7

Fast Fourier transforms

It should be stated immediately that a fast Fourier transform (FFT) is not a 'new transform' but is an *algorithm* for the efficient calculation of the discrete Fourier transform. There are many such methods.

In Chapter 5 we introduced the DFT and considered its properties and inversion. In Chapter 6, largely expository, we investigated computational aspects, and the replacement of the DFT coefficient matrix by a product of sparse matrices was shown to result in a very considerable reduction in effort. In this concluding chapter, the object is to relate the ideas of Chapter 6 to a systematic form for DFT computation. The manner in which we achieve this objective will also show why the factor matrices $\boldsymbol{M}_2, \boldsymbol{M}_4, \boldsymbol{M}_8, \ldots$ have the form they do. (Their definitions in (6.5), (6.8) and (6.12) might have appeared to be somewhat arbitrary, and were only justified by the fact that their application *does* generate the required spectrum.)

Although there are good routines for calculations which involve sparse matrices, the FFT does not incorporate those but directly parallels the structural similarity of factor matrices in an algorithm which can be illustrated graphically (and which also is easy to program.) We whall demonstrate this by decimating the DFT sums algebraically, to obtain the stage equations. That means that at each recursion, any sum is replaced by two sums in each of which the number of terms is halved. The process is repeated until all sums include just two signal values. The results which emerge from this are then considered in reverse order, starting with paired signal values, and recombined through staging until the final output is obtained in the form of a set of numbers satisfying the DFT equations – i.e. the elements of the spectrum. (This is equivalent to the explanation of staging given in the introduction to Chapter 6.) It will be seen that the whole process can be illustrated by constructing a signal flow graph (SFG) of which the essential features are butterfly diagrams and twiddle factors described in Sections 6.6 and 6.7.

We shall consider both decimation in time and decimation in frequency. (The most notable originators of these methods were Cooley, Tukey and Sande. Many alternative and modified FFTs have been developed over the last twenty or so years, with the names of their originators attached. What they have in common is the basic concept of decimation.) As said before, the discussion will be restricted to cases in which N is a power of 2. In Section 7.1 two possible ways of decimating (halving) a sum are outlined.

Once an algorithm has been translated into graphical form, it follows that the resulting SFG can also be used to invert a spectrum, because we showed in Section 5.4 that working with the complex conjugates of signals and spectra allows us to use the same algorithm for both transformation and inversion. This is discussed in detail in Section 7.6 in the context of DIT, and illustrated in Worked Example 7.5 using the decimation-in-frequency SFG.

7.1 FAST FOURIER TRANSFORM ALGORITHMS

In Chapter 6 an estimate was obtained of the reduced computational effort consequent on matrix factorization. The estimated number of real multiplications required using either of the FFTs developed in this chapter is the same, namely $M_F = 2N \log_2 N$.

Halving the DFT sum, referred to in the introduction, is the essential step in the recursive process and, in this context, is done in one of two ways. We have

$$X(n) = \sum_{k=0}^{N-1} x(k) W^{kn} \qquad n = 0, 1, \ldots, N-1 \tag{7.1}$$

in which $W = e^{-j(2\pi/N)}$.

If the decision is made to separate the terms in (7.1) in which k is even from those in which it is odd,

$$X(n) = \sum_{k=2r} x(k) W^{kn} + \sum_{k=2r+1} x(k) W^{kn} \tag{7.2}$$

the eventual outcome is an algorithm known as *decimation in time.* Each of the sums in (7.2) is a sum of $N/2$ terms. All subsequent decimations are carried out on the same basis (for example, the sums are next split according to whether r is even or is odd). The DIT FFT will be described in detail in Sections 7.2–7.5.

An alternative division of the terms in (7.1) is to write

$$X(n) = \sum_{k=0}^{N/2-1} x(k)W^{kn} + \sum_{k=N/2}^{N-1} x(k)W^{kn} \tag{7.3}$$

as a first step and in subsequent stages to distinguish between odd and even values of the *frequency*-domain subscript n. This leads to the *decimation in frequency* algorithm, which will be described in Section 7.7.

Whether considering DIT or DIF, we shall discuss the detail with $N = 8$. For higher values, we would put the appropriate value of N into either (7.2) or (7.3) and follow exactly the same procedures as will be described below.

7.2 DECIMATION IN TIME FOR AN EIGHT-POINT DISCRETE FOURIER TRANSFORM: FIRST STAGE

With $N = 8$, from (7.1) we have

$$X(n) = \sum_{k=0}^{7} x(k)W^{kn} \qquad n = 0, 1, \ldots, 7$$

in which

$$W = W_8 = e^{-j\pi/4}$$

Distinguishing between even and odd values of k, the equivalent of (7.2) written out in full is

$$\begin{aligned} X(n) = &\{x(0)W^0 + x(2)W^{2n} + x(4)W^{4n} + x(6)W^{6n}\} \\ &+ \{x(1)W^n + x(3)W^{3n} + x(5)W^{5n} + x(7)W^{7n}\} \qquad n = 0, 1, \ldots, 7 \end{aligned}$$

Note that W^n is a common factor in the second four-term sum. Let us define

$$F(n) = x(0)W^0 + x(2)W^{2n} + x(4)W^{4n} + x(6)W^{6n} = \sum_{r=0}^{3} x(2r)W^{2rn} \tag{7.4}$$

and

$$G(n) = x(1)W^0 + x(3)W^{2n} + x(5)W^{4n} + x(7)W^{6n} = \sum_{r=0}^{3} x(2r+1)W^{2rn} \tag{7.5}$$

It follows that

$$X(n) = F(n) + W^n G(n) \qquad n = 0, 1, \ldots, 7 \tag{7.6}$$

Now although (7.6) represents eight equations, it is only necessary to calculate $F(n)$ and $G(n)$ for four values of n, because they are periodic functions.

Worked Example 7.1

Show that if $F(n)$ is as defined in (7.4), then $F(n+4) = F(n)$.

