Experiment No.2: CDMA

PART A

(PART A: TO BE REFFERED BY STUDENTS)

A.1 Aim: **To implement a basic function of Code Division Multiple Access (CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation. Write an application based on the above concept.**

A.2 Prerequisite: Knowledge of multiplexing schemes

A.3 Objectives: To understand the importance of security in wirelesss communication by means of spread spectrum technologies.

A.4 Outcomes: Student will be able to articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it.(LO-2)

A.5 Tools Used/programming language: Java

A.6 Theory:

- **Code-division multiple access (CDMA)** is a channel access method **used by various radio communication technologies**. CDMA is an example of multiple accesses, where several transmitters can send information simultaneously over a single communication channel. This allows several users to share a band of frequencies (see bandwidth). To permit this without undue interference between the users, CDMA employs spread spectrum technology and a special codling's scheme (where each transmitter is assigned a code).

CDMA issued as the access method in many mobile phone standards. IS-95,also called" cdma One", and its 3G evolution CDMA2000, are often simply referred to as "CDMA", but UMTS, the 3G standard used by GSM carriers, also uses "wideband CDMA", or W-CDMA, as well as TDCDMA and TD-SCDMA, as its radio technologies.

- **CDMA Orthogonality**:

Techniques generally used are direct sequence spread spectrum modulation (DS-CDMA), frequency hopping or mixed CDMA detection (JDCDMA). Here, a signal is generated which extends over a wide bandwidth. A code called spreading code is used to perform this action. Using a group of codes, which are orthogonal to each other, it is possible to select a signal with a given code in the presence of many other signals with different orthogonal codes.

- **CDMA Autocorrelation:**

Autocorrelation of the sequence, it determines the ability to synchronize and lock the spreading code for the received signal.

A.7 Procedure:

- ✓ The station encodes its data bit as follows.
  - If bit = 1 then +1
  - If bit = 0 then -1
  - no signal(interpreted as 0) if station is idle
- ✓ Each station is allocated a different orthogonal sequence (code) which is N bit long for N stations

- ✓ Each station does a scalar multiplication of its encoded data bit and code sequence.
- ✓ The resulting sequence is then stored on the channel.
- ✓ Since the channel is common, amplitudes add up and hence resultant channel sequence is the sum of sequences from all channels.

- ✓ If station 1 wants to listen to station 2, it multiplies (inner product) the channel sequence with code of station S2
- ✓ The inner product is then divided by N to get data bit transmitted from station 2.

- **Working**

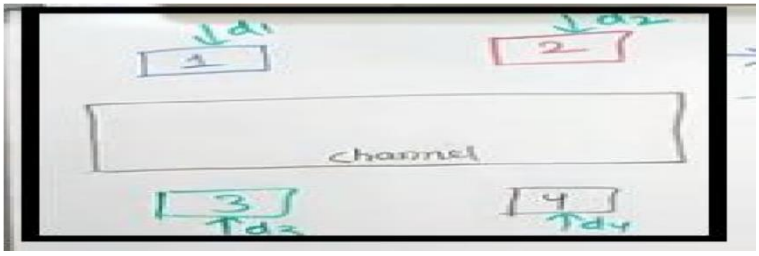
- ✓ CDMA uses orthogonal codes to transmit different signal over the same channel
- ✓ CDMA is use in 3'rd generation wireless communication like CDMA 2000, w-CDMA,HSDPA (high speed downlink packet access), HSUPA(high speed uplink packet access) ✓

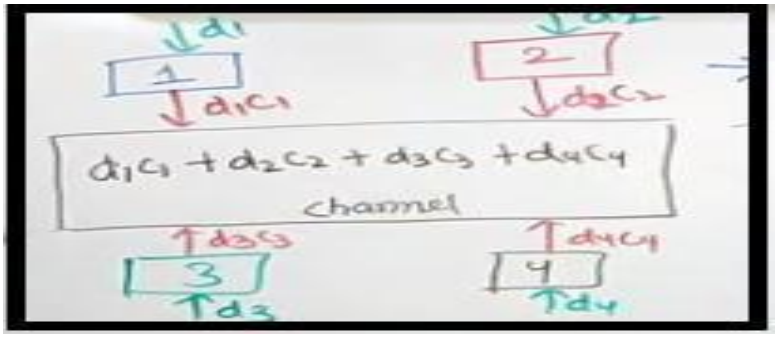
CDMA stands for Code Division Multiple Access.

- ✓ It is a digital cellular standard that utilizes spread-Spectrum Technology.
- ✓ It spreads the signal over a fully available spectrum or over multiple channels through division.

