Problem Statement

Implement CDMA technique for multiple access of a common channel by n stations.

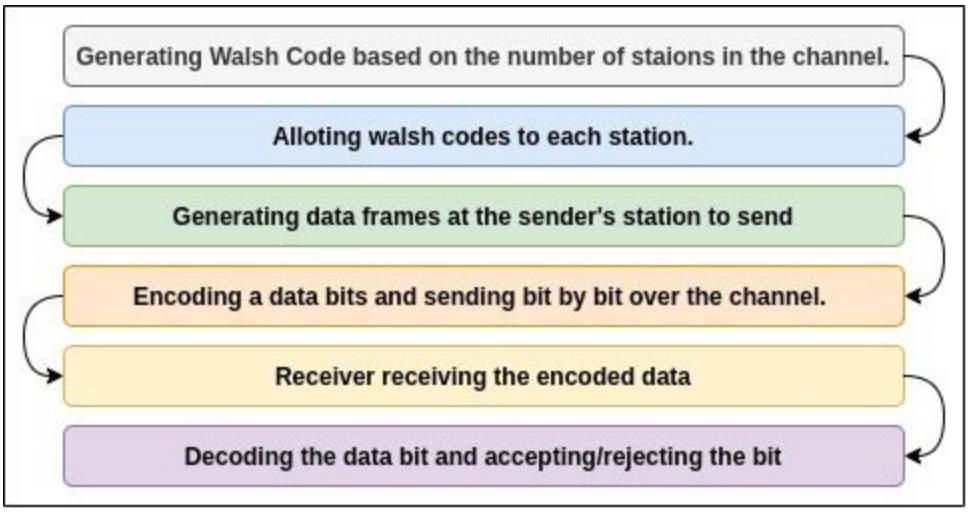
Design description

`Purpose of the program:` The program tries to simulate the environment of data communication at the Media Access Control layer for understanding the inner work flow of Code Division Multiple Access(CDMA) technique. The key points are -

1. Generating frames to send.
2. Encoding per bit using CDMA technique.
3. Sending the data in the channel.
4. Receiving the data at the receiver's end.
5. Decoding the data bit.
6. Accepting or rejecting the bit based on the decoded value.

In the later sections of the report, the complete procedure over various possible test cases is shown for further understanding of method used.

`Structural design:`



1. The raw data is generated randomly (8 bits).
2. Walsh Codes, on the basis of the number of stations, are generated and assigned to each station.
3. Every station sends to a receiver station.
4. The station sends a bit after encoding it based on the CDMA technique.
5. The receiving station receives the data bit.
6. Data bit is decoded at the receiver station and accepted or rejected based on the results.

Input and output format:

1. The input is read from the file “input.txt”.
2. The logs and the output is shown at the terminal

Terminology:

3. SSID - Sender’s station id
4. RSID - Receiver’s station id

Code Snippet

- The channel asks for the number of stations to be involved in the simulation process.
- Walsh codes are generated based on the input.

```
void generate()
    {
        wc = new int[num_of_stations][num_of_stations];
        wc[0][0]=1;
        int prev=1,next=2;
        while(next<=num_of_stations)
        {
            for(int i=prev;i<next;i++)
                for(int j=0;j<prev;j++) wc[i][j]=wc[i-prev][j];

            for(int i=0;i<prev;i++)
                for(int j=prev;j<next;j++)  wc[i][j]=wc[i][j-prev];

            for(int i=prev;i<next;i++)
                for(int j=prev;j<next;j++)
                   wc[i][j]=(-1)*wc[i-prev][j-prev];
            prev=next;
            next*=2;
        }
    }
```

- Each station is assigned a unique chip set from the Walsh codes generated above.
- Random input is generated at each station to send

```
void initialise_data() throws IOException {
        data = new int[n][8];
        ass1.files f = new ass1.files();
        String s = "";
```

```
        for(int i=0;i<n;i++) {
            s+="Station #"+Integer.toString(i)+"\n";
            for(int j=0;j<8;j++) {
                Random rand=new Random();
                double r = rand.nextDouble();
                if(r<0.5) data[i][j]=0;
                else data[i][j]=1;
                s+=Integer.toString(data[i][j]);
            }
            if(i < n-1) s+="\n\n";
        }
        f.writeFile("./ass4/input.txt",s);
    }
```

- Each bit is encoded and sent over the channel. The station may send or not with a probability of 25%.

```
void send() {
        clearSentBits();
        Random rand = new Random();
        System.out.println("\nSSID#\tBit Sent\tSentData\n--------------");
        for(int i=0;i<n;i++) {
            if(lastBit[i]<7) {
                Double r=rand.nextDouble();
                if(r>0.25) { // probability to send
System.out.print("#"+Integer.toString(i)+"\tBit["+Integer.toString(lastBit
[i]+1)+"]: "+Integer.toString(data[i][lastBit[i]+1])+"\t");
                    if(data[i][++lastBit[i]]==0)
                        for(int j=0;j<n;j++) {
                            sentBit[j] += -1*wCode.wc[i][j];
                            System.out.print(-1*wCode.wc[i][j]);
                            System.out.print(" ");
                        }
                    else
                        for(int j=0;j<n;j++) {
                            sentBit[j] += wCode.wc[i][j];
                            System.out.print(wCode.wc[i][j]);
                            System.out.print(" ");
                        }
```

```
                    System.out.println();
                }
            }
            if(lastBit[i]==7) done[i]=true;
        }
        System.out.println();
    }
```

- The encoded data is sent to all the receiving stations.
- The stations decoded the data as (0,1,Reject).

```
void receive() {
        for(int i=0;i<n;i++) {
            int val=0;
            if(lastReceived[i]<8){
                for(int j=0;j<n;j++) {
                    val += wCode.wc[rsid[i]][j]*sentBit[j];
                }
                if(val!=0) {
                    receivedMessage[i][lastReceived[i]++] = ((val/n)+1)/2;
                }
            }
        }
    }
```

- The program outputs the final summary of the data transfer.

```
void printSummary() {
        System.out.println("\nData Transfer Summary");
System.out.println("-------------------------------\nRSID\tSSID\tData\n---
---------------------------");
        for(int i=0;i<n;i++) {
System.out.print("#"+Integer.toString(i)+"\t#"+Integer.toString(rsid[i])+"
\t");
            for(int j=0;j<8;j++) {
                System.out.print(receivedMessage[i][j]);
                System.out.print(" ");
            }
            System.out.println(); } }
```

Output Logs

Number of stations: 4

```
The Walsh Code being used:
1 1 1 1
1 -1 1 -1
1 1 -1 -1
1 -1 -1 1

SSID     RSID     Chip Set
------------------------
#0       #1       1 1 1 1
#1       #2       1 -1 1 -1
#2       #3       1 1 -1 -1
#3       #0       1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[0]: 0         -1 -1 -1 -1
#1       Bit[0]: 1         1 -1 1 -1
#2       Bit[0]: 1         1 1 -1 -1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[1]: 0         -1 -1 -1 -1
#3       Bit[0]: 1         1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[2]: 0         -1 -1 -1 -1
#3       Bit[1]: 1         1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[3]: 0         -1 -1 -1 -1
#3       Bit[2]: 0         -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#1       Bit[1]: 0         -1 1 -1 1
#2       Bit[1]: 1         1 1 -1 -1
#3       Bit[3]: 1         1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[4]: 1         1 1 1 1
#1       Bit[2]: 1         1 -1 1 -1
#2       Bit[2]: 1         1 1 -1 -1
#3       Bit[4]: 1         1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[5]: 1         1 1 1 1
#2       Bit[3]: 1         1 1 -1 -1
#3       Bit[5]: 0         -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#2       Bit[4]: 0         -1 -1 1 1

SSID#    Bit Sent SentData
--------------------------------
```

Number of stations: 6

```
The Walsh Code being used:
1 1 1 1 1 1 1 1
1 -1 1 -1 1 -1 1 -1
1 1 -1 -1 1 1 -1 -1
1 -1 -1 1 1 -1 -1 1
1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 1 -1 1
1 1 -1 -1 -1 -1 1 1
1 -1 -1 1 -1 1 1 -1

SSID     RSID     Chip Set
------------------------
#0       #1       1 1 1 1 1 1
#1       #2       1 -1 1 -1 1 -1
#2       #3       1 1 -1 -1 1 1
#3       #4       1 -1 -1 1 1 -1
#4       #5       1 1 1 1 -1 -1
#5       #0       1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[0]: 1         1 1 1 1 1 1
#1       Bit[0]: 0         -1 1 -1 1 -1 1
#2       Bit[0]: 1         1 1 -1 -1 1 1
#3       Bit[0]: 0         -1 1 1 -1 -1 1
#4       Bit[0]: 1         1 1 1 1 -1 -1
#5       Bit[0]: 0         -1 1 -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[1]: 0         -1 -1 -1 -1 -1 -1
#1       Bit[1]: 1         1 -1 1 -1 1 -1
#2       Bit[1]: 0         -1 -1 1 1 -1 -1
#5       Bit[1]: 1         1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#1       Bit[2]: 0         -1 1 -1 1 -1 1
#2       Bit[2]: 1         1 1 -1 -1 1 1
#4       Bit[1]: 1         1 1 1 1 -1 -1
#5       Bit[2]: 1         1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[2]: 0         -1 -1 -1 -1 -1 -1
#1       Bit[3]: 0         -1 1 -1 1 -1 1
#3       Bit[1]: 0         -1 1 1 -1 -1 1
#5       Bit[3]: 1         1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[3]: 0         -1 -1 -1 -1 -1 -1
#2       Bit[3]: 1         1 1 -1 -1 1 1
#3       Bit[2]: 0         -1 1 1 -1 -1 1
#4       Bit[2]: 1         1 1 1 1 -1 -1
#5       Bit[4]: 1         1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
```

```
#0       Bit[6]: 1         1 1 1 1
#2       Bit[5]: 0         -1 -1 1 1
#3       Bit[6]: 1         1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#1       Bit[3]: 0         -1 1 -1 1
#3       Bit[7]: 0         -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[7]: 0         -1 -1 -1 -1
#1       Bit[4]: 0         -1 1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#1       Bit[5]: 1         1 -1 1 -1
#2       Bit[6]: 0         -1 -1 1 1

SSID#    Bit Sent SentData
--------------------------------
#1       Bit[6]: 0         -1 1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#1       Bit[7]: 1         1 -1 1 -1
#2       Bit[7]: 1         1 1 -1 -1

Data Transfer Summary
--------------------------------
RSID     SSID     Data
--------------------------------
#0       #3       1 1 0 1 1 0 1 0
#1       #0       0 0 0 0 1 1 1 0
#2       #1       1 0 1 0 0 1 0 1
#3       #2       1 1 1 1 0 0 0 1
```

```
#0       Bit[4]: 0         -1 -1 -1 -1 -1 -1
#1       Bit[4]: 1         1 -1 1 -1 1 -1
#4       Bit[3]: 1         1 1 1 1 -1 -1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[5]: 1         1 1 1 1 1 1
#1       Bit[5]: 0         -1 1 -1 1 -1 1
#4       Bit[4]: 1         1 1 1 1 -1 -1
#5       Bit[5]: 1         1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[6]: 1         1 1 1 1 1 1
#1       Bit[6]: 0         -1 1 -1 1 -1 1
#2       Bit[4]: 0         -1 -1 1 1 -1 -1
#4       Bit[5]: 1         1 1 1 1 -1 -1
#5       Bit[6]: 1         1 -1 1 -1 -1 1

SSID#    Bit Sent SentData
--------------------------------
#0       Bit[7]: 1         1 1 1 1 1 1
#1       Bit[7]: 0         -1 1 -1 1 -1 1
#5       Bit[7]: 0         -1 1 -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#2       Bit[5]: 1         1 1 -1 -1 1 1
#3       Bit[3]: 1         1 -1 -1 1 1 -1
#4       Bit[6]: 1         1 1 1 1 -1 -1

SSID#    Bit Sent SentData
--------------------------------
#3       Bit[4]: 1         1 -1 -1 1 1 -1
#4       Bit[7]: 0         -1 -1 -1 -1 1 1

SSID#    Bit Sent SentData
--------------------------------
#2       Bit[6]: 0         -1 -1 1 1 -1 -1
#3       Bit[5]: 1         1 -1 -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#2       Bit[7]: 0         -1 -1 1 1 -1 -1
#3       Bit[6]: 1         1 -1 -1 1 1 -1

SSID#    Bit Sent SentData
--------------------------------
#3       Bit[7]: 1         1 -1 -1 1 1 -1

Data Transfer Summary
--------------------------------
RSID     SSID     Data
--------------------------------
#0       #5       0 1 0 1 1 0 0 0
#1       #0       1 0 0 0 0 0 1 1
#2       #1       0 1 0 0 1 0 0 0
#3       #2       1 0 0 0 0 0 0 0
#4       #3       0 0 0 0 0 0 0 1
#5       #4       1 0 0 0 0 1 1 0
```

Comments

- The assignment was pretty simple to implement. I just needed to think for a few minutes on managing the data over the channel. The main implementation of CDMA encoder and decoder was quite easy.