

IMPLEMENTATION OF AREA OPTIMIZED REVERSIBLE LOGIC BASED ADAPTIVE RECURSIVE KARATSUBA

M. Sivaprakash¹, V. Sarvanan^{2,3}, M. Thangavel³

¹Research Scholar, Knowledge Institute of Technology, Kakapalayam, Salem, Tamil Nadu – 637504

^{2,3}Associate Professor, Knowledge Institute of Technology, Kakapalayam, Salem, Tamil Nadu – 637504

ABSTRACT

Nowadays, efficient binary multiplication is required by computationally intensive applications like DSP, image processing, floating point processors, and communication technologies. This block often consumes the most power and time. In order to cut down on time, this study suggests an effective architecture for unsigned binary multiplication. A 1616-bit multiplier based on the Vedic Karatsuba algorithm has been created. It is optimized utilizing a square-root carry-select-adder, reversible logic, and an adaptive and recursive technique. Verilog was used to code the designs, while Xilinx was used to synthesis them.

Index Terms—Square Root Carry Select Adder, Karatsuba Multiplier, reversible logic, Recursive adaptive Karatsuba algorithm.

I. INTRODUCTION

The fundamental component of many ASIP and general-purpose processors is the multiplier. Therefore, the processing rate, size, and power dissipation of the multipliers determine how well these systems work. Although traditional algorithms like Wallace Tree and Dadda [1]–[4] have stood the test of time, the demand for quicker multiplication-algorithms/designs has developed due to technological innovation and frequent changes in VLSI architectures. For decimal numbers, the Vedic Multiplier Algorithm (VMA) [5] employs a unique technique to cut down on the number of partial product terms.

A multiplier based on the Vedic Karatsuba algorithm is proposed in this research. This multiplier is updated with the "adaptive" notion, reversible logic, and high-speed adders, and it is used for the binary system. In order to test and assess the performance, Wallace and Dadda multipliers of the 1616-bit kind are used. Three alternative adder designs are used in the studies, with the Square-Root-Carry-Select-Adder (SRCSA) producing the best delay findings. The best design is suggested after all ideas have been tested on an ASIC platform.

II. LITERATURE SURVEY

Mr. Vijay Kumar Reddy Implementation of a Modified High Speed Vedic Multiplier The research project's modified binary Vedic multiplier is described utilizing Vedic sutras from classical Vedic mathematics. It offers adjustment to the Vedic multiplier that was initially established. The updated binary Vedic multiplier is recommended since it has improved in terms of both device consumption and time delay.

Verilog HDL was used to design and implement the suggested method. Modelsim tool is used for HDL simulation, while Xilinx is utilized for circuit synthesis. The simulation was performed for the multiplication process using 4 bits, 8 bits, and 16 bits. Only the simulation results for the 16 bit binary Vedic multiplier approach are displayed. For higher sizes, this modified multiplication method is expanded. This multiplication method's results are contrasted with those of other traditional Vedic multipliers.

III. RECURSIVE KARATSUBA MULTIPLICATION

Recursive Karatsuba, which increases performance when bit size is large, is based on adding the Karatsuba algorithm recursively at each level. Before iteratively applying the Karatsuba algorithm to the segmented bits, the approach separates the number of bits (N) into groups of half that number of bits (N/2). For a 16-bit multiplication, for instance, the result is separated into 8-bit multiplication, 4-bit, and then 2-bit, which is the final stage before doing a standard multiplication. For the third product term, we have used adaptive Karatsuba at every step.

Standard Karatsuba Multiplier

Assume that X and Y are 'n'-bit inputs. Assuming that X and Y are divided into two equal portions, the higher order bits are represented by X_H and Y_H and the lower order bits by X_L and Y_L. Their output is calculated as:

$$XY = \left(2^{\frac{n}{2}}X_H + X_L\right)\left(2^{\frac{n}{2}}Y_H + Y_L\right)$$

$$XY = 2^n(X_H Y_H) + 2^{\frac{n}{2}}(X_H Y_L + X_L Y_H) + (X_L Y_L) \quad (1)$$

The calculation in the Karatsuba algorithm is recast as:

$$(X_H Y_L + X_L Y_H) = (X_H + X_L)(Y_H + Y_L) + (X_H Y_H) + (X_L Y_L) \quad (2)$$

Therefore, $(X_H + Y_H)(X_L + Y_L)$, $X_H Y_H$, and $X_L Y_L$ are all $3n/2$ -bit multiplications of $4n/2$ -bit multiplications. The typical Karatsuba multiplier is seen in Fig. 1 at a point where the inputs are n bits.

