Lecture 4:
Fast Fourier Transform

Prof. Tai-kyong Song
Dept. of Electronic Engineering
SOGANG UNIVERSITY

Introduction

N-point DFT

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

where $W_N = \exp(-j2\pi/N)$

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

Computational Advantage of FFT

	# of Multiplications	# of Addidtions
N-point DFT	N^2	N(N-1)
N-point FFT	(N/2) $\log_2$N	N $\log_2$N

Note) Multiplications and addtions are complex operations here.

Ex) N = 1024

➔ **1M mul**

➔ **5K mul**

$\times \frac{1}{200}$

Introduction

Basic Idea of DFT

Rather than directly computing DFT of a sequence of length N, computation of DFT is decomposed into successively smaller DFT's until one gets m stage ($N = b^m$) b-point DFT's. Then, the results are added.

Also, we use the following properties:

1) $W_N^{k(N-n)} = \left(W_N^{kn}\right)^*$: Symmetric property

2) $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$: Periodicity

→ **Radix b N-FFT**

Divide and Conquer method

Introduction

Depending on how decomposition is made,

two basic classes of FFT exist:

1) Decimation-In-Time (DIT) FFT:

Time sequence x(n) is decomposed into successively smaller sequence.

2) Decimation-In-Frequency (DIF) FFT:

Sequence of DFT X(k) is decomposed into successively smaller sequence.

Note) Here, we consider only the special (and most commonly used) case of N, being an integer power of 2 (Radix 2), i.e., $N = 2^m$

Radix 2 DIT FFT

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

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

$$+ \sum_{n=0}^{N-1} x(n) W_N^{nk} \underline{(n = odd)}$$

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

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

(N/2 - point DFT   N/2 - point DFT)

$$\boxed{\because W_N^2 = (e^{-j2\pi/N})^2 = e^{-j2\pi/N/2} = W_{N/2}^1}$$

$\rightarrow X(k) = G(k) + W_N^k H(k)$ (A)

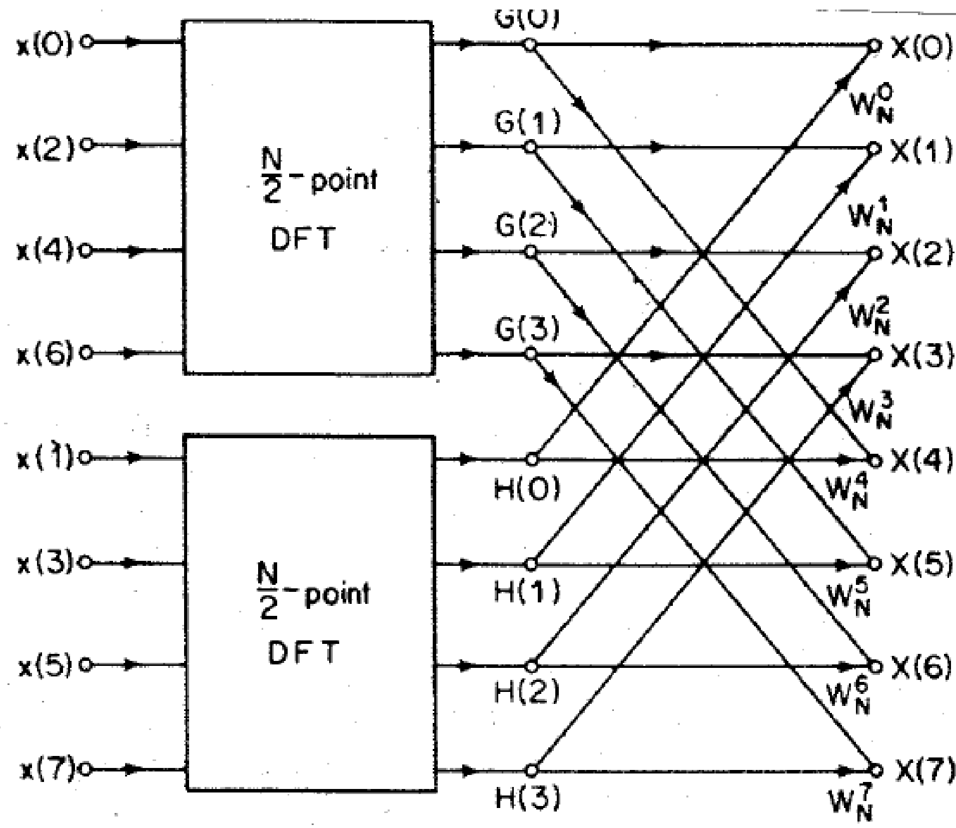
where G(k) and H(k) are periodic function with period of N/2.

