International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Improved Architectures for Fused Floating Point Add-Subtract Unit

Pooja Potdar[1], S. S. Tamboli[2]

Annasaheb Dange Collage of Engineering & Technology, Ashta

Proessor, Annasaheb Dange Collage of Engineering & Technology, Ashta

Abstract: ***The fused floating point add-subtract unit is useful for digital signal processing (DSP) applications Such as fast Fourier transform (FFT) & discrete cosine transform (DCT) butterfly operations. To improve the performance of fused floating point add-subtract unit, a dual path algorithm& pipelining algorithms are useful .The designs are implemented for both single and double precision. The fused floating point add-subtract unit saves area and power consumption compared to discrete floating point add-subtract unit. The dual path design reduces latency compared to discrete design with area and power consumption between discrete and fused design. The fused dual path floating point add-subtract unit can be split into two pipeline stages, since latencies of two pipeline stages will be fairly well balanced.***

Keywords: Digital signal processing(DSP), Floating point arithmetic, Fused floating point operation, High speed computer arithmetic.

1. Introduction

Current digital signal processing (DSP) systems are making the transition from fixed-point arithmetic (used initially because of its simplicity) to floating-point arithmetic. The latter has several advantages including the freedom from overflow and underflow and ease of interfacing to the rest of the system (which generally will use IEEE-754 Standard floating-point arithmetic). To improve the performance of floating-point arithmetic, several fused floating-point operations have been introduced: Fused Multiply–Add (FMA) , Fused Add–Subtract and Fused Two-Term Dot-Product. The fused floating-point operations not only improve the performance, but also reduce the area and power consumption compared to discrete floating-point implementations. This project presents improved architecture designs and implementations for a fused floating-point add–subtract unit. Many DSP applications such as fast Fourier transform (FFT) and discrete cosine transform (DCT) butterfly operations can get benefit from the fused floating-point add–subtract unit. Therefore, the improved fused floating-point add–subtract unit will contribute to the next generation floating-point arithmetic and DSP application development.

In this paper we will propose improved architectures for a fused floating point add-subtract unit which will generates sum and difference simultaneously. It will supports all five rounding modes specified in IEEE-754 standard. Different techniques will be applied to achieve low area, low power consumption, and high speed.

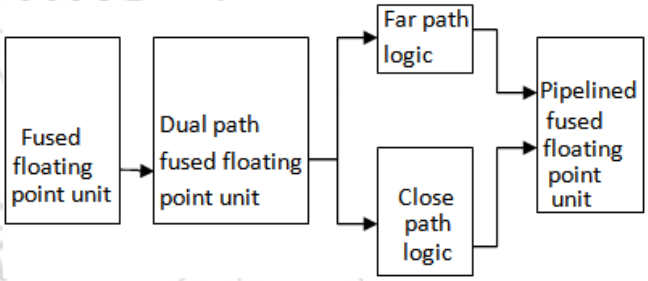

Figure 1: Blok Diagram of Proposed System

2. Literature Review

Floating point adder accepts normalized numbers, supports all four IEEE rounding modes, and outputs the correctly normalized rounded sum/difference in the format required by the IEEE Standard. The FP-adder design achieves a low latency by combining various optimization techniques such as: A nonstandard separation into two paths, a simple rounding algorithm, unification of rounding cases for addition and subtraction, sign-magnitude computation of a difference based on one's complement subtraction, compound adders, and fast circuits for approximate counting of leading zeros from borrow-save representation. They presented technology-independent analysis and optimization of their implementation based on the Logical Effort hardware model and determined optimal gate sizes and optimal buffer insertion. Estimated the delay of optimized design at 30.6 FO4 delays for double precision operands (15.3 FO4 delays per stage between latches). This algorithm gives shorter latency and cycle time compared to fastest algorithm [2]. Many DSP applications such as fast Fourier transform (FFT) and discrete cosine transform (DCT) butterfly operations can benefit from the fused floating-point add–subtract unit. Therefore, the improved fused floating-point add–subtract unit will contribute to the next generation floating-point arithmetic and DSP application development [3].

A floating-point fused add-subtract unit is described that performs simultaneous floating-point add and subtract

operations on a common pair of single-precision data in about the same time that it takes to perform a single addition with a conventional floating-point adder [4].

The fast Fourier transform is a case in point, it uses a complex butterfly operation. For a radix-2 implementation, the butterfly consists of a complex multiply followed by the complex addition and subtraction of the same pair of data. These butterfly operations can be implemented with two fused primitives, a fused two-term inner product and a fused add subtract unit [5].

3. Traditional Floating Point Add-subtract Unit

A direct way to implement the floating-point add–subtract operation is to use two identical floating-point adders in parallel as shown in Fig. 2. One of the adders performs an addition and the other performs a subtraction to produce the sum and difference simultaneously. A traditional floating-point adder performs addition and subtraction simultaneously but it requires separate adders to perform operation simultaneously.

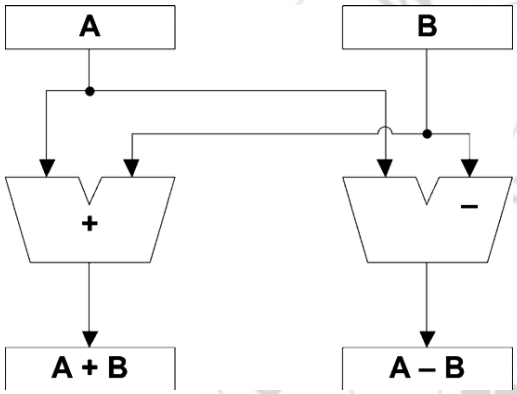

Figure 2: Discrete parallel floating add-subtract unit

4. Fused Floating Point Add-subtract Unit

The discrete floating-point add–subtract unit produces the sum and difference simultaneously by executing two identical floating-point additions. However, much of the logic such as exponent comparison, significand swap and alignment in the two floating-point adders is nearly the same for the two operations.

In order to reduce the overhead, a fused floating-point add–subtract unit shares the common logic for the two operations. The fused floating-point add–subtract unit produces the sum and difference results simultaneously by executing the shared logic such as the exponent comparison, significand swap and alignment. Also, the fused floating-point add–subtract unit performs only one significand addition and subtraction for each operation. It performs the operation by considering sign decision table.

Table 1: Sign decision table

A sign	B sign	Comp.	Sum	Difference
+	+	$\|A\| < \|B\|$	$\|A\| + \|B\|$	$-(\|B\| - \|A\|)$
+	+	$\|A\| > \|B\|$	$\|A\| + \|B\|$	$\|A\| - \|B\|$
+	–	$\|A\| < \|B\|$	$-(\|B\| - \|A\|)$	$\|A\| + \|B\|$
+	–	$\|A\| > \|B\|$	$\|A\| - \|B\|$	$\|A\| + \|B\|$
–	+	$\|A\| < \|B\|$	$\|B\| - \|A\|$	$-(\|A\| + \|B\|)$
–	+	$\|A\| > \|B\|$	$-(\|A\| - \|B\|)$	$-(\|A\| + \|B\|)$
–	–	$\|A\| < \|B\|$	$-(\|A\| + \|B\|)$	$\|B\| - \|A\|$
–	–	$\|A\| > \|B\|$	$-(\|A\| + \|B\|)$	$-(\|A\| - \|B\|)$

5. Dual Path Fused Floating Point Add-Subtract Unit

In order to achieve a high-performance fused floating-point add–subtract unit, in this paper proposes a dual-path approach. Most high-speed floating-point adders employ the dual-path algorithm. The dual-path algorithm skips the normalization step depending on the exponent difference. Since the normalization after the subtraction is one of the bottlenecks in the fused floating-point add–subtract unit. Hence the dual-path approach improves the performance.
The dual-path approach consists of two methods:
A. Far Path Logic
B. Close Path Logic

6. Pipelined Fused Floating Point Add-Subtract Unit

To increase the throughput, pipelining can be applied. The proposed dual path design will be split into two pipeline stages. By properly arranging the components, latencies of the two pipeline stages will be balanced so that throughput of the entire design will be increases. In this several components will be executes in parallel.In order to achieve a proper pipelining in fused floating-point add–subtract unit, the latencies of the components in the proposed design need to be investigated. Considering the latencies of components and their parallel execution, the proposed design is split into two pipeline stages. Each pipeline stage is executed every cycle so that the largest latency determines the throughput of the design. In pipelined fused floating-point add–subtract unit it contains two stages. Each stage requires latches as many data and control signals are passes from the first stage to the next. Although the latches and control signals in the pipeline stages increase the total area, latency and power consumption, the throughput is increases compared to the non-pipelined dual path design.

7. Conclusion

The floating point add-subtract unit is useful for digital signal processing application(DSP) such as FFT & DCT butterfly operations. In this paper we have apply dual path algorithm & pipelining to the fused floating point add-subtract unit & it compares the area, latency & power consumption with the traditional implementation.

The fused floating point add-subtract unit saves area & power consumption compared to the traditional discrete floating point add-subtract unit by sharing common logic.

The fused floating point add-subtract unit also reduces latency due to its simplified control logic.

The dual path fused floating point add-subtract unit reduces latency compared to discrete design by performing add-subtract operation for each case in parallel.

By pipelining implementation it increases the throughput of dual path fused floating point add-subtract unit.

8. Acknowledgment

I wish to thank Professor S. S. Tamboli for many of her ideas which are included in this paper, and for her continuous support.

References

[1] Jongwook Sohn, *StudentMember, IEEE*, and Earl E. Swartzlander, Jr., *Life Fellow, IEEE* " Improved architectures for fused floating point add-subtract unit," IEEE transaction 2012
[2] P. M. Seidel and G. Even, "Delay-optimized implementation of IEEE floating-point addition," *IEEE Trans. Comput.*, vol. 53, no. 2, pp. 97–113, Feb. 2004.
[3] E. E. Swartzlander, Jr. and H. H. Saleh, "FFT implementation with fused floating-point operations," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 284–288, Feb. 2012.
[4] H. H. Saleh and E. E. Swartzlander, Jr., "A floating-point fused add–subtract unit," in *Proc. 51st IEEE Midwest Symp. Circuits Syst.*, 2008, pp. 519–522.
[5] E. E. Swartzlander, Jr. and H. H. Saleh, "Fused floating-point arithmetic for DSP," in *Proc. 42nd Asilomar Conf. Signals, Syst., Comput.*,2008.
[6] T. Lang and J. D. Bruguera, "Floating-point fused multiply-add with reduced latency," *IEEE Trans. Comput.*, vol. 53, no. 8, pp. 988–1003, Aug. 2004.
[7] H. H. Saleh and E. E. Swartzlander, Jr., "A floating-point fused dotproduct unit," in *Proc. IEEE Int. Conf. Comput. Design*, 2008, pp. 427–431
[8] Beaumont-Smith, N. Burgess, S. Lefrere, and C. Lim, "Reduced latency IEEE floating-point standard adder architectures," in *Proc. 14th IEEE Symp. Comput. Arithmetic*, 1999, pp. 35–43.