ANALOG AND DIGITAL COMMUNICATIONS (THEORY)

PROJECT REPORT
QPSK SIMULATION USING LabVIEW

Final Project Report

Faculty: Prof. Ashok Ranade

Group Members:

Vidish Joshi
AU-1841019

Jeet Shah
AU-1841007

Harshil Mehta
AU-1841010

Devam Shah
AU-1841044

Introduction

Digital communication is widely used in all fields of communication. Different digital modulation schemes based on keying techniques are used for implementing digital communication systems. There are different types of keying like amplitude shift keying, frequency-shift keying, phase-shift keying, etc. In our project QPSK (Quadrature Phase Shift Keying), a type of phase-shift keying is implemented. Phase shift keying uses four points on the constellation diagram to represent the four phases. QPSK can encode two bits per phase. Here we have used LABVIEW (Laboratory Virtual Instrumentation Engineering Workbench) as the simulation platform. This programming environment being graphical gives a good visualization of the results.

QPSK is a constant amplitude M-ary digital modulation scheme. Since QPSK provides high performance on bandwidth efficiency and bit error rate, it is considered as the most widely used modulation technique in the digital communication system. Using QPSK, we can have a processing rate twice as that of a normal mode.

The flow in which we have implemented our project is:

INPUT Signal -> Modulation Using QPSK -> Transmission -> Demodulation -> OUTPUT Signal. We have tried to simulate the above flow in the LabView platform.

Theory

Basic Terminology:

- **Formatting:** Formatting refers to the conversion of an input signal into a sequence of [1] binary digits.
    - Analog signals -> Sampling, Quantization,Encoding
    - Textual data ->Encoding (ASCII code)

- **Modulation:** Modulation refers to the conversion of binary digit sequence into waveforms suitable for transmission on the chosen medium.

  For eg:

Binary Digit Sequence	Respective Waveform
00	$s_1(t)$
01	$s_2(t)$
10	$s_3(t)$
11	$s_4(t)$

- **Demodulation:** It is the exact opposite process of modulation.

PSK Modulation:

Phase-shift keying (PSK) is a digital modulation process that conveys data by changing (modulating) the phase of a constant frequency reference signal (the carrier wave). I will briefly describe the two types of PSK modulation we have used.

BPSK (Binary Phase Shift Keying) modulation:

Binary Digit Sequence	Respective Waveform
0	$s_1(t)$
1	$s_2(t)$

$s_1(t) = A\cos(w\_c\, t)$
$s_2(t) = A\cos(w\_c\, t + \pi)$
$= - A\cos(w\_c\ \ t)$

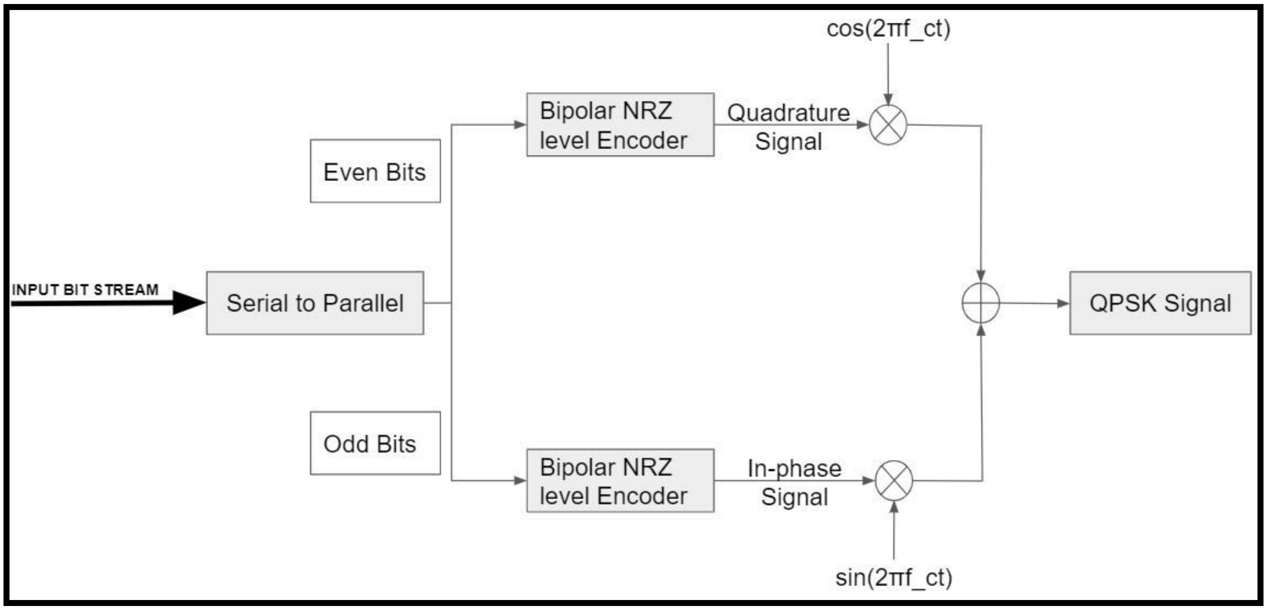
QPSK (Quadrature Phase Shift Keying) modulation in its idea is similar to the previously discussed BPSK modulation. The difference is using four possible phase shifts instead of two. Four offsets numbered in a binary system require two bits. Therefore, two bits are encoded in one carrier period. Thus, with the same carrier frequency, QPSK modulation allows for twice as fast data transmission than BPSK. In a typical case, when the probabilities of occurrence of each of the two bits (00, 01, 10, 11) are equal, phase shifts are arranged symmetrically. These can be, for example, 45 °, 135 °, 225 °, and 315 ° angles.

Binary Digit Sequence	Respective Phase
10	45 °
11	135 °
01	225 °
00	315 °

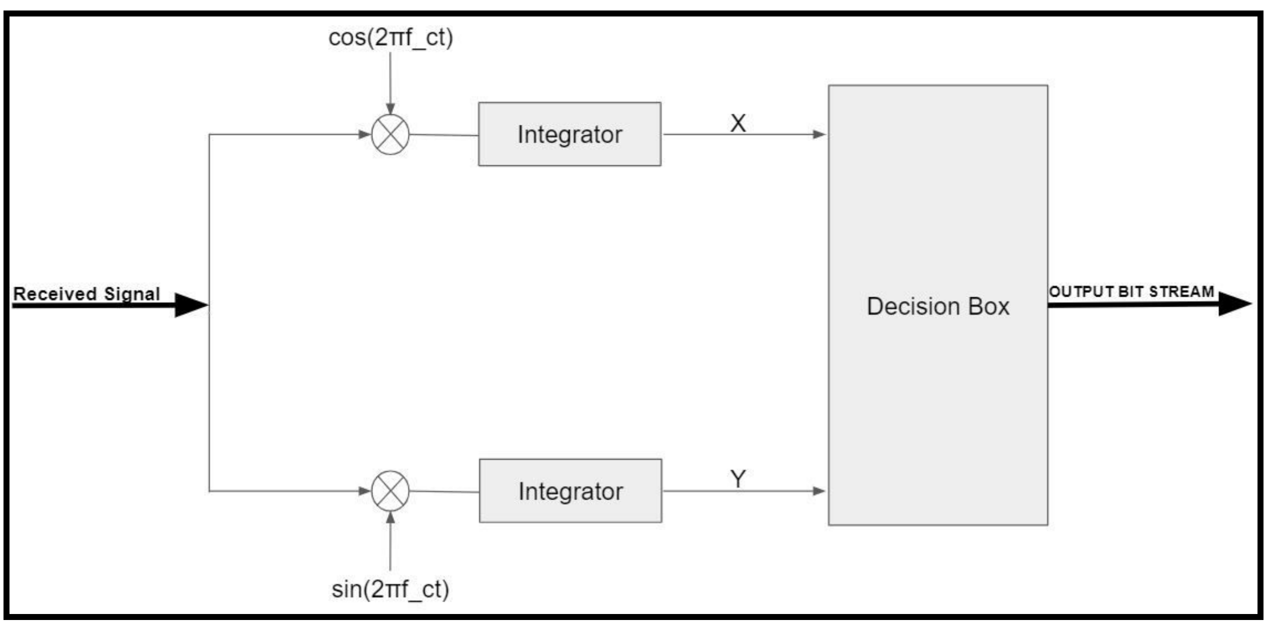
QPSK modulation can be done in many ways like- Conventional QPSK, Offset QPSK, π/4-QPSK. We have used the π/4-QPSK method which is described below.

In the π/4-QPSK method, the input bitstream is separated into 2 different streams of odd and even bits. Both these streams are modulated using BPSK modulation and then the modulated signals are added. The resultant signal is QPSK modulated with phase angles 45°, 135°, 225° and 315°.

