An Improved Fused Floating Point Three Term Adder

Ms. Jeevan Jyoti[1] Mr. Mahendra Tyagi[2]
[1]Students [2]Professor
[1,2]Department of Electronics and Communication Engineering
[1,2]L R Group of Institute, Himachal Pradesh (India)

Abstract— Floating Point addition and subtraction units are widely used in various digital and signal processing applications. A traditional Floating Point Three Term Adder performs two additions using discreet units. To perform two additions in a single unit, architecture must be fused so that two adder units work as a single unit. In fused floating point adder, several optimization techniques are applied in order to further enhance the results like exponent compare and alignment unit, dual reduction, leading zero anticipator etc. In the proposed approach, fused floating point adder is implemented with each of the units and optimization techniques working in parallel, so as to reduce the delay in computation. Besides the pipelined approach, Carry Select Adder which is the fastest adder in the literature and is a combination of two ripple carry adders is used in the addition for the further enhancement in timing performance of the unit. Results also show that the timing is improved by reasonable value with the combined implementation of carry select adder and pipelined approach.

Key words: FMA, Fused Adder, Three term Adder, Normalization, LZD

I. Introduction

Most of the floating point arithmetic units are designed using the concepts of two fundamental units. Thus the techniques used for optimizations of the fundamental units are equally valid for the implementation of the advanced units also [1]. It works on the two operands which must be in the form of IEEE standard and their sign, exponent and mantissa can be easily fetched for further computations. In order to improve the basic procedure, several techniques can be applied: Compound addition and fast rounding, Leading zero anticipation (LZA) for fast normalization, and Dual-path algorithm [2].The basic floating-point addition operation can be performed as in the first stage, exponents are compared and significand alignment logic are described. In this step comparison among the exponents is determined and then according to the comparison the significand of the smaller number is shifted by same amount as that of the difference. In the next stage, two significands are then passed to the sticky logic [3]. The excess LSB bits are discarded after shifting by the sticky logic. The operation selection logic takes two operands (significands) and performs the addition or subtraction operation. If the operation is subtraction and the carry-out is positive, indicating the sum of the significands is negative; the sum is complemented to convert it to a positive number. Since the carry-out indicates the significand comparison in the case of subtraction, it is passed to the sign logic [4]. Leading zero detectors is used to detect the MSB bit of the sum. This operation is mainly performed to convert the sum back to the IEEE standard. This operation is termed as normalization. The detected MSB is then passed to the exponent adjust logic. The exponent adjust logic is used to adjust the value of exponent according to the IEEE specified format by adding and subtracting to the exponent. Also, the exponent adjust logic sets the exception flags (i.e., overflow, underflow and inexact) based on the adjusted exponent. The sign logic takes the two sign bits, opcode, exponent comparison and significand comparison, and generates the sign bit of the sum. The round logic rounds the sum and post normalizes the value depending on the modes specified in the standard [5].

II. Floating Point Three Term Adder

Floating point (FP) three term adders performs two additions simultaneously in one unit in order to acquire improved performance and correctness. FP three term adder designs could be implemented for both single and double precision [6]. The conventional fused FP three-term adder is a preliminary design so that optimizations might be applicable to enhance the performance. Here five optimizations for three term adder are given:

1) A new exponent compare and significand alignment design
2) Dual-reduction to avoid the need for complementation after the significand addition
3) Early normalization
4) Three-input LZA
5) Compound addition and rounding

Generally fused floating point three term adders get three normalized operand to execute following operation:

$$S = A + B + C$$

There are some demerits of FP three term adders. These are Complex exponent processing and significand alignment, enormous cancellation management and complex round processing, complementation after the significand addition, large precision significand adder were several vital design problems for fused FP three term adders.

III. Related Work

Sohn, Jongwook et al. [1] proposed an enhanced architecture for fused FP three term adder. The proposed architecture was implemented for both single and double precision and executed in 45nm CMOS technology. The proposed three term adder minimized the area and consumption of power by 20% and also minimized the latency by 35% as contrasted to the traditional discrete FP three term adders. More so, it was divided into three pipeline stages which were well balanced and throughput of proposed design was increased by 2.7 times that of non-pipeline design. Some parameters such as power consumption, area and latency were studied to compute the performance of design.

J.D. Bruguera et al [2] proposed an improved architecture for a floating–point Multiply–Add–Fused (MAF) unit that reduced the latency of the traditional MAF units. In this architecture, the normalization was carried out before the addition because the rounding position was not

known until the normalization has been performed. This architecture was based on the combination of the final addition and the rounding, by using a dual adder.

Sohn, Jongwook et al. [3] presented a novel design for a fused floating-point four-term dot product unit. The presented design evaluated the four-term dot product in a single unit to attain improved performance and precision compared to conventional floating-point multipliers and adders, which was referred as a discrete design. To enhance the performance further, the fused floating-point four-term dot product unit, significand alignment scheme, dual-reduction, early normalization, four-input Leading Zero Anticipation (LZA), and compound addition and rounding were applied. The proposed design reduces the area, power consumption and latency by about 40% compared to the discrete design.

Saikumar Addula et.al. [4] Proposed high speed floating point double precision adder/subtractor and it was implemented on a Virtex-7 FPGA. All the modules of proposed design support Verilog simulation in the Modelsim and synthesized using Xilinx 14.1 ISE software. The presented floating point unit architecture has attained operating frequencies of 371.858 MHz and further more the designs attained the operating frequencies of 363.76MHz, 414.714MHz and 452.694MHz with an area of 660, 648 and 841 slices respectively.

Earl E. [5] proposed two fused floating-point operations. Furthermore, two fused operation such as two-term dot product and add-subtract unit also being performed. In this paper, butterfly operation has been utilized by the FFT processor which comprised of addition, subtraction and multiplication of complex valued data. The proposed two fused operation has been implemented effectively with help of radix-2 and radix-4 butterflies. The experiment result demonstrated that the fused FFT butterflies were approximately 15% faster and 30% smaller than other traditional implementation.

Jeevan Jyoti et al [6] surveyed the floating point unit design for several arithmetic operations. It explained that in this modern technology, circuit complexity is growing day by day. So the power dissipation plays vital role in designing of any digital circuit. In order to minimize the power dissipation, reversible logic has been designed. This reversible logic approach is fast in term of computation and must dissipate less power. Moreover, multi-operands unit also surveyed and it used for both fused and distributed concept.

Blomgren, James S [7] proposed architecture executes two additions in one unit to attain the accuracy and good performance. Various optimization approaches has been applied in order to obtain the better performance of three term adder. It can be used to begin the normalization and it also help to avoid the increase in delay. The proposed MAF reduced the delay by about 15% − 20% for double precision formats, as compared to the traditional floating–point MAF unit.

IV. PROPOSED METHODOLOGY

The proposed methodology consists of three important steps. These are:

1) Exponent Compare and Significant Alignment
2) Addition Unit
3) Post Normalization

A. *Exponent Compare and Significant Alignment:*

In this unit the input exponents are compared and their difference is used to align the mantissa of the three operands. The approach needs to perform exponent subtractions, complementation and significand shift consecutively. The exponent variations are used for the significant shifters. The new exponent compare and alignment logic reduces the latency compared to the standard technique by performing the exponent comparison and alignment operations at the same time.

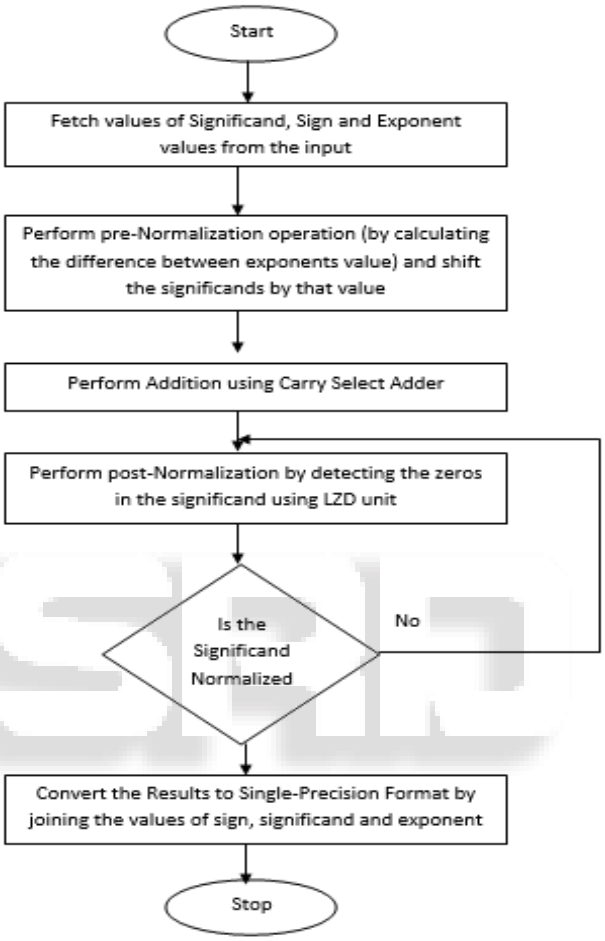

Fig. 1: Flow Diagram of the Proposed Methodology

B. *Addition Unit:*

Carry Select adders are considered as the fastest adders in the literature. CSAs are used for the implementation of the floating point adder in order to reduce the further delay. CSAs are made up of two Ripple Carry adders with each adder is used to calculate the output according to the carry taken as 1 and 0. Finally multiplexer selects the output according to the input carry value.

C. *Post Normalization:*

Post Normalization operation is performed after the addition operation. In this operation if there is a carry then the exponent in the third unit is also increased or decreased according to the value obtained after the addition operation. Normalization shift quantity is deducted just in case large cancellation happens throughout the subtraction. Since the normalization shift amount is made before the significand addition, only the two bit carry-out addition affects the

important path. Figure I show the flow diagram of the proposed methodology.

V. RESULTS AND DISCUSSIONS

The proposed methodology is implemented using Xilinx Spartan 3 FPGA. The coding language is Verilog and the software environment is Xilinx ISE and simulation is done using Xilinx ISim. The whole system is divided into three main stages. Figure II shows the top level module of the proposed methodology.

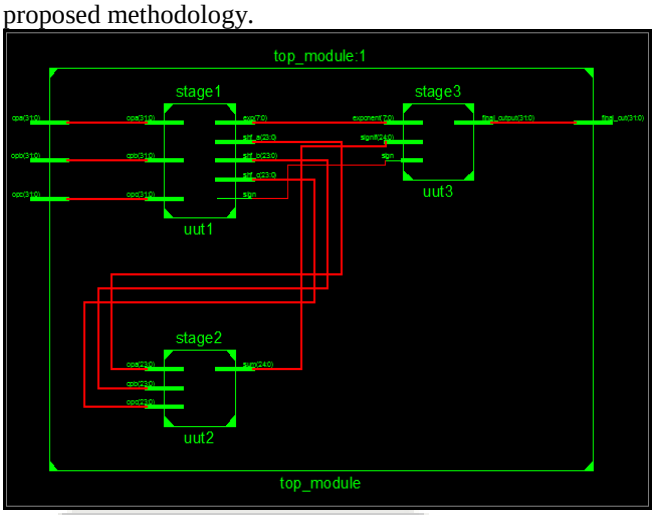

Fig. 2: Top Module of the Proposed Approach

The critical path Delay is the maximum delay which a circuit must have from input to output. The proposed approach shows the decrease in delay in the proposed approach with an increase in number of devices used. Figure III shows the simulation waveforms of the proposed methodology. Table I shows the comparison of the resources used and the delay of the proposed methodology with the basic approach.

Parameters	Basic Approach	Proposed Approach
Delay (ns)	68.680	53.172
Max. Frequency (MHz)	14.56	18.8
Number of Slices	611	618
Number of 4-input LUTs	1091	1111
Number of Bonded IOs	128	128

Table 1: Comparison Table

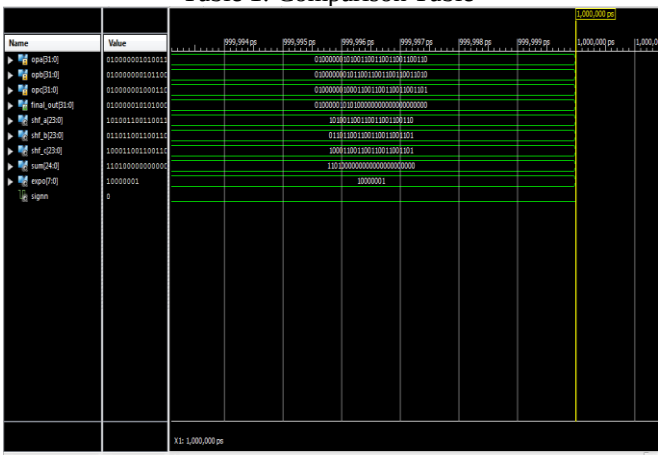

Fig. 3: Simulation Waveform of the Proposed Methodology

VI. CONCLUSION

The Fused Floating Point three term adder is implemented using the Carry Select Adder with all units working in parallel to improve the timing efficiency. The results also show that the timing is improved using the proposed methodology with the cost of resources utilized and the power. The results also show that the resources used in implementation are increased by a slight amount and the timing is far improved. As the resource utilization of FPGA increases the power utilization for the proposed methodology also increases.

REFERENCES

[1] Sohn, Jongwook, and Earl E. Swartzlander. "A fused floating-point three-term adder." IEEE Transactions on Circuits and Systems I: Regular Papers61, no. 10 (2014): 2842-2850.

[2] T.Lang,.;Bruguera, J.D., "Floating- point multiply-add-fused with reduced latency," IEEE Transactions on Computers, Volume-53, Issue-8, PP 988-1003, August 2004.

[3] Sohn, Jongwook, and Earl E. Swartzlander. "A Fused Floating-Point Four-Term Dot Product Unit." IEEE Transactions on Circuits and Systems I: Regular Papers 63, no. 3 (2016): 370-378.

[4] SravanthiKavitha, AddulaSaikumar. "An FPGA Based Double Precision Floating Point Arithmetic Unit using Verilog." In International Journal of Engineering Research and Technology, vol. 2,.ESRSA Publications, 2013.

[5] Swartzlander, Earl E., and Hani HM Saleh."FFT implementation with fused floating-point operations." IEEE transactions on computers 61, no. 2 (2012): 284-288.

[6] Jeevan Jyoti. "Literature Review Based On Fused Floating Point Three Term Adder." Imperial Journal of Interdisciplinary Research 2, no. 9 (2016).

[7] Blomgren, James S., and Terence M. Potter. "Type conversion using floating-point unit." U.S. Patent 9,264,066, issued February 16, 2016.

