

Systematic Design of an Approximate Adder: The Optimized Lower Part Constant-OR Adder

Ayad Dalloo¹, Ardalan Najafi¹, and Alberto Garcia-Ortiz²

Abstract—Exploiting the tradeoff between accuracy and hardware cost has a tremendous potential to improve the efficiency of integrated systems. Using this concept, numerous approximate adders have been proposed in the last ten years. Although conceptually different, all previous architectures have been obtained with an *ad hoc* and nonsystematic methodology. Instead, this brief generalizes and systematically optimizes an architectural template for approximate adders. The outcome, called optimized lower part constant-OR adder (LOCA), outperforms previous approaches in terms of accuracy and hardware cost. For example, an 8-bit approximate adder implemented with our new approach improves the mean squared error by 58.5%, while simultaneously reducing the cost by 7.2% with respect to the previously reported best architecture.

Index Terms—Adder architecture, approximate computing, error-cost tradeoff, stochastic computing.

I. INTRODUCTION

Stochastic computing has begun to emerge in response to the languishing benefits of technology scaling. Rather than hiding variations under expensive guard bands, designers have begun to relax traditional correctness constraints and deliberately expose hardware variability to higher levels of the computing stack [1]. Approximate computing, a promising technique to reduce power, area, and delay in VLSI design, approximates a system by redesigning its logic circuit [2]. It exploits the gap between the level of accuracy required by the applications and that provided by the computing system, for achieving diverse optimizations.

The researchers in the field of approximate computing have paid special attention to adders, one of the key components of arithmetic circuits. In fact, a surprisingly large number of approximate adders [3]–[10] have been proposed in the literature: segmented adders where an n -bit adder is divided into k -bit subadders [3]–[5]; carry select adders in which multiple submodules are used [6], [7]; approximate full adders where the full adder is approximated [9], [10]; and speculative adders which are built upon the observation that the critical path is rarely activated in traditional adders [11]–[13]. The current situation is such, that even a fair comparison of approximate adders is a challenging endeavor [14], [15]. Although all the architectures are conceptually different, they share a common characteristic: they have been obtained with an *ad hoc* and nonsystematic methodology. A remarkable exception is the generic accuracy configurable adder (GeAr) that uses the idea of template [8] but is not optimal.

Among all the purely combinatorial approximate adders, the lower part OR adder (LOA) [9] shows the best error versus hardware cost

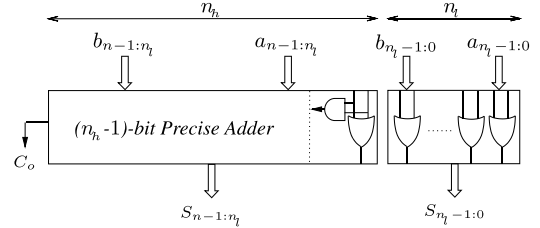


Fig. 1. Hardware architecture of the LOA.

tradeoff [14], [15]. As can be seen in Fig. 1, LOA [9] divides an n -bit adder into two subadders. While the higher significant subadder consists of an $(n_h - 1)$ -bit exact adder, the lower part subadder is simply constructed by n_l OR gates (bits 0 to $n_l - 1$). To generate the carry-in signal for the accurate adder, an extra AND gate is used which combines the adder inputs of bit position n_l , i.e., a_{n_l} and b_{n_l} . The key advantage of LOA with respect to other architectures as equal segmentation adder (ESA) [4], error tolerant adder (ETAII) [5], or almost correct adder [3] is that the approximation is restricted to the least significant bits, and therefore, the magnitude of the errors is limited.

The goal of this brief is to improve LOA systematically. First, we generalize the LOA architecture in the form of an architectural template; then, studying all the possible choices to implement that template, we obtain an optimal architecture for the presented template focusing on mean squared error (MSE). We call it optimized LOCA (OLOCA). Since LOA is the superior adder among the existing approximate adders, our optimized architecture outperforms all the existing approximate adders when considering the tradeoff between hardware cost and accuracy. The experimental evidence reported in this brief corroborates this fact.

Following the aforementioned goals, this brief is organized as follows. Section II describes the structures of the architectural template and of OLOCA. Afterward, in Section III, we quantify the advantages of OLOCA using experimental results; furthermore, we validate the mathematical formulas developed in Section II. Finally, Section IV concludes this brief.

II. ARCHITECTURE

To obtain systematically an optimal¹ approximate adder, we progress in three steps: 1) We describe the error metrics and hardware cost quantifying the quality of the architecture; 2) we generalize the architecture of LOA into a more abstract template; and 3) we optimize the template, regarding MSE, to produce OLOCA.

A. Metrics

Different metrics need to be considered to evaluate the quality of approximate adders; they quantify the tradeoff between error and hardware cost.

The error is defined as the difference between the approximate and accurate output results of the adder, that is,

$$\varepsilon = \tilde{S} - S \quad (1)$$

¹Throughout this brief, “optimal” refers to “optimal for the given template.”

Manuscript received August 29, 2017; revised November 26, 2017 and February 1, 2018; accepted March 13, 2018. This work was supported by the German Research Foundation under Project GA 763/4-1. (All authors contributed equally to this work.) (Corresponding author: Alberto Garcia-Ortiz.)

The authors are with the Institute of Electrodynamics and Microelectronics, University of Bremen, 28359 Bremen, Germany (e-mail: ardalan@item.uni-bremen.de; agarcia@item.uni-bremen.de).

Color versions of one or more of the figures in this brief are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2018.2822278

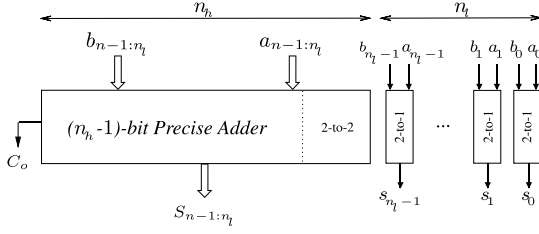


Fig. 2. Hardware structure of the general template.

TABLE I
ERROR METRICS AND UNIT GATE CHARACTERISTICS
OF THE POSSIBILITIES FOR 2-TO-1 BLOCKS

	$\hat{\mu}$	$\hat{\sigma}^2$	MSE	\hat{A}	\hat{D}
AND	$-3/4$	$3/16$	$3/4$	1	1
OR	$-1/4$	$3/16$	$1/4$	1	1
Buffer	$-1/2$	$1/4$	$1/2$	0	0
Cte-0	-1	$1/2$	$3/2$	0	0
Cte-1	0	$1/2$	$1/2$	0	0

where \tilde{S} is the approximate (erroneous) output of the adder and S is the accurate result. The magnitude of the error can be quantified with several metrics; among them, the most common ones are the average error (μ), the standard deviation (STD or σ), the MSE, and the mean absolute error (MAE). They can be calculated as

$$\mu = E[\varepsilon] \quad (2)$$

$$\sigma = \sqrt{E[(\varepsilon - \mu)^2]} \quad (3)$$

$$MSE = E[\varepsilon^2] = \mu^2 + \sigma^2 \quad (4)$$

$$MAE = E[|\varepsilon|] \quad (5)$$

where E is the expectation operator. It should be mentioned that it is also common to employ the normalized version of the previous metrics dividing them by the range of the adder, i.e., 2^n .

In order to evaluate the hardware efficiency of the architectures, the area and delay of the designs need to be considered. In the rest of this brief, A and D denote the hardware area and delay, respectively. In the mathematical analysis, we use the unit-gate model [16] where simple monotonic two-input gates (AND, OR, NAND, and so on) have a cost of one in area and delay, and simple nonmonotonic two-input gate (XOR and XNOR) have a cost of two in area and delay. Obviously, in the experimental results, the actual area and delay of the circuit are considered.

B. General Template Architecture Based on LOA

As discussed in Section I, considering the error versus hardware cost tradeoff, experimental results show that LOA is the best architecture among all the existing approximate adders [14], [15]. Studying LOA's architecture carefully, it can be generalized as Fig. 2: the lower significant subadder can be divided into n_l 2-to-1 logic blocks (bits 0 to n_l-1), and a single 2-to-2 logic block. This later block receives the inputs of the adder in bit position n_l to generate the input carry for the exact part using an AND gate, and its sum signal can be generated inexactly. Finally, the higher significant subadder is an exact adder. Clearly, the architecture of LOA can be described by taking the proposed general template, putting OR gates in each bit of the lower significant subadder, and replacing the first bit of the higher significant subadder with approximate circuitry of OR_AND.

In principle, any Boolean function with the right size can provide a choice for the blocks. Note that even a constant function equal to one (Cte-1) or zero (Cte-0) is a valid selection. For concreteness,

TABLE II
ERROR METRICS AND UNIT GATE CHARACTERISTICS
OF THE POSSIBILITIES FOR THE 2-TO-2 BLOCKS

	$\hat{\mu}$	$\hat{\sigma}^2$	MSE	\hat{A}	\hat{D}
Half-Adder	0	0	0	3	2(1)
OR_AND	$1/4$	$3/16$	$1/4$	2	1(1)
Cte-1_AND	$1/2$	$1/4$	$1/2$	1	0(1)
Buffer_AND	0	$1/2$	$1/2$	1	0(1)

TABLE III
FORMULAS OF ERROR METRICS, AREA, AND DELAY

	LOA	OLOCA
μ	$\frac{1}{4}$	$\frac{-3}{16} 2^{n_l}$
σ^2	$\frac{1}{4} 4^{n_l} - \frac{1}{16}$	$\frac{53}{768} 4^{n_l} - \frac{1}{6}$
MSE	$\frac{1}{4} 4^{n_l}$	$\frac{5}{48} 4^{n_l} - \frac{1}{6}$
MAE	$\frac{3}{8} 2^{n_l} - \frac{3}{8}$	$\frac{15}{64} 2^{n_l} - \frac{3}{4} 2^{-n_l}$
A	$(n_h - 1) \cdot A_{FA} + A_{AND}$ + $(n_l + 1) \cdot A_{OR}$	$(n_h - 1) \cdot A_{FA} + A_{HA}$ + $(n_l - n_{cte}) \cdot A_{OR}$
D	$(n_h - 1) \cdot t_c + T_{AND}$	$(n_h - 1) \cdot t_c + T_{AND}$

the relevant choices for 2-to-1 and 2-to-2 blocks are tabulated in Tables I and II, respectively. Although we have studied all the possibilities, the blocks with higher error values for the same cost have been eliminated from Tables I and II. In order to have an *optimal* architecture for the template, the best combination of blocks from each table should be chosen. For uniform distributed data, each bit is uncorrelated and the error metrics of the template (T) can be calculated as a function of the error characteristics of each block. Since the total error, ε_T , is the summation of the errors of each block, $\hat{\varepsilon}_i$, with the corresponding weight, i.e., $\varepsilon_T = \sum_{i=0}^{n_l} \hat{\varepsilon}_i 2^i$, we obtain

$$\mu_T = \sum_{i=0}^{n_l} \hat{\mu}_i 2^i \quad (6)$$

$$\sigma_T^2 = \sum_{i=0}^{n_l} \hat{\sigma}_i^2 2^{2i} \quad (7)$$

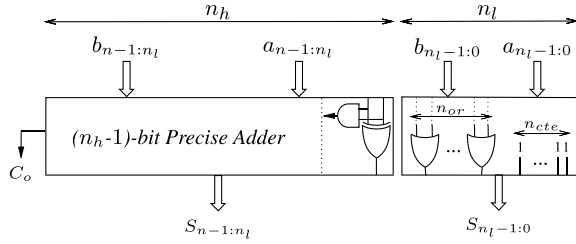
$$MSE_T = \sum_{i=0}^{n_l} \hat{\sigma}_i^2 2^{2i} + \left(\sum_{i=0}^{n_l} \hat{\mu}_i 2^i \right)^2 \quad (8)$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i^2$ are the average error and the variance of error associated with the instantiated block in bit position i . The corresponding values are given in Table I for bits 0 to n_l-1 and in Table II for the bit n_l , under the column names $\hat{\mu}$ and $\hat{\sigma}^2$, respectively. For example, using this method, we obtained the error metrics for LOA given in Table III which agree with the simulation results of [15]. The key question, now, is whether the particular choices made by LOA are optimal, and if not, which is the optimal alternative for the selected template. Section II-C addresses this topic.

C. Optimized Architecture

Depending on the error metrics which are chosen, different optimization results might be obtained. Illustratively, here, we choose the MSE as the error metric because of its relevance in data processing applications. In order to obtain the optimal architecture out of the general template, we need to evaluate all the possible combinations of 2-to-1 and 2-to-2 logic blocks of Tables I and II. Let us proceed first intuitively and then more formally.

The errors in the upper bits have a higher weight than in the lower ones [see (8)]. Thus, it is more profitable to expend resources in the 2-to-2 block than in the lower 2-to-1 blocks. The best

Fig. 3. Structure of LOCA; $n_l = n_{cte} + n_{or}$.

2-to-2 blocks are the OR_AND and half adder. Replacing the half adder with an OR_AND does not improve the delay and improves the area only marginally; the penalty is a large increase in the MSE. For this reason, the idea of LOA (to use the OR_AND for the 2-to-2 block) is not efficient. Once we fix the 2-to-2 block to a half adder, we can observe that the average error introduced by the 2-to-2 block is zero or positive, while the 2-to-1 blocks introduce a zero or negative average error. Thus, it is only useful to use blocks with small $\hat{\mu}$ (the Cte-1) or small $\hat{\sigma}$ (the OR). Therefore, the optimal disposition of 2-to-1 blocks should be OR blocks followed by Cte-1 blocks in the lower bits where the errors are less relevant. Since the adder is constructed using Cte-1s and OR gates, we call it LOCA. The structure of LOCA is depicted in Fig. 3 and its error metrics can be expressed as follows:

$$\mu_{LOCA} = 2^{n_{cte}-2} - 2^{n_l-2} \quad (9)$$

$$\sigma_{LOCA}^2 = 2^{2n_l-4} + \frac{5}{3}2^{2n_{cte}-4} - \frac{1}{6} \quad (10)$$

$$MAE_{LOCA} = 2^{n_l-2} - 2^{n_{cte}-2} \quad (11)$$

$$+ \frac{1}{3} \left(\frac{3}{4} \right)^{n_l-n_{cte}} \left(2^{n_{cte}} - \frac{1}{2^{n_{cte}}} \right) \quad (12)$$

$$MSE_{LOCA} = \frac{1}{6}2^{2n_l-2n_{or}} + 2^{2n_l-3} - 2^{2n_l-n_{or}-3} - \frac{1}{6}. \quad (13)$$

To determine the optimal number of OR gates, we can minimize (13) versus n_{or} , resulting the optimal value in $n_{or} = \log_2(8/3)$. The closest integer numbers, $n_{or} = 1$ and $n_{or} = 2$, produce the same MSE and are optimal. We prefer $n_{or} = 2$ that provides a better STD. We call this architecture OLOCA. Although remarkably simple, it outperforms LOA regarding STD, MSE, and MAE.

The error metrics, area, and delay of LOA and OLOCA are tabulated in Table III. Those formulas provide a better understanding of the architectures and make the comparison easier. As can be seen in Table III, the average error of OLOCA is slightly larger than that of LOA's, while its STD is much smaller. Hence, the MSE of OLOCA is almost $2.4\times$ smaller than the MSE of LOA, which represents a considerable improvement for practical circuits. Regarding the MAE, LOA has a $1.6\times$ larger error with respect to OLOCA. Although OLOCA does not improve the delay over LOA, its silicon area is clearly smaller (for $n_l > 2$).

It is also possible to obtain the optimal architecture out of the general template more rigorously. First, let us observe that (8) and Table II imply an architecture, where the 2-to-2 block is not a half adder, has necessarily an MSE of at least $MSE_T \geq \hat{\sigma}_{n_l}^2 4^{n_l} \geq (3/16)4^{n_l}$, which is worse than the MSE of OLOCA (see Table III). Thus, the 2-to-2 block has to be a half adder in the optimal architecture.

In order to demonstrate that the selection of 2-to-1 blocks of OLOCA is optimal in terms of MSE for the given template (Fig. 2), we can proceed by induction, using n_l as the induction variable. A simple computation of all the possibilities, using (8), shows that OLOCA is indeed optimal for $n_l = 1$, $n_l = 2$, and $n_l = 3$. Let

us analyze an architecture with $n_l = K$, assuming the optimality of OLOCA for an architecture with $n_l = K - 1$. Observe that the total error, ε_T , can be decomposed into the independent contributions of the block in bit position 0, $\hat{\varepsilon}_0$, and the remaining blocks, ε_{MSBs} . Since $\varepsilon_T = \varepsilon_{MSBs} + \hat{\varepsilon}_0$, the MSE_T can be expressed as a function of the statistical characteristics of $\hat{\varepsilon}_0$ and ε_{MSBs} ; more precisely

$$MSE_T = MSE_{MSBs} + MSE_0 + 2\hat{\mu}_0\mu_{MSBs} \quad (14)$$

where μ_{MSBs} and MSE_{MSBs} can be calculated using (6) and (8), respectively, iterating i from 1 to K .

Note that the optimization of block 0 and the remaining $K - 1$ blocks are not independent due to the term $2\hat{\mu}_0\mu_{MSBs}$. However, if we prove that block 0 is a Cte-1 in the optimal architecture, then it follows that $\hat{\mu}_0 = 0$ and the term $2\hat{\mu}_0\mu_{MSBs}$ disappears. In this case, the optimization of MSE_{MSBs} , consisting of $K - 1$ blocks, yields an OLOCA architecture by the induction hypothesis.

Let us show that the block in bit position 0 has to be Cte-1 (for $K \geq 3$) in order to have the optimal architecture regarding MSE. First, note that the alternative of choosing Cte-1 for the blocks 1 to $K-1$, which produces $MSE_{MSBs} = (1/6)4^K - (2/3)$ and $\mu_{MSBs} = 0$, is suboptimal. This is due to the fact that the resulting MSE_T , which is greater than or equal to MSE_{MSBs} , is worst than that of OLOCA. As a result, at least one of the blocks should not be a Cte-1 in the optimal architecture. For any of those cases, $\mu_{MSBs} = \sum_{i=1}^{K-1} \hat{\mu}_i 2^i \leq (-1/2)$, because each term in the addition is strictly negative. A simple calculation of the MSE for each of five possibilities for the block number 0, using (14), shows that a Cte-1 is the optimal selection when $\mu_{MSBs} \leq (-1/2)$. This observation concludes the proof.

III. EXPERIMENTAL RESULTS

To assess the circuit characteristics and evaluate the presented architectures in Section II, we have generated VHDL description of the adders. Different configurations of these adders are synthesized in a commercial low-power 65-nm library, for 16- and 8-bit operands. Using back-annotated simulations, dynamic power dissipation of the adders is evaluated after synthesis for the frequency = 1 GHz. Ripple carry adders are used as the subadders of all the approximate adders. All the adders have been simulated for 10^7 uniformly distributed random input patterns. 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. Regarding LOA and OLOCA, the number is the size of the lower significant subadder; i.e., n_l .

In order to check the accuracy of the formulas, as well as comparing the adder architectures, the error versus cost of the adders for different values of n_l s are depicted in Fig. 4. MAE and MSE versus area-delay product (ADP) of the 16-bit adders are shown in two graphs. LOCA has been simulated for different number of constants in each n_l case. As can be seen in the graphs, replacing OR gates with Cte-1s decreases the MAE and MSE values and at the same time the ADP; the trend continues until the point where two OR gates remain. After that point, the error values start increasing while the cost of the adder decreases. As a result, the optimal architecture, considering the error cost tradeoff, is obtained keeping two OR gates and replacing Cte-1s for the rest of 2-to-1 blocks. This verifies the discussion in Section II that the optimal architecture has two OR gates. Replacing all the 2-to-1 blocks with Cte-1s considerably increase the error values. Although, replacing all the 2-to-1 blocks with Cte-1s results in an architecture which is still better than that of LOA; it is not the optimal architecture as shown in Fig. 4. It can also be seen that the OLOCA's and LOA's formulas (see Table III) perfectly predict the behavior of the adders for all n_l s. Moreover, for all n_l s, OLOCA outperforms LOA, both from cost and error points of view; for the