Solution

We have

$$F(n+4) = x(0)W^0 + x(2)W^{2(n+4)} + x(4)W^{4(n+4)} + x(6)W^{6(n+4)}$$

However, $W^8 = e^{-j2\pi} = 1$, and so $W^{16} = (W^8)^2 = 1$ and similarly $W^{24} = 1$, whence

$$F(n+4) = x(0)W^0 + x(2)W^{2n} + x(4)W^{4n} + x(6)W^{6n} = F(n)$$

For example, $F(4) = F(0)$, $F(5) = F(1)$ and so on. ●

Both $F(n)$ and $G(n)$ have period 4. As a result it is sufficient to use (7.4) and (7.5) with $n = 0, 1, 2, 3$ only. This means that (7.6) represents the eight equations

$$\begin{aligned}
X(0) &= F(0) + G(0)W^0 \\
X(1) &= F(1) + G(1)W^1 \\
X(2) &= F(2) + G(2)W^2 \\
X(3) &= F(3) + G(3)W^3 \\
X(4) &= F(0) - G(0)W^0 \\
X(5) &= F(1) - G(1)W^1 \\
X(6) &= F(2) - G(2)W^2 \\
X(7) &= F(3) - G(3)W^3
\end{aligned} \tag{7.7}$$

These expressions should be compared with the final output, (6.14), obtained when using the matrix-factorization approach, with particular reference to $\boldsymbol{M}_8$ as given by (6.12).

7.3 THE SECOND STAGE: FURTHER PERIODIC ASPECTS

Equations (7.4) and (7.5) are the subject of the next recursion, and we distinguish between even and odd values of r by putting either

$r = 2p$ or $r = 2p + 1$. From (7.4), explicitly,

$$F(n) = \{x(0)W^0 + x(4)W^{4n}\} + \{x(2)W^{2n} + x(6)W^{6n}\}$$

Here, W^{2n} is a common factor in the second sum. We now define (for $k = 2r = 2(2p) = 4p$),

$$f(n) = x(0)W^0 + x(4)W^{4n} = \sum_{p=0}^{1} x(4p)W^{4pn} \tag{7.8}$$

and (for $k = 2r = 2(2p+1) = 4p + 2$)

$$g(n) = x(2)W^0 + x(6)W^{4n} = \sum_{p=0}^{1} x(4p+2)W^{4pn} \tag{7.9}$$

and then

$$F(n) = f(n) + W^{2n}g(n) \qquad n = 0, 1, 2, 3 \tag{7.10}$$

Reordering (7.5)

$$G(n) = \{x(1)W^0 + x(5)W^{4n}\} + \{x(3)W^{2n} + x(7)W^{6n}\}$$

If we define (for $k = 2r + 1 = 2(2p) + 1 = 4p + 1$)

$$\hat{f}(n) = x(1)W^0 + x(5)W^{4n} = \sum_{p=0}^{1} x(4p+1)W^{4pn} \tag{7.11}$$

and (for $k = 2r + 1 = 2(2p+1) + 1 = 4p + 3$)

$$\hat{g}(n) = x(3)W^0 + x(7)W^{4n} = \sum_{p=0}^{1} x(4p+3)W^{4pn} \tag{7.12}$$

it follows that

$$G(n) = \hat{f}(n) + W^{2n}\hat{g}(n) \qquad n = 0, 1, 2, 3 \tag{7.13}$$

The amount of computation is again halved because the newly introduced function $f(n) \cdots \hat{g}(n)$ are periodic, and the period is 2. Note the similarity in structure of (7.6)–(7.13).

Worked Example 7.2

Show that $\hat{g}(n+2) = \hat{g}(2)$.

Solution

From (7.12), since $W^8 = 1$,

$$\hat{g}(n+2) = x(3) + x(7)W^{4(n+2)} = \hat{g}(n).$$ ●

Equations (7.8), (7.9), (7.11) and (7.12), with $n = 0, 1$ only, therefore provide all the information needed to find $F(n)$ and $G(n)$ from (7.10) and (7.13). Putting $f(2) = f(0), f(3) = f(1)$, etc., and noting that $W^4 = -1$, (7.10) gives

$$\begin{aligned} F(0) &= f(0) + g(0) \\ F(1) &= f(1) + g(1)W^2 \\ F(2) &= f(0) - g(0) \\ F(3) &= f(1) - g(1)W^2 \end{aligned} \tag{7.14}$$

Similarly, from (7.13),

$$\begin{aligned} G(0) &= \hat{f}(0) + \hat{g}(0) \\ G(1) &= \hat{f}(1) + \hat{g}(1)W^2 \\ G(2) &= \hat{f}(0) - \hat{g}(0) \\ G(3) &= \hat{f}(1) - \hat{g}(1)W^2 \end{aligned} \tag{7.15}$$

The coefficients in these equations should be compared with the elements of factor matrix $\boldsymbol{M}_4$, defined in (6.8), and the output from two four-point DFTs, (6.10), is consistent with equations (7.14) and (7.15) if the quantities $f(n)$ and $\hat{g}(n)$ are the same.

7.4 THE THIRD STAGE

In general we would repeat the process by next distinguishing between $p = 2q$ and $p = 2q + 1$, and halve the sums defining $f(n)$ and $\hat{g}(n)$. It is not necessary in this case as we have now arrived at pairings of signal values, and so the third and last recursion results in one-term 'sums', $x_0, x_1, \ldots, x_7$. Explicitly, we have, from (7.8), (7.9), (7.11) and (7.12)

$$\begin{aligned} f(0) &= x_0 + x_4 \\ f(1) &= x_0 - x_4 \\ g(0) &= x_2 + x_6 \\ g(1) &= x_2 - x_6 \\ \hat{f}(0) &= x_1 + x_5 \\ \hat{f}(1) &= x_1 - x_5 \\ \hat{g}(0) &= x_3 + x_7 \\ \hat{g}(1) &= x_3 - x_7 \end{aligned} \tag{7.16}$$

The breakdown is therefore fully consistent with the definition of $\boldsymbol{M}_2$ given in (6.6) and the output from the application of $\boldsymbol{M}_2$ of the bit-reversed signal vector, (6.7).

7.5 CONSTRUCTION OF A FLOW GRAPH

Equations (7.16) give the outputs from four two-point DFTs and there are no complex coefficients. We take these as the starting point to obtain the SFG for an eight-point DIT FFT. We shall use the standard butterfly diagram (Fig. 6.2(b)) throughout.

As the signal values are combined in bit-reversed pairs, we enter them in that order on the left of an eight-line diagram, of which the top four lines are associated with even-subscripted x_k, as shown in Fig. 7.1.

