BLOCK DIAGRAM:

PULSE CODE MODULATION:

- FUNCTION GENERATOR
- CHECKER CIRCUIT
- DEMODULATED O/P
- TIMING LOGIC
- OUTPUT LOGIC
- ERROR CHECKER
- SHIFT REGISTER
- ERROR CHECKCODE
- ANALOG TO DIGITAL CONVERTER
- LATCH
- SHIFT RECEIVER
- DIGITAL TO ANALOG CONVERTER
- ERROR CHECKER
PULSE CODE MODULATION AND DEMODULATION

AIM:

To perform the PCM Encoder and Decoder and plot the characteristic of output waveform.

APPARATUS REQUIRED:

- ST 2103 TDM pulse code modulation transmitter and receiver trainer.
- CRO.
- Connecting wires.

THEORY:

Pulse code modulation (PCM) is a digital scheme for transmitting analog data. The signals in PCM are binary; that is, there are only two possible states, represented by logic 1 (high) and logic 0 (low). Using PCM, it is possible to digitize all forms of analog data, including full-motion video, voices, music, telemetry, and virtual reality (VR).

To obtain PCM from an analog waveform at the source (transmitter end) of a communications circuit, the analog signal amplitude is sampled (measured) at regular time intervals. The sampling rate, or number of samples per second, is several times the maximum frequency of the analog waveform in cycles per second or hertz. The instantaneous amplitude of the analog signal at each sampling is rounded off to the nearest of several specific, predetermined levels. This process is called quantization. The output of a pulse code modulator is thus a series of binary numbers, each represented by some power of 2 bits.

At the destination (receiver end) of the communications circuit, a pulse code demodulator converts the binary numbers back into pulses having the same quantum levels as those in the modulator. These pulses are further processed to restore the original analog waveform.
MODEL GRAPH
PULSE CODE MODULATION

INPUT SIGNAL

CARRIER SIGNAL

PCM MODULATED SIGNAL

DEMODULATED OUTPUT

Time in ms

Amplitude in volts (V)
**Procedure:**

Step 1: Give the connections as per the block diagram.

Step 2: Function Generator of 1 KHz is connected to the channel of the transmitter blocks and also measures the input signal using CRO.

Step 3: Observe the sample instant and PAM output.

Step 4: Observe the PCM modulated output.

Step 5: Connect the PCM modulated output and send to the receiver.

Step 6: Finally monitor the PCM demodulated output using CRO and plot the graph.
<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**BLOCK DIAGRAM:**

**DELTA MODULATION:**
DELTA MODULATION AND DEMODULATION

AIM
To perform delta modulation and demodulation techniques and to plot its wave form characteristics.

APPARATUS REQUIRED
- ST 2103 TDM pulse code modulation transmitter and receiver trainer.
- CRO.
- Connecting wires.

THEORY:
Delta Modulation (DM) is an analog-to-digital and digital-to-analog signal conversion technique used for transmission of voice information. DM is the simplest form of Differential Pulse-Code Modulation (DPCM) where the difference between successive samples are encoded into n-bit data streams. In delta modulation, the transmitted data are reduced to a 1-bit data stream.

The modulator is made by a quantizer which converts the difference between the input signal and the average of the previous steps. In its simplest form, the quantizer can be realized with a comparator referenced to 0, whose output is 1 or 0 if the input signal is positive or negative. It is also a bit-quantizer as it quantizes only a bit at a time. The demodulator is simply an integrator (like the one in the feedback loop) whose output rises or falls with each 1 or 0 received. The integrator itself constitutes a low-pass filter.
MODEL GRAPH
DELTA MODULATION

INPUT SIGNAL

PULSE SIGNAL

INTEGRATOR OUTPUT

BISTABLE OUTPUT

DEMODULATED OUTPUT

Amplitude in volts (V)

Time in ms

Time in ms

Time in ms

Time in ms

www.Vidyarthiplus.com
PROCEDURE

Step1: Give the connections as per the block diagram.

Step2: Function Generator of 1 KHz is connected to the input of the comparator and measures the input signal using CRO.

Step3: Observe the bipolar and integrated output.

Step4: Connect the delta modulated output as input to the demodulator.

Step5: Finally observed the reading of delta demodulated output using CRO and plot the graph.
**TABULATION:**

<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RESULT:
Circuit Diagram:
LINE CODING

AIM:

To study the various Line Coding techniques used in communication systems and draw their corresponding waveforms

COMPONENTS REQUIRED:

1. ST2156 Techbook.
2. 2 mm Banana cable
3. Oscilloscope

THEORY:

Line coding consists of representing the digital signal to be transported, by an amplitude- and time-discrete signal that is optimally tuned for the specific properties of the physical channel. The waveform pattern of voltage or current used to represent the 1s and 0s of a digital signal on a transmission link is called line encoding. The common types of line encoding are unipolar, polar, bipolar and Manchester encoding. The Manchester code is quite popular. It is known as a self-clocking code because there is always a transition during the bit interval. Consequently, long strings of zeros or ones do not cause clocking problems. The format may be selected to meet one or more of the following criteria:

- Minimize transmission hardware
- Facilitate synchronization
- Ease error detection and correction
- Minimize spectral content
- Eliminate a dc component

Classification of Line Codes:
Model Graph:

Clock

Data

NRZ (L)

NRZ (M)

RZ

Biphase (Manchester)

Biphase (Mark)

RB

AMI
Procedure:

1. Connect the power supply of ST2156 but do not turn on the power supplies until connections are made for this experiment.
2. Make the connections as shown in the figure.
3. Switch 'ON' the power.
4. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘Clock In’ and observe the waveforms.
5. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘NRZ (L)’ and observe the waveforms.
6. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘NRZ (M)’ and observe the waveforms.
7. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘RZ’ and observe the waveforms.
8. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘Biphase (manchester)’ and observe the waveforms.
9. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘Biphase (Mark)’ and observe the waveforms.
10. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘RB’ and observe the waveforms.
11. Connect oscilloscope CH1 to ‘Data In’ and CH2 to ‘AMI’ and observe the waveforms.
**TABULATION:**

<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BLOCK DIAGRAM
AMPLITUDE SHIFT KEYING
DATE:

AMPLITUDE SHIFT KEYING

AIM:

To perform Amplitude Shift Keying modulation and demodulation techniques and to plot its wave form characteristics.

APPARATUS REQUIRED:

<table>
<thead>
<tr>
<th>S NO</th>
<th>APPARATUS</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Data formatting and carrier modulation trainer kit</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Power chord</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>CRO</td>
<td>1</td>
</tr>
</tbody>
</table>

THEORY:

Amplitude-Shift Keying (ASK) is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. In an ASK system, the binary symbol 1 is represented by transmitting a fixed-amplitude carrier wave and fixed frequency for a bit duration of $T$ seconds. If the signal value is 1 then the carrier signal will be transmitted; otherwise, a signal value of 0 will be transmitted.

ASK operates as a switch, using the presence of a carrier wave to indicate a binary one and its absence to indicate a binary zero. This type of modulation is called On-Off Keying (OOK), and is used at radio frequencies.
Model Graph:

Message Signal

Carrier Signal

Modulated Signal

Demodulated Signal

Amplitude in volts (V)

Time in ms
PROCEDURE:

1. Connections are made as per the circuit diagram.
2. Switch on the trainer kit.
3. Connect the signal input and carrier input to the modulator circuit.
4. Observe the output on CRO.
5. Note down the amplitude and frequency of the input.
6. Note down the amplitude and frequency of the ASK output.
7. Obtain the amplitude and frequency of the demodulated output.
**TABULATION:**

<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BLOCK DIAGRAM

FREQUENCY SHIFT KEYING
EXP.NO:

DATE:

**FREQUENCY SHIFT KEYING**

**AIM:**

To perform Frequency Shift Keying modulation and demodulation techniques and to plot its wave form characteristics.

**APPARATUS REQUIRED:**

<table>
<thead>
<tr>
<th>S NO</th>
<th>APPARATUS</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Data formatting and carrier modulation trainer kit</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Power chord</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>CRO</td>
<td>1</td>
</tr>
</tbody>
</table>

**THEORY:**

Frequency Shift Keying (FSK) is one of several techniques used to transmit a digital signal on an analogue transmission medium. The frequency of a sine wave carrier is shifted up or down to represent either a single binary value or a specific bit pattern. The simplest form of frequency shift keying is called Binary Frequency Shift Keying (BFSK), in which the binary logic values one and zero are represented by the carrier frequency being shifted above or below the centre frequency. In conventional BFSK systems, the higher frequency represents a logic high (one) and is referred to as the mark frequency. The lower frequency represents a logic low (zero) and is called the space frequency. The two frequencies are equi-distant from the centre frequency.
MODEL GRAPH:

- **Data**
- **Carrier**
- **Modulated Signal**
- **Demodulated Signal**

Axes:
- **Amplitude in volts (V)**
- **Time in ms**
PROCEDURE:

1. Connections are made as per the circuit diagram.
2. Switch on the trainer kit.
3. Connect the signal input and carrier input to the modulator circuit.
4. Observe the output on CRO.
5. Note down the amplitude and frequency of the input.
6. Note down the amplitude and frequency of the FSK output.
7. Obtain the amplitude and frequency of the demodulated output.
### TABULATION:

<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BLOCK DIAGRAM

PHASE SHIFT KEYING
PHASE SHIFT KEYING

AIM:

To Perform Phase Shift Keying modulation and demodulation techniques and to plot its wave form characteristics.

