

# Review Paper on Fast DHT Algorithm using Vedic Mathematics

Shirali Parsai  
M tech Scholar NIIST, Bhopal

Swapnil Jain  
Associate Professor NIIST,  
Bhopal

Jyoti Dangi  
Assistant Professor NIIST,  
Bhopal

## ABSTRACT

Discrete Hartley Transform (DHT) is one of the transform used for converting data in time domain into frequency domain using only real values. DHT can be used for highly modular and parallel processing of data in VLSI applications. We have proposed a new algorithm for calculating DHT of length  $2^N$ , where  $N=3$  and  $4$ . We have implemented multiplier as an improvement in place of simple multiplication used in conventional DHT. This paper gives a comparison between conventional DHT algorithm and proposed DHT algorithm in terms of delays and area.

## Keywords

Discrete Hartley Transform (DHT), Urdhwa Multiplier, and Xilinx Vertex family.

## 1. INTRODUCTION

Digital signal processing (DSP) includes processing of data in various domains based on their applications. DSP has vast applications in various fields such as space, medical, commercial, industrial and scientific. Each requires processing of vast data for collecting useful information [1]. Transform is a technique used in DSP for converting one form of data in another. A family of transform is available in DSP for data processing. Fourier analysis one of the oldest technique used in this family. Fourier analysis is named after Jean baptiste joseph fourier (1768-1830) a French mathematician and physicist. It was used for periodic continuous signals [2-3]. Fourier series is a technique which decomposes a signal in time domain into a no. of sine and cosine waves in frequency domain. But it was not applicable for non-periodic signals. Then came Fourier transform into existence which removes the drawback of fourier series and thus can be used for non-periodic continuous signals. Fourier transform is a mathematical tool using integrals [3]. But fourier transform is not suitable for non-stationary signals. Since both transforms are not applicable for discrete signals, so there is a need for new transform for discrete signals [4]. Discrete time fourier transform (DTFT) is used for signals that extend from positive to negative infinity but are not periodic. DTFT is not used for periodic discrete signals so discrete fourier transform (DFT) can into existence. DFT is a discrete numerical equivalent of FT using summation instead of integrals. DFT is used for signals that repeat themselves in periodic fashion extending from positive to negative infinity. FFT is an improvement of DFT in which computation has becomes faster [4].

All the family members of fourier till now works on complex values which requires large storage space and computationally complex in nature but Discrete Hartley transform (DHT) converts real values into real values. Therefore, it needs lesser storage space and less computational complexity [5].

The classical split-radix algorithm is difficult to implement on VLSI due to its irregular computational structure and due to the fact that the butterflies significantly differs from stage to stage. Thus, it is necessary to derive new such algorithms that are suited for a parallel VLSI system. In this brief, a new VLSI DHT algorithm that is well suited for a VLSI implementation on a highly parallel and modular architecture. It can be used for designing a completely novel VLSI architecture for DHT.

Discrete Hartley Transform is abbreviated for DHT and this transform was proposed by R. V. L. Hartley in 1942 [5]. DHT is the analogous to Fast Fourier transform which provides the only real value at any cost. The main difference from the DFT is that it transforms the real inputs to real outputs with no intrinsic involvement of complex value. DFT can be used to compute the DHT, and vice versa [6].

A Vedic multiplier provides lesser gates and enhanced speed for processor which can be designed by using half adder, full adder and novel compressor. Multipliers are Booth multiplier, modified booth multiplier, high speed multiplier and (4:2 and 7:2) compressor. Among them 7:2 compressor is much more high speed adder and lesser area multiplier designing technique.

## 2. LITERATURE REVIEW

H. M. de Oliveira et al. [1], discrete transforms such as the Discrete Fourier Transform (DFT) or the Discrete Hartley Transform (DHT) furnish an indispensable tool in Signal Processing. The successful application of transform techniques relies on the existence of the so-called fast transforms. In this paper some fast algorithms are derived which meet the lower bound on the multiplicative complexity of a DFT/DHT. The approach is based on the factorization of DHT matrices. New algorithms for short block lengths such as  $N = 3, 5, 6, 12$  and  $24$  are presented. In this paper we have seen DHT algorithms for  $N= 12$  that are used 52 adder and 4 multiplier.

