Lecture 6
Discrete Fourier Transform (DFT) and its Fast Implementation

DFT Definition

DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}}, \qquad k = 0, 1, \dots, N-1$$

$$= \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}\Big|_{\omega=\frac{2\pi k}{N}} = X(e^{j\omega})\Big|_{\omega=2\pi k/N}$$

Using: $W_N = e^{-j2\pi/N}$, we can rewrite the DFT as

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0, 1, \dots, N-1.$$

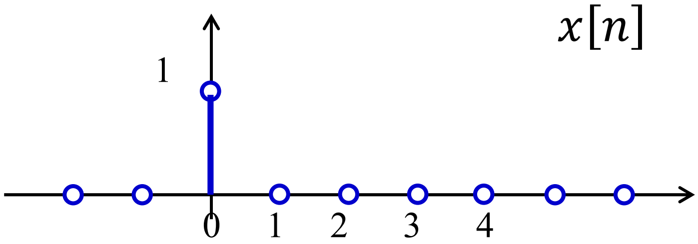
IDFT:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \qquad n = 0, 1, \dots, N-1.$$

DTFT vs. DFT

- Both apply to discrete time signal
- DTFT is for infinite length of discrete time signal
- DFT is for finite length of discrete time signal
- For a length-N sequence, N values of $X\left(e^{j\omega}\right)$, at N distinct frequency points, $\omega = \omega_k$, $k = 0, 1, \ldots, N-1$, are sufficient to determine $x[n]$, uniquely.
- Q: Can we reconstruct the DTFT spectrum (continuous in ω) from the DFT?

Example 1



$x[n] \qquad 0 \le n \le N-1$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0, 1, \ldots, N-1$$

$$= x[0] W_N^0 = 1$$

N point DFT

Example 2

- Consider a length N sequence defined for $0 \le n \le N{-}1$

$$g[n] = \cos\left(\frac{2\pi r}{N} n\right), \qquad 0 \le r \le N-1$$

- **Solution:** Since

$$g[n] = \frac{1}{2}\left(e^{\frac{j2\pi r}{N}n} + e^{-\frac{j2\pi r}{N}n}\right) = \frac{1}{2}\left(W_N^{-rn} + W_N^{rn}\right)$$

- Thus,

$$G[k] = \sum_{n=0}^{N-1} g[n] W_N^{kn} = \frac{1}{2}\sum_{n=0}^{N-1}\left(W_N^{-(r-k)n} + W_N^{(r+k)n}\right)$$

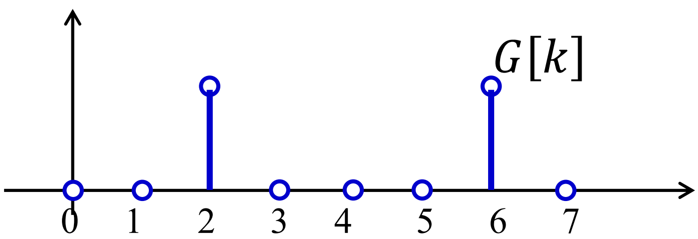
$$G[k] = \frac{1}{2}\sum_{n=0}^{N-1}\left(W_N^{-(r-k)n} + W_N^{(r+k)n}\right)$$

- Making use of the identity:

$$\sum_{n=0}^{N-1} W_N^{-(k-l)n} = \begin{cases} N, & \text{for } k-l = mN, m \text{ an integer} \\ 0, & \text{otherwise} \end{cases}$$

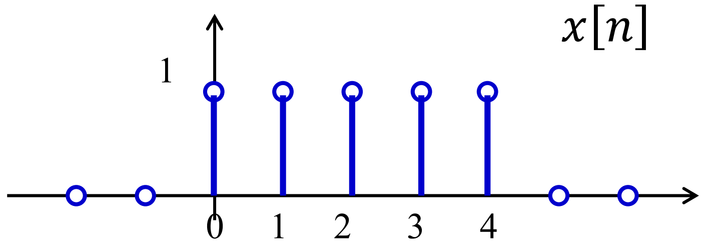
- We get

$$G[k] = \begin{cases} N/2, & k = r \\ N/2, & k = N-r \\ 0, & \text{otherwise} \end{cases} \qquad 0 \le k \le N-1$$



for $N = 8$, $r = 2$

Example 3



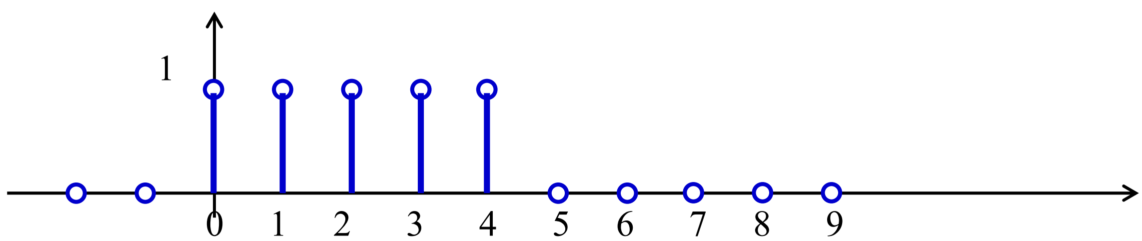
- Take N=5

$$X[k] = \begin{cases} \sum_{n=0}^{4} x[n] W_5^{kn}, & k = 0, 1, 2, 3, 4 \\ 0 & \text{otherwise} \end{cases}$$

$$= 5\delta[k]$$

5 point DFT

- Q: What if we take N = 10?

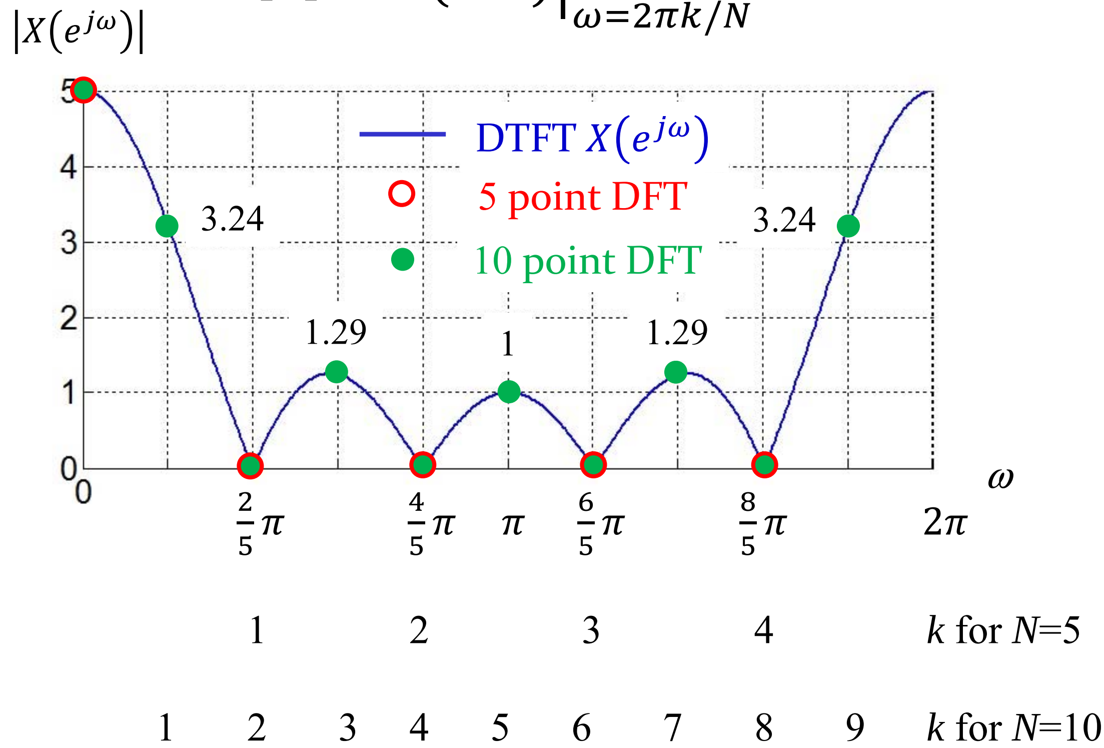


$$X[k] = \begin{cases} \sum_{n=0}^{4} x[n] W_{10}^{kn}, & k = 0, 1, 2, \ldots, 9 \\ 0 & \text{otherwise} \end{cases}$$

10 point DFT

$$= \begin{cases} e^{-j\frac{4\pi}{10}k} \dfrac{\sin\left(\dfrac{\pi k}{2}\right)}{\sin\left(\dfrac{\pi k}{10}\right)} & k = 0, 1, 2, \ldots, 9 \\ 0 & \text{otherwise} \end{cases}$$

$$X[k] = X(e^{j\omega})\Big|_{\omega=2\pi k/N}$$



Four Types of Fourier Transform

Time Domain	Non-Periodic	Periodic	
Continuous	Continuous Time Fourier Transform (CTFT)	Fourier Series (FS)	Non-Periodic
Discrete	Discrete Time Fourier Transform (DTFT)	Discrete Fourier Transform (DFT)	Periodic
	Continuous	Discrete	Frequency Domain

DTFT vs. DFT

- Q: Can we reconstruct the DTFT spectrum (continuous in ω) from the DFT?

$$x[n] \xrightarrow{DFT} X[k] \xrightarrow{?} X(e^{j\omega})$$

- A: Since the N-length signal $x[n]$ can be exactly recovered from both the DFT coefficients and the DTFT spectrum, we expect that the DTFT spectrum (that is within $[0, 2\pi]$) can be exactly reconstructed by the DFT coefficients.

Reconstruct DTFT from DFT

- By substituting the inverse DFT into the $x[n]$, we have

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}$$

$$= \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N} \right] e^{-j\omega n}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \boxed{\sum_{n=0}^{N-1} e^{j2\pi kn/N} e^{-j\omega n}}$$

Sum of a geometric sequence with $q = e^{-j(\omega - 2\pi k/N)}$

$$\sum_{n=0}^{N-1} e^{j2\pi kn/N} e^{-j\omega n} = \frac{1-e^{-j(\omega N-2\pi k)}}{1-e^{-j(\omega-2\pi k/N)}}$$

$$= \frac{1-e^{-j(\omega N-2\pi k)}}{1-e^{-j(\omega N-2\pi k)/N}} \boxed{\times \frac{e^{j\frac{\omega N-2\pi k}{2}}}{e^{j\frac{\omega N-2\pi k}{2N}}} \times \frac{e^{-j\frac{\omega N-2\pi k}{2}}}{e^{-j\frac{\omega N-2\pi k}{2N}}}}$$

$$= \frac{e^{-j\frac{\omega N-2\pi k}{2}}}{e^{-j\frac{\omega N-2\pi k}{2N}}} \times \frac{e^{j\frac{\omega N-2\pi k}{2}} - e^{-j\frac{\omega N-2\pi k}{2}}}{e^{j\frac{\omega N-2\pi k}{2N}} - e^{-j\frac{\omega N-2\pi k}{2N}}}$$

$$= \frac{\sin\left(\frac{\omega N-2\pi k}{2}\right)}{\sin\left(\frac{\omega N-2\pi k}{2N}\right)} e^{-j\left(\omega-\frac{2\pi k}{N}\right)\left(\frac{N-1}{2}\right)}$$

- Therefore,

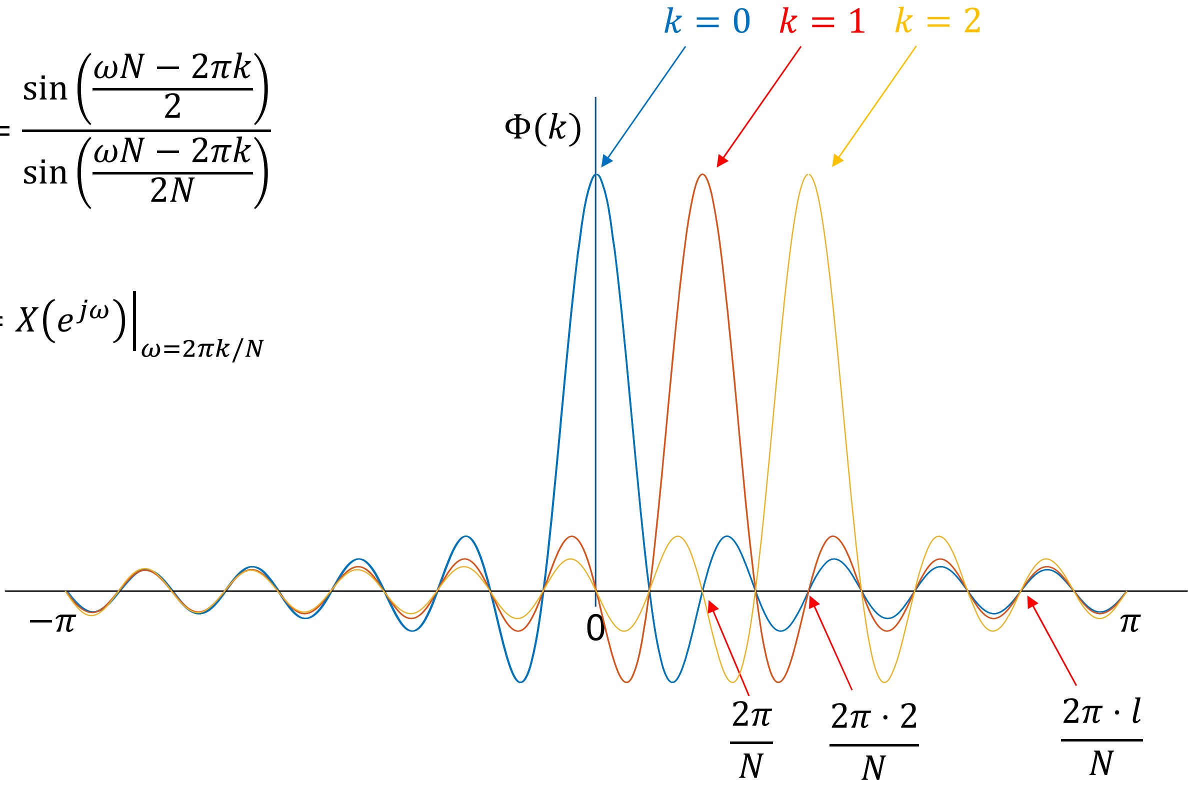
$$X(e^{j\omega}) = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \underbrace{\boxed{\frac{\sin\left(\frac{\omega N-2\pi k}{2}\right)}{\sin\left(\frac{\omega N-2\pi k}{2N}\right)}}}_{\Phi} e^{-j\left(\omega-\frac{2\pi k}{N}\right)\left(\frac{N-1}{2}\right)}$$

- When $\omega = \frac{2\pi l}{N}$ for $0 \le l \le N-1$, if $l = k$, $\Phi = 1$, and if $l \ne k$, $\Phi = 0$. Therefor, $X(e^{j\omega})\big|_{\omega=\frac{2\pi l}{N}} = X[l]$.
- To recover the DTFT $X(e^{j\omega})$ of a length-N sequence $x[n]$, for $n = 0, 1, \ldots, N-1$ from a K-point DFT sequence $X[k]$, for $k = 0, 1, \ldots, K-1$, K must be $\ge N$

$$X(e^{j\omega}) = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\frac{\sin\left(\frac{\omega N - 2\pi k}{2}\right)}{\sin\left(\frac{\omega N - 2\pi k}{2N}\right)} e^{-j\left(\omega - \frac{2\pi k}{N}\right)\left(\frac{N-1}{2}\right)}$$

$$\Phi(k) = \frac{\sin\left(\frac{\omega N - 2\pi k}{2}\right)}{\sin\left(\frac{\omega N - 2\pi k}{2N}\right)}$$

$$X[k] = X(e^{j\omega})\Big|_{\omega = 2\pi k/N}$$



Sampling the DTFT

- Consider a length M sequence $x[n]$ $(0 \le n \le M-1)$ going through the following transforms and operations:

$$x[n] \xrightarrow{\text{DTFT}} X(e^{j\omega}) \xrightarrow[\ ]{\substack{\text{Sample} \\ N \text{ points}}} Y[k] \xrightarrow{\substack{N-\text{point} \\ \text{IDFT}}} y[n]$$

Find the relation between $x[n]$ and $y[n]$.

- Since

$$Y[k] = X(e^{j\omega_k}) = X(e^{j(2\pi k/N)}) = \sum_{l=-\infty}^{\infty} x[l]\, W_N^{kl}$$

$$\text{for } 0 \le k \le N-1$$

- And

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} Y[k] W_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=-\infty}^{\infty} x[l]\, W_N^{kl}\, W_N^{-kn}$$

$$= \sum_{l=-\infty}^{\infty} x[l]\left[\frac{1}{N}\sum_{k=0}^{N-1} W_N^{-k(n-l)}\right], \qquad \text{for } 0 \le n \le N-1$$

- Recalling from the identity that

$$\sum_{k=0}^{N-1} W_N^{-k(n-l)} = \begin{cases} N, & \text{for } l = n + mN, m \text{ is any integer} \\ 0, & \text{otherwise} \end{cases}$$

we have

$$y[n] = \sum_{m=-\infty}^{\infty} x[n+mN], \qquad 0 \le n \le N-1$$

- This relation indicates:
  - $y[n]$ is obtained from $x[n]$ by adding an infinite number of shifted replicas of $x[n]$, with each replica shifted by an integer multiple of N sampling instants, and observing the sum only for the interval $0 \le n \le N-1$.
  - If $M \le N$, then $y[n] = x[n]$ for $0 \le n \le N-1$, and $x[n]$ can be recovered from $y[n]$ by extracting M samples of $y[n]$ for $0 \le n \le M-1$.
  - If $M > N$, there is a time-domain aliasing of samples of $x[n]$ in generating $y[n]$, and $x[n]$ cannot be recovered from $y[n]$.

Example

- Let $\{x[n]\} = \{0,1,2,3,4,5\}$ for $0 \le n \le 5$
- By sampling the DTFT of $x[n]$ by 8 samples at $\omega_k = 2\pi k/8,\ \ 0 \le k \le 7$, and then applying an 8-point IDFT to these samples, we arrive at the sequence
  $$y[n] = x[n] + x[n+8] + x[n-8] + \cdots, 0 \le n \le 7$$
  that is, $\{y[n]\} = \{0,1,2,3,4,5,0,0\}$, for $0 \le n \le 7$
  $x[n]$ can be recovered from $y[n]$
- By sampling the DTFT of $x[n]$ by 4 samples, we arrive
  $$y[n] = x[n] + x[n+4] + x[n-4] + \cdots, 0 \le n \le 3$$
  that is, $\{y[n]\} = \{4,6,2,3\}$, for $0 \le n \le 3$
  $x[n]$ cannot be recovered from $y[n]$

DFT and Inverse DFT

- Both computed similarly.....let's play

$$Nx^*[n] = N\left(\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}\right)^*$$
$$= \sum_{k=0}^{N-1} X^*[k]W_N^{kn} = \text{DFT}\{X^*[k]\}$$

- Also,

$$Nx^*[n] = N(\text{ IDFT }\{X[k]\})^*$$

DFT and Inverse DFT

- So,

$$\mathrm{DFT}\{X^*[k]\} = N(\,\mathrm{IDFT}\,\{X[k]\})^*$$

- Or,

$$\mathrm{IDFT}\,\{X[k]\} = \frac{1}{N}(\mathrm{DFT}\,\{X^*[k]\})^*$$

- Implement IDFT by:
  - Take complex conjugate
  - Take DFT
  - Multiply by $1/N$
  - Take complex conjugate

Why useful?

DFT as Matrix Operator

Let

$$\mathbf{W}_N = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi\frac{1}{N}} & e^{-j2\pi\frac{2}{N}} & e^{-j2\pi\frac{3}{N}} & \cdots & e^{-j2\pi\frac{N-1}{N}} \\ 1 & e^{-j2\pi\frac{2}{N}} & e^{-j2\pi\frac{4}{N}} & e^{-j2\pi\frac{6}{N}} & \cdots & e^{-j2\pi\frac{2(N-1)}{N}} \\ 1 & e^{-j2\pi\frac{3}{N}} & e^{-j2\pi\frac{6}{N}} & e^{-j2\pi\frac{9}{N}} & \cdots & e^{-j2\pi\frac{3(N-1)}{N}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi\frac{N-1}{N}} & e^{-j2\pi\frac{2(N-1)}{N}} & e^{-j2\pi\frac{3(N-1)}{N}} & \cdots & e^{-j2\pi\frac{(N-1)(N-1)}{N}} \end{bmatrix}$$

- The (n, k)-th element of $\mathbf{W}_N$ is W_N^{nk}

DFT

$$\mathbf{W}_N \cdot \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} = \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix}$$

IDFT

$$\frac{1}{N} \cdot \mathbf{W}_N^* \cdot \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}$$

Straightforward implementation requires N^2 complex multiplies :-(

DFT as Matrix Operator Cont.

- Can write compactly as:

$$\boldsymbol{X} = \mathbf{W}_N \boldsymbol{x}$$

$$\boldsymbol{x} = \frac{1}{N} \mathbf{W}_N^* \boldsymbol{X}$$

- So

$$\boldsymbol{x} = \frac{1}{N} \mathbf{W}_N^* \boldsymbol{X} = \frac{1}{N} \mathbf{W}_N^* \mathbf{W}_N \boldsymbol{x} = \frac{1}{N} (N\mathbf{I}_N)\, \boldsymbol{x} = \boldsymbol{x}$$

Why

As expected

Circular Time-Reversal

- let $x[n]$ is defined for the range of $0 \le n \le N - 1$
- Time-reversed sequence $x_1[n] = x[-n]$ is no longer defined for the range of $0 \le n \le N - 1$
- Define the **circular time-reversal** $y[n]$

$$y[n] = x[\langle -n \rangle_N]$$

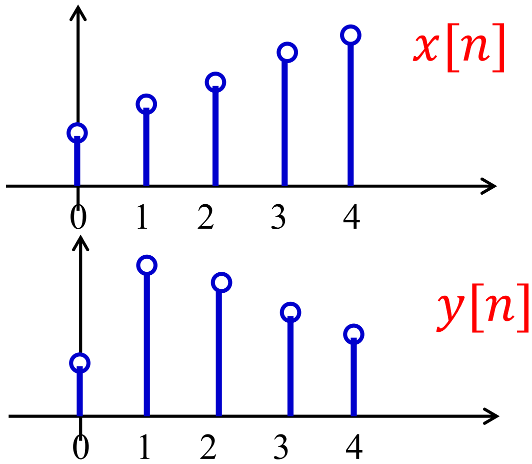
where $\langle k \rangle_N = k$ modulo N.

  - Mathematically, if we let $r = \langle k \rangle_N$, then $r = k + lN$, where l is an integer chosen to make $k + lN$ an integer between 0 and $N-1$.

Example

- $\{x[n]\} = \{x[0], x[1], x[2], x[3], x[4]\}$
- Then the circular reversed sequence $\{y[n]\}$ is given by:

$$y[n] = x[\langle -n\rangle_N]$$
$$y[0] = x[\langle -0\rangle_5] = x[0],$$
$$y[1] = x[\langle -1\rangle_5] = x[4],$$
$$y[2] = x[\langle -2\rangle_5] = x[3],$$
$$y[3] = x[\langle -3\rangle_5] = x[2],$$
$$y[4] = x[\langle -4\rangle_5] = x[1],$$



- Hence, $\{y[n]\} = \{x[0], x[4], x[3], x[2], x[1]\}$

Symmetry of Finite-Length Sequence

- Circular Conjugate Symmetry
  - An N-point sequence is said to be **circular conjugate symmetric sequence** if
$$x[n] = x^*[\langle -n\rangle_N] = x^*[\langle N-n\rangle_N]$$
  - An N-point sequence is said to be **circular conjugate anti-symmetric sequence** if
$$x[n] = -x^*[\langle -n\rangle_N] = -x^*[\langle N-n\rangle_N]$$

- An arbitrary N-point complex sequence can be expressed as the sum of the circular Conjugate Symmetric and circular conjugate anti-symmetric parts, i.e.,

$$x[n] = x_{cs}[n] + x_{ca}[n], \qquad 0 \le n \le N-1$$

where,

$$x_{cs}[n] = \frac{1}{2}(x[n] + x^*[\langle -n\rangle_N]), \qquad 0 \le n \le N-1,$$
$$x_{ca}[n] = \frac{1}{2}(x[n] - x^*[\langle -n\rangle_N]), \qquad 0 \le n \le N-1,$$

are the **circular conjugate-symmetric** and **circular conjugate-antisymmetric** parts, respectively.

Example

- Q: Find the circular conjugate-symmetric and circular conjugate-antisymmetric parts of

$$\{u[n]\} = \{\underset{\uparrow}{1} + j4, -2 + j3, 4 - j2, -5 - j6\}$$

- A: We first form its complex conjugate sequence

$$\{u^*[n]\} = \{\underset{\uparrow}{1} - j4, -2 - j3, 4 + j2, -5 + j6\}$$

Then compute $\{u^*[\langle -n\rangle_4]\}$:

$$u^*[\langle -0\rangle_4] = u^*[0], \qquad u^*[\langle -1\rangle_4] = u^*[3]$$
$$u^*[\langle -2\rangle_4] = u^*[2], \qquad u^*[\langle -3\rangle_4] = u^*[1]$$

Hence, $\{u^*[\langle -n\rangle_4]\} = \{\underset{\uparrow}{1} - j4, -5 + j6, 4 + j2, -2 - j3\}$

$$\{x_{cs}[n]\} = \{\underset{\uparrow}{1}, -3.5 + j4.5, 4, -3.5 - j4.5\}$$

$$\{x_{ca}[n]\} = \{\underset{\uparrow}{j4}, 1.5 - j1.5, -j2, -1.5 - j1.5\}$$

Symmetry of Finite-Length Sequence

- Geometric Symmetry
  - A length-N symmetric sequence
    $$x[n] = x[N-1-n], \qquad 0 \le n \le N-1$$
  - A length-N antisymmetric sequence
    $$x[n] = -x[N-1-n], \qquad 0 \le n \le N-1$$

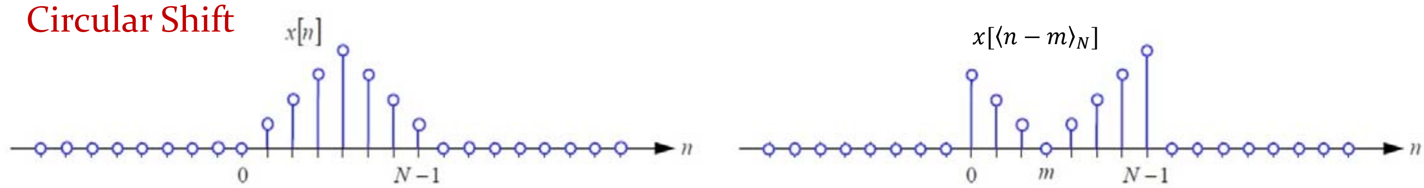
Circular Shift of A Sequence

- let $x[n]$ is defined for the range of $0 \le n \le N-1$
- Linear shifted sequence $x_1[n] = x[n-m]$ is no longer defined for the range of $0 \le n \le N-1$
- Define
  $$x_c[n] = x[\langle n-m \rangle_N]$$
  where $\langle k \rangle_N = k$ modulo N.
- For $m > 0$ (right circular shift), the above equation implies:
  $$x_c[n] = \begin{cases} x[n-m], & \text{for } m \le n \le N-1, \\ x[n-m+N], & \text{for } 0 \le n < m \end{cases}$$

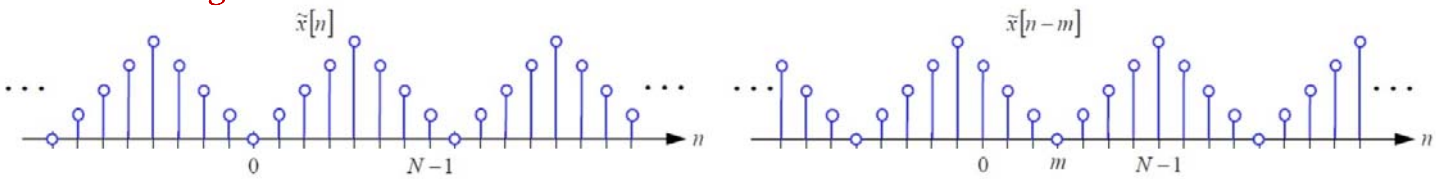
$$x_c[n] = x[\langle n-m\rangle_N] = \begin{cases} x[n-m], & \text{for } m \le n \le N-1, \\ x[n-m+N], & \text{for } 0 \le n \le m \end{cases}$$

Linear Shift

Circular Shift



Periodic Signal



Circular Convolution

- Definition:

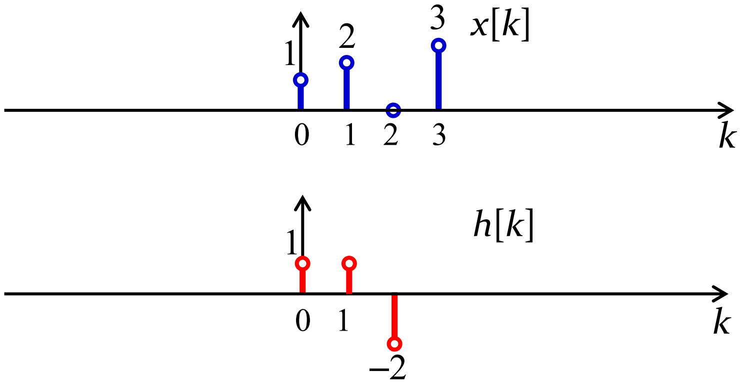
$$y_c[n] = x[n] \circledN h[n] \triangleq \sum_{m=0}^{N-1} x[m]h[\langle n-m\rangle_N]$$

for two signals of length N

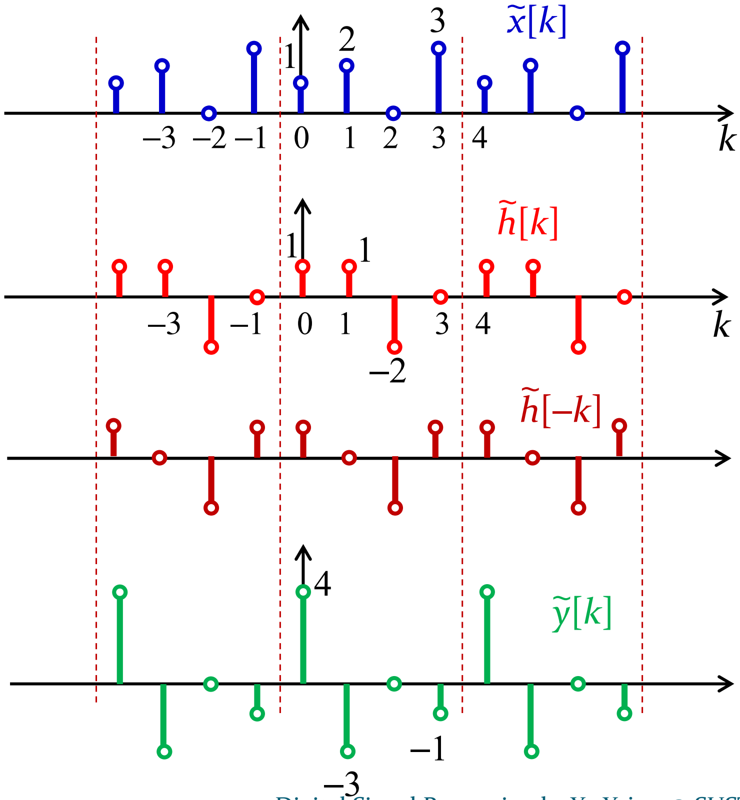
- Circular convolution is commutative:

$$x[n] \circledN h[n] = h[n] \circledN x[n]$$

Example 4



Example 4 (continued)



Properties of DFT

- Many are analogous to the properties of DTFT, but replacing shifting to circular shifting
- Inherited from Fourier Transform, so no need to be proved

General Properties of the DFT

Properties	Sequence	N-point DFT
	$g[n]$ $h[n]$	$G[k]$ $H[k]$
Linearity	$\alpha g[n] + \beta h[n]$	$\alpha G[k] + \beta H[k]$
Circular time shifting	$g[\langle n - n_0 \rangle_N]$	$W_N^{kn_0} G[k]$
Circular frequency shifting	$W_N^{-k_0 n} g[n]$	$G[\langle k - k_0 \rangle_N]$
Duality	$G[n]$	$Ng[\langle -k \rangle_N]$
Circular Convolution	$g[n] \circledN h[n]$	$G[k]H[k]$
Modulation	$g[n]h[n]$	$\frac{1}{N}\sum_{l=0}^{N-1} G[l]H[\langle k - l \rangle_N]$
Parseval's Theorem	$\sum_{n=0}^{N-1} g[n]h^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} G[k]H^*[k]$	

Symmetry Properties of the DFT

an N-point Complex Sequence

Length-N Sequence	**N-point DFT**
$x[n] = x_{\mathrm{re}}[n] + jx_{\mathrm{im}}[n]$	$X[k] = X_{\mathrm{re}}[k] + jX_{\mathrm{im}}[k]$
$x[\langle -n\rangle_N]$	$X[\langle -k\rangle_N]$
$x^*[\langle -n\rangle_N]$	$X^*[k]$
$x^*[n]$	$X^*[\langle -k\rangle_N]$
$x_{\mathrm{re}}[n]$	$X_{\mathrm{cs}}[k] = \frac{1}{2}\{X[k] + X^*[\langle -k\rangle_N]\}$
$jx_{\mathrm{im}}[n]$	$X_{\mathrm{ca}}[k] = \frac{1}{2}\{X[k] - X^*[\langle -k\rangle_N]\}$
$x_{\mathrm{cs}}[n]$	$X_{\mathrm{re}}[k]$
$x_{\mathrm{ca}}[n]$	$jX_{\mathrm{im}}[k]$

Symmetry Properties of the DFT

an N-point Real Sequence

Length-N Sequence	**N-point DFT**
$x[n] = x_{\mathrm{ev}}[n] + x_{\mathrm{od}}[n]$	$X[k] = X_{\mathrm{re}}[k] + jX_{\mathrm{im}}[k]$
$x_{\mathrm{ev}}[n]$	$X_{\mathrm{re}}[k]$
$x_{\mathrm{od}}[n]$	$jX_{\mathrm{im}}[k]$
Conjugate Symmetric	$X[k] = X^*[\langle -k\rangle_N]$
	$X_{\mathrm{re}}[k] = X_{\mathrm{re}}[\langle -k\rangle_N]$
	$X_{\mathrm{im}}[k] = -X_{\mathrm{im}}[\langle -k\rangle_N]$
	$\lvert X[k]\rvert = \lvert X[\langle -k\rangle_N]\rvert$
	$\arg\{X[k]\} = -\arg\{X[\langle -k\rangle_N]\}$

Example 5

- Q: if a complex sequence $x[n] \leftrightarrow X[k]$, then $x^*[n] \leftrightarrow X^*[\langle -k \rangle_N]$
- Proof: since $X^*[k] = \sum_{n=0}^{N-1} x^*[n] e^{j2\pi kn/N}$

  then $X^*[\langle -k \rangle_N] = \sum_{n=0}^{N-1} x^*[n] e^{j2\pi(\langle -k \rangle_N)n/N}$

  For $k = 0$, we have $X^*[\langle -k \rangle_N] = \sum_{n=0}^{N-1} x^*[n]$

  For $1 \leq k \leq N-1$, we have

  $X^*[\langle -k \rangle_N] = X^*[N-k] = \sum_{n=0}^{N-1} x^*[n] e^{j2\pi(N-k)n/N}$

  $= \sum_{n=0}^{N-1} x^*[n] e^{j2\pi n} e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x^*[n] e^{-j2\pi kn/N}$

  Combining the above two results, we get

  $X^*[\langle -k \rangle_N] = \sum_{n=0}^{N-1} x^*[n] e^{-j2\pi kn/N}$, i.e., the DFT of $x^*[n]$

Example 6

- Q: If real sequence $x[n] \leftrightarrow X[k]$, then $X[\langle -k \rangle_N] = X[\langle N-k \rangle_N] = X^*[k]$
- Proof:

$$X[\langle N-k \rangle_N] = \sum_{n=0}^{N-1} x[n] W_N^{(N-k)n} = \sum_{n=0}^{N-1} x[n] W_N^{Nn} W_N^{-kn}$$

$$= \sum_{n=0}^{N-1} x[n] W_N^{-kn} = X^*[k]$$

Example 7

- Q: Consider a length-N sequence $x[n], 0 \leq n \leq N-1$, with N even. Define two subsequences of length-$\left(\frac{N}{2}\right)$ each: $g[n] = x[2n]$ and $h[n] = x[2n+1], 0 \leq n \leq \frac{N}{2} - 1$. Denote $X[k], 0 \leq k \leq N-1$, the N-point DFT of $x[n]$, and $G[k]$ and $H[k], 0 \leq k \leq \frac{N}{2} - 1$, the $\left(\frac{N}{2}\right)$-point DFT of $g[n]$ and $h[n]$, respectively. Express $X[k]$ as a function of $G[k]$ and $H[k]$.

- A: Given the DFT of the original sequence, $X[k]$, we can express it in terms of even and odd parts.

$$
\begin{aligned}
X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} &= \sum_{r=0}^{\frac{N}{2}-1} x[2r] W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] W_N^{(2r+1)k} \\
&= \sum_{r=0}^{\frac{N}{2}-1} x[2r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] W_{N/2}^{rk} \\
&= \sum_{r=0}^{\frac{N}{2}-1} g[r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} h[r] W_{N/2}^{rk} \\
&= G\left[\langle k \rangle_{\frac{N}{2}}\right] + W_N^k H\left[\langle k \rangle_{\frac{N}{2}}\right], \qquad 0 \leq k \leq N-1
\end{aligned}
$$

Properties of the DFT

- Circular Convolution: Let $x_1[n]$ and $x_2[n]$ be length N with DFT $X_1[k]$ and $X_2[k]$

$$x_1[n] \circledN x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

  - Very useful!!! (for linear convolutions with DFT)

- Multiplication (Modulation): Let $x_1[n]$ and $x_2[n]$ be length N with DFT $X_1[k]$ and $X_2[k]$

$$x_1[n] \cdot x_2[n] \leftrightarrow \frac{1}{N} X_1[k] \circledN X_2[k]$$

Linear Convolution

- Next....
  - Using DFT, circular convolution is easy
  - But, **linear** convolution is useful, not circular
  - So, show how to perform linear convolution with circular convolution
  - Use DFT to do linear convolution

Linear Convolution

- We start with two non-periodic sequences:

$$x[n] \quad 0 \le n \le L-1$$
$$h[n] \quad 0 \le n \le P-1$$

  for example, $x[n]$ is a signal, and $h[n]$ an impulse response of a system.

- • We want to compute the linear convolution:

$$h[n] \circledast x[n] = \sum_{k=0}^{L-1} x[k]h[n-k]$$

  $y[n]$ is nonzero for $0 \le n \le L+P-2$ with length $M=L+P-1$

- • Requires $L \cdot P$ multiplications if computed directly

Linear Convolution via Circular Convolution

- Zero-pad $x[n]$ by $P-1$ zeros

$$x_{zp}[n] = \begin{cases} x[n] & 0 \le n \le L-1 \\ 0 & L \le n \le L+P-2 \end{cases}$$

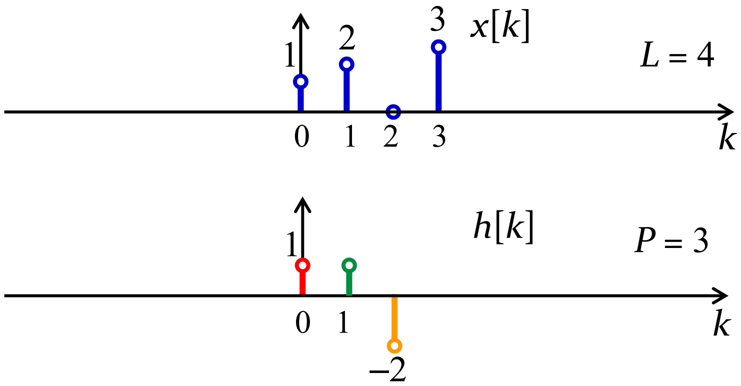
- Zero-pad $h[n]$ by $L-1$ zeros

$$h_{zp}[n] = \begin{cases} h[n] & 0 \le n \le P-1 \\ 0 & P \le n \le L+P-2 \end{cases}$$

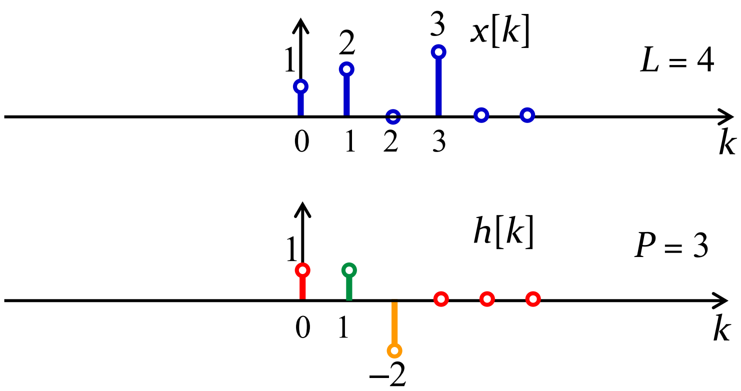
- Now, both sequences are of length $M=L+P-1$
- We can now compute the linear convolution using a circular one with length $M = L+P-1$

$$y[n] = h[n] * x[n] = x_{zp}[n] \, ⓝ \, h_{zp}[n]$$

Example 8

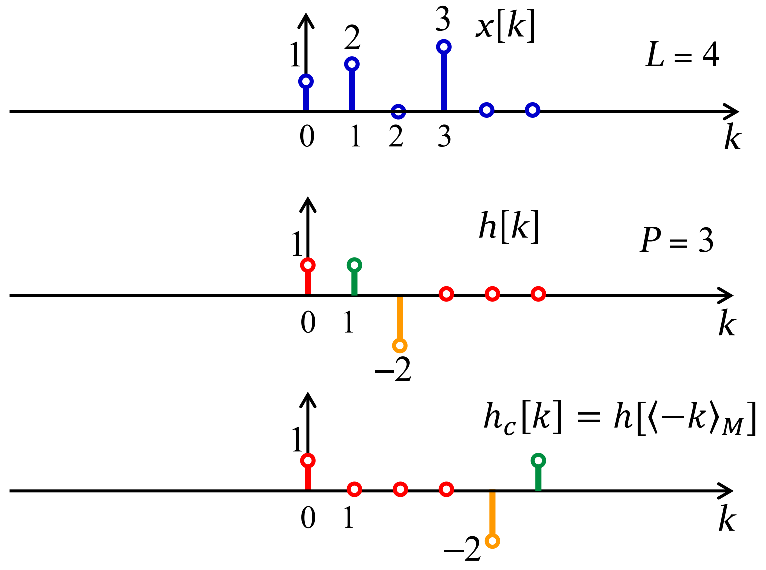


Example 8 (continued)



$M=L+P-1=6$

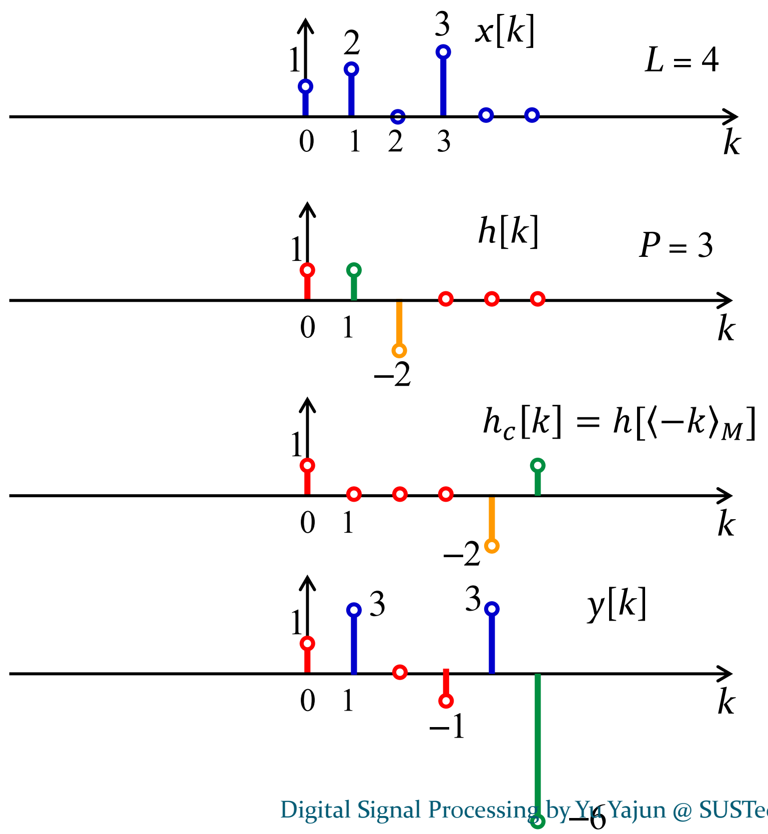
Example 8 (continued)



M=L+P−1=6

Circular Flip
(Circular time-reversal)

Example 8 (continued)



M=L+P−1=6

Circular Flip
(Circular time-reversal)

Linear Convolution using DFT

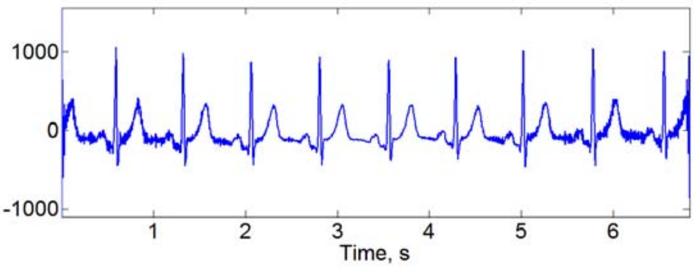
- In practice we can implement a circular convolution using the DFT property:

$$h[n] \circledast x[n] = x_{zp}[n] \, \circledN \, h_{zp}[n]$$
$$= \mathcal{IDFT}\{\mathcal{DFT}\{x_{zp}[n]\} \cdot \mathcal{DFT}\{h_{zp}[n]\}\}$$

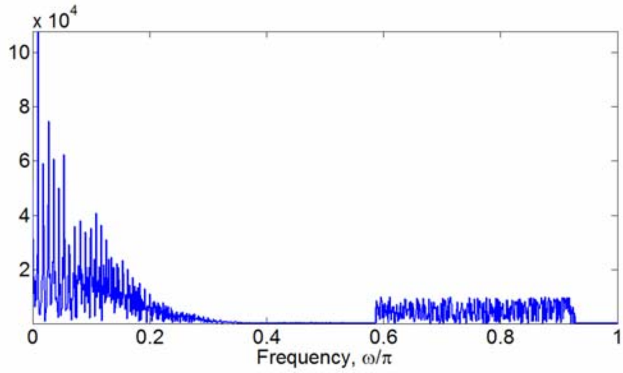
- Advantage: DFT can be computed with $N \log_2 N$ complexity (FFT algorithm!)
- Drawback: Must wait for all the samples -- huge delay -- incompatible with real-time implementation

Fourier-Domain Filtering

- Remove some frequency bands directly from Fourier-Domain, since $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$



ECG Signal + Noise



ECG Signal + Noise

- In this particular example, set

$$H(e^{j\omega}) = \begin{cases} 0, & 0.55\pi \le \omega \le 0.95\pi \\ 1, & \text{otherwise} \end{cases}$$

Fourier-Domain Filtering

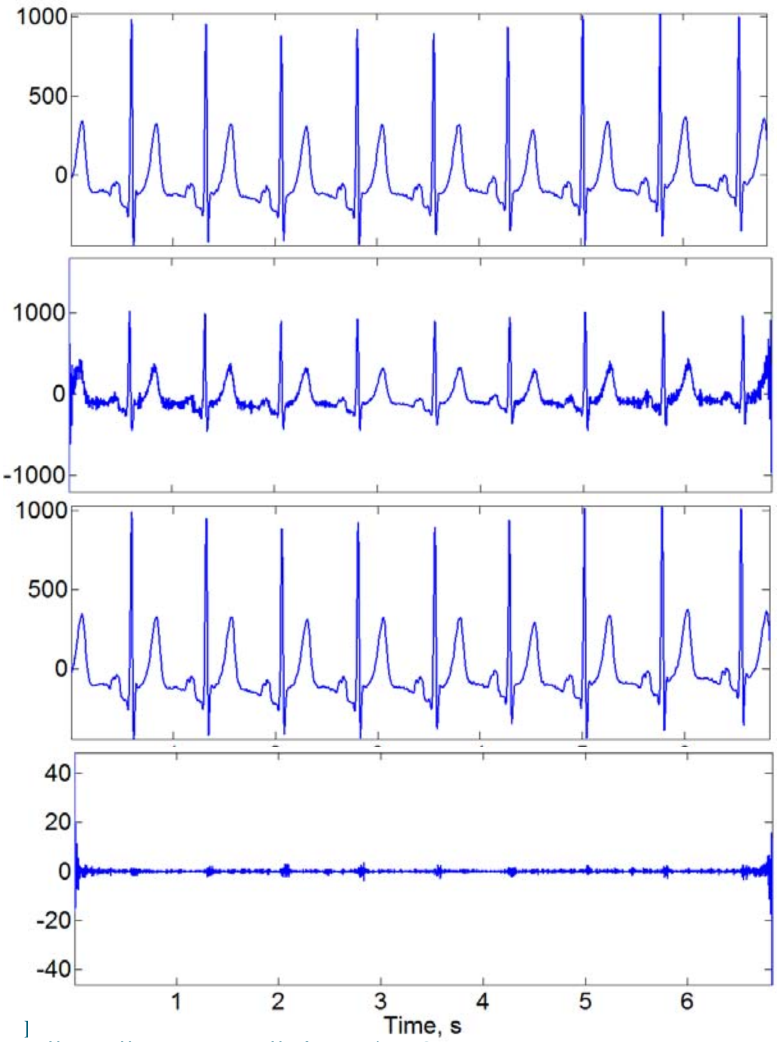
- Find the DTFT of the ECG signal to get $X(e^{j\omega})$, multiply with $H(e^{j\omega})$ to obtain $Y(e^{j\omega})$, and find the IDTFT of $Y(e^{j\omega})$
- We can use DFT to compute $X(e^{j\omega})$ and $Y(e^{j\omega})$ at frequency values of $\omega = 2\pi k/N$, for , $k = 0, 1, \dots, N-1$
- This approach is equivalent to the circular convolution of the finite-length signal $x[n]$ and the finite-length ideal filter $h[n]$.
- However, the ideal filter has an infinite length impulse response. Sampling the Fourier transform to create DFT samples leads to the time domain aliasing.

Original Signal without noise $s[n]$

$x[n]$: Original Signal with noise

$y[n]$: Filtered $x[n]$

$s[n] - y[n]$: time aliasing of the recovered signal



Computation of DFT

- The N-point DFT $X[k]$ of a length-N sequence $x[n]$, $0 \le n \le N-1$, is defined by

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0, 1, \dots, N-1$$

where $W_N = e^{-j2\pi/N}$.

- Direct computation of all N samples of $\{X[k]\}$ requires N^2 complex multiplications and $N(N-1)$ complex additions.

Decimation-in-time FFT algorithm

- Most conveniently illustrated by considering the special case of N an integer power of 2, i.e, $N = 2^v$.
- Since N is an even integer, we can consider computing $X[k]$ by separating $x[n]$ into two $(N/2)$-point sequences consisting of the even numbered point in $x[n]$ and the odd-numbered points in $x[n]$, with the substitution of variable $n = 2r$ for n even and $n = 2r+1$ for n odd

- We have

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

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

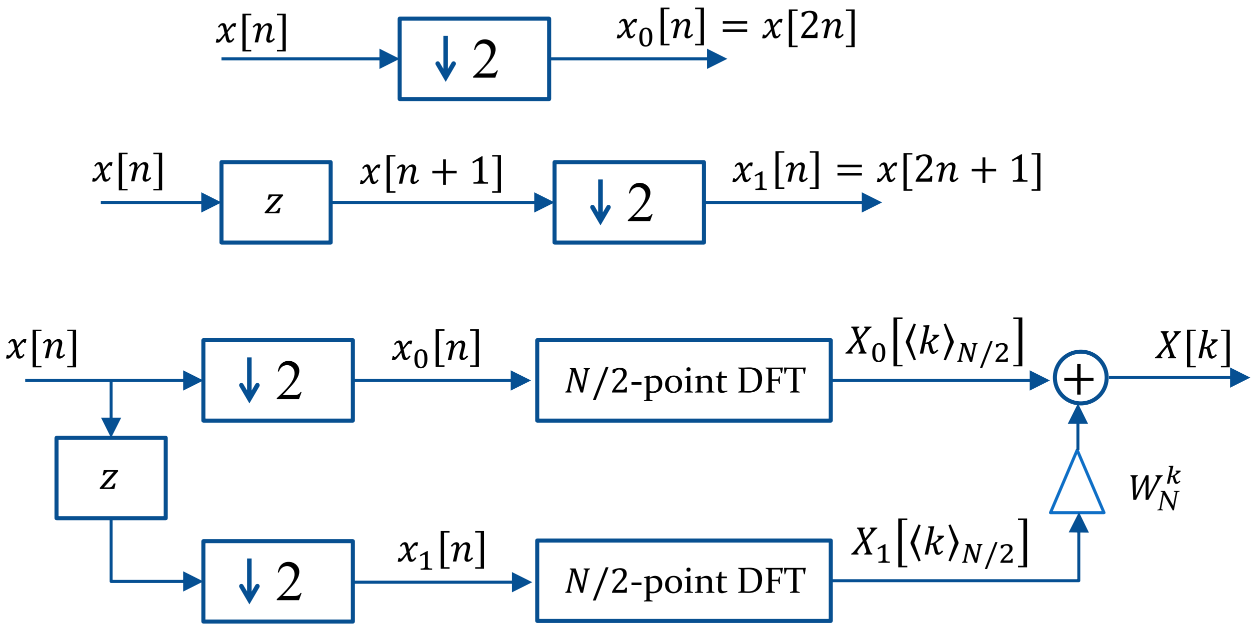
$$= X_0[\langle k \rangle_{N/2}] + W_N^k X_1[\langle k \rangle_{N/2}], \qquad 0 \le k \le N-1$$

where $X_0[k] = \sum_{r=0}^{\frac{N}{2}-1} x_0\,[r]\, W_{N/2}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} x[2r]\, W_{N/2}^{rk}$ and

$X_1[k] = \sum_{r=0}^{\frac{N}{2}-1} x_1[r]\, W_{N/2}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]\, W_{N/2}^{rk}$, $0 \le k \le \frac{N}{2} - 1$

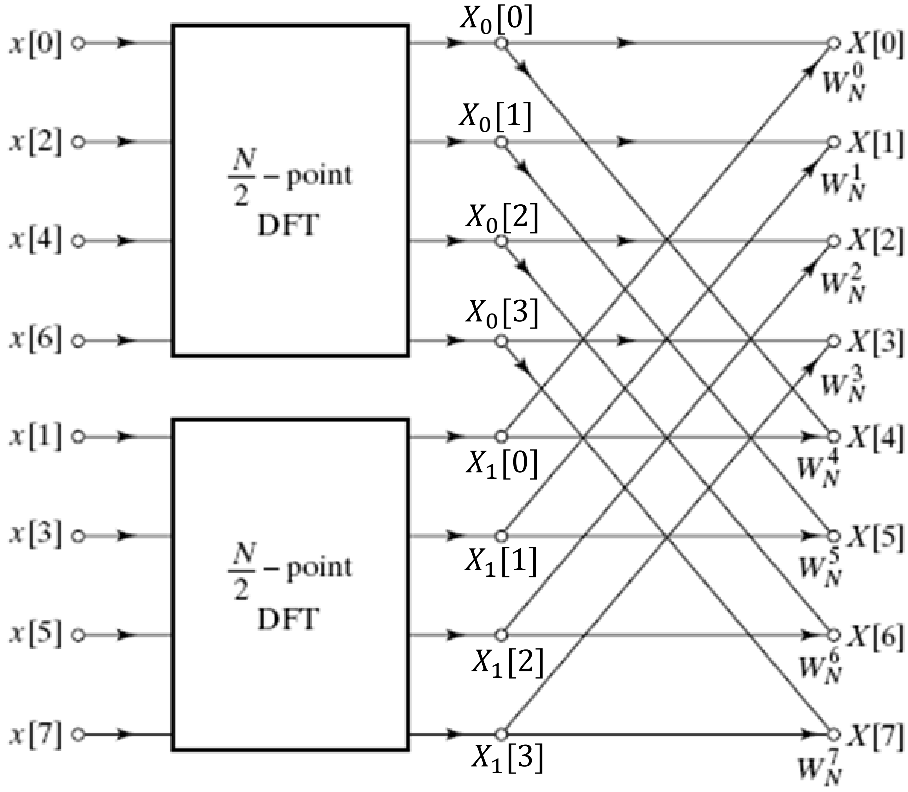
- Both $X_0[k]$ and $X_1[k]$ can be computed by $(N/2)$-point DFT

Structure Interpretation



- Decomposing N-point DFT into two $(N/2)$-point DFT for the case of N=8

$X[k] = X_0[\langle k \rangle_{N/2}] + W_N^k X_1[\langle k \rangle_{N/2}],$



Cost to compute an *N*-point DFT (1)

- A direct computation
  - ➢ N^2 complex multiplications
  - ➢ $N^2 - N \cong N^2$ complex additions
- Using a decomposition computing two $(N/2)$-point DFTs
  - $N + 2(N/2)^2 = N + N^2/2$ complex multiplications
  - Approximately $N + N^2/2$ complex additions
- For $N \geq 3$, $N + N^2/2 < N^2$

- We can further decompose the $(N/2)$-point DFT into two $(N/4)$-point DFTs.

$$X_0[k] = X_{00}[\langle k \rangle_{N/4}] + W_{N/2}^{k} X_{01}[\langle k \rangle_{N/4}], 0 \le k \le N/2 - 1$$

$$X_1[k] = X_{10}[\langle k \rangle_{N/4}] + W_{N/2}^{k} X_{11}[\langle k \rangle_{N/4}], 0 \le k \le N/2 - 1$$

where,

$$X_{00}[k] = \sum_{r=0}^{\frac{N}{4}-1} x_{00}[r] W_{N/4}^{rk} = \sum_{r=0}^{\frac{N}{4}-1} x_0[2r] W_{N/4}^{rk}$$

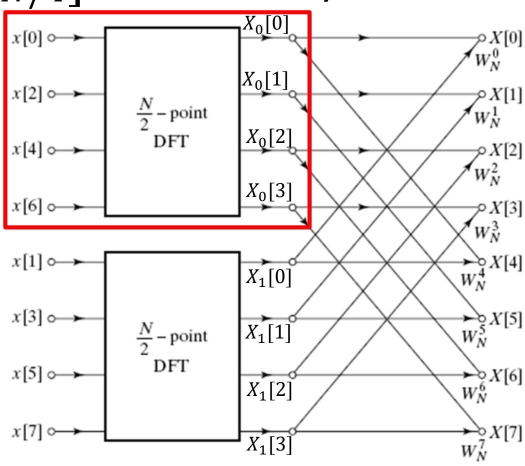
$$= \sum_{r=0}^{\frac{N}{4}-1} x[4r] W_{N/4}^{rk}$$

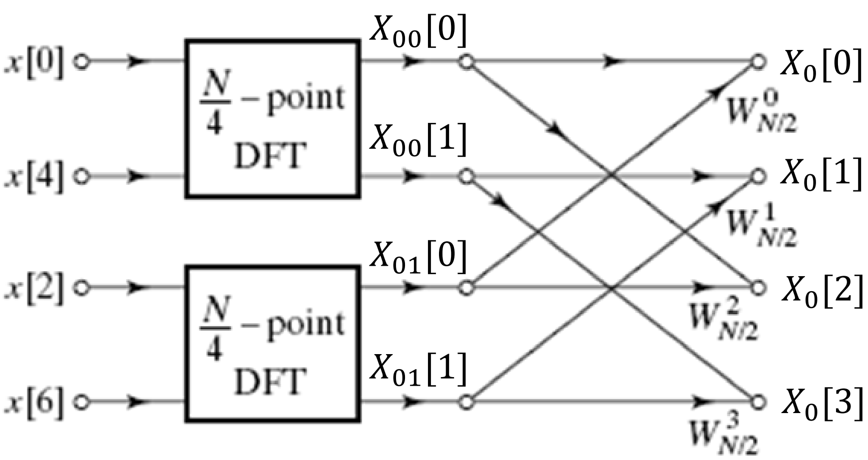
etc.

- For example, the upper half of the previous diagram, corresponding to

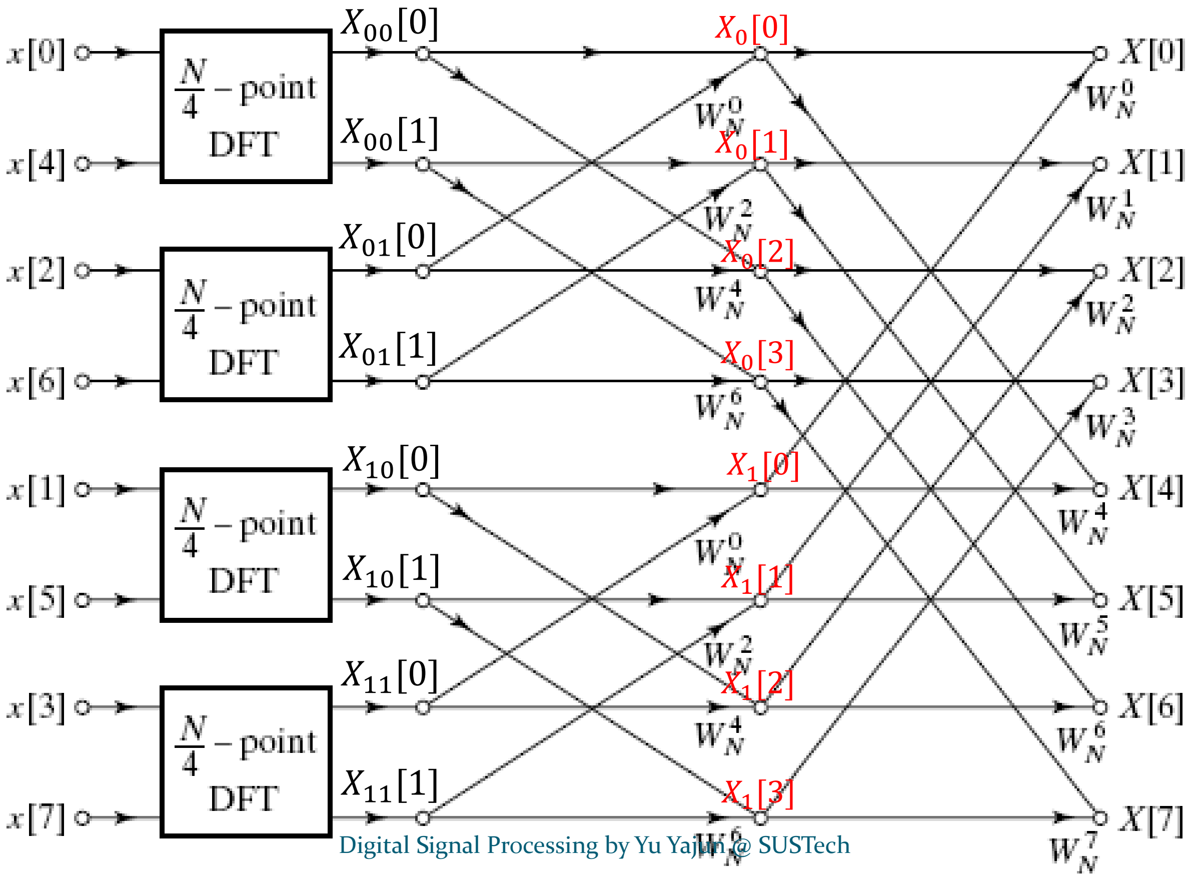
$$X_0[k] = X_{00}[\langle k \rangle_{N/4}] + W_{N/2}^{k} X_{01}[\langle k \rangle_{N/4}], 0 \le k \le N/2 - 1$$

can be de-composed as

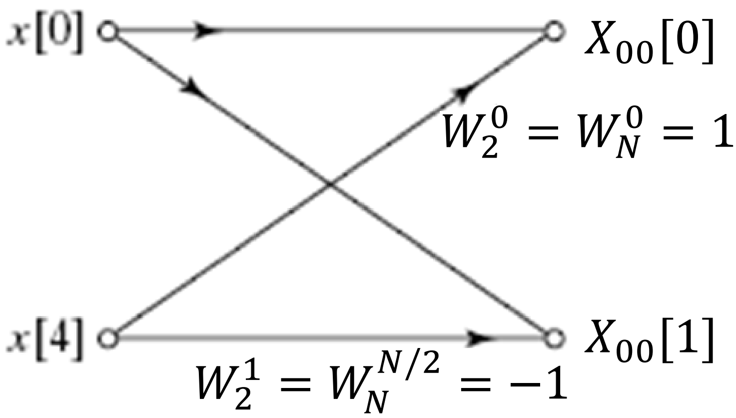




- Hence, the 8-point DFT can be obtained by the following diagram with four 2-point DFTs.

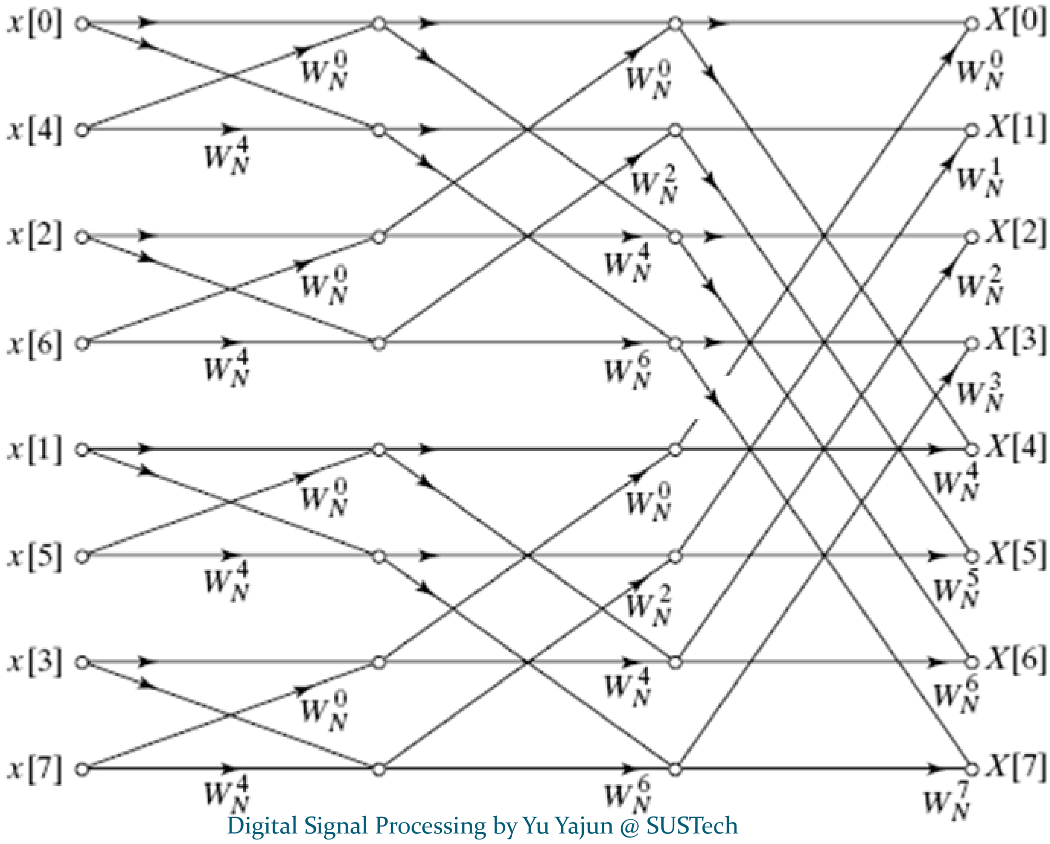


- Finally, each 2-point DFT can be implemented by the following signal-flow graph, where no multiplications are needed.



$$X_{00}[0] = x[0]W_2^0 + x[4]W_2^0$$
$$X_{00}[1] = x[0]W_2^0 + x[4]W_2^1$$

- Complete flow graph of decimation-in-time decomposition of an 8-point DFT.



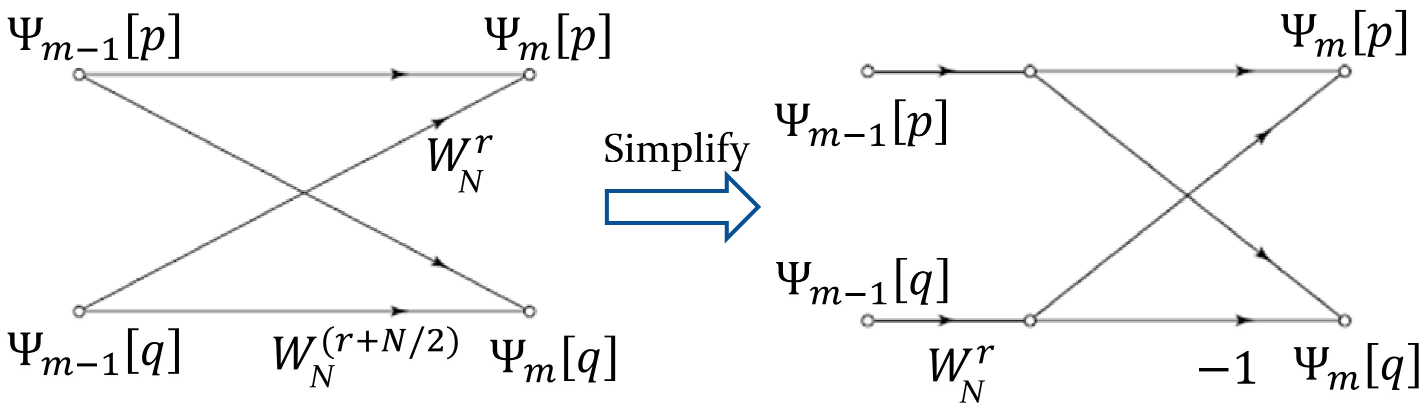
Cost to compute an *N*-point DFT (2)

- Eight-point FFT consists of three stages.
  - The 1st stage computes the four 2-point DFTs; the 2nd stage computes the two 4-point DFTs; the 3rd stage computes the desired 8-point DFT.
  - The number of complex multiplications and additions performed at each stage is 8, the size of transformation. In total 24 complex multiplications and additions are required.
- When N is the power of 2, $N = 2^v$, it requires $v = \log_2 N$ stages of computation. The number of complex multiplications and additions required is therefore $Nv = N\log_2 N$.

- In each stage of the decimation-in-time FFT algorithm, there are a basic structure called the butterfly computation:

$$\Psi_m[p] = \Psi_{m-1}[p] + W_N^r \Psi_{m-1}[q]$$

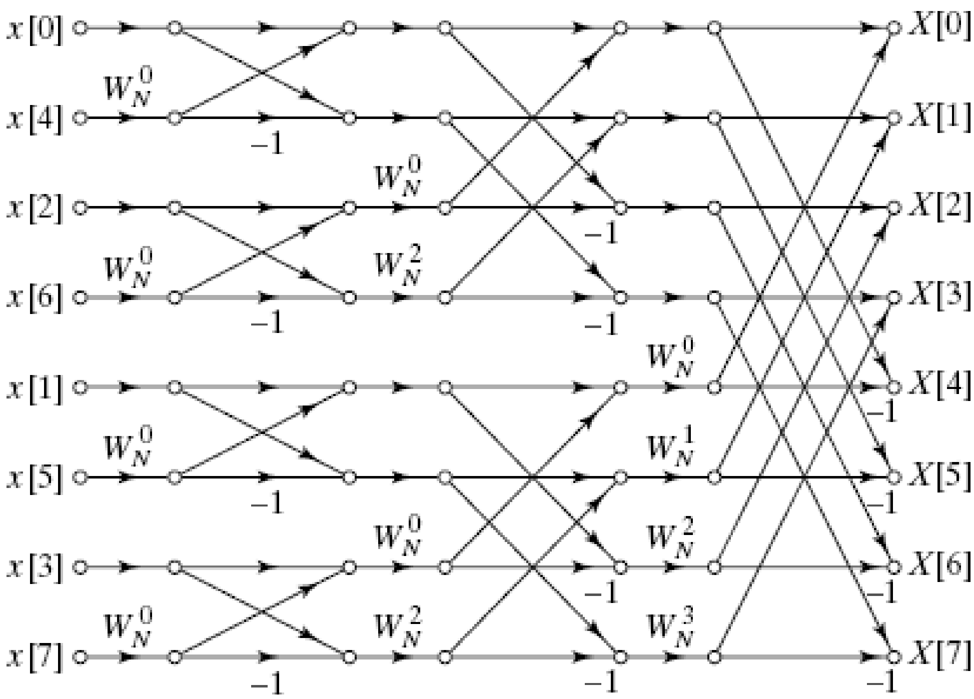
$$\Psi_m[q] = \Psi_{m-1}[p] + W_N^{r+(N/2)} \Psi_{m-1}[q]$$



Flow graph of a basic butter-fly Computation in FFT.

Simplified butterfly computation.

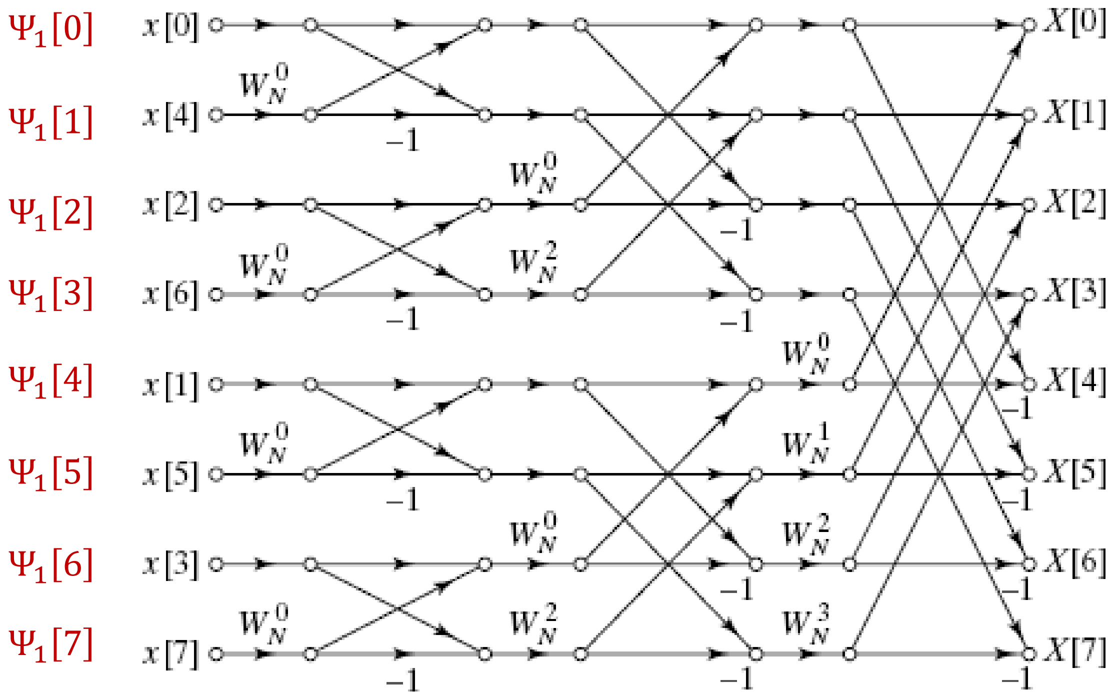
- Flow graph of 8-point FFT using the simplified butterfly computation



Cost to compute an *N*-point DFT (3)

- Using the simplified butterfly computation, the number of complex multiplications ~~and additions~~ performed at each stage is reduced to $N/2$. Thus the total numbers become $Nv/2 = \frac{N}{2}\log_2 N$
- By excluding trivial complex multiplications with $W_N^0 = 1$ and $W_N^{N/2} = -1$, the exact count of non-trivial complex multiplications are even less, given by $\frac{N}{2}(\log_2 N - 2) + 1$

- Note the ordering of the input sequence $x[n]$, while the DFT samples $X[k]$ appear at the output in a sequential order.



- Let $\Psi_1[m]$ be the sequentially ordered new representation of the input sample $x[n]$.

$\Psi_1[m]$	$x[n]$
$\Psi_1[0]$	$x[0]$
$\Psi_1[1]$	$x[4]$
$\Psi_1[2]$	$x[2]$
$\Psi_1[3]$	$x[6]$
$\Psi_1[4]$	$x[1]$
$\Psi_1[5]$	$x[5]$
$\Psi_1[6]$	$x[3]$
$\Psi_1[7]$	$x[7]$

m is a bit-reversed version of n

When *N* is not the power of 2

- By zero-padding a sequence into an N-point sequence with $N = 2^v$, we can choose the nearest power-of-two FFT algorithm for implementing a DFT.
  - ➢ The FFT algorithm of power-of-two is also called the Cooley-Tukey algorithm since it was first proposed by them.

Decimation-in-frequency FFT algorithm

The decimation-in-time FFT algorithms are all based on structuring the DFT computation by forming smaller and smaller subsequences of the input sequence $x[n]$. Alternatively, we can consider dividing the output sequence $X[k]$ into smaller and smaller subsequences in the same manner.

$$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{nk} \qquad k=0,1,\ldots,N-1$$

$$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{nk} \qquad k=0,1,\ldots,N-1$$

The even-numbered frequency samples are

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

$$X[2r]=\sum_{n=0}^{(N/2)-1} x[n]W_N^{2nr}+\sum_{n=0}^{(N/2)-1} x[n+(N/2)]W_N^{2r(n+(N/2))}$$

Since $W_N^{2r[n+(N/2)]}=W_N^{2rn}W_N^{rN}=W_N^{2rn}$

$$X[2r]=\sum_{n=0}^{(N/2)-1}(x[n]+x[n+(N/2)])W_{N/2}^{rn} \qquad r=0,1,\ldots,(N/2)-1$$

$$X[2r] = \sum_{n=0}^{(N/2)-1} (x[n] + x[n+(N/2)])W_{N/2}^{rn} \qquad r = 0,1,\ldots,(N/2)-1$$

The above equation is the $(N/2)$-point DFT of the $(N/2)$-point sequence obtained by adding the first and the last half of the input sequence.

Adding the two halves of the input sequence represents time aliasing, consistent with the fact that in computing only the even-number frequency samples, we are sub-sampling the Fourier transform of $x[n]$.

We now consider obtaining the odd-numbered frequency points:

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

Since $\displaystyle\sum_{n=N/2}^{N-1} x[n]W_N^{n(2r+1)} = \sum_{n=0}^{(N/2)-1} x[n+(N/2)]W_N^{(n+N/2)(2r+1)}$

$$= W_N^{(N/2)(2r+1)} \sum_{n=0}^{(N/2)-1} x[n+(N/2)]W_N^{n(2r+1)}$$

$$= -\sum_{n=0}^{(N/2)-1} x[n+(N/2)]W_N^{n(2r+1)}$$

We obtain

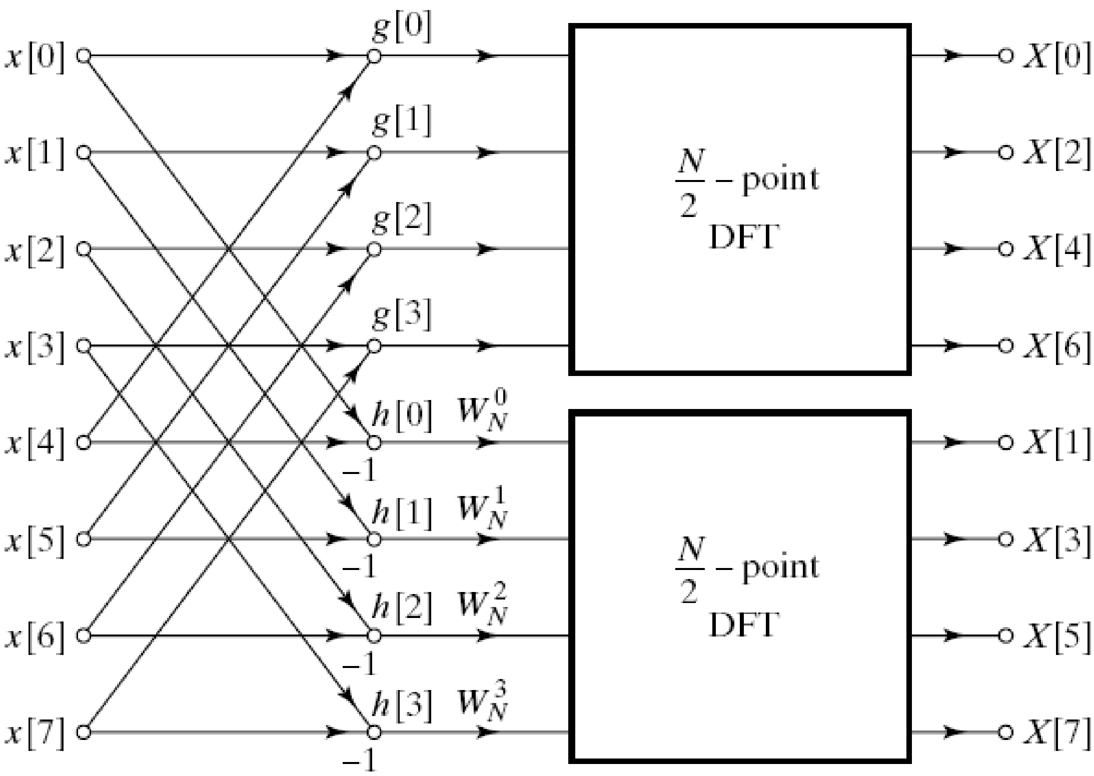
$$X[2r+1] = \sum_{n=0}^{(N/2)-1} (x[n] - x[n+N/2]) W_N^{n(2r+1)}$$

$$= \sum_{n=0}^{(N/2)-1} \boxed{(x[n] - x[n+N/2]) W_N^{n}} W_{N/2}^{nr} \qquad r = 0, 1, \ldots, (N/2)-1$$

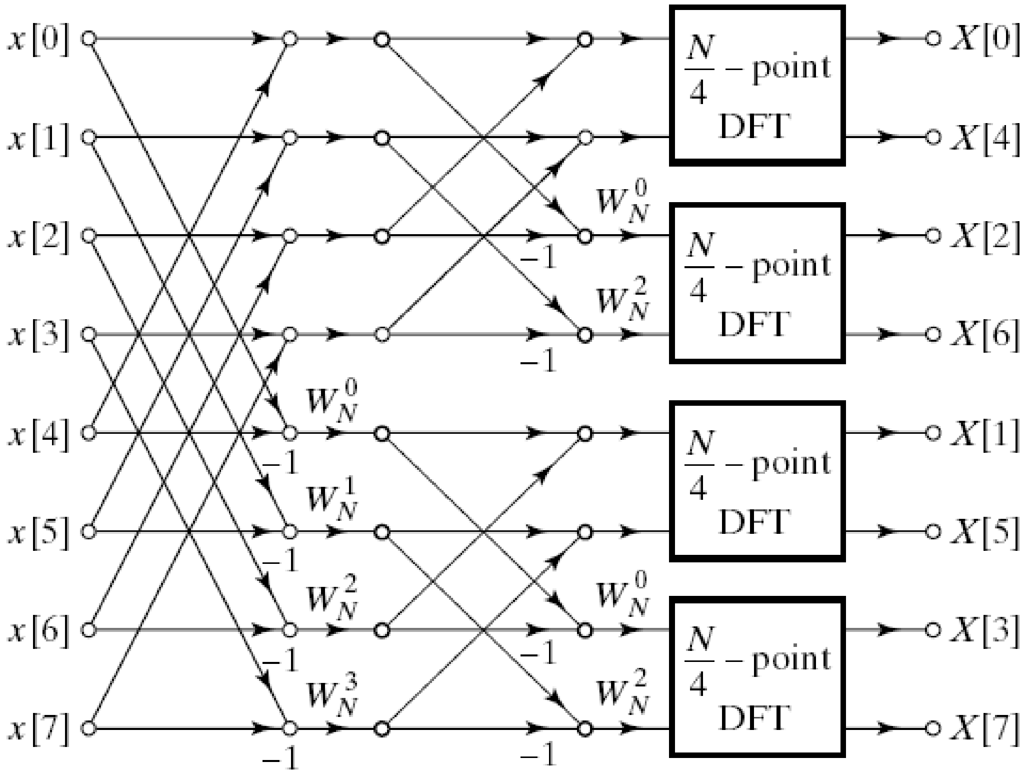
The above equation is the $(N/2)$-point DFT of the sequence obtained by subtracting the second half of the input sequence from the first half and multiplying the resulting sequence by W_N^n.

Let $g[n] = x[n] + x[n+N/2]$ and $h[n] = x[n] - x[n+N/2]$, the DFT can be computed by forming the sequences $g[n]$ and $h[n]$, then computing $h[n]W_N^n$, and finally computing the $(N/2)$-point DFTs of these two sequences.

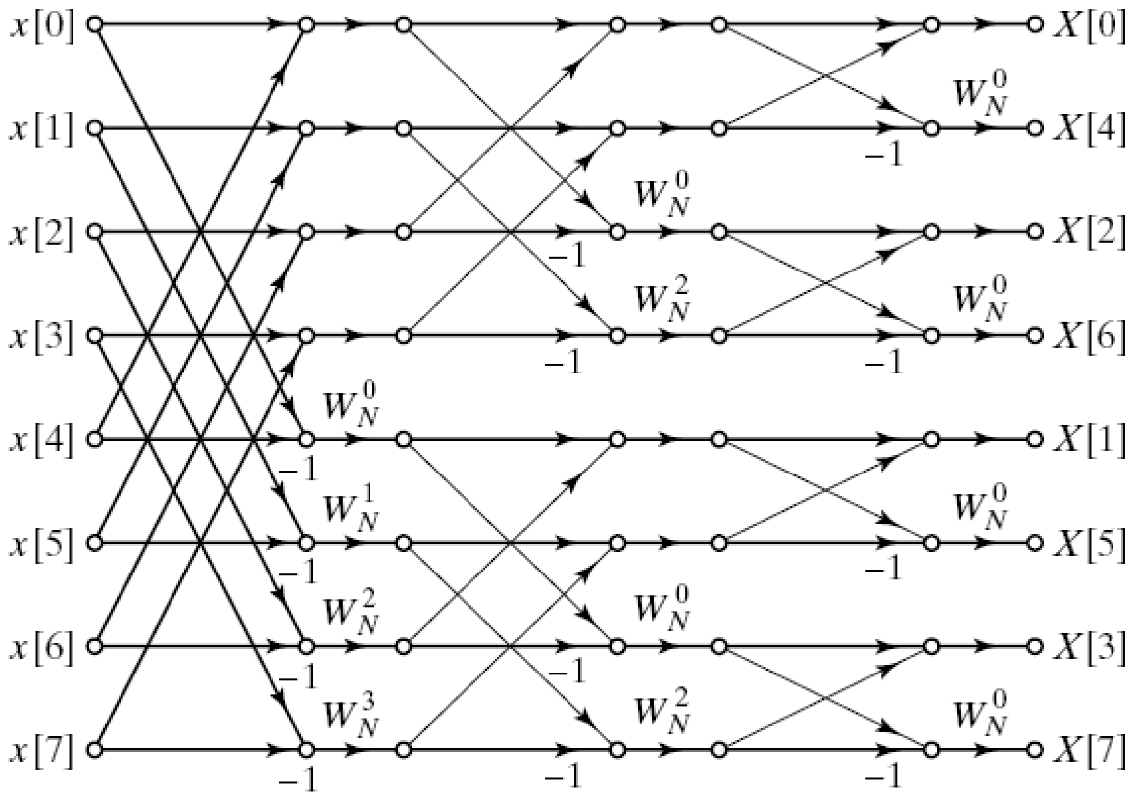
Flow graph of decimation-in-frequency decomposition of an N-point DFT ($N = 8$).



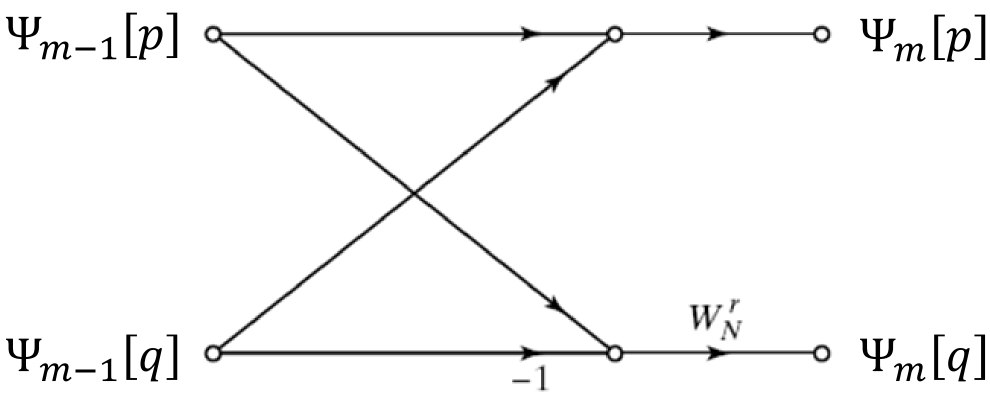
Recursively, we can further decompose the $(N/2)$-point DFT into smaller substructures:



Finally, we have



Butterfly structure for decimation-in-frequency FFT algorithm:



The decimation-in-frequency FFT algorithm also has the computation complexity of $O(N\log_2 N)$