Design and Simulation of 1D DHT for communication based applications

Rajni Jounjare#1, Poonam Sinha*2, R. P. Singh#3

1, 2,3Department of Electronics and communication, Barkatullah University, Bhopal, India

3Vice Chancellor, SSSVTMS, Sehore, Bhopal, India

1jounjarerajni@gmail.com

2poonamuit@yahoo.com

3prof.rpsingh@gmail.com

Abstract— **The data in time domain can be converted or transformed into frequency domain with the help of one of the transforming technique like Discrete Hartley Transform using real values. For highly modular and parallel processing of data applications in VLSI design, DHT can be used. We have proposed a new algorithm for calculating DHT of length 2^N, where N=3 and 4. As an [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.**

I. Introduction

The data processing of various domains based on their applications are carried out by digital signal processing (DSP). It has a large number of applications as space, medical, commercial, industrial and scientific research. For collecting useful information for a particular application requires processing of vast data. For converting one form of data in another the technique used is called transform technique used in DSP. To process data DSP has a variety of transform technique based on their application. One of the oldest techniques used in this family is Fourier analysis. Fourier analysis is named after a French mathematician and physicist, Jean Baptiste Joseph Fourier (1768-1830). This technique is used for periodic continuous signals. A signal in time domain can be transformed or decomposed into a no. of sine and cosine waves in frequency domain by the use of fourier series technique. But in case of non-periodic signals fourier series is not applicable and to remove this drawback Fourier transform came in picture as a result thereafter non-periodic continuous signals can be processed. This transform technique is a mathematical tool using integrals. In case of non-stationary signals Fourier transform is not a better choice to work on. As both the transform techniques are not valid for discrete signals hence there is a need for new transform technique that can be used for discrete signals. For signals that extend from positive to negative infinity but are not periodic discrete time Fourier transform (DTFT) is used for such signals. DTFT is not used for periodic discrete signals. DFT is a discrete numerical equivalent of FT using summation instead of integrals. Signals that repeat themselves in periodic fashion extending from positive to negative infinity DFT is used. FFT is a enhancement of DFT in which computation has becomes faster. The drawback of all the family member of Fourier is that they works on complex values which requires large storage space and computationally complex in nature, hence there came a new member of transform called Discrete Hartley transform (DHT) which converts real input values into real output values. Therefore, it needs lesser storage space and less computational complexity. In this paper DHT is computed using simple addition and multiplication, which is done by using Urdhwa multiplier. Urdhwa multiplier uses different 4:2 and 7:2 compressors along with full adder and half adder for complete multiplication process.

II. Literature Review

In Reference [1] R.N.Bracewell et.al. proposes a fast Hartley transform design. A fast algorithm has been designed for performing the Discrete Hartley Transform (DHT) for a data sequence of N elements in a time proportional to N log 2 N. In Reference [2] H.V.Sorenson et.al. focused on a complete set of fast algorithms for computing the DHT, including decimation-in-frequency, radix-4, split radix, prime factor, and Winograd transform algorithm techniques. In Reference [3] Chakrabarti C..et. al. proposes prime factor mapping for the discrete Hartley transform. Their work proposes a improved method of Lun and Sin for computing prime factor based decomposed discrete Hartley transform over N points. The author shows that the algorithm can be extended to eliminate the 2N extra additions in the algorithms. In Reference [4] R I Hartley et.al. a paper that examined methods for optimizing the design of CSD multipliers, and in particular the gains that can be made by sharing sub-expressions.

In Reference [5] Sung bum pan_et.al. proposed a paper that shows a general unified systolic array designed for different discrete transform for fast computational process. The proposed unified systolic array architecture can compute the DCT, the DST, and the DHT by defining different coefficient values specific for each transform. In Reference [6] Lizhi Cheng et.al. proposed design for circular de-convolution by implementing a Fast Hartley transform and truncated singular value algorithm. The work investigates for reducing no. of arithmetic operations that required for computing FFT de-convolution algorithm. It is implemented for computing DHT circular de-convolution. In Reference [7] Guoan Bi et.al. published their work for the discrete Hartley transform based on New split-radix algorithm. This paper proposes a design for the discrete Hartley transforms that can flexibly compute for various sequence lengths by implementing split-radix algorithm. It showed that the length-$3*2^m$ DHTs need a smaller number of multiplications than the length-2^m DHTs.

