# 第四章 快速傅里叶变换

*Fast Forurier Transform*

# 第四章　快速傅里叶变换

**Fast Forurier Transform**

## 4.5　进一步而减少运算量的措施

华东理工大学信息科学与工程学院 万永菁

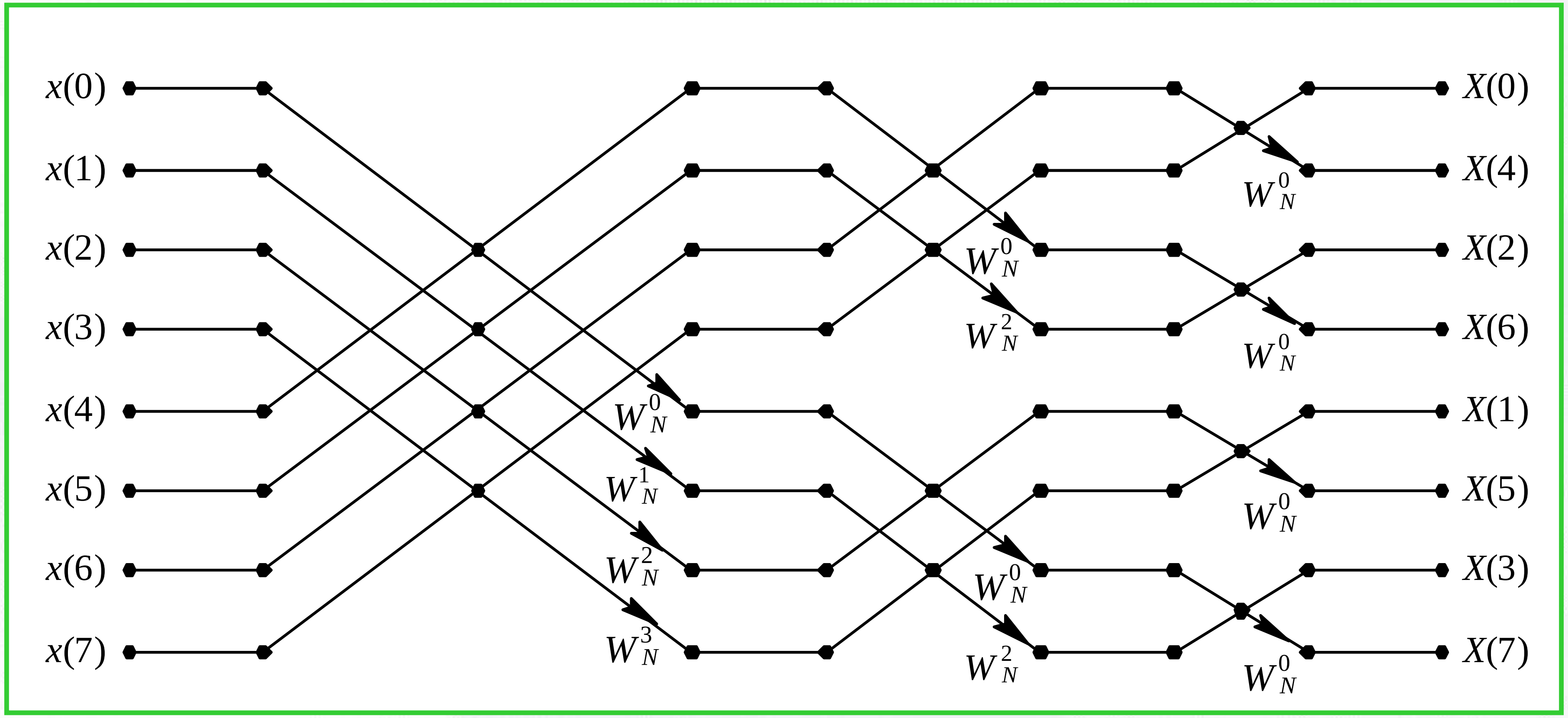
## 进一步减少运算量的措施

- 通过多类蝶形单元运算减少运算量
- 旋转因子的生成
- 实序列的FFT算法

# 4.5 进一步而减少运算量的措施

## 一、通过多类蝶形单元运算减少运算量

**DIF—FFT运算流图($N$=8)**



# 4.5 进一步而减少运算量的措施

一个旋转因子对应一个蝶形单元：

$$\boxed{\begin{aligned} A_L(J) &\Leftarrow A_{L-1}(J) + A_{L-1}(J+B)W_N^p \\ A_L(J+B) &\Leftarrow A_{L-1}(J) - A_{L-1}(J+B)W_N^p \end{aligned}}$$

(1) 若程序中包含了所有的旋转因子，则称为含一类蝶形单元。

(2) 若去掉有关于$W_N^r=\pm 1$旋转因子的乘法运算，则称含二类蝶形单元。

(3) 若在(2)的基础上，再去掉有关于$W_N^r=\pm j$旋转因子的乘法运算，则称含三类蝶形单元。

(4) 若在(3)的基础上，再特殊处理关于$W_N^r=\frac{\sqrt{2}}{2}(1-j)$旋转因子的乘法运算，则称含四类蝶形单元。

# 4.5 进一步而减少运算量的措施



表4.1 基2-FFT在各种蝶形单元下所需的实数乘法的次数

| 蝶形单元种类 | 一类蝶形单元实数乘法次数 | 二类蝶形单元实数乘法次数 | 三类蝶形单元实数乘法次数 | 四类蝶形单元实数乘法次数 |
|---|---|---|---|---|
| $N$=2 | 4 | 0 | 0 | 0 |
| $N$=8 | 48 | 20 | 8 | 4 |
| $N$=32 | 320 | 196 | 136 | 108 |
| $N$=128 | 1792 | 1284 | 1032 | 908 |
| $N$=512 | 9216 | 7172 | 6152 | 5644 |
| $N$=2048 | 45056 | 36868 | 32776 | 30372 |

## 二、旋转因子的生成

将旋转因子中正弦和余弦函数值存放在数组中，在程序执行时查表得到，比直接让程序计算正弦余弦值的效率更高。

## 三、实序列的FFT算法

**1、用一次*N*点FFT计算两个*N*点的实序列的FFT**

一个实输入作为实部，另一个实输入作为虚部，计算完成后将结果进行简单的分解即可得到它们各自的FFT结果。

$$w(n)=x_1(n)+jx_2(n)$$

$$W(k)=X_1(k)+\boxed{jX_2(k)}=W_{ep}(k)+\boxed{W_{op}(k)}$$

$$X_1(k)=\frac{1}{2}[W((k))_N+W^*((N-k))_N]R_N(k)$$

$$X_2(k)=\frac{1}{2j}[W((k))_N-W^*((N-k))_N]R_N(k)$$

## ➢ 实序列的FFT算法

- ◆ 两个$N$点的实序列同时进行FFT运算；

$$w(n)=x_1(n)+jx_2(n)$$

- ◆ 一个$N$点的实序列进行FFT运算

对$N$点的实序列进行分解，用一次$N/2$点的FFT完成运算

## 2、用*N*/2点的FFT计算一个*N*点长的FFT

设$x(n)$为N点实序列，取$x(n)$的偶数点和奇数点分别作为新构造序列$y(n)$的实部和虚部，即：

$$\left.\begin{aligned} x_1(n) &= x(2n) \\ x_2(n) &= x(2n+1) \end{aligned}\right\} \quad n = 0,1,\ldots,\frac{N}{2}-1$$

$$y(n) = x_1(n) + jx_2(n) \qquad n = 0,1,\ldots,\frac{N}{2}-1$$

对$y(n)$进行*N*/2点FFT，输出$Y(k)$，有：

$$\left.\begin{aligned} X_1(k) &= Y_{ep}(k) \\ X_2(k) &= -jY_{op}(k) \end{aligned}\right\}, \quad k = 0,1,\ldots,\frac{N}{2}-1$$

