American International University- Bangladesh

Department of Computer Engineering

COE 3201: Data Communication Laboratory

Title: Message Passing and Receiving Using Modulator (part 1: Transmitter Side)

Abstract:

This experiment is designed to-

1.To understand the concept of message encoding and decoding.
2.To understand the concept of serial transmission and reception of message.
3.To develop understanding of data transmission and reception process.

Introduction:

Consider the problem of transmitting and receiving a text message, such as

Data Communication is fun!

over a waveform channel such as a twisted pair cable or a wireless RF (radio frequency) link. The design of a system that can accomplish this task requires the following ingredients:

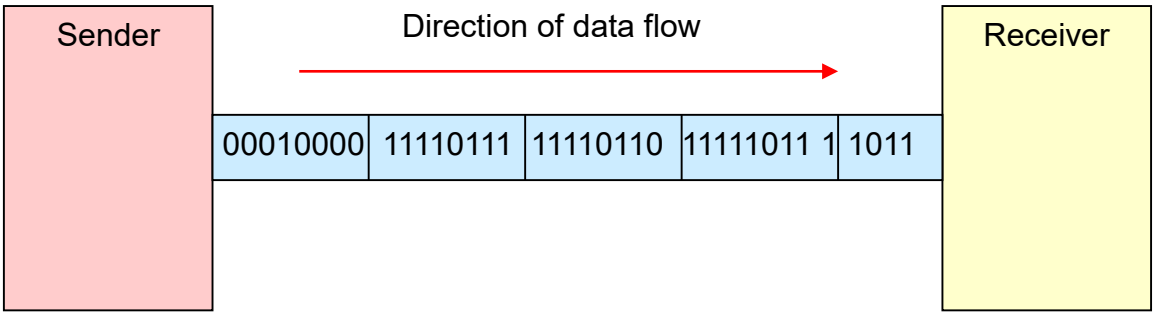
In transmitter side:

1. **Step 1:** Encoding of the letters of the alphabet, the numbers, punctuation, etc. For example, “A” could be encoded as 0, “B” as 1, “C” as 2, etc.
2. **Step 2:** Conversion of the encoded message into a serial data stream, e.g., of 0’s and 1’s in the case of a binary transmission system.
3. **Step 3:** Modulation by the serial data stream of a CT waveform that can be transmitted through the waveform channel.

In receiver side:

4. **Step 4:** Demodulation of the received waveform at the output of the waveform channel to obtain the received serial data stream.
5. **Step 5:** Conversion of the received serial data stream to a sequence of character codes.
6. **Step 6:** Decoding of the received character codes to the received message.

Step 1: Decimal to Serial Binary Conversion: To transmit a message over a communication channel with one input and one output, the bits of each ASCII encoded character need to be sent serially, one after another. One convention that needs to be established is whether the LSB (least significant bit) or MSB (most significant bit) of each code is sent first.



Synchronous transmission

Another consideration is how many bits should be sent per character. Even though the ASCII code has seven bits, it is customary to send 8 bits per character and set the MSB to zero. Thus, if the convention of sending the LSB first is used, the word **Red** (decimal 82,101,100) is converted to the binary data sequence (commas are only shown for clarity, they are not part of the data sequence)

0 1 0 0 1 0 1 0, 1 0 1 0 0 1 1 0, 0 0 1 0 0 1 1 0

In Matlab the following function (**asc2bin)** can be built to convert from a text string txt to a binary data sequence **dn**:

```
function dn = asc2bn(txt)

dec=double(txt)   %Text to ASCII (decimal)
p2=2.^(0:-1:-7)   % 2^0,2^-1,.......,2^-7
B=mod(floor(p2'*dec),2) %Decimal to binary conversion
                         %Columns of B are bits of chars
dn=reshape(B,1,numel(B));%Bytes to serial conbversion
end
```

```
clc;
clear all;
close all;
Transmitted_Message= 'Red'
%Converting Information Message to bit%
x=asc2bn(Transmitted_Message); % Binary Information
```

```
bp=.000001;
% bit period
disp(' Binary information at Trans mitter :');
disp(x);
```

Transmitted_Message =

Red

Binary information at Trans mitter :

