第四章 快速傅里叶变换

FFT: Fast Fourier Transform

1965年，Cooley, Tukey

《计算数学》

“机器计算傅里叶级数的一种算法”

学 习 目 标

- 掌握按时间抽选的基-2FFT算法的算法原理、运算流图、所需计算量和算法特点
- 掌握按频率抽选的基-2FFT算法的算法原理、运算流图、所需计算量和算法特点
- 理解IFFT算法
- 了解混合基、分裂基和基-4FFT算法

一、直接计算DFT的问题及改进途径

N点有限长序列$x(n)$

$DFT:$

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

$IDFT:$

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

运算量

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

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

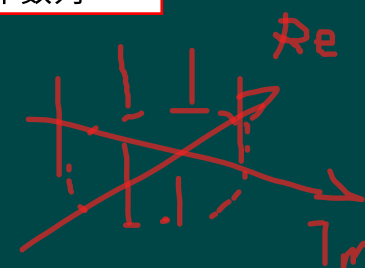
这里其实是使用了周期性，将n变为-n之后函数的周期性没有发生变化，最第一个下箭头处是将n变为了n-N，函数不变，在第二个下箭头处是将n变为了n-nN（可行的原因是n也是整数）

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

$$\underbrace{W_N^{(N-n)k}}_{W_N^{Nk}\cdot W_N^{-nk}} \qquad \underbrace{W_N^{n(N-k)}}_{W_N^{nN}\cdot W_N^{-nk}}$$

周期性 $\quad W_N^{nk} = W_N^{(N+n)k} = W_N^{n(N+k)}$

更形象一点来说，这个玩意就是在复平面中的单位圆上的N个数列



可约性 $\quad W_N^{nk} = W_{mN}^{mnk} \qquad W_N^{nk} = W_{N/m}^{nk/m}$

$$W_{mN}^{mnk} = e^{-j\frac{2\pi}{mN}mnk}$$

$$e^{-j\frac{2\pi}{N}\cdot\frac{N}{2}} = e^{-j\pi} = -1$$

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

*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

二、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:

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}=\sum_{n=0}^{N-1}x_1(n)W_N^{nk}+\sum_{n=0}^{N-1}x_2(n)W_N^{nk}$$

n为偶数　　　n为奇数

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

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

$$\left(W_{N\cdot\frac{1}{2}}^{2\cdot\frac{1}{2}}\right)^{rk}$$

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

$$=\underset{e}{X_1}(k)+W_N^k\underset{o}{X_2}(k)\qquad r,k=\underline{0,1,\ldots N/2-1}$$

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

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

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

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

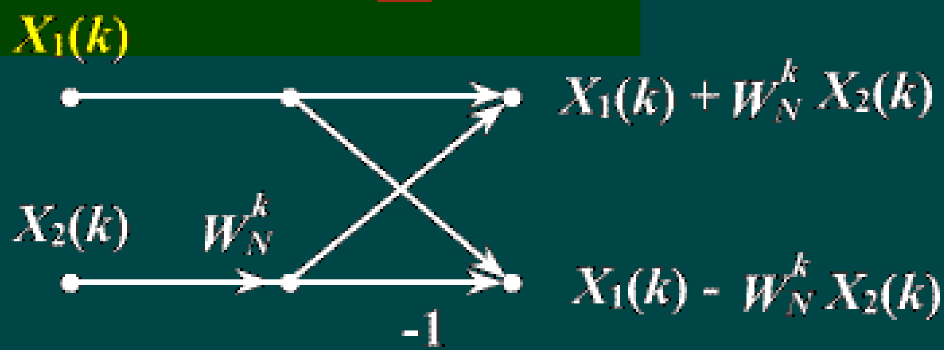


图4-1 时间抽选法蝶形运算流图符号

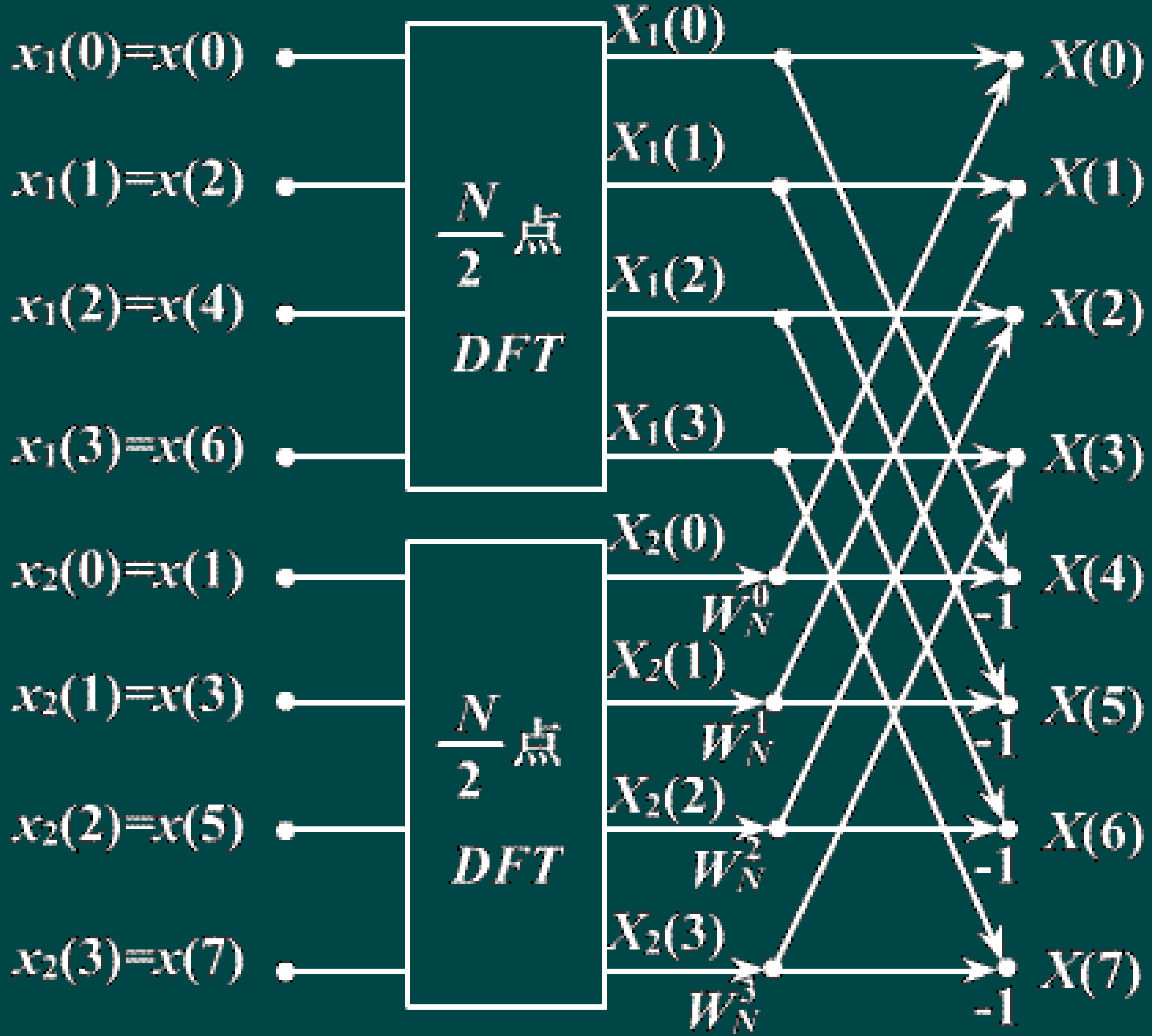


图4-2 按时间抽选，将一个N点DFT分解为两个$N/2$点DFT

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	$N^2/2+N/2$ $\approx N^2/2$	$N\left(N/2-1\right)+N$ $\approx N^2/2$

运算量减少了近一半

$N/2$仍为偶数，进一步分解：$N/2 \longrightarrow N/4$

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

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

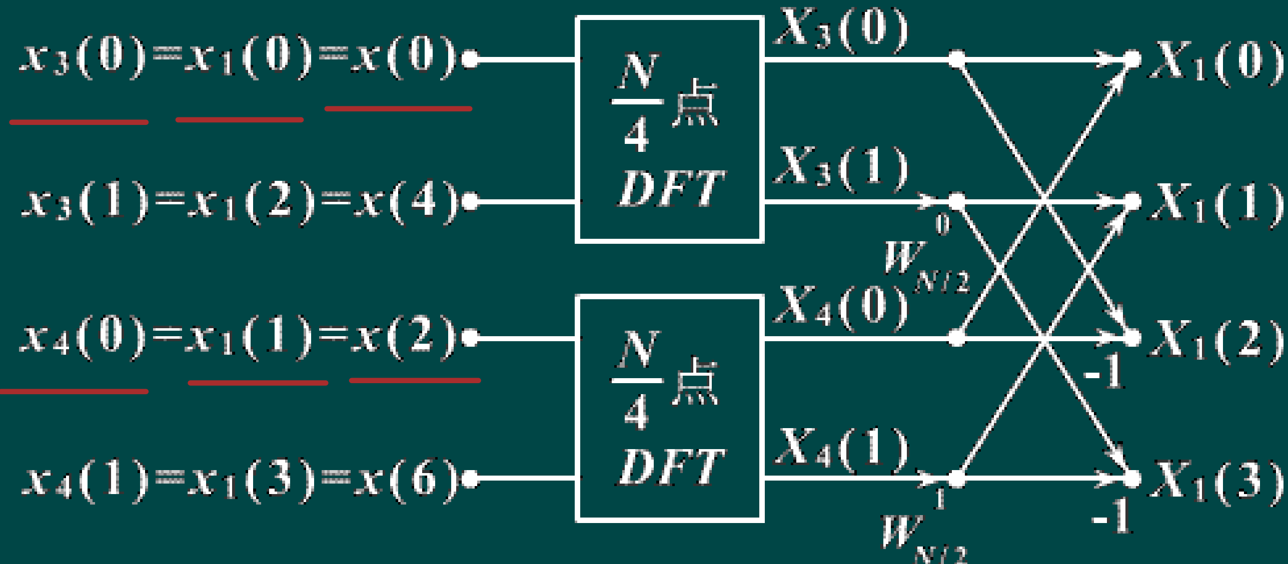


图4-3 由两个$N/4$点DFT组合成一个$N/2$点DFT

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：

$$X_5(k) = DFT[x_5(l)] = DFT[x_2(2l)] \quad l = 0,1,\ldots,N/4-1$$

$$X_6(k) = DFT[x_6(l)] = DFT[x_2(2l+1)]$$

统一系数：$W_{N/2}^{k} \to W_{N}^{2k}$

低一级时域序列分奇偶做DFT得到的X(K)可通过蝶形算法表示为将低一级序列组合成的完整序列的DFT，以此迭代X(K)，最后表示为原x(n)的X(K)

