A Low Power Design Of Floating Point Multiply Add Unit

Insha Ishteyaq, Kantesh Kumar Guarav, Heena Gupta
Panchkulla Engineering College (PEC)
Panchkulla India

***Abstract*—Signal Processing and Image Processing applications require floating point operations in digital circuit design. In order to increase the accuracy of the results the floating point operations are preferred in almost all digital design applications. Various arithmetic operations use floating point calculation and can be used for implementation of various computational and logic unit operations. In the proposed work a fused multiply- addition unit is proposed which utilizes the common addition block for both addition and multiplication operations. The floating point number is first converted into the IEEE 754 format and then the calculation for both addition and multiplication is performed. The significand is extracted from the number and the calculations are performed on the basis of the exponent difference between the numbers. In the proposed approach a parallel architecture is designed which first extracts the significand and exponent value in the first unit and multiplication-addition operations on the second block. The final output is carried out on the third block where normalization and zero detector operations are performed. The proposed approach is then compared with the basic approach and shows improvement in power consumption and maximum combinational path delay. Results shows that the delay is decreased by approx. 17% and power is decreased by approx. 77%.**

***Keywords*- FPU, FMA, Floating Point Arithmetic**

I. INTRODUCTION

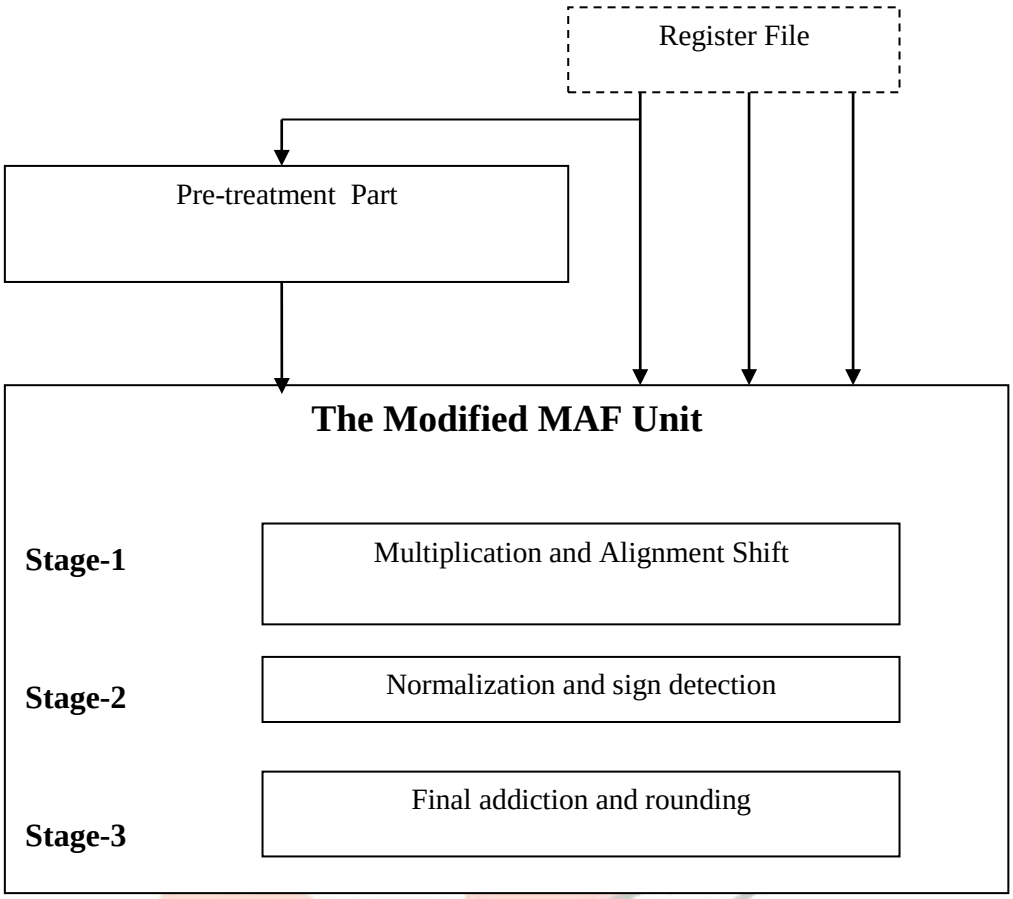
Floating-point number is a fractional number which is formed by dividing one integer by another one. A computer can be seen as an integer machine which can process and realize real numbers only by taking them as complex codes. Thus from many years, floating-point representation is taken into account. Since it can pursue a large range of numbers in between, thus only because of it has gained popularity for code representation of real numbers and it has been called as IEEE-754-2008 floating-point standard.

