DEPARTMENT OF COMPUTER ENGINEERING

Semester	T.E. Semester VI – Computer Engineering
Subject	Mobile Computing
Subject Professor In-charge	Prof. Sneha Annappanavar
Assisting Teachers	Prof. Sneha Annappanavar
Laboratory	M310A

Student Name	Deep Salunkhe
Roll Number	21102A0014
TE Division	A

Title:

CDMA

Explanation:

1. **CDMA (Code Division Multiple Access)**: CDMA is a multiple access technique used in wireless communication systems. Unlike other multiple access techniques such as TDMA and FDMA, where users share the same frequency channels by dividing them into time slots or frequency bands, CDMA allows multiple users to transmit simultaneously over the same frequency band using unique spreading codes.

2. **Orthogonal Codes**: In CDMA, orthogonal codes are used to distinguish between different users' signals. Orthogonal codes are sequences of binary digits (1s and -1s) that are mutually orthogonal to each other, meaning their dot product is zero. This property allows the receiver to separate the desired signal from interference caused by other users' signals.

3. **Bipolar Encoding**: Bipolar encoding is a method of representing binary data using bipolar signals, where the binary 1s are represented by positive voltages and the binary 0s are represented by negative voltages. In the context of CDMA, bipolar encoding is used to convert binary data into bipolar signals before transmission.

4. **Signal Multiplication**: In CDMA, the transmitted signal is the result of multiplying the encoded data signal with the spreading code. This process spreads the signal across a wider frequency band, making it more resistant to interference and allowing multiple users to transmit simultaneously.

