## **Chapter 12**

# **The Mixed-Radix and Split-Radix FFTs**

#### **12.1 The Mixed-Radix FFTs**

There are two kinds of mixed-radix FFT algorithms. The first kind refers to a situation arising naturally when a radix-*q* algorithm, where  $q = 2^m > 2$ , is applied to an input series consisting of  $N = 2^k \times q^s$  equally spaced points, where  $1 \leq k \leq m$ . In this case, out of necessity, *k* steps of radix-2 algorithm are applied either at the beginning or at the end of the transform, while the rest of the transform is carried out by *s* steps of the radix-*q* algorithm. For example, if  $N = 2^{2m+1} = 2 \times 4^m$ , the mixed-radix algorithm combines one step of the radix-2 algorithm and  $m$  steps of the radix-4 algorithm.<sup>1</sup>

The second kind of mixed-radix algorithms in the literature refers to those specialized for a composite  $N = N_0 \times N_1 \times \cdots \times N_k$ . Different algorithms may be used depending on whether the factors satisfy certain restrictions. The FFT algorithms for composite *N* will be treated in Chapter 15.

### **12.2 The Split-Radix DIT FFTs**

After one has studied the radix-2 and radix-4 FFT algorithms in Chapters 3 and 11, it is interesting to see that the computing cost of the FFT algorithm can be further reduced by combining the two in a split-radix algorithm. The split-radix approach was first proposed by Duhamel and Hollmann in 1984 [39]. There are again DIT versions and DIF versions of the algorithm, depending on whether the input time series or the output frequency series is decimated.

The split-radix DIT algorithm is derived from (3.1), which defines the discrete

<sup>&</sup>lt;sup>1</sup>It is of historical interest to note that a program for  $N = 2^{2m+1}$  was written by Gentleman and Sande [47] in 1966, where they claimed a doubling of efficiency by this approach. However, Singleton observed in [83] that when computing with all the data stored in memory, a good radix-2 program was nearly as efficient as a radix-4 plus one step of radix-2 program and was simpler.

Fourier transform of a complex time series:

$$
X_r = \sum_{\ell=0}^{N-1} x_{\ell} \omega_N^{r\ell}, \quad r = 0, 1, ..., N-1,
$$
  
\n(12.1) 
$$
= \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \omega_N^{r(2k)} + \sum_{k=0}^{\frac{N}{4}-1} x_{4k+1} \omega_N^{r(4k+1)} + \sum_{k=0}^{\frac{N}{4}-1} x_{4k+3} \omega_N^{r(4k+3)}
$$

$$
= \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \omega_N^{r(2k)} + \omega_N^r \sum_{k=0}^{\frac{N}{4}-1} x_{4k+1} \omega_N^{r(4k)} + \omega_N^{3r} \sum_{k=0}^{\frac{N}{4}-1} x_{4k+3} \omega_N^{r(4k)}.
$$

By decimating the time series into three sets, namely the set  $\{y_k | y_k = x_{2k}, 0 \le k \le k\}$ *N/*2 − 1}, the set { $z_k|z_k = x_{4k+1}$ ,  $0 \le k \le N/4 - 1$ }, and the set { $h_k|h_k = x_{4k+3}$ ,  $0 \le k \le N/4 - 1$ }  $k \leq N/4 - 1$ , the three subproblems are defined after the appropriate twiddle factors  $\omega_{\frac{N}{2}} = \omega_N^2$  and  $\omega_{\frac{N}{4}} = \omega_N^4$  are identified.

