Design/Practical Experience [EEN1010]
Department of Electrical Engineering
(Final Report)

Academic Year: 2021-22
Semester: 2

Date of Submission of Report: 01/05/2022

1. **Name of the Student:** Shashank Kumar
2. **Roll Number:** B20EE063
3. **Title of the Project:** Design and implementation of scrambling and descrambling circuits for communication.
4. **Project Category:** 3
5. **Targeted Deliverables:**
   >Implementation of scrambler, descrambler on text data using python/c/MATLAB.
   >Implementation of scrambler, descrambler on information signal.
   >We will try to implement a real time application.

6. **Work Done:**
   1} Designed a basic Scrambler and Descrambler circuit based on 4-bit LFSR on breadboards using IC7474.
   2} Implemented and analyzed different LFSRs and NLFSRs on Matlab Simulink to get the optimal pseudo random sequence.
   3} Implemented a Scrambler and Descrambler for text based data on Python.
   Followings are the kind of sequences we implemented:
   - m-sequence.
   - Gold sequence.
   - JPL sequence

   4} Implemented on a FPGA board to get pn-sequence

7. **Concluding Remarks:**
   The function of Scrambler is:
   - To remove long strings of 1s and 0s from digital binary data.
   - To maintain secrecy of data.

The LFSR m-sequence serves us better in removing long sequences of 1s and 0s but does not provide better secrecy and is only good for single user communication.

So we then implemented a Gold sequence which serves us better in both, the communication issue and secrecy issue and better for multiple user communication.

8. **References:** https://people.ece.ubc.ca/edc/4340.fall2014/lectures/lec12.pdf

   https://en.wikipedia.org/wiki/Gold_code#:~:text=A%20Gold%20code%2C%20also%20known,are%20named%20after%20Robert%20Gold.
   https://www.gps.gov/technical/icwg/IS-GPS-200J.pdf
   http://surl.li/bwuxd
   Source Book: Sequences and their applications.

9. **Declaration:** I declare that no part of this report is copied from other sources. All the references are properly cited in this report.

Shashank kumar.

Signature of the Student **Signature of the Supervisor**

____

<u>Supervisor's Recommendation for the Evaluation</u>

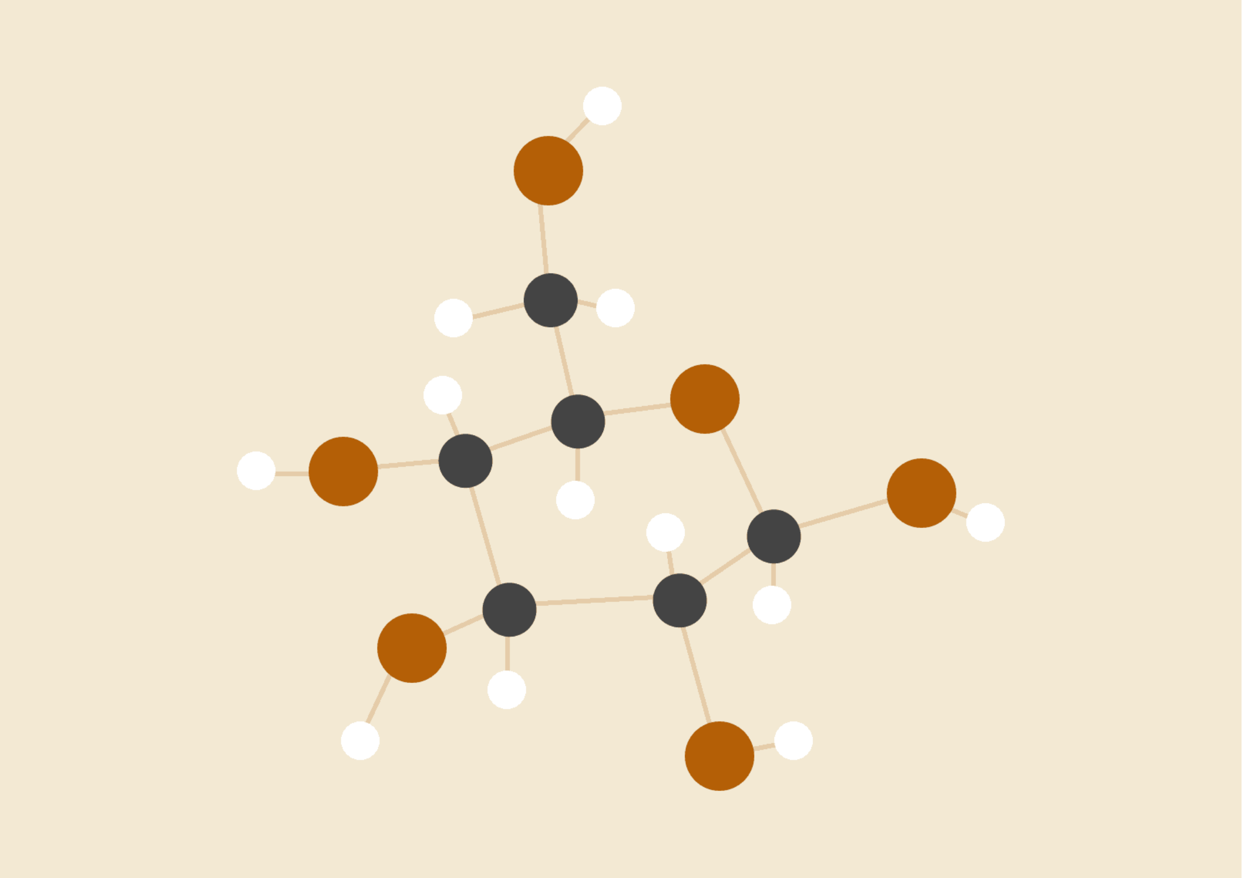
Please tick any one of the following

1. The work done is satisfactory, and sufficient time has been spent by the student. The submission by the student should be evaluated in this term.
2. The work is not complete. Continuity Grade should be given to the student. The student would need to be evaluated in the next semester for the same Design Project with me.
3. The work is not satisfactory. There is no need for evaluation. The students should look for another Design Credit Project for the next semester.
4. [Other Comment, if 1-3 are not valid] ______________________________

Signature of the Supervisor

DESIGN CREDIT

Scrambler and Descrambler



Shashank Kumar (B20EE063)

Pratham Chaurasia (B20EE041)

AIM:

- Implementation of scrambler and descrambler on text data using Python /C /MATLAB.
- Implementation of scrambler, descrambler on information signal.

WORK DONE:

- Designed a basic scrambler and Descrambler pseudo random number generator on a breadboard essential for smooth traversing of long sequences of 1s and 0s, the scrambler sequences.
- Implemented and analyzed different LFSRs and NLFSRs on Matlab Simulink to get the optimal pseudo random sequence.
- Written Python code for Scrambling and Descrambling on text data.
- Implemented on an FPGA board to get pn-sequence.

Scrambling:

Data scrambling is the process to obfuscate or remove sensitive data. This process is irreversible so that the original data cannot be derived from the scrambled data.

Real world data contains repetitive components. Examples include sequences of constant values such as a document with sections that are all one level, or repetitive data values in a file or database.

The two main possible problems introduced by this non-random data:

1. Long sequences of certain values will result in a signal that may not have enough transitions to allow for clock recovery.
2. Secrecy of data.

To solve these problems most communication systems use **scramblers** to remove undesirable patterns in data.

For scrambling of data we need to first generate waveforms whose values are predictable, these types of signals are called pseudo-random signals. Because of their noise-like characteristics they are also called pseudo-noise (PN) sequences.

Properties of PN sequences:

PN sequences have many important applications in communication systems. Here, we will discuss a particular Maximal-Length (ML) sequence, how they are generated and their application in scrambling of data.

- A PN sequence is called maximal length sequence if the sequence has a period of 2^k-1, where k is the number of bits in Shift Register.
- There are exactly 2^{k-1} ones and 2^{k-1} zeros.
- The sequence is approximately orthogonal to any shift of itself.

Generating of Maximal Length PN sequence:

We generate ML PN sequences using maximal length feedback polynomial called Characteristic polynomial.

History:

During World War II, many scrambler signals were installed to prevent accidental or intentional eavesdropping. The Frequency Changer, by Winston Churchill referred to as the Scrambler Phone, developed in 1939 by the British General Post Office. It was based on inversion of the voice spectrum.

Theory:

1. **LFSR** - Linear Feedback Shift Register, where input bit is a linear function of its previous state. The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value.

