A Comparative Analysis of Multiplier Topologies using Different Vedic Sutras

K. MOUNIKA[1], A. DAMODAR REDDY[2]
[1]PG Scholar, Dept of ECE, K.M.M College of Engineering, Tirupati, AP, India.
[2]Assistant Professor, Dept of ECE, K.M.M College of Engineering, Tirupati, AP, India.

Abstract: The need of low area and high speed Multiplier is increasing as the need of high speed processors are needed. The multipliers used in Square and cube architecture have to be more efficient in area and also in speed. In this paper a multiplier is implemented based on UT and Nikhilam sutra which gives efficient results. The ripple carry adder in the multiplier architecture increases the speed of addition of partial products. Comparison is made between UT and Nikhilam sutra in terms of area (i.e., No of 4-input LUT's and slices) and speed. Synthesis is done on Xilinx FPGA Device using, Xilinx Family: Spartan 3E.

Keywords: Urdhava, Nikhilam, Vedic Maths, FPGA, Array Multiplier.

I. INTRODUCTION

Real time digital signal processing today involves rigorous multiplication operations, which increases the computational complexity of modern day signal processors. The performance of such processors largely rests on the effectiveness of the multiplier units embedded within. A typical multiplier block comprises of a chain of AND gates to generate the partial product terms and an adder assembly to add them. The speed limitation associated with conventional multiplier architectures is largely due to the latency introduced by long adder tree structures. The power consumption of a multiplier unit is also a major design concern. Several research works have been reported over the years to optimize the performance of multiplier topologies. The basic approaches are to reduce the switching activity of the partial products [1], to reduce the number of nonzero partial product terms by effective encoding of the multiplier inputs and to realize a high speed adder tree for fast addition of the partial products. The Booth encoded multiplier topology [2, 3], Wallace tree based multiplier design [4-5] and compressor based multiplier architecture have provided noticeable performance enhancement over the conventional array multiplier. However the use of Vedic mathematics for multiplication [6-10] resulted in significant improvement in the overall speed and power consumption of a multiplier topology, due to the parallel computing approach.

The paper proposes low power multiplier architectures based on Vedic mathematics for high speed computing. The proposed 4-bit and 8-bit multiplier topologies based on the Urdhva Tiryakbhyam (vertically and crosswise) [11] sutra of Vedic mathematics are realized using 45nm CMOS process technology in Cadence EDA tool. A 5T AND gate design, based on pass transistor logic and transmission gate logic, has been used in this work for generation of partial products instead of the conventional 6T CMOS based design. The adder chains used in the multiplier units comprise of 14T full adders and 9T half adders optimized for low power and high speed arithmetic. The use of such modified topologies results in a smaller on chip silicon area requirement. The overall delay associated with the proposed architectures is also reduced as the number of transistors in the critical path is fewer, with the introduction of the new adder topologies. The performance analysis of the proposed designs is carried out for both schematic and layout stages with 1V voltage supply.

II. HISTORY OF VEDIC MATHEMATICS

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus. His Holiness Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications. Swahiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda. Obviously these formulae are not to be found in present text of Atharva Veda because these formulae were constructed by Swamiji himself. Vedic mathematics is not only a mathematical wonder but also it is logical. That"s why VM has such a degree of eminence which cannot be disapproved. Due these phenomenal characteristic, VM has already crossed the boundaries of India and has become a leading topic of research abroad. VM deals with several basic as well as complex mathematical operations. Especially, methods of basic arithmetic are extremely simple and powerful. The word „Vedic" is derived from the word „Veda" which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various

branches of mathematics like arithmetic, algebra, geometry etc. These Sutras along with their brief meanings are enlisted below alphabetically.

- (Anurupye) Shunyamanyat – If one is in ratio, the other is zero.
- Chalana-Kalanabyham – Differences and Similarities.
- EkadhikinaPurvena – By one more than the previous One.
- EkanyunenaPurvena – By one less than the previous one.
- Gunakasamuchyah – The factors of the sum is equal to the sum of the factors.
- Gunitasamuchyah – The product of the sum is equal to the sum of the product.
- NikhilamNavatashcaramamDashatah – All from 9 and last from 10.
- ParaavartyaYojayet – Transpose and adjust.
- Puranapuranabyham – By the completion or noncompletion.
- Sankalana- vyavakalanabhyam – By addition and by subtraction.
- ShesanyankenaCharamena – The remainders by the last digit.
- ShunyamSaamyasamuccaye – When the sum is the same that sum is zero.
- Sopaantyadvayamantyam – The ultimate and twice the penultimate.
- Urdhva-tiryakbhyam – Vertically and crosswise.
- Vyashtisamanstih – Part and Whole.
- Yaavadunam – Whatever the extent of its deficiency.

These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral), and applied mathematics of various kinds. As mentioned earlier, all these Sutras were reconstructed from ancient Vedic texts early in the last century. Many Sub-sutras were also discovered at the same time, which are not discussed here. The beauty of Vedic mathematics lies in the fact that it reduces the otherwise cumbersome-looking calculations in conventional mathematics to a very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing. The multiplier architecture can be generally classified into three categories. First is the serial multiplier which emphasizes on hardware and minimum amount of chip area. Second is parallel multiplier (array and tree) which carries out high speed mathematical operations. But the drawback is the relatively larger chip area consumption. Third is serial-parallel multiplier which serves as a good trade-off between the times consuming serial multiplier and the area consuming parallel multipliers.

III. MULTIPLIER ARCHITECTURE:

The multiplier architectures proposed in this paper are based on UT sutra of Vedic mathematics. The partial product generation in all the designs is realized by 5T pass transistor and transmission gate based AND gates which is described shortly. The use of such a modified AND gate design ensures a significant reduction in the overall on-chip area occupied by the multiplier. As the number of AND gates used for partial product generation increases quadratically with the word length, the use of this modified design plays a greater role for higher order multipliers (32-bit, 64-bit etc.) The adder chain, for the proposed multiplier architectures, incorporates new full adder and half adder topologies that are optimized for low power, high speed and full swing applications. The overall improvement in delay characteristics of the multipliers is attributed to the use of these new adder units which addresses the main performance bottleneck associated with the conventional designs i.e. the latency introduced by the adder assembly. The adder architectures are also discussed below.

A. AND Gate

A 5T based AND gate, given in Fig. 1, is utilized in this paper for the partial product generation. The AND gate utilises pass transistor logic and transmission gate based logic [12]. It is basically a multiplexer based approach which implies the output A AND B, with any arbitrary operands A and B, is same as B for A being equal to logic 1 and is grounded for A being logic 0.
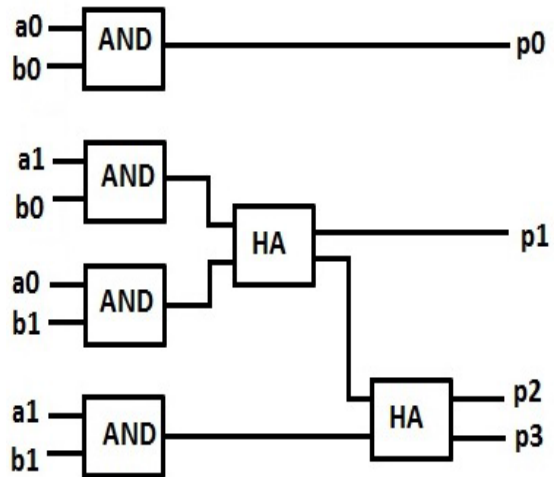

Fig.1. UT based 2-bit Multiplier Architecture.

The design makes use of fewer (i.e. 5) transistors as compared to the conventional 6T based CMOS design and hence results in a reduction of the overall on chip area requirement of the complete multiplier architecture. An 8-bit binary multiplier incorporates 64 AND gates for partial product generation and thus the use of the new AND gate topology brings down the number of transistors by 64 per each multiplier unit. However, the use of such designs does not affect the delay characteristics significantly as the partial product generation is performed parallelly.

B. Half Adder

The half adder block used in this paper is a 9T design that ensures high speed, low power and full swing arithmetic. As shown in Fig.1, the topology uses a 5T AND gate for carry generation while the sum generation unit comprises of a 4T XOR gate. The smaller transistor count ensures a compact design and smaller power dissipation along with an improved delay characteristics. The full swing output logic realization enables the topologies to be used in cascaded arrangements efficiently without the need of any additional swing restoring buffer units. Fig.1 Modified AND gate design.

IV. ALGORITHMS OF VEDIC MATHEMATICS

A. Vedic Multiplication

The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. Vedic multiplication based on some algorithms, some are discussed below:

B. Urdhva Tiryakbhyam Sutra

The multiplier is based on an algorithm Urdhva-Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. UrdhvaTiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Triyakbhyam explained in fig 2. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency.
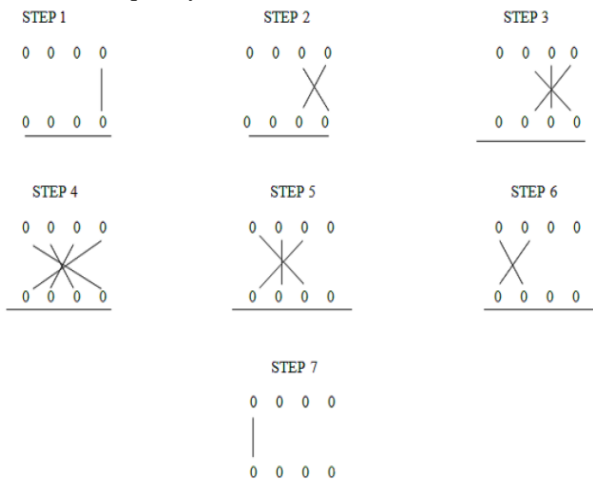

Fig.2. Urdhava Triyakbhyam example.

The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. By adopting the Vedic multiplier, microprocessors designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed.

C. Nikhilam Sutra

Nikhilam Sutra factually means "all from 9 and last from 10". Although it is valid to all cases of multiplication, it is more capable when the numbers involved are big. Since it checks out the compliment of the large number from its adjacent base to perform the multiplication operation on it, better is the original number, lesser the complexity of the multiplication as shown in Fig.3. We first illustrate this Sutra by in view of the multiplication of two decimal numbers (96 * 93) where the chosen base is 100 which is nearest to and greater than both these two numbers
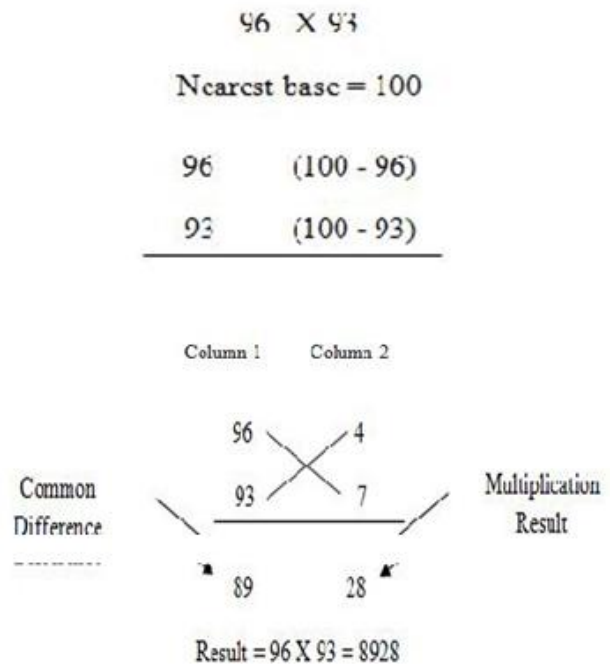

Fig.3. Nikhilam Sutra example.

D. Multiplication Using Nikhilam Sutra

The right hand side (RHS) of the product can be attained by just multiplying the numbers of the Column 2 (7*4 = 28). The left hand side (LHS) of the product can be found by cross subtracting the succeeding number of Column 2 from

the initial number of Column 1 or vice versa, i.e., 96 - 7 = 89 or 93-4=89. The concluding result is attained by concatenating RHS and LHS (Answer = 8928). This sutra is used in this work to find the cube of a number. The number M of N bits having its cube to be calculated is divided in two partitions of N/2 bits, say a and b, and then the Anurupye Sutra is applied to locate the cube of the number. In the above algebraic explanation of the Anurupye Sutra, we have seen that a3 and b3 are to be calculated in the final computation of (a+b) 3. Nikhilam Sutra stipulates subtraction of a number from the nearest power of 10 ie 10, 100, 1000, etc. The power of 10 from which the difference is calculated is called the Base. These numbers are considered to be references to find out whether given number is less or more than the Base. If the given number is 104, the nearest power of 10 is 100 and is the base. Hence the difference between the base and the number is 4, which is Positive and it is called NIKHILAM. The value of Nikhilam may be reference base, the Nikhilam of 87 is -13 and that of 113 is +13 respectively.

Procedure for implementation multiplier using Nikhalam sutra: NIKALAM PROCEDURE for DECIMAL calculation: Follow the procedure as done in KCM and finally divide the LSB and MSB part keep the LSB part as it is and after carry save adder addition multiply with 100 to MSB value if is 2-digit number (or) multiply with 1000 if it is 3-digit value and so on.

KCM Multiplier: The proposed method is based on ROM approach however both the inputs for the multiplier can be variables. In this proposed method a ROM is used for storing the squares of numbers as compared to KCM where the multiples are stored.

Method: To find (a x b), first we have to find whether the difference between 'a' and 'b' is odd or even. Based on the difference, the product is calculated.

- In case of Even Difference

Result of Multiplication= [Average]2- [Deviation] 2

- In case of Odd Difference Result of

Multiplication = [Average x (Average + 1)] -[Deviation x (Deviation+ I)]

Where, Average = [(a+b)/2] and Deviation = [Average - smallest(a, b)]

Example3(Even difference) and Example 4 (Odd difference) depict the multiplication process. Thus the two variable multiplicationsare performed by averaging, squaring and subtraction. To find the average[(a+b)/2], which involves division by 2 is performed by right shifting the sum by one bit. If the squares of the numbers are stored in a ROM, the result can be instantaneously calculated. However, in case of Odd difference, the process is different as the average is a floating point number. In order to handle floating point arithmetic, EkadikenaPurvena - the Vedic Sutra which is used to find the square of numbers end with 5 is applied. Example 5 illustrates this. In this case, instead of squaring the average and deviation, [Average x (Average + 1)] - [Deviation x (Deviation+ I)] is used. However, instead of performing the multiplications, the same ROM is used and using equation (10) the result of multiplication is obtained.

$$n (n+l) = (n2+n)$$

Example 3: 16 x 12 = 192
- Find the difference between (16 -12) = 4 -7 Even Number
- For Even Difference, Product = [Average] 2- [Deviation] 2
  - Average = [(a+b)/2]=[(16+12)12] = [28/2] = 14
  - Smallest (a, b) = smallest (l6, 12) =12
  - Deviation = Average-Smallest(a,b) = 14 -12 =2

- Product = 142-22= 196 - 4 = 192

Example 4: 15 x 12 = 180

- Find the difference between (15-12) =3 -7 Odd Number
- For Odd Number Difference find the Average and Deviation.
  - Average = [(a+b)/2] = [(12+ 15)/2] = 13.5

  - Deviation = [Average - smallest (a, b)]

    = [12.5 - smallest (l3, 12)] = [1 3.5 - 12] = 1.5

- Product = (13x14) - (1x2) = 182 - 2 =180

V. SIMULATION RESULTS FOR UT SUTRA
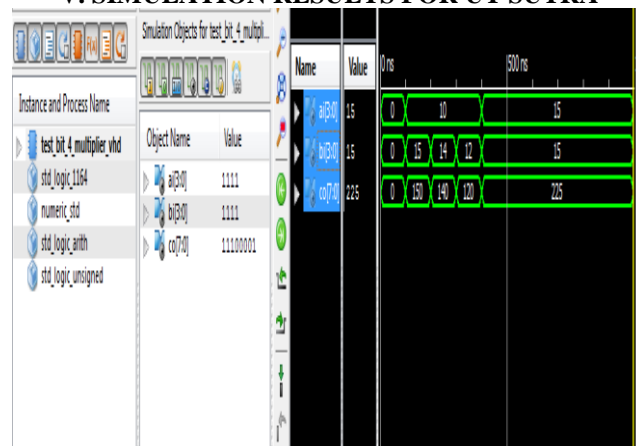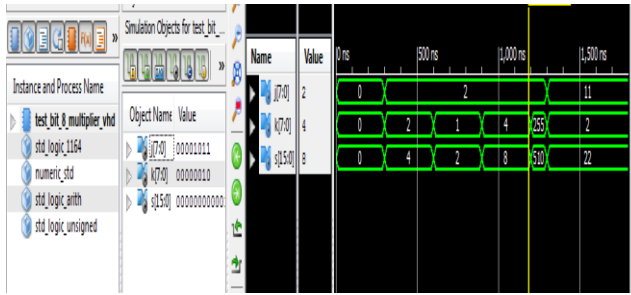

Fig.4. 4-bit.

In the above simulation result we can observe the result as225 for applied stimulus as a=15 and b=15 as shown in Fig.4.
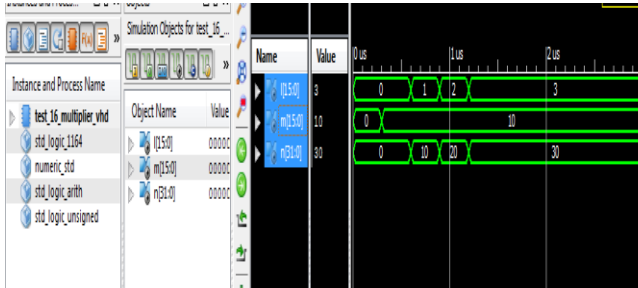
Fig.5. 8-bit.

In the above simulation result we can observe the result as 8 for applied stimulus as a=2 and b=4 as shown in Fig.5.
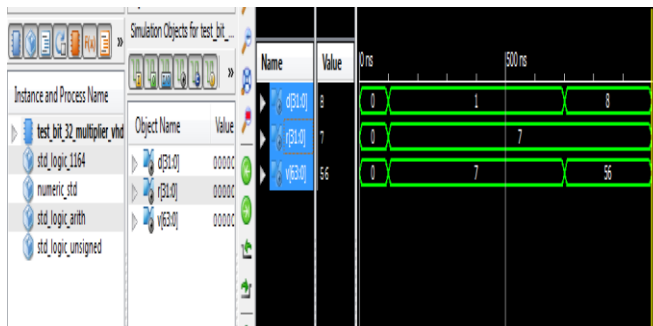

Fig.6. 16-bit.

In the above simulation result we can observe the result as 30 for applied stimulus as a=3 and b=10 as shown in Fig.6.

Fig.7. 32-bit.

In the above simulation result we can observe the result as 56 for applied stimulus as a=8 and b=7 as shown in Fig.7.

Fig.8. 64-bit.

In the above simulation result we can observe the result as 12 for applied stimulus as a=12 and b=1 as shown in Fig.8.

VI. SIMULATION RESULTS FOR NIKHALAM SUTRA

Fig.9. 4-bit.

In the above simulation result we can observe the result as 196 for applied stimulus as a=14 and b=14 as shown in Fig.9.

Fig.10. 8-bit.

In the above simulation result we can observe the result as 194 for applied stimulus as a=97 and b=2 as shown in Fig.10.
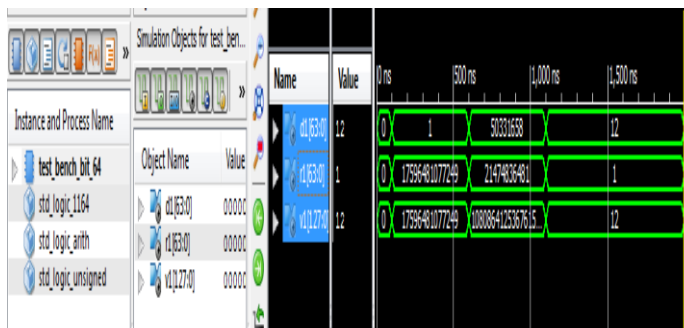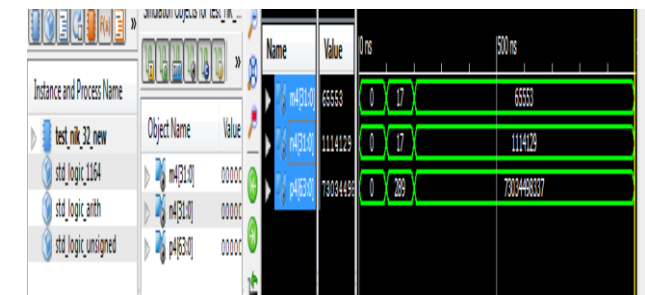

Fig.11. 16-bit.

In the above simulation result we can observe the result as 5130 for applied stimulus as a=2565 and b=2 as shown in Fig.11.

Fig.12. 32-bit.

In the above simulation result we can observe the result as 7303449337 for applied stimulus as a=65553 and b=1114129 as shown in Fig.12.
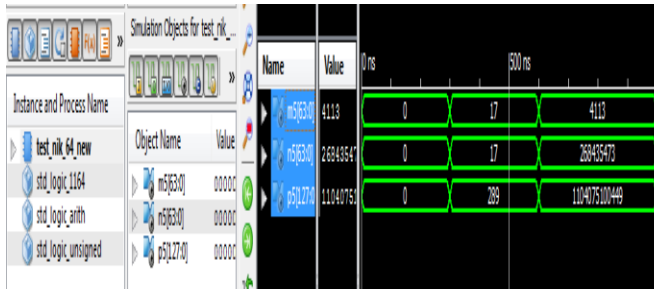

Fig.13. 64-bit.

In the above simulation result we can observe the result as 110407508811 for applied stimulus as a=4113 and b=26843547 as shown in Fig.13.

TABLE I: Comparison Table

	Urdhva Tiryakbhyam			Nikhilam		
	No of LUTs	Delay	NO of silces	No of LUTs	Delay	No of silces
4-bit	34	12.41ns	19	16	13.297ns	9
8-bit	174	22.216ns	98	33	15.988ns	19
16-bit	768	41.761ns	437	73	21.026ns	41
32-bit	3222	75.150ns	1831	261	36.747ns	141
64-bit	13202	132.816ns	7497	921	62.236ns	483

VII. CONCLUSION

Thus the proposed multiplier provides higher performance for higher order bit multiplication. In the proposed multiplier for higher order bit multiplication i.e. for 8x8 and more, the multiplier is realized by instantiating the lower order bit multipliers like 4x4. This is mainly due to memory constraints. Effective memory implementation and deployment of memory compression algorithms can yield even better results in terms of area and speed which improves the overall performance of the design.

VIII. REFERENCES

[1] N.-Y. Shen and O. T.-C.Chen, "Low-power multipliers by minimizing switching activities of partial products", Proc. IEEE, ISCAS 2002, vol.4, pp. 93–96, May 2002.

[2] A.D. Booth, "A Signed Binary Multiplication Technique", Qrt. J. Mech. App. Math., vol. 4, 1951, pp. 236–240.

[3] J. Hu, L. Wang, and T. Xu, "A Low-Power Adiabatic Multiplier Based on Modified Booth Algorithm", Proc. IEEE, ISIC'07, pp. 489-492, Sept. 26-28, 2007.

[4] C. S Wallace, "A Suggestion for a Fast Multiplier," IEEE Trans. On Computers, vol. EC13, pp. 14-17, December 1964

[5] J. M. Rabaey, A. Chandrakasan and B. Nikolic, " Digital integrated circuits-A design perspective", second edition, PHI Learning Pvt. Ltd.

[6] DevikaJaina, KabirajSethi, Rutuparna Panda, "Vedic mathematics based multiply accumulate unit", International conference on computational intelligenceand communication networks",pp. 754- 757,7-9 October 2011.

[7] L. Sriraman, T. N. Prabakar, "Design and implementation of two variable multiplier using KCM and vedicmathematics", 1st International conference on recent advances in information technology, pp.782- 787,15-17 March 2012.

[8] Pavan Kumar U. C. S., Sai Prasad G. A., A. Radhika, "FPGA Implementation of high speed 8-bit Vedic multiplier using barrel shifter", IEEE proceedings on international conference on energy efficient technologies for sustainability, 2013, pp 14-17.

[9] M. Ramalatha, K. Deena Dayalan, P. Dharani, S. Deborah Priya, "High speed energy efficient ALU design using vedic multiplication techniques", ACTEA, July 15 - 17, 2009, Lebanon.

[10] Sushma R. Huddar, Sudhir Rao, Kalpana M., Surabhi Mohan, "Novel high speed vedic mathematics multiplier using compressors", IEEE, 2013.

[11] Swami BharatiKrisnaTirthaji Maharaja, "Vedic mathematics", original edition, MotilalBanarasidass publishers.

[12] Sung-Mo Kang and Yusuf Leblebici, "CMOS digital integrated circuitsanalysisand design", third edition, Tata McGraw-Hill education Pvt.Ltd.

[13] SuryasnataTripathy, L B Omprakash, B. S. Patro, Sushant K. Mandal, "A comparative analysis of different 8-bit adder topologies at 45 nm technology", International journal of engineering research andtechnology, vol. 2, Issue 10, October 2013.

[14] SuryasnataTripathy, L B Omprakash, B. S. Patro, Sushanta K. Mandal, "Low power, high speed full adder architectures in 45nm technology", International conference on VLSI and signal processing, IIT Kharagpur, Jan 10-12, 2014.

[15] D. Radhakrishnan, "Low-voltage low-power CMOS full adder", Proc. Inst. Elect. Eng., Circuitts Devices and Systems, vol. 148, no. 1, pp. 19- 24, 2001.

[16] Deepa Sinha, Tripti Sharma, K.G.Sharma, Prof.B.P.Singh, " Design and analysis of low power 1 bit full adder cell", IEEE 3rd International Conference on Electronics Computer Technology (ICECT) , vol.2, pp.303-305, 8-10 April 2011.

[17] SuryasnataTripathy, Sushanta K. Mandal, " High speed low power datapath design using multi threshold logic in 45 nm technology for signal processing application", National Conference on VLSI Signal Processing and Trends in Telecommunication (VSATT-2014), C V Raman College of Engineering, Odisha, May 9-10, 2014.

[18] C.Senthilpari, Ajay Kumar Singh and K.Diwakar Member, "low power and high speed 8×8 bit multiplier usingnon-clocked pass transistor logic", International conference on intelligent and advanced systems 2007, pp 1374-1378.

[19] YavuzDelican, TülayYildirim, "High performance 8-bit mux based multiplier using Mos current mode logic", IEEE proceedings on 7[th]International conference on Electrical and Electronics Engineering, 2011, pp II-89 – II-93.

[20] Youn Sang Lee, JeongBeom Kim, "Design of a low power 8×8 bit parallel multiplier using Mos current mode

logic circuit ", IEEE proceedings on 8th international conference on solid state and integrated circuit technology, 2006, pp 1502-1504

[21] Swami BharatiKrshnaTirthaji, Vedic Mathematics. Delhi: MotilalBanarsidass Publishers, 1965.

[22] K.K.Parhi "VLSI Digital Signal Processing Systems - Design and Implementation" John Wiley & Sons, 1999.

[23] Harpreet Singh Dhillon and Abhijit Mitra "A Digital Multiplier Architecture using Urdhava Tiryakbhyam Sutra oj Vedic Mathematics" IEEE Conference Proceedings,200S.

[24] AsmitaHaveliya "A Novel Design ./i)r High Speed Multiplier .fir Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach)" International Journal of Technology And Engineering System (IJTES):Jan - March 2011- Vo12 .Nol.