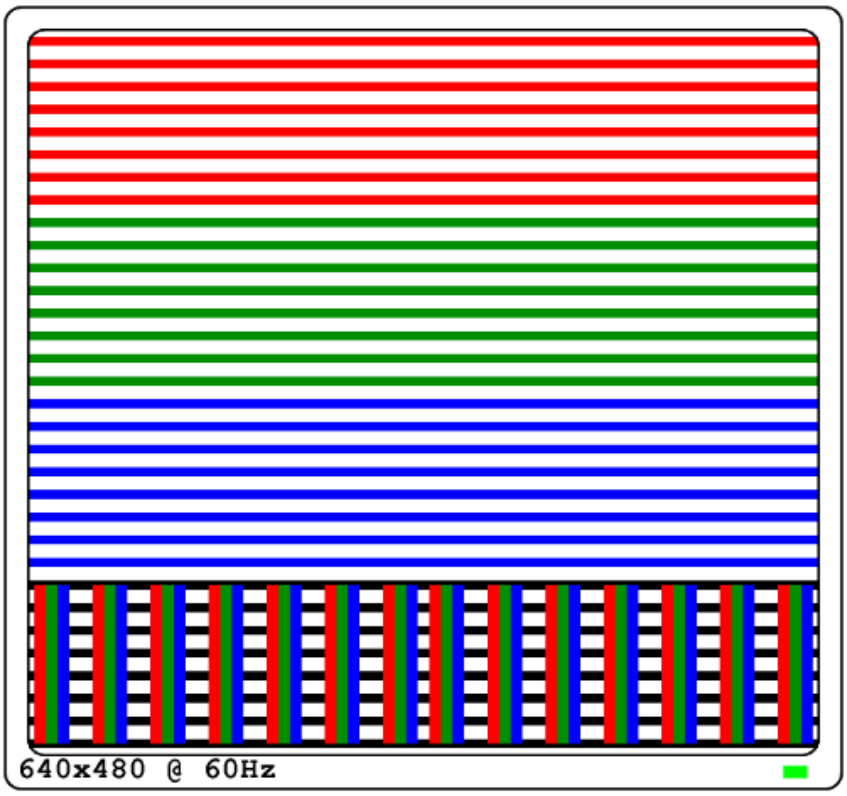
Electrical & Computer Engineering Department
University of Thessaly

CE430 – Digital Circuits Laboratory
3rd Assignment 2018-2019
Victoria Biliouri 2138

VGA CONTROLLER

PURPOSE

The purpose of this project is the implementation of a VGA (Video Graphics Array) driver. In particular, in the first part of the project, a video RAM, which includes information of a specific image, is created, and in the final part, this image is controlled and displayed in a conventional screen.



The image that I have created is:

(with some, of minor significance, alterations).

Part A

In this part, I created a video RAM which contains the colours of the image. The implementation of the VRAM, uses three block RAMs, each one of them represents one of the three basic colours of Spartan 3 FPGA, red, green, blue.

In order to save some space of the FPGA, the size of the VRAM I created is 128x96, which is magnified by the VGA controller.

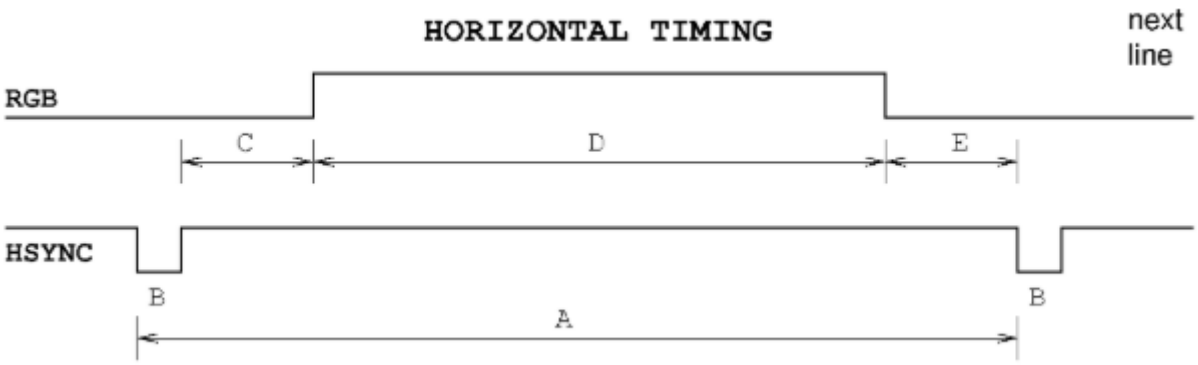
The block RAMs contain the value of bits in hexadecimal system, so every value in VRAM module corresponds to four pixels of the image.

Finally the VRAM modules gets two signals from the VGA Controller in order to know, whether it is time to send a pixel's colours, or to show black colour ('0' values).

RED	GREEN	BLUE	COLOUR
0	0	0	black
0	0	1	blue
0	1	0	green
0	1	1	magenda
1	0	0	red
1	0	1	purple
1	1	0	yellow
1	1	1	white

PART B

In part B, the horizontal driver is created and the HSYNC signal is controlled. Spicifically, the hsync_controller module, controlles the duration of the periods that the pixels are about to be displayed. The HSYNC signal gets the value '0' before a line is displayed, and the value '1', in every other period, as described in the diagram below:



In order to accomplish the correct timing, I designed an FSM ,the states of which respresents the period times B, C, D, E. In every state, a counter counts the time, according to the duration of each period as shown in the table:

PERIOD	DESCRIPTION	TIME VALUE
A	Scanline Time	32μsec
B	Pulse Width	3.84μsec
C	Back Porch	1.92μsec
D	Display Time	25.6μsec
E	Front Porch	0.640μsec