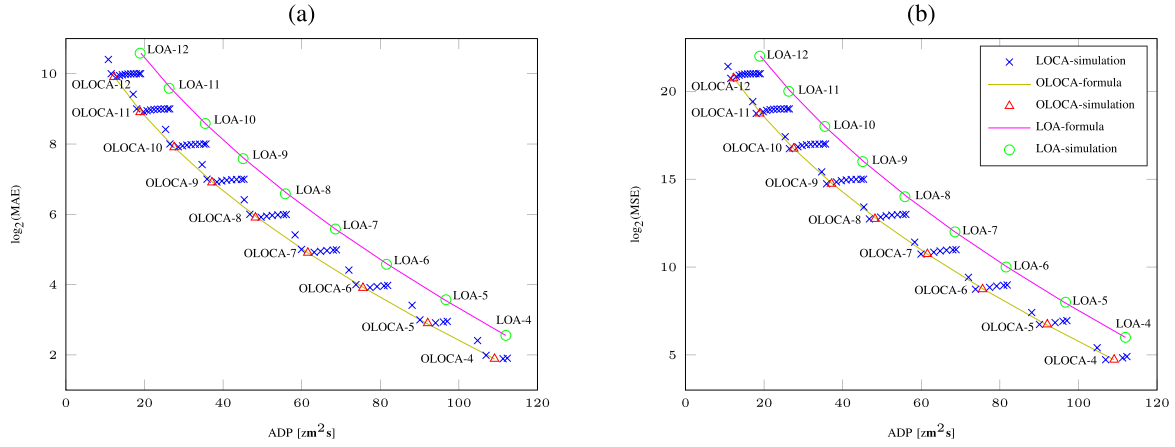


Fig. 4. Comparison of 16-bit LOA and OLOCA synthesized in a 65-nm technology: simulation and formulas results. (a) MAE versus ADP. (b) MSE versus ADP.

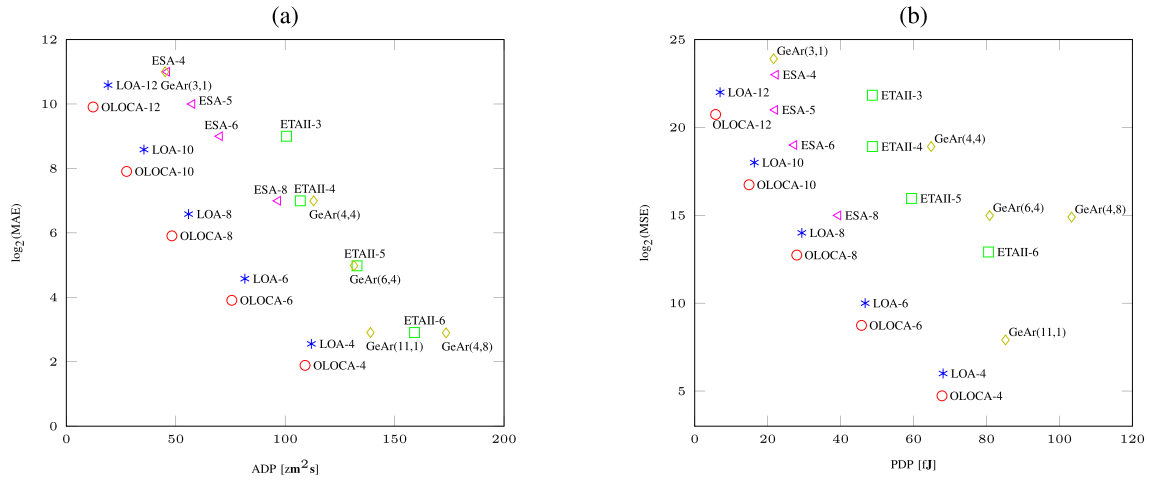


Fig. 5. Comparison of 16-bit approximate adders synthesized in a commercial 65-nm technology with various configurations. (a) MAE versus ADP. (b) MSE versus PDP.

TABLE IV
SIMULATION AND FORMULAS RESULTS FOR 8-BIT ADDERS SYNTHESIZED IN A COMMERCIAL 65-NM TECHNOLOGY

		$n_l=2$		$n_l=3$		$n_l=4$		$n_l=5$		$n_l=6$	
		Sim.	Formula	Sim.	Formula	Sim.	Formula	Sim.	Formula	Sim.	Formula
MAE	LOA	1.38	1.38	2.88	2.88	5.87	5.88	11.87	11.88	23.86	23.88
	OLOCA	0.75	0.75	1.78	1.78	3.70	3.70	7.48	7.48	14.96	14.99
MSE	LOA	4.00	4.00	16.00	16.00	63.93	64.00	255.90	256.00	1023.39	1024.00
	OLOCA	1.50	1.50	6.53	6.50	26.50	26.50	106.57	106.50	424.95	426.50
STD	LOA	1.99	1.98	3.99	3.99	7.99	8.00	16.00	16.00	31.99	32.00
	OLOCA	0.97	0.97	2.06	2.06	4.18	4.18	8.40	8.40	16.79	16.81
ADP [zm^2s]	LOA	26.82	26.82	19.19	19.50	13.19	13.32	7.95	8.28	3.94	4.38
	OLOCA	27.00	27.00	18.89	19.20	12.24	12.36	6.74	7.02	3.05	3.18

same values of errors, OLOCA improves the cost almost 25% and for the same values of cost, the error values of OLOCA are almost half of the LOAs. As an example, a 16-bit OLOCA-8 improves the cost by 13.6%, MAE by 37.4%, and MSE by 58% in comparison with LOA-8.

