I.J. Mathematical Sciences and Computing, **2016, 4, 24-33**
Published Online November 2016 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijmsc.2016.04.03

Available online at ***http://www.mecs-press.net/ijmsc***

Implementation of Fast and Efficient Mac Unit on FPGA

Sachin Raghav [a], Dr. Rinkesh Mittal [a]

[a] *CEC Landran, Mohali/ECE Dept, Chandigarh, 140307, India*

Abstract

Floating-point arithmetic operations on digital systems have become an important aspect of research in recent times. Many architecture have been proposed and implemented by various researchers and their merits and demerits are compared. Floating point numbers are first converted into the IEEE 754 single or double precision format in order to be used in the digital systems. The arithmetic operations require various steps to be followed for the correct and accurate steps. In the proposed approach a fast and area efficient Carry Select Adder are implemented along with the parallel processing of various units used in the architecture. The result also verifies the proposed approach that shows a decrement of 27 % in the combinational path delay with an increment of around 8% in the number of LUTs used.

Index Terms: MAC Unit, FPGA, FMA.

1. Introduction

In DSP applications the floating-point FMA instruction is currently available in many general-purpose processors. It enhances its performance by decreasing the latency factor and increases the level of accuracy with no intermediate routing. Most of the processors like intel, IBM has involves the fused multiply fused unit in its FP units in order to carry out double precision FMA operation. This FMA operation is significant when an FP multiplication is followed by FP addition. It has the capability to enhance the arithmetic precision of algorithm as rounding steps have reduced. FMA is also helped 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 another advantage is that it reduces delay and hardware required by sharing components.

In the year 1990, the first floating-point FMA unit was introduced on IBM RISC System/6000 that is used for single instruction execution operation as an indivisible operation. In basic FMA unit FADD and FMUL executions is not possible. The development of FMA architecture utilizes the multiplier array followed by

* Corresponding author.
E-mail address: sachin.raghav219@gmail.com, hod.coeece@cgc.edu.in

alignment blocks. Thus, the range of input of the multiplier tree should be expanded because of the huge variable multiplier tree is necessary. Some explanation has been detected in Lang/Bruguera FMA architecture that is developed for minimizing the latencies. FP operation is utilized for some real time applications in multimedia and graphics and also in DSP processor. as it gives extensive range of real numbers. Many new processors execute FP operation according to IEEE 754-1985 standards.

2. Literature Review

Han, Liu, HaoZhanget 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, Kuldeep et al. [2] presented mutual procedure of floating point division. The objective of this work is to enhance the presentation and precision of application where this procedure was applied. The divide adds fussed unit provides the same architecture to the floating point but divider makes some difference as it is executed by utilizing digit-recurrence method. This works indicates that presented architecture has improved precision as compared to its units such as divider unit and adder unit. This architecture is appropriate for less latency, precision, and its price. The presented architecture is implemented using 4vfx60ff672-12 Xilinx Spartan-6 FPGA.

Kakde, Sandeep, Mithilesh Mahindra et al. [3] this paper developed a Multiplier by utilizing reduced complexity Wallace Multiplier to reduce the latency. In this work, the average delay of proposed approach is 37.673ns. Normalization and Alignment Shifter has also been designed using barrel shifter. In order to obtain the higher precision and lower latency, this shifter has been designed. The total delay is found to be 5.845 ns for this shifter.

Manolopoulos, K., D. Reiss et al. [4] presented multi-functional, multiple precision floating-point Multiply-Add Fused (MAF) unit. The proposed multiply-add fused unit is capable of performing a quadruple accuracy. In this work, implementation of proposed design is done on a 65 nm silicon process attaining highest operating frequency of 293.5 MHz at 381 mW power.

He, Jun, Ying Zhu et al. [5] presented Design of a quadruple floating-point FMA unit. The proposed design supports multiple floating-point arithmetics 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] presented a new hardware /software collaborative approach which is capable of performing workloads with improved utilization of fused multiply -adder. the host ISA of a hardware /software processor has been extended with a new CMA instruction that executes an FMA procedure with an intermediate rounding.

