Multi-Sport Scoreboard

An FPGA circuit to help engineers keep track of their many sports games.

List of Authors (John Pak, Danijel Spasic, Mark Srbinovski, Alexandru Stoica)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: pak@oakland.edu, dspasic@oakland.edu, mnsrbinovski@oakland.edu, astoica@oakland.edu

***Abstract* - A scoreboard that has the capabilities of utilizing different scoring systems from different games was explored in this project. The idea was to use a RAM emulator from previous experiments and expand upon it with different components to display the score of two teams based on the specific sport that is selected. Many components were selected to go with the RAM emulator, which gave needed conversions for sports related data. Through many registers and hardware related component codes, the FPGA is able to display the required score of two teams on the seven-segment displays with different scoring increments.**

INTRODUCTION

To sate the most competitive of people, scoreboards were created to keep track of scores from varying sports to mundane activities users can come up with. The purpose of this project and the report is to implement a scoreboard on an FPGA board alongside many different sports to be able to keep scores of games ranging from fútbol to basketball. To utilize the power of a scoreboard in an FPGA using VHDL, a counter was utilized and coded with the scoreboard in Vivado. A counter is something that is used in most electronics to count up or count down, used as a way to keep track of rates of change or time [1]. Having a counter with this potential will be vital to the implementation of the scoreboard for the group to keep track of the scores that get inputted for each team. To get the scoreboard functioning to a pace with professional equipment, a variety of subjects taught in class were used, as follows: decoders, registers, multiplexors, and seven segment displays. Even with the many components that were studied in class, the core of this project lay upon Laboratory 5 from ECE 2700 in which students were given the task to recreate a Random Memory Access (RAM) Emulator. In Lab 5, different values were stored in different addresses which allowed multiple values to coexist at once, which is a vital component of this Scoreboard project. However, this project has more to offer than using just a RAM code, so the Lab 5 implementation was built upon while using different components and registers for different sports. This scoreboard can be used by engineering students with a handy FPGA by their side, but it can also be expanded upon in multiple places. Ideally, this project and code is best accompanied at a sports center with multiple different games that can be played, in which a user can just change their desired sport and start keeping score right away. Ultimately, with the use of different components in the project, a scoreboard was able to be implemented to keep track of different sports and count by the respective scores of each different game.

METHODOLOGY

The first design problem that had to be faced was the implementation of the code and the main body for how the functioning scoreboard would run. The group decided to look at an already existing laboratory project and build upon its success to yield the multi-sport scoreboard. Laboratory 5 from Dr. Daniel Llamocca's ECE 2700 class involved creating the aforementioned RAM emulator in which eight distinct addresses are available to store separate 4-bit data values in each address [2]. This would allow different registers to be accessed depending on the input address which would influence the game that is being scored in the scenario. Using the Lab 5 code and the theory behind the RAM emulator, the core processing power of the FPGA scoreboard was created and based around this part.

The next major addition to the circuit was to build off of the RAM emulator and add counters and registers for each gaming instance. At launch and at demonstration, the multi-sport scoreboard will have support for fútbol, basketball, and American football. Basketball and American football have unique scoring that needed to be code, such as football's 6-point touchdown and basketball's 3-pointer shot. Fútbol only increments by one point for every field goal, so this sport setting can be universal and a general case for most games, which can act as a more lenient scoring system for the user. To build this new scoring system, components that were taught in class had to be used in order for an effective implementation of the code. For instance, a binary to BCD converter is needed to convert the coding output into a suitable number for the seven-segment decoder to analyze and output. On top of that, there are registers that read the inputs and will output certain values of score based on the register code. There are currently three registers: fútbol, basketball, and

football. These had to be added to the circuit in order to analyze the scoring systems more effectively and deliver an output. However, one of the major additions to the circuit was the 7-segment serializer. This serializer was introduced in Professor Llamocca's Unit 7 notes in class, and it is introduced in this project in order to gain full usability of all 7-segment displays. Llamocca states that the serializer is able to take the one-digit restriction and display digits on multiple displays, which is four in this specific case [3]. This new serializer circuit component is vital in displaying the necessary digits for the multi-sport scoreboard as there needs to be a place for two opponents to battle out their scores and the scores need to go a high enough value for the real competitors. These new components are able to turn a RAM emulator into a functioning scoreboard that is functionable and easy to use. These new ideas yielded the datapath circuit found in Appendix A. This is the architecture behind the multi-sport scoreboard and what guided it through brainstorming and onto Vivado. The VHDL of all components was able to be precured from previous involvements in labs and from Professor Llamocca's personal website. In Appendix A, the serializer circuit is condensed into a large box at the end, but the serializer is composed as follows:

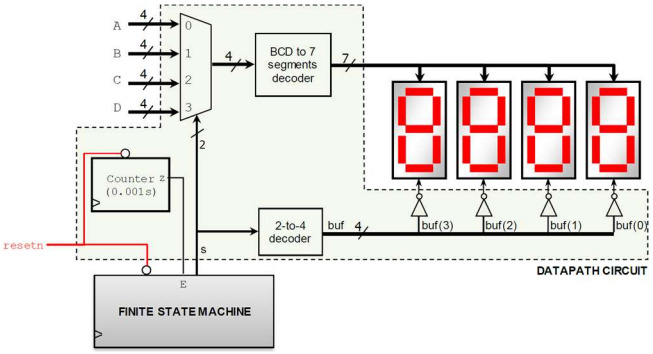


Figure 1: A serializer circuit as detailed by Professor Llamocca which allows for multiple 7-segments displays to display digits [3].

Figure 1 details how this serializer works and what components drive the important parts. The finite state machine (FSM) allows each digit to be illuminated for 1ms every 4ms, which was the coding power behind this serializer [3]. Using this component in the scoreboard project was a must, alongside with the previously discussed parts. Building off of the RAM emulator, the multiple additional parts to the circuit helped create a unique FPGA scoreboard that tracked multiple games.

Finally, one of the final components that needed to be added to this circuit was the debouncer and pulse detector circuits. The project team wanted to use buttons to increment scores in the scoreboard, which comes with separate problems when using buttons. In EGR 2800, it was taught that mechanical mechanisms like buttons can produce bad signals which fool the program. The Arduino website expertly describes this mechanical fault as follows: "these transitions [button presses] may be read as multiple presses in a very short time fooling the program" [4]. As taught, one button press can send multiple signals into a digital system which yields undesired results. This is seen in the following figure:

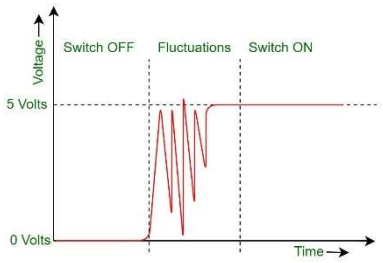


Figure 2: A graphical representation of button bouncing in which multiple signal fluctuations are recorded in one button press [5].

Once more, Professor Llamocca's code is used with permission for the debouncer and pulse detector. The idea behind the code is that it reads two inputs on a graph such as in Figure 2 to see if the signal fluctuates or not, and then it reads the input. This allows the button presses to be used in the FPGA properly, and with five distinct buttons in the scoreboard, it was the final needed piece to add to the many different components in the VHDL code.

With the code being laid out in front of the group, the next major concern was working together. First, one of the biggest difficulties faced through this project is the timing. During this unfortunate and historical circumstance of distancing away from partners, the principle of working together as a team has become much more difficult than it has become. Without the in-person interaction of the other team members, this led to a complex system of dividing the work for everyone to get a chance to work on every part. That big setback would mean that work and communication will vary significantly depending on each individual team member. However, a great project group is gauged off of their adaptability, and this group was very adaptable. Looking at online communication resources and screen sharing applications, the group was able to effortlessly continue their work as previously planned, but without the in-person meetings.

However, even because communication became resolved, this still led to the next pressing issue, which was the work at hand. With the limited resources that are being provided, the team has agreed to work off of just the FPGA that has been used with the course thus far, providing the hardware needs for the project. Alongside this new hardware limitation, by the consistent use of internet calls and meetings, the team has been able to keep up with the workload for the project. The new setup for the project that led to success was with using screensharing applications in which one person was writing the master code while others were providing feedback and research different components for testing. These new methods provided an even clearer approach to coding the project that might not have even been considered, which led to new

possible recommendations for the project that can help with brainstorming.

Overall, the methodology of this project revolved around building and coding new components around a RAM emulator while dealing with social distancing problems and limitations. This led to inventive ways to tackle the problems and display the scoreboard on an FPGA. The challenges and goals outlined in the methodology section of the report helped pave way for a better outcome and how the project was designed.