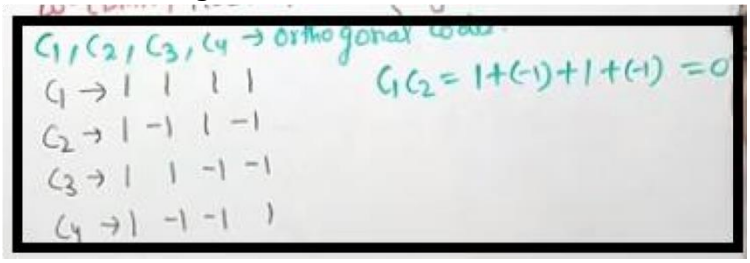
- ✓ It is a more secure and private line.
- ✓ It has good voice and data communication capabilities.
- ✓ The information is sent simultaneously through several transmitters over a single communication channel.

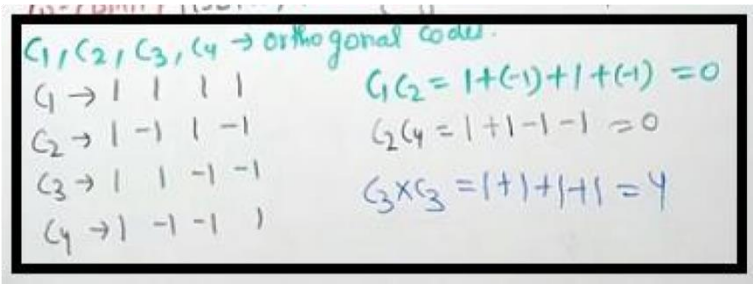
- ✓ **Consider there is single channel having four users, user1,user2,user3,user4**





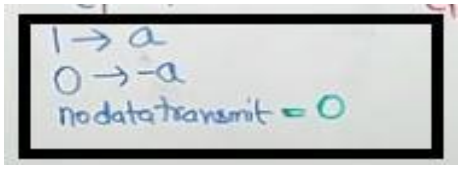
- ✓ Assume there are four orthogonal codes



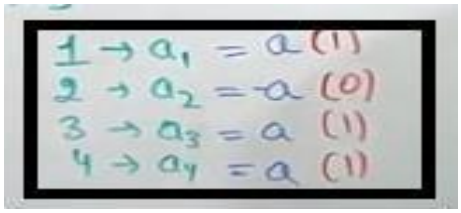


- ✓ If two diff orthogonal codes are multiplied it will always give 0; this is the property of orthogonal code.

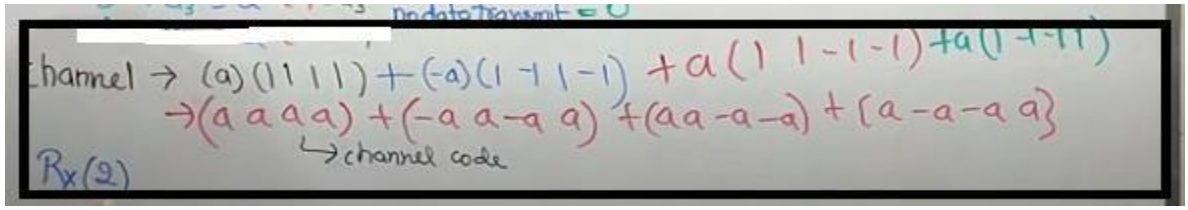
- ✓ User send data 1,0 (stream of data)or use may be silent that is no data to transmit.



✓ Users are having following data

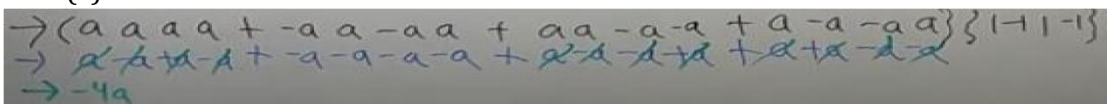


✓ Data in channel is;