2. **NLFSR** - A nonlinear-feedback shift register (NLFSR) is a shift register whose input bit is a nonlinear function of its previous state.

3. **Gold Sequences** - A Gold sequence is a type of binary sequence, used in telecommunication and satellite navigation. A set of gold code sequences consists of 2^n+1 sequences each one with a period of 2^n-1. A set of Gold codes can be generated by taking two maximal length sequences of the same length 2^n-1 such that their absolute cross correlation is less than or equal to $2^{(n+2)/2}$ where n is the size of linear feedback shift register used to generate maximal length sequence.

   The highest absolute cross-correlation in this set of codes is $2^{(n+2)/2} + 1$ for even n and $2^{(n+1)/2} + 1$ for odd n.

4. **JPL Code** - JPL sequences or JPL codes consist of two linear feedback shift registers (LFSRs) whose code sequence lengths L_a and L_b must be prime (relatively prime).

   Overall Length of sequence $L_c = L_a * L_b = (2^m-1)*(2^n-1)$.

   Areas of application include distance measurements utilizing spread spectrum signals for satellites and in space technology. They are also utilized in the more precise military P/Y code used in the Global Positioning System (GPS).

PROCEDURE

1. **To design the basic scrambler and descrambler circuit on breadboard:**
   - First we designed a pseudo random number generator on a breadboard essential for smooth traversing of long sequences of 1s and 0s, the scrambler sequence, using D-flip flops.

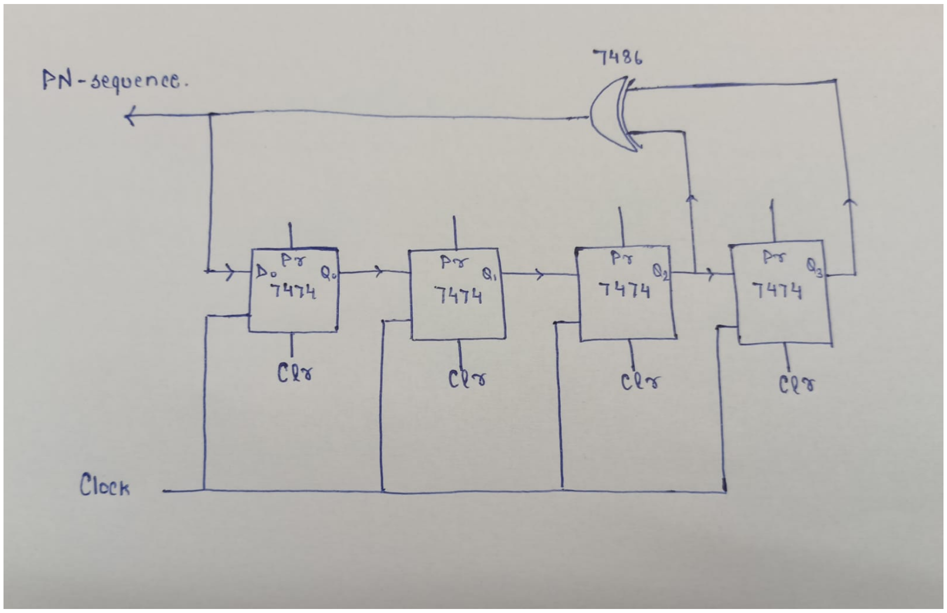
     Feedback polynomial taken for that was:

     $$\mathbf{X^4 + X^3 + 1}$$

     Which is a maximal length characteristic polynomial for 4-bit LFSR.

     And the D-flip flop that we used was **IC-7474.**

The circuit diagram for the PN sequence generator is shown:

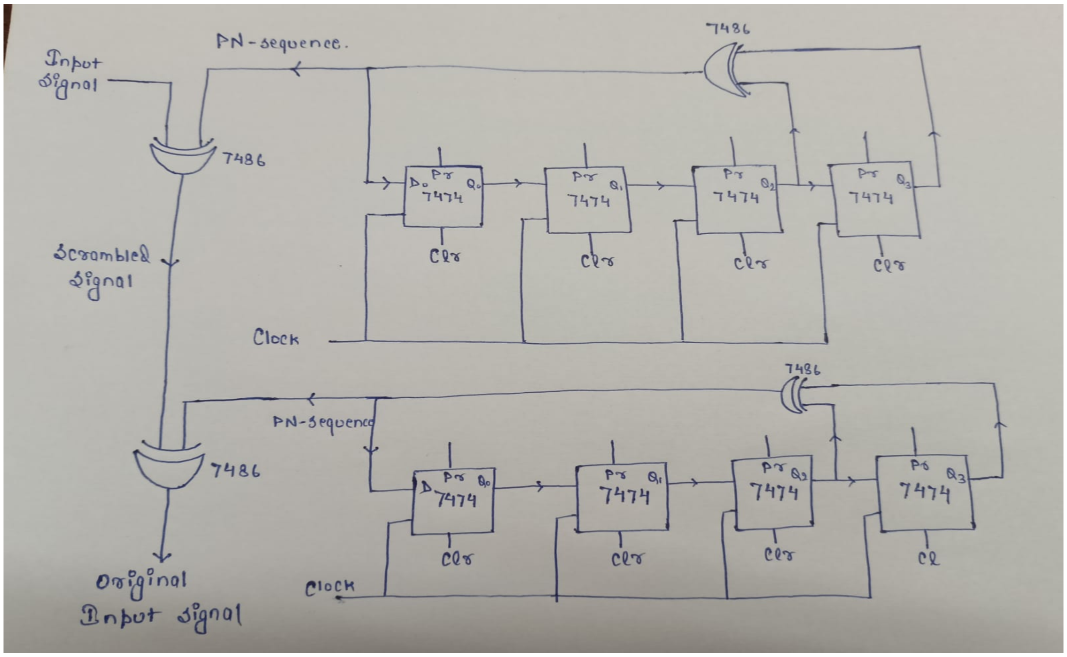


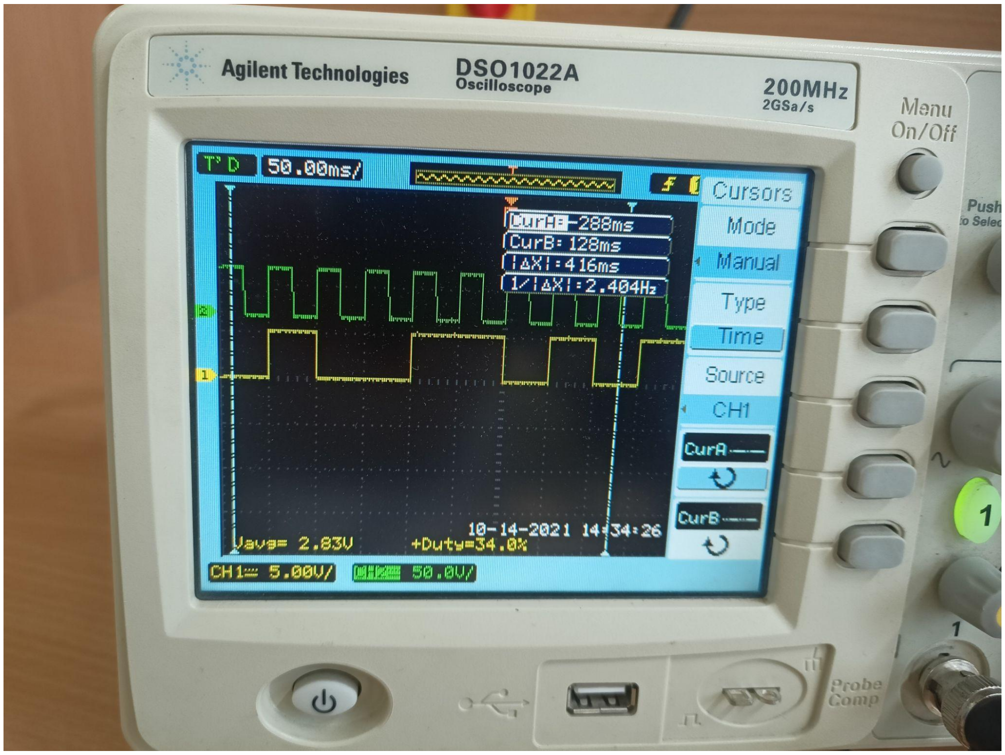
PN sequence generated here - “1 1 0 0 0 1 0 0 1 1 0 1 0 1 1”

- Then, we took the XOR of this scrambled sequence to produce a scrambled signal, which as result removes the continuous sequence of 1s and 0s from the input signal on one end.
- After that, we took XOR of this scrambled signal again with the same scrambled sequence to get back the original signal.

