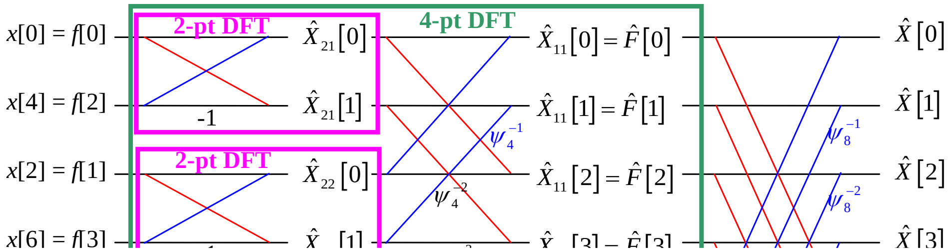


ECS455: Chapter 5

OFDM

5.3 Implementation: DFT and FFT

Dr.Prapun Suksompong

www.prapun.com

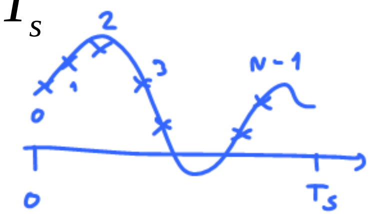
Office Hours:

Library (Rangsit)	Mon	16:20-16:50
BKD 3601-7	Wed	9:20-11:20

Discrete Fourier Transform (DFT)

Transmitter produces

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



Sample the signal in time domain every T_s / N gives

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

n = 0, 1, ..., N-1

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

≡ N IDFT{S}[n]

We can implement OFDM in the discrete domain!

CTCF	FT
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 pth row and qth column is given by $\psi_N^{-(p-1)(q-1)}$:

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

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

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

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

Key Property:

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

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

$\frac{1}{\sqrt{N}}\Psi_N$ 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

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

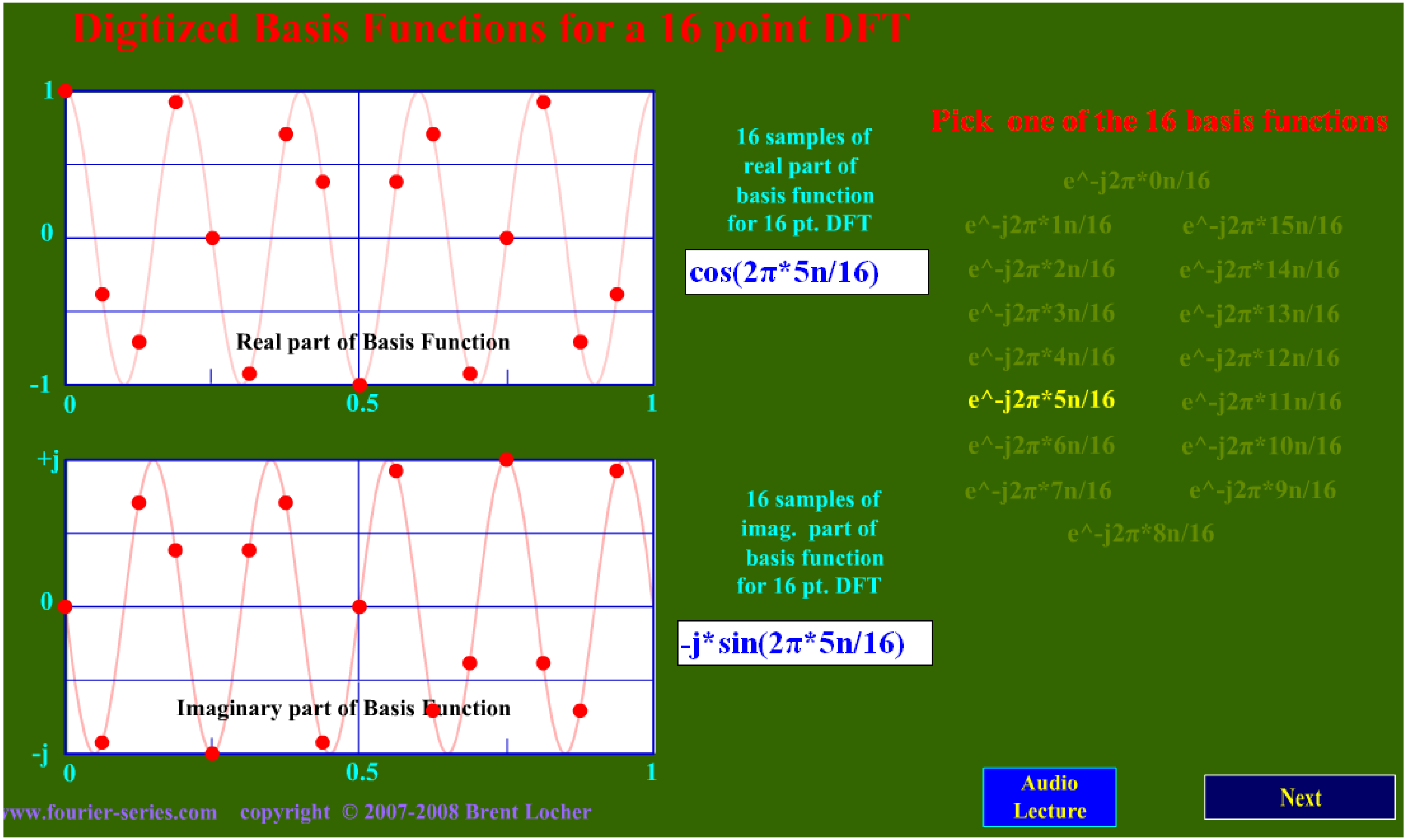
The inverse DFT is given by

$$\underset{0 \le n < N}{x[n]} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \psi_N^{nk} \underset{\mathrm{DFT}^{-1}}{\overset{\mathrm{DFT}}{\rightleftharpoons}} \underset{0 \le k < N}{X[k]} = \sum_{n=0}^{N-1} x[n] \psi_N^{-nk}$$

In matrix form,

$$x = \frac{1}{N} \Psi_N^* X \underset{\mathrm{DFT}^{-1}}{\overset{\mathrm{DFT}}{\rightleftharpoons}} X = \Psi_N \times x.$$

