数字信号处理

周治国

2023.10

第四章 快速傅里叶变换

§4-6 分裂基FFT算法

一、背景

对更快速算法的需求

$$基-2\ FFT \sim \frac{N}{2}\log_2^N$$

$$\downarrow$$

1984年，杜梅尔（P.Douhamel）

霍尔曼（H.Hollman）

基-2 FFT

基-4 FFT

$$\longrightarrow 分裂基\ FFT \sim \frac{N}{3}\log_2^N$$

§ 4-6 分裂基FFT算法

基4-DIF FFT算法

$$\forall\, x(n), \qquad 0 \le n \le N-1, N = 2^{\nu}$$

***DFT*[*x*(*n*)]** → 更快的FFT算法

设 ***N=Pq, P=N/4, q=4***

令 $n=Pn_1+n_0=N/4\, n_1+n_0$;　$0\le n_1 \le 3$,　$0\le n_0 \le (N/4)-1$

$k=4k_1+k_0$;　$0\le k_1 \le (N/4)-1$,　$0\le k_0 \le 3$

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

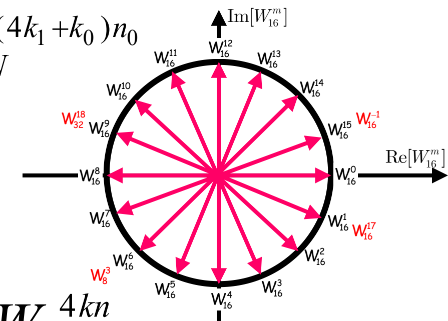
$$=\sum_{n_0=0}^{N/4-1}\sum_{n_1=0}^{3}x(\frac{N}{4}n_1+n_0)W_N^{k(\frac{N}{4}n_1+n_0)}$$

$$=\sum_{n_0=0}^{N/4-1}\left[x(n_0)W_4^{0}+x(n_0+\frac{N}{4})W_4^{k}+x(n_0+\frac{N}{2})W_4^{2k}+x(n_0+\frac{3N}{4})W_4^{3k}\right]W_N^{kn_0}$$

$$X(k)=X(4k_1+k_0) \qquad k_0=0,1,2,3 \qquad (k_1\to k,\ n_0\to n)$$

$$=\sum_{n_0=0}^{N/4-1}\left[x(n_0)+x(n_0+\frac{N}{4})W_4^{(4k_1+k_0)}+x(n_0+\frac{N}{2})W_4^{2(4k_1+k_0)}+x(n_0+\frac{3N}{4})W_4^{3(4k_1+k_0)}\right]W_N^{(4k_1+k_0)n_0}$$

$$=\sum_{n_0=0}^{N/4-1}\left[x(n_0)+x(n_0+\frac{N}{4})W_4^{k_0}+x(n_0+\frac{N}{2})W_4^{2k_0}+x(n_0+\frac{3N}{4})W_4^{3k_0}\right]W_N^{(4k_1+k_0)n_0}$$



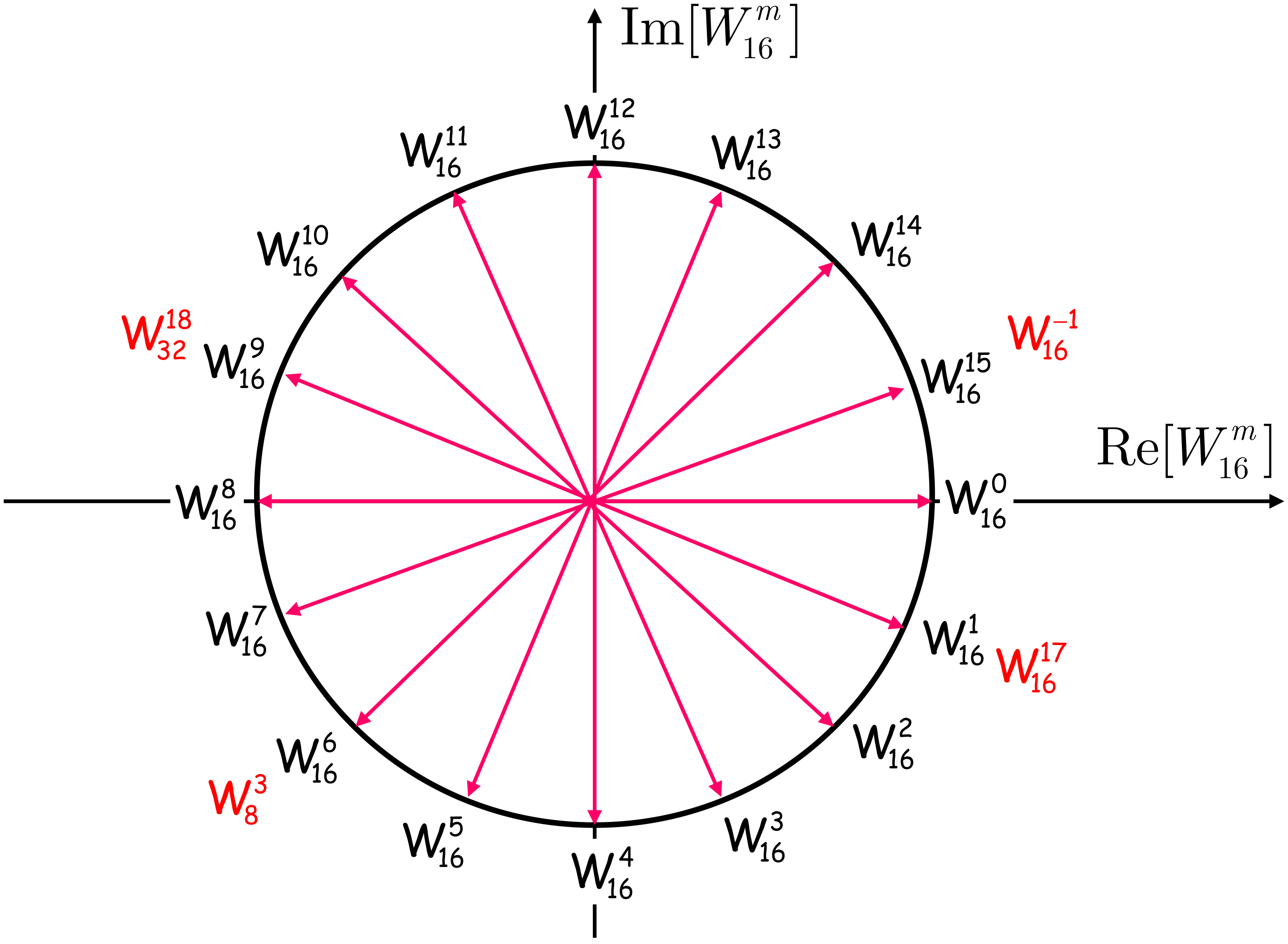
在$k_0=0,1,2,3$时，用k表示k_1，n表示n_0

$$\begin{cases}
X(4k)=\sum_{n=0}^{N/4-1}\left[x(n)+x(n+\frac{N}{4})+x(n+\frac{N}{2})+x(n+\frac{3N}{4})\right]W_N^{4kn} \\
X(4k+1)=\sum_{n=0}^{N/4-1}\left[x(n)-jx(n+\frac{N}{4})-x(n+\frac{N}{2})+jx(n+\frac{3N}{4})\right]W_N^{4kn+n} \\
X(4k+2)=\sum_{n=0}^{N/4-1}\left[x(n)-x(n+\frac{N}{4})+x(n+\frac{N}{2})-x(n+\frac{3N}{4})\right]W_N^{4kn+2n} \\
X(4k+3)=\sum_{n=0}^{N/4-1}\left[x(n)+jx(n+\frac{N}{4})-x(n+\frac{N}{2})-jx(n+\frac{3N}{4})\right]W_N^{4kn+3n}
\end{cases}$$

$\downarrow$

基4-*DIF*

$$0\le k\le\frac{N}{4}-1 \qquad (4\text{-}43)$$

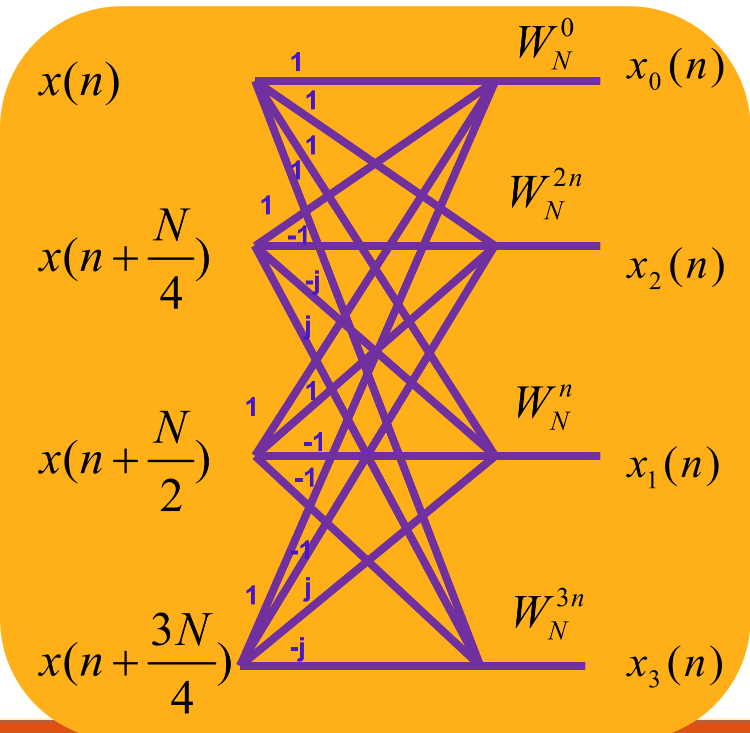


$\mathrm{Im}[W_{16}^{m}]$
$\mathrm{Re}[W_{16}^{m}]$
W_{16}^{0}
W_{16}^{1}
W_{16}^{17}
W_{16}^{2}
W_{16}^{3}
W_{16}^{4}
W_{16}^{5}
W_{16}^{6}
W_{8}^{3}
W_{16}^{7}
W_{16}^{8}
W_{16}^{9}
W_{32}^{18}
W_{16}^{10}
W_{16}^{11}
W_{16}^{12}
W_{16}^{13}
W_{16}^{14}
W_{16}^{15}
W_{16}^{-1}

参考P96 图3-21

基4-DIF FFT算法

$$\begin{cases} X(4k)=\sum_{n=0}^{N/4-1}\left[x(n)+x(n+\frac{N}{4})+x(n+\frac{N}{2})+x(n+\frac{3N}{4})\right]W_N^0 W_{N/4}^{kn} \\ X(4k+1)=\sum_{n=0}^{N/4-1}\left[x(n)-jx(n+\frac{N}{4})-x(n+\frac{N}{2})+jx(n+\frac{3N}{4})\right]W_N^n W_{N/4}^{kn} \\ X(4k+2)=\sum_{n=0}^{N/4-1}\left[x(n)-x(n+\frac{N}{4})+x(n+\frac{N}{2})-x(n+\frac{3N}{4})\right]W_N^{2n} W_{N/4}^{kn} \\ X(4k+3)=\sum_{n=0}^{N/4-1}\left[x(n)+jx(n+\frac{N}{4})-x(n+\frac{N}{2})-jx(n+\frac{3N}{4})\right]W_N^{3n} W_{N/4}^{kn} \end{cases} \qquad 0\le k\le \frac{N}{4}-1$$



$$\begin{cases} x_0(n)=\left[x(n)+x(n+\frac{N}{4})+x(n+\frac{N}{2})+x(n+\frac{3N}{4})\right]W_N^0 \\ x_1(n)=\left[x(n)-jx(n+\frac{N}{4})-x(n+\frac{N}{2})+jx(n+\frac{3N}{4})\right]W_N^n \\ x_2(n)=\left[x(n)-x(n+\frac{N}{4})+x(n+\frac{N}{2})-x(n+\frac{3N}{4})\right]W_N^{2n} \\ x_3(n)=\left[x(n)+jx(n+\frac{N}{4})-x(n+\frac{N}{2})-jx(n+\frac{3N}{4})\right]W_N^{3n} \end{cases}$$

蝶形运算：×— 3次

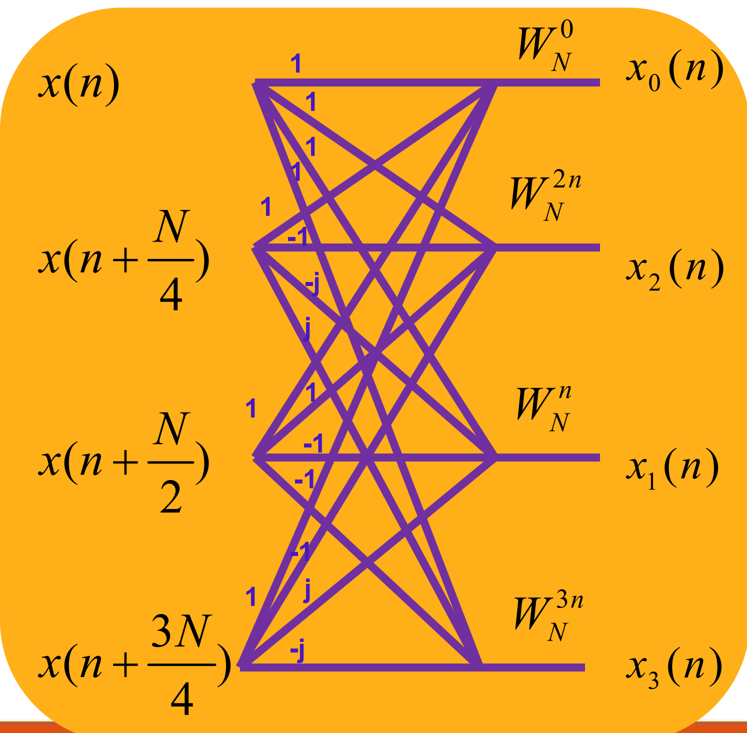
+— ?次

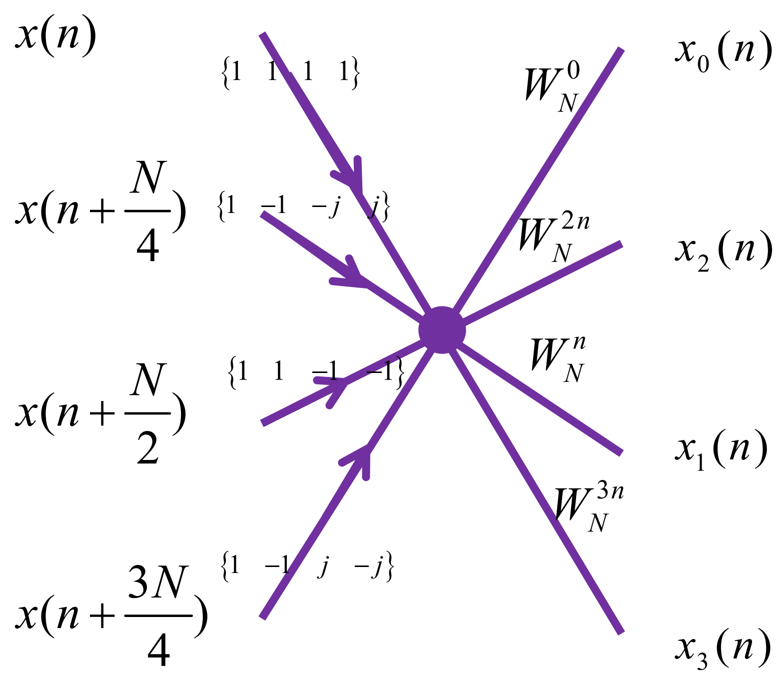
基4-DIF FFT算法

$$
\begin{cases}
x_0(n) = \left[x(n) + x(n+\frac{N}{4}) + x(n+\frac{N}{2}) + x(n+\frac{3N}{4}) \right] W_N^0 \\
x_1(n) = \left\{ \left[x(n) - x(n+\frac{N}{2}) \right] + j\left[x(n+\frac{3N}{4}) - x(n+\frac{N}{4}) \right] \right\} W_N^n \\
x_2(n) = \left[x(n) - x(n+\frac{N}{4}) + x(n+\frac{N}{2}) - x(n+\frac{3N}{4}) \right] W_N^{2n} \\
x_3(n) = \left\{ \left[x(n) - x(n+\frac{N}{2}) \right] + j\left[x(n+\frac{N}{4}) - x(n+\frac{3N}{4}) \right] \right\} W_N^{3n}
\end{cases}
$$

蝶形运算：×— 3次

+— 8次





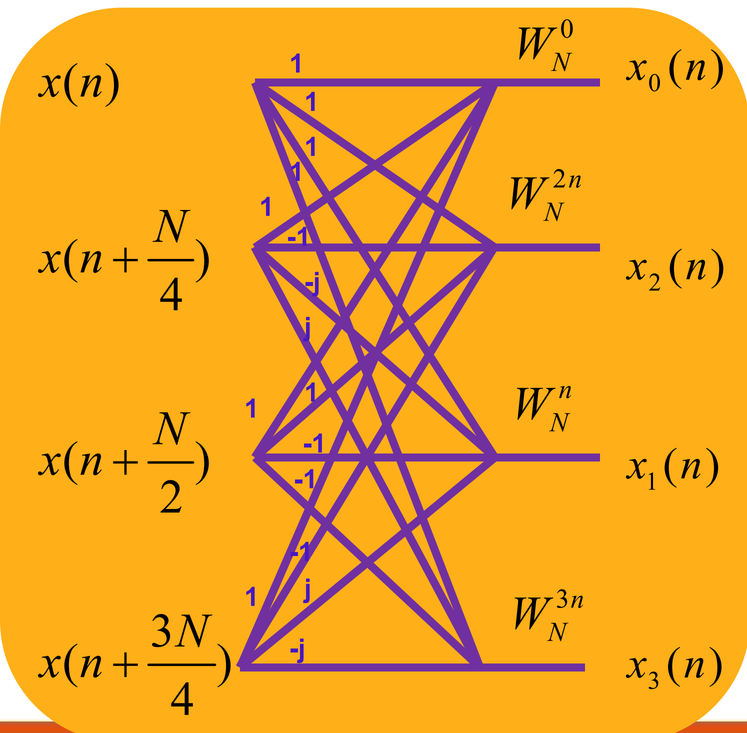
基4-DIF FFT算法

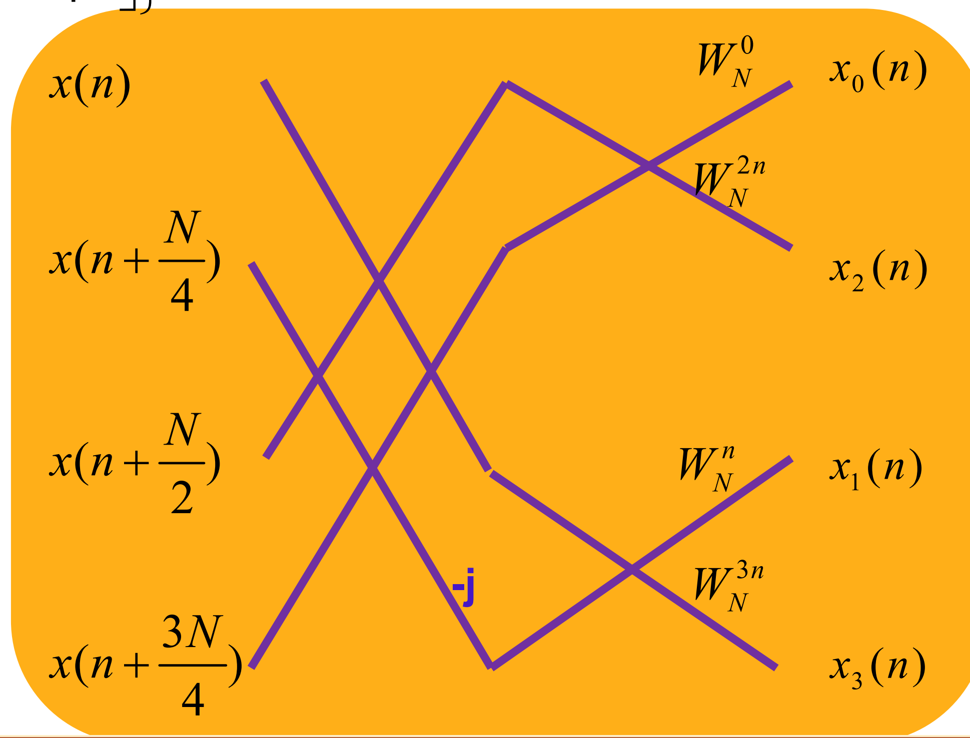
$$
\begin{cases}
x_0(n)=\left\{\left[x(n)+x(n+\frac{N}{2})\right]+\left[x(n+\frac{N}{4})+x(n+\frac{3N}{4})\right]\right\}W_N^0\\
x_1(n)=\left\{\left[x(n)-x(n+\frac{N}{2})\right]-j\left[x(n+\frac{N}{4})-x(n+\frac{3N}{4})\right]\right\}W_N^n\\
x_2(n)=\left\{\left[x(n)+x(n+\frac{N}{2})\right]-\left[x(n+\frac{N}{4})+x(n+\frac{3N}{4})\right]\right\}W_N^{2n}\\
x_3(n)=\left\{\left[x(n)-x(n+\frac{N}{2})\right]+j\left[x(n+\frac{N}{4})-x(n+\frac{3N}{4})\right]\right\}W_N^{3n}
\end{cases}
$$

$$\frac{N}{2}\log_2^N \Rightarrow \frac{3}{4}\frac{N}{2}\log_2^N$$

蝶形运算：×— 3次

+— **8次** (4个x)





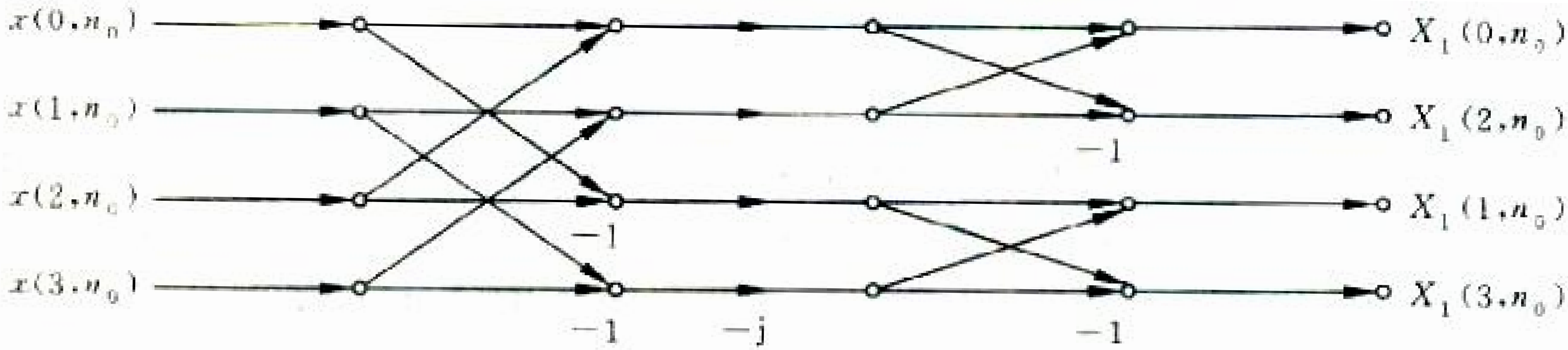


图 4-21 一个基-4 FFT 基本运算的信号流图

基-4

按时间

Or 抽取 FFT算法流图？

按频率

N=16 基-4 按频率抽取FFT流图

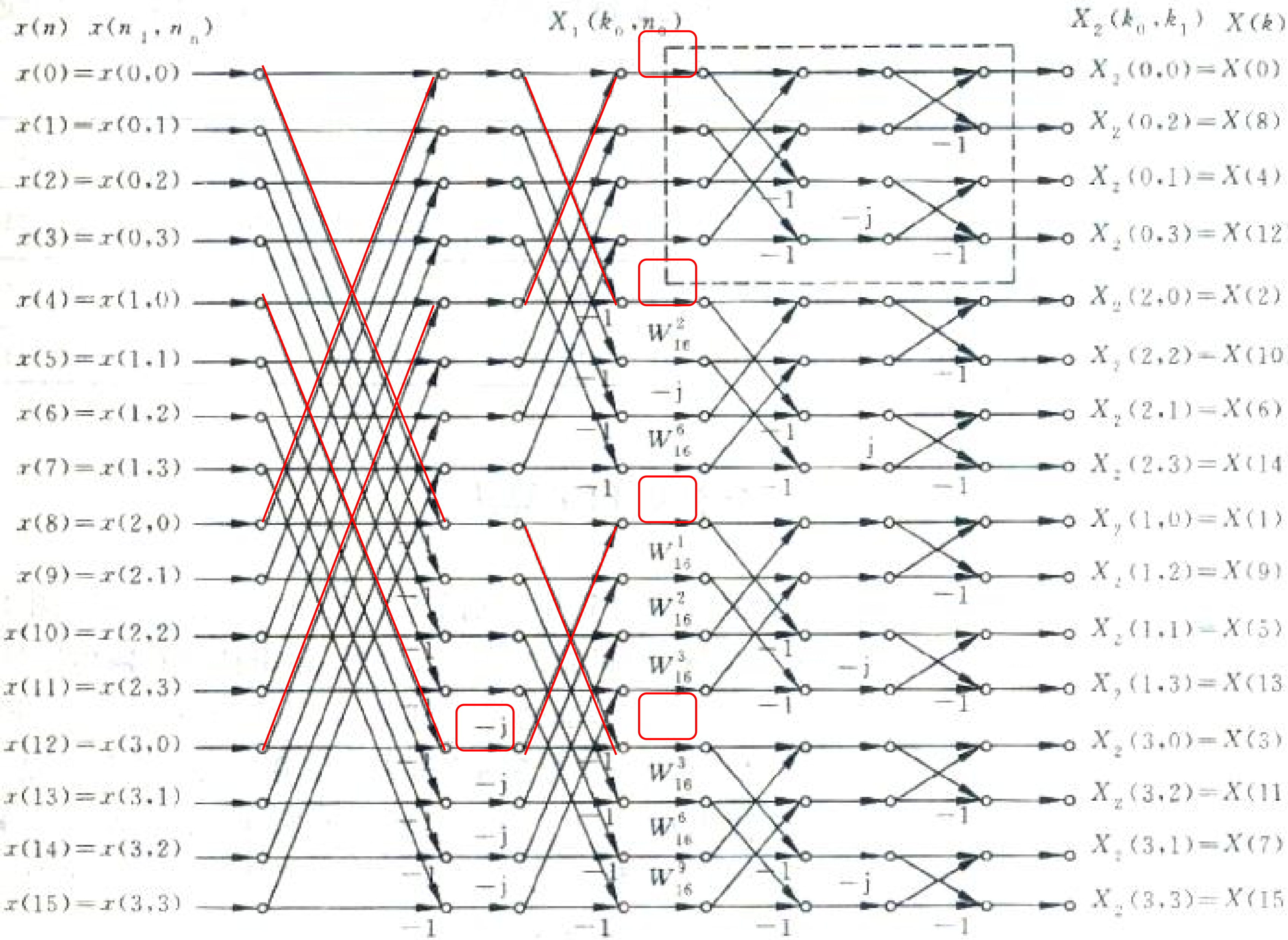


图 4-22 按时间抽选基-4 FFT 流图

回忆： N=12 组合数 基-3x4 FFT流图

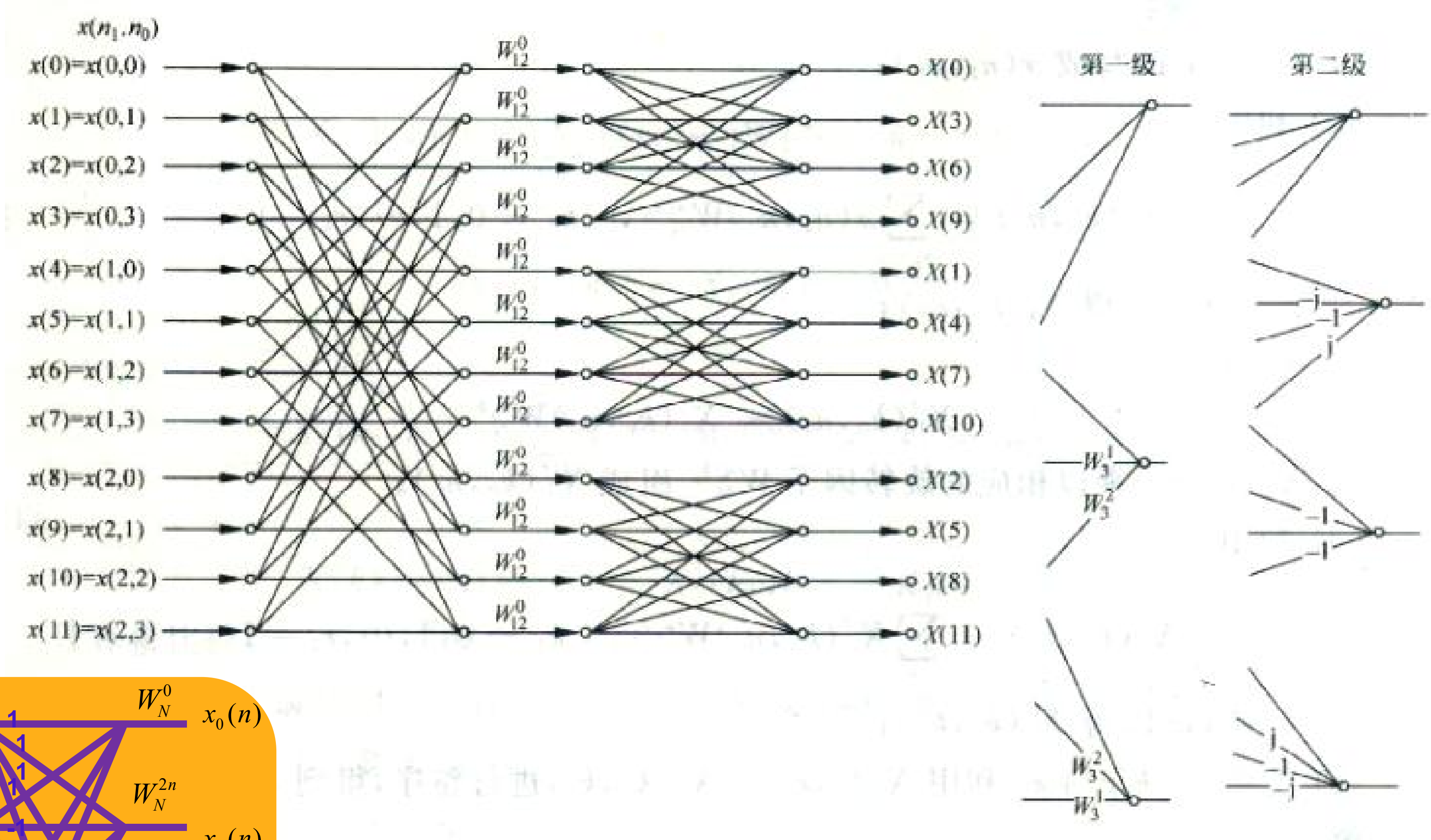
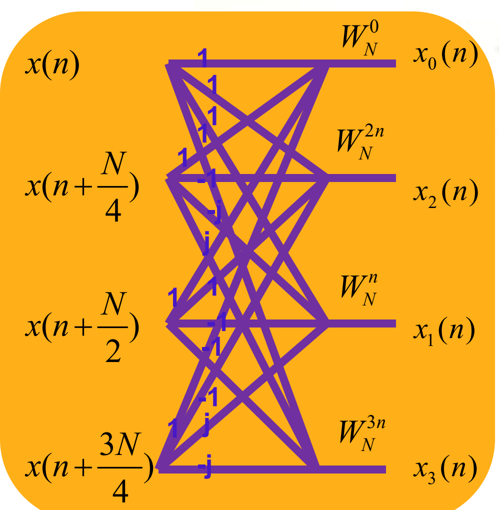


图 P4-5



N=30 基-3x2x5 组合数 FFT流图

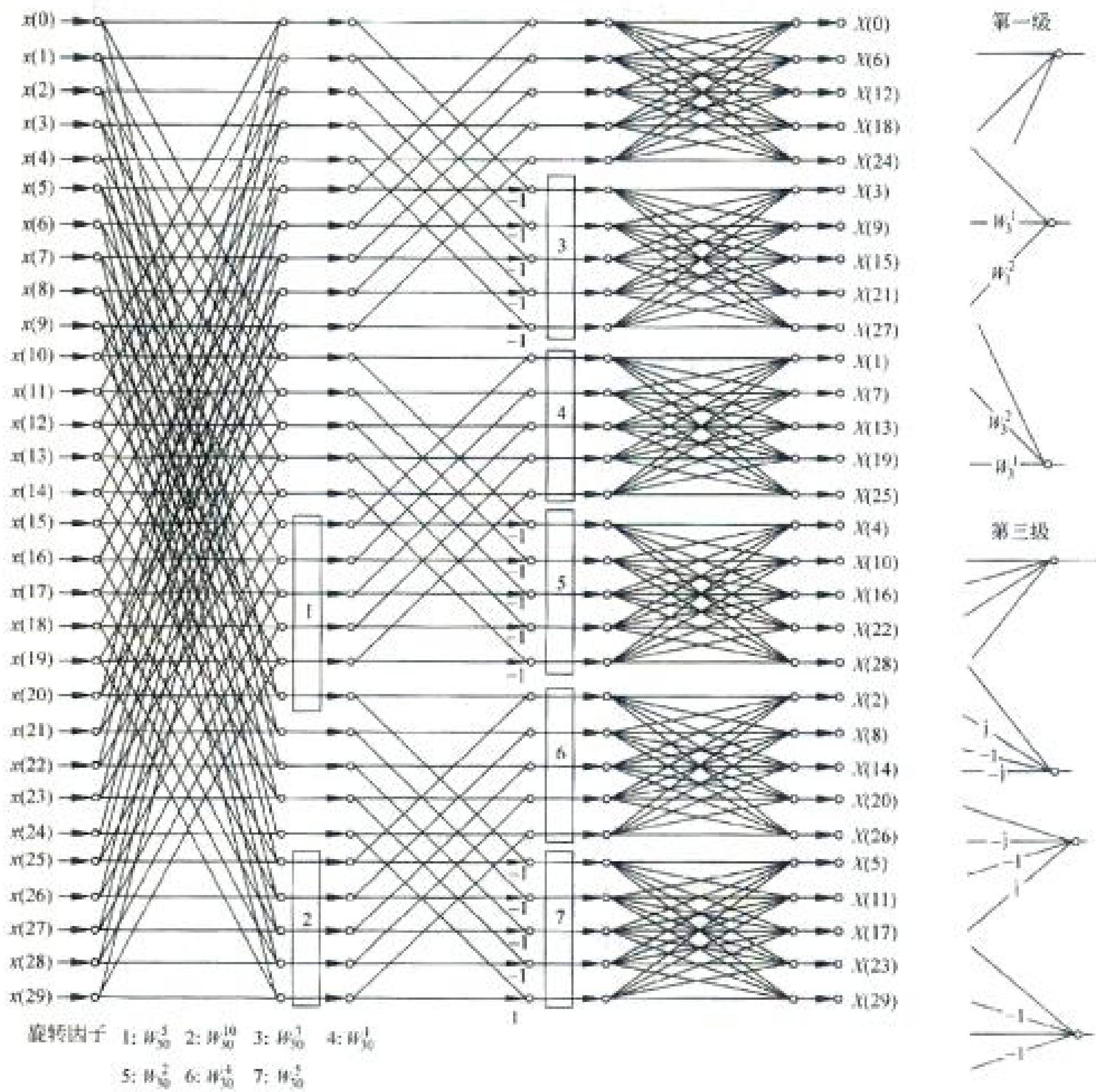
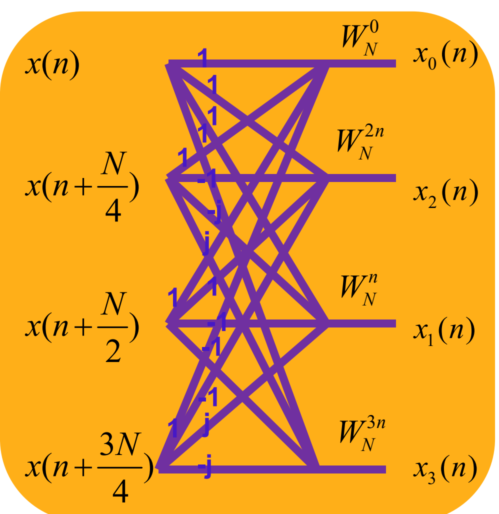


图 P4-6



分裂基FFT算法原理:

合并$X(4k)$与$X(4k+2)$:

$$\begin{cases} X(2k)=\sum_{n=0}^{\frac{N}{2}-1}[x(n)+x(n+\frac{N}{2})]W_N^{2kn}, \qquad 0\le k\le \frac{N}{2}-1 \\ X(4k+1)=\sum_{n=0}^{\frac{N}{4}-1}\left\{\left[\left(x(n)-x(n+\frac{N}{2})\right)-j\left(x(n+\frac{N}{4})-x(n+\frac{3N}{4})\right)\right]W_N^{n}\right\}W_N^{4kn} \\ X(4k+3)=\sum_{n=0}^{\frac{N}{4}-1}\left\{\left[\left(x(n)-x(n+\frac{N}{2})\right)+j\left(x(n+\frac{N}{4})-x(n+\frac{3N}{4})\right)\right]W_N^{3n}\right\}W_N^{4kn} \end{cases}$$

$$0\le k\le \frac{N}{4}-1 \qquad (4\text{-}44)$$

令

$$\begin{cases} x_2(n) \stackrel{\Delta}{=} x(n) + x(n+\frac{N}{2}), \qquad 0 \le n \le \frac{N}{2}-1 \\ x_4^1(n) \stackrel{\Delta}{=} \left[\left(x(n) - x(n+\frac{N}{2}) \right) - j \left(x(n+\frac{N}{4}) - x(n+\frac{3N}{4}) \right) \right] W_N^{n} \\ x_4^2(n) \stackrel{\Delta}{=} \left[\left(x(n) - x(n+\frac{N}{2}) \right) + j \left(x(n+\frac{N}{4}) - x(n+\frac{3N}{4}) \right) \right] W_N^{3n} \end{cases}$$

$$0 \le n \le \frac{N}{4}-1$$

$\downarrow$

L形蝶形运算：×— 2次

+— 6次

则(4-44)式变为:

$$\begin{cases} X(2k)=\sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_N^{2kn}=\sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_{\frac{N}{2}}^{kn}=DFT[x_2(n)] \\ X(4k+1)=\sum_{n=0}^{\frac{N}{4}-1} x_4^1(n)W_N^{4kn}=\sum_{n=0}^{\frac{N}{4}-1} x_4^1(n)W_{\frac{N}{4}}^{kn}=DFT[x_4^1(n)] \\ X(4k+3)=\sum_{n=0}^{\frac{N}{4}-1} x_4^2(n)W_N^{4kn}=\sum_{n=0}^{\frac{N}{4}-1} x_4^2(n)W_{\frac{N}{4}}^{kn}=DFT[x_4^2(n)] \end{cases}$$

