Mobile Communications

TCS 455

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Lecture 21

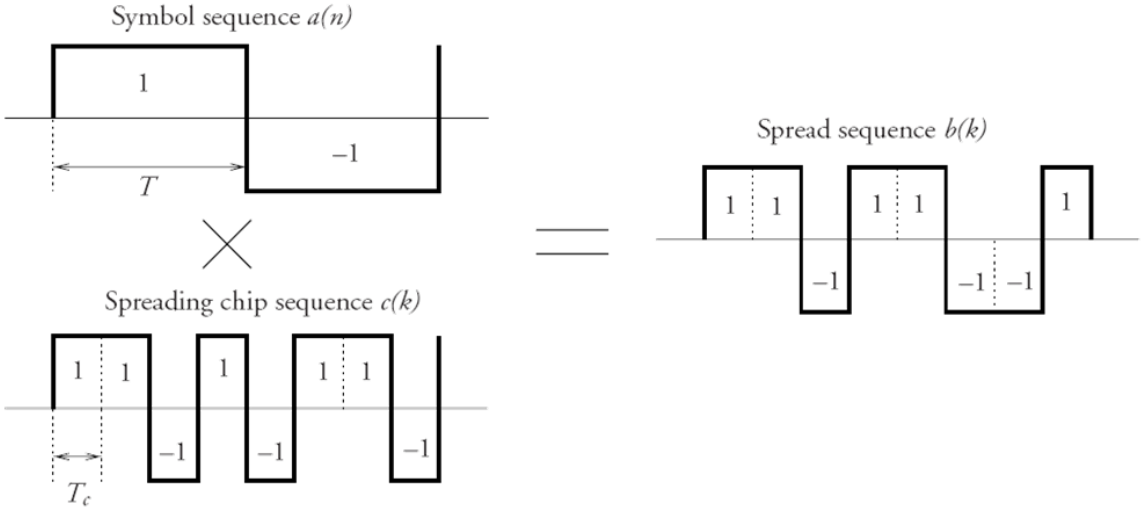
Office Hours:
BKD 3601-7
Tuesday 14:00-16:00
Thursday 9:30-11:30

Announcements

- Read
  - Chapter 9: 9.1 – 9.5
- HW5 is posted.
  - Due: Feb 5 (This Friday)

DSSS and Maximum-length Codes

- Maximal length codes have **excellent auto-correlation** properties (for ISI rejection), they have a number of properties that make them highly suboptimal for exploiting the multiuser capabilities of spread spectrum.
- There are only a small number of maximal length codes of a given length.
- Moreover, maximal length codes generally have relatively **poor cross-correlation** properties, at least for some sets of codes.



SSMA

- For spread spectrum systems with **multiple users**, codes such as Gold, Kasami, or Walsh codes are used instead of maximal length codes
- Superior cross-correlation properties.
- Worse auto-correlation than maximal length codes.
  - The autocorrelation function of the spreading code determines its multipath rejection properties.

Orthogonality

- Two vectors/functions are **orthogonal** if their **inner product** is zero.
- The symbol $\perp$ is used to denote orthogonality.

Vector:

$$\left\langle \vec{a}, \vec{b} \right\rangle = \vec{a} \cdot \vec{b}^{*} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}^{*} = \sum_{k=1}^{n} a_k b_k^{*} = 0$$

Time-domain:

$$\left\langle a, b \right\rangle = \int_{-\infty}^{\infty} a(t) b^{*}(t) dt = 0$$

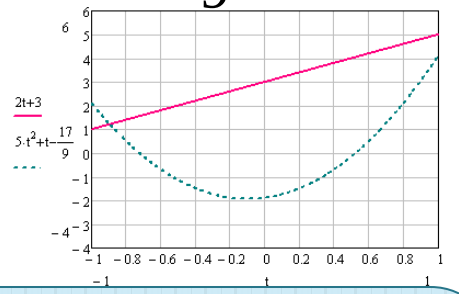
Complex conjugate

Frequency domain:

$$\left\langle A, B \right\rangle = \int_{-\infty}^{\infty} A(f) B^{*}(f) df = 0$$

Example:

$$2t + 3 \text{ and } 5t^2 + t - \frac{17}{9} \text{ on } [-1,1]$$



Example (Fourier Series):

$$\sin\left(2\pi k_1 \frac{t}{T}\right) \text{ and } \cos\left(2\pi k_2 \frac{t}{T}\right) \text{ on } [0,T]$$

$$e^{j2\pi n \frac{t}{T}} \text{ on } [0,T]$$

Parseval's Theorem

$$\langle x, y \rangle = \int_{-\infty}^{\infty} x(t)\, y^*(t)\, dt = \int_{-\infty}^{\infty} X(f)\, Y^*(f)\, df = \langle X, Y \rangle$$

If $x(t) \perp y(t)$, then $X(f) \perp Y(f)$.

