# Correspondence

## Fast Walsh–Hadamard–Fourier Transform Algorithm

Monir T. Hamood and Said Boussakta

***Abstract*—An efficient fast Walsh–Hadamard–Fourier transform algorithm which combines the calculation of the Walsh–Hadamard transform (WHT) and the discrete Fourier transform (DFT) is introduced. This can be used in Walsh–Hadamard precoded orthogonal frequency division multiplexing systems (WHT-OFDM) to increase speed and reduce the implementation cost. The algorithm is developed through the sparse matrices factorization method using the Kronecker product technique, and implemented in an integrated butterfly structure. The proposed algorithm has significantly lower arithmetic complexity, shorter delays and simpler indexing schemes than existing algorithms based on the concatenation of the WHT and FFT, and saves about 70%–36% in computer run-time for transform lengths of 16–4096.**

***Index Terms*—Discrete Fourier transform (DFT), fast Walsh–Fourier transform (FWFT), algorithm, Walsh–Hadamard transform (WHT).**

### I. Introduction

The combination of the Walsh–Hadamard transform (WHT) with the discrete Fourier transform (DFT) has attracted a lot of interest, and has proved to be a promising candidate for future wireless communication systems, specifically orthogonal frequency division multiplexing (OFDM) introducing the so-called (WHT-OFDM) system [1]–[5]. This technique has been found to provide considerable improvement in terms of bit error rate (BER) [5] and peak-to-average-power ratio (PAPR) for OFDM [6]–[8]. The range of applications of the combined WHT and DFT has also been extended beyond the conventional OFDM systems as shown in [9]–[14], to include multi-carrier code-division multiple access (MC-CDMA) [10], [13], [14], space-division multiple access (SDMA) [12], wireless LAN systems [11], [12], and the multi-band OFDM ultra-wideband (MB-OFDM-UWB) systems [9].

The conventional method for computing the WHT-DFT system is carried out based on successively computing the WHT and DFT using well-known fast algorithms. However, this method is not efficient because it requires the calculation of two transforms separately involving high number of arithmetic operations and indexing schemes.

In this work, the development of radix-4 fast Walsh–Hadamard–Fourier transform (FWFT) algorithm that combines the calculation of the two transforms into a single fast algorithm is introduced for the fast calculation of a sequence whose length equals to a power of four. It provides far more efficient performance when compared with the existing WHT-DFT algorithms.

Manuscript received April 18, 2011; revised July 05, 2011 and July 10, 2011; accepted July 10, 2011. Date of publication July 25, 2011; date of current version October 12, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jarmo H. Takala. This work was supported in part by the Royal Society of Engineering under Grant Ref. 10508/366 and EPSRC under Grant GR/S98160/02.

The authors are with the School of Electrical, Electronic and Computer Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, U.K. (e-mail: m.t.hamood@ncl.ac.uk; s.boussakta@ncl.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2162836

In addition to reducing the number of arithmetic operations, the main benefit of the radix-4 FWFT algorithm is the fact that it computes both transforms (WHT and DFT) simultaneously using a unified butterfly, therefore reducing the number of arithmetic operations, delays and indexing.

The rest of the paper is organized as follows. Section II presents the mathematical development of the proposed algorithm. Section III outlines the arithmetic complexity, computer run-time, and comparisons. Section IV presents the conclusion.

### II. Algorithm Derivation

The radix-4 FWFT algorithm begins by formulating the $\mathbf{T}_N$ matrix which is the WHT matrix multiplied by the DFT matrix, as follows:

$$\mathbf{T}_N = \frac{1}{N}\mathbf{H}_N\hat{\mathbf{F}}_N \tag{1}$$

where $\hat{\mathbf{F}}_N$ is used to indicate the DFT matrix in digit (radix-4) row reverse order and $\mathbf{H}_N$ is the Walsh–Hadamard matrix. The DFT matrix in (1) is of order $N$ and can be written in terms of matrices of lower order $N/4$ [15] as shown in (2) at the bottom of the next page, where $\mathbf{D}^n_{N/4}$ for $n = 1, 2, 3$ is a diagonal matrix whose elements have values of $W^{ni} = \exp(-j\frac{2\pi ni}{N})$ for $i = 0, 1, \ldots, \frac{N}{4} - 1$.

Equation (2) can be factorized as

$$\begin{aligned}\hat{\mathbf{F}}_N = &\begin{bmatrix} \hat{\mathbf{F}}_{N/4} & 0 & 0 & 0 \\ 0 & \hat{\mathbf{F}}_{N/4} & 0 & 0 \\ 0 & 0 & \hat{\mathbf{F}}_{N/4} & 0 \\ 0 & 0 & 0 & \hat{\mathbf{F}}_{N/4} \end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{I}_{N/4} & 0 & 0 & 0 \\ 0 & \mathbf{D}_{N/4} & 0 & 0 \\ 0 & 0 & \mathbf{D}^2_{N/4} & 0 \\ 0 & 0 & 0 & \mathbf{D}^3_{N/4} \end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{I}_{N/4} & \mathbf{I}_{N/4} & \mathbf{I}_{N/4} & \mathbf{I}_{N/4} \\ \mathbf{I}_{N/4} & -j\mathbf{I}_{N/4} & \mathbf{I}_{N/4} & j\mathbf{I}_{N/4} \\ \mathbf{I}_{N/4} & \mathbf{I}_{N/4} & -\mathbf{I}_{N/4} & -\mathbf{I}_{N/4} \\ \mathbf{I}_{N/4} & j\mathbf{I}_{N/4} & -\mathbf{I}_{N/4} & -j\mathbf{I}_{N/4} \end{bmatrix}.\end{aligned} \tag{3}$$

The factorization in (3) can be expressed in terms of tensor (Kronecker) product as

$$\begin{aligned}\hat{\mathbf{F}}_N &= (\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{N/4})\Delta_N(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{N/4}) \\ &= (\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}})\Delta_{4^m}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-1}})\end{aligned} \tag{4}$$

where $N = 4^m$ and $\hat{\mathbf{F}}_4$ is given by

$$\hat{\mathbf{F}}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \tag{5}$$

and

$$\begin{aligned}\Delta_N &= \left(\mathbf{I}_{N/4} \oplus \mathbf{D}_{N/4} \oplus \mathbf{D}^2_{N/4} \oplus \mathbf{D}^3_{N/4}\right) \\ &= \left(\mathbf{I}_{4^{m-1}} \oplus \mathbf{D}_{4^{m-1}} \oplus \mathbf{D}^2_{4^{m-1}} \oplus \mathbf{D}^3_{4^{m-1}}\right).\end{aligned} \tag{6}$$

$\mathbf{I}_{N/4}$ is an identity matrix of order-$N/4$, operators $\otimes$ and $\oplus$ stand for tensor product and direct sum, respectively.

1053-587X/$26.00 © 2011 IEEE

Following the same procedure, we can also factorize $\mathbf{H}_N$ as

$$\mathbf{H}_N = \begin{bmatrix} \mathbf{H}_{N/4} & \mathbf{H}_{N/4} & \mathbf{H}_{N/4} & \mathbf{H}_{N/4} \\ \mathbf{H}_{N/4} & -\mathbf{H}_{N/4} & \mathbf{H}_{N/4} & -\mathbf{H}_{N/4} \\ \mathbf{H}_{N/4} & \mathbf{H}_{N/4} & -\mathbf{H}_{N/4} & -\mathbf{H}_{N/4} \\ \mathbf{H}_{N/4} & -\mathbf{H}_{N/4} & -\mathbf{H}_{N/4} & \mathbf{H}_{N/4} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{N/4} & 0 & 0 & 0 \\ 0 & \mathbf{H}_{N/4} & 0 & 0 \\ 0 & 0 & \mathbf{H}_{N/4} & 0 \\ 0 & 0 & 0 & \mathbf{H}_{N/4} \end{bmatrix} \times \begin{bmatrix} \mathbf{I}_{N/4} & \mathbf{I}_{N/4} & \mathbf{I}_{N/4} & \mathbf{I}_{N/4} \\ \mathbf{I}_{N/4} & -\mathbf{I}_{N/4} & \mathbf{I}_{N/4} & -\mathbf{I}_{N/4} \\ \mathbf{I}_{N/4} & \mathbf{I}_{N/4} & -\mathbf{I}_{N/4} & -\mathbf{I}_{N/4} \\ \mathbf{I}_{N/4} & -\mathbf{I}_{N/4} & -\mathbf{I}_{N/4} & \mathbf{I}_{N/4} \end{bmatrix}. \quad (7)$$

Likewise, (7) can be also expressed in terms of tensor product as

$$\begin{aligned} \mathbf{H}_N &= (\mathbf{I}_4 \otimes \mathbf{H}_{N/4})(\mathbf{H}_4 \otimes \mathbf{I}_{N/4}) \\ &= (\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-1}})(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}}) \end{aligned} \quad (8)$$

where

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \quad (9)$$

Substituting (4) and (8) into (1), the general radix-4 $\mathbf{T}$ matrix of order $N = 4^m$ can be written as

$$\begin{aligned} \mathbf{T}_N = \frac{1}{N}&(\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-1}})(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}}) \\ &\times (\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}})\Delta_{4^m}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-1}}). \end{aligned} \quad (10)$$