In Reference [8] S C Pei et.al. proposes a paper on Discrete fractional Hartley and Fourier transforms that shows the relation between the definitions of the discrete fractional Hartley transform (DFRHT) and the discrete fractional fourier transform (DFRFT). In order to remove chirp interference a filtering technique in the fractional Fourier transform domain is used. In Reference [9] 2000, P. Hao et.al. presented a paper that shows a comparative study of colour transforms for image coding and derivation of integer reversible colour transform. In this paper, a comparative study is carried out of colour transforms for colour image coding in order to find the best one among 11 published colour transforms: YCrCb, NTSC, PAL, HDTV, WW, XYZ, DCT, DHT, two approximate K-L transforms (K1K2K3 and KLT) and the original reversible colour transform (ORCT) adopted in JPEG-2000.

In Reference [10] 2000, Guoan Bi et.al. proposed a paper that presented fast algorithms for type-II, type-III, and type-IV generalized discrete Hartley transform. This paper shows an approach for sequence length that contains multiple odd factors, a new odd-factor algorithms are derived to support the transform technique. In Reference [11] 2002, C. C. Tseng et.al. proposes work on Eigen values and eigen vectors of generalized DHT, DCT-IV and DST-IV matrices. In this paper, the eigen values and eigen vectors of the generalized discrete Fourier transform (GDFT), the generalized discrete Hartley transform (GDHT), the type-IV discrete cosine transform (DCT-IV), and the type-IV discrete sine transform (DST-IV) matrices are investigated in a unified framework. In Reference [12] 2003, P K Meher et.al. published a paper that focused on Near lossless image compression using lossless hartley like transform. It showed that to perform an integer-to-integer transformation the discrete Hartley transform (DHT) of length N = 4 can be used.

In Reference [13] 2003, German Rodriguez et. al. Proposed a paper on simulation and analysis of wave records through Fast hartley transform. For the analysing and synthesis of ocean surface wave records the hartley transform is used as an efficient tool also it is a real-valued alternative to the complex Fourier transform. In Reference [14] 2004, S Bouguezel et.al. proposed a new split-radix FHT algorithm for length-$q*2^m$DHTs. In proposed work for computing the discrete Hartley transform (DHT) of an arbitrary length $N=q*2^m$, where q is an odd integer, a new split-radix fast Hartley transform (FHT) algorithm is proposed for computing the discrete Hartley transform (DHT). In Reference [15] 2004, C C Tseng et.al. proposed a Quantum circuit design of 8×8 discrete Hartley transform. In this paper, the 8 × 8 discrete Hartley transform is implemented by using quantum elementary gates such as controlled NOT gates and Hadamard gates.

In Reference [16] 2011, S Bouguezel et.al. proposed a New Parametric Discrete Fourier and Hartley Transforms, and algorithms for Fast Computation. In this paper the author proposed a new reciprocal-orthogonal parametric discrete Fourier transform (DFT) by appropriately replacing some specific twiddle factors in the kernel of the classical DFT by independent parameters that can be chosen arbitrarily from the complex plane. In Reference [17] 2011, H.B. Kekre et.al. published their work on Performance Enhancement of Fractional Coefficients of the Discrete Hartley Transform for Palm Print Recognition. Their paper provides a solution to this problem based on an analysis of the transformed image obtained by using the Discrete Hartley Transform.

In Reference [18] 2011, D F Chiper et.al. proposed a design for the Discrete Hartley Transform of Type II by implementing the Fast Radix-2 Algorithm. The author designed a new efficient method for the computation of the discrete Hartley transform of type II and radix-2 length N =2 n.

