Journal of Basic and Applied Engineering Research
p-ISSN: 2350-0077; e-ISSN: 2350-0255; Volume 5, Issue 7; October-December, 2018, pp. 500-502

http://www.krishisanskriti.org/Publication.html

Implementation of DHT Algorithm for a VLSI Architecture

Deepti Gautam[1] and Anshuman Singh[2]

[1]Noida Institute of Engg. & Technology Greater Noida, U.P., India
[2]Noida Institute of Engg. & Technology Greater Noida, U.P., India
E-mail: [1]deepti.gautam44@gmail.com, [2]anshumansingh1317@gmail.com

Abstract—*This paper presents an algorithm and its implementation which is well suited for VLSI architecture. The objective of using DHT i.e. Discrete Hartley Transform Algorithm is to reduce complexity of VLSI design. With the advent of VLSI designs, 100,000 transistors can be implemented on a single chip. For the designers, the verification of these circuits was not at all possible on breadboards. Computer aided design techniques were also came into existence but again the designers had to do connections manually on gate level. Then Hardware Description Language (HDL) came into existence. Hardware description Languages such as Verilog and VHDL came into existence. With integration of DHT algorithm and its implementation using Verilog, we hereby try to reduce complexity of the VLSI circuits. With reduced complexity, ultimately the power consumption gets reduced and hence area also gets reduced. DHT is a radix-2 algorithm. It is mainly useful in image and signal processing. Here, we present a DHT algorithm for length N=8.*

Keywords*: Discrete Hartley Transfrom (DHT), DHT domain processing, Discrete Fourier Transform (DFT).*

1. INTRODUCTION

Discrete Hartley Transform is a radix-2 algorithm. It is mainly useful in DSP such as image compression, signal compression, filter banks [2], signal representations or harmonic analysis [3]. The main objective behind using DHT is to reduce complexity of VLSI hardware architecture [1]. Though the computations with the DHT become intensive but hardware complexity gets reduced. The reason behind is that the forward and inverse of DHT is absolutely same except a scaling factor. Or DHT is its own inverse. Therefore, the same hardware can be used for both forward and inverse calculations. In other words, we can say that same number of adders, multipliers etc is used in both forward and inverse implementations.

DHT is very similar to Discrete Fourier Transform (DFT). Or DHT is obtained from DFT. The DFT contains both real and imaginary parts while DHT has only real parts. The calculation of imaginary parts is negligible and therefore complex arithmetic is avoided. In other words, DHT is used as a replacement of DFT when sequence is real [4], [5].

There are many other ways available with which VLSI implementations can be performed one of them are systolic arrays [7]-[9]. But difficultly in using them is that they use pipelining in particular and not the parallel processing, hence high-speed processing cannot be achieved.

There are also many spilt radix algorithm available for computing. And these can be implemented with even low arithmetic cost [10]-[11]. The disadvantage with classical spilt-radix algorithm is its irregular computational structure.

Delays in VLSI architecture are mainly introduced by multipliers, as they consume large portion of chip area. Hence, to implement multipliers memory based solutions are being taken into consideration. To implement multipliers with look-up table based solutions, it is necessary that one operand to be constant [12]. It is then becomes possible to store all the results in partial ROM when one of the operand is constant. And the memory words are reduced from 2^{2L} to 2^{L} [1].

With DHT algorithm, we can implement VLSI architecture which will be based on high parallelism. Moreover, by sharing the multipliers by same constant and using sub-expression sharing technique, hardware complexity gets reduced [13]-[14]

1.1 DHT Algorithm

1.1.1 Forward and Inverse DHT

Let N ≥ 4 and N be a power of 2, since it's a radix-2 algorithm. For any real input sequence such as

{ x(i) } i = 0,1,2…..N - 1 }

DHT (N) is defined by

X (k) = DHT (N) { x(i) }

$$= \sum_{i=0}^{N-1} x(i) \cdot \cos\left[\frac{2ki\pi}{N}\right] + \sin[\frac{2ki\pi}{N}]) \quad (1)$$

for k = 0,1,2,3,4……..N-1

Inverse relation is defined as:

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

for n=0,1……N-1

Also, cas(x) = cos(x) + sin(x)

1.1.2 DHT algorithm for length L=8

Here, we present algorithm for length 8.

