Volume 3, No. 7, Nov-Dec 2012
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Low Power and Area Efficient 2C Multiply-Accumulate Unit and Its Application to a DTMAC Unit

Vimal Raj. V* and C.S Manikandababu
Department of Electronics and Communication Engineering (P.G)
Sri Ramakrishna Engineering College
Coimbatore, TamilNadu, India.
vimalrv02@gmail.com* and manikanda_babu@rediffmail.com

Abstract—we propose a low power and area efficient two-cycle multiply-accumulate (2C-MAC) architecture which supports 2's complement numbers, and includes accumulation guard bits and saturation circuitry. The first MAC pipeline stage contains only partial-product circuitry which is for generating partial product. And the second stage consists of sign-extension block, saturation unit and all other functionality. Proposed architecture does not need any additional cycles to generate the final result. It efficiently produces the addition of the accumulated value and the product in each cycle. And extend the new architecture to create a double throughput MAC, which can perform either multiply or multiply-accumulate operations.

Keywords—Arithmetic circuits, area efficient, low power, high speed adders, multiply-accumulate unit.

I. INTRODUCTION

The multiply-accumulate (MAC) unit is a general digital block which is commonly used in microprocessors and digital signal processors for much type of applications. For example, many filters, time division multiplexing, frequency-division multiplexing, and channel estimators that require FIR or FFT/IFFT computations that MAC unit can efficiently accelerate the operation. A general MAC architecture consists of a multiplier and an accumulator organized as in Fig. 1. Inputs are fed to the multiplier, and successive products are summed by the accumulator. Multipliers are typically consists of a partial-product unit (the PP unit) and a carry-propagate adder (the final adder) [1].

For increasing the performance of MAC, we can reduce the critical path delay. This is achieved by insertion of an extra pipeline register, either inside the partial product unit (PP unit) or between the PP unit and the carry propagation adder. This creates a three-cycle multiply and accumulate (3C-MAC) architecture (Fig.1), but increases the parameters in terms of delay, power and area. Much prior work focus on design techniques for reducing the delay of multiplier, either in the PP unit or the carry propagation adder. In the PP unit, the partial-product circuitry is implemented using the Baugh-Wooley algorithm [2]. Performing two different carry propagations in the same MAC circuit is much time consuming process because of carry propagation in the adder. Accumulation is handled by the final adder of the multiplier and only one carry-propagating stage is required. The problem is that this optimization only applies to one-cycle MAC's, where the long critical delay is a limiting factor in most applications.

If a pipe line register were to be inserted, the MAC output would no longer produce the correct result each cycle. In fact, to get the final result, we would add an extra, empty cycle after the final multiply-accumulate cycle of a loop. Furthermore, it is not obvious how guard bits can be accommodated in these designs. Guard bits are important for avoiding overflow when computing long sequences of multiply-accumulate operation.

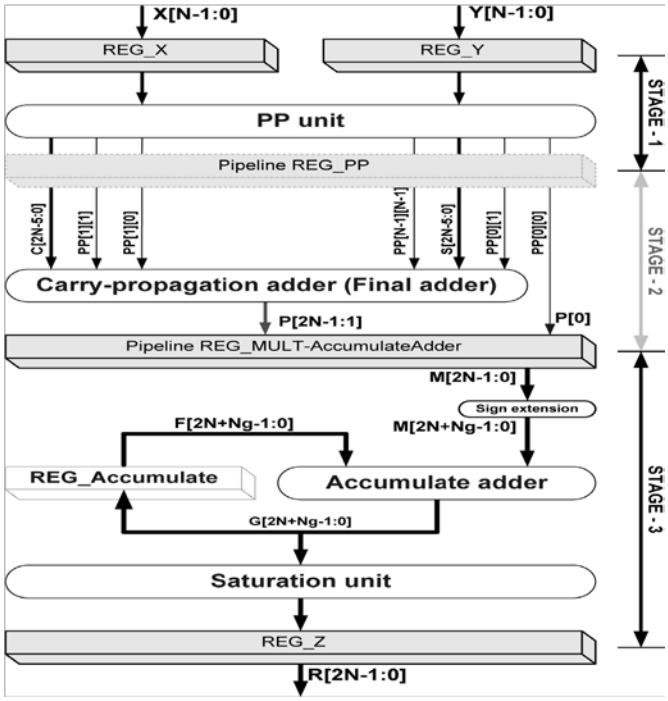

Figure.1. Block diagram of a general MAC architecture. Here, the register between the PP unit and the final adder is removed / included to obtain two/three cycle MAC architecture.

