RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder

Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram

***Abstract*—In this brief, we propose a fast yet energy-efficient reconfigurable approximate carry look-ahead adder (RAP-CLA). This adder has the ability of switching between the approximate and exact operating modes making it suitable for both error-resilient and exact applications. The structure, which is more area and power efficient than state-of-the-art reconfigurable approximate adders, is achieved by some modifications to the conventional carry look ahead adder (CLA). The efficacy of the proposed RAP-CLA adder is evaluated by comparing its characteristics to those of two state-of-the-art reconfigurable approximate adders as well as the conventional (exact) CLA in a 15 nm FinFET technology. The results reveal that, in the approximate operating mode, the proposed 32-bit adder provides up to 55% and 28% delay and power reductions compared to those of the exact CLA, respectively, at the cost of up to 35.16% error rate. It also provides up to 49% and 19% lower delay and power consumption, respectively, compared to other approximate adders considered in this brief. Finally, the effectiveness of the proposed adder on two image processing applications of smoothing and sharpening is demonstrated.**

***Index Terms*—Approximate computing, carry look-ahead adder (CLA), quality of service, reconfigurable.**

I. Introduction

POWER consumption and performance are critical parameters in the design of digital circuit in general. In the case of digital processing systems (especially the portable ones), owing to limited power budgets and reliability concerns, achieving a desired performance level can be challenging. Adders are key building blocks in arithmetic and logic units. Adders, which are utilized to perform other operations such as subtraction, multiplication, and division are among the most power hungry components in processors and are often hot-spot locations [1].

To meet the power and speed design constraints, a variety of methods at different design abstraction levels have been suggested. Approximate computing which is an emerging paradigm deals with demand performance and power efficiency constraints for adders at the cost of reducing the computational accuracy of the result [2]. This paradigm is especially suitable for applications where the answer is not unique and a set of approximate answers are acceptable.

Approximate arithmetic units are mainly based on the simplification of the arithmetic units circuits. Different structures for approximate adders have been proposed (see [3], [5]–[14]). Most of the proposed adders are fully approximate and may only be utilized in error resilient applications. At the same time, some of these structures have a constant level of deviation from the actual result, i.e., during the system operation their accuracies are not tunable. Runtime accuracy reconfigurability, however, may be considered as a useful feature for a system providing different levels of quality of service during operation [3]–[5]. Here, by reducing the quality, the total computation time and/or power consumption of the unit are reduced resulting in higher energy efficiency. In addition, some digital systems, such as general purpose processors, may be utilized for both approximate and exact computing, necessitating the ability to dynamically switch between exact and approximate operating modes. This feature may be obtained by adding a correction unit to the approximate circuit that would turn the approximate solution to an exact solution. The correction unit, however, increases the delay, power, and area overhead of the design. Also, the error correction procedure may require more than one clock cycle (see [6], [13]), which could slow down the processing.

In this brief, we present a reconfigurable approximate carry look-ahead adder (RAP-CLA) which has two operating modes of exact and approximate. The structure of the adder, which is based on the conventional carry look ahead adder (CLA), does not require an external correction unit for the exact add operation. While the delay and power consumption of the proposed adder, in the exact mode, are about the same as those of the conventional CLA, they are considerably smaller in approximate mode. This is because, in the approximate mode by exploiting the power gating technique, the power consumption is significantly reduced. The rest of this brief is organized as follows. In Section II, some recent related works are reviewed and the internal structure of the proposed reconfigurable approximate adder is described in Section III. The accuracy of the proposed adder is studied in Section IV and the results for the efficacy of the proposed adder are discussed in Section V. Finally, this brief is concluded in Section VI.

II. Related Work

Recently, many approximate architectures for arithmetic units which are based on the logic and circuits simplification at different design levels have been proposed (see [7], [12]). In the case of adders, the truncation of the carry propagation chain and dividing the adder into two accurate and

Manuscript received September 1, 2016; revised October 19, 2016; accepted November 18, 2016. Date of publication November 29, 2016; date of current version July 31, 2018. The work of O. Akbari, M. Kamal, and A. Afzali-Kusha was supported by the Iran National Science Foundation. This brief was recommended by Associate Editor M. H. Chowdhury.

O. Akbari and A. Afzali-Kusha are with the School of Electrical and Computer Engineering, University of Tehran, Tehran 4563-11155, Iran (e-mail: akbari.o@ut.ac.ir; afzali@ut.ac.ir).

M. Kamal is with the School of Electrical and Computer Engineering, University of Tehran, Tehran 4563-11155, Iran, and also with the School of Computer Science, Institute for Research in Fundamental Sciences, Tehran 19538-33511, Iran (e-mail: mehdikamal@ut.ac.ir).

M. Pedram is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2562 USA (e-mail: pedram@usc.edu).

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

Digital Object Identifier 10.1109/TCSII.2016.2633307

approximate parts, are the two major techniques which have been proposed widely (see [7], [9]). In [7], the full adder (FA) gates of the least significant (LS) part of the adder have been replaced by OR gates leading to smaller delay and power consumption at the cost of high error distances (EDs).

In [8], an approximate adder which was based on a redesign technique which cut the critical path in the carry chain was suggested. In [9], an error tolerant adder (ETA-I) which utilized a modified XOR gate instead of the FAs for the LS part of the adder was discussed. Note that the LS and most significant (MS) part of the adder in this approach were independent from each other and the carry input of the MS part was zero. An extension of the ETA-I (ETA-II) has been proposed in [10]. In this extension, the adder was divided to more than two segments where the input carry of each segment was generated approximately by the previous segment. All the above approximate adders operated only in the approximate mode.

A reconfigurable approximate adder which makes use of a dual-mode FA has been proposed in [3]. The dual-mode FA contains the proposed approximate FAs in [12] and an exact conventional FA. The outputs (e.g., *sum* and *carry*) of the dual-mode FA is selected from the outputs of these FAs through two multiplexers. Due to using two extra multiplexers for each dual-mode FA, the delay, area and power usage of this approach in the accurate operating mode was considerably large. An accuracy-configurable adder (ACA) has been proposed in [11] where the proposed adder operated in both accurate and approximate modes. The core of the ACA is an approximate adder where during the exact mode, an error detection and correction (EDC) unit is employed to generate exact results. In the approximate mode, this structure utilized a large number of subadder which led to large power and area usage. During the accurate mode, the adder used extra cycles for the correction resulting in a large delay in addition to the disadvantage of the area overhead of the correction unit.

In [14], a segmented approximate adder which enhanced by a carry speculation structure for increasing the accuracy was suggested. In addition, an error reduction unit was used to reduce the error of the adder. The delay of this adder with and without the error reduction unit was larger than the conventional exact CLA. Note that by increasing the segment size, the delay of this adder became considerably larger than that of the conventional CLA. In [13], a generalized model for generating an ACA has been proposed where the generated adders were called GeAr. Also, a configurable error correction unit was discussed. The error correction unit was a multicycle sequential unit requiring up to k cycles to determine the exact value of summation where k was the number of the subadders. A gracefully-degrading ACA has been proposed in [5]. The structure was based on the ACA without supporting the error correction mechanism.

Compared to the above approximate adders, our proposed reconfigurable approximate adder operates in two modes of the exact and approximate. Its delay and power are considerably smaller (especially in the approximate mode) compared to those of the exact CLA.

III. Proposed Reconfigurable Approximate CLA

In the conventional CLA, the carry output of the ith stage is determined from [15]

$$C_{i+1} = G_i + G_{i-1}P_i + \cdots + G_0 \prod_{j=1}^{i} P_j + C_{in} \prod_{j=0}^{i} P_j \qquad (1)$$

where C_{in} is the input carry and P_i and G_i are the propagate ($A_i \oplus B_i$) and generate (A_iB_i) signals of the ith stage, respectively. By increasing the width of CLA, the delay, area usage and power consumption of the carry generator units increase. In this brief, we propose to split (1), into two segments. Given a window size of W, the first segment contains W MS terms of (1) while the other segment contains the remaining (LS) terms. Hence, (1) may be rewritten as

$$C_{i+1} = \left(\sum_{j=i-W+1}^{i} G_j \left(\prod_{k=j+1}^{i-1} P_k \right) \right) + \left(\sum_{j=0}^{i-W} G_j \left(\prod_{k=j+1}^{i-1} P_k \right) + C_{in} \prod_{j=0}^{i} P_j \right) \qquad (2)$$

where the left part is called *approximate part* and the right part is called *augmenting part*. If both of these parts are employed to calculate the carry output, the generated C_{i+1} is exact whereas if only the approximate part is used, the generated C_{i+1} is imprecise.

It is clear that computing imprecise C_{i+1} is faster and consumes less power compared to computing C_{i+1} precisely. Based on this segmentation, two exact and approximate operating modes are realized for the proposed reconfigurable approximate adder. Thus, compared to the conventional CLA, only one multiplexer is added in the circuitry for computing C_{i+1}. The inputs of this multiplexer are approximate and augmenting carry outputs with the operating mode signal as the selector. The operating mode signal determines the generation of the carry output in the exact or approximate mode. As an example, the gate-level netlist of calculating C_4 in the proposed structure which is called RAP-CLA is shown in Fig. 1. In order to remove the power consumption of augmenting part when the carry generation block is in the approximate operating mode, this part is power gated using pMOS headers.

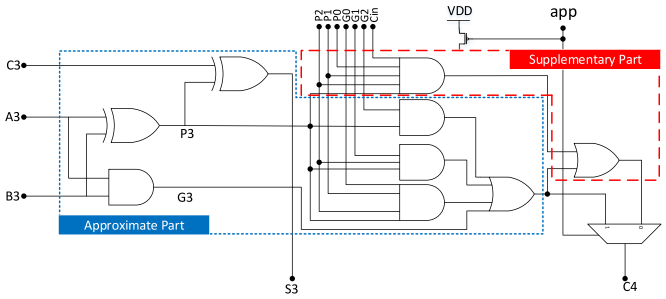

Fig. 1. Gate-level netlist of determining C4 in the proposed RAP-CLA structure.

If all carry generators of an adder are based on the proposed structure, and all the operating signals of these units are controlled by only one signal for the operating mode, all output bits of the adder in the approximate mode may be inaccurate. One may increase the accuracy of the approximate mode by exploiting the exact carry generator for the MS group of bits of the adder similar to the approach suggested in [7] and [9]. In fact, the technique may be utilized in designing CLAs with different levels of accuracy. For structures with adjustable accuracy, the adder should be partitioned into some segments with their own operating mode signals. In this brief, to compare the efficacy of the proposed structure with those of the previous works, the two operating modes of approximate and exact discussed before are considered.

IV. Accuracy of RAP-CLA

In this section, first the error metrics employed to study the accuracy of the approximate blocks are introduced and in the case of the proposed adder, analytically expressions for them are presented. Next, the accuracies of the approximate adder of this brief along with some other state-of-the-art approximate adders are compared.

A. Error Metrics

In the RAP-CLA structure, the accuracy of the add operation depends on the accuracy of the generated approximate carry (C') in each bit position. Here, we define $\lambda(C'_i)$ function which determines the correctness of the value of the approximate carry in the ith position (C'_i) based on the input operands. $\lambda(C'_i)$ is defined as

$$\lambda\left(C'_i\right) = \prod_{j=i-w}^{i-1} P_j \times \left[\sum_{l=-1}^{i-w-1} G_l \prod_{k=l+1}^{i-w-1} P_k - G_{i-w-1}\right] \quad (3)$$

where w is window size, and $\lambda(C'_i) \in \{0, 1\}$. When $\lambda(C'_i)$ is one (zero), the value of C'_i is wrong (correct). By employing (3), the probability of the error of the ith generated approximate carry is determined from

$$P\left(C'_i\right) = \left(\frac{1}{2}\right)^{w+1} \sum_{j=0}^{i-w-2} \left(\frac{1}{4}\right)^{i-w-j-1}. \quad (4)$$

Now, for the n-bit RAP-CLA, the error probability (error rate) of proposed adder is obtained from the union of $P(C'_i)$ with $w + 1 \leq i < n$ as

$$P(\text{error}(w, n)) = \sum_{k=w+1}^{n} (-1)^{k+1} \left(\sum_{w+1 \leq i_1 < \cdots < i_k \leq n} \left| P\left(C'_{i1}\right) \cap \ldots \cap P\left(C'_{ik}\right) \right| \right). \quad (5)$$

In addition to the error probability, the ED and Mean ED (MED) are other important metrics in the analysis of the accuracy. Based on (3), the ED of the proposed approximate adder is defined by

$$\text{ED}(A, B, w, n) = \sum_{i=w+1}^{n-1} 2^i (-1)^{P_i} \lambda\left(C'_i\right) + 2^n \lambda\left(C'_n\right). \quad (6)$$

In addition, the parameter normalized ED (NED) is another metric which is almost independent of the adder size. This parameters is defined as [16]

$$\text{NED} = \frac{\text{MED}}{D} = \frac{1}{2^{2n}} \sum_{i=1}^{2^{2n}} \frac{\text{ED}_i}{D} \quad (7)$$

where D is the maximum error value. Finally, the parameter mean relative ED (MRED) is the other error metric to evaluate the accuracy of an approximate adder. This metric is expressed by

$$\text{MRED} = \frac{1}{2^{2n}} \sum_{i=1}^{2^{2n}} \frac{|\text{ED}_i|}{S_i} \quad (8)$$

where S_i is the exact result for ith input set.

B. Accuracy Analysis

To compare the accuracy of the proposed adder with those of the recent previous works, the parameters discussed in the previous section are considered. The comparative study includes the proposed approximate adder structure in [13], which is called GeAr, and the proposed approximate adder in [14]. Note the proposed adder of [14] is considered in the two cases of with and without error reduction unit whose details are discussed in [14]. For this section, the error metrics are studied in the case of an 8-bit adder under different window (segment) sizes. Note that in the case of [14], one may only use the window size in the form of 2^k. In Table I, the error metrics for the approximate adders under all the possible input combinations of an 8-bit adder (set of 2^{16} inputs) are reported. As the figures in the table, the proposed approximate adder in [14] has the highest accuracy especially when it is equipped with the error reduction unit. In the considered window sizes, the ER and MRED of the proposed adder of [14] with (without) using the error reduction unit are about 21% (21%) and 74% (4%) smaller than those of RAP-CLA. On the other hand, the proposed approximate adder and the approximate adder of [14] have the same NED values for the same segment size. In addition, the accuracy of the RAP-CLA is considerably higher than that of the GeAr. The ER, MRED, and NED of the proposed approximate adder are smaller than those of GeAr by 48%, 10%, and 48%, respectively. Note that while the approximate adder of [14] is more accurate compared to our proposed adder in its approximate operating mode, as we will show in Section V, its design parameters (e.g., delay) in the window sizes larger than 1 are considerably larger than our proposed adder. In addition, in the exact operating mode, the design parameters of our design are significantly smaller.

TABLE I
ERROR METRICS OF THE PROPOSED ADDER COMPARED TO OTHER APPROXIMATE ADDERS UNDER DIFFERENT WINDOW SIZES

Adder Structure	W	Error Rate (%)	MRED	NED
[14] W/O ERU*	1	35.16	0.1266	0.1094
	2	12.4	0.0620	0.0570
	4	1.37	0.0136	0.0137
[14] W ERU*	1	35.16	0.0633	0.1094
	2	12.4	0.0155	0.0570
	4	1.37	0.0008	0.0137
GeAr	1	60.16	0.1369	0.1875
	2	30.08	0.0707	0.1076
	3	12.5	0.0351	0.0586
	4	4.69	0.0165	0.0273
	5	1.56	0.0070	0.0117
	6	0.39	0.0023	0.0039
RAP-CLA	1	35.16	0.1266	0.1094
	2	15.53	0.0646	0.0570
	3	6.25	0.0317	0.0293
	4	2.34	0.0147	0.0137
	5	0.78	0.0062	0.0059
	6	0.2	0.0020	0.0020

* ERU: Error Reduction Unit

TABLE II
PERCENT OF OUTPUTS WITH NED VALUES SMALLER THAN A SPECIFIC VALUE WHEN $N = 8$ AND $W = 4$

RED (%)	≤ 5%	≤ 10%	≤ 20%	≤ 25%	≤ 35%	≤ 50%	≤ 100%
GeAr	95.3%	95.8%	97.2%	97.7%	98.5%	99.5%	100%
[14]	98.6%	98.6%	98.6%	98.6%	98.6%	98.6%	100%
RAP-CLA	97.7%	97.7%	97.9%	98.1%	98.5%	99.0%	100%

Finally, in order to compare the accuracy of approximate adders, the percent of the outputs with relative ED (RED) smaller than a specified value for RAP-CLA, GeAr and [14] approximate adders are reported in Table II. The results, which have been obtained for the case of $N = 8$ and $W = 4$, RAP-CLA leads to slightly larger (smaller) number of outputs with RED values smaller than the 25% when compared to the case of the GeAr (the multiplier of [14]). Also, note that for the RED values larger than 35%, GeAr and RAP-CLA lead to slightly larger number of outputs.

V. RESULTS AND DISCUSSION

In this section, first, different design parameters of the 8- and 32-bit proposed adder of [14] (with and without error

TABLE III
DESIGN PARAMETERS OF THE 8-BIT ADDER STRUCTURES UNDER DIFFERENT WINDOW SIZES

Structure	W	Delay (ps)	Area (μm^2)	Power (μW)	Energy (aJ)	EDP (ps×aJ)
CLA	-	68.49	12.37	8.67	594	40670
[14] W/O ERU	1	27.40	16.91	9.34	256	7012
	2	52.22	15.14	7.93	414	21625
	4	68.70	15.82	8.76	602	41344
[14] W ERU	1	32.23	16.94	9.59	309	9962
	2	55.57	17.06	8.39	466	25909
	4	71.77	21.92	11.31	812	58257
GeAr	1	12.99	8.99	7.36	96	1242
	2	30.08	15.33	9.53	287	8623
	3	36.10	14.11	9.02	326	11755
	4	48.40	15.72	8.66	419	20287
	5	57.27	15.93	8.00	458	26239
	6	69.03	13.67	7.88	544	37549
RAP-CLA	1	18.42	8.96	6.03	111	2046
	2	30.44	9.58	7.53	229	6977
	3	37.75	9.82	8.05	304	11472
	4	44.54	10.36	8.14	363	16148
	5	46.45	10.73	8.02	373	17304
	6	53.18	11.94	7.16	381	20249

TABLE IV
DESIGN PARAMETERS OF THE 32-BIT ADDER STRUCTURES WHEN THE WINDOW SIZE IS FOUR

Adder	Delay (ps)	Area (μm^2)	Power (μW)	Energy (aJ)	EDP (ps×aJ)
CLA	99.44	89.80	58.22	5789	575698
[14] W/O ERU	79.45	62.84	47.70	3790	301097
[14] W ERU	88.30	66.47	51.49	4547	401462
GeAr	48.45	80.61	42.91	2079	100727
RAP-CLA	44.84	65.66	41.80	1874	84044

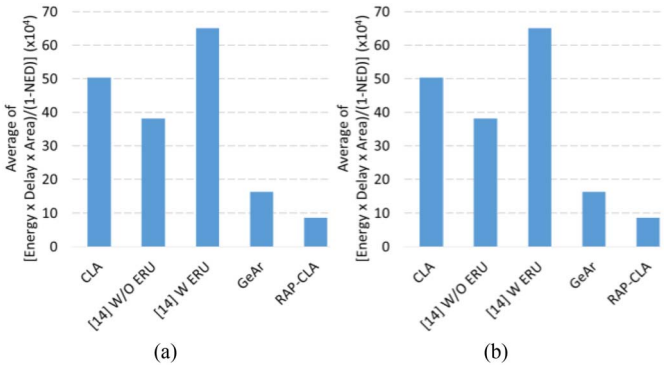

Fig. 2. (a) Average of Energy×Delay×Area/(1-NED) of 8-bit approximate adders for the window sizes of 1, 2, and 4. (b) Energy×Delay×Area/(1-NED) of 32-bit approximate adders for the window size is 4.

reduction unit), GeAr [13], and our proposed approximate adder (RAP-CLA) are studied. Except for the adder of [14] which does not support exact operating mode, the results include both approximate and exact operating modes. All the adders were synthesized by Synopsys Design Compiler in a 15 nm technology library [17]. Also, by employing this tool, the design parameters of the considered adders were extracted. At the end of this section, also, the results of employing these adders in two image processing applications are presented.

A. Design Parameters

The results for the delay, area, power, energy, and energy-delay-product (EDP) of the 8-bit approximate adders and the exact CLA have been reported in Table III. As the figures indicate, by increasing the window (segment) size, all the design parameters are increased. In the case of the proposed adder in this brief, for all the window sizes, the parameters are smaller than those of the exact CLA. In the case of GeAr, however, by increasing the windows size to a larger value than 6 (1), the delay (area) becomes larger than that of the exact CLA. In addition, when the window size become larger than 4, the delay of the proposed adder in [14] exceeds the delay of the exact CLA. Also, for all the window sizes, the power and area of the adder of [14] are larger than those of the exact CLA. For the window sizes larger than 4, the RAP-CLA outperforms the GeAr. The results also show that, in the case of the lowest (highest) accuracy of $W = 1$ ($W = 6$), the delay, power, area, energy and EDP of our 8-bit proposed approximate adder are about 73% (22%), 28% (3%), 30% (17%), 81% (36%), and 95% (50%), respectively, lower compared to those of the exact 8-bit CLA.