This was the Scrambling and Descrambling part of the information signal. On matching the final output signal and the input signal, they were both same and hence we get back our same signal which we have scrambled at the sender’s end.

Complete Scrambler and Descrambler Circuit:





T'D 50.00ms/

CurA: -288ms

CurB: 128ms

|ΔX|: 416ms

1/|ΔX|: 2.404Hz

Mode: Manual

Type: Time

Source: CH1

Vavg= 2.83V

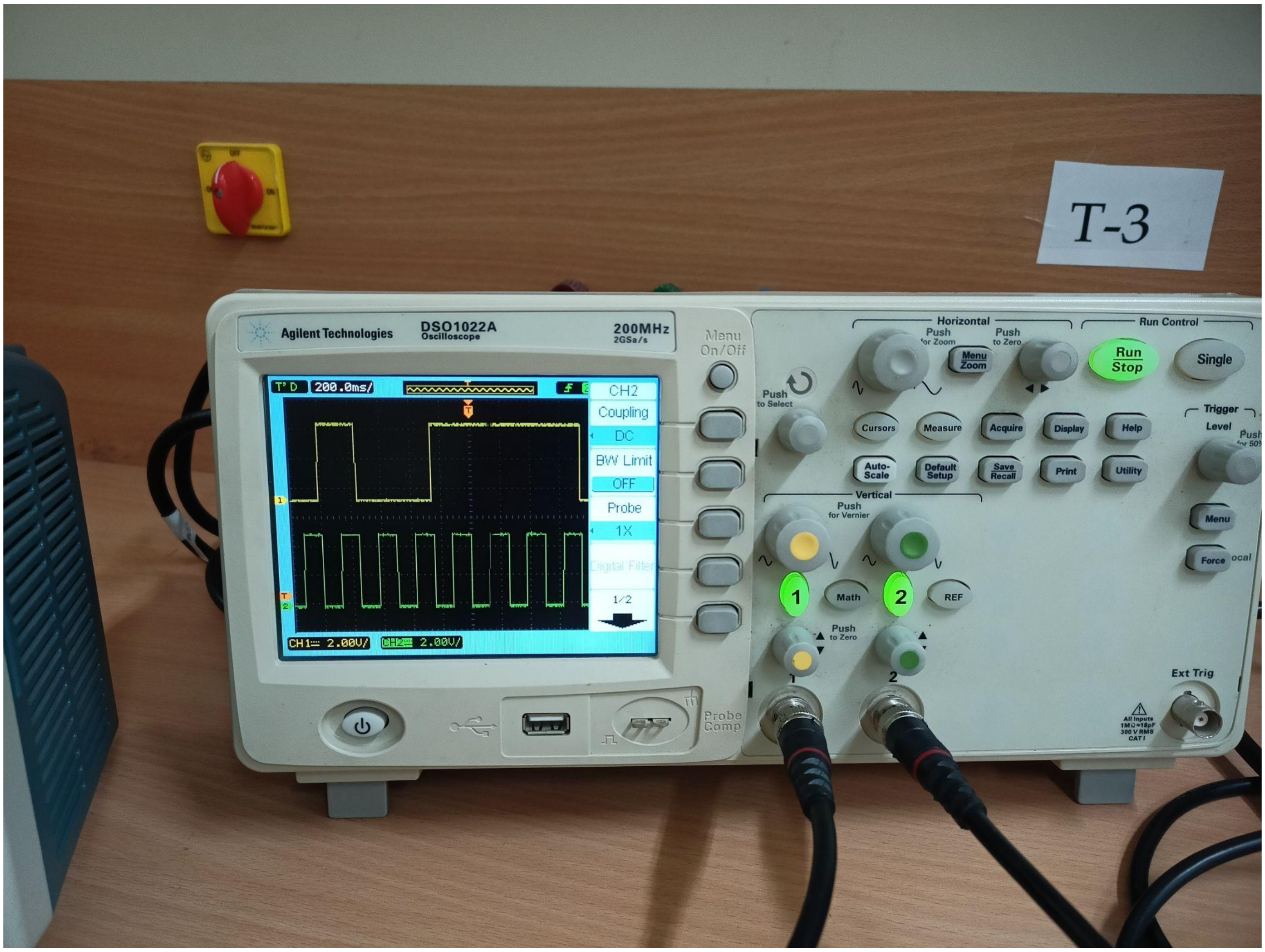
+Duty=34.0%

10-14-2021 14:34:26

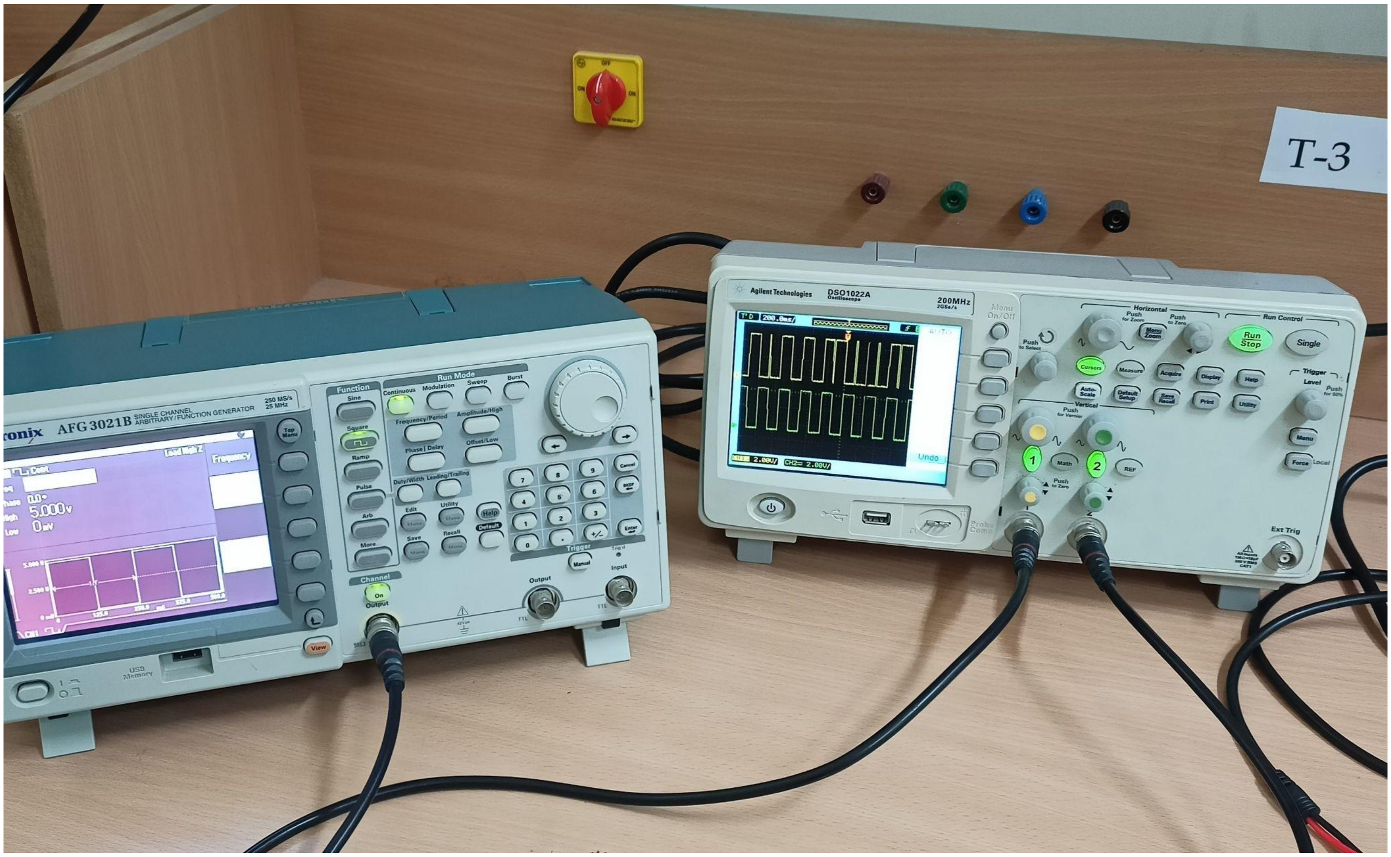
CH1 5.00V/

CH2 50.0V/

PN sequence on oscilloscope (Clock in Green and PN sequence in Yellow)

