순환 행렬 분해에 의한 DCT/DFT 하이브리드 구조 알고리듬

DCT/DFT Hybrid Architecture Algorithm Via Recursive Factorization

박 대 철*

Dae-chul Park*

요 약

본 논문은 순환 행렬 분해에 의한 DCT 와 DFT의 고속 계산을 위한 하이브리드 아키텍처 알고리듬을 제안한다. DCT-II 와 DFT 변환 행렬의 순환 분해는 알고리듬적으로 구현하기가 유사한 구조를 제공하며 이것은 단순히 스위칭 모드의 제어에 의해 공통 아키텍처를 사용할 수 있게한다. 두 변환간의 연계는 행렬 순환 공식에 기초하여 유도되었다. DCT/DFT 행렬 분해를 위한 하이브리드 구조 설계를 가능하도록 생성 행렬, 삼각함수 항등식 과 관계식을 사용하여 유도되었다. DCT/DFT 하이브리드 아키텍처를 수용하는 쿨리-투키 유형의 고속처리 아키텍처에 대한 데이터 흐름도를 작성하였다. 이 데이터 흐름도로부터 적절한 크기의 N에 대해 제안한 알고리듬의 계산 복잡도는 기존의 고속 DCT 알고리듬과 비교할만하다. 다른 직교변환 계산에 FFT 구조의 다중 모드 사용 확장을 위해 좀더 확장된 연구가 필요하다.

Abstract

This paper proposes a hybrid architecture algorithm for fast computation of DCT and DFT via recursive factorization. Recursive factorization of DCT-II and DFT transform matrix leads to a similar architectural structure so that common architectural base may be used by simply adding a switching device. Linking between two transforms was derived based on matrix recursion formula. Hybrid acrchitectural design for DCT and DFT matrix decomposition were derived using the generation matrix and the trigonometric identities and relations. Data flow diagram for high-speed architecture of Cooley-Tukey type was drawn to accommodate DCT/DFT hybrid architecture. From this data flow diagram computational complexity is comparable to that of the fast DCT algorithms for moderate size of N. Further investigation is needed for multi-mode operation use of FFT architecture in other orthogonal transform computation.

Keywords : recursive factorization, DCT-II, DFT, hybrid architecture, fast computation

I. Introduction

DCT (Discrete Cosine Transform) and DFT (Discrete Fourier Transform) have been found in many applicati ons for signal classification, representation and image coding. DCT was found as the best suboptimal transform in its performance, which is very close to that of KLT(Karhunen-Loeve Transform) for picture coding[1]. Also most FFT (Fast Fourier Transform) algorithms based on Cooly-Tukey data flow diagram were implemented and applied to signal processing and communication fields such as OFDM transmission and orthogonal code designs. A link between these transfo rms was attempted in this paper by exploiting the characteristics of sparse matrix decomposition of the unitary matrix. Many researchers investigated the links among unitary transforms using QR-like factorization called Jacket-like sparse matrix deco

*Dept. of Information and Communication Engineering, Hannam University

논문 번호 : 2007-1-8 접수 일자 : 2007. 2. 22
심사 완료 : 2007. 4. 18
※이 논문은 2006년도 한남대학교 학술조성연구비 지원에 의해 연구되었음.

mposition[2,3,4,7,10].
Lee [3] has proposed the Reverse Jacket Transform(RJT) based on a generalization of the WHT(Walsh Hadamard Transform). Recently, Park et al.[4] also discovered an interesting relationship in Jacket-like sparse matrix representation of DFT. In this paper, we propose a hybrid architecture algorithm, which links DCT/DFT by using recursive decomposition of a Jacket-like sparse matrix.

II. Recursive Factorization

2.1 Jacket Matrix Concept and Properties

Any normal matrix (for example, real symetric, hermitian, skew-hermitian, unitary matrix) could be diagonalized by unitary transform. As a general case , any matrix with linearly independent columns can be factored into AE=QR. Such factorization can be viewed as a Jacket matrix representation meaning a jacket with inside and outside.. Normally, matrix R is a sparse matrix and matrix Q is a unitary matrix.
As a special case of QR decomposition, the matrix Q can be fixed to a trigonometric transform matrix by a constraint.
Being viewed with this concept, a Jacket(J) matrix is a generalized weighted Hadamard transform matrix [5]. Here $[S]_m$ is a sparse matrix of $[J]_m$.

