Pabna University of Science and Technology

Faculty of Engineering and Technology
Department of
Information and Communication Engineering (ICE)
Lab Report

Course Code: **ICE-3206**
Course Title: Digital Communication Sessional

Submitted By: **Name: Md. Maruf Hossain** **ID: 170626** **Session: 2016-17** **Dept. of Information and Communication Engineering.** **Pabna University of Science and Technology.** **Date of issue:** **Date of Submission: 03-07-2021**	***Submitted To:*** ***Tarun Debnath*** *Lecturer* ***Taskin Noor Turna*** *Lecturer* **Dept. of Information and Communication Engineering.** **Pabna University of Science and Technology** **Signature :**

Table of Contents

Exp. No — 01

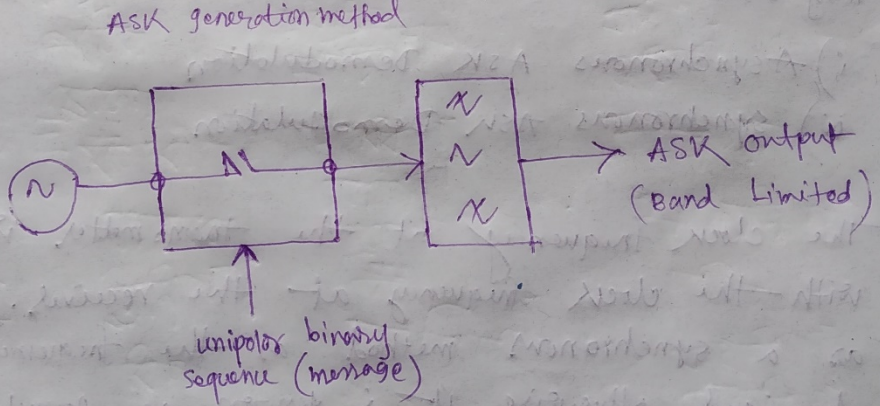
Exp. name:- Write a MATLAB program for binary amplitude shift keying modulation & Demodulation

Theory:

Amplitude shift keying (ASK):

ASK is a type of amplitude modulation which represent the binary data in the form of variations in the amplitude of a signal. Any modulated signal has a high frequency carrier. The binary signal when ASK modulated gives a zero value for low input while it gives the carrier output for high input.

ASK modulation



The ASK modulation block diagram comprises of the carrier signal generation, the binary sequence form the message signal and the band limited filter. The

carrier generator sends a continuous high frequency carrier. The binary sequence from the message signal makes the unipolar input to be either high or low. The high signal closes the switch allowing a carrier wave. Hence the output will be the carrier signal at high input. When there is low input, the switch opens, allowing no voltage to appear. Hence the output will be low. The band limiting filter shapes the pulse depending upon the amplitude & phase characteristics of the band limiting filter or the pulse shaping filter.

ASK Demodulation

There are two types of ASK demodulation techniques. They are

i) Asynchronous ASK Demodulation
ii) Synchronous ASK Demodulation

The clock frequency at the transmitter when matches with the clock frequency at the receiver. It is known as a synchronous method as the frequency get synchronized otherwise, it is known as Asynchronous.

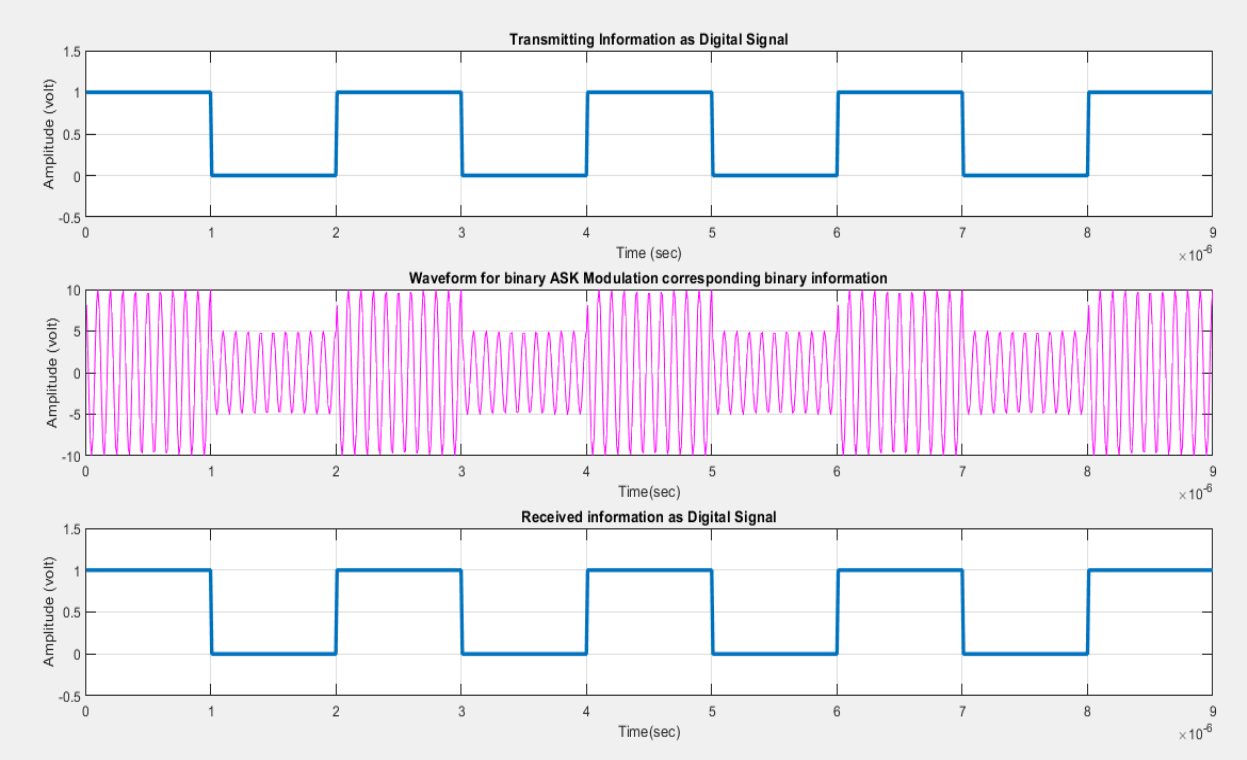
MATLAB Source Code:

```
clc;
clear all;
close all;
x=[1 0 1 0 1 0 1 0 1];
bp=0.000001;
disp('Binary Information at transmitter');
disp(x);
%Representation of Transmitting binary information as digital signal
bit=[];
for n=1:1:length(x)
 if x(n)==1
 se=ones(1,100);
 else
 x(n)=0;
 se=zeros(1,100);
 end
 bit=[bit se];
end
t1=bp/100:bp/100:100*length(x)*(bp/100);
subplot(3,1,1)
plot(t1,bit,'linewidth',2.5);
grid on;
axis([0 bp*length(x) -0.5 1.5]);
ylabel('Amplitude (volt)');
xlabel('Time (sec)');
title('Transmitting Information as Digital Signal');
%Binary ASK Modulation
A1=10;
A2=5;
br=1/bp;
f=br*10;
```

```
t2=bp/99:bp/99:bp;
ss=length(t2);
m=[];
for(i=1:1:length(x))
 if x(i)==1
 y=A1*cos(2*pi*f*t2);
 else
 y=A2*cos(2*pi*f*t2);
 end
 m=[m y];
end
t3=bp/99:bp/99:bp*length(x);
subplot(3,1,2)
plot(t3,m,'m');
grid on;
xlabel('Time(sec)');
ylabel('Amplitude (volt)');
title('Waveform for binary ASK Modulation corresponding binary information');
%Binary ASK Demodulation
mn=[];
for n=ss:ss:length(m)
 t=bp/99:bp/99:bp;
 y=cos(2*pi*f*t);
 mm=y.*m((n-(ss-1)):n);
 t4=bp/99:bp/99:bp;
 z=trapz(t4,mm);
 zz=round((2*z/bp));
 if(zz>7.5)
 a=1;
 else
 a=0;
 end
```

```
 mn=[mn a];
end
disp('Binary Information at Receiver');
disp(mn);
%Represntation of Binary data into Digital signal
bit=[];
for n=1:length(mn)
 if mn(n)==1
 se=ones(1,100);
 else
 se=zeros(1,100);
 end
 bit=[bit se];
end
t4=bp/100:bp/100:100*length(mn)*(bp/100);
subplot(3,1,3)
plot(t4,bit,'linewidth',2.5);
grid on;
axis([0 bp*length(mn) -0.5 1.5]);
xlabel('Time(sec)');
ylabel('Amplitude (volt)');
title('Received information as Digital Signal');
```

Input & Output:



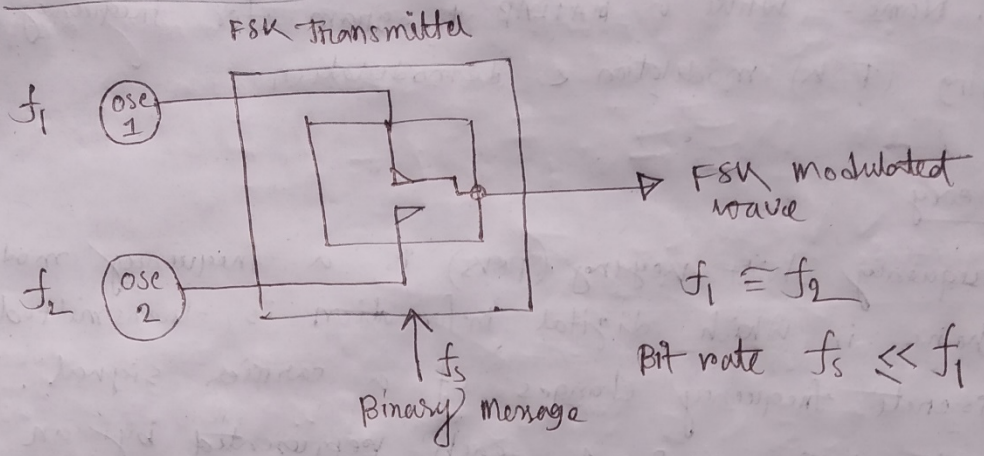
Exp. No. — 02

Exp. Name — Write a MATLAB program for frequency shift Keying (FSK) modulation & demodulation.

Theory

Frequency shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier signal. The binary states logic 0 & 1 each represented by an analog wave form. Logic 0 is represents by a wave at a specific frequency & Logic -1 is represented by a wave at a different frequency. A modem Converts the binary data from a computer to FSK for transmission over telephones line, cables, optical fiber or wireless media. The modem also converts incoming FSK signals to digital Low and high states which the computer can "understand". Today a standard telephone modem operates at thousands of bits per second cable & wireless modems work at more than 10^6 bps & optical fibes modems function at many mbps. But the basic principle of FSK has not changed in more than half a centuary.

FSK modulation



The FSK modulation blocks diagram comprises of two oscillator with a clock & the input binary sequence following is bits blocks diagram. The two oscillator producing higher & lower frequency signals are connected to a switch along with an internal clock. To avoid the abrupt phase discontinuous of the output waveform during the transmission of the message.

FSK demodulation:

There are different methods for demodulating of FSK wave. The main methods of FSK detection are asynchronous detector & synchronous detector. The synchronous detector is a coherent are while asynchronous detector is a non-coherent one.

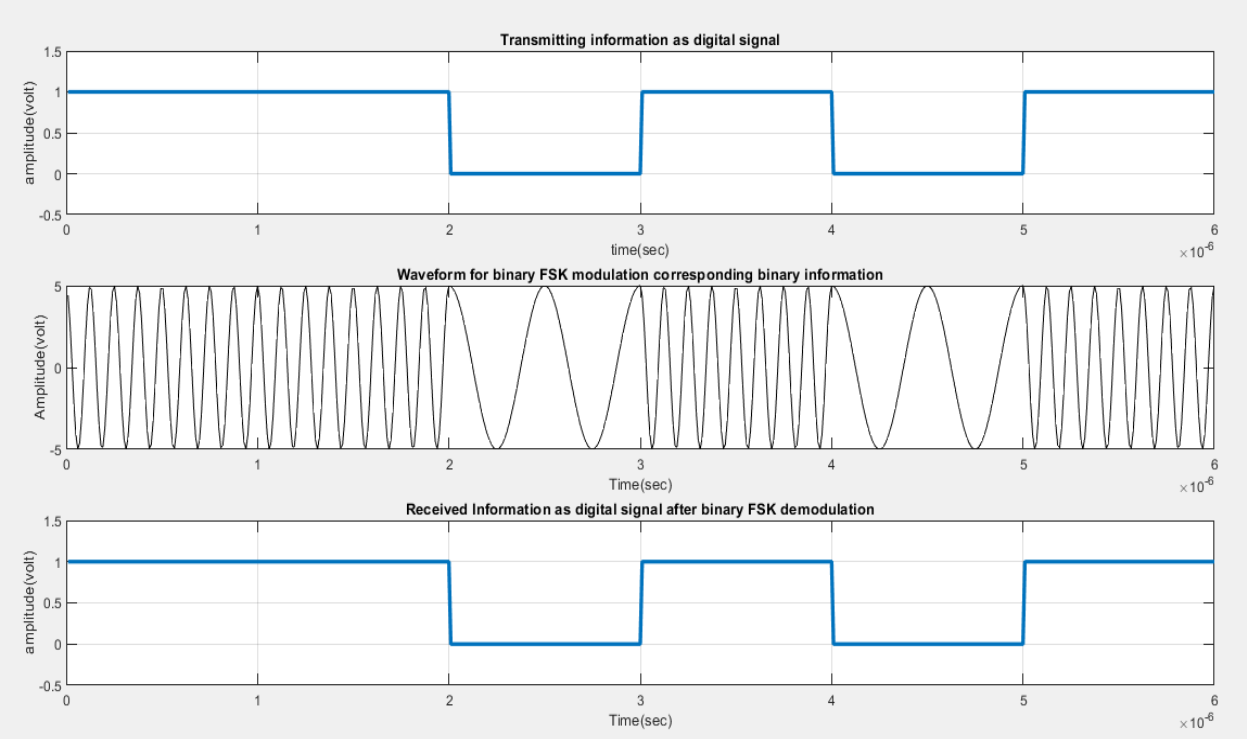
MATLAB Source Code:

```
clc;
clear all;
close all;
x=[1 1 0 1 0 1];
bp= 0.000001;
disp('Binary information at transmitter');
disp(x);
% Representation of Transmitting binary information as digital signal
bit=[];
for n=1:1:length(x)
 if x(n)==1
 se=ones(1,100);
 else x(n)=0;
 se=zeros(1,100);
 end
 bit=[bit se];
end
t1=bp/100:bp/100:100*length(x)*(bp/100);
subplot(3,1,1);
plot(t1,bit,'linewidth',2.5);
grid on;
axis([0 bp*length(x) -0.5 1.5]);
ylabel('amplitude(volt)');
xlabel('time(sec)');
title('Transmitting information as digital signal');
%Binary FSK Modulation
A=5;
br=1/bp;
f1=br*8;
f2=br*2;
t2=bp/99:bp/99:bp;
```

```
ss=length(t2);
m=[];
for(i=1:1:length(x))
 if(x(i)==1)
 y=A*cos(2*pi*f1*t2);
 else
 y=A*cos(2*pi*f2*t2);
 end
 m=[m y];
end
t3=bp/99:bp/99:bp*length(x);
subplot(3,1,2);
plot(t3,m,'k');
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Waveform for binary FSK modulation corresponding binary information');
% Binary FSK Demodulation
mn=[];
for n=ss:ss:length(m)
 t=bp/99:bp/99:bp;
 y1=cos(2*pi*f1*t);
 y2=cos(2*pi*f2*t);
 mm=y1.*m((n-(ss-1)):n);
 mmm=y2.*m((n-(ss-1)):n);
 t4=bp/99:bp/99:bp;
 z1=trapz(t4,mm);
 z2=trapz(t4,mmm);
 zz1=round(2*z1/bp);
 zz2=round(2*z2/bp);
 if(zz1>A/2)%logic level=(0+A)/2 or (A+0)/2 or 2.5 in this case
 a=1;
 else(zz2>A/2);
```

```
 a=0;
 end
 mn=[mn a];
end
disp('Binary information at Receiver');
disp(mn);
%Representation of Binary Information as digital signal
bit=[];
for n=1:length(mn)
 if(mn(n)==1)
 se=ones(1,100);
 else mn(n)=0;
 se=zeros(1,100);
 end
 bit=[bit se];
end
t4=bp/100:bp/100:100*length(mn)*(bp/100);
subplot(3,1,3);
plot(t4,bit,'Linewidth',2.5);
grid on;
axis([0 bp*length(mn) -0.5 1.5]);
ylabel('amplitude(volt)');
xlabel('Time(sec)');
title('Received Information as digital signal after binary FSK
demodulation');
```

Input & Output:



Exp. NO. — 03

Exp. Name — Write a MATLAB program for phase shift keying (PSK) modulation & demodulation.

Theory

PSK is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine & cosine inputs at a particular time. PSK technique is widely used for wireless LAN, bio-metric, Contactless operations. along with RFID & bluetooth Communications. PSK is of two types, depending upon the phases the signal gets shifted. They are ——

i) Binary phase shift keying (BPSK)
ii) Quadrature phase shift keying (QPSK)

BPSK: This is also called as 2-phase PSK or phase reversed keying. In this technique, the sine wave carrier takes two phase reversals such as 0° & 180°. BPSK is a double side band suppressed carrier (DSBSC) modulation scheme for the message being the digital information.

QPSK: This is a PSK, in which the sine wave carrier takes four-phase reversals such as 0°, 90°, 180° & 270°. If this kind of technique is further extended

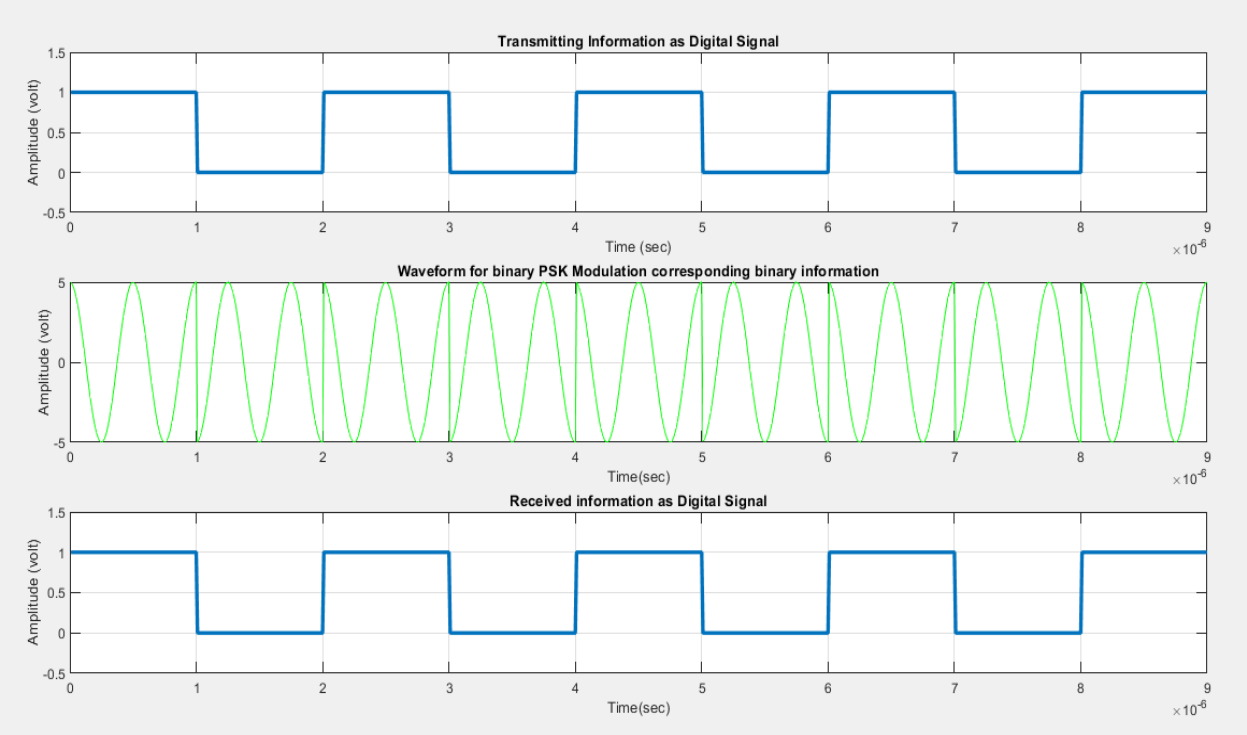
MATLAB Source Code:

```
clc;
clear all;
close all;
x=[1 0 1 0 1 0 1 0 1];
bp=0.000001;
disp('Binary Information at transmitter');
disp(x);
%Representation of Transmitting binary information as digital signal
bit=[];
for n=1:1:length(x)
 if x(n)==1
 se=ones(1,100);
 else
 x(n)=0;
 se=zeros(1,100);
 end
 bit=[bit se];
end
t1=bp/100:bp/100:100*length(x)*(bp/100);
subplot(3,1,1)
plot(t1,bit,'linewidth',2.5);
grid on;
axis([0 bp*length(x) -0.5 1.5]);
ylabel('Amplitude (volt)');
xlabel('Time (sec)');
title('Transmitting Information as Digital Signal');
%Binary PSK Modulation
A=5;
br=1/bp;
f=br*2;
t2=bp/99:bp/99:bp;
```

```
ss=length(t2);
m=[];
for(i=1:1:length(x))
 if x(i)==1
 y=A*cos(2*pi*f*t2);
 else
 y=A*cos(2*pi*f*t2+pi); %Acos(2*pi*f*t+pi) means -Acos(2*pi*f*t)
 end
 m=[m y];
end
t3=bp/99:bp/99:bp*length(x);
subplot(3,1,2);
plot(t3,m,'g');
grid on;
xlabel('Time(sec)');
ylabel('Amplitude (volt)');
title('Waveform for binary PSK Modulation corresponding binary information');
%Binary FSK Demodulation
mn=[];
for n=ss:ss:length(m)
 t=bp/99:bp/99:bp;
 y=cos(2*pi*f*t); %Carrier signal
 mm=y.*m((n-(ss-1)):n);
 t4=bp/99:bp/99:bp;
 z=trapz(t4,mm); %Intregation
 zz=round(2*z/bp);
 if(zz>0)
 a=1;
 else
 a=0;
 end
 mn=[mn a];
```

```
end
disp('Binary Information at Receiver After PSK Demodulation');
disp(mn);
%Represntation of Binary data into Digital signal
bit=[];
for n=1:length(mn)
 if mn(n)==1
 se=ones(1,100);
 else
 se=zeros(1,100);
 end
 bit=[bit se];
end
t4=bp/100:bp/100:100*length(mn)*(bp/100);
subplot(3,1,3)
plot(t4,bit,'linewidth',2.5);
grid on;
axis([0 bp*length(mn) -0.5 1.5]);
xlabel('Time(sec)');
ylabel('Amplitude (volt)');
title('Received information as Digital Signal');
```

Input & Output:



Exp. No. — 04

Exp. Name — Write a MATLAB Code for quadrature phase shift keying (Qpsk) waveform generation

Theory:

Qpsk is a variation of binary phase shift keying. It is also a double side band suppressed carrier modulation scheme. which sends two bits of information at a time.

Instead of the conversion of digital bits into a series of digital streams. It converts them into bit pairs. This decreases the data bit rate to half which allows space for the other users.

Qpsk Modulation : The Qpsk modulation uses a bit-splitter, two multipliers with a local oscillator, a 2bit serial to parallel converter & a summer circuit.

At the modulators input the message signal's even bits (2nd, 4th, 6th bit etc) & odd bits (1st, 3rd, 5th bit etc) are separated by the bits splitter & are multiplied with the some carrier to generate odd BPSK (called as psk_I) & even BPSK (called as psk_Q). The psk_Q signal is anyhow phase shifted by 90° before being modulated.

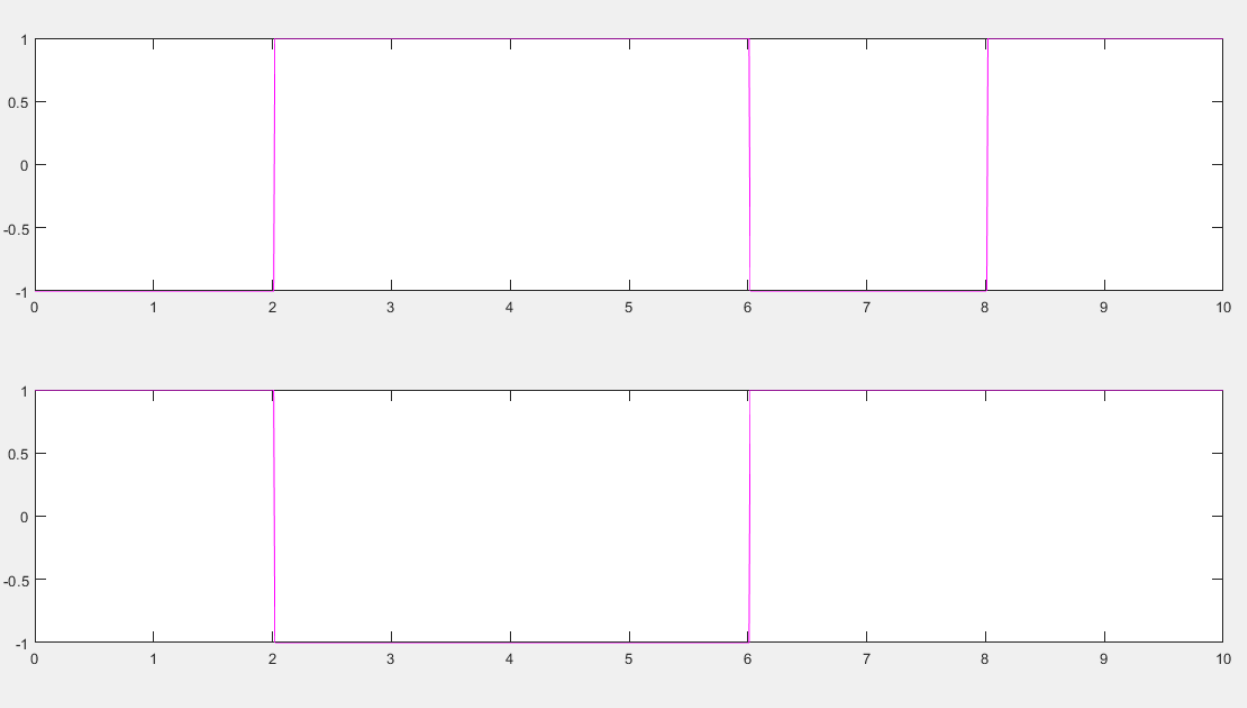
QPSK demodulator: It uses two product demodulators circuits with a local oscillator, two band pass filter two integrators circuits & 2-bit parallel to serial Converter. The two product detectors at the input of the demodulator simultaneously demodulate the two BPSK signals. The pair of bits recovered here from the original data. These signals after procesing. are passed to the parallel serial Converter

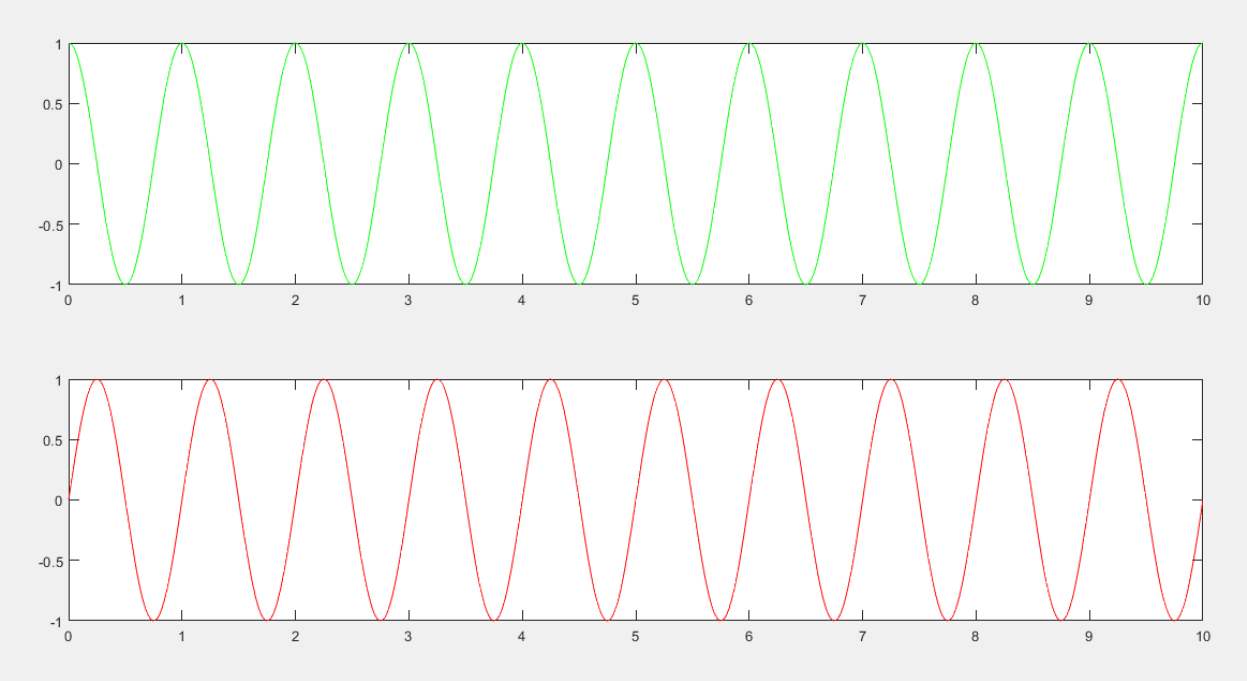
MATLAB Source Code:

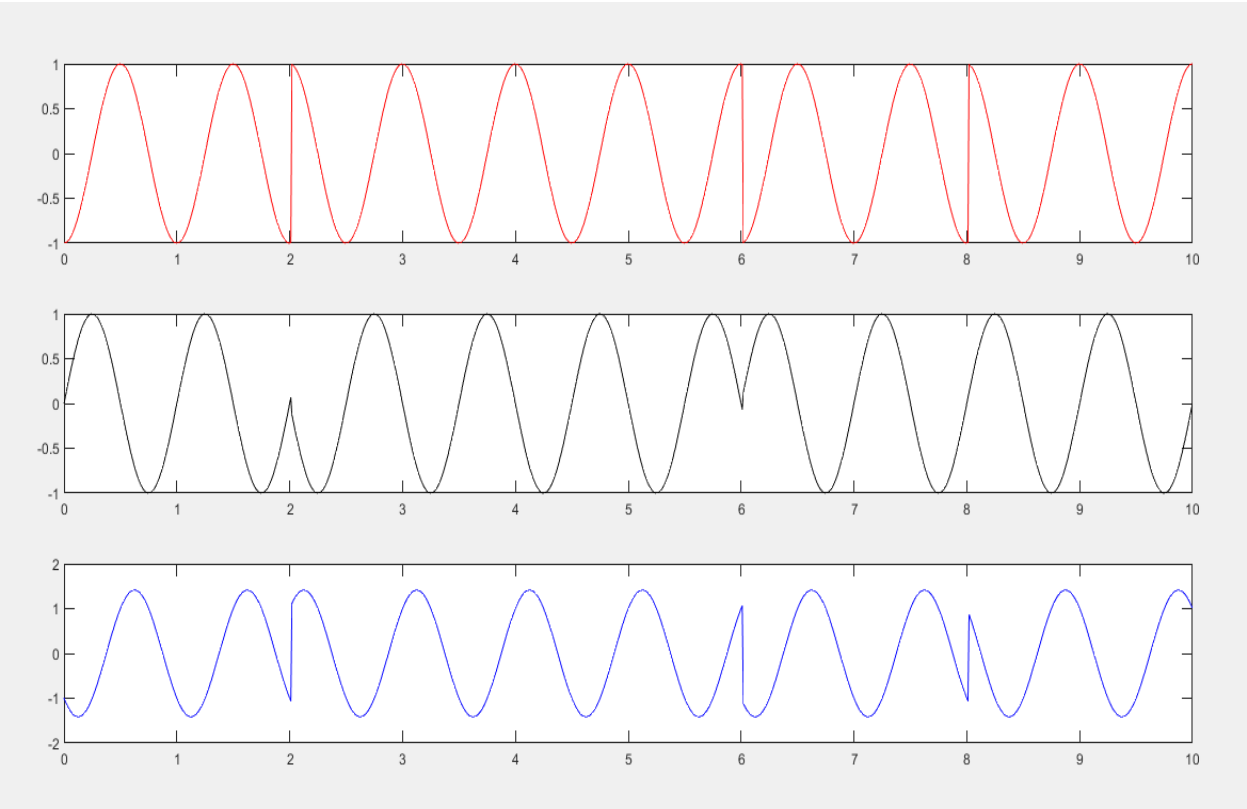
```
%QPSK waveform generation
clc; clear all; close all;
%x=[0 1 0 1]; %input bits
x=randi([0 1],1,10);
%Bits to polar
for i=1:length(x)
 if x(i)==0
 p(i)=-1;
 else
 p(i)=1;
 end
end
%Seperation of even and odd sequences
even_seq=p(1:2:length(x));
odd_seq=p(2:2:length(x));
%NRZ polar line coder signal generation
i=1;
t=0:0.01:length(x);
m=2:2:length(x);
for j=1:length(t)
 if t(j)<=m(i)
 even_ps(j)=even_seq(i);

 else
 even_ps(j)=even_seq(i);
 i=i+1;
 end
end
i=1;
m=2:2:length(x);
for j=1:length(t)
```

```
 if t(j)<=m(i)
 odd_ps(j)=odd_seq(i);
 else
 odd_ps(j)=odd_seq(i);
 i=i+1;
 end
end
figure(1);
subplot(211);
plot(t,even_ps,'m');
subplot(212);
plot(t,odd_ps,'m');
%Carrier signals generation
c1=cos(2*pi*1*t);
c2=sin(2*pi*1*t);
figure(2);
subplot(211);
plot(t,c1,'g');
subplot(212);
plot(t,c2,'r');
%QPSK Wveform generation
r1=even_ps.*c1;
r2=odd_ps.*c2;
qpsk_sig=r1-r2;
figure(3);
subplot(311);
plot(t,r1,'r');
subplot(312);
plot(t,r2,'k');
subplot(313);
plot(t,qpsk_sig,'b');
```

Input & Output:






1
0.5
0
-0.5
-1
0 1 2 3 4 5 6 7 8 9 10
1
0.5
0
-0.5
-1
0 1 2 3 4 5 6 7 8 9 10
2
1
0
-1
-2
0 1 2 3 4 5 6 7 8 9 10

Exp. NO – 05

Exp. Name – Write a MATLAB program for pulse Amplitude Modulation (PAM) waveform generation.

Theory: pulse amplitude modulation is defined as a process of varying the amplitude of the signal pulse in accordance to the modulating signal variations. PAM is the basic form of analog pulse modulation in which the width & position characteristics of the pulse are kept constant while varying the amplitude. There are two sampling techniques used for sampling the modulating signal in PAM. which are flat top sampling & natural sampling.

There are two types of PAM which are based on the criteria of signal priority.

i) Single polarity PAM: it is fixed level of DC bias added to the modulating signal. So the modulating signal output is always positive.

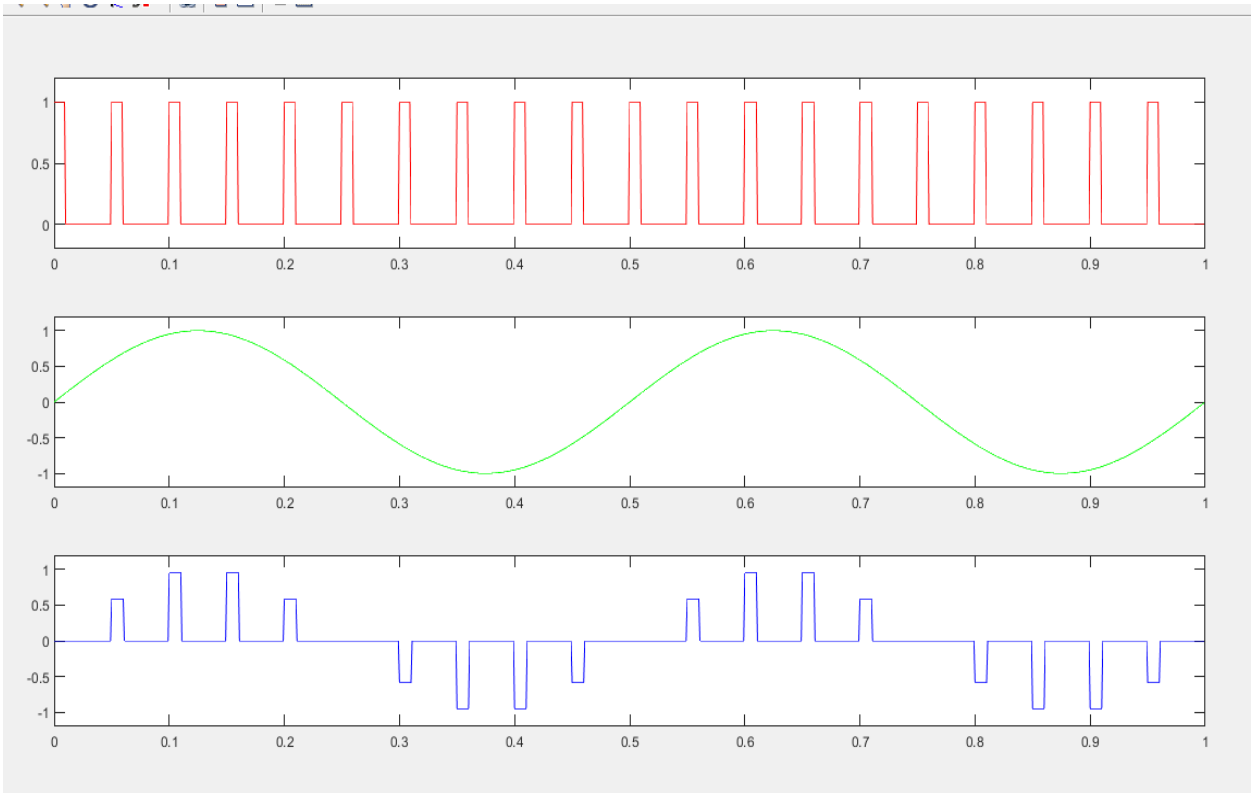
ii) Double polarity PAM: The output of this form of modulating signal comprise of both positive & negative sides.

MATLAB Source Code:

```
%PAM(Pulse Amplitude Modulation) waveform generation
clc;
clear all;
close all;
fc=20;
fm=2;
fs=1000;
t=1;
n=[0:1/fs:t];
n=n(1:end-1);
duty=20;
s= square(2*pi*fc*n,duty);
s(s<0) = 0;
figure(1);
m=sin(2*pi*fm*(n-1));
period_samp=length(n)/fc;
ind=1:period_samp:length(n);
on_samp=ceil(period_samp*duty/100);
on_samp;
pam=zeros(1,length(n));
for i=1:length(ind)
 pam(ind(i):ind(i)+on_samp) = m(ind(i));
end
subplot(3,1,1);
plot(n,s,'r');
ylim([-0.2 1.2]);
subplot(3,1,2);
plot(n,m,'g');
ylim([-1.2 1.2]);
subplot(3,1,3);
plot(n,pam,'b');
```

```
ylim([-1.2 1.2]);
```

Input & Output:



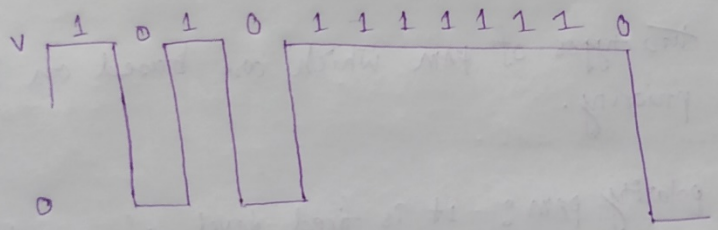
Exp. No — 06

Exp. Name — Write a MATLAB program for unipolar Non-Return to Zero (NRZ) line coding.

Theory

Unipolar NRZ Signaling

In this line code, symbol 1 is represented by transmitting a pulse of amplitude A voltage V for the duration symbol a symbol 0 is represented by switching off the pulse as illustrated in the following figure ——

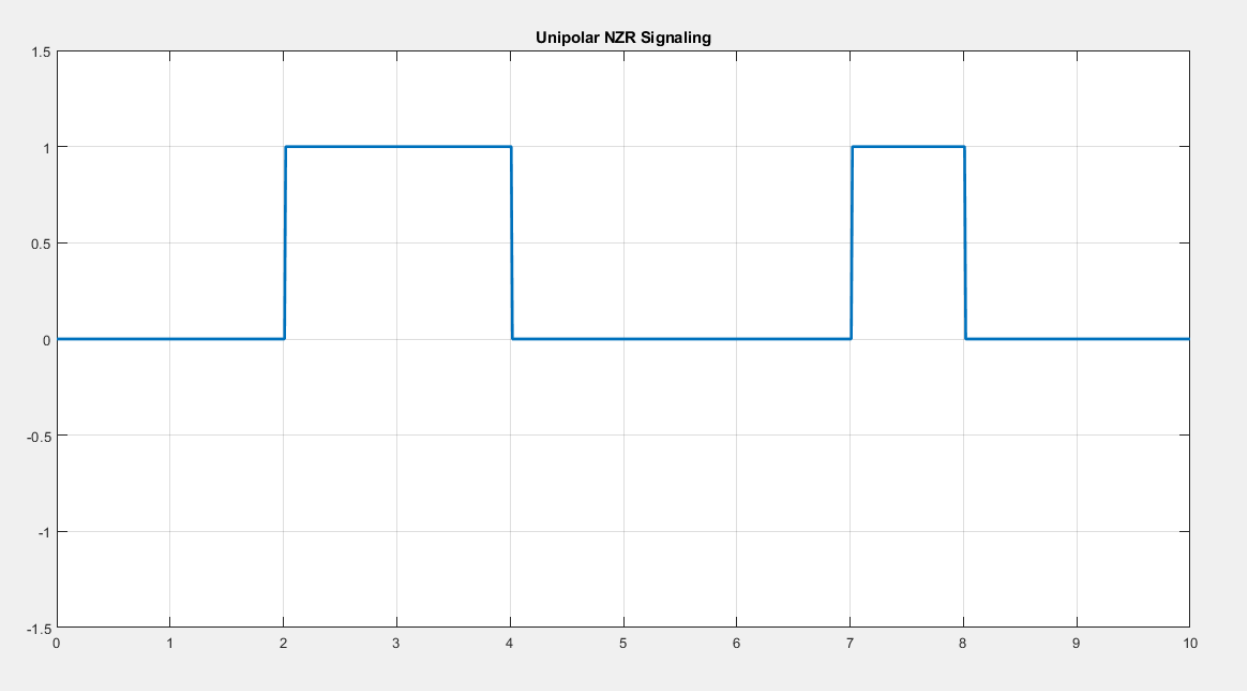


MATLAB Source Code:

```
%Unipolar Non Return to Zero Line Coding
clc;
clear all;
close all;
N=10; %Number of bits
n=randi([0,1],1,N); %Random bit generation
%Mapping Function
for m=1:N
 if n(m)==1
 nn(m)=1;
 else
 nn(m)=0;
 end
end
nn
%Signal Shaping
i=1;
t=0:0.01:length(n); %100 Times duration set up for a single binary bit
for j=1:length(t) %Indexing set-up for time duration
 if t(j)<=i %Binary input data index Check-up Condition
 y(j)=nn(i); %Assign value from the mapping function
 else
 y(j)=nn(i);
 i=i+1; %Binary input data index increment
 end
end
plot(t,y, 'linewidth',2);
axis([0,N,-1.5,1.5]); %Axis set-up
grid on;
title("Unipolar NZR Signaling");
```

Input & Output:

nn = 0 0 1 1 0 0 0 1 0 0



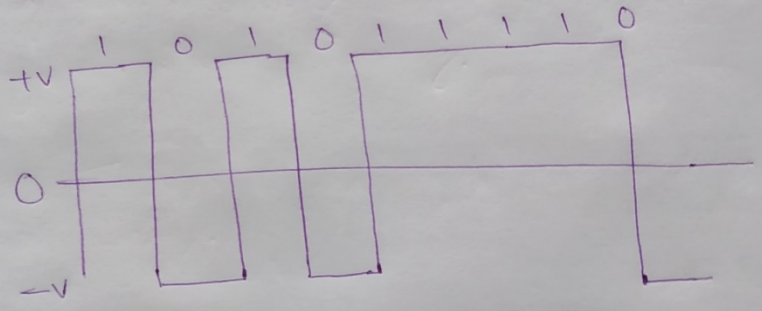
Exp. NO— 07

Exp. Name — Write a MATLAB program for polar NON Return to zero (NRZ) line coding.

Theory

Polar NRZ Signaling

In this line code, symbols 1 & 0 are represented by transmitting pulses of amplitudes +A & −A respectively as illustrated in the following figure

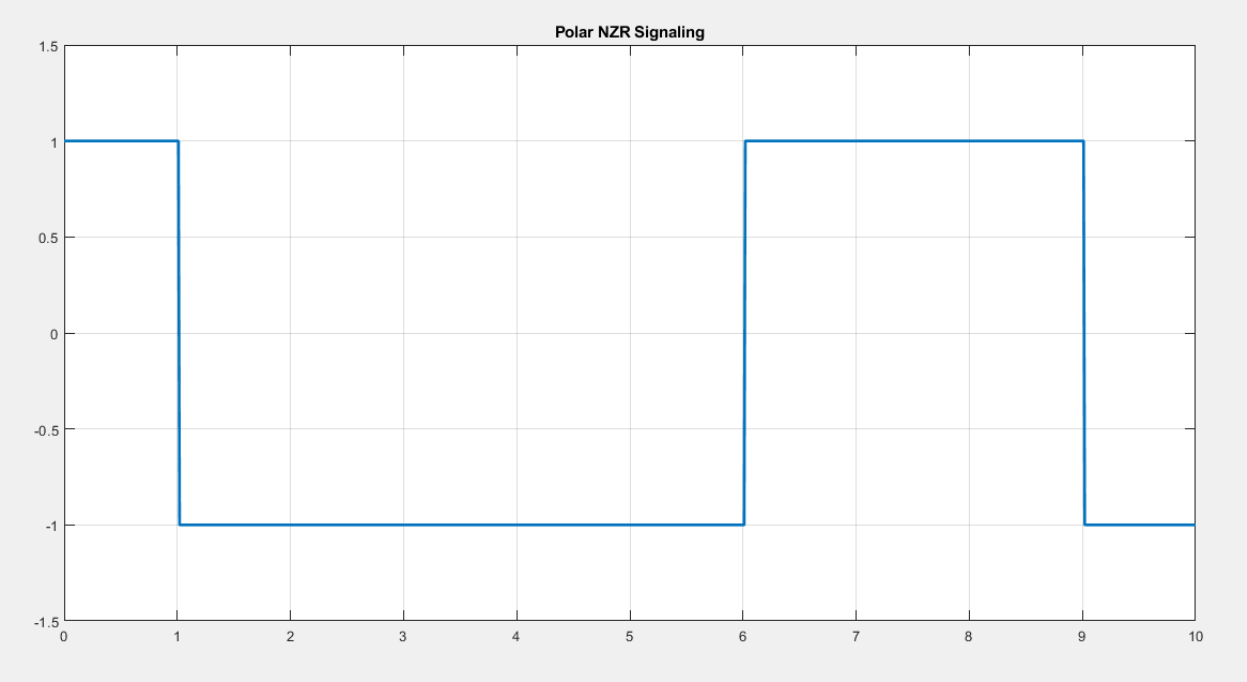


MATLAB Source Code:

```
%Polar Non Return to Zero Line Coding
clc;
clear all;
close all;
N=10; %Number of bits
n=randi([0,1],1,N); %Random bit generation
%Mapping Function
for m=1:N
 if n(m)==1
 nn(m)=1;
 else
 nn(m)=-1;
 end
end
nn
%Signal Shaping
i=1;
t=0:0.01:length(n); %100 Times duration set up for a single binary bit
for j=1:length(t) %Indexing set-up for time duration
 if t(j)<=i %Binary input data index Check-up Condition
 y(j)=nn(i); %Assign value from the mapping function
 else
 y(j)=nn(i);
 i=i+1; %Binary input data index increment
 end
end
plot(t,y, 'linewidth',2);
axis([0,N,-1.5,1.5]); %Axis set-up
grid on;
title("Polar NZR Signaling");
```

Input & Output:

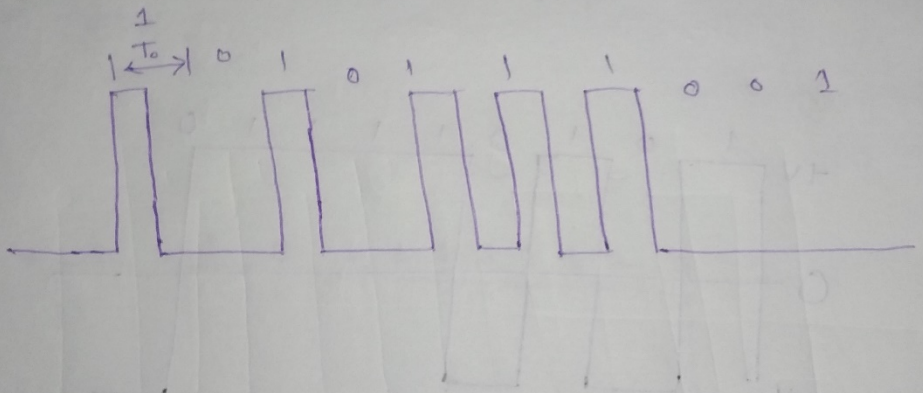
nn = 1 -1 -1 -1 -1 -1 1 1 1 -1



Exp. No — 08

Exp. Name — Write a MATLAB program for unipolar return to zero (RZ) line coding.

Theory: In this line code symbol 1 is represented by a rectangular pulse of amplitude A. voltage V for half symbol width & symbol 0 is represented by transmitting no pulse as illustrated in figure.

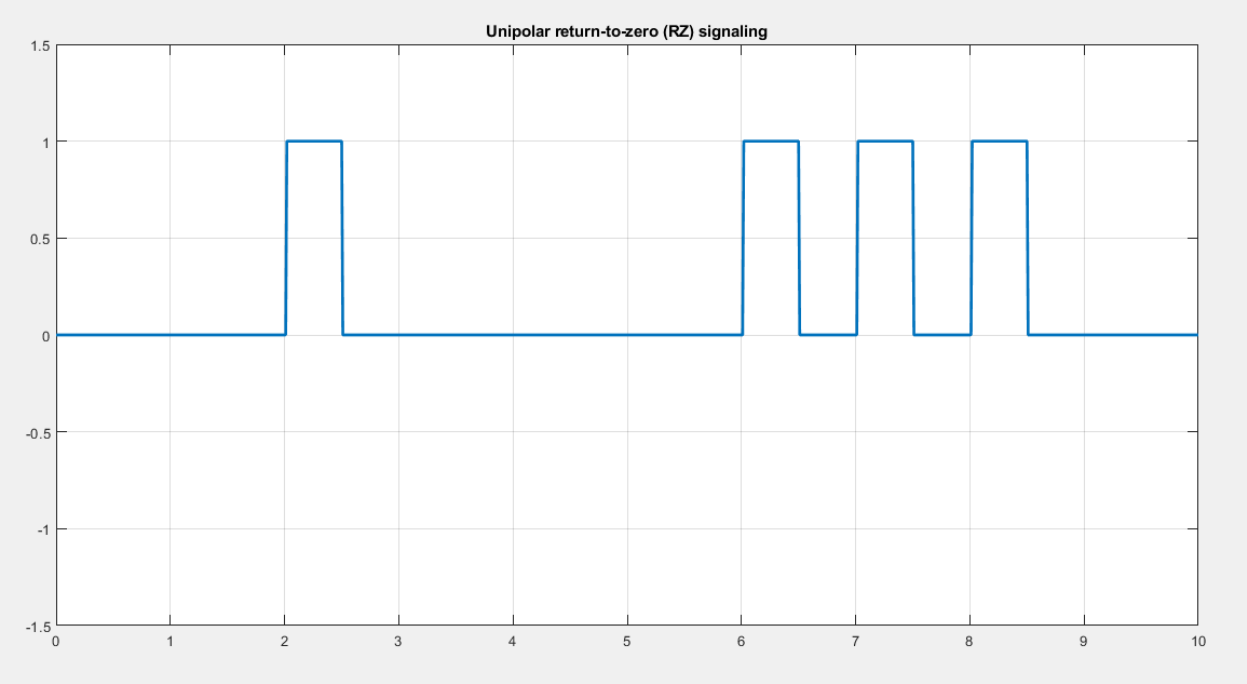


MATLAB Source Code:

```
%RZ Unipolar line coding
clc;
clear all;
close all;
N=10; %Number of bits
n=randi([0,1],1,N); %Random bit generation
%RZ Pulse Shaping
i=1;
a=0; %Initial value for the first half cycle
b=0.5; %Initial value for the second half cycle
t=0:0.01:length(n);
for j=1:length(t)
 if t(j)>=a && t(j)<=b %Condition for the first half cycle
 y(j)=n(i); %Assign first 50 values for
 elseif t(j)>b && t(j)<=i %Condition for the Second half cycle
 y(j)=0; %Set all values 0 for the second half cycle
 else
 i=i+1; %Binary input data index increment
 a=a+1; %Initial value for the first half cycle increment
 b=b+1; %Initial value for the second half cycle increment
 end
 n
end
plot(t,y,'lineWidth', 2); %Linewidth 2 for clear visualization
axis([0,N,-1.5,1.5]); %Axis set-up
grid on;
title('Unipolar return-to-zero (RZ) signaling');
```

Input & Output:

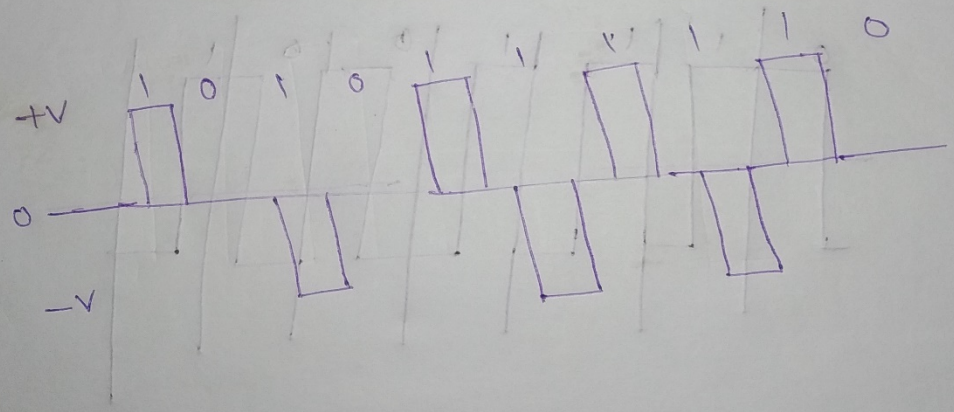
n = 0 0 1 0 0 0 1 1 1 0



Exp. No. — 09

Exp. Name — Write a MATLAB program for Bipolar Return to Zero (RZ) signaling.

