NOVEL RECURSIVE ALGORITHM FOR REALIZATION OF ONE-DIMENSIONAL DISCRETE HARTLEY TRANSFORM

M. N. Murty
Department of Physics, National Institute of Science & Technology,
Berhampur - 761008, Odisha, India
Email: mnarayanamurty@rediffmail.com

ABSTRACT

Discrete Hartley transform is an important tool in digital signal processing. This paper presents a novel recursive algorithm for realization of one-dimensional discrete Hartley transform of even length. The transform is constructed by single folding of input data and using Chebyshev Polynomial. Single folding algorithm provides data throughput two times of that achieved by the conventional methods. Compared to some other algorithms, the proposed algorithm achieves savings on the number of additions and multiplications. The recursive algorithms are appropriate for VLSI implementation.

Keywords: *Digital signal processing, Discrete Hartley transform, Recursive algorithm.*

1. INTRODUCTION

The discrete Hartley transform (DHT) [1], [2] plays an important role in many digital signal processing (DSP) applications since it is a good alternative to the discrete Fourier transform (DFT) for its real-number operations. One of the main attractions of DHT is that it only involves real computations in contrast to complex computations in the DFT. In addition, the inverse DHT has the same form as the forward DHT, except for a scaling factor. Therefore, a single kind of program or architecture can be used to carry out both the forward and inverse DHT computations.

Over the years, the DHT has been established as a potential tool for signal processing and communication applications, e.g., computation of circular convolution, and deconvolution [3], [4], interpolation of real-valued signals [5], image compression [6], [7], error control coding [8], adaptive filtering [9], multi-carrier modulation and many other applications [10]-[12]. Fast implementation of one-dimensional (1-D) DHT has attracted many attentions [13]-[15]. However, DHT is computation intensive.

In this paper, a novel recursive algorithm for realization of 1-D DHT of even length by single folding of input data is proposed and the DHT for transform lengths 4 and 6 is implemented by recursive structures.

The rest of the paper is organized as follows. The derivation of recursive algorithm for 1-D DHT is presented in Section 2. Two examples for realization of 1-D DHT are given in Section 3. The conclusion is presented in Section 4. Finally, references are given in section 5.

2. PROPOSED RECURSIVE ALGORITHM FOR 1-D DHT

The one dimensional Discrete Hartley Transform for real input data sequence $X(n)$ of length N is defined as

$$H(k)=\sum_{n=0}^{N-1} X(n)\, cas\left(\frac{2\pi kn}{N}\right)$$

$$\text{for } k = 0, 1, 2, \ldots, N-1 \qquad (1)$$

where $cas\left(\frac{2\pi kn}{N}\right)$ is the transform's Kernel and $cas\,\theta = \cos\theta + \sin\theta$.

The H values represent the transformed data.

Single Data Folding (SDF) folds $H(k)$ as below, when $N = 2*m$ and $m \geq 1$.

$$H(k)=\sum_{n=0}^{\frac{N}{2}-1}\left[X(n)+(-1)^k X\left(n+\frac{N}{2}\right)\right] cas\left(\frac{2\pi kn}{N}\right) \qquad (2)$$

$$= \sum_{n=0}^{\frac{N}{2}-1} W_k(n)\, cas\left(\frac{2\pi kn}{N}\right) \tag{3}$$

where $$W_k(n) = X(n) + (-1)^k X\left(n + \frac{N}{2}\right) \tag{4}$$

Replacing n by $\frac{N}{2} - 1 - n$ in (3), $H(k)$ becomes

$$H(k) = \sum_{n=0}^{\frac{N}{2}-1} W_k\left(\frac{N}{2} - 1 - n\right) cas\left[\frac{2\pi k}{N}\left(\frac{N}{2} - 1 - n\right)\right]$$

$$= \sum_{n=0}^{\frac{N}{2}-1} W_k\left(\frac{N}{2} - 1 - n\right)\left[\cos\left[\pi k - \frac{2\pi k}{N}(n+1)\right] + \sin\left[\pi k - \frac{2\pi k}{N}(n+1)\right]\right] \tag{5}$$

Divide (5) into two parts according to whether k is even or odd.

2.1 *k* Even

When k is even, $\cos \pi k = 1$ and $\sin \pi k = 0$. Then (5) becomes

$$H(k) = \sum_{n=0}^{\frac{N}{2}-1} W_k\left(\frac{N}{2} - 1 - n\right)\left[\cos\left[\frac{2\pi k}{N}(n+1)\right] - \sin\left[\frac{2\pi k}{N}(n+1)\right]\right] \tag{6}$$

$$= G_{\frac{N}{2}-1}(k) - I_{\frac{N}{2}-1}(k)$$

where

$$G_{\frac{N}{2}-1}(k) = \sum_{n=0}^{\frac{N}{2}-1} W_k\left(\frac{N}{2} - 1 - n\right)\cos\left[\frac{2\pi k}{N}(n+1)\right] \tag{7}$$

$$I_{\frac{N}{2}-1}(k) = \sum_{n=0}^{\frac{N}{2}-1} W_k\left(\frac{N}{2} - 1 - n\right)\sin\left[\frac{2\pi k}{N}(n+1)\right] \tag{8}$$

Equations (7) and (8) are written as

$$G_j(k) = \sum_{n=0}^{j} W_k(j-n)\cos[(n+1)\theta_k] \tag{9}$$

$$I_j(k) = \sum_{n=0}^{j} W_k(j-n)\sin[(n+1)\theta_k] \tag{10}$$

where $j = \frac{N}{2} - 1$ and $\theta_k = \frac{2\pi k}{N}$.

Then (6) can be written as

$$H(k) = G_j(k) - I_j(k) \tag{11}$$

It is easy to show the following trigonometric identities.

$$\cos(r\theta_k) = 2\cos[(r-1)\theta_k]\cos\theta_k - \cos[(r-2)\theta_k] \tag{12}$$

$$\sin(r\theta_k) = 2\sin[(r-1)\theta_k]\cos\theta_k - \sin[(r-2)\theta_k] \tag{13}$$

These equations are known as the Chebyshev Polynomial. Using the recursive expression in (12), $G_j(k)$ given by (9) can be written as

$$\begin{aligned}
G_j(k) &= \sum_{n=0}^{j} W_k(j-n)\left[\ 2\cos n\theta_k \cos\theta_k - \cos[(n-1)\theta_k]\ \right] \\
&= 2\cos\theta_k\left[W_k(j) + \sum_{n=1}^{j} W_k(j-n)\cos n\theta_k\right] \\
&\qquad - \left[W_k(j)\cos\theta_k + W_k(j-1) + \sum_{n=2}^{j} W_k(j-n)\cos[(n-1)\theta_k]\right] \\
&= 2\cos\theta_k\left[W_k(j) + \sum_{n=0}^{j-1} W_k(j-1-n)\cos[(n+1)\theta_k]\right] \\
&\qquad - \left[W_k(j)\cos\theta_k + W_k(j-1) + \sum_{n=0}^{j-2} W_k(j-2-n)\cos[(n+1)\theta_k]\right]
\end{aligned}$$

Using (9) in the above expression,

$$G_j(k) = W_k(j)\cos\theta_k + 2\cos\theta_k G_{j-1}(k) - W_k(j-1) - G_{j-2}(k) \tag{14}$$

Similarly, applying (13) to (10), $I_j(k)$ can be written as

$$\begin{aligned}
I_j(k) &= \sum_{n=0}^{j} W_k(j-n)\left[\ 2\sin n\theta_k \cos\theta_k - \sin[(n-1)\theta_k]\ \right] \\
&= 2\cos\theta_k\left[\sum_{n=1}^{j} W_k(j-n)\sin n\theta_k\right] + W_k(j)\sin\theta_k - \sum_{n=2}^{j} W_k(j-n)\sin[(n-1)\theta_k] \\
&= 2\cos\theta_k\left[\sum_{n=0}^{j-1} W_k(j-1-n)\sin[(n+1)\theta_k]\right] + W_k(j)\sin\theta_k \\
&\qquad - \sum_{n=0}^{j-2} W_k(j-2-n)\sin[(n+1)\theta_k]
\end{aligned}$$

Using (10) in the above expression

$$I_j(k) = W_k(j)\sin\theta_k + 2\cos\theta_k I_{j-1}(k) - I_{j-2}(k) \tag{15}$$

Using (14) and (15) in (11), $H(k)$ can be written as

$$\begin{aligned}
H(k) = &\left[\ W_k(j)\cos\theta_k - W_k(j-1) + 2\cos\theta_k G_{j-1}(k) - G_{j-2}(k)\ \right] \\
&- \left[\ W_k(j)\sin\theta_k + 2\cos\theta_k I_{j-1}(k) - I_{j-2}(k)\ \right]
\end{aligned} \tag{16}$$

