**Purpose:** Demonstrate competence using Verilog and Xilinx Vivado to implement a timing-critical interfacing project using the Digilent Basys 3 FPGA board. You may collaborate regarding source code and design concepts with other individuals (including online sources) regarding this project; however, you must provide fair attribution to the source(s) of this material. **In summary, you must identify all contributors to your design by clear attribution in your code and your written report.**

**Procedure:** Your task is to design, simulate, and test in hardware (on the Digilent Basys 3 FPGA prototyping board) an implementation which displays variable color patterns on a standard computer monitor, using the VGA output connector on the Basys 3 board, with the monitor being driven at the SVGA mode of $800 \times 600$ with a 72 Hz refresh. Levels of achievement for this project are listed below (such that the point ranges listed are associated with the score, on a 100-point scale, to be assigned to your project):

1. (up to 50 points) If you don't submit a working project, your award begins at 0. How well you describe how it *should* have worked, in the written report, dictates how close to 50 points you get.

2. (up to 70 Points) A simple modification of the provided VGAStart design. The design is available on the ECE network drive and the WyoCourses website. Look under Verilog Examples, VGA, VGAStart. This level of achievement is characterized by a simple display that changes field colors and/or sizes based upon user switch and/or button inputs.

3. (up to 85 Points) This level of achievement is characterized by a single moving "icon or sprite" on the display. The span of points in this category can be enhanced by the detail of the "icon or sprite" (for example, if it changes in any way (size, velocity, color, path) with interaction with static points on the display or boundaries of the display).

4. (up to 100 Points) This level of achievement is characterized by a complex "interactive" display involving two or more moving "icons or sprites" which change in display form through interaction with each other via "contact" on the field of the display. This can include either no user interaction (i.e., the display is self-generating) or via user input (think of the classic "Pong" video game).

**Turn in:** For this Project, turn in electronically (as one zip-file attachment) the following items. Send the zip file as an email attachment, sent to `chgw@uwyo.edu`. If a single zip file is too large to send, you can send two zip files, but be sure to add the "part1" and "part2" suffixes to the file name. The zip file(s) must contain:

- A one-page written description of the functionality to be provided by your design, as a PDF file. That is, explain in layman's terms (i.e., language that does not require an education in electrical or computer engineering) specifically what your "Device" does. **Explicitly state what level of achievement (i.e., 1–4) you are claiming for your design.**

- A 2–4 page technical description of your modular implementation of the "Device," as a PDF file. This description should include the following:

  - An "RTL Schematic" of the top-level design of your implementation. If needed for clarity, this can be a separate PDF file, with a descriptive file name.
  - An inventory (tabular list, for example) of the Basys 3 user-interface resources which are utilized by your design. For example, which switches, buttons and Basys 3 LEDs are utilized for what user-interface functionality.

- A brief description of the functionality of the individual Verilog modules (that is, from a black-box sense, what functionality each module provides). Feel free to include ASM charts for one or more modules (as an appendix that doesn't count against the page limit) to document the operation of your FSMs, if you feel it helps convey the desired information.

➪ Your zip-file must also include the following (in a separate folder within the zip file):

- All Verilog source files (filename extension `.v`) utilized in the implementation of your project. You should also include any test benches and simulation results to demonstrate the desired behavior of your appropriate modules.
- The Xilinx Vivado design constraints file (filename extension `.xdc`) utilized by your project.
- The bitstream file (filename extension `.bit`) produced by Vivado to program the FPGA.

➪ Finally, to visually show how your project works, submit a **brief** video file (in a standard file format such as `.mp4`) that demonstrates your project in operation. Your team members should show up briefly at the start of the video as a way to show that it's your design. Don't go overboard with this! When I wrote "brief" above, I meant it. I highly recommend the `.mp4` format (there are free converters online that convert other formats to mp4) as this format is the most efficient regarding file size.

These specific files **(and ONLY THESE FILES, no other content)** should be placed in a zipped archive file with the following name:

`EE4490_Proj03_Lastname1_Lastname2.zip`

The two students for the team should substitute their last names in the obvious two positions of the filename. Note that the separators in the filename are underscore characters, not spaces. This file must be emailed or uploaded no later than 1600L on Friday December 13, 2019.

You are free to create the written documents described above using any program, such as LaTeX, Microsoft Word, etc., but the file you submit must be a valid PDF file, having a page size of 8.5 x 10 inches, with approximately 1-inch margins all around, and with the main body font size no smaller than 10 points (12 points is preferred). The writing style should be serious[1] yet also brief, clear, and concise.

❄❄❄ Enjoy. . . ❄❄❄

[1] Ask yourself if you'd be comfortable submitting this to your boss at your first engineering job after graduation. Proper grammar and spelling, along with logical, organized sentences are required.