Theory : This line code uses three amplitude levels as indicated in the following figure specifically, positive & negative pulses of equal amplitude (+A & −A) are used alternatively for symbol 1 with each pulse having a half symbol width, no pulse is always used for symbol 0.



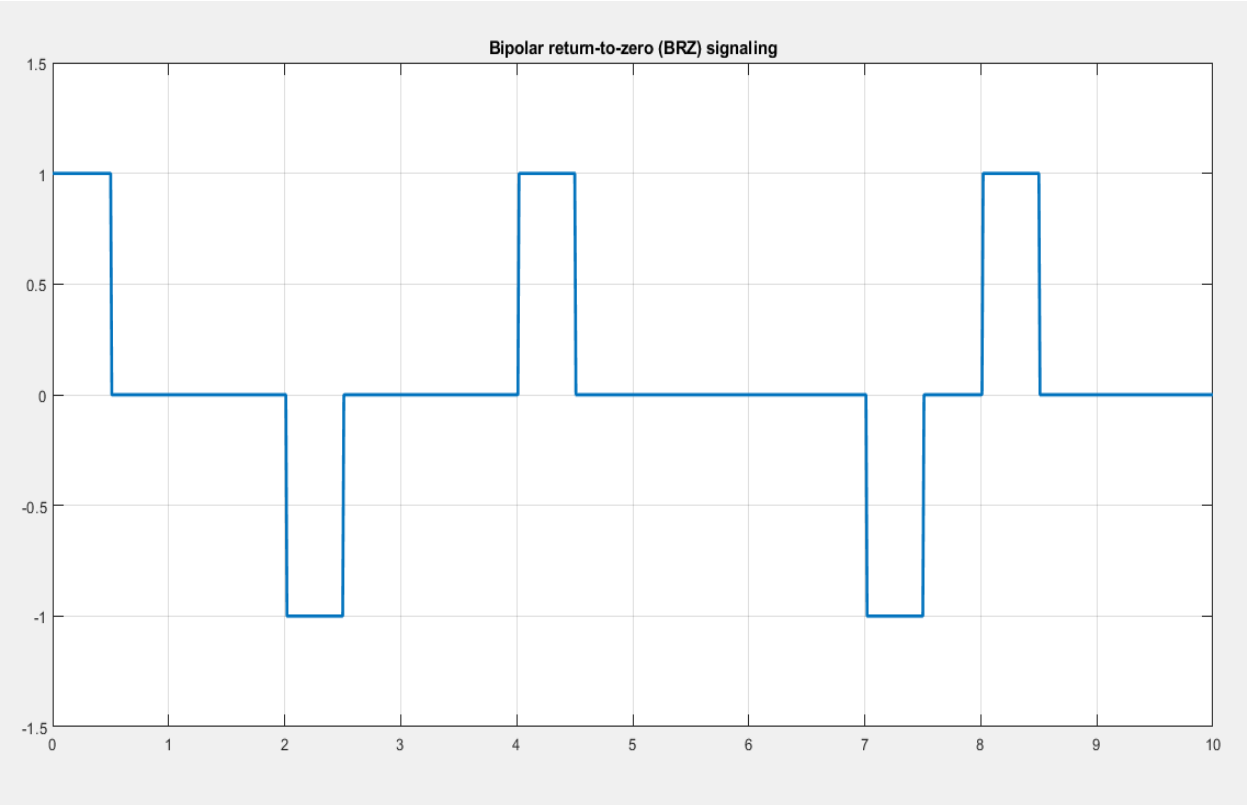
MATLAB Source Code:

```
%Bipolar Return to Zero Signaling
clc;
clear all;
close all;
N=10; %Number of bits
n=randi([0,1],1,N); %Random bit generation
%Binary to Bipolar Conversion
f=1;
for m=1:N
 if n(m)==1
 if f==1
 nn(m)=1;
 f=-1;
 else
 nn(m)=-1;
 f=1;
 end
 else
 nn(m)=0;
 end
end
nn
%Bipolar RZ Pulse Shaping
i=1;
a=0; %Initial value for the first half cycle
b=0.5; %Initial value for the second half cycle
t=0:0.01:length(n);
for j=1:length(t)
 if t(j)>=a && t(j)<=b %Condition for the first half cycle
 y(j)=nn(i); %Assign first 50 values for
 elseif t(j)>b && t(j)<=i %Condition for the Second half cycle
```

```
y(j)=0; %Set all values 0 for the second half cycle
else
i=i+1; %Binary input data index increment
a=a+1; %Initial value for the first half cycle increment
b=b+1; %Initial value for the second half cycle increment
end
end
plot(t,y,'lineWidth', 2); %Linewidth 2 for clear visualization
axis([0,N,-1.5,1.5]); %Axis set-up
grid on;
title('Bipolar return-to-zero (BRZ) signaling');
```

Input & Output:

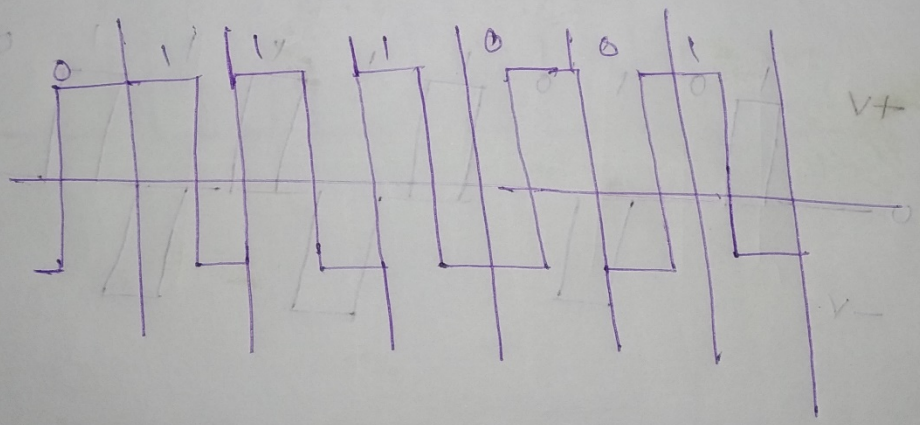
nn = 1 0 -1 0 1 0 0 -1 1 0



Exp. No — 10

Exp. Name — Write a MATLAB program for split phase (manchester code)

Theory: In this method of signaling illustrated in the figure. symbol 1 is represented by a positive pulse of amplitude followed by a negative pulse of amplitude with both pulses being half symbol wide. For a symbol the polarities of these two pulses are reserved.



MATLAB Source Code:

```
%Split Phase-Manchester Coding
clc;
clear all;
close all;
N=10; %Number of bits
n=randi([0,1],1,N); %Random bit generation
%Binary to Manchester Conversion
nnn=[];
for m=1:N
 if n(m)==1
 nn=[1 -1];
 else
 nn=[-1 1];
 end
 nnn=[nnn nn];
end
nnn
%Manchester Coding Pulse Shaping
i=1;
l=0.5;
t=0:0.01:length(n);
for j=1:length(t)
 if t(j)<=l %Condition for the first half cycle
 y(j)=nnn(i); %Assign first 50 values for
 else
 y(j)=nnn(i);
 i=i+1;
 l=l+0.5;
 end
end
plot(t,y,'lineWidth', 2); %Linewidth 2 for clear visualization
```

```
axis([0,N,-1.5,1.5]); %Axis set-up
grid on;
title('Manchester Coding');
```

Input & Output:

```
nnn = 1  -1   1  -1  -1   1  -1   1  -1   1  -1   1   1  -1   1  -1   1  -1   1  -1
```