Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 2, 102-106 (2019)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications

Samraj Daphni*, Kasinadar Sundari Vijula Grace

Electronics and Communication Engineering, Noorul Islam Centre for Higher Education, 629180, India

ARTICLE INFO

Article history:
Received: 26 January, 2019
Accepted: 27 February, 2019
Online: 15 March, 2019

Keywords:
Parallel Prefix Adder
Ladner Fischer Adder
Power Gating Technique
Carry Generation Stage
Carry Look ahead Adder

ABSTRACT

The basic processes like addition, subtraction can be done using various types of binary adders with dissimilar addition times (delay), area and power consumption in any digital processing applications. To minimize the Power Delay Product (PDP) of Digital Signal Processing (DSP) processors is necessary for high performance in Very Large Scale Integration (VLSI) applications. In this paper, a 32–bit various Parallel Prefix adders design is proposed and compared the performance results on the aspects of area, delay and power. Implementation (Simulation and Synthesis) results really achieve significant improvement in power and power-delay product when compared with the previous bit adders which is used in processors. To reduce the power, here apply the energy recovery logic like power gating technique for all three adders. All the simulations and synthesis results can be noted using Xilinx ISE 14.2i tool.

1. Introduction

Generally, the basic processes such as addition, subtraction, division that can be done by using different types of binary adders in any digital based processors and control systems [1]. The adder performance which is used in the device is only measured the high speed and accuracy of a processor or system. Previously the processors are used 32 bit carry adders like Ripple Carry Adder (RCA), Carry Propagate Adder (CPA), and Carry Look ahead Adders (CLA) with different addition times (delay), area and power consumption [2]. How fast the carry reaches for every single bit position, from which the delay of any binary adder is calculated. Henceforth, the carry chain which generates the carry bit is the major challenge in binary adder design. But the above existing 32 bit basic carry adders having high delay value in higher order bits because each level of adder has to wait for the previous carry result [3]. Due to the above problem of 32 bit basic existing carry adders, in today's world of technology, PPA is well suitable designed adder for high speed addition process with less delay in VLSI technology [4]. Also the PPA is one of the most popular designs and provides good negotiation amongst area, speed and power [5]. The low order PPA is designed at earlier like 8 bit and 16 bit. This paper is designed as follows, in second section: describes briefly about PPA, in third section: explains the design of 32 bit proposed Parallel Prefix (Kogge Stone Adder, Brent kung Adder, Ladner Fischer Adder), in fourth section: it expresses the simulation results (waveform and reports) of 32 bit PPA which we have designed in the previous section with the performance aspects (delay, area and power). The last section is concluded that Kogge Stone Adder performance is best among other adders with low power and less delay from the above analyzed results of 32 bit PPA.

2. Parallel Pefix Adder

Now a days, to avoid the higher delay problem of existing carry adders the PPA is used which is simply the modified design form of CLA. The Prefix adders can be designed in many different ways based on the different requirements and the production of carries [6]. Recently, use the tree structure form of adders to raise the speed of addition function in any kind of processors. PPA are fastest adders with tree structure based and used for high performance arithmetic processes in successive industries and DSP laboratories [7].

The PPA's are also called as logarithmic delay adders because the delay value is established using logarithmic functions [8]. Addition in PPA can be processed using three main actions such as Pre-computation (P and G signal generation), Prefix-computation (carry signals group generation), Post-computation (Sum signal generation) [9].

2.1. Pre – Computation

In the pre-processing stage, propagate functions and generate functions are calculated depends upon the given input signals [10]. The propagate functions are carried out by the equation (1).

*Samraj Daphni, W/O. Mr. D. Thavasumony, North Midalakkadu, Palappallam Post, Kanyakumari District – 629159, Tamil Nadu, India, +91 8675857541 & daphnithavasumony@gmail.com

$$P [j] = X [j] \oplus Y [j] \qquad (1)$$

It is stated that X and Y are the input signals that composed by XOR logic gate. The generate functions are carried out by the equation (2).

$$G [j] = X [j] \bullet B [j] \qquad (2)$$

It is stated that X and Y are the input signals that composed by AND logic gate. Since the above equations (1) and (2) are done in parallel, it does not increasing a significant calculation of area consumption and delay fully depends upon the bit size which is desired at the input [11].

