

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Orthogonal Discrete Fourier and Cosine Matrices for Signal Processing

Daechul Park¹ and Moon Ho Lee²
¹Hannam University,
²Chonbuk National University
 Korea

1. Introduction

The A DFT (Discrete Fourier Transform) has been studied and applied to signal processing and communication theory. The relation between the Fourier matrix and the Hadamard transform was developed in [Ahmed & Rao, 1975; Whelchel & Guinn, 1968] for signal representation and classification and the Fast Fourier-Hadamard Transform (FFHT) was proposed. This idea was further investigated in [Lee & Lee, 1998] as an extension of the conventional Hadamard matrix. Lee et al [Lee & Lee, 1998] has proposed the Reverse Jacket Transform (RJT) based on the decomposition of the Hadamard matrix into the Hadamard matrix (unitary matrix) itself and a sparse matrix. Interestingly, the Reverse Jacket (RJ) matrix has a strong geometric structure that reveals a circulant expansion and contraction properties from a basic 2x2 sparse matrix.

The discrete Fourier transform (DFT) is an orthogonal matrix with highly practical value for representing signals and images [Ahmed & Rao, 1975; Lee, 1992; Lee, 2000]. Recently, the Jacket matrices which generalize the weighted Hadamard matrix were introduced in [Lee, 2000], [Lee & Kim, 1984, Lee, 1989, Lee & Yi, 2001; Fan & Yang, 1998]. The Jacket matrix¹ is an abbreviated name of a reverse Jacket geometric structure. It includes the conventional Hadamard matrix [Lee, 1992; Lee, 2000; Lee et al., 2001; Hou et al., 2003], but has the weights, ω , that are j or 2^k , where k is an integer, and $j = \sqrt{-1}$, located in the central part of Hadamard matrix. The weighted elements' positions of the forward matrix can be replaced by the non-weighted elements of its inverse matrix and the signs of them do not change between the forward and inverse matrices, and they are only as element inverse and transpose. This reveals an interesting complementary matrix relation.

Definition 1: If a matrix $[J]_m$ of size $m \times m$ has nonzero elements

$$[J]_m = \begin{bmatrix} j_{0,0} & j_{0,1} & \cdots & j_{0,m-1} \\ j_{1,0} & j_{1,1} & \cdots & j_{1,m-1} \\ \vdots & \vdots & & \vdots \\ j_{m-1,0} & j_{m-1,1} & \cdots & j_{m-1,m-1} \end{bmatrix}, \quad (6-1)$$

$$[J]_m^{-1} = \frac{1}{C} \begin{bmatrix} 1/j_{0,0} & 1/j_{0,1} & \dots & 1/j_{0,m-1} \\ 1/j_{1,0} & 1/j_{1,1} & \dots & 1/j_{1,m-1} \\ \vdots & \vdots & & \vdots \\ 1/j_{m-1,0} & 1/j_{m-1,1} & \dots & 1/j_{m-1,m-1} \end{bmatrix}^T \quad (6-2)$$

where C is the normalizing constant, and T is of matrix transposition, then the matrix $[J]_m$ is called a Jacket matrix [Whelchel & Guinn, 1968],[Lee et al., 2001],[Lee et al, 2008; Chen et al., 2008]. Especially orthogonal matrices, such as Hadamard, DFT, DCT, Haar, and Slant matrices belong to the Jacket matrices family [Lee et al., 2001]. In addition, the Jacket matrices are associated with many kind of matrices, such as unitary matrices, and Hermitian matrices which are very important in communication (e.g., encoding), mathematics, and physics.

In section 2 DFT matrix is revisited in the sense of sparse matrix factorization. Section 3 presents recursive factorization algorithms of DFT and DCT matrix for fast computation. Section 4 proposes a hybrid architecture for implementation of algorithms simply by adding a switching device on a single chip module. Lastly, conclusions were drawn in section 4.

2. Preliminary of DFT presentation

The discrete Fourier transform (DFT) is a Fourier representation of a given sequence $x(m)$, $0 \leq m \leq N - 1$ and is defined as

$$X(n) = \sum_{m=0}^{N-1} x(m)W^{nm}, \quad 0 \leq n \leq N - 1, \quad (6-3)$$

where $W = e^{-j\frac{2\pi}{N}}$. Let's denote N -point DFT matrix as $F_N = [W^{nm}]_N$, $n, m = \{0, 1, 2, \dots, N - 1\}$,

where $W = e^{-j\frac{2\pi}{N}}$ (see about DFT in appendix), and the $N \times N$ Sylvester Hadamard matrix as $[H]_N$, respectively. The Sylvester Hadamard matrix is generated recursively by successive Kronecker products,

$$[H]_N = [H]_2 \otimes [H]_{N/2}, \quad (6-4)$$

for $N=4, 8, 16, \dots$ and $[H]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. For the remainder of this chapter, analysis will be

concerned only with $N=2^k$, $k=1, 2, 3, \dots$ as the dimensionality of both the F and H matrices.

Definition 2: A sparse matrix $[S]_N$, which relates $[F]_N$ and $[H]_N$, can be computed from the factorization of F based on H .

The structure of the S matrix is rather obscure. However, a much less complex and more appealing relationship will be identified for S [Park et al., 1999].

To illustrate the DFT using direct product we alter the denotation of W to lower case $w = e^{-j2\pi}$, so that $w^{n/N}$ becomes the n -th root of unit for N -point W . For instance, the DFT matrix of dimension 2 is given by:

$$[F]_2 = \begin{bmatrix} w^{0/2} & w^{0/2} \\ w^{0/2} & w^{1/2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [H]_2 . \tag{6-5}$$

Let's define

$$[W]_2 = \begin{bmatrix} w^{0/4} & 0 \\ 0 & w^{1/4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix} \text{ and } [E]_2 = [F]_2 [W]_2 = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix} , \tag{6-6}$$

and in general, We define

$$[W]_N = \text{diag}(w^{0/2N}, w^{1/2N}, w^{2/2N}, \dots, w^{(N-1)/2N})$$

and

$$[E]_N = [F]_N [W]_N = [P]_N^T [\tilde{F}]_N [W]_N . \tag{6-7}$$

where $[P]_N$ is a permutation matrix and $[\tilde{F}]_N = [P]_N [F]_N$ is a permuted version of DFT matrix $[F]_N$.

