EE 512

DIGITAL SIGNAL PROCESSING

Session 15

October 20, 1992

M. Azimi 800-525-4950, ext. 7956

Colorado State University

Computation of the Discrete Fourier Transform

Consider the DFT of a finite duration sequence

$\{x(n)\}$

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nK} \qquad , W_N = e^{-j\frac{2\pi}{N}}$$

$K = 0, \cdots N-1$

The IDFT is defined by

$$x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(K) W_N^{-Kn} \qquad n = 0, \cdots N-1$$

The two expressions differ only in the sign of exponent W_N and in scale factor $1/N$, thus the computation of both would be the same. $x(n)$ and $X(K)$ can both be complex in general.

1) The direct evaluation of N-point DFT requires N^2 complex multiplication (for complex $x(n)$) and

$4N^2$ real multiplications

$N(N-1)$ complex additions
$N(4N-2)$ real additions

2) Moreover the implementation of DFT on a general purpose digital computer or special-purpose hardware requires storing $x(n)$ and all

the basis W_N^{Kn}.

Thus for large N the direct evaluation requires tremendous amount of computation and storage.

Now if N is even by dividing and breaking the sequence into 2 $N/2$- point sequences, it would require $(N/2)^2 \times 2 = N^2/2$ i.e a saving by factor two. This process can be continued (for $N = 2^n$) until it reduces to 2-point DFT

Most approaches to improving the efficiency of the computation of DFT exploit the following properties of W_N^{Kn}

1. $(W_N^{Kn})^* = W_N^{K(N-n)}$

2. $W_N^{Kn} = W_N^{K(n+N)} = W_N^{(K+N)n}$

Decimation-in-Time FFT Algorithms

Algorithm is based on decomposition of sequence $x(n)$ into successively smaller subsequences

Assume $N = 2^p$

Now define two $N/2$-point sequences $x_1(n)$ and $x_2(n)$ as the even and odd members of $x(n)$

$$x_1(n) = x(2n) \qquad n = 0, 1, \ldots N/2 - 1$$
$$x_2(n) = x(2n+1) \qquad n = 0, 1, \ldots N/2 - 1$$

$$x(n) = \{x_1(n)\} \cup \{x_2(n)\}$$

Then the N-point DFT of $\{x(n)\}$ is

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

Replace $n = 2n$ and $n = 2n+1$

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

($x(2n) = x_1(n)$, $x(2n+1) = x_2(n)$)

Note that $W_N^2 = \left(e^{-j\frac{2\pi}{N}}\right)^2 = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$

Thus

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

For $N/2 \le k \le N-1$ let $k = k' + N/2$ then $0 \le k' \le N/2 - 1$

Thus

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

$$= \sum_{n=0}^{N/2-1} x_1(n)\, W_{N/2}^{nk'} + W_N^{(k'+N/2} \sum_{n=0}^{N/2-1} x_2(n)\, W_N^{2n(k'+N/2)}$$

$$= \sum_{n=0}^{N/2-1} x_1(n)\, W_{N/2}^{nk'} - W_N^{k'} \sum_{n=0}^{N/2-1} x_2(n)\, W_{N/2}^{nk'}$$

or

$$X(k) = X_1(k - N/2) - W_N^{(k-N/2)}\, X_2(k - N/2)$$

$$= X_1(k - N/2) + W_N^k\, X_2(k - N/2)$$

$$X(K) = X_1(K) + W_N^K X_2(K)$$

$$X_1(K) = DFT\{x_1(n)\}_{N/2} \quad , \quad X_2(K) = DFT\{x_2(n)\}_{N/2}$$

$K \in [0, \frac{N}{2}-1]$

For $0 \le K \le N-1$ we have

Twiddle factor

$$X(K) = \begin{cases} X_1(K) + W_N^K X_2(K) & 0 \le K \le N/2 - 1 \\ X_1(K-N/2) - W_N^{K-N/2} X_2(K-N/2) & N/2 \le K \le N-1 \end{cases}$$

proof ←

$$X_1(K-N/2) + W_N^K X_2(K-N/2)$$

Because $W_N^{K+N/2} = -W_N^K$

As a result, the sequence $x(n)$ is shuffled into even and odd membus to give $x_1(n)$ and $x_2(n)$ which are then Transformed to give $X_1(K)$ and $X_2(K)$

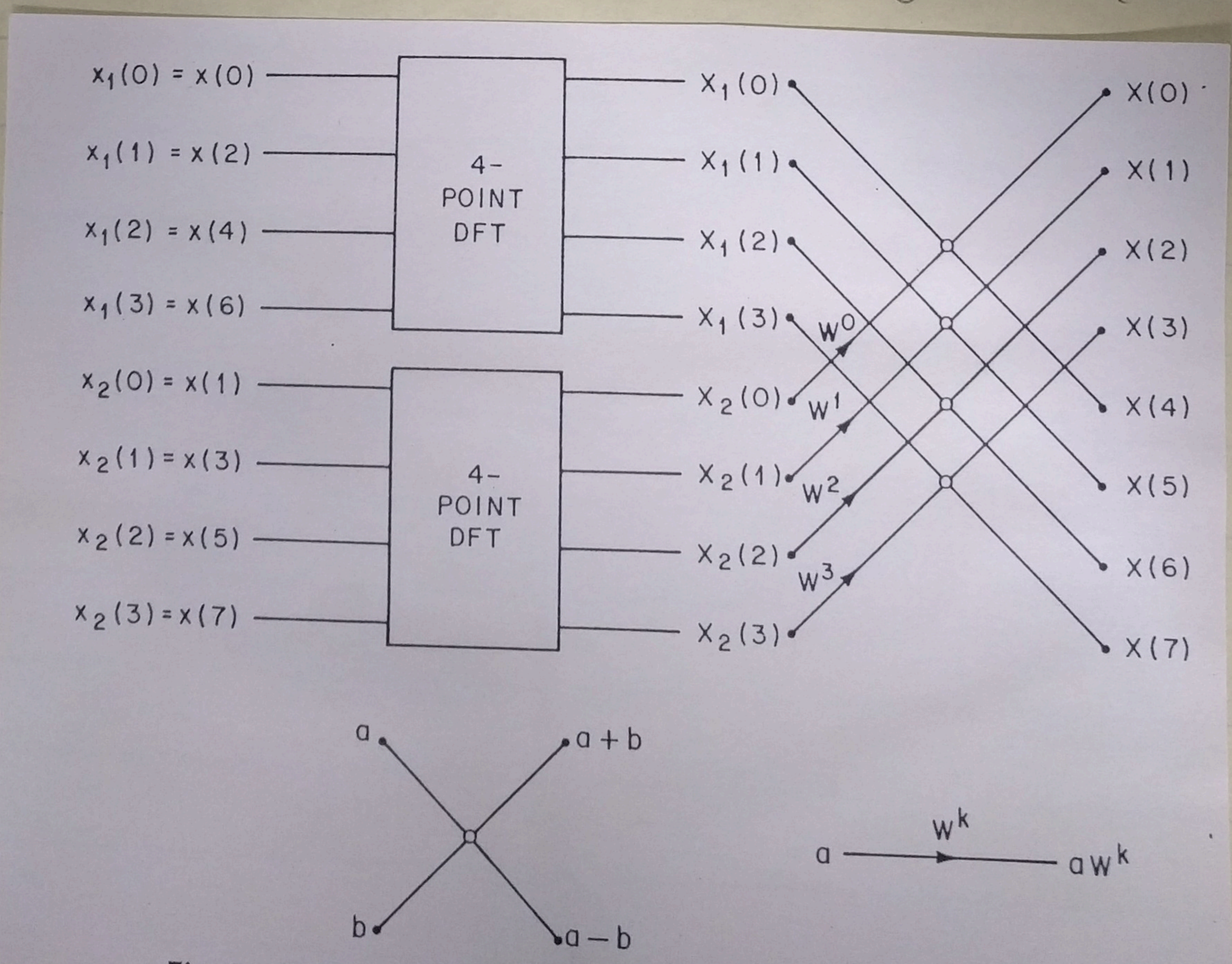


Fig. 6.1 Construction of an eight-point DFT from two four-point DFT's.

Now each $X_1(k)$ and $X_2(k)$ requie $\frac{N^2}{2}$ Complex multiplications using direct method. Moreover we need N Complex multi as a result of multiplying W_N^k by $X_2(k)$. Thus the total is $N^2/_2 + N$ i.e Saving of $\simeq$ % 50 in computational time.

If this process is continued $X_1(k)$ and $X_2(k)$ can be expressed in terms of $N/4$ - point DFT by repeating the same procedure.

$$X_1(k) = \sum_{r=0}^{N/2-1} x_1(r) W_{N/2}^{rk}$$

$$= \sum_{l=0}^{N/4-1} x_1(2l) W_{N/2}^{2lk} + \sum_{l=0}^{N/4-1} x_1(2l+1) W_{N/2}^{(2l+1)k}$$

$$= A(k) + W_{N/2}^{k} B(k)$$

Similarly

$$X_2(k) = C(k) + W_{N/2}^{k} D(k)$$

where

$$C(k) = \sum_{l=0}^{N/4-1} x_2(2l) W_{N/4}^{lk}$$

and

$$D(k) = \sum_{l=0}^{N/4-1} x_2(2l+1) W_{N/4}^{lk}$$

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

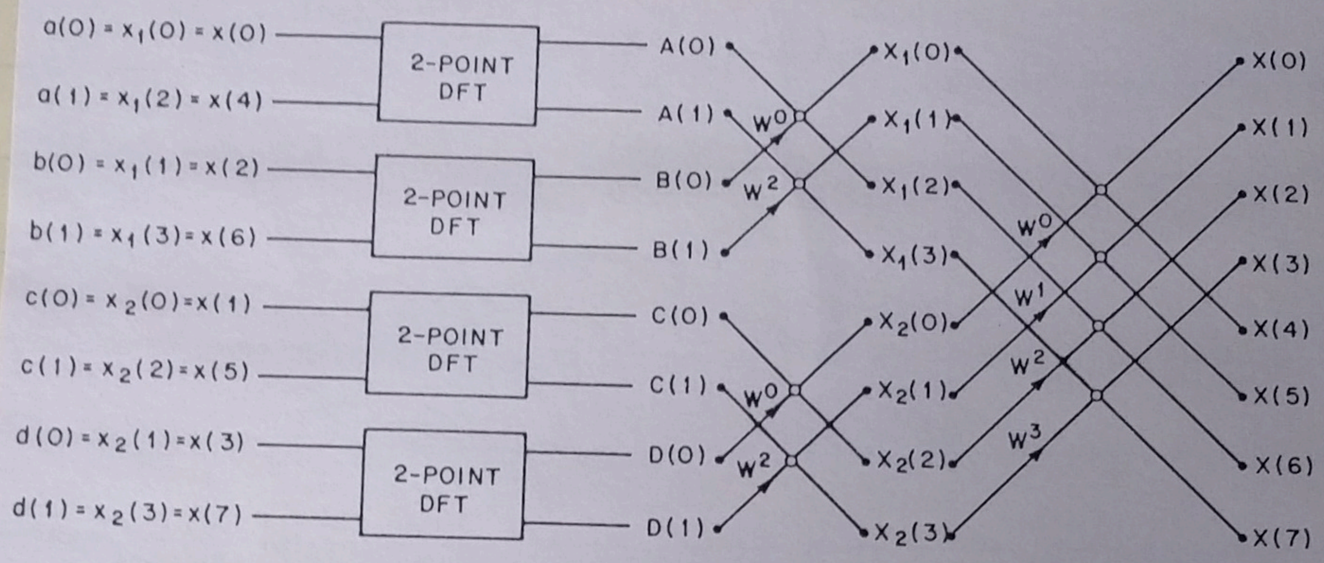


Fig. 6.2 Construction of an eight-point DFT from two four-point DFT's that are in turn constructed from four two-point DFT's.

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

This process continues until we left with two-point DFT's. A Two point DFT can be evaluated with no multiplication at all and only two additions

Computational Count for FFT

At each stage of the FFT there are $N/2$ complex multiplications are required to combine the results of the previous stage. Since there are $\log_2 N$ stages, the Total number of multis (complex) is approximately

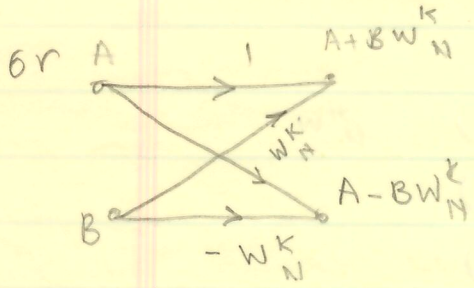
$$N_{FFT} = N/2 \log_2 N$$

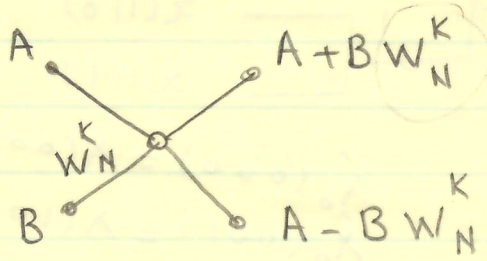
It is approximate because W_N^0, $W_N^{N/2}$, $W_N^{N/4}$ and $W_N^{3N/4}$ are really just complex additions and substractions.

