

Stochastic Mixed-PR: A Stochastically-Tunable Low-Error Adder

Ardalan Najafi^{ID}, *Student Member, IEEE*, and Alberto Garcia-Ortiz^{ID}, *Member, IEEE*

Abstract—Approximate computing is a promising paradigm in low-power design to trade off power efficiency by accuracy. However, its usability in real applications is restricted due to the lack of a dynamic configuration of the error characteristics. Most of the existing approximate adders have a fixed level of accuracy. On the other hand, the approximate adders which are configurable can only switch between an exact mode and a fixed level of approximation. This brief presents the mathematical stochastic error analysis of the adders, for the first time. Furthermore, based on the analysis, we propose to use a mixed adder as a reconfigurable architecture. This adder outperforms the state-of-the-art configurable adder. Moreover, it reduces the energy-delay product considerably in comparison with its conventional counterpart.

Index Terms—Approximate computing, stochastic computing, adder architecture, reconfigurability, error-cost trade-off.

I. INTRODUCTION

APPROXIMATE computing, a technique to reduce power, area and delay in VLSI design, optimizes a system by redesigning its logic circuit, allowing inexact functionality [1], [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.

Stochastic computing is another technique to trade off energy and quality of a circuit. Instead of 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 [3]. The considerable power and performance overheads imposed by worst-case design practices can be relaxed by scaling up the frequency, i.e., frequency overscaling (FOS), or scaling down the operating voltage, i.e., voltage overscaling (VOS). An overscaled design consumes lower power than its worst-case designed counterparts. At the same time, since the delay of some paths may now be greater than the clock period under certain conditions, timing violation may occur [3]. Overscaling can be considered as a knob to dynamically tune the exactness of the adder architectures [4]. This

technique, however, has not been remarkably embraced by the researchers. The rationale is that the conventional fast adder architectures start to make big errors as soon as they enter the stochastic regime.

Approximate adders, as key components of approximate arithmetic units, have attracted special attention of the researchers in the past few years. Among the existing configurable architectures, [5] and [6] are remarkable designs which use the idea of template. However, most of the existing architectures fix the level of hardware approximation statically, and hence, the level of approximation can not be dynamically tuned based on the application demand.

Nowadays, adaptation plays a major role in approximate processing [7]. In several scenarios, such as general purpose processors, DSP applications with variable quality signals (SNR), etc., both approximate and exact functionalities are required [8]–[10]. Fundamentally, this feature can be obtained in two ways: 1) by configuring the circuit logic or 2) by applying stochastic techniques.

In the first case, reconfigurability may be obtained by shutting down some parts of the logic [8] or bit truncation strategies [9]. These techniques use a control signal to switch between approximate and exact modes dynamically. However, both techniques are able to switch between an exact mode and an approximate mode with a fixed approximation level. The system can also be reconfigurable by adding an error-correction unit to the approximate architecture [11], [12]. This technique, however, increases the latency, silicon area and power consumption of the design.

Depending on the application demands for performance, different adder architectures are preferred. For a relaxed timing constraint, an exact adder is implemented with a Ripple-Carry Adder (RCA), Fig. 1(a). With stringent timing constraints, an exact adder is implemented with a Parallel-Prefix Adder (PPA), Fig. 1(b) as an example. For the cases in between, the synthesis tools implement an exact adder with a mixed adder of PPA and RCA, where the RCA is used for the MSBs to decrease the number of worst paths. Fig. 1(c) is as an example of the above-mentioned mixed adder. In this brief, we propose the use of a mixed adder with late input MSB arrival time, which allows embracing the idea of overscaling for the sake of reconfigurability. Aiming that, a precise analysis of the error behavior of adders in stochastic regime is presented, for the first time, in this brief.

The rest of this brief is organized as follows: Section II analyzes the stochastic error behavior of adders mathematically. Afterwards, in Section III, we quantify the advantages of the proposed mixed adder using experimental results. Moreover, we validate the mathematical analysis developed in Section II. Finally, Section IV concludes this brief.

Manuscript received July 29, 2019; revised October 2, 2019; accepted November 10, 2019. Date of publication November 14, 2019; date of current version October 5, 2020. This work was supported by DFG project under Project GA 763/4-1. This brief was recommended by Associate Editor M. Chrzanowska-Jeske. (Corresponding author: Ardalan Najafi.)

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 article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSII.2019.2953617

1549-7747 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

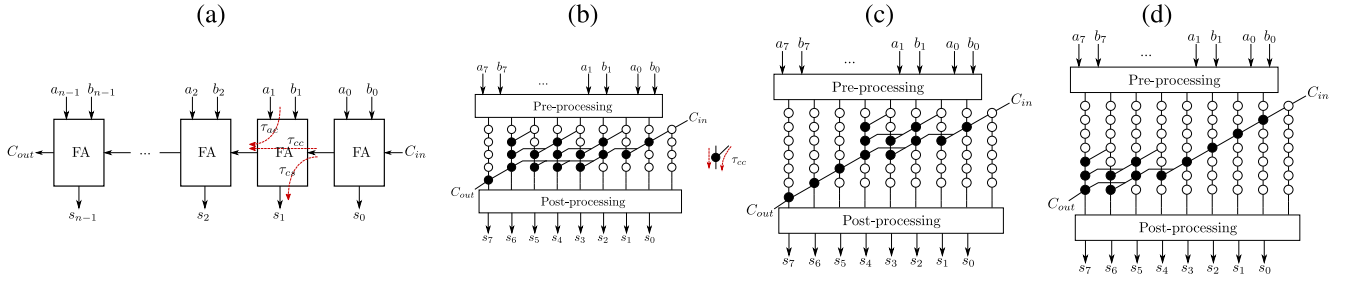


Fig. 1. The illustrative architectures of the exact adders: (a) Ripple-carry adder, (b) Parallel-prefix adder (Sklansky), (c) Std-mixed adder, (d) Mixed-PR adder.

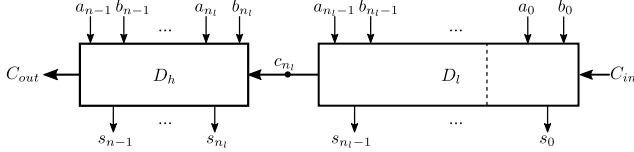


Fig. 2. The template of an adder split into two sub-blocks, used for the error analysis.

II. STOCHASTIC ANALYSIS

Frequency (voltage) overscaling is a conventional technique to dynamically trade-off the energy and accuracy of an adder. However, even a slight frequency (voltage) overscaling in conventional adders causes large timing errors to occur and degrades the quality of the adders rapidly. This rapid quality loss under overscaling was one of the main reasons to disregard stochastic computing techniques to dynamically tune the adder architectures.

As discussed in [4] and [13], with overscaling, RCA fails moderately when entering the stochastic regime. Unlike that, a pure PPA fails catastrophically with overscaling. However, RCA is extremely slow and starts to make errors at lower frequencies than a PPA. To solve the problem, we propose to use the mixed adder architecture with late input MSB arrival time, which shows a gradual increment in error when entering the stochastic regime. Hence, the proposed adder can be dynamically configured using FOS (VOS) techniques. We call the mixed adder with late input MSB arrival time, *Mixed-PR*, while the conventional mixed adder realized by the synthesis tool is called *Std-mixed*, for the rest of this brief.

In the rest of this section, firstly, we introduce the idea of our stochastic analysis and the nomenclature used in this brief. Then, the error characteristics of RCA in stochastic regime is analyzed. Once the behavior of an overscaled RCA is understood, we can justify why our proposed adder can be efficiently used in the stochastic regime. Subsequently, the error formulas of the proposed adder are presented mathematically.

A. Nomenclature and Analysis Strategy

In order to analyze the error behavior of the adders in the stochastic regime, we consider the abstract adder depicted in Fig. 2, divided into two sub-blocks, where n_l and n_h are the bit-width of the blocks D_l and D_h , respectively. Let us consider the transition of inputs at time 0, and sampling the data at time T . Correspondingly, a^- and b^- represent the previous values of inputs a and b , respectively. In addition, due to the fact that we are analyzing blocks with late arrival signals, in

order to perform our analysis, we have to differentiate between the logic (ideal) values (e.g., a and a^-), and the temporal evolution of signal (e.g., $a[t]$ where $a[0^-] = a^-$ and $a[0^+] = a$). Considering FOS, if we assume a scenario (clock period) in which D_h is the block that makes errors, with the information of the signal $c_{n_l}[t]$, we can characterize the stochastic errors of the adder. As a result, the problem of analyzing the error behavior of the adder of Fig. 2 in the stochastic regime can be split into two parts: 1) to characterize the temporal evolution of signal $c_{n_l}[t]$, and 2) to calculate the error as the result of the temporal evolution of signal $c_{n_l}[t]$.

Since the concepts of prefix structures are used often in this brief, P, G and K are used for the terms propagate, generate and kill, respectively. Traditionally, these terms are defined as $p_i = a_i \oplus b_i$, $g_i = a_i \cdot b_i$ and $k_i = \overline{a_i} \cdot \overline{b_i}$, where i is the bit position. The theory of the parallel prefix architectures, [14], can be used to accelerate the calculation of carries using the prefix operator \bullet defined by:

$$g_{i:m} = g_{i:j} + p_{i:j}g_{j:m}, \quad p_{i:m} = p_{i:j}p_{j:m} \quad (1)$$

where $i \geq l \geq j \geq m$. Using the generate and propagate signals, the carry signals c_i can be computed as follows:

$$c_i = g_{i-1:0} + p_{i-1:0} \cdot c_{in}. \quad (2)$$

In order to develop our error models, we use real delays instead of unit-gate delays. Therefore, the terms τ_{cc} , τ_{cs} are used. The notations are defined in Fig. 1. In addition, the error is defined as:

$$\varepsilon = \tilde{S} - S, \quad (3)$$

where \tilde{S} and S are the erroneous and the exact outputs, respectively.

In order to develop the error formulas of the adders, let us first consider a single FA with a late carry input signal. As a result, the temporal evolutions of $S[t]$ and $C_{out}[t]$ depend on the temporal evolution of $C_{in}[t]$ and logical values of inputs a and b . More precisely, the logical equation of sum bit, $s_i = p_i \oplus c_i$, can be rewritten arithmetically, as follows:

$$\begin{aligned} s_i[T] &= p_i + c_i[T - \tau_{cs}] - 2p_i c_i[T - \tau_{cs}] \\ &= p_i + (-1)^{p_i} c_i[T - \tau_{cs}]. \end{aligned} \quad (4)$$

Taking Eq. (3) and Eq. (4) into consideration, the error corresponding to each sum bit of an adder is:

$$\varepsilon_{s_i}[T] = (-1)^{p_i} \cdot (c_i[T - \tau_{cs}] - c_i) \cdot 2^i, \quad (5)$$

TABLE I
POSSIBLE TRANSITIONS ON $c_1[t]$

Transition of a_0, b_0	Transition of $c_1[t]$	Prob.
$k_0^- \rightarrow k_0$	$0 \rightarrow 0$	$1/16$
$k_0^- \rightarrow p_0$	$0 \rightarrow C_{in}$	$1/8$
$k_0^- \rightarrow g_0$	$0 \rightarrow 1$	$1/16$
$g_0^- \rightarrow k_0$	$1 \rightarrow 0$	$1/16$
$g_0^- \rightarrow p_0$	$1 \rightarrow C_{in}$	$1/8$
$g_0^- \rightarrow g_0$	$1 \rightarrow 1$	$1/16$
$p_0^- \rightarrow k_0$	$C_{in}^- \rightarrow 0$	$1/8$
$p_0^- \rightarrow p_0$	$C_{in}^- \rightarrow C_{in}$	$1/4$
$p_0^- \rightarrow g_0$	$C_{in}^- \rightarrow 1$	$1/8$

where ε_{s_i} is the error corresponding to the output at bit position i . In a similar manner, the following equation models the error of the output carry of an n -bit adder:

$$\varepsilon_{C_{out}}[T] = p_{n-1} \cdot (c_{n-1}[T - \tau_{cc}] - c_{n-1}) \cdot 2^n. \quad (6)$$

B. Stochastic Error Analysis of RCA

In order to analyze the error behavior of RCA in stochastic regime, the blocks of Fig. 2 are instantiated with RCAs. Consequently, Eq. (5) and Eq. (6) can be rewritten in respect to node c_{n_l} as follows:

$$\varepsilon_{s_i}[T] = (-1)^{p_i} \cdot (c_{n_l}[T - \tau_{cs} - (i - n_l)\tau_{cc}] - c_{n_l}) \cdot 2^i \cdot p_{i-1:n_l}, \quad (7)$$

$$\varepsilon_{C_{out}}[T] = (c_{n_l}[T - n_h\tau_{cc}] - c_{n_l}) \cdot 2^n \cdot p_{n-1:n_l}. \quad (8)$$

For the case where $\tau_{cc} = \tau_{cs}$, Eq. (7) and Eq. (8) can be combined as follows:

$$\varepsilon_{rca}[T] = \sum_{i=0}^{n_h-1} (p_{n_l+i-1:n_l}) \cdot (c_{n_l}[T - (i+1)\tau_{cc}] - c_{n_l}[T - i\tau_{cc}]) \cdot 2^{n_l+i}. \quad (9)$$

The equations presented above show that to calculate the error of a RCA for a given clock period, the temporal evolution of $c_{n_l}[t]$ is needed.

In order to analyze the temporal evolution of $c_{n_l}[t]$, let us split the block D_l of Fig. 2 into two sub-blocks in a way that the higher significant sub-block meets the timing constraints of the specified clock period, while the lower significant sub-block violates the timing. The value of $c_{n_l}[t]$ is constant “1” or “0” over time, if the higher significant sub-block does not propagate the carry. Correspondingly, the adder does not make any error, due to constant behavior of signal $c_{n_l}[t]$. Alternatively, if the higher significant sub-block propagates the carry, the behavior of the carry out signal of the lower significant sub-block is copied to the signal $c_{n_l}[t]$. Consequently, $c_{n_l}[t]$ is just the delayed version of $c_i[t]$.

Let us consider $n_h = 1$, which results in a correspondence between $c_{n_l}[t]$ of Fig. 2 and $c_1[t]$ of Fig. 1-a. Considering the possible transitions of the inputs of the first FA, i.e., a_0 and b_0 , the transitions of signal $c_1[t]$ can be found in Tab. I. Depending on C_{in} , the probabilities of changes in $c_1[t]$ are different. The probabilities of transitions on $c_1[t]$ are tabulated in Tab. II for the scenario in which C_{in} is constant “0”, as well as the scenario that C_{in} is an uniformly distributed random input (i.e., equals “0” or “1” with a probability of $1/2$). In order to

TABLE II
PROBABILITY OF CHANGES ON $c_1[t]$

$C_{in} = 0$			random C_{in}		
$c_1[0]$	$c_1[\tau]$	prob.	$c_1[0]$	$c_1[\tau]$	prob.
0	0	$9/16$	0	0	$1/4$
0	1	$3/16$	0	1	$1/4$
1	0	$3/16$	1	0	$1/4$
1	1	$1/16$	1	1	$1/4$

TABLE III
PROBABILITY OF CHANGES ON $c_{i+1}[t]$ FOR THE TRANSITIONS OF INPUT

a_i, b_i transition	changes in $c_{i+1}[t]$					Prob.
	$t=0$	$t=\tau$	$t=2\tau$...	$t=T$	
$\bar{p}_i^- \rightarrow \bar{p}_i$	c_{i+1}^-	c_{i+1}	c_{i+1}	...	c_{i+1}	$1/4$
$p_i^- \rightarrow \bar{p}_i$	c_i^-	c_{i+1}	c_{i+1}	...	c_{i+1}	$1/4$
$\bar{p}_i^- \rightarrow p_i$	c_{i+1}^-	c_i^-	c_i	...	c_i	$1/4$
$p_i^- \rightarrow p_i$	c_i^-	c_i^-	c_i	...	c_i	$1/4$

generalize the relation between input transitions and the carry signals, Tab. III tabulates the changes on the output carry of the FA in bit position i , i.e., $c_{i+1}[t]$, depending on the transitions of its inputs, i.e., a_i and b_i , and the input carry of the FA, i.e., $c_i[t]$. Considering Tab. III, and having $c_1[t]$, the possible changes of $c_2[t]$ and corresponding probabilities can be derived. Having the changes of $c_2[t]$, the characteristic of the signal $c_3[t]$ can be derived and so on. Let us formulate characterizing of the $c_i[t]$ embracing the information presented in Tab. II, and Tab. III. If $Pr(C_1)$ is the prob. column of the table containing the characterization of $c_1[t]$, i.e., Tab. II, $Pr(C_i)$ for $i \in \{2, n-1\}$ can be calculated as follows for the case C_{in} is uniformly random:

$$Pr(C_i) = \frac{1}{8} \begin{bmatrix} M \\ M \end{bmatrix} + \frac{1}{8} \begin{bmatrix} Pr(C_{i-1}) \\ Pr(C_{i-1}) \end{bmatrix} + \frac{1}{4} \begin{bmatrix} Pr(C_{i-1})_h \\ 0 \\ 0 \\ Pr(C_{i-1})_l \end{bmatrix}, \quad (10)$$

where M is a vector with the same size as $Pr(C_{i-1})$ which only the first and the last elements of this vector are one and the rest are zero, i.e., $[1 \ 0 \ 0 \ \dots \ 0 \ 1]^T$; $Pr(C_{i-1})_h$ and $Pr(C_{i-1})_l$ are the upper and lower half of the probability column of the previous table, i.e., c_{i-1} . Note that, the number of changes on a carry bit is linked to its bit position, and accordingly, the size of the table representing transition possibilities of a carry bit depends on its bit position. Therefore, the size of the table for c_1 , Tab. II, is 2^2 , the size of the table for c_2 is 2^3 and so on.

C. Stochastic Error Analysis of Mixed-PR Adder

Conventionally, when the timing constraints lie between the delay of RCA and PPA, these two adder structures are mixed, where the lower bits a parallel prefix structure is realized, and a serial prefix is used for the upper bits. However, if D_l in Fig. 2 is a PPA, the behavior of signal $c_{n_l}[t]$ is different from that analyzed above. The rationale is that the critical path to node c_{n_l} instead of being one path as in RCA, are multiple paths because of a parallel prefix structure. Hence, the transition combinations of signal $c_{n_l}[t]$ that produce errors are more probable which results in a dramatic increase of error in stochastic regime [4], [13].

Let us analyze mathematically the error behavior of the adder of Fig. 2 when instead of RCA, a PPA is instantiated as the upper block (D_h).

Using Eq. (3), Eq. (4) and Eq. (2), we can develop the error formulas of the Mixed-PR adder in the stochastic regime, considering an Sklansky architecture, as an instance. Obviously, the general equations Eq. (5) and Eq. (6) are also valid here for the PPA. Similar to the RCA, we can write these equations in respect to the node c_{n_l} :

$$\varepsilon_{s_l}[T] = (-1)^{p_i} \cdot (c_{n_l}[T - \tau_{cs} - \lceil \log_2(i - n_l + 1) \rceil \tau_{cc}] - c_{n_l}) \cdot 2^i \cdot p_{i-1:n_l}, \quad (11)$$

$$\varepsilon_{c_{out}}[T] = (c_{n_l}[T - \lceil \log_2(n_h + 1) \rceil \tau_{cc}] - c_{n_l}) \cdot 2^n \cdot p_{n-1:n_l}. \quad (12)$$

If we assume that $\tau_{cs} = \tau_{cc}$, and if the size of D_h is a power of 2, Eq. (11) and Eq. (12) can be combined as follows:

$$\varepsilon_{mix}[T] = \sum_{i=0}^{\log_2(n_h-1)} (p_{n_l+i-1:n_l}) \cdot (c_{n_l}[T - (i+1)\tau_{cc}] - c_{n_l}[T - i\tau_{cc}]) \cdot 2^{n_l+i}. \quad (13)$$

Considering the RCA as an architecture with a serial prefix-processing stage, Eq. (9) and Eq. (13) show that decreasing the clock period, the magnitude of error of each level of the prefix-processing stage corresponds to the first bit of that level. In other words, as Eq. (11)-(13) show, the Mixed-PR adder groups the bits of each level of the prefix-processing stage, and their errors compensate each other. This fact explains the reason why the error increment in Mixed-PR adder is gradual in stochastic regime.

III. EXPERIMENTAL RESULTS

To assess the circuit characteristics and evaluate the architectures presented in the previous section, we have generated VHDL description of the adders. The adders are synthesized in a commercial low-power 65-nm library for 12- and 16-bit operands. Using back-annotated simulations, dynamic power dissipation of the adders is evaluated after synthesis. All the adders have been simulated for 10^7 uniformly distributed random input patterns. In this section, the mixed adders' name are followed by two numbers. The numbers represent the bit-widths of the higher significant and the lower significant blocks, respectively.

In order to check the accuracy of the presented models, the mean absolute error (MAE) versus clock period of the RCA and the Mixed-PR adder are depicted in Fig. 3. The model for 12- and 16-bit adders are compared with the simulation results. As can be seen in the graph, the behavior of the adders in the stochastic regime is predicted precisely using the presented models.

In order to evaluate the efficiency of the Mixed-PR adder, we have studied and compared 12-bit adders for different clock periods. The results are shown in Fig. 4. In this figure, the MAE versus energy-delay product (EDP) as well as area-delay product (ADP) of the adders are depicted. As illustrated in this figure, for the same values of error, the Mixed-PR outperforms all the other architectures from both the EDP and the ADP points of view. As an example, if an application limits the maximum error to $\log_2(MAE) = 2.5$, the Mixed-PR-84 reduces the EDP and ADP about 40% in comparison with the

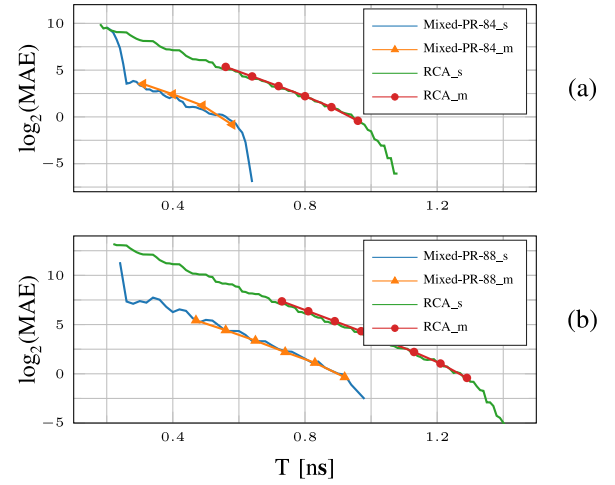


Fig. 3. Comparison of the simulation ($_s$) and the model ($_m$) results for the RCA and Mixed-PR for (a) 12-bit adders and (b) 16-bit adders, synthesized in a 65-nm technology.

TABLE IV
COMPARISON OF MIXED-PR-84 ADDER WITH 12-BIT ADDERS
GENERATED BY THE SYNTHESIS TOOL IN A COMMERCIAL 65NM

Adder	Energy [fJ]	Area [μm^2]	$T(\varepsilon=0)$ [ns]	$T(\varepsilon=5)$ [ns]
Mixed-PR-84	147.30	171.72	0.65	0.37
RCA	143.00	108.00	1.09	0.78
Std-mixed-48	151.70	171.72	0.66	0.63
PPA	232.20	182.88	0.32	0.31

conventional mixed adder, i.e., the Std-mixed-48 adder. The Mixed-PR-84 also reduces the EDP and ADP about 21% and 31%, respectively, in comparison with the adder of [9] when k is fixed to the optimal value of 4.

The energy consumption, silicon area, and the maximum clock period for a given error are reported in Tab. IV. As can be seen in the table, in contrast to the extreme architectures (i.e., RCA and PPA), our Mixed-PR-84 provides a promising trade-off; it requires only 147 fJ and achieves a speed of 0.65 ns. Furthermore, regarding the maximum achievable speed for a given MAE (5 in this example), we observe that Mixed-PR-84 improves the speed by a factor of 1.75x when entering into the approximate domain. It is the largest improvement of any other adder. Incidentally, we can note that the performance of the traditional Std-mixed-48 is very poor in terms of the maximum speed for a given MAE, which is 0.63 ns versus the 0.37 ns of our adder. Because of this, stochastic techniques had been rejected until now as a suitable method to construct inexact adders.

To exemplify the advantages of our approach, let us consider two scenarios. In the first one, an adder needs to work in exact mode with a clock period of 1.1 ns, while in approximate mode a maximum MAE of 5 is allowed. For this low-demanding scenario, even the RCA and the adder of [9] with $k = 4$ will work. They run in exact mode very efficiently (143 fJ), but the maximum speed in the inexact mode is very limited. The use of a PPA will allow a max speed of 0.31 ns in the inexact mode, but in the relaxed exact operation as much as 232 fJ is required. Our approach will operate in exact mode consuming only 147 fJ (factor 1.6 smaller than the PPA) and allowing a maximum speed in the inexact mode of 0.37 ns

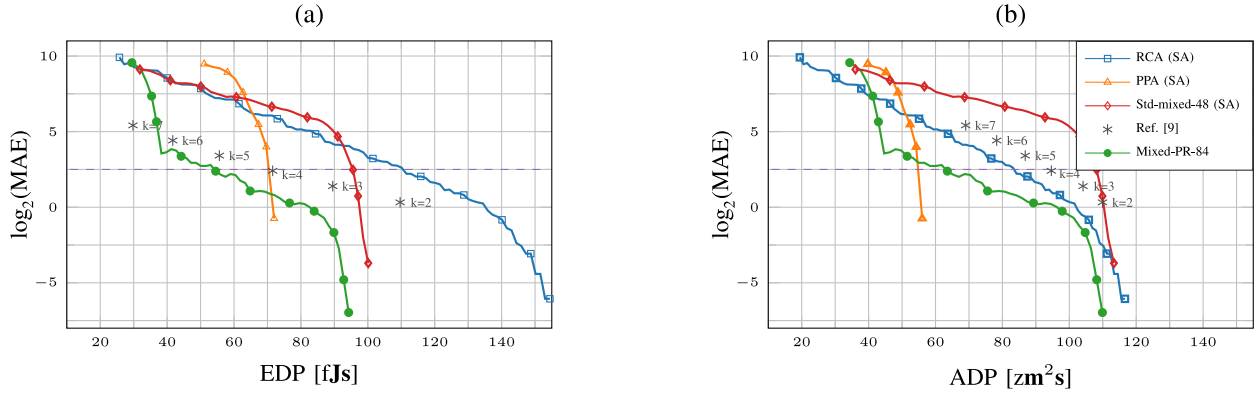


Fig. 4. 12-bit adders in stochastic regime: comparison of the Mixed-PR adder with the conventional exact adders generated by the synthesis tool (SA) as well as the approximate mode of the configurable adder of [9], synthesized in a commercial 65nm technology : (a) MAE vs. EDP, (b) MAE vs. ADP.