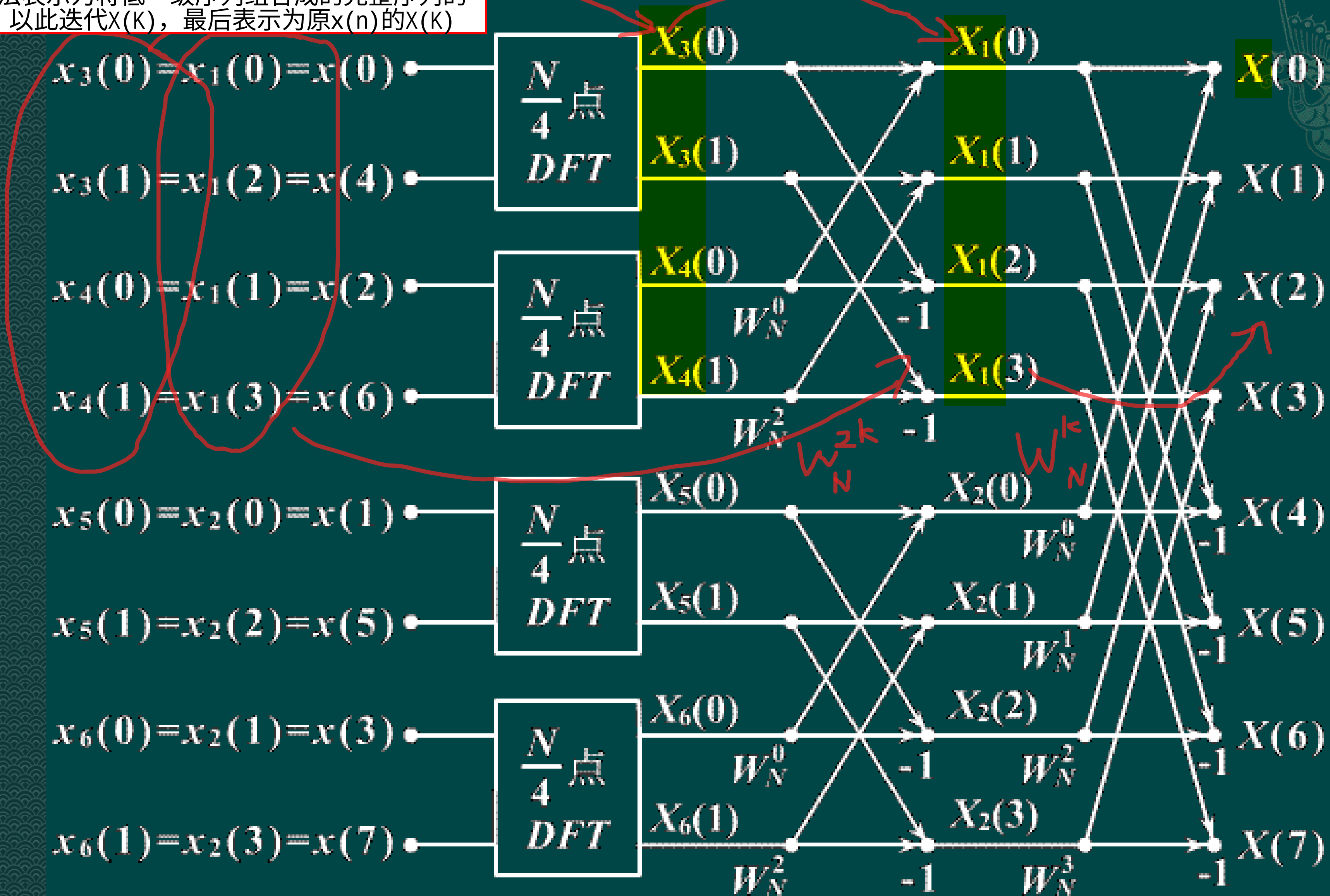


图4-4 按时间抽选，将一个N点DFT分解为四个$N/4$点DFT(N=8)

这样逐级分解，直到2点DFT

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

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

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

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

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

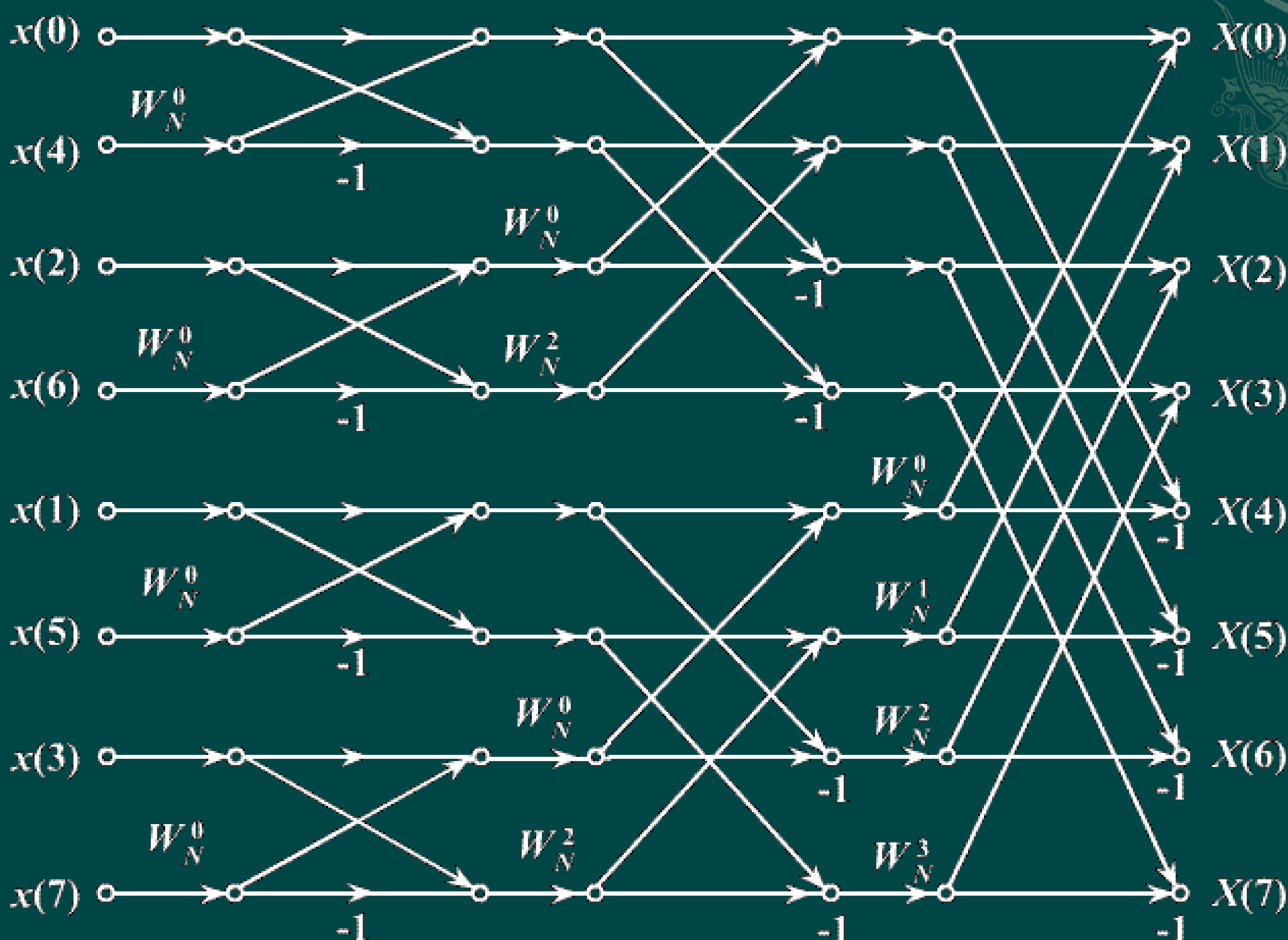


图4-5 N=8按时间抽选法FFT运算流图

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：$m_F = \dfrac{N}{2} L = \dfrac{N}{2} \log_2 N$

复数加法：$a_F = NL = N \log_2 N$

比较DFT

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

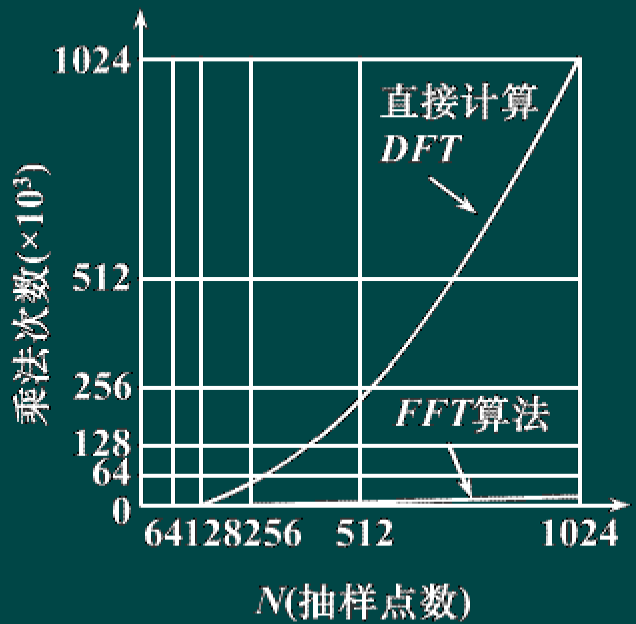


图4-6　直接计算DFT与FFT算法所需乘法次数的比较

3、算法特点

1）原位计算：蝶形结运算完成后，输出的两节点值放到原输入两节点的存储器中

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

m表示第m级迭代，k，j表示数据所在的行数

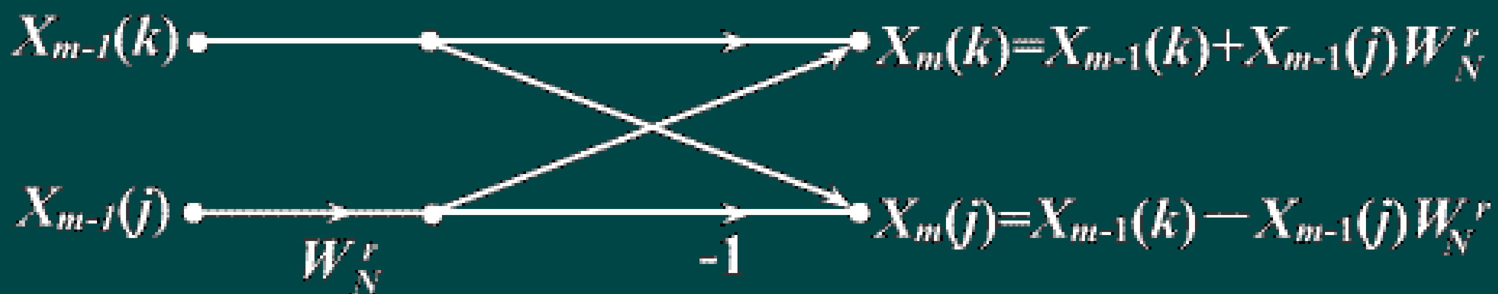


图 4-7 按时间抽选蝶形运算结构

2）输入倒位序 $x(n) \quad n=(n_2 n_1 n_0)_2$

输出自然序

n_0	n_1	n_2
0	0	0
		1
	1	0
		1
1	0	0
		1
	1	0
		1

倒位序 $\hat{n}=(n_0 n_1 n_2)_2$		自然序 $n=(n_2 n_1 n_0)_2$	
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