Modulator and Demodulator Block Diagram:



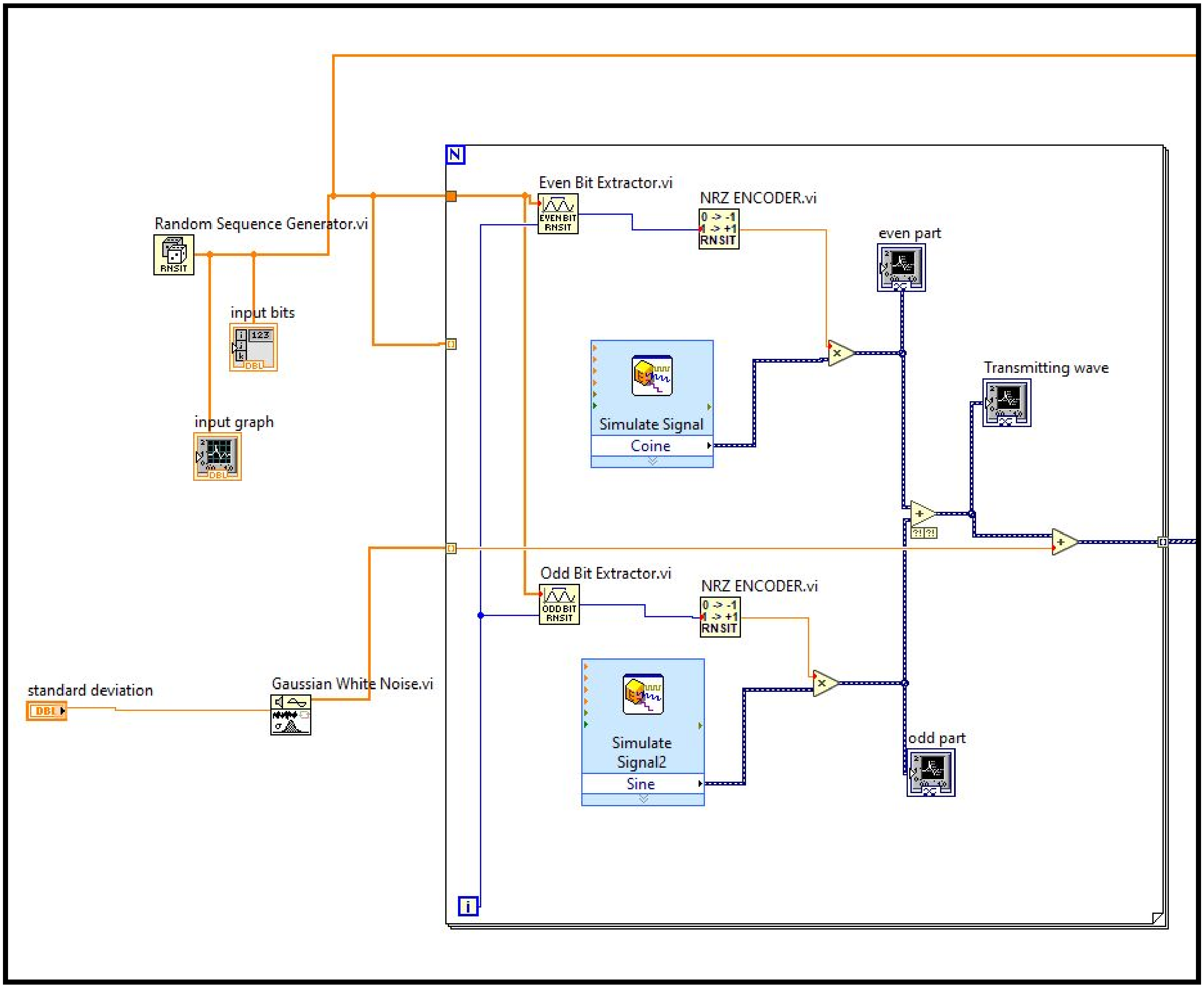
QPSK Modulator



QPSK Demodulator

LabView Simulations:

- **Modulated Wave Generation**



1. We are generating a Random digital signal using a random sequence generator. In which a random number between 0 - 1 is generated and then is rounded off to the nearest integer value, later stored in an array.

2. Which is then passed through even and odd bits extractor. Bits at 0 and 2 are extracted in an even array while bits at 1 and 3 are in an odd array.

3. Then through NRZ encoder(Non-returning to zero). In this the are values are compared with 0, if equal to zero then "**-1**" is passed, else "**1**" is passed .

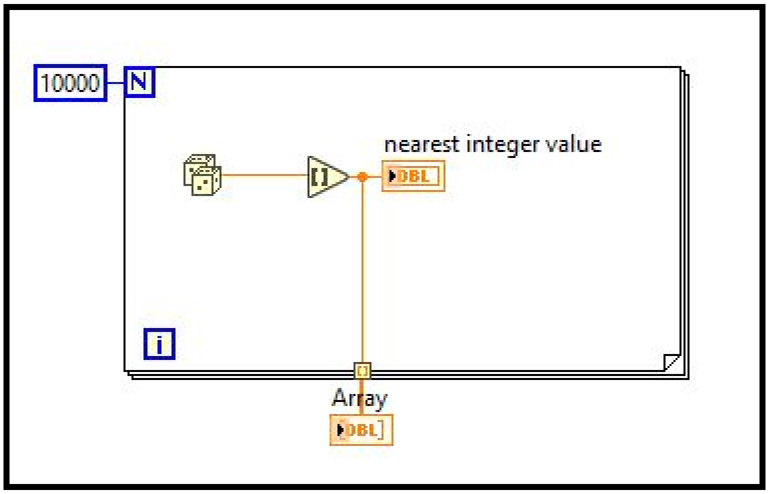
4. Then after the signals are multiplied by cosine and sine carrier waves respectively for even and odd. Rest details about the signals are given later in the snapshots. This are BPSK signals

5. Later both BPSK signals are added to get the QPSK transmission waveform.

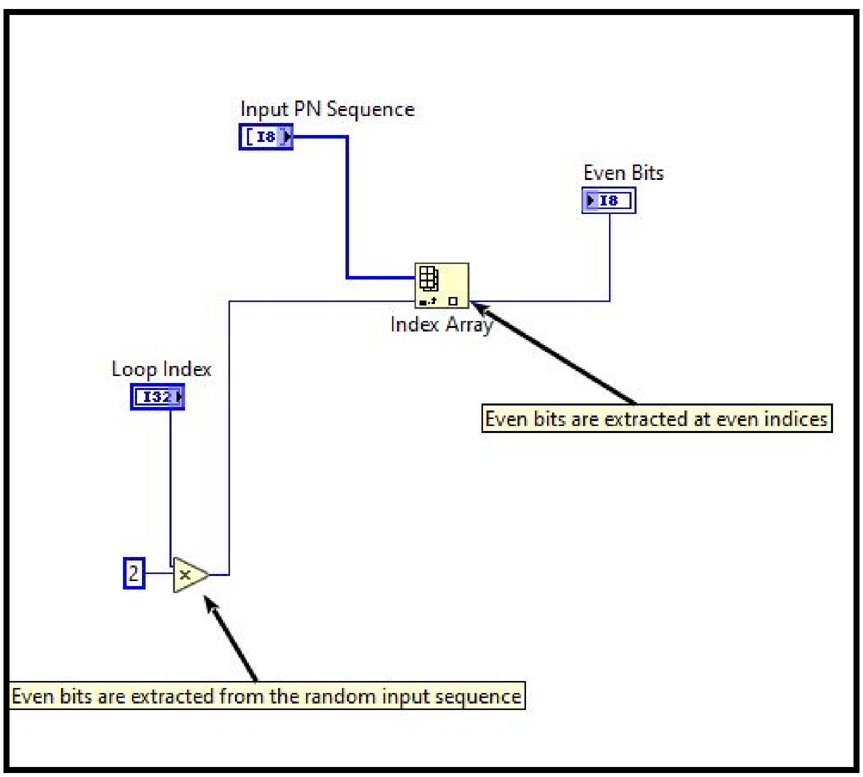
6. **Finally, white Gaussian noise is added to the transmission signal to simulate channel characteristics.**

Sub VIs of Modulated wave generation -

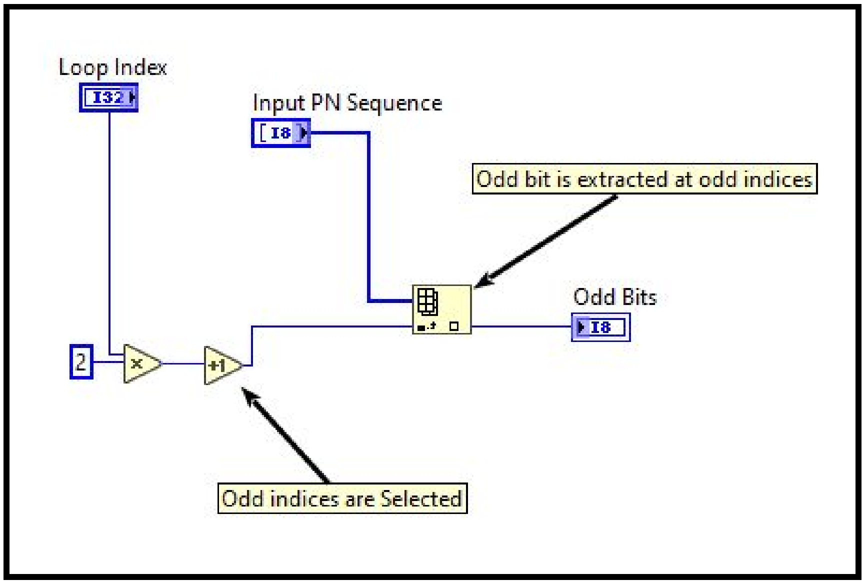
- **Random Sequence generator**



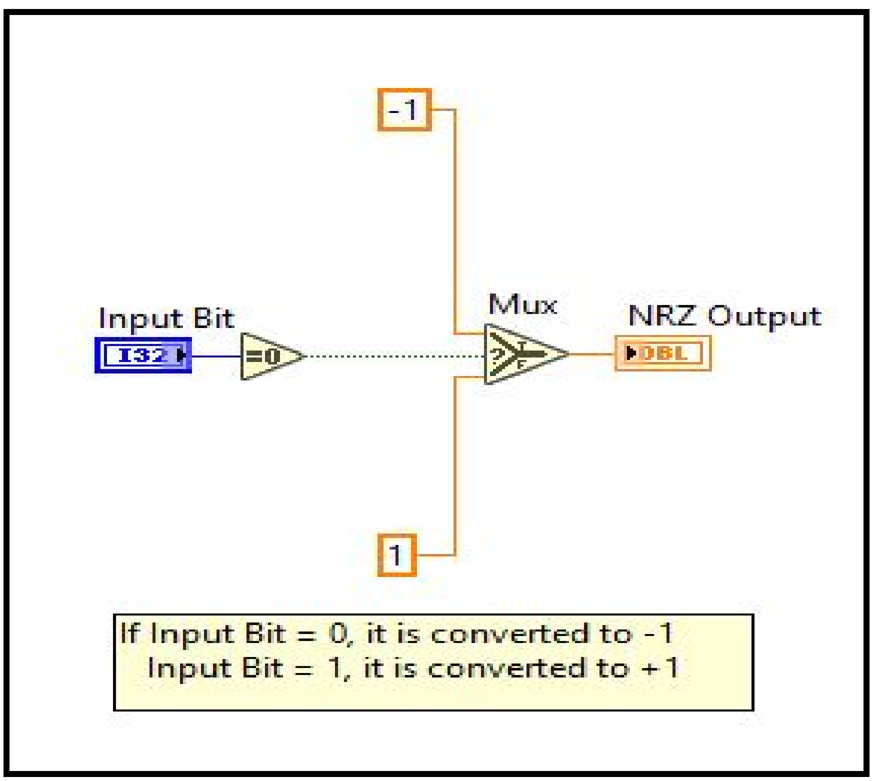
- **Even bit extractor**



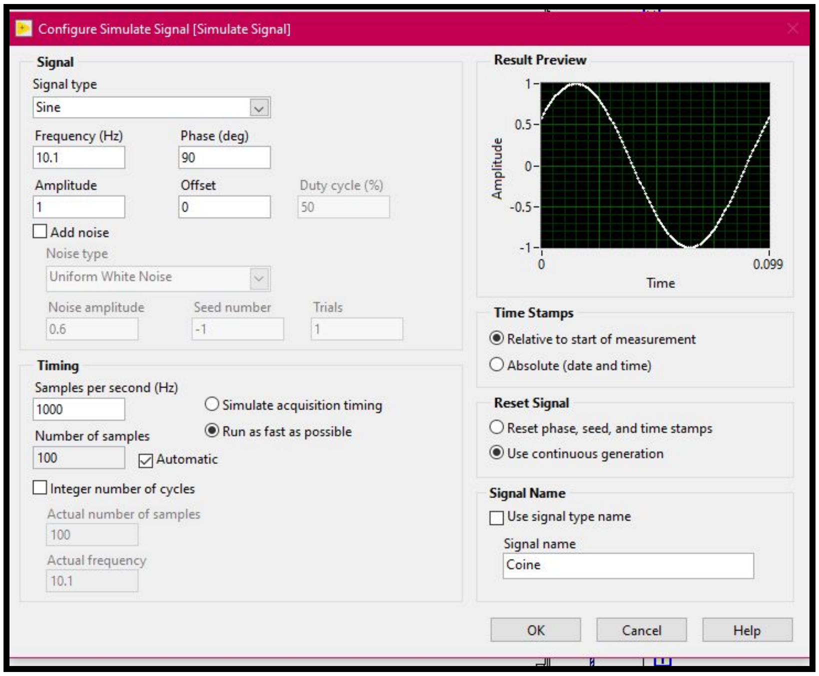
- **Odd Bit Extractor**



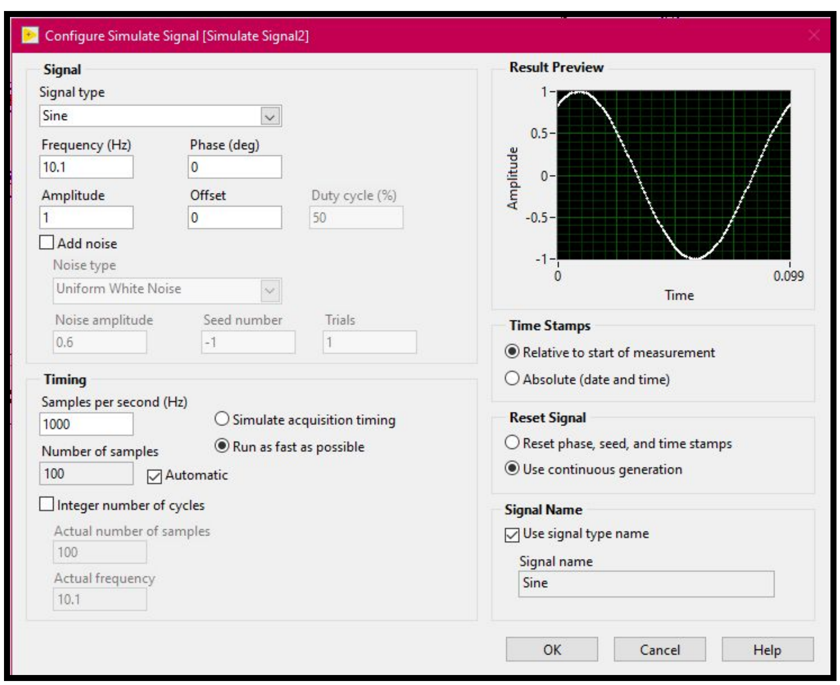
- **NRZ Encoder**



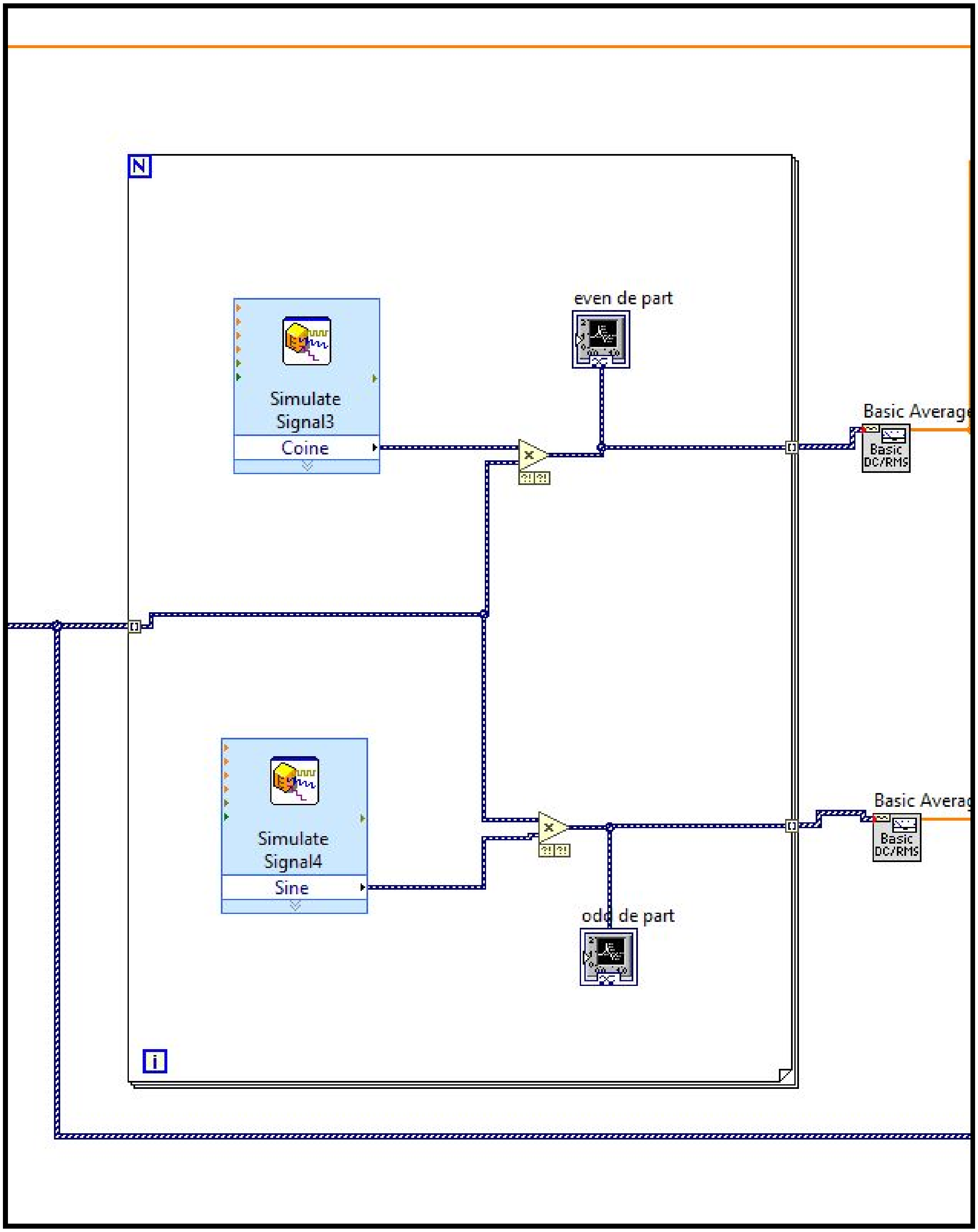
- Cosine wave configuration



- Sine wave configuration



- **Demodulation block:**



Demodulation Process

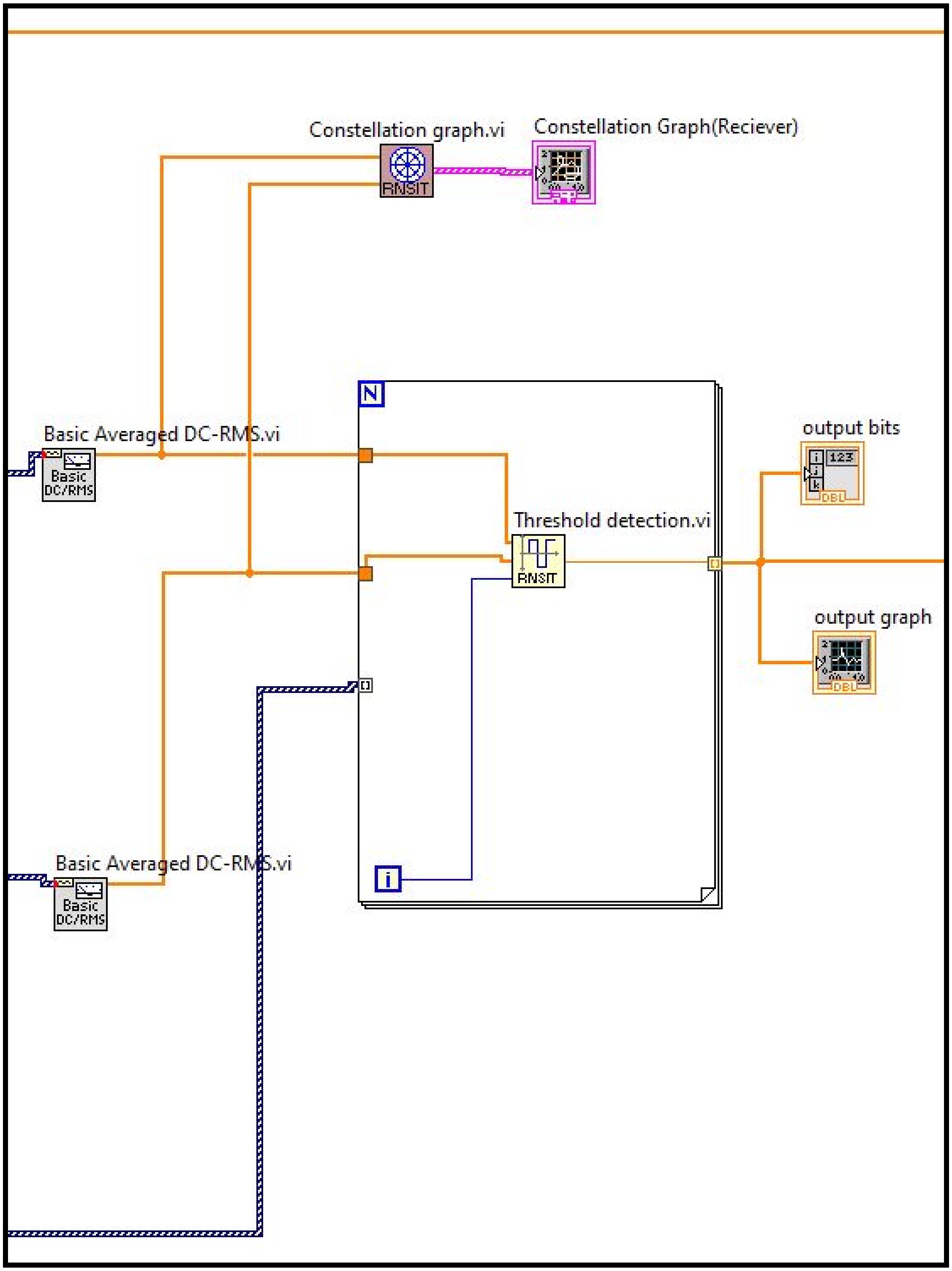
1. The transmitted wave is given in another for loop and multiplied by carrier waves, i.e Cosine for even part and to Sine for getting the odd part.