3. A sparse matrix factorization of orthogonal transforms

3.1 A sparse matrix analysis of discrete Fourier transform

Now we will present the Jacket matrix from a direct product of a sparse matrix computation and representation given by [Lee, 1989], [Lee & Finlayson, 2007]

$$[J]_m = \frac{1}{m} [H]_m [S]_m , \tag{6-8}$$

where $m = 2^{k+1}, k \in \{1, 2, 3, 4, \dots\}$ and $[S]_m$ is sparse matrix of $[J]_m$. Thus the inverse of the Jacket matrix can be simply written as

$$([J]_m)^{-1} = ([S]_m)^{-1} [H]_m . \tag{6-9}$$

As mentioned previously, the DFT matrix is also a Jacket matrix. By considering the sparse matrix for the 4-point DFT matrix $[F]_4$,

$$[F]_4 = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-j\frac{\pi}{2}} & e^{-j\frac{\pi}{2} \times 2} & e^{-j\frac{\pi}{2} \times 3} \\ 1 & e^{-j\frac{\pi}{2} \times 2} & e^{-j\frac{\pi}{2} \times 4} & e^{-j\frac{\pi}{2} \times 6} \\ 1 & e^{-j\frac{\pi}{2} \times 3} & e^{-j\frac{\pi}{2} \times 6} & e^{-j\frac{\pi}{2} \times 9} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} .$$

we can rewrite $[F]_4$ by using permutations as

$$\begin{aligned}
 [\tilde{F}]_4 = [\text{Pr}]_4 [F]_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \\ 1 & j & -1 & -j \end{bmatrix} = \begin{bmatrix} [\tilde{F}]_2 & [\tilde{F}]_2 \\ E_2 & -E_2 \end{bmatrix} \\
 &= \begin{pmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{pmatrix} \begin{bmatrix} [\tilde{F}]_2 & 0 \\ 0 & E_2 \end{bmatrix}^T, \tag{6-10}
 \end{aligned}$$

where $E_2 = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}$, its inverse matrix is from element-inverse, such that

$$(E_2)^{-1} = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix} = \begin{pmatrix} 1/1 & -1/j \\ 1/1 & 1/j \end{pmatrix}^T. \tag{6-11}$$

In general, we can write that

$$[\tilde{F}]_N = [\text{Pr}]_N [F]_N = \begin{bmatrix} [\tilde{F}]_{N/2} & [\tilde{F}]_{N/2} \\ E_{N/2} & -E_{N/2} \end{bmatrix} = \begin{pmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{pmatrix} \begin{bmatrix} [\tilde{F}]_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix}^T, \tag{6-12}$$

where $[F]_2 = [\tilde{F}]_2$. And the submatrix E_N could be written from (6-7) by

$$[E]_N = [F]_N [W]_N = [\text{Pr}]_N [\tilde{F}]_N [W]_N, \tag{6-13}$$

where $[W]_N = \begin{bmatrix} W^0 & 0 & \dots & 0 \\ 0 & W^1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & W^{N-1} \end{bmatrix}$, and W is the complex unit for $2N$ point DFT matrix.

For example, $[E]_2 = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}$ can be calculated by using

$$[E]_2 = [\text{Pr}]_2 [\tilde{F}]_2 [W]_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} W^0 & 0 \\ 0 & W^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix} = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}. \tag{6-14}$$

By using the results from (6-12) and (6-13), we have a new DFT matrix decomposition as

$$\begin{aligned}
 [\tilde{F}]_N &= [\text{Pr}]_N [F]_N = \begin{pmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{pmatrix} \begin{bmatrix} [\tilde{F}]_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix}^T \\
 &= \begin{bmatrix} [\tilde{F}]_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 [\tilde{F}]_N &= \begin{bmatrix} [\tilde{F}]_{N/2} & 0 \\ 0 & \text{Pr}_{N/2}[\tilde{F}]_{N/2}W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\
 &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & \text{Pr}_{N/2} \end{bmatrix} \begin{bmatrix} [\tilde{F}]_{N/2} & 0 \\ 0 & [\tilde{F}]_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}. \tag{6-15}
 \end{aligned}$$

Finally, based on the recursive form we have

$$\begin{aligned}
 [F]_N &= ([\text{Pr}]_N)^{-1} [\tilde{F}]_N = ([\text{Pr}]_N)^T \begin{bmatrix} I_{N/2} & 0 \\ 0 & \text{Pr}_{N/2} \end{bmatrix} \begin{bmatrix} [\tilde{F}]_{N/2} & 0 \\ 0 & [\tilde{F}]_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\
 &= ([\text{Pr}]_N)^T \begin{bmatrix} I_{N/2} & 0 \\ 0 & \text{Pr}_{N/2} \end{bmatrix} \cdots \begin{bmatrix} I_{N/4} \otimes I_2 & 0 \\ 0 & \text{Pr}_2 \end{bmatrix} \begin{bmatrix} I_{N/2} \otimes F_2 \\ I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & W_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}. \tag{6-16}
 \end{aligned}$$

Using (6-16) butterfly data flow diagram for DFT transform is drawn from left to right to perform $X = [F]_N \mathbf{x}$.