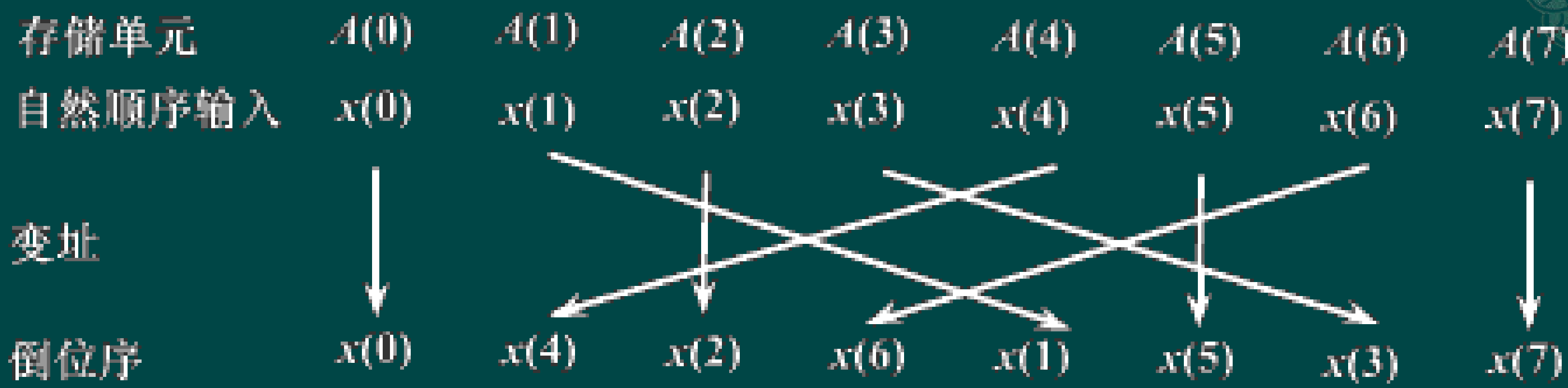


图4-9　倒位序的变址处理

3）蝶形运算

对$N=2^L$点FFT，输入倒位序，输出自然序，

第m级运算每个蝶形的两节点距离为 2^{m-1}

第m级运算：

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

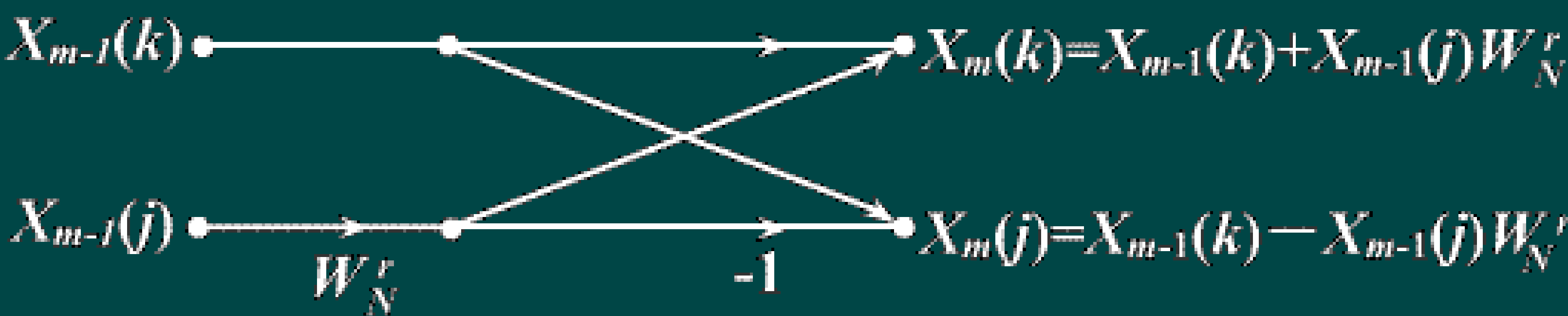


图 4-7 按时间抽选蝶形运算结构

W_N^r 的确定

蝶形运算两节点的第一个节点为k值，表示成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。

$$r=(k)_2\cdot 2^{L-m}$$

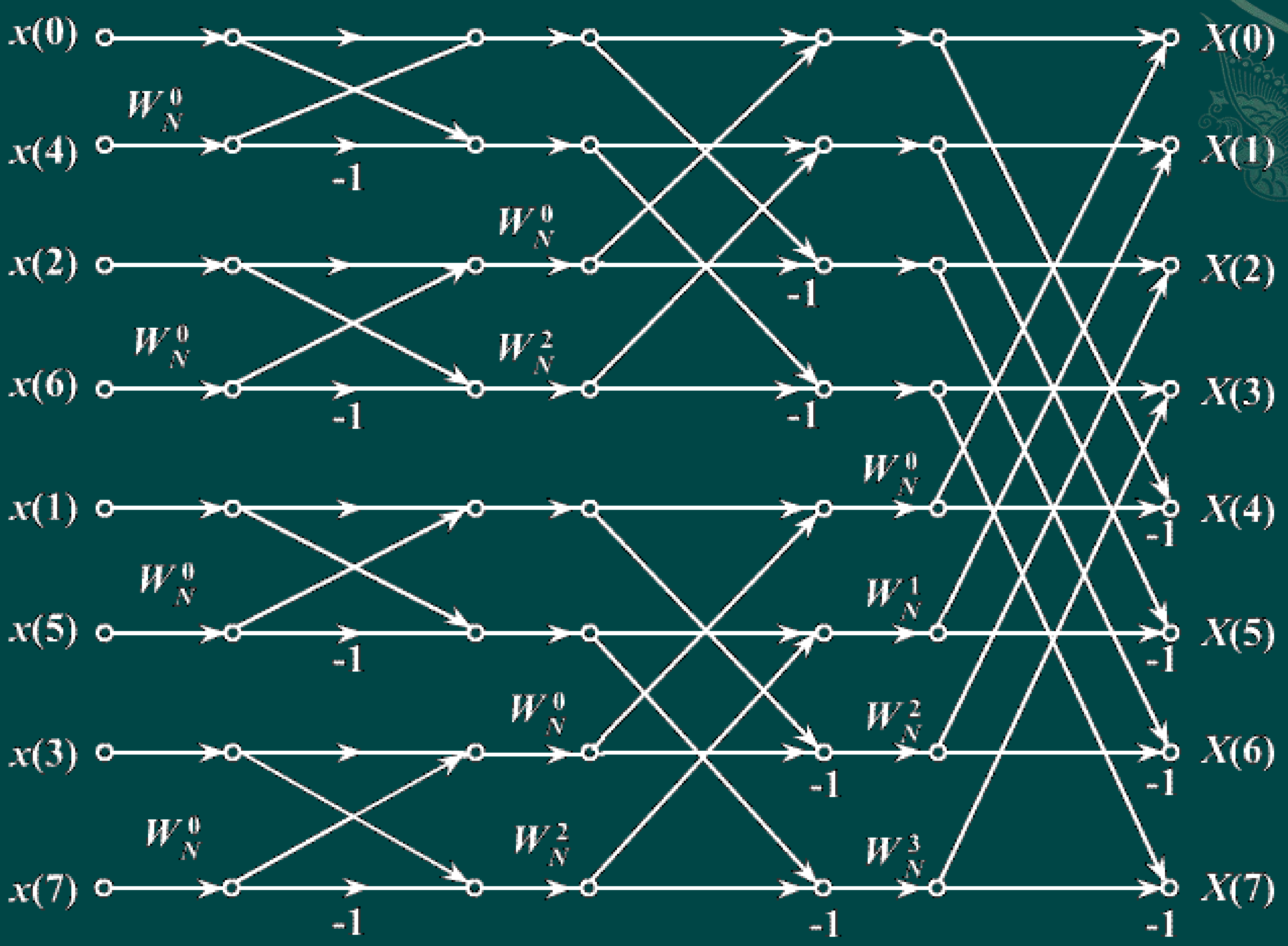


图4-5　N=8按时间抽选法*FFT*运算流图

4）存储单元

输入序列$x(n)$ ：N个存储单元

系数W_N^r：$N/2$个存储单元

4、DIT算法的其他形式流图

- 输入倒位序输出自然序
- 输入自然序输出倒位序
- 输入输出均自然序
- 相同几何形状
  - 输入倒位序输出自然序
  - 输入自然序输出倒位序

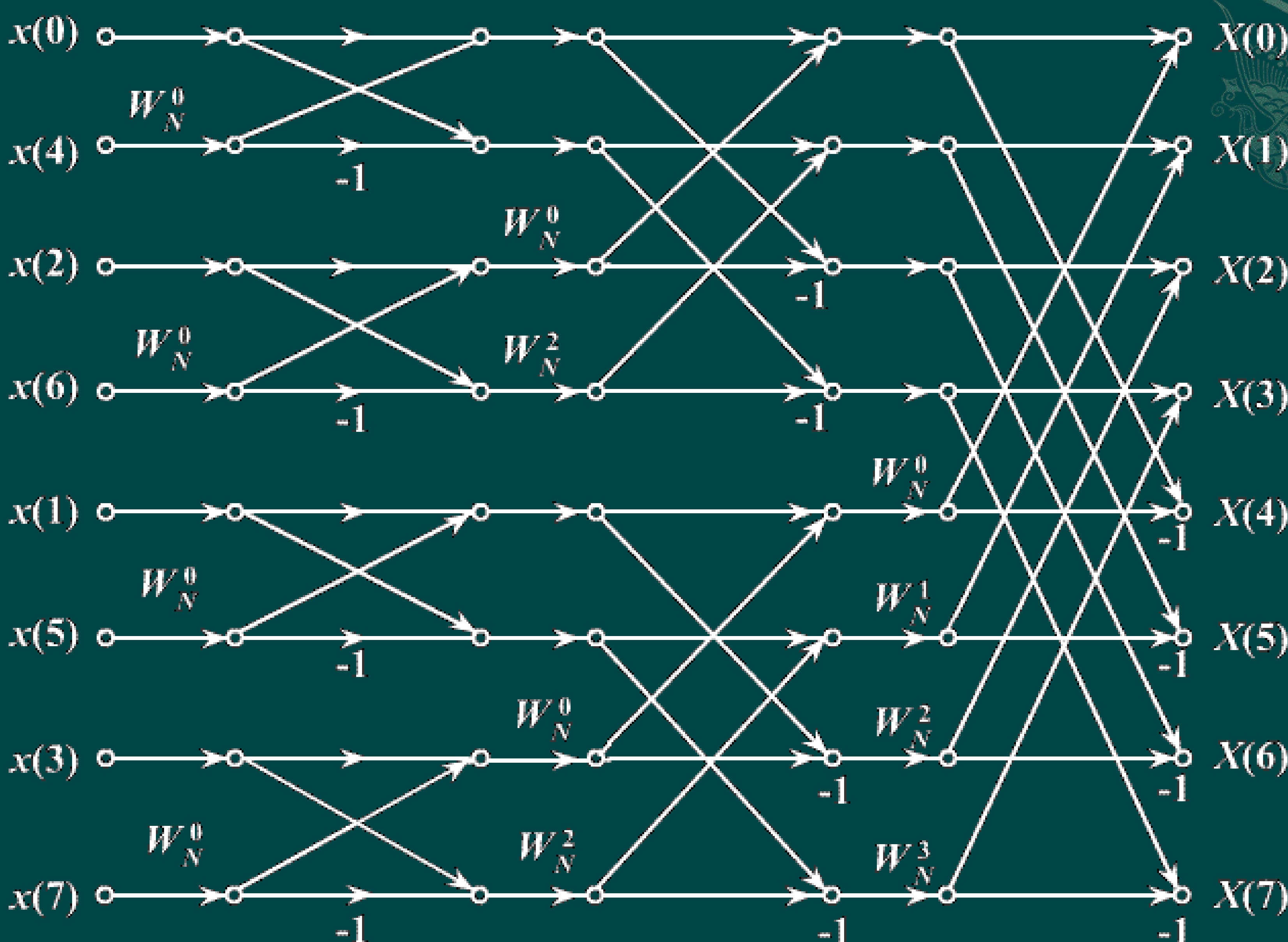


图4-5　N=8按时间抽选法FFT运算流图

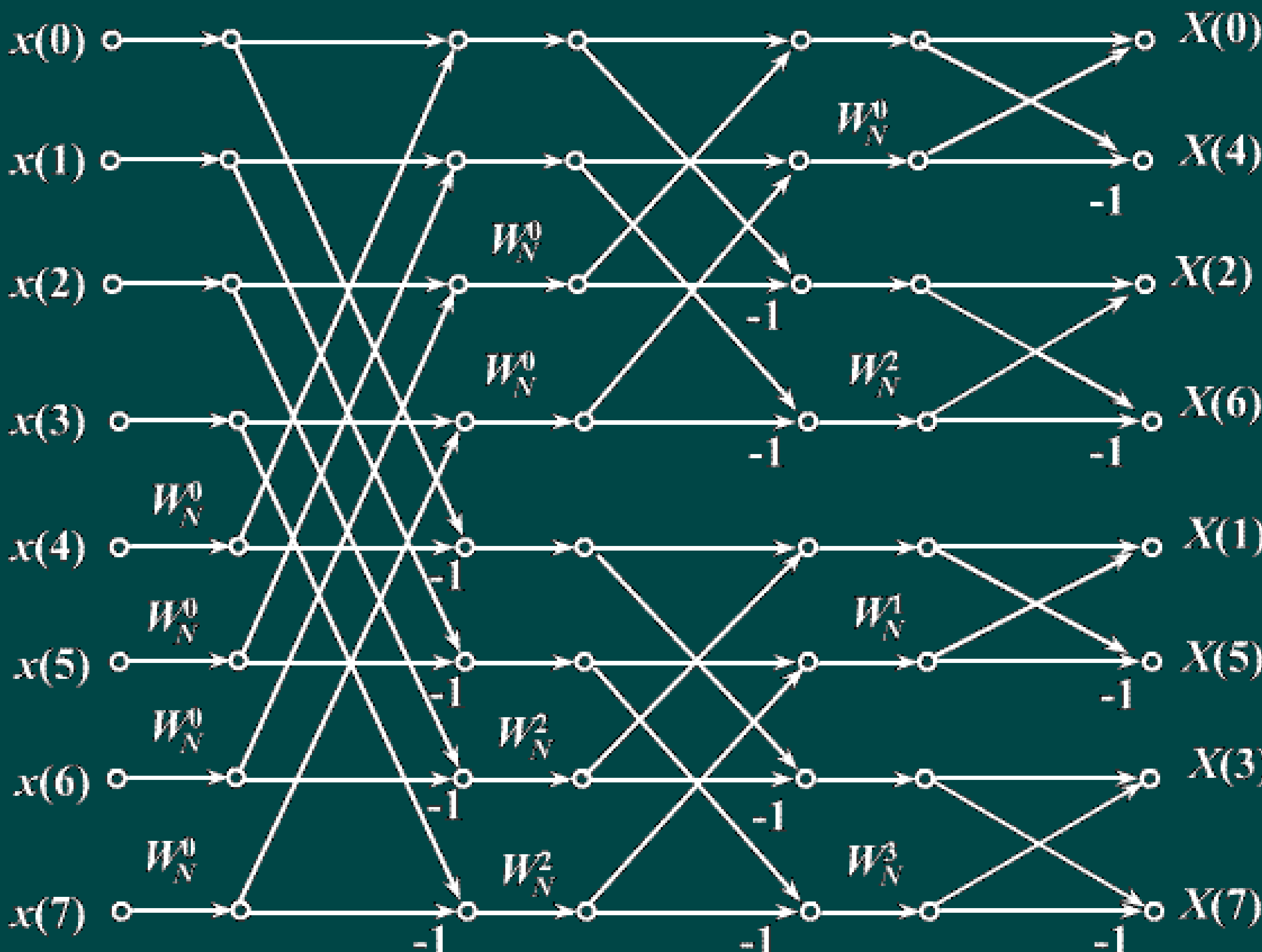


