This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LES.2017.2778341, IEEE Embedded Systems Letters

A Novel Heterogeneous Approximate Multiplier for Low Power and High Performance

Ihsen Alouani[1], Hamzeh Ahangari[2], Ozcan Ozturk[2], and Smail Niar[1]

[1]LAMIH lab, University of Valenciennes, France
[2]Department of Computer Engineering, Bilkent University, Ankara, Turkey

***Abstract*—Approximate computing is a design paradigm considered for a range of applications that can tolerate some loss of accuracy. In fact, the bottleneck in conventional digital design techniques can be eliminated to achieve higher performance and energy efficiency by compromising accuracy. In this paper, a new architecture that engages accuracy as a design parameter is presented where an approximate parallel multiplier using heterogeneous blocks is implemented. Based on design space exploration, we demonstrate that introducing diverse building blocks to implement the multiplier rather than cloning one building block achieves higher precision results. We show experimental results in terms of precision, delay and power dissipation as metrics and compare with 3 previous approximate designs. Our results show that the proposed heterogeneous multiplier achieves more precise outputs than the tested circuits while improving performance and power tradeoffs.**

I. Introduction

With the increase in the amount of data and complexity of tasks supported by battery-operated electronic devices, there is a continuous for design techniques to conserve power consumption, while achieving the desired performance. In fact, new generations of embedded systems are designed to process power hungry applications that handle heavy workloads. For example, in mobile devices, systems need to process multimedia content, recognize patterns, and interact intelligently with their environment. This trend impacts directly the computing paradigm due to the new specific demands in applications which are not necessarily aiming at a precise numerical result; instead, they try to achieve a sufficient quality of results. Therefore, Digital Signal Processing (DSP) has become one of the most attractive topics in semiconductor industry in the past 30 years. According to previous studies [18], the global market share of DSP architectures exceeds 95% of the total volume of processors sold. A wide range of multimedia applications such as image, voice and video processing, data searching, recognition ...etc are highly tolerant to errors and their quality of service is not affected by a certain amount of precision loss. In [7], authors analyzed a benchmark suite of 12 recognition, mining and search applications and found that on average, 83% of the runtime computations can tolerate at least some degree of approximation. Hence, for this type of applications, there is a change in design methodology towards approximate computing rather than the classical accurate computing design. Approximate computing relies on the range of tolerated inaccuracy in the computational process to improve power efficiency and performance.

II. Related Work

To reduce power consumption of CMOS circuits, a commonly used approach is to aggressively scale supply voltage beyond the nominal value. However, this technique has considerably negative drawbacks on the quality of service and leads to a degradation in terms of performance. While algorithmic noise tolerance schemes [17] are meant to compensate this degradation, the new circuits already have very low voltages and are no longer allow systematic use of this technique.

Previous works proposed reducing combinational circuit complexity through approximate computing systems. The main objective is to design circuits with lower number of transistors leading to a reduction in delay and power consumption. A reduction in circuit complexity at transistor level in an adder circuit provides a more important reduction in power consumption compared to the conventional low power design techniques [15]. In [16], authors proposed a logic synthesis approach to design circuits for implementing approximate functions by considering error rate (ER) as metric for accuracy.

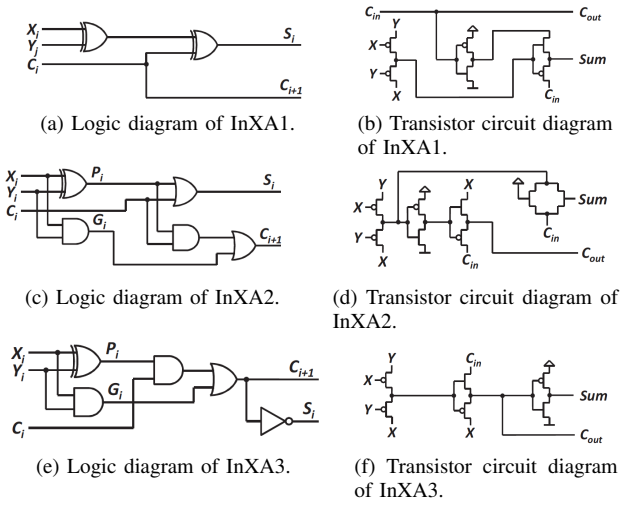
As one of the key components in arithmetic circuits, many approximation schemes of adder implementations were proposed. Segmented adders are implemented in [12]–[14] by several smaller adders operating in parallel where the carry propagation sequence is truncated into shorter segments. Another method for reducing the critical path delay and power dissipation of a conventional combinational circuit is by approximating their elementary full adder blocks [8]–[11].

While adders have been extensively studied, there has been relatively less work in the literature that focus on approximate multipliers. In [3], approximate partial products are computed based on approximate 2 by 2 elementary multipliers, while a tree of accurate adders is used to accumulate the elementary products. Authors of [4] studied the implementation of approximate adders for the final stage addition in a multiplier design. In [5], authors proposes the error tolerant multiplier (ETM) which suggests splitting the multiplier into an accurate multiplication part for MSBs and a non-accurate multiplication part for LSBs.

All of these works consider a homogeneous design pattern and rely on a single implementation of approximate elements

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LES.2017.2778341, IEEE Embedded Systems Letters

to build their circuits. In this work, we propose a new approximate circuit design methodology in which we consider a set of different adder blocks to build a heterogeneous multiplier. We use the three approximate implementations of full adders proposed by [1] shown in Figure 1 as possible adder implementations and explore the design space to converge to an optimal heterogeneous design.



(a) Logic diagram of InXA1.

(b) Transistor circuit diagram of InXA1.

(c) Logic diagram of InXA2.

(d) Transistor circuit diagram of InXA2.

(e) Logic diagram of InXA3.

(f) Transistor circuit diagram of InXA3.

Fig. 1: The three considered inexact adder cells proposed in [1].

III. Proposed Architecture

In this work, we use a new methodology to build approximate circuits for a heterogeneous approximate multiplier. We propose to rely on a set of different inexact elementary blocks instead of one block that is replicated to build the desired circuit. The purpose is to take advantage of the design flexibility given by diversifying the approximate elements to comprehensively benefit from the error masking mechanisms; thereby reducing precision loss. More specifically, the logical masking mechanism is applied, where an error propagates to reach a gate's input while another input is in controlling state (for example, a "0" input of a AND gate). Hence, the idea is to explore, at design time, the set of inexact implementations to identify those with the minimum number of errors propagating through the circuit to the output. This way, we increase the precision of the overall circuit. Therefore, we proceed to a design space exploration phase in order to select the most accurate design combination.

Algorithm 1 details the design space exploration phase. The objective is to find the combination of full adders that minimizes the mean error distances (MED) for the overall design, where MED is the average arithmetic deviation from the accurate design. The idea is based on Genetic Algorithm (GA) exploration algorithm and adapted to our design problem. The exploration process is achieved in three steps corresponding to the architecture full adders lines. The first step lunches GA exploration on the first line full adders (see Figures 2) with considering the remaining subsequent blocs as exact circuits.

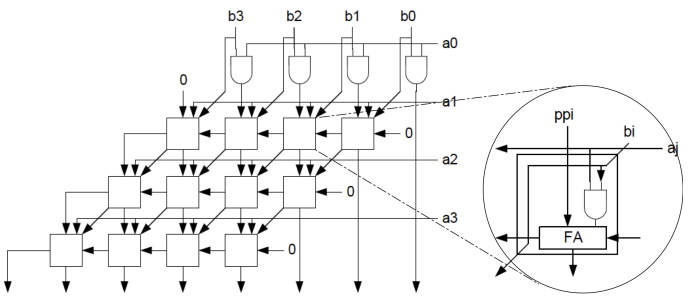


Fig. 2: A 4x4 multiplier architecture.

The result of this step is a set of implementations with the lowest MED that we refer to as S_1. In the next step, we iterate the exploration with the first and the second lines with an exact implementation of the third line. The second line is explored based on GA while the considered exploration space of the first line is S_1. From this step we extract a new set of best implementations that we refer to as S_2. Finally, the same process is applied for the whole circuit to get the overall best implementation.

Algorithm 1: Exploration to minimize MED

// explore the first line ;
Init($Line_1$);
$S_1 = Genetic\_Op(Line_1)$;
// explore the second line ;
Init($Line_2$);
$S_2 = Genetic\_Op(\{S_1; Line_2\})$;
// explore the third line ;
Init($Line_3$);
$C_{mult} = Genetic\_Op(\{S_1; S_2; Line_3\})$;
$return(C_{mult})$;
// return the combination with minimum MED ;

IV. Precision Evaluation

To assess the precision of our architecture, simulations are pursued and results are compared with the previous approximate designs. The following performance metrics are used for evaluation purposes:

- Error Distance (ED): ED is the arithmetic difference between the exact result R^* and the approximate result R, i.e.,
$$ED = \mid R^* - R \mid . \quad (1)$$
- Mean Relative Error Distance (MRED), which is the average of the Relative Error Distance, where RED is given by:
$$RED = \frac{ED}{R^*}. \quad (2)$$

Figure 3 shows the results in terms of Mean Relative Error (MRED) along with the relative number of transistors (%) compared with respect to the exact implementation. While the implementation of the proposed multiplier costs only 35%

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LES.2017.2778341, IEEE Embedded Systems Letters

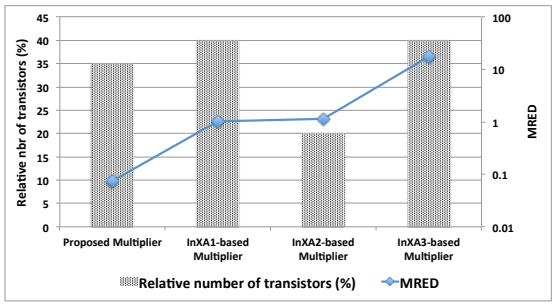


Fig. 3: MRED compared with previous homogeneous approximate multipliers.

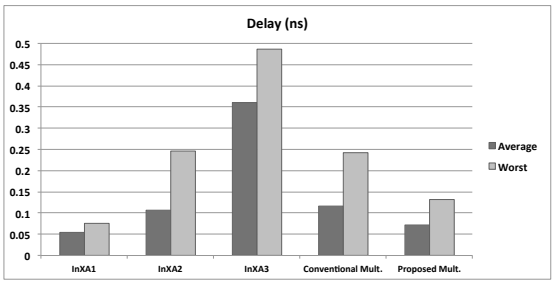


Fig. 5: Average and worst case delay for previously proposed approximate multipliers, our multiplier, and conventional multiplier.

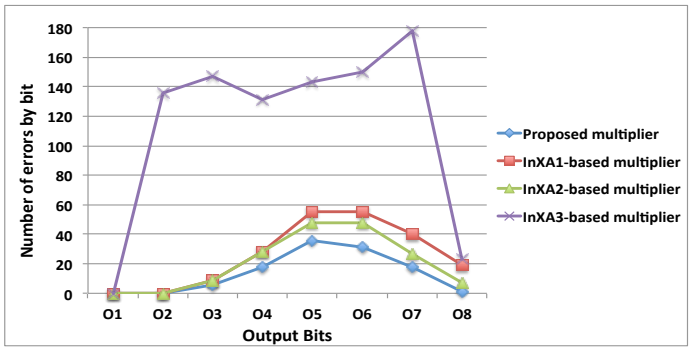


Fig. 4: Distribution of ED within the minimal-error circuits compared to the InXA1, InXA2, and InXA3-based multiplier.

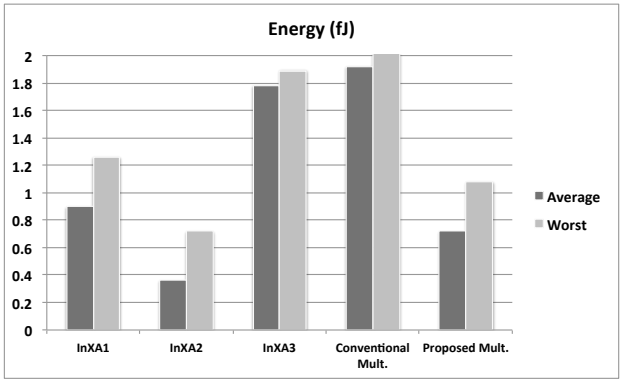


Fig. 6: Average and worst case energy dissipation for previous, proposed approximate multipliers and conventional multiplier.

of the number of transistors used in the exact multiplier, it achieves the lowest MRED, thereby achieving the most accurate results compared with the homogeneous implementations.

