第九章：离散傅里叶变换的计算

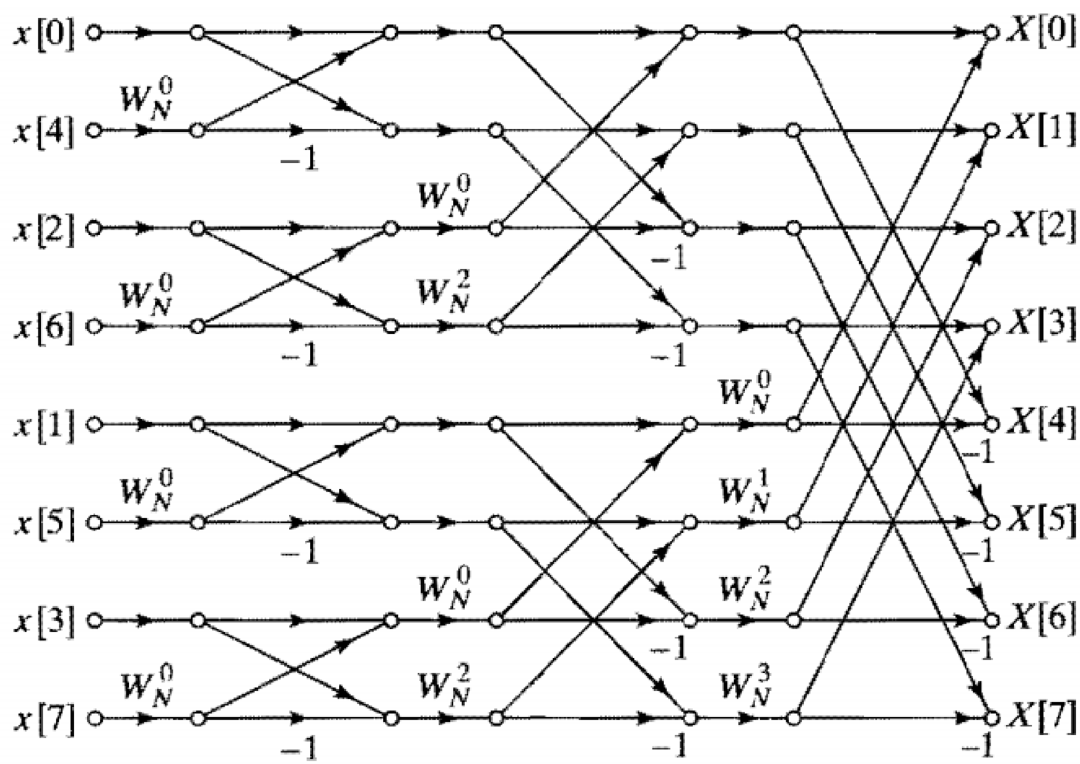
9.3.1 同址计算

◆DFT计算中的数据存储

DFT实现流图同时给出

1） DFT实现计算方法

2） DFT的初始数据和中间结果的存储方法



初始数据和中间结果的两种存储方法

① 直接存储

- 需采用2列存储寄存器
- 一列存储待计算数据，另一列存储计算结果

② 同址存储

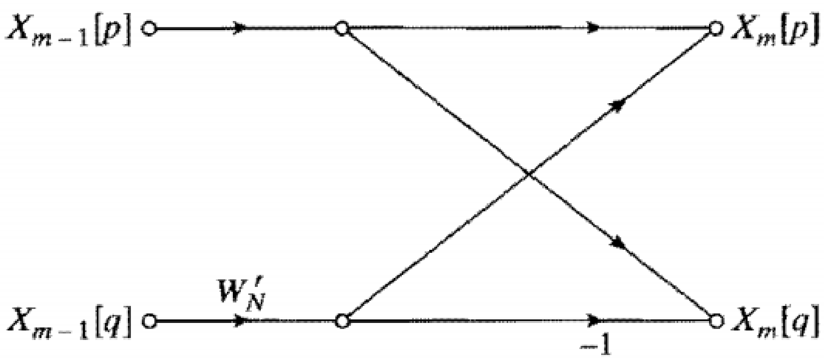
- 仅需一列存储寄存器，同时完成待计算数据和计算结果的存储

9.3.1 同址计算

◆同址存储与同址计算

令第m级计算的输入和输出序列分别为 $X_{m-1}[l]$ 和 $X_m[l]$，其中

$l=0,1,\cdots,N-1;\quad m=1,2,\cdots,v$



第1级计算的输入为DFT输入样本序列，记为 $X_0[l]$

因此，第m级蝶形计算可表示为

$$X_m[p]=X_{m-1}[p]+W_N^r X_{m-1}[q]$$
$$X_m[q]=X_{m-1}[p]-W_N^r X_{m-1}[q]$$

其中 p，q，r 随级数 m 的不同而改变

对于每个碟形计算，要计算第 m 列的 p 和 q 位置上的节点值，只需要第 $m-1$ 列在 p 和 q 位置上的节点（输出）值

若将$X_m[p]$和 $X_m[q]$ 分别存放在原存储 $X_{m-1}[p]$和 $X_{m-1}[q]$ 的相同地址的寄存器中(同址存储)，则DFT计算只需一列N个复数寄存器

9.3.1 同址计算

◆同址计算中序列的存储与读取

为实现同址计算，输入序列不能按原来的先后顺序储存或读取

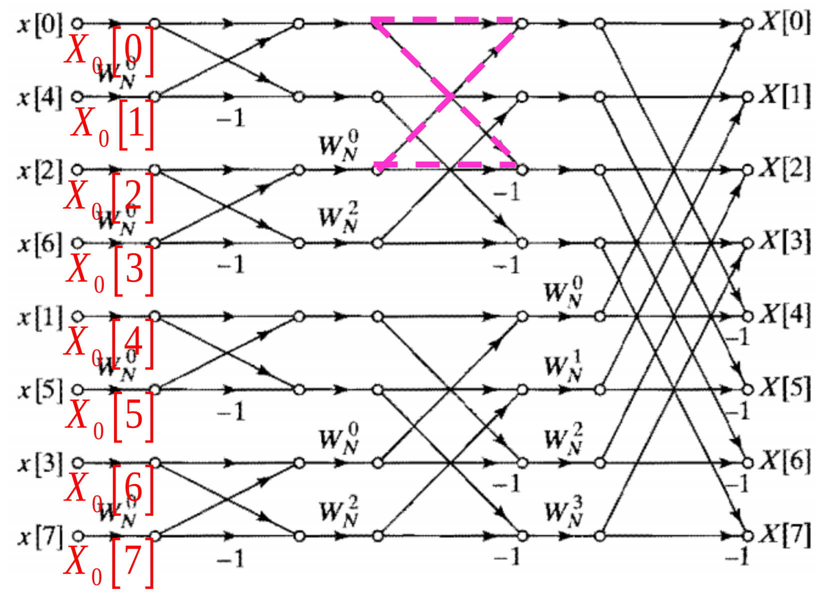
例：N=8的时间抽取DFT，输入序列存储和读取顺序（位序）

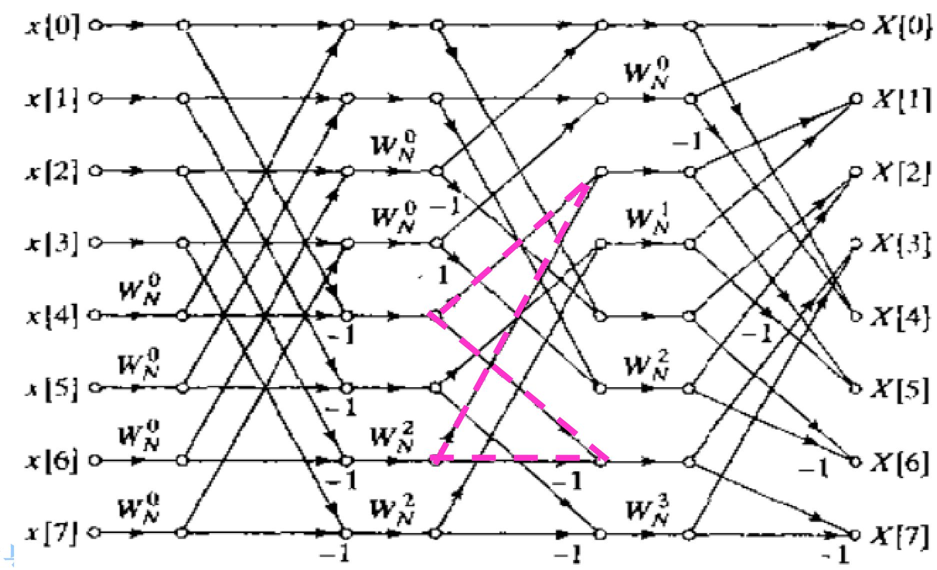
$X_0[0]=x[0]$ $\quad$ $X_0[4]=x[1]$

$X_0[1]=x[4]$ $\quad$ $X_0[5]=x[5]$

$X_0[2]=x[2]$ $\quad$ $X_0[6]=x[3]$

$X_0[3]=x[6]$ $\quad$ $X_0[7]=x[7]$





为实现同址计算，输入序列的存储或读取顺序必须使得每个蝶形计算的输入和输出端保持水平位置（存储位置相同）。

9.3.1 同址计算

◆ 倒位序（表示）

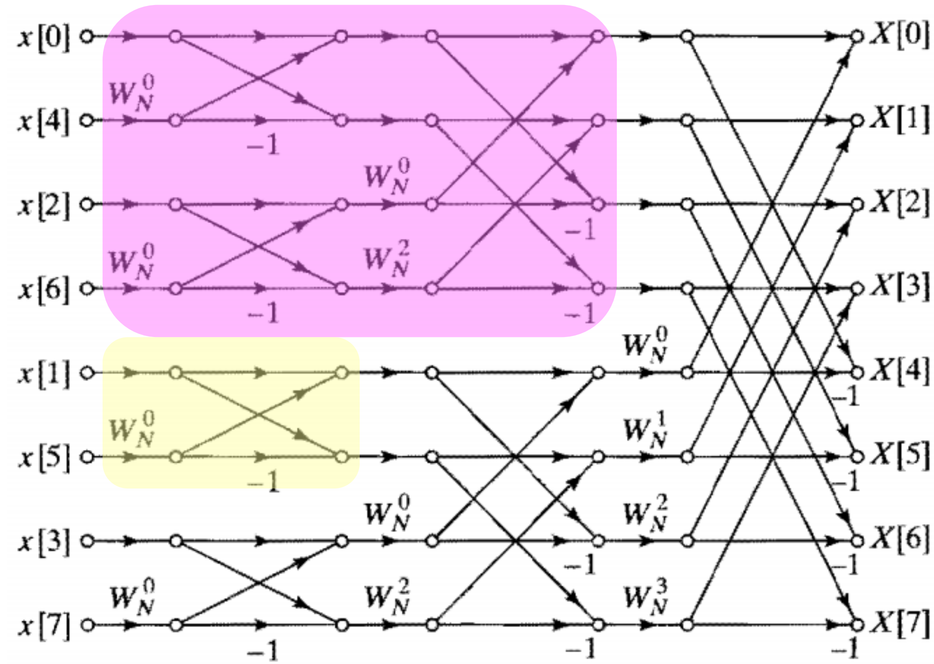
N=8的DFT（时间抽取FFT）输入序列的标号（存储和读取位序）采用二进制码可表示为

$X_0[000]=x[000]$　$X_0[100]=x[001]$

$X_0[001]=x[100]$　$X_0[101]=x[101]$

$X_0[010]=x[010]$　$X_0[110]=x[011]$

$X_0[011]=x[110]$　$X_0[111]=x[111]$

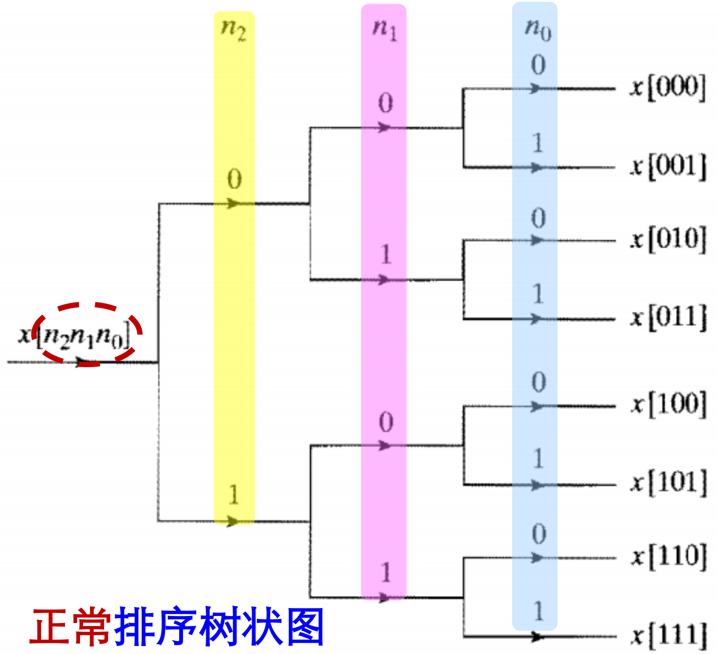


若(n_2,n_1,n_0)为输入序列$x[n]$标号的二进制表示，则序列值$x[n_2,n_1,n_0]$存放的序列值位置为$X_0[n_0,n_1,n_2]$，即按标号位序颠倒放置。

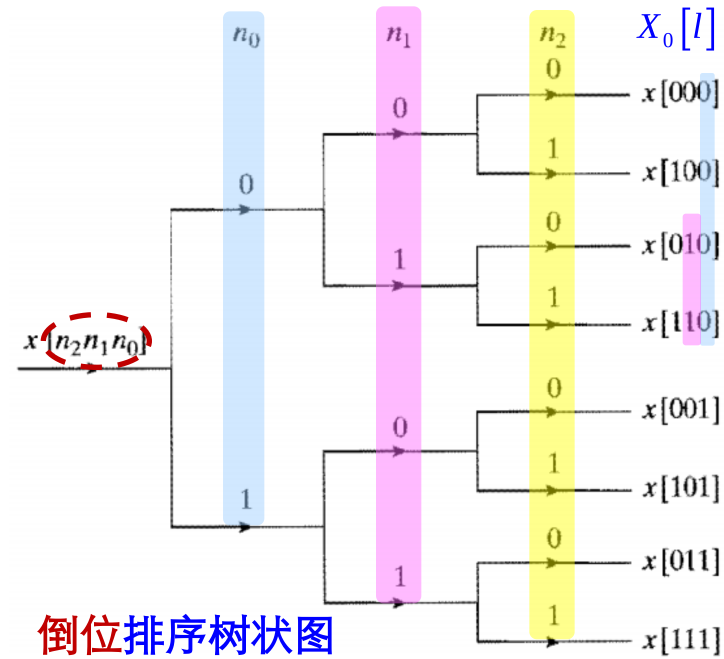
序列$x[n]$之所以要倒位排序是由于DFT的计算要逐次分解为较短的DFT计算以得到蝶形实现结构所致，即每次分解需做奇偶排序。

