Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-7, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Review Paper on Fused Floating Point Arithmetic Unit for DSP Application

Priya Purohit[1], Monika Gupta[2], Abhay Sharma[3]
[1]M Tech Student, VLSI Design, F.O.T.U.T.U Dehradun
[2]Assistant Professor, VLSI Design, F.O.T.U.T.U Dehradun
[3]Assistant Professor, Graphic Era Hill University, Dehradun

Abstract*: For DSP application that requires a large dynamic range floating point implementations are more suitable than fixed point representation Floating point arithmetic is much useful in the implementation of various DSP applications as it allows the designer and the user to concentrate on the algorithms and architecture without worrying about the numerical issues. Many applications use floating point hardware to perform DSP tasks in real time and hence overcome the limitations imposed by the use of fixed point numeric systems. The multi-functional floating-point unit includes a multiplier unit, a fused Add-Subtract unit which uses hardware more efficiently in comparison to the separate unit blocks for each operation. This method reduces the area of the designed block but the speed of operation is also reduced. The blocks are reduced based on the common operation between each designed unit.*

1. Introduction

Fixed-point arithmetic has been used for the longest time in computer arithmetic calculations due to its ease of implementation compared to floating-point arithmetic and the limited integration capabilities of available chip design technologies in the past. The design of binary fixed-point adders, multipliers, subtractors, and dividers has been covered in numerous textbooks and conference papers. However, advanced technology applications require a data space that ranges from the infinitesimally small to the infinitely large. Such applications require the design of floating-point hardware. A floating point number representation can simultaneously provide a large range of numbers and a high degree of precision. As a result, a portion of most microprocessors are often dedicated to hardware for floating point computation. Floating-point data representation provides a wide dynamic range, which makes DSP designers free from scaling and overflow/underflow concerns that arise with fixed point representations.

Much research has been done on the floating point fused multiply add (FMA) unit [1]. It has several advantages over discrete floating point adders and multipliers in a floating point unit design. An FMA unit can reduce the latency of an application that executes a multiplication followed by an addition; this unit may entirely replace a processor's floating point adder and floating point multiplier. Many DSP algorithms have been rewritten to take advantage of the presence of FMA units. For example in [2] a radix-16 FFT algorithm is presented that speeds up FFTs in systems with FMA units. High throughput digital filter implementations are possible with the use of FMA units [3]. FMA units are utilized in embedded signal processing and graphics application [4], used to perform division [5], argument reduction [6], and this is why the FMA has become an integral unit of many commercial processors such as those of IBM [7], HP [8] and Intel [9].

Similar to operations performed by a FMA, in other fields, both the sum and difference of a pair of operands are needed for subsequent processing. Another frequently used DSP operation is calculating the sum of the products of two pairs of operands (dot product). For example, these operations are required in computation of FFT and DCT butterfly operations.

Traditional floating- point unit may perform these operations in a serial fashion. These operations may be performed in parallel, which is expensive.

1.2 Traditional Floating Point Add Subtract Unit

In traditional floating-point hardware arithmetic operations may be performed in a serial fashion which limits the throughput. Alternatively, the addition and subtraction may be performed in parallel with two floating-point adders which is expensive (in silicon area and in power consumption)

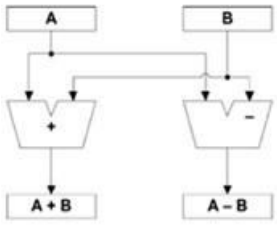

Figure 1: Discrete parallel floating-point add-subtract unit

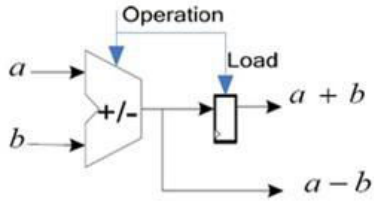

Figure 2: Serial floating-point add-subtract unit

The fused add subtract (FAS) unit executes at the same speed and provides substantial saving in area and power consumption when compared to the conventional parallel approach used to realize the simultaneous add subtract.

1.3 Fused Add-Subtract unit

In many algorithms in DSP and other fields both of the sum and difference of a pair of operands are needed. For example, this is required in computation of the FFT & DCT butterfly operations. The uses of fused add-subtract (FAS) unit accelerates the butterfly operation. Alternatively, add and subtract may be performed in parallel with two independent floating-point adders which is expensive (in silicon area and in power consumption).

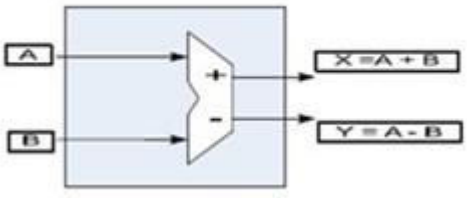

Figure 3: Fused floating-point add-sub unit

1.4 Fused Dot Product Unit

The numerical operation performed by this unit can be used to improve many DSP algorithms. Specifically, multiplication of complex operands benefits greatly from the FDP. For example, implementations of the FFT butterfly operation, the DCT butterfly operation, vector multiplication and the wavelet transform could all benefit largely from the speed up offered by this unit.
The numerical operation performed by this unit can be used to improve many DSP algorithms.
The FDP unit [11] performs the following operation:

$$y = a * b \pm c * d$$

The block diagram of the FDP unit is shown in figure 4.

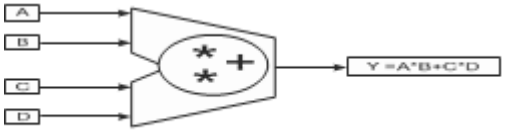

Figure 4: Fused Dot Product Unit Concept.

2. Review on Floating Point Arithmetic units

2.1 Floating-Point Fused Add-Subtract unit

A circuit device includes a first input to receive a first operand and a second input to receive a second operand. The circuit device further includes sign logic to receive sign bits associated with the first and second operands and to determine first and second sign output values and includes exponent difference and select logic to receive exponent bits from the first and second operands and to determine an exponent adjustment value and a shift control signal. The circuit device also includes first and second 2:1 multiplexers to select from the first and second operands to produce first and second values and includes a shift circuit adapted to shift the second value based on the shift control signal. Further, the circuit device includes an add/round and post-normalize circuit to add the first value and the shifted second value to produce a sum and to round and normalize the sum to produce a sum output and includes a subtract/round and post-normalize circuit to subtract the first value and the shifted second value to produce a difference and to round and normalize the difference to produce a difference output. The circuit device further includes logic to combine the first sign output value and the sum output to produce a sum result at a first output and to combine the second sign output value and the difference output to produce a difference result at a second output.

The architecture of the fused add-subtract unit [10] is derived from the floating-point add unit. The exponent difference, significant shift and exponent adjustment functions can be performed once with a

single set of hardware, with the results shared by both add and the subtract operations. The blocks with green background are additional blocks used to perform the subtract operation, and the blocks with yellow background are similar to the floating-point add blocks, but with extended functionality to calculate the sign and exponent for the new subtract operation.

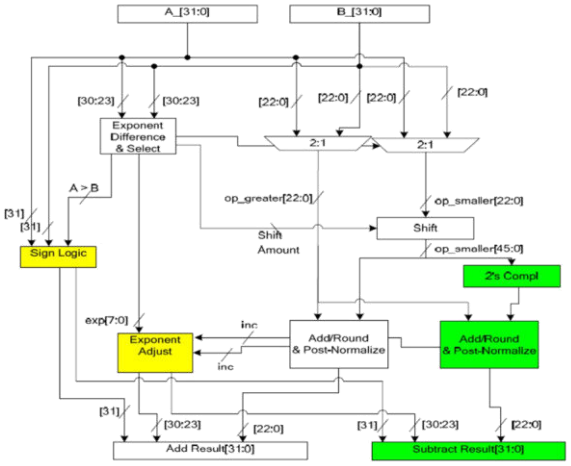

Figure 5: Floating-Point Add-Subtract Unit [13]

2.2 Floating-Point Fused two term Dot-Product unit

This unit performs single-precision floating-point multiplication and addition operations on two pairs of data in a period of time that is only 150% greater than that required for a single conventional floating-point multiplication. This unit uses the IEEE-754 single-precision format and supports all rounding modes. The fused dot-product unit occupies about less area needed to implement a parallel dot-product unit using conventional floating-point adders and multipliers implemented with the same process. The speed of the fused dot-product is faster than the conventional parallel approach. The numerical result of the fused unit is more accurate because only one rounding operation is used, versus three for the conventional approach. Calculating the sum of the products of two sets of operands is a frequently used operation in the computation of the FFT and DCT butterfly operations. In a traditional implementation, the dot-product is performed with two multiplications and an addition. These operations may be performed in a serial fashion by utilizing a single adder and a single multiplier with multiplexers and registers for intermediate results. It has low throughput, has a small area and low power consumption. Alternatively, the multiplications may be performed in parallel with two independent multipliers followed by an adder. This parallel approach is expensive in silicon area and in power consumption. The floating-point fused two-term dot-product unit architecture is derived from the architecture of a floating-point fused multiplier-adder.

2.3. Floating-Point Fused Add-Multiply Unit

Multiplication operation can be done as a successive addition which introduces delay in the processing. In which implementation of this technique with delay constraints leads to more challenging because the internal modules generate delay in propagation of processed output of adder unit. With a solution to this problem, the optimized method of multiplier and adder units are fused in a single module with modified booth encoding called Fused add multiply (FAM) is designed [12].

The first FMA is introduced in 1990 by IBM RS/6000. After that FMA is implemented by several companies like HP, MIPS, ARM and Intel. It is a key feature of the floating-point unit because it greatly increases the floating-point performance and accuracy since rounding is performed only once for the result addition-multiplication rather than twice for the multiplier and then for the adder.

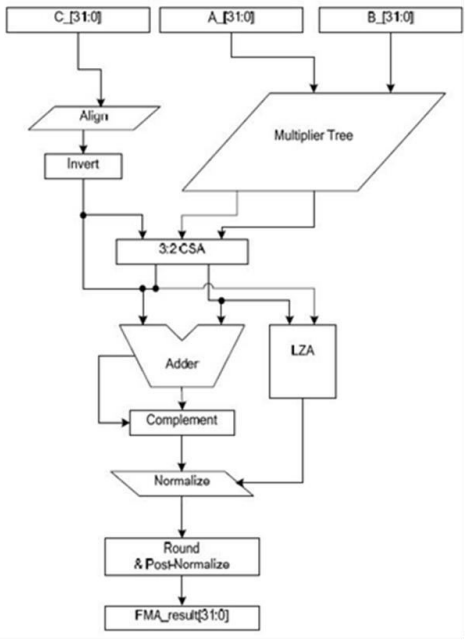

Figure 6: Fused Multiplier-adder architecture [15]

3. The Fast Fourier Transform (FFT) algorithm

A Fast Fourier Transform, or FFT, is not a new transform, but is a computationally efficient

algorithm for the computing the DFT. While it produces the same result as the other approaches, it often reduces the computation time by a factor of ten or more for large sequences. Gauss was the first to propose the techniques that we now call the fast Fourier transform (FFT) for calculating the coefficients in a trigonometric expansion of an asteroid's orbit in 1805. FFTs achieved widespread awareness and impact with the Cooley and Tukey algorithm published in 1965.

There are two flavors of the FFT algorithm; decimation in time (DIT) where the time domain sequence is split into even and odd parts for processing, or decimation in frequency (DIF) where the frequency components are divided into even and odd parts for processing. The DIF and DIT are both equivalent algorithms and it is straight forward to convert from one to the other. Both the DIT and DIF can accept inputs either in order or in bit reversed order to produce bit reversed or in order outputs, respectively.

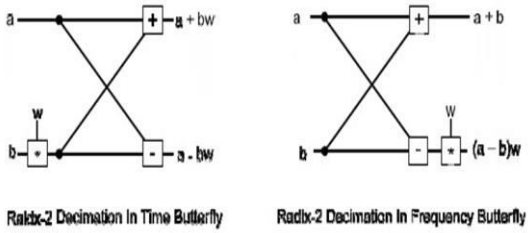

Figure 7: Radix-2 Butterflies

3.1. Radix-2 Butterfly Design Approach

There are multiple conventional approaches that can be taken for an implementation of the floating-point radix-2 FFT butterfly function.

Two of the possible implementations are the parallel approach and the serial approach. The parallel implementation uses six adders and four multipliers that operate in parallel. The serial implementation uses two adders, a multiplier, as well as multiplexers and storage elements to realize the butterfly function.

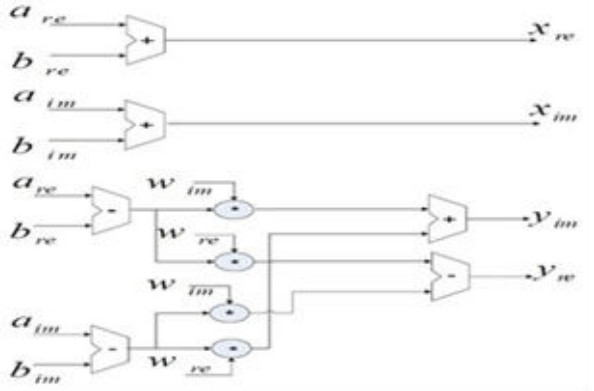

Figure 8: Parallel implementation of Radix-2 Decimation in frequency FFT Butterfly unit

4. Results

Table 1 [13] provides comparison of Adder unit, Add-Sub unit and Multiplier unit on the basis of Power analysis.

Table 1: Power analysis

	Adder unit	**Add-Sub unit**	**Multiplier**
Power Analysis	136.02 mw	190.04mw	124.38mw

The area and latency of the two conventional approaches and the FDP unit are compared in Table 2 [14].

Table 2: Comparison of Radix-2 Butterfly approaches

Unit	**Area (µm2)**	**Latency (ps)**	**Power (µW)**
FP Adder	3,811	1,644	1,302
Fused Add-Sub	5,947	1,804	2,051
FP Multiplier	9,482	2,721	5,877
Fused Dot-Product	16,104	4,621	7,205

5. Conclusion

This paper concludes that, after comparing conventional and fused floating point arithmetic units, the fused approach is best suited for DSP applications. The fused dot product is intermediate in area between the conventional serial and the conventional parallel approach and about half that of the conventional serial approach. Its latency is about 80% of that of the conventional parallel approach and about half that of the conventional serial approach.

6. Future Work

The fused add-subtract and fused two-term dot-product primitive units can be used to realize many other DSP algorithms, including the basic butterfly computation of the discrete cosine transform and many forms of the wavelet transform.

The proposed fused add-subtract and two-term dot-product designs were implemented with no pipelines. The two units could be redesigned employing pipelining to achieve higher operation speeds. If proper pipeline gating were employed, then power consumption could be reduced as well.

The fused two-term dot-product unit can be modified to perform two-term addition, two-term subtraction, and fused multiply add. If the implementation results show a reasonable overhead over an equivalent fused multiply-add unit then the enhanced fused two-term dot-product unit could be used as a building block for microprocessors and digital signal processors.

7. References

[1] E. Hokenek, R. K. Montoye and P. W. Cook, "Second-generation RISC floating point with multiply-add fused," IEEE Journal of Solid- State Circuits, vol. 25, pp. 1207-1213, 1990.
[2] D. Takahashi, "A radix-16 FFT algorithm suitable for multiply-add instruction based on Goedecker method," Proceedings 2003 International Conference on Multimedia and Expo, vol. 2, July 2003, pp. II-845-II-848.
[3] C. Xu, C. Y. Wang and K. K. Parhi, "Order-configurable programmable power-efficient FIR filters," Proceedings 3rd International Conference on High Performance Computing, December 1996, pp. 357-361. 770
[4] C. Hinds, "An Enhanced Floating Point Coprocessor for Embedded Signal Processing and Graphics Applications,"
[5] A. D. Robison, "N-Bit Unsigned Division Via N-Bit Multiply-Add," Proceedings of the 17th IEEE Symposium On Computer Arithmetic, pp. 131-139, 2005.
[6] R.-C. Li, S. Boldo and M. Daumas, "Theorems on Efficient Argument Reductions," Proceedings of the 16th IEEE Symposium on Computer Arithmetic, pp. 129-136, 2003.
[7] F. P. O'Connell and S. W. White, "POWER3: The Next Generation of PowerPC Processors," IBM Journal of Research and Development, vol. 44, pp. 873-884, 2000.
[8] A. Kumar, "The HP PA-8000 RISC CPU," IEEE Micro Magazine, vol. 17, Issue 2, pp. 27-32, April, 1997.
[9] B. Greer, J. Harrison, G. Henry, W. Li and P. Tang, "Scientific Computing on the Itanium Processor," Proceedings of the ACM/IEEE SC 2001 Conference, pp. 1-8, 2001.
[10] Hani Saleh and Earl Swartzlander, Jr., "A Floating-Point Fused Add- Subtract Unit," 2008 IEEE Midwest Symposium on Circuits and Systems (MWSCAS), Knoxville, TN 2008.
[11] Hani Saleh and Earl Swartzlander, Jr., "A Floating-Point Fused Dot-Product Unit," IEEE International Conference on Computer Design (ICCD), Lake Tahoe, CA, pp. 427-431, 2008.
[12] M. Nepolean, K. Sivasubramanian, "An optimized design for fused Add-Multiply operation"-2016.
[13] Jyoti Sharrma, Pabbisetty Tarun, et al. "Fused Floating point Add-Subtract unit."-2015 online international conference on Green Engineering and Technologies.
[14] E. E. Swartzlander, Jr. and H. H. Saleh, "Fused floating-point arithmetic for DSP," 2008.
[15] E. E. Swartzlander, Jr. and H. H. Saleh " FFT Implementation with Fused Floating- Point Operation.