Columns 1 through 16

0 1 0 0 1 0 1 0 1 0 1 0 0 1 1 0

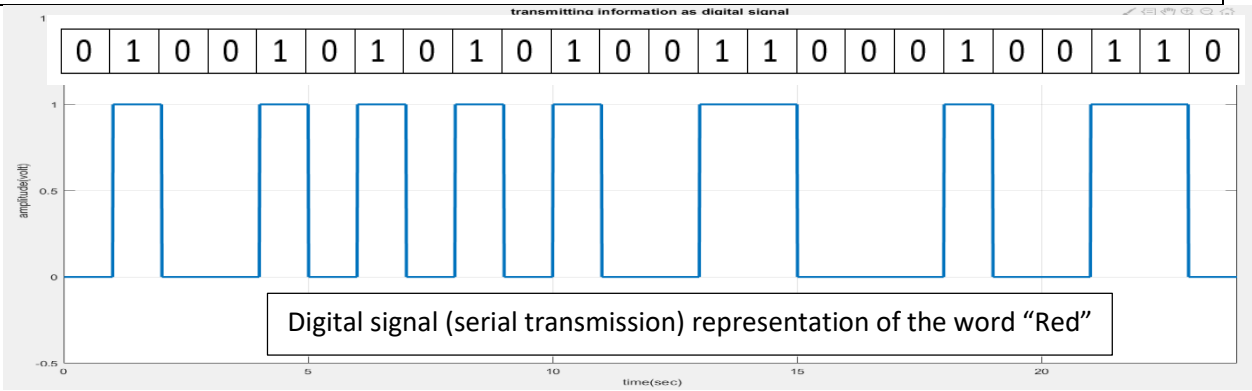
Columns 17 through 24

0 0 1 0 0 1 1 0

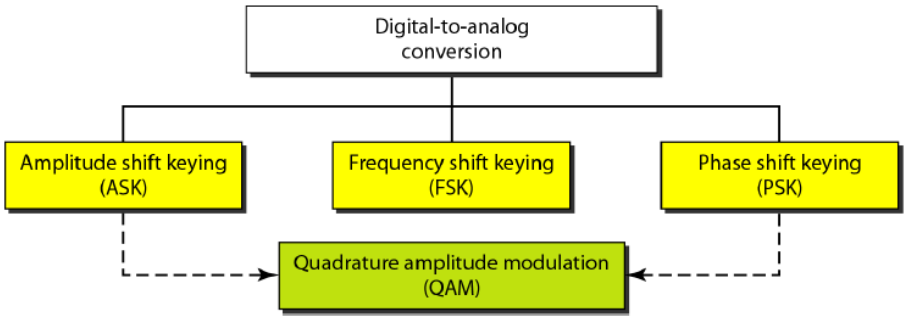
Step 2: Representation of transmitting binary information as digital signal

```
%XX representation of transmitting binary information as
digital signal XXX
bit=[];
for n=1:1:length(x)
    if x(n)==1;
       se=5*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(4,1,1);
plot(t1,bit,'lineWidth',2.5);grid on;
axis([ 0 bp*length(x) -.5 6]);
```

```
ylabel('amplitude(volt)');
xlabel(' time(sec)');
title('Transmitting information as digital signal');
```

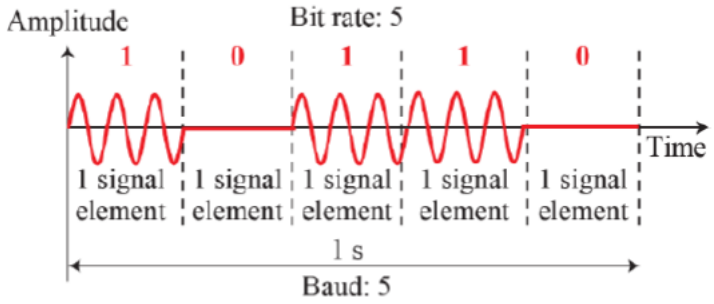


Step 3: Modulation of serial data stream to waveform: In Matlab the **asc2bin** (See A Message Passing and Receiving Using PAM (Part 1)) function can convert a text string txt to a binary data sequence. The binary data sequence is then converted into digital signals. Now, if we want to transmit the signal, we need to modulate the signal int analog.



We will perform ASK modulation. ASK: In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements Both frequency and phase remain constant while the amplitude changes.

