International Conference on Recent Trends in Engineering Science and Technology (ICRTEST 2017) ISSN: 2321-8169
Volume: 5 Issue: 1(*Special Issue 21-22 January 2017*) 291 - 294

Design of 32-bit MAC Unit for complex Numbers in VHDL

Prof. R. H. Deshmukh,
Assistant professor, Department of Electronics and Communication Eng.,
SDCOE, Selukate, Wardha

Abstract:- This paper describes the 32*32 bit MAC Unit designed by using DADDA Multiplier algorithm.In recent year, Multiply Accumulate (MAC) unit developing for various high performance application.MAC unit performs multiplication and accumulation process. DADDA Multiplier has been used in the MAC Unit and comparison done based on the power, speed and area. This proposed method is to achieve low propogation delay, resource utilization and to increase the speed of processor. Basic MAC unit consist of multiplier, adder and accumulator. Speed of the multiplier is very important to any digital signal processors (DSPs).In this paper, 32-bit DADDA Multiplier and 64-bit carry look ahead adder are used. Carry look ahead adder (CLA) are widely used because of the solves this problem by calculating the carry signal in advance based on the input signal. The proposed research work is coded in VHDL & analysis in terms of speed, power & area is done on using Xilinx ISE 13.1 tool.

Keywords-*MAC unit,DADDAMultiplier,carry look ahead adder (CLA), VHDL,Xilinx ISE 13.1*

*****

I. INTRODUCTION

The multiply-accumulate operation is common step that compute the product of two numbers and add that product to an accumulator. The Hardware unit that performs the operation is known as Multiply Accumulate (MAC). With the increasing popularity of the smart phones and TABs, speed of the processor has become so important nowadays. The need for the processor's speed is more exploited for gaming and multimedia application purposes, it can also be arranged in fields like medical for faster diagnosis, in automation industry for higher throughput and so on. Since most of the signal processing operations is done by adders and multipliers units, efficient design of these units increases the speed of the processor. MAC unit is mainly used in several multimedia applications, careful design of this MAC unit leads to the design of high performance processor. The speed of the conventional MAC unit is optimized by using various frontline multipliers like Wallace tree multiplier, Booth Multiplier, Baugh-Wooley multiplier.

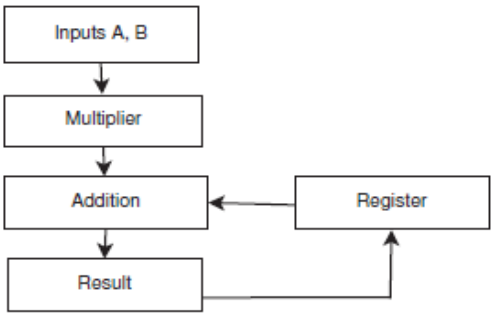

Fig 1: Block Diagram of MAC

The MAC operation modifies an accumulator :

$$a \leftarrow a + (b \times c)$$

Based on the proposed technique 8, 16, 32 and 64-bit DADDA multiplier are developed & compared with the regular DADDA multiplier. A closer reconsideration of Wallace and Dadda multipliers has been done and proved that the Dadda multiplier is slightly faster than the Wallace multiplier and the hardware required for Dadda multiplier is lesser than the Wallace multiplier [5-6]. Since the Dadda multiplier has a faster performance, we implement the proposed techniques in the same and the improved performance.

II. RELATED WORK

Mohamed AsanBasiri Met.al [1] have proposed a MAC Design in Digital Filters with Complex Numbers. They designed a novel fixed point complex number multiply accumulate circuit, which is used in real time digital signal processing applications.The architecture consists of multiplier-cum-accumulator which can be used as multiplier as well as MAC (multiplication-cum-accumulation). In the proposed architecture, the accumulation can be done along with multiplication (multiplication-cum-accumulation).In conventional radix-2 pipelined architectures the previous MAC result is added with the first carry save stage of the multiplier-cum-accumulator.

