

第四章 快速傅里叶变换

1. 引言
2. 直接计算DFT的问题及改进的途径
3. 按时间抽选 (DIT) 的基-2FFT算法
4. 按频率抽选 (DIF) 的基-2FFT算法
5. 离散傅里叶反变换IDFT的快速计算方法



1. 引言

FFT不是一种新算法, 只是DFT的一种快速算法。

Coolley和Tukey在1965年发表的“机器计算傅里叶级数的一种算法”

桑德和图基的快速算法的出现。



2. 直接计算DFT的问题及改进的途径

DFT和IDFT的变换公式

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, k = 0, 1, \dots, N-1 \quad (4.1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, n = 0, 1, \dots, N-1 \quad (4.2)$$

二者的差别就在于 W_N 的指数符号不同，以及相差一个常数乘因子 $1/N$ 。

每计算一个 $X(k)$ 值，需要 N 次复数乘法，以及 $(N-1)$ 次复数加法，而 $X(k)$ 一共有 N 个值点，因此完成整个DFT运算总共需要 N^2 次复数乘法， $N(N-1)$ 次复数加法。

复数运算实际上是由实数运算来完成的，
因此(4.1)式可写成

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{N-1} \left\{ \operatorname{Re}[x(n)] + j \operatorname{Im}[x(n)] \right\} \left\{ \operatorname{Re}[W_N^{nk}] + j \operatorname{Im}[W_N^{nk}] \right\} \\ &= \sum_{n=0}^{N-1} \left\{ \operatorname{Re}[x(n)] \operatorname{Re}[W_N^{nk}] - \operatorname{Im}[x(n)] \operatorname{Im}[W_N^{nk}] + j \left(\operatorname{Re}[x(n)] \operatorname{Im}[W_N^{nk}] + \operatorname{Im}[x(n)] \operatorname{Re}[W_N^{nk}] \right) \right\} \end{aligned} \quad (4.3)$$

可以看出，一次复数乘法需要4次实数乘法和2次实数加法，一次复数加法需要2次实数加法。

因此，每计算一个 $X(k)$ 值，需要 $4N$ 次实数乘法，以及 $2N + 2(N-1) = 2(2N-1)$ 次实数加法，整个DFT运算总共需要 $4N^2$ 次实数乘法， $2N(2N-1)$ 次实数加法。

存在问题：

直接计算DFT，乘法次数和加法次数都是和 N^2 成正比。
当 N 很大时，运算量很大，例如， $N=8$ 时，需要64次复乘， $N=1024$ 时，需要1048576次复乘。



减少DFT运算工作量的途径：利用 W_N^{nk} 的固有特性。

(1) W_N^{nk} 的对称性： $(W_N^{nk})^* = W_N^{-nk}$

(2) W_N^{nk} 的周期性： $W_N^{nk} = W_N^{(n+N)k} = W_N^{n(k+N)}$

(3) W_N^{nk} 的可约性： $W_N^{nk} = W_{mN}^{mnk} = W_{N/m}^{nk/m}$

可以得出

$$W_N^{n(N-k)} = W_N^{(N-n)k} = W_N^{-nk}, \quad W_N^{N/2} = -1, \quad W_N^{(k+N/2)} = -W_N^k$$

$$\begin{aligned} W_N^{n(N-k)} &= W_N^{nN} \cdot W_N^{-nk} \\ &= e^{-j\frac{2\pi}{N}nN} \cdot W_N^{-nk} = W_N^{-nk} \end{aligned}$$

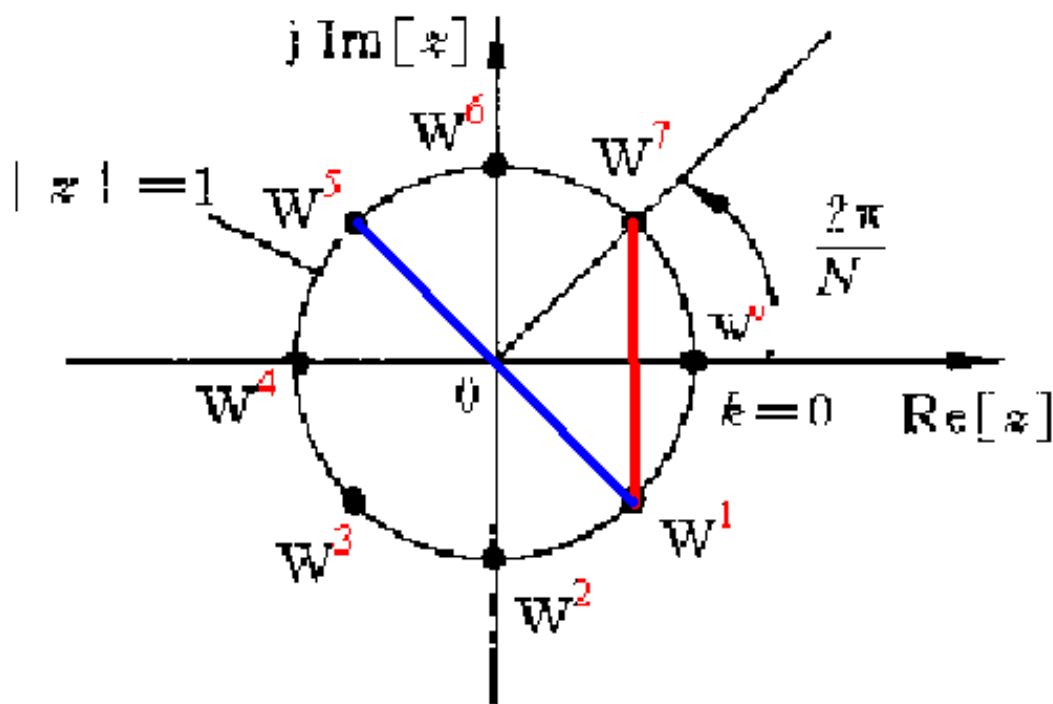
实际办法：

(1) 用上述特性对项合并

(2) 将长序列的DFT分解为短序列的DFT，减小N值。



对称性与周期性



$$W_N^{(n+N/2)k} = -W_N^{nk}$$

$$\left(W_N^{nk}\right)^* = W_N^{-nk}$$



四点的DFT

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

进行化简，得

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & -1 & -W^1 \\ 1 & -1 & 1 & -1 \\ 1 & -W^1 & -1 & W^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$



将该矩阵的第二列和第三列交换，得

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & W^1 & -W^1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -W^1 & W^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix}$$

由此得出

$$X(0) = [x(0) + x(2)] + [x(1) + x(3)]$$

$$X(1) = [x(0) - x(2)] + [x(1) - x(3)]W^1$$

$$X(2) = [x(0) + x(2)] - [x(1) + x(3)]$$

$$X(3) = [x(0) - x(2)] - [x(1) - x(3)]W^1$$



快速傅里叶变换正是基于这样的思想发展进来的，主要分为两大类：

DIT：按时间抽选

DIF：按频率抽选



3. 按时间抽选的基-2FFT算法

3.1 算法原理

先设序列点数为 $N = 2^L$ ，按 n 的奇偶进行分解

$$\left. \begin{aligned} x(2r) &= x_1(r) \\ x(2r+1) &= x_2(r) \end{aligned} \right\}, \quad r = 0, 1, \dots, \frac{N}{2} - 1 \quad (4-4)$$

将DFT化为

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n \text{ 为偶数}} x(n)W_N^{nk} + \sum_{n \text{ 为奇数}} x(n)W_N^{nk} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x_1(r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x_2(r)(W_N^2)^{rk} \end{aligned}$$

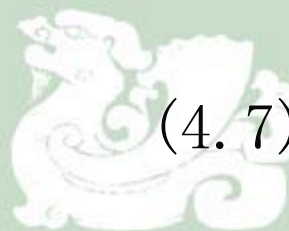


利用系数 W_N^{nk} 的可约性, 即 $W_N^2 = e^{-j\frac{2\pi}{N} \cdot 2} = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$
得

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x_2(r) W_{N/2}^{rk} = X_1(k) + W_N^k X_2(k) \quad (4.5)$$

$$X_1(k) = \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{N/2}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_{N/2}^{rk} \quad (4.6)$$

$$X_2(k) = \sum_{r=0}^{\frac{N}{2}-1} x_2(r) W_{N/2}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_{N/2}^{rk} \quad (4.7)$$



应用系数的周期性 $W_{N/2}^{rk} = W_{N/2}^{r(k+\frac{N}{2})}$, 可得

$$X_1\left(\frac{N}{2}+k\right) = \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{N/2}^{r\left(\frac{N}{2}+k\right)} = \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{N/2}^{rk} = X_1(k) \quad (4.8)$$

$$X_2\left(\frac{N}{2}+k\right) = X_2(k) \quad (4.9)$$

再考虑性质 $W_N^{\left(\frac{N}{2}+k\right)} = W_N^{N/2} W_N^k = -W_N^k$ (4.10)

把(4.8), (4.9), (4.10)代入(4.5)式, 将 $X(k)$ 表达成前后两部分, 前部分为

$$X(k) = X_1(k) + W_N^k X_2(k), \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (4.11)$$

后部分为

$$\begin{aligned} X\left(k + \frac{N}{2}\right) &= X_1\left(k + \frac{N}{2}\right) + W_N^{r\left(k+\frac{N}{2}\right)} X_2\left(k + \frac{N}{2}\right) \\ &= X_1(k) - W_N^k X_2(k), \quad k = 0, 1, \dots, \frac{N}{2} - 1 \end{aligned} \quad (4.12)$$



这样，4.11、12式只要0-(N/2-1)区间的所有 $X_1(k)$ 和 $X_2(k)$ 的值，即可求0到(N-1)区间所有 $X(k)$ 值。
4.11和4.12式用蝶形图表示。

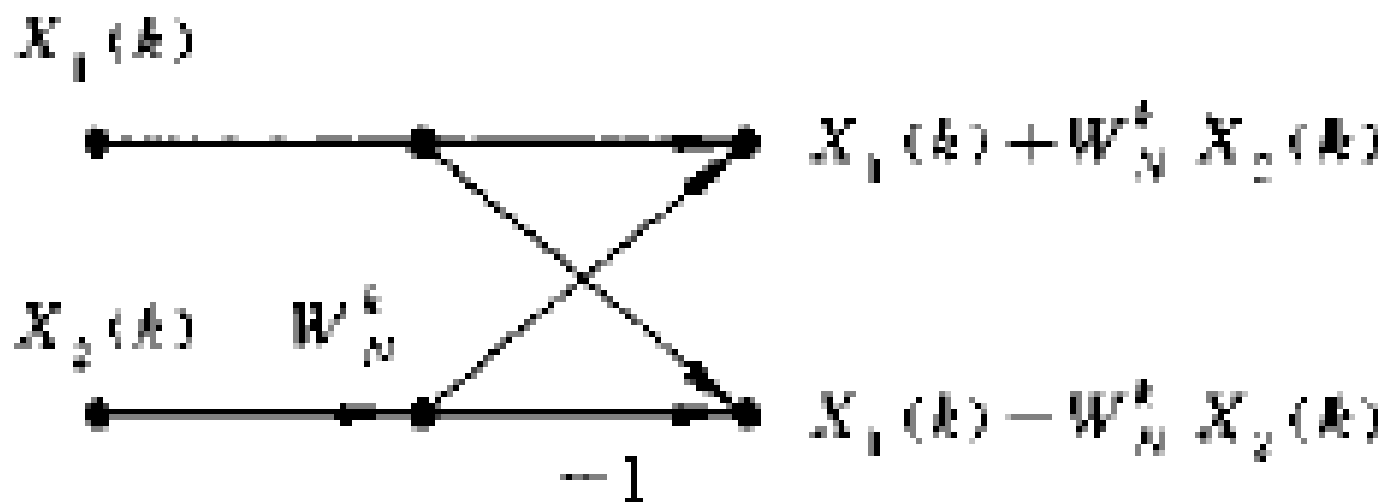


图 4-1 时间抽选法蝶形运算流图符号

N=8的情况如图所示:

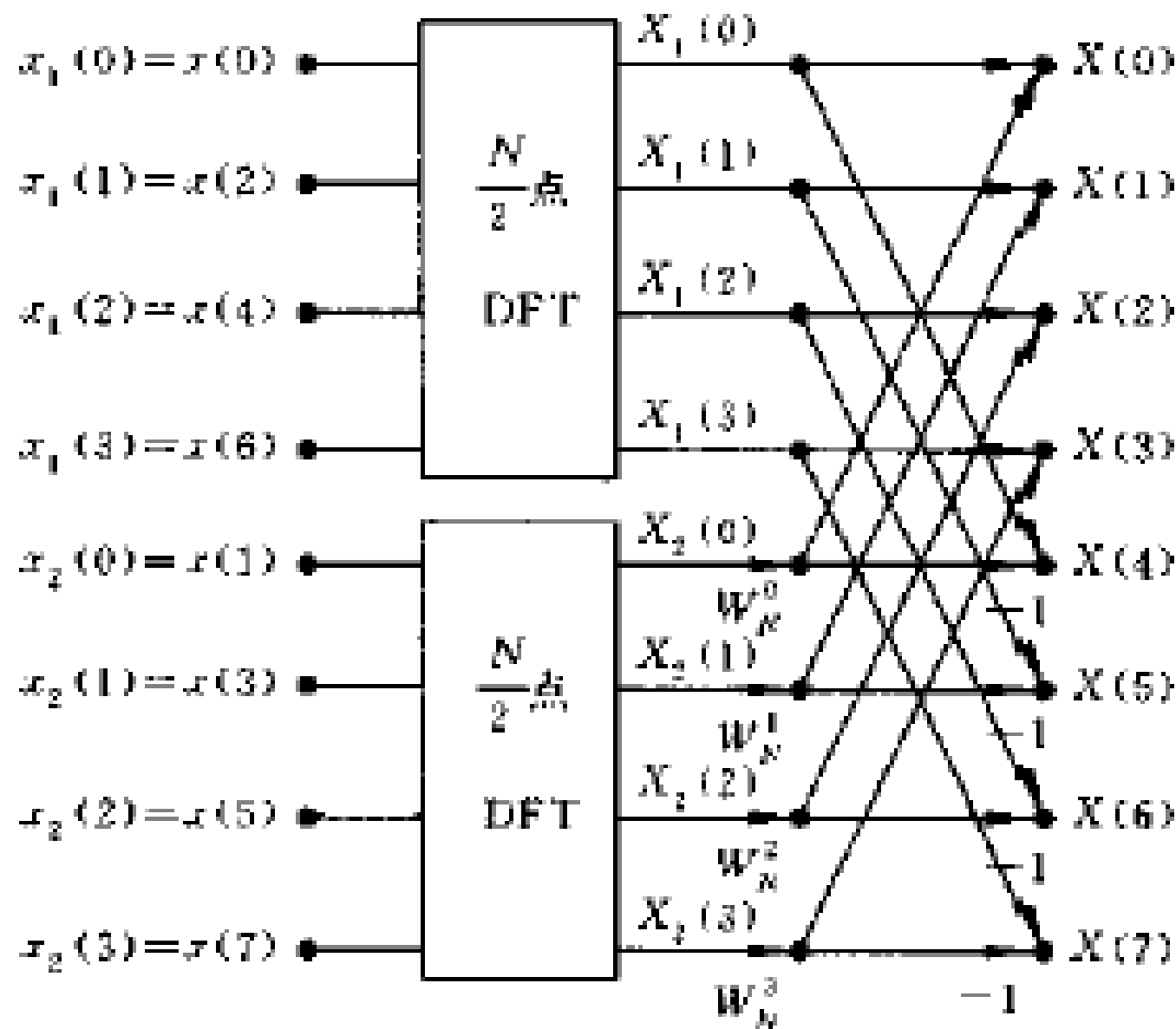


图 4-2 按时间抽选, 将一个 N 点 DFT 分解为两个 $N/2$ 点 DFT

分析：每个蝶形运算需要一次复数乘法 $X_1(k)W_N^k$ 及两次复数加（减）法。通过分解后运算工作量差不多减少到一半。



进一步把N/2点子序列再按奇偶部分分解为两个N/4点的子序列

$$\left. \begin{aligned} x_1(2l) &= x_3(l) \\ x_1(2l+1) &= x_4(l) \end{aligned} \right\}, \quad l = 0, 1, \dots, \frac{N}{4} - 1$$

$$\begin{aligned} X_1(k) &= \sum_{l=0}^{\frac{N}{4}-1} x_1(2l) W_{N/2}^{2lk} + \sum_{l=0}^{\frac{N}{4}-1} x_1(2l+1) W_{N/2}^{(2l+1)k} \\ &= \sum_{l=0}^{\frac{N}{4}-1} x_3(l) W_{N/4}^{lk} + W_{N/2}^k \sum_{l=0}^{\frac{N}{4}-1} x_4(l) W_{N/4}^{lk} \\ &= X_3(k) + W_{N/2}^k X_4(k), \quad k = 0, 1, \dots, \frac{N}{4} - 1 \end{aligned}$$

