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

4.3　基于频率抽取的基-2-FFT快速算法原理

华东理工大学信息科学与工程学院　万永菁

设序列$x(n)$长度为$N=2^M$，首先将$x(n)$前后对半分开，得到两个子序列，其DFT可表示为如下形式：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}=\sum_{n=0}^{N/2-1}x(n)W_N^{kn}+\underline{\sum_{n=N/2}^{N-1}x(n)W_N^{kn}}$$

$$=\sum_{n=0}^{N/2-1}x(n)W_N^{kn}+\underline{\sum_{n=0}^{N/2-1}x(n+N/2)W_N^{k(n+N/2)}}$$

式中，$W_N^{kN/2}=e^{-j\frac{2\pi}{N}\frac{N}{2}k}=e^{-jk\pi}=(-1)^k$

$$=\sum_{n=0}^{N/2-1}x(n)W_N^{kn}+\sum_{n=0}^{N/2-1}\underline{(-1)^k}x(n+N/2)W_N^{kn}\quad \boxed{k=0,1,\ldots N-1}$$

$$\therefore X(k)=\sum_{n=0}^{N/2-1}\left[x(n)+(-1)^k x(n+N/2)\right]W_N^{kn}\quad ,k=0,1,\ldots\ldots N-1$$

按k的奇偶可以把$X(k)$分成两部分：

$$令 k=2r, 及 k=2r+1 \qquad ,r=0,1,2....N/2-1$$

k为偶数时，

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

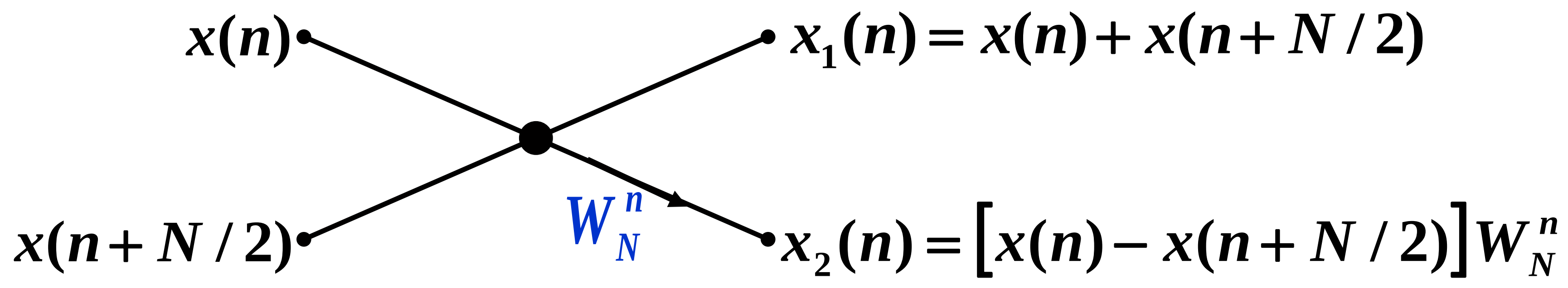
k为奇数时，

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

4.3 基于频率抽取的基-2-FFT快速算法原理

令 $$\begin{cases} x_1(n) = \underline{x(n) + x(n+N/2)} \\ x_2(n) = \underline{\left[x(n) - x(n+N/2)\right]W_N^n} \end{cases} \qquad n = 0,1,\ldots,{}^{N}\!/_{2} - 1$$

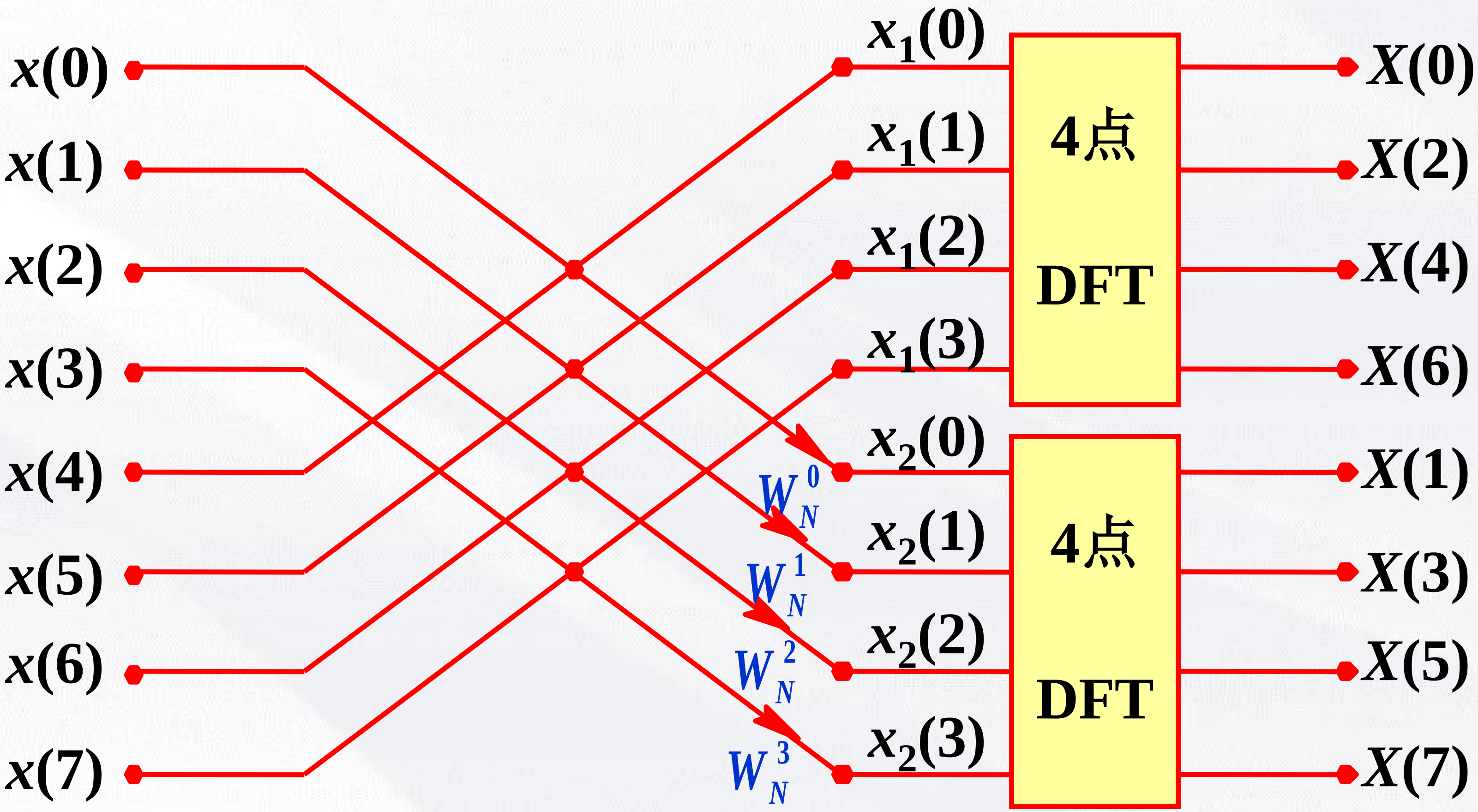
$$\Rightarrow \begin{cases} X_1(k) = X(2r) = \sum_{n=0}^{N/2-1} x_1(n) W_{N/2}^{nr} \\ X_2(k) = X(2r+1) = \sum_{n=0}^{N/2-1} x_2(n) W_{N/2}^{nr} \end{cases}$$



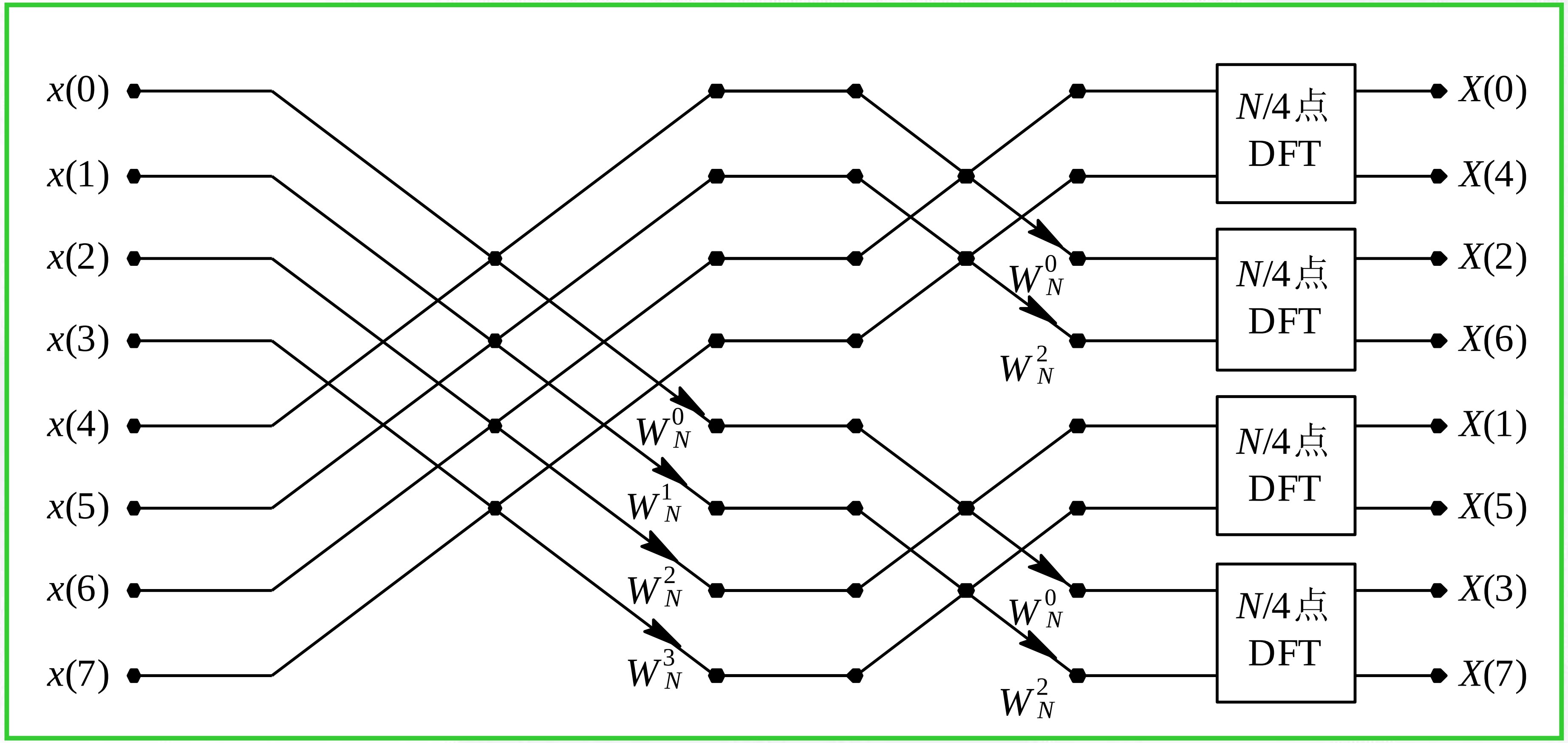
DIF-FFT 算法思路小结

- 将输入时域信号 $x(n)$ 进行前后对半分开的分解操作
- 经过DIF的蝶形运算
- 再经过FFT，得到偶序号和奇序号的$X(k)$

DIF—FFT一次分解运算流图(N=8)