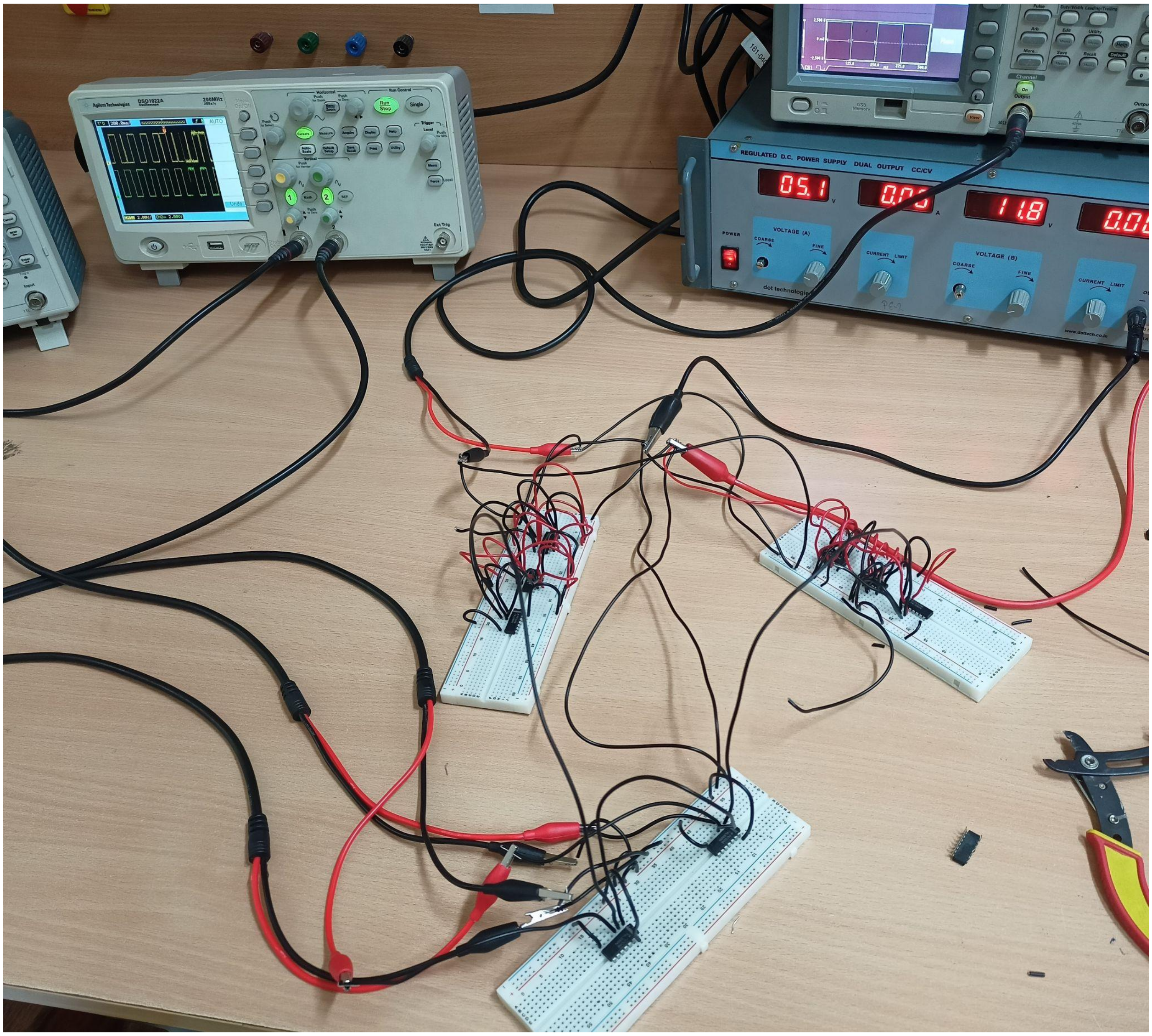


Input Signal Vs Scrambled Signal (below in green is input signal and above in yellow is scrambled signal)



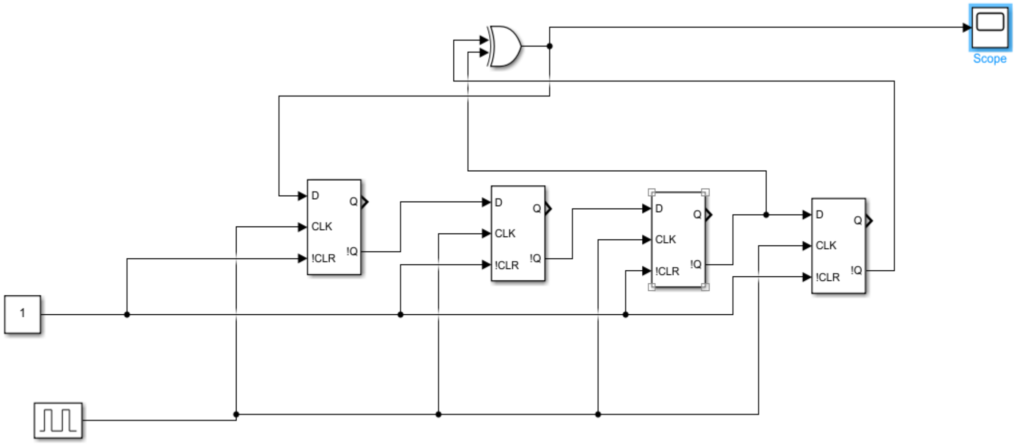
Input signal vs Final Output signal after scrambling and descrambling (below is input signal and above is output signal)

Here you can observe that the output signal is not exactly the same as the input signal as there are some clock synchronization issues. (due to time constraints in physical mode we were not able to properly solve that issue).

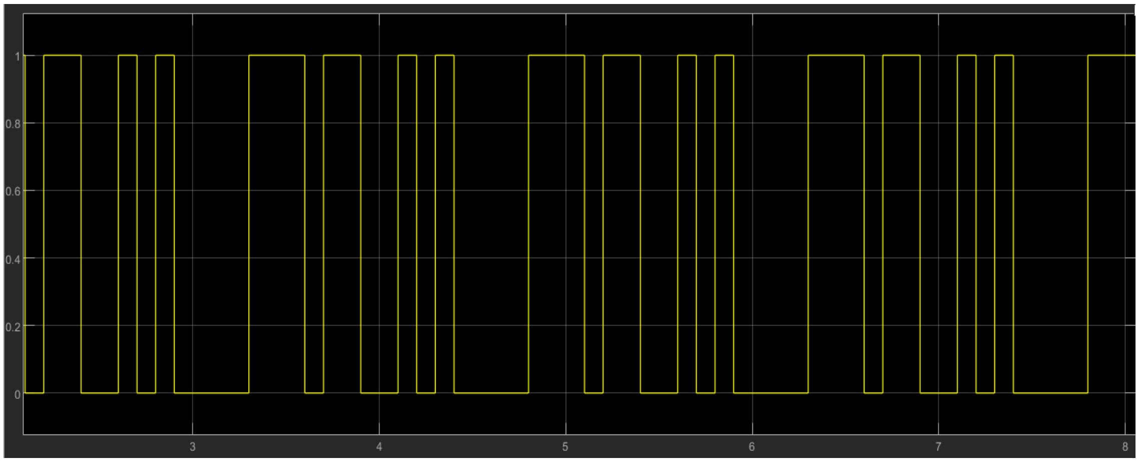

Agilent Technologies DSO1022A
REGULATED D.C. POWER SUPPLY DUAL OUTPUT CC/CV
05.1
11.8
VOLTAGE (A)
VOLTAGE (B)
CURRENT LIMIT
COARSE
FINE
POWER

Breadboard circuits for Scrambler and Descrambler.

2. **Implemented and analyzed different LFSRs and NLFSRs on Matlab Simulink to get the optimal pseudo random sequence.**
    - First we implemented the same circuit we made in physical mode on breadboard on simulink this time.

