Computer Organization & Design Lab Project

Information & Communication Technology Department

School of Technology, PDEU

Name Of Group Members	Roll. No
Rishil Trivedi(Group Leader)	**20BIT103**
Vishwas Darji	**20BIT039**
Ravi Prajapati	**20BIT021**
Jay Suthar	**20BIT077**
Amit Sagathiya	**20BIT139**

Table Of Content

- **Aim:-** Design a Fibonacci Generator which will generate Fibonacci numbers automatically.

Working Principle:-

- The Fibonacci Generator follows the same procedure as we do in any Programming Language:

```
def fibogen(n)
        a=0;
        b=1;
        c=1;
for I in range (0,n):
        print(a);
        c=a+b;
        b=c;
        a=b;
```

- We have implemented this logic in the Logisim. First we have used 3 Registers of 32-bits as variables. Registers stores the data and update the data when it's clock pulse goes low to high. We use pull up registers to assigning 0(low) for Register clock inputs because without low signal clock can't go high.
- We gave the registers predefined values like A=0,B=1,C=1. Then we have used loop for updating the Registers. 1st print A which is 0, then update C to be A+B then done shifting of B=C, A=B.
- We used control buffers for assigning B to 1(We didn't directly connect constant 1 to B because it can create conflict). Control buffers decide whether to drive the signal on the output, when the bottom line also called control line goes low to high means 0 to1 it allows the signal to go through output line.

- When we are setting the circuit we control one buffer to assign 1 to B, but when we are using the loop we drive value of C to B.
- We used SR-Latch to control buffers. For assigning 1 to B we connected SR latch's inverse/complement output to control buffer(when the Register B's clock pulse will change it will assign B=1). The second output is connected to other control which is going to update B with the value of C when Loop starts. Also we connected output of B to input of Register A so the shifting of B=A can happen during Loop.
- We have used multiple buttons connected to the different input of Registers and SR Latch. By manually press button Reset which connected at clear line of all the components to clear data. Then press B button to load the B with 1 then press Enable button for enabling SR Latch. After that click button C to load data in register C. Press button it will update the content of A with C. By pressing button B you update content of B with C. Do this in order C, A and B. we can see output in probe in decimal format.

- **Making the Circuit Automatic:-**

- We want to control circuit automatically by enabling the clock and clicking one button.. For that we used demux control circuit. Demux circuit can be implemented by using a Demux with 2 selecting bits. At select line we used a counter(counter basically counts how many time signal goes low(0) to high(1)).
- We have used an inverter at the input of counter the reason we want the counter to update to next line of Demux when the clock output goes low if we don't connect inverter it will send a signal to and count at the same time which will not give correct output we want.