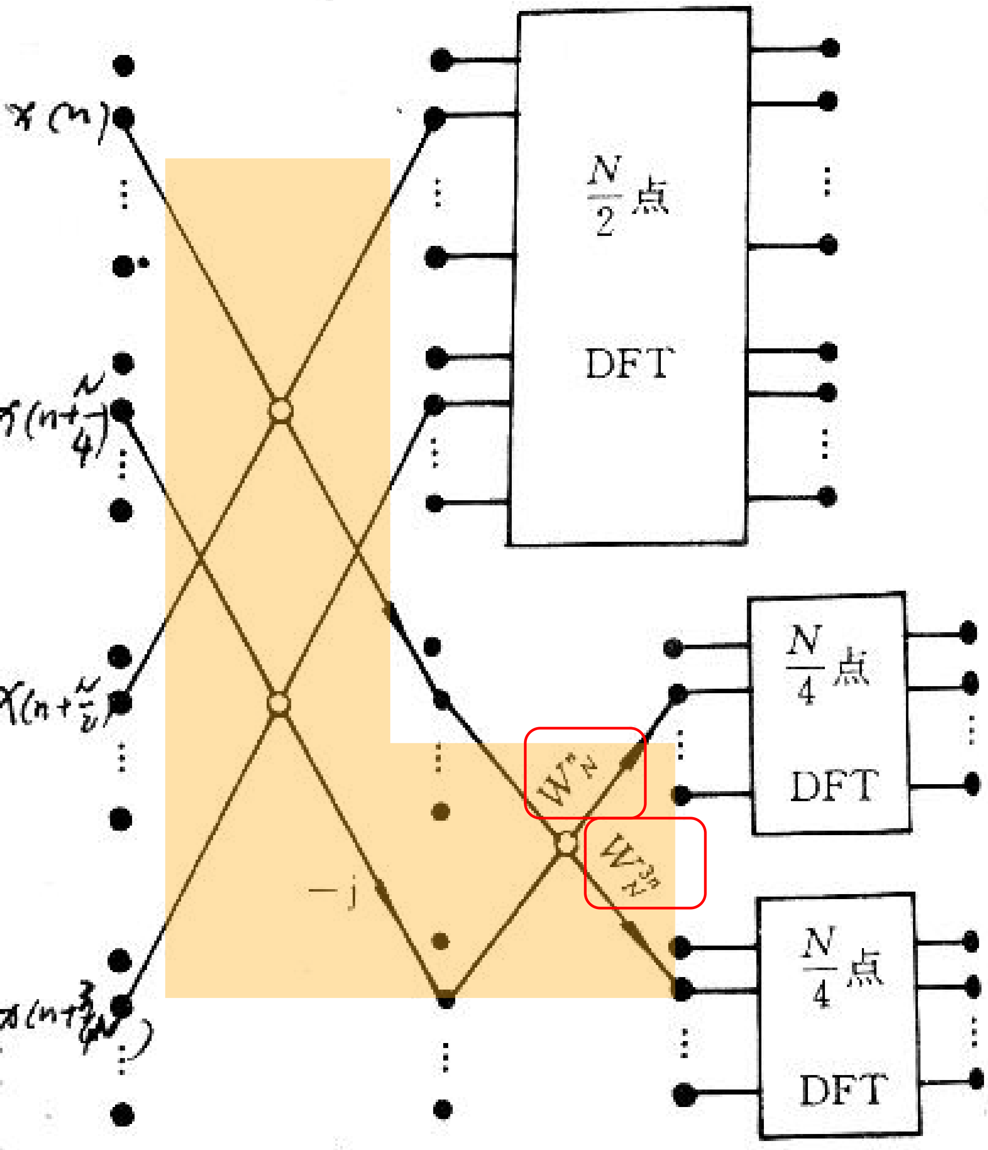


图 4—21　分裂基算法第一次分解 L 形流图

与基4比较W_N

$$X(2k)=\sum_{n=0}^{\frac{N}{2}-1}x_2(n)W_N^{2kn}$$

$$=DFT[x_2(n)]$$

$$X(4k+1)=\sum_{n=0}^{\frac{N}{4}-1}x_4^1(n)W_N^{4kn}$$

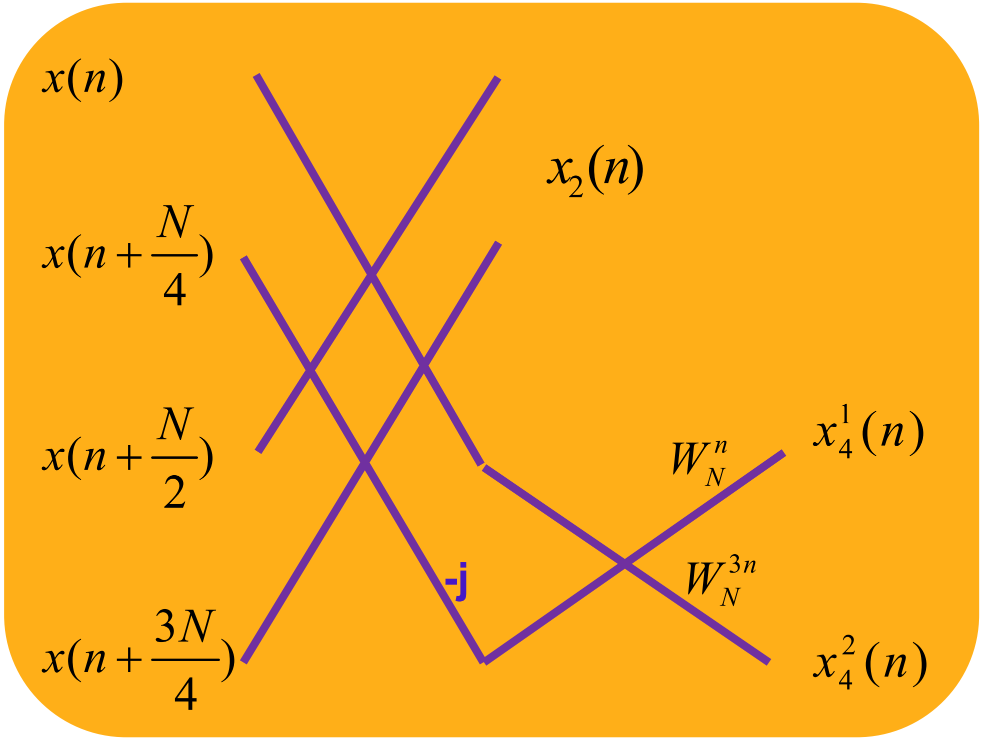
$$=DFT[x_4^1(n)]$$

$$X(4k+3)=\sum_{n=0}^{\frac{N}{4}-1}x_4^2(n)W_N^{4kn}$$

$$=DFT[x_4^2(n)]$$

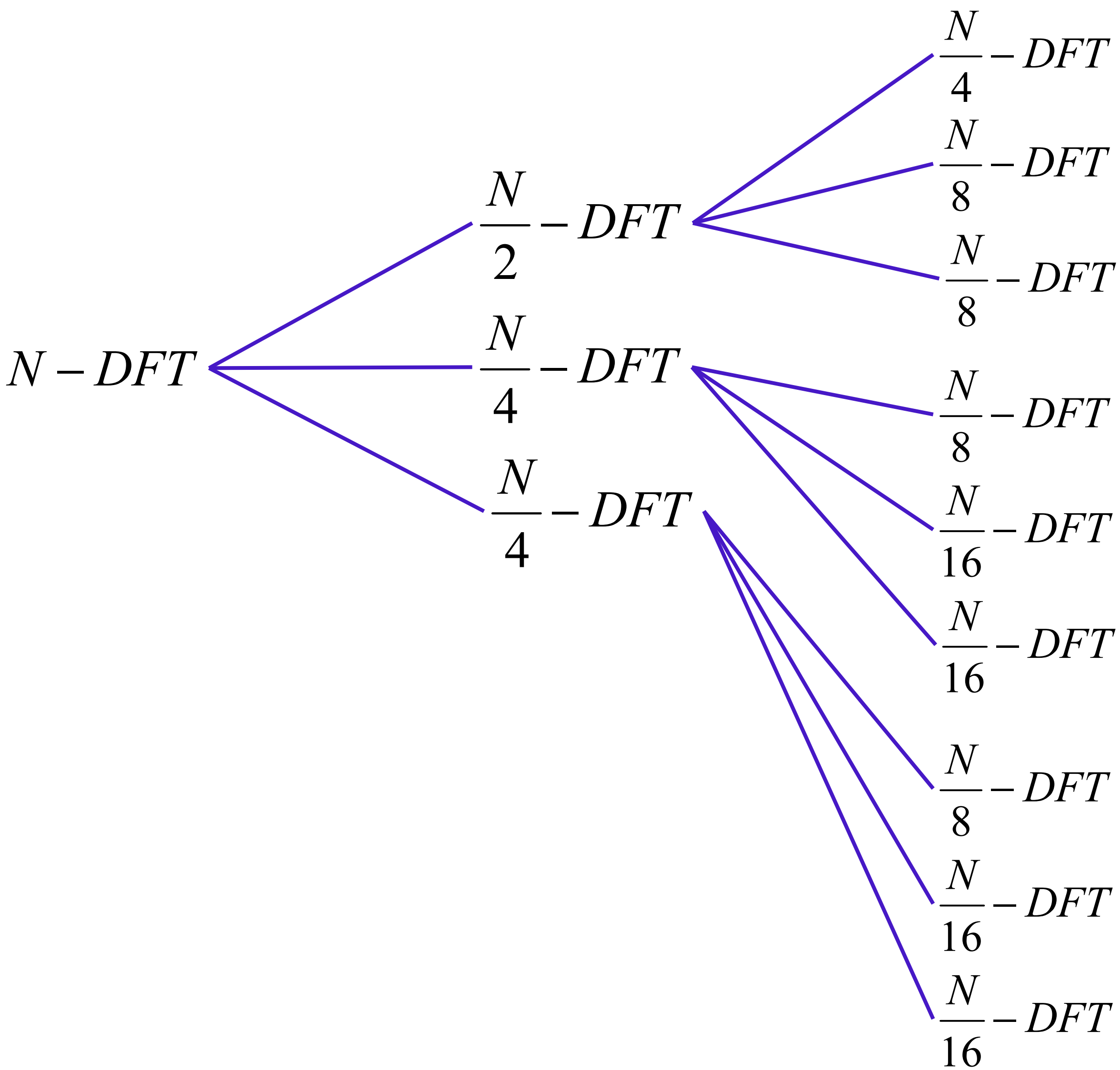
L形蝶形(流图)

图4-21/P.146



$$
\begin{cases}
x_2(n) \overset{\Delta}{=} x(n)+x(n+\frac{N}{2}), \quad 0\le n\le \frac{N}{2}-1 \\
x_4^1(n) \overset{\Delta}{=} \left[\left(x(n)-x(n+\frac{N}{2})\right)-j\left(x(n+\frac{N}{4})-x(n+\frac{3N}{4})\right)\right]W_N^n, 0\le n\le \frac{N}{4}-1 \\
x_4^2(n) \overset{\Delta}{=} \left[\left(x(n)-x(n+\frac{N}{2})\right)+j\left(x(n+\frac{N}{4})-x(n+\frac{3N}{4})\right)\right]W_N^{3n}, 0\le n\le \frac{N}{4}-1
\end{cases}
$$

可见：


$N-DFT$
$\frac{N}{2}-DFT$
$\frac{N}{4}-DFT$
$\frac{N}{4}-DFT$
$\frac{N}{4}-DFT$
$\frac{N}{8}-DFT$
$\frac{N}{8}-DFT$
$\frac{N}{8}-DFT$
$\frac{N}{16}-DFT$
$\frac{N}{16}-DFT$
$\frac{N}{8}-DFT$
$\frac{N}{16}-DFT$
$\frac{N}{16}-DFT$

例：N=16　　分裂基第一次分解L形流图：图4-22 P.147

分解1：$x_1(n) \rightarrow x_2(n) \qquad 0 \le n \le 7 \qquad (\frac{N}{2}-1)$

$x_4^1(n) \qquad 0 \le n \le 3 \qquad (\frac{N}{4}-1)$

$x_4^2(n) \qquad 0 \le n \le 3 \qquad (\frac{N}{4}-1)$

分解2：$x_2(n) \rightarrow y_2(n) \qquad 0 \le n \le 3 \qquad (\frac{N}{4}-1) = \frac{N/2}{2}-1$

