Designing of a RoBA Multiplier in Verilog HDL for High-Speed, Energy-Efficient Digital Signal Processing

[1]G.Srinivas, [2]P.Venkatramulu, [3]A.Soujanya, [4]M.Saivenkat

[1,2,3]Assistant Professor, [4]UG Student, [1,2,3,4]Department of Electronics and Communication Engineering, Visvesvaraya College of Engineering & Technology, Hyderabad, India.

Abstract

In this study, we provide a speedy but efficient imperfect multiplier. The approach is to modify the operands to the nearest two examples. In this way, the computationally advanced element of the enhancement is prevented from enhancing speed and energy usage at the cost of a little error. The suggested approach is applicable to both signed and unsigned increases. We suggest using the approximate multiplier in three different ways, one for unsigned jobs and two for marked tasks. By comparing the suggested multiplier's performance to several estimated and precise multipliers using different plan parameters, the usefulness of the multiplier is evaluated. The practicality of the anticipated multiplier provided is also taken into account in two applications for creating pictures, namely honing and smoothing.

Keywords: Accuracy, approximatecomputing, energy efficient, error analysis, high speed, multiplier

INTRODUCTION

One of the top design criteria for practically all electronic systems, particularly mobile ones like smart phones, tablets, and other devices, is energy efficiency [1]. Winnable this reduction with token performance (speed) penalty is highly sought [1]. Blocks for digital signal processing (DSP) are essential components of those mobile devices that enable a variety of multimedia system applications. The arithmetic logic unit, which accounts for the majority of the arithmetic operations carried out by these DSP systems, is the procedural core of those blocks [2]. Therefore, increasing multipliers' speed and power/energy-efficiency qualities is essential for increasing CPU efficiency.

Many of the DSP cores implement image and video process- ing algorithms wherever final outputs ar either pictures or videos ready for humanconsumptions. This reality allows U.S.A. to use approximations for up the speed/energy potency. This originates from the restricted sensory activity skills of groups of people in perceptive a picture ora video. additionally to the image and video processapplications, there ar alternative areas wherever the accuracy of the arithmetic operations isn't importantto the practicality of the system &#40;see [3], [4]&#41;. the flexibility} to use the approximate computing provides the designer with the ability of constructing tradeoffs between the accuracy and also the speed similarly as power/energy consumption [2], [5]. Applying the approximation to the arithmetic units may be performed at totally different style abstraction levels together with circuit, logic, and design levels, similarly as algorithmic rule and software package layers [2].

In this paper, we tend to target proposing a high-speed low- power/energy however approximate multiplier factor acceptable for error resilient DSP applications. The planned approximate mul- tiplier, that is additionally space economical, is built by mod- ifying the traditional multiplication approach at the algo- rithm level forward rounded input values. we tend to decision this rounding-based approximate (RoBA) multiplier factor. The planned multiplication approach is applicable to each signed and unsigned multiplications that 3 optimized archi- tecturesar conferred. The efficiencies of these structures ar assessed by scrutiny the delays, power and energy consumptions, energy-delay product (EDPs), and areas with those of some approximate and correct (exact) multipli- ers. The contributions of this paper will be summarized as follows:

Presenting a brand new theme for RoBA multiplication by modifying the traditional multiplication approach;

Describing 3 hardware architectures of the pro- posed approximate multiplication theme for sign and unsigned operations.

The rest of this paper is organized as follows. Section II discusses the connected works concerning approximate multipliers. The planned theme of the approximate multiplication, its hard- ware implementations, and its accuracy results ar conferred in Section III. In Section IV, the characteristics of the pro- posed approximate multiplier factor compared with the correct and approximate multipliers, and additionally its effectiveness in image process applications arstudied. Finally, the conclusion is drawn in Section V.

EXISTINGWORK

In this section, some of the preceding works in the discipline of approximate multipliers are temporarily reviewed. In [3], an approx- imate multiplier and an approximate adder based on atech- nique named broken-array multiplier (BAM) have been proposed. By making use of the twoBAM approximation approach two of [3] two tothetraditional modified Booth multiplier, anapproximate signed Booth multiplier wasintroduced in [5]. The approximate multiplierfurnished electricity consumption financial savingsshape 28% to 58.6% and location discounts from19.7% to 41.8% for exclusive word lengths inassessment with a ordinary Booth multiplier. Kulkarni et al. [6] recommended an approximatemultiplier con- sisting two of a quantity of 22 inaccurate building blocks that saved the powerwith the aid of 31.8%–45.4% over an accuratemultiplier. An approximate signed 32-bit multiplierfor speculation functions in pipelined processorswas once designed in [7]. It used to be 20% quickerthan a full-adder-based tree multiplier while havinga probability of error of round 14%. In [8], an error-tolerant multiplier, which computed theapproximate result with the aid of dividing the multiplication into one correct and one approximatepart, used to be introduced, in which the accuracies for one-of-a-kind bit widths were reported. In thecase of a 12-bit multiplier, a electricity saving ofextra than 50% was reported. In [9], twoapproximate 4:2 compressors for utilizing in aeveryday Dadda multiplier were designed and analyzed.

The use of approximate multipliers in image processing applications, which leads to discounts in electricity consumption, delay, and transistor be counted compared with these of an genuine multiplier design, has been mentioned in the literature. In [10], an accuracy-configurable multiplier structure (ACMA) used to be counseled for error-resilient systems. To enlarge its through- put, the ACMA made use of a technique known as carry-in prediction that labored based on a precomputation logic. When compared with the exact one, the proposed approximate multiplication resulted in nearly 50% reduction in the latency via reducing the fundamental path. Also, Bhardwaj et al. [11] presented an approximate Wallace tree multiplier (AWTM). Again, it invoked the carry-in prediction to limit the essential path. In this work, AWTM was used in a real-time benchmark image software showing about 40% and 30% discounts in the energy and area, respectively, without any image nice loss in contrast with the case of using anaccurate Wallace tree multiplier (WTM) structure.

In [12], approximate unsigned multiplication and division based totally on an approximate logarithm of the operands two have two been proposed. In the proposed multiplication, the summation of the approximate logarithms determines the result of the operation. Hence, the multiplication is simplified to some shift and add operations. In [13], a technique for increasing the accuracy of the multiplication approach of [12] was proposed. It was based on the decomposition of the input operands. This technique extensively multiplied the common error at the rate of growing the hardware of the approximate multiplier with the aid of about two times.

In [16], a dynamic section method (DSM) is presented, which performs the multiplicationoperation on an m-bit section beginning from the main one bit of the input operands. A dynamic vary unbiased multiplier (DRUM) multiplier, whichselects an m-bit segment starting from the leading one bit of the input operands and units the least giant bit of the truncated values to one, has been proposed in [17]. In this structure, the truncatedvalues are multiplied and shifted to left to generate the remaining output. In [18], an two approximate 44 WTM has been proposed that uses an inaccurate 4:2 counter. In addition, an error correction unit for correcting the outputs has been suggested. To two assemble large multipliers, this 44 inaccurate Wallace multiplier can be used in an array structure. Most of the earlier proposed approximate multipliers are primarily based on either editing the shape or complexity discount of a precise accurate multiplier. In this paper, similar to [12], we recommend performing the approximatemultiplication thru simplifying the operation. The difference between our work and [12] is that, even though the ideas in both works are nearly comparable for unsigned numbers, the imply error of our proposed strategy is smaller. In addition, we recommend some approximation strategies when the multiplication is performed for signed numbers.

PROPOSED ARCHITECTURE

A. Multiplication Algorithm of RoBA Multiplier

The main plan behind the planned approximate number is to build use of the convenience of operation once the numbers square measure 2 to the power n (2n). To elaborate on the operation of the approximate number, first, allow us to denote the rounded numbers of the input of A and B by Ar and Br , severally. The multiplication of A by B could also be rewritten as

$$A \times B = (A_r - A) \times (B_r - B) + A_r \times B$$
$$+ B_r \times A - A_r \times B_r \, . \qquad (1)$$

The key perception is that the augmentationsof ArBr ,Ar B, and Br A might be executed just by the move operation. The equipment execution of (Ar A) (Br B), be that as it may, is somewhat unpredictable. The heaviness of this term in the last outcome, which relies upon contrasts of the accuratenumbers from their adjusted ones, is normally little. Subsequently, we propose to exclude this part from (1), rearranging the increase task. Subsequently, to play out the increase procedure, the accompanying articulation is utilized:

$$A \times B \sim= A_r \times B + B_r \times A - A_r \times B_r \, . \qquad (2)$$

Thus, one can perform the multiplication operation the use of three shift and twoaddition/subtraction operations. In this approach, the nearest values for A and B in the structure of 2n need to be determined. When the value of A (or B) is equalto two the 3 two twotwo 2 p−2 two (where two p is two an two arbitrary wonderful integerlarger than one), it two has two two nearest two values in two the form of two 2n with equal absolute variations that are 2 p and 2 p−1. While each values lead two to two the two identical two effect on two the two accuracy of theproposed multiplier, choosing the larger one (except for the case of two p 2) two leads to a smaller hardware implementation for two identifying the nearest two rounded two value, two and two hence, two it isviewed in this paper. It originates from the reality that the numbers in the structure of three 2 p−2 are considered as do notcare in both rounding up and down simplifying the process, and smaller logic expressions can also be executed if they are used in the rounding up.The solely exception is for three, which in this case, two is considered as its nearest cost in the proposed approximate multiplier.

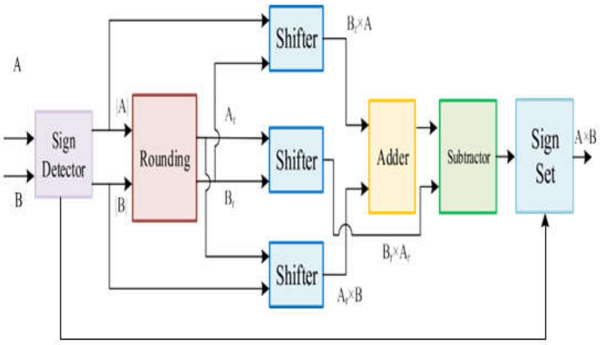
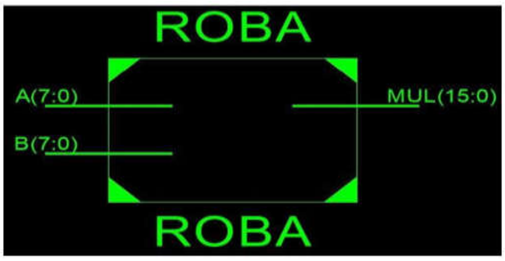


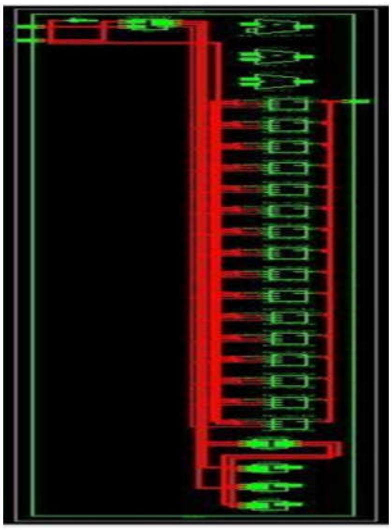
Fig. 1. Block diagram for the hardware implementation of the proposed multiplier.

It ought to be noted that opposite to the previous work the place the approximate end result is smaller than the precise result, the remaining result calculated with the aid of the RoBA multiplier may additionally be both large or smaller than the two exact result depending on the magnitudes of Ar and Br two compared with those of two A and B, respectively. Note that if one of the operands (sayA) is smaller than its corresponding rounded fee whilst the other operand (say B) is large than its corresponding rounded value, then the approximate end result will be large than the genuine result. This is due to the reality that, in this case, the multiplication end result of (Ar A) (Br B) will be negative. Since the difference between (1) and (2) isprecisely this product, the approximate result turns into larger than the genuine one. Similarly, if both A and B are large or both are smaller than two Ar two and Br , then two theapproximate end result will be smaller than the precise result. Finally, it be cited the two advantage of two the proposedRoBA multiplier exists solely for high-quality two inputs because two in the two's complement representation, the rounded values of poor 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 ofthe multiplication end result based totally on theinputs signs be determined and then the operation be carried out for unsigned numbers two and, two at the ultimate stage, the proper signal be applied tothe unsigned result. The hardware implementationof the proposed approximate multiplier is explained next.

RESULTS

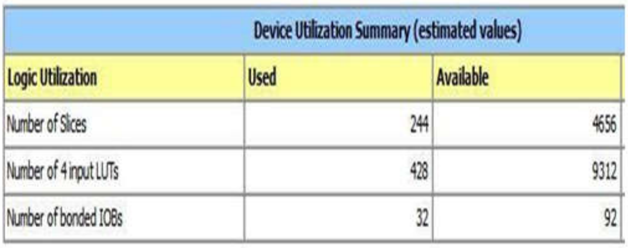


Top level block



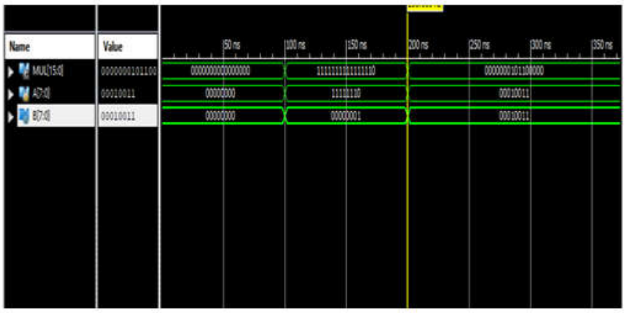
RTL Schematic

Technological schematic



Device Utilization Summary (estimated values)		
Logic Utilization	Used	Available
Number of Slices	244	4656
Number of 4 input LUTs	428	9312
Number of bonded IOBs	32	92

Design summery



Simulation output

CONCLUSION

In this study, we introduced the RoBA multiplier, a high-speed yet energy-efficient approximation multiplier. The suggested multiplier's great accuracy was partly due to the inputs being rounded to the power 2n. By doing so, it was possible to increase speed and energy consumption at the expense of a tiny inaccuracy by skipping the computationally costly part of the multiplication process. The suggested method used to be applicable to both signed and unsigned multiplications. One for unsigned operations and two for signed operations, totaling three hardware implementations of the approximation multiplier, have been considered. The effectiveness of the suggested multipliers was assessed by comparing two of them with certain precise and approximate multipliers utilizing specific plan characteristics. The outcomes revealed that, in most (all) cases, the RoBA multiplier architectures out- carried out the corresponding approximate (exact) multipliers. Also, the efficacy of the proposed approximate multiplication method used to be studied in two photograph processing functions of sharpening and smoothing. The contrast printed the equal photo qualities as those of actual multiplication algorithms.

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 digitalsignal processing,” in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADS), Oct. 2013, pp.25–30.