Project Proposal:   
FPGA Video Game

**Author: Dale Andreoli**

**Date: 25 September 2021**

**Class: ECEN 397**

# Purpose

This project will demonstrate knowledge and skills expected of a senior-level Computer Engineering student by using a Field Programmable Gate Array to implement a video game. The project will use an FPGA to display video, output sound and take input from a game controller while also handling the game logic. Each of these components will require significant research and testing in order to be successfully implemented.

There is already a lot of documentation to facilitate using an FPGA to handle the output of audio and video. Game controllers are also well documented. It should not be a huge obstacle to implement one as an input device. The biggest obstacle I foresee is creating game logic from scratch in a hardware description language. I have encountered many FPGA projects online that can run retro games from older systems, but only a few of them mention building a game from scratch. My brief research so far has led me to believe that it will indeed be possible.

As a Computer Engineering student, gaining further knowledge and skills relating to FPGAs will be valuable as some of my potential career paths involve hardware description languages. This project will provide new challenges to overcome as I research and implement its various components. This will increase my competence with hardware description languages and demonstrate an ability to learn new skills through research.

# Methods and Procedure

The plan for this project is to implement it using technologies I am already familiar with from prior courses. This would make the board of choice a Basys 3 Artix-7 FPGA similar to those available on campus. The hardware description language used in the project will be Verilog. Vivado will be used to write, synthesize and analyze the design. Each of the various components of this project will be implemented as one or more Verilog modules.

The following broad tasks will each implement a different major component of the project:

## Video Output

The project will need to output video to a screen. The simplest way to do this would likely be using VGA to output video. The Basys 3 Artix-7 FPGA has a built in VGA output. Utilizing this provided resource will likely save some hassle. A computer monitor that has VGA input will be an accessible resource as the screen for this project. Research will need to be done to other options such as HDMI in case there are significant benefits or challenges to other video output formats.

## Audio Output

The project will need to output audio to a speaker. The FPGA does not feature a built-in speaker, so I will need to research what kind of speaker will work best for this project. If the game sounds are simple tones and beeps, a cheap buzzer or general-purpose speaker should do the job just fine. More research can be done on other options for sound output after it becomes clear what kinds of sounds the game needs to produce.

## Controller Input

The project will need to get input from a game controller in order to play the game. In my preliminary research, I found a lot of useful documentation for the classic Nintendo Entertainment System controller. This controller sends all of its button values through a single data pin, which is read with the help of a clock signal and a latch signal. The NES controller seems to be a simple and well documented input device for this project. However, I would like to perform further research into other readily available controllers and input devices. It is possible that there are better options.

## Game Design

The project will need to play a video game. This game will be built from scratch in Verilog. Research will need to be done to determine exactly what the limitations of Verilog and an FPGA are for programming a video game. Something as simple as Pong or table tennis would certainly be doable, but something more complex would be much more satisfying. Researching what can reasonably be done in the timeline of this project will help determine exactly what kind of game will be created. It may even be feasible in the scope of this project to build multiple games onto the FPGA.

## Game Development

The game logic will all be created using Verilog and implemented onto the FPGA. This will include keeping track of the various game objects and their characteristics such as position, velocity, color, size and shape. It will need to handle events such as collisions and scoring. The game logic will also need to render what should be sent to the VGA output module. Specific needs for game logic will be greatly influenced by what kind of game is decided on during game design.

## Testing and Demonstration

Each of the above components will be tested individually and collectively. It will be important to ensure that the audio and video output work as intended before trying to get a game working on them. The controller input can be tested individually by using the built-in LEDs on the FPGA.

The completed project will be tested and demonstrated by connecting the VGA output to a computer monitor or TV and playing the game that is built on the FPGA.

# Work Schedule

The broad tasks outlined in the Methods and Procedures section will serve as the most significant milestones that need to be achieved. Each milestone can be broken down into specific tasks to help reach that milestone.

