Parallel Prefix Adders- A Comparative Study For Fastest Response

Er. Aradhana Raju, Richi Patnaik, Ritto Kurian Babu, Purabi Mahato
Department of ECE, Silicon Institute of Technology, Bhubaneswar, Khurda (Dist),
Odisha, Indiaaradhanaraju@gmail.com, patnaik.richi840@gmail.com, rittobabu@gmail.com, purabimahato2014@gmail.com

***Abstract*- VLSI, in modern day technology has seen extensive use of PPA with a better delay performance. These pre-compute the carries and thus have upper hand over the commonly used Ripple Carry Adder (RCA). Addition has been an indispensible operation in most of the widely used applications. We present the 4,8,16 and 32 bit of different adders- Carry Look Ahead (CLA), Carry Save Adder (CSA), Kogge Stone Adder (KSA), Sparse Kogge Stone Adder (SKSA), Brent Kung Adder (BKA), Sklansky Adder, Lander Fischer Adder (LFA) and Han Carlson Adder (HCA). They have been categorized and ranked as per delay, device utilization and cell usage. These adders are implemented in VHSIC Hardware Description Language (VHDL) using Xilinx Integrated Software Environment (ISE) 9.2i Design Suite.**

***Keywords*- Adders, KSA, SKSA, BKA, simulation, synthesis**

I. INTRODUCTION

Binary addition is a fundamental operation that continues to have pivotal role in most modern day digital, control systems, DSP circuits. There are a variety of adders; each has certain performance and its own importance. Each type of adder is selected depending on where the adder is to be used. So the adders are required to have faster computation for great accuracy and small area consumption. Binary adders are one of the most essential logic elements within a digital system. These determine the performance of design. Also, binary adders are helpful in units other than Arithmetic Logic Units (ALU), such as multipliers, dividers and memory addressing units and so on. Any improvements in binary addition can result in a performance boost for any computing device and, thus improve the performance of the entire system. The major drawback for binary addition is the carry chain. With the increase in width of the input operands, the length of the carry chain also increases. In order to improve the performance of the carry-propagate adders, it is possible to accelerate the carry chain, but not eliminate it. As a result, most digital designers come up building faster adders by optimizing computer architecture, because they tend to set the critical path for most computations. In this paper we mainly focus on finding a better adder in terms of delay constraints, device utilization and cell usage.

II. WORKING ARCHITECTURE

Parallel Prefix Adders employ three basic stages to for the adder operation, namely[1]:

1) Pre-computation of P_i and G_i
2) Carry generation network
3) Post-computation

1)Pre-computation stage

This step involves computation of generate and propagate signals corresponding to each pair of bits in A and B.

$Pi = Ai \oplus Bi$

$Gi = Ai \cdot Bi$

2) Carry generation network

The PPA's differ from each other n this stage, as the structures are different for different PPA's. Hence, it forms the deciding factor of the adders. It consists of the processing components and buffer components. Lesser the number of processing components lesser is the delay. The output of buffer component is same as that of input. The output of processing component is

$G_{i:k} = G_{i:j} + P_{i:j} \cdot G_{j-1:k}$

$P_{i:k} = P_{i:j} \cdot P_{j-1:k}$

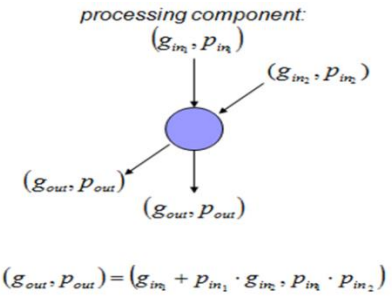

Fig 1. Processing Component

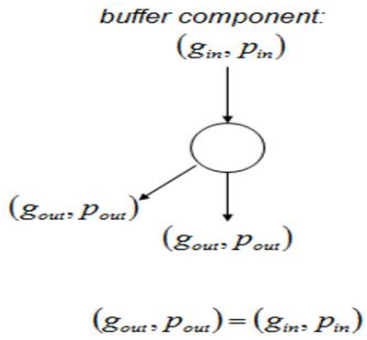

Fig 2. Buffer Component

3)Post-computation

The sum and the final carry is calculated in this stage.

