



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 2, February 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com

Review on Discrete Hartley Transform using Different Multiplier Technique

Anoop Mishra¹, Prof. Swati Gupta²

M. Tech. Scholar, Department of Electronics and Communication Engineering, Vidhyapeeth Institute of Science and Technology, Bhopal, M.P, India¹

Head of Dept., Department of Electronics and Communication Engineering, Vidhyapeeth Institute of Science and Technology, Bhopal, M.P, India²

ABSTRACT: A discrete Hartley transform (DHT) algorithm can be efficiently implemented on a highly modular and parallel architecture having a regular structure is consisting of multiplier and adder. Discrete Hartley Transform (DHT) is one of the transform used for converting data in time domain into frequency domain using only real values. We have proposed a new algorithm for calculating DHT of length 2^N , where $N=3$ and 4 . We have implemented Urdhwa multiplier based on compressor 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, 4:2 Compressor, 7:2 compressor and Xilinx Vertex family

I. 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. 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. 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. 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. 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 improvement of DFT in which computation has becomes faster.

All the family members of Fourier till now works on complex values which requires large storage space and computationally complex in nature. So, now comes a new member of transform called Discrete Hartley transform (DHT) which converts real values into real values. Therefore, it needs lesser storage space and less computational complexity.

In this paper, DHT is computed using simple addition and multiplication is done by using Urdhwa multiplier uses different 4:2 and 7:2 compressors along with half adder, full adder, XOR Gate and XNOR Gate for complete multiplication process.

II. LITERATURE REVIEW

Nikhil Advait Gudala et al. [1], one of the most substantial areas of research in VLSI is the design of power-efficient and high-speed data path logic systems. The speed of addition is constrained in digital adders by the time taken to propagate a carry through the adder. When the previous bit has been added and the carry generated from this addition is

propagated onto the next bit, the sum for each bit in an elementary adder is generated sequentially in this manner. The core concept of this study is to use the Binary to Excess-1 Converter (BEC) to achieve high speed and low-power consumption instead of the RCA in the regular CSLA. The results have been analyzed and compared for implementation of three adders (conventional CSA, CSA with RCA, CSA with BEC). The results from the performance evaluations of the adders are compared with each other. All the simulation is carried out in 45nm technology.

Deepti Gautam et al. [2], 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 $N=8$.

Omid Akbari et al. [3], in this paper, we propose four 4:2 compressors, which have the flexibility of switching between the exact and approximate operating modes. In the approximate mode, these dual-quality compressors provide higher speeds and lower power consumptions at the cost of lower accuracy. Each of these compressors has its own level of accuracy in the approximate mode as well as different delays and power dissipations in the approximate and exact modes. Using these compressors in the structures of parallel multipliers provides configurable multipliers whose accuracies (as well as their powers and speeds) may change dynamically during the runtime. The efficiencies of these compressors in a 32-bit Dadda multiplier are evaluated in a 45-nm standard CMOS technology by comparing their parameters with those of the state-of-the-art approximate multipliers. The results of comparison indicate, on average, 46% and 68% lower delay and power consumption in the approximate mode. Also, the effectiveness of these compressors is assessed in some image processing applications.

G.Challa Ram et al. [4], this paper describes the design of high speed Vedic multiplier that uses the techniques of Vedic mathematics based on 16 sutras (algorithms) to improve the performance. The efficiency of Urdhva Tiryagbhyam (vertical and crosswise) Vedic method for multiplication which is different from the process of normal multiplication is presented. Urdhva -Tiryagbhyam is the most efficient algorithm that gives minimum delay for multiplication for all types of numbers irrespective of their size. Vedic multiplier is coded in Verilog HDL and stimulated and synthesized by using XILINX software 12.2 on Spartan 3E kit. Further the design of array multiplier is compared with the proposed multiplier in terms of delay, memory and power consumption.

