International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 4, August 2016, pp. 1654~1661
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i4.10469

New Decimation-in-Time Fast Hartley Transform Algorithm

Mounir T. Hamood
Electrical Engineering Departement, University of Tikrit, Iraq

Article Info

Article history:

Received Mar 12, 2016
Revised Jun 15, 2016
Accepted Jun 28, 2016

Keyword:

Decimation-in-time algorithm
Fast transform algorithms
discrete hartley transform
Radix-2^2 algorithms

ABSTRACT

This paper presents a new algorithm for fast calculation of the discrete Hartley transform (DHT) based on decimation-in-time (DIT) approach. The proposed radix-2^2 fast Hartley transform (FHT) DIT algorithm has a simple butterfly structure that offers flexibility for various powers-of-two transform lengths, significantly reducing the computational complexity with regular bit reversing orderfor the output sequence. The algorithm is derived through the three-dimensional index mapping approach and by incorporating two stages of the signal flow graph into an integrated butterfly. The algorithm is implemented and its arithmetic complexity has been analysed and compared with the current FHT algorithms, revealing that it is substantially minimize the structural complexity with better indexing arrangement that is suitable for efficient implementation.

Corresponding Author:

Mounir Taha Hamood,
Electrical Engineering Departement,
College of Engineering, University of Tikrit,
Salah Al-Deen Governorate, Tikrit, Iraq.
Email:m.t.hamood@tu.edu.iq

1. INTRODUCTION

The discrete Hartley transform (DHT) first introduced by Bracewell [1],[2], has become increasingly popular as an alternative to the discrete Fourier transform DFT for applications involving real data [3]-[7]. The DHT is mainly attractive for the case of real value sequence, its validity being made possible by the fact that all the properties related to the DFT like the shifting and circular convolution theorems, are also applicable to the DHT [8]. In general, given the DHT coefficients of a signal, it is easy to compute the corresponding DFT coefficients, and vice versa [9]. However, the DHT has two advantages over the DFT. First, it is a real-to-real transform, so no complex arithmetic is required. Second, the DHT transform matrix is symmetrical, and therefore the forward and inverse transforms are identical. The second point is quite significant because it means that in applications which need both the forward and inverse transforms such as fast convolution algorithms, only one routine for computing the DHT is necessary, since it will also perform the inverse DHT. This feature is of high importance and in particular for the multidimensional applications where the amounts of data are very large [10].

The amount of arithmetical operations for the computation of the DHT is severe and needs N^2 multiplications and additions, therefore the fast Hartley transform (FHT) algorithms have been presented to calculate the transform at high speed to meet the requirements of real time applications. The first FHT algorithm introduced by Bracewell [11] implements the DHT in complexity proportionate to $N\log_2 N$ using radix-2 decimation in-time (DIT) algorithm. Further fast algorithms for computing the DHT with powers of two transform lengths are developed by Meckeburg and Lipka [12], Kwong and Shiu [13]and Hou [14]. Moreover, Sorenson *et al* [15] presented a complete set of FHT algorithms using the indexing map method [16], realizing the algorithms decimation-in-time (DIT) and decimation-in-frequency (DIF) approaches and confirmed that all the well-known FFT algorithms can also be used to the calculation of the FHT. Malvar [17] proposed an FHT algorithm based on discrete cosine transfrom (DCT) that offer an improved

computational complexity. Chan and Siu [18] introduced a different approach by means of a discrete symmetric cosine transform (DSCT). Duhamel *et al* [19] introduced an algorithm that divides a length-N DHT calculation into length-$N/2$ DHT and two length-$N/4$ DFTs, which uses the lowest known number of additions without increasing the number of multiplications. All these algorithms can be classified as the split-radix FHTs that appear to require the minimum arithmetic complexity. However, like FFTs they have a complicated butterfly structure, making them difficult to implement [20].

Recently, new classes of FFT algorithms known as radix-2^2 algorithms [21], and their variant algorithms [22],[23], have been introduced as an alternative to split-radix algorithms and to resolve the length limitation of the higher radix algorithms for hardware implementation of FFTs. The idea for radix-2^2 algorithms is to realize both the regular butterfly structure provided by the radix-2 algorithm and the condensed number of twiddle factor multiplications offered by the higher radix algorithms.