The usual multiplication algorithm's time complexity calls for:

$$O(n) = n^2 \quad (3)$$

Whereas Karatsuba multiplication algorithm requires:

$$O(n) = n^{1.58} \quad (4)$$

Where O is the order of complexity, with $O(1)$ equal to 1, and n is the number of bits. This demonstrates analytically that the Karatsuba method, which has a complexity of $n^{1.58}$, is quicker than ordinary multiplication owing to n 's logarithmic power.

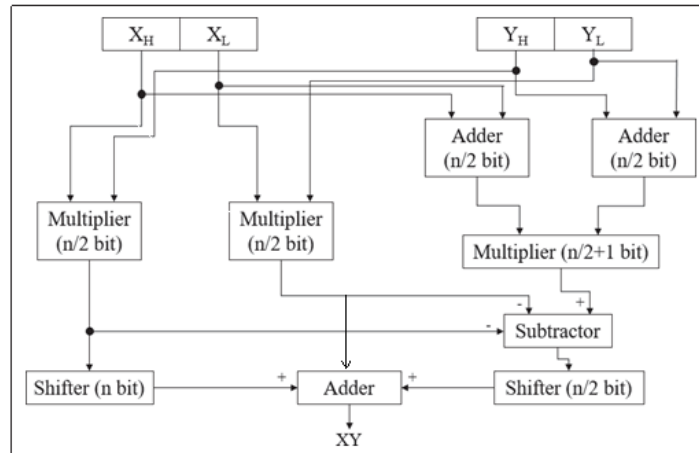


Fig. 1. N-Bit Standard Karatsuba Multiplier

We change the Karatsuba method to make it more effective in computing the third product. If X and Y each have " n " bits of input, the result is an argument of size $(n/2 + 1)$ bits. In order to represent the arguments with Z and U , we will remove $(n/2 - 1)$ bits from each. Given that the carry-outs of the third product arguments $(X_H + Y_H)$ and $(X_L + Y_L)$ identify the higher order bits Z_H and U_H , respectively, the lower order bits Z_L and U_L may be viewed similarly. Therefore, the aforementioned assumption may be assessed in light of Z_H and U_H as stated in Table I.

THIRD PRODUCT COMPUTATION		
Z_H	U_H	Product ₃
0	0	$Z_L U_L$
0	1	$2^{\frac{n}{2}} Z_L + Z_L U_L$
1	0	$2^{\frac{n}{2}} U_L + Z_L U_L$
1	1	$2^n + 2^{\frac{n}{2}}(U_L + Z_L) + Z_L U_L$

Table 1: Product computation

Table I suggests that in addition to the $(n/2 + 1)$ bit multiplier at each step, one multiplication of $(n/2 - 1)$ bits and additional shifting, adding, and multiplexing operations are required to compute the third product using $(n/2 + 1)$ bits. This allows the Karatsuba implementation to be recursive without the need for specialized hardware. Figure 2 depicts the adaptive concept for $(n/2 + 1)$ bit computing at a stage in which the inputs are n bits and the "Shift and Add" block is applicable under the conditions specified in Table I.

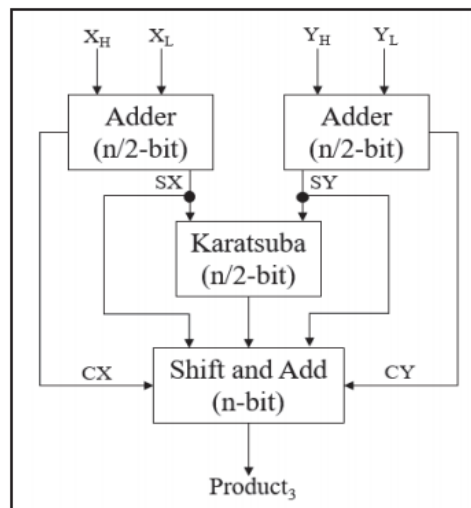


Fig. 2. Adaptive Concept for 3rd Product Computation

The generated partial products are combined in the standard Wallace and Dadda multiplier through 3:2 compressors. The proposed multiplier is based on the adaptive, recursive Karatsuba technique, and the results show that it has lower latency than the traditional Wallace and Dadda multiplier.