Excluding these would result in

$$N_{FFT} = N/2 \log_2 N - N + 1$$

The basic operation in decimation-in-time algorithm is the so-called butterfly



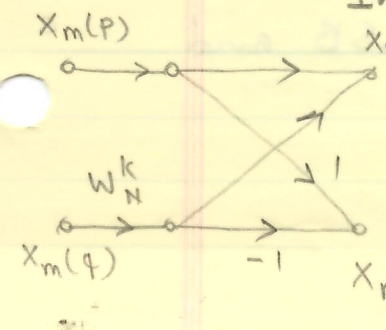


twiddle factor (rotation factor)

At each stage, there are $(N/2)$ butterflies, there is only one complex multiplication per butterfly (BW_N^K) can be computed and saved.

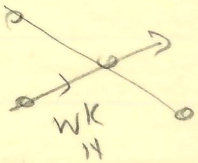
Remarks

1) The result of intermediate stages of FFT can be stored in the same memory location in which the original data is stored. Therefore the array used to store the input is also the same array into which the output sequence will be stored. The algorithm which does that is called "In-place algorithm) i.e if



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

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



$X_{m+1}(p)$ and $X_{m+1}(q)$ are stored in the same storage registers as $X_m(p)$ and $X_m(q)$, respectively

2) In order to have an output in natural order in decimation-in-time algorithm the input should be shuffled in "bit-Reversed" order

If $(n_2 n_1 n_0)$ is the binary representation of the index of sequence $x(n)$, then the sequence $x(n_2 n_1 n_0)$ is stored in the array position $X_0(n_0 n_1 n_2)$. That is in determining the position of $x(n_2 n_1 n_0)$ in the input array, we must reverse the order of bits.

3)

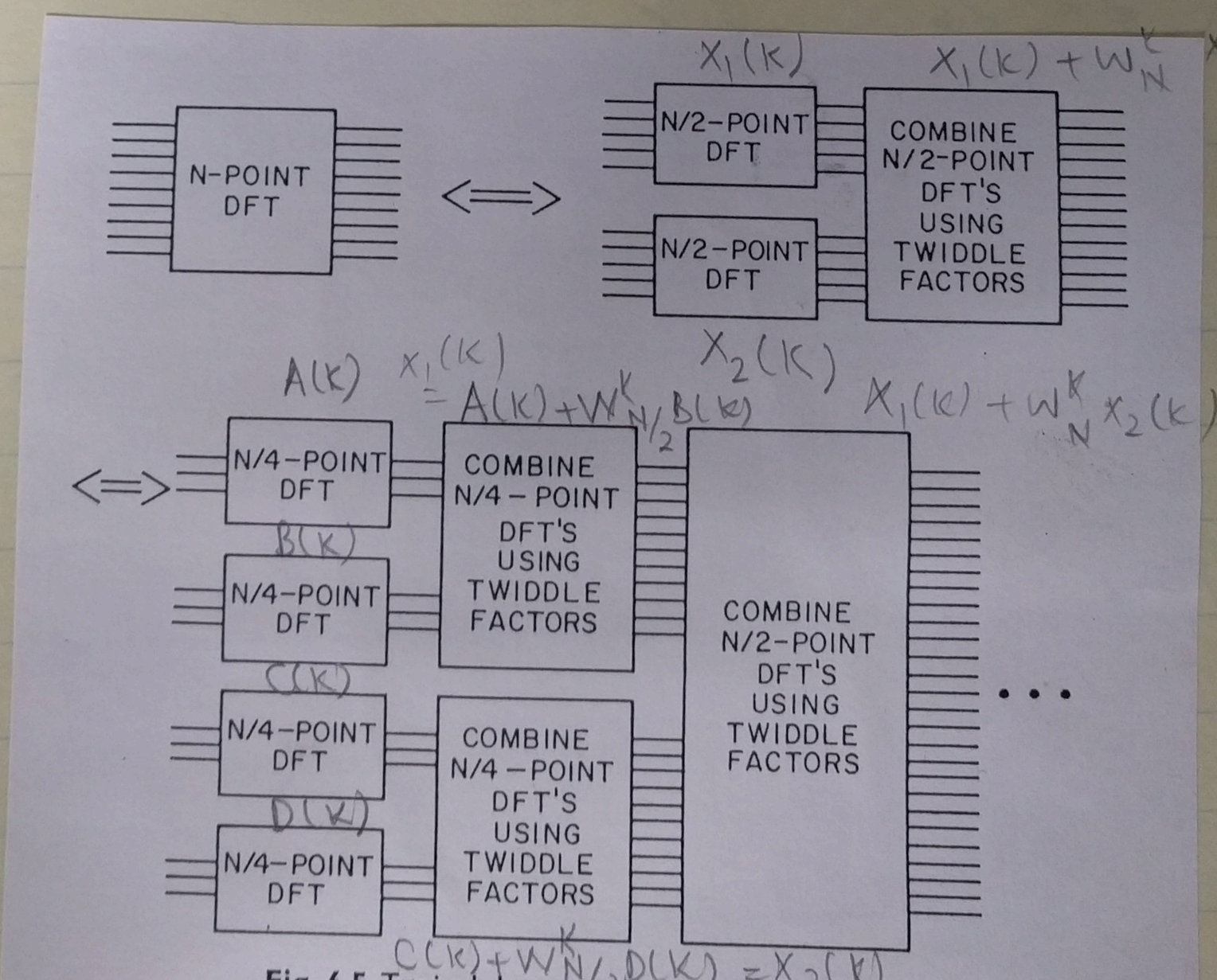
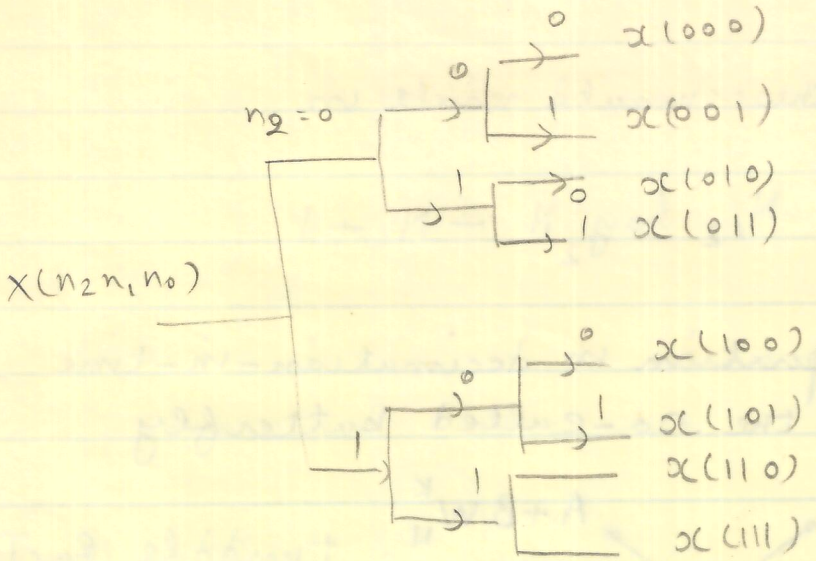


Fig. 6.5 Typical decomposition for radix 2 FFT's.