9.3.1 同址计算

◆ 倒位序的序列值位置确定



正常排序树状图



倒位序排序树状图

通过从低位到高位依次检查序列 $x[n]$ 的序号编码位，可确定序列值的放置位置（获取相应位置的序列值）。

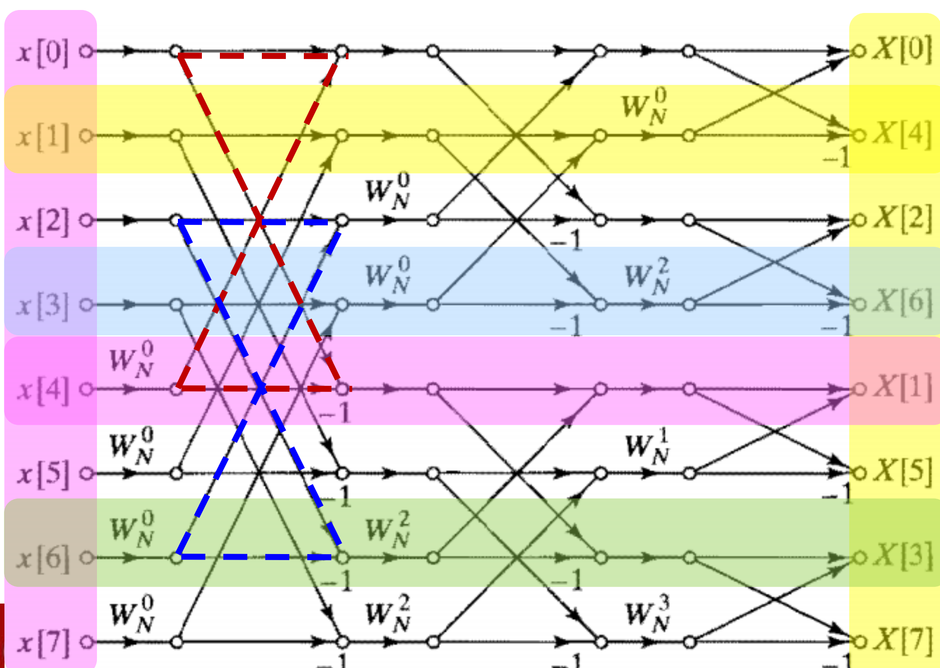
例：若第 $[n_0]$ 位为0，则 $x[n_2 n_1 0]$ 位于存储数列 $X_0[l]$ 的上半部分中；若第 $[n_1]$ 位再为1，则 $x[n_2 10]$ 位于 $X_0[l]$ 的上半数列的下半子数列中；以此类推，重复上述过程，直到第最高位确定对应的数列位置为止。