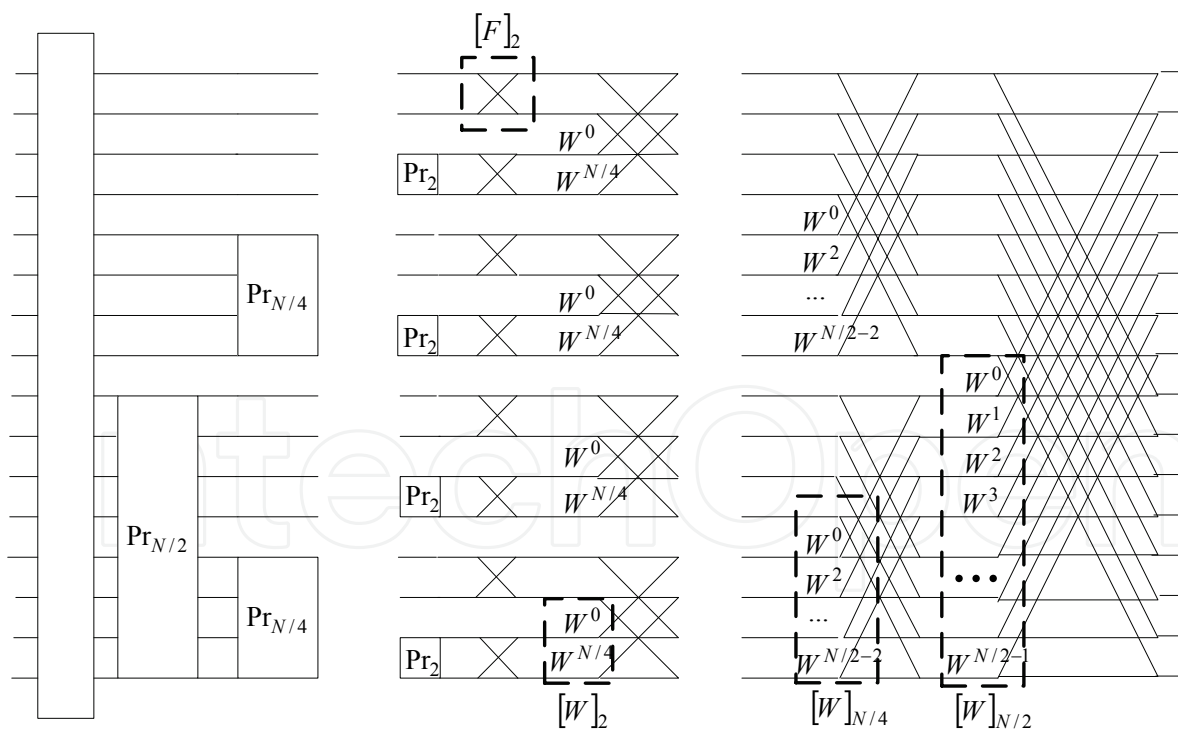


Fig. 1. Butterfly data flow diagram of proposed DFT matrix with order N

3.2 A Sparse matrix analysis of discrete cosine transform

Similar to the section 3.1, we will present the DCT matrices by using the element inverse or block inverse Jacket like sparse matrix [Lee, 2000; Park et al., 1999] decomposition. In this

section, the simple construction and fast computation for forward and inverse calculations and analysis of the sparse matrices, was very useful for developing the fast algorithms and orthogonal codes design.

Discrete Cosine Transform (DCT) is widely used in image processing, and orthogonal transform. There are four typical DCT matrices [Rao & Yip, 1990; Rao & Hwang, 1996],

$$\text{DCT - I: } [C_{N+1}^I]_{m,n} = \sqrt{\frac{2}{N}} k_m k_n \cos \frac{mn\pi}{N}, \quad m, n = 0, 1, \dots, N; \quad (6-17)$$

$$\text{DCT - II: } [C_N^{II}]_{m,n} = \sqrt{\frac{2}{N}} k_m \cos \frac{m(n+\frac{1}{2})\pi}{N}, \quad m, n = 0, 1, \dots, N-1; \quad (6-18)$$

$$\text{DCT - III: } [C_N^{III}]_{m,n} = \sqrt{\frac{2}{N}} k_n \cos \frac{(m+\frac{1}{2})n\pi}{N}, \quad m, n = 0, 1, \dots, N-1; \quad (6-19)$$

$$\text{DCT - IV: } [C_N^{IV}]_{m,n} = \sqrt{\frac{2}{N}} \cos \frac{(m+\frac{1}{2})(n+\frac{1}{2})\pi}{N}, \quad m, n = 0, 1, \dots, N-1, \quad (6-20)$$

where

$$k_j = \begin{cases} 1, & j = 1, 2, \dots, N-1 \\ \frac{1}{\sqrt{2}}, & j = 0, N \end{cases}.$$

To describe the computations of DCT, in this chapter, we will focus on the DCT - II algorithm, and introduce the sparse matrix decomposition and fast computations.

The 2-by-2 DCT - II matrix can be simply written as

$$[C]_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C_4^3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}}, \quad (6-21)$$

where $1/\sqrt{2}$ 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 is of the form

$$[C]_4 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^4 & C_8^8 \\ C_8^3 & C_8^7 & C_8^8 & C_8^5 \end{bmatrix}, \quad (6-22)$$

we can write

$$[P]_4[C]_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix}, \quad (6-23)$$

where $[P]_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is permutation matrix. $[P]_N$ permutation matrix is a special case

which has the form

$$[P]_2 = [I]_2, \text{ and } [P]_N = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & & 0 & 1 & & 0 \\ 0 & 0 & 1 & & 0 & 0 & & 0 \\ 0 & 0 & 0 & & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 1 \end{bmatrix}, \quad N \geq 4,$$

where

$$[P]_N = [pr_{i,j}]_N,$$

with

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

where $i, j \in \{0, 1, \dots, N - 1\}$.

Since $C_8^1 = -C_8^7, C_8^2 = -C_8^6, C_8^3 = -C_8^5$, we rewrite (6-23) as

$$[P]_4[C]_4 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & -C_8^2 & -C_8^2 & C_8^2 \\ C_8^1 & C_8^3 & -C_8^3 & -C_8^1 \\ C_8^3 & -C_8^1 & C_8^1 & -C_8^3 \end{bmatrix}, \quad (6-24)$$

and let us define a column permutation matrix $[Pc]_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, and $[Pc]_N$ is a

reversible permutation matrix which is defined by

$$[Pc]_2 = [I]_2, \text{ and}$$

$$[Pc]_N = \begin{bmatrix} I_{N/4} & 0 & 0 & 0 \\ 0 & I_{N/4} & 0 & 0 \\ 0 & 0 & 0 & I_{N/4} \\ 0 & 0 & I_{N/4} & 0 \end{bmatrix}, N \geq 4.$$

Thus we have

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

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & -C_8^2 & C_8^2 & -C_8^2 \\ C_8^1 & C_8^3 & -C_8^1 & -C_8^3 \\ C_8^3 & -C_8^1 & -C_8^3 & C_8^1 \end{bmatrix} = \begin{bmatrix} C_2 & C_2 \\ B_2 & -B_2 \end{bmatrix}, \quad (6-25)$$

where $[C]_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$, the 2-by-2 DCT - II matrix and $[B]_2 = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix}$. Thus we can write that

$$[P]_4 [C]_4 [Pc]_4 = \begin{bmatrix} C_2 & C_2 \\ B_2 & -B_2 \end{bmatrix} = \left(\begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \begin{bmatrix} C_2 & 0 \\ 0 & B_2 \end{bmatrix} \right)^T, \quad (6-26)$$

it is clear that $\begin{bmatrix} C_2 & 0 \\ 0 & B_2 \end{bmatrix}$ is a block inverse matrix, which has

$$\begin{bmatrix} C_2 & 0 \\ 0 & B_2 \end{bmatrix}^{-1} = \begin{bmatrix} (C_2)^{-1} & 0 \\ 0 & (B_2)^{-1} \end{bmatrix}. \quad (6-27)$$

The (6-27) is Jacket -like sparse matrix with block inverse.