Said Boussakta et al. [2], the discrete Hartley transform (DHT) has proved to be a valuable tool in digital signal/image processing and communications and has also attracted research interests in many multidimensional applications. Although many fast algorithms have been developed for the calculation of one- and two-dimensional (1-D and 2-D) DHT, the development of multidimensional algorithms in three and more dimensions is still unexplored and has not been given similar attention; hence, the multidimensional Hartley transform is usually calculated through the row-column approach. However, proper multidimensional algorithms can be more efficient than the row-column method and need to be developed. Therefore, it is the aim of this paper to introduce the concept and derivation of the three-dimensional (3-D) radix-2 2 2 algorithm for fast calculation of the 3-D discrete

Hartley transform. The proposed algorithm is based on the principles of the divide-and-conquer approach applied directly in 3-D. It has a simple butterfly structure and has been found to offer significant savings in arithmetic operations compared with the row-column approach based on similar algorithms. In this paper we have seen DHT algorithm for  $N=8$  that are used 39 adder and 24 multiplier.

Gautam Abhaychand Shah et al. [3], the radix-4 decimation-in-time fast Hartley transform and algorithm for DHT was introduced by Bracewell. A set of fast algorithms were further developed by Sorenson et al. In this paper, a fast radix-4 decimation-in-time algorithm that requires less number of multiplications and additions is proposed. It utilizes four different structures in the signal flow diagram. It exhibits a recursive pattern and is modular. The operational counts for the proposed algorithm are determined and verified by implementing the program in C. An analog architecture to implement the algorithm is proposed. The validity of the same is tested by simulating it with the help of the Orcad PSpice. In this paper we have seen DHT algorithm for  $N=4$  that are used 8 adder and 0 multiplier.

Doru Florin Chiper et al. [4], we present a new efficient method for the computation of the discrete Hartley transform of type II and radix-2 length. This recursive method requires a reduced number of arithmetic operations compared with existing methods and can be easily implemented. A new efficient method for the direct computation of a length  $N$  type-II DHT from two adjacent DHT-II sequences of length  $N/2$  is also presented. In this paper we have seen DHT algorithm for  $N=8$  that are used 28 adder and 10 multiplier.

M. N. Murty et al. [5], 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. In this paper we have seen DHT algorithm for  $N=4$  that are used 7 adder and 6 multiplier.

Doru Florin Chiper et al. [6], A new very large scale integration (VLSI) algorithm for a  $2N$ -length discrete Hartley transform (DHT) that can be efficiently implemented on a highly modular and parallel VLSI architecture having a regular structure is presented. The DHT algorithm can be efficiently split on several parallel parts that can be executed concurrently. Moreover, the proposed algorithm is well suited for the sub-expression sharing technique that can be used to significantly reduce the hardware complexity of the highly parallel VLSI implementation. Using the advantages of the proposed algorithm and the fact that we can efficiently share the multipliers with the same constant, the number of the multipliers has been significantly reduced such that the number of multipliers is very small comparing with that of the existing algorithms. Moreover, the multipliers with a constant can be efficiently implemented in VLSI. In this paper we have seen DHT algorithm for  $N=8$  that are used 16 adder and 2 multiplier.

### 3. DISCRETE HARTLEY TRANSFORM

Discrete Hartley transform is used to convert real values into real ones. It requires decomposition of data into stages using butterfly similar to FFT. But the butterfly used in DHT is quite different in terms of coefficients or multipliers. With the increase in number of DHT sequence length the number of coefficients is also increased simultaneously. We have proposed 16 point DHT butterfly with each data sequence is of 8 bit.

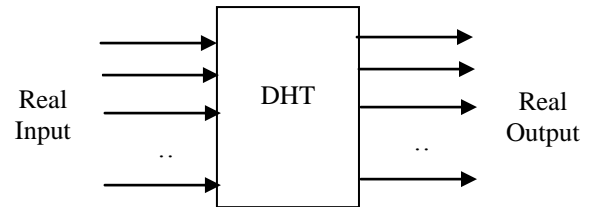


Figure 1: Block Diagram of DHT

### 4. ALGORITHM FOR 16 POINT DHT-

We present an implementation of fast DHT algorithm for a length  $N=16$ . There are six stages required to complete the butterfly design of  $N=16$  length DHT. These stages include summing stages and coefficient multiplying stages.

