A VHDL Code for Area and Delay Efficient 64-Bit Carry Skip Adder

C.Yamini[#1] **M.Krishnamurthy**[*2]

[#]PG Scholar, VLSI Design,
PSNA college of engineering and technology,
Dindigul, India.
[1]Yaminipriyanka06@gmail.com
Assistant professor, Department of ECE,
PSNA college of engineering and technology,
Dindigul, India.
[2]k_murthym@yahoo.co.in

***Abstract*—Adders are the basic element in many data processing processors. By improving a performance of the adder would advance the execution of binary operations inside a circuit. The speed of addition is limited by the time required to transmit a carry through the adder. The aim of this paper is to develop 32 bit carry skip adder architecture by the technique of replacing ripple carry adder with carry look-ahead adder in the conventional structure which reduces the number of gates and delay. The delay is also reduced by carry chain optimization technique. This work uses a simple and an efficient gate-level modification in the regular structure which reduces the area and delay of the CSKA and this work estimates the performance of the proposed designs with conventional structure in terms of Area, delay and synthesis are implemented in Xilinx 14.2 of VHDL. The results are compared with proposed structure and conventional structure of CSKA.**

***Keywords*—Ripple Carry Adder (RCA), Carry Look-ahead adder (CLA), Carry Skip Adder (CSKA).**

I. INTRODUCTION

In the fast moving world, the major issues are area, low power and less delay requirements in the designs. In many applications such as digital signal processor, Encryption in cryptography, signal processing in FIR filter etc Binary adders are the basic operation in the circuit. There are many adder techniques are available such as Ripple carry adder, Carry look-ahead adder, Carry bypass adder, Carry select adder, Parallel prefix adder, etc which has its own disadvantages and advantages. The major speed limitation issues are carry propagation and carry generation in any adder. Carry skip adder is developed to solve carry propagation delay which reduces area and delay drastically to maximum extent.

For the improvement in operational speed, different adder related designs and optimization methods are described. *Mitra et al* presented the detailed design of Reversible Fault Tolerant-Full Adder (RFT-FA) [3] with the minimum quantum cost. The merging of minimization of gates and garbage outputs in RFT based Carry Look-ahead Adders (CLA) provided. The creation of area efficient and low power high speed MAC unit comprised Carry Skip Adders (CSKA). Hence, the research works turn to CSKA Design and the optimization in CSKA. *Kalaiselvi et al* further reduced the power consumption and improved the operational speed [4]. The comparison of power consumption and performance analysis between the proposed and existing methods conveyed the suitability of CSKA in future MAC designs. *Rajmohan et al* improved the design parameters gate count, area and power by the integration of reversible mechanism with the CSKA [5]. The raise of some problems during the integration of DSP processors are with the quantum computers. *Shukla et al* designed the low power arithmetic and data path units by using the reversible logic implementation in Carry Look-ahead Adder (CLA) [6]. The investigation about the delay performance with the evolution of carry tree adders was an important research area. *Cury et al* discussed the design of fastest type adders by the application of carry chain in traditional RCA [7] and support the minimum delay performance for various bit sizes ranging from 128 to 256 bit. The heat dissipation by the components was more. Hence, *Misra et al* introduced the reversible logic gate called Inventive0gate [8] which was an efficient and optimized design in order to reduce the gate count. Thereby, the minimization of heat dissipation is achieved.

In this paper, CSKA structure is modified by replacing ripple carry adder with carry look-ahead adder and also uses AOI-OAI logic instead of multiplexers. It further reduces the number of gate count, delay, power and area. Moreover, 32-bit CSKA structure is designed by the Xilinx tool of VHDL. In this paper work in brief is organized as follows: In section II proposed carry skip adder architecture is discussed. In section III results and discussions are presented and finally from results conclusion are drawn in section IV.

II. PROPOSED CARRY SKIP ADDER DESIGN

This section presents the proposed 32-bit Carry Skip Adder (CSKA) design by replacement of multiplexers with AOI and OAI logic and ripple carry adder is replaced by carry look-ahead adder in Xilinx of VHDL environment. By using Boolean function gate minimization, the critical path delay minimization is achieved. The following are the description of conventional structure and proposed structure of carry skip adder logical design.

