Systolic Architecture for Realization of Two Dimensional Discrete Hartley Transform

M.Narayan Murty¹ and H.Panda²

¹(Visiting Faculty, Department of Physics, Sri Sathya Sai Institute of Higher Learning, Prasanthi Nilayam – 515134, Andhra Pradesh, India)  
²(Department of physics, S.K.C.G. Autonomous College, Paralakhemundi -761200, Odisha, India)

Abstract: Discrete Hartley transform is an important tool in digital signal processing. This paper presents an algorithm for realization of two dimensional discrete Hartley transform (2-D DHT) of size N×N using systolic architecture. The 2-D DHT can be realized in two stages. For computation of these two stages, two systolic architectures are presented for realization of 2-D DHT of size 3×3. These two systolic architectures can be orthogonally interspersed together for realization 2-D DHT without requiring any transposition buffer. The number of processing elements (PEs) used in the combined structure is 2N². Each PE consists of one multiplier, one adder and one register.

Keywords: Discrete Hartley transform, discrete Fourier transform, systolic architecture.

I. 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-carryer 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.

Several algorithms for the fast computation of multi-dimensional DHT are available in the literature. But the 2D-DHT is the most popular among the DHTs of higher dimensions due to its applications in various signal processing and image processing applications [16]-[19].

This paper presents an algorithm for realization of two dimensional discrete Hartley transform (2-D DHT) of size N×N using systolic architecture. The 2-D DHT can be realized in two stages. For computation of these two stages, two systolic architectures are presented for realization of 2-D DHT of size 3×3. These two systolic architectures can be interspersed together without the transposition buffer. The systolic architecture has the following characteristics:

- A massive and non-centralized parallelism
- Local communications
- Synchronous evaluation

The systolic arrays are used in the design and implementation of high performance digital signal processing equipment. Systolic architectures are established as the most popular and dominant class of VLSI structures due to the simplicity of their processing elements (PEs), modularity of their structure, regular and nearest neighbour interconnections between the PEs, high level of pipelinability, small chip area and lower dissipation. In the systolic architectures, the desired data are pumped rhythmically in regular intervals across the PEs for yielding high throughput by fully pipelined processing. The systolic array concept can also be exploited at bit level in the design of individual VLSI chips. The highly regular structure of systolic circuits renders them comparatively easy to design and test.

The rest of the paper is organized as follows. The proposed algorithm for 2-D DHT is presented in Section-II. The systolic architecture for implementation of 2-D DHT of size 3×3 is presented in Section-III. Conclusion is given in Section-IV.
II. Proposed Algorithm For 2-D DHT

The one dimensional discrete Hartley transform (1-D DHT) for real input data sequence \( \{x(m); m = 0, 1, 2, ..., N - 1\} \) of length \( N \) is defined as

\[
H(k) = \sum_{m=0}^{N-1} x(m) \cos \left( \frac{2\pi km}{N} \right) \cos \left( \frac{2\pi l n}{N} \right)
\]

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

Where \( \cos \left( \frac{2\pi km}{N} \right) \) is the transform’s kernel and \( \cos \theta = \cos \theta + \sin \theta \). The \( H \) values represent the transformed data. The 1-D DHT defined by (1) can be extended to two dimensional discrete Hartley transform (2-D DHT).

The 2-D DHT of input data array \( x(m, n) \) of size \( N \times N \) is given by

\[
H(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) C_N^{km} C_N^{ln}
\]

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

Defining \( C_N^{ij} = \cos \left( \frac{2\pi ij}{N} \right) \), (2) can be written as

\[
H(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) C_N^{km} C_N^{ln}
\]

Define

\[
T(k, n) = \sum_{m=0}^{N-1} x(m, n) C_N^{km}
\]

Substituting (4) in (3), we have

\[
H(k, l) = \sum_{n=0}^{N-1} T(k, n) C_N^{ln} \quad \text{for} \ k, l = 0, 1, 2, ..., N - 1.
\]

The 2-D DHT given by (5) can be realized by the following two steps.

Step 1: computation of \( T(k, n) \) given by (4) using a systolic architecture.

Step 2: computation of \( H(k, l) \) given by (5) using a systolic architecture similar to that used in step 1.

An example for realizing the 2-D DHT of size \( N \times N = 3 \times 3 \) is given in the next section.

III. Systolic Architecture For Implementation of 2-D DHT of Size 3×3

To clarify the proposed algorithm, a 2-D DHT of size 3×3 is considered.

3.1 Step 1:

Substituting \( n = 0, 1, 2 \) successively in (4), we obtain the following three expressions for \( N = 3 \) & \( k = 0 \).

\[
T(0, 0) = x(0, 0) C_3^{00} + x(1, 0) C_3^{01} + x(2, 0) C_3^{02}
\]

\[
T(0, 1) = x(0, 1) C_3^{00} + x(1, 1) C_3^{01} + x(2, 1) C_3^{02}
\]

\[
T(0, 2) = x(0, 2) C_3^{00} + x(1, 2) C_3^{01} + x(2, 2) C_3^{02}
\]

Substituting \( N = 3, k = 1 \) & \( n = 0, 1, 2 \) in (4), we have

\[
T(1, 0) = x(0, 0) C_3^{10} + x(1, 0) C_3^{11} + x(2, 0) C_3^{12}
\]

\[
T(1, 1) = x(0, 1) C_3^{10} + x(1, 1) C_3^{11} + x(2, 1) C_3^{12}
\]

\[
T(1, 2) = x(0, 2) C_3^{10} + x(1, 2) C_3^{11} + x(2, 2) C_3^{12}
\]

Similarly, substituting \( N = 3, k = 2 \) & \( n = 0, 1, 2 \) in (4), we get
Systolic Architecture for Realization of Two Dimensional Discrete Hartley Transform

\[ T(2,0) = x(0,0)C_{3}^{20} + x(1,0)C_{3}^{21} + x(2,0)C_{3}^{22} \]  
\[ T(2,1) = x(0,1)C_{3}^{20} + x(1,1)C_{3}^{21} + x(2,1)C_{3}^{22} \]  
\[ T(2,2) = x(0,2)C_{3}^{20} + x(1,2)C_{3}^{21} + x(2,2)C_{3}^{22} \]  
\[ T(0,0) = x(0,0)C_{3}^{00} + T(0,1)C_{3}^{01} + T(0,2)C_{3}^{02} \]  
\[ T(0,1) = T(0,0)C_{3}^{10} + T(0,1)C_{3}^{11} + T(0,2)C_{3}^{12} \]  
\[ T(0,2) = T(0,0)C_{3}^{20} + T(0,1)C_{3}^{21} + T(0,2)C_{3}^{22} \]  
\[ T(1,0) = T(1,0)C_{3}^{00} + T(1,1)C_{3}^{01} + T(1,2)C_{3}^{02} \]  
\[ T(1,1) = T(1,0)C_{3}^{10} + T(1,1)C_{3}^{11} + T(1,2)C_{3}^{12} \]  
\[ T(1,2) = T(1,0)C_{3}^{20} + T(1,1)C_{3}^{21} + T(1,2)C_{3}^{22} \]  
\[ H(0,0) = T(0,0)C_{3}^{00} + T(0,1)C_{3}^{01} + T(0,2)C_{3}^{02} \]  
\[ H(0,1) = T(0,0)C_{3}^{10} + T(0,1)C_{3}^{11} + T(0,2)C_{3}^{12} \]  
\[ H(0,2) = T(0,0)C_{3}^{20} + T(0,1)C_{3}^{21} + T(0,2)C_{3}^{22} \]  
\[ H(1,0) = T(1,0)C_{3}^{00} + T(1,1)C_{3}^{01} + T(1,2)C_{3}^{02} \]  
\[ H(1,1) = T(1,0)C_{3}^{10} + T(1,1)C_{3}^{11} + T(1,2)C_{3}^{12} \]  
\[ H(1,2) = T(1,0)C_{3}^{20} + T(1,1)C_{3}^{21} + T(1,2)C_{3}^{22} \]  
\[ H(2,0) = T(2,0)C_{3}^{00} + T(2,1)C_{3}^{01} + T(2,2)C_{3}^{02} \]  
\[ H(2,1) = T(2,0)C_{3}^{10} + T(2,1)C_{3}^{11} + T(2,2)C_{3}^{12} \]  
\[ H(2,2) = T(2,0)C_{3}^{20} + T(2,1)C_{3}^{21} + T(2,2)C_{3}^{22} \]  

The nine relations from (6) to (14) are realized using the systolic architecture shown in Figure 1. The symbol ‘•’ denotes a delay element. This architecture consists of \( N^2 = 9 \) identical PEs. Each PE consists of one multiplier, one adder and one register for storing \( c_{N}^{m} \). The function of each PE is shown in Figure 3. When the input data \( x(m,n) \) moves down the architecture, \( T(k,n) \) given by the relations from (6) to (14) will be generated as shown in Fig. 1.