As shown in this figure all multiplications are due to combining operations and multiplying by twid factors



$$\hat{X}_0(000) = X(000)$$
$$\hat{X}_0(001) = X(100)$$
$$\hat{X}_0(010) = X(010)$$
$$\hat{X}_0(011) = X(110)$$
$$X_0(100) = X(001)$$
$$X_0(101) = X(101)$$
$$X_0(110) = X(011)$$
$$X_0(111) = X(111)$$

odd and even

This separation can be carried out by examining the LSB (n_0) in the index n.

If LSB = 0 $\Rightarrow$ even no. samples and thus appears in the Top half.

LSB = 1 $\Rightarrow$ odd No. samples and appears in the bottom half.

The same process can be repeated for the even and odd subsequences by examining the 2nd LSB and so forth.

Arrange the butterfly diagram for an 4-point FFT.

The N-point DFT of $\{x(n)\}$ is defined by

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nK} = \underbrace{\sum_{n=0}^{N-1} x(n) W_N^{nK}}_{n \text{ even}} + \underbrace{\sum_{n=0}^{N-1} x(n) W_N^{nK}}_{n \text{ odd}}$$

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

where $W_N \triangleq e^{\frac{-2\pi j}{N}}$, $j = \sqrt{-1}$

On the other hand W_N^2 can be written as

$$W_N^2 = \left(e^{\frac{-2\pi j}{N}}\right)^2 = e^{-2\pi j / N/2} = W_{N/2}$$

Therefore

$$X(K) = \sum_{n=0}^{N/2-1} x_1(n) W_{N/2}^{nK} + W_N^K \sum_{n=0}^{N/2-1} x_2(n) W_{N/2}^{nK}$$

$$\begin{cases} X(K) = X_1(K) + W_N^K X_2(K) & 0 \le K \le 1 \quad (K = 0,1) \\ X(K) = X_1(K - N/2) + W_N^K X_2(K - N/2) & K = 2, 3 \end{cases}$$

$$X_1(K) = \sum_{n=0}^{1} x(2n) W_2^{nK} = x(0) + x(2) W_2^K$$

$$X_2(K) = \sum_{n=0}^{1} x(2n+1) W_2^{nK} = x(1) + x(3) W_2^K$$

Then

$$X(0) = X_1(0) + X_2(0)$$

$$X(1) = X_1(1) + X_2(1) W_4^1$$

$$X(2) = X_1(0) + W_4^2 X_2(0) = X_1(0) - X_2(0)$$

$$X(3) = X_1(1) - X_2(1)$$

$$X_1(0) = x(0) + x(2)$$

$$X_2(0) = x(1) + x(3)$$

$$X_1(1) = x(0) + W_2^1 x(2) = x(0) - x(2)$$

$$X_2(1) = x(1) + W_2^1 x(3) = x(1) - x(3)$$

Where

$$x_1(n) = x(2n) \qquad n = 0, 1, \ldots, N/2 - 1$$

and

$$x_2(n) = x(2n+1) \qquad n = 0, 1, \ldots, N/2 - 1$$

Thus we get

$$\boxed{X(K) = X_1(K) + W_N^K \, X_2(K)}$$

where $X_1(K)$ and $X_2(K)$ are seen to be the $N/2$-point DFT's of $x_1(n)$ and $x_2(n)$.

Now for $N = 4$, we have

$$X(K) = \underbrace{[x_0 + x_2 W_4^{2K}]}_{X_1(K)} + W_4^K \underbrace{[x_1 + x_3 W_4^{2K}]}_{X_2(K)}$$

But $W_4^2 = \exp(-j\pi) = -1$ and $W_4^4 = 1$

Therefore:

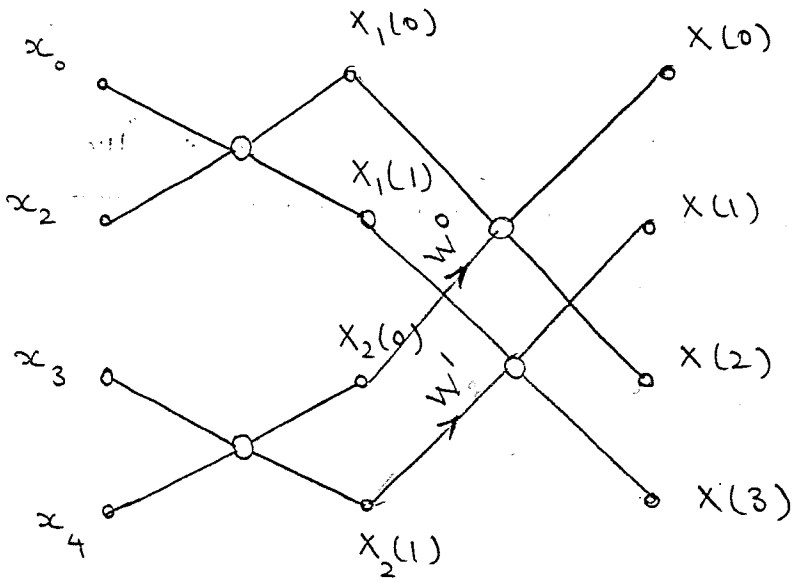
$$X(0) = \underbrace{(x_0 + x_2)}_{X_1(0)} + \underbrace{(x_1 + x_3)}_{X_2(0)} = X_1(0) + X_2(0)$$

$$X(1) = (x_0 - x_2) + W_4(x_1 - x_3) = X_1(1) + W_4 X_2(1)$$

$$X(2) = (x_0 + x_2) - (x_1 + x_3) = X_1(0) - X_2(0)$$

$$X(3) = (x_0 - x_2) - W_4(x_1 - x_3) = X_1(1) - W_4 X_2(1)$$

Using the above equations the flow graph for an 4-point DFT is arranged in the following Fig.



Construction of 4-point DFT from two 2-point DFT's.

