MODULE-II
DISCRETE AND FAST FOURIER TRANSFORMS

DISCRETE FOURIER TRANSFORM (DFT):
Fourier series representation of finite duration sequence is known as Discrete Fourier Transform (DFT). It is used to obtain discrete frequency domain representation of a given discrete time domain representation.

COMPUTATION OF DFT:
Discrete frequency domain representation of a given finite duration sequence x(n) of length N over the range $0 \le n \le N-1$ can be defined as

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

Where,

X(k) is Discrete frequency domain of x(n) and range of k is $0 \le k \le N-1$.

$W_N^{nk} = e^{-j\frac{2\pi nk}{N}}$ is Phase Factor or Twidle Factor or Complex Quantity.

N is duration of x(n) or X(k)

COMPUTATION OF IDFT:
Inverse Discrete Fourier Transform of X(k) of length N over the range $0 \le k \le N-1$ can be defined as

$$IDFT[X(k)] = x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$$

RELATION BETWEEN DTFT & DFT:

From basic definition of DFT is

$$\begin{aligned} DFT[x(n)] &= \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \\ &= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi nk}{N}} \\ &= \sum_{n=0}^{N-1} x(n)\, e^{-j\left(\frac{2\pi k}{N}\right)n} \\ &= DTFT[x(n)] \end{aligned}$$

DFT and DTFT are equal, if $2\pi k / N = \omega$

PROPERTIES OF PHASE FACTOR:

Property 1 : $W_N^0 = 1$

Property 2 : $W_N^N = 1$

Property 3 : $W_N^{nk \pm N} = W_N^{nk}$

Property 4 : $W_N^{n_1 k}\, W_N^{n_2 k} = W_N^{(n_1 + n_2)k}$

Property 5 : $\left| W_N^{nk} \right| = 1$

PROPERTIES OF DFT:

LINEARITY PROPERTY:

Let $x_1(n)$, $x_2(n)$ are two finite duration sequences, with a equal duration of N samples and $DFT[x1(n)] = X_1(k)$, $DFT[x2(n)] = X_2(k)$, then according to linear property of DFT,
$DFT[\, a\, x_1(n) + b\, x_2(n)\,] = a\, X_1(k) + b\, X_2(k)$. Where a & b are arbitrary constants

PROOF:

From basic definition of DFT

$$DFT[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by $a\, x_1(n) + b\, x_2(n)$

$$\begin{aligned}
DFT[\,a\, x_1(n) + b\, x_2(n)\,] &= \sum_{n=0}^{N-1} [\,a\, x_1(n) + b\, x_2(n)\,]\, W_N^{nk} \\
&= \sum_{n=0}^{N-1} \left[a\, x_1(n)\, W_N^{nk} + b\, x_2(n)\; W_N^{nk} \right] \\
&= \sum_{n=0}^{N-1} \left[a\, x_1(n)\, W_N^{nk} \right] + \sum_{n=0}^{N-1} \left[\; x_2(n)\; W_N^{nk} \right] \\
&= a \sum_{n=0}^{N-1} x_1(n)\; W_N^{nk} + b \sum_{n=0}^{N-1} x_2(n)\; W_N^{nk} \\
&= a\, DFT[\,x_1(n)\,] + b\, DFT[\,x_2(n)\,] \\
&= a\, X_1(k) + b\, X_2(k)
\end{aligned}$$

PERIODIC PROPERTY:

Let x(n) be a finite duration sequence, with a duration of N samples and DFT[x(n)] = X(k), then according to periodic property of DFT,

(a) $X(N + k) = X(k)$

(b) $x(N + n) = x(n)$

PROOF:

From basic definition of DFT

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

Replace k by N + k

$$\begin{aligned}
X(N + k) &= \sum_{n=0}^{N-1} x(n)\, W_N^{n(N+k)} \\
&= \sum_{n=0}^{N-1} x(n)\, W_N^{(nN+nk)} \\
&= \sum_{n=0}^{N-1} x(n)\, W_N^{nN}\, W_N^{nk} \\
&= \sum_{n=0}^{N-1} x(n)\;(1)\, W_N^{nk} \\
&= \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \\
&= X(k)
\end{aligned}$$

$$\mathrm{DFT}[W_N^{-nk_0}x(n)] = \sum_{n=0}^{N-1} W_N^{-nk_0} x(n) W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x(n) \; W_N^{nk} W_N^{-nk_0}$$

$$= \sum_{n=0}^{N-1} x(n) \; W_N^{n(k-k_0)}$$

$$= \mathrm{DFT}[x(n)] \text{ at } k = k - k_0.$$

$$= X(k - k_0)$$

TIME REVERSAL PROPERTY:

Let x (n) be a finite duration sequence, with a duration of N samples and DFT[x(n)] = X(k), then according to time reversal property of DFT, DFT[x(N – n)] = X(N – k).

PROOF:

From basic definition of DFT

$$\mathrm{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by x(N – n)

$$\mathrm{DFT}[x(N-n)] = \sum_{n=0}^{N-1} x(N-n)\, W_N^{nk} \quad \text{, Let } N - n = m$$

$$= \sum_{m=N}^{1} x(m)\, W_N^{(N-m)k}$$

$$= \sum_{m=N}^{1} x(m)\, W_N^{Nk} \; W_N^{-mk}$$

$$= \sum_{m=N}^{1} x(m)\,(1)\; W_N^{-mk}$$

$$= \sum_{m=N}^{1} x(m) \; W_N^{-mk}(1)$$

$$= \sum_{m=0}^{N-1} x(m) \; W_N^{-mk} \; W_N^{mN}$$

$$= \sum_{m=0}^{N-1} x(m) \; W_N^{m(N-k)}$$

$$= \mathrm{DFT}[x(n)] \text{ at } k = N - k.$$

$$= X(N-k)$$

CONJUGATE PROPERTY:

Let x(n) be a finite duration sequence, with a duration of N samples and DFT[x(n)] = X(k), then according to conjugate property of DFT,

(a) DFT[x*(n)] = X*(N – k).

(b) DFT[x*(N-n)] = X*(N + k).

PROOF:

(a) From basic definition of DFT

$$DFT[x(n)] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by x*(n)

$$\begin{aligned} DFT[x^*(n)] &= \sum_{n=0}^{N-1} x^*(n)\, W_N^{nk} \\ &= \sum_{n=0}^{N-1} x^*(n) \left[W_N^{-nk} \right]^* \\ &= \sum_{n=0}^{N-1} \left[x(n)\, W_N^{-nk} \right]^* \\ &= \left[\sum_{n=0}^{N-1} x(n)\, W_N^{-nk} \right]^* \end{aligned}$$

$$\begin{aligned} DFT[x^*(n)] &= \left[\sum_{n=0}^{N-1} x(n)\,(1)\, W_N^{-nk} \right]^* \\ &= \left[\sum_{n=0}^{N-1} x(n)\,(W_N^{nN})\, W_N^{-nk} \right]^* \\ &= \left[\sum_{n=0}^{N-1} x(n)\; W_N^{n(N-k)} \right]^* \\ &= [\,DFT \text{ of } x(n) \text{ at } k = N-k\,]^* \\ &= [\,X(N-k)\,]^* \\ &= X^*(N-k) \end{aligned}$$

(b) From basic definition of DFT

$$DFT[x(n)] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by x*(N-n)

$$DFT[\,x^*(N-n)\,] = \sum_{n=0}^{N-1} x^*(N-n)\, W_N^{nk} \quad \text{, Let } N-n=m$$

$$= \sum_{m=N}^{1} x^*(m)\, W_N^{(N-m)k}$$

$$= \sum_{m=1}^{N} x^*(m)\, W_N^{Nk}\; W_N^{-mk}$$

$$= \sum_{m=0}^{N-1} x^*(m)\,(1)\; W_N^{-mk}$$

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

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

$$= \left[\sum_{n=0}^{N-1} x(n)\, W_N^{nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\,(1)\, W_N^{nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\,(W_N^{nN})\, W_N^{nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\; W_N^{n(N+k)} \right]^*$$

$$= [\,\text{DFT of } x(n) \text{ at } k = N+k\,]^*$$

$$= [\,X(N+k)\,]^*$$

$$= X^*(N+k)$$

PARSEVALLS THEOREM:

Let x(n) be a finite duration sequence, with a duration of N samples and N-Point DFT[x(n)] = X(k), then Parsevalls theorem provides the relation between x(n) and its frequency domain X(k) as

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

PROOF:

$$\text{LHS} = \sum_{n=0}^{N-1} |x(n)|^2$$

$$= \sum_{n=0}^{N-1} x(n) \left[x(n)\right]^*$$

$$= \sum_{n=0}^{N-1} x(n) \left[\text{IDFT}[X(k)]\right]^*$$

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

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

Change the order of two sums

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

$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\, \text{DFT}[x(n)]$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\, X(k)$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$$

$$= \text{RHS}$$

TIME CONVOLUTION THEOREM:

Let $x_1(n)$, $x_2(n)$ are two finite duration sequences, with a equal duration of N samples and DFT[$x1(n)$] = $X_1(k)$, DFT[$x2(n)$] = $X_2(k)$, then according to time convolution theorem, DFT[$x_1(n) \otimes x_2(n)$] = $X_1(k)\ X_2(k)$. i.e "Convolution in time domain leads to multiplication in frequency domain"

PROOF:
From basic definition of DFT

$$\text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace $x(n)$ by $x_1(n) \otimes x_2(n)$

$$\text{DFT}[x_1(n) \otimes x_2(n)] = \sum_{n=0}^{N-1} [x_1(n) \otimes x_2(n)]\, W_N^{nk}$$

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

Change the order of summation

$$
\begin{aligned}
DFT[\, x_1(n) \otimes x_2(n)] &= \sum_{m=0}^{N-1} \left[x_1(m) \left(\sum_{n=0}^{N-1} x_2(n-m)] \quad W_N^{nk} \right) \right] \\
&= \sum_{m=0}^{N-1} \left[x_1(m) \; (DFT(x_2(n-m)) \right] \\
&= \sum_{m=0}^{N-1} \left[x_1(m) \left(W_N^{mk} \; X_2(K) \right) \right] \\
&= X_2(K) \sum_{m=0}^{N-1} \left[x_1(m) \, W_N^{mk} \right] \\
&= X_2(K) \; X_1(K) \\
&= X_1(K) X_2(K)
\end{aligned}
$$

FREQUENCY CONVOLUTION THEOREM :

Let $x_1(n)$, $x_2(n)$ are two finite duration sequences, with a equal duration of N samples and DFT[$x_1(n)$] = $X_1(k)$, DFT[$x_2(n)$] = $X_2(k)$, then according to frequency convolution theorem, DFT[$x_1(n)\, x_2(n)$] = [$X_1(k) \otimes X_2(k)$] / N. i.e "Convolution in frequency domain leads to multiplication in time domain"

PROOF:

From basic definition of IDFT

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

Replace X(k) by $X_1(k) \otimes X_2(k)$

$$
\begin{aligned}
IDFT[\, X_1(k) \otimes X_2(k)\,] &= \frac{1}{N} \sum_{k=0}^{N-1} [\, X_1(k) \otimes X_2(k)\,]\, W_N^{-nk} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} [\, X_1(m) X_2(k-m)] \right) W_N^{-nk}
\end{aligned}
$$

Change the order of two sums

$$
\begin{aligned}
IDFT[\, X_1(k) \otimes X_2(k)\,] &= \frac{1}{N} \sum_{m=0}^{N-1} \left(X_1(m) \sum_{k=0}^{N-1} X_2(k-m)\, W_N^{-nk} \right) \\
&= \frac{1}{N} \sum_{m=0}^{N-1} \left(X_1(m)\, N \left(\frac{1}{N} \sum_{k=0}^{N-1} X_2(k-m)\, W_N^{-nk} \right) \right) \\
&= \frac{1}{N} \sum_{m=0}^{N-1} (X_1(m)\, N\, (IDFT(X_2(k-m))) \\
&= \frac{1}{N} \sum_{m=0}^{N-1} \left(X_1(m)\, N \left(W_N^{-mn}\, x_2(n) \right) \right) \\
&= N\, x_2(n) \; \frac{1}{N} \sum_{m=0}^{N-1} \left(X_1(m)\, W_N^{-mn} \right) \\
&= N\, x_1(n) \; x_2(n)
\end{aligned}
$$

$$
DFT[\, x_1(n) \; x_2(n)\,] = \frac{X_1(k) \otimes X_2(k)}{N}
$$

LINEAR FILTERING METHODS BASED ON DFT :

There are two methods using linear filtering based DFT.

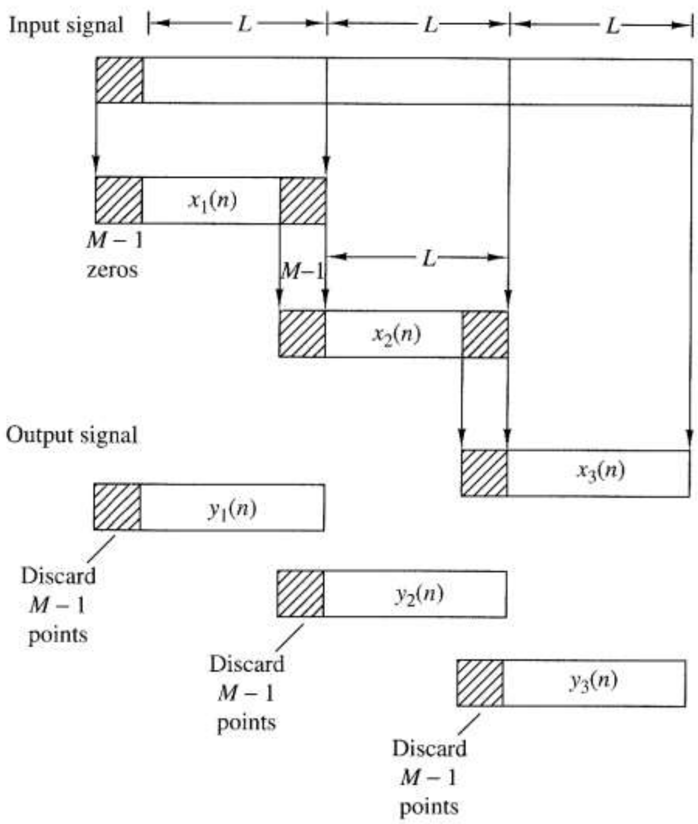
1. Overlap save method
2. Overlap add method

Overlap-save method. In this method the size of the input data blocks is $N = L + M - 1$ and the DFTs and IDFT are of length N. Each data block consists of the last $M-1$ data points of the previous data block followed by L new data points to form a data sequence of length $N = L + M - 1$. An N-point DFT is computed for each data block. The impulse response of the FIR filter is increased in length by appending $L-1$ zeros and an N-point DFT of the sequence is computed once and stored. The multiplication of the two N-point DFTs $\{H(k)\}$ and $\{X_m(k)\}$ for the mth block of data yields

$$x_1(n) = \{\underbrace{0, 0, \ldots, 0,}_{M-1 \text{ points}} x(0), x(1), \ldots, x(L-1)\}$$

$$x_2(n) = \{\underbrace{x(L-M+1), \ldots, x(L-1),}_{M-1 \text{data points from } x_1(n)} \underbrace{x(L), \ldots, x(2L-1)}_{L \text{ new data points}}\}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots, x(2L-1),}_{M-1 \text{ data points from } x_2(n)} \underbrace{x(2L), \ldots, x(3L-1)}_{L \text{ new data points}}\}$$

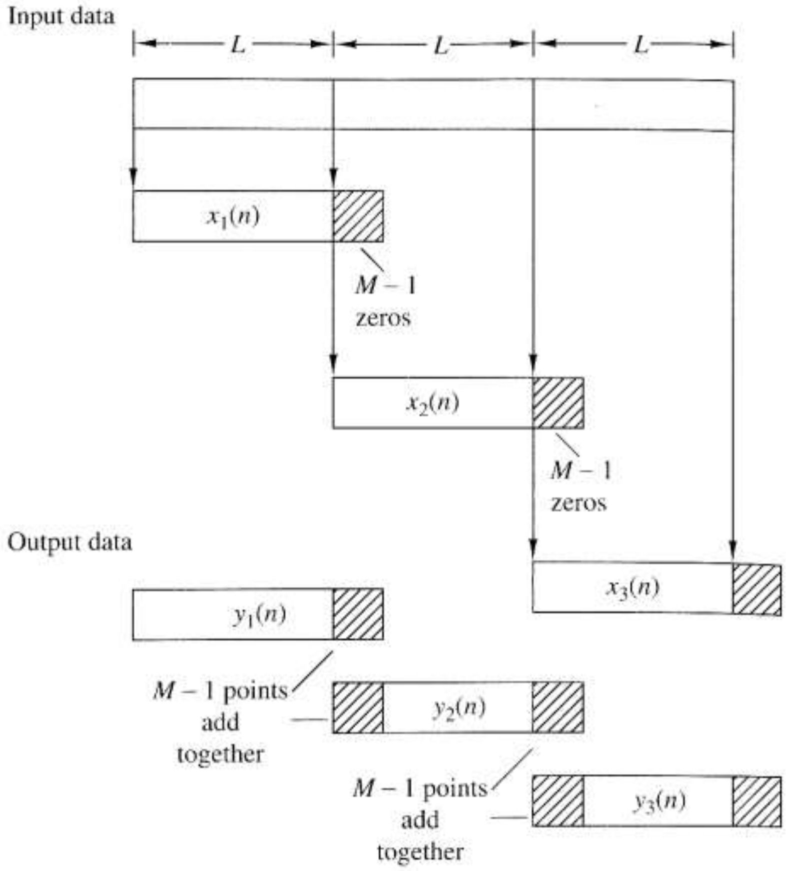


Overlap-add method. In this method the size of the input data block is L points and the size of the DFTs and IDFT is $N = L+M-1$. To each data block we append $M-1$ zeros and compute the N-point DFT. Thus the data blocks may be represented as

$$x_1(n) = \{x(0), x(1), \ldots, x(L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

$$x_2(n) = \{x(L), x(L+1), \ldots, x(2L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

$$x_3(n) = \{x(2L), \ldots, x(3L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$



There are two types of convolution

- Linear Convolution
- Circular Convolution

LINEAR CONVOLUTION:

Linear convolution of a first sequence $x_1(n)$ having N_1 samples $[0 \le n \le N_1 - 1]$ and a second sequence $x_2(n)$ having N_2 samples $[0 \le n \le N_2 - 1]$ can be defined as

$$x(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

Where

$x(n)$: Linear convoluted sequence, with a duration of $N = N_1 + N_2 - 1$

N_1 : Duration of $x_1(n)$, $0 \le n \le N_1 - 1$

N_2 : Duration of $x_2(n)$, $0 \le n \le N_2 - 1$

N : Duration of linear convoluted sequence $x(n)$,

Duration of linear convoluted sequence $x(n)$ and $x_1(n)$ or $x_2(n)$ are different, therefore linear convolution is also known as aperiodic convolution. DFT does not support linear convolution, due to unequal durations.

Procedure for Evaluating Linear Convolution:

The following four steps were required to compute linear convolution

1. Folding : Fold $x_2(m)$ about k=0, to obtain $x_2(-m)$
2. Shifting : Shift the folded sequence $x_2(-m)$ by n units left and/or right, to obtain $x_2(n-m)$.
3. Multiplication : Multiply $x_1(m)$ and $x_2(n-m)$, to obtain the product sequence $x_1(m) . x_2(n-m)$,
4. Summation : Sum all the values of product sequence at every instant, to obtain $x(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$

CIRCULAR CONVOLUTION:

Circular convolution of a first sequence $x_1(n)$ having N samples $[0 \le n \le N - 1]$ and a second sequence $x_2(n)$ having N samples $[0 \le n \le N - 1]$ can be defined as

$$x(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

Where

$x(n)$: Circular convoluted sequence, with a duration of $N = N - 1$

N : Duration of $x_1(n)$ or $x_2(n)$ or $x(n)$, $0 \le n \le N - 1$

Durations of circular convoluted sequence $x(n)$, first sequence $x_1(n)$ and second sequence $x_2(n)$ are equal, therefore circular convolution is also known as periodic convolution. DFT supports circular convolution, due to equal durations.

Procedure for Evaluating Circular Convolution:

The following four steps were required to compute circular convolution

1. Folding : Fold $x_2(m)$ about k=0 and take periodic extension, to obtain $x_2(-m)$
2. Shifting : Shift the folded sequence $x_2(-m)$ by n units left and/or right, to obtain $x_2(n-m)$.
3. Multiplication : Multiply $x_1(m)$ and $x_2(n-m)$, to obtain the product sequence $x_1(m) . x_2(n-m)$,
4. Summation : Sum all the values of product sequence at every instant, to obtain $x(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$

Derivation for Circular Convolution:

Let $x_1(n)$ and $x_2(n)$ are two finite duration sequences with a equal duration of N samples, assume $x(n)$ be the circular convoluted sequence with a duration of N samples

$x(n) = x_1(n) \otimes x_2(n)$, convolution in time domain leads to multiplication in frequency domain. i.e $X(k) = X_1(k)\ X_2(k)$.

IDFT of X(k) can be defined as

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

Replace X(k) = $X_1(k)\ \ X_2(k)$

$$IDFT[X(k)] = \frac{1}{N}\sum_{k=0}^{N-1} X_1(k)\, X_2(k) W_N^{-nk}$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_1(k)\,]\,DFT[x_2(n)\,]\, W_N^{-nk}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left[X_1(k) \left(\sum_{m=0}^{N-1} x_2(m)\, W_N^{mk} \right) W_N^{-nk} \right]$$

Change the order of two sums

$$x(n) = \frac{1}{N}\sum_{m=0}^{N-1}\left[x_2(m) \left(\sum_{k=0}^{N-1} X_1(k) W_N^{mk}\, W_N^{-nk} \right) \right]$$

$$= \sum_{m=0}^{N-1}\left[x_2(m) \left(\frac{1}{N}\sum_{k=0}^{N-1} X_1(k) W_N^{-(n-m)k} \right) \right]$$

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

LINEAR CONVOLUTION THROUGH CIRCULAR CONVOLUTION:

Let $x_1(n)$ and $x_2(n)$ are two finite duration sequences, with a duration of $N_1$and N_2 samples, then the duration of linear convoluted sequence $N_1 + N_2 – 1$. Following steps were required to compute the linear convolution through circular convolution

Pad the first sequences $x_1(n)$ by "$N_2 – 1$" number of zeroes, to obtain $N_1 + N_2 – 1$ length.

1. Pad the second sequences $x_2(n)$ by "$N_1 – 1$" number of zeroes, to obtain $N_1+N_2–1$ length.
2. Now determine the circular convolution of $x_1(n)$ and $x_2(n)$ called linear convolution.

RESPONSE OF DISCRETE LTI SYSTEM THROUGH CIRCULAR CONVOLUTION:

Response of the system can be defined as the Linear convolution of input x(n) and impulse response h(n). Let N_1 be the duration of input sequence x(n) and N_2 be the duration of impulse response h(n), then the duration of response of the system is $N_1+N_2–1$. Following steps were required to compute the response of the discrete LTI system through circular convolution

1. Pad the input sequences x (n) by "$N_2 – 1$" number of zeroes, to obtain $N_1 + N_2 – 1$ length.
2. Pad the impulse response sequences h (n) by "$N_1 – 1$" number of zeroes, to obtain $N_1+N_2–1$ length.
3. Now determine the circular convolution of x (n) and h(n) called Response of the given discrete LTI system.

CIRCULAR CONVOLUTION THROUGH DFT & IDFT:

Let $x_1(n)$ and $x_2(n)$ are two finite duration sequences, with a equal duration of N samples, then the duration of circular convoluted sequence is also N samples. Following steps were required to compute the circular convolution through DFT approach.

1. Compute N –Point DFT of $x_1(n)$, i.e. $X_1(k)$.
2. Compute N –Point DFT of $x_2(n)$, i.e. is $X_2(k)$.
3. Now determine N samples of X(k) from X(k) = $X_1(k)$. $X_2(k)$.
4. Compute N –Point IDFT of X(k), i.e. x(n) called circular convoluted sequence

FREQUENCY ANALYSIS OF SIGNALS USING DFT:

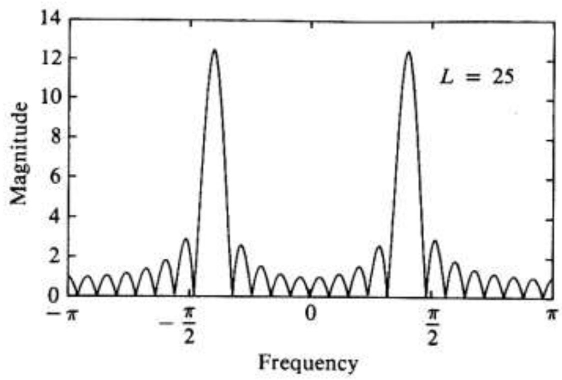
If the signal to be analyzed is an analog signal, we would first pass it through an antialiasing filter and then sample it at a rate $F_s \geq 2B$, where B is the bandwidth of the filtered signal. Thus the highest frequency that is contained in the sampled signal is $F_s/2$. Finally, for practical purposes, we limit the duration of the signal to the time interval $T_0 = LT$, where L is the number of samples and T is the sample interval. As we shall observe in the following discussion, the finite observation interval for the signal places a limit on the frequency resolution; that is, it limits our ability to distinguish two frequency components that are separated by less than $1/T_0 = 1/LT$ in frequency.

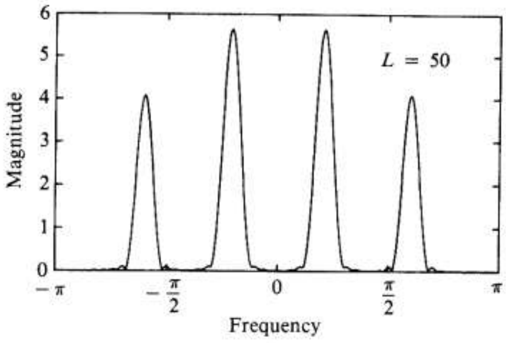
Let $\{x(n)\}$ denote the sequence to be analyzed. Limiting the duration of the sequence to L samples, in the interval $0 \leq n \leq L-1$, is equivalent to multiplying $\{x(n)\}$ by a rectangular window $w(n)$ of length L. That is,

$$\hat{x}(n) = x(n)w(n)$$

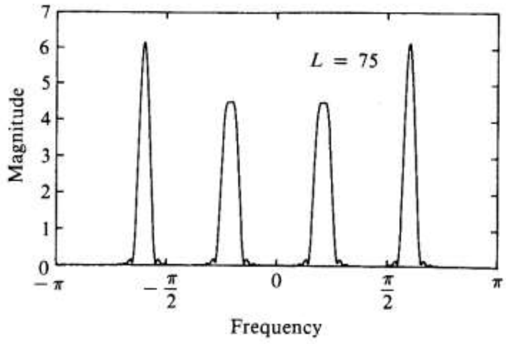
where

$$w(n) = \begin{cases} 1, & 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases}$$

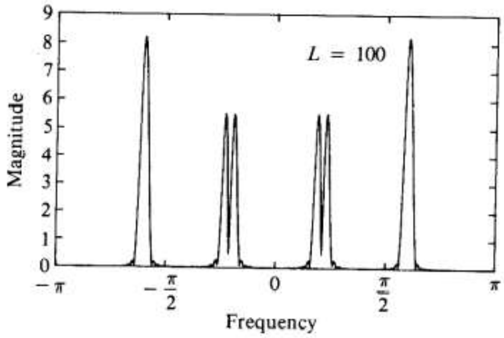




(a)



(b)



FAST FOURIER TRANSFORM (FFT):

INTRODUCTION:

Fast Fourier Transform is a method or algorithm is used to compute Discrete Fourier Transform with reduced number of calculations (Complex additions & multipliers). From basic definition of DFT of a N – point sequence x(n) is

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

$$X(k) = x(0)\, W_N^{0k} + x(1)\, W_N^{1k} + x(2)\, W_N^{2k} + \ldots\ldots\ldots\ldots + x(N-1)\, W_N^{(N-1)k}$$

$$k = 0 \Rightarrow X(0) = x(0)\, W_N^{0} + x(1)\, W_N^{0} + x(2)\, W_N^{0} + \ldots\ldots\ldots\ldots + x(N-1)\, W_N^{0}$$

$$k = 1 \Rightarrow X(1) = x(0)\, W_N^{0} + x(1)\, W_N^{1} + x(2)\, W_N^{2} + \ldots\ldots\ldots\ldots + x(N-1)\, W_N^{N-1}$$

$$k = 2 \Rightarrow X(2) = x(0)\, W_N^{0} + x(1)\, W_N^{2} + x(2)\, W_N^{4} + \ldots\ldots\ldots\ldots + x(N-1)\, W_N^{(N-1)2}$$

.

.

.

$$k = N-1 \Rightarrow X(N-1) = x(0)\, W_N^{0} + x(1)\, W_N^{N-1} + x(2)\, W_N^{2(N-1)} + \ldots\ldots + x(N-1)\, W_N^{(N-1)(N-1)}$$

It is very clear from above N-point DFT computation, for each line requires "N" number of complex multiplications and "N – 1" number of complex additions. Such lines are N, therefore for the complete computation of N-point DFT

- Number of Complex additions used = N(N – 1).
- Number of Complex multiplications used = N^2.

To reduce above number of calculations, FFT technique is used. In a FFT method, N can be expressed as $N = r^m$.

Where, r : Radix number (minimum value is 2).

m : Required number of stages to compute N-Point DFT

In a FFT method,

- Number of Complex additions used = $N \log_2 N$.
- Number of Complex multiplications used = $(N/2) \log_2 N$.

N – point DFT – FFT is a indirect method, where the total computation is divided into m number of stages and all these m stages are cascaded to compute N-point DFT. The total computation consists of

Two N/2-point DFT's

Four N/4-point DFT's

Eight N/8-point DFT's and so on.

The process of converting N-point DFT into smaller point DFT,s is known as decimation. The decimation process may starts from time domain or from frequency domain. Based on the decimation in time or in frequency, FFT algorithms are classified into two types.

1. Decimation in Time (DIT) Radix-2 FFT algorithm.
2. Decimation in Frequency (DIF) Radix-2 FFT algorithm.

COMPARISON OF DFT & FFT:

No. of points N	Direct Computation of N-Point DFT		Indirect Computation of N-Point DFT (DIT Radix 2 – FFT / DIF Radix 2 – FFT)		
	Complex Additions Required $A=N(N-1)$	Complex Multiplications Required $B=N^2$	No. of Stages Required $m=\log_2 N$	Complex Additions Required $a=N.m$	Complex Multiplications Required $b=Nm/2$
4	12	16	2	8	4
8	56	64	3	24	12
16	240	256	4	64	32
32	992	1,024	5	160	80
64	4,032	4,096	6	384	192
128	16,256	16,384	7	896	448
256	65,280	65,536	8	2,048	1,024
512	2,61,632	2,62,144	9	4,608	2,304
1024	10,47,552	10,48,576	10	10,240	5,120

No. of points N	% of saving due to		Speed of FFT due to	
	Complex Additions (1 – a / A) x 100	Complex Multiplications (1 – b / B) x 100	Complex Additions A / a	Complex Multiplications B / b
4	33.33%	75.00%	1.50	4.00
8	57.14%	81.25%	2.33	5.33
16	73.33%	87.50%	3.75	8.00
32	83.87%	92.19%	6.20	12.80
64	90.48%	95.31%	10.50	21.33
128	94.49%	97.27%	18.14	36.57
256	96.86%	98.44%	31.86	64.00
512	98.24%	99.12%	56.78	113.78
1024	99.02%	99.51%	102.3	204.8

From above two tables it can be observed that for a large value of N (if N increases),

- Number of calculations decreases.
- The percentage saving increases.
- Speed of FFT method increases.

DECIMATION IN TIME (DIT) RADIX – 2 FFT ALGORITHM:

In DIT Radix – 2 FFT, the time domain N-point sequence x(n) is decimated into possible number of 2-point sequences. Now compute possible number of 4-point DFT's by combining two 2-point DFT's. Then compute possible number of 8-point DFT's by combining two 4-point DFT's . This process is continued until we get N-point DFT. Following steps were required to implement the DIT-FFT algorithm

STEP1:
From basic definition of N-point DFT of a sequence x(n)

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

Separate even n and odd n by replacing n by 2n and n by 2n + 1

$$= \sum_{n=Even}^{N-1} x(n)\, W_N^{nk} + \sum_{n=Odd}^{N-1} x(n)\, W_N^{nk}$$

$$= \sum_{n=0,2,4..}^{N-2} x(n)\, W_N^{nk} + \sum_{n=1,3,5..}^{N-1} x(n)\, W_N^{nk}$$

$$= \sum_{2n=0,2,4..}^{N-2} x(2n)\, W_N^{2nk} + \sum_{2n+1=1,3,5..}^{N-1} x(2n+1)\, W_N^{(2n+1)k}$$

$$= \sum_{n=0,1,2..}^{\frac{N}{2}-1} x(2n)\, W_N^{2nk} + \sum_{n=1,2,3..}^{\frac{N}{2}-1} x(2n+1)\, W_N^{2nk} W_N^{k}$$

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

$$= \frac{N}{2} - \text{point DFT } [\, x(2n)\,] + W_N^{k}\ \frac{N}{2} - \text{point DFT } [\, x(2n+1)\,]$$

$$X(k) = I_1(k) + W_N^{k}\ I_2(k)$$

Where $I_1(k)$ and $I_2k)$ are periodic with a period of N/2 samples

$$I_1(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)\, W_{N/2}^{nk} \quad \text{Equation} - 1, \quad I_1(N/2 + k) = I_1(k)$$

$$I_2(k) = \sum_{n=0..}^{\frac{N}{2}-1} x(2n+1)\, W_{N/2}^{nk} \quad \text{Equation} - 2, \quad I_2(N/2 + k) = I_2(k)$$

Here we have to compute N samples of X(k)

EX: Take N=8, for this we have to compute 8 samples of X(k), are
{X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7) } over the range $0 \le k \le 7$ and use
$I_1(4+k) = I_1(k)$ $\Rightarrow I_1(4) = I_1(0)$

$\Rightarrow I_1(5) = I_1(1)$

$\Rightarrow I_1(6) = I_1(2)$

$\Rightarrow I_1(7) = I_1(3)$

$I_2(4+k) = I_2(k)$ $\Rightarrow I_2(4) = I_2(0)$

$\Rightarrow I_2(5) = I_2(1)$

$\Rightarrow I_2(6) = I_2(2)$

$\Rightarrow I_2(7) = I_2(3)$

$$X(0) = I_1(0) + W_8^0 \; I_2(0)$$

$$X(1) = I_1(1) + W_8^1 \; I_2(1)$$

$$X(2) = I_1(2) + W_8^2 \; I_2(2)$$

$$X(3) = I_1(3) + W_8^3 \; I_2(3)$$

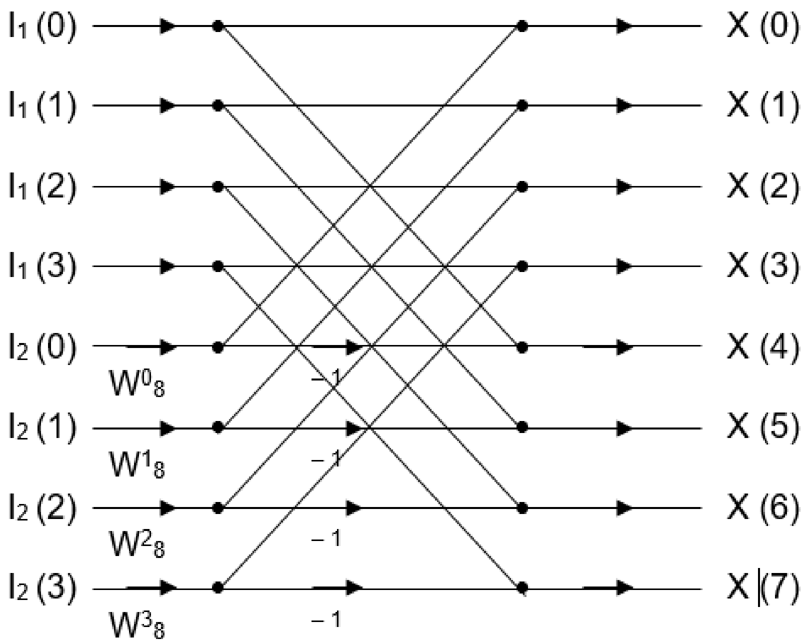
$$X(4) = I_1(4) + W_8^4 \; I_2(4) = I_1(0) - W_8^0 \; I_2(0)$$

$$X(5) = I_1(5) + W_8^5 \; I_2(5) = I_1(1) - W_8^1 \; I_2(1)$$

$$X(6) = I_1(6) + W_8^6 \; I_2(6) = I_1(2) - W_8^2 \; I_2(2)$$

$$X(7) = I_1(7) + W_8^7 \; I_2(7) = I_1(3) - W_8^3 \; I_2(3)$$

Signal flow graph of the stage (Butterfly Structure):



