Intelligent Signal Processing

Fast Fourier Transform

Angelo Ciaramella

Introduction

- The Discrete Fourier Transform (DFT) has an important role for signal analysis

- In the sixties of the last century a fast approach for DFT was introduced
  - Fast Fourier Transform
  - Work of Cooley and Tukey

Classis

- Decimation in time
  - The source signal x(n) is divided in shorter sequences

- Decimation in frequency
  - The DFT coefficients X(k) are divided in shorter sequences

DFT

$$X(k) = \begin{cases} \sum_{n=0}^{N-1} x(n) W_N^{kn} & 0 \le k \le N-1 \\ 0 & \text{altrove} \end{cases}$$

Analysis

$$x(n) = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} & 0 \le n \le N-1 \\ 0 & \text{altrove} \end{cases}$$

Synthesis

$$X(k) = \sum_{n=0}^{N-1} \left\{ \left(\mathrm{Re}[x(n)]\mathrm{Re}\left[W_N^{kn}\right] - \mathrm{Im}[x(n)]\mathrm{Im}\left[W_N^{kn}\right] \right) + j\left(\mathrm{Re}[x(n)]\mathrm{Re}\left[W_N^{kn}\right] + \mathrm{Im}[x(n)]\mathrm{Im}\left[W_N^{kn}\right] \right) \right\}$$

$$k = 0, 1, \ldots, N-1$$

DFT

$$X(k)=\begin{cases}\sum_{n=0}^{N-1} x(n)W_N^{kn} & 0\le k\le N-1\\ 0 & \text{altrove}\end{cases}$$

Analysis

$$x(n)=\begin{cases}\frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn} & 0\le n\le N-1\\ 0 & \text{altrove}\end{cases}$$

Synthesis

$$X(k)=\sum_{n=0}^{N-1}\left\{\left(\mathrm{Re}[x(n)]\mathrm{Re}\left[W_N^{kn}\right]-\mathrm{Im}[x(n)]\mathrm{Im}\left[W_N^{kn}\right]\right)+j\left(\mathrm{Re}[x(n)]\mathrm{Re}\left[W_N^{kn}\right]+\mathrm{Im}[x(n)]\mathrm{Im}\left[W_N^{kn}\right]\right)\right\}$$

$$k=0,1,\ldots,N-1$$

$X(k)$ needs of $4N$ real products and $(4N-1)$ real sums for ech k. Totally, we have $4N^2$ real products e $N(4N-1)$ real sums.

Time decimation

- We use the symmetry and periodicity of the complex exponential

$$W_N^{kn} = e^{-j\left(\frac{2\pi}{N}\right)kn}$$

- The sequence is a power of two

$$N = 2^v$$

Time decimation

- X(k) is calculated dividing x(n) in two subsequences

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi}{N}\right)kn}$$

$$X(k) = \sum_{\substack{n=0 \\ n \text{ even}}}^{N-1} x(n) e^{-j\left(\frac{2\pi}{N}\right)kn} + \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x(n) e^{-j\left(\frac{2\pi}{N}\right)kn}$$

Time decimation

n = 2r   n = 2r+1

$$X(k)=\sum_{r=0}^{N/2-1}x(2r)W_N^{2rk}+\sum_{r=0}^{N/2-1}x(2r+1)W_N^{(2r+1)k}$$

$$=\sum_{r=0}^{N/2-1}x(2r)(W_N^2)^{rk}+W_N^k\sum_{r=0}^{N/2-1}x(2r+1)(W_N^2)^{rk}$$