In Reference [19] 2011, Gautam Abhyachand Shah et.al. proposed a fast radix-4 decimation-in time algorithm that requires less number of multiplications and additions. In the signal flow diagram it utilizes four different structures. It is modular and exhibits a recursive pattern. In Reference [20] 2011, V. Baby Deepa et.al. proposes their research work on the classification filters such as Fast Hartley Transform (FHT) and Chebyshev filters. These filters are used to classify the EEG data signals.

III. URDHWA MULTIPLIER

In DSP processors multipliers are the blocks which requires large amount of storage and processing complexity. An ancient method in Vedic mathematics for mathematical operations is carried out by well known urdhwa triyambakam method which is used for multiplication calculations. It takes the advantage of vertical and crosswise calculations. Simple AND logical operations, full adders and half adders are used for obtaining multiplication of terms. In this paper 8 bit urdhwa multiplier is used. Various designs of adders and compressors are used for the purpose of addition. Here as to provide optimized design of multiplier in terms of delay these compressors and adders are deployed in such a way to achieve the desired goal. Four half adders, two full adders, five 7: 2 compressor and ten 4:2 compressors has been used to realize the function.

For realizing urdhwa multiplier we have proposed three designs. These three designs vary in terms of 4:2 compressor designs. First design uses two full adders for design of the simple 4:2 compressor. Second design uses two 2X1 multiplexers and four XOR gates. Third design of 4:2 compressor two XOR-XNOR gates and four 2X1 multiplexers are used. First design of 4:2 compressor has more delay, but it is simplest among all designs. As compared to previous design next design has lesser delays but at the cost of higher complexity. In our proposed design we have overcome the problem of delays as it has least delays among the others when compared but it occupies maximum area space and also has greatest complexity. We have designed 7:2 compressor using different designs of 4:2 compressor thus we got three variations of 7:2 compressor.

A. 4:2 Compressor

While adding binary numbers having minimal carry propagation we choose compressor adder in-place of other adders. Digital modern circuit designing uses compressor because this has high speed with minimal requirement of gates. This compressor technique becomes an important tool for fast multiplication and adding technique because of its fast processing speed and occupying lesser area.

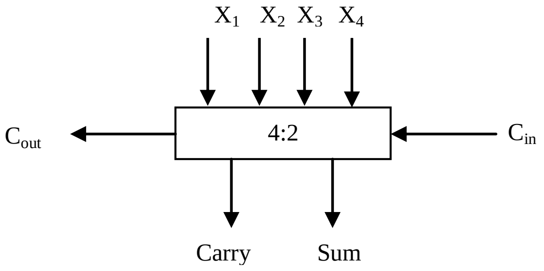


Fig. 1 Block Diagram of 4:2 Compressors based Adder

4:2 compressors gives a 3 bit output and are capable of adding 4 bits and one carry. The 4-2 compressor has 4 inputs X_1, X_2, X_3 and X_4 and 2 outputs Sum and Carry along with a Carry-in (Cin) and a Carry-out (Cout) as shown in fig. 1. The input Cin is termed as the output from the previous lower significant compressor. For the circuit in the next significant stage Cout is the output of the compressor. When compared with a conventional circuit to add 5 bits using full adders and half adders the proposed technique has lower critical path. Similar to the 3:2 compressor, the 4:2 compressor is also governed by the basic equations.