STEP2:

From equation number 1 or N/2-point DFT of a sequence x(2n), i.e

$$I_1(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)\, W_{N/2}^{nk}$$

Replace n by 2n and 2n + 1

$$I_1(k) = \sum_{n=0}^{\frac{N}{4}-1} x(2(2n))\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(2(2n+1))\, W_{N/2}^{(2n+1)k}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+2))\, W_{N/2}^{2nk}\, W_{N/2}^{k}$$

$$I_1(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/4}^{nk} + W_{N/2}^{k} \sum_{n=0..}^{\frac{N}{4}-1} x(4n+2)\, W_{N/4}^{nk}$$

$$= \frac{N}{4} - \text{point DFT } [\, x(4n)\,] + W_{N/2}^{k}\ \frac{N}{4} - \text{point DFT } [\, x(4n+2)\,]$$

$$I_1(k) = I_3(k) + W_{N/2}^{k}\ I_4(k)$$

Where $I_3(k)$ and $I_4(k)$ are periodic with a period of N/4 samples

$$I_3(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/4}^{nk} \qquad \text{Equation} - 3, \quad I_3(N/4+k) = I_3(k)$$

$$I_4(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+2)\, W_{N/4}^{nk} \qquad \text{Equation} - 4, \quad I_4(N/4+k) = I_4(k)$$

Here we have to compute N/2 samples of $I_1(k)$

From equation number 2 or N/2-point DFT of a sequence x(2n+1), i.e

$$I_2(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, W_{N/2}^{nk}$$

Replace n by 2n and 2n + 1

$$I_2(k) = \sum_{n=0}^{\frac{N}{4}-1} x(2(2n)+1)\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(2(2n+1)+1)\, W_{N/2}^{(2n+1)k}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\, W_{N/2}^{2nk}\, W_{N/2}^{k}$$

$$I_2(k)=\sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,W_{N/4}^{nk} + W_{N/2}^{k}\sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,W_{N/4}^{nk}$$

$$=\frac{N}{4}-\text{point DFT }[x(4n+1)] + W_{N/2}^{k}\ \frac{N}{4}-\text{point DFT }[x(4n+3)]$$

$$I_2(k)=I_5(k)+W_{N/2}^{k}\ I_6(k)$$

Where $I_5(k)$ and $I_6(k)$ are periodic with a period of N/4 samples

$$I_5(k)=\sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,W_{N/4}^{nk} \quad \text{Equation }-5,\ \ I_5(N/4+k)=I_5(k)$$

$$I_6(k)=\sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,W_{N/4}^{nk} \quad \text{Equation }-6,\ \ I_6(N/4+k)=I_6(k)$$

Here we have to compute N/2 samples of $I_2(k)$

EX: Take N=8, for this we have to compute 4 samples of $I_1(k)$ and 4 samples of $I_2(k)$ are {$I_1(0)$, $I_1(1)$, $I_1(2)$, $I_1(3)$, $I_2(0)$, $I_2(1)$, $I_2(2)$, $I_2(3)$ } over the range $0 \le k \le 3$ and use

$I_3(2+k) = I_3(k)$ $\Rightarrow I_3(2) = I_3(0)$

$\Rightarrow I_3(3) = I_3(1)$

$I_4(2+k) = I_4(k)$ $\Rightarrow I_4(2) = I_4(0)$

$\Rightarrow I_4(3) = I_4(1)$

$I_5(2+k) = I_5(k)$ $\Rightarrow I_5(2) = I_5(0)$

$\Rightarrow I_5(3) = I_5(1)$

$I_6(2+k) = I_6(k)$ $\Rightarrow I_6(2) = I_6(0)$

$\Rightarrow I_6(3) = I_6(1)$

$$I_1(0)=I_3(0)+W_4^{0}\ I_4(0)$$

$$I_1(1)=I_3(1)+W_4^{1}\ I_4(1)$$

$$I_1(2)=I_3(2)+W_4^{2}\ I_4(2)=I_3(0)-W_4^{0}\ I_4(0)$$

$$I_1(3)=I_3(3)+W_4^{3}\ I_4(3)=I_3(1)-W_4^{1}\ I_4(1)$$

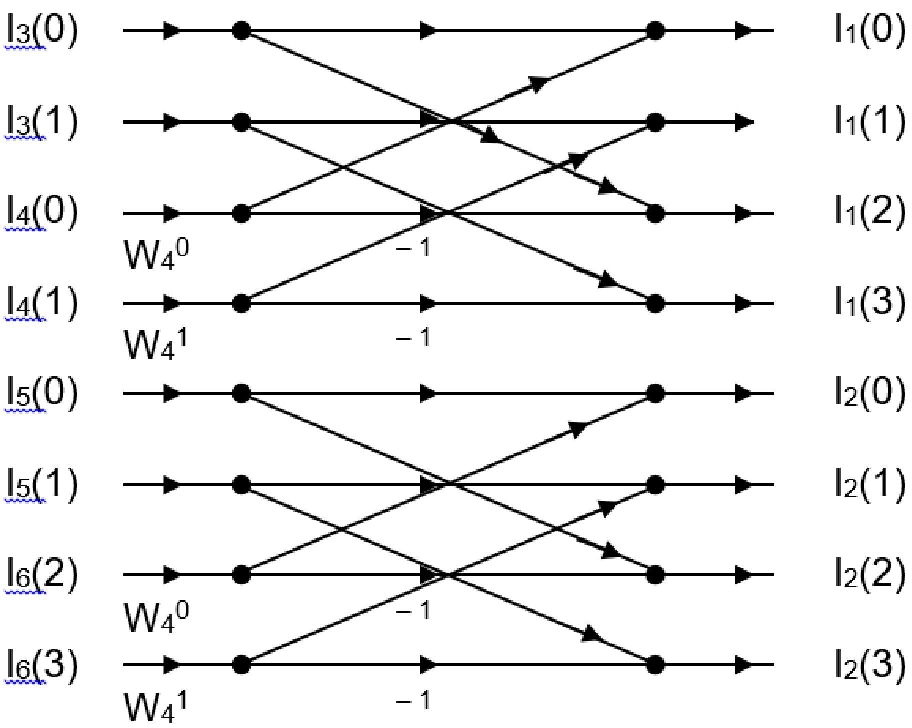
$$I_2(0)=I_5(0)+W_4^{0}\ I_6(0)$$

$$I_2(1)=I_5(1)+W_4^{1}\ I_6(1)$$

$$I_2(2)=I_5(2)+W_4^{2}\ I_6(2)=I_5(0)-W_4^{0}\ I_6(0)$$

$$I_2(3)=I_5(3)+W_4^{3}\ I_6(3)=I_5(1)-W_4^{1}\ I_6(1)$$

Signal flow graph of the stage (Butterfly Structure):



STEP 3:

From equations 3, 4, 5, 6 and take N = 8

$$I_3(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{1} x(4n)\, W_2^{nk} \quad = x(0) + W_2^{k}\, x(4)$$

$$I_3(0) = x(0) + W_2^{0}\, x(4) \text{ and } I_3(1) = x(0) + W_2^{1}\, x(4) = x(0) - W_2^{0}\, x(4)$$

$$I_4(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+2)\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{1} x(4n+2)\, W_2^{nk} \quad = x(2) + W_2^{k}\, x(6)$$

$$I_4(0) = x(2) + W_2^{0}\, x(6) \text{ and } I_4(1) = x(2) + W_2^{1}\, x(6) = x(2) - W_2^{0}\, x(6)$$

$$I_5(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{1} x(4n+1)\, W_2^{nk} \quad = x(1) + W_2^{k}\, x(5)$$

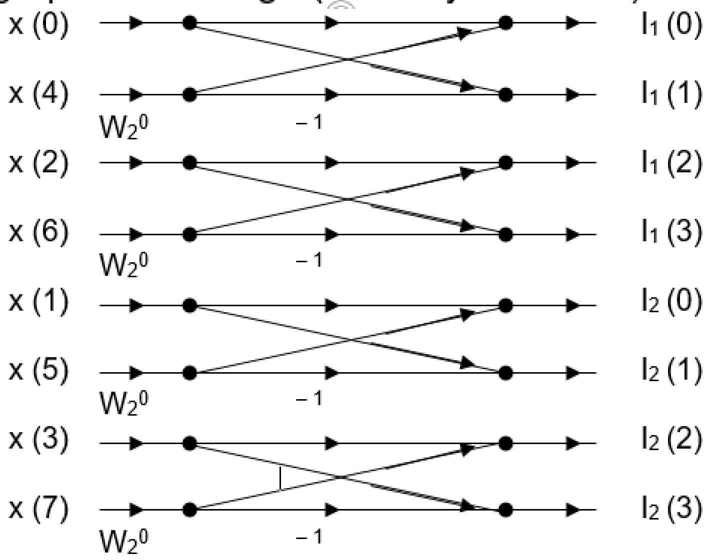
$$I_5(0) = x(1) + W_2^{0}\, x(5) \text{ and } I_5(1) = x(1) + W_2^{1}\, x(5) = x(1) - W_2^{0}\, x(5)$$

$$I_6(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+3)\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{1} x(4n+3)\, W_2^{nk} \quad = x(3) + W_2^{k}\, x(7)$$

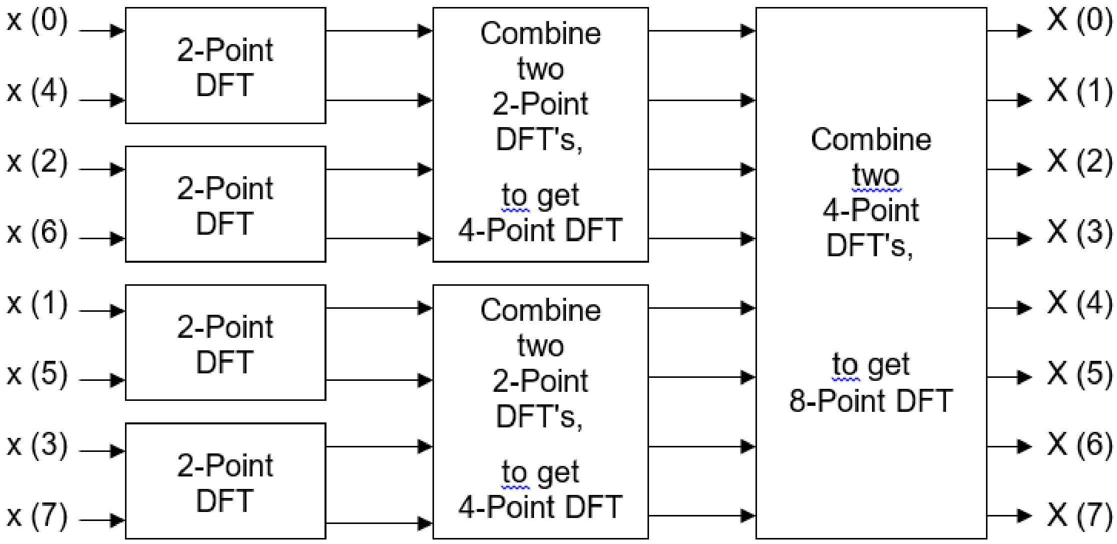
$$I_6(0) = x(3) + W_2^{0}\, x(7) \text{ and } I_6(1) = x(3) + W_2^{1}\, x(7) = x(3) - W_2^{0}\, x(7)$$

Signal flow graph of the stage (Butterfly Structure):



W_2^0 – 1

Three Stage Computational Structure for Radix 2 DIT – FFT:



Normal Order & Bit Reversed Order:

Given sequence x(n) contains samples x(0), x(1), x(2), x(3), x(4), x(5), x(6) and x(7), but here x(n) is decimated into four 2-point DFT's consists of x(0) & x(4), x(2) & x (6), x(1) & x(5), x(3) & x(7). This order is different from actual order is known as bit reversed order, the order of X(k) is known as normal order.

Frequency Domain [X(k)] Normal Order	Time Domain [x(n)] Bit Reversed Order
X(0) = X(000)	x(000) = x(0)
X(1) = X(001)	x(100) = x(4)
X(2) = X(010)	x(010) = x(2)
X(3) = X(011)	x(110) = x(6)
X(4) = X(100)	x(001) = x(1)
X(5) = X(101)	x(101) = x(5)
X(6) = X(110)	x(011) = x(3)
X(7) = X(111)	x(111) = x(7)

	Stage 1	Stage 2	Stage 3	Total
Number of Complex Adders	8	8	8	24
Number of Complex Multipliers	4	4	4	16
Number of Butterflies	4	2	1	7
Different Twiddle Factors	W_2^0	W_4^0, W_4^1	W_8^0, W_8^1, W_8^2, W_8^3	7

DECIMATION IN FREQUENCY (DIF) RADIX – 2 FFT ALGORITHM:

In DIF Radix – 2 FFT, the frequency domain N-point sequence X(k) is decimated into possible number of 2-point sequences. In this algorithm N-point time domain sequence is converted into two N/2 point sequences. Then each N/2 point sequence is converted into two N/4 point sequences. This process is continued until we get required number of 2-point sequences. Following steps were required to implement the DIF-FFT algorithm

STEP1:

From basic definition of N-point DFT of a sequence x(n)

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

Separate the total range of N samples into 0 to N/2 - 1 and N/2 to N - 1.

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

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

$$=\sum_{n=0}^{N/2-1} x(n)\,W_N^{nk}+\sum_{n-N/2=0}^{N-1} x(n)\,W_N^{nk},\ \text{and Let } n-N/2=m$$

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

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

$$=\sum_{n=0}^{N/2-1} x(n)\,W_N^{nk}+\sum_{m=0}^{N/2-1} x(N/2+m)\,W_2^{k}\,W_N^{mk}$$

Replace the dummy variable m by n.

$$=\sum_{n=0}^{N/2-1} x(n)\,W_N^{nk}+\sum_{n=0}^{N/2-1} x(N/2+n)\,W_2^{k}\,W_N^{nk}$$

$$X(k)=\sum_{n=0}^{N/2-1} [\,x(n)+W_2^{k}\,x(N/2+n)\,]\,W_N^{nk}$$

Replace k by 2k for even location

$$X(2k)=\sum_{n=0}^{N/2-1} [\,x(n)+W_2^{2k}\,x(N/2+n)\,]\,W_N^{n2k}$$

$$=\sum_{n=0}^{N/2-1} [\,x(n)+x(N/2+n)\,]\,W_{N/2}^{nk}$$

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

$$=\frac{N}{2}-\text{point DFT of } g_1(n)$$

Where $g_1(n) = x(n) + x(N/2 + n)$

Replace k by 2k +1 for odd location

$$X(2k+1) = \sum_{n=0}^{N/2-1} [\, x(n) + W_2^{2k+1}\, x(N/2+n)\,]\, W_N^{n(2k+1)}$$

$$= \sum_{n=0}^{N/2-1} [\, x(n) + W_2^{2k}\, W_2^{1}\, x(N/2+n)\,]\, W_N^{2nk}\, W_N^{n}$$

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

$$X(2k+1) = \sum_{n=0}^{N/2-1} g_2(n)\, W_{N/2}^{nk} \quad \text{-----------------------------------} \rightarrow \quad (2)$$

$$= \frac{N}{2} - \text{point DFT of } g_2(n)$$

Where $g_2(n) = [\, x(n) - x(N/2 + n)\,]\, W_N^n$

EX: Take N =8

For this example, we have to compute 8 samples from $g_1(n)$ and $g_2(n)$.

$g_1(n) = x(n) + x(4 + n)$ $\Rightarrow g_1(0) = x(0) + x(4)$

$\Rightarrow g_1(1) = x(1) + x(5)$

$\Rightarrow g_1(2) = x(2) + x(6)$

$\Rightarrow g_1(3) = x(1) + x(7)$

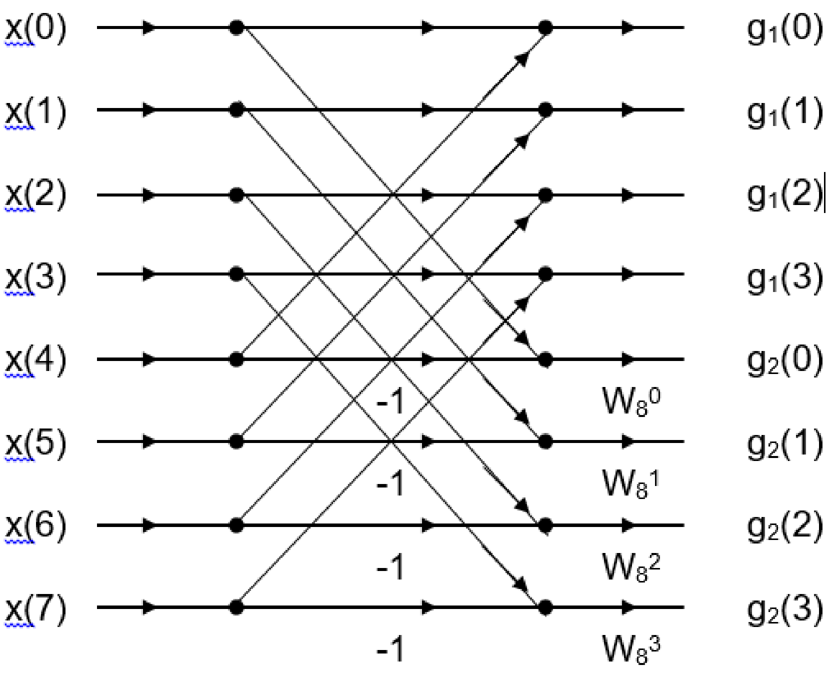
$g_2(n) = [\, x(n) - x(4 + n)\,]\, W_8^n$ $\Rightarrow g_2(0) = [\, x(0) - x(4)\,]\, W_8^0$

$\Rightarrow g_2(1) = [\, x(1) - x(5)\,]\, W_8^1$

$\Rightarrow g_2(2) = [\, x(2) - x(6)\,]\, W_8^2$

$\Rightarrow g_2(3) = [\, x(3) - x(7)\,]\, W_8^3$

Signal Flow Graph of Step 1 (Butterfly Structure)



STEP2: From equation 1

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

Separate the total range of N/2 samples into 0 to N/4 -1and N/4 to N -1.

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{n=N/4}^{N/2-1} g_1(n)\,W_{N/2}^{nk}$$

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{n=N/4}^{N/2-1} g_1(n)\,W_{N/2}^{nk}$$

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{n-N/4=0}^{N/2-1} g_1(n)W_{N/2}^{nk}, \text{ and Let } n-N/4=m$$

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{m=0}^{N/4-1} g_1(N/4+m)\,W_{N/2}^{(N/4+m)k}$$

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{m=0}^{N/4-1} g_1(N/4+m)\,W_{N/2}^{Nk/4}\;W_{N/2}^{mk}$$

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{m=0}^{N/4-1} g_1(N/4+m)\,W_2^{k}\;W_{N/2}^{mk}$$

Replace the dummy variable m by n.

$$=\sum_{n=0}^{N/4-1} g_1(n)\,W_{N/2}^{nk}+\sum_{n=0}^{N/4-1} g_1(N/4+n)\,W_2^{k}\;W_{N/2}^{nk}$$

$$X(2k)=\sum_{n=0}^{N/4-1}[\,g_1(n)+W_2^{k}\;g_1(N/4+n)]\,W_{N/2}^{nk}$$

Replace k by 2k for even location

$$X(2.2k)=\sum_{n=0}^{N/4-1}[\,g_1(n)+W_2^{2k}\;g_1(N/4+n)\,]\,W_{N/2}^{n.2k}$$

$$X(4k)=\sum_{n=0}^{N/4-1}[\,g_1(n)+\;g_1(N/4+n)]\,W_{N/4}^{nk}$$

$$=\sum_{n=0}^{N/4-1} g_3(n)\;W_{N/4}^{nk}\;\text{-----------------------------}\rightarrow(3)$$

$$X(4k)=\frac{N}{4}-\text{point DFT of } g_3(n)$$

Where $g_3(n) = g_1(n) + g_1(N/4 + n)$

Replace k by 2k+1 for odd location

$$X(2(2k+1))=\sum_{n=0}^{N/4-1}[\,g_1(n)+W_2^{2k+1}\;g_1(N/4+n)\,]\,W_{N/2}^{n(2k+1)}$$

$$X(4k+2)=\sum_{n=0}^{N/4-1}[\,g_1(n)+\;W_2^{2k}W_2^{1}g_1(N/4+n)]\,W_{N/2}^{2nk}\,W_{N/2}^{n}$$

$$=\sum_{n=0}^{N/4-1}[\,g_1(n)-g_1(N/4+n)]\,W_{N/2}^{n}\;W_{N/4}^{nk}$$

$$=\sum_{n=0}^{N/4-1} g_4(n)\;W_{N/4}^{nk}\;\text{-----------------------------}\rightarrow(4)$$

$$X(4k)=\frac{N}{4}-\text{point DFT of } g_4(n)$$

Where $g_4(n) = [\, g_1(n) - g_1(N/4 + n)\,]\, W_{N/2}^{\ n}$

From equation 2

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

Separate the total range of N/2 samples into 0 to N/4 -1 and N/4 to N - 1.

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{n=N/4}^{N/2-1} g_2(n)\, W_{N/2}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{n=N/4}^{N/2-1} g_2(n)\, W_{N/2}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{n-N/4=0}^{N/2-1} g_2(n)\, W_{N/2}^{nk}, \text{ and Let } n - N/4 = m$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_2(N/4+m)\, W_{N/2}^{(N/4+m)k}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_2(N/4+m)\, W_{N/2}^{Nk/4}\, W_{N/2}^{mk}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_2(N/4+m)\, W_{2}^{k}\, W_{N/2}^{mk}$$

Replace the dummy variable m by n.

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{n=0}^{N/4-1} g_2(N/4+n)\, W_{2}^{k}\, W_{N/2}^{nk}$$

$$X(2k+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + W_{2}^{k}\, g_2(N/4+n)\,]\, W_{N/2}^{nk}$$

Replace k by 2k for even location

$$X(2.2k+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + W_{2}^{2k}\, g_2(N/4+n)\,]\, W_{N/2}^{n.2k}$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + g_2(N/4+n)\,]\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_5(n)\, W_{N/4}^{nk} \text{ -------------------------------------} \rightarrow (5)$$

$$X(4k+1) = \frac{N}{4} - \text{point DFT of } g_5(n)$$

Where $g_5(n) = g_2(n) + g_2(N/4 + n)$

Replace k by 2k+1 for odd location

$$X(2(2k+1)+1) = \sum_{n=0}^{N/4-1} [g_2(n) + W_2^{2k+1} g_2(N/4+n)] W_{N/2}^{n(2k+1)}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} [g_2(n) + W_2^{2k} W_2^{1} g_2(N/4+n)] W_{N/2}^{2nk} W_{N/2}^{n}$$

$$= \sum_{n=0}^{N/4-1} [g_2(n) - g_2(N/4+n)] W_{N/2}^{n} W_{N/4}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_6(n)\ W_{N/4}^{nk} \text{-------------------------} \rightarrow (6)$$

$$X(4k+3) = \frac{N}{4} - \text{point DFT of } g_6(n)$$

Where $g_6(n) = [\ g_2(n) - g_2(N/4 + n)\ ]\ W_{N/2}^{\ n}$

EX: Take N = 8

Here we have to compute 8 samples from $g_3(n)$, $g_4(n)$, $g_5(n)$ and $g_6(n)$

$g_3(n) = g_1(n) + g_1(2 + n)$ $\Rightarrow g_3(0) = g_1(0) + g_1(2)$

$\Rightarrow g_3(1) = g_1(1) + g_1(3)$

$g_4(n) = [g_1(n) - g_1(2 + n)]W_4^n$ $\Rightarrow g_4(0) = [\ g_1(0) - g_1(2)\ ]W_4^0$

$\Rightarrow g_4(1) = [\ g_1(1) - g_1(3)\ ]W_4^1$

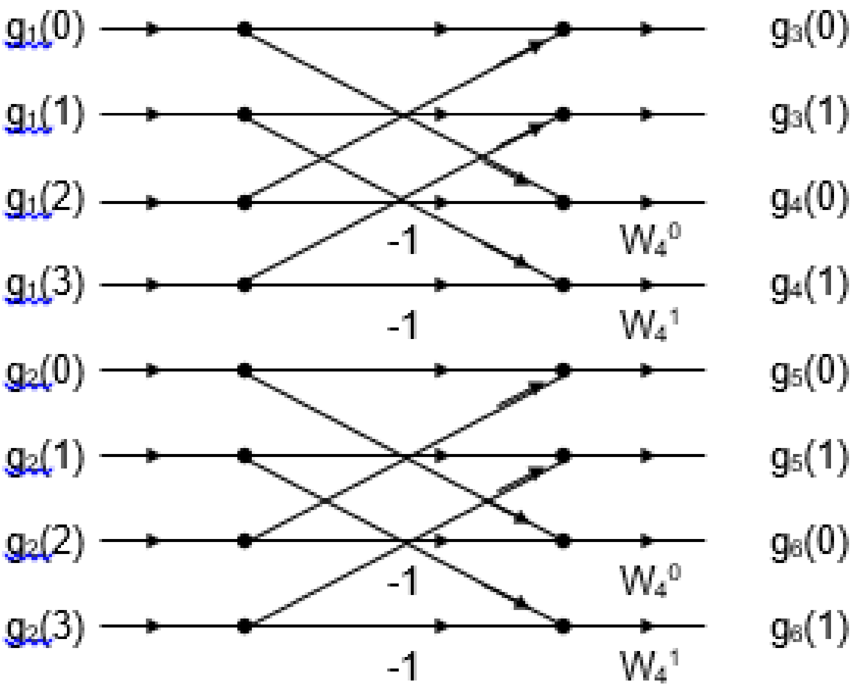
$g_4(n) = g_1(n) + g_1(2 + n)$ $\Rightarrow g_5(0) = g_2(0) + g_2(2)$

$\Rightarrow g_5(1) = g_2(1) + g_2(3)$

$g_5(n) = [g_1(n) - g_1(2 + n)]W_4^n$ $\Rightarrow g_6(0) = [\ g_2(0) - g_2(2)\ ]W_4^0$

$\Rightarrow g_6(1) = [\ g_2(1) - g_2(3)\ ]W_4^1$

Signal Flow graph of Step –2 (Butterfly Structure) :



STEP 3:

This process will continue until we get 2-point DFT, but for this example N = 8 decimation process is not required because of N/4 = 2 called "radix 2". Here we have to compute 8 samples of X(k) from X(4k), X(4k+2), X(4k+1) and X(4k+3).

From equation 3

$$X(4k) = \sum_{n=0}^{1} g_3(n)\, W_2^{nk}$$

$$= g_3(0)\, W_2^{0k} + g_3(1)\, W_2^{1k}$$

$$= g_3(0) + W_2^{k}\, g_3(1)$$

$\Rightarrow X(0) = g_3(0) + W_2^0\, g_3(2) = g_3(0) + g_3(2)$

$\Rightarrow X(4) = g_3(0) + W_2^1\, g_3(2) = [\, g_3(0) - g_3(2)\,]\, W_2^0$

From equation 4

$$X(4k+2) = \sum_{n=0}^{1} g_4(n)\, W_2^{nk}$$

$$= g_4(0)\, W_2^{0k} + g_4(1)\, W_2^{1k} = g_4(0) + W_2^{k}\, g_4(1)$$

$\Rightarrow X(2) = g_4(0) + W_2^0\, g_4(2) = g_4(0) + g_4(2)$

$\Rightarrow X(6) = g_4(0) + W_2^1\, g_4(2) = [\, g_4(0) - g_4(2)\,]\, W_2^0$

From equation 5

$$X(4k+1) = \sum_{n=0}^{1} g_5(n)\, W_2^{nk}$$

$$= g_5(0)\, W_2^{0k} + g_5(1)\, W_2^{1k} = g_5(0) + W_2^{k}\, g_5(1)$$

$\Rightarrow X(1) = g_5(0) + W_2^0\, g_5(2) = g_5(0) + g_5(2)$

$\Rightarrow X(5) = g_5(0) + W_2^1\, g_5(2) = [\, g_5(0) - g_5(2)\,]\, W_2^0$

From equation 6

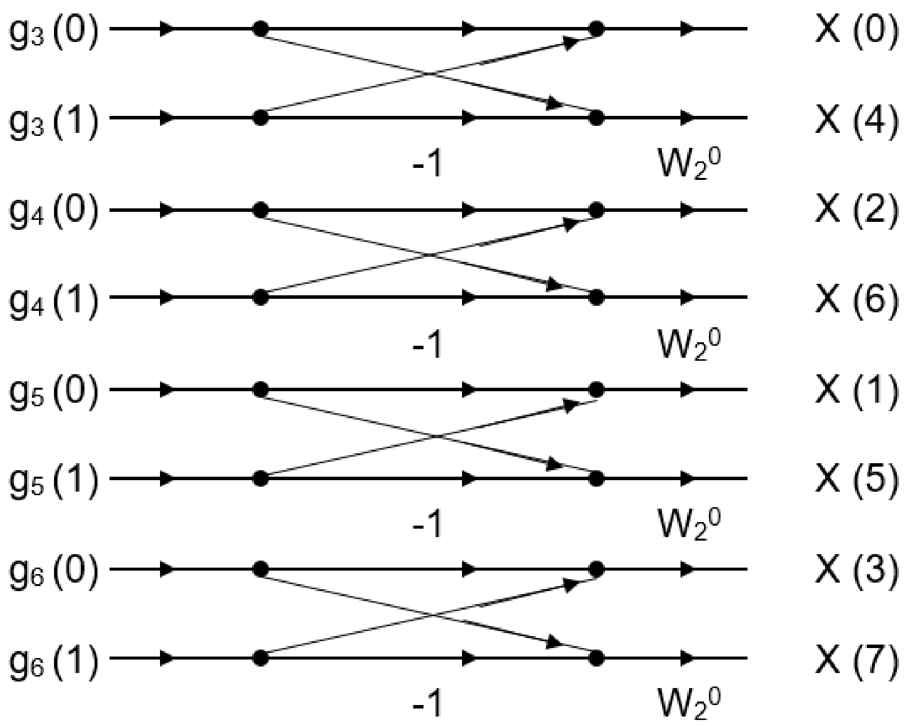
$$X(4k+3) = \sum_{n=0}^{1} g_6(n)\, W_2^{nk}$$

$$= g_6(0)\, W_2^{0k} + g_6(1)\, W_2^{1k} = g_6(0) + W_2^{k}\, g_6(1)$$

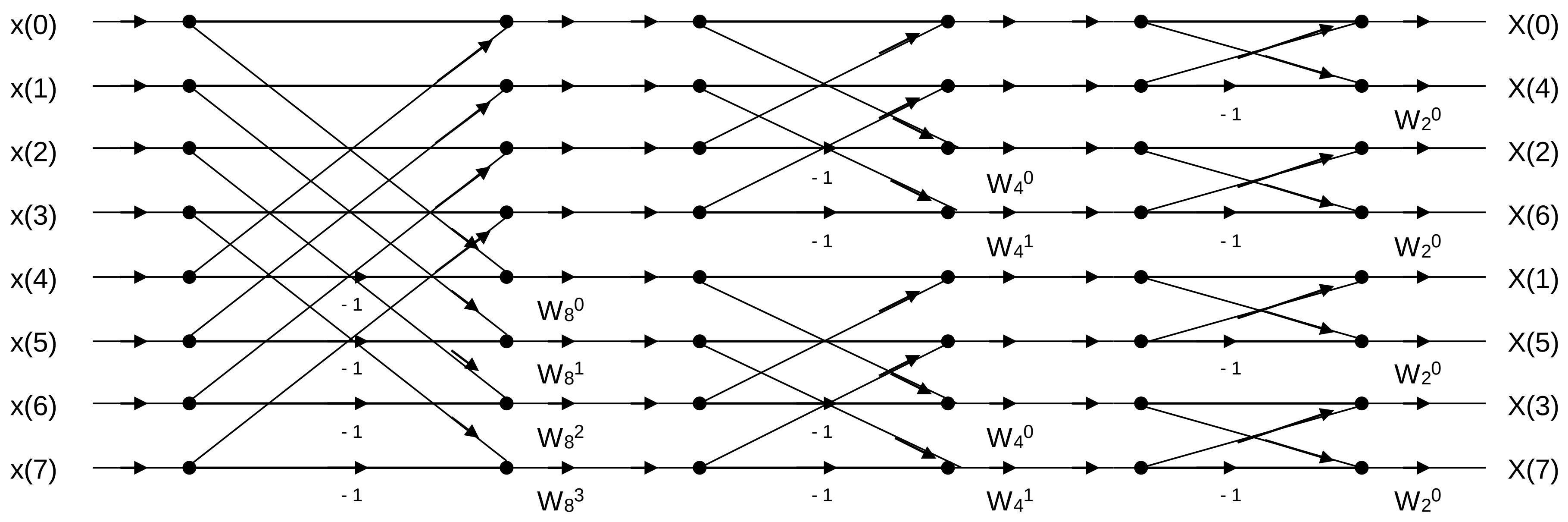
$\Rightarrow X(3) = g_6(0) + W_2^0\, g_6(2) = g_6(0) + g_6(2)$

$\Rightarrow X(7) = g_6(0) + W_2^1\, g_6(2) = [\, g_6(0) - g_6(2)\,]\, W_2^0$

Signal Flow graph of Step –3 (Butterfly Structure)



Butterfly Structure of 8-point Radix-2 DIF-FFT



COMPARISON between DIT AND DIF METHODS:

S.No	Radix 2, DIT-FFT Method	Radix 2, DIF-FFT Method
1.	The time domain sequence x(n) is decimated	The frequency domain sequence X(k) is decimated
2.	The time domain sequence x(n) is in bit reversed order	The time domain sequence x(n) is in normal order order
3.	The frequency domain sequence X(k) is in normal order	The frequency domain sequence X(k) is in bit reversed order
4.	In each stage of computation, the phase factors are multiplied before add and subtract operations	In each stage of computation, the phase factors are multiplied after add and subtract operations
5.	Both the algorithms require same number of operations (Complex additions & multiplications) to compute DFT	
6.	Both the algorithms require bit reversal at some place during computation	
7.	For both the algorithms, the value of N should be expressed as $N = 2^m$. where m is no. of stages	
8.	For both the algorithms, required number of Complex additions = N Log_2 N. Complex multiplications = N/2. Log_2 N.	

INVERSE FFT:

Computation of IDFT of a sequence X(k) = x(n), through FFT known as inverse FFT. From basic definition of IDFT of a sequence

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

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

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

$$= \frac{1}{N}\left(\ \mathrm{DFT}\,(X^*(k))\ \right)^*$$

$$= \frac{y(n)}{N}$$

Where y(n) = DFT[X*(k)]

PROCEDURE:

1. Take complex conjugate of given X(k), to get X*(k).
2. Compute N-point DFT of X*(k) using radix-2 DIT / DIF FFT method.
3. Take complex conjugate of above sequence to get y(n).
4. Finally obtain x(n) using x(n) = y(n) / N.

USES OF FFT:

1. Single FFT algorithm is used to compute both DFT & IDFT.
2. It is used to obtain the response of discrete LTI system.