Orthogonality in Communication

CDMA

$$s(t)=\sum_{k=0}^{\ell-1} S_k c_k(t) \xrightarrow{\mathcal{F}} S(f)=\sum_{k=0}^{\ell-1} S_k C_k(f) \qquad \text{where} \quad c_{k_1} \perp c_{k_2}$$

TDMA

$$s(t)=\sum_{k=0}^{\ell-1} S_k c(t-kT_s) \xrightarrow{\mathcal{F}} S(f)=C(f)\sum_{k=0}^{\ell-1} S_k e^{-j2\pi fkT_s}$$

where $c(t)$ is time-limited to $[0,T]$.

This is a special case of CDMA with $c_k(t)=c(t-kT_s)$

The c_k are non-overlapping in time domain.

FDMA

$$S(f)=\sum_{k=0}^{\ell-1} S_k C(f-k\Delta f)$$

where $C(f)$ is frequency-limited to $[0,\Delta f]$.

This is a special case of CDMA with $C_k(f)=C(f-k\Delta f)$

The C_k are non-overlapping in freq. domain.

Chapter 4
Multiple Access

Synchronous CDMA

Office Hours:
BKD 3601-7
Tuesday 14:00-16:00
Thursday 9:30-11:30

Synchronous CDMA Model

- Timing is important for orthogonality
- It is not possible to obtain orthogonal codes for asynchronous users.
- For synchronous users there is only a finite number of spreading codes that are orthogonal within any given bandwidth.
- Bit epochs are aligned at the receiver
- Require
  - Closed-loop timing control or
  - Providing the transmitters with access to a common clock (such as the Global Positioning System)

Walsh Functions [Walsh, 1923]

- Walsh codes are used in second- (2G) and third-generation (3G) cellular radio systems for providing channelization
- A set of Walsh functions can be ordered according to the number of zero crossing (sign changes)

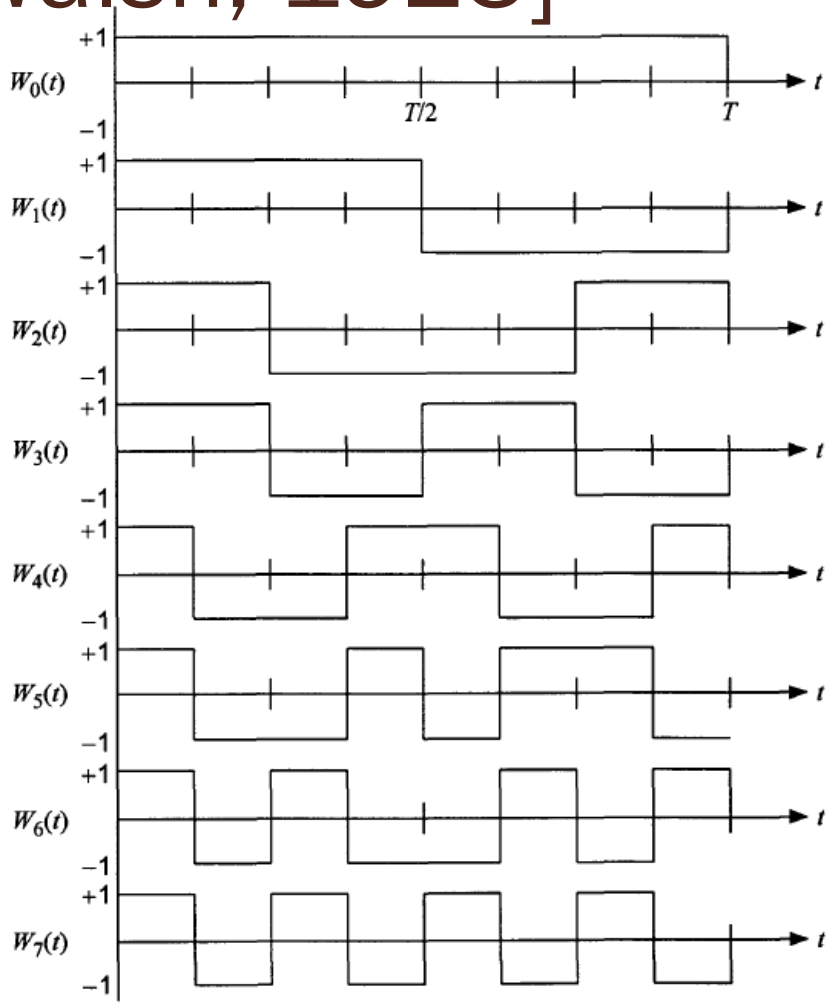


Figure 5.1 The Walsh functions of order 8.