$$[J]_m = \frac{1}{m}[H]_m[S]_m, m = 2^{k+1}, k \in \{1,2,3,4,\ldots\} \tag{1}$$

Jacket matrix $[J]_m$ has an element/block inverse property[2]. The inverse of $[J]_m$ is also a Jacket matrix..
Most trigonometric transform matrix (DFT, DCT, DST, WHT, Haar Transform, Hartley Transform, etc) can be represented as a Jacket-like sparse matrix. For example, the DFT matrix for N=8 case can be decomposed as follows[4]:

$$[F]_8 = \frac{1}{4}[H]_8\begin{bmatrix} 4I_2 & 0 & 0 \\ 0 & 2(G_2H_2)^h & 0 \\ 0 & 0 & (G_4H_4)^h \end{bmatrix}[P]_8 \tag{2}$$
$$= \frac{1}{4}[H]_8[\hat{S}]_8[P]_8 = \frac{1}{4}[\tilde{F}]_8[P]_8$$

where *[F]₈* and *[H]₈* are a 8-point DFT and Walsh Hadamard Transform matrix, respectively. Also $G_2 = [F]_2[D]_2$ and $G_4 = [F]_4[D]_4$, where in general

$$[D]_N = diag(W^{0/2N}, W^{1/2N}, W^{2/2N}, \ldots)$$

2.2 Recursive Factorization of DCT

As mentioned before DCT-II belongs to a family of a Jacket-like sparse matrix, meaning that DCT-II can be factored out by the unitary matrix and a sparse matrix. Discrete Cosine Transform (DCT) is widely used in image processing, and orthogonal transform. A typical DCT matrix is DCT-II [1], which is defined as DCT-II:

$$[C_N]_{mn} = \sqrt{\frac{2}{N}}\, k_m \cos\frac{m(n+\frac{1}{2})\pi}{N}, m.n = 0,1,\ldots,N-1, \tag{3}$$
$$where\ k_j = \begin{cases} 1, & j = 1,2,\ldots,N-1 \\ \frac{1}{\sqrt{2}}, & j = 0,N \end{cases}$$

To generalize a recursive factorization of size N DCT-II , we start with N=2,4, and 8:

$$[C]_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C^{3_4} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{4}$$
$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} C_1 & C_1 \\ B_1 & -B_1 \end{bmatrix}$$

where $C_1 = 1/\sqrt{2}$ can be seen as a special element inverse matrix of order 1, its inverse is $\sqrt{2}$, and $C_l^i = \cos(i\pi/l)$ is the cosine unit for DCT computations.
Furthermore, 4-by-4 DCT - II matrix can be rewritten by using identities
$C_8^1 = -C_8^7, C_8^2 = -C_8^6, C_8^3 = -C_8^5$ as

$$[\mathrm{Pr}]_4[C]_4 = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & -C_8^2 & -C_8^2 & C_8^2 \\ C_8^1 & C_8^3 & -C_8^3 & -C_8^1 \\ C_8^3 & -C_8^1 & C_8^1 & -C_8^3 \end{bmatrix} \tag{5}$$

where $[\mathrm{Pr}]_4$ is a permutation matrix for N=4 for $[\mathrm{Pr}]_N$ which has the bit reversal order(BRO) form of *[I]ₙ* :

$$[\mathrm{Pr}]_2=[I]_2,\ [\mathrm{Pr}]_N=\begin{bmatrix}100\ldots00\ldots0\\000\ldots10\ldots0\\010\ldots00\ldots0\\000\ldots01\ldots0\\001\ldots00\ldots0\\000\ldots00\ldots0\\\ldots\ \ \ldots\ \ \ldots\\0\,000\ldots000\,1\end{bmatrix},N\geq 4 \qquad (6)$$

with

$$\begin{cases}pr_{i,j}=1, & if\ i=2j,\ 0\leq j\leq N/2-1\\ pr_{i,j}=1, if\ i=(2j+1)\bmod N, & N/2\leq j\leq N-1\\ pr_{i,j}=0, & others\end{cases}$$

Let *where* $[\mathrm{Pr}]_N=[pr_{i,j}]_N$

Let us define a column permutation matrix *,[Pc]*$_N$, which has the form as

$$[Pc]_2=[I]_2,\ [Pc]_N=\begin{bmatrix}I_{N/4}&0&0&0\\0&I_{N/4}&0&0\\0&0&0&I_{N/4}\\0&0&I_{N/4}&0\end{bmatrix},N\geq 4 \qquad (7)$$