且 $X_1\left(\frac{N}{4} + k\right) = X_3(k) - W_{N/2}^k X_4(k), \quad k = 0, 1, \dots, \frac{N}{4} - 1$

其中

$$X_3(k) = \sum_{l=0}^{\frac{N}{4}-1} x_3(l) W_{N/4}^{lk} \quad (4-14)$$

$$X_4(k) = \sum_{l=0}^{\frac{N}{4}-1} x_4(l) W_{N/4}^{lk} \quad (4-15)$$

图4-3，给出 $N=8$ 时，在分解为两个 $N/4$ 点DFT，由两个 $N/4$ 点DFT组合成 $N/2$ 点DFT的流图。

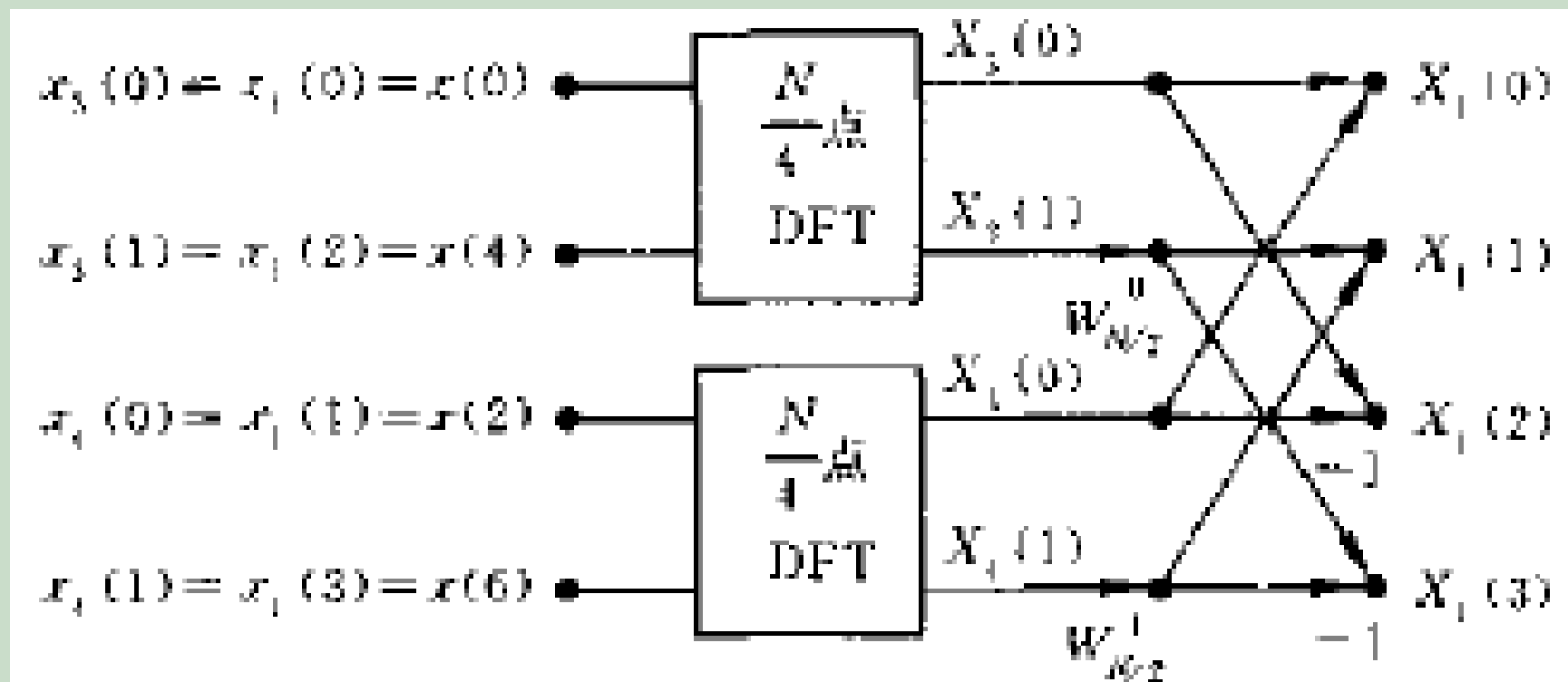


图 4-3 由两个 $N/4$ 点 DFT 组合成一个 $N/2$ 点 DFT

$X_2(k)$ 也可进行同样分解:

$$\left. \begin{aligned} X_2(k) &= X_5(k) + W_{N/2}^* X_6(k) \\ X_2\left(\frac{N}{4} + k\right) &= X_5(k) - W_{N/2}^* X_6(k) \end{aligned} \right\} \quad k = 0, 1, \dots, \frac{N}{4} - 1$$

其中

$$X_5(k) = \sum_{l=0}^{\frac{N}{4}-1} x_2(2l) W_{N/4}^k = \sum_{l=0}^{\frac{N}{4}-1} x_5(l) W_{N/4}^k \quad (4-16)$$

$$X_6(k) = \sum_{l=0}^{\frac{N}{4}-1} x_2(2l+1) W_{N/4}^k = \sum_{l=0}^{\frac{N}{4}-1} x_6(l) W_{N/4}^k \quad (4-17)$$



一个 $N=8$ 点DFT就可分解为四个 $N/4=2$ 点DFT如图

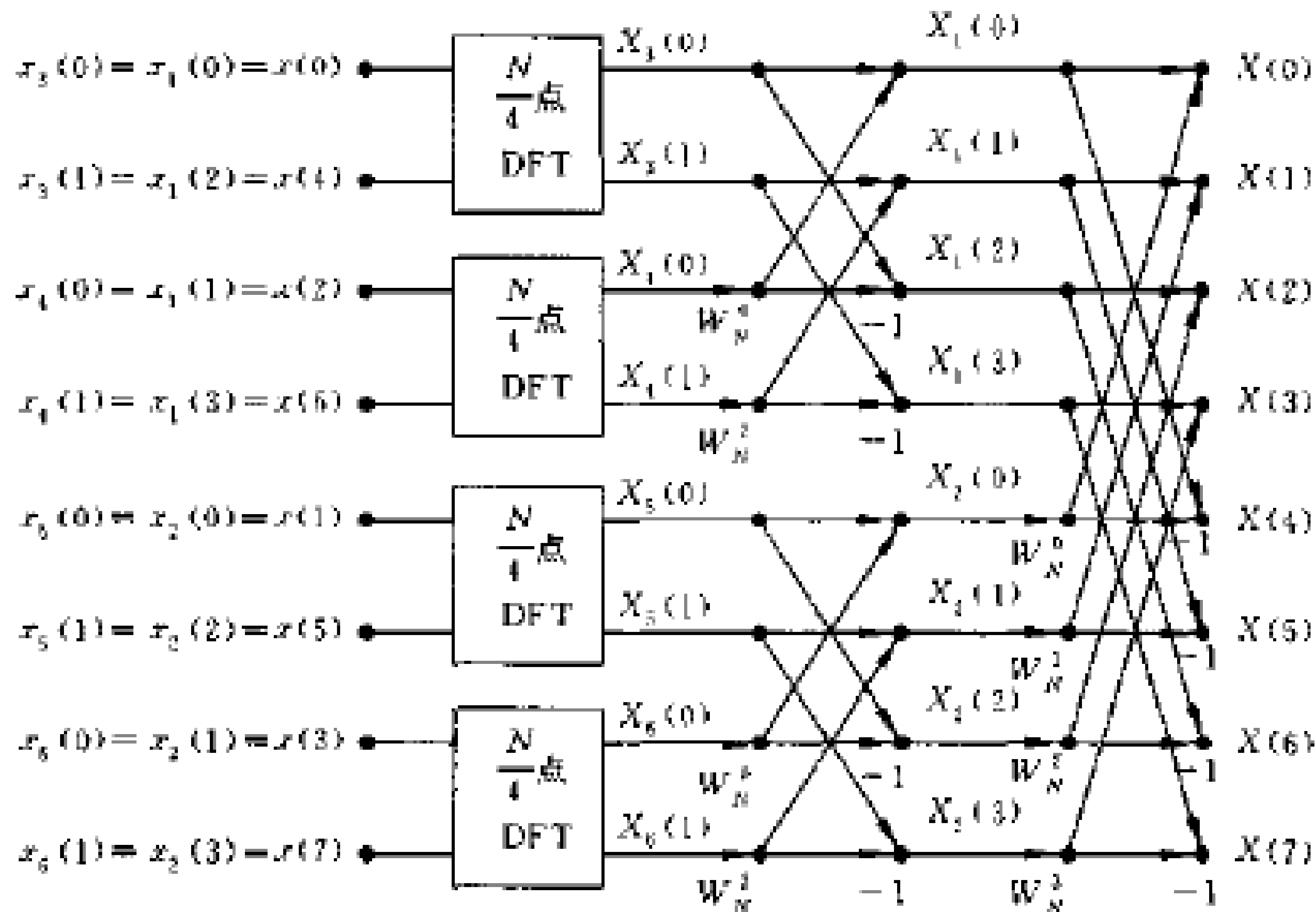


图 4-4 按时间抽选, 将一个 N 点 DFT 分解为四个 $N/4$ 点 DFT ($N=8$)

序列按奇偶分解标号变化讨论 (N=8)

1、第一次分解：两个N/2点序列：

偶序列

$$x_1(2l) = x_1(r)$$

奇序列

$$x_1(2l+1) = x_2(r)$$

$$r = 0, 1, \dots, \frac{N}{2} - 1$$

r	0	1	2	3
$n = 2r$	0	2	4	6

r	0	1	2	3
$n = 2r + 1$	1	3	5	7



2、第二次分解，每个N/2点子序列按其奇偶分解为两个N/4点子序列

$$l = 0, 1, \dots, \frac{N}{4} - 1$$

偶序列中的偶数序列

$$x_1(2l) = x_1(l)$$

l	0	1
$r = 2l$	0	2
$n = 2r$	0	4

偶序列中的奇数序列

$$x_1(2l+1) = x_4(l)$$

l	0	1
$r = 2l+1$	1	3
$n = 2r$	2	6

奇序列中的偶数序列

$$x_2(2l) = x_3(l)$$

l	0	1
$r = 2l$	0	2
$n = 2r+1$	1	5

奇序列中的奇数序列

$$x_2(2l+1) = x_6(l)$$

l	0	1
$r = 2l+1$	1	3
$n = 2r+1$	3	7

3、最后是2点的DFT

对于例子 $N=8$ ，就是4个 $N/4=2$ 点的DFT。

如：

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

$$X_4(0) = \sum_{l=0}^1 x_4(l) W_{N/4}^{lk} = x_4(0) + W_2^0 x_4(1) = x(2) + W_N^0 x(6)$$

$$X_4(1) = \sum_{l=0}^1 x_4(l) W_{N/4}^{lk} = x_4(0) + W_2^1 x_4(1) = x(2) - W_N^0 x(6)$$

$$\text{其中 } W_2^1 = e^{-j\frac{2\pi}{2} \cdot 1} = e^{-j\pi} = -1$$

- 这种方法的每一步分解都是按输入序列在时间上的次序是属于偶数还是属于奇数来分解为两个更短的子序列，所以称为“按时间抽选法”。

3.2 运算量分析

(1) 直接DFT复数算法次数是 N^2

(2) FFT复数乘法次数是 $(N/2) * L$

当 $N=2^L$ 时，一共有 L 级蝶形，每级有 $N/2$ 个蝶形运算组成，一个蝶形运算有1次复乘，2次复加运算。

DFT和FFT算法的计算量之比为

$$\frac{N^2}{\frac{N}{2}L} = \frac{N^2}{\frac{N}{2}\log_2 N} = \frac{2N}{\log_2 N} \quad (4-20)$$

结论：FFT比DFT更优越，当 N 越大时，优点更明显。

表 4-1 FFT 算法与直接 DFT 算法的比较

N	N^2	$\frac{N}{2} \log_2 N$	$N^2 / \left(\frac{N}{2} \log_2 N \right)$
2	4	1	4.0
4	16	4	4.0
8	64	12	5.4
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.4
128	16384	448	36.6
256	65536	1024	64.0
512	262144	2304	113.8
1024	1048576	5120	204.8
2048	4194304	11264	372.4

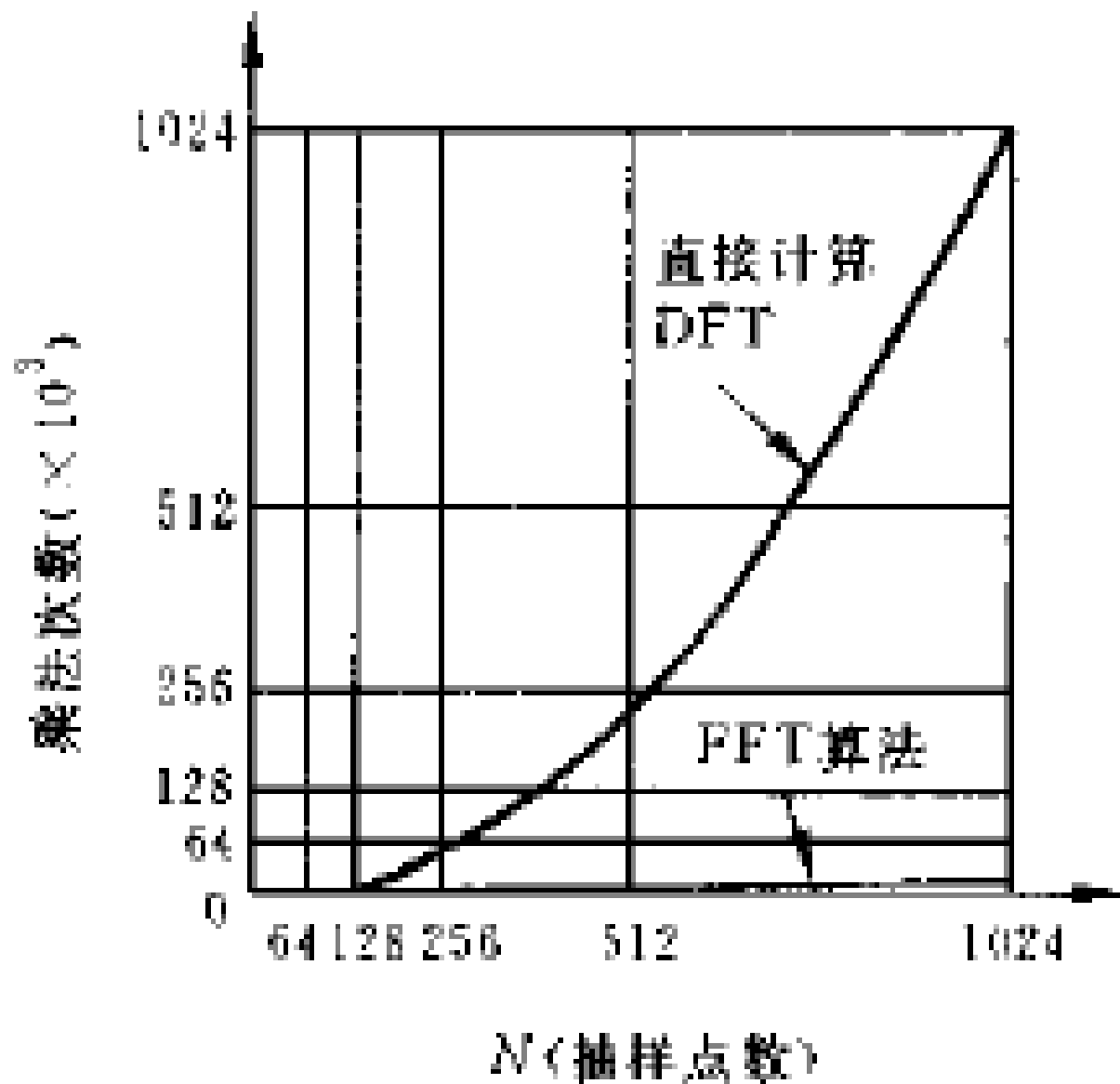


图 4-6 直接计算 DFT 与 FFT 算法所需乘法次数的比较

3.3 按时间抽选的FFT算法特点

1. 原位运算

每个蝶形结构完成下述基本迭代运算：

$$\begin{aligned} X_m(k) &= X_{m-1}(k) + X_{m-1}(j)W_N^r \\ X_m(j) &= X_{m-1}(k) - X_{m-1}(j)W_N^r \end{aligned} \quad (4-21)$$

4.21的蝶形运算如图4-7所示。

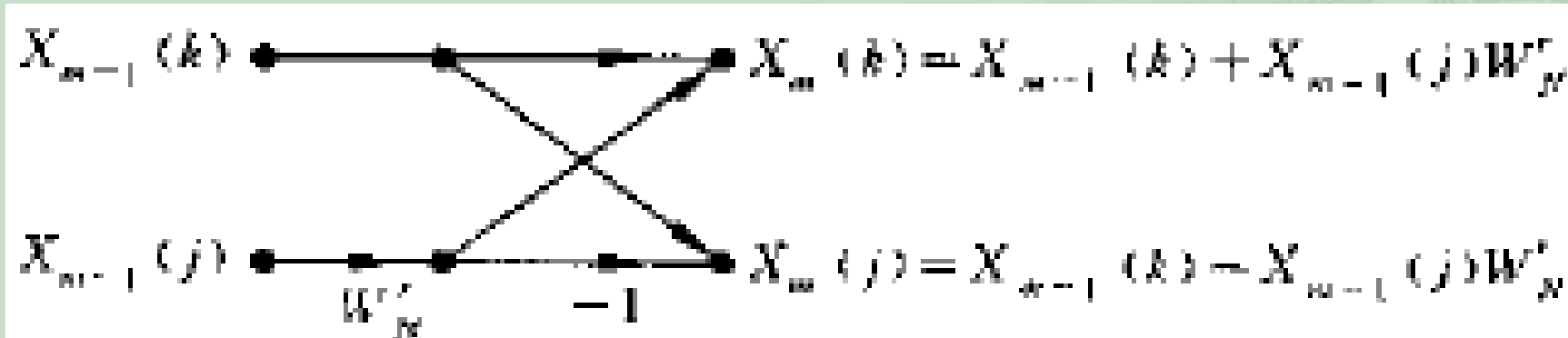


图 4-7 按时间抽选蝶形运算结构

蝶形的两个输出值仍放回蝶形的两个输入值所在的存储器中，中间不需要其他的存储器。每列的 $N/2$ 个蝶形运算完成后，再开始下一列的蝶形运算，这样存储数据就只要 N 个存储单元。

下一级的蝶形运算仍采取这样的原位运算，只不过进入蝶形的组合关系有所不同。



2. 倒位序规律

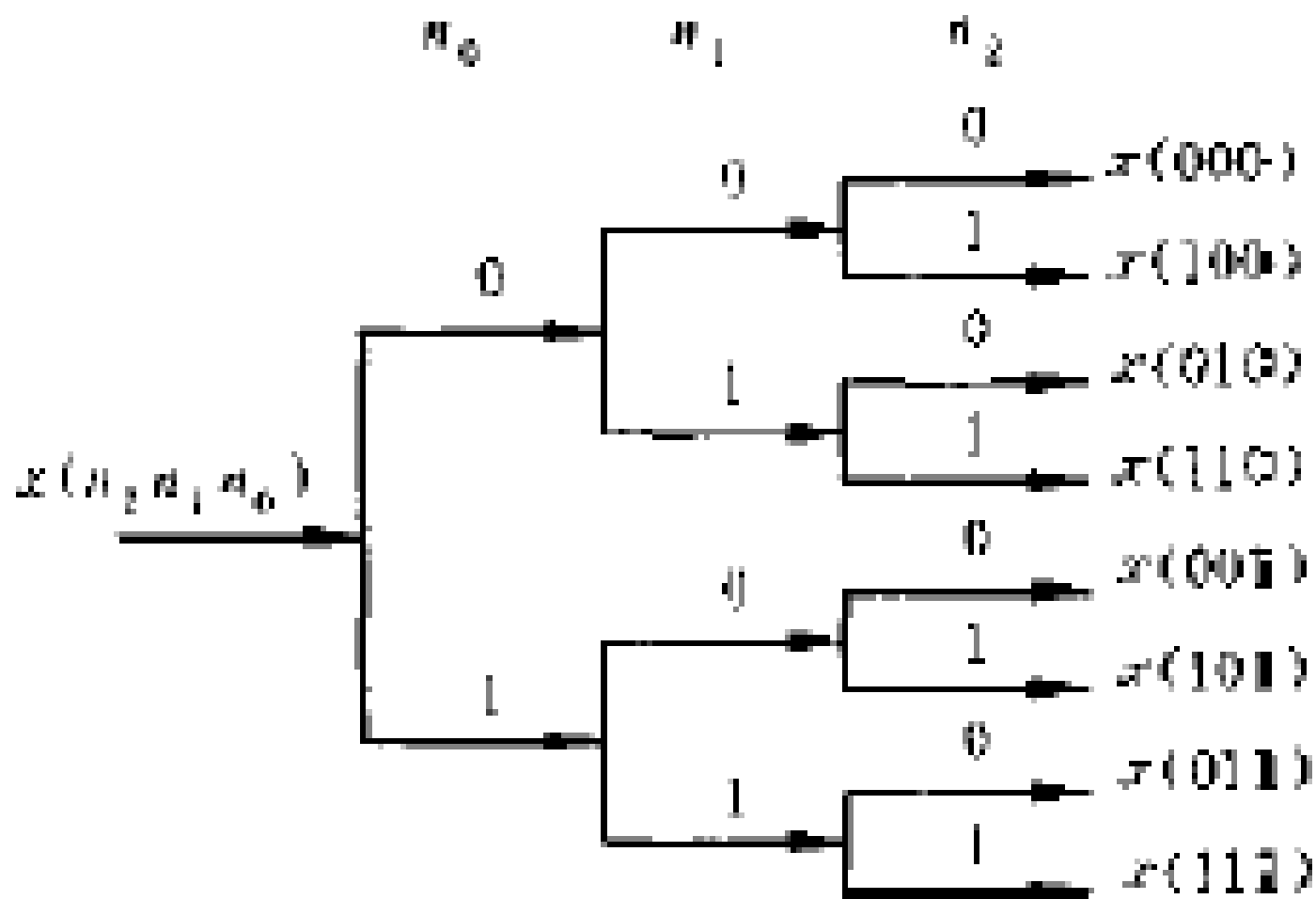


图 4-8 描述倒位序的树状图

3. 倒位序的实现：通过变址运算完成

表 4-2 码位的倒位序 ($N=8$)

自然顺序 (n)	二进制数	倒位序二进制数	倒位序顺序 (n')
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

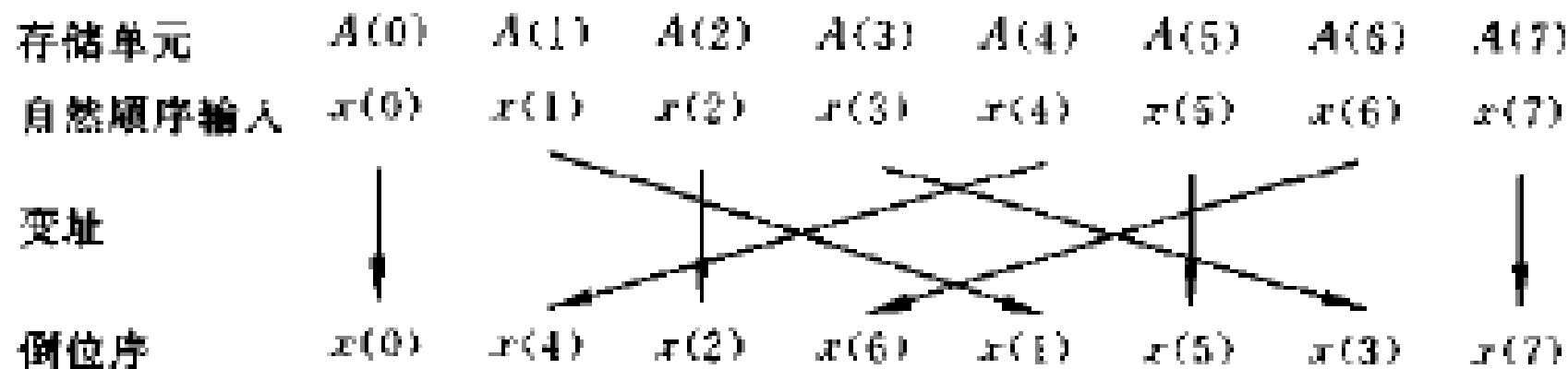


图 4-9 倒位序的变址处理

4. 存储单元

输入序列 $x(n) (n=0, 1, \dots, N-1)$ N 个单元

系数 $W_N^r (r=0, 1, \dots, N/2-1)$ $N/2$ 个单元



3.4 按时间抽选的FFT算法的其它形式流程图

对于任何流图，只要保持各节点所连的支路和传输系数不变，则不论节点怎么排列所得的流图总是等效的，只是数据的提取和存放的次序不同而已。因此就可以有按之间抽取的FFT算法的若干形式。



