

A Refined Fast 2-D Discrete Cosine Transform Algorithm

Yuh-Ming Huang and Ja-Ling Wu

Abstract—In this correspondence, an index permutation-based fast two-dimensional discrete cosine transform (2-D DCT) algorithm is presented. It is shown that the $N \times N$ 2-D DCT, where $N = 2^m$, can be computed using only N 1-D DCT's and some post additions.

Index Terms—Discrete cosine transform, fast algorithms, index permutation.

I. INTRODUCTION

The DCT is widely used in many digital signal processing applications. Fast algorithms have been reported in the literature (i.e., [1]–[6]). Among those algorithms, [5] and [6] are believed to be the most efficient 2-D DCT algorithms in the sense of minimizing any measure of computational complexity.

Recently, Cho and Lee [7] have proposed a fast and modular DCT algorithm in which an $(N \times N)$ -point 2-D DCT could be obtained by computing $N \times N$ -point 1-D DCT's and $\log_2 N + 1$ butterfly stages. In a later work [8], they also provide regular expressions for the input–output relations of the post-addition stages. However, the number of required additions increases at the expense of improving the regularity in the structure.

Based on the idea of [7], in this correspondence, an index permutation-based fast 2-D DCT algorithm is proposed.

II. THE REFINED FAST ALGORITHM FOR COMPUTING THE 2-D DCT

For a given input data sequence $f_{i,j}$, $0 \leq i \leq N-1$, $0 \leq j \leq N-1$, the denormalized 2-D DCT can be expressed as [1]

$$Y_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{i,j} \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2N} \quad (1)$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1.$$

After some permutation of the input data sequence [2], (1) can be written as

$$Y_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(4i+3)m\pi}{2N} \cos \frac{(4j+3)n\pi}{2N} \quad (2.1)$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1$$

Manuscript received July 8, 1996; revised July 31, 1998. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xiang-Gen Xia.

Y.-M. Huang is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., and the Department of Information Engineering, National Chi-Nan University, Puli, Taiwan, R.O.C.

J.-L. Wu is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Publisher Item Identifier S 1053-587X(99)01358-6.

where

$$X_{i,j} = \begin{cases} f_{2i+1, 2j+1} & 0 \leq i, j \leq N/2 - 1 \\ f_{2i+1, 2N-2j-2} & 0 \leq i \leq N/2 - 1, N/2 \leq j \leq N-1 \\ f_{2N-2i-2, 2j+1} & N/2 \leq i \leq N-1, 0 \leq j \leq N/2 - 1 \\ f_{2N-2i-2, 2N-2j-2} & N/2 \leq i, j \leq N-1. \end{cases} \quad (2.2)$$

Based on the idea of [7], let

$$A_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(4i+3)m + (4j+3)n}{2N} \pi \quad (3.1)$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1$$

and

$$B_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(4i+3)m - (4j+3)n}{2N} \pi \quad (3.2)$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1.$$

Then

$$Y_{m,n} = (A_{m,n} + B_{m,n})/2. \quad (4)$$

Since $4j+3$ and N are coprime to each other, i.e., $(4j+3, N)^1 = 1$, the permutation $(4j+3)i + j$ modulo N maps all values of i . Let q_{ij} be the quotient of $(4j+3)i + j$ divided by N . Hence, the kernels of the 2-D transforms in (3.1) and (3.2) can be rewritten as 1-D DCT's by replacing i with $(4j+3)i + j - Nq_{ij}$. That is

$$A_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{((4j+3)i+j)_N, j} \cdot \cos \frac{(4j+3)((4i+1)m+n) - 4Nq_{ij}m}{2N} \pi$$

$$= \sum_{i=0}^{N-1} \left[\sum_{j=0}^{N-1} X_{((4j+3)i+j)_N, j} \cdot \cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi \right]$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1 \quad (5.1)$$

and

$$B_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{((4j+3)i+j)_N, j} \cdot \cos \frac{(4j+3)((4i+1)m-n) - 4Nq_{ij}m}{2N} \pi$$

$$= \sum_{i=0}^{N-1} \left[\sum_{j=0}^{N-1} X_{((4j+3)i+j)_N, j} \cdot \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \right]$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1 \quad (5.2)$$

where $\langle x \rangle_N$ denotes x modulo N .

