International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-5 Issue-2, December 2015

Design and Implementation of an N bit Vedic Multiplier using DCT

Shazeeda, Monika Sharma D

Abstract*— One of the basic and fundamental functions in arithmetic operation is multiplication. Many of the application such as convolution and Fourier transform in digital signal processing, in microprocessors multiplication is very frequently used operation. In this paper we propose a fast multiplication method based on ancient Indian Vedic mathematics. The Vedic mathematics demonstrate the unified structure of mathematics by the 16 formulas. The generalized multiplication formula which is applicable in all cases is called Urdhava Triyakbhyam. In this paper we designed a Vedic multiplier in VHDL (Very High Speed Integrated circuit Hardware Description Language) and synthesis is done in Xilinx ISE series. The combinational delay of this multiplier is estimated and compared with that of Wallace tree multiplier. The results showed a significant improvement in the propagation delay. The Vedic multiplier showed a propagation delay of 10.295 ns and 25.236 ns for 4 and 8 bit multiplication, respectively.

Index Terms*— Vedic multiplier, Wallace tree multiplier, Urdhva Tiryakbhyam, Discrete cosine transform.

I. INTRODUCTION

Indian mathematics is a unique technique of arithmetic computation based on 16 sutras (formulae). It is applicable in fields of mathematics like geometry, trigonometry, quadratic equations and calculus. An extensive research was done by Jagadguru Shankaracharya Bharti Krishna Tirthaji Maharaja (1884-1960) in Vedas and obtained a simple form of calculation. Through an extensive research in atharvavedas, he developed 16 sutras (formulae) and upya sutras (sub formulae)[1]. One of the important and fastest growing field in technology is digital signal processing. Mathematical calculations such as addition, multiplication should be fast in this field such as for convolutions, Discrete Fourier Transform, digital filters and other applications. A brief explanation on Vedic multiplier based on urdhava triyakbhm Sutra is discussed in the following discussion. Asmita Haveliya proposed a significant method to developed a multiplier architecture based on vertical and crosswise structure of ancient Indian Vedic mathematics. The proposed method showed high performance, high throughput and area efficient architecture of multiplier for Field Programmable Gate Array (FPGA) [2]. Chidgupka et. al. proposed the use of multiplication process based on Vedic algorithms and its implementations on 8085 and 8086 microprocessors, resulting in higher processing time [3]. Sharma [4] proposed multiplier architecture which is based on algorithm of ancient Indian Vedic mathematics, high speed applications. This multiplier is based on the vertical and crosswise algorithm of Urdhva Tiryakbhyam sutra generating all partial products and their sums in one step [4]. Saha et. al. proposed a 32 bit multiplier for high speed low power processor. The implementation methodology ensures substantial reduction of propagation delay in comparison with Wallace Tree (WTM), modified Booth Algorithm (MBA), Baugh Wooley (BWM) and Row Bypassing and Parallel Architecture (RBPA) based implementation which are most commonly used architectures [5]. Kavita et. al. proposed a multiplication technique using ancient Indian Vedic mathematics, which described in this paper are Nikhilam sutra, Urdhva tiryakbhyam and Karatsuba-ofman and the performance analysis of these techniques is obtained [6]. Sherif [7] proposed the design and implementation of a 1-Dimensional eight word DCT/IDCT processor that can be used in most of video/audio compression CODECs, such as JPEG. The core was designed taking into consideration maximum area optimization. It can process audio frames and JPEG still images (compression and decompression) with high speed [7]. Muniraj et. al. [8] proposed a method of designing DCT using Vedic mathematics. Multipliers are fundamental and area intensive component in the architecture of any DSP system [8].

This paper proposes an N-bit Vedic multiplier by using generic method. The proposed 8 bit Vedic multiplier is compared with that of 8 bit Wallace tree multiplier in terms of the delay. The delay that is obtained from the Wallace tree multiplier is larger when compared to that of the Vedic multiplier. Vedic multiplier is used in the digital signal processing applications like DCT/IDCT which is proposed and implemented in the present paper.

II. PROCEDURE FOR PAPER SUBMISSION

A. *Urdhva Tiryakbhyam Sutra*

The given Vedic multiplier based on the Vedic multiplication formulae. This Sutra has been traditionally used for the multiplication of two numbers. Urdhva tiryakbhyam sutra is a general multiplication formula applicable to all cases of multiplication. It means “Vertically and crosswise”. The digits on the two ends of the line are multiplied and the result is added with the previous carry. When there are more lines in one step, all the results are added to the previous carry.

Manuscript published on 30 December 2015.
* Correspondence Author (s)
Shazeeda*, School Of Electrical & Electronic Engineering, Universiti Sains Malaysia (USM), Engineering Campus, Seberang Perai Selatan, Nibong Tebal, Penang - 14300, Malaysia.
Monika Sharma D, Department of Electronics and Communication, SJM Institute of Technology (SJMIT), Chitradurga - 577502, Karnataka, India.

The least significant digit of the number thus obtained acts as one of the result digits and the rest act as the carry for the next step. Initially the carry is taken to be as zero [9]. Transistor level implementation for performance parameters such as propagation delay, dynamic leakage power and dynamic switching power consumption calculation of the proposedmethod was calculated and compared with the other design like Wallace Tree Multiplier (WTM) [10].

The algorithm for a 4×4 multiplier using Urdhva tiryakbhyam sutra is as depicted in Figure 1.

