

ECS455: Chapter 5

OFDM

5.3 Implementation: DFT and FFT



Dr. Prapun Suksompong

www.prapun.com

Office Hours:

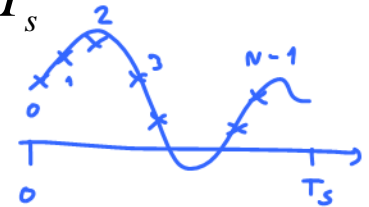
Library (Rangsit) Mon 16:20-16:50

BKD 3601-7 Wed 9:20-11:20

Discrete Fourier Transform (DFT)

Transmitter produces

$$s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kt}{T_s}\right), \quad 0 \leq t \leq T_s$$



Sample the signal in time domain every T_s/N gives

$$s[n] = s\left(n \frac{T_s}{N}\right) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi k}{T_s} n \frac{T_s}{N}\right)$$

$$\stackrel{n=0,1,\dots,N-1}{=} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kn}{N}\right) = \sqrt{N} \text{IDFT}\{S\}[n]$$

$\equiv N \text{IDFT}\{S\}[n]$

CTCF FT

We can implement OFDM in the discrete domain!

DTCF DTFT

DTDF DFT

Discrete Fourier Transform (DFT)

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$



In DFT, we work with N -point signal (finite-length sequence of length N) in both time and frequency domain. To simplify the definition we define

$$\psi_N = e^{j\frac{2\pi}{N}}$$

$$\psi_2 = e^{j\frac{2\pi}{2}} = e^{j\pi} = -1$$

and the DFT matrix $Q = \Psi_N$ whose element on the p th row and q th column is given by $\psi_N^{-(p-1)(q-1)}$:

$$\Psi_N = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \psi_N^{-1} & \psi_N^{-2} & \dots & \psi_N^{-(N-1)} \\ 1 & \psi_N^{-2} & \psi_N^{-4} & \dots & \psi_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \psi_N^{-(N-1)} & \psi_N^{-2(N-1)} & \dots & \psi_N^{-(N-1)(N-1)} \end{bmatrix}$$

$$\Psi_2 = \begin{bmatrix} 1 & 1 \\ 1 & \psi_2^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The “-1” are there because we start from row 1 and column 1.

$$\tilde{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

Key Property:

$$= \frac{1}{N} \Psi_N^H$$

$$\Psi_N^{-1} = \frac{1}{N} \Psi_N^* \cdot \text{Equivalently, } \Psi_N^{-1} \Psi_N = I_N.$$

$$\frac{1}{\sqrt{N}} \Psi_N \text{ is a unitary matrix}$$

DFT operation takes N -pt signal to another N -pt signal.

Definition 5.3. The N -point DFT of an N -point signal (column vector) x is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jnk\frac{2\pi}{N}} = \sum_{n=0}^{N-1} x[n] \psi_N^{-nk} ; 0 \leq k < N.$$

The inverse DFT is given by

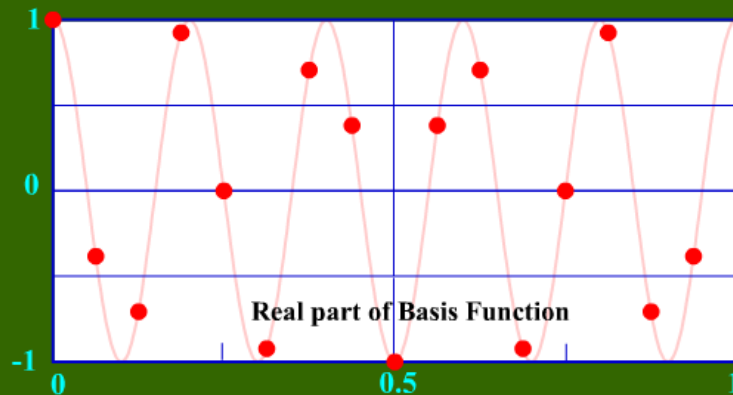
$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \psi_N^{nk} \xleftrightarrow[\text{DFT}^{-1}]{\text{DFT}} X[k] = \sum_{n=0}^{N-1} x[n] \psi_N^{-nk}$$

In matrix form,

$$x = \frac{1}{N} \Psi_N^* X \xleftrightarrow[\text{DFT}^{-1}]{\text{DFT}} X = \Psi_N \times x.$$

