Fast Fourier Transform

$$x[n] = \{\underset{\uparrow}{1}, 3, 2, 5, 3, 4, 2, 6\}$$

$$\underline{N = 8}$$

$$X(0) = \sum_{n=0}^{7} x[n] =$$

$$X(1) = \sum_{n=0}^{7} x[n] e^{-j\frac{\pi}{4}n} = x(0) + x(1)\left[\cos\frac{\pi}{4} - j\sin\frac{\pi}{4}\right) + x(2)\left[\cos\frac{\pi}{2} - j\sin\frac{\pi}{2}\right) + x(3)\left[\cos\frac{3\pi}{4} - j\sin\frac{3\pi}{4}\right]$$

$$+ x(4)$$

$$X(7) = \sum_{n=0}^{7} x(n) e^{-j\frac{7\pi}{4}n} = x(0) + x(1)\left[\cos\frac{7\pi}{4} - j\sin\frac{7\pi}{4}\right] + x(2)\left(\cos\frac{7\pi}{2} - j\sin\frac{7\pi}{2}\right) + x(3)\left[\cos\frac{21\pi}{4} - j\sin\frac{21\pi}{4}\right]$$

N^2 It becomes complex with the in N

FFT(DFT)

- No. of Complex number computation increases with increase in N.
- DFT requires N^2 Complex multiplications and N(N-1) Complex additions
- Efficient computation of DFT is FFT as it requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ Complex additions.
- Exploits the symmetry of the DFT calculation to make its execution much faster
- Speedup increases with DFT size

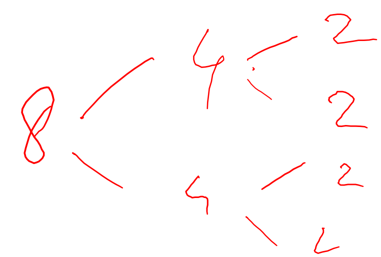
- Discoveries :

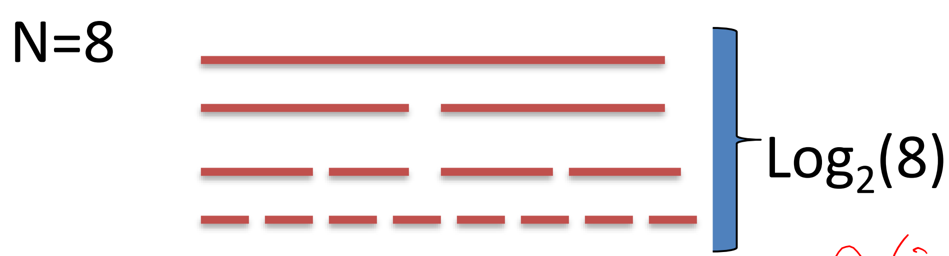
  1965 - algorithm rediscovered (not for the first time) by Cooley and Tukey

- In 1967, calculation of a 8192-point DFT on the top-of-the line IBM 7094 took ....
  - ~30 minutes using conventional techniques
  - ~5 seconds using FFTs

Radix p computation

- split the sum into ‘p’ subsequences of length N/p
- continue until you have N/p subsequences of length p





$N=8 \Rightarrow$ 3 stage

Properties to remember

$e^{-j\frac{2\pi}{N}nk} \, e^{-j\frac{\cancel{2}\pi}{\cancel{N}}\frac{\cancel{N}}{\cancel{2}}}$

- Symmetry property

$$W_N^{\,nk+N/2} = -W_N^{\,nk}$$

$$X(N\text{-}k) = X^*(k) = X(\text{-}k)$$

- Periodicity Property

$$W_N^{\,nk+N} = W_N^{\,nk}$$

Radix-2 FFT Algorithm $N = 2^m$

- Let's take a simple example where only two points are given N=2. (i.e.) n=0, n=1;