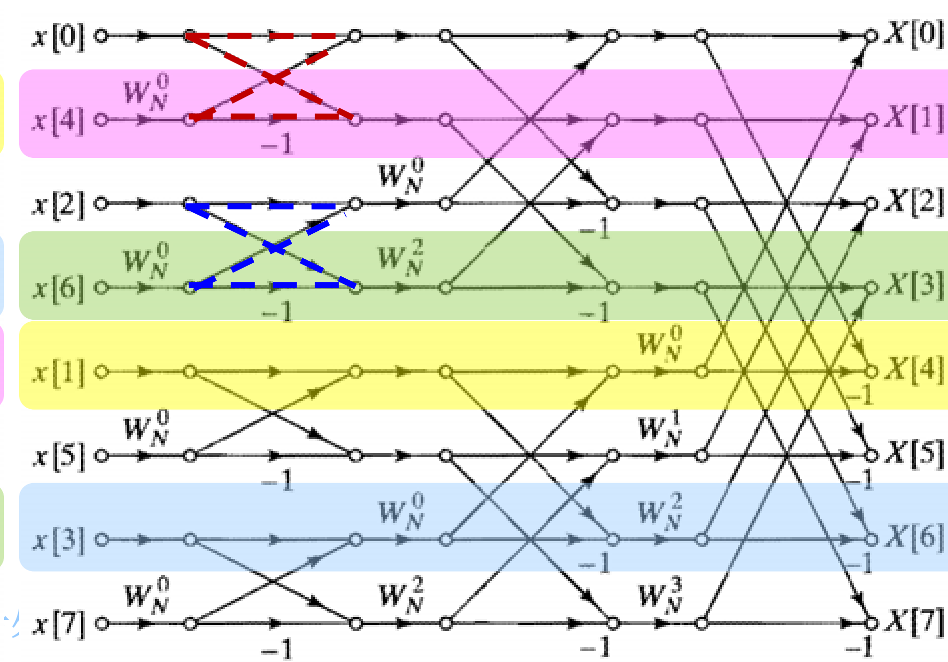
9.3.2 其它形式

按输入序列倒位序流图中出现节点的顺序存储每级计算结果，尽管合理，但不是实现有效DFT计算所必须的。

若把节点与一列复数寄存器的标号关联，只有当（只要）重新排列节点使得每个蝶形运算的输入节点和输出节点呈水平相邻时才（就）可得对应同址运算的流图。

下图中，输入序列是正常（顺）位序，而DFT输出序列是倒位序。





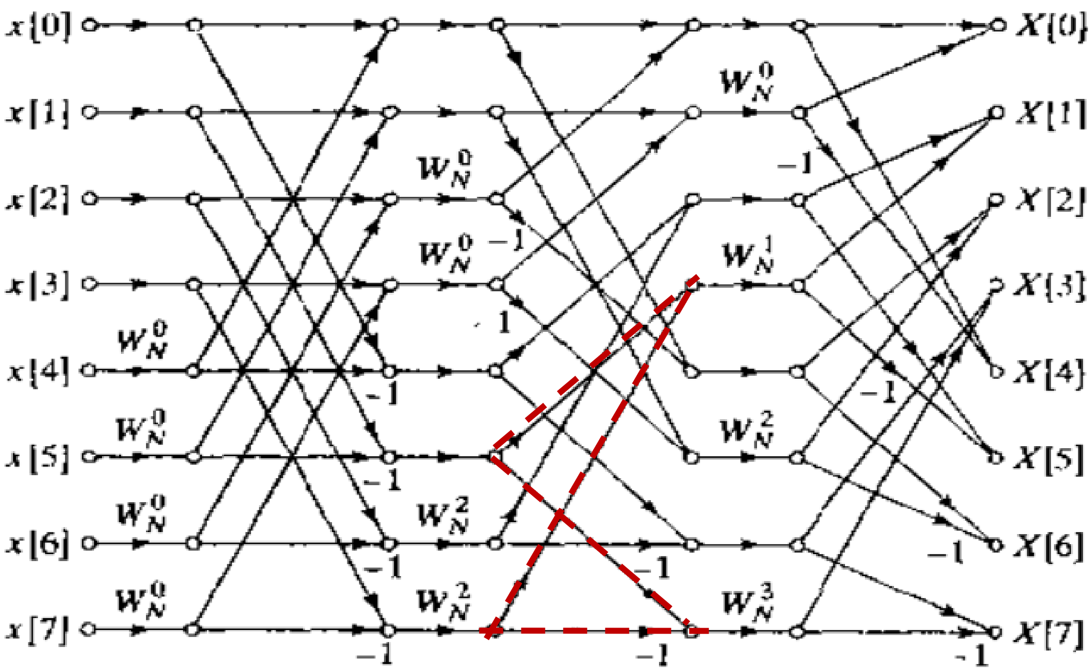
9.3.2 其它形式

◆ 对于同一个系统函数，存在多种可能的排序

下图为输入和输出序列都按正常（顺）位序排列的DFT流图

由于蝶形结构在第一级之后不能继续下去，无法实现同址计算。

DFT计算需2列长度为N的复数寄存器



9.3.2 其它形式

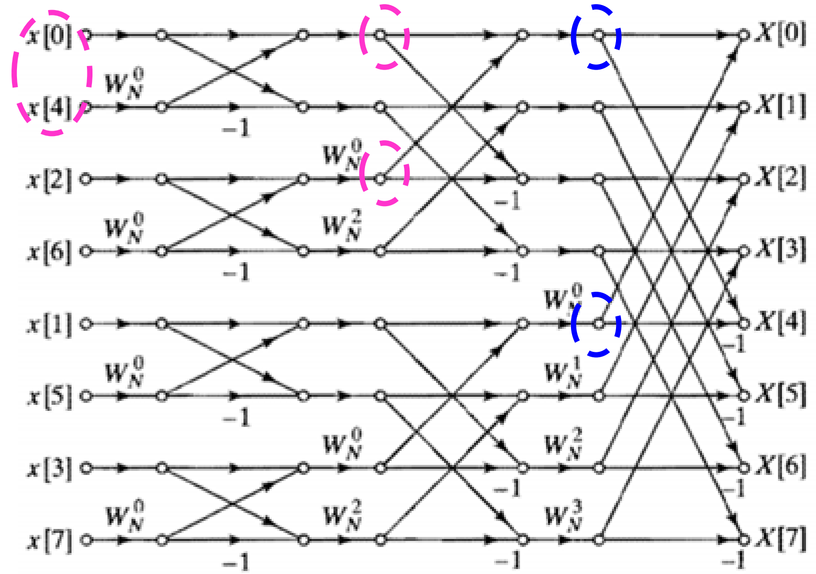
右图所示DFT实现流图，不管**输入序列**是否正常顺序排列，均**不能按顺序读取中间各列**的数。

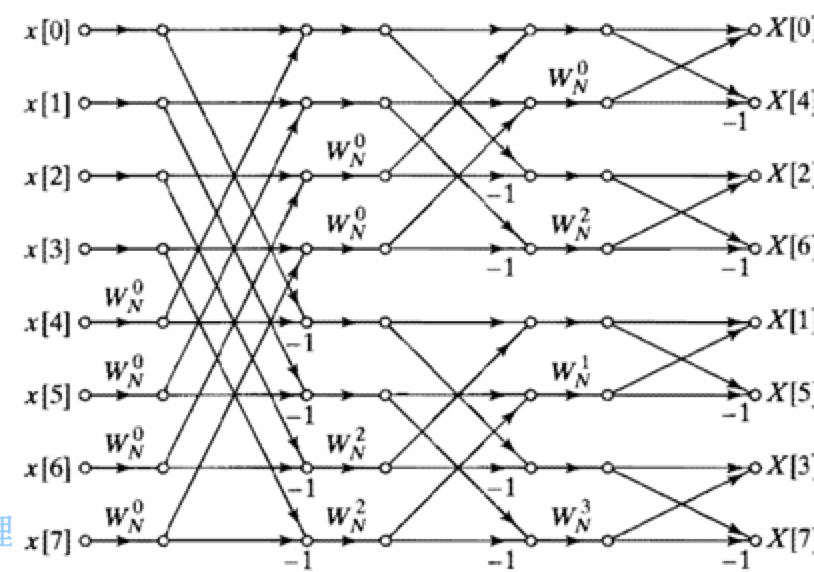
例如：右上图中

- 计算第一列时，每个蝶形计算的两个输入存放在**相邻存储位置**上
- 计算第二列时，蝶形计算的两个输入端**间隔2个存储位置**；
- 计算第三列时，蝶形计算的两个输入端**间隔4个存储位置**；
- 以此类推，最后一级的两个输入端**间隔$N/2$个存储位置**；