[Lee and Miller, 1998, Fig. 5.1]

Walsh Functions (2)

We define the Walsh functions of order N as a set of N time functions, denoted $\{W_j(t); t \in (0,T), j = 0, 1, \ldots, N-1\}$, such that

- $W_j(t)$ takes on the values $\{+1, -1\}$ except at the jumps, where it takes the value zero.
- $W_j(0) = 1$ for all j.
- $W_j(t)$ has precisely j sign changes (zero crossings) in the interval $(0, T)$.
- $$\int_0^T W_j(t)\, W_k(t)\, dt = \begin{cases} 0, & \text{if } j \neq k \\ T, & \text{if } j = k \end{cases}$$ Orthogonality
- Each function $W_j(t)$ is either odd or even with respect to the midpoint of the interval.

Once we know how to generate these Walsh functions of any order N, we can use them in N-channel orthogonal multiplexing applications.

Walsh Sequences

Walsh sequences	
W_0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
W_1	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
W_2	0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
W_3	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
W_4	0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
W_5	0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1
W_6	0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0
W_7	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
W_8	0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
W_9	0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1
W_{10}	0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0
W_{11}	0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1
W_{12}	0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0
W_{13}	0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1
W_{14}	0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0
W_{15}	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

- The Walsh functions, expressed in terms of ± 1 values, form a group under the multiplication operation (multiplicative group).
- The Walsh sequences, expressed in terms of (0, 1) values, form a group under modulo-2 addition (additive group).
- **Closure property**:

$$W_i(t) \cdot W_j(t) = W_r(t)$$

$$W_i \oplus W_j = W_r$$

Walsh sequences of order 64

Table 5.2 Walsh functions of order 64 (indexed by zero crossings)