2. Later we need to integrate the and take the D.C value to produce “X” and “Y” values, this done by the **“basic-average D.C. RMS”** block given in the signal processing.

3. We take DC values because we are concerned with only the signs of integrated values to make the decision for obtaining the bit sequence.

4. Then these X and Y values are passed to the decision-maker circuit.

- **Decision-maker circuit**

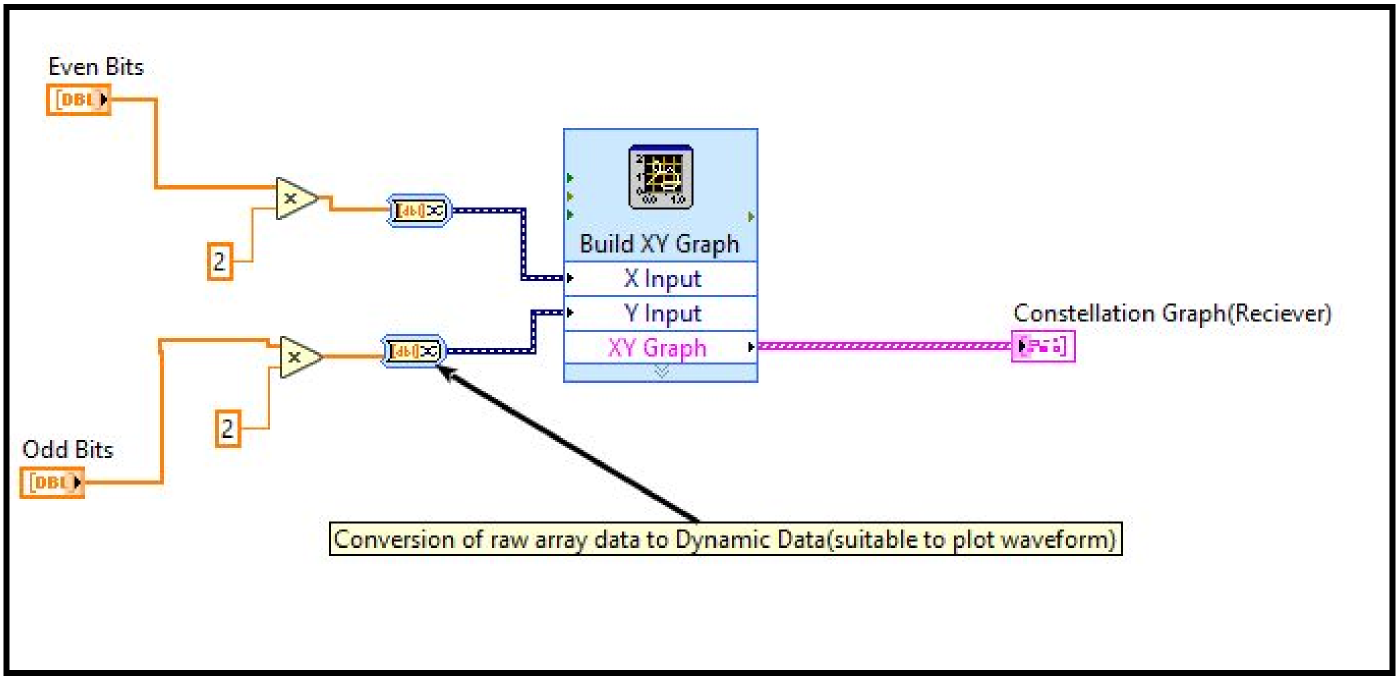


1. Based on the sign of X and Y from the demodulator circuit, we need to decide the bit sequence. It is based on the given table.

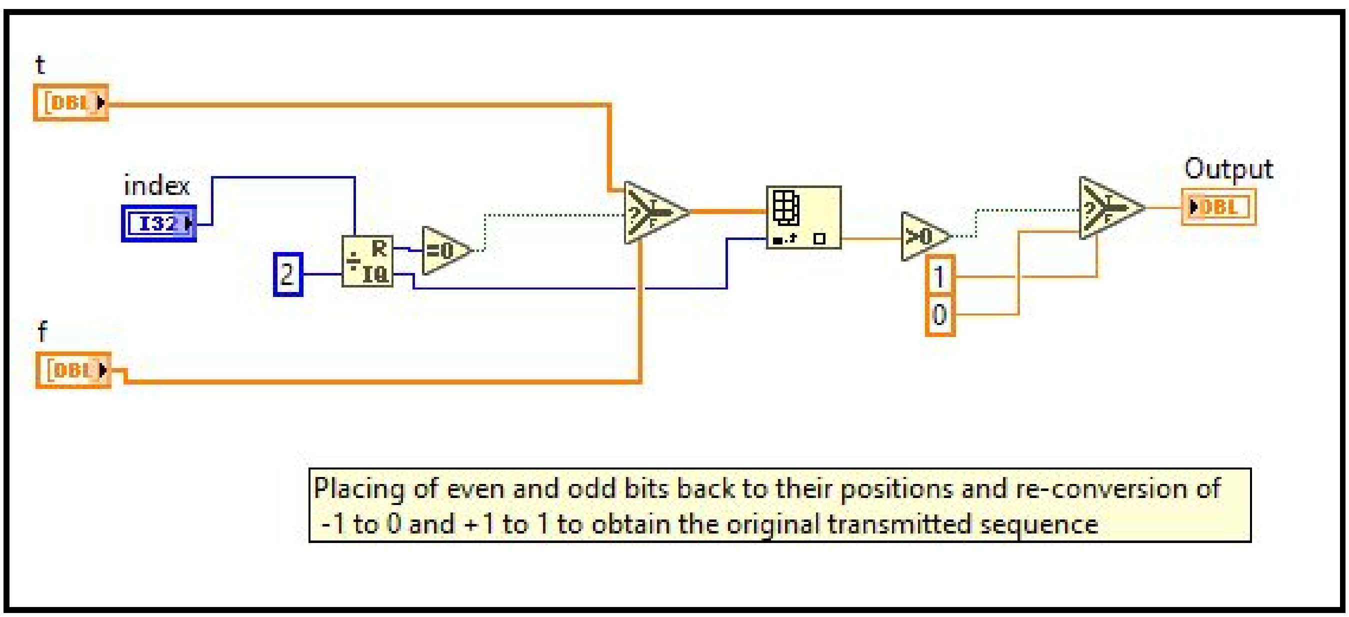
Sign of X	Sign of Y	Bit Sequence
+ve	+ve	11
+ve	-ve	01
-ve	+ve	10
-ve	-ve	00

