polynomial," *Trans. Inst. Elec. Commun. Eng. (in Japanese)*, vol. J68-A, pp. 173–180, 1985.
[8] Z. Wang, "Pruning the fast discrete cosine transform," *IEEE Trans. Commun.*, vol. 39, pp. 640–643, May 1991.
[9] S. B. Narayanan and K. M. M. Prabhu, "Fast Hartley transform pruning," *IEEE Trans. Signal Processing*, vol. 39, pp. 230–233, Jan. 1991.

Index Mappings for the Fast Fourier Transform

James C. Schatzman

Abstract— **General analysis shows which linear index maps for the multidimensional PFA avoid twiddle factors. For any input map of this class, there is a unique output map which omits twiddle factors and modifications of the underlying DFT subblocks. With subblock modifications, all factorizations can be done in-place and in-order.**

I. Introduction

The prime factor algorithm fast Fourier transform is based on the reduction of a 1-D discrete fourier tranform to a multidimensional DFT. Many have studied the underlying index mapping schemes: Burrus (1977), Burrus and Eschenbacher (1981), Rothweiller (1982), Burrus and Parks (1985), Temperton (1985), Siu and Wong (1989), Lun *et al.* (1989), Chan and Ho (1991), Johnsson *et al.* (1991), and Temperton (1992). Temperton (1985) and (1992) and Burrus and Eschenbacher (1981) describe in-place in-order FFT algorithms. Steidl and Tasche (1989) and Lun and Siu (1993) give in-place and in-order algorithms for special factorizations. Linzer and Feig (1993) show that incorporating direct permutation steps into the first or last stage of the FFT does not degrade performance. Burrus (1977) and Lun and Siu (1993) give the clearest and most useful analyses of linear index maps. This work builds on these results to obtain a comprehensive theory.

II. Two-Dimensional Formulation

The discrete Fourier transform of length N may be written

$$c_k = \sum_{n=0}^{N-1} e^{-\imath\frac{2\pi}{N}kn} x_n; 0 \le k \le N-1 \tag{1}$$

and the goal is to simplify the computation by reordering the input and output and by exploiting the redundancy of the transform. For two factors ($N = p_1p_2$), the input index n and the output index k are generally mapped to two dimensions according to a certain linear form modulo N, as follows:

$$\begin{aligned} n &= (N_1n_1 + N_2n_2) \bmod N; 0 \le n_1 \le p_1-1; 0 \le n_2 \le p_2-1 \\ k &= (K_1k_1 + K_2k_2) \bmod N; 0 \le k_1 \le p_1-1; 0 \le k_2 \le p_2-1 \end{aligned} \tag{2}$$

Manuscript received August 29, 1994; revised May 31, 1995. The associate editor coordinating the review of this work and approving it for publication was Dr. K. J. Ray Liu.
J. C. Schatzman is with the University of Wyoming, Laramie, WY 82071 USA (e-mail: jcs@uwyo.edu).
Publisher Item Identifier S 1053-587X(96)02384-5.

TABLE I
Summation Kernel for Combinations of Popular Maps. Here $w_q \equiv e^{-\imath 2\pi/q}$, $p_1' = (p_1^{-1} \bmod p_2)$ and $p_2' = (p_2^{-1} \bmod p_1)$

