A Comparative Analysis of Parallel Prefix Adders

Megha Talsania and Eugene John
Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, TX 78249
megha.talsania@gmail.com, eugene.john@utsa.edu

***Abstract*- All modern processors, including general purpose microprocessors, digital signal processors and GPUs contain an Arithmetic Logic Unit (ALU). The computing efficiency of modern processors mainly depends of the efficiency of the ALU. An adder is the basic building block for an ALU which performs arithmetic as well as logic operations. This paper investigates the performance of six different parallel prefix adders implemented using four different TSMC technology nodes. The parallel prefix adders investigated in this paper are: *Kogge Stone Adder, Brent Kung Adder, Han Carlson Adder, Sklansky Adder, Lander Fischer Adder, and Knowles Adder.* The performance metrics considered for the analysis of the adders are: power, delay and area. Simulation studies are carried out for 16, 32 and 64 bit input data width.**

Keywords- Prefix tree adder, High speed CMOS adder, Low Power VLSI

1. Introduction

The addition of two binary numbers is one of the most fundamental and important arithmetic function in modern digital systems such as microprocessors and digital signal processors. In these systems binary adders are used in arithmetic logic units (ALU), multipliers, dividers and memory address generation. The requirements of adders are that it should be fast and efficient in terms of power and chip area. Most often, the maximum operating speed of most of the modern digital systems depend on how fast adders can process the data and hence responsible for setting the minimum clock cycle time in processors.

The major problem for binary addition is the propagation delay in the carry chain. As the width of the input operand increases, the length of the carry chain increases. To address the carry propagation problem, most of the modern adder architectures are represented as a parallel prefix adder (PPA) structure consisting of pre-processing, carry look-ahead and post processing sections. Parallel Prefix Adders have been established as the most efficient circuits for binary addition in digital systems. Their regular structure and fast performance makes them particularly attractive for VLSI implementation. The delay of a parallel prefix adder is directly proportional to the number of levels in the carry propagation stage.

This paper investigates the performance of six different parallel prefix adders implemented using four different TSMC technology nodes. The parallel prefix adders investigated in this paper are: *Kogge Stone Adder, Brent Kung Adder, Han Carlson Adder, Sklansky Adder, Lander Fischer Adder, and Knowles Adder.* The performance metrics considered for the analysis of the adders are: power, delay and area. In this paper the CMOS adders were realized using TSMC 130nm, 90nm, 65nm and 40 nm technologies. For performance comparison, the adders were realized using various prefix tree algorithms. Using simulation studies, delay, area and power performance of the various adder modules were obtained. It was observed that Kogge Stone Prefix tree adder has better circuit characteristics in terms of delay compared to adders realized using other algorithms.

The rest of the paper is organized as follows: in Section 2 a brief description of all the six different parallel prefix adders are given, in Section 3, the tools and methodology used for the research is explained. Section 4 gives results and performance analysis and finally Section 5 gives conclusions.

2. Parallel Prefix Adders

In this section the six different parallel prefix adders that are investigated in this paper are briefly described.

2.1 Kogge Stone Adder

The schematic of Kogge Stone Adder is given in Figure 1 [8]. It is widely used in high performance applications. The general concept of Kogge Stone adder is almost the same as that of the carry look ahead adder except for the second step, called parallel carry prefix chain. In the first level (L=1), generates and propagates of 2-bit are computed at the same time. In the second level (k=2), generates and propagates of 4-bit are calculated by using the result of 2-bit in level 1. Therefore, the actual carry-out value of the 4th bit would be available while the calculations in level 2 are being computed. In the third level (L=3), the carry-out of the 8th bit is computed by using the 4th bit carry result. The same method adopted in level 3 is applied to get carry-out values of the 16th bit and the 32nd bit in level 4 and level 5. All other carries of bit are also computed in parallel. In Figure 1, red boxes are propagate (P) and generate (G) generators for each bit of two inputs. Yellow boxes contain propagate block and generate block and the delay of one yellow box is equal to two gate delay (D). The blue boxes keep the original generate value transmitted from the previous level. In each level, because all carries are calculated in parallel, the delay is the running time of single yellow box.

