

Designing a game engine in Verilog featuring VGA, mouse and keyboard driver and a PowerPoint UI designer

By Alejandro Cabrerizo and Will Zeurcher

Our goal is to provide an easy to use package that provides all the required functionality to develop a 2D videogame in Verilog.

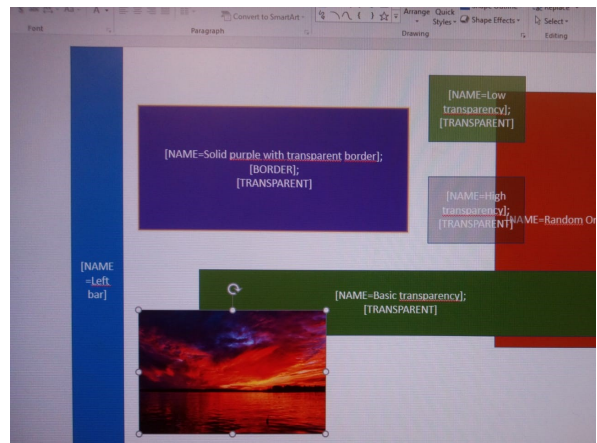
Apart from that, we noticed that designing a UI to be shown in an FPGA takes a long time since a change in the layout would require a full recompilation. (Taking more than 15 minutes sometimes)

That is why we are developing our own PowerPoint to Verilog compiler, which allows you to create a slide and design how the screen will look and describe event and then compile it to Verilog code runnable in a FPGA.

We have already developed the VGA display driver and the basic functionalities to be able to display basic shapes (Allowing transparencies) and pictures.



FPGA output



PowerPoint slide

To be able to accomplish all of that, we had to create and implement several compression algorithms like vectorization, pixel condensation and lowering the color bit count.

All this properties can easily be changed in PowerPoint by adding TAGS to each element.

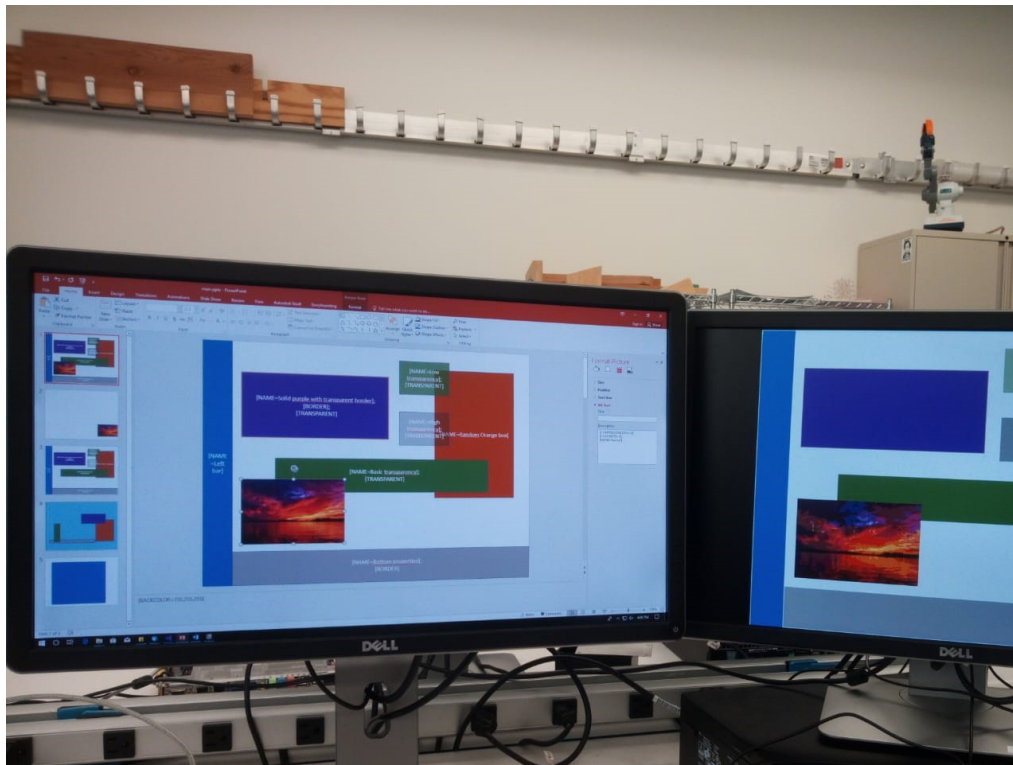
To be able to use Verilog to interact with a UI element, the tag [EXTERNAL] must be used, which makes wires controlling the x, y position and other properties editable in Verilog.

But even more important are the features that are still missing, like a collision system, SRAM data loading and sprite handling.

Currently we are going to try to implement:

SD card data loading / UART data loading

Frame buffered VGA driver. (We need to know how to manage memory for that)



Github repo: <https://github.com/alecamaracm/ECE287Project>