除第一列外，均**不能按顺序**读取

因此，中间结果必须**存放在随机存储器**中。





9.3.2 其它形式

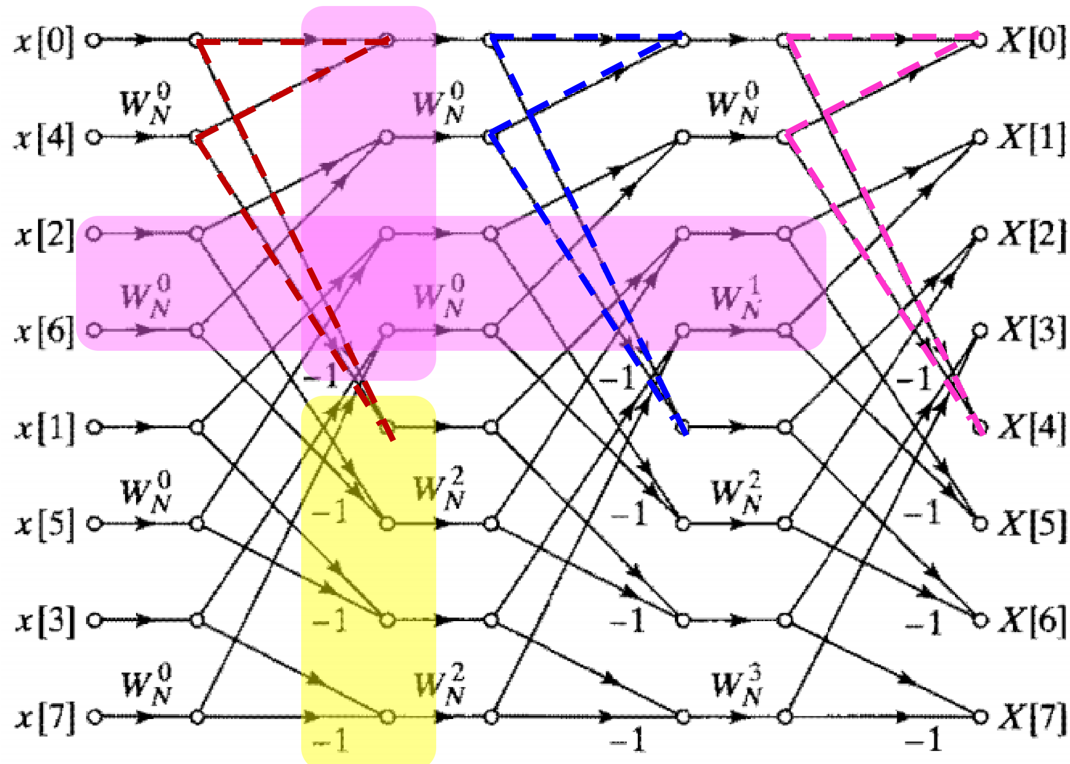
◆每一级均有相同流图形状的DFT实现形式

尽管不能实现同址计算，但具有重要特点：

➢ 对每一级计算采用相同的流图形状

➢各级之间只有支路的传输比(W_N的幂值k) 不同

该结构使得每一级计算的数据存取可按顺序进行。



➢ 每一级各蝶形计算的两个输入端的位置相邻，以便顺序取数

➢ 计算输出的偶和奇序号序列分别按顺序排列，以便顺序存数

该结构特别适合于无随机存储器或由于DFT点数极其大（如上亿点DFT）使得所需寄存器无法实现的应用情况。

第九章：离散傅里叶变换的计算

9.4 按频率抽取FFT算法

◆ 按频率抽取的DFT分解

考虑长度N为2的整数幂的序列 $x[n]$ 的N点DFT，分别计算DFT偶序号和奇序号频率样值。

因为 $x[n]$的N点DFT可表示为

$$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad k=0,1,\cdots,N-1$$

其中偶数序号频率样值为

$$X[2r]=\sum_{n=0}^{N-1} x[n]W_N^{2rn}, \quad r=0,1,\cdots,N/2-1$$

上式可表示为

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

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

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

◆ 按频率抽取的DFT分解（续1）

由于 W_N^{2rn} 的周期性，有 $W_N^{2r[n+(N/2)]}=W_N^{2rn}W_N^{rN}=W_N^{2rn}$

并且有 $W_N^2=W_{N/2}$ ，而 $W_N^{2rn}=W_{N/2}^{rn}$

因此 $x[n]$ 的N点DFT的偶数序号频率样值可表示为

$$X[2r]=\sum_{n=0}^{(N/2)-1}\left(x[n]+x\left[n+(N/2)\right]\right)W_{N/2}^{rn},\quad r=0,1,\cdots,N/2-1$$

序列 $x[n]$ 的N点DFT的偶数频率样值为 $x[n]$ 的前一半序列加上后一半序列得到的和序列的$N/2$点DFT。

9.4 按频率抽取FFT算法

◆ 按频率抽取的DFT分解（续2）

对于DFT的奇数序号频率样点

$$X[2r+1]=\sum_{n=0}^{N-1}x[n]W_N^{(2r+1)n},\quad r=0,1,\cdots,N/2-1$$

上式可表示为

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

$$W_N^{N/2(2r+1)}=W_N^{Nr}W_N^{N/2}=-1$$

利用 $W_N^2=W_{N/2}$，即有 $W_N^{n(2r+1)}=W_N^nW_{N/2}^{nr}$，则

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

序列 $x[n]$ 的N点DFT的奇数频率样值为 $x[n]$ 的前一半减去后一半得到的差序列乘以系数 W_N^n 后的$N/2$点DFT。

9.4 按频率抽取FFT算法

◆ 按频率抽取的DFT分解（续3）

$$g[n]=x[n]+x[n+N/2],\quad h[n]=x[n]-x[n+N/2],\quad n=0,1,\cdots,N/2-1$$

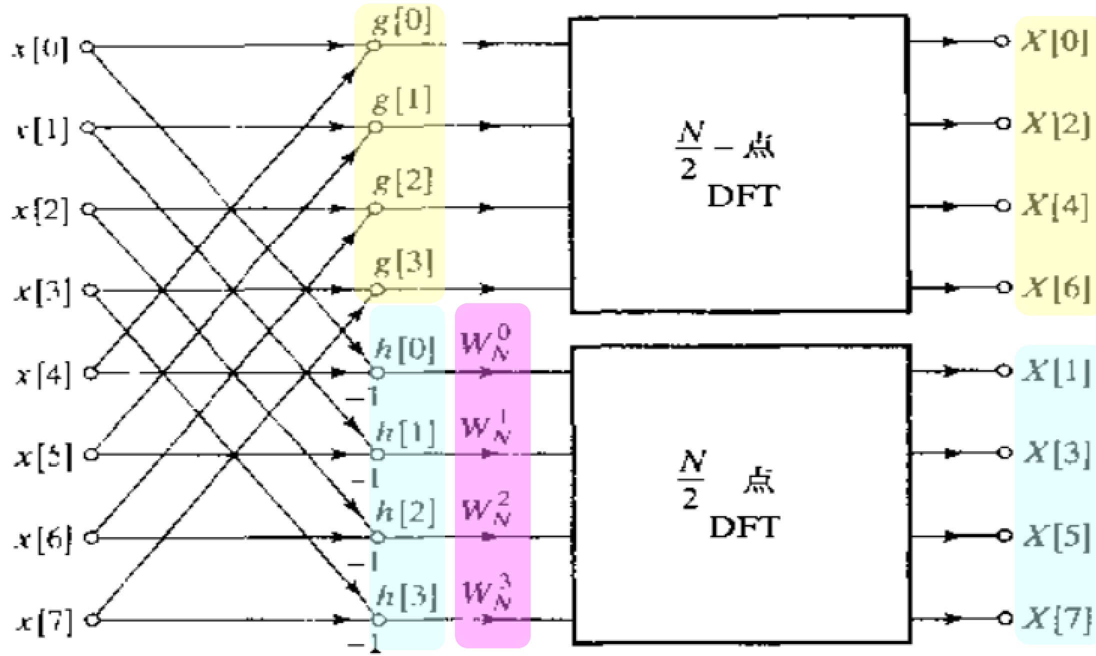
则按频率抽取DFT计算方法如下：

1）首先形成序列 $g[n]$ 和 $h[n]$，并计算 $h[n]W_N^n$ ；

2）然后分别计算 $g[n]$ 和 $h[n]W_N^n$ 的N/2点DFT，获得偶数和奇数序号输出样本。

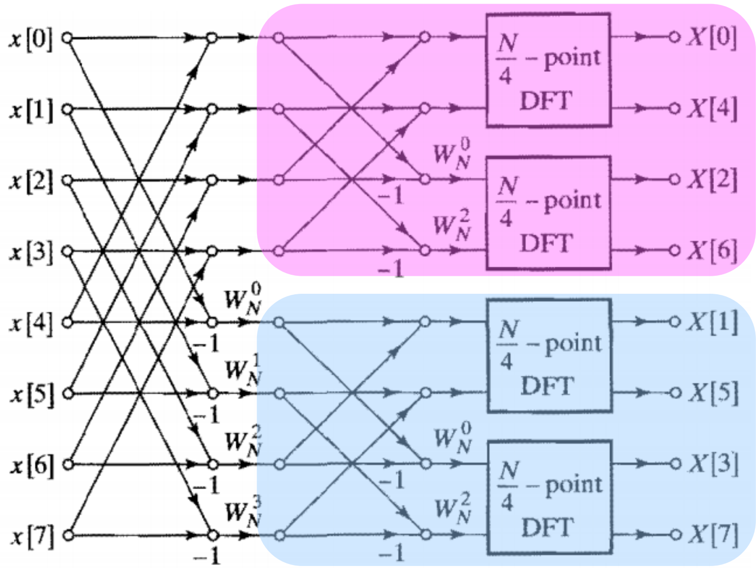
右图为将一个N点DFT分解为两个（N/2）点DFT计算的按频率抽取DFT实现流图。（系数仅与差序列相乘）