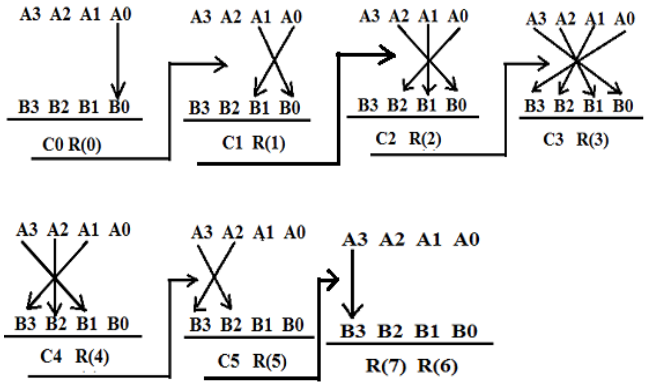

Figure 1. Algorithm of urdhava tiryakbhyam sutra [1]

Equations used for designing 4×4 Vedic multiplier
R(0) = A0.B0
R(1) = A1.B0 +A0.B1+C0(Carry Of R(0))
R(2) = A2.B0 +A0.B2+A1.B1+C1(Carry Of R(1))
R(3) = A3.B0 +A0.B3+A2.B1+A1.B2+C2(Carry Of R(2))
R(4) = A3.B1 +A1.B3+A2.B2+C3(Carry Of R(3))
R(5) = A2.B0 +A0.B2+C4(Carry Of R(4))
R(6) = A2.B0 +C5(Carry Of R(5))
R(7) = C6(Carry Of R(6))

B. 2.1. Alternative multiplication method by using Urdhvatiryakbhyam Sutra

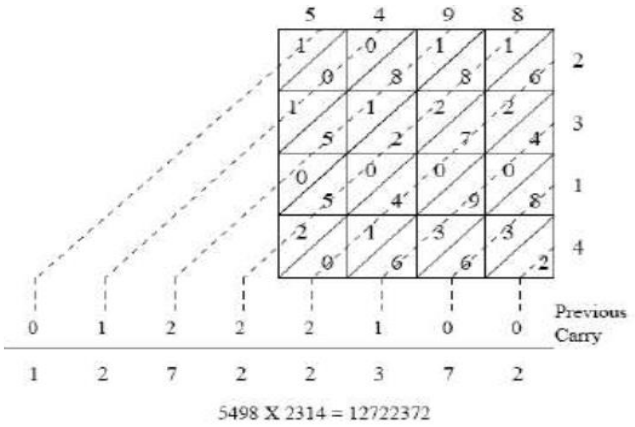

Figure 2 Multiplication of two large integers [6]

C. Mathematical background of Urdhvatiryakbyham sutra

Assume that X and Y are two numbers, to be multiplied. Mathematically X and Y can be represented as

$$A = \sum_{i=0}^{N-1} A_i 10^i$$

.....(1)

$$B = \sum_{j=0}^{N-1} B_j 10^j$$

..... .(2)

Assume that, their product is equal to Z. Then Z can be represented as

$$Z = \left(\sum_{i=0}^{N-1} A_i \sum_{j=0}^{N-1} B_j\right) 10^{i+j}$$

..... .(3)

Where (A_i , B_j € (0,1,2,.....,9) and 'N' may be any number. From the above expression, equation 3, it can observed that each digit is multiplied consecutively and shifted towards the proper positions for partial product generation. Finally the partial products are added with the previous carry to produce the final results [10].

III. WALLACE TREE MULTIPLIER

In a modern day CPU, the multiplier has become an important part of an ALU in terms of both power and performance. Modern day multiplier designs provide some enhancements in power and performance over traditional multiplier designs.

IV. MULTIPLICATION ALGORITHM

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). **This applies to papers in data storage.** For example, write "15 Gb/cm^2 (100 Gb/in^2)." An exception is when English units are used as identifiers in trade, such as "3½ in disk drive." Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown in figure 3

Y= Yn-1 Yn-2 .....Y2 Y1 Y0 Multiplicand
X= Xn-1 Xn-2 .....X2 X1 X0 Multiplier

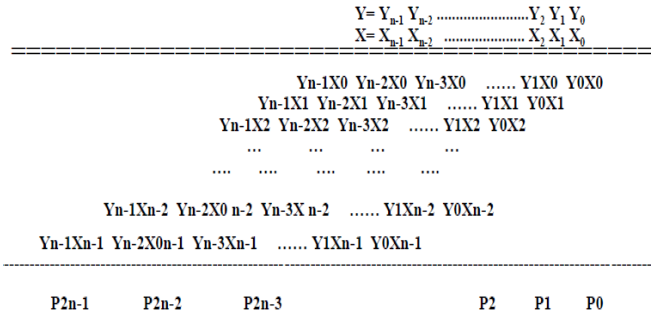

Figure 3: Multiplication algorithm

AND gates are used to generate the Partial Products, PP, If the multiplicand is N-bits and the Multiplier is M-bits then there is N* M partial product. The way that the partial products are generated or summed up is the difference between the different architectures of various multipliers. Multiplication of binary numbers can be decomposed into additions.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Consider the multiplication of two 8-bit numbers A and B to generate the 16 bit product P.

- If the LSB of Multiplier is '1', then add the multiplicand into an accumulator.
- Shift the multiplier one bit to the right and multiplicand one bit to the left.
- Stop when all bits of the multiplier are zero.

Multiplication of binary numbers can be decomposed into additions. Consider the multiplication of two 8-bit numbers A and B to generate the 16 bit product P.

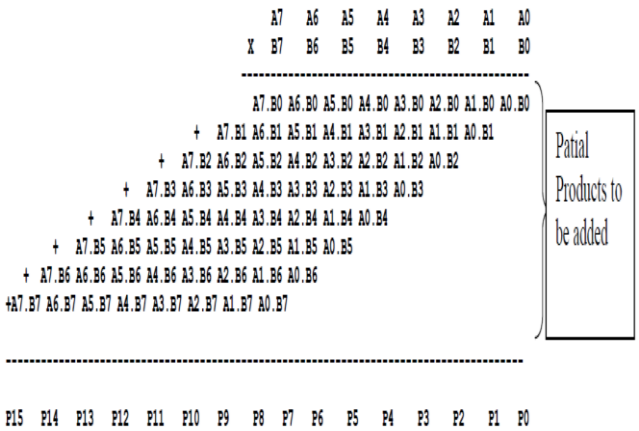

Figure 4 Multiplication of two 8 bit numbers

The general multiplication formula that is used

$$P(m + n) = A(m)\,B(n) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} a_i b_j 2^{i+j} \quad \text{.....(4)}$$

From above it is clear that the multiplication has been changed to addition of numbers. If the Partial Products are added serially then a serial adder is used with least hardware. It is possible to add all the partial products with one combinational circuit using a parallel multiplier. However it is possible to use compression technique then the number of partial products can be reduced before addition is performed. Today's high performance microprocessors must have the capability of supporting fast floating point and integer calculations. In order to support various applications environments. These microprocessors must perform multiply operations, using operands that have a 32 or 64-bit word length, at high clock rates. Typically, these microprocessors rely upon multiplier arrays to perform a set of floating point, integer and graphic multiply instructions. Generally, the number of circuit elements in a multiplier is proportional to N (where N is the word length in bits). Thus, one critical factor affecting implementation of a multiplier is the global layout considerations of the circuit elements. Another critical factor is the speed at which the multiplier performs the multiply operation. Since tree multipliers have a delay proportional to log(N), they are preferable in terms of performance to array multipliers, whose delay is N^2 proportional to N. Tree multipliers require large shifts of data perpendicular to the data path, therefore, implementation of tree multipliers is routing intensive. Thus, even though tree multipliers offer speed advantages over array multipliers, microprocessor designers have traditionally avoided using tree multipliers due to the circuit area required for their implementation. It is, therefore, desirable to provide a tree multiplier floorplan which reduces the circuit area required for its implementation, while still providing the capability to perform high speed calculations.

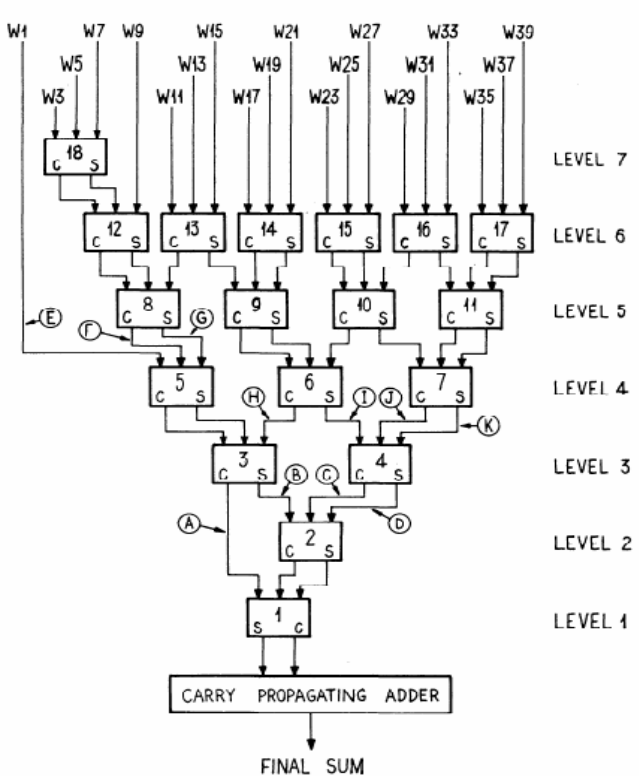

Figure 5. Block diagram of Wallace tree multiplier

The Wallace multiplier is an improvement of the Array multiplier in order to reduce the critical path of the multiplier. It is based on the premise of trying to combining the partial products as quickly as possible, instead of waiting for less significant bits to generate a carry signal. A Wallace multiplier which consists of an optimized carry propagate adder. Figure 5 shown above illustrates the combination of each level in a Wallace tree for one branch of the Wallace tree. These levels are referred to as reductions, for each level takes 3 inputs (A, B, Carry-in), and reduces it to 2 outputs (Sum and Carry-out). Due to the fact that the Carry signal in a Wallace tree propagates down each level, the critical path in a Wallace tree is significantly reduced. In an Array multiplication scheme, the carry signal must propagate across all N bits of both the multiplicand and multiplier. In a Wallace scheme, because the partial products are grouped by bit-weight, much of the initial addition occurs in parallel. The number of levels required depends on the bit-weight group with the largest number of partial products and is logarithmically related to the number of bits being multiplied.

The delay through the Wallace tree is equal to the delay of a full adder times the number of levels. The final Carry Propagate adder is approximately 0.5*N bits in size due to the reduction that occurred in the Wallace tree. Thus the delay through a Wallace multiplier is roughly approximated by 0.5*N + log (N). Due to the less uniform arrangement of adders however, the Wallace multiplier uses a slightly greater number of full adders to perform its task.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

This is because of the fact that the grouping of partial products by their bit weights often ends up not completely utilizing a full adder. Often, a half adder is used for these cases, which occurs in about 1/3 of the bit weights. Despite this, the switching power for a Wallace multiplier is approximately the same as that of an array multiplier [11].

V. DISCRETE COSINE TRANSFORM

DCT exploits cosine functions, it transform a signal from spatial representation into frequency domain. The DCT represents an image as a sum of sinusoids of varying magnitudes and frequencies. DCT has the property that, for a typical image most of the visually significant information about an image is concentrated in just few coefficients of DCT. After the computation of DCT coefficients, they are normalized according to a quantization table with different scales provided by the JPEG standard computed by psycho visual evidence. Selection of quantization table affects the entropy and compression ratio. The value of quantization is inversely proportional to quality of reconstructed image, better mean square error and better compression ratio. In a lossy compression technique, during a step called Quantization, the less important frequencies are discarded, then the most important frequencies that remain are used to retrieve the image in decomposition process. After quantization, quantized coefficients are rearranged in a zigzag order for further compressed by an efficient lossy coding algorithm. DCT has many advantages:

- It has the ability to pack most information in fewest coefficients.
- It minimizes the block like appearance called blocking artifact that results when boundaries between sub-images become visible [10].

The DCT helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality).

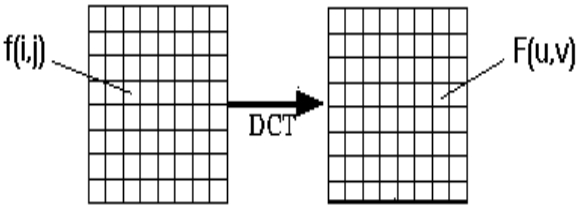

Figure 6 Representation of a DCT

A. DCT Encoding

The general equation for a 1D (*N* data items) DCT is defined by the following equation

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \Lambda(i).cos\left[\frac{\pi.u}{2.N}(2i+1)\right] f(i)$$

……(5)

and the corresponding inverse 1D DCT transform is simple $F^{-1}(u)$, i.e.:

where

$$\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}} & for \xi = 0 \\ 1 & otherwise \end{cases}$$

……(6)

The basic operation of the DCT is as follows

- The input image is N by M.
- f(i,j) is the intensity of the pixel in row i and column j.
- F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix.
- For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.
- Compression is achieved since the lower right values represent higher frequencies, and are often small enough to be neglected with little visible distortion.
- The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level.
- 8 bit pixels have levels from 0 to 255.

B. Architecture of DCT

Calculation of DCT co-efficients can be done by using the following matrices. Matrixrepresentation of the Forward DCT is as shown below.

$$X = \begin{bmatrix} X0 \\ X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \end{bmatrix} \quad R = \begin{bmatrix} A & A & A & A & A & A & A & A \\ D & E & F & G & -G & -F & -E & -D \\ B & C & -C & -B & -B & -C & C & B \\ E & -G & -D & -F & F & D & G & -E \\ A & -A & -A & A & A & -A & -A & A \\ F & -D & G & E & -E & -G & D & -F \\ C & -B & B & -C & -C & B & -B & C \\ G & -F & E & -D & D & -E & F & -G \end{bmatrix} \quad x = \begin{bmatrix} x0 \\ x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \end{bmatrix}$$

X = R .x for Forward DCT

x= R^T . X for Inverse DCT

The values A, B, C, D, E, F, G of the matrix M were extracted from the general formula that describes the 1 -D DCT.

$$G(y) = \sum_{v=0}^{7} F(v) \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

.....(7)

Where

G(y) is the calculated DCT coefficient

F(v) is the input data value

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

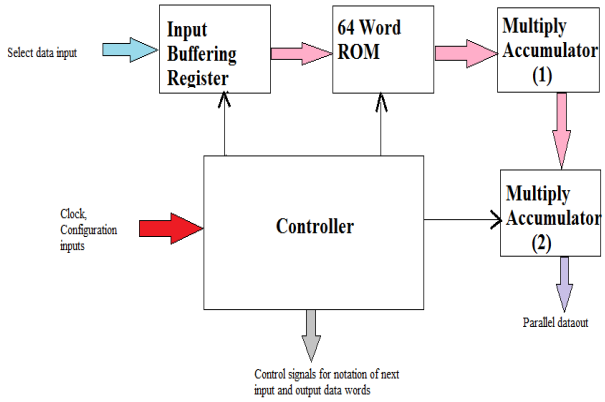

Figure 7 Block diagram of Discrete Cosine Transform [7].

C. Multiply Accumulator

In computing, especially digital signal processing, the multiply–accumulate operation is a common step that computes the product of two numbers and adds that product to an accumulator. The hardware unit that performs the operation is known as a multiplier accumulator (MAC, or MAC unit); the operation itself is also often called a MAC operation.

The MAC unit provides functionality in three related areas

- Signed and unsigned integer multiplies.
- Multiply-accumulate operations supporting signed, unsigned, and signed fractionaloperands.
- Miscellaneous register operations.

In this paper, it is proposed that the vedic multiplier is employed in the accumulator of DCT. As the Vedic multiplier is having the reduced delay it is used in many DSP applications like DCT/IDCT. Hence Vedic multiplier using Urdhva tiryakbhyam sutra is used for designing DCT/IDCT. DCT can be implemented in image processing. It helps separate the image into two parts of differing importance with respect to the image visual quality.

VI. RESULTS AND DISCUSSIO

A. Vedic multiplier

i. RTL schematic of 4bit Vedic multiplier

The Figure 8 shows the RTL schematic of 4bit Vedic multiplier

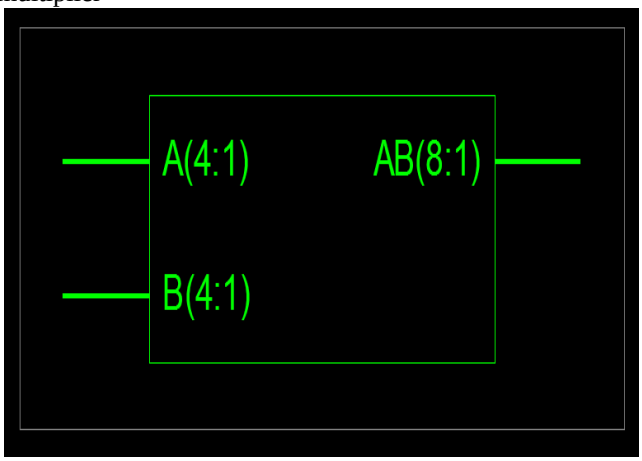

Figure 8.RTL schematic of 4bit Vedic multiplier

ii. Simulation waveform of 4bit Vedic multiplier

The Figure 9 shows the simulation waveform of 4bit Vedic multiplier. The inputs a andb are of 4 bits. As it is shown in the simulation waveform if input a is 10 and input b is 4 then the output ab obtained is 40.

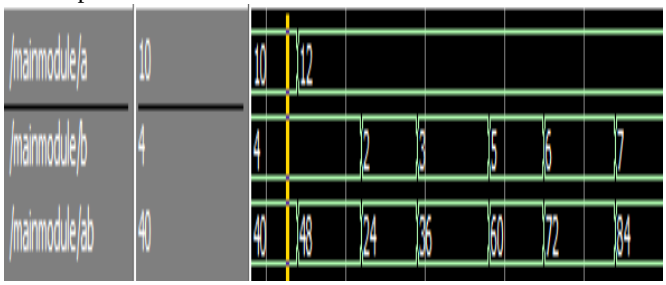

Figure 9 Simulation waveform of 4bit Vedic multiplier

iii. RTL schematic of 8bit Vedic multiplier

The Figure 10 shows the RTL schematic of 8bit Vedic multiplier

Figure 10.RTL schematic of 8bit Vedic multiplier

iv. **Simulation waveform of 8bit Vedic multiplier**

The Figure 11 shows the simulation waveform of 8bit Vedic multiplier. The inputs a andb are of 8 bits. As it is shown in the simulation waveform if input a is 255 and input b is 255 then the output ab obtained is 65025.

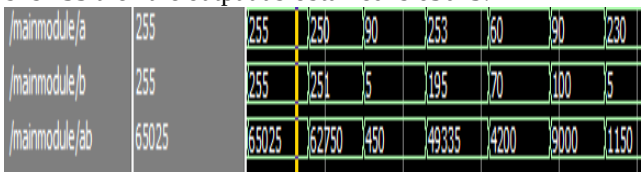

Figure 11.Simulation waveform of 8bit Vedic multiplier

v. RTL schematic of 512 bit Vedic multiplier

The Figure 12 shows the RTL schematic of 512bit Vedic multiplier

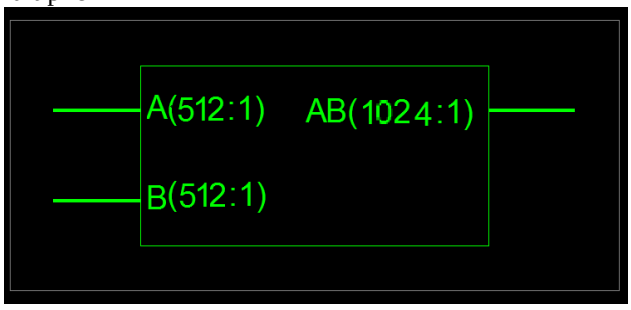

Figure12. RTL schematic of 512bit Vedic multiplier

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

vi. Simulation waveform of 512bit Vedic multiplier

The Figure 13 shows the simulation waveform of 512bit Vedic multiplier. The inputs a andb are of 512 bits. As it is shown in the simulation waveform if input a is 10000 and input b is 5000 then the output ab obtained is 50000000.

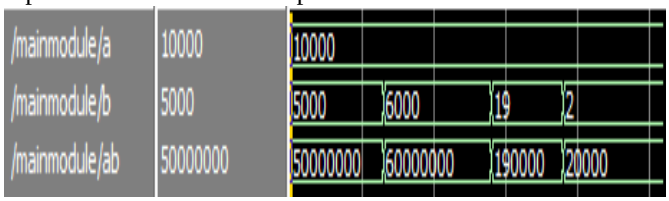

Figure 13Simulation waveform of 512bit Vedic multiplier

The above shows the result of 4,8,512 bit Vedic multiplier. This Vedic multiplier is implemented using Urdhva tiryakbhyaam sutra. N-bit Vedic multiplier is designed and is implemented.

B. Wallace tree multiplier

i. RTL schematic of 4bit Wallace tree multiplier

The Figure 14 shows the RTL schematic of 4bit Wallace tree multiplier

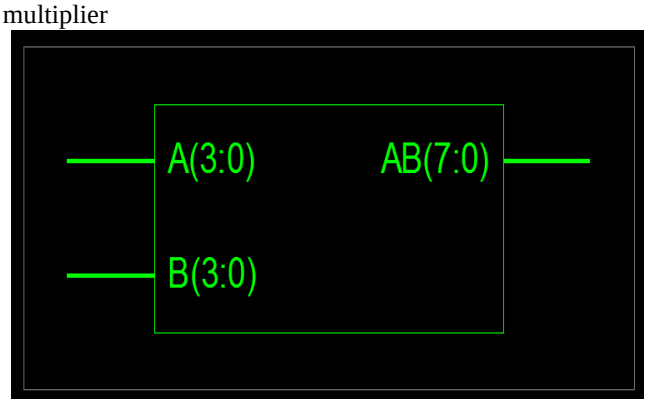

Figure 14: RTL schematic of 4bit Wallace tree multiplier

ii. Simulation waveform of 4bit Wallace tree multiplier

The Figure 15 shows the simulation waveform of 4bit Wallace tree multiplier. The inputs a andb are of 4 bits. As it is shown in the simulation waveform if input a is 4 and input b is 5 then the output ab obtained is 20.

Figure 15 Simulation waveform of 4bit Wallace tree multiplier

iii. RTL schematic of 8 bit Wallace tree multiplier

The Figure 16 shows the RTL schematic of 8bit Wallace tree multiplier

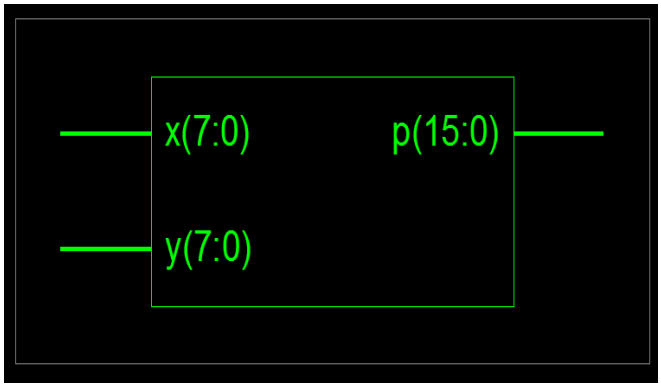

Figure 16 RTL schematic of 8bit Wallace tree multiplier

iv. Simulation waveform of 8bit Wallace tree multiplier

The Figure 17 shows the simulation waveform of 8bit Wallace tree multiplier. The inputs x and y are of 8 bits. As it is shown in the simulation waveform if input x is 146 and input y is 127 then the output xy obtained is 18542.

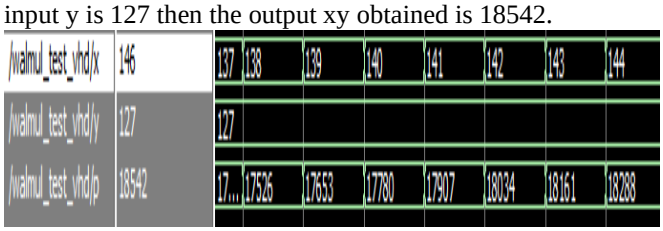

Figure 17 Simulation waveform of 8bit Wallace tree multiplier

The above figure shows the results of 4,8bit Wallace tree multiplier. Delay is estimated for 4,8bit Wallace tree multiplier. The obtained delay is compared with that of 4,8bit Vedic multiplier.

a. Comparison of the propagation delay of 4, 8 bit Vedic multiplier and Wallace tree multiplier

Table1. Comparison of propagation delay

Type of multiplier	Number of bits	Propagation delay
Vedic multiplier	4	10.025ns
	8	25.236ns
Wallace tree multiplier	4	16.104ns
	8	36.904ns

The delay obtained from the 4,8bit Vedic multiplier is compared with that of the 4,8bit Wallace tree multiplier. The delay obtained from Wallace tree multiplier is high when compared with the Vedic multiplier. Hence Vedic multiplier is the efficient multiplier since it has the reduced delay. The computation time of the Vedic multiplier is less when compared to Wallace tree multiplier. This Vedic multiplier in which multiplication is carried out by using the Urdhva tiryakbhyam sutra is used in many DSP applications like DCT/IDCT. Hence DCT/IDCT using this Vedic multiplier is designed and is implemented.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

b. Discrete Cosine Transform (DCT)

i. RTL schematic of DCT

The Figure 18 shows the RTL schematic of Discrete Cosine Transform

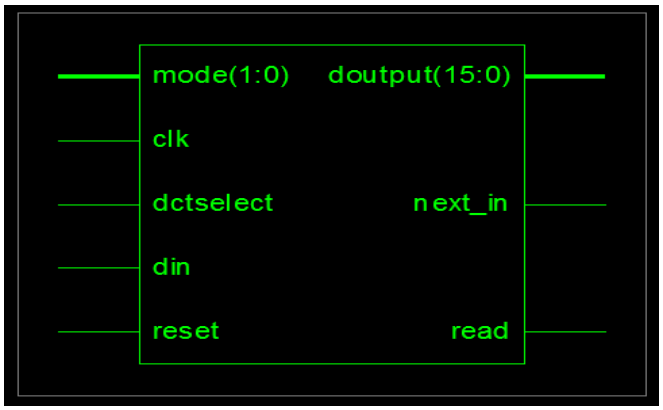

Figure 18 RTL schematic of DCT

ii. Simulation waveform of 12bit adder

Figure 19 shows the simulation waveform of 12bit adder. Both inputs a andb are of 12 bits. Adder performs the addition operation by adding two numbers. If the given input a is 100 and b is 500 then the obtained output is 600.

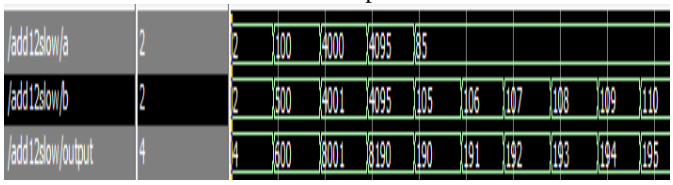

Figure 19 Simulation waveform of 12bit adder

iii. Simulation waveform of 16bit adder

Figure 20 shows the simulation waveform of 16bit adder. Both inputs a andb are of 16 bits. Adder performs the addition operation by adding two numbers. If the given input a is 200 and b is 500 then the obtained output is 700.

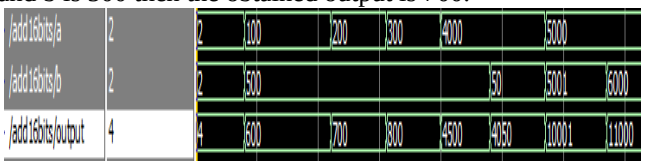

Figure 20.Simulation waveform of 16bit adder

iv. Simulation waveform of AND gate

Figure 21 shows the simulation waveform of AND gate. If enable is high, input value will be obtained at the output, if enable is low (0) the output will be zero.

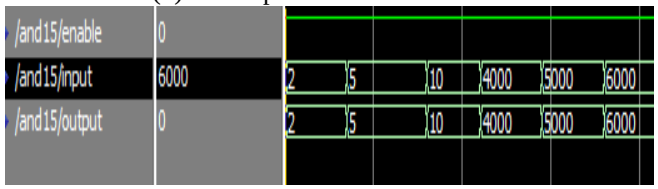

Figure 21Simulation waveform of AND gate

v. Simulation waveform of cycle register

Figure 22 shows the simulation waveform of cycle register.

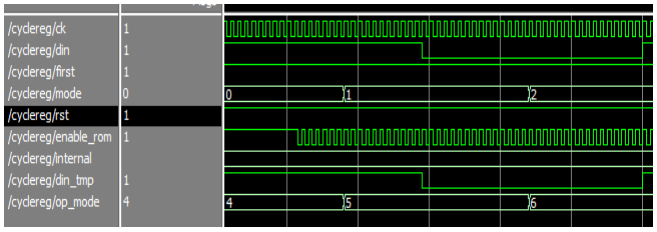

Figure 22 Simulation waveform of cycle register

vi. Simulation waveform of controller

Figure 23 shows the simulation waveform of controller.

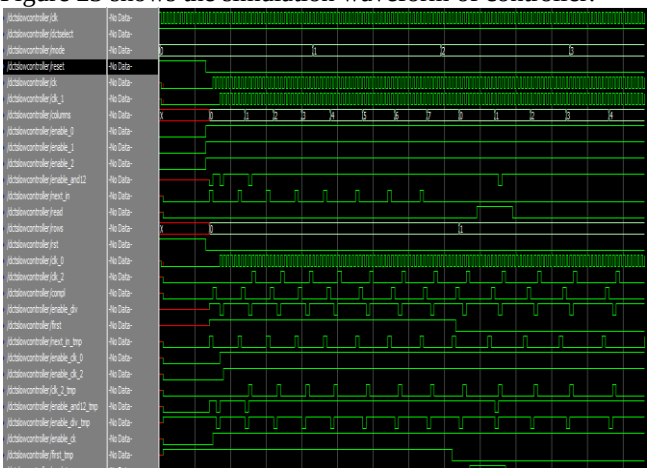

Figure 23 Simulation waveform of controller

vii. Simulation waveform of ROM

Figure 24 shows the simulation waveform of ROM

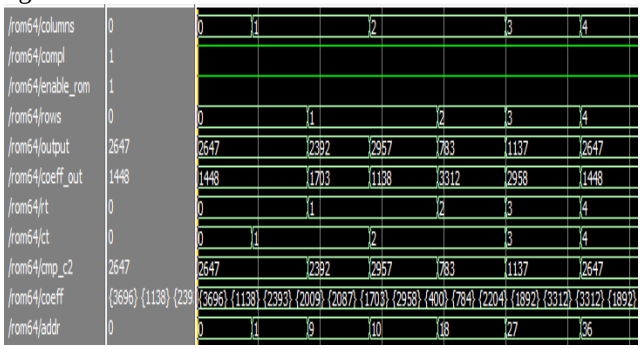

Figure 24 Simulation waveform of ROM

viii. Simulation waveform of division

Figure 25 shows the simulation waveform of division

Figure 25 Simulation waveform of division

ix. Simulation waveform of 12bits register

Figure 26 shows the simulation waveform of 12bits register.ain is the input, clk is given high and if enable is also high then the input will be produced at the output. If enable is low the output will be zero.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

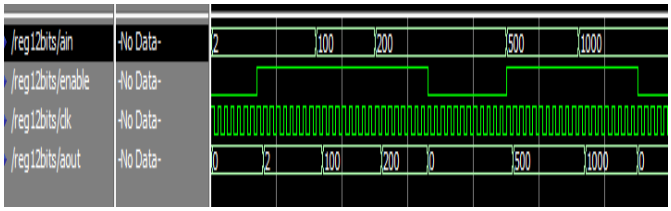

Figure 26 Simulation waveform of 12bits register

x. Simulation waveform of 13bits register

Figure 27 shows the simulation waveform of 13bits register.ain is the input, clk is given high and if enable is also high then the input will be produced at the output. If enable is low the output will be zero.

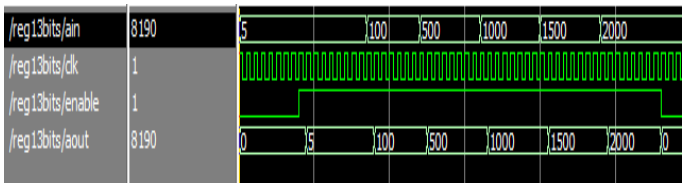

Figure 27 Simulation waveform of 13bits register

xi. Simulation waveform of 16bits register

Figure 28 shows the simulation waveform of 16bits register.ain is the input, clk is given high and if enable is also high then the input will be produced at the output. If enable is low the output will be zero.

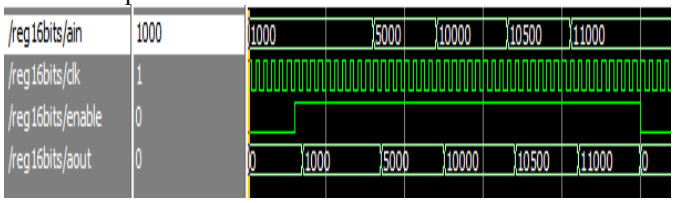

Figure 28 Simulation waveform of 16bits register

4,8bit Vedic multiplier is designed, implemented. 4,8bit Vedic multiplier propagation delay is estimated.4,8bit Wallace tree multiplier is designed and implemented. Its propagation delay is estimated. The delay obtained from the 4,8bit Vedic multiplier is compared with that of the Wallace tree multiplier. The obtained in the Vedic multiplier is less when compared to thet of the Wallace tree multiplier. Hence it is concluded that Vedic multiplier is the efficient multiplier. As Vedic multiplier is the efficient multiplier N-bit Vedic multiplier is designed and implemented. As this Vedic multiplier is having the reduced dealy it is used in many DSP applications like DCT/IDCT. Hence Vedic multiplier using Urdhvatiryakbhyam sutra is used for designing DCT/IDCT. DCT is used in image processing. It helps separate the image into two parts of differing importance with respect to the image visual quality.

VII. CONCLUSION

In this paper we proposed a fast multiplication method based on ancient Indian Vedic mathematics. It is a generic method based on N-bit Vedic Multiplier which is implemented in the digital signal processing. The proposed 4,8bit Wallace tree multiplier is compared with that of Vedic multiplier in terms of the delay. The delay that is obtained from the Wallace tree multiplier is larger when compared to that of the Vedic multiplier. Thus, findings of the present study suggested that the Vedic multiplier is efficient multiplier and useful in the digital signal processing applications like DCT/IDCT.

ACKNOWLEDGMENT

First author would like to express thanks and gratitude to Universiti Sains Malaysia (USM) for USM fellowship award.

REFERENCES

1. Ancient Indian Vedic Mathematics based 32-Bit Multiplier Design for High Speed and Low Power Processors, Nishant G. Deshpande, RashmiMahajan, International Journal of Computer Applications (0975 – 8887) Volume 95– No.24, June 2014.
2. Haveliya, Asmita. "A Novel Design for High Speed Multiplier for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach)."International Journal of Technology and Engineering System (IJTES) 2, no. 1 (2011): 27-31.
3. Purushottam D. Chidgupkar ,Mangesh T. Karad "The Implementation of Vedic Algorithms in Digital Signal Processing" , Vol.8, 2004.
4. Badal Sharma, "Design and Hardware Implementation of 128-bit Vedic Multiplier" International Journal for Advance Research in Engineering and Technology, Vol.3, 2013.
5. Saha P., A. Banerjee, A. Dandapat, P. Bhattacharyya, "Vedic Mathematics Based 32-Bit Multiplier Design for High Speed Low Power Processors" International Journal on Smart Sensing and Intelligent Systems, Vol. 4, 2011.
6. Kavita, UmeshGoyal "Performance Analysis of Various Vedic Techniques for Multiplication" International Journal of Engineering Trends and Technology, Vol.4, 2013.
7. Sherif T. EID, "VLSI Design and Implementation of Different DCT Architectures for Image Compression", 1999-2000.
8. N.J.R. Muniraj,N.Senathipathi, "High Speed DCT Design Using Vedic Mathematics",2011.
9. Badal Sharma, "Design and Hardware Implementation of 128-bit Vedic Multiplier" International Journal for Advance Research in Engineering and Technology, Vol.3, 2013.
10. P. Saha, A. Banerjee, A. Dandapat, P. Bhattacharyya, "Vedic Mathematics Based 32-Bit Multiplier Design for High Speed Low Power Processors" International Journal on Smart Sensing and Intelligent Systems, Vol. 4, 2011.
11. Michael Moeng, Jason Wei, "Optimizing Multipliers for the CPU: A ROM Based Approach" Electrical Engineering and Computer Science University of California: Berkeley.
12. NavpreetSaroya, PrabhpreetKaur, "Analysis of Image Compression Algorithm Using DCT and DWT Transforms", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4, 2014.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)