$$X_1+X_2+X_3+X_4+C_{in} = sum + 2*(carry + C_{out}) \qquad (1)$$

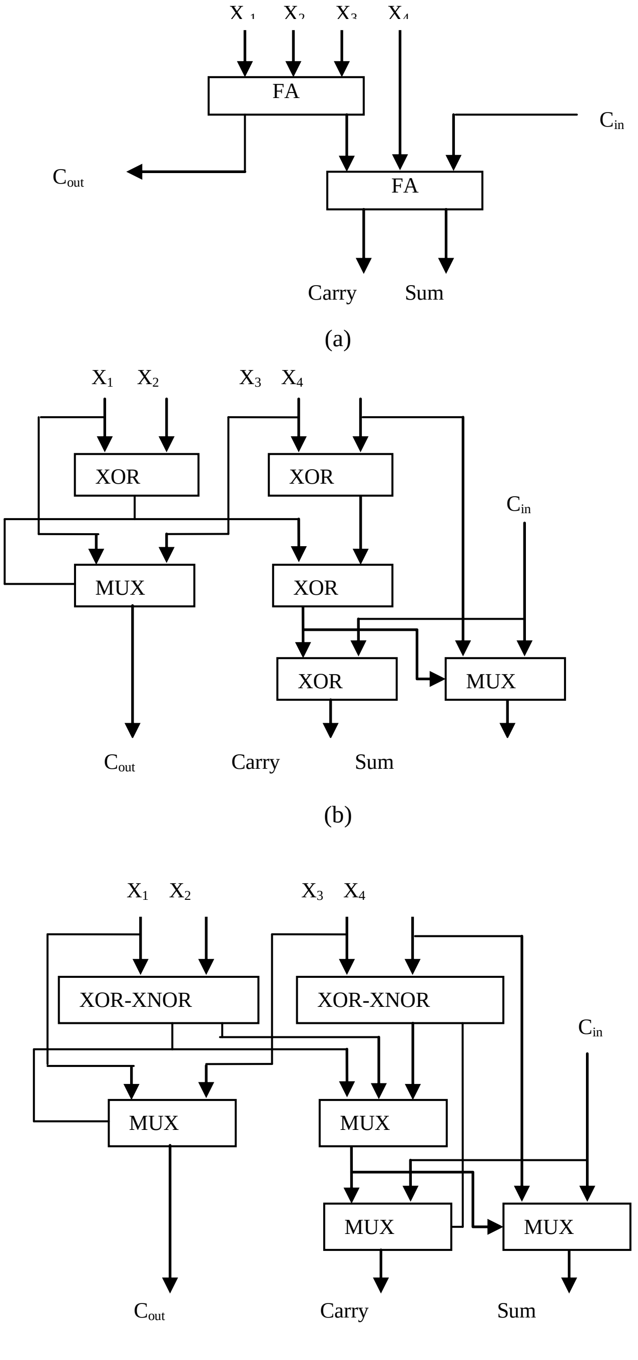
The standard implementation of the 4-2 compressor is done by using 2 Full Adder cells as shown in fig. 2(a). We observed that the overall delay is equal to 4*XOR when the individual full Adders are broken into their constituent XOR blocks. The block diagram in fig. 2(b) shows the existing architecture for the implementation of the 3:2 compressor with a delay of 4*XOR. The equations governing the outputs in the existing architecture are shown below.

$$Sum = X_1 \text{ xor } X_2 \text{ xor } X_3 \text{ xor } X_4\ C_{in} \qquad (2)$$

$$Cout = ((X_1 \text{ xor } X_2) \text{ and } X_3) \text{ or } (\text{not } (X_1 \text{ or } X_2) \text{ and } X_4) \qquad (3)$$

$$Carry = (\ (X_1 \text{ xor } X_2 \text{ xor } X_3 \text{ xor } X_4) \text{ and } C_{in}\ ) \text{ or } (\text{not } (X_1 \text{ xor } X_2 \text{ xor } X_3 \text{ xor } X_4) \text{ and } X_4) \qquad (4)$$

However, like in the case of 3:2 compressors, the fact that both the output and its complement are available at every stage is neglected. In order to achieve a significant improvement in delay we need to replace some XOR blocks with multiplexers.



(a)

(b)

(c)

Fig. 2 Design of 4:2 compressor using (a) Full Adder, (b) XOR and Multiplexer, (c) 4:2 Compressor

Also the MUX block at the SUM output gets the select bit before the inputs arrive and thus the transistors are already switched by the time they arrive. This minimizes the delay to a considerable extent. This is shown in fig. 2(c).

B. 7:2 Compressor

7:2 compressor is capable of adding 7 bits of input and 2 carry's from the previous stages, at a time is similar to its 4:2 compressor counterpart, as shown in fig. 3. In our implementation, we have designed a novel 7:2 compressor utilizing two 4:2 compressors, two full adders and one half adders. We have designed 7:2 compressor using different designs of 4:2 compressor thus we got three variations of 7:2 compressor.

