数字信号处理

周治国

2023.10

第四章 快速傅里叶变换

§4-3 按时间抽取(DIT)的FFT算法
(Cooley-Tukey算法)

一、算法原理

$\forall\, x(n), \quad 0 \le n \le N-1,\ N = 2^{\nu}$ (若$N \ne 2^{\nu}$，可通过补零达到)

$\downarrow$

$FFT \to$ 基-2 $\quad FFT$ / 即N为2的整数幂的FFT

由FFT的定义:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad k = 0,1,\ldots,N-1 \quad (4\text{-}4)$$

$\frac{N}{2} - x_1(n) \qquad DFT$

$N - x(n) \quad DFT \quad \xleftarrow{?}$

$\frac{N}{2} - x_2(n) \qquad DFT$

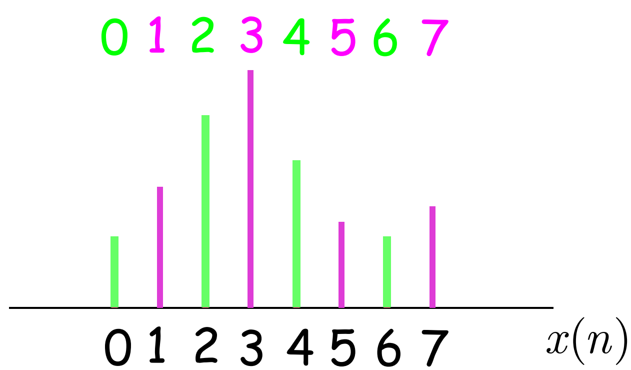
§ 4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

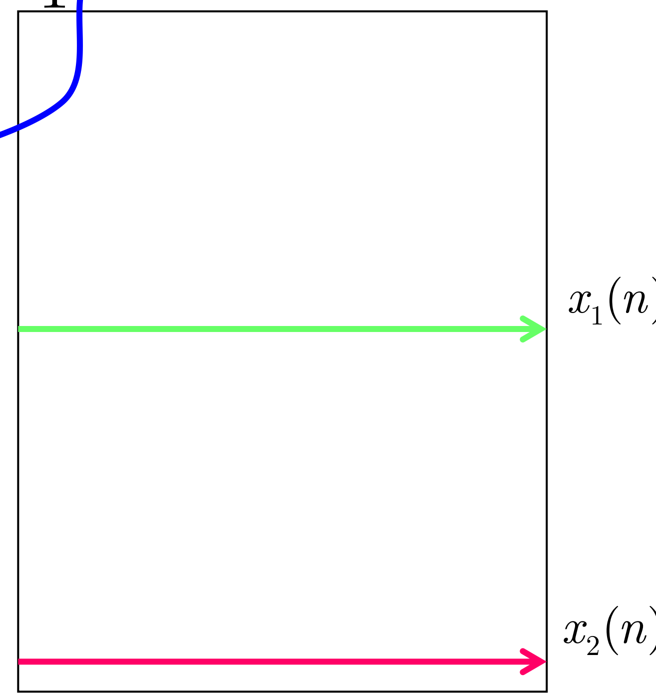
设序列点数 $N=2^L$, L 为整数 (若不满足，则补零)

将序列 $x(n)$ 按 n 的奇偶分成两组：

$$x\left(2r\right) = x_1(r)$$

$$x\left(2r+1\right) = x_2(r) : r = 0, 1, \ldots, \tfrac{N}{2} - 1$$





§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

令 $x_1(n) \triangleq x(2r) \qquad r = 0,1,\ldots,\frac{N}{2}-1$

$$x_2(n) \triangleq x(2r+1) \qquad r = 0,1,\ldots,\frac{N}{2}-1 \qquad (4\text{-}5)$$

代入(4-4)式

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

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

$$= X_1(k) + W_N^k X_2(k) \qquad (4\text{-}7)$$

式中： $\boxed{X(k), 0 \le k \le N-1}$

$X_1(k) = DFT[x_1(n)], 0 \le k \le \frac{N}{2}-1$

$X_2(k) = DFT[x_2(n)], 0 \le k \le \frac{N}{2}-1$

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

可见：

$$N-DFT \begin{array}{l} \nearrow \frac{N}{2}-DFT \searrow \\ \searrow \frac{N}{2}-DFT \nearrow \end{array} \xrightarrow[?]{} N-DFT$$

由(4-7)式

$$\begin{array}{l} X_1(k) \searrow \\ X_2(k) \nearrow \end{array} \longrightarrow X(k), \quad 0 \le k \le \frac{N}{2}-1$$

$$0 \le k \le \frac{N}{2}-1$$

问题：$\frac{N}{2} \le k \le N-1$时，$X(k)=?$

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

利用 $W_{\frac{N}{2}}^{rk}$的周期性， $W_{\frac{N}{2}}^{rk} = W_{\frac{N}{2}}^{r(\frac{N}{2}+k)}$

$$
\begin{aligned}
X_1(k+\frac{N}{2}) &= \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{\frac{N}{2}}^{r(\frac{N}{2}+k)} \\
&= \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{\frac{N}{2}}^{rk} \\
&= X_1(k), \qquad 0 \le k \le \frac{N}{2}-1 \qquad (4\text{-}10)
\end{aligned}
$$

同理有，

$$X_2(k+\frac{N}{2}) = X_2(k), \qquad 0 \le k \le \frac{N}{2}-1 \qquad (4\text{-}11)$$

[可见$X_1(k)$和$X_2(k)$的后半部分完全重复了各自的前半部分]

代入(4-7)式，有:

§ 4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

$$X(\frac{N}{2}+k)=X_1(k)+W_N^{\frac{N}{2}+k}X_2(k)$$

$$\downarrow \quad \because W_N^{\frac{N}{2}}=e^{j\pi}=-1 \qquad 0\le k\le \frac{N}{2}-1$$

$$=X_1(k)-W_N^kX_2(k) \qquad 0\le k\le \frac{N}{2}-1$$

归纳起来有

$$X(k)=X_1(k)+W_N^kX_2(k) \qquad k=0,1,\ldots,\frac{N}{2}-1 \qquad (4\text{-}13)$$

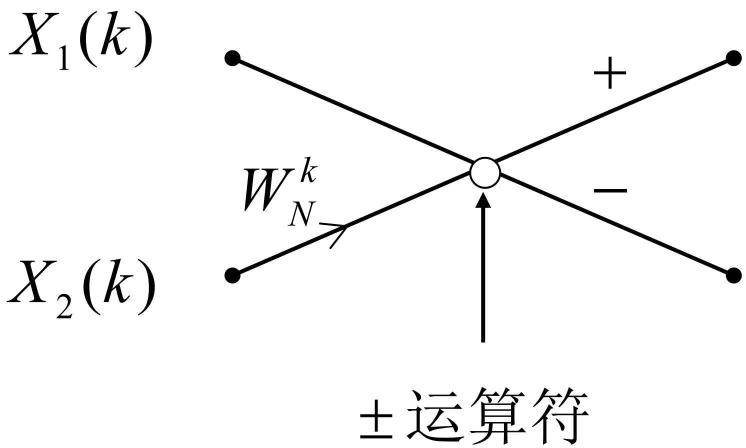
$$X(\frac{N}{2}+k)=X_1(k)-W_N^kX_2(k) \qquad k=0,1,\ldots,\frac{N}{2}-1 \qquad (4\text{-}14)$$

可见，

$$N-DFT \nearrow \frac{N}{2}-DFT \searrow, \quad N-DFT \searrow \frac{N}{2}-DFT \nearrow \;\longrightarrow N-DFT$$

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

上述运算可用下列蝶形信号流图表示:



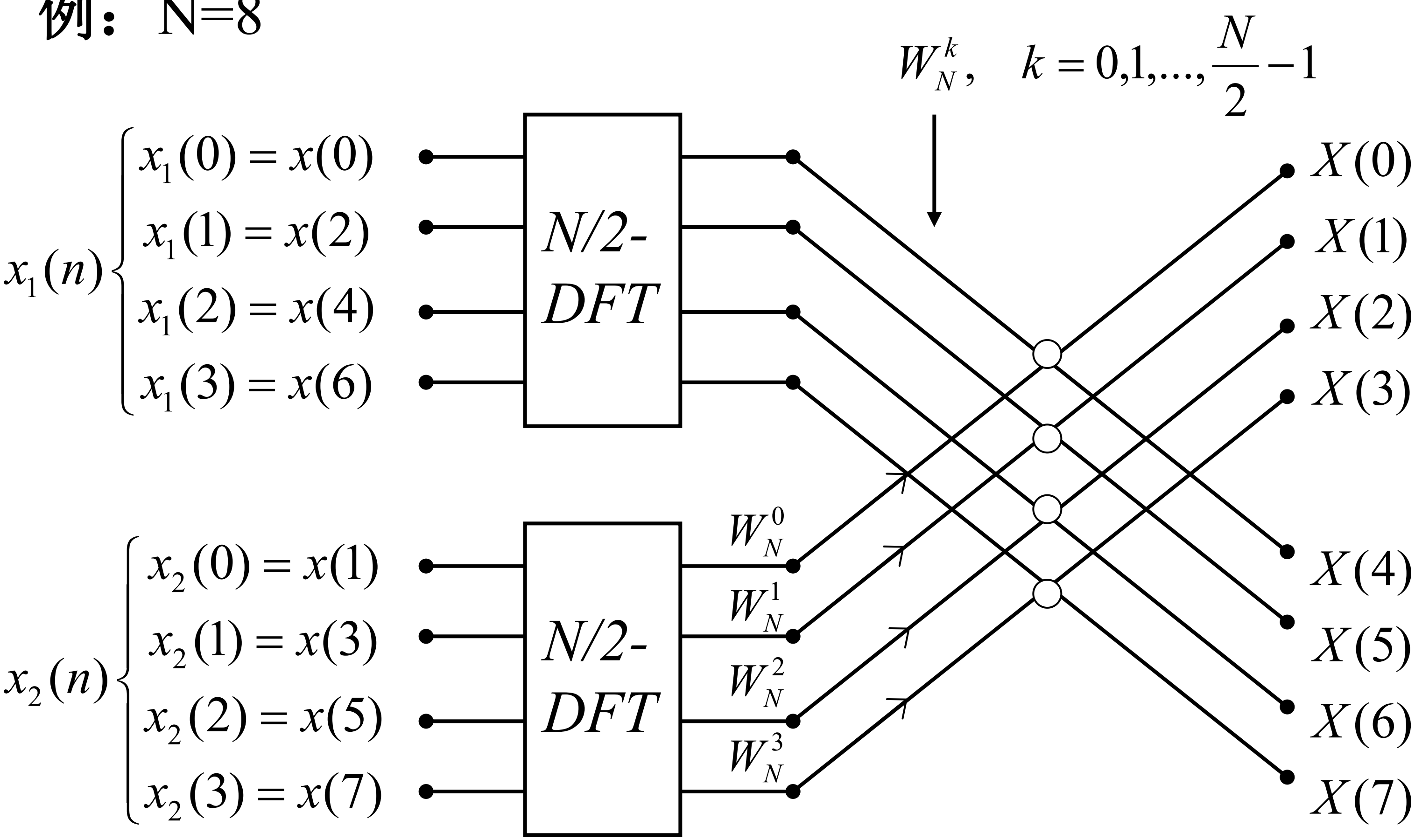
$$X(k) = X_1(k) + W_N^k X_2(k)$$

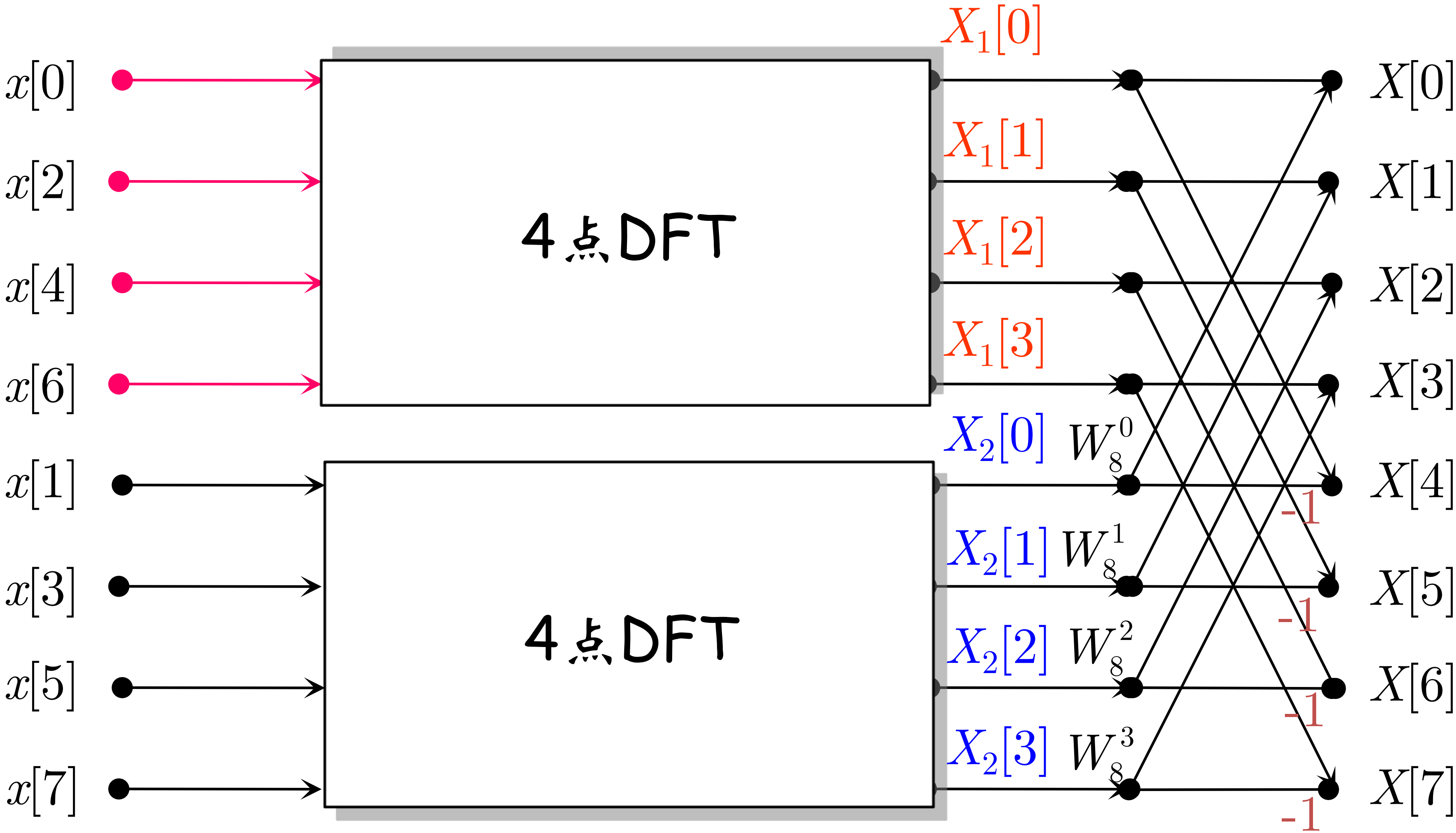
$$X(\frac{N}{2}+k) = X_1(k) - W_N^k X_2(k)$$

图 4-1 蝶形运算流图符号

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

例：N=8





$$X[k] = X_1[k] + W_8^k X_2[k],\ k = 0,1,2,3$$

$$X[k+8/2] = X_1[k] - W_8^k X_2[k],\ k = 0,1,2,3$$

**8点基2时间抽取FFT算法流图

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

计算量分析：

$$*:\ 2\times\left(\frac{N}{2}\right)^2+\frac{N}{2}\doteq\frac{N^2}{2}$$

$$+:\ 2\times\left(\frac{N}{2}\right)\left(\frac{N}{2}-1\right)+N\doteq\frac{N^2}{2}$$

相比N-DFT的

$$*:\ N^2$$

$$+:\ N(N-1)$$

运算量减小了一半。

$$\because N=2^{\nu}$$

$$\therefore \frac{N}{2}=2^{\nu-1}=2\times 2^{\nu-2}$$

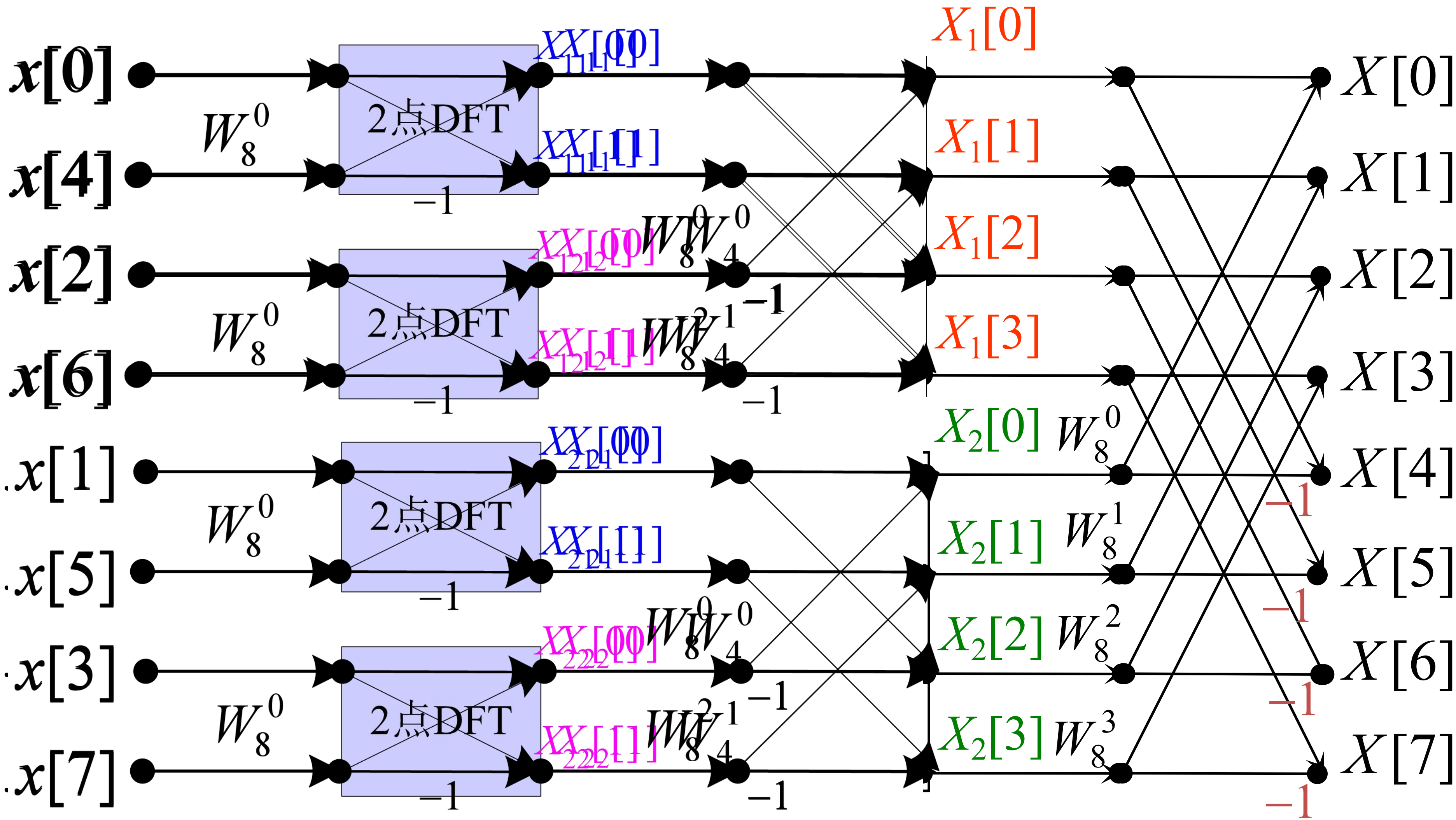
$$\frac{N}{2^{\nu-1}}=2$$

$\frac{N}{2}-DFT$ → $\frac{N}{4}-DFT$ → …… → $2-DFT$, $2-DFT$

$\frac{N}{2}-DFT$ → $\frac{N}{4}-DFT$ → …… → $2-DFT$, $2-DFT$

⋮

$2-DFT$, $2-DFT$



**8点基2时间抽取FFT算法流图

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

例：N=8 (P129)

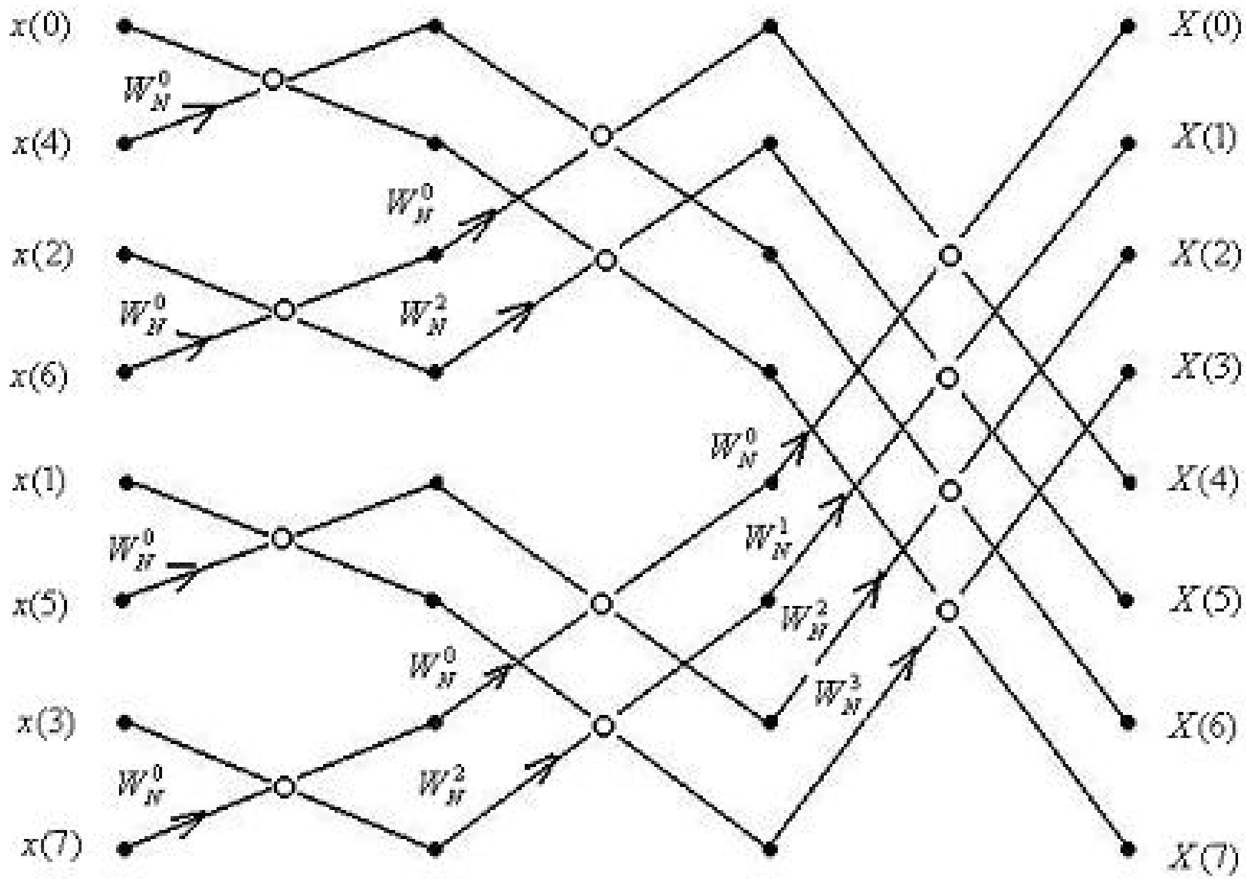


图4-5 N=8时的按时间抽取FFT运算流图

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

例：$N=2^v$ (P130)

