International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Volume 9, Issue 6 - June 2022 - Pages 1-9

ANALYSIS OF PARALLEL PREFIX ADDERS FOR CRYPTOGRAPHIC APPLICATION

G.Themozhi[1], K. Srinivasan,[2] S, Aasha Nandhini,[3] P, Radhakrishnan[4]
gthemozhivijayakumar@gmail.com[1], omsrivas@yahoo.co.in,[2] aasha.nandhu@gmail.com,[3]
AMET University,[1] Chennai, Tagore Engineering College,[2] Amrita School of Computing, Amrita Vishwa Vidyapeetham,[3] Tagore Engineering College[4], Chennai-127

***Abstract*— Parallel prefix adder provides better results when larger bit length data are used.This paper deals with the design of Parallel Prefix Adders (PPA) and comparison among parallel prefix adders. The carry-look ahead adder and tree structure of parallel prefix adders such as Kogge-Stone Adder, Brent Kung Adder, Han-Carlson adder, Knowles adder, Sklansky adder and Ladner-Fischer adder are modelled using Verilog code. Simulation results are obtained for the carry-look ahead adder and parallel prefix adders using Xilinx software. Performance comparisons of the above-mentioned adders are done and proved. Brent-Kung adder is the best suited for cryptographic applications. An encryption and Decryption algorithmic step for advanced encryption standard cryptographic application is obtained using Brent-Kung Adder. Verilog code is written for encryption and decryption and the simulation results are obtained.**

***Index Terms*— Carry-look ahead adder, Parallel prefix adder, LUT, Slices, IOB, Fan-out, Verilog, Xilinx and Model SIM.**

I. Introduction

Binary adders are one of the most essential logic elements within a digital system. In addition, binary adders are also helpful in units other than Arithmetic Logic Units (ALU), such as multipliers, dividers and memory addressing. Therefore, binary addition is essential. Any improvement in a binary addition can result in a performing Boost for any computing system and hence helps to improve the performance of the entire system. The major problem in the Binary addition is the carry chain. As the width the input operand increases, the length of the carry chain increases.Binary adders evolve from linear adders, which have a delay approximately proportional to the width of the adder. The carry-increment adder (CIA) and the Ling adder can enhance the carry chain. Uma (2012) designed and simulated, Carry look-ahead adder (CLA), CIA, carry skip adder (CSA) and carry bypass adder (CBA). It was concluded that the CLA and CIA are suitable for high performance in low power circuits. Carry select adder (CSEA) and carry save adder (CSVA) have less area utilization. RCA, CSA and CBA have least gate count and maximum delay. however, in Very Large Scale Integration (VLSI) digital systems, the most efficient way of offering binary addition involves utilizing parallel-prefix trees, this occurs because they have the regular structures that exhibit logarithmic delay. Anjaneyulu (2013) designed and simulated, RCA, kogge-stone adder (KSA), sparse kogge-stone adder (SKSA). The design was done using Verilog HDL, simulated using Xilinx software. Finally it was said that the speed of parallel prefix adder was higher than CLA. Babulu (2012) designed & simulated KSA, SKSA, Brent-kung adder (BKA) and compared the performance of above three adders RCA and CSA. It was computed that the Parallel prefix adder (PPA) are not as effective as the simple ripple carry adder at low to moderate bit widths. Young-moon choi (1998) implemented the matrix representation for gate sizing which calculate the delay and the total transistor width of the carry propagation graph of adders. The matrix representation is successfully applied to gate sizing of the adders. Renukuntla Kiran (2012) both measured and simulation results from this study have shown that carry tree adders are not as effective as the simple ripple-carry adder at low to moderate bit widths. However, carry-tree adders eventually surpass the performance of the linear adder designs at high bit-widths, expected to be in the 128 to 256 bit range. The explored possible FPGA architectures that could implement a "fast-tree chain" and investigate the possible tradeoffs involved. The testability and possible fault tolerant features of the spanning tree adder are not analyzed by this paper..

Parallel-prefix adders compute addition in two steps. Initial step is to obtain the carry at each bit. In the second step is to compute the sum bit based on the previous bit. Unfortunately, prefix trees are algorithmically slower than fast logarithmic adders, such as the carry propagate adders, however, their regular structures promote excellent results when compared to traditional CLA adders.

Krishna reddy (2013) implemented 32 bit KSA, BKA and Ladner-Fischer Adder (LFA) and found that the logic levels are more delay increases. Padmajarani (2012) identifies that the PPA is the most flexible and widely used for binary addition and best suited for VLSI implementation. In SA, KS, LF, Knowles adder (KA) the delay is reduced, but in BK adder there is no much difference. Anas Zainal Abidin (2007) This study paper found that the improvement of the gate sizing will decrease the propagation delay but, need more power consumption and take more space for the layout area design. Furthermore, the Compound Gate designs are able to reduce the complexity in the cir-

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9

circuit instead of a number of transistors over than Basic Logic Gates are used for schematic. But the compound Gate performance produces glitch noises at the output signals. Gowthami (2013) designed and implemented 16-bit width operands of various parallel prefix adders on Xilinx Spartan FPGA and proved that the parallel prefix adder's architectures faster and area efficient than RCA. Pawan Kumar (2014) this paper modified the already existing Knowles adder by removing the Black-cell for increasing the speed of execution. Reduction in Combinational Path delay by 7.61% to that of normal Knowles adder was founded.

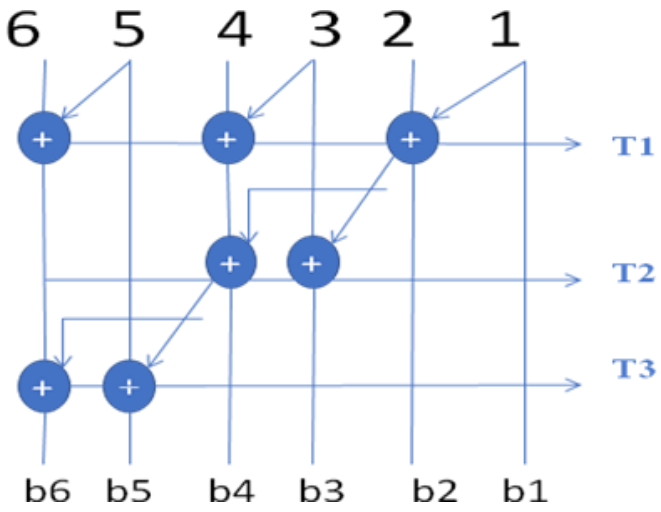
It was concluded that the parallel prefix adders of this type are the best choice in many VLSI applications where power, area and also speed is the main constraint. But he doesn't go for higher bit i.e. 32 bit. Jaspir kauh (2013) This Paper discusses the design of 16 bit PPA and a mixture of two types of adders BKA and KA. It was found that the KA provides better performance from that of parallel prefix and KSA in terms of power, area and Combinational path delay. He doesn't analyze the comparison for higher bits. Martinez (1998) said that the fault tolerant PPA can be implemented using a KSA configuration due to the inherent redundancy in the carry-tree. And also this design can be used to correct a fault in the carry-tree. He suggests that the utilization of a SKSA that is capable of both fault detestability and fault correction. Taeko Matsunaga (2013) designed two additional RCA's that allow fault tolerance to be achieved.

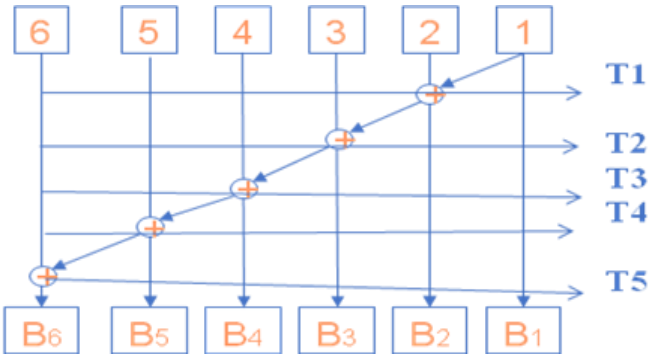
They uses triple mode redundant ripple carry adder (TMR-RCA) for Synthesis and simulation for an. They skip the development of automated techniques for implementing the fully fault tolerant SKSA.

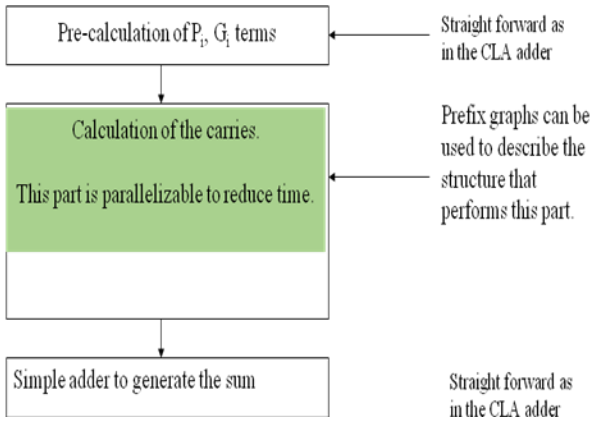
In the above literature, comparisons of Parallel prefix adders are done based on the parameters speed, power consumption. But this work does the comparison based on number of LUT's, number of IOB's, occupied slices, fan-out.

