Floating Point Multiply Add Unit Design- A Review

1Sachin Raghav, 2Dr.Rinkesh Mittal
1Student, 2Professor
CEC Landran (Mohali), INDIA
Sachin.raghav219@gmail.com
Hod.coeece@cgc.edu.in

Abstract: **In the modern digital era floating point operations in digital systems forms an important part of research in recent years. Many researchers have proposed various floating point units and compare their results with the existing approaches. Implementing a floating point unit in digital systems requires the operands to be converted into the IEEE 754 single precision or double precision format. Also both the multiply and add operations must be implemented in unison in order to save the resources in FPGA. In this paper a survey of various implementation methodologies proposed by researchers and digital designers is presented and compared. The papers also focus on the end results of implementation of these units in terms of area utilized and speed of operation.**

Keywords: **FMA, Floating Point unit, FPGA.**

I. INTRODUCTION

In DSP applications the floating-point fused multiply-add instruction is currently available in many general-purpose processors [1]. It enhances its performance by decreasing the latency factor and increases the level of accuracy with no intermediate routing. Many of the commercial processors like IBM, Intel Itanium have involves the FMA unit in its FP units in order to execute double precision fused multiply-add operation. This FMA operation is significant when a FP multiplication is followed by FP addition [2]. The Fused Multiply-Add (FMA) operation has recently been added in the revised IEEE 754-2008 standard. The standard IEEE-754 double precision format consists of 64 bits which are divided into three sections.It has capability to enhance the arithmetic precision of algorithm as rounding steps has reduced [3]. FMA is also helps to improve the performance as it implements FMA instructions which have shorter latencies than floating point multiply and floating point add sequences. FMA implementation has two advantages. The first advantage is that the FMA operation is performed with only one rounding instead of two which helps to reduce overall error due to rounding and other advantage isthat it reduces delay and hardware required by sharing components [4]. In the year 1990, the first floating-point FMA unit was introduced on IBM RISC System/6000 which is used for single instruction execution operation as an indivisible operation. In basic FMA unit FADD and FMUL executions is not possible. In the Concordia FMA architecture is designed, which uses alignment blocks before the multiplier array. So multiplier tree input range widens. Due to this larger variable multiplier tree is required [5]. A few possible solutions have been identified in the Lang/Bruguera fused multiply-add architecture, which is designed for reduced latencies.Floating Point operation has been acknowledged to be useful for many real time graphic and multimedia application as well as DSP processors. Since it provide a wide dynamic range of Representable real numbers [6]. Most modern processor Perform Floating point Operation according to the IEEE 754-1985 standards. The following are the advantages of FMA:

1. Better accuracy because ofone rounding operation instead of two.
2. Increased performance is due to usage of a single instruction.
3. Usually a lower latency of the combined unit and having reduced area.

II. LITERATURE REVIEW

Han, Liu, HaoZhang et al. [1] in this paper, fused multiply add function has been presented. This proposed method is used to increase the speed of Decimal floating-point. Specific decimal redundant encoding system has applied to the fused multiply add architecture. This proposed approach is compared with other architecture and its results demonstrated that speed is increased about 33.7%.

Pande, Kuldeepet al. [2] presented combined operation of floating point division. The objective of this work is to enhance the performance and precision of applications where this operation was applied. The divide add fussed unit provides same architecture to floating point but divider makes some difference as it is implemented using digit-recurrence algorithms. This works indicates that the proposed architecture have better precision with respect to its unit such as divider unit and adder unit. This architecture is suitable for lower latency, accuracy

and its cost. The proposed unit is synthesized for4vfx60ff672-12 Xilinx Spartan-6 FPGA.

Kakde, Sandeep, Mithilesh Mahindra et al. [3] in this paper, Multiplier has been designed by using reduced complexity Wallace Multiplier to reduce the latency. In this paper, the total delay of multiplier designed using reduced complexity Wallace Multiplier is found to be 37.673 ns. Alignment Shifter and Normalization Shifter also been designed using barrel shifter to get higher accuracy and the latency is further reduced by 25–35 % and the total delay of Alignment Shifter and Normalization Shifter using Barrel Shifter is found to be 5.845 ns.

Manolopoulos, K., D. Reisiset al. [4] presented multi-functional, multiple precision floating-point Multiply-Add Fused (MAF) unit The proposed multiply add fused unit is capable to execute a quadruple precision MAF instruction, or two double precision instructions, or four single precision instructions in parallel.In this work,implementation of proposed design is done on a 65 nm silicon process achieving a maximum operating frequency of 293.5 MHz at 381 mW power.

He, Jun, Ying Zhu et al. [5] presented Design of a quadruple floating-point fused multiply-add unit. The proposed design supports multiple floating-point arithmetic with a 7 cycles pipeline. The implementation is done on 65nm technology and works at 1.2 GHz with reduces by 3 cycles. Also, the gate number decreased by 18.77% and the frequency increased about 11.63%.

