JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Implementation and Performance Improvement of 64 bit Posit Multiplier for High Speed VLSI-FPGA Processor

[1]Nisha Khatri[1], [2]Dr. Soheb Munir

[1]Research Scholar, [2]Associate Professor

[1&2]Department of Electronics and Communication Engineering,

[1&2]Lakshmi Narain College of Technology, Bhopal, India

Abstract : A digital multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. A variety of computer arithmetic techniques can be used to implement a digital multiplier. This continuous improvement of the state of the art has been accompanied by an increase in computational complexity and an overhead in hardware resources. Posit number system has been used as an alternative to IEEE floating-point number system in many applications. The latency and speed is important parameter in digital multiplier. This paper proposed 64-bit posit multiplier for high speed VLSI-FPGA processor. Xilinx 14.7 is used to implementation with verilog programming language.

IndexTerms **- Posit, Floating point, Xilinx 14.7, Multiplier, Verilog, Speed.**

I. INTRODUCTION

The Energy minimization is one of the fundamental plan prerequisites in practically any electronic frameworks, particularly the versatile ones, for example, advanced mobile phones, tablets, and unique devices. It is profoundly wanted to accomplish this minimization with insignificant execution (speed) punishment. Advanced signal handling (DSP) squares are key parts of these compact gadgets for acknowledging different sight and sound applications. The computational center of these squares is the number-crunching rationale unit where increases have the best offer among all number juggling tasks performed in these DSP frameworks. Thusly, improving the speed and power/vitality proficiency attributes of multipliers assumes a key job in improving the productivity of processors.

In FIR channel structured, will utilized plan any multipliers, if last continuous years, the multiple constant multiplication (MCM) system will utilized, as a FIR channel plan, yet the disadvantage is MCM strategy won't work both thing of marked and un-marked activity, so it will we have to configuration separate MCM for marked and unsigned augmentation. So here, examined a MCM with Adjusted based surmised multiplier that incorporates both marked and unsigned activity in single multiplier, this multiplier will executed in FIR Channel, and demonstrated the productivity of region, power and delay.

Finite Impulse response (FIR) computerized channel is broadly utilized in a few advanced signal preparing application, for example, discourse handling, uproarious speaker balance, reverberation retraction, versatile clamor wiping out, and different correspondence application, including programming characterize radio etc. A significant number of this application require FIR channel of substantial request to meet the stringent recurrence detail. Regularly these channels need to help high inspecting rate for fast computerized correspondence.

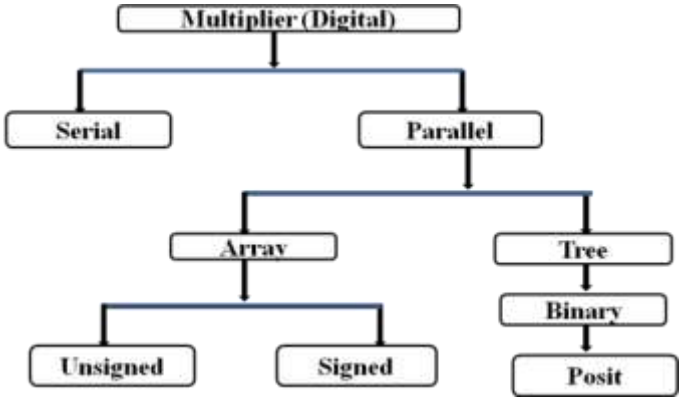

Figure 1: Types of digital multiplier

Figure 1 showing different types of multiplier, posit multiplier is part of binary multiplier but it is also applicable in signed and unsigned multiplier.

The posit number system is proposed as a replacement of IEEE floating-point numbers. It is a floating-point system that trades exponent bits for significand bits, depending on the magnitude of the numbers. Thus, it provides more precision for numbers around 1, at the expense of lower precision for very large or very small numbers. Several works have demonstrated that this trade-off can improve the accuracy of applications. However, the variable-length exponent and significand encoding impacts the hardware cost of posit arithmetic.

In FIR channel planned, will utilized structure any multipliers, if last successive years, the MCM method will utilized, as an of FIR channel plan, yet the disadvantage is MCM system won't work both thing of marked and un-marked task, so it will we have to configuration separate MCM for marked and unsigned augmentation. So here, we are a MCM with Adjusted based inexact multiplier that incorporates both marked and unsigned task in single multiplier, this multiplier will executed in FIR Channel, and demonstrated the productivity of territory, power and delay.

Duplication of paired numbers can be disintegrated into increments. Consider the increase of two 8-bit numbers A and B to create the 16 bit item P.

$$P(m+n) = A(m)B(n) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} a_i b_j 2^{i+j}$$

The equation for the addition is: .