Sub VIs of Decision-maker circuit -

- **Constellation Graph**



- **Threshold Detection**



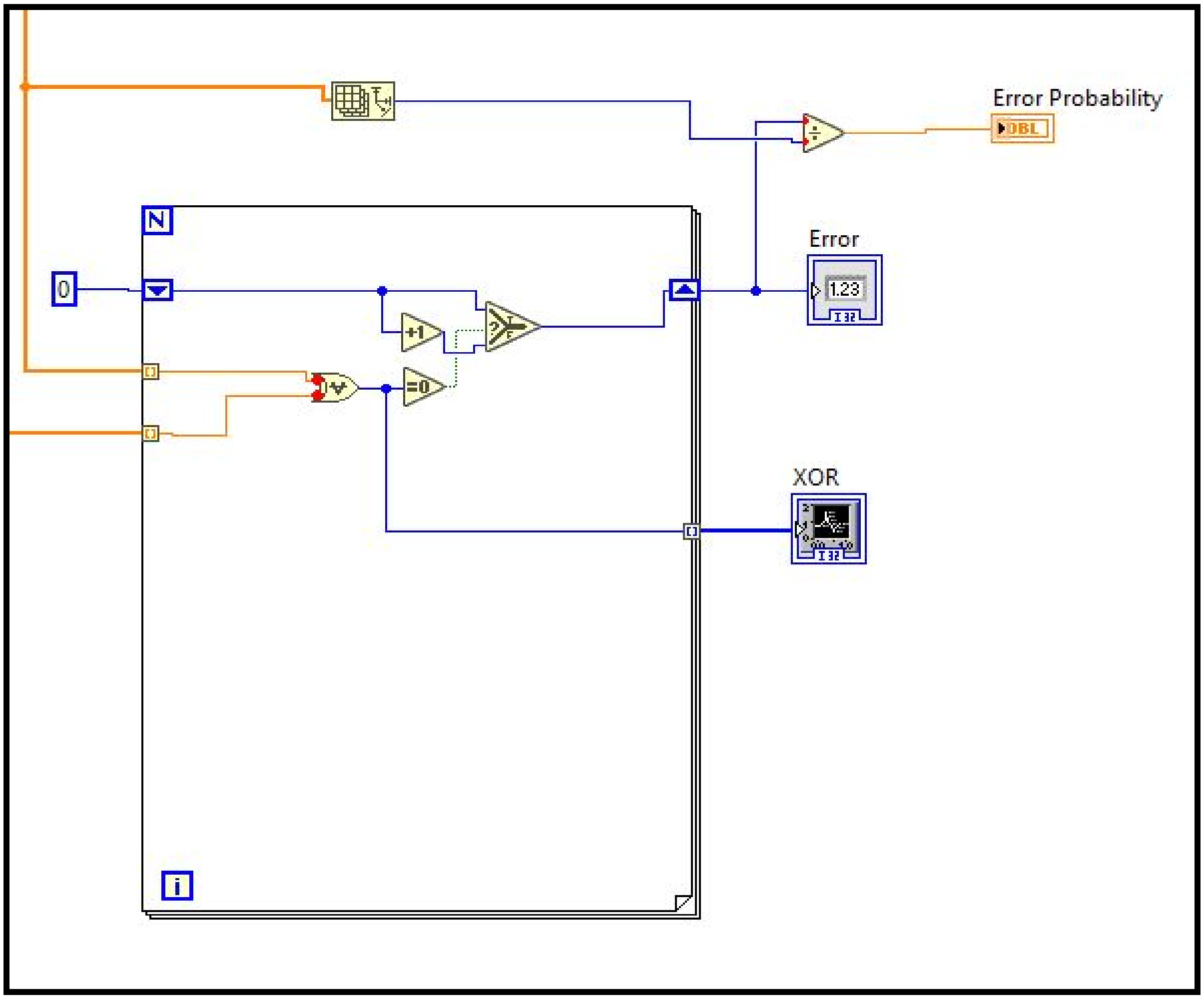
- **Bit Error Rate(BER) calculation:**

The transmitting signal encounters noise when passing through the transmission channel. In this simulation, this noise is represented by the Gaussian noise function of LabVIEW. The standard deviation control of this function allows us to set the deviation/randomness of noise in our channel.

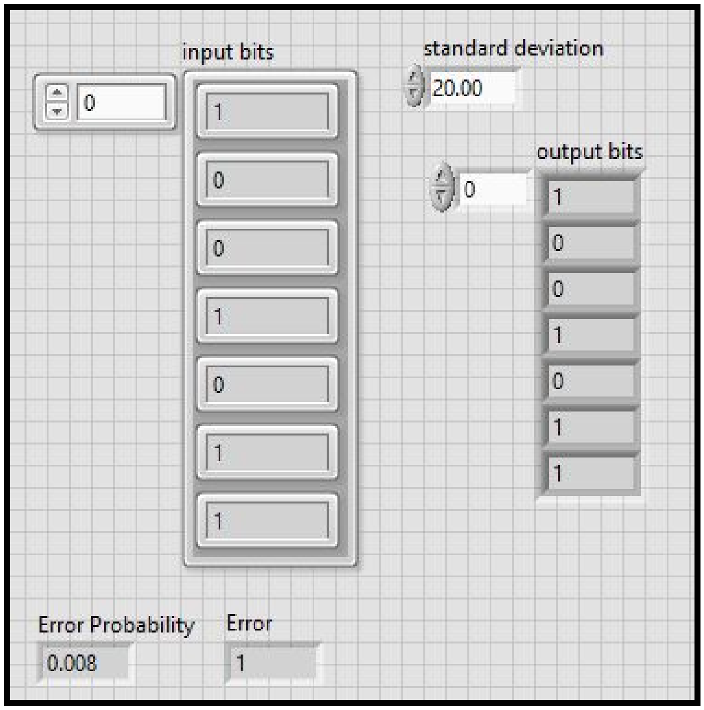
We have calculated the BER by doing "**XOR**" of the input and the output signal. By doing so we can get the error bits as the XORing different bits show "**1**" and the same bits show "**0**". Thus the number of 1's is total error bits.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

We can calculate the error probability of transmission by dividing the total number of error bits(obtained from XORing) with the total number of transmitted bits.