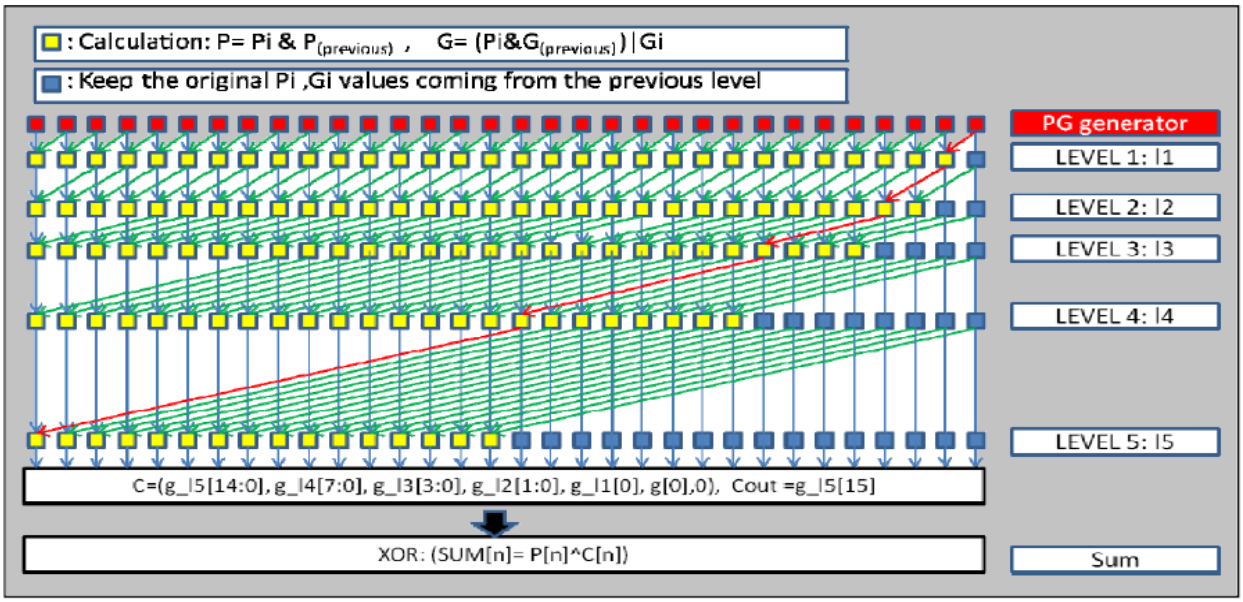

Figure 1:- Schematic of 32 bit Kogge Stone Adder

2.2 Brent Kung Adder

Figure 2 gives the schematic of the Brent Kung Adder. Brent-Kung is a parallel prefix form of the carry look ahead adder. In carry lookahead adder, as the size of the input operands is increased the delay of the result is also increased. Therefore the idea here is to have a gate level depth of O(log2(n)). It takes less area to implement than the other prefix adders such as Kogge-Stone adder and it also has less wiring congestion. Instead of using a carry chain to calculate the output, the method shown in Figure 2 is used. This will reduce the delay without compromising the power performance of the adder.

.

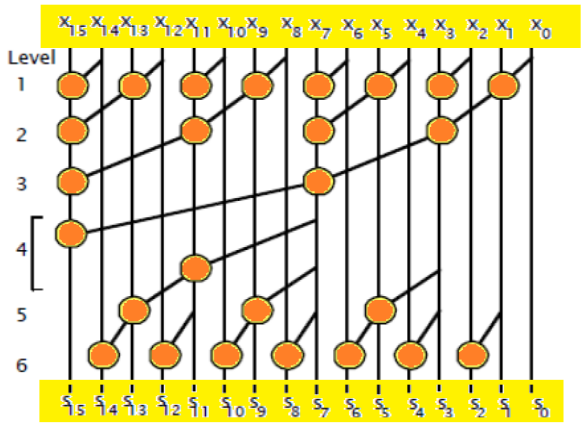

Figure 2:- Schematic of 16-bit Brent Kung Adder

2.3 Han Carlson Adder

Han Carlson adder is a parallel prefix tree. It helps to reduce complexity in Brent Kung adder [7]. It is also a hybrid design combined stages of Brent Kung and Kogge Stone adder. This scheme performs carry-merge operations on even bits only. 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 [15]. Thus, the reduced complexity is at the cost of adding an additional stage to its carry-merge path. Figure 3 represents method of 16 bit Han Carlson Adder [15].