\# complex multiplications = # complex additions

$= N + 2(N/2)^2$ ($< N^2$ for $N > 2$)

Radix 2 DIT FFT

Flow graph of DIT decomposition of 8-point DFT computation into two 4-point DFT's. (Note that H(4) = H(0), G(4) = G(0), H(5) = H(1), G(5) = G(1), ……)



$$X(k) = G(k) + \underline{W_N^k} H(k)$$

$$\boldsymbol{k = 0, 1, \ldots, N-1 (= 7)}$$

Radix 2 DIT FFT

Successive Decomposition

The above procedure of decomposition is continued until it gets down to 2-point DFT. Decomposing G(k) and H(k) into two N/4 even points and N/4 odd points

$$G(k) = \sum_{r=0}^{N/2-1} g(r) W_{N/2}^{rk}$$

$$= \sum_{l=0}^{N/4-1} g(2l) W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} g(2l+1) W_{N/4}^{lk}$$

$$H(k) = \sum_{l=0}^{N/4-1} h(2l) W_{N/4}^{lk}$$

$$+ W_{N/2}^{k} \sum_{l=0}^{N/4-1} h(2l+1) W_{N/4}^{lk}$$

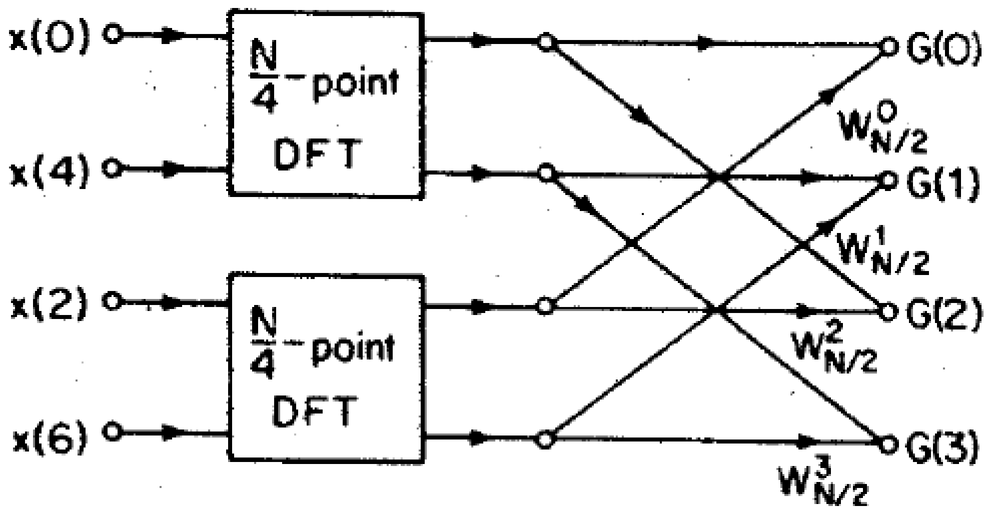
complex multiplications = # complex additions

$= \mathrm{N} + 2\{\mathrm{N}/2 + 2(\mathrm{N}/4)^2\} = \mathrm{N} + \mathrm{N} + 4(\mathrm{N}/4)^2$

Note: $W_{N/2}^{k} = W_{N}^{2k}$

Radix 2 DIT FFT

Example: 8-point DIT FFT



$$G(k) = \sum_{r=0}^{N/2-1} g(r) W_{N/2}^{rk}$$

$$= \sum_{l=0}^{N/4-1} g(2l) W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} g(2l+1) W_{N/4}^{lk}$$

