International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

Design and Analysis of Kogge-Stone and Han-Carlson Adders in 130nm CMOS Technology

Senthil Ganesh R & R. Kalaimathi

[1]*Assistant Professor, Electronics and Communication Engineering, Info Institute of Engineering, Coimbatore, Tamilnadu, India*

[2]*PG Scholar, M.E VLSI Design, Info Institute of Engineering, Coimbatore, Tamilnadu, India*

Abstract — *Adders find applications in Arithmetic and Logic Units (ALUs), microprocessors, DSP systems and ASIC systems. An efficient adder can be of greater assistance in designing of any arithmetic circuits. The various types of adder design are available each with their own advantages and disadvantages. In this paper, we proposed the comparative analysis between the 8-bit Kogge-Stone and Han-Carlson parallel prefix adders. Parallel prefix computation algorithm is employed to improve the speed of the addition. Parallel Prefix Adders (PPA) are extensively used to make a wide range of design tradeoffs in terms of area, delay and power. In this paper, adders are designed and simulated in 130 nm technology using Tanner EDA tool.*

Keywords- *Parallel Prefix Adders, Han-Carlson Adder, Kogge-Stone Adder, Area, Power and Delay*

I. INTRODUCTION

Floating Point (FP) computation has been widely used in many areas such as scientific computation, graphics, digital signal processing, image processing and other applications. Advantage of using FP representation over fixed point representation is due to the fact it support much wider range of values and it is based on scientific notation. The scientific notation is nothing but the way of representing very large or very small numbers in a compact form so that it can be easily used for computing process. The multiplication operation of the FP numbers is greatly affected by multiplier design and its methodology.

Binary addition is the most important arithmetic function in VLSI systems. Binary adders are the one of the most important elements in the processor chips, ALUs, program counter, memory addresses, etc. Ripple Carry adder is one of the first and most fundamental adder performs binary addition. Its speed is proportional to the length of its input operands and therefore delay is high for large size of operands. Later, many conventional adders design were developed and each comes out with their own merits and limitations. Of these, Parallel Prefix Adders (PPA) provides a good results compared to the conventional adders.

In VLSI systems, large complex logic gates adders will be too slow. Therefore, the design is modularized in such a way that breaking the structure into trees of smaller and faster adders. This structure can be implemented easily and effectively for larger bit size. The delay of passing the carry through the carry look-ahead stages are very high for large bit size adders and therefore parallel prefix adders are used. In addition, any decrease in delay directly relate to an increase in the throughput [4]. In nanometer range, it is important to develop an addition algorithm that consumes low area, low power and high performance in operation. The Parallel Prefix adders are suitable for VLSI implementation because of due to its simple cell structure and regular connection between the cells. Each prefix can be defined in terms of logic levels, fan-out and wiring tracks.

The organization of this document is as follows. Section 2 explains about the Parallel Prefix Adders (PPA), Section 3 and 4 details out Kogge-Stone adder [1] and Han-Carlson adder [2] with the block diagram and gate-level diagram. Section 5 shows the comparison of the result in terms of area, delay, power, PDP and EDP between the mentioned two parallel prefix adders. Finally, section 6 is the conclusion of the paper based on the comparison results.

II. PARALLEL PREFIX ADDERS

PPA is a family of adders derived from the carry look ahead adders and these adders are suitable for wider word lengths. PPA circuit uses a tree like structure which reduces the latency to O[log2 (n)] where ‘n’ represents the number of bits [4]. Various types of PPA are available and in this paper 8-bit Kogge-Stone and Han-Carlson adders are discussed. The structure of the PPA is shown in the Fig 1 [7].

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

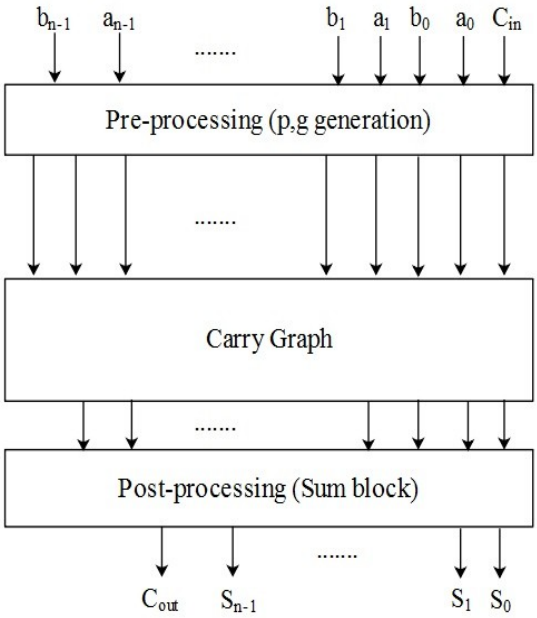

Fig 1: PPA Structure Diagram

2.1 Pre-processing Stage

In this stage, creation of generate and propagate signals are carried out using the logic equations (1) and (2) respectively [2] [8].