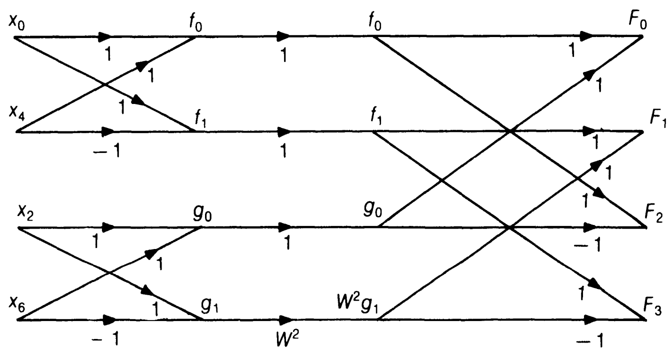

Fig. 7.1 The start of the FFT, even-subscripted inputs.

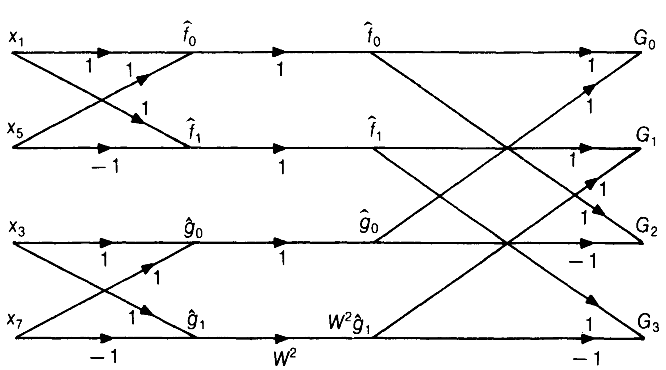

Fig. 7.2 Initial treatment of odd-subscripted inputs.

The first two butterflies give the first four quantities in (7.16) at the output nodes. Now refer to (7.14). This represents a four-point DFT in which $g(1)$ has a coefficient $W^2 = -\mathrm{j}$. On the bottom line of Fig. 7.1 we have indicated a twiddle factor W^2, to give the relevant value. The next two butterflies again form the sums and differences of two inputs. (Notice that the butterflies now combine values from nodes which are *not* on adjacent lines, and confirm that the outputs F_n are as given by (7.14).)

To compute $\hat{f}(0)$, $\hat{f}(1)$, $\hat{g}(0)$ and $\hat{g}(1)$ from (7.16) we enter the odd-subscripted values of x_k, again in bit-reversed order, and the lower four lines of our eight-point DIT FFT chart will be as shown in Fig. 7.2. Having recorded those quantities at the output nodes of the first butterflies, the first stage is completed as far as the *lower* half of the graph is concerned. We proceed to the four-point DFT defined by (7.15), noting that in the computation of $\mathbf{G}, \hat{g}(1)$ is to be multiplied by W^2 (hence the premultiplier on the bottom line).

On the complete (eight-line) diagram there are now recorded the eight numbers **F** and **G** as outputs from the two four-point DFTs, and it remains to compute the eight-point DFT giving the required spectrum $X(n)$. From (7.7) we see that G_1, G_2 and G_3 on our chart should be multiplied by W^1, W^2 and W^3, respectively, to obtain the numbers required at the relevant input nodes of the final butterflies. In Fig. 7.3, therefore, twiddle factors appear on the last three lines.

We see that the final output, $X(n)$, appears in natural order, $n = 0, 1, 2, \ldots, 7$. In the complete SFG we have three stages, and there are four butterfly calculations at each stage. In the SFG for an N-point FFT, if $N = 2^p$ there will be p stages, each of $N/2$ butterflies. Figure 7.4 is a 'blank' chart showing just the multipliers required, further simplified by using an unmarked black arrow to indicate a factor (-1), and omitting altogether the arrows indicating $(+1)$. This illustrates the simplification of the SFG in respect of factors ± 1 which was described at the end of Section 6.8. The reader might prefer to retain all arrows and explicit labels when first making use of a SFG, but after some experience it is sufficient to remember that, of the two outputs from any standard butterfly, the sum of the two inputs is recorded at the upper node and their difference at the lower node.

Depending on the nature of the inputs, twiddle factors can be left in exponential form, or it might be more convenient to use the trigonometric or Cartesian equivalent. This also depends on whether the calculations are being done manually or using a programmed

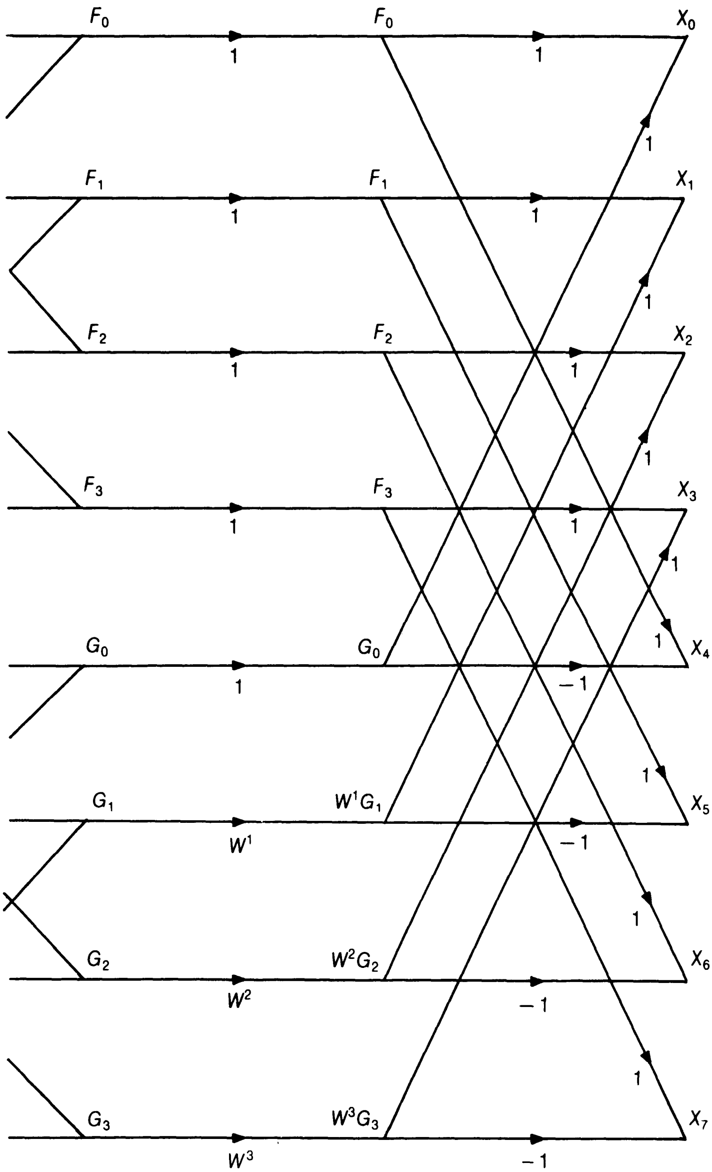

Fig. 7.3 The final stage of the DIT eight-point DFT.

