Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

Implementation of 32-Bit Adders using Different Full Adders

M. Yamini Divya[1]
[1]M.TECH. Student, ECE Department,
Velagapudi Ramakrishna Siddhartha Engineering College
(VRSEC), Vijayawada, India.

B. K. L. Aruna[2]
[2]Assistant Professor, ECE Department,
Velagapudi Ramakrishna Siddhartha Engineering College
(VRSEC), Vijayawada, India.

Abstract: **An adder is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU. They are also used in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators and similar operations. Adders play a vital role in the digital signal processing systems. The design of 32-bit adders is of high importance because 32-bit architecture is common and widely used in many digital systems and processors. In this paper, the design and implementation of various 32-bit adders like Ripple Carry Adder (RCA), Carry Increment Adder (CIA) and Carry Skip (or) Carry Bypass Adder (CSKA) for different full adders is done using Verilog HDL. The results are obtained by executing Verilog code in Xilinx 14.5 ISE for the Spartan 3E family device with speed grade of -5.**

Keywords: Carry Increment Adder (CIA), Carry Skip Adder (CSKA), Conventional Full Adder (FA), Full Adder using 2 by 1 MUX (FA1), Full Adder using 4 by 1 MUX (FA2), 2 by 1 MUX based Full Adder (FA3), Ripple Carry Adder (RCA).

1. INTRODUCTION

Adders are one of the most widely implemented blocks of microprocessor chips and digital components in the digital integrated circuit design. They are the necessary part of Digital Signal Processing (DSP) applications. With the advances in technology, researchers have tried and are trying to design adders which offer either high speed, low power consumption, less area or the combination of them. Every adder generates a carry value that has to be propagated through the circuit within a series of adders. This contributes largely to the critical path delay of the circuit. By reducing the number of stages the carry has to be propagated, the delay in the circuit can be reduced. The required sum is selected using a multiplexer.

The objective of this project is to design different architectures of adders and to study their respective performance, cost and design time. Adders are being used extensively in many processor architectures and computational units. It is a vital part in any processor or chip. By reducing the area occupied by these adders, the critical path delay can also be reduced. This can be done by implementing different designs of these adders. The ultimate aim of reducing the area and design is to optimize the cost of manufacture and improve the efficiency of the processor. Every adder generates a carry value that has to be propagated through the circuit within a series of adders. This contributes largely to the critical path delay of the circuit. By reducing the number of stages the carry has to be propagated, the delay in the circuit can be reduced. This can be done by implementing different architectures of the adder design and by incorporating varied logic to propagate the carry through the least number of stages possible. One by looking ahead of several blocks, i.e., by identifying where the actual output is and by delivering the carry signal right to that stage or by calculating the sum before the propagation is started. The required sum is selected using a multiplexer. In the above cases, the carry signal is not propagated through more than three stages which reduce the delay in the circuit. The various designs of adders are explained as follows. In this project, we implement Ripple Carry Adder (RCA), Carry Increment Adder (CIA) and Carry Bypass Adder (CBYA). The above designs are verified by performing RTL design using synthesis, Logic simulation using Xilinx simulator, Place and Route for power analysis, Synthesis report for area and delay.

2. RELATED WORK

A. Conventional Full Adder (FA)

The traditional full adder circuit is obtained using 2 XOR gates, 3 AND gates and an OR gate. The Boolean expressions used are

Sum = a $\oplus$ b $\oplus$ cin and Carry = (a·b) + (b·cin) + (cin·a)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

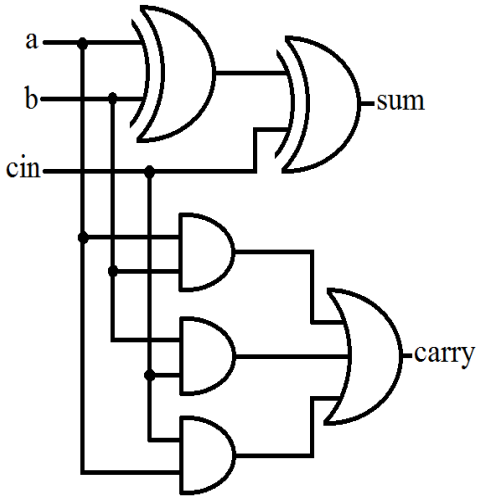

Figure 1. Conventional Full Adder (FA)

B. Full Adder Using 2 by 1 MUX (FA1)

Mux based full adder circuit is obtained using two 2-to-1 multiplexers, 1 XOR gate and an NOT gate.

