ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Design of Low Power MAC Using Modified Booth Recoder

G. Shireesh Kumar [1], Dr. K. Padma Priya [2]

P.G. Student(VLSI), Dept. of Electronics & Communication Engineering, University College of Engineering Kakinada, Kakinada, Andhra Pradesh, India[1]

Professor, Dept. of Electronics & Communication Engineering, University College of Engineering Kakinada, Kakinada, Andhra Pradesh, India[2]

ABSTRACT: In this paper, a low-power Multiplication-Accumulation Computation (MAC) unit using the radix-4 Booth algorithm is proposed. The proposed MAC is a low power, high speed and high throughput. MAC unit consists of multiplier, adder and accumulator. For high speed MAC unit, faster adders and multiplier circuits are required. We are using carry save adder (CSA) for faster operation and Radix-4 modified booth's algorithm used in the multiplier results in improvements in the speed of the operation. For high throughput operation, the result of the accumulator fed back to the adder. So, the final result can be a sum of the previous results. The MAC operation is performed on both signed and unsigned numbers of even and odd bit widths. When compared to conventional MAC the proposed MAC results in low power consumption and high speed. The proposed architecture was synthesized with 90 nm standard CMOS library. Synopsys tool is used for simulation and synthesis of MAC. The proposed MAC can be adapted to various fields where high performance is required, such as the signal processing areas.

KEYWORDS*:* Multiplication-Accumulation Computation (MAC), modified booth's algorithm, carry save adder (CSA).

I. INTRODUCTION

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. The performance and area remain to be two major design goals, power consumption has become a critical concern in today's VLSI system design. Multiplication is a fundamental operation in most signal processing algorithms [1]. Multipliers have large area, long latency and consume considerable power. Therefore, low-power multiplier design has been an important part in low-power VLSI system design. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels.

In this paper, a new design method is proposed for multiplier, multiple adders and fused MAC (Multiply and Accumulate) designs. The fast multiplication process consists of three steps: partial product generation, partial product reduction and final carry-propagating addition [2]. To reduce the number of partial products, recoding techniques have been widely used [2]. Multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined by the multiplier, in general [3]. For high-speed multiplication, the modified radix-4 Booth's algorithm (MBA) is commonly used. In general, a multiplier uses Booth's algorithm [3] and array of full adders (FAs), or Wallace tree instead of the array of FAs.

The speed of multipliers depends on partial product generation for this booth algorithm techniques used in multipliers to reduces the partial products by half. So, it improves in speed of operation. The MAC consists of multiplier, adder and accumulator. The multiplier used is Booth encoding multiplier. Carry save adder is used to add sum and carry. The output of the accumulator is fed back to the adder so, the final result of the MAC unit is sum of the previous result. One of the most advanced types of MAC for general-purpose digital signal processing has been proposed by Elguibaly [3]. It is an architecture in which accumulation has been combined with the carry save adder (CSA) tree that compresses partial products. Additionally, the proposed MAC unit outperforms the conventional ones in comparing the product of area and power consumption.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

II. LITERATURE WORK OF MAC

In this section, basic MAC operation is introduced. A multiplier in the MAC can be divided into three operational steps. The first is radix-2 Booth encoding in which a partial product is generated from the multiplicand (X) and the multiplier(Y) [3]. The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry. The last is the final addition in which the final multiplication result is produced by adding the sum and the carry. A general hardware architecture of this MAC is shown in Fig. 1. It executes the multiplication operation by multiplying the input multiplier X and the multiplicand$Y$. This is added to the previous multiplication result Z as the accumulation step. With $X\times Y+Z$ as the MAC, X and Y are two input data with k bits, and Z is the input datum with $2k$ bits. A $k\times k$-bit multiplier and a Carry Save Adder can be applied to perform this computation. The $k\times k$- bit multiplier is applied to perform the multiplication of X and Y, while the CSA carries out the addition of the multiplication result and a datum, Z, via a carry look ahead adder. If N-bit data are multiplied, the number of the generated partial products is proportional to$N$.

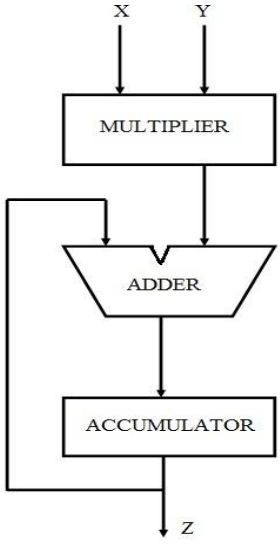
The architecture of a multiplier, which is the fastest, uses radix-2 Booth encoding that generates partial products and a Wallace tree based on CSA as the adder array to add the partial products. If radix-2 Booth encoding is used, the number of partial products, i.e., the inputs to the Wallace tree, is reduced to half, resulting in the decrease in CSA tree step. In addition, the signed multiplication based on 2's complement numbers is also possible. Due to these reasons, most current used multipliers adopt the Booth encoding.