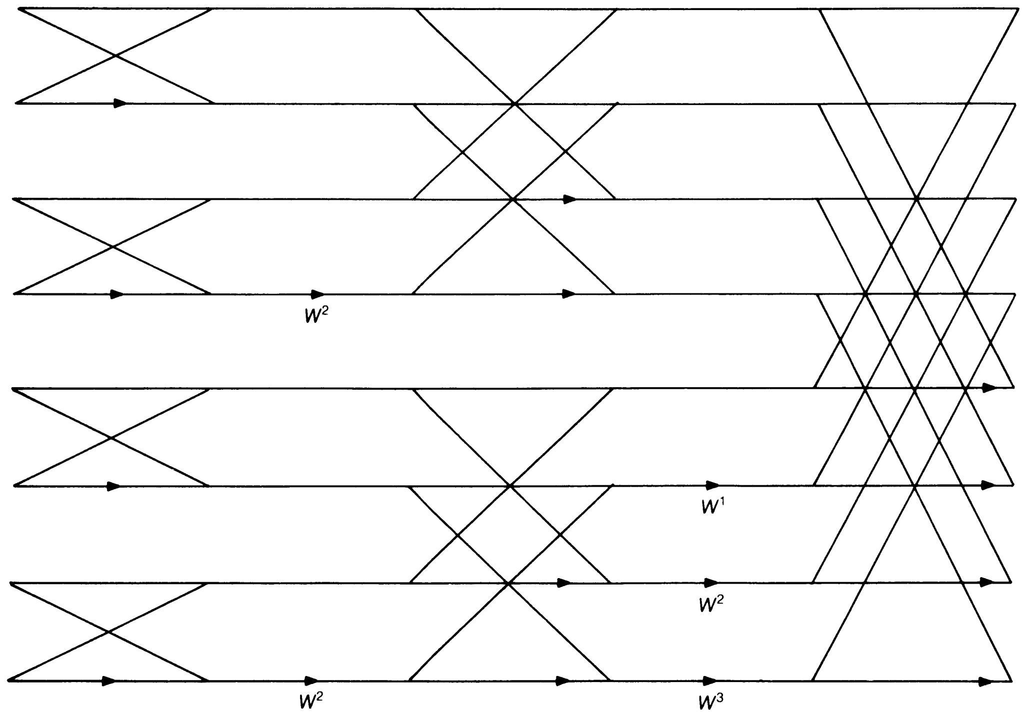

Fig. 7.4 The SFG for an eight-point DIT FFT.

version of the SFG. With $N = 8$ we have

$$W^1 = \mathrm{e}^{-\mathrm{j}\pi/4} = \cos\left(\frac{\pi}{4}\right) - \mathrm{j}\sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}}(1-\mathrm{j})$$

$$W^2 = \mathrm{e}^{-\mathrm{j}\pi/2} = \cos\left(\frac{\pi}{2}\right) - \mathrm{j}\sin\left(\frac{\pi}{2}\right) = -\mathrm{j}$$

$$W^3 = \mathrm{e}^{-\mathrm{j}3\pi/4} = -\cos\left(\frac{3\pi}{4}\right) - \mathrm{j}\sin\left(\frac{3\pi}{4}\right) = -\frac{1}{\sqrt{2}}(1+\mathrm{j}) \qquad (7.17)$$

In general, $N/2 - 1$ twiddle factors are needed.

Worked Example 7.3

Use the Cooley–Tukey FFT illustrated in Fig. 7.4 to find the spectrum of the signal $\mathbf{x}_k = (1,1,0,0,0,0,0,1)^{\mathrm{T}}$, $k = 0, 1, \ldots, 7$.

Solution (The reader should obtain the following from the actual SFG.)

We start by putting $\mathbf{x}_k$ into bit-reversed order, and will use the Cartesian form for factors W^1, W^2, W^3. Input and output values at nodes, for the three sets of butterflies, are then given by six column vectors as follows:

	← Stage 1 →		← Stage 2 →		← Stage 3 →		
k	*Input*	Output	Input	Output	Input	*Output*	n
0	1	1	1	1	1	3	0
4	0	1	1	1	1	$1+\sqrt{2}$	1
2	0	0	0	1	1	1	2
6	0	0	0	1	1	$1-\sqrt{2}$	3
1	1	1	1	2	2	-1	4
5	0	1	1	$1+\mathrm{j}$	$\sqrt{2}$	$1-\sqrt{2}$	5
3	0	1	1	0	0	1	6
7	1	-1	j	$1-\mathrm{j}$	$-\sqrt{2}$	$1+\sqrt{2}$	7

The stage 1 input is the bit-reversed $\mathbf{x}_k$ and the stage 3 output is $\mathbf{X}_n$

