VGA 拼图游戏

命题人：彭睿杰、谭雍昊、易辰朗

整体描述：

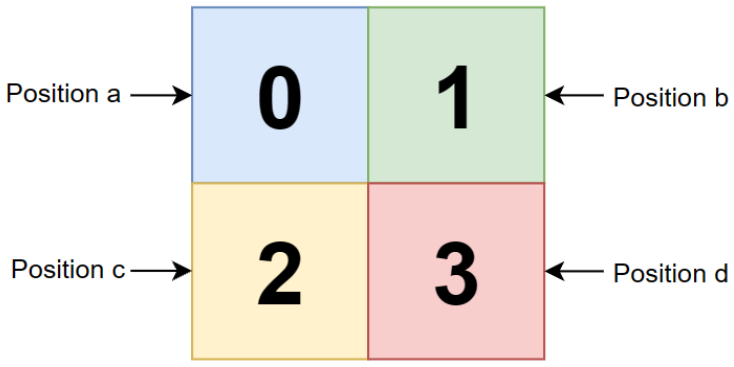
利用 minisys 开发板上的 VGA 接口做一个开发板与显示屏的交互，达成拼图游戏的效果。首先开发者需要事先加载进开发板一幅图像。在游戏的初始状态时，显示屏可以显示一幅静态图像；一个置位按钮可以使图像分块，打乱图片中各区块的位置，分块规则将根据五个开关确定；一个游戏启动的开关可以开始游戏，产生黑块；游戏开始后，可以通过四个按键分别控制黑块上下左右移动。当游戏进行到图像块等于初始化时的图像顺序时，代表游戏通关结束。

基本功能:

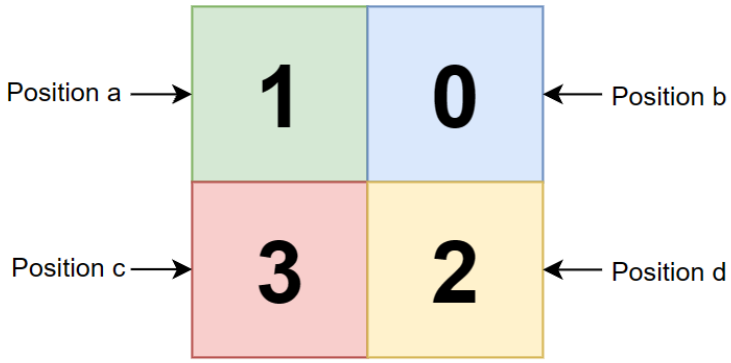
1. 读图：通过 *Block Memory* IP 核读取一副静态图像（利用脚本将.png 文件的像素矩阵转为.coe 文件输入进 IP 核，转化脚本链接我们提供在附件），编写 VGA 驱动使静态图片显示在显示屏上。**（30 分）**

2. 分块（初始置位）：将图像分为 2x2 个区块，通过五个开关（KEY0-KEY4）来控制不同图像区块的排列组合，并设置一个置位按键 (BUT0)，按一下可使其置位到游戏的初始状态。**（20 分）**

举例：假如下图（a）是一副静态图像，现在要对这幅图像进行分块操作。我们假设这个 2x2 区域对应的区块位置分别是 position a, position b, position c, position d. 当正常的显示一副静态图像的话，position a、b、c、d 应该分别对应像素区块 0、像素区块 1、像素区块 2、像素区块 3。而现在我们要实现使这张图片乱序，一个乱序的例子如图（b）。现在要通过 5 个开关来控制不同图像区块的排列组合，首先，我们要考虑清楚一共有几种排列组合的情况（4*3*2*1=24 种），所以我们可以通过五个开关（即 5-bit）去控制不同的排列组合情况，设 5'b00000-5'b10111 为有效位（即对应 24 种排列组合情况），其中 5'b00000 对应图（a），其余 23 种情况可自行配对，只要保证每种情况不重复即可，5'b11000-5'b11111 为无效位。当按下置位按键 BUT0、且五个开关为有效位时，显示屏会根据五个开关的状态显示对应的像素区块情况。如果五个开关为无效位，显示屏将保持原来状态。