The N-bit 2's complement binary number X can be expressed as [3]

$$X = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2}(x_i.2^i) \quad (1)$$

$$X = \sum_{i=0}^{\frac{N}{2}-1} d_i \ .4_i \quad (2)$$

$$d_i = -2x_{2i+1} + x_{2i} + x_{2i-1} \quad (3)$$



Output = X × Y + Z

Fig.1. Basic MAC architecture

If (2) is used, multiplication can be expressed as

$$X \times .Y = \sum_{i=0}^{\frac{N}{2}-1} di2^{2i}\,Y \quad (4)$$

If these equations are used, the multiplication–accumulation results can be expressed as

$$P=X\times Y+Z = \sum_{i=0}^{\frac{N}{2}-1} di2^{2i}Y + \sum_{j=0}^{2N-1} 2^i Z_i \quad (5)$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

III. MODIFIED BOOTH ALGORITHM

The modified-Booth algorithm is extensively used for high-speed multiplier circuits. Once, when array multipliers were used, the reduced number of generated partial products significantly improved multiplier performance. The Modified Booth Multiplier was proposed by O. L. Macsorley in 1961. The recoding method is widely used to generate the partial products for implementation of large parallel multipliers, which adopts the parallel encoding scheme [3]. One of the solutions of realizing high speed multipliers is to enhance parallelism which helps to decrease the number of subsequent stages. The original version of Booth algorithm (Radix-2) had two drawbacks:

1. The number of add subtract operations and the number of shift operations becomes variable and becomes inconvenient in designing parallel multipliers.
2. The algorithm becomes inefficient when there are isolated 1's.

These problems can be overcome by Modified Booth algorithm (MBA). In MBA process three bits at a time are recorded. Recoding the multiplier in higher radix is a powerful way to speed up standard Booth multiplication algorithm.Radix4 Booth algorithm is used. The Modified Booth algorithm is represented in the form :

$$y_k^{MB} = -2y_{2k+1} + y_{2k} + y_{2k+1}$$

Table 1: Modified Booth algorithm recoded table

Binary Inputs			Recoded Values	Operation to be Performed
y2k+1	y2k	y2k-1		
0	0	0	0	0*multiplicand
0	0	1	+1	+1*multiplicand
0	1	0	+1	+1*multiplicand
0	1	1	+2	+2*multiplicand
1	0	0	-2	-2*multiplicand
1	0	1	-1	-1*multiplicand
1	1	0	-1	-1*multiplicand
1	1	1	0	0*multiplicand

To optimize the MAC operator design of the adder unit is fused with MB encoding unit to form a single unit data path. This is done by direct recoding of the sum Y=A+B to its MB form. This Fused Multiply-Accumulate unit has only one adder at the end which results in a significant area reduction. The new techniques have been introduced to implement this multiply-accumulate unit. In all techniques separate designs are implemented for an even and odd number of signed and unsigned bits.

(1) Technique I : SMB1

This technique uses two full adders to implement the design for odd and even width of bits stream. For the even number of bits two FAs are used as FA, a conventional full adder and FA* whose output value is given as:

$$FA^* = -2co + s = -q - q + ci .$$

For odd bit-width an additional FA** is used at the end whose output value is

$$FA^{**} = -2co + s = -p - q + ci .$$

(2) Technique II : SMB2

In this technique , for even bit-width a conventional full adder along with two half adders with the output value

$$HA^* = -2c + s = -p - q$$

HA* used whereas for odd width of bits an additional full adder as FA** has been used at the end.

(3) Technique III : SMB3

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Here in this proposed scheme for even bit-width of input numbers a conventional full adder along with three different half adders has been used where the half adders are conventional HA,HA* and HA**.HA** output value is

$$HA^{**} = 2c - s = -p + q.$$

The odd bit-width input uses a conventional FA, HA and HA* along with and additional FA** at the end of the recoding scheme.

IV. PROPOSED LOW POWER MAC

The proposed MAC architecture is shown in Fig. 2. With *X*, *Y* and *Z* as the three input data for the proposed architecture [4], it is able to perform the multiplication of a 8-bit *X* and a 8-bit *Y* , and to carry out the addition of the multiplication result and a 16 -bit *Z* datum. The radix-4 Booth algorithm is applied to perform the multiplication to design a low-complexity and high-speed architecture. The carry-save addition operation is proposed to reduce its hardware complexity. The functional blocks of proposed low-power MAC unit are illustrated as follows:

A. BOOTH RECODER

The Booth multiplier is also known as recoded booth multiplier, in which the multiplicand is kept as it is and the multiplier is recoded as a recoded multiplier and then the multiplication is done with multiplicand and recoded multiplier.

