IMPLEMENTATION OF VEDIC MULTIPLIER

Prashanth.N. A[#1], Nagendra Prasad S[#2]

BMS Institute of Technology and Management, Avalahalli, Yelahanka, Bangalore, Karnataka, India
PES Institute of Technology, Hosur Road, Electronic city, Bangalore, Karnataka, India

Abstract: *Multiplication is one of the most commonly performed functions by ALU and ALU being one of the significant elements of the processors, the performance of the processors depends on ALU and other factors. Various methods such as Booth's multiplier have been proposed earlier to improve the performance of a multiplier. In this paper we are improving the speed and efficiency of the multiplications based on Vedic mathematics sutras such Urdhva Triyakbhyam, modified Urdhva Triyakbhyam and ekanyunena Purvena sutra and comparing the performance with Booth's multiplier.*

Keywords: Urdhva Triyakbhyam, Modified Urdhva Triyakbhyam, Ekanyunena Purvena, Vedic Multiplier, Verilog HDL.

I. INTRODUCTION

Vedic mathematics is a book comprising of various mathematical techniques written by the author Bharathi Krishna Thritha and was published in the year 1965. The Author states that these mathematical techniques were retrieved from the Vedas. The book mainly consists of 16 sutras and 13 sub sutras which are used to improve the efficiency and speed of the mathematical calculations.

The sutras reduce the steps of basic mathematical calculations hence by increasing the speed and efficiency of the calculations. The efficiency of the calculations has increased the scope of Vedic mathematics in various fields from statistics to astronomy and financial domains.

ALU is an abbreviation of "Arithmetic logic unit" which is like a heart of processors. It is basically a combinational circuit that is used that performs arithmetic and bitwise operations of binary numbers. The various operations of ALU are addition, subtraction, multiplication, division etc.

The performance of the processor depends on various factors and one of the main factors is the performance of the ALU which in turn depends on the speed of calculations. Hence, to improve the performance of the processor in this paper we have proposed an efficient way to do multiplications based on Vedic mathematics sutras such urdhva triyakbyham, modified urdhva triyakbyham and ekanyunena purvena sutras, using high level hardware description language.

Urdhva Tiryakbhyam Sutra is a Multiplication technique used for multiplying N-bit numbers of all cases. In this sutra, NxN multiplier is designed using (N/2) x (N/2) multiplier blocks. The basic building block of the design is 2x2 multiplier. 4x4 multiplier is designed using 2x2 bits multiplier blocks. Similarly, 8x8 bits and 16x16 bits multipliers are designed using 4x4 bits and 8x8 bits multiplier blocks respectively.

Modified "Urdhva Triyakbhyam" sutra can be modified to find square of a number with reduced time delay, area and power. Urdhva sutra is modified to have 3 multiplier blocks instead of 4 multiplier blocks. Reduction in multiplier blocks will result in reduced time delay and power.

"Ekanyunena Purvena" is a technique used in multiplication of any number with 9s series number (Ex:1565x9999). This sutra helps us to calculate the result in short-cut method by just using subtraction and concatenation. Therefore, the time delay and the area will be reduced.

II. VEDIC MATHEMATICS AND SUTRAS

Vedic mathematics is a book comprising of various mathematical techniques written by the author Bharathi Krishna Thritha and was published in the year 1965. The Author states that these mathematical techniques were retrieved from the Vedas. The book mainly consists of 16 sutras and 13 sub sutras which are used to improve the efficiency and speed of the mathematical calculations. The sutras reduce the steps of basic mathematical calculations hence by increasing the speed and efficiency of the calculations. The efficiency of the calculations has increased the scope of Vedic mathematics in various fields from statistics to astronomy and financial domains.

III. DESIGN AND IMPLEMENTATION

The above-mentioned sutras, Urdhva Triyakbhyam, Modified Urdhva Triyakbhyam and Ekanyunena Purvena is implemented using Verilog HDL in Xilinx Vivado. The designs are described in the following paragraphs.

A. Urdhva Triyakbhyam Sutra

Urdhva Tiryakbhyam" Sutra is a Multiplication technique used for multiplying N-bit numbers of all cases. This sutra multiplies in Vertical and crosswise fashion, hence the name. This sutra can be used for any two numbers in decimal number and binary number system. In this sutra, NxN multiplier is designed using (N/2)x(N/2) multiplier blocks. The basic building block of the design is 2x2 multiplier. 4x4 multiplier is designed using 2x2 bits multiplier blocks. Similarly, 8x8 bits and 16x16 bits multipliers are designed using 4x4 bits and 8x8 bits multiplier blocks respectively. The partial products and sum obtained by N/2 x N/2 multiplier blocks and carry lookahead blocks respectively are calculated parallelly. Therefore, it doesn't depend on clock frequency and takes the same amount of time. Thus, the multiplier is more efficient for higher bits. The power and path delay increases slowly with the number of bits which weighs up compared to other stereotyped multipliers. Therefore, it is more efficient for higher bits multiplication.

1) 2X2 MULTIPLIER:

The design of the 2x2 bits multiplier is shown in Figure 1. 2x2 is a basic block multiplier from which other block multipliers can be designed. So, in this multiplier multiplication of two numbers takes place. As shown in the figure, inputs are represented by "a[i]" and "b[i]" and output is represented by "q[i]". As we can see here the inputs a and b are of two bits and output q is 4 bits. In the hardware equivalent shown, the AND gates play a role of 2-bit multipliers and two half adders are used to add the partial products to get the final product. Consider the multiplicand and multiplier (a1 a0) and (b1 b0) respectively, each of size 2 bits. The least significant bit of product is obtained by multiplying LSBs of the inputs i.e., a[0] and b[0]. Similarly, one of the partial products is obtained by multiplying MSBs of inputs i.e., a[1] and b[1]. These constitute the vertical multiplication. One of the partial products of crosswise multiplication is obtained by multiplying the LSB of a (multiplicand) and the next higher bit of b (multiplier). The other partial product is obtained by multiplying LSB of b and higher bit of a. The partial products obtained are given as inputs to half adder where the sum gives the second bit of the final product. The carry of the first half adder and the partial product obtained by vertical multiplication is given as inputs to the next half adder whose sum gives the 3rd bit of output and carry gives the 4thbit of output.

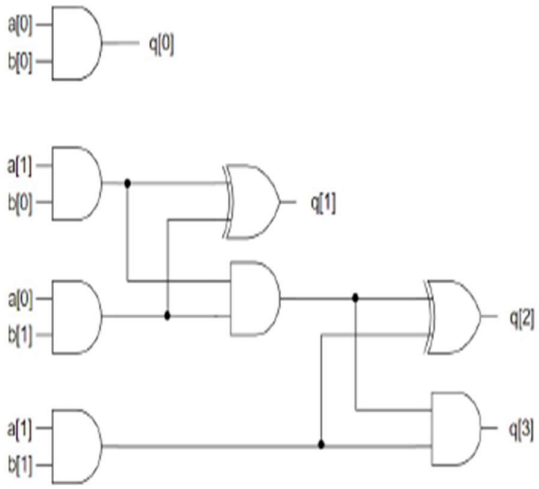

Fig 1: 2x2 Bits Urdhva Triyakbhyam Multiplier

2) 4X4 MULTIPLIER:

The design of the 4x4 bits multiplier is shown in Figure 2. The 2x2 multiplier is the basic building block of this multiplier. As shown in the figure, inputs are represented by "a[i]" and "b[i]" and partial products and output are represented by "q[i]" and "Q[i] respectively. Here we are dividing inputs "a" and "b" into two parts a[1:0] and a[3:2] and b[1:0] and b[3:2] respectively. The vertical and crosswise multiplications are handled by these basic blocks and addition is handled by carry lookahead adder. The design consists of four 2x2 Vedic multipliers, two 6-bit Carry look ahead adders and one 4bit Carry Lookahead adder. Firstly, we are taking two different bits of each input at a time and giving it to a separate 2x2 Vedic multiplier from which the partial products are calculated and their sums are calculated in parallel by using carry lookahead adder and is given as input to the next carry lookahead adder. The output of the later carry lookahead adder constitutes the 6 bits of final product. The final product is obtained by concatenating these 6 bits with the first 4 bits of partial product q0.

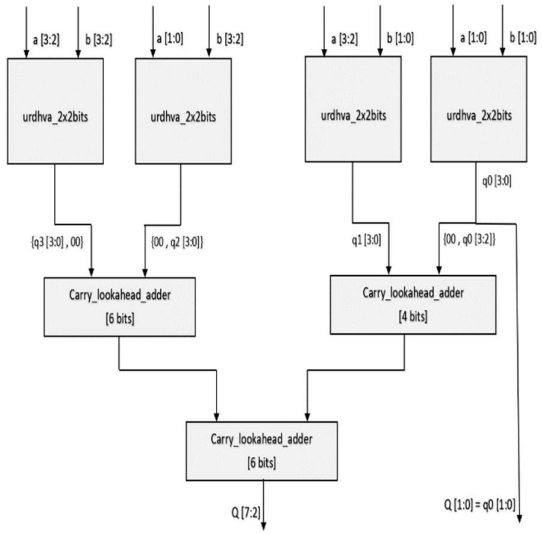

Fig 2: 4x4 Bits Urdhva Triyakbhyam Multiplier

3) 8X8 MULTIPLIER:

The design of 8x8 block multiplier is shown in figure 3. The 8x8 Multiplier is designed by employing 4x4 multiplier blocks. As shown in the figure," a" and "b" are the inputs of 16 bits each, "qi" is the partial products and "Q" is the output which is 16 bits. Similar to the above design, we are dividing inputs "a" and "b" into two parts a[3:0] and a[7:4] and b[3:0] and b[7:4] respectively. Where each pair of input bits are handled by a separate 4x4 Vedic multiplier. The design consists of four 4x4 Vedic multiplier, two 12-bit Carry look ahead adder and one 8-bit Carry Lookahead adder. Similar to the previous design of 4x4 multiplier, we are first taking four different bits of each input at a time and giving it to a separate 4x4 Vedic multiplier from which calculates the partial product and then their sums are calculated parallelly by using carry lookahead adder, producing an output of 16 bits Q[15:0].

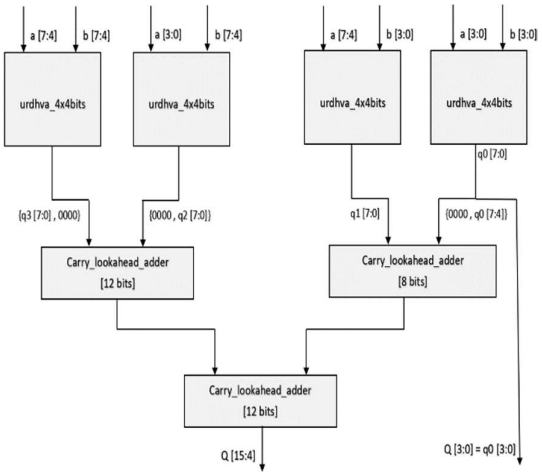

Fig 3: 8x8 Bits Urdhva Triyakbhyam Multiplier

4) 16X16 MULTIPLIER:

The design of the 16x16 block multiplier is shown in Figure 4. 8x8 multiplier blocks have been used to design 16x16 Multiplier. As shown in the figure," a" and "b" are the inputs of 16 bits each," qi" is the partial products and "Q" is the output which is 16 bits. Here we are dividing inputs "a" and "b" into two parts a[7:0] and a[15:8] and b[7:0] and b[15:8] respectively. Where each pair of input bits are managed by a separate 8x8 Vedic, thus the architecture of 16x16 block is designed by using 8x8 blocks. 16x16 multiplier consists of four 8x8 Vedic multiplier, two 24-bit Carry look ahead adder and one 16-bit Carry Look ahead adder. Firstly, we are taking eight different bits of each input at a time and giving it to a separate 8x8 Vedic multiplier, which calculates partial products, and their sums are calculated parallelly by using carry lookahead adder, thus producing a output of 32 bits Q[31:0].

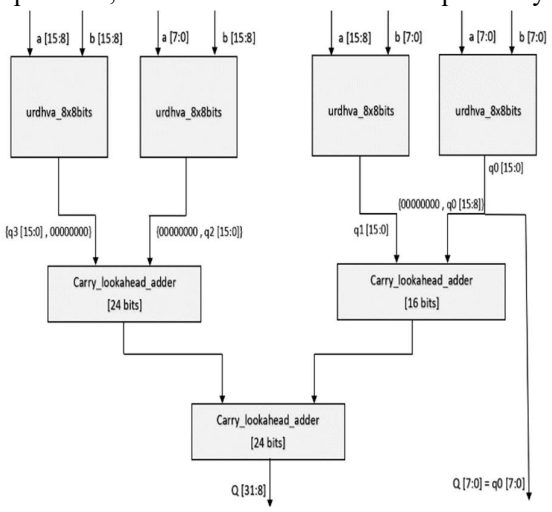

Fig 4: 16x16 Bits Urdhva Triyakbhyam Multiplier

5) CARRY LOOKAHEAD ADDER:

Carry look ahead adder is also called as a fast adder which is used to reduce computation time and propagation delay. Thus, adding of two binary numbers can be done faster using carry look ahead adder. Carry look ahead adder uses the concepts of generating and propagating carriers. Where the function of carry propagator is to propagated to the next level and the function of carry generator is used to generate the output carry regardless of input carry. Figure 5 shown below is the block diagram of a 4 bit carry look ahead adder. Where the number of gate levels required for the carry propagation is determined by the full adder circuit. AND gate and OR gate is required for the signal from the input carry C0 to output carry C4, which constitutes two gate levels. So, if there are four full adders in the parallel adder, the output carry C4 would have 2x4=8 gate level from C0 to C4. Thus, we can say that, for an n-bit parallel adder, the gate levels required to propagate through are 2n. To construct a carry, look ahead adder the Boolean expressions required are given below. Here in this circuit, we need to generate the two signals carry propagator(p) and carry generator (g): propagator pj = aj $\oplus$ bj generator gj = aj & bj Generalization of Sum (Sj) and Carry (Cj) generation: $Sj = pj \oplus cj$ $cj+1 = gj + pjcj$

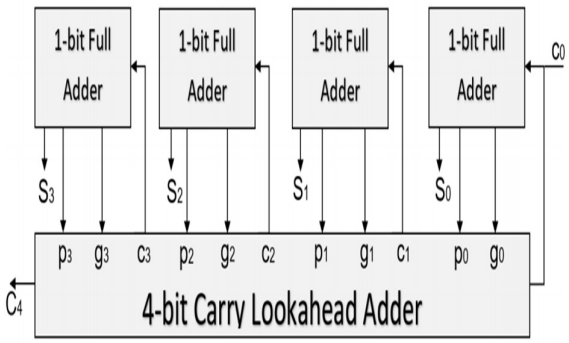

Fig 5: 4-bit Carry Lookahead Adder

B. MODIFIED URDHVA TRIYAKBHYAM:

The Urdhva Triyakbhyam Sutra can be modified to find square of a number with reduced time delay, area and power. Similar to Urdhva Triyakbhyam Sutra, the basic block will be the 2x2 Urdhva Triyakbhyam Multiplier. NxN multiplier is designed using (N/2)x(N/2) multiplier blocks. To find the square of a number, the Urdhva Triyakbhyam Sutra is modified to have 3 multiplier blocks instead of 4 multiplier blocks. Reduction in multiplier block will result in reduced time delay and power. As mentioned earlier, Urdhva Triyakbhyam multiplier is more efficient for higher bits. That is because, as the number of bits increases, the time delay and power of the design increases slowly compared to another stereotyped multiplier. Therefore, this sutra is more advantageous for higher bits multipliers.

1) 4X4 MULTIPLIER:

The design of 4x4 block multiplier is shown in Figure 6. The 2x2 bits multiplier designed for Urdhva Triyakbhyam sutra is used as basic block. As shown in the Figure 6, "a" is the input(multiplicand) which is 4 bits, qi is the partial products and "Q" is the output which is 8 bits. In urdhva triyakbyham sutra, the partial product q1 was obtained by giving a[3:2] and b[1:0] for 2x2 multiplier block and partial product q2 was obtained by giving a[1:0] and b[3:2] for 2x2 multiplier block. In case of squaring, q1 will be equal to q2 that is, a[3:2] x a[1:0] will be equal to a[1:0] x a[3:2]. Therefore, we can eliminate one 2x2 multiplier block. The partial products are then given as input to carry lookahead adder to obtain the final product (Q).

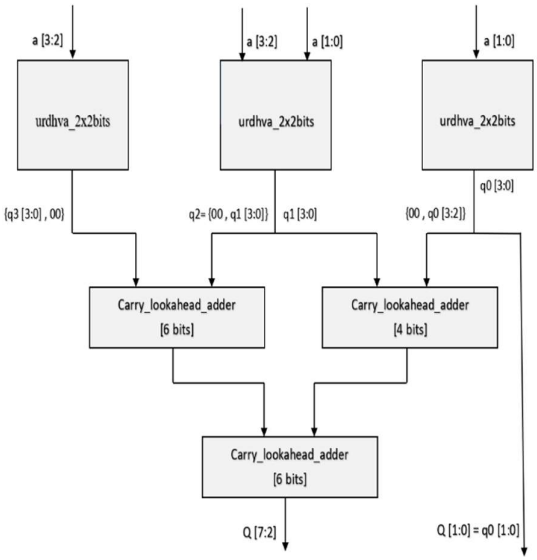

Fig 6: 4x4 Bits Modified Urdhva Triyakbhyam Multiplier

2) 8X8 MULTIPLIER:

Similarly, the design of 8x8 block multiplier is shown in Figure 7. The building blocks used are 4x4 bits multiplier designed for Modified Urdhva Triyakbhyam sutra. As shown in the Figure 7, "a" is the input(multiplicand) which is 8 bits, qi is the partial products and "Q" is the output which is 16 bits. In urdhva triyakbyham sutra, the partial product q1 was obtained by giving a[7:4] and b[3:0] for 4x4 multiplier block and partial product q2 was obtained by giving a[3:0] and b[7:4] for 4x4 multiplier block. In case of squaring, q1 will be equal to q2 that is, a[7:4] x a[3:0] will be equal to a[3:0] x a[7:4]. Therefore, we can eliminate one 4x4 multiplier block. The partial products are then given as input to carry lookahead adder to obtain the final product (Q).

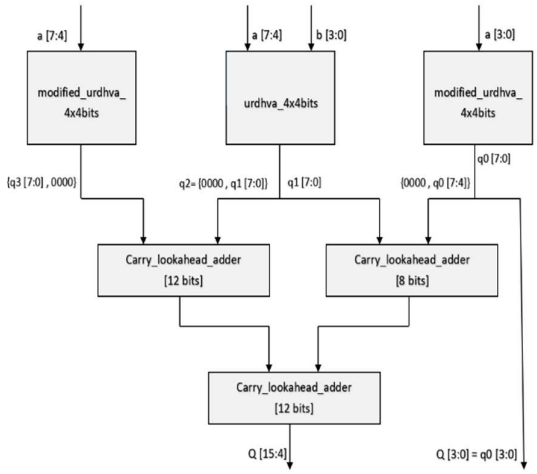

Fig 7:8x8 Bits Modified Urdhva Triyakbhyam Multiplier

3) 16X16 MULTIPLIER:

Similarly, the design of 16x16 block multiplier is shown in Figure 8. The building blocks used are 8x8 bits multiplier designed for Modified Urdhva Triyakbhyam sutra. As shown in the figure, "a" is the input(multiplicand) which is 16 bits, qi is the partial products and "Q" is the output which is 32 bits. In urdhva triyakbyham sutra, the partial product q1 was obtained by giving a[15:8] and b[7:0] for 8x8 multiplier block and partial product q2 was obtained by giving a[7:0] and b[15:8] for 8x8 multiplier

block. In case of squaring, q1 will be equal to q2 that is, a[15:8] x a[7:0] will be equal to a[7:0] x a[15:8]. Therefore, we can eliminate one 8x8 multiplier block. The partial products are then given as input to carry lookahead adder to obtain the final product (Q).

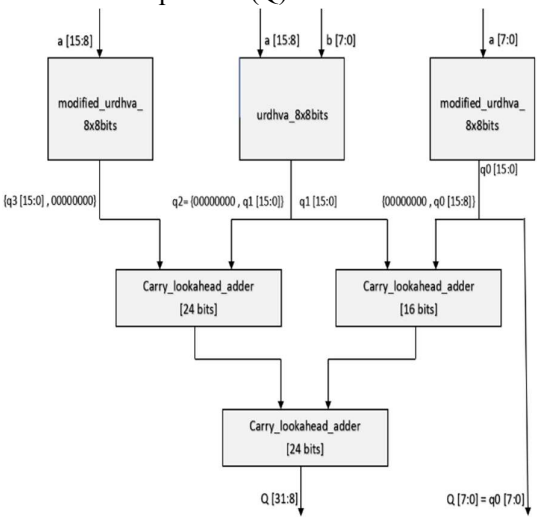

Fig 8: 16x16 Bits Modified Urdhva Triyakbhyam Multiplier

C. EKANYUNENA PURVENA

Ekanyunena Purvena is the technique used in multiplication of any number with 9s series number (Ex: 24565x9999). This sutra helps us to calculate the result in short-cut method by just using subtraction and concatenation. Therefore, the time delay and the area will be reduced.

As shown in the flowchart, a and b are inputs which is 16 bits, t1, t2, t3, t4, t5, t6 and bs (base) are temporary registers and wires, c is output which is in bcd format and is 36 bits. The 1st digit and the last 4 digits of multiplicand used in further calculations (i.e., for step 1 and step 2 as shown in section 2) is obtained by converting it into BCD format. It is then separately converted back to Binary format to perform subtraction. The results of both subtractions are converted back to BCD formats in order to perform concatenation. Therefore, final output is obtained by concatenating these BCD numbers.

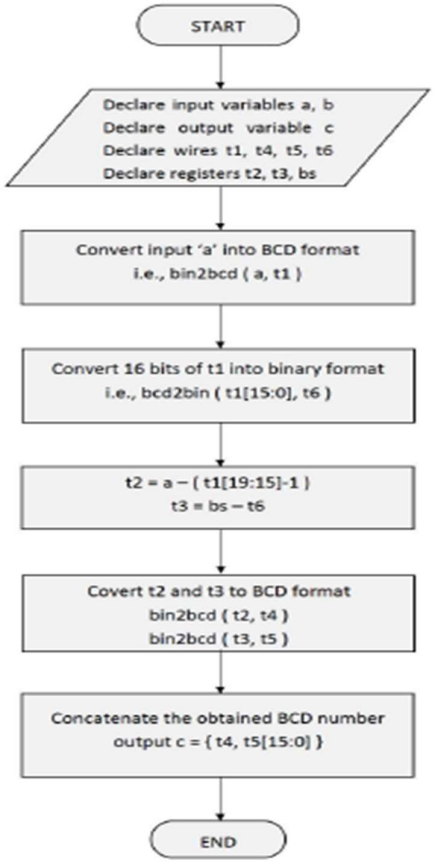

Fig 9: Flowchart for 16x16 Bits Ekanyunena Purvena Multiplier

IV. RESULTS AND DISCUSSIONS

The hardware implementation of Urdhva Tiryakbhyam Sutra, Modified Urdhva Tiryakbhyam Sutra and Ekanyunena Purvena is designed and implemented using electronic design automation (EDA) tool Xilinx Vivado ISE. Simulation and Synthesis of the design is done using the above-mentioned EDA tool. The simulated output of 16-bit Urdhva Tiryakbhyam Sutra, Modified Urdhva Tiryakbhyam Sutra and Ekanyunena Purvena, Booth Multiplier is shown below.

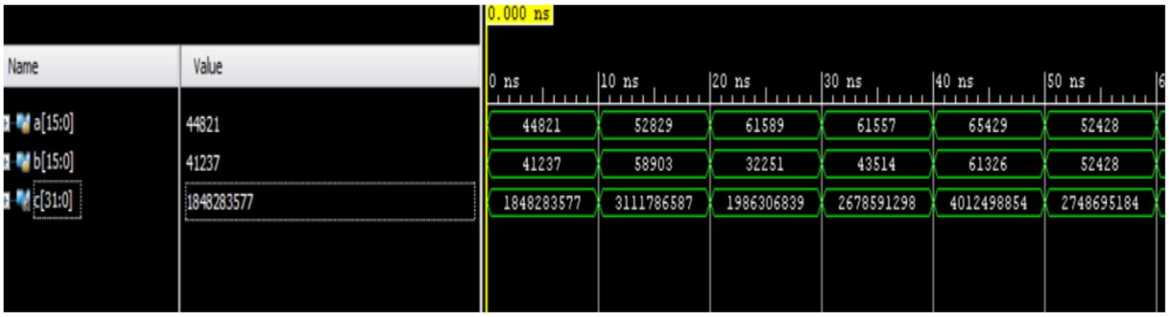

Fig 10: Simulation Result of 16x16 bits Urdhva Triyakbhyam Sutra

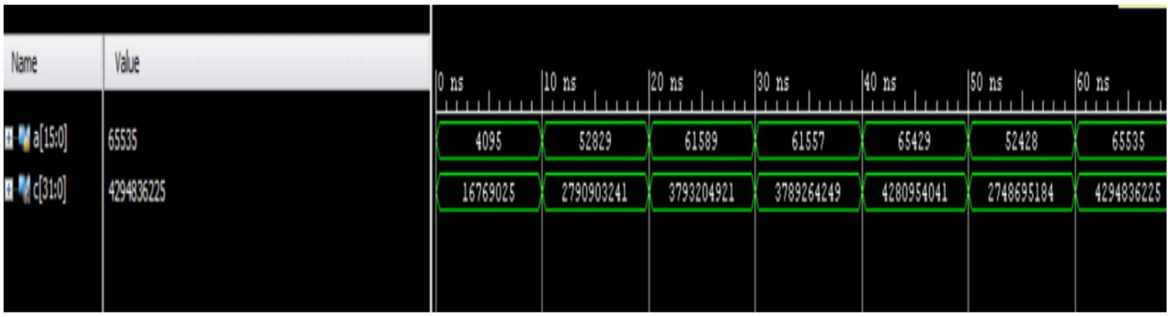

Fig 11: Simulation Result of 16x16 bits Modified Urdhva Triyakbhyam Sutra

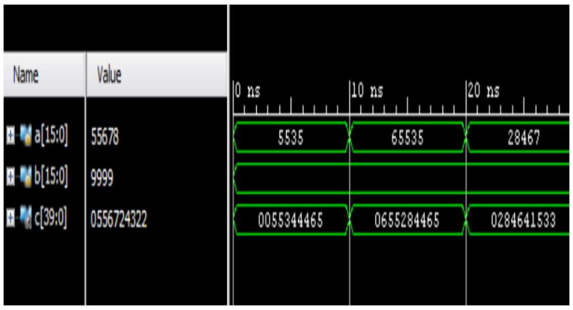

Fig 12: Simulation Result of 16x16 bits Ekanyunena Sutra

V. COMPARISON AND DISCUSSIONS

In our work, both simulation and synthesis has been performed using Xilinx-Vivado software. The propounded Vedic mathematics-based multiplier and the already existing booth multiplier, which uses different algorithm for the multiplication purpose, has been compared in terms of various performance measures such as time delay, power consumption and the number of LUTs used.

Table I below summarizes the comparison of the timing constraints (total delay in terms of ns), Number of LUTs utilised as logic and the total On-chip power consumed while the multiplication process by our Vedic multiplier with that of the conventional booth multiplier.

Table I: Comparison between Vedic multiplier and Booth Multiplier

Algorithm	Time Delay(ns)	Power(W)	LUTs
Urdhva Triyakbhyam	20.523	32.360	357
Ekanyunena Purvena	32.137	41.839	273
Modified Urdhva Triyakbhyam	17.784	28.649	199
Booth Multiplier	62.029	42.935	585

Also, it was evident from the implementation results that the path delay (input pin to output pin delay) significantly reduced when the number of bits were increased from 8 to 16. Additionally, Vedic Architecture requires a lesser number of gates for any given N x N bit implementation, which contributes to small on-chip power when compared to its counterparts.

VI. CONCLUSION

The proposed paper presents a highly methodical multiplication by using ancient Vedic mathematical Algorithms. The objective of the paper, which was to build a multiplier with improved speed and less power usage, was achieved and is evident from the comparison table shown in section V.

Some of the highlights are summarised below-

1) The computation delay for our Vedic multiplier was less than the booth multiplier.

2) Due to lesser numbers of gates required for the Vedic architecture, the on-chip power consumption was very less.

3) Also, the number of LUTs were decreased when compared to the booth multiplier because of less logic on-chip.

This study shows that Vedic algorithms are advantageous in terms of computational benefits, optimization and numbers of steps involved and at the same time, it also reduces the complexity to a large extent because of its simplicity in understanding. With all these merits, the future of the Vedic multiplication in technical industries remains upbeat.

REFERENCES

[1] Pushpa Lata Verma, K. K. Mehta, "Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool", International Journal of Engineering and Advance Technology, Vol.1, no. 5, June, 2012.

[2] Garima Rawat, Khyati Rathore, Siddharth Goyal, Shefali Kala and Poornima Mittal, 'Design and Analysis of ALU: Vedic Mathematics Approach', International Conference on Computing, Communication and Automation (ICCCA2015).

[3] Dravik Kishor Bhai Kahar, Harsh Mehta,' High Speed Vedic Multiplier Used Vedic Mathematics', International Conference on Intelligent Computing and Control Systems ICICCS 2017.

[4] Devika Jaina, Kabiraj Sethi and Rutuparna Panda, 'Vedic Mathematics Based Multiply Accumulate Unit', International Conference on Computational Intelligence and communication systems,2011.

[5]Vedicmathematics(online),link-https://en.wikipedia.org/wiki/Vedic_Mathematics

[6] 'Xilinx ISE User Manual', Xilinx Inc,2007

[7] Parth Mehta, Dhanashri Gawali "Conventional versus Vedic mathematical method for Hardware implementation of multiplier" International Conference on Advances in Computing, Control, and Telecommunication Technologies-2009.

[8] DHARA R. JOSHI Research Scholar, Department of Education, H.N.G.Uni., Patan,' Vedic Mathematics in Modern Era' International Journal of Research in all Subjects in Multi Languages, Vol. 5, Issue: 6, June: 2017 (IJRSML) ISSN: 2321 – 2853.

[9] Vijaya Krishna J, Rajagopala Rao M,' Multiplication of n numbers close to a base using 'Nikhilam' sutra'

[10] Bharati Krishna Tirtha, 1965, Vedic Mathematics, Revised edition 1992.

[11] M.N. Dhanave, M.A. Kangale,' The Implementation of Vedic Mathematics to Algebra and Geometry', IOSR Journal of Mathematics (IOSR-JM) e-ISSN: 2278-5728, p-ISSN:2319-765X. Volume 10, Issue 2 Ver. VII (Mar-Apr. 2014), PP 33-36.

[12] Arushi Somani, Dheeraj Jain, Sanjay Jaiswal, Kumkum Verma and Swati Kash,' Compare Vedic Multipliers with Conventional Hierarchical array of array multiplier', International Journal of computer technology and electronics Engineering (IJCTEE), volume-2, Issue-6