

ECE 385 Final Project Proposal

Spring 2023

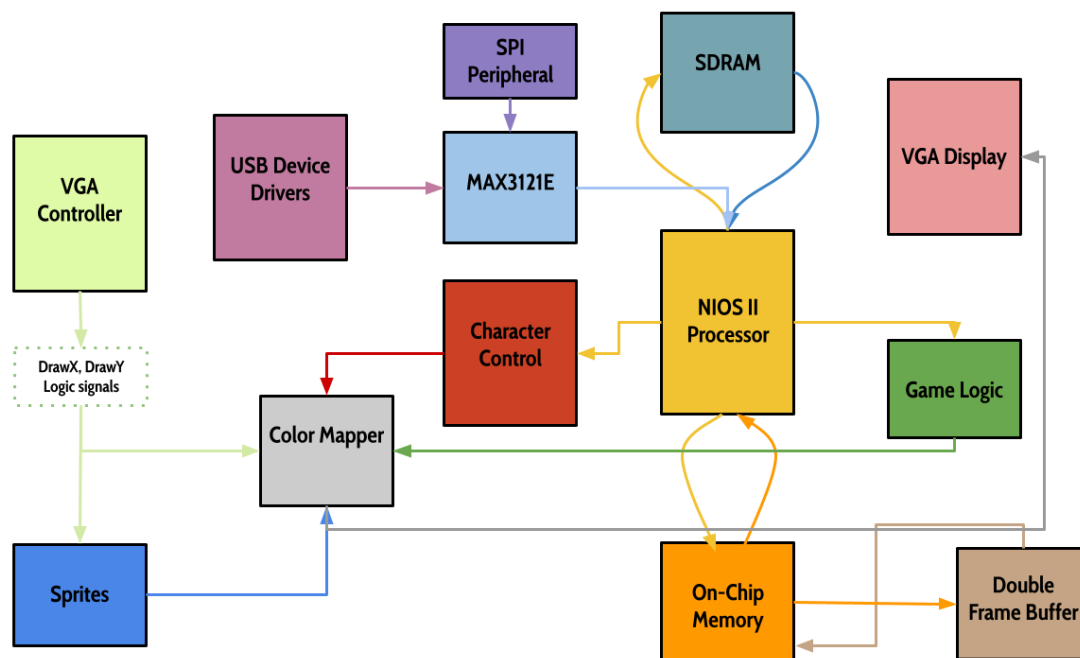
Eshaan Tibrewala (eshaant2) and Shivam Patel (shivamp6)

TA: Shitao Liu

Idea and Overview

We propose to design and implement a simple version of Super Mario Brothers, the popular NES game as a System-on-chip. Our SoC will be implemented using SystemVerilog essential components such as the System Bus, RAM, On-chip memory, video display and keyboard. Our design also incorporates a NIOS II CPU so that our FPGA can interact with the USB keyboard. Our ultimate goal is to use the keyboard and VGA monitor to successfully run a version of Super Mario Brothers.

Block Diagram



List of Features

Basic Functionality

The basic features of our project will be to have a start screen and when we press a key on the keyboard we can begin the level with Mario in the game screen. When we are in the game screen, the selected character will start on the left end of the screen, and will have to jump and move through a series of obstacles and enemies such as goombas and blocks to pass the level. We will implement this using a timer and collision detection logic. This collision logic will also allow the character to kill the goomba if Mario jumps on the goomba but loses health if he collides with the goomba in any other way. Also, we can have coins that Mario can collect as a base feature. These coins will accumulate points for the user, which will be displayed on a scoreboard on the HEX display on the FPGA. We will also add a gravity element to make the game more realistic. There will be a timer and a health bar, and the selected character will have the ability to die if health reaches zero.

Additional Elements for Difficulty

Powerups:

To increase the difficulty of our project we can add power boxes which contain mushroom or 1-up mushrooms powerups. This would mean that when Mario is “big” he does not immediately die upon collision with an enemy but instead becomes “small” again. With 1-up mushrooms we can implement a life count which will make mario respawn at the start of the level as long as he has lives otherwise the game will end. During actual game play, we can also add custom power ups that users can deploy to deal additional damage to their opponents like a fire flower or super star.

Obstacles:

Additionally, we can add more enemies such as koopa troopas or piranha plants. We can implement koopa troopas such that if we jump on them they turn into a shell and we can pick up and even throw the shell. We can add piranha plants as obstacles that will momentarily appear out of pipes if mario is nearby and then go back down. We can even have holes in the ground that can make Mario lose a life and die if he falls into them.

Game interface:

We can also add a point system that is displayed on screen so that the user can see and accumulate their score and have the score increase with leftover time on the timer at the end of the level or with the collection of each coin. If the character dies and runs out of lives, the VGA display will show a death screen which essentially tells the user to reset the game so that they can play again. Similarly, there will be a win screen when Mario reaches the end of the level which tells the user they have successfully completed the level. We can even add multiple levels to our project to increase difficulty so that the user can experience different maps and obstacles. Finally, in order to make our project difficulty more scalable, we can add a main menu function that allows the user to pick their character and level to play.

Audio:

We definitely enjoy the catchy themes of the super mario games and would definitely like to hear that when we play the game. Unfortunately, implementing audio is incredibly hard and may not be possible to be implemented within the given time frame. We definitely would like to add audio for jumping, dying, and even a background theme as the player plays the game and that would be a difficult yet fantastic addition to the project that would increase difficulty to our project.

Expected Difficulty

We believe that our baseline project has an expected difficulty of 7 points. After implementing the additional features, we believe that the expected difficulty will be 10 points out of 10 possible points. We reached this conclusion based on many factors, including the research, complexity, and programming required to implement the game. Since it is a platformer, we need to create a complex finite state machine and create multiple sprites such as mario and the koopa troopas or goombas and implement collision detection and items such that the game is implemented accurately and smoothly. Incorporating music and sound effects will also increase the difficulty because this will require hours of research and debugging because we have not generated sounds in any of the labs this semester, so this is a completely new element that we have to learn and implement. Ultimately, we believe that our overall project has an expected difficulty of 10 points because there are many elements to our project, including designing multiple sprites for the characters and backgrounds, generating music and sound effects, and incorporating fighting mechanics so that we can successfully implement our version of Super Mario Bros.

Proposed Timeline

By the end of the week of April 10th, we plan to have our sprites generated and successfully displayed on the VGA monitor. We also want to implement basic motion so that the characters can move around the map.

By the end of the week of April 17th, we plan to have all of our fighting mechanics implemented. We believe that this is the most complex part of our project since there are a lot of little details to account for, so this will take a whole week to properly design. This is also when we present our mid-project checkpoint, so our overall goal by this point is to be able to have Mario successfully move and die or have some sort of collision detection with an enemy.

We plan to use the remaining two weeks of the semester working on our additional features and implementing elements such as the title screen, menu screen, and adding our additional deliverables that would increase difficulty. We will also use this remaining time to figure out how to implement audio throughout our project. Ultimately, these last two weeks will be used to add all the details and elements that are not directly related to actual gameplay and collision mechanics. The purpose of these elements is to add graphics and audio which enhance our gameplay experience for the users.