The terms of the product $(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}})$ in (10) can be swapped. This can be proved, using the mixed-product property $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC} \otimes \mathbf{BD})$ [16] where $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$ are matrices of appropriate sizes, as follows:

$$\begin{aligned} &(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}}) \\ &\quad = (\mathbf{H}_4\mathbf{I}_4 \otimes \mathbf{I}_{4^{m-1}}\hat{\mathbf{F}}_{4^{m-1}}) = (\mathbf{H}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}}) \\ &(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}})(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}}) \\ &\quad = (\mathbf{I}_4\mathbf{H}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}}\mathbf{I}_{4^{m-1}}) = (\mathbf{H}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}}). \end{aligned} \quad (11)$$

Therefore, the product $(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}})$ in (10) is equal to $(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}})(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})$, hence (10) can be written as

$$\begin{aligned} \mathbf{T}_N &= \frac{1}{N}(\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-1}}) \\ &\quad \times (\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})\Delta_{4^m}(\hat{\mathbf{F}}_4 \otimes \hat{\mathbf{I}}_{4^{m-1}}) \\ &= \frac{1}{N}(\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-1}}\hat{\mathbf{F}}_{4^{m-1}}) \times \mathbf{T}_N^{\mathrm{I}} \end{aligned} \quad (12)$$

where $\mathbf{T}_N^{\mathbf{I}} = (\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})\Delta_{4^m}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-1}})$.

The product $(\mathbf{H}_{4^{m-1}}\hat{\mathbf{F}}_{4^{m-1}})$ in (12) can be factorized further, by substituting $\mathbf{H}_{4^{m-1}}$ and $\hat{\mathbf{F}}_{4^{m-1}}$ into (8) and (4) respectively, we get

$$\begin{aligned} \mathbf{H}_{4^{m-1}} &= (\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-2}})(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}}) \\ \hat{\mathbf{F}}_{4^{m-1}} &= (\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-2}})\Delta_{4^{m-1}}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-2}}). \end{aligned} \quad (13)$$

Substituting $\mathbf{H}_{4^{m-1}}$ and $\hat{\mathbf{F}}_{4^{m-1}}$ by their values in (13) into (12), we obtain

$$\begin{aligned} \mathbf{T}_N = \frac{1}{N}&[\mathbf{I}_4 \otimes ((\mathbf{I}_4 \otimes \mathbf{H}_{4\mathrm{m}-2})(\mathbf{H}_4 \otimes \mathbf{I}_{4\mathrm{m}-2}) \\ &\times (\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4\mathrm{m}-2})\Delta_{4\mathrm{m}-1}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4\mathrm{m}-2}))] \times \mathbf{T}_N^{\mathrm{I}}. \end{aligned} \quad (14)$$

Using the same procedure as in (11), the product $(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-2}})$ in (14) can also be written as $(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-2}})(\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}})$, so (14) can be expressed as

$$\begin{aligned} \mathbf{T}_N &= \frac{1}{N}[\mathbf{I}_4 \otimes ((\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-2}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_{4^{m-2}}) \\ &\quad \times (\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}})\Delta_{4^{m-1}}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-2}}))] \times \mathbf{T}_N^{\mathrm{I}} \\ &= \frac{1}{N}(\mathbf{I}_4 \otimes (\mathbf{I}_4 \otimes \mathbf{H}_{4^{m-2}}\hat{\mathbf{F}}_{4^{m-2}}))[(\mathbf{I}_4 \otimes \mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}}) \\ &\quad \times (\mathbf{I}_4 \otimes \Delta_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-2}})] \times \mathbf{T}_N^{\mathrm{I}} \\ &= \frac{1}{N}(\mathbf{I}_{16} \otimes \mathbf{H}_{4^{m-2}}\hat{\mathbf{F}}_{4^{m-2}}) \times \mathbf{T}_N^{\mathrm{II}} \times \mathbf{T}_N^{\mathrm{I}} \end{aligned} \quad (15)$$

where $\mathbf{T}_N^{\mathrm{II}} = (\mathbf{I}_4 \otimes \mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}})(\mathbf{I}_4 \otimes \Delta_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-2}})$ and $\mathbf{T}_N^{\mathrm{I}}$ is as defined in (12).

This factorization will continue and after $\log_4 N$ stages, we get the last term as

$$\begin{aligned} &(\mathbf{I}_{4^{m-2}} \otimes \mathbf{H}_{16}\hat{\mathbf{F}}_{16}) \\ &\quad = [\mathbf{I}_{4^{m-2}} \otimes ((\mathbf{I}_4 \otimes \mathbf{H}_4)(\mathbf{H}_4 \otimes \mathbf{I}_4) \\ &\qquad \times (\mathbf{I}_4 \otimes \hat{\mathbf{F}}_4)\Delta_{16}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_4))] \\ &\quad = [\mathbf{I}_{4^{m-2}} \otimes ((\mathbf{I}_4 \otimes \mathbf{H}_4)(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_4) \\ &\qquad \times (\mathbf{H}_4 \otimes \mathbf{I}_4)\Delta_{16}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_4))] \\ &\quad = (\mathbf{I}_{4^{m-1}} \otimes \mathbf{H}_4)(\mathbf{I}_{4^{m-1}} \otimes \hat{\mathbf{F}}_4)(\mathbf{I}_{4^{m-2}} \otimes \mathbf{H}_4 \otimes \mathbf{I}_4) \\ &\qquad \times (\mathbf{I}_{4^{m-2}} \otimes \Delta_{16})(\mathbf{I}_{4^{m-2}} \otimes \hat{\mathbf{F}}_4 \otimes \mathbf{I}_4). \end{aligned} \quad (16)$$

Combining (12)–(16) and using the fact $\Delta_{\mathbf{4}} = \mathbf{I}_{\mathbf{4}}$, the $\mathbf{T}$ matrix can be decomposed as

$$\begin{aligned} \mathbf{T}_N = \frac{1}{N}&[(\mathbf{I}_{4^{m-1}} \otimes \mathbf{H}_4)(\mathbf{I}_{4^{m-1}} \otimes \Delta_4)(\mathbf{I}_{4^{m-1}} \otimes \hat{\mathbf{F}}_4) \\ &\times (\mathbf{I}_{4^{m-2}} \otimes \mathbf{H}_4 \otimes \mathbf{I}_4)(\mathbf{I}_{4^{m-2}} \otimes \Delta_{16}) \\ &\times (\mathbf{I}_{4^{m-2}} \otimes \hat{\mathbf{F}}_4 \otimes \mathbf{I}_4) \cdots (\mathbf{I}_4 \otimes \mathbf{H}_4 \otimes \mathbf{I}_{4^{m-2}}) \\ &\times (\mathbf{I}_4 \otimes \Delta_{4^{m-1}})(\mathbf{I}_4 \otimes \hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-2}}) \\ &\times (\mathbf{H}_4 \otimes \mathbf{I}_{4^{m-1}})\Delta_{4^m}(\hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^{m-1}})]. \end{aligned} \quad (17)$$

Since $N$ is a power of four, then (17) can be written in compact form as

$$\begin{aligned} \mathbf{T}_N = \prod_{i=0}^{m-1} &(\mathbf{I}_{4^{m-i-1}} \otimes \mathbf{H}_4 \otimes \mathbf{I}_{4^i})\left(\mathbf{I}_{4^{m-i-1}} \otimes \frac{1}{4}\Delta_{4^{i+1}}\right) \\ &\times (\mathbf{I}_{4^{m-i-1}} \otimes \hat{\mathbf{F}}_4 \otimes \mathbf{I}_{4^i}). \end{aligned} \quad (18)$$

---

$$\hat{\mathbf{F}}_N = \begin{bmatrix} \hat{\mathbf{F}}_{N/4} & \hat{\mathbf{F}}_{N/4} & \hat{\mathbf{F}}_{N/4} & \hat{\mathbf{F}}_{N/4} \\ \hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4} & -j\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4} & -\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4} & j\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4} \\ \hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^2 & -\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^2 & \hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^2 & -\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^2 \\ \hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^3 & j\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^3 & -\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^3 & -j\hat{\mathbf{F}}_{N/4}\mathbf{D}_{N/4}^3 \end{bmatrix} \quad (2)$$

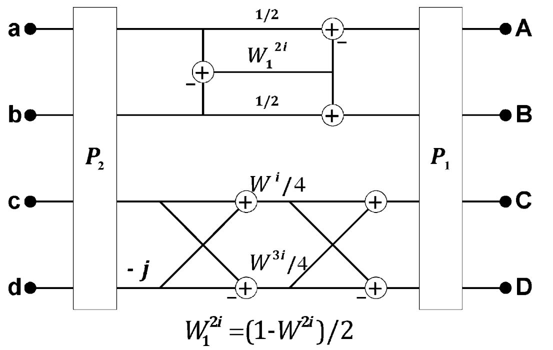


Fig. 1. A single butterfly of the radix-4 FWFT algorithm.

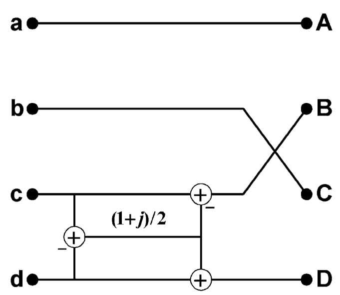


Fig. 2. Trivial radix-4 FWFT butterfly when $i=0$.

Transforms with the matrices having the form described in (18) can be computed with the proposed in-place butterfly structure, shown in Fig. 1. The radix-4 FWFT algorithm is based on the decomposition of the transform matrix into product of sparse matrices with every row containing only four nonzero entries. Additionally, a great saving in arithmetic operations can be achieved when $(i=0)$ as shown in Fig. 2.

As an example, using an input sequence of length of $N=16$, we explain this algorithm clearly. There are two stages to complete this computation. First, according to (18), the $\mathbf{T}$ matrix can be factorized into two sparse matrices, as follows:

$$\begin{aligned}\mathbf{T}_{16} &= \prod_{i=0}^{1}(\mathbf{I}_{4^{1-i}}\otimes\mathbf{H}_4\otimes\mathbf{I}_{4^i})\left(\mathbf{I}_{4^{1-i}}\otimes\frac{1}{4}\Delta_{4^{i+1}}\right)(\mathbf{I}_{4^{1-i}}\otimes\hat{\mathbf{F}}_4\otimes\mathbf{I}_{4^i})\\ &= \mathbf{T}_{16}^{\mathrm{II}}\times\mathbf{T}_{16}^{\mathrm{I}}\end{aligned}\tag{19}$$

The matrices $\mathbf{T}_{16}^{\mathrm{I}}$ and $\mathbf{T}_{16}^{\mathrm{II}}$ in (19) represents the stages of radix-4 computation, the first stage is given by $\mathbf{T}_{16}^{\mathrm{I}}$ as

$$\mathbf{T}_{16}^{\mathrm{I}}=\frac{1}{4}(\mathbf{I}_4\otimes\mathbf{H}_4)(\mathbf{I}_4\otimes\hat{\mathbf{F}}_4)=\begin{bmatrix}\mathbf{I}_4 & 0 & 0 & 0\\ 0 & 0 & \mathbf{I}_4 & 0\\ 0 & \frac{(1-j)}{2}\mathbf{I}_4 & 0 & \frac{(1+j)}{2}\mathbf{I}_4\\ 0 & \frac{(1+j)}{2}\mathbf{I}_4 & 0 & \frac{(1-j)}{2}\mathbf{I}_4\end{bmatrix}\tag{20}$$

and the second stage is given by $\mathbf{T}_{16}^{\mathrm{II}}$ as

$$\begin{aligned}\mathbf{T}_{16}^{\mathrm{II}} &= \frac{1}{4}(\mathbf{H}_4\otimes\mathbf{I}_4)\Delta_{16}(\hat{\mathbf{F}}_4\otimes\mathbf{I}_4)\\ &= \frac{1}{4}\begin{bmatrix}\mathbf{H}_4\hat{\mathbf{F}}_4 & 0 & 0 & 0\\ 0 & \mathbf{H}_4\mathbf{D}_4\hat{\mathbf{F}}_4 & 0 & 0\\ 0 & 0 & \mathbf{H}_4\mathbf{D}_4^2\hat{\mathbf{F}}_4 & 0\\ 0 & 0 & 0 & \mathbf{H}_4\mathbf{D}_4^3\hat{\mathbf{F}}_4\end{bmatrix}.\end{aligned}\tag{21}$$

Second, the submatrices of $\mathbf{T}_{16}^{\mathrm{II}}$ can be further factorized into a sparse diagonal structure as shown in (22)-(23) at the bottom of the page, where in (23), $\mathbf{P_1}$ and $\mathbf{P_2}$ are two matrices of size $4\times 4$ given by

$$\mathbf{P}_1=\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & -1\end{bmatrix}\tag{24}$$

and

$$\mathbf{P}_2=\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{bmatrix}.\tag{25}$$

Following the same development for $\mathbf{H_4D_4\hat{F}_4}$, other matrices can be obtained as shown in (26)-(27) at the bottom of the next page.

Thus, the $\mathbf{T}$ matrix can be factorized into a product of sparse matrices where the rows of each matrix have a maximum of two nonzero elements, reducing both multiplications and additions. Transforms with the matrices having the structure described in (22)–(27) can be computed with the proposed radix-4 algorithm in butterfly style structure shown in Figs. 1 and 2. The signal flow graph of the radix-4 FWFT transform for 16-point input is illustrated in Fig. 3.

## III. Arithmetic Complexity

This section discusses the arithmetic complexity of the proposed algorithm and compares it with existing WHT-FFT based on similar algorithms. First, the number of complex multiplications and additions of the radix-4 FWFT computation can be calculated and the number of real arithmetic operations obtained. Each butterfly calculates four points involving three multiplications and 16 additions. Furthermore, one more addition can also be reduced using the butterfly structure shown in Fig. 1. Therefore, three multiplications and 15 additions are actually utilized in each butterfly. It can be seen from Fig. 3 that the first stage is computed in four butterflies, where each butterfly calculates four points representing the matrix given in (20). This requires three additions only and no multiplications. The second stage is also processed in four butterflies, where each of them represents one of the submatrices given in (21). In total, there are nine complex multiplications and 60 complex additions (note that the twiddle factor values of this algorithm have the digit reverse order range of $i=0,1,\ldots N/4-1$). Therefore,

$$\frac{1}{4}\mathbf{H}_4\hat{\mathbf{F}}_4=\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & \frac{(1-j)}{2} & 0 & \frac{(1+j)}{2}\\ 0 & \frac{(1+j)}{2} & 0 & \frac{(1-j)}{2}\end{bmatrix}\tag{22}$$

and

$$\frac{1}{4}\mathbf{H}_4\mathbf{D}_4\hat{\mathbf{F}}_4=\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ w & w^5 & -w & -w^5\\ w^2 & -w^2 & w^2 & -w^2\\ w^3 & -w^7 & -w^3 & w^7\end{bmatrix}=\frac{1}{4}\begin{bmatrix}(1+w^2) & (1-w^2) & 0 & 0\\ (1-w^2) & (1+w^2) & 0 & 0\\ 0 & 0 & w(1+w^2) & w^5(1-w^2)\\ 0 & 0 & w(1-w^2) & w^5(1+w^2)\end{bmatrix}\mathbf{P}_2\tag{23}$$

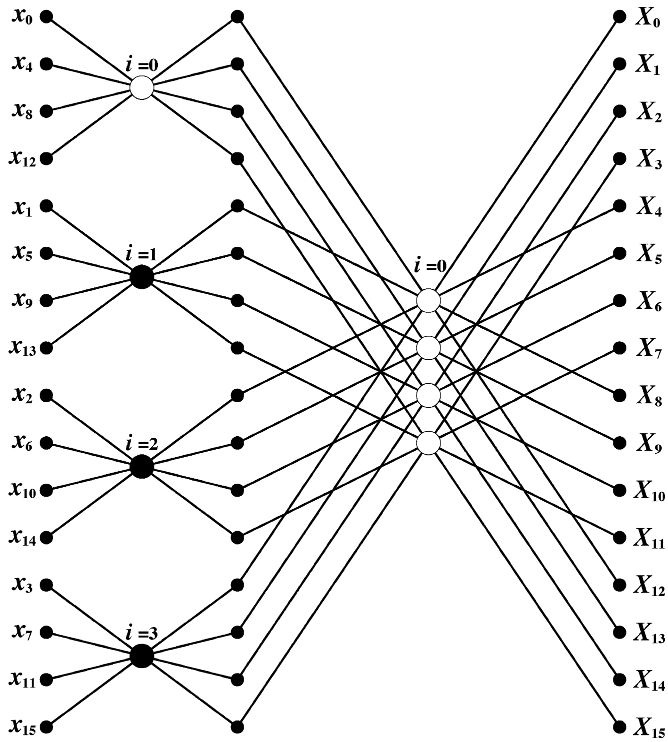


Fig. 3. Signal flow graph of the proposed radix-4 FWFT algorithm when $N=16$; where black and white circles stand for the butterflies shown in Figs. 1 and 2, respectively.

the total number of multiplications and additions for the general case of radix-4 FWFT is calculated using $\frac{N}{4}\log_4 N$ butterflies. This includes $\frac{N-1}{3}$ trivial butterflies with only three complex additions. The rest of the butterflies involve three complex multiplications and fifteen complex additions each. Hence, the number of complex multiplication $M_{\mathrm{C}}$ and complex additions $A_{\mathrm{C}}$ can be calculated as

$$\begin{aligned} M_{\mathrm{C}} &= 3\times\left[\frac{N}{4}\log_4 N - \frac{N-1}{3}\right] \\ &= \frac{3N}{4}\log_4 N - (N-1) \end{aligned} \tag{28}$$

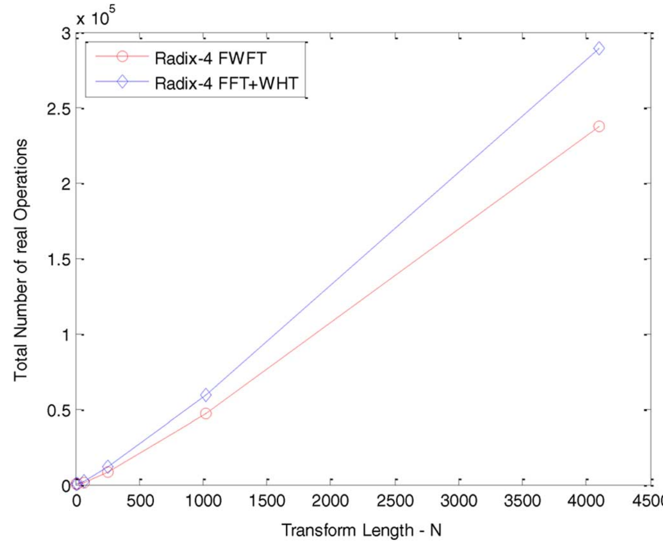


Fig. 4. Total number of real operations (multiplications and additions) for the proposed radix-4 FWFT and WHT + FFT algorithms.

$$\begin{aligned} A_{\mathrm{C}} &= 15\times\left[\frac{N}{4}\log_4 N - \frac{N-1}{3}\right] \\ &\quad + 3\times\left[\frac{N-1}{3}\right] = \frac{15N}{4}\log_4 N - 4(N-1). \end{aligned} \tag{29}$$

It should be noted that (28) and (29) are for general butterfly implementation, thus $w^{N/4}$ is considered as a twiddle factor that is counted as complex multiplication rather than multiplying by $-j$.

The corresponding number of complex multiplications $C_{\mathrm{M}}$ and complex additions $C_{\mathrm{A}}$ for the conventional radix-4 WHT followed by radix-4 FFT based on similar implementation is given as

$$C_{\mathrm{M}} = \frac{3N}{4}(\log_4 N - 1) \tag{30}$$

$$C_A = 4N\log_4 N. \tag{31}$$

Comparing (28) with (30) and (29) with (31) reveals that the proposed algorithm involves $\frac{N}{4}-1$ less complex multiplications and $\frac{N}{4}\log_4 N + 4(N-1)$ less complex additions.

$$\begin{aligned} \frac{1}{4}\mathbf{H}_4\mathbf{D}_4^2\hat{\mathbf{F}}_4 &= \frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}1&1&1&1\\w^2&w^6&-w^2&-w^6\\w^4&-w^4&w^4&-w^4\\w^6&-w^2&-w^6&w^2\end{bmatrix} \\ &= \frac{1}{4}\mathbf{P}_1\begin{bmatrix}(1+w^4)&(1-w^4)&0&0\\(1-w^4)&(1+w^4)&0&0\\0&0&w^2(1+w^4)&w^6(1-w^4)\\0&0&w^2(1-w^4)&w^6(1+w^4)\end{bmatrix}\mathbf{P}_2 \end{aligned} \tag{26}$$

and

$$\begin{aligned} \frac{1}{4}\mathbf{H}_4\mathbf{D}_4^3\hat{\mathbf{F}}_4 &= \frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}1&1&1&1\\w^3&w^7&-w^3&-w^7\\w^6&-w^6&w^6&-w^6\\w&w^5&w&w^5\end{bmatrix} \\ &= \frac{1}{4}\mathbf{P}_1\begin{bmatrix}(1+w^6)&(1-w^6)&0&0\\(1-w^6)&(1+w^6)&0&0\\0&0&w^3(1+w^6)&w^7(1-w^6)\\0&0&w^3(1-w^6)&w^7(1+w^6)\end{bmatrix}\mathbf{P}_2 \end{aligned} \tag{27}$$

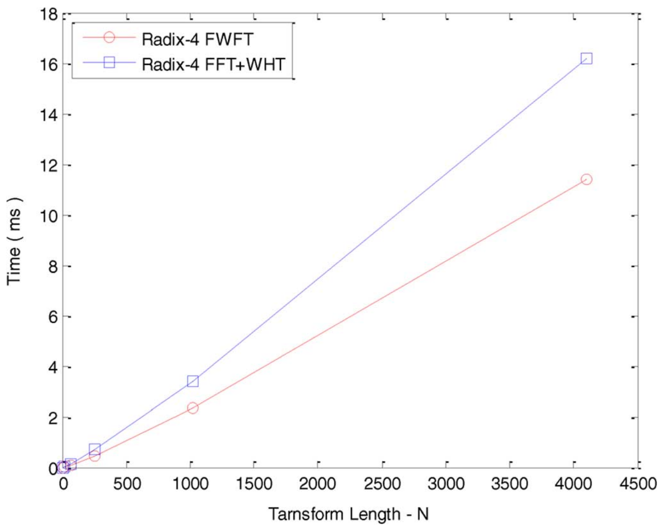


Fig. 5. The run-time for the proposed radix-4 FWFT and WHT + FFT algorithms using Matlab run on Core Duo computer with speed of 2.4 GHz.

TABLE I
A COMPARISON OF REAL ARITHMETIC OPERATIONS FOR WHT-DFT ALGORITHMS USING 4/2 SCHEME

| Length *N* | Proposed radix-4 FWFT Algorithm | | Radix-4 FFT+WHT Algorithm | |
|---|---|---|---|---|
| | *Mults.* | *Adds.* | *Mults.* | *Adds.* |
| 4 | 0 | 6 | 0 | 32 |
| 16 | 36 | 138 | 48 | 280 |
| 64 | 324 | 1098 | 384 | 1728 |
| 256 | 2052 | 6666 | 2304 | 9344 |
| 1024 | 11268 | 35850 | 12288 | 47104 |
| 4096 | 57348 | 180234 | 61440 | 227328 |

Considering four real multiplications and two real additions for each complex multiplication, and two real additions for each complex addition, the developed radix-4 FWFT and the conventional radix-4 WHT-FFT algorithms are compared based on the total number of real arithmetic operations for different transform lengths as shown in Table I, and as depicted in Fig. 4. Furthermore, in order to test the validity of the developed algorithm and confirm the results based on the arithmetic operations, the developed algorithm has been implemented using Matlab, and tested on a Core Duo computer with speed of 2.4 GHz and 2 GB RAM. Fig. 5 shows the comparison of the computer run times for the newly developed algorithm and conventional radix-4 WHT-FFT algorithm. Clearly, the results show that the proposed algorithm is about 70%–36% faster than the conventional algorithm.