Thus we have for N=4

$$[\mathrm{Pr}]_4[C]_4[Pc]_4=\begin{bmatrix}1000\\0010\\0100\\0001\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\C_8^1&C_8^3&C_8^5&C_8^7\\C_8^2&C_8^6&C_8^6&C_8^2\\C_8^3&C_8^7&C_8^1&C_8^5\end{bmatrix}\begin{bmatrix}1000\\0100\\0001\\0010\end{bmatrix}$$

$$=\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\C_8^2&C_8^6&C_8^6&C_8^2\\C_8^1&C_8^3&C_8^5&C7\\C_8^3&C_8^7&C_8^1&C_8^3\end{bmatrix}=\begin{bmatrix}[C]_2&[C]_2\\ [B]_2&-[B]_2\end{bmatrix}=\begin{bmatrix}[C]_2&0\\0&[B]_2\end{bmatrix}\begin{bmatrix}I_2&0\\0&-I_2\end{bmatrix} \qquad (8)$$

It can be shown that $\begin{bmatrix}[C]_2&[C]_2\\ [B]_2&-[B]_2\end{bmatrix}$ satisfies properties of the Jacket matrix. Similarly, for N=8, we obtain the followings:

$$[\mathrm{Pr}]_8[C]_8[Pc]_8=[\tilde{C}]_8=\begin{bmatrix}C_4&C_4\\B_4&-B_4\end{bmatrix} \qquad (9)$$
$$=\begin{bmatrix}C_4&0\\0&B_4\end{bmatrix}\begin{bmatrix}I_4&I_4\\I_4&-I_4\end{bmatrix}$$

In general the permuted DCT-II matrix can be decomposed recursively as follows:

$$[\tilde{C}]_N=[\mathrm{Pr}]_N[C]_N[Pc]_N=\begin{bmatrix}C_{N/2}&C_{N/2}\\B_{N/2}&-B_{N/2}\end{bmatrix} \qquad (10)$$
$$=\left(\begin{bmatrix}C_{N/2}&0\\0&-B_{N/2}\end{bmatrix}\begin{bmatrix}I_{N/2}&I_{N/2}\\I_{N/2}&-I_{N/2}\end{bmatrix}\right)$$

In (10) top left block matrix *[C]*$_{N/2}$ of *[C]*$_N$ has a recursive factorization, but bottom right block matrix *[B]*$_{N/2}$ of *[C]*$_N$ does not.

Many authors [8,9,10] proposed a further decomposition algorithm to derive a fast implementation of *[B]*$_{N/2}$ computation, which normally requires ($N/2\ \times\ N/2$) real multiplication.

The proposed algorithm partitions *[B]*$_{N/2}$ into a recursive form using both the generation matrix and the trigonometric identities and relations to be explained below:

Generation matrix :

$$[B]_{N/2}=[(C_{2N}^{f(m,n)})_{m,n}]_{N/2}, \qquad (11)$$
$$where\begin{cases}f(m,1)=2m-1,\\f(m,n+1)=f(m,n)+f(m,1)*2\end{cases}$$

In case of $N\times N$ DCT-II matrix, *[C]*$_N$ can be represented as in (12).

$$[C]_N=\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\cdots&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\C_{4N}^{2k_0\Phi_0}&C_{4N}^{2k_0\Phi_1}&&C_{4N}^{2k_0\Phi_{N-2}}&C_{4N}^{2k_0\Phi_{N-1}}\\C_{4N}^{2k_1\Phi_0}&C_{4N}^{2k_1\Phi_1}&&C_{4N}^{2k_1\Phi_{N-2}}&C_{4N}^{2k_1\Phi_{N-1}}\\C_{4N}^{2k_2\Phi_0}&C_{4N}^{2k_2\Phi_1}&&C_{4N}^{2k_2\Phi_{N-2}}&C_{4N}^{2k_2\Phi_{N-1}}\\\vdots&&&&\vdots\\C_{4N}^{2k_{N-2}\Phi_0}&C_{4N}^{2k_{N-2}\Phi_1}&\cdots&C_{4N}^{2k_{N-2}\Phi_{N-2}}&C_{4N}^{2k_{N-2}\Phi}\end{bmatrix} \qquad (12)$$

where $k_i=i+1, i\in\{0,1,2,\ldots\}$

According to (11), a $N\times N$ matrix *[B]*$_N$ from *[C]*$_{2N}$ can be derived to (13) by using the trigonometric identities and relations given by

$$C_{4N}^{(2k_i+1)\Phi_m}=2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}-C_{4N}^{(2k_i-1)\Phi_m}$$
$$=-C_{4N}^{(2k_i-1)\Phi_m}+2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m},$$

