Fast Fourier Transform

EE 453 / CE 352
Saad Baig

Complex Multiplications Required?

- Each inner product requires N complex multiplications
  - There are N inner products
- Hence we require N^2 multiplications
- However, the first row and first column are all 1s, and *should not be counted as multiplications*
  - There are $2N - 1$ such instances
- Hence, the number of complex multiplications is $N^2 - 2N + 1$, i.e., $(N - 1)^2$

Complex Additions Required?

- Each inner product requires $N-1$ complex additions
  - There are N inner products
- Hence we require $N(N-1)$ complex additions

- The operation count for multiplications and additions assumes that W_N^k has been computed offline and is available in memory
  - If pre-computed values of W_N^k are not available, then the operation count will increase

Complex Additions Required?

- For large N,

  $(N-1)^2 \approx N^2$

  $N(N-1) \approx N^2$

- Hence both multiplications and additions are $O(N^2)$

- If $N = 10^3$, then $O(N^2) = 10^6$, i.e., a million!

- This makes the straightforward method slow and impractical even for a moderately long sequence

The Divide and Conquer Approach

- Suppose N is even and we split the sequence into two halves.
  - Each sequence has $N/2$ points
- Suppose we compute the $\dfrac{N}{2}$ point DFT of each sequence
  - $\text{Multiplications}: \ 2 \times \left(\frac{N}{2}\right)^2 = \frac{N^2}{2}$
- Suppose we are able to combine the individual DFT results to get the originally required DFT
  - Some computational overhead will be consumed to combine the two results
- If $\frac{N^2}{2} + \text{overhead} < N^2$, then this approach will reduce the operation count

The Divide and Conquer Approach

Let $N = 8$

- Straightforward implementation requires, *approximately*, 64 multiplications

- The "divide and conquer" approach requires, *approximately*, $2 \times \left(\frac{8}{2}\right)^2 + \text{overhead}$, i.e., $32 + \text{overhead}$ multiplications

- Questions:
  - Can the two DFTs be combined to get the original DFT ?
  - If so, how ? What is the overhead involved ?
  - Will $32 +$ overhead be less than 64 ?

The Decimation in Time (DIT) Algorithm

- From $\{x_n\}$ form two sequences as follows:

$$\{g_n\} = \{x_{2n}\} \qquad \{h_n\} = \{x_{2n+1}\}$$

- $\{g_n\}$ contains the even-indexed samples, while $\{h_n\}$ contains the odd-indexed samples
- The DFT of $\{x_n\}$ is

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \, W_N^{nk} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x_{2r} \, W_N^{(2r)k} + \sum_{r=0}^{\frac{N}{2}-1} x_{2r+1} \, W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} g_r \, W_N^{(2r)k} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} h_r \, W_N^{(2r)k} \end{aligned}$$

The Decimation in Time (DIT) Algorithm

- But,

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

  and hence

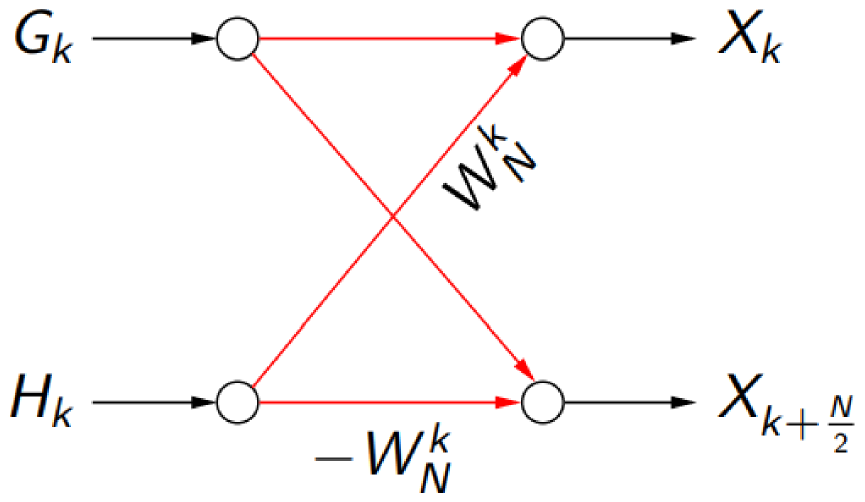
$$\begin{aligned} X_k &= \sum_{r=0}^{\frac{N}{2}-1} g_r\, W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} h_r\, W_{N/2}^{rk} \\ &= G_k + W_N^k H_k \qquad k = 0, 1, \ldots, N-1 \end{aligned}$$

- $\{G_k\}$ and $\{H_k\}$ are $\frac{N}{2}$ point DFTs
- The overhead for combining the two $\frac{N}{2}$ point DFTs is the multiplicative factor W_N^k for $k = 0, 1, \ldots, N-1$
  - W_N^k is called "twiddle factor"

The Decimation in Time (DIT) Algorithm

- The $N/2$ point DFTs $\{G_k\}$ and $\{H_k\}$ are periodic with period $N/2$
  - $G_{k+\frac{N}{2}} = G_k$
    $H_{k+\frac{N}{2}} = H_k$
- $W_N^{k+\frac{N}{2}} = -W_N^k$
- Hence, if $X_k = G_k + W_N^k H_k$, then $X_{k+\frac{N}{2}} = G_k - W_N^k H_k$
  - $W_N^k H_k$ needs to be computed only once for $k = 0$ to $\frac{N}{2} - 1$
- Thus, the multiplication overhead due to the twiddle factors is only $\frac{N}{2}$

Butterfly Diagram

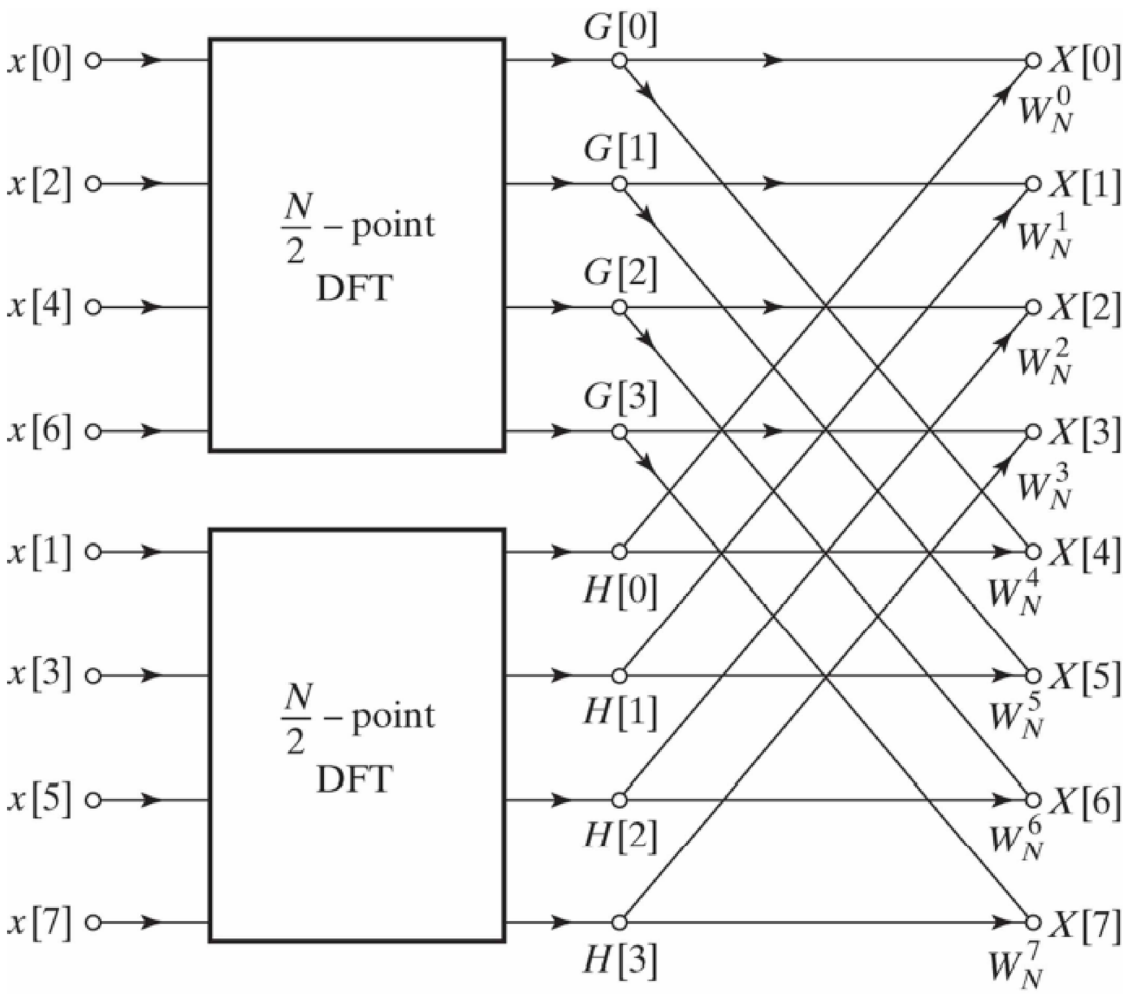


- $X_k = G_k + W_N^k H_k$
- $X_{k+\frac{N}{2}} = G_{k+\frac{N}{2}} + W_N^{k+\frac{N}{2}} H_{k+\frac{N}{2}}$
  $= G_k - W_N^k H_k$

since $W_N^{k+\frac{N}{2}} = e^{-j\frac{2\pi}{N}(k+\frac{N}{2})} = e^{-j\frac{2\pi}{N}k} \cdot e^{-j\frac{2\pi}{N}\frac{N}{2}} = e^{-j\frac{2\pi}{N}k}(-1) = -W_N^k$

The Decimation in Time (DIT) Algorithm

- $X_k = G_k + W_N^k H_k$
- $X_{k+\frac{N}{2}} = G_{k+\frac{N}{2}} + W_N^{k+\frac{N}{2}} H_{k+\frac{N}{2}}$
  $= G_k - W_N^k H_k$



“Divide and Conquer” Results in Savings

- For $N = 8$, the straightforward approach requires, *approximately*, 64 multiplications
- The “Divide and Conquer” approach, after the first stage, requires $32 + 4 =$ 36 multiplications
- Thus, this approach clearly reduces the number of additions and multiplications required

Reusing the "Divide and Conquer" Strategy

- The same idea can be applied for calculating the $\frac{N}{2}$ point DFT of the sequences $\{g_r\}$ and $\{h_r\}$
  - Computational savings can be obtained by dividing $\{g_r\}$ and $\{h_r\}$ into *their* odd- and even-indexed halves

- This idea can be applied recursively $\log_2 N$ times if N is a power of 2

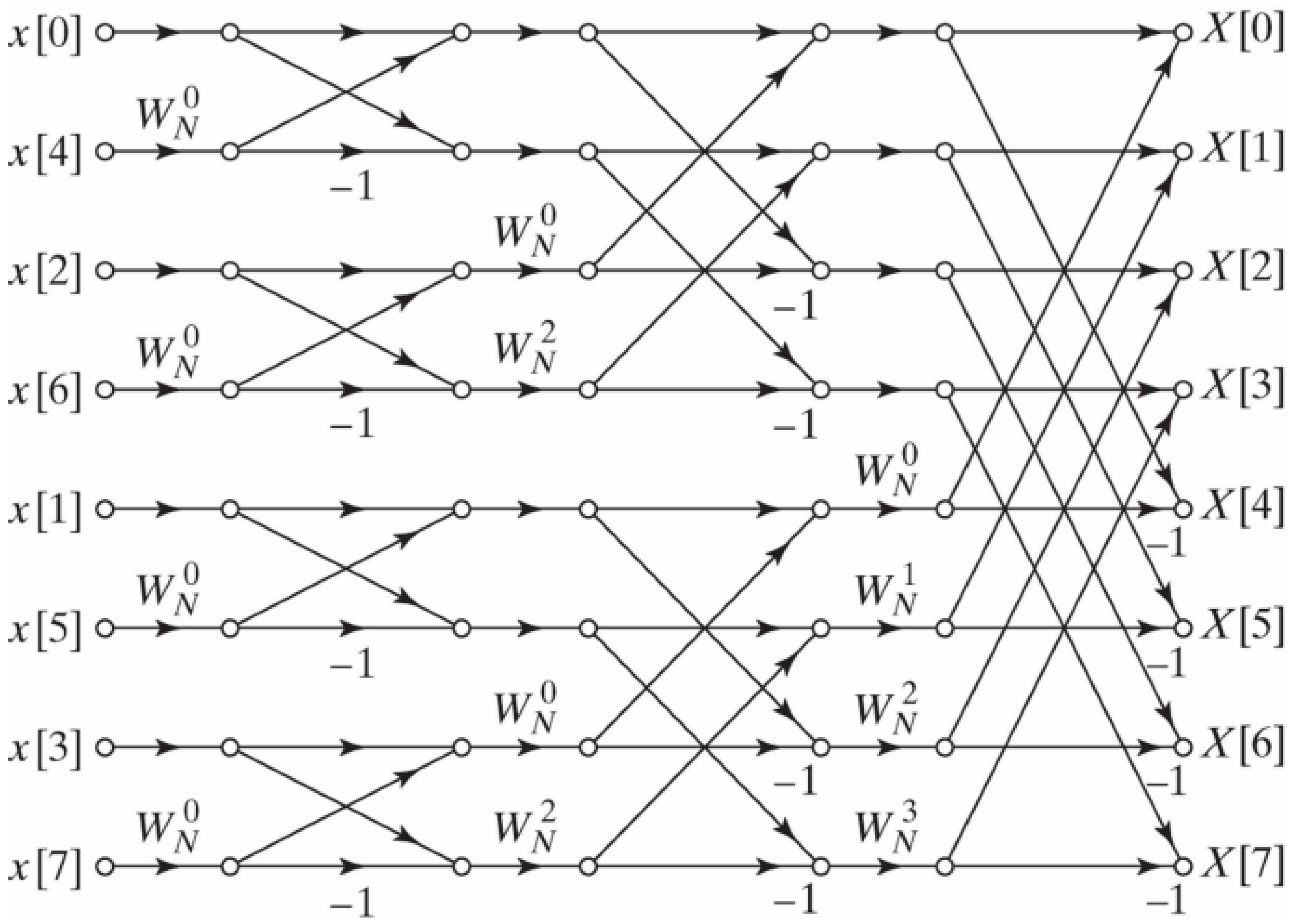
  - Such algorithms are called radix 2 algorithms

- If $N = 2^\gamma$, then the final stage sequences are all of length 2

- For a 2-point sequence $\{p_0, p_1\}$, the DFT coefficients are

$$P_0 = p_0 + p_1 \qquad P_1 = p_0 - p_1$$

DIT Flowgraph for N = 8



Overall Operation Count

- The direct method requires N^2 multiplications
- After the first split, $N^2 \longrightarrow 2\left(\frac{N}{2}\right)^2 + \frac{N}{2}$
  - $\frac{N}{2}$ is due to the *twiddle factors*
- After the second split, $\left(\frac{N}{2}\right)^2 \longrightarrow 2\left(\frac{N}{4}\right)^2 + \frac{N}{4}$

  Hence,

$$N^2 \longrightarrow 2\left(\frac{N}{2}\right)^2 + \underbrace{\frac{N}{2}}_{\text{first stage}} \longrightarrow 4\left(\frac{N}{4}\right)^2 + \underbrace{\frac{N}{2} + \frac{N}{2}}_{\text{second stage}}$$

- Generalizing, if there are $\log_2 N$ stages, the number of multiplications needed will be, *approximately*, $\frac{N}{2}\log_2 N$

Overall Operation Count

- If $W_N^{k+\frac{N}{2}} = -W_N^k$ is not considered, the overhead count will be N and not $\frac{N}{2}$
- In this case,

$$N^2 \longrightarrow 2\left(\frac{N}{2}\right)^2 + \underbrace{N}_{\text{first stage}} \longrightarrow 4\left(\frac{N}{4}\right)^2 + \underbrace{N+N}_{\text{second stage}}$$

- Hence the overall multiplication count will be $N \log_2 N$
- For $N = 1024$

$$N^2 = 1,048,576 \qquad N \log_2 N = 10,240$$

Savings of two orders of magnitude!