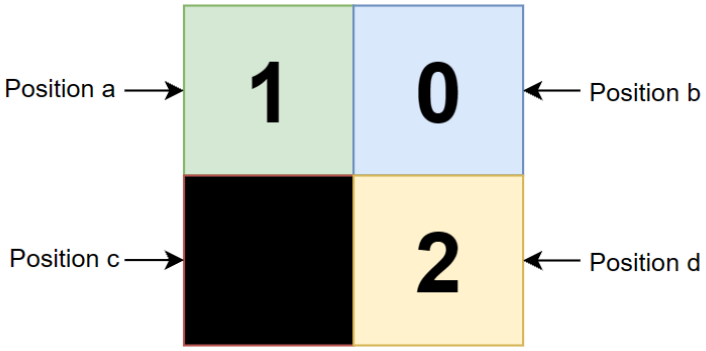
(a)



(b)

3. 挖洞：设置一个开关 KEY5，拉高使像素区块 3 变成黑块（即像素值全部清 0）（无论像素区块 3 位于哪个 position），并代表游戏开始。开关拉低时表示游戏中断，黑块变成原来的像素块。**（10 分）**

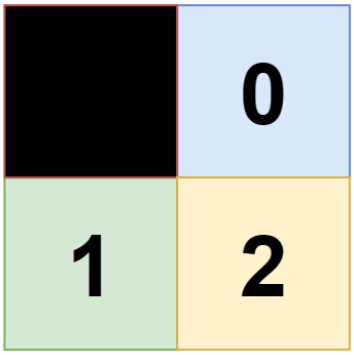
举例：如果在图（b）的情况下拉高 KEY5，则显示屏显示的图像应该是图（c）。



(c)

4. 移动：在 KEY5 拉高（游戏开始）的前提下，通过 4 个按键（BUT1,BUT2,BUT3,BUT4）控制黑块上下左右移动。（要注意边界情况，比如说图（c）中黑块只能上移或右移，此时左移和下移触发应该无反应。） **（10 分）**

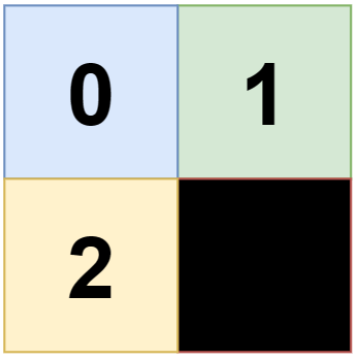
举例：假如在图（c）情况下，KEY5 保持拉高状态，并按下上移按键，结果应为图（d）



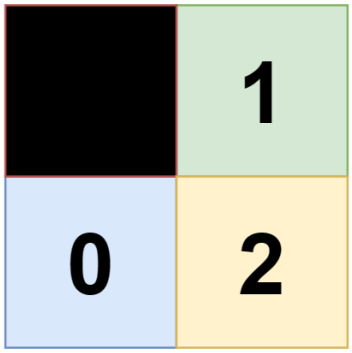
(d)

5. 通关：当 KEY5 持续拉高的情况下，通过移动最后达到了如图 (e) 所示的图像 (即正序)，则在开发板上执行 LED 流水灯，表示游戏通关。(可能会出现无解的情况，例如从图 (d)

就无法到图（e），有兴趣优化无解情况的同学可以在 bonus 部分研究，展示时第 5 条我们直接置位到有解情况测试。） 一种有解的情况如图（f）。 **（10 分）**



（e）



（f）

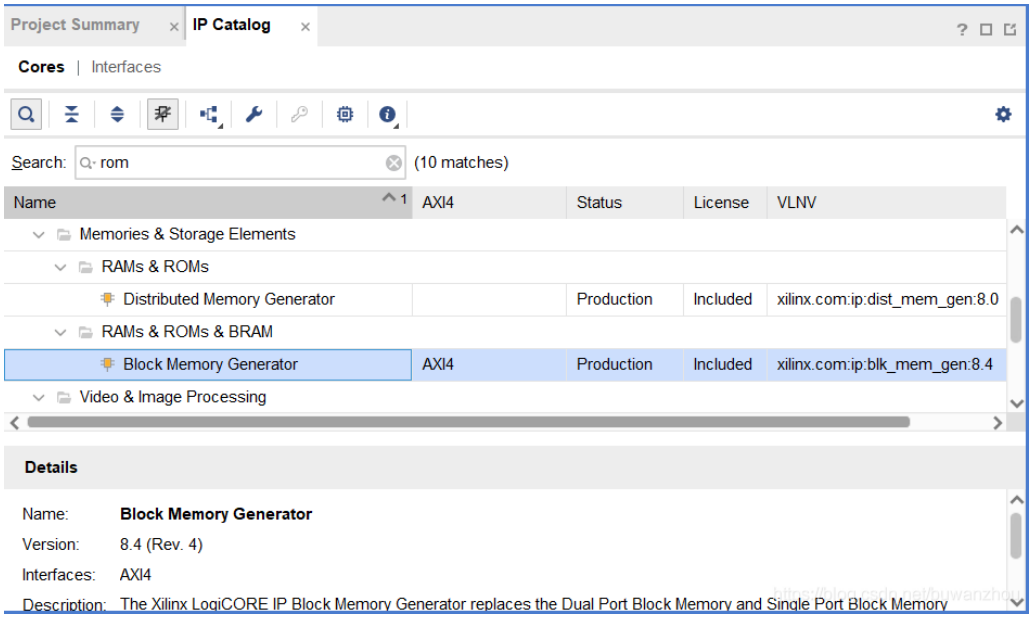
Bonus：（20 分）

1. 利用串口读图代替 Block memory IP 核读图
2. 按钮实现防抖功能（若两次按钮触发时间间隔过短，应当视作只按一次）
3. **按一个按钮可以实现随机置位（随机置位到 24 种状态的任意一种）**
4. 加入计步器与计时器，显示在七段数码管上
5. 将区块拓展为 n*n，甚至 n*m （n>2）
6. **利用算法提前判定是否有解**

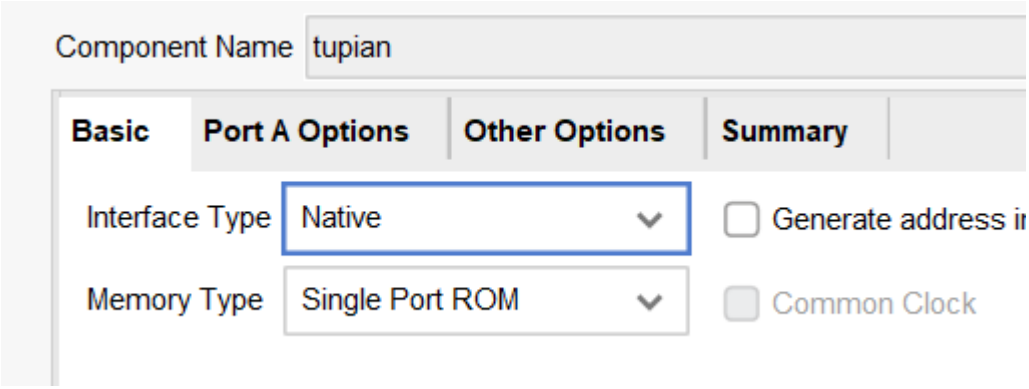
Reference:

1. VGA 显示原理（通俗易懂）：https://blog.csdn.net/shichao1470/article/details/81840978
2. 图片转 .coe 文件转化脚本: https://www.icfedu.cn/archives/10430
3. Block memory IP 核的使用：

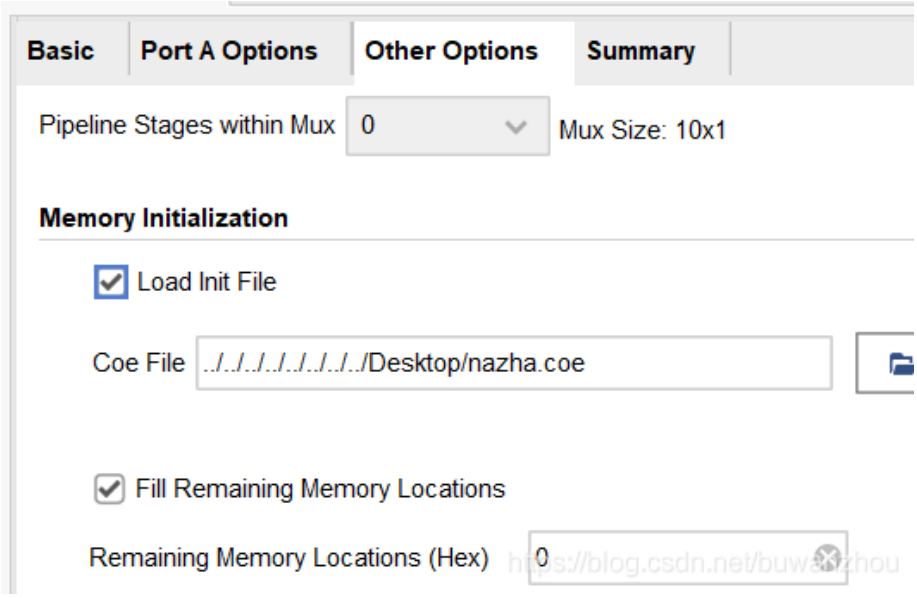
（1）创建 IP 核



（2）更改模块名，选择 Single Port ROM

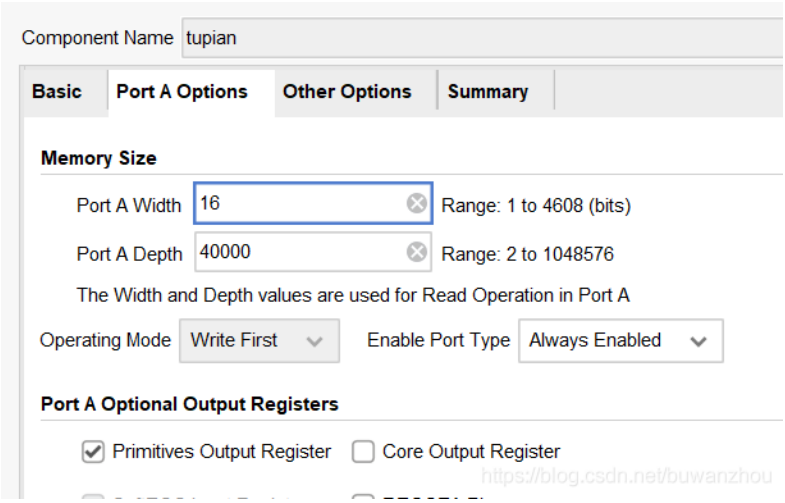


（3）选择生成的.coe 图像数据文件



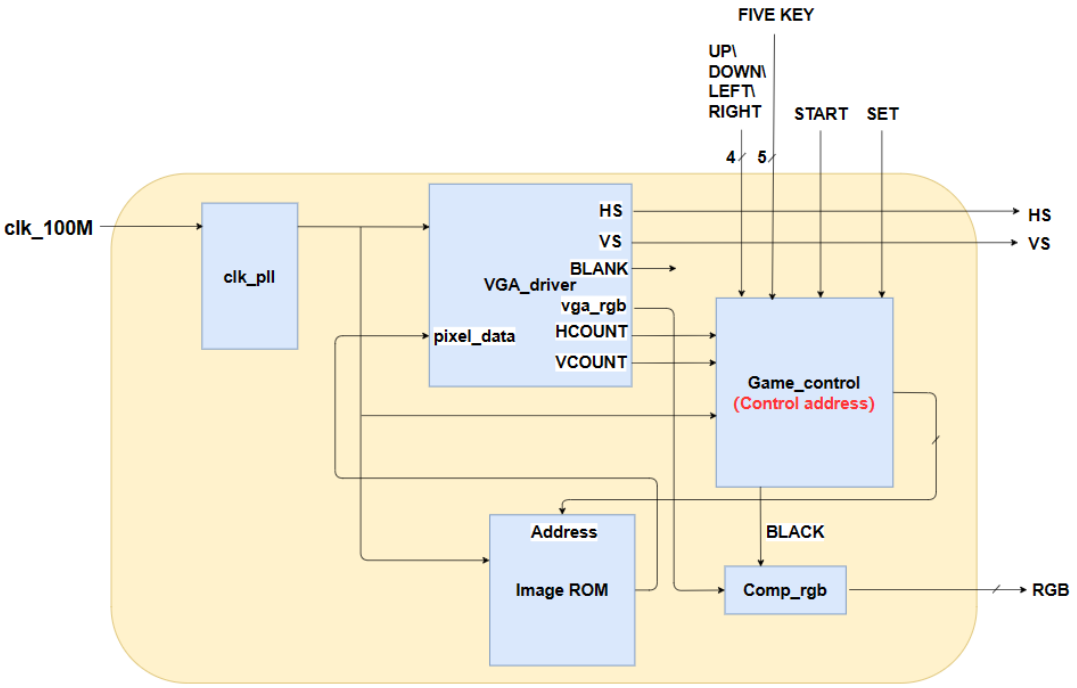
（4）根据 coe 文件的 RGB 位数和大小选择。

例如：此处的图片格式为 RGB565，为 5+6+5 = 16 位，图片 200*200，故 Depth 为 40000。



（5）生成 IP 核，在顶层模块中实例化此 IP 核即可在模块中读取图像数据。

4. 一种可参考的基本逻辑架构图



clk_pll 为时钟分频器，可输出你想要的时钟频率（<100M Hz），可调用 IP 核使用。
VGA_driver 为 VGA 驱动模块，主要是开发板与显示屏的交互（参考 VGA 通信协议）。
Game_Control 是这个项目的核心模块，需要通过前面描述的各项输入逻辑来控制 Image ROM 此刻所要输出数据的内存地址（例如，对于一幅 200*200 的图像，内存地址就为 0-39999）。（HCOUNT 和 VCOUNT 信号控制的是显示区域，即上图所示的 position a、b、c、d，而 Address 的信号控制的是输出的像素区块，即上图所示的 0、1、2、3.）
Image ROM 可以是 Reference3 提到的 Block Memory IP 核，主要用于存储图像数据。
Comp_rgb 为一简单组合逻辑模块，用于判定输出像素值还是黑块。

整个架构中 clk_100M 为 FPGA 开发板时钟频率，在 minisys 开发板中由 Y18 管脚提供（可参考 minisys 开发板用户手册）。UP/DOWN/LEFT/RIGHT 为四个移动按键（对应基本功能中的第 4 点），FIVE KEY 为置位状态开关（对应基本功能中的第 2 点），START 为游戏开始开关（对应基本功能中的第 3 点），SET 为置位按钮（对应基本功能中的第 2 点）。HS、VS 为 VGA 行同步信号与场同步信号，RGB 为输出像素，需要根据 VGA 通信协议映射到 VGA 的管脚中。

注意：我们使用的 minisys 开发板的 VGA 接口只支持 RGB444（Red-4 位，Green-4 位，Blue-4 位）的格式，所以在转化成.coe 文件时需要点 RGB444 的选项，block memory 位宽应选择 12 位！