II. PARALLEL PREFIX ADDERS

In parallel prefix adders the word parallel involves the execution of an operation in parallel. This is done by segmentation into smaller pieces that are computed in parallel. Prefix indicates the outcome of the operation depends on the initial inputs. Operation involves arbitrary primitive operator "o" (i.e.) associative is parallelizable.

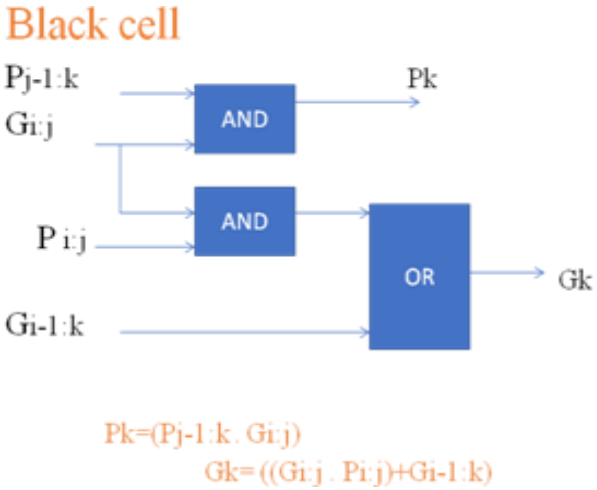






Figs. 1(a) Parallel implementation
Fig. 1(b) Serial implementation

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9



Figs. 3(a) Operation of Black cell

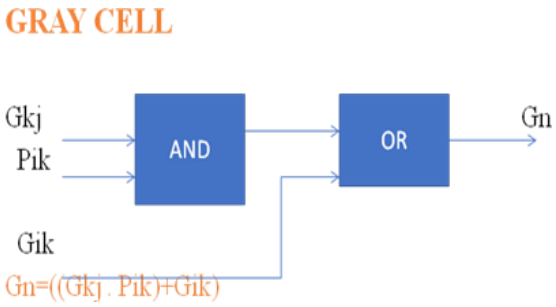


Fig . 3(b) Operation of Grey cell

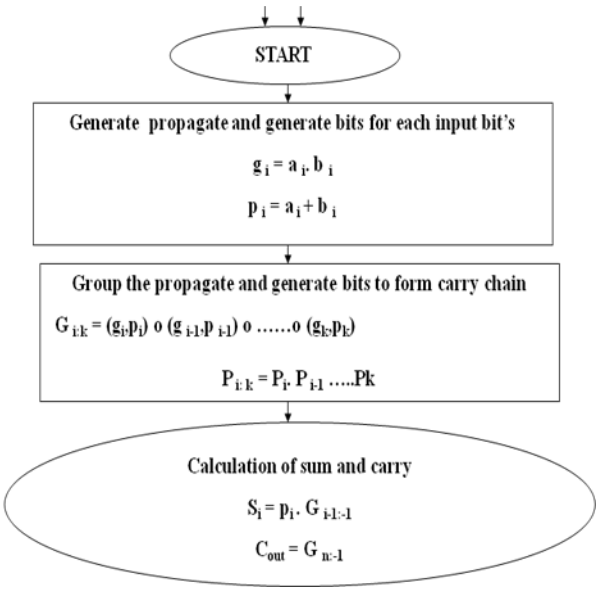


Fig. 4 Flow chart of Parallel prefix adder

The operation of parallel prefix adder and carry-lookahead adders are shown in Figs 1(a) and 1(b) respectively.Parallel-prefix structures can usually be divided into three stages, pre-computation stage, prefix tree stage and post-computation stage. Figure 2 presents the algorithmic steps involved in the parallel prefix adder.It is different from CLA only in the carry generation part; it generates faster carry through parallel process.Figure 4 shows the flow chart for parallel prefix adder which provides the steps to calculate the final sum and carry through propagate and generate bits for individual inputs and group of carry chain.

A. Kogge-Stone adder tree structure

The key of building a prefix tree is according to the specific features of that type of prefix tree and apply the rules. Figure 3(a), (b) shows Gray cells are inserted similar to black cells except that the gray cells final output carry outs instead of intermediate G/P group.

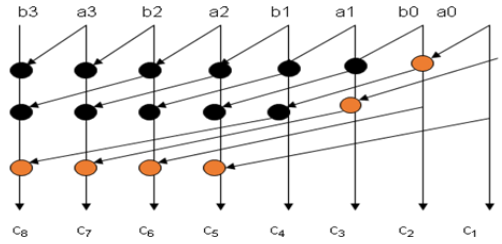


Fig. 58-bit kogge-stone tree structure

Figure 3 shows the 8-bit Kogge-Stone adder (KSA) with propagate paths. From the kogge-stone tree structure at logic level 1, input span is equal to 1.

