# 可编程逻辑器件大作业报告

硕8052 高育哲 3118105034

## 作业内容

基于DE1-SOC开发板，使用Verilog语言开发了一个坦克大战小游戏。以下为该小游戏的特点解析：

* 玩法为人机对战的生存模式：敌方坦克无限复活，我方坦克只有10条生命。生命用完后，将结束游戏，显示ROM中存储的GAME OVER图像。
* 游戏中一共有一辆我方坦克和两辆敌方坦克（可以通过实例化模块增加）。
* 我方坦克的生命数通过开发板上的LED灯显示。复位后，LED0~9全亮，此后每损失一条命就有个LED灯熄灭。我方坦克每次被击中时，都会闪烁一次，并损失一条生命。
* 敌方坦克被击中后，会在2秒后在固定位置复活。复活期间会有动画，并无法移动且不能被击中。
* 游戏期间击中的敌方坦克数量作为最终的游戏分数。游戏分数通过七段数码显示管显示，显示范围为0~9999。
* 敌方坦克的运动和发射子弹由伪随机数产生模块控制。
* 通过四个防抖动的按键KEY0~4控制我方坦克的运动，通过SW9(slide switc)控制我方坦克的子弹发射。
* 来自不同坦克的子弹相遇时会碰撞消失。
* 任何坦克的前进路径上如果有其他坦克，会被卡住无法前进。
* 使用VGA输出640 x 480 @ 60 Hz的图像。
* 通过SW1控制是否复位。
* 通过SW2控制是否暂停游戏。

## 设计背景和设计难点

Tankwar是一个世界知名的小游戏，承载着许多人的童年回忆。因此我想用硬件描述语言来最大程度的复现这个小游戏。

在前期的调研过程中，我以tankwar为关键词在Github上搜索以往相关的项目，发现大多数项目均是以c/c++来实现，极少有用硬件描述语言来实现。仅有的几个tankwar的verilog/vhdl项目，均为双人对战模式，玩法与我预想中的玩法不同，因此参考价值有限。本作业并未参考他们的代码，全部由我亲自完成。

我认为这个项目的主要设计难点在于实体(entity)太多。子弹和坦克都作为一个独立的实体。如果游戏中一共有三辆坦克，每辆坦克最多同时存在4枚子弹，那么就需要去追踪15个实体的运动方向、运动速度和位置坐标，以及它们是否相撞，这会导致一个很大的判断逻辑模块。

如果用大量的if-else模块，可能会导致组合逻辑的输出延时过大，导致下一级DFF的建立时间不再满足。但是受VGA显示的原理所限，当前时刻必须给出确定的RGB信号，不可能利用流水线的方式拆分组合逻辑。因此如果if-else的方式不满足时序要求的话，必须要考虑组合逻辑的并行运行，然后用DFF去采样。但是这样的设计也会提升设计的复杂性，而且可能会因为毛刺被采样而带来一些其他问题，因此我优先考虑if-else的设计方法。综合后上板测试的结果显示，这样的设计方法满足了时序要求。

因此设计的难点可以总结为以下几点：

1. 坦克和子弹的移动速度和移动方向追踪以及碰撞检测
2. 敌我子弹的区别，只有被敌对阵营的坦克子弹击中才算死亡。
3. VGA显示的问题。这个问题反而是最让我头疼的问题，网上有无数的文章讲解了VGA的原理，也有很多现成的代码。我将网上的示例代码在板上运行（已经额外给DAC的三个信号赋值），但无一例外的都是黑屏，即使是最简单的彩条显示。后来我发现只要让VGA显示时序段的x=0以及y=0处的RGB={0,0,0}，其他的同步脉冲信号以及显示前后沿的时间间隔都没有任何变化，所有坐标点的输出就完全正常了。按理说一个坐标点的RGB输出不应该对其他坐标点的输出产生影响。我怀疑是开发板的问题，因为换了一个显示器也是这样的。我在intel FPGA的论坛上询问，并贴上了简短的示例代码以及实测图，但是并没有什么有价值的回复。

## 游戏逻辑流程图



Figure 1 游戏逻辑流程图

## 设计实现方法



Figure 2 系统模块关系图

Tank\_i:输入tank\_exist,tank\_direction，输出tank\_x,tank\_y,，实现坐标计算功能。

Bullet\_ij:输入bullet\_exist,bullet\_direction，输出bullet\_x,bullet\_y，实现坐标计算功能。

PLL时钟：输入为DE1SOC开发板上晶振提供的50Mhz的时钟，输出为VGA显示需要的25Mhz。

ROM：存储了GAME OVER的图像。输入为addr和clk，输出RGB信号。

VGA显示：输出VGA显示所需的RGB信息、hsync、vsync以及DAC所需的dac\_blank、dac\_sync、dac\_clk

逻辑控制模块：根据各个坦克和子弹的坐标信息，判断他们是否相撞，实现游戏逻辑流程图所示的功能，

七段数码显示管：提供七段数码显示管和LED灯的信号

顶层模块：调用下面所有模块，分配引脚