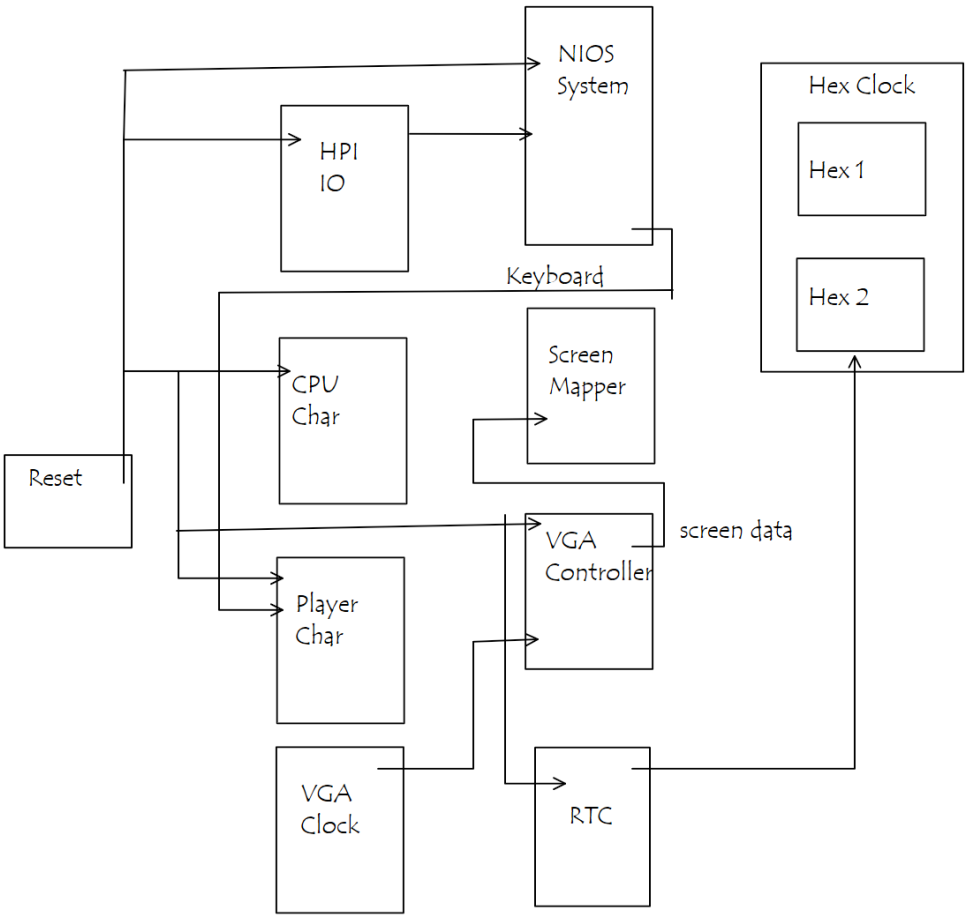
ECE 385 – Spring 2017
Final Project Proposal
Jerry Chen - jchen238
Noah Mathes -nmathes2

1. Idea and Overview

We will be designing and implementing a fighting game, similar to Mortal Combat and Injustice 2, on the FPGA. We will be implementing basic SystemVerilog essentials such as the System, RAM, Video Display, and general PIO blocks for communication between C code and Hardware. Our design will also include a NIOS II CPU for the purpose of interfacing with the USB keyboard as in lab 8 to control the movements and abilities of the fighter. In addition, the NIOS II CPU will be used to control in-game logic. Our goal is to demonstrate a simple game that combines the functionalities of hardware and software.

2. Block Diagram



3. List of Features

Baseline:

1. 2-D motion for a character.
2. 2-D motion for a CPU controlled character.
3. Bullet movement and hitbox registration.
4. Melee action and hitbox registration.
5. Player health and game timer.

Extra:

1. Sound effects when bullet is shot, player melees, and player is hit: Audio
2. Health bar display for user and CPU with real time updates: HUD
3. Pop up display for round start and round won/loss: HUD
4. Combo moves: combining specific move sequences
5. Power Ups: power bar with special moves
6. Character Select: different graphics, different abilities
7. Realistic Physics: game logic
8. Scrolling Screen: game logic

4. Expected Difficulty

5-8. We believe the baseline features deserve a difficulty score of 5. Primarily, we have to generate CPU controlled motion which relies on the user's 2D motion. We want to put extra effort into the CPU motion so it appears as if a user is playing and the CPU's aptitude is ideal. Since we created user controlled 2D motion in lab 8, we have a firm foundation which will ease us into the more difficult tasks. Generating bullet logic, managing hitbox registration and controlling two entities will contain many edge cases that will be difficult to manage. Having a game timer will complicate logic by adding interrupt handling into our game. Overall, these features extend the knowledge gained from lab 8, giving a very realistic baseline.

We believe the extra features are what really enhances the difficulty. By adding sound, we need to manage interfacing with SDRAM to read and play the audio correctly. This requires a firm understanding of interacting with audio files and proper memory management. Adding graphics will always complicate our efforts, and we wish to do so in many of the extra features. The combo moves and power ups intend to require even more skill learning, and we expect these to add tremendous effort, as we can scale the level of difficulty of these categories. Realistic physics adds a mathematical effort which complements our ECE background.

5. Proposed Timeline

By Week 2: We expect to have modules, such as the ones listed in the Overview, created for our SoC. These modules shall interface with SystemVerilog, and we shall be able to prove its functionality by simulating signal control through basic pin assignments or test module.

By Week 3: We expect to have all baseline capabilities done with simple graphics. Simple graphics will consist of limited animations. We should be able to showcase a functional game at this point.

By Week 4: Over the course of this week and next week, we will be adding extra features, listed above. Adding audio effects and a HUD will be our main priority. If time permits, more graphics/animations will be added to the game.

By Week 5: The final week, we will aim to add additional game logic such as scrolling and realistic physics. An example of realistic physics would be gravity during a jump move.