To reduce the number of partial products in the multiplier. A radix-4 multiplier produces N/2 partial products. Each radix-4 multiplier produces N/2 partial products. Each partial product is 0, Y, 2Y, or 3Y,depending on a pair of bits of X. Computing 2Y is a simple shift, but 3Y is a hard multiple requiring a slow carry-propagate addition of Y + 2Y before partial product generation begins. Higher-radix Booth encoding is possible, but generating the other hard multiples appears not to be worthwhile for multipliers of fewer than 32 bits. The figure 2 shows the proposed architecture of MAC.

B. CARRY SAVE ADDER

Carry -save addition unit is required to perform carry-save addition on partial products and carry-in bits produced by the Booth encoder. Carry -save addition is normally implemented using a full adder and a half adder. However, half adder is only able to take two bits in the same bit position and add them up to produce two bits in different bit positions so that the number of bits can't be reduced.

C. ACCUMULATOR UNIT

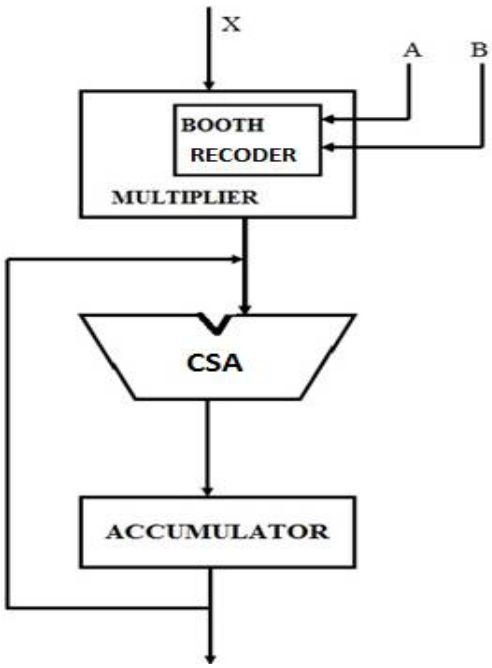
Multiply and accumulate is a very common basic-level operation seen in many DSP designs/algorithms. Two numbers are multiplied together, and added into an accumulator register. The accumulator accumulate the result from carry save adder and this accumulate result can be feedback to the adder for high throughput. So, the final output of MAC is the sum of the previous products.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016



Output= X×Y+Z

Fig.2. Low power MAC architecture

In this Low power MAC multiplier, we are using one of the modified booth technique to improve the parallelism of the multiplier. The modified booth recoder divide the multiplicand into two parts and encode the data. Therefore the partial products can be reduced and it improves the speed of operation.

S-MB technique:

IN SMB recording technique considering by taking two inputs i.e. X and Y. In these inputs Y is sum of two inputs A and B and we recode the sum A and B, the recoded sum is the input to the multiplier. MB encoding digit includes three bits and most significant bit represents sign. In this method we are using conventional and bit level signed Half adder and Full adder in order to design and explore three new alternative schemes in SMB recoding technique. The schemes can easily applied either signed or unsigned[5] which are consisting of even and odd number of bits.

$$Y=A+B= y_k \cdot 2^{2k} + \sum_{j=0}^{k-1} 2^{2j} \cdot y_j^{MB} \qquad (6)$$

V. RESULTS

Simulation waveform:

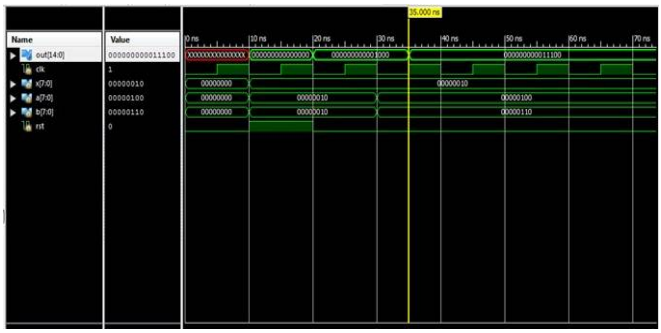


Fig .3.Low power MAC

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

When inputs x=2,a=2 and b=2.then a and b are added and corresponding result multiplied with x and it has given the result is 8.When any one of the input changes then the corresponding result will be added to previous result.