Error Probability
DBL
Error
1.23
I32
0
N
+1
=0
XOR
I32
i

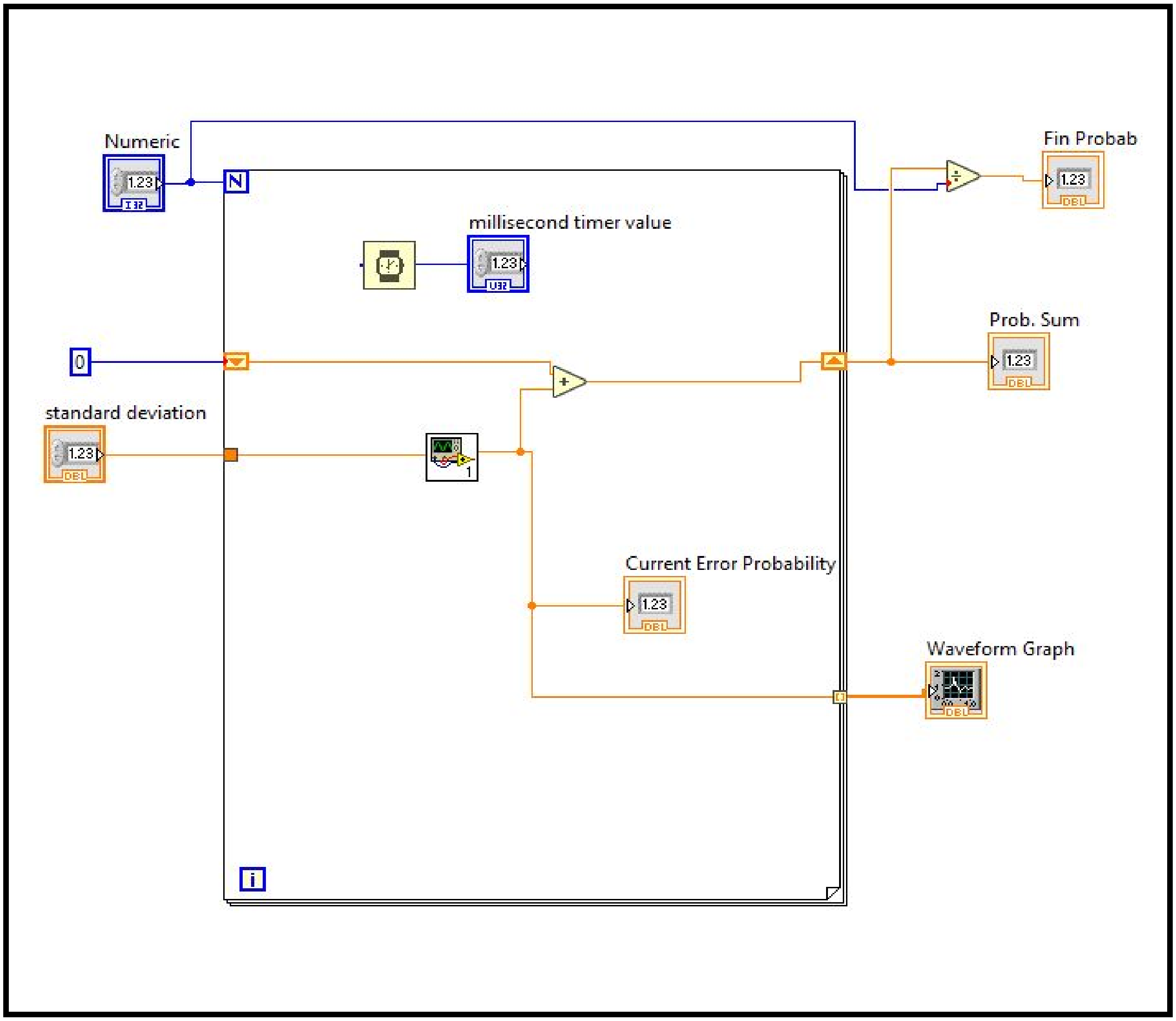
Below is the front panel representing the above implementation.



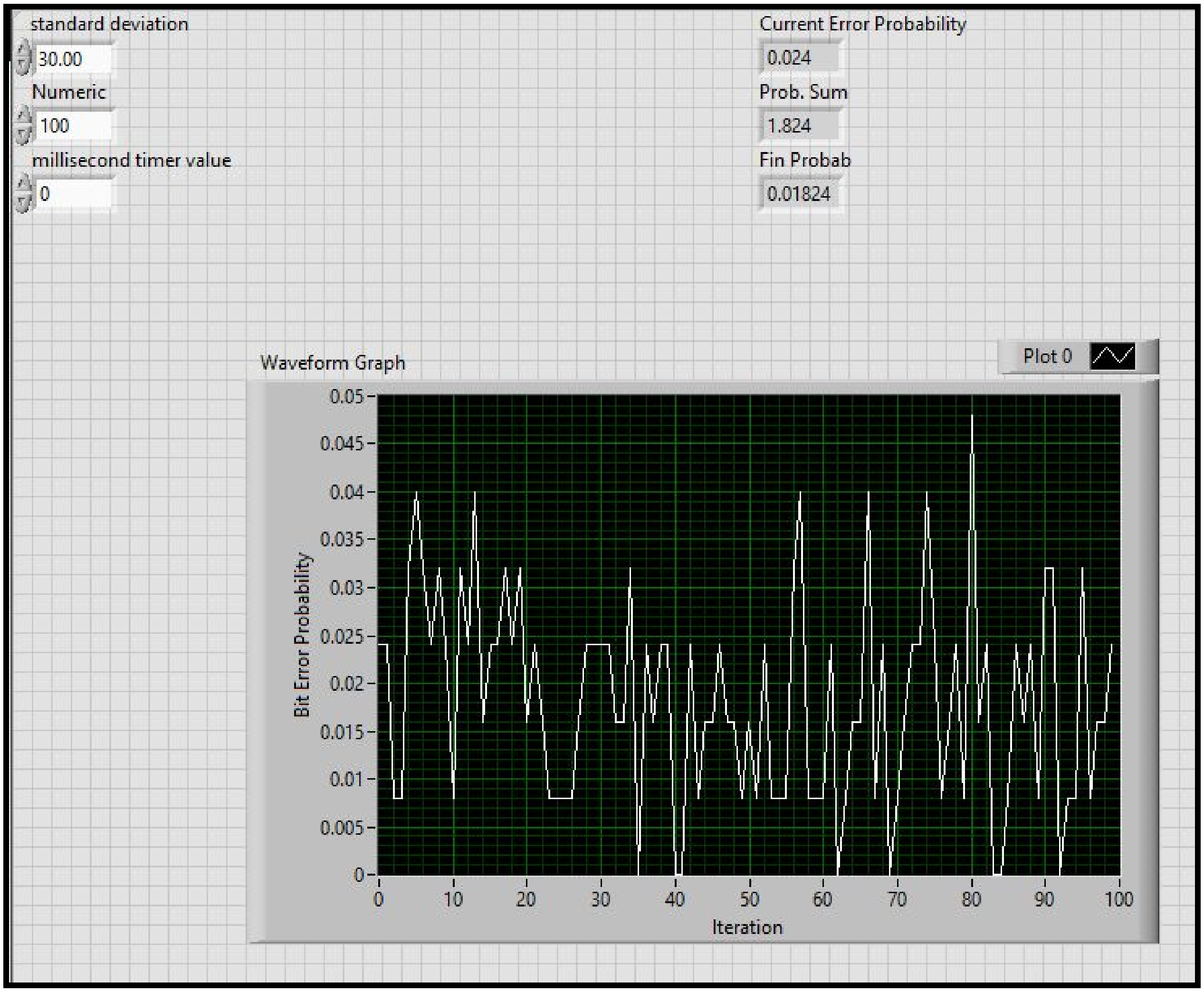
In this simulation system, we can transfer a maximum of *125 bits* per transmission. The above snippet shows that when the standard deviation of the noise function is kept 20, then there is 1 bit in error out of the 125 transmitted bits, resulting in BER to be 0.008.

But to get a better confidence in BER, we take the mean BER for a particular value of standard deviation by running the simulation for fixed SD value for N(here: 100) iterations. We then take the mean BER from this N simulations to assign a probability value for transmitting a signal.

- **Block for calculating average Bit Error Rate(BER):**



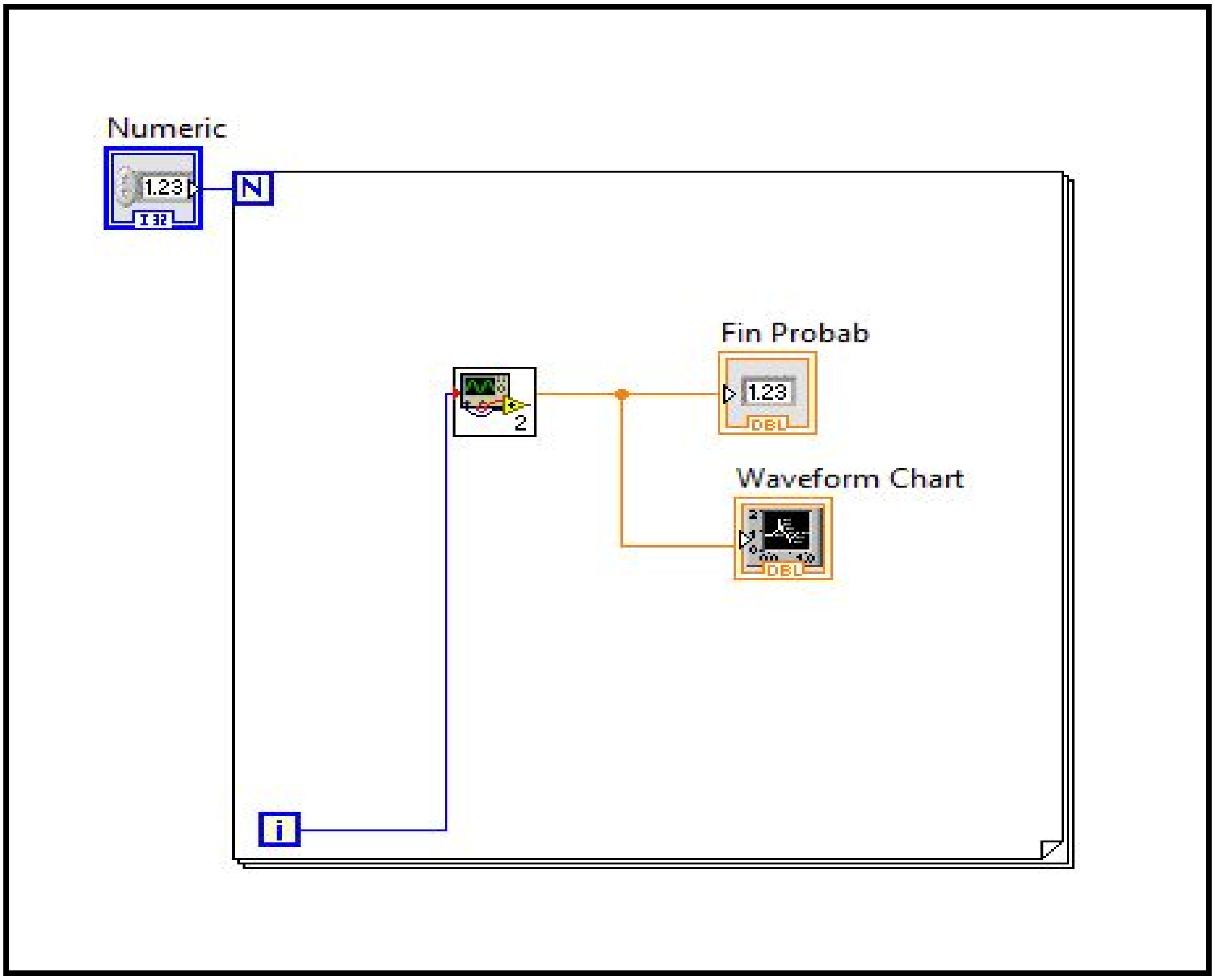
Below are the results of simulating the above block diagram and calculating the bit error probability for standard deviation value 30 for 100 iterations.