APPARATUS REQUIRED:

<table>
<thead>
<tr>
<th>S NO</th>
<th>APPARATUS</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Data formatting and carrier modulation trainer kit</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Power chord</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>CRO</td>
<td>1</td>
</tr>
</tbody>
</table>

THEORY:

PHASE SHIFT KEYING:

Phase-shift keying (PSK) is a method of digital communication in which the phase of a transmitted signal is varied to convey information.

The simplest PSK technique is called Binary Phase-Shift Keying (BPSK). It uses two opposite signal phases (0 and 180 degrees). The digital signal is broken up timewise into individual bits (binary digits). The state of each bit is determined according to the state of the preceding bit. If the phase of the wave does not change, then the signal state stays the same (0 or 1). If the phase of the wave changes by 180 degrees—that is, if the phase reverses then the signal state changes (from 0 to 1, or from 1 to 0). Because there are two possible wave phases, BPSK is sometimes called biphase modulation.
PROCEDURE:

1. Connections are made as per the circuit diagram.
2. Switch on the trainer kit.
3. Connect the signal input and carrier input to the modulator circuit.
4. Observe the output on CRO.
5. Note down the amplitude and frequency of the input.
6. Note down the amplitude and frequency of the PSK output.
7. Obtain the amplitude and frequency of the demodulated output.
## TABULATION:

<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BLOCK DIAGRAM

QUADRATURE PHASE SHIFT KEYING
QUADRATURE PHASE SHIFT KEYING

AIM:
To perform Quadrature Phase Shift Keying modulation and demodulation techniques and to plot its wave form characteristics.

APPARATUS REQUIRED:

<table>
<thead>
<tr>
<th>S NO</th>
<th>APPARATUS</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Data formatting and carrier modulation trainer kit</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Power chord</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>CRO</td>
<td>1</td>
</tr>
</tbody>
</table>

THEORY:

QUADRATURE PHASE SHIFT KEYING:

Quadrature Phase Shift Keying (QPSK) is the digital modulation technique. Quadrature Phase Shift Keying (QPSK) is a form of Phase Shift Keying in which two bits are modulated at once, selecting one of four possible carrier phase shifts (0, π/2, π, and 3π/2). QPSK perform by changing the phase of the In-phase (I) carrier from 0° to 180° and the Quadrature-phase (Q) carrier between 90° and 270°. This is used to indicate the four states of a 2-bit binary code. Each state of these carriers is referred to as a Symbol.

QPSK perform by changing the phase of the In-phase (I) carrier from 0° to 180° and the Quadrature-phase (Q) carrier between 90° and 270°. This is used to indicate the four states of a 2-bit binary code. Each state of these carriers is referred to as a Symbol. Quadrature Phase-shift Keying (QPSK) is a widely used method of transferring digital data by changing or modulating the phase of a carrier signal. In QPSK digital data is represented by 4 points around a circle which correspond to 4 phases of the carrier signal. These points are called symbols.
MODEL GRAPH:

ENTER INPUT BIT STREAMS [1 0 0 1 0 0 1 1]

- **INPUT BIT STREAMS**
  - Amplitude vs. Time

- **ODD BIT STREAMS WITH TWICE CLOCK PERIOD**
  - Amplitude vs. Time

- **EVEN BIT STREAMS WITH TWICE CLOCK PERIOD**
  - Amplitude vs. Time

- **COSINE WAVESFORM**
  - Amplitude vs. Time

- **SINE WAVESFORM**
  - Amplitude vs. Time

- **COSINE MULTIPLIED WITH ODD BIT STREAM WAVEFORM**
  - Amplitude vs. Time

- **SINE MULTIPLIED WITH EVEN BIT STREAM WAVEFORM**
  - Amplitude vs. Time

- **QPSK WAVEFORM**
  - Amplitude vs. Time
PROCEDURE:

1. Connections are made as per the circuit diagram.
2. Switch on the trainer kit.
3. Connect the signal input and carrier input to the modulator circuit.
4. Observe the output on CRO.
5. Note down the amplitude and frequency of the input.
6. Note down the amplitude and frequency of the QPSK output.
7. Obtain the amplitude and frequency of the demodulated output.
<table>
<thead>
<tr>
<th>Name of the Signal</th>
<th>Amplitude (Volts)</th>
<th>Time Period (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EXP.NO:

DATE:

GENERATION AND DETECTION OF DIGITAL MODULATION TECHNIQUE USING MATLAB

AIM:-

To plot the wave form for Binary Amplitude Shift Keying (BASK) signal using MATLAB for a stream of bits.

SOFTWARE USED:

MATLAB 7.1

THEORY:

AMPLITUDE SHIFT KEYING:

Amplitude-Shift Keying (ASK) is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. In an ASK system, the binary symbol 1 is represented by transmitting a fixed-amplitude carrier wave and fixed frequency for a bit duration of T seconds. If the signal value is 1 then the carrier signal will be transmitted; otherwise, a signal value of 0 will be transmitted.

FREQUENCY SHIFT KEYING:

Frequency Shift Keying (FSK) is one of several techniques used to transmit a digital signal on an analogue transmission medium. The frequency of a sine wave carrier is shifted up or down to represent either a single binary value or a specific bit pattern. The simplest form of frequency shift keying is called Binary Frequency Shift Keying (BFSK), in which the binary logic values one and zero are represented by the carrier frequency being shifted above or below the centre frequency.

PHASE SHIFT KEYING:

Phase-shift keying (PSK) is a method of digital communication in which the phase of a transmitted signal is varied to convey information. The simplest PSK technique is called Binary Phase-Shift Keying (BPSK). It uses two opposite signal phases (0 and 180 degrees). The digital signal is broken up timewise into individual bits (binary digits).

QUADRATURE PHASE SHIFT KEYING:

Quadrature Phase Shift Keying (QPSK) is the digital modulation technique. Quadrature Phase Shift Keying (QPSK) is a form of Phase Shift Keying in which two bits are modulated at once, selecting one of four possible carrier phase shifts (0, \( \pi/2 \), \( \pi \), and \( 3\pi/2 \)). QPSK perform by changing the phase of the In-phase (I) carrier from 0° to 180° and the Quadrature-phase (Q) carrier between 90° and 270°. This is used to indicate the four states of a 2-bit binary code. Each state of these carriers is referred to as a Symbol.
**ALGORITHM:**

**ASK:**
Step 1: Get the input values.

Step 2: Plot the values on the corresponding axis.

Step 3: Perform the ASK operation.

Step 4: Output values are displayed in the command window.

Step 5: The required waveforms are displayed in the output window (figure 1).

**FSK:**
Step 1: Get the input values.

Step 2: Plot the values on the corresponding axis.

Step 3: Perform the FSK operation.

Step 4: Output values are displayed in the command window.

Step 5: The required waveforms are displayed in the output window (figure 2).

**PSK:**
Step 1: Get the input values.

Step 2: Plot the values on the corresponding axis.

Step 3: Perform the PSK operation.

Step 4: Output values are displayed in the command window.

Step 5: The required waveforms are displayed in the output window (figure 3).

**QPSK:**
Step 1: Get the input values.

Step 2: Plot the values on the corresponding axis.

Step 3: Perform the QPSK operation.

Step 4: Output values are displayed in the command window.

Step 5: The required waveforms are displayed in the output window (figure 4).
ASK - MATLAB PROGRAM:

clear;
cle;
b = input('Enter the Bit stream \n'); \%b = [0 1 0 1 1 0];
n = length(b);
t = 0:.01:n;
x = 1:1:(n+1)*100;
for i = 1:n
    for j = i:.1:i+1
        bw(x(i*100:(i+1)*100)) = b(i);
    end
end
bw = bw(100:end);
sint = sin(2*pi*t);
st = bw.*sint;
subplot(3,1,1)
plot(t,bw)
grid on; axis([0 n -2 +2])
subplot(3,1,2)
plot(t,sint)
grid on; axis([0 n -2 +2])
subplot(3,1,3)
plot(t,st)
grid on; axis([0 n -2 +2])
**FSK - MATLAB PROGRAM:-**

