

## Fast Hybrid Transform: DCT-II/DFT/HWT

Dan-Ping Xu<sup>a)</sup>, Dae-Chol Shin<sup>a)</sup>, Wei Duan<sup>a)</sup>, Moon Ho Lee<sup>a)†</sup>

### Abstract

In this paper, we address a new fast DCT-II/DFT/HWT hybrid transform architecture for digital video and fusion mobile handsets based on Jacket-like sparse matrix decomposition. This fast hybrid architecture is consist of source coding standard as MPEG-4, JPEG 2000 and digital filtering discrete Fourier transform, and has two operations: one is block-wise inverse Jacket matrix (BIJM) for DCT-II, and the other is element-wise inverse Jacket matrix (EIJM) for DFT/HWT. They have similar recursive computational fashion, which mean all of them can be decomposed to Kronecker products of an identity Hadamard matrix and a successively lower order sparse matrix. Based on this trait, we can develop a single chip of fast hybrid algorithm architecture for intelligent mobile handsets.

Keyword : DCT-II, DFT, Wavelet, Jacket Matrix, Element/Block-wise Inverse, Sparse Matrix decomposition, Hybrid Transform, mobile handsets.

### I. Introduction

THE Last Decade has been seen a quiet hybrid fusion revolution in digital video technology. Digital video is everywhere such as DVD, players, computers and mobile handsets. Nowadays, many of us are just as likely to catch the latest news on the web as on the smart TV and iphone. Video compression is essential to all of these applications. The discrete cosine transform (DCT-II) is popular compression structures and it is usually accepted as the best suboptimal transformation that its performances is very close to that of the statistically optimal Karhunen-Loeve transform for MPEG 4 and H.264 standard<sup>[1][2]</sup>. Discrete

orthogonal transform has found applications in signal classification and representation<sup>[1-7]</sup>. The discrete signal processing of DFT (Discrete Fourier Transform) is a popular transformation for OFDM-4G (Orthogonal Frequency Division Multiplexing) and 4 Generation Mobile Communication<sup>[8-9]</sup>. The OFDM is a key technology for next-generation cellular communication (3GPP-LTE, mobile WiMAX, IMT-Advanced) as well as wireless LAN. (IEEE 802.11a, IEEE 802.11n) Wireless PAN (Multiband OFDM), and a broadcasting (DAB, DVB, DMB) which is based on DFT. Furthermore, the discrete wavelet transform based on the Haar Transform (HWT) is also very useful in JPEG 2000 standard and signal analysis<sup>[7][10]</sup>. To analyze these four different transforms, we now focus on the sparse unified matrix factorization for unified chip set based on Jacket matrix<sup>[11-17][20-22]</sup>.

The analysis and decomposition of the sparse matrix demonstrated as a useful tool to develop the fast computations and character generalization. The DCT-II, DFT and

a) 전북대학교 전자공학부  
Chonbuk National University, Institute of Information and Communication

† 교신저자 : 이문호(moonho@chonbuk.ac.kr)

※ This work was supported by World Class University, R32-2009-000-20014-0 and 2010-0020942 NRF, Republic of Korea.

· 접수일(2011년7월19일), 수정일(2011년8월22일), 게재확정일(2011년9월15일)

HWT matrices can be decomposed to one orthogonal character matrix and a special sparse matrix. The inverse of the sparse matrix is from block-wise inverse or element-wise inverse (proved in appendix). Mathematically, let  $C$  be a matrix, then the matrix  $A$  is a Jacket matrix<sup>[15-16]</sup>. Obviously the special sparse matrix belongs to Jacket matrix.

We focus on the unified architecture of the sparse matrix decomposition and propose hybrid architecture to combine the DCT-II, DFT and HWT together. This paper is organized as follows: in section II, we introduce the block-wise inverse sparse matrix decomposition for DCT-II matrix. In section III and section IV, we introduce the element-wise inverse sparse matrix decomposition for DFT and HWT matrices, respectively. In section V, the hybrid architecture of the three transforms and their relationships are expressed. The conclusion is given in section VI.

## II. BLOCK-WISE Inverse sparse matrix Decomposition for DCT-II Transform

The DCT-II<sup>[1][3][20-21][23-25]</sup> matrix can be expressed as:

$$\begin{aligned} [C_N]_{m,n} &= c_m \sqrt{\frac{2}{N}} \cos \frac{m(2n+1)\pi}{2N} = \sqrt{\frac{2}{N}} [C_N]_{m,n} \\ c_m &= \begin{cases} 1/\sqrt{2} & , m = 0 \\ 1 & , m \neq 0 \end{cases}, \quad m, n = 0, 1, \dots, N-1 \end{aligned} \quad (1)$$

The basic matrix is  $2 \times 2$  matrix, which is given as follows:

$$[C]_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [C]_2 \quad (2)$$

The  $N \times N$  DCT-II matrix can be given as:

$$[C]_N = \sqrt{\frac{2}{N}} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \dots & 1/\sqrt{2} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & \dots & C_{4N}^{2k_0\Phi_{N-1}} \\ \dots & \dots & \dots & \dots \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & \dots & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix} = \sqrt{\frac{2}{N}} [C]_N \quad (3)$$

where

$$C_i^k = \cos(k\pi/l), k_i = i+1, \Phi_j = 2j+1, i=0,1,L, N-2, j=0,1,L, N-1$$

We can build a Jacket-like matrix to decompose the DCT-II matrix easily by multiplying some permutation matrices. Thus, we can define a row permutation matrix [10] as

$$[Pr]_N = [Pr_{i,j}]_N = \begin{cases} 1, & \text{if } i = 2j \bmod N, 0 \leq j \leq N/2-1 \\ 1, & \text{if } i = (2j+1) \bmod N, N/2 \leq j \leq N-1, \quad i, j = 0, 1, L, N-1 \\ 0, & \text{others} \end{cases} \quad (4)$$

Furthermore, we can also define a reversible permutation matrix, the column permutation matrix as follows

$$[Pc]_N = \begin{bmatrix} I_{N/2} & 0 \\ 0 & \bar{I}_{N/2} \end{bmatrix}, \quad (5)$$

where  $I$  is identity matrix, and  $\bar{I}$  is opposite identity matrix.

