Class:	CPE 200L – Digital Logic Design II Lab	Semester:	Spring 2016

Points		Document author:	Reiner Dizon, Saul Mendoza
		Author's email:	dizonr1@unlv.nevada.edu mendos1@unlv.nevada.edu
		Document topic:	Final Project

Instructor's comments:

Goal

The goal of this project is to implement our idea of a rhythm game in the DE2 board. We utilized the Nios II soft processor as the center of the entire operation. With the Nios II, we are able to use the toggle switches, key buttons, and the VGA display in order to implement our game. In order to access these components, we wrote the program for the game in C programming language and assigned pointers for the addresses of each component based on the documentation for the Altera DE2 Media Computer.

Background Theory

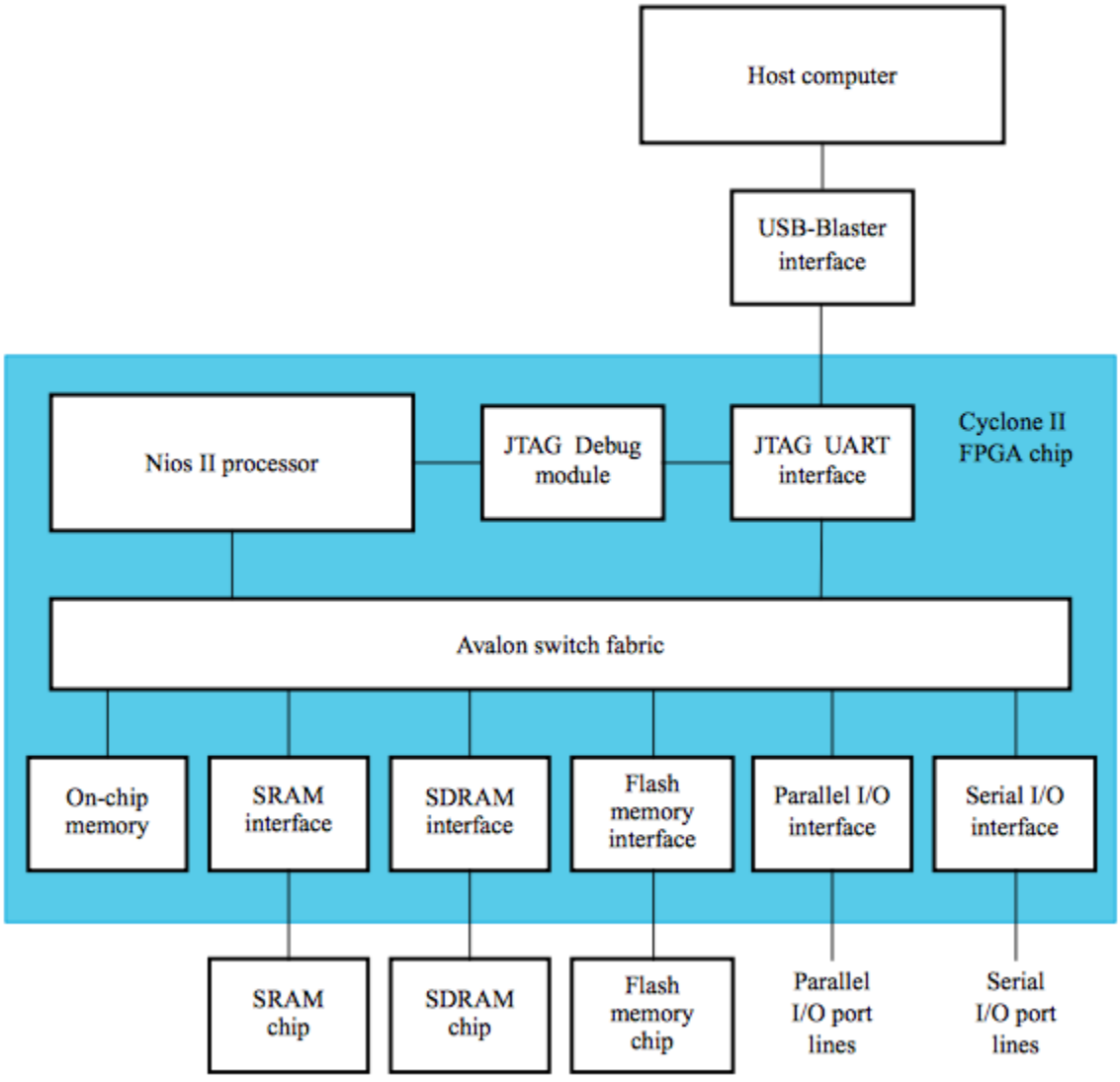
In our project, we created a rhythm based game, wrote the program to run it in C, and implemented the code in the Nios II processor. Using the DE2 Media Computer documentation as our main reference, we created pointers in our C program to store the addresses of each of the components used within the DE2 board. With another set of integer memory locations, we store the values of the pointer variables to these integer variables in order to use them for decision making within the program. Using the VGA Adapter page as our starting place for our code, we expanded these functions by creating more robust functions for creating shapes, moving shapes, and controlling input from the switches and buttons and output to the VGA.