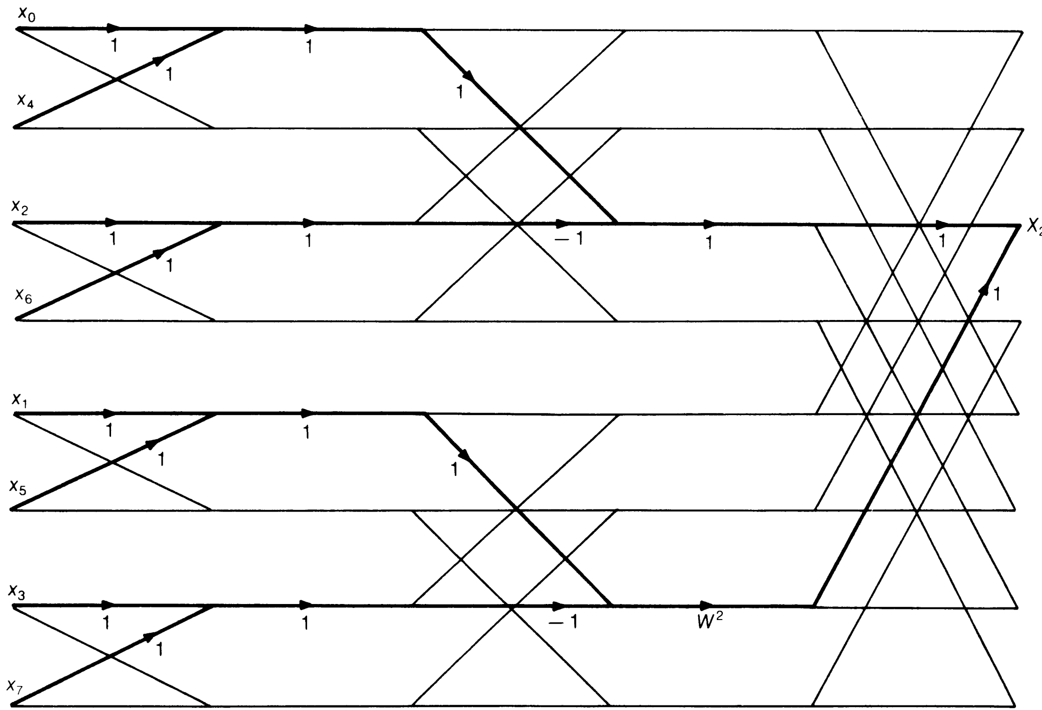

Fig. 7.5 The composition of X_2.

in natural order. (We may observe that $\sum|x_k|^2 = 3$ and $\sum|X_n|^2 = 24$, and Parseval's theorem is satisfied.) In all these vectors, the elements are ordered in the same way as in Figs 7.1, 7.2 and 7.3 from top to bottom as they appear on the SFG. They were obtained by working directly on the graph, and we have presented them in this form for clarity. ●

To make it absolutely clear that we can use a chart without reference to equations we can select a particular element of the spectrum and trace backwards through the graph until we have a 'map' showing through what multiplicative routes the various input elements contribute to the selected output element. Suppose we do this for X_2. In Fig. 7.5 we have highlighted the various lines and diagonals in question, and not included any multipliers that are not relevant to the computation of $\mathbf{X}_2$.

If we next look at each input in turn, its connection to X_2 shows by what it is multiplied within and between butterflies. We obtain

$$\begin{aligned} X_2 = (1)x_0 + (1)x_4 + (-1)x_2 + (-1)x_6 + (W^2)x_1 \\ + (W^2)x_5 + (-1\cdot W^2)x_3 + (-1\cdot W^2)x_7 \end{aligned}$$

which is the result given by $\mathbf{X} = \boldsymbol{M}\mathbf{x}$, in which $\boldsymbol{M}$ has been simplified but not factorized. (The reader might repeat this exercise for another element, say X_3.)

7.6 INVERSION USING THE SAME DECIMATION-IN-TIME SIGNAL FLOW GRAPH

In Section 5.4 we showed that if we regard $\{\bar{X}_n\}$ as an input sequence then its DFT is the sequence $\{N\bar{x}_k\}$. This result appeared in (5.13), and since it was derived solely from the transform pair after looking at the conjugates of those equations we can use the DIT chart of Fig. 7.4 to invert a known spectrum $\{X_n\}$. It is therefore now $\{X_n\}$ which is bit-reversed and then conjugated. The output will be $\{N\bar{x}_k\}$ in natural order, so division by N and a second conjugation leads to $\{x_k\}$.

Worked Example 7.4

Given that $N = 8$ and that a spectrum is

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} 3 \\ (1+\sqrt{2})j \\ 1 \\ (1-\sqrt{2})j \\ -1 \\ (1-\sqrt{2})j \\ 1 \\ (1+\sqrt{2})j \end{bmatrix}$$

find the inverse.

Solution

We can present the values to be recorded at the SFG nodes in the form of vectors, as in Worked Example 7.3. Again we recommend verifying these results by working on a copy of the graph.

$$\begin{array}{c} \\ n \\ \begin{matrix} 0 \\ 4 \\ 2 \\ 6 \\ 1 \\ 5 \\ 3 \\ 7 \end{matrix} \end{array}
\begin{array}{cc} \multicolumn{2}{c}{\longleftarrow \text{Stage 1} \longrightarrow} \\ \textit{Input} & \text{Output} \\ \begin{bmatrix} 3 \\ -1 \\ 1 \\ 1 \\ -(1+\sqrt{2})j \\ -(1-\sqrt{2})j \\ -(1-\sqrt{2})j \\ -(1+\sqrt{2})j \end{bmatrix} & \begin{bmatrix} 2 \\ 4 \\ 2 \\ 0 \\ -2j \\ -2\sqrt{2}j \\ -2j \\ 2\sqrt{2}j \end{bmatrix} \end{array}
\begin{array}{cc} \multicolumn{2}{c}{\longleftarrow \text{Stage 2} \longrightarrow} \\ \text{Input} & \text{Output} \\ \begin{bmatrix} 2 \\ 4 \\ 2 \\ 0 \\ -2j \\ -2\sqrt{2}j \\ -2j \\ 2\sqrt{2} \end{bmatrix} & \begin{bmatrix} 4 \\ 4 \\ 0 \\ 4 \\ -4j \\ 2\sqrt{2}(1-j) \\ 0 \\ -2\sqrt{2}(1+j) \end{bmatrix} \end{array}
\begin{array}{cc} \multicolumn{2}{c}{\longleftarrow \text{Stage 3} \longrightarrow} \\ \text{Input} & \textit{Output} \\ \begin{bmatrix} 4 \\ 4 \\ 0 \\ 4 \\ -4j \\ -4j \\ 0 \\ 4j \end{bmatrix} & \begin{bmatrix} 4(1-j) \\ 4(1-j) \\ 0 \\ 4(1+j) \\ 4(1+j) \\ 4(1+j) \\ 0 \\ 4(1-j) \end{bmatrix} \end{array}
\begin{array}{c} \\ k \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{array}$$

In this, the stage 1 input is $\{X_n\}$ after bit-reversal and conjugation. The stage 3 output is $\{8\bar{x}_k\}$, and so we conclude that $x_0 = x_1 = x_7 = \frac{1}{2}(1+j)$, $x_2 = x_6 = 0$, and $x_3 = x_4 = x_5 = \frac{1}{2}(1-j)$. Similarly, any other FFT graph can be used for both transformation and inversion. ●

7.7 DECIMATION IN FREQUENCY FOR AN EIGHT-POINT DISCRETE FOURIER TRANSFORM

In Section 7.1 we suggested that another way of replacing an N-term summation by two sums of $N/2$ terms would be to halve their natural ordering. With $N = 8$, the equivalent of (7.3) is

$$X(n) = \sum_{k=0}^{3} x(k)\, W^{kn} + \sum_{k=4}^{7} x(k) W^{kn} \qquad n = 0, 1, \ldots, 7$$

where $W = W_8 = e^{-j\pi/4}$.

