ROUNDING BASED APPROXIMATE MULTIPLIER FOR HIGH SPEED DIGITAL SIGNAL PROCESSING WITH FIR FILTER

[1]K.Ashokkumar ,[2]C.Bhavani Vishnu Priya,
[1]Assistant professor , [2]PG Scholar,Dept. Of E.C.E
[1,2] PBR Visvodaya Institute of Technology science, Kavali, Nellore District,A.P

Abstract

In this paper, we propose an approximate multiplier that is high speed yet energy efficient. The approach is to round the operands to the nearest exponent of two. This way the computational intensive part of the multiplication is omitted improving speed and energy consumption at the price of a small error. The proposed approach is applicable to both signed and unsigned multiplications. We propose three hardware implementations of the approximate multiplier that includes one for the unsigned and two for the signed operations. The efficiency of the proposed multiplier is evaluated by comparing its performance with those of some approximate and accurate multipliers using different design parameters. In addition, the efficacy of the proposed approximate multiplier is studied in two image processing applications, i.e., image sharpening and smoothing.

I.INTRODUCTION

Multipliers are one of the most significant blocks in computer arithmetic and are generally used in different digital signal processors. There is growing demands for high speed multipliers in different applications of computing systems, such as computer graphics, scientific calculation, and image processing and so on. Speed of multiplier determines how fast the processors will run and designers are now more focused on high speed with low power consumption. The multiplier architecture consists of a partial product generation stage, partial product reduction stage and the final addition stage. The partial product reduction stage is responsible for a significant portion of the total multiplication delay, power and area. Therefore in order to accumulate partial products, compressors usually implement this stage because they contribute to the reduction of the partial products and also contribute to reduce the critical path which is important to maintain the circuit's performance.

This is accomplished by the use of 3-2, 4-2, 5-2 compressor structures. A 3-2 compressor circuit is also known as full adder cell. As these compressors are used repeatedly in larger systems, improved design will contribute a lot towards overall system performance. The internal structure of compressors is basically composed of XOR-XNOR gates and multiplexers. The XOR-XNOR circuits are also building blocks in various circuits like arithmetic circuits, multipliers, compressors, parity checkers, etc. Optimized design of these XOR-XNOR gates can improve the performance of multiplier circuit. In present work, a new XOR-XNOR module has been proposed and 4-2 compressor has been implemented using this module. Use proposed circuit in partial product accumulation reduces transistor count as well as power consumption.

Addition and multiplication are widely used operations in computer arithmetic; for addition full-adder cells have been extensively analysed for approximate computing Liang et al. has compared these adders and proposed several new metrics for evaluating approximate and probabilistic adders with respect to unified figures of merit for design assessment for inexact computing applications. For each input to a circuit, the error distance (ED) is defined as the arithmetic distance between an erroneous output and the correct one. The mean error distance (MED) and normalized error distance (NED) are proposed by considering the averaging effect of multiple inputs and the normalization of multiple-bit adders. The NED is nearly invariant with the size of an implementation and is therefore useful in the reliability assessment of a specific design. The tradeoff between precision and power has also been quantitatively evaluated.

However, the design of approximate multipliers has received less attention. Multiplication can be thought as the repeated sum of partial products; however, the straightforward application of approximate adders when designing an approximate multiplier is not viable, because it would be very inefficient in terms of precision, hardware complexity and other performance metrics. Several approximate multipliers have been proposed. Most of these designs use a truncated multiplication method; they estimate the least significant columns of the partial products as a constant. An imprecise array multiplier is used for neural network applications by omitting some of the least significant bits in the partial products (and thus removing some adders in the array). A truncated multiplier with a correction constant is proposed.

II.EXISTING SYSTEM

A.BOOTH MULTIPLIER

Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on crystallography at Birkbeck College in Bloomsbury, London.[1] Booth's algorithm is of interest in the study of computer architecture. Booth's algorithm examines adjacent pairs of bits of the 'N'-bit multiplier Y in signed two's complement representation, including an implicit bit below the least significant bit, $y_{-1} = 0$. For each bit y_i, for i running from 0 to $N - 1$, the bits y_i and y_{i-1} are considered. Where these two bits are equal, the product accumulator P is left unchanged. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times 2^i is added to P; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2^i is subtracted from P. The final value of P is the signed product.

