www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

PERFORMANCE AND ANALYSIS OF LOW POWER, AREAEFFICIENT AND HIGH SPEED CARRYFAST ADDER

[1*].M.AntoBennet,[2]S.Sankaranarayanan[3]V.BanuPriya[4]PJayaPretheena[5]S.Yamini[6] S.Supriya

[1] Faculty of Electronics and Communication Department, vel tech, Chennai, India.

[2] UG Students of Electronics and Communication Department, vel tech , Chennai, India.

Email: bennetmab@gmail.com

Submitted: May 27, 2017 Accepted: June 15, 2017 Published: Sep 1, 2017

Abstract- Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 8, 16,32,and 64-bit square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power. Binary to Excess-1 Converter (BEC) instead of RCA with the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder structure. The delay and area evaluation methodology of the basic adder blocks. The SQRT CSLA has been chosen for comparison with the proposed design as it has a more balanced delay, and requires lower power and area. Reducing the area and power consumption in the CSLA. Efficient gate-level modification to significantly reduce the area and powerof the CSLA.

Index terms*: *Carry Select Adder (CSLA),Binary to Excess-1 Converter (BEC),Carry-propagation adder (CPA)

I. INTRODUCTION

High-speed data path logic systems are one of the most substantial areas of research in VLSI system design. The speed of addition is limited by the time required to propagate a carry through the adder in digital adders. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position. The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry. Design of high speed data path logic systems are one of the most substantial research area in VLSI system design. High-speed addition and multiplication has always been a fundamental requirement of high-performance processors and systems. The major speed limitation in any adder is in the production of carries and many authors have considered the addition problem. The basic idea of the proposed work is using n-bit binary to excess-1 code converters (BEC) to improve the speed of addition. The detailed structure and function of BEC . This logic can be implemented with any type of adder to further improve the speed. The proposed 16, 32 and 64-bit adders are compared in this paper with the conventional fast adders such as carry save adder (CSA) and carry look ahead adder (CLA). This paper has realized the improved performance of the CSA with BEC logic through custom design and layout The final stage CPA constitutes a dominant component of the delay in the parallel multiplier. Signals from the multiplier partial products summation tree do not arrive at the final CPA at the same time. This is due to the fact that the number of partial-product bits is larger in the middle of the multiplier tree. Due to un-even arrival time of the input signals to the final CPA, the selection of the ASIC Implementation of Modified Faster Carry Save Adder 54 final adder is an important work in parallel multipliers . Therefore decrease in carry propagation delay will result in major enhancement of the speed of the adder and multiplier. An overview of the 4-bit binary to excess-1 logic is provided.It deals with the proposed modified carry save adder (MCSA) architecture. Among the myriad of aggressive techniques, carry select adder (CSL) has been an eminent technique in the space-time tug-of-war of CPA design. It exhibits the advantage of logarithmic gate depth as in any structure of the distant-carry adder family. Conventionally, CSL is implemented with dual ripple-carry adder (RCA) with the carry-in of 0 and 1, respectively. Depending on the configuration of block length, CSL is further classified as either linear or square root. The basic idea of CSL is anticipatory parallel computation. Although it can achieve high speed by not waiting for the carry-in from previous sub-block before computation can begin, they consume more power due to doubling the amount of circuitry needed to do the parallel addition of which half of the speculative computations will be redundant.. Digital Adders are the core block of DSP processors. The final carry propagation adder (CPA) structure of many adders constitutes high carry propagation delay and this delay reduces the overall performance of the DSP processor. This paper proposes a simple and efficient approach to reduce the maximum delay of carry propagation in the final stage. Based on this approach a 16, 32 and 64-bit adder architecture has been developed and compared with conventional fast adder architectures. This work identifies the performance of proposed designs in terms of delay-area-power through custom design and layout in process technology. The result analysis shows that the proposed architectures have better performance in reduction of carry propagation delay than contemporary architectures. The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size of . When variable, the block size should have a delay, from addition inputs A and B to the carry out, equal to that of the multiplexer chain leading into it, so that the carry out is calculated just in time. The delay is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added, since that will yield an equal number of MUX delays. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder.