It is more convenient if both sums have the same limits. In the second, putting $k = m + 4$ gives

$$\sum_{k=4}^{7} x(k) W^{kn} = \sum_{m=0}^{3} x(m+4) W^{(m+4)n}$$

The factor W^{4n} can be written before the summation sign, and we note that $W^{4n} = (e^{-j\pi})^n = (-1)^n$. Changing the dummy summation integer again, putting $m = k$, the eight-point DFT $X(n)$ can now be expressed

$$X(n) = \sum_{k=0}^{3} x(k) W^{kn} + (-1)^n \sum_{k=0}^{3} x(k+4) W^{kn} \qquad n = 0, 1, \ldots, 7 \tag{7.18}$$

Decimation in the *frequency* domain means that we are dividing the *output*, distinguishing between even and odd values of n (not k). For the cases $n = 2r$ and $n = 2r + 1$, from (7.18) we have

$$X(2r) = \sum_{k=0}^{3} [x(k) + x(k+4)] W^{2kr} \qquad r = 0, 1, 2, 3 \tag{7.19}$$

and

$$X(2r+1) = \sum_{k=0}^{3} [x(k) - x(k+4)] W^{k} \cdot W^{2kr} \qquad r = 0, 1, 2, 3 \tag{7.20}$$

These can be thought of as two four-point DFTs if regarded as the transforms of sequences

$$\{y_k\} = \{x_k + x_{k+4}\}$$

and

$$\{y_k\} = \{(x_k - x_{k+4}) W^k\}$$

respectively, because $W_8^{2r} = W_4^r$. Subsequent decimations also represent DFTs.

We shall continue the process, but in terms of $\{x_k\}$ throughout.

The sums in (7.19) and (7.20) are halved in the next decimation and, where $\sum_{k=2}^{3}$ is seen, the substitution $k = m + 2$ will enable us to express all results as sums over $k = 0, 1$.

As an example, we will consider (7.19) with $r = 2p$, and write

$$X(4p) = \sum_{k=0}^{1} [x(k) + x(k+4)]W^{4kp} + \sum_{k=2}^{3} [x(k) + x(k+4)]W^{4kp} \tag{7.21}$$

Putting $k = m + 2$ in the second sum gives

$$\sum_{m=0}^{1} [x(m+2) + x(m+6)]W^{4(m+2)p}$$

in which $W^{8p} = 1$ has appeared. Replacing m by k we have, from (7.21),

$$X(4p) = \sum_{k=0}^{1} \{[x(k) + x(k+4)] + [x(k+2) + x(k+6)]\}W^{4kp}$$

In full (i.e. putting $k = 0$ and 1),

$$X(4p) = (x_0 + x_4) + (x_2 + x_6) + (x_1 + x_5)W^{4p} + (x_3 + x_7)W^{4p} \qquad p = 0, 1 \tag{7.22}$$

Putting $r = 2p$ in (7.20) leads to

$$X(4p+1) = (x_0 - x_4) + (x_2 - x_6)W^2 + (x_1 - x_5)W^{4p+1} + (x_3 - x_7)W^{4p+3} \qquad p = 0, 1 \tag{7.23}$$

and substituting $r = 2p + 1$ in (7.19) and (7.20) gives

$$X(4p+2) = (x_0 + x_4) - (x_2 + x_6) + (x_1 + x_5)W^{4p+2} - (x_3 + x_7)W^{4p+2} \qquad p = 0, 1 \tag{7.24}$$

and

$$X(4p+3) = (x_0 - x_4) - (x_2 - x_6)W^2 + (x_1 - x_5)W^{4p+3} - (x_3 - x_7)W^{4p+5} \qquad p = 0, 1 \tag{7.25}$$

The confirmation of these results is asked for in Problem 7.6.

The third decimation is to distinguish between cases $p = 0$ (even) and $p = 1$ (odd), but the consequences are sufficiently apparent from the preceding four equations and need not be made explicit. On first inspection, it might appear that four butterfly calculations should be carried out using bit-reversed inputs. This is a misleading impression as one could not then accommodate the required coefficients in subsequent stages without in effect reproducing the decimation in *time* SFG, which is not our intention. In several texts, the regrouping of terms in (7.22)–(7.25) is completed and the SFG constructed accordingly, with $\{x_k\}$ entered in natural order. This we leave for the reader to do, in Problem 7.12. Instead, we will illustrate how the SFG can be obtained from the factorized matrix, appropriately transposed.

The factor matrices defined in Chapter 6 were formally verified as they emerged naturally in the development of the DIT FFT (Sections 7.2, 7.3 and 7.4), and in Section 6.5 it was established that the product $\boldsymbol{M}_2^{\mathrm{T}}\boldsymbol{M}_4^{\mathrm{T}}\boldsymbol{M}_8^{\mathrm{T}}\mathbf{x}_k$ will give $\mathbf{X}_n$ in bit-reversed order (if the permutation matrix $\boldsymbol{E}$ is omitted).

Transposing $\boldsymbol{M}_8$, given by (6.12), the first stage output is

$$\boldsymbol{M}_8^{\mathrm{T}}\mathbf{x}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & W^1 & 0 & 0 & 0 & -W^1 & 0 & 0 \\ 0 & 0 & W^2 & 0 & 0 & 0 & -W^2 & 0 \\ 0 & 0 & 0 & W^3 & 0 & 0 & 0 & -W^3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$= \begin{bmatrix} (x_0+x_4) \\ (x_1+x_5) \\ (x_2+x_6) \\ (x_3+x_7) \\ (x_0-x_4) \\ (x_1-x_5)W^1 \\ (x_2-x_6)W^2 \\ (x_3-x_7)W^3 \end{bmatrix} \tag{7.26}$$

With reference to Fig. 7.6, we see that if we enter $\{x_k\}$ in natural order on a graph and carry out an eight-point DFT at the *first* stage, the butterfly outputs, suitably postmultiplied by twiddle factors, provide the values obtained in (7.26).

