

## COMPUTATION OF THE DISCRETE FOURIER TRANSFORM - PART 1

## 1. Lecture 18 - 49 minutes

a.

Computation of the DFT	Direct Computation	Fast Fourier Transform (FFT)
$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$ $x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}$ $W_N = e^{-j \frac{2\pi}{N}}$	$x(n) W_N^{kn} \Rightarrow \begin{matrix} 1 \text{ complex} \\ \text{multiply} \end{matrix}$ <p>(4 mults, 2 adds)</p> $X(k) \quad k=0, 1, \dots, N-1$ <p><math>N^2</math> complex multiplies  <math>N(N-1)</math> complex adds  <math>\approx N^2</math> MADS</p>	$N = P_1 P_2 \dots P_U$ <p>Complex MADS <math>\propto N [P_1 + P_2 + \dots + P_U]</math></p> $N = 2^U \Rightarrow N \log_2 N$ $X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$ $= \underbrace{\sum_{\substack{n \text{ even} \\ n=2r}} x(n) W_N^{nk}} + \underbrace{\sum_{\substack{n \text{ odd} \\ n=2r+1}} x(n) W_N^{nk}}$ $r = 0, 1, \dots, \frac{N}{2} - 1$

b.

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

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

$$W_N^{(2r+1)k} = W_N^k W_N^{2rk}$$

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

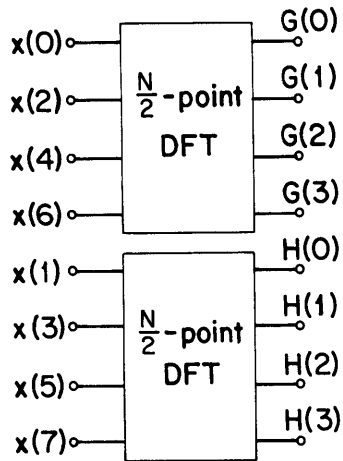
$$X(k) = \underbrace{\sum_{r=0}^{\frac{N}{2}-1} x(2r) W_{N/2}^{rk}}_{\substack{\frac{N}{2} \text{ POINT DFT} \\ G(k)}} + W_N^k \underbrace{\sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_{N/2}^{rk}}_{\substack{\frac{N}{2} \text{ POINT DFT} \\ H(k)}}$$

DFT of a sequence in terms of the DFT of the even and odd numbered points.

$$X(k) = G(k) + W_N^k H(k)$$

$$2\left(\frac{N}{2}\right)^2 + N = N + \frac{N^2}{2}$$

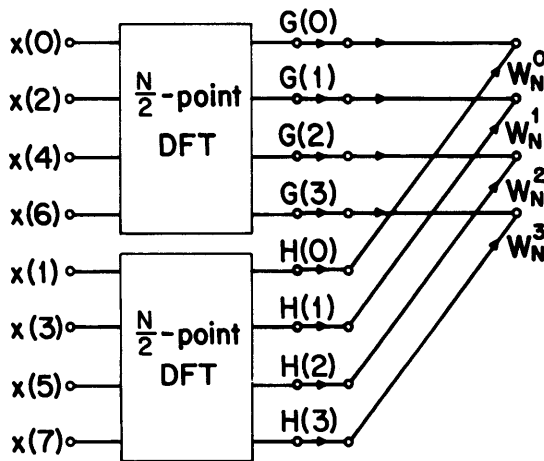
c.



N/2-point DFT's of even and odd-numbered points

$$\Sigma(k) = G(k) + W_N^k H(k)$$

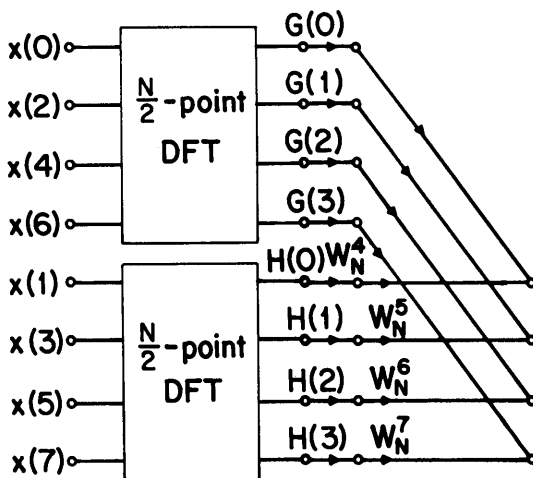
d.



