

|                        |  |                  |  |                    |
|------------------------|--|------------------|--|--------------------|
| Class:                 | <b>CPE 200L - Digital Circuits Design II</b> |                  | Semester:  | <b>Spring 2017</b> |
| Points                 |  | Document author: | <b>Chris Barr, Christina Cheung, Jett Guerrero</b>   |                    |
|                        |  | Author's email:  | <a href="mailto:barrc1@unlv.nevada.edu">barrc1@unlv.nevada.edu</a><br><a href="mailto:cheunc1@unlv.nevada.edu">cheunc1@unlv.nevada.edu</a><br><a href="mailto:guerrj1@unlv.nevada.edu">guerrj1@unlv.nevada.edu</a> |                    |
|                        |  | Document topic:  | <b>Final Lab</b>   |                    |
| Instructor's comments: |  |                  |  |                    |

**a.) The Goal**

The goal of Soccer Blocker was to implement the NIOS flash from the DE2 board in order to read our C Code to generate the pixels onto the screen. By using Altera Monitor and perl, we were able to access the board by using the hexadecimal addresses on the board.

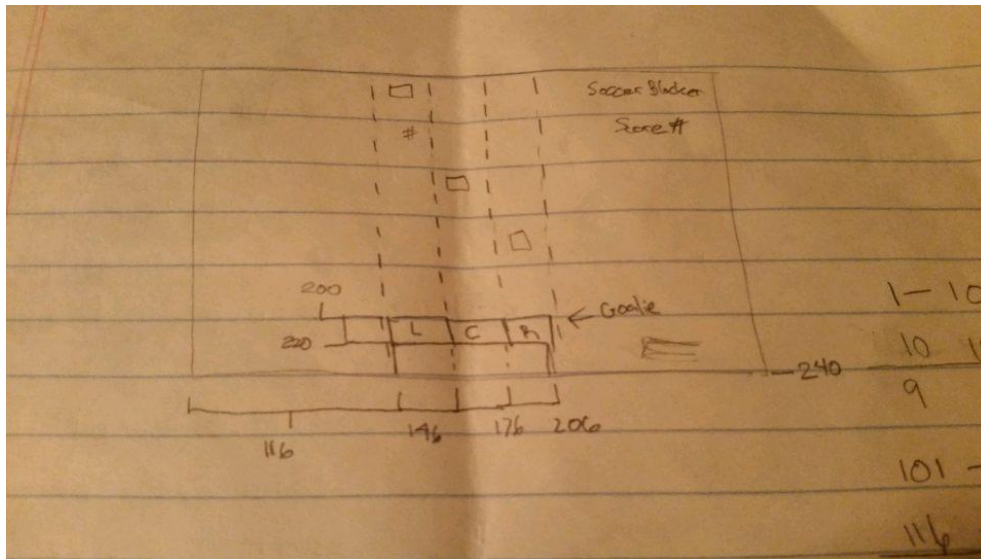
**b.) Roles of Each Member Taken in the Project**

Each group member took part in writing the code for Soccer Blocker. We all took turns implementing different ideas in the code. In addition to this, we all participated in constantly updating the code in order to debug certain issues in the game as well as collaborating to find solutions.

**c.) Background Theory**

In theory, Soccer Blocker is an idea that we developed that is built upon the desire to use C code language and the DE2 board's switches and pushbuttons as a controller to create an interactive game. Our focus was to create a game that was simple enough for anyone to play but still captured the player's focus throughout the game. The player's objective is to move the goalie to the left, middle, or right of the soccer goal in order to block as many soccer balls as possible from entering the goal. The more soccer balls that the player is able to block, the higher their score will be.

#### d.) Diagrams



This is a picture of the initial diagram that we used to create our project. This diagram shows the placement of the objects on our screen and how our group wanted to display them. We designed our program to output these objects to the screen as they are shown in the diagram above. We also used this diagram to calculate the measurements of the pixel resolution for outputting the products of our C code to the external monitor/screen.

