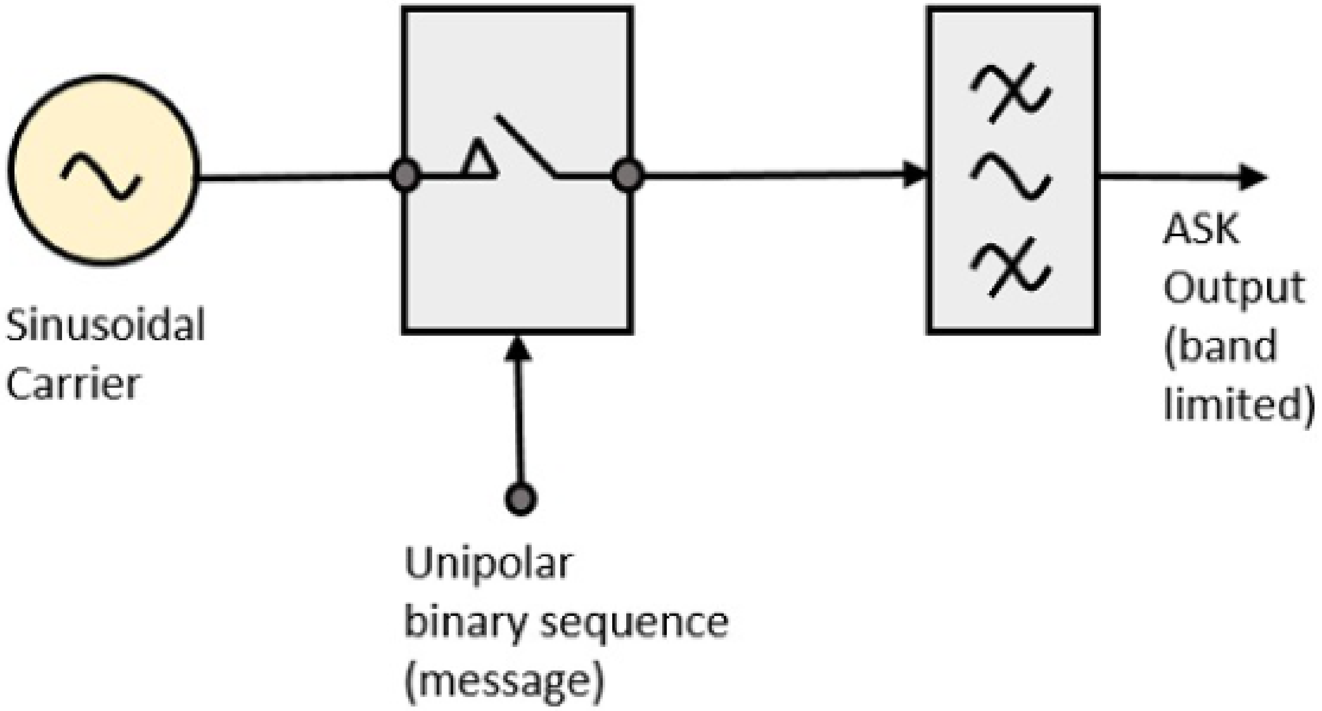
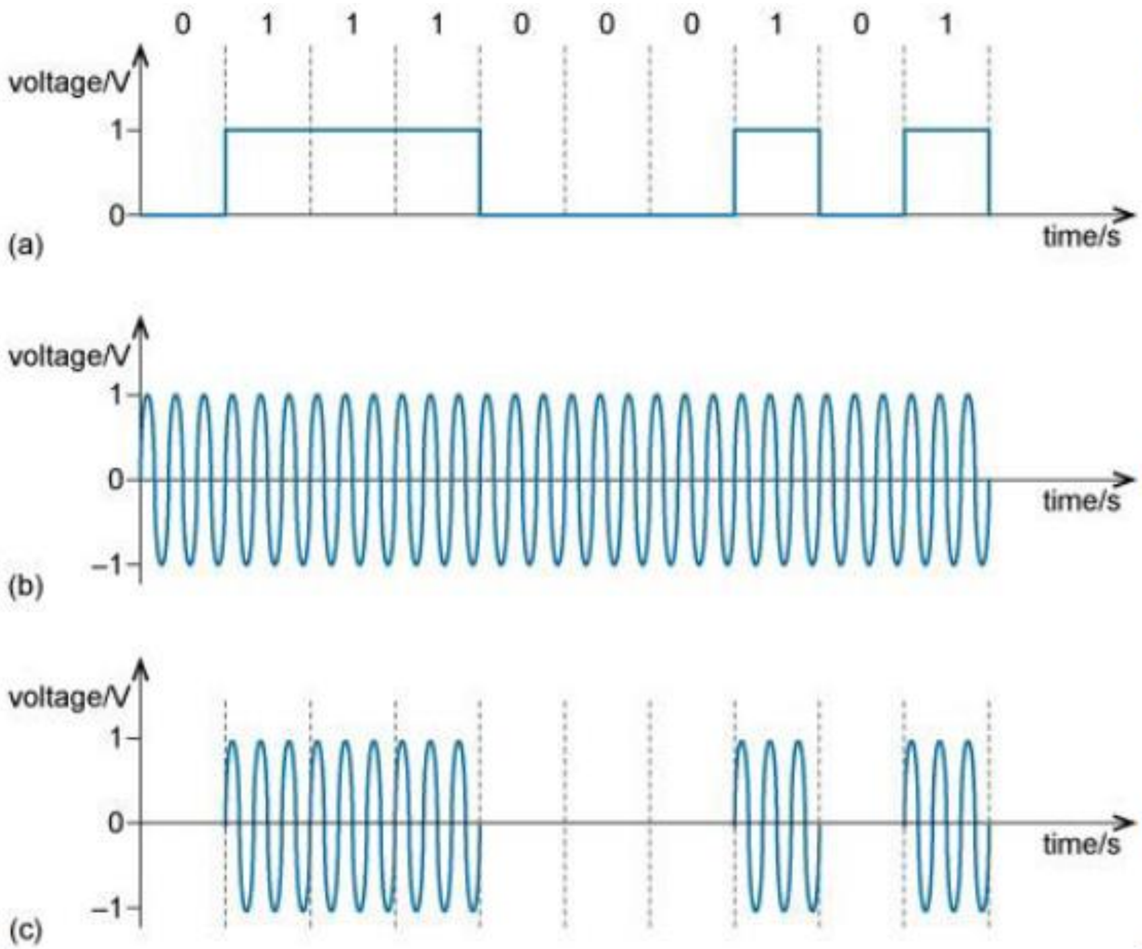
Exp 4
ASK Modulation & Demodulation
(Hardware)

ASK Modulator



Waveform



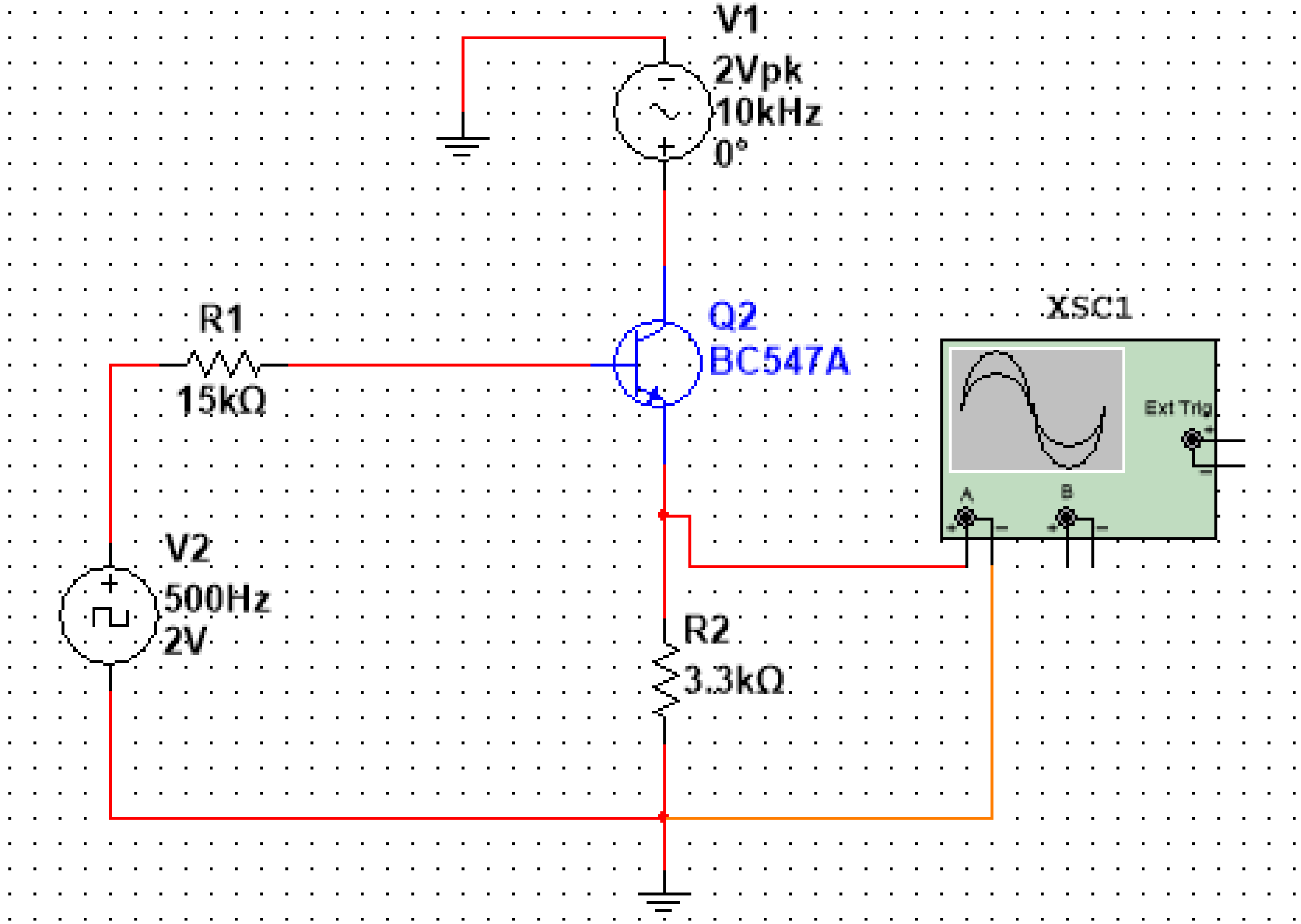
ASK:

(a) data;

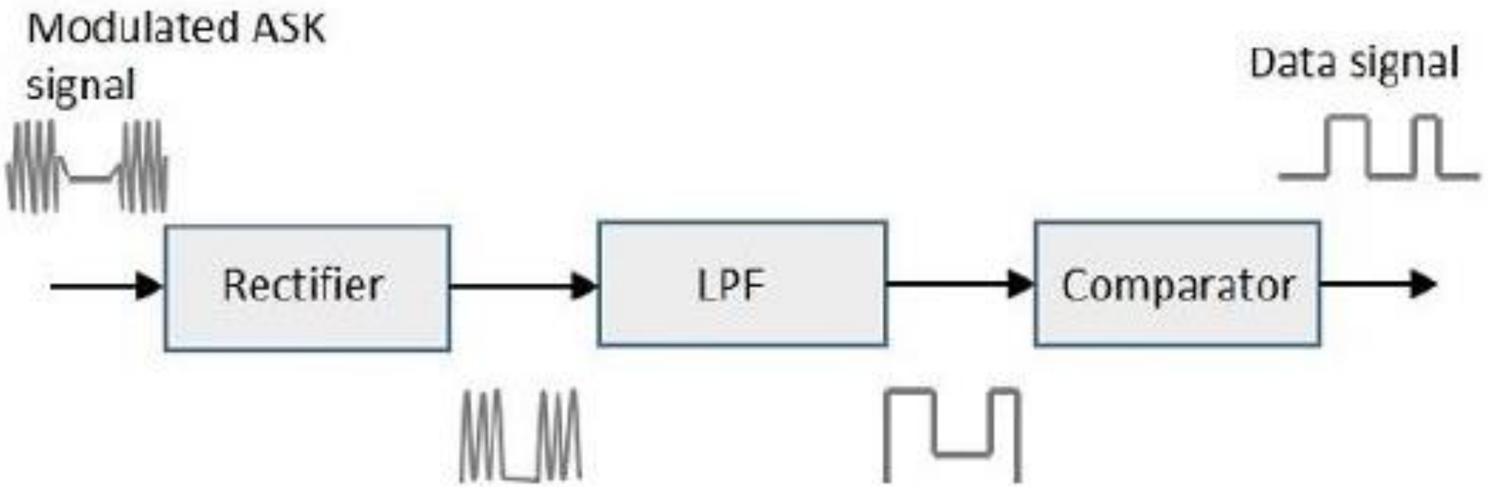
(b) unmodulated carrier;

(c) On–Off keying (OOK);