$m\in 0,1,2,\ldots$

Now we have

$$[B]_N=\begin{bmatrix}C_{4N}^{\Phi_0}&C_{4N}^{\Phi_0}&\cdots&C_{4N}^{\Phi_0}\\C_{4N}^{(2k_0+1)\Phi_0}&C_{4N}^{(2k_0+1)\Phi_1}&\cdots&C_{4N}^{(2k_0+1)\Phi_{N-1}}\\C_{4N}^{(2k_1+1)\Phi_0}&C_{4N}^{(2k_1+1)\Phi_1}&\cdots&C_{4N}^{(2k_1+1)\Phi_{N-1}}\\\vdots&&&\vdots\\C_{4N}^{(2k_{N-2}+1)\Phi_0}&C_{4N}^{(2k_{N-2}+1)\Phi_1}&\cdots&C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}}\end{bmatrix}$$
$$=[K]_N[C]_N[D]_N \qquad (13)$$

where

$$[K]_N=\begin{bmatrix}\sqrt{2}&0&0\cdots\\-\sqrt{2}&2&0\cdots\\\sqrt{2}&-2&2\cdots\\\vdots&\vdots&\vdots\ \ddots\end{bmatrix}, \text{and}$$

$$[D]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \cdots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix} \tag{14}$$

The proof for this is shown in appendix.
By using the results obtained from the previous equations (11)~(14) , we have a new form for DCT - II matrix as :

$$[C]_N = [\mathrm{Pr}]_N^{-1}[\tilde{C}]_N[Pc]_N^{-1} = [\mathrm{Pr}]_N[\tilde{C}]_N[Pc]_N \tag{15}$$

where

$$\begin{aligned} [\tilde{C}]_N &= [\mathrm{Pr}]_N[C]_N[Pc]_N \\ &= \left(\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \right)^T \\ &= \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} C_{N/2} & 0 \\ 0 & K_{N/2}C_{N/2}D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \end{aligned} \tag{16}$$

It can be easily shown that center diagonal matrix can now be factorized in a recursive manner as

$$\begin{aligned} &\begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/4} & 0 & 0 & 0 \\ 0 & K_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & 0 \\ 0 & 0 & 0 & K_{N/4} \end{bmatrix} \bullet \begin{bmatrix} C_{N/4} & 0 & 0 & 0 \\ 0 & C_{N/4} & 0 & 0 \\ 0 & 0 & C_{N/4} & 0 \\ 0 & 0 & 0 & C_{N/4} \end{bmatrix} \bullet \\ &\begin{bmatrix} I_{N/4} & 0 & 0 & 0 \\ 0 & D_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & 0 \\ 0 & 0 & 0 & D_{N/4} \end{bmatrix} \bullet \begin{bmatrix} I_{N/4} & I_{N/4} & 0 & 0 \\ I_{N/4} & -I_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & I_{N/4} \\ 0 & 0 & I_{N/4} & I_{N/4} \end{bmatrix} \end{aligned} \tag{17}$$

2.3 Recursive Factorization of DFT

In a similar way, for N=4, we can factorized a DFT matrix $[F]_4$, where $[F]_4 = [W^{nm_4}] = [e^{-j\frac{2\pi mn}{4}}]$, into a recursive form:

$$\begin{aligned} [\tilde{F}]_4 = [\mathrm{Pr}]_4[F]_4 &= \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \\ &= \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} F_2 & F_2 \\ E_2 & -E_2 \end{bmatrix} \end{aligned} \tag{18}$$

In general, we have

$$\begin{aligned} [\tilde{F}]_N &= [\mathrm{Pr}]_N[F]_N \\ &= \left(\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} \widetilde{F_{N/2}} & 0 \\ 0 & E_{N/2} \end{bmatrix} \right)^T \\ &= \begin{bmatrix} \widetilde{F_{N/2}} & 0 \\ 0 & E_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} \widetilde{F_{N/2}} & 0 \\ 0 & \mathrm{Pr}_{N/2}\widetilde{F_{N/2}}W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & \mathrm{Pr}_{N/2} \end{bmatrix} \begin{bmatrix} \widetilde{F_{N/2}} & 0 \\ 0 & \widetilde{F_{N/2}} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \end{aligned} \tag{19}$$

where $[F]_2 = [\tilde{F}]_2$. And the submatrix $E_{N/2}$ could be written by $[E]_N = [\mathrm{Pr}]_N[\tilde{F}]_N[W]_N$, where

$$[W]_N = \begin{bmatrix} W^0 & 0 & \cdots & 0 \\ 0 & W^1 & & 0 \\ 0 & & \ddots & 0 \\ 0 & & \cdots & W^{N-1} \end{bmatrix},$$ and W is the n-th root of unity for $2N$ point DFT matrix.