No. of complex multiplications = 1

For generation of twiddle factors use recursive equation

$$W_N^K = (W_N^{K-L})\, W_N^L$$

with initial condition $W_N^0 = 1$ to start the algorithm

Example

Arrange the butterfly diagram for an 8-point FFT and count the number of true complex multiplications

$$X(k) = \sum_{n=0}^{7} x(n) W_8^{nk}$$

$$= x(0) + x(1) W_8^{K} + x(2) W_8^{2K} + x(3) W_8^{3K}$$

$$+ - - - + x(7) W_8^{7K}$$

Arrange in bit reversed order

$$= [x(0) + x(2) W_8^{2K} + x(4) W_8^{4K} + x(6) W_8^{6K}]$$

$$+ [x(1) W_8^{K} + x(3) W_8^{3K} + x(5) W_8^{5K} + x(7) W_8^{7K}]$$

$-10-$ DFT$\{x_1(n)\}$

$$X(K) = \underbrace{\left[x(0) + x(2)W_4^K + x(4)W_4^{2K} + x(6)W_4^{3K}\right]}$$

$$+ W_8^K \underbrace{\left[x(1) + x(3)W_4^K + x(5)W_4^{2K} + x(7)W_4^{3K}\right]}$$

DFT$\{x_2(n)\}$

Moreover

$$X(K) = x(0) + x(4)W_2^K + W_4^K\left[x(2) + x(6)W_2^K\right]$$

$$W_8^K\left[\left(x(1) + x(5)\underbrace{W_2^K}_{1}\right) + W_4^K\left(x(3) + x(7)\underbrace{W_2^K}_{1}\right)\right]$$

Thus

$$X(0) = \underbrace{\left[x(0) + x(4)\right]}_{A(0)} + \underbrace{\left[x(2) + x(6)\right]}_{B(0)}$$
$$+ \underbrace{\left[x(1) + x(5)\right]}_{C(0)} + \underbrace{\left[x(3) + x(7)\right]}_{D(0)}$$

$$X(1) = \underbrace{\left[x(0) - x(4)\right]}_{A(1)} + W_4^{W_8^2}\underbrace{\left[x(2) - x(6)\right]}_{B(1)}$$
$$+ W_8^{W'}\left\{\underbrace{\left[x(1) - x(5)\right]}_{C(1)} + \underbrace{\left[x(3) - x(7)\right]}_{D(1)}\right.$$

Chapter 6: Computation of DFT

Consider $x(n) = 0 \quad n<0,\ n>N-1$

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

$\forall k \in [0, N-1]$.

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

observations

1- Some computations

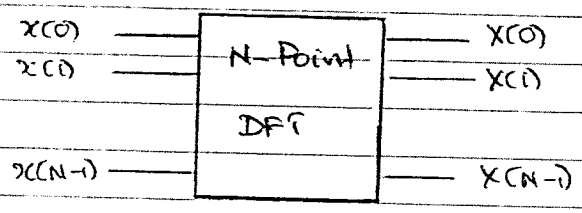
2- Direct evaluation of DFT requires

# of multiplication $= N^2$ (complex multiplication) $4N^2$ real multi

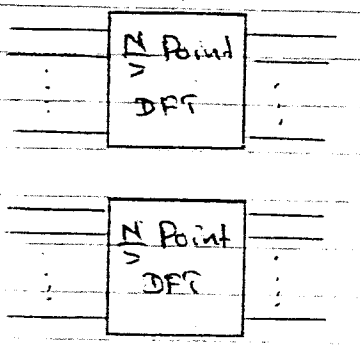
# of additions $= N(N-1)$ (complex additions)

3- storage for $x(n)$ and basis W_N^{kn}.

This is a very complex calculation involving large cpu usage. The objective of fast Fourier Transform (FFT) is to partition the points so that it save more than 50% computation time.



N^2 multiplication



$2 \cdot \frac{N^2}{4} = \frac{N^2}{2}$ multiplications

Also utilize the properties

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

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

Decimation-in-time Fast Fourier Transform (FFT)

Based upon the decomposition of sequence $x(n)$ into successively smaller subsequences.

Assume # of points $= 2^p$

$$x(n) = 0 \qquad 0 > n,\ n \geq N-1$$

decompose $x(n)$ into subsequences $x_1(n)$ (even samples), $x_2(n)$ (odd samples)

$$\left.\begin{aligned} x_1(n) &= x(2n) \quad n = 0, 1, \ldots, \tfrac{N}{2}-1 \\ x_2(n) &= x(2n+1) \quad n = 0, 1, \ldots, \tfrac{N}{2}-1 \end{aligned}\right\} \{x(n)\} = \{x_1(n)\} \cup \{x_2(n)\}$$

$$X(k) = DFT\{x(n)\}$$

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

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

Note that

$$W_N^{2nk} = e^{-\frac{2\pi j (2n)k}{N}} = e^{-\frac{j 2\pi nk}{(\frac{N}{2})}}$$

$$= W_{\frac{N}{2}}^{nk}$$

$$W_N^{(2n+1)k} = W_N^{k} \, W_{\frac{N}{2}}^{nk}$$

$$\therefore X(k) = X_1(k) + W_N^{k} \, X_2(k)$$

$$\therefore X_1(k) = DFT_{\frac{N}{2}} \{ x_1(n) \}$$

$$X_2(k) = DFT_{\frac{N}{2}} \{ x_2(n) \}.$$

ie saving of 50% computations. (possible only if n is even).

** note k in $\underline{X(k)}$ & $\underline{X_1(k)}$ & $\underline{X_2(k)}$ have different ranges.

$$X(k) = \begin{cases} X_1(k) + W_N^{k} \, X_2(k) & 0 \le k \le \frac{N}{2}-1 \\ X_1(k-\frac{N}{2}) + W_N^{k-\frac{N}{2}} \, X_2(k-\frac{N}{2}) \text{ or } W_N^{k} & \frac{N}{2} \le k \le N-1 \end{cases}$$

how ?

$$W_N^{k-\frac{N}{2}} = W_N^{k} \, W_N^{-\frac{N}{2}} = -W_N^{k}$$

$$\therefore X(k) = \begin{cases} X_1(k) + W_N^{k} \, X_2(k) & \text{for } 0 \le k \le \frac{N}{2}-1 \\ X_1(k-\frac{N}{2}) - W_N^{k} \, X_2(k-\frac{N}{2}) & \text{for } \frac{N}{2} \le k \le N-1 \end{cases}$$

? +

W_N^{k}, is called Twiddle Factors.