3.2 Step 2:
Substituting \( l = 0,1,2 \) successively in (5), we get the following three expressions for \( N = 3 \) & \( k = 0 \).
\[ H(0,0) = T(0,0)C_{3}^{00} + T(0,1)C_{3}^{01} + T(0,2)C_{3}^{02} \]  
\[ H(0,1) = T(0,0)C_{3}^{10} + T(0,1)C_{3}^{11} + T(0,2)C_{3}^{12} \]  
\[ H(0,2) = T(0,0)C_{3}^{20} + T(0,1)C_{3}^{21} + T(0,2)C_{3}^{22} \]  
Substituting \( N = 3 \), \( k = 1 \) & \( l = 0,1,2 \) in (5), we obtain
\[ H(1,0) = T(1,0)C_{3}^{00} + T(1,1)C_{3}^{01} + T(1,2)C_{3}^{02} \]  
\[ H(1,1) = T(1,0)C_{3}^{10} + T(1,1)C_{3}^{11} + T(1,2)C_{3}^{12} \]  
\[ H(1,2) = T(1,0)C_{3}^{20} + T(1,1)C_{3}^{21} + T(1,2)C_{3}^{22} \]  
Similarly, substituting \( N = 3 \), \( k = 2 \) & \( l = 0,1,2 \) in (5), we have
\[ H(2,0) = T(2,0)C_{3}^{00} + T(2,1)C_{3}^{01} + T(2,2)C_{3}^{02} \]  
\[ H(2,1) = T(2,0)C_{3}^{10} + T(2,1)C_{3}^{11} + T(2,2)C_{3}^{12} \]  
\[ H(2,2) = T(2,0)C_{3}^{20} + T(2,1)C_{3}^{21} + T(2,2)C_{3}^{22} \]  
The 2D-DHT of size 3x3 is realized by computing the expressions (15) to (23) using the systolic architecture shown in Figure 2. The output data generated in Figure 1 enters the systolic architecture shown in Figure 2. The symbol ‘•’ denotes a delay element. This architecture consists of \( N^2 = 9 \) identical PEs. Each PE consists of one multiplier, one adder and one register for storing \( c_{N}^{m} \). The function of each PE is shown in Figure 3. When the data \( T(k,n) \) moves down the architecture, the output 2-D DHT components \( H(k,l) \) given by the expressions from (15) to (23) will be generated as shown in Fig. 2.
Systolic Architecture for Realization of Two Dimensional Discrete Hartley Transform

Fig. 2. Systolic architecture for computation of 2-D DHT, \( H(k, l) \), given by (5) for \( N = 3 \).

Table 1. Comparison of computation complexities

<table>
<thead>
<tr>
<th>Structure</th>
<th>Multipliers</th>
<th>Adders</th>
<th>Cycle-time (( T ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>( 2N^2 )</td>
<td>( 2N^2 )</td>
<td>( T_M + T_A )</td>
</tr>
<tr>
<td>2-D DFT [20]</td>
<td>( 4N^2 )</td>
<td>( 2N^2 )</td>
<td>( T_M + 2T_A )</td>
</tr>
<tr>
<td>2-D DFT [21]</td>
<td>( 2N(2N + 1) )</td>
<td>( 4N^2 + 6N - 4 )</td>
<td>( T_M + T_A )</td>
</tr>
</tbody>
</table>

IV. Conclusion

This paper presents two systolic architectures for realization of 2-D DHT of size \( N \times N \). These two systolic architectures can be orthogonally interspersed together for realization 2-D DHT without requiring any transposition buffer. The number of processing elements (PEs) used in the combined structure is \( 2N^2 \). Each PE consists of one multiplier, one adder and one register. So, \( 2N^2 \) multipliers and equal numbers of adders are used for realization of 2-D DHT. The computation complexities of this systolic architecture are compared with systolic architectures of 2-D DFT [20] and [21]. This comparison shows that the proposed architecture is better.
than those two systolic architectures using 2-D DFT. The systolic architecture utilizes parallel structures to achieve high speed performance. The highly regular structure of systolic circuits renders them comparatively easy to design and test. The systolic architecture is suitable for VLSI implementation.

References