Square Root Carry Select Adder:

Square Root Carry Select Adder (SRCSA) [9]-[10] block size may vary. Instead of utilizing a uniform block size of four (as previously done), a 16-bit adder may be developed

using block sizes of 2-2-3-4-5 [8]. The full analysis is left out for the purpose of concision. When both the Full-Adder delay and the MUX delay are zero, this split is optimal. The proposed SRCSA adder for 16 bits has the symbols A and B as its inputs, C in as its carry-in, and S and C as its outputs in the block diagram (Cout).

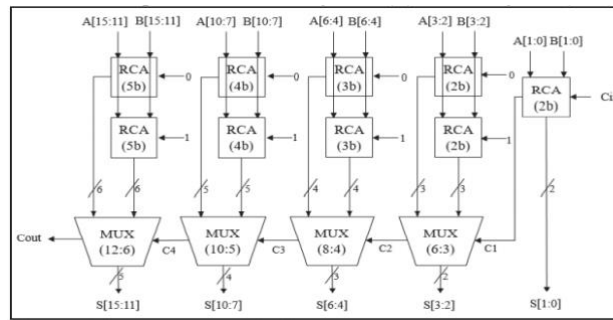


Fig. 3. 16-bit SRCSA (proposed adder)

REVERSIBLE GATES:

Reversible logic operations generate extremely little heat and do not erase (lose) information. As a result, reversible logic will almost certainly be needed in high-speed, low-power systems. Low-power CMOS design; optical computing, quantum computing, and nanotechnology all make use of reversible circuits. Most of the time, reversible logic is used in quantum computers.

Networks of quantum logic gates make up quantum computers. Each gate uses one, two, or more qubits to work (two-state quantum systems). The traditional bit values of 0 and 1 are represented by each qubit, which stands for a basic unit of information. Due to the reversibility of unitary operations, You can't just figure out how quantum networks that do addition, multiplication, and exponentiation work by looking at their classical Boolean equivalents (classical logic gates such as AND or OR are clearly irreversible). Therefore, reversible logic elements must be used to construct quantum arithmetic.

PERES GATE:

A 3*3 Peres gate is seen in Fig. 5. The output vector is O, while the input vector is I (A, B, C) (P, Q, R). $P = A$, $Q = A \text{ xor } B$, and $R = AB \text{ xor } C$ define the output. A Peres gate has a quantum cost of 4. The Peres gate is utilized in the suggested design because it has the lowest quantum cost. When C is 0, Peres gate functioned as a half adder.

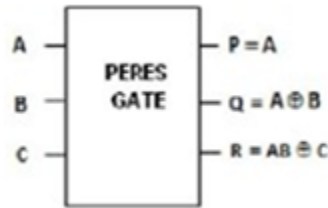


Fig 4: Peres gate

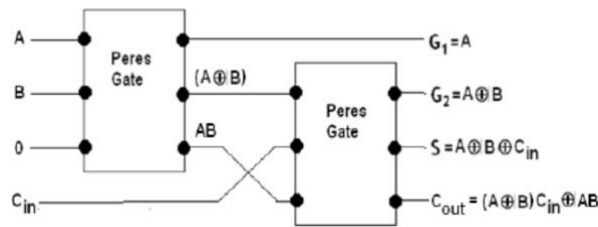


Fig 5: Full adder using reversible logic gates

IV. RESULTS

RTL SCHEMATIC: The register transfer level, often known as RTL, is an acronym for the architecture's blueprint. It is used to compare the designed architecture to the ideal architecture that still has to be created. By utilizing the coding language known as verilog or vhdl, the HDL language is utilized to translate the description or summary of the architecture into the functioning summary. Even the internal connection blocks are specified in the RTL schematic for easier analysis. The planned architecture's RTL schematic diagram is shown in the picture below.

