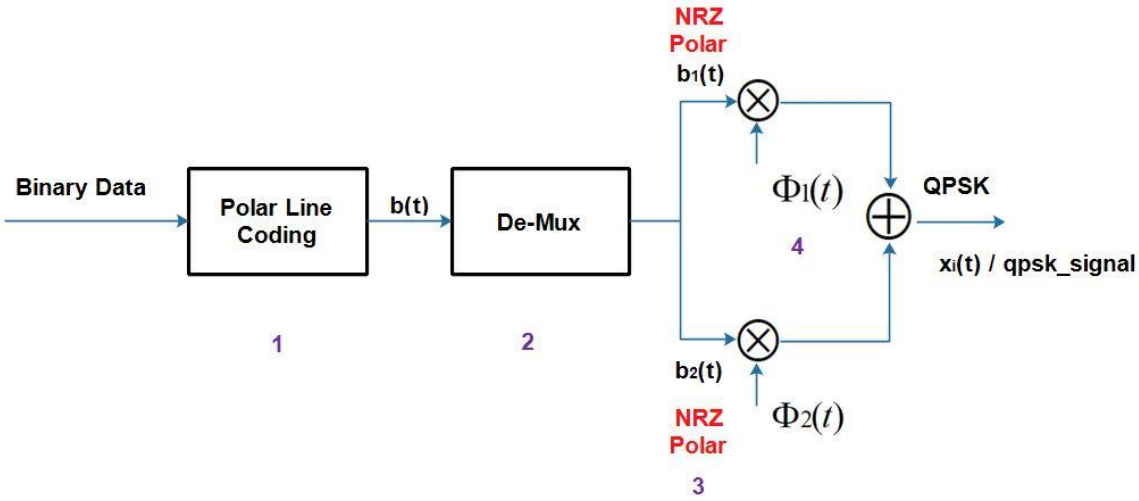
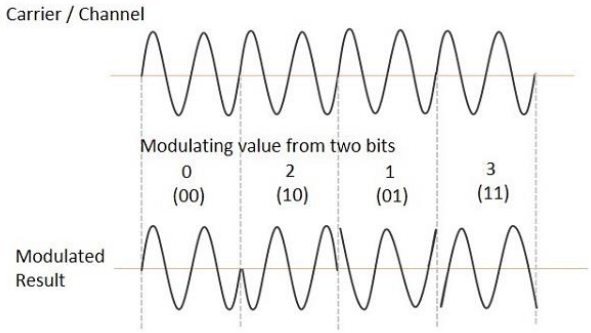
1. QPSK Signal Generation:

The QPSK Modulator uses a bit-splitter, two multipliers with local oscillator, a 2-bit serial to parallel converter, and a summer circuit. Following is the block diagram for the same.



The QPSK waveform for two-bits input is as follows, which shows the modulated result for different instances of binary inputs.



2. Matlab 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,'r');
subplot(212);
plot(t,odd_ps,'r');

%Carrier signals generation
c1=cos(2*pi*1*t);
c2=sin(2*pi*1*t);

figure(2);
subplot(211);
plot(t,c1,'r');
subplot(212);
plot(t,c2,'b');

%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,'b');
subplot(313);
plot(t,qpsk_sig,'b');
```

3. Output:

Binary Data: n =

1 0 1 1 0 1 1 1 1 1

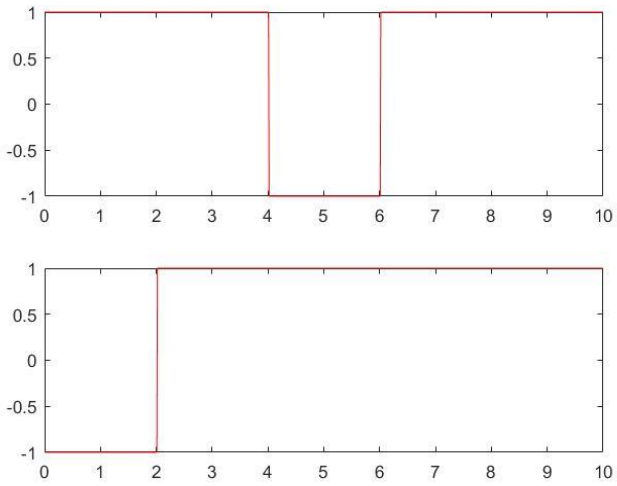


Figure_1: NRZ Polar Line Coded Signal

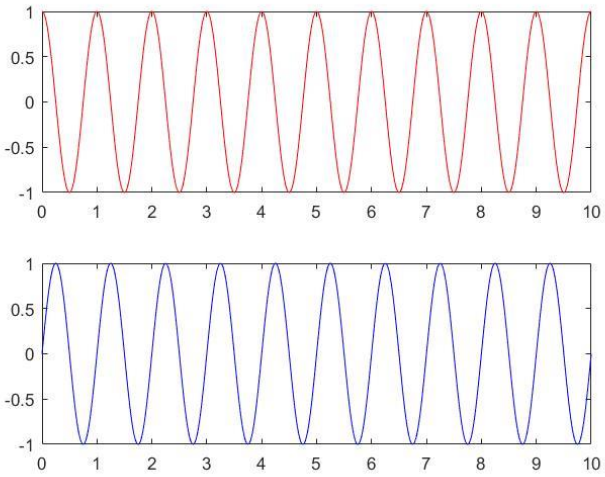


Figure_2: Cosine and Sine signal.

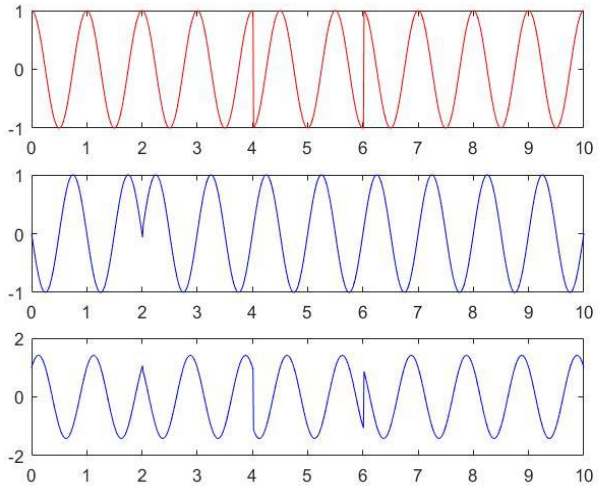


Figure_3: Even signal, Odd signal & QPSK signal

4. **Draft:**