

# Random Number Generator using Shift Registers and XOR Gates

Anirudh Saikrishnan  
CS22BTECH11004

May 18, 2023

## 1 Introduction

Random number generators (RNGs) are essential in various computational applications that require randomness, such as simulations, cryptography, and gaming. In this report, we will discuss the concept of creating a random number generator using shift registers and XOR gates.

## 2 Components

1. Breadboard
2. Seven Segment Display : Common Anode
3. Seven Segment Display Decoder [7447]
4. FlipFlop [7474] x2
5. XOR gate [7486]
6. 555 IC
7. Resistors [10M $\Omega$ , 1K $\omega$  x2]
8. Capacitors [47nF, 470nF]
9. USB micro B breakout board
10. Jumper wires

## 3 Shift Registers

Shift registers are sequential logic circuits composed of flip-flops connected in a chain. They can store and shift binary data based on a clock signal. The data stored in the registers is shifted from one flip-flop to another, allowing for sequential processing.

## 4 XOR Gates

XOR gates are digital logic gates that produce an output of "1" if the number of "1" inputs is odd, and "0" otherwise. XOR gates are commonly used in various applications, including binary addition, error detection, and randomness generation.

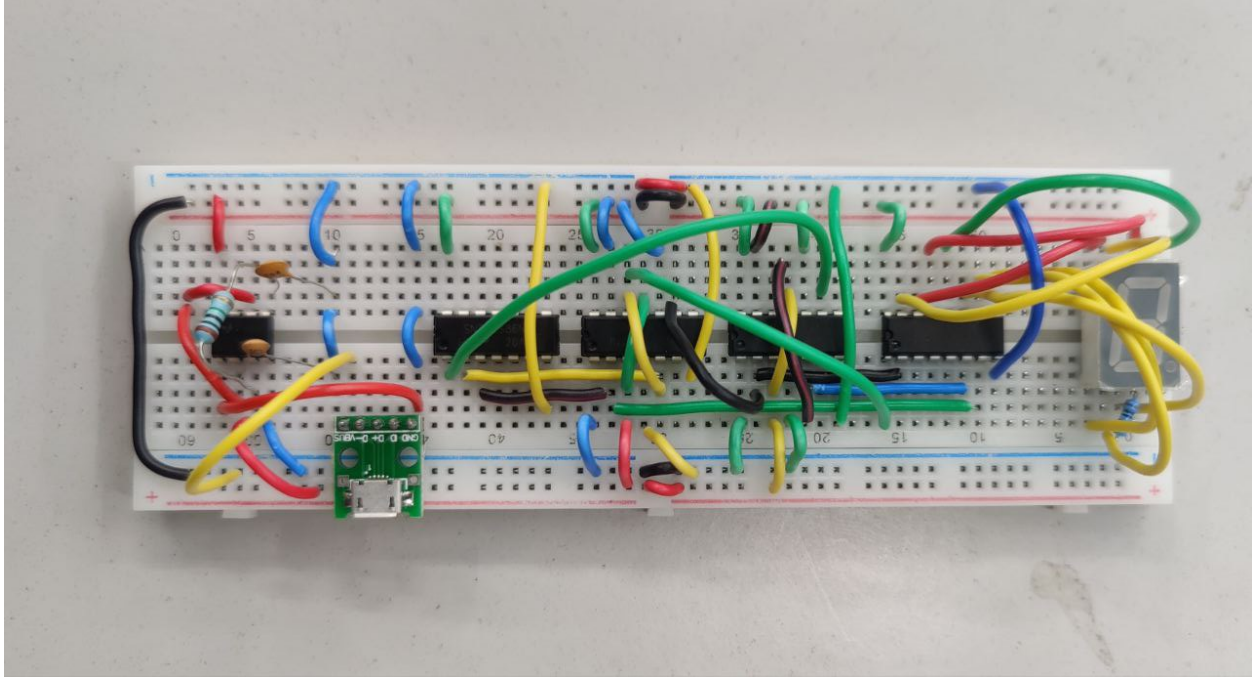


Figure 1: Circuit Board

## 5 Creating a Random Number Generator

To create a random number generator using shift registers and XOR gates, we can utilize the concept of feedback shift registers (FSRs) with XOR operations. FSRs generate a sequence of pseudo-random numbers based on the feedback of selected flip-flops through XOR gates.

### 5.1 Algorithm

The following algorithm describes the steps to create a random number generator using shift registers and XOR gates:

1. Initialize the shift register with an initial seed value.
2. Generate a clock signal.
3. Shift the register contents one bit to the right.
4. Compute the next bit based on the XOR operation of selected flip-flop outputs.
5. Append the computed bit to the leftmost side of the register.
6. Repeat steps 3-5 to generate the next random number.

## 6 Conclusion

In this report, we discussed the concept of creating a random number generator using shift registers and XOR gates. We explained the working principle of shift registers and XOR gates and described the algorithm to generate random numbers. We also provided a Python implementation as an example. By utilizing feedback shift registers with XOR operations, we can generate pseudo-random numbers efficiently.

Random number generators are widely used in various applications, and understanding the underlying mechanisms can help in designing and implementing robust randomness generation techniques.

## 7 Block Diagram

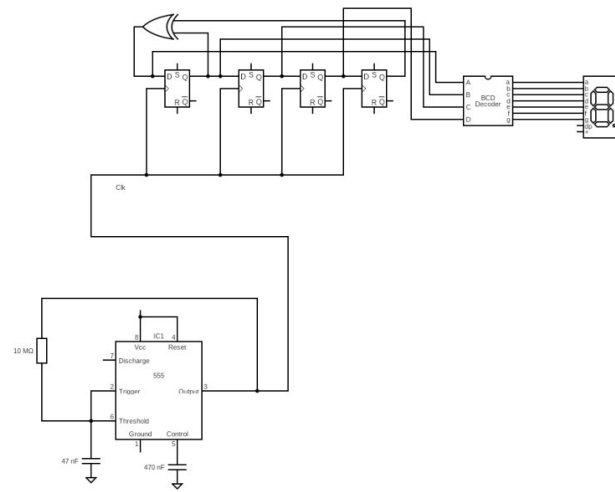


Figure 2: Block Diagram