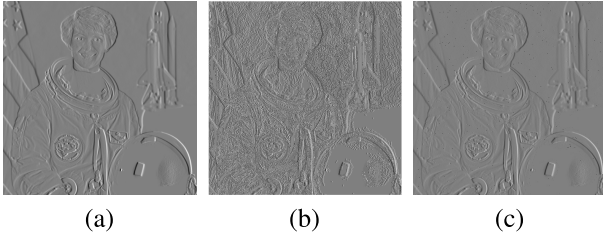


Fig. 5. Comparison of the *Astronaut* image after Average and Sobel filtering using (a) exact adder, (b) stochastic Std-mixed adder, and (c) proposed stochastic Mixed-PR adder. Stochastic adders work with a clock period of 0.5 ns.

(factor 2.5 faster than RCA). As a second scenario, let us consider that the adder needs to work with a period of 0.66 ns in the exact mode. In this case the RCA approach (and that of [9]) will not meet the timing requirements. Our approach still works and provides the same energy advantages versus PPA, as mentioned above.

In order to show the superiority of the Mixed-PR adder over its conventional counterpart, i.e., Std-mixed adder, the adders are evaluated in a more realistic scenario where the inputs have non-uniform distribution. In this case-study, the adders are used in a two-step image processing algorithm. In the first step, the *Astronaut* image is filtered with the average filtering using a 3×3 square kernel. In the second step, the result of the average filtering is fed to the Sobel filter to do the edge detection. The results of the process using the exact, Std-mixed and Mixed-PR adders are shown in Fig. 5. As can be seen in the figure, while the resulting image processed by the use of the Std-mixed is not usable, the Mixed-PR adder is still working perfectly in a relatively critical frequency.

IV. CONCLUSION

In this brief, using precise stochastic error analysis, the use of a mixed adder with late input MSB arrival time has been proposed. Our stochastically tunable Mixed-PR adder shows a gradual decrease in accuracy in contrast to the abrupt accuracy decrease of its conventional counterparts. Moreover, unlike the existing reconfigurable adders which switch between two modes (i.e., fixed approximate and exact modes), the proposed adder can be configured in multiple modes depending on the operating frequency and supply voltage. Furthermore, using

overscaling techniques, the mixed adder does not require any additional logic to be configured to the approximate modes. Using experimental results, the superiority of the proposed adder over its conventional counterparts as well as existing configurable adders has been shown.

REFERENCES

- [1] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surveys*, vol. 48, no. 4, pp. 1–33, May 2016.
- [2] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, "Systematic design of an approximate adder: The optimized lower part constant-or adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 8, pp. 1595–1599, Aug. 2018.
- [3] J. Sartori and R. Kumar, "Stochastic computing," *Found. Trends[®] Electron. Design Autom.*, vol. 5, no. 3, pp. 153–210, Mar. 2011.
- [4] 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 Model. Optim. Simulat. (PATMOS)*, Thessaloniki, Greece, Sep. 2017, pp. 1–6.
- [5] 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)*, San Francisco, CA, USA, Jun. 2015, pp. 1–6.
- [6] A. Najafi, M. Weißbrich, G. Payá-Vayá, and A. Garcia-Ortiz, "Coherent design of hybrid approximate adders: Unified design framework and metrics," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 4, pp. 736–745, Dec. 2018.
- [7] S. Xu and B. C. Schafer, "Toward self-tunable approximate computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 4, pp. 778–789, Apr. 2019.
- [8] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 8, pp. 1089–1093, Aug. 2018.
- [9] F. Frustaci, S. Perri, P. Corsonello, and M. Alioto, "Energy-quality scalable adders based on non-zeroing bit truncation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 4, pp. 964–968, Apr. 2019.
- [10] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2013, pp. 48–54.
- [11] 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.
- [12] D. Esposito, D. D. 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.
- [13] S. Narayanan, J. Sartori, R. Kumar, and D. L. Jones, "Scalable stochastic processors," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, Dresden, Germany, Mar. 2010, pp. 335–338.
- [14] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. dissertation, Swiss Federal Inst. Technol. (ETH) Zürich, Zürich, Switzerland, 1998.