Arbitrary-Length Walsh-Jacket Transforms

Jian-Jiun Ding, Soo-Chang Pei, and Po-Hung Wu
Department of Electrical Engineering
National Taiwan University, Taipei
djj@cc.ee.ntu.edu.tw, pei@cc.ee.ntu.edu.tw, bz400@hotmail.com

***Abstract*—Due to the efficiency in implementation, the Walsh (Hadamard) transform plays an important role in signal analysis and communication. Recently, Lee generalized the Walsh transform into the Jacket transform. Since the entries of the Jacket transform can be $\pm 2^k$, it is more flexible than the Walsh transform. Both the Walsh transform and the Jacket transform are defined for the case where the length *N* is a power of 2. In this paper, we try to extend the Walsh transform and the Jacket transform to the case where *N* is not a power of 2. With the "folding extension algorithm" and the Kronecker product, the arbitrary-length Walsh-Jacket transform can be defined successfully. As the original Walsh and Jacket transforms, the proposed arbitrary-length Walsh-Jacket transform has fast algorithms and can always be decomposed into the 2-point Walsh-Jacket transforms. We also show the applications of the proposed arbitrary-length Walsh-Jacket transforms in step-like signal analysis and electrocardiogram (ECG) signal analysis.**

I. Introduction

The Walsh (Hadamard) transform [1] is defined as

$$\mathbf{W}_2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},\quad \mathbf{W}_4=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\\ 1 & -1 & 1 & -1\end{bmatrix}, \qquad (1)$$

$$\mathbf{W}_{2M}=\mathbf{P}_{2M}\left(\mathbf{W}_2\otimes\mathbf{W}_M\right), \qquad (2)$$

where $\mathbf{P}_{2M}[2s-1, s] = \mathbf{P}_{2M}[2s, M+s] = 1$ if s is odd,
$\mathbf{P}_{2M}[2s-1, M+s] = \mathbf{P}_{2M}[2s, s] = 1$ if s is even,
$\mathbf{P}_{2M}[m, n] = 0$ otherwise.

(Note that the index used in this paper starts at 1. Therefore, for an $N\times N$ matrix $\mathbf{A}$, the entries are $\mathbf{A}[m, n]$ where m = 1, 2, …, N and n = 1, 2, …, N.) The inverse of the Walsh transform is its transpose multiplied by N:

$$\mathbf{W}_N^{-1}=\frac{1}{N}\mathbf{W}_N^T. \qquad (3)$$

Since the Walsh transform requires only the multiplication of 1 (the exception is that the inverse Walsh transform needs the multiplication of $1/N$), it is very efficient in implementation. The Walsh transform is useful in signal expansion, electrocardiogram (ECG) signal analysis, data compression, image processing, feature extraction, and error control coding. More, in digital communication, it also plays an important role in code division multiple access (CDMA) [1][2][3].

Recently, Lee et. al. generalized the Walsh transform into the Jacket transform [4][5][6][7][8][9][10]. For example, the 4-point Jacket transform is defined as [3][4]:

$$\mathbf{J}_4=\begin{bmatrix}a & b & b & a\\ b & c & -c & -b\\ a & -b & -b & a\\ b & -c & c & -b\end{bmatrix},\quad a=2^{k_a},\quad b=2^{k_b},\quad c=2^{k_c}, \qquad (4)$$

and its inverse is

$$\mathbf{H}_4=\frac{1}{4}\begin{bmatrix}1/a & 1/b & 1/b & 1/a\\ 1/b & -1/c & 1/c & -1/b\\ 1/b & 1/c & -1/c & -1/b\\ 1/a & -1/b & -1/b & 1/a\end{bmatrix},\quad \mathbf{H}_4\mathbf{J}_4=\mathbf{I}. \qquad (5)$$

Note that, instead of ±1, and the entries of the Jacket transform can be a power of 2. Since the multiplication of $\pm 2^m$ can be implemented by the bit shifting operation, as the Walsh transform, the Jacket transform is also efficient in implementation.

The Jacket transform is more general and flexible than the Walsh transform. With proper assignment of parameters, the Jacket transform can work similar to the sinusoid transform [10]. The Jacket transform has been found to be useful in MIMO system analysis, spread spectrum communication, and CDMA [4][5][6][7][8][9]. The (2*M*)-point Jacket transform can be defined by the Kronecker product of the *M*-point Jacket transform and the 2-point Walsh transform:

$$\mathbf{J}_{2M}=\mathbf{P}_{2M}\left(\mathbf{W}_2\otimes\mathbf{J}_M\right), \qquad (6)$$

where $\mathbf{W}_2$ is the 2-point Walsh transform and $\mathbf{P}_{2M}$ is defined the same as that in (2).

The 2^k-point Walsh transform and the 2^k-point Jacket transform can be successfully defined form (2) and (6). However, there is a problem: How do we define these transforms with arbitrary number of points?

If one constrains that the entries of the Walsh transform should be 1 or −1, then the arbitrary-point Walsh transform is impossible to define. In this paper, we apply the concept of the Jacket transform and relax the constraint. We give the definition of the Walsh-Jacket transform, which is a discrete transform whose entries have the values of $\pm 2^m$ or zero. It can be viewed as a generalization of the Walsh transform and a further generalization of the Jacket transform.

In Section 2, we give the definition of the Walsh-Jacket transform. In Section 3, we show the general form of the 2-point and the 3-point Walsh-Jacket transforms. In Section 4, we propose the "folding extension algorithm" that can extend the *M*-point and the (M +1)-point Walsh-Jacket transform into the (2M +1)-point Walsh-Jacket transform. Therefore, the 5-point Walsh-Jacket transform can be derived from the 2-point and 3-point Walsh-Jacket transforms, the 7-point Walsh-Jacket transform can be derived from the 3-point and 4-point Walsh-Jacket transforms, and so on. With the folding extension method together with the Kronecker product, the arbitrary-point Walsh-Jacket transform can be derived successfully. In Section 5, we show the fast implementation algorithm of the Walsh-Jacket transform. The applications of the Walsh-Jacket transform in electrocardiogram (ECG) signal analysis and step-like signal analysis is given in Section 6.

II. Definition of the Walsh-Jacket Transform

First, we give the definition of the "Walsh-Jacket transform". The Walsh-Jacket transform is a discrete operation whose entries should satisfy the following constraints. Suppose that the transform matrices of the forward and the inverse N-point Walsh-Jacket transforms are $\mathbf{W_N}$ and $\mathbf{U_N}$, respectively, and their entries are denoted by $\mathbf{W_N}[m, n]$ and $\mathbf{U_N}[m, n]$ ($m, n = 1, 2, \ldots, N$). Then

(Constraint A) The entries of both the forward transform $\mathbf{W_N}$ and the inverse transform $\mathbf{U_N}$ should be powers of 2 or zero.

(Constraint B) $\mathbf{W_N}[m, n] = \pm\mathbf{W_N}[m, N+1-n]$ should be satisfied.

(Constraint C) As the original Walsh transform, the numbers of zero crossing of the rows of the Walsh-Jacket transform should be 0, 1, 2, 3, …, N–1, respectively. **The k^{th} row** of the Walsh-Jacket transform should have **k–1 zero-crossings**.

Note that, due to Constraints B and C, the Walsh-Jacket transform works similar to the Walsh transform. However, from Constraint A, we can see that the Walsh-Jacket transform relaxes the constraints that the entry should be ±1, which is required by the Walsh transform. Note that, since the multiplication of the power of 2 can be implemented by bit-shifting, the Walsh-Jacket transform can still be implemented in an efficient way.

Also note that the Walsh-Jacket transform is also more general than the Jacket transform [4][5], since the entries of the Walsh-Jacket transform can be zero and the constraint that $\mathbf{U_N}[m, n] = 1/\mathbf{W_N}[n, m]/N$ is not required. The Walsh transform and the Jacket transform with arbitrary number of points are hard to define, but it is possible to define the arbitrary-length Walsh-Jacket transform.