These form the input node values at the next stage. Standard butterfly calculations applied to the stage 2 four-point DFTs, with postmultiplication by W^2 on the lowest line of each, gives the values $\boldsymbol{M}_4^{\mathrm{T}}(\boldsymbol{M}_8^{\mathrm{T}}\mathbf{x})$. These form the inputs to four two-point butterflies, and the final output is $\mathbf{X}_n$ in bit-reversed order. (Verification of these statements is asked for in Problem 7.7.)

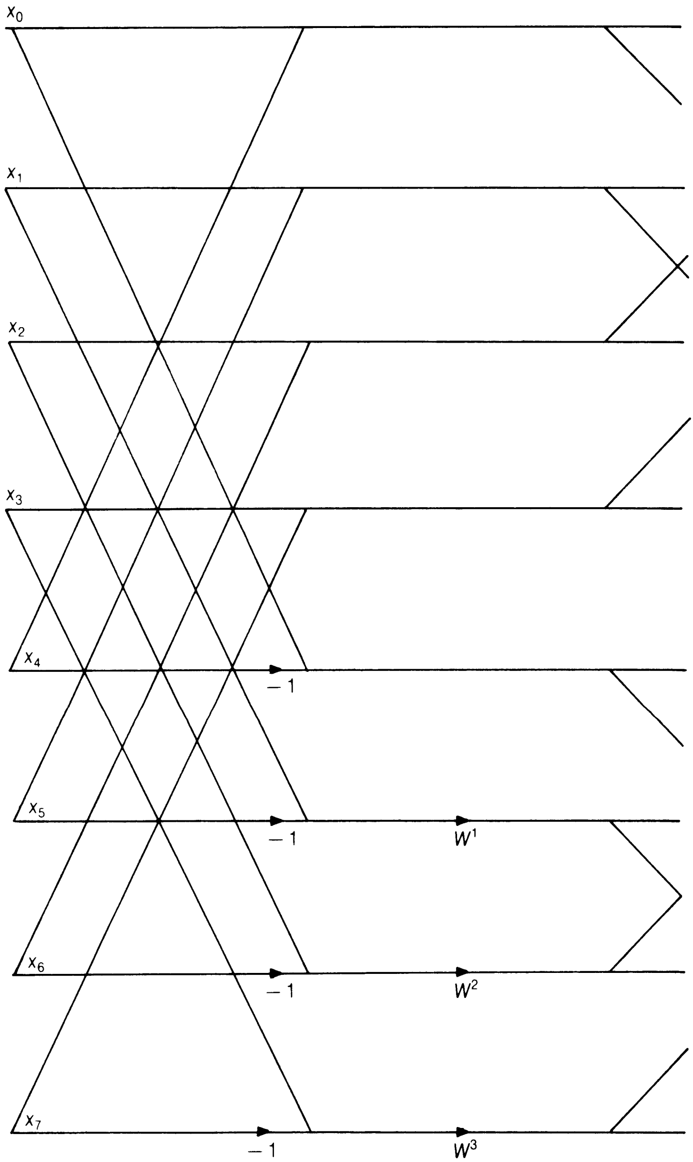

Fig. 7.6 The first stage of a DIF chart.

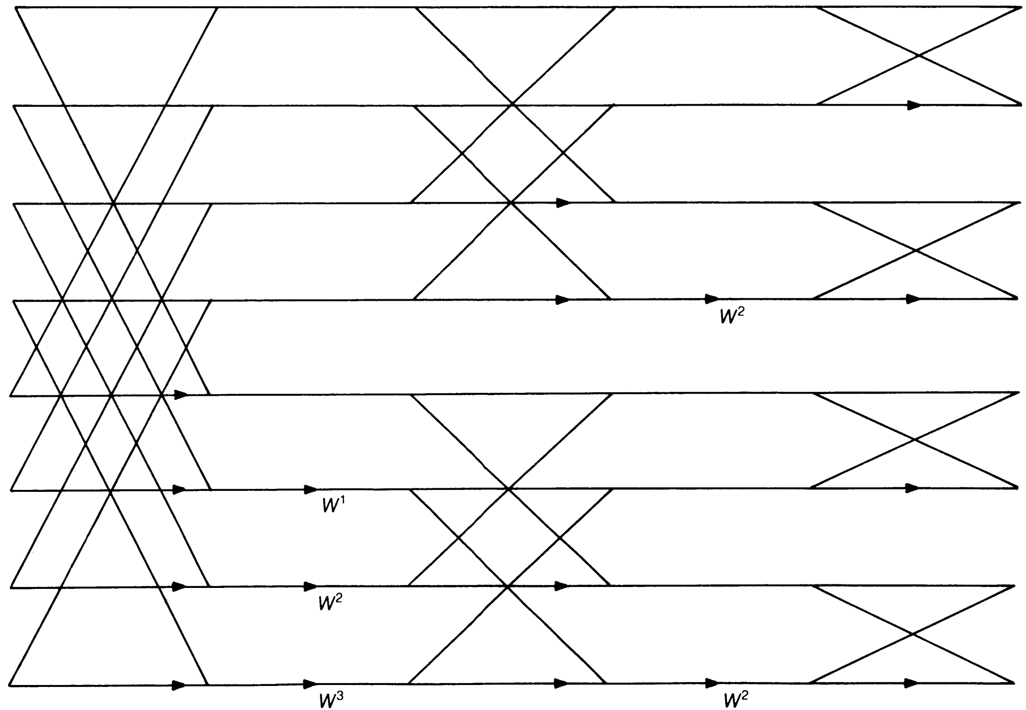

Fig. 7.7 The SFG for an eight-point DIF FFT.

The SFG for the DIF FFT is shown in Fig. 7.7. It is a left-to-right mirror image of the SFG for the DIT FFT which appeared as Fig. 7.4. Figure 7.7 also can be used for inversion, the difference being that $\{\bar{X}_n\}$ must now be entered in *natural* order, and the output will be $\{N\bar{x}_k\}$ in bit-reversed order.

