**EXPERIMENT – 1**

Simulate to elaborate operation of multiple access techniques for CDMA.

Student Name: Roll No:

Date of Practical Performed: Date of Submission:

Faculty Signature: Grade:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AIM**: Simulate to elaborate operation of multiple access techniques for CDMA.

**SOFTWARE REQUIREMENT:**

Windows / linux based System,

Python

**THEORY:**

In modern communication systems, multiple techniques are employed to enable multiple users to share the same communication channel. Code Division Multiple Access (CDMA) is a widely used technique that allows multiple users to access the channel simultaneously by using unique spreading codes assigned to each user.

Spreading Codes: In CDMA, each user is assigned a unique spreading code that spreads the narrowband signal to be transmitted. These spreading codes are carefully designed to be orthogonal to each other, ensuring that signals from different users do not interfere with each other.

At the receiver, the transmitted signal from each user is multiplied by the assigned spreading code to extract the user's signal from the mixed signal received. One of the challenges in CDMA is the interference from signals of other users, where the presence of overlap of multiple signals can lead to interference at the receiver. Due to the orthogonality of spreading codes, interference can be mitigated.

**CODE:**

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)

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**PROCEDURE**

1. Open your Python IDE.
2. Take multiple user inputs for data bits.
3. Write a section to merge channels.
4. Implement the required inner product operation to extract data.
5. Run the Python script.
6. Note the observations.

**CONCLUSION**

Simulating multiple users and studying CDMA operation helps in understanding how nodes enable simultaneous transmission and reception of data, integrating communication seamlessly.