

Hybrid Han Carlson Adder Architecture for Reducing Power and Delay

K. Kaarthik and C. Vivek

Department of Electronics and Communication Engineering,
M. Kumarasamy College of Engineering (Autonomous), Karur, India

Abstract: Adder plays a vital role in Arithmetic Logic Unit. It is commonly employed in building of an IC chip. In the manufacturing of IC chip the Implementation of Adder is used to Reduce the Area by reducing the Number of gates. If the number of gates gets reduced then Power Consumption is also gets reduced simultaneously the processing speed will be Increased by reduction in Delay. These process of decreased Area, Power Consumption, Delay and Increased Speed will be obtained in Low Power VLSI Design process. In this Paper we consider the Parallel Prefix Adder (PPA). The Parallel Prefix Adders are used to Speed up the Binary Addition Process. The work involves the comparison of various parallel prefix adder to obtain a Adder which can able to obtain the Output in short span of time with consumption of less Power and Area. Their outputs are simulated and a Comparison table have been provided to Justify that the Preferred adder can obtain better Result.

Key words: Parallel Prefix Adder (PPA) • Arithmetic Logic Unit • Power • Delay • Area

INTRODUCTION

Computation speeds have multiplied dramatically throughout the past 3 decades ensuing from the event of various technologies. The speed of associate degree mathematical operation could be operating of 2 factors. The primary is that the circuit technology and therefore the second is that the used algorithmic program. What is more, in any technology, logic path delay depends upon several different factors [1] the quantity of gates through that an indication has got to pass before a choice is created, the logic capability of each gate, accumulative distance among all such serial gates, the electrical signal propagation time of the medium per unit distance, etc. as a result of the logic path delay is thanks to the delay internal and external to logic gates, a comprehensive model of performance would ought to embody technology, distance, placement, layout, electrical and logical capabilities of the gates. 1-bit full adder cell is most vital and basic block of arithmetic unit of a system. The activity of energy dissipation and analysis of performance of the system is finished by calculative PDP.

The purpose of this paper is to Represent the comparative analysis of the Parallel prefix adder and to prove that the Han Carlson Adder can be a right choice to

provide the better results. In the Han Carlson adder [2]. How to implement some techniques to bring it to a efficient Adder by the Implementation of Hybrid Architecture.

Existing System: The Parallel Prefix Adder can be a type of adder used to Speed up the binary addition. They undergone the process of Parallel Computation and Parallel Execution [3]. The processing steps involves the Pre processing, Prefix carry tree and Post processing. In which the Two operands of A and B are given as input where the values can be Generated and Propagate to provide the Specific output. The sum bits, S_i are finally obtained from a post processing step represented by Fig. 1.

The pre-processing and post-processing stages remain the same in all parallel prefix adders. It is only how the carry computation takes place within the prefix tree that changes and varies among all trees.

Overview of Parallel Prefix Adders: The Parallel Prefix Adder can be of different types and they can be able to perform different process. They can also Differ in their Logic depth, Fanout requirement, Size, Speed, Power consumption, Power delay Product, LUT slices etc., [4].

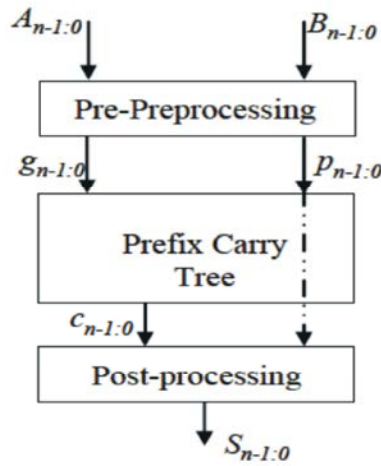


Fig. 1: Parallel Prefix Adder Steps

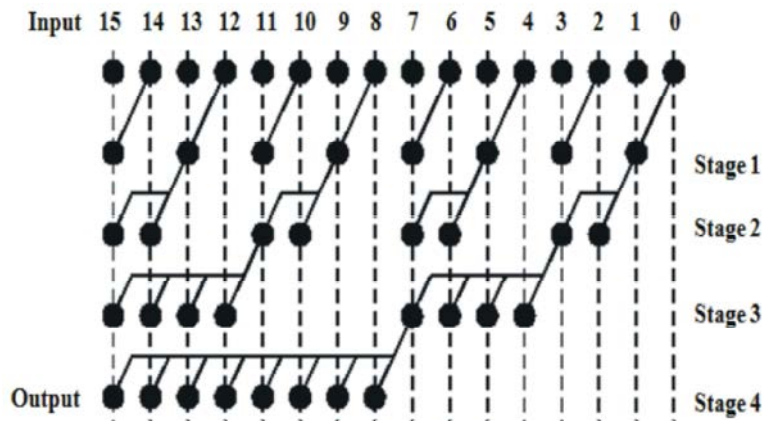


Fig. 2: Ladner-Fischer adder

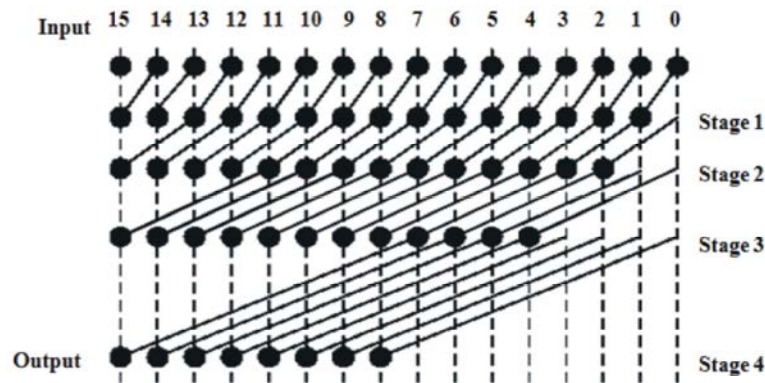


Fig. 3: Kogge-Stone adder

The Parallel Prefix Adders can Be explained as follows. they are, Ladner-Fischer adder: Fig. 2. represents half dozen is that the parallel prefix graph of a Kogge-Stone adder. This adder structure has minimum logic depth, and full binary tree with minimum fan-out, leading to a quick adder however with an oversized space [5].

Kogge-Stone Adder: The Kogge-Stone adder generates carry signals in $O(\log n)$ time, and is taken into account to be the quickest adder. The parallel prefix graph of Kogge-Stone adder is shown in Fig. 3. The high speed of Kogge-Stone adder is attributable to its minimum logic depth and lower fan-out [6]. The most disadvantage of

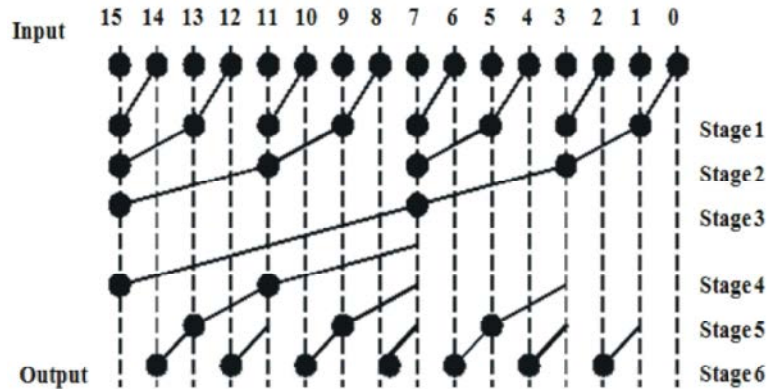


Fig. 4: Brent-Kung adder

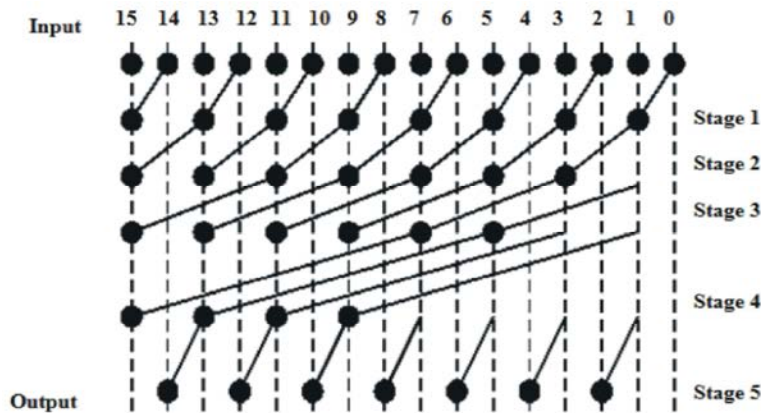


Fig. 5: Han-Carlson adder

Kogge Stone adder is that it occupies massive space and has high wiring congestion. The Kogge-Stone adder generates carry signals in $O(\log n)$ time, and is taken into account to be the quickest adder. The parallel prefix graph of Kogge-Stone adder is shown in Fig. 3. The high speed of Kogge-Stone adder is attributable to its minimum logic depth and lower fan-out [7]. The most disadvantage of Kogge Stone adder is that it occupies massive space and has high wiring congestion. Fig. 3 is that the parallel prefix graph of a Brent-Kung adder. This adder is that the extreme case of most logic depth and minimum space [5].

Brent-Kung Adder: The Brent-Kung adder is one of the most advanced adder designs. Its performance is lower compared to Kogge-Stone adder, but it takes less area to implement and has less wiring congestion [5]. The parallel prefix graph of Brent-Kung adder is shown in Fig. 4

Fig. 4 is that the parallel prefix graph of a Han-Carlson adder. This adder features a hybrid style combining stages from the Brent-Kung and Kogge-Stone adder [8].

Han-Carlson Adder: The Han-Carlson adder is a blend of the Brent-Kung and Kogge-Stone adders [9]. It uses one Brent-Kung stage at the beginning followed by Kogge-Stone stages, terminating with another Brent-Kung stage to compute the odd numbered prefixes. It provides better performance compared to Kogge-Stone for smaller adders [5]. The parallel prefix graph of Han-Carlson adder is shown in Fig. 5.

Table 1: Comparative Study on Prefix adders

Adder Type	Number of Computation Nodes	Logic Depth
Brent-Kung	$[2 * n - 2 - \log_2 n]$	$[(2 * \log_2 n) - 2]$
Kogge-Stone	$[(n * \log_2 n) - n + 1]$	$\log_2 n$
Han-Carlson	$[\frac{n}{2} * (\log_2 n)]$	$[(\log_2 n) + 1]$
Ladner-Fischer	$[\frac{n}{2} * (\log_2 n)]$	$[(\log_2 n) + 1]$

Proposed Architecture: The proposed design will be possible by the techniques called Pipelined design and Folding architecture implemented in the Han Carlson Adder.

Pipelined Design: Pipelining a design will increase its throughput. The improvement through pipelined design is by the use of registers and latency in the circuit. The Complicated design of the combinational circuits and implementation of the Additional registers within some clock cycles can be obtained by using the Pipelined Design [10].

If pipelining is used to perform a long sequence of similar tasks with Less throughput time duration. Moreover, these three points has to be present:

The basic function is repeatedly executed. The basic function must be divisible into independent stages having minimal overlap with each other. The stages must be of similar complexity [11].

Parallel adders will respect the above mentioned conditions. So we can able to convert a parallel adder normally into a pipelined parallel adder. Consider a the 4-bit parallel adder. The adder Performs as follows: For every clock cycle a new input is given to the circuit. Due to the registers the first result can be obtained by taking three clock cycles. The Delay obtained between the first input and the first output is called the latency. Here the Latency is Three clock Cycles [6]. Then for applying each clock cycle, a new result will be obtained at the output. This is process is called the throughput. The throughput of this circuit is one clock cycle plus TCO (the time from one clock cycle to the output of a register).

A straight forward binary tree implementation in presence of folding technique is applied for Parallel Adder. This approach costs a significant amount of area as n inputs require $p = n - 1$ PEs. The Pipelining can be implemented in throughput to reduce area and power. If a layer of PE finish the processing with the help of Classic Binary tree adder. The results are passed on and new calculations can already recommence independently [11].

The idea obtained in this architecture is to fold the adder back onto itself to maximally reuse the PEs. In this process, P is proportional to $n/2$ and the area is cut into half of its original size. So, the inter-connect is also reduced. Similarly in other side, throughput decreases by a factor of $\log_2(n)$ but since the sample rate of different physical phenomena relevant for WSNs does not exceed 100 kHz, this provides enough room to be made tradeoff. The new proposed Folded Tree (FT) topology is depicted in Fig. 6. on the right, which is functionally equivalent to the binary tree on the left [11].

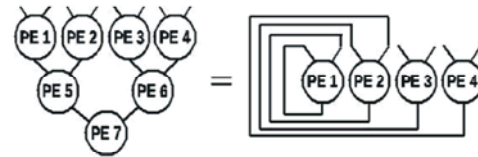


Fig. 6: Binary tree equivalent to proposed folded tree

Folded Tree Architecture: A straight forward binary tree implementation of Blleloch’s approach consumes a reasonable cost and significant amount of area as n inputs require $p = n - 1$ PEs. Inorder to reduce area and power, pipelining process can be traded for throughput, with a classic binary tree. Then if the layer of PEs finishes processing the final results have been obtained on and new calculations can already recommence independently [12].

Folded-tree architecture (FTA) in Parallel Prefix operations for on-the-node data processing in wireless sensor networks [13]. The data locality in hardware reduces area and power consumption in the Proposed Architecture.

Folding Architecture: Folded-tree architecture (FTA) is designed to reduce the Area, Delay and Power consumption and reuse the PE nodes to reduces half of the total area [14].

It limiting the data set by pre-processing with Parallel-Prefix operations. Reuse of the binary tree as a folded tree.

Simulation Environment: The Simulation of various Parallel Prefix Adder designs were carried out with ModelSim tool. All the Parallel Prefix Adder structures were implemented using CMOS logic family. The parameters considered for comparison are power consumption, worst case delay and power-delay product. The various PPA structures were then compared with the number of computation nodes needed for circuit realizations.

RESULTS AND DISCUSSION

The Comparative analysis Various parameters of the Parallel prefix adders are analyzed with the Proposed method as shown in Fig. 7 and the Table 2, 3 and 4 shows the results of that proposed adder.

The Comparison of the various Design parameters such as the operations obtained as per the type of Adder called as Han Carlson Adder and Hybrid Han Carlson Adder can be considered. The number of Usage of Gates may vary according to the type of gates.

Table 2: Comparative Result of Adder Structure

Name of the Adder	Logic Depth	Fan-out Requirement	Size
Ladner Fischer adder	High	Low	Low
Kogge-Stone adder	Low	Low	High
Brent-Kung adder	Low	High	Low
Han-Carlson adder	Low	Low	Low
Proposed Architecture	Low	Low	Low

Table 3: Leakage Power and Dynamic Energy for One PE Under Normal Conditions

Processing Element	Active PE Core	Idle PE Core	PE Instrument Memory
Dynamic energy/ Instrument(pJ)	14.6	4.7	2.10
Leakage power (μ W)	0.03	0.03	0.01
Total Power at 20 MHz (μ W)	41.7	13.5	6.0

Table 4: Comparative Result of Han Carlson adder with Hybrid Han Carlson Adder

Design Parameters	Han Carlson adder (N=32)	Hybrid Han Carlson adder (N=32)
Prefix Operations	80	63
No. of Gates	940	645
Delay(ns)	0.68	0.42
Dynamic Power (mW)	445.98	405.23
Leakage Power (mW)	7.45	5.79

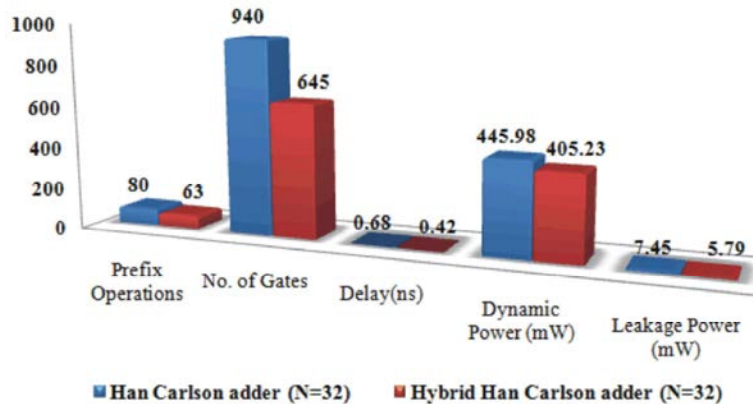


Fig. 7: Comparative Result of Han Carlson adder with Hybrid Han Carlson Adder

In the Normal Han Carlson Adder the Number of Gate Usage can be High when compared with the Hybrid Han Carlson Adder. If the Number of Gate usage is reduced then the Power consumption and the Delay can also reduced and the results can be listed in the Table 4 and they can be Dramatically represented in the Chart as shown in Fig. 7.

CONCLUSION

From the above discussed results table IV clearly shows that the Proposed Han Carlson adder will provide the better results with reduced complexity and Hence it can be used in the construction of Various VLSI architecture where better result is required with good speed and Low power and Area are Required.

ACKNOWLEDGMENT

Our thanks to M.Kumarasamy college of Engineering for offering us the opportunity to do this wonderful project, and to Dr. V. Kavitha , Principal and The HOD Prof. A. Sri Devi, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project, especially in writing this paper.

REFERENCES

1. Nesenbergs, M. and V.O. Mowery, 1959. Logic synthesis of high speed digital comparators, Bell System Technical Journal, 38: 19-44.
2. Deepa Yagain, A. Vijaya Krishna and Akansha Baliga, 2012. Design of High-Speed Adders for Efficient Digital Design Blocks.

3. Sreenivaas Muthyala Sudhakar, Kumar P. Chidambaram and Earl E. Swartzlander Jr., 2012. Hybrid Han-Carlson Adder, The University of Texas at Austin.
4. Harris, D., 2003. A Taxonomy of Parallel Prefix Networks, Proc. 37th Asilomar Conf. Signals Systems and Computers, pp: 2213-7.
5. Ramanathan, P. and P.T. Vanathi, 2009. Hybrid Prefix Adder Architecture for Minimizing the Power Delay Product, World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering in 3(4).
6. Deepa, Yagain and A. Vijaya Krishna, 2011. High Speed Digital Filter Design using register Minimization Timing & Parallel Prefix Adders.
7. Kogge, P.M. and H.S. Stone, 1973. A parallel algorithm for the efficient solution of a general class of recurrence equations, IEEE Trans. Comput., C-22(8): 786-793.
8. Giorgos Dimitrakopoulos and Dimitrios Nikolos, 2005. High-Speed Parallel-Prefix VLSI Ling Adders, IEEE Trans. On Computer, 54(2).
9. Han, T. and D.A. Carlson, 1987. Fast area-efficient VLSI adders, Proc.IEEE 8th Symp. Comput. Arith. (ARITH), May 18-21, pp: 49-56.
10. Avinash shrivastava, Chandrasahasahu, 2015. Performance Analysis of Parallel Prefix Adder Based on FPGA, International Journal of Engineering Trends and Technology (IJETT), 21(6).
11. Sandhya, R. and B. Sanjai Prasada Rao, 2015. Optimized Analytical Approach For Wireless Sensory Nodes Based On Low Power DSP Architecture, International Journal of Engineering And Computer Science ISSN: 2319-7242, 4(9): 14298-14306
12. Tejasvi, J. and B. Bhavani, 2015. Design and Implementation of Folded & Un-folded Tree Architectures for Processing Unit in Wireless Sensor Nodes, ISSN 2319- 8885, 04(25): 4895-4899.
13. Swapna K. Gedam and Pravin P. Zode, 2014. Parallel Prefix Han-Carlson Adder, International Journal of Research in Engineering and Applied Sciences of ISSN, 02(02).
14. Ranjithkumar, K. and T.R.V. Anandharajan, 2015. Performance Optimization approach for adder in Folding Tree Architecture, International Journal of VLSI and Embedded Systems, 06, Article 06598.