**Vga小游戏实验报告**

515030910211 姜子悦

1. 实验目的
2. 理解vga接口的原理，学会使用verilog控制vga显示
3. 感受并行编程编写游戏和面向对象编程的区别
4. 实验设计

目的是做出一个vga显示的flippy bird

输入：

clk： 50mhz时钟

sw9、sw8：控制每次跳跃高度、自然下降高度

key3: 控制小人跳跃

key0: reset

输出：

vga所需的25mhz clk、blank\_n（控制是否在画面外）、rgb（每色8个bit）、hs、vs（行、场同步信号）

hex0、1: 当前游戏得分

hex5、6：历史最高分

1. 实验步骤

第一步：尝试连接vga屏幕，在经历痛苦调试后，能在屏幕上显示设定好的静止图片。其中需要注意的是vga信号的产生、根据网上的教程设置了各个参数，对于hs、vs等的计算需要特别仔细，因为不仅是表面看到的640\*480的分辨率，还有front magin、同步时间等周期也需要考虑进去。

第二步：在屏幕上绘制了一个小人，这一步主要是设计了一个显示对象需要哪些参数来标记。

第三步：为小人增加跳跃功能和自然下降功能，这一步并不难，值得注意的是为了使跳跃的像素数和下降像素数合理，调试了较长时间。最终下降频率选择了5hz，这样既不会下降的太快，画面连贯性也比较好。选择50hz的画下降太快，而2hz又能明显感受到卡顿。

第四步：绘制静止障碍物，障碍物拥有所处横坐标、宽度、空缺所在坐标、高度等参数。

第五步：使障碍物能够移动，当一个障碍物移出屏幕时，将它放到障碍物队列的最后一个，等待下一次出现。

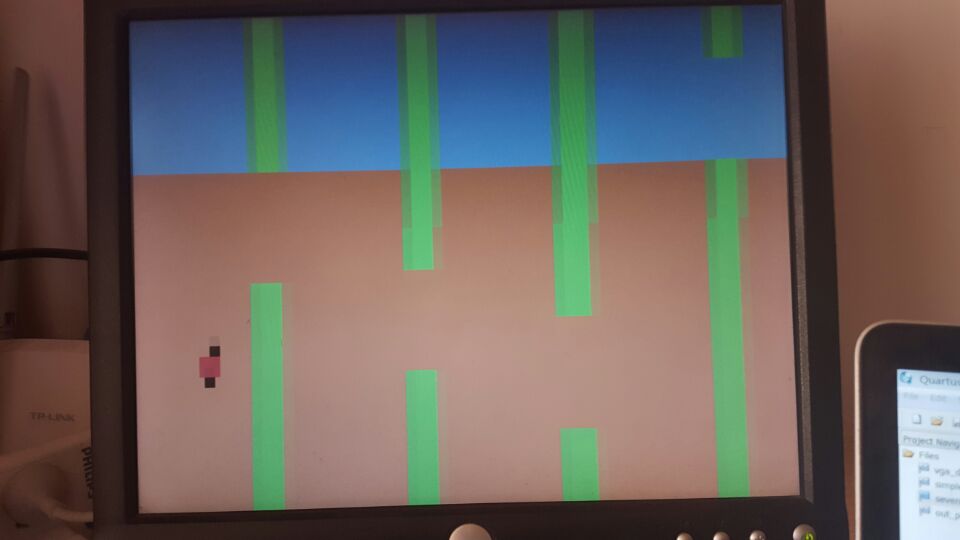
第六步：增加game over判定，当小人离开屏幕或者碰到障碍物使，游戏结束

第七步：增加得分与历史最高分判定，输出在hex上

第八步：增加挂档操作，可以在一定范围内控制每次up距离和down速度，使游戏更具操作性

第九步：尝试添加随机函数使障碍物能随机出现

1. 实验结果与总结



游戏画面

总体来说，是我想要达成的效果。嗯，其实做的比较简单，原来想做个卡牌游戏的，但是发现用fpga并行编写考虑起来相当困难，vga的显示写起来也觉得挺奇怪的，和平时用面向对线编程的时候感觉很不一样。

所以就写了一个flippy bird，显示不难又有一些可玩性的游戏。

遗憾的事情是没有做出文字显示，比如把分数显示在屏幕上而不是板子的hex上，还有没有能添加声效——看官网上的教程没有看懂。