In general, two-cycle MAC architectures have a first (multiplication) stage that is significantly slower than the second (accumulation) stage. We propose a new two-cycle MAC architecture in which the second stage is somewhat slower, but the first stage is significantly faster, leading to a better delay balance between the two stages [3].

New architecture is the implementation of product sign extension, the sign-extension circuitry is located in the second stage, together with the accumulate adder and the saturation unit.

II. PROPOSED ARCHITECTURE

The proposed two's complement MAC architecture is shown in Fig. 2. Compared to the basic architecture in Fig. 1, the new design replaces the final adder in the first stage with a carry-skip adder in the second [4]. The multiplier is based on the Baugh – Wooley algorithm. First we compute the product of the two inputs. Then this result is sign extended to have the same size as the accumulate adder.

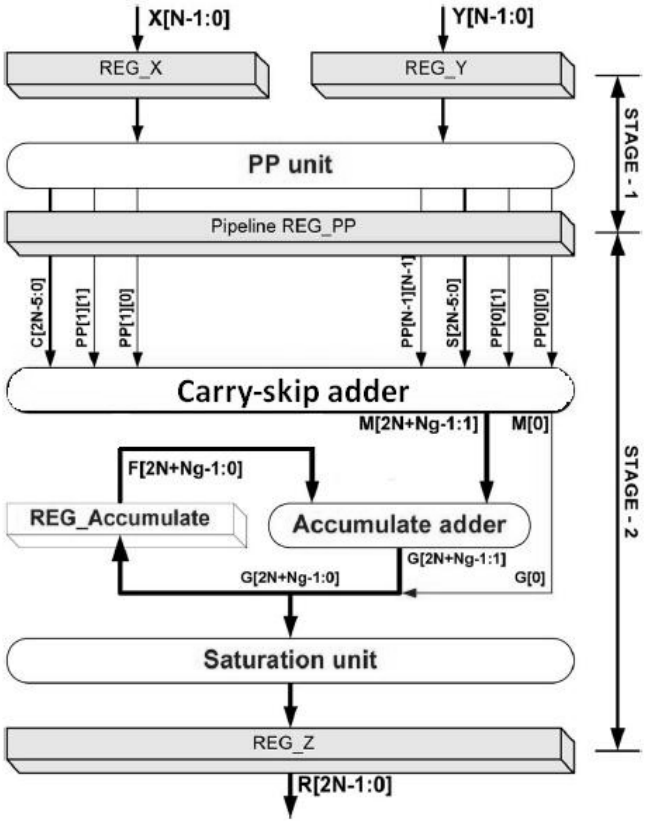

Figure. 2. Block diagram of the proposed MAC.

Finally, the sign extended product is added to the stored accumulated value. Proposed architecture do not needs any additional cycles to generate the final result. It efficiently produces the addition of the accumulated value and the product in each cycle.

The MAC architecture's critical path delay still depends on the PP unit, but the delays of the two stages are now similar. The second stage remains faster, especially for larger operand sizes, which allows the accumulate adder to accommodate more guard bits.

In our architecture, we use a carry-skip adder composed of full adders. This adder sums the accumulated value with the two outputs from the PP unit's output registers.

Our MAC architecture offers a number of advantages in terms of latency, area and power.

a. If we compare to a two-cycle MAC (Fig. 1), the proposed MAC architecture needs no final adder.
b. If we compare to a basic three-cycle MAC, our architecture allows us to remove not only the final adder but also one pipeline register level and the corresponding clock power without degrading speed.
c. Because our architecture is smaller, it uses shorter interconnects.

III. EVALUATION

We consider three architectures that share the same structure for the PP unit, the final adder and the accumulate adder:

a. MAC-2C: This conventional 2-cycle MAC has a critical path that goes through the PP unit and the final adder.
b. MAC-3C: This conventional 3-cycle MAC has a critical path that is located inside the PP unit.
c. MAC-NEW: Our proposed 2-cycle MAC exploits the fact that the delay of the accumulate adder is shorter than the delay of the PP unit, by at least an amount corresponding to the delay of a full-adder cell.

