An Efficient CSA implementation of MAC unit for Floating Point Operation

Insha Ishtiyaq, Kantesh Kumar Gaurav, Heena Gupta
Panchkula Engineering College (PEC), Panchkula, India

***Abstract* --** In this modern era, floating point operation has become the significant area of research. In real-time DSP operations a MAC operation is required whose speed determines the performance of the system. Several architectures which have already been implemented are being discussed and also their benefits and disadvantages are compared. In this operation, initially a floating-point is converted into IEEE-754 format (which can be single or double precision) required to operate in digital systems. In IEEE-754 there is swapping unit at first place and next is operand based on its value. The arithmetic unit and post-normalization unit are converted to IEEE format. The arithmetic unit required for addition and multiplication operations undergo several steps just to obtain precise results. Floating-point calculations are required to make the processing and computational operations quite fast. In this work its required to design an efficient MAC unit with all its processing units used in the proposed architecture. The FMA operation requires calculation in number of steps with diverse rounding operations along with normalization. Thus in this work its kept in view to reduce the number of steps of processing and to reduce the computational delays during the operations in order to increase the speed of processors.

***Keywords* --** CSA, FMA, Floating-point Arithmetic

I. INTRODUCTION

A floating-point unit synonymous to a co-processor is made to operate on floating-point numbers. Its one of the important applications involved in different architectures to provide the accuracy .In the present times, the floating-point unit of majority of processors have implemented multiply-add units so as to execute the FMA (fused multiply add) operation i.e; A+(B.C) with no intermediate rounding. A number of applications such as modern computers, DSP's, Graphics etc require FMA operations for efficient working. In the beginning IBM was first to launch FMA unit in 1990 with the name IBM(RS/6000) that was later modified and implemented by number of companies like Hp, Intel and many others. FMA is considered to be one the important features of FPU (floating-point unit).By using FMA instruction, performance and efficiency of processors is increased to a considerable extent. In FMA operation, the multiply and add operations are performed as a single instruction instead of separate add and multiply. In this execution i.e; A+(B C) during multiply A=0 and during add either B=1 or C=1. The operation can be taken as:

MULTIPLY: 0+ (B . C)

ADD: A+(1 . C) or A+(B . 1)

The efficiently designed FMA unit can provide the good results in terms of cost, area, power etc i.e; it can increase the overall performance of processors. In order to test these VLSI designs for better accuracy etc, FPGA (field programmable gate arrays) are usually used. Nowadays majority of processors have FMA units for executing FMA instruction. In FMA operation, first the floating-point multiplication is done and after that floating-point addition but as a single instruction. The instruction has been set as a standard by IEEE, which has specified it as IEEE-754-2008 so to perform the operation with only one significant rounding due to which an accurate and fast result is expected.

The IEEE-754 standard have set the floating-point formats as:

a. Binary Interchange format (with radix- 2)
b. Decimal Interchange format (with radix- 10)

The formats can be:

I. Single precision format (32 bit)
II. Double precision format (64 bit)
III. Quadruple precision format (128 bit)

Which can be shown as

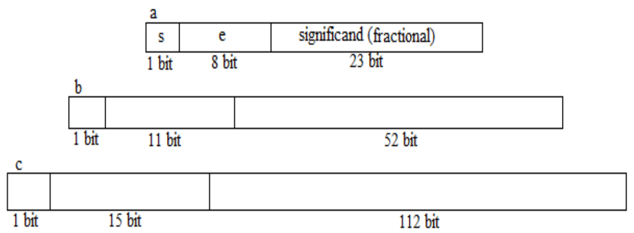

Fig. 1:- a - single precision format
b - Double Precision format
c - Quadruple Precision format

Generally the floating-point number is represented as:

N=M. R^ e

Where, M-mantissa, fraction or significand
R-radix (10 or 2)
e- exponent

The sign bit can be either '0' or '1'.