## Audio Output – 7 hours

The audio output should be the simplest of all the components of this project. I would like to spend some time researching different methods for implementing sound in an FPGA project (2 hours). After finding the best method, implementing the sound output module in Verilog should not be too complicated. (2 hours). It will be important to test the module with different types of sounds before we mix in other components (1 hour). I would like to know the limits of the speaker and what we are allowed to output. It would be important to see what I have control over such as volume, pitch range, wave types (2 hours).

## Video Output – 12 hours

The video output will take a little more work than the sound. It will be important to research various video output methods instead of automatically proceeding with the built-in VGA output (4 hours). After deciding on an output method, a Verilog module can be created (4 hours). Before mixing the video display with other components, it will be important to sufficiently test it with different outputs (1 hour). I will need to become comfortable with the module and how it works so that I can get full use out of it as I build the game. I would like to play around with the module and see what kind of things I can draw and in what colors and resolutions it can be done (3 hours).

## Controller Input – 12 hours

The first step to implementing a controller as an input device will be researching the various options (4 hours). The NES controller seems doable from what I have seen in its documentation, but perhaps a modern controller that uses USB will work just as well. The Basys 3 Artix-7 FPGA board even has a built-in USB port. After deciding on an input device, creating a Verilog module for it will be the next step (6 hours). This module can be tested by assigning a built-in LED on the FPGA to each of the buttons on the controller (2 hours).

## Game Design – 6 hours

Before I begin work on building the game in Verilog, I will need to research methods and strategies for creating a game in a hardware description language (3 hours). This will guide my decision for what type of game to create. It will be helpful to map out the game and its features before writing and Verilog code (3 hours). This design will include what kinds of game objects I will need to implement, and what sorts of interactions they will have with each other. By having all the different parts of the game logic decided on, it will make writing the Verilog code more manageable.

A game will be designed that fulfills the time requirements and scope of this project.

## Game Development

Developing the various modules that handle different aspects of the game will be a significant portion of this project. The specifics will become clearer as the type of game is determined in the research and design task. For now, the process of game development can be divided into categories that will be common to any type of game.

### Game objects – 30 hours

A Verilog module will need to be made for each game object that will be used. For example, a game of Pong would best be implemented with a separate module for each player, a module for the ball, and a module to keep track of the score. A more complex game would require even more objects to track. Each Verilog module would take a few hours to design, implement, and test. The game design task will ensure that there are at least six game objects to create (5 hours each).

### Game Events – 25 hours

Each game event such as movement, collision or other physics interactions could each be handled by their own Verilog module. Each module will take a few hours to implement and test. The game design task will ensure that there are at least 5 game events that can occur (5 hours each).

### Rendering – 10 hours

A Verilog module will need to be created to handle the rendering of the game (10 hours). This module will be passed in data on the game objects that it needs to draw. This will send data to the video output module to be displayed on the screen.

## Collective Testing

Once all the modules are put together, it will be necessary to test the device as a whole and troubleshoot any potential problems that arise from combining everything.

## Milestone Chart

The following chart provides a goal to be worked toward each week. The current work schedule adds up to 102 hours.

|  |  |
| --- | --- |
| Date | Milestone |
| Week 02 – 10/2 | Audio output module |
| Week 03 – 10/9 | Video output research and design |
| Week 04 – 10/16 | Video output module |
| Week 05 – 10/23 | Controller research and design |
| Week 06 – 10/30 | Controller input module |
| Week 07 – 11/6 | Game design |
| Week 08 – 11/13 | Game objects |
| Week 09 – 11/20 | Game objects |
| Week 10 – 11/27 | Game events |
| Week 11 – 12/4 | Game events |
| Week 12 – 12/11 | Game rendering |
| Week 13 – 12/18 | Collective Testing – Finishing touches |
| Week 14 – 12/20 | Final Report |

# Results

This project will demonstrate knowledge and skills that can be expected from a senior-level Computer Engineering student. It will also prove my ability to learn new skills and techniques through research. It will show significant competence with hardware description languages and provide valuable experience relevant to a potential career path.