$$
(12.2)
$$

$$
Y_r = \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \omega_N^{r(2k)} = \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \left(\omega_N^2\right)^{rk} = \sum_{k=0}^{\frac{N}{2}-1} y_k \omega_{\frac{N}{2}}^{rk}, \quad r = 0, 1, \dots, N/2 - 1.
$$
\n
$$
(12.3)
$$
\n
$$
Z_r = \sum_{k=0}^{\frac{N}{2}-1} x_{4k+1} \omega_N^{r(4k)} = \sum_{k=0}^{\frac{N}{4}-1} x_{4k+1} \left(\omega_N^4\right)^{rk} = \sum_{k=0}^{\frac{N}{4}-1} z_k \omega_{\frac{N}{4}}^{rk}, \quad r = 0, 1, \dots, N/4 - 1.
$$
\n
$$
(12.4)
$$
\n
$$
H_r = \sum_{k=0}^{\frac{N}{2}-1} x_{4k+3} \omega_N^{r(4k)} = \sum_{k=0}^{\frac{N}{4}-1} x_{4k+3} \left(\omega_N^4\right)^{rk} = \sum_{k=0}^{\frac{N}{4}-1} h_k \omega_{\frac{N}{4}}^{rk}, \quad r = 0, 1, \dots, N/4 - 1.
$$

After these three subproblems are each (recursively) solved by the split-radix algorithm, the solution to the original problem of size *N* can be obtained according to (12.1) for  $r = 0, 1, \ldots, N - 1$ . Because  $Y_{r+k\frac{N}{2}} = Y_r$  for  $0 \le r \le N/2 - 1$ ,  $Z_{r+k\frac{N}{4}} = Z_r$  for  $0 \le r \le N/4 - 1$ , and  $H_{r+k\frac{N}{4}} = H_r$  for  $0 \le r \le N/4 - 1$ , equation (12.1) can be rewritten in terms of  $Y_r$ ,  $Y_{r+\frac{N}{4}}$ ,  $Z_r$ , and  $H_r$  for  $0 \le r \le N/4 - 1$  as shown below.

(12.5)  
\n
$$
X_r = Y_r + \omega_N^r Z_r + \omega_N^{3r} H_r
$$
\n
$$
= Y_r + (\omega_N^r Z_r + \omega_N^{3r} H_r), \quad 0 \le r \le \frac{N}{4} - 1,
$$

(12.6) 
$$
X_{r+\frac{N}{4}} = Y_{r+\frac{N}{4}} + \omega_N^{r+\frac{N}{4}} Z_r + \omega_N^{3(r+\frac{N}{4})} H_r
$$

$$
= Y_{r+\frac{N}{4}} - j \left( \omega_N^r Z_r - \omega_N^{3r} H_r \right), \quad 0 \le r \le \frac{N}{4} -
$$

(12.7)  

$$
X_{r+\frac{N}{2}} = Y_r + \omega_N^{r+\frac{N}{2}} Z_r + \omega_N^{3(r+\frac{N}{2})} H_r
$$

$$
= Y_r - (\omega_N^r Z_r + \omega_N^{3r} H_r), \quad 0 \le r \le \frac{N}{4} - 1,
$$

(12.8) 
$$
X_{r+\frac{3N}{4}} = Y_{r+\frac{N}{4}} + \omega_N^{r+\frac{3N}{4}} Z_r + \omega_N^{3(r+\frac{3N}{4})} H_r
$$

$$
= Y_{r+\frac{N}{4}} + j \left( \omega_N^r Z_r - \omega_N^{3r} H_r \right), \quad 0 \le r \le \frac{N}{4} - 1.
$$

The computation represented by  $(12.5)$ ,  $(12.6)$ ,  $(12.7)$ , and  $(12.8)$  is referred to as an unsymmetric DIT butterfly computation in the literature as shown in Figure 12.1.

<sup>4</sup> <sup>−</sup> <sup>1</sup>*,*



#### **12.2.1 Analyzing the arithmetic cost**

To determine the arithmetic cost of the split-radix DIT FFT algorithm, observe that  $\omega_N^r Z_r$  and  $\omega_N^{3r} H_r$  must be computed before the two partial sums can be formed. Since these two subproblems are each of size  $N/4$ ,  $N/2$  complex multiplications and  $N/2$  complex additions are required in order to obtain the partial sums. Among the *N/*2 complex multiplications, there are four special cases which were already identified in the earlier discussion of the radix-4 algorithm: they are two cases of multiplication by 1 and two cases of multiplication by an odd power of  $\omega_8$ . Recall that the former two cases are trivial, and only four real operations rather than six would be used in the latter two cases. Thus,  $(3+3)\times(N/2-4)+4\times2+2\times(N/2)=4N-16$  nontrivial real operations are performed in the first stage of butterfly computation. In the second stage of the butterfly computation, only *N* complex additions or 2*N* real operations are required. The total cost for a single split-radix step thus involves 6*N* −16 nontrivial real operations (flops).