The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position. The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum in reference1. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin=0 and Cin =1, then the final sum and carry are selected by the multiplexers (MUX). The basic idea of this work is to use Binary to Excess- 1 converter (BEC) instead of RCA with Cin = 1 in the regular CSLA to achieve lower area and power consumption in ref 2–4. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. The details of the BEC logic are discussed. Carry-ripple adder (CRA) is the simplest approach.

However, the carry-look ahead adder (CLA) and its fast version, the parallel-prefix CLA, is the selected scheme for time-critical applications with a considerable cost in terms of silicon area and power dissipation. The CSA provides a compromise between a RCA and a CLA adder. Due to rapidly growing system-on-chip industry, not only the faster units but also smaller area and less power has become a major concern for designing very large scale integration (VLSI) circuits. Digital circuits make use of digital arithmetic's. Among various arithmetic operations, multiplication is one of the fundamental operation used and is being performed by an adder. There are many ways to build a multiplier each providing trade-off between delays and other characteristics, such as area and energy dissipation. However no design is considered as superior. Carry-Save based Multiplier is one of the promising techniques in terms of speed. It provides a compromise between ripple carry adder and carry look-ahead adder, but to a lesser extent at the cost of its area.

II. LITERATURE SURVEY

High-speed addition and multiplication has always been a fundamental requirement of high-performance processors and systems. The major speed limitation in any adder is in the production of carries and many authors have considered the addition problem. High-speed data path logic systems are one of the most substantial areas of research in VLSI system design. The speed of addition is limited by the time required to propagate a carry through the adder in digital adders.More power consumption[1]. Digital Adders are the core block of DSP processors. The final carry propagation adder (CPA) structure of many adders constitutes high carry propagation delay and this delay reduces the overall performance of the DSP processor. This paper proposes a simple and efficient approach to reduce the maximum delay of carry propagation in the final stage. Based on this approach a 16, 32 and 64-bit adder architecture has been developed and compared with conventional fast adder architectures. This work identifies the performance of proposed designs in terms of delay-area-power through custom design[2]. A carry-select adder can be implemented by using single ripple carry adder and an add-one circuit instead of using dual ripple-carry adders. This paper proposes a new add-one circuit using the first zero finding circuit and multiplexers to reduce the area and power with no speed penalty. For bit length n = 64, this new carry-select adder requires fewer transistors than the dual ripple-carry carry-select adder and fewer transistors than Chang's carry-select adder using single ripple carry adder[3,7].In this ,Carry-select method has deemed to be a good compromise between cost and performance in carry propagation adder design. However, conventional carry-select adder (CSL) is still area-consuming due to the dual ripple carry adder structure. The excessive area overhead makes CSL relatively unattractive but this has been circumvented by the use of add-one circuit introduced recently. In this paper, an area efficient square root CSL scheme based on a new first zero detection logic is proposed. The proposed CSL witnesses a notable power-delay and area-delay performance improvement by virtue of proper exploitation of logic structure and circuit technique[4,5,6].

III. PROPOSED SYSTEM

A structure of 4-bit BEC and the truth table is shown. How the goal of fast addition is achieved using BEC together with a multiplexer (mux) is described , one input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial product results in parallel and the muxes are used to select either BEC output or the direct inputs according to the control signal Cin.

The Boolean expressions of 4-bit BEC are listed below, (Note: functional symbols,

~ NOT, & AND, ^ XOR).