Lupon, Marc, EnricGibert et al.[6] propose a novel hardware /software collaborative technique which is able to execute workloads with increased utilization of fused multiply adder. the host ISA of a hardware /software co-designed processor has been extended with a new Combined Multiply-Add (CMA) instruction that performs an FMA operation with an intermediate rounding.

Bruguera, Javier D et al. [7] In this work a design for double precision floating point multiply add fused unit has been presented . this computation based on MAF operation A + (B × C) which allows to compute FP addition with lower latency than FP multiplication and multi[ly-add fused. This novel architecture allows to skip pipeline stage which is first step.this first step is related with multiplication B*C. for a multiply add fused unit (MAF) unit pipelined into three or five stages, the latency of the floating—point addition is decreased to two or three cycles, respectively.

Atish Khobragade, et al. [8] In this paper 128-Bit Fused Multiply Add Unit has been proposed for crypto processor. this fused multiply add unit has been implemented on the FPGA(field programmable gate array).the major goal of this work is to decrease latency. the Multiplier is designed Using Wallace multiplier in order to reduce the latency and it is found that latency gets decreased up to 25%.to get higher precision, explicitly Alignment Shifter and Normalization Shifter has been designed in the FMA unit

Dhanabal, R. et al. [9] a modified architecture for Floating-Point Fused Multiply-Add (FMA) unit has been proposed in this paper. This architecture is usually for low power and reduced area applications. In FMA unit, floating-point $(A \times B) + C$ operation has been computed in single instruction. Also, bridge unit is being used. Bridge unit is a link between existing floating point multiplier and add round unit in the co-processor in order to perform FMA operation. The goal of this paper is to reuse the components and allows parallel FP addition in to add-round unit. Every unit is designed using verilog HDL. The design is simulated using Altera ModelSim and is synthesized using Cadence RTL compiler in 45 nm and FP arithmetic's are implemented in IEEE-754 double precision format.

Wu, Kun-Yi, et al. [10] presented a multiple-mode FP multiply-add fused unit. This proposed architecture utilizes the iterative multiplication and truncated addition techniques in order to support seven operating modes with various errors for low power applications. In this paper one multiply accumulate operation with three modes can executed. On comparison with IEEE754 single-precision FP MAF, the proposed unit has 23% longer delay and 4.5% less area.

III. CONCLUISON

In this work a review of various techniques proposed in recent times is presented. The operands are first converted into IEEE 754 format in order to perform multiply and add operations on them. Many researchers focus their implementation work on reducing the delay of the unit. The speed of the unit is increased to a maximum of 33.7% with a further reduction in latency up to 35%. While others focused on reducing the area required for the implementation and reduces the gate count to a significant value of approx. 18%, while increasing the frequency of operations. These values can further be reduces by selecting the performance parameter according to the application and then implement the unit with optimized units available in the literature.

REFERENCES

[1] Han, Liu, Hao Zhang, and Seok-Bum Ko. "Decimal floating-point fused multiply-add with redundant internal encodings." *IET Computers & Digital Techniques* (2015).

[2] 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.

[3] Kakde, Sandeep, Mithilesh Mahindra, AtishKhobragade, and Nikit Shah. "FPGA Implementation of 128-Bit Fused Multiply Add Unit for Crypto Processors." In *Security in Computing and Communications*, pp. 78-85. Springer International Publishing, 2015.

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

[5] He, Jun, and Ying Zhu. "Design and implementation of a quadruple floating-point fused multiply-add unit." In *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*.Atlantis Press, 2013.

[6] Lupon, Marc, EnricGibert, GrigoriosMagklis, Sridhar Samudrala, RaúlMartínez, KyriakosStavrou, and David R. Ditzel. "Speculative hardware/software co-designed floating-point multiply-add fusion." *ACM SIGPLAN Notices* 49, no. 4 (2014): 623-638.

[7] Bruguera, Javier D., and Tomás Lang. "Floating-point fused multiply-add: reduced latency for floating-point addition." In *Computer Arithmetic, 2005. ARITH-17 2005. 17th IEEE Symposium on*, pp. 42-51. IEEE, 2005

[8] Atish Khobragade, and Nikit Shah. "FPGA Implementation of 128-Bit Fused Multiply Add Unit for Crypto Processors." In *Security in Computing and Communications*, pp. 78-85. Springer International Publishing, 2015.

[9] Dhanabal, R., Sarat Kumar Sahoo, and V. Bharathi. "Implementation of Low Power and Area Efficient Floating-Point Fused Multiply-Add Unit." In*Proceedings of the International Conference on Soft Computing Systems*, pp. 329-342. Springer India, 2016.

[10] Wu, Kun-Yi, Chih-Yuan Liang, Kee-Khuan Yu, and Shiann-Rong Kuang. "Multiple-mode floating-point multiply-add fused unit for trading accuracy with power consumption." In *Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on*, pp. 429-435. IEEE, 2013.