Floating-point numbers are processed upon by using a co-processor called floating-point unit (FPU). Its an important application used to provide accuracy to most of hardware designs. In this recent era, floating-point multiply-add units are being implemented in the floating-point units so to perform fused multiply-add (FMA) operations. In this very proposed, a fused multiply-add unit is designed for smaller latencies in comparison to all other traditional units. The designed units are analyzed for number of parameters mainly for area, power-consumption etc. the performance of adder and multiplier units was tested by applying various optimization techniques and as a result it was observed that this proposed system have minimized the area, reduced the delay and improved power-consumption. In FPGAs (Field Programmable Gate Arrays) the number of arithmetic operations are done by implementing floating-point calculations. Fused multiply-add operation requires calculation in steps consisting of diverse rounding operations along with normalization. All these operations ought to generate high-computational delays almost at each step of calculation.

In order to implement the floating-point multiplication and addition units, the different units for multiplication and addition are concatenated and along with this other redundant blocks are removed. So far as the multiplication and addition units are optimized the area and timing constraints of the particular units might be improved. Floating-point unit is one of the most important and essential logic design unit offering a vast field in the research needed for large number of computations. Some of the areas which could be worked on include large precision, complexity of design, big cancel management and complex rounding processes. In order to overcome the disadvantages and drawbacks of the previous designs, a number of techniques regarding optimization needs to be worked on for the future scopes. This work is focused on the design of a floating-point fused multiply-add unit architecture for proficient and quite comparable performance of processors that are to be relied on IEEE-754-2008 standard. The design is made to use decimal floating point representation of numbers.

The key feature or an essential component for performing floating-point arithmetic is floating point unit (FPU).

The architecture of floating-point unit can be shown as:



It increases the performance of floating –point operations along with accuracy as such because rounding is to be performed only once for the whole result instead of two times i.e; separately for multiplier unit and then for adder. The floating-point fused multiply-add operations (FMA) reduces the overall latencies by performing the combined multiply-add operations as a single-instruction instead of using separate multiplier and addition operations. In these times, the floating-point units of majority of commercial processors like IBM power PC, Intel/ Hp Itanium, MIPS-compatible Loongson-2F have included an FMA (floating-point fused multiply-add) units in order to execute the double-precision fused multiply-add operation as an un-divisible operation as such with no intermediate rounding.

II. GENERAL PROPOSED UNIT

In this very proposed approach we are required to implement certain techniques for fused multiply – add unit. In our proposed architecture the adder part of addition unit is common for multiplication as well. Its implemented in the architecture in order to reduce the total area and to improve latency by reducing maximum combinational path delay. The proposed unit also utilizes the leading zero analyzer circuit. A leading zero analyzer (LZA) is a simple logic circuit that calculates the number of zero's in view of which the output can be converted back to IEEE – 754 standard format. The resulting operation is called post normalization. This unit is taken as a parallel processing unit.

The units proposed in the technique include significand alignment unit where the comparison of exponents and the alignment of significands take place. Then carry–select adder unit and post normalization units are implemented with certain modifications.

In significand alignment unit, its required to extract the exponents, significands and sign bit from the given input operands. Then, the comparison is done between the exponents. The significant of the larger exponent is shifted to right in accordance with the difference of two numbers. Thus the exponents are first compared and then on the basis of comparison the greater significand is shifted accordingly. However, if there will be no difference in the exponents then the significand should remain same i.e. there will be no shifting.

Let us take 'X' and 'Y' as two input floating point numbers that were initially converted to IEEE – 754 Standard format. So its required to extract all parts of floating point numbers and then compare their exponents. Now let us suppose the exponent of input 'Y' is greater than input 'X' then its required to shift the significand of 'X' to the right by an amount equal to difference of exponent value. Then again the significands of the inputs are translated to IEEE format that is 1.xxx--- form. Thus it means that '1' is concatenated with significand value before the 'X' significand is shifted.

After that the addition / multiplication unit is implemented, which is to be designed by using a carry – select adder architecture. Here in our proposed work carry-select adder is implemented by using carry look-ahead adders by which maximum combinational path delay is reduced and area is improved.

Finally normalization is done by using post normalization unit which is implemented by using LZAs (leading zero analyzer).which is a simple logic circuit for calculating number of zeroes in order to convert final output back to IEEE standard.

III. RESULTS

The proposed methodology is being implemented using the Xilinx Zynq FPGA using hardware descriptive language VHDL . The RTL view of the proposed methodology is shown in figure 1. The figure contains three major blocks- first is Significand alignment unit, second is the carry select unit and the third is post normalization unit.

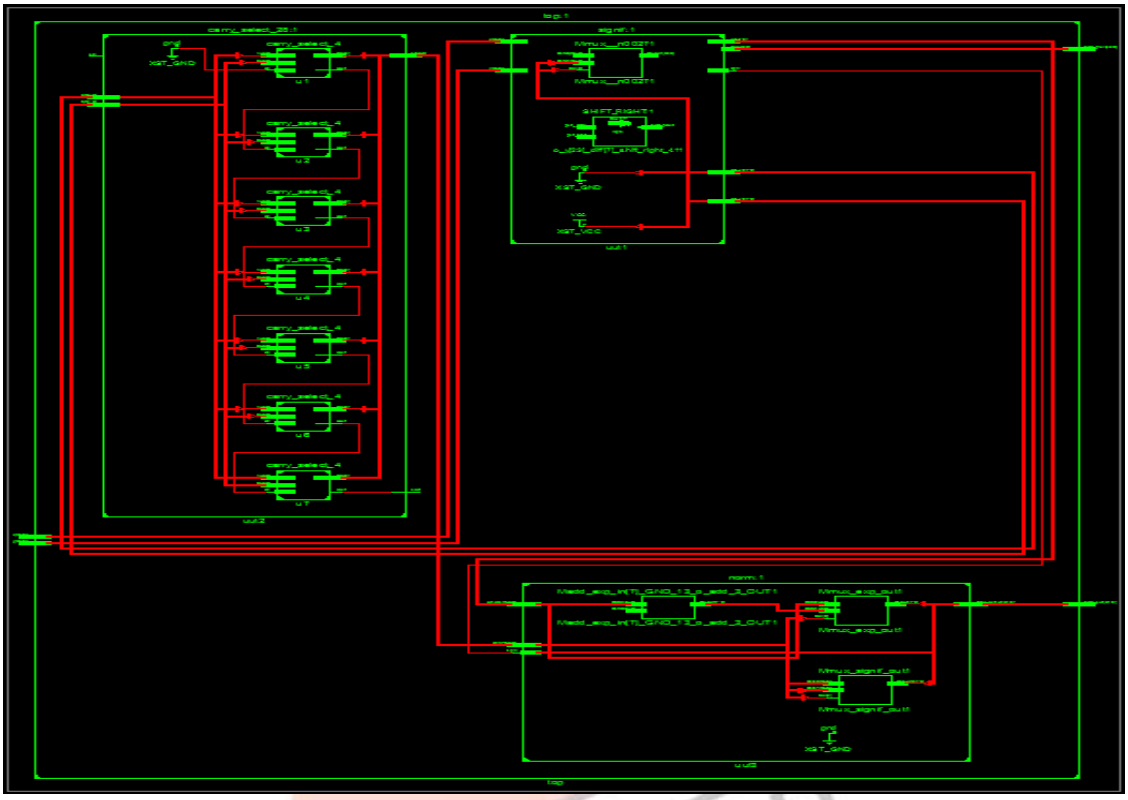


Fig 1: RTL Diagram of Proposed Methodology

Simulation of the current circuit is performed using the Xilinx I-Sim simulator. The simulation waveform is shown in figure 2.

