HIGH SPEED MULTIPLIER USING VEDIC MATHEMATICS

Uttara Bhatt[1], Deepak Bhoir[2], K. Narayanan[3]

[1, 2, 3]*Department of Electronics Engineering, Fr. Conceicao Rodrigues College of Engineering, Mumbai, India*

Abstract

The digital signal processing in today's time need high speed computation. The basic building block of signal processing in Communication, Biomedical signal processing, and Image processing remains Fast Fourier Transform (FFT). FFT computation involves multiplications and additions. Speed of the DSP processor mainly depends on the speed of the multiplier. Time delay, power dissipation and the silicon chip area. These are the most important parameters for the fast growing technology. The conventional multiplication method requires more time and area and hence more power dissipation. In this paper an ancient Vedic multiplication method called "Urdhva Triyakbhyam" is implemented. It is a method based on 16 sutras of Vedic mathematics. Vedic Mathematics reduces the number of operations to be carried out compared to the conventional method. The code description is simulated and synthesized using FPGA device Spartan XC3S400-PQ208.

Keywords— *Vedic Multiplication, Urdhva Tiryakbhayam , FFT*

----------***----------

1. INTRODUCTION

In communication, while processing out any data, analysis done in frequency domain is more preferred. The Discrete Fourier Transform requires N^2 multiplication for an N point data. Fast Fourier Transform has two algorithms namely decimation in Time and decimation in Frequency commonly known as DIT-FFT and DIF-FFT. Here the computational complexity reduces from N^2 to $N \log_2 N$ [1]. Multiplication is one of the silicon-intensive functions, especially when implemented in Programmable Logic. Multipliers are key components of many high performance systems such as Digital Signal Processors and Microprocessors [2].

A system's performance is generally determined by the performance of the multiplier. Hence optimizing the speed and area of the multiplier is a major issue. This paper presents High Speed Multiplier implementation with an algorithm which reduces the propogation delay [3].This optimizes the speed. The algorithm is based on Urdhva Triyakbham sutra [4] of Vedic mathematics.

1.1 Vedic Mathematics

Vedic mathematics is the ancient Indian system of mathematics which mainly deals with Vedic mathematical and their application to various branches of mathematics. The word 'Vedic' is derived from the word 'Veda' which means the store-house of knowledge. Vedic mathematics was reconstructed from the ancient Indian scriptures (Vedas) by Shri Bharati Krashna Tirthaji (1884-1960), after his great research on Vedas [4]. According to his research, Vedic mathematics is mainly based on sixteen principles or word formulae which are termed as Sutras. The multiplication is been carried out using Vedic mathematics using the sutra 14 i.e. Urdhva Triyakbham as follows [4]. Fig.1 explains the line diagram for the multiplication of two numbers.

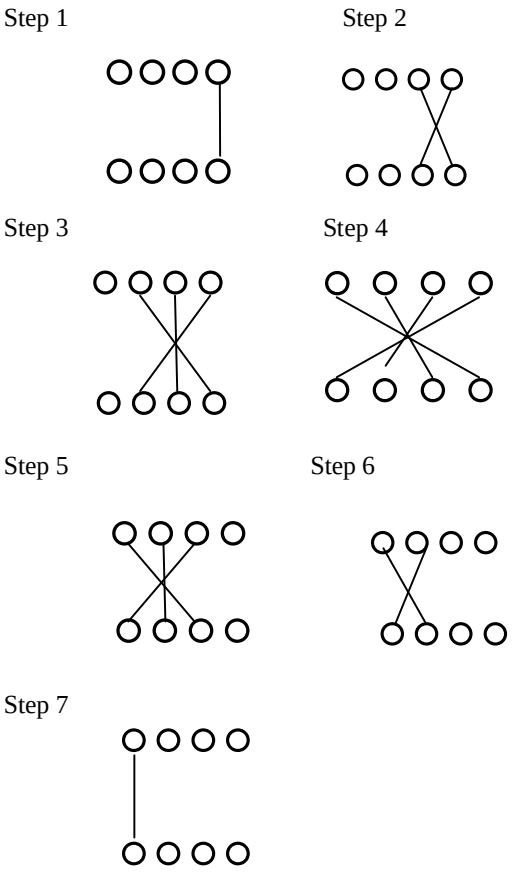
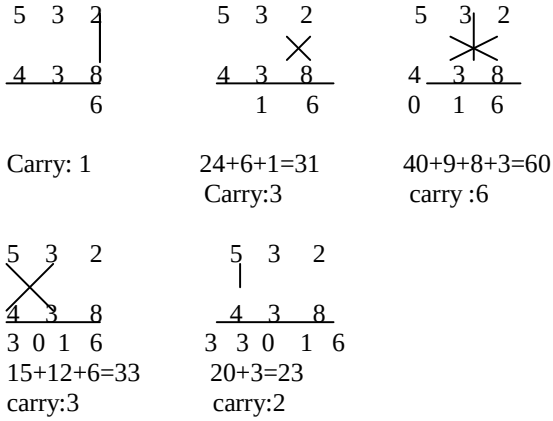


Fig.1 Line Diagram of 4 bit multiplier

The example of multiplication of two numbers using the line diagram is explained in the Fig.2.

Vedic mathematics is very flexible and highly efficient approach to mathematics. The Vedic mathematics is more coherent approach and it can be used as a mental exercise.



The final answer = 233016

Fig 2 Example of Urdhva Triyakbham Sutra

1.2 Fast Fourier Transform

A Fast Fourier transform (FFT) is an algorithm to compute the Discrete Fourier Transform (DFT) and the inverse Fourier Transform [1]. Basically, the computational problem for the DFT is to compute the sequence {X(k)} of N complex-valued numbers given another sequence of data {x(n)} of length N, according to the formula [1]. Using the twiddle factor in the formula , the DFT of the sequence in time domain given by x(n) with length N and k discrete frequency components ,is given by,

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad k=0,1,2\ldots.N-1$$

Similarly, the IDFT becomes,

$$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(k) W_N^{-nk} \qquad n=0,1,2\ldots.N-1$$

Since DFT and IDFT involve basically the same type of computations, the discussion of efficient computational algorithms for the DFT applies as well to the efficient computation of the IDFT.

It is observed that for each value of k, direct computation of X(k) involves N complex multiplications (4N real multiplications) and N-1 complex additions (4N-2 real additions). Consequently, to compute all N values of the DFT requires N 2 complex multiplications and N 2-N complex additions [1]. Direct computation of the DFT is basically inefficient primarily because it does not exploit the symmetry and periodicity properties of the phase factor WN.

There are different algorithms for FFT namely Decimation in time and Decimation in frequency more commonly called as DIT-FFT and DIF-FFT respectively. Fig.3 is the butterfly diagram for two point FFT. This two point FFT butterfly diagram is the basic or the smallest unit of bigger N point FFT diagram.

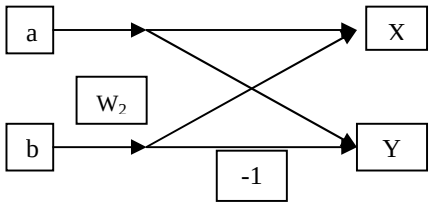


Fig.3 Two point Butterfly Diagram

For DIT FFT algorithm for N is 8, there are four such two point butterfly diagrams in the first stage. The second stage has two 4-point diagrams. And the third stage has one 8-point butterfly diagram. For DIF-FFT algorithm the order is exactly opposite to that of DIT-FFT [1].

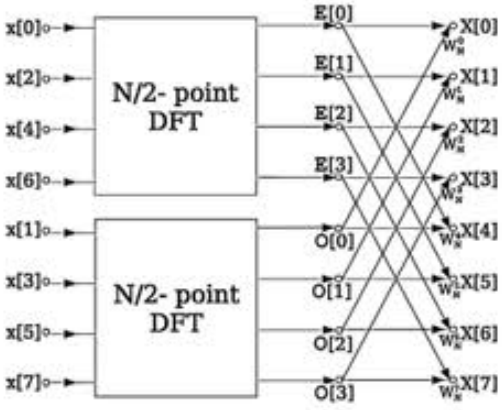


Fig.4 Eight point FFT Diagram

The number of computation in DFT are N2 multiplication and N(N-1) additions. For a FFT, if there are N points or samples in the signal in the time domain, then there are L stages available in the FFT butterfly diagram so that N=2L This means for each stage, there are N multiplications. Thus total number of multiplications are N*L. This means total number of multiplications are NLog2N for example: for 8 point DFT, the number of multiplications are 64 and additions are 56. While for FFT the multiplications are reduced to 24.

Speed of the DSP processor mainly depends on the speed of the multiplier [1]. Silicon chip area, Time delay and power dissipation are the most important parameters for the fast growing technology [2]. The normal multiplication method requires more time and area and hence more power dissipation. Therefore our aim is to reduce the computational complexity further. Thus here is an attempt to make a high speed multiplier [5] using Vedic mathematics.

1.3 Wallace Tree Method

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, devised by an Australian Computer Scientist Chris Wallace in 1964 [7].

The Wallace tree has three steps:
(1) Multiply (that is - AND) each bit of one of the arguments, by each bit of the other.
(2)Reduce the number of partial products to two by layers of full and half adders.
(3)Group the wires in two numbers, and add them with a conventional adder.

The Wallace Tree is used with the Carry Look Ahead Adder [8] as the last stage.

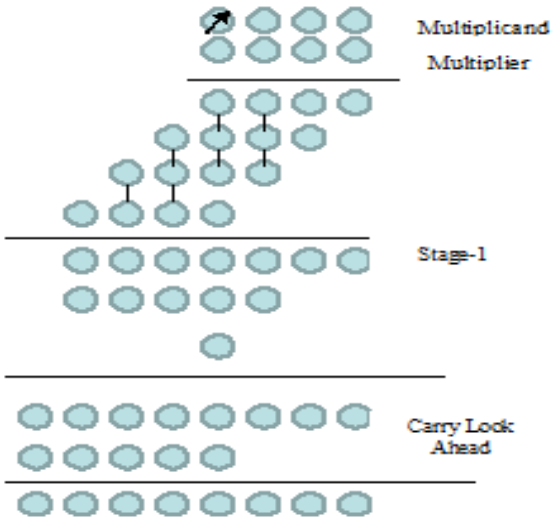


Fig. 5 Wallace Tree for 4 digits Multiplication

The Wallace Tree multiplier consists of two stages. Stage 1 carries out the normal Wallace Tree multiplication. And the stage 2 involves the Carry Look Ahead adder [8]. Booth Wallace tree is also an alternative method for multiplication [7]. Here N digit multiplication can be carried out .The addition of the terms at the end are carried out columnwise. It is carried out such a way that carry generated in the every column in the first stage is taken to the next column in the second stage. Same way, the carry generated in every column in the second stage is taken to the next column in the third stage and so on. This is continued till the final answer is obtained.

2. VEDIC APPROACH VS OTHER APPROACH

First the computations required for the normal and the Vedic multiplication algorithm [6] are carried out. The total number of computations required for the multiplication of two decimal numbers [3] of different bit stream size using the Vedic and normal approach are calculated and given in Table.1. This table gives the computations required for numbers of 2bit, 3 bit and 4 bit respectively.

Table 1 Computations for the Vedic & normal approach

	Normal Method	**Vedic Method**
2 bit multiplication	6	5
3 bit multiplication	16	14
4 bit multiplication	31	25

The multiplication of two 4 digit numbers using Wallace Tree [7] and Vedic mathematics are carried out. The Wallace Tree used here involves carry look ahead adder. This makes Wallace Tree also faster.With this, the adders used in both the approaches are computed .The same is shown in the Table.2.

Table 2 Adders used in Vedic and Booth Wallace approach

Method	Number of half adder	Number of full adder
Vedic Approach	4	5
Wallace Tree	10	4 + carry look ahead adder

Algorithm for Vedic Multiplication

The multiplication of two numbers (4 bits) is carried out as follows.
[a0-a3] – Multiplicand
[b0-b3] -Multiplier
(1) Multiply the LSBs that gives ,$P_0 = a_0b_0$
(2) Multiply LSB column with the next column in cross fashion,

$$P_1 = a_0b_1 + a_1b_0$$

(3) Multiply 1st and 3rd column crosswise and second column

$$P_2 = a_0b_2 + a_2b_0 + a_1b_1$$

(4) 1st and 4th columns & 2nd and 3rd columns are crossed multiplied.

$$P_3 = a_0b_3 + a_3b_0 + a_1b_2 + a_2b_1$$

(5) Repeat the same starting with the second and get P_4 & P5
(6) Repeat the same for the last column and get P_6
(7) Pad zeros to get all the equations 8 terms
(8) Add all the four columns to get the final answer

The code for the Vedic approach is written in VHDL.The same is synthesized using Xilinx FPGA Device XC3S400- Spartan-3 Family-Package-PQ208.

The Table.2 indicates that Vedic algorithm uses lesser hardware. Hence the 4 bit multiplication using Vedic algorithm [9], [10] and Wallace Tree are carried out in VHDL [11]. The synthesis report gives the delay and the hardware used in implementing the same. The comparison is shown in Table 3.

Table 3 Comparison of Vedic and Booth Wallace approach

Algorithm	Vedic	Wallace Tree
Delay	12.485 ns	13.141 ns
Cell Usage	114	119
Number of LUTs	21	34
Number of IOs	16	17
Number of Slices	11	19
Number of IOBs	16	17

The code for Vedic approach & Wallace Tree approach are synthesized using Xilinx FPGA Device XC3S400- Spartan-3 Family-Package-PQ208.The comparison is as shown in the Table.3 . It shows that the Vedic algorithm offers lesser delay and hardware. The delay per multiplier is reduced by 0.656ns.This reduction is 5% Hence this faster Vedic multiplier is simulated in VHDL. The multiplier simulation results are shown in the Fig.6 and Fig.7

3. RESULTS & DISCUSSION

The Vedic algorithm is used to implement 4 bit and 8 bit multiplier. Multiplication of 1010 and 1011 is carried out .This 4 bit multiplication simulation is shown in the Fig.5.The answer is 01101110.

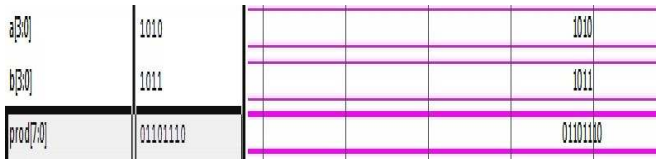


Fig.6 Simulation of 8 bit Multiplier

Multiplication of 11110000 and 11001100 is carried out .This 8 bit multiplication simulation is shown in the Fig.6.The answer is 1011111101000000.

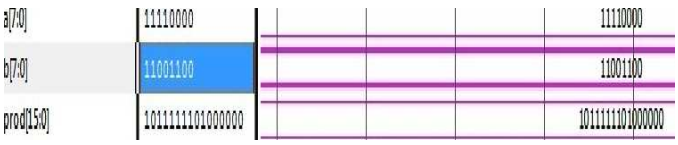


Fig.7 Simulation of 8 bit Multiplier

Table 3 indicates that the propagation delay in the Wallace Tree Algorithm is greater than the Wallace Tree algorithm. The same is indicated in the bar diagram Fig.7

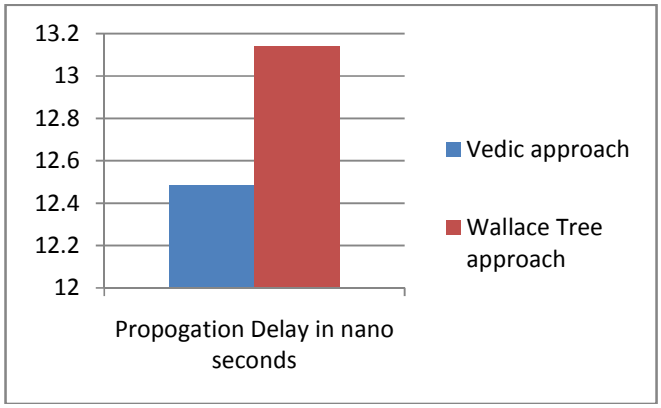


Fig.7 Comparison of the Propagation Delay

Fig 8 indicates the hardware usage of the two approaches. The cell usage for the Wallace Tree approach is greater than the Vedic approach.

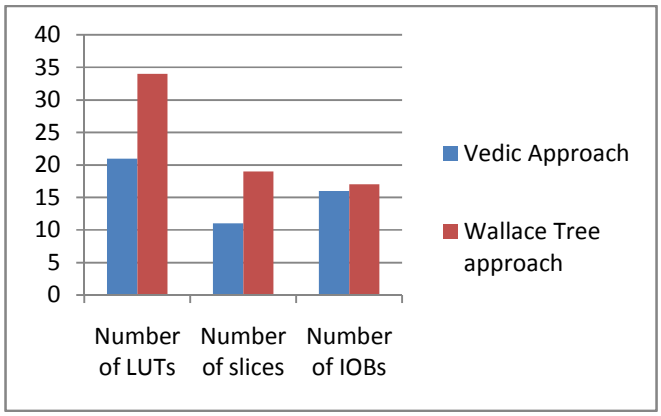


Fig. 8 Comparison of the hardware

CONCLUSIONS

The proposed method gives the design, implementation, and analysis of Vedic Multiplication in VHDL. The delay in 4 bit multiplication using the Vedic multiplication is 12.485ns.It is lower than that given by the Wallace Tree algorithm, 13.141 ns. The mathematical complexity is reduced in Vedic Multiplier. The hardware used is also lowered. Thus overall, the Multiplier is made more efficient and smaller in size for Digital signal processors. Thus the package density of Digital signal processors is increased.

REFERENCES

[1]. John G. Proakis & Dimitris K. Manolaski , “Digital Signal Processing”
[2]. K. Eshraghan & D.A. Pucknell, “ Fundamentals of VLSI”
[3]. A. Kumar, Dr. R.K Sarin ,Ashish R, Dr. Arun Khosla,”Small area Reconfigurable FFT Design by Vedic mathematics, 978-1-4244-5586- 7/2010 IEEE

[4]. Vedic Mathematics by Jagadguru Swamishri Bharati Krashna Tirthaji Maharaja
[5]. G. Ganeshkumar," Design of High speed vedic multiplier using Vedic Mathematics Technique", International Journal of Scientific And Research Publications, Volume 2, Issue 3 , Mar. 2012
[6]. Nidhi Mittal, Abhijit Kumar , Hardware Implementation of FFT using Vertical & Crosswise algorithm , International Journal of Computer Applications(0975-8887), Volume 35-No. 1, Dec. 2011
[7]. C.S.Wallace, " A suggestion for a Fast Multiplier " , IEEE Transaction on Electronic Computers,1964,pgs. 14-16
[8]. M. Moris Mano , "Digital Logic and Computer Design" , published by Pearson', First Edition, 2004
[9]. Devika Jaina , Kabiraj sethi , Rutuparna Panda , "Vedic Mathematics Based Multiply Accumulate Unit ,978-0-7695-4587-5/11, 2011 IEEE
[10]. H.Thapliyal and M.B. Srinivas, " High Speed Efficient N*N Bit Parallel Hierarchical Overlay Multiplier Architecture based on Ancient Indian Vedic Mathematics" , Enformatica Trans, vol 2 pp.225-228,Dec.2004
[11]. Douglas Perry , "Programming in VHDL"