From Eq. (4) and Eq. (5), note that

$$[Pr]_2 = [I]_2, [Pc]_2 = [I]_2, [Pr]_N^{-1} = [Pr]_N, [Pc]_N^{-1} = [Pc]_N \quad (6)$$

Fortunately, by multiplying and matrices on the both sides of DCT-II matrix, respectively, we can get the Jacket-like matrix and decompose the DCT-II in an easy method.

The permuted  $2 \times 2$ ,  $4 \times 4$  and  $8 \times 8$  DCT-II matrices are

$$\begin{aligned} [\tilde{C}]_2 &= [Pr]_2 [C]_2 [Pc]_2 = [I]_2 [C]_2 [I]_2 = [C]_2 = [C]_2 \\ [\tilde{C}]_4 &= \sqrt{\frac{2}{4}} [Pr]_4 [C]_4 [Pc]_4 = \frac{1}{\sqrt{2}} \begin{bmatrix} C_2 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \\ [\tilde{C}]_8 &= \sqrt{\frac{2}{8}} [Pr]_8 [C]_8 [Pc]_8 = \frac{1}{2} \begin{bmatrix} C_4 & 0 \\ 0 & B_4 \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix} \end{aligned} \quad (7)$$

Generally, the permuted DCT-II matrix  $[\tilde{C}]_N$  can be recursively formed by using

$$[\tilde{C}]_N = \sqrt{\frac{2}{N}} [\text{Pr}]_N [C]_N [Pc]_N = \sqrt{\frac{2}{N}} \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \quad (8)$$

where  $[B]_{N/2}$  can be calculated by

$$[B]_N = \left[ C_{4N}^{(2k_i-1)\phi_j} \right]_N, k_i = i+1, \phi_j = 2j+1, i, j = 0, 1, \dots, N-1 \quad (9)$$

where  $m, n \in \{1, 2, \dots, N/2\}$ .

The inverse form of (8) can be simply computed by

$$[\tilde{C}]_N^{-1} = \sqrt{\frac{N}{8}} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2}^{-1} & 0 \\ 0 & B_{N/2}^{-1} \end{bmatrix}. \quad (10)$$

Then, we have the block-wise inverse Jacket sparse matrix

$$\begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix}^{-1} = \begin{bmatrix} C_{N/2}^{-1} & 0 \\ 0 & B_{N/2}^{-1} \end{bmatrix}. \quad (11)$$

Furthermore, it was provided in [24] that the submatrix  $[B]_N$  can be represented by

$$[B]_N = [L]_N [C]_N [D]_N \quad (12)$$

where

$$[L]_N = \begin{bmatrix} \sqrt{2} & 0 & \dots & 0 \\ -\sqrt{2} & 2 & 0 & \dots \\ \sqrt{2} & -2 & 2 & \dots \\ \dots & \dots & \dots & 2 \end{bmatrix}, [D]_N = \text{diag} \left[ C_{4N}^{\Phi_0}, \dots, C_{4N}^{\Phi_{N-1}} \right]_N,$$

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

**Proof the expression of Eq. (12):**

Obviously, by using the relationship of sum and difference formulas of cosine function, we have

$$\begin{aligned} 2C_{4N}^{2k_0\phi_m} C_{4N}^{\phi_m} - C_{4N}^{\phi_m} &= C_{4N}^{(2k_0+1)\phi_m} \quad k_i = i+1, i = 0, 1, \dots, N-1 \\ 2C_{4N}^{2k_1\phi_m} C_{4N}^{\phi_m} - 2C_{4N}^{2k_0\phi_m} C_{4N}^{\phi_m} + C_{4N}^{\phi_m} &= C_{4N}^{(2k_1+1)\phi_m} \\ 2C_{4N}^{2k_2\phi_m} C_{4N}^{\phi_m} - 2C_{4N}^{2k_1\phi_m} C_{4N}^{\phi_m} + 2C_{4N}^{2k_0\phi_m} C_{4N}^{\phi_m} - C_{4N}^{\phi_m} &= C_{4N}^{(2k_2+1)\phi_m} \\ 0 &= 0 \\ 2C_{4N}^{2k_{N-2}\phi_m} C_{4N}^{\phi_m} - 2C_{4N}^{2k_{N-3}\phi_m} C_{4N}^{\phi_m} + L + 2C_{4N}^{2k_0\phi_m} C_{4N}^{\phi_m} - C_{4N}^{2k_{N-2}\phi_m} &= C_{4N}^{(2k_{N-2}+1)\phi_m}, \end{aligned} \quad (13)$$

By taking Eq. (13) and into the right hand side of Eq. (12), we have

$$[L]_N [C]_N [D]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & L & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & L & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ C_{4N}^{(2k_1+1)\Phi_0} & C_{4N}^{(2k_1+1)\Phi_1} & L & C_{4N}^{(2k_1+1)\Phi_{N-1}} \\ M & M & M & M \\ C_{4N}^{(2k_{N-2}+1)\Phi_0} & C_{4N}^{(2k_{N-2}+1)\Phi_1} & L & C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}} \end{bmatrix}. \quad (14)$$

According to Eq. (9), the left hand side of Eq. (12) matrix  $[B]_N$  from  $[C]_{2N}$  can be represented by

$$[B]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & \dots & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & \dots & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ \dots & \dots & \dots & \dots \\ C_{4N}^{(2k_{N-2}+1)\Phi_0} & C_{4N}^{(2k_{N-2}+1)\Phi_1} & \dots & C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}} \end{bmatrix}. \quad (15)$$

So we can obtain Eq. (14) and Eq. (15) are the same and the expression of Eq. (12) is correct. ■

Thus the permuted DCT-II matrix can be written by

$$\begin{aligned} [\tilde{C}]_N &= \sqrt{\frac{2}{N}} \begin{bmatrix} C_{N/2} & 0 \\ 0 & L_{N/2} C_{N/2} D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= \sqrt{\frac{2}{N}} \begin{bmatrix} I_{N/2} & 0 \\ 0 & L_{N/2} \end{bmatrix} [I_2 \otimes C_{N/2}] \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}. \end{aligned} \quad (16)$$

The DCT-II general recursive form is shown as follow

$$[C]_N = \sqrt{\frac{2}{N}} [\Pr]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & L_{N/2} \end{bmatrix} [I_2 \otimes C_{N/2}] \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} [Pc]_N \quad (17)$$