B. Brent-Kung Adder

Brent-Kung (BKA)has maximum logic depth and minimum area. The delay is estimated as the number of logic levels (i.e. L).

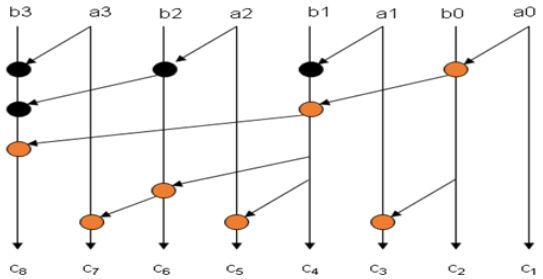


Fig. 6 8-bit Brent –Kung Adder

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9

Figure 6 shows the 8-bit BKA with six logic levels to complete the carry propagate and generate operations. It requires less black cells than KSA, so, circuit complexity was reduced. The BKA is the advanced version of KSA.

C. Sklansky Adder structure

Sklansky(SA)prefix tree takes the least logic levels to compute the carries. Figure 7 shows the 16-bit sklansky prefix tree with critical path in solid line.

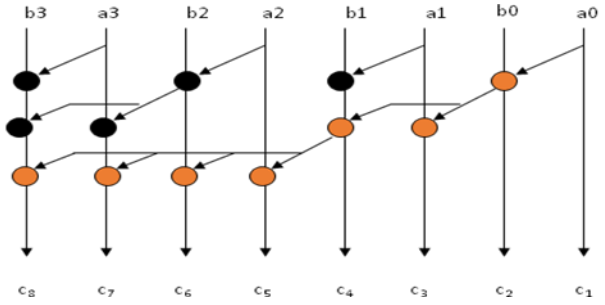


Fig. 7 8-bit Sklansky Adder structure

The structure can be viewed as a compacted version of BKA, where logic level is reduced and fan-out increased. A similar pseudo-code listed for BKA can be used to generate a SA.

D. Han-Carlson Adder

The idea of Han-Carlson Adder (HCA) is similar to KSA structure since it has a maximum fan-out of 2 or f=0. In Fig. 8 the difference is that HCA uses much less cells and wire tracks than KSA. The cost is one extra logic level.

E. Knowles Adder

Knowles proposed a family of prefix trees with flexible architectures. KA uses the fan-out at each logic level to name their family members. Figure 9 shows the 8-bit KA parallel prefix tree structure.

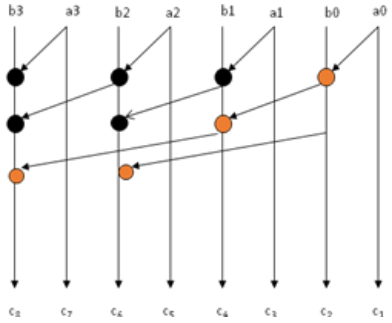


Fig. 8 8-bit Han Carlson Adder

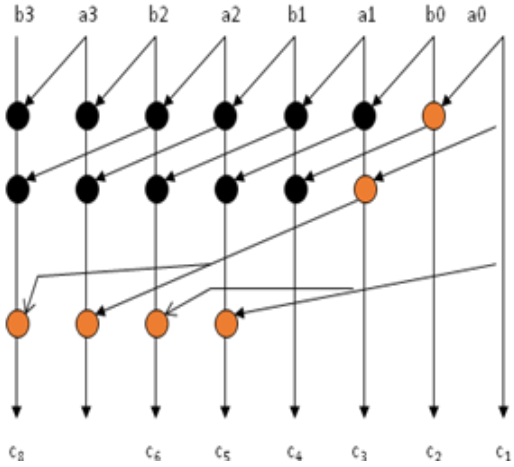


Fig. 9 8-bit Knowles Adder

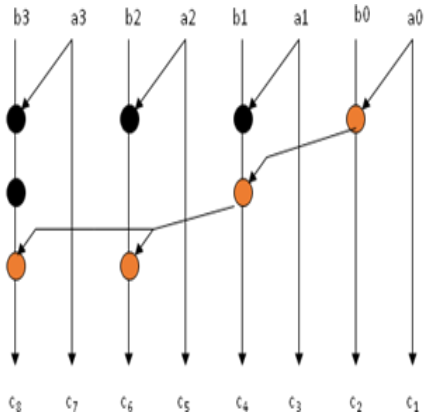


Fig. 10 8-bit Ladner Fischer Adder

F. Ladner Fischer Adder

Ladner fischer Adder (LFA) is a structure that lies between BKA and SA. Sklansky prefix tree has the minimum logic levels, and uses fewer cells than Kogge-Stone and Knowles prefix trees. The major problem of Sklansky prefix tree is its high fan-out. LFA is proposed to relieve this problem. To reduce fan-out without adding extra cells, more levels have to be added. Figure 10 shows an 8-bit structure of LFA.