$S_i = P_i \cdot G_{i-1:-1}$

$C_{out} = G_{n:-1}$

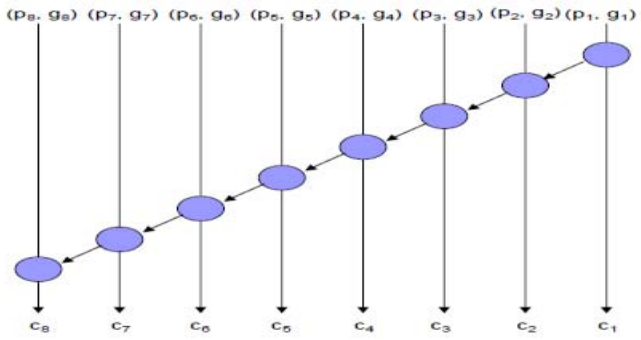

Fig 3.Basic Parallel Prefix Adder Structure

III. DIFFERENT TYPES OF PPA'S

A. 16 Bit Kogge Stone Adder

Kogge-Stone adder is a common design for high performance adders. It is a parallel form of carry look ahead adder[2]. It has low logic depth, high node count and minimum fan-out[3]. Low logic depth and minimum fan-out gives faster performance, but high node count results in larger area. It is generally considered as the fastest adder[4].

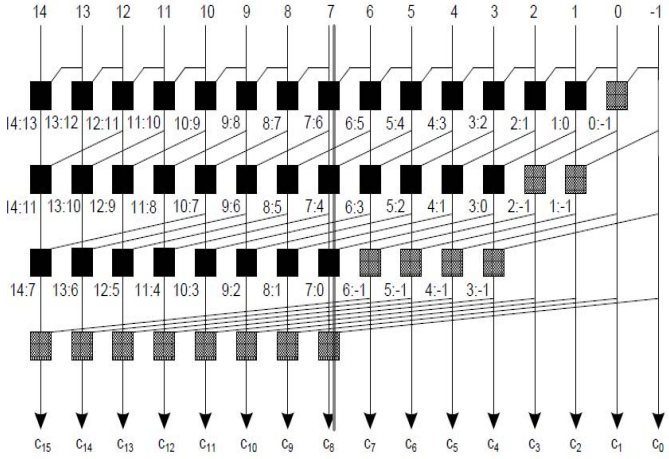

Fig 4. 16 Bit Kogge-Stone Adder

B. 16 Bit Sparse Kogge Stone Adder

The sparse Kogge-Stone adder consists of several smaller ripple carry adders (RCAs) on its lower half, a carry tree on its upper half[5]. The number of carries generated is less in a Sparse Kogge Stone adder compared to the regular Kogge-Stone adder [6]. The functionality of the black cell and the gray cell remains exactly the same as in the regular Kogge-Stone adder. Only in the Sparse Kogge-Stone adder, the design terminates with a 4- bit RCA.

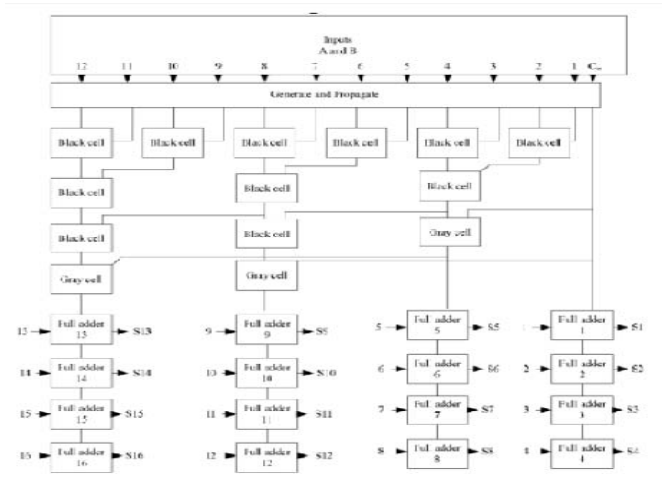

Fig 5. 16 Bit Sparse Kogge-Stone Adder

C. 16 Bit Brent Kung Adder

It is another form of PPA having a different structure as shown below. It has a minimum number of fan-out. It has less wiring congestion and takes less area to implement. The number of processing components is less as compared to KSA or SKSA. Hence, it reduces delay without compromising the power performance of the adder[7].