While the DIF approach for the radix-2^2 FHT algorithm has been recently developed [24]; however, for any transform to stand as a good candidate for real time applications, its entire fast algorithms need to be developed, such as the DIT approach. Therefore, it is the purpose of this paper to develop such an algorithm for fast calculation of the DHT.

The content of the paper is organisedas follows. Section 2 reviews the definitions and basic properties of the DHT, and then the multidimensional index mapping technique is briefly reviewed in section 2.1. The complete developments of the proposed algorithm are presented in section 3, and then the algorithm's performance is analysed and compared with existing FHT algorithms in section 4. Finally, a conclusion in given in section 5.

2. REVIEW OF DHT TRANSFORM

The DHT transform pair for a real value sequence $x(n)$ of length N is defined as [1]:

$$X(k) = \sum_{n=0}^{N-1} x(n)\,\mathrm{cas}(\theta nk)$$
$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,\mathrm{cas}(\theta nk) \quad (1)$$

where $\mathrm{cas}(\theta) = \cos(\theta) + \sin(\theta)$, $\theta = 2\pi/N$ and $N = 2^m$ is the transform length.

The DHT is closely related to the DFT and the relationship between them can be obtained using the identity $[\mathrm{cas}(\theta nk) = \mathrm{Re}(W_N^{nk}) - \mathrm{Im}(W_N^{nk})]$, where W_N^{nk} is the DFT kernel defined as $W_N = \exp(-j2\pi/N)$. Therefore, the DHT can be obtained by subtracting the real part from the imaginary part of the DFT of a real value sequence as:

$$\mathrm{DHT}[x(n)] = \mathrm{Re}\{\mathrm{DFT}[x(n)]\} - \mathrm{Im}\{\mathrm{DFT}[x(n)]\} \quad (2)$$

On the other hand, the real and imaginary parts of the DFT of the real value sequence can be obtained from the DHT using the identities $\left[\mathrm{Re}(W_N^{nk}) = \frac{1}{2}\{\mathrm{cas}(-\theta nk) + \mathrm{cas}(\theta nk)\}\right]$ and $\left[\mathrm{Im}(W_N^{nk}) = \frac{1}{2}\{\mathrm{cas}(-\theta nk) - \mathrm{cas}(\theta nk)\}\right]$, as follows:

$$\mathrm{Re}\{\mathrm{DFT}[x(n)]\} = \frac{1}{2}\{\mathrm{DHT}[x(N-n)] + \mathrm{DHT}[x(n)]\}$$
$$\mathrm{Im}\{\mathrm{DFT}[x(n)]\} = \frac{1}{2}\{\mathrm{DHT}[x(N-n)] - \mathrm{DHT}[x(n)]\} \quad (3)$$

Many properties of the DHT are comparable to the equivalent theorems for the DFT. One of the most convenient properties is the DHT shift theorem [2], given as:

$$\mathrm{DHT}[x(n+n_0)] = X(k)\cos(\theta n_0 k) - X(N-k)\sin(\theta n_0 k) \quad (4)$$

and the DHT convolution theorem of two real-valued sequences $x(n)$ and $h(n)$ is given by [15]:

$$\mathrm{DHT}[x(n)*h(n)]=\frac{1}{2}[X(k)H(k)+X(k)H(N-k)+X(N-k)H(k)-X(N-k)H(N-k)] \quad (5)$$

where $X(k)$ and $H(k)$ are the DHTs of $x(n)$ and $h(n)$ respectively.

2.1. Multidimensional index map

The concepts of the multidimensional linear index mapping technique will be briefly reviewed in this section, because it is an important part corresponds to the presented algorithm. This technique provides an efficient approach to the development of fast algorithm [16], and it can be used for the derivation of the FFT as well as other fast algorithms.