W_0	0000000000000000 0000000000000000 0000000000000000 0000000000000000
W_1	0000000000000000 0000000000000000 1111111111111111 1111111111111111
W_2	0000000000000000 1111111111111111 1111111111111111 0000000000000000
W_3	0000000000000000 1111111111111111 0000000000000000 1111111111111111
W_4	0000000011111111 1111111100000000 0000000011111111 1111111100000000
W_5	0000000011111111 1111111100000000 1111111100000000 0000000011111111
W_6	0000000011111111 0000000011111111 1111111100000000 1111111100000000
W_7	0000000011111111 0000000011111111 0000000011111111 0000000011111111
W_8	0000111111110000 0000111111110000 0000111111110000 0000111111110000
W_9	0000111111110000 0000111111110000 1111000000001111 1111000000001111
W_{10}	0000111111110000 1111000000001111 1111000000001111 0000111111110000
W_{11}	0000111111110000 1111000000001111 0000111111110000 1111000000001111
W_{12}	0000111100001111 1111000011110000 0000111100001111 1111000011110000
W_{13}	0000111100001111 1111000011110000 1111000011110000 0000111100001111
W_{14}	0000111100001111 0000111100001111 1111000011110000 1111000011110000
W_{15}	0000111100001111 0000111100001111 0000111100001111 0000111100001111
W_{16}	0011110000111100 0011110000111100 0011110000111100 0011110000111100
W_{17}	0011110000111100 0011110000111100 1100001111000011 1100001111000011
W_{18}	0011110000111100 1100001111000011 1100001111000011 0011110000111100
W_{19}	0011110000111100 1100001111000011 0011110000111100 1100001111000011
W_{20}	0011110011000011 1100001100111100 0011110011000011 1100001100111100
W_{21}	0011110011000011 1100001100111100 1100001100111100 0011110011000011
W_{22}	0011110011000011 0011110011000011 1100001100111100 1100001100111100
W_{23}	0011110011000011 0011110011000011 0011110011000011 0011110011000011
W_{24}	0011001111001100 0011001111001100 0011001111001100 0011001111001100
W_{25}	0011001111001100 0011001111001100 1100110000110011 1100110000110011
W_{26}	0011001111001100 1100110000110011 1100110000110011 0011001111001100
W_{27}	0011001111001100 1100110000110011 0011001111001100 1100110000110011
W_{28}	0011001100110011 1100110011001100 0011001100110011 1100110011001100
W_{29}	0011001100110011 1100110011001100 1100110011001100 0011001100110011
W_{30}	0011001100110011 0011001100110011 1100110011001100 1100110011001100
W_{31}	0011001100110011 0011001100110011 0011001100110011 0011001100110011
W_{32}	0110011001100110 0110011001100110 0110011001100110 0110011001100110
W_{33}	0110011001100110 0110011001100110 1001100110011001 1001100110011001
W_{34}	0110011001100110 1001100110011001 1001100110011001 0110011001100110
W_{35}	0110011001100110 1001100110011001 0110011001100110 1001100110011001
W_{36}	0110011010011001 1001100101100110 0110011010011001 1001100101100110
W_{37}	0110011010011001 1001100101100110 1001100101100110 0110011010011001
W_{38}	0110011010011001 0110011010011001 1001100101100110 1001100101100110
W_{39}	0110011010011001 0110011010011001 0110011010011001 0110011010011001
W_{40}	0110100110010110 0110100110010110 0110100110010110 0110100110010110
W_{41}	0110100110010110 0110100110010110 1001011001101001 1001011001101001
W_{43}	0110100110010110 1001011001101001 0110100110010110 1001011001101001
W_{44}	0110100101101001 1001011010010110 0110100101101001 1001011010010110
W_{45}	0110100101101001 1001011010010110 1001011010010110 0110100101101001
W_{46}	0110100101101001 0110100101101001 1001011010010110 1001011010010110
W_{47}	0110100101101001 0110100101101001 0110100101101001 0110100101101001
W_{48}	0101101001011010 0101101001011010 0101101001011010 0101101001011010
W_{49}	0101101001011010 0101101001011010 1010010110100101 1010010110100101
W_{50}	0101101001011010 1010010110100101 1010010110100101 0101101001011010
W_{51}	0101101001011010 1010010110100101 0101101001011010 1010010110100101
W_{52}	0101101010100101 1010010101011010 0101101010100101 1010010101011010
W_{53}	0101101010100101 1010010101011010 1010010101011010 0101101010100101
W_{54}	0101101010100101 0101101010100101 1010010101011010 1010010101011010
W_{55}	0101101010100101 0101101010100101 0101101010100101 0101101010100101
W_{56}	0101010110101010 0101010110101010 0101010110101010 0101010110101010
W_{57}	0101010110101010 0101010110101010 1010101001010101 1010101001010101
W_{58}	0101010110101010 1010101001010101 1010101001010101 0101010110101010
W_{59}	0101010110101010 1010101001010101 0101010110101010 1010101001010101
W_{60}	0101010101010101 1010101010101010 0101010101010101 1010101010101010
W_{61}	0101010101010101 1010101010101010 1010101010101010 0101010101010101
W_{62}	0101010101010101 0101010101010101 1010101010101010 1010101010101010
W_{63}	0101010101010101 0101010101010101 0101010101010101 0101010101010101

What's wrong with this list?!

[Lee and Miller, 1998, Table 5.2]

Walsh Function Generation

- The Walsh functions can be generated (or computed) by many methods. We can construct the Walsh functions by:
  - Using Rademacher functions;
  - Using **Hadamard matrices**;
  - Exploiting the symmetry properties of Walsh functions themselves.
- The **Hadamard matrix** is a square array of plus and minus ones, {+1, -1}, whose rows and columns are mutually orthogonal.
- If the first row and first column contain only plus ones, the matrix is said to be in **normal form**.
- We can replace "+1" with "0" and "-1" with "1" to express the Hadamard matrix using the logic elements (0, 1).
- The 2×2 Hadamard matrix of order 2 is

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Hadamard matrix (1)

Suppose H_N is an $N \times N$ Hadamard matrix. $N \geq 1$ is called the order of a Hadamard matrix

1. $N = 1$, 2, or $4t$, where t is a positive integer.
2. $H_N H_N^T = N I_N$ where I_N is the $N \times N$ identity matrix.
3. If H_a and H_b are Hadamard matrices of order a and b, respectively, then we define $H_a \otimes H_b$ to be the Hadamard matrix H_{ab} of order ab whose elements are found by substituting H_b for $+1$ (or logic 0) in H_a and $-H_b$ (or the complement of H_b) for -1 (or logic 1) in H_a.