DFT: Example

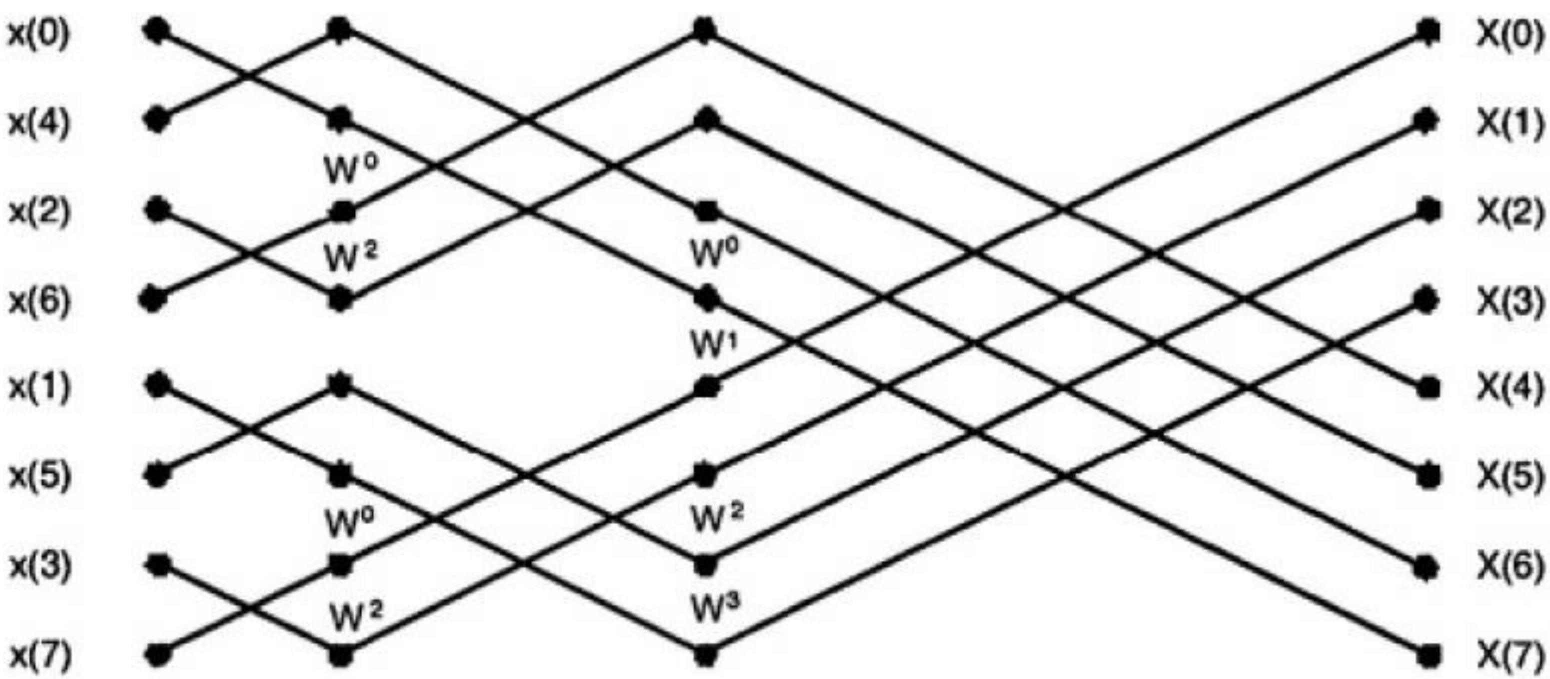


[http://www.fourier-series.com/fourierseries2/DFT_tutorial.html]

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



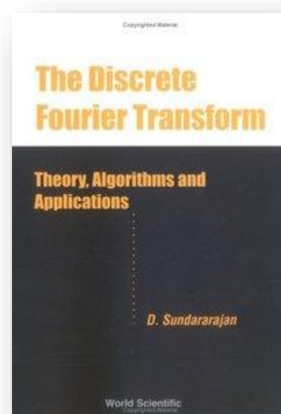
[Bahai, 2002, Fig. 2.9]

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.

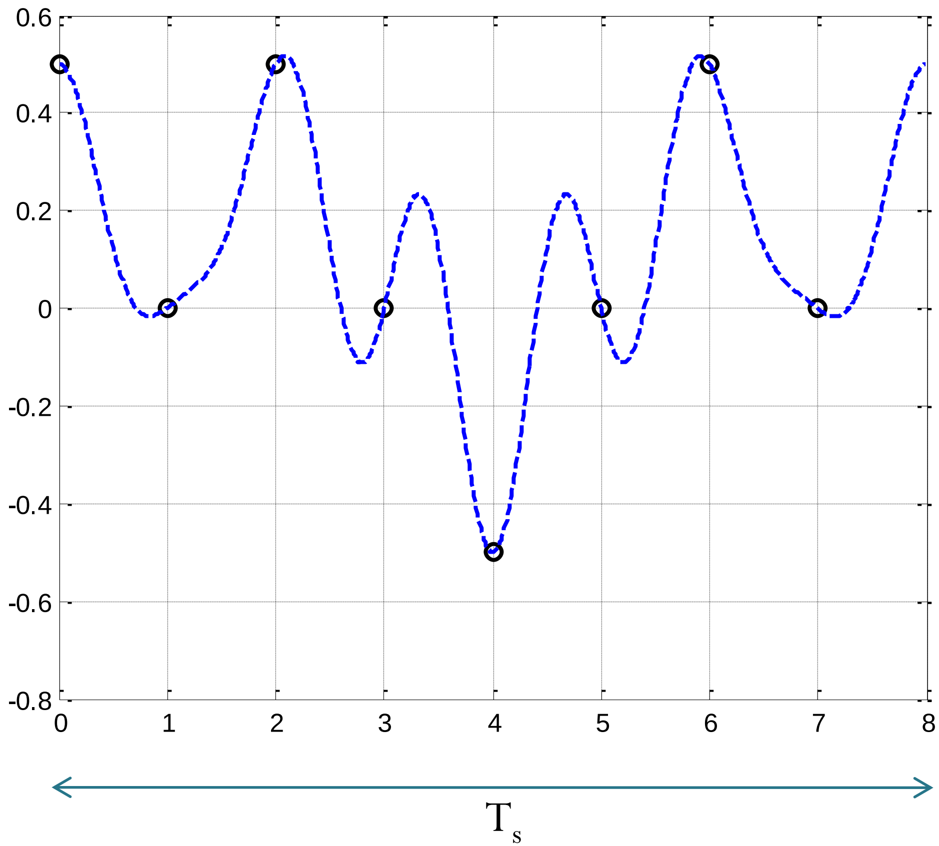
References: E. Oran Brigham, *The Fast Fourier Transform*, Prentice-Hall, 1974.