$$g_i = a_i\, b_i \quad (1)$$

$$p_i = a_i \oplus b_i \quad (2)$$

where 'g_i' represents the generate signals and 'p_i' represents the propagate signals.

2.2 Prefix-Computation Stage

In this stage, PPA construction depends on the generation of group carry propagate and group generate signals. This stage is the more important in tree adders and consists of three major components are black cell, grey cell and buffers shown in the Fig 2[2] [8]. Black cells are used to compute the both group generate and group propagate signals. Grey cells are used to compute only generate signals which are needed in the post-processing stage. Group propagate and group generate signals are defined by the logic equations (3) and (4) respectively [2] [8].

$$G_{[i:k]} = \begin{cases} g_i, & \text{if } i = k \\ G_{[i:j]} + P_{[i:j]}\, G_{[j-1:k]} & \text{otherwise} \end{cases} \quad (3)$$

$$P_{[i:k]} = \begin{cases} p_i, & \text{if } i = k \\ P_{[i:j]}\, P_{[j-1:k]} & \text{otherwise} \end{cases} \quad (4)$$

2.3 Post-processing Stage

This is the final stage where computation of sum and carry bits for each individual operand bits takes place. The logic equations for the sum and carry calculation is given by the equations (5) and (6) respectively [2] [8].

$$C_i = G_{[i:0]} \quad (5)$$

$$S_i = p_i \oplus C_{i-1} \quad (6)$$

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

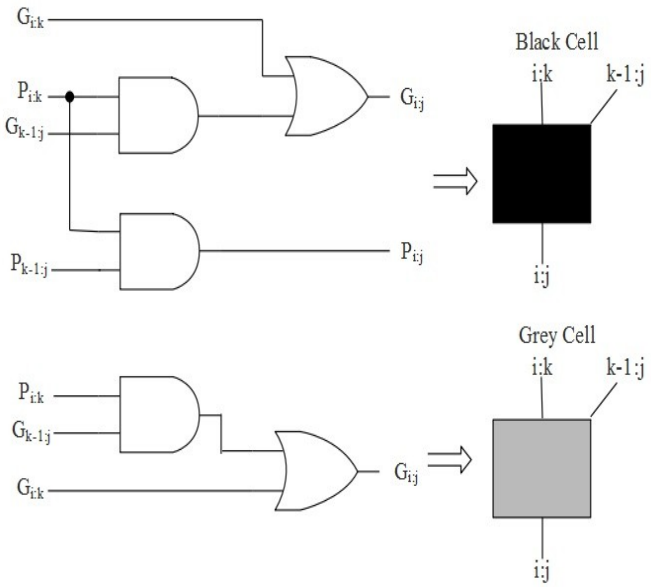

Fig 2: Gate-level diagram of Black and Grey cells

III. KOGGE-STONE ADDER

The Kogge-Stone adder algorithm was proposed by Kogge and Stone (1973) has both low fan-out and optimal depth but produces complex circuit and therefore large number of interconnects. Kogge-Stone structure is very useful for high speed applications. The delay of this structure is given by [$\log_2$ (n)] and the computational nodes is given by [n $\log_2$ (n) –n + 1]. This structure addresses the problem of fan-out by recursive doubling algorithm concept. Kogge-Stone adder has regular layout as well [1].

The block diagram of 8-bit Kogge-Stone adder is shown in the Fig 3[6]. It has four stages of prefix graph and each stage is designed with black and grey cells. The gate level diagram of 8-bit Kogge-Stone adder is shown in the Fig 4 [1].

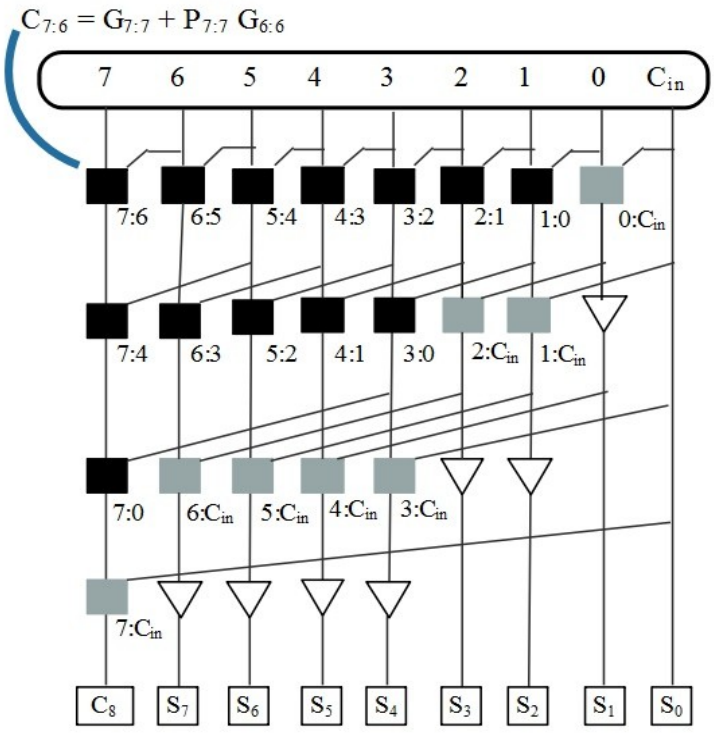

Fig 3: Block diagram of 8-bit Kogge-Stone adder

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

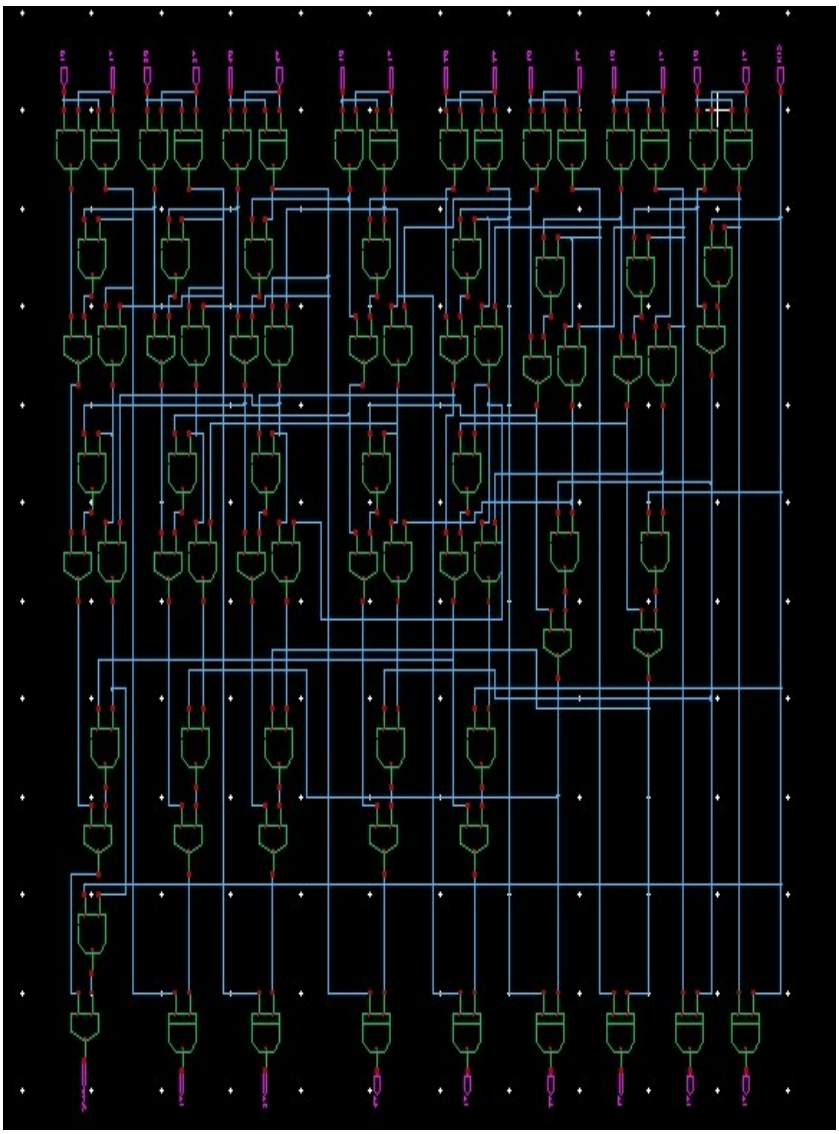

Fig 4: Gate-level diagram of 8-bit Kogge-Stone adder

IV. HAN-CARLSON ADDER

The Han-Carlson adder was proposed by Han and Carlson (1987) to achieve a trade-off between logic depth, number of computation nodes and interconnects. Han-Carlson trees are a family of networks between Kogge-Stone and Brent-Kung adders. The outer rows of this adder perform Brent-Kung addition methodology while the inner rows additions are performed by Kogge-Stone methodology. Han-Carlson prefix trees uses less cells and wire tracks than Kogge-Stone but with the cost of extra logic level of carry merge for calculating missing carries [2].

The block diagram of 8-bit Han-Carlson adder is shown in the Fig 5 [2]. It uses Brent-Kung at the beginning and at the end of the prefix graph and the number logic level is defined as $[\log_2 (n) + 1]$ where 'n' represents the number of bits. The black and grey cells are placed at the odd bit positions in the initial stage and the middle stages. Only grey cells are placed at the even bit positions in the final stage where carry merge operation happens. The delay of this structure is $[\log_2 (n) +1]$ and the hardware complexity is given by $[(n/2) \log_2 (n)]$ [3]. Han-Carlson gives good trade-off between fan-out, number of logic cells and the number of black cells compared to Kogge-Stone. The gate-level diagram of this adder is shown in the Fig 6 [7].

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

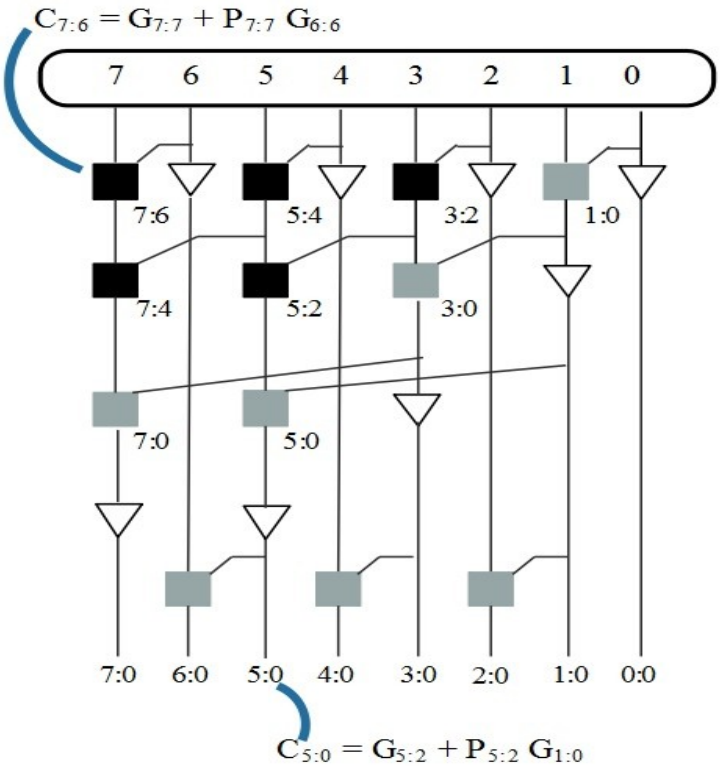

Fig 5: Block diagram of 8-bit Han-Carlson adder

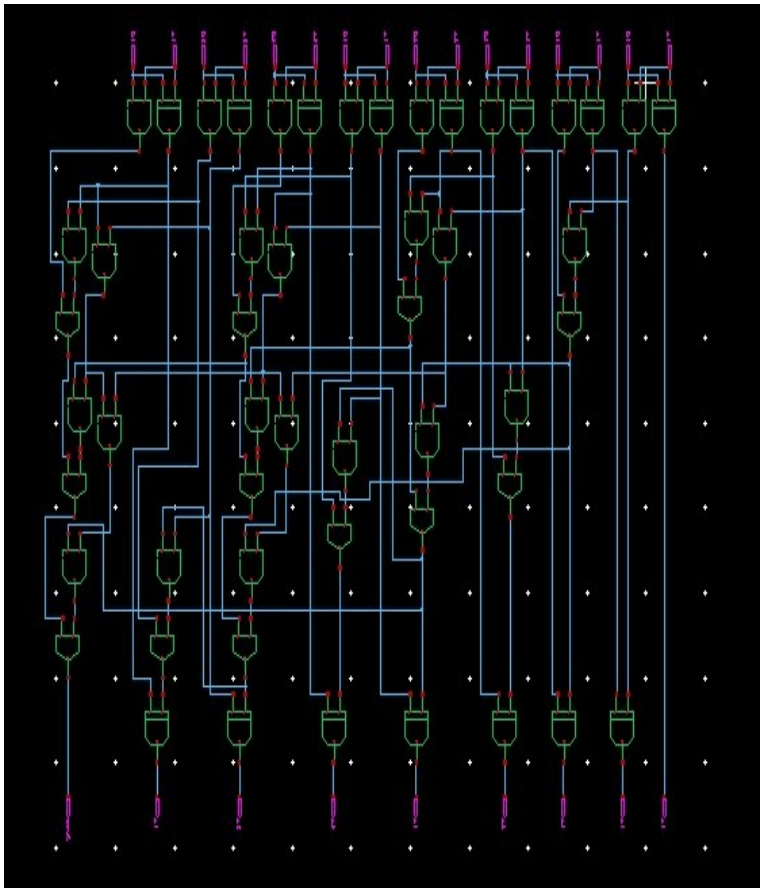

Fig 6: Gate-level diagram of 8-bit Han-Carlson adder

V. COMPARISON RESULTS

Table 1 gives the comparison of the various parameters for 8-bit KSA and HCA. It is designed and simulated using Tanner tool of 130nm technology.

- PDP is calculated from power and delay, unit is 'fJ' where f denotes 'femto'. PDP = (Power * Delay) [5].
- EDP is calculated from PDP and delay, unit is 'yJs' where y denotes 'yocto'. EDP = (PDP * Delay) [5].

Table 1. Comparison of 8-bit Parallel Prefix Adders

Parameters	Kogge-Stone Adder	Han-Carlson Adder
Transistor Count	564	402
Area (μm^2)	0.352	0.251
Delay (ns)	9.3	9.1
Power (nW)	155.76	114.15
PDP (fJ)	1.4486	1.0388
EDP (yJs)	13.472	9.4528

VI. CONCLUSION

In terms of cost or area and power between the two PPAs, the Han-Carlson adder is the best choice. The Han-Carlson adder's area or power rises as the bit size increases but not rise as drastically compared to Kogge-Stone adder. Because, Han-Carlson adder gives good trade-off between fan-out and the number of logic cells. In terms of the time propagation delay, Kogge-Stone adder is the better choice. Kogge-Stone has very low propagation delay. Only for the bit size less than 16 bits Kogge-Stone has longer propagation delay time. Therefore, only for fast addition Kogge-Stone adder is used but for better area usability, low power consumption and high speed Han-Carlson adder is employed. This paper compares the results of 8-bit size adder. Similarly, the results for the 16, 24, 32 bits size can also be estimated.

REFERENCES

[1] P. M. Kogge & H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. Computers, Vol. C-22, pp 786-793, 1973.

[2] T. Hans & D. A Carlson, "Fast Area-Efficient VLSI Adders," Proc. 8th IEEE Symposium on Computer Arithmetic, pp 49-56, 1987.

[3] Sreenivaas Muthyala Sudhakar, Kumar P. Chidambaram, and Earl E. Swartzlander, "Hybrid Han-Carlson adder," Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on, pp.818 – 821, 5-8 Aug. 2012.

[4] Y. Choi, "Parallel Prefix Adder Design with Matrix Representation", Proc. 17th IEEE Symposium on Computer Arithmetic, pp 90-98, 27th June 2005.

[5] S.V. Saravanan, R. Senthil Ganesh, "Design of Low Power and Area Efficient Dynamic Comparator", International Journal of Printing, Packaging & Allied Sciences, vol.04, no.02, pp.1464-1470, 2016.

[6] R. Kalaimathi, R. Senthil Ganesh, "Design of High Speed, Low Power and Area Efficient Floating Point Multiplier", International Journal of Advanced Engineering and Research Development, vol.04, Issue 11, pp.5-6, Nov. 2017.

[7] A. Beaumont-Smith and C.C. Lim, "Parallel-Prefix Adder Design," Proc. 15th Symp. Computer Arithmetic, pp. 218-225, June 2001.

[8] Haikun Zhu, Chung-Kuan Cheng, Ronald Graham, "Constructing zero-deficiency parallel prefix adder of minimum depth," in Proceedings of the 2005 Asia South Pacific Design Automation Conference, pp. 883 – 888, 2005.