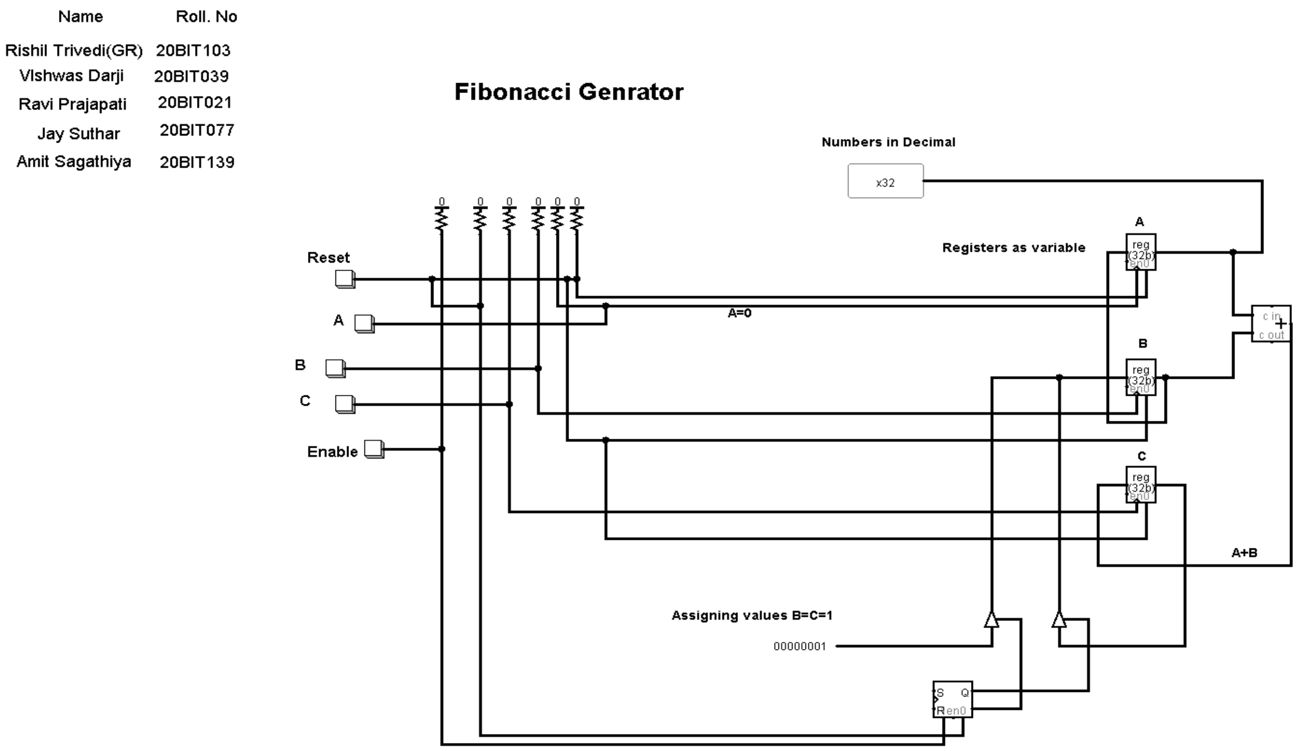
- We have used a AND gate and SR latch at the input side where AND gate's one input is connected to a clock(Frequency:2Hz) and second input is connected to the output of SR Latch. When the SR Latch is enabled the signal will go through AND gate and gives the signal to Demux.
- The control buffers been used at output of Demux. 1st line of output is connected to both Register A and SR Latch's clear line so both are zero at starting. 3rd buffer is connected to Register B's clock input so it will update the Register B. 4th Buffer is connected to preset line of SR Latch to enable the SR Latch.
- We have to disable the SR Latch in the Demux control circuit so we can enter our main loop. For disabling it we used a falling edge detector (when signal goes high to low it detect it and send a brief signal).
- We made the falling edge detector using one AND gate and two NOT gates(When input is low(0) 1st input will be high(1) and 2nd will be low(0) but signals don't propagate instantly there will be bit Lag at second NOT gate so for that one moment the edge detector will go high).
- We connected the Falling edge detector to the third cycle so when the Demux's third cycle goes low the edge detector disables the SR Latch.

- <u>For Main Loop:-</u>

- Now for the main loop 1st we need to pulse Register C's clock pulse so we put 1st control buffer to update Register C, two more buffers for shifting C into B and B into A.
- We have used another Demux control circuit for controlling the main loop. In this control circuit Mux will have 2 outputs because we want to shift C into B and B into A at the same time otherwise we can use more output lines.

- The second Demux control circuit's SR Latch will be enabled when the First Demux control circuit's SR Latch goes low or we can say first loop is over.
- We implemented a button to reset the whole Circuit when you will press it all the registers get cleared and the whole circuit will start from the beginning.
- We have used a probe at the output of Register A as in Programming language to print value of A. The probe will give the output of A in decimal format which is easy to understand instead of Hexadecimal or Binary.
- The whole circuit gets reset when you press reset button. You can also make it stop by disabling the clock. For getting the numbers at faster rate we can change the frequency of clock from the Simulate option in tool bar.