2.2 *k* Odd

When k is odd, $\cos\pi k = -1$ and $\sin\pi k = 0$. Then (5) becomes

$$H(k)=\sum_{n=0}^{\frac{N}{2}-1} W_k\left(\frac{N}{2}-1-n\right)\left[\sin\left[\frac{2\pi k}{N}(n+1)\right]-\cos\left[\frac{2\pi k}{N}(n+1)\right]\right]$$

Using (7)-(10), the above expression can be written as

$$H(k)=I_j(k)-G_j(k) \tag{17}$$

Then using (14) and (15), (17) becomes

$$\begin{aligned} H(k)=&\left[\ W_k(j)\sin\theta_k+2\cos\theta_k I_{j-1}(k)-I_{j-2}(k)\ \right] \\ &-\left[\ W_k(j)\cos\theta_k-W_k(j-1)+2\cos\theta_k G_{j-1}(k)-G_{j-2}(k)\ \right]\end{aligned} \tag{18}$$

The 1-D DHT can be realized using the recursive relations (16) and (18).

3. EXAMPLES

3.1 *k* Even and *N* = 4

Let us use a 4-point DHT with input sequence $\{X(n): n=0,1,2,3\}$ to clarify our proposal.

From (4), we get

$$\begin{aligned} W_k(0)&=X(0)+X(2) \\ W_k(1)&=X(1)+X(3)\end{aligned} \tag{19}$$

As $j=\frac{N}{2}-1=1$ for $N=4$, using recursive relations (14) and (15), $H(k)$ in (16) is given by

$$\begin{aligned} H(k)=&[W_k(0)2\cos\theta_k+W_k(1)]\cos\theta_k-W_k(0)\cos\theta_k+W_k(0)[\cos\theta_k-1] \\ &-[W_k(0)2\cos\theta_k+W_k(1)]\sin\theta_k\end{aligned} \tag{20}$$

Equation (20) can be realized using the recursive structure shown in Fig. 1.

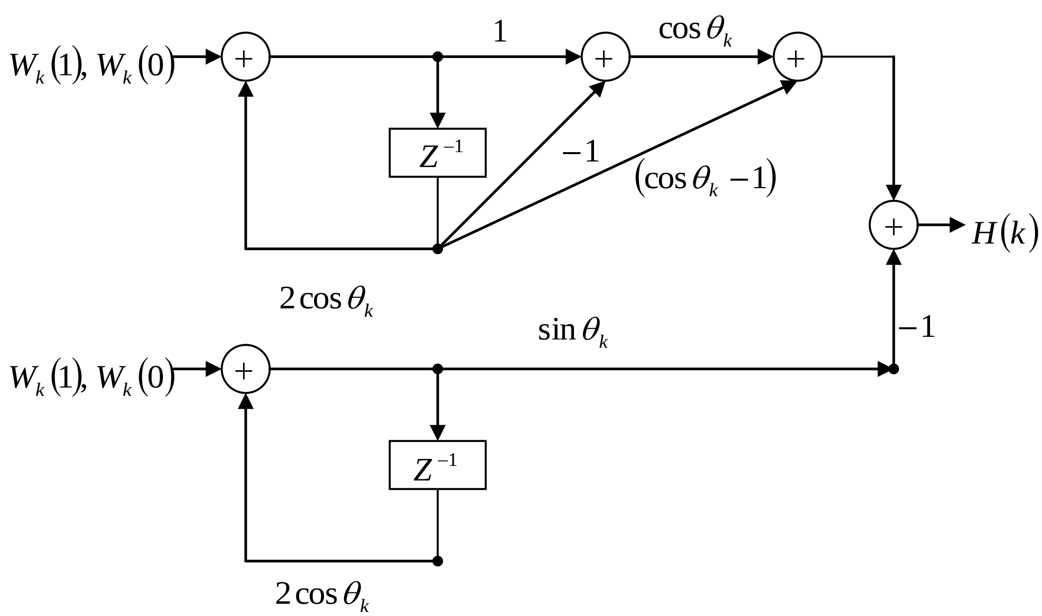

Figure 1. *Recursive structure for computing the DHT for k even and N = 4*

3.2 *k* Odd and *N* = 4

Similarly using (14) and (15), $H(k)$ in (18) is given by

$$H(k) = [W_k(0)2\cos\theta_k + W_k(1)]\sin\theta_k - \left[[W_k(0)2\cos\theta_k + W_k(1)]\cos\theta_k - W_k(0)\cos\theta_k + W_k(0)(\cos\theta_k - 1) \right] \quad (21)$$

where

$$\begin{aligned} W_k(0) &= X(0) - X(2) \\ W_k(1) &= X(1) - X(3) \end{aligned} \quad (22)$$

Equation (21) can be realized using the recursive structure shown in Fig. 2.

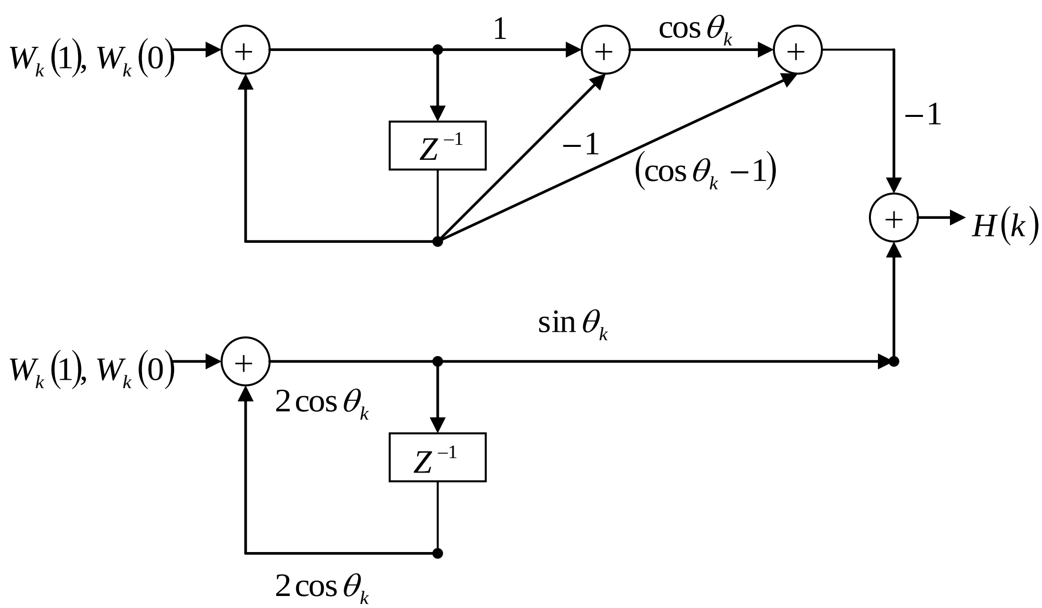

Figure 2. *Recursive structure for computing the DHT for k odd and N = 4*

3.3 *k* Even and *N* = 6

Consider 6-point DHT with input sequence $\{X(n): n = 0,1,2,3,4,5\}$

Using (14) and (15), $H(k)$ in (16) for N = 6 is given by

$$\begin{aligned} H(k) = \Big\{ &\left[W_k(2) + 2\cos\theta_k [W_k(1) + W_k(0)2\cos\theta_k - W_k(0)] \right]\cos\theta_k \\ &- W_k(0)\cos\theta_k + W_k(0)[\cos\theta_k - 1] - W_k(1)\cos\theta_k \\ &+ W_k(1)[\cos\theta_k - 1] + W_k(0)[2\cos\theta_k - 1][\cos\theta_k - 1] \Big\} \\ &- \left\{ \left[W_k(2) + 2\cos\theta_k [W_k(0)2\cos\theta_k + W_k(1)] \right]\sin\theta_k - W_k(0)\sin\theta_k \right\} \end{aligned} \quad (23)$$

From (4), we have

$$
\begin{aligned}
W_k(0) &= X(0) + X(3) \\
W_k(1) &= X(1) + X(4) \\
W_k(2) &= X(2) + X(5)
\end{aligned}
\tag{24}
$$

Equation (23) can be realized using the recursive structure shown in Fig. 3.

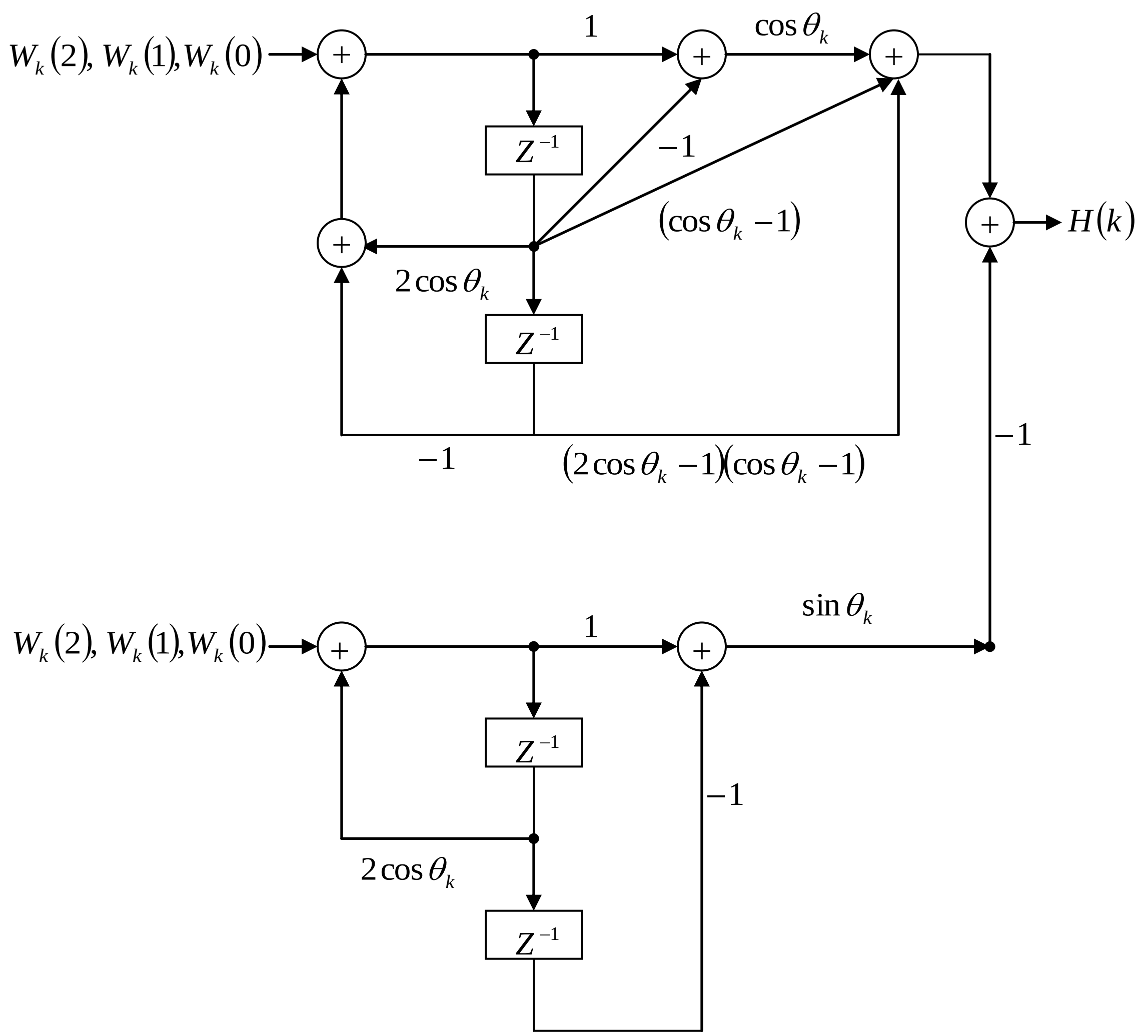

Figure 3. *Recursive structure for computing the DHT for k even and N = 6*

3.4 *k* Odd and *N* = 6

Similarly the DHT for k odd and $N = 6$ can be realized using the recursive structure shown in Fig. 4. For this case, we have

$$
\begin{aligned}
W_k(0) &= X(0) - X(3) \\
W_k(1) &= X(1) - X(4) \\
W_k(2) &= X(2) - X(5)
\end{aligned}
\tag{25}
$$

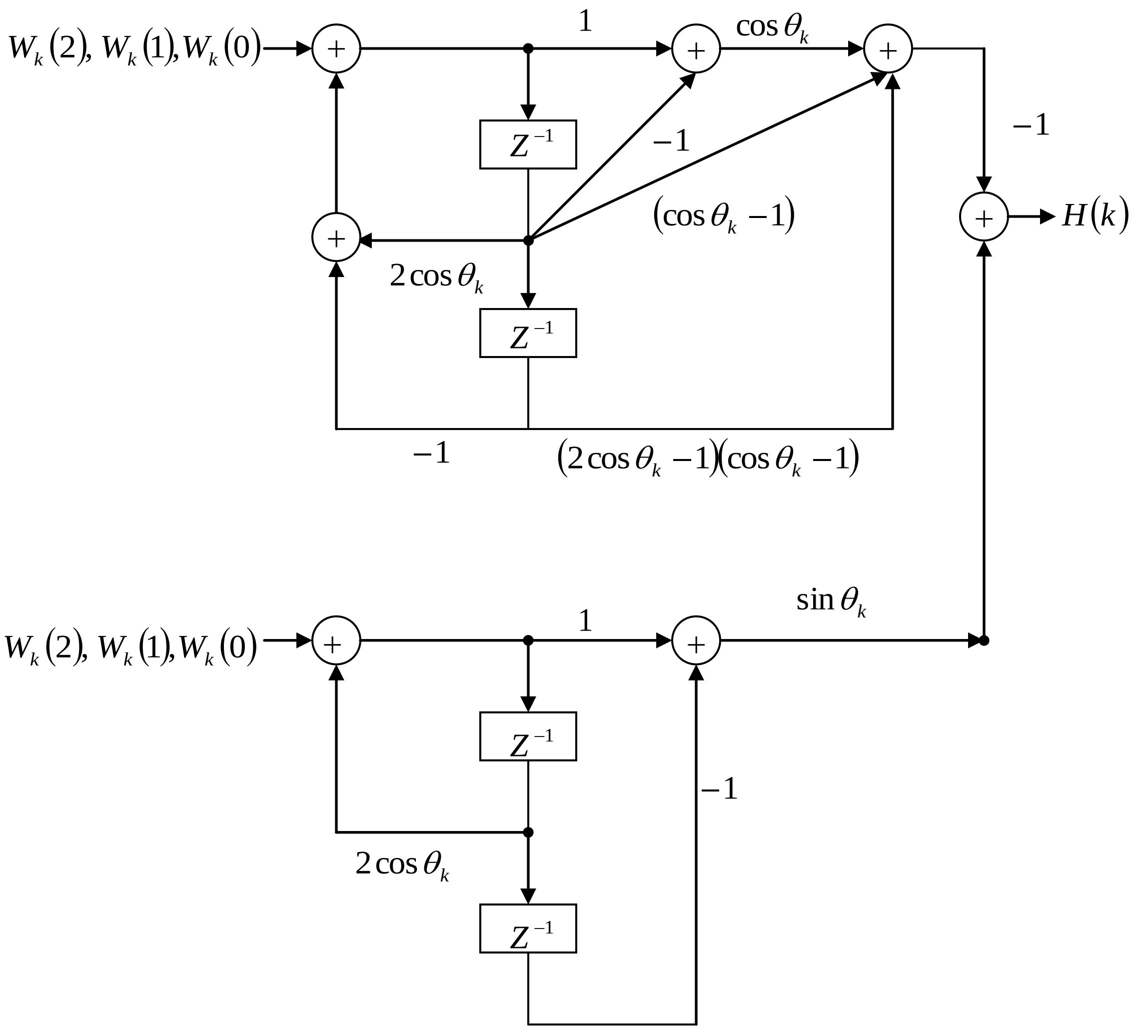

Figure 4. *Recursive structure for computing the DHT for k odd and N = 6*

Tables I shows the comparison of adders and multipliers required by the proposed structure with some existing structures.

Table 1. *Comparison of the number of adders & multipliers required by different algorithms to compute the N-Point DHT.*

N	No. of Adders					No. of Multipliers				
	[16]	[17]	[18] DIF type	[18] DIT type	Proposed	[16]	[17]	[18] DIF type	[18] DIT type	Proposed
4	8	18	10	11	7	16	16	10	10	6
6	12	28	14	15	15	24	24	14	14	12

It can be seen that the proposed algorithm requires less number of additions and multiplications in comparison with other architectures [16] – [18].

4. CONCLUSION

In this paper, new recursive formulas are proposed for realizing 1-D DHT of even transform length *N*. The recursive algorithms are derived by single folding of input data for both even and odd component parts *k*. Single folding algorithm provides data throughput two times of that achieved by the conventional methods. The number of adders and multipliers used in the proposed algorithm are less compared with that of some existing structures. Multiplications are more time consuming than additions. Since the number of multiplications in the proposed algorithm are considerably less in comparison with some other structures, saving in time can be achieved by the proposed algorithm in its realization. As DHT is computation intensive, the recursive structures are changing with the change of transform length *N*. Recursive structures are suitable for parallel VLSI implementation.

5. REFERENCES

[1] R. N. Bracewell, "Discrete Hartley transform, " J. Opt. Soc. Amer., vol. 73, no. 12, pp. 1832-1835, 1983.

[2] R. N. Bracewell, "The fast Hartley transform, " Proc. IEEE, vol. 72, pp. 1010-1018, 1984.

[3] R. N. Bracewell, The Fourier Transform and its Applications, 3rd ed., New York: McGraw-Hill, 2000.

[4] L. –Z. Cheng, L. Tong, and Z. –R. Jiang, "Real transform algorithm for computing discrete circular deconvolution," in Proc. 3rd IEEE Int. Conf. Signal Process., vol. 1, pp. 166-169, Oct. 1996.

[5] G. S. Maharana and P. K. Meher, "Algorithm for efficient interpolation of real-valued signals using discrete Hartley transform," Comp. Elect. Eng., vol. 23, no. 2, pp. 129-134, Mar. 1997.

[6] K. S. Tzou and T. R. Hsing, "A study of discrete Hartley transform for image compression application," in Proc. SPIE Int. Soc. Opt. Eng., vol. 534, Jan. 1985.

[7] C. H. Paik and M. D. Fox, "Fast Hartley transform for image processing," IEEE Trans. Med. Imag., vol. 7, no. 6, pp. 149-153, Jun. 1988.

[8] J. –L. Wu and J. Shiu, "Discrete Hartley transform in error control coding," IEEE Trans. Signal Process., vol. 39, no. 10, pp. 2356-2359, Oct. 1991.

[9] P. K. Meher and G. Panda, "Unconstrained Hartley-domain least mean square adaptive filter," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 40, no. 9, pp. 582-585, Sep. 1993.

[10] C. –L. Wang, C. –H. Chang, J. L. Fan, and J. M. Cioffi, "Discrete Hartley transform based multicarrier modulation," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'00), vol. 5, Jun. 2000, pp. 2513-2516.

[11] R. N. Bracewell, "Aspects of the Hartley transform," Proc. IEEE, vol. 82, no. 3, pp. 381-387, Mar. 1994.

[12] C. –L. Wang, C. –H. Chang, "A DHT-based FFT/IFFT processor for VDSL transceivers," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'01), vol. 2, May 2001, pp. 1213-1216.

[13] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A New split-radix FHT algorithm for length – $q*2^m$," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 10, pp. 2031-2043, Oct. 2004.

[14] P. K. Meher, T. Srikanthan, and J. C. Patra, "Scalable and modular memory-based systolic architectures for discrete Hartley transform," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 5, pp. 1065-1077, May 2006.

[15] P. K. Meher, J. C. Patra, and M. N. S. Swamy, "High-throughput memory-based architecture for DHT using a new convolutional formulation," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 7, pp. 606-610, Jul. 2007.

[16] L. –W. Chang and S. –W. Lee, "Systolic arrays for the discrete Hartley transform," IEEE Trans. Acoust., Speech, Signal Processing, vol. 39, pp. 2411-2418, Nov. 1992.

[17] K. J. R. Liu and C. –T. Chiu, "Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms," IEEE Trans. Acoust., Speech, Signal Processing, vol. 41, pp. 1357-1377, Mar. 1993.

[18] W. –H. Fang and J. –D. Lee, "Efficient CORDIC-based systolic architectures for the discrete Hartley transform," IEE Proc. – Comput. Digit. Tech., vol. 142, pp. 201-207, May 1995.