Group Member Roles

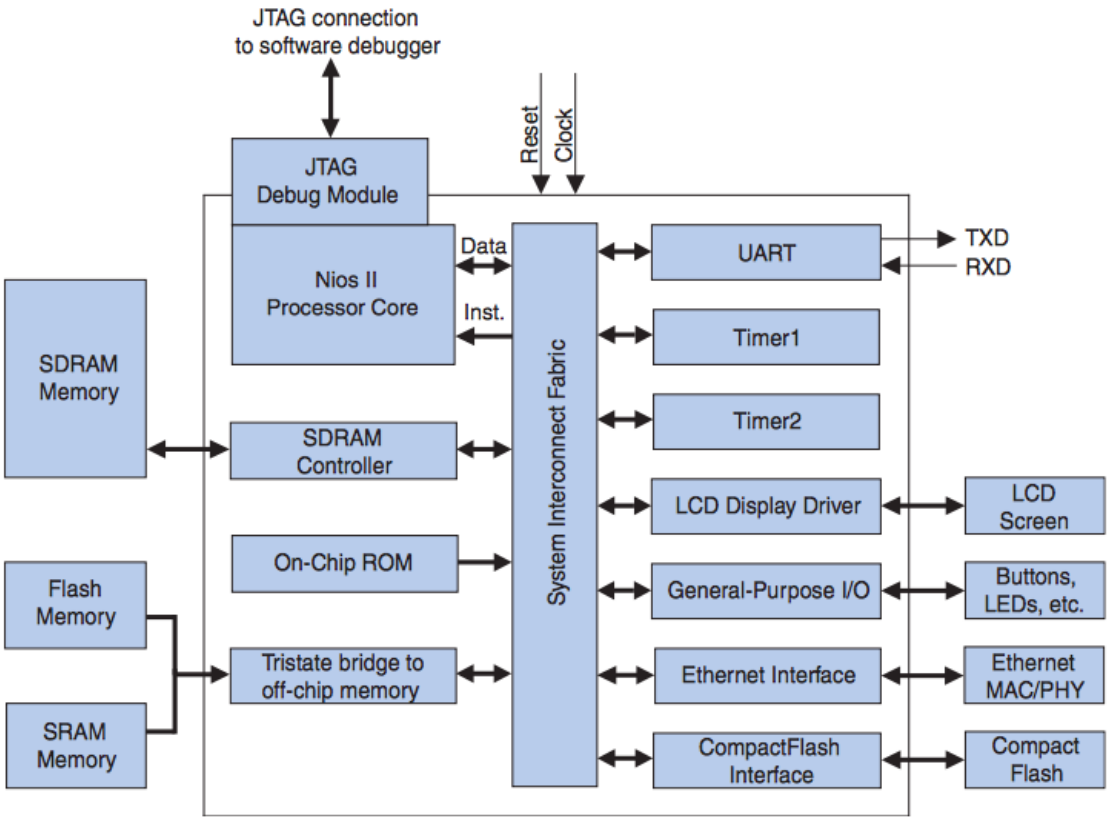
For the most part, we had similar roles for the entire project. We both brainstormed and collaboratively came up with the idea for the project. We both wrote and edited the program while learning the syntax of the C programming language. Like any programming writing process, the program was riddled with many errors in every step, particularly in the initial stages. Both of us also did the debugging of the program to ensure that it is properly implemented in the Nios II processor. Reiner's main role was to test the design for the project to make sure that the program was working before implementation. Also, he wrote and formatted most of the final project report. Saul's main role was to test the implementation in the DE2 board in order to ensure the proper circuit operation of the program and that the interfaces are accessed correctly. Also, he formatted and created most of the presentation for the final project.

Schematics/Diagrams

Since we used the Nios II processor, this is the schematic of the entire DE2 board with the emphasis on the Cyclone II FPGA chip. Within the Cyclone II chip, the Nios II processor is embedded which can control all of the interfaces in the entire board. The schematic is shown below:



In the following schematic, the mapping of the Nios II internal components are shown as well as how it interfaces with the hardware onboard the Altera DE2 board:

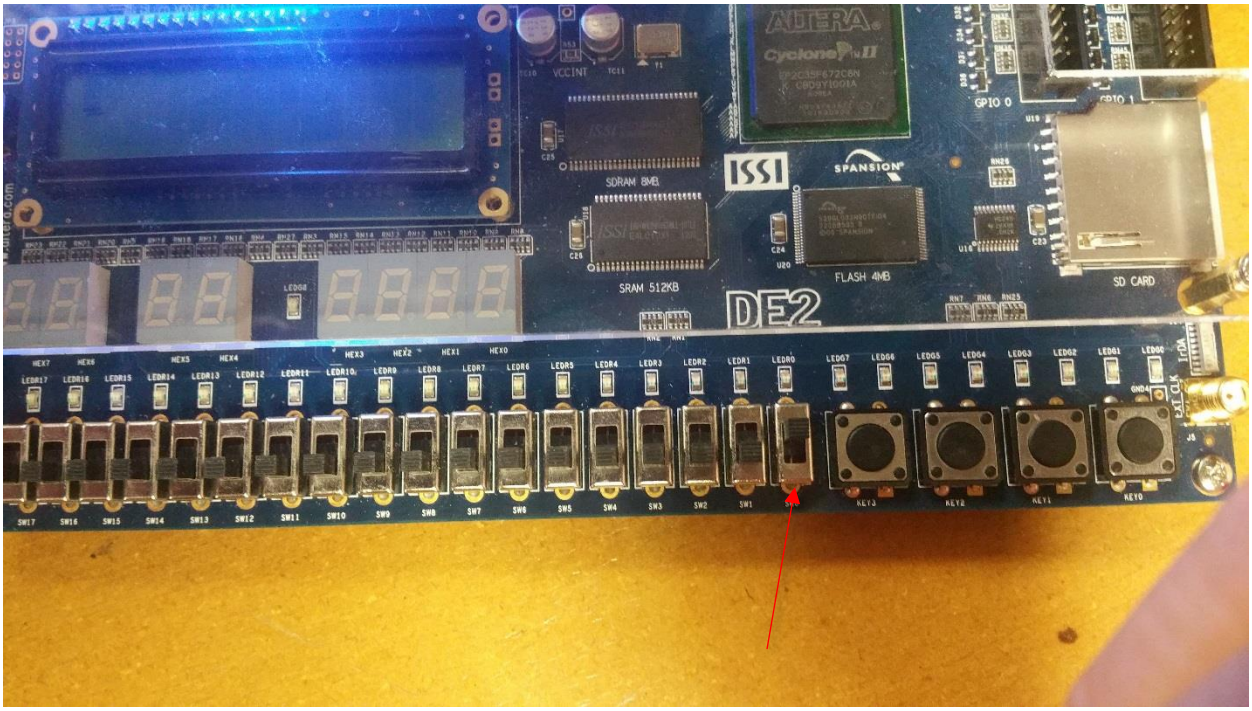


For our project, the VGA, the toggle switches, and the key buttons are accessed by the Nios II processor. The VGA is primarily the output destination of the display for the program. The toggle switches are used to set the game into action, to pause the game, or to restart the game. While the game is running, the key buttons, except the rightmost button reserved as a kill switch, are the main controls for the game.

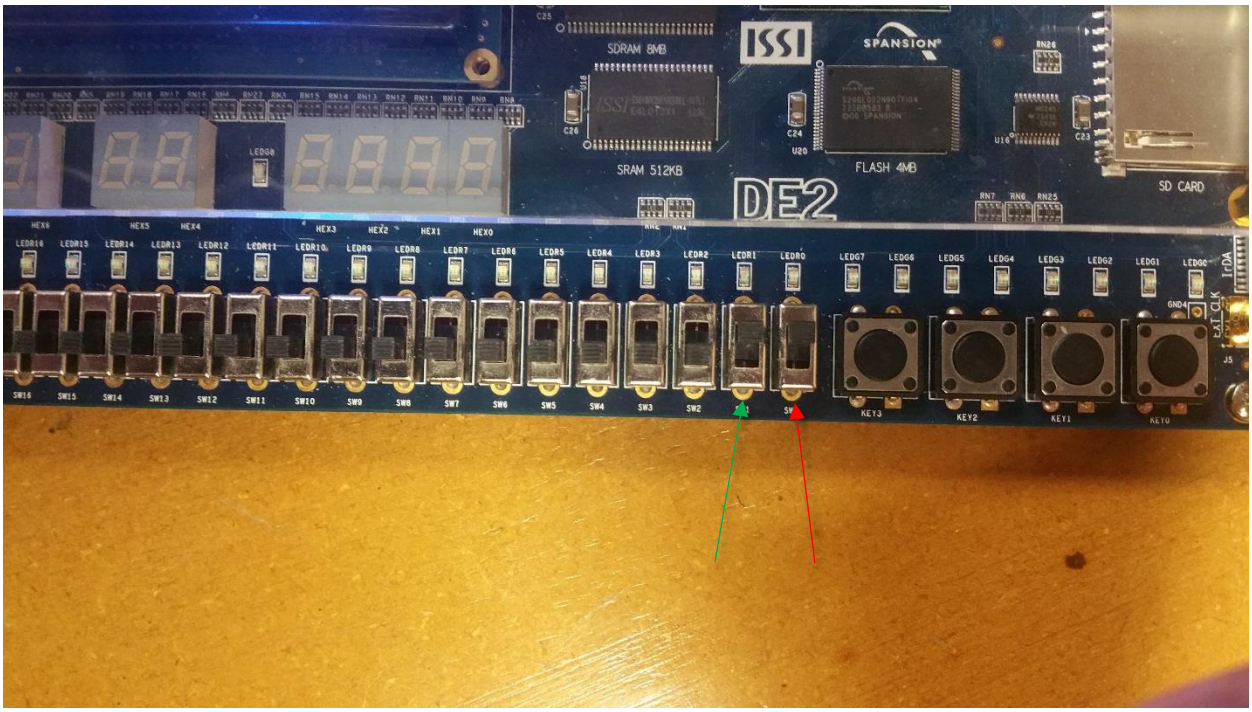
Circuit Operation

Violin Rush is a single player game where the user is able to make points by pressing the KEY pads 1-3 on the DE2 board. In order to get the point the user has to press the keypad when the block on the screen reaches the bottom of the screen. The user is also able to reset the game using the SW 0 switch and is also able to pause the game using the SW 1 switch. KEY 0 will resent the whole program.

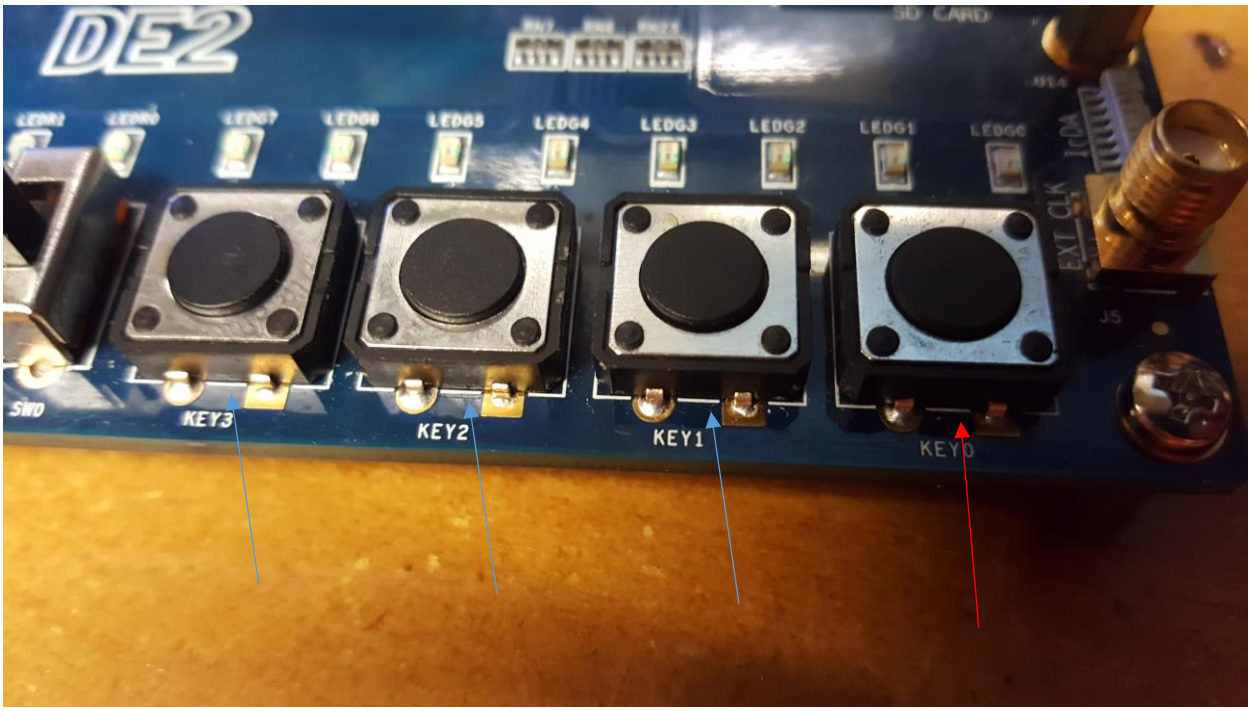
The rightmost toggle switch controls whether to start the game or the reset the game:



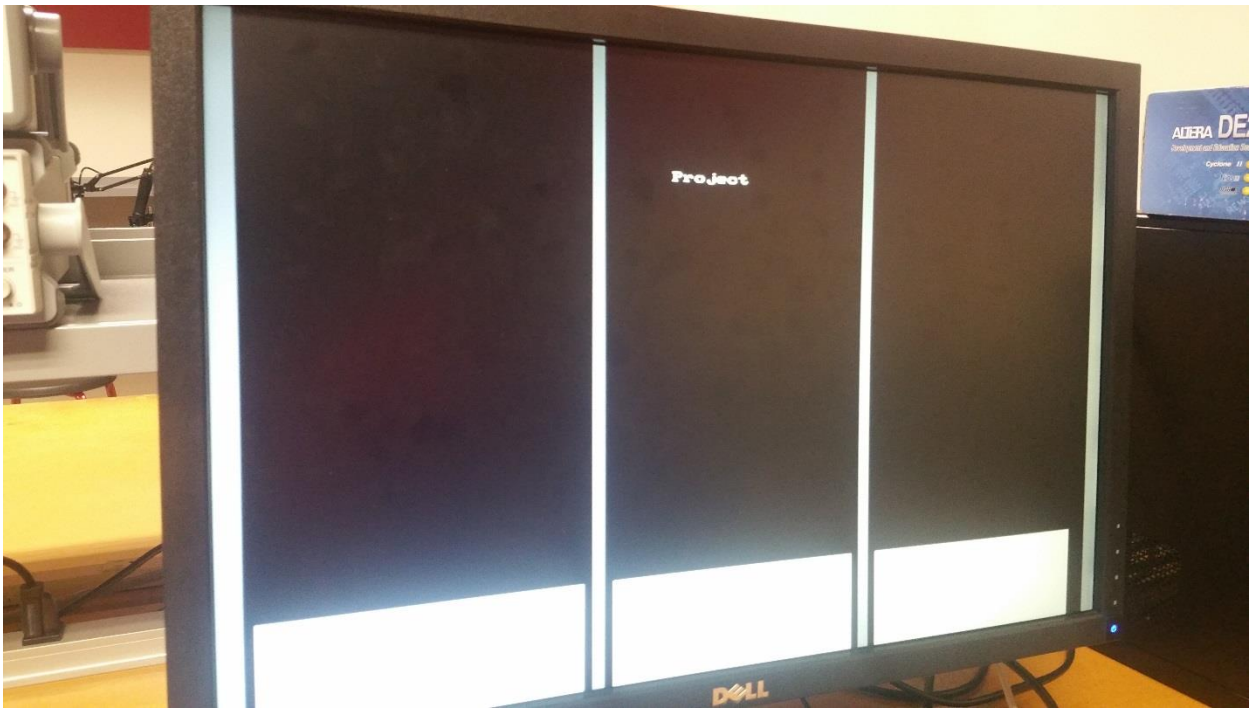
The second rightmost toggle switch controls whether to pause the game given that the rightmost toggle switch is also flipped upwards:



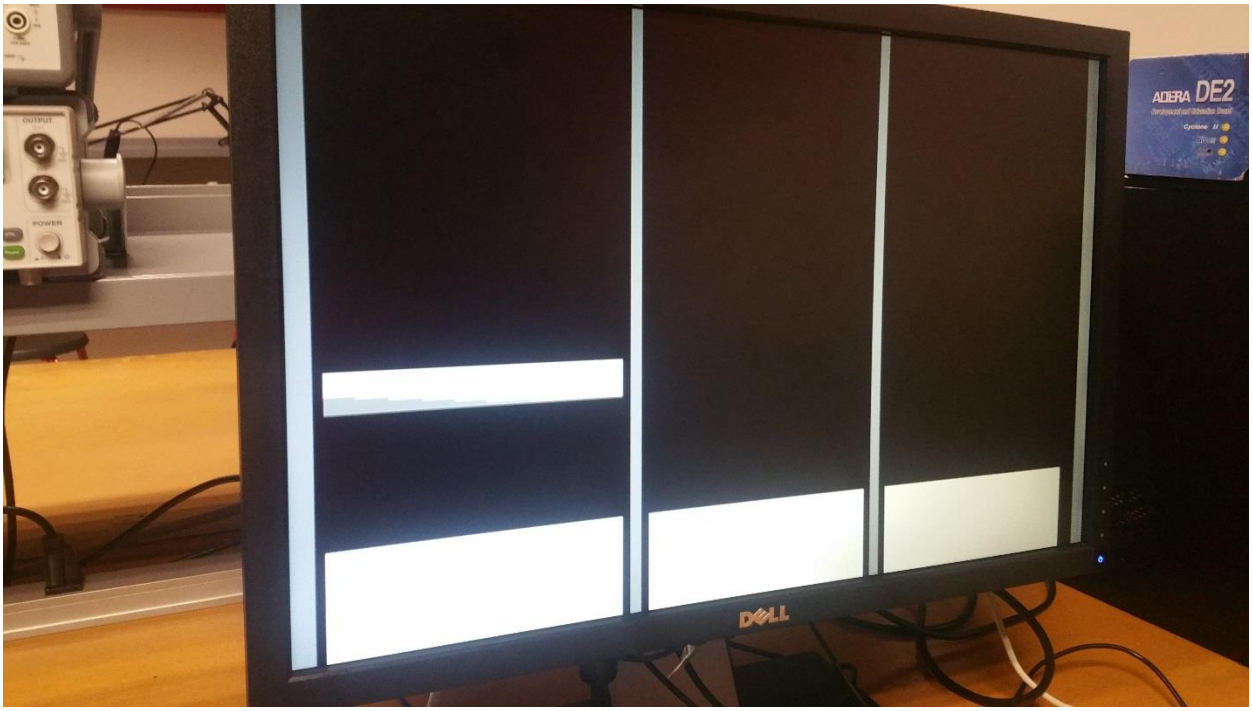
The three leftmost key buttons are the main controls for the game. These three buttons corresponds to the three bottom blocks in the VGA display. The rightmost key button is a kill switch for the entire board.



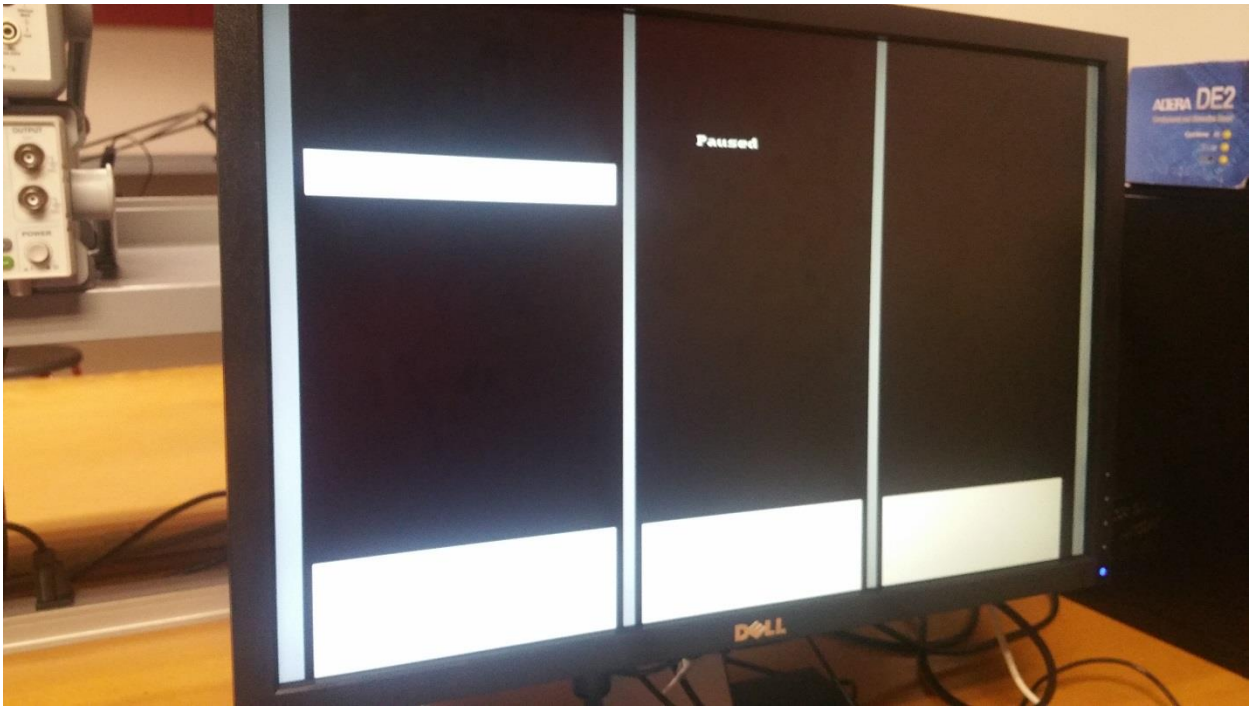
The title screen in the VGA display is shown below:



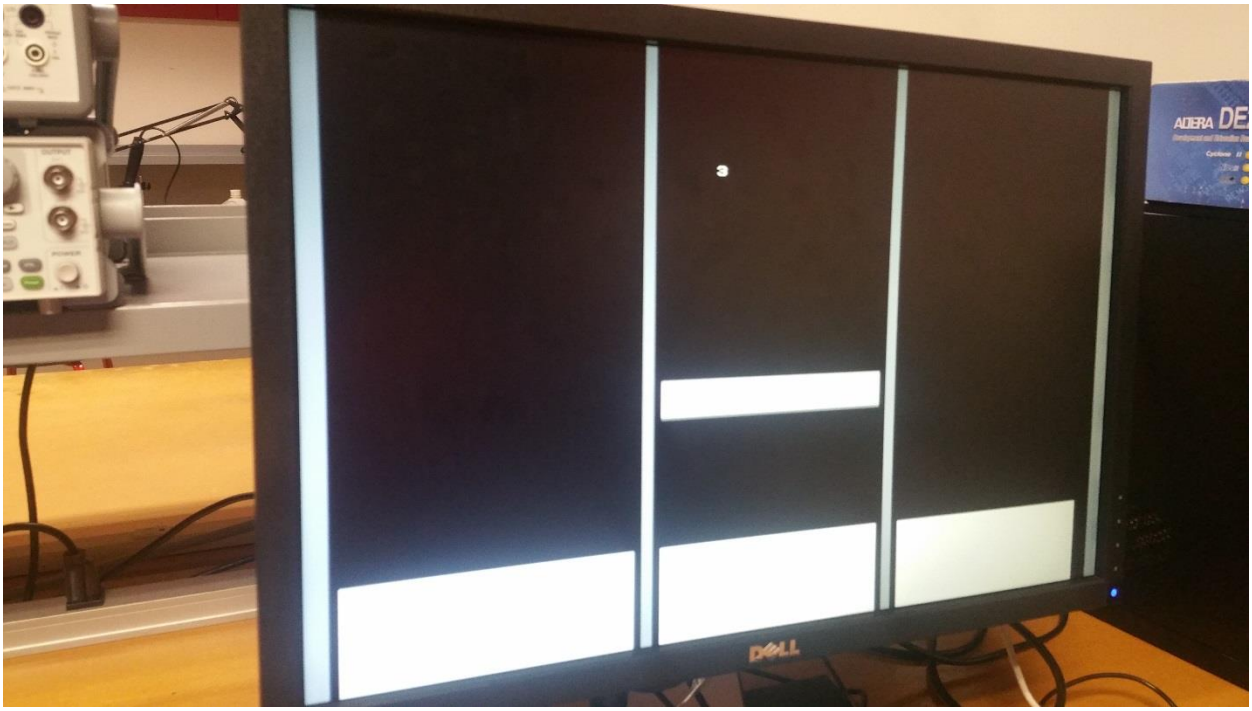
An example of the game running is shown below:



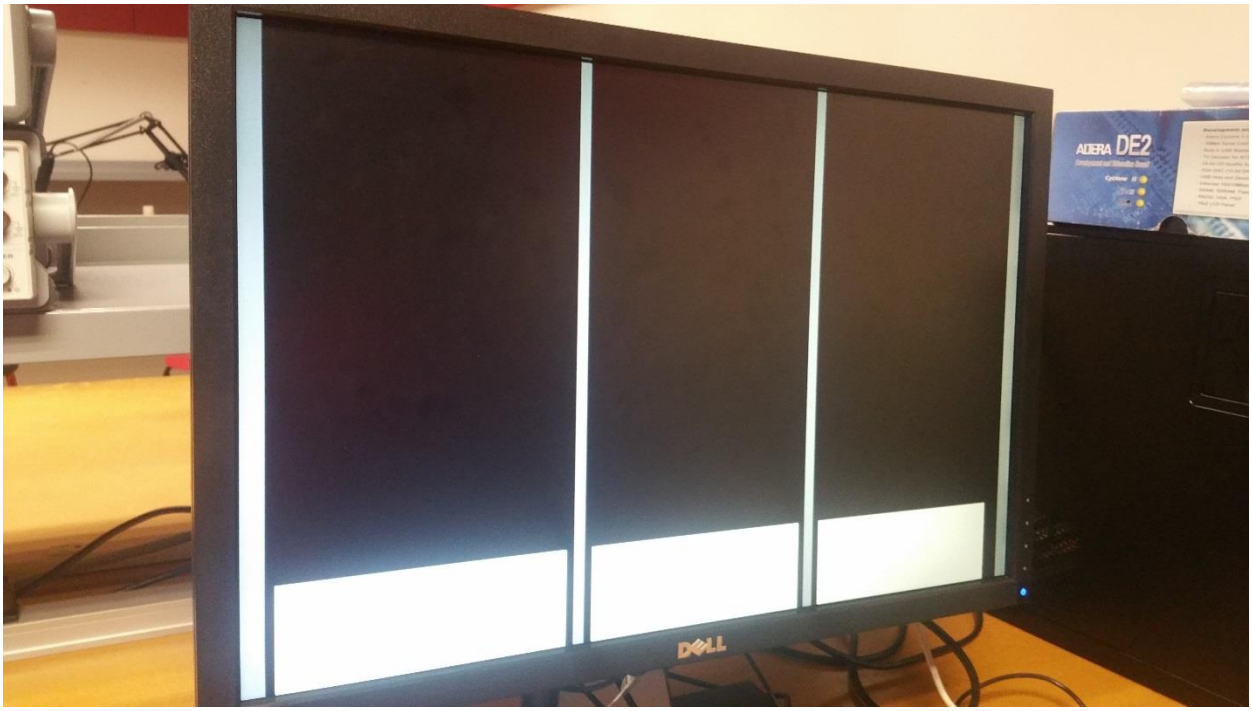
The pause screen when the two rightmost toggle switches are flipped upwards:



The score of the game is displayed near the center top of the screen while the game is running:



The restart screen is when all of the toggle switches are turned down:



Encountered Problems

As we started to look for ways to implement our idea on the LCD monitor in our lab, we encountered the problem of having to use the DE2's LCD monitor drive. In order to use this drive, we looked for ways such as using Verilog to control it. We also tried using VHDL. VHDL worked fine, however, we then came to the next problem. When we tried to add functionality to the current VHDL program we were not able to properly manipulate the pixels on the screen to print the layout of the project that we had in mind. After a couple of days of trying to implement the VHDL code correctly, we researched how to use the NIOS II processor using C language. We worked diligently for two days until we were able to not only successfully implement the layout that we had in mind: we were also able to change pixel colors and use the DE2's hardware.

Moving forward to the details of the game, our next hurdle was trying to make blocks on the LCD monitor move in a fashion that we desired. In order to overcome this problem we used a couple of "for" loops and "while" loops to control the speed of which the blocks moved and also the size of the blocks. In order to make the colors on the screen change we had to access the DE2's "KEY" and "SW" buttons and switches directly using pointers to the addresses. The main idea was to use a webcam to detect when we touched the screen that part of the screen would give us the point if the block was at the bottom of the screen. We were not able to solve this problem as we did not have enough time to research the material needed to do so with a webcam. Also, we were not able to implement sound to the DE2 board as again we fell short of time.

Conclusion

In the process of working in our final project, we picked up another programming language C in order to implement our VGA controller in the Nios II processor. Also, we learned how to access the interfaces onboard the DE2 board using the Nios II processor and used them in our C program to create the decision making process for the game. We also realized how to display pixels, shapes, moving images, and text both by accessing the appropriate addresses in the processor as well as the creating robust functions within the C program of our project. Debugging and testing were also integral lessons that we gleaned from the process of working our final project.

References

[1] Media Computer System for the Altera DE2 Board – http://www-ug.eecg.toronto.edu/msl/manuals/DE2_Media_Computer.pdf
[2] VGA Adapter on DE2 - http://www-ug.eecg.toronto.edu/msl/nios_devices/dev_vga.html
[3] CPE 200L Logic Design II Laboratory 12: Final Project - http://faculty.unlv.edu/eelabs/docs/labs/cpe200L/cpe200L_18_experiment_12.pdf