B. IMPLEMENTATION

Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values A and S to a product P, then performing a rightward arithmetic shift on P. Let **m** and **r** be the multiplicand and multiplier, respectively and let x and y represent the number of bits in **m** and **r**.

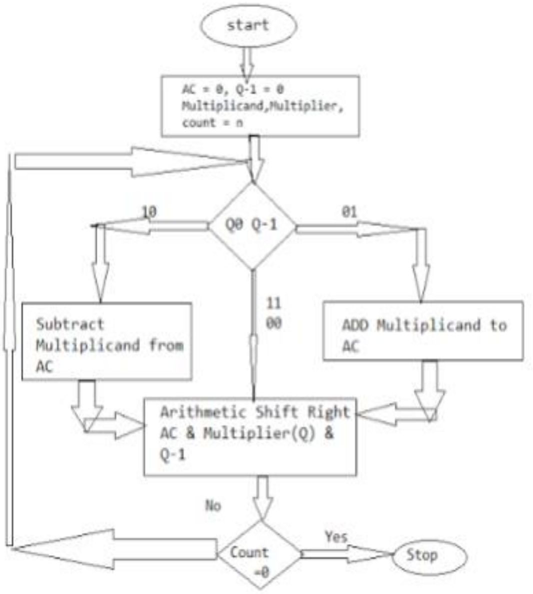


Fig 1: Flow chart representation of booth algorithm

III.PROPOSED SYSTEM:

In this paper, we focus on proposing a high-speed low power/energy yet approximate multiplier appropriate for error resilient DSP applications. The proposed approximate multiplier, which is also area efficient, is constructed by modifying the conventional multiplication approach at the algorithm level assuming rounded input values. We call this rounding-based approximate (RoBA) multiplier.

The proposed multiplication approach is applicable to both signed and unsigned multiplications for which three optimized architectures are presented. The efficiencies of these structures are assessed by comparing the delays, power and energy consumptions, energy-delay products (EDPs), and areas with those of some approximate and accurate (exact) multipliers. The contributions of this paper can be summarized as follows: 1) presenting a new scheme for RoBA multiplication by modifying the conventional multiplication approach; 2) describing three hardware architectures of the proposed approximate multiplication scheme for sign and unsigned operations.

A. MULTIPLICATION ALGORITHM OF ROBA MULTIPLIER

The main idea behind the proposed approximate multiplier is to make use of the ease of operation when the numbers are two to the power n (2n). To elaborate on the operation of the approximate multiplier, first, let us denote the rounded numbers of the input of A and B by Ar and Br, respectively. The multiplication of A by B may be rewritten as

$$A \times B = (A_r - A) \times (B_r - B) + A_r \times B + B_r \times A - A_r \times B_r.$$

The key observation is that the multiplications of Ar × Br, Ar ×B, and Br ×A may be implemented just by the shift operation. The hardware implementation of (Ar − A) × (Br − B), however, is rather complex. The weight of this term in the final result, which depends on differences of the exact numbers from their rounded ones, is typically small. Hence, we propose to omit this part, helping simplify the multiplication operation. Hence, to perform he multiplication process, the following expression is used:

$$A \times B \cong A_r \times B + B_r \times A - A_r \times B_r$$

In this approach, the nearest values for A and B in the form of 2n should be determined. When the value of A (or B) is equal to the 3 × 2p−2 (where p is an arbitrary positive integer larger than one), it has two nearest values in the form of 2n with equal absolute differences that are 2p and 2p−1. While both values lead to the same effect on the accuracy of the proposed multiplier, selecting the larger one (except for the case of p = 2) leads to a smaller hardware implementation for determining the nearest rounded value, and hence, it is considered in this paper.

It originates from the fact that the numbers in the form of 3 × 2p−2 are considered as do not care in both rounding up and down simplifying the process, and smaller logic expressions may be achieved if they are used in the rounding up. The only exception is for three, which in this case, two is considered as its nearest value in the proposed approximate multiplier.

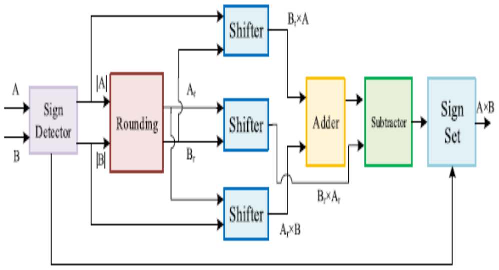


Fig.2.Block diagram for the hardware implementation of the ROBA multiplier.

