# 快速傅里叶变换

许向阳
xuxy@hust.edu.cn

参考：

何东健，数字图像处理，西安电子科技大学出版社，2003，7.2节

# 快速傅立叶变换



- 问题的提出
- 解决问题的思路与方法
- 基2时间抽取FFT算法
- 基2时间抽取FFT算法的计算复杂度
- 基2时间抽取FFT算法流图规律
- 基2频率抽取FFT算法
- FFT算法的实际应用

# 问题的提出



4点序列{2，3，3，2} DFT的计算复杂度

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

$$X[0]=2W_N^0+3W_N^0+3W_N^0+2W_N^0=10$$

$$X[1]=2W_N^0+3W_N^1+3W_N^2+2W_N^3=-1-j$$

$$X[2]=2W_N^0+3W_N^2+3W_N^4+2W_N^6=0$$

$$X[3]=2W_N^0+3W_N^3+3W_N^6+2W_N^9=-1+j$$

复数加法 ***N*(*N*-1)**　　复数乘法 ***N*** $^2$

如何提高DFT的运算效率?

# 解决问题的思路

华中科技大学

1. 将长序列DFT分解为短序列的DFT

2. 利用旋转因子$W_N^{km}$的周期性、对称性、可约性。

# 旋转因子 $W_N^{km}$ 的性质

1)周期性 $$W_N^{(k+N)m}=W_N^{k(m+N)}=W_N^{km}$$

2) 对称性

$$W_N^{mk+\frac{N}{2}}=-W_N^{mk} \qquad \left(W_N^{km}\right)^*=W_N^{-mk}$$

3)可约性 $$W_N^{mk}=W_{nN}^{nmk}$$

$$W_N^{mk}=W_{N/n}^{mk/n}, \quad N/n\text{为整数}$$

# 解决问题的方法

华中科技大学

将时域序列逐次分解为一组子序列，利用旋转因子的特性，由子序列的DFT来实现整个序列的DFT。

**基2时间抽取(Decimation in time)FFT算法**

$$x[k] \to \begin{cases} x[2r] \\ x[2r+1] \end{cases} \quad r = 0,1,\cdots\frac{N}{2}-1$$

**基2频率抽取(Decimation in frequency)FFT算法**

$$X[m] \to \begin{cases} X[2m] \\ X[2m+1] \end{cases}$$

# 基2时间抽取FFT算法思想

设$N=2M$

$$F(u)=\sum_{x=0}^{2M-1} f(x)W_{2M}^{ux}$$

$$=\sum_{x=0}^{M-1} f(2x)W_{2M}^{u(2x)}+\sum_{x=0}^{M-1} f(2x+1)W_{2M}^{u(2x+1)}$$

$$=\sum_{x=0}^{M-1} f(2x)W_{M}^{ux}+\sum_{x=0}^{M-1} f(2x+1)W_{M}^{ux}W_{2M}^{u}$$

# 基2时间抽取FFT算法思想

$$F_e(u)=\sum_{x=0}^{M-1} f(2x)W_M^{ux} \qquad u,x=0,1,\ldots,M-1$$

$$F_o(u)=\sum_{x=0}^{M-1} f(2x+1)W_M^{ux} \qquad u,x=0,1,\ldots,M-1$$

$$F(u)=F_e(u)+W_{2M}^{u}F_o(u) \qquad \mathrm{u}=0,1,\ldots.,\mathrm{M}-1$$

$$F(u+M)=\sum_{x=0}^{M-1} f(2x)W_M^{(u+M)x}+\sum_{x=0}^{M-1} f(2x+1)W_M^{(u+M)x}W_{2M}^{(u+M)}$$

$$=\sum_{x=0}^{M-1} f(2x)W_M^{ux}-\sum_{x=0}^{M-1} f(2x+1)W_M^{ux}W_{2M}^{u}$$

$$=F_e(u)-W_{2M}^{u}F_o(u)$$

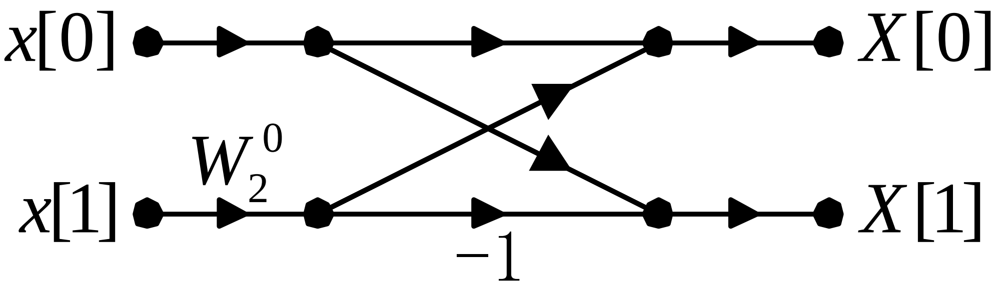
# 基2时间抽取FFT算法流图

$N$=2　　　　$x[k]=\{x[0], x[1]\}$

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

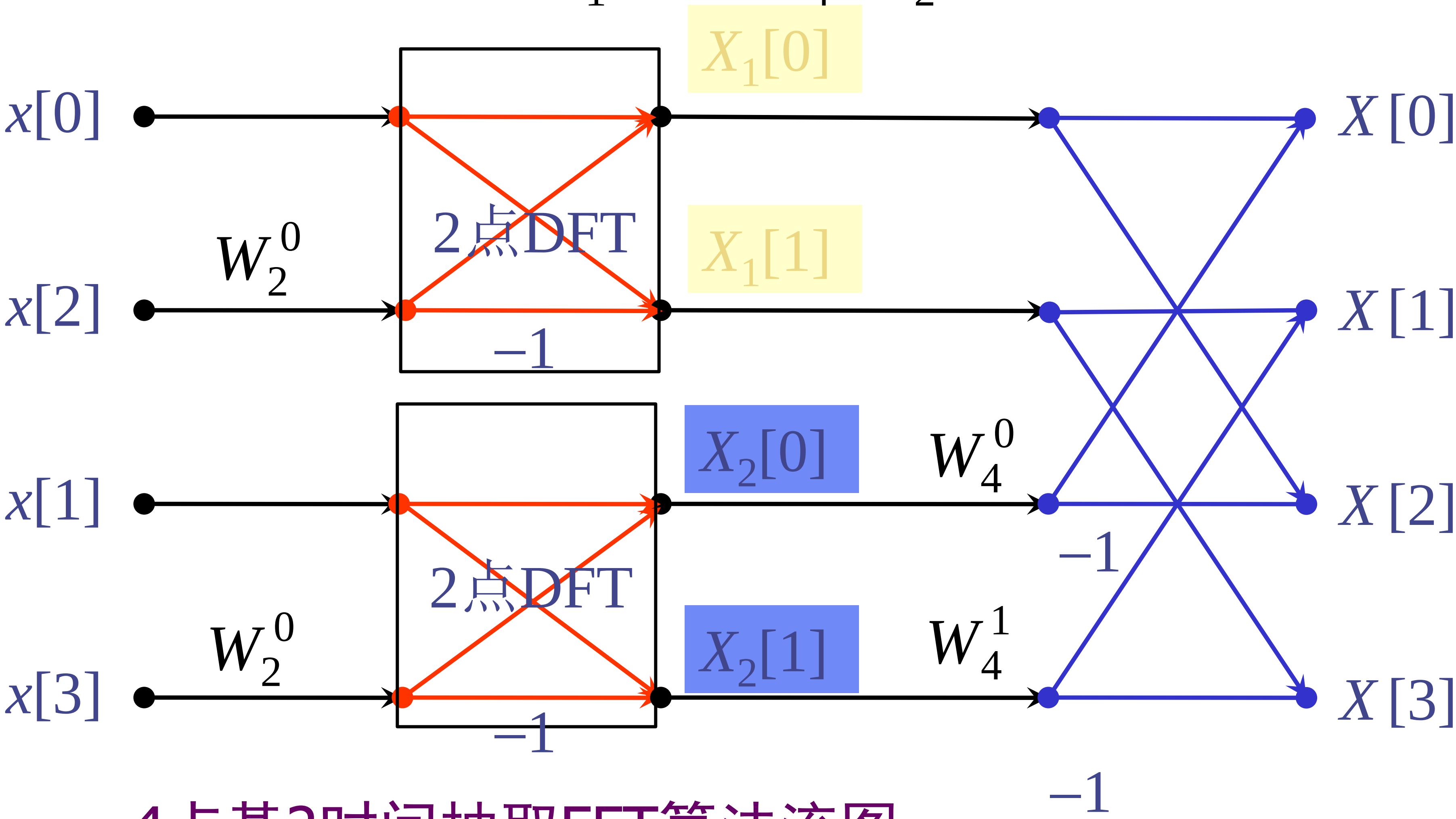
$$X[0]=x[0]+W_2^0x[1]$$

$$X[1]=x[0]+W_2^1x[1] = x[0]-W_2^0x[1]$$



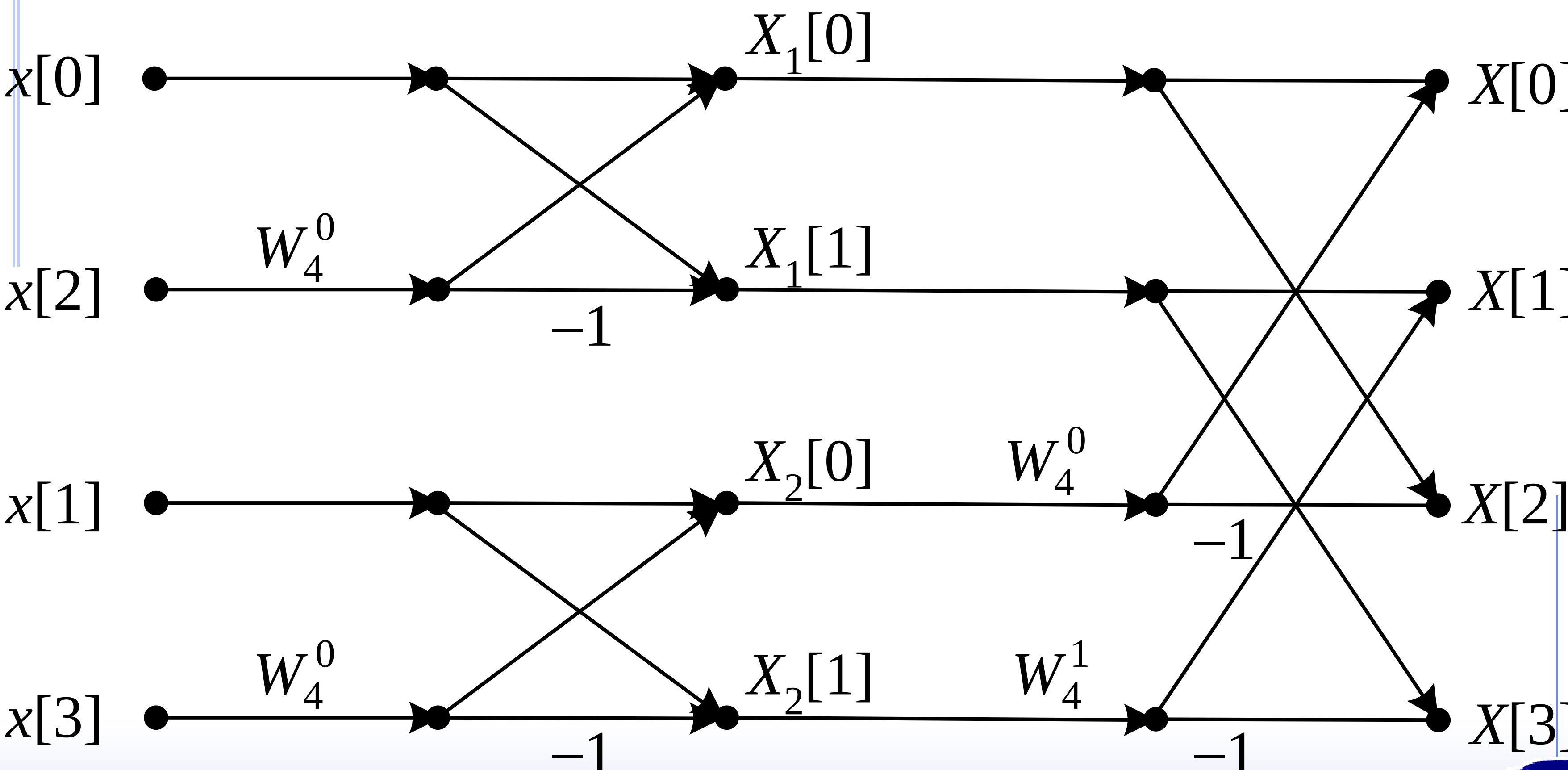
$$X[m] = X_1[m] + W_4^m X_2[m], \quad m = 0,1$$

$$X[m+2] = X_1[m] - W_4^m X_2[m], \quad m = 0,1$$



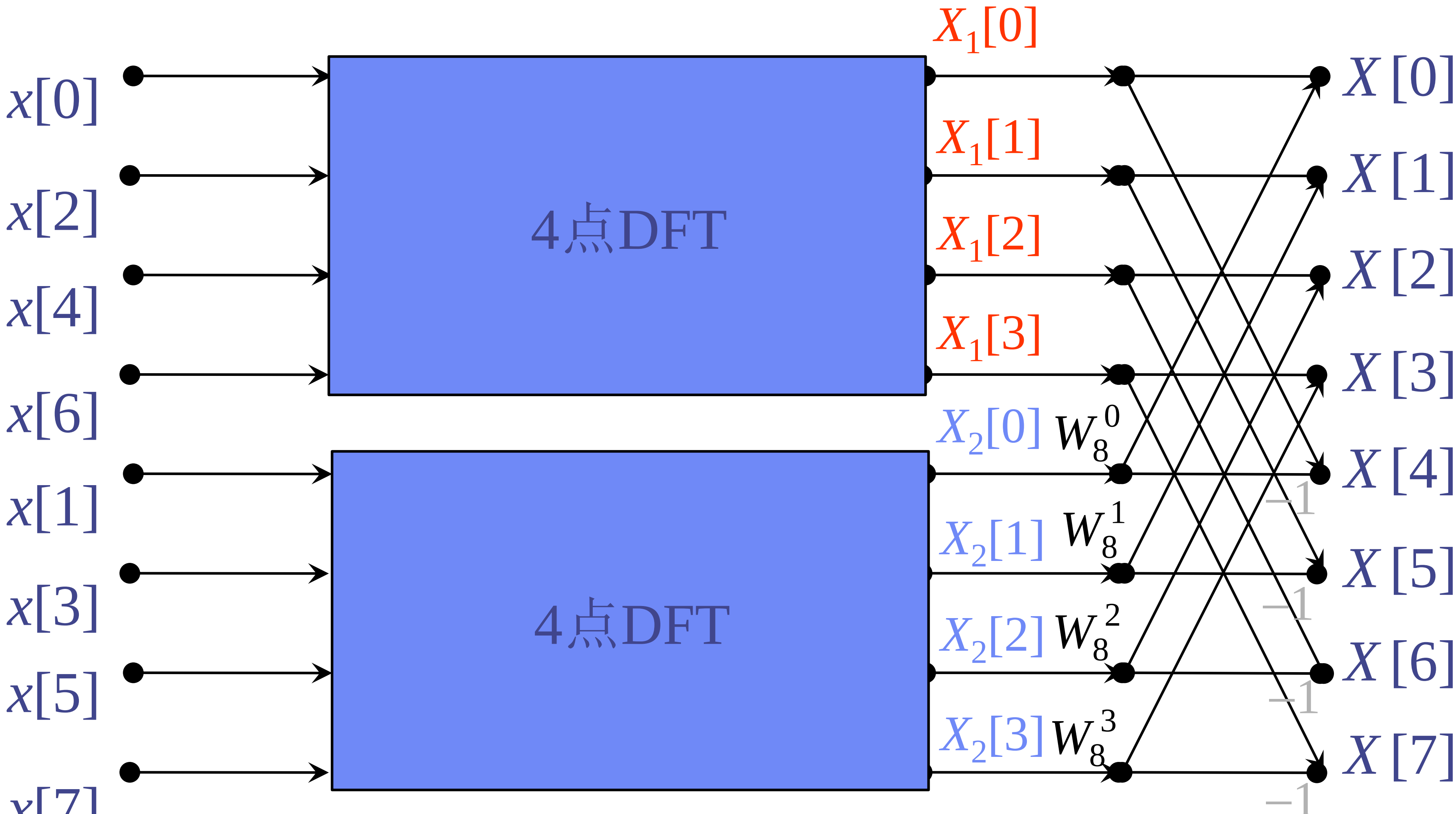

4点基2时间抽取FFT算法流图

# 4点基2时间抽取FFT算法流图



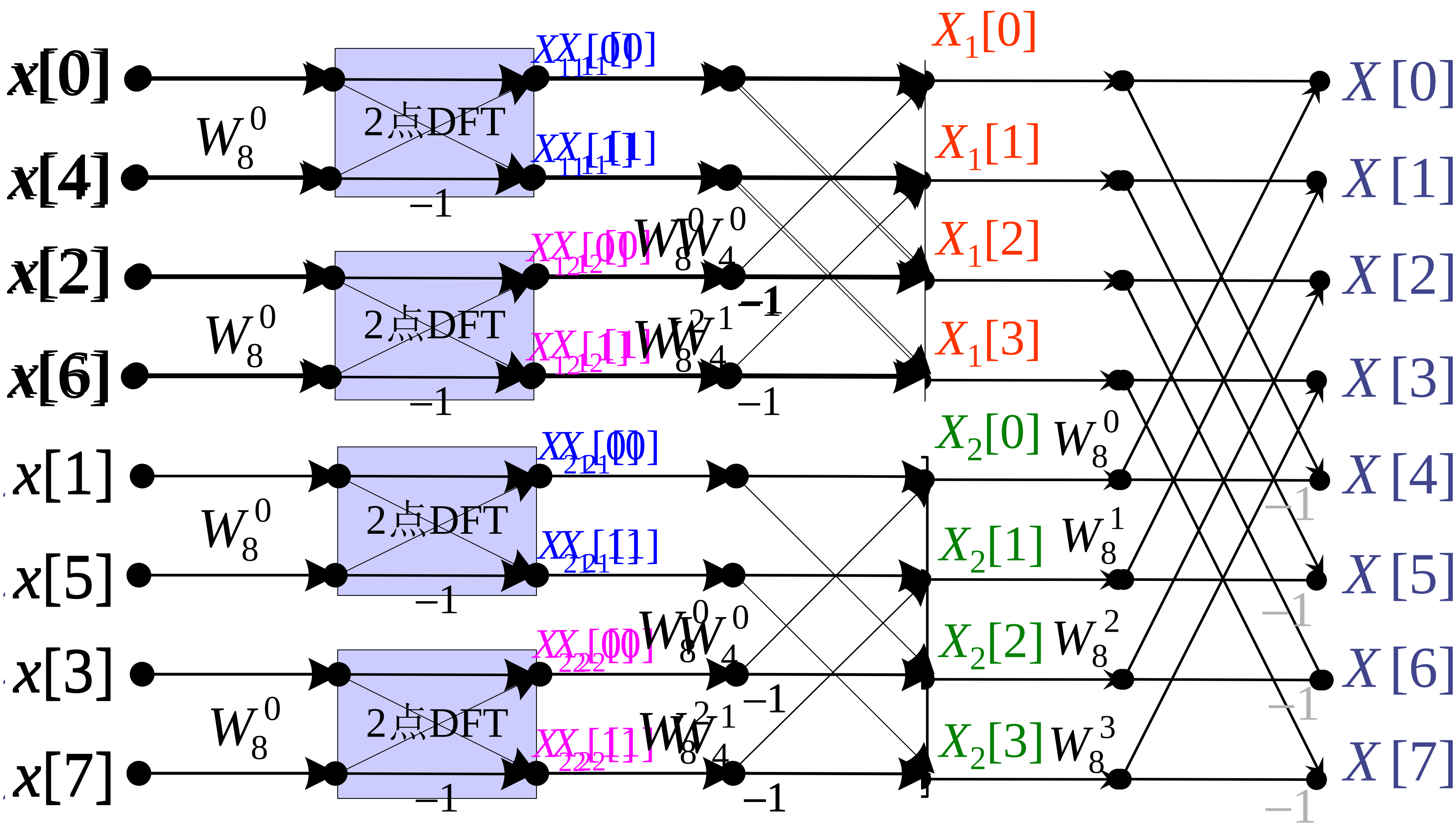
$$X[m] = X_1[m] + W_8^m X_2[m], \quad m = 0,1,2,3$$

$$X[m+4] = X_1[m] - W_8^m X_2[m], \quad m = 0,1,2,3$$

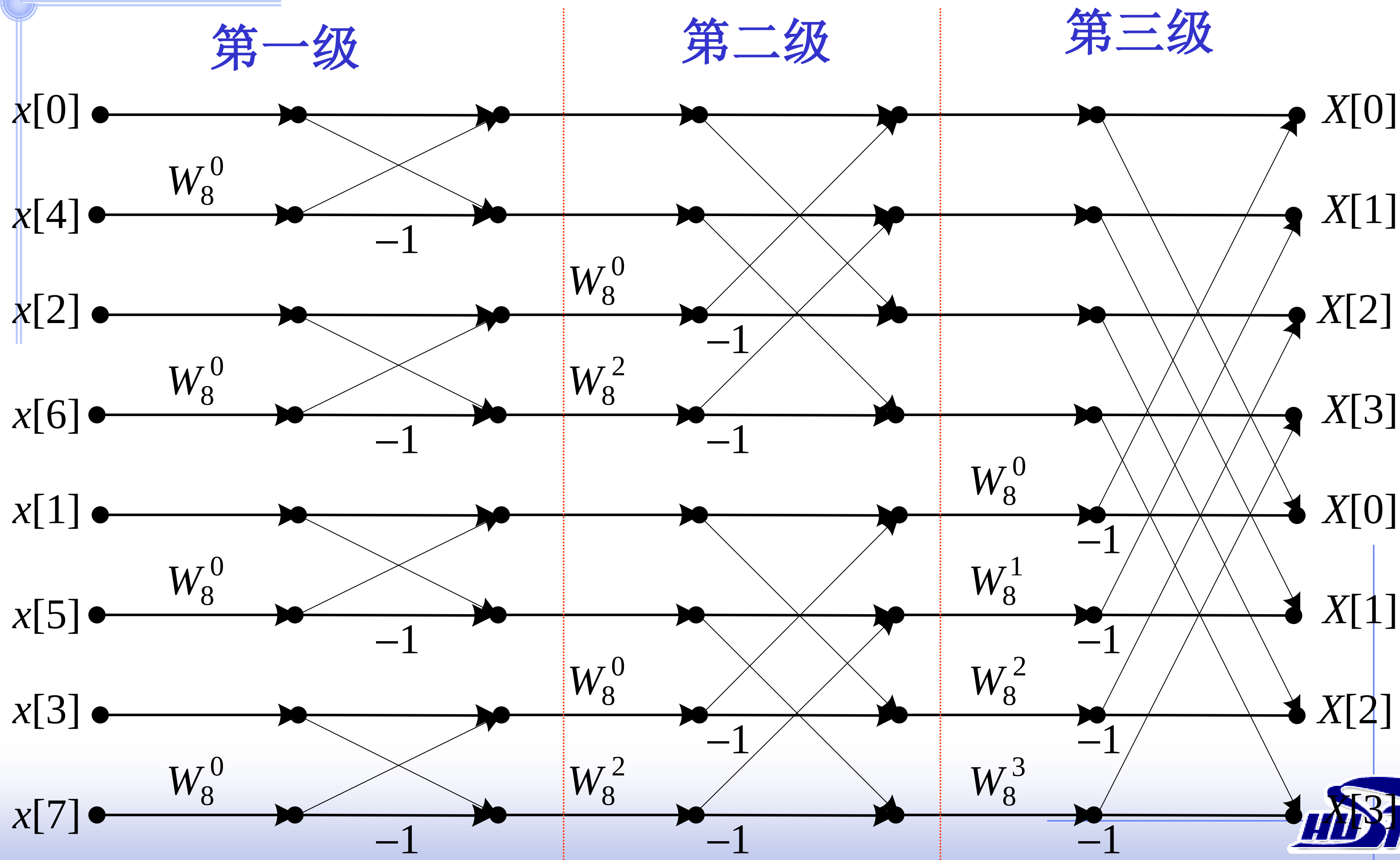



8点基2时间抽取FFT算法流图

# 8点基2时间抽取FFT算法流图



# 基2时间抽取FFT算法

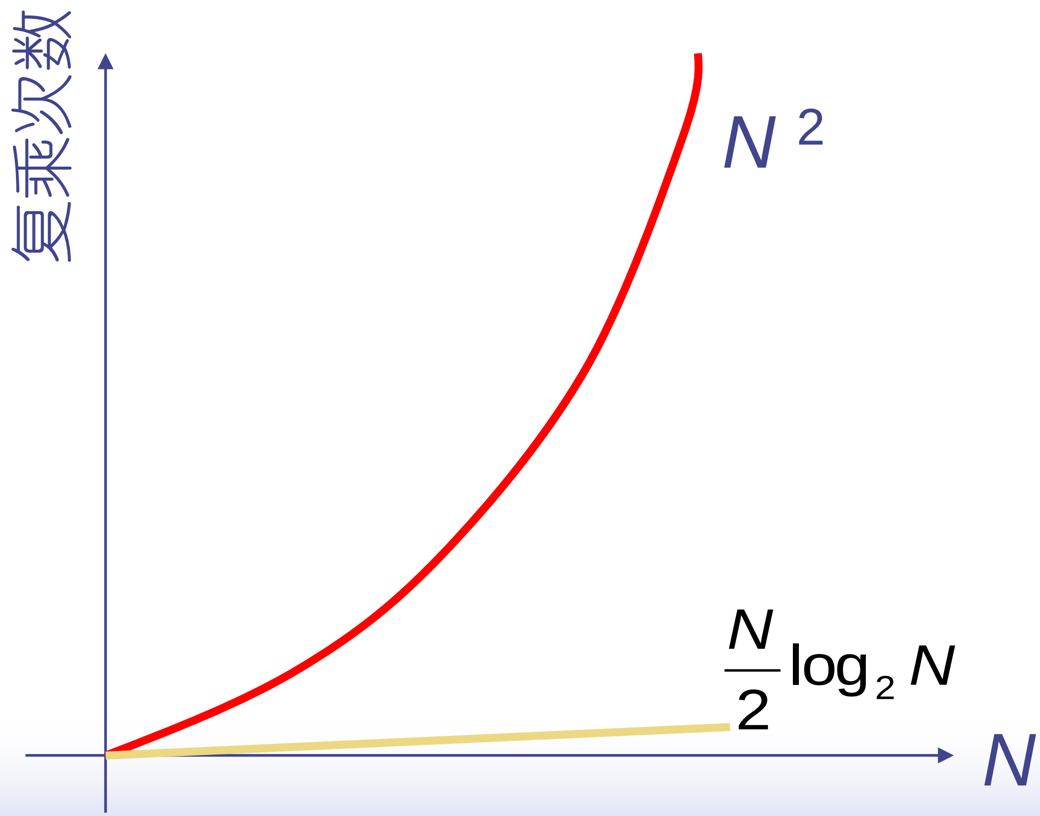


# 算法的计算复杂度

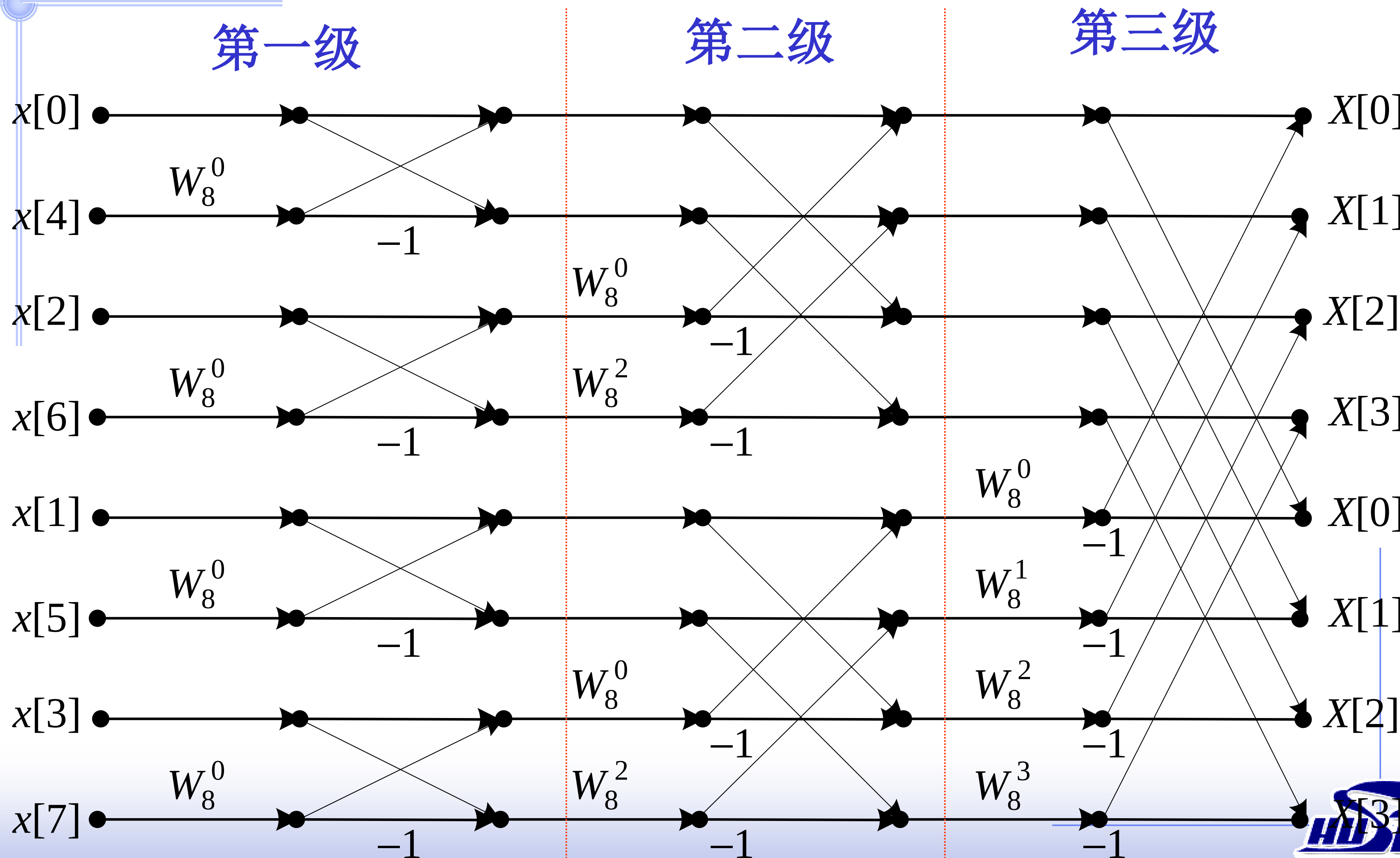


复乘次数

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



# 基2时间抽取FFT算法流图



# FFT算法流图旋转因子 $W_N^P$ 规律

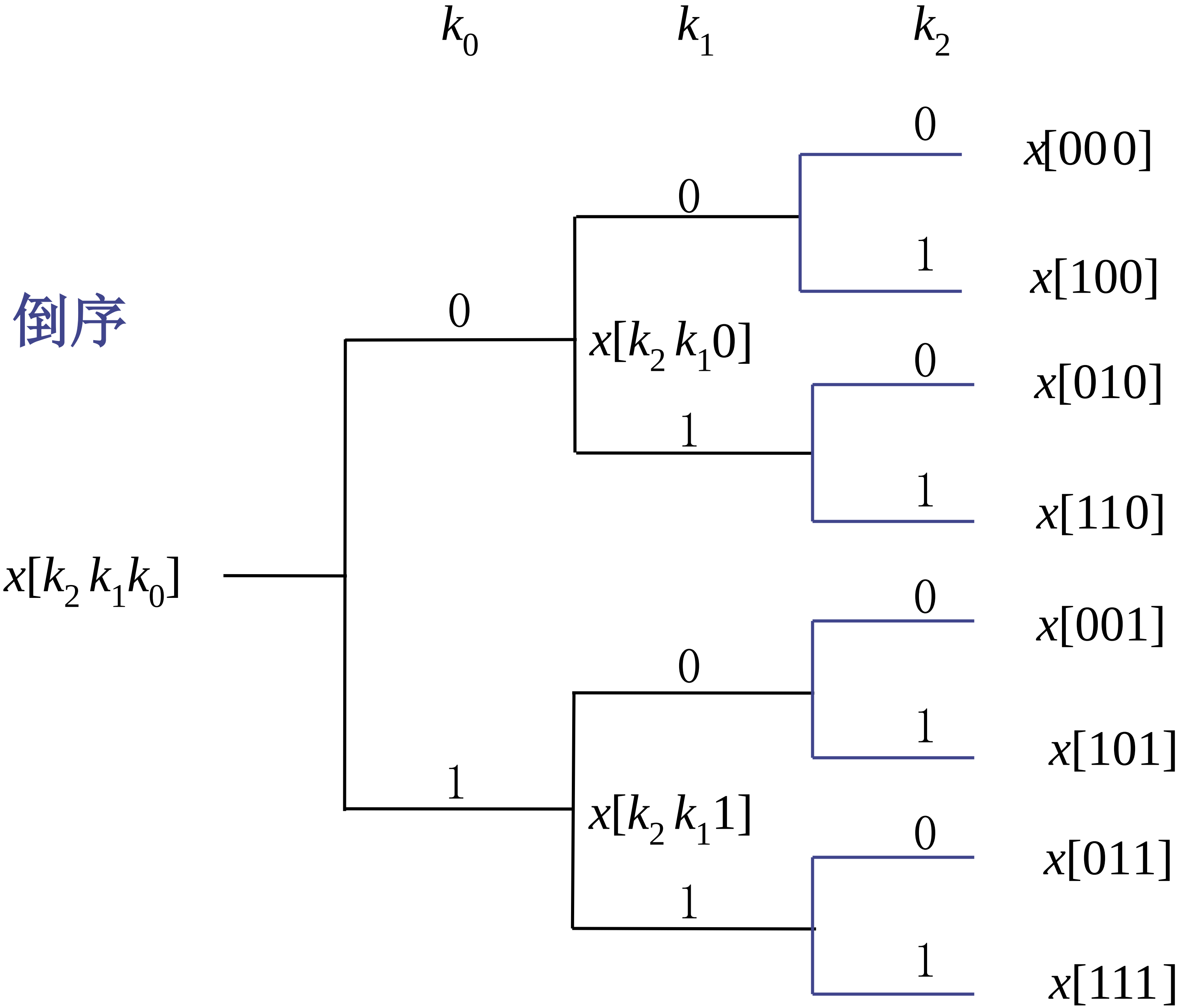
第一级的蝶形系数均为 $W_N^0$ 蝶形节点的距离为1。

第二级的蝶形系数为 $W_N^0,\ W_N^{N/4}$，蝶形节点的距离为2。

第三级的蝶形系数为 $W_N^0,\ W_N^{N/8},\ W_N^{2N/8},\ W_N^{3N/8}$，蝶形节点的距离为4。

第$M$级 的蝶形系数为 $W_N^0,\ W_N^1,\ \cdots,\ W_N^{(N/2-1)}$，蝶形节点的距离为$N/2$。

倒序



$k_0$
$k_1$
$k_2$
0
1
$x[k_2 k_1 k_0]$
$x[k_2 k_1 0]$
$x[k_2 k_1 1]$
$x[000]$
$x[100]$
$x[010]$
$x[110]$
$x[001]$
$x[101]$
$x[011]$
$x[111]$

# 基2频率抽取FFT算法

华中科技大学

$$X[m]=\sum_{k=0}^{N/2-1}x[k]W_N^{mk}+\sum_{k=N/2}^{N-1}x[k]W_N^{mk}$$

$$=\sum_{k=0}^{N/2-1}x[k]W_N^{mk}+\sum_{k=0}^{N/2-1}x[k+N/2]W_N^{m(k+N/2)}$$

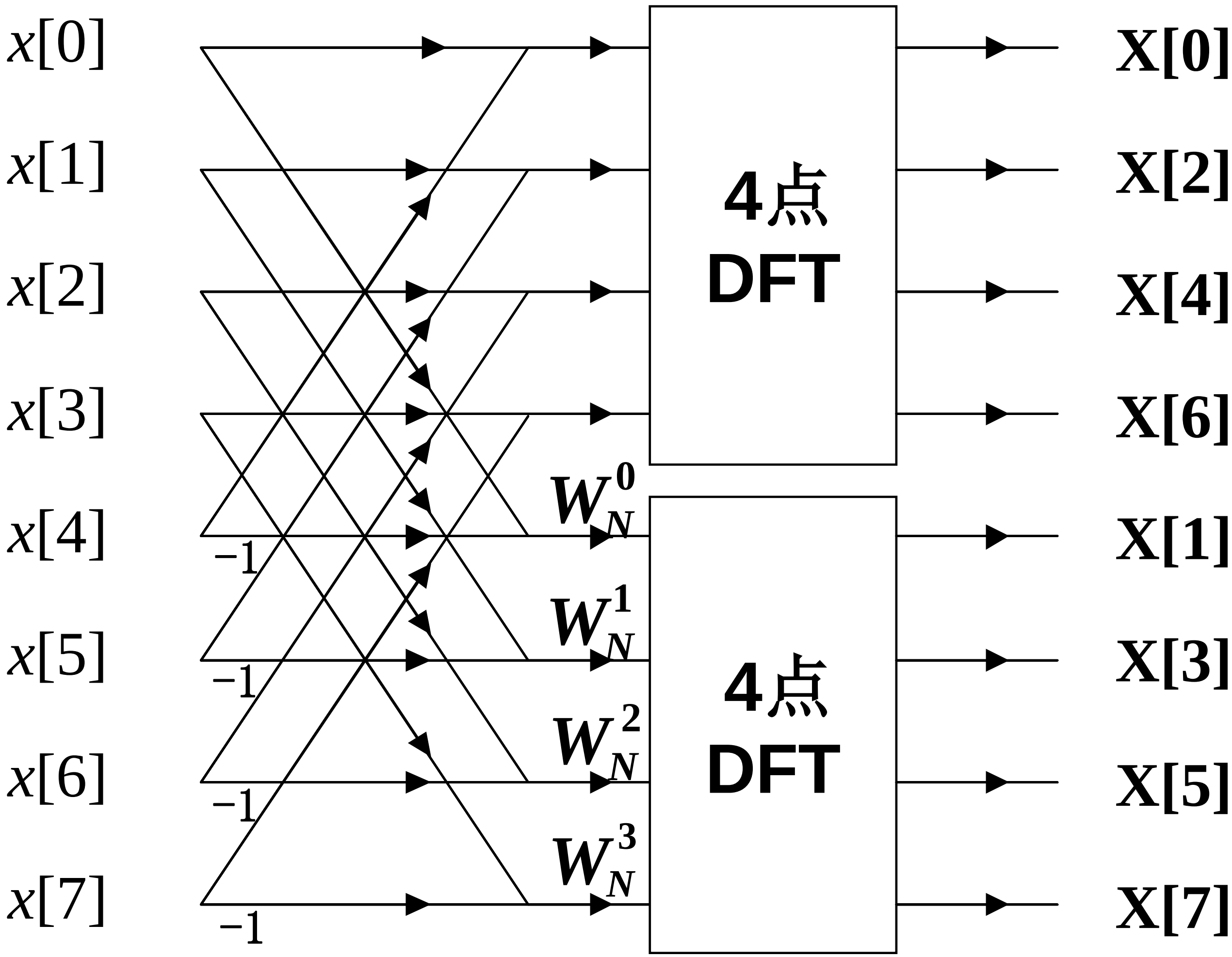
$$=\sum_{k=0}^{N/2-1}\left(x[k]+(-1)^m x[k+N/2]\right)W_N^{mk}$$

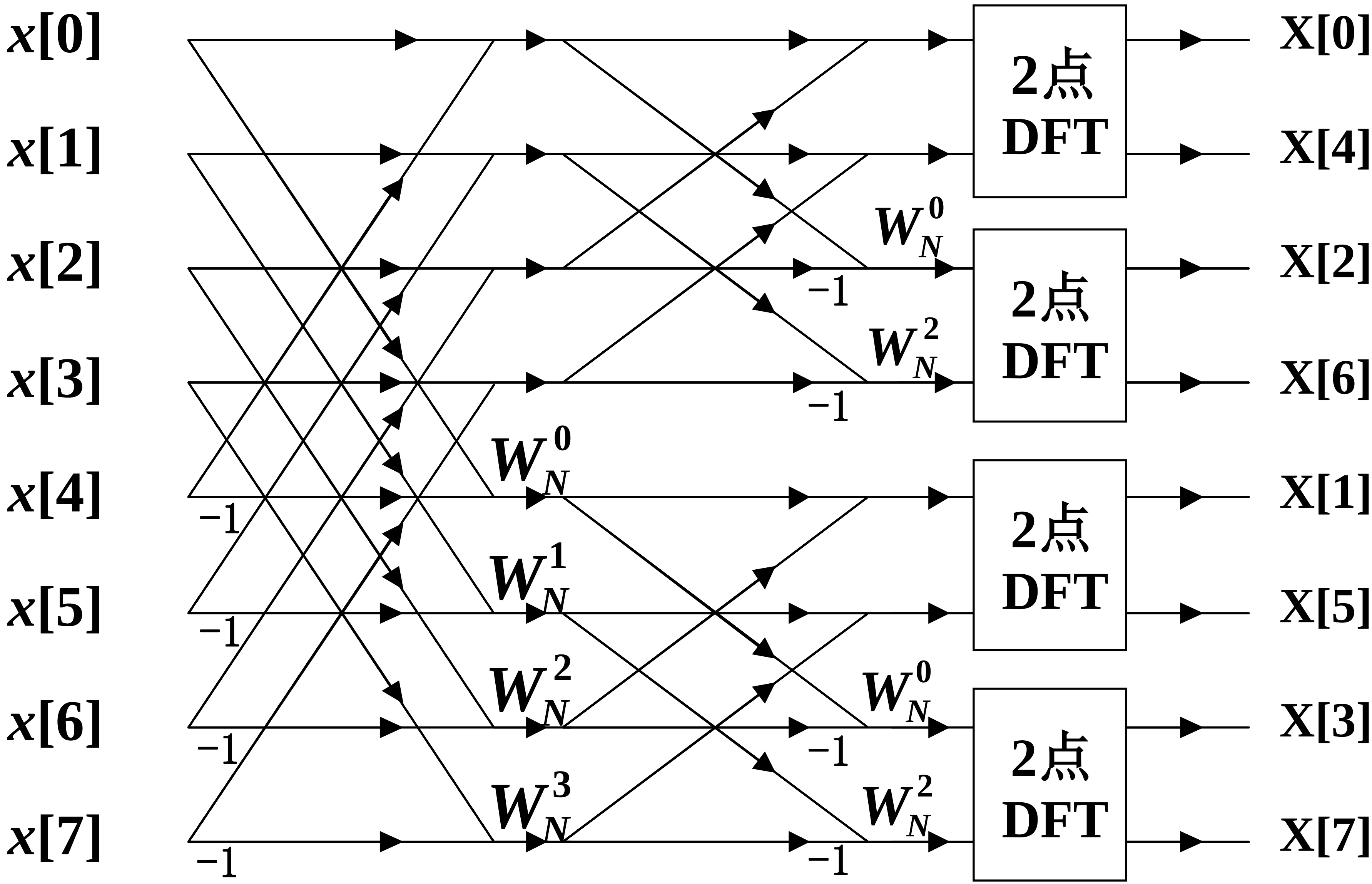
$$X[2r]=\sum_{k=0}^{N/2-1}\left(x[k]+x[k+N/2]\right)W_{N/2}^{rk}$$

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

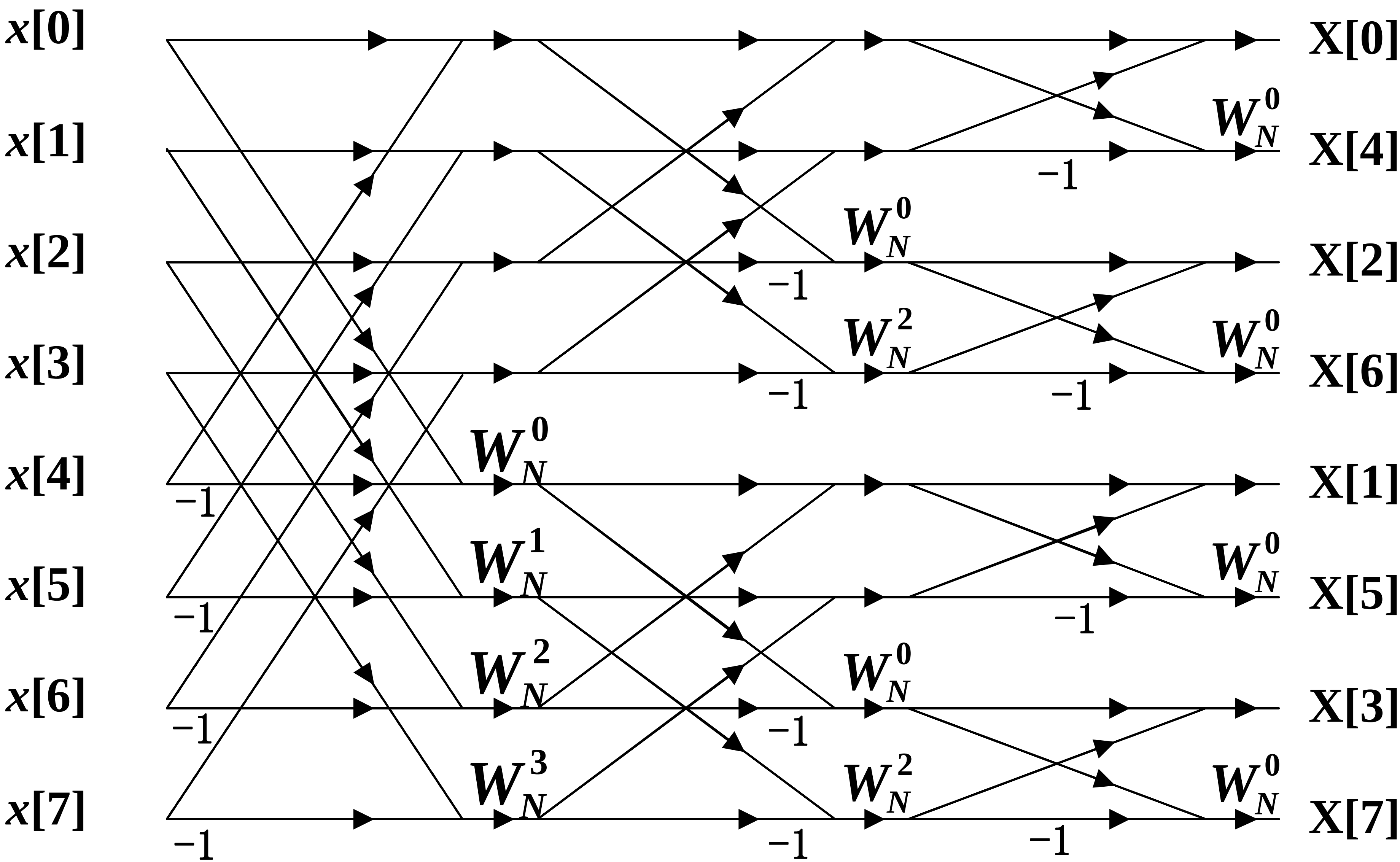
$$X[2r]=\sum_{k=0}^{N/2-1}\left(x[k]+x[k+N/2]\right)W_{N/2}^{rk} \qquad r=0,1\ldots N/2-1$$

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





x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]
−1
−1
−1
−1
$W_N^0$
$W_N^1$
$W_N^2$
$W_N^3$
−1
−1
−1
−1
$W_N^0$
$W_N^2$
$W_N^0$
$W_N^2$
2点 DFT
2点 DFT
2点 DFT
2点 DFT
X[0]
X[4]
X[2]
X[6]
X[1]
X[5]
X[3]
X[7]



x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]
X[0]
X[4]
X[2]
X[6]
X[1]
X[5]
X[3]
X[7]
$W_N^0$
$W_N^1$
$W_N^2$
$W_N^3$
$W_N^0$
$W_N^2$
$W_N^0$
$W_N^2$
$W_N^0$
$W_N^0$
$W_N^0$
$W_N^0$
−1

# FFT算法应用



- 利用*N*点复序列的FFT计算两个*N*点实序列FFT
- 利用*N*点复序列的FFT，计算2*N*点序列的FFT
- 利用FFT计算IFFT

# 利用N点复序列的FFT算法计算

## 两个N点实序列FFT

$x_1[k]$, $x_2[k]$是实序列，

将其构成复序列$y[k]=x_1[k]+j\,x_2[k]$

$$DFT\{x_1[k]\}=?$$

$$DFT\{x_2[k]\}=?$$

$$DFT\{x_1[k]+j\,x_2[k]\}=Y_R[m]+jY_I[m]$$

$$DFT\{x_1[k]-jx_2[k]\}=Y_R[(-m)_N]-jY_I[(-m)_N]$$

$$DFT\{x_1[k]\}=\frac{1}{2}\{Y_R[m]+Y_R[(-m)_N]+j(Y_I[m]-Y_I[(-m)_N])\}$$

$$DFT\{x_2[k]\}=\frac{1}{2j}\{Y_R[m]-Y_R[(-m)_N]+j(Y_I[m]+Y_I[(-m)_N])\}$$

# 利用$N$点复序列的FFT，计算2$N$点序列的FFT

$y[k]$是一个长度为2$N$的序列

$$y[k] \rightarrow \begin{cases} x_1[k] = y[2k] \\ x_2[k] = y[2k+1] \end{cases} \qquad k = 0,1,\cdots N-1$$

$$\begin{aligned} Y[m] &= X_1[m] + W_{2N}^{m} X_2[m] \\ Y[m+N] &= X_1[m] - W_{2N}^{m} X_2[m] \end{aligned} \qquad m = 0,1,\ldots,N-1$$

问题：如何**利用$N$点FFT，计算4$N$点序列的FFT**？

# 利用FFT实现IFFT

$$X[m] = DFT\{x[k]\} = \sum_{k=0}^{N-1} x[k]W_N^{mk}$$

$$x[k] = IDFT\{X[m]\} = \frac{1}{N}\sum_{m=0}^{N-1} X[m]W_N^{-mk}$$

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

步骤：*A*) 将$X[m]$取共轭

*B*) 用*FFT*流图计算$DFT\{X^*[m]\}$

*C*) 对*B*)中结果取共轭并除以$N$

# 思考题：

二维离散傅立叶变换如何快速实现?
傅立叶变换具有哪些性质?
傅立叶变换和卷积运算有何关系?