'0' for positive and '1' for negative

The FMA operation reduces the overall latencies by performing the combined multiply and add operations as a single instruction instead of using separate FADD (floating-point add) and FMUL (floating-point multiply) operations. FMA has ability to improve the arithmetic accuracy of algorithms. The use of FMA unit in general purpose processors have a number of benefits as:

First, the precision of MAF unit is quite better. Since it performs operation in single rounding it provides accurate results. Second, as MAF operation is done as a single instruction, the overall latency is improved and also there is less hardware requirement. Thus the performance of system becomes better. The basic architecture of FMA unit consists of multipliers, adders, shifters etc. The basic FMA architecture that needs to be studied for the design of an improved architecture is shown as:

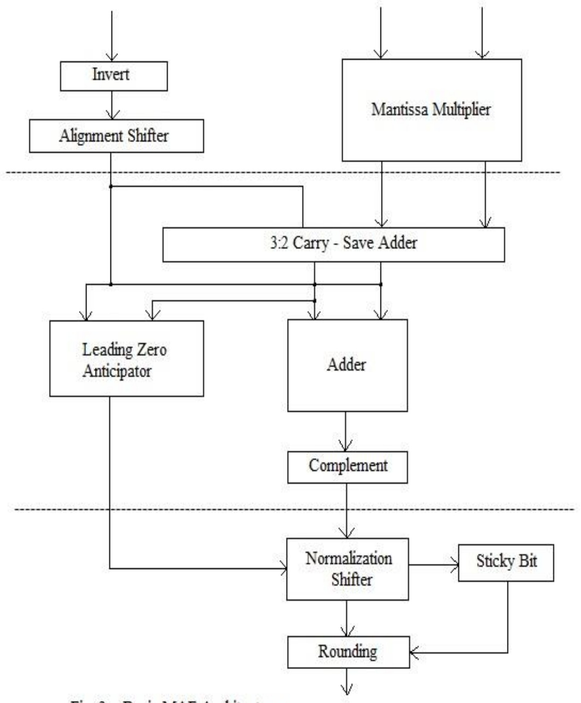

Fig. 2 :- Basic MAF Architecture

In our work its required to implement different multiplier and adder designs in comparison based on the study of number of previous architectures in such ways as to improve the accuracy and performance of the system. As it is always a requirement to present the improved versions of already implemented work, hence in this work its imperative to implement a design of such an FMA unit which will increase the speed of processing since its desirable to have such processors whose speed is very attractive feature. Here an effort is made to design a MAC unit whose processing speed will be very good to a considerable extent which can be achieved by reducing the computational delays etc.

II. RELATED WORK

A number of architectures for FMA unit have been proposed since 1990 in a view to improve the efficiency in terms of cost, area, power, latency etc. Different architectures are proposed in order to meet the demands of this changing modern world. A number of designs have been studied in the review work in order to get our required highly efficient FMA unit. A double-precision MAF unit [1] has been proposed that computes the addition having less latency compared to multiplication etc by skipping the first step of the pipeline stage. This approach when compared to others computes three operations with similar latencies. In this double data- path has been used so as to improve critical path. A multi-functional and a multiple-precision floating-point FMA architecture [2] is explained. This FMA is re-configurable and executes a QP instruction. Its also based on dual-path and it works efficiently for QP, single or double precision in order to improve latencies. The work has applied two fused floating-point operations [3] to FFT processors to perform the processing on complex valued data. The results of these fused operations are quite efficient as it requires less rounding. A dual-mode MAF design [4] is proposed which performs double-precision or two single-precision operations. It is implemented to achieve lower latency by introducing a number of new technique, an improvement on previous architectures. Here a high-precision QPFMA [5] is designed with 7 cycles of pipeline. A dual-adder, optimized LZA and shifter logic using normalization had led in the decrease of cost and latency. An MAF architecture [6] is implemented which is capable of processing de-normalized numbers. This work represents the On-Fly floating-point number processing but with little extra latency. The work [7] presents a high-speed floating-point double-precision architectures. The designs are implemented on Virtex-7 FPGA and is simulated using Verilog and synthesized in Xilinx software. An efficient MAC unit [8] in which area efficient CSA is implemented with parallel approach. The design reduces the delay by 27% but along with the increase in the number of LUT's used. This work shows a customized 40-bit SQRT CSA [9] architecture which decreases the power consumption in comparison to already existing one but there is some increase in delay. A multi-core CELL processor [10] is shown which supports 2-way and 4-way SIMD's. The approach is implemented with 6-cycle latency. The area, power etc are the key parameters with regard to efficiency of the work. It's the implementation of Divide-Add Fused [11] architecture. It shows an improvement in latency and cost of applications. The design is synthesized on Xilinx Spartan-6 FPGA in order to show better precision. All the above architectures are studied and compared for the reason to design an architecture which would present better results in terms of any one of the parameters like area, latency, delay, cost, power consumption etc.

