TÉCNICO LISBOA

Projecto e Controlo em Lógica Digital

4th° Laboratory

Test Image

Objective: To implement test images in the VGA

Without using a pre-recorded image in memory, display in a computer monitor a set of test images. The image to be shown is controlled by the switches:

SW[2:0]=0 : All the display is black
SW[2:0]=1 : All the display is green
SW[2:0]=2 : All the display is red
SW[2:0]=3 : All the display is blue
SW[2:0]=4 : The image on the right hand side

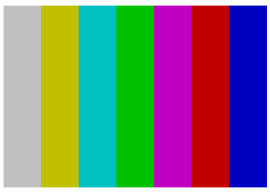

Image definition: Seven vertical bars with an intensity of 75%. The colors of the bars are, from left to right: white, yellow, cyan, green, magenta, red and blue.

Start by defining the output resolution and the refresh rate of the screen that compatible with the display to be used.
Build the VGA clock. Use one of the crystals on DE2 and a PLL to generate a signal with the required frequency.
Build the synchronism pulses (horizontal and vertical) for the VGA, respecting the VGA signals structure.
Build a module that outputs the RGB signal according to the scan position. Implement the top module that controls the RGB output in such a way that the system has the required behavior.

FROM THIS POINT ONWARDS THERE ARE OPTIONAL TASKS THAT WILL NOT BE TAKEN IN CONSIDERATION IN ANY WAY FOR THE EVALUATION OF THIS LABORATORY. HOWEVER THEY CAN BE USEFUL FOR THE FUTURE…

Extra Task - Optional

Typically the display in a screen is made in two steps, or two modules, using an intermediate memory. A first module writes in the memory the color representation for the pixels. The second module controls the screen pixels, transferring the information contained in the memory to the pixels in the screen. In this way the “display” module does not make decisions and as such there is no timing constraints for decisions.

Change the previous module in such a way that it reads the data from a memory according to the position of the pixel in the scan. There should be a direct relation between the pixel being addressed ($pixel_{i,j}$) and the memory position. Play with the values in the memory. You can, for example, write “PCLD”.

Imagine you have a ball (represented in a 64 pixels memory). How could you “introduce” the ball? And if you it to move?