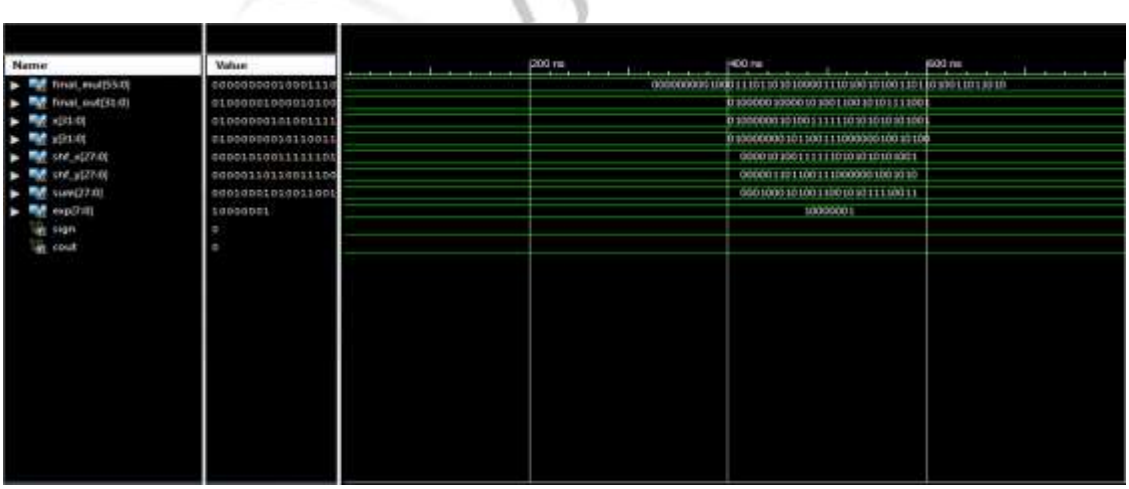


Fig 2: Simulation Waveform of the Proposed Methodology

Delay described here is maximum combinational path delay. The critical path delay is calculated from input to output and is defined as the maximum delay in the circuit. Table 1 given here compares the maximum delay of the proposed approach with the basic approach given in [11].

Table 1:Delay And Area Utilization

Parameters	Basic approach [11]	Proposed approach
Delay (ns)	7.499	6.158
Area (number of slice LUTs)	-	43

The maximum power utilized by the design is calculated using the Xilinx Power Estimator. Table 2 shows the maximum power utilized by the design and comparison to the basic approach.

Table 2: Power Consumption

Parameters	Basic approach [11]	Proposed approach
Power (mW)	3.82	.86

IV. CONCLUSION

Floating point arithmetic operations form an important part in various digital signal processing applications. Fused operations in floating point unit is an important area of research in recent times and various researchers have proposed architectures to improve the efficiency and the accuracy of the arithmetic operations. In the present work a fused architecture is proposed to improve the timing delay in the design. The architecture utilizes the adder resources and used them for the multiplication. The architecture is also compared with the available architecture and compared on the basis of maximum combinational path delay and power consumption. The proposed architecture performs better as compared to the basic approach in terms of delay which is reduced by approx. 17% and power consumption is reduced by 77%. In future the timing delay of the design can be further optimized by using the optimized carry-select adder architectures.

REFERENCES

[1] Wahba, Ahmed A., and Hossam AH Fahmy. "Area Efficient and Fast Combined Binary/Decimal Floating Point Fused Multiply Add Unit." *IEEE Transactions on Computers* 66, no. 2 (2017): 226-239.

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

[3] Montoye, R.K.; Hokenek, E.; Runyon, S.L., "Design of the IBM RISC System/6000 floating-point execution unit," in IBM Journal of Research and Development , Volume-34, Issue-1, PP 59-70.

[4] 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.

[5] Li, Gongqiong, and Zhaolin Li. "Design of A Fully Pipelined Single-Precision Multiply-Add-Fused Unit." In *20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07)*, pp. 318-323. IEEE, 2007.

[6] Amaricai, Alexandru, MirceaVladutiu, and OanaBoncalo. "Design of floating point units for interval arithmetic." In *Research in Microelectronics and Electronics, 2009. PRIME 2009. Ph. D.*, pp. 12-15. IEEE, 2009.

[7] Fahmy, Hossam AH, RamyRaafat, Amira M. Abdel-Majeed, RodinaSamy, Tarek ElDeeb, and Yasmin Farouk. "Energy and delay improvement via decimal floating point units." In *Computer Arithmetic, 2009. ARITH 2009. 19th IEEE Symposium on*, pp. 221-224. IEEE, 2009.

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

[9] Qi, Zichu, Qi Guo, Ge Zhang, Xiangku Li, and Weiwu Hu. "Design of low-cost high-performance floating-point fused multiply-add with reduced power." In *2010 23rd International Conference on VLSI Design*, pp. 206-211. IEEE, 2010

[10] Kim, Donghyun, and Lee-Sup Kim. "A floating-point unit for 4D vector inner product with reduced latency." *IEEE Transactions on computers* 58, no. 7 (2009): 890-901.

[11] 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.*