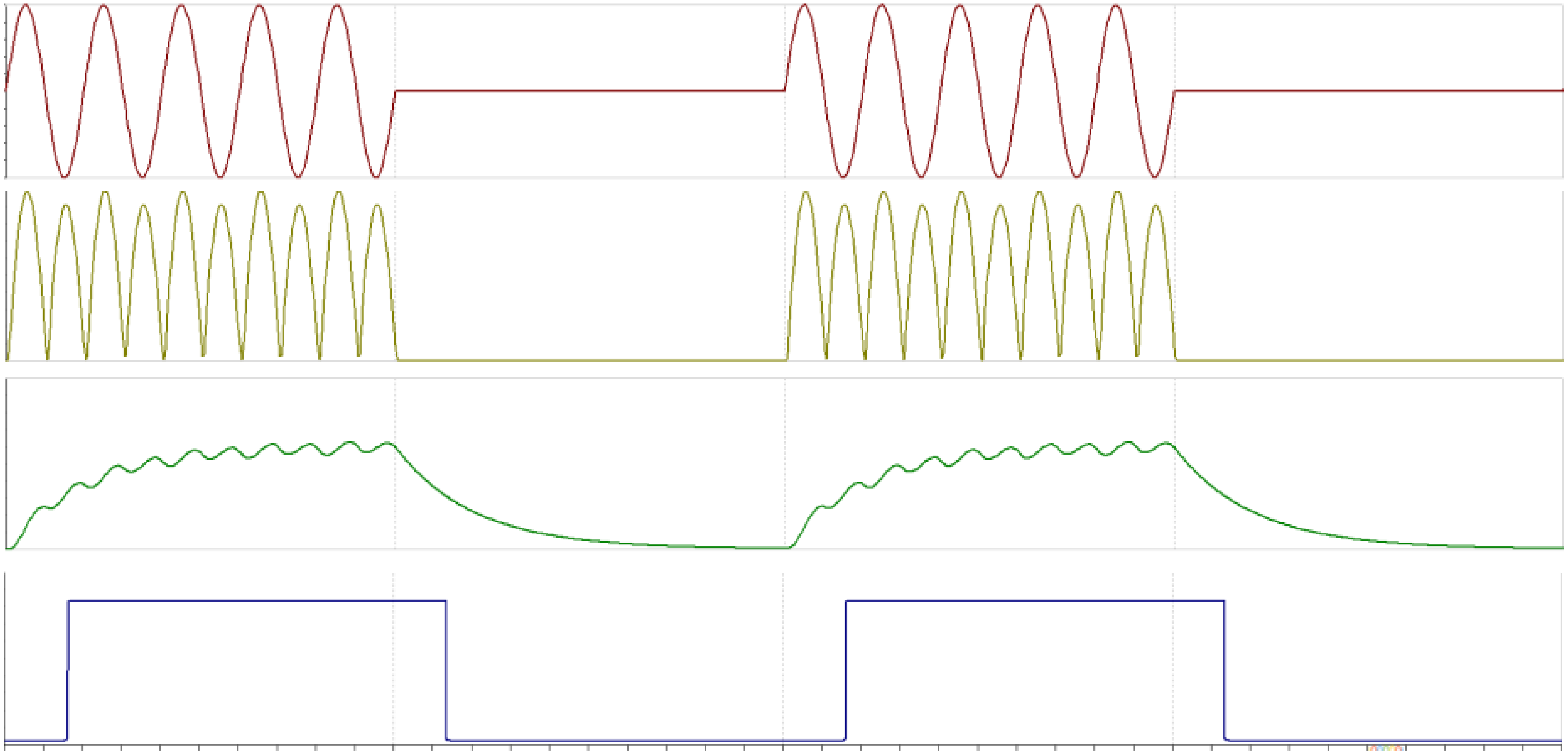
Circuit diagram



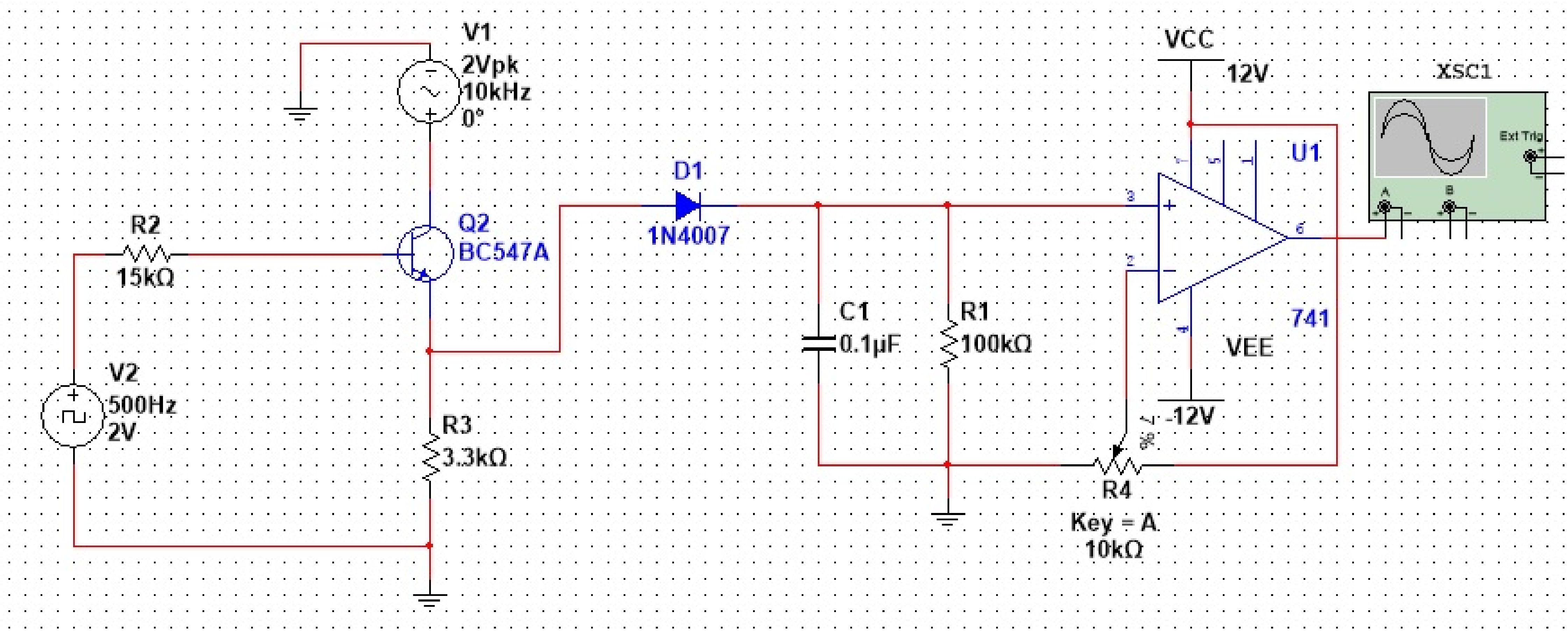
ASK Demodulator



Waveform



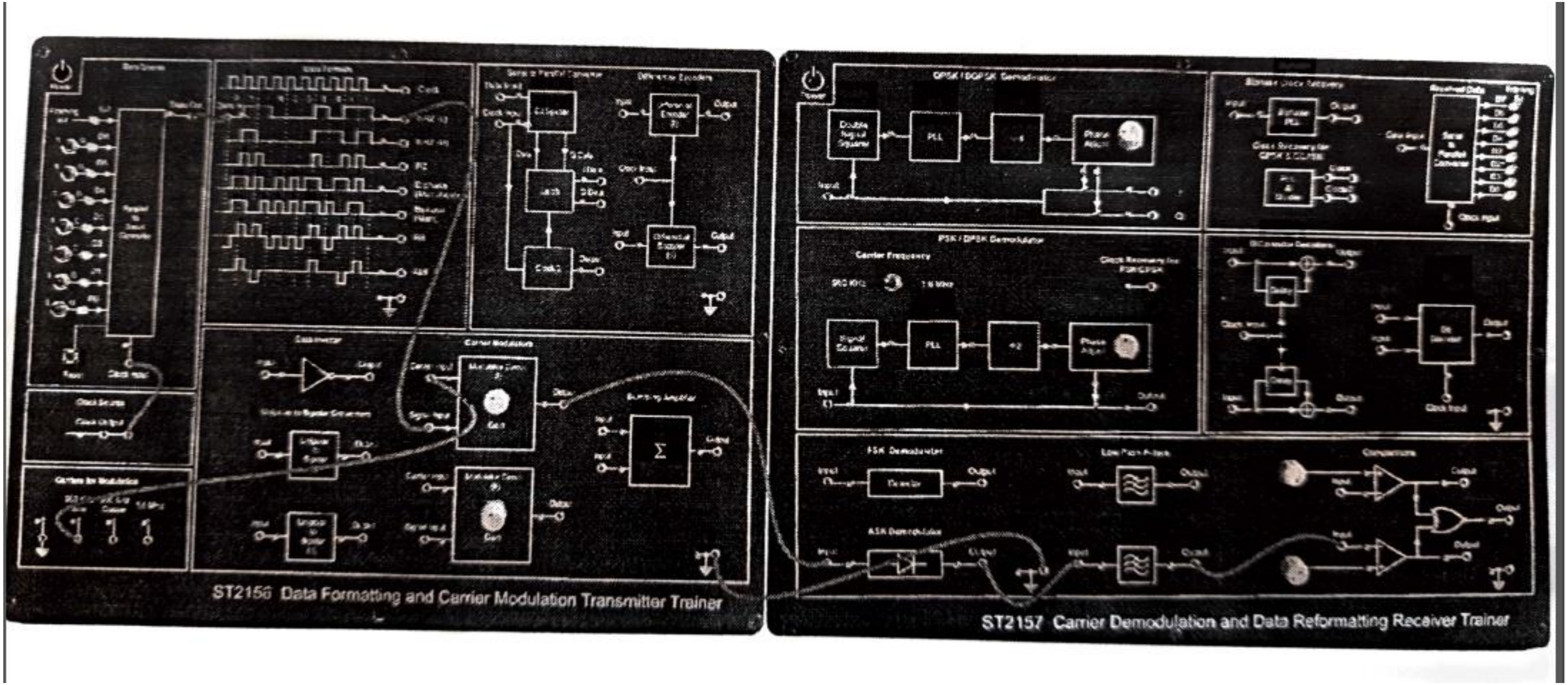
Circuit diagram



Tabulation

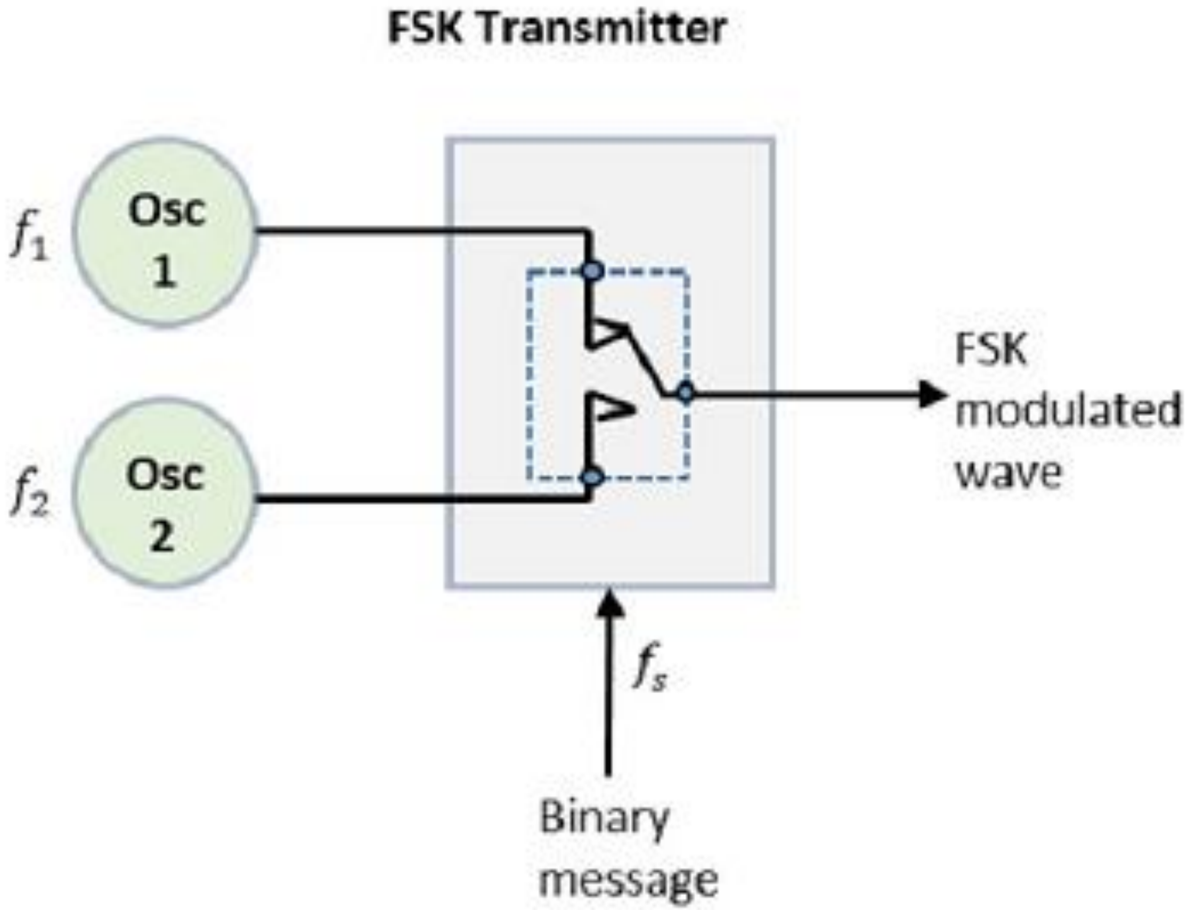
Signals	Amplitude (v)	Time period (s)
Input signal (Digital sequence)		
Carrier signal		
ASK Modulated signal		
Demodulated signal		