It should be noted that contrary to the previous work where the approximate result is smaller than the exact result, the final result calculated by the RoBA multiplier may be either larger or smaller than the exact result depending on the magnitudes of Ar and Br compared with those of A and B, respectively. Note that if one of the operands (say A) is smaller than its corresponding rounded value while the other operand (say B) is larger than its corresponding rounded value, then the approximate result will be larger than the exact result. This is due to the fact that, in this case, the multiplication result of (Ar − A) × (Br − B) will be negative.

Since the difference between them is precisely this product, the approximate result becomes larger than the exact one. Similarly, if both A and B are larger (or) both are smaller than Ar and Br, then the approximate result will be smaller than the exact result. Finally, it should be noted the advantage of the proposed RoBA multiplier exists only for positive inputs because in the two's complement representation, the rounded values of negative inputs are not in the form of 2n.

B. FIR Filter using ROBA Multiplier

Finite Impulse Response (FIR) filters are widely used in Digital Signal Processing (DSP) applications due to their stability and linear-phase property. Finite Impulse Response filters are the most important element in signal processing and communication. Area optimization and speed are the key requirements of Finite Impulse Response filters. Finite Impulse Response filter involves multiplications, additions and shifting operations. As the multiplier is the slowest element in the system, it will affect the performance of the FIR filter.

In today scenario, low power consumption and less area are the most important parameter for the fabrication of DSP systems and high performance systems. Nowadays, many finite impulse response (FIR) filter designs aimed at either low area or high speed or reduced power consumption are developed. With the increase in area, hardware cost of these FIR filters are increasing. This leads to design a low area FIR filter with the advantage of moderate speed performance. The implementation of an FIR filter requires three basic building blocks. They are Multiplication, Addition and Signal delay. Multipliers consume the most amount of area in a FIR filter design. As the multiplier is the slowest element in the system, it will affect the performance of the FIR filter. Here, Conventional multipliers are replaced by a modified roba multiplier.

As the multiplier is the slowest element in the system, it will affect the performance of the FIR filter. So, a ROBA multiplier is suggested since it reduces area and it is faster than other conventional multipliers. The proposed low area-cost FIR filter using a ROBA multiplier is shown in Figure below.

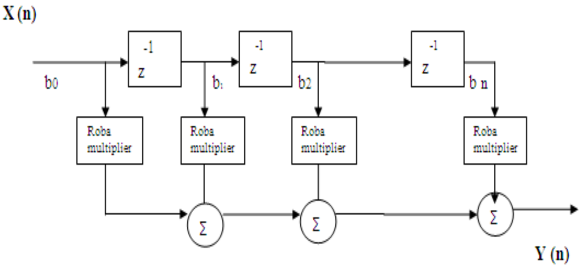


Fig 3. FIR filter using ROBA multiplier

VI..SIMULATION RESULTS:

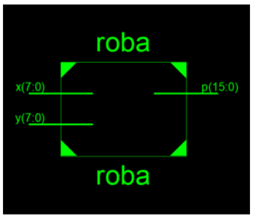


Fig 4:RTL schematic of ROBA Multiplier

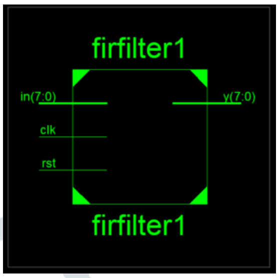
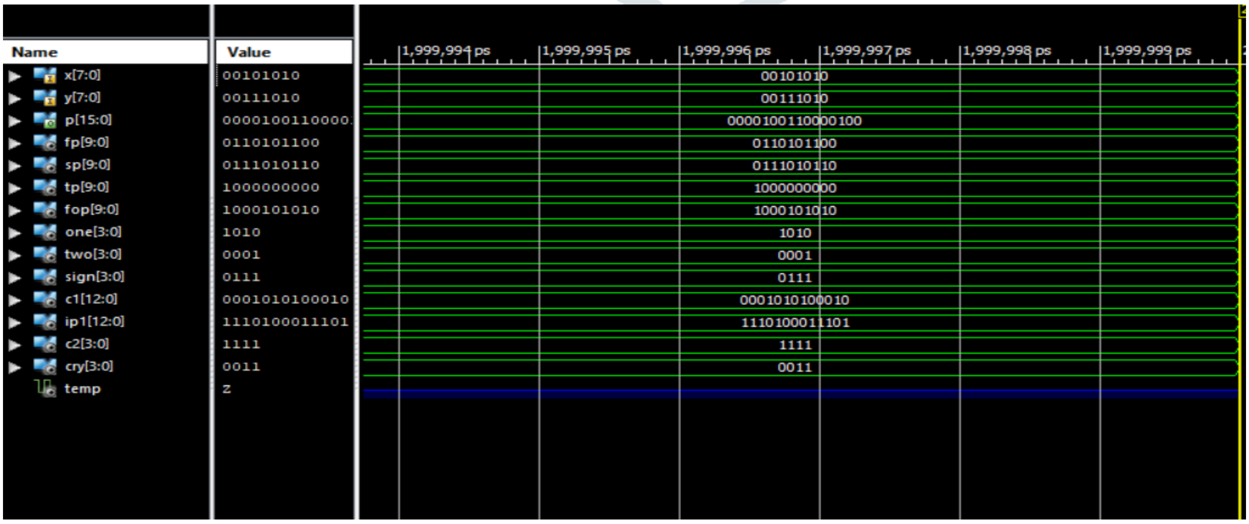
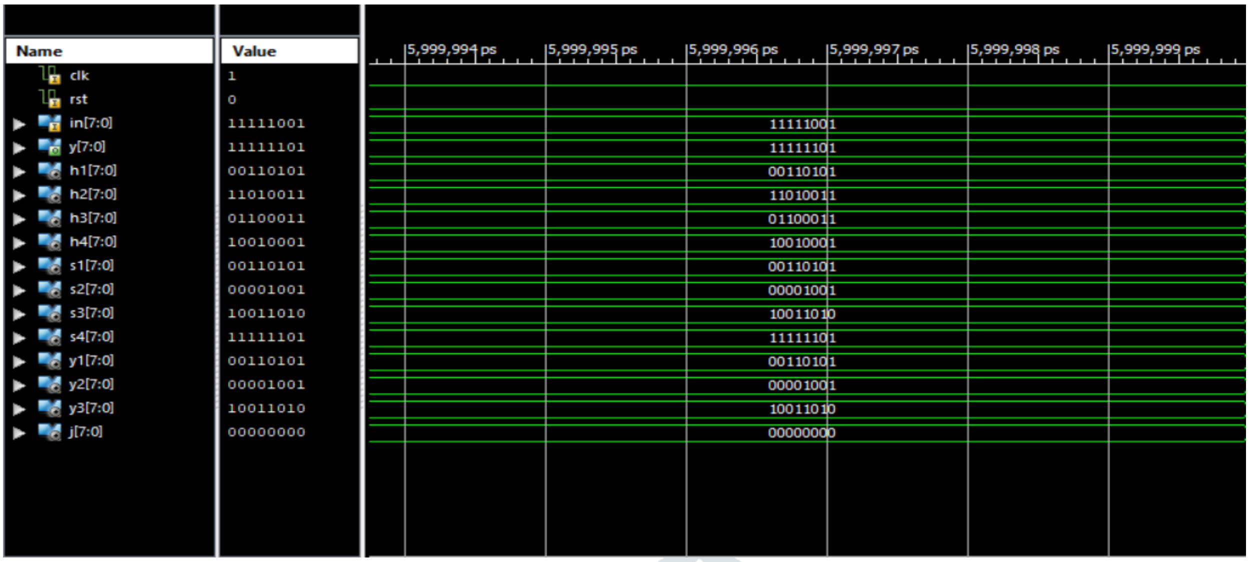


Fig 5:RTL schematic of ROBA Multiplier



Name	Value
x[7:0]	00101010
y[7:0]	00111010
p[15:0]	0000100110000
fp[9:0]	0110101100
sp[9:0]	0111010110
tp[9:0]	1000000000
fop[9:0]	1000101010
one[3:0]	1010
two[3:0]	0001
sign[3:0]	0111
c1[12:0]	0001010100010
ip1[12:0]	1110100011101
c2[3:0]	1111
cry[3:0]	0011
temp	z

Fig 6:Simulation Results of ROBA multiplier



Name	Value
clk	1
rst	0
in[7:0]	11111001
y[7:0]	11111101
h1[7:0]	00110101
h2[7:0]	11010011
h3[7:0]	01100011
h4[7:0]	10010001
s1[7:0]	00110101
s2[7:0]	00001001
s3[7:0]	10011010
s4[7:0]	11111101
y1[7:0]	00110101
y2[7:0]	00001001
y3[7:0]	10011010
j[7:0]	00000000

Fig 7:Simulation Results of FIR Filter using ROBA Multiplier

VII.CONCLUSION

In this paper, we proposed a high-speed yet energy efficient approximate multiplier called RoBA multiplier. The proposed multiplier, which had high accuracy, was based on rounding of the inputs in the form of 2n. In this way, the computational intensive part of the multiplication was omitted improving speed and energy consumption at the price of a small error. The proposed approach was applicable to both signed and unsigned multiplications. Three hardware implementations of the approximate multiplier including one for the unsigned and two for the signed operations were discussed. The efficiencies of the proposed multipliers were evaluated by comparing them with those of some accurate and approximate multipliers using different design parameters. The results revealed that, in most (all) cases, the RoBA multiplier architectures outperformed the corresponding approximate (exact) multipliers.

This work is extended to implement fir filter using roba multiplier. It offers great advantage in the reduction of delay.

REFERENCES

[1] M. Alioto, “Ultra-low power VLSI circuit design demystified and explained: A tutorial,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2012.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-power digital signal processing using approximate adders,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[3] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[4] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, “MACACO: Modeling and analysis of circuits for approximate computing,” in Proc. Int. Conf. Comput.-Aided Design, Nov. 2011, pp. 667–673.

[5] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, “New approximate multiplier for low power digital signal processing,” in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADS), Oct. 2013, pp. 25–30.

[6] P. Kulkarni, P. Gupta, and M. Ercegovac, “Trading accuracy for power with an underdesigned multiplier architecture,” in Proc. 24th Int. Conf. VLSI Design, Jan. 2011, pp. 346–351.

[7] D. R. Kelly, B. J. Phillips, and S. Al-Sarawi, “Approximate signed binary integer multipliers for arithmetic data value speculation,” in Proc. Conf. Design Archit. Signal Image Process., 2009, pp. 97–104.

[8] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, “Low-power high-speed multiplier for error-tolerant application,” in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.

[9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, “Design and analysis of approximate

compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[10] K. Bhardwaj and P. S. Mane, "ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip," in Proc. 8th Int. Workshop Reconfigurable Commun.-Centric Syst.-Chip, 2013, pp. 1–6.

[11] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power-and area-efficient approximate wallace tree multiplier for error-resilient systems," in Proc. 15th Int. Symp. Quality Electron. Design (ISQED), 2014, pp. 263–269.

[12] J. N. Mitchell, "Computer multiplication and division using binary logarithms," IRE Trans. Electron. Comput., vol. EC-11, no. 4, pp. 512–517, Aug. 1962.

[13] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," IEEE Trans. Comput., vol. 55, no. 12, pp. 1523–1535, Dec. 2006.

[14] Nangate 45nm Open Cell Library, accessed on 2010. [Online]. Available: http://www.nangate.com/

[15] H. R. Myler and A. R. Weeks, The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.

[16] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[17] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 418–425.

[18] C.-H. Lin and I.-C.Lin, "High accuracy approximate multiplier with error correction," in Proc. 31st Int. Conf. Comput. Design (ICCD), 2013, pp. 33–38.

[19] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proc. 49th Design Autom. Conf. (DAC), Jun. 2012, pp. 820–825.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[21] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 62, no. 9, pp. 1760–1771, Sep. 2013.

Author's Profile:

Mr.K. Ashok Kumar M.Tech, MISTE is currently working as an Assistant Professor in Electronics and communication Engineering (ECE) Department, PBR Visvodaya Institute of Technology science, Kavali, Nellore District, A.P affliated to the Jawaharlal Nehru technological university Ananthapur.

C. Bhavani Vishnu Priya is currently pursuing M.Tech degree in VLSI in Electronics and Communication Engineering from PBR Visvodaya Institute of Technology science, Kavali, Nellore District, AP, India.She received her B.Tech in Electronics and Communication Engineering from Visvodaya Engineering College, Kavali, Nellore District,A.P.