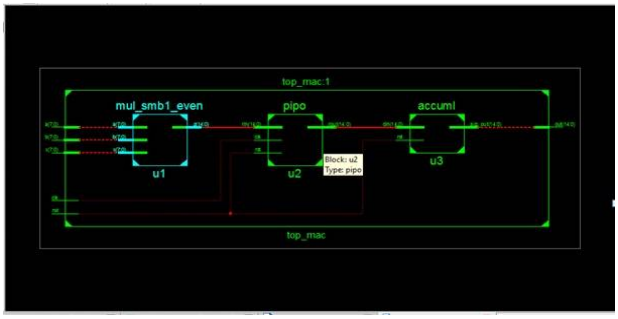


Fig.4. Schematic view of MAC

Table 2 Area and power measurements of FAM designs

	Area(nm)	**Power(uw)**
S-MB1	3203.6638	218.32
MAC	4670.8	56.12
Proposed MAC	5290.5	38.07

Compared to the conventional MAC the proposed MAC reduces the power consumption by 32.14%

VI. CONCLUSION

This paper deals with optimization of MAC operator by using direct recoding. The proposed method is compared with the existing methods and it gives the improvements in the proposed methods. The same S-MB schemes are used in MAC unit and it improved in low power but area increases. A hardware complexity of the conventional MAC architectures is looked at in this work to propose a MAC unit with high computational performance and low power consumption. The limitation is area increases for this new techniques has to be proposed.A new MAC architecture to execute the multiplication-accumulation operation, which is the key operation, for digital signal processing and multimedia information pro-cessing efficiently the overall MAC performance has been improved almost twice as much as conventional MAC in power consumption.

REFERENCES

[1] Z. Huang, "High-Level Optimization Techniques for Low-Power Multiplier Design," Ph.D., University of California, Department of Computer Science, Los Angeles, CA, 2003.
[2] O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bitMAC design using fast 5: 3 compressor cells," *J. VLSI Signal Process. Syst.*, vol. 31, no. 2, pp. 77–89, Jun. 2002.
[3] Y.-H. Seo and D.-W. Kim, "A new VLSI architecture of parallel multiplier–accumulator based on Radix-2 modified Booth algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp. 201–208, Feb. 2010.
[4] L.-H. Chen, O. T.-C. Chen, T.-Y.Wang, and Y.-C. Ma, "A multiplication-accumulation computation unit with optimized compressors and minimized switching activities," in *Proc. IEEE Int, Symp. Circuits and Syst.*, Kobe, Japan, 2005, vol. 6, pp. 6118–6121.
[5] Kosatas Tsoumams , Sotiris Xydis , Nikos Moschopoulos , Kiamal Pekmestzi, "An Optimized Modified Booth Recoder For Efficientdesign Of The Add-Multiply Operator".IEEE Transactions On Circuits And Systems:Regular Papers, Vol.61, No. 4, April 2014.
[6] Z. Huang andM. D. Ercegovac, "High-performance low-power left-toright array multiplier design," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 272–283, Mar. 2005.
[7] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, "Estimation of signal transition activity in FIR filters implemented by a MAC architecture," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 1, pp. 164–169, Jan. 2000.
[8]A.S.Prabhu V.Elakya "Design of Modified Low Power Booth Multiplier" Department of ECE, Bannari Amman Institute of Technology, Sathyamangalam, Tamil |Nadu, India.
[9] Wen-Chang Yeh and Chein-Wei Jen, "High-Speed Booth Encoded Parallel Multiplier Design" IEEE TRANSACTIONS ON COMPUTERS, VOL. 49, NO. 7, JULY 2000.
[10]O. L. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67–91, Jan. 1961.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

[11]R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-ofproducts," in Proc. Asilomar Conf. Signals, Syst. Comput., Pacific Grove, Washington, DC, 2003, pp. 867–872.
[12]B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. Oxford: Oxford Univ. Press, 2000.
[13] Shaik.Kalisha Baba, D.Rajaramesh " Design and Implementation of Advanced Modified Booth Encoding Multiplier " International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726 www.ijesi.org Volume 2 Issue 8 ǁ August. 2013 ǁ PP.60-68
[14] D.J. Magenheimer, L. Peters, K.W. Pettis, and D. Zuras, "IntegerMultiplication and Division on the HP Precision Architecture,"IEEE Trans. Computers,vol. 37, no. 8, pp. 980-990, Aug. 1988.
[15] A.D. Booth, "A Signed Binary Multiplication Technique,"Quarterly J. Mechanical Applications of Math.,vol. IV, no. 2, pp. 236-240,1951.
[16] R. Bernstein, "Multiplication by Integer Constants,"Software—Practice and Experience,vol. 16, no. 7, pp. 641-652, July 1986.
[17] N. Boullis and A. Tisserand, "Some Optimizations of Hardware Multiplication by Constant Matrices,"Proc. 16th IEEE Symp.Computer Arithmetic (ARITH 16),J. -C. Bajard and M. Schulte, eds.,pp. 20-27, June 2003.