

SANJIT ANAND U19EC008

21-02-2022

PRACTICAL ASSIGNMENT - 5

WIRELESS AND MOBILE COMMUNICATION

AIM:

To study CDMA spreading/despreading techniques and apply it on the Communication link in MATLAB.

APPARATUS:

MATLAB Software

THEORY:

Multiple Access Techniques

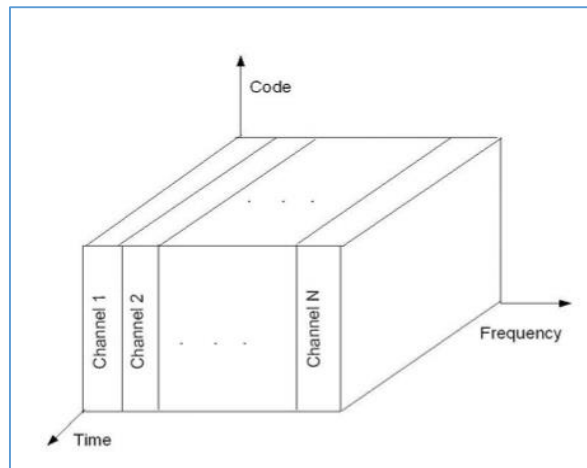
Multiple access techniques are used to allow a large number of mobile users to share the allocated spectrum in the most efficient manner.

- As the spectrum is limited, so the sharing is required to increase the capacity of cell or over a geographical area by allowing the available bandwidth to be used at the same time by different users.
- And this must be done in a way such that the quality of service doesn't degrade within the existing users. –A cellular system divides any given area into cells where a mobile unit in each cell communicates with a base station.
- The main aim in the cellular system design is to be able to increase the capacity of the channel i.e. to handle as many calls as possible in a given bandwidth with a sufficient level of quality of service.

These includes mainly the following:

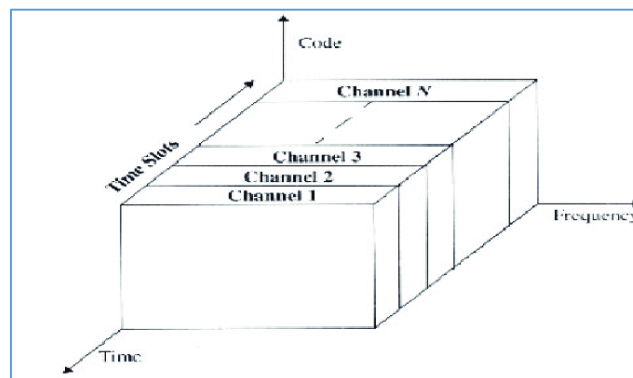
1. Frequency division multiple-access (FDMA)
2. Time division multiple-access (TDMA)
3. Code division multiple-access (CDMA)

1. Frequency division multiple-access (FDMA)



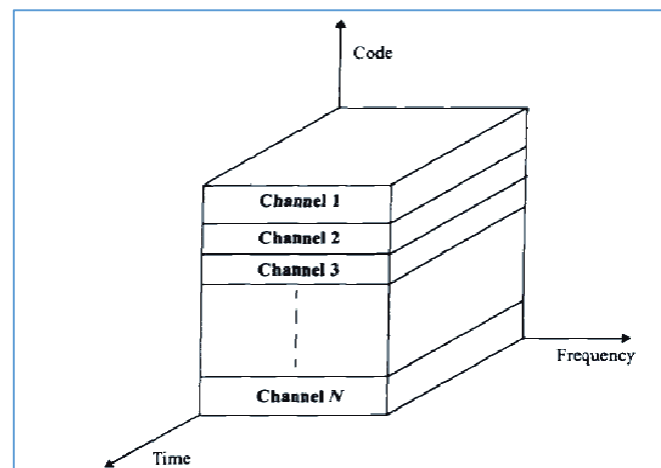
- Each individual user is assigned a pair of frequencies while making or receiving a call as shown in Figure.
- One frequency is used for downlink and one pair for uplink. This is called frequency division duplexing (FDD).