Bruguera, Javier D et al. [7] In this work a design for double precision floating point FMA unit has been presented. This computation based on FMA operation A + (B × C) which allows to calculating FP addition with less latency than FP multiplication and multi[ly-add fused. This novel architecture allows skipping 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.

AtishKhobragade,et al. [8] In this paper 128-Bit Fused Multiply Add Unit has been proposed for a crypto processor. this fused multiply add unit has been executed on the FPGA(field programmable gate array).the major goal of this work is to decrease latency. It is found that latency gets decreased up to 25%. In order to obtain the higher accuracy, normalization and alignment shifter has been developed in MAF unit.

Dhanabal, R. et al. [9] an improved design for Floating-Point FMA unit has been proposed in this paper. This architecture is usually used for less power and condensed area applications. The fused unit, floating-point $(A \times B) + C$ operation has been computed in a single instruction. Also, bridge unit is being used. Bridge unit is a link between accessible floating point multiplier and adds round unit in the co-processor. The goal of the

paper is to use again the components and allows parallel FP addition into the add-round unit. The proposed architecture is implemented by utilizing Altera ModelSim and is simulated using Cadence RTL compiler in 45 nm.

Wu, Kun-Yi, et al. [10] presented a multiple-mode FP multiply-add fused unit. This proposed architecture utilizes the multiplication and shortened addition approaches in order to maintain 7 modes of operation having several errors for low power applications. In this paper, one multiply accumulate procedure with three modes can execute. The analysis of result can be concluded that the proposed unit has 23% longer delay and 4.5% less area.

Huang, Libo, Li Shenet al. [11] in this work, a novel framework for MAF unit has been proposed. The presented MAF unit may execute the double precision and single precision which occupy about 18% hardware and also increase in 9% delay. The proposed MAF unit is being compared with the other conventional double precision MAF unit and it is found that the proposed unit is far better than conventional one in terms of area and delay, computational time. Various fundamental modules of double precision MAF unit should be redesigned in order to contain the simultaneous computation of two single-precision MAF operations. The proposed unit is completely pipelined. The result analysis indicates that it is appropriate for the processors having floating point unit (FPU).

Quinnell, Eric, Earl E. et al. [12] in this paper, two floating-point FMA unit for the single instruction execution of (A times B) + C has been proposed. In this paper, 3 path frameworks utilizes parallel path which is same as in dual path floating adders. The novel bridge framework utilizes floating-point elements to add a FMA instruction. Every new framework and collection of floating-point arithmetic units is designed using 65 nm cmos technology. The experimental result indicates that the proposed architecture is suitable for various processors.

Saleh, Hani, and Earl E. Swartzlander et al. [13] in this paper, floating-point fused add-subtract unit has been designed. The proposed design executes both operations i.e sum and difference. These two operations performed at same speed as a floating-point adder with less than 40% area overhead.

Samy, Rodina, Hossam AH Fahmyet al. [14] in this work, implementation of a parallel decimal floating-point FMA unit has been presented. In this paper, $\pm (A \times B) \pm C$ operation has been performed on decimal floating point operand. The proposed module is completely support the IEEE 754-2008 standard and carries the two standard formats that are decimal64 and decimal128. Moreover, presented design is used to execute the arithmetic operation like addition, subtraction, multiplication as separate operation.

Harpreet et al [15] in this paper, a novel single bit comparator is proposed which uses a reversible logic gate SKAR gate. The reversible approach is then compared with the existing logic circuits and obtain high fundamental cost and quantum cost.

Saleh, Hani, and Earl E. Swartzlander et al. [16] in this paper, floating-point fused add-subtract unit has been designed. The proposed design executes both operations i.e sum and difference. These two operations performed at same speed as a floating-point adder with less than 40% area overhead.

Samy, Rodina, Hossam AH Fahmyet al. [17] in this work, implementation of a parallel decimal floating-point FMA unit has been presented. In this paper, $\pm (A \times B) \pm C$ operation has been performed on decimal floating point operand. The proposed module is completely support the IEEE 754-2008 standard and carries the two standard formats that are decimal64 and decimal128. Moreover, presented design is used to execute the arithmetic operation like addition, subtraction, multiplication as separate operation.

Galal, Sameh, and Mark Horowitzet al. [18] proposed an approach for making trade-off curve which may be utilizes to approximate the floating point performance at some parameters. The proposed model has been designed in 90nm CMOS technology at 1 W/mm2 and may attain 27 GFlops/mm2 single precision, and 7.5 GFlops/mm double precision. In order to minimize the throughput of the system, some register file overhead should be added. A 1 W/mm2 design at 90 nm is a "high-energy" design, thus scaling is required to a lower energy design in 45 nm still produces a $7\times$ performance gain, while a more balanced 0.1 W/mm2 design only speeds up by $3.5\times$ when scaled to 45 nm. Scaling may reduces gradually when it is below 45nm and there some improvement of approx 3 times for for both power densities when scaling to a 22 nm technology.

Preiss, Jochen, Maarten Boersmaet al. [19] proposed a fine-grain clock gating approach which is used for fused multiply-add-type floating-point units (FPU). This clockgating is depending on the instruction type and value of operand. The proposed approach focus on minimizing the power where every floating point unit stages is used in each cycle. Based on the instruction mix, the approach allows 18% to 74% of the register bits.

Amaricai, Alexandru, MirceaVladutiuet al. [20] proposed combined operation of floating-point (FP) division followed by addition/subtraction-the divide-add fused (DAF). The main aim of this proposed method is to enhance the precision and performance of application where this combine designed is implementing. The presented DAF unit represented same architecture to FMA unit. The major difference can be signifies by divider that is applied by utilizing digit-recurrence algorithms. The significant module DAAF is signified by the essential quotient bits. The proposed method can be computed in terms of some parameters suc as cost, time, performance and accuracy. Therefore, two implementations have been presented: one pro-accuracy and one pro-performance. The result indicated that presented implementation shows better accuracy and it is appropriate for lower latency presents the best cost-performance tradeoff.

3. Proposed Technique

In the proposed design, a single precision MAC floating unit is designed. In the design the floating point number comprising of the sign, exponent and mantissa is first extracted and then the operations are performed in a parallel manner. The floating point unit operations are broadly divided into three main blocks. These blocks are:

Conditional Swap unit: In the first block the exponent, sign and mantissa value of the numbers are extracted and compared. The comparison is performed on the basis of exponent value and their difference is stored. For any operation to be performed on the floating point numbers, the mantissa must first be represented in the form of $1.xxxx$. So ‘1’ is concatenated in both the mantissa value on the MSB side. The mantissa of the higher exponent value is stored in the accumulator and the lower exponent value is shifted to the right using the shifters by the value of the difference. These mantissa values are then produced at the output of the block.

Arithmetic operations unit: In this block, the arithmetic operations such as the multiplication and the addition are performed on the shifted mantissa value. The operations are performed using the sharing process. The addition unit proposed in the architecture either used to perform the addition independently or used in the multiplication operation. The adder proposed is a fast and area efficient Carry Select Adder (CSLA). In the proposed addition unit, the carry is generated by taking the values as ‘0' and ‘1' of the input carry and finally selected using the multiplexer unit based on the actual carry input value. Figure 1-3 shows the architecture of the proposed approach. Figure 1 shows the proposed CSLA architecture. In figure 2 the half adder unit is shown and then in figure 3 the carry generation unit is shown. The carry generation unit is used to generate the carry on ‘1' and ‘0' independently. Finally, the carrying value is selected using the multiplexer and XORed with the sum value generated in the first unit to produce the final sum. The adder proposed in the design is shared with the multiplier unit which uses the addition unit for the multiplication operation.

