Design and implementation of Kogge-stone, Sparse Kogge-stone and Spanning tree adder

Dr B N Shashikala
Dept. of Electronics and Communication Engg
Siddaganga Institute of Technology
Tumakuru, India
deepashashikala@sit.ac.in

Pooja R
Dept. of Electronics and Communication Engg
Siddaganga Institute of Technology
Tumakuru, India
1si19ec068@sit.ac.in

Pavan Prahallad
Dept. of Electronics and Communication Engg
Siddaganga Institute of Technology
Tumakuru, India
1si19ec066@sit.ac.in

Ramya S K
Dept. of Electronics and Communication Engg
Siddaganga Institute of Technology
Tumakuru, India
1si19ec075@sit.ac.in

Ramyashree M V
Dept. of Electronics and Communication Engg
Siddaganga Institute of Technology
Tumakuru, India
1si19ec076@sit.ac.in

2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES) | 979-8-3503-4729-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSSES58299.2023.10200644

***Abstract*—Many Very Large Scale Integration (VLSI) circuits typically employ Parallel Prefix Adders (PPA). Power, size, and speed are the key constraints for designing an adder circuit because there is a trade-off between them. In-depth research is still being done on minimising the power delay and thus improving adder performance because the binary adder is an essential part of the majority of digital circuit designs, as well as Digital Signal Processors (DSP) and microchip knowledge path units. The Kogge-stone (KSA), Sparse Kogge-stone (SKS), and Spanning tree adders are examined in this research along with current adder designs. Their performances are compared in light of various design constraints to discover which adder performs the best among them.**

***Index Terms*—Accuracy, adder, area, delay, power, speed.**

I. Introduction

Adders play a crucial role in processors and data route units in microprocessors. Parallel adders are known to perform the best in VLSI implementations. For many practical designs involving mobile DSP and telecommunication applications, reconfigurable logic, such as Field Programmable Gate Arrays (FPGAs), has been growing in popularity in recent years. This is because FPGAs offer better performance in terms of speed and power over DSP- and microprocessor-based solutions. In comparison to Application Specific Integrated Circuit (ASIC) designs, a significant reduction in development time and cost is necessary. With the growing popularity of mobile and portable electronics, which heavily rely on DSP functions, the power advantage is crucial. Parallel prefix adders will operate differently than VLSI versions due to the design of the programmable logic and routing resources in FPGAs. Real-time signal processing, like signal processing, also uses floating point arithmetic units and arithmetic logic units (ALUs). When arithmetic computations are base ten decimals, they are simple for humans to perform. However, if binary numbers are given, they turned pragmatist. Any enhancement to binary addition can enhance system performance. The performance of the adders has a major impact on the system's speed and accuracy. The structure of the programmable logic and routing resources in FPGAs, however, allows parallel- prefix adders to function differently from VLSI counterparts. Modern FPGA specifically use a fast-carry chain that stream- lines the carry path for the simple Ripple Carry Adder. PPA adders are a subfamily of the well-known carry look-ahead adders. KS, SKS, and Spanning tree adders are the parallel prefix adders. These adaptable adders are used to accelerate binary additions. The benefit of employing a tree structure form is that it accelerates mathematical operations. The objective of this paper is to help adder designers in spreading their knowledge, and it concentrates on giving a current overview of the state of the art in parallel adders and application.