$k = 0, 1$

$x(n) \to N$

$X(k) \to N$

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

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

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

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

$$k = 0; X(0) = x(0) + x(1)$$

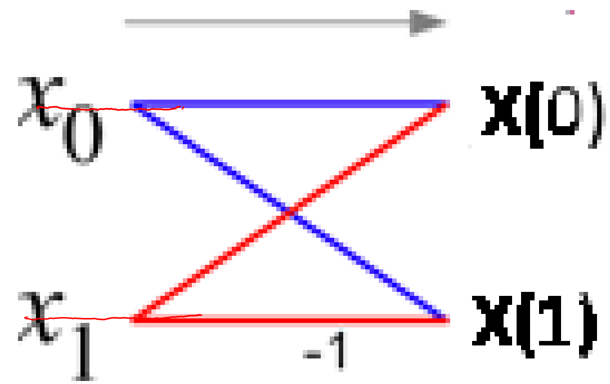
$$k = 1; X(1) = x(0) - x(1)$$

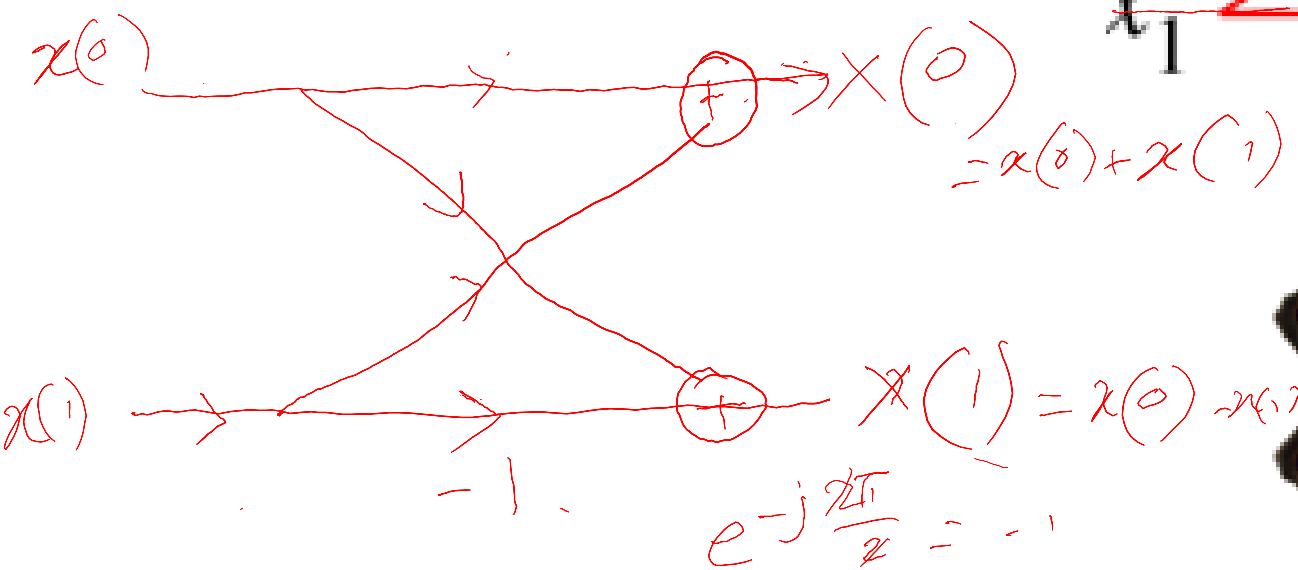
Radix-2 FFT Algorithm

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

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

Butterfly FFT





Linear computation of Radix-2 Decimation in Time

- First break x[n] into even and odd

$$X(k) = \sum_{n=even} x(n) W_N^{nk} + \sum_{n=odd} x(n) W_N^{nk}$$

- Let n=2m for even and n=2m+1 for odd

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

- Even and odd parts are both DFT of a N/2 point sequence

$$W_N^{2mk} = W_{\frac{N}{2}}^{mk}$$

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

$$X[k=0] = \sum_{m=0}^{0} W_1^{0.0} x[0] + W_2^{0} (\sum_{m=0}^{0} W_1^{0.0} x[1])$$

$$= x[0] + x[1]$$

$$X[k=1] = \sum_{m=0}^{0} W_1^{0.1} x[0] + W_2^{1} (\sum_{m=0}^{0} W_1^{0.1} x[1])$$

$$= x[0] + W_2^{1} x[1] = x[0] - x[1]$$

DIT continued

- Therefore if $N=2^m$

  - N-point DFT → two N/2 – point DFT
  - N/2-point DFT → two N/4 – point DFT
  - N/4-point DFT → two N/8 – point DFT
  - ……
  - two 2-point DFT

- 2-point DFT is the smallest unit which cannot be decomposed further

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

$$W_N^{\ 2mk} = W_{N/2}^{\ \ mk}$$

$$X(k) = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km}$$

$$k = 0, 1, \ldots \frac{N}{2} - 1$$

$$= F_1(k) + W_N^k F_2(k)$$

$$X(k) = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km}$$

even

$$= F_1(k) + W_N^k F_2(k)$$

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

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k)$$

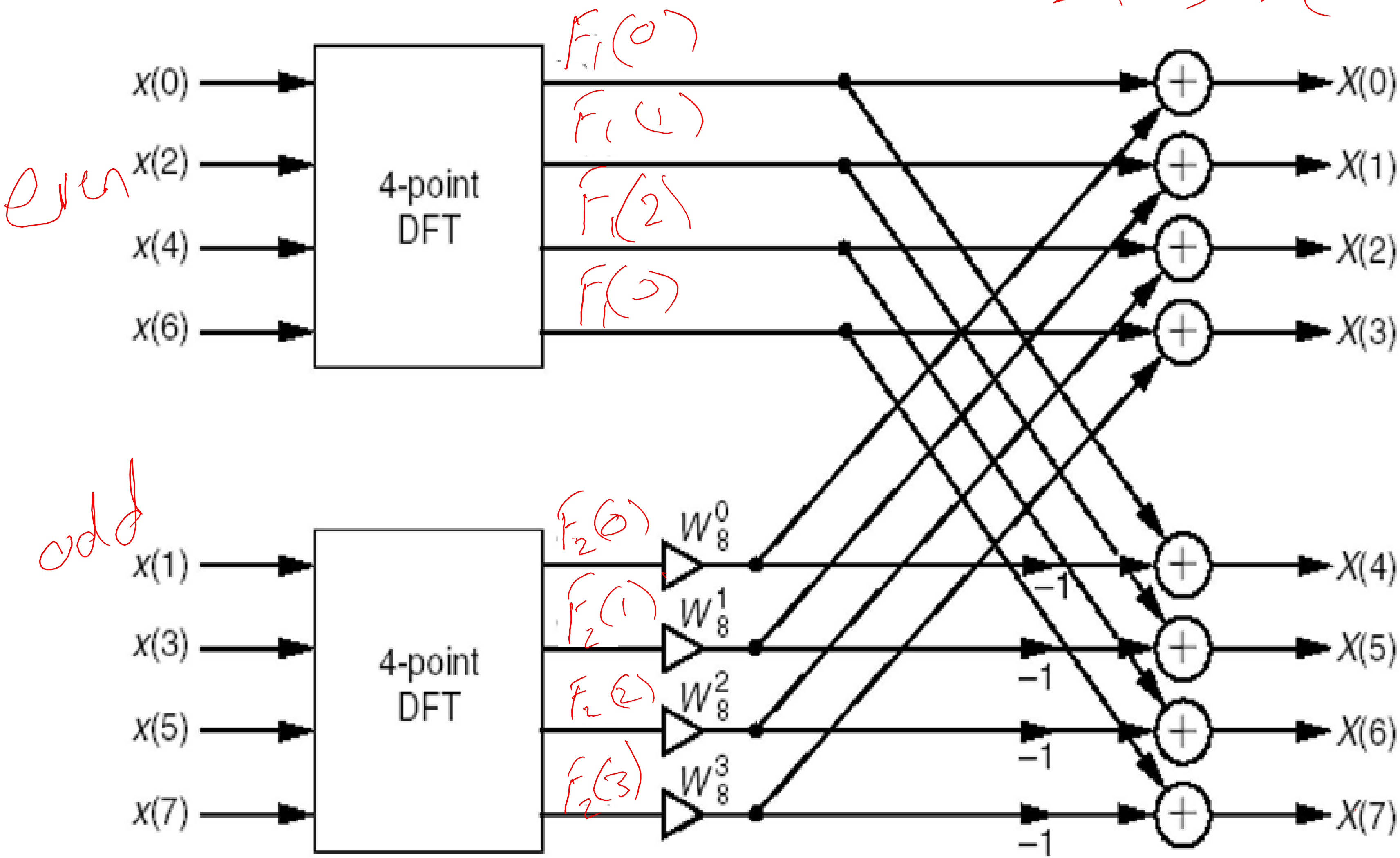
odd

F_1 (k) and F_2(k) are periodic with period N/2 .So,

$F_1(k+N/2) = F_1(k)$ and $F_2(k+N/2) = F_2(k)$; $W_N^{(k+N/2)} = - W_N^k$

$N = 8 \Rightarrow 2(N=4)$

$N = 4 \Rightarrow 2(N=2)$



$\underset{\text{even}}{F_1(k)} + W_N^k \underset{\text{odd}}{F_2(k)}$

$$G_1(k) = F_1(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$G_2(k) = W_N^k F_2(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X(k) = G_1(k) + G_2(k) \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X(k + \frac{N}{2}) = G_1(k) - G_2(k) \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

