数字信号处理

Digital signal processing

第三章 离散傅里叶变换及快速算法(2)

华中科技大学 自动化学院

- School of Automation.
- Huazhong University of Science and Technology

蔡超

caichao@hust.edu.cn

3.5.1、离散傅里叶变换的计算量

N点有限长序列$x(n)$

$DFT:$

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

$IDFT:$

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

运算量

$$X(k)=\sum_{n=0}^{N-1} x(n) W_N^{nk}$$

	复数乘法	复数加法
一个$X(k)$	N	$N-1$
N个$X(k)$ (N点DFT)	N^2	$N(N-1)$

$$(a+jb)(c+jd)=(ac-bd)+j(ad+cb)$$

	实数乘法	实数加法
一次复乘	4	2
一次复加		2
一个$X(k)$	$4N$	$2N+2(N-1)=2(2N-1)$
N个$X(k)$ (N点DFT)	$4N^2$	$2N(2N-1)$

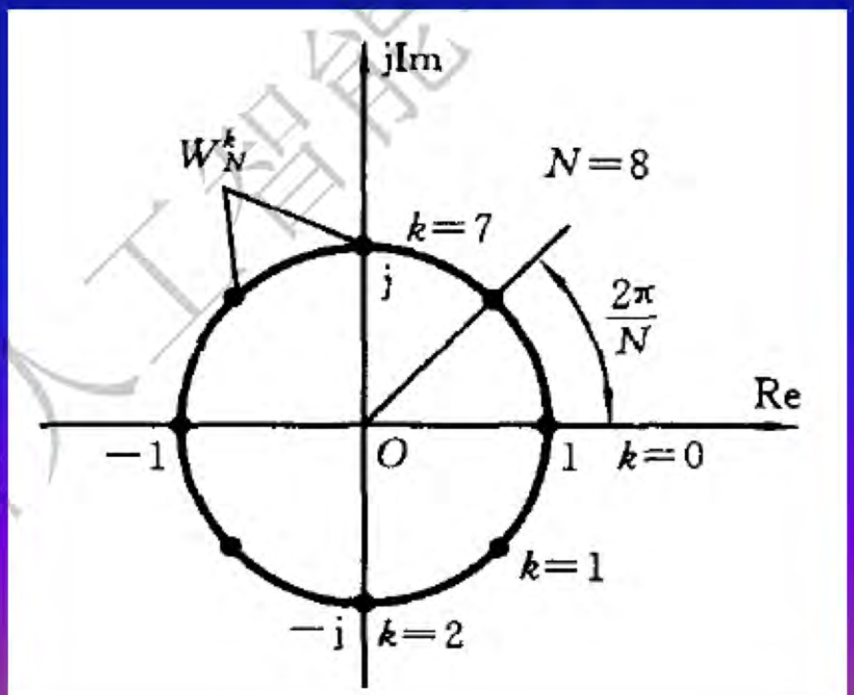
W_N^{nk}的特性 $\quad W_N^{nk}=e^{-j\frac{2\pi}{N}nk}$

周期性 $\quad W_N^{k+rN}=W_N^{k}$

对称性 $\quad (W_N^{nk})^*=W_N^{-nk}=W_N^{N-nk}$

特殊点： $\quad W_N^{0}=1 \quad W_N^{N/2}=-1$

$$W_N^{k+\frac{N}{2}}=W_N^{N/2}W_N^{k}=-W_N^{k} \quad W_N^{2}=W_{N/2}^{1}$$



*FFT*算法的基本思想：

利用*DFT*系数的特性，合并*DFT*运算中的某些项，

把长序列$DFT \rightarrow$短序列*DFT*，从而减少其运算量。

*FFT*算法分类:

- 时间抽选法

  DIT: Decimation-In-Time

- 频率抽选法

  DIF: Decimation-In-Frequency

3.5.2、按时间抽选的基-2FFT算法

1、算法原理

设序列点数 $N=2^L$，L 为整数。

若不满足，则补零

N为2的整数幂的FFT算法称基-2FFT算法。

将序列$x(n)$按n的奇偶分成两组：

$$x(2r)=x_1(r) \qquad r=0,1,\ldots,N/2-1$$
$$x(2r+1)=x_2(r)$$

则$x(n)$的DFT:

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}=\sum_{\text{偶数}n}x(n)W_N^{nk}+\sum_{\text{奇数}n}x(n)W_N^{nk}$$

$$=\sum_{r=0}^{N/2-1}x(2r)W_N^{2rk}+\sum_{r=0}^{N/2-1}x(2r+1)W_N^{(2r+1)k}$$

$$=\sum_{r=0}^{N/2-1}x_1(r)\left(W_N^2\right)^{rk}+W_N^k\sum_{r=0}^{N/2-1}x_2(r)\left(W_N^2\right)^{rk}$$

因为：$W_N^2=W_{N/2}^1$,所以：

$$=\sum_{r=0}^{N/2-1}x_1(r)W_{N/2}^{rk}+W_N^k\sum_{r=0}^{N/2-1}x_2(r)W_{N/2}^{rk}$$

