Comparative Study of Approximate Multipliers

Mahmoud Masadeh , Osman Hasan and Sofiène Tahar
Dept. of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada
{m_masa,o_hasan,tahar}@ece.concordia.ca

ABSTRACT

Approximate multipliers are widely being advocated for energy-efficient computing in applications that exhibit an inherent tolerance to inaccuracy. In this paper, we identify three decisions for design and evaluation of approximate multiplier circuits: (1) the type of approximate full adder (FA) used to construct the multiplier, (2) the architecture, i.e., array or tree, of the multiplier and (3) the placement of sub-modules of approximate and exact multipliers in the target multiplier module. Based on FA cells implemented at the transistor level (TSMC65nm), we developed several approximate building blocks of 8x8 multipliers, as well as various implementations of higher order multipliers. These designs are evaluated based on their power, area, delay and error and the best designs are identified. We validate these designs on an image blending application using MATLAB, and compare them to related work.

CCS CONCEPTS

• **Hardware** → *Arithmetic and datapath circuits*; *Circuits power issues*;

KEYWORDS

Approximate Computing; Approximate Multiplier; Power-efficiency

ACM Reference Format:
Mahmoud Masadeh , Osman Hasan and Sofiène Tahar. 2018. Comparative Study of Approximate Multipliers. In *GLSVLSI '18: 2018 Great Lakes Symposium on VLSI, May 23–25, 2018, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3194554.3194626

1 INTRODUCTION

The pervasive, portable, embedded and mobile nature of present age computing systems has led to an increasing demand for ultra low power consumption, small footprint, and high performance. Approximate computing (AC) [1] is a nascent computing paradigm that allows us to achieve these objectives by compromising the arithmetic accuracy. Many systems used in domains, like multimedia and big data analysis, exhibit inherent tolerances to a certain level of inaccuracies in computation, and thus can benefit from AC.

Functional approximation [2], in hardware, mostly deals with the design of approximate arithmetic units, such as adders and multipliers, at different abstraction levels, i.e., transistor, gate, register tansfer level and application. Some notable approximate adders include *speculative adders*, *segmented adders*, *carry select adders* and *approximate full adders* [3]. The transistor level approximation provides the highest flexibility due to the ability to tweak most of the design parameters at this level. Various approximate full adder (FA) at the transistor level have been proposed including the mirror adders [4], the XOR/XNOR based FA [5] and the inexact FA [6].

Approximate multipliers have been mainly designed using three techniques, i) *Approximation in partial products generation*, ii) *Approximation in partial product tree*, and iii) *Approximation in partial products summation*. Jiang et al.[7] compared the characteristics of different approximate multipliers, implemented in VHDL based on these different techniques. *We target approximate multipliers based on approximation in partial products summation.*

In this paper, we evaluate and compare the accuracy and circuit characteristics of different approximate multipliers. These multipliers are designed based on three identified decisions: (1) the type of approximate FA used to construct the multiplier, (2) the architecture of the multiplier, and (3) the placement of sub-modules of approximate and exact multipliers in the target multiplier module.

2 PROPOSED METHODOLOGY

The design space for approximate multipliers based on different approximate FAs and compressors is quite huge. However, it is difficult to select the most suitable design for a specific application. Figure 1 presents an overview of our proposed methodology to build different approximate multipliers and compare their design metrics to select the most suitable design. It consists of the following steps:

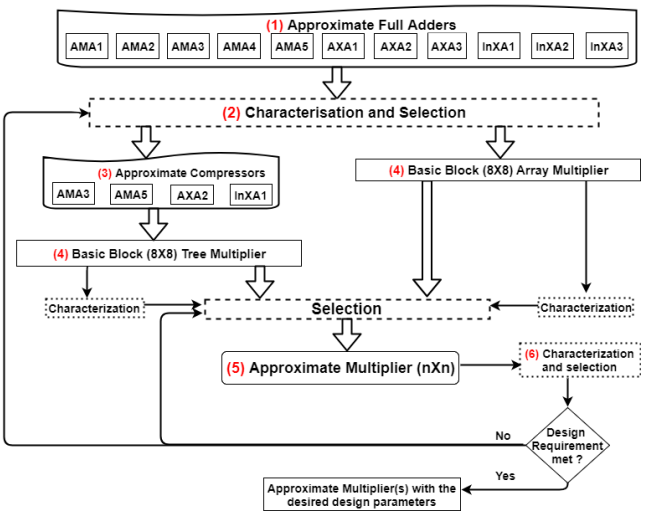

Figure 1: Methodology Overview

GLSVLSI '18, May 23–25, 2018, Chicago, IL, USA

ACM ISBN 978-1-4503-5724-1/18/05…$15.00
https://doi.org/10.1145/3194554.3194626

(1) *Building a library of elementary approximate FAs using the TSMC65nm technology in Cadence Spectre*: We use the default transistors of this technology to build 11 approximate FA designs comprising of 5 mirror FAs, 3 XOR/XNOR FAs and 3 inexact FAs.