III. RESULTS AND DISCUSSION

Tree structure of Parallel Prefix Adders design and simulations are done by using Xilinx software Verilog code for PPA tree structures are run by using the ISim simulator and XST synthesis tool. Family Spartan3E, Device – XC3S500E

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	36	9,312	1%
Number of occupied Slices	21	4,656	1%
Number of Slices containing only related logic	21	21	100%
Number of Slices containing unrelated logic	0	21	0%
Total Number of 4 input LUTs	36	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.44		

Fig. 11 Design summary of 16-bit CLA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	77	9,312	1%
Number of occupied Slices	41	4,656	1%
Number of Slices containing only related logic	41	41	100%
Number of Slices containing unrelated logic	0	41	0%
Total Number of 4 input LUTs	77	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.82		

Fig. 2 Design summary of 16-bit KSA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	41	9,312	1%
Number of occupied Slices	22	4,656	1%
Number of Slices containing only related logic	22	22	100%
Number of Slices containing unrelated logic	0	22	0%
Total Number of 4 input LUTs	41	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.26		

Fig. 13 Design summary of 16-bit BKA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	50	9,312	1%
Number of occupied Slices	27	4,656	1%
Number of Slices containing only related logic	27	27	100%
Number of Slices containing unrelated logic	0	27	0%
Total Number of 4 input LUTs	50	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.56		

Fig. 14 Design summary of SA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	49	9,312	1%
Number of occupied Slices	27	4,656	1%
Number of Slices containing only related logic	27	27	100%
Number of Slices containing unrelated logic	0	27	0%
Total Number of 4 input LUTs	49	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.57		

Fig. 15 Design summary of 16-bit HCA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	87	9,312	1%
Number of occupied Slices	46	4,656	1%
Number of Slices containing only related logic	46	46	100%
Number of Slices containing unrelated logic	0	46	0%
Total Number of 4 input LUTs	87	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.79		

Fig. 16 Design summary of 16-bit KA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	50	9,312	1%
Number of occupied Slices	28	4,656	1%
Number of Slices containing only related logic	28	28	100%
Number of Slices containing unrelated logic	0	28	0%
Total Number of 4 input LUTs	50	9,312	1%
Number of bonded IOBs	50	190	26%
Average Fanout of Non-Clock Nets	2.39		

Fig. 17 Design summary of 16-bit LFA

TABLE I: 8-Bit COMPARISONS

ADDERS	Number of input LUTs	No. of occu-pied slices	Average fan -out of Non-clock nets
KSA	22	13	2.15
BKA	18	10	2.06
HCA	20	12	2.14
SA	18	10	2.38
LFA	19	10	2.30
KA	34	19	2.55

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9

TABLE II: 16-BIT COMPARISONS			
ADDERS	**NO. of input LUTs**	**NO. of occupied slices**	**Average fan-out of Non-clock nets**
KSA	77	41	2.82
BKA	41	22	2.26
HCA	49	27	2.57
SA	50	27	2.56
LFA	50	28	2.39
KA	87	46	2.79

TABLE III: DELAY COMPARISON OF PPA			
ADDERS	**4-BIT**	**8-BIT**	**16-BIT**
CLA	11.048	12.887	20.347
KSA	10.045	10.221	15.375
KA	7.550	11.414	12.561
LFA	7.422	9.337	14.443
SA	7.506	10.290	12.112
HCA	7.535	10.131	11.865
BKA	7.856	12.475	12.986

Figures 11, 12, 13, 14, 15, 16 and 17 shows the design summary of CLA, KSA, BKA, SA, HCA, KA and LFA respectively. Tables 1, 2 and 3 show the comparison of 8-bit and 16-bit PPA's. Based on the parameters such as number of LUTs, Slice utilization and number of fan-out's used in the adders, it is understood that BKA shows better performance than other adders. Graphical representation of 8-bit and 16-bit PPA's using the parameters namely LUT, Slices and Fan-Out are shown in Figures18, 19, 20, 21, 22and 23 respectively.