This can be further decomposed by factor of 2

$$v_{11}(n) = f_1(2n) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1$$

$$v_{12}(n) = f_1(2n+1) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1$$

and $f_2(n)$ would yield

$$v_{21}(n) = f_2(2n) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1$$

$$v_{22}(n) = f_2(2n+1) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1$$

- By computing N/4 DFTs we obtain N/2 DFTs

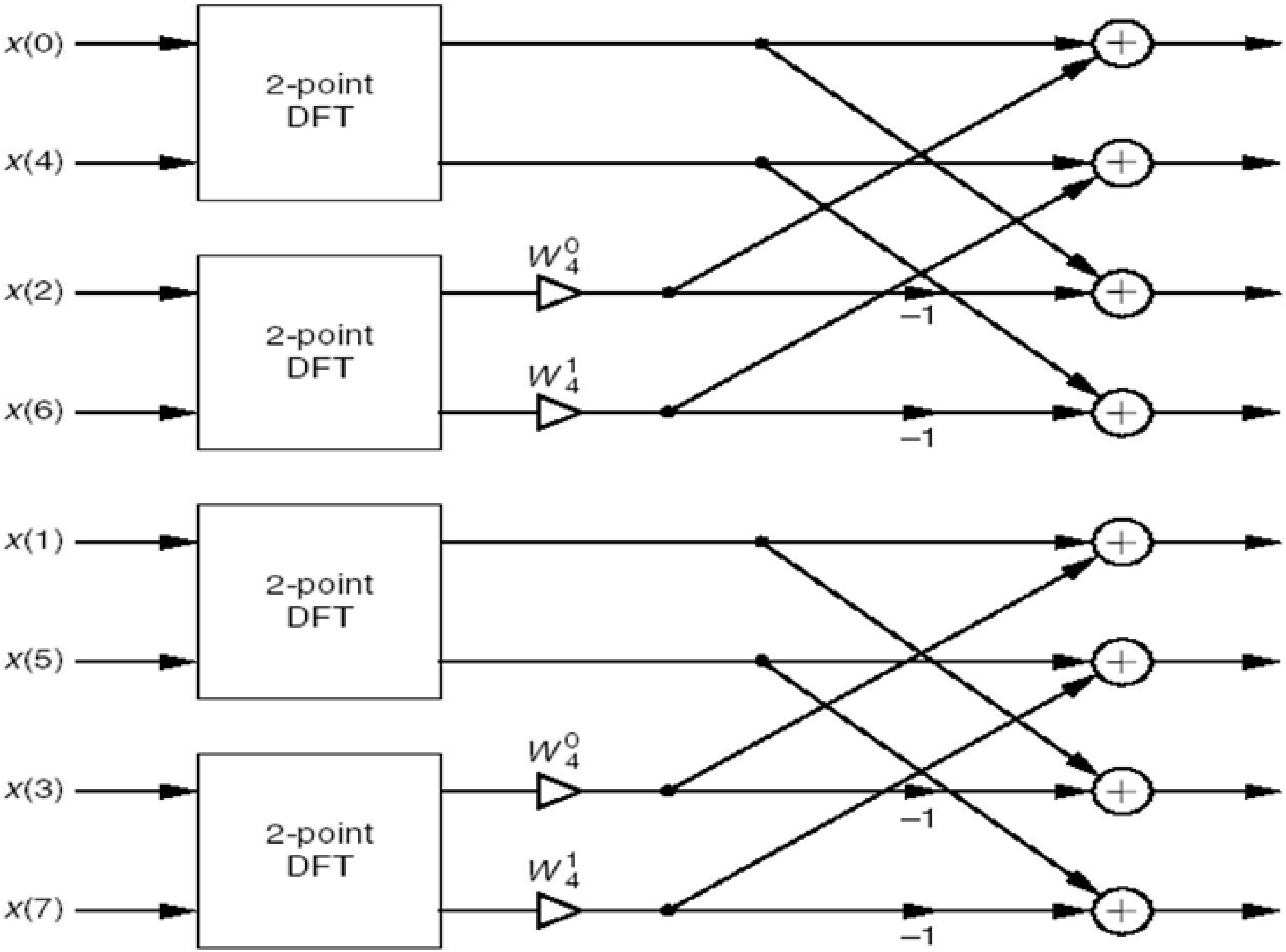
$$F_1(k) = V_{11}(k) + W_{N/2}^{k} V_{12}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1$$