X0 = ~ B0 (1)

X1 = B0 ^ B1 (2)

X2 = B2 ^ (B0 & B1) (3)

X3 = B3 ^ (B0 & B1 & B2)

XOR GATE MODEL

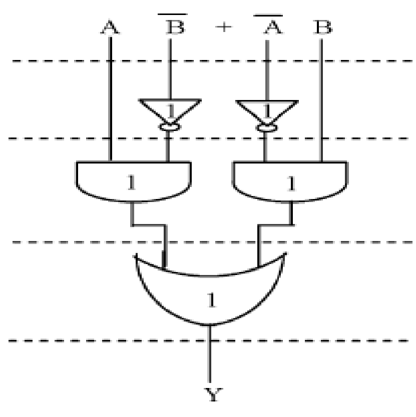


Fig - 1 XOR gate model

Today, one of the major challenges for high-performance microelectronic systems is the power dissipation, both static and dynamic. The circuit designer must, therefore, find an optimum between power and speed, instead of targeting them independently, and this is represented by the power-delay product, which represents the average energy dissipated for one switching event. Due to the rapidly growing mobile industry, not only faster arithmetic units but also smaller and lower power arithmetic units are demanded. However, it has been difficult to do well both in speed and in area. Ripple-carry adder (RCA) provides a compact design but suffers from a long delay time. Carry look ahead adder (CLA) gives a fast design but has a large area. Carry-select adder (CSA) is intermediate in regard to speed and area. Addition is by far the most fundamental arithmetic operation. It has been ranked the most extensively used operation among a set of real-time digital signal processing benchmarks from application-specific DSP to general purpose processors. Carry-propagation adder (CPA) is frequently part of the critical delay path limiting the overall system performance due to the inevitable carry propagation chain. The increase in the popularity of portable systems as well as the rapid growth of the power density in integrated circuits have made power dissipation one of the important design objectives, second only to performance shown in fig 1..

The carry select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The carry select adder partitions the adder into several groups, each of which performs two additions in parallel. Therefore, two copies of ripple-carry adder act as carry evaluation block per select stage. One copy evaluates the array chain assuming the block carry-in is zero, while the other assumes it to be one. The conventional n-bit CSA consists of one n/2-bit adder for the lower half of the bits and two n/2-bit adders for the upper half of the bits. Of the two latter adders, one performs the addition with the assumption that Cin=0, whereas the other does this with the assumption that Cin=1. Using a multiplexer and the value of carry out that is propagated from the adder for the n/2 least significant bits, the correct value of the most significant part of the addition can be selected. Although this technique has the drawback of increasing the area, it speeds up the addition operation.

A 16-bit carry select adder with a uniform block size of 4 can be created with three of these blocks and a 4-bit ripple carry adder. Since carry-in is known at the beginning of computation, a carry select block is not needed for the first four bits. The delay of this adder will be four full adder delays, plus three MUX delays. Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in. Since one ripple carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result. The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer.

The conventional carry select adder consists of k/2 bit adder for the lower half of the bits i.e. least significant bits and for the upper half i.e. most significant bits (MSB's) two k/2 bit adders. In MSB adders one adder assumes carry input as one for performing addition and another assumes carry input as zero. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. This technique of dividing adder in to stages increases the area utilization but addition operation fastens.

IV. METHODOLOGIES

The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different digital systems. It is sometimes necessary to use the output of one system as the input to another. A conversion circuit must be inserted between the two systems if each uses different codes for the same information. Thus, a code converter is a circuit that makes the two systems compatible even though each uses a different binary code shown in fig 2..

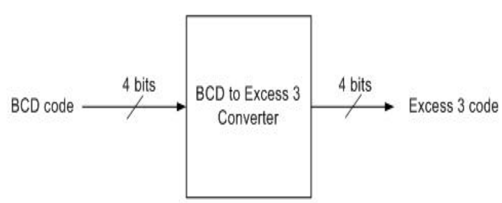


Fig.2 Block Diagram Of BCD To Excess 3 Code Converter

The bit combinations assigned to the BCD and excess-3 codes. Both BCD code and Excess 3 code use 4 bits to represent the numbers. But only 10 of 16 combinations are listed in the truth table. The rest 6 combinations not listed for the input variables are treated as don’t cares. Therefore, the corresponding output variables can be assigned as either 1 or 0, whichever gives asimpler circuit shown in table 1.