Finally, based on the recursive form we have

$$\begin{aligned} [F]_N &= [\mathrm{Pr}]_N^{-1}[\tilde{F}]_N \\ &= [\mathrm{Pr}]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & \mathrm{Pr}_{N/2} \end{bmatrix} \bullet \begin{bmatrix} \tilde{F}_{N/2} & 0 \\ 0 & \widetilde{F_{N/2}} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \end{aligned} \tag{20}$$

2.2.3 Link between Two Transforms

From eqns (15) and (20) we have

$$[C]_N = [\mathrm{Pr}]_N^{-1}[\tilde{C}]_N[Pc]_N^{-1} = [\mathrm{Pr}]_N[\tilde{C}]_N[Pc]_N, \tag{21}$$

where

$$[\tilde{C}]_N = \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \bullet \begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix} \bullet \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \bullet \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \qquad (22)$$

And $[F]_N = [\Pr]_N [\tilde{F}]_N [I]_N$ (23)

where

$$[\tilde{F}]_N = \begin{bmatrix} I_{N/2} & 0 \\ 0 & \Pr_{N/2} \end{bmatrix} \bullet \begin{bmatrix} \overline{F_{N/2}} & 0 \\ 0 & \overline{F_{N/2}} \end{bmatrix} \bullet \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \bullet \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \qquad (24)$$

Main differences lie in center matrices. However the architectural structure of the center matrices resembles each other. This feature lets us utilize common butterfly data flow diagram based on Cooley-Tukey that was used in the development of FFT. We will show that in the next section.

III. Hybrid Architecture Design

3.1 Cooley-Tukey Type Data Flow Diagram of DCT

Using (22) butterfly data flow diagram for DCT-II transform is drawn from left to right to perform X=$[C]_N$ x.

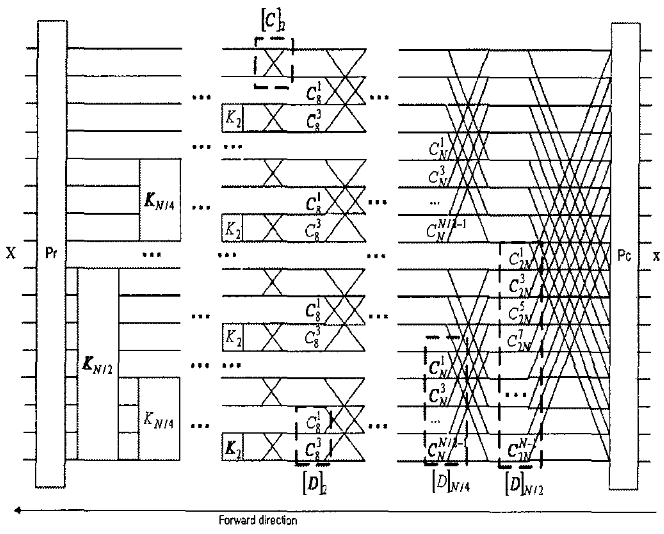

Fig. 1. Butterfly data flow diagram of the proposed DCT - II matrix with order N

3.2 Cooley-Tukey Type Data Flow Diagram of DFT

Similarly, using (23) butterfly data flow diagram for DFT transform is drawn from left to right to perform X=$[F]_N$ x.

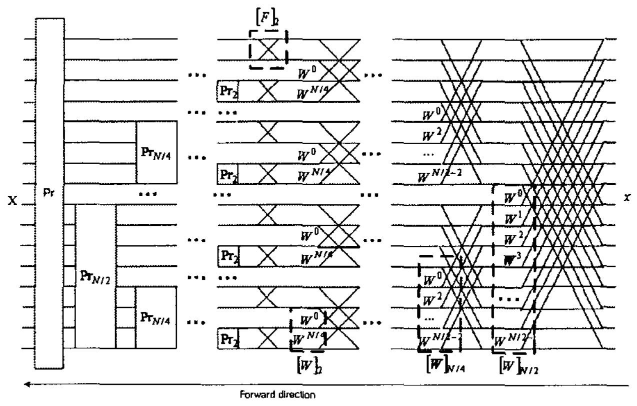

Fig.2. Butterfly data flow diagram of the proposed DFT matrix with order N

It is clear that the form of (22) is the same as that of (24), where we only need change $[K]_m$ to $[Pr]_m$ and $[D]_m$ to $[W]_m$, with $m \in \{2,4,8,..., N/2\}$.

Hence a simple generalized block diagram for DCT/DFT hybrid architecture algorithm can be drawn as Fig.3. In this figure, we can combine DCT and DFT in one processing architecture, and use the switching box to control mode of operation. It will be useful in developing a single chip to perform DCT and/or DFT processing task.

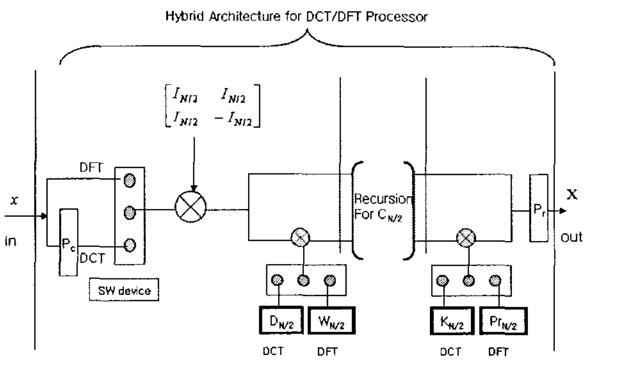

Fig.3. A simple generalized block diagram for proposed DCT/DFT hybrid architecture algorithm

Computational complexity[10] of the proposed recursive factorization scheme is comparable to FFT, Chen's DCT[8] and Wang's DCT algorithms [9] as shown below:

FFT : Complex additions($N\log^{N/2}$)
DCT-II (Chen) : Real additions
$3N/2(\log^{N/2})+2, N\geq 4$
DCT-II(Wang) : Real additions
$3N/2(\log^{N/2}-1)+2, N\geq 4$
DCT-RF(proposed) : Real additions
$N\log^{N/2}+N/2-1$

IV. Conclusions

In this paper, we derive the recursive formulas for DCT - II and DFT matrices. The results have been shown that the DCT - II and DFT can be unified by using the same architectural base with minor change. Hence a unified fast processing block to implement DCT/DFT hybrid architecture algorithm can be designed by adding switching device to control dual function of DCT and DFT processing depending on mode of operation. Also computational complexity is comparable to the fast DCT algorithms for moderate size of N. Also hybrid architecture has several advantages in making a single chip that operates in multifunction mode.
Further investigation is needed for multi-mode ope ration use of FFT architecture in other trigo nometric transform computation.

Appendix

Proof for $[B]_N=[K]_N[C]_N[D]_N$
In case of $N\times N$ DCT - II matrix, $[C]_N$, it can be represented by using the form as

$$[C]_N=\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & & C_{4N}^{2k_0\Phi_{N-2}} & C_{4N}^{2k_0\Phi_{N-1}} \\ C_{4N}^{2k_1\Phi_0} & C_{4N}^{2k_1\Phi_1} & & C_{4N}^{2k_1\Phi_{N-2}} & C_{4N}^{2k_1\Phi_{N-1}} \\ C_{4N}^{2k_2\Phi_0} & C_{4N}^{2k_2\Phi_1} & & C_{4N}^{2k_2\Phi_{N-2}} & C_{4N}^{2k_2\Phi_{N-1}} \\ \vdots & & & & \vdots \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & \cdots & C_{4N}^{2k_{N-2}\Phi_{N-2}} & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix}$$

(A-1)

,where $k_i=i+1, i\in 0,1,2,...,$.
According to generation matrix (11), a $N\times N$ matrix $[B]_N$ from $[C]_{2N}$ can be simply presented by

$$[B]_N=\begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & \cdots & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & \cdots & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ C_{4N}^{(2k_1+1)\Phi_0} & C_{4N}^{(2k_1+1)\Phi_1} & \cdots & C_{4N}^{(2k_1+1)\Phi_{N-1}} \\ \vdots & & & \vdots \\ C_{4N}^{(2k_{N-2}+1)\Phi_0} & C_{4N}^{(2k_{N-2}+1)\Phi_1} & \cdots & C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}} \end{bmatrix} \quad \text{(A-2)}$$