For the simplicity of notation, $X_{((4j+3)i+j)_N, j}$ is denoted as $\bar{X}_{i,j}$. Then, it can be seen that the 2-D input data sequence is grouped into

¹ $\gcd(a, b)$ denotes the gcd of a and b .

N distinct data sets of size N , that is, $\{(\bar{X}_{i,j}, 0 \leq j \leq N-1), 0 \leq i \leq N-1\}$, and the equations

$$\sum_{j=0}^{N-1} \bar{X}_{i,j} \cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi \quad (6.1)$$

and

$$\sum_{j=0}^{N-1} \bar{X}_{i,j} \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \quad (6.2)$$

correspond to one of the 1-D DCT's of the data sequence $\hat{X}_{i,j}$ or equal to zero with respect to m and n . That is, by defining

$$h_{il} = \sum_{j=0}^{N-1} \bar{X}_{i,j} \cos \frac{(4j+3)l}{2N} \pi \quad (7)$$

we can see that (6.1) and (6.2) correspond to one of $+h_{il}$ or $-h_{il}$ for some $l = 0, 1, \dots, N-1$ or equal to zero. Besides, through an index permutation, (7) can be implemented by a 1-D DCT as

$$h_{il} = \sum_{j=0}^{N-1} \hat{X}_{i,j} \cos \frac{(2j+1)l}{2N} \pi \quad (8.1)$$

where

$$\hat{X}_{i,j} = \begin{cases} \bar{X}_{i,(j-1)/2} & j: \text{odd} \\ \bar{X}_{i,N-1-j/2} & j: \text{even.} \end{cases} \quad (8.2)$$

Hence, for the computation of an $(N \times N)$ -point 2-D DCT, only the computation of $N \times N$ -point 1-D DCT's and some post-additions are required.

Next, we will show that the post-addition stages can be implemented by a butterfly-like structure. Since

$$\begin{aligned} & \cos \frac{(4j+3)((4(i+N/2)+1)m+n)}{2N} \pi \\ &= \pm \cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi \end{aligned} \quad (9.1)$$

and

$$\begin{aligned} & \cos \frac{(4j+3)((4(i+N/2)+1)m-n)}{2N} \pi \\ &= \pm \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi. \end{aligned} \quad (9.2)$$

Hence, $Y_{m,n}$ can be expressed as in (10), shown at the bottom of the page. Let $\bar{X}_{i,j} + \bar{X}_{(i+N/2),j} = \bar{R}_{i,j}$, $\bar{X}_{i,j} - \bar{X}_{(i+N/2),j} = \bar{S}_{i,j}$.

For m even, since

$$\begin{aligned} & \cos \frac{(4j+3)((4(i+N/4)+1)m+n)}{2N} \pi \\ &= \pm \cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi \end{aligned} \quad (11.1)$$