其中（N/2）点DFT可按频率抽取法继续分解为两个（N/4）点的DFT来实现。

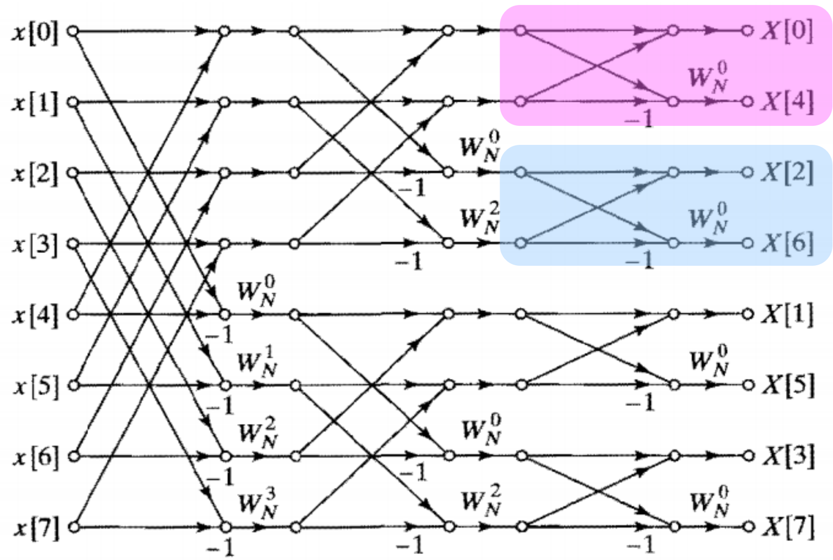


9.4 按频率抽取FFT算法

◆ 例：8点DFT分解的按频率抽取流图

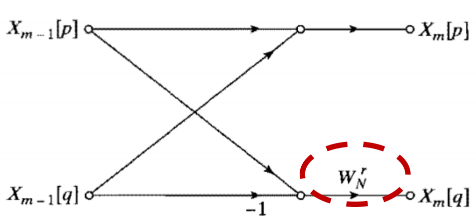


分解为4个2点DFT计算的按频率抽取流图



按频率抽取8点DFT完整流图

其中按频率抽取2点DFT流图为



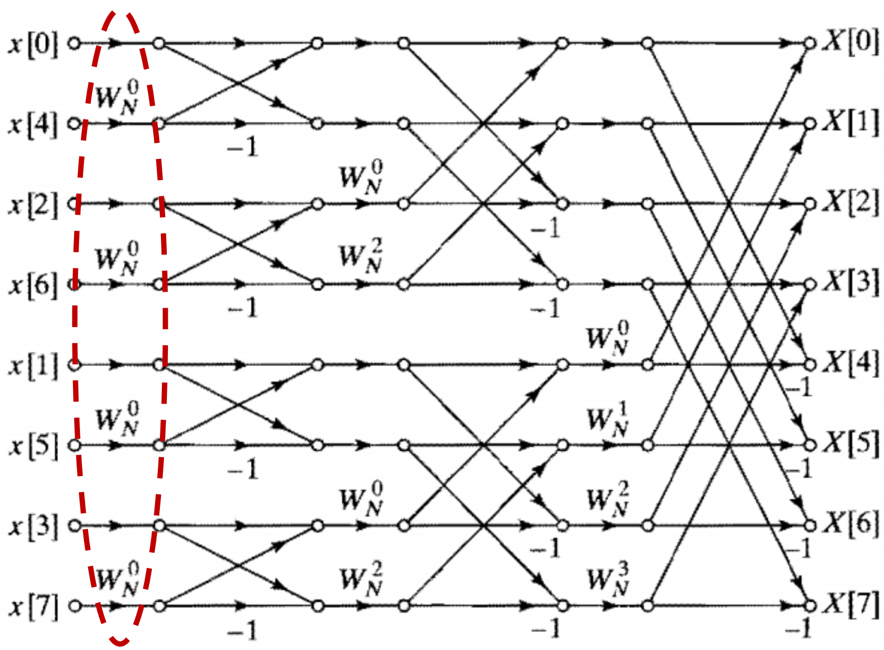
一般对于$N=2^v$ 的情况，按频率抽取DFT计算需：

- ◆ （$N/2$）$\log_2 N$次复乘
- ◆ $N\log_2 N$次复加

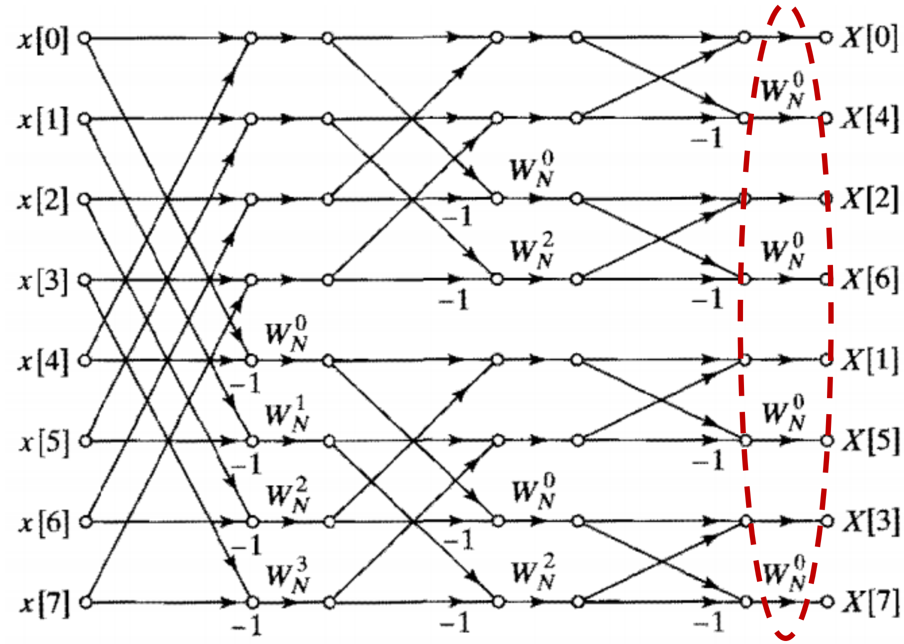
按频率抽取DFT算法与按时间抽取DFT算法具有相同的计算量

9.4.1 同址计算

◆ 按频率抽取的FFT计算结构



按时间抽取8点DFT完整流图



按频率抽取8点DFT完整流图

按频率抽取DFT算法流图的基本计算仍为蝶形计算，因此也是一种（也可以支持）同址运算。

9.4.2 其它形式

◆ 按频率抽取的DFT与按时间抽取DFT之间的关系

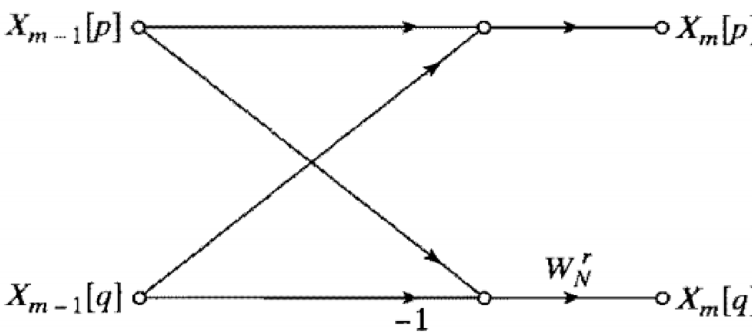
对按时间抽取DFT算法流图进行转置即可获得相应的按频率抽取DFT计算方法

令第m级计算的输出序列为 $X_m[l]$，其中 $l=0,1,\cdots,N-1$，$m=1,2,\cdots,v$

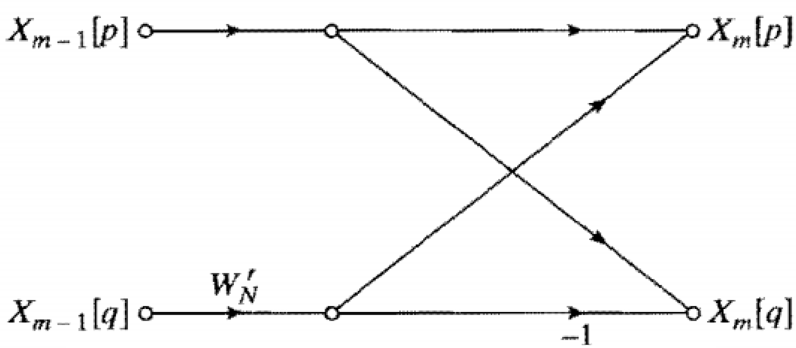
则按频率抽取DFT算法第m级蝶形计算可表示为

$$X_m[p]=X_{m-1}[p]+X_{m-1}[q],$$
$$X_m[q]=\left(X_{m-1}[p]-X_{m-1}[q]\right)W_N^r$$

按频率抽取DFT蝶形计算流图可通过按时间抽取DFT蝶形算法流图的转置获得。



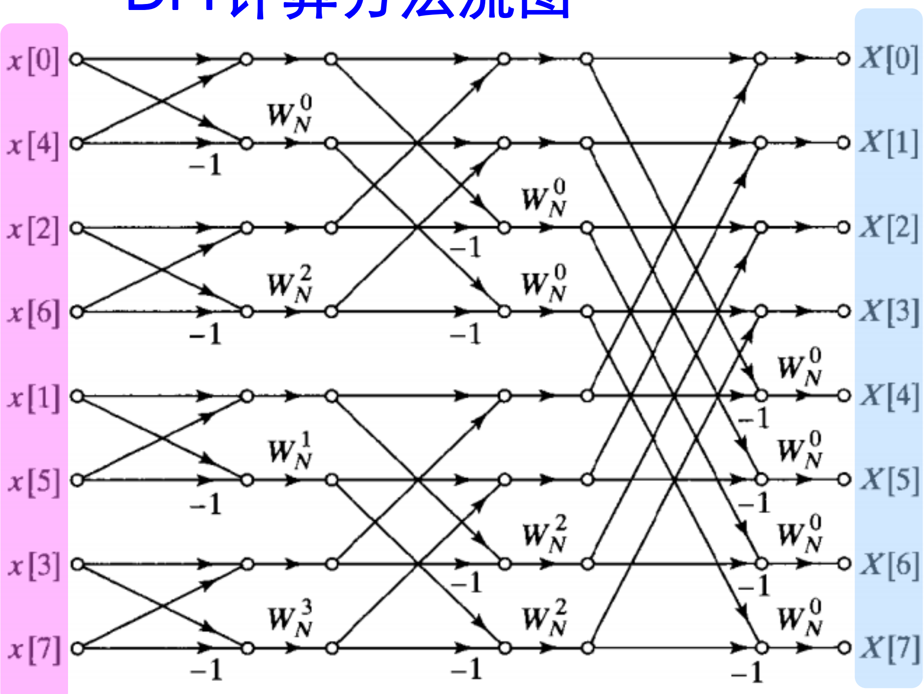
按频率抽取DFT蝶形计算流图



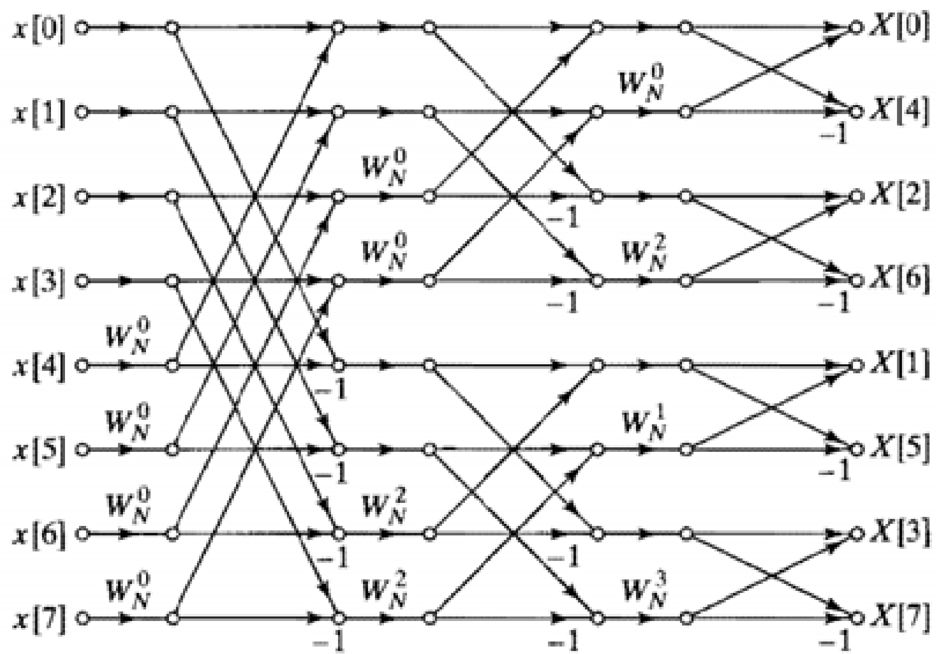
按时间抽取DFT蝶形计算流图

9.4.2 其它形式

◆ 示例：由按时间抽取DFT算法流图进行转置获得按频率抽取DFT计算方法流图



按频率抽取DFT计算方法流图
（输入倒位序/输出顺位序）

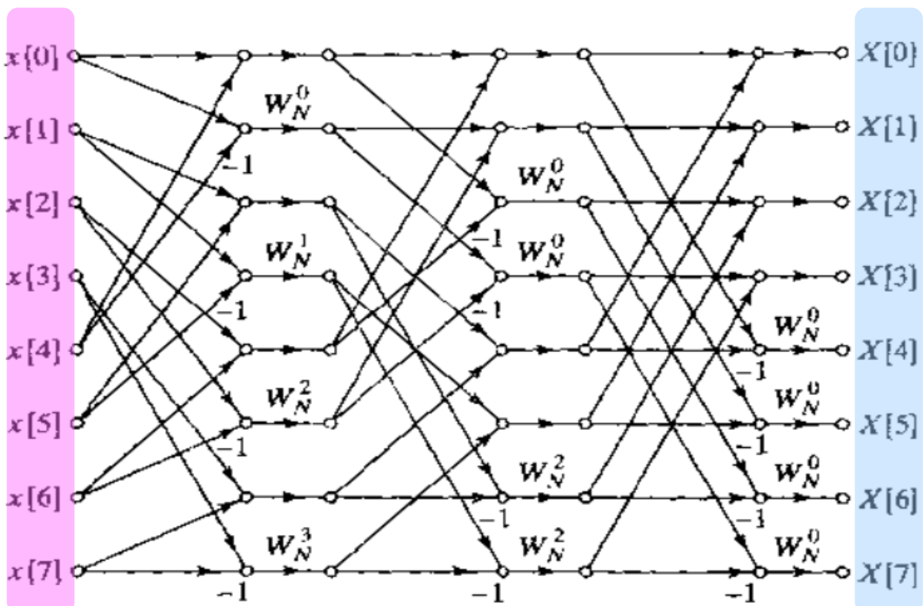


按时间抽取DFT计算方法流图
（输入顺位序/输出倒位序）

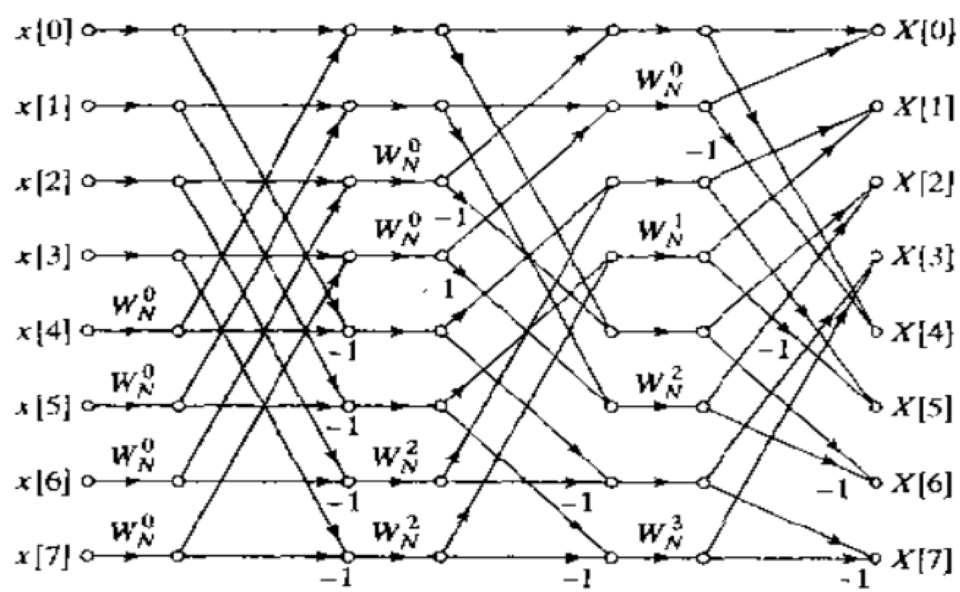