2. Time division multiple-access (TDMA)

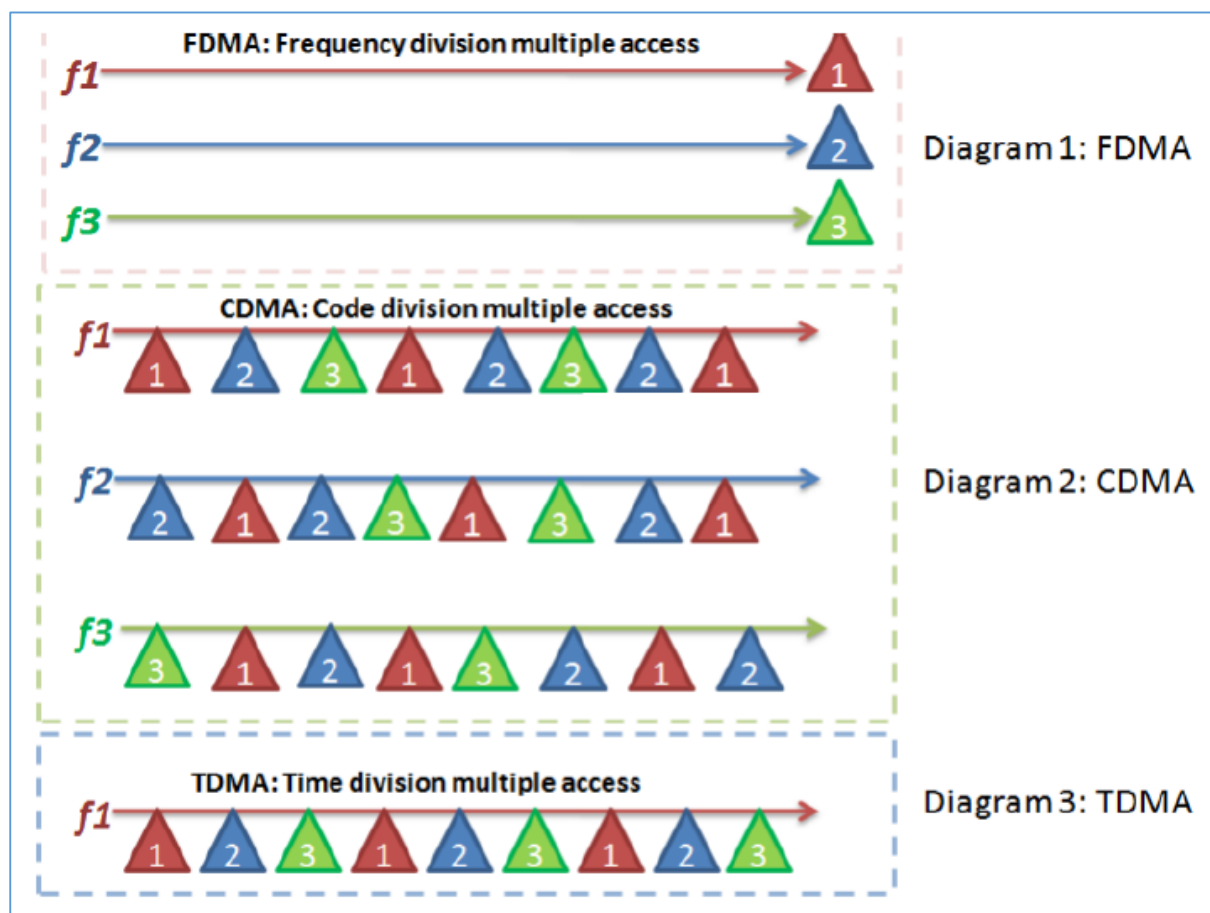


- In digital systems, continuous transmission is not required because users do not use the allotted bandwidth all the time. In such cases, TDMA is a complimentary access technique to FDMA.
- Global Systems for Mobile communications (GSM) uses the TDMA technique.
- In TDMA, the entire bandwidth is available to the user but only for a finite period of time. The users are allotted time slots during which they have the entire channel bandwidth at their disposal, as shown in Figure

