Fast Fourier Transform

Advisor: Jian-Jiun Ding
Presenter: Yun-Ting Tsai
Presentation Date: 2022/12/29

Outline

- **Basic Concept of Calculation Simplification**
- **Cooley Tukey Algorithm**

Data Compression

- **Required memory?**

Case 1

1	0	0	1	0
0	1	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	1	0

Case 2

1	1	1	1	1
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0

Computation Simplification

$$\begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} & \boldsymbol{a} \\ \boldsymbol{a} & \boldsymbol{a} \end{bmatrix} \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{bmatrix}$$

Original

$$\begin{cases} y_1 = ax_1 + ax_2 \\ y_2 = ax_1 + ax_2 \end{cases}$$

4 multiplications

Computation simplification

$$\begin{cases} y_1 = a(x_1 + x_2) \\ y_2 = y_1 \end{cases}$$

1 multiplication

Complex Number Multiplication

$$(\boldsymbol{a} + \boldsymbol{jb})(\boldsymbol{c} + \boldsymbol{jd}) = \boldsymbol{ac} - \boldsymbol{bd} + \boldsymbol{j}(\boldsymbol{ad} + \boldsymbol{bc}) = \boldsymbol{e} + \boldsymbol{jf}$$

Original:
4 real multiplications

Computation simplification:

$$\begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} c & -d \\ d & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$= \begin{bmatrix} c & c \\ c & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} 0 & -c-d \\ d-c & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\begin{cases} z_1 = c(a+b) \\ z_2 = z_1 \\ z_3 = (-c-d)b \\ z_4 = (d-c)a \\ e = z_1 + z_3 \\ f = z_2 + z_4 \end{cases}$$

3 real multiplications

Special Case: 3 × 3 Discrete Fourier Transform

- Discrete Fourier Transform: $X[m] = \sum_{n=0}^{2} x[n] e^{-j\frac{2\pi mn}{3}}$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1/2 & -1/2 \\ 1 & -1/2 & -1/2 \end{bmatrix} + j \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$$

Original:

$4N^2 = 36$ real multiplications

→

Computation simplification:

$$\begin{cases} z_1 = -\sqrt{3}/2(x_1 - x_2) \\ z_2 = -z_1 \end{cases}$$

$(0 + 1) \times 2 = 2$ real multiplications

Outline

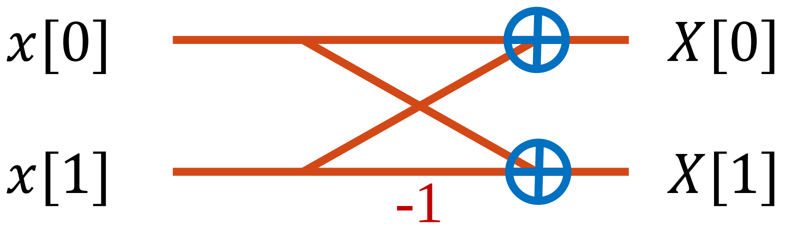
- Basic Concept of Calculation Simplification
- **Cooley Tukey Algorithm**

Basic Concept of Cooley Tukey Algorithm

- Discrete Fourier Transform: $X[m] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi mn}{N}}$

- Special case when N=2: $X[m] = \sum_{n=0}^{1} x[n] e^{-j\frac{2\pi mn}{2}}$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} + j \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$



Derivation of Cooley Tukey Algorithm

When $N = 2^k$

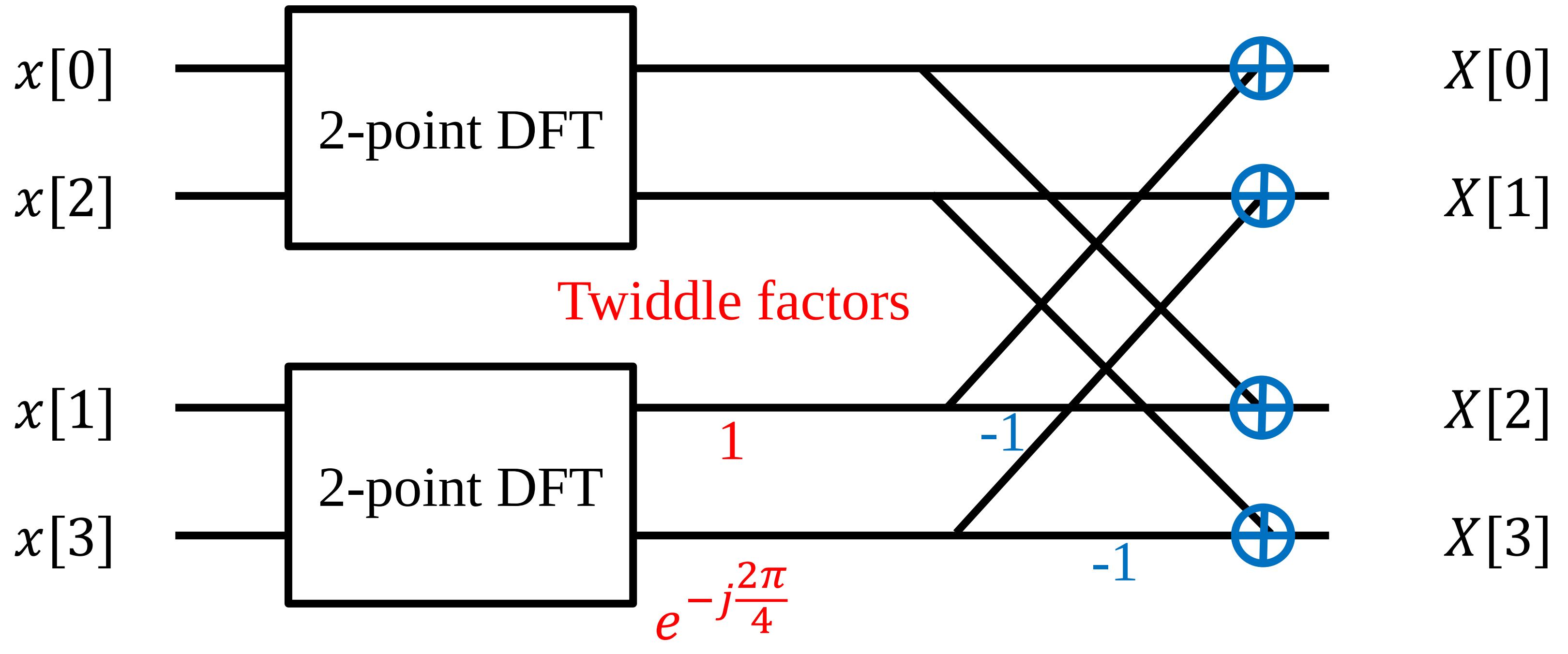
$$X[m] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi mn}{N}}$$

$$= \sum_{n=0}^{N/2-1} x[2n]e^{-j\frac{2\pi m(2n)}{N}} + \sum_{n=0}^{N/2-1} x[2n+1]e^{-j\frac{2\pi m(2n+1)}{N}}$$

$$= \sum_{n=0}^{N/2-1} x_1[n]e^{-j\frac{2\pi mn}{N/2}} + \boxed{e^{-j\frac{2\pi m}{N}}} \sum_{n=0}^{N/2-1} x_2[n]e^{-j\frac{2\pi mn}{N/2}}$$

Twiddle factors

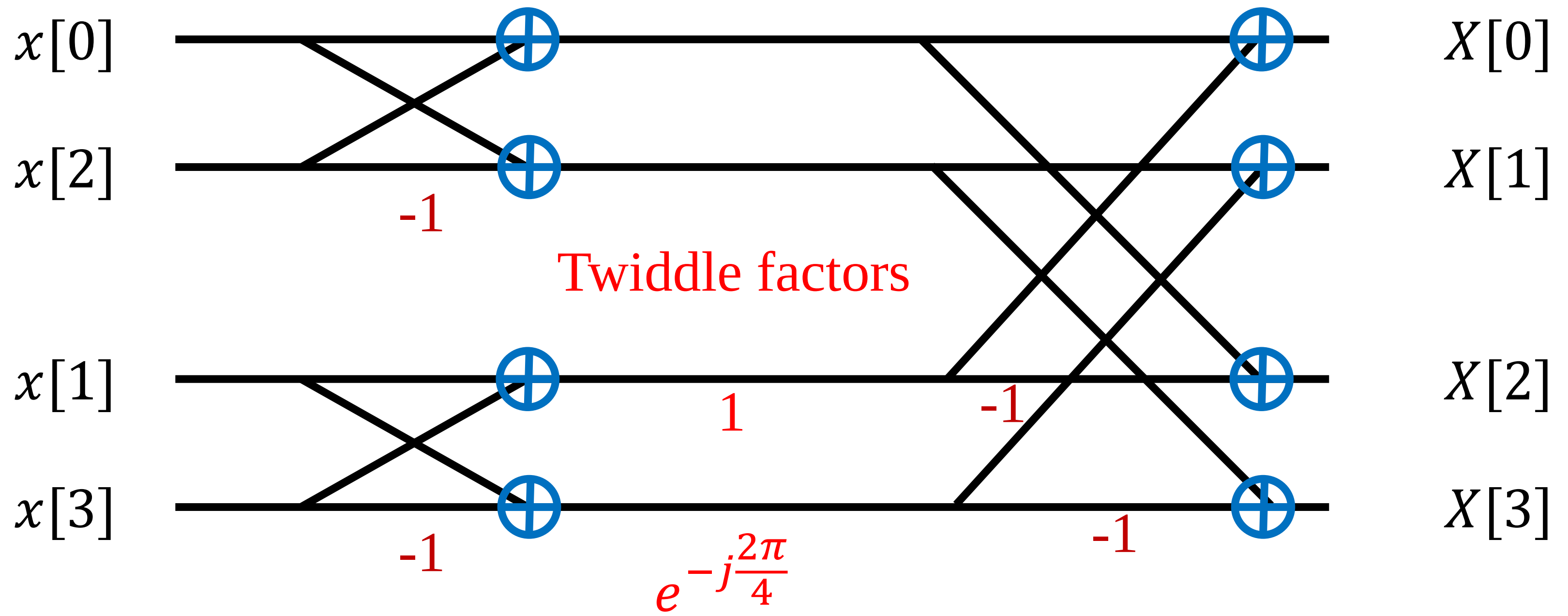
Example: 4-Point DFT



Note:

- $X\left[m+\frac{N}{2}\right] = \sum_{n=0}^{N/2-1} x[n] e^{-j\frac{2\pi(m+\frac{N}{2})n}{N/2}} = e^{-j2\pi n} X[m] = X[m]$
- $e^{-j\frac{2\pi}{N}(m+\frac{N}{2})} = -e^{-j\frac{2\pi}{N}m}$

Example: 4-Point DFT



Note:

- $X\left[m+\frac{N}{2}\right] = \sum_{n=0}^{N/2-1} x[n] e^{-j\frac{2\pi(m+\frac{N}{2})n}{N/2}} = e^{-j2\pi n} X[m] = X[m]$
- $e^{-j\frac{2\pi}{N}(m+\frac{N}{2})} = -e^{-j\frac{2\pi}{N}m}$

Complexity of Cooley Tukey Algorithm

For a 2^k-point DFT

- k stages, and $k-1$ decompositions
- 2^{k-1} twiddle factors between two adjacent stages
- Three real multiplications for one twiddle factor computation
- Total real multiplications: $3 \times 2^{k-1}(k-1) = \frac{3}{2}N(\log_2 N - 1)$
- Complexity of a N-point DFT:

$$O(N \log_2 N)$$