Table 1 TRUTH TABLE OF 4-BIT BINARY TO EXCESS-1

BINARY	EXCESS-1
0000	0001
0001	0010
0010	0011
0011	0100
0100	0101
0110	0110
0110	0111
0111	1000
1000	1001
1001	1010
1010	1011
1011	1100
1100	1101
1101	1110
1110	1111
1111	0000

4.1MODULE DESCRIPTION

The CSLA is used in many computational systemsto alleviate the problem of carry propagation delay byindependently generating multiple carries and thenselect a carry to generate the sum in reference1.However, the CSLA is not area efficient because ituses multiple pairs of Ripple Carry Adders (RCA) togenerate partial sum and carry by considering carryinputCin=0 and Cin =1, then the final sum and carryare selected by the multiplexers (mux).The basic idea of this work is to use Binary to Excess-1 converter (BEC) instead of RCA with Cin = 1 in theregular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number oflogic gates than the n-bit Full Adder (FA) structure. However, the carry-look ahead adder (CLA) and its fast version, the parallel-prefix CLA, is the selected scheme for time-critical applications with a considerable cost in terms of silicon area and power dissipation. The CSA provides a compromise between a RCA and a CLA adder. Hybrid adders combine elements of different approaches to obtain adders with a higher performance, reduced area and low power consumption.

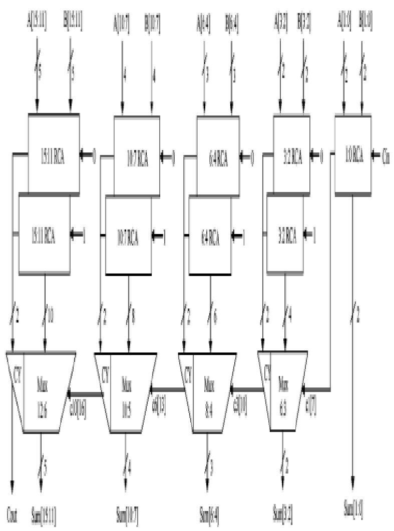


Fig 3 REGULAR 16 BIT SQRT CSLA

In this module Half Adder is a digital combinational circuit that is used for the addition of two bits and provides an output in the form of a sum bit and a carry bit. The logical functional equations that relate the outputs S and C of a half adder circuit to the input bits are Sum(S) = A XOR B, Carry(C) = A.B. Thus a half adder circuit can easily be synthesized by using 1 X-OR gate and 1 AND gate. Since a half adder circuit can only be used to add two bits, it becomes obsolete in case of multi-bit addition in practical applicationsshown in fig 3..

4.2MODULE 2 MODULE

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with in the regular CSLA to achieve lower area and power consumption .The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

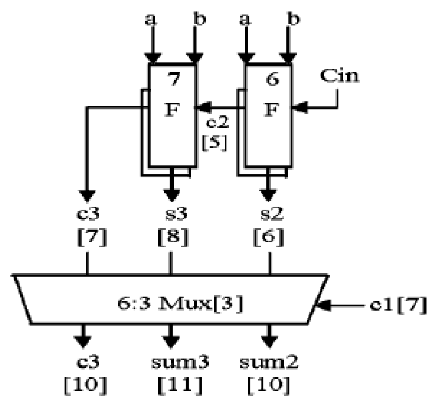


Fig 4:GROUP 2 MODULE DIAGRAM

The main idea of this work is to use BEC instead of the RCA with in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an (n+1)-bit BEC is required. The group2 has two sets of 2-b RCA. Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA's. The one set of 2-b RCA in group2 has 2 FA for Cin=1 and the other set has 1 FA and 1 HA for Cin=0 shown in fig 4.