Concerning the comparison of the MAC critical path delay, we notice that MAC-2C and MAC-3C represent architectures that put an upper and a lower bound, respectively, on critical path delay.

A. Evaluation Methodology:

All PP units are based on the power-efficient Baugh–Wooley algorithm [5] for partial-product generation. The adder for accumulation is of simple conditional-sum adder type and has an extension of eight guarding bits. Finally, the final adder in is used to support fast addition of the PP outputs in the case of MAC-2C and MAC-3C. The accumulate adder is of conditional-sum type and has an extension of eight guard bits.

This allows the MAC unit to support loops of up to 256 iterations without requiring the output to be right-shifted to avoid overflow. A final adder supports fast addition of the PP unit outputs in the MAC-2C and MAC-3.

The VHDL codes were developed for MAC-3C and proposed MAC, using several 16-bit and 32-bits of the input data. We simulated the VHDL codes using MODELSIM Design suite and synthesized using XILINX. To avoid biasing the evaluation, we use a bottom-up synthesis method, meaning that the PP unit, the final adder and the accumulate adder are synthesized individually.

B. Results and Discussion:

Table I presents the detailed results of our evaluation. Since the delay is through the PP unit for all two designs and our proposed architecture uses pipeline registers at the bottom of the PP unit, MAC-NEW obviously can operate at the same speed as MAC-3C.

As far as power dissipation is concerned, because the final adder is replaced by the simple carry-skip adder, MAC-3C dissipates more power than MAC-NEW for the same operating frequency and timing constraint.

Table I: Evaluation results of 16-bit and 32-bit MAC-3C and MAC NEW in terms of delay, area and power.

Operand size	Architecture	Power (mW)	Area (gate count)	Delay (ns)
16-bit	MAC general - 2C	221	7886	78.457
	MAC general - 3C	380	13930	77.746
	Proposed MAC	210	7784	78..214
32-bit	MAC general - 2C	385	34885	143.386
	MAC general - 3C	1244	52439	154.516
	Proposed MAC	221	33356	147.363

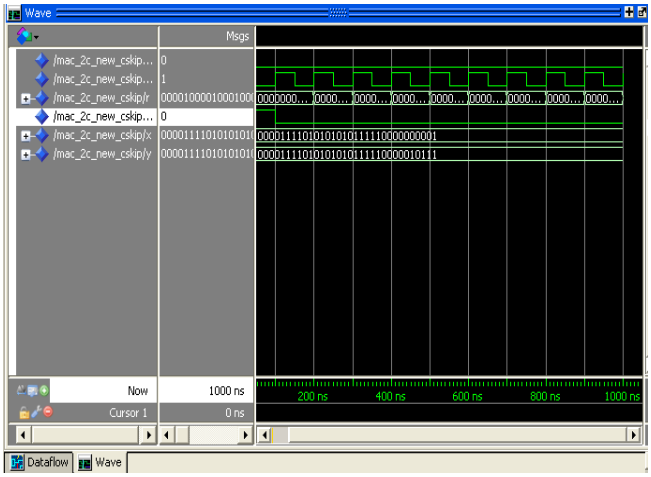

Figure. 3. Simulation result of 16-bit proposed MAC.

IV. EXTENSION TO A DTMAC UNIT

Adapting circuits to operate on the actual data precision of an application can save energy, as demonstrated in micro-processors and dedicated circuits. Many embedded applications are based on a 16-bit dynamic range, while embedded processors generally have a 32-bit data path. Thus, potentially 32-bit data path could accommodate the execution of two simultaneous16-bitoperations [6]. When the dynamic range of the data varies significantly a cross applications, run-time adaptation of the computational precision of a single circuit would be useful, rather than using several circuits that each has its own fixed operand size [7].

We refer to a MAC unit that can optionally switch between N-bitoperationsand2 × N/2- bit operations as a double-throughput MAC unit (DTMAC). A 32-bit instance of such a MAC unit could be implemented by tying together two separate, 16-bit MAC units. To support 32-bit operations the two16-bit multipliers must be combined into one 32-bit multiplier, which requires complex routing and is difficult to implement efficiently. In FPGA technology, which offers re-configurability [8] [9] that can support double -throughput multiply accumulate operations, but FPGAs are still inefficient in terms of speed and power compared to the ASIC solutions.