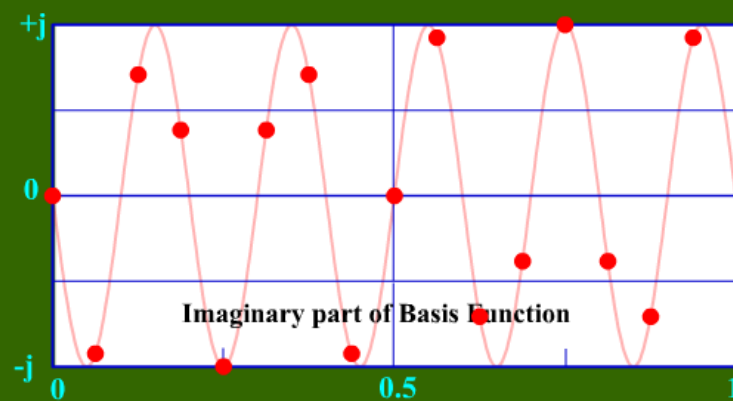
DFT: Example

Digitized Basis Functions for a 16 point DFT



16 samples of
real part of
basis function
for 16 pt. DFT

$\cos(2\pi \cdot 5n/16)$



16 samples of
imag. part of
basis function
for 16 pt. DFT

$-j \cdot \sin(2\pi \cdot 5n/16)$

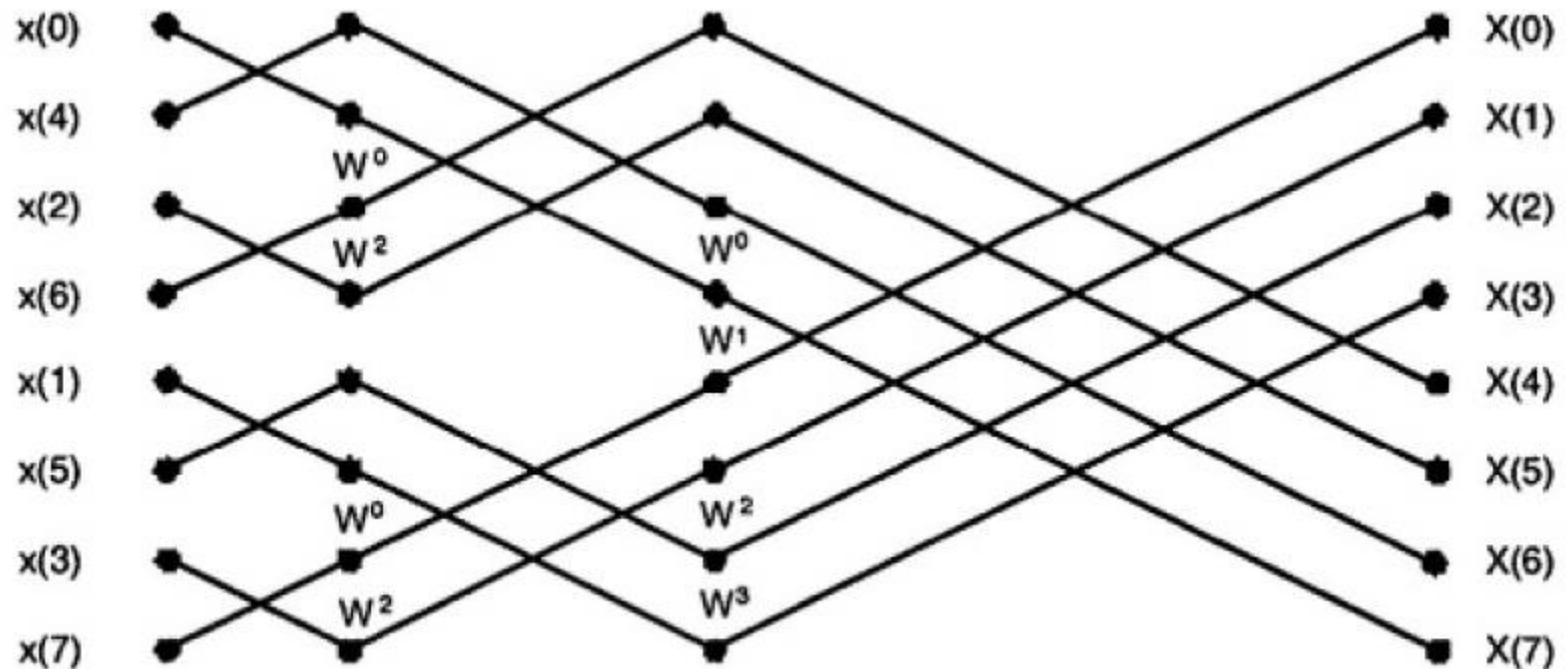
Pick one of the 16 basis functions

- $e^{-j2\pi \cdot 0n/16}$
- $e^{-j2\pi \cdot 1n/16}$ $e^{-j2\pi \cdot 15n/16}$
- $e^{-j2\pi \cdot 2n/16}$ $e^{-j2\pi \cdot 14n/16}$
- $e^{-j2\pi \cdot 3n/16}$ $e^{-j2\pi \cdot 13n/16}$
- $e^{-j2\pi \cdot 4n/16}$ $e^{-j2\pi \cdot 12n/16}$
- $e^{-j2\pi \cdot 5n/16}$** $e^{-j2\pi \cdot 11n/16}$
- $e^{-j2\pi \cdot 6n/16}$ $e^{-j2\pi \cdot 10n/16}$
- $e^{-j2\pi \cdot 7n/16}$ $e^{-j2\pi \cdot 9n/16}$
- $e^{-j2\pi \cdot 8n/16}$

DTFT to DFT

- Start with a sequence in discrete time $x[n]$.
- Z-transform: $X(z) = \sum_n x[n]z^{-n}$
- Discrete-Time Fourier Transform: $X(e^{j\omega}) = \sum_n x[n]e^{-j\omega n}$
- N points in time domain: $X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}$
- DFT: $X_k = X(e^{j\omega}) \Big|_{\omega=\omega_k=\frac{k}{N}2\pi} = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$

Efficient Implementation: (I)FFT



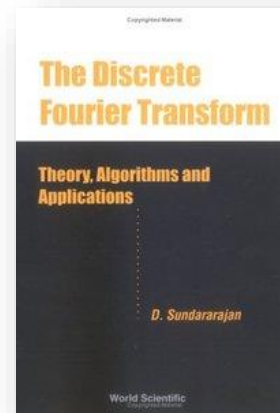
[Bahai, 2002, Fig. 2.9]

An N -point FFT requires only on the order of $N \log N$ multiplications, rather than N^2 as in a straightforward computation.

FFT

- The history of the FFT is complicated.
- As with many discoveries and inventions, it arrived before the (computer) world was ready for it.
- Usually done with N a power of two.
 - Very efficient in terms of computing time
 - Ideally suited to the binary arithmetic of digital computers.
 - Ex: From the implementation point of view it is better to have, for example, a FFT size of 1024 even if only 600 outputs are used than try to have another length for FFT between 600 and 1024.

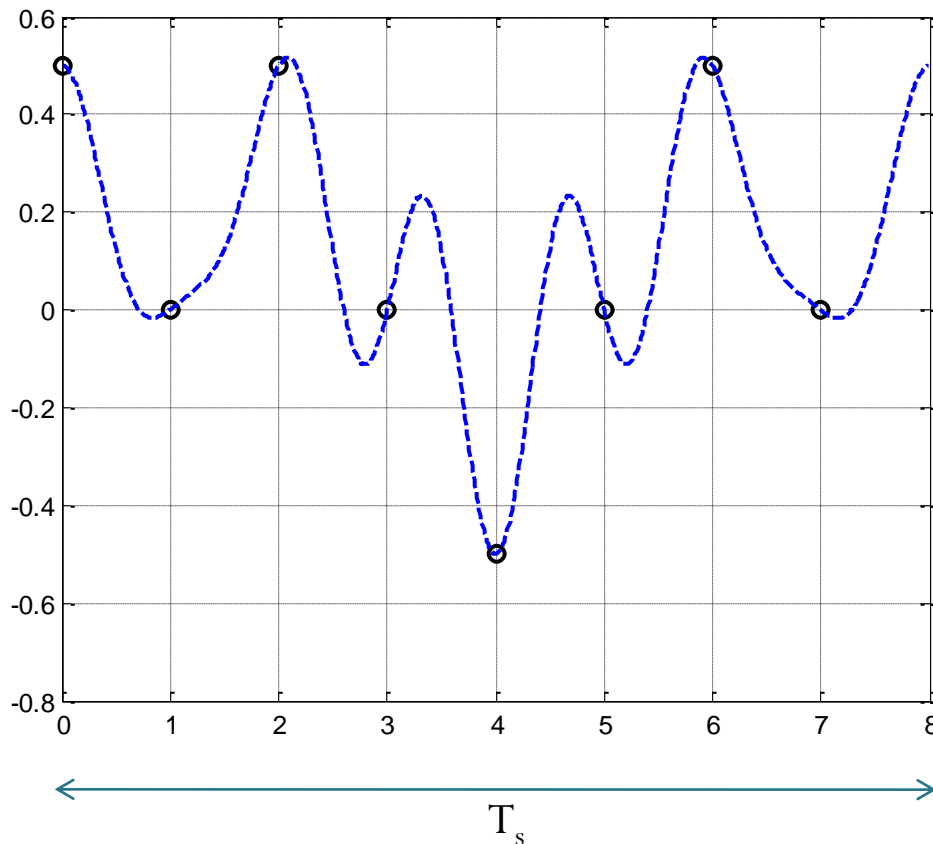
References: E. Oran Brigham, *The Fast Fourier Transform*, Prentice-Hall, 1974.