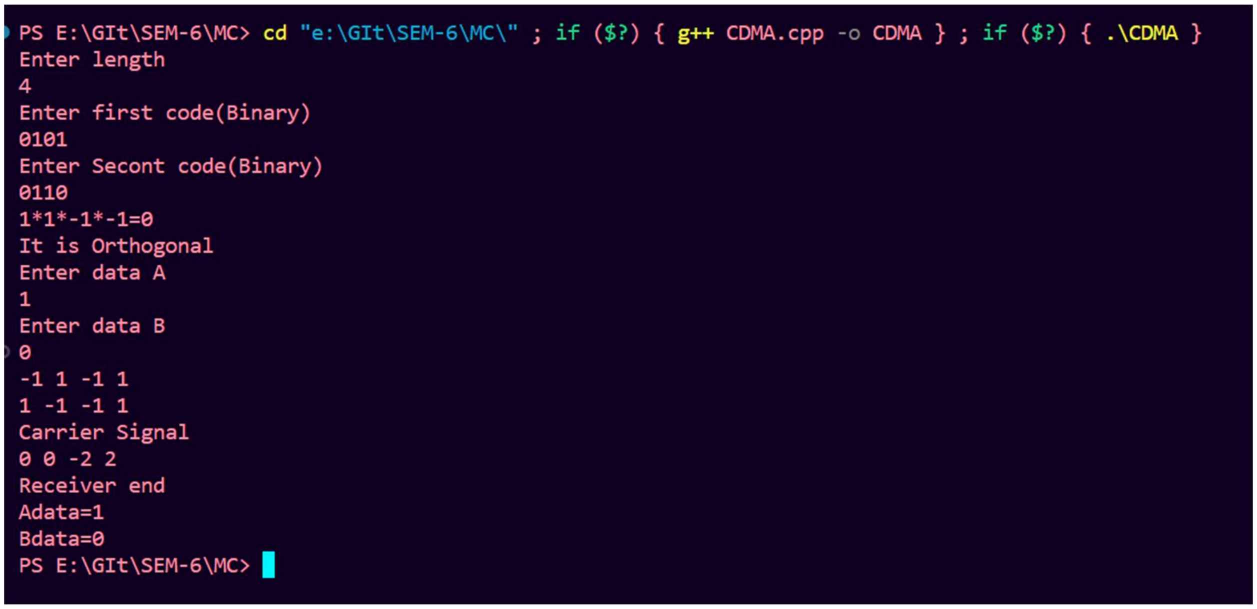
Implementation:

```
#include<iostream>
#include<vector>
using namespace std;

void Bipolar(vector<int>&C,string s){
    int size=s.length();
    for(int i=0;i<size;i++){
        if(s[i]=='1'){
            C.push_back(1);
        }else{
            C.push_back(-1);
        }
    }
}

void    Multiply(vector<int>A,vector<int>B,vector<int>&M){
    int size=A.size();
    for(int i=0;i<size;i++){
        M.push_back(A[i]*B[i]);
    }
}

void TakeCode(string &s,int size){

    int originalsize=size;

    cin>>s;
    if(s.length()!=size){
            cout<<"Invalid size "<<endl;
            return;
    }
    for(int i=0;i<size;i++){
        if(s[i]=='1'|| s[i]=='0'){
            continue;
     }else{
        cout<<"Not a Binary"<<endl;
        return;
     }

    }

}
```

```
void printvector(vector<int> vec){

    int n=vec.size();
    for(int i=0;i<n;i++){
        cout<<vec[i]<<" ";
    }
    cout<<endl;

}

int  getorthogonals(vector<int>&AC,vector<int>&BC){
    int len=0;
    cout<<"Enter length"<<endl;
    cin>>len;
    string A,B;
    cout<<"Enter first code(Binary)"<<endl;
    TakeCode(A,len);
    cout<<"Enter Secont code(Binary)"<<endl;
    TakeCode(B,len);
    vector<int>M;
    Bipolar(AC,A);
    Bipolar(BC,B);
    Multiply(AC,BC,M);
    int temp=0;
    int size=A.size();
    for(int i=0;i<size;i++){
    temp=temp+M[i];
    }

    for(int i=0;i<len;i++){
        cout<<M[i];
        if(i!=len-1) cout<<"*";
    }
    cout<<"="<<temp<<endl;

    if(temp==0){
        cout<<"It is Orthogonal"<<endl;


    }else
    {
            cout<<"It is Not Orthogonal"<<endl;
    }
    return 0;
}

void TakeData(vector<int>&A,vector<int>&B){

    string Bd,Ad;

    cout<<"Enter data A"<<endl;

    cin>>Ad;
    int la=Ad.length();
    for(int i=0;i<la;i++){
        if(Ad[i]=='1')
        A.push_back(1);
        else{
            A.push_back(-1);
        }
    }

    cout<<"Enter data B"<<endl;

    cin>>Bd;
    int lb=Bd.length();
    for(int i=0;i<lb;i++){
        if(Bd[i]=='1')
        B.push_back(1);
        else{
            B.push_back(-1);
        }
    }


}

void cal_data_signal(vector<int>&data,vector<int>&code,vector<int>&signal){

    int n=code.size();
    for(int i=0;i<n;i++){
        signal.push_back(data[0]*code[i]);
    }


}
```

```
void cal_carier_signal(vector<int>Asignal,vector<int>Bsignal,vector<int>&Csig-
nal){
    int n=Asignal.size();
    for(int i=0;i<n;i++){

        Csignal.push_back(Asignal[i]+Bsignal[i]);
    }
}

void receiver(vector<int>Csignal,vector<int>AK,vector<int>BK){

    vector<int> Adata,Bdata;
    int nak=AK.size();
    int nbk=BK.size();

    //filling Adata;
    for(int i=0;i<nak;i++){
        int temp=0;
        temp=Csignal[i]*AK[i];
        Adata.push_back(temp);
    }

    //filling Bdata;
    for(int i=0;i<nbk;i++){
        int temp=0;
        temp=Csignal[i]*BK[i];
        Bdata.push_back(temp);
    }

    int sum_of_Adata=0;
    int sum_of_Bdata=0;

    for(int i=0;i<nak;i++){
        sum_of_Adata=sum_of_Adata+Adata[i];
    }

    for(int i=0;i<nbk;i++){
        sum_of_Bdata=sum_of_Bdata+Bdata[i];
    }

    if(sum_of_Adata>0){
        cout<<"Adata=1"<<endl;
    }else{
        cout<<"Adata=0"<<endl;
    }

    if(sum_of_Bdata>0){
        cout<<"Bdata=1"<<endl;
    }else{
        cout<<"Bdata=0"<<endl;
    }

}

int main(){
    //currently this code works only for the values of data is 1bit and the code
is 4 bit
    vector<int > AK,BK;
    getorthogonals(AK,BK);
    vector<int>Adata,Bdata;
    TakeData(Adata,Bdata);
    vector<int>Asignal,Bsignal,Csignal;
    cal_data_signal(Adata,AK,Asignal);
    cal_data_signal(Bdata,BK,Bsignal);
    printvector(Asignal);
    printvector(Bsignal);
    cal_carier_signal(Asignal,Bsignal,Csignal);
    cout<<"Carrier Signal"<<endl;
    printvector(Csignal);
    cout<<"Receiver end"<<endl;
    receiver(Csignal,AK,BK);
    return 0;
}
```

End Result:



```
PS E:\GIt\SEM-6\MC> cd "e:\GIt\SEM-6\MC\" ; if ($?) { g++ CDMA.cpp -o CDMA } ; if ($?) { .\CDMA }
Enter length
4
Enter first code(Binary)
0101
Enter Secont code(Binary)
0110
1*1*-1*-1=0
It is Orthogonal
Enter data A
1
Enter data B
0
-1 1 -1 1
1 -1 -1 1
Carrier Signal
0 0 -2 2
Receiver end
Adata=1
Bdata=0
PS E:\GIt\SEM-6\MC>
```

Conclusion:

In this lab experiment, we explored the principles of CDMA communication. We implemented a simple CDMA transmitter and receiver system using C++ programming. The transmitter encodes binary data using bipolar encoding and spreads the signal using orthogonal codes. The resulting signal is then transmitted over a shared communication channel. At the receiver end, the received signal is despread using the same orthogonal codes used at the transmitter. The despread signal is then decoded to recover the original binary data. Through this experiment, we gained practical insights into the operation of CDMA systems and learned about the importance of orthogonal codes in enabling multiple access in wireless communication networks. Additionally, we observed the advantages of CDMA, such as increased capacity and improved resistance to interference, compared to other multiple access techniques.