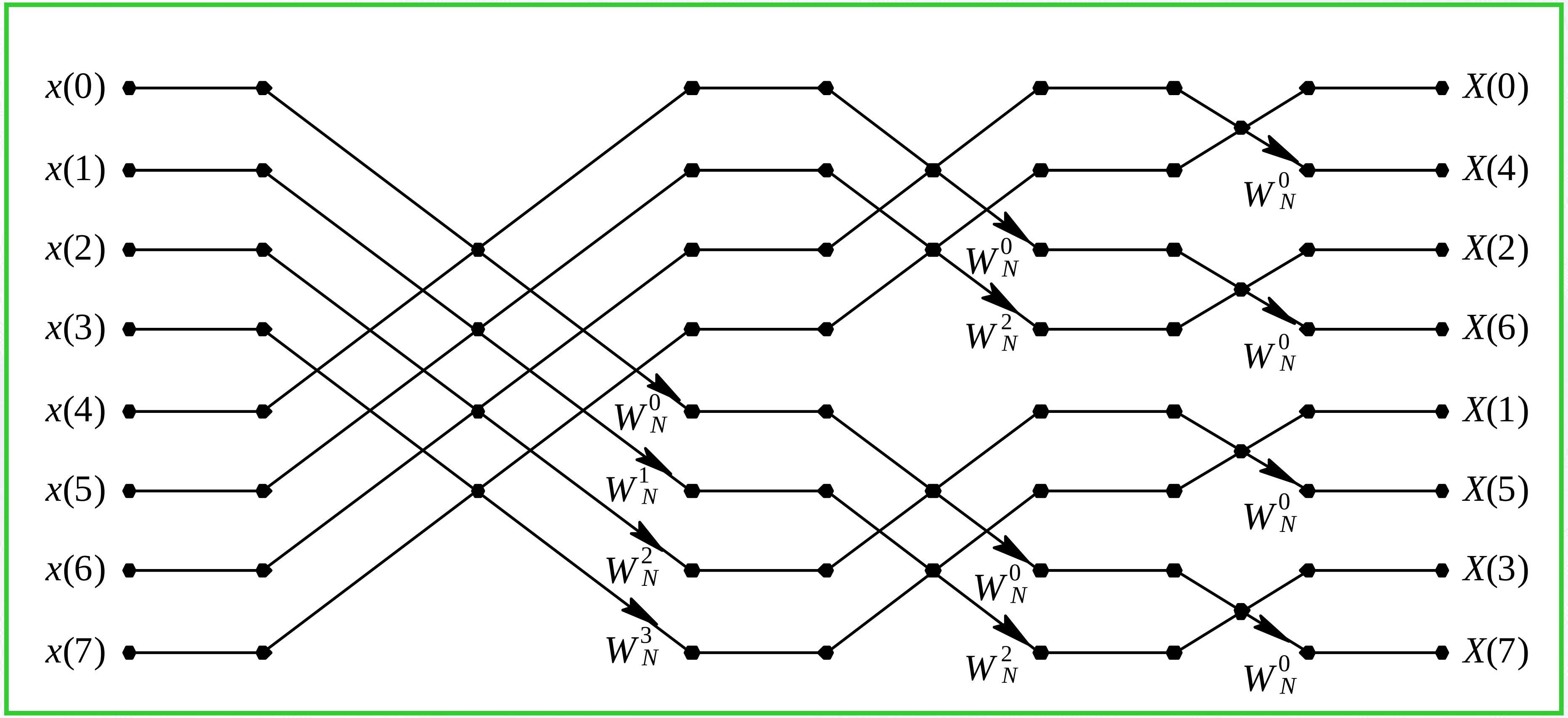


DIF—FFT二次分解运算流图(*N*=8)



4.3 基于频率抽取的基-2-FFT快速算法原理

DIF—FFT运算流图(N=8)



时间抽取算法与频率抽取算法的比较

(1) 频率抽取法和时间抽取法总的计算量是相同的。

复乘：$\frac{N}{2}\log_2 N$	复加：$N\log_2 N$

(2) 频率抽取法和时间抽取法一样，都适用于原位运算，即蝶形的输入和输出占用同一个存储单元。

(3) 均存在码位倒序问题。

(4) 频率抽取法和时间抽取法一样，基本运算也是蝶形运算。但两者的蝶形形式略有不同。

第四章　快速傅里叶变换

Fast Forurier Transform

4.3 基于频率抽取的基-2-FFT快速算法原理

华东理工大学信息科学与工程学院 万永菁