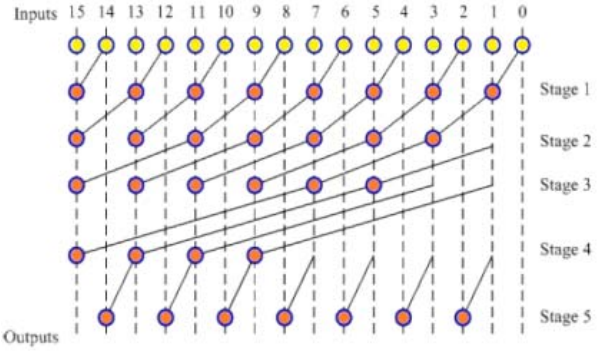

Figure 3: Schematic 16-bit Han Carlson Adder

2.4 Sklansky Adder

Sklansky Adder [7] is another form of parallel prefix adder and its schematic is shown in Figure 7. 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].

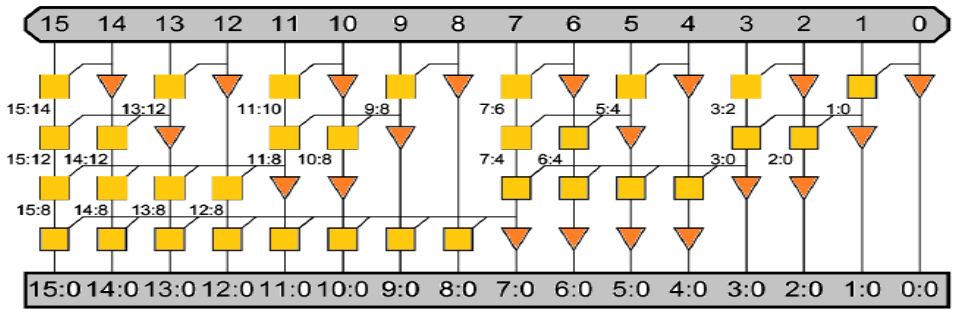

Figure 4: Schematic of 16-bit Sklansky Adder

2.5 Lander Fisher Adder

The schematic of Lander Fischer Adder [9] is shown in Figure 5. This adder structure has minimum logic depth, but has large fan-out requirement up to n/2 [9].

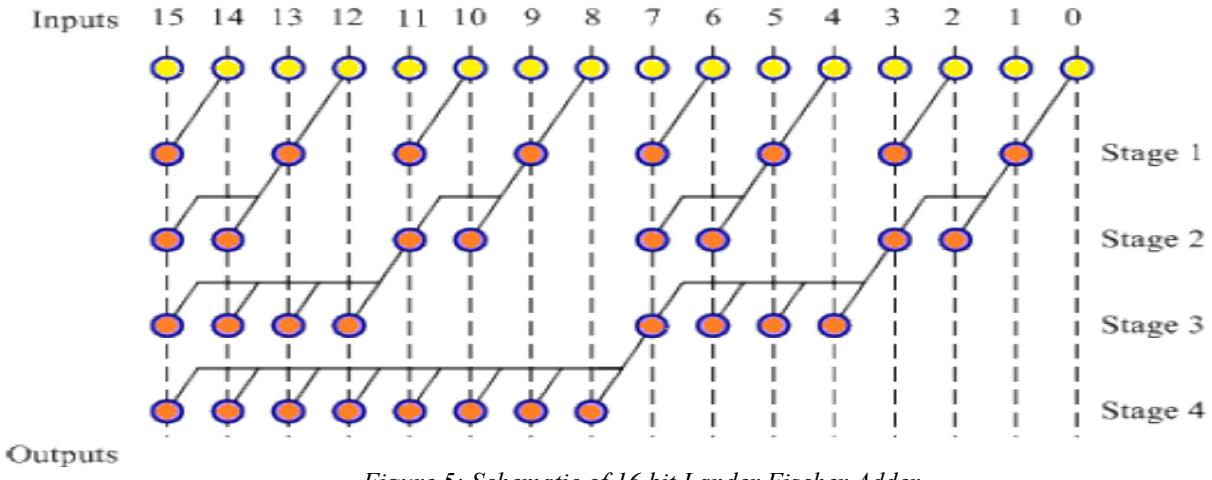

Figure 5: Schematic of 16 bit Lander Fischer Adder

2.6 Knowles Adder

Knowles adder is similar to Kogge Stone Adder, but it has different logic to calculate the output. Figure 6 illustrates a 16-bit Knowles Adder [8].

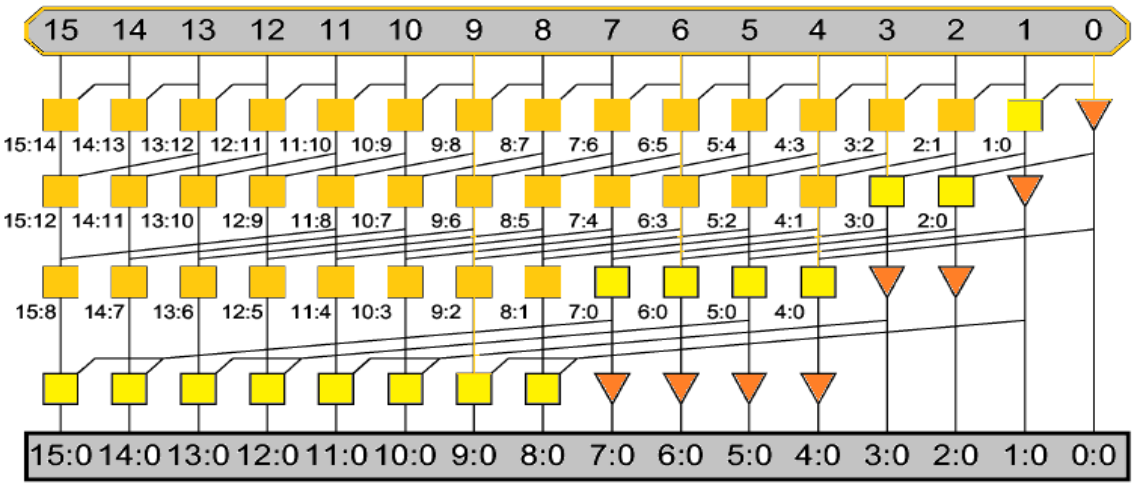

Figure 6: Schematic of 16-bit Knowles Adder

3. Tools and Methodology

For the the VLSI implementation of all the parallel prefix adders investigated in this research the tools used are ModelSim and Cadence Encounter. The technology used for this research are the TSMC 130nm process (TCBN130GHPBC), TSMC 90m process, TSMC 65nm process (TCBN65LPBWP7T) and TSMC 40nm process (TCBN40LPBWP). The simulations were carried out to obtain the power, area and the worst case delay of all the six different parallel prefix adders. The designs of each adder were generated by creating Verilog source file using ModelSim. Then each design was synthesized to simulate the functional result for functionality verification. After that each design is verified using VCS. After confirming the accuracy of each design, the source was used to create the netlist for schematic circuit diagram with Cadence Encounter. Finally, the performance evaluation for each design was calculated using TSMC 130nm, 90nm, 65nm and 40nm process libraries.

4. Results and Performance Analysis

A. Schematic and Layout Synthesis

All the six different parallel prefix adders were synthesized using Cadence Encounter using TSMC 130nm, 90nm, 65nm and 40nm technology nodes. In this section the synthesis results of 32 bit Brent Kung Adder and 64-bit Kogge Stone Adder are presented in Figures 7 and 8 respectively.

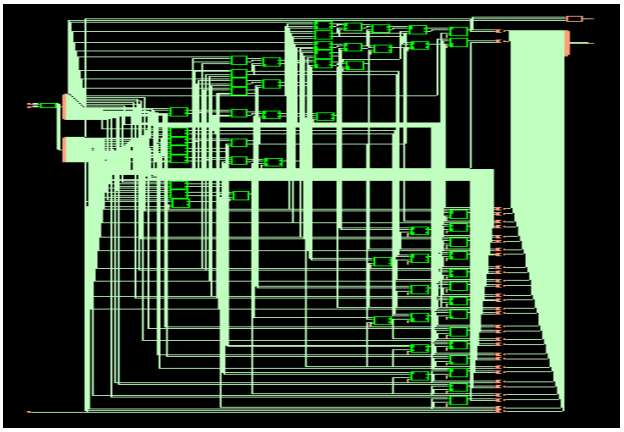

Figure 7: Schematic of 32 bit Brent Kung Adder

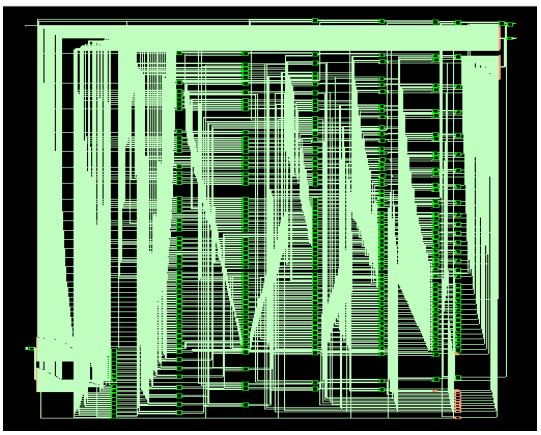

Figure 8: Schematic of 64 bit Kogge Stone Adder

Layout of low power adders were generated to analyze the area of the different parallel prefix adders using cadence encounter tool for TSMC 90nm technology node. Figure 9 illustrates the layout of 64 bit Kogge Stone Adder (KSA).

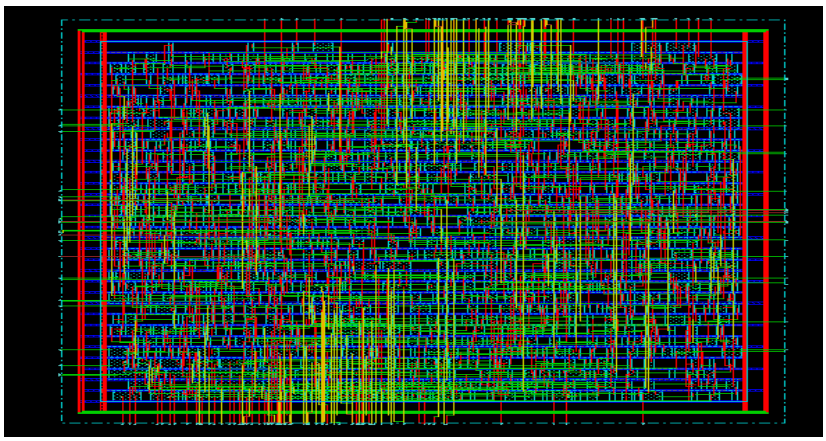

Figure 9: Layout of 64 bit KSA for TSMC 90nm technology node

B. Performance Analysis of Various Adders

The delay, power and area of all the six different prefix adders were investigated using simulations for varying input bit width for TSMC 130nm, 90nm, 65nm and 40nm technology nodes. Table 1 presents the area, power and delay results for all the six parallel prefix adders for 16 bit input width for 90nm technology node. From the results presented in Table 1 it can be inferred that 16 bit Brent Kung Adder occupies less area and power compared to any other adder investigated. But, in terms of delay it has the worst performance among all the adders compared in this study. Speed and Power are difficult characteristics to balance. As expected the 16-bit Kogge Stone Adder is the fastest adder among all the adders investigated together with Knowles Adder. This is one of the reasons why Kogge Stone Adder has been used in high speed applications.

Table 1: Comparison of area, power and delay for 16 bit input data width (90nm TSMC)

Adder	Area(µm2)	Total Power (µW)	Delay(ps)
Brent Kung	329.83	20.93	522
Han Carlson	366.05	22.48	444
Knowles	502.15	27.35	428.8
Lander Fischer	335.87	21.17	458.6
Sklansky	366.05	22.60	429.6
Kogge Stone	502.16	27.35	428.8

Table 2 presents the area, power and delay results for all the six parallel prefix adders for 64 bit input width for 90nm technology node. In this case also, Kogge Stone Adder has the best delay performance and the Brent Kung Adder has the best area and power performance.

Table 2: Comparison of area, power and delay for 64 bit input data width (90nm TSMC)

Adder	Area(µm2)	Total Power (µW)	Delay(ps)
Brent Kung	1354.99	85.08	825.9
Han Carlson	1832.34	102.51	593.9
Knowles	2345.57	127.64	565.7
Lander Fischer	1512.49	90.21	676.8
Sklansky	1681.52	99.58	831.8
Kogge Stone	2669.91	137.22	561.5

Figures 10 - 13 presents comparison of all the parallel prefix adders investigated in this research in terms of total power consumption and area using four different TSMC technologies. Figures 10 and 11 are for 32 bit adders and Figures 12 and 13 are for 64 bit adders. From the data presented in Tables 1 and 2 it can be seen that the Kogge Stone adders are the fastest among all the adders compared in this research. But from the simulation results presented in Figures 10 – 13 it can be clearly inferred that in terms of power and area performance Kogge Stone Adders are not among the best. The Brent Kung Adder exhibit the best performance in terms of area and power consumption for both the 32 and 64 bit adder categories.

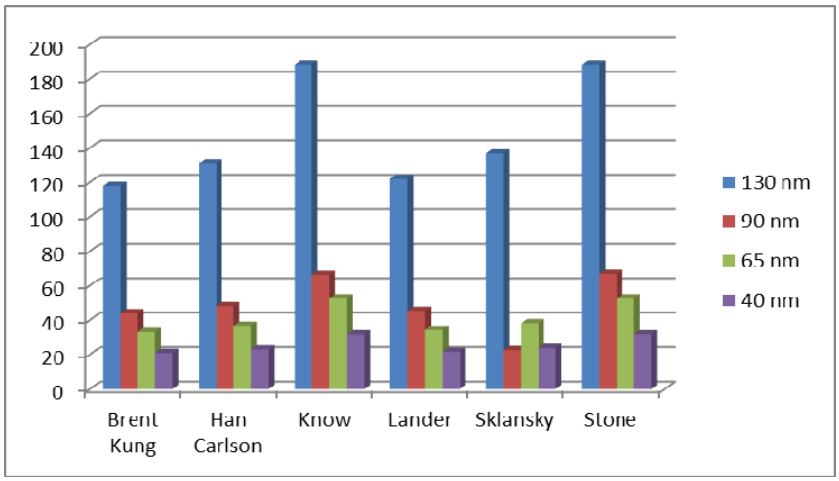

Figure 10: Total power consumption (µW) for various 32 bit adders

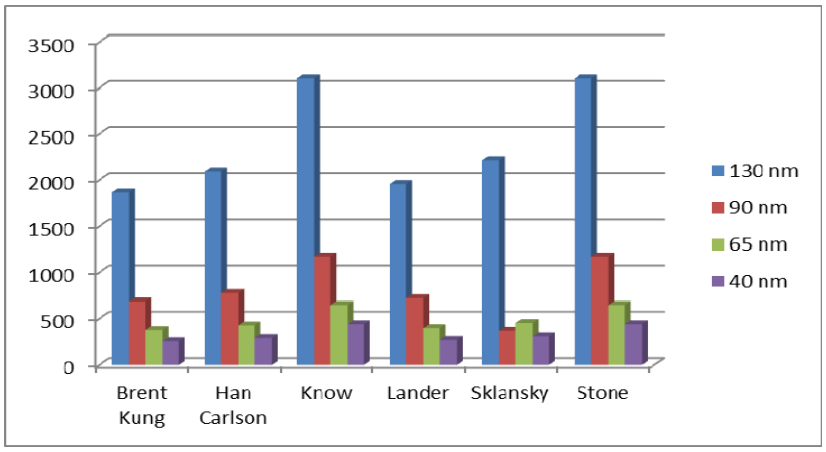

Figure 11: Total area (µm2) for various 32 bit adders

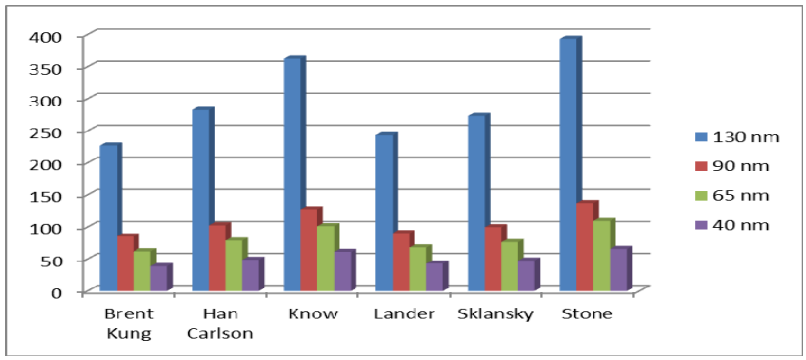

Figure 12: Total power consumption (µW) for 64 bit adders

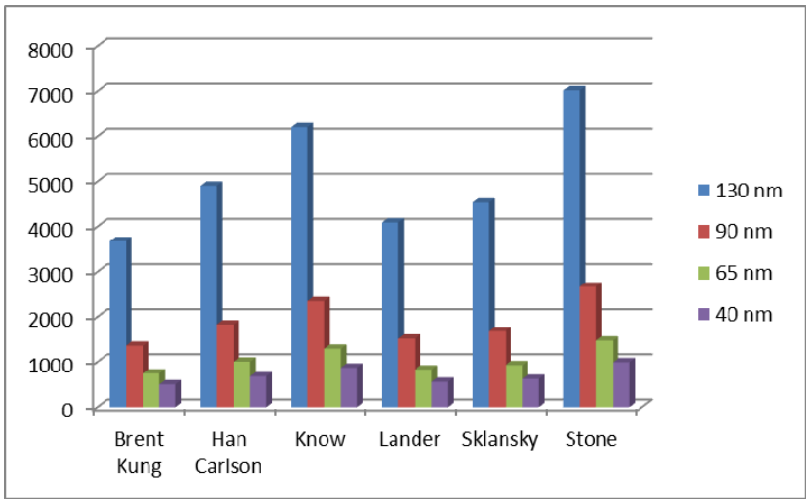

Figure 13: Total area (µm2) for various 64 bit adders

Figure 14 shows the delay comparison for all the adders for the TSMC 90 nm technology node for 16, 32 and 64 bit input data width. As expected, Kogge Stone Adder has the best performance among all the adders for all the input data width considered.

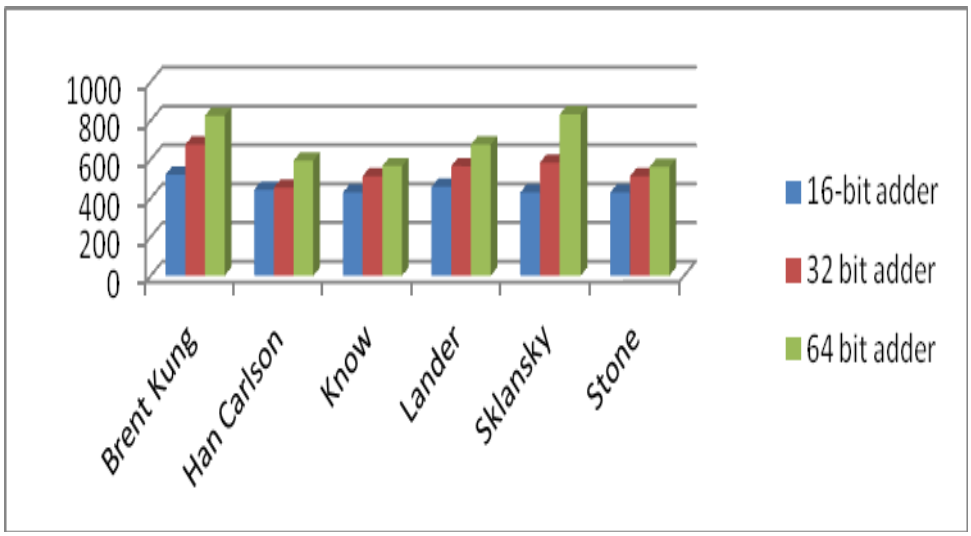

Figure 14: Delay (ps)
Comparison between all adders (90nm TSMC)

5. Conclusions

The primary objective of this research is the design, realization and performance comparison of various parallel prefix adders. In this paper several adders were analyzed to identify the optimal adder modules that can be used for the realization of high speed or low power adder structures. The addition algorithms that were studied include six different types of prefix tree adders. The performance analysis was based on the silicon area required for the implementation of the algorithm in hardware, the power dissipation during computation, and the worst case delay in performing the operation. In this research the CMOS adders were realized using TSMC 130nm, 90nm, 65nm and 40 nm technologies. Based on our simulation studies the Kogge Stone Adder has the best performance among all the adders for all the input data width considered. It is widely used in high performance applications and it has the merits of uniform structure and balanced loading in each internal node to get high speed performance. The Brent Kung Adder exhibited the best performance in terms of area and power consumption for all the input data width considered.

Acknowledgement: Research reported in this paper was supported by National Institute of General Medical Sciences of the National Institutes of Health under award number 1SC3GM096937-01A1. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] K. Rawwat, T. Darwish, and M. Bayoumi, ".A low power carry select adder with reduces area", Proc. Of Midwest Symposium on Circuits and Systems, pp. 218- 221, 2001.

[2] A. Tyagi, "A reduced area scheme for carry-select adders", IEEE Trans. on Computer, vol. 42, pp. 1163-1170, 1993

[3] Padma Devi, Ashima Girdher, and Balwinder Singh, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International Journal of Computer Applications (0975 – 8887)Volume 3 – No.4, June 2010.

[4] S.-L. Lu, "Speeding Up Processing with Approximation Circuits," Computer, vol. 37, no. 3, pp. 67-73, 2004

[5] Gurkayna, F.K. , "Higher radix Kogge-Stone parallel prefix adder architectures ", 2000

[6] D. Harris, R. F. Sproull, I. E. Sutherland, "Logical Effort: Designing Fast CMOS Circuits" ,M. Kaufmann, 1999.

[7] P.Sunitha, "A Novel Approach For Designing A Low Power Parallel Prefix Adders", Vol. 1 Issue 8, October – 2012

[8] P. Kogge et al., IEEE Trans. Computers, vol. C-22, p. 786 (1973).

[9] D. Harris, "A Taxonomy of Parallel Prefix Networks," in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.

[10] Padma Devi, Ashima Girdher, and Balwinder Singh, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International Journal of Computer Applications (0975 – 8887)Volume 3 – No.4, June 2010.

[11] S.-L. Lu, "Speeding Up Processing with Approximation Circuits," Computer, vol. 37, no. 3, pp. 67-73, 2004.

[12] Rabaey, J. M., "Digital Integrated Circuits: A Design perspective", New Jersey, Prentice-Hall, 1996.

[13] N. Weste and K. Eshragian, "Principles of CMOS VLSI Designs: A System Perspective", 2nd ed., Addison-Wesley, 1985-1993.

[14] B. Parhami, "Computer Arithmetic, Algorithm and Hardware Design", Oxford University Press, New York, pp.91-119, 2000.

[15] Sudhakar, S.M.,"Hybrid Han Carlson adder",2012