DFT Samples

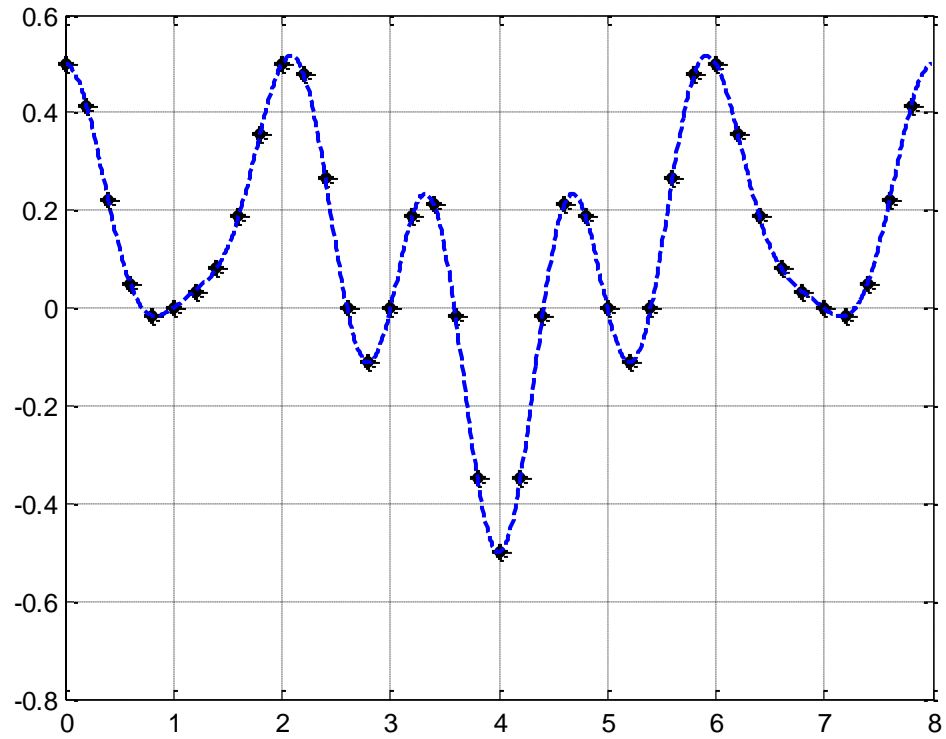
$$s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kt}{T_s}\right), \quad 0 \leq t \leq T_s$$

- Here are the points $s[n]$ on the continuous-time version $s(t)$:



$$\begin{aligned} s[n] &= s\left(n \frac{T_s}{N}\right) \\ &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kn}{N}\right) \\ &= \sqrt{N} \text{IDFT}\{S\}[n] \\ &0 \leq n < N \end{aligned}$$

Oversampling



Oversampling (2)

- Increase the number of sample points from N to LN on the interval $[0, T_s]$.
- L is called the **over-sampling factor**.