$$\boldsymbol{k = 0, 1, \dots, N/2 - 1 (= 3)}$$

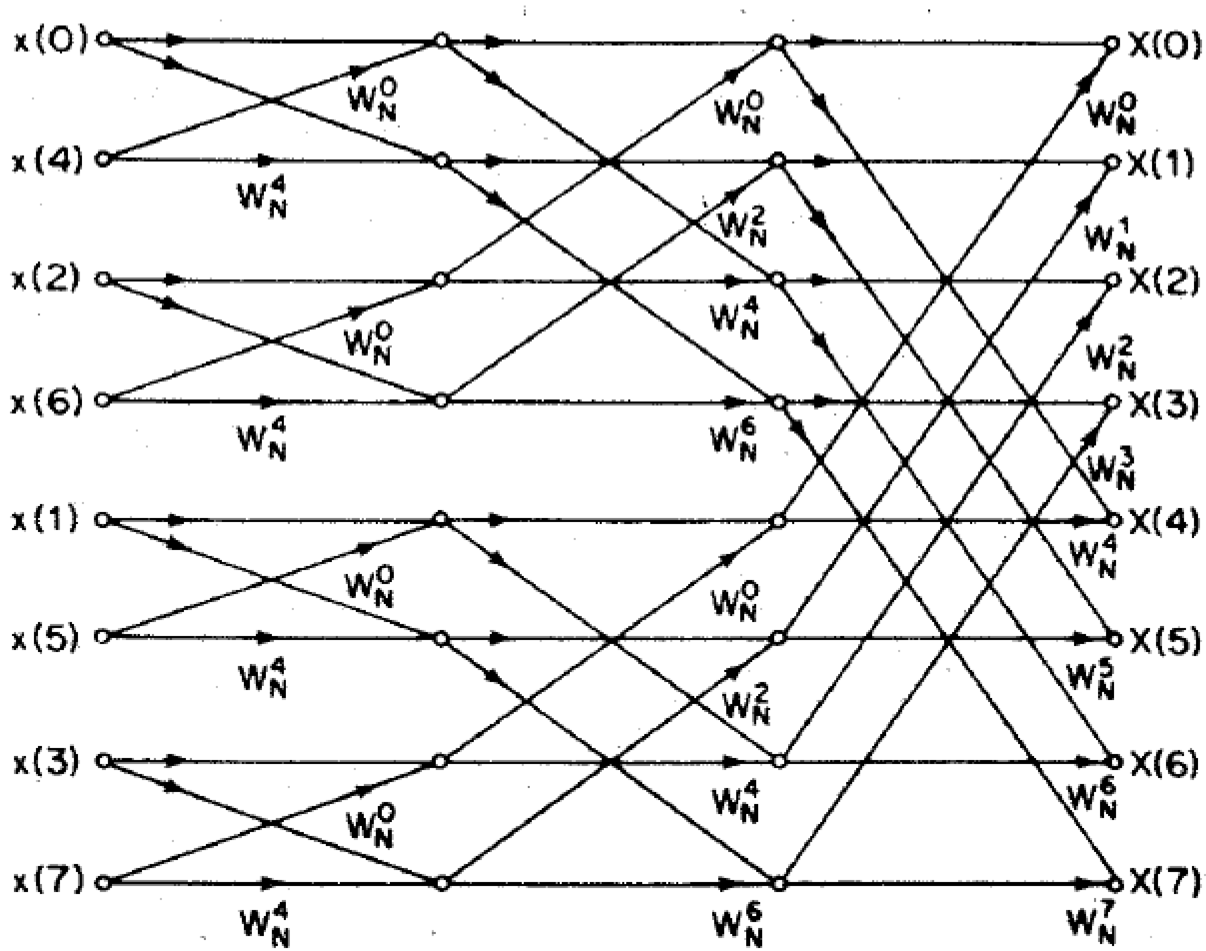
$$H(k) = \sum_{l=0}^{N/4-1} h(2l) W_{N/4}^{lk}$$

$$+ W_{N/2}^{k} \sum_{l=0}^{N/4-1} h(2l+1) W_{N/4}^{lk}$$

Radix 2 DIT FFT

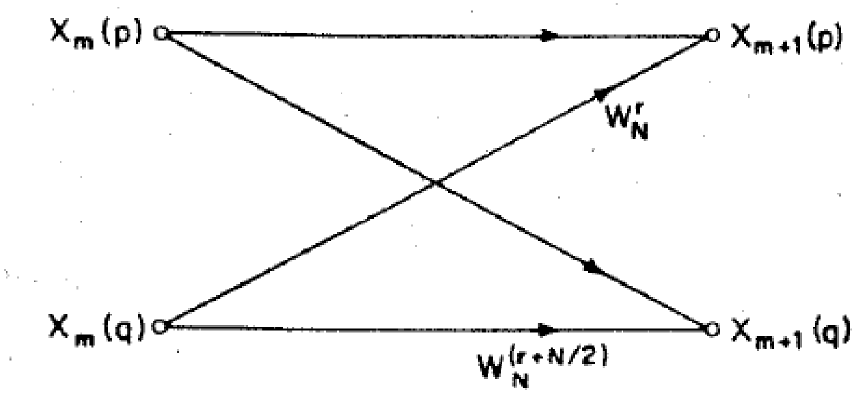
Flow graph of complete DIT decomposition of

8-point DFT calculation.

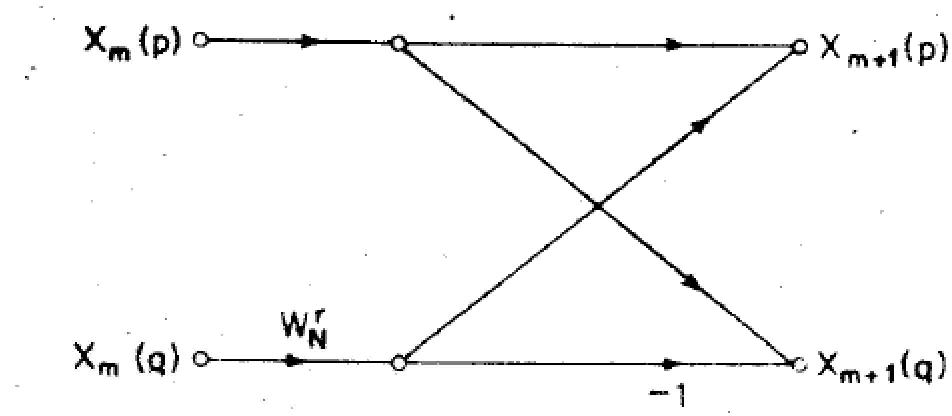


Radix 2 DIT FFT

The Butterfly form



$$X_{m+1}(p) = X_m(p) + W_N^r X_m(q)$$
$$X_{m+1}(q) = X_m(p) + W_N^{r+N/2} X_m(q)$$