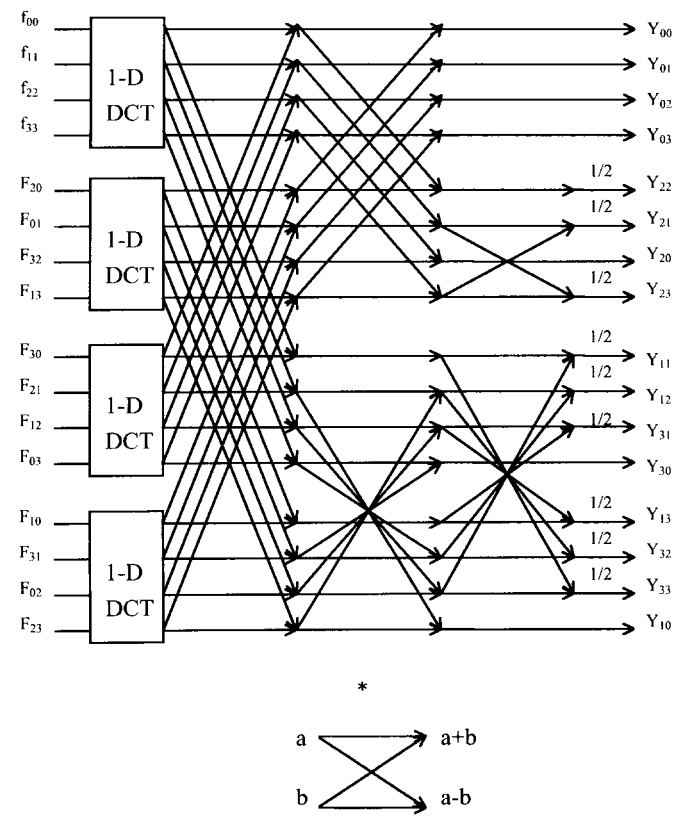


Fig. 1. Signal flow graph for 4×4 DCT.

and

$$\begin{aligned} & \cos \frac{(4j+3)((4(i+N/4)+1)m-n)}{2N} \pi \\ &= \pm \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \end{aligned} \quad (11.2)$$

$Y_{m,n}$ can be written as in (12), shown at the bottom of the page, where $k = 0, 1, \dots, N/4 - 1$.

For example, if $N = 4$, by (12), we have

$$\begin{aligned} Y_{2,n} = \frac{1}{2} \sum_{j=0}^3 \left\{ (\bar{R}_{0,j} - \bar{R}_{1,j}) \left(\cos \frac{(4j+3)(2+n)}{2N} \pi \right. \right. \\ \left. \left. + \cos \frac{(4j+3)(2-n)}{2N} \pi \right) \right\}. \end{aligned} \quad (13)$$

$$Y_{m,n} = \begin{cases} \frac{1}{2} \sum_{i=0}^{N/2-1} \sum_{j=0}^{N-1} \left\{ (\bar{X}_{i,j} + \bar{X}_{(i+N/2),j}) \left(\cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi + \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \right) \right\}, & \text{for } m \text{ even} \\ \frac{1}{2} \sum_{i=0}^{N/2-1} \sum_{j=0}^{N-1} \left\{ (\bar{X}_{i,j} - \bar{X}_{(i+N/2),j}) \left(\cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi + \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \right) \right\}, & \text{for } m \text{ odd} \end{cases} \quad (10)$$

$$Y_{m,n} = \begin{cases} \frac{1}{2} \sum_{i=0}^{N/4-1} \sum_{j=0}^{N-1} \left\{ (\bar{R}_{i,j} + \bar{R}_{(i+N/4),j}) \left(\cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi + \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \right) \right\}, & \text{for } m = 2(2k) \\ \frac{1}{2} \sum_{i=0}^{N/4-1} \sum_{j=0}^{N-1} \left\{ (\bar{R}_{i,j} - \bar{R}_{(i+N/4),j}) \left(\cos \frac{(4j+3)((4i+1)m+n)}{2N} \pi + \cos \frac{(4j+3)((4i+1)m-n)}{2N} \pi \right) \right\}, & \text{for } m = 2(2k+1) \end{cases} \quad (12)$$

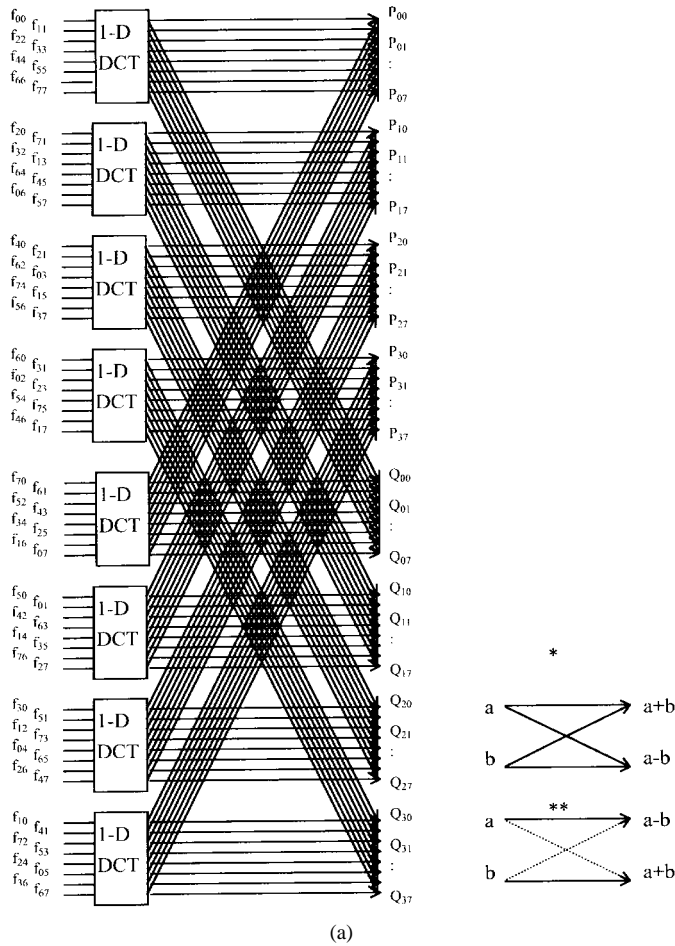


Fig. 2. Signal flow graph for 8×8 DCT. (a) First stage of the post-addition stages.

For m odd, let

$$k_{il} = \sum_{j=0}^{N-1} \bar{S}_{i,j} \cos \frac{(4j+3)l}{2N} \pi. \quad (14)$$

Since

$$\begin{aligned} & \cos \frac{(4j+3)((4(i+N/4)+1)m+n)}{2N} \pi \\ &= \pm \sin \frac{(4j+3)((4i+1)m+n)}{2N} \pi \end{aligned} \quad (15.1)$$

and

$$\begin{aligned} & \cos \frac{(4j+3)((4(i+N/4)+1)m-n)}{2N} \pi \\ &= \pm \sin \frac{(4j+3)((4i+1)m-n)}{2N} \pi. \end{aligned} \quad (15.2)$$

However

$$\sin \frac{(4j+3)l}{2N} \pi = -\cos \frac{(4j+3)(N-l)}{2N} \pi \quad (16)$$

that is, the 1-D discrete sine transform can be directly computed from the 1-D discrete cosine transform. Therefore, for some r and

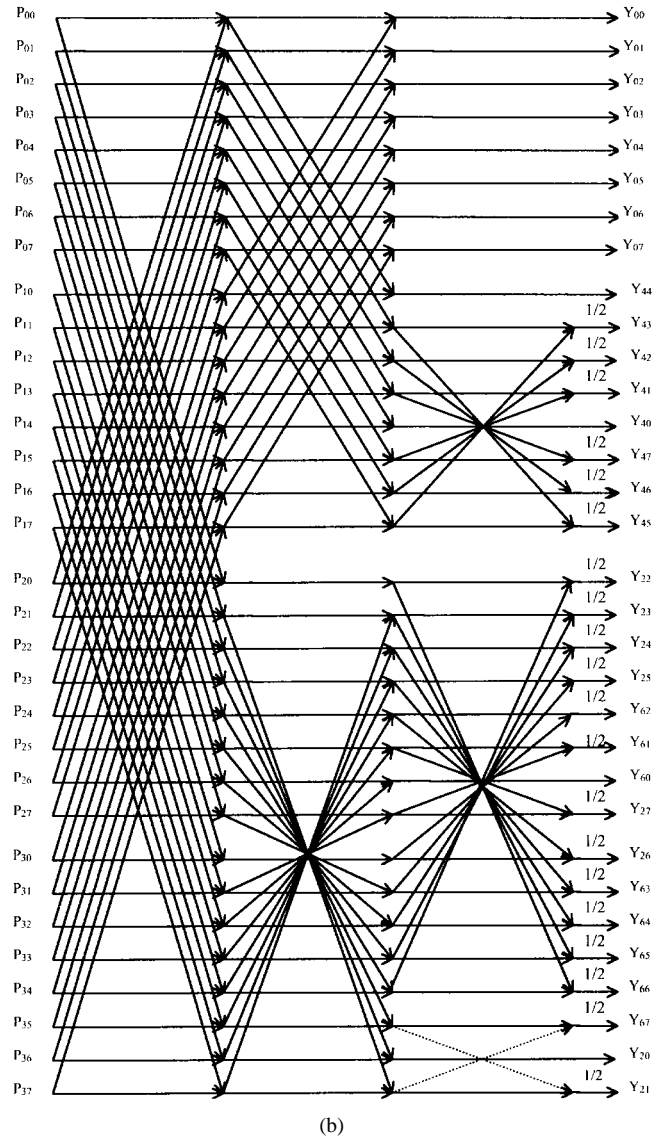


Fig. 2. (Continued). Signal flow graph for 8×8 DCT. (b) Second, third, and fourth stages of the post-addition stages (for n is even).

s , $0 \leq r, s \leq N$, $Y_{m,n}$ can be written as shown in (17), shown at the bottom of the page, where $k = 0, 1, \dots, N/4 - 1$, and \bar{G}_{ir} and \bar{H}_{is} are, respectively, equal to $\pm k_{ir}$ and $\pm k_{is}$.

For example, if $N = 4$, $r = 3$, and $s = 1$, by (17), we have

$$Y_{3,2} = \frac{1}{2} \{(-k_{03} + k_{11}) + (k_{01} + k_{13})\}.$$

As a result, the computation of an $(N \times N)$ -point 2-D DCT can be achieved by recursively applying the above decompositions [(12) and (17)]. The signal flow graphs for a 4×4 and an 8×8 DCT are shown in Figs. 1 and 2, respectively.

$$Y_{m,n} = \begin{cases} \frac{1}{2} \sum_{i=0}^{N/4-1} \{(\bar{G}_{ir} - \bar{G}_{(i+N/4)(N-r)}) + (\bar{H}_{is} + \bar{H}_{(i+N/4)(N-s)})\} & \text{for } m = 2(2k) + 1 \\ \frac{1}{2} \sum_{i=0}^{N/4-1} \{(\bar{G}_{ir} + \bar{G}_{(i+N/4)(N-r)}) + (\bar{H}_{is} - \bar{H}_{(i+N/4)(N-s)})\} & \text{for } m = 2(2k+1) + 1 \end{cases} \quad (17)$$

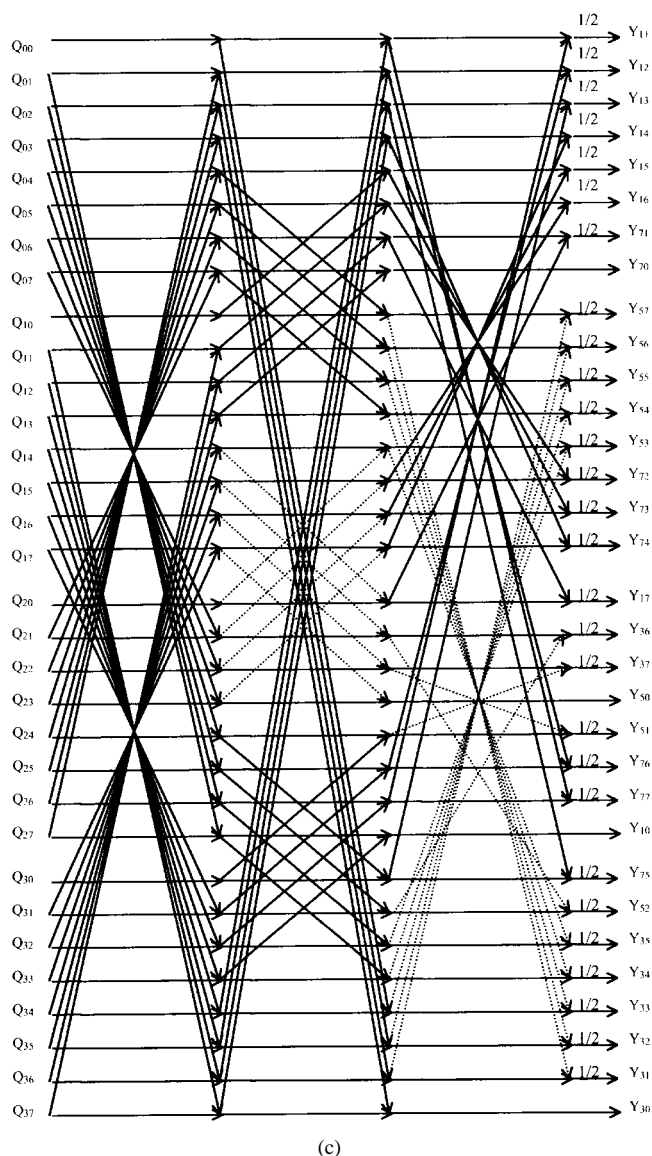


Fig. 2. (Continued). Signal flow graph for 8×8 DCT. (c) Second, third, and fourth stages of the post-addition stages (for n is odd).

III. COMPLEXITY ANALYSIS OF THE POST-ADDITION STAGES

For the post-addition stages, let $A(N)$ and $B(N)$, respectively, denote the number of all required additions and the number of additions required in the final stage, and let $C(N)$ denote the number of nodes that do not require butterfly computations in the first $\log_2 N$ stages. From (13) and (16), we have $C(4) = 2$ and $C(N) = C(N/2) + N/2$ for $N \geq 8$, and $B(N) = N^2 - 2N$. Therefore, $A(N) = N^2 \log_2 N - C(N) + B(N) = N^2(1 + \log_2 N) - 3N + 2$.

IV. CONCLUSIONS

An index-permutation based 2-D DCT algorithm has been presented in this correspondence. The succinct derivation of the proposed algorithm makes it easier to describe the process of how to map one 2-D DCT into a number of 1-D DCT's. From the idea of [8], a matrix-form-based systematic expression for the post-addition stages

in the proposed algorithm, which may improve the regularity of the structure, is currently under investigation.

REFERENCES

- [1] R. K. Rao and P. Yip, *Discrete Cosine Transform: Algorithm, Advantages, and Applications*. New York: Academic, 1990.
- [2] M. J. Narasimha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COMM-26, pp. 934–936, June 1978.
- [3] H. S. Hou, "A fast recursive algorithms for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1455–1461, Oct. 1987.
- [4] P. Lee and F.-Y. Huang, "Restructured recursive DCT and DST algorithms," *IEEE Trans. Signal Processing*, vol. 42, pp. 1600–1609, July 1994.
- [5] P. Duhamel and C. Guillemot, "Polynomial transform computation of 2-D DCT," in *Proc. ICASSP*, Apr. 1990, pp. 1515–1518.
- [6] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, Sept. 1992.
- [7] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-D DCT," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 297–305, Mar. 1991.
- [8] N. I. Cho, I. D. Yun, and S. U. Lee, "On the regular structure for the fast 2-D DCT algorithm," *IEEE Trans. Circuits Syst.*, vol. 40, pp. 259–266, Apr. 1993.

An Effective Memory Addressing Scheme for FFT Processors

Yutai Ma

Abstract—The memory organization of FFT processors is considered. The new memory addressing assignment allows simultaneous access to all the data needed for butterfly calculations. The advantage of this memory addressing scheme lies in the fact that it reduces the delay of address generation nearly by half compared to existing ones.

I. INTRODUCTION

Many high-speed FFT processors have been obtained by implementing the fast Fourier transform in pipelined digital hardware with a butterfly calculation unit, two-port data memory, ROM for storing twiddle factors, and memory addressing controller integrated on a chip. It is possible to use an in-place strategy that stores butterfly outputs in those memory locations used by the inputs to the same butterfly. The in-place strategy requires only a minimum amount of memory. For this reason, only the in-place radix-2 decimation-in-time version of the fast Fourier transform is considered here.

If the butterfly unit has parallel inputs and outputs, then the two butterfly inputs will be accessed in the memory, and two butterfly outputs will be written back to the same memory in each cycle. In order to avoid this memory bottleneck, the two-port memory module

Manuscript received May 23, 1997; revised July 22, 1998. This work was supported by the National Key Project of Fundamental Research, P.R. China. The associate editor coordinating the review of this paper and approving it for publication was Dr. Elias S. Manolakos.

The author was with the Center for High Performance Computing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P. R. China. He is now with the Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Publisher Item Identifier S 1053-587X(99)00736-9.