3. Code division multiple-access (CDMA)

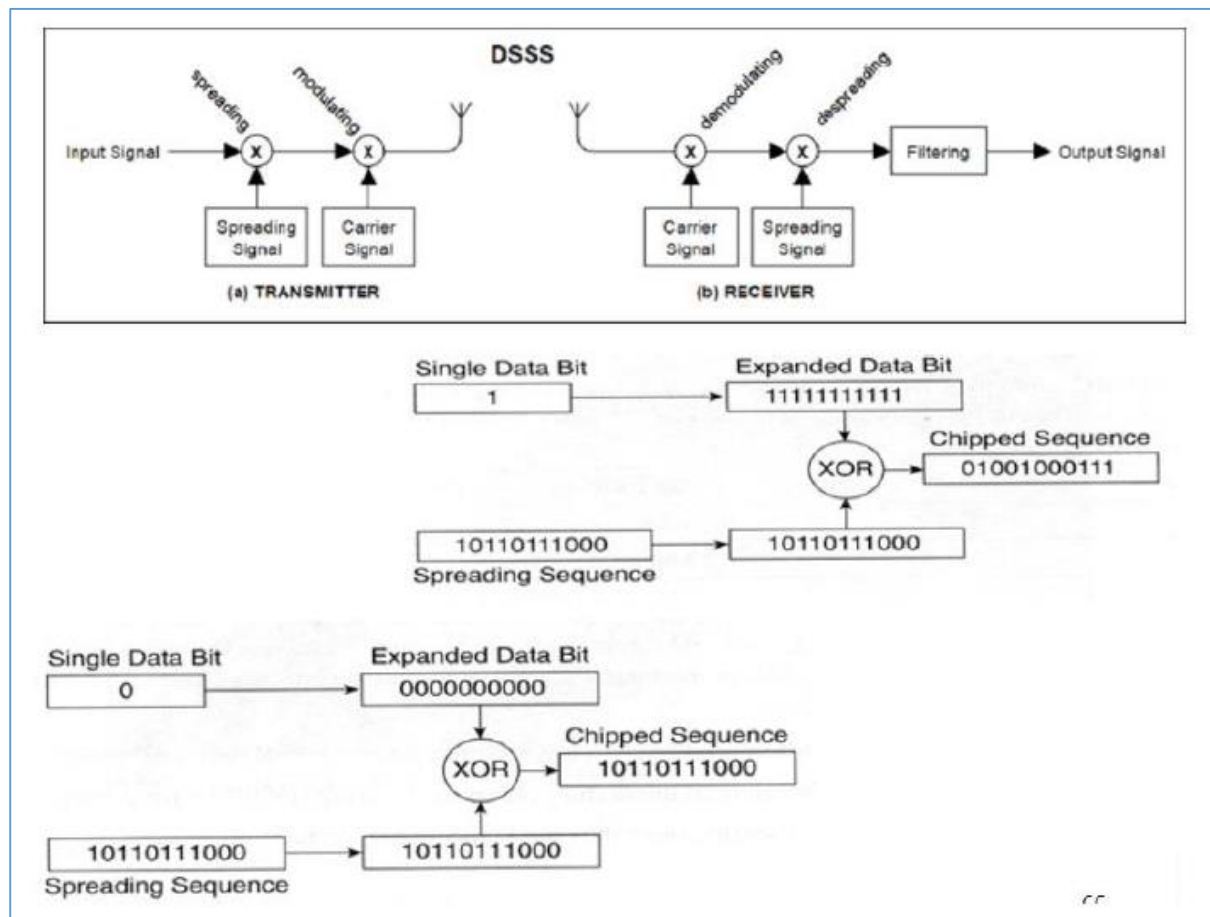


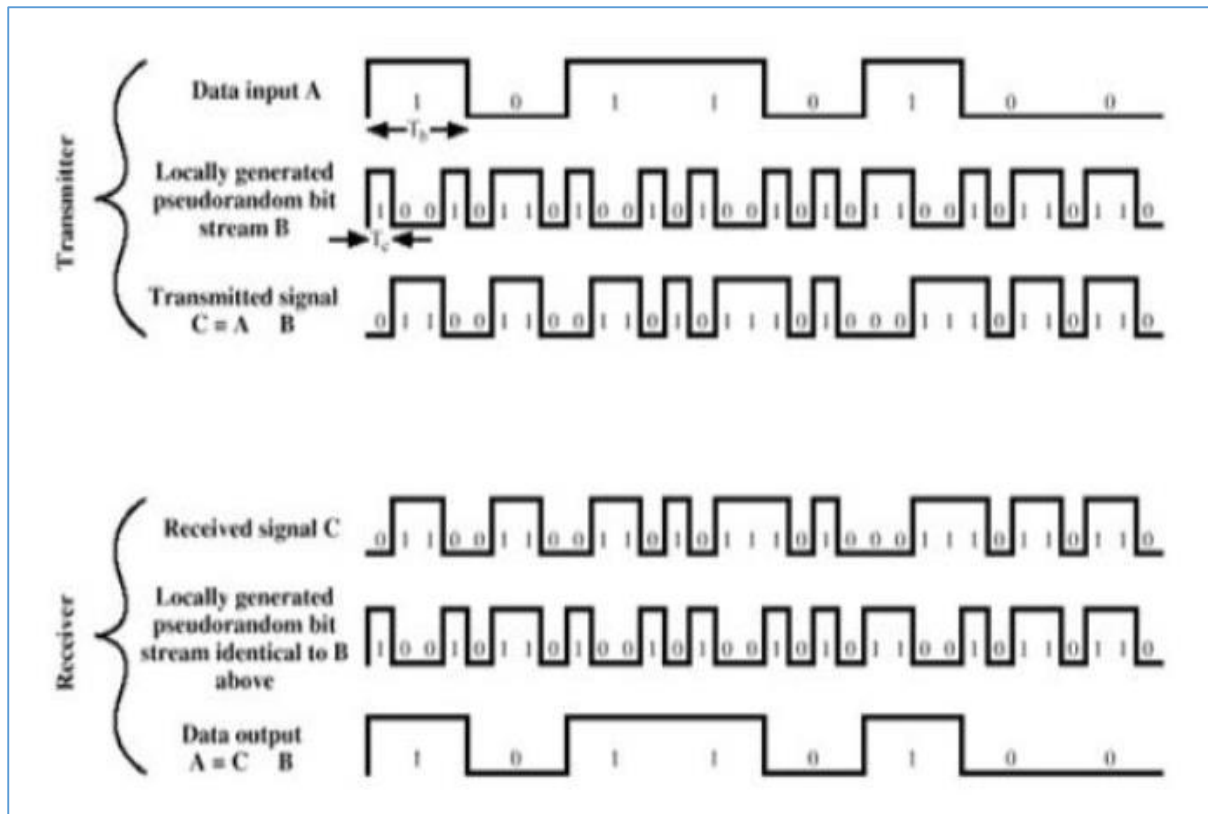
- In CDMA, the same bandwidth is occupied by all the users, however they are all assigned separate codes, which differentiates them from each other shown in Figure
- CDMA utilizes a spread spectrum technique in which a spreading signal (which is uncorrelated to the signal and has a large bandwidth) is used to spread the narrow band message signal.



Direct Sequence Spread Spectrum (DSSS)

- In DS-SS, the message signal is multiplied by a Pseudo Random Noise Code.
- Each user is given his own codeword which is orthogonal to the codes of other users and in order to detect the user, the receiver must know the codeword used by the transmitter.





PROCEDURE:

Part 1

- Generate a binary message signal using **rand** function. Take size as (1000,1) initially and later change it as per (1XXX,1), where XXX is your three digit roll No.
- Generate a PN sequence using **commsrc.pn** from communication system tool box of MATLAB.
- Expand the message bit using **repmat**, so as to make it of size [1000, 8]
- $\text{Xor}(\text{msg}, \text{pn})$ size[1000,8]:
- Convert the vector into row vector: **Spread data [8000, 1]**
- Change the binary data into uint8 before modulation
- Choose mod=4, QPSK/QAM
- Add awgn for a single value of SNR
- Demodulate the signal using `qamdemod` / `pskdemod`
- Using reshape, convert the serial demodulated data vector to parallel before despreading. [1000, 8]
- Despread the signal by using xor function
despread_data= xor(demod_data, pn)
- Contract the data to original format by

