# VLSI ARCHITECTURE OF AN 8-BIT MULTIPLIER USING VEDIC MATHEMATICS IN 180NM TECHNOLOGY

S. Karunakaran[1] and B. Poonguzharselvi[2]
[1]Professor, Department of ECE, Vardhaman College of Engineering (Autonomous),
Shamshabad, Hyderabad-501 218, India
[2]Assistant Professor, Department of CSE, Chaitanya Bharathi Institute of Technology
(Autonomous), Kokapet, Hyderabad-500 075, India

***ABSTRACT***

*A Multiplier is one of the key hardware blocks in most fast processing system which requires less power dissipation. A conventional multiplier consumes more power. This paper presents a low power 8 bit Vedic Multiplier (VM) based on Vertically & Crosswise method of Vedic mathematics, a general multiplication formulae equally applicable to all cases of multiplication. It is based on generating all partial products and their sum in one step. The implementation is done using cadence Virtuoso tool. The power dissipation of 8x8 bit Vedic multiplier obtained after synthesis is compared with conventional multipliers such as Wallace tree and array multipliers and found that the proposed Vedic multiplier circuit seems to have better performance in terms of power dissipation.*

***KEYWORDS:*** *Array Multiplier, Wallace Tree Multiplier, Vedic Multiplier*

## I. INTRODUCTION

Vedic Mathematics is the name given to the ancient system of Indian Mathematics which was rediscovered from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirthaji (1884-1960). According to his research all of mathematics is based on sixteen Sutras, or word-formulae. For example, 'Vertically and crosswise` is one of these Sutras.

Perhaps the most striking feature of the Vedic system is its coherence. Instead of a hotchpotch of unrelated techniques the whole system is beautifully interrelated and unified: the general multiplication method, for example, is easily reversed to allow one-line divisions and the simple squaring method can be reversed to give one-line square roots. And these are all easily understood. This unifying quality is very satisfying, it makes mathematics easy and enjoyable and encourages innovation.

In the Vedic system 'difficult' problems or huge sums can often be solved immediately by the Vedic method. These striking and beautiful methods are just a part of a complete system of mathematics which is far more systematic than the modern 'system'. Vedic Mathematics manifests the coherent and unified structure of mathematics and the methods are complementary, direct and easy.

Multiplication is the most important arithmetic operation in signal processing applications. 8x8 vedic multiplier is designed using vertical and cross-wise algorithm (Urdhva Tiryagbhyam) and its performance has been compared with the conventional multiplier such as array multiplier and Wallace tree multiplier and found that the proposed Vedic multiplier circuit seems to have better performance in terms of power dissipation

## II. VEDIC MULTIPLIER

Multiplication is the most important arithmetic operation in signal processing applications. All the signal and data processing operations involve multiplication. As speed is always a constraint in the multiplication operation, increase in speed can be achieved by reducing the number of steps in the computation process[1].The speed of multiplier determines the efficiency of such a system. In any system design, the three main constraints which determine the performance of the system are speed, area and power requirement.

Urdhva Tiryagbhyam[2] is the most generalized sutra for implementation of Vedic Multiplier designs because with increase in number of bits both area and delay increase slowly.

The beauty of Vedic Multiplier lies in the fact that they can be used to solve cumbersome mathematical operations orally thereby improving speed.

The multiplier architecture is based on Urdhva Tiryagbhyam (vertical and cross-wise algorithm) sutra[3]. The 2x2 multiplication has been done in a single line in Urdhva Tiryagbhyam sutra, whereas in shift and add (conventional) method, four partial products have to be added to get the result. Thus, by using Urdhva Tiryagbhyam Sutra in binary multiplication, the number of steps required calculating the final product will be reduced and hence there is a reduction in computational time and increase in speed of the multiplier. Illustration is given in Fig.1.

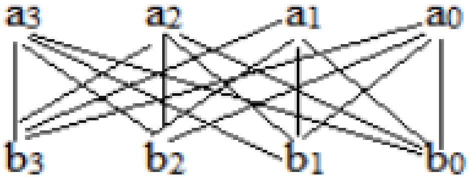


**Fig.1** Illustration of Urdhva Tiryagbhyam sutra

## 2.1 ARRAY MULTIPLIER:

The multiplier module is used to multiply the filter coefficients with incoming input signals. Parallel multipliers are classified mainly into array multipliers and tree multipliers. Basically, the array multipliers include two-dimensional full adder arrays. The array multiplier[4] is an efficient layout of a combinational multiplier. It consists of rows of carry-save adders. The summands are generated in parallel with AND gates, and then added to the array of 1-bit full-adders. Such an n×n multiplier array consists of (n-1) rows of carry save adders in which each row contains (n-1) full adders. The last row is a ripple carry adder for carry propagation. [9] [10]

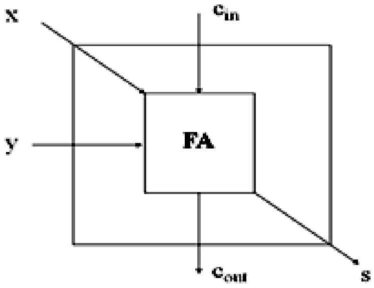


**Fig.2** Array multiplier basic cell

The basic cell of an array is shown in Figure 2. This is the full adder. The inputs are x, y, $c_{in}$ (carry in). The outputs are s (sum) and $c_{out}$ (carry out).

sum= a ^ b
carryout = a&b

The rules for array multiplication are,

- Matrices must have the same dimensions.
- Dimensions of the resulting matrix are the same as the two multiplied matrices.
- Resulting elements are the products of corresponding elements in the multiplied matrices as shown in Figure 3

$$
\begin{array}{cccccccccc}
 & & & & & a_4 & a_3 & a_2 & a_1 & a_0 \\
 & & & \times & & x_4 & x_3 & x_2 & x_1 & x_0 \\
\hline
 & & & & & a_4x_0 & a_3x_0 & a_2x_0 & a_1x_0 & a_0x_0 \\
 & & & & a_4x_1 & a_3x_1 & a_2x_1 & a_1x_1 & a_0x_1 & \\
 & & & a_4x_2 & a_3x_2 & a_2x_2 & a_1x_2 & a_0x_2 & & \\
 & & a_4x_3 & a_3x_3 & a_2x_3 & a_1x_3 & a_0x_3 & & & \\
 & a_4x_4 & a_3x_4 & a_2x_4 & a_1x_4 & a_0x_4 & & & & \\
\hline
p_9 & p_8 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

**Fig.3** Array multiplication

## 2.2WALLACE TREE MULTIPLIER

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

The Wallace tree has two steps: [5] [6] [7] [8]

1. Reduce the number of partial products to two by layers of full and half adders.
2. Group the wires in two numbers, and add them with a conventional adder.

The first phase works as follows. As long as there are three or more wires with the same weight add a following layer:

- Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, input them into a half adder.
- If there is just one wire left, connect it to the next layer.

The benefit of the Wallace tree is that there are only reduction layers, and each layer has propagation delay. As making the partial products is and the final addition is, the multiplication is only, not much slower than addition (however, much more expensive in the gate count). Naively adding partial products with regular adders would require time. From a complexity theoretic perspective, the Wallace tree algorithm puts multiplication in the class NC. These computations only consider gate delays and don't deal with wire delays, which can also be very substantial. The Wallace tree can be also represented by a tree of 3/2 or 4/2 adders. It is sometimes combined with Booth encoding.

A fast process for multiplication of two numbers was developed by Wallace. Using this method, a three step process is used to multiply two numbers; the bit products are formed, the bit product matrix is reduced to a two row matrix where sum of the row equals the sum of bit products, and the two resulting rows are summed with a fast adder to produce a final product.

Three bit signals are passed to a one bit full adder (“3W”) which is called a three input Wallace tree circuit, and the output signal (sum signal) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage full adder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the full adder located at a one bit higher position.

Wallace tree is a tree of carry-save adders arranged as shown in Figure 4. A carry save adder consists of full adders like the more familiar ripple adders, but the carry output from each bit is brought out to form second result vector rather being than wired to the next most significant bit. The carry vector is ‘saved’ to be combined with the sum later. In the Wallace tree method, the circuit layout is not easy although the speed of the operation is high since the circuit is quite irregular.

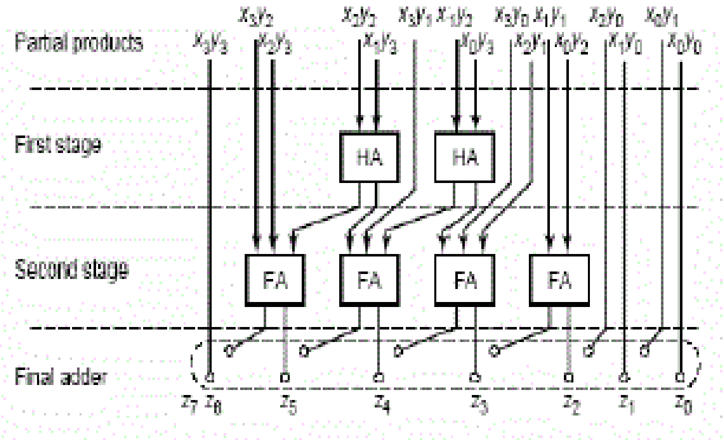
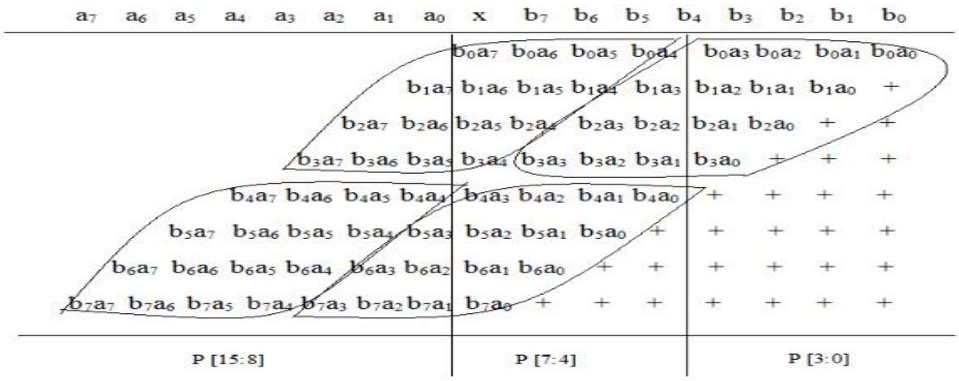

Partial products
First stage
Second stage
Final adder
HA
HA
FA
FA
FA
FA
$x_3y_3$ $x_3y_2$ $x_2y_3$ $x_3y_2$ $x_1y_3$ $x_3y_1$ $x_1y_2$ $x_0y_3$ $x_3y_0$ $x_1y_1$ $x_2y_1$ $x_0y_2$ $x_2y_0$ $x_1y_0$ $x_0y_1$ $x_0y_0$
$z_7$ $z_6$ $z_5$ $z_4$ $z_3$ $z_2$ $z_1$ $z_0$

**Fig.4** Architecture of Wallace Tree multiplier

## 2.3 8x8 VEDIC MULTIPLIER:



$a_7$ $a_6$ $a_5$ $a_4$ $a_3$ $a_2$ $a_1$ $a_0$ x $b_7$ $b_6$ $b_5$ $b_4$ $b_3$ $b_2$ $b_1$ $b_0$

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $b_0a_7$ | $b_0a_6$ | $b_0a_5$ | $b_0a_4$ | $b_0a_3$ | $b_0a_2$ | $b_0a_1$ | $b_0a_0$ |
| | | | | | | $b_1a_7$ | $b_1a_6$ | $b_1a_5$ | $b_1a_4$ | $b_1a_3$ | $b_1a_2$ | $b_1a_1$ | $b_1a_0$ | + |
| | | | | | $b_2a_7$ | $b_2a_6$ | $b_2a_5$ | $b_2a_4$ | $b_2a_3$ | $b_2a_2$ | $b_2a_1$ | $b_2a_0$ | + | + |
| | | | | $b_3a_7$ | $b_3a_6$ | $b_3a_5$ | $b_3a_4$ | $b_3a_3$ | $b_3a_2$ | $b_3a_1$ | $b_3a_0$ | + | + | + |
| | | | $b_4a_7$ | $b_4a_6$ | $b_4a_5$ | $b_4a_4$ | $b_4a_3$ | $b_4a_2$ | $b_4a_1$ | $b_4a_0$ | + | + | + | + |
| | | $b_5a_7$ | $b_5a_6$ | $b_5a_5$ | $b_5a_4$ | $b_5a_3$ | $b_5a_2$ | $b_5a_1$ | $b_5a_0$ | + | + | + | + | + |
| | $b_6a_7$ | $b_6a_6$ | $b_6a_5$ | $b_6a_4$ | $b_6a_3$ | $b_6a_2$ | $b_6a_1$ | $b_6a_0$ | + | + | + | + | + | + |
| $b_7a_7$ | $b_7a_6$ | $b_7a_5$ | $b_7a_4$ | $b_7a_3$ | $b_7a_2$ | $b_7a_1$ | $b_7a_0$ | + | + | + | + | + | + | + |
| P [15:8] | | | | | | | P [7:4] | | | | P [3:0] | | | |

**Fig.5** Multiplication of 8bit number using Vedic mathematics

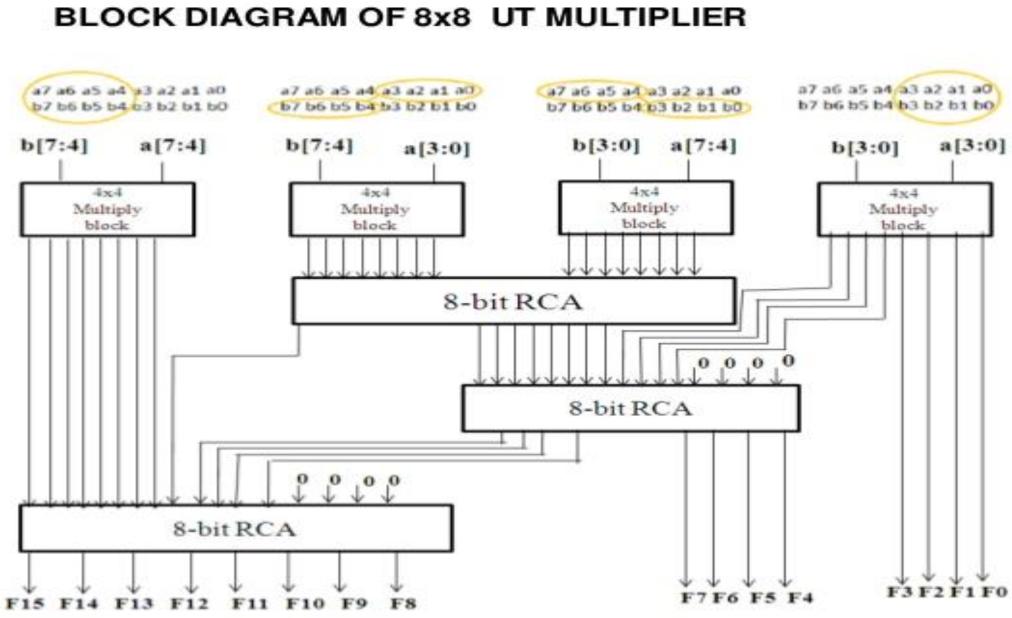


**Fig.6** Module of 8 bit Vedic multiplier

The 8x8 bit Vedic multiplier module as shown in the block diagram in Fig. 6 can be easily implemented by using four 4x4 bit Vedic multiplier modules as discussed in the previous section.

Let’s analyze 8x8 multiplications, say A= A7 A6 A5 A4 A3 A2 A1 A0 and B= B7 B6 B5B4 B3 B2 B1B0. The output line for the multiplication result will be of 16 bits as – S15 S14 S13 S12 S11 S10 S9 S8 S7 S6 S5 S4 S3 S2 S1 S0. Let’s divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL. The 16 bit product can be written as P = A x B = (AH-AL) x (BH-BL) = AH x BH + (AH x BL + AL x BH) + AL x BL Using the fundamental of Vedic multiplication, taking four bits at a time and using 4 bit multiplier block a we can perform the multiplication. The outputs of 4x4 bit multipliers are added accordingly to obtain the final product. Here total three 8 bit Ripple-Carry Adders are required The module of 8X8 Vedic multiplier shown in Figure 6 can be implemented by using four 4x4 bit Vedic multiplier modules . Analyzing 8x8 multiplications, inputs are a7-a0 and b7- b0 and the multiplication’s 16 bits output will be $S_{15}$-$S_0$.In this , four 4x4 bit Vedic multipliers and three 8 bit RC Adders(having 2 input of 8 bits) are required. Inputs are given to the 4x4 bit Vedic multipliers and the output of multiplier is of 8 bits. Now, the input of the 1st RC Adder is the output of the 2nd and 3rd 4x4 bit multipliers which gives output of 8 bits (7-0) + one carry. The 2nd RC Adder will add the output of 1st RC Adder (7-0) and 4 bits (7-4) output of 1st 4x4 bit Vedic multiplier, other 4bit of are considered as 0.So output is of 8- bits(7-0) and one carry(carry is discarded).The 3rd RC Adder will add the output of 4th 4x4 bit multiplier(7-0) and 4 bits of output of 2nd RC Adder (7-4), other 4 bits are carry of 1st RC Adder and 0.Now ,the output of 8x8 multiplier is s(3-0) output of 1st 4x4 bit multiplier(3-0), s(7-4) is output of 2nd RC Adder (3-0) and s(15-8)=8 is output of 3rd RC Adder(7-0).Thus, implementation of NxN bit Vedic multiplier for N bits are achieved. Hence, efficiency and performance can be improved.

### 2.3.1 8-BIT BINARY MULTIPLICATION USING URDHVA TIRYAKBHYAM SUTRA

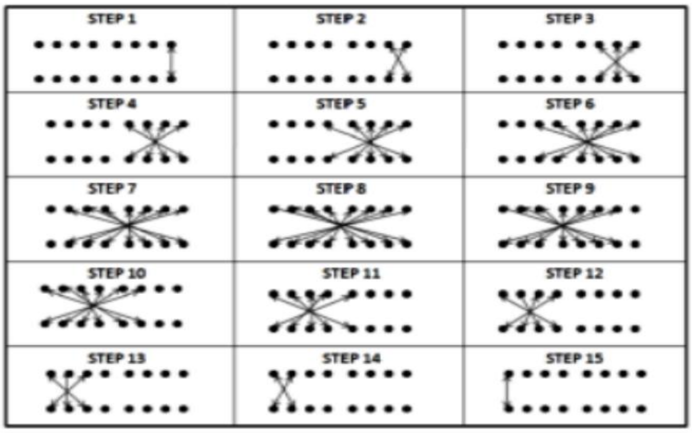


**Fig.7** Steps involved in Vedic Multiplication

## III. SIMULATION RESULTS

### 3.1ARRAY MULTIPLIER

The simulation results for array multiplier, Wallace tree multiplier and Vedic multiplier are shown in Figure 8,9,10 respectively.

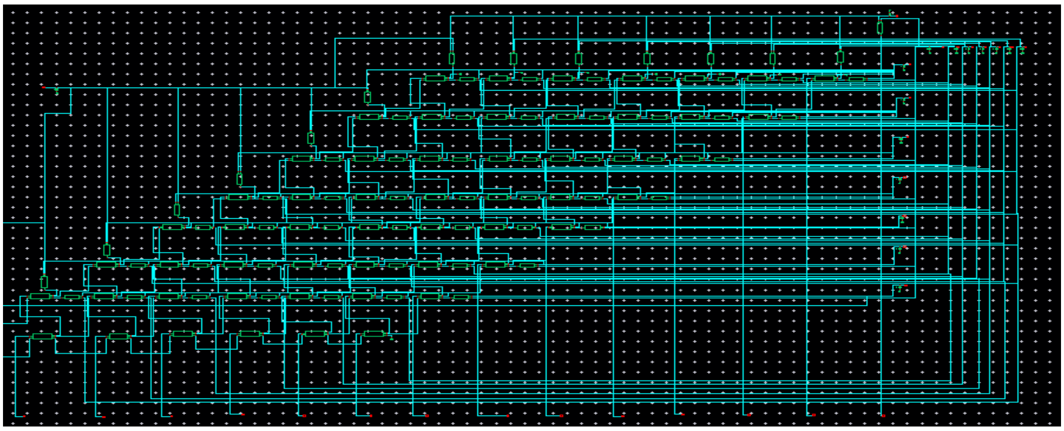


**Fig.8** Schematic View of Array Multiplier

### 3.2WALLACE TREE MULTIPLIER ARCHITECTURE:

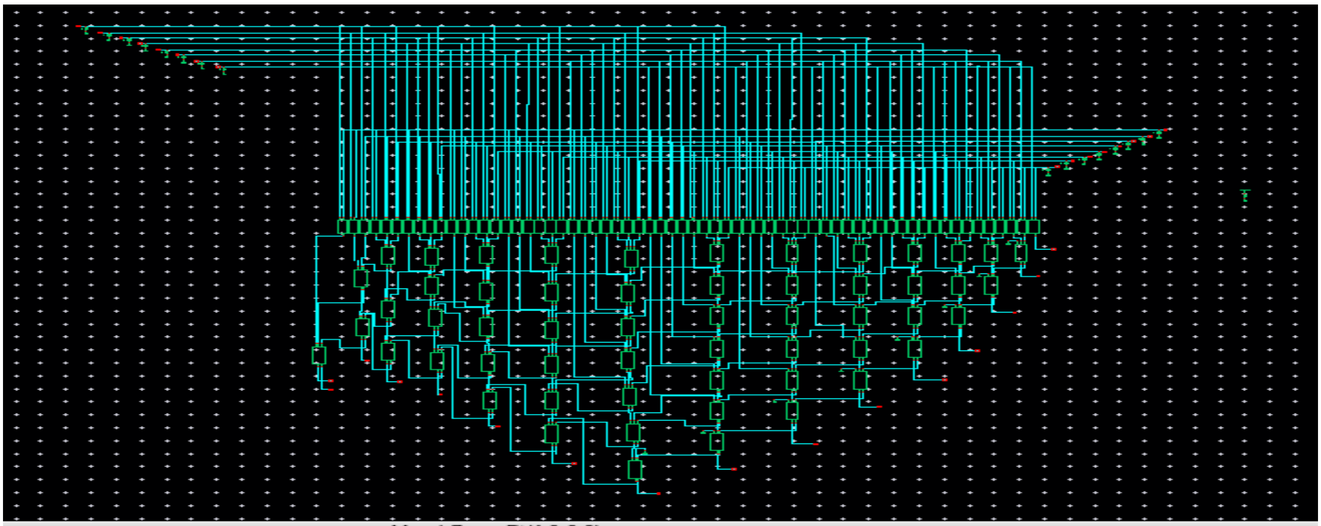


**Fig.9** Schematic View of Wallace Tree Multiplier

### 3.3 8-BIT VEDIC MULTIPLIER:

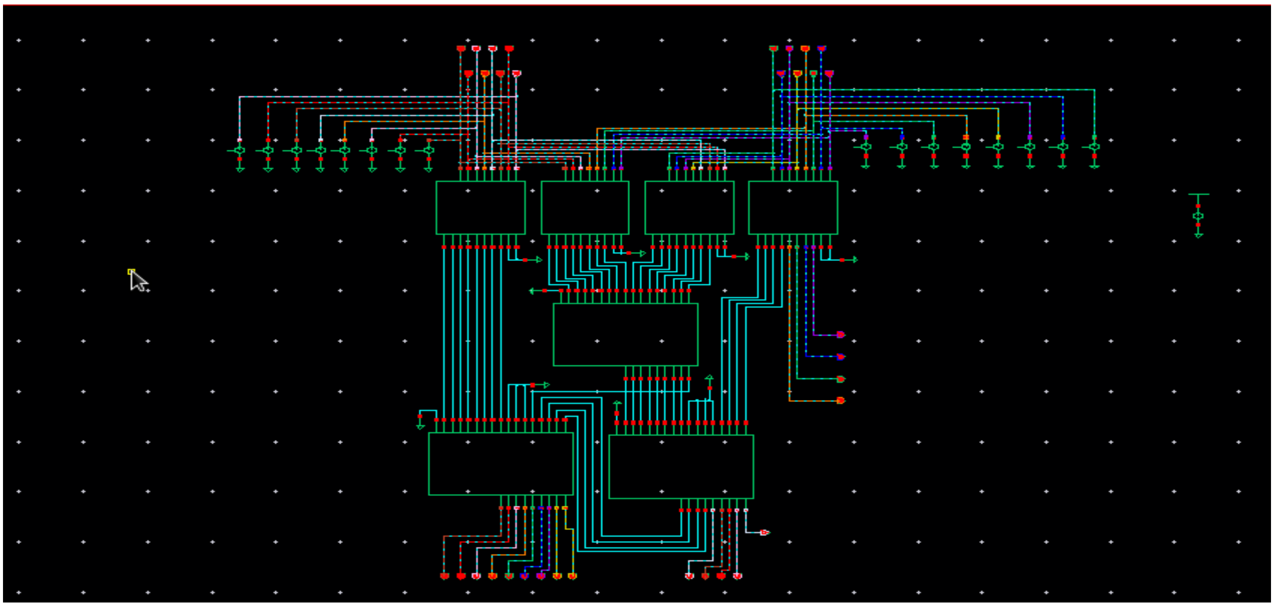


**Fig.10** Schematic View of 8-Bit Vedic Multiplier

## 3.4 8X8 VEDIC MULTIPLIER SIMULATION WAVEFORMS:

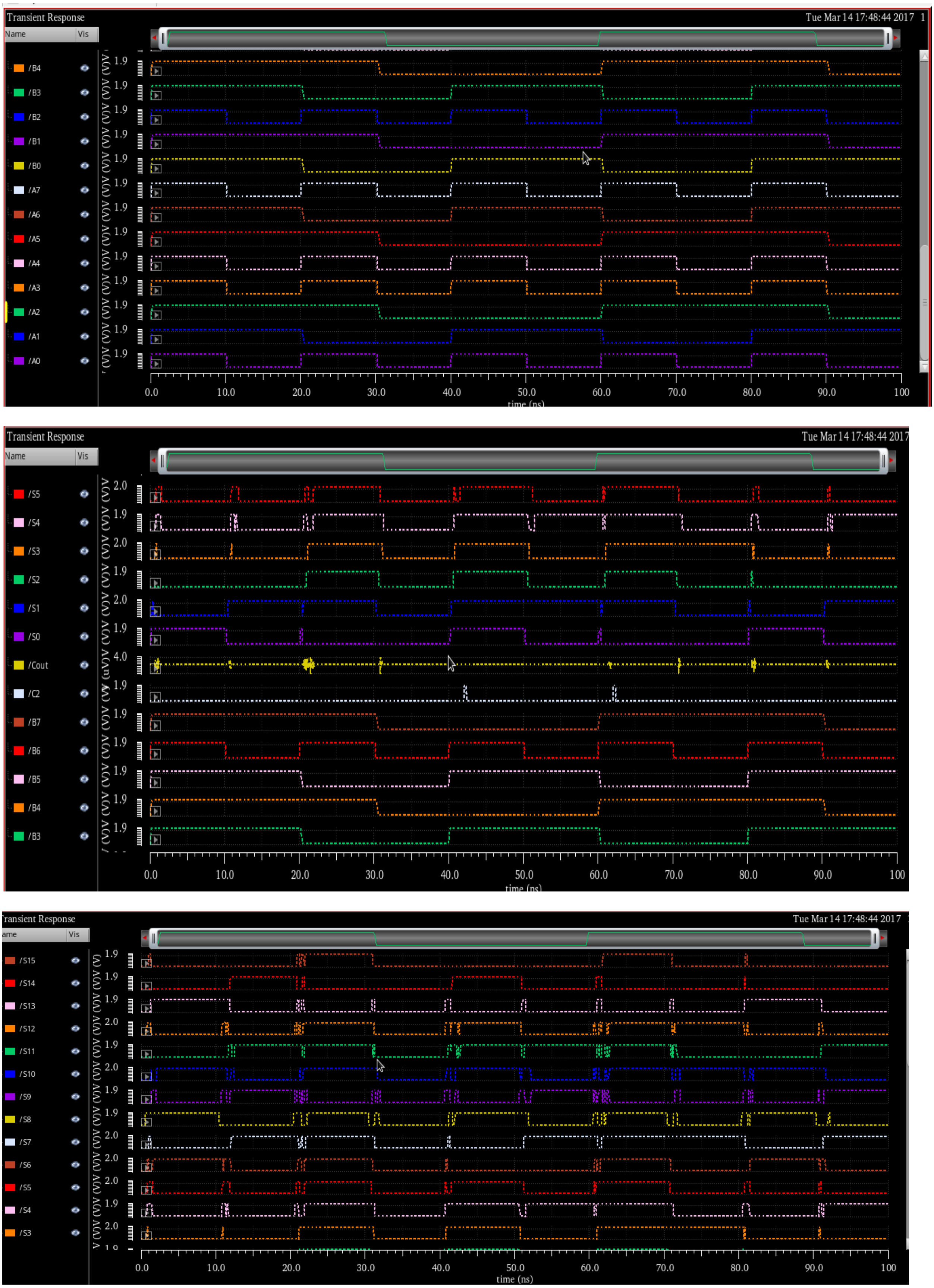


**Fig.11** Waveforms of Vedic Multiplier

The output of Vedic multiplier for different input combinations are represented for the above waveforms. In the above table ,X represents multiplier, Y represents multiplicand , S represents Output

| | | | |
|---|---|---|---|
| 1) X= 01100110<br>Y=11011101<br>S= 0100000101010010 | 2) X= 10111101<br>Y=01101011<br>S= 0111111110111001 | 3) X= 11011011<br>Y=10110110<br>S=1111110010111010 | 4) X= 00000000<br>Y=00000000<br>S= 000000000000000 |
| 5) X= 11111111<br>Y=11111111<br>S= 1000001010100100 | 6) X= 10110110<br>Y= 11011011<br>S= 1111110010111010 | 7) X= 11011101<br>Y= 01100110<br>S= 0100000101010010 | 8) X= 01101011<br>Y= 10111101<br>S= 0111111110111001 |

## IV. RESULTS AND DISCUSSIONS

### 4.1ARRAY MULTIPLIER:

The Power consumed by the outputs i.e. from S0-S15 are given in Table 1

**Table 1:** Power Consumed at every output Pin(Array Multiplier)

| **Output** | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |
|---|---|---|---|---|---|---|---|---|---|
| **Power Consumed (mW)** | 907.1 | 3.016 | 544.2 | 558.4 | 408.9 | 707.4 | 849.2 | 568.5 | 764 |

| **Output** | S9 | S10 | S11 | S12 | S13 | S14 | S15 |
|---|---|---|---|---|---|---|---|
| **Power Consumed (mW)** | 898.9 | 1193 | 1197 | 1168 | 699.5 | 895.6 | 720.8 |

### 4.2WALLACE TREE MULTIPLIER

The Power consumed by the outputs i.e from S0-S15 are shown in Table 2

**Table 2:** Power Consumed at output Pins (Wallace Tree Multiplier)

| **Output** | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |
|---|---|---|---|---|---|---|---|---|---|
| **Power Consumed (mW)** | 551.2 | 717.4 | 871.6 | 877.5 | 442.7 | 729.8 | 816.1 | 464.6 | 689.8 |

| **Output** | S9 | S10 | S11 | S12 | S13 | S14 | S15 |
|---|---|---|---|---|---|---|---|
| **Power Consumed (mW)** | 599.4 | 554.9 | 1460 | 750.1 | 1187 | 552.8 | 531.6 |

### 4.3 VEDIC MULTIPLIER

The Power Consumed by the outputs i.e. from S0-S15 are shown in Table 3.

**Table 3:** Power dissipated at output Pins in Vedic Multiplier

| **Output** | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |
|---|---|---|---|---|---|---|---|---|---|
| **Power Consumed (mW)** | 907.1 | 2.189 | 532.3 | 356 | 399.2 | 563.3 | 714.4 | 545.9 | 278.6 |

| **Output** | S9 | S10 | S11 | S12 | S13 | S14 | S15 |
|---|---|---|---|---|---|---|---|
| **Power Consumed (mW)** | 571.9 | 1057 | 570.9 | 866.9 | 697.4 | 511.4 | 368.3 |

## V. Power Comparison

Hence, vedic multiplier has been chosen for experimental purpose and it has been compared with the array multiplier and Wallace tree multiplier.The power consumed by vedic multiplier is are 14.22 % and 18.86% lesser when compared with the power dissipation of array multiplier and Wallace tree multiplier.

## VI. Conclusion

It can be concluded that Vedic Multiplier is superior in aspects like power consumption. However Array Multiplier requires more power consumption and gives optimum number of components required, but delay for this multiplier is larger than Wallace Tree Multiplier which is very complex to design. Hence for low power requirement Vedic multiplier is suggested. Ancient Indian Vedic Mathematics gives efficient algorithms or formulae for multiplication which increase the speed of devices.

Urdhva Tiryakbhyam, is general mathematical formula and equally applicable to all cases of multiplication. Also, the architecture based on this sutra is seen to be similar to the popular array multiplier where an array of adders is required to arrive at the final product. Due to its structure, it suffers from a high carry propagation delay in case of multiplication of large number. This problem can solved by Urdhva Triyakbhyam Sutra which reduces the multiplication of two large numbers to the multiplication of two small numbers.

The power of Vedic Mathematics can be explored to implement high performance multiplier in VLSI applications. Urdhva Triyakbhyam Sutra in Vedic Mathematics is less complex than other sutras which can be tested with its implementation with different logics in VLSI.

The power consumed by Vedic multiplier are 14.22 % and 18.86% lesser when compared with the power dissipation of array multiplier and Wallace tree multiplier.

## VII. Future Scope

The multiplier design can be extended for 16 bits. Multiplication is the most time consuming computation in digital signal processing algorithm computation. The research can be extended in developing low power consumption FIR filter by designing low power consumption multiplier.

## References

[1]. Krishnaveni D, Umarani T.G, “VLSI Implementation of Vedic Multiplier With Reduced Delay”, IJATER, NCET-Tech, ISSN No:-2250- 3536, Volume 2, Issue 4, July 2012.

[2]. Poornima M, Shivaraj Kumar Patil, Shivu kumar, Shridhar KP,Sanjay H, “Implementation of Multiplier Using Vedic Algorithm”,IJITEE,ISSN:- 2278-3075,Volume-2,Issue-6,May-2013.

[3]. Premananda B.S, Samarth S. Pai, Shashank B, Shashank S.Bhat, “Design And Implementation of 8-bit Vedic Multiplier”,IJAREEIE,Vol.2,Issue 12,ISSN:2320-3765,Dec-2013.

[4]. Abu-Khater, I., Bellaouar, A. and Elmasry, M. “Circuit techniques for CMOS low-power high performance multipliers”, IEEE Journal of SolidState Circuits, Vol.31, No.10, pp.1535- 1546, 1996.

[5]. R.Bala Sai Kesava, B.Lingeswara Rao, K. Bala Sindhuri, N.Udaya Kumar, “Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter” Advances in Signal Processing (CASP), Conference on 9-11 June 2016, INDIA Pages: 248 - 253, DOI: 10.1109/CASP.2016.7746174

[6]. Kokila Bharti Jaiswal, Nithish Kumar V, Pavithra Seshadri, Lakshminarayanan G, “Low Power Wallace tree multiplier using modified full adder”, 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN),INDIA Pages: 1 – 4, DOI: 10.1109/ICSCN.2015.7219880

[7]. Vishal Shankarrao Muley, Anchu Tom, Vigneswaran T,” Design of Baugh Wooley and Wallace tree multiplier Using two phase clocked adiabatic static CMOS logic ”, International Conference on Industrial Instrumentation and Control(ICIC) Year: 2015,INDIA Pages: 1178 – 1183, DOI: 10.1109/IIC.2015.7150926

[8]. Abdullah A. AlJuffri; Aiman S. Badawi; Mohammed S. BenSaleh; Abdulfattah M. Obeid; Syed Manzoor Qasim,”FPGA implementation of scalable microprogrammed FIR filter architectures using Wallace tree and Vedic multipliers”, 2015 Third International Conference on Technological

Advances in Electrical, Electronics and Computer Engineering (TAEECE)Year: 2015 INDIA Pages: 159 – 162, DOI: 10.1109/TAEECE.2015.7113619

[9]. Takahiro Yamamoto, Ittetsu Taniguchi, Hiroyuki Tomiyama, Shigeru Yamashita, Yuko Hara-Azumi, "A systematic methodology for design and analysis of approximate array multipliers", 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) Year: 2016 Pages: 352 – 354, DOI: 10.1109/APCCAS.2016.780397

[10]. S. Srikanth, I. Thahira Banu,G. Vishnu Priya, G. Usha"Low power array multiplier using modified full adder" 2016 IEEE International Conference on Engineering and Technology (ICETECH) Year: 2016 INDIA Pages: 1041 - 1044, DOI: 10.1109/ICETECH.2016.7569408

## AUTHORS BIOGRAPHY

**Dr. S. Karunakaran** was born in Erode, Tamilnadu, INDIA, in the year1980. He received the Bachelor of Engineering degree in Electrical and Electronics Engineering from the Periyar University, Salem in 2002 and the Master of Engineering degree in VLSI design from the Anna University ,Chennai , in the year 2002 . He obtained his Ph.D. degree with the Department of Electronics and Communication Engineering, Anna University Chennai in the year 2015. Presently, he is working as a Professor in the department of ECE in Vardhaman College of Engineering, Hyderabad .His research interests include VLSI design, ASIC design, Low power VLSI design etc.

**Smt. B. Poonguzharselvi** was born in Erode, Tamilnadu, INDIA in the year 1989.She received the Bachelor of Engineering degree in Computer Science and Engineering from Anna University, Chennai in 2010 and the Master of Engineering in Computer Science and Engineering from the Anna University ,Chennai ,in the year 2012 .Presently, she is working as a Assistant Professor in the Department of CSE in Chaitanya Bharathi Institute of Technology, Hyderabad .Her research interests include data structures, DBMS, C programming, Java Programming etc.