Caution: Some textbooks write this symbol as ×. It is not the regular matrix multiplication

Hadamard matrix (2)

- Consequently, if N is a power of two and it is understood that $H_1 = [+1] \equiv [0]$, then H_{2N} can be found as follows:

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & \overline{H_N} \end{bmatrix}$$

  where $\overline{H_N}$ is the negative (complement) of H_N.

- Hadamard matrices of order $N = 2^t$ can be formed by repeatedly multiplying ($\otimes$) the normal form of the $N = 2$ Hadamard matrix by itself.

Hadamard matrix: Examples

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Caution: This is not the usual matrix multiplication!

$$\mathbf{H}_4 = \mathbf{H}_2 \times \mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{H}_8 = \mathbf{H}_2 \times \mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$$\mathbf{H}_{16} = \mathbf{H}_2 \times \mathbf{H}_8 =$$

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

In MATLAB, use

```
hadamard(k)
```

Two ways to get H_8 from H_2 and H_4

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H_8 = H_2 \otimes H_4$$

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

$$H_8 = H_4 \otimes H_2$$

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

Walsh–Hadamard Sequences

- All the row (or column) sequences of Hadamard matrices are Walsh sequences if the order is $N = 2^t$.
- The Walsh functions generated by the Hadamard matrix method are not indexed according to the number of sign changes.
- Used in synchronous CDMA
  - It is possible to synchronize users on the downlink, where all signals originate from the same transmitter.
  - It is more challenging to synchronize users in the uplink, since they are not co-located.
    - Asynchronous CDMA

Hadamard Matrix in MATLAB

- We use the `hadamard` function in MATLAB to generate Walsh functions of length eight.

```
N = 8;  % Length of Walsh (Hadamard) functions
hadamardMatrix = hadamard(N)
hadamardMatrix =

     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
```

- The Walsh functions in the matrix are not arranged in increasing order of their sequencies or number of zero-crossings (i.e. 'sequency order') .

Walsh Matrix in MATLAB

- The Walsh matrix, which contains the Walsh functions along the rows or columns in the increasing order of their sequencies is obtained by changing the index of the `hadamardMatrix` as follows.

```
HadIdx = 0:N-1;                          % Hadamard index
M = log2(N)+1;                           % Number of bits to represent the index
```

- Each column of the sequency index (in binary format) is given by the modulo-2 addition of columns of the bit-reversed `Hadamard index` (in binary format).

```
binHadIdx = fliplr(dec2bin(HadIdx,M));    % Bit reversing of the binary index
binHadIdx = uint8(binHadIdx)-uint8('0'); % Convert from char to integer array
binSeqIdx = zeros(N,M-1,'uint8');         % Pre-allocate memory
for k = M:-1:2
    % Binary sequency index
    binSeqIdx(:,k) = xor(binHadIdx(:,k),binHadIdx(:,k-1));
end
SeqIdx = bin2dec(int2str(binSeqIdx));     % Binary to integer sequency index
walshMatrix = hadamardMatrix(SeqIdx+1,:) % 1-based indexing
```

```
walshMatrix =

     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
```

CDMA via Hadamard Matrix

- This signal is formed using weighted Walsh functions, so the WHT should return non-zero values equal to the weights at the respective indices

Encoding
Decoding

```
N = 8;
H = hadamard(N);                          % Hadamard matrix
% Construct a signal by adding a few weighted Walsh functions
x = 8.*H(1,:) + 12.*H(3,:) + 18.*H(5,:) + 10.*H(8,:);
y = fwht(x,N,'hadamard')
y =

     8     0    12     0    18     0     0    10
```

Discrete Walsh-Hadamard transform

Specify the order of the Walsh-Hadamard transform coefficients. ORDERING can be 'sequency', 'hadamard' or 'dyadic'. Default ORDERING type is 'sequency'.