- `msg_rx=round(mean(despread_data,1)); mean(x,1)` row vector
`mean(x,2)` column vector
- Example: $[1\ 0\ 1\ 1\ 1\ 1\ 1\ 1] = 7/8 = 0.875$
- Calculate the BER
 - `BER=mean(abs(msg_rx-msg));`
 - This is for a single value or SNR.
- Now, vary SNR `[-10:1:10]`
 - Plot BER vs SNR
 - Output plots [choose msg bits=4]
- Msg signal
- PN sequence
- Spreaded data
- Modulated signal
- Demodulated signal
- Despreaded signal
- Received message signal
- SNR versus BER plot [msg bits=1000]

Part 2

- Choose 3 different message signals of same length using rand function.
- Allocate three PN sequence for each message signal.
- Perform spreading, modulation, demodulation, despreading same as in Part-I for three messages and three
 - PN sequence.
 - Plot BER vs SNR for three messages.

MATLAB CODE & OUTPUTS:

Part 1: a) Modulated and Demodulated Signal

```
% Lab5 U19EC008
% Modulated, Demodulated
clc;
clear all;
close all;

% Generating Transmitted Signal (4x1)
msg = randi([0,1],4,1);

% Repeating message signal
msg_re = repmat(msg,[1,8]);

H = commsrc.pn('GenPoly', [3 2 0], ...
```

```

        'InitialStates',[0 0 1], ...
        'CurrentStates', [0 0 1],...
        'Mask', [0 0 1], ...
        'NumBitsOut', 8);

% Generating PN sequence
pn = generate(H);
rn = pn;
% Repeating PN sequence
pn = repmat(pn',[4, 1]);

% taking xor (Spreaded Data)
x = xor(msg_re,pn);
spread_i = x(:);
spread_i = uint8(spread_i);

snr = 4;
% Modulated Signal
y1 = qammod(spread_i,4);
y2 = awgn(y1,snr);

% Demodulated Signal
y3 = qamdemod(y2,4);
y3 = reshape(y3,[],8);

% Despreaded Signal
despread_d = xor(y3,pn);

% Received Signal
rxsignal = round(mean(despread_d,2));
BER = mean(abs(rxsignal - msg));

% PLOTS
subplot(7,1,1);
plot(msg);
title('U19EC008 Message Signal');

subplot(7,1,2);
plot(rn);
title('U19EC008 Pseudo Random Noise');

subplot(7,1,3);
plot(spread_i);
title('U19EC008 Spreaded Data');

subplot(7,1,4);
plot(y1);
title('U19EC008 Modulated Signal');

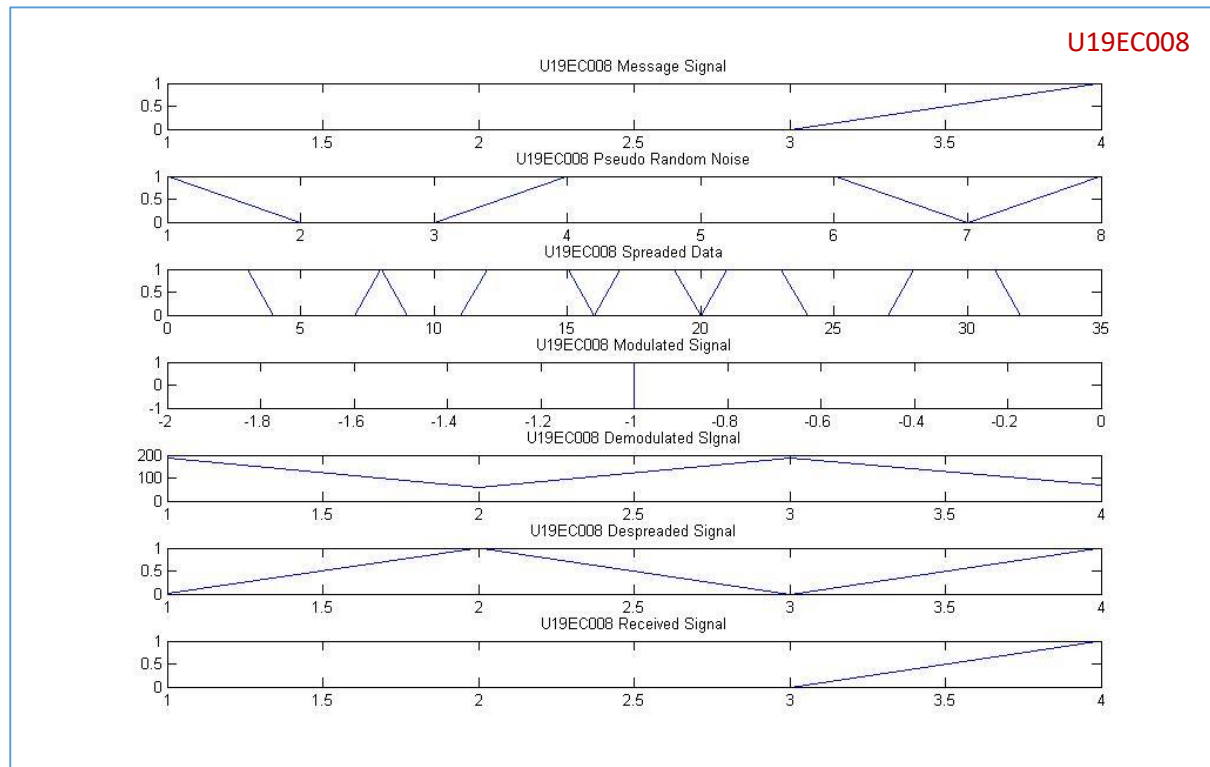
subplot(7,1,5);
plot(bi2de(y3));
title('U19EC008 Demodulated Signal');

subplot(7,1,6);
plot(despread_d);
title('U19EC008 Despreaded Signal');

```

```
subplot(7,1,7);
plot(rxsignal);
title('U19EC008 Received Signal');
```

Output- Part 1: a) Modulated and Demodulated Signal



Part 1: b) Signal to Noise Ratio vs Bit Error Rate

```
% Lab5 U19EC008
% Modulated, Demodulated
clc;
clear all;
close all;

% Roll No. = U19EC008
msgbits = 1008;
% Generating Transmitted Signal (1008x1)
msg = randi([0,1],msgbits,1);
% Repeating message signal
msg_re = repmat(msg,[1,8]);

% Generating PN sequence
pn = generate(commsrc.pn('GenPoly', [3 2 0], ...
    'InitialStates',[0 0 1], ...
    'CurrentStates', [0 0 1],...
    'Mask', [0 0 1], ...
    'NumBitsOut', 8));

pn = reshape(pn,[1,8]);
% Repeating PN sequence
```



```

pn = repmat(pn,msgbits,1);

% taking xor (Spreaded Data)
x = xor(msg_re,pn);
spread_i = x(:);
spread_i = uint8(spread_i);
BER=[];

% For SNR values between -10 to 10
for snr = -10:1:10
    y1 = qammod(spread_i,4);
    y1 = awgn(y1,snr);

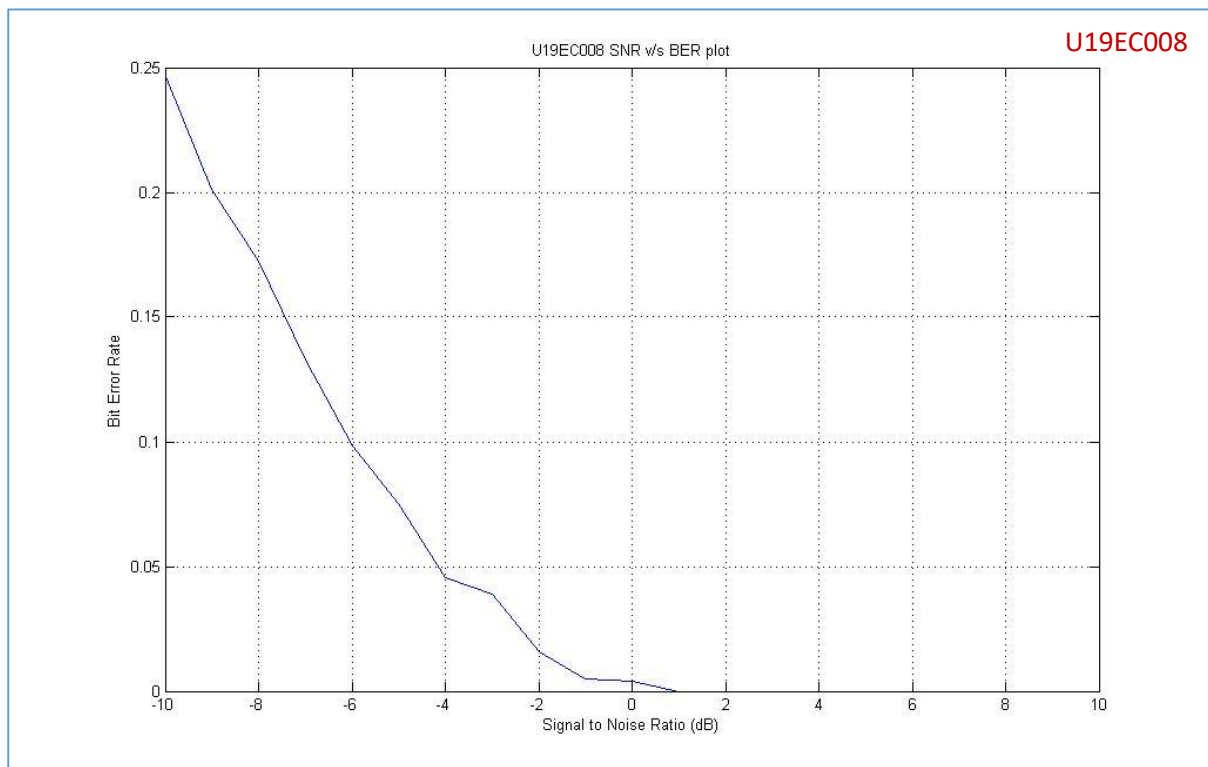
    y2 = qamdemod(y1,4);
    y2 = reshape(y2,[msgbits,8]);
    despread_d = xor(y2,pn);
    msg_rx = round(mean(despread_d,2));

    ber = mean(abs(msg_rx - msg));
    BER = [BER ber];
end

% PLOT
snr = -10:1:10;
plot(snr,BER);
title('U19EC008 BER v/s SNR plot');
xlabel('Signal to Noise Ratio (dB)');
ylabel('Bit Error Rate');
grid on;

