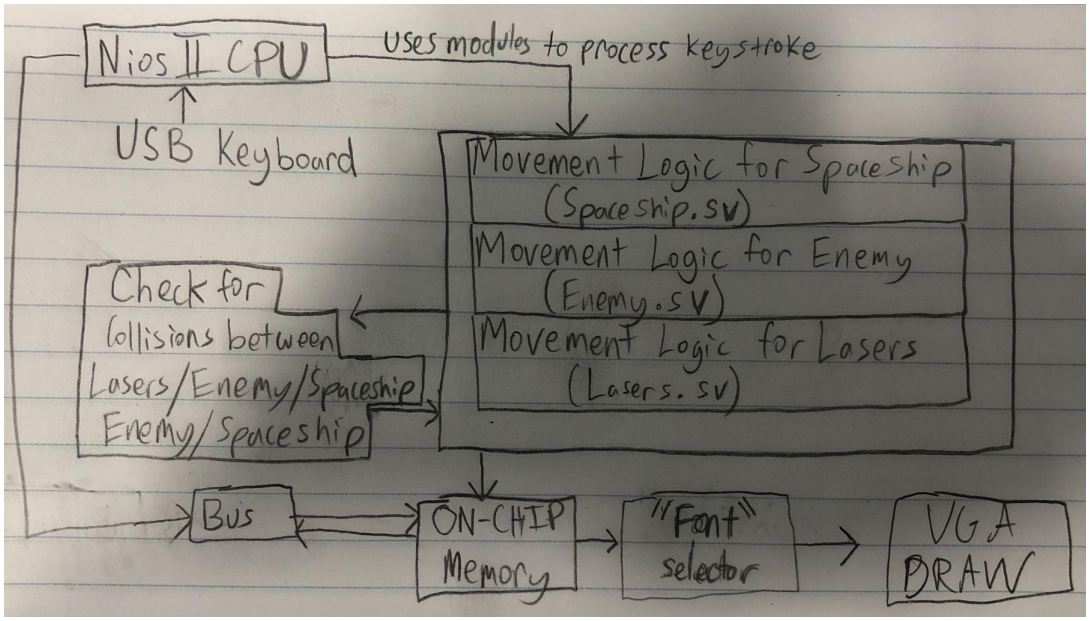
Idea and overview

We propose to design and implement a space shooter game on the FPGA System on Chip using the NIOS II CPU. The player will move a spaceship and fire lasers at enemy ships using a USB keyboard. 2D collision detection and enemy movement, and actions will be controlled by hardware. The rendering of the spaceships and lasers will be implemented using On-Chip RAM and similar hardware as the font_rom file found in lab 7. All of this will be displayed on screen via VGA Draw (controller and color mapper)..
The software will control how often waves of enemy ships appear and what type of enemy ships are contained in the wave.
Our goal is to demonstrate a working version of our game.

Block Diagram for Game Implementation



List of features

List of necessary features to be considered working

- Color output on VGA monitor
- Spaceship control from USB Keyboard
- Enemies behave in a threatening manner
- Lives are lost based on contact with an enemy ship and/or laser
- Score is correctly calculated and displayed based on number of enemies destroyed
- Game increases in difficult as game goes on

`

List of features that are extra to achieve bonus difficulty points

- Implementation of sporadic power up drops that affect ship performance
- Implementation of “bosses” at different score checkpoints
- Explosion animation where enemy ship is destroyed
- Sound for the game **
- Implement option between 2-player and 1-player version of the game **

Expected Difficulty for Project

The expected difficulty for our project is dependent on how many of the extra features we are able to correctly implement. If all of the necessary features are implemented, our project has a difficulty of 6/10 because of the inherent complexity of the Galaga-type game. Each of the first 3 extra features (the ones without **) would add 1 difficulty point if correctly implemented, making a total of 9 difficulty points if all features are present except for sound and 1-player/2-player compatibility. This is because the additional features add a good deal of logic and nuance to the game that will take a lot of time and effort to correctly create. If either sound feature or the 1-player/2-player feature is implemented in addition to all of the extra features, the difficulty level would increase to a 10/10, since implementing both could prove very challenging and is not something that we have developed much experience with in ECE 385 Labs.

Timeline

These goals relate to what is shown on screen.

Week 1:

- Start and finish USB, VGA, FPGA communication
- Start and finish movement and laser control for the player’s spaceship
- Start enemy spawn, movement, and shooting

Week 2 (midpoint):

- Finish enemy spawn, movement, and shooting
- Start and Finish 2D collision and implement player lives and scores
- Start implementing difficulty of enemies (different types of enemies with different stats)
- Midpoint Demo: Player’s ship moves and shoots, enemy ships move and shoot, player loses life if hit, gains score if enemy ship is hit.

Week 3:

- Finish implementing difficulty of enemies
- Start and Finish explosion animation and power ups
- Start implementation of boss stages

Week 4:

- Finish boss stages
- Work on implementing sound or multiplayer