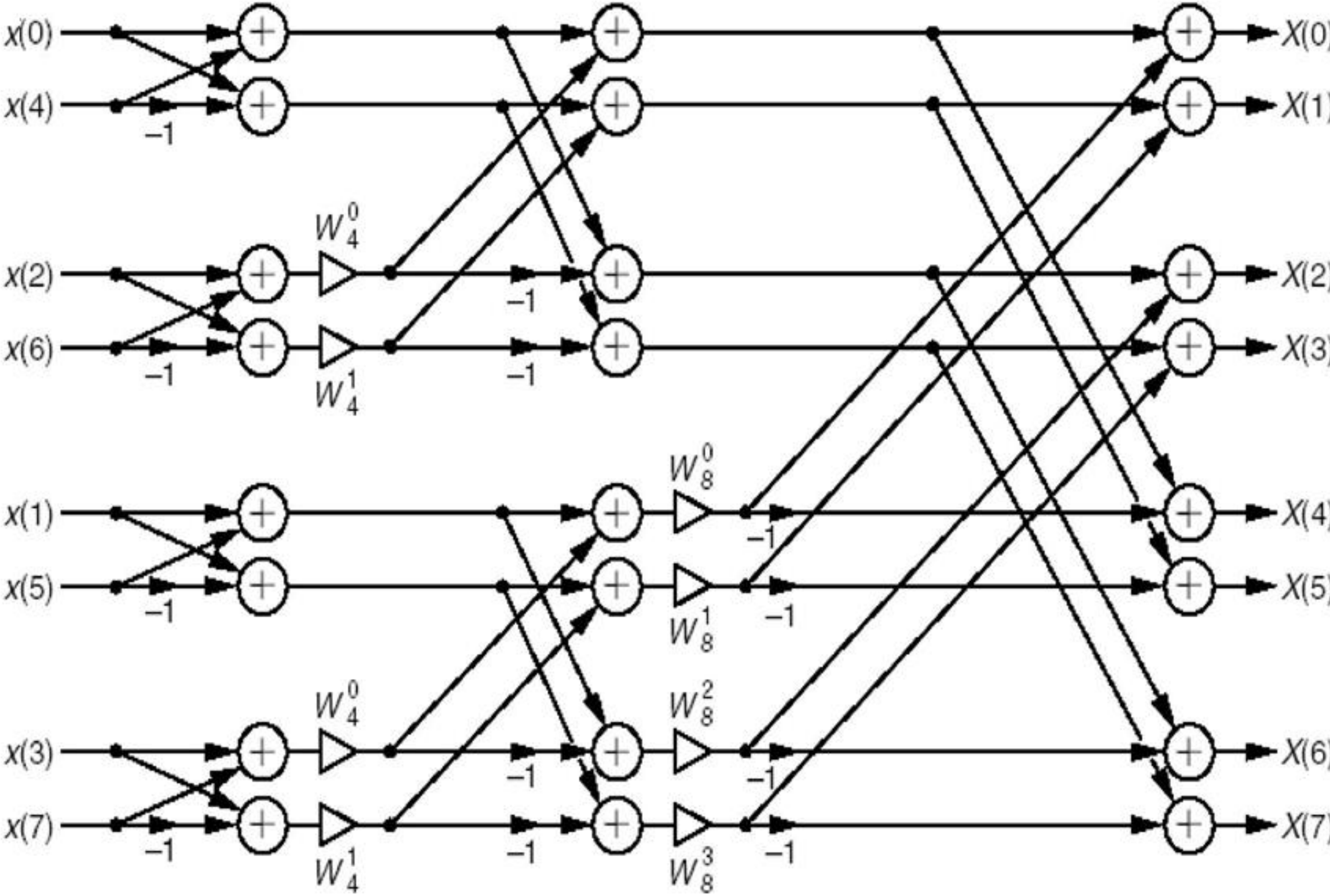
$$F_2(k) = V_{21}(k) + W_{N/2}^{k} V_{22}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1$$

$$F_1\left(k + \frac{N}{4}\right) = V_{11}(k) - W_{N/2}^{k} V_{12}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1$$

$$F_2\left(k + \frac{N}{4}\right) = V_{21}(k) - W_{N/2}^{k} V_{22}(k) \qquad k = 0, \ldots, \frac{N}{4} - 1$$

- The decimation of the data sequence can be repeated again and again until the resulting sequence reduces to one point sequence


x(0)
x(4)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
2-point DFT
2-point DFT
2-point DFT
2-point DFT
W_4^0
W_4^1
W_4^0
W_4^1
−1
−1
−1
−1


x(0)
x(4)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
−1
W_4^0
W_4^1
W_8^0
W_8^1
W_8^2
W_8^3
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)

Computation efficiency

- For $N = 2^m$, this decimation can be performed in $m=\log_2 N$ stages

$m = \log_2 8 \longrightarrow \log_2 2^3 \Rightarrow \underline{\underline{3}}$

- Thus the total number of complex multiplication is reduced to $(N/2) \log_2 N$

N^2

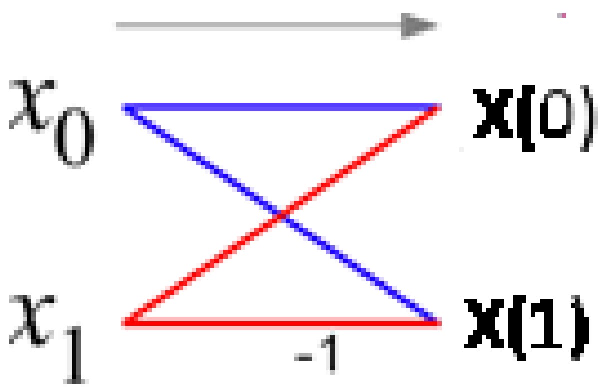
- The number of complex addition is $N\log_2 N$