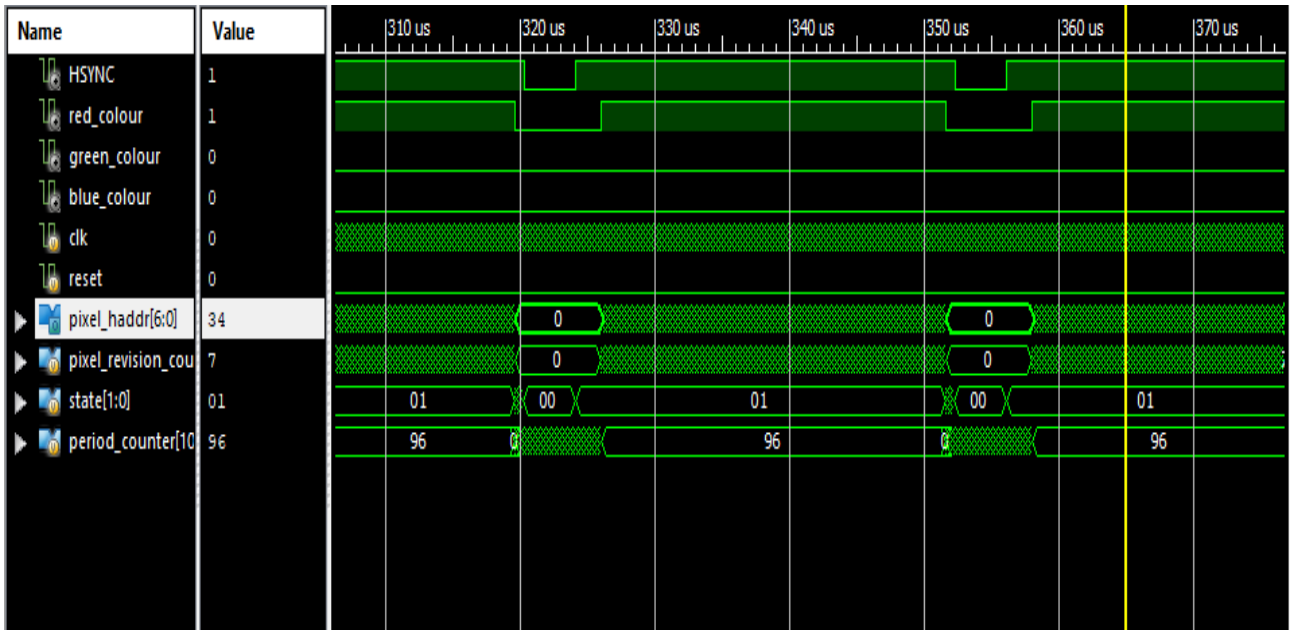
The size of the screen that I want to use is 640x240, but the VRAM that I have

created is 128x96. For that purpose, every pixel should be displayed five times (or ten times because I used a 50MHZ clock). So, I added an additional counter (pixel_revision_counter), which counts the times that a pixel is shown. When it reaches its high value (1001), it returns to zero.

Finally hsync_controller module, gives as output, the signal haddr_enable, for the Video RAM. This signal is '1' only in period D (Display Time).

After the implementation of hsync controller, I checked the correctness of my program by creating the tesbench and the simulation (in which I display the second line of my vram).

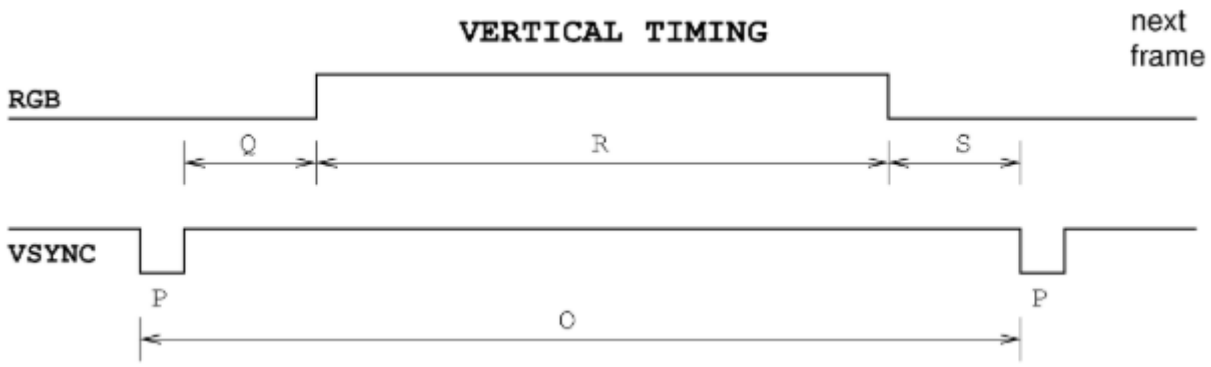
Simulation Results:



PART C

The part C of the project, it to create the vertical display controller, and join it with the previous parts, so as to complete the VGA Controller.

The vsync_controller module functions in a similar way with the previous part.



Each period's duration is:

PERIOD	DESCRIPTION	TIME VALUE
O	Total Fram Time	16.67 msec
P	Pulse Width	64 μsec
Q	Back Porch	928μsec
R	Active Video Time	15.36msec
S	Front Porch	320μsec

VSYNC Signal gets the value '0' in period P, and the value '1' in every other period.

The period_counter and pixel_revision_counter counters, function similarly

with the horizontal process.

The additional part of this section, is that I added one more counter, which counts 32μsec (horizontal A period), so that the vertical controller can understand when a full line is displayed, and when we should proceed to the next line.

After the implementation of the full VGA controller, I checked the correctness of my program by creating the tesbench and the simulation. Also we have implemented the bit file to the Spartan 3 FPGA.

Simulation Results: