Shane Lambert and Sam Horn

Abstract

An attempt to use Quartus II software including block diagrams and VHDL with Logic done through an Altera DE2-115 FPGA board.

ECE 287 Project report

Implementation of connect four through Quartus II software

Problem:

We were tasked with creating a program which interacts with the FPGA board that we have been working on for the semester and applying what we have learned in class and in lab. This project was open-ended, with the option to choose what we develop. We chose to develop a connect four game that will be displayed through the VGA output on the board, taking inputs from buttons and switches on the Altera DE2 board. Our end goal was to make the game as close to a simulation of the original game as possible.

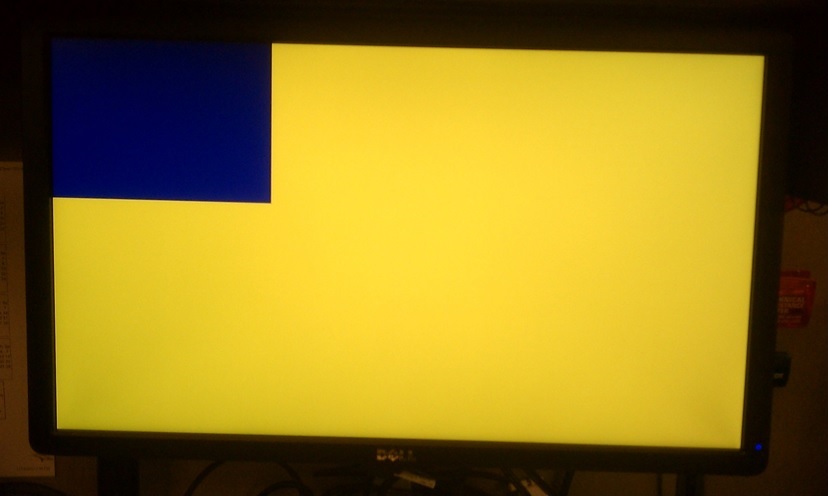
Background:



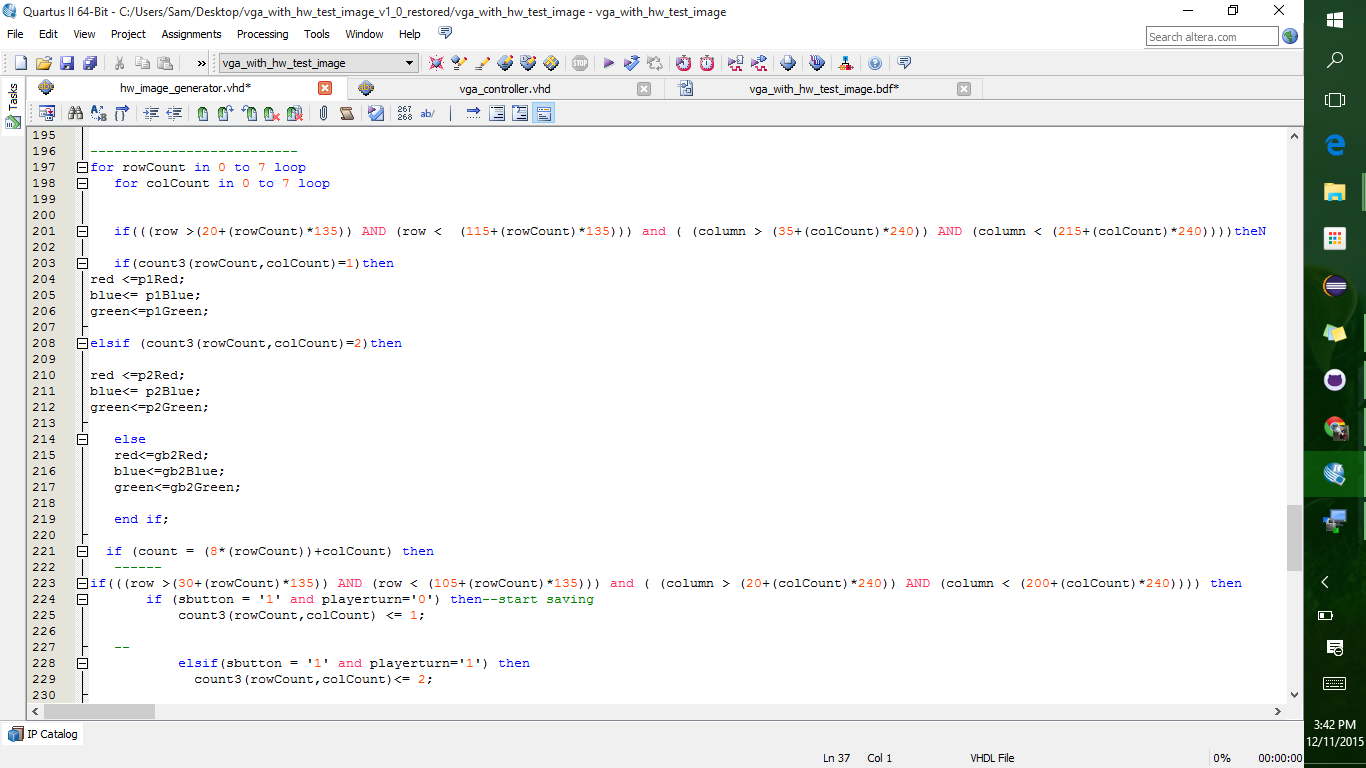
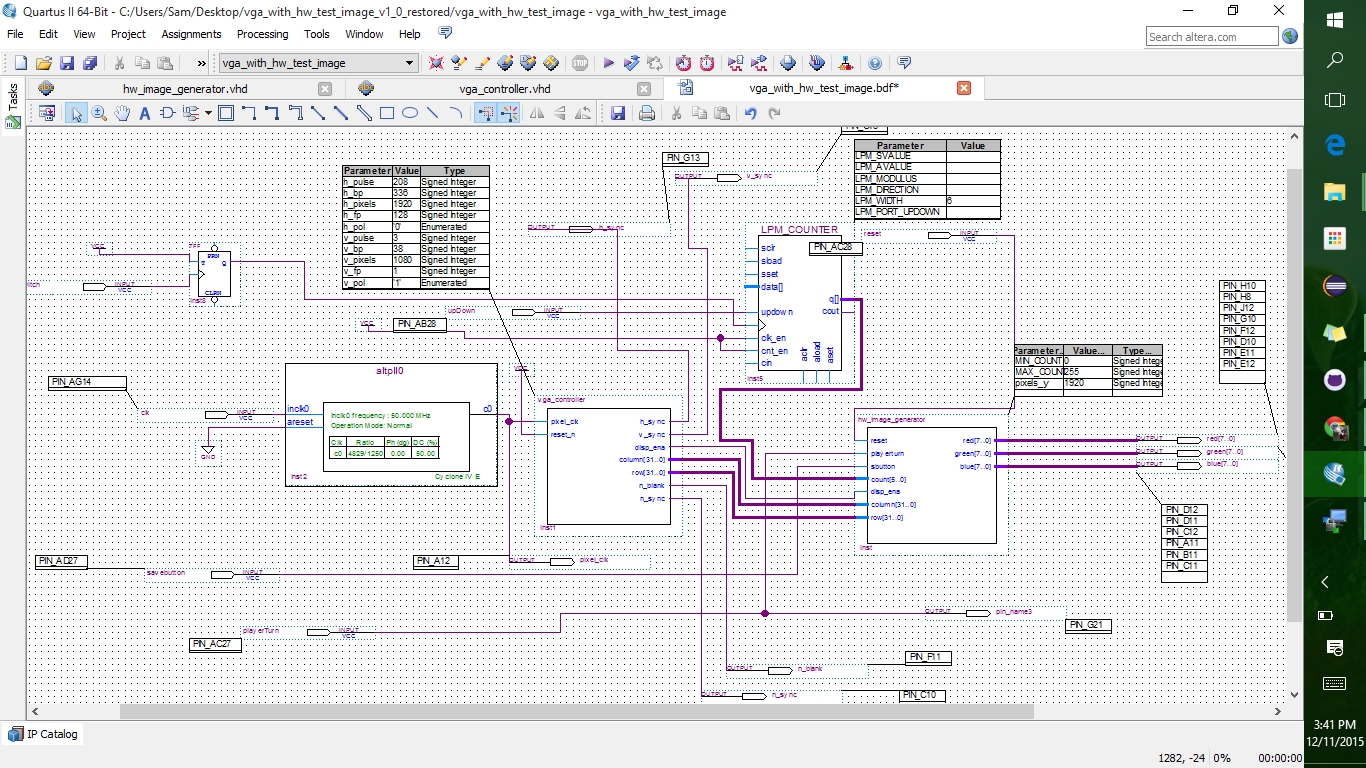
First published in 1974 by The Hasbro Company, connect four has long been an exciting game that people of all ages can play. Requiring two players, the goal of the game is the stated in the name, to connect four game pieces in a row by either horizontal, vertical, or diagonal pattern, first of the two players to do so wins. The purpose of our project is emulate this game through digital display and controls.

Implementation:

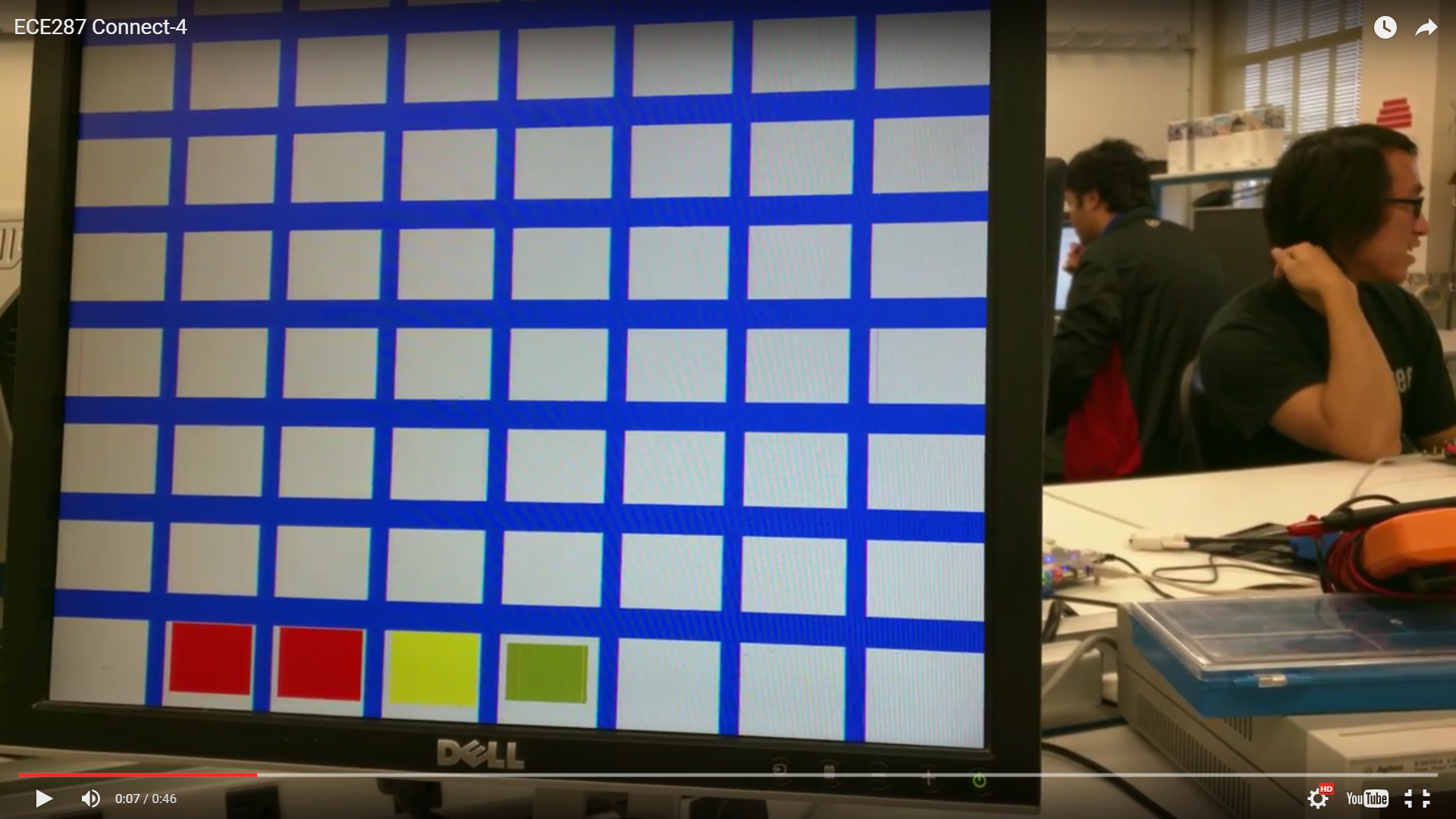
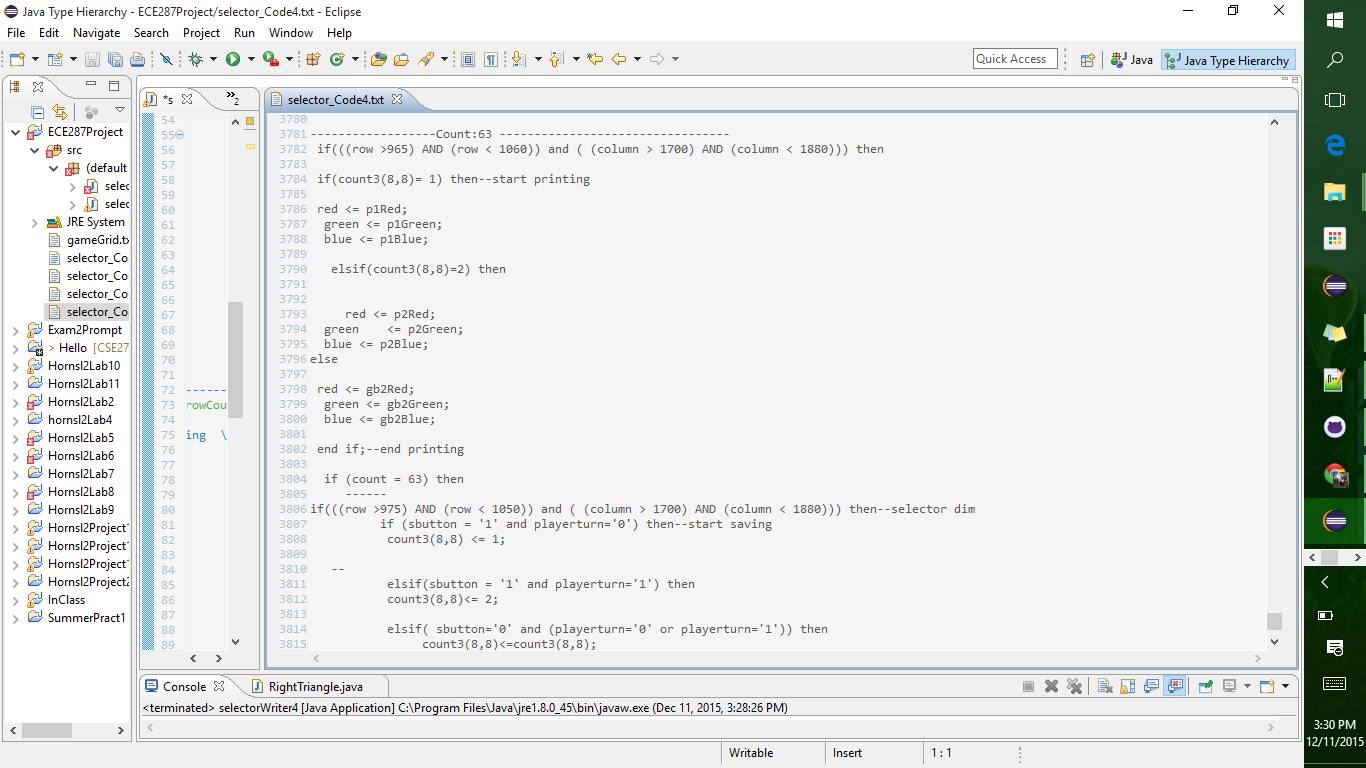
To begin implementation of this game, we first had to figure out a way to control the VGA display output from the FPGA board using either VHDL (VHSIC Hardware Description Language), block diagram, or a combination of both of these tools. Due to having no prior knowledge of how to control a VGA display, we took to the internet to find a starting basis for it, where we a page from eewiki.net that gave a great basis to build from.



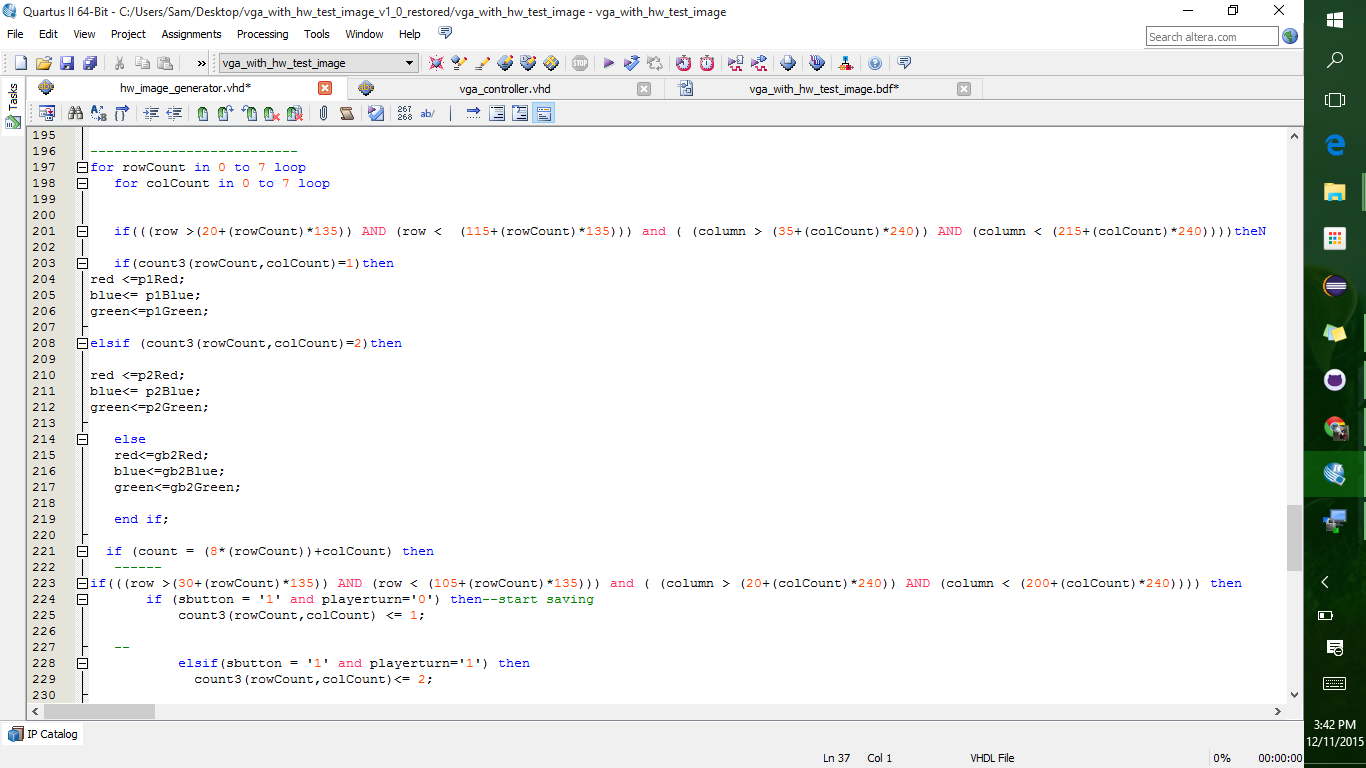
Using this as a building block, we moved along, first understanding what exactly the code is doing and how it operates, using VHDL to create blocks to put into the block diagram, controlling what was actually would be happening through the VGA output.