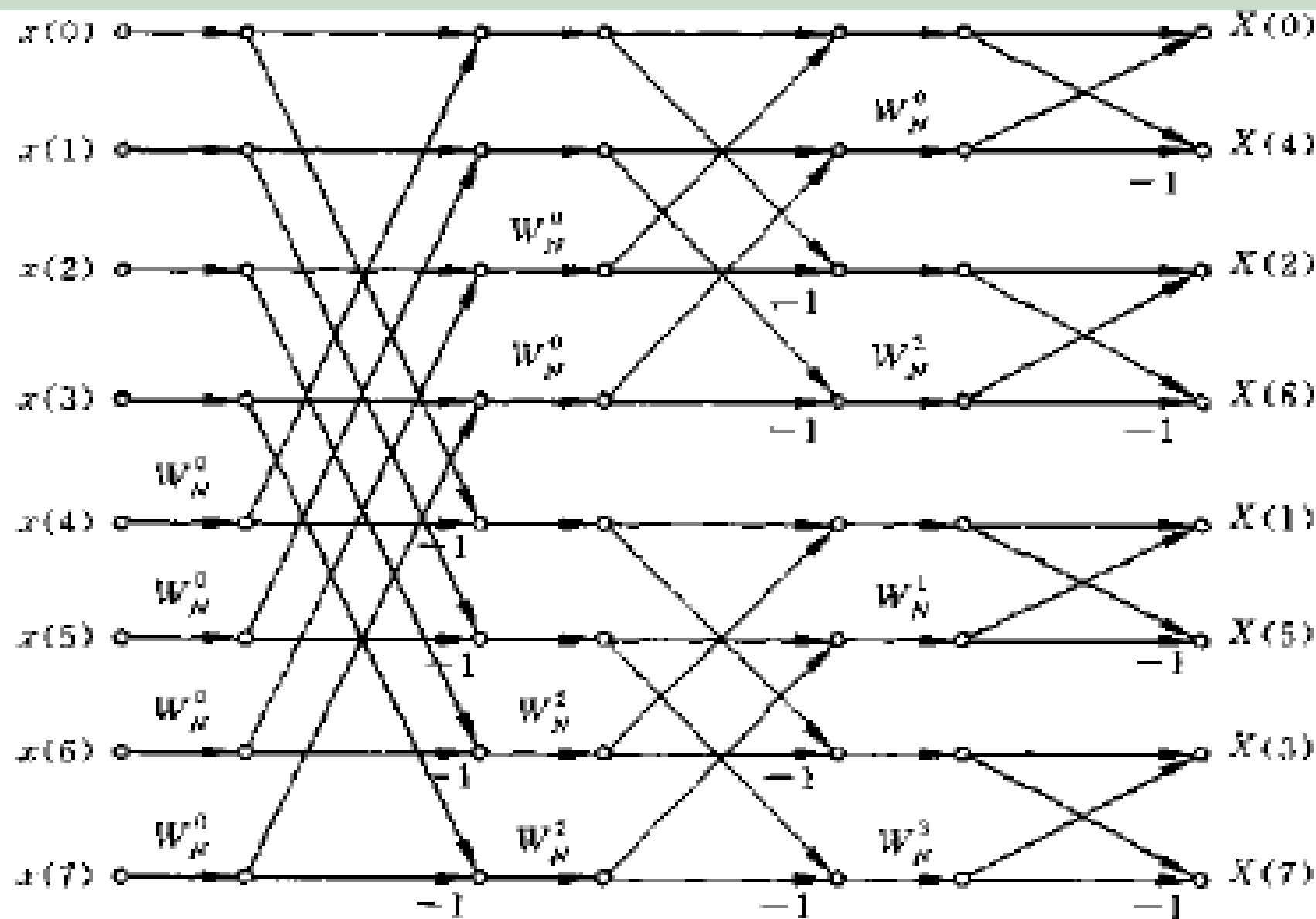


图 4-10 按时间抽选, 输入自然顺序、输出倒位序的 FFT 流图

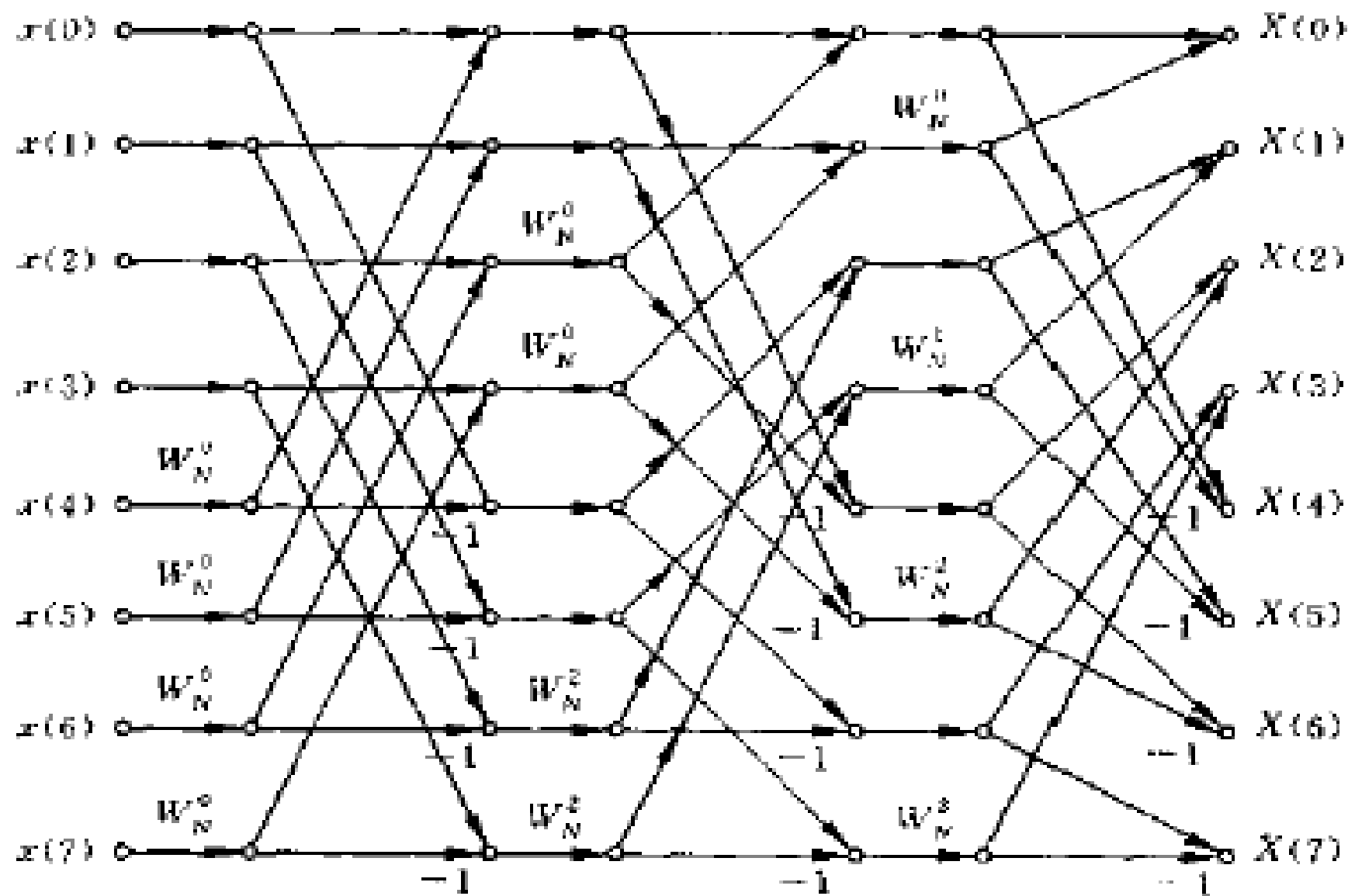


图 4-11 按时间抽选+输入输出皆为自然顺序的 FFT 流图

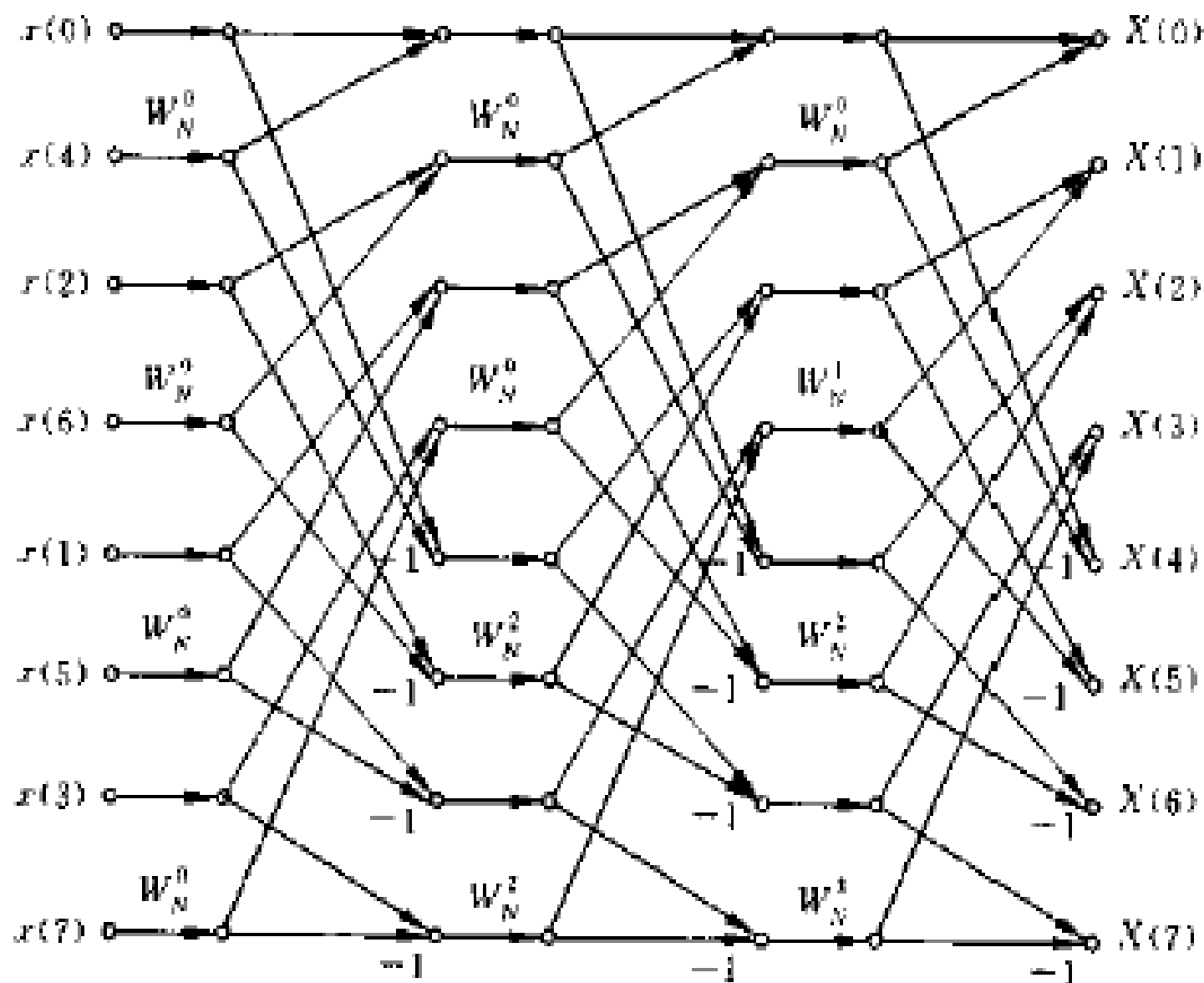


图 4-12 按时间抽选, 各级具有相同几何形状, 输入倒位序, 输出自然顺序的 FFT 流图

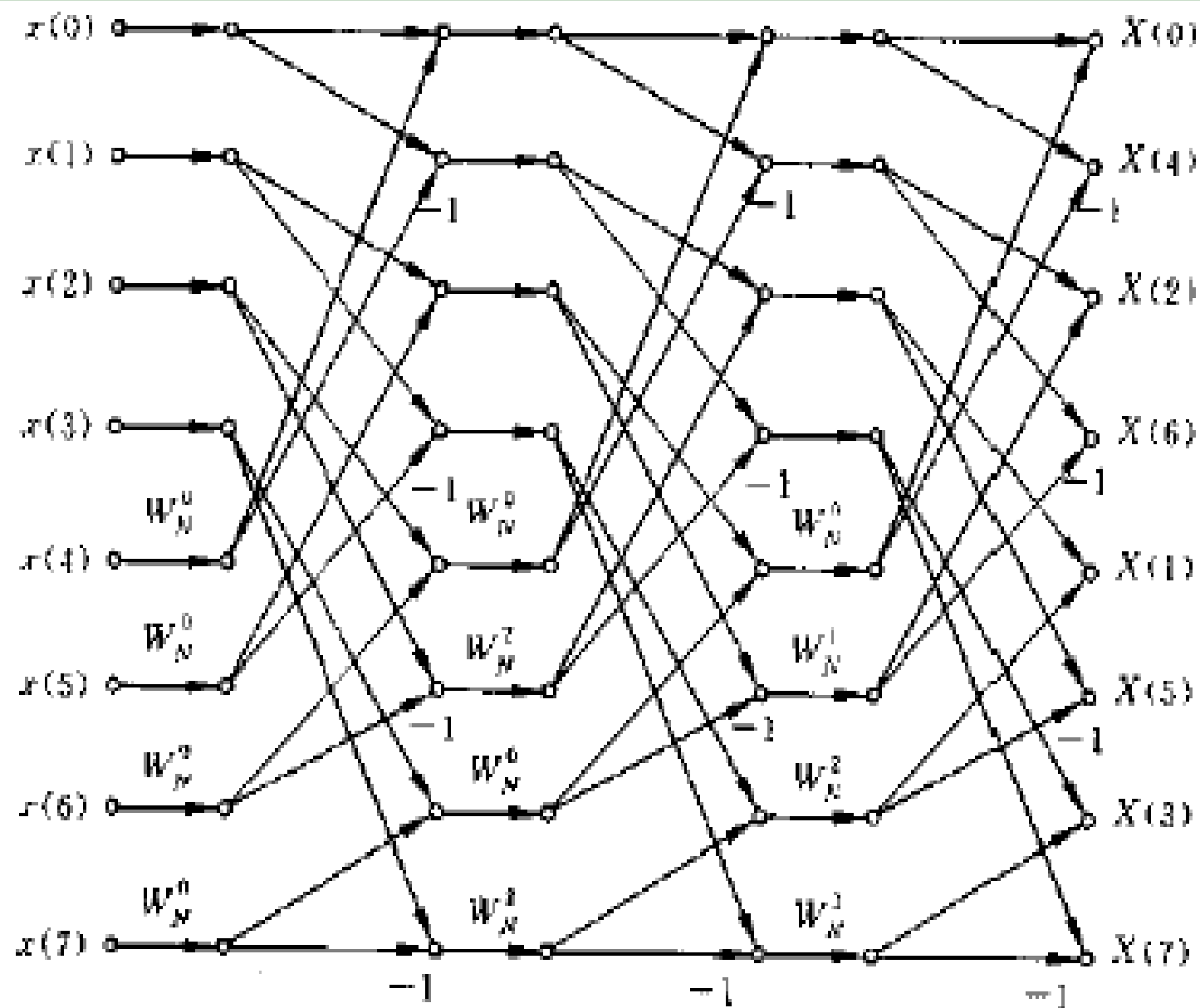


图 4-13 按时间抽选, 各级具有相同几何形状, 输入自然顺序、输出倒位序的 FFT 流图

4. 按频率抽选（DIF）的基-2FFT算法 （桑德一图基算法）

DIF算法是将输出序列 $X(k)$ 按其顺序的奇偶分解为越来越短的序列。仍设序列的点数为 $N=2^L$, L 为整数。

先将输入序列按 n 的顺序分为前后两半。

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{nk} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x(n + \frac{N}{2})W_N^{(n+\frac{N}{2})k} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x(n + \frac{N}{2}) \cdot W_N^{Nk/2} \right] \cdot W_N^{nk}, \quad k = 0, 1, \dots, N-1 \end{aligned}$$



由于 $W_N^{N/2} = -1$, 因此 $W_N^{Nk/2} = (-1)^k$

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

当 k 为偶数时, $(-1)^k=1$; 当 k 为奇数时, $(-1)^k=-1$ 。

因此按 k 的奇偶可将 $X(k)$ 分为两部分。



$$\left. \begin{array}{l} \text{令} \quad k = 2r \\ \quad k = 2r + 1 \end{array} \right\}, \quad r = 0, 1, \dots, \frac{N}{2} - 1$$