To set up the recurrence equation, the boundary conditions for both *N* = 4 and  $N = 2$  are needed; when the size of a subproblem is reduced to 8, the three subsequent subproblems are of sizes 4, 2 and 2. As noted earlier,  $T(4) = 16$  flops, and  $T(2) = 4$ flops. The cost of the split-radix FFT algorithm (in terms of nontrivial flops) can be represented by the following recurrence:

(12.9) 
$$
T(N) = \begin{cases} T(\frac{N}{2}) + 2T(\frac{N}{4}) + 6N - 16 & \text{if } N = 4^n > 4, \\ 16 & \text{if } N = 4, \\ 4 & \text{if } N = 2. \end{cases}
$$

Solving (12.9) (see Appendix B), one obtains the solution

(12.10) 
$$
T(N) = 4N \log_2 N - 6N + 8.
$$

#### **12.3 The Split-Radix DIF FFTs**

A split-radix DIF FFT algorithm can be derived by recursively applying both radix-2 and radix-4 DIF FFT algorithm to solve each subproblem resulting from decimating the output frequency series in a similar fashion. That is, the frequency series is recursively decimated into three subsets, i.e., the set denoted by  $Y_k = X_{2k}$  for  $0 \le k \le N/2-1$ , the set denoted by  $Z_k = X_{4k+1}$  for  $0 \le k \le N/4 - 1$ , and the set denoted by  $H_k = X_{4k+3}$ for  $0 \leq k \leq N/4 - 1$  as shown below. The derivation begins with the discrete Fourier transform defined by (3.1). Using the results developed earlier for the radix-2 DIF algorithm in (3.11), one obtains

(12.11)  
\n
$$
X_r = \sum_{\ell=0}^{\frac{N}{2}-1} x_{\ell} \omega_N^{r\ell} + \sum_{\ell=\frac{N}{2}}^{N-1} x_{\ell} \omega_N^{r\ell}
$$
\n
$$
= \sum_{\ell=0}^{\frac{N}{2}-1} \left( x_{\ell} + x_{\ell+\frac{N}{2}} \omega_N^{r\frac{N}{2}} \right) \omega_N^{r\ell}, \quad r = 0, 1, ..., N-1.
$$

By letting  $Y_k = X_{2k}$ ,  $y_\ell = x_\ell + x_{\ell + \frac{N}{2}}$ , one subproblem of half the size is defined by

(12.12)  

$$
Y_k = X_{2k} = \sum_{\ell=0}^{\frac{N}{2}-1} \left( x_{\ell} + x_{\ell + \frac{N}{2}} \right) \omega_{\frac{N}{2}}^{k\ell}
$$

$$
= \sum_{\ell=0}^{\frac{N}{2}-1} y_{\ell} \omega_{\frac{N}{2}}^{k\ell}, \quad k = 0, 1, ..., N/2 - 1.
$$

To construct the other two subproblems of size  $N/4$ , begin with the DFT definition in (3.1) and use the results developed earlier for the DIF radix-4 algorithm in (11.22).

(12.13) 
$$
X_r = \sum_{\ell=0}^{N-1} x_{\ell} \omega_N^{r\ell}, \quad r = 0, 1, ..., N-1,
$$

$$
= \sum_{\ell=0}^{\frac{N}{4}-1} \left( x_{\ell} + x_{\ell + \frac{N}{4}} \omega_4^r + x_{\ell + \frac{N}{2}} \omega_4^{2r} + x_{\ell + \frac{3N}{4}} \omega_4^{3r} \right) \omega_N^{r\ell}.
$$

