



IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com

+A High-Performance MAC Unit by Integrating Additions and Accumulations into PPR Process

**Dr R.Gomala Valli M.E.Ph.D¹, T.Chandi Priya², B.Deepa³, D.Charan Kumar⁴, Y.Harshitha⁵,
G.Dilli Prasad⁵**

Professor, Department of Electronics and Communication Engineering, Siddharth Institute of Engineering & Technology, Puttur, India ¹

Department of Electronics & Communication Engineering, Siddharth Institute of Engineering & Technology Puttur, India^{2,3,4,5}

ABSTRACT: In this paper, we propose a low-power high-speed pipeline multiply-accumulate (MAC) architecture. In a conventional MAC, carry propagations of additions (including additions in multiplications and additions in accumulations) often lead to large power consumption and large path delay. To resolve this problem, we integrate a part of additions into the partial product reduction (PPR) process. In the proposed MAC architecture, the addition and accumulation of higher significance bits are not performed until the PPR process of the next multiplication. To correctly deal with the overflow in the PPR process, a small-size adder is designed to accumulate the total number of carries. Compared with previous works, experimental results show that the proposed MAC architecture can greatly reduce both power consumption and circuit area under the same timing constraint.

KEYWORDS: Digital Circuits, Logic Circuits, Multiplying Circuits, Pipeline Processing, Power Dissipation.

INTRODUCTION

A basic MAC unit consists of an adder, multiplier and an accumulator. Usually, adders implemented are Carry- Select or Carry-Save adders, as speed is of utmost importance in DSP. But if on chip area is to be considered then, Carry increment Adder can be used with an additional advantage of high speed.

One implementation of the multiplier could be Vedic Multiplier.

1.1 Operation of MAC Unit

The inputs of MAC unit are fetched from memory cell and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder and then store the result in to a memory location. This entire process is to be achieved in a single clock cycle. The proposed design consists of 32-bit floating point multiplier based on Vedic multiplication technique, 65-bit accumulator using carry select adder. To multiply the values of A and B, Floating point Vedic multiplier is used instead of conventional multiplier because Vedic multiplier can increase the speed of MAC unit. Carry Save Adder is used as an accumulator in this design. Apparently, together with the utilization of Vedic multiplier approach, Carry select adder as the accumulator, This MAC unit design is not only reducing the standby power consumption but also can enhance the MAC unit speed so as to gain better system performance. The below figure shows the block diagram of proposed MAC Unit consisting of 32 bit IEEE 754 based Vedic Multiplier unit. The result of multiplier is provided to the input of Accumulator through Carry select adder.

In the proposed design of MAC Unit the Multiplier used is a 32 BIT IEEE 754 floating point multiplier based on Vedic multiplication technique. The multiplier for the floating point numbers represented in IEEE 754 format can be divided in four sections: Mantissa Calculation Unit, Exponent Calculation Unit, Sign Calculation Unit, and Control Unit.

The Vedic Multiplication technique is chosen for the implementation of this unit. This technique gives promising result

in terms of speed and power. The Vedic multiplication system is based on 16 Vedic sutras or aphorisms, which describes natural ways of solving a whole range of mathematical problems. Out of these 16 Vedic Sutras the Urdhvatriyakbhyam sutra is suitable for this purpose.

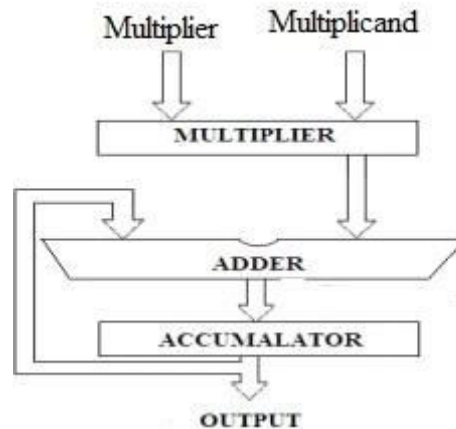


FIG:1.Block Diagram of MAC unit

The multiply-accumulate (MAC) unit is a fundamental block for digital signal processing (DSP) applications. Especially, in recent years, the development of real-time edge applications has become a design trend. Thus, there is a strong demand for high-speed low-power MAC units.

A conventional MAC unit is composed of two individual blocks: a multiplier and an accumulator (i.e., an accumulate adder). An N -bit MAC unit includes an N -bit multiplier and a $(2N+\alpha-1)$ -bit accumulator (adder), where α is the number of guard bits used to avoid overflow (caused by long sequences of multiply-accumulate operations). A lot of previous works paid attention to the optimization of multiplier and the optimization of adder, respectively.