Combination of G(k) and H(k) to obtain first half of X(k)

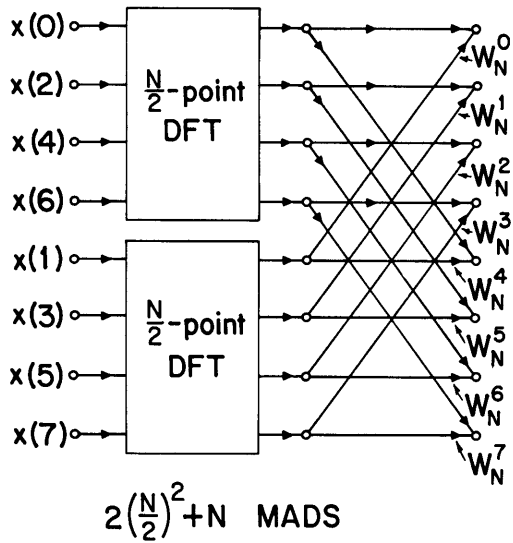
$$\Sigma(k) = G(k) + W_N^k H(k)$$

e.



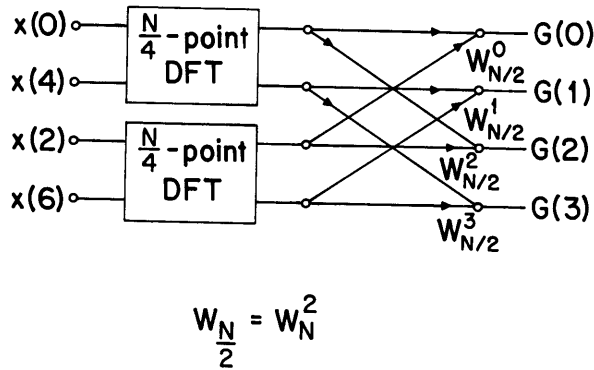
Combination of G(k) and H(k) to obtain second half of X(k)

$$\Sigma(k) = G(k) + W_N^k H(k) \quad \begin{matrix} G(k+4) = G(k) \\ H(k+4) = H(k) \end{matrix}$$



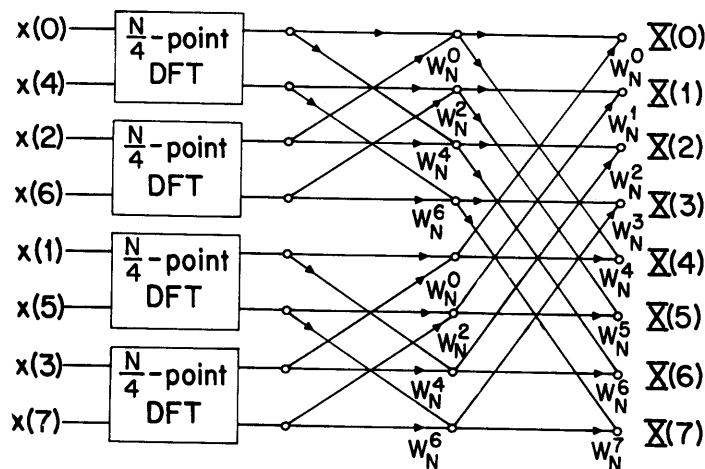
Combination of  $G(k)$  and  $H(k)$  to obtain  $X(k)$

g.



