Department of Computer Engineering

Academic Term: July-Oct 2020

Class : B.E Computer Sem -VII
Subject: Mobile Communication And Computing

Practical No:	**3**
Title:	To implement Code Division Multiple Access (CDMA)
Date of Performance:	24-09-2020
Date of Submission:	25-09-2020
Roll No:	8364
Name of the Student:	Vedant Sahai

Evaluation:

Sr. No	Rubric	Grade
1	**On time Completion & Submission (2)**	
2	**Output (3)**	
3	**Code Optimization (3)**	
4	**Knowledge of the topic (2)**	
5	**Total (10)**	

Signature of the Teacher :

PRACTICAL - 3

Title: : To implement Code Division Multiple Access (CDMA)

Objective: To study code division Multiplexing.

Reference: Mobile communication by Schiller, Mobile Computing by RajKamal

Prerequisite: Knowledge of orthogonal codes and Code Division Multiplexing.

Description:

Code Division Multiple Access (CDMA) is a method of multiplexing that does not divide a channel by time as in TDMA or frequency as in FDMA. Instead all active users use the same frequency at the same time. Separation of channels is now achieved by code . This scheme encodes data using special code associated with each channel called chipping sequence (or Pseudo random Noise sequence). The codes used here are orthogonal and has good auto-correlation property.

CDMA multiples the data being transmitted by a "noise" signal (chipping sequence). This noise signal is a pseudo random sequence of 1 and −1 values, at a frequency much higher than that of the original signal, thereby spreading the energy of the original signal into a much wider band.

De spreading requires the receiver to apply the same PN sequence on the received signal to recover data.

<u>ORTHOGONAL CODES:</u>

Two vectors are said to be orthogonal if their inner product is zero. Consider two vectors (2,0,3) & (3,5,-2). Their inner product is (2*3)+(0*5)+(3*-2)=6+0+(-6)=0. Hence they are orthogonal.
Two codes are orthogonal if following equation is satisfied.

Where 'n' is the length if code.
Ex.
Code 1 : 0 1 0 1
Code 2 : 0 1 1 0
Bipolar Code1 : -1 1 -1 1
Bipolar Code2 : -1 1 1 -1

Code1 * Code 2 : 1 1 -1 -1

Sum = (1+1-1-1) = 0
Hence two codes are orthogonal.

WALSH CODES:

- Walsh codes are also known as Walsh Hardmard Codes.
- The Walsh code is a linear code, which maps binary strings of length n to binary codewords of length 2n. Further these codes are mutually orthogonal.
- Walsh codes are most commonly used orthogonal codes in CDMA application.
- Length –> power of 2 (1,2,4,8---)
- Walsh codes are used for spreading in the forward link.

Generation of the Walsh code matrices

$$W_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$W_{2^n} = \begin{bmatrix} W_{2^{n-1}} & W_{2^{n-1}} \\ W_{2^{n-1}} & \overline{W}_{2^{n-1}} \end{bmatrix}$$

Example of WC sequence generation:

$$W_4 = \begin{bmatrix} W_2 & W_2 \\ W_2 & \overline{W}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$W_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- Code is given as a row in WC matrix
- To generate a code
  - “0” -> “1”
  - “1” -> “-1”
- Example: Codes $W_{4,2}$ and $W_{4,3}$
  - $W_{8,2}$: (0,0,1,1,0,0,1,1) -> (1,1,-1,-1,1,1,-1,-1)
  - $W_{8,3}$: (0,1,1,0,0,1,1,0) -> (1,-1,-1,1,1,-1,-1,1)

- When synchronized – codes are orthogonal

$$W_{8,2} \cdot W_{8,3} = (1,1,-1,-1,1,1,-1,-1) \cdot (1,-1,-1,1,1,-1,-1,1) = 0$$

- When out of sync – codes are not orthogonal

$$W_{8,2} \cdot \text{shift}\,(W_{8,3},1) = (1,1,-1,-1,1,1,-1,-1) \cdot (1,1,-1,-1,1,1,-1,-1) = 8$$

CDMA Example:

1) Sender A's data Ad = 1 => Bipolar Ad = +1

   Sender B's data Bd = 0 => Bipolar Bd = -1

2) A's Chip code is codeA[ ] : 0 0 1 1 0 0 1 1 =>

   Bipolar conversion is : -1 -1 +1 +1 -1-1 +1 +1

   B's Chip code is codeB[]: 0 1 1 0 0 1 1 0

Bipolar conversion is : -1 +1 +1 -1 -1 +1 +1 -1

3) Spread A's data

As =1 * (-1 -1 +1 +1 -1 -1 +1 +1) = (-1 -1 +1 +1 -1 -1 +1 +1)

Spread B's data

Bs = -1 * (-1 +1 +1 -1 -1 +1 +1 -1) = (+1 -1 -1 +1 +1 -1 -1 +1)

4) Send the sum of As+Bs

Cs = As + Bs = (0 -2 0 2 0 -2 0 2)

5) Recover As Data from received signal Cs

Cs * codeA[ ] = (0 -2 0 2 0 -2 0 2) * (-1 -1 +1 +1 -1-1 +1 +1)

= (0 2 0 2 0 2 0 2)

Sum = 8 > 0 hence A's transmitted data was Ad=1

6) Recover B's Data from received signal Cs

Cs * codeB[ ] = (0 -2 0 2 0 -2 0 2) * (-1 +1 +1 -1 -1 +1 +1 -1)

= (0 -2 0 -2 0 -2 0 -2)

Sum = -8 < 0 hence B's transmitted data was Bd = 0

Algorithm:

1) Start
2) Enter sender A's data: Ad, Convert into bipolar
3) Enter sender B's data: Bd, Convert into bipolar
4) Enter A's PN sequence: codeA [] and Convert into bipolar
5) Enter B's PN sequence: codeB [] and Convert into bipolar
6) Spread A's data: As [] = Ad * codeA []
7) Spread B's data: Bs [] = Bd * codeB []
8) Add As [] and B []: c [] = As [] + Bs []
9) De spread A's signals
   ResultA [] = c [] * codeA []
   Add values of ResultA[ ]
   If sum > 0 then A's transmitted data is 1 else 0.
10) De spread B's signals
   ResultB [] = c [] * codeB[]
   Add values of ResultB []
   If sum > 0 then A's transmitted data is 1 else 0.
11) Stop.

Code:

```
#!/usr/bin/env python
# coding: utf-8

def buildWalsh( walsh , l , i1 , i2 , j1 , j2 , bar ):
```

```
    if l == 2:

        one = -1 if bar else 1

        walsh[i1][j1] = one
        walsh[i1][j2] = one
        walsh[i2][j1] = one
        walsh[i2][j2] = -one

        return


    midi = (i1+i2)//2
    midj = (j1+j2)//2

    nl = l//2

    buildWalsh( walsh , nl, i1, midi, j1, midj, bar)
    buildWalsh( walsh , nl, i1, midi, midj + 1, j2, bar)
    buildWalsh( walsh , nl, midi + 1, i2, j1, midj, bar)
    buildWalsh( walsh , nl, midi + 1, i2, midj + 1, j2, not bar)


def showWalsh(walsh):
    print("---------")
    print(" Walsh Table Output : ")
    for w in walsh:
        print(*w)
    print("----------")


def setup( walsh , chnlSeq ,  data ):

    for i , d in enumerate( data ):
        w = dotMul( walsh[i] , d)
        chnlSeq = addMat( chnlSeq  , w )

    return chnlSeq


def listenTo( walsh , chnlSeq , source ):

    print(" Listening to the Channel ", source)
    inner = sum(innerProd( walsh[source-1] , chnlSeq ))
    print(" Data recieved is using Walsh Matrix is : " , inner//stations)
```

```
def addMat(am , bm):
    nm = []
    for a , b in zip(am , bm):
        nm.append(a+b)
    return nm

def dotMul(am , c):
    nm = []
    for a in am:
        nm.append( a*c )
    return nm

def innerProd(am , bm):
    nm = []
    for a , b in zip(am , bm):
        nm.append(a*b)
    return nm

print(" Enter the number of stations : ")
stations = int(input())
walsh = [ [0 for _ in range (stations) ] for __ in range(stations) ]
chnlSeq =  [0 for _ in range (stations) ]
data = []
for i in range(4):
    print(" Enter Data for Station : ",i+1)
    d = int(input())
    data.append(d)
buildWalsh( walsh, stations,  0 , stations-1 , 0 , stations-1 , False)
showWalsh(walsh)
chnlSeq = setup( walsh , chnlSeq , data )
print(" The Resultant Channel Sequence from Walsh Matrix : ",chnlSeq)
print(" Enter the channel number who's data you wish to receive : ")
n = int(input())
listenTo( walsh , chnlSeq , n )
```