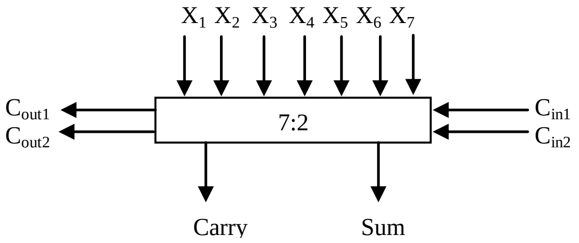


Fig. 3 Block Diagram of 7:2 Compressors

IV. Discrete Hartley Transform Algorithm Design

Discrete Hartley transform is used to convert real input values into real output ones. It decomposes the data similar to FFT using butterfly technique, but the butterfly used in DHT is quite different in terms of coefficients or multipliers. The number of coefficients is also increased with the increase in number of DHT sequence length. We have proposed 16 point DHT butterfly which has a data sequence of 8 bit.

A. Algorithm For 16 Point DHT

We present an implementation of fast DHT algorithm which has six stages required to complete the butterfly design of N=16 length DHT. These stages include summing stages and coefficient multiplying stages. The data sequences are arranged in bit reversed pattern by using any of the method like permutation is done before first stage and then the pairs of bit reversed patterns are added to form eight terms in the first stage. In the second stage, one third of the terms are again added and subtracted to form further three terms. This is shown in fig. 4.

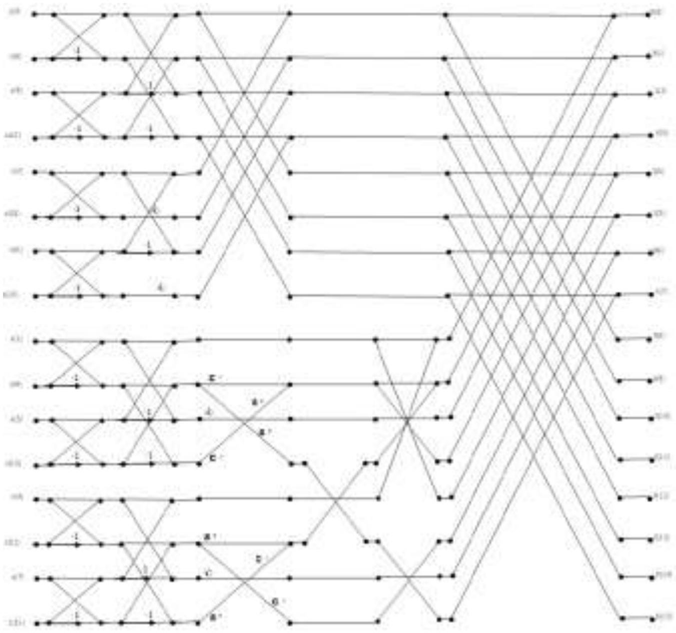


Fig. 4 16-point DHT Butterfly

First two stages do not include any multiplication. Remaining terms are multiplied by the first coefficient. Two new coefficients are introduced in the next stage again which is multiplied by the lower half of the third stage. In each stage summing is done after the multiplication of coefficient stage. After coefficient multiplication it is preceded by its summing stage to form the common terms used in the final stage. In the last we perform only summation of terms. Finally we get the transformed data sequence in order and do not need any permutation.

B. *Mathematical Calculation For N=16*

$$X(0) = x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(6) + x(7) + x(8) + x(9) + x(10) + x(11) + x(12) + x(13) + x(14) + x(15)$$

$$X(1) = x(0) + x(8) + x(4) - x(12) + c1\{x(2) - x(10)\} + c3\{x(7) - x(15) + x(3) - x(11)\} + c2\{x(3) - x(11) - x(7) + x(15)\}$$

$$X(2) = x(0) + x(8) - x(4) - x(12) - x(6) - x(14) + c1\{x(1) + x(9) - x(5) - x(13)\} + x(10) + x(11) + x(12) + x(13) + x(14) + x(15)$$

$$X(3) = x(0) - x(8) + x(12) - x(4) + c1\{x(6) - x(14)\} - c2\{x(3) - x(11) - x(7) + x(15)\} - c3\{x(7) - x(15) + x(3) + x(11)\}$$

$$X(4) = x(0) + x(8) + x(4) + x(12) + x(10) + x(6) + x(14) + x(1) + x(9) + x(5) + x(13) - x(3) - x(11) - x(7) - x(15)$$