A. RIPPLE CARRY ADDER LOGIC

The Ripple carry adder (RCA) is the simplest adders to design. This ripple carry adder takes in 2 N-bit inputs and produces 1-bit carry out bit and (N+1) output bits as N-bit

Sum. The RCA is built from N full adders connected in series, with the carry out bit of first full adder is connected to the carry in bit of the next full adder. Ripple carry adder for 4-bit addition is shown in Fig.1. The conventional structure is implemented with ripple carry adder block which consists of one half adder and 3 full adder connected in series. In RCA, carry of first half adder is ripple to next full adder so there will occur some delay. To reduce ripple carry delay, carry look-ahead adder is used in proposed CSKA structure. The proposed CSKA is compared with conventional structure which is shown in fig.2.

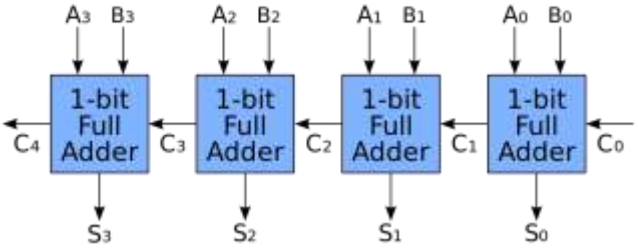

Fig.1. Logic circuit of 4-bit Ripple Carry Adder

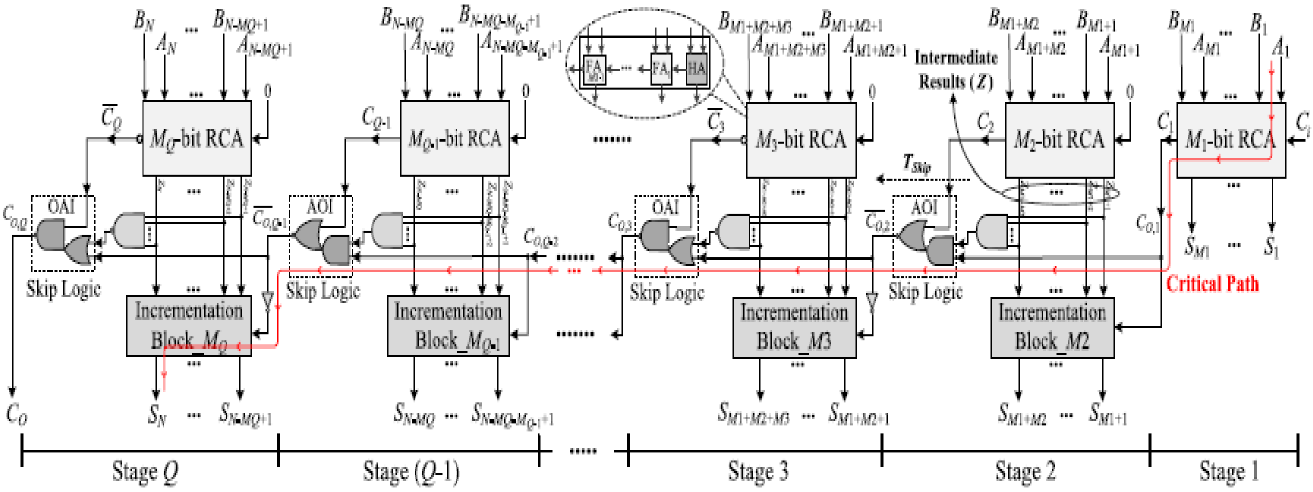

Fig.2.CSKA conventional structure

The Design of proposed work comprises basic half adders and full adders, sequence of 32-bit Carry Skip Adder (CSKA) in Xilinx of VHDL environment. The modules of designing 32-bit carry skip adder are the following:

- Carry look-ahead adder logic
- Skip logic and Incrementation logic
- 32-bit CSKA design

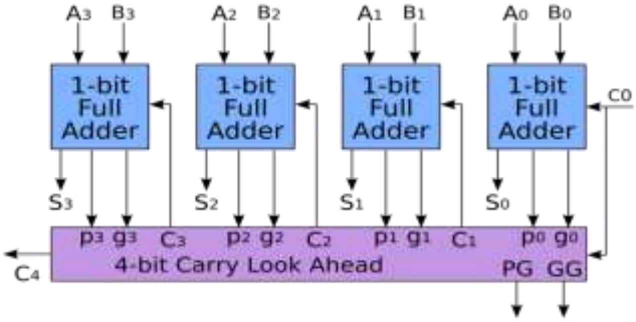

Fig. 3.Logic Circuit of 4-bit Carry Look-ahead adder

B. *CARRY LOOK AHEAD ADDER LOGIC*

Carry look-ahead adder is simple adder which uses the concepts of carry generation and carry propagation. The carry look-ahead adder logic will determine that bit pair will propagate a carry or generate a carry. This allows the circuit to determine the carry ahead of time by pre-process the two numbers being added. The logic circuit of 4-bit Carry Look-ahead adder is shown in Fig. 3. The carry look-ahead adder is replaced instead of ripple carry adder due to reduction of number of logical gate count. For single bit addition of RCA is built using 7 components of logic gates but carry look-ahead adder of single bit addition is built within 5 components. Even though RCA and CLA have same execution time and delay for single bit addition, for 32 bit design, number of gates reduction leads to less delay than ripple carry adder.

In RCA, data flow in a chain as a bit length go on increasing delay and takes time to propagate the carry to overcome this problem , here CLA is used to solve by determining the carry in advance. Each 4bit CLA are considered as stage 1 and for 32-bit, 8 stages of CLA logic is designed. Each 1 stage has 4 bit carry look-ahead adder. By the following equations the carry out signals of 4-bit carry look-ahead adder are computed:

$$c_1 = g_0 + p_0c_0 \quad \ldots\ldots\ldots\ldots\ldots (1)$$

$$c_2 = g_1 + p_1c_1 = g_1 + p_1g_0 + p_1p_0c_0 \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$c_3 = g_2 + p_2c_2 = g_2 + p_2g_1 + p_2\, p_1g_0 + p_2p_1p_0c_0 \quad \ldots (3)$$

$$c_4 = g_3 + p_3c_3 = g_3 + p_3g_2 + p_3p_2g_1 + p_3p_2\, p_1g_0 + p_3p_2p_1p_0c_0 \quad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

C. SKIP LOGIC AND INCREMENTATION BLOCK

Instead of using multiplexer for carry selection, skip logic is implemented. Skip logic is nothing but an AOI (AND-OR-Invert) and OAI (OR-AND-Invert) LOGIC which contains small number of transistors with the lower delay, power and area consumption. AOI and OAI logic helps to generate carry and get added with next stage sum process simultaneously with help of incrementation block which adds a single bit carry from skip logic and produce final sum.

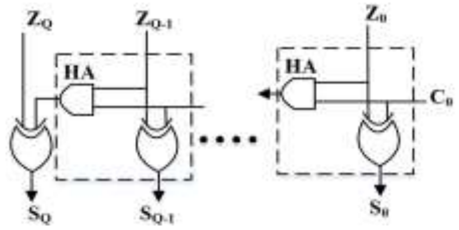

Fig.4.Incrementation structure

The modified incrementation block contains of Half Adders in chain as shown in fig.4. The incrementation block produces the number of intermediate results up to the level is defined by,

$$\mathrm{Kj} = \sum_{r=1}^{j-1} \mathrm{Mr} \quad (\mathrm{j} = 2, \dots \dots \dots \dots \mathrm{Q}) \qquad (5)$$

The considerable reduction of delay is provided with the consideration of carry output generated in overall design rather than the carry output of incrementation stage. From fig.4, the carry outputs of qth stage Co, q is obtained on the basis of intermediate results and carry output of previous stage $Co, q-1$ and the carry output of RCA stage Cj. If Cj is one, then Co, q is also one. For Cj is zero, then check whether the product of intermediate results is one, then the output is same as $Co, q-1$.

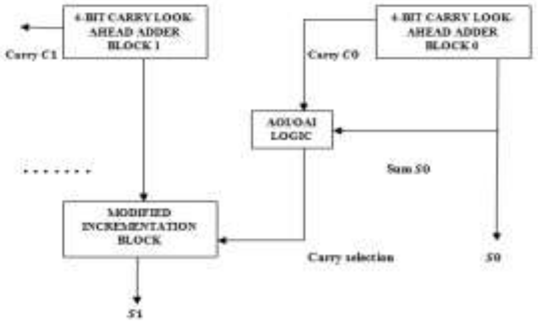
4-BIT CARRY LOOK AHEAD ADDER BLOCK 1
Carry C1
4-BIT CARRY LOOK AHEAD ADDER BLOCK 0
Carry C0
AOI/OAI LOGIC
Sum S0
MODIFIED INCREMENTATION BLOCK
Carry selection
S0
S1

Fig.5.Proposed CSKA Structure

D. 32-bit CSKA DESIGN

The proposed CSKA Design is similar to conventional CSKA with the difference introduced in incremental block where the multiplexers are replaced by the Boolean function based gate minimization and ripple carry adder is replaced by carry look-ahead adder. The proposed CSKA structure is shown in fig.5. For 32 bit addition, 8 stages of carry look-ahead adder are implemented. The sum of first stage and carry from CLA is processed by skip logic and carry is selected by complementing the given inputs and carry output $c0$ is given to incrementation block where the carry and next stage sum is getting added and produce the final result of this stage. Similarly the overall 32 bit addition takes place in the proposed system.

E. PERFORMANCE ANALYSIS

In proposed CSKA Design, The utilization of modified incrementation block reduces gate count, power and delay due to the reduction of delay in critical path. In general, CSKA structure of critical path delay in conventional one depends upon the carry delay and sum output expressed as

$$T_{D} = [M \times T_{CARRY}] + [(\tfrac{N}{M} - 1) \times T_{MUX}] + [(M - 1) \times T_{CARRY} + T_{SUM}] \qquad (6)$$

The proposed Design contains three parts namely; the path of the first stage of FA chain, the path of skip logics and the incrementation block in last stage. The total critical path delay depends upon the delay of each individual parts and expressed as

$$T_{D} = [M_1 T_{CARRY}] + [(Q - 2) T_{SKIP}] + [(M_Q - 1) T_{AND} + T_{XOR}]$$
(7)

The delay of skip logic is computed by taking the average of AOI-OAI logic defined by,

$$\mathrm{Tskip} = \frac{\mathrm{Taoi+Toai}}{2} \qquad (8)$$

With this modification, the equation (7) is modified as,

$$T_{D} = [M_1 T_{CARRY}] + [(Q - 2) \frac{Taoi+Toai}{2}] + [(M_Q - 1) T_{AND} + T_{XOR}]$$
(9)

From the equation (6) and (9), the delay of skip logic is minimized for the same number of operational stages as conventional. Compared to T_{CARRY} and T_{SUM} , T_{AND} and T_{XOR} are small. Hence, the reduction of delay in skip logic reduces the delay of overall structure.

III. RESULTS AND DISCUSSION

The table I show that comparative analysis of proposed Design with the conventional structures on the parameters of slices, lut's, clocking frequency, throughput and average fanout and level of logic of 32- bit addition. The reduction of delay in path effectively optimizes these parameters. The speed is evaluated with respect to the delay. From the utilization of number of slices and Look-up tables, Area can be calculated. The Xilinx 14.2 version of VHDL is used for the implementation of existing and proposed carry skip adder. The results of 32-bit addition and 64-bit addition are tabulated in table 1 and table 2 respectively. The comparison is done for two factors: speed and area. Fig 8.Shows that Simulation results of 64-bit proposed carry skip adder. Fig.9 and fig.10 shows that Comparison of 32-bit and 64-bit carry skip adder of proposed and conventional carry skip adder

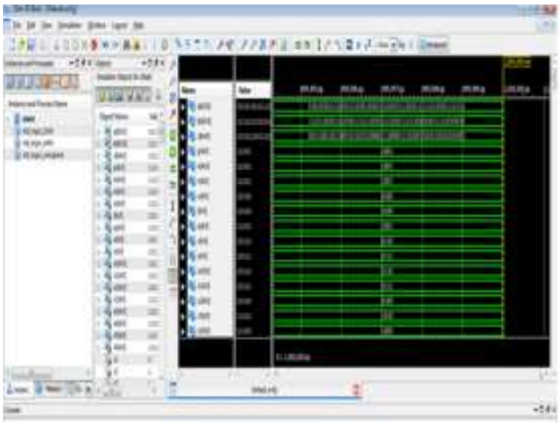

Fig.8.Simulation result of 64-bit proposed carry skip adder

PARAMETERS	CONVENTIONAL SYSTEM	PROPOSED SYSTEM
SLICES	93	32
LUT'S	126	57
PATH DELAY TIME(ns)	15.095	7.578
AVERAGE FANOUT	2.54	2.51
LEVEL OF LOGIC	13	14
CLOCKING FREQUENCY(MHz)	66.246	131.96
THROUGHPUT (Mbps)	2119.89	4222.75

