Digital Signal Analysis and Processing

Madhav P Pandey*
DoEEE, KU
* Compiled by

Discrete Fourier Transform (DFT)

What and Why DFT?

Frequency Sampling and DFT Equations

Properties of DFT and Circular Convolution

What and Why FFT?

Divide and Conquer Principle

DIT and DIF Radix-2 algorithms

What is DFT?

- Discrete Fourier Transform (DFT) provides a frequency domain representation for a discrete time signal $x[n]$.

- DFT converts a sequence of numbers in time domain to another sequence of numbers in frequency domain.

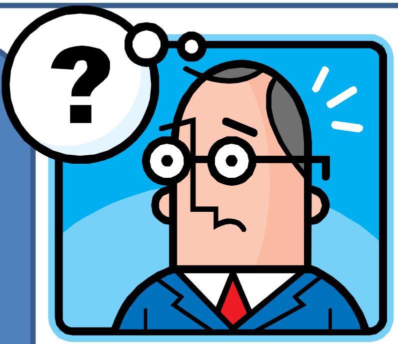
- It is a powerful computational tool for frequency analysis of DT signals.

- Mathematically, N point DFT of a sequence $x[n]$ is denoted and defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n}$$

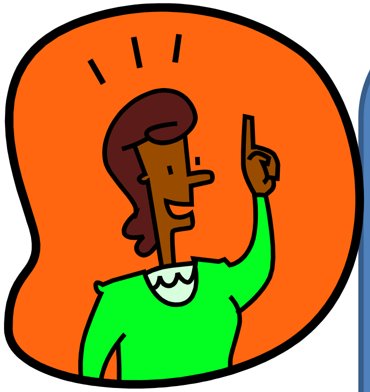
- Whereas the inverse transform is: $$x[n] = \sum_{k=0}^{N-1} X[k] e^{jk\frac{2\pi}{N}n}$$

Why DFT?



- As studied earlier, frequency domain representation of a DT sequence x[n] is given by its Discrete Time Fourier Transform (DTFT) $X(e^{j\Omega})$. Then why do we need DFT?

- Is DFT related to DTFT?



- Yes, DTFT provides the frequency domain representation, but it is continuous function of frequency Ω. Hence is not a convenient representation for DT systems.

- DFT is in fact related to the samples of DTFT and is obtained by frequency domain sampling.

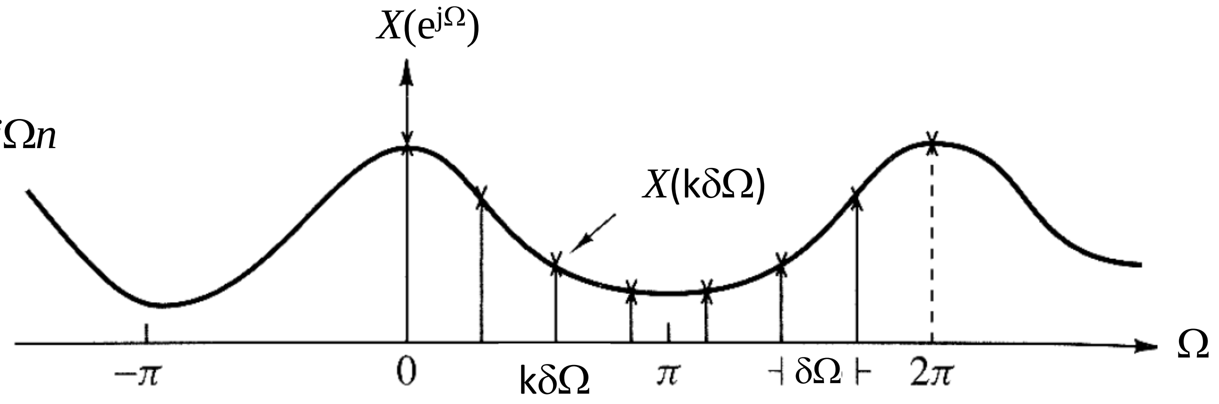
Why and What DFT?

- When digital processors are used for signal processing, they require all information to be stored as sequence of numbers.
- Hence frequency analysis using DSPs require frequency domain representation of signals also as a sequence of numbers.
- DTFT however provides such representation as continuous function of frequency. Hence is not suitable representation.
- Hence, the samples of DTFT are used as frequency domain representation of the signals.
- Such sampling of DTFT results an alternative transform known as DFT.

Frequency Domain Sampling

Let us consider a finite duration aperiodic DT signal x[n] whose DTFT is $X(e^{j\Omega})$ as shown. As known, $X(e^{j\Omega})$ is periodic with period 2π.

$$X\left(e^{j\Omega}\right) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n}$$



If $X(e^{j\Omega})$ is sampled such that N samples are obtained in one period ($0 \leq \Omega < 2\pi$) , the samples will be spaced in frequency $\delta\Omega$ apart where, $\delta\Omega = \dfrac{2\pi}{N}$

The sample values will be given by $X(k\delta\Omega)$.

Frequency Domain Sampling

The samples of the DTFT are:

$$X\left(\frac{2\pi}{N}k\right)=\sum_{n=-\infty}^{\infty}x[n]e^{-j\frac{2\pi}{N}kn} \qquad k=0,1,2,........N-1$$

$$=.......+\sum_{n=-N}^{-1}x[n]e^{-j\frac{2\pi}{N}kn}+\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn}+\sum_{n=N}^{2N-1}x[n]e^{-j\frac{2\pi}{N}kn}+.....$$

$$=\sum_{l=-\infty}^{\infty}\sum_{n=lN}^{lN+N-1}x[n]e^{-j\frac{2\pi}{N}kn}$$

Changing the inner summation index as $p=n-lN$

$$=\sum_{l=-\infty}^{\infty}\left[\sum_{p=0}^{N-1}x[p+lN]\right]e^{-j\frac{2\pi}{N}k(p+lN)}=\sum_{l=-\infty}^{\infty}\left[\sum_{p=0}^{N-1}x[p+lN]\right]e^{-j\frac{2\pi}{N}kp}$$

Equivalently,

$$=\sum_{l=-\infty}^{\infty}\left[\sum_{n=0}^{N-1}x[n+lN]\right]e^{-j\frac{2\pi}{N}kn}$$

Frequency Domain Sampling

The signal inside the bracket is obtained by repeating $x[n]$ every N samples and hence is periodic with period N. Its one period is same to $x[n]$.

If we define $x_p[n] = \sum_{l=\infty}^{-\infty} x[n-lN]$,

Then,
$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1} x_p[n] e^{-j\frac{2\pi}{N}kn}$$

Since $x_p[n]$ is periodic, it can be represented as Fourier Series with coefficients:

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p[n] e^{-j\frac{2\pi}{N}kn} \qquad k = 0,1,2,\ldots\ldots N-1$$

$$= \frac{1}{N} X\left(\frac{2\pi}{N}k\right) \qquad k = 0,1,2,\ldots\ldots N-1$$

Frequency Domain Sampling

The Fourier series representation thus is:

$$x_p[n] = \sum_{k=0}^{N-1} \frac{1}{N} X\left(\frac{2\pi}{N}k\right) e^{jk\frac{2\pi}{N}n} \qquad n = 0,1,2,\ldots\ldots N-1$$

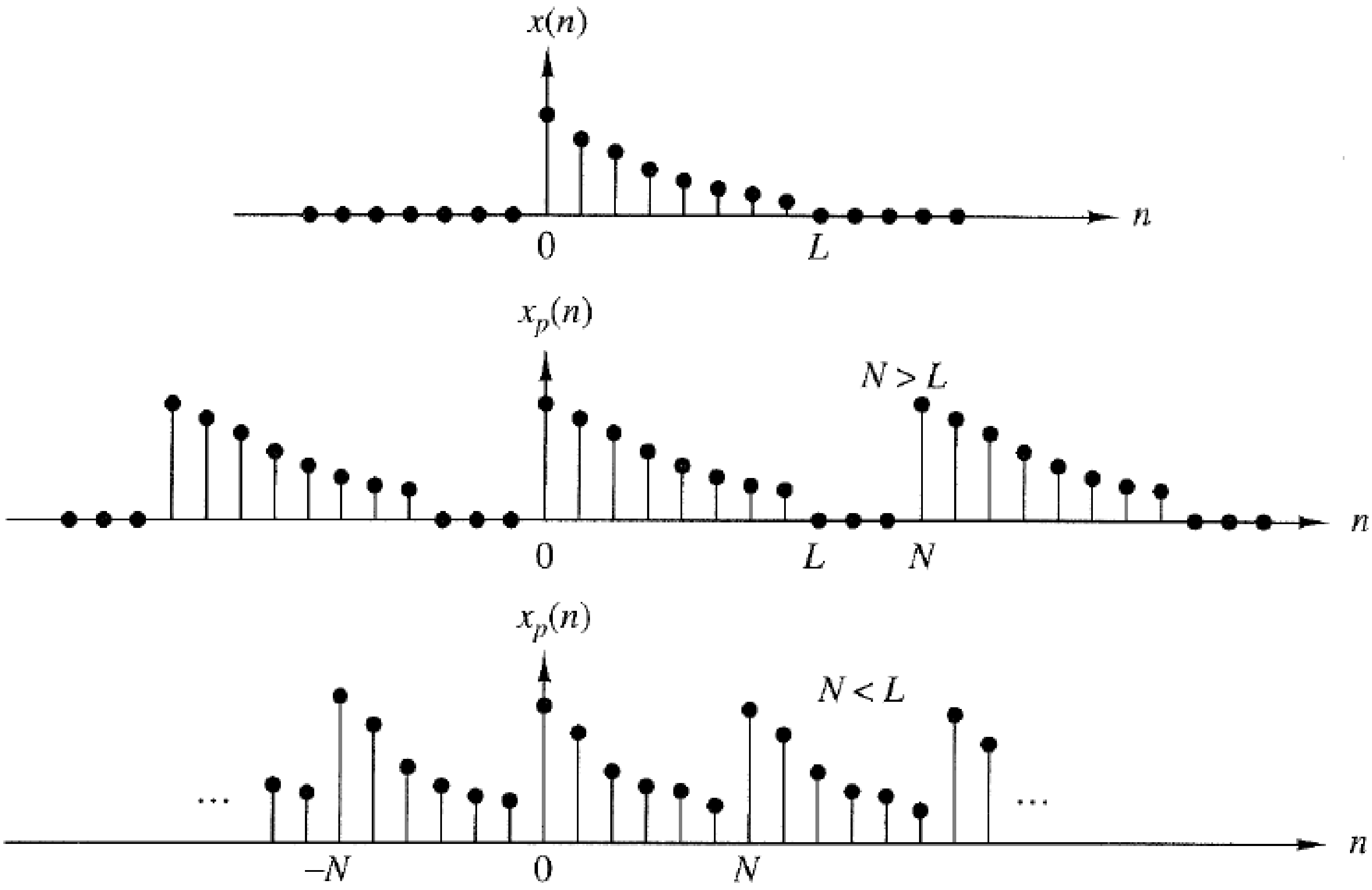
The above expression provides a method to recover $x_p[n]$ from the samples of $X(e^{j\Omega})$. Since one period of $x_p[n]$ is same to $x[n]$, it also provides a method to recover $x[n]$ from the samples of its DTFT.

However, it requires that there is no aliasing (overlapping) while repeating $x[n]$ to generate $x_p[n]$.

For this, if L is the length of $x[n]$, then no of samples of DTFT taken, N must be greater than or equal to L.

Considering $N>L$ does not provide additional information.

Time Domain Aliasing



Frequency Samples and Signals

Thus, N samples of DTFT in fact does not represent spectrum of finite duration signal $x[n]$ but its periodic extension $x_p[n]$. However, when there is no time domain aliasing ($N \geq L$), $x[n]$ can be determined from $x_p[n]$.

Thus a finite duration sequence x[n] with length L has DTFT

$$X\left(e^{j\Omega}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} = \sum_{n=0}^{L-1} x[n]e^{-j\Omega n}$$

and its samples

$$X[k] = X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{L-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

DFT Equations

For convenience,

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \qquad k = 0,1,2,......N-1}$$

and

$$\boxed{x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]]e^{j\frac{2\pi}{N}kn} \qquad n = 0,1,2,......N-1}$$

are known as N point Discrete Fourier Transform and Inverse Discrete Fourier Transform equations.

Also,

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad k = 0,1,2,......N-1}$$

$$\boxed{x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]]W_N^{-kn} \qquad n = 0,1,2,......N-1}$$

Where,

$$W_N = e^{-j\frac{2\pi}{N}}$$

DFT as Linear Transformation

The set of equations to compute N point DFT can be written as

$$\mathbf{X_N} = \mathbf{W_N}\ \mathbf{x_N}$$

Where,

$$\mathbf{x}_N = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \qquad \mathbf{X}_N = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

$$\mathbf{W}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

And,

$$\mathbf{x_N} = \mathbf{W_N}^{-1}\ \mathbf{X_N} = \frac{1}{\mathbf{N}}\mathbf{W_N}^{*}\ \mathbf{X_N}$$

Matrix of twiddle factors

The $N \times N$ matrix for $N=4$ is

$$\mathbf{W}_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Properties of DFT

Periodicity:

If, $x[n] \xleftrightarrow{N\ DFT} X[k]$

then, $x[n+N] = x[n] \qquad for \quad all \quad n$

and, $X[k+N] = X[k] \qquad for \quad all \quad k$

Proof:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}kn} \qquad n = 0,1,2,\ldots\ldots N-1$$

So,

$$x[n+N] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}k(n+N)}$$

$$x[n+N] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}kn} \qquad \because \quad e^{j\frac{2\pi}{N}kN} = 1$$

$$= x[n]$$

Properties of DFT

Symmetry Properties:

If,

$$x[n] \xleftrightarrow{N\ DFT} X[k] = X_R[k] + jX_I[k]$$

then,

for real $x[n]$,

$$X[N-k] = X^*[k] = X[-k]$$

For real and even $x[n]$,

$$X_I[k] = 0 \quad and \qquad X[k] = \sum_{n=0}^{N-1} x[n]\cos\left(\frac{2\pi}{N}kn\right)$$

For real and odd $x[n]$,

$$X_R[k] = 0 \quad and \qquad X[k] = -j\sum_{n=0}^{N-1} x[n]\sin\left(\frac{2\pi}{N}kn\right)$$

Properties of DFT

Linearity:

If, $x_1[n] \xleftrightarrow{N\ DFT} X_1[k]$ and $x_2[n] \xleftrightarrow{N\ DFT} X_2[k]$

then, for constants A and B, real $x[n]$,

$$x_3[n] = A\,x_1[n] + B\,x_2[n] \xleftrightarrow{N\ DFT} X_3[k] = A\,X_1[k] + B\,X_2[k]$$

Proof:

$$X_3[k] = \sum_{n=0}^{N-1} x_3[n] W_N^{kn} = \sum_{n=0}^{N-1} \left[A\,x_1[n] + B\,x_2[n]\right] W_N^{kn}$$

$$= \sum_{n=0}^{N-1} A\,x_1[n] W_N^{kn} + \sum_{n=0}^{N-1} B\,x_2[n] W_N^{kn}$$

$$= A\,X_1[k] + B\,X_2[k]$$

Properties of DFT

Circular shift and related property:

Consider the periodic extension of x[n]

$$x_p[n] = \sum_{l=\infty}^{-\infty} x[n - lN]$$

and its shifted version,

$$x'_p[n] = x_p[n-k] = \sum_{l=-\infty}^{\infty} x[n-k-lN]$$

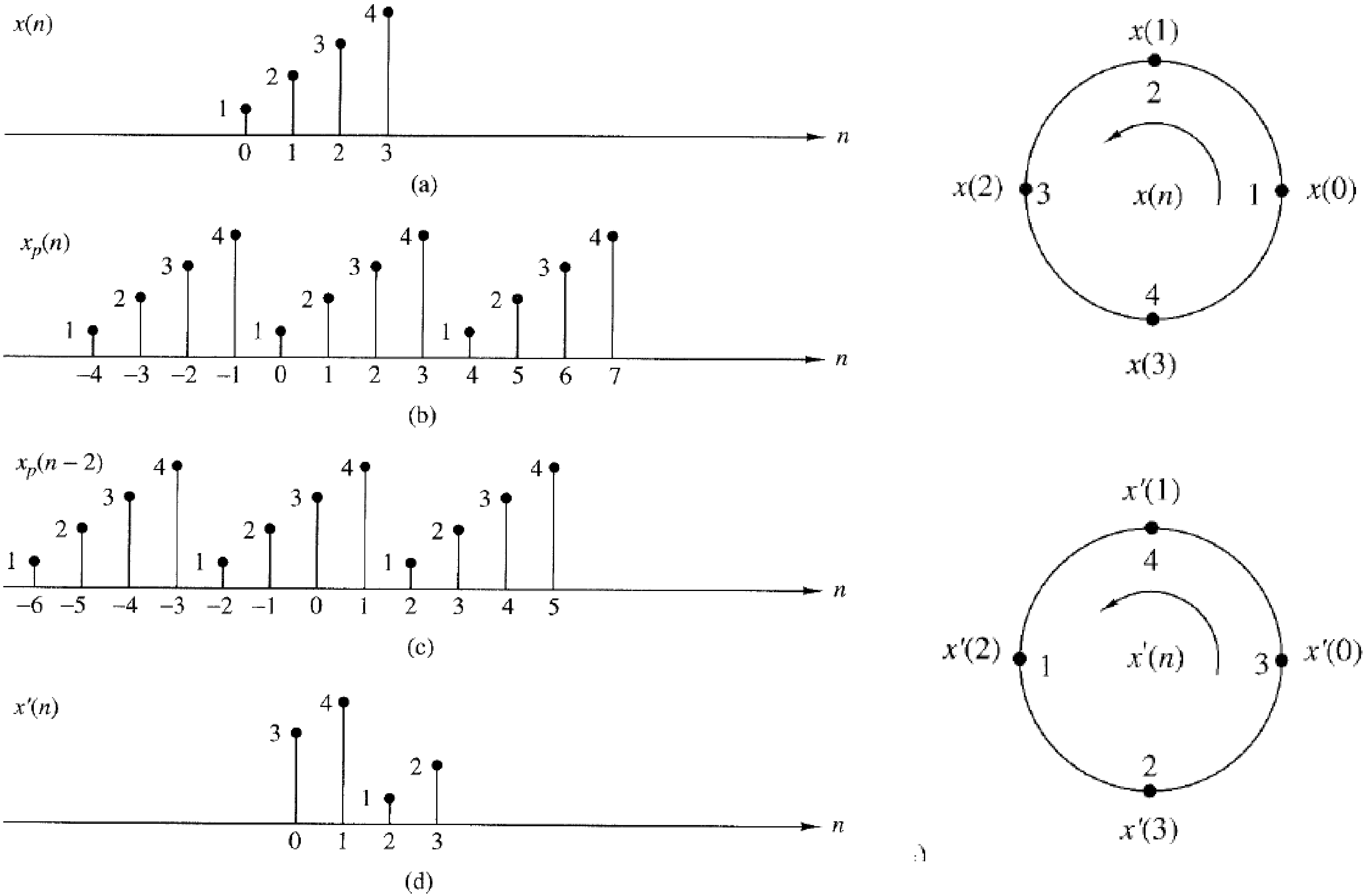
Also a finite duration sequence

$$x'(n) = \begin{cases} x'_p(n), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases}$$

If we notice the relation between $x[n]$ and $x'[n]$, they are related by circular shifting. Circular shift of a finite sequence is equal to the linear shift of its periodic extension.

Properties of DFT

The operations are illustrated in an example with N=4



Properties of DFT

Mathematically, the circular shifting operation is represented as:

$$x'[n] = x(n\text{-}k, \text{ modulo } N) = x((n\text{-}k))_N$$

For the example shown before,

$$x'[n] = x(n\text{-}2, \text{ modulo } 4) = x((n\text{-}2))_4$$

$$x((n\text{-}k))_N = x[n\text{-}k] \quad \text{if } \; n\text{-}k \geq 0$$

$$x((n\text{-}k))_N = x[n\text{-}k+N] \quad \text{if } \; n\text{-}k < 0$$

$$x'(0) = x((-2))_4 = x(2)$$

$$x'(1) = x((-1))_4 = x(3)$$

$$x'(2) = x((0))_4 = x(0)$$

$$x'(3) = x((1))_4 = x(1)$$

Properties of DFT

Time reversal

If, $x[n] \xleftrightarrow{N\ DFT} X[k]$

then,

$$x((-n))_N = x[N-n] \xleftrightarrow{N\ DFT} X((-k))_N = X[N-k]$$

Proof:

$$\mathrm{DFT}\{x(N-n)\} = \sum_{n=0}^{N-1} x(N-n)e^{-j2\pi kn/N}$$

change the index from n to $m = N - n$, then

$$= \sum_{m=0}^{N-1} x(m)e^{-j2\pi k(N-m)/N}$$

$$= \sum_{m=0}^{N-1} x(m)e^{j2\pi km/N}$$

$$= \sum_{m=0}^{N-1} x(m)e^{-j2\pi m(N-k)/N} = X(N-k)$$

Properties of DFT

Circular Time Shifting Property

If, $x[n] \xleftrightarrow{N\ DFT} X[k]$

then, $x((n-l))_N \xleftrightarrow{N\ DFT} X[k]e^{-j\frac{2\pi}{N}kl}$

Proof: $$\text{DFT}\{x((n-l))_N\} = \sum_{n=0}^{N-1} x((n-l))_N e^{-j2\pi kn/N}$$

$$= \sum_{n=0}^{l-1} x((n-l))_N e^{-j2\pi kn/N} + \sum_{n=l}^{N-1} x(n-l)e^{-j\pi kn/N}$$

But $x((n-l))_N = x(N-l+n)$. Consequently,

$$\sum_{n=0}^{l-1} x((n-l))_N e^{-j2\pi kn/N} = \sum_{n=0}^{l-1} x(N-l+n)e^{-j2\pi kn/N}$$

$$= \sum_{m=N-l}^{N-1} x(m)e^{-j2\pi k(m+l)/N}$$

Properties of DFT

Furthermore,

$$\sum_{n=l}^{N-1} x(n-l)e^{-j2\pi kn/N} = \sum_{m=0}^{N-1-l} x(m)e^{-j2\pi k(m+l)/N}$$

$$x(n)e^{j2\pi ln/N} \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X((k-l))_N$$

Therefore,

$$\text{DFT}\{x((n-l))\} = \sum_{m=0}^{N-1} x(m)e^{-j2\pi k(m+l)/N}$$

$$= X(k)e^{-j2\pi kl/N}$$

Circular Frequency Shifting Property:

If,

$$x[n] \xleftrightarrow{N\ DFT} X[k]$$

Then,

$$x[n]e^{j\frac{2\pi}{N}nl} \xleftrightarrow{N\ DFT} X((k-l))_N$$

Properties of DFT

Complex Conjugate Property

If,

$$x[n] \xleftrightarrow{N\ DFT} X[k]$$

then,

$$x^*[n] \xleftrightarrow{N\ DFT} X^*((-k))_N = X^*(N-k)$$

and

$$x^*((-n))_N = x^*(N-n) \xleftrightarrow{N\ DFT} X^*[k]$$

Properties of DFT

Circular Convolution Property

If,

$$x_1[n] \xleftrightarrow{N\ DFT} X_1[k]$$

and,

$$x_2[n] \xleftrightarrow{N\ DFT} X_2[k]$$

then,

$$x_1[n] \circledN x_2[n] \xleftrightarrow{N\ DFT} X_1[k]X_2[k]$$

where, $x_1[n] \circledN x_2[n]$ is denotes the circular convolution of the sequence $x_1[n]$ and $x_2[n]$.

$$x_1[n] \circledN x_2[n] = \sum_{m=0}^{N-1} x_1[m] x_2((n-m))_N$$

Circular convolution in time domain is equivalent to multiplication of two DFTs.

Properties of DFT

Proof:

Let $X_3[k] = X_1[k]X_2[k]$

By definition,

$$x_3[m] = \frac{1}{N}\sum_{k=0}^{N-1} X_3[k]e^{j\frac{2\pi}{N}km}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left[X_1[k]\,X_2[k]\right]e^{j\frac{2\pi}{N}km}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{n=0}^{N-1} x_1[n]e^{-j\frac{2\pi}{N}kn}\sum_{l=0}^{N-1} x_2[l]e^{-j\frac{2\pi}{N}kl}\right]e^{j\frac{2\pi}{N}km}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x_1[n]\sum_{l=0}^{N-1} x_2[l]\left[\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}k(m-n-l)}\right]$$

Properties of DFT

The sum $\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}k(m-n-l)}$ has the form of $\sum_{k=0}^{N-1} a^k$

Such that $a = e^{j\frac{2\pi}{N}(m-n-l)}$

As we know, $\sum_{k=0}^{N-1} a^k = \begin{cases} N & for\ a = 1 \\ \dfrac{1-a^N}{1-a} & for\ \ a \neq 1 \end{cases}$ But, $a^N = e^{j2\pi(m-n-l)} = 1$

Thus,

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & for\ a = 1 \\ 0 & for\ \ a \neq 1 \end{cases}$$

and for a=1, (m-n-l) must be an integer multiple of N

$$m - n - l = pN \Rightarrow l = m - n + pN$$

$$l = ((m-n))_N$$

Properties of DFT

$$\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}k(m-n-l)} = \begin{cases} N & for \quad l = ((m-n))_N \\ 0 & otherwise \end{cases}$$

Using this,

$$x_3[m] = \frac{1}{N}\sum_{n=0}^{N-1} N\, x_1[n]\, x_2((m-n))_N$$

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n]\, x_2((m-n))_N$$

$$= x_1[n] \; Ⓝ \; x_2[n]$$

Since it is convolution using circular shifting, it is called circular convolution.

Properties of DFT

Multiplication Property

If $$x_1[n] \xleftrightarrow{N\ DFT} X_1[k]$$

And
$$x_2[n] \xleftrightarrow{N\ DFT} X_2[k]$$

Then,

$$x_1[n]x_2[n] \xleftrightarrow{N\ DFT} \frac{1}{N} X_1[k] \; \text{Ⓝ} \; X_2[k]$$

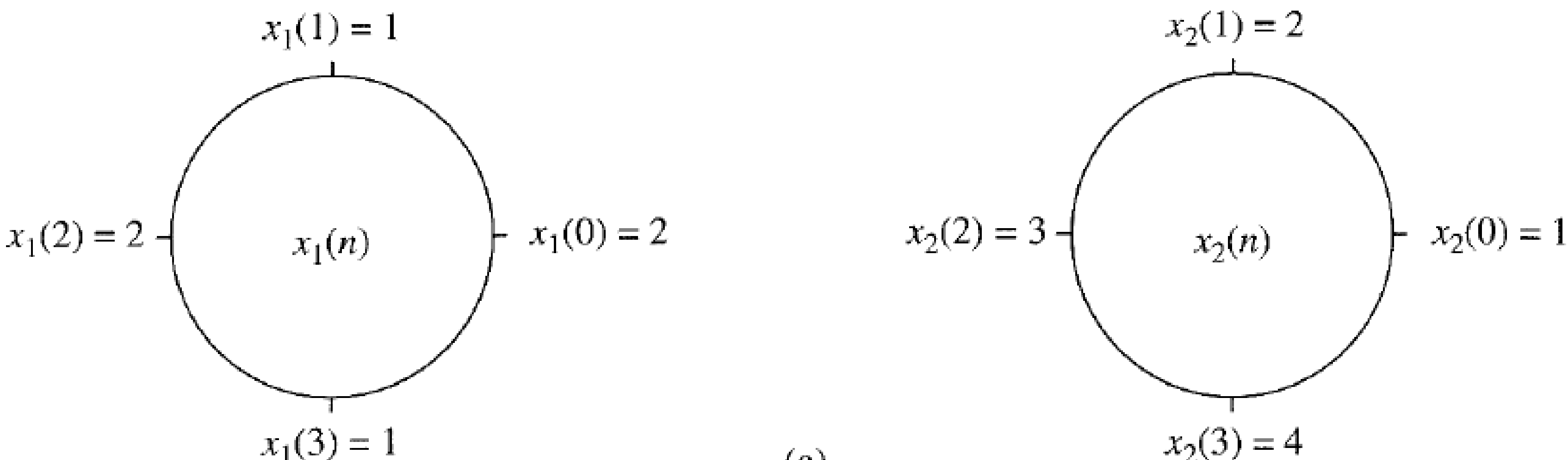
This is an inverse property to convolution

Circular Graph and Circular Convolution

- Circular convolution can be evaluated graphically using circular graphs.
- This is illustrated using an example.

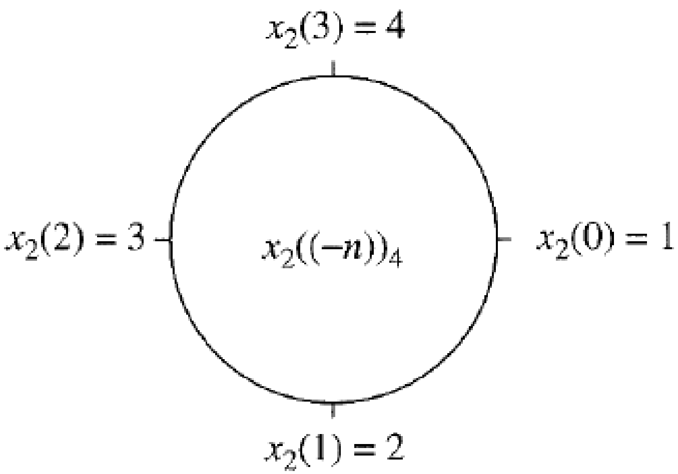
Perform the circular convolution of the following two sequences:

$$x_1(n) = \{\underset{\uparrow}{2}, 1, 2, 1\} \qquad x_2(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$$

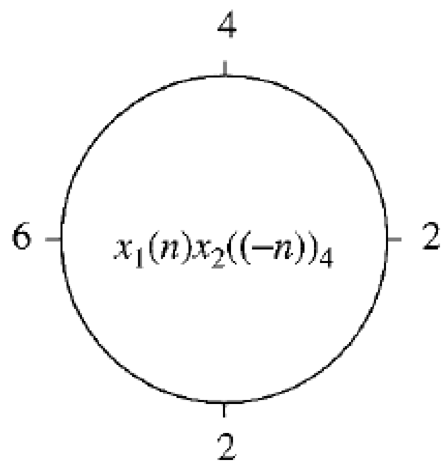


(a)

Circular Graph and Circular Convolution



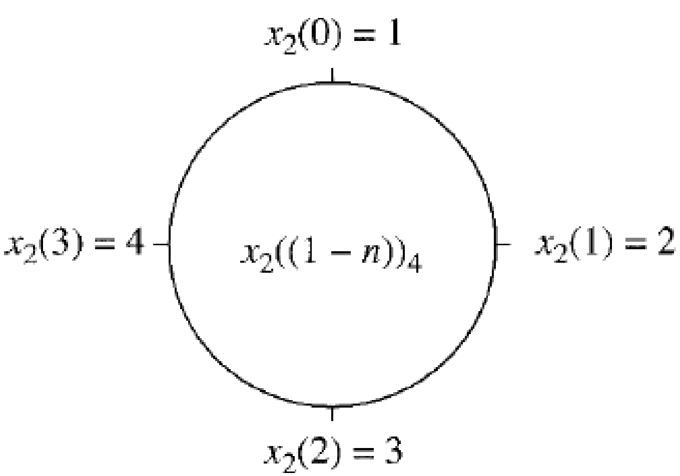
Folded sequence



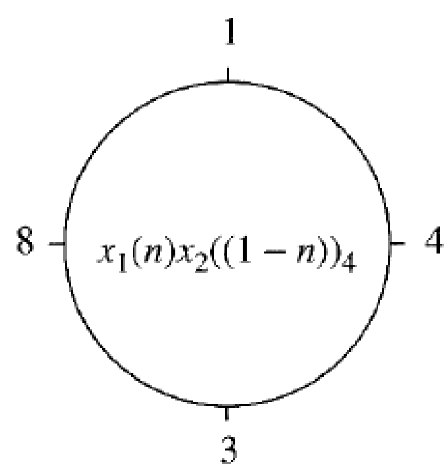
Product sequence

$$x_3(0) = \sum_{n=0}^{3} x_1(n)x_2((-n))N$$

$$x_3(0) = 14$$



Folded sequence rotated by one unit in time

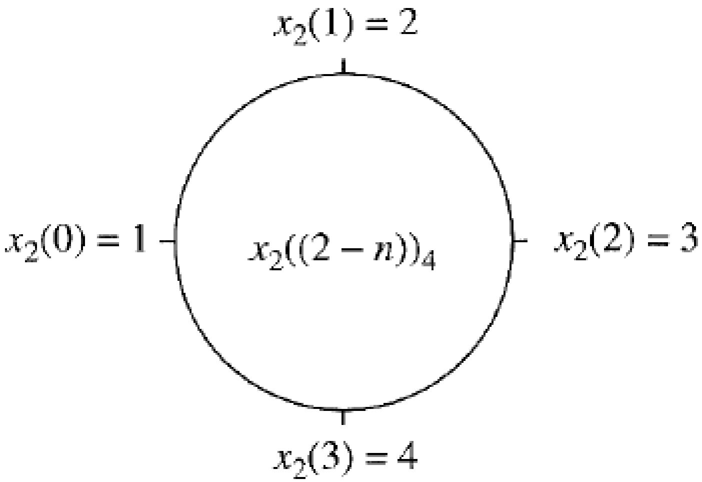


Product sequence

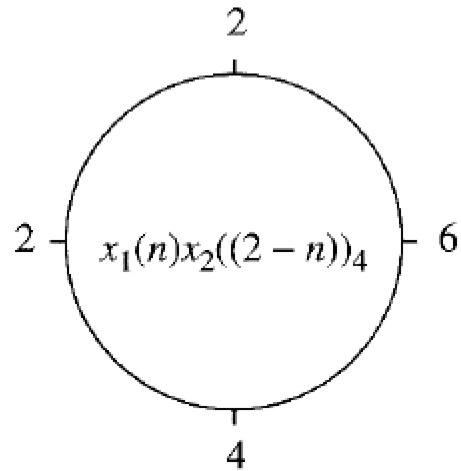
$$x_3(1) = \sum_{n=0}^{3} x_1(n)x_2((1-n))_4$$

$$x_3(1) = 16$$

Circular Graph and Circular Convolution



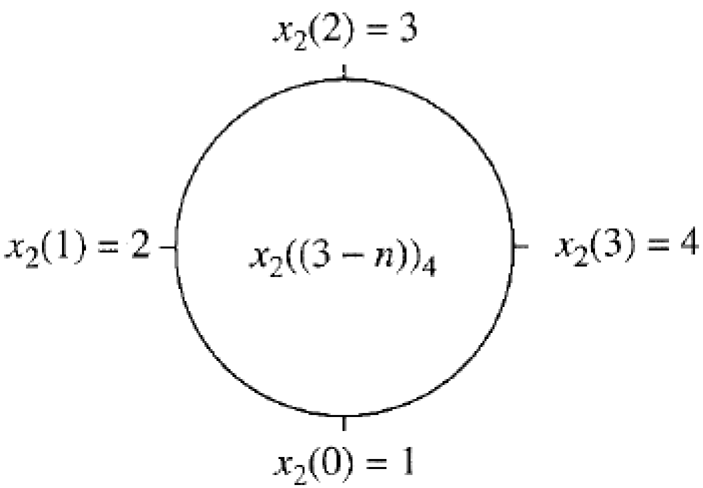
Folded sequence rotated by two units in time



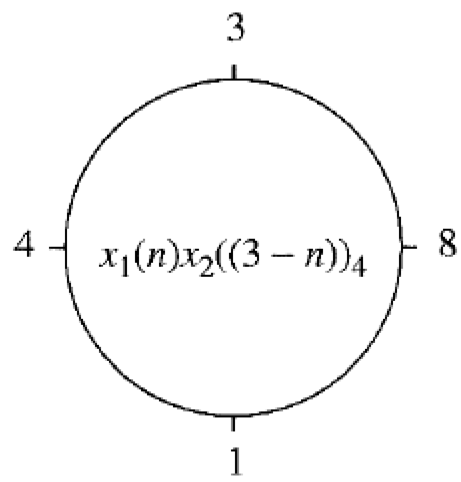
Product sequence

$$x_3(2) = \sum_{n=0}^{3} x_1(n)x_2((2-n))_4$$

$$x_3(2) = 14$$



Folded sequence rotated by three units in time



Product sequence

$$x_3(3) = \sum_{n=0}^{3} x_1(n)x_2((3-n))_4$$

$$x_3(3) = 16$$

Circular Convolution and DFT-IDFT

- Convolution property enables the computation of circular convolution using DFT and IDFT.
- Steps:
  - Compute N point DFTs of sequences to be convolved.
  - Compute the product of the two DFTs.
  - Compute N point inverse DFT

Perform the circular convolution of the following two sequences:

$$x_1(n) = \{\underset{\uparrow}{2}, 1, 2, 1\} \qquad x_2(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$$

$$X_1(k) = \sum_{n=0}^{3} x_1(n) e^{-j2\pi nk/4}, \qquad k = 0, 1, 2, 3$$

$$= 2 + e^{-j\pi k/2} + 2e^{-j\pi k} + e^{-j3\pi k/2}$$

$$X_1(0) = 6, \qquad X_1(1) = 0, \qquad X_1(2) = 2, \qquad X_1(3) = 0$$

Circular Convolution and DFT-IDFT

- The DFT of $x_2(n)$ is

$$X_2(k) = \sum_{n=0}^{3} x_2(n) e^{-j2\pi nk/4}, \qquad k = 0, 1, 2, 3$$

$$= 1 + 2e^{-j\pi k/2} + 3e^{-j\pi k} + 4e^{-j3\pi k/2}$$

$$X_2(0) = 10, \qquad X_2(1) = -2 + j2, \qquad X_2(2) = -2, \qquad X_2(3) = -2 - j2$$

$$X_3(k) = X_1(k) X_2(k)$$

$$X_3(0) = 60, \qquad X_3(1) = 0, \qquad X_3(2) = -4, \qquad X_3(3) = 0$$

Now, the IDFT of $X_3(k)$ is

$$x_3(n) = \sum_{k=0}^{3} X_3(k) e^{j2\pi nk/4}, \qquad n = 0, 1, 2, 3$$

$$= \frac{1}{4}(60 - 4e^{j\pi n})$$

$$x_3(0) = 14, \qquad x_3(1) = 16, \qquad x_3(2) = 14, \qquad x_3(3) = 16$$

What and Why FFT?

- Fast Fourier Transform (FFT) are family of algorithms that allow faster computation of frequency domain representation (DFT) of DT signals.

- Exploits the symmetry of DFT calculation to make its execution faster.

- Highly efficient for DFTs of higher sizes.

- Symmetry property: $W_N^{k+N/2} = -W_N^k$

  Periodicity property: $W_N^{k+N} = W_N^k$

- Also utilize divide and conquer principle.

Computational Complexity of DFT

- For each value of k, direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications + $2N$ real additions) and N-1 complex additions ($2N$-2 real additions)→In total $4N$ real multiplications and $4N$-2 real additions.

- To compute all values in an N point DFT, thus we require N^2 complex multiplications and N^2-N complex additions ($4N^2$ real multiplications and $4N^2-2N$ real additions).

- Computational complexity in general can be considered to be proportional to square of the DFT point N.

- So, for large point DFTs, number of computations becomes very high (for N=256, we need 65536 multiplications and 65280 additions)

Divide and Conquer Approach

- Decomposition of N point DFT into successively smaller DFTs

- If N is a composite number such that

$$N = L\,M$$

even when N is not composite, it can be made composite by padding zeros.

- Now sequence $x[n]$ with N data samples can be addressed as two dimensional array indexed by l (row index) and m (column index) where,

$$0 \le l \le L-1 \quad and \quad 0 \le m \le M-1$$

Divide and Conquer Approach

$n \longrightarrow$	0	1		...	$N-1$
	$x(0)$	$x(1)$	$x(2)$	...	$x(N-1)$

row index l \ m column index	0	1		$M-1$
0	$x(0, 0)$	$x(0, 1)$	...	
1	$x(1, 0)$	$x(1, 1)$	...	
2	$x(2, 0)$	$x(2, 1)$	...	
⋮	⋮	⋮	...	⋮
$L-1$			...	

- Samples in $x[n]$ and $x[l, m]$ can be related in different ways.

Divide and Conquer Approach

- Row wise mapping: $n = M\,l + m$
- Column wise mapping: $n = m\,L + l$
- Similar arrangements can be made for DFT sequence $X[k]$ using indices p and q.

$n = Ml + m$

$l \backslash m$	0	1	2		$M-1$
0	$x(0)$	$x(1)$	$x(2)$	$\cdots$	$x(M-1)$
1	$x(M)$	$x(M+1)$	$x(M+2)$	$\cdots$	$x(2M-1)$
2	$x(2M)$	$x(2M+1)$	$x(2M+2)$	$\cdots$	$x(3M-1)$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$L-1$	$x((L-1)M)$	$x((L-1)M+1)$	$x((L-1)M+2)$	$\cdots$	$x(LM-1)$

$n = l + mL$

$l \backslash m$	0	1	2		$M-1$
0	$x(0)$	$x(L)$	$x(2L)$	$\cdots$	$x((M-1)L)$
1	$x(1)$	$x(L+1)$	$x(2L+1)$	$\cdots$	$x((M-1)L+1)$
2	$x(2)$	$x(L+2)$	$x(2L+2)$	$\cdots$	$x((M-1)L+2)$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$L-1$	$x(L-1)$	$x(2L-1)$	$x(3L-1)$	$\cdots$	$x(LM-1)$

- Row wise: $k = M\,p + q$
- Column wise: $k = q\,L + p$

Divide and Conquer Approach

- Using such double indices, and choosing column wise mapping for $x[n]$ and row wise mapping for $X[k]$, DFT equation becomes:

$$X[p,q] = \sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x[l,m] W_N^{(Mp+q)(mL+l)}$$

But,

$$W_N^{(Mp+q)(mL+l)} = W_N^{MLmp} W_N^{mLq} W_N^{Mpl} W_N^{lq}$$

and,

$$W_N^{Nmp} = 1,\ W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq},\ \text{and}\ W_N^{Mpl} = W_{N/M}^{pl} = W_L^{pl}$$

Using these,

$$X[p,q] = \sum_{l=0}^{L-1}\left\{W_N^{lq}\left[\sum_{m=0}^{M-1} x[l,m] W_M^{mq}\right]\right\} W_L^{lp}$$

Divide and Conquer Approach

- Thus we can compute N point DFT as M point and L point DFTs.
- First we compute M point DFTs

$$F[l,q] = \sum_{m=0}^{M-1} x[l,m] W_M^{\,mq} \quad 0 \le q \le M-1$$

for each row

- Then we compute a array $G[l, q]$ as:

$$G[l,q] = W_N^{\,lq} F[l,q]$$

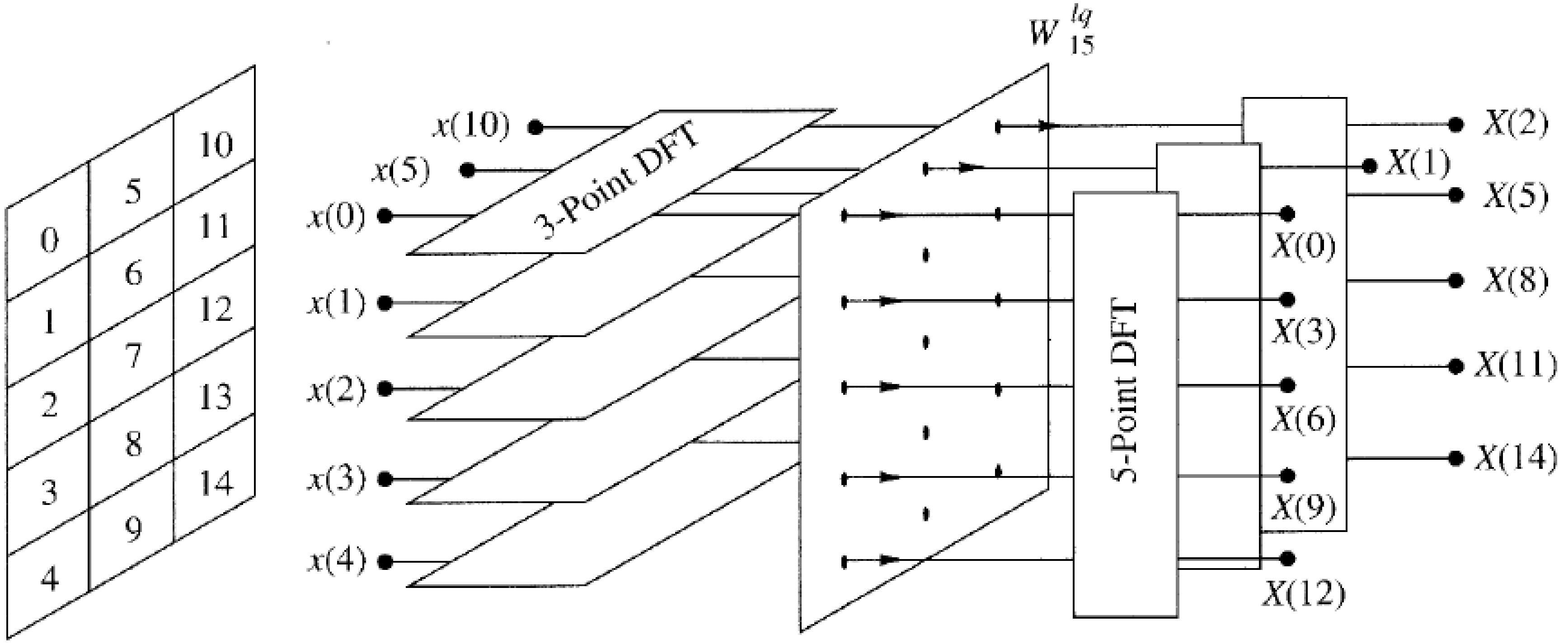
- Finally we compute L point DFTs

$$X[p,q] = \sum_{l=0}^{L-1} G[l,q] W_L^{\,lp}$$

for each column

Dive and Conquer-Illustrated

- For N=15, L=5 and M=3



- When N is highly composite, the principle can be repeated until we reach prime numbers.

Radix-2 Algorithms

- When N is highly composite and can be expressed as

$$N = r^{\upsilon} = r \times r \times r \times r.........$$

the repetitive divide and conquer principle has a regular pattern and is called radix *r* FFT.

- A special case is when *r*=2, and is called radix-2 FFT.

- In such case one factor *L* or *M* is always 2.

Decimation in Time Radix-2 Algorithm

- Let L=2 and M=N/2. So $x[n]$ is separated in two sequences with N/2 samples each. One sequence has even numbered samples while other has odd numbered samples.

$$f_1(n) = x(2n)$$

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

- Now, N point DFT of $x[n]$ is:

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

$$= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn}$$

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

Decimation in Time Radix-2 Algorithm

- Since, $W_N^2 = W_{N/2}$,

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

$$= F_1(k) + W_N^k F_2(k), \qquad k = 0, 1, \ldots, N-1$$

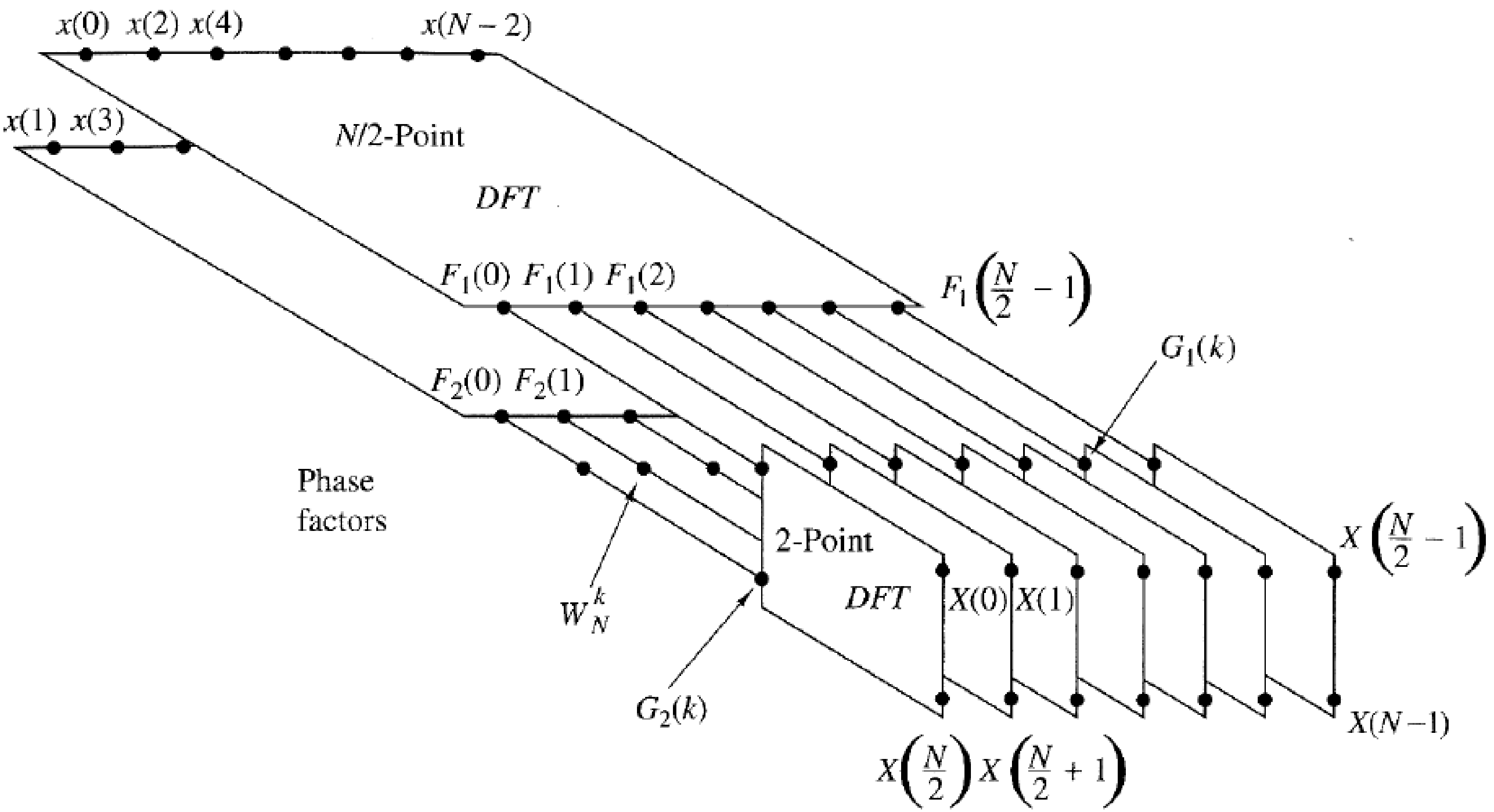
where, $F_1[k]$ *and* $F_2[k]$ are $N/2$ point DFTs of $f_1[n]$ *and* $f_2[n]$

- using periodicity property of $F_1[k]$ *and* $F_2[k]$,

$$X(k) = F_1(k) + W_N^k F_2(k), \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k), \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

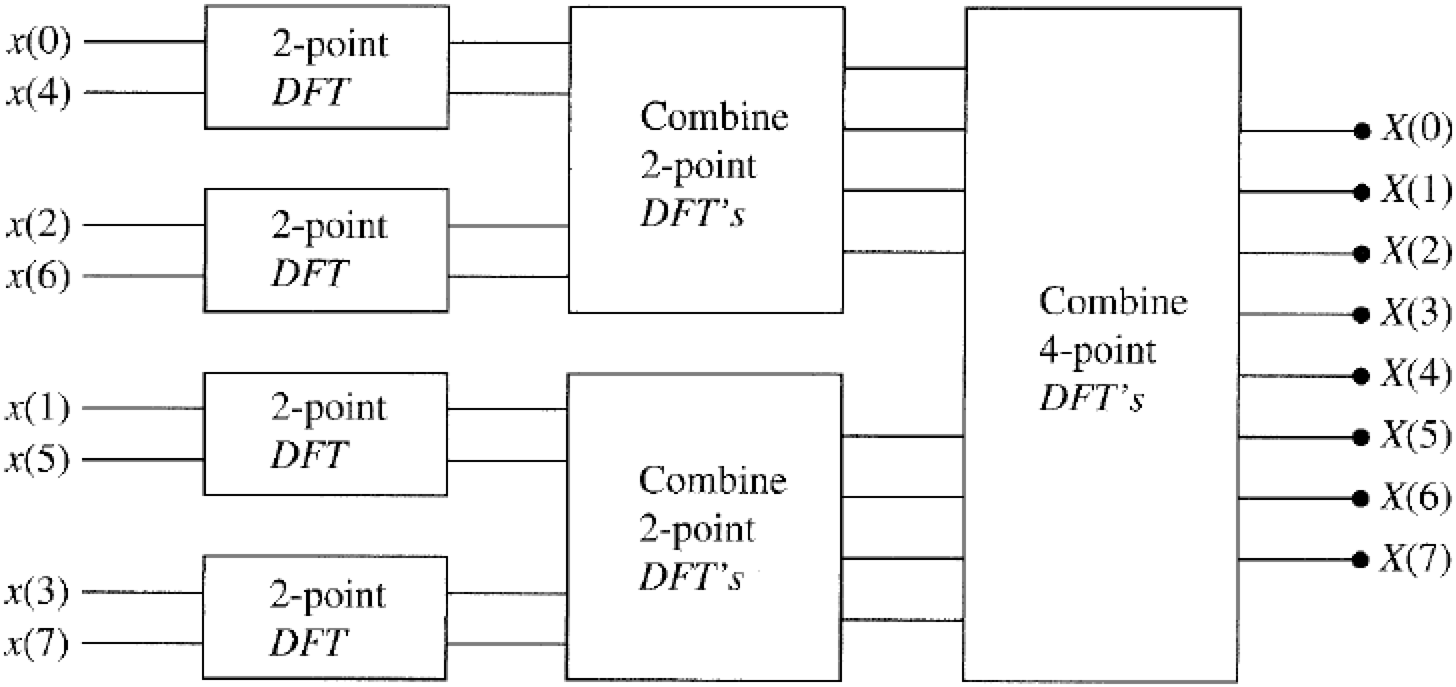
Decimation in Time Radix-2 Algorithm



Decimation in Time Radix-2 Algorithm

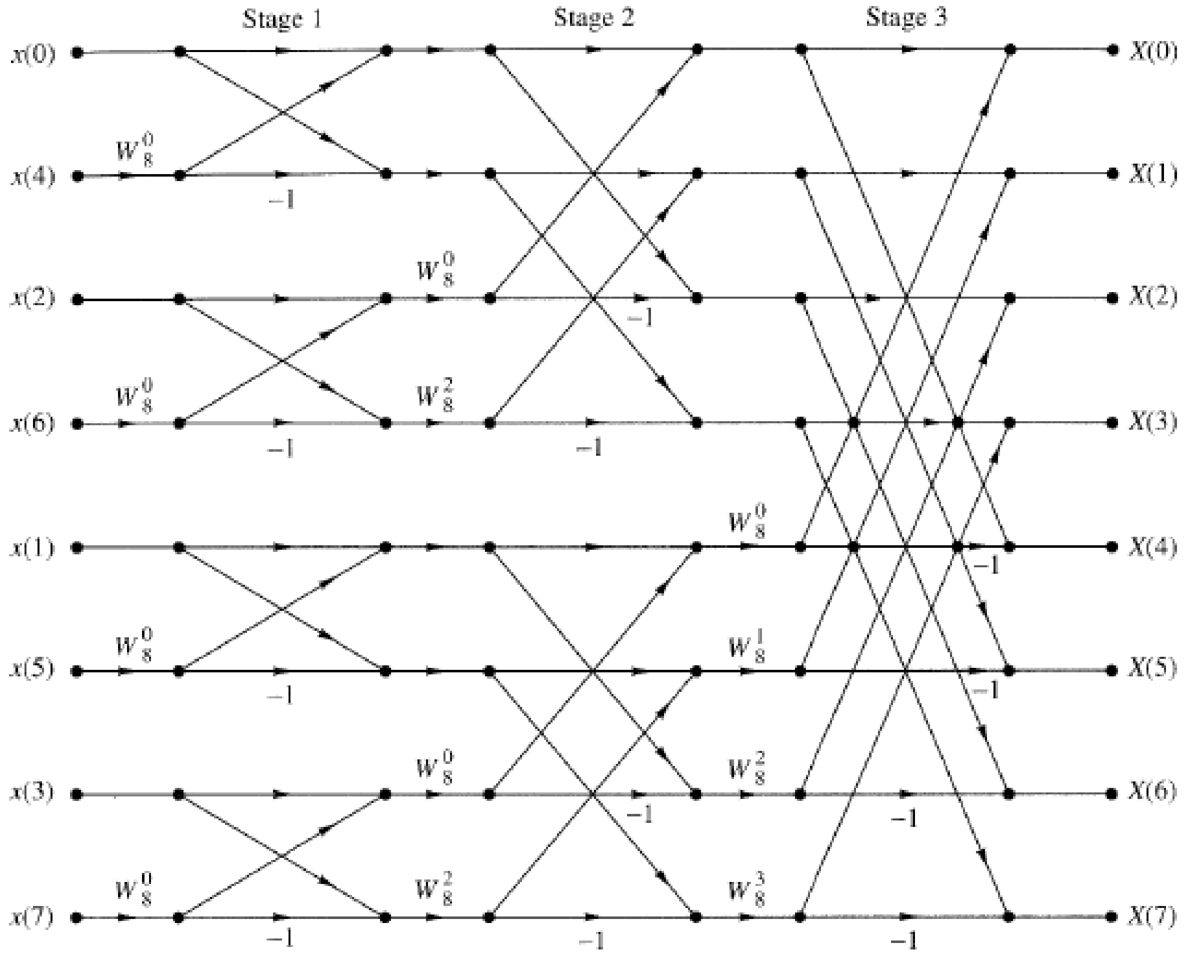
- Repeating the process, $N/2$ point DFTs can be computed using $N/4$ point DFTs.

- Finally we will reach to two point DFT. For N=8 process is shown:



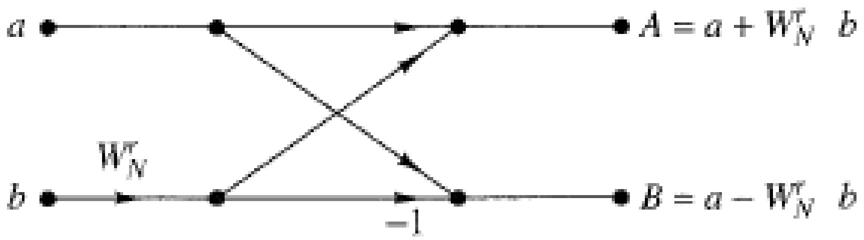
Decimation in Time Radix-2 Algorithm

- The exact operations involved is shown below and is called a butterfly diagram for DIT radix-2 FFT.



Decimation in Time Radix-2 Algorithm

- The basic computation involved is called basic butterfly.



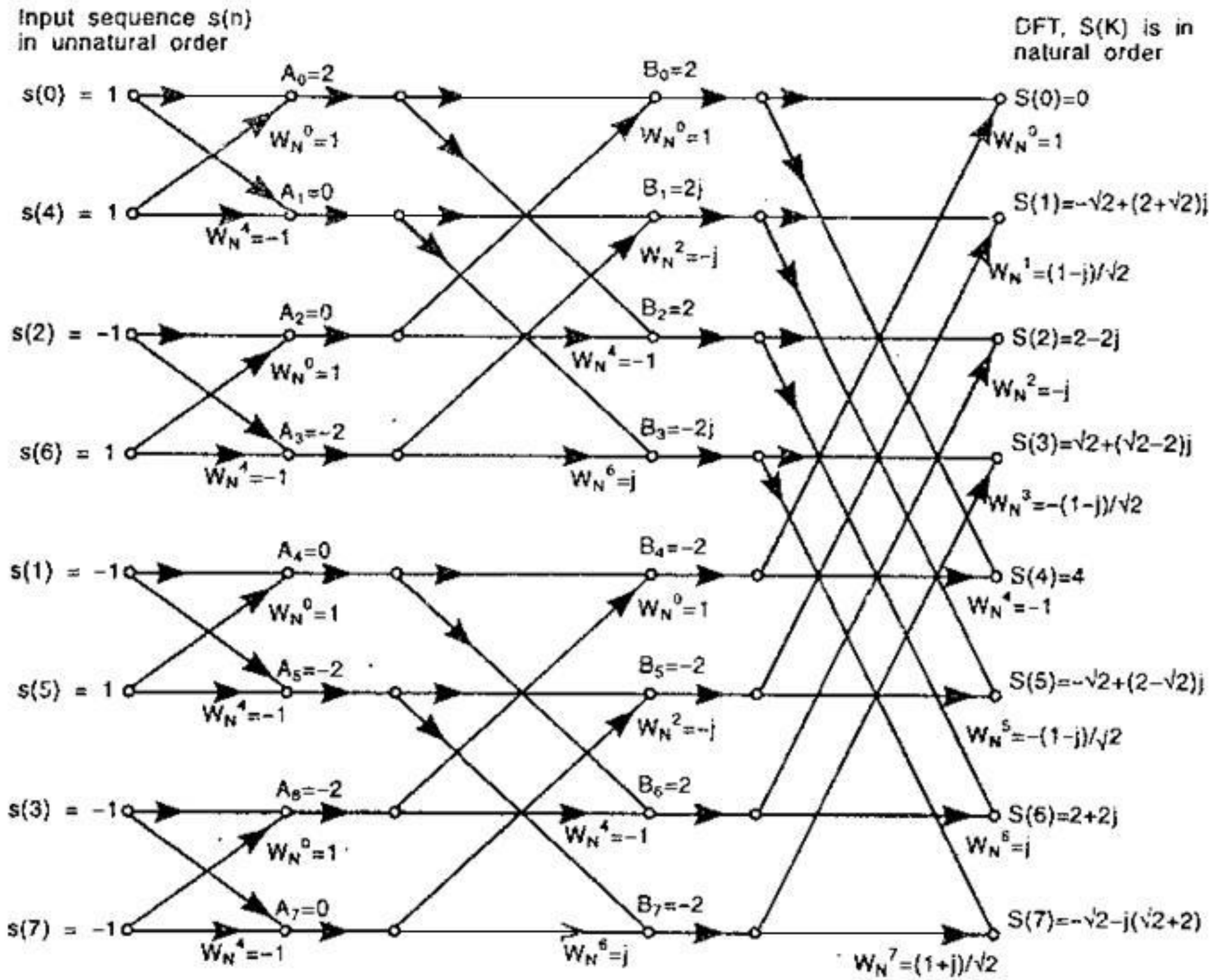
- The sequence $x[n]$ is first altered in order while the computed DFT values are in order. So called DIT.

- For N point DFT, number of decimation is υ-1. (for N=8, 2 decimations)

- Total complex multiplications is $(N/2)\log_2 N$ and complex additions is $N \log_2 N$

Decimation in Time -example

- $x[n] = \{1,-1,-1,-1,1,1,1,-1\}$



Decimation in Frequency

- Let M=2 and L=N/2. So $x[n]$ is separated in two sequences with N/2 samples each. One sequence has first N/2 samples while other has last N/2 samples.

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

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

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

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

Decimation In Frequency

- Decimating $X[k]$ into even and odd numbered samples,

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

And,

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

Also,

$$X(2k) = \sum_{n=0}^{(N/2)-1} g_1(n) W_{N/2}^{kn}$$

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} g_2(n) W_{N/2}^{kn}$$

where,

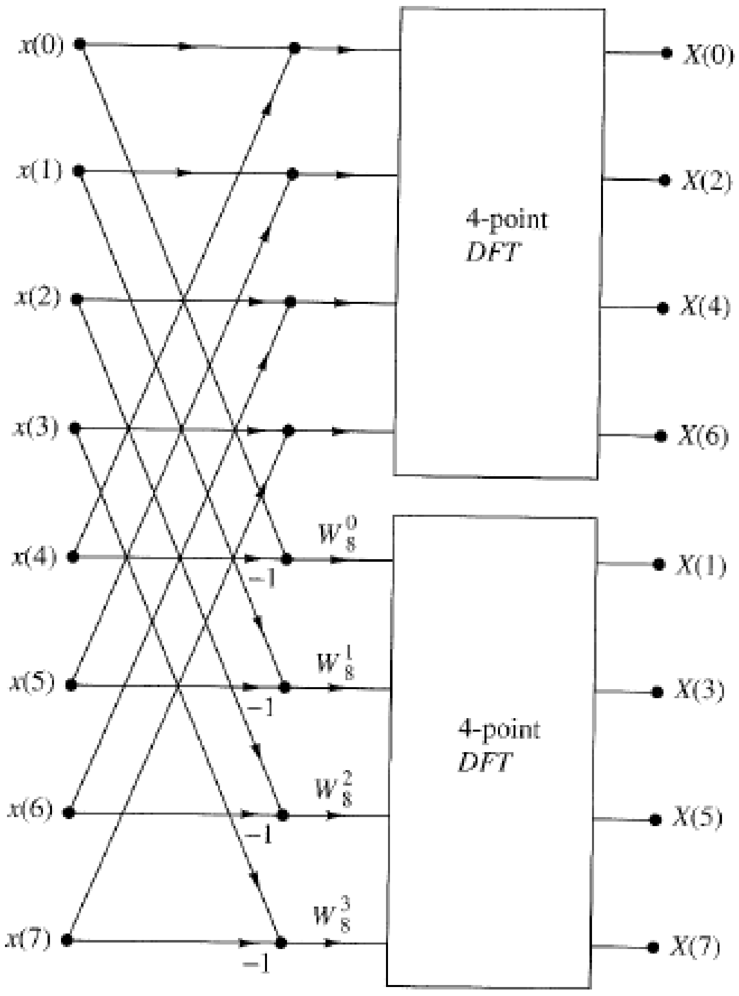
$$g_1(n) = x(n) + x\left(n + \frac{N}{2}\right)$$

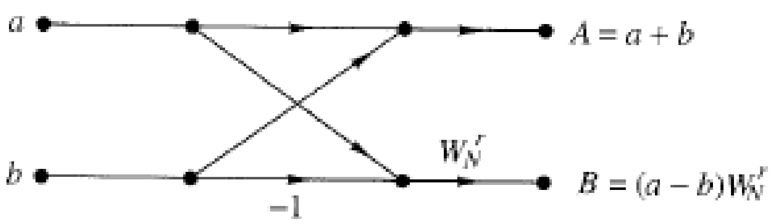
$$g_2(n) = \left[x(n) - x\left(n + \frac{N}{2}\right)\right] W_N^n,$$

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

Decimation In Frequency-Butterfly

- Time sequence in normal order, DFT in decimated order





DIF Butterfly for N=8