$y_4^1(n) \qquad 0 \le n \le 1 \qquad (\frac{N}{8}-1) = \frac{N/2}{4}-1$

$y_4^2(n) \qquad 0 \le n \le 1 \qquad (\frac{N}{8}-1) = \frac{N/2}{4}-1$

分解3：

$$y_2(n) \to z_2(n) \qquad 0 \le n \le 1$$

$$z_4^1(n) \qquad n = 0$$

$$z_4^2(n) \qquad n = 0$$

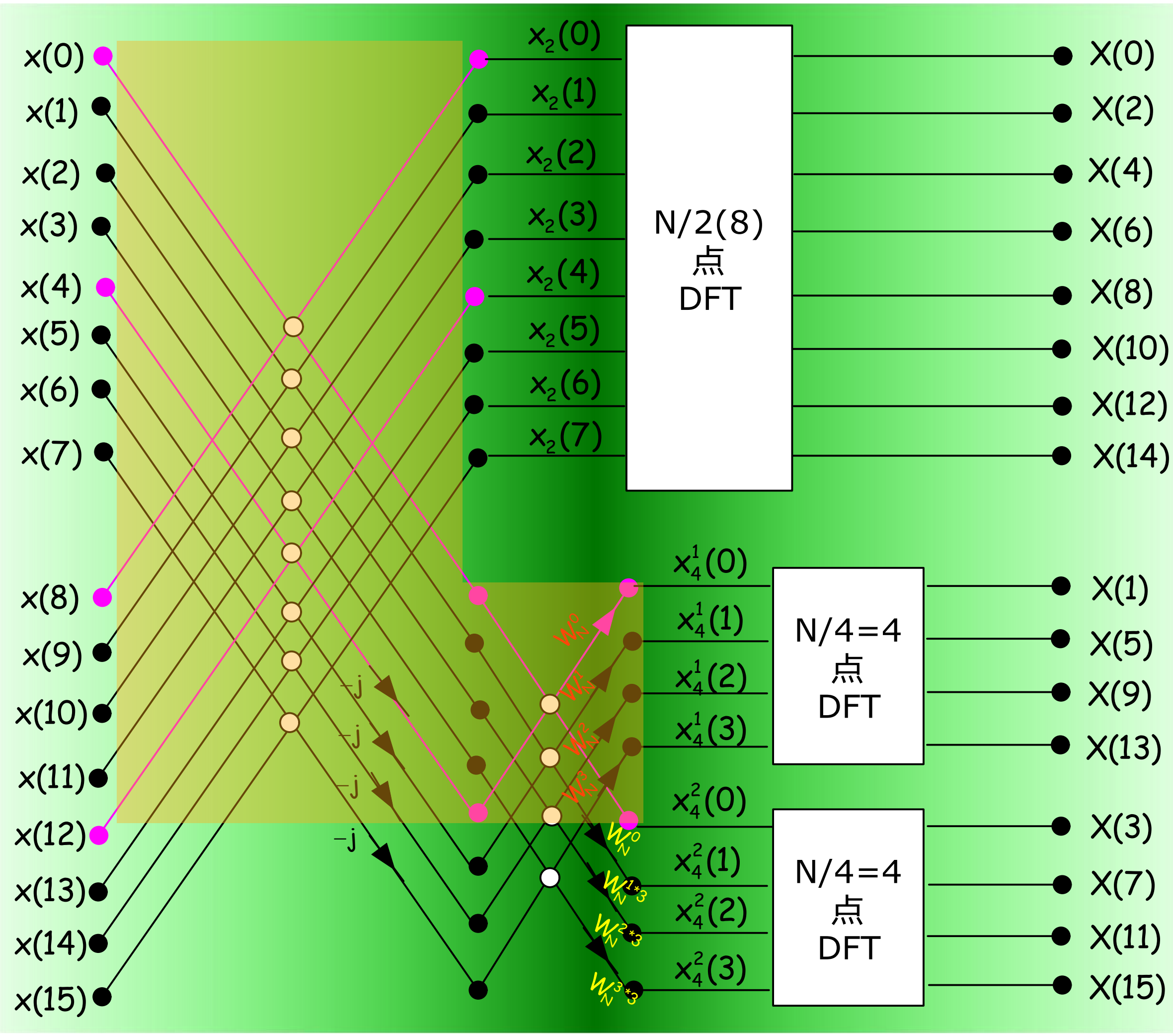
4点分裂基
L形运算流图
图4-24/P.148

$$x_4^1(n) \to \cdots\cdots$$

$$x_4^2(n) \to \cdots\cdots$$

图4-24 $\to$ 图4-23 $\to$ 图4-22 $\Rightarrow$ 图4-25 P.148

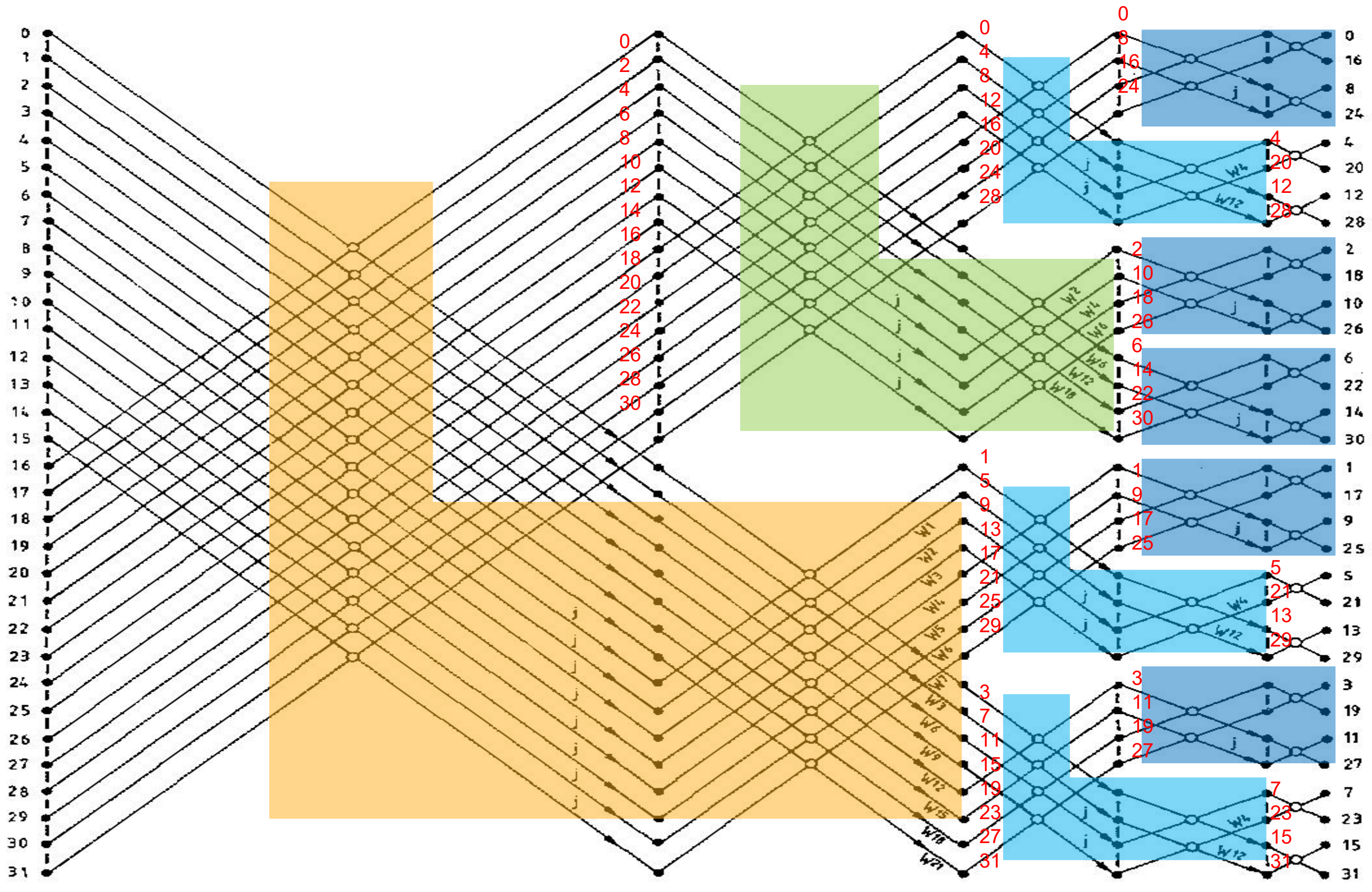
16点分裂基DIF-FFT算法流图



$X(2k)$ 长偶

$X(4k+1)$ 长奇之偶

$X(4k+3)$ 长奇之奇



蝶形个数：8 4 6 5