4.3GROUP 3 MODULE

The group3 has one 2-b RCA which has 1 FA and 1 HA for Cin=0. Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values.The arrival time of selection input C3[time(t)=1] of 8:4 mux is earlier than the S6 and C6 and later than the S4. Thus, the sum6 and final C6 (output from mux) are depending on S6 and mux and partial C6(input to mux) and mux, respectively. The sum4 depends on C3 and mux. In this module, We applied three input signal as mux_in and m. we obtain the output (specific signal) from given input signal. If A greater than B is equal to 1 will get mux_in signal. If A lesser than B we get m signal. Otherwise we get 0(8 bit signal) shown in fig 5.

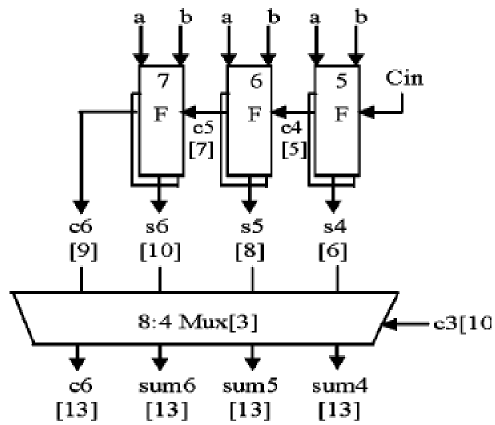


Fig 5 GROUP 3 MODULE DIAGRAM

A data selector, more commonly called a **Multiplexer,** shortened to "Mux" or "MPX", are combinational logic switching devices that operate like a very fast acting multiple position rotary switch. They connect or control, multiple input lines called "channels" consisting of either 2, 4, 8 or 16 individual inputs, one at a time to an output. Then the job of a multiplexer is to allow multiple signals to *share* a single common output. For example, a single 8-channel multiplexer would connect one of its eight inputs to the single data output. Multiplexers are used as one method of reducing the number of logic gates required in a circuit or when a single data line is required to carry two or more different digital signals.

GROUP 4 MODULE

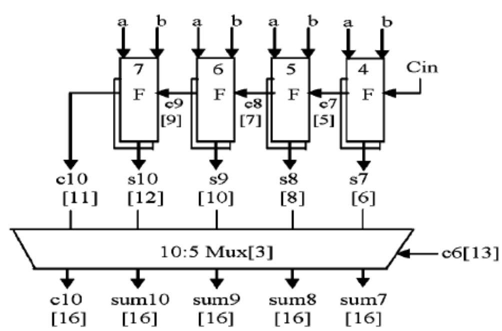


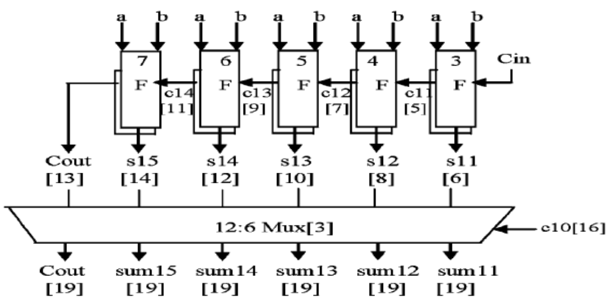
Fig 6 GROUP 4 MODULE DIAGRAM

The group4 has one 2-b RCA which has 1 FA and 1 HA for Cin=0. Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values.The arrival time of selection input C6[time(t)=16] of 10:5 mux is earlier than the S8 and C10 and later than the S7. Thus, thesum8 and final C10 (output from mux) are depending on S8 and mux and partial C10(input to mux) and mux, respectively. The sum7 depends on C6 and MUXshown in fig 6.

4.4GROUP 5 MODULE

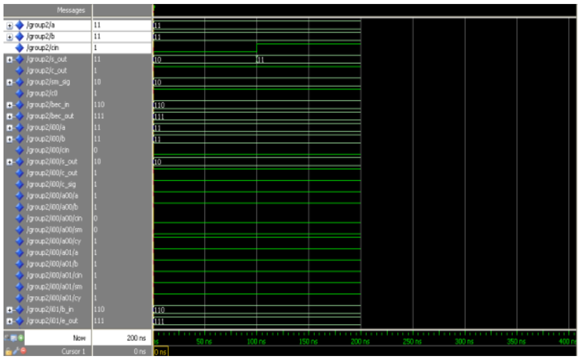
The group5 has one 2-b RCA which has 1 FA and 1 HA for Cin=0. Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values . The arrival time of selection input C10[time(t)=19] of 12:6 mux is earlier than the S10 and Cout and later than the S11. Thus, the sum10 and final Cout (output from mux) are depending on S10 and mux and partial Cout(input to mux) and mux, respectively.

Fig 7 GROUP 5 MODULE DIAGRAM



A standard 8-bit ripple-carry adder built as a cascade from eight 1-bit full-adders. The input switches or use the following bind keys: ('c') for carry-in, for A0..A7 and B0..B7. To demonstrate the typical behavior of the ripple-carry adder, very large gate-delays are used for he gates inside the 1-bit adders - resulting in an addition time of about 0.6 seconds per adder.Note that each stage of the adder has to wait until the previous stage has calculated and propagates its carry output signal. Obviously, the longest delay results for operands like A = 0b0000000, B=0b11111111 or A=0b01010101 and B=0b10101010.Therefore, the total delay of a ripple-carry adder is proportional to the number of bits. Faster adders are often required for bit widths of 16 or greater shown in fig 7.

V. EXPERIMENTAL RESULTS:



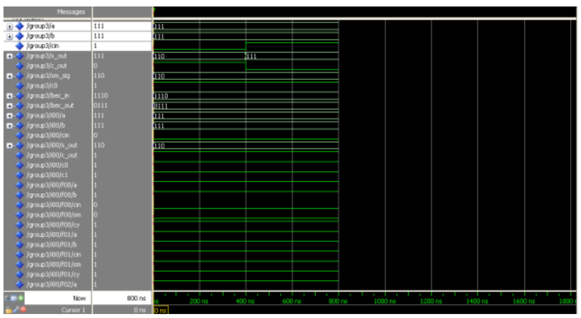
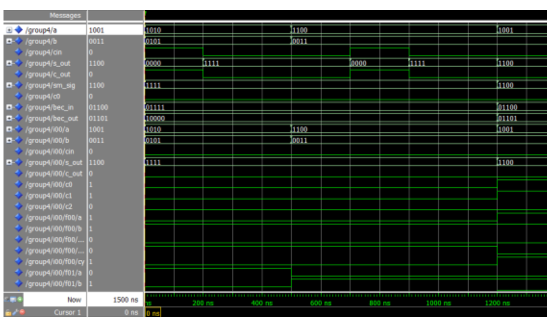


Fig8:GROUP8MODULE **Fig 9:GROUP 3 MODULE**



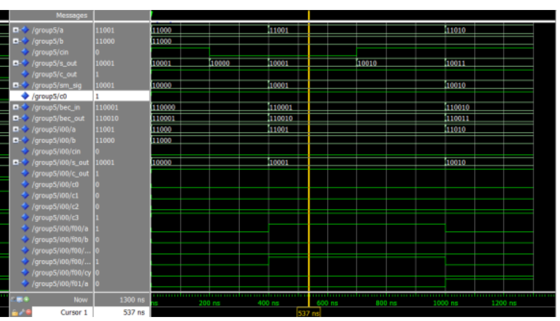


Fig10:GROUP4MODULE **Fig 11:GROUP 5 MODULE**

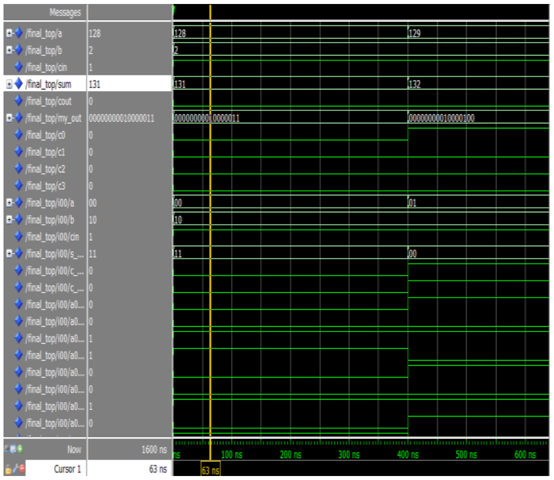


Fig 12:FINAL TOP OUTPUT

The implemented design in this work has been simulated using Verilog -HDL. The adders (of various size 16, 32, 64 and 128) are designed and simulatedwith Modelsim. All the V files (Regular and Modified) are also simulated in Modelsim and corresponding results are compared.After simulation the different size codes are synthesized using Xilinx ISE 9.1i. The simulated V files are imported into the synthesized tool and resultantvalues of delay and area are noted. The synthesized reports include area and delay values for different sized adders. The similar design flow is followed for both the regular and modified SQRT CSLA of different sizes are shown in fig 8-12

VI. CONCLUSION

All the adders designed are 32-bits wide. CSAS 5 stage consists of 5 stages with each block from LSB block to MSB blocks are [6-6-5-5-10] bits wide. These adders are faster than ripple carry adders but slower than carry select adders. All the adders are designed using VHDL (Very High Speed Integration Hardware Description Language), Xilinx Project Navigator 9.1i is used as a synthesis tool and ModelSim XE III 6.2g for simulation. FPGA Spartan3 is used for implementing the designs Gate count reduction is a sign of area reduction. Gate count of csas 2 stage is 11 less than carry select, 80 less than 3 stage carry select and 110 than 4 stage carry select adder. explore in this work is to design the adder in a way to reduce the delay as the area and power reduces. Wherever there is need of smaller area and low power consumption, while some increase in delay is tolerated, such designs can be used. These adders are faster than RCA and slower than CSA.

REFERENCES

[1] Aizat Azmi, Ahmad Amsyar Azman, Sallehuddin Ibrahim, and Mohd Amri Md Yunus, "Techniques In Advancing The Capabilities Of Various Nitrate Detection Methods: A Review", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 2, June 2017, pp. 223-261.

[2] Tsugunosuke Sakai, Haruya Tamaki, Yosuke Ota, Ryohei Egusa, Shigenori Inagaki, Fusako Kusunoki, Masanori Sugimoto, Hiroshi Mizoguchi, "Eda-Based Estimation Of Visual Attention By Observation Of Eye Blink Frequency", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 2, June 2017, pp. 296-307.

[3] Ismail Ben Abdallah, Yassine Bouteraa, and Chokri Rekik , "Design And Development Of 3d Printed Myoelctric Robotic Exoskeleton For Hand Rehabilitation", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 2, June 2017, pp. 341-366.

[4] S. H. Teay, C. Batunlu and A. Albarbar, "Smart Sensing System For Enhanceing The Reliability Of Power Electronic Devices Used In Wind Turbines", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 2, June 2017, pp. 407- 424

[5] SCihan Gercek, Djilali Kourtiche, Mustapha Nadi, Isabelle Magne, Pierre Schmitt, Martine Souques and Patrice Roth, "An In Vitro Cost-Effective Test Bench For Active Cardiac Implants, Reproducing Human Exposure To Electric Fields 50/60 Hz", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 1, March 2017, pp. 1- 17

[6] P. Visconti, P. Primiceri, R. de Fazio and A. Lay Ekuakille, "A Solar-Powered White Led-Based Uv-Vis Spectrophotometric System Managed By Pc For Air Pollution Detection In Faraway And Unfriendly Locations", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 1, March 2017, pp. 18- 49

[7] Samarendra Nath Sur, Rabindranath Bera and Bansibadan Maji, "Feedback Equalizer For Vehicular Channel", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 1, March 2017, pp. 50- 68

[8] Yen-Hong A. Chen, Kai-Jan Lin and Yu-Chu M. Li, "Assessment To Effectiveness Of The New Early Streamer Emission Lightning Protection System", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 1, March 2017, pp. 108- 123

[9] Iman Heidarpour Shahrezaei, Morteza Kazerooni and Mohsen Fallah, "A Total Quality Assessment Solution For Synthetic Aperture Radar Nlfm Waveform Generation And Evaluation In A Complex Random Media", International Journal on Smart Sensing and Intelligent Systems., VOL. 10, NO. 1, March 2017, pp. 174- 198

[10] P. Visconti ,R.Ferri, M.Pucciarelli and E.Venere, "Development And Characterization Of A Solar-Based Energy Harvesting And Power Management System For A Wsn Node Applied To Optimized Goods Transport And Storage", International Journal on Smart Sensing and Intelligent Systems., VOL. 9, NO. 4, December 2016 , pp. 1637- 1667

[11] YoumeiSong,Jianbo Li, Chenglong Li, Fushu Wang, "Social Popularity Based Routing In Delay Tolerant Networks", International Journal on Smart Sensing and Intelligent Systems., VOL. 9, NO. 4, December 2016 , pp. 1687- 1709

[12] Seifeddine Ben Warrad and OlfaBoubaker, "Full Order Unknown Inputs Observer For Multiple Time-Delay Systems", International Journal on Smart Sensing and Intelligent Systems., VOL. 9, NO. 4, December 2016 , pp. 1750- 1775

[13] Rajesh, M., and J. M. Gnanasekar. "Path observation-based physical routing protocol for wireless ad hoc networks." International Journal of Wireless and Mobile Computing 11.3 (2016): 244-257.

[14]. Rajesh, M., and J. M. Gnanasekar. "Congestion control in heterogeneous wireless ad hoc network using FRCC." Australian Journal of Basic and Applied Sciences 9.7 (2015): 698-702.

[15]. Rajesh, M., and J. M. Gnanasekar. "GCCover Heterogeneous Wireless Ad hoc Networks." Journal of Chemical and Pharmaceutical Sciences (2015): 195-200.

[16]. Rajesh, M., and J. M. Gnanasekar. "CONGESTION CONTROL USING AODV PROTOCOL SCHEME FOR WIRELESS AD-HOC NETWORK." Advances in Computer Science and Engineering 16.1/2 (2016): 19.

[17]. Rajesh, M., and J. M. Gnanasekar. "An optimized congestion control and error management system for OCCEM." International Journal of Advanced Research in IT and Engineering 4.4 (2015): 1-10.

[18]. Rajesh, M., and J. M. Gnanasekar. "Constructing Well-Organized Wireless Sensor Networks with Low-Level Identification." World Engineering & Applied Sciences Journal 7.1 (2016).

[19] L. Jamal, M. Shamsujjoha, and H. M. Hasan 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.

[20] S. N. Mahammad and K. Veezhinathan, "Constructing online testable circuits using reversible logic," IEEE Transactions on Instrumentation and Measurement, vol. 59, pp. 101–109, 2010.

[21] W. N. N. Hung, X. Song, G. Yang, J. Yang, and M. A. Perkowski, "Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 25, no. 9, pp. 1652–1663, 2006.

[22] F. Sharmin, M. M. A. Polash, M. Shamsujjoha, L. Jamal, and H. M. Hasan Babu, "Design of a compact reversible random access memory," in 4th IEEE International Conference on Computer Science and Information Technology, vol. 10, june 2011, pp. 103–107.

[23] Dr. AntoBennet, M, Sankar Babu G, Suresh R, Mohammed Sulaiman S, Sheriff M, Janakiraman G ,Natarajan S, "Design & Testing of Tcam Faults Using T_H Algorithm", Middle-East Journal of Scientific Research 23(08): 1921-1929, August 2015 .

[24] Dr. AntoBennet, M "Power Optimization Techniques for sequential elements using pulse triggered flipflops", International Journal of Computer & Modern Technology , Issue 01 ,Volume01 ,pp 29-40, June 2015.