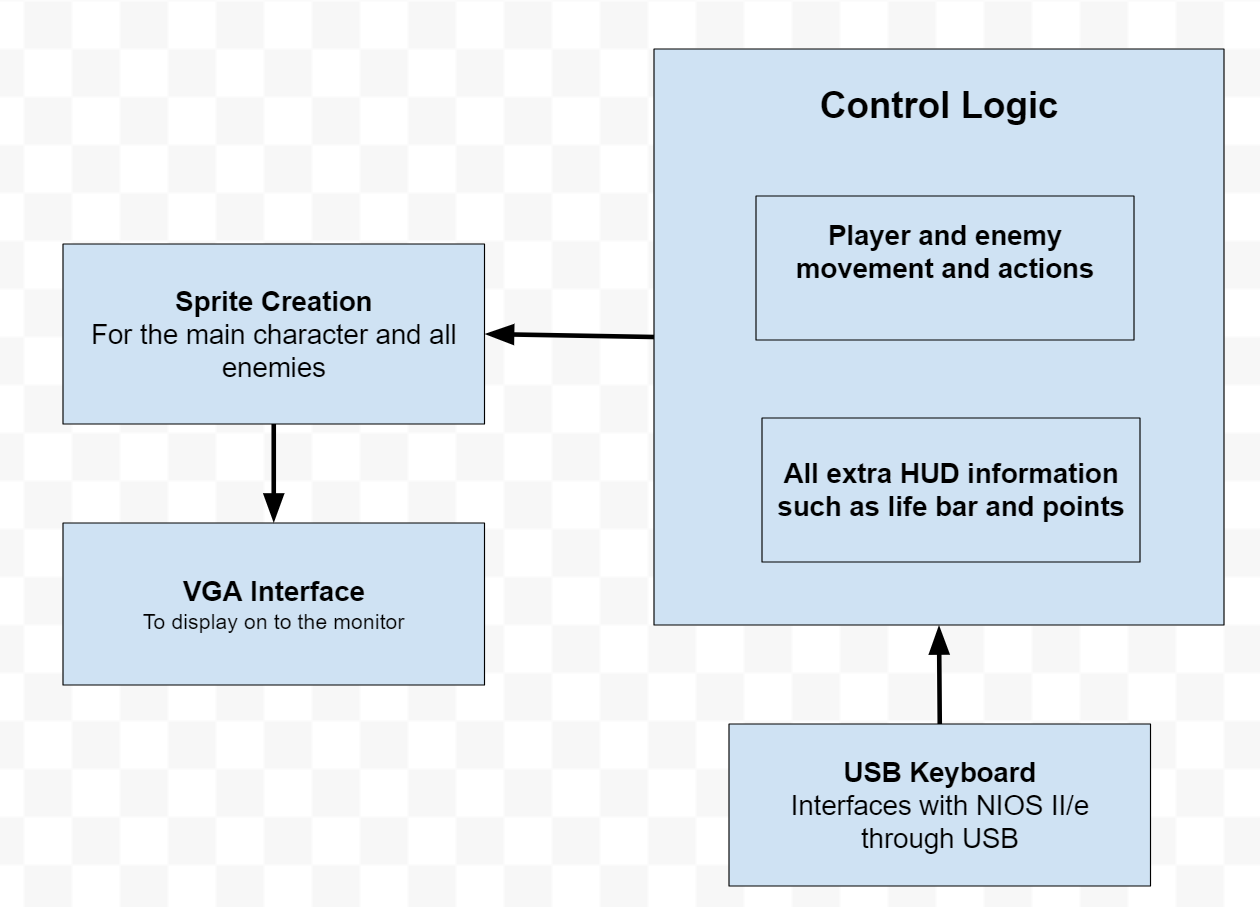
1. Idea and Overview

We propose to design and implement the first level of Mega man, that was first released on the NES back in 1987. This will use NIOS II/e on the FPGA as a System-on-chip. Our SoC will be based around our code from lab 8. Additionally we will implement using SystemVerilog essential components such as Video Display, and Keyboard. Our design will also include a NIOS II/e CPU for the purposes of interfacing with the USB keyboard as in lab 8 so the user can control the main character. We will use C to implement USB input and output. Our goal is to demonstrate the functionality of playing some of the first level of the game “Mega Man” using the USB keyboard and VGA monitor. The final demo will be carried out by the TA by him/her playing our game and judging it based on the added features. We will need an external keyboard to play. We plan on maybe using some code from this github, but it doesn’t make much sense as of yet. Link: <https://github.com/mantrarush/Altera-DE-2>