尽管转置后总体输出特性与原流图相同，但由于FFT计算有多个输入和输出端（无输入输出一一对应关系），所以转置定理不适用于FFT算法流图。

9.4.2 其它形式

◆ 示例：由按时间抽取DFT算法流图进行转置获得按频率抽取DFT计算方法流图（续2）



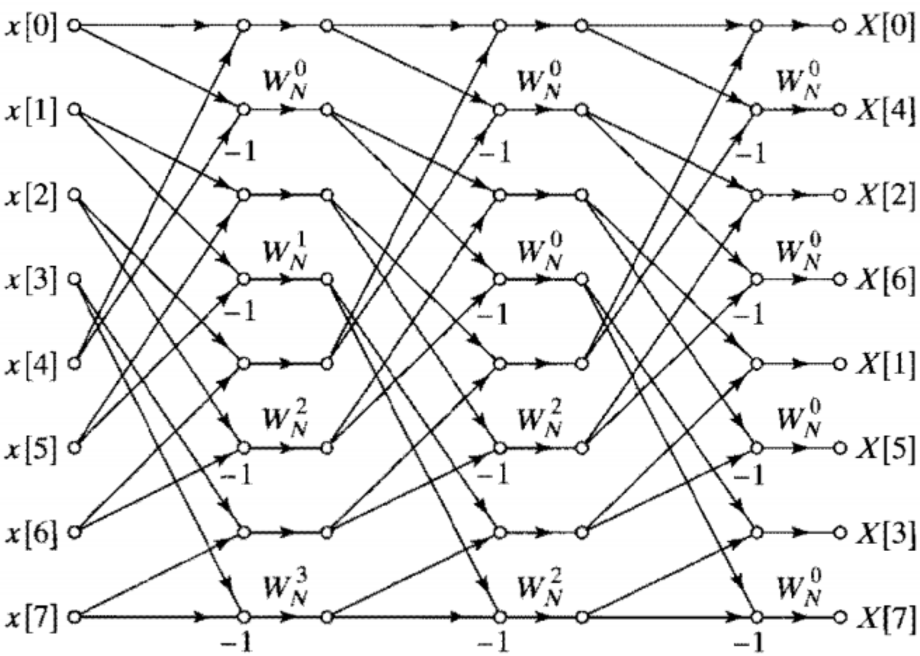
按频率抽取DFT计算方法流图
（输入/输出顺位序）



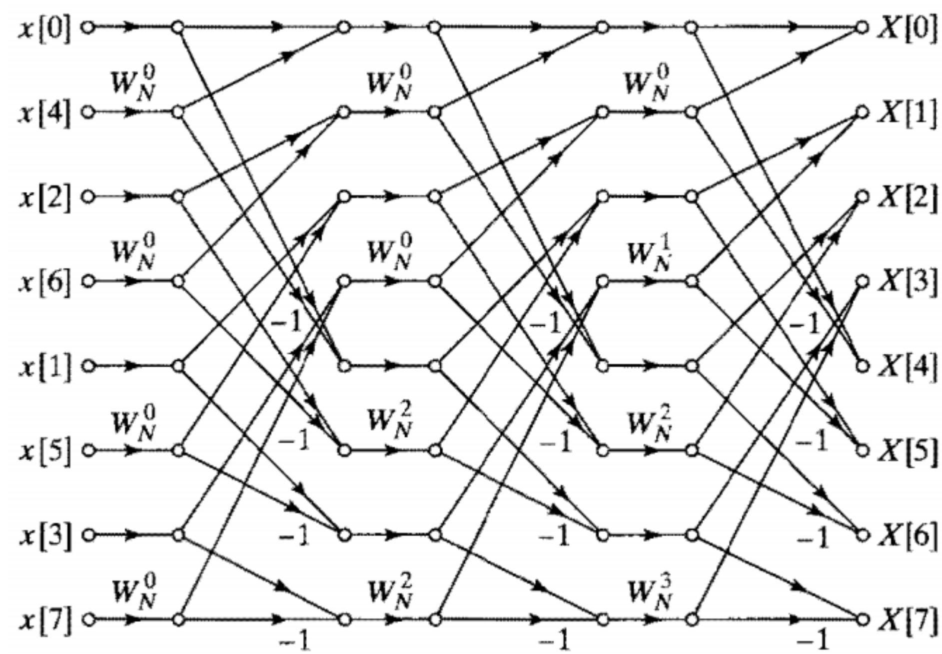
按时间抽取DFT计算方法流图
（输入/输出顺位序）

9.4.2 其它形式

◆ 示例：由按时间抽取DFT算法流图进行转置获得按频率抽取DFT计算方法流图（续3）



按频率抽取DFT计算方法流图
（输入顺位序/输出倒位序）



按时间抽取DFT计算方法流图
（输入倒位序/输出顺位序）