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

$$X(6) = x(0) + x(8) + x(6) + x(14) - x(2) - x(10) - x(4) - x(12) + c1\{x(3) + x(11) + x(7) - x(15)\}$$

$$X(7) = x(0) + x(8) - x(4) + x(12) + c1\{x(6) - x(14)\} - c2\{x(13) - x(5) + x(1) - x(9)\} - c3\{x(13) + x(9) - x(5) - x(1)\}$$

$$X(8) = x(0) - x(1) + x(2) - x(3) + x(4) - x(5) + x(6) - x(7) + x(8) - x(9) + x(10) - x(11) + x(12) - x(13) + x(14) - x(15)$$

$$X(9) = x(0) - x(8) + x(4) - x(12) + c1\{x(2) - x(10)\} - c2\{x(3) - x(11) - x(7) + x(15)\} - c3\{x(7) - x(15) + x(11) - x(3)\}$$

$$X(10) = x(0) + x(8) - x(4) - x(12) - x(6) - x(14) - c1\{x(1) + x(9) - x(5) - x(13)\}$$

$$X(11) = x(0) - x(8) + x(12) - x(4) + c1\{x(6) - x(14)\} + c2\{x(3) - x(11) - x(7) + x(15)\} + c3\{x(7) - x(15) + x(3) - x(11)\}$$

$$X(12) = x(0) + x(8) + x(12) + x(4) + x(2) + x(10) + x(6) + x(14) - x(1) - x(9) - x(5) - x(13) + x(3) + x(11) + x(7) + x(15)$$

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

$$X(14) = x(0) + x(8) - x(2) - x(10) + x(6) + x(14) - x(4) - x(12) - c1\{x(3) + x(11) - x(7) - x(15)\}$$

$$X(15) = x(0) + x(8) - x(4) + x(12) - c1\{x(6) - x(14)\} + c2\{x(13) - x(5) + x(1) - x(9)\} + c3\{x(13) + x(9) - x(5) - x(1)\}$$

Here, the DHT of length N=16 point requires 67 additions and 12 multiplications. We have used a different technique called urdhwa tiryakbhyam of ancient Vedic times for multiplication. This multiplication technique reduces the complexity and delay. By implementing associated circuits of full adders, half adders and AND gate, this helps in converting multiplication into simple logical AND operation. Further, the area is reduced by using various compressors for adding the partial products of multiplication.

V. Simulation Results

There are commonly used three architectures of DHT transform. First architecture is designed using simple urdhwa multiplier. In simple urdhwa multiplier, we have used 4:2 compressor designed by using full adder and half adder i.e. the basic design of 4:2 compressor. Second architecture is designed using modified urdhwa multiplier. In modified urdhwa multiplier, we have used 4:2 compressor designed by using XOR gates and multiplexer. Third architecture is designed using modified urdhwa multiplier. In modified urdhwa multiplier, we have used 4:2 compressor designed by using XNOR gates and multiplexer. In this work the simulation RTL of the 16-bit DHT is shown in fig. 5 and fig. 6.