Worked Example 7.5

Invert the spectrum $X(n) = 1 + \mathrm{j}^n$ using decimation in frequency. (This was done 'directly' in Problem 5.10.)

Solution

Tabulating X_n and its conjugate, we have

n	0	1	2	3
X_n	2	$1+\mathrm{j}$	0	$1-\mathrm{j}$
$\bar{X}_n$	2	$1-\mathrm{j}$	0	$1+\mathrm{j}$
n	4	5	6	7

and $\bar{X}_n$ is to be entered in natural order. Values found at the nodes, using Fig. 7.7, are as follows:

	← Stage 1 →		← Stage 2 →		← Stage 3 →		
n	*Input*	Output	Input	Output	Input	*Output*	k
0	2	4	4	4	4	8	0
1	$1-\mathrm{j}$	$2-2\mathrm{j}$	$2-2\mathrm{j}$	4	4	0	4
2	0	0	0	4	4	0	2
3	$1+\mathrm{j}$	$2+2\mathrm{j}$	$2+2\mathrm{j}$	$-4\mathrm{j}$	-4	8	6
4	2	0	0	0	0	0	1
5	$1-\mathrm{j}$	0	0	0	0	0	5
6	0	0	0	0	0	0	3
7	$1+\mathrm{j}$	0	0	0	0	0	7

The stage 1 butterfly output and the stage 2 input are the same because all twiddle factors are applied to zeros. The output, $8\bar{x}_k$, is real, and the second conjugation is therefore redundant. Bit-reversal shows that the only non-zero values are $x_0 = x_6 = 1$.

SUMMARY

In Chapter 7 we have shown that a decimation process which replaces a set of N equations by sets of N, $N/2$, $N/4, \ldots, 2$ equations, all of a much simpler nature, can be used to construct a signal flow graph for the computation of a DFT. We considered the decimation-in-time FFT in some detail, and in particular showed that it gave an explanation of the origin of the factor matrices defined in Chapter 6, as their elements are identifiable from the butterfly calculations and twiddle factors used in each recursion. We have described how an alternative algebraic decimation process can be used in the frequency domain, but reverted to the concept of matrix factorization (that having by then been validated) to produce the related SFG. In Problems 7.11 and 7.12 we invite any determined reader to extend the DIT algorithm to a case in which $N > 8$, and to compute the DIF algorithm *without* recourse to the factored coefficient matrix.

PROBLEMS

7.1 (i) Show that $G(n+4) = G(n)$, where $G(n)$ is defined by (7.5).
(ii) Show that the functions $f(n)$, $g(n)$ and $\hat{f}(n)$, defined in (7.8), (7.9) and (7.11), have period 2.

7.2 The sequence $\mathbf{x}_k, k = 0, 1, 2, \ldots, 7$ given by

$$\mathbf{x}_k = (0, \tfrac{1}{4}, \tfrac{1}{2}, \tfrac{3}{4}, 1, \tfrac{3}{4}, \tfrac{1}{2}, \tfrac{1}{4})^{\mathrm{T}}$$

is the sampled Λ-function which appeared in Problem 5.11. Working on the DIT FFT chart (Fig. 7.4), with initial bit-reversal, show that it has a spectrum as stated in that earlier question.

7.3 Proceeding as in Worked Example 7.2, show that

$$\mathbf{x}_k(0, 0, 1, -1+\sqrt{2}, 2-\sqrt{2}, -2+\sqrt{2}, 1-\sqrt{2}, -1)^{\mathrm{T}}$$

has a spectrum given by $X_1 = (-4 + 2\sqrt{2}) - 2\sqrt{2}\mathrm{j}, X_4 = 8 - 4\sqrt{2}$, $X_7 = \bar{X}_1$, and $X_0 = X_2 = X_3 = X_5 = X_6 = 0$.

7.4 Use the DIT FFT to invert the spectrum $\{X_n\}$ obtained in the previous example, showing that $\mathbf{x}_k$ as defined is recovered.

7.5 Invert the spectrum

$$\{X_n\} = \{0, 0, 8, 0, 0, 0, 0, 0\}^{\mathrm{T}}$$

using the DIT SFG, showing that $\{x_k\}$ is given by $x_k = \mathrm{e}^{\mathrm{j}k\pi/2}$, $k = 0, 1, \ldots, 7$.

7.6 From (7.19) and (7.20), obtain the expressions for $X(4p+1)$, $X(4p+2)$ and $X(4p+3)$ given in (7.23), (7.24) and (7.25).

7.7 With values of $\boldsymbol{M}_8^{\mathrm{T}}\mathbf{x}_k$ as given in (7.26), verify that $\boldsymbol{M}_4^{\mathrm{T}}(\boldsymbol{M}_8^{\mathrm{T}}\mathbf{x}_k)$ is the same as the input to the third-stage butterflies on the SFG shown in Fig. 7.7, and that completion of the calculations produces the spectrum $\mathbf{X}_n$ in bit-reversed order. ($\boldsymbol{M}_4$ was defined by (6.8).)

7.8 Repeat Problem 7.5, but now use the DIF SFG to invert the spectrum.

7.9 $\{x_k\}, k = 0, 1, \ldots, 7$ is given by

$$\mathbf{x}_k = (0, 0, \mathrm{j}, 1, 0, -\mathrm{j}, -1, 0)^{\mathrm{T}}$$

Use the decimation-in-time algorithm to show that

$$\mathbf{X}_n = (0, (1-j), 0, (\sqrt{2}-1)(1-j), -2(1-j), (1-j),$$
$$2(1-j), -(\sqrt{2}+1)(1-j)), \qquad n = 0, 1, \ldots, 7$$

Sketch the amplitude spectrum $|X_n|$. Show that Parseval's theorem is satisfied.

7.10 Use the decimation-in-frequency SFG to find the spectrum of $\mathbf{x}_k$ as defined in Problem 7.9 and also to invert $\mathbf{X}_n$.

7.11 By following the methods used in Sections 7.2–7.5, describe how you would develop a fast DIT algorithm for a 16-point DFT.

7.12 With reference to decimation in frequency and to equations (7.18)–(7.25), if $\mathbf{x}_k$ is to be entered on a chart in natural order, how would you use those results to construct the SFG?