IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

Design of Efficient Fused Floating-Point Units for DSP Applications

Bhagirth Sharma[1], Mrs. Suman Dagar[2]

M. Tech Scholar, NGF College of Engineering and Technology, Palwal, Haryana[1]

Assistant Professor, NGF College of Engineering and Technology, Palwal, Haryana[2]

Abstract: The fused floating point units are advantageous for various Digital Signal Processing (DSP) applications such as butterfly operations of Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT) and complex multiplication. In this thesis, two floating point fused units are proposed: 1) Floating point Fused dot product unit, 2) Floating point fused add subtract unit. Fused dot product unit performs the multiplication and addition of two pairs of operands. Fused add subtract unit perform the addition and subtraction of same operands simultaneously. In order to enhance the performance for the fused dot product unit, DADDA multiplication algorithm is applied. FDP unit with DADDA Multiplier improved the speed by 61.1 % compared to the FDP unit with Booth multiplier. The fused dot product unit and fused add subtract unit are implemented for single precision and synthesized in 90nm technology.

Keywords: Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Digital Signal Processing (DSP)

I. INTRODUCTION

Floating Point (FP) arithmetic has the power of implementation of various applications in Digital Signal Processing (DSP) as the designer can focus on the architecture and algorithms without any take care of numerical issues like scaling, underflow and over ow. Earlier, fixed point arithmetic is used by various DSP application units for the reason of the high cost (in silicon area, power consumption and delay) of FP arithmetic units. Many DSP applications could perform DSP tasks in real time using FP hardware and thus overcome the constraints by the use of fixed-point numeric systems. Digital Signal Processing (DSP) can be divided into two groups that is Fixed point and Floating point. These indicate to the format used to keep and form the arithmetic representations of data. Fixed point Digital signal processors are designed to represent and manipulate integers i.e. positive and negative numbers by using possible bit values (2^16). Floating point Digital signal processors represent and form the rational numbers by using minimum of 32 bits in a manner identical to the scientific notation, where a number used represented with a significant or mantissa or fraction and an exponent (for instance, M 2E, where 'M' is significand or mantissa and E is the exponent), taking up to possible bit values (2^32).

II. DESIGN METHODOLOGY

In this thesis, fused floating-point units are explored in order to make efficient designs in terms of speed, area, power consumption. In order to design and implement the fused floating-point units, general steps are taken to do this work:

- Study literature about the IEEE -754 floating point arithmetic concepts, architectures, and implementations.
- Study literature about fused floating point units' concepts, architecture and implementations.
- Develop the architecture for the floating point fused add subtract and fused dot product units.
- Design the floating-point units (FDP, FAS) in Verilog RTL language and simulate using Modelsim tool.
- ASIC implementation using the Synopsys tool.

III. FLOATING POINT ADDER

The FP adder is taking two operands as inputs and generates a truncated result. In distinction to the fixed-point units, a FP adder is a much complex than the FP multiplier because of the alignment and normalization process. The comparison of the exponent is done by the exponent compare logic to determine which one is bigger. The comparison result of the exponent and the difference are taken for the significand swapping, alignment procedure and sign logic process. The significand swap logic is taking the two significands as input and determines the significand of the bigger and smaller operand based upon the exponent comparison. The two significands are also passing to the alignment and sticky logic. The significand of the smaller operand is shifted by exponent difference amount and the least significant bit (LSB) of the shifted significand are discarded by the sticky logic.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

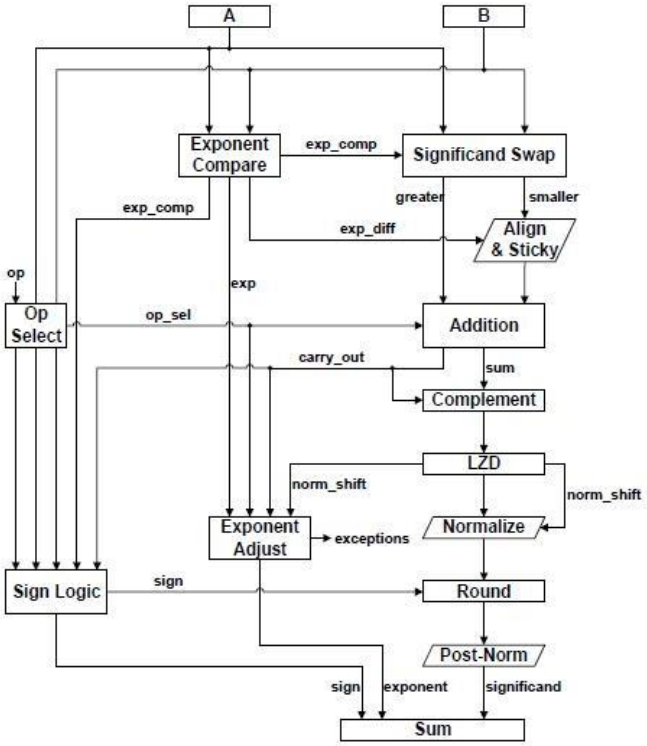

Figure 2: Floating Point adder

IV. FLOATING POINT MULTIPLIER

The FP multiplier will take two operands as inputs and produces a truncated result as a product. Even if the floating-point multiplier is simpler in terms of the architecture, it has needed higher logic area and power consumption in comparison of the FP adder. The exponent sum logic produces the sum of the two exponents. The result will pass to the exponent adjust logic. to the exponent adjust logic. The multiplier tree will take the two significands as inputs and perform the reduction tree to produce the sum and carry. The significand pair is aligned to the number of final significand bits including round, guard, and sticky bits in order to cut down the significant addition

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 5, May 2019

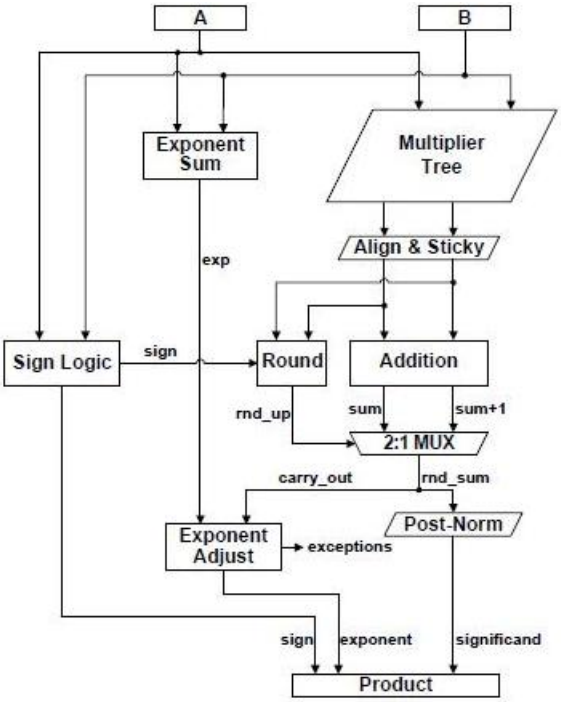

Figure 1: Floating Point Multiplier

V. FUSED MULTIPLY ADD UNIT

The multiply add is a basic operation in several applications of digital signal processing. For instance, Digital Signal Processing (DSP) introduced algorithms that use the (A x B) + C equation as a single instruction Quindell et al. (2007). This is introduced by the survey which shows that almost fifty percent of the multiply instructions are followed by add or subtract instructions. Some of the steps involved in the multiply add operation such as the multiplication and the addition of the result with another operand could be performed simultaneously Swartz lander & Saleh (2008). Therefore, the multiply add can be taken as an only one operation called fused multiply add (FMA). The benefits of the direct implementation of the FMA are given below:

- The (A x B) + C is performed with single rounding rather than two, therefore much accuracy is gained.
- Various building blocks will be shared. Hence, it produces reduced area.
- Critical path delay will be reduced by using the efficient parallel implementation.

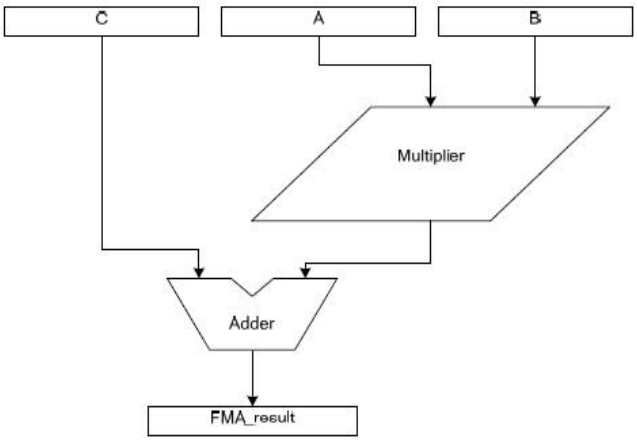

Figure 3: Fused Multiply Add

VI. FUSED ADD SUBTRACT UNITS

Several Digital Signal processing (DSP) applications such as the Fast Fourier Trans-form (FFT) and Discrete Cosine Transform (DCT) needs both the sum and difference of a pair of the two operands for the butter y operations execution. The floating point add subtract unit is suitable for these type of applications by generating the sum and difference concurrently Sohn & Swartzlander (2012). The floating point add subtract unit will take two operands as input and

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 5, May 2019

generates the sum and difference concurrently. There are two ways to design the floating point add subtract unit. The two design ways for the floating point add subtract unit are given: 1) Discrete floating point add subtract unit and 2) Fused floating point add subtract unit.

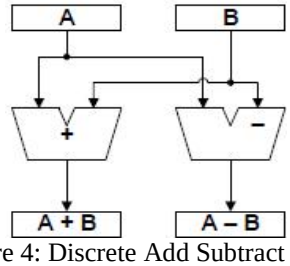

Figure 4: Discrete Add Subtract Unit

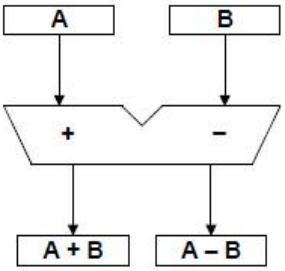

Figure 5: Fused Add Subtract Unit

VII. RESULT-FUSED DOT PRODUCT UNIT

The two-term floating point fused dot product unit takes the four operands as inputs in the single precision format and generates the output result in the single precision format. Figure 4.1 shows the simulation in the Modelsim Simulator and the result is

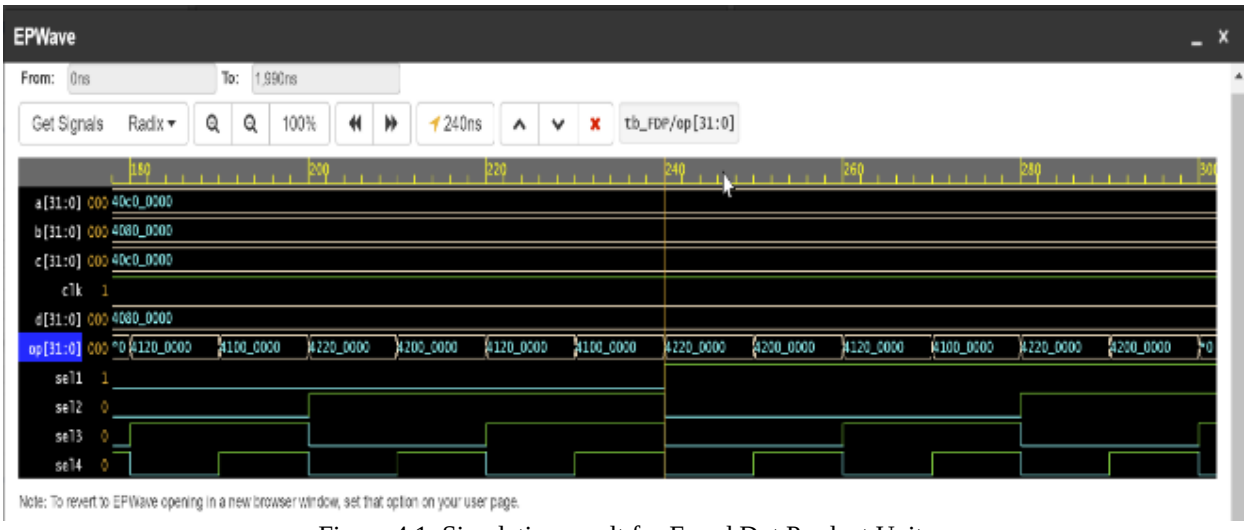

Figure 4.1: Simulation result for Fused Dot Product Unit

A=01000000110000000000000000000000,
B=10000001000000000000000000000000,
C=10000001100000000000000000000000,
D=10000001000000000000000000000000
op=01000001000000000000000000000000

VIII. RESULT-FUSED ADD SUBTRACT UNIT

The two-term floating point fused Add Subtract unit takes the four operands as inputs in the single precision format and generates the output result in the single precision format. Figure 4.2 shows the simulation in the Modelsim Simulator and the result is in IEEE single precision format and the figure shows the waveform in binary number system.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

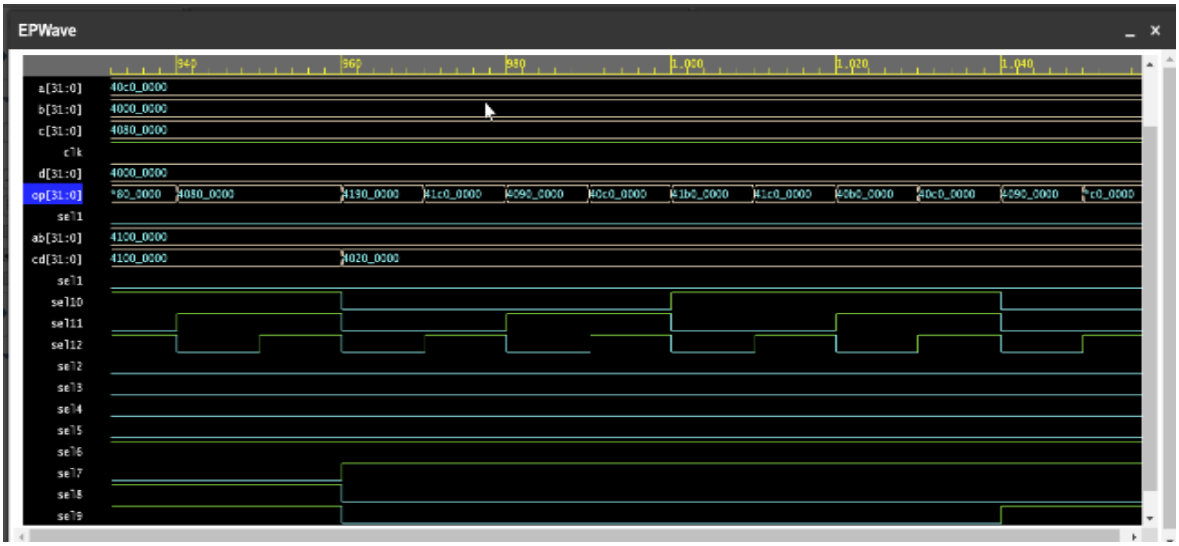

Figure 4.2: Simulation result for Fused Add Subtract Unit

The inputs and output values are given below:
a = 10000001100000000000000000000000
b = 01000000100000000000000000000000
c = 01000000110000000000000000000000
d = 01000000100000000000000000000000
op= (A + B) + (C + D)
op= (A + B) (C + D)
op =01000000111000000000000000000000

IX. CONCLUSION

To build special purpose DSP hardware in Systems On Chips (SoC) at present, several FP primitives like FP multipliers and FP adders are required. In several DSP algorithms (like Fast Fourier Transforms (FFT)), the result of the addition and sub-traction for the same two operands are required at the same time. At present, this could be done with either an only one adder and two cycles (one for the addition and one for the subtraction) or with two discrete adders and one cycle. The sum of the products of two pairs of operands is a much frequent operation in FFT which require one floating point add and two floating point multiplies to be performed. To do these operations; two ways are currently in use. The first way is to use one FP multiplier and one FP adder with storage to do the operations in sequential manner, that is the much useful from power and an area point of view, but too slow for several applications. Another way is to use an adder and two multipliers, to do these operations in parallel manner. This provides the required speed; however, the large area and high-power consumption have a dominant impact on many applications such as cell phone and handhold devices.

REFERENCES

[1]. Montoye, R. K., Hokenek, E. & Runyon, S. L. (1990), `Design of the ibm risc system/6000 floating-point execution unit', IBM Journal of research and development 34(1), 59{70.}
[2]. Quinnell, E. Swartzlander, E. E. & Lemonds, C. (2007), Floating-point fused multiply-add architectures, in `Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on', IEEE.
[3]. Saleh, H. & Swartzlander, E. E. (2008), A floating-point fused add-subtract unit, in `Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on', IEEE
[4]. Schmookler, M. S. & Nowka, K. J. (2001), Leading zero anticipation and detection-a comparison of methods, in `Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium on', IEEE, pp. 7{12.
[5]. Sohn, J. & Swartzlander, E. E. (2012), `Improved architectures for a fused floating-point add-subtract unit', IEEE Transactions on Circuits and Systems I: Regular Papers 59(10), 2285{2291.
[6]. Sohn, J. & Swartzlander, E. E. (2013), Improved architectures for a floating-point fused dot product unit, in Computer Arithmetic (ARITH), 2013 21st IEEE Symposium on', IEEE, pp. 41{48}.
[7]. Sohn, J. & Swartzlander, E. E. (2016), `A fused floating-point four-term dot product unit', IEEE Transactions on Circuits and Systems I: Regular Papers 63(3), 370{378.
[8]. Swartz lander, E. E. & Saleh, H. H. (2008), Fused floating-point arithmetic for dsp, in Signals, Systems and Computers, 2008 42nd Asilomar Conference on', IEEE, pp. 77{771.Swartzlander, E. E. & Saleh, H. H. (2012), `Fft implementation with fused. Floating-point operations', IEEE transactions on computers 61(2), 284{288.Wang,

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 5, May 2019

[9]. Wang, F., Wei, S. & Li, Z. (2016f), `A pipelined area-efficient and high-speed recon Gurable processor for floating-point t/i t and dct/idct computations', Microelectronfics Journal 47, 19{30.Zuras, D. Cowlishaw, M., Aiken, A.Applegate, M., Bailey, D., Bass, S.,Bhandarkar, D., Bhat, M., Bindel, D., Boldo, S. et al. (2008),Ieee standard for floating-point arithmetic', IEEEStd 754-2008 pp. 1{70.