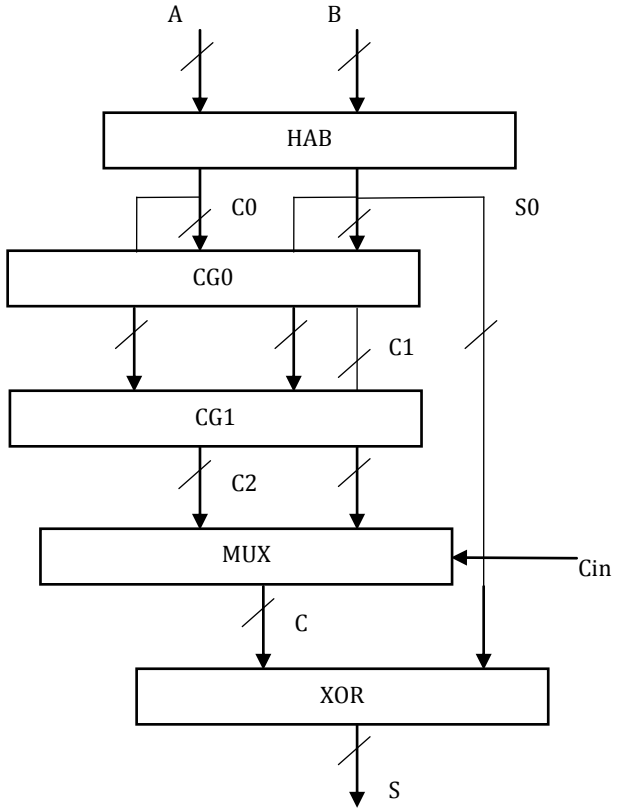

Fig.1. Proposed CSLA Architecture

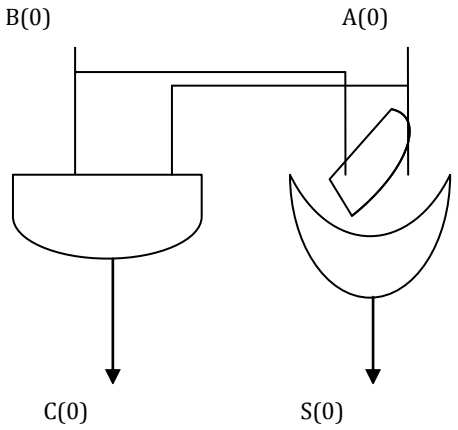

Fig.2. Half Adder Block

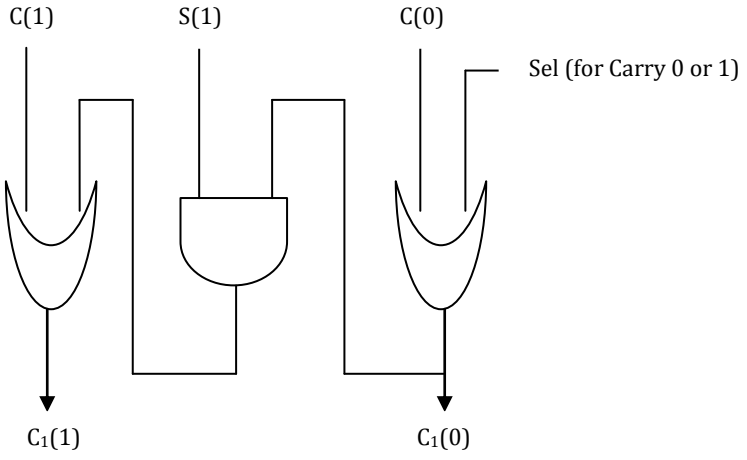

Fig.3. Carry Generation Unit

Post-Normalization Unit: In this unit, the output of the arithmetic unit is normalized and converted to the IEEE 754 format. The mantissa from the second block is again converted into the form $1.xxxx$ and then the exponent value from the first block is shifted according to the leading zero value. The three values i.e. sign shifted exponent and the mantissa is concatenated to produce the final result.

