PRACTICAL   
  
CODE :

import numpy as np

# CDMA codes

c1=[1,1,1,1]

c2=[1,-1,1,-1]

c3=[1,1,-1,-1]

c4=[1,-1,-1,1]

print("CDMA Implementation")

print("Codes:")

print("C1=",c1)

print("C2=",c2)

print("C3=",c3)

print("C4=",c4)

# Check orthogonality

print("\nOrthogonality Test:")

print("C1 x C2 =",np.dot(c1,c2))

print("C1 x C3 =",np.dot(c1,c3))

print("C1 x C4 =",np.dot(c1,c4))

print("C2 x C3 =",np.dot(c2,c3))

print("C2 x C4 =",np.dot(c2,c4))

print("C3 x C4 =",np.dot(c3,c4))

# Check if orthogonal

orthogonal\_results = [np.dot(c1,c2), np.dot(c1,c3), np.dot(c1,c4), np.dot(c2,c3), np.dot(c2,c4), np.dot(c3,c4)]

if all(x == 0 for x in orthogonal\_results):

print("Result: Codes are ORTHOGONAL")

else:

print("Result: Codes are NOT ORTHOGONAL")

# Check autocorrelation

print("\nAutocorrelation Test:")

print("C1 x C1 =",np.dot(c1,c1))

print("C2 x C2 =",np.dot(c2,c2))

print("C3 x C3 =",np.dot(c3,c3))

print("C4 x C4 =",np.dot(c4,c4))

# Get data bits

print("\nEnter data bits:")

d1=int(input("D1: "))

d2=int(input("D2: "))

d3=int(input("D3: "))

d4=int(input("D4: "))

# Encoding

r1=np.multiply(c1,d1)

r2=np.multiply(c2,d2)

r3=np.multiply(c3,d3)

r4=np.multiply(c4,d4)

print("\nEncoded signals:")

print("S1=",r1)

print("S2=",r2)

print("S3=",r3)

print("S4=",r4)

# Channel signal

channel=r1+r2+r3+r4

print("Channel signal:",channel)

# Decoding

station=int(input("\nEnter station to decode (1-4): "))

if station==1:

code=c1

elif station==2:

code=c2

elif station==3:

code=c3

elif station==4:

code=c4

inner\_product=np.multiply(channel,code)

print("Inner product:",inner\_product)

result=sum(inner\_product)/4

print("Decoded data:",result)  
  
  
(base) kjcoemr@kjcoemr-HP-Pro-SFF-280-G9-Desktop-PC:~/Desktop/BE-15$ python practical2.py

CDMA Implementation

Codes:

C1= [1, 1, 1, 1]

C2= [1, -1, 1, -1]

C3= [1, 1, -1, -1]

C4= [1, -1, -1, 1]

Orthogonality Test:

C1 x C2 = 0

C1 x C3 = 0

C1 x C4 = 0

C2 x C3 = 0

C2 x C4 = 0

C3 x C4 = 0

Result: Codes are ORTHOGONAL

Autocorrelation Test:

C1 x C1 = 4

C2 x C2 = 4

C3 x C3 = 4

C4 x C4 = 4

Enter data bits:

D1: 1

D2: -1

D3: 1

D4: -1

Encoded signals:

S1= [1, 1, 1, 1]

S2= [1, -1, 1, -1]

S3= [1, 1, -1, -1]

S4= [-1, 1, 1, -1]

Channel signal: [2, 2, 2, -2]

Enter station to decode (1-4): 2

Inner product: [2, -2, 2, 2]

Decoded data: 1.0

(base) kjcoemr@kjcoemr-HP-Pro-SFF-280-G9-Desktop-PC:~/Desktop/BE-15$