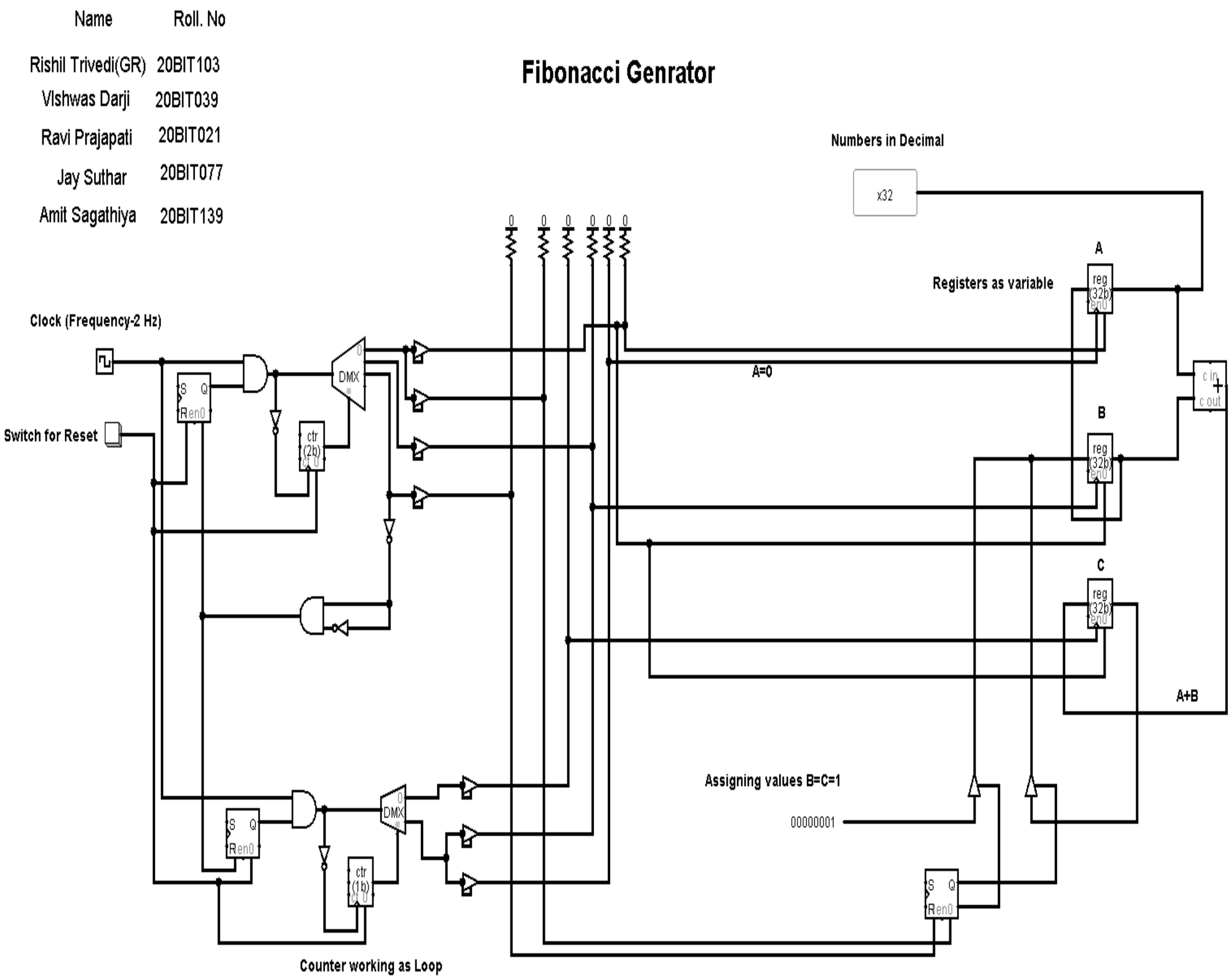
❖ Circuit:-



Name	Roll. No
Rishil Trivedi(GR)	20BIT103
VIshwas Darji	20BIT039
Ravi Prajapati	20BIT021
Jay Suthar	20BIT077
Amit Sagathiya	20BIT139

Fibonacci Genrator

Fibonacci Generator which will generate numbers mannually



Name	Roll. No
Rishil Trivedi(GR)	20BIT103
VIshwas Darji	20BIT039
Ravi Prajapati	20BIT021
Jay Suthar	20BIT077
Amit Sagathiya	20BIT139

Fibonacci Generator That will Generate Fibonacci Numbers Automatically

- ❖ **Result:-** The Fibonacci generator which will generate Fibonacci number automatically has been implemented successfully.

❖ Skills Developed (each member):-

- Rishil Trivedi (20BIT103):-

  As a Group Leader I learned

  - ☞ How to distribute task among the team members.
  - ☞ How to take out best from the members and to assign appropriate work in which they are expert in.
  - ☞ I worked upon the logic and provided a prototype code , So that it becomes easy to apply in circuit and gets easy to make connections.

- Amit Sagathiya (20BIT139):-

  - ☞ Understood the overall plan and got to know about my work.
  - ☞ Realised the importance of register in overall project . Until now I thought its just a thing that stores bits but while make Fibonacci Series I understood how crucial the use of memory is for the circuit.

- Jay Suthar (20BIT077):-

  - ☞ Able to Understand and explain all the components to group members. Improved verbal Communication.
  - ☞ Got the idea how a simple code from the Programming language can be implemented as a circuit.
  - ☞ Worked on the main connections of the circuit and made complete input part with clock through which I realised the importance of clock signal

- Vishwas Darji (20BIT039):-

    - ☞ I worked upon debugging of the circuit. Many times desired output does not came.
    - ☞ From changing the functions of Registers and Demultiplexer and fixing the mistakes in the circiut which I faced once in the Lab Sessions.
    - ☞ Overall i t was good experience in Working as a team.

- ➢ Ravi Prajapati (20BIT021):-
    - ☞ Realized how a manual circuit can be made into automatic circuit by using clock signals and loops.
    - ☞ Understood and able to implement loop in the circuit input.
    - ☞ Learnt about counters and Edge frequency detector . It was great experience to work as a group.