

# A Fair Comparison of Adders in Stochastic Regime

Ardalan Najafi<sup>†</sup>, Moritz Weißbrich<sup>‡</sup>, Guillermo Payá Vayá<sup>‡</sup>, and Alberto Garcia-Ortiz<sup>†</sup>

<sup>†</sup>Institute of Electrodynamics and Microelectronics, Universität Bremen

Otto-Hahn-Allee 1, 28359 Bremen, Germany, Email: {ardalan, agarcia}@item.uni-bremen.de

<sup>‡</sup>Institute of Microelectronic Systems, Leibniz Universität Hannover

Appelstraße 4, 30167 Hannover, Germany, Email: {weissbrich, guipava}@ims.uni-hannover.de

**Abstract**—The demands of high-speed and power-efficient systems have resulted into the emergence of the approximate computing. Existing approximate circuits as well as stochastic techniques have shown promising advances in improving various figures of merit. However, a through fair comparison of arithmetic units still remains an issue which has not been studied. This paper reviews the prerequisites for a fair comparison of approximate arithmetic units. As one of the key components of arithmetic circuits, adders are the focus of this paper. For the first time in this paper, approximate and exact adders are studied together in the stochastic regime. Simulation results show that both the equal segmentation adder (ESA) and the error tolerant adder type II (ETAIL) outperform exact adders working stochastically, if and only if the right configuration and sub-adder architectures are chosen. Otherwise, there is no reason to use the aforementioned architectures. In all, considering the cost-error trade-off, Lower-part OR adder (LOA) has the best behavior in the stochastic regime.

## I. INTRODUCTION

Increasing vulnerability of computing systems to errors in underlying circuits is a growing concern nowadays. Variations in process, temperature, and power supply continue to strongly influence delay, and due to leakage dominance, their impact on power is increasing dramatically. Approximate computing has become a promising technique to reduce power, area and delay constraints in VLSI design. There are two methodologies for approximation: The first is the so-called stochastic computing [1], which proposes a new vision for energy and performance efficiency in which some errors are allowed, as long as they are corrected or tolerated by hardware, software, or the end user. Over-scaling technique is one of the most popular and widely used stochastic techniques. The second methodology approximates a system by redesigning a logic circuit. Although each of the methodologies has its own benefits and target applications, they provide a trade-off between error and efficiency. Hence, both techniques should be studied together, which *has not been done before*.

Adders, as one of the key components of arithmetic circuits, have attracted lots of researchers' attention in the field of approximate computing. Approximate adders have been proposed by truncating the carry propagation chain resulting in speculative adders. In the literature, there is lack of any study comparing approximate and stochastic adders. In other words, approximate and stochastic adders have not

been studied together. Owing to this fact, looking at the existing research works, it is not clear if an exact adder working stochastically performs better than approximate adders or not.

A thorough comparison of approximate adders is another missing part of the research area of approximate computing. Indeed, in [2]–[4], different approximate adders have been compared for their circuit characteristics and error values. However, each paper has its own deficiencies. In [2], which is the most complete existing comparison of approximate adders, the authors compared various approximate adders with different configurations. Nonetheless, the delay and power reports do not make sense in some cases. The most notable difference of our paper with the aforementioned paper is to include stochastic behavior, and considering different internal architectures. Authors in [3] evaluated different approximate adders to use in neuromorphic applications. In this paper, the error values have been reported without considering the Carry-out bit in the reference architecture. In [4], approximate adders operating as a recoding adder have been compared. Since the approximate adders have been simulated for a specific function, the simulation results cannot be compared with other research studies. As we discuss with more details later, the lack of fair and reproducible comparison of adders is slowing down the research in this field.

The paper is organized as follows: In Section II, the existing approximate adders are reviewed. Section III presents the necessary factors affecting the comparison of arithmetic units. In Section IV, the mentioned factors are evaluated using experimental results. A final comparison of the adders are also made considering accuracy versus cost of the adders. And finally, in Section V, the paper is concluded.

## II. BACKGROUND

Various topologies of parallel-prefix adders and their characteristics can be found in [5]. It is obvious that each of these architectures results in different trade-off of approximate adders when placed as their sub-adders. Using experimental results, it will be shown later in this paper that not only the topology of an approximate adder is important, but also the architecture used **inside** an approximate adder

plays a decisive role in its performance and error-cost trade-off.

Among the existing approximate adders [6]–[12], in this paper, we are comparing the combinational ones with better performance. More information can be found in [2]. In the rest of this section, approximate adders used for the comparison are introduced.

#### A. The Equal Segmentation Adder (ESA)

A segmented adder divides an  $n$ -bit adder into a number of smaller sub-adders which operate in parallel with fixed carry inputs. Let  $\mathbb{k} = \{k_1, k_2, \dots, k_s\}$  denote a vector including size of sub-adders, where  $s$  is the number of sub-adders. In Figure 1, a segmented adder divided into  $s$  sub-adders is shown, where  $k_1$  is the size of the first (the lowest significant) sub-adder,  $k_2$  is the size of the second sub-adder, and so on.

The equal segmentation adder (ESA) is a type of segmented adders with equally sized sub-adders, i.e.  $k_1 = k_2 = \dots = k_s$ . Conventionally, ESA is considered as an  $n$ -bit adder divided into  $\frac{n-l}{k}$  equally sized sub-adders in addition to the lowest significant sub-adder with the size  $l$ . Accordingly, the delay and the area of an ESA is dependent to the structure of the sub-adders.

The performance of an ESA is dependent on sub-adders' architecture(s). Although an ESA implemented using serial prefix algorithm is smaller and has less complexity than other algorithms, it has a relatively large delay. Based on the definitions presented in [5], the performance of approximate adders implemented using different prefix algorithms can be derived, which is out of the scope of this paper. Moreover, an ESA shows different behavior in the stochastic regime depending on the architectures used as its sub-adders. This is proved later in this paper, using experimental results.

#### B. The Error Tolerant Adder Type II (ETAI)

ETAI, proposed in [11], is an approximate adder based on segmented adders. It splits the entire carry propagation path into a number of short paths and completes the carry propagations in these short paths concurrently. Here, like the previous subsection, we first consider the general

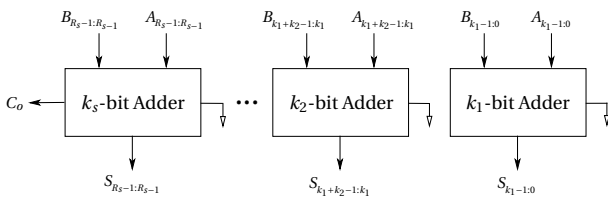


Fig. 1. The structure of the Segmented Adder,  $R_X = \sum_{i=1}^s k_i$

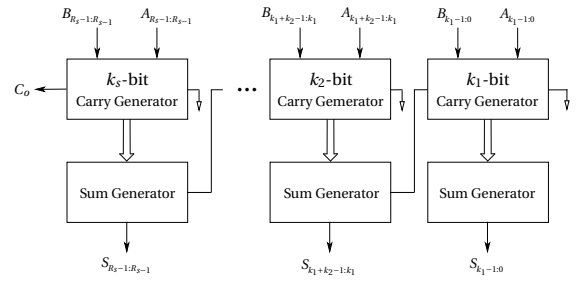


Fig. 2. The structure of the ETAII,  $R_X = \sum_{i=1}^s k_i$

case of the ETAII with arbitrary block sizes. As depicted in figure 2, the architecture of ETAII is divided into smaller blocks. Each block has an arbitrary number of bits and, different from ESA, consists of two separate circuitries - Carry Generator and Sum Generator. As the name implies, the Carry Generator creates the Carry-out signal. It does not take the carry signal from the previous block. The Sum Generator, however, takes the Carry-in signal from the previous block to generate its sum output bits. Consequently, the carry propagation only exists between two neighboring blocks instead of lying along the entire adder structure [11]. Conventionally, ETAII is divided into  $\frac{n-l}{k}$  equally sized blocks, in addition to the lowest significant block with the size  $l$ .

#### C. The Lower-part OR Adder

The Lower-part OR Adder (LOA) [12] divides an  $n$ -bit adder into two sub-adders. While the higher significant sub-adder is an  $(n - n_{or})$ -bit exact adder, the lower part sub-adder is simply constructed by  $n_{or}$  number of OR gates. To generate the Carry-in signal for the higher significant exact sub-adder, an extra AND gate is used which ANDs the most significant input bits of the lower significant sub-adder. The critical path delay of LOA then depends on the size of the exact sub-adder. The other figures of merit are also dependent on the exact sub-adder architecture. Figure 3 shows the topology of a LOA. As can be seen, an  $n$ -bit LOA exploits a regular smaller precise adder that computes the precise values of the  $(n - n_{or})$  most significant bits of the result along with OR gates that approximate the  $n_{or}$  least significant result bits by applying bitwise OR to the respective input bits.

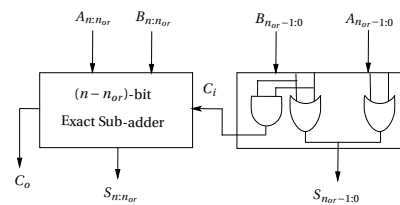


Fig. 3. The hardware structure of LOA

### III. RULES FOR FAIR COMPARISON

Choosing the best architecture is always made by considering the requirements for the specific application. In order to compare arithmetic units, all the factors affecting the result of the comparison should be clearly defined and specified. However, in the literature the definition has not been done properly. Due to the lack of unique and clear definitions, comparing different papers' results is almost impossible. In this section, the rules for a fair comparison are listed and the importance of proper specification of each factor is discussed.

#### A. Reference Architecture

The first important factor impacting the comparison of arithmetic units is the specification of the executable reference architecture. This obvious observation has frequently been violated in the literature. Since the focus of this paper is on adder structures, the importance of the reference architecture is discussed for adders.

Regarding adder structures, like every other arithmetic unit, it is extremely important to precisely specify inputs and outputs; and in order to have an equitable comparison it should be consistent for all the architectures. For the first time in this paper, the impact of considering Carry-out in the calculations on the error values of the approximate adders is studied. Taking Carry-in as a uniformly distributed random bit does not make a considerable difference in comparison with a fixed input carry. As a result, most of the time, the consideration is a fixed input carry, i.e.  $C_{in} = 0$ . However, in some literature [3], [4], error values have been reported without considering  $C_o$  as part of the adders' outputs, without mentioning how it can affect the error values. It will be shown in the next section that error values for the same architectures can be two times bigger when  $C_o$  is excluded from the reference architecture.

#### B. Metrics

Metrics are defined based on the target applications. However, different conclusions might be come out of different metric choices. Dealing with approximate adders as well as working in the stochastic regime, the most decisive metric to choose an adder is the error metric. Existing research works have used various error metrics. The authors in [13] defined the Error Distance and the *Mean Error Distance* (MED) to evaluate the arithmetic performance of approximate circuits. These metrics, based on the definitions, are absolute error and *Mean Absolute Error* (MAE), respectively. In [14], MED, and error rate have been chosen to show the error characteristics of approximate adders. In [4], to compare approximate adders as well as approximate multiplier designs, authors made use of MED and Pass Rate metrics. In [2], [15], [16], relative error metrics have been used to evaluate designs. In this paper, MAE and *Mean*

*Squared Error* (MSE) are used to compare different adder structures.

Since more accuracy cannot be gained without an increase in the silicon area and/or power consumption, comparing approximate adders without considering their costs seems to be unfair. This is even more important working in the stochastic regime. It will be shown in the next section how cost if defined can change the superiority of an adder over the others. In addition, depending on which cost metric (either Area-Delay Product (ADP), Power-delay Product (PDP), or Power-Delay-Area Product (PDAP)) is chosen, different understandings can be concluded.

#### C. Internal Architectures

Another important factor affecting approximate adders' behavior in the stochastic regime is the architectures used as their sub-adders. Although changing the exact architectures used as sub-adders does not change the approximate adders' error values, it significantly affects the stochastic behavior of the adders. Consequently, in order to compare adders, the internal architectures should be specified. In [2]–[4], the costs of the approximate adders have been compared without indicating the architectures used inside the adders. In the next section, by changing constraints in the synthesis tool, which results to different netlists and corresponds to change in internal architectures, we show the impact of different internal architectures on stochastic behavior of approximate adders.

#### D. Possible Configurations

Obviously, different approximate adders' configurations do not show similar behavior in the stochastic regime, which should be taken into account. As discussed in the previous section, approximate adders like ESA and ETAII can be divided into equal or non-equal segments. Since the configuration of approximate adders is a knob to improve them, it should be mentioned which configuration has been chosen. In the next section, the impact of different

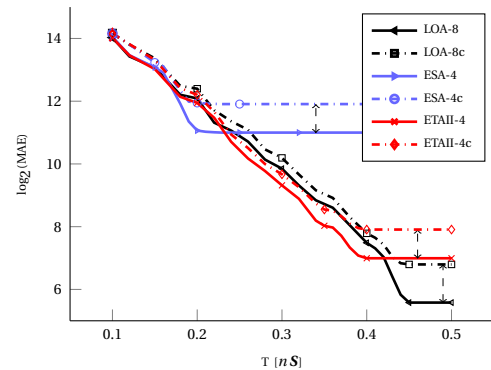


Fig. 4. Impact of excluding  $C_o$  from the reference architecture - 'c' indicates the results *without* considering  $C_o$ .

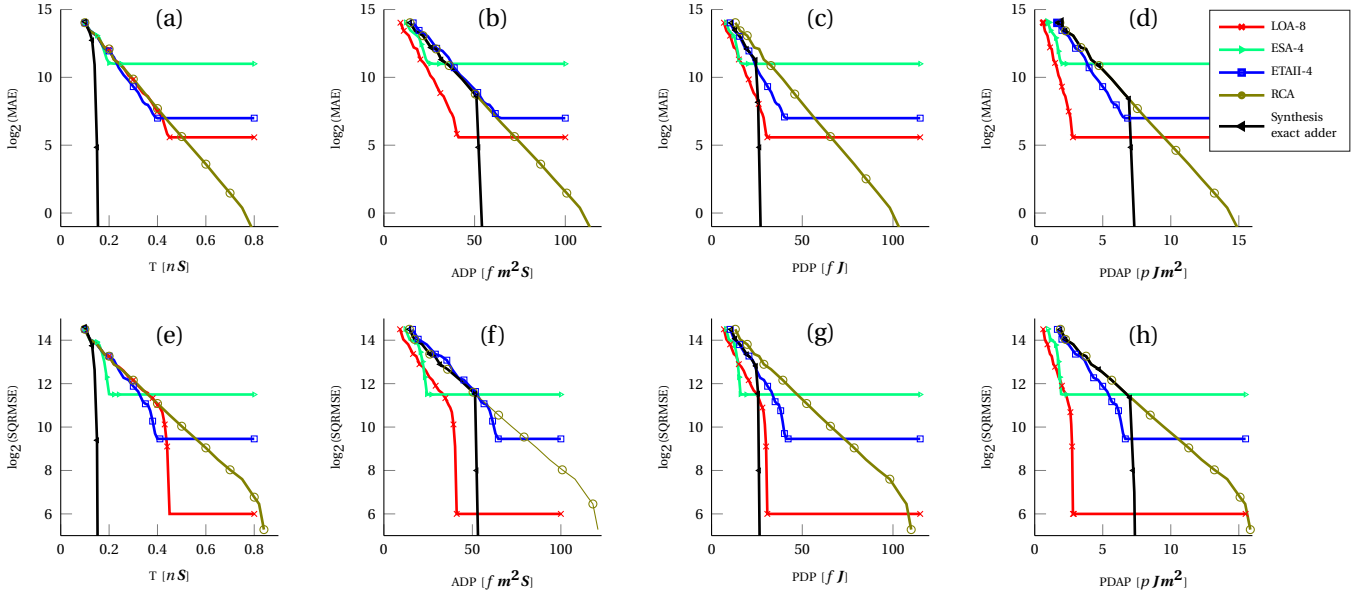


Fig. 5. (a)-(d) Mean absolute error, (e)-(h) Square root of mean squared error, of the adders vs. different figures of merit.

configurations on behavior of approximate adders in the stochastic regime is shown using experimental results.

#### IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

To assess the circuit characteristics and evaluate the claims stated in the previous section, we have generated VHDL description of the adders. Different configurations of these adders are synthesized in a commercial 65 nm library, for 16-bit operands. Using back-annotated simulations, dynamic power dissipation of the adders are evaluated after synthesis. All the adders have been simulated for  $10^7$  uniformly distributed random input patterns. Using frequency over-scaling different approximate and exact adders have been compared. In this section, each adder's name is followed by one number. For ESA and ETAII, this number is the size of the equal segments,  $k$ . Regarding LOA, the number is the size of the lower significant sub-adder; i.e. the number of OR gates. In the cases where more than one number follow ETAII and ESAs' names, they show segment sizes from the lowest to the highest significant, from left to right, respectively. Although in these cases the adder is not an *equal* segmented adder anymore, in order to prevent any confusion, we still call it ESA. For instance, ESA-3445 is a 16-bit segmented adder with the lowest significant sub-adder of size 3, and highest significant sub-adder of size 5.

As mentioned in the previous section, considering or not considering  $C_o$  in the error calculation affects the error values of the adder. The error values of approximate adders working with nominal frequencies are tabulated in table I. Different configurations of the approximate adders have been simulated and the error values are shown in the table.

The error values for the same architectures are also included in the table for the case in which  $C_o$  is not considered in the calculations. Therefore, *with  $C_o$*  and *without  $C_o$*  in the first and second row of the table indicate the calculations with and without considering Carry-out in the outputs of the reference adder architecture, respectively.

As can be seen in table I, the error values for ESA and ETAII can be up to 2 times higher when Carry-out is not considered in calculations. This number is even bigger for LOA, i.e. nearly 2.3x higher error values.

Figure 4 shows graphically the impact of  $C_o$  on the error values of the approximate adders working in the stochastic regime. Although at higher frequencies the effect is less relevant, it can also be seen in the figure that without  $C_o$ , the error values can considerably increase. As a result, when reporting error values of the adders, it should be specified whether  $C_o$  has been considered in the calculations or not. Nevertheless, since the normal case is considering  $C_o$  as part of the outputs of the adder, when it is not considered, must be stated.

In order to show the impact of considering different metrics on the comparison of the adders, we compare four adders using frequency over-scaling. Figure 5 shows how metrics can affect the comparison. Figures 5(a)-(d) show mean absolute error of the adders versus their efficiencies, while the same graphs for square root of mean squared error (SQRMSE) of the adders are depicted in figures 5(e)-(h). As can be seen, considering MSE, in some periods, ESA and ETAII, due to their lower error values, show superiority. Nevertheless, since the results of MAE and MSE do not show a considerable difference, for the rest of this paper, MAE is the chosen error metric to show the accuracy of the adders.

TABLE I  
MEAN ABSOLUTE ERROR - IMPACT OF  $C_o$  ON ERROR VALUES

|               | ESA-4   | ESA-3445 | ESA-5   | ESA-6   | ETAI-4 | ETAI-3445 | ETAI-5 | ETAI-6 | LOA-6 | LOA-8  |
|---------------|---------|----------|---------|---------|--------|-----------|--------|--------|-------|--------|
| with $C_o$    | 2046.79 | 1022.89  | 1022.83 | 511.36  | 127.42 | 63.53     | 31.6   | 7.52   | 11.87 | 47.85  |
| without $C_o$ | 3843.47 | 1981.51  | 1980.56 | 1005.89 | 240.25 | 123.19    | 61.43  | 14.94  | 27.79 | 111.16 |

As depicted in figure 5, when error values of the over-scaled adders are measured without considering the cost of the adders, figure 5(a) and 5(e), there is no reason to use LOA in the stochastic regime. While the cases in which error values are shown versus the cost of the adders, LOA outperforms the other approximate adders in the stochastic regime.

Indeed, depending on which figures of merit are of interest, figures 5(b)-(d) are taken into consideration. In all, taking the cost into account, LOA shows the best behavior in the stochastic regime, with the current configurations. As can be seen in graph (b), when the goal is to have optimized area, there is no reason to use ETAI-4, and the exact adder working stochastically has less error for every given area-delay product. Note that, here, all the approximate adders use serial prefix algorithm in their sub-adders. In addition to that, conventional configurations of the approximate adders have been chosen to show the effect of metrics in figure 5.

As discussed in the previous section, internal architectures of arithmetic units play an important role in their behavior in the stochastic regime. Figure 6 shows how approximate adders can behave differently using different sub-adders. Indeed, here, the approximate adders are synthesized using different timing constraints which corresponds to different sub-adder architectures. As can be seen in the figure, changing internal architectures of the approximate adders does not change the error values, but makes a change in the slopes in the stochastic regime. It should be taken into consideration that even a different setting of the synthesizer can affect the results. This is a powerful motivator to have a framework for a fair and

reproducible comparison.

Another important factor which should be specified is the configuration of the approximate adders. Figure 7 shows the stochastic behavior of the approximate adders for different configurations. As can be seen in the figure, in the nominal frequency, ETAI-5 and ETAI-6 have lower error values than LOA-8. As a result, it cannot be claimed that with absolute certainty LOA outperforms ETAI, or vice-versa, and the configuration has a significant effect. As depicted in figure 7, ETAI-5 and ETAI-6 outperform LOA-8 as long as working with nominal frequencies or less. However, working stochastically in an over-scaled frequency, LOA-8 outperforms ETAIs.

Considering all the aforementioned factors, we compare approximate and exact adders in the stochastic regime. Figure 8 shows the comparison of the approximate and exact adders in the stochastic regime. The best gate-level netlist, for each adder, has been generated by the synthesizer applying the same constraints for all the adders. As can be seen in the figure, the exact adder is the superior design to the point that makes no errors. Whereas in the stochastic regime, as soon as the exact adder starts making errors, LOA outperforms all the other adders. As depicted in figure 8(a), where the error values are shown versus PDP, for higher frequencies, ESA outperforms the other adders. However, the error values are big enough to claim that it does not makes sense to make use of ESA stochastically.

Due to the fact that LOA is of a different category of approximate adders, some researchers exclude it from their comparison. In this case, if we do not consider LOA, based on the graphs depicted in figure 5, there is no reason to

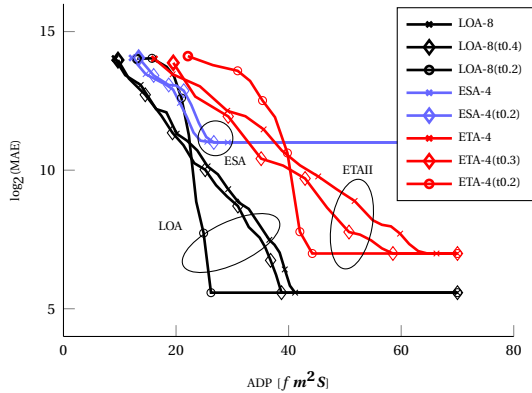


Fig. 6. A comparison of approximate adders with different sub-adder architectures - the number following each 't' shows the timing constraint.