#### e.) Circuits Operation

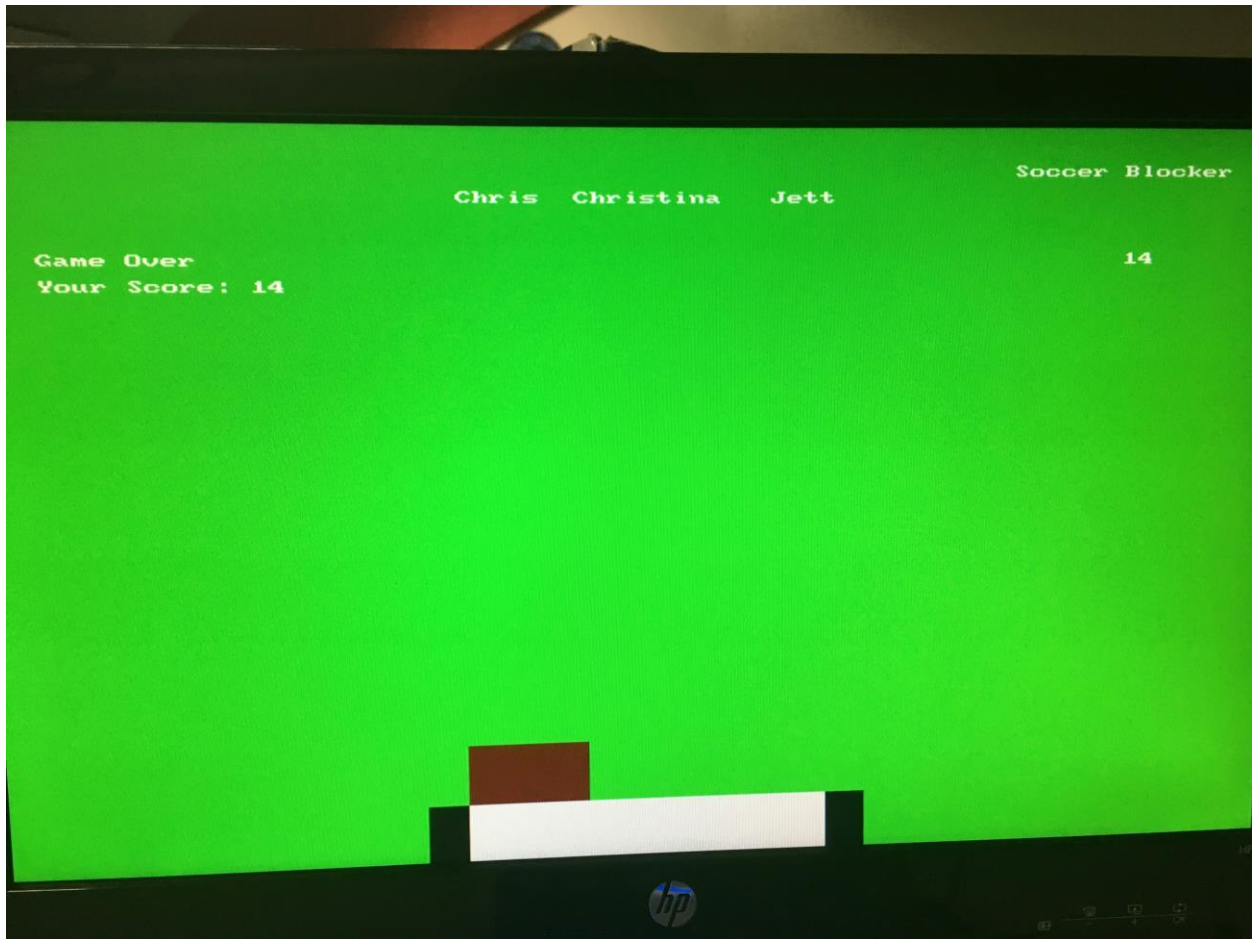
A majority of our design was created through C code so we did not pin any specific switches or buttons through an application such as Quartus. The way we accessed our desired switches and/or buttons was through hexadecimal addresses. Afterwards, it was just a series of logic arguments that determined the status and order of the said switch/button to dictate which direction the code will go.



The switches that we used are SW[0] and SW[1]. SW[0] was used to start the game, and SW[1] was used to pause the game. When switch SW[0] is flipped to the on position, it will continuously go through a while loop within our code until SW[0] is set to false, which is the off position. Likewise for switch SW[1], when flipped to the on position, it will continuously go through a loop within our code until it's turned off.

Because the DE2 board, by default, uses the 50 MHz oscillator, Altera Monitor is running our code at a fairly fast rate. Because of this, we were able to use all the pushbuttons in KEY[0-3] in if statements to determine if the status had changed or not

#### f.) Altium Output (if applicable)



This is the output from Altera Monitor using our code. We were able to output our C code onto the screen by connecting a monitor to our DE2 board with a VGA cable. The resolution was set to 320x240 pixels. We are outputting color by using 0xRGBA color code. We are also outputting the text, and other generated pixels, onto the screen using a multitude of for loops to maintain them. However, this came to be a problem due to pixels trying to overlap each other which caused screen tearing and flickering. The solution we came up with was to not involve too many pixels to be generated in the same spot.

#### g.) Conclusions

This project was a difficult program to implement using the Altera monitor program to write our C code, but we were able to connect it to our DE2 board. It was challenging to find a way to output our C code to the monitor, but we were able to find an online source to help us generate pixels onto the screen. We encountered many complications with our program, but we were eventually able to solve them all before the presentation day. This required a lot of debugging by continuously going back and forth through Altera Monitor with the process of compiling/loading and downloading it onto the board. Although this project took a lot of time and effort, we felt accomplished by the product we were able to produce in the allotted time available to us.