- If the LSB of Multiplier is '1', at that point include the multiplicand into a collector.
- Shift the multiplier one piece to the right and multiplicand one piece to one side.
- Stop when all bits of the multiplier are zero.

Equipment usage of digital signal preparing (DSP) algorithms and mixed media applications in advances, for example, field programmable gate arrays (FPGAs) and digital signal processors requires countless. Frequently, the general execution of the plan is restricted by imperatives on the speed, vitality utilization, and zone necessities of the accessible multiplier structure choices. This is especially valid for applications revolved around current handheld sight and sound gadgets, where physical size, chip area and power. Therefore, research has been centered on the improvement of effective, propelled multiplier procedures to help these requesting applications.

II. PROPOSED METHODOLOGY

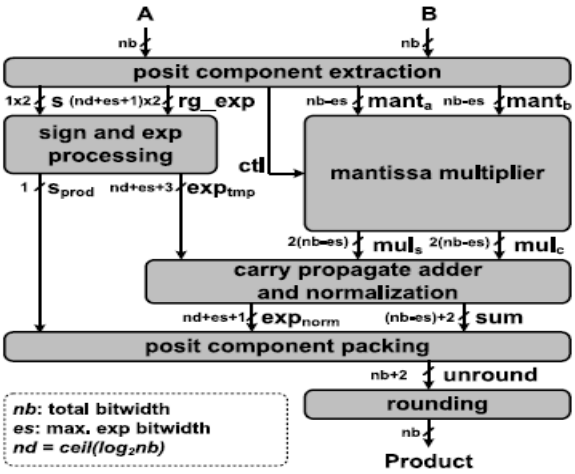

Figure 2: Flow Chart

As has been mentioned, while posit encoding may differ from usual floating-point, the core of the operations is quite similar between these number formats, with exception on the decoding and encoding of the posit fields [6]. In addition to this, in the PNS there are no special cases to being taken care of, as the denormal numbers in the case of floatingpoint based formats, a single rounding mode, i.e. round to nearest even, and unique representations for zero and infinite values. Provided that a posit number X is represented by the tuple (SX, KX, EX, FX), where SX, KX, EX, FX, are the sign, regime, exponent and fraction values, respectively, the multiplication of two posit values C = A × B is depicted in Fig. 2. The computation of the different fields is defined by (3)

Thus, the numerical value X of a generic Posithn, is expressed by (1).

X = (−1)s × (22 es) k × 2 e × (1 + f), (1)

k = −xn−2 + xi6=Xxn−2 i=n−2 (−1)1−xi (2)

S = SA ⊕ SB, (3)

K = KA + KB, (4)

E = EA + EB, (5)

F = (1 + FA) × (1 + FB) (6)

Note that (1 + FA) (respectively 1 + FB) is obtained by apending a hidden bit with value 1 to the binary representation of FA (respectively FB).

Therefore, the resulting posit C is obtained as described in (7) to (10).

SC = S, (7)

KC = (K if EC ≥ E, K + 1 otherwise, (8)

EC = (E mod 2es if F < 2, (E + 1) mod 2es otherwise, (9)

C = (F − 1 if F < 2, F/2 − 1 otherwise. (10)

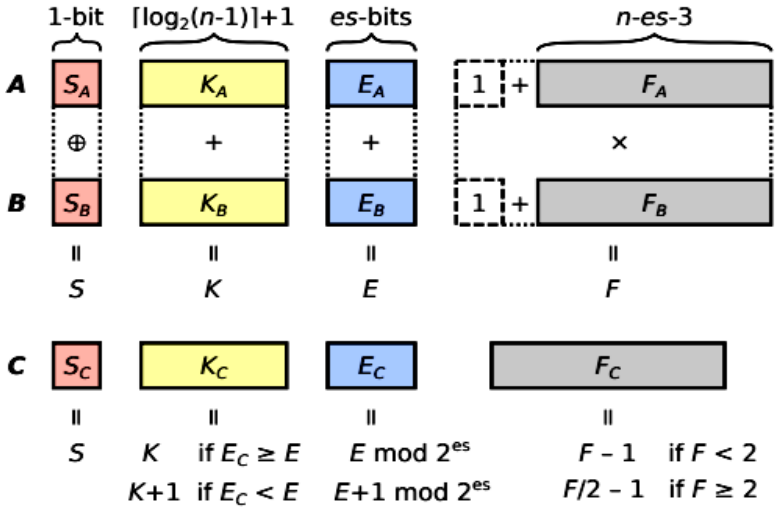

Figure 3: Posit multiplication.

