The Fast Fourier Transform - Part 2

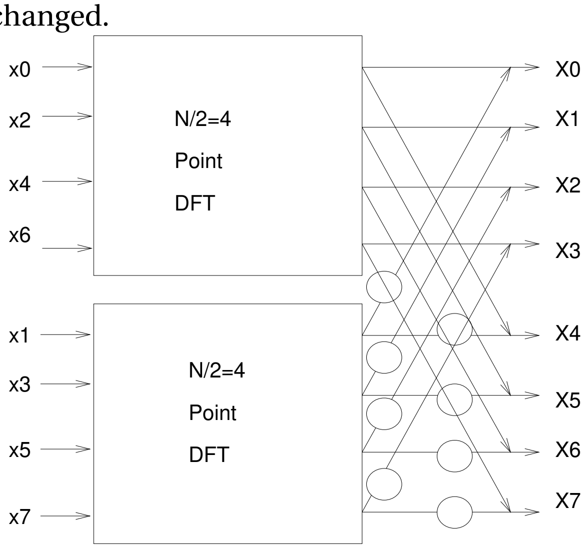
DIT FFT Signal Flow Graph

The equation for the first decimation in time (DIT) step:

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

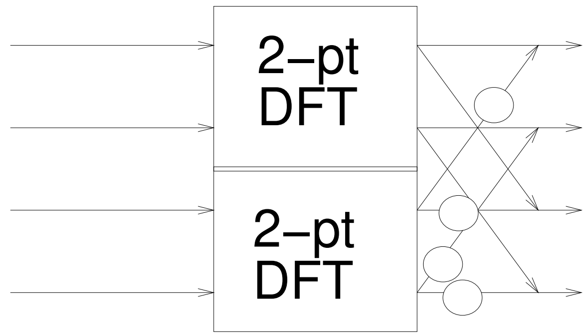
shows that each $X(k)$ is the sum of the results of two $N/2 = 4$-point DFTs. The results of the DFT of the odd sequence are scaled by W_N^k before being added to the results of the DFT of the even sequence.

Using an 8-point DFT as an example we can draw the signal flow graph for the first step in the process. We have drawn the 4-point DFT as a box. Note that the order of the inputs has changed.



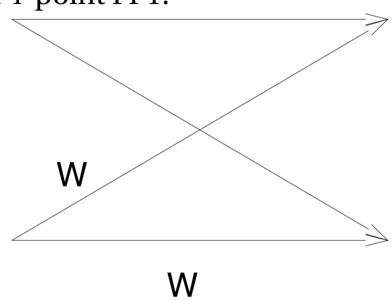
Exercise: Each of the circles represents a power of W_N. Fill in the subscripts and power of W in the diagram above.

The decimation process can be carried out for each of the two 4-point DFTs. Each of these will be divided into the sum of two 2-point DFTs. Again, the order of the inputs will change so that the even numbered (relative to the 4-point DFTs) samples go into one 2-point DFT and the odd numbered samples into the other.



Exercise: Assume the diagram above is computing the lower of the two 4-point DFTs in the previous diagram. What will be the four inputs on the left? Each of the circles represents a power of W factor. Fill in the subscripts and power of W for each circle in the diagram above.

The final step is to fill in the two radix-2 FFTs for each 4-point FFT:



This basic signal flow graph is known as a *butterfly* and is the basis of the FFT. Examination a complete FFT signal flow graph shows that whole FFT is built up out of these basic butterflies.

Computational Complexity

After the decimation is complete down to radix-2 DFTs there will be $\log_2 N$ steps each with $N/2$ butterflies. Each butterfly involves two additions and two multiplications. So there will be a total of $N \log_2 N$ multiplications and the same number of additions required to compute the complete FFT. For large N this represents a significant reduction in computation compared to the N^2 operations required for the DFT.

Exercise: What is the reduction in the number of computations for $N = 16$? For $N = 1024$?

In-Place Computation

The computation of the FFT proceeds across the signal flow graph from left to right. At each of the $\log_2 N$ stages we compute the N values for the next stage using the N values from the previous stage. We could alternate using two arrays of N elements to hold the values during the computations. However, it's possible to use only a single N-element array. This is because each butterfly only depends on the same two elements that are used in computing these same elements' new values (we do need to use one temporary variable while computing the result of each butterfly).

Reordered Input

At each decimation step we need to rearrange the input sequences into even- and odd-indexed sequences. If we consider the indexes of the original input samples as binary numbers we see that the first decimation step moved all samples whose least significant bit was zero (i.e. the even numbered samples, 0, 2, 4, 6) into the even sequence at the top of the graph. The next decimation step then rearranged these four samples according to their even/odd position in this new set of four. This is the same as moving all of those whose second least significant bit of the sample index is zero to the top (e.g. 0, and 4 have their second least significant bit equal to zero). Eventually we see that position of a sample in the input will be determined by the bits in its sample index read from the least-significant to most-significant bit. This is called *reordered, bit-reversed indexed* or *shuffled* order.