$$\begin{array}{ccc}
 \boxed{s[n] = s\left(n \frac{T_s}{N}\right)} & \rightarrow & \boxed{s^{(L)}[n] = s\left(n \frac{T_s}{LN}\right)} \\
 0 \leq n < N & & 0 \leq n < LN
 \end{array}$$

$$\begin{aligned}
 s^{(L)}[n] &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi k}{T_s} n \frac{T_s}{LN}\right) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kn}{LN}\right) \\
 &= \frac{1}{\sqrt{N}} LN \left(\frac{1}{LN} \sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kn}{LN}\right) \right) \\
 &= L\sqrt{N} \left(\frac{1}{LN} \left(\sum_{k=0}^{N-1} S_k \exp\left(j \frac{2\pi kn}{LN}\right) + \sum_{k=N}^{NL-1} 0 \exp\left(j \frac{2\pi kn}{LN}\right) \right) \right) \\
 &= L\sqrt{N} \left(\frac{1}{LN} \sum_{k=0}^{NL-1} \tilde{S}_k \exp\left(j \frac{2\pi kn}{LN}\right) \right) = L\sqrt{N} \text{IDFT}\{\tilde{S}\}[n]
 \end{aligned}$$

Zero padding:

$$\tilde{S}_k = \begin{cases} S_k, & 0 \leq k < N \\ 0, & N \leq k < LN \end{cases}$$

Oversampling: Summary

N points

$$s[n] = s\left(n \frac{T_s}{N}\right) = \sqrt{N} \text{IDFT}\{S\}[n]$$
$$0 \leq n < N$$

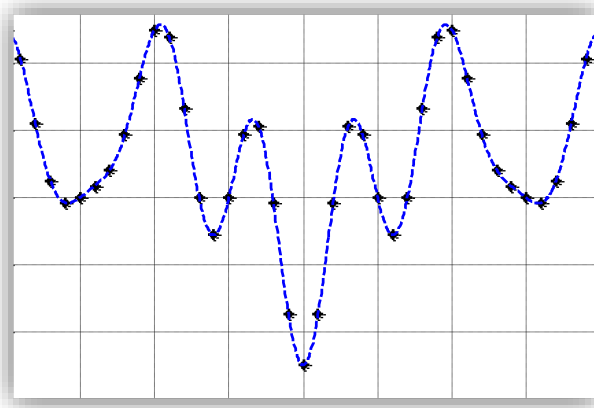
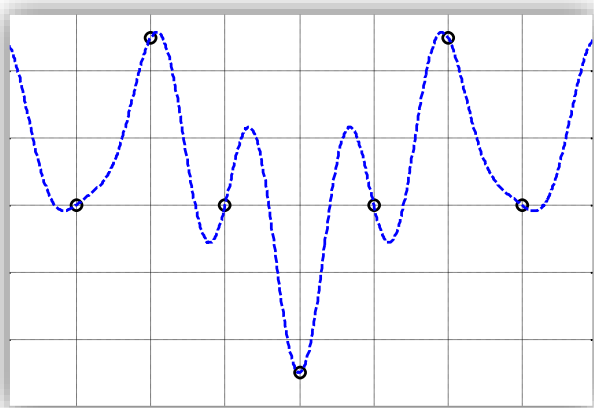


LN points

$$s^{(L)}[n] = s\left(n \frac{T_s}{LN}\right) = L\sqrt{N} \text{IDFT}\{\tilde{S}\}[n]$$
$$0 \leq n < LN$$