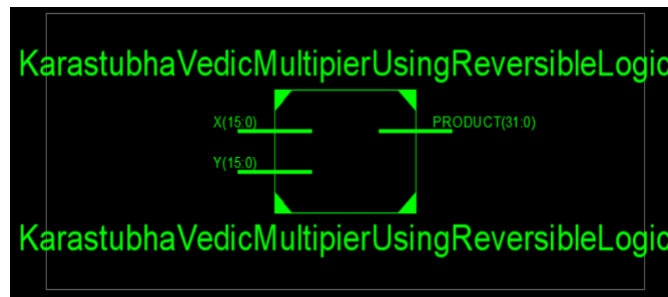


Fig6: RTL Schematic of Proposed Vedic multiplier

TECHNOLOGY SCHEMATIC: The technology schematic generates a LUT format representation of the architecture, where the LUT is regarded a square unit and is utilized in VLSI to estimate the size of the architecture design. The LUT is also used in FPGA to represent the memory allocation of the code.

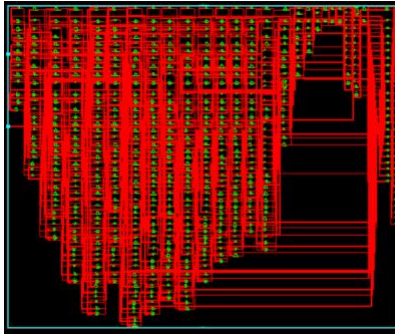


Fig7: View Technology Schematic of proposed Vedic multiplier

SIMULATION: In terms of how it functions, the simulation is the process, whilst the schematic is used to verify the connections and building pieces. On the tool's home screen, the simulation window is activated as the tool switches from implantation to simulation. The simulation window displays the output in the form of wave shapes. Here, it is flexible enough to offer several radix number systems.

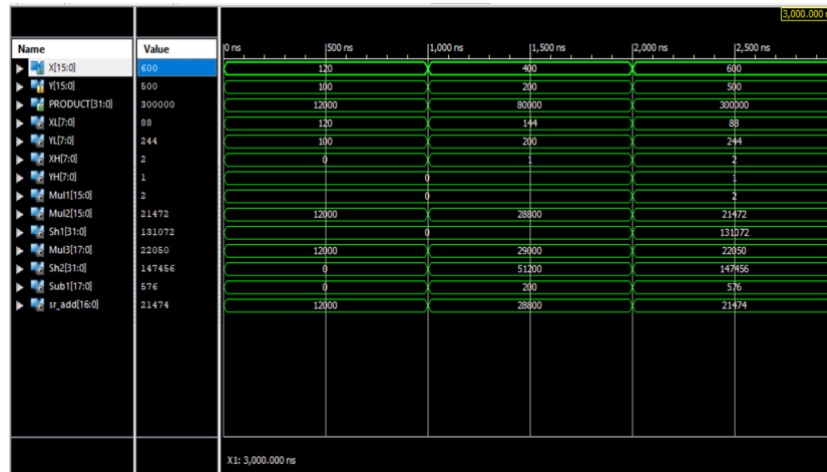


Fig 8: Simulated Waveforms of proposed Vedic multiplier

PARAMETERS: Area, delay, and power are three VLSI metrics that may be used to contrast different architectural designs. Power and area factors are taken into account in this situation. Verilog is utilized as the HDL language, and they are obtained using the XILINX 14.7 tool.

Parameter	Existed multiplier design	Proposed multiplier design
No of LUTs	278	267
Power (m.watt)	4.216	4.049