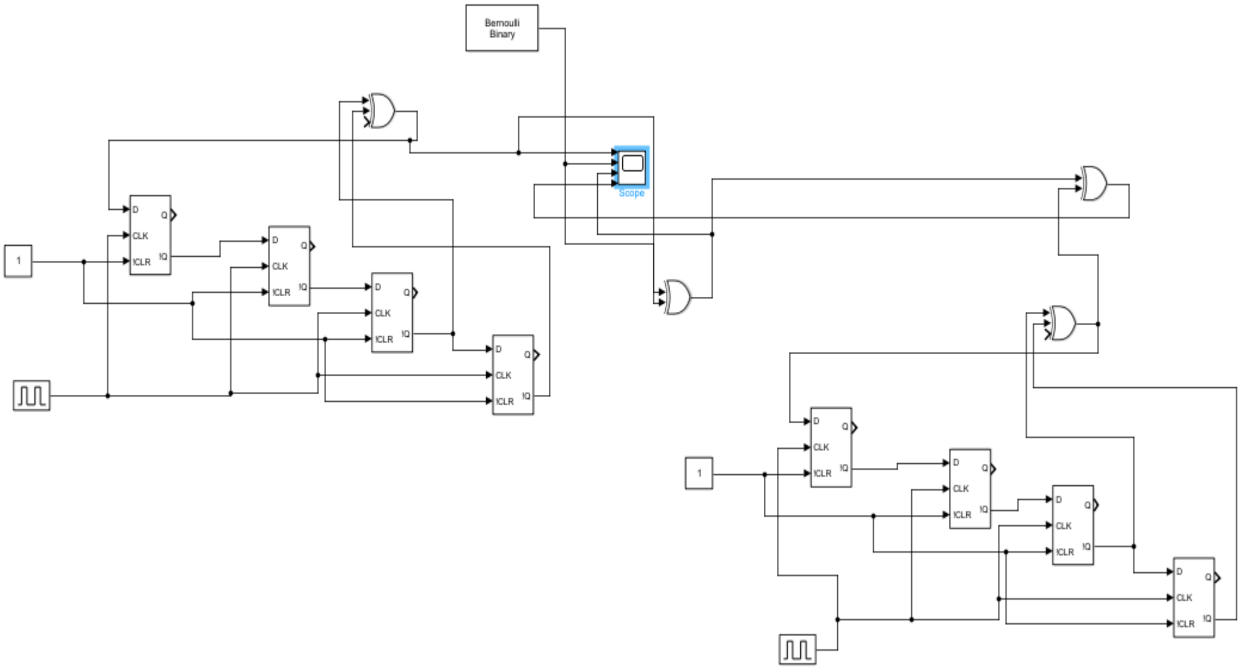


PN sequence generator

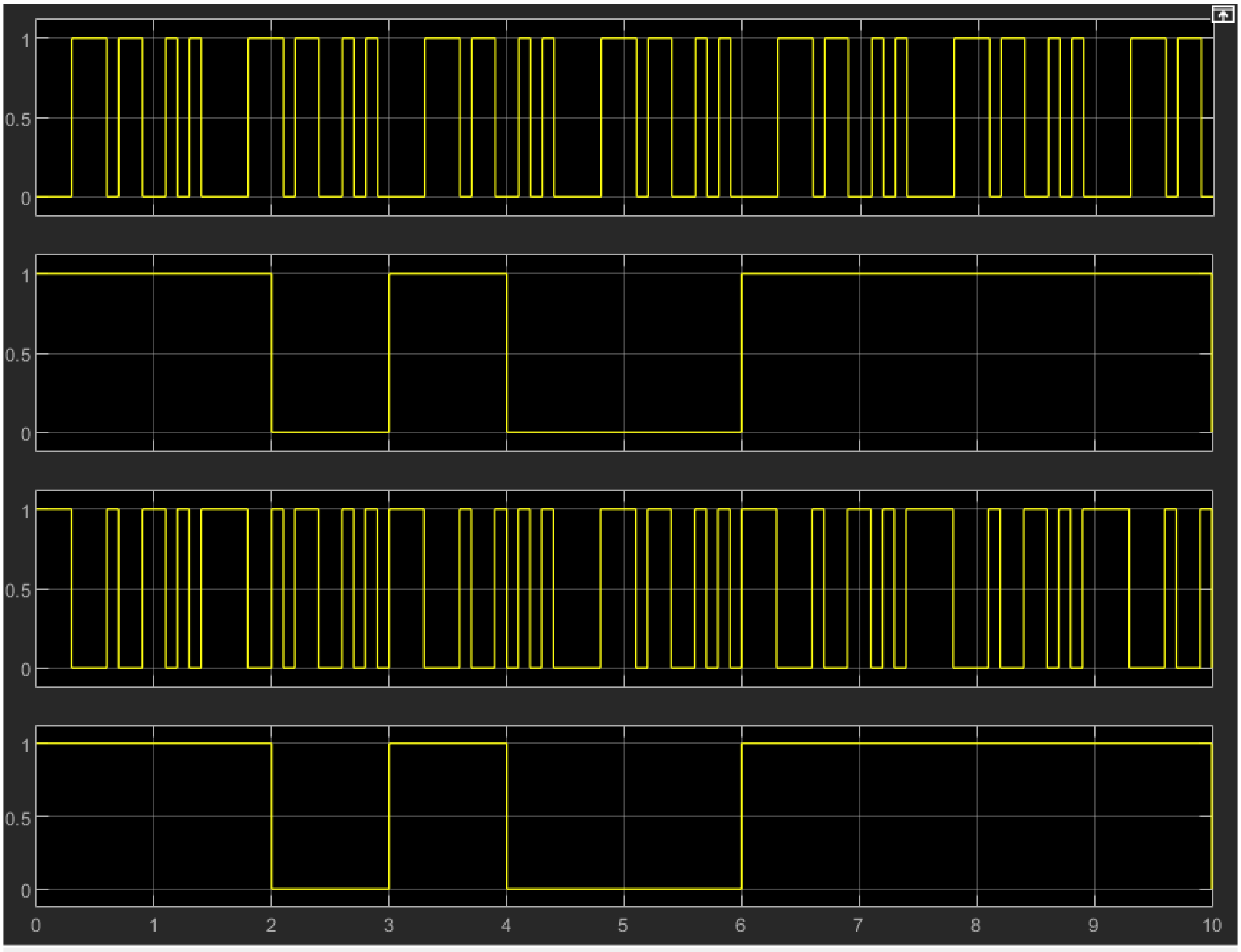


PN sequence for feedback polynomial $X^4 + X^3 + 1$, that repeats after every 15 bits.

Then after that the complete Scrambler And Descrambler implementation:



The waveforms:



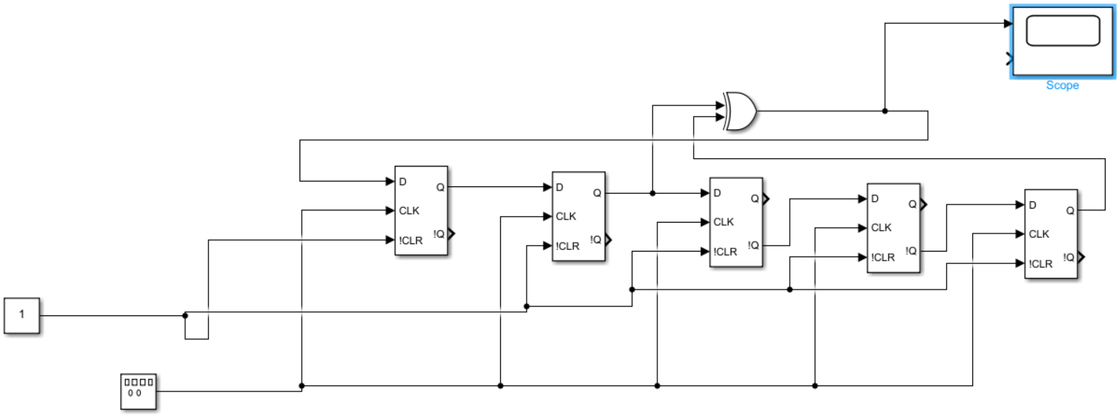
From top to bottom:

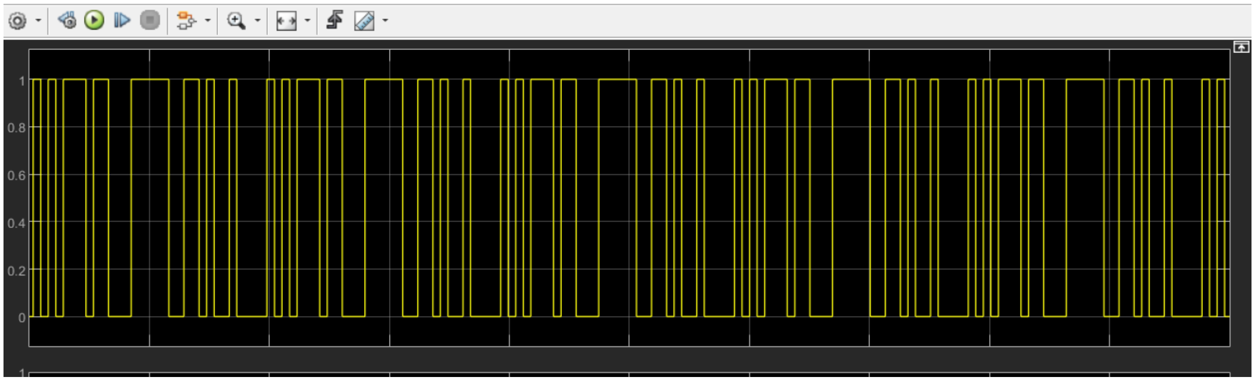
Topmost is the PN sequence.
Second from top: Input Signal
Third from top: Scrambled Signal
Bottom: Descrambled Signal

Here you can see that the Scrambling part removes the long sequences of 1s and 0s from the input signal and after Descrambling the scrambled signal we got our original signal back.