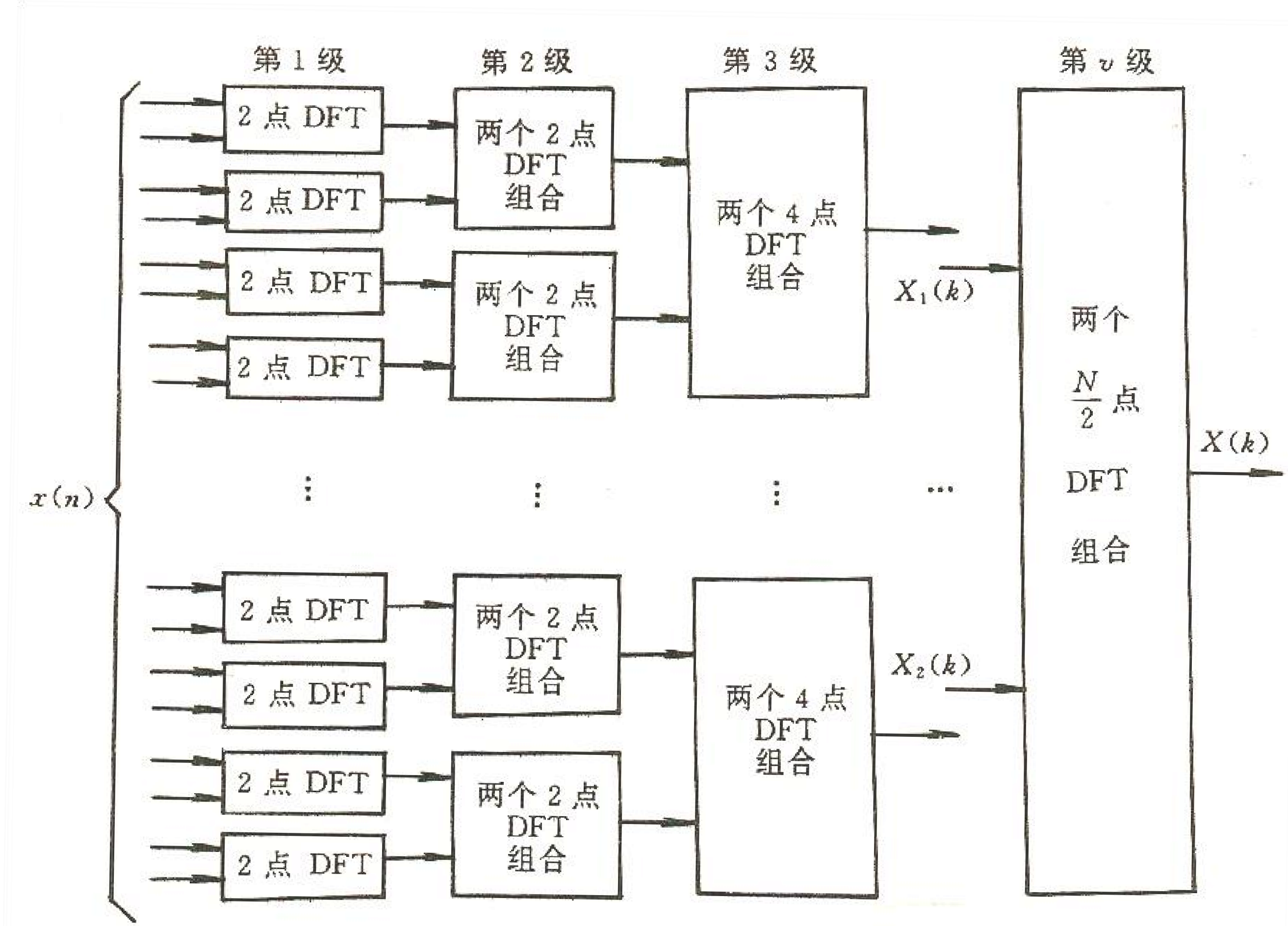


图4-6 N点基-2FFT的v级迭代过程

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

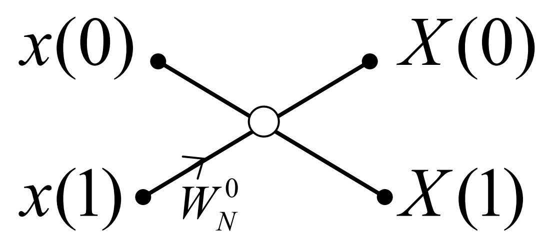
2-DFT：

$$X(k)=\sum_{n=0}^{1}x(n)W_2^{kn}$$

$$=x(0)W_2^0+x(1)W_2^k \qquad k=0,1$$

$$X(0)=x(0)+x(1)=x(0)+W_N^0x(1)$$

$$X(1)=x(0)-x(1)=x(0)-W_N^0x(1)$$



可见仅需计算“+/-”运算。

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

二、运算量比较

1.DIT-FFT：N=2$^{\nu}$

由图4-6可见，

$N-DFT \rightarrow \nu$ 级分解/蝶形运算

每一级：均有$\dfrac{N}{2}$蝶形运算 $\times$— $\dfrac{N}{2}$

$+$— N

$$\left.\begin{aligned}\text{所有}\nu\text{ 级：}\times\text{—}\frac{N}{2}\times\nu=\frac{N}{2}\log_2^N \\ +\text{—}N\times\nu=N\log_2^N\end{aligned}\right\}$$

§ 4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

二、运算量比较

2. DFT

$$\left.\begin{array}{l} \times \quad — \quad N^2 \\ + — N(N-1) \end{array}\right\} \sim N^2$$

3. DIT-FFT的运算效率