# 4.5　进一步而减少运算量的措施



根据DIT—FFT的思想，序列按奇偶序号分开后有：

$$X(k) = X_1(k) + W_N^k X_2(k)$$

前半部分

$$k = 0,1,\cdots,N/2-1$$

$$X(N/2+k) = X_1(k) - W_N^k X_2(k)$$

后半部分

由于$x(n)$是实序列，所以$X(k)$是共轭对称的，有：

$$X(N-k) = X^*(k) \qquad k = 0,1,\ldots,\frac{N}{2}-1$$

# 4.5 进一步而减少运算量的措施



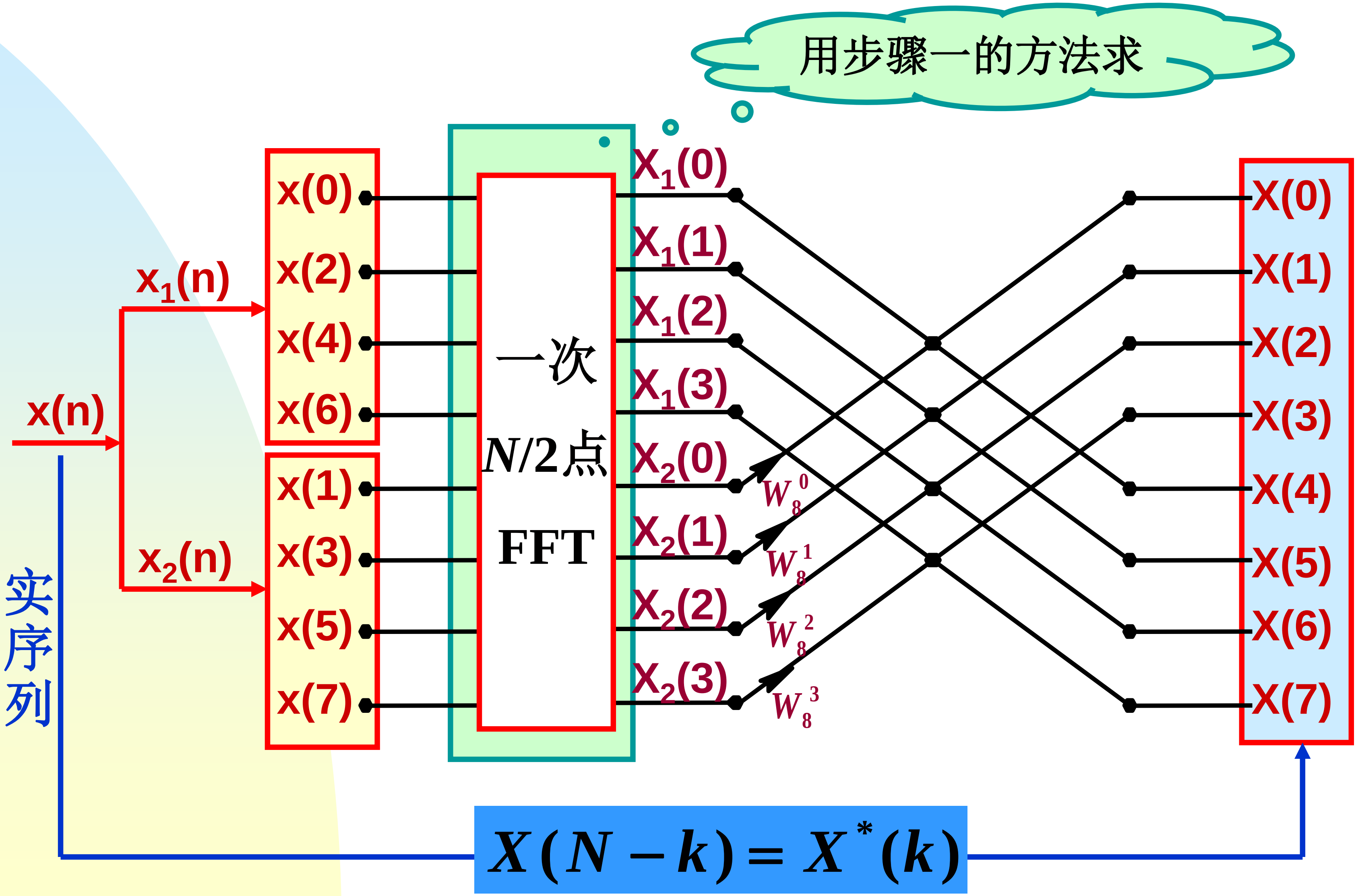
步骤一、分解$x(n)$为偶序号$x_1(n)$和奇序号$x_2(n)$，构造$y(n)$：

$$y(n) = x_1(n) + jx_2(n)$$

然后，对$y(n)$求$N/2$点FFT，并由结果$Y(k)$分解出$X_1(k)$和$X_2(k)$。

$$y(n) = x_1(n) + jx_2(n) \longrightarrow \boxed{N/2\text{点 FFT}} \xrightarrow{Y(k)} \begin{cases} X_1(k) = Y_{ep}(k) \\ X_2(k) = -jY_{op}(k) \end{cases}$$

步骤二、由基2-DIT-FFT算法思想，$x(n)$序列的偶序号的FFT结果与奇序号FFT结果经蝶形运算后能求出$X(k)$的前半部分和后半部分。



# 第四章　快速傅里叶变换

*Fast Forurier Transform*

## 4.5　进一步而减少运算量的措施

华东理工大学信息科学与工程学院 万永菁