DFT Samples

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

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



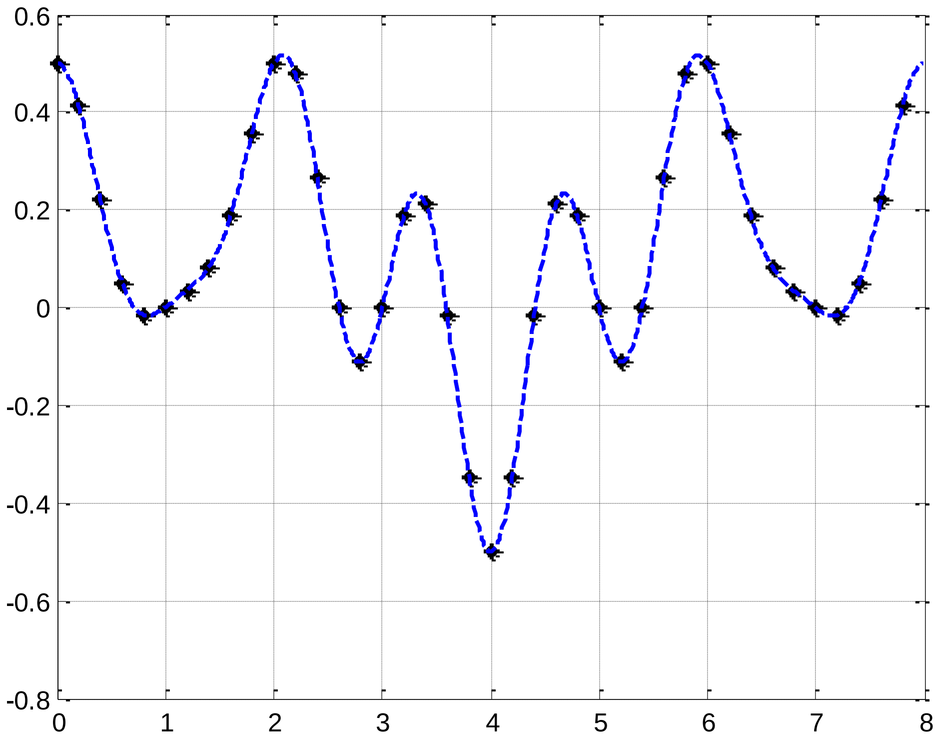
$$s[n] = s\left(n \frac{T_s}{N} \right)$$

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

$$= \sqrt{N}\ \mathrm{IDFT}\{S\}[n]$$

$$0 \le n < N$$

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**.