FFT. contd...

$$X(k) = \begin{cases} X_1(k) + W_N^k X_2(k) & 0 \le k \le \frac{N}{2}-1 \\ X_1(k-\frac{N}{2}) + W_N^k X_2(k-\frac{N}{2}) & \frac{N}{2} \le k \le N-1 \end{cases}$$

where

$$X_1(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)\, W_{\frac{N}{2}}^{nk}$$

$$X_2(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, W_{\frac{N}{2}}^{nk}$$

Continue the process for $X_1(k)$ and $X_2(k)$

$$X_1(k) = \sum_{l=0}^{\frac{N}{4}-1} x_1(2l)\, W_{\frac{N}{2}}^{2lk} + \sum_{l=0}^{\frac{N}{4}-1} x_1(2l+1)\, W_{\frac{N}{2}}^{(2l+1)k}$$

$$X_1(k) = A(k) + W_{\frac{N}{2}}^{k} B(k)$$

where $$A(k) = \sum_{l=0}^{\frac{N}{4}-1} x_1(2l)\, W_{\frac{N}{4}}^{lk} = DFT_{N/4}\{x_1(2l)\}$$

$$B(k) = \sum_{l=0}^{\frac{N}{4}-1} x_1(2l+1)\, W_{\frac{N}{4}}^{lk} = DFT_{N/4}\{x_1(2l+1)\}$$

Similarly for $X_2(k)$

$$X_2(k) = C(k) + W_{\frac{N}{2}}^{k} D(k)$$

$$C(k) = \sum_{l=0}^{\frac{N}{4}-1} x_2(2l)\, W_{\frac{N}{4}}^{lk}$$

$$D(k) = \sum_{l=0}^{\frac{N}{4}-1} x_2(2l+1)\, W_{\frac{N}{4}}^{lk}$$

for 2-point DFT

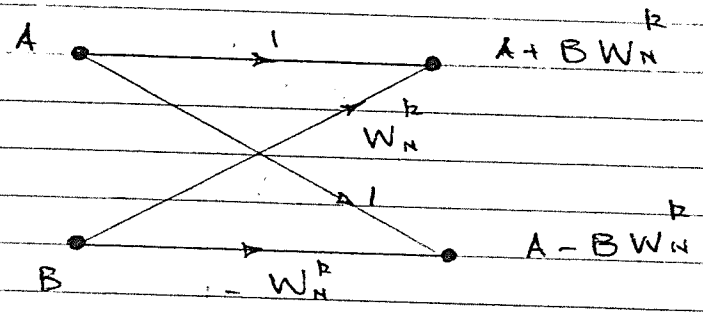
$$X_m(k) = \sum_{n=0}^{1} x_m(n)\, W_2^{nk}$$

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

$$X_m(0) = x_m(0) + x_m(1)$$

$$X_m(1) = x_m(0) - x_m(1).$$

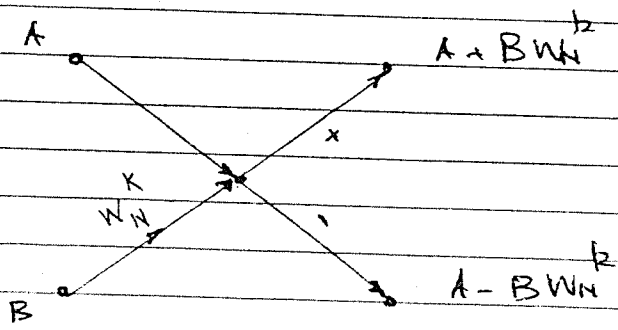
Hence, we continue the decomposition until it is 2-point DFT which can be obtain by addition instead of multiplication.



Butterfly

W_N^k : Twiddle factor.

We can reduce the butterfly as follows



Each butterfly
= 1 Complex multi
+ 2 Complex Addition
(except when k=0)

Example:

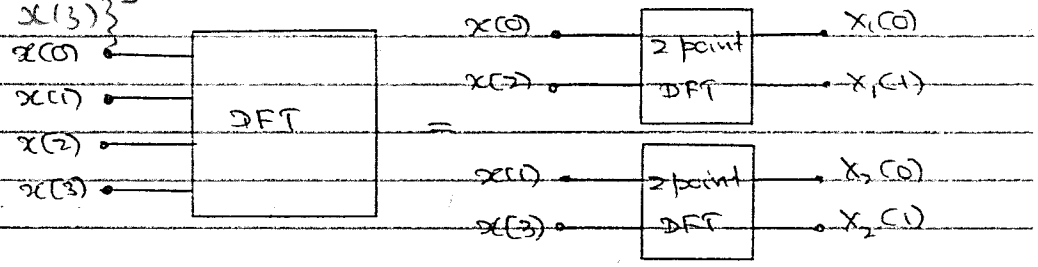
Arrange butterfly structure for a 4-point FFT.

$$X(k) = X_1(k) + W_4^k X_2(k) \qquad k = 0, 1$$

$$X(k) = X_1(k-2) + W_4^k X_2(k-2) \qquad k = 2, 3$$

$x_1(n) = \{x(0), x(2)\}$

$x_2(n) = \{x(1), x(3)\}$



where

$$X_1(k) = \sum_{n=0}^{1} x(2n)\, W_2^{nk}$$

$$= x(0) + x(2)\, W_2^k$$

$$X_2(k) = \sum_{n=0}^{1} x(2n+1)\, W_2^{nk}$$

$$= x(1) + x(3)\, W_2^k$$

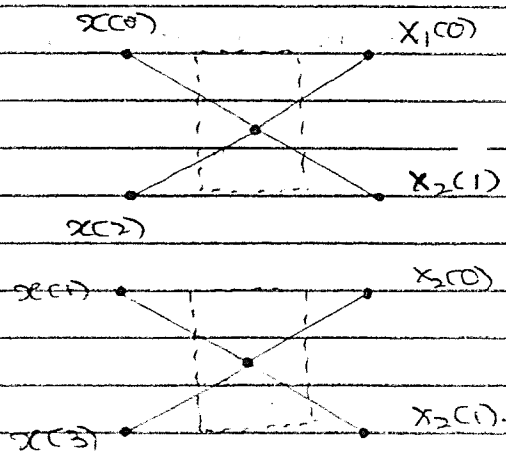
$X_1(0) = x(0) + x(2)$

$X_1(1) = x(0) + x(2)\, W_2^1$

$= x(0) - x(2)$.

$X_2(0) = x(1) + x(3)$

$X_2(1) = x(1) - x(3)$

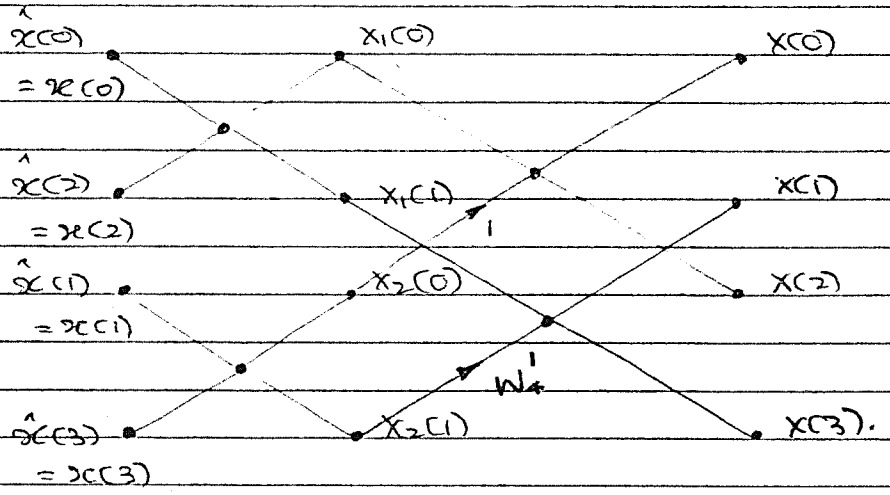


$$X(0) = X_1(0) + X_2(0)$$

$$X(1) = X_1(1) + W_4^1 X_2(1)$$

$$X(2) = X_1(0) + W_4^2 X_2(0) = X_1(0) - X_2(0).$$

$$X(3) = X_1(1) - W_4^1 X_2(1)$$



∴ no of complex multiplication = 1

no of complex additions = 8 ?

$x(00) \longrightarrow \hat{x}(0,0)$ } 1st or
$x(01) \longrightarrow \hat{x}(1,0)$ } even seq — the mapping

$x(10) \longrightarrow \hat{x}(0,1)$ } 2nd or
$x(11) \longrightarrow \hat{x}(1,1)$ } odd seq.

in general

$$x(n_2, n_1, n_0) \longrightarrow x(n_0, n_1, n_2)$$

where n_2, n_1, n_0 are the bits in binary representation of n.

Remarks:

(1) In order to have the output in the natural order, the input array should be shuffled in "bit-reversed order"

ie $x(n_2 n_1 n_0)$ is mapped to $\hat{x}(n_0 n_1 n_2)$

(2) In general for a radix 2 FFT when $N = 2^p$. Then there are p stages.

10-06-88

Lecture (13)

8 point DFT.

$x(000) = x(0)$
$x(100) = x(4)$
$x(010) = x(2)$
$x(110) = x(6)$

} 1st subsequence

$x(001) = x(1)$
$x(101) = x(5)$
$x(011) = x(3)$
$x(111) = x(7)$

} 2nd subsequence

Remarks contd...

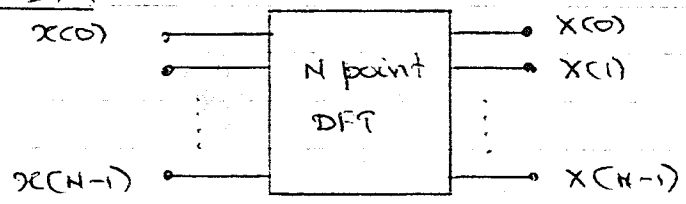
(2) In general for a radix 2 FFT we have $p = \log_2 N$ stages where $N = 2^p$.
At each stage we have $\frac{N}{2}$ complex multiplications by the twiddle factors.

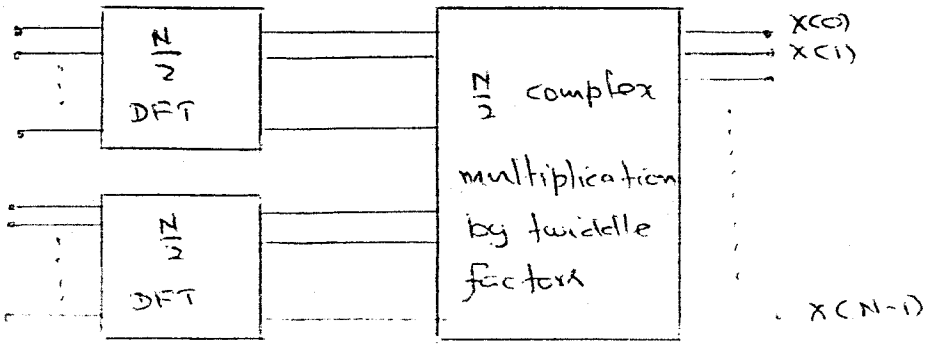
Total # of complex multiplications $\cong \frac{N}{2} \log_2 N$

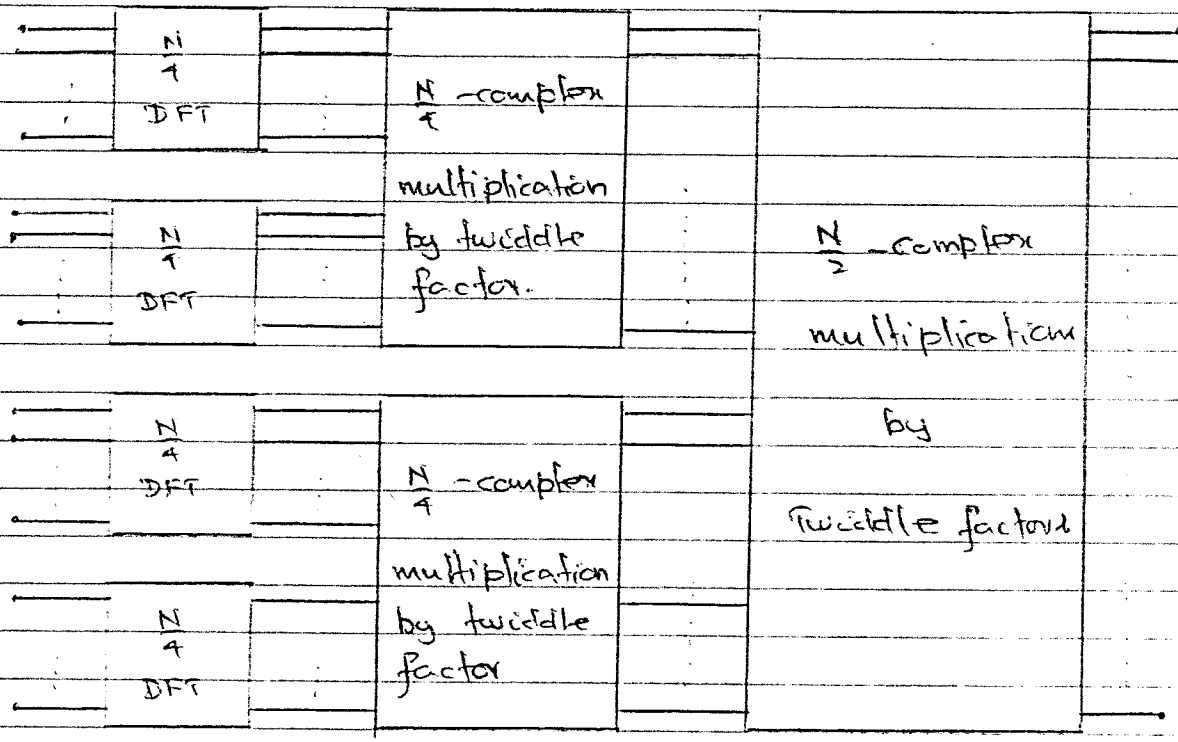
exclude multiplications by W_N^0, $W_N^{\frac{N}{4}}$, $W_N^{\frac{N}{2}}$ and $W_N^{\frac{3N}{4}}$.

$$N_{total} = \frac{N}{2} \log_2 N - N + 1$$

N point DFT







can decompose until end-up with 2-point DFT.

(3) The results of intermediate stages can be stored in the same memory location as the input array was stored. This is called In Place computation.

HW #3.

* arrage a butterfly structure to an 8-point FFT.

EE 512

DIGITAL SIGNAL PROCESSING

Session 16

October 22, 1992

M. Azimi 800-525-4950, ext. 7956

Colorado State University

Digital Signal Processing (EE587)

Assignment 4

Solution

The N-point DFT of sequence $\{x(n)\}$ can be written as

$$X(K) = \sum_{\substack{n=0 \\ n=\text{even}}}^{N-1} x(n) W_N^{nK} + \sum_{\substack{n=0 \\ n=\text{odd}}}^{N-1} x(n) W_N^{nK}$$

$$= \sum_{n=0}^{N/2-1} x_1(n) W_{N/2}^{nK} + W_N^K \sum_{n=0}^{N/2-1} x_2(n) W_{N/2}^{nK}$$

where $x_1(n) = x(2n)$ $\quad n = 0, 1 - - N/2-1$

$x_2(n) = x(2n+1)$ $\quad n = 0, 1 - - N/2-1$

Therefore $X(K) = X_1(K) + W_N^K X_2(K)$

$X_1(K)$ and $X_2(K)$ are $N/2$ - point DFT's of $x_1(n)$ and $x_2(n)$. Note that $N = 2^P$.

For the case of $N = 8$, we have

$$X(K) = \left\{ \left[x(0) + x(4)(W^2)^{2K} \right] + (W^2)^K \left[x(2) + x(6)(W^2)^{2K} \right] \right\}$$

$$+ W^K \left\{ \left[x(1) + x(5)(W^2)^{2K} \right] + (W^2)^K \left[x(3) + x(7)(W^2)^{2K} \right] \right\}$$

$$= X_1(K) + W^K X_2(K)$$

Since $W^4 = -1$ and $W^8 = 1$

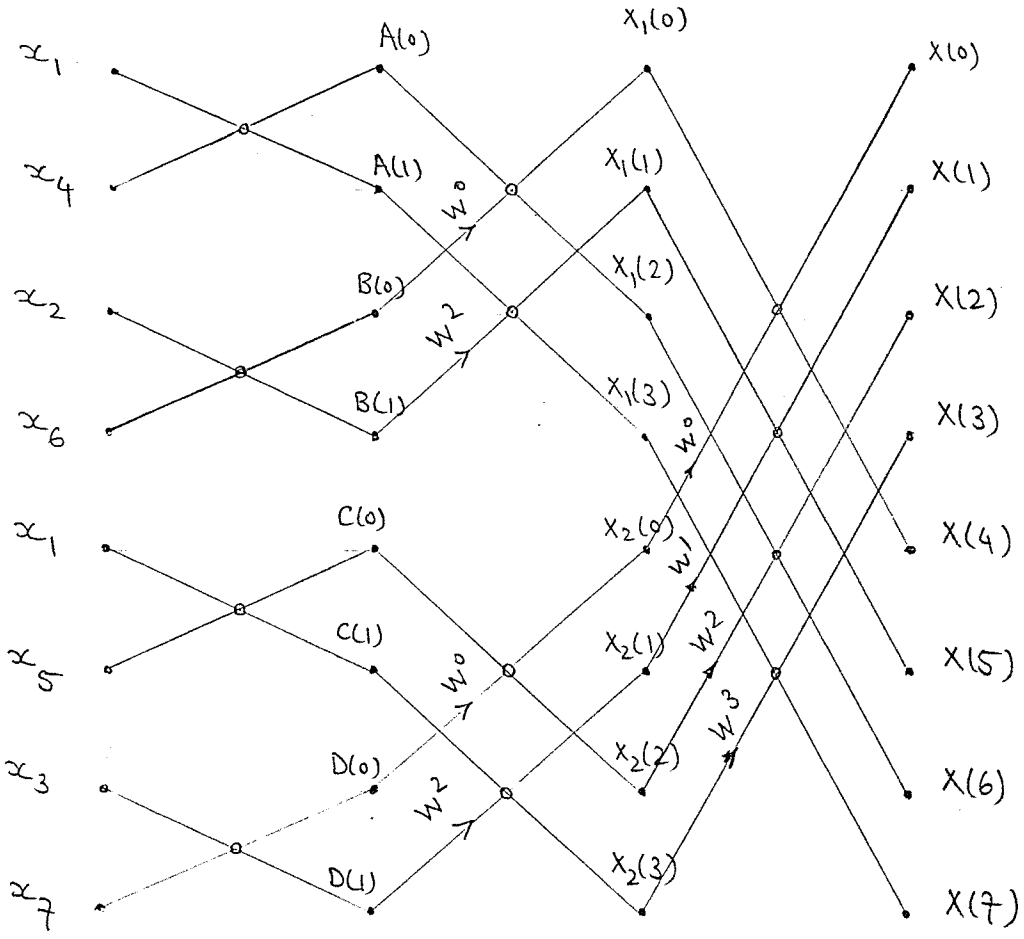
If we denote

$$A(K) = x_0 + x_4 (W^2)^{2K}$$
$$B(K) = x_2 + x_6 (W^2)^{2K}$$
$$C(K) = x_1 + x_5 (W^2)^{2K}$$
$$D(K) = x_3 + x_7 (W^2)^{2K}$$

then we have

$$X(0) = A(0) + B(0) + C(0) + D(0)$$
$$X(1) = A(1) + W^2 B(1) + W[C(1) + D(1)]$$
$$X(2) = A(0) - B(0) + W^2 [C(0) - D(0)]$$
$$X(3) = A(1) - W^2 B(1) + W^3 [C(1) - W^2 D(1)]$$
$$X(4) = A(0) + B(0) - [C(0) + D(0)]$$
$$X(5) = A(1) + W^2 B(1) - W[C(1) + W^2 D(1)]$$
$$X(6) = A(0) - B(0) - W^2 [C(0) - D(0)]$$
$$X(7) = A(1) - W^2 B(1) - W^3 [C(1) - W^2 D(1)]$$

The butterfly structure for an 8-point can then be arranged as shown in the following.


x1
x4
x2
x6
x1
x5
x3
x7
A(0)
A(1)
B(0)
B(1)
C(0)
C(1)
D(0)
D(1)
W^0
W^2
W^0
W^2
X1(0)
X1(1)
X1(2)
X1(3)
X2(0)
X2(1)
X2(2)
X2(3)
W^0
W^1
W^2
W^3
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)