Experimental Setup

Now that the project was designed and coded, alongside came testing. This project had many iterations which were tested at different intervals for accuracy until it landed to the final project design in Appendix A. The power of using Vivado for the code rests in the ability to set up and test bench and simulate results in the software, while also programming the FPGA and testing results with the physical board. Simulating the results allows for a better dissection of the code to spot where coding problems happen for an easier debugging period. When there was an issue with how the code reacted with the project, the longest arduous process was going through the list of the code to see what issues had arisen due to the coding of the project. This becomes especially difficult when only one person can analyze the project hardware while the other team members have to help over a call and can only look at code, not with the FPGA. However, that person looking at the FPGA is able to debug it with the FPGA, and those looking at the code can use the test bench. This system allowed for a more fluid approach to debugging that played into the strengths of all group members.

For the testbench simulation, the final product is recorded and placed in Appendix B for clearer viewing of the simulation. This testbench was vital in debugging and looking for answers in what the code has issues with. This simulation was color coded for an easier reading, where both score 1 and score 2 have the same blue color, and their BCD conversion is red. In iterations where this code was not working, the 7-segment displays on the FPGA would display values, but never the right ones, or ones that were incremented properly (for instance, an increment of one would increase by two points, which is incorrect). The test bench simulation would have shown this in the score tests which led to a better result in the debugging process. The expectations were a working scoreboard with proper increments of scores, which is what eventually happened.

Altogether, the ability to debug was just as important as the coding of this final project as it led to new realizations and the completion of the code. Utilizing Vivado's simulation capabilities was a vital aspect of the project when it came to see what the multi-sport scoreboard was capable of performing. Appendix B and the ability to use a physical FPGA board aided in seeing where problems arose on the code. Overall, the experimental setup involved numerous tests which led to success in the project.

Results

Once all of the code is compiled and coded onto the FPGA board, the multi-sport scoreboard successfully works. Unfortunately, a picture cannot capture all of the FPGA with its 7-segment displays and switches without one of them being blurry or not displaying properly. However, on a Nexys A7 board, four 7-segment displays are used, where two correspond to each team. The right most switches (SW0 to SW1) influence which sport is being played, such as "00" corresponding to fútbol. The leftmost switch, SW15, changes the score input for the team. For instance, if the switch is active low and a button is pressed, it would change the score of one team. But if SW15 is active high, it would change the score of the other team. Finally, five buttons are used in the FPGA: BTNU, BTNL, BTNC, BTNR, and BTND. These are all the score keeping buttons which increment the score. These mechanical switches and buttons will impact the scoreboard and display the score as a competitive game progresses.

Looking at Appendix C.1, the datapath circuit was cropped within the blue dashed lines in order to simulate this portion, negating the debouncer, pulse detector, and the two BCD converters. The purpose of this was to simulate a "real-life" scenario and look at a one-cycle pulse. The simulation in C.2 was a resultant of this limitation, where one reset period was taken as one cycle. The different button combinations can be mapped to the resultant scores of the scoreboard and it perfectly encapsulates how the FPGA will behave once this code is mapped onto the board. Appendix B and more-so Appendix C both highlight the importance of simulating the VHDL code in the testbench on Vivado. Looking at the whole circuit in general in Appendix B gives a thorough examination of the code, whereas Appendix C can help debug and analyze a specific part of the code which is vital for a project like the multi-sport scoreboard.

Wrapping up the results, this multi-sport scoreboard came out as desired from the project team. Games can be tracked with an FPGA and knowledge of how to increment the score, and it works flawlessly with ease of access: this makes it user friendly. Going into the project, there was anticipation for how many components would be needed for the end product, but none of the project members anticipated a list as extensive as the one in Appendix A. Multiple components were used, some that were even at first not thought of. For instance, the button debouncer component was thought as not needed, because the use of a debouncer was left in EGR 2800. This led to the group realizing the imperfections of the Nexys A7 board, yet also admiring its capabilities. When met with a problem, there was nothing that a simple change to a code could fix. This is best seen in the case of the 7-segment serializer. At first glance, it would be enough to speculate that the FPGA board would be

able to code multiple 7-segment displays at once without the need of an auxiliary circuit, but that was not the case. Discussions with Professor Llamocca and further investigations of the subject led the group to the serializer, which was the perfect fix to the previous problem. The work and thought that went into the multi-sport scoreboard was worth the time as more was learned from lectures and about the intricacies of Vivado and the board. Had the group been able to get together, there would have been plans to turn the displays into a more visually pleasing scoreboard that would mimic a professional scoreboard. However, the current setup and code is more than satisfactory and sufficient for the group to fully understand the code.