分裂基

按时间

Or 抽取 FFT算法流图？

按频率

N=16 分裂基 按时间抽取 FFT算法流图

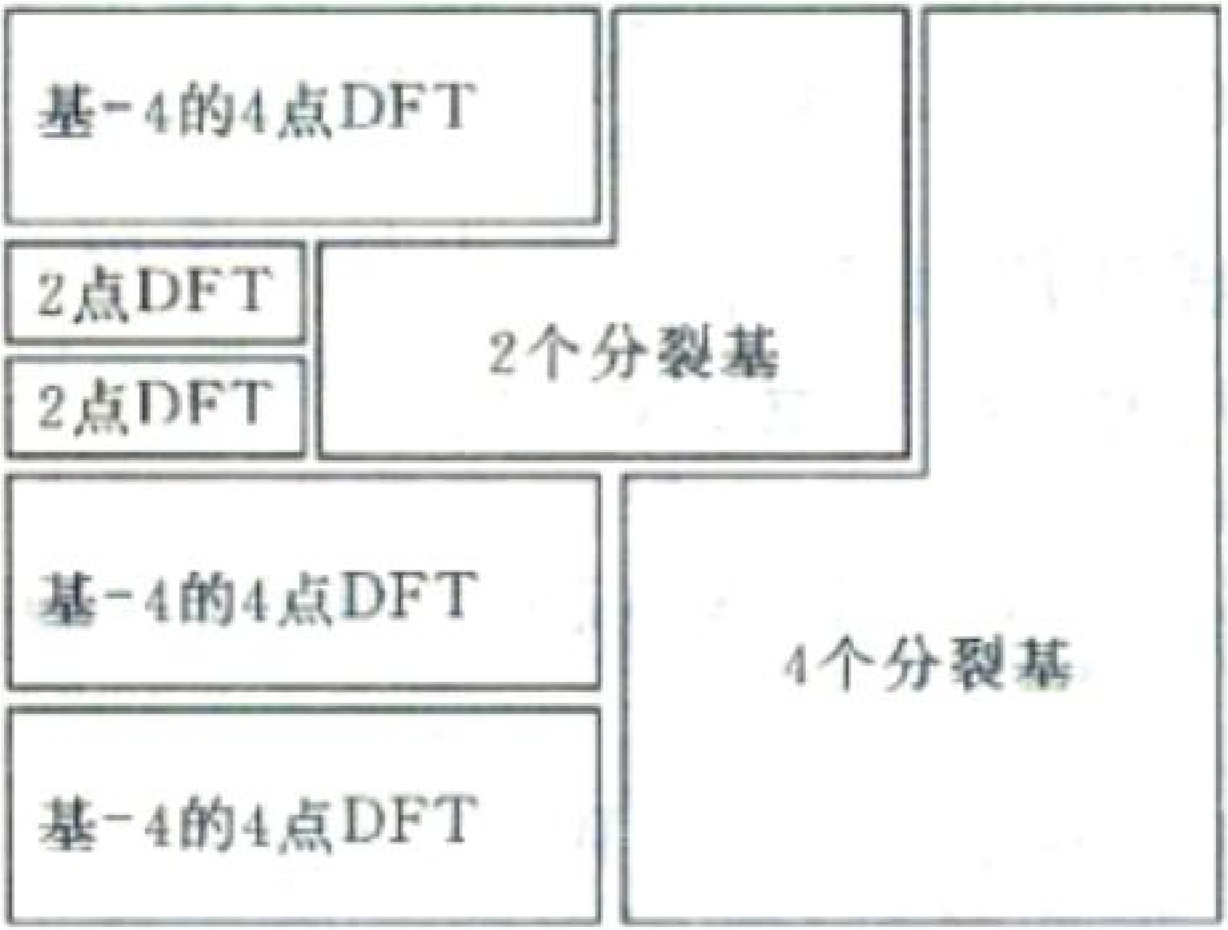


图 4-25 $N=2^4=16$ 点的分裂基 FFT 的示意图

N=16 分裂基 按时间抽取 FFT算法流图

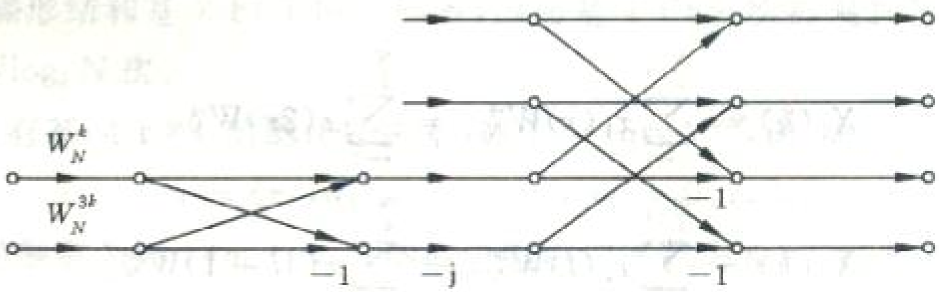


图 4-24　分裂基 FFT 算法的一个基本蝶形运算

图 4-25　$N=2^4=16$ 点的分裂基 FFT 的示意图

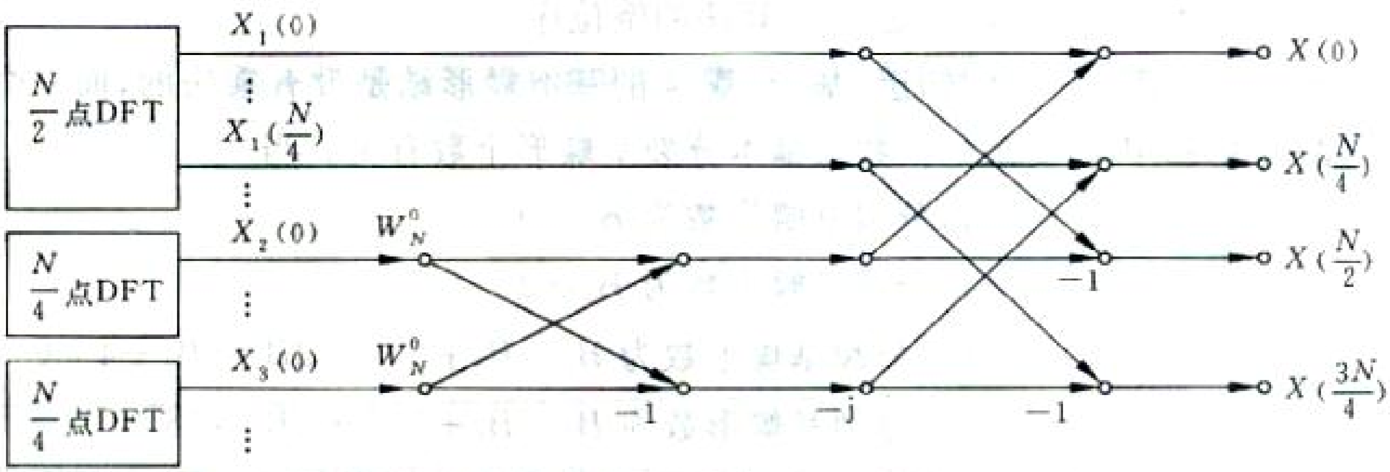


图 4-23　分裂基 FFT 算法(时间抽选)的第一级流图

N=16 分裂基 按时间抽取 FFT算法流图

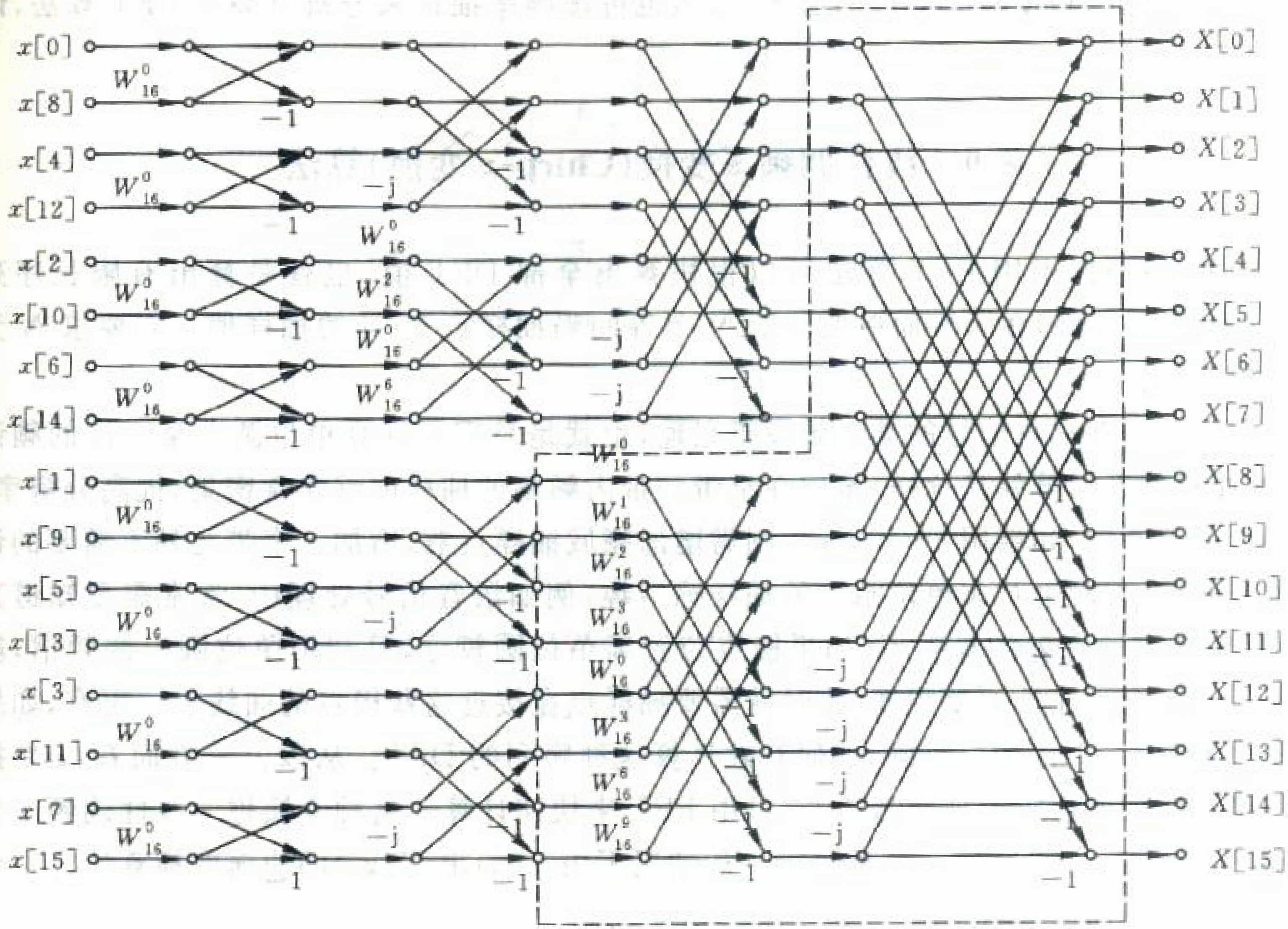


图 4-26　$N=2^4=16$ 分裂基 FFT 算法(按时间抽选)的流图

(输入二进制倒位序，输出正常顺序)

注：上图只用虚线框表示了一级的倒 L 结构

N=16 基-2 按时间抽取FFT流图

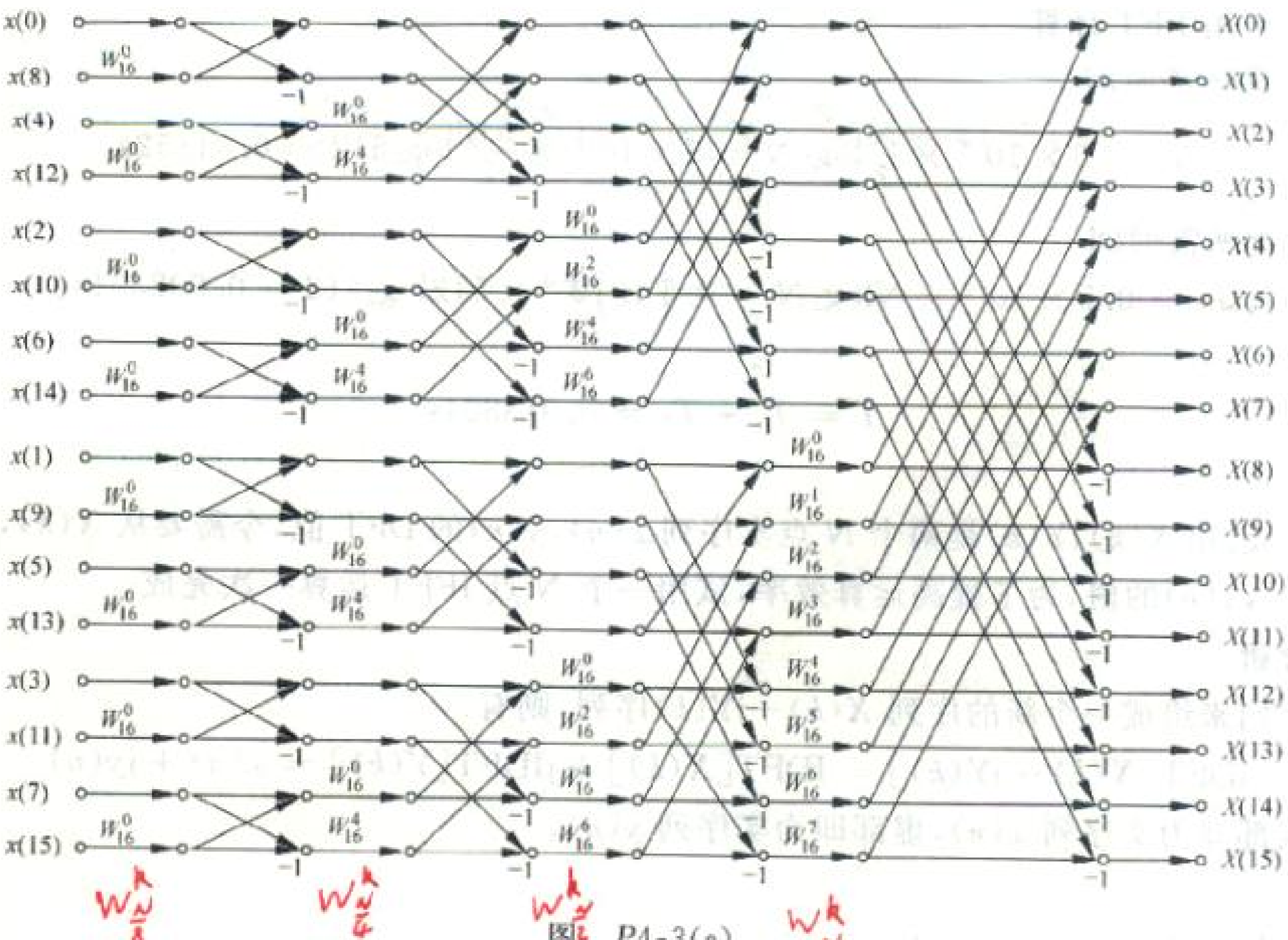


图 P4-3(a)

