UCL DEPARTMENT OF ELECTRONIC AND ELECRICAL ENGINEERING

INTEGRATED GRADUATE DEVELOPMENT PROGRAMME

MSC TELECOMMUNICATIONS

Discrete Fourier Transform
Fast Fourier Transform

Prof Yiannis Andreopoulos

University College London

Discrete Fourier Transform: Motivation

- We defined the continuous Fourier Transform as:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)\exp(-j\omega t)\,dt \quad f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)\exp(j\omega t)\,dt$$

- However, this integral equation of the Fourier Transform is not suitable to perform frequency analysis in digital communication systems since:

➢ Continuous nature cannot be handled by computers

➢ The limits of integration cannot be from $-\infty$ to $+\infty$. Only finite length sequences can be handled by a computer.

Discrete Fourier Transform: Definition

- Let $x(n)$ be a finite length signal.
  The N-point DFT of $x(n)$ defined as $X(k)$ is :

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}, \quad k = 0,1, \ldots, N\text{-}1$$

  - k represents the harmonic of the transform component
  - n is the finite length sequence interval defined as
    $0 \le n \le N\text{-}1,\ N$ is the sequence length
    → *sampling and computing the DFT*
- $X(k)$ is complex, so that the k th harmonic of $X(k)$ is:

$$X(k) = R(k) + jI(k)$$

Discrete Fourier Transform: Properties

- The four properties of the DFT:

1. Periodicity

If $X(k)$ is the N-point DFT of $x(n)$,

$$x(n+N) = x(n), \quad \text{for all } n$$

$$X(k+N) = X(k), \quad \text{for all } k$$

It shows that DFT is periodic with period N, also known as the cyclic property of the DFT

2. Linearity

If $X_1(k)$ and $X_2(k)$ are the N-point DFT of $x_1(n)$ and $x_2(n)$,

$$ax_1(n) + bx_2(n) \xleftrightarrow{\text{DFT}} aX_1(k) + bX_2(k)$$

Discrete Fourier Transform: Properties

3. **Circular Shifting**

Let $x(n)$ be a sequence of length N and $X(k)$ is its N-point DFT. Let the sequence $x_m(n)$ be obtained from $x(n)$ by shifting cyclically by m units. Then,

$$x_m(n) \xleftrightarrow{\text{DFT}} X(k)e^{-j2\pi km/N}$$

4. **Parseval's Theorem** (= DFT is a unitary transform)

if $x(\mathrm{n}) \xleftrightarrow{\text{DFT}} X(k)$ and

$y(n) \xleftrightarrow{\text{DFT}} Y(k)$

thus, $\sum_{n=0}^{N-1} x(n)y^*(n) = 1/N \sum_{k=0}^{N-1} X(k)\,Y^*(k)$

The Discrete Fourier Transform and the Z Transform

- The Z-transform of the sequence, $x(n)$ is given by:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$, ROC includes the unit circle

by defining $z_k = e^{j2\pi k/N}$, $k = 0, 1, 2, \ldots, N\text{-}1$

$$X(k) = X(z)\big|_{z_k = e^{j2\pi k/N}}, \ k = 0,1,2, \ldots, N\text{-}1$$

$$= \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N}$$

where $\omega_k = 2\pi k/N$, $k = 0,1,2,\ldots,N\text{-}1$

Discrete Fourier Transform: Properties

- To perform convolution using the DFT, we need to:

1. Find N-point DFT of the sequences $h(n)$ and $x(n)$.
2. Multiply DFTs to form $Y(k) = H(k)X(k)$
3. Perform inverse DFT to obtain $y(n)$.

Discrete Fourier Transform: Examples

EXAMPLE 1:

Find the DFT for: $x(n) = \{ ¼ , ¼ , ¼ \}$

Solution :

1. Determine the sequence length, N: $N = 3$, $k = 0,1,2$
2. Use DFT formula to determine $X(k)$

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}, \quad k = 0, 1, 2$$

$$X(0) = ¼ + ¼ + ¼ = ¾$$

$$X(1) = ¼ + ¼e^{-j2\pi/3} + ¼e^{-j4\pi/3}$$

$$= ¼ + ¼ [\cos(2\pi/3) - j\sin(2\pi/3)] + ¼ [\cos(4\pi/3) - j\sin(4\pi/3)]$$

$$= ¼ + ¼ [-0.5 - j0.866] + ¼ [-0.5 + j0.866]$$

$$= ¼ + ¼ [-1] = 0$$

Discrete Fourier Transform: Examples

- Continued from EXAMPLE 1:

$$X(2) = \tfrac{1}{4} + \tfrac{1}{4}\,e^{-j4\pi/3} + \tfrac{1}{4}\,e^{-j8\pi/3}$$
$$= \tfrac{1}{4} + \tfrac{1}{4}\,[\cos(4\pi/3) - j\sin(4\pi/3)] + \tfrac{1}{4}\,[\cos(8\pi/3) - j\sin(8\pi/3)]$$
$$= \tfrac{1}{4} + \tfrac{1}{4}\,[-0.5 + j0.866] + \tfrac{1}{4}[-0.5 - j0.866]$$
$$= 0$$

Thus, $X(k) = \{\tfrac{3}{4}, 0, 0\}$

Discrete Fourier Transform: Examples

EXAMPLE 2:

Perform DFT for $x(n) = \{1,1,2,2,3,3\}$

Solution :

1. Determine the sequence length: $N = 6$.
2. Use DFT formula to determine $X(k)$.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}, \quad k = 0,1,2,3,4,5$$

$X(0) = 12,\ X(1) = -1.5 + j2.598$

$X(2) = -1.5 + j0.866,\ X(3) = 0$

$X(4) = -1.5 - j0.866,\ X(5) = -1.5 - j2.598$

Thus,

$X(k) = \{12,\ -1.5 + j2.598,\ -1.5 + j0.866,\ 0,\ -1.5 - j0.866,\ -1.5 - j2.598\}$

Discrete Fourier Transform: Examples

EXAMPLE 3:

Find the DFT for the convolution of two signals:

$x_1(\mathrm{n}) = \{2, 1, 2, 1\}$ & $x_2(\mathrm{n}) = \{1, 2, 3, 4\}$

Solution :

1. Determine the sequence length for each sequence, $N = 4$. Thus, $k = 0,1,2,3$
2. Perform DFT for each sequence,

   (i) $X_1(0) = 6, X_1(1) = 0, X_1(2) = 2, X_2(3) = 0$

   $X_1(k) = \{6, 0, 2, 0\}$

   (ii) $X_2(0) = 10, X_2(1) = -2+j2, X_2(2) = -2, X_2(3) = -2-j2$

   $X_2(k) = \{10, -2+j2, -2, -2-j2\}$
3. Perform Convolution by : $X_3(k) = X_1(k)\, X_2(k) = \{60, 0, -4, 0\}$

Inverse Discrete Fourier Transform: Definition

- The finite length sequence can be obtained from the Discrete Fourier Transform by performing IDFT.

- The IDFT is defined as :

$$x(n) = 1/N \sum_{k=0}^{N-1} X(k)\, e^{j2\pi nk/N}, \text{ where } n = 0,1, \ldots, N\text{-}1$$

Inverse Discrete Fourier Transform: Example

EXAMPLE 4:

Obtain the finite length sequence, $x(\mathrm{n})$ from the DFT sequence in Example 3.

Solution :

1. The sequence in Example 3 is : $X_3(k) = \{60, 0, -4, 0\}$
2. Use IDFT formula to obtain $x(n)$:

$$x_3(n) = 1/4 \sum_{k=0}^{3} X(k)\, e^{j2\pi nk/4},$$

$$x_3(0) = 14, \quad x_3(1) = 16, \quad x_3(2) = 14, \quad x_3(3) = 16$$

Thus, the finite length sequence is:

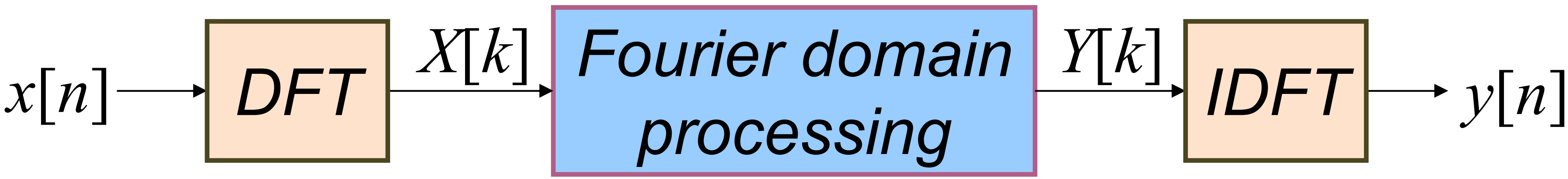
$$x_3(k) = \{14, 16, 14, 16\}$$

The Fast Fourier Transform

1. Calculation of the DFT
2. The Fast Fourier Transform algorithm

1. Calculation of the DFT

- Filter design so far has been oriented to time-domain processing - cheaper!
- But: frequency-domain processing makes some problems very simple:



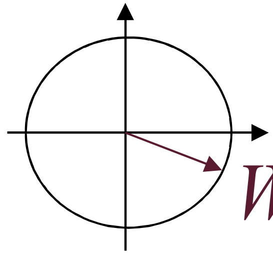
- Need an efficient way to calculate DFT!

The DFT

- Recall the DFT:

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

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



W_N @ $2\pi/N$

$\Rightarrow W_N^{\,r}$ has only N distinct values

  - discrete transform of discrete sequence

- Matrix form:

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ . & . & . & & . \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \dots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ . \\ x[N-1] \end{bmatrix}$$

Lots of structure $\rightarrow$ opportunities for efficient algorithms

Computational Complexity

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

- N cpx multiplies + N-1 cpx adds per pt × N points ($k = 0..N$-1)
  - cpx mult: (a+jb)(c+jd) = ac - bd + j(ad+bc) = 4 real mults + 2 real adds
  - cpx add = 2 real adds
- Total: $4N^2$ real mults, $4N^2$-$2N$ real adds

2. Fast Fourier Transform FFT

- Reduce complexity of DFT from $O(N^2)$ to $O(N \cdot \log N)$
  - grows significantly slower with larger N
- Works by decomposing large DFT into several stages of smaller DFTs
- Often provided as a highly optimized library

Decimation in Time (DIT) FFT

- Can rearrange DFT formula in 2 halves:

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

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

$= W_{\frac{N}{k}}$

Arrange terms in pairs

$$= \sum_{m=0}^{\frac{N}{2}-1} \left(x[2m] \cdot W_N^{2mk} + x[2m+1] \cdot W_N^{(2m+1)k} \right)$$

Group terms from each pair

$$= \underbrace{\sum_{m=0}^{\frac{N}{2}-1} x[2m] \cdot W_{\frac{N}{2}}^{mk}}_{X_0[\langle k\rangle_{N/2}]} + W_N^k \underbrace{\sum_{m=0}^{\frac{N}{2}-1} x[2m+1] \cdot W_{\frac{N}{2}}^{mk}}_{X_1[\langle k\rangle_{N/2}]}$$

*$N/2$ pt DFT of x for **even** n*　　*$N/2$ pt DFT of x for **odd** n*

Decimation in Time (DIT) FFT

- Finite sequence $x[n]$, $0 \le n < N$, $N = 2^M$
  - i.e. length is a power of 2
- Divide Z-transform into parts coming from even and odd values of n:

$$X(z) = \sum_{n=0}^{N-1} x[n] z^{-n} = X_0(z^2) + z^{-1} X_1(z^2)$$

$$X_0(z) = \sum_{n=0}^{\frac{N}{2}-1} x[2n] z^{-n}$$

ZT of $N/2$ point sequence formed from even pts of $x[n]$

$$X_1(z) = \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] z^{-n}$$

ZT of $N/2$ point sequence formed from **odd** pts of $x[n]$

Decimation in Time (DIT) FFT

$$\mathrm{DFT}_N\{x[n]\} \triangleq X[k] = X(z)\Big|_{z=e^{j2\pi k/N}=W_N^{-1}}$$

$$= X_0\left(\left(e^{j2\pi k/N}\right)^2\right) + e^{-j2\pi k/N} X_1\left(\left(e^{j2\pi k/N}\right)^2\right)$$

- Now,

$$\left(e^{j2\pi k/N}\right)^2 = e^{j2\pi k/(N/2)} = W_{\frac{N}{2}}^{-1}$$

but

$$X[k] = \mathrm{DFT}_{\frac{N}{2}}\{x[2n]\} + W_N^k \mathrm{DFT}_{\frac{N}{2}}\{x[2n+1]\}$$

- Hence:

$$= X_0\left[\langle k \rangle_{\frac{N}{2}}\right] + W_N^k X_1\left[\langle k \rangle_{\frac{N}{2}}\right]$$

$k = 0..N\text{-}1$

N/2 point DFT defined only for $k = 0..N/2\text{-}1$

Decimation in Time (DIT) FFT

$$\mathrm{DFT}_N\{x[n]\} = \mathrm{DFT}_{\frac{N}{2}}\{x_0[n]\} + W_N^k\,\mathrm{DFT}_{\frac{N}{2}}\{x_1[n]\}$$

- We can evaluate an N-pt DFT as two $N/2$-pt DFTs (plus a few mults/adds)
- But if $\mathrm{DFT}_N\{\bullet\} \sim O(N^2)$
  then $\mathrm{DFT}_{N/2}\{\bullet\} \sim O((N/2)^2) = 1/4\ O(N^2)$

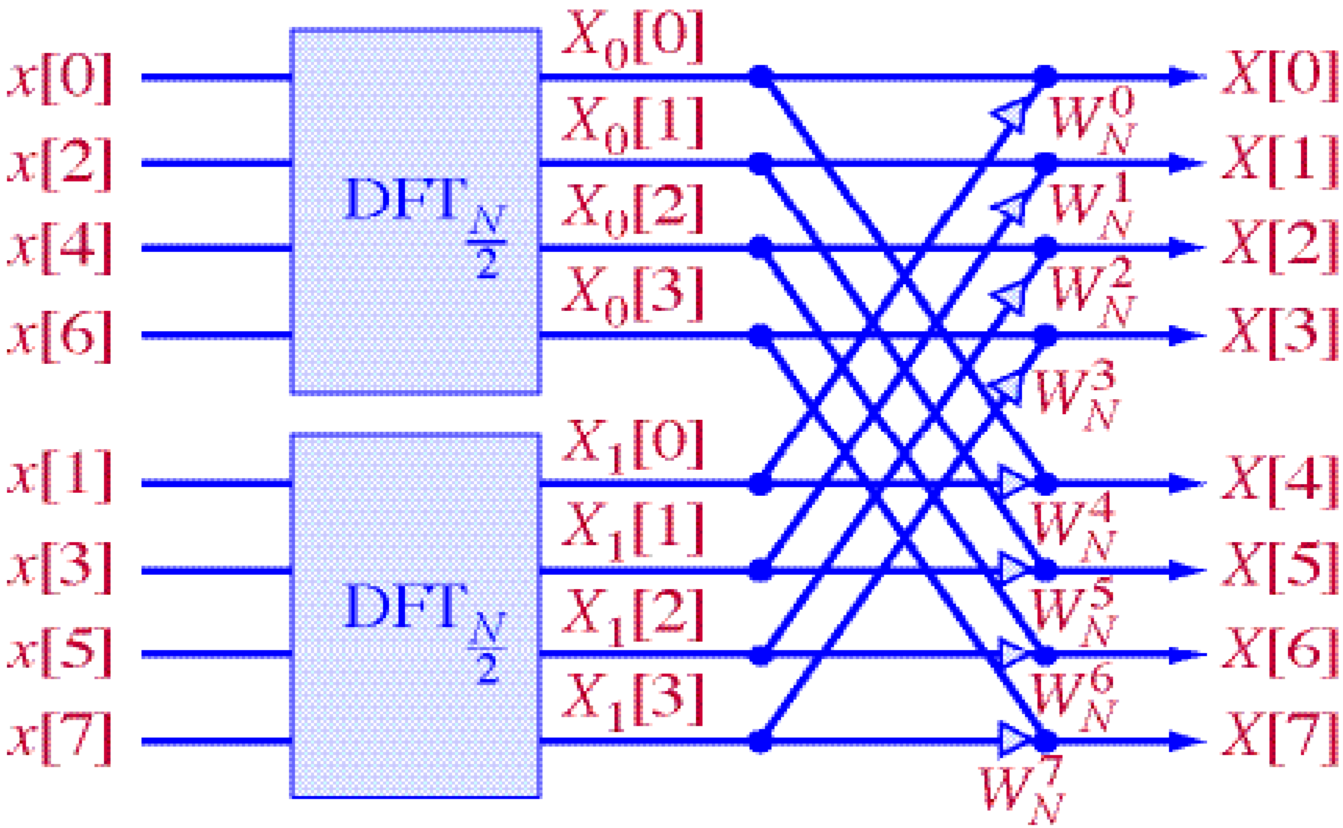
$\Rightarrow$ Total computation $\sim 2 \times 1/4\ O(N^2)$
$= 1/2$ the computation $(+\varepsilon)$ of direct DFT

One-Stage DIT Flowgraph

$$X[k] = X_0\left[\langle k \rangle_{\frac{N}{2}}\right] + W_N^k X_1\left[\langle k \rangle_{\frac{N}{2}}\right]$$

Even points from $x[n]$

Odd points from $x[n]$



Classic FFT structure

"twiddle factors" always apply to odd-terms output ***NOT*** *mirror-image*

Same as $X[0..3]$ except for factors on $X_1[\bullet]$ terms

Multiple DIT Stages

- If decomposing one DFT_N into two smaller $\mathrm{DFT}_{N/2}$'s speeds things up...
  Why not further divide into $\mathrm{DFT}_{N/4}$'s ?

  i.e. $X[k] = X_0\left[\langle k \rangle_{\frac{N}{2}}\right] + W_N^k X_1\left[\langle k \rangle_{\frac{N}{2}}\right]$

  $0 \le k < N$

  and then $X_0[k] = X_{00}\left[\langle k \rangle_{\frac{N}{4}}\right] + W_{\frac{N}{2}}^k X_{01}\left[\langle k \rangle_{\frac{N}{4}}\right]$

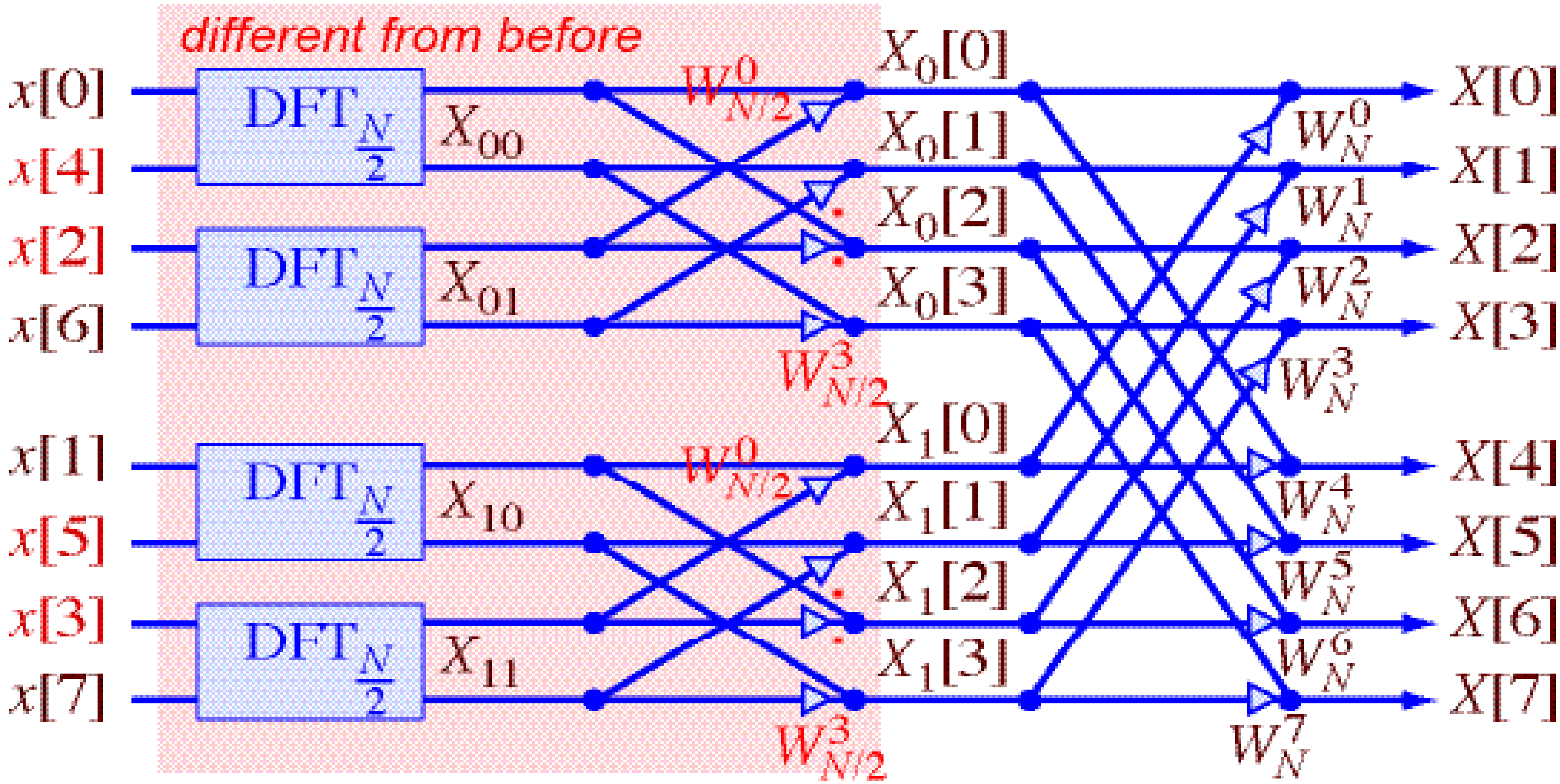
  $0 \le k < N/2$

  *N/4-pt DFT of **even** points in **even** subset of $x[n]$*

  *N/4-pt DFT of **odd** points from **even** subset*

- Similarly, $X_1[k] = X_{10}\left[\langle k \rangle_{\frac{N}{4}}\right] + W_{\frac{N}{2}}^k X_{11}\left[\langle k \rangle_{\frac{N}{4}}\right]$

Two-Stage DIT Flowgraph

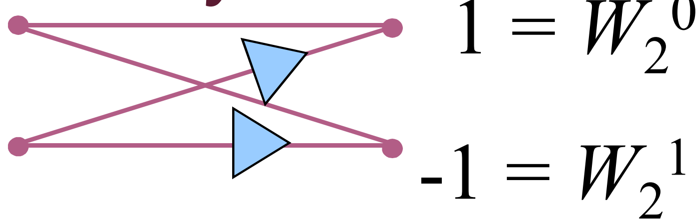


Multi-stage DIT FFT

- Can keep doing this until we get down to 2-pt DFTs:

DFT_2 $\quad X[0] = x[0] + x[1]$, $\quad X[1] = x[0] - x[1]$ $\quad\equiv$

"butterfly" element

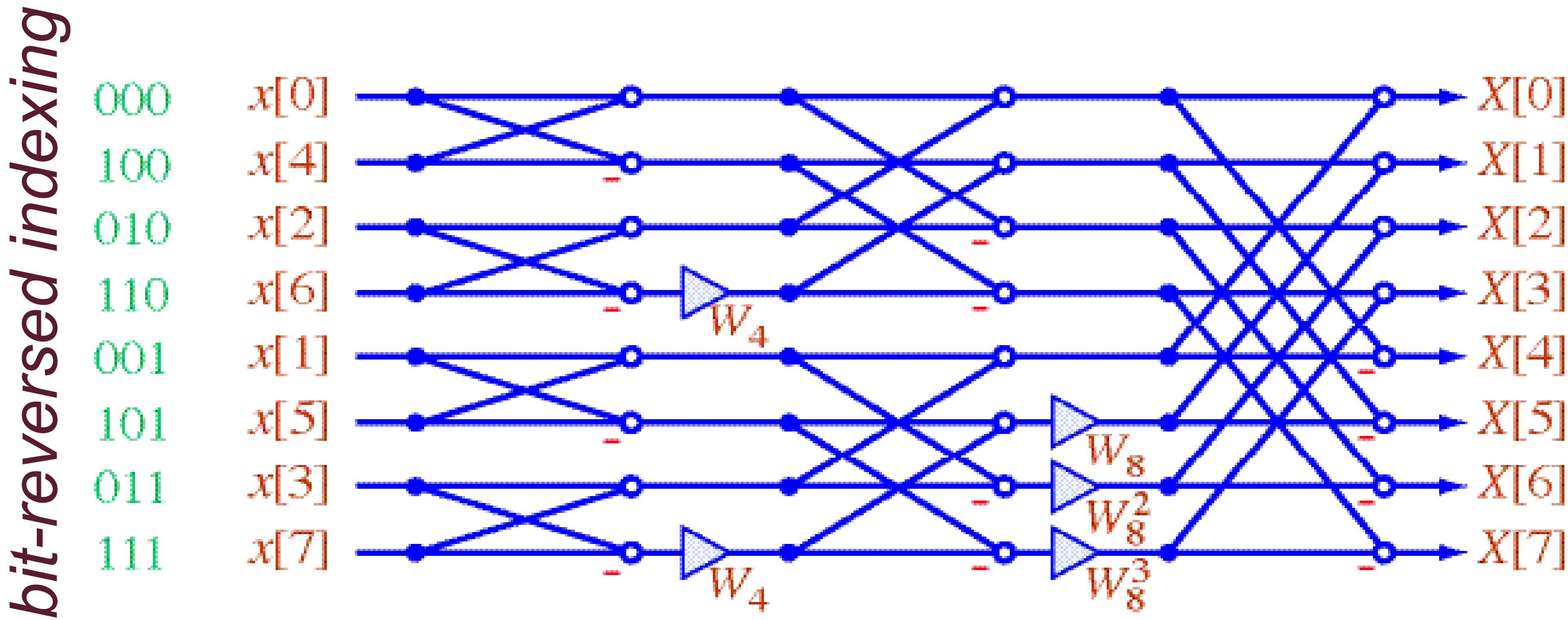


$\rightarrow$ $N = 2^M$-pt DFT reduces to M stages of twiddle factors & summation ($O(N^2)$ part vanishes)

$\rightarrow$ real mults $< M \cdot 4N$, real adds $< 2 \times M \cdot 2N$

$\rightarrow$ complexity ~ $O(N \cdot M) = O(N \cdot \log_2 N)$

8-pt DIT FFT Flowgraph



- -1's absorbed into summation nodes
- $W_N{}^0$ disappears
- 'in-place' algorithm: sequential stages

FFT for Other Values of N

- Having $N = 2^M$ meant we could divide each stage into 2 halves = "radix-2 FFT"
- Same approach works for:
  - $N = 3^M$ radix-3
  - $N = 4^M$ radix-4 - more optimized radix-2
  - etc...
- Composite $N = a \cdot b \cdot c \cdot d \rightarrow$ mixed radix (different N/r point FFTs at each stage)
  - .. or just zero-pad to make $N = 2^M$

Inverse FFT

only differences from forward DFT

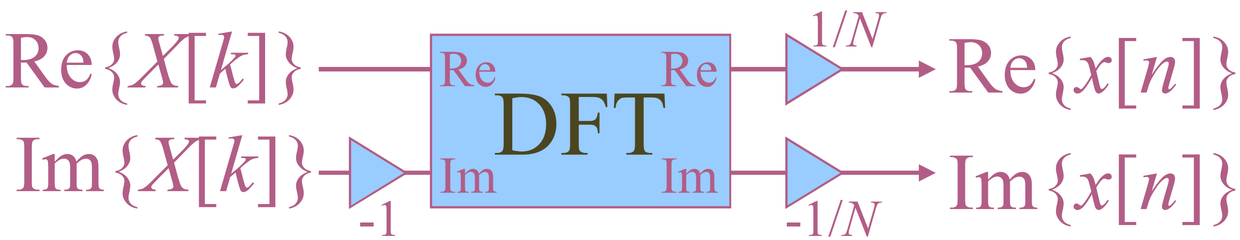
- Recall IDFT: $x[n] = \boxed{\frac{1}{N}} \sum_{k=0}^{N-1} X[k] W_N^{\boxed{-}nk}$

- Thus: ***Forward** DFT of $x'[n] = X^*[k]|_{k=n}$*
  i.e. time sequence made from spectrum

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

- Hence, use FFT to calculate IFFT:

$$x[n] = \frac{1}{N} \left[\sum_{k=0}^{N-1} X^*[k] W_N^{nk} \right]^*$$



In detail:

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

Multiply both sides by N and take complex conjugate

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

Forward DFT of the complex conjugate of the spectrum gives the signal-complex-conjugate & scaled by N!

we calculate IDFT using DFT and conjugate of X instead of creating IDFT

Hence, now divide by N and take complex conjugate again:

$$x[n] = \frac{1}{N}\left[\sum_{k=0}^{N-1} X^*[k]\,W_N^{nk}\right]^*$$

I.e. IDFT is: DFT of complex-conjugate spectrum divided by N and taking complex conjugate of the result

DFT of Real Sequences

- If $x[n]$ is pure-real, DFT wastes mult's

- Real $x[n] \rightarrow \; X[k] = X^*[-k]$

- Given two real sequences, $x[n]$ and $y[n]$ define:

  $v[n] = x[n] + j \cdot y[n]$

- N-pt DFT $V[k] = X[k] + j \cdot Y[k]$

  but: $V[k] + V^*[-k] = X[k] + \overset{X[k]}{X^*[-k]} + j \cdot Y[k] - j \cdot \overset{Y[k]}{Y^*[-k]}$

$\Rightarrow X[k] = {}^1/_2(V[k] + V^*[-k])$, $Y[k] = {}^{-j}/_2(V[k] - V^*[-k])$

- i.e. compute DFTs of **two** N-pt real sequences with a single N-pt DFT

Summary

- The DFT is a discrete-frequency version of the Fourier Transform, suitable for digital implementation
- The FFT is the fast computation enabling communications applications using large DFTs to operated in real time
- The FFT design represents a good example of symmetry analysis in signal processing for communications systems