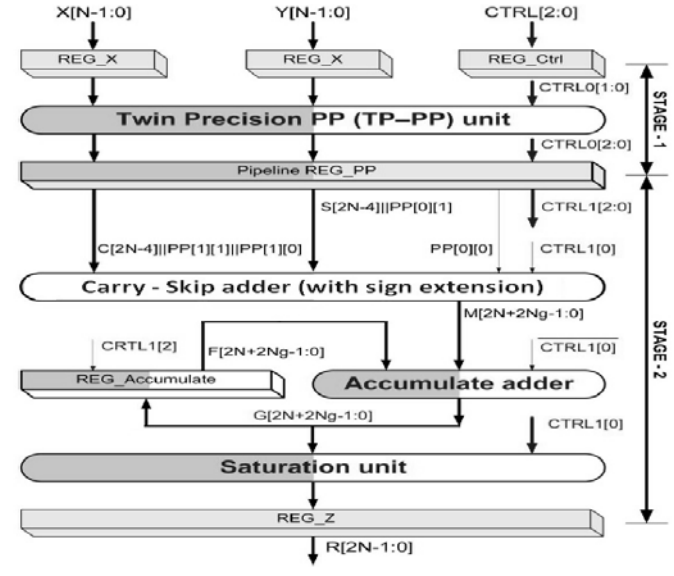

Figure. 4. Architecture of DTMAC Unit.

A critical feature of any double-throughput MAC unit is that it should support several operating modes, [10] without incurring any significant over heads on timing and power for thedefault1×N-bitmode.

While other schemes, such as Kuang and Wang's scheme [11], may be used, our two's complement DTMAC unit employs the Twin-Precision (TP) technique [12]. A twin-precision partial-product reduction tree generates the TP-PP unit's outputs, which in conventional schemes are fed to a final adder in order to obtain the final product. Instead, here we insert the proposed carry-skip adder that sums the TP-PP unit outputs and the accumulate adder output. The output of the carry-skip adder is fed to an accumulate adder that performs the carry propagation to produce the final result.

As for conventional MACs, the TP-PP unit dominates the critical path delay. The DTMAC unit actually has the same critical delay as that of a basic three-cycle 32-bit MAC architecture, in which a pipeline register is inserted between the PP unit and the final adder to shorten the critical path of the multiplication. The result is that, despite the operating-mode flexibility, the DTMAC unit has small area requirements, low power dissipation, and short critical path delay.

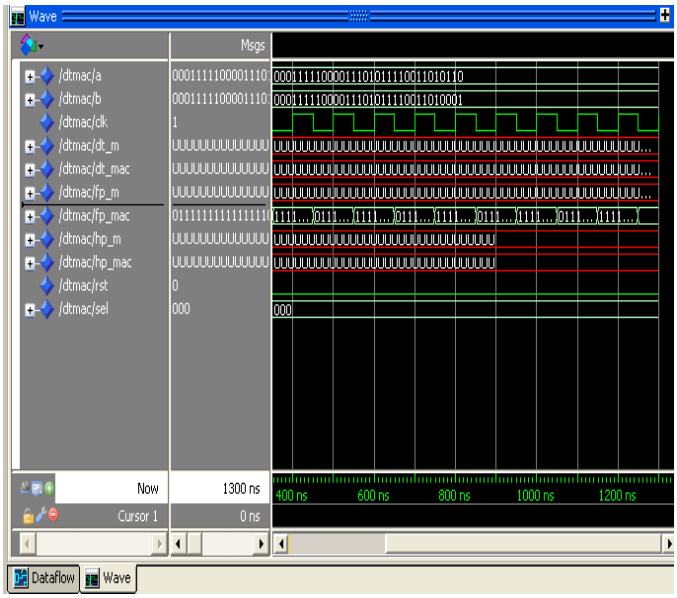

Figure. 5. Simulation result of DTMAC in Full precision mode.