The Eq. (17) can be represented by

$$\begin{aligned} [C]_N &= \sqrt{\frac{2}{N}} [\Pr]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & L_{N/2} \end{bmatrix} \dots [I_{N/2} \otimes [\Pr]_4] \begin{bmatrix} I_{N/2} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \end{bmatrix} [I_{N/2} \otimes C_2] \\ &\begin{bmatrix} I_{N/2} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \end{bmatrix} [I_{N/2} \otimes [Pc]_4] \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} [Pc]_N \\ &= \sqrt{\frac{2}{N}} [\Pr]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & L_{N/2} \end{bmatrix} \dots [I_{N/2} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix}] [I_{N/2} \otimes C_2] \begin{bmatrix} I_{N/2} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \end{bmatrix} \\ &\begin{bmatrix} I_{N/2} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} [Pc]_N \end{aligned} \quad (18)$$

The Eq. (18) of butterfly data flow graph is shown as in Fig 1.

### III. Element-wise Inverse Sparse Matrix Decomposition for DFT Transform

The DFT is a Fourier representation of a given sequence,  $x(m)$ ,  $0 \leq m \leq N-1$  and it is defined by [3-4].

$$X(n) = \sum_{m=0}^{N-1} x(m)W^{nm}, \quad 0 \leq n \leq N-1 \quad (19)$$

where  $W = e^{-j\frac{2\pi}{N}}$ ,  $j = \sqrt{-1}$ . The N-point DFT matrix can be denoted by  $[F]_N = [W^{nm}]_N$ .

$$[\tilde{F}]_2 = [\Pr]_2 [F]_2 = [I]_2 [F]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [F]_2 \quad (20)$$

Similar to the section II, we can write a permuted 4x4 DFT matrix by using

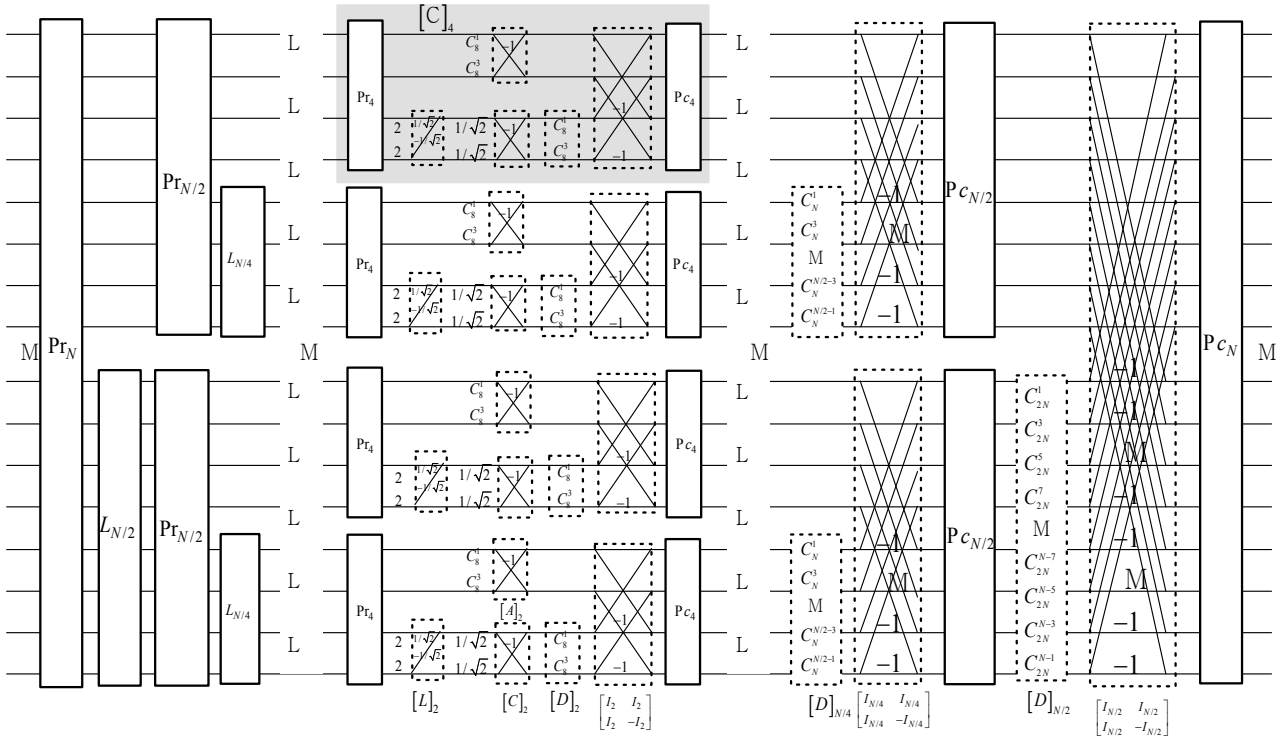


그림 1. N×N DCT 고속신호처리도  
 Fig. 1. Butterfly data flow graph of the proposed computation of the N×N DCT-II transform

$$[\tilde{F}]_4 = [\text{Pr}]_4 [F]_4 = \begin{bmatrix} \tilde{F}_2 & 0 \\ 0 & E_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} = \begin{bmatrix} F_2 & 0 \\ 0 & E_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \quad (21)$$

where  $[E]_2 = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}$

The permuted 8×8 DFT matrix is

$$[\tilde{F}]_8 = [\text{Pr}]_8 [F]_8 = \begin{bmatrix} \tilde{F}_4 & 0 \\ 0 & E_4 \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix} \quad (22)$$

Generally, the N×N permuted DFT matrix has

$$[\tilde{F}]_N = [\text{Pr}]_N [F]_N = \begin{bmatrix} \tilde{F}_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}, \quad (23)$$

where

$$[E]_{N/2} = [W_N^{k_i \phi_j}]_{N/2}, k_i = 2i + 1, \phi_j = j, i, j = 0, 1, L, N/2 - 1, \quad (24)$$

Its inverse form can be obtained by using EIJM, which is attached in the appendix

$$[E]_{N/2}^{-1} = [W_N^{-k_i \phi_j}]_{N/2}^T, k_i = 2i + 1, \phi_j = j, i, j = 0, 1, L, N/2 - 1, \quad (25)$$

Just like in 4×4 permuted DFT matrix, we can get the inverse of submatrix by Jacket matrix as below

$$[E]_2^{-1} = \begin{bmatrix} 1/1 & -1/j \\ 1/1 & 1/j \end{bmatrix}^T = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}. \quad (26)$$

The submatrix  $[E]_N$  can be written by

$$[E]_N = [\text{Pr}]_N [\tilde{F}]_N [W]_N, \quad (27)$$

where  $W$  is the complex unit for 2N-point DFT matrix. Similar to (16), we can rewrite the permuted DFT matrix by using

$$[\tilde{F}]_N = \begin{bmatrix} \tilde{F}_{N/2} & 0 \\ 0 & \text{Pr}_{N/2} \tilde{F}_{N/2} W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} = \begin{bmatrix} I_{N/2} & 0 \\ 0 & \text{Pr}_{N/2} \end{bmatrix} [I_2 \otimes \tilde{F}_{N/2}] \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \quad (28)$$

As a result, the general recursive form DFT matrix can

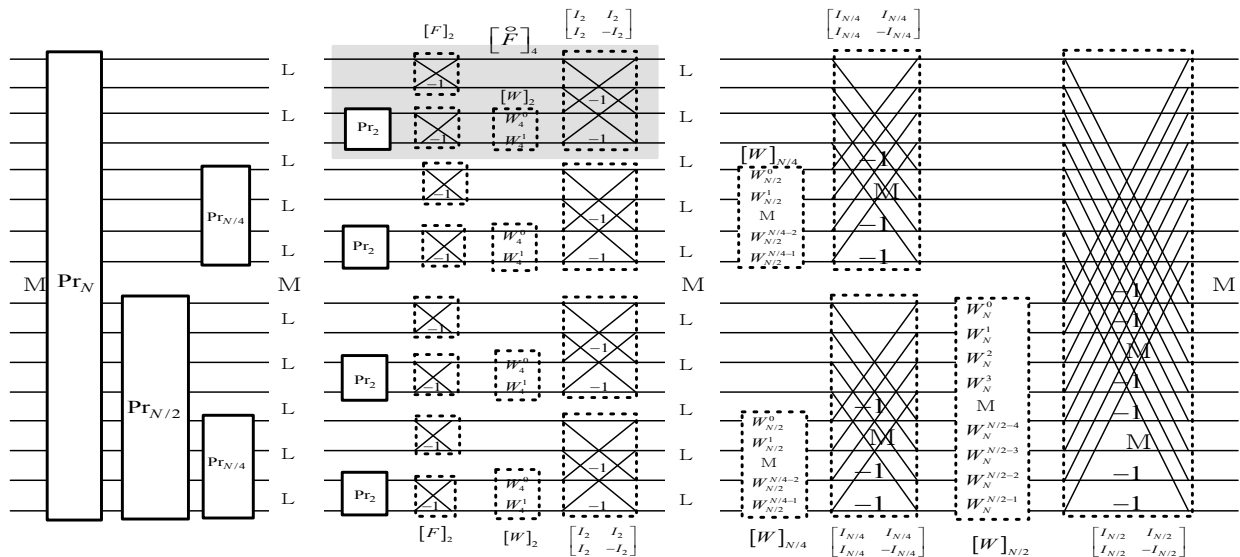


그림 2. N×N DFT 고속신호처리도  
Fig. 2. Butterfly data flow graph of the proposed computation of the N×N DFT transform

be expanded as

$$[F]_N = [\text{Pr}]_N^{-1} [\hat{F}]_N = [\text{Pr}]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & \text{Pr}_{N/2} \end{bmatrix} \cdots \begin{bmatrix} I_{N/2} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & \text{Pr}_2 \end{bmatrix} \\ I_{N/2} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & W_2 \end{bmatrix} \end{bmatrix} [I_{N/2} \otimes F_2] \quad (29)$$

The Fig.2 shows the butterfly data flow graph of Eq. (29).

#### IV. Element-WISE Inverse Sparse Matrix Decomposition for HWT Transform

The discrete wavelet transform based on the Haar matrix (HWT) [7], from  $N$  components of the signal to  $N$  wavelet coefficients, is expressed by a  $N \times N$  matrix  $S$ . Here come the two directions: Synthesis in discrete time:  $x = Sb$ , Analysis in discrete time:  $y = Hx$ , where  $H = -S^{-1} = S^r$ .

As we know  $[H]_2 = \begin{bmatrix} r & r \\ r & -r \end{bmatrix} = [S]_2$  and the scaled coefficient:  $r = 1/\sqrt{2}$ . Its inverse matrix is from element-wise inverse, such that

$$[H]_2^{-1} = \frac{1}{2} \begin{bmatrix} 1/r & 1/r \\ 1/r & -1/r \end{bmatrix}^T = \begin{bmatrix} r & r \\ r & -r \end{bmatrix} = [H]_2 \quad (30)$$

$$[\hat{H}]_2 = [Pi]_2 [H]_2 [Pj]_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r & r \\ r & -r \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} r & -r \\ r & r \end{bmatrix} \quad (31)$$

where:

$$[Pi]_N = \begin{bmatrix} 0 & I_{N/2} \\ I_{N/2} & 0 \end{bmatrix}, [Pj]_N = [\text{Pr}]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & -I_{N/2} \end{bmatrix} \quad (32)$$

Moreover, set

$$[Pa]_N = [Pi]_N^{-1} = [Pi]_N, [Pb]_N = r [Pj]_N^{-1} = r \begin{bmatrix} I_{N/2} & 0 \\ 0 & -I_{N/2} \end{bmatrix} [\text{Pr}]_N \quad (33)$$

Similar to the section II, we can write a permuted  $4 \times 4$

HWT matrix by using

$$\begin{aligned} [\hat{H}]_4 &= [Pi]_4 [H]_4 [Pj]_4 \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r^2 & r^2 & r^2 & r^2 \\ r^2 & r^2 & -r^2 & -r^2 \\ r & -r & 0 & 0 \\ 0 & 0 & r & -r \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = r \begin{bmatrix} I_2 & I_2 \\ H_2 & -H_2 \end{bmatrix} \\ &= r \begin{bmatrix} I_2 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} = r \begin{bmatrix} I_2 & 0 \\ 0 & \text{Pr}_2^{-1} \hat{H}_2 \text{Pr}_2^{-1} \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \\ &= \begin{bmatrix} I_2 & 0 \\ 0 & Pa_2 \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & \hat{H}_2 \end{bmatrix} \begin{bmatrix} rI_2 & 0 \\ 0 & Pb_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \end{aligned} \quad (34)$$

For the 8-point HWT, the permuted  $8 \times 8$  HWT is

$$\begin{aligned} [\hat{H}]_8 &= [Pi]_8 [H]_8 [Pj]_8 = r \begin{bmatrix} I_4 & 0 \\ 0 & H_4 \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix} \\ &= r \begin{bmatrix} I_4 & 0 \\ 0 & \text{Pr}_4^{-1} \hat{H}_4 \text{Pr}_4^{-1} \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix} \\ &= \begin{bmatrix} I_4 & 0 \\ 0 & Pa_4 \end{bmatrix} \begin{bmatrix} I_6 & 0 \\ 0 & Pa_2 \end{bmatrix} \begin{bmatrix} I_6 & 0 \\ 0 & \hat{H}_2 \end{bmatrix} \begin{bmatrix} I_4 & 0 \\ 0 & rI_2 \end{bmatrix} \begin{bmatrix} rI_4 & 0 \\ 0 & Pb_2 \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix} \end{aligned} \quad (35)$$

In general, we can write that

$$\begin{aligned} [\hat{H}]_N &= [Pi]_N [H]_N [Pj]_N = r \begin{bmatrix} I_{N/2} & 0 \\ 0 & H_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= r \begin{bmatrix} I_{N/2} & 0 \\ 0 & \text{Pr}_{N/2}^{-1} \hat{H}_{N/2} \text{Pr}_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pa_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & \hat{H}_{N/2} \end{bmatrix} \begin{bmatrix} rI_{N/2} & 0 \\ 0 & Pb_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \end{aligned} \quad (36)$$

The Eq. (36) of general recursive form for HWT matrix can be rewritten as

$$\begin{aligned} [H]_N &= [Pi]_N^{-1} [\hat{H}]_N [Pj]_N^{-1} \\ &= \frac{1}{r} [Pa]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pa_{N/2} \end{bmatrix} \begin{bmatrix} I_{3N/4} & 0 \\ 0 & Pa_{N/4} \end{bmatrix} \begin{bmatrix} I_{N-2} & 0 \\ 0 & Pa_2 \end{bmatrix} \begin{bmatrix} I_{N-2} & 0 \\ 0 & \hat{H}_2 \end{bmatrix} \begin{bmatrix} I_{N-4} & 0 \\ 0 & rI_2 \end{bmatrix} \begin{bmatrix} I_{N-4} & 0 \\ 0 & Pb_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \\ &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & rI_{N/4} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pb_{N/4} \end{bmatrix} \begin{bmatrix} I_{N/4} & I_{N/4} \\ I_{N/4} & -I_{N/4} \end{bmatrix} \begin{bmatrix} rI_{N/2} & 0 \\ 0 & Pb_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} [Pb]_N \end{aligned} \quad (37)$$

Corresponding to Eq. (37), we can draw the butterfly data flow graph in Fig 3.

### V. proposed Hybrid architecture

In this paper, we derive the recursive formulas for DCT-II /DFT and wavelet matrices. The results show that the DCT-II /DFT and wavelet matrices can be unified by using the same sparse matrix decomposition algorithm based on Jacket matrix, and recursive architecture within some characters changed.

Clearly, the butterfly data flow graphs corresponding to the Eq. (18), (29) and (37) have the similar recursive flows  $\begin{bmatrix} I_N & I_N \\ I_N & -I_N \end{bmatrix}$ . What's more, all of the three graphs have  $2h+1 (h = \log_2 N - 1)$  steps, and we can derive Fig. 2 and 3 from Fig. 1 by the following steps after input the similar recursive partas Eq. (38).

To get the equation of (29) as same as that of (18), we change from  $[Pr], [L]$  to  $[Pr]$  and  $[D]_{N/2} [Pc]_n$  to  $[W]_{N/2}$ , with the parameters  $\{n \in 2, 4, 8, \dots, N/2\}$ , and then we can

get the DFT matrix as we need. For the equation (37), we can change the switching from  $[Pr]_n [L]_{n/2}$  to  $[Pa]_n$  and  $[D]_{n/2} [Pc]_n$  to  $[Pb]_n$ , after input the data, multiply  $[M_1]$  at the first h steps, and when the last h steps coming, multiply the  $[M_2]$  and  $[M_3]$  matrices.

$$[M] = [I]_{N/2^{h+1}} \otimes \begin{bmatrix} I_{2^h} & I_{2^h} \\ I_{2^h} & -I_{2^h} \end{bmatrix} \tag{38}$$

$$[M_1] = [I]_{2^{h+1}-1} \otimes \left( [I]_{N/2^h} \oplus [Pa]_{N/2^h}^{-1} \right) \oplus [I]_{N/2^{h+1}} \tag{39}$$

$$[M_2] = [I]_{N-2^{h+1}} \oplus r [I]_{2^h} \oplus [I]_{2^h} \tag{40}$$

$$[M_3] = [I]_{N/2^{h+1}-1} \otimes \begin{bmatrix} I_{2^h} & -I_{2^h} \\ I_{2^h} & I_{2^h} \end{bmatrix} \oplus [I]_{2^{h+1}} \tag{41}$$

where  $\otimes$  is Kronecker product and  $\oplus$  is the diagonal sum

As illustrated in Fig.1, Fig.2, Fig.3, we find that the DFT computation can be from the computation of the DCT-II matrix by replacing the submatrix  $[D]_{N/2} [Pc]_N$  to  $[W]_{N/2}$ , and the permutation matrix  $[Pr]_N [L]_{N/2}$  to  $[Pr]_{N/2}$ . As to the HWT, we /not only need to replace submatrix  $[D]_{N/2} [Pc]_N$

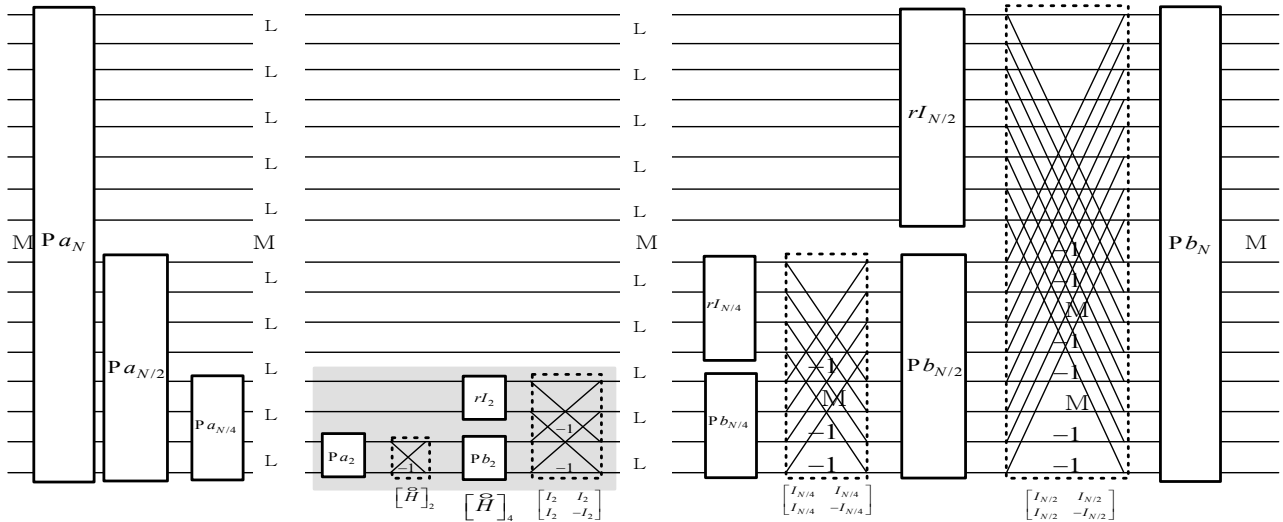


그림 3. N×N HWT 고속신호처리도  
 Fig. 3. Butterfly data flow graph of the proposed computation of the N×N HWT transform

Table I. omputational Complexity

	Addition		Multiplication	
	Directly	Proposed	Directly	Proposed
DCT-II	$N(N-1)$	$N(3\log_2 N + N - 1) / 4$	$N^2$	$(N^2 / 32 + 2N)\log_2 N - N^2 / 16 - N / 2$
DFT		$N\log_2 N$		$N(\log_2 N - 1 / 2)$
HWT		$\log_2 N(2 + N / 2) - N / 2$		$(3 + 3N / 4)\log_2 N + 1 + N$

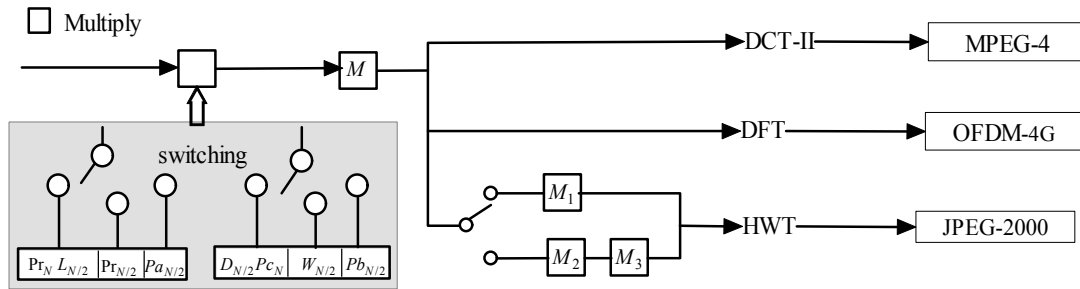


그림 4. 하이브리드 DCT-II/DFT /HWT 구조 블록이그림  
 Fig. 4. A simple DCT-II /DFT/HWT hybrid architecture block diagram

by  $[Pa]_{N/2}$ , and the permutation matrix  $[Pr]_{N/2}$  by  $[Pa]_{N/2}$ , but also multiply some special matrices at the output of HWT. As a result, a simple generalized block diagram for DCT-II /DFT/HWT hybrid architecture and its fast algorithm can be shown in Fig.4, which is also mathematically perfectly proved in aforementioned. The computation complexity is shown in Table I.

### VI. Conclusion

In this paper, we derive the fast hybrid transform of DCT-II, DFT and HWT matrices for hybrid fusion video coding or mobile handset. The results show that the DCT-II, DFT and HWT matrices can be unified by using the same sparse matrix decomposition algorithm based on Jacket matrix, and recursive architecture within some characters changed. This algorithm is useful to develop the unified orthogonal transform for video transcoding and

OFDM-4G. In future work, source-channel Reed Solomon coding for B4G handsets [27-31].

### References

- [1] K.R. Rao, and P. Yip. Jones, Discrete Cosine Transform Algorithms, Advantages, Applications, Academic Press, USA, 1990.
- [2] Iain E. Richardson, The H.264 Advanced Video Compression Standard, Second Edition, Wiley, 2010.
- [3] N. Ahmed, K. R. Rao, Orthogonal Transforms for Discrete Signal Processing, Spring-Verlog, 1975.
- [4] K.R.Rao, D.N Kim and J.J.Hwang, Fast Fourier Transform: Algorithms And Applications, Springer Science+ Business Media B.V. 2010.
- [5] Seung Son Kang, and Moon Ho Lee, "An expanded 2-D DCT algorithm based on convolution," IEEE Trans. on Consumer Electronics, vol.39, no.3, pp.159-165, 1993.
- [6] O'neil P V, Advanced Engineering Mathematics, 5th Edition, Thomson Business Information, 2003.
- [7] Gilbert Strang and Truong Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, 1996.
- [8] Yong Soo Cho et.al, MIMO-OFDM Wireless Communications with MATLAB, Wiley, 2010.
- [9] Moon Ho Lee, "High Speed Multidimensional Systolic Arrays for



- Discrete Fourier Transform," IEEE Trans. on Circuits Syst. II vol. 39 no. 12, pp876-879, Dec., 1992.
- [10] S.J.Chang, M.H.Lee, and J.Y.Park, "A High Speed VLSI Architecture of Discrete Wavelet Transform for MPEG-4," IEEE Trans. on Consumer Electronics, vol.43, no.3, pp.623-627, 1997.
- [11] K. Wahid, S. Shimu, M. Islam, D. Teng, Moon Ho Lee, and S-B. Ko. "Efficient Hardware Implementation of Hybrid Cosine-Fourier-Wavelet Transforms on a Single FPGA," IEEE ISCAS 2009, Taiwan, PP2325-2328.
- [12] Zhu Chen, Moon Ho Lee. "On Fast Hybrid Source Coding Design," IEEE ISITC 2007, Korea, PP. 143-147.
- [13] Moon Ho Lee, "A New Reverse Jacket Transform and Its Fast Algorithm," IEEE, Trans. Circuits System, vol. 47. no.1, pp .39-47, Jan. 2000.
- [14] Available online at [http://en.wikipedia.org/wiki/Jacket\\_matrix](http://en.wikipedia.org/wiki/Jacket_matrix); <http://en.wikipedia.org/wiki/Category:Matrices>
- [15] Moon Ho Lee, Y.L. Borissov. "A proof of Non-existence of Bordered Jacket matrices of odd Order over some Fields," Electronics Letters, Vol. 46, No.5, 4thMarch,2010.
- [16] Moon Ho Lee, "The Center Weighted Hadamrd Transform," IEEE Trans. on Circuits Syst. Vol.36, No.9, PP.1247-1249, Sept.1989.
- [17] Moon Ho Lee, Xiao-Dong Zhang,"Fast Block Center Weighted Hadamard Transform," IEEE Transactions on Circuits and Systems-I: Regular papers, Vol.54,No.12, December 2007.
- [18] M.H.Lee and J.Hou, "Fast Block Inverse Jacket Transform," IEEE Signal Processing Letters, Vol.13,No.8,PP.461-464, Aug.2006.
- [19] G.H.Zeng, M.H.Lee, "A Generalized Reverse Block Jacket Transform," IEEE Transactions on Circuits and Systems I. Vol.55, No.6,PP.:1589-1600,July,2008.
- [20] Zhu Chen, Moon Ho Lee, Guihua Zeng,"Fast Cocyclic Jacket Transform," IEEE Trans. on Signal Processing, Vol.56 No.5 PP 2145, 12 April 2008.
- [21] Khan A. Wahid, M.A.Islam, Samia S.Shimu, Moon Ho Lee, Seok-Bum Ko, "Hybrid Architecture and VLSI Implementation of the Cosine-Fourier - Haar Transforms," Circuits Syst. Signal Processing, Vol.29, No.6, PP.1193-1205, July 2009.
- [22] Jia Hou, Moon Ho Lee, Dae Chul Park, Kwang Jae Lee, "Simple element inverse DCT/DFT hybrid architecture algorithm," IEEE International Conference on Acoustics, Speech, and signal processing. May 14-19 2006 Toulouse, France.
- [23] Moon Ho Lee, "On Computing 2-D Systolic Algorithm for Discrete Cosine Transform," IEEE Trans. on Circuit and Systems, vol. 37, no.10, pp 766-769, Oct. 1990.
- [24] Y.-J. Chuang and J.-L. Wu, "An efficient matrix-splitter and merger for SIM Dinstuctions," IEICETrans. Inf. & Syst., vol.E88-D, no.7, July 2005.
- [25] C.-L. Wang, C.-Y. Chen, "High-throughput VLSI architectures for the 1-D and 2-D discrete cosine transform," IEEE Trans. Circuits and Systems for Video Technology, vol. 5, no. 1, pp. 31 - 40, Feb. 1995.
- [26] M.H.Lee, K.Finlayson, "A Simple Element Inverse Jacket Transform Coding," IEEE Signal Processing Letters, Vol.14, No.5, PP:325-328, May 2007.
- [27] Richard E. Blahut, Algebraic Codes for Data Transmission, Cambridge Univ. Press, 2003.
- [28] Zhu Chen, Moon Ho Lee, Chang Joo Kim, "Fast Hybrid DFT/DCT Architecture for OFDM in Cognitive Radio System," IEEE Future Generation Communication and Networking (FGCN), Vol.1, PP.301-306, 6-8 Dec.2007.
- [29] Moon Ho Lee, Y.Borissov, "On Jacket Transform over Finite Fields," IEEE International Symposium on Information Theory (ISIT), PP.2803-2807, June 28 - July 3, 2009, Seoul, Korea.
- [30] P.Triforov, Moon Ho Lee, "A Concatenated Coding Scheme Based on Folded Reed-Solomon Codes," IEEE International Symposium on Information Theory (ISIT), July 31-Aug. 6, 2011, St.pectenebeg, Russia.
- [31] Peter Trifonov, Moon Ho Lee, "Soft-Decision Decoding of Concatenated Codes Based on Folded Reed-Solomon Codes," Revised at IEEE Trans. On Information Theory,2011.

## Appendices

### Inverse Jacket Matrix

#### ① *Element-wise Inverse Jacket Matrix (EIJM)*

In mathematics, a jacket matrix is a square matrix  $[J]_N = [a_{ij}]_N$  of order N if its entries are non-zero and real, complex, or from a finite field, and

$$[J]_N [J]_N^{-1} = [J]_N^{-1} [J]_N = [I]_N \quad (A-1)$$

where  $[I]_N$  is the identity matrix, and

$$[J]_N = \frac{1}{N} [a_{ij}^{-1}]_N^T \quad (A-2)$$

where T denotes the transpose of the matrix.

In other words, the inverse of a Jacket matrix is determined its element-wise. The definition above may also be expressed as:

$$\forall u, v \in \{1, 2, \dots, n\} : a_{uu}, a_{vv} \neq 0, \quad \sum_{i=1}^n a_{ui}^{-1} a_{iv} = \begin{cases} 1, & u = v \\ 0, & u \neq v \end{cases} \quad (A-3)$$

The jacket matrix is a generalization of the Hadamard matrix [14-15] [26].