X (0) = [(x (0) + x (4)) + x(2) + x(6))] + [(x (1) + x (5)) + x(3) + x(7))]

X (2) = [(x (0) + x (4)) - x(2) + x(6))] + [(x (1) + x (5)) - x(3) + x(7))]

X (4) = [(x (0) + x (4)) + x(2) + x(6))] - [(x (1) + x (5)) + x(3) + x(7))]

X (6) = [(x (0) + x (4)) - x(2) + x(6))] - [(x (1) + x (5)) - x(3) + x(7))]

X(1) = [x (0) - x (4)] + [x(2) - x(6)] + c [(x (1) - x (5)]

X(3) = [x (0) - x (4)] - [x(2) - x(6)] + c [(x (3) - x (7)]

X(5) = [x (0) - x (4)] + [x(2) - x(6)] - c [(x (1) - x (5)]

X(7) = [x (0) - x (4)] - [x(2) - x(6)] - c [(x (3) - x (7)]

With c =√2

We can further reduce the number of multipliers since we are multiplying with the same constant ‘c’. For the same purpose we can use same multiplier.

2. HARDWARE FLOW DIAGRAM

The above equations from (3) – (10) can be implemented in hardware by considering following flow diagram. The block simply consists of few ADDERs’ and two MULTIPLIER blocks for multiplication with ‘c’.

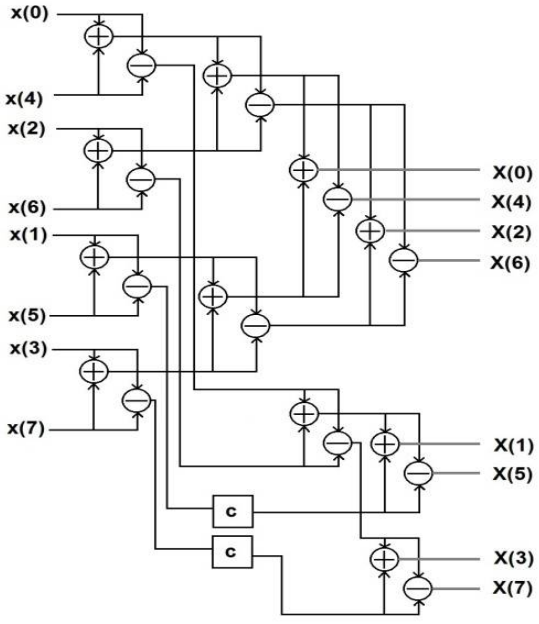

Figure 1: Hardware Flow Diagram

3. RTL FLOW DIAGRAM

The fig. below shows the synthesis of the block diagram in form of an RTL, synthesized using XILINX ISE 14.7. All the blocks are simple adders and subtractors.

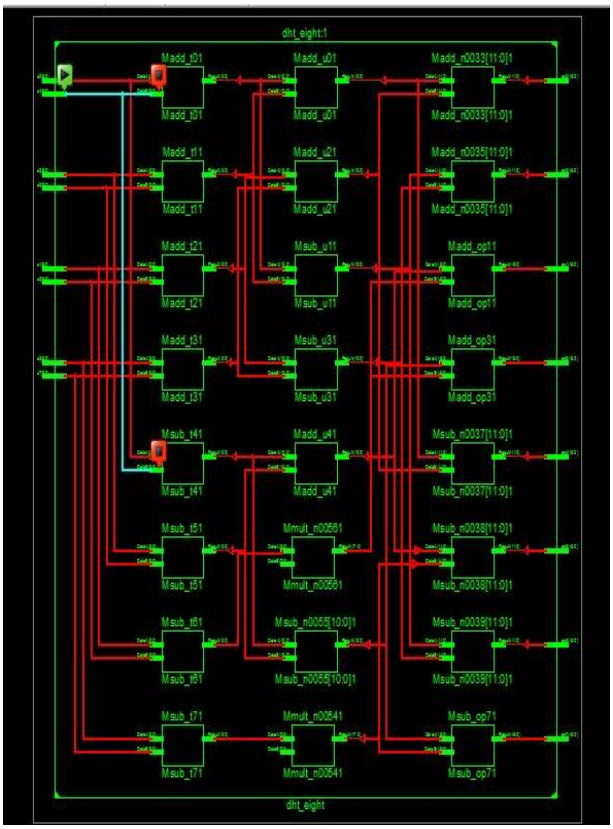

Figure 2 RTL Flow Diagram

The above RTL can also be represented as:

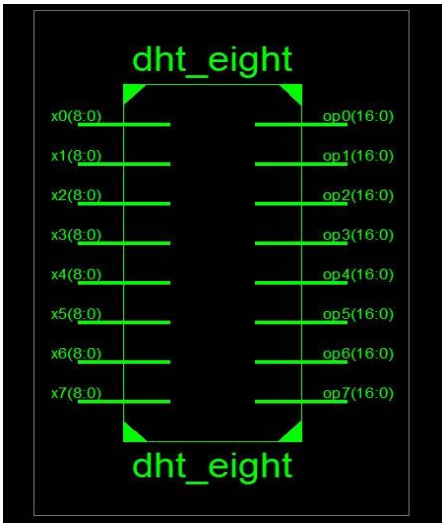

Figure 3 RTL Block Diagram

4. COMPARISON WITH OTHER ALGORITHMS

Table 1

	Radix 2 [13]		Radix 2 [13]*		Radix 2 [11]**		DHT	
N	M	A	M	A	M	A	M	A
8	-	-	-	-	4	26	2	16

M-Multiplier A-Adder

With reference to above table it is very clear that architectures based on DHT algorithm uses less number of multipliers and adders. Hence, the complexity of VLSI architecture will be reduced.

5. SIMULATION & RESULTS

Figure 4

Figure 5

6. CONCLUSION

In this paper, a DHT algorithm for highly parallel and regular VLSI structure is presented for the length N=8. The application of this algorithm leads to less hardware complexity and hence power consumption also gets reduced which remains a major concern while dealing with VLSI circuits. DHT algorithm implements sub-sharing expression technique and sharing of multipliers which are having same constants. DHT is most useful to use in signal processing and image compression applications.

REFERENCES

[1] Doru Florin Chiper, *Senior Member, IEEE* "A novel VLSI DHT algorithm for a highly parallel and modular architecture" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 60, Issue 5, MARCH 2013

[2] Doru Florin Chiper, *Senior Member, IEEE* "Radix-2 Fast Algorithm for Computing Discrete Hartley Transform of Type III" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 59, NO. 5, MAY 2012

[3] Michela Svaluto Moreolo, Raül Muñoz, and Gabriel Junyent "Novel Power Efficient Optical OFDM Based on Hartley Transform for Intensity-Modulated Direct-Detection Systems "JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 28, NO. 5, MARCH 1, 2010

[4] Chin-Kuo Jao, Syu-Siang Long, and Muh-Tian Shiue, *Member, IEEE* "DHT-Based OFDM System for Passband Transmission Over Frequency-Selective Channel "IEEE SIGNAL PROCESSING LETTERS, VOL. 17, NO. 8, AUGUST 2010

[5] Saad Bouguezel, M. Omair Ahmad, *Fellow, IEEE*, and M. N. S. Swamy, *Fellow, IEEE* "New Parametric Discrete Fourier and Hartley Transforms, and Algorithms for Fast Computation "IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 58, NO. 3, MARCH 2011

[6] Liang Tao and Hon Keung Kwan, *Senior Member, IEEE* "Multirate-Based Fast Parallel Algorithms for 2-D DHT-Based Real-Valued Discrete Gabor Transform "IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 21, NO. 7, JULY 2012

[7] Pramod Kumar Meher, *Senior Member, IEEE*, Thambipillai Srikanthan, *Senior Member, IEEE*, and Jagdish C. Patra, *Member, IEEE* "Scalable and Modular Memory-Based Systolic Architectures for Discrete Hartley Transform "

[8] IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 53, NO. 5, MAY 2006

[9] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[10] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[11] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[12] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[13] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[14] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.