Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.917 – 923

RESEARCH ARTICLE

Design and Implementation of a New Multi-Functional Fused Dot Product in FPGA

Amir Baniasad Azad[1], Amir Sabbagh Mollahoseini[2]

[1]Department of Computer Kerman Science and Research branch, Islamic Azad University, Iran
[2]Department of Computer Islamic Azad University, Kerman, Iran

[1] a.azad@srbiau.ac.ir; [2] sabbagh@iauk.ac.ir

Abstract*— *The floating-point Fused Dot Product (FDP) is one of the most used operations in digital signal processing (DSP), Graphics Processor Units (GPU), etc. The design of multi-functional floating-point units is a method for using hardware more efficient in order to reduce the cost of conventional hardware units.
In this paper, a floating-point Fused two term Dot Product (AB ± CD) with the new capabilities offered. Because the fused operations performed one rounding operation, not only the accuracy increased but also delay and cost reduced due to the sharing of hardware .If we want to use this hardware for typical floating-point operations such as floating-point addition and multiplication, we normally can do one addition, subtraction and multiplication, while the architecture presented in this paper as a multi-functional double precision FDP has the ability to do two parallel floating point addition, subtraction or multiplication with a same precision(double precision).After implementation of this architecture, in comparison with a conventional architecture in FPGA, we find that the architecture was designed with increased 10.23% slice number and 12.77% delay, while the parallel operations of addition, subtraction and multiplication has doubled.

Keywords- Floating-Point; Fused Dot Product; Multi Functional; FPGA (Field Programmable Gate Array)

I. INTRODUCTION

The floating-point Fused Dot Product (FDP) is one of the most used operations in digital signal processing (DSP), graphics processor units (GPU), etc [1, 2]. The dot product with two term (AB+CD) [3] or four term (AB+CD+EF+GH) [4] usually used for this applications. Because the fused operations performed one rounding operation, not only the accuracy increased but also delay and cost reduced due to the sharing of hardware. Although FPGA designs are flexible and have less cost for implementation of floating-point circuits, there is more delay in them rather than ASIC (Application-Specific Integrated Circuit) implementation. But because of their flexibility, they are common used and many researchers interest in this type of implementation. After presenting the idea of combining multiply and accumulate operations (AB + C), majority of researches have been done in Fused Multiply Add (FMA) field [5] and also the other attempts [6, 7, 8] in which researchers have achieved a variety of improvements. Some ideas raised more efficient with using of hardware units which were introduced as a multi precision unit [9, 10], and multi-functional designs [11].Multi precision designs have two or multiple modes. For example, they are able to work in double precision and in single precision with parallel twice operation. Multi functional designs can do different operations with same hardware units. In[9], a FMA capable of doing calculations in single and double precision presents. This FMA can do one operation in

double precision or two operations in a single precision one. In [11] two multi functional and multi precision FMA with dot product support presented and implemented .In [12], one floating point dual precision addition and multiplication in FPGA as embedded Floating Point Unit(FPU) implemented. In [13] with using Bridge idea with a combination of addition and multiplying units, one FMA was designed. The benefit of this design was little delay and area overhead, no need to change design of adder and multiply hardware units and bridge can adapt these ones to implement one FMA. In [14] a multi functional four term dot product for special application present that can do dot product with parallel multiplications. Past papers indicate multi precision and multi-functional designs were presented in order to decrease the cost of floating-point implementation.The presented architecture is a multi functional FDP which can do five types of floating-point operations in a same precision: one FDP function, two parallel addition functions, two parallel subtraction functions, one addition and one subtraction functions, and two parallel multiplication functions. All of these types operate in double precision. After implementation of this design and compare with conventional FDP, our design has 12.77% more delay and 10.23% more slice numbers, but the number of parallel addition, subtraction and multiplication functions are doubled and using of FDP parts get maximum optimization.
In section2 conventional FDP is introduced, then we present new multi functional FDP in section3 and implement this design in FPGA in section4. Finally, conclusion is in section 5.

II. CONVENTIONAL FDP

In floating-point dot product operation (AB±CD), A, B, C and D are four floating-point inputs and the calculation of this expression needs two multiplication and one addition/ subtraction. The fused dot product uses a combinational hardware unit consist of two multiplier and one adder functions with normalization and rounding units. The fused dot product algorithm is explained below:
Assume a, b, c and d are normalized floating-point numbers and m means mantissa (significant), e means exponent and s is sign of number. Several steps must be done to calculate dot product.

Step1- Significant Multiply, Exponent Difference, Maximum Exponent, Effective Operator
$M_{ab}=m_a * m_b$, $M_{cd}=m_c * m_d$, $E_{ab}=E_a+E_b$, $E_{cd}=E_c+E_d$, $Shift\_amount=E_{diff}=|E_{ab}-E_{cd}|-bias$, $E_p=MAX\{E_{ab}, E_{cd}\}$

, $S_{ab}=S_a$ *xor* S_b,$S_{cd}=S_c$ *xor* S_d , $op_{eff}=s_{ab}$ *xor op xor* s_{cd}

Step2-Alignment, Inversion
$M_{align}=RightShift(M_{smaller},Shift\_amount)$

If op=1 then

$M_{neg}=2's\text{-}complement(M_{align})$

Else

$M_{neg}=M_{align}$

End if

Step3-Addition, Predict number of Leading Zeros
$M_p=M_{bigger}+M_{neg}$, $Shift\_norm=LZA(M_{bigger},M_{neg})$

Step4-Normalization, Rounding, Exponent Adjust, Sign Detection
$M_{normal}=LeftShift(M_p,Shift\_norm)$

$MR=Round(M_{normal})$

$ER=E_p\text{-}Shift\_norm\pm M_p(C_{out})+Round(overflow)$,$SR=SignLogic(S_{ab},S_{cd},op,exp\_comp,M_p(C_{out})$
Final Result is:
$result=(-1)^{SR} * M_R * 2^{ER}$

The figure1 show conventional FDP [15] that acts based on above algorithm.

III. NEW MULTI FUNCTIONAL FDP

Based on previous descriptions one FDP unit which can calculate AB±CD, additionally can do an addition or subtraction with assume one for two operands just as, A±C=A×1±C×1. It also can do one multiplication if assume zero for one operands, for example A×B=A×B±0×D. Therefore, despite a large adder and multiplier,

only one of each kind of operations can execute. Consequently, existing hardware to perform these operations are sub major operation, not be efficiently utilized.

Based on standard IEEE-754 [16], our multi-functional FDP architecture with changes in conventional architecture, the number of parallel floating-point operations include addition unit, subtraction and multiplication, is doubled. Figure 2 is a schematic architecture of our design. Major changes that have taken place in this architecture are as follows:

- Second operator (OP2) has added to the inputs. The operator specifies a parallel addition and subtraction operations.

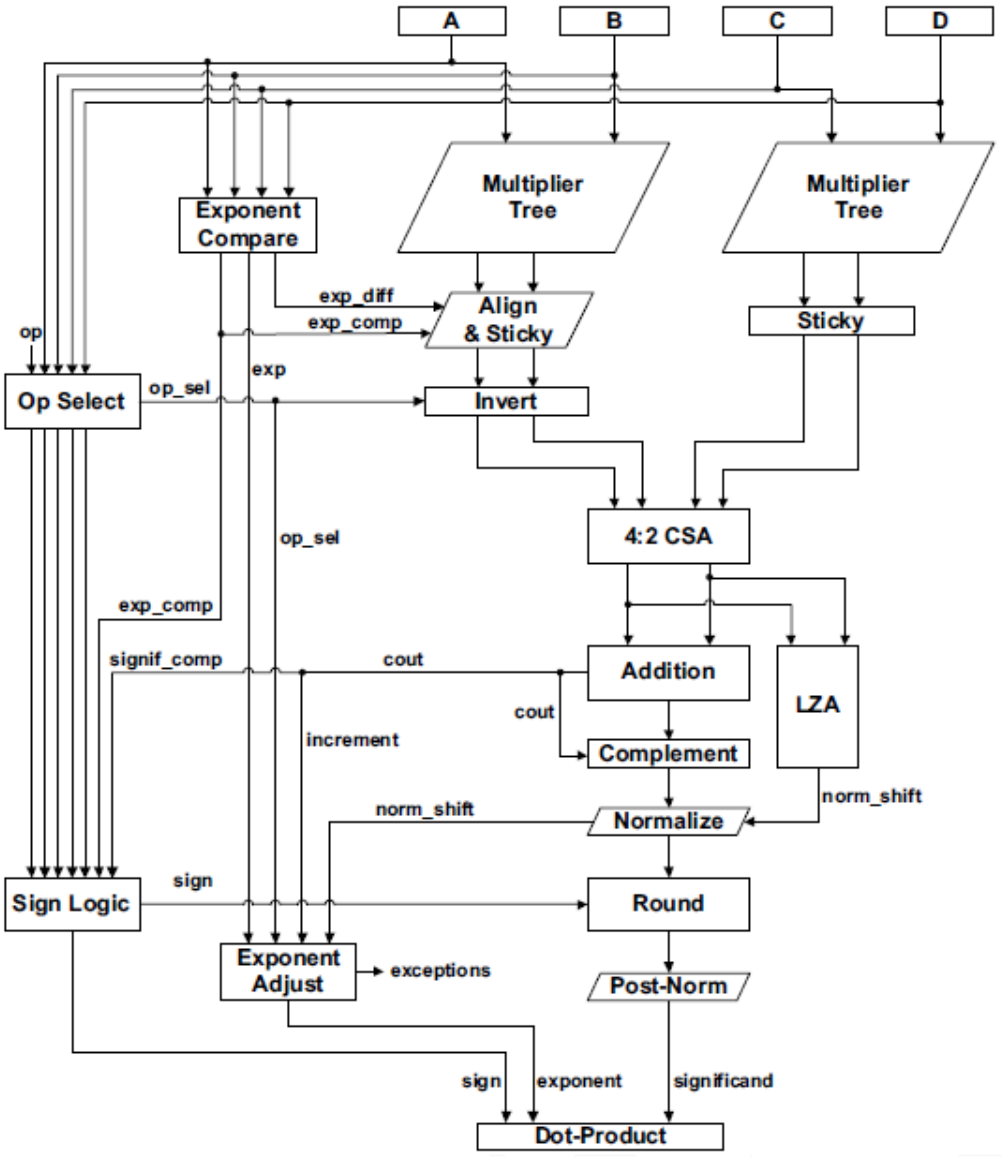

Fig. 1- Conventional FDP[15]

- Two-bit input *mode* is used to control the operations. In Figure 2, the direction control signal has been identified as dashed line. *Mode* indicates three state: (a) If the mode = 00 ,two multiplications are performed in parallel.(b) If mode =01 , depending on the sign of inputs and OP1, OP2 two addition or two subtraction or one addition and one subtraction are performed in parallel.(c) if mode = 10,then the main operation of the circuit, namely FDP is done.
- Two multiplier architecture without any changes to the base architecture placed in the new architecture, but in the operations of addition and subtraction, operands (mantissa) not used of two multiplier functions.

- Unit consists of a number of multiplexers as **Select & Swap** has been added to the circuit. Based on *mode* signal operands to select the next step.two multiplication don't need to use Align & Sticky, Invert and Carry save adder 4 to 2(CSA 4:2) and only Mul Sticky unit added to design to calculate two sticky bits related two multiplications. Also, CSA 4:2 only uses for FDP operation and other operations don't need this component.

- The next unit is **Select for Final Addition**, based on the operation type specified by the *mode* selects two operand for the final addition. All states need this unit to complete operations. Similar to Select & Swap this unit consists of a number of multiplexers .

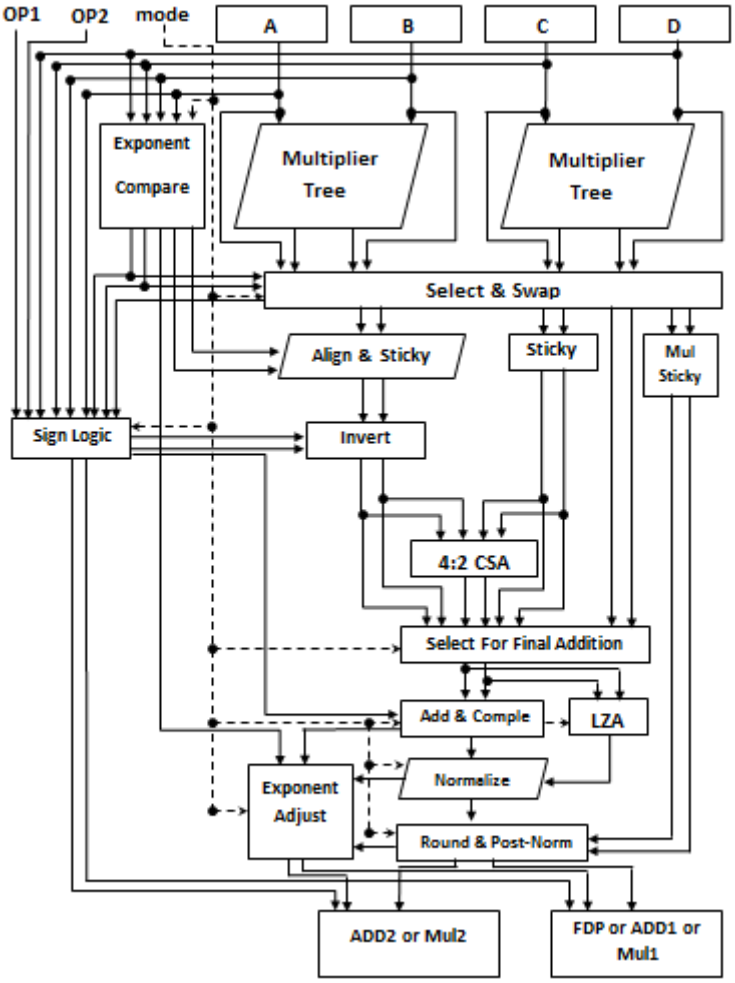

Fig. 2- New Multi Functional FDP

- Final adder in our architecture shown in Figure 3, divided into 56 bit smaller adders. By controlling the carry output of the first adder, two small sum functions for addition, subtraction and multiplication or one big sum for FDP can complete. For second adder, a Flagged Prefix Adder is used that it can produce X + Y and X + Y +1 for X, Y as inputs [20]. Because the two adders produce sum and carry from the first to the second adder is not propagated, the delay is about half.

- All operations except multiplying need LZA (Leading Zero Anticipator) to predict leading zero for normalization as soon as final addition completes. Similar to [18, 19], we use LZA for positive and negative numbers with concurrent correction and split LZA to smaller part [9].
- For normalization, we use Barrel Shifter like [17, 9]. This shifter can operate in two states. Firstly, two 56-bit left shift units are used for two additions, subtractions or multiplications. Secondly, one 112-bit left shift is used in FDP normalization.
- A rounding unit is added for second operation. This unit in FDP is not used.
- Exponent Compare, Exponent Adjust and Sign logic have been changed in a way which can do all needed calculations with the signal *mode* for all operations. Therefore, require additional hardware units specifically some multiplexer are added to choose appropriate input and output.
- A 64-bit register is added to the output for the second operation of addition, subtraction and multiplication. FDP operation just uses the first register.

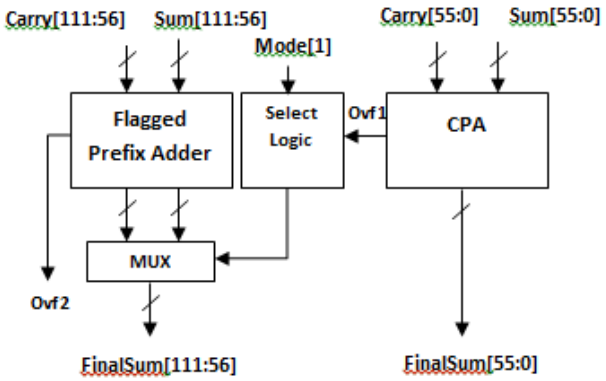

Fig.1- Final Adder Structure

IV. Implementation and Evaluation

For implementation and fair comparison, we describe conventional FDP and new multi functional FDP for double precision as structural VHDL code, an accurate description of all circuits of the same type of hardware is used. The simulation is done by ModelSim in both architectures which were tested by written Test Bench with random data patterns and each architecture was synthesized on the FPGA Virtex-6 by Xilinx ISE Design Suite 12.3 software. Finally, results from synthesis for all these two architectures are shown in Table I.

TABLE II

COMPARISION OF CONVENTIONAL FDP AND MULTI FUNCTIONAL FDP

	Conventional FDP *	**Our Multi functional FDP ****	Comparision ***
Delay (ns)	70.79	**79.83**	12.77%
Slice LUT #	11022	**12150**	10.23%
Functionality	1FDP+ 1Add/Sub+ 1Mul	**1FDP+ 2Add+ 2Sub+ 1Add,1Sub+ 2Mul**	

As be shown in Table III our FDP architecture compared with the conventional architecture in terms of Total Delay has increased at a rate of 12.77%. This delay caused by the multiplexers added in select units and other changes in new architecture, but because the adder divided to smaller adders, the delay is partly compensated.
Number of Slice LUT in FPGA of our architecture 10.23% increase compared to the conventional architecture. This increase was due to changes in the conventional architecture but Normal operations of addition, subtraction, and multiplication can be implemented in this architecture has doubled and the total number and a variety of operations performed in this architecture are enhanced.
In addition, these operations do not require a separate circuit design and Our architecture is designed to support all operations are applicable. Our architecture compared to floating point addition and multiplication is more delay, But for this operation (addition, subtraction and multiplication) can do two operations in parallel and in computing such as digital signal processing (DSP) that series of addition and multiplication can be used continuously has better performance.

V. Conclusion

In this paper, we design and implement a multi-functional fused dot product with the ability to perform addition, subtraction, and multiplication of floating point. Despite the increase in delay and 10.23% and 12.77% cost compared with traditional architecture, the number of ordinary floating-point operations applicable increase to double simultaneously. Also, for the applications with high repeated add and high repeated multiply, this architecture can significantly reduce the delay because can execute two add or multiply in parallel per one cycle, therefore increase total performance.

REFERENCES

[1] E. E. Swartzlander Jr.and H. H. Saleh "*FFT implementation with fused floating-point operations*", IEEE Trans. Computer, vol. 61, no. 2, pp.284 -288, 2012.

[2] H.H. Saleh, "*Fused Floating-Point Arithmetic for DSP*", PhD dissertation, Univ. of Texas, 2008.

[3] E. Quinnell, E. Swartzlander, and C. Lemonds, "*Floating-Point Fused Multiply-Add Architectures,*" Proc. 41st Asilomar Conf. Signals, Systems, and Computers, (ACSSC '07), pp. 331-337, 2007.

[4] D. Kim and L. S. Kim, "*A Floating-Point Unit for4D Vector Inner Product with Reduced Latency*", IEEE Trans. Computers, vol. 58, no.7, pp.890-901, July 2009.

[5] R.K. Montoye, E. Hokenek, and S.L. Runyon, "*Design of the IBM RISC System/6000 Floating-Point Execution Unit*," IBM J. Research and Development, vol. 34, pp. 59-70, 1990.

[6] R. Jessani and C. Olson, "*The Floating-Point Unit of the PowerPC 603e*," IBM Journal of Research and Development. Vol. 40, pp. 559-566, 1996.

[7] T. Lang and J. D. Bruguera, "*Floating-Point Fused Multiply-Add with Reduced Latency*," Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 145-150, 2002.

[8] J.D. Bruguera and T. Lang, *"Floating-Point Fused Multiply-Add: Reduced Latency for Floating-Point Addition*," Proceedings of the 17th IEEE Symposium on Computer Arithmetic. pp. 42-51, June 2005.

[9] L. Huang, L. Shen, K. Dai, and Z.Wang, "*A New Architecture for Multiple-Precision Floating-Point Multiply-Add Fused Unit Design*," Proc. IEEE 18th Symp. Computer Arithmetic, pp. 69-76, June 2007.

[10] L.Huang, S. Ma, L.Shen,Z.Wang, N. XiaoNong, "*low-Cost Binary128 Floating-Point FMA Unit Design with SIMD Support*, " IEEE Trans, Computers, vol. 61, no. 5,pp. 745-751, May 2012.

[11] M. Gok and M.M. Ozbilen, "*Multi-Functional Floating-Point MAF Designs with Dot Product Support*," Microelectronics J., vol. 39, pp. 30-43, Jan. 2008.

[12] Vee Jern Chong and Sri Parameswaran, "*Configurable multi mode embedded floating-point units for FPGAs*", IEEE trans. on very large scale integr.(VLST) systems, vol. 19, no. 11, pp. 2033-44, Nov 2011.

[13] E. Quinnell, E. Swartzlander , and C. Lemonds, "*Bridge Floating-Point Fused Multiply-Add Design*," IEEE Transactions on VLSI Systems, Vol. 16, No. 12. (December 2008), pp. 1727-1731.

[14] Y. Chang, J. Wei , W. Guo , J. Sun "*A multi-functional dot product unit with SIMD architecture for embedded 3D graphics engine*," , 2011 IEEE 54th International Midwest Symp. on Systems, Circuits and Systems (MWSCAS), Vol. 16, No. 12,pp. 1727-1731, December 2008.

[15] J.Sohn, E. Swartzlander, "*Improved Architectures for a Floating-Point Fused Dot Product Unit*," Proc. IEEE 21th Symp. Computer Arithmetic, pp. 41-48,April 2013.

[16] IEEE Standard for Floating-Point Arithmetic, ANSI/IEEE Standard 754-2008, 2008.

[17] R. Kolla, et. al., "The IAX Architecture : Interval Arithmetic Extension", *Technical Report 225*, Universitat Wurzburg,1999.

[18] J. D. Bruguera and T. Lang, "Leading—One prediction with concurrent position correction," *IEEE Trans. Comput.*, vol. 48, no. 10, pp.1083–1097, Oct. 1999.

[19] R. Ji, Z. Ling, X. Zeng, B. Sui, L. Chen, J. Zhang, Y. Feng, and G. Luo, "Comments on 'Leading one prediction with concurrent position correction'," *IEEE Trans. Comput.*, vol. 58, no. 12, pp. 1726–1727,Dec. 2009.

[20] N. Burgess*." The Flagged Prefix Adder for Dual Addition".*In Proceedings of the SPIE Conference: Advanced Signal Processing Algorithms, Architectures, and Implementations, pages 567–575, July 1998.