Example

$$[J]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, [J]_4^{-1} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1/2 & 1/2 & -1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (\text{A-4})$$

or more general

$$[J]_4 = \begin{bmatrix} a & b & b & a \\ b & -c & c & -b \\ b & c & -c & -b \\ a & -b & -b & a \end{bmatrix}, [J]_4^{-1} = \frac{1}{4} \begin{bmatrix} 1/a & 1/b & 1/b & 1/a \\ 1/b & -1/c & 1/c & -1/b \\ 1/b & 1/c & -1/c & -1/b \\ 1/a & -1/b & -1/b & 1/a \end{bmatrix} \quad (\text{A-5})$$

We have

$$[J]_N [J]_N^{-1} = [I]_N \quad (\text{A-6})$$

### ② Block-wise Inverse Jacket Matrix (BIJM)

If we instead the element of Eq. (A-5) to block matrix, then we can get the BIJM. Set a and b are the  $2 \times 2$  identity matrices, and c is the  $1/\sqrt{2}$  times of the lowest order Hadamard matrix, i.e,

$$[A]_2 = [B]_2 = [I]_2, [C]_2 = [H]_2 / \sqrt{2} \quad (\text{A-7})$$

where  $[H]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ .

The lowest order BWJM of order 8 is defined<sup>[16-18]</sup> to be

$$[J]_8 = [J]_{2 \times 4} = \begin{bmatrix} I_2 & I_2 & I_2 & I_2 \\ I_2 & -C_2 & C_2 & -I_2 \\ I_2 & C_2 & -C_2 & -I_2 \\ I_2 & -I_2 & -I_2 & I_2 \end{bmatrix} \quad (\text{A-8})$$

With each block being  $2 \times 2$  submatrices .Since

$$[C]_2 [C]_2^T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^T = [I]_2 \quad (\text{A-9})$$

$$[C]_2^{-1} = [C]_2^T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (\text{A-10})$$

We have

$$\begin{bmatrix} I_2 & I_2 & I_2 & I_2 \\ I_2 & -C_2 & C_2 & -I_2 \\ I_2 & C_2 & -C_2 & -I_2 \\ I_2 & -I_2 & -I_2 & I_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 & I_2 & I_2 \\ I_2 & -C_2^{-1} & C_2^{-1} & -I_2 \\ I_2 & C_2^{-1} & -C_2^{-1} & -I_2 \\ I_2 & -I_2 & -I_2 & I_2 \end{bmatrix} = \begin{bmatrix} 4I_2 & 0 & 0 & 0 \\ 0 & 4I_2 & 0 & 0 \\ 0 & 0 & 4I_2 & 0 \\ 0 & 0 & 0 & 4I_2 \end{bmatrix} = 4I_8 \quad (\text{A-11})$$

The inverse of (A-8) is

$$[J]_8^{-1} = [J]_{2 \times 4}^{-1} = \frac{1}{4} \begin{bmatrix} I_2 & I_2 & I_2 & I_2 \\ I_2 & -C_2^{-1} & C_2^{-1} & -I_2 \\ I_2 & C_2^{-1} & -C_2^{-1} & -I_2 \\ I_2 & -I_2 & -I_2 & I_2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} I_2 & I_2 & I_2 & I_2 \\ I_2 & -C_2^T & C_2^T & -I_2 \\ I_2 & C_2^T & -C_2^T & -I_2 \\ I_2 & -I_2 & -I_2 & I_2 \end{bmatrix} \quad (\text{A-12})$$

Certainly,

$$[J]_8 [J]_8^{-1} = \frac{1}{4} \begin{bmatrix} 4I_2 & 0 & 0 & 0 \\ 0 & 4I_2 & 0 & 0 \\ 0 & 0 & 4I_2 & 0 \\ 0 & 0 & 0 & 4I_2 \end{bmatrix} = [I]_8 \quad (\text{A-13})$$

This choice of block weighting was indicated, to a large extent, by requirement of digital hardware simplicity. With the aid of Kronecker product and Hadamard matrices, the higher order of BIJM is given by the following recursive relation:

$$[J]_{2N} \equiv [J]_N \otimes [H]_2, N \geq 2 \quad (\text{A-14})$$

We are able to show that

$$[J]_{2N}^{-1} = \frac{1}{N} [J]_{2N}^T \quad (\text{A-15})$$

We can use the induction method to prove this assertion. From (A-12) and (A-15) hold for  $2N=8$ . Assume that (A-9) holds for N, i.e.,

$$[J]_N [J]_N^T = \frac{N}{2} [I]_N, [J]_N^{-1} = \frac{2}{N} [J]_N^T \quad (\text{A-16})$$

Now we show that (A-16) holds for  $2N$

$$[J]_{2N} [J]_{2N}^T = \frac{N}{2} [I]_N \otimes (2[I]_2) = N [I]_{2N} \quad (A-17)$$

Hence, (A-15) holds. Therefore, BIJM is a class of transforms which are simple to calculate and easily invert-

ed. Further, the inverse of BIJM can be written as follows:

$$[x]_{2N} = [J]_{2N}^{-1} [y]_{2N} = \frac{1}{N} [J]_{2N}^T [y]_{2N} \quad (A-18)$$

Certainly, the BIJM is the dimension extending of EIJM.

---

저 자 소 개

---



쉬 단 평

- 2010년 9월 : 전북대학교 정보통신공학과 석사과정
- 주관심분야 : 통신신호처리, MIMO-OFDM



신 태 철

- 2001년 8월 : 전북대학교 정보통신공학과 석사
- 2007년 ~ 현재 : 전북대학교 정보통신공학과 박사과정
- 관심분야 : 통신신호처리, MIMO-OFDM



단 위

- 2010년 3월 : 전북대학교 정보통신공학과 석사과정
- 주관심분야 : Precoding, MIMO



이 문 호

- 1984년 2월 : 전남대학교 전기공학과(공학박사, 통신기술사)
- 1985년 ~ 1986년 : University of Minnesota 포스트 닥터
- 1990년 : 동경대학교 공학박사
- 1970년 ~ 1980년 : 남양 MBC 송신소장
- 1980년 10월 ~ 2010년 2월 : 전북대학교 전자정보공학부 교수
- 2010년 2월 ~ 현재 : 전북대 WCU-II 연구책임 교수
- 주관심분야 : 정보 통신의 원형, 뿌리 찾는 연구