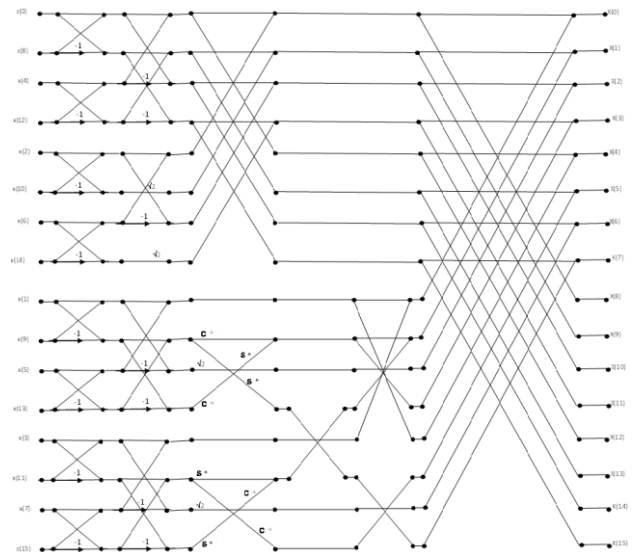


Figure 2: 16-point DHT Butterfly

Before first stage the data sequence are arranged in bit reversed pattern by using any method like permutation. Then in the first stage the pairs of bit reversed patterns are added to form eight terms. In the second stage, one third of the terms are again added and subtracted to form further three terms.

First two stages do not include any multiplication. Remaining terms are multiplied by the first coefficient. In the next stage again two new coefficients are introduced which is multiplied by the lower half of the third stage. In each stage multiplying of coefficients stage precedes its summing stage. After coefficient multiplication it is preceded by its summing stage to form the common terms used in the final stage. Last stage includes only summing of terms. Finally we get the

transformed data sequence in order and do not need any permutation.

- Mathematical calculation for N=16

$$\begin{aligned}
 X(0) &= x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(6) + x(7) \\
 &\quad + x(8) + x(9) + x(10) + x(11) + x(12) + x(13) + x(14) + x(15) \\
 X(1) &= x(0) + x(8) + x(4) - x(12) + c1\{x(2) - x(10)\} + c3\{x(7) - x(15) \\
 &\quad + x(3) - x(11)\} + c2\{x(3) - x(11) - x(7) + x(15)\} \\
 X(2) &= x(0) + x(8) - x(4) - x(12) - x(6) - x(14) + c1\{x(1) + x(9) \\
 &\quad - x(5) - x(13)\} + x(10) + x(11) + x(12) + x(13) + x(14) + x(15) \\
 X(3) &= x(0) - x(8) + x(12) - x(4) + c1\{x(6) - x(14)\} - c2\{x(3) \\
 &\quad - x(11) - x(7) + x(15)\} - c3\{x(7) - x(15) + x(3) + x(11)\}
 \end{aligned}$$

$$\begin{aligned}
 X(4) &= x(0) + x(8) + x(4) + x(12) + x(10) + x(6) + x(14) + x(1) \\
 &\quad + x(9) + x(5) + x(13) - x(3) - x(11) - x(7) - x(15) \\
 X(5) &= x(0) - x(8) + x(4) - x(12) - c1\{x(2) - x(10)\} + c3\{x(7) - x(15) + x(11) \\
 &\quad - x(3)\} + c3\{x(7) - x(15) + x(3) - x(11)\} + c3\{x(13) + x(9) - x(5) - x(1)\} \\
 &\quad + c2\{x(13) - x(5) + x(1) - x(9)\} \\
 X(6) &= x(0) + x(8) + x(6) + x(14) - x(2) - x(10) - x(4) - x(12) \\
 &\quad + c1\{x(3) + x(11) + x(7) - x(15)\} \\
 X(7) &= x(0) + x(8) - x(4) + x(12) + c1\{x(6) - x(14)\} - c2\{x(13) - x(5) \\
 &\quad + x(1) - x(9)\} - c3\{x(13) + x(9) - x(5) - x(1)\} \\
 X(8) &= x(0) - x(1) + x(2) - x(3) + x(4) - x(5) + x(6) - x(7) \\
 &\quad + x(8) - x(9) + x(10) - x(11) + x(12) - x(13) + x(14) - x(15) \\
 X(9) &= x(0) - x(8) + x(4) - x(12) + c1\{x(2) - x(10)\} - c2\{x(3) - x(11) \\
 &\quad - x(7) + x(15)\} - c3\{x(7) - x(15) + x(11) - x(3)\}
 \end{aligned}$$