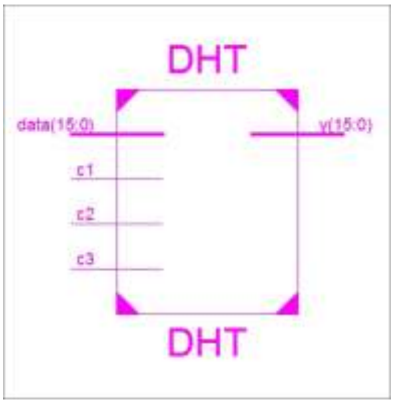


Fig. 5 RTL view of DHT

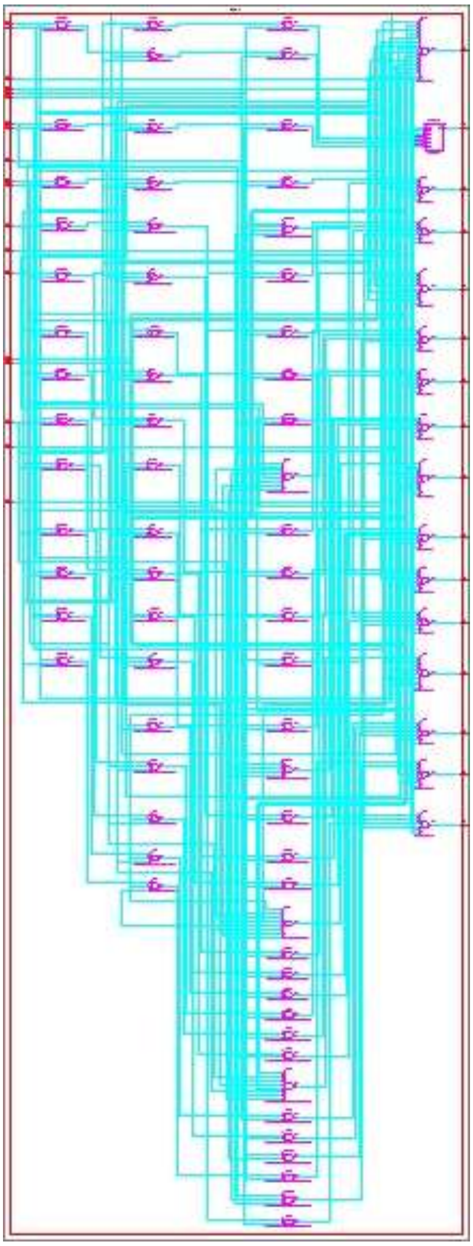


Fig. 6 Expanded view of RTL Design in the present work

The hardware utilization summary based on the FPGA for the present 16-bit DHT design is shown in Table-I. This is performed on Xilinx ISE Tool.

TABLE I
HARDWARE UTILIZATION SUMMARY OF THE DHT DESIGN IN PRESENT WORK

Xilinx FPGA XC3S500E- PQ208	**Available hardware**	**Used hardware**	**Percentage hardware used**
No. of Slices	4656	37	0
No. of 4 input LUTs	9312	64	0
No. of bounded IOBs	158	35	22

The Test bench based simulation waveform of present design is shown in fig. 7.

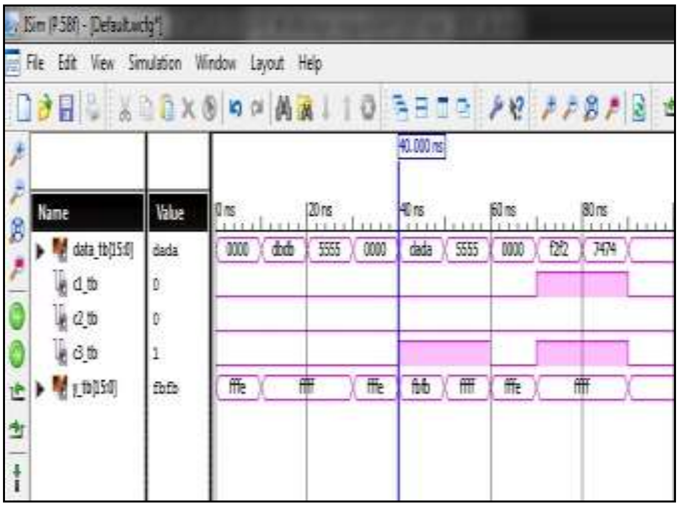


Fig. 7 Testbench based Simulation Waveform of the present design

Maximum combinational path delay for this work is 10.113ns and the total number of logic levels of the designed logic is '6'.

VI. CONCLUSIONS

The new transform which is DHT is used to transform real input value to real output value. An ancient Vedic technique for multiplication is named Urdhwa Triyambakam. DHT has a vast field of applications such as image processing, space science, scientific applications etc. The two key factors which need to be keep in mind while realizing the circuit is area required and delay provided by the hardware. Here we present DHT by using full adder, OR and XNOR gates. The design in the present work requires less path delay and requires less number of slices for design synthesis.

