

High Performance Low Power Arithmetic and Logic Unit: A Trade off

Neha Narula¹ and Shruti Kalra²

^{1,2}Jaypee Institute of Information Technology
 E-mail: ¹nehanarula1992@gmail.com

Abstract—Arithmetic Logic Unit (ALU) is one of the essential unit in general purpose processors and major source of power dissipation. Amongst all the units available in the ALU, major source of power dissipation are adder and multiplier block. Various architectures are available in the literature that compares the performance in terms of area, delay and power for different adders and multipliers. Moreover, the general trend is to compare either one of the architecture with two or three existing one. In this paper, a comparison has been made between all the adders and the multipliers available in the literature on 90nm technology node. The results are obtained on Synopsys Design Vision. This helped in deciding the trade-off to be kept between power, performance and area which will in turn lead to an optimized design for ALU at lower technology node.

1. INTRODUCTION

In electronic world, systems are becoming fast and powerful but more power consuming. The main consequences of high power consumption are addition of cooling circuits that reduces life time of battery for embedded systems [1]. ALU is basic fundamental building block of CPU and main source of power dissipation. ALU performs the arithmetic and logic operations. When we talk about the best ALU, it should consume low area, should have high performance and dissipate low power.

In this paper, a comparison has been made on various architectures of adders and multipliers available in the literature and obtained an optimized ALU based on the application by use of optimized adder and multiplier block. A tradeoff has to be kept between power, area and performance and therefore, a comparison based on energy, delay and area is being done on one platform. As per the result obtained in [1], square root carry select adder (SQRT-CSA) has less delay but more power consumption, ripple carry adder has less power consumption but low speed. But common boolean Logic adder has a trade off in term of power and delay [3]. For multiplier circuits, [3] shows that an array multiplier is slowest one and Vedic multiplier is fastest one. Other adder circuits available are: carry select adder (CSA), carry look ahead adder(CLA). Similarly, multiplier blocks available are: Wallace multiplier, Barun multiplier, Baugh Wooley multiplier.

The paper is organized as follows: In Section 2 we discussed about ALU design. Section 3 deals with different architecture of adders. Section 4 deals with different architecture of multipliers. Section 5 deals with subtractors. Section 6 deals with divider. Section 7 deals with analysis of performance parameter of different architecture of adders and multipliers. Section 8 conclusion of paper.

2. ARITHMETIC LOGIC UNIT

ALU is the basic fundamental unit of any central processing unit and it is a part of every microprocessor. ALU performs the arithmetic and logic operations depending on select lines. Table 1 shows the truth table for the operations performed by the Arithmetic and Logic Unit based on the status of the select signal. For SEL=10X (where X symbolizes don't care, either 0 or 1) the operation will take as 100 for addition. 16-bit optimized adder will be used for addition.

For SEL= 101, complement of second input will be added with another input using same adder design. SEL 110 will add 1 to first input when SEL 110 and SEL 111 will subtract 1 from first input only. When SEL 011 the operation will take place for multiplication using optimized multiplier.

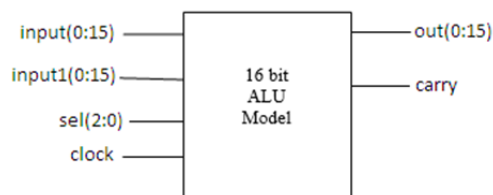


Fig. 1 ALU Model

Table. Operations of ALU

SEL1	SEL2	SEL3	OPERATIONS
0	0	0	XOR
0	0	1	XNOR
0	1	0	OR
0	1	1	MULTIPLICATION
1	0	0	ADDITION
1	0	1	SUBTRACTION
1	1	0	INCREMENT
1	1	1	DECREMENT

3. DIFFERENT ARCHITECTURE OF ADDER

3.1. Ripple Carry Adder(RCA)

Ripple Carry Adder is formed by cascading full adder in series. Carry of first full adder is fed to next full adder as third input. For n-bit ripple carry adder requires n full adder. Fig 2 shows block diagram of n bit ripple carry adder. Delay in ripple carry adder is due to rippling of carry since full adder block have to wait of carry generated by previous block. Delay increases as the number of bit increases. [3]

3.2. Carry Select Adder(CSA)

Carry select adder is composed of full adders and multiplexers. Fig 3 shows the block diagram of 4-bit carry select adder. Addition of two 4-bit number by carry select adder. This adder computes when carry is 0 and when carry is 1 and accurate sum and accurate carry is obtained by the multiplexer when correct carry is known. For the design of N bit carry select adder n full adders for carry 0 and n full adder for carry 1 and n number of multiplexer is used. The more number of multiplexer in the circuit power dissipation and area is increased. But carry select adder is fastest among others. [2]

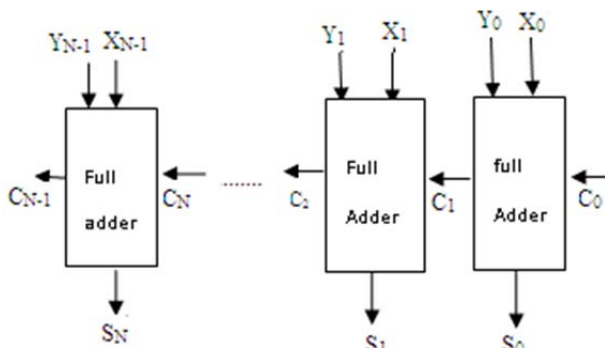


Fig. 2: Block diagram of N bit Ripple Carry Adder[3]

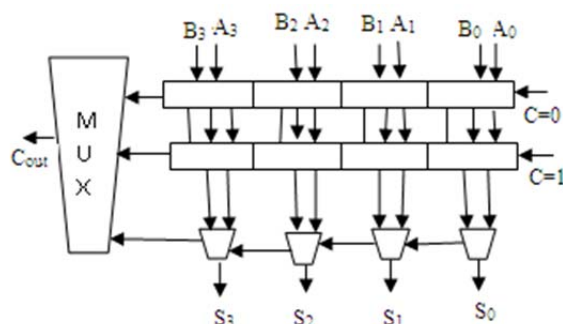


Fig. 3.4: bit CSA[2]

3.3. Square Root Carry Select Adder(SQRT-CSA)

CSA is created by cascading by CSA block with variable input bits. SQRT adder is a dual ripple carry adder and multiplexer. Fig 5 shows the 16 bit SQRT adder. In this block diagram 16

bit divides into 5 groups of different input bits. We have an adder with block sizes of 2, 2, 3,4, and 5, respectively. The disadvantage of CSA is it has maximum area due to multiple use of adder. This adder is efficient in delay and utilize the area also. [1]

3.4 Carry Look Ahead Adder(CLA)

Carry look ahead adder is based on propagating carry and generating carry. Before sum this adder calculates one or more carry. Advantage of this adder is it reduces the time to wait to calculate the result of higher bits. Fig 4. Shows the block diagram of 4-bit carry look ahead adder. Partial full adder is used instead of full adder. Boolean expressions are P_i equation for carry propagate, G_i for carry generate, S_i for sum and C_{i+1} for carry out.

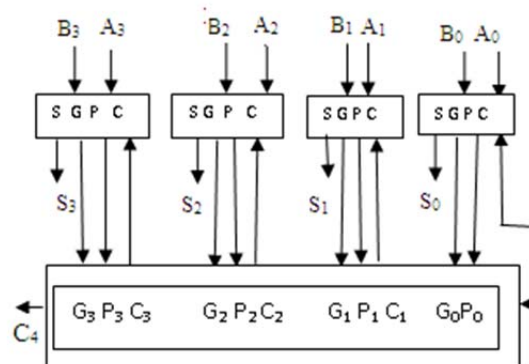


Fig. 4.4: bit CLA[1]

3.5. Common Boolean Logic Adder(CBL)

CBL works on the criteria that if propagating carry is '0' then current sum and carry out is XOR and AND of two input bits respectively, and if propagating carry is '1' then the current sum and carry out is XNOR and OR of two input bits respectively. [3]

$$S = (A \oplus B)C_{in} + (A \otimes B)C_{in} \quad \dots (1)$$

$$C = (AB)C_{in} + (A + B)C_{in} \quad \dots (2)$$

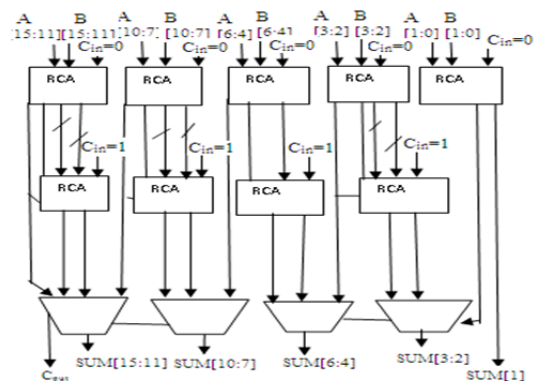


Fig. 5: SQRT[1]

Final result will decide by multiplexer depending on the propagating carry of previous cell. If the previous propagating carry is '0' then the sum will be $S_{1,0}$. & if carry is '1' then the sum will be $S_{1,1}$. If C_0 is 1 then the carry will be $C_{1,1}$ Otherwise carry will be $C_{0,1}$. as in fig 6.

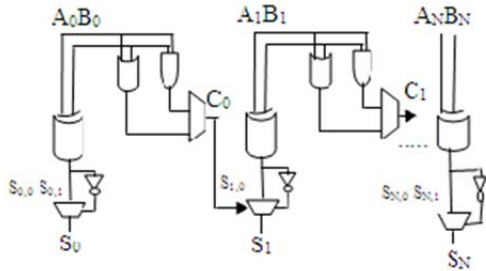


Fig. 6: CBL[3]

4. DIFFERENT ARCHITECTURE OF MULTIPLIERS

4.1. Wallace Multiplier

C.S Wallace suggested a multiplying scheme which is fast parallel scheme that reduces the sequential adding stages to reduce the partial products. Wallace multiplier is a tree like fashion which reduces the critical path and the number of adder cell needed. Fig 7 shows the Wallace multiplier. This multiplier also reduces the propagation delay. Propagation delay through the tree is equal to $O(\log_{3/2}(N))$. Disadvantage of Wallace multiplier is irregular structure. [4]

4.2. Barun Multiplier

Barun multiplier is used for unsigned bit multiplication. partial products computed in parallel and then by adders these partial products are collected. This multiplier is also known as carry save array multiplier. Using this multiplier technique speed of digital circuits increase with less propagation delay. Carry save adder and ripple carry adders have the same function. The block diagram of multiplier that is only suited for positive operands as shown in fig 8.[4]

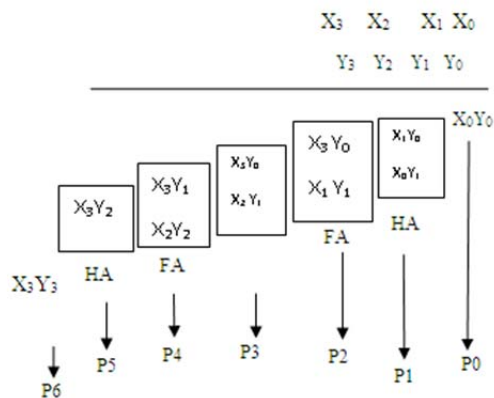


Fig. 7: Wallace Multiplier [4]

4.3. Baugh Wooley

This multiplier is used for signed or unsigned number multiplication. Technique of baugh wooley multipliers design for the direct multiplication of two's complement numbers. Each of partial products to be added is a sign number after multiplication of two's complement number. This two's complement signed multiplication is best algorithm for signed multiplication as it maximize the linearity. Fig 10 shows the Baugh wooley diagram [5]

4.4. Array Multiplier

This multiplier multiplies the two number based on shifting and adding. Though it is regular structure but it has maximum delay and high power consumption. Fig 9 shows array multiplier [8].

4.5. Vedic Multiplier

Vedic Multiplier is based on vedic multiplication. vedic mathematics reduces complex calculation into simpler one. It is the methodology to increase the speed of implementation. It is very efficient and less hardware to implement.[8]

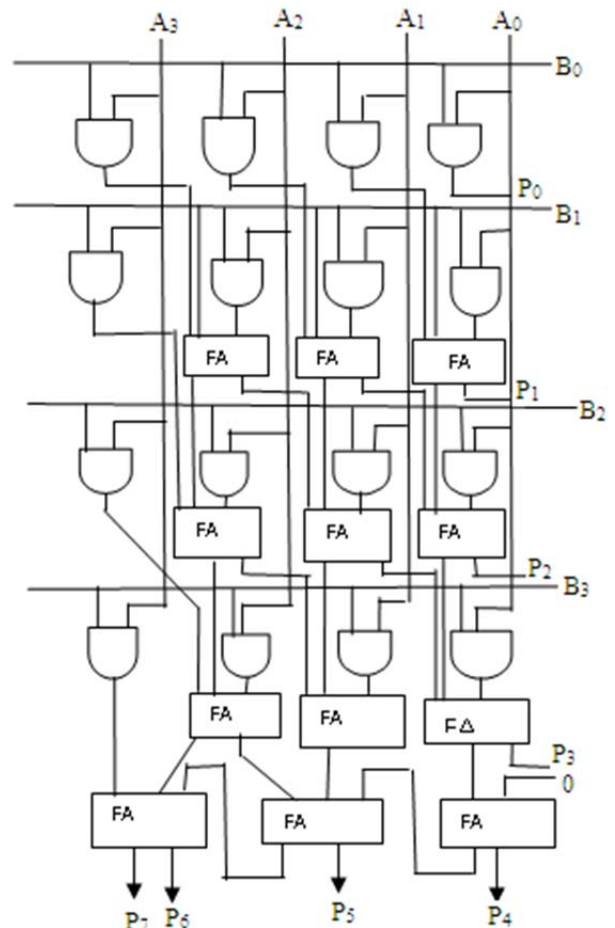


Fig. 8: Barun Multiplier[4]

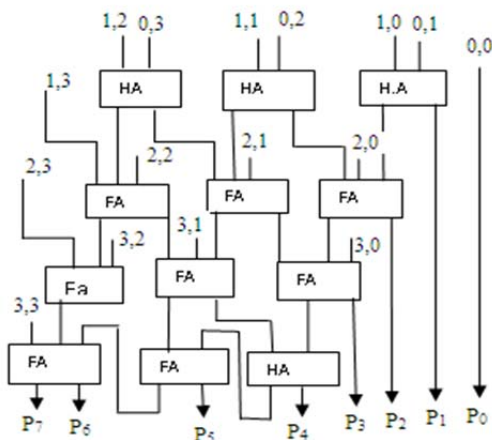


Fig. 9: Array Multiplier [6]

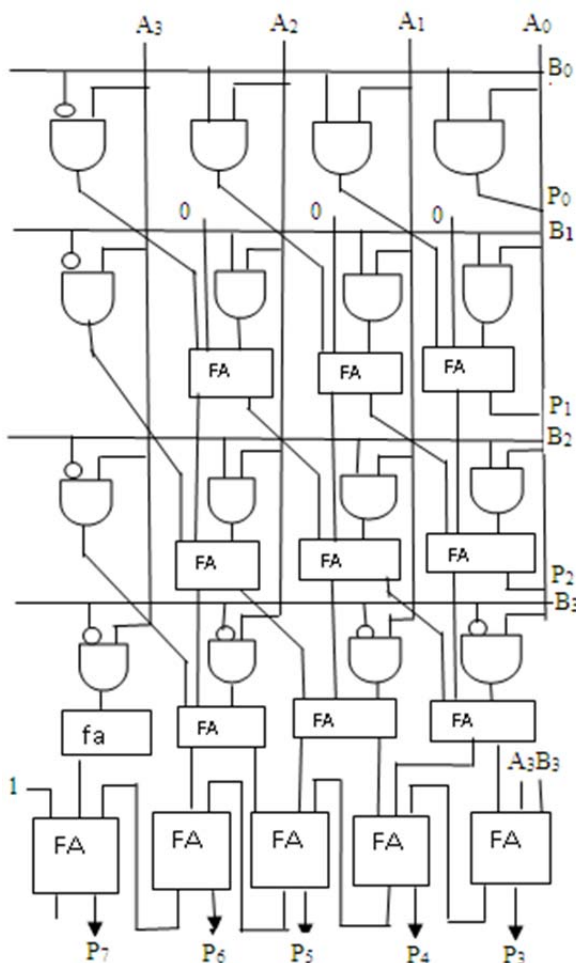


Fig. 10: Baugh Wooley Multiplier [5]

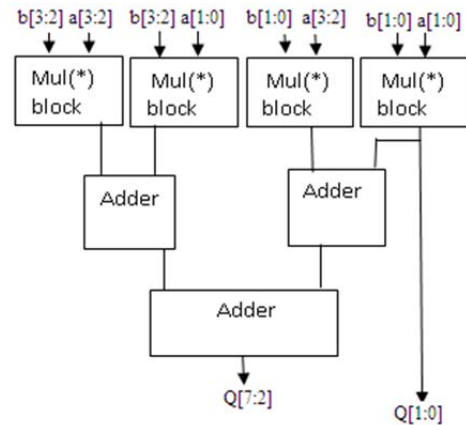


Fig. 11: Vedic multiplier [8]

5. SUBTRACTOR

Subtraction is done by taking the 2's complement of subtrahend i.e B and adding it to minuend i.e A. Using this technique subtraction operation becomes the addition operation. That requires the full adders. Full subtractor performs subtraction of two bits, one is minuend and other is subtrahend. In full subtractor '1' is borrowed by the previous bit. Hence there are three bits are considered at the input of a full subtractor. There are two outputs, that are output D and output Bo. The Bo output indicates that the minuend bit requires borrow '1' from the next minuend bit. If we compare output D and output Bo with full adder, it can be seen that the output D is the same as that for the SUM output. The output Bo is similar to CARRY-OUT. In the case of a half-subtractor

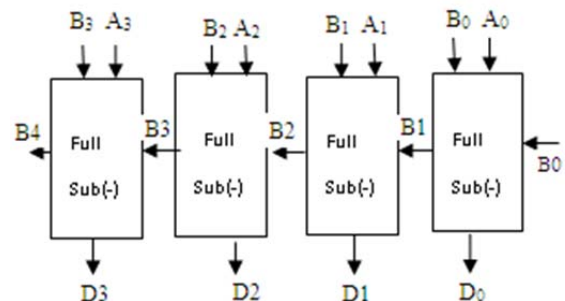


Fig. 12.4: bit subtractor

6. DIVIDER

ALU has basic operation i.e division. Division is also a very important part of arithmetic. This process is quite similar to the decimal division. Let us take A = 11010 and B = 101 We want to divide A by B The structure of operation of binary division is similar to that of decimal division. A division algorithm is which, given two integers N and D ,i.e their quotient and/or remainder, the result of division.

Division algorithms are of two main categories: slow division and fast division. Slow division algorithms produce one digit of the final quotient per iteration. Fast division methods start with a close to the final quotient and produce twice as many digits of the final quotient on each iteration. $N/D=(Q,R)$, where

N = dividend D =divisor

Q = Quotient R = Remainder

7. ANALYSIS OF DIFFERENT ARCHITECTURE OF ADDER AND MULTIPLIERS

In this section we have seen the comparison of delay, power and area using 90 nm technology on Synopsis Design Compiler.

Table 2: Comparatively study of different topologies of adders

ADDERS		4 BIT	8 BIT	16 BIT
Ripple Carry Adder(RCA)	Delay(ns)	5.85	6.01	11.60
	Power(uw)	17.0720	39.251	286.87
	Area(um ²)	240.07	559.26	947.06
Carry Select Adder(CSA)	Delay(ns)	2.60	4.0	7.5
	Power(uw)	39.307	100.57	400.015
	Area(um ²)	467.37	669.54	1141.01
SQRT	Delay(ns)	2.25	3.5	7.00
	Power(uw)	41.65	163.50	445.55
	Area(um ²)	478.88	771.05	1839.00
Carry Look ahead Adder(CLA)	Delay(ns)	4.46	5.85	10.79
	Power(uw)	14.3170	38.832	294.46
	Area(um ²)	184.336	450.26	819.504
Common Boolean Adder(CBL)	Delay(ns)	4.25	5.55	8.56
	Power(uw)	21.7288	77.69	388.53
	Area(um ²)	273.304	545.25	959.877

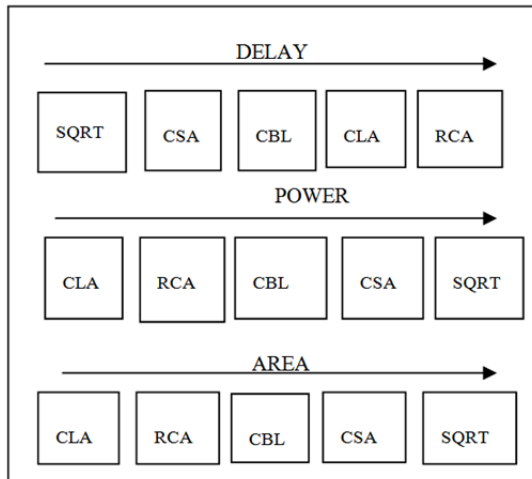
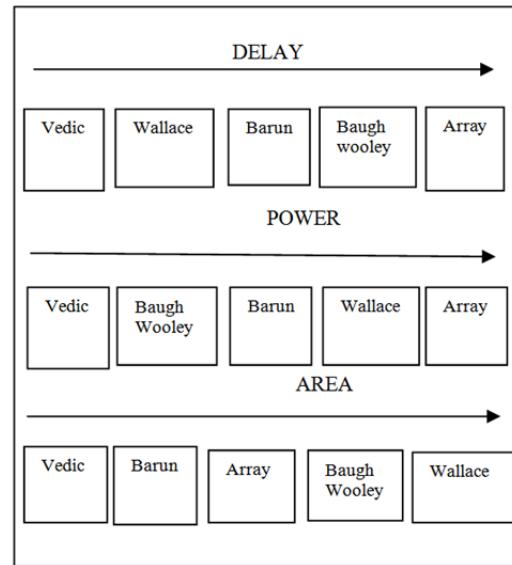


Table 3: Comparatively study of different topologies of multipliers

Multipliers		4BIT	8BIT	16BIT
Vedic	Delay(ns)	0.53	1.17	22.94
	Power(uw)	15.635	20.5635	327.0302
	Area(um ²)	456.5	756.98	510005.198
Wallace	Delay(ns)	6.46	16.52	49.93
	Power(uw)	70.358	403.4812	2.6049e+03
	Area(um ²)	738.1003	3366.98	39361.65
Barun	Delay(ns)	7.36	16.62	55.10
	Power(uw)	57.3068	399.5117	2.3500e+03
	Area(um ²)	600.1003	2991.35	13424.03
Array	Delay(ns)	13.5	54.49	192.17
	Power(uw)	150.478	482.4679	2.6738e+03
	Area(um ²)	920.36	3221.1821	12623.97
Baugh Wooley	Dealy(ns)	8.04	17.25	60.12
	Power(uw)	48.8707	362.677	1.9900e+03
	Arae(um ²)	520.36	2993.92	26630.23



8. CONCLUSION

We have seen that SQRT adder is fastest adder but not efficient in power. Ripple carry adder is power efficient but slowest. Common boolean logic adder is efficient both in terms of delay and power. Carry look ahead adder is fastest among all. So for power & delay efficient ALU we must use common Boolean adder. For multipliers, vedic multiplier is most efficient in terms of area and power. Thus, our ALU is designed using vedic multiplier and common Boolean adder and therefore has high performance with low power as well as area efficient.

REFERENCES

- [1] Y. He, C.-H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 4, pp. 4082–4085, May 2005.
- [2] Prasad, Y. Bhavani, et al. "Design of low power and high speed modified carry select adder for 16 bit Vedic Multiplier." *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*. IEEE, 2014.
- [3] I.-C.Wey, C.-C.Ho, Y.-S. Lin, and C.-C. Peng, "An area-efficient carry select adder design by sharing the common boolean logic term," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '12)*, pp. 1091–1094, Hong Kong, March 2012.
- [4] Singh, Khuraijam Nelson, and H. Tarunkumar. "A review on various multipliers designs in VLSI." *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015.
- [5] Sjalander, Magnus, and Per Larsson-Edefors. "High-speed and low-power multipliers using the Baugh-Wooley algorithm and HPM reduction tree." *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*. IEEE, 2008.
- [6] Gujmagadi, Pavan, et al. "Design of Vedic multiplier for high fault coverage and comparative analysis with conventional multipliers." *Advance Computing Conference (IACC), 2015 IEEE International*. IEEE, 2015.
- [7] Ramalatha, M., et al. "High speed energy efficient ALU design using Vedic multiplication techniques." *Advances in Computational Tools for Engineering Applications, 2009. ACTEA'09. International Conference on*. IEEE, 2009.
- [8] Akshata R., Prof. V.P. Gejji, Prof. B.R. Pandurangi"Analysis of vedic multiplier" International conference of on computing, communication and energy system (ICCCES-16).