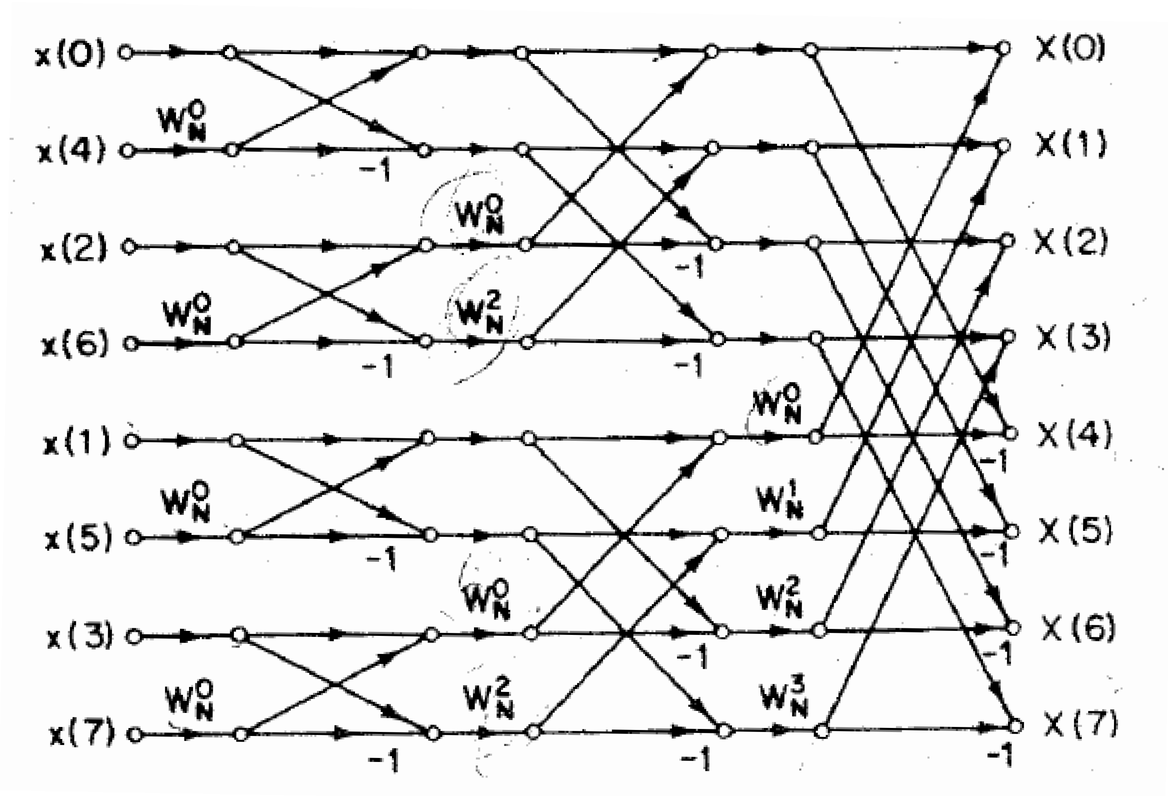


$$X_{m+1}(p) = X_m(p) + W_N^r X_m(q)$$
$$X_{m+1}(q) = X_m(p) - W_N^r X_m(q)$$

Since $W_N^{N/2} = \exp(-j2\pi N/2/N) = \exp(-j\pi) = -1$

Radix 2 DIT FFT

Final result: N = 8 case



$log_2 N$ stages

N/2 butterflies

multiplications: N/2 $\log_2$N

additions: N $\log_2$N

Radix 2 DIT FFT

In-place computation and bit reversal

Bit reversal in FFT

Index in Decimal number	Binary Representation	Bit reversed Binary	Bit reverse index in decimal
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Radix 2 DIF FFT

