

## High Speed and Energy Efficient ALU Design using Ancient Computational Technique

Garima

Assistant Professor, Department of ECE, Maharaja Surajmal Institute of Technology, Delhi, India  
1592garima@gmail.com

**Abstract:** An Arithmetic logic unit (ALU) is the heart of all digital microprocessors. All high performance systems such as FIR filters, digital signal processors require a faster and energy efficient ALU to perform arithmetic operations with greater speed. Area and speed being the two conflicting constraints in VLSI design. This paper aims to find out the best trade off solution among both of them. While comparing the conventional ALU designs, Vedic ALU turns out to be the fastest amongst all.

**Keywords:** Vedic, Arithmetic Logic Unit (ALU), Area, Speed

### I. INTRODUCTION

The design of high speed integrated circuits is a long and complex operation. In accordance with Moore's law, the scale of integration is growing, thus more and more sophisticated systems are being implemented on a single VLSI chip. This leads to an increase in the packing density but power consumption also increases. These signal processing applications not only demand greater computation capacity but also consume considerable amount of energy. While speed and Area remain to be the two major design criteria, power consumption has also become a critical concern in today's VLSI system design.

The microprocessor is central processing unit of all microcomputers. It contains ALU as its one of the major units to perform all arithmetic and logic operations. Because of their small size, low cost and low power consumption, microprocessors have revolutionized the digital computer system technology. The speed of microprocessor determines the maximum speed of microcomputer. Hence faster the components of a microprocessor work, greater will be the speed of operation. With ever increasing need for higher clock frequency, it becomes imperative to design a faster arithmetic unit. To perform arithmetic operations speedily, we have used Vedic Mathematics techniques to design a faster and energy efficient ALU.

Vedic Mathematics contains multiple algorithms for one arithmetic operation [1]. We have used some of these algorithms to design various modules for division multiplication, addition and subtraction for ALU.

Vedic mathematics is the name given to the ancient Indian system of mathematics that was rediscovered in the early twentieth century from ancient Indian sculptures (Vedas) [2]. It mainly deals with Vedic mathematical formulae and their application to various branches of mathematics. The algorithms based on

conventional mathematics can be simplified and even optimized by the use of Vedic Sutras [3]. Vedic Mathematics is a system of mental calculation that is used to reduce lengthy and cumbersome arithmetic operations into simplified calculations. Vedic Mathematics consists of 16 sutras or aphorisms. Vedic Sutras deal with various branches of mathematics like arithmetic, algebra, geometry etc. These Sutras are enlisted below:

1. (Anurupya) Shunyamanyat
2. Chalana-Kalanabyham
3. Ekadhikina Purvena
4. Ekanyunena Purvena
5. Gunakasamuchyah
6. Gunitasamuchyah
7. Nikhilam Navatashcaramam Dashatah
8. Paraavartya Yojayet
9. Puranapuranabyham
10. Sankalana-Vyavakalanabhyam
11. Shesanyankena Charamena
12. Shunyam Saamyasamuccaye
13. Sopaantyadvayamantyam
14. Urdhva-tiryakbhyam
15. Vyashtisamanstih
16. Yaavadunam

Swami Bharati Krishna Tirthaji (1884-1960), former Jagadguru Sankaracharya of Puri culled a set of 16 Sutras (aphorisms) and 13 Sub - Sutras (corollaries) from the Atharva Veda [4]. He developed methods and techniques for amplifying the principles contained in the aphorisms and their corollaries, and called it Vedic Mathematics. The Sutras apply to and cover almost every branch of Mathematics. They apply even to complex problems involving a large number of mathematical operations [5]. Application of the Sutras saves a lot of time and effort in solving the problems, compared to the formal methods presently in vogue.

The proposed ALU design uses Urdhva-Tiryakbhyam method for Multiplication as well as division [6]. Anurupya Sutra has been used for implementing Cubing circuit in proposed Vedic ALU.

Rest of the paper is arranged as follows: section II explains conventional ALU design, section III explains vedic ALU design, section IV deals with detailed simulation analysis and the last section concludes the paper.

## II. CONVENTIONAL ALU DESIGN

The Basic ALU architecture consists of four different modules each for a different arithmetic operation – Addition, Subtraction, Multiplication and Division. Fig. 1 shows the basic modules of 8 bit Conventional ALU. a [7:0], b [7:0] are 8 bit inputs, f [15:0] is the output.

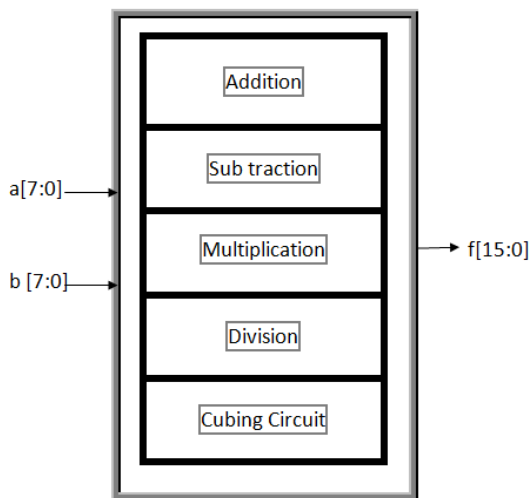


Fig. 1. Block diagram of Conventional ALU

Addition is the fundamental arithmetic operation used in all microprocessors, digital signal processors, especially digital computers. Also, it serves as a building block for synthesis of all other arithmetic operations. In the conventional ALU Design, we have used Ripple carry adders for performing addition operation.

The most straight forward implementation of final stage adder is Ripple Carry Adder in which cascaded full adders are used carry generated in previous full adder works as input carry for next stage full adder. Fig 2 shows that N bit Ripple Carry Adder require N full adders.

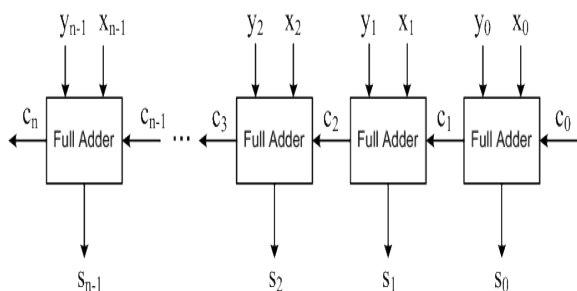


Fig. 2. Block Diagram of RCA

Drawbacks of using Ripple Carry Adder:

- It is not efficient when large numbers of bits are used.
- Carry propagation delay increases linearly with bit length as next stage output is dependent on previous stage output.

Logic Equations:

$$C_i = x_i \& y_i$$

$$P_i = x_i \wedge y_i$$

$$S_i = P_i \oplus C_i$$

The Subtraction operation in conventional ALU has been performed by using the subtraction operator '-'. The function of subtraction module is to subtract the subtrahend (b [7:0]) from minuend (a[7:0]) and to generate a difference term (f [15:0]).

Subtraction is alternatively addition of a negative number with a positive number. In computer, first 2's complement of negative number is taken and added to the positive number.

Suppose, M is Minuend and N is subtrahend

Then,  $M - N$  can be done based on three steps discussed next:

Step 1: Take 2's complement of N and add it to M.

$$M - N = M + (2^n - N)$$

Step 2: If M is greater than or equal to N then end carry is discarded from the result.

$$M - N = M + (2^n - N) - 2^n$$

Step 3: If M is less than N then take 2's complement of the result and append negative '-' sign in front

$$\text{So binary subtraction formula, } M - N = (-) [2^n - (M + (2^n - 1))]$$

Digital multiplication is a series of bit shifts and series of bit additions, where the two numbers, the multiplicand and the multiplier are combined into the result. Considering the bit representation of the multiplicand  $x = x_{n-1} \dots x_1 x_0$  and the multiplier  $y = y_{n-1} \dots y_1 y_0$  in order to form the product up to n shifted copies of the multiplicand are to be added. The entire process consists of three steps partial product generation, partial product reduction and addition. Conventional ALU comprises of conventional multiplication module which makes use of successive addition and shifting of partial products. This technique works well for small numbers but it becomes difficult to follow in case of large numbers (generally greater than 3 digit numbers). It requires lot of memory space to store the partial products and then adding them all further requires adders at every stage. This amounts to an

increase in the delay, hence decrease in computational speed.

The cubing operation is one of the most important operations in arithmetic process and it is found to be complicated, as we go for higher radix numbers. Cubing operation can be performed using ordinary multipliers, which are scalable but they have a larger delay. Structure based array implementations are faster but scalability increases design complexity as well as expense. Moreover, multipliers occupy large area, have long latency and consume considerable power. Therefore, multipliers which offer either of the following design targets-scalability, re-configurability, high speed, low power consumption, regularity of layout and less area or even a combination of some of these features are used.

### III. PROPOSED VEDIC ALU ARCHITECTURE

All In Proposed Vedic ALU, multiplication operation has been performed using Vedic Mathematics algorithm-Urdhvatiiryagbhyam sutra and division has been performed by using a combination of two sutras-Urdhvatiiryagbhyamsutra and Dhvanjaka sutra [7]. These sutras provide hands on calculation to solve lengthy computations. Both of these sutras are applicable in all cases of multiplication and division and also yield faster results. Addition and Subtraction have been done using '+' and '-' operators.

Fig. 3 shows the proposed Vedic ALU with subsequent Vedic Multiplication and division modules using Urdhva – Tiryagbhyam sutra for Multiplication and a combination of Urdhva – Tiryagbhyam and Dhvanjaka Sutra for Division. In this fig., a [7:0], b[7:0] are inputs and f[15:0] is the output.

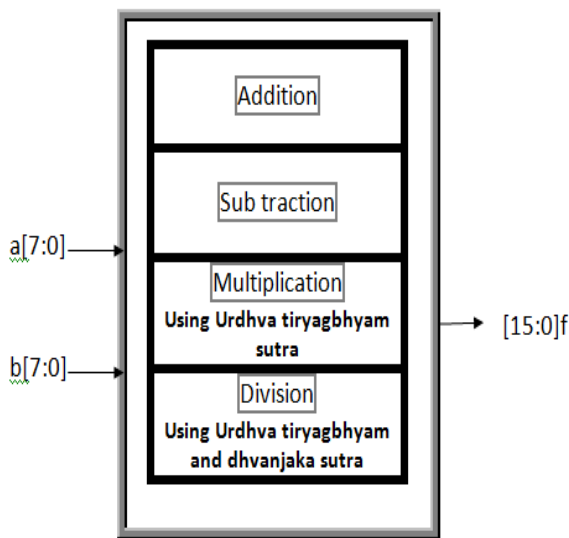


Fig. 3. Block Diagram of Proposed Vedic ALU

#### A. Multiplication- UrdhvaTiryagbhyam Sutra:

Urdhva – Tiryagbhyam is the general formula applicable to all cases of multiplication and also in the division of a large number by another large number [8].

The 4 x 4 multiplication has been done in a single line in this method, whereas in successive shift and add method (Conventional) four partial products have to be added to get the result.

Vedic technique has an advantage over the conventional technique as the no. of adders have been reduced. Also, it enhances the computation speed. The Vedic multiplication is implemented using the following equations:-

$$X = a_3 a_2 a_1 a_0$$

$$Y = b_3 b_2 b_1 b_0$$

$$P_0 = a_0.b_0$$

$$P_1 = a_1.b_0 + a_0.b_1$$

$$P_2 = a_2.b_0 + a_1.b_1 + a_0.b_2 + P_1(1)$$

$$P_3 = a_3.b_0 + a_2.b_1 + a_1.b_2 + a_0.b_3 + P_2(2 \text{ to } 1)$$

$$P_4 = a_3.b_1 + a_2.b_2 + a_1.b_3 + P_3(2 \text{ to } 1)$$

$$P_5 = a_3.b_2 + a_2.b_3 + P_4(2 \text{ to } 1)$$

$$P_6 = a_3.b_3 + P_5(2 \text{ to } 1)$$

Product = P6 & P5(0) & P4(0) & P3) & P(2) & P(1) & P(0) &- Concatenate

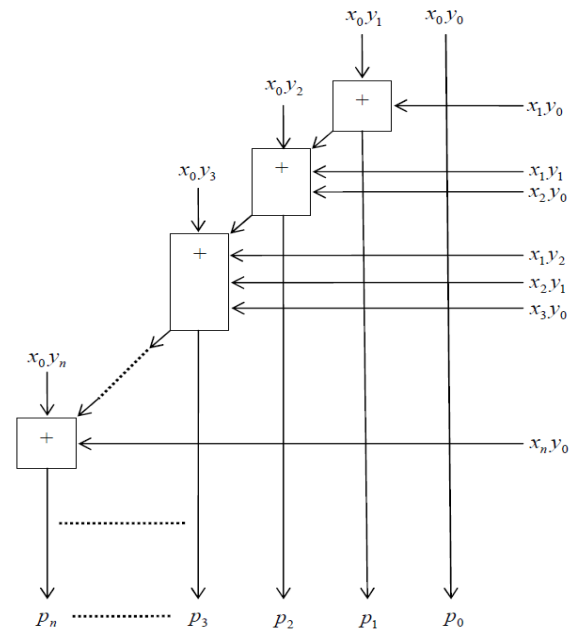


Fig. 4. Vedic Multiplier

Fig.4 shows how multiplication is being done using Vedic sutra. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array [9, 10].

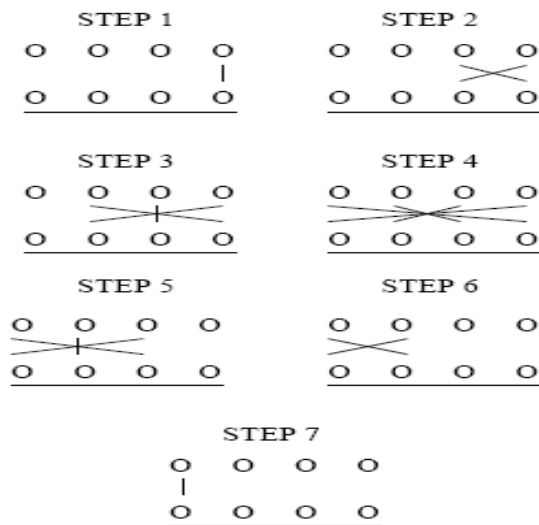


Fig. 5. Line Diagram for Vedic Multiplication

Fig.5 depicts how multiplication is being done using ancient computational techniques. Hence this is the general mathematical formula applicable to all cases of multiplication.

All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array.

#### B. Division- Urdhva Tiryagbhyam Sutra and Dhvanjaka Sutra:

The word Dhvanjaka means "on the top of the flag". For Example  $43852 \div 54$

**Step 1:** Put down the first digit (5) of the divisor (54) in the divisor column as operator and the other digit (4) as flag digit. Separate the dividend into two parts where the right part has one digit. This is because the flag digit is single digit.

**Step 2:** (i) Divide 43 by the operator 5. Now  $Q=8$  and  $R=3$ . Write this  $Q=8$  as the 1<sup>st</sup>

Quotient - digit and prefix  $R=3$ , before the next digit i.e. 8 of the dividend, as shown below.

Now 38 becomes the gross-dividend ( G.D. ) for the next step.

(ii) Subtract the product of flag digit (4) and first quotient digit (8) from the G.D. (38) i.e.  $38-(4 \times 8)=38-32=6$ . This is the net - dividend (N.D) for the next step.

**Step 3:** Now N.D Operator gives Q and R as follows.  $6 \div 5$ ,  $Q=1$ ,  $R=1$ . So  $Q=1$ , the second quotient-digit and  $R=1$ , the prefix for the next digit (5) of the dividend.

**Step 4:** Now G.D = 15; product of flag-digit (4) and 2nd quotient - digit (1) is  $4 \times 1=4$  Hence  $N.D=15-4=11$

divide N.D by 5 to get  $11 \div 5$ ,  $Q=2$ ,  $R=1$ . The representation is

**Step 5:** Now the R.H.S part has to be considered. The final remainder is obtained by subtracting the product of flag-digit (4) and third quotient digit (2) from 12 i.e.,  $12 - (4 \times 2) = 12 - 8 = 4$ .

Thus  $43852 \div 54$  gives  $Q=812$  and  $R=4$ .

Consider the algebraic proof for the above problem. The divisor 54 can be represented by  $5x+4$ , where  $x=10$ .

The dividend 43852 can be written algebraically as  $43x^3 + 8x^2 + 5x + 2$  since  $x^3 = 10^3 = 1000$ ,  $x^2 = 10^2 = 100$ .

This method results in fast on hand computational result checking.

#### C. Cubing Circuit- Anurupya Sutra:

Cubing of 2-digit numbers can also be performed using a sutra called the Anurupya Sutra [11]. To use this, follow the procedure below:

- Put down the cube of the left digit of the number to be cubed as the left most number in a row of 4 numbers.
- Put down the square of the left digit multiplied by the right digit as the second number in the same row of numbers.
- Put down the square of the right digit multiplied by the left digit as the third number in the same row of numbers.
- Put down the cube of the right digit as the right most number in this row of numbers.
- Under the second number in the row above, put down twice the second number.
- Similarly, under the third number in the first row, put down twice the third number.
- Add them up, making sure to carry over excess digits from right to left. That is the final answer.

According to this sutra,

$$(ab)^3 = a^3 + 3a^2b + 3ab^2 + b^3.$$

Where a, b are the digits of the number. Here (+) does not indicate ordinary addition. Hence the cube of a number can be found by applying the sutra.

**Example:** Let us consider a two digit decimal number 43. According to the sutra,

**Step 1:** Let  $a=4$  and  $b=3$

**Step 2:** Applying the sutra,  $(43)^3 = (4)^3 + 3(4)^2(3) + 3(4)(3)^2 + (3)^3$

**Step 3:** Now add the partial products from right by

shifting them one digit.

i.e.  $b^3 = (3*3*3) = 27$

$3b^2a = (3*3*3*4) = 108$

$3a^2b = (3*4*3*4) = 144$

$a^3 = (4*4*4) = 64$

$(ab)^3 = 79507$

From the above operations, it is obvious that multipliers are to be employed. There are numerous multiplication techniques, which are employed in processors and other applications. The choice of one is based on their structure, easy application and performance.

#### IV. RESULT ANALYSIS

The design of conventional ALU and proposed Vedic ALU is being implemented using Verilog HDL (hardware description language) in Xilinx ISE Design Suite.

Device utilization summary gives us the information about number of occupied slices of Flip Flops number of bonded IOBs and LUTs out of the maximum range available as shown in table I for Conventional ALU whereas table II for Vedic ALU .

Table I. Device Utilization Summary of Conventional ALU

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	2,914	3,840	75%
Number of occupied Slices	1,467	1,920	76%
Total Number of 4 input LUTs	2,915	3,840	75%
Number of bonded IOBs	49	173	28%
Number of BUFGMUXs	1	8	12%

Table III. Device Utilization Summary of Vedic ALU

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	13,763	29,504	46%
Number of occupied	7,099	14,752	48%

Slices			
Total Number of 4 input LUTs	13,801	29,504	46%
Number of bonded IOBs	46	250	18%
Number of BUFGMUXs	1	24	4%

The basic purpose of this work was to know the time and power trade-offs between different ALU implementation. Both theoretical and practical comparisons of all the ALU Designs taken into consideration are shown in Table III.

Table IIIII. Detailed Comparison Analysis

S. N.		Conventional ALU	Proposed Vedic ALU
1	Combinational Path Delay(ns)	24.743	6.912
2	Maximum Frequency (MHz)	100.321	238.623
3	Clock Delay (ns)	6.044	4.19
4	On Chip Power Utilization(mW)	18.49	79.27

#### V. CONCLUSIONS

From the above comparison it is clear that Vedic ALU has the least combinational delay therefore it is the fastest. So, proposed Vedic ALU is better than conventional ALU for high speed applications. ALU implementation using Vedic Mathematics resulted in least computational delay.

#### VI. REFERENCES

- [1] Jagadguru Swami Sri Bharati, Krishna Tirathji, "Vedic Mathematics or Sixteen Simple Sutras From The Vedas", MotilalBanarsidas, Varanasi (India),1986.
- [2] Gopinath, K. Jini, and R. Krishnan. "Vedic mathematics training in specific learning difficulty: A study on upper primary children." Indian Journal of Positive Psychology 9, no. 1, pp 97-102, 2018.
- [3] Tirtha, S. B. Krishna, and VasudevaSharanaAgrawala. Vedic mathematics. Vol. 10. MotilalBanarsidass Publ., 1992.
- [4] Akhter, Shamim, and S.Chaturvedi. "Modified Binary Multiplier Circuit Based on Vedic Mathematics." In 2019 6th International

Conference on Signal Processing and Integrated Networks (SPIN), pp. 234-237. IEEE, 2019.

- [5] Barve, Sampada, S. Raveendran, C. Korde, T.Panigrahi, and M. H. Vasantha. "FPGA Implementation of Square and Cube Architecture Using Vedic Mathematics." In 2018 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), pp. 6-10. IEEE, 2018.
- [6] B.N.K Reddy, Design and implementation of high performance and area efficient square architecture using Vedic Mathematics. Analog Integrated Circuits and Signal Processing, pp.1-6, 2019.
- [7] Vamsi, AkellaSrinivasa Krishna, and S. R. Ramesh. "An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics." 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2019.
- [8] Kumar, G. Ganesh, and V. Charishma. "Design of high speed vedic multiplier using vedic mathematics techniques." International Journal of Scientific and Research Publications 2, no. 3, 2012.
- [9] Jaina, Devika, K. Sethi, and R. Panda. "Vedic mathematics based multiply accumulate unit." In 2011 International Conference on Computational Intelligence and Communication Networks, pp. 754-757. IEEE, 2011.
- [10] Saha, Prabir, A. Banerjee, P. Bhattacharyya, and A.Dandapat. "High speed ASIC design of complex multiplier using vedic mathematics." In IEEE Technology Students' Symposium, pp. 237-241. IEEE, 2011.
- [11] M. Ramalatha, K. Deena Dayalan, P. Dharani, and S. Deborah Priya. "High speed energy efficient ALU design using Vedic multiplication techniques." In 2009 International Conference on Advances in Computational Tools for Engineering Applications, pp. 600-603. IEEE, 2009.