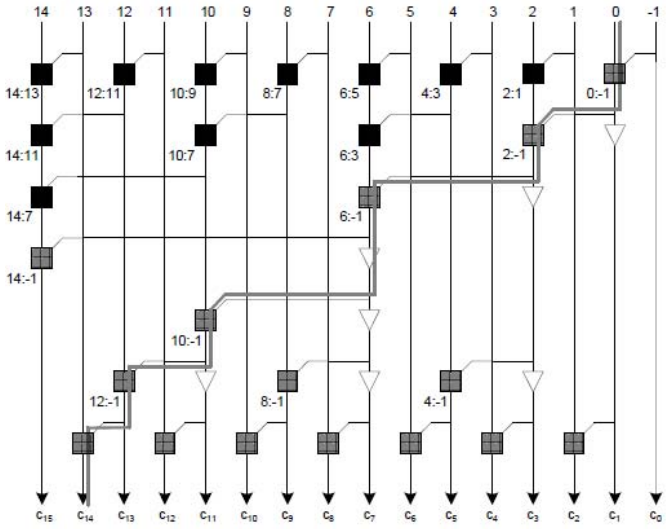

Fig 6. 16 Bit Brent-Kung Adder

D. 16 Bit Sklansky Adder

The structure can be viewed as a compacted version of Brent-kung's, where logic levels is reduced and fan-out increased. In this adder, binary tree of propagate and generate cells will first simultaneously generate all the carries, Cin. It builds recursively 2-bit adders then 4-bit adders, 8-bit adders, 16-bit adder and so on by abutting each time two smaller adders. The architecture is simple and regular, but it suffers from fan-out problems. Besides in some cases it is possible to use less propagate and generate cells with the same addition delay [7].

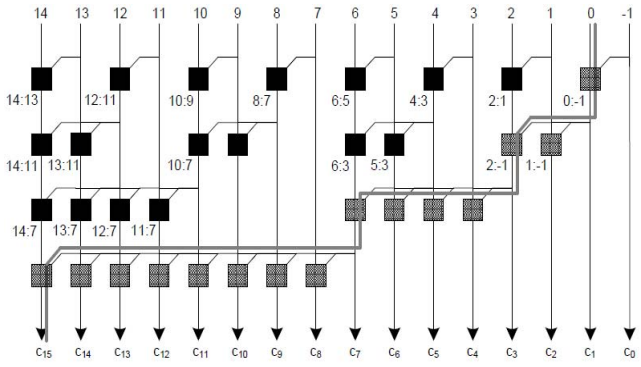

Fig 7. 16 Bit Sklansky Adder

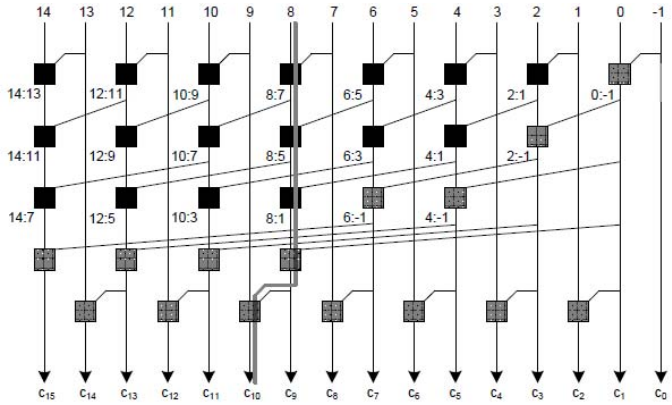

Fig 9. 16 Bit Han-Carlson Adder

E. 16 Bit Ladner Fischer Adder

This adder structure has minimum logic depth, but has large fan-out requirement up to n/2 [8]. In 1980, Fischer and Richard Ladner presented a parallel algorithm for computing prefix sums efficiently. They show how to construct a circuit that computes the prefix sums in the circuit, each node performs an addition of two numbers. With their construction, one can choose a trade-off between the circuit depth and the number of nodes[9].

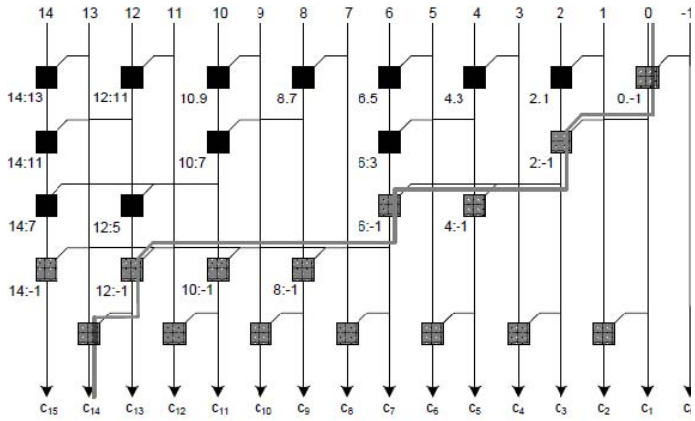

Fig 8. 16 Bit Ladner-Fischer Adder

F. 16 Bit Han-Carlson Adder

It is a hybrid design of Kogge-Stone and Brent Kung. It has less number of cells and wire tracks as compared to Kogge-Stone at the cost of one extra logic level. Generate and propagate signals of odd bits are transmitted down the prefix tree. They recombine with even bits carry signals at the end to produce the true carry bits. Thus, the reduced complexity is at the cost of adding an additional stage to its carry-merge path[7].

IV. SIMULATION AND RESULTS

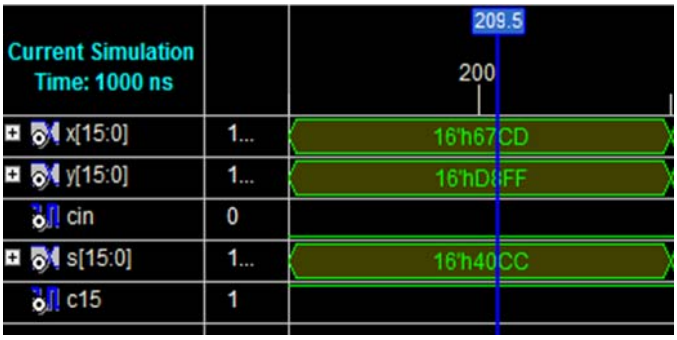

Fig 10. 16 Bit KSA

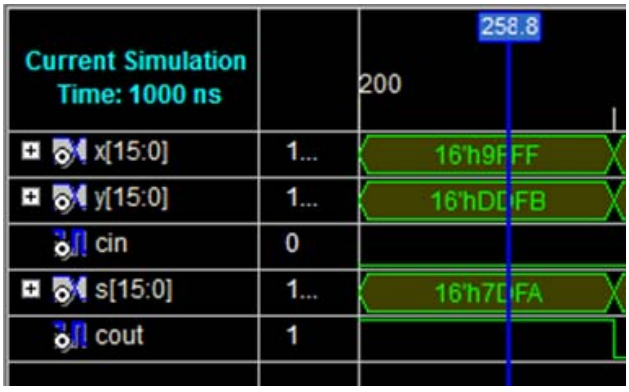

Fig 11. 16 Bit SKSA

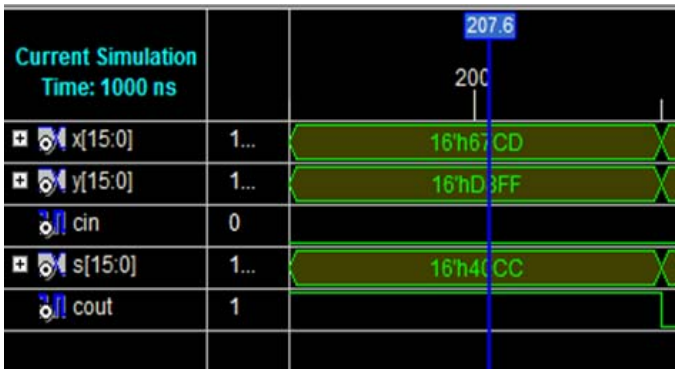

Fig 12. 16 Bit BKA

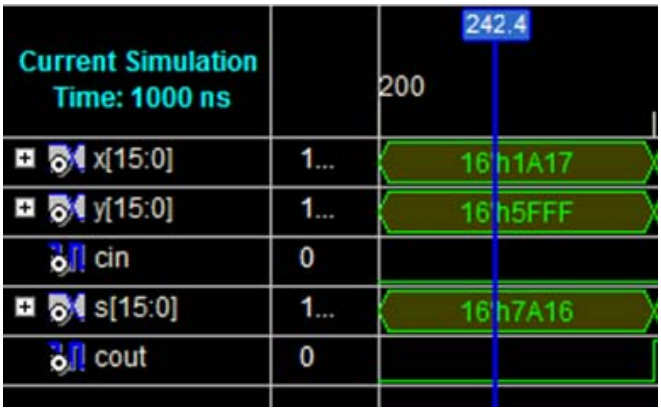

Fig 14. 16 Bit Sklansky Adder

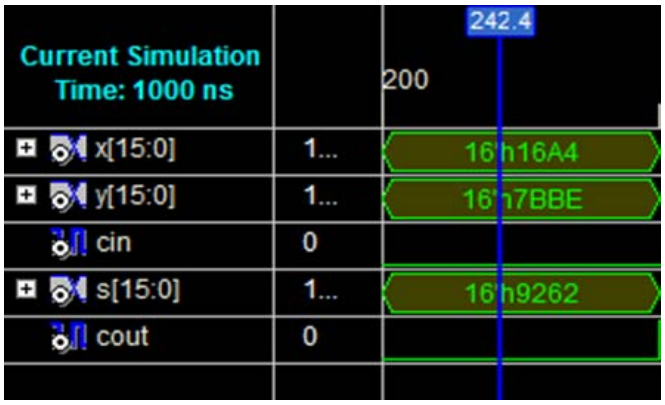

Fig 13. 16 Bit Ladner Fischer Adder

Fig 15. 16 Bit Han Carlson Adder

V. COMPARISION TABLES

A. Delay Analysis

Table I. 4 Bit Adders

SL. NO.	NAME of ADDER	LOGIC DELAY (ns)	ROUTE DELAY (ns)	TOTAL DELAY (ns)
1.	CLAG	6.219(78.1%)	1.748(21.9%)	7.967
2.	**KSA**	**5.898(78.9%)**	**1.581(21.1%)**	**7.479**
3.	SKSA	6.602(75.5%)	2.146(24.5%)	8.748
4.	BKA	6.602(76.3%)	2.050(23.7%)	8.652

Table II. 8 Bit Adders

SL. NO.	NAME of ADDER	LOGIC DELAY (ns)	ROUTE DELAY (ns)	TOTAL DELAY (ns)
1.	CLAG	9.418 (69.8%)	4.074 (30.2%)	13.492
2.	KSA	9.418 (69.8%)	4.074 (30.2%)	13.492
3.	SKSA	9.418 (70.5%)	3.934 (29.5%)	13.352
4.	**BKA**	**8.714 (69.8%)**	**3.766 (30.2%)**	**12.480**

Table III. 16 Bit Adders

SL. NO.	NAME of ADDER	LOGIC DELAY (ns)	ROUTE DELAY (ns)	TOTAL DELAY (ns)
1.	CLAG	15.050 (65.7%)	7.847 (34.3%)	22.897
2.	KSA	13.518 (66.7%)	6.742 (33.3%)	20.260
3.	**SKSA**	**11.530 (67.4%)**	**5.572 (32.6%)**	**17.102**
4.	BKA	12.938 (64.0%)	7.292 (36.0%)	20.230
5.	Sklansky	13.580 (65.4%)	7.172 (34.6%)	20.752
6.	LFA	13.259 (64.5%)	7.302 (35.5%)	20.561
7.	HCA	12.938 (64.8%)	7.026 (35.2%)	19.964

Table IV. 32 Bit Adders

SL. NO.	NAME of ADDER	LOGIC DELAY (ns)	ROUTE DELAY (ns)	TOTAL DELAY (ns)
1.	**KSA**	**13.642 (64.0%)**	**7.684 (36.0%)**	**21.326**
2.	SKSA	19.854 (63.8%)	11.274 (36.2%)	31.128
3.	BKA	13.963 (63.0%)	8.194 (37.0%)	22.157
4.	LFA	20.237 (62.0%)	12.403 (38.0%)	32.640
5.	HCA	15.754 (62.7%)	9.356 (37.3%)	25.110
6.	Sklansky	19.533 (62.6%)	11.694 (37.4%)	31.227

B. Device Utilization and Cell Usage

Table V. 4 Bit Adders

SL. NO.	NAME of ADDER	DEVICE UTILIZATION SUMMARY				CELL USAGE			
		NO. of SLICES	4 INPUT LUT's	IO's	BONDED IOB'S	BELS	IOB's	IBUF	OBUF
1.	CLAG	3	6	14	13	8	13	8	5
2.	KSA	4	7	14	14	8	14	8	6
3.	SKSA	4	7	14	14	8	14	8	6
4.	BKA	4	7	14	14	8	14	8	6

Table VI. 8 Bit Adders

SL. NO.	NAME of ADDER	DEVICE UTILIZATION SUMMARY				CELL USAGE			
		NO. of SLICES	4 INPUT LUT'S	IO's	BONDED IOB's	BELS	IOB's	IBUF	OBUF
1.	CLAG	9	15	26	26	16	26	16	10
2.	KSA	9	15	26	26	16	26	16	10
3.	SKSA	9	15	26	26	16	26	16	10
4.	BKA	11	19	26	26	22	26	16	10

Table VII. 16 Bit Adders

SL. NO.	NAME of ADDER	DEVICE UTILIZATION SUMMARY				CELL USAGE			
		NO. of SLICES	4 INPUT LUT'S	IO's	BONDED IOB'S	BELS	IOB's	IBUF	OBUF
1.	CLAG	18	32	50	50	33	50	32	18
2.	KSA	36	64	50	50	73	50	32	18
3.	SKSA	23	42	50	50	48	50	32	18
4.	BKA	24	43	50	50	50	50	32	18
5.	Sklansky	20	37	50	50	44	50	32	18
6.	LFA	22	40	50	50	47	50	32	18
7.	HCA	30	52	50	50	55	50	32	18

Table VIII. 32 Bit Adders

SL. NO.	NAME of ADDER	DEVICE UTILIZATION SUMMARY				CELL USAGE			
		NO. OF SLICES	4 INPUT LUT'S	IO'S	BONDED IOB'S	BELS	IOB'S	IBUF	OBUF
1.	KSA	89	157	98	98	164	98	64	34
2.	SKSA	57	104	98	98	123	98	64	34
3.	BKA	49	88	98	98	103	98	64	34
4.	LFA	46	84	98	98	100	98	64	34
5.	HCA	59	107	98	98	122	98	64	34
6.	Sklansky	50	90	98	98	106	98	64	34

VI. CONCLUSION

Parallel-prefix structures have been found to be attractive for adders due to its logarithmic delay. This project analysis has paved the way for the development of Adders Design with reduced delay. Our sole aim was finding out the fastest adder among the various kinds of binary and PPA's discussed above. The findings showed us that there is hardly any difference in delays when the lower bits are taken into account, but as we increase the number of bits there is a remarkable difference in delays. Again, higher the LUT's used higher will be the area consumed. Results suggest that no single type of architecture is the best for all bit values, rather offer enough insights of which type of adders is the best for a specified bit value. The best numbers in the comparison view are highlighted in bold fonts. Looking forward to the future aspects we can elaborate on other important design parameters viz. area, power, energy and can be compared. It would be worthwhile for future FPGA designs to include an optimized carry path, so that tree based adder designs can be optimized for place and routing.

