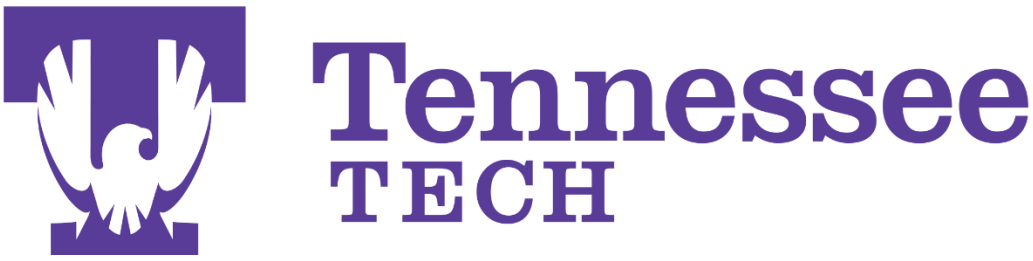

Ship, Captain & Crew

Submitted by:
JieJun Stowell, Marco Ibrahim

Fall 2020

Department of Electrical and Computer Engineering

Tennessee Technological University, Cookeville, TN 38505 USA

Submitted to Dr. J.W. Bruce in partial fulfillment of the requirements of ECE 2110 Introduction to Digital Systems

Ship, Captain & Crew is a game consisting of two player rolling five dices to obtain a six(ship), a five(captain), and a four(crew), with the last two dices representing the number of cargos. The goal of the player is to obtain the highest number of cargos point possible. Our Motivation for this project was to apply knowledge learned from class into actual practice to see how far we can get, and the project is built based on combination of logic gates. At the start of the project, we broke down the game into multiple section, consisting of the Random number generator, lock system, number of roll system, round system, score adder system, and the RGB-LED system that display the state of the game.

The pseudo random number generator we made is not a true random generator when enough observation data is taken compared to the prebuilt random generator in Digital App, however given that there is enough sequence produce by the number generator we built which make it useable. Also, while considering the cost of our build, we constantly ran into different problem which made the cost our least priority until we got all the systems working. We ended up using the combination of whole bunch of AND gates, OR Gates, NOT, Priority Encoders, decoder, and comparator, also counter to make our pseudo random number generator that would more random number.

This whole project took a little bit less than a week combine of our free time. Each designer is assigned the with different system to work on, Stowell mainly worked on making the pseudo random number generator, the roll system, and the lock system. While Ibrahim work on the round system and the score adder system, and the RGB-LED system that display the state of the game.

While working on the pseudo random number generator the biggest challenge Stowell faced was finding a way to get rid of the 0 and 7. With the help of Linear-feedback shift register on the project specification page, after drawing out the diagram and writing out the truth table to find the excitation equation, etc. The first prototype was made based on the combination only consisting of AND gates and OR gates making the zero and seven to repeat one of state between one and six. However, problem started to occur where the random number generator output was not random enough. So, three-bits decoder was used instead to break down which number was generated, if the PRNG generated a zero or seven it would send the input into a counter which randomly output a number 3 bits number between one and six. In the end five different pseudo random number generator was made using the same basic design while adding a random XOR or XNOR gate between each D flip-flops. Also, once the roll system was made, when all three rolls are used up, a AND gate is used to prevent the current player from rolling any further. The roll system was made using the combination of AND gate, OR gate that feed input into two D flip-flops and sending the next state output to a two bits decoder that send the output into an SR flip-flop so that when done or the Reset button is pressed the LED for the three rolls would also be reset to zero. At first the reset button was not taken into consideration while making this system, however knowing how SR latch work it was an easy fix. The Rolling system took the least amount of time to make since it only consists of two bits current state and 1 bit input (the throw button).

The Lock system was made using a three selector bits priority encoder that receive a 3 bit data from the Pseudo random number generator, and the priority encoder output is connected to an 2 selector bits MUX that check whether dice 1 to 5 got either a six, five or four from the random number generator, if the random number generator output a six, the six would be lock onto the dice 1 display, and a 5 would be lock onto the dice 2 display, and a 4 would be lock onto the dice 3 display, and the last two dice only represent the number of cargos for the score system. There is another priority encoder connected as the selector bits of the MUX that would out a constant value (6,5,4) depend on which dice roll a 6, 5 or a 4, and when all three rolls is used up by the player the DONE button is pressed display for the five dices will be reset back to 0, this is to prevent the current player from stealing the previous player's cargo score.

