PARTIAL PRODUCT REDUCTION PROCESS BY INTEGRATING ADDITIONS AND ACCUMULATIONS INTO HIGH-PERFORMANCE MULTIPLY-ACCUMULATE UNIT

[1]K.Harshavardhan Reddy, [2]D.Vijaya Kumar, [3]J.Sunil Kumar, [4]Ramesh. Nosina

[1,2,3]Dept of Electronics and Communication Engineering, Sree Venkateswara College Of Engineering, Nellore (Dt), Andhra Pradesh, India.

[4]Dept of Computer Science and Engineering, Sree Venkateswara College Of Engineering, Nellore (Dt), Andhra Pradesh, India.

ABSTRACT

This research suggests a pipelined Mac architecture with minimal power and fast performance. Carry propagation of ads has higher power consumption and longer route latency; to address this issue, we suggest a way. In this, we include a portion of additions into a procedure that only partially reduces the result. The PPR procedure of subsequent multiplication does not accomplish addition or accumulation of MSB bits. The total number of carries is meant to be accumulated by a small size adder in order to properly contrast with surplus in the PPR process. Using XilinxISE14.7, the efficacy of the suggested technique is calculated.

Keywords: MAC unit, dadda multiplier, Arithmetic circuits, alpha-bit adder

INTRODUCTION

In the proposed method we use two stage MAC unit with 8 and 16 bit. In Partial Product Generation (PPG) process, PPR performed in the first stage, executes the (k+)-bit addition in the second stage to generate the accumulation result. The following is a list of the primary characteristics of this suggested architecture: We incorporate a portion of improvements into the PPR process to shorten the carry propagations. To prevent overflowing in the PPR process, a -bit the adder is employed to count all the carries. The second phase can only be carried out utilising the gating method throughout the final cycle in order to save electricity.

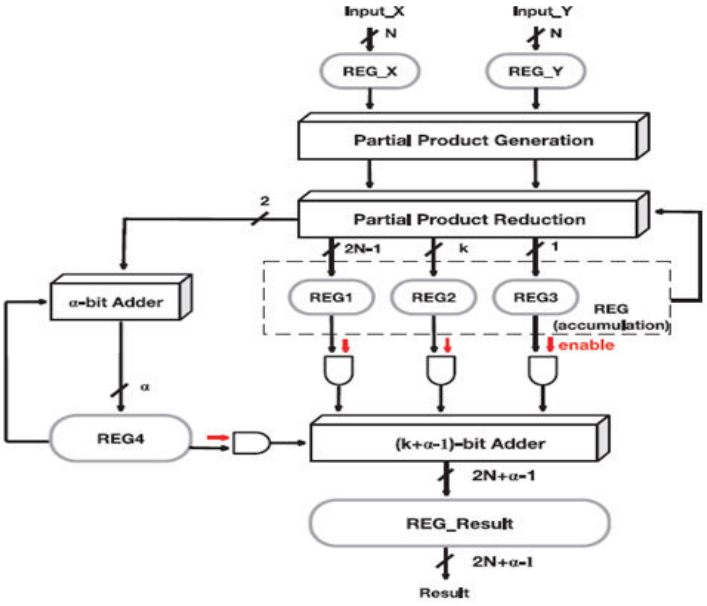


Fig.1. Architecture of Proposed MAC Unit

The proposed architecture of MAC shown in above figure. Our PPM (for the PPR procedure) is made up of two PPMs, one of which was obtained through the PPG and the other through accumulation. The final inclusion for greater importance bits is not made in the proposed MAC's initial stage. The PPM obtained from the accumulation is kept in register accumulation. As a result, register accumulation consists of three parts: REG1 (the first row), which has 2N-1 bits, REG2, which has k bits that the user can define, and REG3 (the third row), which has 1 bit. We use the Dadda tree method in the PPR procedure to condense our PPM to two rows of data. After dividing our PPM into two rows, we carry out the final (2N-k-1)-bit addition.

Since the last two rows acquired using the Dadda tree method are added using a (2N- k-1)-bit adder, a bigger k can result in less carry propagation. However, a bigger k yields a larger PPM for the PPR stage since the final accumulation and addition of k greater importance bits is carried out in the PPR procedure of the subsequent multiplication. The accumulation result is produced in the 2nd stage of the suggested MAC unit.

Register accumulation and register REG4 are the inputs for the second stage. The accumulation was successfully carried out in the next PPR phase as well as the -bit addition in the suggested MAC unit. In order to save energy, we can disable the (k+)-bit adding in other cycles if we only require the final outcome in the last cycle.

Advantages

Low power consumption

Carry propagation is not considered in this method and hence delay will be reduced

Applications

DSP

Image processing

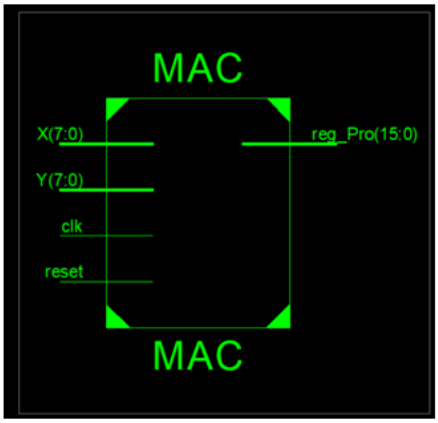
Audio Applications

Related Work

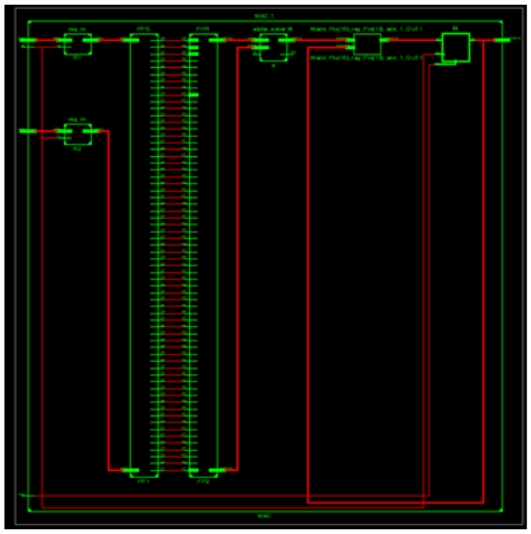
Two carry propagations—adds in multiplication and additions in accumulation—must be carried out in the traditional MAC unit. The dada multiplier is used for multiplication in the MAC unit. The Dadda method varies from the Wallace method in that it only uses the bare minimum of full and half adders to achieve predefined stage heights. The following are the height limitations for the stage: #2, #3, #4, #6, #9, #13 #19 #28 #42 Each phase is 1.5 times as big as the one above it. The 1.5 reduction ratio is produced by the use of entire adders, which convert three partial products into two partial products. Therefore, the greatest elevation of the following stage is equal to twice that of the stage before it. Dadda uses the fewest possible adders to get to the CPA stage in an effort to improve the region of the multiplier.

SIMULATION RESULTS

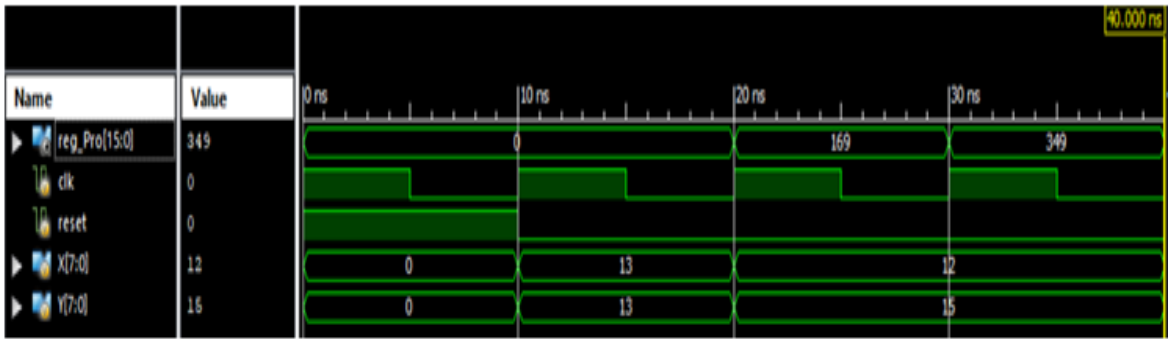
RTL



INTERNAL BLOCK DIAGRAM



SIMULATION RESULTS



CONCLUSION

We have observed several MAC unit applications for a variety of purposes. The Booth multiplier is the one used in MAC units since it operates at the fastest pace while using the least amount of electricity. The primary hardware for the systems in digital signal processing is the MAC unit, which is in high demand. All types of mathematical operations, including addition, multiplication, division, squares, and square roots, are performed

using MAC units. The MAC unit needs to be better in terms of complexity, area, delay, power consumption, and speed.

REFERENCES

1. Meenu S Ravi1, R H Khade and Ajit Saraf, “Design of Fast Floating Point Multiply Accumulate Unit using Ancient Mathematics for DSP Applications.” European Journal of Advances in Engineering and Technology, pp.55-59, 2015.
2. Ku. Shweta N. Yengade, Associate Prof. P. R. Indurkar, “Review On Design Of Low Power Multiply And Accumulate Unit Using BaughWooley Based Multiplier.” International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 02 Feb -2017.
3. Shaik Nasar, K. Subbarao, “Design & Implementation of MAC Unit Using Reversible Logic.” International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue5, September-October 2012, pp.1848-1855
4. G.Indira, G. Madhusudhana Rao, P.Jaya Babu, M. Ravi Kiran, “An Efficient Architecture of MAC Unit with LAW Multiplier.” International Journal of Advanced Scientific Technologies in Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.2, Issue.11, November.2016
5. 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.
6. 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.
7. Michael F. Cowlisha, “Decimal FloatingPoint: Algorism for Computers.” Proceedings of the 16th IEEE Symposium on Computer Arithmetic (ARITH'03) IEEE, 2003
8. A. D. Robison, “N-Bit Unsigned Division via N-Bit Multiply-Add,” Proceedings of the 17th IEEE Symposium on Computer Arithmetic, pp. 131-139, 2005.
9. Singh, Harpreet, and Chakshu Goel, "Design Approaches for a Novel Reversible 4-bit Comparator." IJMSC-International Journal of Mathematical Sciences and Computing (IJMSC) (2015).