(2) *Characterization and early space reduction*: We perform area, power, latency and quality characterizations of different approximate FAs to filter out non-Pareto designs.

(3) *Building a library of approximate compressors*: We build a Cadence library of approximate compressors using the optimal approximate FAs, as recommended by [4].

(4) *Building approximate multipliers basic blocks*: Based on approximate FAs and compressors, we design various approximate 8x8 array and tree multipliers, respectively.

(5) *Designing target approximate multipliers*: Based on different configurations of 8x8 approximate multipliers, the target multiplier modules are designed and characterized.

(6) *Selection of design points*: Considering the required quality constrains of a specific application, a subset of power-efficient design points are selected.

In order to evaluate the efficiency of the proposed approximate designs, *power consumption* and *area*, represented by the number of transistors used, are measured. Circuit performance is measured by the maximum *delay* between changing the inputs and observing the output(s). Besides these basic design metrics, we also measure *accuracy* using, among others, Error Rate (ER) and Normalized Mean Error Distance (NMED) [8].

3 APPROXIMATE FAS AND COMPRESSORS

Low power approximate binary adders are generally constructed by replacing the accurate FAs with approximate FAs. We consider five approximate mirror adders (AMA1, AMA2, AMA3, AMA4 and AMA5) [4], three approximate XOR/XNOR based full adders (AXA1, AXA2 and AXA3) [5] and three inexact adder cells (InXA1, InXA2 and InXA3) [6].

Table 1: Characteristics of Different Approximate FAs

FA Type	Size (A)	Power(nw) (P)	Delay(ps) (D)	# of Error Cases (E)	PDP(fJ)
Exact FA	28	763.3	244	0	186.25
AMA1 (M1)	20	612	195	2	119.34
AMA2 (M2)	14	561.1	366	2	205.36
AMA3 (M3)	11	558.1	360	3	200.92
AMA4 (M4)	15	587.1	196	3	115.07
AMA5 (m5)	8	412.1	150	4	61.82
AXA1 (X1)	8	676.2	1155	4	781
AXA2 (X2)	6	358.7	838	4	300.59
AXA3 (X3)	8	396.5	1467	2	582
InXA1 (In1)	6	410	740	2	303.4
InXA2 (In2)	8	355.1	832	2	295.44
InXA3 (In3)	6	648	767	2	753.5

Table 1 shows the characteristics of the 11 considered approximate FAs including Size (A), Power consumption (P), Delay (D), number of Erroneous outputs (E), which indicates the likelihood of at least one output (Cout or Sum) being wrong, and Power-Delay-Product (PDP). *All approximate FAs are Pareto-points*, i.e., they provide less area and power consumption compared to the exact design at the cost of compromising accuracy [9]. In [10], AMA5 is considered as a *wire* with zero area and zero power consumption. However, this is unrealistic as the output of AMA5 has to drive other signals. Thus, we used two buffers instead of two wires to design it. Assuming that the characteristics of approximate FAs are linearly applied to approximate arithmetic circuits, there is no single approximate FA, which is superior in all aspects. Therefore, we propose to use a *fitness function* to evaluate FA designs, or any approximate circuit, based on its design metrics.

$$Fitness = C1 * A + C2 * P + C3 * D + C4 * E \quad (1)$$

where *C1*, *C2*, *C3* and *C4* are application-dependent design coefficients within the range [0,1] which provide weights to specific design metrics for a specific application, e.g., E equals zero for the exact design, and P is small for low power designs. *A minimal fitness value is preferred since the goal is to minimize A, P, D and E.* For the remainder of this work, we use all 11 Pareto-design approximate FAs as elementary cells to construct approximate array multipliers.