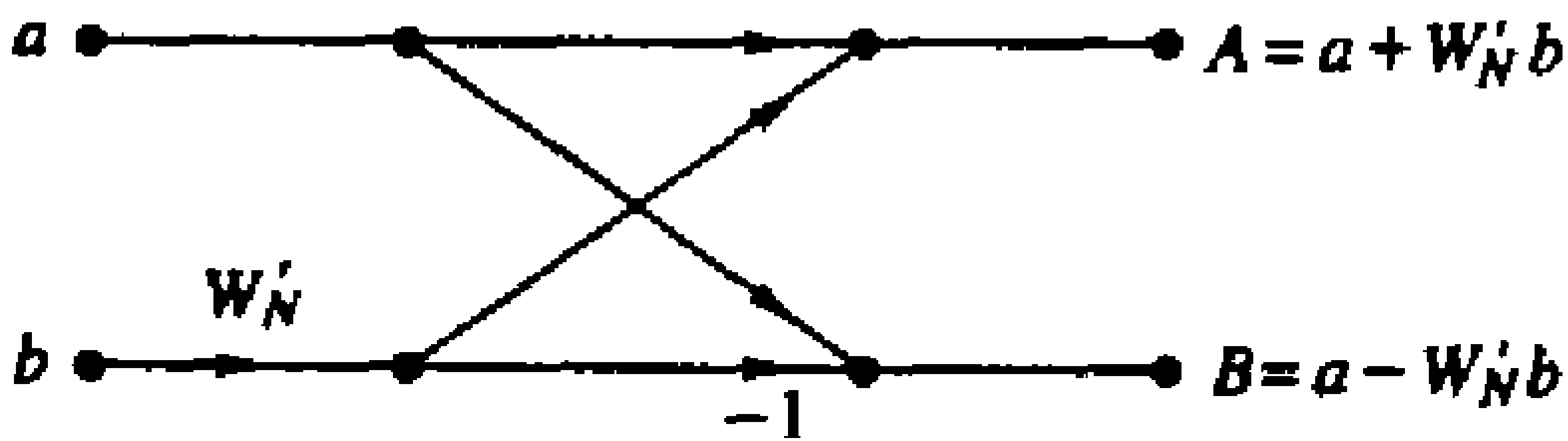
$N(N-1)$

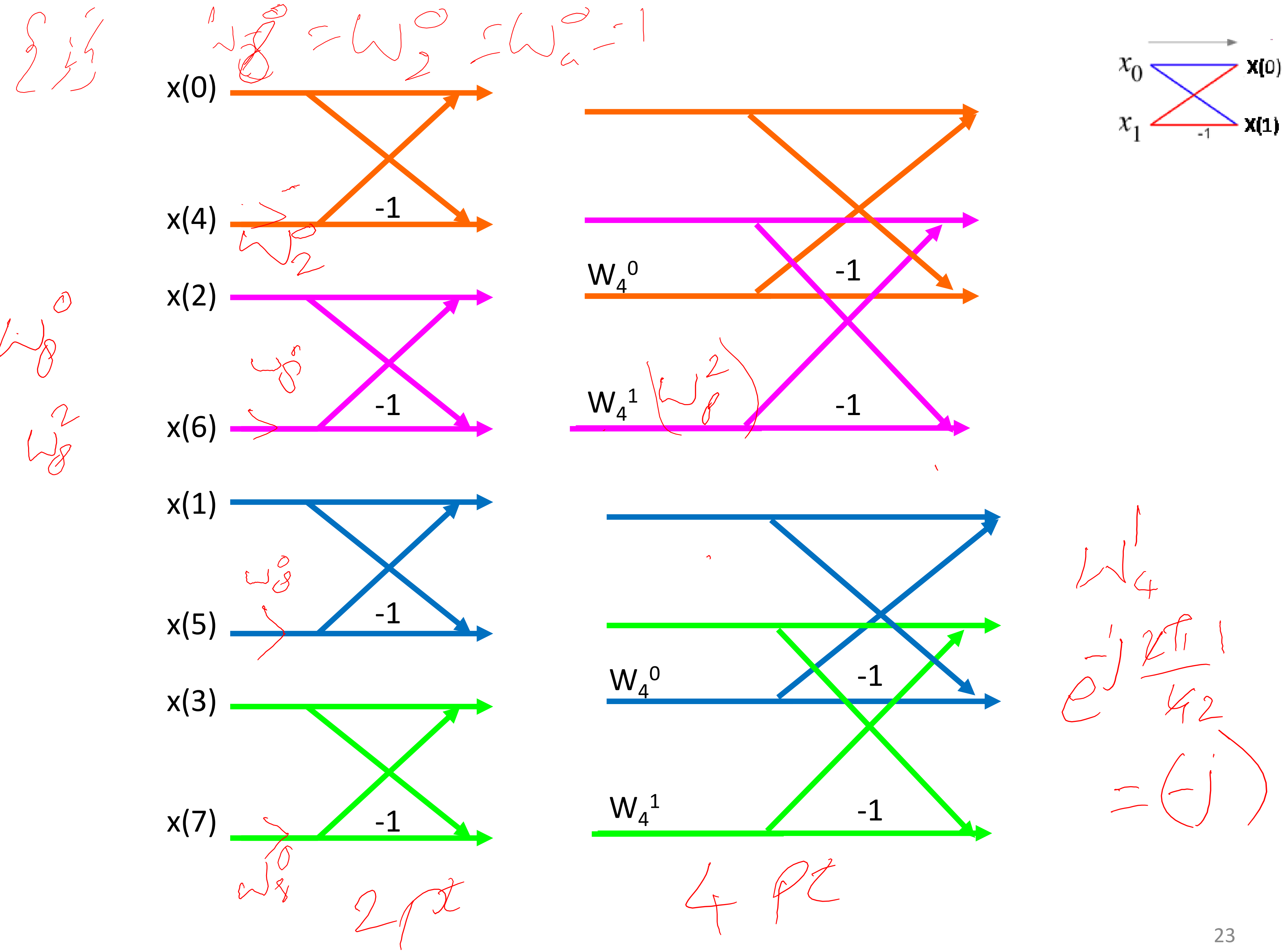
Comparison of computational complexity b/w DFT and FFT

Number of Points, N	Complex Multiplications in Direct Computation, N^2	Complex Multiplications in FFT Algorithm, $(N/2)\log_2 N$	Speed Improvement Factor
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1,024	80	12.8
64	4,096	192	21.3
128	16,384	448	36.6
256	65,536	1,024	64.0
512	262,144	2,304	113.8
1,024	1,048,576	5,120	204.8

Butterfly computation in FFT(DIT)





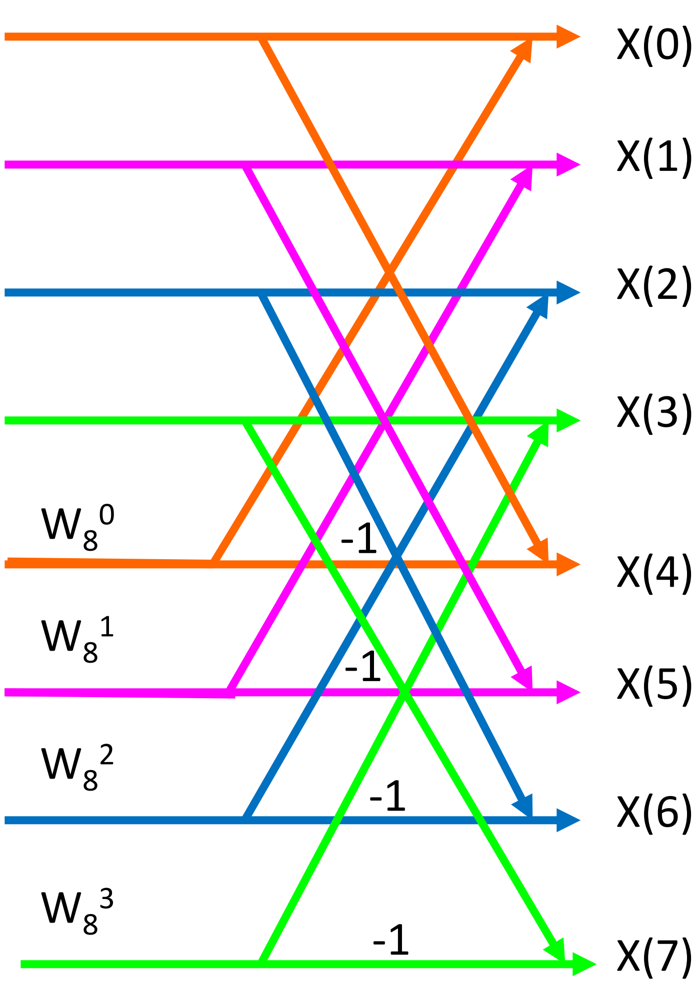

x(0)
x(4)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
-1
W4 0
W4 1
x0
x1
X(0)
X(1)
2 pt
4 pt