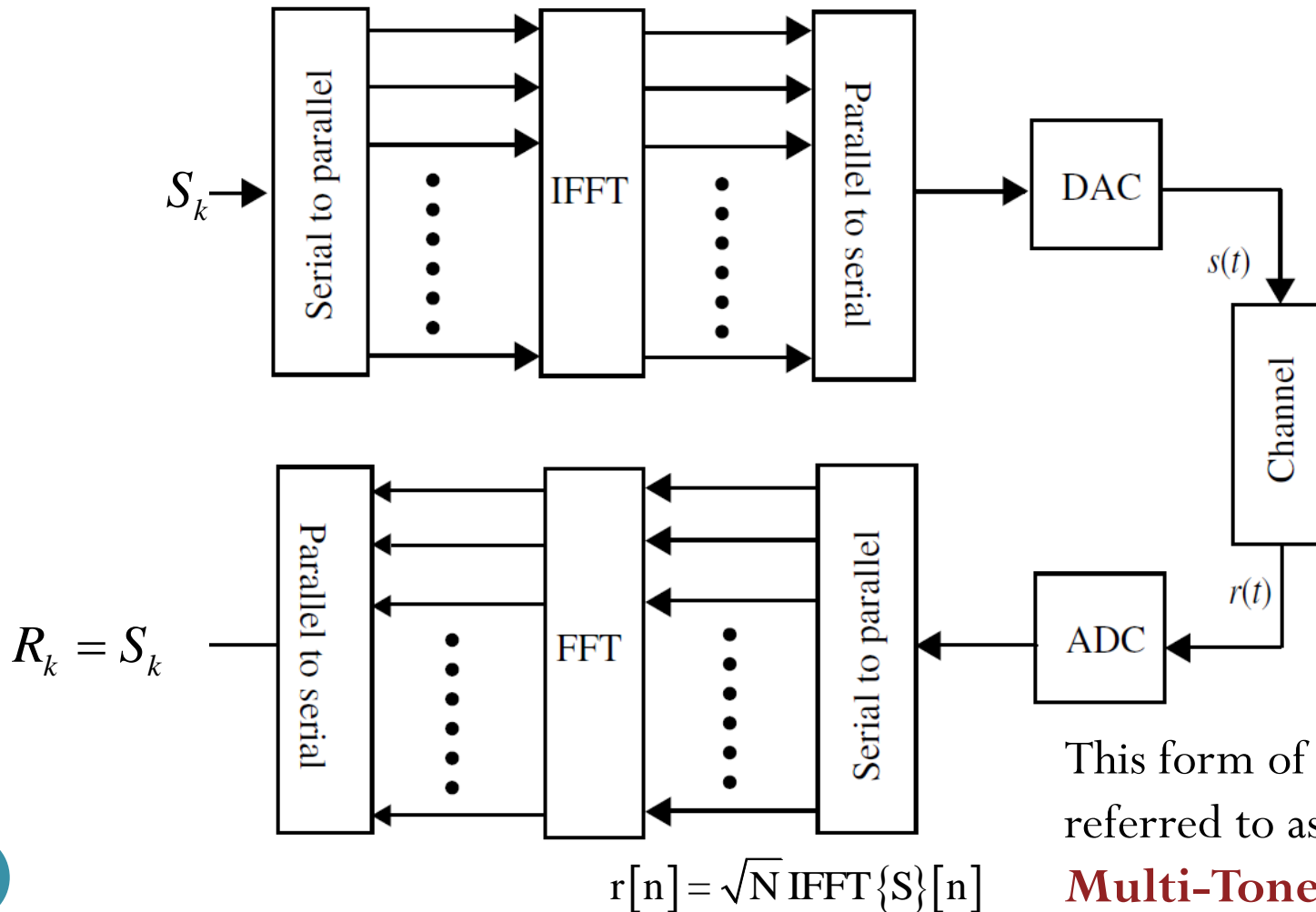
Zero padding:

$$\tilde{S}_k = \begin{cases} S_k, & 0 \leq k < N \\ 0, & N \leq k < LN \end{cases}$$



OFDM implementation by IFFT/FFT

$$s^{(L)}[n] = s\left(n \frac{T_s}{LN}\right) = L\sqrt{N} \text{IFFT}^{(L)}\{\tilde{S}\}[n]$$



This form of OFDM is often referred to as **Discrete Multi-Tone (DMT)**.

OFDM with Memoryless Channel

$$h(t) = \beta\delta(t)$$

[should be $h(t) = \beta\delta(t - \tau)$]

$$r(t) = h(t) * s(t) + w(t) = \beta s(t) + w(t)$$

Additive white Gaussian noise

Sample every T_s/N

$$r[n] = \beta s[n] + w[n]$$

$$s[n] = \sqrt{N} \text{IFFT}\{S\}[n]$$

FFT

$$R_k = \frac{1}{\sqrt{N}} \text{FFT}\{y\}[n] = \beta S_k + \frac{1}{\sqrt{N}} W_k$$

Sub-channel are independent.

(No ICI)

Channel with Finite Memory

Discrete time baseband model:

$$y[n] = \{h * s\}[n] + w[n] = \sum_{m=0}^{\nu} h[m]s[n-m] + w[n]$$

[Tse Viswanath, 2005, Sec. 2.2.3]