Sum = $\overline{(b \oplus cin)}a + (b \oplus cin)\bar{a}$

Carry = $\overline{(b \oplus cin)}b + (b \oplus cin)a$

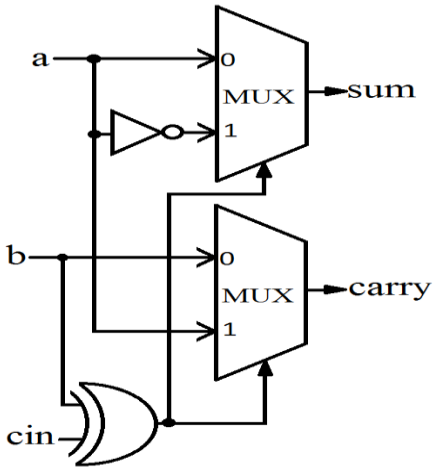

Figure 2. Full Adder using 2 by 1 MUX (FA1)

C. Full Adder using 4 by 1 MUX (FA2)

Mux based full adder circuit is obtained using two 4-to-1 multiplexers and an inverter (NOT gate).

Sum = $\bar{a}\bar{b}cin + \bar{a}b\overline{cin} + a\bar{b}\,\overline{cin} + abcin$

S = cin $(\overline{a\bar{b}} + ab) + \overline{cin}\,(\bar{a}b + a\bar{b})$

S = cin $(\overline{a \oplus b}) + \overline{cin}\,(a \oplus b)$

SUM = $a \oplus b \oplus cin$

Carry = $(\bar{a}\bar{b}*0) + \bar{a}bcin + a\bar{b}cin + (ab*1)$

C = $ab + cin\,(\bar{a}b + a\bar{b})$

CARRY = $ab + (a \oplus b)cin$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

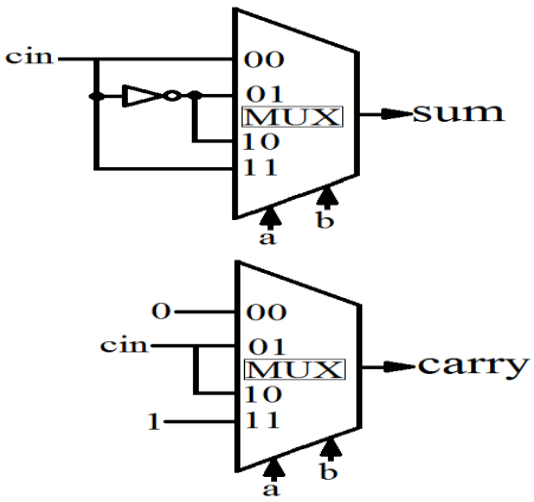

Figure 3. Full Adder using 4 by 1 MUX (FA2)

D. Two by One MUX based Full Adder (FA3)

Mux based full adder circuit is obtained using two 2-to-1 multiplexers, 1 XOR gate and an XNOR gate.

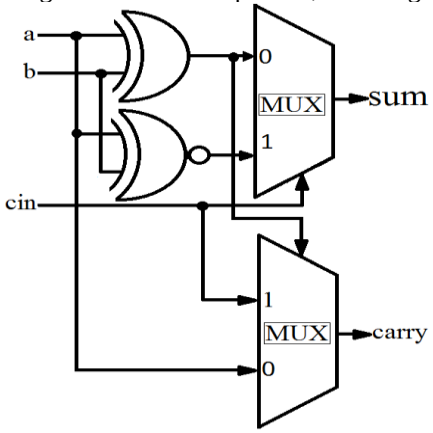

Figure 4. Two by One MUX based Full Adder (FA3)

$S = (a \oplus b)\overline{cin} + (\overline{a \oplus b})cin$
$Sum = a \oplus b \oplus cin$
$Carry = (a \oplus b)cin + (\overline{a \oplus b})a$

3. IMPLEMENTATION OF 32-BIT ADDERS

A. Ripple Carry Adder (RCA)

It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is called a ripple-carry adder (RCA), since each carry bit "ripples" to the next full adder. Note that the first (and only the first) full adder may be replaced by a half adder (under the assumption that $C_{in} = 0$). RCA contains series structure of Full Adders (FA); each FA is used to add two bits along with carry bit. The carry generated from each full adder is given to next full adder and so on. Hence, the carry is propagated in a serial computation. Hence, delay is more as the number of bits is increased in RCA.