G. Gokhale et al. [5], in this paper, area-efficient Vedic multiplier is designed using modified Carry Select Adder (CSLA). As the multiplication is nothing but subsequent addition process, adder is important block in the design of multiplier. Simple Ripple Carry adder (RCA) can be used for implementing multiplier. Digital adder has problem of carry propagation, thus carry select adder is used instead. Carry select adder is known to be one of the fastest adder structures. Here novel technique that is Vedic multiplier is implemented instead of normal multipliers like add and shift multiplier, array multiplier etc. Here multiplier is designed based on ancient Vedic multiplication technique. The goal of this paper is to design area-efficient Vedic multiplier based on crosswise and vertical algorithms. Conventional CSLA designs are compared with proposed design to prove its efficiency. On an average, modified CSLA has 29 % less area than Binary to Excess one Converter (BEC) based CSLA for different bit widths. Proposed Vedic multiplier is designed using modified CSLA, which has approximately 10 % less area than BEC based Vedic multiplier. It shows improved performance in terms of area. Proposed design is also compared with the Booth multiplier. Proposed multiplier showed more excellent results than Booth multiplier.

G. Gokhale et al. [6], in this paper, Vedic multiplier is designed using area-efficient Carry Select Adder (CSLA). As the multiplication is the process of subsequent addition, adder is important block in implementation of multiplier. Digital adder has problem of carry propagation, thus carry select adder is used instead of simple Ripple Carry Adder (RCA). Carry select adder is known to be one of the fastest adder structures. Here Vedic multiplier is implemented instead of normal multipliers like add and shift multiplier, array multiplier etc. The goal of this paper is to design Vedic multiplier based on crosswise and vertical algorithms using area-efficient CSLA. Conventional CSLA designs like

Binary to Excess one Converter (BEC) based CSLA and Modified CSLA (MCSLA) are compared with proposed CSLA design to prove its efficiency. It shows improved performance in terms of area. This renovated CSLA is used to design proposed Vedic multiplier. It has 6% less area than Vedic multiplier using MCSLA and 16% less area than Vedic multiplier using BEC-based CSLA. Proposed design is also compared with the Booth multiplier. Proposed multiplier showed more excellent results than Booth multiplier.

Shirali Parsai et al. [7], 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. Our work have proposed a new algorithm for calculating DHT of length $2N$, where $N=3$ and 4. Our work 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.

Doru Florin et al. [8], in this paper another expansive scale mix (VLSI) calculation for a $2N$ - length discrete Hartley change (DHT) that can be productively actualized on an exceptionally secluded and parallel VLSI construction modeling having a general structure is introduced. The DHT calculation can be productively part on a few parallel parts that can be executed simultaneously. Additionally, the proposed calculation is appropriate for the sub expression sharing procedure that can be utilized to fundamentally lessen the equipment multifaceted nature of the exceptionally parallel VLSI execution. Utilizing the upsides of the proposed calculation and the way that we can productively share the multipliers with the same consistent, the quantity of the multipliers has been fundamentally diminished such that the quantity of multipliers is little contrasting and that of the current calculations. Besides, the multipliers with a consistent can be productively actualized in VLSI.

III. URDHWA MULTIPLIER

Multipliers are the blocks in DSP processors which requires large amount of storage and processing complexity. Vedic mathematics has well known an ancient method for mathematical operations which are easy to use Urdhwa triyambakam is a method used for multiplication calculations. It takes the advantage of vertical and crosswise calculations. In this technique multiplication of terms is obtained by using simple AND logical operations, full adders and half adders.

In this paper a 8 bit Urdhwa multiplier is used. For this purpose of addition, we have been used various designs of adders and compressors. Here, these adders and compressors are deployed in such a way so as to provide optimized design of multiplier in terms of delay. Four half adders, two full adders, five 7: 2 compressor and ten 4:2 compressors has been used.

We have presented three designs for Urdhwa multiplier. These three designs vary in terms of 4:2 compressor designs. First design uses the simple 4:2 compressor which uses two full adders. Second design uses four XOR gates and two multiplexers (2×1). Third design of 4:2 compressors is two XOR-XNOR gates and four multiplexers (2×1). First design of 4:2 compressors is simplest one, but more delay. Next design has lesser delay as compared to the first design at the cost of higher complexity. But the last which is our proposed design has least amount of delay but has greatest complexity and occupies maximum space among all of them.

We have designed 7:2 compressor using different designs of 4:2 compressor thus we got three variations of 7:2 compressor.

• 4:2 Compressor

To add binary numbers with minimal carry propagation we use compressor adder instead of other adder. Compressor is a digital modern circuit which is used for high speed with minimum gates requires designing technique. This compressor becomes the essential tool for fast multiplication adding technique by keeping an eye on fast processor and lesser area.

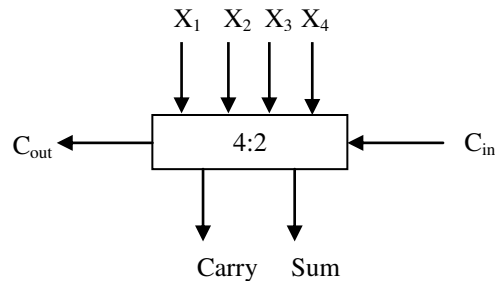


Figure 1: Block Diagram of 4:2 Compressors

4:2 compressors are capable of adding 4 bits and one carry, in turn producing a 3 bit output. 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 (C_{in}) and a Carry-out (C_{out}) as shown in Figure 1. The input C_{in} is the output from the previous lower significant compressor.

The C_{out} is the output to the compressor in the next significant stage. The critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adders. Similar to the 3-2 compressor the 4-2 compressor is governed by the basic equation the standard implementation of the 4-2 compressor is done using 2 Full Adder cells as shown in Figure 2(a). When the individual full Adders are broken into their constituent XOR blocks, it can be observed that the overall delay is equal to $4 \times \text{XOR}$.

$$X_1 + X_2 + X_3 + X_4 + C_{in} = \text{sum} + 2 * (\text{Carry} + C_{out})$$

$$\text{Sum} = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in}$$

$$C_{out} = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).X_4$$

$$C_{arry} = U.C_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)}.X_4$$

Where

$$U = X_1 \oplus X_2 \oplus X_3 \oplus X_4$$

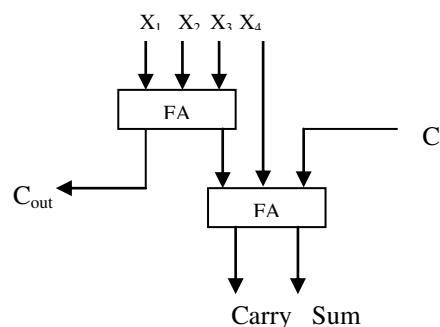


Figure 2(a): 4:2 Compressors using Full Adder

The standard implementation of the 4-2 compressor is done using 2 Full Adder cells as shown in Figure 2(a). When the individual full Adders are broken into their constituent XOR blocks, it can be observed that the overall delay is equal to $4 \times \text{XOR}$.

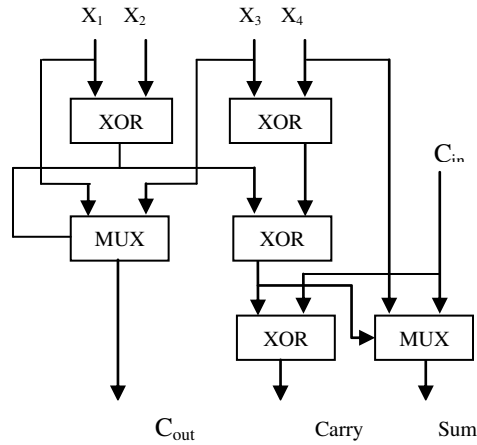


Figure 2(b): 4:2 Compressors using XOR Gate

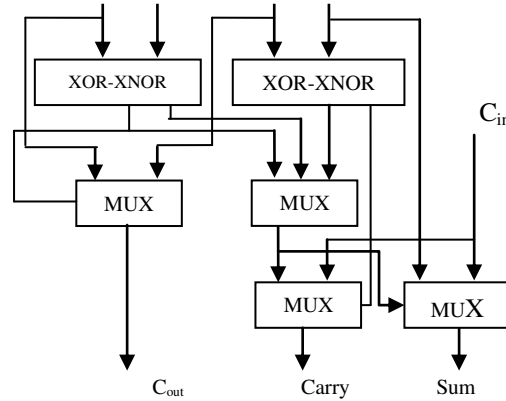


Figure 2(c): 4:2 Compressors using XOR-XNOR Gate

Thus replacing some XOR blocks with multiplexer's results in a significant improvement in delay. 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 Figure 2(c).

The equations governing the outputs in the proposed architecture are shown below

$$Sum = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).(X_3 \oplus X_1).C_{in}$$

$$C_{out} = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).X_1$$

All the designs have been captured by VHDL and the functionality is verified by RTL and gate level simulation shown in table 1. Designs are implemented on a Xilinx Spartan 3 FPGA using VHDL as the RTL language with the help of Xilinx ISE design suite 14.1.

TABLE 1: Device utilization summary (Vertex-4) of 4:2 Compressors using full adder, 4:2 Compressor using XOR Gate and 4:2 Compressor using XOR-XNOR Gate

Design	No. of Slices	No. of 4 input LUTs	MCPD (ns)
4:2 Compressor	3	6	6.257
4:2 Compressor using XOR gate	2	4	5.663
4:2 Compressor using XOR-XNOR gate	2	3	5.280

• 7:2 Compressor

Similar to its 4:2 compressor counterpart, the 7:2 compressors as shown in Figure 3, is capable of adding 7 bits of input and 2 carry's from the previous stages, at a time. 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.

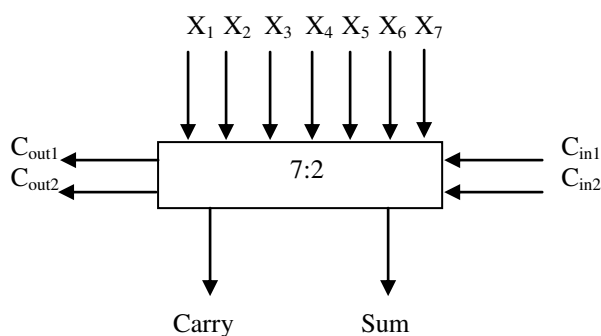


Figure 3: Block Diagram of 7:2 Compressors

IV. DISCRETE HARTLEY TRANSFORM

Discrete Hartley Transform is abbreviated for DHT and this transform was proposed by R. V. L. Hartley in 1942. 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.



Figure 4: Block Diagram of DHT

In other words, 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.

Let $N \geq 4$ be a power of two. For any real input sequence $\{x(i) : i = 0, 1, 2, \dots, N-1\}$

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

$$= \sum_{i=0}^{N-1} x(i) \cdot \text{cas}[2ki\pi / N] \quad \text{for } k = 0, 1, \dots, N-1$$

Where $\text{cas}(x) = \cos(x) + \sin(x)$

V. PROPOSED TECHNOLOGY

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

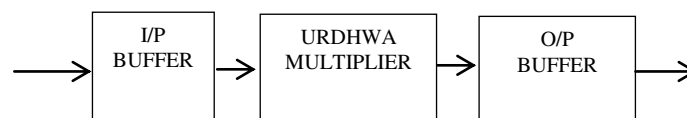


Figure 5: Proposed Architecture

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 8×8 Urdhwa multiplier by using full adder, OR and XNOR gates in different types of compressors. Among all three compressors, compressor provides the least amount of delay. Also, it takes least number of slices i.e. Occupy least area among all three compressors.

REFERENCES

- [1] Nikhil Advait Gudala and Trond Ytterdal, "Implementation of High Speed and Low Power Carry Select Adder with BEC", International Midwest Symposium on Circuits and Systems IEEE 2021.
- [2] N. Hamed S. Timarchi "Low-Power and Fast Full Adder by Exploring New XOR and XNOR Gates" IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2018.
- [3] Low Power Approximate Adders for General Computing Using Differential Transmission Gate" on 28.03.2018 National Conference on Innovations in Communication and Computing NCICC" 18 SNS College of Technology.
- [4] Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram, "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 34, Issue 7, 2017.



- [5] Doru Florin Chipur, *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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda", Delhi (2011).
- [10] Sumit Vaidya and Depak Dandekar. "Delay-power perfor-mance comparison of multipliers in VLSI circuit design". International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.
- [11] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Procesing", Global J. of Eng. Edu, Vol.8, No.2, 204, UICEE Published in Australia.
- [12] Asmita Haveliya, "Design and Simulation of 32-Point FFT Using Radix-2 Algorithm for FPGA Implementation", Second International Conference on Advanced Computing & Communication Technologies IEEE 2012.
- [13] S. Correa, L. C. Freitas, A. Klautau and J. C. W. A. Costa, "VHDL Implementation of a Flexible and Synthesizable FFT Processor", IEEE LATIN AMERICA TRANSACTIONS, VOL. 10, NO. 1, JAN. 2012.



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details