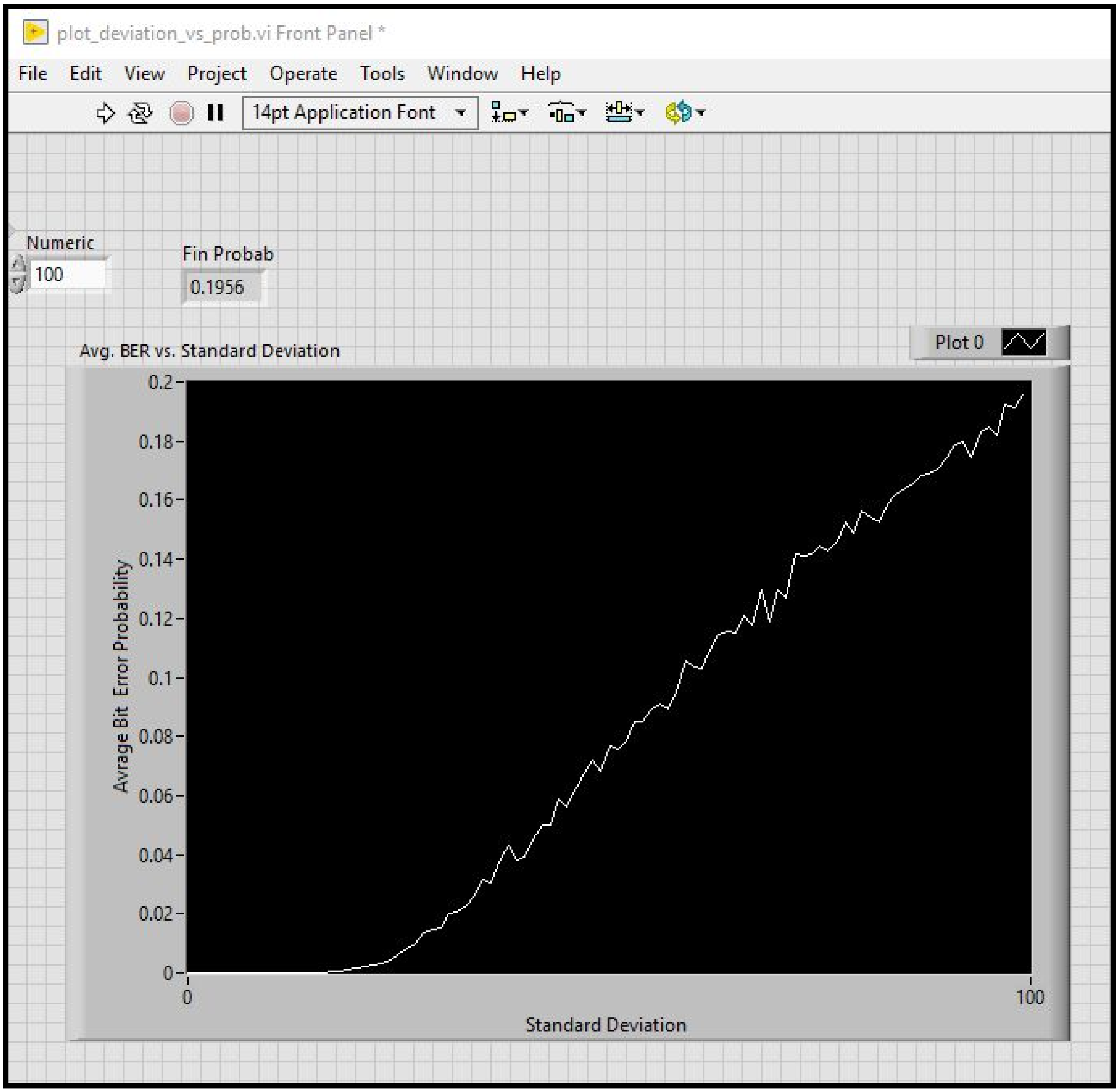
Here the graph represents the different values of error probability in the 100 executed iteration. The *'Fin Probab'* control value shows the mean value of probability which in this case turned out to be 0.01824.

To observe the behaviour of change in value of mean Bit Error Probability with change in value of standard deviation of noise, we run the above block diagram multiple times while changing the SD values and plot it. It is observed that with increasing the standard deviation of gaussian noise, the BER also increases.

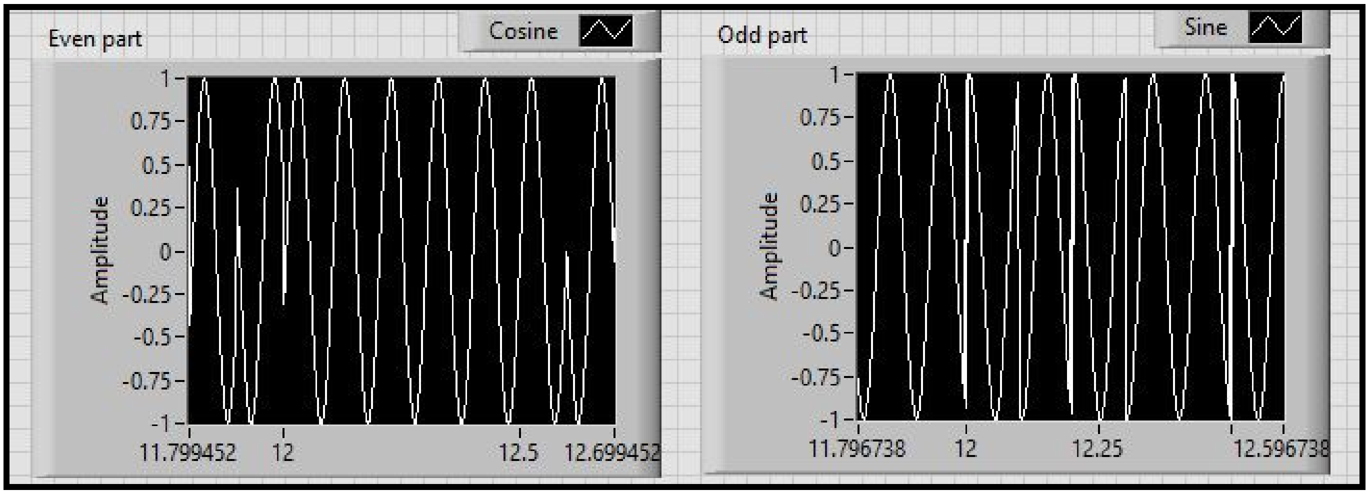
Below is the block diagram representing the above implementation.



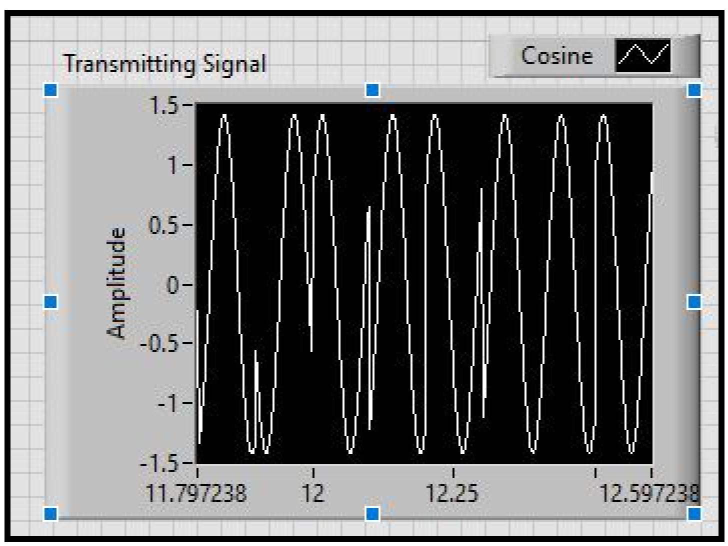
The output of running the above block 100 times for standard deviation values from 0 to 100 is shown below. As mentioned, the graph has a positive slope.