Higher-order compressors, e.g., 5-to-3 and 8-to-4 [11], allow us to construct high speed tree multipliers. Therefore, we also developed approximate FA based compressors, for evaluation purposes. Considering all options, the total combination of compressor settings grows exponentially, e.g., for an 8-to-4 compressor, we have O((# of FA designs)$^{\#\text{ of FAs in compressor}}$) = O $(11)^4$ = 14641 combinations. Therefore, in order to show the effectiveness of designing approximate compressors based on approximate FAs, we chose four FAs only, i.e., *AMA5*, *AXA2*, *InXA1* and *AMA3* as explained in detail in [8]. These selected FAs are used to build approximate high-order compressors, which in turn can be used for designing approximate tree multipliers. A detailed overview of the characteristics for the chosen approximate compressors can be found in [8]. However, these selected compressors are not guaranteed to be optimal. But, they exhibit some improvements compared to the exact designs.

4 MULTIPLIER BASIC BLOCKS

In this section, we use the approximate FAs and compressors, described earlier, to design 8x8 array and tree based multipliers, respectively. Which will act as our basic blocks for designing higher-order multipliers, e.g., 32x32 , as it will be discussed in Section 5.

4.1 8x8 Array Multiplier

An n-bit array multiplier [12] is composed of n^2 AND gates for partial products generation, and *n-1* n-bit adders for partial products accumulation. The design space of an *nxn* approximate array multiplier is quite huge, since it depends on the type of FA used in the array, and the *number* of approximate FAs (from 0 to *n*) used in the n-bit adder. Considering all options, the total combination of multiplier settings grows exponentially O((# of FAs)$^{\text{MultiplierSize}^2}$) = O $((11)^{n^2})$ = $(11)^{64}$ in our case.

We have used all 11 Pareto approximate FAs, described in Section 3, to construct 8x8 approximate array multipliers, based on only one FA type per design to avoid the exponential growth of the design space. Regarding the degree of approximation, we have used two options: i) all FAs are approximate, and ii) FAs that contribute to the least significant 50% of the resultant bits are approximated in order

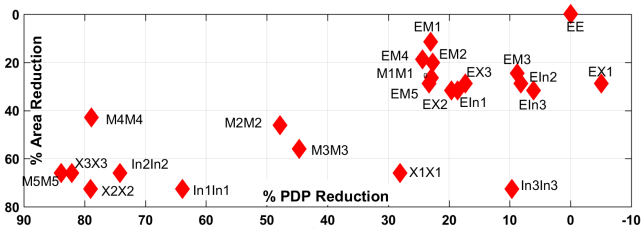

Figure 2: Area and PDP Reduction of 8x8 Array Multiplier

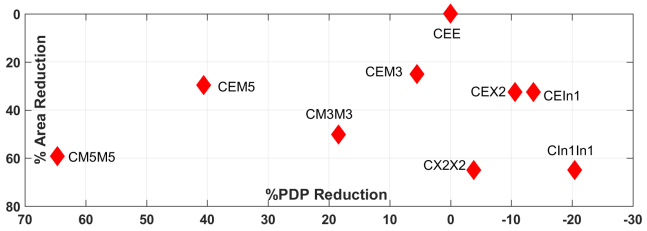

Figure 3: Area and PDP Reduction of 8x8 Tree Multiplier

to maintain acceptable accuracy as recommended by [4]. Thus, we have designed, evaluated and compared 22 different options for building 8x8 approximate array multipliers, using the TSMC65nm technology. Various tables showing the design characteristics for the considered approximate multipliers can be accessed from [8]. The name of the 8x8 array multiplier consists of two parts. For example, for the EM1 multiplier, the most significant part is based on an exact (E) adder and the least significant part is based on the mirror adder 1 (M1). Fully approximate multipliers have high NMED. The approximate multiplier size exhibits a linear relationship with the degree of approximation. There is no single design that is superior in all design metrics. Therefore, a Pareto-analysis for the improvements in area and PDP is shown for different proposed designs throughout this work.

Figure 2 shows the area and PDP reduction of 8x8 array multipliers. The best designs are located on the bottom left corner. *M5M5* is a Pareto-design with PDP reduction of 84% and area reduction of 65%. The design *X3X3* is non-Pareto because it has the same area reduction as the *M5M5* but with a smaller PDP reduction. However, we have to consider other *error metrics*. Some designs such as *EX1* have increased PDP due to excessive switching activity compared to the original design.

4.2 8x8 Tree Multiplier

The design space for approximate 8x8 tree multipliers is also quite large, depending on the *compressor type* and *approximation degree*. To avoid the exponential growth of the design space, we choose to use compressors of the same type. Also, we use two options for approximation degree: i) all compressors are approximate, and ii) compressors that contribute to the lowest significant 50% of the resultant bits are approximated to maintain an acceptable accuracy. Thus, based on the four shortlisted compressors, explained in Section 3, we compared 8 options for approximate 8x8 tree multipliers and the full results are given in [8]. The name of the multiplier consists of three parts. For example, CEM1 represents a compressor based multiplier (C), where the most significant part is based on an exact (E) compressor and the least significant part is composed of the mirror adder 1 (M1) based compressor. There is no single design superior is all metrics, but some designs are the best in some metrics. As depicted in Figure 3, the best designs are on the left bottom corner, i.e., *CM5M5* and *CX2X2* are Pareto designs while *CEM5* is a non-Pareto design.

5 HIGHER-ORDER MULTIPLIERS

The 8x8 multiplier basic modules can be used to construct higher-order target multiplier modules. In this paper, we use the example

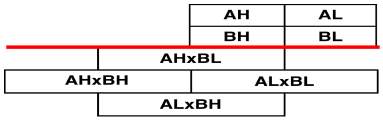

Figure 4: 16x16 Multiplier

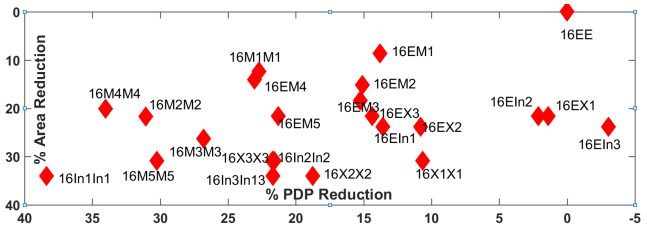

Figure 5: Area and PDP Reduction of 16x16 Array Multiplier

of designing a 16x16 multiplier to illustrate this process. The partial product tree of the 16x16 multiplication can be broken down into four products of 8x8 modules, which can be executed concurrently, as shown in Figure 4. We choose to design 16x16 multipliers with an exact AHxBH multiplier, and with exact MSBs and approximate LSBs for both AHxBL and ALxBH, and a fully approximate or approximate LSBs for ALxBL. Any other approximation degree can be found based on the required accuracy metric.

5.1 16x16 Array Multiplier

The simulation results for 16x16 approximate array multipliers, shows high similarities with the 8x8 version. The multiplier name is based on the type of ALxBL module. Fully approximate designs exhibit the minimal delay due to reduced circuit complexity. Generally, designs based on approximate mirror adders have the lowest power consumption, due to the elimination of static power dissipation. Since, the design size grows linearly with the FA size, fully approximate designs based on 6 transistors cells have the smallest area. As depicted in Figure 5 for area and PDP reduction, the best designs are on the lower left corner, i.e., *16In1In1* and *16In3In3* are Pareto designs while *16M4M4* is a non-Pareto design.

5.2 16x16 Tree Multiplier

The characterization of 16x16 and 8x8 approximate tree multipliers shows high similarities. As depicted in Figure 6, regarding area and PDP reduction, *16CEM5*, *16CEIn1* and *16CM5M5* are Pareto designs while *16CEM3* is a non-Pareto design.

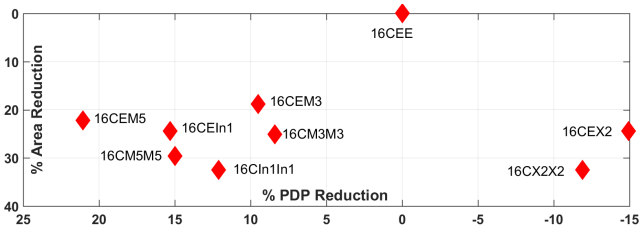

Figure 6: Area and PDP Reduction of 16x16 Tree Multiplier

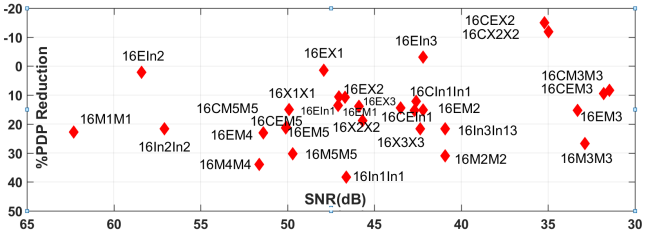

Figure 7: %PDP Reduction and SNR of Multipliers

5.3 Discussion and Comparison

The considered approximate multipliers are implemented using Cadence's Spectre based on TSMC65nm process, with V_{dd} = 1.0V at T=27C°. The circuit inputs are provided by independent voltage sources, and a load of 10fF is utilized. We evaluated and compared the design characteristics (Area, Power and Delay). We found out that the 8x8 exact tree multiplier exhibits lower delay, power and size compared to the 8x8 exact array multiplier.