So this type of scrambler solves the communication problem that is to avoid long runs of 1s and 0s but this is not really secure as it is only 4-bit LFSR it takes someone only 8 attempts that is (2 * n, for LFSR) to find the sequence. So we then analyzed different types of sequences which could serve us better for both the problems.

- 5 - bit LFSR:

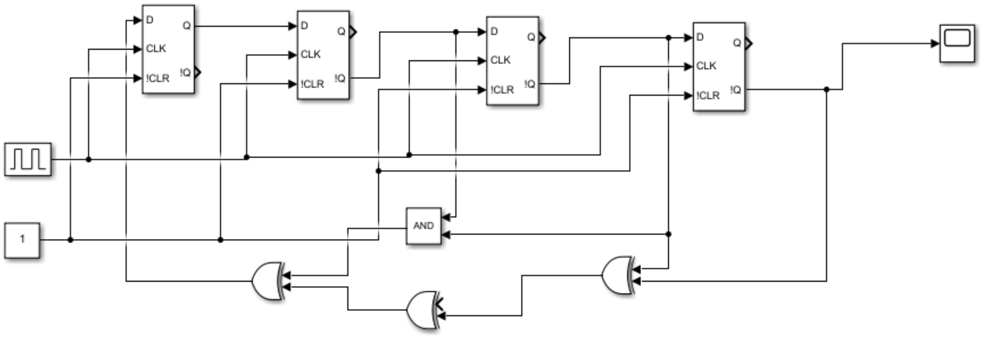


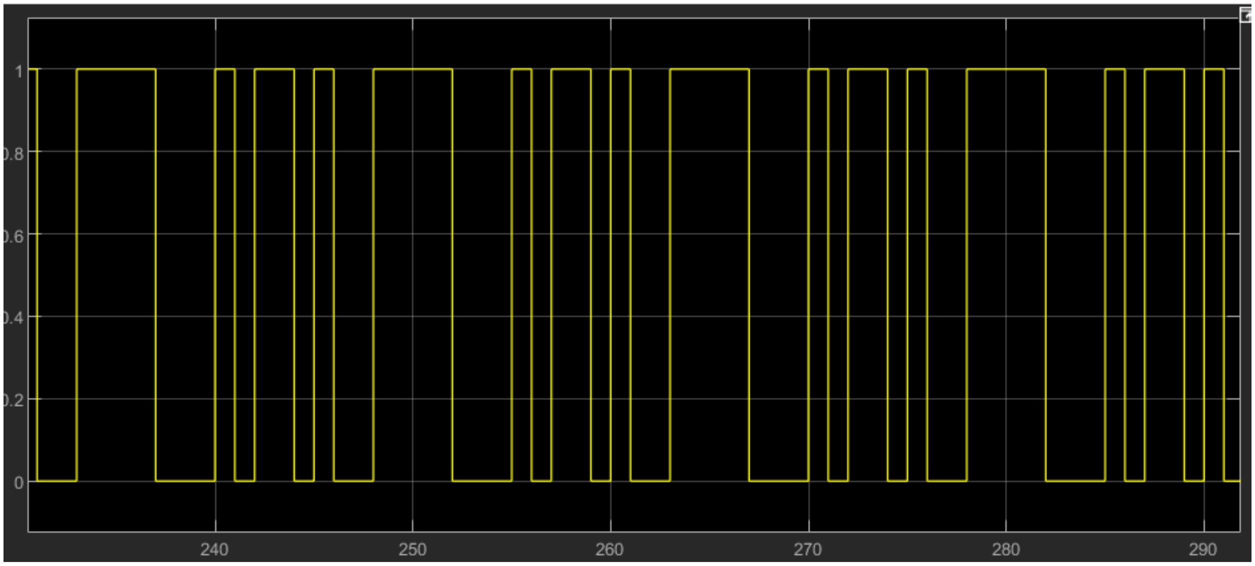


Here, increasing the number of bits in shift registers gave us longer sequences.

Then we moved to Non Linear Feedback Shift Registers:

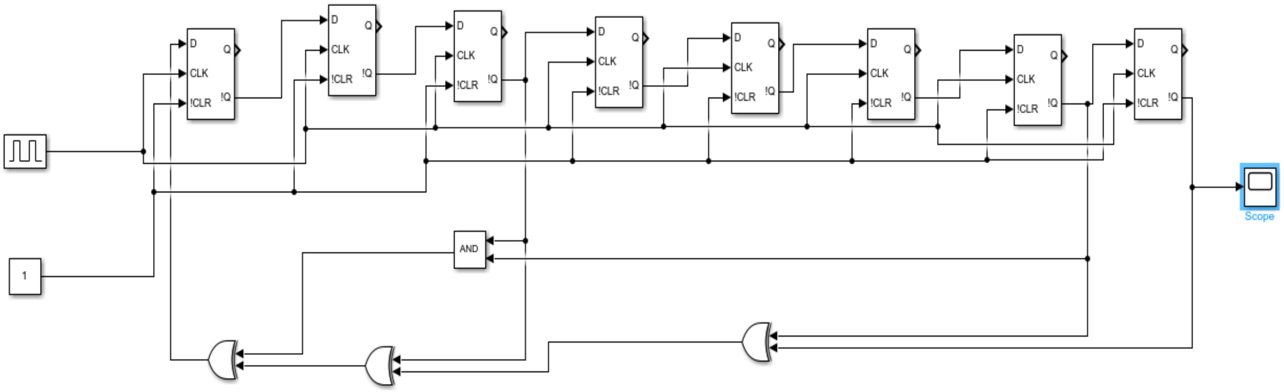
- 5-bit NLFSR:

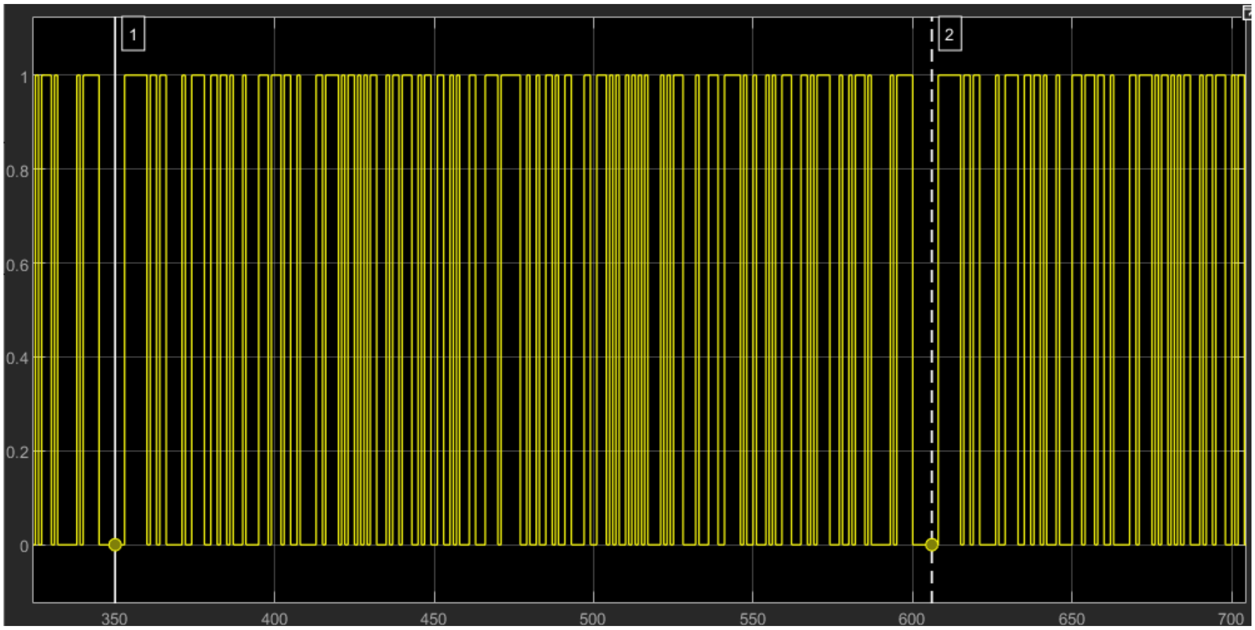




PN sequence
(Not of maximal length)

- 8-bit NLFSR:





PN sequence
(of maximal length, repeats after 255 bits)

Now, cracking a NLFSR is far more difficult than a LFSR as there is no specific algorithm to crack a Non Linear Feedback Shift register, but NLFSR does not have the desired correlation for a better PN sequence.