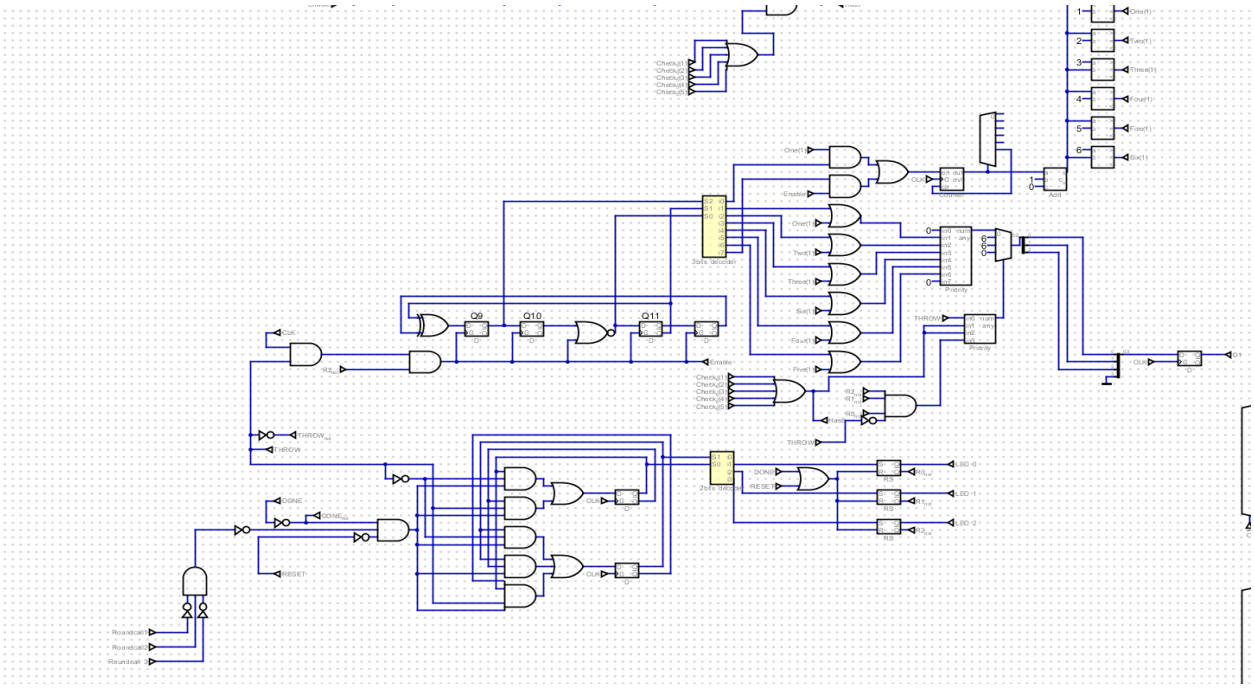

Figure1: This is an example for the PRNG, and Roll system, with the lock system.

The round system Is built based off the RGB_LED STAT which indicate the state of the game. Whenever the system is turned on the game will always be in the green state (game is ready), and there are three D flip flop that hold the different combination of input from the player, game will only start when DONE button is pressed, and a counter is used to track the number of round, round will only increase if the Done button has been pushed by both player, and the counter would the output to a decoder that check if is it round 5, and if the Done button has been pressed by the second player, and is at round five, the RGB_LED would indicate a red signal which mean the game is over.