Output:

```
'''
(rasa) C:\Users\ASUS\Desktop>python mcc3.py
 Enter the number of stations:
4
 Enter Data for Station:  1
1
 Enter Data for Station:  2
```

```
-1
 Enter Data for Station:  3
0
 Enter Data for Station:  4
1
---------
 Walsh Table Output:
1 1 1 1
1 -1 1 -1
1 1 -1 -1
1 -1 -1 1
----------

 The Resultant Channel Sequence from Walsh Matrix: [1, 1, -1, 3]
 Enter the channel number who's data you wish to receive:
2
 Listening to the Channel 2
 Data received is using Walsh Matrix is:  -1

(rasa) C:\Users\ASUS\Desktop>
'''
```

Conclusion: CDMA has been studied.

PostLab assignment:

1. **What are advantages of CDMA technology over GSM Technology**

MCC POSTLAB

Q1. Ans:- * Technology —

(i) The CDMA is based on spread spectrum technology which makes the optimal use of available bandwidth. It allows each user to transfer over the entire frequency spectrum all the time.
(ii) The GSM operates on the wedge called a carrier. This carrier is divided into a number of time slots and each user is assigned a different time slot so that until the ongoing call is finished, no other subscriber can have access to this.

* Security —

(i) In CDMA technology, more security is provided as compared with the GSM technology because encryption is inbuilt in the CDMA.
(ii) A unique code is provided to each and every user and all the conversations between two users are encoded ensuring a greater level of security of CDMA users.
(iii) The signal cannot be traced easily in CDMA as compared to the signals of GSM which are concentrated in the narrow bandwidth.
(iv) ∴ CDMA phone calls are more secure than GSM calls. In terms of encryption, GSM technology has to be upgraded so as to make it operate more securely.

* Radiation Exposure —

(i) The GSM phones emit continuous wave pulses, so there is a large need to reduce the exposure to electromagnetic fields focused on cell phones "with continuous wave pulses".
(ii) The CDMA cell phones do not produce these pulses. GSM phones emit about 28 times more radiation on average as compared to CDMA phones. Moreover, GSM phones are more biologically reactive as compared to CDMA.

2. **Compare FDM, TDM, CDM, and SDM.**

Q2) Ans:-

SDM	FDM	TDM	CDM
· premise that if we have entities wishing to communicate using a single channel, then as long as we space them far enough apart interference will not occur	· divides the available frequency into non-overlapping bands with guard spaces between to avoid overlapping (adjacent channel interference)	· allows access to entire frequency bandwidth but for a limited amount of time	· All channels use the same frequency, however each channel is given its own unique code.
· To reduce further, the risk of interference place guard space between the frequency spaces.	· receiver only has to know the frequency to tune in to	· all senders are same frequency in at different time	· each node must be sufficiently orthogonal to allow appropriate guard space.
· SDM is faster than TDM	· used in analogue system.	· if two transmissions overlap known as co-channel interference	· large range of codes provides significant expansion security
· it require one physical connectn per communicatn pair	· channel allocated even if no data	· precise clock synchronizatn required	· Highly complex scheme
· network infrastructure cost is high		· time slots allocated even if no data	· Receiver has to know the code & be able to separate and