Bradley Miller & Victoria Aguirre

Electrical & Computer Engineering

EE 354 Final Project

December 3, 2024

Donkey Kong

Abstract:

For this project, we decided to recreate the famous video game, Donkey Kong. We used the given FPGA board to create a controller with up, down, right, left, and pause buttons in order to move Mario throughout the game, avoiding obstacles (barrels). We used a VGA in order to create the visual aspect of the game, showing the platforms Mario climbs throughout the level using stairs and where the barrels are moving. Instead of having Mario jump over the barrels, we used the stairs as a way of avoiding the obstacle by climbing above it as it passes. However, if the barrel touched Mario, Mario would be reset to the beginning of the level.

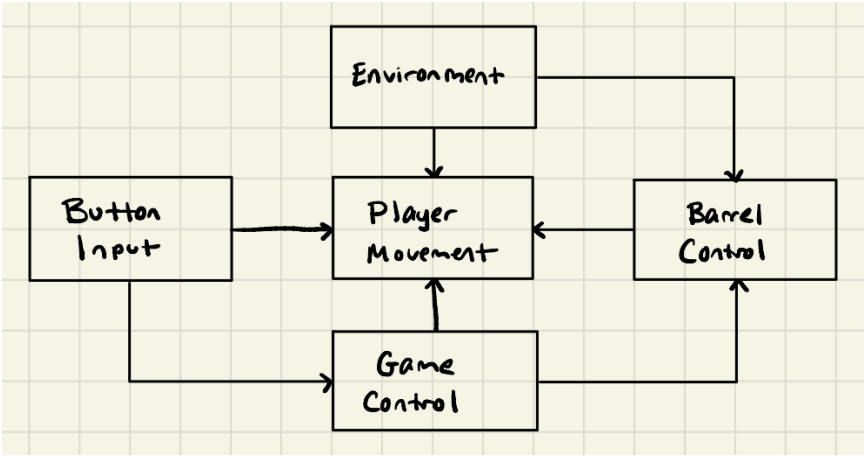
Introduction and Background:

In order to complete this project, we used the skills learned from our previous labs, such as Verilog syntax, use of the buttons on the FPGA board, and the logic from the demo code in order to build on our project. One of the most beneficial previous labs we used as reference was the “Introduction to Nexys 4 Board - Detour Signal Lab”, as it showed us how to implement a state machine to organize our logic and Verilog code. Another lab that was important in our foundation for this project was the “Introduction to Verilog - Number Lock (Part II)” Lab, as it showed us how to implement our Verilog code onto our FPGA board and important skills using state memory and next state logic. All of the labs and homework assignments we have completed

throughout the semester played an important role in preparing us for this project, and we made sure to go back and use our work in order to complete Donkey Kong.

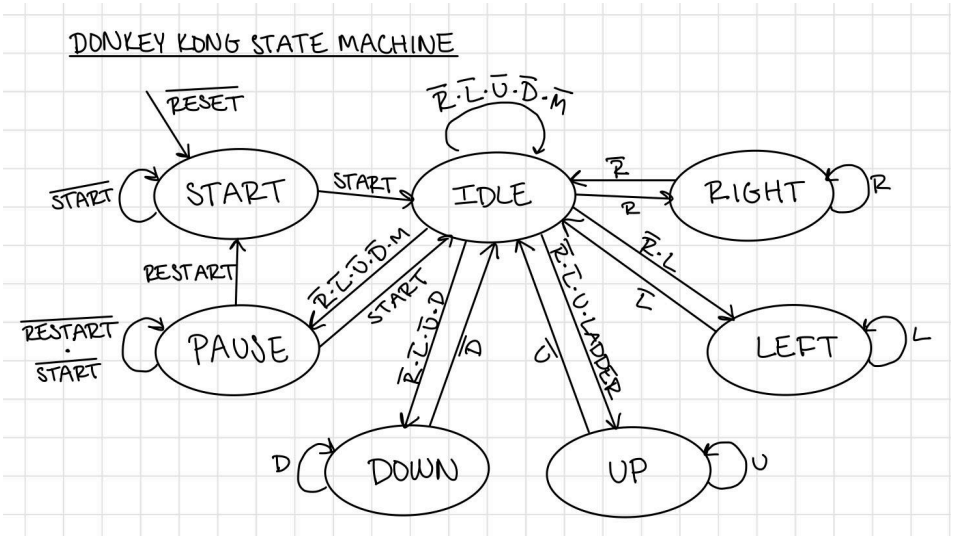
The Design:

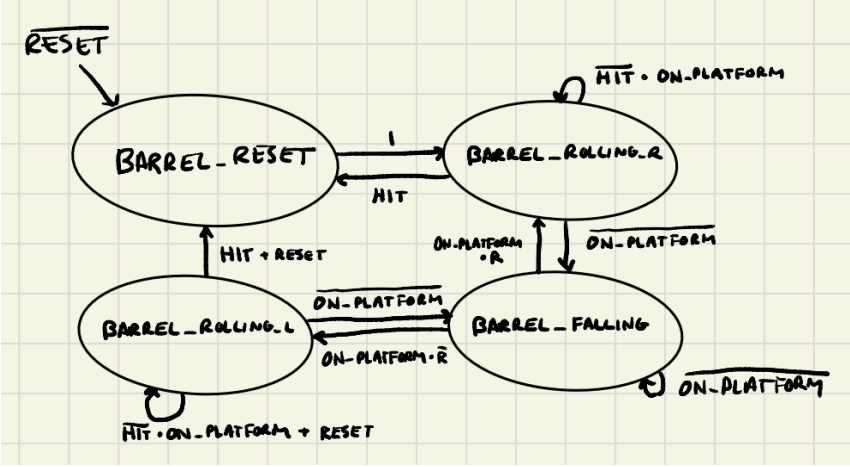
The objective of our design was to use the FPGA board as a video game controller, using BTNU, BTND, BTNR, BTNL, and BTNM to move our “Mario” pixel block up and down the ladders, right and left across the platforms, and to pause, unpause, and restart the game. This served as our button input, which would first take in the input created by the user on the FPGA board. Upon receiving the input, the input signal would move either to the player movement or the game control. This would then either pause/unpause the screen or make Mario move throughout the screen. The game control block then also affects the barrel control as pause makes the barrel stop moving and unpause makes it start moving again. Both the barrel control and the game control affect player movement as pause makes Mario freeze and if a barrel hits Mario, it resets its location to the beginning of the level. Finally, the environment affects both the player's movement and the barrel control as it determines where the two can move and bounds.

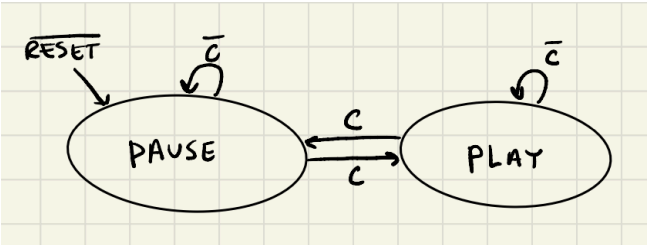


For this project, we implemented three separate state machines, which can be found below. We created one for Mario, one for the barrel movement, and one for the pause and unpause buttons. For Mario, we used BTNU, BTND, BTNR, BTNL, BTNM. If none of these

buttons are pressed, then Mario doesn't do anything and the state remains IDLE. However, if BTND is pressed and Mario overlaps a ladder, then Mario moves down. If BTNU is pressed and Mario overlaps a ladder, then Mario moves up. If BTNL is pressed, Mario moves to the left. If BTNR is pressed, Mario moves to the right. In states DOWN, UP, RIGHT, and LEFT, if their respective buttons are not pressed, they return to state IDLE. If BTNM is pressed, the game is paused. The state goes back to IDLE if the game is started again. We also created two smaller state machines for the barrel movement and pause/unpause buttons to organize our code, which can be found below.







User Interface:

The user interface for our design included BTNs and VGA. Some of the significant inputs included BTNU, BTND, BTNR, BTNL, BTNM. The significant outputs included Mario moving up, down, right, left, and the screen freezing and unfreezing with the middle button pressed. Additionally, if Mario reaches the top of the level's platforms or a barrel touches Mario, Mario's position is reset to the beginning of the level.

Challenges:

While working on this project, we faced some challenges that ultimately led us to make decisions about changing parts of our design. One of the biggest challenges we faced was tackling the jumping logic needed in order to fully recreate the original Donkey Kong game. This would have required us to create a physics design. Instead, we decided to use the ladders we created in order to climb up and down to avoid the barrels. Another challenge we faced was working with bounds. We found that in our original design, as soon as Mario hit the bound, it would still increment the pixels one more time, leading Mario to get stuck and not be able to move back. In order to tackle this problem, we would set the value to equal the maximum if the value overpassed the maximum, allowing Mario to keep moving.

Future Work:

In the future, there are some enhancements that could be added to the project in order to make it much stronger. One example would be the implementation of jumping physics logic in order to make Mario capable of jumping over the barrels. This would make the game more

similar to the original Donkey Kong game. In addition to this improvement, future students could also increase the number of barrels, making it more difficult to successfully pass the level. With an increase in barrels, additional levels could be made so that when one is passed, the player is automatically led into the next level. Finally, a debounce function should be added to the pause/unpause buttons so that there is less lag when the button is pressed.

Conclusion:

Overall, this was a very fun and interesting project to work on. I think it tied together all of the previous labs very well, and allowed us to gain further confidence in Verilog. The labs throughout the semester were enjoyable and good exposure to working with an FPGA board and writing Verilog code. Nevertheless, I think it would be more beneficial to spend time writing more of the Verilog code throughout the semester instead of receiving most of it, in order to grow confidence in these skills. I also think learning how to use Vivado was challenging at the beginning of the semester, which is why a more in depth set of instructions on how to set up a project could be helpful. The labs throughout the semester were engaging and I appreciate the amount of time given in order to complete the labs. The course provided a flexible timeline, which I thoroughly enjoyed.