It is based on mapping of a one-dimensional array of length $N=N_1N_2$ into a two-dimensional array of size $N_1 \times N_2$. If there is a common factor between N_1 and N_2, a prime factor map (PFM) will result, whereas if there is a no common factor between them the common factor map (CFM) can be obtained. Since that the proposed algorithm in this paper is based on Cooly-Tukey methods which require that the length N is a power of some number, i.e. (r^m) where r is called the radix of the algorithm, therefore the CFM is of our interest. The two dimensional CFM map for the time index n and frequency index k for the DFT matrix is given by:

$$\begin{aligned} n &= n_1 + N_1 n_2 \qquad n_1 = 0,1,...,N_1-1; \quad n_2 = 0,1,...,N_2-1 \\ k &= k_2 + N_2 k_1 \qquad k_1 = 0,1,...,N_2-1; \quad k_2 = 0,1,...,N_1-1 \end{aligned} \quad (6)$$

choosing $N_1=r$ and $N_2=N/r$ will result the DIT algorithms.

3. ALGORITM DERIVATION

The derivation of radix-2^2 FHT DIT algorithm begins by applying the three-dimensional linear index map by replacing the indices n and k as:

$$\begin{aligned} n &= n_1 + 2n_2 + 4n_3 \qquad \{n_1, n_2 = 0,1;\ n_3 = 0,1,...,\tfrac{N}{4}-1\} \\ k &= k_3 + \tfrac{N}{4}k_2 + \tfrac{N}{2}k_1 \qquad \{k_1, k_2 = 0,1;\ k_3 = 0,1,...,\tfrac{N}{4}-1\} \end{aligned} \quad (7)$$

Therefore the CFM map of (1) becomes:

$$X(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)=\sum_{n_1=0}^{1}\sum_{n_2=0}^{1}\sum_{n_3=0}^{N/4-1}x(4n_3+2n_2+n_1)\,\mathrm{cas}\left(\theta(4n_3+2n_2+n_1)(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)\right) \quad (8)$$

Using the cas identity:

$$\mathrm{cas}(x+y)=\cos(x)\mathrm{cas}(y)+\sin(x)\mathrm{cas}(-y) \quad (9)$$

The cas term in (8) can be expanded as:

$$\begin{aligned} \mathrm{cas}\left(\theta(4n_3+2n_2+n_1)(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)\right) &= \cos\left(\theta(2n_2+n_1)(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)\right)\mathrm{cas}\left(4\theta n_3k_3\right) \\ &\quad -\sin\left(\theta(2n_2+n_1)(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)\right)\mathrm{cas}\left(-4\theta n_3k_3\right) \end{aligned} \quad (10)$$

Substituting (10) into (8), we get:

$$\begin{aligned} X(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)=\sum_{n_1=0}^{1}\sum_{n_2=0}^{1}\Big[& X_{4n_3+2n_2+n_1}(k_3)\cos\left(\theta(2n_2+n_1)(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)\right) \\ & -X_{4n_3+2n_2+n_1}(\tfrac{N}{4}-k_3)\sin\left(\theta(2n_2+n_1)(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)\right)\Big] \end{aligned} \quad (11)$$

where $X_{4n_3+2n_2+n_1}(k_3)$ and $X_{4n_3+2n_2+n_1}(\frac{N}{4}-k_3)$ in (11) are two DHTs of length $\frac{N}{4}$, defined as:

$$X_{4n_3+2n_2+n_1}(k_3)=\sum_{n_3=0}^{N/4-1}x(4n_3+2n_2+n_1)\text{cas}\left(4\theta n_3k_3\right)$$
$$X_{4n_3+2n_2+n_1}(\tfrac{N}{4}-k_3)=\sum_{n_3=0}^{N/4-1}x(4n_3+2n_2+n_1)\text{cas}\left(-4\theta n_3k_3\right) \quad (12)$$

If (11) is to be computed, then an ordinary radix-4 DIT FHT [9] results with the output arranged in digit-reverse order. However as given in [25], if we consider the first two steges of decomposition together, then the output will be arranged in bit- reverse order rather than in digit-reverse order by swapping the places of the intermediate twiddle-factors [$\cos(.)$ and $\sin(.)$], thus (11) can be modified to:

$$\begin{aligned}X(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)&=\sum_{n_2=0}^{1}\sum_{n_1=0}^{1}\Big[X_{4n_3+2n_1+n_2}(k_3)\cos\left(\theta(\tfrac{N}{2}n_1k_1+\tfrac{N}{4}n_1k_2+\tfrac{N}{2}n_2k_2+(2n_2+n_1)k_3)\right)\\&\quad-X_{4n_3+2n_1+n_2}(-k_3)\sin\left(\theta(\tfrac{N}{2}n_1k_1+\tfrac{N}{4}n_1k_2+\tfrac{N}{2}n_2k_2+(2n_2+n_1)k_3)\right)\Big]\\&=\sum_{n_2=0}^{1}\sum_{n_1=0}^{1}\Big[X_{4n_3+2n_1+n_2}(k_3)\cos\left((\pi n_1k_1)+(\pi n_2+\tfrac{\pi}{2}n_1)k_2+\theta(2n_2+n_1)k_3\right)\\&\quad-X_{4n_3+2n_1+n_2}(-k_3)\sin\left((\pi n_1k_1)+(\pi n_2+\tfrac{\pi}{2}n_1)k_2+\theta(2n_2+n_1)k_3\right)\Big]\end{aligned} \quad (13)$$

Equation (13) can be simplified further by expanding the first summation with index n_1, we get:

$$\begin{aligned}X(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)&=\sum_{n_2=0}^{1}\Big[X_{4n_3+n_2}(k_3)\cos\left(\pi n_2k_2+2\theta n_2k_3\right)-X_{4n_3+n_2}(-k_3)\sin\left(\pi n_2k_2+2\theta n_2k_3\right)\\&\quad+X_{4n_3+n_2+2}(k_3)\cos\left(\pi k_1+(\pi n_2+\tfrac{\pi}{2})k_2+\theta(2n_2+1)k_3\right)\\&\quad-X_{4n_3+n_2+2}(-k_3)\sin\left(\pi k_1+(\pi n_2+\tfrac{\pi}{2})k_2+\theta(2n_2+1)k_3\right)\Big]\end{aligned} \quad (14)$$

The second summation in (14), can also be expanded with index n_2 as:

$$\begin{aligned}X(k_3+\tfrac{N}{4}k_2+\tfrac{N}{2}k_1)&=X_{4n_3}(k_3)+\Big[X_{4n_3+2}(k_3)\cos\left(\pi k_1+\tfrac{\pi}{2}k_2+\theta k_3\right)-X_{4n_3+2}(-k_3)\sin\left(\pi k_1+\tfrac{\pi}{2}k_2+\theta k_3\right)\Big]\\&\quad+\Big[X_{4n_3+1}(k_3)\cos\left(\pi k_2+2\theta k_3\right)-X_{4n_3+1}(-k_3)\sin\left(\pi k_2+2\theta k_3\right)\Big]\\&\quad+\Big[X_{4n_3+3}(k_3)\cos\left(\pi k_1+\tfrac{3\pi}{2}k_2+3\theta k_3\right)-X_{4n_3+3}(-k_3)\sin\left(\pi k_1+\tfrac{3\pi}{2}k_2+3\theta k_3\right)\Big]\end{aligned} \quad (15)$$

Equation (15) is the general decomposition formula for the proposd radix-2^2 FHTDIT algorithm; solving it for k_1 and k_2 gives the desired output points. Thus, the first point of the decompositions $X(k)$ can be obtained by setting values of k_1 and k_2 in (15) to zero, yields:

$$\begin{aligned}X(k_3)&=X_{4n_3}(k_3)+\Big[X_{4n_3+1}(k_3)\cos\left(2\theta k_3\right)+X_{4n_3+1}(\tfrac{N}{4}-k_3)\sin\left(2\theta k_3\right)\Big]\\&\quad+\Big[X_{4n_3+2}(k_3)\cos\left(\theta k_3\right)+X_{4n_3+2}(\tfrac{N}{4}-k_3)\sin\left(\theta k_3\right)\Big]\\&\quad+\Big[X_{4n_3+3}(k_3)\cos\left(3\theta k_3\right)+X_{4n_3+3}(\tfrac{N}{4}-k_3)\sin\left(3\theta k_3\right)\Big]\end{aligned} \quad (16)$$

Using the following trigonometric identities:

$$\begin{aligned}
\cos\left(\tfrac{\pi}{2}+\theta k_3\right) &= \cos\left(\tfrac{\pi}{2}\right)\cos\left(\theta k_3\right)-\sin\left(\tfrac{\pi}{2}\right)\sin\left(\theta k_3\right) = -\sin\left(\theta k_3\right)\\
\sin\left(\tfrac{\pi}{2}+\theta k_3\right) &= \sin\left(\tfrac{\pi}{2}\right)\cos\left(\theta k_3\right)+\cos\left(\tfrac{\pi}{2}\right)\sin\left(\theta k_3\right) = \cos\left(\theta k_3\right)\\
\cos\left(\pi+\theta k_3\right) &= \cos\left(\pi\right)\cos\left(\theta k_3\right)-\sin\left(\pi\right)\sin\left(\theta k_3\right) = -\cos\left(\theta k_3\right)\\
\sin\left(\pi+\theta k_3\right) &= \sin\left(\pi\right)\cos\left(\theta k_3\right)+\cos\left(\pi\right)\sin\left(\theta k_3\right) = -\sin\left(\theta k_3\right)\\
\cos\left(\tfrac{3\pi}{2}+\theta k_3\right) &= \cos\left(\tfrac{3\pi}{2}\right)\cos\left(\theta k_3\right)-\sin\left(\tfrac{3\pi}{2}\right)\sin\left(\theta k_3\right) = \sin\left(\theta k_3\right)\\
\sin\left(\tfrac{\pi}{2}+\theta k_3\right) &= \sin\left(\tfrac{3\pi}{2}\right)\cos\left(\theta k_3\right)+\cos\left(\tfrac{3\pi}{2}\right)\sin\left(\theta k_3\right) = -\cos\left(\theta k_3\right)
\end{aligned} \tag{17}$$

Other decompositions can be computed as:

$$\begin{aligned}
X(k+\tfrac{N}{4}) = X_{4n}(k) &-\left[X_{4n+1}(k)\cos\left(2\theta k\right)+X_{4n+1}(\tfrac{N}{4}-k)\sin\left(2\theta k\right)\right]\\
&+\left[X_{4n+2}(\tfrac{N}{4}-k)\cos\left(\theta k\right)-X_{4n+2}(k)\sin\left(\theta k\right)\right]\\
&-\left[X_{4n+3}(\tfrac{N}{4}-k)\cos\left(3\theta k\right)-X_{4n+3}(k)\sin\left(3\theta k\right)\right]
\end{aligned} \tag{18}$$

$$\begin{aligned}
X(k+\tfrac{N}{2}) = X_{4n}(k) &+\left[X_{4n+1}(k)\cos\left(2\theta k\right)+X_{4n+1}(\tfrac{N}{4}-k)\sin\left(2\theta k\right)\right]\\
&-\left[X_{4n+2}(k)\cos\left(\theta k\right)+X_{4n+2}(\tfrac{N}{4}-k)\sin\left(\theta k\right)\right]\\
&-\left[X_{4n+3}(k)\cos\left(3\theta k\right)+X_{4n+3}(\tfrac{N}{4}-k)\cos\left(3\theta k\right)\right]
\end{aligned} \tag{19}$$

$$\begin{aligned}
X(k+\tfrac{3N}{4}) = X_{4n}(k) &-\left[X_{4n+1}(k)\cos\left(2\theta k\right)+X_{4n+1}(\tfrac{N}{4}-k)\sin\left(2\theta k\right)\right]\\
&-\left[X_{4n+2}(\tfrac{N}{4}-k)\cos\left(\theta k\right)-X_{4n+2}(k)\sin\left(\theta k\right)\right]\\
&+\left[X_{4n+3}(\tfrac{N}{4}-k)\cos\left(3\theta k\right)-X_{4n+3}(k)\sin\left(3\theta k\right)\right]
\end{aligned} \tag{20}$$

The in-place butterfly for the radix-2^3 FHT DIT algorithm can be obtained by combining 8 points together as shown in Figure 1.

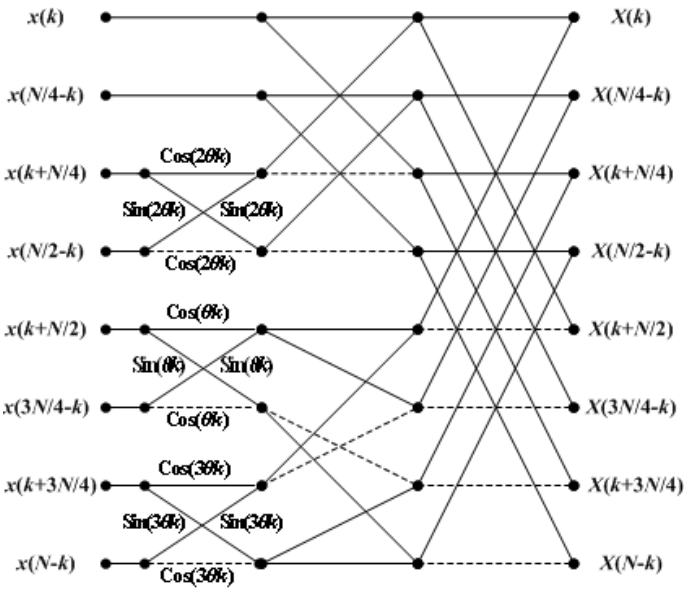

Figure 1. A butterfly of the radix-2^3 FHT DIT algorithm; where $\theta = 2\pi f/N$, dotted and solid lines stand for subtraction and additions respectively

By applying the procedure above repeatedly to the remaining DHTs of length $N/4$, the whole radix-2^2 DIT FHT algorithm is achieved. Figure 2 shows a 16-point example for calculating the DHT using the proposed radix-2^2 DIT algorithm.

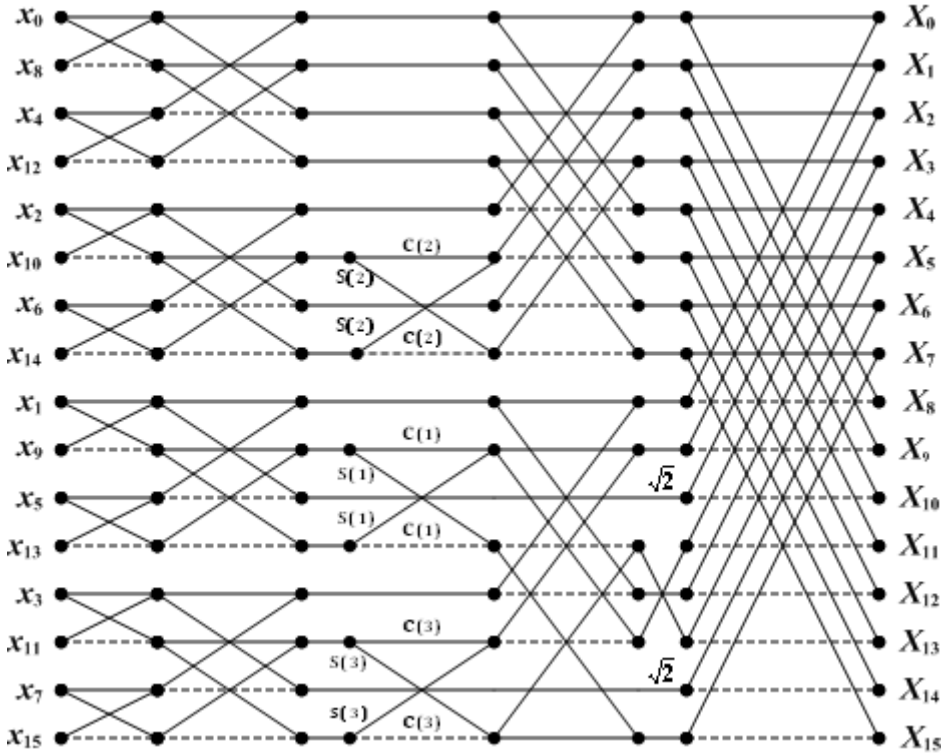

Figure 2. Signal flow graph for the 16-point radix-2^2 FHT DIT algorithm; where $C(t) = \cos(2\pi t/N)$ and $S(t) = \sin(2\pi t/N)$, dotted and solid lines stand for subtraction and additions respectively