A multiplier usually has three steps. The first step is the partial product generation (PPG) process. For example, AND gates can be used to generate a partial product matrix (PPM) for an unsigned multiplication. The second step is the partial product reduction (PPR) process. By using the Dadda tree approach or the Wallace tree approach, the PPM can be reduced to become two rows. The third step is the final addition. An adder (called the final adder) is used to perform the summation of the final two rows. For an N -bit multiplier, a $(2N-1)$ -bit adder is required for the final addition.

II. EXISTING METHOD

In a conventional MAC unit, carry propagations of additions (including final additions in multiplications and additions in accumulations) often result in large power consumption and large path delay. To resolve this problem, we are motivated to reduce the lengths of carry propagations in the final addition and the accumulation. Our basic idea is to integrate a part of additions (including a part of the final addition and a part of the accumulation) into the PPR process. As a result, the lengths of carry propagations can be reduced.

A conventional MAC unit is composed of two individual blocks: a multiplier and an accumulator (i.e., an accumulate adder). An N -bit MAC unit includes an N -bit multiplier and a $(2N+\alpha-1)$ -bit accumulator (adder), where α is the number of guard bits used to avoid overflow (caused by long sequences of multiply-accumulate operations). A lot of previous works paid attention to the optimization of multiplier and the optimization of adder, respectively. A multiplier usually has three steps. The first step is the partial product generation (PPG) process. For example, AND gates can be used to generate a partial product matrix (PPM) for an unsigned multiplication. The second step is the partial product reduction (PPR) process. By using the Dadda tree approach or the Wallace tree approach, the PPM can be reduced to become two rows. The third step is the final addition. An adder (called the final adder) is used to perform the summation of the final two rows. For an N -bit multiplier, a $(2N-1)$ -bit adder is required for the final addition.

III. PROPOSED METHOD

In this proposed method, we present the proposed two-stage (i.e., two cycle) MAC architecture. The first stage performs the PPG process, the PPR process (based on the PPM that combines the PPG result and the accumulation result), the $(2N-k-1)$ -bit addition (i.e., a part of the final addition) and the α -bit addition (for dealing with the overflow in the PPR process). Then, the second stage performs the $(k+\alpha)$ -bit addition to produce the accumulation result.

The main features of the proposed architecture are below:

- To reduce the lengths of carry propagations, we integrate a part of additions into the PPR process.
- To handle overflow in the PPR process, an α -bit adder is used to count the total number of carries.
- By applying the gating technique, the second stage can only be executed in the last cycle (of the entire sequence of multiply-accumulate operations) for power saving.

The proposed two-stage pipeline MAC unit is displayed in Fig. 2. Our PPM (for the PPR process) is composed of two PPMs: one PPM is derived by the PPG and the other PPM is derived by the accumulation.

For an unsigned MAC unit, in the PPG process, “AND” gates can be directly used to generate the PPM. For a signed MAC unit, because the influences of the sign bit should be taken into account, several PPG algorithms have been proposed to generate the signed PPM. In the proposed architecture, the Baugh-Wooley algorithm is adopted in the PPG process to generate the signed PPM.

In the first stage of the proposed MAC unit, we only perform the $(2N-k-1)$ -bit final addition. In other words, the final addition of higher significance bits is not performed. Register *accumulation* is used to store the PPM derived by the accumulation (i.e., the result after the PPR process and the $(2N-k-1)$ -bit final addition). Thus, register *accumulation* includes three parts: REG1 (i.e., the first row) has $2N-1$ bits, REG2 (i.e., the second row) has k bits, which can define by the user, and REG3 (i.e., the third row) has 1 bit. Using our 4-bit MAC shown in Fig. 1(b) as an example, REG1 has 7 bits, REG2 has 3 bits, and REG3 has 1 bit. Note that, the initial value of each bit in register *accumulation* is ‘0’. In each cycle, the result of register *accumulation* will be combined with the PPG result to form our PPM for the PPR process.

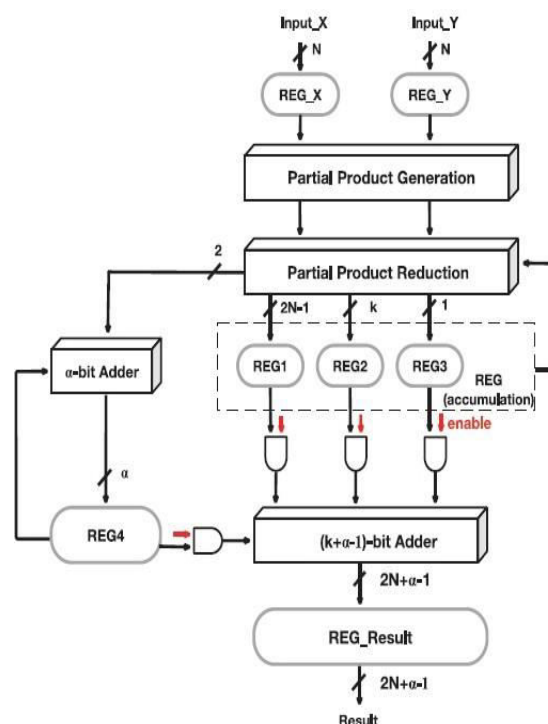


FIG 2: The proposed MAC architecture SOFTWARE REQUIREMENTS

- Xilinx ISE 14.7 version
- Verilog HDL

ADVANTAGES & APPLICATIONS ADVANTAGES:

- Critical path delays and power dissipations caused by carry propagations can be reduced.

Applications:

- Digital signal processing (DSP).
- Graphics designing,
- Multimedia .
- Image processing

Result:

RTL Schematic diagrams: 4 bit signed design

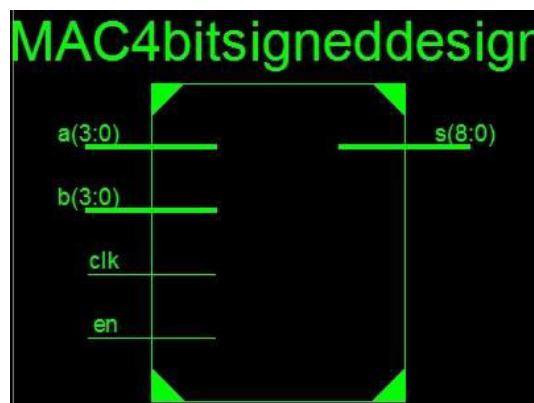


FIG 3: MAC4bitsigneddesign

4 bit unsigned design

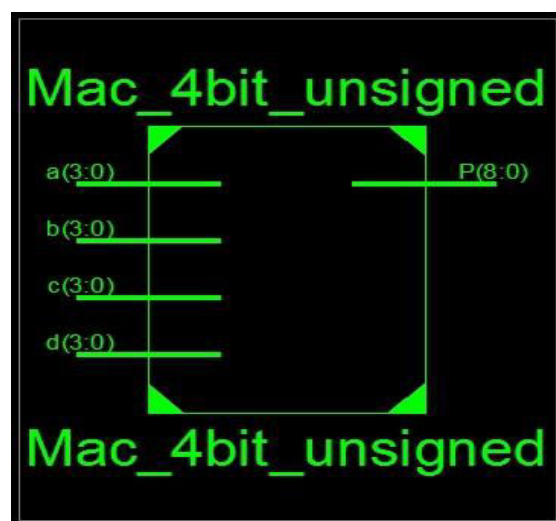


FIG 4:Mac_4bit_unsigned

Simulations:

8 bit unsigned design



FIG:5 Mac_8bit_unsigned

16 bit unsigned design

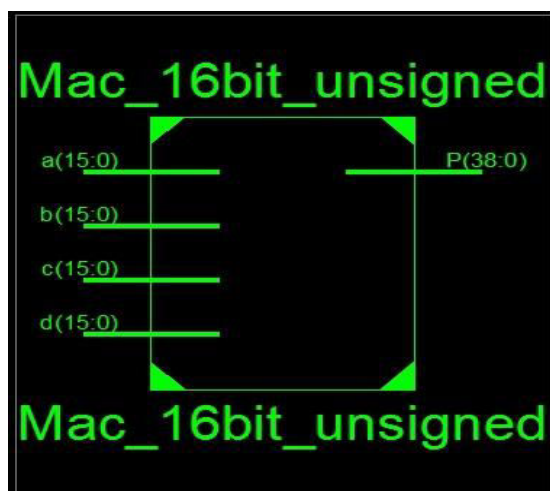


FIG:6 Mac_16bit_unsigned

4-Bit signed



FIG:7

4-Bit unsinged



FIG:8

8-Bit unsigned

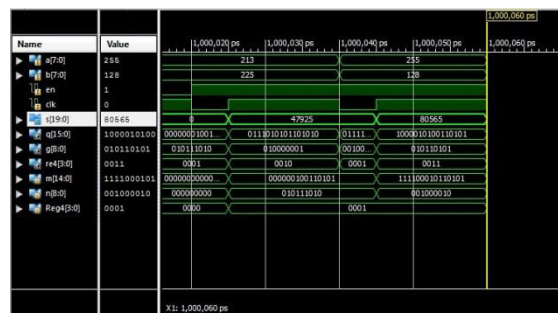


FIG:9

16-Bit unsigned



IV. CONCLUSION

To correctly deal with the overflow during the PPR process, an α -bit adder is used to count the total number of carries. The proposed MAC architecture is applicable to both the design of an unsigned MAC unit and the design of a signed MAC unit. The only differences between the unsigned MAC unit and the signed MAC unit are the PPM structure and the α -bit addition mechanism.

V. FUTURE SCOPE

Various adder architectures we can use for the trade-offs among delay, area, and power. Furthermore, we can develop various MAC unit models which can be developed by replacing the multiplier as well as the accumulator (adder) with various architectures.

REFERENCES

1. Abdelgawad, "Low Power Multiply Accumulate Unit (MAC) for Future Wireless Sensor Networks", in Proc. IEEE Int. Symp. Sensors Applications, Galveston, TX, 2013.
2. H.O. Ahmed, M. Ghoneima and M. Dessouky, "Concurrent MAC Unit Design using VHDL for Deep Learning Networks on FPGA", in Proc. IEEE
3. Int. Conf. Computer Applications and Industrial Electronics, Penang, Malaysia, 2018.
4. V. Camus, C. Enz and M. Verhelst, "Survey of Precision-Scalable Multiply-Accumulate Units for Neural-Network Processing", in Proc. IEEE Int. Conf. Artificial Intelligence Circuits and Systems,
5. Hsinchu, Taiwan, 2019
6. W.J. Townsend, E.E. Swartzlander, J.A. Abraham, "A Comparison of Dadda and Wallace Multiplier Delays", in Proc. SPIE Annual Meeting Optical Science and Technology, San Diego, CA, 2003.
7. K.L.S. Swee and L.H. Hiung, "Performance Comparison Review of 32-Bit Multiplier Designs", in Proc. IEEE Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 2012, pp. 836-841.
8. S. Asif and Y. Kong, "Design of an Algorithmic Wallace Multiplier using High Speed Counters", in Proc. IEEE Int. Conf. Computer Engineering & Systems, Cairo, Egypt, 2015, pp. 133-138.
9. C.W. Tung and S.H. Huang, "Low-Power High-Accuracy Approximate Multiplier Using Approximate High-Order Compressors", in Proc. IEEE Int. Conf. Communication Engineering and Technology, Nagoya, Japan, 2019.
10. Nagendra, M.J. Irwin and R.M. Owens, "Area-Time-Power Tradeoffs in Parallel Adders", IEEE Trans. Circuits and Systems -- II: Analog and Digital Signal Processing, vol. 43, no. 10, pp. 689- 702, Oct. 1996.
11. J. Saini, S. Agarwal and Aditi Kansal, "Performance, Analysis and Comparison of Digital Adders", in Proc. IEEE Int. Conf. Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 81-83.
12. L. Pilato, S. Saponara and L. Fanucci, "Performance of Digital Adder Architectures in 180nm CMOS Standard-Cell Technology", in Proc. IEEE Int. Conf. Applied Electronics, Pilsen, Czech Republic, 2016.
13. P. Jebashini, R. Uma, P. Dhavachelvan and
14. H.K. Wye, "A Survey and Comparative Analysis of Multiply-Accumulate (MAC) Block for Digital Signal Processing Application on ASIC and FPGA", *Journal of Applied Science*, vol. 15, no. 7, pp. 934-946, 2015.
15. P.A. Patil and C. Kulkarni, "A Survey on Multiply Accumulate Unit", in Proc. IEEE Int. Conf. Computing

Communication Control and Automation, Pune, India, 2018.

16. P.F. Stelling and V.G. Oklobdzija, "Implementing Multiply- Accumulate Operation in Multiplication Time," in *Proc. IEEE Int. Symp. Computer Arithmetic*, Asilomar, CA, 1997, pp. 99– 106.
17. A. Abdelgawad and M. Bayoumi, "High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications", in *Proc. IEEE Int. Symp. Circuits and Systems*, New Orleans, LA, 2007, pp. 3199– 3202.
18. M. D. Ercegovac and T. Lang, *Digital Arithmetic*. San Mateo, CA: Morgan Kaufmann, 2003.
19. T. T. Hoang, M. Sjalander and P. Larsson- Edefors., "A High-Speed Energy-Efficient Two- Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit", *IEEE Trans. Circuits and Systems*, vol. 57, no. 12, pp. 3073-3081, Dec. 2010.
20. B. Liebig, J. Huthmann and A. Koch, "Architecture Exploration of High-Performance Floating-Point Fused Multiply-Add Units and their Automatic Use in High-Level Synthesis", in *Proc. IEEE Int. Symp. Parallel & Distributed Processing*, Cambridge, MA, 2013, pp. 134-143.
21. A.A. Wahba and H.A.H. Fahmy, "Area Efficient and Fast Combined Binary/Decimal Floating Point Fused Multiply Add Unit", *IEEE Trans. Computers*, vol. 66, no. 2, pp. 226-239, Feb. 2017.
22. A. Peiman, Z.D. Koozehkanani and F. Nazari, "An Ultra High Speed Digital 4-2 Compressor in 65-nm CMOS", *Int. Journal of Computer Theory and Engineering*, vol. 5, no. 4, pp. 593-597, Aug. 2013.
23. C.P. Narendra and K.M.R. Kumar, "Low Power Compressor Based MAC Architecture for DSP Applications", in *Proc. IEEE Int. Conf. Signal Processing, Informatics, Communication and Energy Systems*, Kozhikode, India, 2015.
24. A. Rezai and P. Keshavarzi, "High- Throughput Modular Multiplication and Exponentiation Algorithms Using Multibit-Scan– Multibit-Shift Technique", *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1710-1719, Sep. 2015.
25. Rezai and P. Keshavarzi, "High- Performance Scalable Architecture for Modular Multiplication Using a New Digit-Serial Computation", *Microelectronics Journal*, vol. 55, pp. 169-178, Sep. 2016.
26. Rezai and P. Keshavarzi, "Compact SD: A New Encoding Algorithm and Its Application in Multiplication", *International Journal of Computer Mathematics*, vol. 94, no. 3, pp. 554-569, 2017.
27. D. Nguyen, D. Kim and J. Lee, "Double MAC: Doubling the Performance of Convolutional Neural Networks on Modern FPGAs", in *Proc. IEEE Design, Automation & Test in Europe Conference & Exhibition*, Lausanne, Switzerland, 2017.
28. M. Sjalander and P. Larsson-Edefors, "High- Speed and Low-Power Multipliers Using the Baugh-Wooley Algorithm and HPM Reduction Tree", in *Proc. IEEE Int. Conf. Electronics, Circuits and Systems*, St. Julien's, Malta, Sep. 2008, pp. 33-36.
29. J.H. Tu and L.D. Van, "Power-Efficient Pipelined Reconfigurable Fixed-Width Baugh- Wooley Multipliers", *IEEE Trans. Computers*, vol. 58, no. 10, pp. 1346-1355, Oct. 2009.
30. K. Tsoumanis, S. Xydis, C. Efstathiou, N. Moschopoulos and K. Pekmestzi, "An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator", *IEEE Trans. Circuits and Systems*, vol. 61, no. 4, pp. 1133-1143, Apr. 2014.
31. D.I. Moldovan and J.A.B. Fortes, "Partitioning and Mapping Algorithms into Fixed Size Systolic Arrays", *IEEE Trans. Computers*, vol. 35, no. 1, pp. 1-12, Jan1986.
32. N. Petkov, *Systolic Parallel Processing*. New York, NY: Elsevier Science Inc., 1992.
33. X. Wei, C.H. Yu, P. Zhang, Y. Chen, Y. Wang,
34. H. Hu, Y. Liang and J. Cong, "Automated Systolic Array Architecture Synthesis for High Throughput CNN Inference on FPGAs", in *Proc. IEEE Design Automation Conf.*, Austin, TX, 2017.
35. J.J. Zhang, K. Basu and S. Garg, "Fault- Tolerant Systolic Array Based Accelerators for Deep Neural Network Execution", *IEEE Design & Test*, vo. 36, no. 5, pp. 44-53, May 2018.



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details