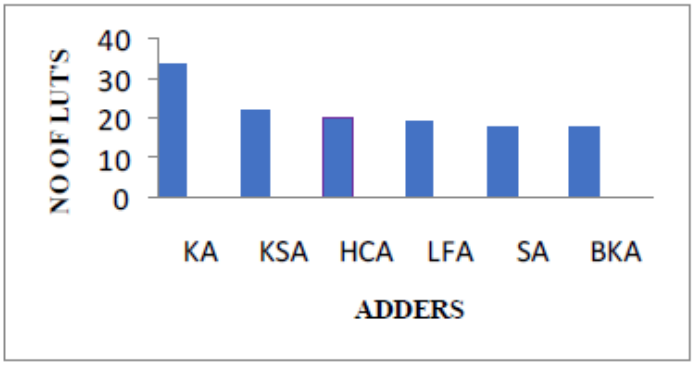


Fig. 18 Graphical plot representing Number of LUTs Vs. Adders (8-bit)

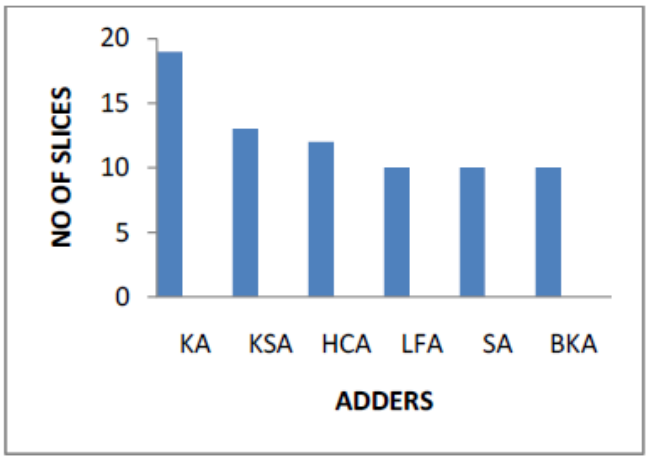


Fig. 19. Graphical plot representing Number of slices Vs. Adders (8-bit)

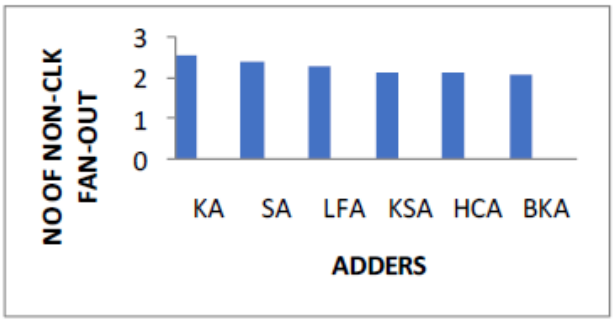


Fig. 20 Graphical plot representing Fan-out Vs. Adders (8-bit)

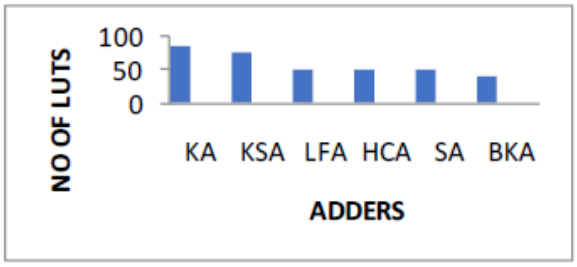


Fig. 21 Graphical plot representing Number of LUTs Vs. Adders (16-bit)

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Volume 9, Issue 6 - June 2022 - Pages 1-9

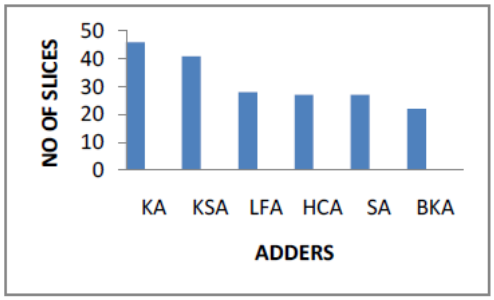


Fig. 22 Graphical plot representing Number of slices Vs. Adders(16-bit)

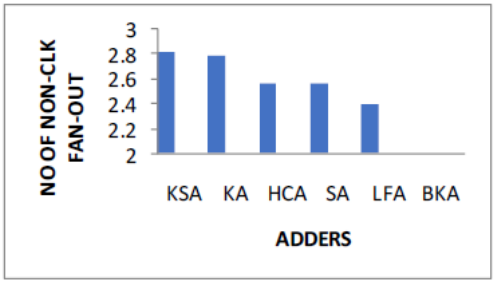


Fig. 23 Graphical plot representing Fan-out Vs. Adders

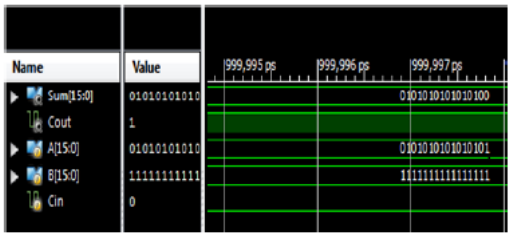


Fig.24 Simulation result of 16-bitCLA

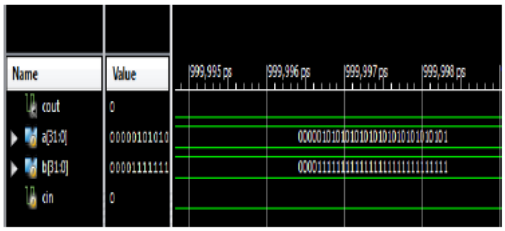


Fig. 25 simulation result of 16-bit KSA

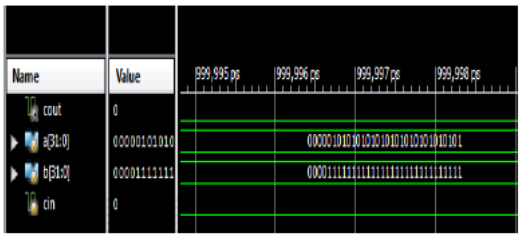


Fig. 26 Simulation result of 16-bit BKA

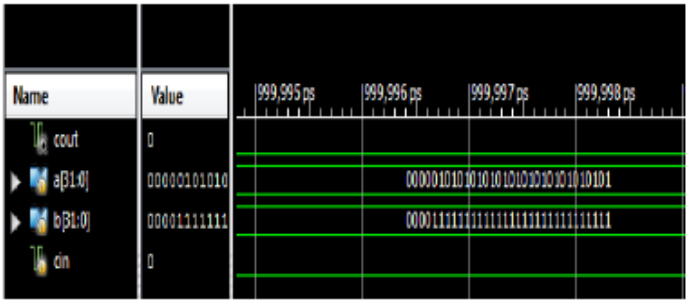


Fig.27 Simulation result of 16-bit SA

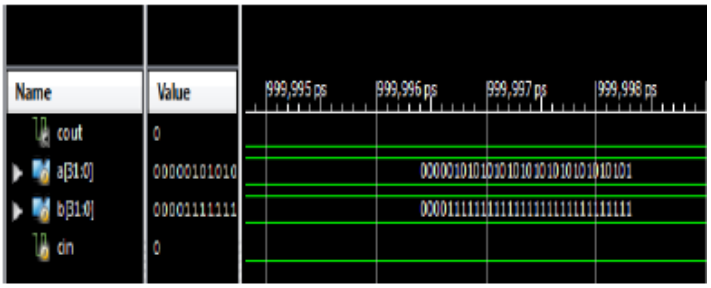


Fig. 28 Simulation result of 16-bit LFA

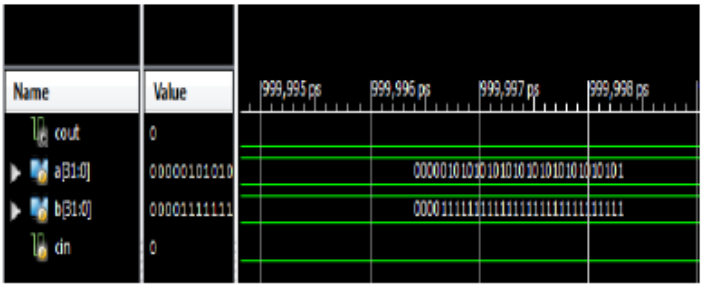


Fig. 29 Simulation result of 16-bit KA

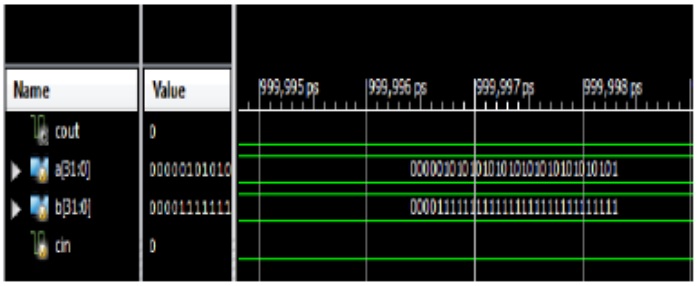


Fig. 30 Simulation result of 16-bit HCA

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9

The simulation results of CLA, KSA, BKA, SA, HCA, KA and LFA are shown in Figures 25, 26, 27, 28, 29 and 30 respectively. The above simulations for all the PPA's have the following inputs and outputs.

a =0101010101010101 and b =1111111111111111 Cin= 0, Cout =1 and Sum=0101010101010101.

IV. CRYPTOGRAPHIC APPLICATIONS

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and that are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. It uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key. Figures 31 and 32 shows the 128-bit AES encrypted and decrypted data.