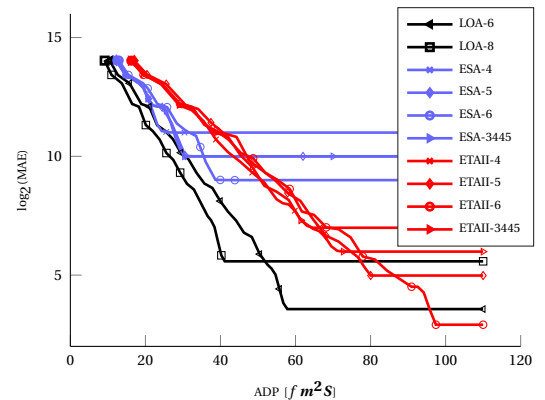


Fig. 7. A comparison of approximate adders with various configurations

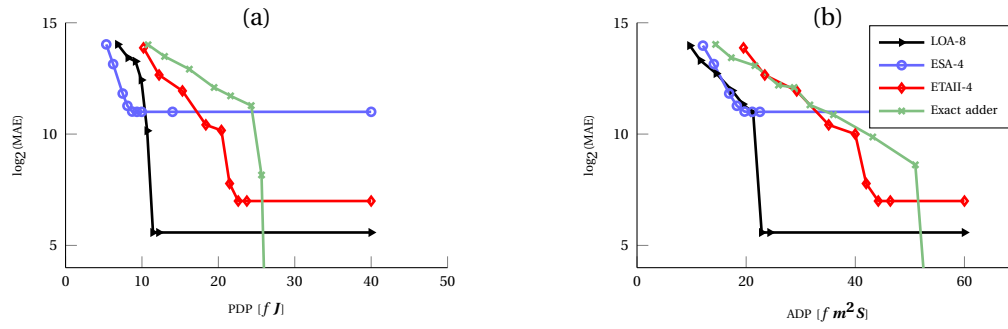


Fig. 8. Comparison of optimized adder structures in the stochastic regime; (a) MAE vs. PDP , (b) MAE vs. ADP

use approximate adders. Seeing that, for any given error value, an exact adder can be found working stochastically outperform both ETAII and ESA architectures. Nonetheless, as depicted in figure 8, if the right configuration and sub-adder architectures be chosen for ETAII and ESA, they outperform the exact adder in the stochastic regime. As can be seen in figure 8, since efficient architectures are used as sub-adders, in a big range, ETAII-4 working stochastically outperforms the efficient exact adder.

## V. CONCLUSION

In this paper, the error behavior of approximate and exact adders in the stochastic regime has been evaluated. A fair and reproducible comparison of the adders has been provided considering the cost of the adders. Using experimental results, the impact of different reference architectures, metrics, internal architectures, and configurations have been evaluated. It is concluded that, once the exact adder starts making errors, LOA outperforms all the other architectures. Putting LOA aside, there is no reason to use approximate adders, while exact adders working stochastically perform better than approximate adders taking cost and accuracy into consideration. However, if the configuration and the sub-adder architectures of approximate adders are chosen appropriately, they can outperform exact adders in the stochastic regime.

## ACKNOWLEDGMENT

This work is funded by the German Research Foundation (DFG) project GA 763/4-1.

## REFERENCES

- [1] J. Sartori and R. Kumar, "Stochastic computing," *Found. Trends Electron. Des. Autom.*, vol. 5, no. 3, pp. 153–210, Mar. 2011.
- [2] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*, ser. GLSVLSI '15. New York, NY, USA: ACM, 2015, pp. 343–348.
- [3] Y. Kim, Y. Zhang, and P. Li, "Energy efficient approximate arithmetic for error resilient neuromorphic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2733–2737, Nov 2015.
- [4] H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 booth multipliers for low-power and high-performance operation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2638–2644, Aug 2016.
- [5] R. Zimmermann, "Binary adder architectures for cell-based vlsi and their synthesis," Ph.D. dissertation, Swiss Federal Institute of Technology (ETH) Zurich, Hartung-Gorre Verlag, 1998.
- [6] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.
- [7] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2012, pp. 1257–1262.
- [8] 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.
- [9] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proceedings of the 49th Annual Design Automation Conference*, ser. DAC '12. New York, NY, USA: ACM, 2012, pp. 820–825.
- [10] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.
- [11] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Proceedings of the 2009 12th International Symposium on Integrated Circuits*, Dec 2009, pp. 69–72.
- [12] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, April 2010.
- [13] 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, Sept 2013.
- [14] C. Liu, J. Han, and F. Lombardi, "An analytical framework for evaluating the error characteristics of approximate adders," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1268–1281, May 2015.
- [15] J. Schlachter, V. Camus, and C. Enz, "Near/sub-threshold circuits and approximate computing: The perfect combination for ultra-low-power systems," in *2015 IEEE Computer Society Annual Symposium on VLSI*, July 2015, pp. 476–480.
- [16] J. Schlachter, V. Camus, K. V. Palem, and C. Enz, "Design and applications of approximate circuits by gate-level pruning," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1694–1702, May 2017.