9.2: The Prime Factor Algorithm

If the DFT is calculated directly using the equation in 9.1: Introduction, the algorithm is called a prime factor algorithm and was discussed in Winograd's Short DFT Algorithms. When the short DFT's are calculated by the very efficient algorithms of Winograd discussed in Factoring the Signal Processing Operators, the PFA becomes a very powerful method that is as fast or faster than the best Cooley-Tukey FFT's.

A flow graph is not as helpful with the PFA as it was with the Cooley-Tukey FFT, however, the following representation in Fig. 9.2.1 below, which combines the figures in The Index Map and Winograd Fourier Transform Algorithm (WFTA) gives a good picture of the algorithm with the example of Multidimensional Index Mapping.

If \( N \) is factored into three factors, the DFT of the equation would have three nested summations and would be a three-dimensional DFT. This principle extends to any number of factors; however, recall that the Type-1 map requires that all the factors be relatively prime. A very simple three-loop indexing scheme has been developed which gives a compact, efficient PFA program for any number of factors. The basic program structure is illustrated below with the short DFT's.
As in the Cooley-Tukey program, the DO 10 loop steps through the M stages (factors of N) and the DO 20 loop calculates the N/N1 length-N1 DFT's. The input index map of the equation is implemented in the DO 30 loop and the statement just before label 20. In the PFA, each stage or factor requires a separately programmed module or butterfly. This lengthens the PFA program but an efficient Cooley-Tukey program will also require three or more butterflies.

Because the PFA is calculated in-place using the input index map, the output is scrambled. There are five approaches to dealing with this scrambled output. First, there are some applications where the output does not have to be unscrambled as in the case of high-speed convolution. Second, an unscrambler can be added after the PFA to give the output in correct order just as the bit-reversed-counter is used for the Cooley-Tukey FFT. The third method does the unscrambling in the modules while they are being calculated. This is probably the fastest method but the program must be written for a specific length. A fourth method is similar and achieves the unscrambling by choosing the multiplier constants in the modules properly. The fifth method uses a separate indexing method for the input and output of each module.

Contributor

- ContribEEBurrus