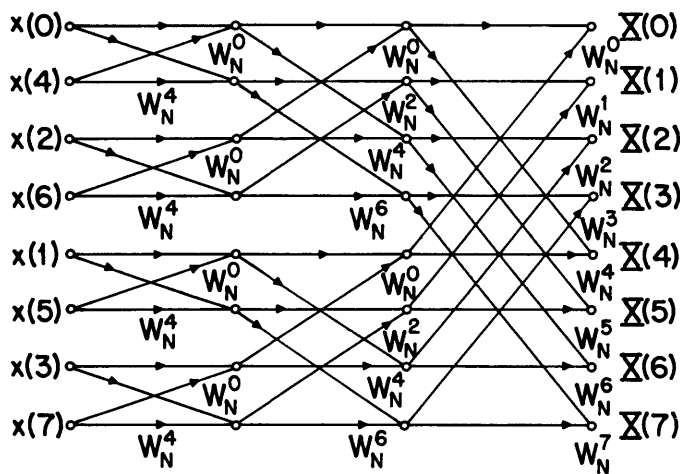
Computation of  $G(k)$  in terms of two  $N/4$ -point DFT's

h.



Computation of  $X(k)$  by combining flow-graph g and h.

i.



Flow-graph for computation of  $X(k)$

j.

$$N = 2^v$$

1 N POINT DFT

2  $\frac{N}{2}$  POINT DFT'S +

4  $\frac{N}{4}$  POINT DFT'S +

8  $\frac{N}{8}$  POINT DFT'S +

$$\begin{array}{c}
 N^2 \\
 \downarrow \\
 2 \left(\frac{N}{2}\right)^2 + N \\
 \downarrow \\
 4 \left(\frac{N}{4}\right)^2 + N + N \\
 \downarrow \\
 8 \left(\frac{N}{8}\right)^2 + N + N + N \\
 \vdots \\
 \underbrace{N + N + \dots + N}_v \text{ TIMES}
 \end{array}$$

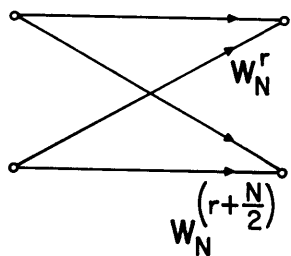
$$\left(\frac{N}{2}\right)^2 - 2\left(\frac{N}{4}\right)^2 + \frac{N}{2}$$

$$\left(\frac{N}{4}\right)^2 - 2\left(\frac{N}{8}\right)^2 + \frac{N}{4}$$

$$\left(\frac{N}{8}\right)^2 - 2\left(\frac{N}{16}\right)^2 + \frac{N}{8}$$

Savings in computation resulting from successive decimation in time

k.

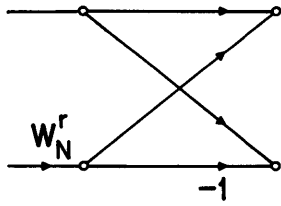


Basic butterfly computation in flow-graph j.

$$W_N^{(r+\frac{N}{2})} = W_N^r W_N^{\frac{N}{2}}$$

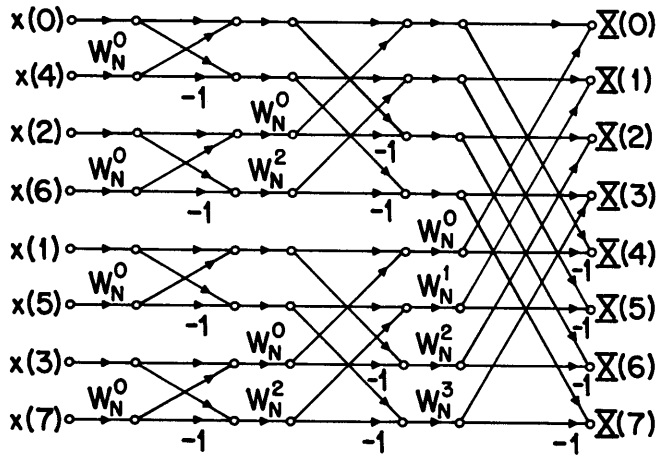
$$W_N^{\frac{N}{2}} = e^{-j\frac{2\pi}{N}\frac{N}{2}} = e^{-j\pi} = -1$$

l.



