**CSL702 - Mobile App. Development Tech. Lab Experiment 04**

Name: Hayden Cordeiro Roll No: 05 Class: BE COMPS

**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.

**CODE****:**

import java.util.\*;

public class CDMACODE {

    private int[][] wtable;

    private int[][] copy;

    private int[] channel\_sequence;

    public void setUp(int[] data, int num\_stations)

    {

        wtable = new int[num\_stations][num\_stations];

        copy = new int[num\_stations][num\_stations];

        buildWalshTable(num\_stations, 0, num\_stations - 1, 0,

                                        num\_stations - 1, false);

        showWalshTable(num\_stations);

        for (int i = 0; i < num\_stations; i++) {

            for (int j = 0; j < num\_stations; j++) {

                // Making a copy of walsh table

                // to be used later

                copy[i][j] = wtable[i][j];

                // each row in table is code for one station.

                // So we multiply each row with station data

                wtable[i][j] \*= data[i];

            }

        }

        channel\_sequence = new int[num\_stations];

        for (int i = 0; i < num\_stations; i++) {

            for (int j = 0; j < num\_stations; j++) {

                // Adding all sequences to get channel sequence

                channel\_sequence[i] += wtable[j][i];

            }

        }

    }

    public void listenTo(int sourceStation, int num\_stations)

    {

        int innerProduct = 0;

        for (int i = 0; i < num\_stations; i++) {

            // multiply channel sequence and source station code

            innerProduct += copy[sourceStation][i] \* channel\_sequence[i];

        }

        System.out.println("The data received is: " +

                            (innerProduct / num\_stations));

    }

    public int buildWalshTable(int len, int i1, int i2, int j1,

                                            int j2, boolean isBar)

    {

        // len = size of matrix. (i1, j1), (i2, j2) are

        // starting and ending indices of wtable.

        // isBar represents whether we want to add simple entry

        // or complement(southeast submatrix) to wtable.

        if (len == 2) {

            if (!isBar) {

                wtable[i1][j1] = 1;

                wtable[i1][j2] = 1;

                wtable[i2][j1] = 1;

                wtable[i2][j2] = -1;

            }

            else {

                wtable[i1][j1] = -1;

                wtable[i1][j2] = -1;

                wtable[i2][j1] = -1;

                wtable[i2][j2] = +1;

            }

            return 0;

        }

        int midi = (i1 + i2) / 2;

        int midj = (j1 + j2) / 2;

        buildWalshTable(len / 2, i1, midi, j1, midj, isBar);

        buildWalshTable(len / 2, i1, midi, midj + 1, j2, isBar);

        buildWalshTable(len / 2, midi + 1, i2, j1, midj, isBar);

        buildWalshTable(len / 2, midi + 1, i2, midj + 1, j2, !isBar);

        return 0;

    }

    public void showWalshTable(int num\_stations)

    {

        System.out.print("\n");

        for (int i = 0; i < num\_stations; i++) {

            for (int j = 0; j < num\_stations; j++) {

                System.out.print(wtable[i][j] + " ");

            }

            System.out.print("\n");

        }

        System.out.println("-------------------------");

        System.out.print("\n");

    }

    // Driver Code

    public static void main(String[] args)

    {

        int num\_stations = 4;

        int[] data = new int[num\_stations];

        //data bits corresponding to each station

        data[0] = -1;

        data[1] = -1;

        data[2] = 0;

        data[3] = 1;

        CDMACODE channel = new CDMACODE();

        channel.setUp(data, num\_stations);

        // station you want to listen to

        int sourceStation = 3;

        channel.listenTo(sourceStation, num\_stations);

    }

}

# Output

****

**Conclusion:** We have successfully implemented a Code Division Multiple Access (CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation.