Performance Analysis of efficient Discrete Hartley Transform using Kogge-Stone Adder

[1]Bhangale Sushant Vijay, [2]Dr Neetesh Raghuwanshi, [3]Dr Bharti Chourasia

[1]M. Tech Scholar, Department of Electronics and Communication, RKDF IST SRK University, Bhopal, India,

[2]Assistant Professor, Department of Electronics and Communication, RKDF IST SRK University, Bhopal, India,

[3] HOD, Department of Electronics and Communication, RKDF IST SRK University, Bhopal, India.

Abstract: **Discrete Hartley transform is one of the most important algorithms of the signal processing and image processing system. Now a day in every field required an ever growing demand for high speed processing and low area design. Many types of discrete Hartley transform algorithm are design in different adder but day by day is required high speed adder. In this paper, we introduce a novel architecture to perform high speed adder using half adder (HA) and XOR gate techniques. Here, for enhancing the speed of addition, we are proposing Kogge Stone adder instead of other adder like ripple carry adder, look a-head carry adder in different-different manner, which has less propagation delay. All design is verified in Xilinx tool with different device family and the timing and area of the design, on the same have been calculated.**

IndexTerms **- Discrete Hartley Transform (DHT), Kogge-stone adder, and Xilinx Spartan family**

I. Introduction

The processor's speed mostly depends on multiplier design techniques. For a highly modular and complex parallel architecture can be constructed by using novel DHT algorithm and multipliers. There are so many ways and algorithm to design multiplier efficiently. Propagation delay and area can be reduced by using ripple carry generator and fast adder. A Vedic mathematics multiplier for DHT algorithm is important tool for communication and other high speed processor. In this paper all the designing and experiment are done on Xilinx 14.2i updated version. Xilinx 9.2i requires low memory and other beneficiary quality is fast debug and programming.

The discrete Hartley transform is the essential terminology of the digital signal processing which can be implemented in the form of either linearly or circularly. With the advent of new fast technology devices must be high efficient and portable. Day by day numbers of the transistors are increasing at drastically manner. So designer must be kept in mind that device should be structured with less number of gates and other active and passive elements. Here we are designing a fast efficient convolution technique which can be used in digital signal processing, image signal processing, biological and mathematical solutions. With the aid of discrete Hartley transform to noise ratio can be managed and blurred signal be filtered. Convolution of the two signals is just like a multiplication of the binary information but difference is that carry does not propagate to the next bit. So, if we design fast multiplication then convolution will be automatically fast. The core element of the multiplier is adder. Here we are using a modified Kogge Stone adder which is better than other parallel adder like Ripple carry, Look ahead carry and carry select adder. And multiplication calculations will be done by using Indian ancient Vedic mathematics technique. All the experiment and simulation will be done on Xilinx 14.2 software, Spartan 3 series.

Novel Vedic multiplier is not only for fast multiplication but also used in so many important technologies that are essential for digital era with new advent. Here, I am using this high speed multiplier in node multiplier for DHT algorithm. DFT is used in many digital processing.

The classical split radix algorithm is difficult to implement on VLSI due it its irregular computational structure and due to its irregular computational structure and due to fact that the butterflies significantly differ from stage to stage. Another method is systolic array implementation of DHT.

II. Discrete HartleyTransform

Discrete Hartley Transform is truncated for DHT and this change was proposed by R. V. L. Hartley in 1942. DHT is the similar to Fast Fourier change which gives the main genuine quality at any expense. The principle distinction from the DFT is that it changes the genuine inputs to genuine yields with no inborn inclusion of complex worth. DFT can be utilized to register the DHT, and the other wayaround.

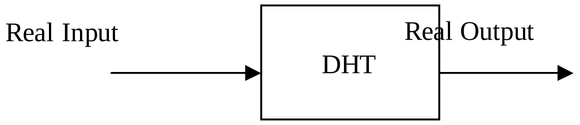


Figure 1: Block diagram for DHT

Discrete Hartley Transform is a linear operator because it interprets as the multiplication of the vector (X_1, $X_{2......}X_{n-1}$) by an N-by-N matrix. For the real inputs Xn, the DFT output X_k has a real part (H_k + H_{N-K})/2 and an imaginary part (H_kH_{N-K})/2. Eventually, the DHT is equivalent to computing the DFT of Xn multiplied by 1+I, then taking the real part of the result. According to Discrete Hartley Transform the real numbers X_0, X_1, X_2........X_{N-1} are converted into the real numbers H_0, H_1, H_2.........H_{N-1}

$$Hk = \sum_{n=0}^{N-1} XN\left(\cos\frac{2n\pi k}{N} + \sin\frac{2n\pi k}{N}\right)$$

DHT algorithm is applied where we talk about fast discrete data transform. There are several split radix algorithms for computing DHT with a low arithmetic cost. Thus Sorensen and Malvar proposed split radix algorithms for DHT with a low arithmetic cost. Bi proposed another split radix algorithm where the odd-indexed transform out form outputs is computed using an indirect method.

The classical split radix algorithm is difficult to implement on VLSI due it its irregular computational structure and due to its irregular computational structure and due to fact that the butterflies significantly differ from stage to stage.Another method is systolic array implementation of DHT. Systolic array architectures are modular and regular, but they use particularly pipelining and not parallel processing to obtain a high speed processing.

There are several split radix algorithms for computing DHT with a low arithmetic cost. Thus Sorensen and Malvar proposed split radix algorithms for DHT with a low arithmetic cost. Bi proposed another split radix algorithm where the odd-indexed transform out form outputs is computed using an indirect method. The classical split radix algorithm is difficult to implement on VLSI due it its irregular computational structure and due to its irregular computational structure and due to fact that the butterflies significantly differ from stage to stage.Another method is systolic array implementation of DHT. Systolic array architectures are modular and regular, but they use particularly pipelining and not parallel processing to obtain a high speed processing.

III. DIFFERENT TYPES OFADDER

Generally we are having two type of adder such as serial and parallel adder. But in thesis we seeking parallel adder which provides the addition of many bits parallel. In digital circuit system we are having so many parallel adders but some adder is frequently used to develop a processor's speed. Some important parallel adders are followings:

- Ripple carry adder
- Carry select adder
- Look ahead carry adder
- Kogge-Stone adder

Ripple Adder: - Parallel adder is much faster in speed than to serial adder and ripple carry adder is one of them. Ripple carry adder can be constructed using of full adder circuits connected in parallel or cascade. The carry bit of each adder is connected to the carry input of the next higher order adder that is why this adder is called a ripple carry adder.

Ripple carry adder can be enhanced that either a half adder can be used for the least significant position or the carry input of a full adder is made $0$because there is no carry into the least significant bit position. For instance we have to design a 4 bit ripple carry and inputs are (A_3, A_2, A_1, A_0) and (B_3, B_2, B_1, B_0) then output are

$$(C_{out}, S_3, S_2, S_1, S_0) S_0 = (A_0 \oplus B_0) \oplus C_{in} \qquad (2)$$

$$S_1 = (A_1 \oplus B_1) \oplus C_0 \qquad (3)$$

And

$$S_2 = (A_2 \oplus B_2) \oplus C_1 \qquad (4)$$

$$S_3 = (A_3 \oplus B_3) \oplus C_2 \qquad (5)$$

Where, $C_0, C_1, C_2$are carry output of the full adder.

Kogge-stone Adder:- Kogge Stone Adder was proposed by Peter M. Kogge and Harold S. Stone. Kogge Stone Adder is an advanced technology of Look a- head Carry Adder. That is also called parallel prefix adder. It has more area than to Brent Kung Adder but less Fan-out. This adder provides the carry signal time $(O_{log n})$and become fastest adder for industrial level. There are so many adders those provides high processing speed but not well in propagation delay. KSA adder gives the low propagation delay for high bit addition. This can be used for multiplication purpose because multiplier is comprised with so many adders. A high speed processor can be designed by using KSA adders. The efficient routine of Kogge-Stone adder is because of its minimum logic depth and bounded fan-out. On the other side it occupies large silicon area. The Kogge–Stone adder is a parallel prefix form carry look-ahead adder and other parallel adder [9].

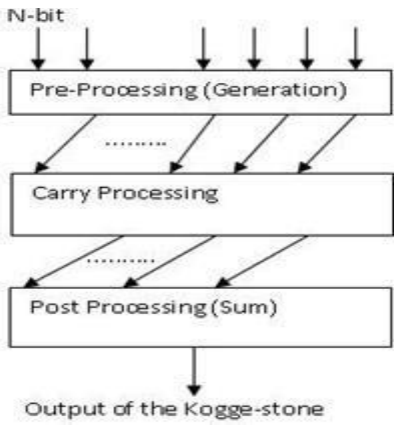


Figure 4: Block diagram of Kogge stone Adder

$$P_n = A_n \oplus B_n \qquad (6)$$

$$G_n = A_n . B_n \qquad (7)$$

The middle stage of the KSA adder is Carry processing stage which provides the carries corresponding to each bit [9]. Execution of these bit operation is carried out from parallel. After finding the carries in parallel they are segmented in to smaller pieces. The calculation of the middle stage must be fast as possible as and circuit must be less complex.

$$CP_{n-1} = P_{n-1} \oplus P_n \qquad (8)$$

$$CG_{n-1} = (P_n \oplus G_{n-1}) + G_n \qquad (9)$$

Bottom block is summation block which provides the summation bits. That blocks are comprised with XOR gate. If one input is different from another then output will be high. And if inputs are same then outputs will be low. Kogge Stone provides the less area than to other parallel adder like carry select adder, carry save adder, Manchester and look ahead adder.

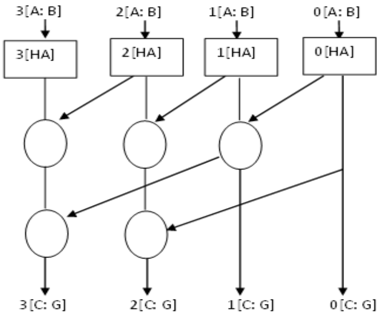


Figure 5: Function Diagram of Kogge-stone

IV. PROPOSED ALGORITHM

In order to clearly illustrate the features and advantages of the algorithm, the VLSI architecture for a DHT of length N=8 is presented in figure 2.2. it can be seen that the architecture is highly parallel and has a modular and regular structure being formed of only a few sharing block: addition, sub tractor and multiplier. Each multiplier is shared by two inputs with a constant.

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 anypermutation.

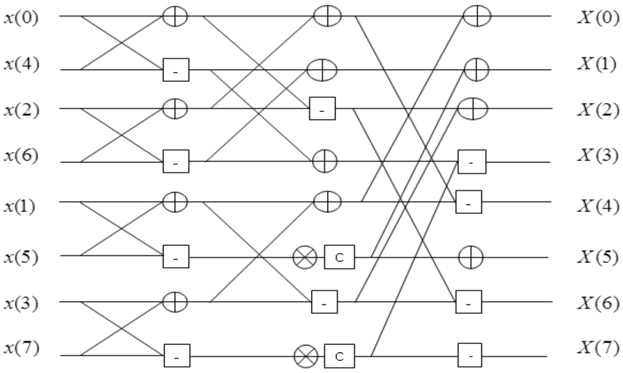


Figure 6: 8-point Discrete Hartley Transform

- Mathematical calculation for N=8

$$X(0) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square [(x(1) \square x(5)) \square (x(3) \square x(7))]$$
$$X(2) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square [(x(1) \square x(5)) \square (x(3) \square x(7))]$$
$$X(4) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square [(x(1) \square x(5)) \square (x(3) \square x(7))]$$
$$X(6) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square [(x(1) \square x(5)) \square (x(3) \square x(7))]$$
$$X(1) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square c(x(1) \square x(5))$$
$$X(3) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square c(x(3) \square x(7))$$
$$X(5) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square c(x(1) \square x(5))$$
$$X(3) \square [(x(0) \square x(4)) \square (x(2) \square x(6))] \square c(x(3) \square x(7))$$