```

Output - Part 1: b) Signal to Noise Ratio vs Bit Error Rate



Part 2: SNR vs BER for 3 different signals

```
% Lab5 U19EC008
% SNR vs BER for 3 different Signals
clc;
clear all;
close all;

% Roll Number: U19EC008
msgbits = 1008;
colors = ['r', 'g', 'b'];

for j=1:3
    % Generating Transmitted Signal (1008x1)
    msg = randi([0,1],msgbits,1);
    % Repeating message signal
    msg_re = repmat(msg,[1,8]);

    % Generating PN sequence
    if j==1
        H = commsrc.pn('Genpoly',[3 2 0],...
            'InitialStates',[0 0 1],...
            'CurrentStates',[0 0 1],...
            'Mask',[0 0 1],...
            'NumBitsOut',8);
        pn = generate(H);
    elseif j==2
        H = commsrc.pn('Genpoly',[4 3 0],...
            'InitialStates',[0 0 0 1],...
            'CurrentStates',[0 0 0 1],...
            'Mask',[0 0 0 1],...
            'NumBitsOut',8);
        pn = generate(H);
    else
        H = commsrc.pn('Genpoly',[5 3 0],...
            'InitialStates',[0 0 0 0 1],...
            'CurrentStates',[0 0 0 0 1],...
            'Mask',[0 0 0 0 1],...
            'NumBitsOut',8);
        pn = generate(H);
    end

    pn = reshape(pn,[1,8]);
    % Repeating PN sequence
    pn = repmat(pn,msgbits,1);

    % taking xor (Spreaded Data)
    x = xor(msg_re,pn);
    spread_i = x(:);
    spread_i = uint8(spread_i);