则

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

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



$$\left. \begin{aligned} \text{令 } x_1(n) &= x(n) + x\left(n + \frac{N}{2}\right) \\ x_2(n) &= \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \end{aligned} \right\} \quad n = 0, 1, \dots, \frac{N}{2} - 1$$

$$\text{则 } X(2r) = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) \cdot W_{N/2}^{nr}$$

$$X(2r+1) = \sum_{n=0}^{\frac{N}{2}-1} x_2(n) \cdot W_{N/2}^{nr}$$



可以用蝶形运算来表示

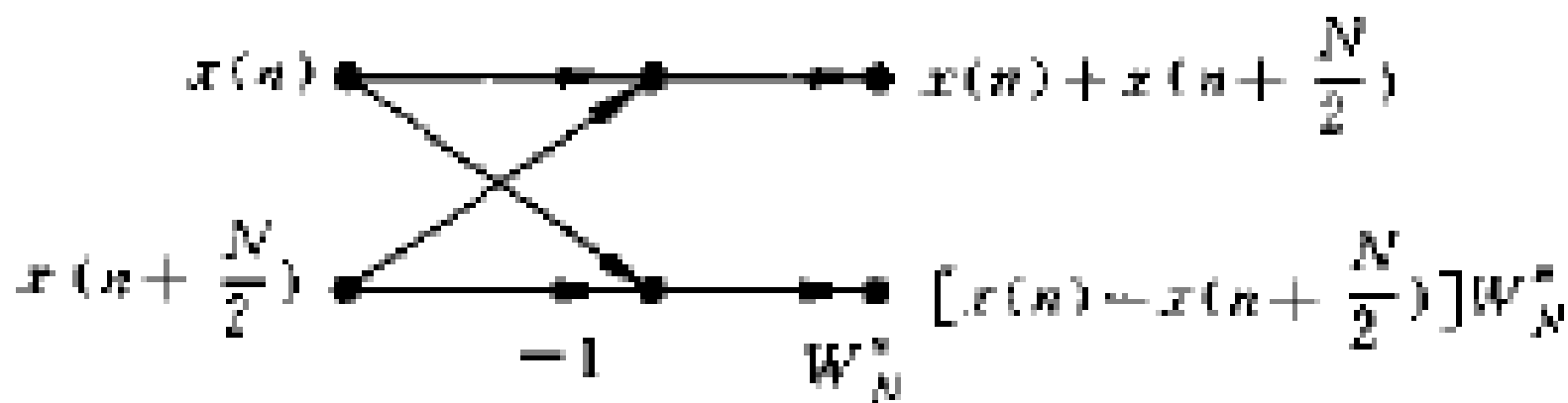
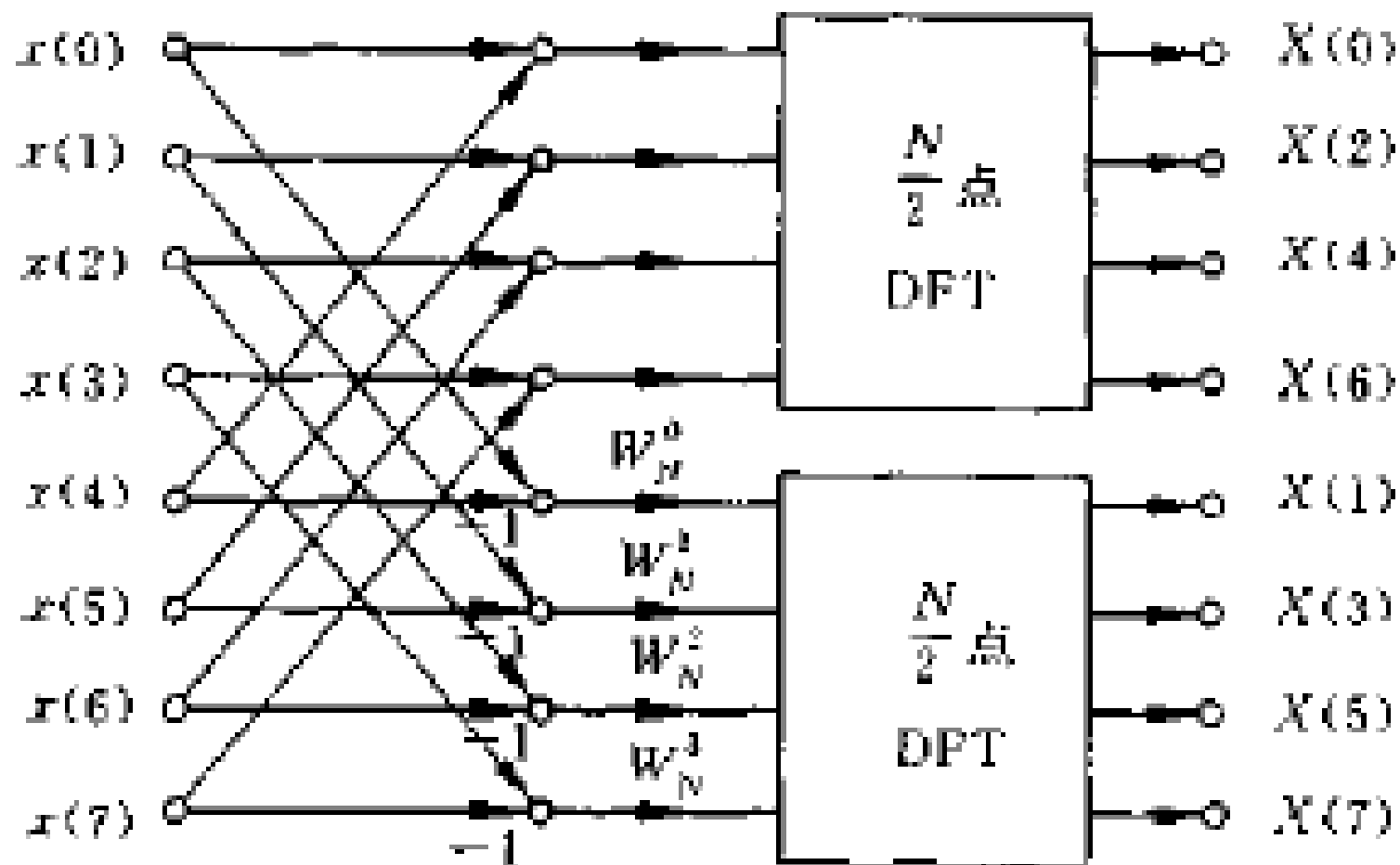


图 4-14 按频率抽选蝶形运算流图符号



N=8时，按k分解结果如图



按频率抽选，将 N 点 DFT 分解为两个 $N/2$ 点 DFT 的组合 ($N=8$)

5. 离散傅里叶反变换IDFT的快速计算方法

DFT和IDFT的变换公式

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

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

比较可知，只要把DFT运算中的每一个系数 W_N^{nk} 变成 W_N^{-nk} ，最后再乘常数 $1/N$ ，则以上所有按时间抽选或按频率抽选的FFT都可以拿来运算IDFT。



上面的方法比较简单，但是要稍微改动FFT的程序和参数才能实现，下面的方法不改变FFT的程序计算IFFT：
对DFT的反变换公式取共轭

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

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

只要先将 $X(k)$ 取共轭，然后直接利用FFT子程序，最后将结果取共轭，并乘以 $1/N$ ，就可以得到 $x(n)$ 的值。

例：假设一次复乘的时间为 $1\ \mu\text{s}$ ，而且假定一个DFT总共需要的时间由计算所有乘法所需的时间确定。

1) 直接计算一个1024点的DFT需要多少时间？

2) 计算基2—FFT需要多少时间？

解：1) $t = N^2 \cdot 10^{-6}\text{s} \approx 1.05\text{s}$

2) 基2—FFT复乘的次数为 $(N/2) \log_2 N = 5120$

$t = 5120 \cdot 10^{-6}\text{s} \approx 5.12\text{ms}$



例：对一个连续时间信号 $x_a(t)$ 进行采样，在1s内采集了4096个数据，

- 1) 若采样后没有产生频谱混叠，则信号 $x_a(t)$ 的最高频率时多少？
- 2) 若计算采样信号的4096点的DFT，频谱之间的频率间隔是多少Hz？

解：1) 因为在1s内采集了4096个数据，

所以 $f_s = 4096\text{Hz}$

要采样后不产生频谱混叠， $f_s > 2f_h$

因此信号的最高频率 $< f_s/2 = 2048\text{Hz}$

2) 频谱之间的频率间隔也就是频率分辨率 F_0

$$f_s / F_0 = N \quad F_0 = f_s / N = 4096\text{Hz} / 4096 = 1\text{Hz}$$

课外作业

P200

3、