- We then tried implementing several important sequences like:
  1} JPL Sequence.
  2} Gold Sequence.
  3} Kasami Sequence.

And then finally implemented m-sequence, Gold sequence and JPL sequence on python for scrambling text based data.

3. Implemented a Scrambler and Descrambler for text based data on Python.

Followings are the sequence we implemented on python:
1} m-sequence
2} Gold Sequence.
3} JPL sequence.

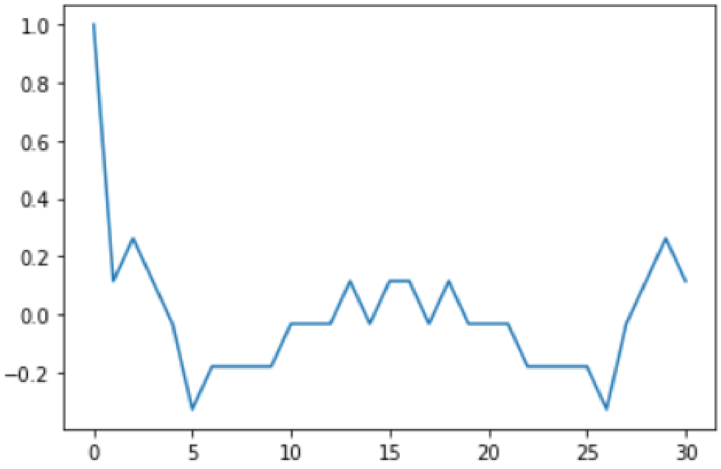
The python code we have written performs following functions:

1. Read text data. (Unicode 7)
2. Asks a level of secrecy from 1 to 5.
3. Generate a random sequence with physical entities depending on the level of secrecy chosen.
4. Returns a scrambled text data with a key generated by the hashed logic of the model implemented in the backend. This key is necessary to be supplied while descrambling this encoded data.
5. Finally Descrambling the data.

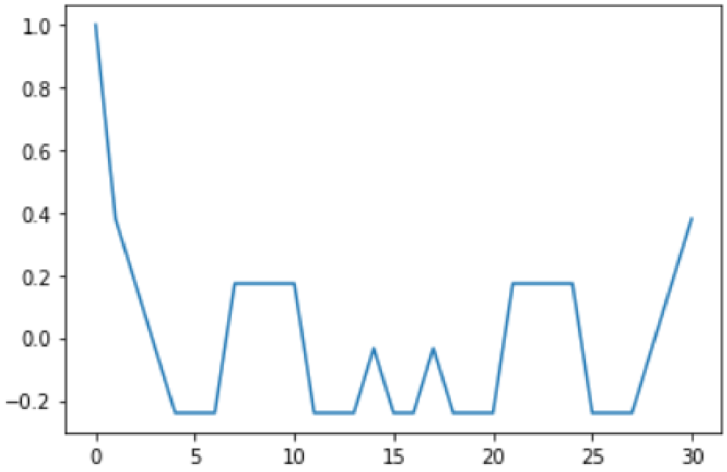
Plots:

Shifted correlation plot:

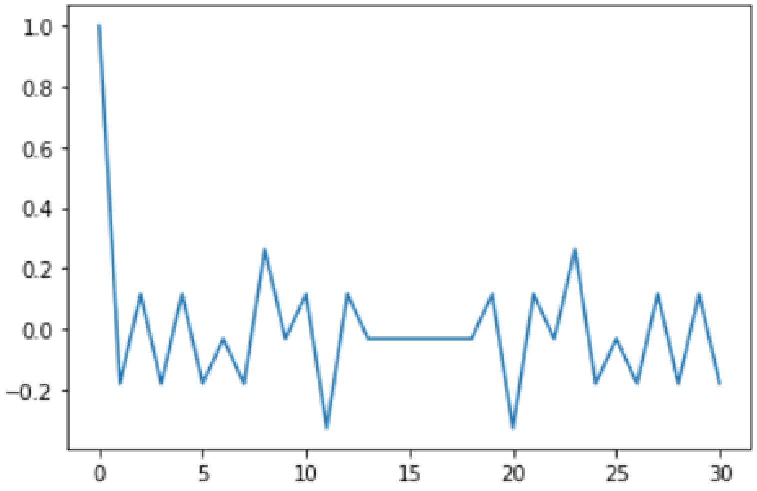
- Gold [3]



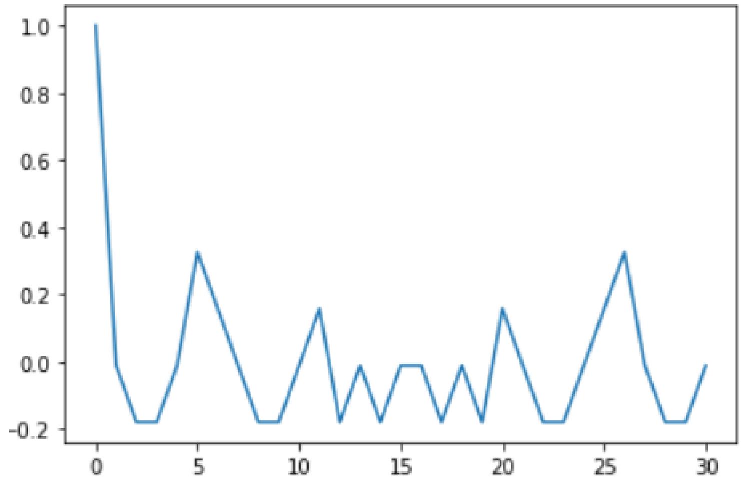
- Gold [4]



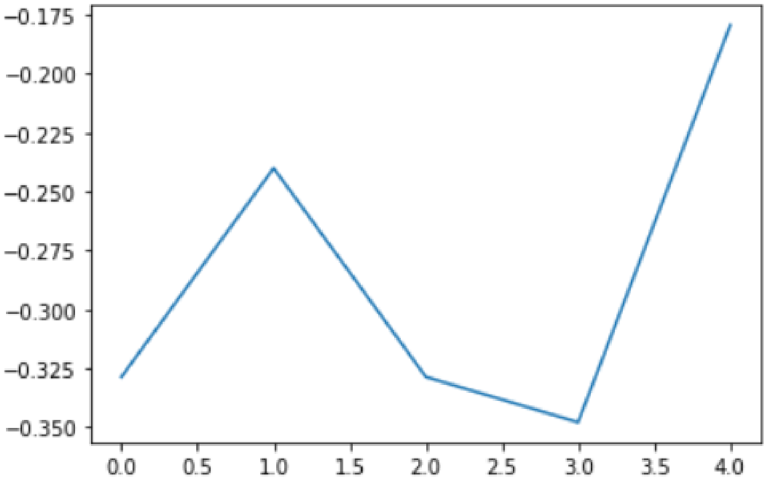
- Gold [5]


1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30

- Gold [6]