where $h[n] = 0$ for $n < 0$ and $n > \nu$

$$w[n] \stackrel{i.i.d.}{\sim} \mathcal{CN}(0, N_0)$$

We will assume that $\nu \ll N$

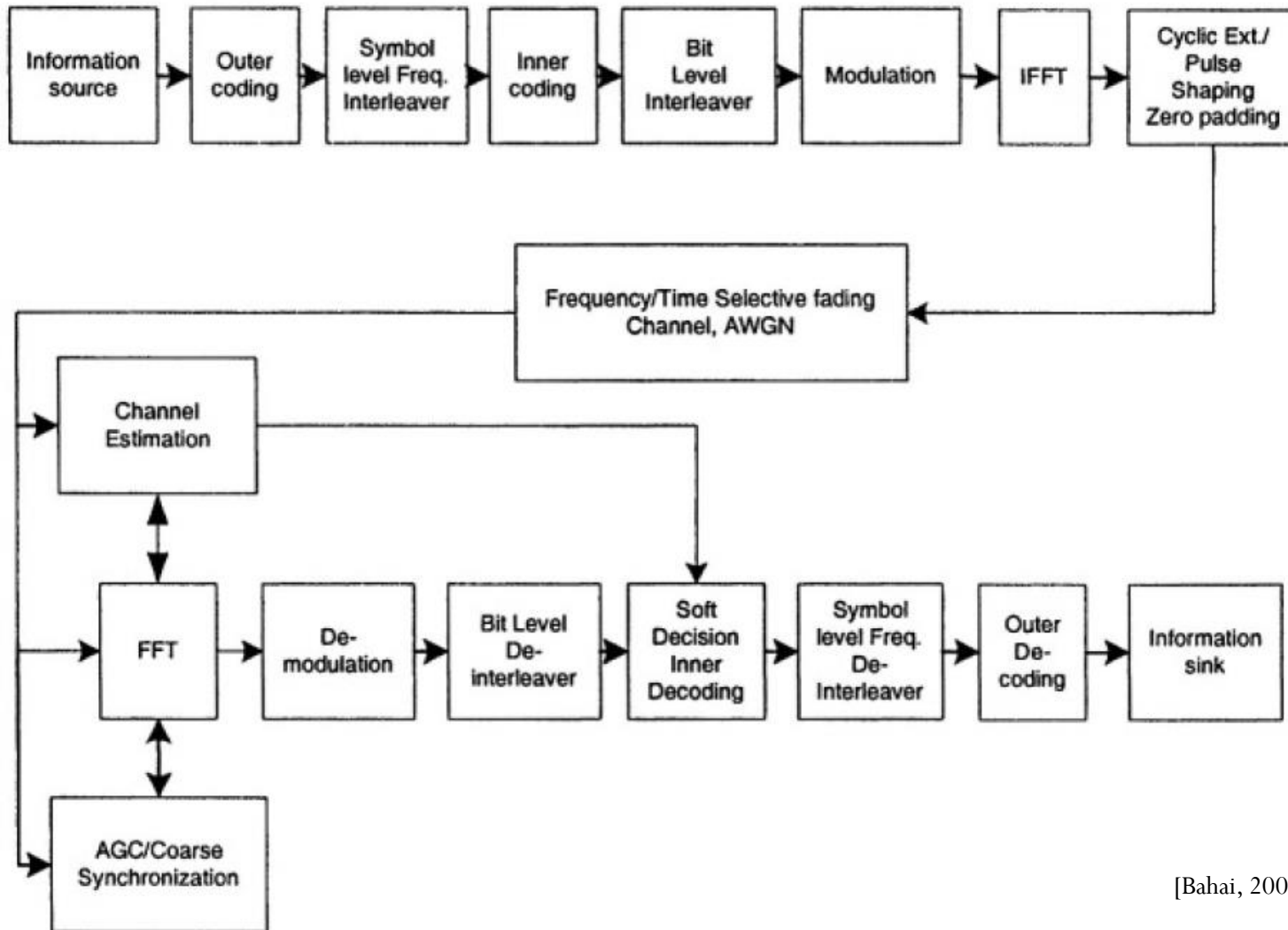
Remarks:

$Z = X + jY$ is a **complex Gaussian** if X and Y are jointly Gaussian.

If X, Y is i.i.d. $\mathcal{N}(0, \sigma^2)$, then $Z = X + jY \sim \mathcal{CN}(0, \sigma_Z^2)$ where $\sigma_Z^2 = 2\sigma^2$ with

$$f_Z(z) = f_{X,Y}(\operatorname{Re}\{z\}, \operatorname{Im}\{z\}) = \frac{1}{\pi\sigma_Z^2} e^{-\frac{|z|^2}{\sigma_Z^2}}.$$

OFDM Architecture



[Bahai, 2002, Fig 1.11]