As the next step, the design parameters of the 32-bit approximate adders in the case of the window size of 4 are presented in Table IV. In this case, the RAP-CLA has the lowest parameters compared among all the adders.

Besides these design parameters, we have extracted the results for the Energy×Delay×Area/(1-NED) as an important figure of merit for the approximate units. Obviously, higher speed and quality, and lower power consumption for the design, leads to smaller values for this parameter. Fig. 2 shows the (average) Energy×Delay×Area/(1-NED) of the 8- and 32-bit for the approximate adders. Note that in the case of the 8-bit adders, the average of Energy×Delay×Area/(1-NED), under the window sizes of 1, 2, and 4 are reported while in the case of 32-bit adders, window size is 4. As the figures indicate the proposed approximate adder has the smallest average Energy×Delay×Area/(1-NED).

As mentioned before, both GeAr and RAP-CLA have the ability to operate in the exact mode too. Tables V and VI report the design parameters of these adders in their exact mode. As the results indicate, the delay, power, and area of the 8- and 32-bit GeAr and RAP-CLA are larger than those of the exact CLA. The parameters, however, are smaller in the case of the proposed adder compared to those of the GeAr. Note that the error recovery in the GeAr is a multicycle approach, hence, the delay of its exact mode is considerably larger. In the case of 8-bit adder, the delay, area, power, energy, and EDP of the proposed adder are on average 1% (65%), 7% (47%), 5% (18%), 6% (72%), and 6% (91%) higher (lower) compared to those of the exact CLA (GeAr), respectively. The numbers exhibit lower overheads for RAP-CLA in the exact mode. The same trend is also observed for the case of the 32-bit adders (Table VI). Note that the overheads of the proposed adder compared to the exact CLA are lower while the overheads of the GeAr are higher. Hence, RAP-CLA outperforms the GeAr in both exact and approximate modes.

B. Image Processing Applications

In this section, we employ proposed approximate adder in [14] (without utilizing the error reduction unit), GeAr, and RAP-CLA in sharpening [18] and smoothing [19] applications to evaluate the impact of the approximate adders on the quality of the output images. In both applications, 16-bit adders are utilized.

For both of these applications, the approximate adders (in approximate mode) along with the exact CLA are

TABLE V
DESIGN PARAMETERS OF THE 8-BIT OF THE APPROXIMATE ADDERS IN THEIR EXACT OPERATING MODE UNDER DIFFERENT WINDOW SIZES

Structure	W	Delay (ps)	Area (μm^2)	Power (μW)	Energy (aJ)	EDP (ps×aJ)
CLA	-	68.49	12.37	8.67	594	40670
GeAr	1	196.91	21.43	8.38	1650	324922
	2	178.20	26.10	10.41	1855	330636
	3	176.75	25.46	11.19	1978	349582
	4	247.80	31.06	14.28	3539	876861
	5	222.45	25.56	12.14	2701	600736
	6	167.72	19.96	10.27	1722	288895
RAP-CLA	1	68.61	13.94	9.34	641	43966
	2	68.88	13.67	9.07	625	43034
	3	68.92	13.05	9.03	622	42892
	4	69.07	13.08	9.32	644	44463
	5	68.83	12.91	9.45	650	44770
	6	68.82	12.73	8.75	602	41442

TABLE VI
DESIGN PARAMETERS OF 32-BIT APPROXIMATE ADDERS IN THEIR EXACT OPERATING MODE WHEN THE WINDOW SIZE IS 4

Adder	Delay (ps)	Area (μm^2)	Power (μW)	Energy (aJ)	EDP (ps×aJ)
CLA	99.44	89.80	58.22	5789	575698
GeAr	1233.96	138.12	48.90	60341	74457941
RAP-CLA	100.18	102.17	62.96	6307	631869

TABLE VII
PSNRS OF OUTPUT IMAGES OF SMOOTHING AND SHARPENING APPLICATIONS FOR DIFFERENT APPROXIMATE ADDERS

Application	Smoothing			Sharpening		
benchmark	GeAr	[14] w/o ERU	RAP-CLA	GeAr	[14] w/o ERU	RAP-CLA
girl	12.0	17.2	13.8	15.4	26.0	15.4
Tiffany	6.8	14.5	8.4	17.5	26.2	18.9
moon surface	11.7	17.3	13.0	14.0	26.3	14.5
cameraman	13.1	18.4	12.7	16.2	26.5	16.5
einstein	12.5	16.7	14.3	14.8	26.0	15.8
elaine	11.1	16.3	12.0	14.8	26.4	15.6
lena	13.4	17.2	14.7	15.9	26.4	16.7
house	12.4	17.6	13.7	16.9	26.6	17.9
peppers	13.0	16.8	13.2	15.2	26.3	16.3
tulips	11.7	16.3	13.3	15.5	26.3	16.5

considered for ten benchmark images. The peak signal-to-noise ratio (PSNR) of the output images (compared to the case of the exact filtering) are given in Table VII. The highest PSNR belongs to the adder of [14] while the GeAr leads to the lowest PSNR in both applications. Our proposed approximate adder increased the PSNR of ten benchmark images, on average, by 9.3% and 5.1% compared to the case of using the GeAr in smoothing and sharpening applications, respectively. Note that, as the accuracy analysis of Section IV-B also showed, the approximate adder of [14] provides the highest accuracy, and hence, the quality of the output images for this adder is higher than that of RAP-CLA. On average, the approximate adder of [14] results in 30.2% and 60.3% higher quality in smoothing and sharpening applications, respectively.

VI. CONCLUSION

In this brief, a high-speed yet energy-efficient RAP-CLA was suggested. The adder enjoyed the ability of switching between the approximate and exact operating modes making it suitable for both error-resilient and exact applications. The structure of the proposed adder was based on some modifications to the structure of the conventional CLA. The results showed up to 49% and 19% lower delay and power consumption, respectively, compared to those of the approximate adders at the price of up to an error rate of 35.16%. Also, the effectiveness of the proposed adder on two image processing applications is studied as well.

REFERENCES

[1] B. K. Mohanty and S. K. Patel, "Area–delay–power efficient carry-select adder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 6, pp. 418–422, Jun. 2014.

[2] B. Shao and P. Li, "Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 4, pp. 1081–1090, Apr. 2015.

[3] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 846–857, Mar. 2016.

[4] M. S. Khairy, A. Khajeh, A. M. Eltawil, and F. J. Kurdahi, "Equi-noise: A statistical model that combines embedded memory failures and channel noise," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 407–419, Feb. 2014.

[5] 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, 2013, pp. 48–54.

[6] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proc. Design Autom. Test Europe (DATE)*, *Munich, Germany,* 2008, pp. 1250–1255.

[7] 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.

[8] D. Shin and S. K. Gupta, "A re-design technique for datapath modules in error tolerant applications," in *Proc. 17th Asian Test Symp. (ATS)*, Sapporo, Japan, 2008, pp. 431–437.

[9] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[10] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Proc. Int. Symp. Integr. Circuits (ISIC)*, Singapore, 2009, pp. 69–72.

[11] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proc. 49th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2012, pp. 820–825.

[12] 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.

[13] 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, 2015, pp. 1–6.

[14] J. Hu and W. Qian, "A new approximate adder with low relative error and correct sign calculation," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, Grenoble, France, 2015, pp. 1449–1454.

[15] C. Efstathiou, Z. Owda, and Y. Tsiatouhas, "New high-speed multioutput carry look-ahead adders," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 10, pp. 667–671, Oct. 2013.

[16] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1760–1771, Sep. 2013.

[17] *NanGate—The Standard Cell Library Optimization Company*. (2016). [Online]. Available: http://www.nangate.com/

[18] M. S. K. Lau, K.-V. Ling, and Y.-C. Chu, "Energy-aware probabilistic multiplier: Design and analysis," in *Proc. Int. Conf. Compilers Archit. Synthesis Embedded Syst.*, Grenoble, France, 2009, pp. 281–290.

[19] H. R. Myler and A. R. Weeks, *The Pocket Handbook of Image Processing Algorithms in C*. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.