4. ARITHMETIC COMPLEXITY

The performance of the developed algorithm is analysed in this section, by computing the number of multiplications and additions. This calculation is based on the in-place butterfly of the presented algorithm shown in Figure 1. In general, radix -2^2 DIT algorithm needs $\log_2 N$ stages of butterfly calculation. Every stage uses ($11N/4$-10) additions and ($3N/2$-12) multiplications. Moreover, four $N/4$ length DHTs must be computed, thus the complete radix -2^2 FHT algorithm fulfills the recurrences,

$$M(N) = 4M\left(\frac{N}{4}\right) + \frac{3N}{2} - 12$$
$$A(N) = 4A\left(\frac{N}{4}\right) + \frac{11N}{4} - 10 \quad (21)$$

Solving (21) by repeated substitutions of the initial values gives the closed form complexities. For this case, the initial values can be the number of operations that are required by length-4 and length-8 DHTs, which are equal to $M(4) = 0$, $A(4) = 8$, $M(8) = 2$ and $A(8) = 22$ respectively, we get:

$$M(N) = \frac{3N}{4}\log_2 N - \frac{5N}{2} + 4$$
$$A(N) = \frac{11N}{8}\log_2 N - \frac{19N}{12} + \frac{10}{3} \quad (22)$$

An assessment has been made between the developed algorithm, radix -2 and radix -4 algorithms with regard to the number of multiplications and additions, as depicted in Table1. The results of this assessment shows that the presented algorithm is better than radix -2 algorithm and obviously exceeding radix -4.

Table 1. Comparison in the number of multiplications and additions between radix-2^2, radix-2 and radix-4 DIT FHT algorithms

Transform Length	Radix-2^2 FHT algorithm		Radix-2 FHT algorithm [4]		Radix-4 FHT algorithm [9]	
N	*Mults.*	*Adds.*	*Mults.*	*Adds.*	*Mults.*	*Adds.*
8	2	22	4	26	-	-
16	12	66	20	74	14	70
32	44	166	68	194	-	-
64	132	430	196	482	142	450
128	356	1006	516	1154	-	-
256	900	2414	1284	2690	942	2498
512	2180	5422	3076	6146	-	-
1024	5124	12462	7172	13826	5294	12802
2048	11780	27310	16388	30722	-	-
4096	26628	61102	36868	67586	27310	624466

5. CONCLUSION

This paper has been devoted to the derivation of a new fast algorithm called radix-2^2 FHT based on decimation-in-time approach. The development of the presented algorithm is based on the in-place butterfly structure to provide a wider range of radices with better structural complexity and without increasing the required computational complexity. The presented algorithm is of substantial importance for hardware and software implementations, because this algorithm, along with the fact that the DHT is an efficient alternative to the DFT of real-valued data, makes it likely that for a single hardware or software module to be utilized for the calculationof the DHT, as well as for the calculation of the forward real-valued DFT and its inverse.