III. Generalized 2-Point and 3-Point Walsh-Jacket Transforms

A. 2-Point Walsh-Jacket Transform

First, we show the general form of the 2-point Walsh-Jacket transform. Due to the three constraints in Section 2, the 2-point Walsh-Jacket transform should have the form of:

$$\mathbf{W_2} = \begin{bmatrix} a & b \\ c & -d \end{bmatrix}, \tag{7}$$

where a, b, c and d are powers of 2 and positive.
Its inverse is

$$\mathbf{U_2} = \mathbf{W_2^{-1}} = \frac{1}{ad+bc}\begin{bmatrix} d & b \\ c & -a \end{bmatrix}. \tag{8}$$

Since the entries of $\mathbf{U_2}$ must also be powers of 2, the following constraint should be satisfied:

$$ad + bc = 2^q, \text{ where } q \text{ is an integer.} \tag{9}$$

Any 2×2 matrix that has the form as in (7) and satisfies the constraint in (9) will satisfy all the three constraints in Section 2 and can be treated as the **2-point Walsh-Jacket transform**. For example,

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{10}$$

$$\begin{bmatrix} 1 & 2 \\ 2 & -4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & -2 \end{bmatrix} \tag{11}$$

are all the special cases of the 2-point Walsh-Jacket transform.

B. 3-Point Walsh-Jacket Transform

From the three constraints in Section 2, the 3-point Walsh-Jacket transform should have the form of

$$\mathbf{W_3} = \begin{bmatrix} a & b & a \\ c & 0 & -c \\ d & -e & d \end{bmatrix}, \tag{12}$$

where a, b, c, d and e are powers of 2 and positive. Since

$$\mathbf{U_3} = \mathbf{W_3^{-1}} = \frac{1}{2c(bd+ae)}\begin{bmatrix} ce & bd+ae & bc \\ 2cd & 0 & -2ac \\ ce & -bd-ae & bc \end{bmatrix}. \tag{13}$$

Therefore, to make the entries of $\mathbf{U_3}$ to be the powers of 2, the following constraint should be satisfied.:

$$bd + ae = 2^q, \text{ where } q \text{ is an integer.} \tag{14}$$

Moreover, if $ae = k(bd)$, then $(1+k)bd = 2^q$. Since bd is a power of 2, $1+k$ must also be a power of 2. Therefore, the only possible value of k is 1 and

$$\boldsymbol{ae = bd} \tag{15}$$

should be satisfied.

Any 3×3 matrix that has the form as in (12) and satisfy the constraint in (15) will satisfy all the three requirements in Section 2 and can be treated as the **3-point Walsh-Jacket transform**. For example, we can choose $\{a, b, c, d, e\} = \{1, 2, 1, 1, 2\}$ in (12). Then

$$\mathbf{W_3} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix} \tag{16}$$

is a special case of the 3-point Walsh-Jacket transform and its inverse is

$$\mathbf{U_3} = \mathbf{W_3^{-1}} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & -1/4 \\ 1/4 & -1/2 & 1/4 \end{bmatrix}. \tag{17}$$

We can see that all the entries of $\mathbf{W_3}$ and $\mathbf{U_3}$ are either powers of 2 or zero. Moreover, the 1st and the 3rd rows of $\mathbf{W_3}$ are even symmetric and the 2nd row of $\mathbf{W_3}$ is odd symmetric. Also note that the 1st, the 2nd, and the 3rd rows in (16) have 0, 1, and 2 zero-crossings, respectively. Therefore, all the three constraints in Section 2 are satisfied. We can also choose $\{a, b, c, d, e\} = \{1, 1, 1, 1, 1\}$ and the obtain the forward and inverse 3-point Walsh-Jacket transforms as follows:

$$\mathbf{W_3} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 1 \end{bmatrix}, \quad \mathbf{U_3} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ 1/4 & -1/2 & 1/4 \end{bmatrix}. \quad (18)$$

They also satisfy all the three requirements in Section 2.

IV. Arbitrary-Length Walsh-Jacket Transform

We derived the general forms of the 2-point and the 3-point Walsh-Jacket transforms in the previous section. In this section, we introduce two important theories that are helpful for deriving the arbitrary-length Walsh-Jacket transform.

A. *Folding Extension Algorithm for Odd N*

[Theorem 1] If N is an **odd** number, then we can use the following *folding extension algorithm* to derive the N-point Walsh-Jacket transform from the M-point and the (M+1)-point Walsh Jacket transforms, where $N = 2M+1$.

Suppose that $\mathbf{W_M}$ and $\mathbf{W_{M+1}}$ are the transform matrices of the M-point and the (M+1)-point Walsh Jacket transforms and $\mathbf{U_M}$ and $\mathbf{U_{M+1}}$ are their inverse transform matrices, respectively. Then the **(2*M*+1)-point Walsh-Jacket transform** can be derived from:

$$\mathbf{W_N} = \mathbf{P_N} \begin{bmatrix} \tilde{\mathbf{W}}_{\mathbf{M+1}} & 2\mathbf{c}_{\mathbf{M+1}} & \left(\tilde{\mathbf{W}}_{\mathbf{M+1}}\right)\leftarrow \\ \mathbf{W_M} & 0 & -\left(\mathbf{W_M}\right)\leftarrow \end{bmatrix}, \quad (19)$$

where $N = 2M+1$, $\tilde{\mathbf{W}}_{\mathbf{M+1}}$ means the first M columns of $\mathbf{W_{M+1}}$ and $\mathbf{c_{M+1}}$ is the last column of $\mathbf{W_{M+1}}$, i.e.,

$$\mathbf{W_{M+1}} = \left[\tilde{\mathbf{W}}_{\mathbf{M+1}} \quad \mathbf{c_{M+1}}\right], \quad (20)$$

and ( )← in (19) is the "column reverse operation". That is, if a matrix $\mathbf{A}$ contains K columns, then

$$B[m,n] = A[m, K+1-n] \quad \text{if } \mathbf{B} = (\mathbf{A})\leftarrow. \quad (21)$$

The permutation matrix $\mathbf{P_N}$ in (19) is

$\mathbf{P_N}[2s-1, s] = 1$ for $s = 1, 2, \ldots, M+1$,
$\mathbf{P_N}[2s, M+1+s] = 1$ for $s = 1, 2, \ldots, M$,
$\mathbf{P_N}[m, n] = 0$ otherwise. (22)

Moreover, the **inverse (2*M*+1)-point Walsh-Jacket transform** can be defined from:

$$\mathbf{U_N} = \frac{1}{2}\left[\begin{array}{c|c} \mathbf{U_{M+1}} & \begin{matrix} \mathbf{U_M} \\ 0 \end{matrix} \\ \hline \hat{\mathbf{U}}_{\mathbf{M+1}}\uparrow & -\mathbf{U_M}\uparrow \end{array}\right]\mathbf{P_N^T}, \quad (23)$$

where $\hat{\mathbf{U}}_{\mathbf{M+1}}$ is a matrix that contain the first M rows of $\mathbf{U_{M+1}}$ and ↑ means the "row reverse operation". That is, if $\mathbf{A}$ contains H rows, then

$$B[m,n] = A[H+1-m, n] \quad \text{if } \mathbf{B} = \mathbf{A}\uparrow. \quad (24)$$

From Theorem 1, we can generate the (2M+1)-point Walsh Jacket transform that satisfy all the requirements in Section 2 successfully. For example, we can use the 2-point and 3-point Walsh-Jacket transform to generate the 5-point Walsh-Jacket transform. If the 2-point Walsh-Jacket transform we use is defined as in (10) and the 3-point Walsh-Jacket transform we use is defined as in (16), then, from (19), obtained the **5-point Walsh Jacket transform** is

$$\mathbf{W_5} = \begin{bmatrix} 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 \\ 1 & -2 & 2 & -2 & 1 \end{bmatrix}. \quad (25)$$

From (23), the inverse 5-point Walsh Jacket transform is

$$\mathbf{U_5} = \begin{bmatrix} 1/8 & 1/4 & 1/4 & 1/4 & 1/8 \\ 1/8 & 1/4 & 0 & -1/4 & -1/8 \\ 1/8 & 0 & -1/4 & 0 & 1/8 \\ 1/8 & -1/4 & 0 & 1/4 & -1/8 \\ 1/8 & -1/4 & 1/4 & -1/4 & 1/8 \end{bmatrix}. \quad (26)$$

We can see that the rows of $\mathbf{W_5}$ in (25) are either even or odd symmetric and their numbers of zeros crossings are 0, 1, 2, 3, and 4, respectively. Both the 5-point Walsh transform and the 5-point Jacket transform are hard to define, but now we define the 5-point Walsh-Jacket transform successfully.