$$s[n] = s\left(n\frac{T_s}{N}\right) \quad 0 \le n < N$$

$$\Rightarrow$$

$$s^{(L)}[n] = s\left(n\frac{T_s}{LN}\right) \quad 0 \le n < LN$$

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

$$= \frac{1}{\sqrt{N}} LN \left(\frac{1}{LN}\sum_{k=0}^{N-1} S_k \exp\left(j\frac{2\pi kn}{LN}\right)\right)$$

$$= L\sqrt{N}\left(\frac{1}{LN}\left(\sum_{k=0}^{N-1} S_k \exp\left(j\frac{2\pi kn}{LN}\right) + \sum_{k=N}^{NL-1} 0 \exp\left(j\frac{2\pi kn}{LN}\right)\right)\right)$$

$$= L\sqrt{N}\left(\frac{1}{LN}\sum_{k=0}^{NL-1} \tilde{S}_k \exp\left(j\frac{2\pi kn}{LN}\right)\right) = L\sqrt{N}\,\text{IDFT}\{\tilde{S}\}[n]$$

LN

(1) Zero padding:

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

(2) scaling

Oversampling: Summary

N points

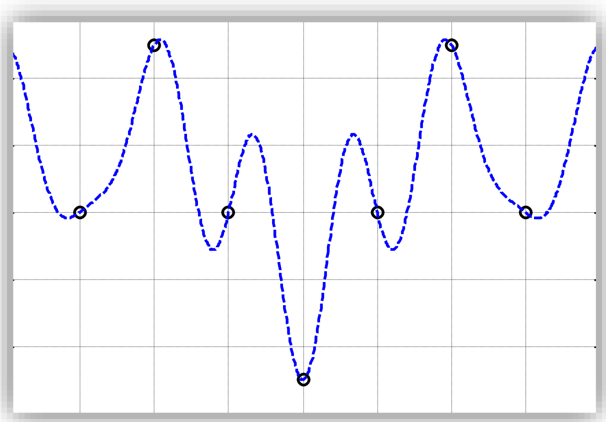
$$s[n] = s\left(n\frac{T_s}{N}\right) = \sqrt{N}\,\mathrm{IDFT}\{S\}[n]$$
$$0 \le n < N$$

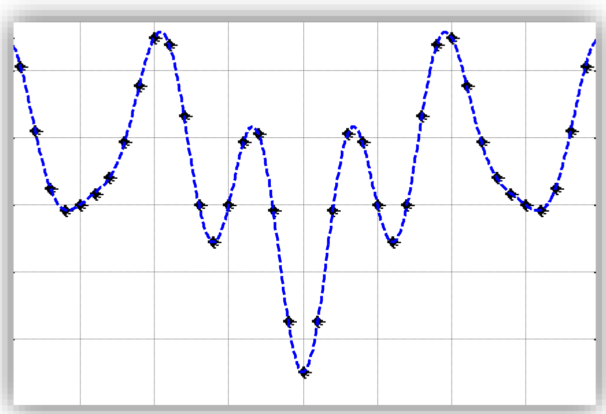
LN points

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

Zero padding:

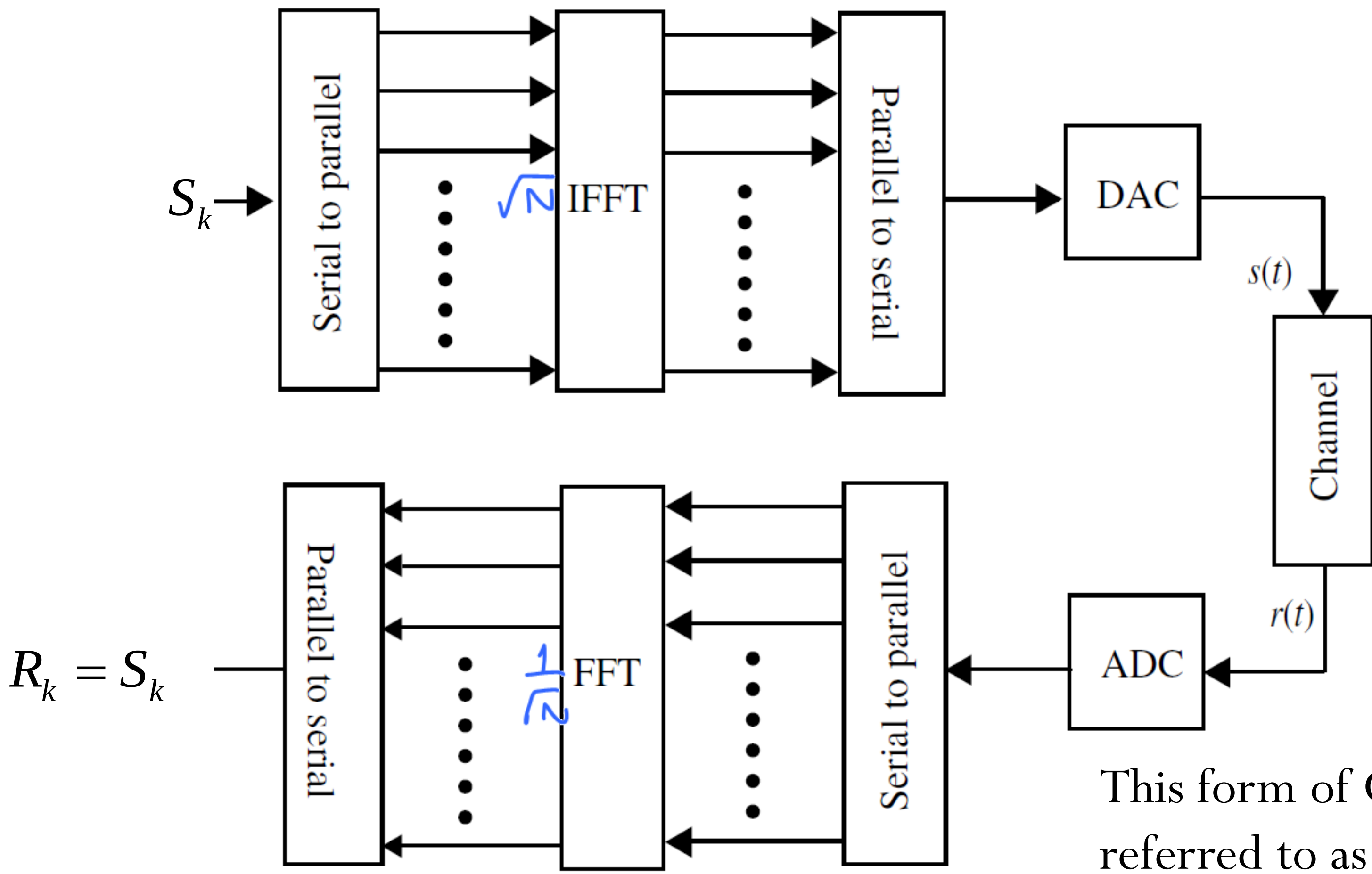
$$\tilde{S}_k = \begin{cases} S_k, & 0 \le k < N \\ 0, & N \le k < LN \end{cases}$$





OFDM implementation by IFFT/FFT

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



$$r[n] = \sqrt{N}\,\mathrm{IFFT}\{S\}[n]$$

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

OFDM with Memoryless Channel

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

$\left[\text{should be } h(t) = \beta\delta(t-\tau)\right]$

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

Additive white Gaussian noise

Sample every T_s/N

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

FFT

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

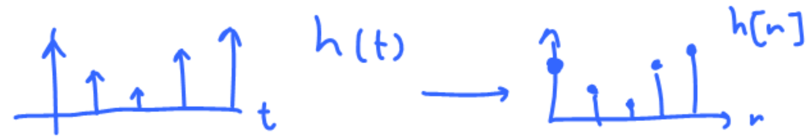
$$R_k = \frac{1}{\sqrt{N}}\text{FFT}\{y\}[n] = \beta S_k + \frac{1}{\sqrt{N}}W_k$$

Sub-channel are independent.

(No ICI)

Channel with Finite Memory

channel memory length



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 $n > \nu$

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

We will assume that $\nu \ll N$

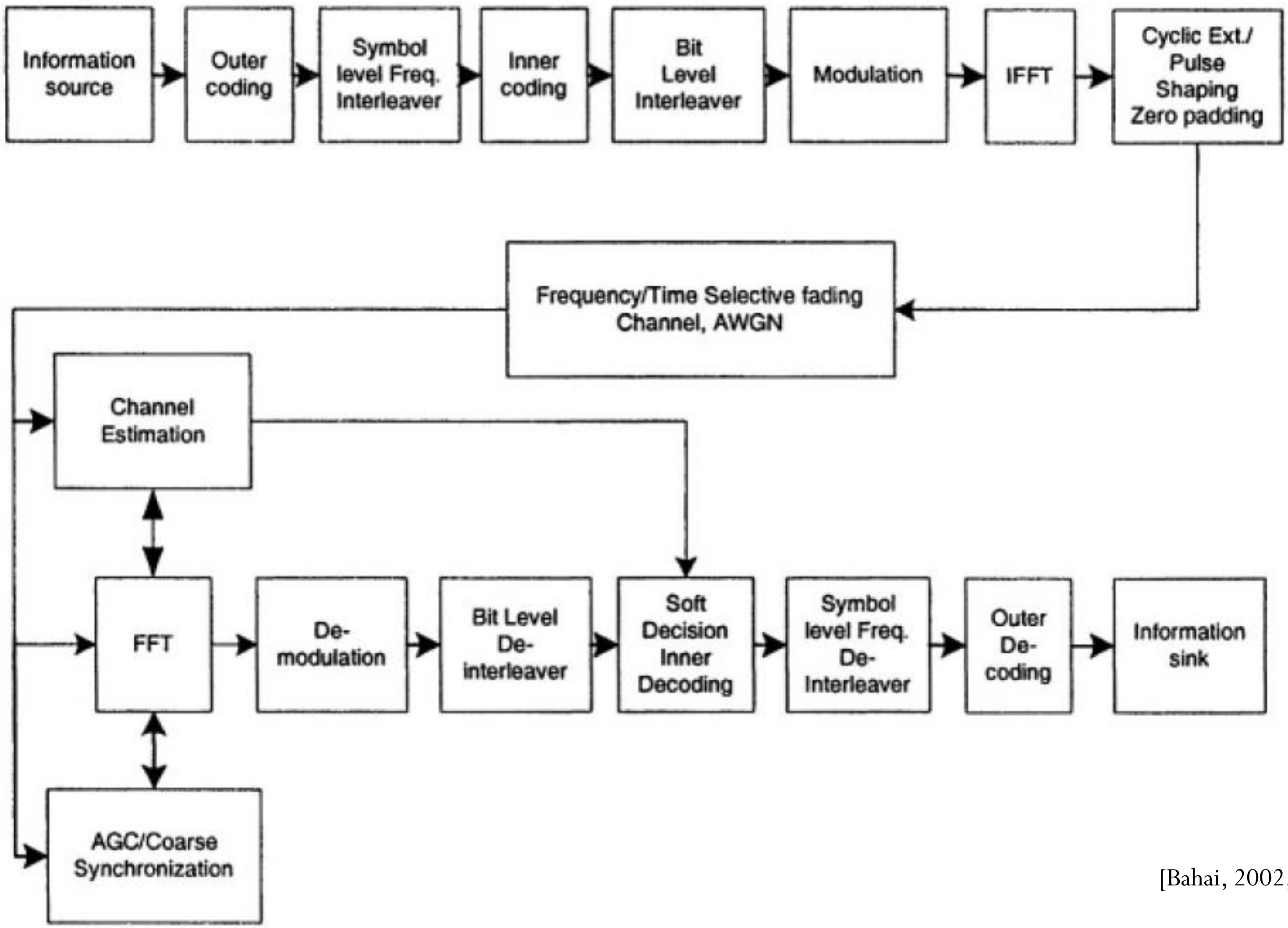
Remarks:

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

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

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

OFDM Architecture



[Bahai, 2002, Fig 1.11]