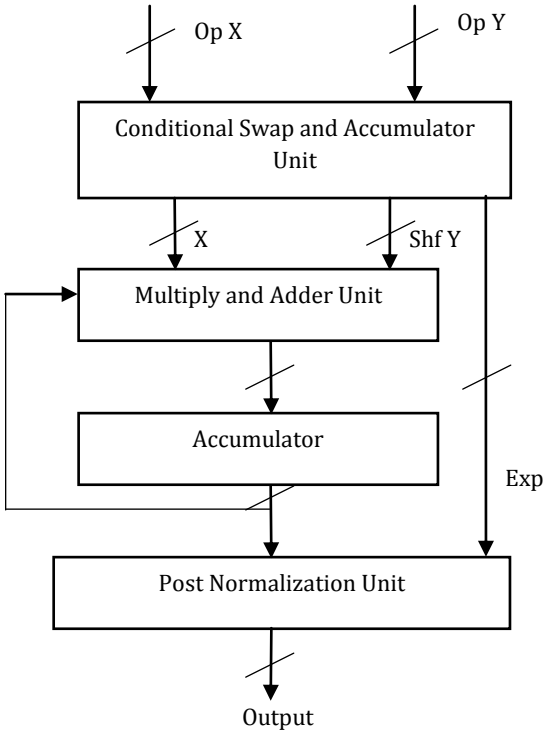

Fig.4. MAC Unit

Figure 4 shows the MAC unit architecture with the use of Accumulator in the proposed methodology.

4. Results and Discussions

The proposed methodology is implemented using the Xilinx Spartan 3 FPGA and VHDL. The basic approach is compared with the proposed methodology on the basis of various performance parameters. The simulation waveform of the top level module is shown in figure 5. Table 1 describes the comparison of the basic and proposed approach on the basis of combinational path delay and area utilized by the device. It is clear from the table shown above that the delay in the proposed parallel approach is reduced while slight increment in the number of slices and the LUTs. This implies that the combinational path delay in the circuit i.e. critical path delay is reduced or the complexity of the path or time is reduced.

Fig.5. Simulation of Top Module

Table 1. Comparison Table

Parameters	Basic Approach	Proposed Approach
Number of Slices	175	190
Number of 4 input LUTs	319	345
Delay (ns)	56.084	40.959
Max. Frequency (MHz)	17.80	24.41

It is clear from the table shown above that the delay in the proposed parallel approach is reduced while slight increment in the number of slices and the LUTs. This implies that the combinational path delay in the circuit i.e. critical path delay is reduced or the complexity of the path or time is reduced.

Figure 6 and 7 shows the comparison graphs between the basic approach and the proposed approach. It is evident from the graphs that the proposed techniques utilize slightly more resources as compared to the basic approach but there is a significant decrease in combinational path delay and increment in maximum operating frequency.