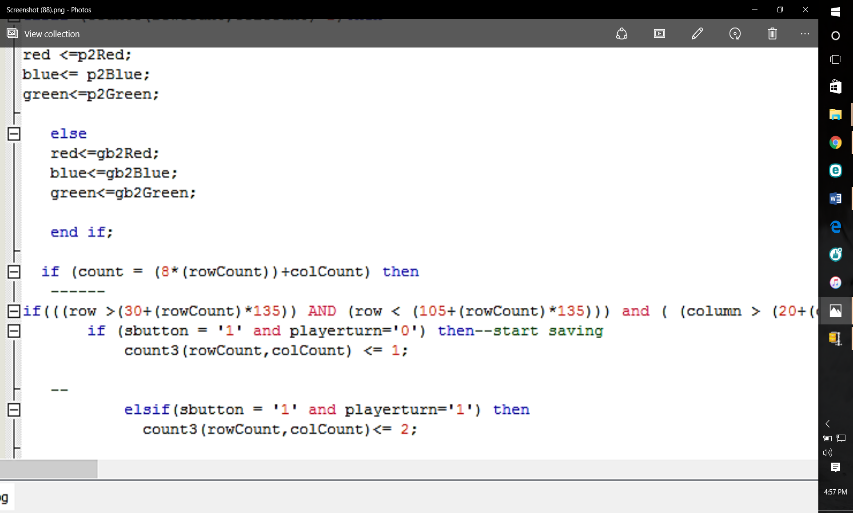
Much like any other project, we did end up encountering some issues such as not being able to display two separate background images, but we continued on, slowly adding to what was becoming the skeleton of the game. We First started by making a cross, slowly adding more and more until we got the game board we desired. Since we were unable to get the results that we wanted with loops in VHDL, we turned to java for help. To simplify the process of making the game board and to make the possibility of adding changes to the board easier, this java code utilizes a for loop that creates the set of if statements, dimensions and arguments for each individual block, therefore simplifying the process of making changes and or adding to the game board.



Creating the game:

In order to re-create connect four digitally on through the VGA output, we created the game board, but what is a game board with no game pieces, thus welcome our next task, creating a moveable piece so that the players can choose where to set there pieces to attempt to win the game. In order to do this, we needed to first create a counter that would be able to count up and or down, based on which direction was desired. We did this with based on a counter that we learned and applied in one of our Labs, and using an LPM Counter we were successfully able to get pieces to move one block at a time on the game board. Using for loops, we were able to track the location of the block on the board and give each block its own group of if statements that will set a count variable based and either add one or subtract one from based on which direction the user chooses to go. 

After successfully figuring out how to get move the blocks, our next goal was to allow the players to save their move when their block reached a spot that they thought they wanted to make a move at. To do this, we utilized another T Flip Flop that would control when the save button is pressed in time with the move button. Once through this, depending on what number the counter was on, it would change the value in a ternary variable array to either ‘1’ or ‘2’ based on which player was going. This was added to the group of if statements for each individual block so that it was based on what number counter was on.



