International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 15
May 2018

Design A Approximate Parallel Multiplier For Medical Applications

[1] K KRANTH KIRAN, [2] K VENKAT RAO, [3] K. ANKA SIVAPRASAD
[1]M.Tech- student, Dept. of ECE, Chintalapudi engineering college, Ponnur, Guntur DT.
[2]Associate Professor, Dept. of ECE, Chintalapudi engineering college, Ponnur, Guntur DT.
[3] HOD, Dept. of ECE, Chintalapudi engineering college, Ponnur, Guntur DT.

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.

Index Terms: Accuracy, approximate computing, energy efficient, error analysis, high speed, multiplier.

I. INTRODUCTION

Energy minimization is one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is highly desired to achieve this minimization with minimal performance (speed) penalty. Digital signal processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems.
Therefore, improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors.

Many of the DSP cores implement image and video processing algorithms where final outputs are either images or videos prepared for human consumptions. This fact enables us to use approximations for improving the speed/energy efficiency. This originates from the limited perceptual abilities of human beings in observing an image or a video. In addition to the image and video processing applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the system. Being able to use the approximate computing provides the designer with the ability of making tradeoffs between the accuracy and the speed as well as power/energy consumption.

Applying the approximation to the arithmetic units can be performed at different design abstraction levels including circuit, logic, and architecture levels, as well as algorithm and software

layers. The approximation may be performed using different techniques such as allowing some timing violations (e.g., voltage over scaling or over clocking) and function approximation methods (e.g., modifying the Boolean function of a circuit) or a combination of them. In the category of function approximation methods, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested. 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.

II. EXISTED SYSTEM

This constitutes the core engine of our Electrical Capacitance Tomography (ECT) system since it performs the image reconstruction task. Figure 1 shows the detailed VLSI architecture. It is modular and divided into four main modules: a parallel processing module, a data variable input/output memories module to store the data and image matrices a sequencer and memory controller module to schedule parallel memory accesses between the memory module and the parallel processing unit, and a post processing module which performs image normalization and quantization. The Design of the FPGA module considered two main issues: the execution speed and hardware scalablity. While the image reconstruction process needs to be achieved in real-time, the hardware should accommodate the various sizes of matrices to be processed.

The parallel processing module is a parallel like architecture which is composed of several similar adder/multiplier processing units. Each of these units is scalable and consists of three pipelined stages: decomposition stage, basic unit operating stage, and composition stage. Figure 2 shows these stages at both matrix and bit levels.

Decomposition Stage: The matrix decomposition consists to divide the two matrices to multiply into regular square blocks (qxq) which are then simultaneously processed by the parallel processing units using bit decomposition techniques. In case one of the two matrices has a number of columns which are not

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 15
May 2018

multiply of q, then this matrix is filled with all zeros columns to generate a new matrix where both rows and columns are multiply of q. This does not engender further hardware cost or computation delay since only PEs with non null input operands are provided in the architecture.

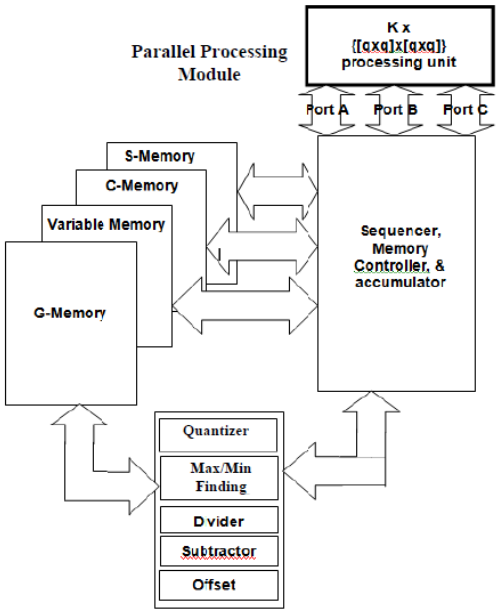

Fig1. Image reconstruction implementation block diagram

The parallel processing module then proceeds by executing simultaneously several (qxq) square matrix multiplications to obtain the values of several pixels within a same clock cycle. These matrices are stored into different block RAM (BRAM) of the FPGA and are controlled by a dedicated address controller module.

Basic unit operation stage: The basic unit operator stage in matrix level covers decomposition stage in bit level, basic unit operator stage in bit level and composition stage in bit level. Ports A and B are the input ports of the K- processing unit supplied by the " sequencer and memory controller" module. These ports receive continuously the elements of the decomposed matrices S and C in every clock cycle.

Composition Stage – Matrix Level: The composition stage in the matrix level accumulates the $[q\times q] \times [q\times q]$ result to produce a final matrix multiplication result. The final matrix multiplication result is formed by accumulating the whole multiplication result $[q\times q]$ between the row of left-hand operand matrix with the column of right-hand operand matrix.

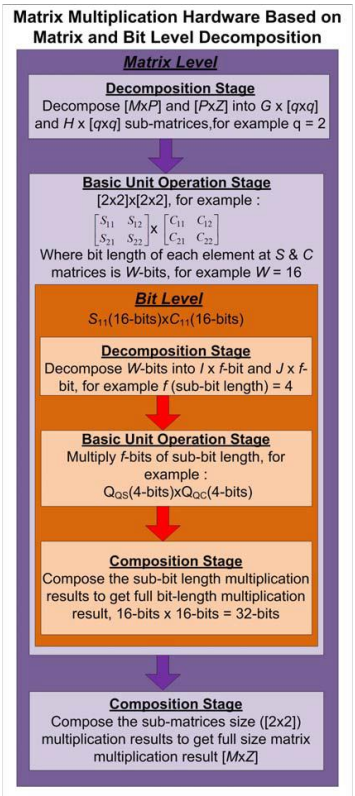

Fig 2. The decomposition, basic unit operation, and composition stages in matrix and bit level

III. PROPOSED SYSTEM

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

$$A \times B = (Ar - A) \times (Br - B) + Ar \times B + Br \times A - Ar \times Br \ . \ (1)$$

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 from (1), helping simplify the multiplication operation. Hence, to perform the multiplication process, the following expression is used:

$$A \times B = Ar \times B + Br \times A - Ar \times Br \ . \ (2)$$

Thus, one can perform the multiplication operation using three shift and two addition/subtraction operations. 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.

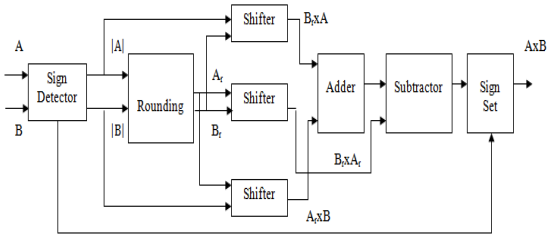

Fig. 3. Block diagram of the proposed multiplier.

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 is smaller than its corresponding rounded value while the other operand 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 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. Hence, we suggest that, before the multiplication operation starts, the absolute values of both inputs and the output sign of the multiplication result based on the inputs signs be determined and then the operation be performed for unsigned numbers and, at the last stage, the proper sign be applied to the unsigned result.

We provide the block diagram for of the proposed multiplier in Fig. 3 where the inputs are represented in two's complement format. First, the signs of the inputs are determined, and for each negative value, the absolute value is generated. Next, the rounding block extracts the nearest value for each absolute value in the form of 2n. It should be noted that the bit width of the output of this block is n (the most significant bit of the absolute value of an n-bit number in the two's complement format is zero).

Having determined the rounding values, using three barrel shifter blocks, the products Ar × Br , Ar × B, and Br × A are calculated. Hence, the amount of shifting is determined operand. Here, the input bit width of the shifter blocks is n, while their outputs are 2n. A single 2n-bit Kogge-Stone adder is used to calculate the summation of Ar × B and Br × A. The output of this adder and the result of Ar × Br are the inputs of the subtractor block whose output is the absolute value of the output of the proposed multiplier. Because Ar and Br are in the form of 2n, the inputs of the subtractor may take one of the three input patterns.

Finally, if the sign of the final multiplication result should be negative, the output of the subtractor will be negated in the sign set block. To negate values, which have the two's complement representation, the corresponding circuit based on $\bar{X}$ +1 should be used. To increase the speed of negation operation, one may skip the incrementation process in the negating phase by accepting its associated error. As will be seen later, the significance of the error decreases as the input widths increases.

In this paper, if the negation is performed exactly (approximately), the implementation is called signed RoBA (S-RoBA) multiplier [approximate S-RoBA (AS-RoBA) multiplier]. In the case where the inputs are always positive, to increase the speed and reduce the power consumption, the sign detector and sign set blocks are omitted from the architecture, providing us with the architecture called unsigned RoBA (U-RoBA) multiplier. In this case, the output width of the rounding block is n +1 where this bit is determined based on Ar [n] = A[n − 1] · A[n − 2]. This is because in the case of unsigned 11x . . . x (where x denotes do not care) with the bit width of n, its rounding value is 10…0 with the bit width of n + 1. Therefore, the input bit width of the shifters is n + 1. However, because the maximum amount

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 15
May 2018

of shifting is n − 1, 2n is considered for the output bit width of the shifters.

IV.RESULTS

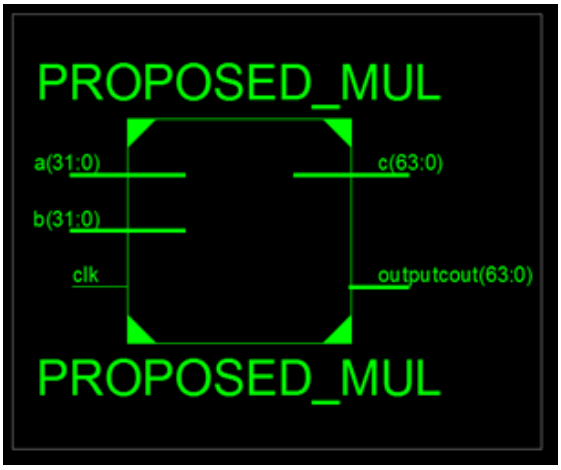

Fig 4. RTL Schematic

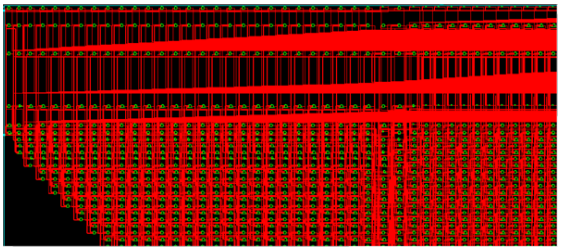

Fig 5. Technology Schematic

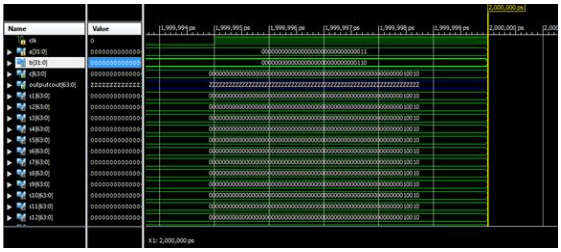

Fig 6. Output

V. 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. 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. Also, the efficacy of the proposed approximate multiplication approach was studied in two image processing applications of sharpening and smoothing. The comparison revealed the same image qualities as those of exact multiplication algorithms.

VI. 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.

K KRANTH KIRAN pursuing M.Tech at chintalapudi engineering college His area of interest is VLSI Design.

K.VENKATA RAO working as Associate Professor In Dept of ECE Chintalapudi Engineering College, Ponnur. His area of interest is V.L.S.I.

K. ANKA SIVA PRASAD completed his post graduated from bapatla engineering college. He has 11 years of teaching experience and at present he is working as HOD in chintalapudi engineering college. His area of interest is communication and signal processing.