And based on trigonometric relations ,we have the formula

$$\begin{aligned} C_{4N}^{(2k_i+1)\Phi_m} &= 2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}-C_{4N}^{(2k_i-1)\Phi_m} \\ &=-C_{4N}^{(2k_i-1)\Phi_m}+2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}, \\ &,where\ m\in 0,1,2,... \end{aligned} \quad \text{(A-3)}$$

Thus we can calculate that
$[B]_N=[K]_N[C]_N[D]_N$

$$=\begin{bmatrix} \sqrt{2} & 0 & 0 & 0\cdots 0 \\ -\sqrt{2} & 2 & 0 & 0 \quad 0 \\ \sqrt{2} & -2 & 2 & 0 \quad 0 \\ -\sqrt{2} & 2 & -2 & 2\cdots \vdots \\ \sqrt{2} & -2 & 2 & -2 \quad 2 \\ \vdots & & & \ddots \end{bmatrix}\cdot$$

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & & C_{4N}^{2k_0\Phi_{N-2}} & C_{4N}^{2k_0\Phi_{N-1}} \\ C_{4N}^{2k_1\Phi_0} & C_{4N}^{2k_1\Phi_1} & & C_{4N}^{2k_1\Phi_{N-2}} & C_{4N}^{2k_1\Phi_{N-1}} \\ C_{4N}^{2k_2\Phi_0} & C_{4N}^{2k_2\Phi_1} & & C_{4N}^{2k_2\Phi_{N-2}} & C_{4N}^{2k_2\Phi_{N-1}} \\ \vdots & & & & \vdots \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & \cdots & C_{4N}^{2k_{N-2}\Phi_{N-2}} & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix}\cdot$$

$$\begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \cdots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix}$$

$$=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ -1+2C_{4N}^{2k_0\Phi_0} & -1+2C^{2k_0\Phi_1} & \cdots & -1+2C_{4N}^{2k_0\Phi_{N-1}} \\ 1-2C_{4N}^{2k_0\Phi_0}+2C_{4N}^{2k_1\Phi_0} & 1-2C_{4N}^{2k_0\Phi_1}+2C_{4N}^{2k_1\Phi_1} & \cdots & 1-2C_{4N}^{2k_0\Phi_{N-1}}+2C_{4N}^{2k_1\Phi_{N-2}} \\ \vdots & & \cdots & \vdots \end{bmatrix}\cdot$$

$$\begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \cdots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix}$$

$$=\begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & \cdots & C_{4N}^{\Phi_{N-1}} \\ -C_{4N}^{\Phi_0}+2C_{4N}^{2k_0\Phi_0}C_{4N}^{\Phi_0} & -C_{4N}^{\Phi_1}+2C^{2k_0\Phi_1}C_{4N}^{\Phi_1} & \cdots & -C_{4N}^{\Phi_{N-1}}+2C_{4N}^{2k_0\Phi_{N-1}}C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{\Phi_0}-2C_{4N}^{2k_0\Phi_0}C_{4N}^{\Phi_0}+2C_{4N}^{2k_1\Phi_0}C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1}-2C_{4N}^{2k_0\Phi_1}C_{4N}^{\Phi_1}+2C_{4N}^{2k_1\Phi_1}C_{4N}^{\Phi_1} & \cdots & \cdots \\ \vdots & & \cdots & \vdots \end{bmatrix}$$

(A-4)

Since $k_0=1$, we get

$$-C_{4N}^{\Phi_m}+2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m}=-C_{4N}^{(2k_0-1)\Phi_m}+2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m}=C_{4N}^{(2k_0+1)\Phi_m}$$

(A-5)

and

$$C_{4N}^{\Phi_m}-2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m}=-(-C_{4N}^{(2k_0-1)\Phi_m}+2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m})=-C_{4N}^{(2k_0+1)\Phi_m} \tag{A-6}$$

In case of $k_i=i+1$, we have $(2k_{i-1}+1)\Phi_m=(2(k_i-1)+1)\Phi_m=(2k_i-1)\Phi_m$, then we get

$$\begin{aligned}C_{4N}^{\Phi_m}-2C_{4N}^{2k_{i-1}\Phi_m}C_{4N}^{\Phi_m}+2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}&=-C_{4N}^{(2k_{i-1}+1)\Phi_m}+2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}\\&=-C_{4N}^{(2k_i-1)\Phi_m}+2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}=C_{4N}^{(2k_i+1)\Phi_m}\end{aligned} \tag{A-7}$$