$$\begin{aligned}
 X(10) &= x(0) + x(8) - x(4) - x(12) - x(6) - x(14) - c1\{x(1) + x(9) \\
 &\quad - x(5) - x(13)\} \\
 X(11) &= x(0) - x(8) + x(12) - x(4) + c1\{x(6) - x(14)\} + c2\{x(3) - \\
 &\quad x(11) - x(7) + x(15)\} + c3\{x(7) - x(15) + x(3) - x(11)\} \\
 X(12) &= x(0) + x(8) + x(12) + x(4) + x(2) + x(10) + x(6) + \\
 &\quad x(14) - x(1) - x(9) - x(5) - x(13) + x(3) + x(11) + x(7) + x(15) \\
 X(13) &= x(0) - x(8) + x(4) - x(12) - c1\{x(2) - x(10)\} - c3\{x(7) - x(15) + \\
 &\quad x(11) - x(3)\} - c3\{x(7) - x(15) + x(3) - x(11)\} - c2\{x(13) - x(5) + x(1) - x(9)\} \\
 &\quad + c3\{x(13) + x(9) - x(5) - x(1)\} \\
 X(14) &= x(0) + x(8) - x(2) - x(10) + x(6) + x(14) - x(4) - x(12) \\
 &\quad - c1\{x(3) + x(11) - x(7) - x(15)\} \\
 X(15) &= x(0) + x(8) - x(4) + x(12) - c1\{x(6) - x(14)\} + c2\{x(13) - x(5) + x(1) \\
 &\quad - x(9)\} + c3\{x(13) + x(9) - x(5) - x(1)\}
 \end{aligned}$$

## 5. COMPARATIVE ANALYSIS

Here, the DHT of length N=16 point requires 67 additions and 12 multiplications. All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx tool updated version.

It is also a well-known fact that the multiplier unit forms an integral part of processor design. Due to this regard, high speed multiplier architectures become the need of the day. In this future scope, we introduce a novel architecture to perform high speed multiplication using ancient Vedic mathematics techniques. A new high speed approach utilizing 4:2 compressors and novel 7:2 compressors for addition has also been incorporated in the same and has been explored.

**Table 1: Comparisons the multipliers and adder for different number of order**

Design	Number of order	Addition	Multiplier
H. M. de Oliveira et al. [1]	12	52	4
	24	138	12
Said Boussakta et al. [2]	8	39	24
	16	78	48
Gautam Shah et al. [3]	4	8	0
	16	68	12
Doru Florin Chipier et al. [4]	8	28	10
	16	92	32
M. N. Murty et al. [5]	4	7	6
	6	15	12
Doru Florin Chipier et al. [6]	8	16	2
	16	67	12

## 6. CONCLUSION

DHT is a new transform used for real value to real value conversion. Urdhwa Triyambakam is an ancient technique for multiplication. DHT is used in various fields such as image processing, space science, scientific applications etc. Delay provided and area required by hardware are the two key factors which are need to be consider. Here we present DHT with 8x8 urdhwa multiplier by using full adder, OR and XNOR gates in different types of compressors.

DHT algorithm that is well suited for a VLSI implementation on a highly parallel and modular architecture. It can be used for designing a completely novel VLSI architecture for DHT.

## 7. REFERENCES

- [1] H. M. de Oliveir a, R. J. de Sobr al Cintr a, R. M. Camp el lo de Souza, "A F A CTORIZA TION SCHEME FOR SOME DISCRETE HAR TLEY TRANSFORM MATRICES", VOL. 60, NO. 5, MAY 2013.
- [2] Said Boussakta, *Member, IEEE*, Osama Hamoud Alshibami, *Student Member, IEEE*, and Mohammed Yunis Aziz, *Student Member, IEEE*, "Radix-2 2 2 Algorithm for the 3-D Discrete Hartley Transform", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 49, NO. 12, DECEMBER 2001.

- [3] Gautam Abhaychand Shah, Tejmal Saubhagyamal Rathore, "A Fast Radix-4 Algorithm and Architecture for DHT", Vol-2 No 5 October, 2011.
- [4] Doru Florin Chiper, Senior Member, IEEE, "Fast Radix-2 Algorithm for the Discrete Hartley Transform of Type II", IEEE SIGNAL PROCESSING LETTERS, VOL. 18, NO. 11, NOVEMBER 2011.
- [5] M. N. Murty, "NOVEL RECURSIVE ALGORITHM FOR REALIZATION OF ONE-DIMENSIONAL DISCRETE HARTLEY TRANSFORM", IJRRAS 10 (2) • February 2012.
- [6] Doru Florin Chiper, *Senior Member, IEEE*, "A Novel VLSI DHT Algorithm for a Highly Modular and Parallel Architecture", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 60, NO. 5, MAY 2013.
- [7] Sushma R. Huddar and Sudhir Rao, Kalpana M., "Novel High Speed Vedic Mathematics Multiplier using Compressors", 978-1 - 4673-5090-7/13/\$31.00 ©2013 IEEE.
- [8] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A, "Implementation of Vedic multiplier for Digital Signal Processing", International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011, Proceedings published by International Journal of Computer Applications® (IJCA), pp.1 -6
- [9] Himanshu Thapaliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad, India.
- [10] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda", Delhi (2011).