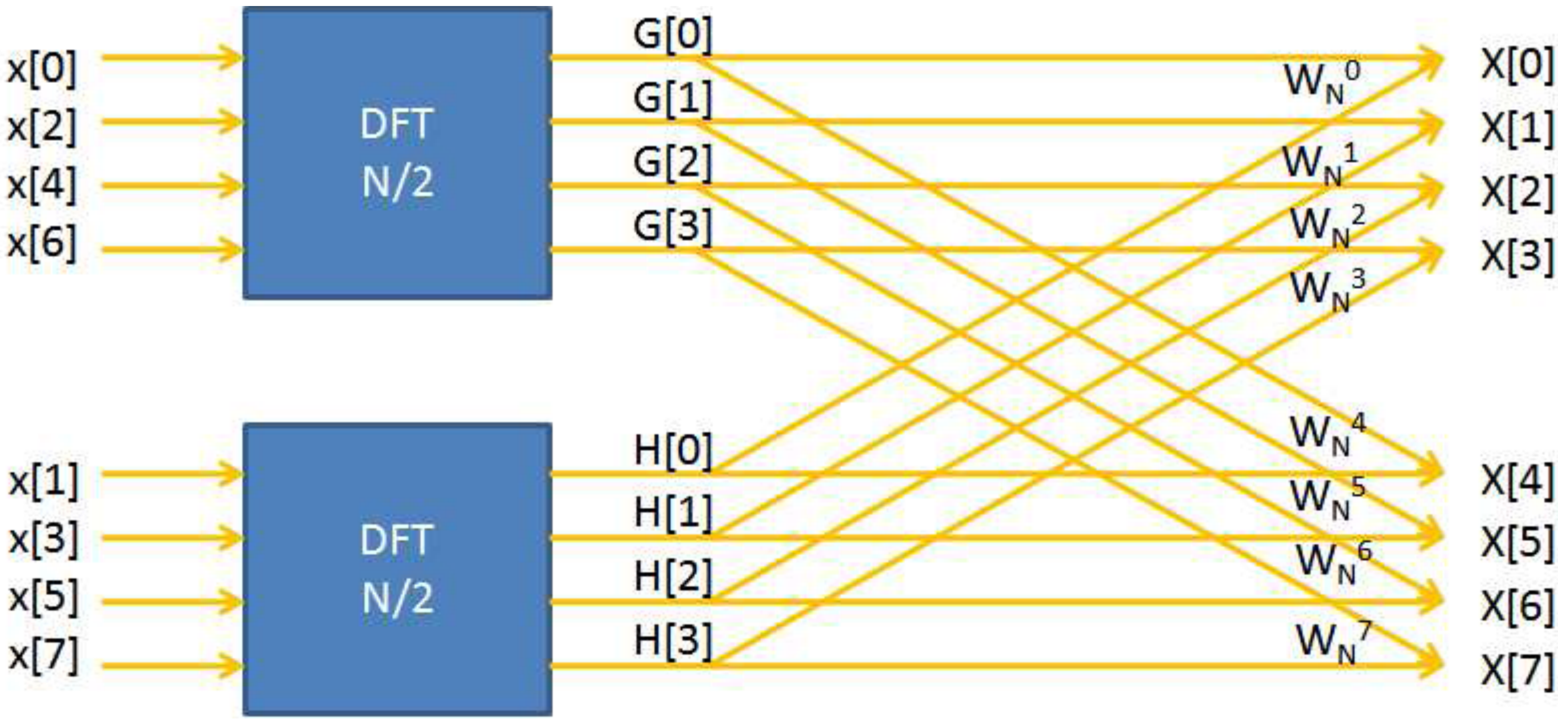
$$W_N^2=e^{-2j(2\pi/N)}=$$

$$e^{-j2\pi/(N/2)}=W_{N/2}$$

$$X(k)=\sum_{r=0}^{N/2-1}x(2r)W_{N/2}^{rk}+W_N^k\sum_{r=0}^{N/2-1}x(2r+1)W_{N/2}^{rk}$$

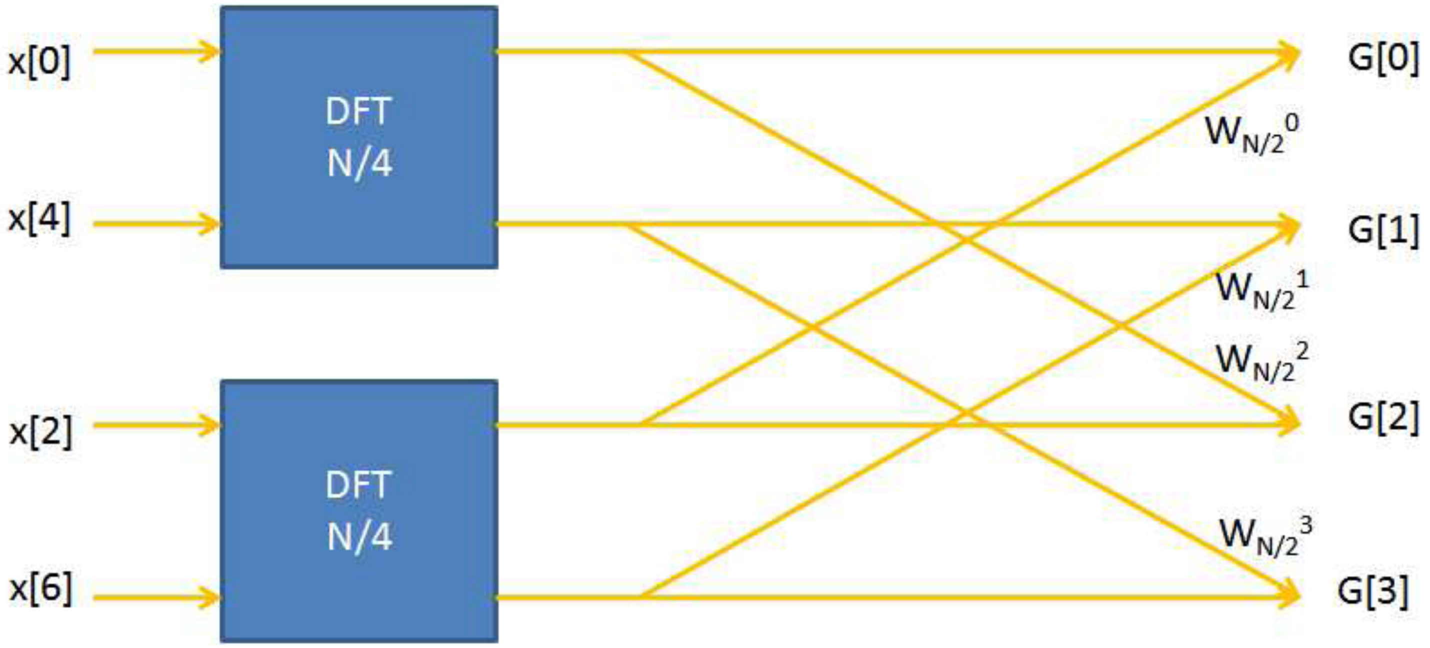
$$=G(k)+W_N^kH(k)$$

Flow graph



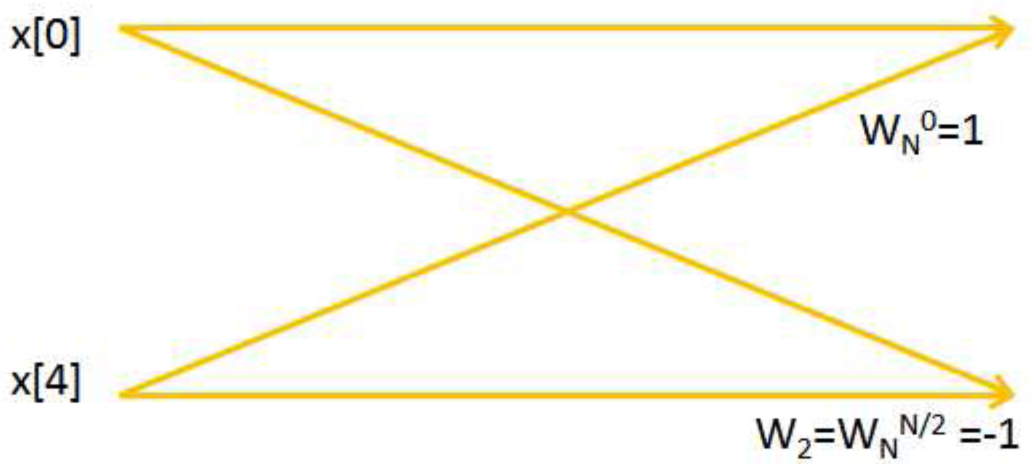
Flow Graph for a DFT with N=8

Flow graph



Flow Graph for a DFT with *N*=4

Flow graph



Flow Graph for a DFT with $N=2$

FFT algorithm

- The FFT is obtained by a recursive algorithm based on a divide-et-impera strategy

- Fourier coefficients

$$X[k] = \sum_{j=0}^{N/2-1} x[j] W_N^{kj}$$

$$x = (x[0], x[1], \ldots, x[N-1])$$

FFT algorithm

```
FFT-Ricorsiva(x)
   N = length(x);
   if N==1
      then return x[0]
   WN = exp(j  2 PI/N)
   W = 1
   Xp = FFT-Ricorsiva([x[0], x[2], x[4], . . . , x[N-2]])
   Xd = FFT-Ricorsiva([x[1], x[3], x[5], . . . , x[N-1]])
   for k = 0 to N/2 - 1
         do
            X[k] = Xp + W Xd
            X[k+N/2] = Xp - W Xd
            W = W W_N
     return X
```

Time complexity

- The asymptotic time complexity is

$$T(N) = 2T(N/2) + \Theta(N) = \Theta(N \log N)$$

- It is the same also for the inverse transform

Convolution theorem

- A faster convolution can be obtained

$$a * b = \mathbf{DFT}_{2N}^{-1}\left(\mathbf{DFT}_{2N}(a)\mathbf{DFT}_{2N}(b)\right)$$

zero padding