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

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

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

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

because $W_N^{(N/2)k} = \exp(-j\pi k) = (-1)^k$

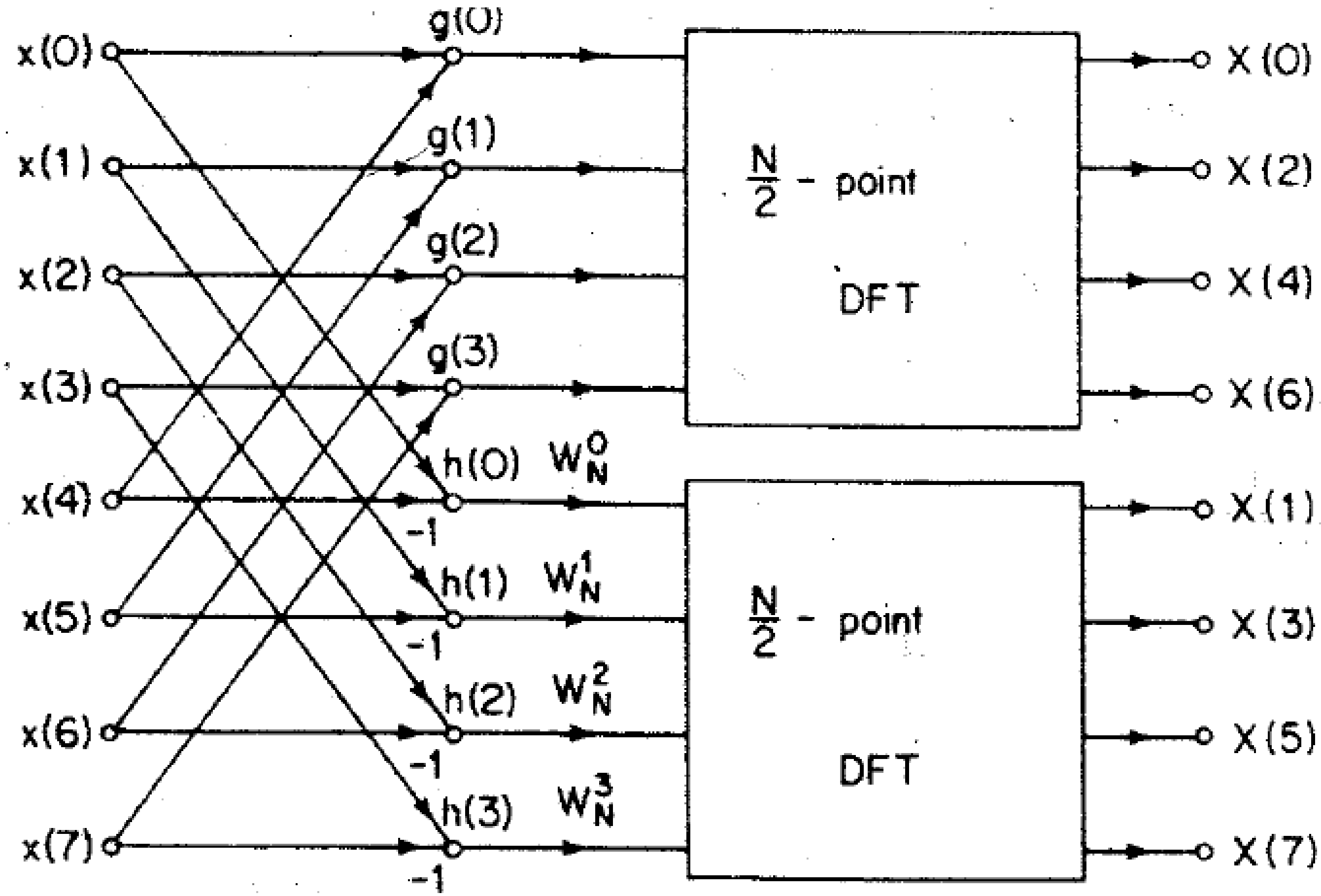
$$X(2k) = \sum_{n=0}^{N/2-1} \left[x(n) + x(n+N/2) \right] \cdot W_N^{2nk}$$

