

A Modified Fused Floating Point Three Term Adder



Neeraja P K, Ramadass Narayanadass

Abstract—This paper is about a modified architecture for a fused floating point three term adder. The important feature of a fused floating-point three-term adder is its ability to do multiple additions in same block to get better performance as well as accuracy compared to a conventional discrete floating point adder. The parallel prefix adder is one amongst the fastest adders and out of which the han-carlson adder represents a blend of the kogge-stone adders and brent-kung adder. In this work, han carlson adder is used to enhance the performance of the three term adder along with various optimization techniques. The adder is implemented using Verilog language in Xilinx ISE Design suite 14.2 and all Simulations are carried out in Isim simulator. Synthesis is done using Cadencetool.

Keywords—Floating point adder, Parallel prefix addition, Kogge Stone adder, Han Carlson adder.

I. INTRODUCTION

Among most of the DSP implementations, multiple floating-point additions are executed consecutively. Floating point calculates a way of expressing an estimation of real number in such a way that it can uphold wide scope of values. It is described in IEEE-754 Standard for the floating point arithmetic[1]. Mostly, the numbers are expressed approximately to a specific number of significant digits and scaled utilizing an exponent. The base can be 2, 6 or 10. Generally, a typical number can be expressed as:

$$(-1)^{\text{sign}} \times \text{significant digits} \times \text{base}^{\text{exponent}}$$

Main feature of floating-point multi-term adder is that it can take different operands and can execute numerous additions with an operation to produce a sum. There are two methods to design the floating point three-term adder namely a) Discrete floating-point three-term adder and b) Fused floating-point three-term adder. Complex processes like alignment, normalization and rounding are essential for floating point operation, but this increases area, power consumption and latency. So as to eliminate such issues, fused floating-point units have been proposed, that perform numerous operations in a single block to reduce the area, power consumption and latency. Some problems for the implementation of the fused floating point three term adder are studied in the past work [3], [4]: 1) Complex exponent processing and significand alignment, 2)

Significand addition followed by complementation, 3) Large precision significand addition, 4) Massive cancellation management and 5) Complex round processing. Here in this work, such issues are addressed as well as performance is enhanced by introducing many optimization techniques and by modifying the type of parallel prefix adder used.

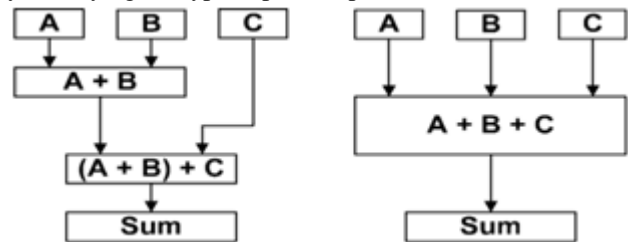


Fig. 1 Discrete Vs Fused floating point three term adders
Section II explains the traditional fused floating-point three term adder. Improved architectures for the fused floating point three term adder design are presented in next section. Here, various optimization techniques are utilized to enhance the performance of the fused floating point three term adder and are briefly described in section III [2]. Based on a data flow analysis, modification is applied to the improved fused floating-point three-term adder. In section IV, the proposed design is implemented for single precision. For this work han-carlson adder is used as a parallel prefix adder that is a combination of brent-kung adder and kogge stone adders section V gives results and conclusion is given in section VI.

II. CONVENTIONAL FUSED FLOATINGPOINT THREE TERMADDER

In order to implement a floating point three term adder, it is possible to fuse two identical floating point adders. One among the high performance floating point adders [8]–[10] can be used for the discrete design. The discrete floating point adder takes much more area compared to single term adder even though it has the capability to do multi term additions. For decreasing the overhead, fused floating point three term adders are introduced, that can carry out two additions in a single block [3], [4]. In order to decrease the area, power consumption and delay, a common logic is shared by the adder. Fig. 1 shows comparison of the discrete design and presented fused design. Moreover, the fused floating point three term adder improves the accuracy of design. The discrete floating point three term adder performs rounding twice whereas the fused floating point three term adder do the rounding only once, thereby improving the accuracy.

Revised Manuscript Received on October 25, 2020.

*Correspondence Author

Neeraja P K, Department of Electronics and Communication Engineering, College of Engineering Guindy, Anna University, Chennai, India. Email: neerajapkn@gmail.com

Ramadass Narayanadass*, Department of Electronics and Communication Engineering, College of Engineering Guindy, Anna University, Chennai, India. Email: ramadassn@annauniv.edu

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

A Modified Fused Floating Point Three Term Adder

Fig. 2 shows a typical fused floating point three term adder [3], [4]. This adder takes three operands and performs the two additions at once.

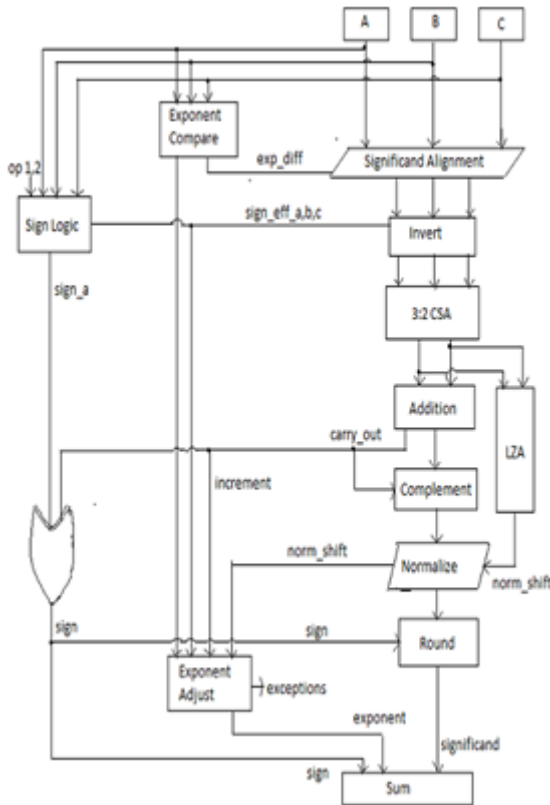


Fig. 2 Typical fused floating point three-term adder

III. OPTIMIZATION TECHNIQUES

The conventional fused floating point three term adder lessens the latency, area and power consumption compared to the discrete version. In this section, a few optimizations are explained that can be used for improving the performance of the adder design: 1) A new exponent compare and significand alignment method 2) Dual reduction 3) Early normalization 4) Three input LZA and 5) Compound addition and rounding.

1) Novel exponent compare with significand alignment method is presented. Here, three pairs of exponent differences are assessed in parallel and differences are utilized for significand alignment. The exponent difference evaluation and also significand alignment can be overlapped by shifting the significands using partial difference results. The control logic is responsible for finding out the largest exponent and three aligned significands. At the same time, exponent processing and significand alignment are performed. This thereby can lessen the latency.

2) Two significand pairs such as positive as well as negative significand can be generated by two reduction trees. Positive significand pair is chosen based on the significand comparison. This method is called Dual reduction. So, complementation after the significand addition can be skipped as the chosen significand pair produces a positive sum, which reduces the delay.

3) Early normalization is applied to decrease the significand addition size. Adder size is then reduced significantly by performing the normalization before the significand addition. Latency is again reduced

afterrounding.

As the normalization is done before significand addition, the Leading Zero Anticipation (LZA) as well as normalization shift are on the critical path. A three-input LZA is presented so as to decrease the latency, which covers up the delay of the 3:2 reduction trees.

For fast rounding, compound addition is utilized. Rounded and unrounded sums are resulted by compound addition and the round logic finds out the accurate result so that the delay of the rounding is hidden.

IV. FUSED FLOATING POINT THREE TERM ADDER USING HAN CARLSON ADDER

In the conventional fused floating point three term adder, several optimization techniques have been used. The first part of significand addition is performed after the reduction to balance with the delay of LZA. The necessity for an adder is that it is fast as well as efficient according to power consumption and chip area. So here comes the advantage of parallel prefix adders. It is a technique for improving the speed of addition.

A. Kogge Stone adder:

A Kogge stone adder has been utilized as a parallel prefix adder in the improved adder architecture. The Kogge Stone has low logic depth, high node count, and less fan out. High node count implies a bigger area whereas the low logic depth and minimal fanout helps for faster performance. Mainly there are three computational stages in Kogge Stone Adder. They are:

1. Preprocessing
2. Carry generation network
3. Postprocessing

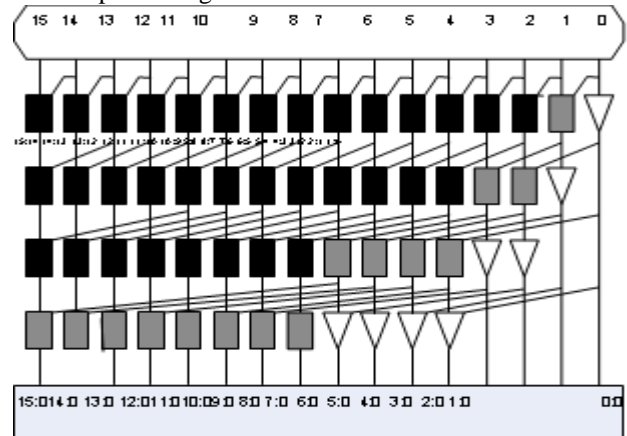


Fig. 3 16 bit Kogge Stone adder

$$P_i = a_i \oplus b_i \quad (1)$$

$$g_i = a_i b_i \quad (2)$$

$$P_i^j = p_i p_{i-1} \dots p_{i-(j-1)} \quad (3)$$

$$G_i^j = g_i + g_{i-1} p_i + \dots + g_{i-(j-1)} p_i p_{i-1} \dots p_{i-(j-2)} \quad (4)$$

Kogge Stone construction that takes $\log_2 n$ stages and the Brent kung construction that takes $2\log_2 n - 1$ stages [5,6]. Main advantage of Brunt kung design is that it takes a lesser area. where a_i and b_i are i^{th} bits of two significant and j is the level of the prefix tree adder. But still Kogge Stone adder faces issues with its complex wiring circuitry. In the modified fused floating point three term adder, a Han Carlson adder is being used. Calculation of Carry generate as well as Propagate signals Calculation of all Carry signals in parallel Calculation of last sum as well as carry signals

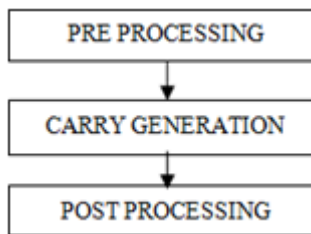


Fig. 4 Parallel Prefix mechanism

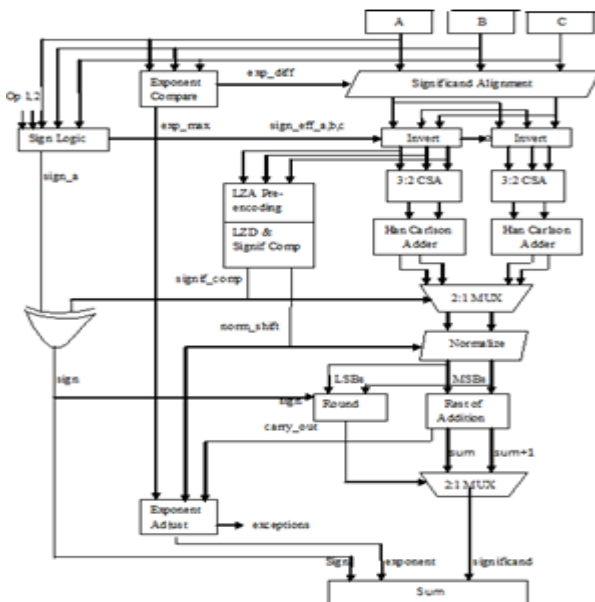


Fig. 5 A modified fused floating point three term adder

B. Han Carlson adder:

Han Carlson adders are the group of networks comes in middle of Kogge stone and Brent Kung adders. Here, for 54 bit addition, Han Carlson adder is utilized. This sort of computation gives a speed nearer to Kogge Stone adder. A hybrid version of parallel prefix adders is put forward by T. Han and D.A. Carlson by utilizing two designs like

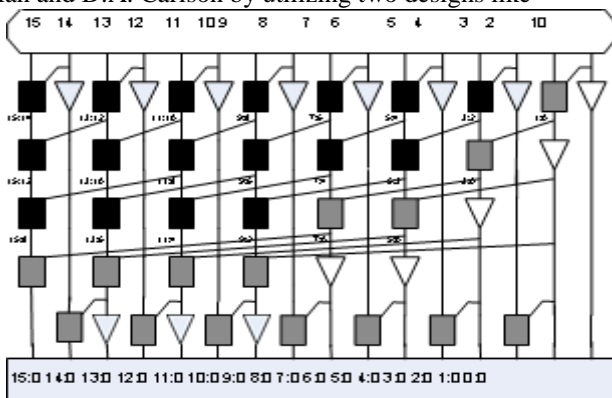


Fig. 6 16 bit Han Carlson adder

Thus Han Carlson adder has best features; high speed

obtained from Kogge Stone adder and low area obtained from Brunt Kung design. Both features are combined to provide a better speed with low complexity. This 16 bit Han Carlson adder has five stages out of which three stages in the middle is similar to the Kogge Stone structure. Number of cells used by this adder is less compared to that used by Kogge Stone adder. And also this adder has shorter span wires so that wiring complexity is lesser than that in Kogge Stone structure [7]. The pseudo code for Han Carlson adder can be easily obtained by modifying that of Kogge Stone adder. The number of carry stages is $\log_2 n + 1$ where as for Kogge Stone adder, it is $\log_2 n$. Fig. 7 shows method of Computation of carry in different stages.

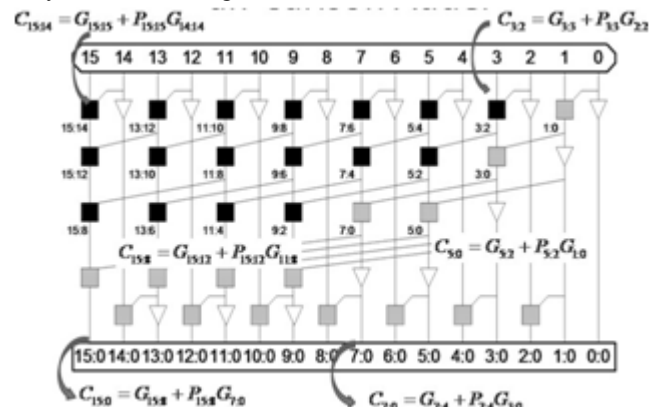


Fig. 7 Carry Computation method of Han Carlson Adder

V. RESULTS

The proposed fused floating point three term adder is designed using Verilog language in Xilinx ISE Design suite 14.2.

ISim simulator is utilized for performing simulations. Later, the performance of conventional and proposed adders are analyzed and compared. Implementation code for 54-bit Kogge Stone adder as well as Han Carlson adder are developed in this work and the respective values of delay and area are measured. Table-I shows the algorithm analysis of parallel prefix adders and Table-II shows the synthesis result. Result comparisons are shown in tables 3,4. Simulation waveform results are shown in figures 7,8 and 9.

Table-I: Algorithm Analysis of PPA's

Type of PPA	Logic level	Area	Fan Out	Wire track
Kogge Stone	$\log_2 n$	$n \log_2 n - n + 1$	2	$n/2$
Han Carlson	$\log_2 n$	$(n/2) \log_2 n$	2	$n/4$

Simulation Results:

A Modified Fused Floating Point Three Term Adder



Fig.7 KSA Simulation output



Fig.8 Han Carlson adder Simulation output

Table-II: Synthesis Result

Adder	No: of Slice LUTs	No: of Bonded IOBs	No: of Slice registers	No: of fully used LUT-FF pairs
Traditional	956	1657	59	59
Modified	829	1495	59	59

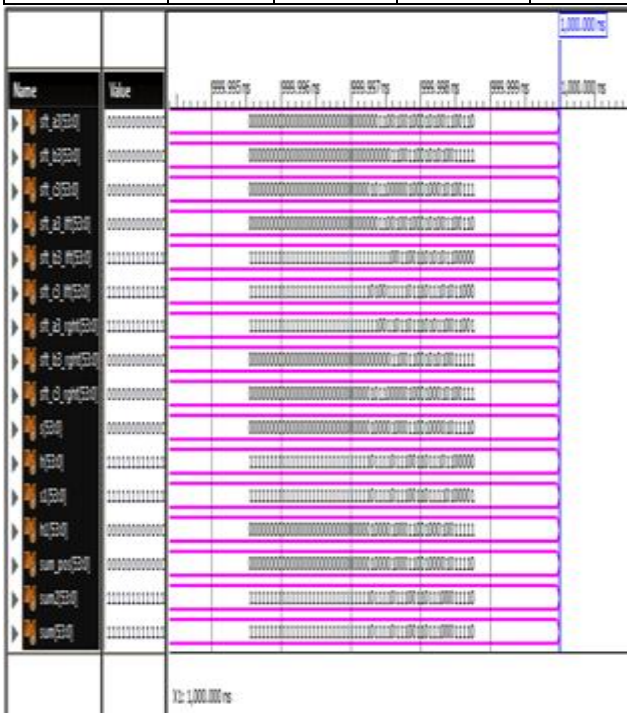


Fig.9 Simulation result of proposed adder

Both Kogge stone and Han Carlson adders require parallel wiring for wide bit adders. Han Carlson is so good as it requires only half number of columns when the interconnect is considered. Kogge stone tree has least number of logic levels; yet difficult to propagate as well as generate whereas Han Carlson has more number of logic levels; yet less number of cells.

Table-III: Result Comparison of PPAs

Parallel Prefix Adder Type	Cell Area	Leakage Power (nW)	Delay
Kogge Stone	1280	316	+0 528F
Han Carlson	1071	182	+0 6168F

Cell area and Leakage power consumption are more in Kogge stone adder

compared to Han Carlson adder.

Table-IV: Comparison of Traditional and Modified adder results

Adder	Cell Area	Leakage Power (nW)	Combinational Path Delay(ns)
Traditional	6973	1415	3.298
Modified	5780	930	3.311

II. FUTUREWORK

The work carried out in this paper can be extended to perform addition with terms more than three operands. Pipelining can also be employed along with other algorithms and optimization techniques discussed in the paper. Physical design implementation of the resulting adder also comes in the scope of future work.

III. CONCLUSION

In this paper, a modified fused floating point three term adder is designed using a parallel prefix adder called Han Carlson adder. Various critical issues were there for the conventional three term adder. Those issues have been resolved by modified fused floating point three term adder. Han Carlson adder that is having less wiring complexity than Kogge Stone adder is used. The improved architecture reduces number of cells, area, and leakage power. But there a slight increase in delay is observed.

REFERENCES

1. IEEE Standard for Floating-Point Arithmetic ANSI/IEEE Standard 754-2008, IEEE, Inc., 2008.
2. J.Sohn and E.E.Swartzlander, Jr., "A Fused Floating-Point Three-Term Adder", *IEEE Transactions On Circuits And Systems-I: Regular Papers*, Vol. 61, No. 10, October 2014.
3. A. Tenca, "Multi-operand floating-point addition," in *Proc. 21st Symp. Computer Arithmetic*, 2009, pp.161-168.



4. Y. Tao, G. Deyuan, F. Xiaoya, and R. Xianglong, "Three-operand floating-point adder," in *Pro. 12th IEEE Int. Conf. Comput. Inf. Technol.*, 2012, pp. 192–196.
5. Swapna K. Gedam and Pravin P. Zode, "Parallel Prefix Han-Carlson Adder", *International Journal of Research in Engineering and Applied Sciences*.
6. Sreenivaas Muthyala Sudhakar, Kumar P. Chidambaram and Earl E. Swartzlander Jr. "Hybrid Han-Carlson Adder", *978-1-4673-2527-1/12/\$31.00©2012 IEEE*
7. Geeta Rani, Sachin Kumar," Delay Analysis of Parallel-Prefix Adders", *International Journal of Science and Research(IJSR)*
8. M. P. Farnwald, "On the Design of High Performance Digital Arithmetic Units," Ph.D. dissertation, Computer Science, Stanford University, Stanford, CA, USA, 1981.
9. S.F.Oberman, H.AI-Twajiry, and M.J. Flynn, "The SNA Pproject: Design of floating point arithmetic units," in *Proc. 14th IEEE Symp. Computer Arithmetic*, 1997, pp.156–165.
10. P. M. Seidel and G. Even, "Delay-optimized implementation of IEEE floating-point addition," *IEEE Trans. Computers*, vol. 53, no. 2, pp. 97–113, Feb. 2004.

AUTHORS PROFILE



Neeraja P K was born in Kozhikode, Kerala, India in 1991. She received the B.Tech degree in Electronics and Communication Engineering from Kerala University in 2013 and M.E degree in VLSI design from Anna University, Chennai, India in 2016. She has completed Advanced diploma course in VLSI Physical Design from NIELIT, Calicut, India in 2019. She has worked as VLSI/Matlab Developer in Softroniics, Calicut and has completed training from Infosys, Mysore, Karnataka. She has also pursued an internship with Utkarshini Edutech in the field of Technical research- Digital Logic. Her research interests include VLSI design, Image processing and Signal Processing.



Ramadass Narayanadass was born in Chennai, Tamilnadu, India in 1975. He received the B.E degree in Electrical and Electronics Engineering from Madras University in 1997, M.E degree in Applied Electronics and Ph.D from Anna University, Chennai, India in 2001 and 2008, respectively. He has been associated with the Faculty of Information and Communication Engineering, Anna University, Chennai, since 2001. He is currently an Associate Professor in Department of Electronics and Communication Engineering at Anna University, Chennai. His research interests include Embedded Systems, VLSI design and Reconfigurable Computing.