$$\frac{N^2}{\frac{N}{2}\log_2^N} = \frac{2N}{\log_2^N} \rightarrow \text{表}\,4-1/\text{P.131} \quad \text{图}4\text{-}7$$

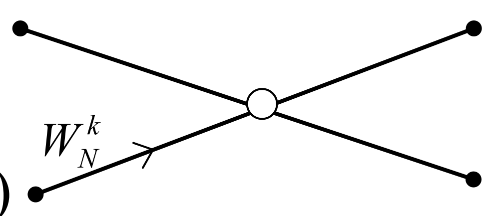
§ 4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

三、DIT-FFT算法的特点

1. 原位运算(In-place)

$A_{m-1}(i)$

$A_{m-1}(j)$



$$A_m(i) = A_{m-1}(i) + W_N^k A_{m-1}(j)$$

$$A_m(j) = A_{m-1}(i) - W_N^k A_{m-1}(j)$$

m — 第m列迭代

i, j — 数据所在的行数

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

2. 输入序列的序号及整序规律

由图4-5可见，

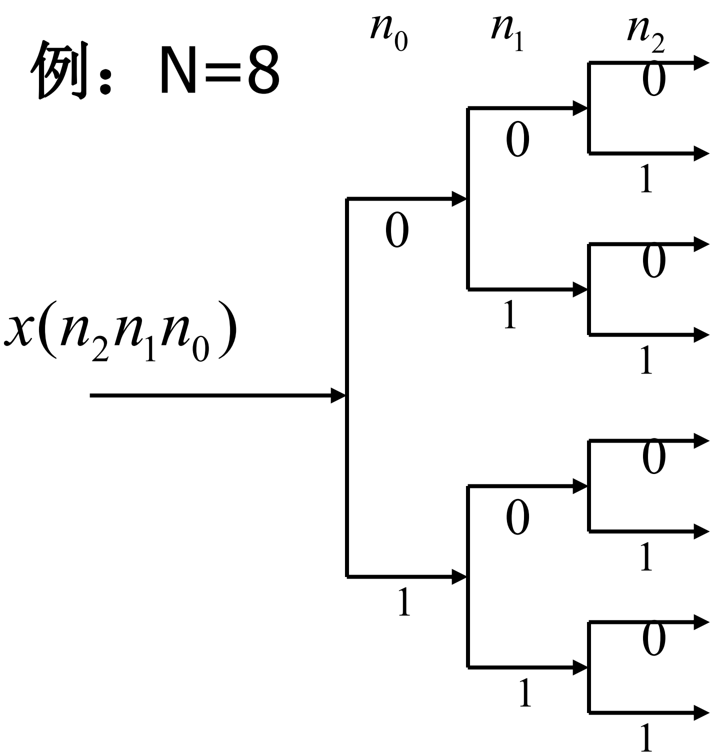
输入$x(n)$：乱序的 $\longrightarrow$ 如何做到？ $\longrightarrow$ 整序

输出$X(k)$：顺序的

§ 4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

乱序的原因

例：N=8



	$x(n)$
$x(000)=x(0)$	$x(000)$
$x(100)=x(4)$	$x(001)$
$x(010)=x(2)$	$x(010)$
$x(110)=x(6)$	$x(011)$
$x(001)=x(1)$	$x(100)$
$x(101)=x(5)$	$x(101)$
$x(011)=x(3)$	$x(110)$
$x(111)=x(7)$	$x(111)$