Table II: parameter comparison

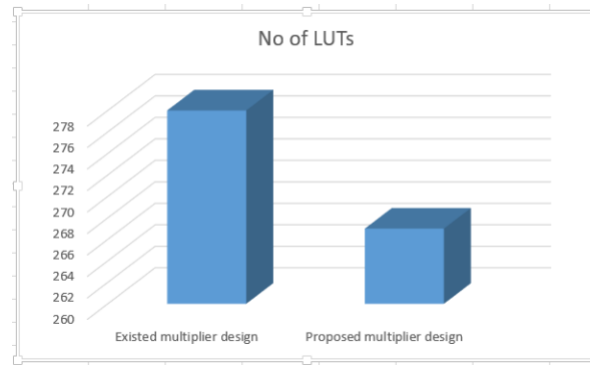


Fig 9: LUT comparison bar graph

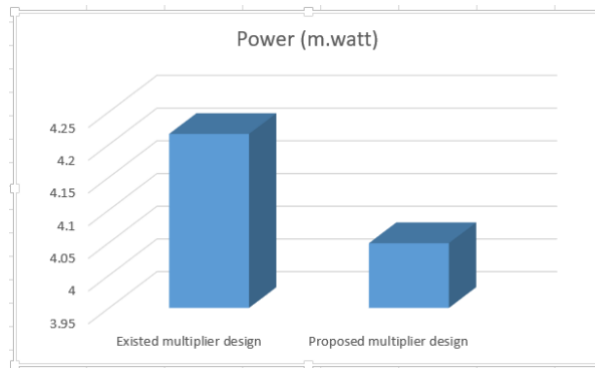


Fig 10: power comparison bar graph

V. CONCLUSION

Vedic algorithms have proved helpful in creating function circuits that are fast and have a straightforward architecture. The problem, though, was that because all of these methods are based on decimal number systems, binarization sometimes resulted in a trade-off between performance and simplicity of architecture because of conversion-hardware cost. However, there has been a recent increase in interest in Vedic algorithms, especially as a result of ingenious circuit realization, notably multipliers. The current project is significant because it modifies a well-known algorithm (Karatsuba) to incorporate an adaptive feature that permits recursive operation. Reversible logic has been presented and used to demonstrate a 16-bit multiplier with the main goal of area and power minimization so it can be used in DSP, image processing, and ASIPs that do a lot of work with computers.

The Vedic Karatsuba algorithm is its base which produces fewer partial product terms. In order to increase performance, the method is further refined utilizing an adaptive idea for computing the third product term. Additionally, by combining the reversible logic with the

suggested Square Root Carry Select Adder (SRCSA) as mentioned in this article, the compression speed of the partial product terms is also improved. The verilog HDL language is used for the implementation, synthesis, and simulation in the XILINX-ISE tool. The addition of the approximation to the design and the use of this multiplier implementation, which removes gate delays, will improve performance in dsp applications, image processing, filters, and cryptography applications in the future. Applications focused on area, speed, and power will employ it in the future.

REFERENCES

- [1] Swami Bharati Krishna Tirthaji Maharaja, “Vedic Mathematics”, Motilal Banarsidass Publishers, 1965.
- [2] Rakshith T R and RakshithSaligram, “Design of High Speed Low Power Multiplier using Reversible logic: a Vedic Mathematical Approach”, International Conference on Circuits, Power and Computing Technologies (ICCPCT-2013), ISBN: 978-1-4673-4922-2/13, pp.775-781.
- [3] M.E. Paramasivam and Dr. R.S. Sabeenian, “An Efficient Bit Reduction Binary Multiplication Algorithm using Vedic Methods”, IEEE 2nd International Advance Computing Conference, 2010, ISBN: 978-1-4244-4791-6/10, pp. 25-28.
- [4] Sushma R. Huddar, Sudhir Rao Rupanagudi, Kalpana M and Surabhi Mohan, “Novel High Speed Vedic Mathematics Multiplier using Compressors”, International Multi conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 22-23 March 2013, Kottayam, ISBN: 978-1-4673-5090-7/13, pp.465-469.
- [5] L. Sriraman and T. N. Prabakar, “Design and Implementation of Two Variables Multiplier Using KCM and Vedic Mathematics”, 1st International Conference on Recent Advances in Information Technology (RAIT -2012), ISBN: 978-1-4577-0697-4/12.
- [6] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya and Anup Dandapat, “High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics”, Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January,2011, IIT Kharagpur, ISBN: 978-1-4244-8943-5/11, pp.237-241.