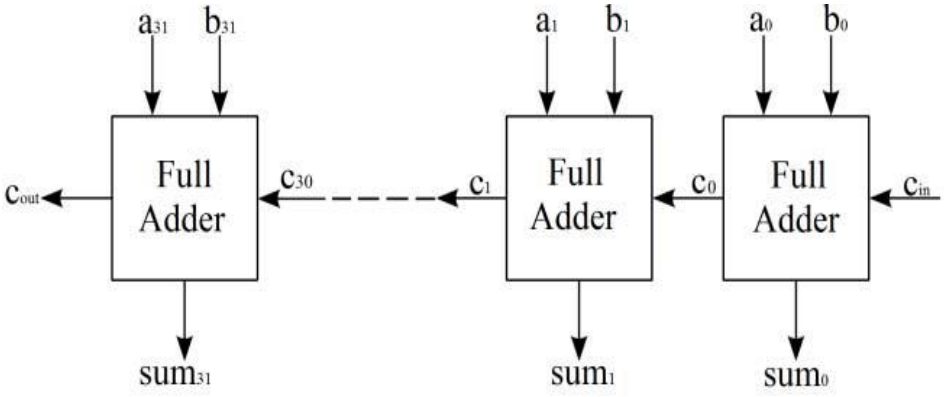

Figure 5. 32-Bit Ripple Carry Adder (RCA)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

B. Carry Increment Adder (CIA)

The design of Carry Increment Adder (CIA) consists of RCA's and incremental circuitry. The incremental circuit can be designed using HA's in ripple carry chain with a sequential order. The addition operation is done by dividing total number of bits in to group of 4bits and addition operation is done using several 4bit RCA's. The architecture of CIA is shown in fig. 6 .

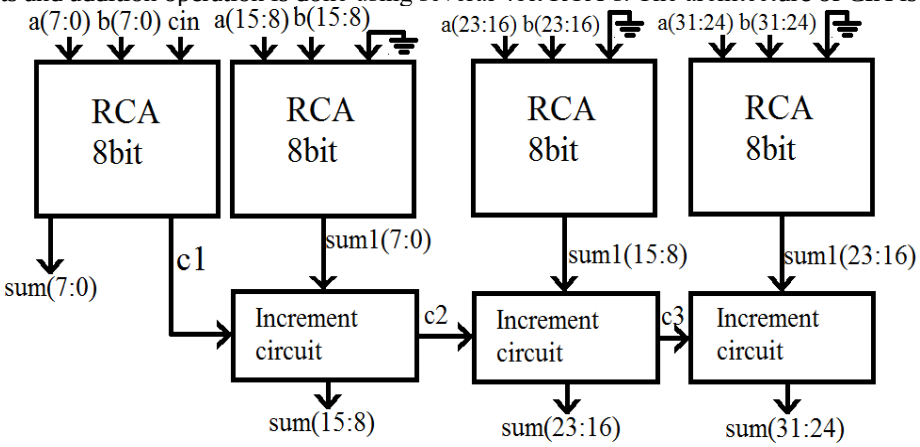

Figure 6. 32-Bit Carry Increment Adder (CIA)

In carry increment adder, various RCA blocks are used to compute the results. The first RCA block is given carry-in as input along with addends to get carry (c1) and sum. For the rest RCA blocks, the carry-in is given as logic '0' to get temporary sum (sum1) and temporary carry which are given to increment circuit. Increment circuit consists of half adders which add temporary sum and carries to get the actual sum (sum) and carry (cy). The carry-out of increment circuit is obtained by performing OR operation between carry (cy) and carry-out of the previous stage. As the carry-in for the RCA stages is logic '0' and hence the carry propagation delay decreases.

C. Carry Skip Adder (CSKA)

A carry-skip adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder. Unlike other fast adders, carry-skip adder performance is increased with only some of the combinations of input bits. This means, speed improvement is only probabilistic. Carry skip adder uses the principle of skip logic in carry propagation. It is used to speed addition operation by adding a propagation of carry bit around entire adder. It consists two logical gates AND gate is used for carry -in bit which compares with propagated signals.

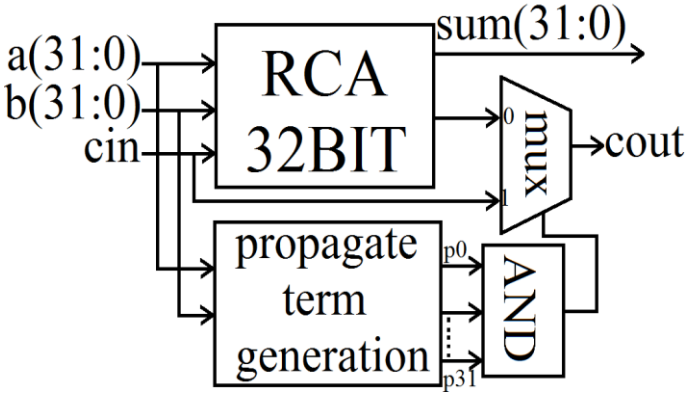

Figure 7. 32-Bit Carry Skip Adder (CSKA)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

4. SIMULATION RESULTS

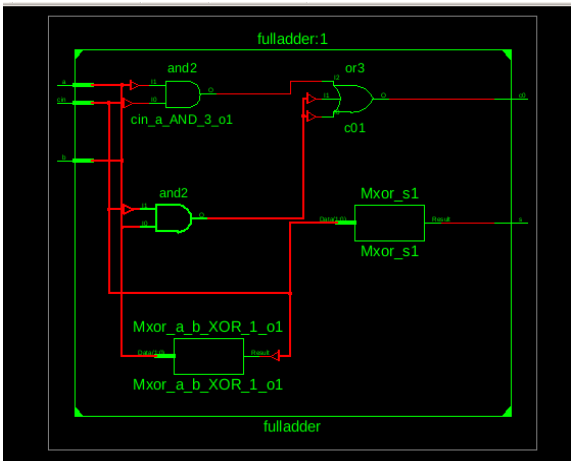

Figure 8. RTL Schematic of Conventional Full Adder (FA)

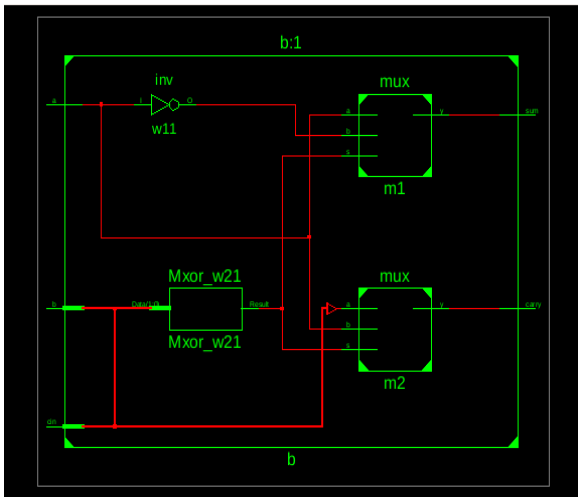

Figure 9. RTL Schematic of Full Adder using 2 by 1 MUX (FA1)

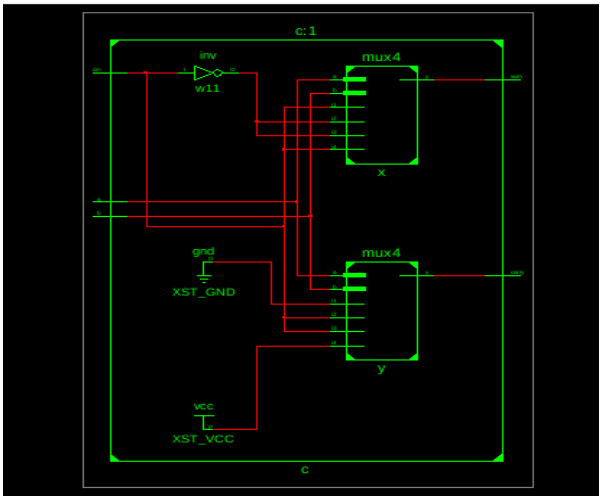

Figure 10. RTL Schematic of Full Adder using 4 by 1 MUX (FA2)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

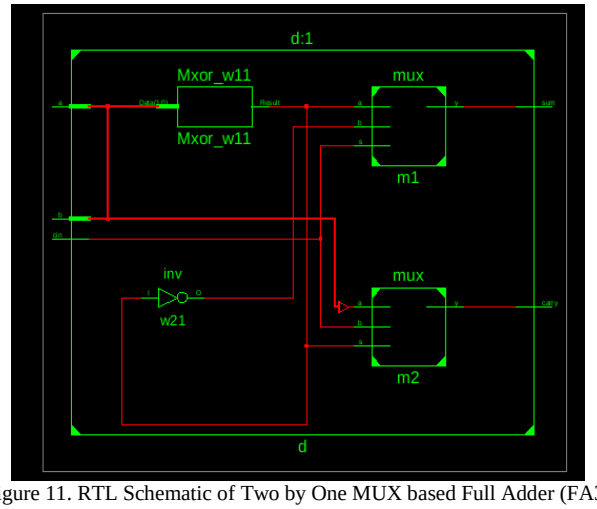

Figure 11. RTL Schematic of Two by One MUX based Full Adder (FA3)

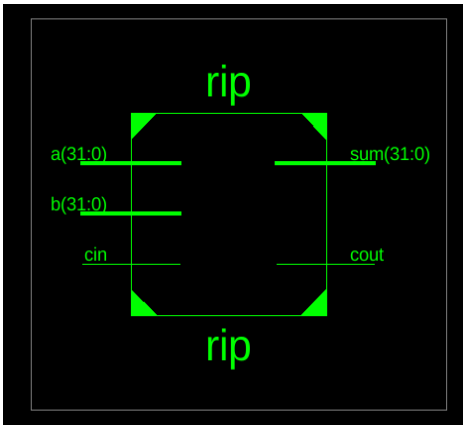

Figure 12. RTL Schematic of 32-bit Ripple Carry Adder (RCA)

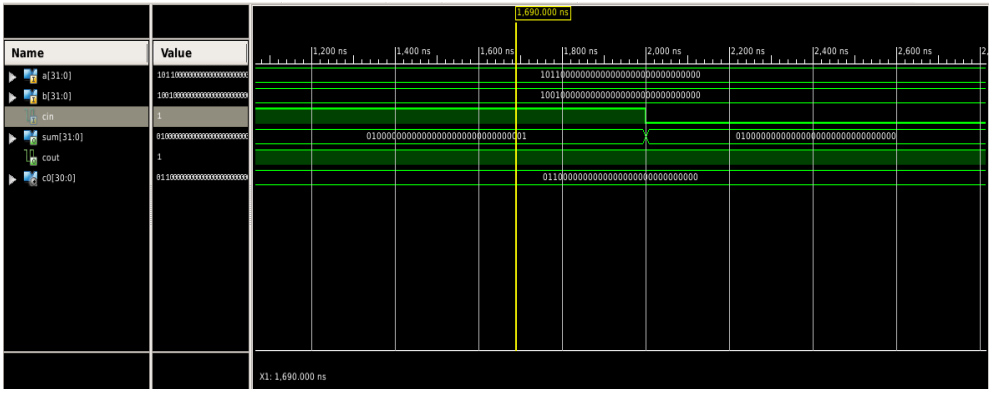

Figure 13. Simulation result of 32-bit Ripple Carry Adder (RCA)

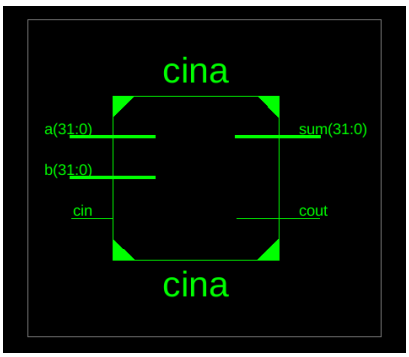

Figure 14. RTL Schematic of 32-bit Carry Increment Adder (CIA)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

Figure 15. Simulation result of 32-bit Carry Increment Adder (CIA)

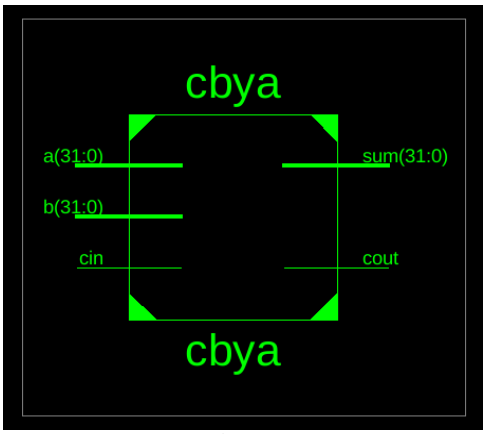

Figure 16. RTL Schematic of 32-bit Carry Skip Adder (CSKA)

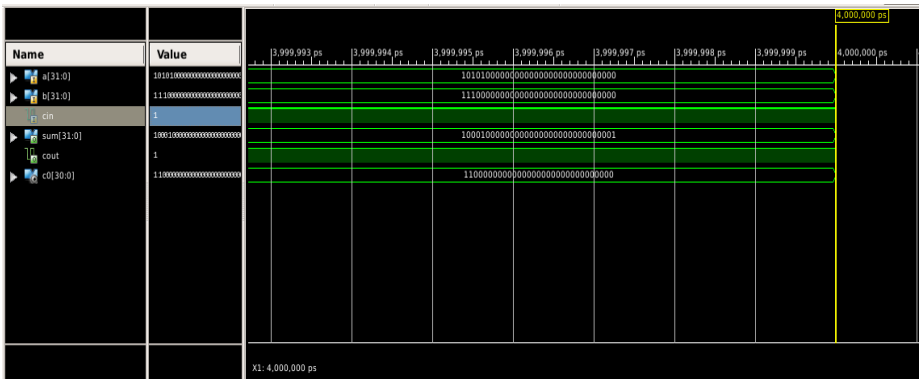

Figure 17. Simulation result of 32-bit Carry Skip Adder (CSKA)

5. SYNTHESIS REPORT

RCA, CSKA and CIA are designed by using Verilog HDL. These adders are simulated using different full adders. Simulation and synthesis are done by using Xilinx ISE 14.5 for the Spartan-3E family device with a speed grade of -5.

	LUTs	Slices	Delay(ns)
RCA (FA)	64	37	38.665
CIA (FA)	88	50	26.57
CSKA (FA)	101	56	25.514

Table 1. Comparison for different 32-bit adders with FA

	LUTs	Slices	Delay(ns)
RCA (FA1)	64	37	37.456
CIA (FA1)	88	51	26.336
CSKA (FA1)	102	57	24.63

Table 2. Comparison for different 32-bit adders with FA1

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 10, October-2020

	LUTs	Slices	Delay(ns)
RCA (FA2)	64	37	38.665
CIA (FA2)	88	50	26.57
CSKA (FA2)	102	57	26.489

Table 3. Comparison for different 32-bit adders with FA2

	LUTs	Slices	Delay(ns)
RCA (FA3)	64	37	38.665
CIA (FA3)	92	52	26.354
CSKA (FA3)	102	57	26.465

Table 4. Comparison for different 32-bit adders with FA3

From the above tables, various comparisons for adders are obtained and observed. It is evident that using one of full adder the LUTs, slices is very less for RCA by using any full adder when compared with CSKA and CIA whereas CSKA surpasses other two adders in terms of delay performance.
The delay for RCA is more than the CIA and CSKA. It is also shown that delay is reduced for RCA, CIA and CSKA using the FA1.

6. CONCLUSION

In this work, 32-bit Ripple Carry Adder, 32-bit Carry Increment Adder and 32-bit Carry Skip Adder for different full adders are designed and implemented using Verilog HDL and results are obtained by executing Verilog code in Xilinx 14.5 ISE for the Spartan 3E family device with speed grade -5. On performing synthesis, it is evident that the LUTs are less for RCA. The LUTs increased by 38% and 59% for CIA and CSKA respectively. The slices (area) are increased by 37% and 54% for CIA and CSKA respectively when compared with the RCA. CSKA with FA1 has 36% less delay than the RCA with FA and hence provides the least delay. It is also evident that the delay produced by opting to FA2 results in decrease of the delay for RCA, CIA and CSKA. Hence for implementation of RCA, CIA and CSKA, the use of MUX based full adder as basic cell will improve the performance in terms of the delay provided. Further, this work can be extended in designing and comparing for different sized adders like 64-bit, 128 bit. The work can also be extended in designing and comparing other 32-bit adders like carry save adder, carry look ahead adder and carry select adder.

REFERENCES

[1] Prof. S.Murugeswari and Dr. S.Kaja Mohideen, “Design of area efficient and low power multiplier using multiplexer based full adder” in International Conference on Current Trends in Engineering and Technology, ICCET, Coimbatore, pp.388-392, 2014.
[2] Kokila Bharti Jaiswal et al., “Low Power Wallace Tree Multiplier Using Modified Full Adder” in International Conference on Signal Processing, Communication and Networking, ICSCN, Chennai, 2015, ISSN 78-1-4673-6823-0.
[3] Sreehari Veeramachaneni et al., “ Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors ” in International Conference on VLSI Design, VLSID, IEEE Computer Society, 2007, ISSN 0-7695-2762-0.
[4] B. Ram Kumar, Harish M Kittur, “Low power and area efficient carry select adder”, IEEE Trans. VLSI pp.371-375, vol.20, No.2, 2012.
[5] K.Bala Sindhuri, Prof. N .Uday Kumar, D.V.N. Bharathi, B. Tapasvi, “128-bit area efficient carry select adder”, vol.3, pp.48- 53,ISSN: 2321-9653,nov 2014.
[6] R.Uma et al., “Area, delay and power comparison of adder topologies” in International Journal of VLSI design & Communication Systems ,VLSICS, pp.153-168, Vol.3, No.1, February 2012.
[7] Maroju SaiKumar and Dr. P. Samundiswary, “Design and performance analysis of various adders using verilog” in International Journal of Computer Science and Mobile Computing, pp.128-138,Vol.2, Issue 9 , September 2013.
[8] B.Rajeswar Reddy et al., “Multi precision arithmetic adders” in International Conference on Computer Communication and Informatics, ICCCI, Coimbatore,2016.
[9] Jasbir Kaur and Lalit Sood, “Comparison between various types of adder topologies” in International Journal of Computer Science and technology, pp.62-66, Vol.6, Issue-1, 2015.
[10] Manchal Ahuja and Sakshi, “Design of Bypassing Multipier with Different Adders” in Universal Journal of Electrical and Electronic Engineering, UJEEE, Vol.2 Issue 5 pp. 217-221, 2014.
[11] Anand Kumar, Switching theory and logic design, 2nd ed. Delhi, PHI Learning Private limited, 2014.
[12] M. Morris Mano, Digital Logic and Computer Design, 3rd edition New Delhi, PHI Private Limited, 2007.
[13] Bhavani Koyada et al., “A Comparative Study on Adders”, IEEE, pp.2226-2230, 2017.
[14] Vishwa Shah, Urvisha Fatak, Jagruti Makwana, “Design and Performance Analysis of 32 Bit VLSI Hybrid adder”, International Conference on Trends in Electronics and Informatics (ICOEI), pp.1070-1075, 2019 IEEE.
[15] Aribam Balarampyari Devi, Manoj Kumar and Romesh Laishram, “Design and Implementation of an Improved Carry Increment Adder”.
[16] N Aruna Kumari, M Sai Srinivas, K Aravind, “Implementation of 64 Bit Arithmetic Adders”, International Journal of Recent Technology and Engineering (IJRTE), Volume-7, Issue-5S4, pp.437-444, February 2019.
[17] Neha Varshney, Greeshma Arya, “Design and Execution of Enhanced Carry Increment Adder using Han-Carlson and KoggeStone adder Technique”, International Conference on Electronics Communication and Aerospace Technology (ICECA), pp.1163-1170, 2019.
[18] S.Nagaraj, Dr. G.M.Sreerama Reddy, “Analysis of different Adders using CMOS, CPL and DPL logic”, 2017 IEEE.
[19] Ananthakrishnan, Anaswar Ajit, Arathi PV, Kiran Haridas, “FPGA Based Performance Comparison of Different Basic Adder Topologies with Parallel Processing Adder”, International Conference on Electronics Communication and Aerospace Technology (ICECA), pp.87-92, 2019.
[20] Anjana Bhardwaj, Swati Gupta, “Design and Comparative Analysis of Conventional Adder and Pipelined Adder”, International Journal of Engineering and Advanced Technology (IJEAT), Volume-3, Issue-2, pp.185-188, December 2013.