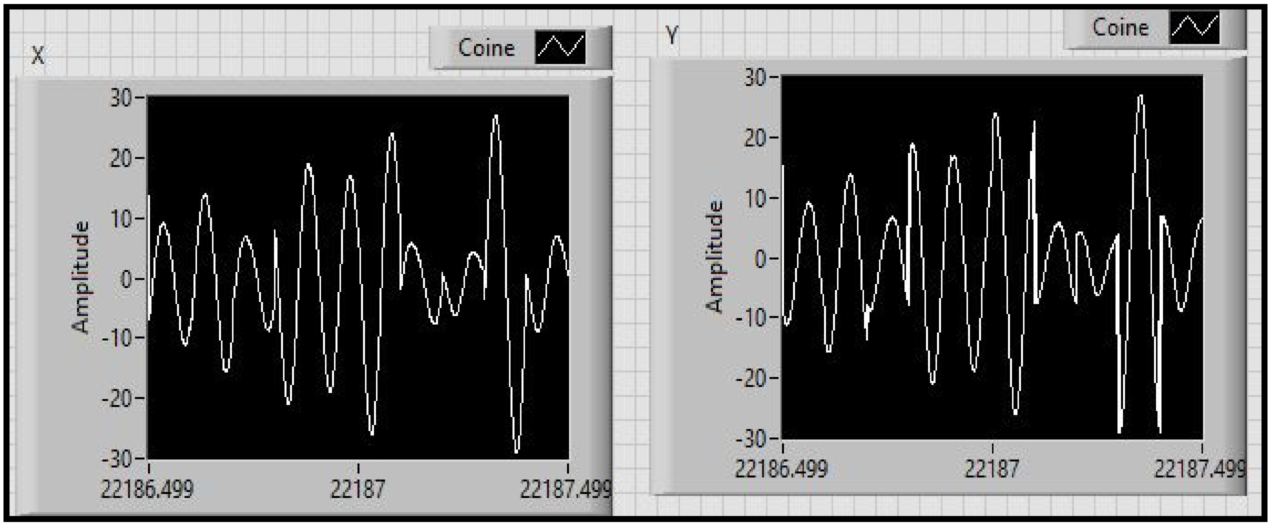
Simulation Results:



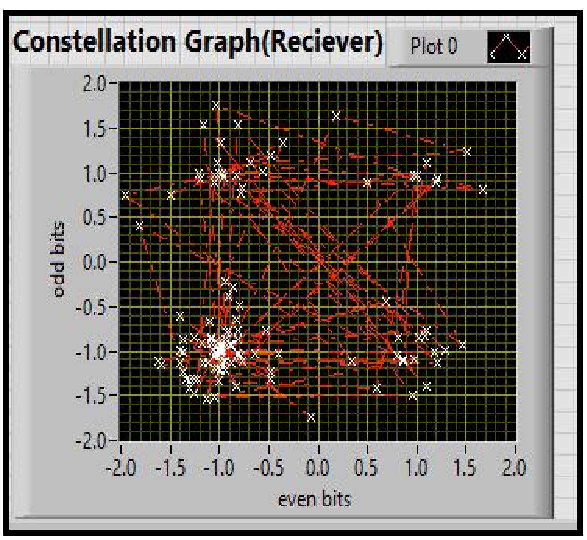
Results of even part and the odd part after passing through NRZ and multiplying with carrier signals.(Modulation Block)



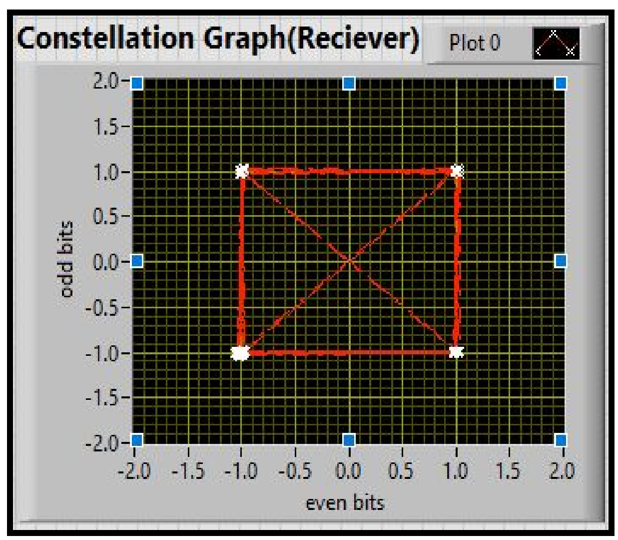
QPSK Signal without noise(Modulation Block)



Output signals X and Y that go into the decision block after multiplying it the sin and cosine carrier waves again.

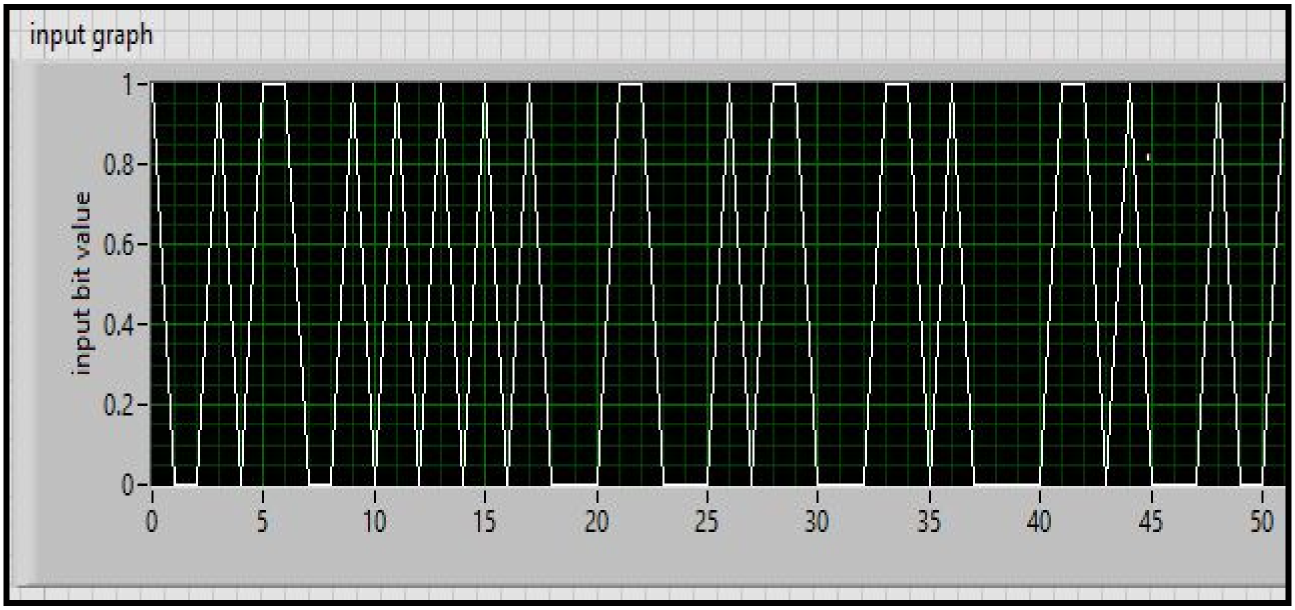


Standard Deviation of Gaussian noise 30.

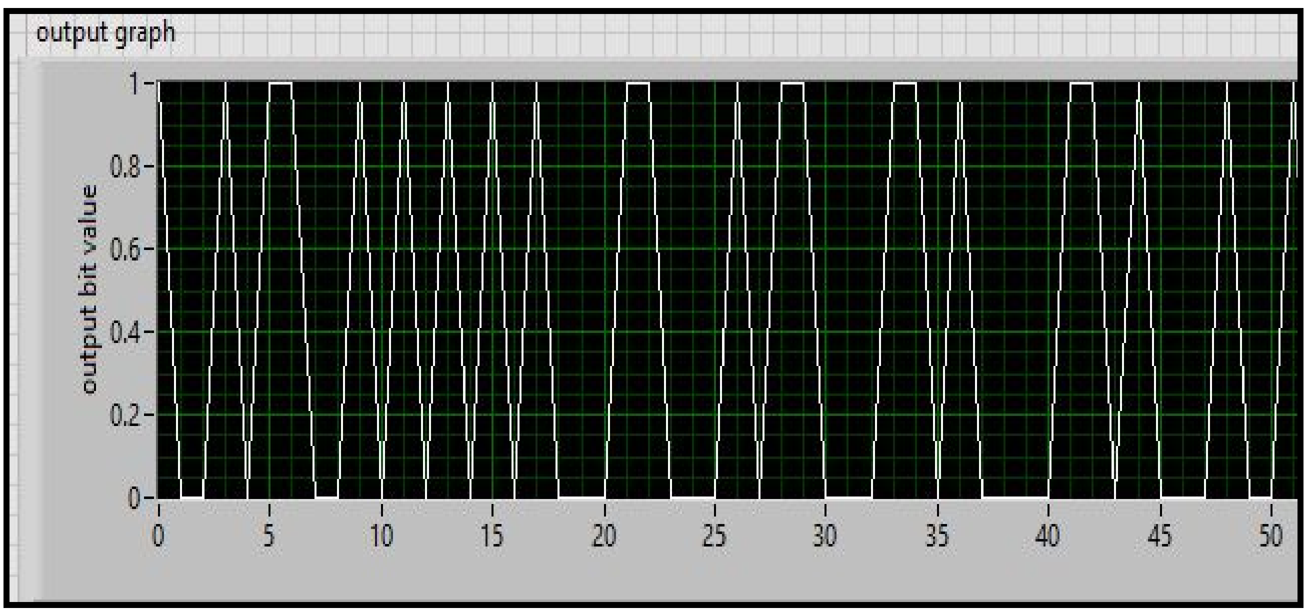


Standard Deviation of Gaussian noise is Is 1.

These graphs are simply plotted by plotting the even and the odd values on X and Y-axis respectively. By doing so we get a rough constellation diagram for QPSK with points at 45°, 135°, 225° and 315° angles.



Input Bit Stream



Output Bit Stream

Simulation Video

https://drive.google.com/drive/folders/1gxqLhxgSwhTxtidbEu_Rt45zRwmDIKJd?usp=sharing