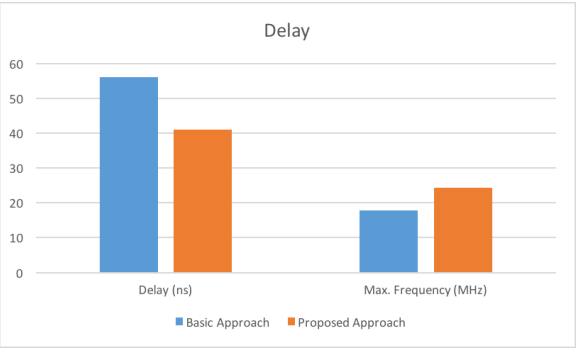

Fig.6. Delay

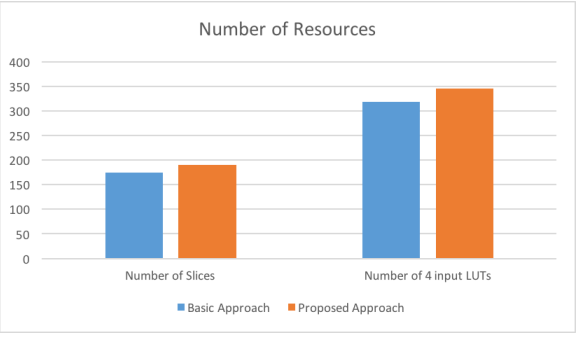

Fig.7. Number of Resources Utilized

5. Conclusion

Floating point operations in the digital circuits are the most important processing blocks in various digital and image processing applications. In this paper, a parallel approach is proposed in which the architecture is divided into various blocks which perform the operations independently. A resource sharing approach is also proposed which further reduces the area complexity of the proposed design. In this approach, the multiplication operations share the resources of the addition unit to decrease the area used by the architecture. Results also show that with the slight increase in the area of the design the delay is reduced by around 27 %. In future the addition operation used in the design must utilize fewer resources and the timing delay of the unit must be further improved by using the pipelining and improved addition- multiplication approach.

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, Raúl Martínez, 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] 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.

[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-RongKuang. "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.

[11] Huang, Libo, Li Shen, Kui Dai, and Zhiying Wang. "A new architecture for multiple-precision floating-point multiply-add fused unit design." In 18th IEEE Symposium on Computer Arithmetic (ARITH'07), pp. 69-76. IEEE, 2007.

[12] Quinnell, Eric, Earl E. Swartzlander, and Carl Lemonds. "Floating-point fused multiply-add architectures." In 2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, pp. 331-337. IEEE, 2007.

[13] Saleh, Hani, and Earl E. Swartzlander. "A floating-point fused add-subtract unit." In 2008 51st Midwest Symposium on Circuits and Systems, pp. 519-522. IEEE, 2008.

[14] Samy, Rodina, Hossam AH Fahmy, RamyRaafat, Amira Mohamed, Tarek ElDeeb, and Yasmin Farouk. "A decimal floating-point fused-multiply-add unit." In 2010 53rd IEEE International Midwest Symposium on Circuits and Systems, pp. 529-532. IEEE, 2010.

[15] Singh, Harpreet, and Chakshu Goel. "Design Approaches for a Novel Reversible 4-bit Comparator." IJMSC-International Journal of Mathematical Sciences and Computing (IJMSC) 1.3 (2015):20.

[16] Quinnell, Eric, Earl E. Swartzlander Jr, and Carl Lemonds. "Bridge floating-point fused multiply-add design." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 16, no. 12 (2008): 1727-1731.

[17] Saleh, Hani, and Earl E. Swartzlander. "A floating-point fused add-subtract unit." In 2008 51st Midwest Symposium on Circuits and Systems, pp. 519-522. IEEE, 2008.

[18] Samy, Rodina, Hossam AH Fahmy, RamyRaafat, Amira Mohamed, Tarek ElDeeb, and Yasmin Farouk. "A decimal floating-point fused-multiply-add unit." In 2010 53rd IEEE International Midwest Symposium on Circuits and Systems, pp. 529-532. IEEE, 2010.

[19] Galal, Sameh, and Mark Horowitz. "Energy-efficient floating-point unit design." IEEE Transactions on Computers 60, no. 7 (2011): 913-922.

[20] Preiss, Jochen, Maarten Boersma, and Silvia Melitta Mueller. "Advanced clockgating schemes for

fused-multiply-add-type floating-point units." InComputer Arithmetic, 2009. ARITH 2009. 19th IEEE Symposium on, pp. 48-56. IEEE, 2009.

Author's Profiles

Sachin Raghav is a Post Graduate student of M.Tech VLSI at Chandigarh Engineering College Landran, Mohali, India. He completed his B.Tech degree in Electronics and Communication from ICL institute of Engineering and Technology, Ambala, India. His area of interest includes VLSI, Digital and Wireless communication.

Dr. Rinkesh Mittal is working in ECE department in CEC, Landran, Mohali, Punjab. He has 12 Years of rich teaching research experience .His research areas include Antenna, Wireless, VLSI, and Embedded related fields. He has guided many research students.

How to cite this paper: Sachin Raghav, Rinkesh Mittal,"Implementation of Fast and Efficient Mac Unit on FPGA", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.2, No.4, pp.24-33, 2016.DOI: 10.5815/ijmsc.2016.04.03