REFERENCES

[1] R. N. Bracewell, "Discrete Hartley transform," *Journal of the Optical Society of America,* vol. 73, pp. 1832-1835, 1983.
[2] R. N. Bracewell, Ed., "The Hartley transform," Oxford University Press, Inc., 1986.
[3] J. Liu, *et al.*, "Super-aperture metrology: overcoming a fundamental limit in imaging smooth highly curved surfaces," *Journal of microscopy,* 2015.
[4] J. Liu, *et al.*, "Synthetic complex superresolving pupil filter based on double-beam phase modulation," *Applied Optics,* vol. 47, pp. 3803-3807, 2008.
[5] Z. Liu, *et al*., "Image encryption based on double random amplitude coding in random Hartley transform domain," *Optik - International Journal for Light and Electron Optics,* vol. 121, pp. 959-964, 2010.
[6] Z. Liu, *et al.*, "Color image encryption by using the rotation of color vector in Hartley transform domains," *Optics and Lasers in Engineering,* vol. 48, pp. 800-805, 2010.
[7] Z. Liu, *et al.*, "Optical color image hiding scheme based on chaotic mapping and Hartley transform," *Optics and Lasers in Engineering,* vol. 51, pp. 967-972, 2013.
[8] A. V. Oppenheim, *et al.*, "Discrete-time signal processing, Prentice hall Englewood Cliffs, NJ, vol. 2, 1989.
[9] K. Jones, "Design and parallel computation of regularised fast Hartley transform," in *Vision, Image and Signal Processing, IEE Proceedings-*, pp. 70-78, 2006.
[10] D. Narmadha, *et al.*, "An Optimal HSI Image Compression using DWT and CP," *International Journal of Electrical and Computer Engineering,* vol. 4, pp. 411-421, 2014.
[11] R. N. Bracewell, "The fast Hartley transform," *Proceedings of the IEEE,* vol. 72, pp. 1010-1018, 1984.
[12] H. J. Meckelburg and D. Lipka, "Fast Hartley transform algorithm," *Electronics Letters,* vol. 21, pp. 341-343, 1985.
[13] C. Kwong and K. Shiu, "Structured fast Hartley transform algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing, ,* vol. 34, pp. 1000-1002, 1986.
[14] H. S. Hou, "The fast Hartley transform algorithm," *IEEE Transactions on Computers,* vol. 100, pp. 147-156, 1987.
[15] H. Sorensen, *et al.*, "On computing the discrete Hartley transform," *IEEE Transactions on Acoustics, Speech and Signal Processing.,* vol. 33, pp. 1231-1238, 1985.
[16] C. Burrus, "Index mappings for multidimensional formulation of the DFT and convolution," *IEEE Transactions on Acoustics, Speech and Signal Processing.,* vol. 25, pp. 239-242, 1977.
[17] H. Malvar, "Fast computation of discrete cosine transform through fast Hartley transform," *Electronics Letters,* vol. 22, pp. 352-353, 1986.
[18] Y. H. Chan and W. C. Siu, "New formulation of fast discrete Hartley transform with the minimum number of multiplications," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, vol. 1, pp. 323-326, 1991.
[19] P. Duhamel and M. Vetterli, "Improved Fourier and Hartley transform algorithms: Application to cyclic convolution of real data," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 35, pp. 818-824, 1987.
[20] R. Bracewell, "Alternative to split-radix Hartley transform," *Electronics Letters,* vol. 23, pp. 1148-1149, 1987.

[21] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *Parallel Processing Symposium, The 10th International Proceedings of IPPS'96,* pp. 766-770, 1996.
[22] A. Cortés, *et al.*, "Radix FFTs: Matricial representation and SDC/SDF pipeline implementation," *IEEE Transactions on Signal Processing, ,* vol. 57, pp. 2824-2839, 2009.
[23] I. A. Qureshi and F. Qureshi, "Impact of FFT algorithm selection on switching activity and coefficient memory size," *TELKOMNIKA Indonesian Journal of Electrical Engineering,* vol. 12, pp. 6224-6229, 2014.
[24] M. T. Hamood and S. Boussakta, "New radix-based FHT algorithm for computing the discrete Hartley transform," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11),* pp. 1581-1584, 2011.
[25] C. S. Burrus, "Unscrambling for fast DFT algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing.,* vol. 36, pp. 1086-1087, 1988.

BIOGRAPHIES OF AUTHORS

Mounir Taha Hamood received the B.Sc. degree in electrical engineering from University of Technology, Baghdad, Iraq in 1990 and the M.Sc. degree in electronic and communications engineering from Al-Nahrain University, Baghdad, Iraq in 1995. He graduated from Newcastle University, Newcastle upon Tyne, U.K in 2012 with the PhD degree in communications and signal processing. His doctoral research was in the development of efficient algorithms for fast computation of discrete transforms.
He is currently a Lecturer in Signal Processing for Communications at the Department of Electrical Engineering, College of Engineering, Tikrit University, Tikrit, Iraq. His research interest include discrete transforms, fast algorithms for digital signal processing in one and multidimensional applications, and communication systems.