II. Operating Principle of PPA

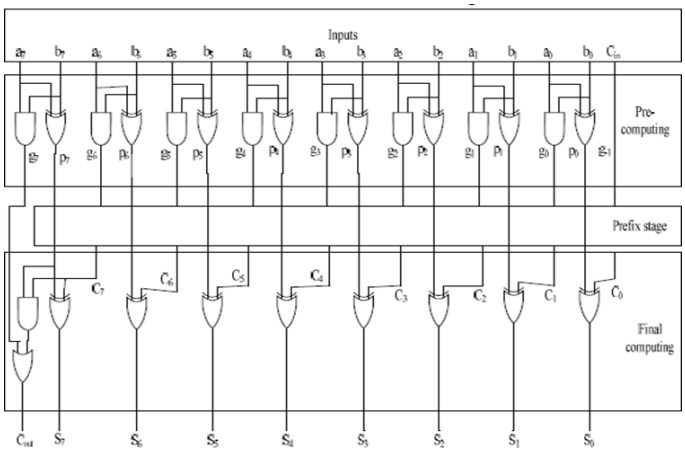

Fig. 1. Block diagram of PPA

Fig. 1 shows the block diagram of PPA. High performance adders are discovered to frequently use parallel-prefix architec-

tures because the delay is inversely related to the adder width. Precalculation, Prefix stage, and Final computation make up the three main phases of PPAs.

1. Pre computation Stage

Propagates and generates are computed for the given inputs using the specified equations (1) and (2) during the pre-computation stage.

$$Pi = Ai\ xor\ Bi \tag{1}$$

$$Gi = Ai\ and\ Bi \tag{2}$$

2. Prefix Stage In the prefix stage, group generate/propagate signals are computed at each bit using the given equations. The black cell (BC) generates the ordered pair in equation (3), the gray cell (GC) generates only left signal, following (4) and (5)

$$(GL, PL)o(GR, PR) = (GL + PL.GR, PL, PR) \tag{3}$$

$$G(i:k) = G(i:j) + P(i:j).G(j-1:k) \tag{4}$$

$$P(i:k) = P(i:j).P(j-1:k) \tag{5}$$

More practically, the equations (4) and (5) can be expressed using a symbol "o "denoted by Brent and Kung. Its function is exactly the same as that of a black cell i.e.

$$(G:P)i:k = ((G:P)i:j)o(G:P)j-1:k) \tag{6}$$

The "o" operation will help make the rules of building prefix structures.

3. Final Computation Stage In the final computation, the sum and carryout are the final output.

$$Si = Pi.G(i-1:-1) \tag{7}$$

$$Cout = G(n:-1) \tag{8}$$

Where "-1" is the position of carry-input. The generate/propagate signals can be grouped in different fashion to get the same correct carries. Based on different ways of grouping the generate/propagate signals, different prefix architectures can be created.

III. LITERATURE REVIEW

The carry-tree adders i.e, KS, SKS, spanning tree, BKS Adders are proposed and implemented using Spartan 3E FPGA chip [1]. Most effective circuits for binary addition are known to be PPA. A lot of study has been done to reduce the complexity of the circuit and the power used by the adder because the final carry is generated prior to the formation of the sum. Implemented and evaluated with the straightforward RCA and CSA are the four different types of carry-tree adders (KS, SKS, spanning tree, and BKS Adder). These configurations with different bit widths are simulated using a Spartan 3E FPGA from Xilinx. According to the experimental findings, the BKS Adder uses less power (10.82 mW) and has a shorter latency (5.96 ns) than other adders. The power usage served as the basis for the performance study.

Parallel Prefix form of Carry Look Ahead Adders (CLAA) are described and comparison is done [2]. The PPA Adders (KS, SKS, Spanning Tree and Brent kung adders) are designed and implemented using Xilinx vertex 5FPGA and the adder delays were estimated. The 4-bit RCA marks the end of the 16-bit SKA, which computes the carries using Black Cells (BCs) and Grey Cells (GCs). The employed 16-bit STA terminates with RCA, employs BCs, GCs, and complete adders, although there are differences in how they are connected. In comparison to SKA and STA, the 16-bit KSA design had reduced delay and utilised 16-bit BCs and 15 GCs. Since the 16-bit BKA utilised fewer BCs and GCs than the KSA did, it has a simpler architecture.

KSA with low power consumption and delay is proposed and compared with RCA [3]. RCAs are chosen over all other sorts of conventional approaches because they offer the fastest design times when adding two N- bit values. However, RCAs have the restriction that each full adder block must wait until carry bits have been created from earlier full adder blocks. The KSA that was put into use was a parallel prefix for CLAA. PPAs are a type of tree-based structure that accelerates binary addition. Prefix adders are thus employed in quick addition algorithms. In compared to other traditional logics, the experimental result demonstrates that addition utilising KSA reduces power consumption and delay (20.26 ns). The power consumption served as the basis for the performance study.

Four different forms of PPAs: KSA, STA, BKA, and SKSA are described and implemented using Spartan 6 FGPA chip [4]. With the use of the Xilinx Integrated Software Environment (ISE) 13.2 Design Suite, these adders are implemented in verilog Hardware Description Language (HDL). These designs are implemented using Spartan 6 FPGAs from Xilinx. (FPGA). Measurements of the adder's delay, power, and area are made, and they are all compared based on research into analyses of delay, area, and power. When RCA is compared to KSA, efficiency has increased by 6.5 percent, and for the Han Carlson adder (HCA) in the Spartan 6 FPGA chip, it has increased by 2.53 percent. Because RCA uses the least amount of logic in its compression, it is the best.

KS, SKS, ST and BK adders are described and implemented using Xilinx vertex 5 FPGA and the adder delays were estimated [5]. The adder delays were approximated. With BCs and GCs to compute the carriers, the 16-bit SKA concludes with a 4-bit RCA. The 16- bit STA in use uses full adders, BCs, GCs, and termination with RCA, but there are differences in how they are connected. With less delay than SKA and STA, the 16-bit KSA design utilised 16-bit BCs and 15 GCs. The 16-bit BKA used 14 BCs and 11 GCs, which is less than KSA and results in a smaller architecture

and less space usage. According to the measurement of the delay, SKA and BKA have about the same delay.

KS PPA is implemented using different design techniques [6]. CMOS and GDI are the two basic design techniques used (Gate Diffusion Input). Design and simulation of logic gates employing GPDK 180nm technology were performed using CADENCE Design Suit 6.1.6, Virtuoso, and ADE Environment. For KSA performance, factors like as delay, transistor count/gate count (area), and power are considered. Simulation is used to study input data with four, eight, and sixteen bits. The results of the comparison investigation demonstrated how the performance of adder architecture with 4 and 16 bits can differ.

PPA based on KS adder is implemented [7]. An adder architecture is designed: Hyper-Parallel Prefix Adder (HPPA) based on the Grouped-KS Tree. In improved HPPA the addition process is divided into two levels, both of which uses PPA architecture. The addition procedure is split into two layers in improved HPPA, both of which employ PPA architecture. By shortening the wires in the essential routes, it will decrease the delay of wire loads, greatly enhancing the adder's performance. Also, a Grouped-KS tree, an efficient carry tree structure built on the KS tree, is created. A 64-bit complete adder was created by fusing HPPA and GKS trees. To test the performance, Verilog HDL was used to design these adders. The adder is implemented using CMOS 0.13um technology. The proposed adder configuration has a maximum delay of 578 ps and an average power of 18.3 mW.

Basic SKS adder is implemented first and implementation of the SKS adder using carry select logic is performed secondly that result in reduction in critical path delay and increase in speed [8]. The ISI (VHDL/Verilog) simulator was used to simulate and synthesise the SKS adder. Slices, flip-flops, LUTs, and IOBs have been used to calculate the necessary area. When compared to a simple SKS adder, the updated design exhibits an increase in speed and a sizable decrease in delay. The SKS adder can therefore be adapted to be utilised for a variety of high-speed applications. In future, the PPA must be tested for other adders also to optimize the area and timing both.

The comparative analysis is done between the 8-bit KS and Han Carlson PPA [9]. Parallel prefix algorithm is employed to improve the speed of the addition. The adders are designed and simulated in 130nm technology using Tanner EDA tool. In terms of cost, area, power between the two PPAs, the Han-Carlson adder is the best choice. The Han-Carlson adders's area or power raises as the bit size increases but it does not raise drastically compared to KS adder. Han-Carslon adder gives good trade-off between fan-out and the number of logic cells. But in terms of the time propogation delay, KS adder is the better choice. Ks has very low propogation delay, but as the bit size increases the delay also increases. Therefore, only for fast addition KS adder is used but for better area usability, low power consumption and high speed Han-Carlson adder is employed.

The KS adder, SKS adder, Spanning tree and BK adders are compared by using power and delay constraints [10]. These adders are subjected to simulation and synthesis using Model Sim6.4b and Xilinx ISE9.2i. Performance-wise, the BK adder outperforms the RCA and CSA. The SKS adder and Spanning tree adder can improve the power-delay performance. In many VLSI applications where power is the primary restriction, these parallel prefix adders are the optimum option. Future FPGA designs could incorporate an efficient carry path to support tree-based adders.

The three types of carry-tree adders KS, SKS and then spanning tree adder were investigated and are compared with simple RCA [11]. These designs, which used different bit widths, were implemented on a Xilinx Virtex5 FPGA, and delay values were derived through static timing analysis of synthesis outcomes received from the Xilinx ISE design suite 10. The BK adder takes up less space than other adders, but it doesn't differ significantly with the new technique. Skalansky, LF adders take up a little bit more space in the new approach than the old way. In the new strategy, KS and Knowles adders take up greater space. The HC adder hardly changes at all with the new method.

The three types of carry-tree adders i.e KS, SKS and spanning tree adder are compared with the simple RCA and CSA [12]. These designs with various bit-widths were built on a Xilinx Spartan 3E FPGA, and high-performance logic analyzer measurements of delay were taken. Up to 128 bits, the RCA designs perform better in terms of delay performance since they have a quick carry-chain. As bit widths go closer to 256, it is anticipated that the carry-tree adders would outperform the RCA in terms of speed. At high bit-widths, which are anticipated to be in the 128 to 256 bit range, carry-tree adders finally outperform linear adder designs in terms of performance.

The Gate Division Input (GDI) technique was used to design the circuits for high speed and low power applications [13]. Using the EDA tool MENTOR GRAPHICS, a KSA schematic based on 64 bit GDI logic was created in 130nm technology. Performance parameters like delay and average power consumption were measured and the best adder in terms of performance was observed as the one with a delay of 407.07ps designed in GDI Technique. When compared to the other adders, KS Adder was identified as the fastest adder and also had a lower fan-out at the output which increased its performance but on the other hand, it occupied much area (507 nW)and created wiring congestion problems. Much more advanced tool support was required to design KSA with the exact expected performance.

A new PPA architecture called KSA was proposed for 8, 16, 32 and 64-bit addition [14]. Implementing the suggested method and comparing the KSA and CSKA in terms of area, delay, speed, and power consumption corroborated the findings. In comparison to the CSKA (796 nW), the proposed KSA's achieved findings showed the lowest power usage (507 nW). High speed applications can benefit greatly from the suggested strategy.

The Quaternary logic was proposed in place of full adders for improving the performance of the PPA [15]. In addition, based on the quantity of components (no of full adders), performance comparisons of the corresponding adders are carried out. Binary logic gates and radix converters are used to create quaternary half adders. This architecture can be used to build a high performance adder like a spanning tree, which can have fewer full adder components by using quaternary full adders instead of conventional full adders.

Ripple Carry Adder was implemented using half adder and full adders [16]. Any amount of additions can be performed using RCA. The RCA serial adder has a commutation delay issue. The delay is increased when both the half adder and the full adder are increased. PPA consists of the KS, SKS, spaning tree, and brentkung adders. These adders were created and implemented using the FPGA Sparton3E kit, model sim6.4b, and Xilinx ISE 10.1. The results of this study demonstrate that, at low to moderate bit widths, parallel-prefix adders are not as efficient as the straightforward ripple-carry adder. This is predicted given that the ripple carry adder performs best on the Xilinx FPGA's fast carry chain.

A carry lookahead adder circuit was proposed to carry generation network [17]. Prefix trees are used to implement it, resulting in two types of PPA, the KS Adder and the LFA. Verilog HDL was used for design, and the Xilinx Spartan 3E100CP132 was used to implement the code. A modified parallel prefix adder structure was suggested, which performs better than conventional adders and can be widely used in industries to meet desired performance goals. In comparison to the standard Carry Lookahead adder and the conventional tree adders, the proposed adders are effective in terms of latency and area. when the number of bits are less, Ladner Fischer performance is better than the KS adders performance but as the number of bit increases and reaches to a very large number KS becomes faster than Ladner Fischer adder, because of variable fan out of LFA.

The four types of PPA' proposed are KSA, STA, BKA and SKA [18]. Additionally, RCA, CLA and CSA were also investigated. Using the Xilinx Integrated Software Environment (ISE) 13.4 Design Suite, these adders were implemented in verilog HDL. All varieties of PPA's had their area, delay, and power consumption examined. LUTs and IOBs are used to describe the adder design's area. Implementing the adder designs allowed for comparisons of each adder's delay, power, and area.

A 32–bit various PPA was designed and compared based on area, delay and power consumption [19]. In comparison to the prior bit adders used in CPUs, implementation findings show a significant reduction in power and power-delay product. Using the energy recovery logic, such as the power gating technique, to all three adders will minimise the power. With the Xilinx ISE 14.2i tool, all simulation and synthesis results may be noted. KSA, BKA, and LFA are examples of efficient 32-bit parallel prefix adders that were created. The reduction of latency provided by the proposed 32 bit adder addition operation is a significant benefit. The designed adders are also contrasted in terms of power, area usage, and latency for low power VLSI applications. The synthesis findings show that KSA, among the suggested adders, is able to save some power-delay product since it uses less power. But the area delay product is little increased, compared to other adders due to high area consumption.

High speed adders such as BKA, KSA, and LFA are designed and compared [20]. 64-bit adder circuits are used to validate the suggested designs. The proposed designs were assessed in terms of hardware, latency, space, and power consumption. We compared the KSA, BKA, and LFA's slice count, power, and speed. The usefulness of the results is demonstrated by comparing them to current fast-adder designs. The tool, Xilinx ISE 14.7, was used for simulation and synthesis.

The most effective circuit for binary addition from the literature survey is known to be PPA. A lot of study has been done to reduce the complexity of the circuit and the power used by the adder because the final carry is generated prior to the formation of the sum. KS, SKS, spanning tree, and BKS adders are proposed as carry tree adders. Among these, the BKS adder reported to have low power consumption (10.82 mW) and shorter latency (5.96 ns) compared to others. The adders are implemented using Xilink vertex 5 FPGA and their delays are estimated. The BKA has simpler architecture with fewer BCs and GCs than KSA. The implementation of KS PPA is explored using different design techninques, including CMOS and GDI(Gate diffusion Input). Logic gates are designed and simulated using GPDK 180nm technology and CADENCE Design Suite.

IV. COMPARISON

Table 1 shows the comparison of PPA. The Kogge-Stone adder is a parallel prefix adder that offers efficient computation of the sum and carry signals. It achieves a low worst-case delay and has good scalability (29.38 ns). However, it requires a relatively large number of logic gates (2838) and interconnects, leading to a higher area (471) and power consumption compared to some other adders. The Spanning Tree adder is a type of carry-lookahead adder that reduces the number of logic gates (759) and interconnects required

TABLE I
COMPARISON OF PPA

S. No	Method Name	Area			Delay		
		LUT	Slices	Gates	Max Delay	Gate Delay	Path Delay
1	Kogge Stone Adder 64 Bit	471	247	2838	29.638ns	14.730ns	14.908ns
2	Sparse Kogge Stone Adder 64 Bit	248	143	1059	57.425ns	23.616ns	33.809ns
3	Spanning Tree Adder 64 Bit	126	93	759	107.911ns	40.697ns	67.214ns

compared to the Kogge-Stone adder. The Sparse Kogge-Stone adder is a modification of the Kogge-Stone adder that reduces the number of active gates and interconnects, resulting in lower power consumption (248). The reduction in active logic can introduce an additional delay (57.2 ns) and may limit its application in certain scenarios.

V. RESULT ANALYSIS

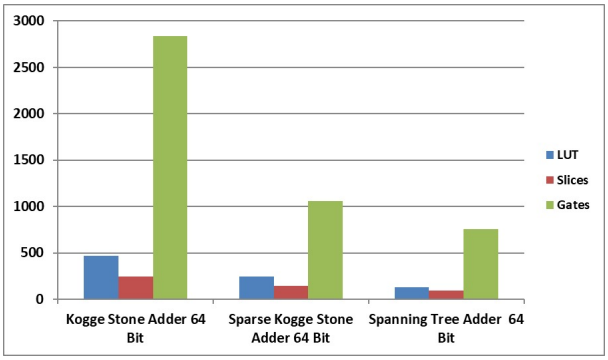

Fig. 2. Area Graph

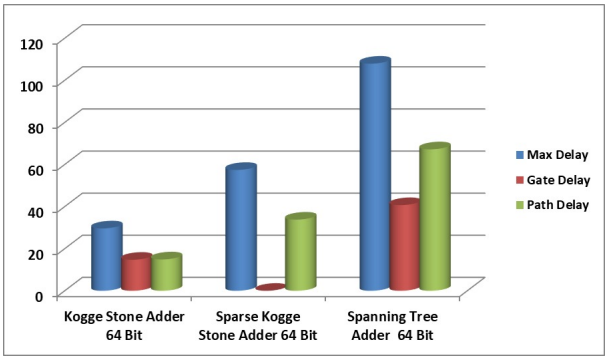

Fig. 3. Delay Graph

Fig. 2 shows the area graph and Fig. 3 shows the delay graph. The Kogge-Stone adder is a parallel prefix adder that offers efficient computation of the sum and carry signals. It achieves a low worst-case delay and has good scalability.The Sparse Kogge-Stone adder is a modification of the Kogge-Stone adder that reduces the number of active gates and interconnects, resulting in lower power consumption. However, reduction in active logic can introduce additional delay and may limit its application in certain scenarios. The Spanning Tree adder is a type of carry-lookahead adder that reduces the number of logic gates and interconnects required compared to the Kogge-Stone adder. It achieves a good compromise between area and performance.

VI. CONCLUSION

In order to determine the most effective adder in terms of delay and power consumption, the paper's main goal is to make a survey of all the details and specifications of the Parallel Prefix Adders (KS, SKS and Spanning Tree adders). The survey results concludes that Spanning Tree adder achieves a good compromise between power consumption and area (Uses 128 LUTs out of 66560) and Kogge Stone Adder achieves a low worst-case delay (29.638 ns) and good scalability.

REFERENCES

[1] B.Chandrika and G.Poorna Krishna, "Implementation and Estimation of Delay, Power and Area for Parallel Prefix Adders", International Journal for Modern Trends in Science and Technology Volume: 02, Issue No: 11, November 2016.

[2] Sudheer kumar yezerla, B Rajendra, "Design and estimation of delay,power and area for parallel prefix adders", Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 - 08 March, 2014.

[3] T.S. Athriya, R. Divya, M. Karthik, A. Manikandan, "Design of Kogge-Stone for fastaddition", International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-5, Issue-4, Apirl-2017.

[4] Avinash Shrivastava, Shefali Churhe, Hemlata Bhagat, Rajni Wamankar, "Design and Estimation of delay, power and area for Parallel prefix adders", Int. Journal of Engineering Research and Application, ISSN : 2248-9622, Vol. 7, Issue 4, pp.01-08, April 2017.

[5] Vanga Mahesh and R. Nirmala Devi, "Design and Characterization of Efficient Parallel Prefix Adders using FPGAs", International Journal of Engineering Research and Technology (IJERT), Vol. 3, Issue 9, pp. 1467-1472, Sept- 2014.

[6] C. N. Shilpa, Kunjan D. Shinde, H. V. Nithin, "Design,Implementation and Comparative Analysis of Kogge Stone Adder using CMOS and GDI design: A VLSI Based Approach", IEEE, pp. 570-574, Dec. 2016.

[7] Yancang Chen, Minlei Zhang, Pei Wei, Sai Sui, Yaxin Zhao and Lunguo Xie, "Implementation of a Parallel Prefix Adder Based on Kogge-Stone Tree", International Conference on Communications, Information Management and Network Security , pp. 234- 237, Aug. 2016.

[8] Shubham and Navdeep Prashar, "VHDL Implementation Of 32-Bit Sparse Kogge-Stone Adder Using Carry Select Logic", International Journal of Technical Research and Applications, ISSN: 2320-8163, Vol. 3, Issue 5, pp. 116-119. Oct. 2015.

[9] Senthil Ganesh R and R. Kalaimathi, "Design and Analysis of Kogge-Stone and Han- Carlson Adders in 130nm CMOS Technology", International Journal of Research, ISSN: 2348- 6848, Vol. 05, Issue 07, pp. 1063-1068, March 2018.

[10] V. Krishna Kumari, Y. Sri Chakrapani and Dr. M. Kamaraju, "Design and Characterization of Koggestone, Sparse Koggestone, Spanning tree and Brentkung Adders", International Journal of Scientific and Engineering Research, ISSN 2229-5518, Vol. 04, pp. 1502-1506, Oct. 2013.

[11] 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, Vol. 02, Issue 06, pp. 467-479, July 2013.

[12] David H.K. Hoe, Chris Martinez and Sri Jyothsna Vundavalli "Design and Characterization of Parallel Prefix Adders using FPGAs", International Journal of Technical Research and Applications, Vol. 02, Issue 05, pp. 1502-1507 July 2011.

[13] Bujjibabu Penumutchi, Satyanarayana vella, Harichndra prasad satti, “Kogge Stone Adder with GDI Technique in 130nm Technology for High performance DSP Applications”, International Journal of Scientific Engineering and Technology Research, Vol. 04, Issue 06, pp.542-551, July 2017.

[14] U. Penchalaiah and V.G. Siva Kumar “Design of High-Speed and Energy-Efficient ParallelPrefix Kogge Stone Adder”, International Journal of Technical Research and Applications, Vol. 02, Issue 06, pp. 567-571, Aug. 2013.

[15] Garnepudi Sai Chand, S. Prabhu Das ”Implementation of High-Performance Spanning Tree Adder using Quaternary Logic”, International Journal of VLSI system Design and Communication systems, ISSN:2322-0929, Vol. 02, Issue.11, pp. 1186- 1193, Dec. 2014.

[16] M.Venkata Durga rama raju, A.Deepthi ”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, Vol. 6, Issue 5 , pp. 41-48, Aug. 2013.

[17] Ravi Payal, Mahima Goel, Prachi Manglik ”Design and Implementation of Parallel PrefixAdder for Improving the Performance of Carry Lookahead Adder”, International Journal of Engineering Research Technology (IJERT) ISSN: 2278-0181, Vol. 4, Issue 12, pp. 566-571, Dec. 2015.

[18] Mr. Deepak Raj, Mrs. Sahana K Adyanthaya, Prof. Praveen J, Prof. Raghavengra Rao ” Design and Implementation of different types of efficient parallel prefix adders” , Vol. 3, Special Issue 1, pp. 172-176, April 2015.

[19] Samraj Daphni, Kasinadar Sundari Vijula Grace ”Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications”, Advances in Science, Technology and Engineering Systems Journal, Vol. 4, pp. 102-106, March 2019.

[20] T.Gowri, Ch. Bala Bharath, M. Anvitha, A. Mohana Rava ”Design and performance analysis of 64-Bit parallel prefix adders for area efficient VLSI applications”, ISSN NO:0377- 9254, Vol 13, Issue 06, pp. 1099-1110, June 2022.