2. Block Diagram



3. List of Features

1. Baseline set of features for the project to be considered working
2. Main character will respond to user inputs, specifically, will move left, right, and jump from the arrow keys.
3. Going up a ladder is a different button press than the up arrow key
   * 1. Can let go of that ladder by pressing that same button, this should make the main character fall to the nearest platform below him

3. Screen will jump when the main character transitions from stage to stage

1. Additional features that may be implemented for extra difficulty
   1. Stationary enemies are present, but contact to main character will do no damage.
      1. Stationary enemies are present, but contact to main character will do damage to main character.
   2. Moving enemies are present, but contact to main character will do no damage.
      1. Moving enemies are present, but contact to main character will do damage to main character.
      2. All moving enemies will follow you to try and damage you
   3. Main character can shoot projectiles out of his arm cannon
      1. Projectiles can go through walls
   4. Create an end boss
      1. The end boss can throw a boomerang at the main character
   5. Have damage animations when the main character gets hit by an enemy
      1. Invincibility for a time after taking damage
   6. Background sound during the gameplay
   7. Life bar
      1. Life bar decreases when main character takes damage
   8. Score counter
      1. Score counter goes up when main character kills an enemy
   9. When an enemy dies, they drop a red or white orb
      1. These drops are random
      2. If main character picks up a white orb he gains back a little bit of life
      3. If main character picks up a red orb, it will be stored and added up to his total score at the end of the level
   10. Running, jumping, climbing, and shooting animations rather than just sliding around the screen
   11. Pseudo gravity to cap jump height
   12. Three shots on the screen at one time
   13. Pause menu that freezes the screen and provides options.
   14. Camera will scrollas the main character transitions from stage to stage

4. Expected Difficulty

We plan to break this project up into our baseline goal for this project as well as extra features we plan on adding. Our baseline goal includes our hero as the only sprite with the ability to move and jump. This includes a background that moves on stage transitions and, for that reason, we expect this to be a 2/10 difficulty. By adding feature A, as listed above, we expect the difficulty to increase to a 3/10. Feature A includes extra sprites that take damage from the main character but do not move or fight back. By adding Feature B we expect to add another difficulty point, making it a 4/10 difficulty. This feature makes the enemies move and enables them to damage to the main character. This entails extra conditionals to include movement and position checks. Feature C adds more difficulty because it includes the creation and movement of another sprite (a projectile represented as an oval). This projectile does damage to enemy sprites so we must also keep track of its location with respect to enemy locations. Adding features C, G, H, and J add a total of 1 difficulty point to make difficulty a 5/10. These features essentially incorporate logistics in the game like health and score. Adding sound and all animations add one point each. Sound refers to background music and animations refer to the damage animation, running, jumping, climbing, and shooting. Adding these features bring our total difficulty to a 7/10. Lastly, the boss and various other enemies require an element of artificial intelligence. This includes the enemy (or boss) following the main character and firing at him with the objective of lowering his health toward zero. Therefore, our final maximum difficulty will be 8/10.

5. Proposed Timeline

By the end of first week, we would like to have the background and the main character displaying on the monitor. The main character should be able to move and jump without the effect of gravity. No scrolling of the background yet.

By the end of week two, we intend to have the main character on the monitor with full movement capabilities along with gravity affecting the main character’s jump. We will also have a scrolling background.

By the end of week three, we intend to have moving enemies without the AI component, meaning that they will move aimlessly. The level should also be mostly complete, with the boss at the end. Animations and sound implementation will be in progress but will not be complete.

By the end of week four, we will have animations and sound. The enemies will follow around the main character. The game will be playable using a USB keyboard and VGA monitor.