$$X(2k+1) =$$

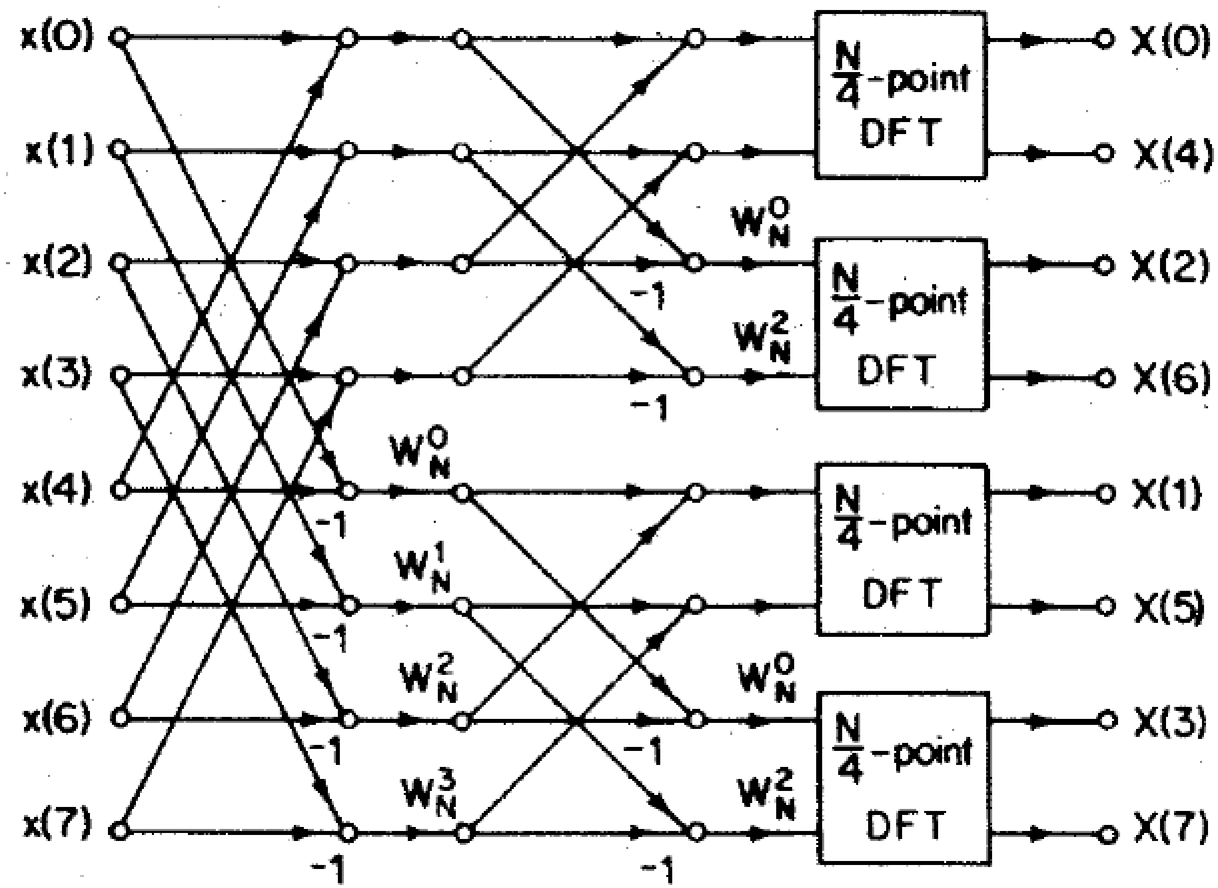
$$\sum_{n=0}^{N/2-1} \left[x(n) - x(n+N/2) \right] \cdot W_N^{n} \cdot W_N^{2nk}$$

(k = 0, 1, …, N/2-1)