Taking the (A-5)-(A-7) to (A-4), we can rewrite that

$$\begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & \cdots & C_{4N}^{\Phi_{N-1}} \\ -C_{4N}^{\Phi_0}+2C_{4N}^{2k_0\Phi_0}C_{4N}^{\Phi_0} & -C_{4N}^{\Phi_1}+2C^{2k_0\Phi_1}C_{4N}^{\Phi_1} & \cdots & -C_{4N}^{\Phi_{N-1}}+2C_{4N}^{2k_0\Phi_{N-1}}C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{\Phi_0}-2C_{4N}^{2k_0\Phi_0}C_{4N}^{\Phi_0}+2C_{4N}^{2k_1\Phi_0}C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1}-2C_{4N}^{2k_0\Phi_1}C_{4N}^{\Phi_1}+2C_{4N}^{2k_1\Phi_1}C_{4N}^{\Phi_1} & \cdots & \cdots \\ \vdots & & \cdots & \vdots \end{bmatrix}$$
$$=\begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & \cdots & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & \cdots & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ C_{4N}^{(2k_1+1)\Phi_0} & C_{4N}^{(2k_1+1)\Phi_1} & \cdots & C_{4N}^{(2k_1+1)\Phi_{N-1}} \\ \vdots & & & \vdots \end{bmatrix}=[B]_N \tag{A-8}$$

Proof for $[B]_N=[K]_N[C]_N[D]_N$ is completed.

References

[1] K.R. Rao, J.J. Hwang, *Techniques & Standards for Image Video & Audio Coding*, Prentice Hall, 1996.

[2] M. H. Lee, and Ken Finlayson, "A simple element inverse Jacket transform coding," *IEEE Information Theory Workshop* 2005, ITW 2005, Rotorua, New Zealand, 29 Aug 1st Sept. 2005.

[3] M. H. Lee, B. S. Rajan, and Ju Yong Park, "A Generalized Reverse Jacket Transform," *IEEE Trans. Circuits Syst. II*, vol.48, no. 7, pp.684-690, July, 2001.

[4] D. Park, M.H. Lee, and Euna Choi, "Revisited DFT matrix via the reverse jacket transform and its application to communication," *The 22nd symposium on Information theory and its applications* (SITA 99), Yuzawa, Niigata, Japan, Nov.30-Dec.3, 1999.

[5] M.H. Lee, "The Center Weighted Hadamard Transform," *IEEE Trans. On Circuits and Systems*, vol. 36, no.9, pp.1247~1249, Sep. 1989

[6] S. R. Lee, J. H. Yi, "Fast Reverse Jacket Transform as an Altenative Representation of N point Fast Fourier Transform," *Journal of Mathematical Imaging and Vision*, KL1419-03, pp.1413-1420, Nov. 2001.

[7] Hsieh S. Hou, A Fast Recursive Algorithm For Computing the Discrete Cosine Transform, *IEEE Trans. on ASSP*, Oct. 1987

[8] W.H. Chen et al. , 'A fast computational algorithm for the discrete cosine transform," *IEEE Trans. COM.*, Vol.. 25, No. 9, pp. 1004-1008, 1977

[9] Z. Wang, "Fast Algorithms for the Discrete W Transform for the Discrete Fourier Transform," *IEEE Trans. ASSP*, Vol. 32, No.4, pp. 803-816, 1984

[10] Y-J Chuang and J-L Wu, "An Efficient Matrix Based 2-D DCT Splitter and Merger for SIMD Instructions," *IEICE Trans. INF & SYST.* Vol. E88-D, No. 7, July, 2005

Dae-chul Park

He received the B.S. degree in Ele ctronics Engineering from Sogang U niversity, Korea, in 1977, the M.S. Degree and the Ph.D. degree in El ectrical and Computer Engineering from the University of New Mexico, Albuquerque, NM, USA, in 1985 and 1989, respectively. Since 1989 he had worked in ETRI, Korea as a member of senior technical staff til 1993. Since 1993, he has been with the Department of Information and Com munication Engineering, Hannam University, Daejeon, where he is a currently a professor. His research interests include image processing, mobile multimedia communication and signal processing algorithm development.