ACKNOWLEDGMENT

The authors thank Mr. Piyush Jain (Director, Innovative Technology Design and Training Center, Bhopal, India) for sharing his ideas in writing this paper.

REFERENCES

[1] R. N. Bracewell, "The fast Hartley transform" ,Proc. IEEE, vol. 72, no. 8, pp. 1010- 1018, August, 1984.

[2] H. V. Sorenson, et.al, "On computing the discrete Hartley transform", IEEE Transaction, Acoustic., Speech, Signal Processing, vol. ASSP-33, No. 4, pp. 1231–1238, October, 1985.

[3] C.Chakrabarti, "On prime factor mapping for the discrete hartley transform - comment", IEEE transactions on signal processing, pp. 1551-1552, 1994.

[4] R. I. Hartley, "Sub expression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, vol. 43, no. 10, pp. 677–688, October, 1996.

[5] Sung Bum Pan et al "Unified systolic array for fast computation of the discrete cosine transform, discrete sine transform, and discrete Hartley transform", Opt. Eng. 36(12), pp. 3439-3444 December 01, 1997.

[6] Lizhi Cheng et.al. "Fast Hartley transform and truncated singular value algorithm for circular de convolution", SPIE .Opt.Engg., pp. 2137-2142, 1997.

[7] Guoan Bi et.al." New split-radix algorithm for the discrete Hartley transform", IEEE Transactions on Signal Processing, pp. 297-302, 1997.

[8] Pei S C, Tseng C C, Yeh M H, et al. "Discrete fractional Hartley and Fourier transform", IEEE Trans Circ System II, pp. 665—675, 1998.

[9] In Hao, P. and Shi, Q. "Comparative Study of Color Transforms for. Image Coding and Derivation of Integer Reversible Color Trans form," pp. 224-227, IEEE 2000.

[10]Guoan Bi et.al. "Fast algorithms for generalized discrete Hartley transform of composite sequence lengths", IEEE Transactions on Vol.47, Issue: 9, 2000.

[11]Tseng C C et.al."Eigen values and eigenvectors of generalized DFT, generalized DHT, DCT-IV and DST-IV matrices", IEEE Trans Signal Process, pp. 866—877, 2002.

[12]P. K. Meher et.al. "Near Lossless Image Compression Using Lossless Hartley Like Transform", Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, vol. 1, pp.213-217, December 2003.

[13]German rodriguez et. al. "Analysis and simulation of wave records through Fast hartley transform", Ocean Engineering,Volume 30, Issue 17, December 2003.

[14]Bouguezel, S. et.al. "A new split-radix FHT algorithm for length-q*2mDHTs", Circuits and Systems I: Regular Papers, IEEE Transactions, pp. 2031- 2043, 2004.

[15]Chien-Cheng Tseng et.al. "Quantum circuit design of 8 × 8 discrete Hartley transform", Circuits and Systems, ISCAS '04. Proceedings of the 2004 International Symposium, Vol.3, 2004 .

[16]S Bouguezel et.al. "New Parametric Discrete Fourier and Hartley Transforms, and Algorithms for Fast Computation", Circuits and Systems I: Regular Papers, IEEE Transactions on,Vol.58, Issue: 3 2011.

[17]H.B. Kekre et.al. "Performance Enhancement of Fractional Coefficients of the Discrete Hartley Transform for Palm Print Recognition", Journal of Telecommunications, Vol.11, Issue 1, october 2011.

[18]D F Chiper et.al. "Fast Radix-2 Algorithm for the Discrete Hartley Transform of Type II", Signal Processing Letters, IEEE Vol.18 , Issue: 11, November, 2011.

[19]Gautam Abhyachand Shah et.al. "A Fast Radix-4 Algorithm and Architecture for DHT", Int. J Comp Sci. Emerging Tech Vol-2 No 5 October, 2011.

[20]V. Baby Deepa et.al. "A study on classification of EEG Data using the Filters", IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 4, 2011.