The DTMAC unit supports six operating modes— three for multiply-accumulate operations and three for multiply operations—as determined by the value of the three-bit control signal (CTRL):

a. 000: Full-Precision 32-bit multiply-accumulate (FP_MAC).
b. 001: Half-Precision 1 × 16-bit multiply-accumulates (HP_MAC).
c. 011: Double-Throughput 2 × 16 bit multiply accumulate (DT_MAC).
d. 100: Full-Precision 32-bit multiplication (FP_MULT).
e. 101: Half-Precision 1 × 16-bit multiplication (HP_MULT).
f. 111: Double-Throughput 2 × 16-bit multiplication (DT_MULT).

In Fig. 4, CRTL, CTRL0, and CTRL1 denote the three-bit control signal, its one-cycle delayed version,

and its two-cycle delayed version, respectively. Moreover in CTRL[2:0], CTRL[2] is the left most of the three bits and is used to force the output of the accumulate register to zero during multiply operations.

V. CONCLUSION

Here describe a new low power, area efficient two's complement, two-cycle multiply-accumulate(MAC) architecture. Replacing the final adder of the multiplier by a carry-skip adder with a new sign extension technique makes our two-cycle MAC architecture area- and power-efficient than basic three-cycle MAC architecture and which shows that the new architecture, while only requiring two cycles for completing the MAC computation, still performs the MAC operation at the same top operating frequency as a 3-cycle MAC unit, at lower power dissipation. We use the new architecture to develop a versatile MAC unit that supports several different operating modes: three for multiply-accumulate operations and three for multiply operations.

VI. ACKNOWLEDGEMENT

The authors thank the Management and Principal of Sri Ramakrishna Engineering College, Coimbatore for providing excellent computing facilities and encouragement.

VII. REFERENCES

[1] O. L. MacSorley, "High-speed arithmetic in binary computers," Proc. Inst. Radio Eng. (IRE), vol. 49, pp. 67–91, Jan. 1961.

[2] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," IEEE Trans. Comput., vol. 45, no. 3, pp. 294–306, Mar. 1996.

[3] P. F. Stelling and V. G. Oklobdzija, "Implementing multiply-accumulate operation in multiplication time," in Proc. Int. Symp. Comput. Arithmetic (ARITH), July 1997, pp. 99–106.

[4] T. T. Hoang, M. Själander, and P. Larsson-Edefors, "High-speed, energy- efficient 2-cycle multiply- accumulate architecture," in Proc. IEEE Int. SOC Conf.(SOC), Sep. 2009, pp. 119–122.

[5] C. R. Baugh and B. A. Wooley, "A two'scomplement parallel array multiplication algorithm," IEEE Trans. Comput., vol. C-22, pp. 1045–1047, Dec 1973.

[6] M. Själander and P. Larsson-Edefors, "Multiplication acceleration through twin precision," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.17, pp. 1233–1246, Sep. 2009.

[7] H. Eriksson, P. Larsson-Edefors, M. Sheeran, M. Själander, D. Johanssonand M. Schölin, "Multiplier reduction tree with logarithmic logic depth and regular connectivity," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2006, pp. 4–8.

[8] R. K. Kolagotla, J. Fridman, B. C. Aldrich, M. M. Hoffman, W. C. Anderson, M. S. Allen, D. B. Witt, R. R. Dunton, and L. A. Booth, "High performance dual- MAC DSP architecture," IEEE Signal Process. Mag., vol. 19, no. 4, pp. 42–53, Jul. 2002.

[9] S. Hong and S.-S. Chin, "Reconfigurable embedded MAC core design for low-power coarse-grain FPGA," Electron. Lett., vol. 39, no. 7, pp. 606–608, Apr. 2003.

[10] T. T. Hoang, M. Själander, and P. Larsson-Edefors, "Double throughput multiply-accumulate unit for FlexCore processor enhancements," presented at the IEEE Int. Symp. Parallel Distrib. Process. (IPDPS), Reconfigurable Archit. Workshop (RAW), Rome, Italy, May 2009.

[11] S.-R. Kuang and J.-P. Wang, "Design of power- efficient configurable booth multiplier," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 568–580, Mar. 2010.

[12] M. Själander, H. Eriksson, and P. Larsson-Edefors, "An efficient twinprecision multiplier," in Proc. IEEE Int. Conf. Comput. Des. (ICCD), Oct. 2004, pp. 30–33.