Rearrangement of the butterfly computation in 1.

m.



Decimation-in-time FFT algorithm utilizing the butterfly computation in m.

n.

---

## 2. Comments

---

The most direct computation of the DFT requires an amount of computation proportional to  $N^2$  where  $N$  is the length of the transform. However by exploiting symmetry and periodicity properties of the complex exponential sequences it is possible to reduce dramatically the amount of computation required. In this lecture we begin a discussion of the set of algorithms which result.

## 3. Reading

---

Text: Sections 9.0 (page 581), 9.1 and 9.3 up to (but not including) section 9.3.1.

## 4. Problems

---

### Problem 18.1

In figure 9.3 of the text (page 589) is shown the flowgraph for implementing an eight-point DFT by first computing two four-point DFT's. Draw the corresponding flow-graph for implementing a sixteen-point DFT by first computing two eight-point DFT's.

### Problem 18.2

Suppose that a program is available for evaluating a DFT,

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)kn}, \quad k = 0, 1, \dots, N-1$$

Show how this same program can be used to compute the inverse DFT,

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(2\pi/N)kn}, \quad n = 0, 1, \dots, N-1$$

---

### Problem 18.3

Consider the FFT algorithm for  $N$  a power of 2, implemented in the form characterized by Figure 9.10 in the text. We are assuming that  $N$  is an arbitrary power of 2, not that  $N = 8$ . In indexing through the data in an array, we shall assume that points in an array are stored in consecutive complex (double) registers numbered 0 through  $N - 1$ . The arrays are numbered 0 through  $\log_2 N$ . The array holding the initial data is the zero<sup>th</sup> array. The output of the first stage of butterflies is the first array, etc. All of the following questions relate to the computation of the  $m^{\text{th}}$  array where  $1 \leq m \leq \log_2 N$ . The answer should be in terms of  $m$ . All of the questions can be answered by generalizing the results for  $N = 8$ .

- (a) How many butterflies are to be computed?
- (b) What are the powers of  $W_N$  involved in computing the  $m^{\text{th}}$  array from the  $(m - 1)$ st array?
- (c) What is the separation between the addresses of the two complex input points to a butterfly?
- (d) What is the separation between the addresses of the first points of butterflies utilizing the same coefficients?

### Problem 18.4

In computing the DFT of real sequences it is possible to reduce the amount of computation by utilizing the fact that the sequence is real.

In particular, consider two real-valued sequences  $x_1(n)$  and  $x_2(n)$  with DFT's  $X_1(k)$  and  $X_2(k)$  respectively. Let  $g(n)$  be the complex sequence given by  $g(n) = x_1(n) + j x_2(n)$ , and let  $G(k)$  be its DFT. Let  $G_{\text{OR}}(k)$ ,  $G_{\text{ER}}(k)$ ,  $G_{\text{OI}}(k)$  and  $G_{\text{EI}}(k)$  denote, respectively, the odd part of the real part, the even part of the real part, the odd part of the imaginary part, and the even part of the imaginary part of  $G(k)$ . Using the fact that for a real sequence, the real part of its DFT is even and its imaginary part is odd (see Table 8.2, page 547 of the text), determine  $x_1(k)$  and  $x_2(k)$  in terms of  $G_{\text{OR}}(k)$ ,  $G_{\text{ER}}(k)$ ,  $G_{\text{OI}}(k)$  and  $G_{\text{EI}}(k)$ .

The result derived in this problem is commonly utilized in one of two ways. If there are two real sequences for which we want the DFT, we can compute their transforms simultaneously and then separate the transforms using the result in the problem. As a second possibility, we can separate a real sequence into two subsequences, consisting of its even and odd numbered points, simultaneously compute the DFT of each of these subsequences, and then combine these to obtain the DFT of the total sequence according to equation (9.14) in the text.

MIT OpenCourseWare  
<http://ocw.mit.edu>

Resource: Digital Signal Processing  
Prof. Alan V. Oppenheim

The following may not correspond to a particular course on MIT OpenCourseWare, but has been provided by the author as an individual learning resource.

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.