14

The Fast Fourier Transform

14.4 Other Common FFT Algorithms

In the previous section we saw the radix-2 DIT algorithm, also known as the Cooley-Tukey algorithm. Here we present a few more FFT algorithms, radix-2 DIF, the prime factor algorithm (PFA), non-power-of-two radixes, split-radix, etc. Although different in details, there is a strong family resemblance between all these algorithms. All reduce the N^2 complexity of the straightforward DFT to $N \log N$ by restructuring the computation, all exploit symmetries of the W_N^{nk}, and all rely on the length of the signal N being highly composite.

First let us consider the decimation in frequency (DIF) FFT algorithm. The algebraic derivation follows the same philosophy as that of the DIT. We start by partitioning the time sequence, into left and right subsequences

$$\begin{aligned} x_n^L &= x_n \qquad\qquad \text{for } n = 0, 1, \ldots \frac{N}{2} - 1 \\ x_n^R &= x_{n+\frac{N}{2}} \end{aligned}$$

and splitting the DFT sum into two sums.

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x_n \, W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x_n \, W_N^{nk} \qquad (14.5) \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_n^L \, W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x_n^R \, W_N^{nk} W_N^{\frac{Nk}{2}} \end{aligned}$$

Now let's compare the even and odd X_k (decimation in the frequency domain). Using the fact that $W_N^2 = W_{\frac{N}{2}}$

$$\begin{aligned} X_{2k} &= \sum_{n=0}^{\frac{N}{2}-1} (x_n^L \, W_{\frac{N}{2}}^{nk} + x_n^R \, W_{\frac{N}{2}}^{nk} \, W_N^{Nk}) \\ X_{2k+1} &= \sum_{n=0}^{\frac{N}{2}-1} (x_n^L \, W_{\frac{N}{2}}^{nk} + x_n^R \, W_{\frac{N}{2}}^{nk} \, W_N^{Nk} \, W_N^{\frac{N}{2}}) \; W_N^n \end{aligned}$$

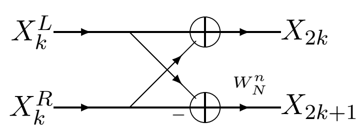
and then substituting $W_N^{kN} = 1$ and $W_N^{\frac{N}{2}} = -1$ we find the desired expressions.

$$\begin{aligned} X_{2k} &= \sum_{n=0}^{\frac{N}{2}-1} (x_n^L + x_n^R) \, W_{\frac{N}{2}}^{nk} = (X_k^L + X_k^R) \\ X_{2k+1} &= \sum_{n=0}^{\frac{N}{2}-1} (x_n^L - x_n^R) \, W_{\frac{N}{2}}^{nk} \, W_N^n \end{aligned}$$

Unlike the DIT case, the odd frequency components here can not be elegantly separated into DFTs of half-length subsequences due to the dependence of W_N^n on n; but this does not rule out recursive computation of the DFT. We need only multiply all odd outputs of the previous stage by W_N^n before continuing.

Just as for the DIT we found similarity between Fourier components in different frequency partitions, for DIF we find similarity between frequency components that are related by decimation.

It is thus evident that the DIF butterfly can be drawn



which is different from the DIT butterfly, mainly in the position of the twiddle factor.

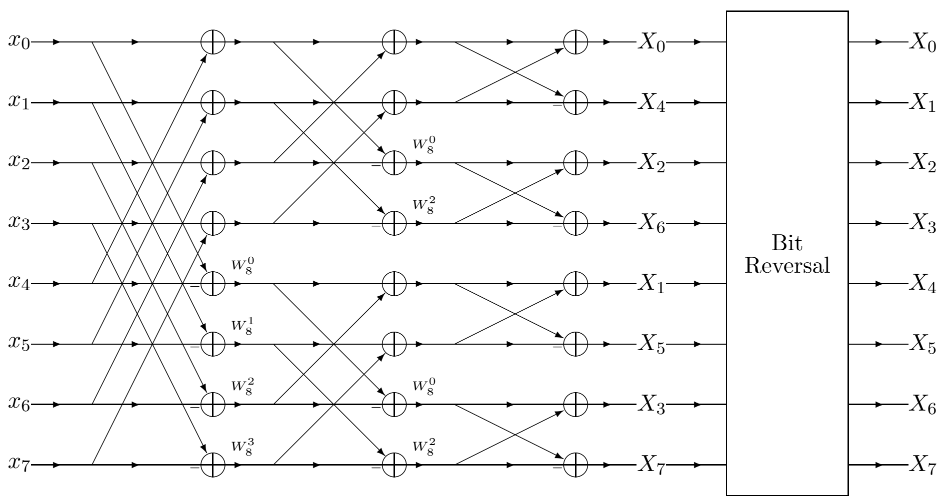


Figure 14.6: Full eight-point radix-2 DIF DFT, with bit reversal on outputs.

We leave as an exercise to complete the decomposition, mentioning that once again bit reversal is required, only this time it is the outputs that need to be bit reversed. The final eight-point DIF DFT is depicted in Figure 14.6.