**HW\_LAB\_4: VGA and Sprites**

**SOC Design with Programmable Logic**

ECE 540

Winter 2023

Due date: Tuesday, Feb 14th, 2023

**HW\_LAB\_4.A:**

Add the initialized RAM/ROM, dtg, and related connections and display to the VGA display.

* Make a new vga module and add it to the filelist for the FPGA and instantiate/hook it up inside swervolf\_core.v
* Add the new VGA definitions to your XDC file from the included xdc. Pushbuttons are not defined in this version and you can merge this with your own with pushbuttons.
* Add the new ports to rvfpganexys.v
* Add the dtg.v (Display Timing Generator) to your file list and instantiate it inside your new vga module. Add the connections from the horiz\_sync, vert\_sync through any module levels up to the top of the design.
* Create the IP for the RAM/ROM frame buffer initialized with COE file to the new vga module as shown in class slides.
  + Instantiate it inside your vga module.
  + Connect the clk port for both read/write ports, but tie off any other write ports to 1'b0 so no writes happen.
  + The read address is the pix\_num from dtg.
  + The read output of the memory will get tied to identically to the VGA Red, Green, Blue outputs.
* Add a Clock Divider for 31.5 Mhz for use with the vga.
  + Instantiate this at the top rvfpganexys.sv and connect it through to your vga module.

If all went well, you can write your bitstream and the image will be displayed. No assembly code needed.

**Submission: Submit PDF with verilog that has changed and video of the image (Video of Part A/B can be combined).**

**HW\_LAB\_4.B:**

Add a mouse cursor sprite to your system.

Add a sprite that is 4x4 in pixels. To start it can just be a white box. It is preferred that it is in the shape of a cursor with transparency if possible.

* This will be an initialized 4x4 array, but does not need to be a block ram, but can just be an inferred array.
* Expand your Pushbutton Peripheral to make a read/write register as well for the sprite row/column location on the screen. This will be add address 0x1504 for the row and address 0x1508 for the column.
* Make new Verilog code to check the pixel\_row/column from dtg and compare it. When it is in the correct range, display the cursor instead of the background.
* Make assembly to connect the pushbuttons to the movement of the cursor. Initially, you can tie it together in Verilog, but the end submission should be with coding that changes the cursor location; you could make it switch configurable to move with/without the assembly coding.

**Submission: Submit PDF with verilog that has changed and video of the image (Video of Part A/B can be combined)**