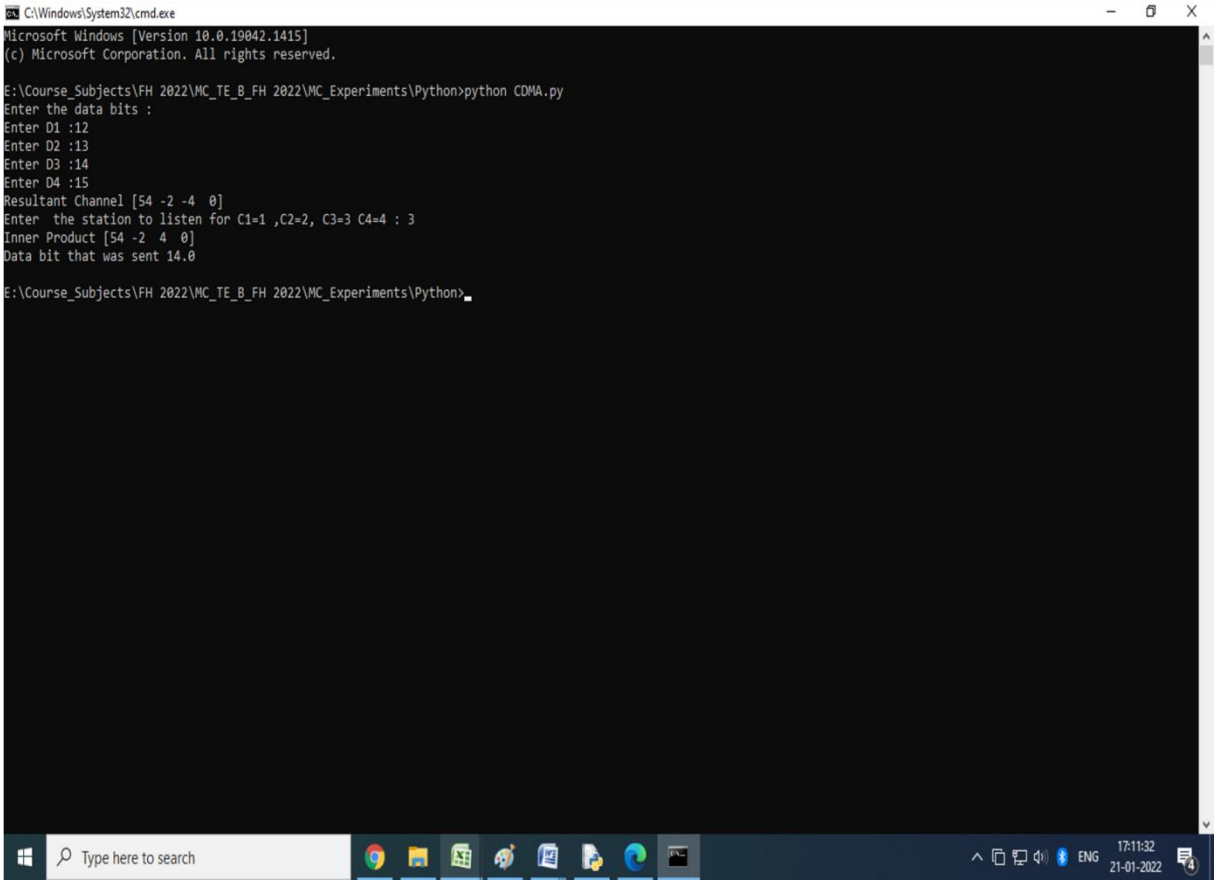
✓ If data 2 want to receive then

R2= C(x)*C2



And hence we are getting –a after dividing by 4 and data send by user 2 is -a

Sample Output:


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.
E:\Course_Subjects\FH 2022\MC_TE_B_FH 2022\MC_Experiments\Python>python CDMA.py
Enter the data bits :
Enter D1 :12
Enter D2 :13
Enter D3 :14
Enter D4 :15
Resultant Channel [54 -2 -4 0]
Enter the station to listen for C1=1 ,C2=2, C3=3 C4=4 : 3
Inner Product [54 -2 4 0]
Data bit that was sent 14.0
E:\Course_Subjects\FH 2022\MC_TE_B_FH 2022\MC_Experiments\Python>
Type here to search
ENG
17:11:32
21-01-2022

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)

Roll No. B30	Name: Pranjal Bhatt
Class : TE COMPS B	Batch :B2
Date of Experiment:	Date of Submission
Grade :	

B.1 Question of Curiosity:

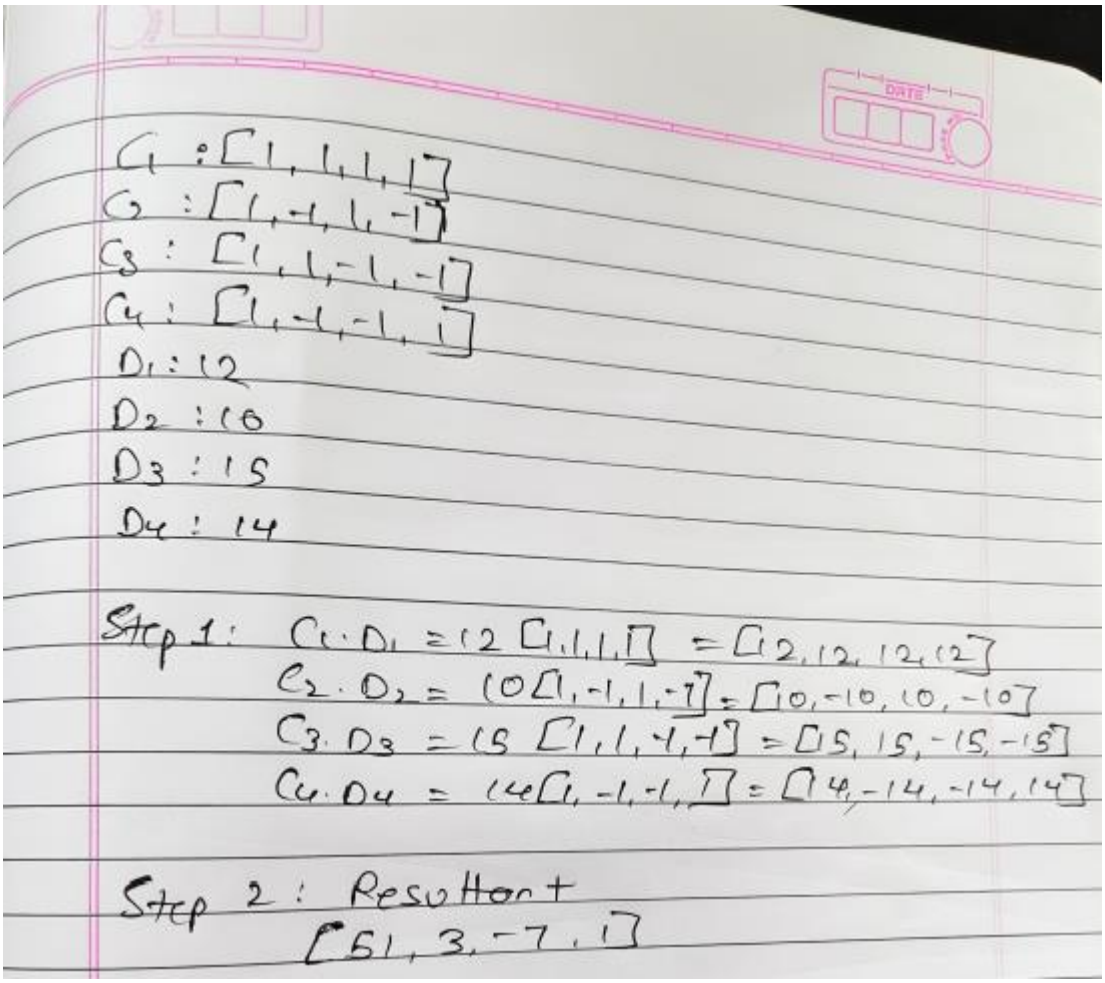
Q.1: Source Code (students need to implement CDMA using any programming language like Java, Python , etc)

```
import numpy as np

c1 = [1, 1, 1, 1]
c2 = [1, -1, 1, -1]
c3 = [1, 1, -1, -1]
c4 = [1, -1, -1, 1]
rc = []

print("Enter the data bits:")
d1 = int(input("Enter D1: "))
d2 = int(input("Enter D2: "))
d3 = int(input("Enter D3: "))
d4 = int(input("Enter D4: "))

r1 = np.multiply(c1, d1)
r2 = np.multiply(c2, d2)
```

```
r3 = np.multiply(c3, d3)
r4 = np.multiply(c4, d4)

resultant_channel = r1 + r2 + r3 + r4
print("Resultant Channel:", resultant_channel)

Channel = int(input("Enter the station to listen for (C1=1, C2=2, C3=3, C4=4): "))
if Channel == 1:
   rc = c1
elif Channel == 2:
   rc = c2
elif Channel == 3:
   rc = c3
elif Channel == 4:
   rc = c4

inner_product = np.multiply(resultant_channel, rc)
print("Inner Product:", inner_product)

res1 = sum(inner_product)
data = res1 / len(inner_product)
print("Data bit that was sent:", data)
```

Q.2: Output of CDMA

```
PS C:\Users\HP\Documents\Experiment\pranjal> python .\mcexp2.py
Enter the data bits:
Enter D1: 112
Enter D2: 10
Enter D3: 15
Enter D4: 14
Resultant Channel: [151 103  93 101]
Enter the station to listen for (C1=1, C2=2, C3=3, C4=4): 3
Inner Product: [ 151  103  -93 -101]
Data bit that was sent: 15.0
PS C:\Users\HP\Documents\Experiment\pranjal>
```



Q.3: Explain CDMA with one suitable example.

CDMA (Code Division Multiple Access) is a communication technique that allows multiple signals to occupy a single transmission channel, optimizing the use of available bandwidth. Each communication is assigned a unique code, allowing multiple users to transmit over the same frequency without interfering with each other.

How CDMA works:

1. **Unique Codes**: Each user is assigned a unique code, often called a **spread spectrum code**. This code is used to "spread" the signal over a wide frequency band.
2. **Signal Transmission**: When users transmit their data, the information is multiplied by their unique code. This results in a signal that is spread out across the spectrum.
3. **Reception**: At the receiver end, the signal is decoded using the same unique code, which extracts the original information from the spread signal.

Since each user's signal is spread differently using their unique code, multiple users can share the same frequency channel without interfering with each other.

Example of CDMA:

Imagine a situation in a city where many people are trying to make calls using the same frequency channel. Without CDMA, all signals would overlap, causing interference. However, with CDMA, each user is assigned a unique code.

For instance:

- **User A** is assigned the code "101010"
- **User B** is assigned the code "110011"
- **User C** is assigned the code "111000"

Now, when each user transmits their signal, it is multiplied by their unique code. So, even though all users are transmitting at the same time over the same frequency, their signals don't interfere with each other because each signal is spread differently due to the unique codes.

At the receiver end, the signals are decoded by applying the same codes used for transmission, which separates the signals of different users and ensures they are correctly received.

Q.4: What is spread spectrum? List types of spread spectrum technologies.

Spread Spectrum is a technique used in communication systems to spread the signal over a wider frequency band than the minimum required bandwidth. This is done to improve the performance and security of the communication system. The spread spectrum method helps reduce interference, improve signal-to-noise ratio (SNR), and make the system more resistant to jamming and eavesdropping.

Types of Spread Spectrum Technologies:

1. **Frequency Hopping Spread Spectrum (FHSS)**:

    - **How it works**: The transmitter rapidly changes its carrier frequency in a sequence known to both the transmitter and receiver. This hopping occurs over a wide band of frequencies.
    - **Example**: Bluetooth devices use FHSS to minimize interference and improve security by constantly switching frequencies within a set range.

1. **Direct Sequence Spread Spectrum (DSSS)**:

- **How it works**: The data signal is multiplied by a higher-frequency pseudo-random noise (PN) code, spreading the signal over a wider frequency band. Both the transmitter and receiver must use the same code to decode the signal.
- **Example**: Wi-Fi (802.11b/g) uses DSSS to transmit data over a wide frequency spectrum, which reduces the likelihood of interference and increases the reliability of the connection.

Q.5: Differentiate between FDM and FHSS.

Feature	Frequency Division Multiplexing (FDM)	Frequency Hopping Spread Spectrum (FHSS)
Definition	A multiplexing technique that divides the available bandwidth into multiple frequency channels for simultaneous transmission.	A spread spectrum technique that rapidly switches (hops) between different frequencies during transmission.
Primary Purpose	Used to transmit multiple signals simultaneously over different frequency bands.	Used to enhance security and resistance to interference by spreading the signal over multiple frequencies.
Bandwidth Usage	Fixed frequency allocation per channel.	Uses a wide range of frequencies but transmits at a single frequency at any given time.
Interference Handling	Less resistant to interference; affected by narrowband interference.	Highly resistant to interference and jamming due to frequency hopping.
Security	Less secure as frequencies remain fixed.	More secure as hopping makes eavesdropping difficult.
Synchronization	Requires precise frequency allocation and synchronization.	Requires synchronization for hopping sequence but is more robust against signal interception.

Feature	Frequency Division Multiplexing (FDM)	Frequency Hopping Spread Spectrum (FHSS)
Example Applications	FM radio, television broadcasting, cable TV, and telephone networks.	Bluetooth, military communications, and some wireless LANs.

B.2 Conclusion:

This experiment demonstrates the concept of **Code Division Multiplexing (CDM)** using orthogonal codes for data transmission and retrieval. By assigning unique orthogonal codes to different data streams, multiple signals are transmitted simultaneously over a shared channel without interference.

The **resultant channel** is obtained by summing the encoded signals, and the original data bit is retrieved by performing an inner product operation with the corresponding code. This method ensures efficient multiplexing and accurate data recovery, provided that the codes are truly orthogonal.

Such a technique is fundamental in **spread spectrum communication**, particularly in applications like **CDMA (Code Division Multiple Access)**, where multiple users can share the same frequency bandwidth without causing interference.