正好为 DIT-FFT 的输入

↓ 比较

正序 $x(n_2n_1n_0) \rightarrow x(n_0n_1n_2)$ 反序

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

输入序列的序号及整序规律

顺　序		倒　序	
十进制数 I	二进制数	二进制数	十进制数 J
0	0 0 0	0 0 0	0
1	0 0 1	1 0 0	4
2	0 1 0	0 1 0	2
3	0 1 1	1 1 0	6
4	1 0 0	0 0 1	1
5	1 0 1	1 0 1	5
6	1 1 0	0 1 1	3
7	1 1 1	1 1 1	7

$$x(n_2 n_1 n_0) \rightarrow x(n_0 n_1 n_2)$$

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

3. W_N^k的变化规律

间1，间2，间4

§4-3 按时间抽取(DIT)的FFT算法 (Cooley-Tukey算法)

四、DIT-FFT算法的若干变体

[详见P.134-135:图4-11~图4-14]

变换原则

往年真题：

试导出按时间抽取基-2 FFT算法的蝶形运算公式，并画出相应的**N=16**时的算法流图，并说明算法的特点。（要求输入反序，输出正序，原位运算）

N=16 基-2 按时间抽取FFT流图

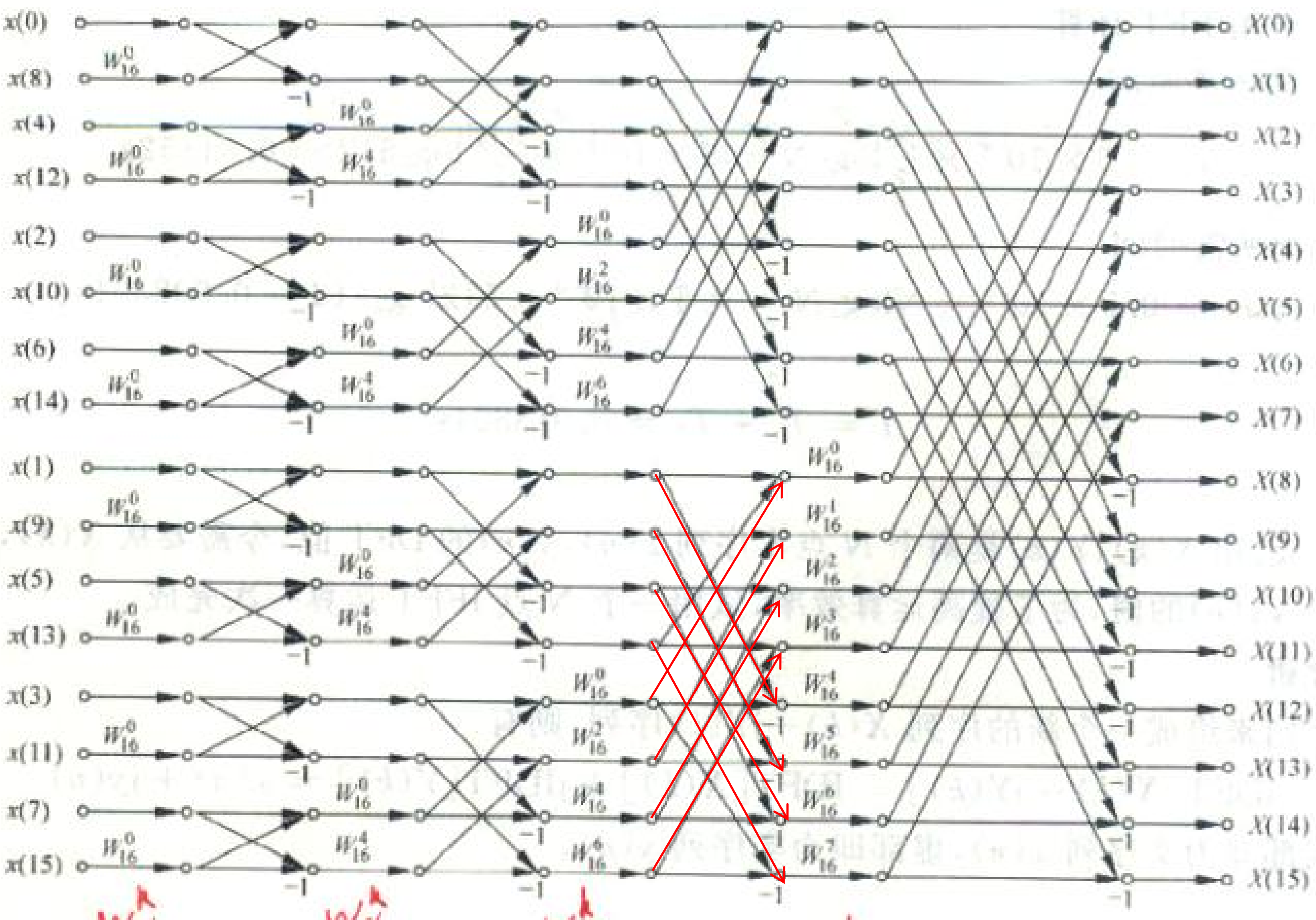


图 P4-3(a)