Table 1.Results of conventional and proposed CSKA adder for 32-bit addition

PARAMETERS	CONVENTIONAL SYSTEM	PROPOSED SYSTEM
SLICES	123	62
LUT'S	217	117
PATH DELAY TIME (ns)	28.885	14.501
AVERAGE FANOUT	2.54	2.51
LEVEL OF LOGIC	25	26
CLOCKING FREQUENCY (MHz)	34.62	68.96
THROUGHPUT (Mbps)	2215.6829	4413.4807

Table2. Results of proposed and conventional structure for 64-bit addition

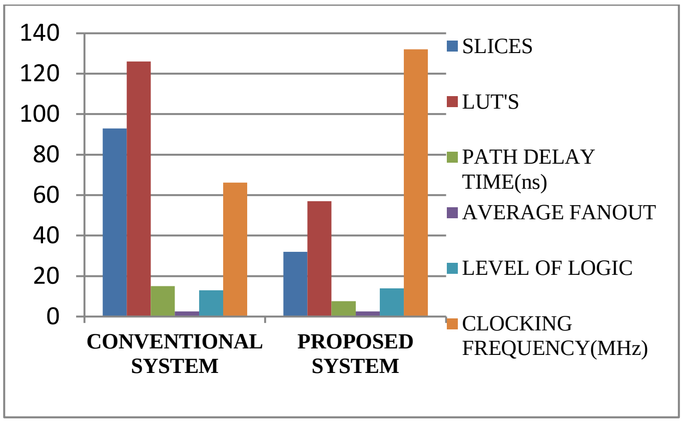

Fig.9.Comparison of 32-bit carry skip adder of proposed and conventional carry skip adder

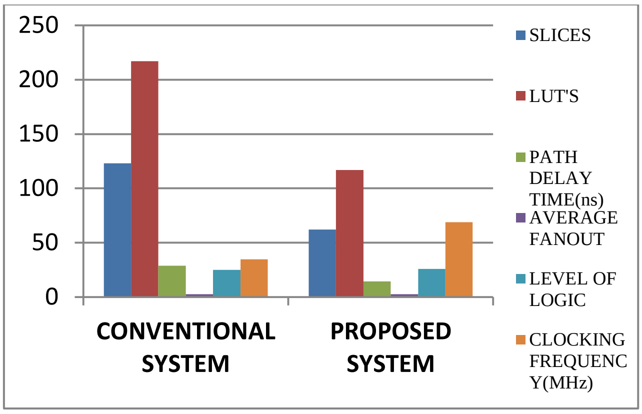

Fig.10.Comparison of 64-bit carry skip adder of proposed and conventional carry skip adder

IV. CONCLUSION

In this paper, an efficient approach is proposed to reduce the area and delay of CSKA architecture. The reduction in the number of gates is obtained by simply replacing ripple carry adder with carry look-ahead adder in the structure. This paper presents the implementation of CSKA conventional structure and proposed carry skip adder for 32-bit addition. To evaluate the performance of these adders, Xilinx 14.2 version of VHDL is used. The utilization of slices and LUT’s for 32-bit addition is less for the carry skip adder. The speed performance is estimated with delay parameter, the delay of CSKA is least when compared with conventional one. The results analysis shows that the proposed CSKA structure is better than conventional one. Therefore for VLSI hardware implementation the proposed CSKA architecture is low area and high speed approaches. To enhance the overall system performance, the proposed carry skip adder can be further used in signal processing applications such as FIR filter, cryptographic applications such as encryption and hashing functions etc.

ACKNOWLEDGMENT

C.Yamini would like to thank Mr.M.Krishnamurthy, Assistant professor ECE department who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work effectively and successfully.

REFERENCES

[1] M.Bahadori, M.Kamal, Ali Afzali-Kusha," High speed energy efficient carry skip adder operating under a wide range of supply voltage levels "IEEE transactions on VLSI systems, February 13, 2015.

[2] Daljit Kaur et al," Performance Analysis of 64-Bit Carry Look Ahead Adder" , International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 689-693.

[3] S. K. Mitra and A. R. Chowdhury, "Minimum Cost Fault Tolerant Adder Circuits in Reversible Logic Synthesis," in *25th International Conference on VLSI Design (VLSID), 2012*, pp. 334-339.

[4] K. Kalaiselvi and H. Mangalam, "Area Efficient High Speed and Low Power MAC Unit," *International Journal of Computer Applications,* vol. 67, pp. 40-44, 2013.

[5] M. Rajmohan and D. S. Lenin, "A Novel Design of Carry Skip BCD Adder using Reversible Gates," *International Journal of Computer Applications,* vol. 73, pp. 46-51, 2013.

[6] S. Shukla, T. Verma, and R. Jain, "Design of 16 Bit Carry Look Ahead Adder Using Reversible Logic," *International Journal of Electrical, Electronics and Computer Engineering,* vol. 3, pp. 83-89, 2014.

[7] C. Cury and M. Nisanth, "Design of Parallel Prefix Adders using FPGAs," *IOSR Journal of VLSI and Signal Processing,* vol. 4, pp. 45-51, 2014.

[8] N. K. Misra, M. K. Kushwaha, S. Wairya, and A. Kumar, "Cost efficient design of reversible adder circuits for low power applications," *arXiv preprint arXiv: 1509.04618,* 2015.

[9] P. Gurjar, R. Solanki, P. Kansliwal, and M. Vucha, "VLSI Implementation of adders for high speed ALU," in *Annual IEEE India Conference (INDICON), 2011* 2011, pp. 1-6.

[10] J. Luu, C. McCullough, W. Sen, S. Huda, Y. Bo, C. Chiasson, *et al.*, "On Hard Adders and Carry Chains in FPGAs," in *IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2014* 2014, pp. 52-59.

[11] W. Sung Sik and R. Sarpeshkar, "A spiking-neuron collective analog adder with scalable precision," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2013* 2013, pp. 1620-1623.

[12] T. Francis, T. Joseph, and J. K. Antony, "Modified MAC unit for low power high speed DSP application using multipler with bypassing technique and optimized adders," in *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT),2013* 2013, pp. 1-4.

[13] N. Jain, P. Gour, and B. Shrman, "A high speed low power adder in dynamic logic base on transmission gate," in *International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2015* 2015, pp. 1-5

[14] B. K. Mohanty and S. K. Patel, "Area&#x2013;Delay&#x2013;Power Efficient Carry-Select Adder," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 61, pp. 418-422, 2014.

[15] P. Sreenivasulu, K. S. Rao, M. Reddy, and A. V. Babu, "Energy and Area efficient Carry Select Adder on a reconfigurable hardware," *International Journal of Engineering Research and Applications,* vol. 2, pp. 436-440, 2012.

[16] K. Saranya, "Low Power and Area-Efficient Carry Select Adder," *International Journal of Soft Computing and Engineering (IJSCE) ISSN,* pp. 2231-2307, 2013.

[17] P. Saxena, U. Purohit, and P. Joshi, "Analysis of Low Power, Area Efficient & High Speed Fast Adder," *International Journal of Advanced research on Computer and Communication Engineering,* 2013.

[18] M. Dorrigiv and G. Jaberipur, "Low area/power decimal addition with carry-select correction and carry-select sum-digits," *VLSI Journal,* vol. 47, pp. 443-451, 9// 2014.

[19] K. Shirisha and D. S. Rao, "Area-Delay-Power Efficient Carry-Select Adder," 2015.

[20] L. Jamal, M. Shamsujjoha, and H. H. Babu, "Design of optimal reversible carry look-ahead adder with optimal garbage and quantum cost," *International Journal of Engineering and technology,* vol. 2, pp. 44-50, 2012.

[21] S. K. Patnayak, M. Raja, and D. Sailaja, "Design Carry Select Adder With D-Latch," 2015.

[22] H. Bhagyalakshmi and M. Venkatesha, "Optimized design of BCD adder and Carry skip BCD adder using reversible logic gates," *International journal on computer science and Engineering,* vol. 3, pp. 1439-1449, 2011.

[23] M. S. R. Md.Belayet Ali, Tahmina Parvin, "Optimized Design of Carry Skip BCD adder using new FHNG reversible logic gates," *International Journal of Computer Science,* vol. 9, p. 3, 2012.