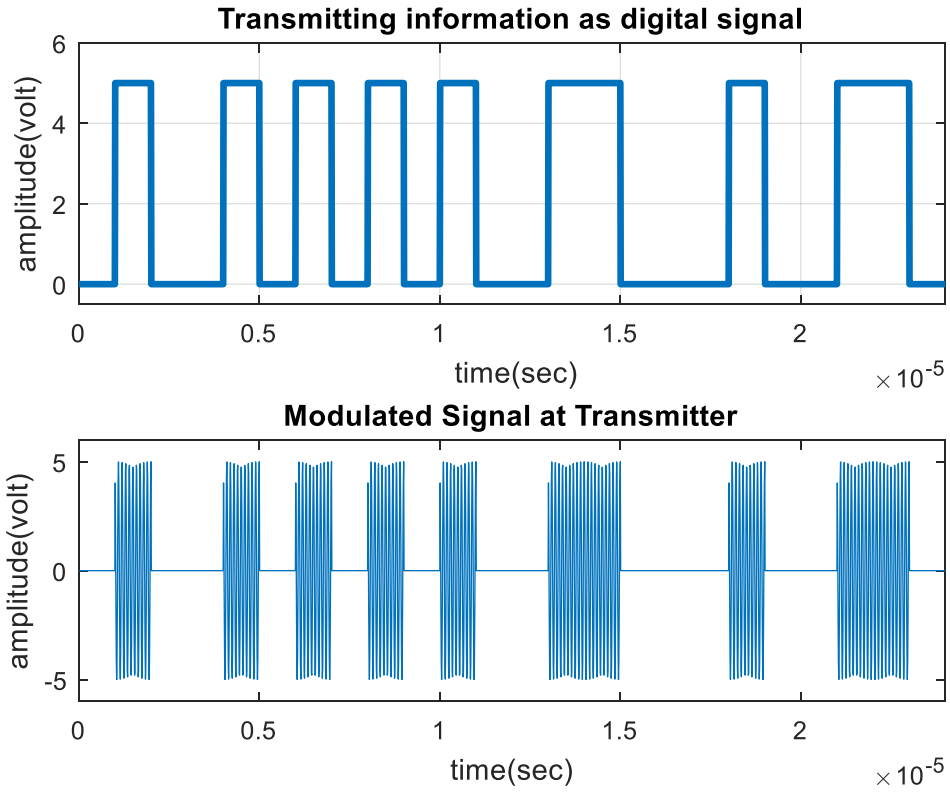
Binary amplitude shift keying



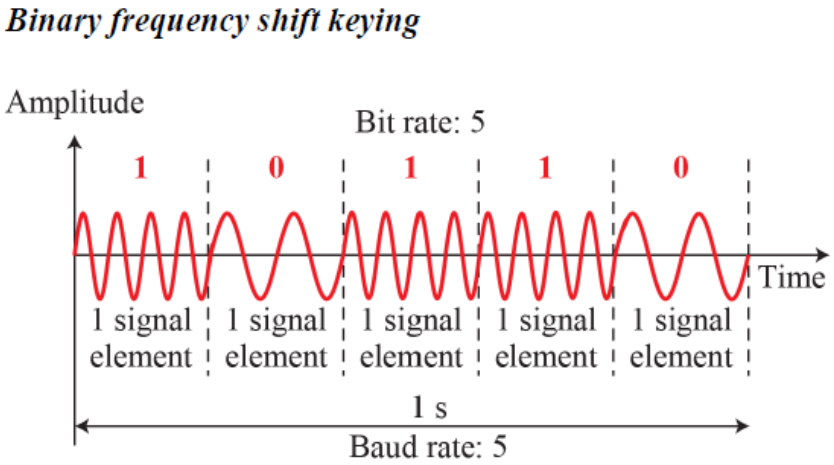
Sample code for ASK modulation in transmitting side:

```
%XXXXXXXXXXXXXXXXXXXXXXXX Binary-ASK modulation
XXXXXXXXXXXXXXXXXXXXXXXXXXXX%
A1=5;                          % Amplitude of carrier signal for
information 1
A2=0;                           % Amplitude of carrier signal for
information 0
br=1/bp;
% bit rate
f=br*10;                                                        %
carrier frequency
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(4,1,2);
plot(t3,m);
axis([ 0 bp*length(x) -6 6]);
xlabel('time(sec)');
ylabel('amplitude(volt)');
title('Modulated Signal at Transmitter');
```

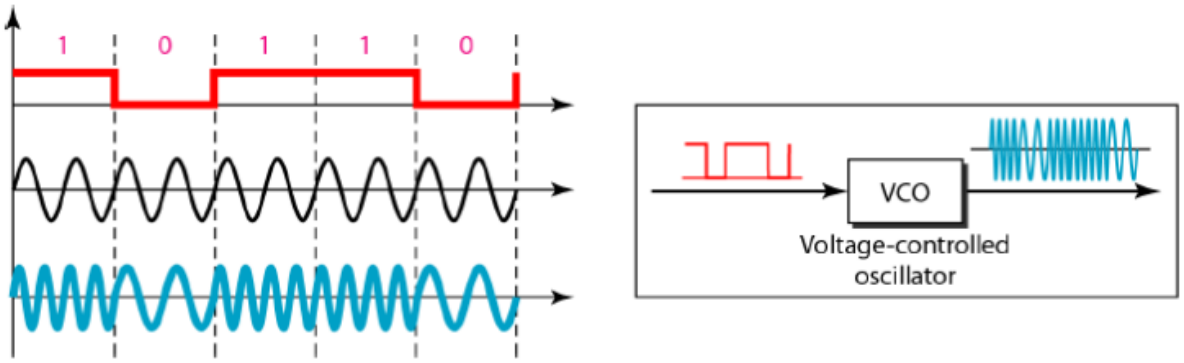
Output:



FSK: In frequency shift keying, the frequency of the carrier signal is varied to represent data The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes Both peak amplitude and phase remain constant for all signal elements



Implementation of BFSK:



We can perform FSK modulation for the input text “Digital” as well.

Performance Task for Lab Report: (your ID = AB-CDEFG-H)

(a) Generate a function which will convert a text message into binary bit sequence.

(b) Generate the digital signal where the bit duration is 1 sec.

(c) Formulate the code in (a), so that it can perform asynchronous transmission (10 bits).

(d) Write necessary code so that it will ask the users whether to perform synchronous/asynchronous transmission and then perform accordingly (a, b).