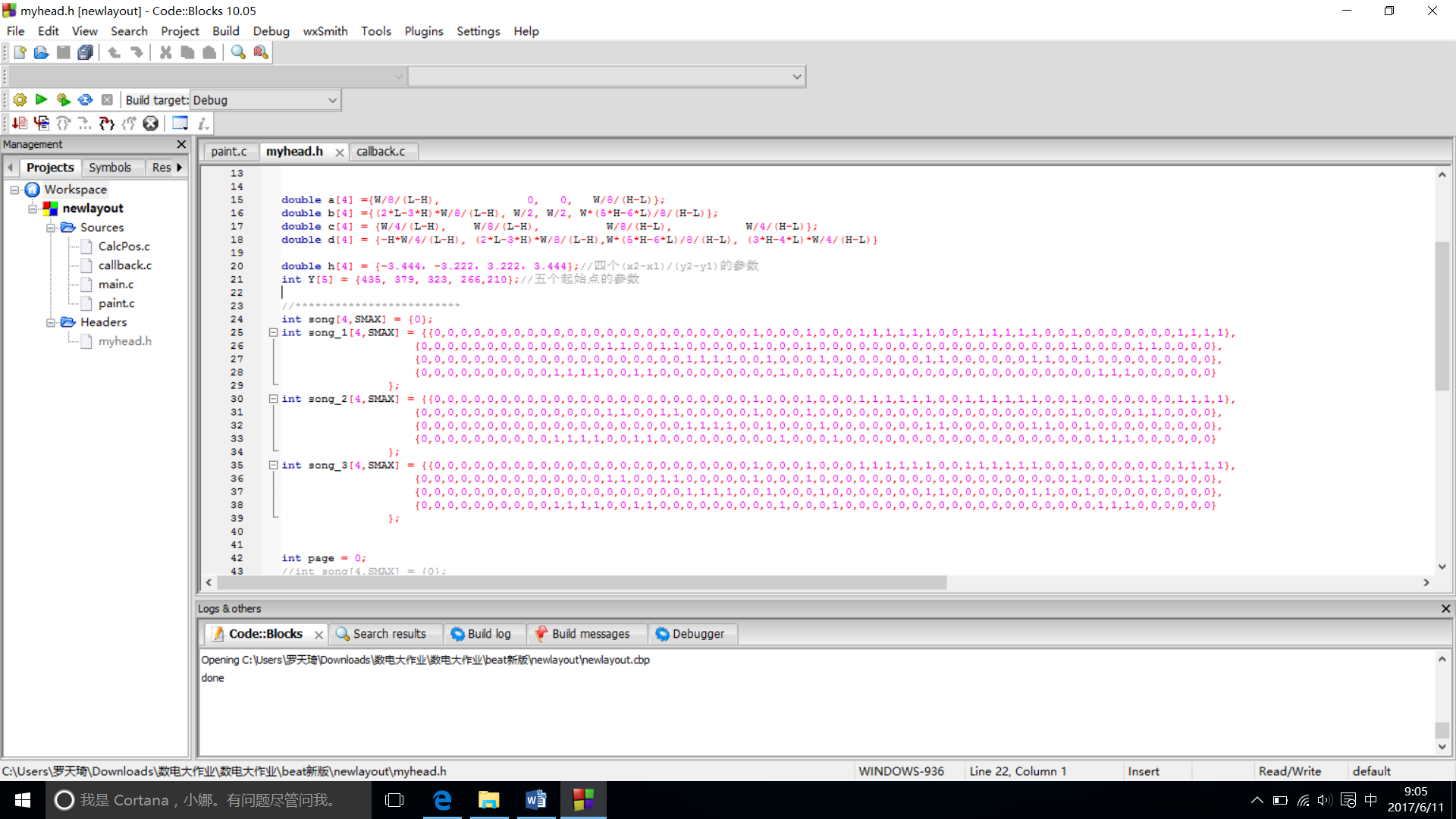


图 11. 滑块数组

### 游戏界面

在设计中，我把游戏分为了三个部分:选歌界面，游戏界面，得分界面。三个界面的切换和所有交互功能由键盘完成。刷新时，用一个page变量记录当前所处界面。如果此界面是进入后第一次刷新，则绘制背景图。