Several multiplier designs, based on *AMA5*, have the lowest delay and power consumption, due to the basic structure of the FA cell, which is composed of two buffers only. Also, they have the lowest NMED and a small size. Regarding accuracy, the designs based on *InXA1* have low ER and NMED. Similarly, the designs based on the 6 transistors FA, have the minimal size. Thus, it can be observed that the characteristics of approximate FA are generally propagated in the corresponding approximate multipliers as well.

In terms of architecture, we found out that the tree multiplier designs tend to have a lower power consumption than the array multipliers, especially the designs based on low power consumption FAs, such as *AMA3* and *AMA5*. In terms of the 8x8 sub-module placement to form higher-order multipliers, with a fixed configuration for AHxBH, AHxBL and ALxBH, we have noticed that the quality-loss increases, while the size, power consumption and delay decrease for designs with fully approximate ALxBL.

Compared to the 24 different designs reported in [7], where 92% of the designs have ER close to 100%, only 80% of our proposed designs have high ER. Regarding NMED, almost all our designs have a value less than 10^{-5}, which is the minimum value reported by the 24 approximate designs in [7]. Comparing the PDP reduction, most of the designs in [7] have a high PDP reduction because they are based on truncation and a high degree of approximation. However, our designs are superior in PDP reduction for designs with a high degree of approximation.

6 APPLICATION

We evaluate and compare the accuracy of the built approximate multipliers based on an image blending application, where two images are multiplied pixel-by-pixel. While in previous sections, we used Cadence Spectre to build the circuits and evaluate their area, performance and power consumption, for experimentation purposes, here we use MATLAB to evaluate error metrics for an image processing application. The library of implemented cells and multiplier circuits, and the results of the image blending application can be found at *https://sourceforge.net/projects/approximatemultiplier*. The signal to noise ratio (SNR) is used to measure the image quality for different designs. Figure 7 shows a comparison of the SNR and the percentage of PDP reduction for different approximate multipliers. Clearly, designs on the bottom left corner, have the highest PDP reduction and the best quality (high SNR) [8]. Generally, all multiplier designs have an acceptable SNR (acceptable quality).

7 CONCLUSIONS

In this paper, we designed, evaluated and compared approximate multipliers, based on approximation in partial product summation. The design space of approximate multipliers is found to be primarily dependent on the type of the approximate FA used, the architecture, and the placement of 8x8 sub-modules in the higher-order nxn multipliers. The proposed designs are compared based on PDP, area, delay, power, ER and NMED. Various optimal designs have been identified in terms of the considered design metrics. An image blending application is used to compare the proposed multiplier designs in terms of SNR and PDP. Our designs show comparative results compared to 24 different approximate designs reported in [7]. In the future, we plan to investigate the design space of higher-order multiplier modules (e.g., 64x64) using the already considered metrics and configurations. Moreover, we also plan to evaluate the possibility of having mixed FAs in the 8x8 multiplier block.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *IEEE ETS*, 2013, pp. 1–6.
[2] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in *IEEE VLSI Design*, 2011, pp. 346–351.
[3] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *ACM GLSVLSI*, 2015, pp. 343–348.
[4] 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, 2013.
[5] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate xor/xnor-based adders for inexact computing," in *IEEE Nanotechnology*, 2013, pp. 690–693.
[6] H. A. F. Almurib, T. N. Kumar, and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in *IEEE DATE*, 2016, pp. 660–665.
[7] H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, and J. Han, "A comparative evaluation of approximate multipliers," in *IEEE Nanoscale Architectures*, 2016, pp. 191–196.
[8] M. Masadeh, O. Hasan, and S. Tahar, *Comparative Study of Approximate Multipliers*, Technical Report, ECE Department, Concordia University, Montreal, QC, Canada. http://arxiv.org/abs/1803.06587, 2018.
[9] Z. Yang, J. Yang, K. Xing, and G. Yang, "Approximate compressor based multiplier design methodology for error-resilient digital signal processing," in *IEEE ICSIP*, 2016, pp. 740–744.
[10] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, and J. Henkel, "Architectural-space exploration of approximate multipliers," in *ACM DAC*, 2016, pp. 1–8.
[11] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015.
[12] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Prentice-Hall, 2002.