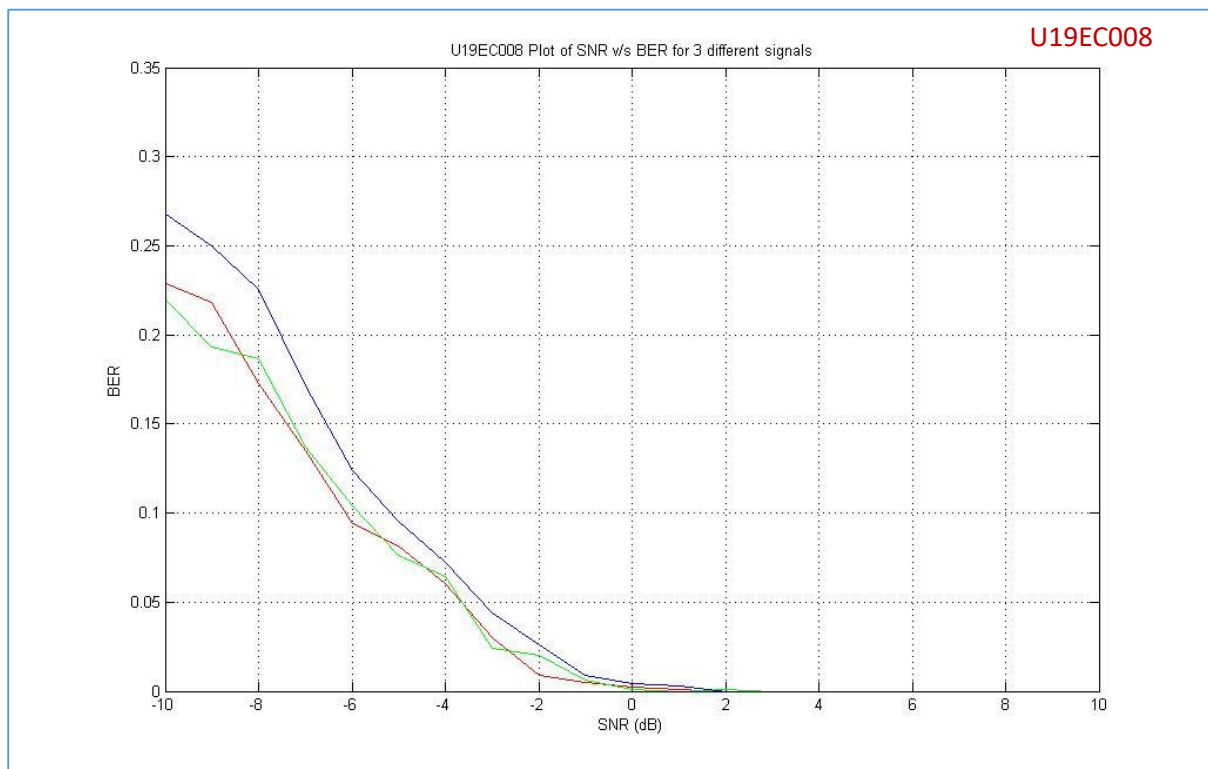
    BER=[];
    for snr = -10:1:10
        y1 = gammod(spread_i,4);
        y1 = awgn(y1,snr);
```

```

y2 = qamdemod(y1,4);
y2 = reshape(y2,[msgbits,8]);
despread_d = xor(y2,pn);
msg_rx = round(mean(despread_d,2));
ber = mean(abs(msg_rx - msg));
BER = [BER ber];
end
% PLOTS
snr = -10:1:10;
plot(snr,BER, 'DisplayName',sprintf('msg=%d',j),
'color',colors(j));
title('U19EC008 Plot of SNR v/s BER for 3 different signals');
xlabel('SNR (dB)');
ylabel('BER');
grid on;
legend;
hold on;
end

```

Output - Part 2: SNR vs BER for 3 different signals



CONCLUSION:

In this experiment, we have implemented CDMA (Code Division Multiple Access) spreading/dispreading techniques and applied it on the communication link using MATLAB and plotted Signal to Noise Ratio (SNR) versus Bit Error Rate (BER) plot. We have also plotted SNR vs BER graph for 3 different signals. We have observed that when the value of SNR is negative, the value of BER is high, but as the value of SNR becomes non-negative, the value of BER tends to zero. We have observed that in DS-SS, the message signal is multiplied by a Pseudo Random Noise Code (PRN). Each user is given his own codeword which is orthogonal to the codes of other users and in order to detect the user, the receiver must know the codeword used by the transmitter.