**Require**

ZYBO which is provided by Xilinx.

vivado 2016.2.

VGA cable and micro\_usb cable.

**How the code is organized:**

The module vgacontroller(Space\_incaders.v) should be set as the top module, and this module has instanced those necessary sub\_modules such as debounce, spaceship, rocket, ect.

***module vgacontroller:***

There are rst, clk, btnR, btnL, fire, startbtn as inputs.

I set rst port to G15(which has been set in constraints.xdc);

clk is set to L16 which is the clock generated by the crystal oscillator of ZYBO board;

btnL is set to Y16, and this signal is used to control the spaceship to move towards left, btnL is also the input of module spaceship. When spaceship(the x coordinate of spaceship) hasn’t reached the left side(spaceship is bigger than 11), the spaceship will move 1 pix per clk.

btnR is set to V16, and the function of btnR is the same as btnL, the only difference is btnR controls spaceship to move right.

fire is set to P16, which is used to control the spaceship to fire.

And the main function of this module is to control the VGA output, you can find the VGA protocol here:

<https://en.wikipedia.org/wiki/Video_Graphics_Array>

This module also defines the mechanism of how the aliens move, how the aliens fire, where the board is, when to display the results pictures, etc. If you are interested in creating your own game mode or level of difficulty, you can change this module to whatever you like.

***module horizontal:***

This module is based on the VGA protocol, and has decided how is each row scanned, this is decided by which form of VGA output you use, you should change the clock frequency in order to meet the requirement of VGA protocol.

***module vertical:***

This module is very similar as the horizontal module, the only difference is that vertical is used to control field scanning, and the horizontal module is used to control row scanning.

***module debounce:***

This module is used to generate the suitable btnL, btnR, fire signal. And you can change the speed and fire speed of the spaceship by change the value of DELAY1.

***module spaceship:***

This module describes the defalut position of spaceship, and how the spaceship moves, and in what kind of situation the aliens hit the spaceship. It also defines how many lives do you left(I define the default number of lives to 2, you can change it if you want).

***module rocket:***

This module mainly defines how the aliens are hit and gone.

***module speaker:***

This module was intended to generate a sound output when the bullet hit the spaceship or the aliens, but I didn’t find a way to make it happen on the board, I may try to realize this later if I have time.

***module alien:***

This module decides the fire frequency of aliens and the state of aliens, it also defines whether the alien is alive and how the aliens move

***module Pulse:***

This module uses a simple FSM to generate a pulse which is used to match the btnL, btnR, fire, etc. Output pulse will be set to 1’b1 when count reach 200000, that means under a 50MHZ clock, the output will generate a pulse every 0.004 second.

**Pictures preparation:**

Xilinx vivado offer a method to store static picture in RAMs or ROMs. In order to use the ip Block Memory Generator, we need to create an array in .coe format. This can be learned by this blog which has implemented a static picture VGA display function.

https://blog.csdn.net/rzjmpb/article/details/49914197

Blk\_mem\_gen\_0~7 is used to initialize the start scene, background, three kinds of aliens, spaceship and win or game over result. The BMG ip provided by vivado will be realized by the on-chip BRAM. And I have chosen the single port RAM, because we just need to read data from the RAM and don’t need to change the data.

**1、FPGA中RAM的基本概念**

　　在FPGA的设计中，常用的数据缓存IP有FIFO和RAM，其中RAM又分为单口RAM，伪双口RAM和双口RAM。

　　单口RAM与双口RAM的区别在于，单口RAM只有一组数据线与地址线，因此读写不能同时进行。而双口RAM有两组数据线与地址线，读写可以同时进行。

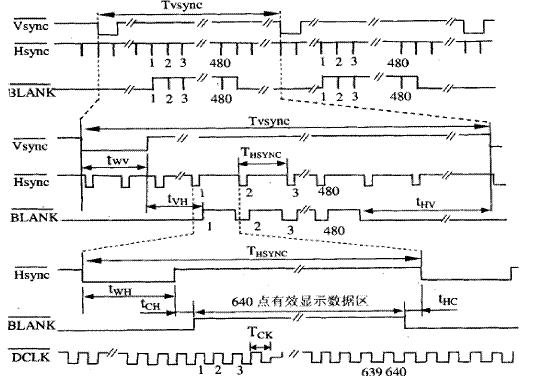
　　伪双口RAM（Xilinx称为Simple two-dual RAM），一个端口只读，另一个端口只写。

     双口RAM：双口RAM 是在一个SRAM 存储器上具有两套完全独立的数据线、地址线和读写控制线，并允许两个独立的系统同时对该存储器进行随机性的访问。即共享式多端口存储器，读写可同时进行。而单口RAM只有一组数据线与地址线，因此读写不能同时进行。

      FIFO也是一个端口只读，另一个端口只写。FIFO存储器分为写入专用区和读取专用区。读操作与写操作可以异步进行，写入区上写入的数据按照写入的顺序从读取端的区中读出，类似于吸收写入端与读出端速度差的一种缓冲器。FIFO与伪双口RAM的区别在于，FIFO为先入先出，没有地址线，不能对存储单元寻址；而伪双口RAM两个端口都有地址线，可以对存储单元寻址。

**2、PL如何驱动VGA输出**

VGA的协议比较简洁，主要是有五个信号线组成，行同步信号 HSYNC，场同步信号VSYNC和3条色彩电压传输信号（R、G、B分别对应）。



zybo开发板上PL的时钟是125MHZ，又因为实现的分辨率是640\*480，帧率是60Hz,所需要的时钟是25Mhz。因此要采用分频（调用ip：clocking wizard，输入125M的clk，输出两个频率分别问25M和50M的clk）

Module horizontal以及module vertical控制hsync以及vsync。

Module horizontal(clk, rst, haddr, hsync, display\_area);

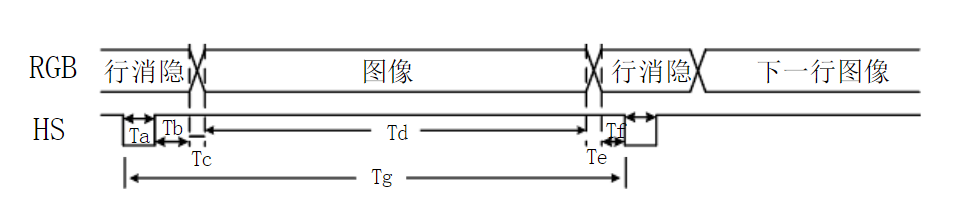
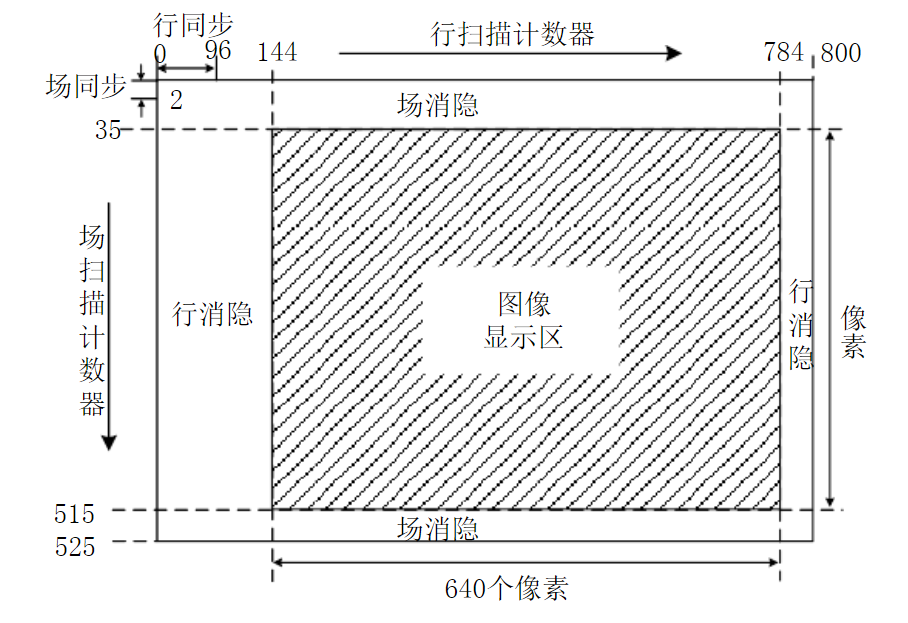
Input clk, rst;

Output hsync;

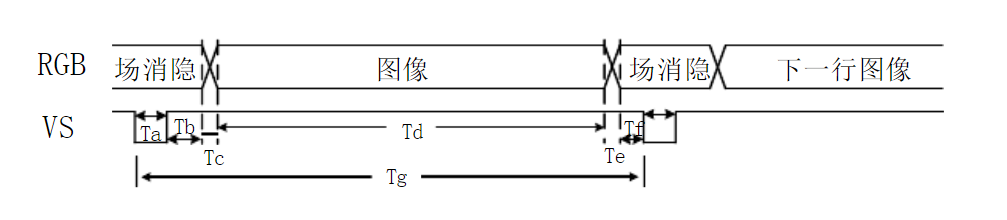
output reg [9:0] haddr; //haddr为待显示的像素的水平坐标（也就是x坐标）

output reg display\_area;

大概流程是：VGA扫描是从左上角扫描到右下角。采用2个计数器进行计数，行计数器的驱动时钟为25MHZ，场计数器的驱动时钟为行计数器的溢出信号，因为例化时使用的时钟是CLK\_50所以计数值从799变成1599。



Ta是行同步头、Td是行图像



Ta是场同步头、Td是场图像。

消隐期间RGB数据线上数据全部强制输出全0。用MUX2\_1实现，非消隐期间才输出图像数据。



horizontal HSYNC(CLK\_50,rst,haddr,VGA\_hSync,Xdisplay);

vertical VSYNC(CLK\_50,rst,vaddr,VGA\_vSync,Ydisplay);

VGA\_hSync和VGA\_vSync是场和行的同步信号，用来同步输出。