$\textit{with } c \square \sqrt{2}$

Where c is the multiplier

V. SIMULATION RESULT

Implementation Doru Florin Chiper [1] and proposed discrete Hartley transform based on modified 16-bit CSLA. The proposed design has been captured by VHDL and functionality is verified by RTL and gate level simulation.

VHDL is an acronym for VHSIC (Very high Speed Integrated Circuit) Hardware Description Language. It is a hardware description language that can be used to describe the structure and/or behavior of hardware designs and to model digital systems.

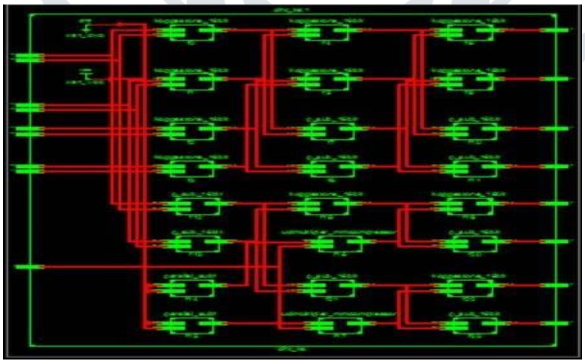
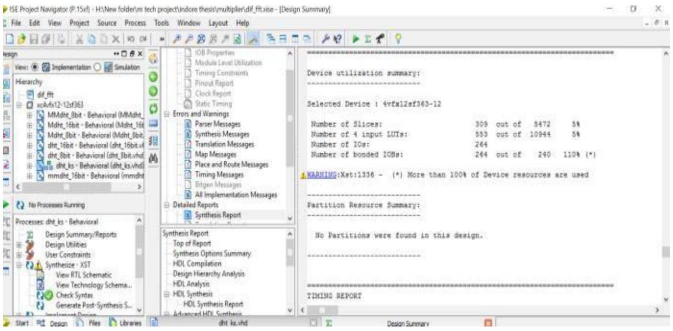
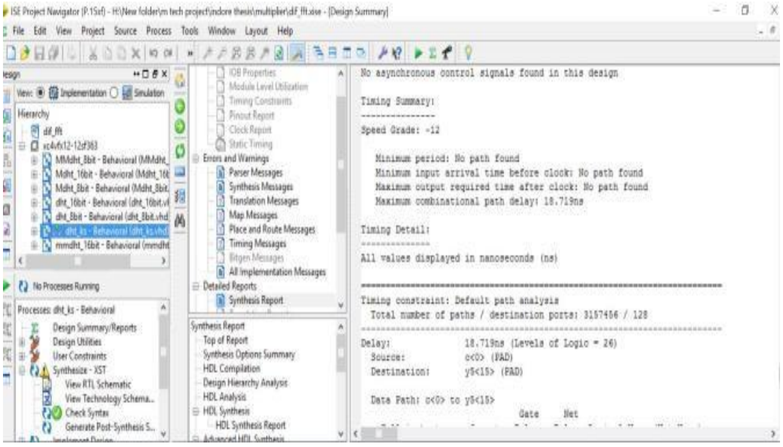


Figure 7: RTL view of Proposed DHT using Kogge-stone adder



```
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 4vfx12sf363-12

 Number of Slices:                      309  out of   5472     5%
 Number of 4 input LUTs:                553  out of  10944     5%
 Number of IOs:                         264
 Number of bonded IOBs:                 264  out of    240   110% (*)

WARNING:Xst:1336 -  (*) More than 100% of Device resources are used

---------------------------
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.

---------------------------


=========================================================================
TIMING REPORT
```



```
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -12

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 18.719ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 3157456 / 128
-------------------------------------------------------------------------
Delay:               18.719ns (Levels of Logic = 26)
  Source:            c<0> (PAD)
  Destination:       y5<15> (PAD)

  Data Path: c<0> to y5<15>
                                Gate     Net
```

Figure 8: Simulation result for proposed 8-point DHT using Kogge Stone adder Table 2:

Comparison result of the existing algorithm and proposed algorithm

Architecture	Number of Slice	Number of LUTs	IOB	MCPD
Existing Algorithm	617	1093	264	37.57ns
Proposed Algorithm	530	940	264	30.34ns

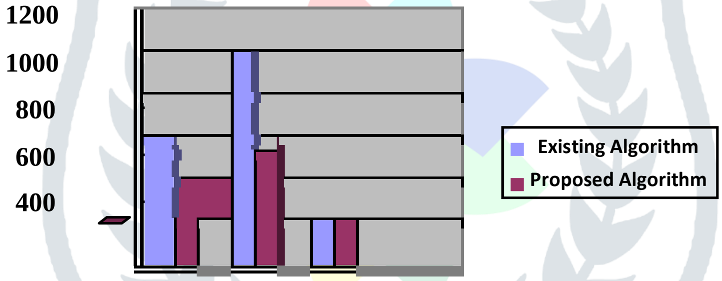
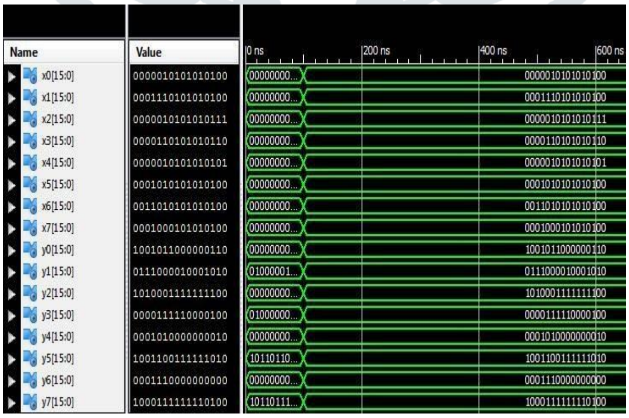


Figure 9: Show the bar graph of 8-bit DHT



Name	Value
x0[15:0]	0000010101010100
x1[15:0]	0001110101010100
x2[15:0]	0000010101010111
x3[15:0]	0000110101010110
x4[15:0]	0000010101010101
x5[15:0]	0001010101010100
x6[15:0]	0011010101010100
x7[15:0]	0001000101010100
y0[15:0]	1001011000000110
y1[15:0]	0111000010001010
y2[15:0]	1010001111111100
y3[15:0]	0000111110000100
y4[15:0]	0001010000000010
y5[15:0]	1001100111111010
y6[15:0]	0001110000000000
y7[15:0]	1000111111110100

Figure 10: Output Waveform of 8-bit DHT with word length 16

VI. 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. Modified SQRT CSLA is used not only for fast addition but also used in so many important technologies that are essential for digital era with new advent. Here, we using this high speed modified SQRT CSLA in node addition for DHT algorithm. DFT is used in many digital processing applications as in signal and image compression techniques, filter banks, signal representation, or harmonic analysis. The discrete Hartley transform (DHT) can be used to efficiently replace the DFT when the input sequence is real.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] B. Ramkumar and Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 20, NO. 2, February 2012.

[5] Sushma R. Huddar and SudhirRao, Kalpana M., "Novel High Speed Vedic Mathematics Multiplier using Compressors", 978-1 - 4673-5090- 7/13/$31.00 ©2013 IEEE.

[6] S. S. Kerur, PrakashNarchi, 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 Joural of Computer Applications® (IJCA), pp.1 -6.

[7] HimanshuThapaliyal 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.

[8] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda", Delhi(2011).

[9] SumitVaidya and DepakDandekar. "Delay-power perfor-mance comparison of multipliers in VLSI circuit design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 20