clear;
clc;
b = input('Enter the Bit stream
'); %b = [0 1 0 1 1 1 0];
n = length(b);
t = 0:.01:n;
x = 1:1:(n+1)*100;
for i = 1:n
    if (b(i) == 0)
        b_p(i) = -1;
    else
        b_p(i) = 1;
    end
end
for j = i:.1:i+1
    bw(x((i*100):(i+1)*100)) = b_p(i);
end
bw = bw(100:end);
wo = 2*(2*pi*t);
W = 1*(2*pi*t);
sinHt = sin(wo+W);
sinLt = sin(wo-W);
st = sin(wo+(bw).*W);
subplot(4,1,1)
plot(t,bw)
grid on; axis([0 n -2 +2])
subplot(4,1,2)
plot(t,sinHt)
grid on; axis([0 n -2 +2])
subplot(4,1,3)
plot(t,sinLt)
grid on; axis([0 n -2 +2])
subplot(4,1,4)
plot(t,st)
grid on; axis([0 n -2 +2])
Fs=1;
figure %pburg(st,10)
periodogram(st)
PSK - MATLAB PROGRAM:

clear;
cle;
b = input('Enter the Bit stream n '); \% b = [0 1 0 1 1 0];
n = length(b);
t = 0:.01:n;
x = 1:1:(n+1)*100;
for i = 1:n
if (b(i) == 0)
b_p(i) = -1;
else
b_p(i) = 1;
end
for j = i:.1:i+1
bw(x(i*100:(i+1)*100)) = b_p(i);
end
end
bw = bw(100:end);
sint = sin(2*pi*t);
st = bw.*sint;
subplot(3,1,1)
plot(t,bw)
grid on ; axis([0 n -2 +2])
subplot(3,1,2)
plot(t,sint)
grid on ; axis([0 n -2 +2])
subplot(3,1,3)
plot(t,st)
grid on ; axis([0 n -2 +2])
OUTPUT:

Figure 1

MESSAGE WAVEFORM

CARRIER WAVEFORM

FSK WAVEFORM
QPSK - MATLAB PROGRAM:

clear;
clc;
b = input('Enter the Bit stream \n'); %b = [0 1 0 1 1 0];
n = length(b);
t = 0:.01:n;
x = 1:1:(n+2)*100;
for i = 1:n
    if (b(i) == 0)
        b_p(i) = -1;
    else
        b_p(i) = 1;
    end
end
for j = i:.1:i+1
    bw(x(i*100:(i+1)*100)) = b_p(i);
    if (mod(i,2) == 0)
        bow(x(i*100:(i+1)*100)) = b_p(i);
        bow(x((i+1)*100:(i+2)*100)) = b_p(i);
    else
        bew(x(i*100:(i+1)*100)) = b_p(i);
        bew(x((i+1)*100:(i+2)*100)) = b_p(i);
    end
end
if (mod(n,2)~= 0)
    bow(x(n*100:(n+1)*100)) = -1;
    bow(x((n+1)*100:(n+2)*100)) = -1;
end
end
%be = b_p(1:2:end);
%bo = b_p(2:2:end);
bw = bw(100:end);
bew = bew(100:(n+1)*100);
obw = bow(200:(n+2)*100);
cost = cos(2*pi*t);
sint = sin(2*pi*t);
st = bew.*cost+bow.*sint;
subplot(4,1,1)
plot(t,bw)
grid on ; axis([0 n -2 +2])
subplot(4,1,2)
plot(t,bow)
grid on ; axis([0 n -2 +2])
subplot(4,1,3)
plot(t,bew)
grid on ; axis([0 n -2 +2])
subplot(4,1,4)
plot(t,st)
grid on ; axis([0 n -2 +2])
IMPLEMENTATION OF LINEAR BLOCK CODES

AIM:
To implement linear block codes using MATLAB.

APPARATUS REQUIRED:
MATLAB 7.1

THEORY:
Code words are produced on a block by block basis called as linear block codes. Code word is a sequence of symbols. Encoded block of `n` bits is called a code word. The sum of two code words belonging to the code. The all zero word is always a code word. The minimum distance between two code words of a linear code is equal to the minimum weight of the code.