CONCLUSIONS

In conclusion, through the hardware and software integration done through Vivado, something as simple as a scoreboard shows that there are a lot of complexities developed within the coding and execution. When looking at the official scoreboards at basketball or baseball stadiums, it can be observed that there would be a lot more complex programming going on due to the amount of LED lights that it needs to light up to present a score or a number. Not only that, but more professional equipment is now moving away to large monitors or animations on their scoreboards. Long gone are the days of large 7-segment displays in stadiums, but their effects on the people are everlasting. Unlike those professional systems, this project is trying to keep track of scores from other games which also brings a lot of challenges to this project. The process of working with a team and utilizing every member's strengths to the advantages of the team was an eye-opening look into engineering teams. As mentioned previously in the results section, this work led to a great satisfactory outcome of the multi-sport scoreboard. It has the capabilities of keeping score of up to 99 points, as was done intentionally to demonstrate this succinctly. Not only this, but this program only functions on a Nexys A7. To improve upon the product and make it more appealing to customers, the group would simply just extend the score to keep track of larger digits, which would need tweaks to the serializer. On top of this, making an interface or external scoreboard to improve upon the look of the board will make it more appealing to a customer. However, even with all these slight imperfections, the board offers great promise for scorekeeping multiple sports at once with the main RAM emulator. The group saw the need for VHDL with its many applications while exploring different components. The multi-sport scoreboard is the perfect amalgamation of different components and resources in a neat FPGA body that built a great tool for competitive sports and users.

REFERENCES

[1] "Counter and Types of Electronic Counters," Electrical Technology, 08-Nov-2018. [Online]. Available: https://www.electricaltechnology.org/2018/02/counter-and-types-of-electronic-counters.html. [Accessed: 14-Apr-2020].

[2] D. Llamocca, "Laboratory 5". Oakland University, Rochester Michigan, 2020.

[3] D. Llamocca, "Notes – Unit 7". Oakland University, Rochester Michigan, 2020.

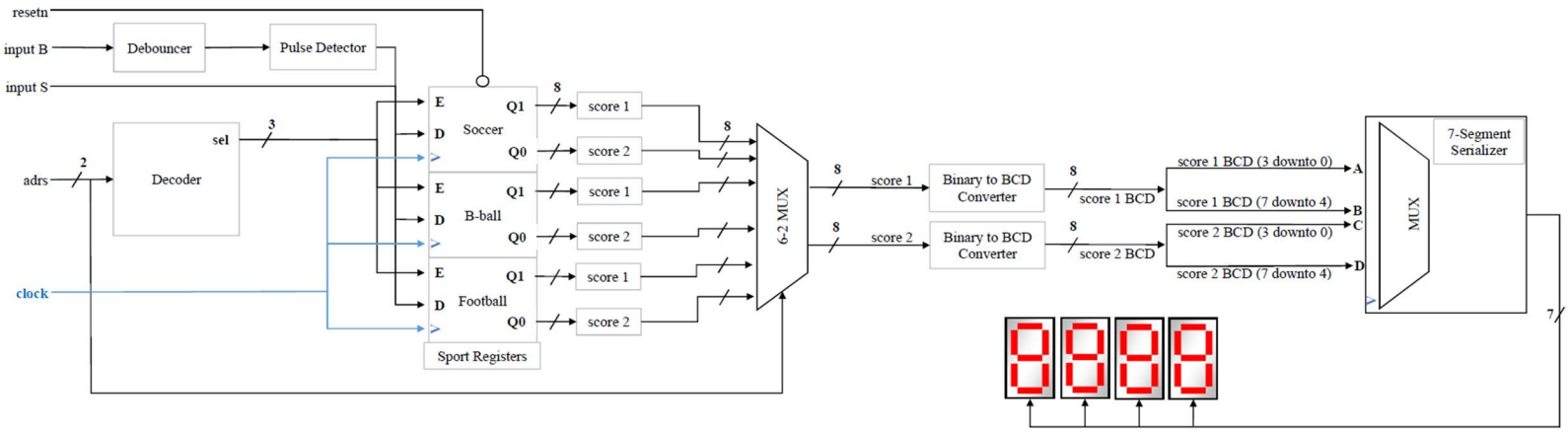
[4] "Debounce," Arduino, 29-Jul-2015. [Online]. Available: https://www.arduino.cc/en/tutorial/debounce. [Accessed: 16-Apr-2020].

[5] SujayPosam, "Switch Debounce in Digital Circuits," GeeksforGeeks, 20-Nov-2019. [Online]. Available: https://www.geeksforgeeks.org/switch-debounce-in-digital-circuits/. [Accessed: 16-Apr-2020].

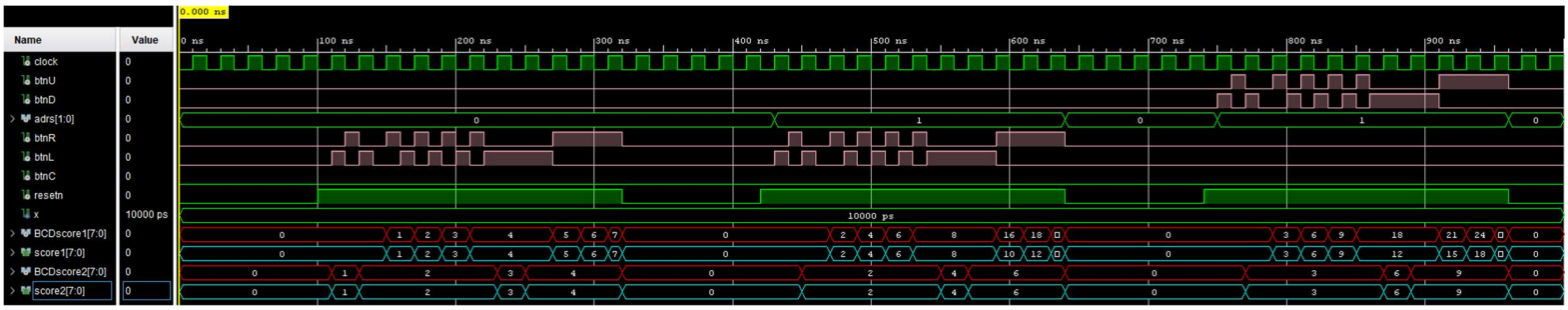
APPENDIX

The following are images that need to be placed in an appendix in the end because they are too long to fit within the columns.

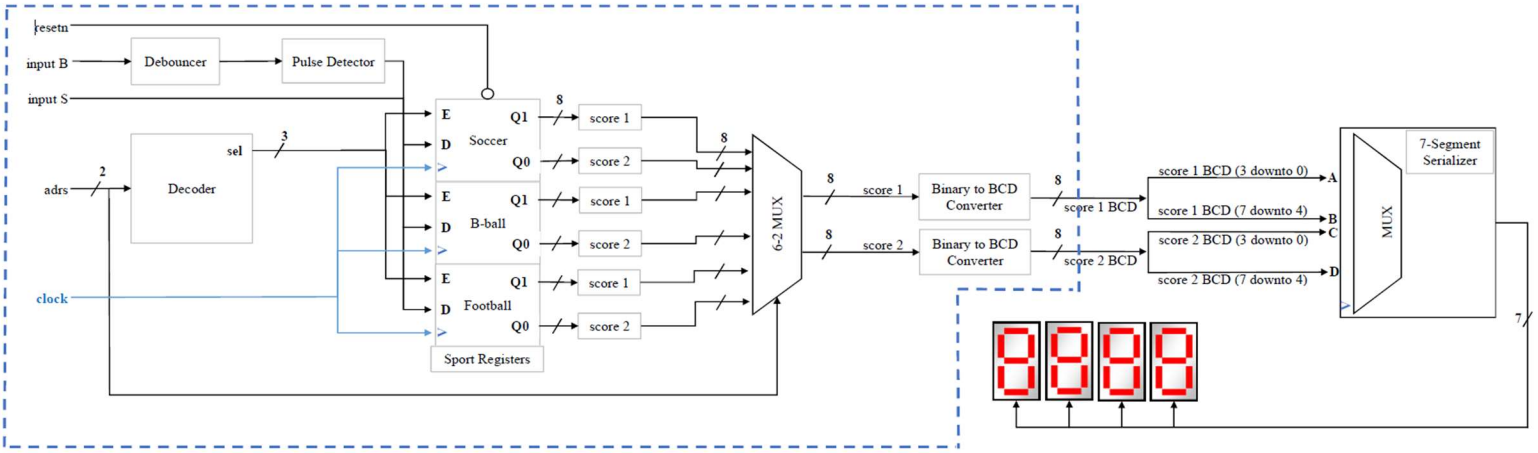
Appendix A: Datapath circuit of finalized code



Appendix B: Simulation of code without the debouncer and pulse detector, color coded for clarity.



Appendix C: Simulation of the blue area, without the BCD converters and the debouncer and pulse detector.
C.1) The datapath circuit.



C.2) The simulation.