By substituting  $r = 4k + 1$  and  $r = 4k + 3$ , one again obtains

$$
(12.14)
$$

$$
Z_k = X_{4k+1} = \sum_{\ell=0}^{\frac{N}{4}-1} \left( x_{\ell} + x_{\ell+\frac{N}{4}} \omega_4^{4k+1} + x_{\ell+\frac{N}{2}} \omega_4^{2(4k+1)} + x_{\ell+\frac{3N}{4}} \omega_4^{3(4k+1)} \right) \omega_N^{(4k+1)\ell}
$$
  
= 
$$
\sum_{\ell=0}^{\frac{N}{4}-1} \left( \left( x_{\ell} - x_{\ell+\frac{N}{2}} \right) - j \left( x_{\ell+\frac{N}{4}} - x_{\ell+\frac{3N}{4}} \right) \right) \omega_N^{\ell} \omega_{\frac{N}{4}}^{k\ell}
$$
  
= 
$$
\sum_{\ell=0}^{\frac{N}{4}-1} z_{\ell} \omega_{\frac{N}{4}}^{k\ell}, \quad k = 0, 1, ..., N/4 - 1.
$$

(12.15)

$$
H_k = X_{4k+3} = \sum_{\ell=0}^{\frac{N}{4}-1} \left( x_{\ell} + x_{\ell+\frac{N}{4}} \omega_4^{4k+3} + x_{\ell+\frac{N}{2}} \omega_4^{2(4k+3)} + x_{\ell+\frac{3N}{4}} \omega_4^{3(4k+3)} \right) \omega_N^{(4k+3)\ell}
$$
  
= 
$$
\sum_{\ell=0}^{\frac{N}{4}-1} \left( \left( x_{\ell} - x_{\ell+\frac{N}{2}} \right) + j \left( x_{\ell+\frac{N}{4}} - x_{\ell+\frac{3N}{4}} \right) \right) \omega_N^{3\ell} \omega_{\frac{N}{4}}^{k\ell}
$$
  
= 
$$
\sum_{\ell=0}^{\frac{N}{4}-1} h_{\ell} \omega_{\frac{N}{4}}^{k\ell}, \quad k = 0, 1, ..., N/4 - 1.
$$

To form these three subproblems using two stages of unsymmetric butterfly computation, the computation of the partial sums is again rearranged to facilitate the butterfly computation.

(12.16) 
$$
y_{\ell} = \left(x_{\ell} + x_{\ell + \frac{N}{2}}\right), \quad 0 \leq \ell \leq \frac{N}{4} - 1.
$$

(12.17) 
$$
y_{\ell + \frac{N}{4}} = \left(x_{\ell + \frac{N}{4}} + x_{\ell + \frac{3N}{4}}\right), \quad 0 \le \ell \le \frac{N}{4} - 1.
$$

(12.18) 
$$
z_{\ell} = \left( \left( x_{\ell} - x_{\ell + \frac{N}{2}} \right) - j \left( x_{\ell + \frac{N}{4}} - x_{\ell + \frac{3N}{4}} \right) \right) \omega_{N}^{\ell}, \quad \leq \ell \leq \frac{N}{4} - 1.
$$

(12.19) 
$$
h_{\ell} = \left(j\left(x_{\ell + \frac{N}{4}} - x_{\ell + \frac{3N}{4}}\right) + \left(x_{\ell} - x_{\ell + \frac{N}{2}}\right)\right)\omega_N^{3\ell}, \quad 0 \leq \ell \leq \frac{N}{4} - 1.
$$

The computation represented by (12.16), (12.17), (12.18), and (12.19) again yields an unsymmetric DIF butterfly computation as depicted in [Figure](#page-5-0) 12.2.

<span id="page-5-0"></span>

#### **12.4 Notes and References**

The split-radix FFT was originally developed by Duhamel and Hollmann [39] in 1984, and it was subsequently extended and implemented for complex, real and real-symmetric data by Duhamel in [38]. In 1986, Sorensen, Heideman, and Burrus presented an indexing scheme which efficiently implemented the Duhamel-Hollmann split-radix FFT [86]. Both DIF and DIT Fortran programs were presented in [86]. The history of the ideas on the fast Fourier transforms from Gauss to the split-radix algorithm is presented in [41].