In general the permuted DCT - II matrix $[\tilde{C}]_N$ can be constructed recursively by using

$$[\tilde{C}]_N = [P]_N [C]_N [Pc]_N = \begin{bmatrix} C_{N/2} & C_{N/2} \\ B_{N/2} & -B_{N/2} \end{bmatrix} = \left(\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \right)^T \quad (6-28)$$

where $[C]_{N/2}$ denotes the $\frac{N}{2} \times \frac{N}{2}$ DCT - II matrix, and $[B]_{N/2}$ can be calculated by using

$$[B]_{N/2} = \left[\left(C_{2N}^{f(m,n)} \right)_{m,n} \right]_{N/2}, \quad (6-29)$$

where

$$\begin{cases} f(m,1) = 2m - 1, \\ f(m,n+1) = f(m,n) + f(m,1) \times 2, \end{cases} \quad m,n \in \{1,2,\dots,N/2\}. \quad (6-30)$$

For example, in the 4-by-4 permuted DCT - II matrix $[\tilde{C}]_4$, B_2 could be calculated by using $f(1,1) = 1$, $f(2,1) = 3$, $f(1,2) = f(1,1) + f(1,1) \times 2 = 3$, and $f(2,2) = f(2,1) + f(2,1) \times 2 = 9$,

$$[B]_2 = \left[\left(C_8^{f(m,n)} \right)_{m,n} \right]_4 = \begin{bmatrix} \left(C_8^{f(1,1)} \right)_{1,1} & \left(C_8^{f(1,2)} \right)_{1,2} \\ \left(C_8^{f(2,1)} \right)_{2,1} & \left(C_8^{f(2,2)} \right)_{2,2} \end{bmatrix}. \quad (6-31)$$

and its inverse is of $[\tilde{C}]_N$ can be simply computed from the block inverse

$$\begin{aligned} ([P]_N [C]_N [Pc]_N)^{-1} &= \left(\left(\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \right)^T \right)^{-1} \\ &= \left(\begin{bmatrix} (C_{N/2})^{-1} & 0 \\ 0 & (B_{N/2})^{-1} \end{bmatrix} \begin{bmatrix} \frac{2}{N} I_{N/2} & \frac{2}{N} I_{N/2} \\ \frac{2}{N} I_{N/2} & -\frac{2}{N} I_{N/2} \end{bmatrix} \right)^T \\ &= \frac{2}{N} \left(\begin{bmatrix} (C_{N/2})^{-1} & 0 \\ 0 & (B_{N/2})^{-1} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \right)^T \end{aligned} \quad (6-32)$$

For example, the 8×8 DCT - II matrix has

$$[C]_8 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{16}^1 & C_{16}^3 & C_{16}^5 & C_{16}^7 & C_{16}^9 & C_{16}^{11} & C_{16}^{13} & C_{16}^{15} \\ C_{16}^2 & C_{16}^6 & C_{16}^{10} & C_{16}^{14} & C_{16}^{18} & C_{16}^{22} & C_{16}^{26} & C_{16}^{30} \\ C_{16}^3 & C_{16}^9 & C_{16}^{15} & C_{16}^{21} & C_{16}^{27} & C_{16}^{33} & C_{16}^{39} & C_{16}^{45} \\ C_{16}^4 & C_{16}^{12} & C_{16}^{20} & C_{16}^{28} & C_{16}^{36} & C_{16}^{44} & C_{16}^{52} & C_{16}^{60} \\ C_{16}^5 & C_{16}^{15} & C_{16}^{25} & C_{16}^{35} & C_{16}^{45} & C_{16}^{55} & C_{16}^{65} & C_{16}^{75} \\ C_{16}^6 & C_{16}^{18} & C_{16}^{30} & C_{16}^{42} & C_{16}^{54} & C_{16}^{66} & C_{16}^{78} & C_{16}^{90} \\ C_{16}^7 & C_{16}^{21} & C_{16}^{35} & C_{16}^{49} & C_{16}^{63} & C_{16}^{77} & C_{16}^{91} & C_{16}^{105} \end{bmatrix},$$

and it can be represented by

$$[P]_8 [C]_8 [Pc]_8 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{16}^4 & -C_{16}^4 & C_{16}^4 & -C_{16}^4 & C_{16}^4 & -C_{16}^4 & C_{16}^4 & -C_{16}^4 \\ C_{16}^2 & C_{16}^6 & -C_{16}^2 & -C_{16}^6 & C_{16}^2 & C_{16}^6 & -C_{16}^2 & -C_{16}^6 \\ C_{16}^6 & -C_{16}^2 & -C_{16}^6 & C_{16}^2 & C_{16}^6 & -C_{16}^2 & -C_{16}^6 & C_{16}^2 \\ C_{16}^1 & C_{16}^3 & -C_{16}^7 & -C_{16}^5 & -C_{16}^1 & -C_{16}^3 & C_{16}^7 & C_{16}^5 \\ C_{16}^5 & -C_{16}^1 & -C_{16}^3 & -C_{16}^7 & -C_{16}^5 & C_{16}^1 & C_{16}^3 & C_{16}^7 \\ C_{16}^3 & -C_{16}^7 & C_{16}^5 & C_{16}^1 & -C_{16}^3 & C_{16}^7 & -C_{16}^5 & -C_{16}^1 \\ C_{16}^7 & -C_{16}^5 & C_{16}^1 & -C_{16}^3 & -C_{16}^7 & C_{16}^5 & -C_{16}^1 & C_{16}^3 \end{bmatrix} = \begin{bmatrix} C_4 & C_4 \\ B_4 & -B_4 \end{bmatrix}.$$

Additionally, it is clearly that the function (6-28) also can be recursively constructed by using different permutations matrices $[\tilde{P}]_N$ and $[\tilde{Pc}]_N$, as

$$[\tilde{P}]_N [\tilde{C}]_N [\tilde{Pc}]_N = \begin{bmatrix} \tilde{C}_{N/2} & \tilde{C}_{N/2} \\ \tilde{B}_{N/2} & -\tilde{B}_{N/2} \end{bmatrix} = \left(\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} \tilde{C}_{N/2} & 0 \\ 0 & \tilde{B}_{N/2} \end{bmatrix} \right)^T, \quad (6-33)$$

where $[\tilde{C}]_{N/2}$ and $[\tilde{B}]_{N/2}$ are the permuted cases of $[C]_{N/2}$ and $[B]_{N/2}$, respectively. The new permutation matrices have the form

$$[\tilde{P}]_N = \begin{bmatrix} [P]_{N/2} & 0 \\ 0 & I_{N/2} \end{bmatrix}, \text{ and } [\tilde{Pc}]_N = \begin{bmatrix} [Pc]_{N/2} & 0 \\ 0 & [Pc]_{N/2} \end{bmatrix}, \quad N > 4. \quad (6-34)$$

Easily, we can check that

$$\begin{aligned} [\tilde{P}]_N [\tilde{C}]_N [\tilde{Pc}]_N &= \begin{bmatrix} P_{N/2} & 0 \\ 0 & I_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & C_{N/2} \\ B_{N/2} & -B_{N/2} \end{bmatrix} \begin{bmatrix} Pc_{N/2} & 0 \\ 0 & Pc_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} P_{N/2} C_{N/2} & P_{N/2} C_{N/2} \\ I_{N/2} B_{N/2} & -I_{N/2} B_{N/2} \end{bmatrix} \begin{bmatrix} Pc_{N/2} & 0 \\ 0 & Pc_{N/2} \end{bmatrix} \end{aligned}$$

$$[\tilde{P}]_N [\tilde{C}]_N [\tilde{Pc}]_N = \begin{bmatrix} P_{N/2} C_{N/2} P_{C_{N/2}} & P_{N/2} C_{N/2} P_{C_{N/2}} \\ I_{N/2} B_{N/2} P_{C_{N/2}} & -I_{N/2} B_{N/2} P_{C_{N/2}} \end{bmatrix} = \begin{bmatrix} \tilde{C}_{N/2} & \tilde{C}_{N/2} \\ \tilde{B}_{N/2} & -\tilde{B}_{N/2} \end{bmatrix}, \tag{6-35}$$

where $[\tilde{B}]_{N/2} = [B]_{N/2} [Pc]_{N/2}$.

For example, the 4-by-4 DCT - II case has

$$[\tilde{B}]_{4/2} = [B]_{4/2} [Pc]_{4/2} = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix}. \tag{6-36}$$

Moreover, the matrix $[B]_2 = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix}$ can be decomposed by using the 2-by-2 DCT - II matrix as

$$[B]_2 = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix} = [K]_2 [C]_2 [D]_2 = \begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & 2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & -C_8^2 \end{bmatrix} \begin{bmatrix} C_8^1 & 0 \\ 0 & C_8^3 \end{bmatrix}, \tag{6-37}$$

where $[K]_2 = \begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & 2 \end{bmatrix}$ is a upper triangular matrix, $[D]_2 = \begin{bmatrix} C_8^1 & 0 \\ 0 & C_8^3 \end{bmatrix}$ is a diagonal matrix, and we use the cosine related function

$$\cos(2k + 1)\phi_m = 2 \cos(2k\phi_m) \cos \phi_m - \cos(2k - 1)\phi_m. \tag{6-38}$$

where ϕ_m is m -th angle.

In a general case, we have

$$[B]_N = [K]_N [C]_N [D]_N, \tag{6-39}$$

where

$$[K]_N = \begin{bmatrix} \sqrt{2} & 0 & 0 & \dots \\ -\sqrt{2} & 2 & 0 & \dots \\ \sqrt{2} & -2 & 2 & \dots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}, [D]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \dots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix}, \tag{6-40}$$

and $\Phi_i = 2i + 1, i \in \{0, 1, 2, \dots, N - 1\}$. See appendix 2 for proof of (6-39).

By using the results from (6-28) and (6-39), we have a new form for DCT - II matrix

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

$$\begin{aligned}
 [\tilde{C}]_N &= \begin{bmatrix} C_{N/2} & 0 \\ 0 & K_{N/2}C_{N/2}D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\
 &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}.
 \end{aligned}
 \tag{6-41}$$

Given the recursive form of (6-41), we can write

$$\begin{aligned}
 [C]_N &= ([Pr]_N)^{-1} \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \dots \begin{bmatrix} I_{N/4} \otimes ([Pr]_4)^{-1} \\ I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} \otimes C_2 \\ I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/4} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \\ I_{N/4} \otimes ([Pc]_4)^{-1} \end{bmatrix} \\
 &\dots \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} ([Pc]_N)^{-1}
 \end{aligned}
 \tag{6-42}$$

By taking all permutation matrices outside, we can rewrite (6-42) as

$$\begin{aligned}
 [C]_N &= ([\tilde{Pr}]_N)^{-1} \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \dots \begin{bmatrix} I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \\ I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} \otimes C_2 \\ I_{N/4} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} ([\tilde{Pc}]_N)^{-1}.
 \end{aligned}
 \tag{6-43}$$

Using (6-43) butterfly data flow diagram for DCT-II transform is drawn as Fig.2 from left to right to perform $X=[C]_N x$.

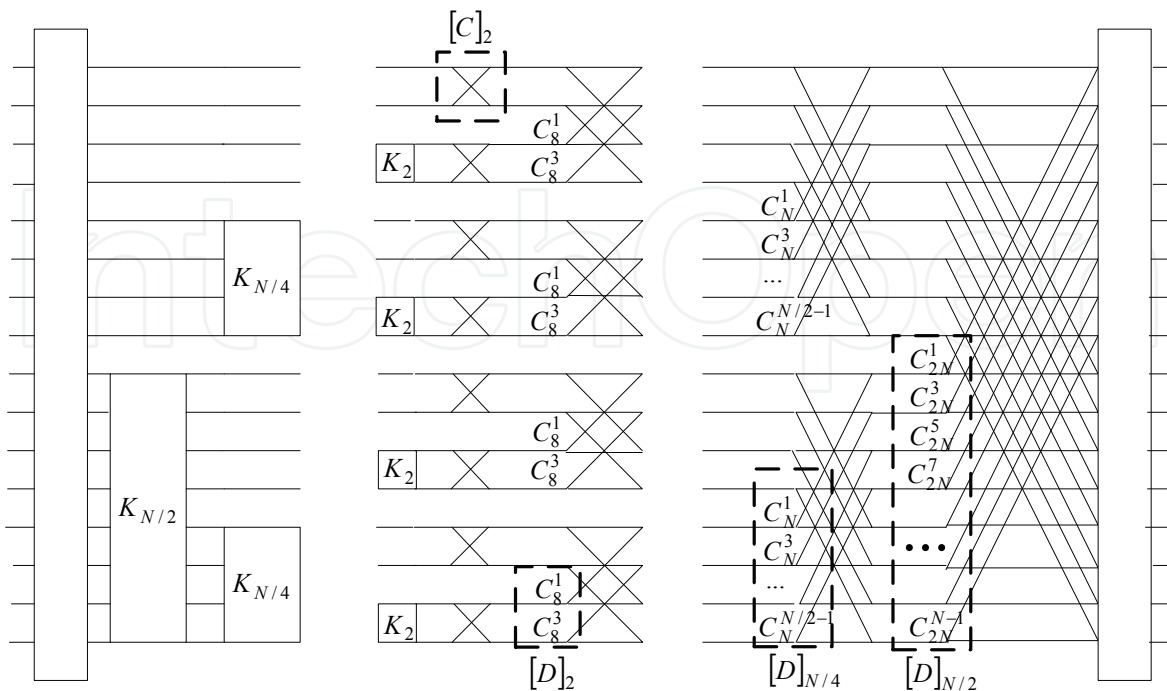


Fig. 2. Butterfly data flow diagram of proposed DCT - II matrix with order N

3.3 Hybrid DCT/DFT architecture on element inverse matrices

It is clear that the form of (6-43) is the same as that of (6-16), where we only need change K_l to Pr_l and D_l to W_l , with $l \in \{2, 4, 8, \dots, N/2\}$. Consequently, the results show that the DCT - II and DFT can be unified by using same algorithm and architecture within some characters changed. As illustrated in Fig.1, and Fig.2, we find that the DFT calculations can be obtained from the architecture of DCT by replacing the $[D]_N$ to $[W]_N$, and a permutation matrix $[Pr]_N$ to $[K]_N$. Hence a unified function block diagram for DCT/DFT hybrid architecture algorithm can be drawn as Fig.3. In this figure, we can joint DCT and DFT in one chip or one processing architecture, and use one switching box to control the output data flow. It will be useful to developing the unified chip or generalized form for DCT and DFT together.

