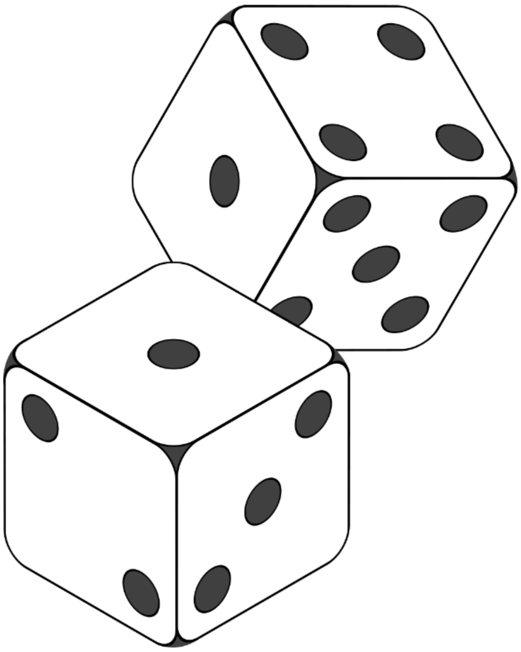
Minor Project Final Report

GROUP-3

TWO PERSON DICE GAME



Prepared by :-

B. CHANDRA MOULI (19ECE1005)

D. UDAY KIRAN (19ECE1006)

DAWSON QUDROS (19ECE1007)

G. VAMSI KRISHNA (19ECE1008)

Under the guidance of :-

Dr. Anirban Chatterjee Sir

Mr. Kalyan Sundar Kola Sir

Introduction:-

First, we took the basic idea of a dice and applied it to digital electronics, and created a two person dice game. The idea is that each person gets a chance to roll a dice where we limited the values of the dice from 0-3 just to make it simple and understandable. When each person rolls their respective rounds then the total scores would be added up and the winner would be declared based on the highest score a person gets. This is done with a comparator component such that a win or a draw or a tie is declared at the output division. This circuit is based on a luck-win situation for a player, we are trying to create a digital dice and the scenario of two persons playing with the dice. This project can used for entertainment purposes and to create digital game. The digital dice is the heart of the circuit that is the pseudo random number generator and the registers are also a crucial part of the circuit storing each person's individual scores.

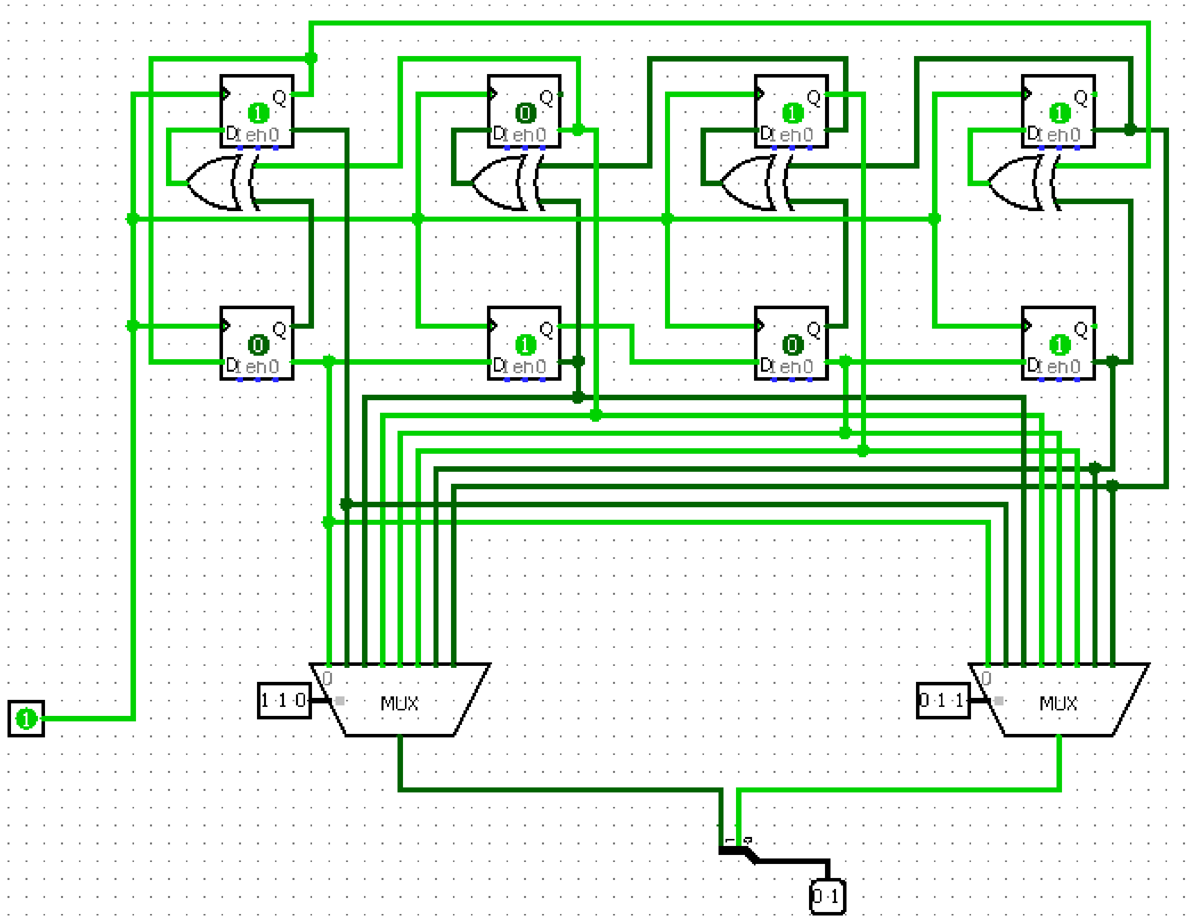
The reason we didn't take all the six values of a dice is that when we take 6 digits of the dice if the person gets the highest score, then it would take very least number of rolls i.e. 3, to complete the game and it wouldn't be any fun to play the game so we increased the number of rolls by decreasing the digits on the dice such that if the person gets highest number i.e. 3 it would take 5 rolls to fill the register's memory and to complete the game.

Background :-

As we mentioned the sole purpose of the project is to create a game with logic components entirely for entertainment purpose. The heart of the project lies in designing the game, the game should resemble a real life situation where two people play with a dice. This project started with components that were already existing, and two get a certain output from these we have modified certain basic input/output & display components such as a Seven segment display and D flip flop and a T flip flop and basic logic gates. We even built a 4 Bit BCD comparator using basic gates. We started the project with a pseudo random number generator and half adder and a two to one Multiplexer and made our own modifications to them using basic gates and wiring components such as a splitter and others. Now that we have modified the components to gain the outputs that we require we connected them in Logism because, the project is done online we used Logism software for the project.

Theory:-

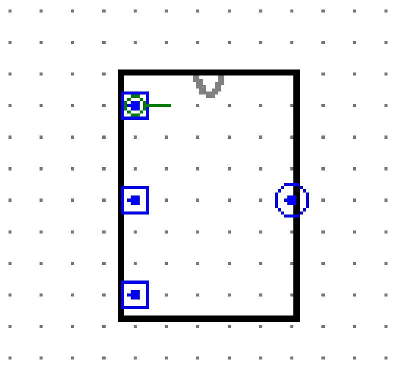
Pseudo-Random Generator:

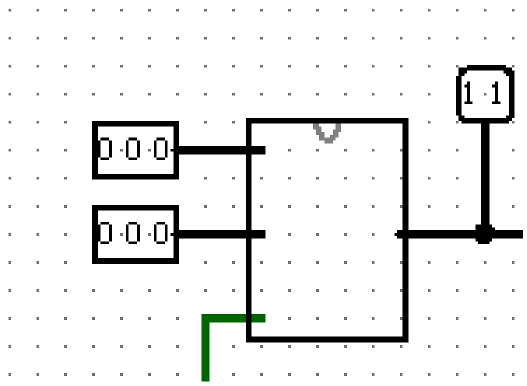


A Pseudo-Random Generator is a digital logic circuit that generates a more or less random number. In this case I made a 2 bit Pseudo-random generator.

The random generator I made consists of a clock, 2 select lines as input and produces a 2 bit output.

The IC view of the pseudo-random generator and its use in the circuit is as below.





Construction:

1) The main part of a random generator I thought was a continuous cycle of numbers so I made use of D flip flops all connected to each other with output of one D flip flop fed as input of the next flip flop. I also made use of XOR gates to increase the length of the sequence as without them it was a very short sequence.

2) I fed the clock simultaneously to all the flip flops. I realised that more the number of flip flops, longer the sequence before it starts repeating, so I settled at 8 flip flops as increasing more would increase the cost.

3) After this I fed the outputs of all flip flops to two 8:1 multiplexers as we need a 2 bit output. The outputs of these two multiplexers give the final 2 bit dice output.

4) The reason I chose to use multiplexers is so that if the players start guessing the sequence we can change the select lines and a new output from different flip flop will be selected as final output resulting in a new sequence.

Seven segment display:

A seven segment display is as the name suggests has seven LED's which is used to display data like 0,1,2,3……9 or A,B,C……. etc.

In this case I will use it to display the winner, it will display A if A is the winner, b if B is the winner and d in case of a tie/draw.

I will use the comparator outputs as the inputs to the logic behind this seven segment display as if A is the winner the A>B output will be high and if B is the winner the A<B output will be high and in case of a draw/tie the A=B Output will be high.

I will use the below diagram as example to show my notations.

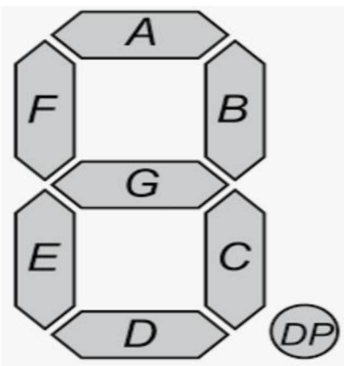
For the 7 segment display to declare the winner as

A: LED's – **A, B, C, E, F, G** should be high. And I will use the

A>B Output as input for this.

b: LED's **C, D, E, F, G** should be high. And I will use **A<B** output as input for this.

d: LED's **B, C, D, E, G** should be high, And I will use **A=B** output as input for this.



Now let us derive the logic for each LED.

- **A>B →** x
- **A=B →** y
- **A<B →** z

A: A is only used in displaying A so, A = x

B: B is used in displaying A and d so, B = x + y

C: C is used in displaying A, b and d so, C = x + y + z

D: D is used in displaying b and d so, D = y + z

E: E is used in displaying A, b and d so, E = x + y + z

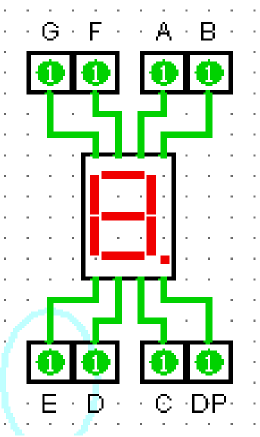
F: F is used in displaying A and b so, F = x + z

G: G is used in displaying A, b and d so, G = x + y + z

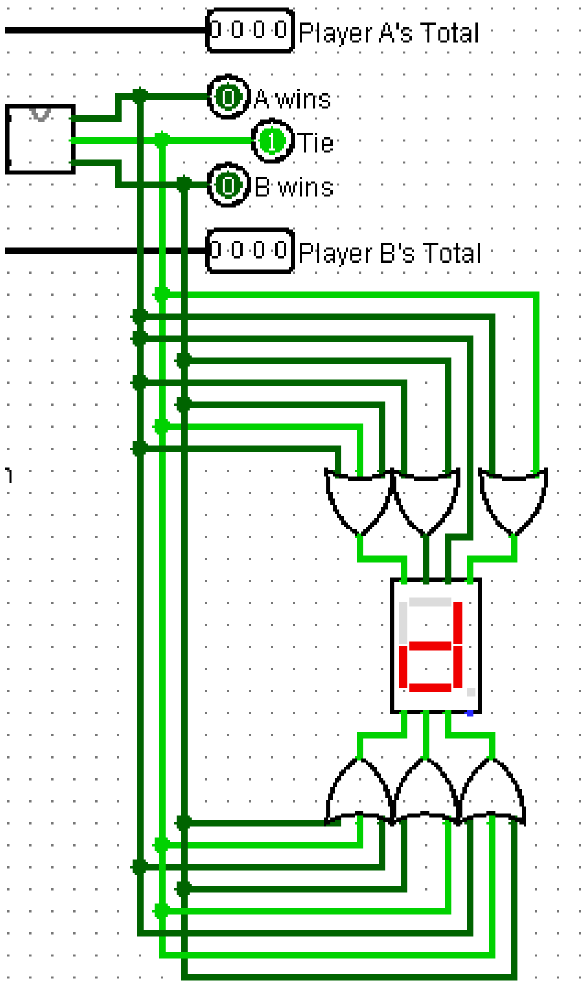
DP: Not used

All this can be implemented using only 2 and 3 input OR gates.

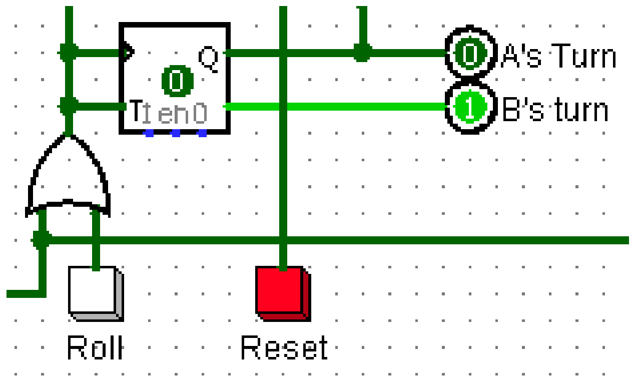
In Logism the LED's are labelled as follows in the figure, so I made the connections accordingly.



Upon final implementation the circuit looks a s follows:



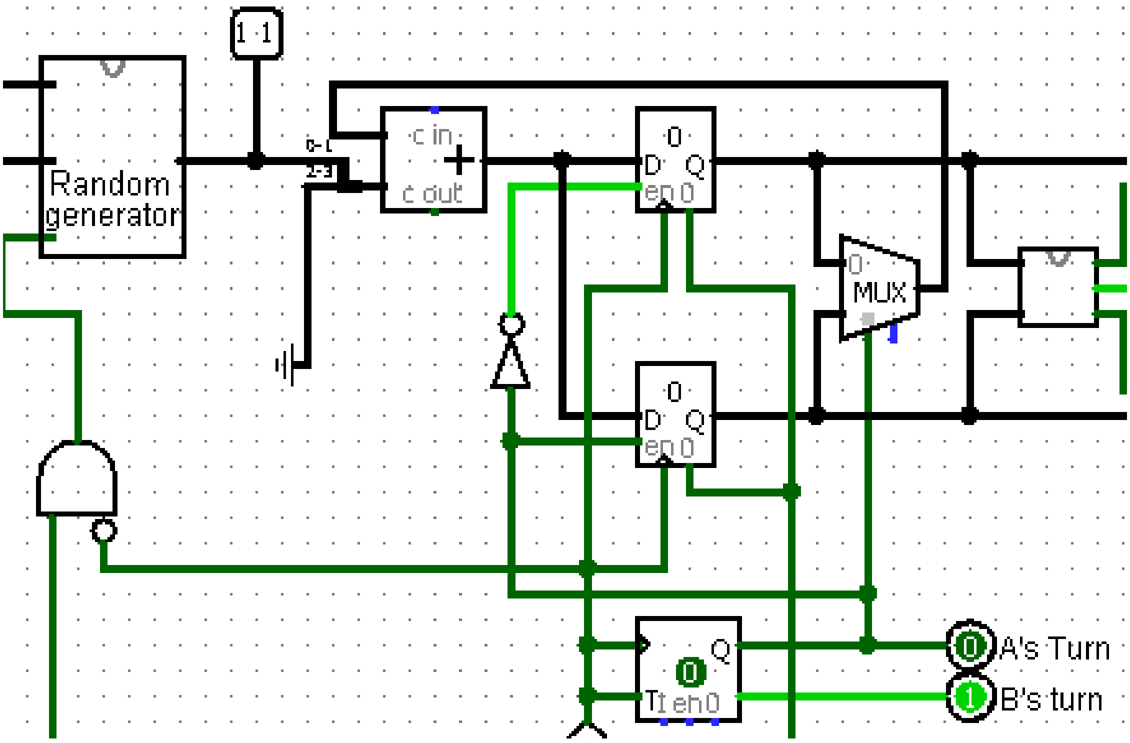
T Flip flop :



A Flip Flop is a sequential circuit used for memory, it is a logic component. A T flip flop is a Flip Flop circuit whose main job is to toggle. Toggling means it complements the previous output. In this project we use this T flip flop to toggle the turns between the 2 players. When it is player A’s turn it shows A’s turn and also enables player A’s register for the data to be stored and vice versa.

Memory Circuit :

Our memory circuit consists of 2 registers, One adder and one 2:1 Multiplexer.



Note: There is a one-step delay in the circuit so when it says it is the current turn the memory circuit is actually processing the previous roll. So if it says A’s turn the circuit is processing B’s previous turn.

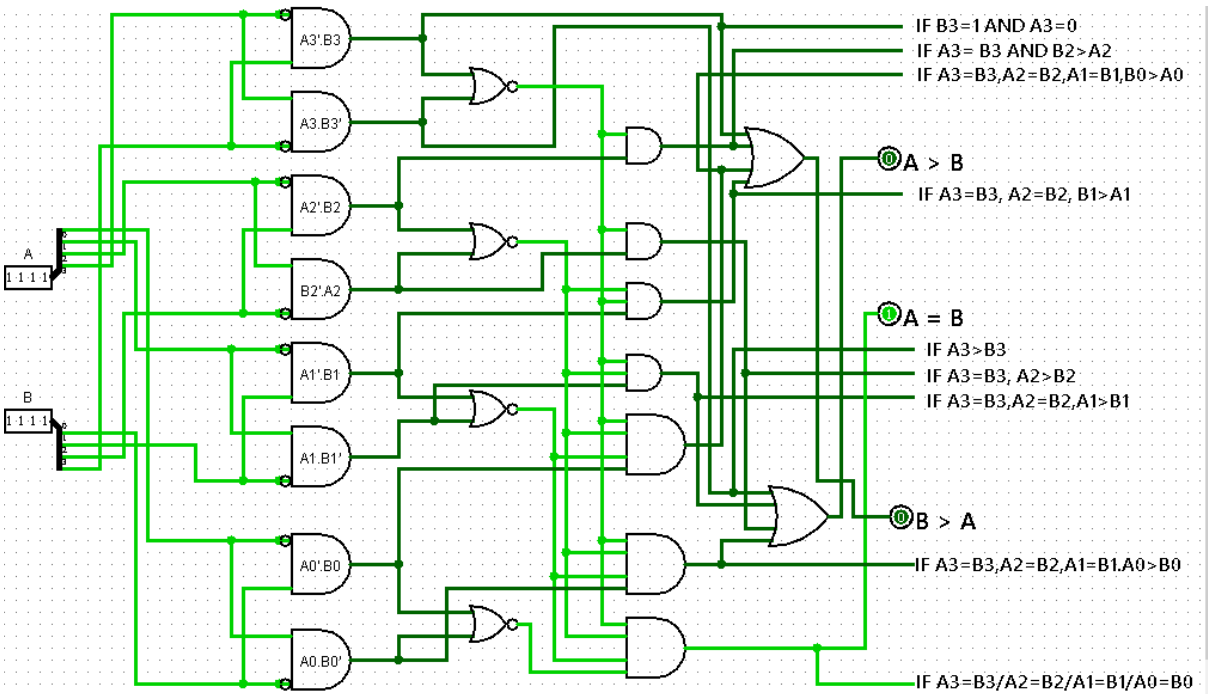
Working:

1) The T flip flop output goes as select line to the multiplexer.
2) Now this mux according to the T flip flop output selects either player A or B's registers stored value and sends it as one of the inputs to the adder.
3) This input is then added to the current rolled dice.
4) And output of this adder is then updated into the register.
5) Which register this gets added to is controlled by the enable which also takes input from the T Flip Flop output.

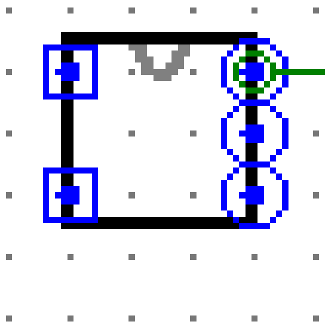
Comparator :

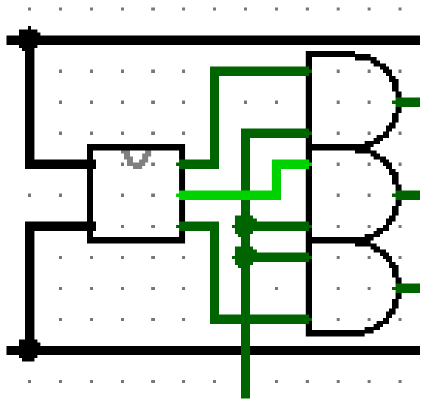
A comparator is a logic device that compares two numbers, and outputs if A>B, A=B or A<B.

For our project we are using a 4-bit comparator which compares two 4 bit numbers and gives output accordingly.



The IC view of the circuit and its use in the circuit is as follows:



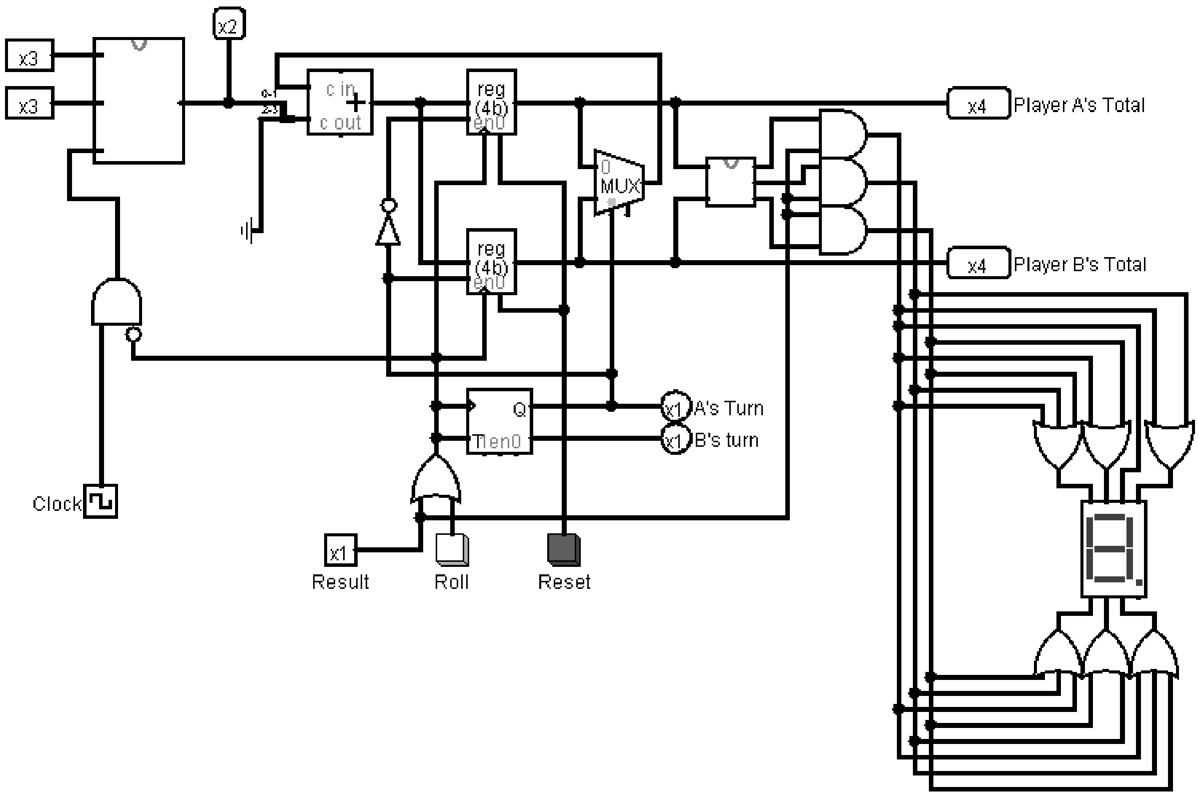


- If B3>A3, or B3=A3 and B2>A2, or B3=A3 and B2=A2 and B1>A1, or B3=A3 and B2=A2 and B1=A1 and B0>A0 then the B>A output is high.
- If A3=B3 and A2=B2 and A1=B1 and A0=B0 then the A=B output is high.
- If A3>B3, or A3=B3 and A2>B2, or A3=B3 and A2 =B2 and A1>B1, or A3=B3 and A2=B2 and A1=B1 and A0>B0 then the A>B output is high.

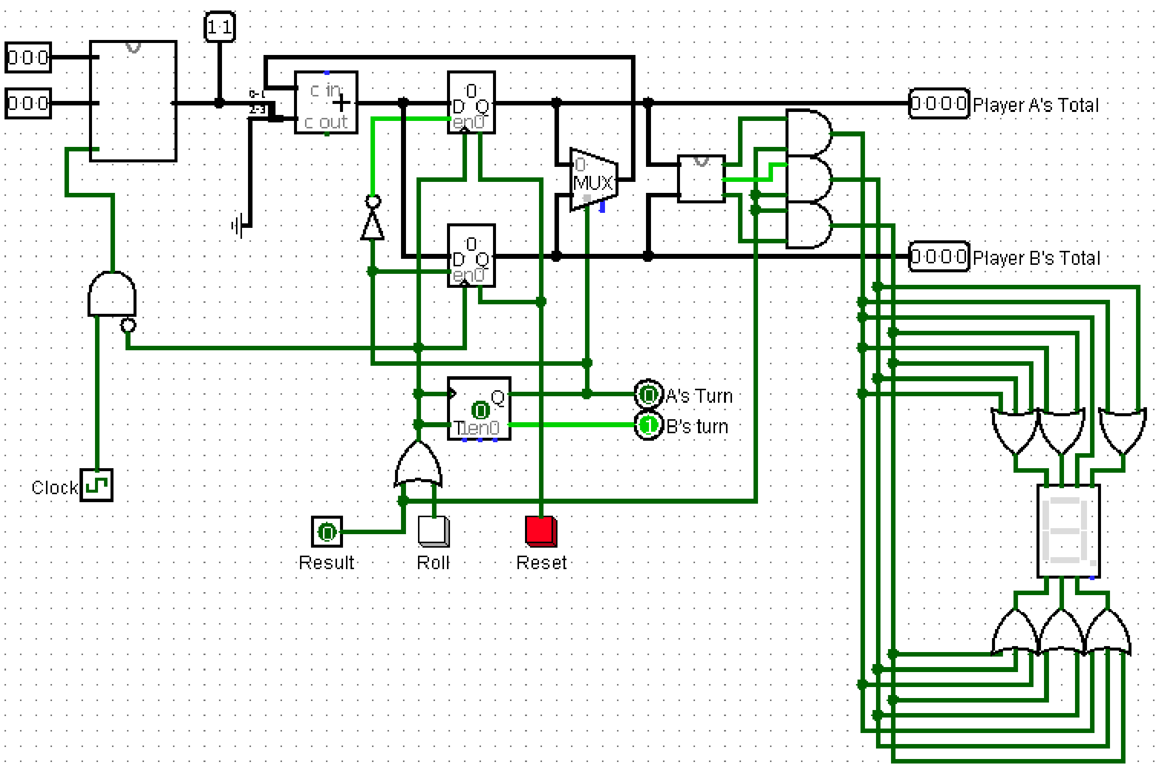
Circuit Diagram :-

The project is executed in LOGISM software.

Two person dice game



Two person dice game



Cost Estimation:-

Pseudo random number generator :

D flip flops = 8 (16/- x 8)

8:1 Mux = 2 (71/- x 2)

Comparator :

2-1 AND = 8 (15.25/- x 8)

2-1 NOR = 4 (18/- x 4)

2-1 AND = 2 (15.25/- x 2)

3-1 AND = 2 (15/- x 3)

4-1 AND = 4 (18/- x 4)

4-1 OR = 2 (23/- x 2)

NOT = 8 (19/- x 8)

Circuit :

T flip flop = 1 (using jk flip flop 48/-)

2: 1 Mux = 1 (20/-)

2-1 AND = 4 (15.25/- x 4)

2-1 OR = 4 (19/- x 4)

3-1 OR = 3 (20/- x 3) & NOT = 2 (19/- x 2)

7 segment display = 1 (12/-)

Half adder =1(13.38/- x 4 NAND gates)

The project cost is Approximately **1178.02/- INR**, it would be varying from 1200-1500 INR.

Observations/Drawbacks:

1) The values of the select lines on both multiplexers should be different because otherwise it produces either 00 or 11 as the output as both multiplexers produce same output.
2) My testing suggested that more or less every time the generator produces a 24 number length sequence before it starts repeating.
3) I tried putting XOR gates at the bottom part of the circuit too but I observed that after a few numbers it was repeating the same number infinitely so I didn't.
4) As I tested the circuit I understood that the pseudo random number generator is generating random numbers and most of the time the outcomes of the games are different.
5) There is a very less probability that the game may end in a draw because its very unlikely to get same totals when the dice is random.
6) It is evident that no logical circuit is random because the circuits are built from logical expressions and all logical expressions have certain outputs for certain inputs yet that is the reason we call this dice as a Pseudo random number generator.

Result:-

1. We created a game through digital electronics components.
2. We learnt and modified the behavior of certain fundamental and combinational and sequential circuits.

References:-

1.Wakerly J F, “Digital Design: Principles and Practices”, Prentice-Hall, 2nd Ed., 2002

2. Mano M. M., “Digital Logic Design”, Prentice Hall 1993.

Image sources :

www.ClipartMax.com

www.PNGitem.com