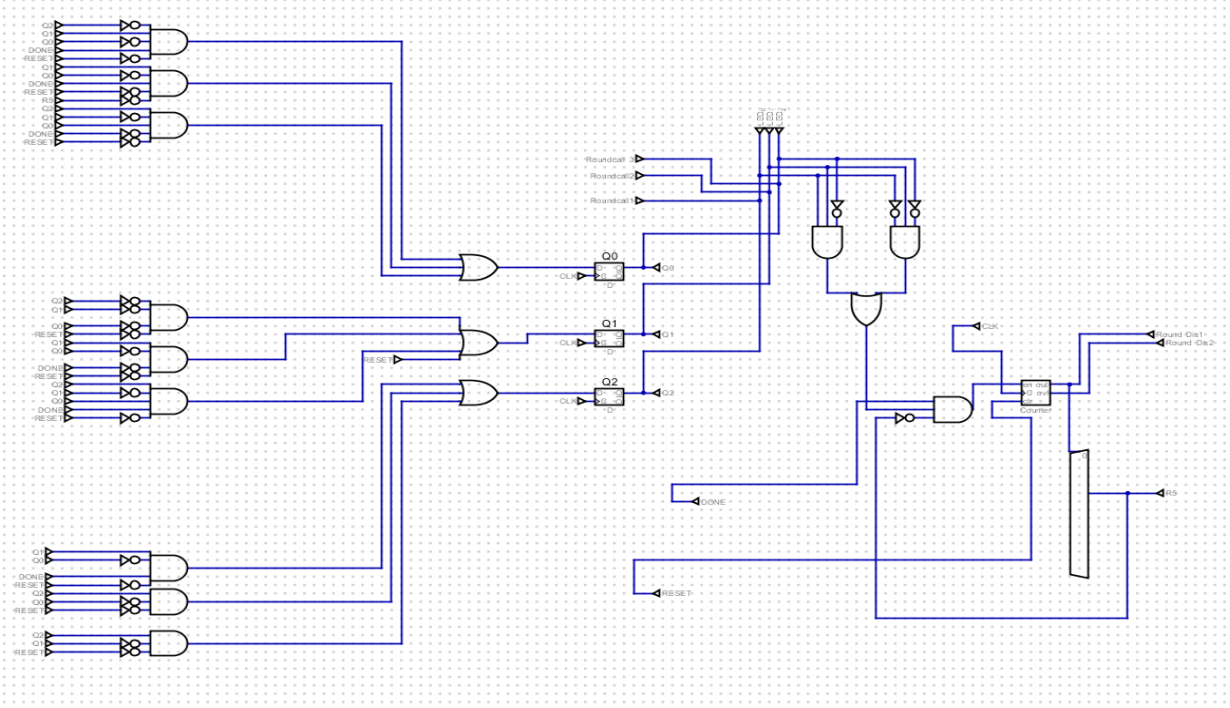

Figure 2: This is the round system, which will show the state of the round throguh an RGB_LED STAT.

The score adder system is made with whole bunch of adder and counter with couple comparator. First the score will be send to the lock system to check if a 6, 5, 4 has been rolled, if it has it will add the cargos score from dice 4 and 5 once done is being pressed by the player, and the score is send to an counter which check which the current round of the game, long as is not the 5 round and player 2 pressed the done button, score will be send to a sub adder system which check for score overflow, and move the point to display 1.

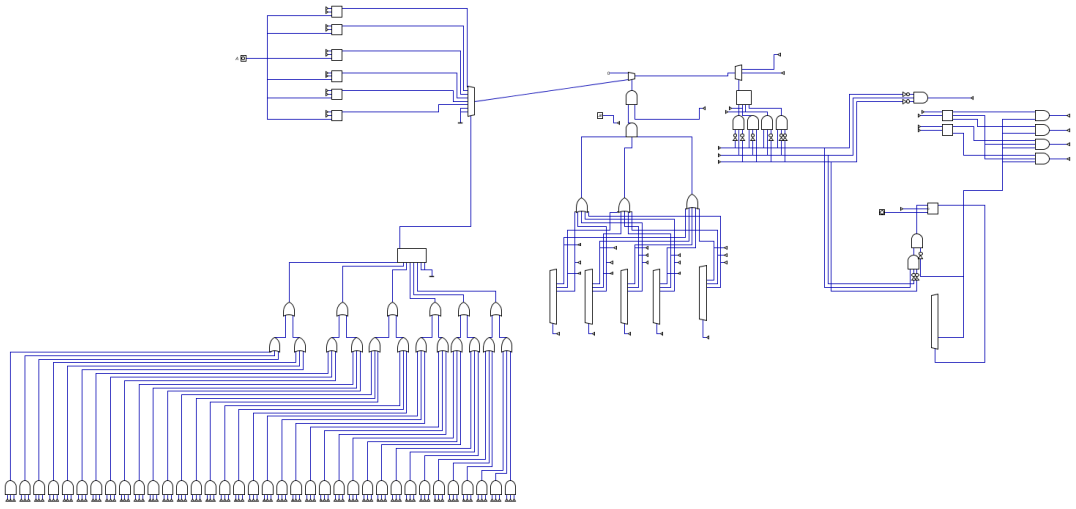

Figure3: This consist of the part of the adder system, and the fail safe for the lock system.

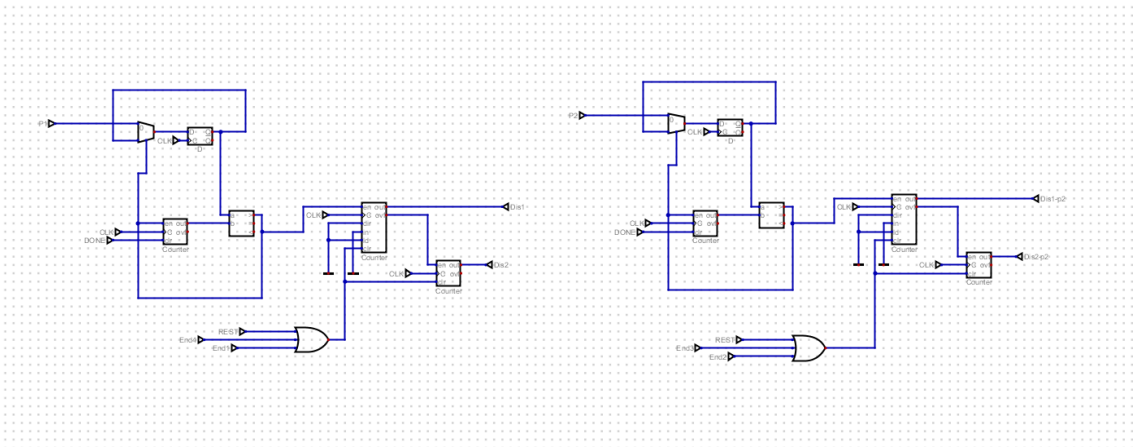

Figure 4: This is the subsystem for the player score adder system.

We didn't use the test case that is built into the Digital system app, our testing was done by running the prototype/the finish product multiple attempts to check the issues with our design. Most of the problem was able to be easily spotted without any high-level testing method. Since the rule for this project was specific, when we manually test the issues of our design we only had to check if it is following what the rules. The draw-back of doing test this way definitely consumed more time than we would like. Since our system is not hazard free the input lag(delay) is very noticeable. Also, there is a lock safe system that will prevent the system mistake occurring when multiple six and five or four is rolled, and this lock safe system contributed to the cost by a large amount.

On our first design, the whole system costed way more than what it should be, but once we got everything working and cleaned up, we reduced the cost by a huge amount. Our final design contains 270 NOT gates, 606 AND gates, 183 OR gates, 220 NAND gates, 41 NOR gates, 10 XOR gates, 6 XNOR gates. The total cost of the finish product is 4488842. I notice that depend on the situation, decoder and comparator can have the same function, since the purpose of comparator is to check what number is generated by the number generator, and the decoder does the same thing, and it is cheaper. Looking at the final cost of the system, also popularity of this product mass production would not be a good idea, even though this game doesn't take much resource to run. The product should not have much impact on the environment since it doesn't use many resources to run. The social impact of this product should be positive since the purpose of this product is just for enjoyment. Also, since this game will probably have a low popularity, so it shouldn't impact employment or create any new jobs.

Since the project was broken into multiple system which made the development of this project much easier, and after making multiple prototypes, the system work like how it should be, however, since the input lag is noticeable which is huge disadvantages compare to the other similar or existing product. If further development was to be made hazard free should be taken into consideration.

REFERENCES

Bruce, J.W. “A Digital Logic Design Project for Fall 2021 ECE 2110.” *Ship-Captain-Crew Game*, http://jwbruce.info/teaching/ece2110/fall21/project/project_f21.html.

Wikimedia Foundation. (2021, November 7). *Linear-feedback shift register*. Wikipedia. Retrieved November 23, 2021, from https://en.wikipedia.org/wiki/Linear-feedback_shift_register.