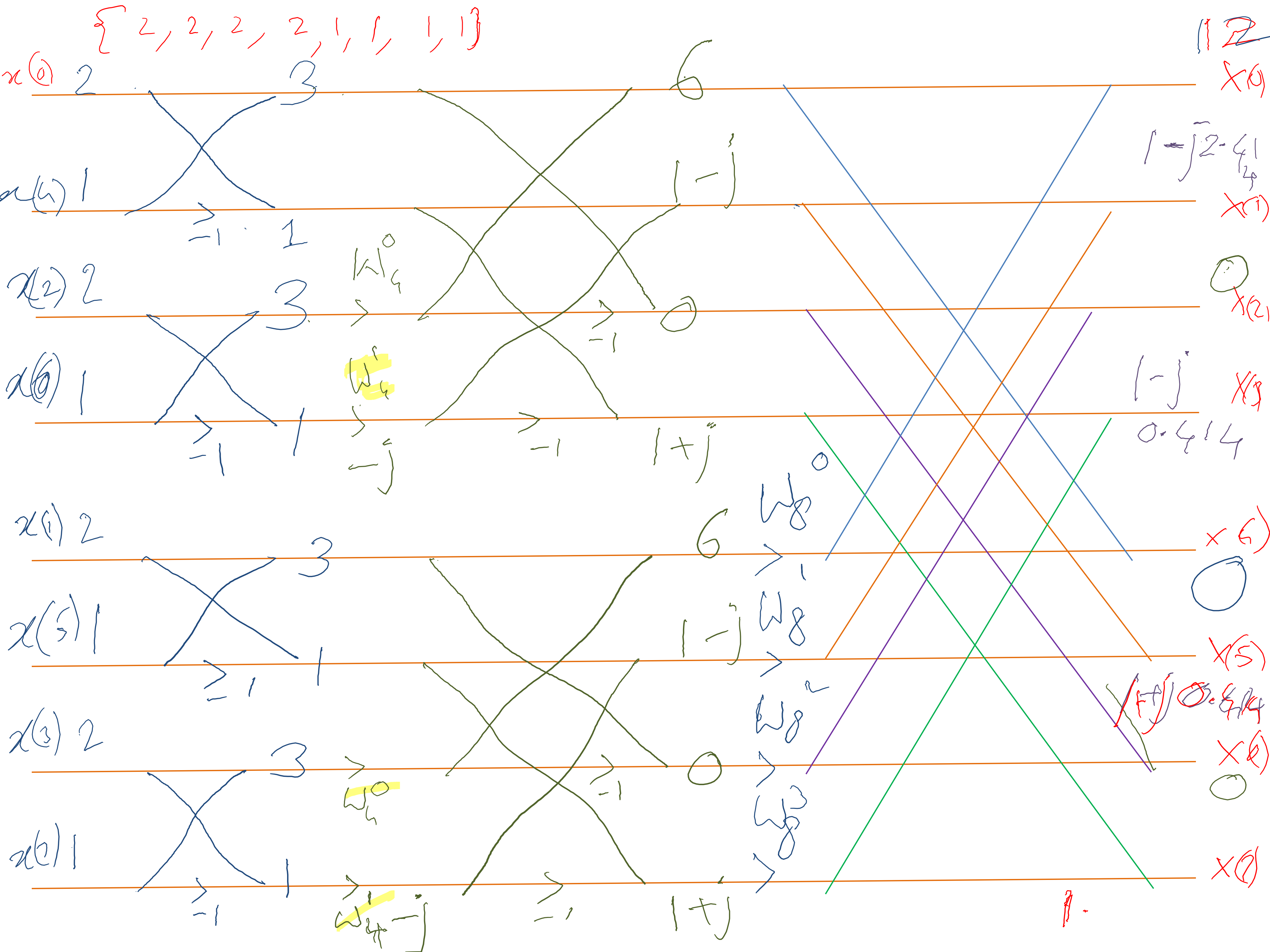
$$W_8^3 = e^{-j}$$
$$= -\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$$
$$= -0.707 - j0.707$$

$$W_8^0 = 1$$

$$W_8^1 = e^{-j\frac{2\pi}{8}}$$
$$= \frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$$
$$= 0.707 - j0.707$$

$$W_8^2 = -j$$





{2, 2, 2, 2, 1, 1, 1, 1}
x(0) 2
x(4) 1
x(2) 2
x(6) 1
x(1) 2
x(5) 1
x(3) 2
x(7) 1
3
1
-1
6
1-j
0
1+j
W_4^0
W_4^1
-j
W_8^0
W_8^1
W_8^2
W_8^3
12
1-j2.414
0
1-j
0.414
1+j 0.414
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)