Note that $y = \frac{1}{N} x H^T$

Chapter 4
Multiple Access

IS-95

Office Hours:
BKD 3601-7
Tuesday 14:00-16:00
Thursday 9:30-11:30

Evolution of cellular network

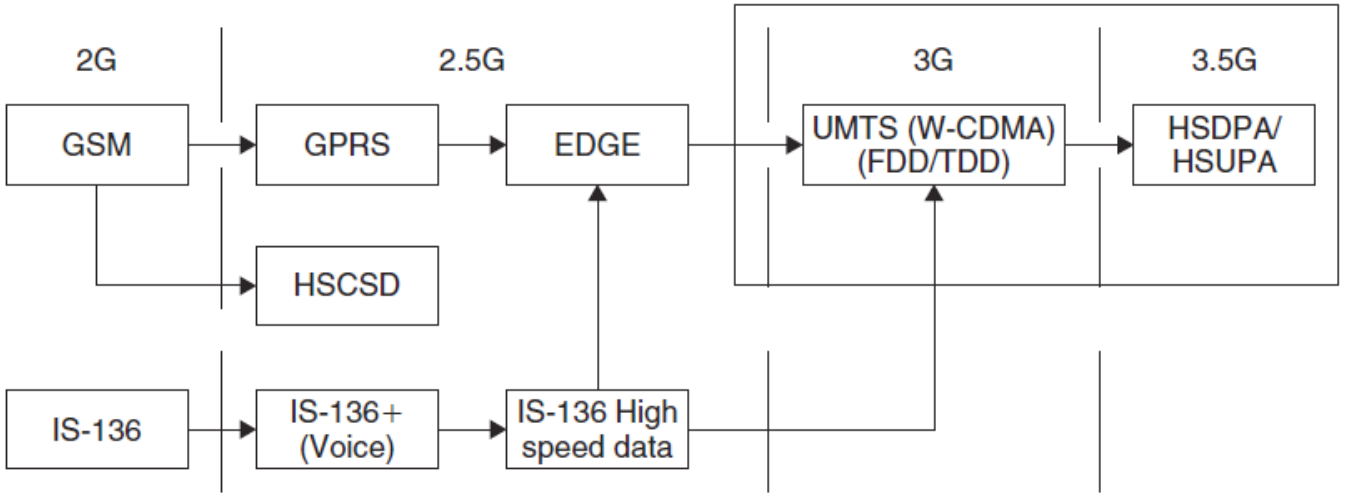


Figure 1.1 *Evolution of 2G networks based on TDMA technology.*

[Abu-Rgheff, 2007]

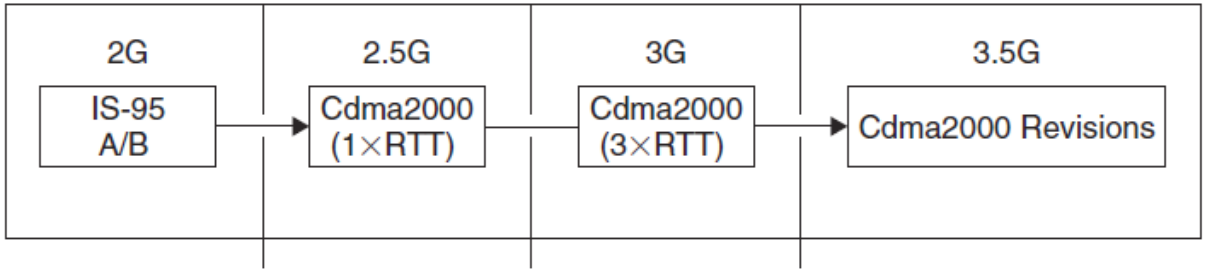


Figure 1.2 *Evolution of 2G networks based on CDMA technology.*

IS-95 System

cdmaOne

- Based on direct sequence CDMA (DS-CDMA)
  - First CDMA-based digital cellular standard.
- The brand name for IS-95 is cdmaOne.
  - Also known as TIA-EIA-95.
- Proposed by Qualcomm in 1989 and adopted in 1933.
  - North America
- Now being replaced by IS-2000 (CDMA2000)
- 1.25 MHz Channel BW
- 1.228 Mb/s chip rate
- Walsh functions of "order 64" are extensively used in the IS-95 system.
- Remarks
  - IS-95B = cdmaOne
    - Ugrade IS-95A
    - Can carry data at rates up to 14.4 kbps for IS-95A and 115 kbps for IS-95B.

64-ary Walsh Functions

Table 5.8 Walsh functions of order 64, as indexed in IS-95 (W_i is the Walsh notation, and H_i is the Hadamard notation)

W_0	H_0	0000000000000000	0000000000000000	0000000000000000	0000000000000000	W_1	H_{32}	0000000000000000	0000000000000000	1111111111111111	1111111111111111
W_{63}	H_1	0101010101010101	0101010101010101	0101010101010101	0101010101010101	W_{62}	H_{33}	0101010101010101	0101010101010101	1010101010101010	1010101010101010
W_{31}	H_2	0011001100110011	0011001100110011	0011001100110011	0011001100110011	W_{30}	H_{34}	0011001100110011	0011001100110011	1100110011001100	1100110011001100
W_{32}	H_3	0110011001100110	0110011001100110	0110011001100110	0110011001100110	W_{33}	H_{35}	0110011001100110	0110011001100110	1001100110011001	1001100110011001
W_{15}	H_4	0000111100001111	0000111100001111	0000111100001111	0000111100001111	W_{14}	H_{36}	0000111100001111	0000111100001111	1111000011110000	1111000011110000
W_{48}	H_5	0101101001011010	0101101001011010	0101101001011010	0101101001011010	W_{49}	H_{37}	0101101001011010	0101101001011010	1010010110100101	1010010110100101
W_{16}	H_6	0011110000111100	0011110000111100	0011110000111100	0011110000111100	W_{17}	H_{38}	0011110000111100	0011110000111100	1100001111000011	1100001111000011
W_{47}	H_7	0110100101101001	0110100101101001	0110100101101001	0110100101101001	W_{46}	H_{39}	0110100101101001	0110100101101001	1001011010010110	1001011010010110
W_7	H_8	0000000011111111	0000000011111111	0000000011111111	0000000011111111	W_6	H_{40}	0000000011111111	0000000011111111	1111111100000000	1111111100000000
W_{56}	H_9	0101010110101010	0101010110101010	0101010110101010	0101010110101010	W_{57}	H_{41}	0101010110101010	0101010110101010	1010101001010101	1010101001010101
W_{24}	H_{10}	0011001111001100	0011001111001100	0011001111001100	0011001111001100	W_{25}	H_{42}	0011001111001100	0011001111001100	1100110000110011	1100110000110011
W_{39}	H_{11}	0110011010011001	0110011010011001	0110011010011001	0110011010011001	W_{38}	H_{43}	0110011010011001	0110011010011001	1001100101100110	1001100101100110
W_8	H_{12}	0000111111110000	0000111111110000	0000111111110000	0000111111110000	W_9	H_{44}	0000111111110000	0000111111110000	1111000000001111	1111000000001111
W_{55}	H_{13}	0101101010100101	0101101010100101	0101101010100101	0101101010100101	W_{54}	H_{45}	0101101010100101	0101101010100101	1010010101011010	1010010101011010
W_{23}	H_{14}	0011110011000011	0011110011000011	0011110011000011	0011110011000011	W_{22}	H_{46}	0011110011000011	0011110011000011	1100001100111100	1100001100111100
W_{40}	H_{15}	0110100110010110	0110100110010110	0110100110010110	0110100110010110	W_{41}	H_{47}	0110100110010110	0110100110010110	1001011001101001	1001011001101001
W_3	H_{16}	0000000000000000	1111111111111111	0000000000000000	1111111111111111	W_2	H_{48}	0000000000000000	1111111111111111	1111111111111111	0000000000000000
W_{60}	H_{17}	0101010101010101	1010101010101010	0101010101010101	1010101010101010	W_{61}	H_{49}	0101010101010101	1010101010101010	1010101010101010	0101010101010101
W_{28}	H_{18}	0011001100110011	1100110011001100	0011001100110011	1100110011001100	W_{29}	H_{50}	0011001100110011	1100110011001100	1100110011001100	0011001100110011
W_{35}	H_{19}	0110011001100110	1001100110011001	0110011001100110	1001100110011001	W_{34}	H_{51}	0110011001100110	1001100110011001	1001100110011001	0110011001100110
W_{12}	H_{20}	0000111100001111	1111000011110000	0000111100001111	1111000011110000	W_{13}	H_{52}	0000111100001111	1111000011110000	1111000011110000	0000111100001111
W_{51}	H_{21}	0101101001011010	1010010110100101	0101101001011010	1010010110100101	W_{50}	H_{53}	0101101001011010	1010010110100101	1010010110100101	0101101001011010
W_{19}	H_{22}	0011110000111100	1100001111000011	0011110000111100	1100001111000011	W_{18}	H_{54}	0011110000111100	1100001111000011	1100001111000011	0011110000111100
W_{44}	H_{23}	0110100101101001	1001011010010110	0110100101101001	1001011010010110	W_{45}	H_{55}	0110100101101001	1001011010010110	1001011010010110	0110100101101001
W_4	H_{24}	0000000011111111	1111111100000000	0000000011111111	1111111100000000	W_5	H_{56}	0000000011111111	1111111100000000	1111111100000000	0000000011111111
W_{59}	H_{25}	0101010110101010	1010101001010101	0101010110101010	1010101001010101	W_{58}	H_{57}	0101010110101010	1010101001010101	1010101001010101	0101010110101010
W_{27}	H_{26}	0011001111001100	1100110000110011	0011001111001100	1100110000110011	W_{26}	H_{58}	0011001111001100	1100110000110011	1100110000110011	0011001111001100
W_{36}	H_{27}	0110011010011001	1001100101100110	0110011010011001	1001100101100110	W_{37}	H_{59}	0110011010011001	1001100101100110	1001100101100110	0110011010011001
W_{11}	H_{28}	0000111111110000	1111000000001111	0000111111110000	1111000000001111	W_{10}	H_{60}	0000111111110000	1111000000001111	1111000000001111	0000111111110000
W_{52}	H_{29}	0101101010100101	1010010101011010	0101101010100101	1010010101011010	W_{53}	H_{61}	0101101010100101	1010010101011010	1010010101011010	0101101010100101
W_{20}	H_{30}	0011110011000011	1100001100111100	0011110011000011	1100001100111100	W_{21}	H_{62}	0011110011000011	1100001100111100	1100001100111100	0011110011000011
W_{43}	H_{31}	0110100110010110	1001011001101001	0110100110010110	1001011001101001	W_{12}	H_{63}	0110100110010110	1001011001101001	1001011001101001	0110100110010110

[Lee and Miller, 1998, Table 5.8]

Walsh Sequences in IS-95

- Forward link
  - QPSK with a chip rate of 1,228,800 per second.
  - The **multiple access scheme** is accomplished by the use of 64-bit spreading orthogonal **Walsh sequences** (functions).
    - The (coded and interleaved) traffic channel signal symbols are multiplied with distinct repeating Walsh sequences that are assigned to each channel for the duration of the call.
  - Every base stations is synchronized with a GPS receiver so transmissions are tightly controlled in time.
- Reverse link
  - The Walsh sequences are employed as an **orthogonal modulation code**, which depends only on the data pattern (not channel), forming a 64-ary orthogonal modulation system.

IS-95

- The reverse link is subject to near-far effects.
- More powerful error correction is employed on the reverse link.
  - A rate 1/2 constraint length 9 convolutional code followed by an interleaver on the forward channel while
  - A rate 1/3 constraint length 9 convolutional code followed by an interleaver is used on the reverse link.
  - Interleaving is utilized to avoid large burst errors, which can be very detrimental to convolutional codes.
- Power control.
  - Use a subchannel on the forward link
  - Every 1.25 ms the base station receiver estimates the signal strength of the mobile unit.
  - If it is too high, the base transmits a 1 on the subchannel. If it is too low, it transmits a 0.
  - In this way, the mobile station adjusts its power every 1.25 ms as necessary so as to reduce interference to other users.

IS-95: Increased Spectral Efficiency

- Improve frequency reuse.
  - Narrow-band systems cannot use the same transmission frequency in adjacent cells because of the potential for interference.
  - CDMA has inherent resistance to interference.
    - N = 1 (theoretically)
    - Although users from adjacent cells will contribute to interference level, their contribution will be significantly less than the interference from the same cell users.
    - Frequency reuse efficiency increases by a factor of 4 to 6.
- When used to transmit voice signals, CDMA systems may exploit the fact that voice activity typically lies at somewhat less than 40%, thus reducing the amount of interference to 40% of its original value.

QCELP

- Qualcomm code-excited linear prediction algorithm
- Used for voice encoding.
- The voice coder exploits gaps and pauses in speech.
- The data rate is variable.
- To keep the symbol rate constant, whenever the bit rate falls below the peak bit rate of 9600 kbit/s, repetition is used to fill the gaps.
  - For example, if the output of the voice coder (and subsequently the convolutional coder) falls to 2400 bit/s, the output is repeated three times before it is sent to the interleaver.
  - Takes advantage of this repetition time by reducing the output power during three out of the four identical symbols by at least 20 dB.
  - In this way, the multiple-access interference is reduced.
- This voice activity gating reduces interference and increases overall capacity.