$$=X_1(k)+W_N^kX_2(k)\qquad k=0,1,\ldots N/2-1$$

再利用周期性求$X(k)$的后半部分

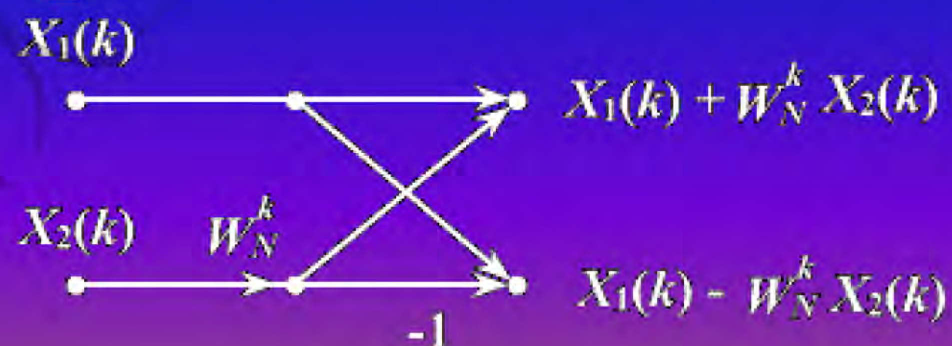
$\because X_1(k), X_2(k)$是以$N/2$为周期的

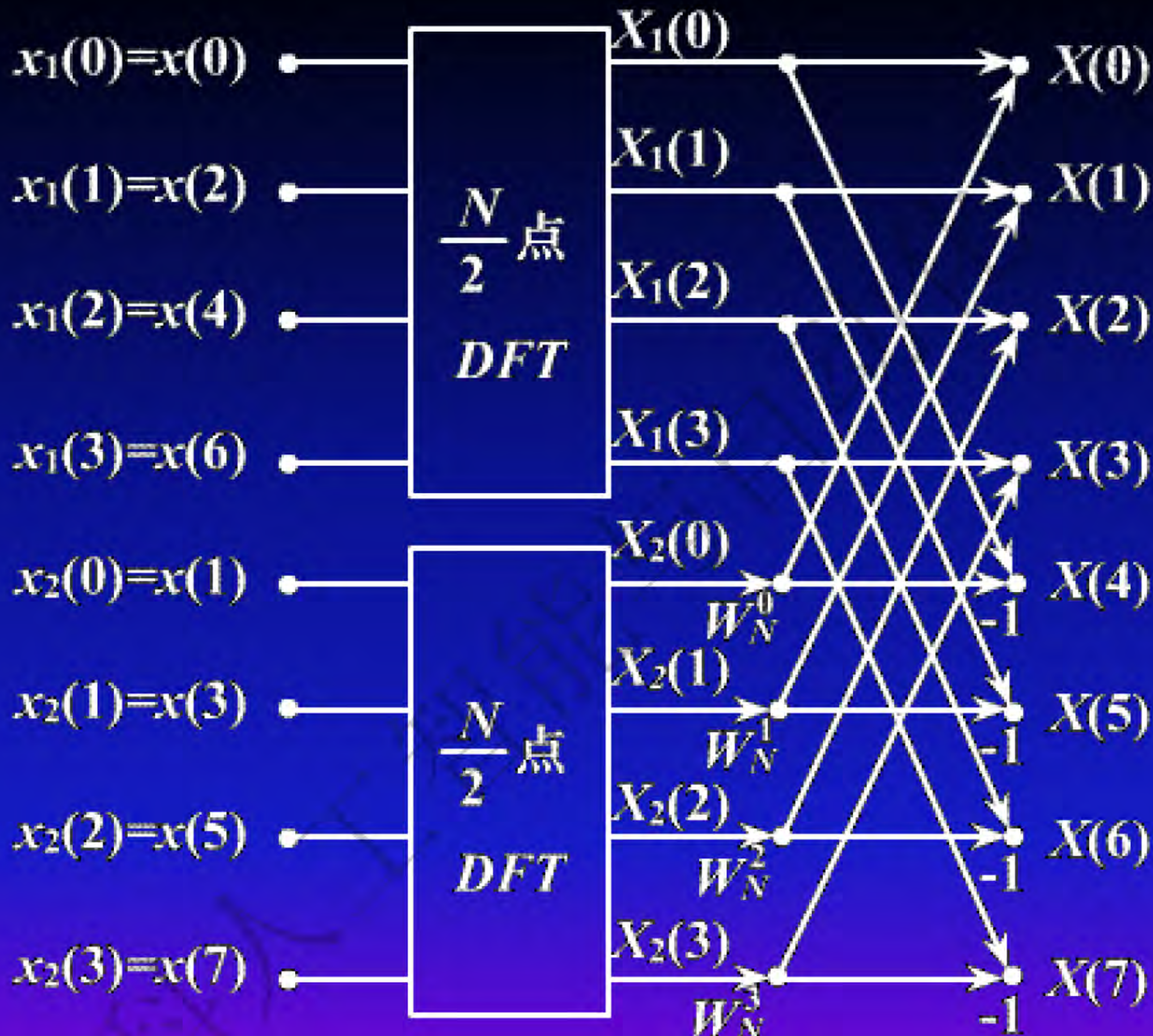
$$\therefore X_1\left(k+\frac{N}{2}\right)=X_1(k) \qquad X_2\left(k+\frac{N}{2}\right)=X_2(k)$$

又$W_N^{k+\frac{N}{2}}=W_N^{N/2}W_N^k=-W_N^k$

$$\therefore \begin{cases} X(k)=X_1(k)+W_N^k X_2(k) \\ X(k+\frac{N}{2})=X_1(k)-W_N^k X_2(k) \end{cases} \quad k=0,1,\ldots,N/2-1$$

时间抽取算法蝶形运算流程图





按时间抽选，将一个N点DFT分解为两个$N/2$点DFT

分解后的运算量:

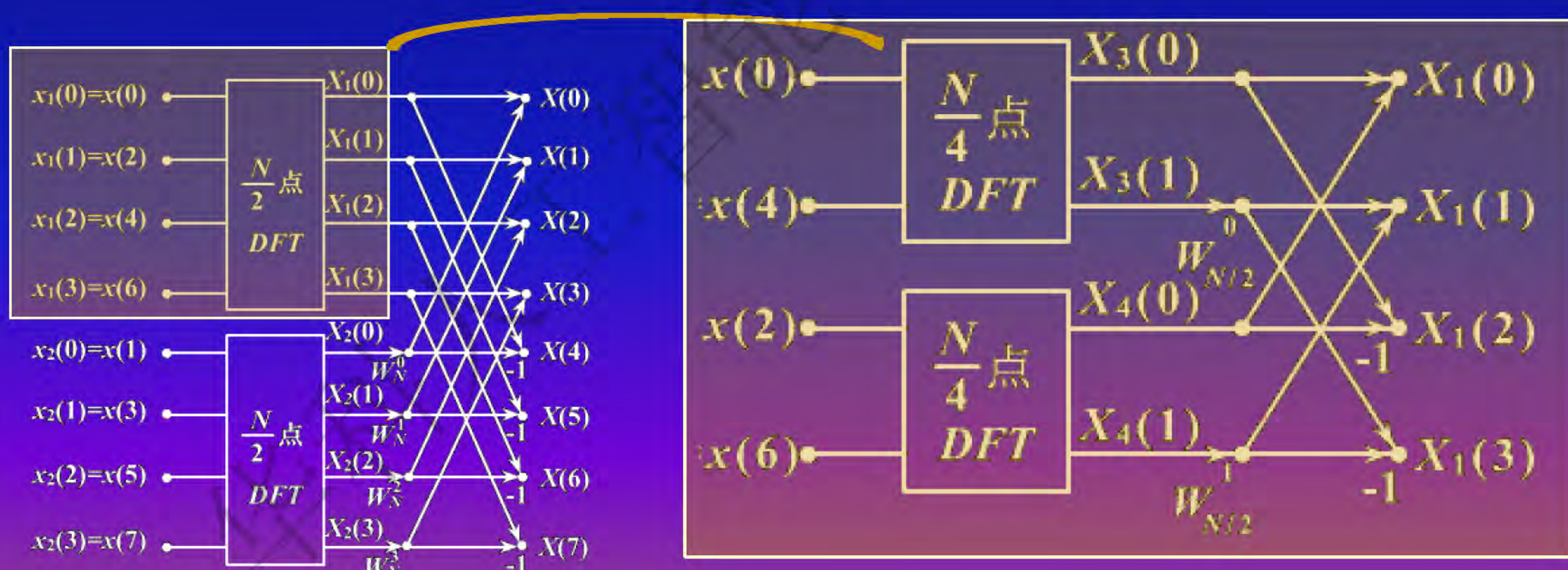
	复数乘法	复数加法
一个$N/2$点DFT	$(N/2)^2$	$N/2\,(N/2-1)$
两个$N/2$点DFT	$N^2/2$	$N(N/2-1)$
一个蝶形	1	2
$N/2$个蝶形	$N/2$	N
总计	$N^2/2+N/2$ $\approx N^2/2$	$N(N/2-1)+N$ $\approx N^2/2$

运算量减少了近一半

$N/2$仍为偶数，进一步分解：$N/2 \rightarrow N/4$

$$\begin{cases} x_1(2l) = x_3(l) \\ x_1(2l+1) = x_4(l) \end{cases} \qquad l = 0,1,\ldots,N/4-1$$

$$\begin{cases} X_1(k) = X_3(k) + W_{N/2}^{k} X_4(k) \\ X_1(k+\frac{N}{4}) = X_3(k) - W_{N/2}^{k} X_4(k) \end{cases} \qquad k = 0,1,\ldots,\frac{N}{4}-1$$



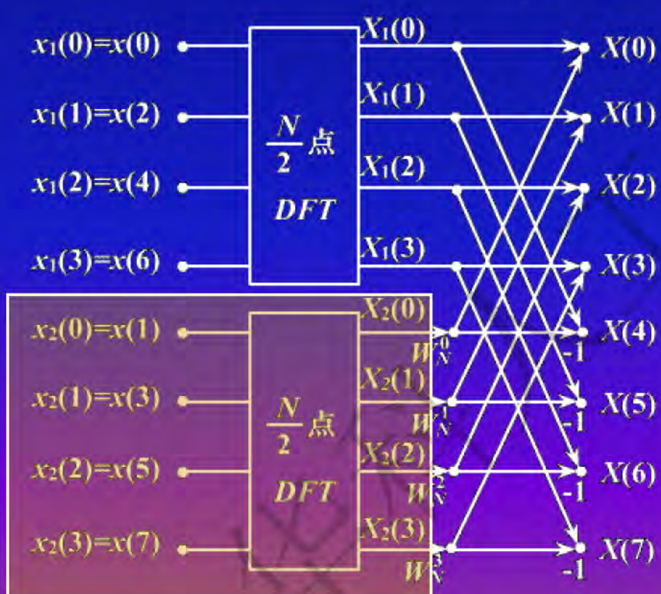
同理：

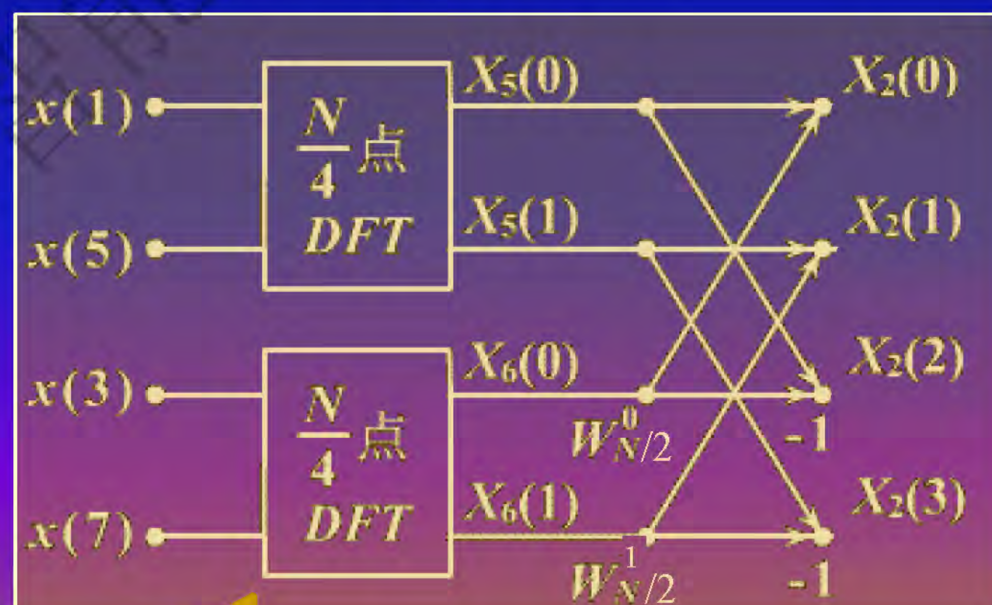
$$\begin{cases} X_2(k) = X_5(k) + W_{N/2}^{k} X_6(k) \\ X_2(k+\dfrac{N}{4}) = X_5(k) - W_{N/2}^{k} X_6(k) \end{cases} \quad k = 0,1,\ldots,\frac{N}{4}-1$$

其中：

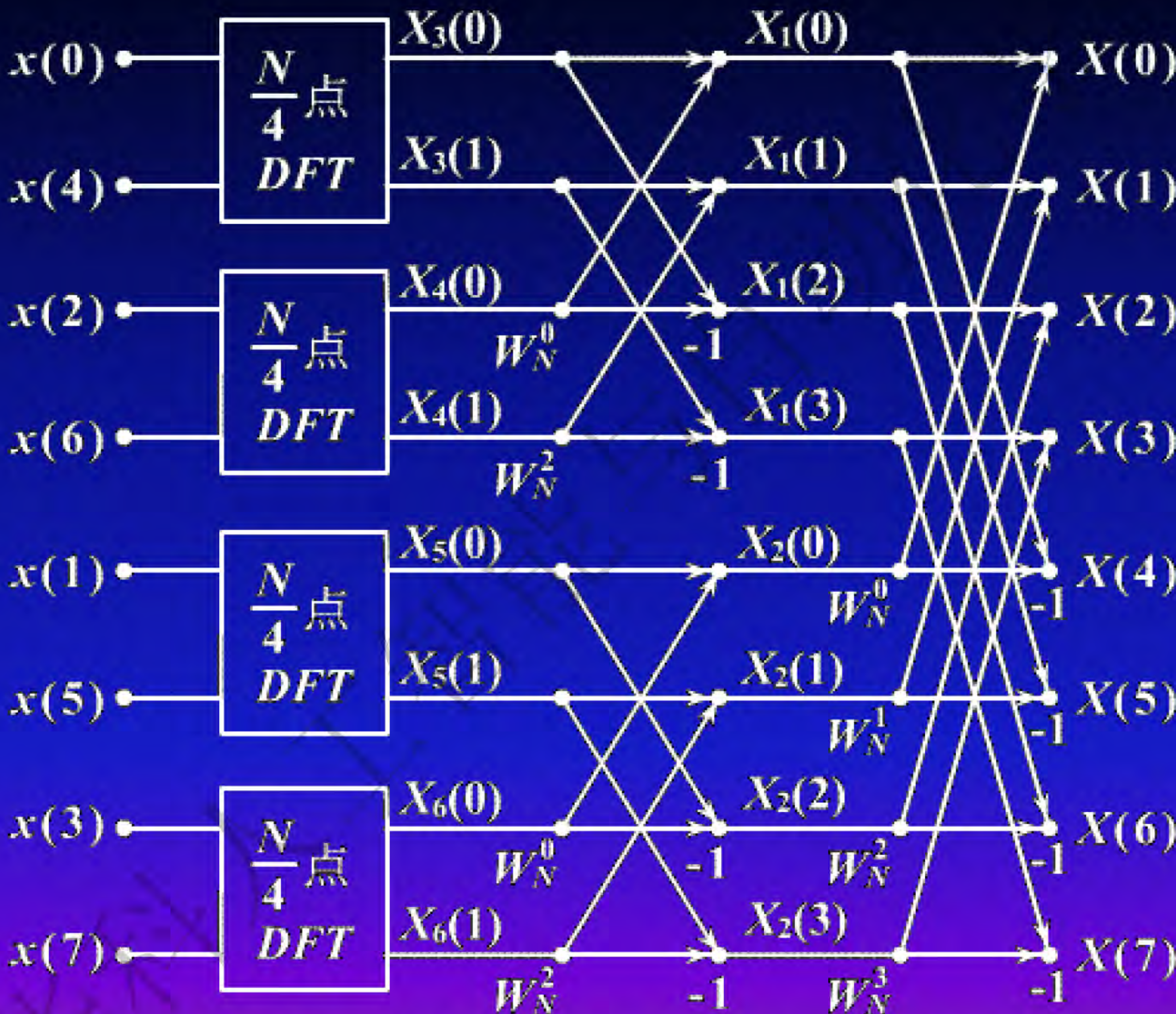
$$X_5(k) = DFT[x_5(l)] = DFT[x_2(2l)]$$

$$X_6(k) = DFT[x_6(l)] = DFT[x_2(2l+1)] \quad l = 0,1,\ldots,N/4-1$$





统一系数：$W_{N/2}^{k} \to W_{N}^{2k}$



按时间抽取，将一个N点DFT分解为四个N/4点DFT(N=8)

这样逐级分解，直到2点DFT

当$N=8$时，即分解到$X_3(k)$，$X_4(k)$，$X_5(k)$，$X_6(k)$，$k=0, 1$

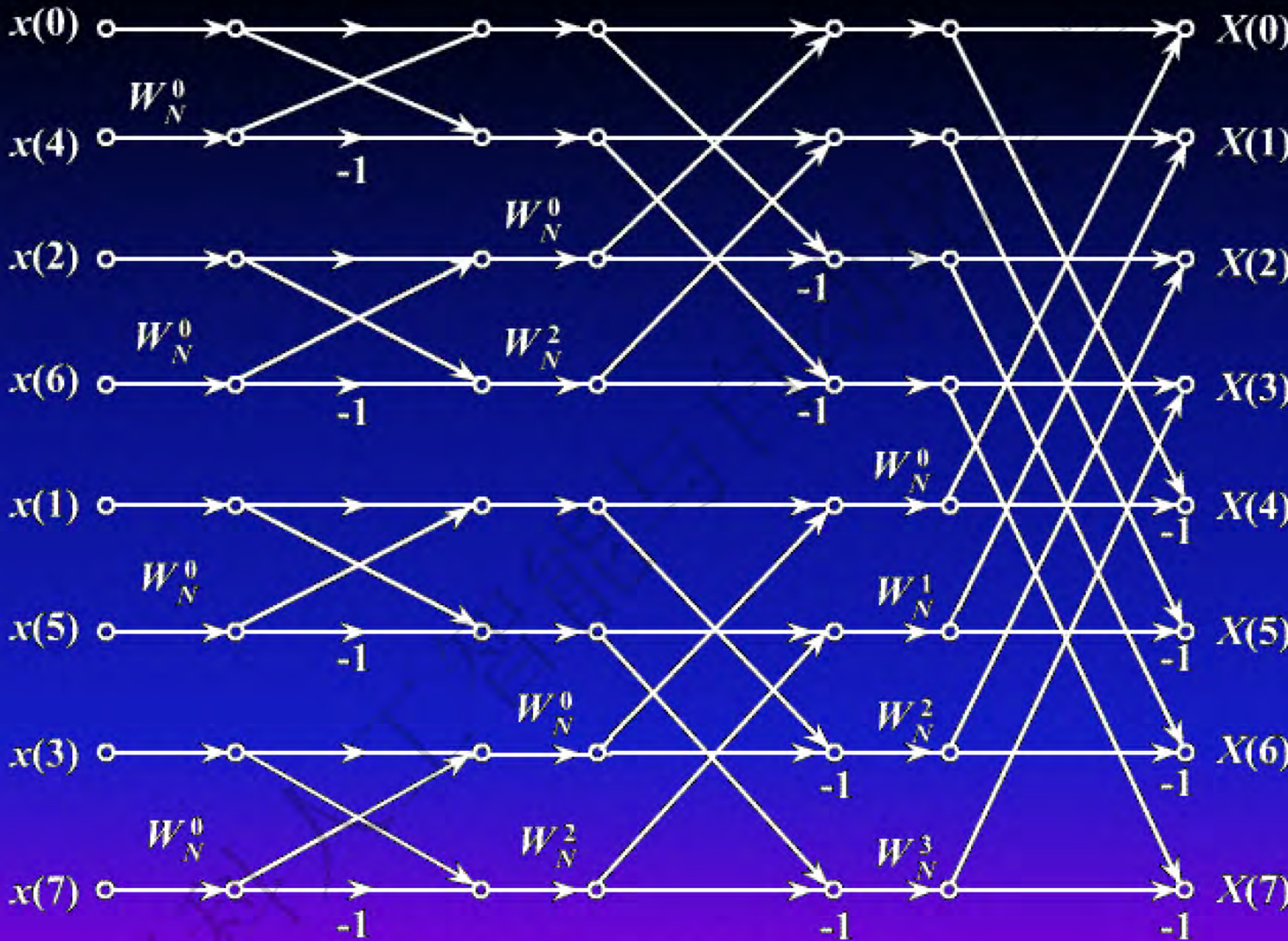
$$X_3(k)=\sum_{l=0}^{N/4-1} x_3(l)W_{N/4}^{lk}=\sum_{l=0}^{1} x_3(l)W_{N/4}^{lk} \qquad k=0,1$$

$$\begin{cases} X_3(0)=x_3(0)W_2^0+W_2^0 x_3(1)=x(0)+W_{N/4}^0 x(4) \\ X_3(1)=x_3(0)W_2^0+W_2^1 x_3(1)=x(0)-W_{N/4}^0 x(4) \end{cases}$$

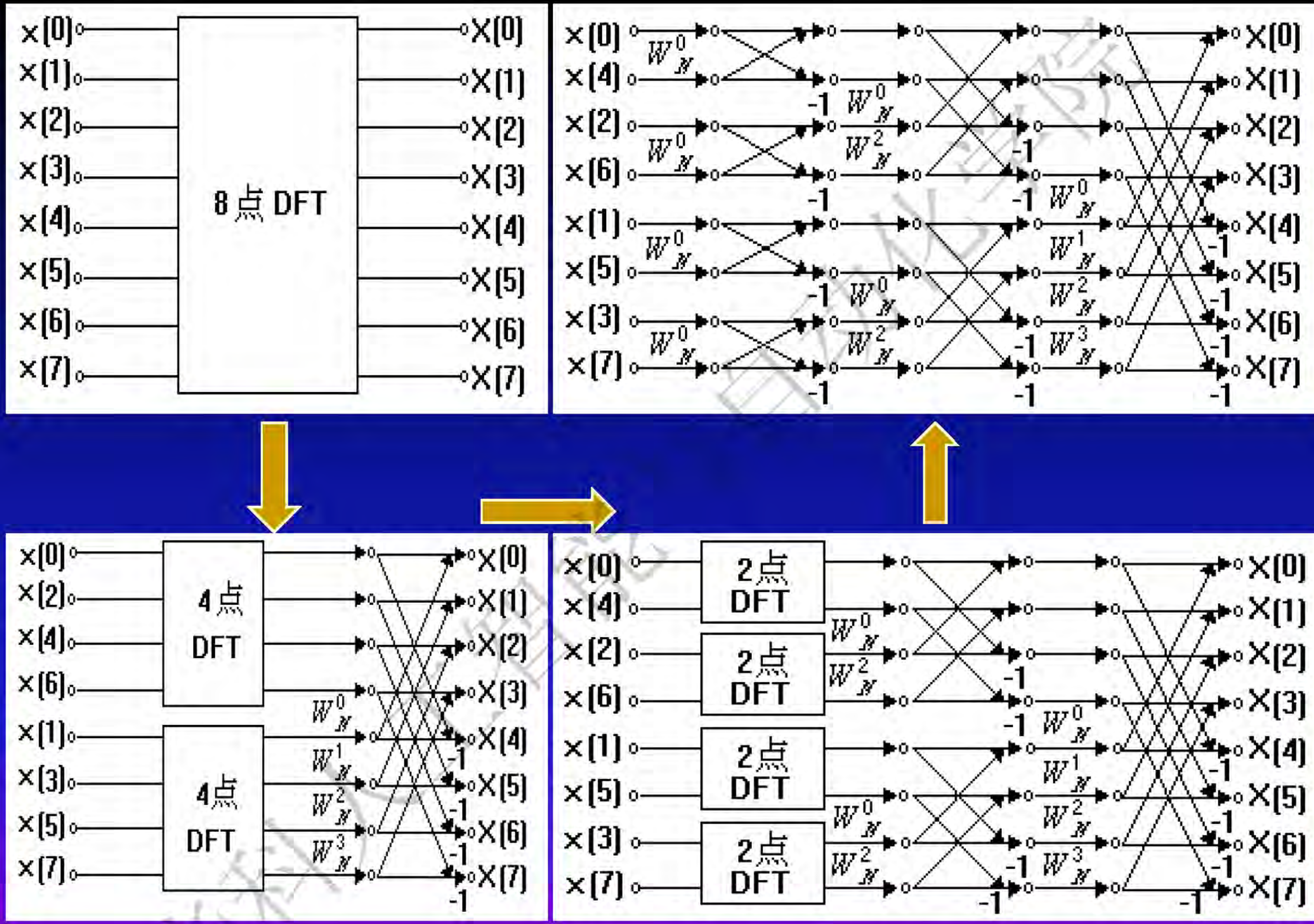
$$X_4(k)=\sum_{l=0}^{N/4-1} x_4(l)W_{N/4}^{lk}=\sum_{l=0}^{1} x_4(l)W_{N/4}^{lk} \qquad k=0,1$$

W_N^0

$$\begin{cases} X_4(0)=x_4(0)W_2^0+W_2^0 x_4(1)=x(2)+W_{N/4}^0 x(6) \\ X_4(1)=x_4(0)W_2^0+W_2^1 x_4(1)=x(2)-W_{N/4}^0 x(6) \end{cases}$$



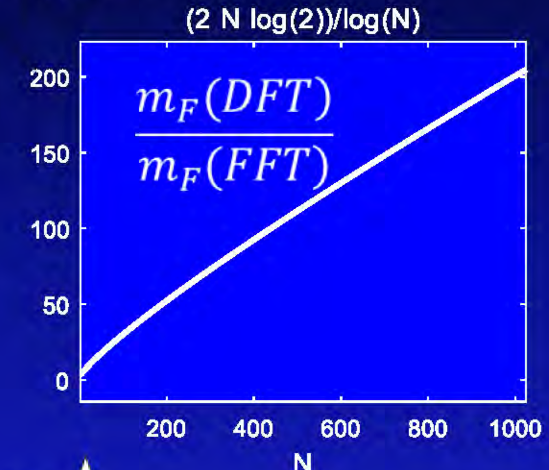
时间抽选的FFT流程图(N=8)



按时间抽取的分解过程及完整流图

2、运算量

当$N=2^L$时，共有L级蝶形，每级$N/2$个蝶形，每个蝶形有1次复数乘法2次复数加法。

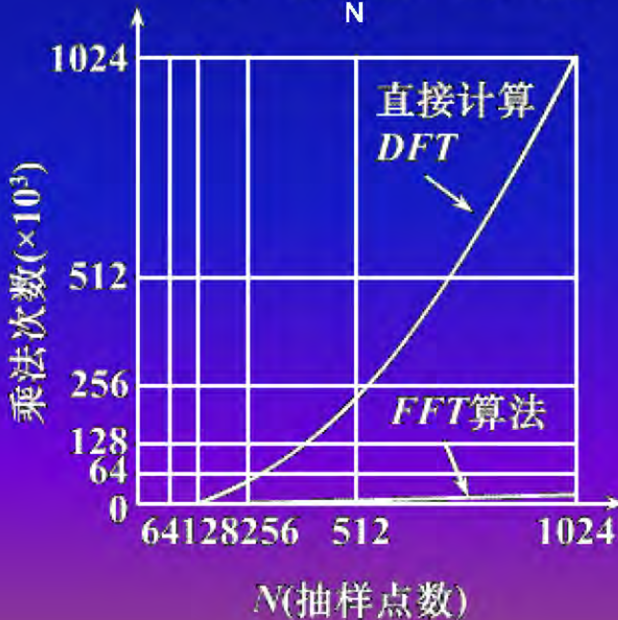


复数乘法： $m_F=\frac{N}{2}L=\frac{N}{2}\log_2 N$

复数加法： $a_F=NL=N\log_2 N$

比较DFT

$$\frac{m_F(DFT)}{m_F(FFT)}=\frac{N^2}{\frac{N}{2}\log_2 N}=\frac{2N}{\log_2 N}$$



3.5.3算法特点

- FFT算法的特点

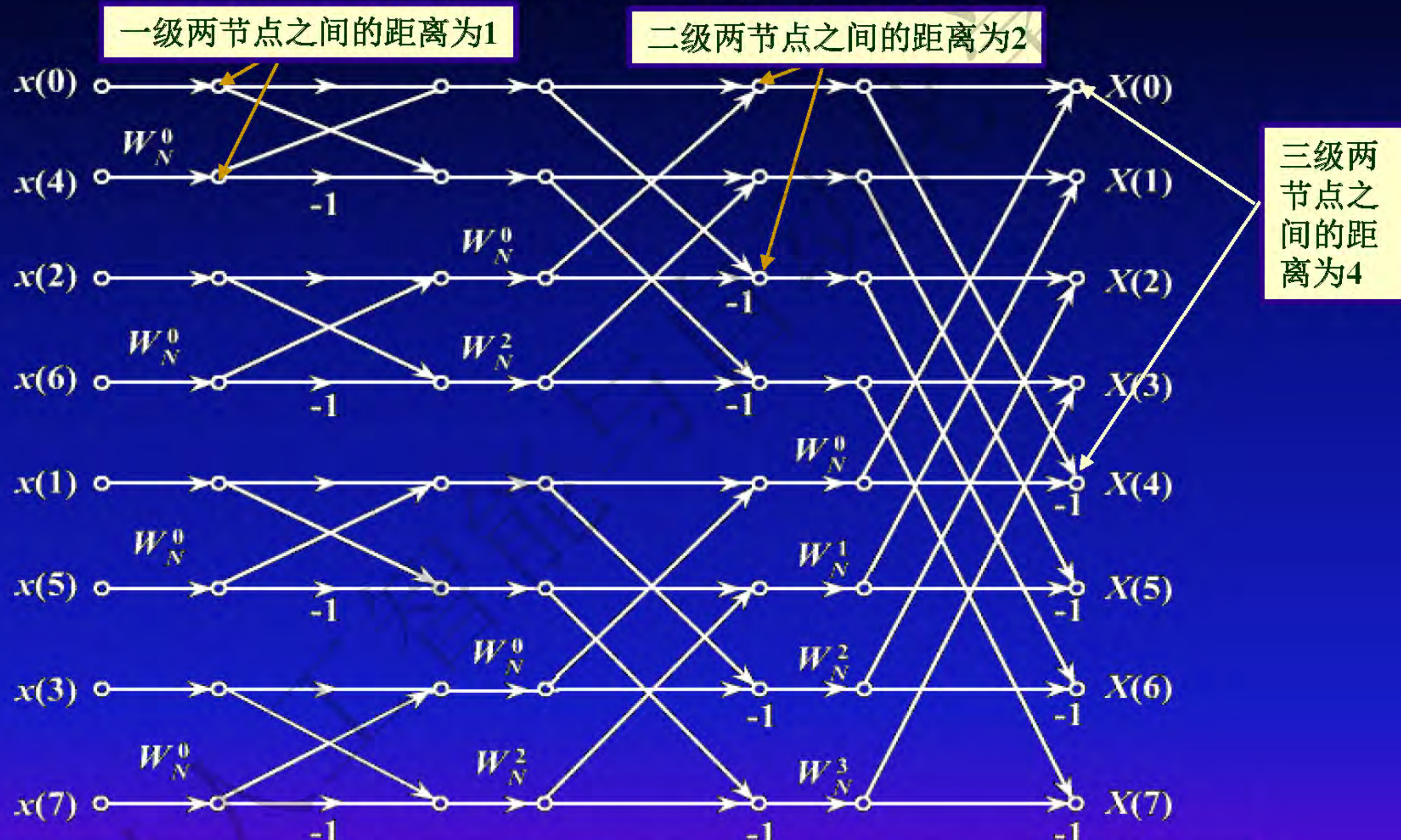
观察完整的FFT流图能发现有两个特点:

倒码和原位运算

1）倒位序（倒码,变址）$x(n)$ $n=(n_2n_1n_0)_2$

倒位序 $\hat{n}=(n_0n_1n_2)_2$		自然序 $n=(n_2n_1n_0)_2$	
000	0	0	000
100	4	1	001
010	2	2	010
110	6	3	011
001	1	4	100
101	5	5	101
011	3	6	110
111	7	7	111

蝶形运算两节点之间的距离



规律：对于共L级的蝶形而言，其m级蝶形运算的节点间的距离为 2^{m-1}

2）同址（原位）计算

对$N=2^L$点FFT，输入倒位序，输出自然序

第m级运算：

$$\begin{cases} X_m(k)=X_{m-1}(k)+X_{m-1}(k+2^{m-1})W_N^r \\ X_m(k+2^{m-1})=X_{m-1}(k)-X_{m-1}(k+2^{m-1})W_N^r \end{cases}$$

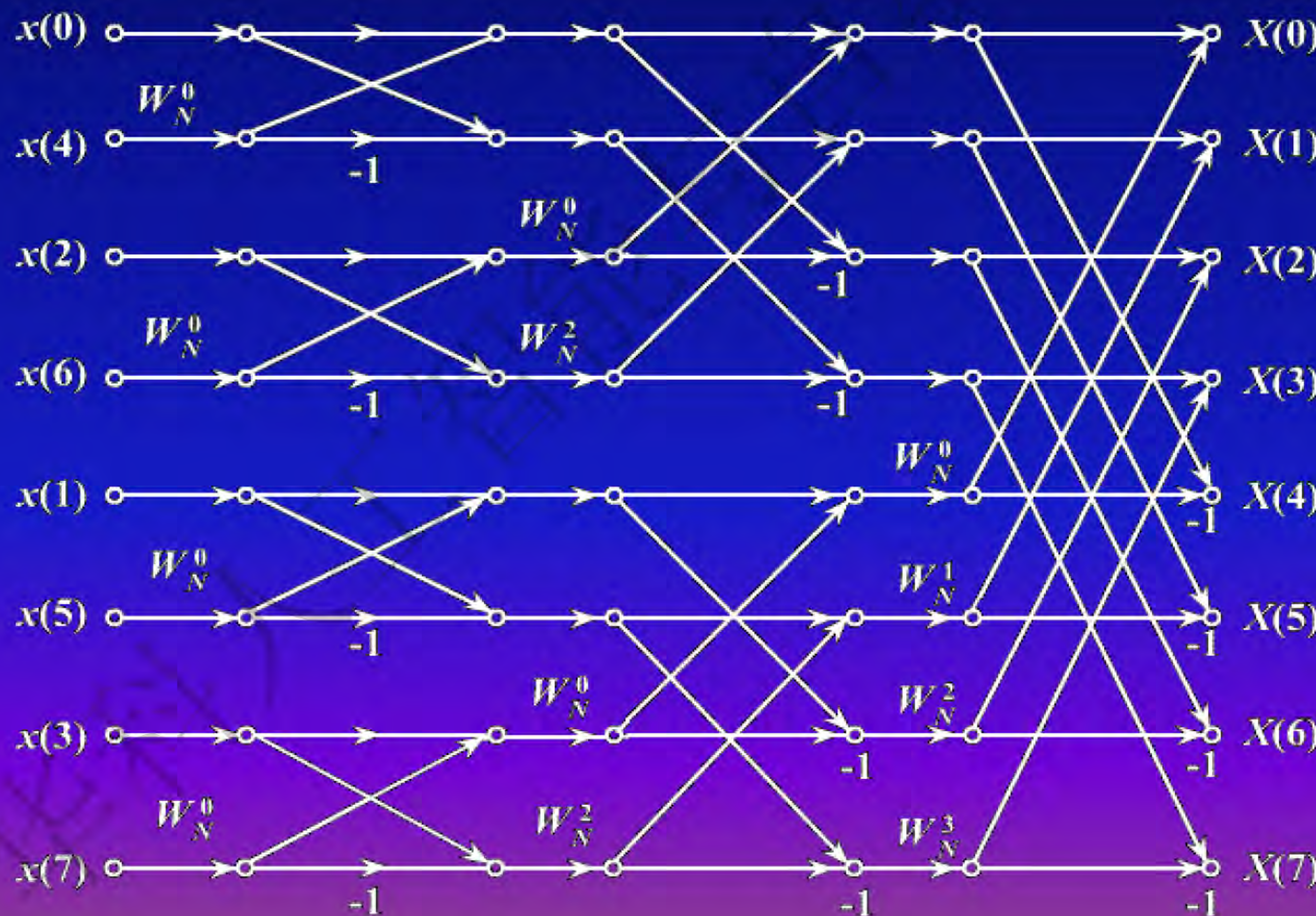
◆计算完一个蝶形后，所得输出数据可立即存入原输入数据所占用的存储单元。这种利用同一个存储单元存储蝶形计算输入、输出数据的方法称为原位计算。

◆采用原位计算，存储数据仅需N个存储单元，下一级的运算仍采用这种原位方式，只是进入蝶形结的组合关系有所不同。——节省存储单元，降低设备成本

W_N^r 的确定

$$\begin{cases} X_m(k) = X_{m-1}(k) + X_{m-1}(k+2^{m-1})W_N^r \\ X_m(k+2^{m-1}) = X_{m-1}(k) - X_{m-1}(k+2^{m-1})W_N^r \end{cases}$$

将上式k表示成L位二进制数，再乘以2^{L-m}，即将L位二进制数左移$L-m$位，把右边空出的位置补零，结果为r的二进制数。$r=(k)_2 \cdot 2^{L-m}$



3.5.5 、IFFT算法

比较：

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

DFT: $$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}$$

FFT: $$W_N^{nk}\rightarrow W_N^{-nk}\quad\times\frac{1}{N}\quad\rightarrow IFFT$$

$$\frac{1}{N}=\left(\frac{1}{2}\right)^L$$

3.7利用FFT计算线性卷积

1、线性卷积的FFT算法

若L点$x(n)$，M点$h(n)$，

则直接计算其线性卷积$y(n)$

$$y(n)=\sum_{m=0}^{M-1}h(m)x(n-m)$$

需运算量乘法次数：$m_d = LM$

需运算量加法次数：$L(M-1)$或$M(L-1)$

FFT法：以圆周卷积代替线性卷积

令 $N = 2^m \geq M + L - 1$

$$x(n) = \begin{cases} x(n) & 0 \leq n \leq L-1 \\ 0 & L \leq n \leq N-1 \end{cases}$$

$$h(n) = \begin{cases} h(n) & 0 \leq n \leq M-1 \\ 0 & M \leq n \leq N-1 \end{cases}$$

则 $y(n) = x(n) * h(n) = x(n) \, \textcircled{N} \, h(n)$

1) $H(k) = FFT\,[h(n)]$ $\qquad N/2 * \log_2 N$

2) $X(k) = FFT\,[x(n)]$ $\qquad N/2 * \log_2 N$

3) $Y(k) = H(k)X(k)$ $\qquad N$

4) $y(n) = IFFT\,[Y(k)]$ $\qquad N/2 * \log_2 N$

$$m_F = N(1 + 3/2 * \log_2 N)$$

比较直接计算和*FFT*法计算的运算量

$$K_m = \frac{m_d}{m_F} = \frac{ML}{2N(1+3/2*\log_2 N)}$$

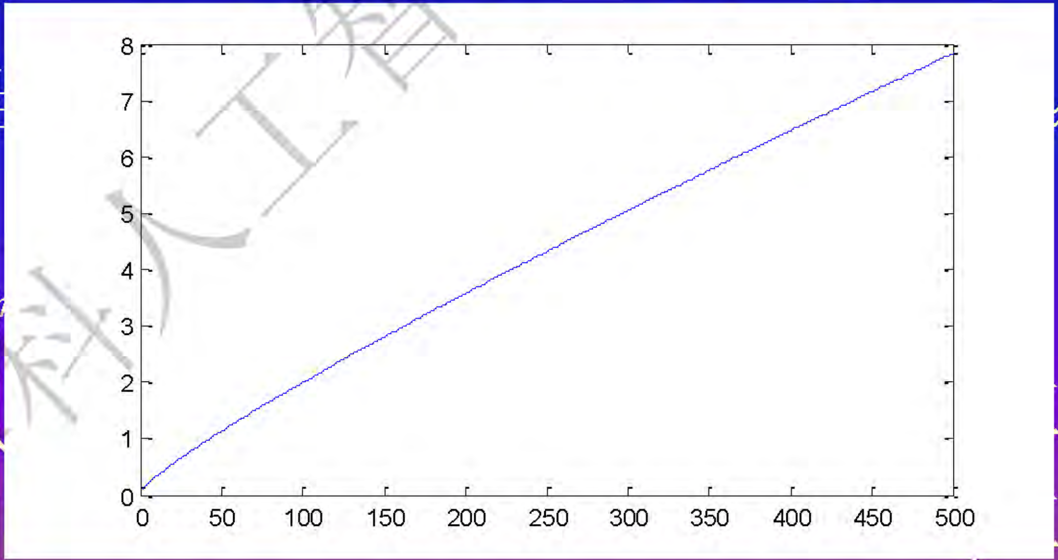
讨论：

1）当 $M \approx L$ 则$N = M + L - 1 \approx 2M$

$$K_m = \frac{M^2}{4M[1+3/2*(1+\log_2 M)]} = \frac{M}{10+6\log_2 M}$$

2）当 $\approx L$

K_m



叠相加法

$L\uparrow$ 叠保留法

1）重叠相加法

对长序列$x(n)$分段，每段L点，

L与$h(n)$的长度M等数量级

$$x_i(n)=\begin{cases} x(n) & iL \le n \le (i+1)L-1 \\ 0 & 其它n \end{cases}$$

则：$x(n)=\sum_{i=-\infty}^{\infty} x_i(n)$

令$N=2^m \ge M+L-1$

$$y(n)=\sum_i y_i(n)$$

$$=\sum_i [x_i(n) * h(n)]$$

$$=\sum_i [x_i(n) \textcircled{N} h(n)]$$

重叠相加法计算步骤

1） $X_i(k) = FFT[x_i(n)]$

2） $H(k) = FFT[h(n)]$

3） $Y_i(k) = X_i(k) \cdot H(k)$

4） $y_i(n) = IFFT[Y_i(k)]$

5） $y(n) = \sum_i y_i(n)$

- 作业：
- P155 3.13,3.16,3.18