# Wireless Network Lab 6

## 108064535 陳文遠

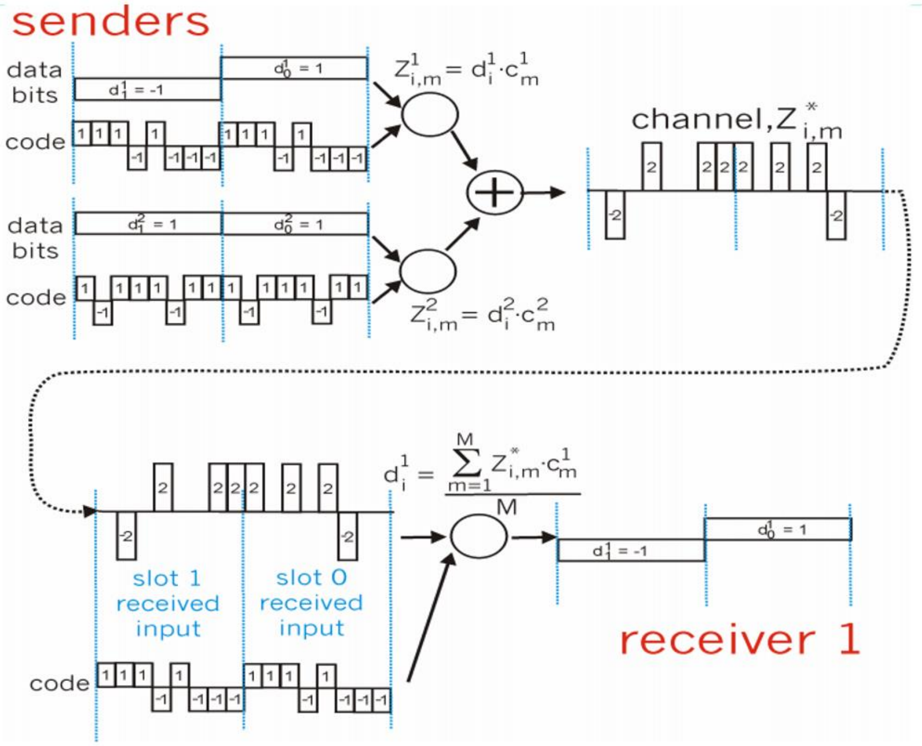
- ### *Objective*

Implement CMDA scheme.

- ### *Requirements*

Generate 8x8 matrix of Walsh codes, and then using it to encode and decode data in a CDMA scheme with at least two senders.

- ### *Architecture*



- ### *Implementation Step*

1. Generate 8x8 Walsh code matrix, and modify all the element 0 to -1.

$$C = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

2. Generate the matrix for two senders, and modify all element 0 to -1 as well. By the way, the first row of D represent the first sender and the second row of D represent the second sender.

$$D = \begin{bmatrix} 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix}$$

3. Encode: using the first row of C to times every element in the first row of D as well as using the second row of C to times every element in the second row of D. Finally, summarize the result to a matrix G.
4. Decode: using the same sender matrix to times the encoded matrix G. Summarize all the result and divide by M, then you can restore the data.

- ***CDMA Scheme Program (CDMA.m)***

```
clc; clear;

% CDMA Simulation for N Transmitter/Receiver Pairs


% data bit stream for each sender
D = [  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 ];


% generate a 8x8 matrix of Walsh codes (W)
C = 0;
for i=1:(4-1)
   C = [C, C; C, ~C];
end
X = size(C); X = X(1);
Y = size(C); Y = Y(2);
% modify 0 element in walsh code matrix to -1
for i=1:X
   for j=1:Y
      if C(i, j) == 0
         C(i, j) = -1;
      end
   end
end


% parameters
M = length(C);      % length (number of bits) of code
Y = size(D);
N = Y(1);          % number of unique senders / bit streams
I = Y(2);          % number of bits per stream
T = [];            % sum of all transmitted and encoded data on channel
RECON = [];         % vector of reconstructed bits at receiver
```

```
% show data bits and codes
'Vector of senders to be transmitted:', D
'Vector of walsh codes used for transmission:', C


% encode bits and transmit
G = zeros(I,M);
for n = 1:N
    Z = zeros(I,M);
    for i = 1:I
        for m = 1:M
            Z(i,m) = [D(n,i)*C(n,m)];
        end
    end
    G = G + Z;
end
 % show channel traffic
for i = 1:I
    T = [ T G(i,:) ];
end
'Resulting traffic on the channel:', T


% decode and reconstruct
for n = 1:N
    TOT = zeros(1,I);
    R   = zeros(I,M);
    for i = 1:I
        for m = 1:M
            R(i,m) = G(i,m) * C (n,m);
            TOT(i) = TOT(i) + R (i,m);
         end
    end
    RECON = [RECON ; TOT / M];
end
'Reconstructed data at the receiver:', RECON
```