$X(0) = 12$

$X(1) = 1 - j2.414$

$X(2) = 0$

$X(3) = 1 - j0.414$

$X(4) = 0$

$X(5) = 1 + j0.414$

$X(6) = 0$

$X(7) = 1 + j2.414$

$X(k) = X^{*}(N-k)$

$X(1) = X^{*}(7)$

$X(3) = X^{*}(5)$

$X(2) = X^{*}(6)$

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

Decimation in Frequency

$N = 8$

$x(0)$
$x(1)$
$x(2)$
$x(3)$

$x(4)$
$x(5)$
$x(6)$
$x(7)$

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

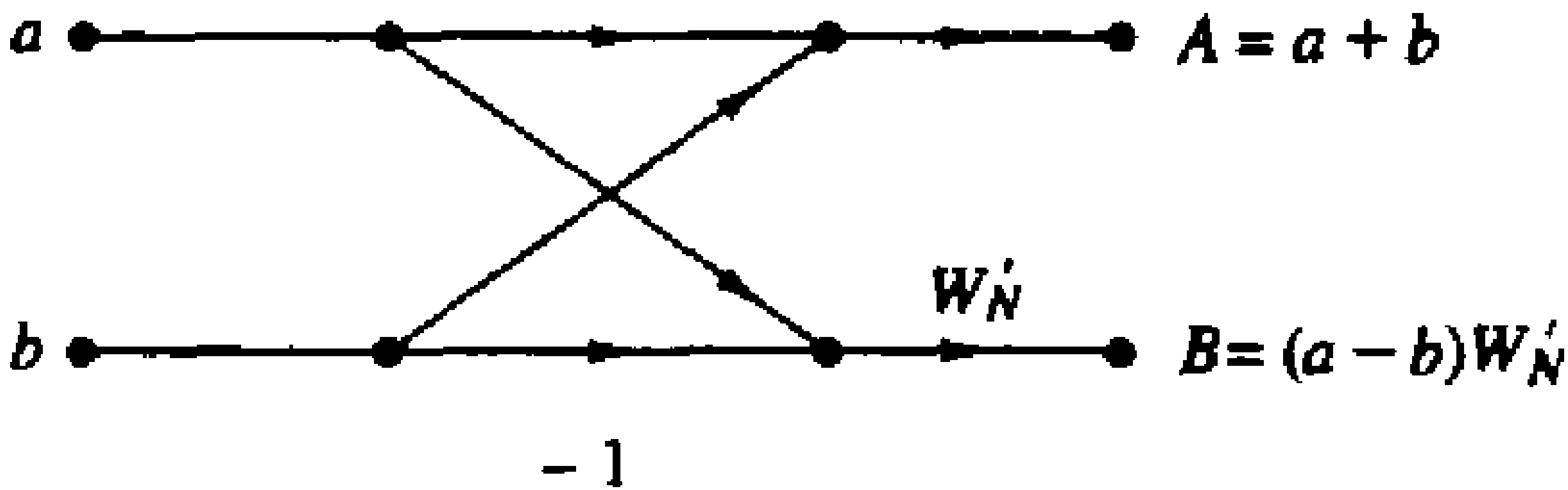
3 7 4

$$= \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{(N/2)-1} x\left(n + \frac{N}{2}\right) W_N^{kn}$$