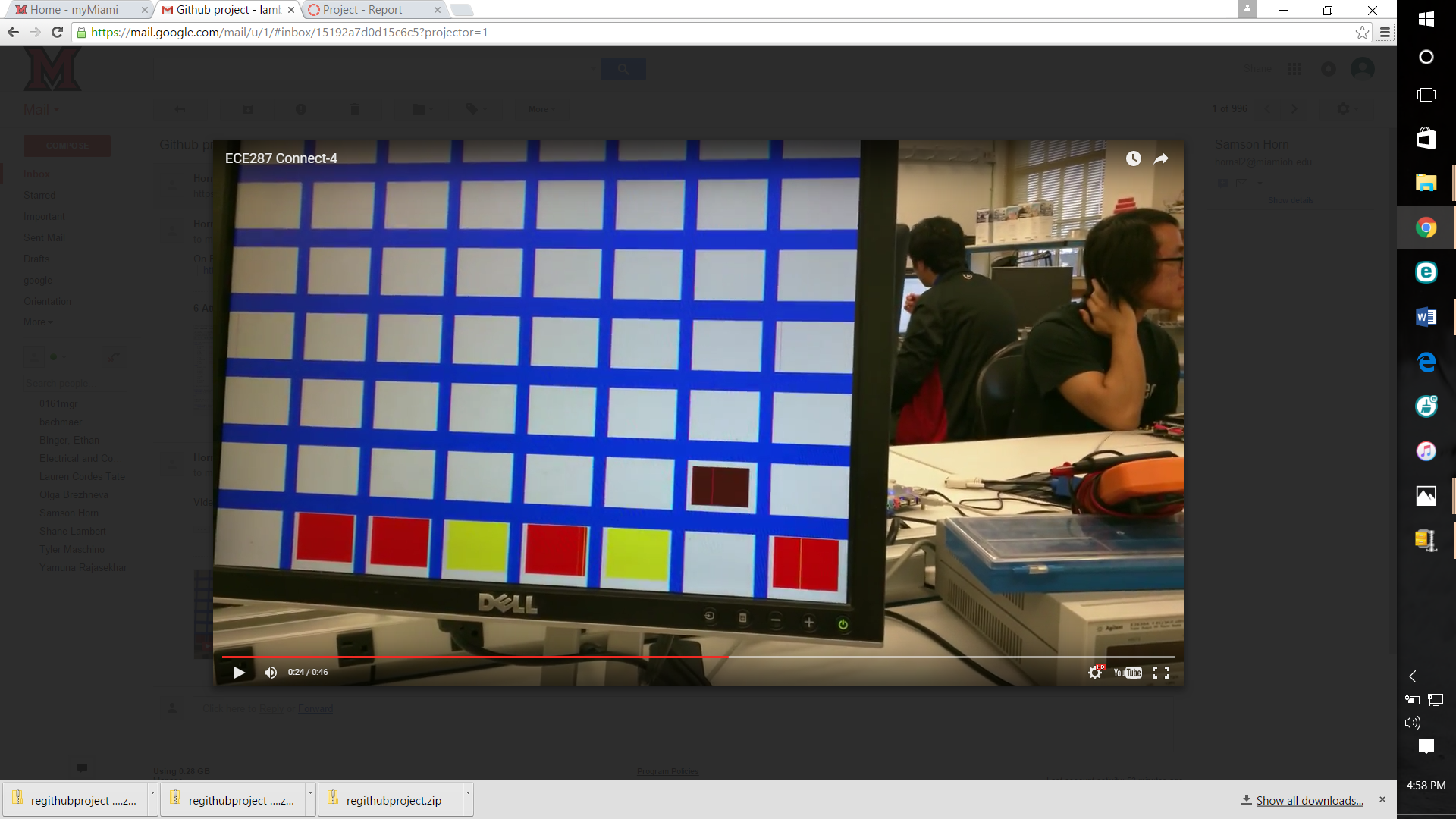
The next step after being able to save player moves was being able to clear player moves. When someone wins, you need to be able to clear the board, in the original game, this was done by removing the bottom piece that held the pieces up, in our design of the game, it is done by flipping a switch. Once this switch is flipped, it will set the ternary array that we had to save moves to all 0’s, thus removing any saved moved that was on the board before and allowing the players to start a new game.

What we couldn’t do:

There were multiple things that we were trying to do with the game that we were just unable to accomplish that were going to be able to add to the playability of the game. One of these things that we were unable to accomplish was allowing the players to play with a controller. This would’ve made the game more playable, allowing the players to simply control their moves from something that might feel a bit more familiar and ergonomic to them. While this was not as big of a focus for us, it is possible that it can still be added in later if wanted. We also were unable to automatically detect a winning combination of moves. This was have simplified the game so that users aren’t worried about trying to see if there is a winning combination on the board or not. Although this was not implanted in the original game, it would make it easier and more enjoyable for the players and user enjoyment is always a big thing when developing a game.

Conclusion:

When it came time to present our project, we were unable to present a fully function connect four game since we had some details missing. By the end, we were hoping to include automatic win detection, being able to move up and down rows, and having animations based on what has been pushed or switched. We were able to allow player movement on a block by block basis, allow them to save their moves and then also allow the players to clear the board when they find a winner. Some of the issues that we had with our demonstration of the game were inconsistent saving abilities as in that every time we programmed our code onto the board, it wouldn’t allow saving on certain blocks and possibly certain player values on those blocks and these blocks were different every time. Another issue we had were that colors would randomly appear that weren’t assigned, and we found that as more code was added, the larger the possibility of an inconsistent display in terms of the color.



Link to youtube video: <https://youtu.be/imFydPm7q8M>

Resources:

Where we got our source code:

Larson, Scott. “VGA Controller” eewiki.net. eewiki, 23 Aug. 2010. Web. 11 Nov. 2015. <https://eewiki.net/pages/viewpage.action?pageId=15925278>