## IV. CONCLUSION

This work has presented an efficient algorithm for fast computation of the WHT and DFT combined in one single step. This can be used to reduce the computational and implementation costs of the WHT-OFDM systems. The newly developed radix-4 FWFT algorithm has been implemented using butterfly structure, and its arithmetic complexity has been calculated and analyzed. A comparison with the conventional WHT-FFT algorithm has been made based on the number of arithmetic operations and computer run times. The results have shown that the proposed algorithm has reduced the number of arithmetic operations and computer run time significantly.

## REFERENCES

[1] W. Shaopeng, Z. Shihua, and Z. Guomei, "A Walsh–Hadamard coded spectral efficient full frequency diversity OFDM system," *IEEE Trans. Commun.*, vol. 58, no. 1, pp. 28–34, 2010.
[2] Z. Dlugaszewski and K. Wesolowski, "WHT/OFDM—An improved OFDM transmission method for selective fading channels," in *Proc. Symp. Commun. Veh. Technol. (SCVT)*, 2000, pp. 144–149.
[3] S. Boussakta and A. G. J. Holt, "Fast algorithm for calculation of both Walsh–Hadamard and Fourier transforms (FWFTs)," *Electron. Lett.*, vol. 25, pp. 1352–1354, 1989.
[4] Y. Wu, C. K. Ho, and S. Sun, "On some properties of Walsh-Hadamard transformed OFDM," in *Proc. Veh. Technol. Conf. (VTC)*, 2002, pp. 2096–2100.
[5] M. S. Ahmed, S. Boussakta, B. Sharif, and C. C. Tsimenidis, "OFDM based new transform with BER performance improvement across multipath transmission," in *Proc. Int. Conf. Commun. (ICC)*, 2010, pp. 1–5.
[6] P. Myonghee, J. Heeyoung, C. Jaehee, C. Namshin, H. Daesik, and K. Changeun, "PAPR reduction in OFDM transmission using Hadamard transform," in *Proc. Int. Conf. Commun. (ICC)*, 2000, vol. 1, pp. 430–433.
[7] R. D. J. van Nee, "OFDM codes for peak-to-average power reduction and error correction," in *Proc. IEEE Global Telecommun. Conf. (IEEE GLOBECOM)*, 1996, vol. 1, pp. 740–744.
[8] A. Zolghadrasli and M. H. Ghamat, "PAPR reduction in OFDM system by using Hadamard transform in BSLM techniques," in *Proc. 9th Int. Symp. Signal Process. Its Appl. (ISSPA)*, 2007, pp. 1–4.
[9] B. Gaffney and A. D. Fagan, "Walsh–Hadamard transform precoded MB-OFDM: An improved high data rate ultrawideband system," in *Proc. 17th Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, 2006, pp. 1–5.
[10] P. Marti-Puig and J. Sala-Alvarez, "A fast OFDM-CDMA user demultiplexing architecture," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2000, vol. 6, pp. 3358–3361.
[11] A. C. McCormick, P. M. Grant, and J. S. Thompson, "A comparison of convolutional and Walsh coding in OFDM wireless LAN systems," in *Proc. 11th Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, 2000, vol. 1, pp. 166–169.
[12] M. Munster and L. Hanzo, "Performance of SDMA multiuser detection techniques for Walsh–Hadamard-spread OFDM schemes," in *Proc. Veh. Technol. Conf. (VTC)—Fall*, 2001, vol. 4, pp. 2319–2323.
[13] N. Yee, J. Linnartz, and G. Fettweis, "Multi-carrier CDMA in indoor wireless radio networks," in *Proc. 12th Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, 1993, pp. 109–113.
[14] Y. Young-Hwan, J. Won-Gi, P. Jong-Ho, and S. Hyoung-Kyu, "A simple construction of OFDM-CDMA signals with low peak-to-average power ratio," *IEEE Trans. Broadcast.*, vol. 49, no. 4, pp. 403–407, 2003.
[15] M. Bellanger, *Digital Processing of Signals: Theory and Practice*. New York: Wiley, 2000.
[16] W. H. Steeb and T. K. Shi, *Matrix Calculus and Kronecker Product With Applications and C++ Programs*. Singapore: World Scientific, 1997.