Analyze and Implementation of FPGA Implementation of HUB Floating-Point Addition

T. Srinivas Reddy, CH. Shekar, J. Prabhakar

Abstract: ***FPGA is progressively existence used to design high performance and computationally intense processors proficient of management in cooperation fixed and floating point mathematical operations. The prominent necessity of Half-Unit-Biased due to its flexibility with the shifting operations of numbers in half unit. This paper analysis the profits of by means of pipeline format to implement floating point (FP) arithmetic below a round to nearest mode from a measureable point of interpretation. With the pipelining format to represent numbers permits the removal of rounding logic of arithmetic units, including sticky bit computation.***

Index Terms: ***HUB format, Floating Point (FP), FPGA, Pipelining***

I. INTRODUCTION

The use of the addition of a floating point today will be the most prevalent floating point operation and more or less half the scientific operation. Floating point arithmetic firmware implementations have been key components of a variety of domain-specific hardware platforms due to the constraints positioned by new applications on those platforms [1][2]. There have been recent mistakes in implementing FPUs by major world suppliers [4][5]. Therefore it is a major component of mathematical coprocessors, DSP processors, embedded arithmetic processors and data processing units. However these elements require advanced economic stability and precision in numerical terms and are often solely based on a floating point. Adding a floating point is an expensive hardware-and timing operation mostly because different types of blocks with variable latency are required. Latency is the overall bottleneck in performance for floating point additions. The overall bandwidth utilization of floating point preservatives was greatly improved.

Along with various waypoints these media groups basically allow RN to be performed by truncation. While on the other hand actually, these new proposals depend entirely on the important changes of the regular arrangements, which can connect any ordinary configuration for all purposes and scientific purposes. In this paper, we focus on the HUB which uses PIPELINING FP groups. The efficiency of using the HUB groups for a settled point representation has been demonstrated in and. By diminishing bit-width even while maintaining the same accuracy, the zone costs and the shift in the use of restricted drive reaction channels were significantly reduced. In this paper, we estimate the benefit acquired by PIPELINING organizations to carry out RN FP calculation frameworks. Some initial results for half-precision FP adders and multipliers have been shown.

II. BACKGROUND WORK

More per logic, we can still say the HUB refers to the ability of a movement in last place of the standard numbers (ULP). Only a part of its essential shows is that the 2's complement is performed by bit-by-bit reversal, round-to-close truncation, and that a similar accuracy requires an indiscriminately large number of bits for capacities from its ordinary partner [9].

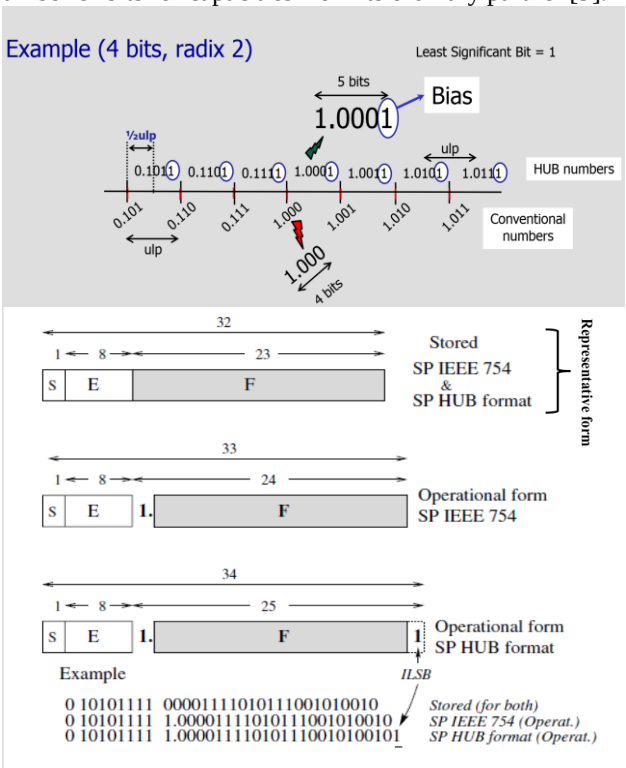

Fig.1: Basic example

In [6] the authors advocated the FP add-subtracting unit as useful for DSP applications like fast Fourier transforms and discrete cosine transforms. As mentioned previously the usage of different pipeline stages and algorithms for floating point operations that are essential for energy efficient implementation.

Revised Manuscript Received on December 22, 2018.

Dr. T. Srinivas Reddy, Associate Professor, Dept. of ECE, Malla Reddy Engineering College (A), Main Campus, Hyderabad, India,.

CH. Shekar, Assistant Professor, Dept. of ECE, Teegala Krishna Reddy Engineering College, Hyderabad, India

J. Prabhakar, Assistant Professor, Dept. of ECE, Nalla Malla Reddy Engineering College, Hyderabad, India.

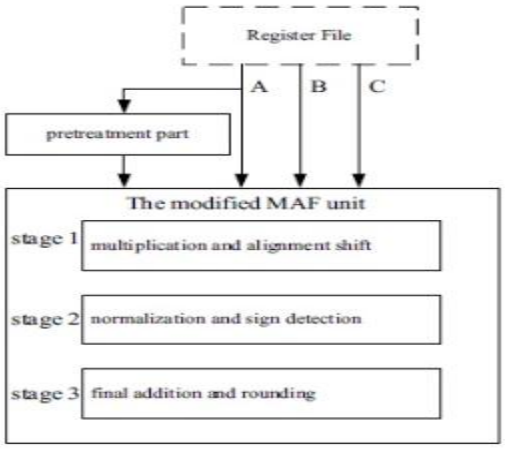

Fig. 2: flow process of Floating Point Unit structure

In [8], the authors portrayed the development and deployment of the user-characterized, fused FP arithmetic procedures used for Radix 2 FFT in DSP processors for complex numbers. The primary requirement for any DSP chips is to need of hardware that supports floating point operations as related to normal fixed point operations. The butterfly consists of a complex procedure for multiplying, adding and subtracting the same pair of data for radix-2 implementation.

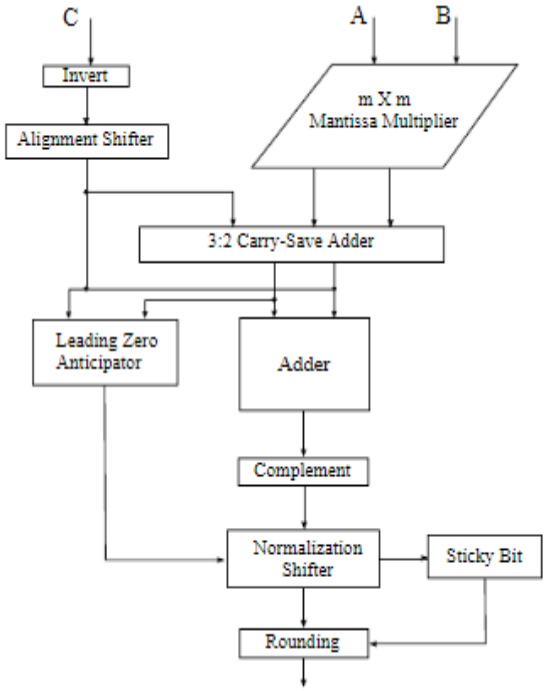

Fig.3: Flowchart for Fused Floating Point Add Subtract and Multiply-Add Module

In [10], the authors presented the FP Division's combined operations and the all kinds of arithmetic formulation so that its overall performance can be improved a lot in terms of accuracy. In [11] the authors portrayed the Add Subtract fused FP unit by using the most IEEE operations number system that supports parallelism for doing addition as well subtraction in single cycle, which reduces the firmware and costs of the designed unit

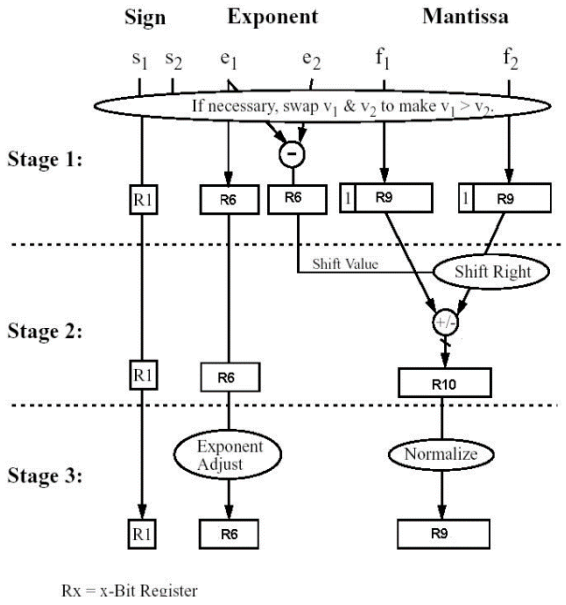

Fig.4: flow process of floating point unit for arithmetic operations

III. METHODS & METHODLOGY

A generalized modified floating point design is shown in [12] Fig.5. Compared to the traditional FP architecture predecessor numbers, perhaps the HUB architecture hasn't used the circuits necessary to approximate the sticky bit is merely truncated.

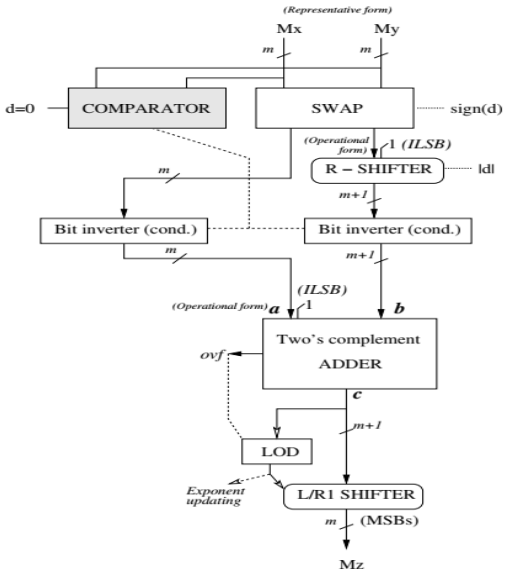

Fig. 5. Modified Floating-point adder structure [12]

It purpose of doing two numbers floating point operations is to ensure that significant of rounding to nearest value for precision to obtain the reliability in the same direction of operations.

FP Multiplication for HUB Numbers using pipelining register:

According to our discussion the unit works clearly as follows:

The brand new exponent is determined by adding the input operand exponents for the first step of the pipeline multiplier, while the meanings are multiplied and even the exponents are chosen to add value round, thus acquiring a value that is below the size. The multiplication result is also normalized in the second phase of the pipeline concept by moving it 1 bit to the right, if necessary.

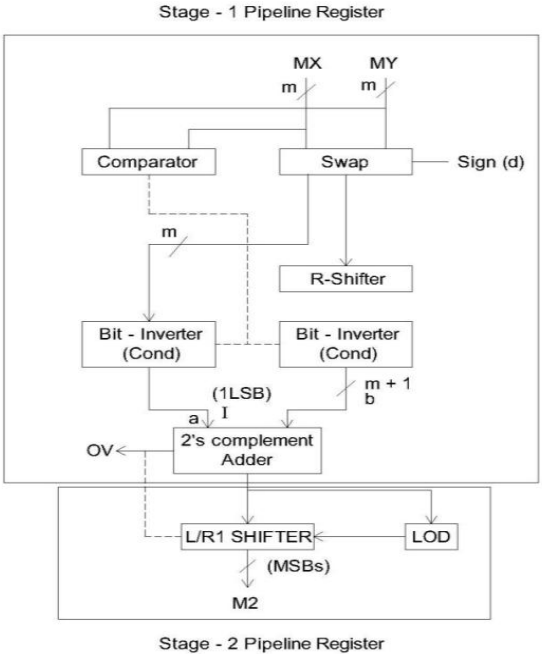

Fig.6: FP Adder using pipelining register

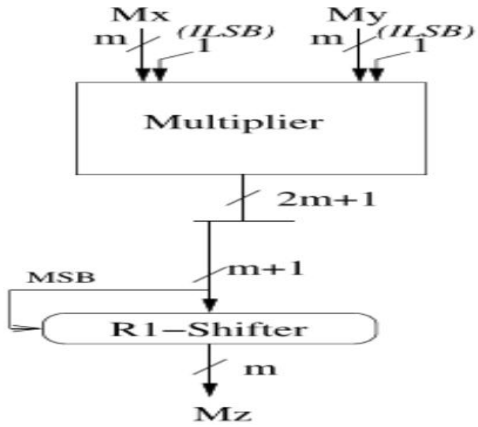

Fig.7: Basic FP multiplier architecture

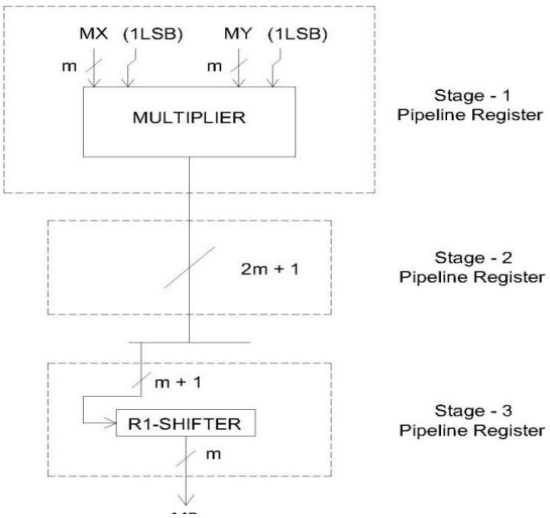

Fig.8. FP multiplier using pipelining register

IV. RESULTS & DISCUSSION

From the simulation, it is clear that for the comfort and accessibility of someone using the suggested HUB FP designs using the pipelines to calculate the real number. Paramount, we perform an investigational error scrutiny to prevent accuracy of the compute as a result of the necessity of PIPELINING formats. In the latter phase, we analyze important hardware implementing results for suggested HUB FP circuits related to the traditional implementation.

Fig.9: Floating point adder without pipelining unit

Fig.10: Floating point adder with 2 stage pipelining unit

Fig.11: Floating point multiplier with 3 stage pipelining unit

V. CONCLUSION

This work audit shows different FP strategies. So many other conventional FP unit designs have been introduced and conveyed. This paper also examines the different sources explicitly for biasing in doing rounding so that it can perform addition of floating point and fused multiplication.

The entire design was recaptured in Verilog Hardware Description (HDL), simulationally tested by Model Tech modelling, placed and routed on the Xilinx Vertex 5 FPGA.

REFERENCES

1. G. Tagliavini, S. Mach, D. Rossi, A. Marongiu, and L. Benini, "A transprecision floating-point platform for ultra-low power computing," in 2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018, 2018, pp. 1051– 1056.
2. M. Langhammer and B. Pasca, "High-performance QR decomposition for FPGAs," in Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ser. FPGA '18. New York, NY, USA: ACM, 2018, pp. 183–188.
3. X. Fang and M. Leeser, "Open-source variable-precision floating-point library for major commercial FPGAs," TRETS, vol. 9, no. 3, pp. 20:1–20:17, 2016. [Online].
4. W. E. Wong, X. Li, and P. A. Laplante, "Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures," Journal of Systems and Software, vol. 133, pp. 68 – 94, 2017.
5. R. Murphy, T. Sterling, and C. Dekate, "Advanced architectures and execution models to support green computing," Computing in Science and Engineering, vol. 12, no. 6, pp. 38–47, Nov 2010.
6. Pooja Potdar, S.S. Tamboli, "Improved Architectures for Fused Floating Point Add Subtract Unit," International Journal of Science and Research ISSN: 2319-7064, 2013.
7. J. D. Bruguera and T. Lang, "Leading-One Prediction with Concurrent Position Correction," IEEE Transactions on Computers, pp. 1083–1097, 1999, vol. 48, no.10.
8. Prasanna Palsodkar, "Improved Fused Floating Point Add-Subtract and Multiply-Add Unit for FFT Implementation," International Conference on Circuits and Systems, 2014. J. Hormigo and J. Villalba, "New formats for computing with real numbers under round-to-nearest," IEEE Transactions on Computers, vol. 65, no. 7, pp. 2158–2168, 2016. Abhinav Parkhi, Shashant Jaykar, Kuldeep Pande, "Design and Implementation of Floating Point Divide-Add fused Architecture," 5th International conference on communication Systems and Network technologies, 2015.
9. Jyoti Sharma, Pabbisetty Tarun, Sambangi Satish kumar and Sivanantham, "Fused Floating-point Add and Subtract Unit" Online International Conference on Green Engineering and Technologies (IC-GET) 2015.
10. J. Hormigo and J. Villalba, "Measuring Improvement When Using HUB Formats to Implement Floating-Point Systems under Round-to-Nearest," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6, pp. 2369–2377, 2016.