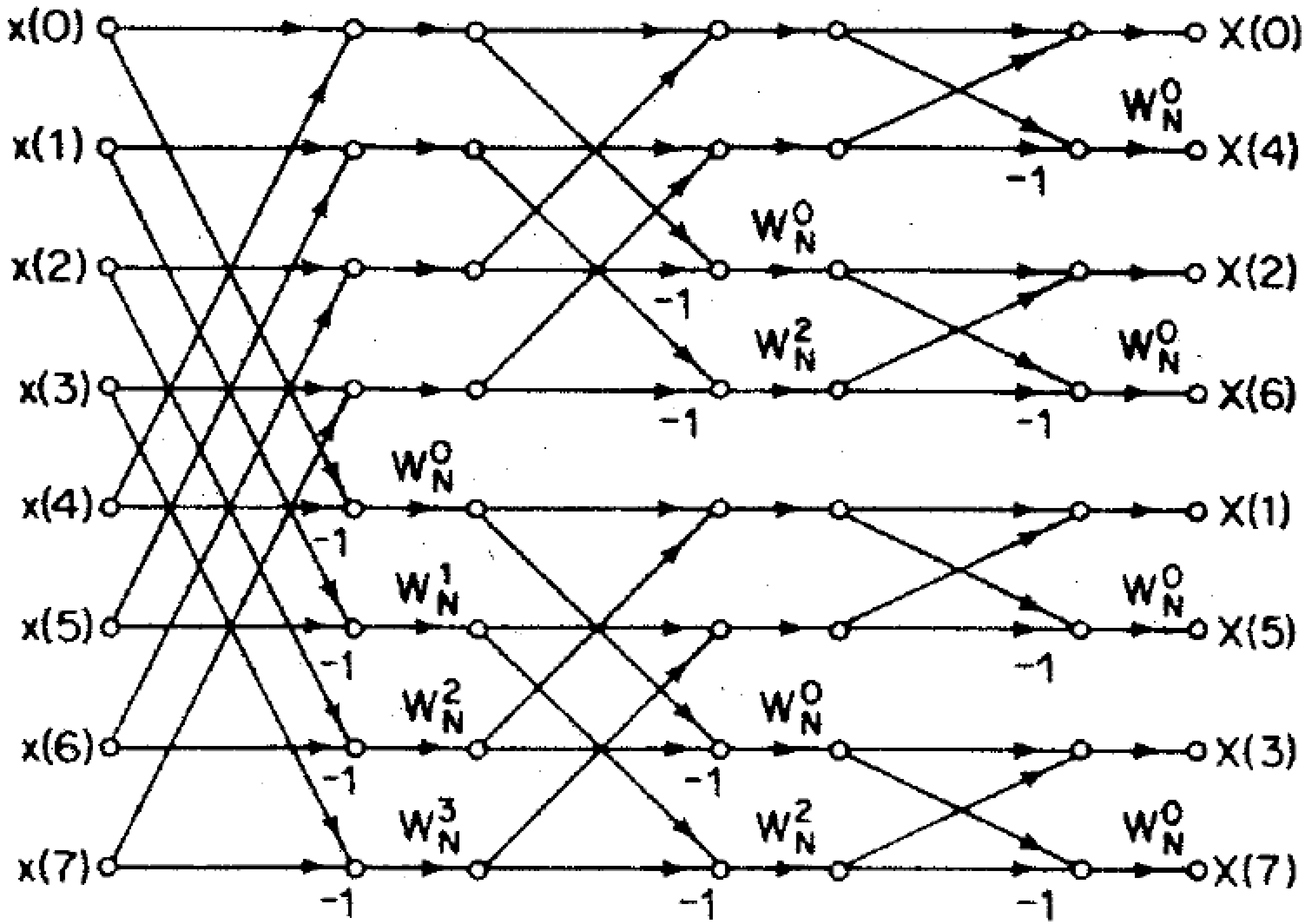
Radix 2 DIF FFT



Radix 2 DIF FFT



Radix 2 DIF FFT



Radix 2 Inverse FFT

Computation of Inverse DFT (IDFT)

To compute IDFT, FFT algorithm can be used without any change in the algorithm.

IDFT of N-point sequence {X(k)}

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

$$\rightarrow \quad N x^*(n) = \sum_{k=0}^{N-1} X^*(k) W_N^{kn} \qquad \text{(K)}$$

$$\rightarrow \quad x(n) = \frac{1}{N}\left\{\sum_{k=0}^{N-1} X^*(k) W_N^{kn}\right\}^* \qquad \text{(L)}$$

Radix 2 FFT

3. Pruned FFT

When the input data sequence has a considerably smaller number of nonzero samples compared with that of zero samples, significant amount of computations can be saved by so called “Pruned FFT”.

Applications of FFT algorithms

Efficient computation of the DFT of two real sequences

$$x(n) = x_1(n) + jx_2(n) \; : \; x_1(n), x_2(n)$$

: two real sequences

$$X(k) = X_1(k) + jX_2(k)$$

$$x_1(n) = \frac{x(n) + x^*(n)}{2}, \quad x_2(n) = \frac{x(n) - x^*(n)}{2j}$$

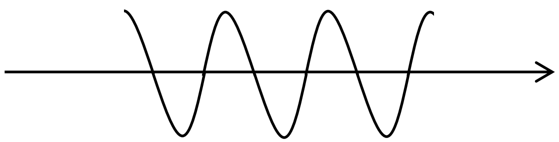
Matlab and FFT

FFT(x, N): N point FFT of $x[n]$

$n = [0:29]$

$x = \cos(2\pi n/10)$

$\theta = 2\pi/10 \quad f = 1/10$

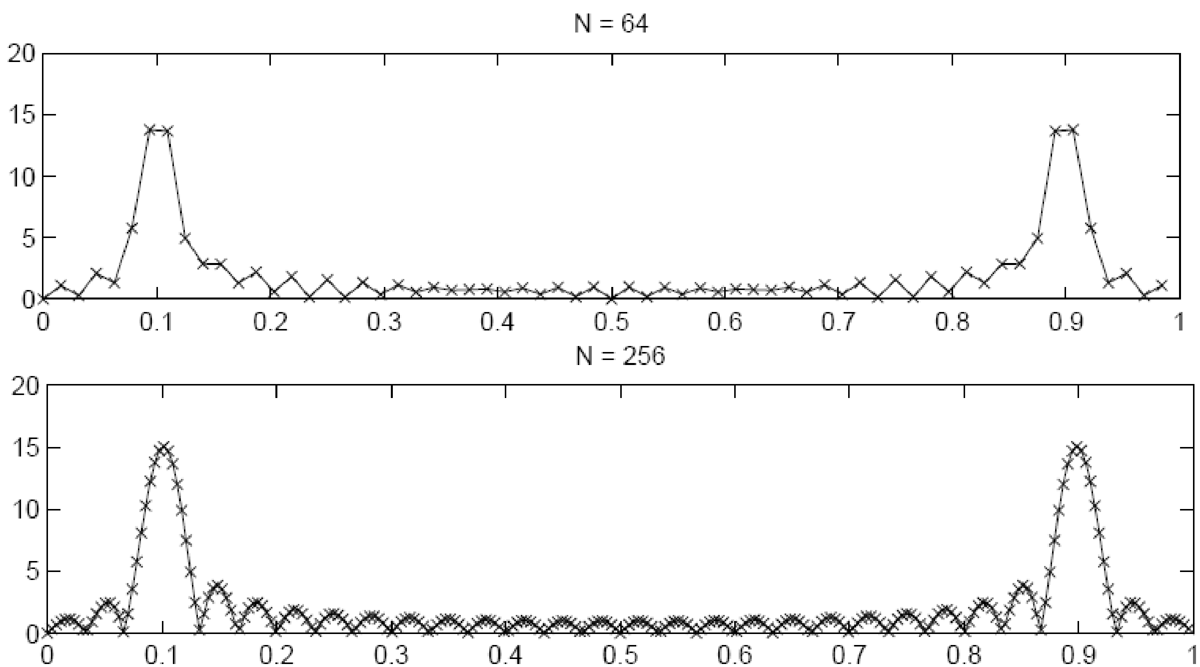


$N = 64$

$XN = abs\big(fft(x, N)\big)$

$F = [0 : N - 1]/N$

$plot(F, XN, '-x'), title('N = 64'), axis([0\ 1\ 0\ 20])$



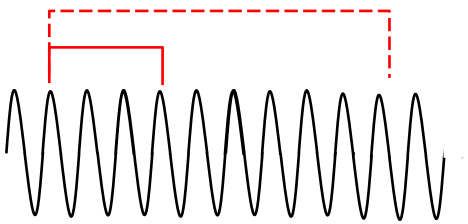
$N = 256$

Matlab and FFT

$n = [0{:}29]$

$x1 = \cos(2\pi n/10)$

$x3 = [x1\ x1\ x1]$



$N = 2048$

$X1 = abs\big(fft(x1, N)\big)$ $X3 = abs\big(fft(x3, N)\big)$

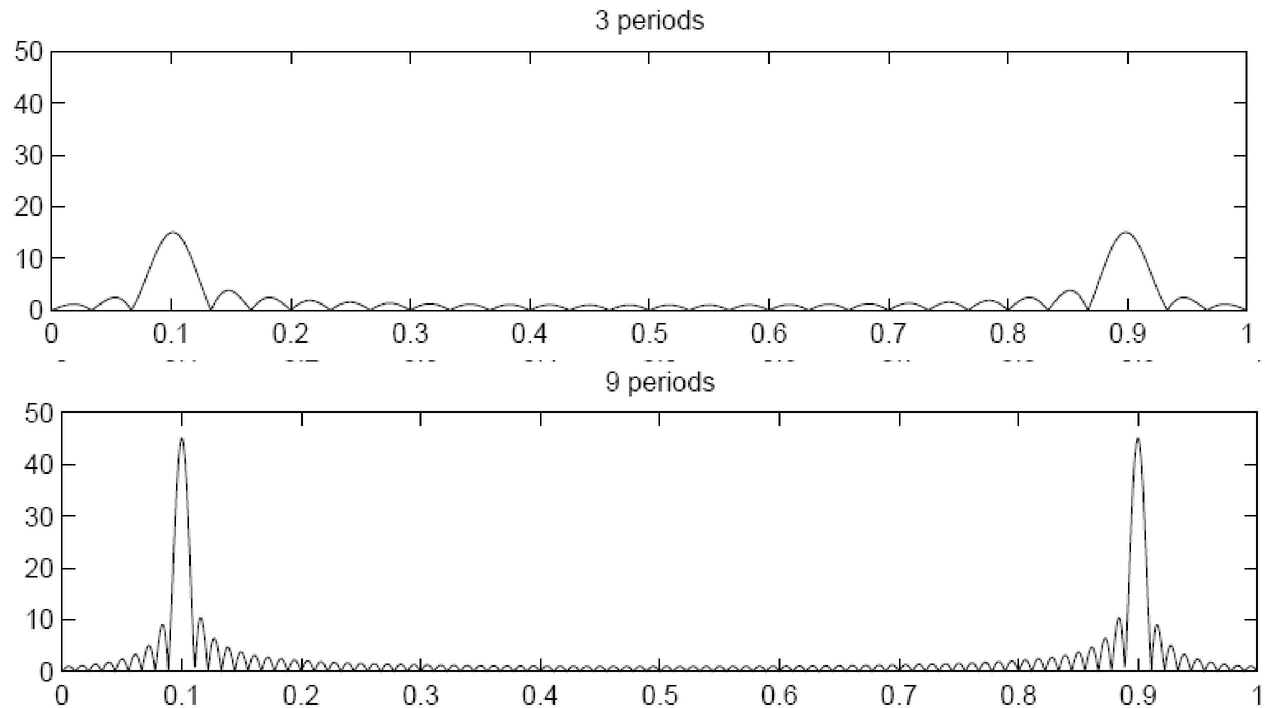
$F = [0 : N\ - 1]/N$

$subplot(2{,}1{,}1)$

$plot(F, X1), title('3\ periods'), axis([0\ 1\ 0\ 50])$

$subplot(2{,}1{,}2)$

$plot(F, X3), title('9\ periods'), axis([0\ 1\ 0\ 50])$



Matlab and FFT

$n = [0{:}149]$

$x1 = \cos(2\pi n/10)$

$N = 2048$

$X1 = fft(x1, N)$

$X1 = fftshift(X1)$

$F = [-N/2 : N/2 - 1]/N$

$plot(F, X1)$

$xlabel('frequency\ /\ f\_s')$