Kit connection diagram

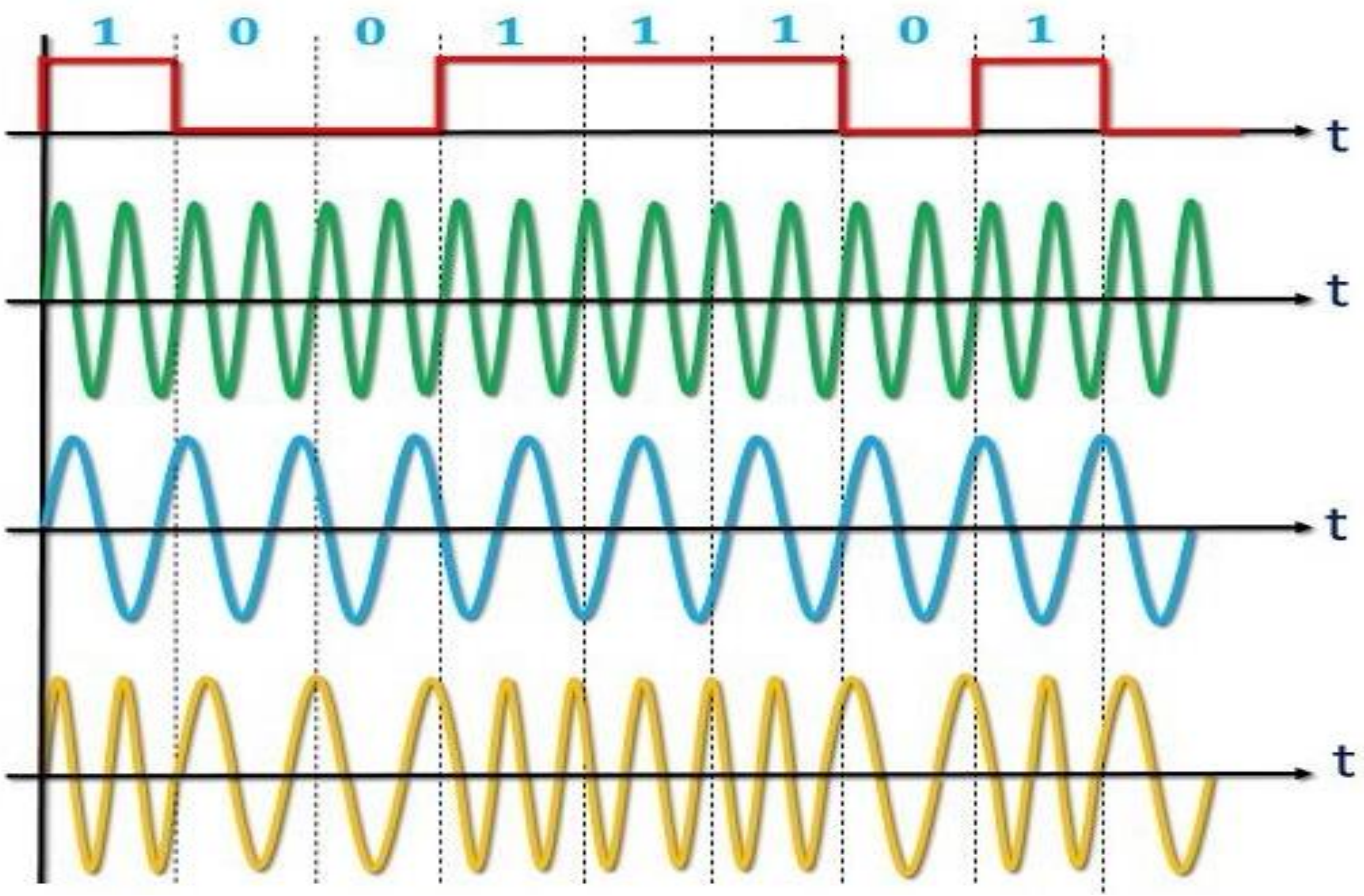


Exp 6
FSK Modulation & Demodulation
(Hardware)

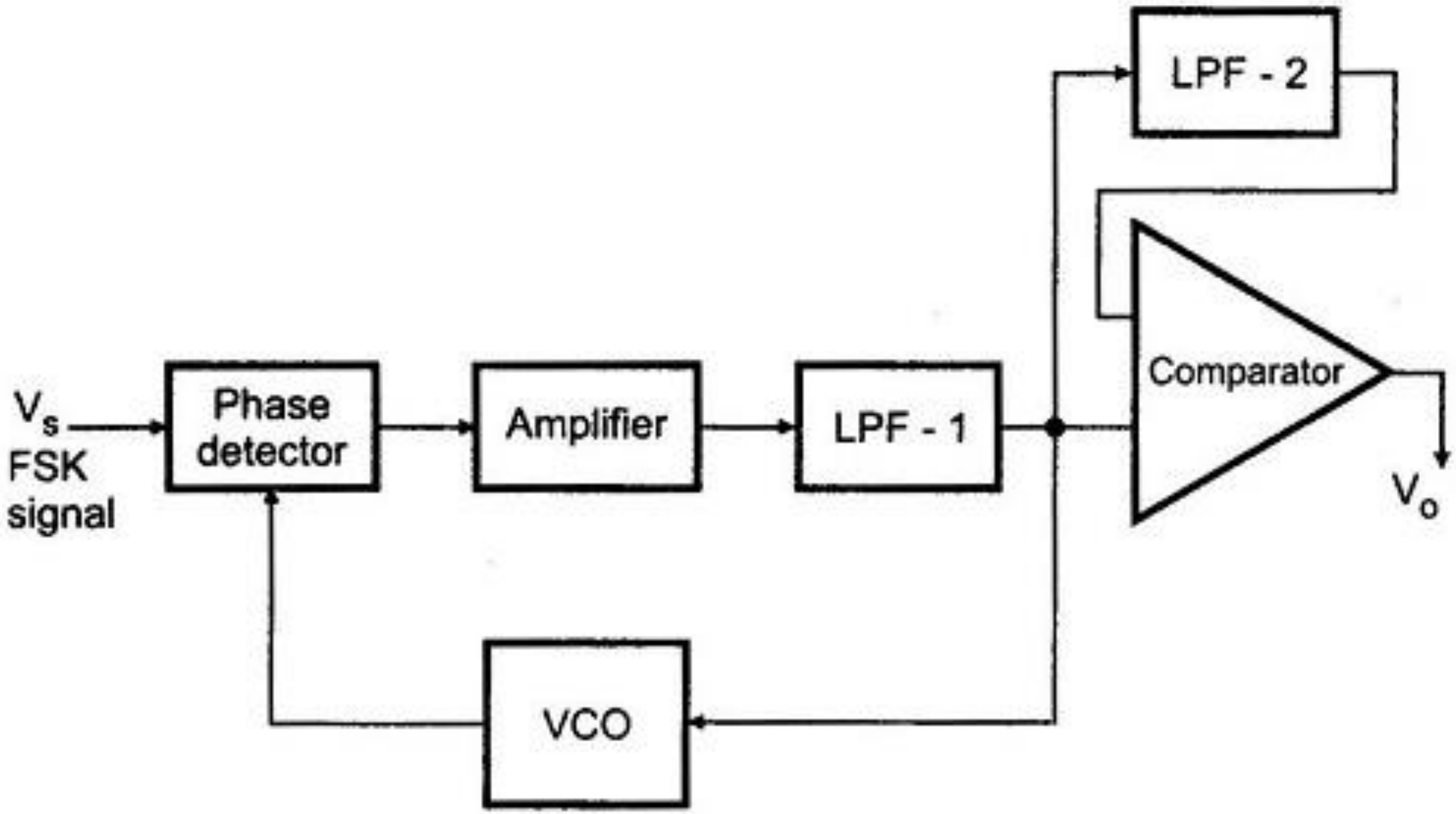
FSK Modulator



Waveform



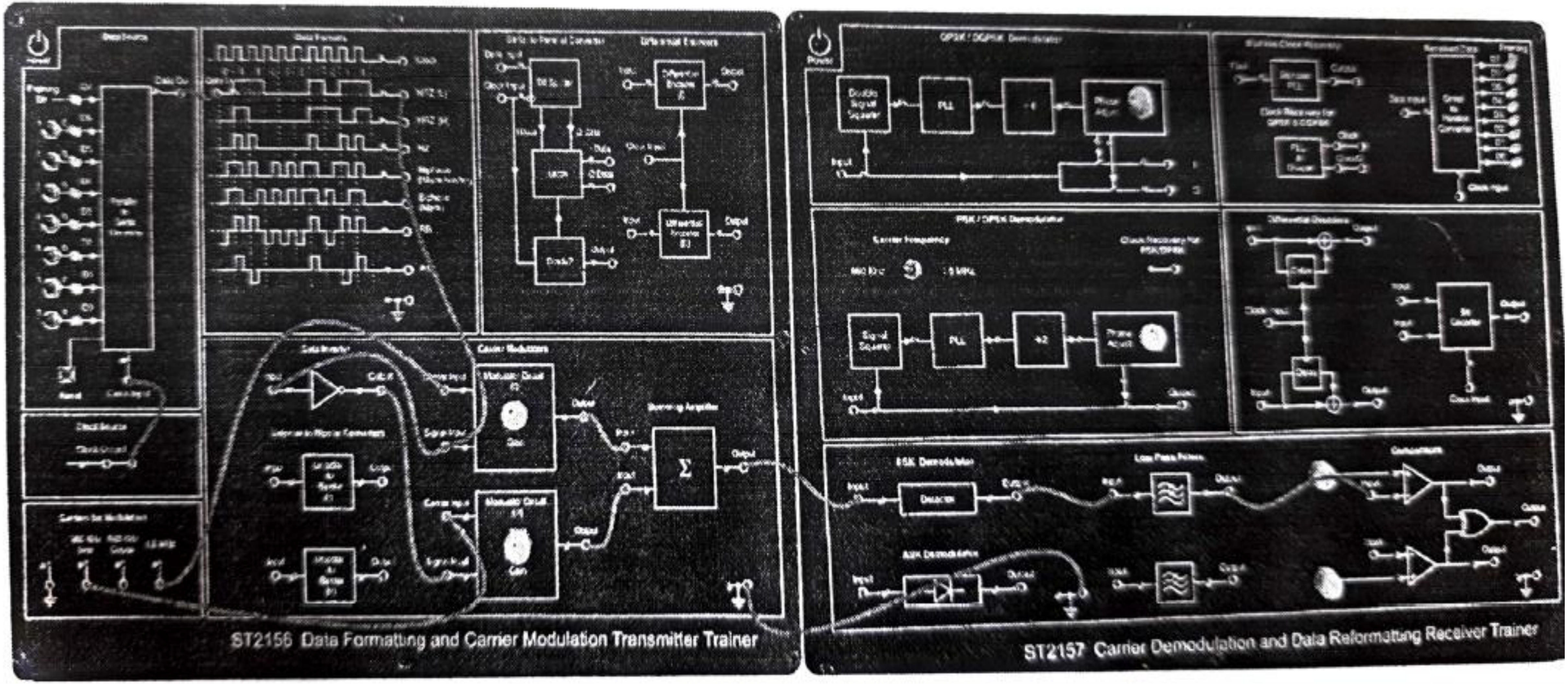
FSK Demodulator



Tabulation

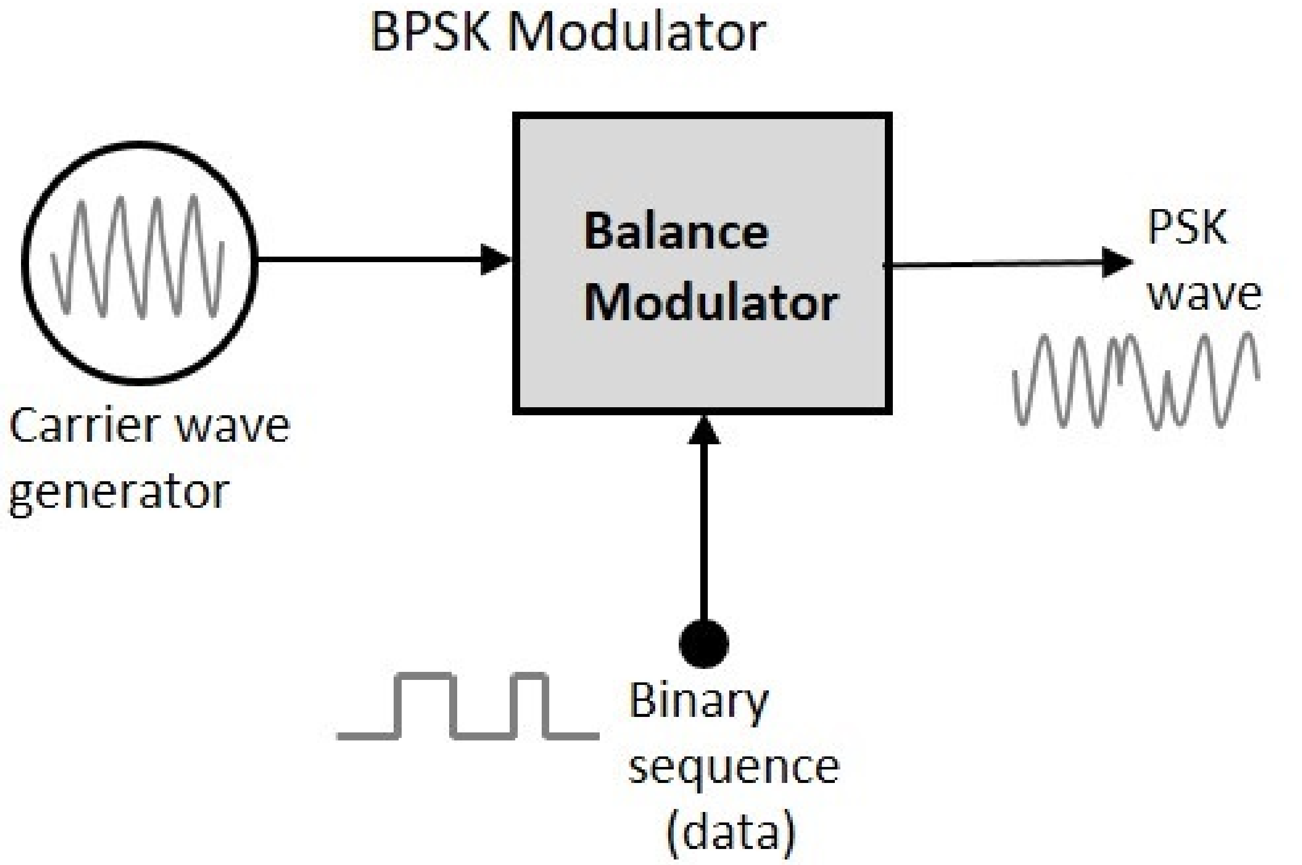
Signals	Amplitude (v)	Time period (s)
Input bit stream		Note down the bit pattern and time period for one bit.
Carrier signal - 1 Carrier signal - 2		
FSK Modulated signal		For 1 – For 0 -
Demodulated signal		Note down the bit pattern and time period for one bit.

Kit connection diagram

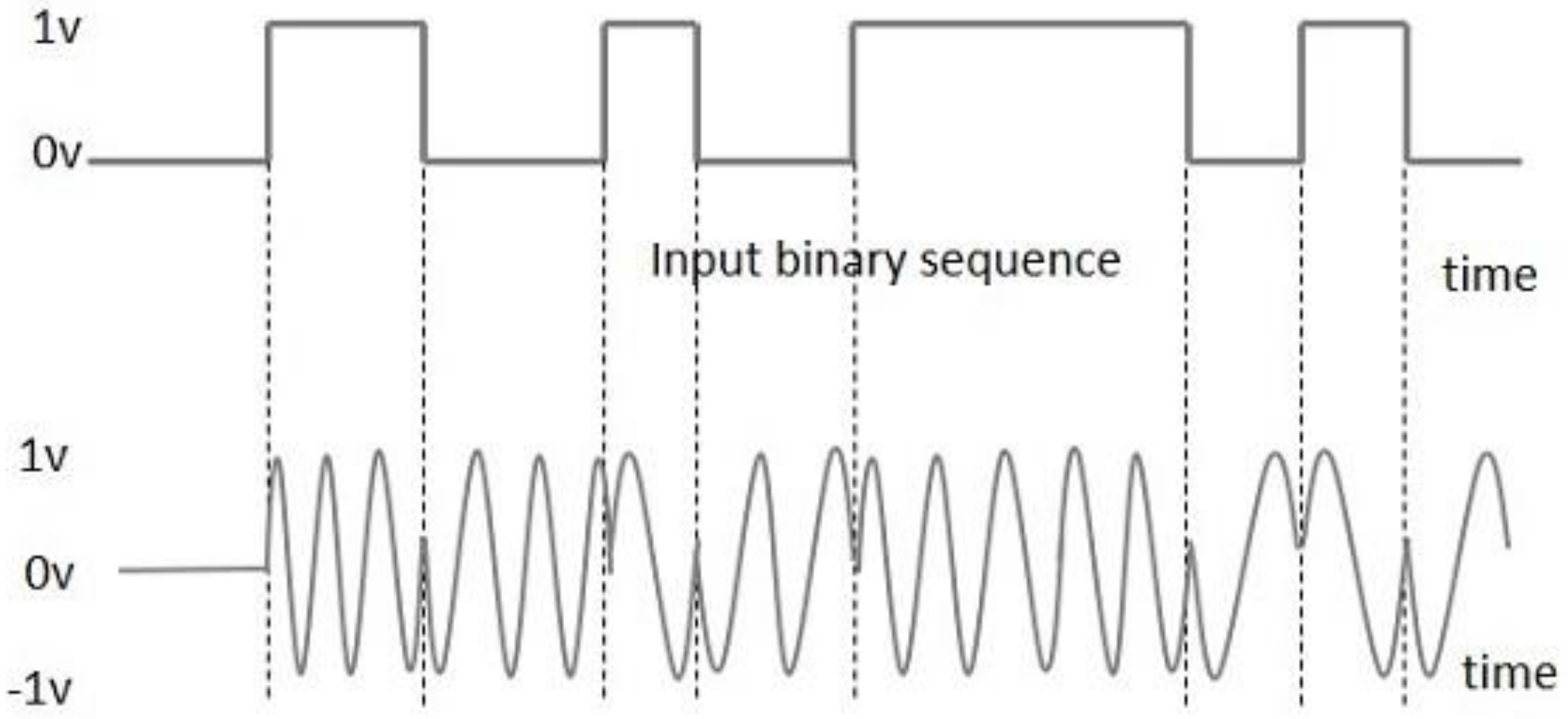


Exp 8
BPSK Modulation & Demodulation
(Hardware)

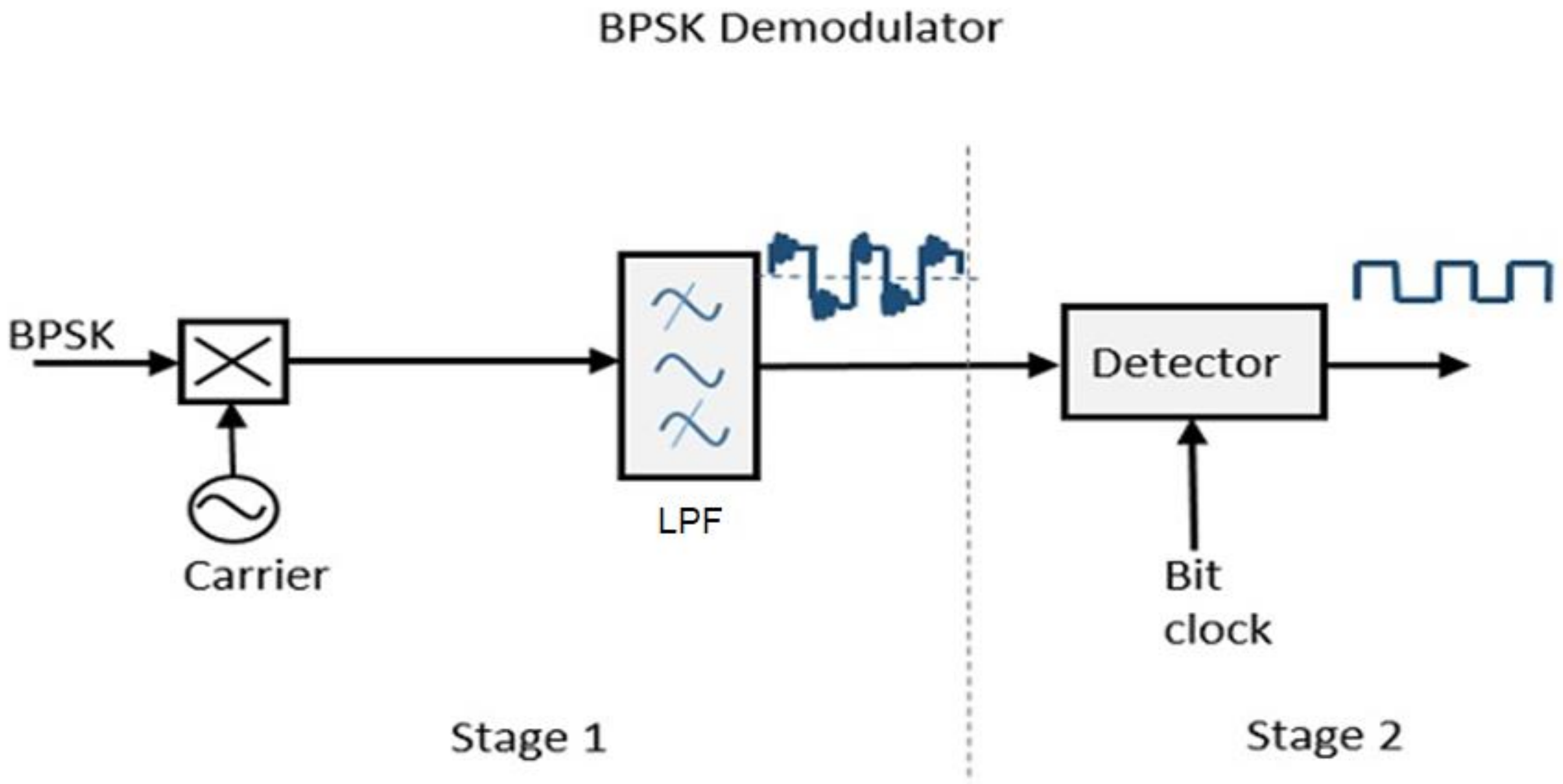
BPSK Modulator



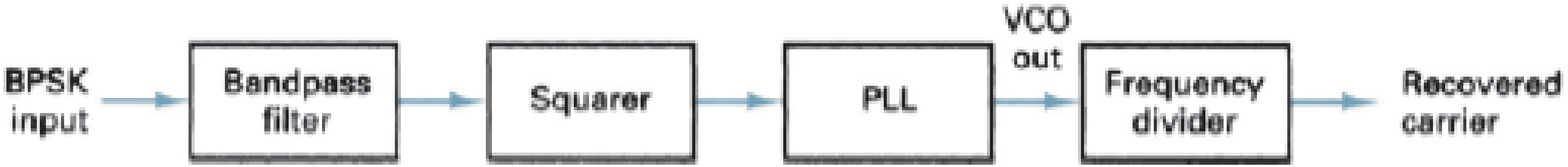
Waveform



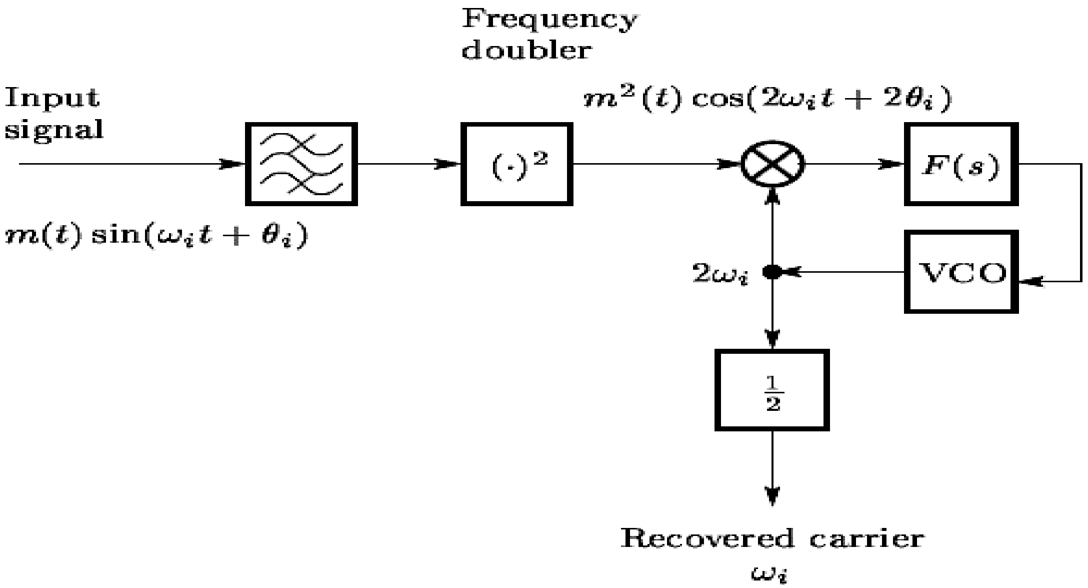
BPSK Demodulator



BPSK Demodulator



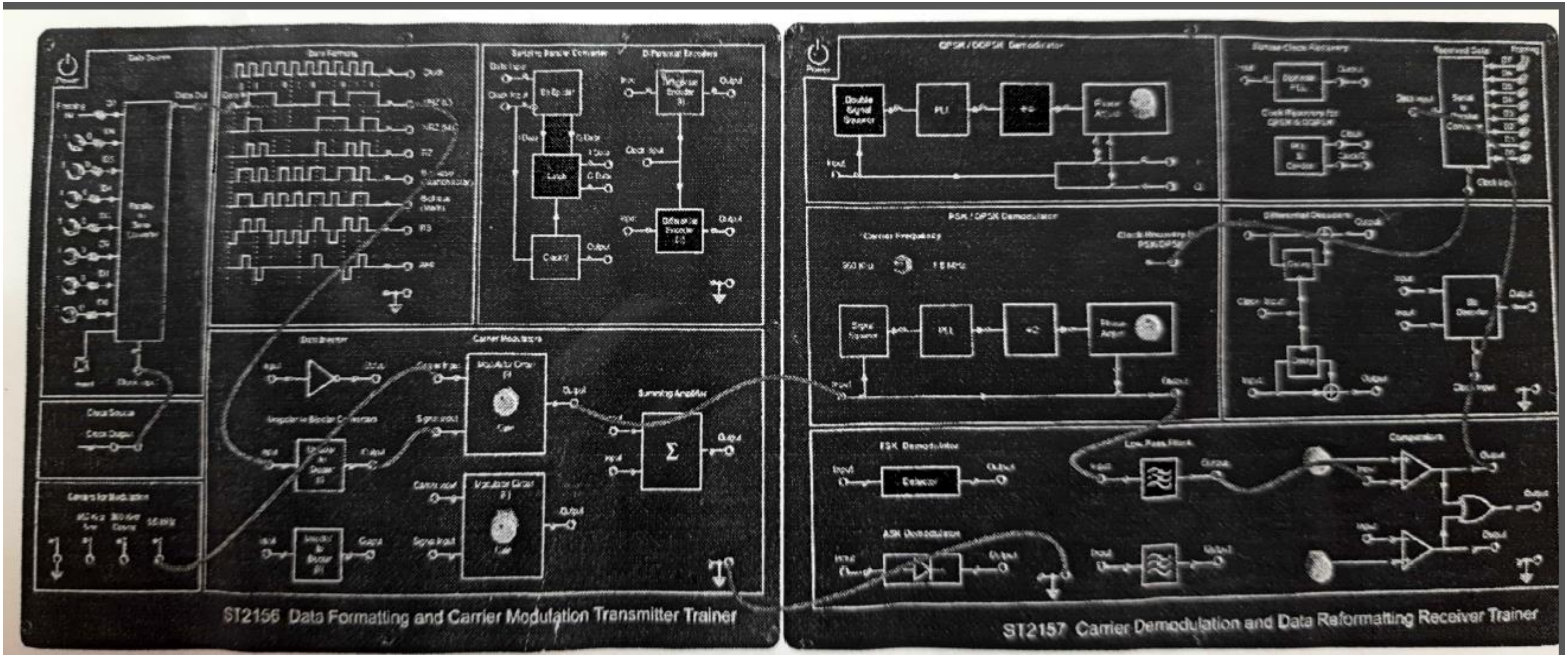
Squaring loop carrier recovery circuit for a BPSK receiver



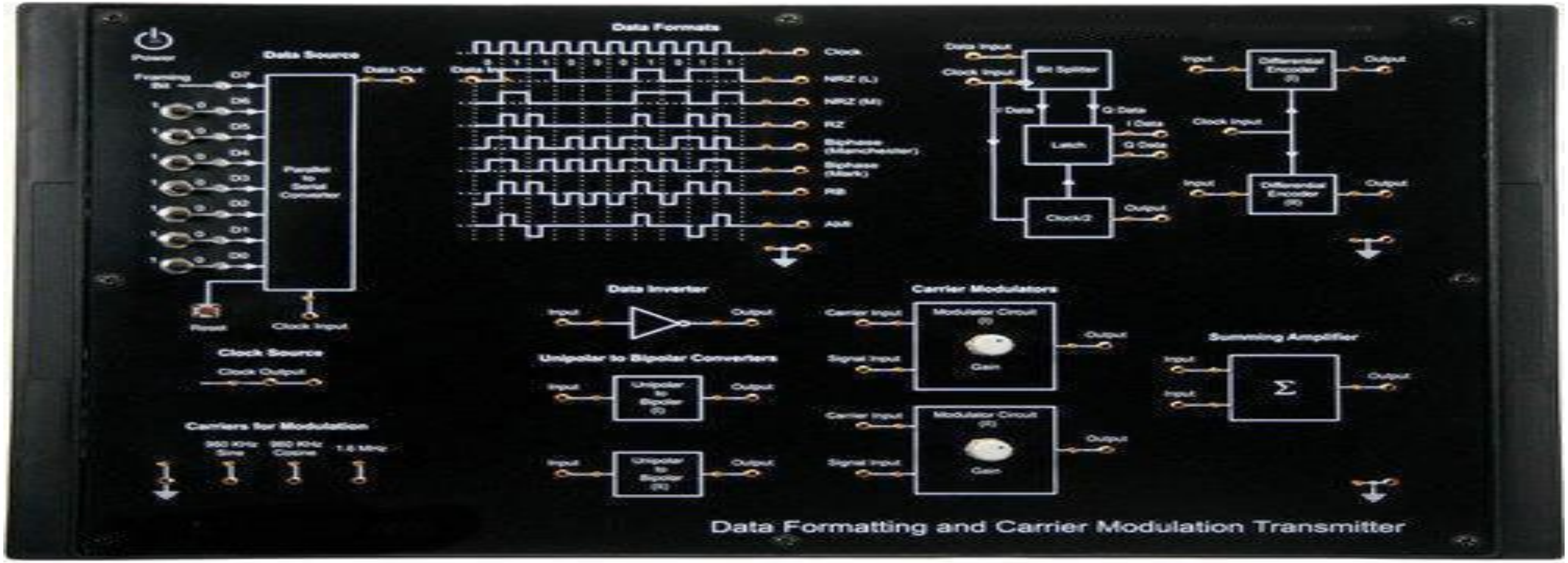
Tabulation

Signals	Amplitude (v)	Time period (s)
Input bit stream		Note down the bit pattern and time period for one bit.
Carrier signal		
BFSK Modulated signal		Note down the time period. For 1 – Phase angle For 0 - Phase angle
Demodulated signal		Note down the bit pattern and time period for one bit.

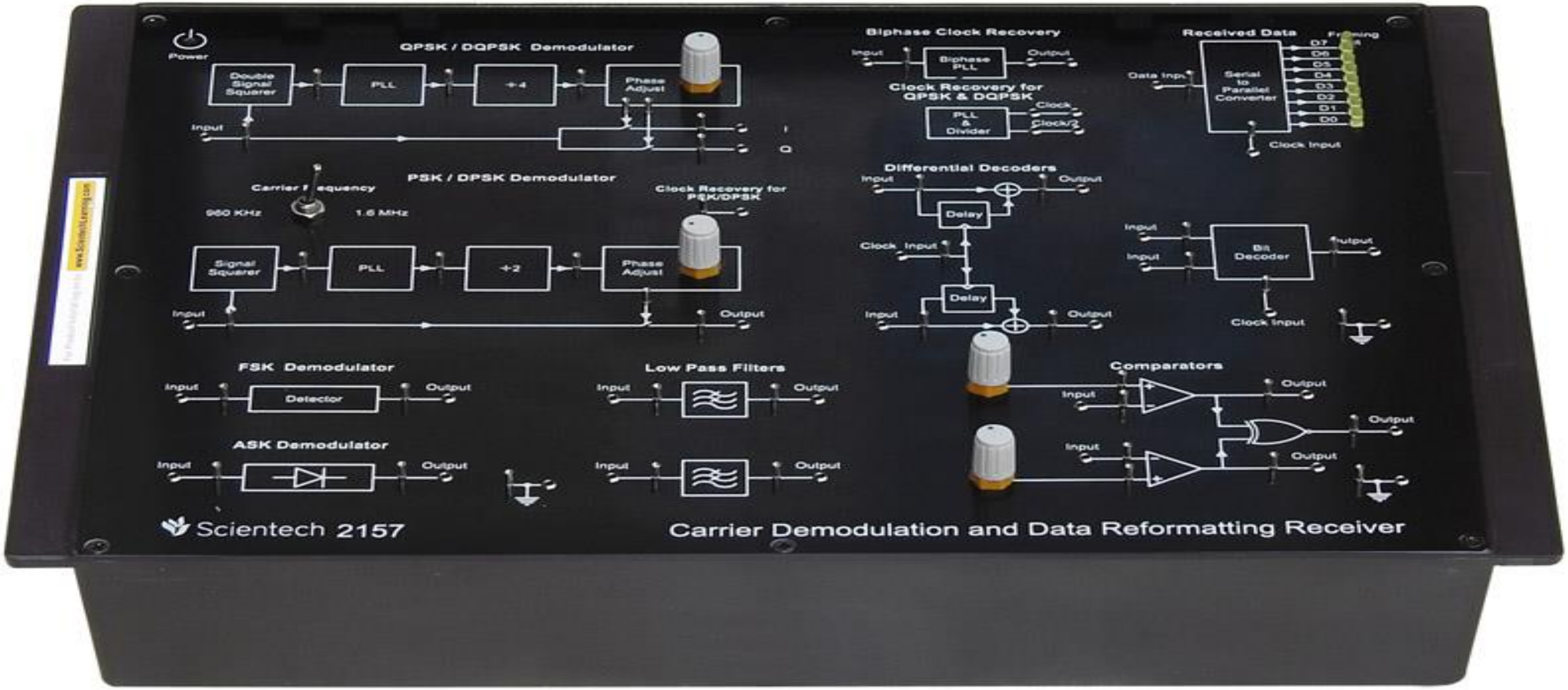
Kit connection diagram



Transmitter kit diagram



Receiver kit diagram



Exp 9
Line Coding Techniques
(Software)

```
%%%%% Unipolar NRZ%%%%

clc
clear all


bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                        %Symbol duration
n=200;                             %bit duration
dt=1/n;


TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency


t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

%line coding :Uni-polar
for(i=0:length(bits)-1)
    if bits(i+1)==1
        x((i*n)+1:(i+1)*n)=1;
    else
        x((i*n)+1:(i+1)*n)=0;
    end
end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('NRZ UNI POLAR-LINE CODING')
axis([0 2500 0 2])

sx=((a^2)/4)*TB*(sinc(x1).^2)+((a^2)/4)*dirac(f);      %power spectra UNIPOLAR
formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title('NRZ UNI POLAR-POWER SPECTRUM-LINE CODING')
```

```
%%%%% Unipolar RZ%%%%

clc
clear all


bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                        %Symbol duration
n=200;                             %bit duration
dt=1/n;


TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency


t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

%line coding :Uni-polar
for(i=0:length(bits)-1)
    if bits(i+1)==1
        x((i*n)+1:(i+0.5)*n)=1;
        x(((i+0.5)*n)+1:(i+1)*n)=0;

    else
         x((i*n)+1:(i+0.5)*n)=0;
        x(((i+0.5)*n)+1:(i+1)*n)=0;
    end
end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('RZ UNI POLAR-LINE CODING')
axis([0 2500 0 2])

sx=((a^2)/4)*TB*(sinc(x1).^2)+((a^2)/4)*dirac(f);      %power spectra UNIPOLAR
formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title('RZ UNI POLAR-POWER SPECTRUM-LINE CODING')
```

```
%%******************* POLAR NRZ********************%%
clc
clear all


bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                        %Symbol duration
n=200;                             %bit duration
dt=1/n;


TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency


t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

%line coding -polar
for(i=0:length(bits)-1)
    if bits(i+1)==1
        x((i*n)+1:(i+1)*n)=1;
    else
        x((i*n)+1:(i+1)*n)=-1;
    end
end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('NRZ POLAR-LINE CODING')
axis([0 2500 -1 1])

sx=(a^2)*TB*(sinc(x1).^2);      %power spectra POLAR formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title('NRZ POLAR-POWER SPECTRUM-LINE CODING')
```

```
%%******************* POLAR RZ********************%%
clc
clear all


bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                        %Symbol duration
n=200;                             %bit duration
dt=1/n;


TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency


t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

%line coding -polar
for(i=0:length(bits)-1)
    if bits(i+1)==1
        x((i*n)+1:(i+0.5)*n)=1;
        x(((i+0.5)*n)+1:(i+1)*n)=0;

    else
         x((i*n)+1:(i+0.5)*n)=-1;
        x(((i+0.5)*n)+1:(i+1)*n)=0;
    end
end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('RZ POLAR-LINE CODING')
axis([0 2500 -1 1])

sx=(a^2)*TB*(sinc(x1).^2);      %power spectra POLAR formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title('RZ POLAR-POWER SPECTRUM-LINE CODING')
```

```
%%***************BI-POLAR - AMI***************
clc
clear all

bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                        %Symbol duration – Full time of
bit sequence
n=200;                             %bit duration
dt=1/n;

TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency

t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

count=0;

%line coding :BI-polar
for(i=0:length(bits)-1)
    if bits(i+1)==1
        if mod(count,2)==0
            x((i*n)+1:(i+1)*n)=1;
        else
            x((i*n)+1:(i+1)*n)=-1;
        end
        count=count+1;
    else
        x((i*n)+1:(i+1)*n)=0;


    end



end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('BI-POLAR-LINE CODING - AMI')
axis([0 2500 -2 2])


 sx=(a^2)*TB*(sinc(x1)).*(sinc(x1)).*(sin(pi*x1)).*(sin(pi*x1));     %power
spectra BIPOLAR formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title(' BI-POLAR-POWER SPECTRUM-LINE CODING-AMI')
```

```
%%****************BI-POLAR - PSEUDOTERNARY***************
clc
clear all

bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                       %Symbol duration – Full time of
bit sequence
n=200;                             %bit duration
dt=1/n;

TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency


t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

count=0;

%line coding :BI-polar
for(i=0:length(bits)-1)
    if bits(i+1)==0
        if mod(count,2)==0
            x((i*n)+1:(i+1)*n)=1;
        else
            x((i*n)+1:(i+1)*n)=-1;
        end
        count=count+1;
    else
        x((i*n)+1:(i+1)*n)=0;


    end


end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('BI-POLAR-LINE CODING-PSUEDOTERNARY')
axis([0 2500 -2 2])


 sx=(a^2)*TB*(sinc(x1)).*(sinc(x1)).*(sin(pi*x1)).*(sin(pi*x1));     %power
spectra BIPOLAR formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title(' BI-POLAR-POWER SPECTRUM-LINE CODING-PSUEDOTERNARY')
```

```
%%************MANCHESTER*****************
clc
clear all


bitrate=1;              %Rb-bitrate
bits=[1 0 0 1 0 1 1 1 0 1 0];
T= length(bits)/bitrate;                        %Symbol duration
n=200;                             %bit duration
dt=1/n;


TB=1/bitrate;                       %1/bitrate value
a=1;
f=0:bitrate/100:2*bitrate;

x1=f*TB;                        % power spectra frequency


t=0:dt:T;                         %t takes the valu 0 to T with innerspace equal
to dt
x=zeros(1,length(t));           %initialisation of x

%line coding :MANCHESTER
for(i=0:length(bits)-1)
    if bits(i+1)==1
        x((i*n)+1:(i+0.5)*n)=1;
        x(((i+0.5)*n)+1:(i+1)*n)=-1;

    else
         x((i*n)+1:(i+0.5)*n)=-1;
        x(((i+0.5)*n)+1:(i+1)*n)=1;
    end
end

subplot(2,1,1)
plot(x);
xlabel('Descrete time');
ylabel('Amplitude');
title('MANCHESTER-LINE CODING')
axis([0 2500 -2 2])
 sx=(a^2)*TB*(sinc(x1/2)).*(sinc(x1/2)).*(sin(pi*x1/2)).*(sin(pi*x1/2));
%power spectra MANCHESTER formila


subplot(2,1,2)
plot(sx);
xlabel('Frequency');
ylabel('Power');
title('MANCHESTER-POWER SPECTRUM-LINE CODING')
```