Advantages:

•Common Multiplier structure for Marked and Unsigned Activity
•Less Rationale measure
•Less Power and delay

III. SIMULATION AND RESULTS

The implementation and simulation is performed using Xilinx ISE 14.7 software. The Isim simulator is used to check the result validation in the Xilinx test bench

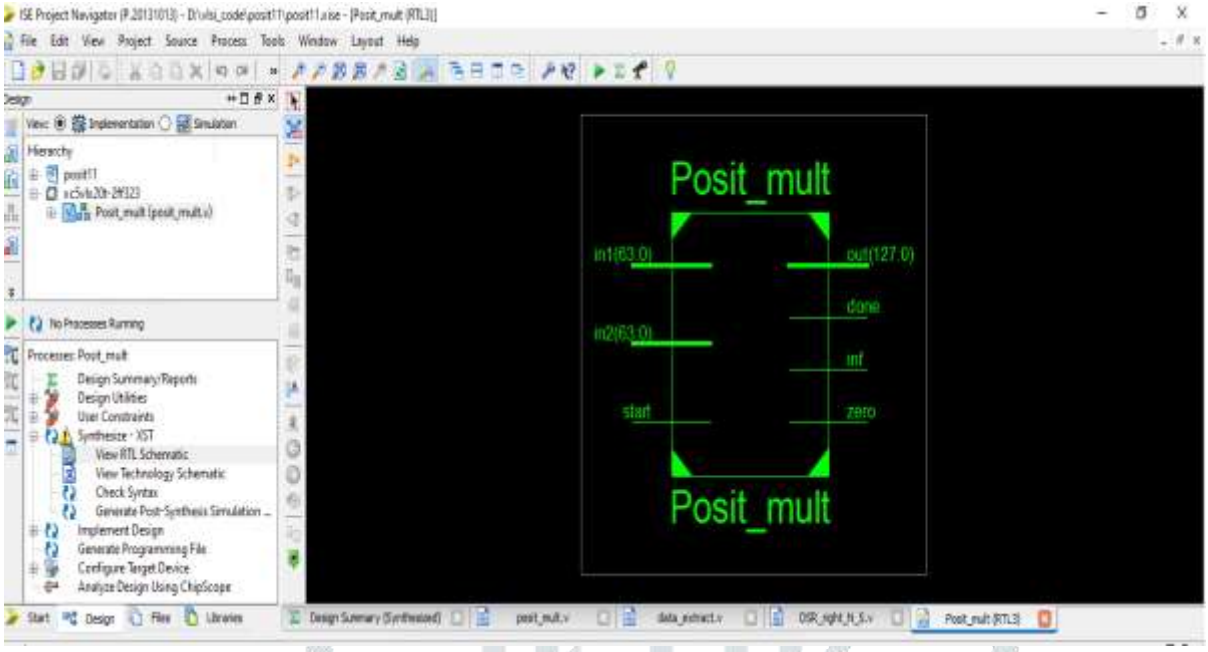

Figure 4: Posit multiplier pin details

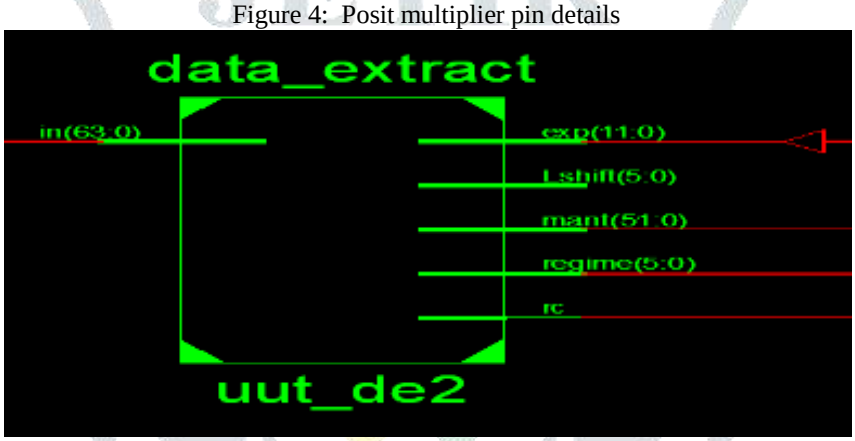

Figure 5: Data extract of posit

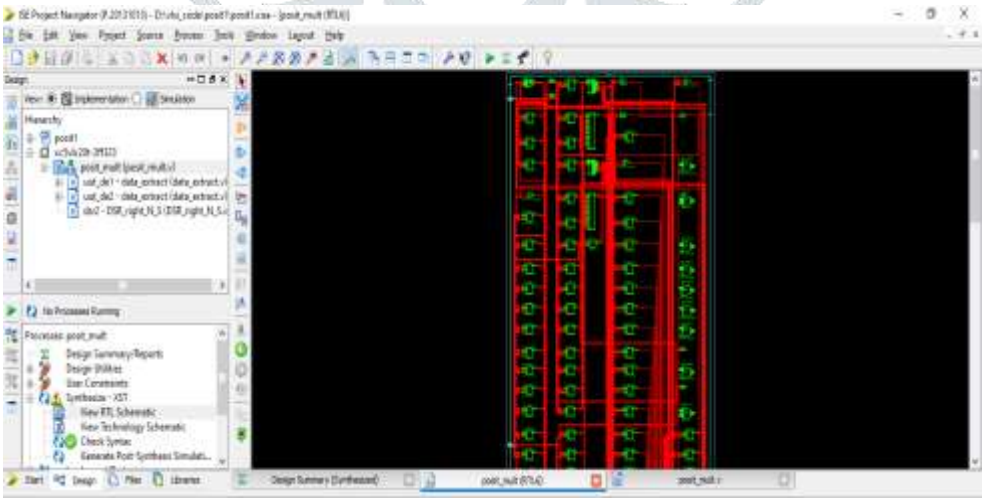

Figure 6: Complete RTL View

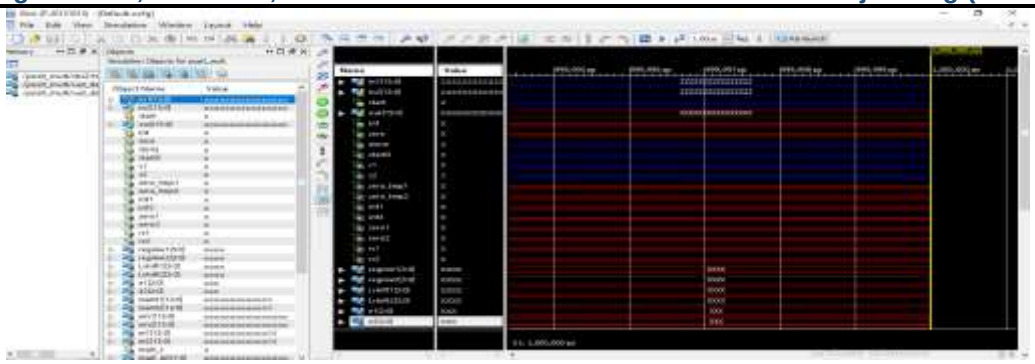

Figure 7: High impendence test bench bar (Top)

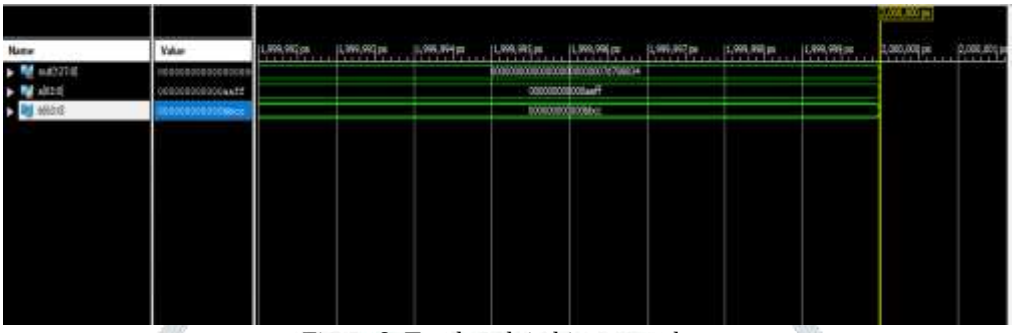

Figure 8: Test bench in binary number

Figure 8 is showing the test bench results, here input 'a' is aaff and input 'b' is bbcc. After the posit multiplication the output 'c' is 7d708834.

Table 1: Comparison with Previous and proposed work

Sr No.	Parameters	Previous work [1]	Proposed work
1	Type of Multiplier	32 Bit posit multiplier	64 Bit posit multiplier
2	Area	18979	1940
3	Delay	49 ns	36.47 ns
4	Power	43.64 mW	36 mW
5	PDP (Power delay product)	2138	1312

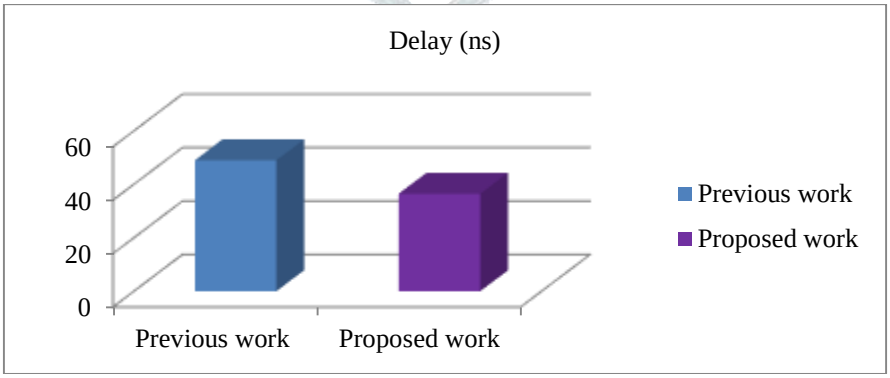

Figure 9: Delay

In figure 9, showing delay of proposed work and previous work. This is graphical representation of result and it is clear that proposed method can be calculate fast so that overall system speed will be improved.

IV. CONCLUSION

Therefore in this research work, implementation and performance improvement of 64 bit posit multiplier for advance digital signal processing and it is clear that such multiple is capable to give fast multiplication of digital signal. Less time and consume less

area. The optimized delay value is 36.47 ns and number of component using 1940. The simulation results is achieved significant improvement in the performance parameters.

REFERENCES

[1]. H. Zhang and S. -B. Ko, "Design of Power Efficient Posit Multiplier," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 861-865, May 2020, doi: 10.1109/TCSII.2020.2980531.

[2]. R. Murillo, A. A. Del Barrio and G. Botella, "Customized Posit Adders and Multipliers using the FloPoCo Core Generator," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180771.

[3]. L. Sommer, L. Weber, M. Kumm and A. Koch, "Comparison of Arithmetic Number Formats for Inference in Sum-Product Networks on FPGAs," 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2020, pp. 75-83, doi: 10.1109/FCCM48280.2020.00020.

[4]. H. Zhang and S. -B. Ko, "Design of Power Efficient Posit Multiplier," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 861-865, May 2020, doi: 10.1109/TCSII.2020.2980531.

[5]. Y. Uguen, L. Forget and F. de Dinechin, "Evaluating the Hardware Cost of the Posit Number System," 2019 29th International Conference on Field Programmable Logic and Applications (FPL), 2019, pp. 106-113, doi: 10.1109/FPL.2019.00026.

[6]. M. K. Jaiswal and H. K. -. So, "PACoGen: A Hardware Posit Arithmetic Core Generator," in IEEE Access, vol. 7, pp. 74586-74601, 2019, doi: 10.1109/ACCESS.2019.2920936.

[7]. H. Zhang, J. He and S. Ko, "Efficient Posit Multiply-Accumulate Unit Generator for Deep Learning Applications," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019, pp. 1-5, doi: 10.1109/ISCAS.2019.8702349.

[8]. R. Chaurasiya et al., "Parameterized Posit Arithmetic Hardware Generator," 2018 IEEE 36th International Conference on Computer Design (ICCD), 2018, pp. 334-341, doi: 10.1109/ICCD.2018.00057.

[9]. M. K. Jaiswal and H. K. -. So, "Universal number posit arithmetic generator on FPGA," 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018, pp. 1159-1162, doi: 10.23919/DATE.2018.8342187.

[10]. S. Bhakar and D. P. Bhatt, "Detection time of machine by posit algorithm through augmented reality," 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), 2017, pp. 71-78, doi: 10.1109/INTELCCT.2017.8324023.

[11]. A. Alavian and M. C. Rotkowitz, "Improving ADMM-based optimization of Mixed Integer objectives," *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, 2017, pp. 1-6.

[12]. D. De Caro, E. Napoli, D. Esposito, G. Castellano, N. Petra and A. G. M. Strollo, "Minimizing Coefficients Wordlength for Piecewise-Polynomial Hardware Function Evaluation With Exact or Faithful Rounding," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 5, pp. 1187-1200, May 2017.

[13]. Mang Liao and A. Chakrabortty, "A Round-Robin ADMM algorithm for identifying data-manipulators in power system estimation," *2016 American Control Conference (ACC)*, Boston, MA, 2016, pp. 3539-3544.

[14]. J. Hormigo and J. Villalba, "Measuring Improvement When Using HUB Formats to Implement Floating-Point Systems Under Round-to-Nearest," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2369-2377, June 2016.

[15]. E. G. Walters, "24-bit significand multiplier for FPGA floating-point multiplication," *2015 49th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2015, pp. 717-721.