**选歌界面：**选择游戏难度，选好后进入游戏。

**游戏界面:** 点击随音乐生成的滑块，在滑块到达屏幕下方时刚好击中则得分。音乐停止时游戏结束进入分数界面。

**分数界面：**显示游戏得分，你可以选择回到游戏界面再来一次，或者回到选歌界面换一首歌。

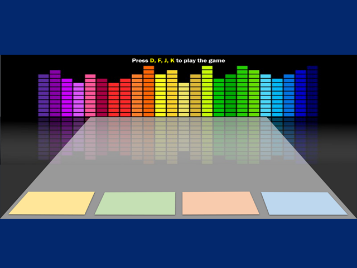


图 12. 三个游戏界面

### 键盘交互

由于我们所接的外设为VGA屏幕和USB键盘，所有交互由键盘完成，操作指南写在每一个界面的上方或者下方

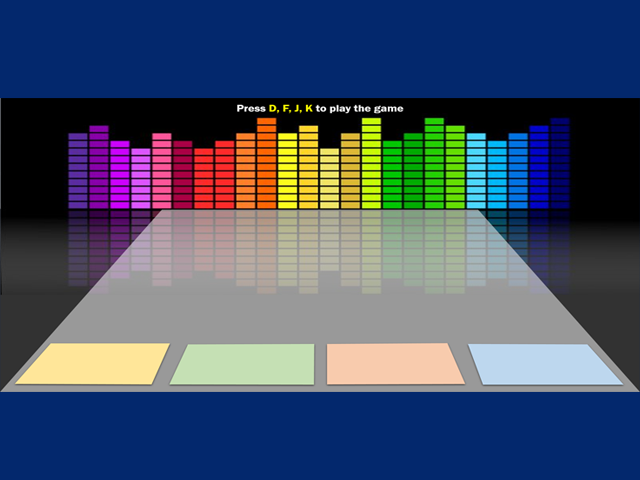
**选歌界面：**使用Up和down选择歌曲，选好后，使用Enter键进入游戏。

**游戏界面:** 使用D,F,J,K四个键控制点击滑块得分。记录点击时间和滑块运动到底部的时间进行对比，判断是否击中。

**分数界面：**点击R回到游戏界面重新开始游戏，点击S回到选歌界面重新选择难度。

### 游戏难点：滑块的运动与生成

这个游戏的设计中，最重要的部分就是滑块如何运动。由于滑道有近大远小的透视效果，所以滑块也随着滑动不断变大。为此我们使用了一个简单的计算公式：



(0,0)

(x1,y1)

(x2,y2)

Y=cx+d

Y=ax+b

图 13. 屏幕坐标系及其计算

如图，整个背景图的左上角为原点，以绿色滑块的轨道为例，假设轨道右边直线的方程为y=ax+b，左边直线的方程为y=cx+d。又由于滑块贴图本身为矩形，所以滑块矩形的右上角（x1,y1）和左下角(x2,y2)落在直线上。

则有方程组： ,再加上矩形比例为常数：

经过变换，解得： 。

因此，可以由(x1,y1)的坐标得到(x2,y2)的坐标，完成滑块沿着轨道运动的透视效果。

至于滑块的生成，虽然轨道看起来是连续的，但实际上每个轨道都被分成四段（如上图），滑块一开始出现在最远的一段，当它运动到这一段底部时进入下一个小段继续滑动。这样设计是为了让屏幕上每个滑道最多只同时出现四个滑块，便于对滑块进行操作。

# 录制视频

、 在作品制作完成后，我们录制了相应的操作和讲解视频，并将剪辑后的视频上传至了百度云盘。

**链接：**http://pan.baidu.com/s/1eR2W7aa

**密码：**epwx

# 总结

## 小组内部分工

林俊宏：负责整体系统框架的搭建，SDK底层代码的封装。

罗天琦：负责SDK程序的顶层编写开发。

## 小结

1. 该课程设计所设计的各种模块和原理与数字电路及FPGA的理论知识紧密结合。所以这样与课本知识关联紧的课程实际比较容易完成，同时不仅加强了对书本基础知识的理解，而且实践环节得到了很好的锻炼，能够激发实践者对随机的兴趣甚至能够自己动手完成一些课题项目。
2. 深入学习了EDA软件的更多用途，可以构建更加复杂的小型系统，极大地提高了对FPGA学习的兴趣。
3. 课程设计的多样化可以开发思维，鼓励创新，同时也检验了自己对所学知识的牢固性，没有扎实的理论知识基础是不可能有自己的设计方案的，所以我们以后应该学好理论知识，才能创造出我们自己感兴趣的设计和作品。