V. CONCLUSION

In this paper, Parallel Prefix Adders (PPA) such as Kogge-Stone Adder (KSA), Brent-Kung Adder (BKA), Ladner-Fischer Adder (LFA), Knowles Adder (KA), Sklansky adder (SA) and Knowles Adder (KA) structures and operation of the adders are explained. The basic building block of all the PPA is CLA. BKA in 4-bit uses 10 LUT's, 5 slices less than CLA, and the fan-out also reduced by 2 using BKA for carry generation, the circuit complexity was reduced. The performance was increased by increasing the input bit length. While going for 8-bit simulation comparison, BKA adder takes 4 slices less than KSA and 5 slices less than CLA, 7 LUT's less than KSA. Here also the fan-out was reduced by 1, BKA has less wire tracks than CLA and KSA. Similarly in 16-bit simulation the performance was further increased in BKA. It requires 18 LUT's less than CLA, 7 LUT's less than KSA, 11 slices less than CLA and 4 slices less than KSA. Here also the fan-out is less in BKA therefore the circuit complexity was reduced. While going for 32-bit simulation comparison, BKA adder takes 4 slices less than KSA and 5 slices less than CLA, 7 LUT's less than KSA. Here also the fan-out was reduced by 1, BKA has less wire tracks than CLA and KSA.

In 8-bit simulation results, BKA provides 3% less LUT and 5% less slice utilization and 3% lesser Fan-Out. In 16-bit simulation analysis, BKA provides 2% less LUT and 6% less slice utilization and 7% lesser Fan-Out. In 32-bit report, BKA gives 4% less LUT, 3% less slices, 7%less delay and 7% higher fan-out. Finally it was concluded that, when the input bit length increases BKA shows better output results for LUT and slice utilization than other adders. BKA uses less LUT's, slices, IOB's and fan-out and also gives less wire tracks in the adder structure, hence the circuit complexity gets reduced while the input bit length is increased. Finally from the simulation, it was concluded that the BKA provides better performance than other adders.Using BKA, the cryptography is performed using AES encryption and decryption. The cryptography application results are obtained by using Model SIM software.

REFERENCES

1. Uma.R, "Area, Delay and power comparison of Adder Topologies", International Journal of VLSI design & communication Systems, Vol.3, No.1, February 2012.
2. Anjaneyulu. O, "An Improved optimization Technique Adder usingFPGA", International Journal of Modern Engineering Research, Vol. 3, Issue. 5,pp- 3107-3115 Sep. 2013.
3. Babulu. K, "Implementation and Performance Evaluation of Prefix Adders using FPGAs", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)ISSN: 2319 – 4200, ISBN No.:2319 –4197 Vol. 1, Issue 1, PP 51-57 (Sep-. 2012).
4. Swaroop Ghosh, "Novel Low Overhead Fault Tolerant Kogge-Stone Adder Using Adaptive Clocking". Dated:2008.
5. Rabaey. J, "Digital Integrated Circuits:A Design Perspective",Prentice Hall, 1996.
6. Anitha. R, (September 2011)."Braun's Multiplier Implementation using FPGA With Bypassing Techniques", International Journal of VLSI design & Communication Systems (VLSICS), Vol.2, No.3.
7. Trite Sharma,"High Speed, Low Power 8T Full Adder Cell with 45% improvement Threshold Loss Problem", International Journal of VLSI design & Communication systems ,Vol.2, no.3.
8. G.Shyam Kishore, "A NovelFull Adder with High Speed Low Area", 2nd National conference on Information and Communicationtechnology(NCICT) 2011, Proceeding Published in International Journal of

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 9, Issue 6 - June 2022 - Pages 1-9

9. Shubhajit Roy Chowghury," A high speed 8 Transistor Full Adder Design using Novel 3 Transistor XOR Gates", International Journal of Electrical and Computer Engineering.
10. Romaba, "Synthesis of Carry Select Adder in 65nm FPGA", IEEE.
11. Shubin, "Analysis and Comparison of Ripple Carry Full Adders by Speed", Micro/Nano Technologies and Electron Devices (EDM) , International Conference and Seminar on 2010, pp.132-135,
12. Pudi, "Low Complexity Design of Ripple Carry and Brent-Kung Adders in QCA, Nanotechnology, IEEE transactions on, vol.11, Issue.1, pp.105 -119, 2012.
13. Jian-Fei, "A New Full Adder Design for Tree Structured Aritmetic Circuits", Computer Engineering and Technology (ICCET), 2010, 2nd International Conference on Vol.4, pp.v4- 246-v4-249, 2010.
14. Animul Islam, "Design and Analysis of Robust Dual Threshold CMOS Full Adder Circuit in 32 nm Technology", International Conference on Advance in Recent Technologies in communication and Computing, 2010.
15. Deepa Sinha, "Design and Analysis of low Power 1-bit Full Adder Cell", IEEE, 2011.