V. Performance and Energy Evaluation

In this section, we compare the proposed approximate multiplier with the previously proposed schemes at 45nm with PTM [20] using Advanced Design System (ADS) simulation platform. All input combinations are tested exhaustively and the results in terms of delay and energy are shown in Figures 5 and 6 for the average and the worst case.

As shown in these figures, the proposed multiplier outperforms InXA2 and InXA3 based approximate multipliers in terms of performance metric. Even though there is a delay overhead compared to InXA1-based circuit, InXA1-based multiplier consumes 24% more energy compared to our multiplier.

VI. Approximate Computing Applications

A. Edge Detection Through Sobel Filter

The Sobel operator, or Sobel filter, is a widely used tool in image processing and computer vision applications, particularly within edge detection algorithms where it emphasizes the edges of a grayscale image. Technically, it consists of a discrete differentiation operator, that approximates the gradient of the image intensity function. At each pixel, the result of the Sobel filter can be the corresponding gradient vector or the norm of the bidirectional gradient. The Sobel operator is based on convolving the image with a small, separable, and integer-valued matrix in the X and Y directions.

TABLE I: Mean Square Error (MSE) compared to exact filter

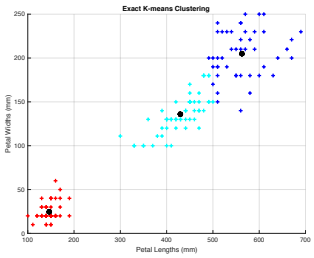
Proposed Filter	InXA1-based	InXA2-based	InXA3-based
9.76	21.32	18.76	52.67

We use the Mean Square Error (MSE) metric to assess the quality of the different approximate filters compared to the reference output achieved by the exact filter. The results shown in Table I demonstrate that the proposed multiplier is much more accurate in the Sobel operator implementation compared to the homogeneous implementations. It clearly has a lower MSE, which means that the deviation from the reference output is very low.

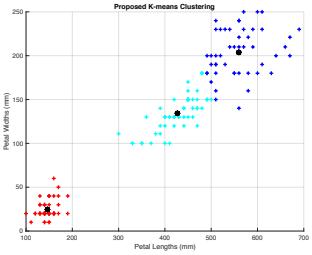
B. K-means based clustering

K-means clustering algorithm aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. We implement k-means for the Fisher Iris dataset using: exact multiplier, the proposed multiplier and the three homogeneous multipliers. As shown in 7, the proposed approximate multiplier based implementation achieves better results with only one erroneous classification case out of 150

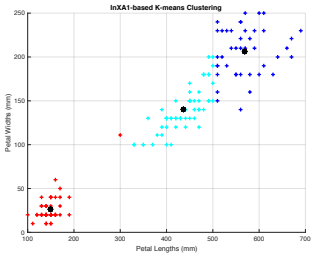
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LES.2017.2778341, IEEE Embedded Systems Letters



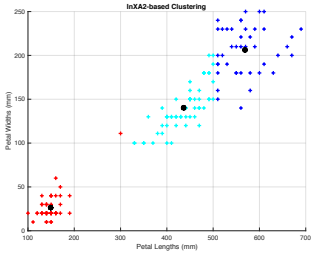
(a) Exact K-means Clustering.



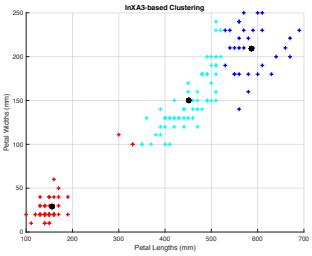
(b) K-means Clustering with the proposed multiplier.



(c) K-means Clustering with the InXA1-based multiplier.



(d) K-means Clustering with the InXA2-based multiplier.



(e) K-means Clustering with the InXA3-based multiplier.

Fig. 7: K-means clustering results.