N=16 基-2 按频率抽取FFT流图

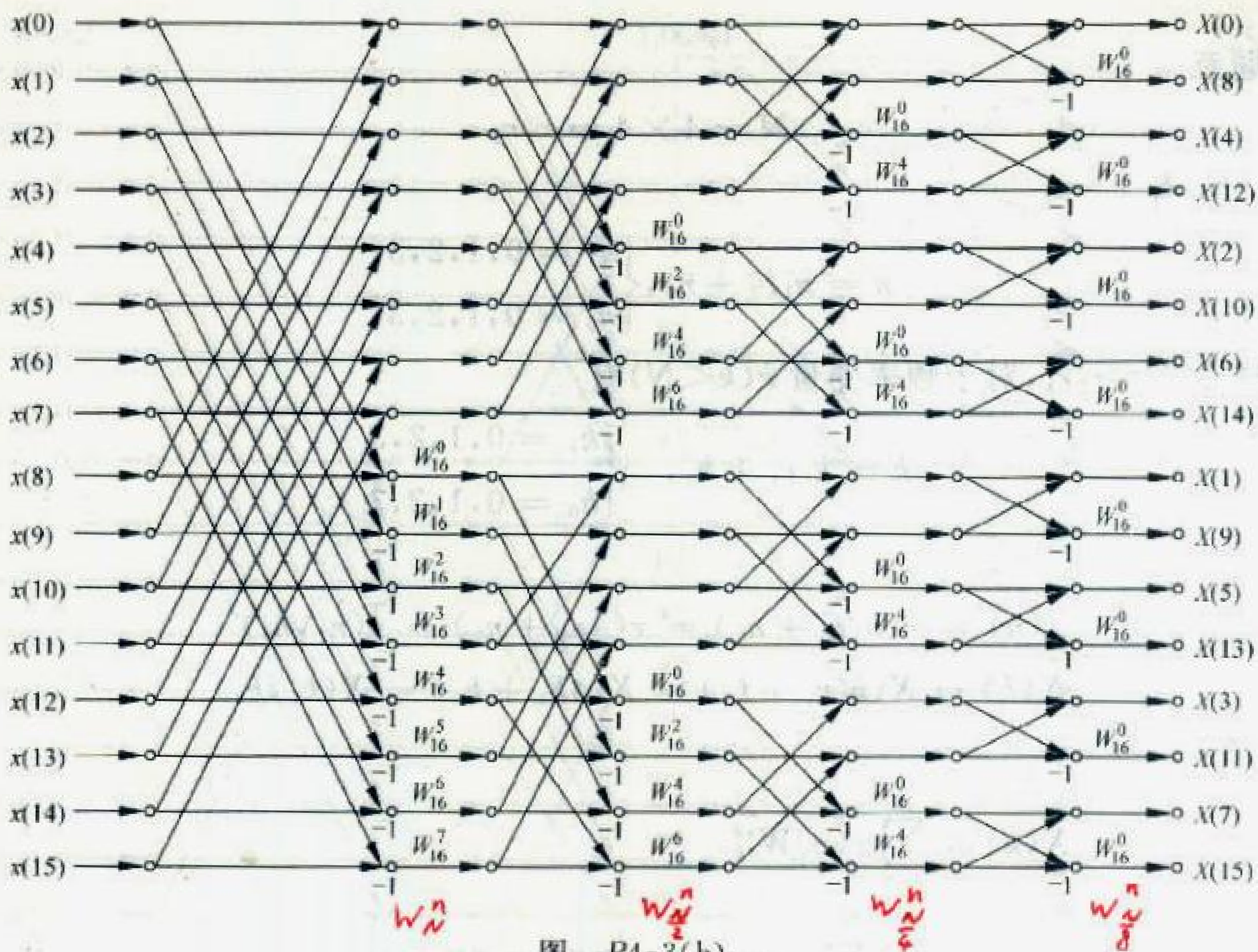


图 P4-3(b)

N=16 基-4 按频率抽取FFT流图

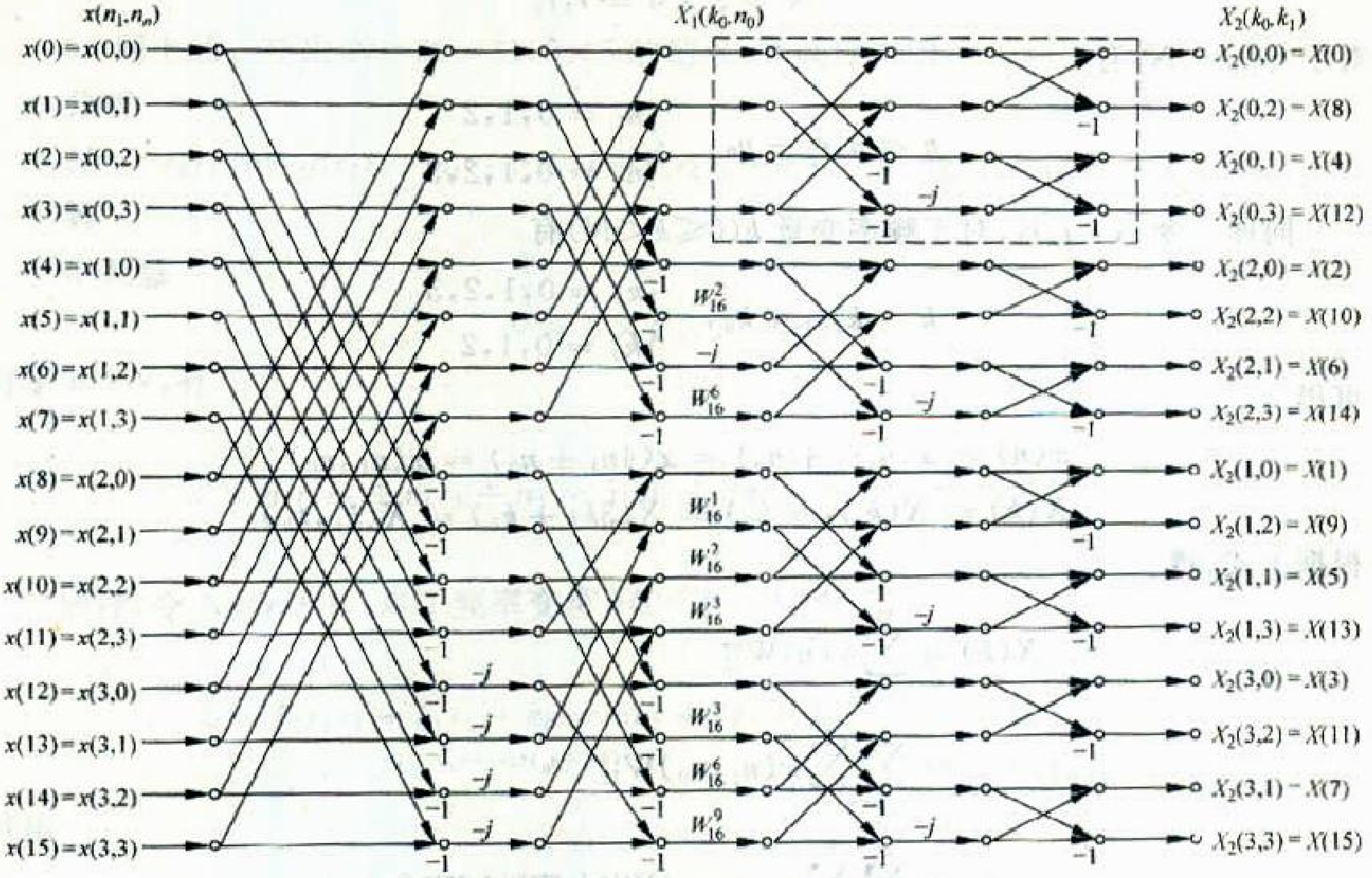


图 P4-4

蝶形数（有问题）			
分裂基：	64点/114	32点/23	16点/9
基4：	64点/144		16点/24
基2：	64点/192	32点/64	16点/32

三、运算量分析

L形分解：共M-1级　　　$N=2^M$

每级L形碟形个数：

$$l_1 = \frac{N}{4}$$

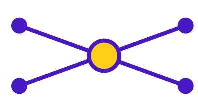
$$l_j = \frac{N}{4} - \frac{l_{j-1}}{2}, \qquad j = 2,3,\ldots,M-1$$

每个L形碟形：×— 2次

总的复数乘法次数：

$$C_M = 2\times\sum_{j=1}^{M-1} l_j = \frac{1}{3}N\log_2^N - \frac{2}{9}N + (-1)^M\frac{2}{9} \qquad \textbf{(4-48)}$$

相比 $\frac{N}{2}\log_2^N$，下降33% $=(\frac{1}{2}-\frac{1}{3})\Big/\frac{1}{2}\times 100\%$

$+ — N\log_2^N$ 相同（∵  个数相同） 理解

§4-7 实序列的FFT算法

一、问题的提出

$$\forall\, x(n) — DFT[x(n)] \rightarrow FFT$$

实数： $\forall\, x(n) = x^*(n), \qquad 0 \le n \le N-1$

$$DFT[x(n)] \longrightarrow FFT\ \ ?$$

可能的办法：

① $x(n) \rightarrow x(n) + j0 \rightarrow y(n) \rightarrow FFT$

② $x(n) \rightarrow DFT[x(n)] \rightarrow$ 专用算法/硬件

③ 能否有更好的方法吗？

二、算法一：用一个N-FFT同时计算两个N点实序列

$$\forall\ x_1(n), \quad 0 \le n \le N-1$$
$$x_2(n), \quad 0 \le n \le N-1$$

$$x(n) \stackrel{\Delta}{=} x_1(n) + jx_2(n)$$
$$\updownarrow \qquad \updownarrow \qquad \updownarrow$$
$$X(k) = X_1(k) + jX_2(k),$$

$X_1(k)$, $X_2(k)$都是复数序列（Matlab）

DFT的性质：[P.91]

$$x(n) = x_r(n) + jx_i(n)$$
$$\updownarrow \qquad \updownarrow \qquad \updownarrow$$
$$X(K) = X_{ep}(k) + X_{op}(k)$$

周期共轭对称分量　　周期共轭反对称分量

$$x_r(n) \stackrel{\Delta}{=} \frac{1}{2}[x(n) + x^*(n)]$$

$$jx_l(n) \stackrel{\Delta}{=} \frac{1}{2}[x(n) - x^*(n)]$$

即：

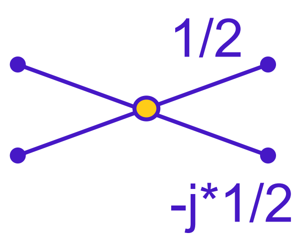
$$X_{ep}(k) \overset{\Delta}{=} DFT[x_r(n)] \quad = \frac{1}{2}[X(k) + X^*(N-k)]$$

$$X_{op}(k) \overset{\Delta}{=} DFT[jx_i(n)] \quad = \frac{1}{2}[X(k) - X^*(N-k)]$$

$$\therefore X_1(k) = X_{ep}(k) = \frac{1}{2}[X(k) + X^*(N-k)] \qquad \textbf{(4-51)}$$

$$X_2(k) = -jX_{op}(k) = -\frac{j}{2}[X(k) - X^*(N-k)] \qquad \textbf{(4-52)}$$

$$\begin{matrix} x_1(n) \\ x_2(n) \end{matrix} \longrightarrow x(n) \xrightarrow{FFT} \begin{matrix} X(k) \\ X^*(N-k) \end{matrix}$$



$$\begin{matrix} X_1(k) \\ X_2(k) \end{matrix} \quad k = 0,1,\ldots,N$$

三、算法二：用一个N-FFT计算一个2N点实序列

$$\forall\, x(n)=x^*(n), \qquad 0\le n\le 2N-1$$

令：

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

$$\downarrow$$

$$y(n)=x_1(n)+jx_2(n) \qquad \begin{cases} X_1(k)=Y_{ep}(k) & =\frac{1}{2}[Y(k)+Y^*(N-k)] \\ X_2(k)=-jY_{op}(k) & =-j\frac{1}{2}[Y(k)-Y^*(N-k)] \end{cases}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$Y(k)=X_1(k)+jX_2(k) \qquad \begin{cases} X(k)=X_1(k)+W_{2N}^kX_2(k) \\ X(N+k)=X_1(k)-W_{2N}^kX_2(k) \end{cases}$$

$$0\le k\le N-1$$

见P127 图4-2

$$x(n)\to \begin{matrix} x_1(n)\\ x_2(n)\end{matrix} \to y(n) \xrightarrow{FFT} \begin{matrix} Y(k)\\ Y^*(N-k)\end{matrix}$$