Furthermore, to derive the 7-point Walsh-Jacket transform, we can use the 4-point Jacket transform introduced by Lee [4]:

$$\mathbf{W_4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad (27)$$

together with the 3-point Walsh-Jacket transform in (16). From (19) and (23), the derived forward and inverse **7-point Walsh Jacket transforms** are

$$\mathbf{W_7} = \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 & -1 & -2 & -1 \\ 1 & 2 & -2 & -2 & -2 & 2 & 1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 2 & -1 & -1 & 1 \\ 1 & -2 & 1 & 0 & -1 & 2 & -1 \\ 1 & -2 & 2 & -2 & 2 & -2 & 1 \end{bmatrix}, \quad (28)$$

$$\mathbf{U_7} = \frac{1}{16}\begin{bmatrix} 2 & 2 & 2 & 4 & 2 & 2 & 2 \\ 2 & 2 & 1 & 0 & -2 & -2 & -1 \\ 2 & 2 & -1 & -4 & -2 & 2 & 1 \\ 2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 2 & -2 & -1 & 4 & -2 & -2 & 1 \\ 2 & -2 & 1 & 0 & -2 & 2 & -1 \\ 2 & -2 & 2 & -4 & 2 & -2 & 2 \end{bmatrix}. \quad (29)$$

It is not hard to show that the 7-point Walsh Jacket transform defined above satisfies all the three constraints in Section 2.

In Corollaries 1-4, we prove that $\mathbf{W_N}$ and $\mathbf{U_N}$ derived in Theorem 1 form a forward and inverse transform pair and they are bound to satisfy all the three constraints in Section2.

[Corollary 1] $\mathbf{W_N}$ and $\mathbf{U_N}$ defined in (19) and (23) form a forward and inverse transform pair, i.e.,

$$\mathbf{U_N}\mathbf{W_N} = \mathbf{I}. \tag{30}$$

(Proof): If we use $\mathbf{w_{N,m}}[n]$ to denote the m^{th} row of $\mathbf{W_N}$ and use $\mathbf{u_{N,p}}[q]$ to denote the p^{th} column of $\mathbf{U_N}$, then, since $\mathbf{w_{N,m}}[n]$ and $\mathbf{u_{N,p}}[q]$ are even symmetric if $m, p \le M+1$ and $\mathbf{w_{N,m}}[n]$ and $\mathbf{u_{N,p}}[q]$ are odd symmetric if $m, p > M+1$,

$$\sum_{n=1}^{N}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] = 0 \ \text{ if } m \le M+1 \text{ and } p > M+1 \text{ or } m > M+1 \text{ and } p \le M+1. \tag{31}$$

In the case where $m \le M+1$ and $p \le M+1$, from (19) and (23),

$$\begin{aligned}
&\sum_{n=1}^{N}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] \\
&= \sum_{n=1}^{M}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] + \mathbf{w_{N,m}}[M+1]\mathbf{u_{N,p}}[M+1] + \sum_{n=M+2}^{N}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] \\
&= \frac{1}{2}\sum_{n=1}^{M}\mathbf{w_{M+1,m}}[n]\mathbf{u_{M+1,p}}[n] + 2\mathbf{w_{M+1,m}}[M+1]\cdot\frac{1}{2}\mathbf{u_{M+1,p}}[M+1] \\
&\quad + \frac{1}{2}\sum_{n=1}^{M}\mathbf{w_{M+1,m}}[M+1-n]\mathbf{u_{M+1,p}}[M+1-n] \\
&= \sum_{n=1}^{M}\mathbf{w_{M+1,m}}[n]\mathbf{u_{M+1,p}}[n].
\end{aligned}$$

Since $\mathbf{U_{M+1}}\mathbf{W_{M+1}} = \mathbf{I}$, i.e., $\sum_{n=1}^{M}\mathbf{w_{M+1,m}}[n]\mathbf{u_{M+1,p}}[n] = \delta_{m,p}$, therefore,

$$\sum_{n=1}^{N}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] = \delta_{m,p} \tag{32}$$

is satisfied for $m \le M+1$ and $p \le M+1$.

When $m > M+1$ and $p > M+1$, from (19) and (23),

$$\begin{aligned}
\sum_{n=1}^{N}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] &= \sum_{n=1}^{M}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] + \sum_{n=M+2}^{N}\mathbf{w_{N,m}}[n]\mathbf{u_{N,p}}[n] \\
&= \frac{1}{2}\sum_{n=1}^{M}\mathbf{w_{M,m-M-1}}[n]\mathbf{u_{M,p-M-1}}[n] \\
&\quad + \frac{1}{2}\sum_{n=1}^{M}\mathbf{w_{M,m-M-1}}[M+1-n]\mathbf{u_{M,p-M-1}}[M+1-n] \\
&= \sum_{n=1}^{M}\mathbf{w_{M,m-M-1}}[n]\mathbf{u_{M+1,p-M-1}}[n] \\
&= \delta_{m,p}.
\end{aligned} \tag{33}$$

From (31), (32), and (33), we can see that the inner product of $\mathbf{w_{N,m}}[n]$ and $\mathbf{u_{N,p}}[q]$ is $\delta_{m,p}$ in all the cases and $\mathbf{U_N}\mathbf{W_N} = \mathbf{I}$. #

[Corollary 2] From (19) and (23), since the entries of $\mathbf{W_N}$ and $\mathbf{U_N}$ are either zero or equal to the entries of $\mathbf{W_M}$, $\mathbf{W_{M+1}}$, $\mathbf{U_M}$, $\mathbf{U_{M+1}}$ multiplied by ±1, 2, or ±1/2, we can conclude that their values are either zero or the powers of 2, which satisfy Constraint A in Section 2.

[Corollary 3] Moreover, from (19), it is obviously that

$$\begin{aligned}
&\mathbf{W_N}[m, n] = \mathbf{W_N}[m, N+1-n] \ \text{ if } m \le M+1, \\
&\mathbf{W_N}[m, n] = -\mathbf{W_N}[m, N+1-n] \ \text{ if } m > M+1,
\end{aligned} \tag{34}$$

which satisfy Constraint B in Section 2.

[Corollary 4] To prove that $\mathbf{W_N}$ in (19) satisfies Constraint C in Section 2, we can set

$$\mathbf{V_N} = \begin{bmatrix} \tilde{\mathbf{W}}_{\mathbf{M+1}} & 2\mathbf{c_{M+1}} & \left(\tilde{\mathbf{W}}_{\mathbf{M+1}}\right)\leftarrow \\ \mathbf{W_M} & 0 & -\left(\mathbf{W_M}\right)\leftarrow \end{bmatrix}. \tag{35}$$

Note that the numbers of zero crossings of the rows of $\mathbf{W_{M+1}}$ are 0, 1, 2, …, and M. Therefore, from (35), the numbers of zero crossings of the first M+1 rows of $\mathbf{V_N}$ are 0, 2, 4, …, and 2M, respectively. For the last M rows of $\mathbf{V_N}$, since there is a zero crossing at the center and the numbers of zero crossings of $\mathbf{W_M}$ are 0, 1, 2, …, and M–1, the last M rows of $\mathbf{V_N}$ have 1, 3, 5, …, and 2M–1 zero crossings, respectively.

Since $\mathbf{W_N} = \mathbf{P_N}\mathbf{V_N}$, where $\mathbf{P_N}$ is defined as in (22), we can see that $\mathbf{W_N}$ satisfies Constraint C in Section 2.

From Corollaries 1-4, we can see that one can indeed use the folding extension algorithm in Theorem 1 to derive the N-point Walsh-Jacket transform that satisfies all the constraints in Section 2 from the M-point and the (M+1)-point Walsh-Jacket transforms, where $M = (N–1)/2$ and N is odd.

To derive the N-point Walsh-Jacket transform where N is even, one can use the Kronecker product algorithm introduced in the following subsection.

B. Kronecker Product Algorithm for Even N

[Theorem 2] If N is **even** and

$$N = 2^k \cdot H, \tag{36}$$

where H is an odd number and k is a positive integer, then we can define the ***N*-point Walsh-Jacket transform** from the *Kronecker product algorithm*:

$$\mathbf{W_N} = \mathbf{P_N}(\mathbf{W_{2^k}} \otimes \mathbf{W_H}), \tag{37}$$

where $\mathbf{W_H}$ and $\mathbf{W_{2^k}}$ are the H-point and the 2^k-point Walsh-Jacket transform matrices. The row permutation matrix $\mathbf{P_N}$ in (37) is defined as

$$\begin{aligned}
&\mathbf{P_N}[(n_2-1)2^k + n_1 + 1,\ n_1H + n_2] = 1 \ \text{ if } n_2 \text{ is odd}, \\
&\mathbf{P_N}[(n_2-1)2^k + 2^k - n_1,\ n_1H + n_2] = 1 \ \text{ if } n_2 \text{ is even}, \\
&\qquad n_1 = 0, 1, \dots, 2^k - 1,\ n_2 = 1, 2, \dots, H, \\
&\mathbf{P_N}[m, n] = 0 \ \text{ otherwise}.
\end{aligned} \tag{38}$$

Moreover, the **inverse *N*-point Walsh-Jacket transform** can be defined from

$$\mathbf{U_N} = (\mathbf{U_{2^k}} \otimes \mathbf{U_H})\mathbf{P_N^T}, \tag{39}$$

where $\mathbf{U_H}$ and $\mathbf{U_{2^k}}$ are the H-point and the 2^k-point inverse Walsh-Jacket transform matrices.

For example, from Theorem 2, the 6-point Walsh-Jacket transform can be derived from the 2-point and 3-point Walsh-

Jacket transforms. If the 2-point one is defined as in (10) and the 3-point one is defined as in (18), then, from (37) and (39), the **6-point Walsh-Jacket transform** we obtain is

$$\mathbf{W_6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 0 & -1 & -1 & 0 & 1 \\ 1 & 0 & -1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}, \qquad (40)$$

and its inverse is

$$\mathbf{U_6} = \frac{1}{8}\begin{bmatrix} 1 & 1 & 2 & 2 & 1 & 1 \\ 2 & 2 & 0 & 0 & -2 & -2 \\ 1 & 1 & -2 & -2 & 1 & 1 \\ 1 & -1 & -2 & 2 & 1 & -1 \\ 2 & -2 & 0 & 0 & -2 & 2 \\ 1 & -1 & 2 & -2 & 1 & -1 \end{bmatrix}. \qquad (41)$$

Similarly, we can also use Theorem 2 together with the 2-point and the 5-point Walsh-Jacket transforms defined as in (10) and (25) to derive the 10-point Walsh-Jacket transform. From (37) and (39), the forward and inverse the **10-point Walsh-Jacket transforms** are

$$\mathbf{W_{10}} = \begin{bmatrix} 1 & 2 & 2 & 2 & 1 & 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 & -1 & -2 & -2 & -2 & -1 \\ 1 & 1 & 0 & -1 & -1 & -1 & -1 & 0 & 1 & 1 \\ 1 & 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 & -1 \\ 1 & 0 & -2 & 0 & 1 & 1 & 0 & -2 & 0 & 1 \\ 1 & 0 & -2 & 0 & 1 & -1 & 0 & 2 & 0 & -1 \\ 1 & -1 & 0 & 1 & -1 & -1 & 1 & 0 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 1 & -1 & 0 & 1 & -1 \\ 1 & -2 & 2 & -2 & 1 & 1 & -2 & 2 & -2 & 1 \\ 1 & -2 & 2 & -2 & 1 & -1 & 2 & -2 & 2 & -1 \end{bmatrix}, \qquad (42)$$

$$\mathbf{U_{10}} = \frac{1}{16}\begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 2 & 0 & 0 & -2 & -2 & -1 & -1 \\ 1 & 1 & 0 & 0 & -2 & -2 & 0 & 0 & 1 & 1 \\ 1 & 1 & -2 & -2 & 0 & 0 & 2 & 2 & -1 & -1 \\ 1 & 1 & -2 & -2 & 2 & 2 & -2 & -2 & 1 & 1 \\ 1 & -1 & -2 & 2 & 2 & -2 & -2 & 2 & 1 & -1 \\ 1 & -1 & -2 & 2 & 0 & 0 & 2 & -2 & -1 & 1 \\ 1 & -1 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 \\ 1 & -1 & 2 & -2 & 0 & 0 & -2 & 2 & -1 & 1 \\ 1 & -1 & 2 & -2 & 2 & -2 & 2 & -2 & 1 & -1 \end{bmatrix}. \qquad (43)$$

It is not hard to see that the derived 6-point and 10-point Walsh-Jacket transforms in (40) and (42) satisfy all the constraints in Section 2. In Corollaries 5-8, we will prove that $\mathbf{W_N}$ and $\mathbf{U_N}$ derived in Theorem 2 can always form a forward and inverse transform pair and they are bound to satisfy all the three constraints in Section2.

[Corollary 5] The forward and inverse Walsh-Jacket transforms defined in (37) and (39) form a reversible transform pair (i.e., $\mathbf{U_N W_N} = \mathbf{I}$).

(Proof): From the facts that $\mathbf{P_N}^{\mathbf{T}}\mathbf{P_N} = \mathbf{I}$ and that $(\mathbf{A_1}\otimes\mathbf{B_1})(\mathbf{A_2}\otimes\mathbf{B_2}) = (\mathbf{A_1A_2}) \otimes (\mathbf{B_1B_2})$, we can prove that

$$\mathbf{U_N W_N} = (\mathbf{U_{2^k}} \otimes \mathbf{U_H})\mathbf{P_N^T P_N}(\mathbf{W_{2^k}} \otimes \mathbf{W_H})$$
$$= \left(\mathbf{U_{2^k} W_{2^k}}\right) \otimes \left(\mathbf{U_H W_H}\right) = \mathbf{I} \otimes \mathbf{I} = \mathbf{I}. \qquad \#$$

[Corollary 6] Since the entries of $\mathbf{W_H}$, $\mathbf{W_{2^k}}$, $\mathbf{U_H}$, and $\mathbf{U_{2^k}}$ are either zero or powers of 2, the entries of the matrices obtained by their Kronecker products should also be zero or powers of 2, which satisfies Constraint A in Section 2.

[Corollary 7] To prove that $\mathbf{W_N}$ derived from (37) satisfies Constraint B in Section 2, we can apply the fact that

$$\mathbf{W_H}\left[m,n\right] = (-1)^{m-1}\,\mathbf{W_H}\left[m, H+1-n\right],$$
$$\mathbf{W_{2^k}}\left[m,n\right] = (-1)^{m-1}\,\mathbf{W_{2^k}}\left[m, 2^k+1-n\right].$$

Therefore, if

$$\mathbf{V_N} = \mathbf{W_{2^k}} \otimes \mathbf{W_H}, \qquad (44)$$

then

$$\mathbf{V_N}\left[\tau,n\right] = (-1)^{\tau_1+\tau_2}\,\mathbf{V_N}\left[\tau, N+1-n\right],$$

where $\tau_1 = \lceil \tau/H \rceil - 1$, $\tau_2 = \tau - \tau_1 H$, (45)

and ($\lceil\ \rceil$ means rounding toward infinity. Then, from (38), we can see that

$$\mathbf{W_N}\left[m,n\right] = \mathbf{V_N}\left[(m_2-1)H + m_1 + 1, n\right] \quad \text{when } m_1 \text{ is even}, \qquad (46)$$

$$\mathbf{W_N}\left[m,n\right] = \mathbf{V_N}\left[(2^k - m_2)H + m_1 + 1, n\right] \quad \text{when } m_1 \text{ is odd}. \qquad (47)$$

where $m_1 = \lceil m/2^k \rceil - 1$ and $m_2 = m - m_1 2^k$. Therefore,

$$\mathbf{W_N}\left[m,n\right] = (-1)^{m_2-1+m_1+1}\,\mathbf{W_N}\left[m, N+1-n\right]$$
$$= (-1)^{m_1 2^k + m_2}\,\mathbf{W_N}\left[m, N+1-n\right] = (-1)^N\,\mathbf{W_N}\left[m, N+1-n\right]$$

when m_1 is even,

$$\mathbf{W_N}\left[m,n\right] = (-1)^{2^k - m_2 + m_1 + 1}\,\mathbf{W_N}\left[m, N+1-n\right]$$
$$= -(-1)^{m_2+m_1}\,\mathbf{W_N}\left[m, N+1-n\right]$$
$$= (-1)^{m_1 2^k + m_2}\,\mathbf{W_N}\left[m, N+1-n\right] = (-1)^N\,\mathbf{W_N}\left[m, N+1-n\right]$$

when m_1 is odd.

and Constraint B in Section 2 is satisfied.

[Corollary 8] To prove that $\mathbf{W_N}$ derived from Theorem 2 satisfies Constraint C in Section 2, we can apply the fact that

$$\mathbf{V_N}\left[\tau, kH\right] = -\mathbf{V_N}\left[\tau, kH+1\right]$$
$$\text{if } \mathbf{W_{2^k}}\left[\tau_1+1, k\right] = -(-1)^{\tau_2-1}\,\mathbf{W_{2^k}}\left[\tau_1+1, k+1\right],$$
$$\mathbf{V_N}\left[\tau, kH\right] = \mathbf{V_N}\left[\tau, kH+1\right] \quad \text{otherwise}, \qquad (48)$$

where $\mathbf{V_N}$, τ_1, and τ_2 are defined the same as those in (44) and (45) and k is ant integer. Therefore, we can conclude that the τ^{th} row ($\tau = \tau_1 H + \tau_2$) of $\mathbf{V_N}$ should have

$(\tau_2 - 1)2^k + \tau_1$ zero crossings when τ_2 is odd, (49)

$$(\tau_2 - 1)2^k + 2^k - 1 - \tau_1 \text{ zero crossings} \quad \text{when } \tau_2 \text{ is even.} \tag{50}$$

Then, from (46), (47), (49), and (50), we can see that

(Case 1): When m_1 is even, from (46) and (45), we can see that $\tau_1 = m_2 - 1$ and $\tau_2 = m_1 + 1$. Since τ_2 is odd, from (49), the number of zero crossings is

$$m_1 2^k + m_2 - 1 = m - 1.$$

(Case 2): When m_1 is odd, from (47) and (45), $\tau_1 = 2^k - m_2$ and $\tau_2 = m_1 + 1$. Since τ_2 is even. From (50), the number of zero crossings is

$$m_1 2^k + 2^k - 1 - (2^k - m_2) = m_1 2^k + m_2 - 1 = m - 1.$$

Therefore, in both cases, the m^{th} row of $\mathbf{W_N}$ has $m-1$ zero crossings, which satisfies Constraint C in Section 2. #

Therefore, from Corollaries 5-8, the Walsh-Jacket transform derived from Theorem 2 satisfies all the three constraints in Section 2.

TABLE I
THE WAYS TO GENERATE THE N-POINT WALSH-JACKET TRANSFORM FOR N = 5~39, WHERE M_1 AND M_2 MEANS THAT THE N-POINT WALSH-JACKET TRANSFORM CAN BE GENERATED FROM THE M_1-POINT AND M_2-POINT WALSH-JACKET TRANSFORMS BY THE METHOD LISTED IN THE 4TH COLUMN

N	M_1	M_2	Method
5	2	3	Theorem 1
6	2	3	Theorem 2
7	3	4	Theorem 1
9	4	5	Theorem 1
10	2	5	Theorem 2
11	5	6	Theorem 1
12	4	3	Theorem 2
13	6	7	Theorem 1
14	2	7	Theorem 2
15	7	8	Theorem 1
17	8	9	Theorem 1
18	2	9	Theorem 2
19	9	10	Theorem 1
20	4	5	Theorem 2
21	10	11	Theorem 1
22	2	11	Theorem 2
23	11	12	Theorem 1
24	8	3	Theorem 2
25	12	13	Theorem 1
26	2	13	Theorem 2
27	13	14	Theorem 1
28	4	7	Theorem 2
29	14	15	Theorem 1
30	2	15	Theorem 2
31	15	16	Theorem 1
33	16	17	Theorem 1
34	2	17	Theorem 2
35	17	18	Theorem 1
36	4	9	Theorem 2
37	18	19	Theorem 1
38	2	19	Theorem 2
39	19	20	Theorem 1

[Theorem 3] Combine Theorems 1 and 2, we can derive the **arbitrary-length** Walsh-Jacket transform successfully.

When N is odd, we can apply the "folding extension algorithm" in Theorem 1 to derive the N-point Walsh Jacket transform from the M-point and the (M+1)-point Walsh-Jacket transforms, where $N = 2M+1$.

When N is even, we can use the "Kronecker product algorithm" to derive the N-point Walsh-Jacket transform from the 2^k-point and the H-point Walsh-Jacket transforms, where $N = 2^k \cdot H$ and H is odd.

No matter what the value of N is, we can derive the N-point Walsh-Jacket transform successfully by **Theorems 1 and 2 and iterative decomposition**.

In Table 1, we summarize the way to generate the N-point Walsh Jacket transform for N = 5~39. For example, when N = 11, the values of M_1 and M_2 are 5 and 6 and the corresponding method is Theorem 1. It means that the 11-point Walsh-Jacket transform can be derived from the 5-point and the 6-point Walsh Jacket transform by Theorem 1. Furthermore, the 5-point Walsh-Jacket transform can be derived from the 2-point and the 3-point Walsh-Jacket transforms by Theorem 1 and the 6-point Walsh-Jacket transform can be derived from the 2-point and the 3-point Walsh-Jacket transforms by Theorem 2. If the 2-point and the 3-point Walsh-Jacket transforms we choose are as in (10) and (16), then the obtained forward and inverse 11-point Walsh Jacket transforms are