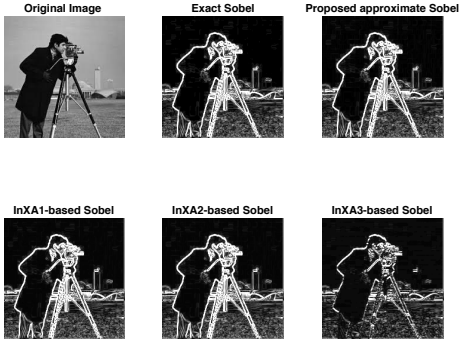


Fig. 8: Original image and the Sobel Filter implementations for: exact implementation, proposed multiplier, InXA1, InXA2 and InXA3.

observations compared with 5, 5 and 16 errors in InXA1, InXA2 and InXA3 based implementations respectfully.

VII. Conclusion

In this paper, we propose a novel heterogeneous architecture that uses accuracy as a design parameter. Specifically, we build an approximate parallel multiplier based on different approximate implementations. After design space explorations, we realized that introducing different elementary architectures to implement the circuit leads to lower error rates compared to the classical homogeneous designs. In fact, the proposed design benefits from the masking mechanisms within logic elements in different cases to limit the overall deviation from the exact results. Our experiments show that the utilized design method results in an approximate multiplier with higher accuracy and better tradeoffs compared with previous circuits.

References

[1] H. A. F. Almurib, T. N. Kumar and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," DATE Conference, Dresden, 2016, pp. 660-665.

[2] Shih-Lien Lu, "Speeding up processing with approximation circuits," in Computer, vol. 37, no. 3, pp. 67-73, Mar 2004.

[3] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in 24th IEEE Intl. Conf. on VLSI Design, 2011, pp. 346-351.

[4] J. Huang, J. Lach, and G. Robins, "A methodology for energy-quality tradeoff using imprecise hardware," in Design Automation Conference DAC 2012, pp. 504-509.

[5] K. Y. Kyaw, W. L. Goh, and K. S. Yeo,"Low-power high-speed multiplier for error-tolerant application" in IEEE Intl. Conf. Electron Devices and Solid-State Circuits (EDSSC), 2010, pp. 1-4.

[6] P. Kulkarni, P. Gupta and M. Ercegovac, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," 2011 24th International Conference on VLSI Design, Chennai, 2011, pp. 346-351.

[7] V. K. Chippa et. al. "Analysis and characterization of inherent application resilience for approximate computing". In Proc. DAC, 2013.

[8] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi. "Approximate XOR/XNOR-based adders for inexact computing". In IEEE-NANO, pages 690-693, 2013.

[9] D. Nanu, R. P. K., D. Sowkarthiga, and K. S. A. Ameen. "Approximate adder design using cpl logic for image compression". International Journal of Innovative Research and Development, 3(4):362-370,2014.

[10] 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., 57(4):850-862, 2010.

[11] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. "Low-power digital signal processing using approximate adders". IEEE Trans. CAD, 32(1):124-137, 2013.

[12] N. Zhu, W. L. Goh, and K. S. Yeo. " An enhanced low-power high-speed adder for error-tolerant application". In ISIC 2009, pages 69-72, 2009.

[13] D. Mohapatra, V. Chippa, A. Raghunathan, and K. Roy. "Design of voltage-scalable meta-functions for approximate computing". In DATE'11, pages 1-6, 2011.

[14] A. B. Kahng and S. Kang. "Accuracy-configurable adder for approximate arithmetic designs". In DAC'12,pages 820-825, 2012.

[15] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32,no. 1, pp. 124-137, Jan 2013.

[16] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proc. Design, Automat. Test Europe, DATE 2010, pp. 957-960.

[17] R. Hegde and N. R. Shanbhag, "Soft digital signal processing", IEEE Trans. On Very Large Scale Integration System, vol. 9, no. 6, pp. 813 -823, Jun.2001.

[18] Liu, D., Embedded DSP Processor Design, Morgan Kaufmann Publishing, 2008. p 808

[19] J. Liang, J. Han, and F. Lombardi, " New metrics for the reliability of approximate and probabilistic adders," IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771, 2013.

[20] Predictive technology model (PTM) website. [Online]. Available: http://ptm.asu.edu