ALGORITHM:
Step 1: Assign block length \( n = 7 \) and message bit \( k = 7 \).
Step 2: Define message and parity matrix.
Step 3: Generate generator matrix of the form \([p;I_{n-k}]\).
Step 4: Multiply each generator matrix row with message column.
Step 5: Linear code is obtained by doing XOR operation with output of matrix formed in step 4.
PROGRAM:
clc;
clear all;
close all;
% input generator matrix
g=input('enter the generator matrix:');
disp('G=')
disp('the order of linear block code for given generator matrix is :')
[n,k]=size(transpose(g))
for i=1:2^k
    for j=k-1:-1:1
        if rem(i-1,2^(-j+k+1))>=2^(-j+k)
            u(i,j)=1;
        else
            u(i,j)=0;
        end
    end
end
u;
disp('the possible code words are:')
c=rem(u*g,2)
disp('the minimum hamming distance d_min for given block code is:')
d_min=min(sum((c(2:2^k,:))'))
% code word
r=input('enter the received code word:')
p=[g(:,n-k+2:n)];
h=[transpose(p),eye(n-k)];
disp('hamming code')
hn=transpose(h);
disp('syndrome of given code word is')
s=rem(r*h,2)
for i=1:size(hn)
    if(ht(i,1:3)==s)
        r(i)=1-r(i);
        break;
    end
end
disp('error s in bit:')
i
disp('the corrected code word s:')
r
MANUAL CALCULATION FOR LINEAR BLOCK CODE
OUTPUT:
enter the generator matrix:
\[
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\( G = \) 

the order of linear block code for given generator matrix is:

\( n = \) 7 

\( k = \) 4 

the possible code words are:

\( c = \)

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix}
\]
the minimum hamming distance $d_{\text{min}}$ for given block code is:

$$d_{\text{min}} = 3$$

enter the received code word: $[1\ 1\ 0\ 1\ 1\ 1\ 1]\$

$$r = 1\ 1\ 0\ 1\ 1\ 1\ 1\$$

hamming code

$$ht =$$
syndrome of given code word is

\[ s = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \]

error s in bit:

\[ i = 3 \]

the corrected code word s:

\[ r = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \]
AIM:

To design a channel equalizer using LMS algorithm in MATLAB.

APPARATUS REQUIRED:

MATLAB 7.1

THEORY:

CHANNEL EQUALISER

Channel equaliser is used to reduce inter symbol interference, in digital communication. Equalisation is done with the help of the Filter. Here we use the Adaptive equaliser.

ADAPTIVE EQUALISER

An adaptive equalizer is an equalizer that automatically adapts to time-varying properties of the communication channel. Many adaptation strategies exist among them, we see

- **LMS**: Here the receiver does not have access to the transmitted signal $x$ when it is not in training mode. If the probability that the equalizer makes a mistake is sufficiently small, the symbol decisions $d(n)$ made by the equalizer may be substituted for $x$.
- **RLS**: A well-known example is the decision feedback equalizer, a filter that uses feedback of detected symbols in addition to conventional equalization of future symbols. Some systems use predefined training sequences to provide reference points for the adaptation process.

LEAST MEAN SQUARES (LMS)

LMS algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time.
**ALGORITHM:**

1. Initialization of all variables and parameters.
2. Generate input sinusoidal signals.
4. Add input signal and noise signal.
5. Calculate the output, bit error rate and SNR.
6. Plot the LMS output waveform.
PROGRAM:

clc;
clear all;
close all;
w=0;
L=512;
M=1024;
G=256;
1=[0:L-1];
fs=10000;
f0=70;
w(:,1)=[0:0];
mu=0.0125;
x1=sin(2*pi*f0*[0:L-1]/fs);
x2=sin(2*pi*f0*[0:L-1]/fs);
pp=1;
x6=0;
while pp<10
    x3=(x1+x2)+0.09*randn;
x6=x6+x3;
    pp=pp+1;
end
x6=x6/10;
subplot(3,1,1)
plot(x6)
grid
title('signal')
xlabel('time(sec)')
ylabel('amplitude(volt)')
x=0.5*randn(G,L);
subplot(3,1,2)
plot(x,'b')
grid
title('signal+noise')
xlabel('time(sec)')
ylabel('amplitude(volt)')

for j=1:G
for i=1:L
    y(j,i)=w(:,i)'+[x3(i)*randn*0.09];
    e(j,i)=x(j,i)-y(j,i);
    w(:,i+1)=w(:,i)+2*0.0125*e(j,i)+[x3(i)*randn*0.09];
end
EE(j,:)=fft(e(j,:),M);
end
u=(sum(abs(EE).^2)/(G))/max((sum(abs(EE).^2)/(G)));
w=w(1:512);
subplot(3,1,3)
plot(w)
grid
title(‘LMS output’)
xlabel(‘time(sec)’)
ylabel(‘amplitude(volt)’)
xg=(sum((x3-w)))/(length(x3));
BER=abs(xg)
sn=max(u);
nn=max(x3);
SNR=nn/sn
OUTPUT: