Python Image Converting Script Tutorial

Converting an Image to SystemVerilog Assignments

The provided python code prints out the SystemVerilog assignment statements for an array. The array will be the size of the image input to the python (in terms of pixels), and will store a 1 or 0 for a particular pixel based on whether that pixel was "drawn" in the image or not. An example output is shown in Figure 1 below, which you would copy and paste into your SystemVerilog files. You can use this code to help with the laborious process of trying to store the shape of complex shapes, letters, etc. in SystemVerilog for outputting to the VGA display.

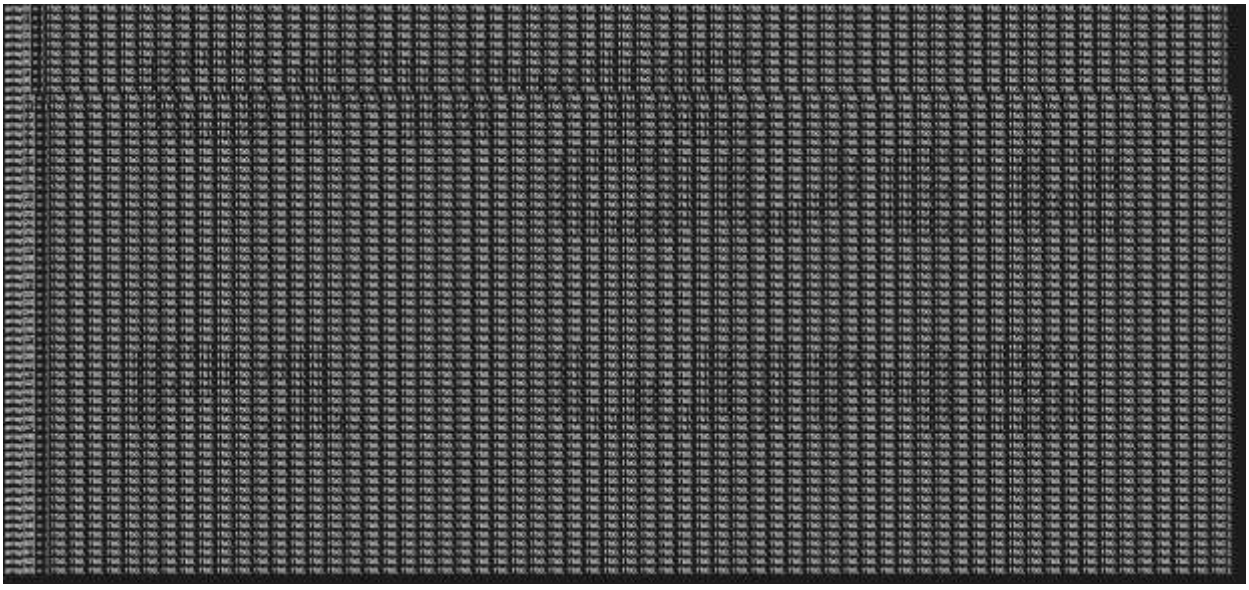

Figure 1: *Example output from the python script – although the output is very small and just a bunch of `1'b0` and `1'b1`, your eyes might be able to discern that this image contains the text "GAME OVER P1 WINS".*

Some Notes:

- This code was only used with a black and white image – more specifically, the image was drawn in black, the background was transparent, and it was saved as a png. It is unknown how other images might behave.
- This handles every pixel in the image. You will likely want to be drawing something that's low resolution or meant to take up a small number of pixels on the screen. Make sure you're creating and inputting an image that is the correct size (probably very small).
- The image needs to be in the same folder as the python file.
- If you don't have python installed on your computer, you can run the script/code in a Jupyter Notebook, using something such as Anaconda or Google's Colab.
- The input to the python function is the name of the image, and you can edit the function to change the name of the array that is being assigned.

Hopefully this helps with some of the tedious assignments that would accompany fancy animation and makes it easier to create better looking projects.