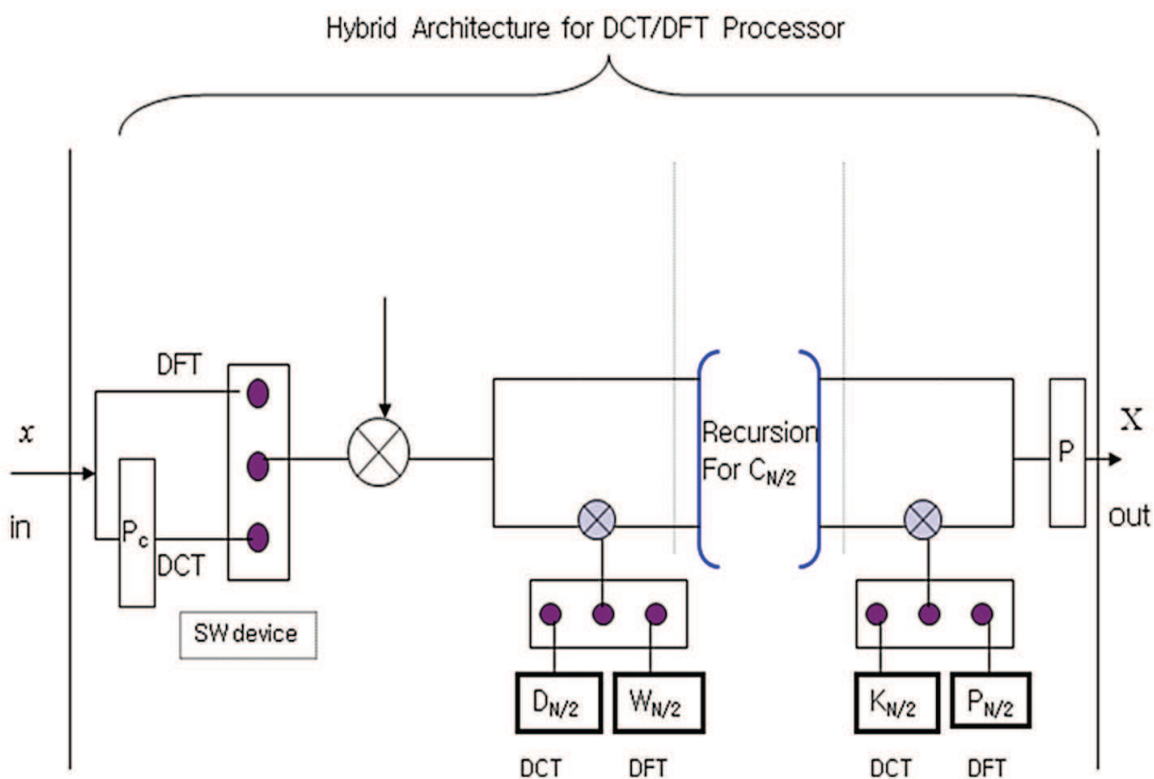


Fig. 3. A unified function block diagram for proposed DCT/DFT hybrid architecture algorithm

4. Conclusion

We propose a new representation of DCT/DFT matrices via the Jacket transform based on the block-inverse processing. Following on the factorization method of the Jacket transform, we show that the inverse cases of DCT/DFT matrices are related to their block inverse sparse matrices and the permutations. Generally, DCT/DFT can be represented by using the same architecture based on element inverse or block inverse decomposition. Linking between two transforms was derived based on matrix recursion formula.

Discrete Cosine Transform (DCT) has applications in signal classification and representation, image coding, and synthesis of video signals. The DCT-II is a popular structure and it is usually accepted as the best suboptimal transformation that its

performance is very close to that of the statistically optimal Karhunen-Loeve transform for picture coding. Further, the discrete Fourier transform (DFT) is also a popular algorithm for signal processing and communications, such as OFDM transmission and orthogonal code designs. Being combined these two different transforms, a unified fast processing module to implement DCT/DFT hybrid architecture algorithm can be designed by adding switching device to control either DCT or DFT processing depending on mode of operation.

Further investigation is needed for unified treatment of recursive decomposition of orthogonal transform matrices exploiting the properties of Jacket-like sparse matrix architecture for fast trigonometric transform computation.

5. Acknowledgement

This work was supported by World Class Univ. R32-2008-000-20014-0, NRF, Korea.

6. Appendix

6.1 Appendix 1

The DFT matrix brings higher powers of w , and the problem turns out to be

$$\begin{bmatrix} 1 & 1 & 1 & \cdot & 1 \\ 1 & w & w^2 & \cdot & w^{n-1} \\ 1 & w^2 & w^4 & \cdot & w^{2(n-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{n-1} & w^{2(n-1)} & \cdot & w^{(n-1)^2} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \cdot \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \cdot \\ y_{n-1} \end{bmatrix}. \quad (\text{A-1})$$

and the inverse form

$$F^{-1} = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & \cdot & 1 \\ 1 & w^{-1} & w^{-2} & \cdot & w^{-(n-1)} \\ 1 & w^{-2} & w^{-4} & \cdot & w^{-2(n-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{-(n-1)} & w^{-2(n-1)} & \cdot & w^{-(n-1)^2} \end{bmatrix}. \quad (\text{A-2})$$

Then, we can define the Fourier matrix as follows.

Definition A.1: An $n \times n$ matrix $F = [a_{ij}]$ is a Fourier matrix if

$$a_{ij} = w^{(i-1)(j-1)}, w = e^{2\pi i/n}, \text{ and } i, j \in \{1, 2, \dots, n\}. \quad (\text{A-3})$$

and the inverse form $F^{-1} = \frac{1}{n} [(a_{ij})^{-1}] = \frac{1}{n} [w^{-(i-1)(j-1)}]$.