VII. REFERENCES

[1] Mr. Deepak Raj, Mrs. Sahana K Adyanthaya , Prof. Praveen J, Prof. Raghavengra Rao R, "Design and Implementation of different types of efficient parallel prefix adders", *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering(IJIREEICE),* Vol. 3,Special Issue 1, April 2015.

[2] T.Kiran Kumar, P.Srikanth," Design of High Speed 128 bit Parallel Prefix Adders", *Journal of Engineering Research and Applications (IJERA),* ISSN : 2248-9622, Vol. 4, Issue 11(Version 3), Nov 2014.

[3] Raghumanohar Adusumilli and Vinod Kumar K, "Design and Implementation of a High Speed 64 Bit Kogge-Stone Adder Using Verilog Hdl", *International Journal of Electrical and Electronic Engineering and Telecommunication (IJEETC),* ISSN-2319-2518, vol-4 No. 1, Jan 2015.

[4] Pakkiraiah Chakali, Madhu Kumar Patnala, "Design of High Speed Kogge Stone Based Carry Select Adder", *International Journal of Emerging Science and Engineering (IJESE),* ISSN: 2319-6378, vol 1, Issue 4, Feb 2014.

[5] E. Sreenivasa Goud, P.C. Praveen Kumar, "Design and Characterization of Sparse Kogge Stone Parallel Prefix Adder Using FPGA", *International Journal of Scientific Engineering and Technology Research,* ISSN 2319-8885, vol 03, Issue.06, July-2013.

[6] A. Madhu Babu, K. Harikrishna, "Evaluation of High Speed and Low Memory Parallel Prefix Adders", *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE),* e-ISSN:2278-1676, p-ISSN:2320-3331, vol 8, Issue 4, Nov-Dec 2013.

[7] Megha Talsania, Eugene John, "A Comparative Analysis of Parallel Prefix Adders", University of Texas at San Antonio.

[8] P. Chaitanya Kumari, R. Nagendra, "Design of 32 bit Parallel Prefix Adders", *IOSR journal of Electronics and communication Engineering (IOSR-JECE),* e-ISSN:2278-2834, p-ISSN:2278-8735, vol. 6, Issue 1 May-June 2013.

[9] T. Kiran Kumar, P. Srikanth, "Design of High Speed 128 bit Paralel Prefix Adders", *International Journal of Engineering Research and Applications,* ISSN: 2248-9622, vol 4, Issue 11(version 3), Nov 2014.

[10] CH Sudha Rani, CH Ramesh,"Design and Implementation of High Performance Parallel Prefix Adders", *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE),* ISSN (Online): 2320-9801, ISSN(Print):2320-9798, vol. 2, Issue 9, Sept 2014.

[11] V. Krishna Kumari, Y. Sri Chakrapani, "Designing and Characterization of Kogge Stone, Sparse Kogge Stone, Spanning Tree and Brentkung Adders", *International Journal of Modern Engineering Research (IJMER),* ISSN: 2249-6645, vol. 3, Issue 4, July-Aug 2013.

[12] Srinivasamanoj. R, M. Sri Hari, B. Ratna Raju,"High Speed VLSI Implementation of 256-bit Parallel Prefix Adders", *International Journal of Wireless Communications and Networking Technologies (WARSE),* ISSN: 2319-6629, vol. 1, No. 1, Aug-Sept 2012.

[13] Sunil M., Ankith R D, Manjunatha G D and Premananda B S, "Design and Implementation of Faster Parallel Prefix Kogge Stone Adder", *International Journal of Electrical and Electronic Engineering & Telecommunications (IJEETC),* ISSN: 2319-2518, vol. 3,No. 1, Jan 2014.

[14] Mangesh B Kondalkar, "Improved Fault Tolerant Sparse Kogge Stone Adder**".** *International Journal of Computer Applications (0975 – 8887) Volume 75– No.10,* Aug 2013.

[15] M.Venkata Durga rama raju, "Design and implementation of Parallel Prefix Adders using FPGAs" . *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 6, Issue 5 PP 41-48,* Jul. - Aug. 2013.

[16] Madagani Nagamani, "Design and Estimation of Delay, Power and Area for Parallel Prefix Adders". *International Journal of Advanced Research Foundation, Volume 2, Issue 8,* August 2015.