III. APPROACH

In this paper work is done to design such a Mac unit for FMA operations. The operation can be sub-divided into number of different sub-blocks. Initially the significand alignment operation is executed which is implemented as a conditional swap. Multiply and Add units are implemented by using high-speed Adders and Multipliers. Redundant blocks are just eliminated. Then normalization operations are performed. The architecture is designed in order to increase the efficiency of the system by increasing the speed of processor as required for speedy, efficient and accurate processing and in the implementation of fast processors.

IV. CONCLUSION

In this paper, fused multiply-add operation is being studied upon. The FMA operation requires calculation in steps consisting of diverse rounding operations and normalization. In this the floating-point arithmetic IEEE standard 754 is to be implemented. Different adder and multiplier structures are studied which are available in literature and are being worked upon as to improve the efficiencies. Since floating-point operative units are optimized, the area or timing complexity must be improved. Also in the proposed technique which is to be worked on in this paper all the modules are to be designed on Xilinx software using hardware descriptive language like VHDL.

V. REFERENCES

[1]. Bruguera, Javier D., and Tomás Lang. "Floating-point fused multiplyadd: reduced latency for floating-point addition." In 17th IEEE Symposium on Computer Arithmetic (ARITH'05), pp. 42-51. IEEE, 2005.

[2]. K., D. Reisis, Manolopoulos, and V. A. Chouliaras. "An efficient multiple precision floating-point Multiply-Add Fused unit." Microelectronics Journal 49 (2016): 10-18

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

[4]. Manolopoulos, Konstantinos, D. Reisis, and Vassilios A. Chouliaras. "An efficient dual-mode floating-point Multiply-Add Fused Unit." In Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on, pp. 5-8.IEEE, 2010.

[5]. He jun and Zhu ying. "Design and Implementation of a quadruple Floating-point Fused multiply-add unit". Proceedings of 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

[6]. He, Hu, Zheng Li, and Yihe Sun. "Multiply-add fused float point unit with on-fly denormalized number processing." In 48th Midwest Symposium on Circuits and Systems, 2005., pp. 1466-1468. IEEE, 2005.

[7]. SravanthiKavitha, AddulaSaikumar. "An FPGA Based Double Precision Floating Point Arithmetic Unit using Verilog." In International Journal of Engineering Research and Technology, vol. 2, no. 10 (October-2013).ESRSA Publications, 2013.

[8]. Sachin Raghav and Dr Rinkesh Mittal. "Imlementation of Fast and Efficient MAC unit on FPGA". I.J Mathematical Sciences and Computing, 2016, 4, 24-33.

[9]. ParthaMitra, Datta, Debarshi, andAvisek Sen. "Low Power 40 bit SQRT Carry Select Adder." In International Journal of Engineering Research and Technology, vol. 1, no. 10 (December-2012).ESRSA Publications, 2012.

[10]. Mueller, Silvia M., Christian Jacobi, H-J. Oh, Kevin D. Tran, Scott R. Cottier, Brad W. Michael, Hiroo Nishikawa et al. "The vector floating-point unit in a synergistic processor element of a cell processor." In 17th IEEE Symposium on Computer Arithmetic (ARITH'05), pp. 59-67. IEEE, 2005.

[11]. Pande, Kuldeep, AbhinavParkhi, ShashantJaykar, and AtishPeshattiwar. "Design and Implementation of Floating Point Divide-Add Fused Architecture." In Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on, pp. 797-800. IEEE, 2015.