For example, in the cases $n = 2$ and $n = 3$, and inverse is an element-wise inverse like Jacket matrix, then, we have

$$F_2 = \begin{bmatrix} w^0 & w^0 \\ w^0 & w^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, F_2^{-1} = \frac{1}{2} \begin{bmatrix} w^0 & w^0 \\ w^0 & w^{-1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

and

$$F_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{i2\pi}{3}} & e^{\frac{i4\pi}{3}} \\ 1 & e^{\frac{i4\pi}{3}} & e^{\frac{i8\pi}{3}} \end{bmatrix}, F_3^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-\frac{i2\pi}{3}} & e^{-\frac{i4\pi}{3}} \\ 1 & e^{-\frac{i4\pi}{3}} & e^{-\frac{i8\pi}{3}} \end{bmatrix}.$$

We need to confirm that FF^{-1} equals the identity matrix. On the main diagonal that is clear. Row j of F times column j of F^{-1} is $(1/n)(1+1+\dots+1)$, which is 1. The harder part is off the diagonal, to show that row j of F times column k of F^{-1} gives zero:

$$, \text{ if } j \neq k . \tag{A-4}$$

The key is to notice that those terms are the powers of :

$$1 + W + W^2 + \dots + W^{n-1} = 0 . \tag{A-5}$$

6.2 Appendix 2

In a general case, we have

$$[B]_N = [K]_N [C]_N [D]_N ,$$

where

$$[K]_N = \begin{bmatrix} \sqrt{2} & 0 & 0 & \dots \\ -\sqrt{2} & 2 & 0 & \dots \\ \sqrt{2} & -2 & 2 & \dots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}, [D]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \dots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix} ,$$

and $\Phi_i = 2i + 1, i \in \{0,1,2,\dots,N-1\}$.

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

$$[C]_N = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & C_{4N}^{2k_0\Phi_2} & \dots & C_{4N}^{2k_0\Phi_{N-2}} & C_{4N}^{2k_0\Phi_{N-1}} \\ C_{4N}^{2k_1\Phi_0} & C_{4N}^{2k_1\Phi_1} & C_{4N}^{2k_1\Phi_2} & \dots & C_{4N}^{2k_1\Phi_{N-2}} & C_{4N}^{2k_1\Phi_{N-1}} \\ C_{4N}^{2k_2\Phi_0} & C_{4N}^{2k_2\Phi_1} & C_{4N}^{2k_2\Phi_2} & \dots & C_{4N}^{2k_2\Phi_{N-2}} & C_{4N}^{2k_2\Phi_{N-1}} \\ \vdots & & & \vdots & & \vdots \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & C_{4N}^{2k_{N-2}\Phi_2} & \dots & C_{4N}^{2k_{N-2}\Phi_{N-2}} & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix} , \tag{A-6}$$

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

According to (6-37), a $N \times N$ matrix $[B]_N$ from $[C]_{2N}$ can be simply presented by

$$[B]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & C_{4N}^{\Phi_2} & \dots & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & C_{4N}^{(2k_0+1)\Phi_2} & \dots & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ C_{4N}^{(2k_1+1)\Phi_0} & C_{4N}^{(2k_1+1)\Phi_1} & C_{4N}^{(2k_1+1)\Phi_2} & \dots & C_{4N}^{(2k_1+1)\Phi_{N-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{4N}^{(2k_{N-2}+1)\Phi_0} & C_{4N}^{(2k_{N-2}+1)\Phi_1} & C_{4N}^{(2k_{N-2}+1)\Phi_2} & \dots & C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}} \end{bmatrix}. \tag{A-7}$$

And based on (6-78), we have the formula

$$C_{4N}^{(2k_i+1)\Phi_m} = 2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m} - C_{4N}^{(2k_i-1)\Phi_m} = -C_{4N}^{(2k_i-1)\Phi_m} + 2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m}, \tag{A-8}$$

where $m \in \{0,1,2,\dots\}$.

Thus we can calculate that

$$[K]_N [C]_N [D]_N = \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 & \dots & 0 \\ -\sqrt{2} & 2 & 0 & 0 & & 0 \\ \sqrt{2} & -2 & 2 & 0 & & 0 \\ -\sqrt{2} & 2 & -2 & 2 & \dots & \vdots \\ \sqrt{2} & -2 & 2 & -2 & 2 & \\ \vdots & & & & & \ddots \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & C_{4N}^{2k_0\Phi_2} & \dots & C_{4N}^{2k_0\Phi_{N-2}} & C_{4N}^{2k_0\Phi_{N-1}} \\ C_{4N}^{2k_1\Phi_0} & C_{4N}^{2k_1\Phi_1} & C_{4N}^{2k_1\Phi_2} & \dots & C_{4N}^{2k_1\Phi_{N-2}} & C_{4N}^{2k_1\Phi_{N-1}} \\ C_{4N}^{2k_2\Phi_0} & C_{4N}^{2k_2\Phi_1} & C_{4N}^{2k_2\Phi_2} & \dots & C_{4N}^{2k_2\Phi_{N-2}} & C_{4N}^{2k_2\Phi_{N-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & C_{4N}^{2k_{N-2}\Phi_2} & \dots & C_{4N}^{2k_{N-2}\Phi_{N-2}} & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix},$$

$$\begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \dots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix} = \begin{bmatrix} 1 & & & & & \\ -1+2C_{4N}^{2k_0\Phi_0} & & & & & \\ 1-2C_{4N}^{2k_0\Phi_0} + 2C_{4N}^{2k_1\Phi_0} & & & & & \\ \vdots & & & & & \\ -1+2C_{4N}^{2k_0\Phi_1} & & & & & \\ 1-2C_{4N}^{2k_0\Phi_1} + 2C_{4N}^{2k_1\Phi_1} & & & & & \\ \vdots & & & & & \\ -1+2C_{4N}^{2k_0\Phi_{N-1}} & & & & & \\ 1-2C_{4N}^{2k_0\Phi_{N-1}} + 2C_{4N}^{2k_1\Phi_{N-2}} & & & & & \\ \vdots & & & & & \end{bmatrix} \begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \dots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix},$$

$$= \begin{bmatrix} C_{4N}^{\Phi_0} & & & & & \\ -C_{4N}^{\Phi_0} + 2C_{4N}^{2k_0\Phi_0}C_{4N}^{\Phi_0} & & & & & \\ C_{4N}^{\Phi_0} - 2C_{4N}^{2k_0\Phi_0}C_{4N}^{\Phi_0} + 2C_{4N}^{2k_1\Phi_0}C_{4N}^{\Phi_0} & & & & & \\ \vdots & & & & & \\ C_{4N}^{\Phi_1} & & & & & \\ -C_{4N}^{\Phi_1} + 2C_{4N}^{2k_0\Phi_1}C_{4N}^{\Phi_1} & & & & & \\ C_{4N}^{\Phi_1} - 2C_{4N}^{2k_0\Phi_1}C_{4N}^{\Phi_1} + 2C_{4N}^{2k_1\Phi_1}C_{4N}^{\Phi_1} & & & & & \\ \vdots & & & & & \\ -C_{4N}^{\Phi_{N-1}} + 2C_{4N}^{2k_0\Phi_{N-1}}C_{4N}^{\Phi_{N-1}} & & & & & \\ C_{4N}^{\Phi_{N-1}} - 2C_{4N}^{2k_0\Phi_{N-1}}C_{4N}^{\Phi_{N-1}} + 2C_{4N}^{2k_1\Phi_{N-1}}C_{4N}^{\Phi_{N-1}} & & & & & \\ \vdots & & & & & \end{bmatrix}, \tag{A-9}$$