图4-10 按时间抽选，输入自然顺序，输出倒位序的*FFT*图

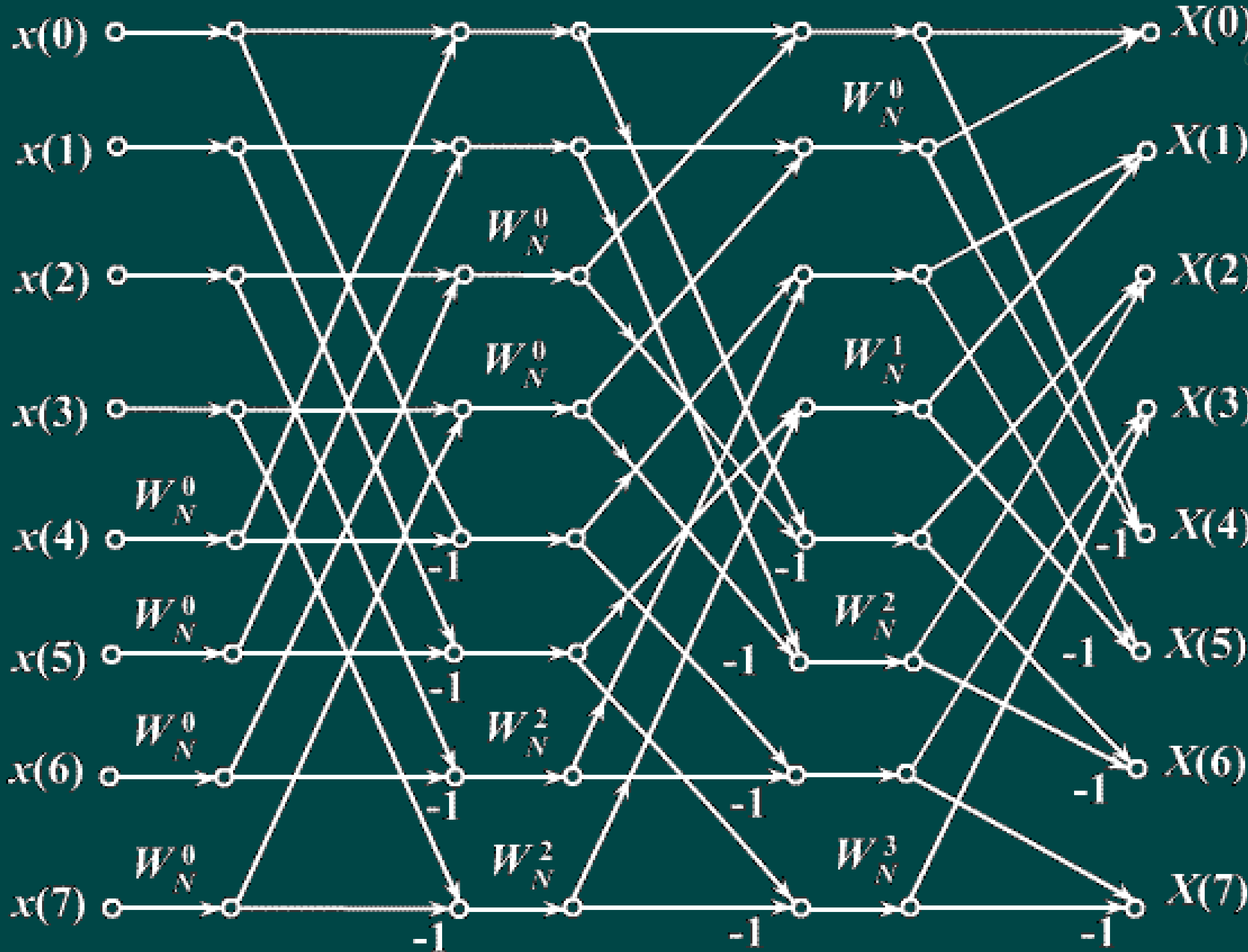


图4-11 按时间抽选,输入输出皆为自然顺序的*FFT*流图

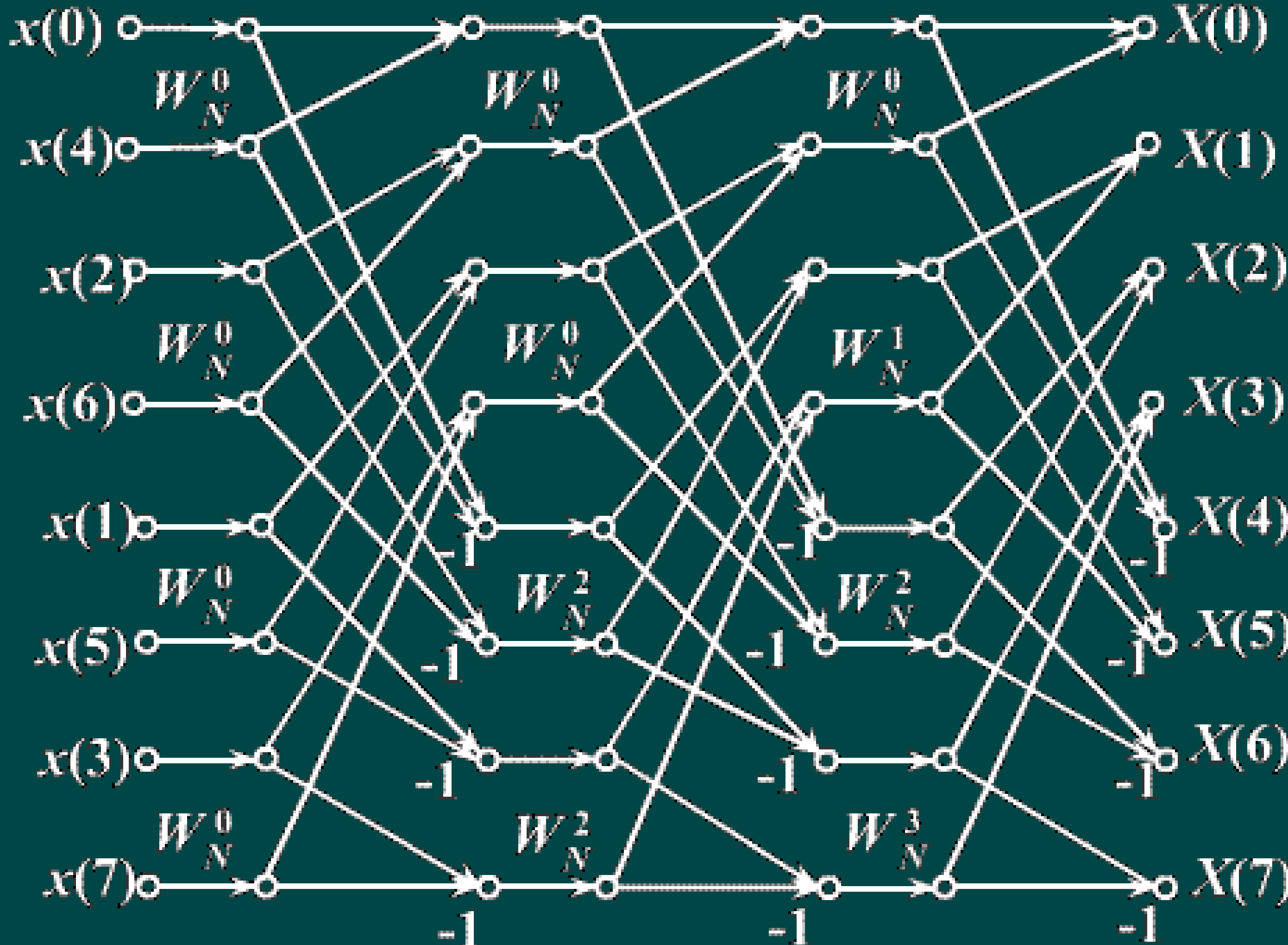


图 4-12 按时间抽选，各级具有相同几何形状，输入倒位序、输出自然顺序的*FFT*流图

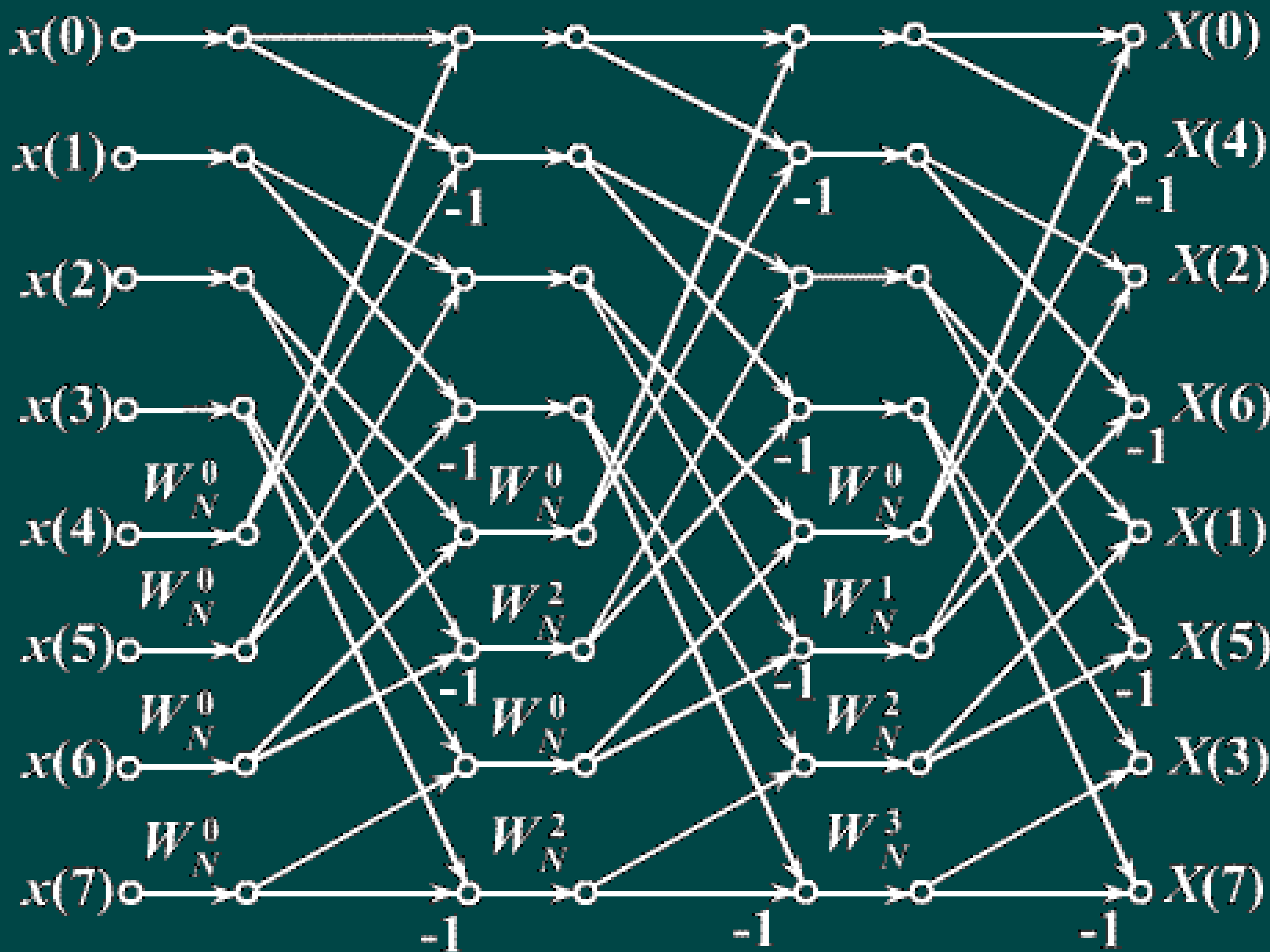


图 4-13 按时间抽选，各级具有相同几何形状，输入自然顺序、输出倒位序的*FFT*流图

三、按频率抽选的基-2FFT算法

1、算法原理

设序列点数$N=2^L$，L 为整数。把输出序列 X(k) 按k的奇偶分解为越来越短的序列。

将$X(k)$ 按k的奇偶分组前，先将输入$x(n)$ 按n 的顺序分成前后两半:

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

$$=\sum_{n=0}^{N/2-1}x(n)W_N^{nk}+\sum_{n=0}^{N/2-1}x\left(n+\frac{N}{2}\right)W_N^{\left(n+\frac{N}{2}\right)k}$$

$$=\sum_{n=0}^{N/2-1}\left[x(n)+x\left(n+\frac{N}{2}\right)W_N^{Nk/2}\right]W_N^{nk}$$

$$W_N^{N/2}=-1$$

$$=\sum_{n=0}^{N/2-1}\left[x(n)+(-1)^k x\left(n+\frac{N}{2}\right)\right]W_N^{nk}$$

$$k=0,1,\ldots,N-1$$

按k的奇偶将X(k)分成两部分：

$$\begin{cases} k = 2r \\ k = 2r+1 \end{cases} \qquad r = 0,1,\ldots,N/2-1$$

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

$$= \sum_{n=0}^{N/2-1}\left[x(n) + x\left(n+\frac{N}{2}\right)\right]W_{N/2}^{nr}$$

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

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

令
$$\begin{cases} x_1(n) = x(n) + x\left(n + \dfrac{N}{2}\right) \\ x_2(n) = \left[x(n) - x\left(n + \dfrac{N}{2}\right)\right]W_N^n \end{cases}$$

$$n = 0,1,\ldots,\frac{N}{2}-1$$

则 $X(2r)$ 和 $X(2r+1)$ 分别是 $x_1(n)$ 和 $x_2(n)$ 的 $N/2$ 点 DFT，记为 $X_1(k)$ 和 $X_2(k)$

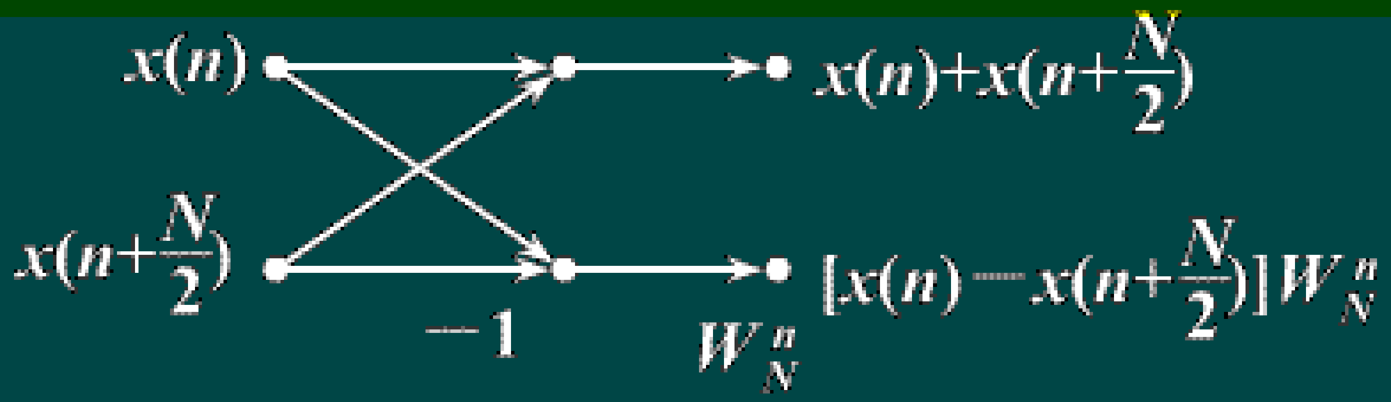
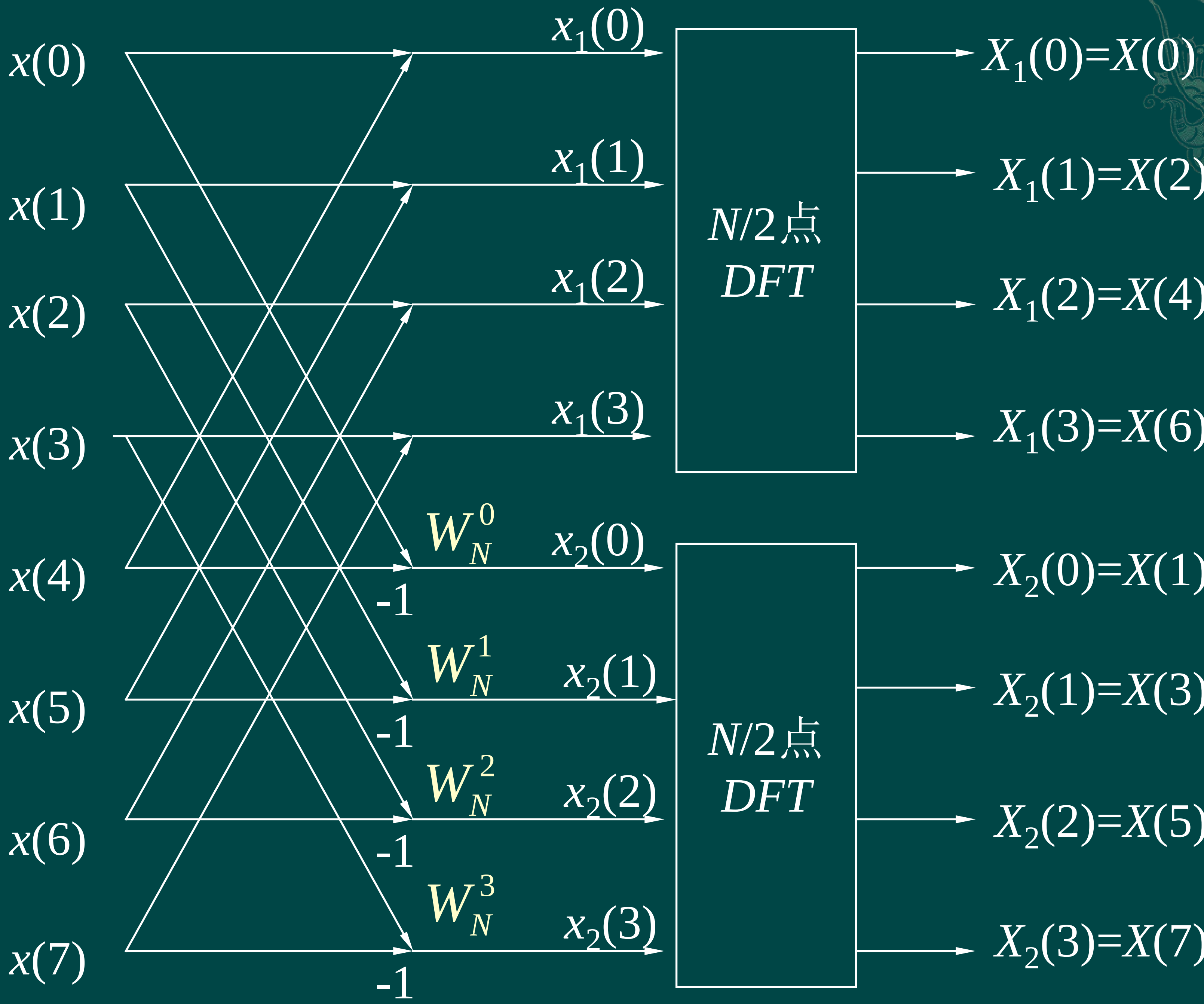


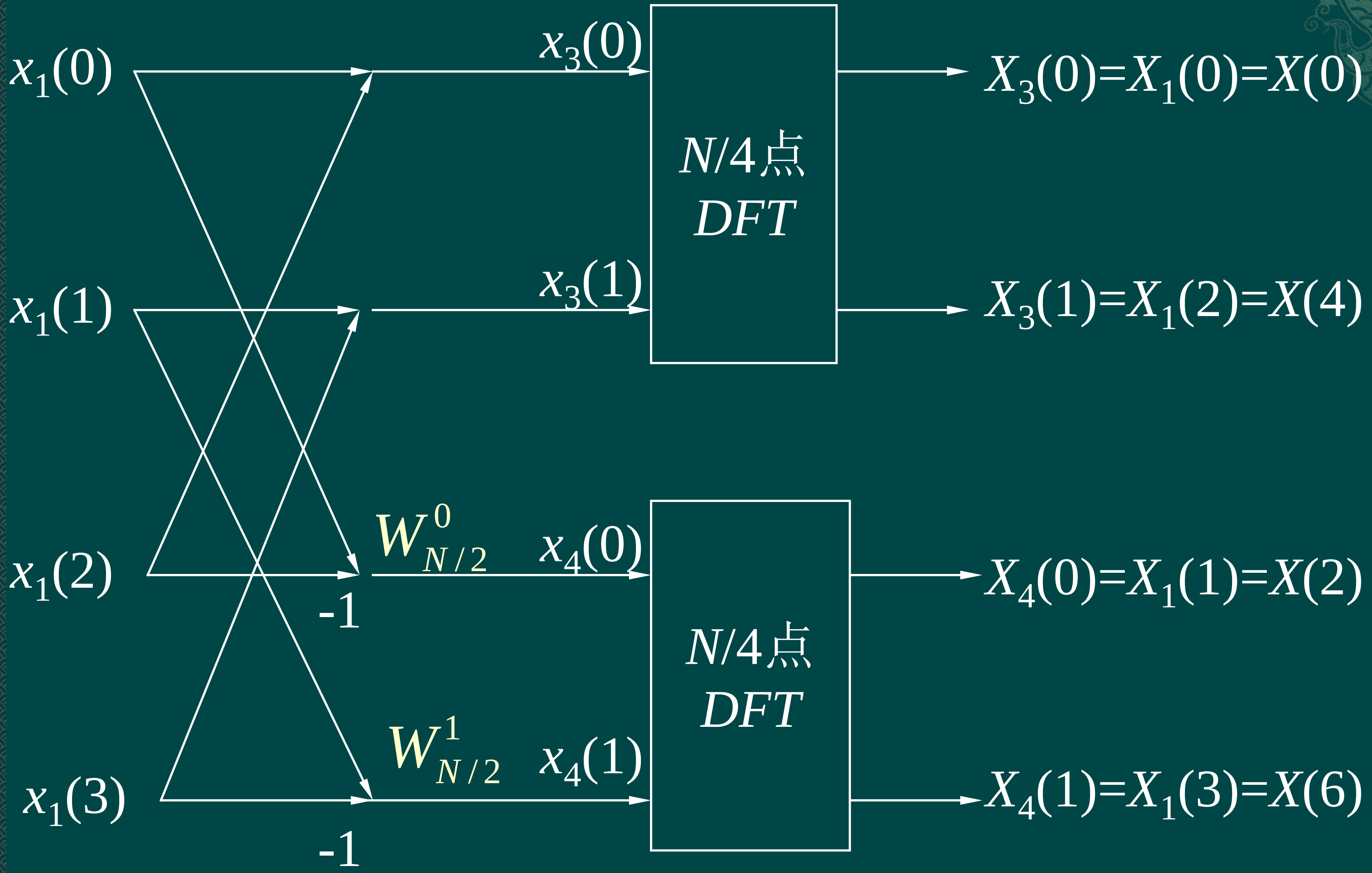
图4-14 按频率抽选蝶形运算流图符号


x(0)
x(1)
x(2)
x(3)
x(4)
x(5)
x(6)
x(7)
x1(0)
x1(1)
x1(2)
x1(3)
x2(0)
x2(1)
x2(2)
x2(3)
W_N^0
W_N^1
W_N^2
W_N^3
-1
-1
-1
-1
N/2点
DFT
N/2点
DFT
X1(0)=X(0)
X1(1)=X(2)
X1(2)=X(4)
X1(3)=X(6)
X2(0)=X(1)
X2(1)=X(3)
X2(2)=X(5)
X2(3)=X(7)

$N/2$仍为偶数，进一步分解：$N/2 \rightarrow N/4$

$$\begin{cases} x_3(n) = x_1(n) + x_1(n+N/4) \\ x_4(n) = [x_1(n) - x_1(n+N/4)]W_{N/2}^{n} \end{cases} \quad n = 0,1,\ldots,\frac{N}{4}-1$$

$$\begin{cases} X_3(k) = X_1(2k) = DFT[x_3(n)] \\ X_4(k) = X_1(2k+1) = DFT[x_4(n)] \end{cases} \quad k = 0,1,\ldots,\frac{N}{4}-1$$



$x_1(0)$
$x_1(1)$
$x_1(2)$
$x_1(3)$
$x_3(0)$
$x_3(1)$
$W_{N/2}^0$
$x_4(0)$
-1
$W_{N/2}^1$
$x_4(1)$
-1
$N/4$点
DFT
$N/4$点
DFT
$X_3(0)=X_1(0)=X(0)$
$X_3(1)=X_1(2)=X(4)$
$X_4(0)=X_1(1)=X(2)$
$X_4(1)=X_1(3)=X(6)$

同理:

$$\begin{cases} X_5(k) = X_2(2k) = DFT[x_5(n)] \\ X_6(k) = X_2(2k+1) = DFT[x_6(n)] \end{cases} \qquad k = 0,1,\ldots,\frac{N}{4}-1$$

其中:

$$\begin{cases} x_5(n) = x_2(n) + x_2(n+N/4) \\ x_6(n) = [x_2(n) - x_2(n+N/4)]W_{N/2}^{n} \end{cases} \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

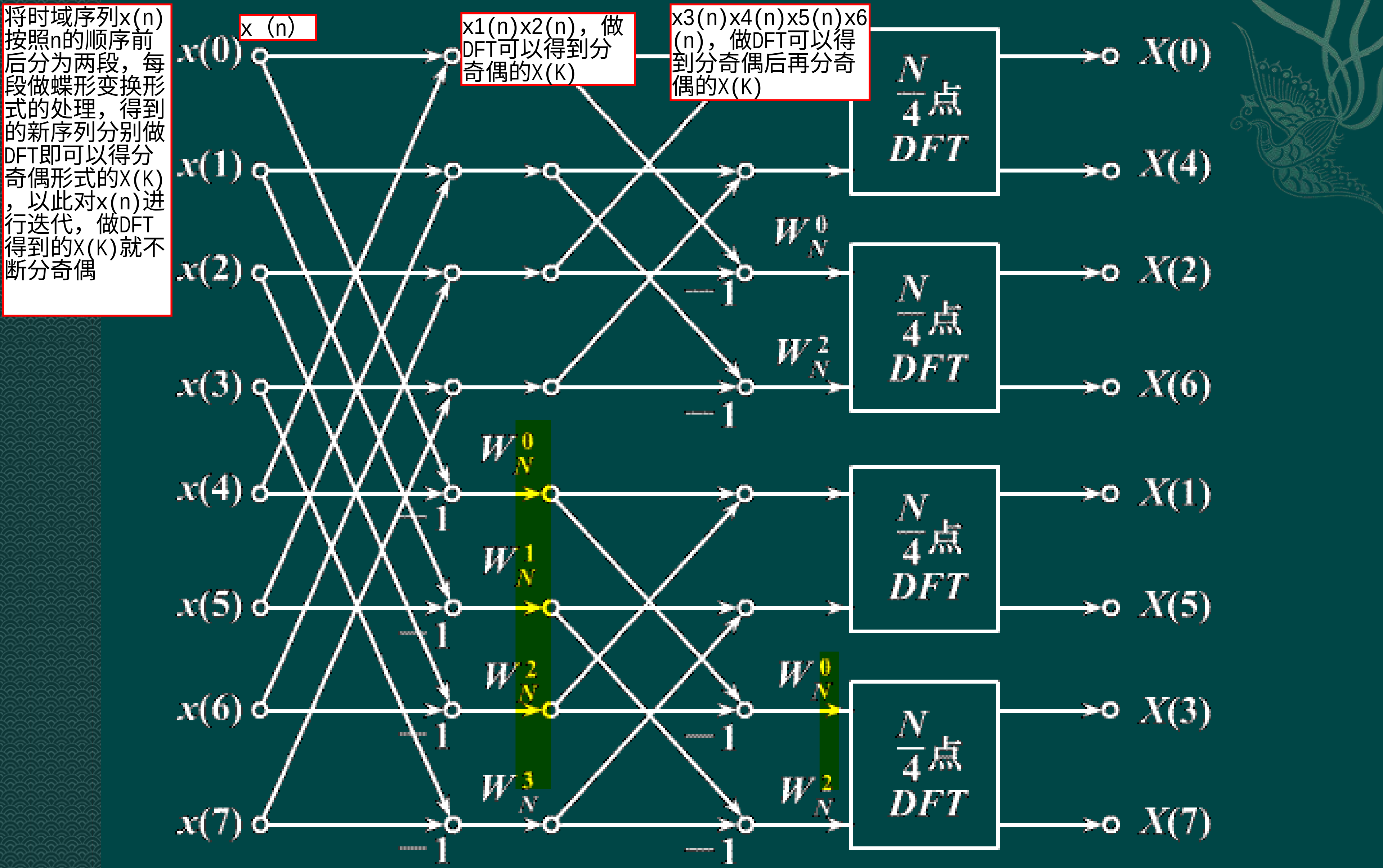


图4-16 按频率抽选，将一个N点DFT分解为4个$N/4$点$DFT(N=8)$

逐级分解，直到2点DFT

当N=8时，即分解到$x_3(n)$，$x_4(n)$，$x_5(n)$，$x_6(n)$，n=0, 1

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

$$\begin{cases} X(0)=X_3(0)=x_3(0)W_2^0+W_2^0x_3(1)=x_3(0)+x_3(1) \\ X(4)=X_3(1)=x_3(0)W_2^0+W_2^1x_3(1)=[x_3(0)-x_3(1)]W_N^0 \end{cases}$$

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

$$\begin{cases} X(2)=X_4(0)=x_4(0)W_2^0+W_2^0x_4(1)=x_4(0)+x_4(1) \\ X(6)=X_4(1)=x_4(0)W_2^0+W_2^1x_4(1)=[x_4(0)-x_4(1)]W_N^0 \end{cases}$$

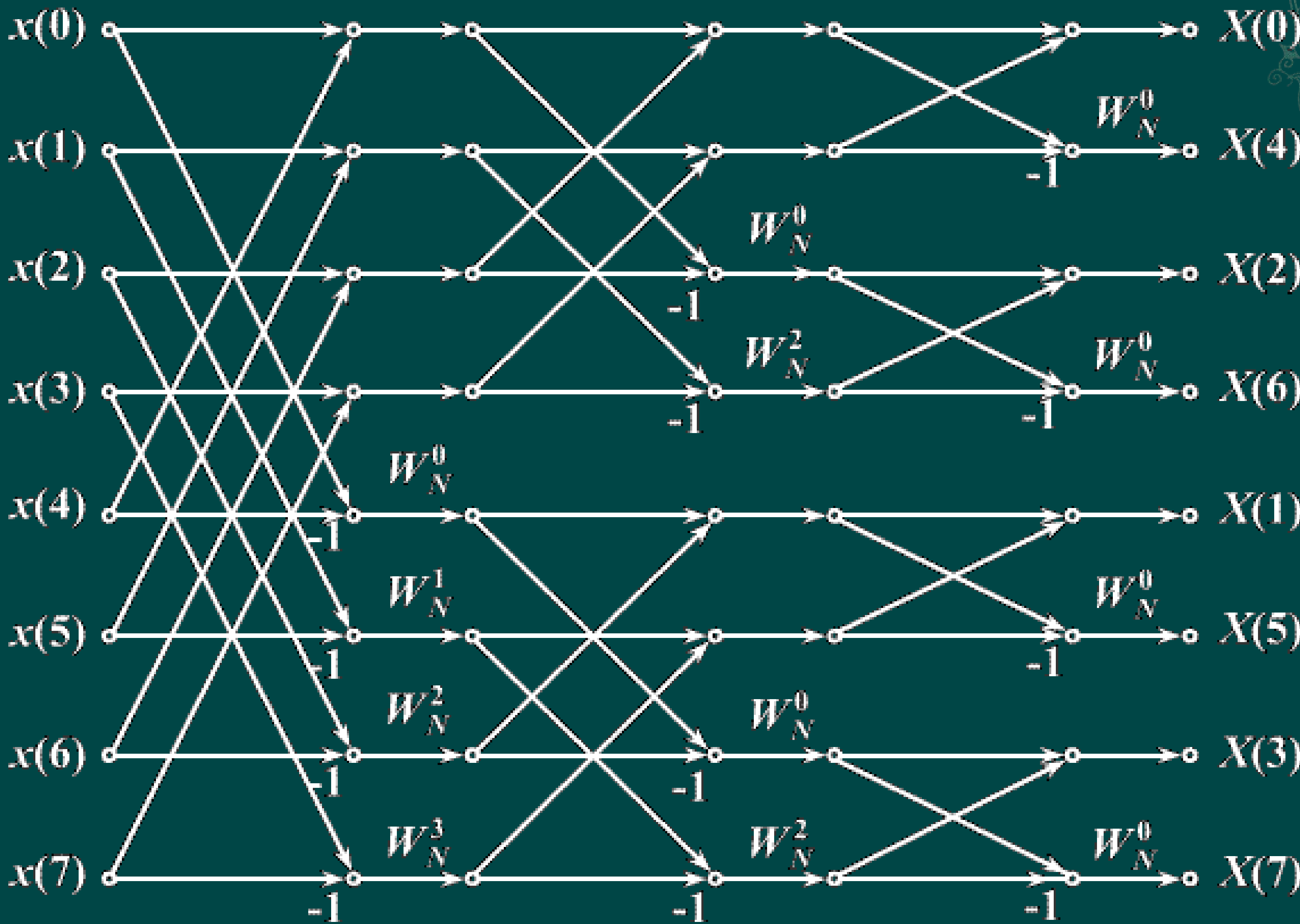


图4-17 按频率抽选*FFT*流图(*N*=8)

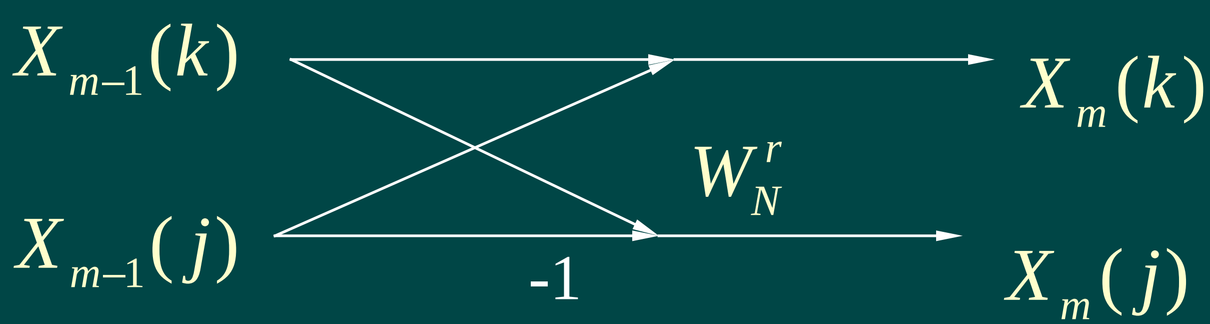
2、算法特点

1）原位计算

L级蝶形运算，每级$N/2$个蝶形，每个蝶形结构：

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

m表示第m级迭代，k，j表示数据所在的行数



2）蝶形运算

对$N=2^L$点FFT，输入自然序，输出倒位序，

两节点距离：$2^{L-m}=N/2^m$

第m级运算:

$$\begin{cases} X_m(k)=X_{m-1}(k)+X_{m-1}\left(k+\dfrac{N}{2^m}\right) \\ X_m\left(k+\dfrac{N}{2^m}\right)=\left[X_{m-1}(k)-X_{m-1}\left(k+\dfrac{N}{2^m}\right)\right]W_N^r \end{cases}$$

W_N^r 的确定

蝶形运算两节点的第一个节点为k值，表示成L位二进制数，左移m-1位，把右边空出的位置补零，结果为r的二进制数。

$$r=(k)_2\cdot 2^{m-1}$$

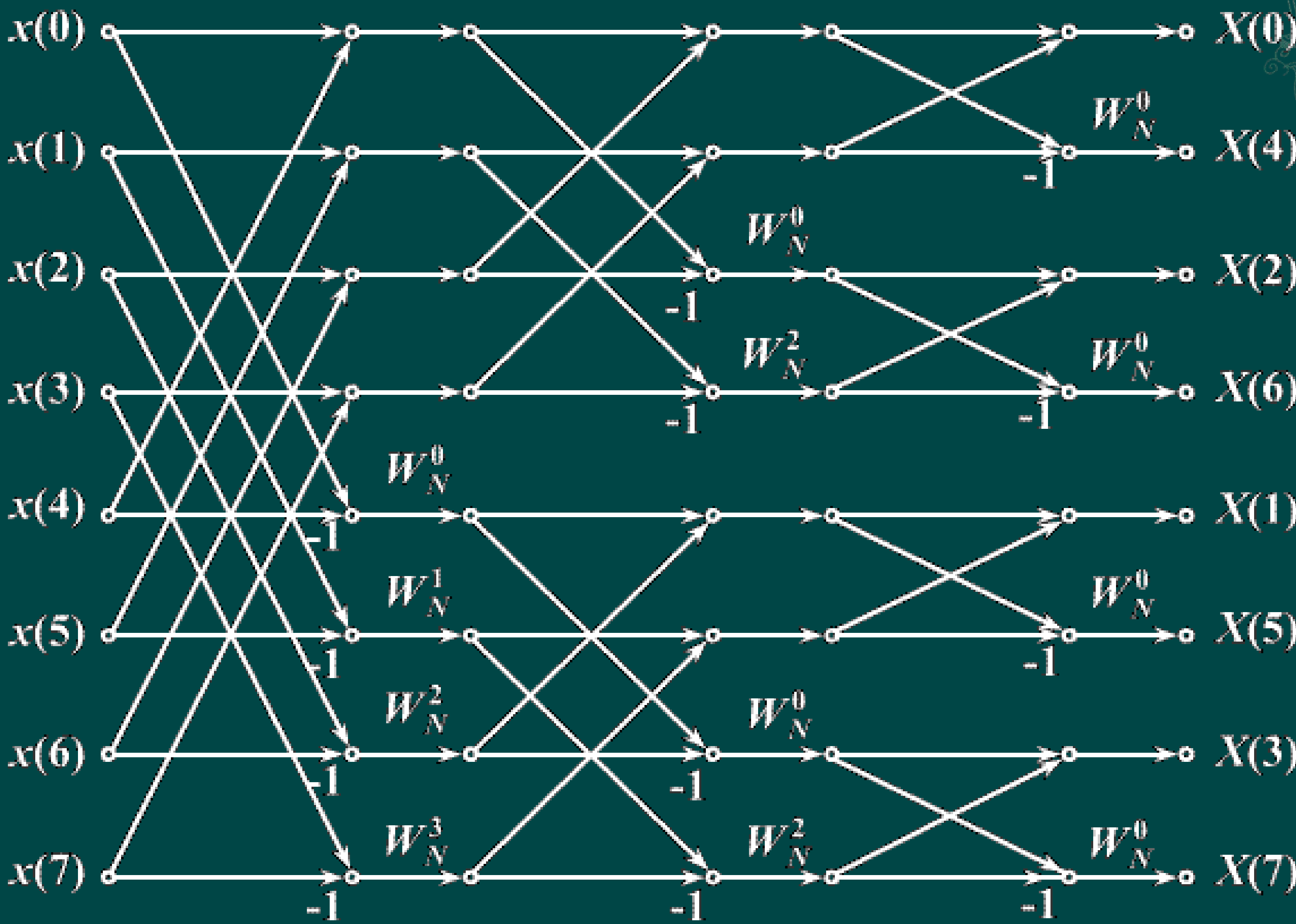


图4-17 按频率抽选*FFT*流图(N=8)

3、DIT与DIF的异同

- 基本蝶形不同
  - *DIT*：先复乘后加减
  - *DIF*：先减后复乘

- 运算量相同

$$m_F = \frac{N}{2}\log_2 N \qquad a_F = N\log_2 N$$

- 都可原位运算
- *DIT*和*DIF*的基本蝶形互为转置

四、IFFT算法

比较：

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

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

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

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

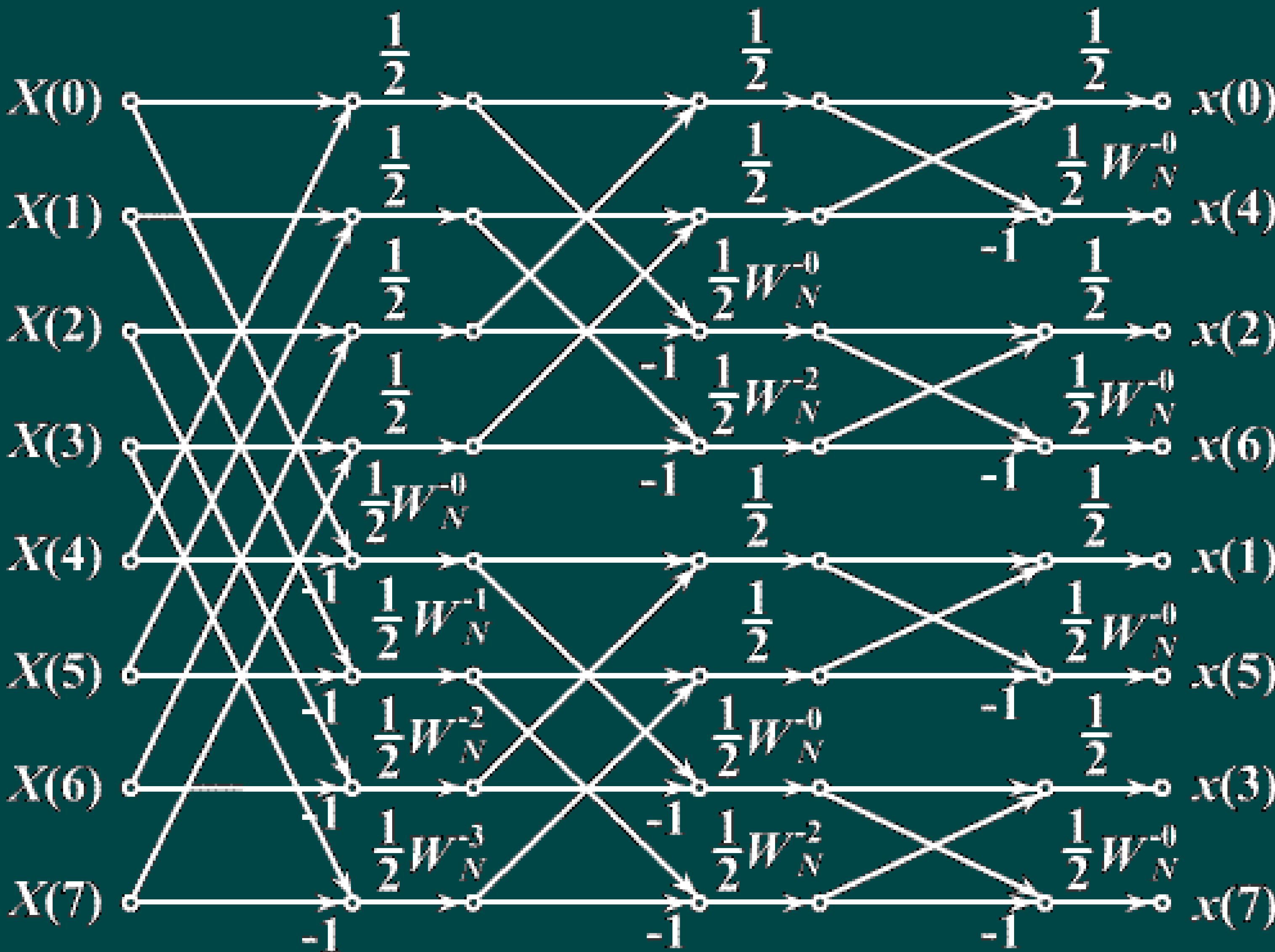


图4-19 *IFFT*流图(*N*=8)

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

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

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

直接调用FFT子程序计算IFFT的方法：

$$\xrightarrow{X(k)}\boxed{\text{共轭}}\xrightarrow{X^*(k)}\boxed{\text{FFT}}\longrightarrow\boxed{\text{共轭}}\longrightarrow\boxed{\text{乘}1/N}\xrightarrow{x(n)}$$

作业：4.1 4.2 4.6

基于4.6的补充内容：

① 直接利用 DFT 计算：复乘次数为 N^2，复加次数为 $N(N-1)$。

② 利用 FFT 计算：复乘次数为 $\frac{N}{2}\log_2 N$，复加次数为 $N\log_2 N$。

4.6 参照图 4.17 的基-2 DIF-FFT 流图，若 N 为大于 8 的整数，满足 $N=2^P$（P 为正整数）。假设数组的编号为 $0\sim\log_2 N$，数组中的数据被存在 $0\sim N-1$ 的复数寄存器中，最左边的数据（输入数据）是第 0 列，第一级蝶形结的输出是第 1 列，以此类推。下列问题均与第 m 列的计算有关，$1\leqslant m\leqslant\log_2 N$，答案用 m 和 N 表示。

(1) 每个蝶形结有多少复数乘法与复数加法运算？整个流程图要计算多少个蝶形结？总共需要多少次复数乘法和复数加法运算？

(2) 由第 $m-1$ 列到第 m 列，包含的 W_N 的幂是多少？

(3) 蝶形结的两个复数输入点的地址的间距是多少？

(4) 利用幂次相同系数的各蝶形结的数据地址间距是多少？

注意：此流图中，蝶形结的相乘系数是在蝶形结输出端的。

由 DIF-FFT 流图可知，每一级蝶形运算包括相同数量的蝶形结，不同之处在于蝶形结中两个复数输入点的地址间距不同，第 1 级的地址间距为$\frac{N}{2}$，最后 1 级的地址间距为 1。

解

(1) 由 DIF-FFT 的基本蝶形结运算流图可知，每个蝶形结有 1 个复数乘法和 2 个复数加法运算。由于每一级蝶形运算的输入数据个数均为 N，因此每一级蝶形运算都有$\frac{N}{2}$个蝶形结，而蝶形运算的级数为 $\log_2 N$，所以整个流程图要计算的蝶形结总数为

$$\frac{N}{2}\cdot\log_2 N$$

所以总共需要的复数乘法次数为

$$\frac{N}{2}\cdot\log_2 N$$

总共需要的复数加法次数为

$$2\cdot\frac{N}{2}\log_2 N = N\log_2 N$$

(2) 每一级蝶形运算中所包含的 W_N 的幂次与蝶形结的分组情况有关。对于 DIF-FFT，第 $m-1$ 列到第 m 列有 2^{m-1}个分组。每个分组中有 $N/2^{m-1}$个数据，其中后一半的数据包含 $W_{N/2^{m-1}}^{r}$因子，r 的取值范围是

$$r=0,1,\cdots,\left(\frac{N}{2^m}-1\right)$$

$$W_{N/2^{m-1}}^{r} = W_N^{r\cdot 2^{m-1}}$$

所以第 $m-1$ 列到第 m 列包含的 W_N 的幂是

$$r\cdot 2^{m-1},\quad r=0,1,\cdots,\left(\frac{N}{2^m}-1\right)$$

(3) 由于 DIF-FFT 是按序列输入顺序分成前后两半，因此第 1 列蝶形结的两个复数输入点的地址间距为$\frac{N}{2}$，第 2 列为$\frac{N}{4}$，…，由此可知第 m 列的间距为

$$\frac{N}{2^m}$$

(4) 由于 DIF-FFT 流图中，蝶形结的相乘系数是在蝶形结输出端的，因此第 m 列利用第 $m-1$ 列的相乘系数，所以第 m 列利用相同系数的各蝶形结的数据地址间距为

$$\frac{N}{2^{m-1}},\quad m=1,2,\cdots,(\log_2 N-1)$$