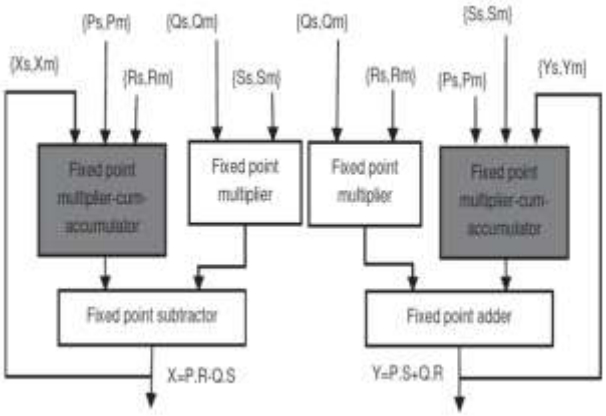

Fig.2 Proposed fixed point complex number multiplier-cum-accumulator

So 2 stage pipeline only is possible. But in this proposed MAC Design, the previous MAC result is added with the last carry save stage of the multiplier-cum accumulator. Fixed point complex number multiplier-cum-accumulator [1] And hence n stage pipeline is possible which tends to increase the operating frequency. In MAC complex multiplier the real and imaginary parts are computed by sending theprevious MAC result as one of the partial product to the present multiplication. So the separate accumulator circuit is avoided. The paper [2] and [6] shows the fixed point MAC design using carry save array multiplier. The drawback with this approach is the result will be produced at every k th clock cycle in a kstage pipelined system. The paper [3] and [7] explain the fixed point MAC design using array multiplier with two stage of the multiplier are sent to the first stage along with partial result.The paper[4] shows the reduction of critical depth of the MAC, where the N bit accumulator is divided into two N/2 bit adders.Here the accumulation can be done in two cycles. The paper [5]shows an FIR filter based on Multiply-Accumulator (MAC) unit to suppress the PLI noise.

Contribution of this paper-In general,MAC operation can be done with multiplicationoperation can be done with multiplication followed by accumulation. So the depth of the MAC circuit is depending on the multiplier andaccumulation circuit. In the proposed architecture, the accumulation can be done along with multiplication (multiplication-cum-accumulation).That is, the previous MAC result is added along with the partial product of the current multiplication. And hence the separateaccumulator circuit is avoided.In this paper, fixed point complex number multiply accumulate circuit is proposed architecture usingDADDA multiplier with/without pipeline. The experimental result of the proposed architecture is compared with the conventional fixed pointcomplex number MAC architecture. The rest of the paper is organized as, section III states theproposed architecture for fixed point complex number MAC.

III. THE PROPOSED ARCHITECTURE OF FIXED POINTCOMPLEX NUMBER MAC

In general, a digital signal can be represented as amplitude (r) with phase (θ), which can be written as a complex number z = r<θ = x + jy, where r = √x2 + y2 and θ = tan−1(y /x). The two complex numbers are represented as (P + jQ), and (R + jS), where P = {Ps, Pm}, Q = {Qs,Qm},R = {Rs,Rm} and S = {Rs,Rm}. Here the suffix represents the sign bit and m represents binary number. Thecomplex number multiplication can be done in several ways.

Let the real part of the multiplication of the two complexnumbers is X and imaginary part is Y , where X = {Xs,Xm} and Y = {Ys, Ym}. According to (3), FPCN multiplication requires three fixed point multipliers and five fixed pointadders.

$$X + jY = (P + jQ)(R + jS) \text{ ---------- (2)}$$

$$X+jY=\{(P-Q)S+(R-S)P\}+j\{(P-Q)S+Q(R+S)\} \text{ ----------(3)}$$

$$X + jY = \{PR - QS\} + j\{PS + QR\} \text{ ----------(4)}$$

Here the multiplication will start after computing (P −Q), (R−S) and (R+S). According to (4), FPCN multiplicationrequires four fixed point multipliers and two fixed point adders. Here the multiplication can start immediately.Because of one additional multiplier, hardware area for (4) is higher than (3). But the time complexity for (4) is log2 n depth lesser than (3). So the proposed architecture is following (3).

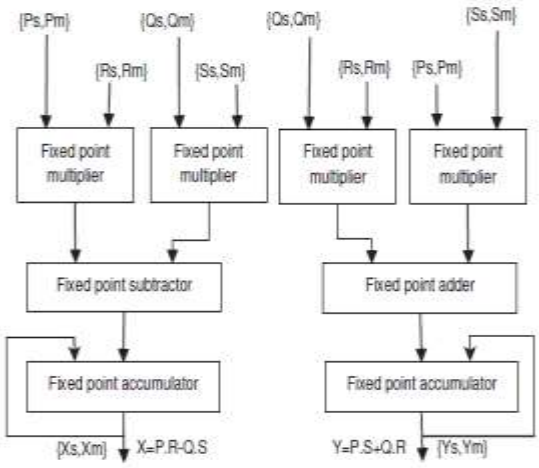

Fig.3 conventional fixed point complex number multiplier-cum-accumulator

ISSN: 2321-8169
291 - 294

The conventional fixed point complex number multiplier-cum-accumulator is shown in Fig. 3, where four fixed pointmultipliers and four fixed point adders are used. In the proposed architecture two fixed point multiplier-cum-accumulators, two fixed point multipliers and two fixed point adders are involved. The Fig. 4 shows the proposed fixed point complex number multiplier-cum-accumulator. So one extra adder depth can be avoided in proposed architecture.

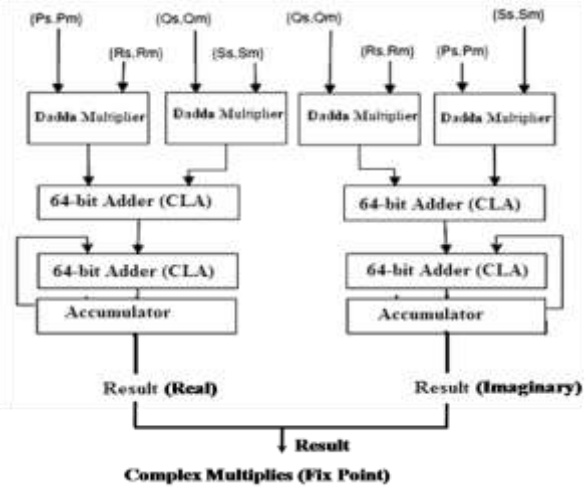

Fig.4-Proposed fixed point complex number multiplier-cum-accumulator

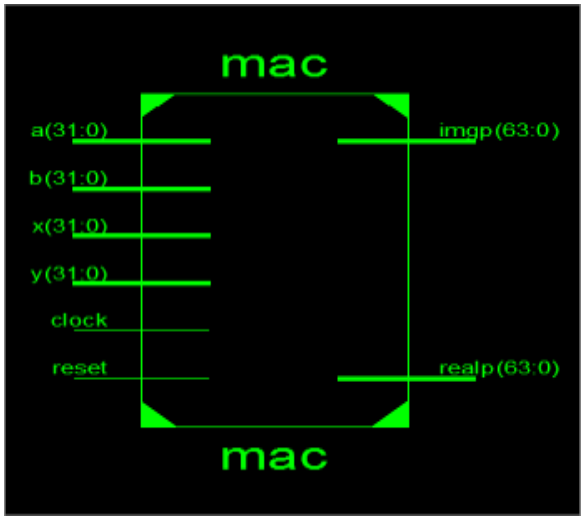

Fig.5 32-bit MAC Unit

Fig.6 shows the 64-bit CLA Adder .The 64*64 bit are input and the output is 64-bit.The methodology is multiply 64*64 bit and calculating partial product term .That caused by 64-bit output and the above partial product are formed these partial value place in triangular form .Perform addition from bottom part of CLA(Carry look ahead adder).These calculation display result. The fixed point multiply-cum-accumulator circuit consists of three parts namely partial conventional, proposed complex number MAC without and with pipeline respectively. But this proposed technique is products generation, carry look ahead adder and sign calculation. The Fig. shows 64-carry look ahead adder.

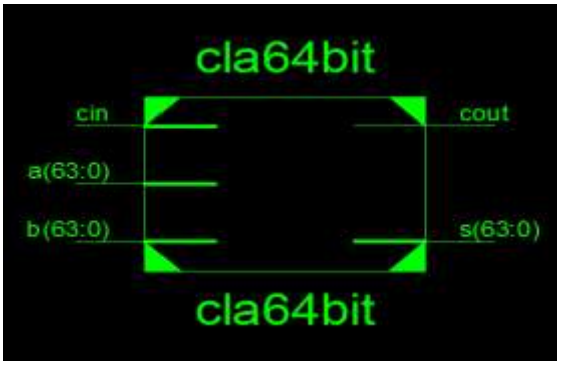

Fig.6 64-bit CLA Adder

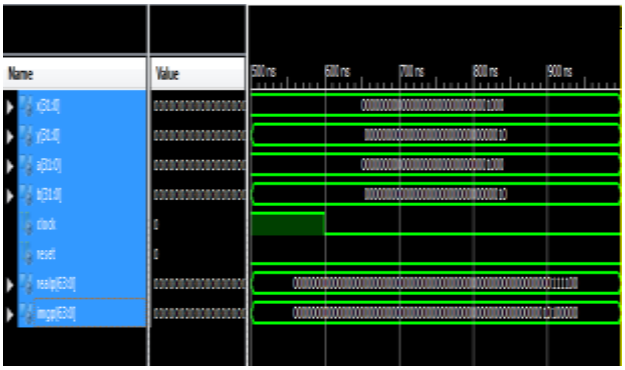

Fig.6 -Waveform of 32-bit DADDA multiplier

IV. FLOW CHART OF 32-BIT DADDA MULTIPLIER

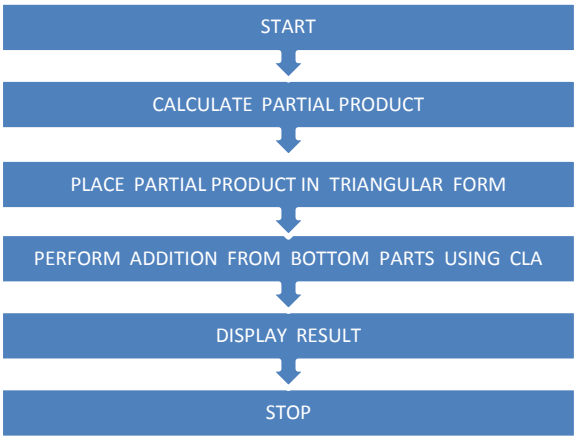

Fig 7- flow chart of multiplier

The whole system is modeled in Verilog HDL. These Verilog HDL models are simulated and verified using Xilinx ISE simulator. The timing, area and power analysis of this implementation has been done with Xilinx ISE 13.1 design tool. The proposed architecture is compared with the

designed only for time optimization and hence, there is no significant difference between proposed and conventional system in power/area perspective.

Table 1 :Camparison of MAC Unit

Parameter	Proposed Work	Ref. Paper [i]	Ref. Paper [ii]	Ref. Paper [iii]
Bits	32-bit	8-bit	16-bit	16-bit
AREA (LUT's)	9373	103	1750	11403
DELAY (ns)	82.847	3.65	2.15	11.4
POWER (uW)	$1.203*10^{-6}$	$2.14*10^{-6}$	14.62	493648.8

V. CONCLUSION

In our design, efforts have been made to reduce the area and propagation delay and achieve an improvement in the reduction of area. In thispaper, a high performance 32-bit fixed point complex number MAC is proposed, where the real and imaginary parts can be computed bysending the previous MAC result as one of the partial product to the present multiplication. So the depth of the MAC is equal to the depth of themultiplier. And hence the separate accumulator circuit are formed. The experimental results are showing the proposed fixed point complexnumber MAC is giving better performance than the conventional fixed point complex number MAC.

VI. ACKNOWLEDGEMENTS

This work is partially supported by my wife Prof. R.R.Deshmukh for her valuable support and encouragement. She kindly read my paper and offered invaluable detailed advices on grammar , organization , and the theme of the paper.

REFERENCES

[1] Mohamed AsanBasiri M, Noor MahammadSk, "An Efficient Hardware Based MAC Design in Digital Filters with Complex Numbers", in IEEE International Conference on Signal Processing and Integrated Networks, 2014

[2] Rahul Narasimhan. A, R. Siva Subramanian, "High Speed Multiply-Accumulator Coprocessor Realized for Digital Filters", in IEEE International Conference on Electrical, Computer & Communication Technologies, 2015.

[3] MarojuSaiKumar, D.Ashok Kumar, Dr.P.Samundiswary, "Design and Performance Analysis of Multiply-Accumulate (MAC) Unit", in IEEE International Conference on Circuit, Power and Computing Technologies, 2014.

[4] K. Paldurai, K. Hariharan, G.C. Karthikeyan, K. Lakshmanan, "Implementation of MAC using Area Efficient and Reduced Delay Vedic Multiplier Targeted at FPGA Architectures", in IEEE International Conference on Communication and Network Technologies, 2014.

[5] RamandeepKaur, Rahul Malhotra, Sujay Deb, "MAC based FIR Filter: A novel approach for Low-Power Real-Time De-noising of ECG signals", in IEEE 19th International Symposium on VLSI Design & Test, 2015.

[6] F. Elguibaly, "A fast parallel multiplier accumulator using the modified Booth algorithm", IEEE Transactions on Circuits Systems, vol. 27, no.9, pp. 902-908, Sep. 2000.

[7] Young-Ho Seo and Dong Wook Kim, "A new VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm", IEEE Transactions on VLSI systems, vol.18, no.2, Feb.2010.

[8] Davide De Caro, NicolaPetra, Antonio Giuseppe Maria Strollo, Fabio Tessitore, Ettore Napoli, "Fixed-Width Multipliers and Multipliers-Accumulators With Min-Max Approximation Error", in IEEE Transactions On Circuits And Systems—I: Regular Papers, Vol. 60, No. 9,September 2013, pp 2375 – 2389.

[9] Charles Roth, "Digital System Design using VHDL", in Cengage Learning, 2010.

[10] Douglas Perry, "VHDL Programming by Examples", in Tata Mcgraw Hill, 2002.