Since $k_0 = 1$, we get

$$-C_{4N}^{\Phi_m} + 2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m} = -C_{4N}^{(2k_0-1)\Phi_m} + 2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m} = C_{4N}^{(2k_0+1)\Phi_m}, \tag{A-10}$$

and
$$C_{4N}^{\Phi_m} - 2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m} = -(-C_{4N}^{(2k_0-1)\Phi_m} + 2C_{4N}^{2k_0\Phi_m}C_{4N}^{\Phi_m}) = -C_{4N}^{(2k_0+1)\Phi_m}. \tag{A-11}$$

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

$$\begin{aligned} C_{4N}^{\Phi_m} - 2C_{4N}^{2k_{i-1}\Phi_m}C_{4N}^{\Phi_m} + 2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m} &= -C_{4N}^{(2k_{i-1}+1)\Phi_m} + 2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m} \\ &= -C_{4N}^{(2k_i-1)\Phi_m} + 2C_{4N}^{2k_i\Phi_m}C_{4N}^{\Phi_m} = C_{4N}^{(2k_i+1)\Phi_m}. \end{aligned} \tag{A-12}$$

Taking the (A-10)-(A-12) to (A-9), we can rewrite that

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

The proof is completed.

7. References

Ahmed, N. & K.R. Rao(1975), *Orthogonal Transforms for Digital Signal Processing*, 0387065563, Berlin, Germany: Springer-Verlag

Chen, Z.; M.H. Lee, & Guihua Zeng(2008), Fast Cocyclic Jacket Transform, *IEEE Trans. on Signal Proc.* Vol. 56, No.5, 2008, (2143-2148) ,1053-587X

Fan, C.-P. & Jar-Ferr Yang(1998), Fast Center Weighted Hadamard Tranform Algorithm, *IEEE Trans.on CAS-II* Vol. 45, No. 3, (1998) (429-432), 1057-7130

Hou, Jia; M.H. Lee, & Ju Yong Park(2003), New Polynomial Construction of Jacket Transform, *IEICE Trans. Fund.* Vol. E86A, No. 3, (2003) (652-660) , 0916-8508

Lee, M.H.(1989), The Center Weighted Hadamard Transform, *IEEE Trans. on Circuits and Syst.* 36(9), (1989) (1247- 1249), 0098-4094

Lee, M.H.(1992), High Speed Multidimensional Systolic Arrays for Discrete Fourier Transform, *IEEE Trans. on .Circuits and Syst.II* 39 (12) , (1992) (876-879), 1057-7130

Lee, M.H.(2000) , A New Reverse Jacket Transform and Its Fast Algorithm, *IEEE, Trans. on Circuit and System* 47(1) ,2000, (39-47), 1057-7130

Lee, M.H. & Ken Finlayson(2007), A simple element inverse Jacket transform coding, *IEEE Signal Processing Lett.* 14 (5) , 2007, (325-328), 1070-9908

- Lee, M. H. & D.Y.Kim(1984), Weighted Hadamard Transform for S/N Ratio Enhancement in Image Transform, *Proc. of IEEE ISCAS'84*, 1984, pp. 65-68.
- Lee, S.R. & M.H. Lee(1998), On the Reverse Jacket Matrix for Weighted Hadamard Transform, *IEEE Trans. On CAS-II*, Vol. 45, No. 3, (1998) (436-441), 1057-7130
- Lee, M.H.; N.L.Manev & Xiao-Dong Zhang (2008), Jacket transform eigenvalue decomposition, *Appl. Math.Comput.*198, (May 2008). (858-864)
- Lee, M.H.; B. S. Rajan, & Ju Yong Park(2001), A Generalized Reverse Jacket Transform, *IEEE Trans. on Circuits and Syst. II* 48 (7), (2001) (684-690), 1057-7130
- Lee, S. R. & J. H. Yi(2002), Fast Reverse Jacket Transform as an Alternative Representation of N point Fast Fourier Transform, *Journal of Mathematical Imaging and Vision* Vol. 16, No. 1,,(2002) (1413-1420), 0924-9907
- Park, D.; M.H. Lee & Euna Choi(1999), Revisited DFT matrix via the reverse jacket transform and its application to communication, *The 22nd symposium on Information theory and its applications (SITA 99)*, 1999, Yuzawa, Niigata, Japan
- Rao, K.R. & P. Yip(1990), *Discrete Cosine Transform Algorithms, Advantages, Applications*,. 0-12-580203-X, Academic Press, San Diego, CA, USA
- Rao, K.R. & J.J. Hwang(1996), *Techniques & Standards for Image Video & Audio Coding*, 0-13-309907-5, Prentice Hall, Upper Saddle River, NJ, USA
- Whelchel, J.E. & D.F. Guinn(1968), The fast fourier hadamard transform and its use in signal representation and classification, *Proc. EASCON'68*, 1968, pp. 561-564

IntechOpen



Fourier Transforms - Approach to Scientific Principles

Edited by Prof. Goran Nikolic

ISBN 978-953-307-231-9

Hard cover, 468 pages

Publisher InTech

Published online 11, April, 2011

Published in print edition April, 2011

This book aims to provide information about Fourier transform to those needing to use infrared spectroscopy, by explaining the fundamental aspects of the Fourier transform, and techniques for analyzing infrared data obtained for a wide number of materials. It summarizes the theory, instrumentation, methodology, techniques and application of FTIR spectroscopy, and improves the performance and quality of FTIR spectrophotometers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Daechul Park and Moon Ho Lee (2011). Orthogonal Discrete Fourier and Cosine Matrices for Signal Processing, Fourier Transforms - Approach to Scientific Principles, Prof. Goran Nikolic (Ed.), ISBN: 978-953-307-231-9, InTech, Available from: <http://www.intechopen.com/books/fourier-transforms-approach-to-scientific-principles/orthogonal-discrete-fourier-and-cosine-matrices-for-signal-processing>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen