



## PROMINENT SPEED ARITHMETIC UNIT ARCHITECTURE FOR PROFICIENT ALU

R. Rashvnee, D. Roshini Keerthana, T. Ravi and P. Umarani

Department of Electronics and Communication Engineering, Sathyabama University, Chennai, India

E-Mail: [rashvnee.ravichandran@gmail.com](mailto:rashvnee.ravichandran@gmail.com)

### ABSTRACT

ALU is one of the most important unit of processor. The computing efficiency of the processor depends on the competency of the ALU. ALU unit performs the arithmetic and logical operations. The adder and multiplier are the main computational units of the arithmetic unit. The performance factors such as delay, power and area. Parallel prefix adders have better delay performance; it involves the execution of the operation in parallel. Brent Kung adder is the most area and power efficient parallel prefix adder. In this paper we proposed high speed Brent Kung adder which consists of Urdhava Tiryakbhyam sutra based Vedic multiplier. In the conventional multiplier speed is restricted by the adders used for partial products. The proposed multiplier is used in the arithmetic unit of an ALU shows better performance in terms of delay. The proposed arithmetic architecture is designed, evaluated and implemented in Xilinx FPGA.

**Keywords:** PPA, vedic multiplier, delay, brent kung adder, RCA, brent kung adder.

### 1. INTRODUCTION

VLSI is the process of integrating thousands of transistors into a single chip. VLSI design is mainly used to minimize the interconnecting fabrics area. The main factors which limit the development of smaller and more complex IC chip are the IC fabrication technology, designer productivity and the cost. Depending on the application, different performance aspects become important. An ALU is an integral part of a processor. It performs all the arithmetic and logical operations. Fast and accurate operation depends on the performance of multiplier [2]. Hence improving the performance of the multiplier improves the efficiency of the ALU. In order to increase the performance of a multiplier we need to make the adder more efficient [3].

Binary addition is the most fundamental and important arithmetic function where the adder should be fast and efficient in terms of area, power and speed. The major problem for binary addition is the propagation delay in the carry chain. As the number of input operands increase the carry propagation delay increases. It is seen that than conventional adders such as ripple carry adder, carry select adder, etc., the parallel prefix adders are more efficient as their structure consists of pre-processing, carry look ahead and post processing sections[4]. The Booth, Wallace, Array multipliers have all been implemented in an ALU, but the multiplier implemented by using the Vedic sutras have been showing high performance when implied in an ALU.

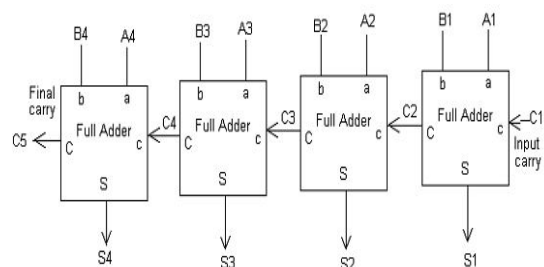
### 2. EXISTING ADDERS AND MULTIPLIERS

There is a wide range of adders and multipliers are used in the ALU design. The adder is the most used part of the ALU.

#### A. Ripple carry adder

Among all the arithmetic operations, multiplication can perform through repeated addition, subtraction (by negating one operand) or division (repeated subtraction). Half Adders can be used to add two

one bit binary numbers. It is also possible to create a logical circuit using multiple full adders to add N-bit binary numbers [10]. Each full adder inputs a  $C_{in}$ , which is the  $C_{out}$  of the previous adder. This kind of adder is a Ripple Carry Adder, since each carry bit "ripples" to the next full adder. The first (and only the first) full adder may be replaced by a half adder [8]. The block diagram of 4-bit Ripple Carry Adder is shown in Figure-1.



**Figure-1.** Ripple carry adder.

#### B. Vedic multiplication

Vedic mathematics is used to solve the complex calculations involve in usual mathematics. Owing to its simple strategy, which mainly includes natural viewpoints of human being, it leads to a straightforward process. It consents to incorporate the arithmetic rules along with high speed and easy implementation, thereby viable for a range of applications based on computing [4, 9].

The Vedic multiplier is based on the algorithm named as "Urdhva Tiryakbhyam" sutra shown in Figure-2. Traditionally, this Sutra has been employed to multiply two given numbers in a decimal number system. However, we put forward the multiplication of two binary numbers using analogous technique with a motive to make this efficient technique compatible for digital hardware systems. It is a universal multiplication formula that can be used in for all multiplications. It literally means "Vertically and Crosswise". The algorithm is viable for multiplication of any two numbers exhibiting bit length



equal to  $n$ . Besides this, it introduces a parallel execution of partial products and sums. The clock frequency is not an obstacle in accounting the calculations. This technique has proven advantageous in combating with large delays and complexity of conventional multipliers [1].

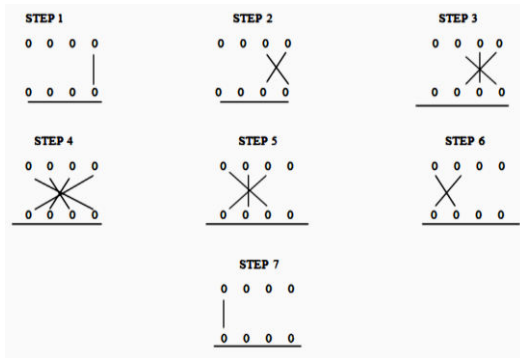


Figure-2. Vedic multiplication algorithm.

### C. Multiplier architectures

#### 2x2 vedic multiplier

A 2X2 Vedic multiplier module is implemented using four AND gates and two half adders which is shown in Figure 3 and Figure-4 shows hardware realization of 2X2 Vedic multiplier.

#### 4x4 vedic multiplier

The 4X4 Vedic multiplier is implemented using four 2X2 bit Vedic multiplier. The Figure-2 is a sample presentation for Vedic multiplication.

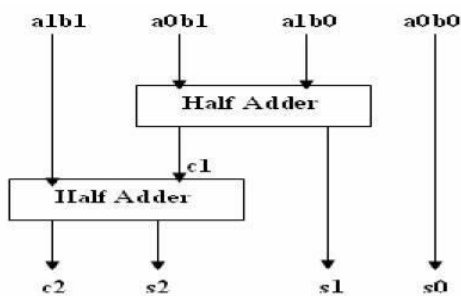


Figure-3. Block diagram of 2x2 vedic multiplier.

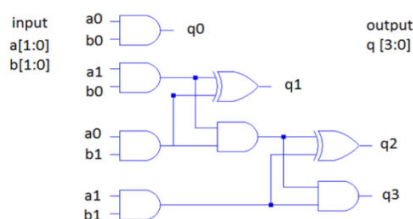


Figure-4. Hardware realization of 2x2 vedic multiplier.

The 2X2 multiplier inputs are A1A0 and B1B0. The last block is 2X2 bit multiplier with inputs A3 A2 and B3 B2. The middle one shows two 2X2 bit multiplier with

inputs A3 A2 and B1B0 and A1A0 and B3 B2. The Block diagram of 4X4 bit Vedic multiplier is shown in Figure-5. To get final product four 2X2 bit Vedic multiplier and three 4-bit Adders are required.

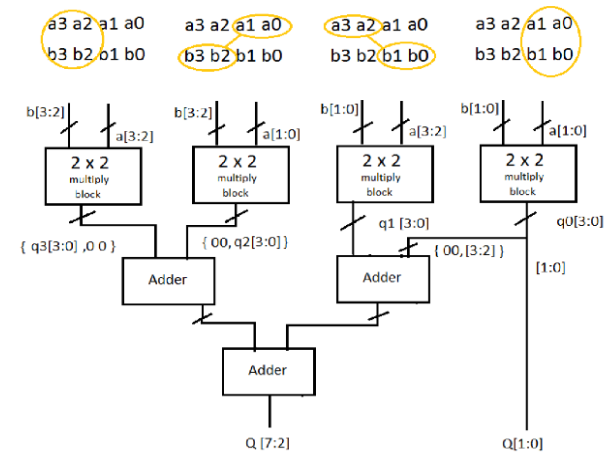


Figure-5. Block diagram of 4X4 vedic multiplier.

#### 8x8 vedic multiplier

8X8 bit Vedic multiplier module as shown in Figure-6. It can be easily implemented by using four 4X4 bit Vedic multiplier modules as discussed in the previous section.

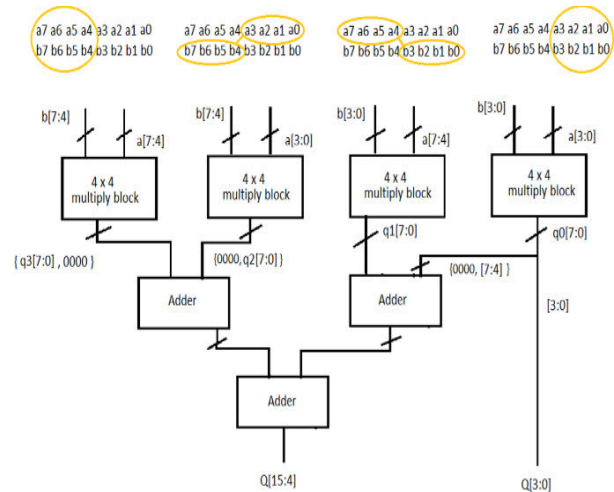


Figure-6. Block diagram of 8x8 vedic multiplier.

### 3. PROPOSED PARALLEL PREFIX ADDER

The adders play a very crucial role in an ALU in the computation of partial products in the multiplier. This calculation contributes to major delay of a processor. In order to reduce the delay modern adder architectures are used which compute in parallel. It employs a 3-stage structure of the CLA adder. Figure-7 shows the process of parallel prefix adder.

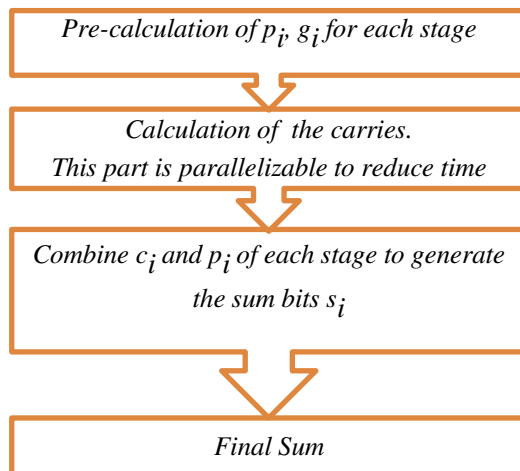


Figure-7. Process of parallel prefix adder.

▪ Computation of carry generation, propagation signals:

Previous carry is calculated to the next bit is called propagate signal and generate is to generate the carry bit below are the signals:

$$G_i = A_i \cdot B_i \quad (1)$$

$$P_i = A_i \oplus B_i \quad (2)$$

▪ Calculation of all carry signals:

$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j} \quad (3)$$

$$P_{i:j} = P_{i:k} \cdot P_{k-1:j} \quad (4)$$

▪ Calculation of Final Sum:

$$S_i = P_i \oplus G_{i-1:0} \quad (5)$$

#### A. Brent kung adder

Brent-Kung has maximum logic depth, minimum area and avoid explosion of wires. The Brent-Kung adder does odd computation first and then even. It computes prefixes for 2-bit groups. These are used to find prefixes for 4-bit groups, which in turn are used to find prefixes for 8-bit groups, and so forth. The 8 bit adder is shown in Figure-8. The prefixes then fan back down to compute the carries-in to each bit. The tree requires  $2\log_2 n - 1$  stages.

The fan-out is limited to 2 at each stage. It takes lesser area to be implemented when compared to other prefix adders such as Kogge Stone adder [8]. This will reduce the delay without compromising the power performance of the adder. But the method used in [2, 3] has not reduced delay to a great extent. The area consumption and power are the lowest in BKA than other adders [2, 5].

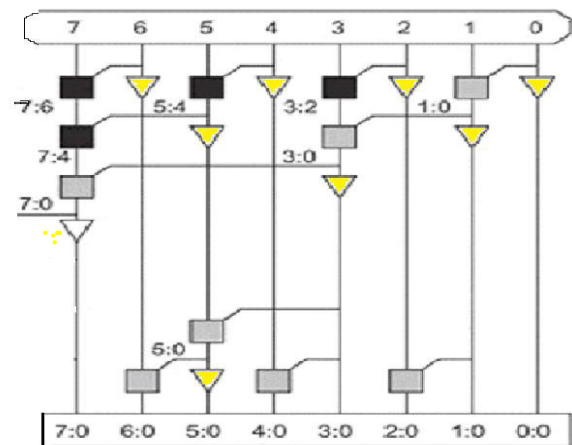


Figure-8. 8-Bit brent kung adder.

The black cells and grey cells are the main computational block in a parallel prefix adder and are shown in Figure-9. The black cells computes both generate and propagate while the grey cells compute propagate only.

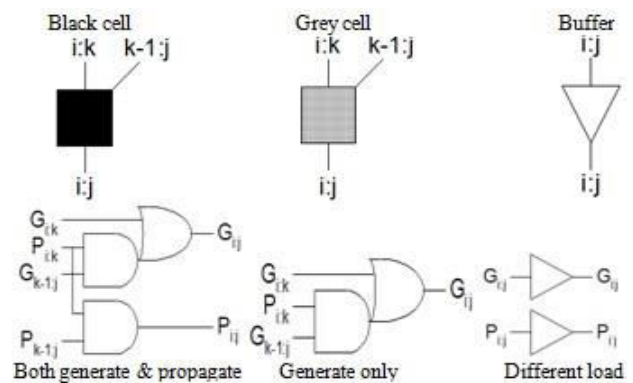


Figure-9. Black, grey and Buffer cells.

#### 4. PROPOSED BRENT KUNG ADDER ARCHITECTURE

The proposed architecture of 8 bit Brent Kung adder the buffers which have been used for the driving compatibility of the gates have been removed to reduce the delay of the adder and is shown in Figure-10. By removal of it, it has not shown any errors or glitches and thus when implemented on the multiplier reduces the delay for the calculation of the carry chain and partial products. As Brent Kung has the simplest structure among all parallel prefix adders our main concern was to reduce the delay.

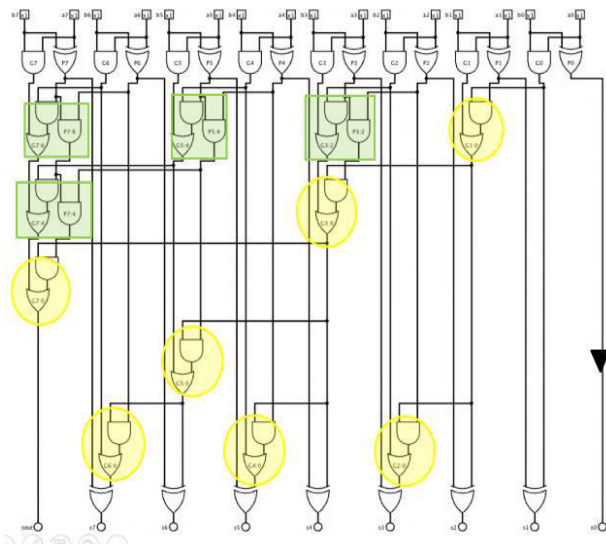


Figure-10. Proposed 8-bit brent kung adder.

#### Proposed multiplier architecture

The Figure-11 shows the modified Brent Kung adder based 8X8 Vedic multiplier. In the proposed architecture delay is reduced and increases the overall performance as compared to the existing Brent Kung adder.

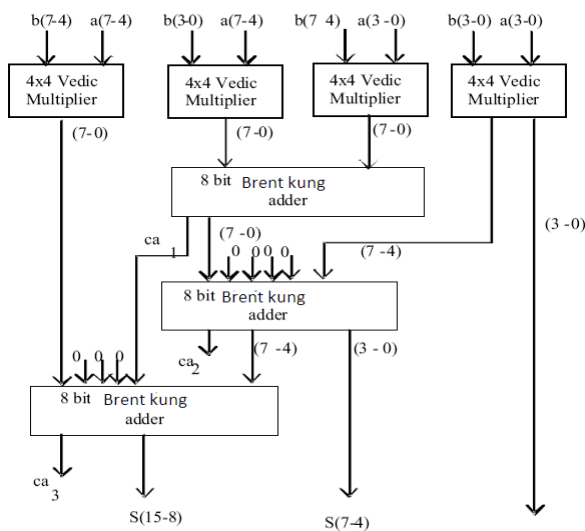


Figure-11. 8X8 vedic multiplier using brent kung adder.

## 5. RESULTS AND DISCUSSIONS

The proposed 8-bit architectures are developed in Xilinx platform. These are designed in verilog HDL, simulated in Xilinx ISim simulator and synthesized using Xilinx XST. These architecture are implemented and analyzed in Xilinx Spartan3 FPGA. The simulation outputs and RTL schematic of modified Adder, Multiplier and ALU are shown in Figure 12, 13, 14 and 15. Table-1 shows synthesize report.

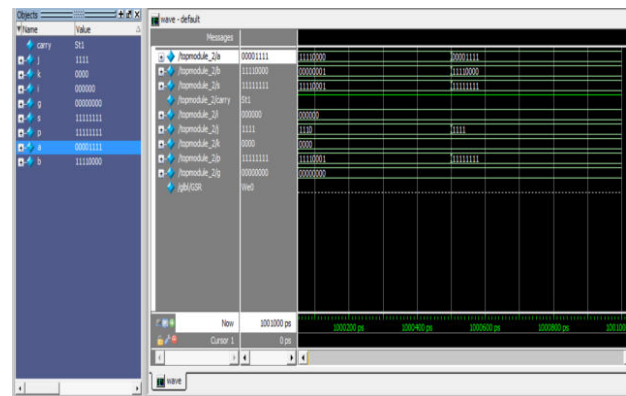


Figure-12. Simulation output of modified brent kung adder.

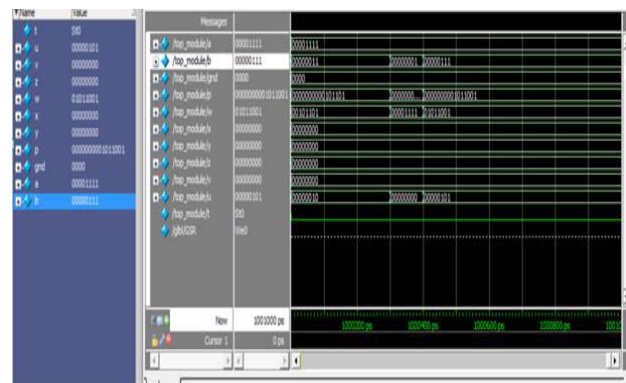


Figure-13. Simulation output of Vedic Multiplier using modified Brent Kung Adder.

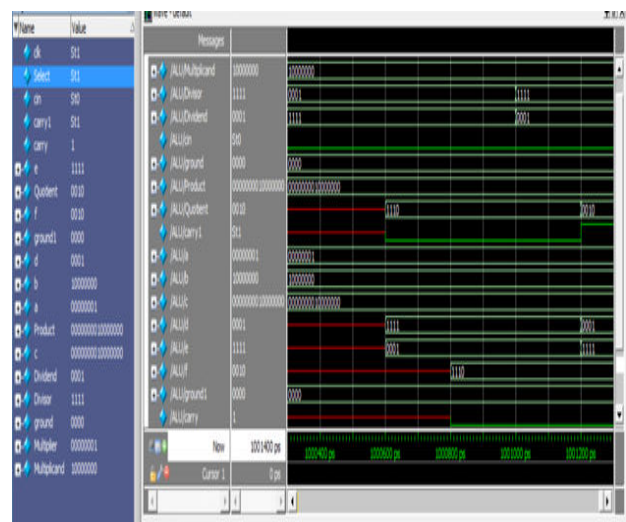
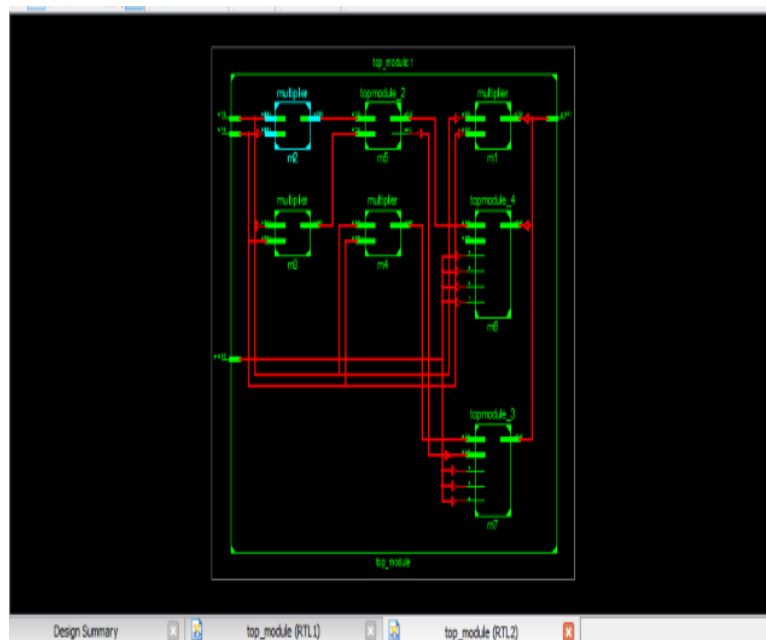


Figure-14. Simulation output of ALU using modified brent kung adder.



**Figure-15.** RTL Schematic of modified 8-bit vedic multiplier.

**Table-1.** Synthesize report.

| S. No. | Parameter                  | Delay (ns) | No. of slices | No. of LUTs |
|--------|----------------------------|------------|---------------|-------------|
| 1      | Ripple Carry Adder         | 18.771     | 9             | 15          |
| 2      | Brent Kung Adder           | 15.178     | 10            | 18          |
| 3      | Vedic Multiplier using RCA | 35.223     | 114           | 202         |
| 4      | Vedic Multiplier using BKA | 32.66      | 120           | 212         |
| 5      | ALU using RCA              | 14.334     | 119           | 211         |
| 6      | ALU using BKA              | 12.383     | 125           | 221         |

## 6. CONCLUSIONS

The proposed Vedic multiplier is using unique addition tree structure, which gives better response in terms of speed in comparison to the conventional Vedic multiplier hardware. Vedic multiplier is using a modified Brent Kung adder which has less delay. The delay of the proposed ALU is reduced by 2ns. The results are concluded that the proposed adder architecture based multiplier and ALU are achieved better computation speed. This proposed ALU is suitable for high speed processor design.

## REFERENCES

- [1] Akshata R. Kanchan, Joshi. A., Prof. V.P. Gejji. 2015. Analysis and Implementation of Vedic Multiplier FPGA. International Journal for Technological Research in Engineering. 2(9): 2181-2183.
- [2] Deepak raj, Sahana, Prof. Praveen J, Prof. Raghavendra Rao R 2015. Design and Implementation of Different Types of Efficient Parallel Prefix Adders. 2015. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering. 3(1).
- [3] Ravi. T. 2015. Design and performance analysis of ultra low power RISC processor using hybrid drowsy logic in CMOS technologies. International Journal of Applied Engineering Research (IJAER). 10(2): 4287-4296.





- [4] M. Manoranjani and T. Ravi. 2015. Multithreshold CMOS Sleep Stack and Logic Stack Technique for Digital Circuit Design. ARPN Journal of Engineering and Applied Sciences. 10(10): 4550-4556.
- [5] M. Prasanna Kumar, V. Siddharthan, Dr. Gopalakrishnan. K. 2015. Comparative Analysis of Brent Kung and Kogge Stone Parallel Prefix Adder for their Area, Delay and Power Consumption. Indian Journal of Applied Research. 5(10).
- [6] MeghaTalsania, Eugene John. 2009. Comparative Analysis of Parallel Prefix Adders. A Comparative Analysis of Parallel Prefix Adders 12<sup>th</sup> Euromicro Conference on Digital System Design, Architectures, Methods and Tools. pp. 281-286.
- [7] Prof.Navneet Dubey and ShyamAkashe. 2014. Implementation of an Arithmetic Logic Using Area Efficient Carry Look Ahead Adder. International Journal of VLSI Design and Communication Systems, VLSICS. 5(6): 29-34.
- [8] R. Ramya and T. Ravi. 2015. Design of Cache Memory Mapping Techniques for Low Power Processor. ARPN Journal of Engineering and Applied Sciences. 10(11): 4783-4788.
- [9] Sudheer Kumar Yezerla, B Rajendra. 2014. Design and Estimation of delay, Power and area for Parallel Prefix Adders. Proceedings of 2014 RAECS UIET Panjab University Chandigarh, IEEE.
- [10] S. Ranjith, T. Ravi, P. Umarani, R. Arunya. 2014. Design of CNTFET based sequential circuits using fault tolerant reversible logic. International Journal of Applied Engineering Research. 9(24): 25789-25804.
- [11] Sunil M, Ankith R D, Manjunatha G D, Premanandha B S. Design and Implementation of Faster Parallel Prefix Kogge Stone Adder. International Journal of Electrical and Electronic Engineering and Telecommunications.