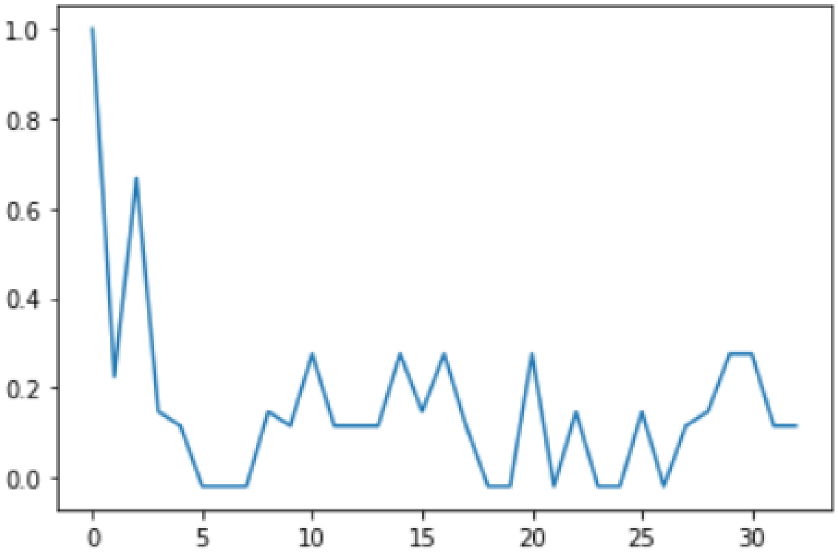

1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
5
10
15
20
25
30

- Gold [7]

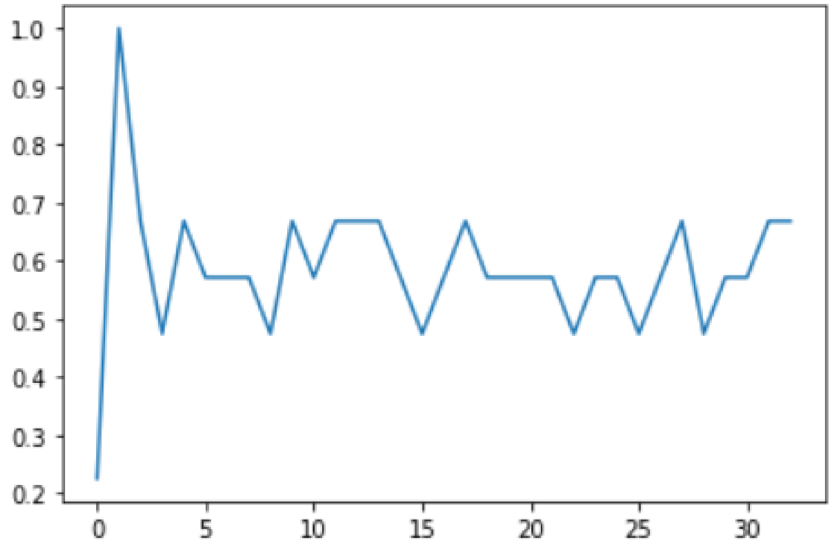


Relative Correlation plot:

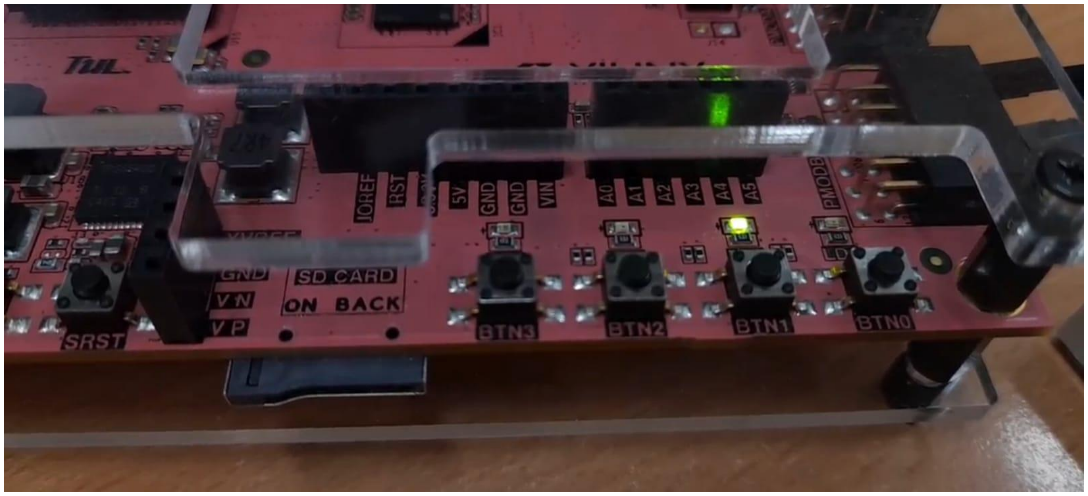
1} Gold [2:] vs Gold [0]:

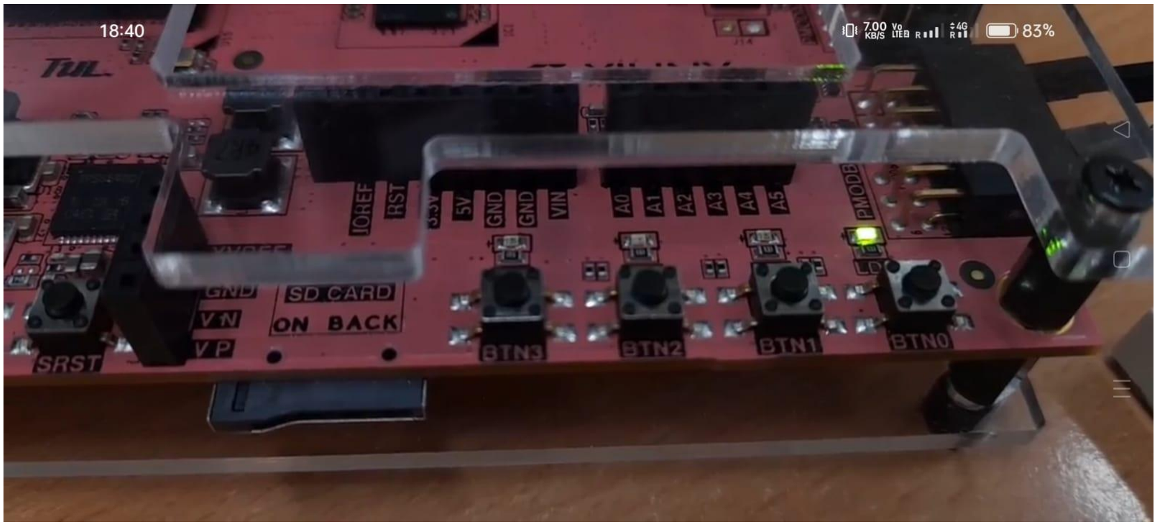


2} Gold [2:] vs Gold [1]:



4. Implemented on a FPGA board to get PN sequence:




18:40
83%
IOREF
RST
3.3V
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
PMODB
SD CARD
ON BACK
GND
VN
VP
SRST
BTN3
BTN2
BTN1
BTN0

CONCLUSION:

In our implementation we observed that the auto-correlation of m-sequence is better than the gold sequence. However the value of cross-correlation of the gold sequence is smaller than m-sequence. Thus the gold sequence has better cross-correlation than m-sequence.

We take a preferred pair of m-sequences which have better auto correlation and generate gold sequences by modulo-2-addition of a pair of m-sequences.

The m-sequence is better for single user communication, while gold sequences are used as chipping sequences that allows better multiple user communication with less interference. Therefore both the m-sequences and gold sequences have their own advantage at correlation and single or multi user communication.

Distribution of Work:

Shashank Kumar: Implementation on breadboard and FPGA and report making.

Pratham Chaurasia: Implementation on Python and FPGA and report making.

REFERENCES:

1. https://people.ece.ubc.ca/edc/4340.fall2014/lectures/lec12.pdf
2. https://en.wikipedia.org/wiki/Gold_code#:~:text=A%20Gold%20code%2C%20also%20known,are%20named%20after%20Robert%20Gold.
3. https://www.gps.gov/technical/icwg/IS-GPS-200J.pdf.
4. http://surl.li/bwuxd
5. https://www.researchgate.net/publication/221355010_Nonlinear_Feedback_Shift_Register_Sequences?enrichId=rgreq-3f5413b03517d5e65211919adc4377e4-XXX&enrichSource=Y292ZXJQYWdlOzIyMTM1NTAxMDtBUzoxMDQ3NzI3MTU2Nzk3NTRAMTQwMTk5MTE3MDQ4NQ%3D%3D&el=1_x_2&_esc=publicationCoverPdf
6. **Source Book: Sequences and their applications.**