- [7] Soma BhanuTej, “Vedic Algorithms to develop green chips for future”, International Journal of Systems, Algorithms & Applications, Volume 2, Issue ICAEM12, February 2012, ISSN Online: 2277-2677.
- [8] Gaurav Sharma, Arjun Singh Chauhan, Himanshu Joshi and Satish Kumar Alaria, “Delay Comparison of 4 by 4 Vedic Multiplier based on Different Adder Architectures using VHDL”, International Journal of IT, Engineering and Applied Sciences Research (IJIEASR), ISSN: 2319-4413, Volume 2, No. 6, June 2013, pp. 28-32.
- [9] Aniruddha Kanhe, Shishir Kumar Das and Ankit Kumar Singh, “Design and Implementation of Low Power Multiplier using Vedic Multiplication Technique”, International Journal of Computer Science and Communication, Vol. 3, No. 1, June 2012, pp. 131-132.
- [10] Anju and V.K. Agrawal, “FPGA Implementation of Low Power and High Speed Vedic Multiplier using Vedic Mathematics”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 2, Issue 5 Jun. 2013, ISSN: 2319 – 4200, pp. 51-57.
- [11] Animul Islam, M.W. Akram, S.D. Pable, Mohd. Hasan, “Design and Analysis of Robust Dual Threshold CMOS Full Adder Circuit in 32 nm Technology”, International Conference on Advances in Recent Technologies in Communication and Computing, 2010.
- [12] Deepa Sinha, Tripti Sharma, K.G.Sharma, Prof.B.P.Singh, “Design and Analysis of low Power 1-bit Full Adder Cell”, IEEE, 2011.
- [13] Nabihah Ahmad, Rezaul Hasan, “A new Design of XOR-XNOR gates for Low Power application”, International Conference on Electronic Devices, Systems and Applications (ICEDSA), 2011.
- [14] R.Uma, “4-Bit Fast Adder Design: Topology and Layout with Self-Resetting Logic for Low Power VLSI Circuits”, International Journal of Advanced Engineering Sciences and Technology, Vol No. 7, Issue No. 2, 197 – 205.
- [15] David J. Willingham and Izzet Kale, “A Ternary Adiabatic Logic (TAL) Implementation of a Four-Trit Full-Adder, IEEE, 2011.
- [16] Padma Devi, Ashima Girdher and Balwinder Singh, “Improved Carry Select Adder with Reduced Area and Low Power Consumption”, International Journal of Computer Application, Vol 3.No.4, June 2010.

- [17] B. Ramkumar, Harish M Kittur, P.Mahesh Kannan, “ASIC Implementation of Modified Faster Carry Save Adder”, European Journal of Scientific Research ISSN 1450-216X Vol.42 No.1, pp.53-58,2010.
- [18] Y. Sunil Gavaskar Reddy and V.V.G.S.Rajendra Prasad, “Power Comparison of CMOS and Adiabatic Full Adder Circuits”, International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.3, September 2011
- [19] Mariano Aguirre-Hernandez and Monico Linares-Aranda, “CMOS Full-Adders for Energy-Efficient Arithmetic Applications”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.19, No. 4, April 2011.
- [20] Ning Zhu, Wang Ling Goh, Weijia Zhang, Kiat Seng Yeo, and Zhi Hui Kong, “Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 8, August 2010.
- [21] Sreehari Veeramachaneni, M.B. Srinivas, “New Improved 1-Bit Full Adder Cells”, IEEE, 2008.International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, February 2012164
- [22] Tripti Sharma, K.G.Sharma, Prof.B.P.Singh, Neha Arora, “High Speed, Low Power 8T Full Adder Cell with 45% Improvement in Threshold Loss Problem”, Recent Advances in Networking, VLSI and Signal Processing.
- [23] G.Shyam Kishore, “A Novel Full Adder with High Speed Low Area”, 2nd National Conference on Information and Communication Technology (NCICT) 2011 Proceedings published in International Journal of Computer Applications® (IJCA).
- [24] Shubhajit Roy Chowdhury, Aritra Banerjee, Aniruddha Roy, Hiranmay Saha, “A high Speed 8 Transistor Full Adder Design using Novel 3 Transistor XOR Gates”, International Journal of Electrical and Computer Engineering 3:12 2008.
- [25] Romana Yousuf and Najeeb-ud-din, “Synthesis of Carry Select Adder in 65 nm FPGA”, IEEE.
- [26] Shubin.V.V, “Analysis and Comparison of Ripple Carry Full Adders by Speed”, Micro/Nano Technologies and Electron Devices (EDM), 2010, International Conference and Seminar on, pp.132-135, 2010.

[27] Pudi. V, Sridhara. K, “Low Complexity Design of Ripple Carry and Brent Kung Adders in QCA”, Nanotechnology, IEEE transactions on, Vol.11, Issue.1, pp.105-119, 2012.

[28] Jian-Fei Jiang; Zhi-Gang Mao; Wei-Feng He; Qin Wang, “A New Full Adder Design for Tree Structured Arithmetic Circuits”, Computer Engineering and Technology (ICCET), 2010, 2nd International Conference on, Vol.4, pp. V4-246-V4- 249, 2010.