$$\mathbf{W_{11}} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 & 0 & -1 & -2 & -2 & -2 & -1 \\ 1 & 2 & 1 & -1 & -2 & -2 & -2 & -1 & 1 & 2 & 1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 & 0 & -1 & -1 \\ 1 & 0 & -1 & -1 & 0 & 2 & 0 & -1 & -1 & 0 & 1 \\ 1 & 0 & -2 & 0 & 1 & 0 & -1 & 0 & 2 & 0 & -1 \\ 1 & 0 & -1 & 1 & 0 & -2 & 0 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -2 & 1 & 1 & -2 & 2 & -2 & 1 & 1 & -2 & 1 \\ 1 & -2 & 2 & -2 & 1 & 0 & -1 & 2 & -2 & 2 & -1 \\ 1 & -2 & 1 & -1 & 2 & -2 & 2 & -1 & 1 & -2 & 1 \end{bmatrix}, \tag{51}$$

$$\mathbf{U_{11}} = \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 & -2 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -2 & -2 & -2 & 0 & 1 & 1 & 1 \\ 1 & 1 & -1 & -2 & -2 & 0 & 2 & 2 & 1 & -1 & -1 \\ 1 & 1 & -1 & -2 & 0 & 2 & 0 & -2 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 & 2 & 0 & -2 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 2 & 0 & -2 & 0 & 2 & -1 & -1 & 1 \\ 1 & -1 & -1 & 2 & -2 & 0 & 2 & -2 & 1 & 1 & -1 \\ 1 & -1 & 1 & 0 & -2 & 2 & -2 & 0 & 1 & -1 & 1 \\ 1 & -1 & 1 & -2 & 0 & 0 & 0 & 2 & -1 & 1 & -1 \\ 1 & -1 & 1 & -2 & 2 & -2 & 2 & -2 & 1 & -1 & 1 \end{bmatrix}. \tag{52}$$

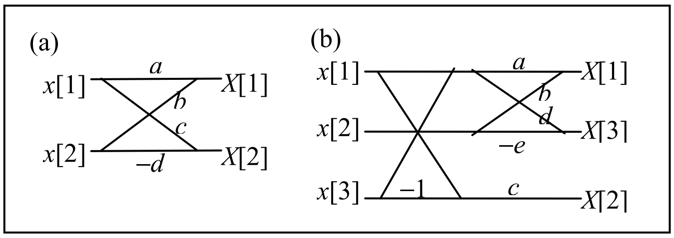

Fig. 1 The implementations of (a) the 2-point Walsh-Jacket transform and (b) the 3-point Walsh-Jacket transform.

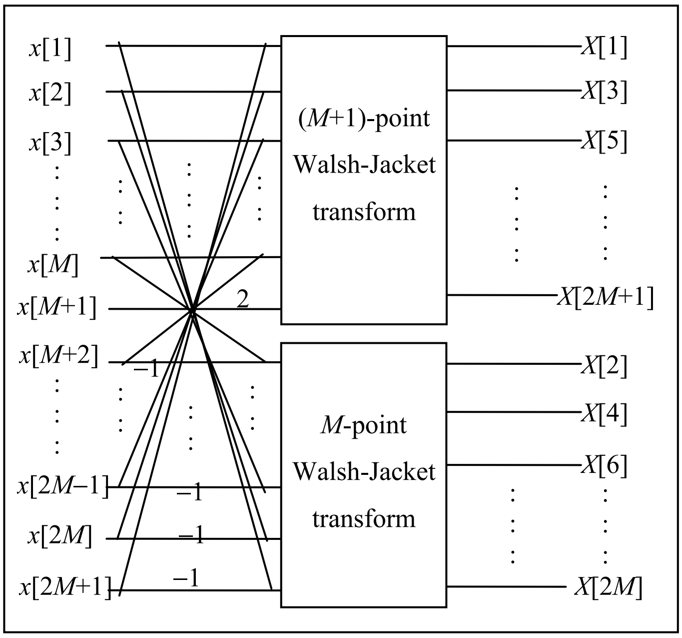

Fig. 2 Implementing the $(2M+1)$-point Walsh-Jacket transform by the M-point and the $(M+1)$-point Walsh-Jacket transforms.

V. FAST IMPLEMENTATION ALGORITHMS

As the 2^k-point Walsh transform and the 2^k-point Jacket transform, the arbitrary-length Walsh-Jacket transform also has the fast implementation algorithm. In fact, no matter what the value of N is, the N-point Walsh-Jacket transform can always be decomposed into **the combination of the 2-point Walsh-Jacket transforms**.

In Fig. 1, we show the implementation structures of the 2-point and the 3-point Walsh-Jacket transforms. Note that the 3-point Walsh Jacket transforms can be decomposed into two butterflies, i.e., two 2-point Walsh-Jacket transforms.

For the case where $N > 3$, we can use the methods as in Figs. 2 and 3 to decompose the N-point Walsh-Jacket transform into the smaller size Walsh-Jacket transforms. From Theorem 1, we can see that when N is odd, the N-point Walsh-Jacket transform can be decomposed into the M-point and the $(M+1)$-point Walsh-Jacket transforms by the method as in Fig. 2 ($N = 2M+1$). When N is even, from Theorem 2, the N-point Walsh-Jacket transform can be decomposed into the 2^k-point and the H-point Walsh-Jacket transforms by the method as in Fig. 3 ($N = 2^kH$ and H is odd).

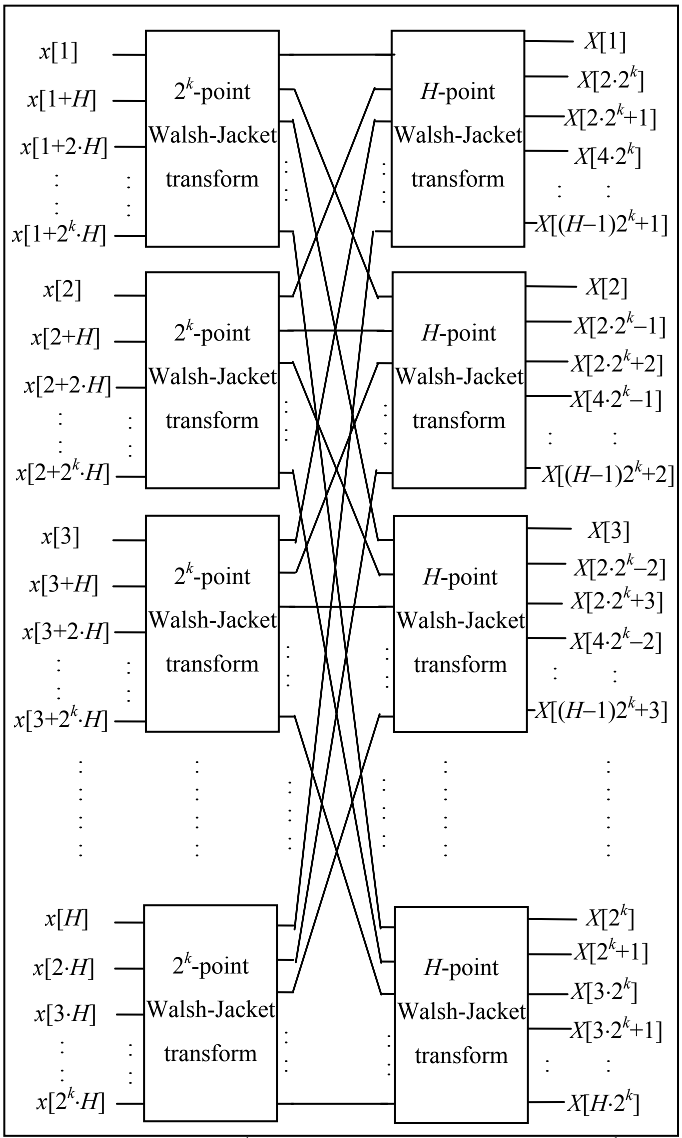

Fig. 3 Implementing the 2^kH-point Walsh-Jacket transform by the 2^k-point and the H-point Walsh-Jacket transforms.

No matter what the value of N is, we can use the structures in Figs. 2 and 3 to decompose the N-point Walsh-Jacket transform into the combination of the 2-point Walsh transform.

For example, for the 10-point Walsh-Jacket transform in (41), we can first use Fig. 3 to decompose it into the 2-point and the 5-point Walsh-Jacket transforms. Then, we use the structure in Fig. 2 to further decompose the 5-point Walsh-Jacket transform into the 2-point and 3-point Walsh Jacket Jackets. Then, from the structure in Fig. 1(b), the 3-point Walsh Jacket Jackets can be implemented by the 2-point Walsh-Jacket transforms. The whole fast implementation structure of the 10-point Walsh-Jacket transform is shown in Fig. 4. It shows that the 10-point Walsh-Jacket transform can be fully implemented by the combination of the 2×2 butterflies (i.e., the 2-point Walsh-Jacket transforms).

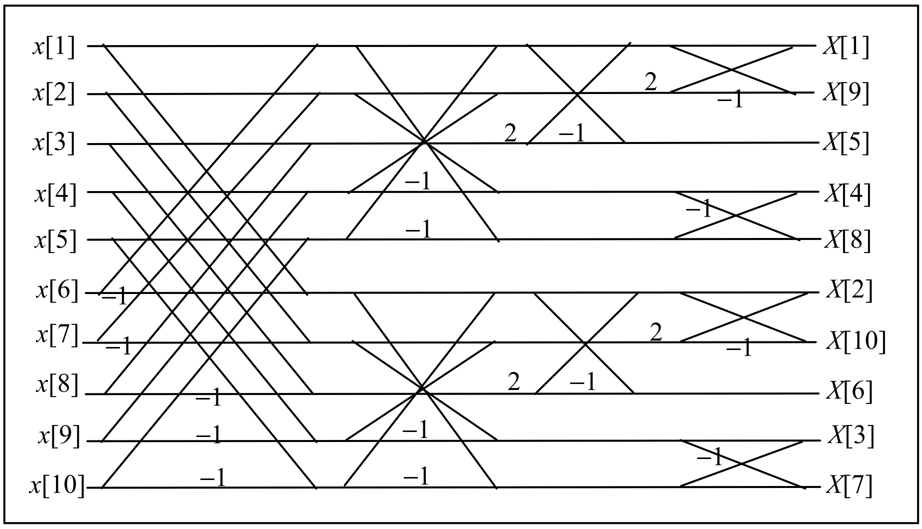

Fig. 4 The fast implementation algorithm of the 10-point Walsh-Jacket transform. The 10-point Walsh-Jacket transform can be fully decomposed into the 2×2 butterflies (i.e., the 2-point Walsh-Jacket transforms).

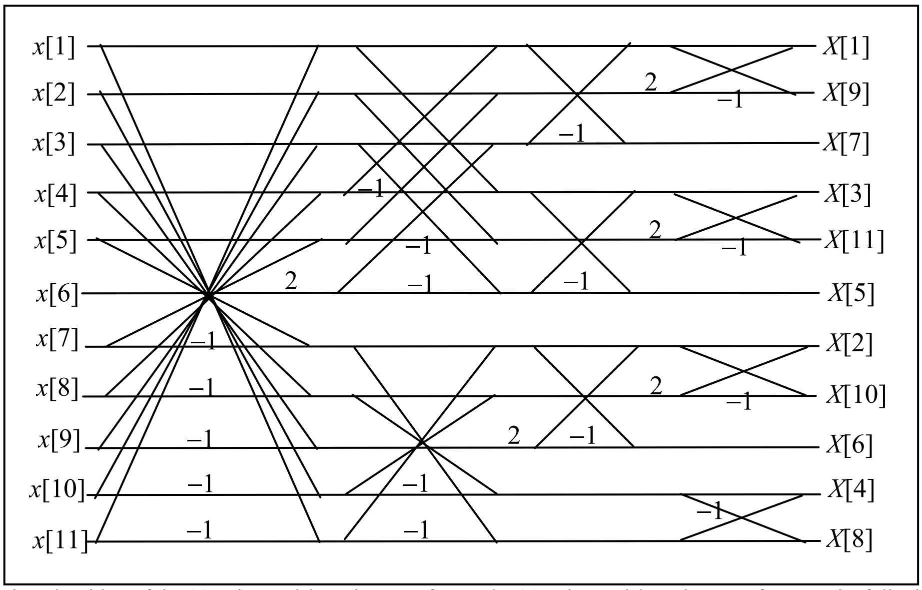

Fig. 5 The fast implementation algorithm of the 11-point Walsh-Jacket transform. The 11-point Walsh-Jacket transform can be fully decomposed into the 2-point Walsh-Jacket transforms.

Similarly, with iterative decomposition, the 11-point Walsh-Jacket transform in (51) can also be implemented by the combination of the 2-point Walsh-Jacket transforms, as in Fig. 5.

Moreover, as the forward transform, the inverse Walsh-Jacket transform also has the fast implementation algorithm and we can decompose the *N*-point inverse Walsh-Jacket transform iteratively into the 2-point Walsh-Jacket transforms, no matter what the value of *N* is.

VI. Application in Signal Analysis

As the 2^k-point Walsh transform and the Jacket transform, the proposed arbitrary-point Walsh-Jacket transform can also be applied in signal analysis, feature extraction, and CDMA.

In fact, all applications of the original Walsh transform and the Jacket transform can also be viewed as the applications of the proposed arbitrary-point Walsh-Jacket transform.

In Figs. 6-9, we perform some simulations that use the proposed Walsh-Jacket transform for electrocardiogram (ECG) signal analysis. The length of the ECG signal in Fig. 6 is 188, which is not a power of 2. It is hard to analyze by the Walsh transform, but can be analyzed by the proposed arbitrary-point Walsh-Jacket transform. In Fig. 7, we show the normalized mean square error (NMSE) of the reconstructed signal when using part of the coefficients of the 188-point Walsh-Jacket transform:

$$NMSE = \frac{\|\mathbf{x_S} - \mathbf{x}\|^2}{\|\mathbf{x}\|^2} \qquad \mathbf{x}\text{: the original signal,} \tag{53}$$

$$\mathbf{y} = \mathbf{W_N x}, \qquad \mathbf{x_S} = \mathbf{U_N y_S},$$

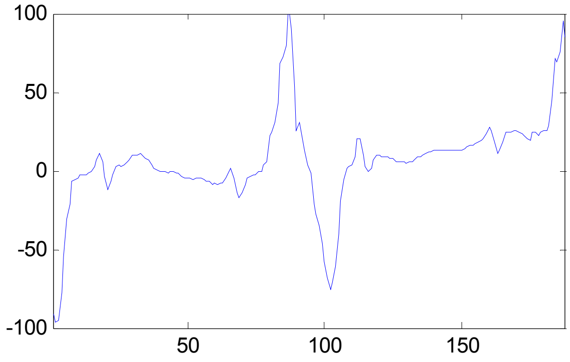

Fig. 6 A 188-length electrocardiogram (ECG) signal.

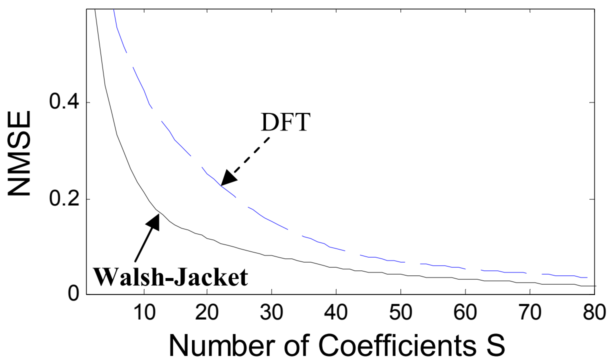

Fig. 7 The normalized mean square error (NMSE) when using part of the coefficients to reconstruct the ECG signal in Fig. 6. Solid line: using the 188-point Walsh-Jacket transform; dash line: using the DFT.

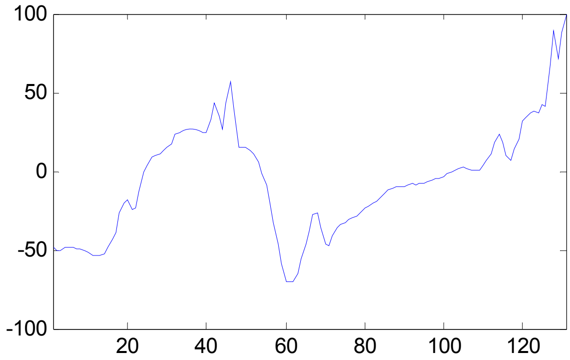

Fig. 8 A 131-length electrocardiogram (ECG) signal.

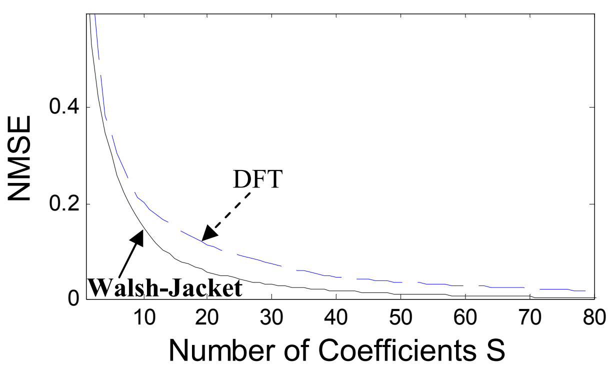

Fig. 9 The NMSE when using part of the coefficients to reconstruct the ECG signal in Fig. 8. Solid line: using the 131-point Walsh-Jacket transform; dash line: using the DFT.

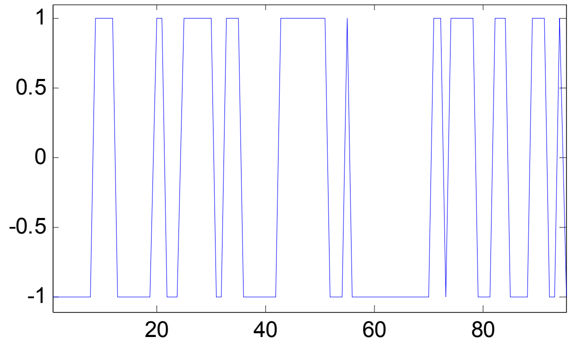

Fig. 10 A 95-point step-like signal.

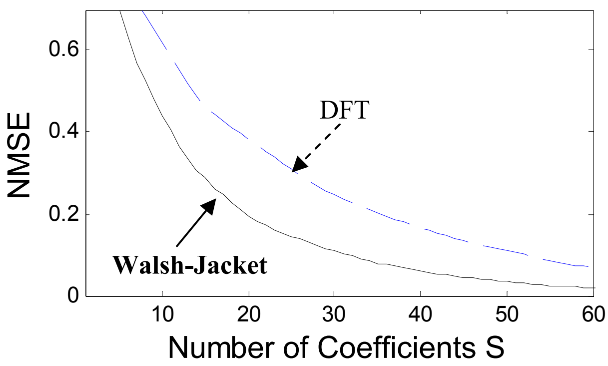

Fig. 11 The NMSE when using part of the coefficients to reconstruct the step-like signal in Fig. 10. Solid line: using the 95-point Walsh-Jacket transform; dash line: using the DFT.

where $\mathbf{W}_\mathbf{N}$ and $\mathbf{U}_\mathbf{N}$ are the forward and inverse Walsh-Jacket transform matrices. The vector $\mathbf{y}_\mathbf{S}$ preserves S coefficients of $\mathbf{y}$ and others are set to zero. In Fig. 7, we also show the NMSE of the reconstructed signal when using the 188-point discrete Fourier transform (DFT). The results in Fig. 7 show that, with the Walsh-Jacket transform, we can achieve the less approximation error when using only S terms (S < 188) to expand the ECG signal.

In Figs. 8 and 9, we give another simulation. The length of the ECG in Fig. 8 is 131. In Fig. 9, we show the NMSE of the reconstructed signal when using part of the coefficients of the 131-point Walsh-Jacket transform and the 131-point DFT. The results also show that the Walsh-Jacket transform is more effective for analyzing the ECG signal.

Moreover, as the description in [1][2], in addition to ECG signal analysis, the Walsh transform is also effective for analyzing the step-like signal. In Fig. 10, we show a 95-length step-like signal. Then, we use the 95-point Walsh-Jacket transform and the 95-point DFT to analyze the signal and plot the NMSE of the reconstructed signal when using part of the coefficients for reconstruction in Fig. 11. It is obviously that the error of the proposed Walsh-Jacket transform is less. Therefore, as the original Walsh transform, the proposed arbitrary-point Walsh-Jacket transform is also effective for step-like signal analysis.

Furthermore, as the original Walsh transform and the Jacket transform can be applied for MIMO system analysis, CDMA, and data encryption, they are also the potential applications of the proposed Walsh-Jacket transform. Since the Walsh-Jacket transform can be defined for arbitrary length, it is more flexible than the Walsh transform and the Jacket transform in these applications.

VII. Conclusions

In this paper, we generalize the Walsh transform and the Jacket transform into the case where the number of points N is not a power of 2. With the folding extension algorithm (i.e., Theorem 1) and the Kronecker product algorithm (Theorem 2), the arbitrary-length Walsh-Jacket transform can be defined successfully.

As the original Walsh transform and the Jacket transform, the proposed arbitrary-point Walsh-Jacket transform also has the fast implementation algorithm. We also show that the N-point Walsh-Jacket transform can always be implemented by the combination of the 2-point Walsh transforms, no matter what the value of N is. Since the proposed Walsh-Jacket transform can be viewed as the arbitrary-point version of the Walsh transform and the Jacket transform, the applications of the Walsh transform and the Jacket transform (such as ECG signal analysis, step-like signal analysis, and digital communication) are also the applications of the proposed Walsh-Jacket transform.

References

[1] K. G. Beanchamp, *Walsh Functions and Their Applications*, Academic Press, New York, 1975.

[2] H. F. Harmuth, "Applications of Walsh functions in communications," *IEEE Spectrum*, vol. 6, no. 11, pp. 82-91, Nov. 1969.

[3] R. Rao and S. Dianat, "Basic of code division multiple access," *Tutorial Text in Optical Engineering*, vol. TT67, Bellingham, Wash, SPIE 2005.

[4] S. R. Lee and M. H. Lee, "On the reverse Jacket matrix for weighted Hadamard transform," *IEEE Trans. Circuit Syst. II*, vol. 45, no. 1, pp. 436-441, March 1998.

[5] M. H. Lee, "A new reverse Jacket transform and its fast algorithm," *IEEE Trans. Circuit Syst. II*, vol. 47, no. 1, pp. 39-47, Jan. 2000.

[6] M. H. Lee, B. S. Rajan, and J. Y. Park, "A generalized reverse Jacket transform," *IEEE Trans. Circuits Syst. II*, vol. 48, pp. 684–690, July 2001.

[7] K. J. Horadam, "A generalised Hadamard transform," *ISIT*, pp. 1006-1008, 2005.

[8] G. Zeng and M. H. Lee, "A generalized reverse block Jacket transform," *IEEE Trans. Circuits Syst. I, Regul. Pap.*, vol. 55, no. 6, pp. 1589-1600, 2008.

[9] W. Song, M. H. Lee, and G. Zeng, "Orthogonal space-time block codes design using Jacket transform for MIMO transmission system," *IEEE International Conference on Communication*, pp. 766-769, 2008.

[10] S. C. Pei and J. J. Ding, "Generalizing the Jacket transform by sub orthogonality extension," *EUSIOCO*, pp. 408-412, Aug. 2009.