International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Delay Analysis of Parallel-Prefix Adders

Geeta Rani[1], Sachin Kumar[2]

[1]M. Tech Student, Department of Electronics & Communication,
Meri College of Engineering & Technology, Sampla, Haryana, India

[2]Faculty, Department of Electronics & Communication,
Meri College of Engineering & Technology, Sampla, Haryana, India

Abstract: ***Parallel-Prefix adders or Carry tree adders are the kind of adders that uses prefix operation in order to do efficient addition. Nowadays Parallel-Prefix adders are the frequently used adders due to the high speed computation properties. So called carry tree adder uses the prefix operation to do the arithmetic addition with way greater speed than the simple parallel adders that is ripple carry adder, carry skip adder, carry select adder etc. Here in this paper we will discuss about the various parallel-prefix adders and analyses there delay with respect to one another so that the fastest adder can be found and also the specific adder for a specific operation can be found. Therefore we will discuss the parallel-prefix adders and compare them in order to find the righteous one.***

Keywords: Kogge-Stone adder, Brent-Kung adder, Ladner-Fischer adder, Han-Carlson adder, S. Knowles adder, Sklansky Conditional-Sum adder

1. Introduction

Adders are one of the indispensable components in digital building blocks, however, the performance of adders become more crucial as technology advances. Addition involves algorithms in Boolean algebra and their respective circuit implementation. The linear-delay adders like ripple-carry adders (RCA), are the most straightforward but slowest one. Carry-skip adder (CSKA), carry-select adder (CSEA) and carry-increment adder (CINA) are linear-based adders with optimized carry-chain. Carry look-Ahead adder (CLA) [1] have logarithmic delay and have evolved to parallel-prefix structures. Carry Look-Ahead adders terminology is equivalent to Parallel-prefix adders, but transistor topology is different.

Parallel-Prefix adders perform parallel addition i.e. most important in microprocessors, DSPs, mobile devices and other high speed applications. Parallel-Prefix adder reduces logic complexity and delay thereby enhancing performance with factors like area and power. Therefore the Parallel-Prefix adders are requisite element in the high speed arithmetic circuits and popular since twenty years. Parallel-prefix computation carries out three necessary or vital steps: 1) Computation of carry generation & carry propagation signals by using no. of input bits. 2) Calculating all the carry signals in parallel that is called prefix computation. 3) Evaluating total sum of given inputs. These steps are given in fig.1 that is given above.

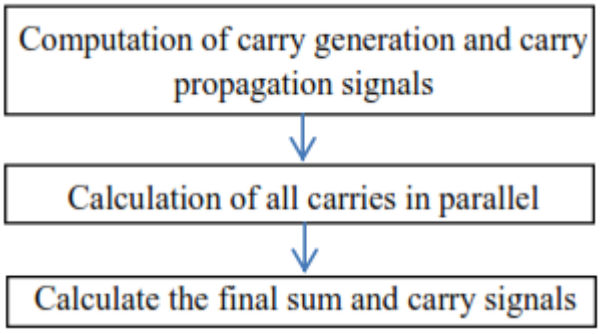

Figure 1: Parallel-Prefix adder mechanism

The process or three step that is carried out in the parallel prefix addition is as follows:

1. Computation of carry generation, propagation signals:
   Previous carry is calculated to the next bit is called propagate signal and generate is to generate the carry bit below are the signals:

$$G_i = A_i \cdot B_i \quad \text{.....(1)}$$
$$P_i = A_i \oplus B_i \quad \text{..... (2)}$$

2. Calculation of all carry signals:

$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j} \quad \text{.....(3)}$$
$$P_{i:j} = P_{i:k} \cdot P_{k-1:j} \quad \text{..... (4)}$$

3. Calculation of Final Sum:

$$S_i = P_i \oplus G_{i-1:0} \quad \text{..... (5)}$$

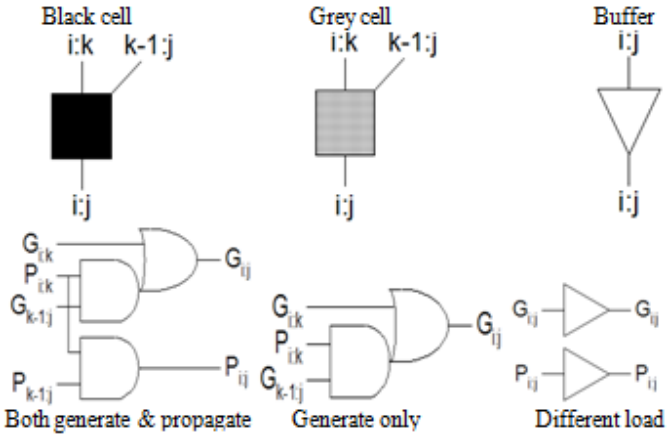

Figure 2: Black, Grey cells, Buffer and there Schematics

There are various types of Parallel-Prefix adders some of which are discussed in this paper. Kogge-Stone adder, Brent-Kung adder, Ladner-Fischer adder, Han-Carlson adder, S.Knowles adder, Sklansky Conditional-Sum adder are the few Parallel-Prefix adders. From all of the above adders Kogge-Stone is the one that is widely and efficiently used.

2. Kogge-Stone adder

Kogge-Stone adder is a parallel-prefix form carry look-ahead adder. Kogge-Stone adder was developed by [3] Peter M. Kogge and Harold S. Stone which they published in1973. KS adder is a fast adder design as it generate carry signal in $O(log_2\ n)$ time and has the best performance in VLSI implementations. KS adder has large area with minimum fan-out which increases its performance. Kogge-Stone adder is widely used in high performance 32-bit, 64-bit, and 128-bit adders as it reduces the critical path to great extent. In fig.3 each vertical stage produce propagate and generate bits. Generate bits are produced in the last stage and XORed with initial propagate and generate bits to produce sum.

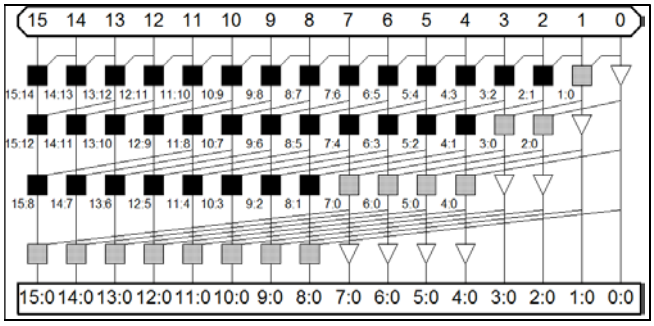

Figure 3: 16-bit Kogge-Stone Adder

Carry Stages: $log_2\ n$;
The number of cells: $nlog_2\ n$;
Maximum fan-out: 2.
KS takes more area to implement than Brent-Kung adder but has lower fan-out and wiring congestion is often a problem.

3. Brent-Kung adder

Brent-Kung parallel-prefix adder was developed by Brent and Kung which they published in 1982. Brent-Kung has maximum logic depth, minimum area and avoid explosion of wires. The Brent-Kung adder does odd computation first and then even. It computes prefixes [12] for 2-bit groups. These are used to find prefixes for 4-bit groups, which in turn are used to find prefixes for 8-bit groups, and so forth. The prefixes then fan back down to compute the carries-in to each bit. The tree requires $2log_2n-1$ stages. The fan-out is limited to 2 at each stage. Fig.4 shows buffers used to minimize the fan-out and loading on the gates, but, in practice, the buffers are generally omitted. The basic blocks used in this case are gray and black cells. This adder is implemented for 8, 16 and 32-bit using CMOS logic and transmission gate logic.

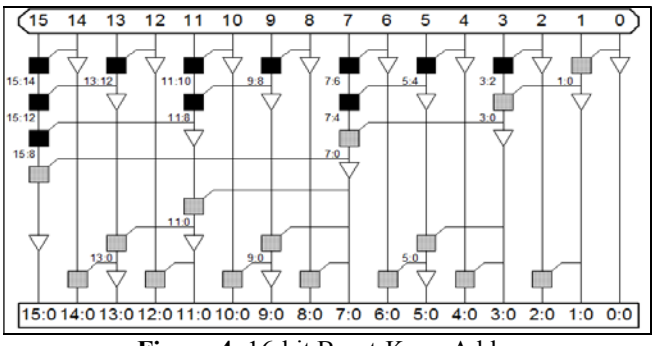

Figure 4: 16-bit Brent-Kung Adder

Carry Stages: $2log_2\ n-1$;
The number of cells: $2(n-1)-log_2\ n$;
Maximum fan-out: 2.

4. Ladner-Fischer adder

Ladner- Fischer parallel prefix adder was developed by R. Ladner and M. Fischer in 1980. Ladner-Fischer prefix tree is a structure that sits between Brent-Kung and Sklansky prefix tree. The LF adder [5] has minimum logic depth but it has large fan-out. Ladner- Fischer adder has carry operator nodes. The delay for the type of Ladner-Fischer prefix tree is log_2n+1. Fig.5 shows the 16-bit LF adder.

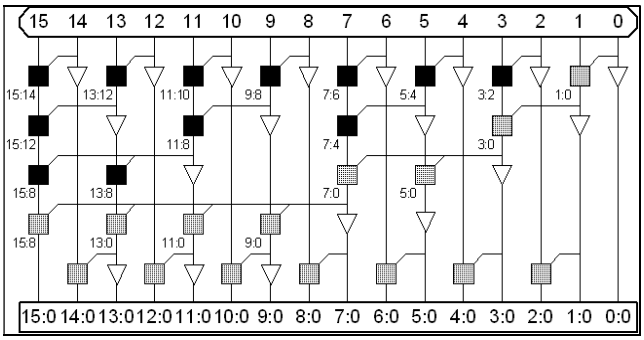

Figure 5: 16-bit Ladner- Fischer Adder

Carry Stages: $log_2\ n$;
The number of cells: $(n/2).log_2n$;
Maximum fan-out: n/2.

5. Han-Carlson adder

The *Han-Carlson* trees are the family of networks between Kogge-Stone and Brent-Kung. Han-Carlson adder can be viewed as a sparse version of Kogge-Stone adder. This scheme is different from Kogge-Stone scheme in the sense that these performs carry-merge operations on even bits and generate/propagate operation on odd bits. At the end, these odd bits recombine with even bits carry signals to produce the true carry bits.

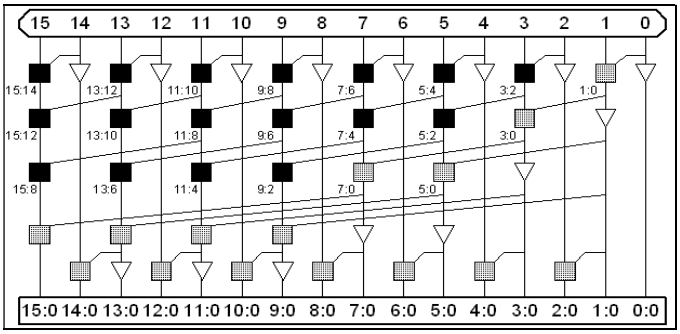

Figure 6: 16-bit Han-Carlson Adder

This adder has five stages in which the middle three stages are resembles with the Kogge-Stone structure. The advantage of this adder is that it uses much less cells and its shorter span wires than the Kogge-Stone adder and thus there is reduction in complexity at the cost of an additional stage for carry-merge path [7]. The pseudo-code for KS adder can be easily modified to build a Han-Carlson adder. Fig.6 shows a 16-bit Han-Carlson adder.

It happens to have the same number cells as Sklansky adder since the cells in the extra logic level can be move up to make the each of the previous logic levels all have *n/2* cells. The area is estimated as $(n/2).log_2n$. It has a maximum fan-out=2. It trades logic level for wire length.

Carry Stages: $log_2 n+1$;
The number of cells: (n/2);
Maximum fan-out: 2.

6. S. Knowles adder

S. Knowles [6] proposed a family of prefix trees with flexible architectures. Knowles adder composed of Kogge-Stone and Sklansky Conditional-Sum adder Knowles adder uses the fan-out at each logic level. Fig.7 shows a 16-bit Knowles adder. Different fan-out in the same logic level is allowed in Knowles prefix trees, which is called hybrid Knowles prefix tree.

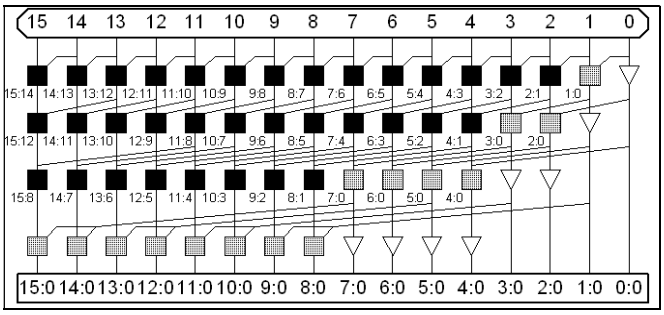

Figure 7: 16-bit S. Knowles Adder

Carry Stages: $log_2 n$;
The number of cells: $log_2 n$;
Maximum fan-out: 3.

7. Sklansky-Conditional Sum adder

Sklansky adder's structure is the simplest among the prefix adders. In Sklansky [9] adder, binary trees of cells generate all the carry input bits simultaneously.

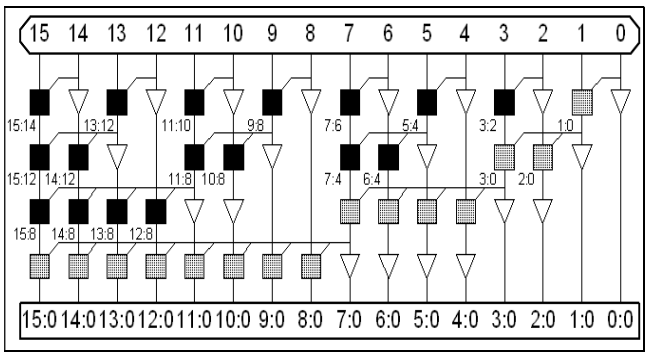

Figure 8: 16-bit Sklansky Conditional-Sum Adder

The Sklansky or divide-and conquer tree reduces the delay to log_2n stages by computing intermediate prefixes along with the large group prefixes. This comes at the expense of fan-outs that double at each level. The gates fan-out to (8, 4, 2, 1) respectively. These high fan-out cause poor performance on wide adders unless the high fan-out gates are appropriately sized, or critical signals are buffered before being used for intermediate prefixes. Transistor sizing can cut into the regularity of the layout because multiple sizes of each cell are required although the larger gates can spread into adjacent columns.

Carry Stages: $log_2 n$;
The number of cells: (n/2);
Maximum fan-out: n/2+1.

Table 1: Algorithm Analysis

Types	*Logic Level*	*Area*	*Fan-out*	*Wire Track*
Kogge-Stone	log_2n	$nlog_2n - n + 1$	2	n/2
Brent-Kung	$2log_2n-1$	$2n - log_2n - 2$	2	1
Ladner-Fischer	log_2n+1	$(n/4)log_2n + 3n/4 -1$	n/4+1	1
Han-Carlson	log_2n	$(n/2)log_2n$	2	n/4
Knowles	log_2n	$n\ log_2n-n+1$	3	n/4
Sklansky	log_2n	$(n/2)\ log_2n$	n/2+1	1

8. Result and Conclusion

The Ideal N-bit tree adder would have:

- L= log N logic levels
- Fan-out of 2
- No more than one wiring track between levels

Kogge-Stone, Han-Carlson and Knowles adders require a large number of parallel wiring for wide bit adders. Thus packing the wires close together will increase the coupling capacitance on each wire. Sklansky architecture becomes slow due to its high fan-out. When interconnect is considered Han-Carlson become attractive one as it requires only half the number of columns.

Individually specifications are like *Kogge-Stone* has least logic levels but hard to P and G. *Brent-Kung* is the very first and bad-one. *Ladner-Fischer* has a bit more logic levels and high fan-out. *Han-Carlson* has more logic levels but less cells. *S. Knowles* possesses many cells and wires and some fan-out. *Sklansky* has least logic levels and highest fan-out. If wire capacitance is neglected *Kogge-Stone adder is the best among the others.*

In table II all the above mentioned Parallel-prefix adders are compared in terms of delay.

Table 2: Comparison in Terms of Delay

	Delay (in ns)			
Name of adder	n=16	n =32	n =64	n =128
Kogge-Stone	9.4	12.4	17	24.8
Brent-Kung	10.4	13.7	18.1	24.9
Ladner-Fischer	9.9	11.5	14.9	18.9
Han-Carlson	9.9	12.1	15.1	19.7
S. Knowles	9.7	12.7	17.3	25.1
Sklansky	13	21.6	38.2	70.8

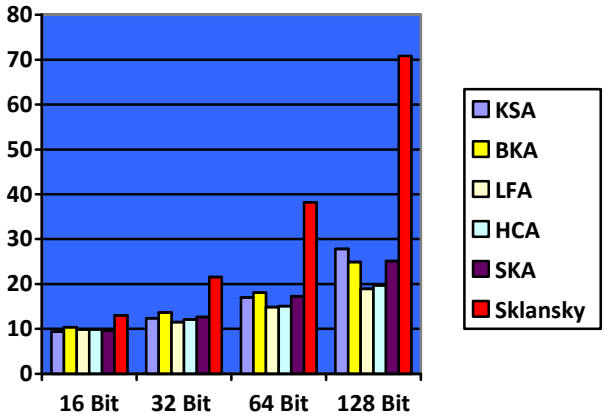

Figure 9: Graphical representation of 16, 32, 64, 128-bit Parallel prefix adders

9. Future Scope

This paper is a survey on the various Parallel-Prefix adders. This survey shows the various aspects of the parallel-prefix adder and there specifications. This is useful in terms of when somebody is looking for a adder with a particular delay, less wiring congestion and with specific fan-out he/she can get the information and will use the particular adder without wasting time. Parallel-prefix computation can used to build fast algorithms for *parallel interpolation.* This prefix based approach can also be used to obtain the generalized divided differences for *Hermite interpolation.*

Reference

[1] Weinberger and J. Smith, "A logic for high-speed addition", National Bureau of Standards, no. Circulation 591, pp. 3–12,195.
[2] M. Mahaboob Basha, Dr. K. Venkata Ramanaiah, Dr. P. Ramana Reddy, B. Lokeshwar Reddy, " An efficient model for design of 64-bit High Speed Parallel Prefix VLSI adder", International Journal of Modern Engineering Research (IJMER) Vol. 3, Issue 5, Sep-Oct 2013.
[3] Kogge P and Stone H, "A Parallel Algorithm for the Efficient Solutions of a General Class of Recurrence Relations", IEEE Transactions on Computers, Vol. C-22, No.8, 1973.
[4] B Pullarao, J.Parveen Kumar, "Design of High Speed Based On Parallel Prefix Adders Using In FPGA", International Journal of Engineering Sciences & Research Technology (IJESRT) Dec, 2013.
[5] R. Ladner and M. Fischer,"Parallel prefix computation", Journal of ACM. La. Jolla CA, Vol.27, pp.831-838,October 1980.
[6] S. Knowles, "A family of adders", in Proc. 15th IEEE Symp. Comp. Arith., June 2001, pp. 277.281.
[7] Han, Carlson, "Fast Area-Efficient VLSI Adders, IEEE" 1987.
[8] Vishal R. Naik, Sonia Kuwelkar," DESIGN OF A CARRY TREE ADDER", International Journal of Pure and Applied Research in Engineering and Technology (IJPRET) Vol. 2(9), 413-424, 2014.
[9] Dakupati. Ravi Sankar, Shaik Ashraf Ali, "Design of Wallace Tree Multiplier by Sklansky Adder", International Journal of Engineering Research and Applications (IJERA) Vol. 3, Issue 1, Jan-Feb 2013.
[10] David Harris and Ivan Sutherland, "Logical effort of Carry Propgate Adders", IEEE 7803-8104, 2003.
[11] Andrew Beaumont-Smith and Cheng-Chew Lim, "Parallel Prefix Adder Design", Department of Electrical and Electronic Engineering, the University of Adelaide, 2001.
[12] Deepa Yagain, Vijaya Krishna A, Akansha Baliga, "Design of High-Speed Adders for Efficient Digital Design Blocks", International Scholarly Research Network ISRN electronics, July2012.

Author Profile

Geeta Rani received her B. Tech degree in Electronics and Communication Engineering from R.I.E.M. College under Maharishi Dayanand University, Rohtak. Currently pursuing M. Tech in Electronics and Communication Engineering from M.E.R.I. College under Maharishi Dayanand University.