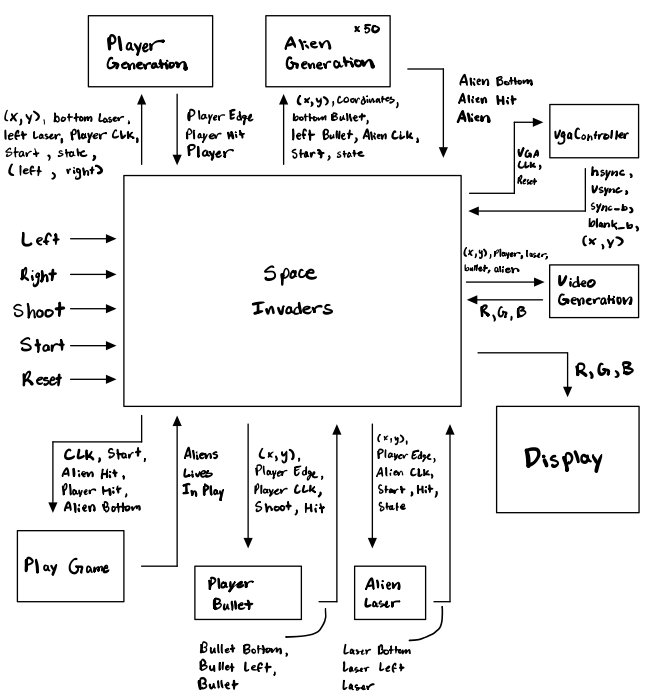
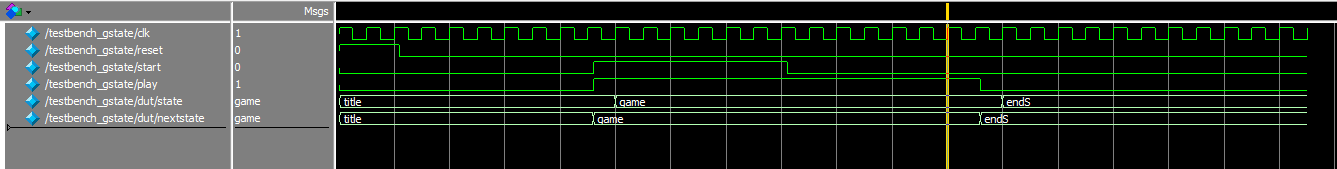
Project Overview

The goal of this project is to emulate the classic arcade game, “Space Invaders,” using a DE2-115 FPGA board. Space Invaders is a single player game, where the player controls a moving laser cannon with the aim of defeating a group of invading aliens. The player can win by defeating the entire wave of enemies. The player loses by allowing the enemies to reach the player, or getting hit by the alien lasers 3 times. The game's controls include pushbuttons to start the game, shoot, and move the player left and right. There is also a reset switch to return to the title screen. The game is then output to a monitor via a VGA signal generated by the FPGA board. Using System Verilog a collection of modules were designed that are then synthesized and uploaded to the FPGA. The project consists of a main module calleds spaceInv that instantiates other modules to create the game. Inside of spaceInv is a state machine that is used to determine what is output to the monitor. It also includes clock dividers to generate a clock for the VGA signal, and clocks for both the player and aliens. The first major module spaceInv instantiates is the vgaController. This module uses the VGA clock from the main module to generate the VGA signal for the entire screen. The next module that results from the main module is videoGen. This module takes signals from the results of other modules and uses them to assign the color to each pixel. The next two modules are the generate player, and generate alien modules. These two modules control the location of the entities on the screen, along with detecting object collision. The module playG that is also instantiated from spaceInv takes the outputs from both generate player and generate alien, and uses them to remain in the game state, along with calculating the result. Overall, the final goal of the project is to emulate the game “Space Invaders” in a way that demonstrates understanding of the lessons taught in CPE 200.

Block Diagram





Simulation Waveform

The waveform shown above from the simulation of the testbench shows the operation of the state machine found in the game’s main module. The state machine starts off in the first state, known as title. From there when the player presses the start button, the start signal goes high, along with the play signal. The state machine then moves to the game state on the next clock edge. While in the game state, the user plays the game. The state machine remains in the game state while play is high. Play remains high as long as the player has neither won or lost the game. Then when play goes low, the game ends, and the state machine moves to the end state. Overall the simulation waveforms resulting from the testbench demonstrate the functionality of the state machine.

Project Conclusions

At the end of the project, the overall design of the game has been implemented. While still using simple objects, the game includes a controllable character, moves enemies, and a shooting system with collisions. The final version of the project includes a main module that instantiates other modules in order to accomplish the feats listed above. Implementing the functionality described above was not easy, with each step forward new problems would arise that forced modifications and adaptation. Ultimately, the changes forced by the issues turned into useful lessons that aided in the further design of the game. The project itself still has room to grow and improve to make playing the game an even more enjoyable experience.

While completing this project, there were a lot of lessons learned, and a lot of problems to overcome. The first of these lessons was manipulating the color output for each pixel. This is done by changing the values of r, g, and b for each pixel. The second thing that was learned was how to implement moving entities in the VGA output. In order to do so, the positions of the entities were stored in latches and updated at each positive clock edge. The next thing that has been learned so far is implementing a state machine for the game itself. This state machine uses an enumerated type and updates from state to next state at reset and each clock edge. It then uses a block of combinational logic to determine which state to move to at the next clock. The final big lesson that was learned while implementing the project, was detecting collisions between the player and alien laser, or between alien and player bullet. The collision detection was implemented by comparing the location of the bullet or laser to the location of the alien or player. Then if they were in the same location a collision had occurred, and the result was used for the game.

Although at this point the game is functional, and reminiscent of the original Space Invaders, there are still some places that can be further developed. One of the possible implementations that can be included is a working score system. This could be done by either printing directly to the game display or the seven segment display on the FPGA board. This counter concept could also apply to printing out the amount of lives the player has while playing. Moving forward, many design implications could also be made if this project were to be worked on more in the future. For instance, the development of complex entities would make the game more aesthetically pleasing as well as similar in design to the original retro Space Invaders. To take that even further, animation of the aliens can also be implemented rather than having a fixated image set to the aliens. The usage of sprites would make these designs possible, and with that the addition of barricades could also be added as additional protection for the player. With the addition of barricades, the addition of multiple levels and increasing difficulty can also be a future development.