Input	Output Map: Trivial	Digit-Reversing	Ruritanian	CRT
Output	$K_1=1$, $K_2=p_1$	$K_1=p_2$, $K_2=1$	$K_1=p_2$, $K_2=p_1$	$K_1=p_2'p_2$, $K_2=p_1'p_1$
Trivial $N_1=1$ $N_2=p_1$	$w_{p_2}^{p_1k_2n_2}w_N^{p_1k_2n_1}$ $w_N^{p_1k_1n_2}w_{p_1}^{p_2'k_1n_1}$	$w_{p_2}^{k_2n_2}w_N^{k_2n_1}$ $w_{p_1}^{k_1n_1}$	$w_{p_2}^{p_1k_2n_2}w_N^{p_1k_2n_1}$ $w_{p_1}^{k_2n_2}$	$w_{p_2}^{k_2n_2}w_N^{k_2n_1}$ $w_{p_1}^{p_2'k_1n_1}$
Digit-Reversing $N_1=p_2$ $N_2=1$	$w_{p_2}^{k_2n_2}w_N^{k_1n_2}$ $w_{p_1}^{k_1n_1}$	$w_{p_2}^{p_1'k_2n_2}w_N^{p_2k_1n_2}$ $w_N^{p_2k_2n_1}w_{p_1}^{p_2k_1n_1}$	$w_{p_2}^{k_2n_2}w_N^{p_2k_1n_2}$ $w_{p_1}^{p_2k_1n_1}$	$w_{p_2}^{p_1'k_2n_2}w_N^{k_1n_2}$ $w_{p_1}^{k_1n_1}$
Ruritanian $N_1=p_2$ $N_2=p_1$	$w_{p_2}^{p_1k_2n_2}w_N^{p_1k_1n_2}$ $w_{p_1}^{k_1n_1}$	$w_{p_2}^{k_2n_2}w_N^{p_2k_2n_1}$ $w_{p_1}^{p_2k_1n_1}$	$w_{p_2}^{p_1k_2n_2}w_{p_1}^{p_2k_1n_1}$	$w_{p_2}^{k_2n_2}w_{p_1}^{k_1n_1}$
CRT $N_1=p_2'p_2$ $N_2=p_1'p_1$	$w_{p_2}^{k_2n_2}w_N^{k_1n_2}$ $w_{p_1}^{p_2'k_1n_1}$	$w_{p_2}^{p_1'k_2n_2}w_N^{k_2n_1}$ $w_{p_1}^{k_1n_1}$	$w_{p_2}^{k_2n_2}w_{p_1}^{k_1n_1}$	$w_{p_2}^{p_1'k_2n_2}w_{p_1}^{p_2'k_1n_1}$

where $N = p_1p_2$ for arbitrary natural numbers p_1 and p_2. For the $n \leftrightarrow (n_1, n_2)$ map to be one-to-one requires (Burrus (1977)) the following.

Case A: $\gcd(p_1, p_2) = 1$*:*

A1) $N_1 = a_1p_2$, $N_2 \neq a_2p_1$, $\gcd(a_1, p_1) = 1$, and $\gcd(N_2, p_2) = 1$; or

A2) $N_1 \neq a_1p_2$, $N_2 = a_2p_1$, $\gcd(N_1, p_1) = 1$, and $\gcd(a_2, p_2) = 1$; or

A3) $N_1 = a_1p_2$, $N_2 = a_2p_1$, $\gcd(a_1, p_1) = 1$, and $\gcd(a_2, p_2) = 1$

Case B: $\gcd(p_1, p_2) > 1$*:*

B1) $N_1 = a_1p_2$, $N_2 \neq a_2p_1$, $\gcd(a_1, p_1) = 1$, and $\gcd(N_2, p_2) = 1$

B2) $N_1 \neq a_1p_2$, $N_2 = a_2p_1$, $\gcd(N_1, p_1) = 1$, and $\gcd(a_2, p_2) = 1$.

where a_1 and a_2 are integers. Particular maps of importance are $N_1 = 1$, $N_2 = p_1$ ("Trivial map," Cases A2/B2); $N_1 = p_2$, $N_2 = 1$ ("Digit-reversing map," Cases A1/B1); $a_1 = 1$, $a_2 = 1$, ("Ruritanian map," Case A3); $a_1 = p_2^{-1} \bmod p_1$, $a_2 = p_1^{-1} \bmod p_2$, ("Chinese Remainder Theorem (CRT) map," Case A3). Mapping x_n to $\hat{x}_{n_1,n_2}$ and c_k to $\hat{c}_{k_1,k_2}$, (1) becomes

$$\hat{c}_{k_1,k_2} = \sum_{n_2=0}^{p_2-1}\sum_{n_1=0}^{p_1-1} e^{-\imath\frac{2\pi}{N}(K_1N_1k_1n_1 + K_2N_1k_2n_1 + K_1N_2k_1n_2 + K_2N_2k_2n_2)}\hat{x}_{n_1,n_2}. \tag{3}$$

Considering all possible combinations of input and output maps from these special cases, we obtain the results in Table I.

Using matrix notation, $\vec{c} = F\vec{x}$ becomes $\vec{c} = P_O^{-1}EP_I\vec{x}$ where P_I and P_O are the input/output map permutation matrices and E is the matrix from the Table II. In Table II, T is the N by N diagonal

TABLE II
MAPPED DISCRETE FOURIER TRANSFORM WRITTEN IN MATRIX NOTATION FOR COMBINATIONS OF POPULAR MAPS. $\otimes$ DENOTES THE KRONECKER MATRIX PRODUCT

Input / Output	Output Map: Trivial $K_1=1$, $K_2=p_1$	Digit-Reversing $K_1=p_2$, $K_2=1$	Ruritanian $K_1=p_2$, $K_2=p_1$	CRT $K_1=p_2'p_2$, $K_2=p_1'p_1$
Trivial $N_1=1$ $N_2=p_1$	*	$(I_{p_2}\otimes F_{p_1})T$ $(F_{p_2}\otimes I_{p_1})$	$(I_{p_2}\otimes F_{p_1})T^{p_1}$ $(F_{p_2}^{[p_1]}\otimes I_{p_1})$	$(I_{p_2}\otimes F_{p_1}^{[p_2']})T$ $(F_{p_2}\otimes I_{p_1})$
Digit-Reversing $N_1=p_2$ $N_2=1$	$(F_{p_2}\otimes I_{p_1})T$ $(I_{p_2}\otimes F_{p_1})$	*	$(F_{p_2}\otimes I_{p_1})T^{p_2}$ $(I_{p_2}\otimes F_{p_1}^{[p_2]})$	$(F_{p_2}^{[p_1']}\otimes I_{p_1})T$ $(I_{p_2}\otimes F_{p_1})$
Ruritanian $N_1=p_2$ $N_2=p_1$	$(F_{p_2}^{[p_1]}\otimes I_{p_1})T^{p_1}$ $(I_{p_2}\otimes F_{p_1})$	$(I_{p_2}\otimes F_{p_1}^{[p_2]})T^{p_2}$ $(F_{p_2}\otimes I_{p_1})$	$F_{p_2}^{[p_1]}\otimes F_{p_1}^{[p_2]}$	$F_{p_2}\otimes F_{p_1}$
CRT $N_1=p_2'p_2$ $N_2=p_1'p_1$	$(F_{p_2}\otimes I_{p_1})T$ $(I_{p_2}\otimes F_{p_1}^{[p_2']})$	$(I_{p_2}\otimes F_{p_1})T$ $(F_{p_2}^{[p_1']}\otimes I_{p_1})$	$F_{p_2}\otimes F_{p_1}$	$F_{p_2}^{[p_1']}\otimes F_{p_1}^{[p_2']}$

matrix of twiddle factors

$$T = \mathbf{diag}\left(\Lambda_{p_1}^{\frac{0}{p_2}}, \Lambda_{p_1}^{\frac{1}{p_2}}, \ldots, \Lambda_{p_1}^{\frac{p_2-1}{p_2}}\right) \quad ;$$
$$\Lambda_q = \mathbf{diag}(e^{-\imath\frac{2\pi}{q}j}; j = 0, 1, 2, \ldots, q-1). \quad (4)$$

$F_p^{[r]}$ is the length p "rotated" DFT formed by taking the rth power of each element of F_p. The starred entries are those than cannot usefully be reduced to 2-D computations. From Table II we see the historically important result that combining Ruritanian and CRT maps leads to true 2-D DFT's ($F_{p_2}\otimes F_{p_1}$) that require no twiddle factors (as in Temperton (1985)). That there are other maps that achieve this result is seen below.

The double use of the Ruritanian or CRT maps leads to true 2-D transformations that require no twiddle factors, but in this case the underlying DFT operator is "rotated." (It is impossible for $p_1' = p_2' = 1$ unless either of p_1 or p_2 is unity, uninteresting cases.) Temperton (1992) claims that if an algorithm exists for computing the DFT with F_p for any p, prime or composite, possibly using the FFT, the corresponding matrix-vector product with $F_p^{[r]}$ may be computed at the same cost as for F_p by modifying the algorithm in two ways: i) Change the multiplier constants in each submodule, and ii) raise all twiddle factors (if any) to the rth power. However, this argument ignores the fact that many of the "multiplier constants" in DFT modules are used implicitly or are modified. Therefore, a nontrivial coding effort is required to convert an F_p module to $F_p^{[q]}$. However, there are no more than $p-1$ modules $F_p^{[q]}$ of interest, so the coding effort is modest provided that p is small. Each can be implemented as efficiently as F_p, and if necessary a test and branch can select the appropriate submodule. Thus, with nearly zero additional computational cost but some additional coding and maintenance cost over F_p, $F_p^{[q]}$ can be implemented.

The Ruritanian and CRT maps are not unique in eliminating twiddle factors, as pointed out in Lun and Siu (1993). We approach this question in a more general framework. A mapping that eliminates twiddle factors must belong to case A3 (easily shown). If the n and k maps both belong to Case A3, with corresponding integers a_1, a_2 for $n \leftrightarrow (n_1, n_2)$ and b_1, b_2 for $k \leftrightarrow (k_1, k_2)$, we obtain

$$\hat{c}_{k_1,k_2} = \sum_{n_2=0}^{p_2-1}\sum_{n_1=0}^{p_1-1} e^{-\imath\frac{2\pi}{N}(a_2b_2p_1^2k_2n_2)} e^{-\imath\frac{2\pi}{N}(a_1b_1p_2^2k_1n_1)}\hat{x}_{n_1,n_2}$$
$$= \sum_{n_2=0}^{p_2-1}\sum_{n_1=0}^{p_1-1} e^{-\imath\frac{2\pi}{p_2}(a_2b_2p_1k_2n_2)} e^{-\imath\frac{2\pi}{p_1}(a_1b_1p_2k_1n_1)}\hat{x}_{n_1,n_2} \quad (5)$$

which amounts to the 2-D transformation $F_{p_2}^{[a_2b_2p_1]}\otimes F_{p_1}^{[a_1b_1p_2]}$.

The case of identical input and output maps is of special interest. As pointed out by many authors, a straightforward in-place in-order algorithm results. Although the algorithm requires complicated addressing, as Lun and Siu (1993) point out, the addressing can be accomplished by indirect addressing or in hardware "using a circular buffer." Unfortunately, indirect addressing is inefficient on most high-speed general purpose computers because of the increased memory references required. On many processors, therefore, this theoretical property of being "in-place and in-order" may be of little or no advantage; explicit mapping is often preferable. Even so, we note that it is possible for identical input and output maps to give a result similar to the crossed Ruritanian/CRT maps: a direct 2-D DFT with no "rotations." This occurs if and only if $a_1^2p_2 \bmod p_1 = 1$ and $a_2^2p_1 \bmod p_2 = 1$; equivalently, if and only if $N_1^2 \equiv p_2 \bmod p_1$ and $N_2^2 \equiv p_1 \bmod p_2$, or that p_1 and p_2 are quadratic residues of each other. For example, $p_1 = 5$, $p_2 = 4$, $N_1 = 8$, $N_2 = 15$, gives a map that is neither Ruritanian nor CRT, and when used for both input and output maps gives a factorization without "rotations." There are many factorizations for which this can be made to work; Lun and Siu [7] list some.

III. M-DIMENSIONAL FORMULATION

While a multifactor PFA can be implemented recursively with two-factor maps, with some restrictions the factorizations extend to more than two factors. The PFA with M factors involves index mappings $n \leftrightarrow (n_1, \ldots, n_M)$. Consider first the $M = 3$ case. Composing two 2-D maps for factors p_1, p_2, p_3, we obtain

$$n = (N_1n_1 + N_2[(M_1n_2 + M_2n_3) \bmod N/p_1]) \bmod N;$$
$$0 \le n_j \le p_j - 1, j = 1, 2, 3. \quad (6)$$

Provided that the outer map is of case A2, A3 or B2, N_2 is a multiple of p_1 so (6) simplifies to

$$n = (N_1n_1 + N_2M_1n_2 + N_2M_2n_3) \bmod N. \quad (7)$$

We may redefine N_2 and obtain

$$n = (N_1n_1 + N_2n_2 + N_3n_3) \bmod N; 0 \le n_j \le p_j - 1, j = 1, 2, 3. \quad (8)$$

By induction, provided that all but the inner map are of case A2, A3 or B2, the composition of M maps leads to a linear form modulo N, as follows:

$$n = (N_1n_1 + N_2n_2 + \ldots + N_Mn_M) \bmod N \quad (9)$$

where $n = 0, 1, \ldots, N-1$, $n_j = 0, 1, \ldots, p_j - 1$, and $N = p_1p_2\ldots p_M$. It is straightforward to show that there is only one other case that leads to (9), which is when all maps are either trivial or digit-reversing. By induction, composition of any M case A3 maps leads to (9) where also $N_j = a_jN/p_j$. The constraint that the mapping be one-to-one requires that $\gcd(a_j, p_j) = 1$.

Applying maps of the type (9) to input and output, we obtain

$$c_{k_1,\ldots,k_M} = \sum_{n_M=0}^{p_M-1} \cdots \sum_{n_2=0}^{p_2-1} \sum_{n_1=0}^{p_1-1} e^{-\imath \frac{2\pi}{N}(N_1 n_1 + \ldots + N_M n_M)(K_1 k_1 + \ldots + K_M k_M)} x_{n_1,\ldots,n_M}. \tag{10}$$

It is straightforward but somewhat tedious to show that the twiddle factors disappear if and only if $N_j K_l \bmod N = 0$ for all distinct j and l. This holds if and only if all maps are of case A3 ("if" is trivial, "only if" follows by contradiction). When all maps are of case A3, and there are therefore no twiddle factors, we obtain

$$c_{k_1,\ldots,k_M} = \sum_{n_M=0}^{p_M-1} \cdots \sum_{n_2=0}^{p_2-1} \sum_{n_1=0}^{p_1-1} e^{-\imath \frac{2\pi}{N}(a_1 b_1 (N/p_1)^2 n_1 k_1 + \ldots + a_M b_M (N/p_M)^2 n_M k_M)} x_{n_1,\ldots,n_M} \tag{11}$$

or

$$\vec{c} = P_O^{-1}\left(F_{p_1}^{[a_1 b_1 N/p_1]} \otimes F_{p_2}^{[a_2 b_2 N/p_2]} \otimes \ldots \otimes F_{p_M}^{[a_M b_M N/p_M]}\right) P_I \vec{x} \tag{12}$$

which shows that the underlying subblocks F_{p_j} are modified by taking the corresponding powers $a_j b_j N/p_j$ of each element. The subblocks are "rotated" unless $a_j b_j \frac{N}{p_j} \bmod p_j = 1$ for each j. Suppose the input map (i.e., the set of coefficients a_j) is given. Elementary number theory tells us that these equations have no solution unless $\gcd(p_j, N/p_j) = 1$, i.e., that the factors p_j are relatively prime. If the factors are relatively prime, then the solutions are simply $b_j = \left(\frac{a_j N}{p_j}\right)^{-1} \bmod p_j$. Since without loss of generality $1 \le K_j \le N-1$ and $K_j = b_j N/p_j$, also without loss of generality we may assume that $1 \le b_j \le p_j - 1$. Thus, we have obtained the result that for any case A3 input map, there is exactly one linear output map that converts the original DFT to a multidimensional DFT with no twiddle factors and no modifications of the basic subblocks. Using the same A3 map for input and output leads to the same conditions as before for the absense of "rotations", as follows: $(a_j^2 N/p_j) \bmod p_j = 1$ or $N_j^2 \equiv N/p_j \bmod p_j$.

P_O^{-1} may not be representable as a linear form modulo N. In practice, this does not significantly complicate coding. However, some maps having this property are especially attractive. There are no 3-D or higher dimensional A3 maps that have linear inverses. For 2-D general linear maps, only some A3 maps have linear inverse maps. Specifically, for given p_1 and p_2, the A3 map given by $a_1 = p_2^{-1} \bmod p_1$, $a_2 = (z p_1) \bmod p_2$, $z = \lfloor a_1 p_2 / p_1 \rfloor$ is the unique A3 map that has a linear inverse; it is its own inverse. For the same p_1 and p_2, there are as many as $p_2 - 2$ A3 maps that have A2 inverses, satisfying $a_1 = p_2^{-1} \bmod p_1$, $K_1 \bmod p_1 = 1$, $a_2 = \frac{z p_1}{1 + z p_1 K_1} \bmod p_2$, $b_2 = a_2^{-1} \bmod p_2$, $b_2 > 0$. These are necessary and sufficient conditions for the parameters of the forward ($N_1 = a_1 p_2$, $N_2 = a_2 p_1$) and inverse ($N_1 = K_1$, $N_2 = b_2 p_1$) maps. The cases with A3 inverses are particularly simple and efficient to implement, as the maps are 2-cycles. For example, $p_1 = 2$, $p_2 = 3$, $N_1 = 3$, $N_2 = 4$, produces the map

$$\boxed{\begin{matrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{matrix}} \rightarrow \boxed{\begin{matrix} 0 & 4 & 2 \\ 3 & 1 & 5 \end{matrix}} \tag{13}$$

which is obviously its own inverse. PFA implementations that use two-factor two-cycle mapping schemes (recursively) are especially simple and efficient to implement. Such maps are trivially in-place without requiring indirect addressing or special hardware. For $M > 2$, there are no case A3 maps that have inverses representable as linear forms modulo N, so recursive algorithms are necessary to obtain this benefit.

IV. Conclusion

For any factorization, there are sets of corresponding input and output linear modulo N maps that decompose a 1-D DFT of length N to a multidimensional DFT, possibly modified. It is the class A3 maps that avoid twiddle factors, and for every A3 input map there is exactly one corresponding output map (also A3) that does not require modifying the subblocks. For general purpose processors, the in-place/in-order property may be less important than the speed with which explicit permutations may be performed. The author recommends two-cycle recursive implementations that require only swaps, and hence are trivially in-place but not in-order. Subblock modifications are required, however.

Rigorous mathematical proofs are available for all claims in this paper. Interested parties may request copies from the author.

References

[1] C. S. Burrus, "Index mappings for multidimensional formulation of the DFT convolution," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-25, pp. 239–242, 1977.

[2] C. S. Burrus and P. W. Eschenbacher, "An in-place, in-order prime factor FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 20, , pp. 806–817, 1981.

[3] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolutional Algorithms: Theory and Implementation.* New York: Wiley, 1985.

[4] S. C. Chan and K. L. Ho, "On indexing the prime factor fast Fourier transform algorithm," *IEEE Trans. Circuits Syst.,* vol. 38, pp. 951–953, 1991.

[5] S. L. Johnsson *et al.*, "Communication efficient multi-processor FFT, *J. Computat. Physics*," vol. 102, pp. 381–397, 1991.

[6] E. N. Linzer and E. Feig, "Implementation of efficient fft algorithms on fused multiply-add architectures," *IEEE Trans. Signal Processing,* vol. 41, pp. 93–107, 1993.

[7] D. P .K. Lun and W. C. Siu, "An analysis for the realization of an in-place and in-order prime factor algorithm," *IEEE Trans. Signal Processing,* vol. 41, 1993.

[8] D. P. K. Lun *et al.*, "On performance evaluation and address generation of prime factor algorithms," in *Proc. Int. Symp. Computer Architect. Digital Signal Processing,* pp. 352–357, 1989.

[9] J. H. Rothweiller, "Implementation of the in-order prime factor transform for variable sizes," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 30, , pp. 105–107, 1982.

[10] W. C. Siu and K. L. Wong, "Fast address generation for the computation of prime factor algorithms," in *Proc. Int. Conf. Acoust. Speech, Signal Processing,* 1989.

[11] G. Steidl and M. Tasche, "Index transforms for multidimensional DFT's and convolutions," *Numerische Mathematik,* vol. 56, pp. 513–528, 1989.

[12] C. Temperton, "Implementation of a self-sorting, in-place prime factor FFT algorithm," *J. Computat. Physics,* vol. 58, pp. 283–299, 1985.

[13] ——, "A generalized prime factor FFT algorithm for any $N = 2^p 3^q 5^r$," *SIAM J. Sci. Statist. Comput.,* vol. 13, 676–686, 1992.