2.2. Prefix – Computation

This prefix computation stage, calculates the carry signal groups directly, which uses the input and values which measured from the first stage. Carry signals generation uses the more than two inputs for which the delay is automatically increased in this process [12]. The carry propagation function and carry generation function [13] is measured by the equations (3) and (4).

$$\text{Carry } P = P [j] \bullet P [j + 1] \qquad (3)$$

$$\text{Carry } G = (G [j] \bullet P [j + 1]) + G [j + 1] \qquad (4)$$

2.3. Post – Computation

In this stage, the sum result is generated by an Ex-OR operation that uses the values of carry generation stage (prefix-computation). The last sum operation is calculated by the equation (5)

$$S [j + 1] = P [j + 1] \oplus C [j] \qquad (5)$$

Where C is the last carry signal and P is the propagate function [14].

3. Design and analysis of proposed 32 bit various PPA

To overcome high delay problem of existing carry adders this work proposed the design of 32-bit various PPA for less delay and low power VLSI applications. This proposed system consists of two modules: The first module is to design of 32 bit PPA like KSA, BKA, and LFA. The second module is to analyses the performance comparison of PPA on the basis of area, delay and power.

In this section, analyse the different technologies of adders to design in the form of parallel prefix, apart from the RCA topology, such as Kogge-Stone, Brent-Kung, and Ladner-Fisher PPA. The important aim is to examine the trade-off between area consumption delay and power consumption in the particular PPA depends upon the design performance. All the designs are using a power gating technique to reduce the power consumption [15].

3.1. Kogge Stone Adder (KSA)

Normally, the KSA attains the key role with fast addition operation and it reflects like prefix form of Carry Look ahead Adder (CLA). Also this type of PPA entirely decreases the delay time in design to generate the carry signals [16]. Henceforth this KSA is popularly used in DSP (Digital Signal Processing) laboratory and Control system industries for fast arithmetic function.

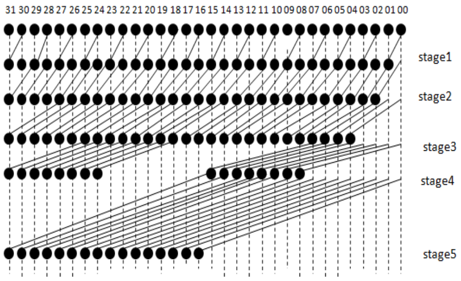


Figure 1: 32 bit Kogge Stone Adder design

The structure of 32-bit KSA design is exposed in Figure 1. This design can be divided into 5 stages [17]. The calculation of Propagate and Generate signals using full adders with carry input that process included in first and second stage. The generation of carry signals which used the values of Propagate and Generate that process included in third and fourth stage. The calculation of sum bits based on the P and carry generation values that is included in the fifth stage [18]. This 32 bit design of KSA is coded by VHDL and viewed the test bench waveform and analyzed the performance and noted the results.

3.2. Brent Kung Adder (BKA)

The BKA calculates the prefixes based on the bit groups. Initially calculate the prefixes values for 2 bit groups. These 2 bit prefix values are used to find the prefix values for the 4 bit groups, that are used to calculate the prefix values for 8 bit groups and so on [19]. Then these prefixes values are used to measure the carry out of the particular bit stage. These carries will be used along with the Group Propagate of the next stage to calculate the Sum bit of that stage [20]. Brent Kung Tree will be using (2log2N-1) stages for any bit design.

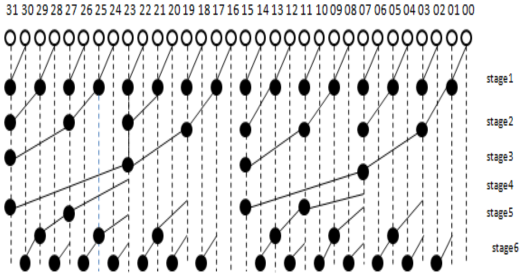


Figure 2: 32 bit Brent Kung Adder design

The structure of 32-bit BKA design is given in Figure 2. Hence the designing of 32- bit adder takes the number of stages will be 9. The fan-out for every bit stage is limited to 2. The above diagram shows the fan out being reduced and the loading on the advance stages being reduced [21]. This 32 bit design of BKA is coded by VHDL and viewed the test bench waveform and analysed the performance and noted the results.

3.3. Ladner Fischer Adder (LFA)

The LFA tree structures are a family of tree networks between Brent Kung and Sklansky tree. It is very close like to Sklansky PPA, but it calculates the prefix values for odd number bits after that again uses another stage which ripple into the even locations [22]. At higher order bits, to get improved in speeds the cells must still be properly sized or grouped.

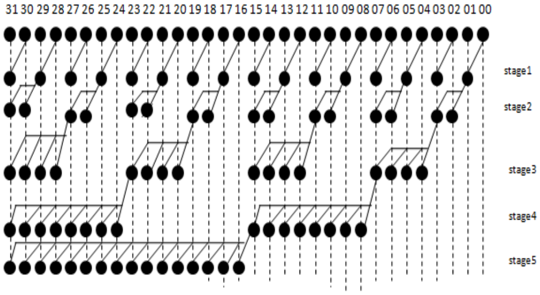
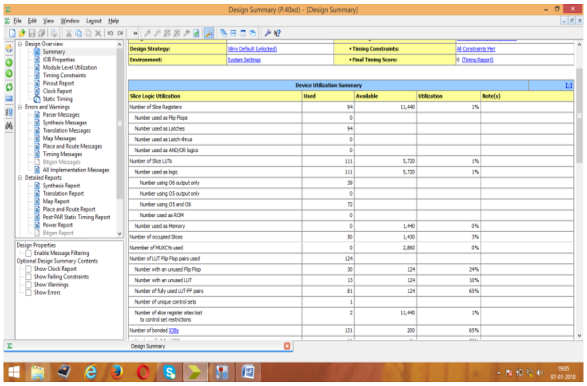


Figure 3: 32 bit Ladner Fischer Adder design

The structure of 32-bit LFA design is exposed in Figure 3. The Ladner Fischer adder is used for high performance arithmetic operation with complicate designs. The LFA consists of black cells and gray cells with 5 stages for 32 bit design. Each black cell encloses only one OR logic gate and two AND logic gates. Each gray cell contains only one AND logic gate [23]. This 32 bit design of LFA is coded by VHDL and viewed the test bench waveform and analysed the performance and noted the results.

4. Simulation results of proposed 32 bit PPA

In this simulation section, took all three types of 32 bit parallel prefix adders (KS, BK, LF) that are discussed above. All the PPA's are designed on VHDL (Very high speed Hardware Description Language) / Verilog project navigator 14.2i is used for synthesis (Xilinx version) [24]. Simulation results are verified on the basis of area, power and delay. In addition to that the waveforms and the comparison results for all three parallel prefix adders are given.

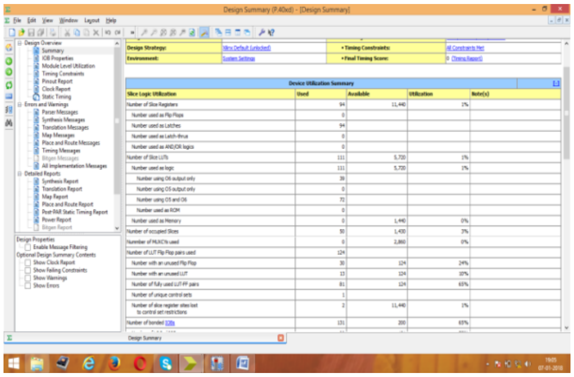

Design Summary (P.40xd) - [Design Summary]

Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	94	11,440	1%	
Number used as Flip Flops	0			
Number used as Latches	94			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	111	5,720	1%	
Number used as logic	111	5,720	1%	
Number using O6 output only	39			
Number using O5 output only	0			
Number using O5 and O6	72			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number of occupied Slices	50	1,430	3%	
Nummber of MUXCYs used	0	2,860	0%	
Number of LUT Flip Flop pairs used	124			
Number with an unused Flip Flop	30	124	24%	
Number with an unused LUT	13	124	10%	
Number of fully used LUT-FF pairs	81	124	65%	
Number of unique control sets	1			
Number of slice register sites lost to control set restrictions	2	11,440	1%	
Number of bonded IOBs	131	200	65%	

Figure 4: Area Consumption report for KSA


Design Summary (P.40xd) - [Design Summary]

Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	94	11,440	1%	
Number used as Flip Flops	0			
Number used as Latches	94			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	111	5,720	1%	
Number used as logic	111	5,720	1%	
Number using O6 output only	39			
Number using O5 output only	0			
Number using O5 and O6	72			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number of occupied Slices	50	1,430	3%	
Nummber of MUXCYs used	0	2,860	0%	
Number of LUT Flip Flop pairs used	124			
Number with an unused Flip Flop	30	124	24%	
Number with an unused LUT	13	124	10%	
Number of fully used LUT-FF pairs	81	124	65%	
Number of unique control sets	1			
Number of slice register sites lost to control set restrictions	2	11,440	1%	
Number of bonded IOBs	131	200	65%	

Figure 5: Power Analyzer report for KSA

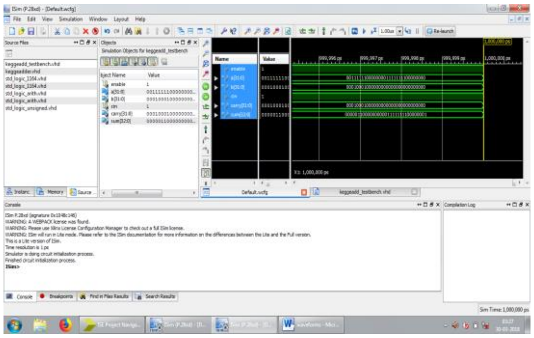
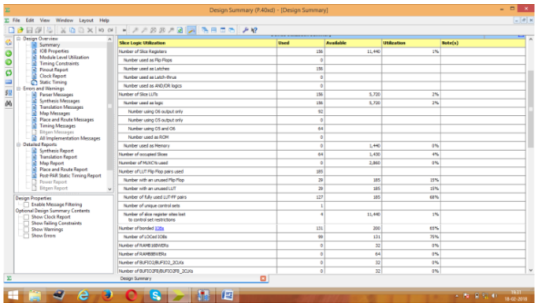
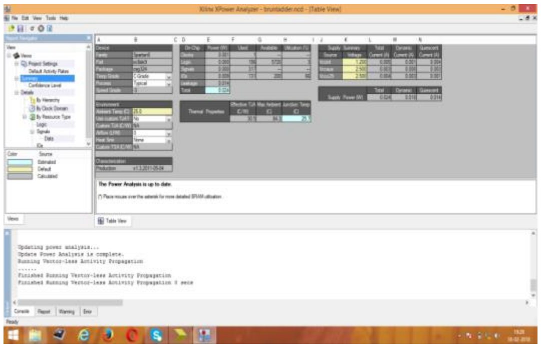

```
ISim P.28xd (signature 0x1048c146)
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 6: Test bench waveform results of KSA


Design Summary (P.40xd) - [Design Summary]

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	156	11,440	1%	
Number used as Flip Flops	0			
Number used as Latches	156			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	156	5,720	2%	
Number used as logic	156	5,720	2%	
Number using O6 output only	92			
Number using O5 output only	0			
Number using O5 and O6	64			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number of occupied Slices	64	1,430	4%	
Nummber of MUXCYs used	0	2,860	0%	
Number of LUT Flip Flop pairs used	185			
Number with an unused Flip Flop	29	185	15%	
Number with an unused LUT	29	185	15%	
Number of fully used LUT-FF pairs	127	185	68%	
Number of unique control sets	1			
Number of slice register sites lost to control set restrictions	4	11,440	1%	
Number of bonded IOBs	131	200	65%	
Number of LOCed IOBs	99	131	75%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	

Figure 7: Area Consumption report for BKA


Xilinx XPower Analyzer - brunta dder.ncd - [Table View]

The Power Analysis is up to date.

```
Updating power analysis...
Update Power Analysis is complete.
Running Vector-less Activity Propagation
......
Finished Running Vector-less Activity Propagation
Finished Running Vector-less Activity Propagation 0 secs
```

Figure 8: Power Analyzer report for KSA

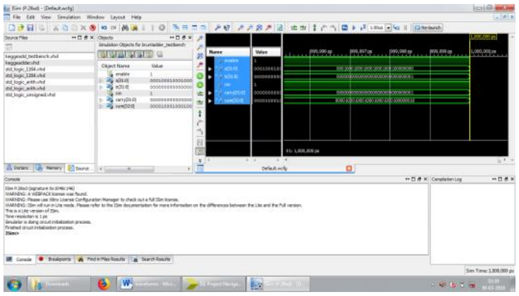
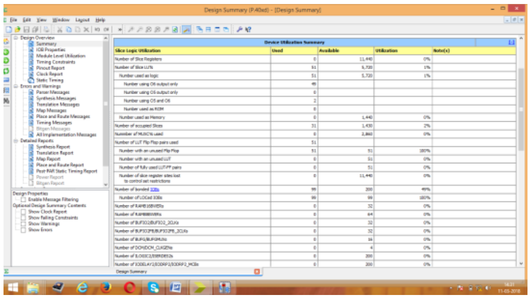

```
ISim P.28xd (signature 0x1048c146)
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

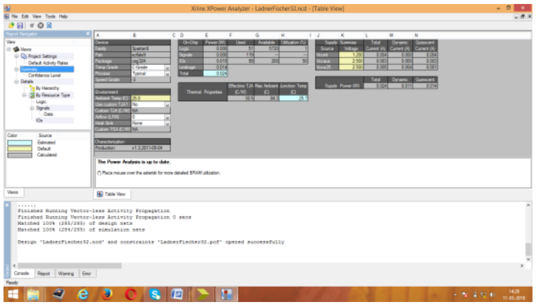
Figure 9: Test bench waveform results of BKA


Design Summary (P.40xd) - [Design Summary]

Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	11,440	0%	
Number of Slice LUTs	51	5,720	1%	
Number used as logic	51	5,720	1%	
Number using O6 output only	49			
Number using O5 output only	0			
Number using O5 and O6	2			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number of occupied Slices	31	1,430	2%	
Nummber of MUXCYs used	0	2,860	0%	
Number of LUT Flip Flop pairs used	51			
Number with an unused Flip Flop	51	51	100%	
Number with an unused LUT	0	51	0%	
Number of fully used LUT-FF pairs	0	51	0%	
Number of slice register sites lost to control set restrictions	0	11,440	0%	
Number of bonded IOBs	99	200	49%	
Number of LOCed IOBs	99	99	100%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	0	16	0%	
Number of DCM/DCM_CLKGENs	0	4	0%	
Number of ILOGIC2/ISERDES2s	0	200	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0%	

Figure 10: Area Consumption report for LFA

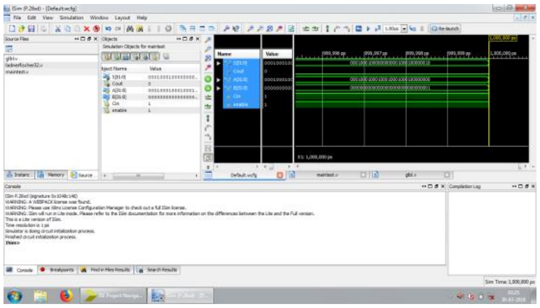


```
......
Finished Running Vector-less Activity Propagation
Finished Running Vector-less Activity Propagation 0 secs
Matched 100% (285/285) of design nets
Matched 100% (284/285) of simulation nets

Design 'LadnerFischer32.ncd' and constraints 'LadnerFischer32.pcf' opened successfully
```

The Power Analysis is up to date.

Figure 11: Power Analyzer report for KSA



```
ISim P.28xd (signature 0x1048c146)
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 12: Test bench waveform results of LFA

Area consumption, power analysis, and the test bench waveform of KSA are shown in figures 4, 5, and 6 respectively. Area consumption, power analysis, and the test bench waveform of BKA are shown in figures 7, 8, and 9 respectively. Also the area consumption, power analysis, and the test bench waveform of LFA are shown in figures 10, 11, and 12 respectively.

From the above figures, the comparison results of all three PPA on the aspects of area, delay and power is given in table I. From the analysis, LFA is better due to the less area consumption but the power utilization is more compared to other adders. Normally PPA's have less delay in any processors while doing addition. Accordance with low power application, KSA is more suitable due to less power utilization in any digital based processors.

Table 1: Comparison results of area, delay and power for various types of PPA

Adder Types	LUT's Used	Delay(ns)	Power (mW)
Kogge Stone Adder	189	6.483	21
Brent kung Adder	156	6.489	24
Ladner Fischer Adder	51	21.879	24

Conclusion

In this paper, an efficient 32 bit Parallel Prefix adders like KSA, BKA, LFA is designed. This proposed 32 bit adder addition operation offers a great advantage in reducing delay. For low power VLSI applications, also the designed adders are compared on the basis of power, area consumption, and delay. The synthesis results reveal that among the proposed adders, KSA is achieved some saving of power-delay product due to less power utilization. But the area delay product is little increased, compared to other adders due to high area consumption. For decreasing the complexity at all performance aspects, further optimization techniques can be achieved on the performance parameters that will be the future work of the paper.

References

[1] P. Patronik, S. J. Piestrak, "Design of RNS Reverse Converters with Constant Shifting to Residue Datapath Channels" Journal of Signal Processing Systems, 2017. https://doi.org / 10.1007/s11265-017-1238-6

[2] D. Kunjan, Shinde, S. Badiger, "Analysis and comparative study of 8-bit adder for embedded application" International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2015. https://doi.org /10.1109/iccicct.2015.7475290

[3] S. Gedam, Pravin Zode, and Pradnya Zode, "FPGA implementation of hybrid Han-Carlson adder" in 2nd International Conference on Devices Circuits and Systems (ICDCS), 2014. https://doi.org/10.1109/icdcsyst.2014.6926185

[4] H. Moqadasi, M. B. Ghazvani-Ghoushchi, "A new parallel prefix adder structure with efficient critical delay path and gradded bits efficiency in CMOS 90nm technology" in 23rd Iranian Conference on Electrical Engineering (ICEE), 2015. https://doi.org /10.1109/iraniancee.2015.7146426

[5] P. Kumar and M. Kiran, "Design of optimal fast adder" in International Conference on Advanced Computing and Communication Systems, 2013. https://doi.org /10.1109/icaccs.2013.6938692

[6] A. Zarandi, A. Molahosseini, M. Hosseinzadeh, S. Sorouri, S. Antão, and L. Sousa, "Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology, and Implementations" IEEE transactions on very large scale integration (VLSI) systems, 2014. https://doi.org/10.1109/tvlsi.2014.2305392

[7] Saxena and Pallavi, "Design of low power and high speed Carry Select Adder using Brent Kung adder" in International Conference on VLSI Systems Architecture Technology and Applications (VLSI-SATA), 2015. https://doi.org/10.1109/vlsi-sata.2015.7050465

[8] N. Shanil Mohamed, T. Y. Siby, "16-bit Velocious Fault Lenient Parallel Prefix Adder" in International Conference on Electronics, Communication and Computational Engineering (ICECCE), 2014. https://doi.org/10.1109/icecce.2014.7086612

[9] S. A. H. Ejtahed, M. B. Ghaznavi-Ghoushchi, "Design and Implementation of a Power and Area Optimized Reconfigurable Superset Parallel Prefix Adder" in 24th Iranian Conference on Electrical Engineering (ICEE), 2016. https://doi.org/10.1109/iraniancee.2016.7585787

[10] I. Marouf, M. M. Asad, A. Bakhuraibah and Q. Abu Al-Haija, "Cost Analysis Study of Variable Parallel Prefix Adders Using Altera Cyclone IV FPGA Kit" in International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2017. https://doi.org/10.1109/icecta.2017.8252011

[11] S. Arthireena, G. Shanmugavadivel, "Efficient Sign Detection using Parallel Prefix Adder" in International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), 2017. https://doi.org/10.1109/iceice.2017.8191852

[12] S. Daphni, K. S. Vijula Grace, "A review analysis of parallel prefix adders for better performance in VLSI applications" in Proceedings of 2017 IEEE International Conference on Circuits and Systems (ICCS), 2017. https://doi.org/10.1109/ICCS1.2017.8325971

[13] F. Liu, Q. Tan, G. Chen, "Formal proof of prefix adders" Mathematical and Computer Modelling, ELSEVIER, 2010. https://doi.org/10.1016/j.mcm.2010.02.008

[14] S. K. Yezerla, B. Rajendra Naik, "Design and Estimation of delay, power and area for Parallel prefix adders" in Proceedings of RAECS UIET Panjab University Chandigarh., 2014. https://doi.org/10.1109/raecs.2014.6799654

[15] A. E. Shapiro, F. Atallah, K. Kim, J. Jeong, J. Fischer, E. G. Friedman, "Adaptive power gating of 32-bit Kogge Stone adder" INTEGRATION, the VLSI journal, ELSEVIER, 2015. https://doi.org/10.1016/j.vlsi.2015.12.001

[16] M. Ozer, M. Eren Çelik, Y. Tukel, A. Bozbey, "Design of RSFQ wave pipelined Kogge–Stone Adder and developing custom compound gates" Cryogenics, ELSEVIER, 2014. https://doi.org/10.1016/j.cryogenics.2014.05.007

[17] P. Chaitanya kumari, R. Nagendra, "Design of 32 bit Parallel Prefix Adders" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), 2013. https://doi.org/10.9790/2834-610106

[18] C. Shilpa, K. D. Shinde, H. V. Nithin, "Design, Implementation and Comparative Analysis of Kogge Stone Adder using CMOS and GDI design: A VLSI Based Approach" in 8th International Conference on Computational Intelligence and Communication Networks, 2016. https://doi.org/10.1109/cicn.2016.117

[19] N. Poornima, V. S. Kanchana Bhaaskaran, "Area Efficient Hybrid Parallel Prefix Adders" in 2nd International Conference on Nano materials and Technologies (CNT), ELSEVIER, 2014. https://doi.org/10.1016/j.mspro.2015.06.069

[20] L. Hars, "Applications of Fast Truncated Multiplication in Cryptography" EURASIP Journal on Embedded Systems, 2007. https://doi.org/10.1155/2007/61721

[21] J. Thomas, R. Pushpangadan, S. Jinesh, "Comparative Study of Performance Vedic Multiplier on The Basis of Adders Used" IEEE International WIE Conference on Electrical and Compter Engineering (WIECON-ECE), 2015. https://doi.org/10.1109/wiecon-ece.2015.7443929

[22] A. Baliga, "Design of High Speed Adders Using CMOS and Transmission Gates in Submicron Technology: A Comparative Study" in Fourth International Conference on Emerging Trends in Engineering & Technology, Nov 2011. https://doi.org/ 10.1109/icetet.2011.25

[23] S. Baba Fariddin, E. Vargil Vijay, "Design of Efficient 16-Bit Parallel Prefix Ladner Fischer Adder" International Journal of Computer Applications, 2013. https://doi.org/ 10.5120/13943-1784

[24] M. Macedo, L. Soares, B. Silveira, C. M. Diniz, Eduardo A. C. da Costa, "Exploring the use of parallel prefix adder topologies into approximate adder circuits" in 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2017. https://doi.org/10.1109/icecs.2017.8292078