In order to evaluate OLOCA with another bitwidth, we have studied 8-bit adders as well; the results are tabulated in Table IV. Table IV shows the accuracy of the presented formulas versus the simulation results, as well as the superiority of OLOCA over LOA for all n_l s. As an example, OLOCA-4 improves MAE by 36.9%, MSE by 58.5%, and cost by 7.2% in comparison with LOA-4.

To show the superiority of OLOCA over all the existing approximate adders, besides LOA, we consider ESA, ETAIL, and GeAr. Among the existing combinational approximate adders, the above-mentioned architectures have proved to have the best performance [14], [15] after LOA. Different configurations of the adders have been simulated and the results are depicted in Fig. 5. Fig. 5(a) depicts the MAE of the adders versus ADP. Similarly, MSE versus power-delay product (PDP) of the adder architectures is illustrated in Fig. 5(b). Although ESA is hardware efficient, it is the least accurate adder architecture. As an example, for the almost same value of ADP, OLOCA-8 improves the error value by 97% in

comparison with ESA-4. OLOCA-8 improves error, ADP, and PDP by 53%, 54.9%, and 42.6% compared with ETAII-4, respectively. The improvements for the MSE are even larger.

IV. CONCLUSION

In this brief, an optimal approximate adder, through generalizing an architectural template for approximate adders, has been proposed. The proposed adder OLOCA shows considerable improvement in both error and hardware cost metrics in comparison with the previously reported best architectures. The superiority of OLOCA over the existing approximate adders has been proven presenting the mathematical analysis and further using experimental results. As an instance, a 16-bit approximate adder implemented with the OLOCA approach improves MSE by 58% while reducing the ADP by 13.8% at the same time, in comparison with an approximate adder implemented with the LOA approach.

REFERENCES

- [1] J. Sartori and R. Kumar, "Stochastic computing," *Found. Trends Electron. Design Autom.*, vol. 5, no. 3, pp. 153–210, Mar. 2011.
- [2] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 62–1–62–33, Mar. 2016.
- [3] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proc. 49th Annu. Design Autom. Conf. (DAC)*, Jun. 2012, pp. 820–825.
- [4] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *Proc. Design, Autom. Test Eur.*, Mar. 2011, pp. 1–6.
- [5] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Proc. 12th Int. Symp. Integr. Circuits*, Dec. 2009, pp. 69–72.
- [6] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, Mar. 2012, pp. 1257–1262.
- [7] I. C. Lin, Y. M. Yang, and C. C. Lin, "High-performance low-power carry speculative addition with variable latency," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 9, pp. 1591–1603, Sep. 2015.
- [8] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.
- [9] 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.
- [10] 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.
- [11] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proc. Conf. Design, Autom. Test Eur. (DATE)*, Mar. 2008, pp. 1250–1255.
- [12] S.-L. Lu, "Speeding up processing with approximation circuits," *Computer*, vol. 37, no. 3, pp. 67–73, Mar. 2004.
- [13] D. Esposito, D. De Caro, and A. G. M. Strollo, "Variable latency speculative parallel prefix adders for unsigned and signed operands," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 8, pp. 1200–1209, Aug. 2016.
- [14] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *Proc. 25th Ed. Great Lakes Symp. VLSI (GLSVLSI)*, 2015, pp. 343–348.
- [15] A. Najafi, M. Weißbrich, G. P. Vayá, and A. Garcia-Ortiz, "A fair comparison of adders in stochastic regime," in *Proc. 27th Int. Symp. Power Timing Modeling, Optim. Simulation (PATMOS)*, Sep. 2017, pp. 1–6.
- [16] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. dissertation, Fed. Inst. Technol., Zürich, Switzerland, 1998.