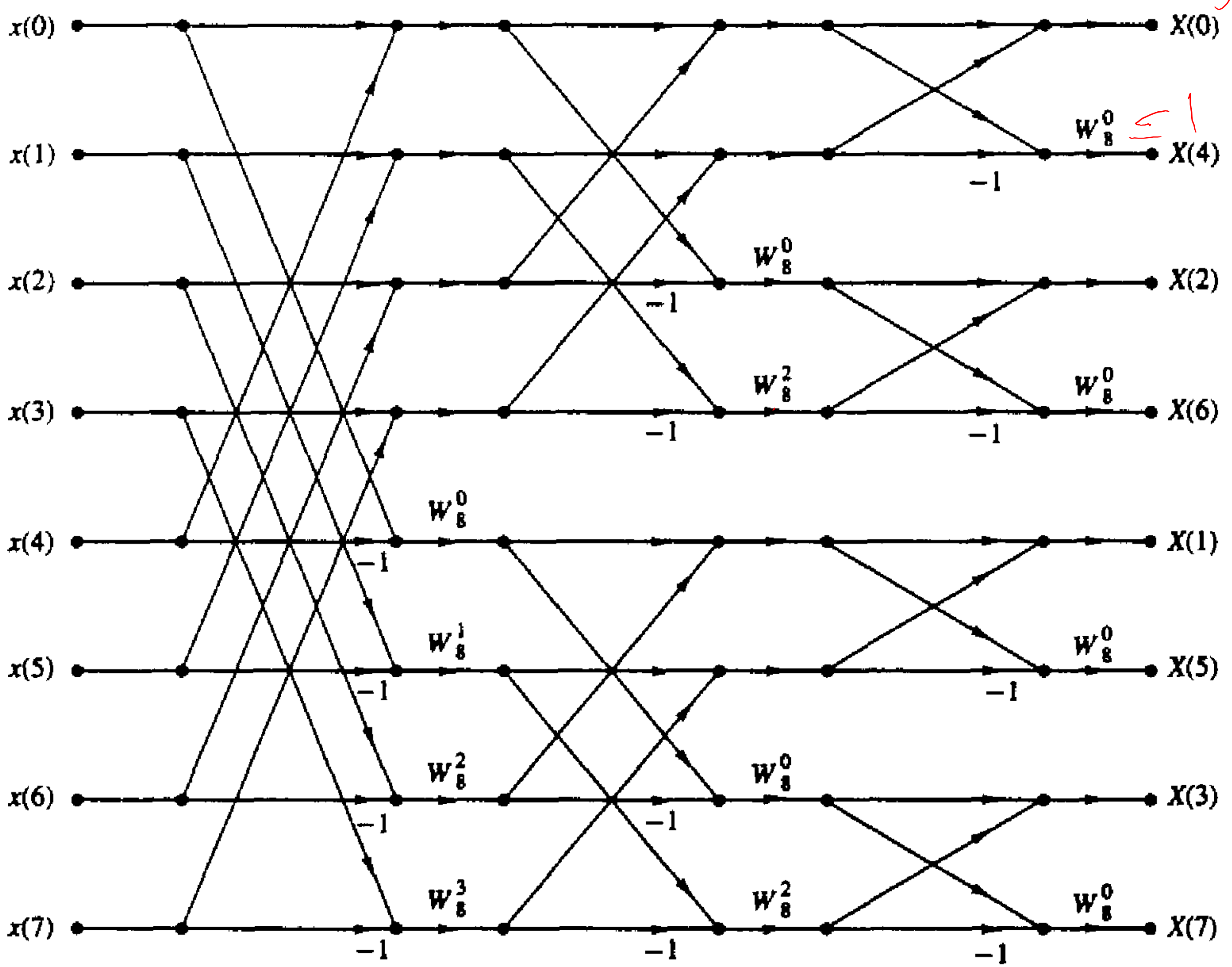
$$W_N^{kN/2} = (-1)^k,$$

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

Butterfly computation



Bit reversed



IFFT

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} = \sum x(n) W_N^{nk}$$

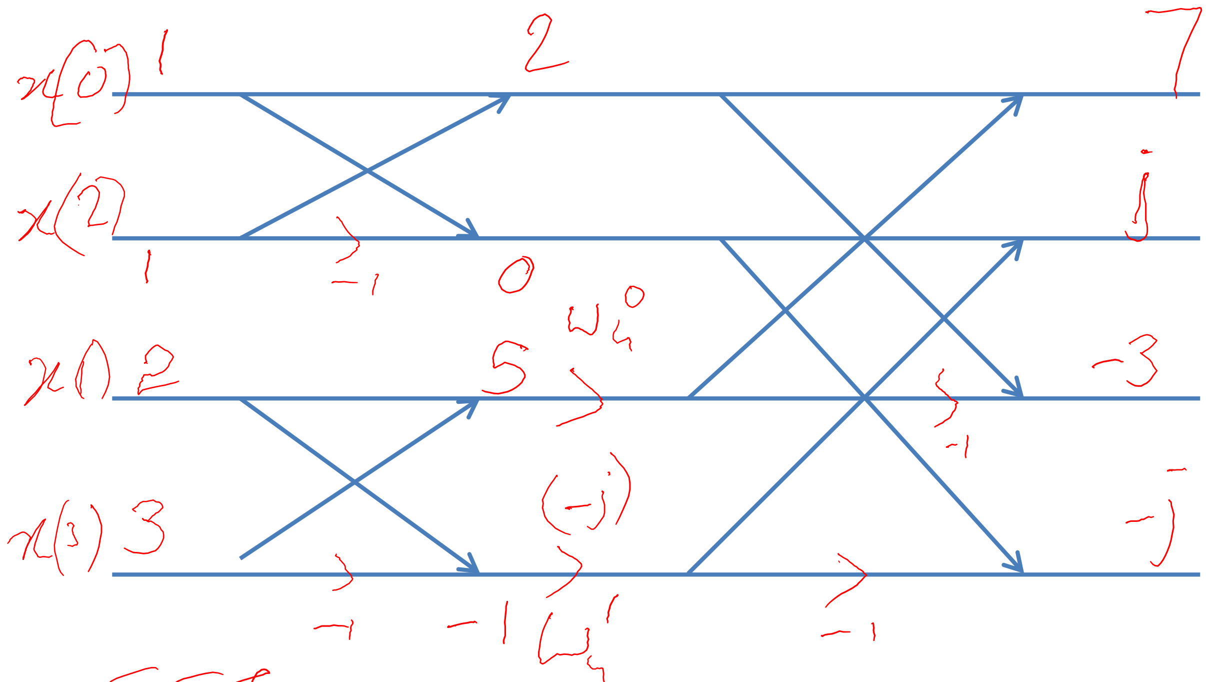
$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}nk} = \frac{1}{N}\sum X(k) W_N^{-nk}$$

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

$$[AB]^* = (A^* B)^*$$

FFT

DIT



$x(n)\{1, 2, 1, 3\}$

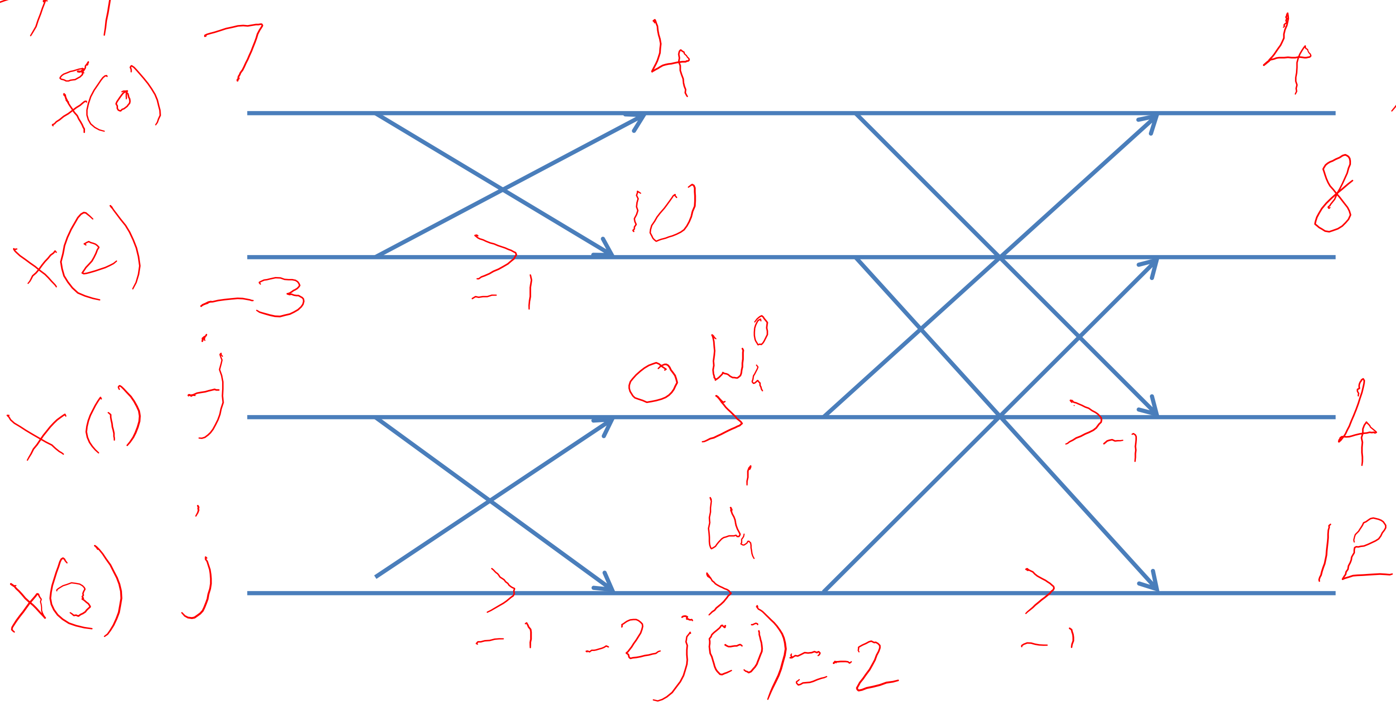
$X(0)$

$X(1)$ $X(k) = \{7, j, -3, -j\}$

$X(2)$ $X^*(k) = \{7, -j, -3, j\}$

$X(3)$

IFFT



$1/N = 1$

2

1

3