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

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

for $k = 0, 1, 2, ..., N-1$ (1)

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

The $H$ values represent the transformed data.

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

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

(2)
\[
W_k(n) = X(n) + (-1)^n X(n + \frac{N}{2})
\]

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

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

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

2.1 \( k \) Even

When \( k \) is even, \( \cos \frac{2\pi k}{N} = 1 \) and \( \sin \frac{2\pi k}{N} = 0 \). Then (5) becomes

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

where

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

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

Equations (7) and (8) are written as

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

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

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

Then (6) can be written as

\[
H(k) = G_j(k) - I_j(k)
\]

It is easy to show the following trigonometric identities.

\[
\cos(r\theta_k) = 2\cos[(r-1)\theta_k]\cos\theta_k - \cos[(r-2)\theta_k]
\]
\[ \sin(r\theta_k) = 2\sin[(r-1)\theta_k]\cos \theta_k - \sin[(r-2)\theta_k] \]  

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

\[
G_j(k) = \sum_{n=0}^{j} W_k(j-n) \left[ 2\cos n\theta_k \cos \theta_k - \cos[(n-1)\theta_k] \right] 
\]

\[
= 2\cos \theta_k \left[ W_k(j) + \sum_{n=1}^{j} W_k(j-n)\cos n\theta_k \right] 
- W_k(j)\cos \theta_k + W_k(j-1) + \sum_{n=2}^{j} W_k(j-n)\cos[(n-1)\theta_k] 
\]

\[
= 2\cos \theta_k \left[ W_k(j) + \sum_{n=0}^{j-1} W_k(j-1-n)\cos[(n+1)\theta_k] \right] 
- W_k(j)\cos \theta_k + W_k(j-1) + \sum_{n=0}^{j-2} W_k(j-2-n)\cos[(n+1)\theta_k] 
\]

Using (9) in the above expression,

\[
G_j(k) = W_k(j)\cos \theta_k + 2\cos \theta_k G_{j-1}(k) - W_k(j-1) - G_{j-2}(k) 
\]  

(14)

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

\[
I_j(k) = \sum_{n=0}^{j} W_k(j-n) \left[ 2\sin n\theta_k \cos \theta_k - \sin[(n-1)\theta_k] \right] 
\]

\[
= 2\cos \theta_k \left[ \sum_{n=1}^{j} W_k(j-n)\sin n\theta_k \right] + W_k(j)\sin \theta_k - \sum_{n=2}^{j} W_k(j-n)\sin[(n-1)\theta_k] 
\]

\[
= 2\cos \theta_k \left[ \sum_{n=0}^{j-1} W_k(j-1-n)\sin[(n+1)\theta_k] \right] + W_k(j)\sin \theta_k 
- \sum_{n=0}^{j-2} W_k(j-2-n)\sin[(n+1)\theta_k] 
\]

Using (10) in the above expression

\[
I_j(k) = W_k(j)\sin \theta_k + 2\cos \theta_k I_{j-1}(k) - I_{j-2}(k) 
\]  

(15)

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

\[
H(k) = \begin{bmatrix} W_k(j)\cos \theta_k - W_k(j-1) + 2\cos \theta_k G_{j-1}(k) - G_{j-2}(k) \\
- W_k(j)\sin \theta_k + 2\cos \theta_k I_{j-1}(k) - I_{j-2}(k) \end{bmatrix} 
\]  

(16)

2.2 k Odd

When \( k \) is odd, \( \cos \frac{\pi k}{2} = -1 \) and \( \sin \frac{\pi k}{2} = 0 \). Then (5) becomes
Using (7)-(10), the above expression can be written as

\[ H(k) = I_j(k) - G_j(k) \]  (17)

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

\[
H(k) = \left[ W_k(j) \sin \theta_k + 2 \cos \theta_k I_{j-1}(k) - I_{j-2}(k) \right] - \left[ W_k(j) \cos \theta_k - W_k(j-1) + 2 \cos \theta_k G_{j-1}(k) - G_{j-2}(k) \right]
\]  (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

\[
W_k(0) = X(0) + X(2) \\
W_k(1) = X(1) + X(3)
\]  (19)

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

\[
H(k) = \left[ W_k(0) 2 \cos \theta_k + W_k(1) \right] \cos \theta_k - W_k(0) \cos \theta_k + W_k(0) \cos \theta_k - 1 - \left[ W_k(0) 2 \cos \theta_k + W_k(1) \right] \sin \theta_k
\]  (20)

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

![Recursive structure for computing the DHT for k even and N = 4](image-url)
\[ H(k) = \left[ W_k(0)2\cos\theta_k + W_k(1)\right] \sin\theta_k \]

\[ -\left[ W_k(0)2\cos\theta_k + W_k(1)\right] \cos\theta_k - W_k(0)\cos\theta_k + W_k(0)(\cos\theta_k - 1) \]  \hspace{1cm} (21)

where

\[ W_k(0) = X(0) - X(2) \]

\[ W_k(1) = X(1) - X(3) \]  \hspace{1cm} (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**

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

\[ -\left[ W_k(0)2\cos\theta_k + W_k(1)\right] \cos\theta_k - W_k(0)\cos\theta_k + W_k(0)(\cos\theta_k - 1) \]  \hspace{1cm} (23)

From (4), we have
Equation (23) can be realized using the recursive structure shown in Fig. 3.

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

$$W_k(0) = X(0) - X(3)$$
$$W_k(1) = X(1) - X(4)$$
$$W_k(2) = X(2) - X(5)$$

(25)
Table 1 shows the comparison of adders and multipliers required by the proposed structure with some existing structures.

<table>
<thead>
<tr>
<th>N</th>
<th>No. of Adders</th>
<th>No. of Multipliers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DIF type</td>
<td>DIT type</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>28</td>
</tr>
</tbody>
</table>

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