A Simple Yet Efficient Accuracy Configurable Adder Design

Wenbin Xu*, Sachin S. Sapatnekar† and Jiang Hu*
*Department of Electrical and Computer Engineering, Texas A&M University
†Department of Electrical and Computer Engineering, University of Minnesota
wbxu@tamu.edu, sachin@umn.edu and jianghu@tamu.edu

***Abstract*—Approximate computing is a promising approach for low power IC design and has recently received considerable research attention. To accommodate dynamic levels of approximation, a few accuracy configurable adder designs have been developed in the past. However, these designs tend to incur large area overheads as they rely on either redundant computing or complicated carry prediction. Some of these designs include error detection and correction circuitry, which further increases area. In this work, we investigate a simple accuracy configurable adder design that contains no redundancy or error detection/correction circuitry and uses very simple carry prediction. Simulation results show that our design dominates the latest previous work on accuracy-delay-power tradeoff while using 39% lower area. Moreover, we propose a delay-adaptive self-configuration technique to further improve accuracy-delay-power tradeoff.**

I. Introduction

Power constraints are a well-known challenge in advanced VLSI technologies. Low power techniques for the conventional exact computing paradigm have been already extensively studied. A comparatively new direction is approximate computing, where errors are intentionally allowed in exchange for power reduction. In many applications, such as audio, video, haptic processing and machine learning, occasional small errors are indeed acceptable. Such error-tolerant applications are found in abundance in emerging applications and technologies.

A great deal of approximate computing research has been concentrated on arithmetic circuits, which are essential building blocks for most of computing hardware. In particular, several approximate adder designs have been developed [1]–[9]. One such design [3] achieves 60% power reduction for DCT (Discrete Consine Transform) computation without making any discernible difference to the images being processed. In realistic practice, accuracy requirements may vary for different applications. In mobile computing devices, different power modes may entail different accuracy constraints even for the same application. Specifically, arithmetic accuracy can be adjusted at runtime using methods such as dynamic voltage and frequency scaling (DVFS) to obtain the best accuracy-power tradeoff. The benefit of runtime accuracy adjustment is demonstrated in [2], but their approximation is realized by voltage over-scaling, where errors mostly occur at the timing-critical path associated with the most significant bits, i.e., errors are often large.

A few accuracy configurable adder designs that use approximation schemes other than voltage over-scaling have been proposed. An early work [10], called ACA, starts with an approximate adder and augments it with an error detection and correction circuit, which can be configured to deliver varying approximation levels or accurate computing. Its baseline approximate adder contains significant redundancy and the error detection/correction circuit further increases area overhead. The ACA design [10] is generalized to a flexible framework GeAr in [11]. In both ACA and GeAr, the error correction must start from the least significant bits and hence accuracy improves slowly in the progression of configurations. The work of Accurus [12] modifies ACA/GeAr to overcome this drawback and achieves graceful degradation. However, in ACA, GeAr as well as Accurus, the error correction circuit is pipelined, implying that the computation in accurate mode takes multiple clock cycles and causes data stalls.

An alternative direction of accuracy configurable adder design is represented by GDA [13] and RAP-CLA [14]. These methods start with an accurate adder and use carry prediction for optional approximation. As such, they no longer need error detection/correction and do not incur any data stall. In addition, they intrinsically support graceful degradation. The GDA design [13] is composed by accurate CRA (Carry Ripple Adder) and extra configurable carry prediction circuitry, similar as the carry look-ahead part of CLA (Carry Look-ahead Adder). Thus, its area is generally quite large. RAP-CLA [14] is based on accurate CLA design and reuses a portion of the carry look-ahead circuit as carry prediction. This leads to an overall area that is less than GDA but greater than CLA. In [14], the carry-prediction-based approach is shown to be superior to error-correction-based design [11].

In this paper, we propose a new carry-prediction-based accuracy configurable adder design: **SARA** (Simple Accuracy Reconfigurable Adder). It is a simple design with significantly less area than CLA, which, to the best of our knowledge, has not been achieved in the past in accuracy configurable adders. SARA inherits the advantages of all previous carry-prediction-based approaches: no error correction overhead, no data stall and allowing graceful degradation. Compared to GDA [13], SARA incurs 50% less PDP (Power Delay Product) and can reach the same PSNR (Peak Signal-to-Noise Ratio). Moreover, SARA demonstrates remarkably better accuracy-power-delay tradeoff than the latest, and arguably the best, previous work RAP-CLA [14]. A delay-adaptive reconfiguration technique is developed to further improve the accuracy-power-delay tradeoff. The proposed designs are also validated by DCT computation in image processing.

II. Prior Works and Rationale of Our Design

We review a few representative works on accuracy configurable adder design and show the relation with our method.

These designs can be generally categorized into two groups: error-correction-based configurations [10]–[12] and carry-prediction-based configurations [13], [14].

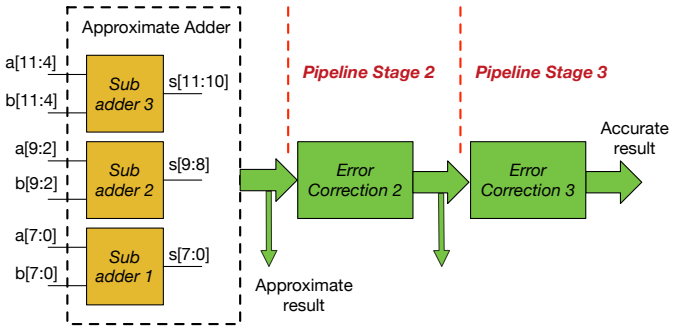

Fig. 1. Error-correction-based configurable adder.

The main idea of an error-correction-based approach [10]–[12] is shown in Figure 1. The scheme starts with an approximate adder (the dashed box), where the carry chain is shortened by using separated sub-adders with truncated carry-in. In order to reduce the truncation error, the bit-width in some sub-adders contains redundancy. For example, *sub-adder2* calculates the sum for only bit 8 and 9, but it is an 8-bit adder using bit $[9:2]$ of the addends, 6 bits of which are redundant. Even with the redundancy, there is still residual error which is detected and corrected by additional circuits. In Figure 1, the errors of *sub-adder2* must be corrected by *error-correction2* before the errors of *sub-adder3* are rectified by *error-correction3*. As such, the configuration progression always starts with small accuracy improvements. The redundancy and error detection/correction incur large area overhead. Since the error correction circuits are usually pipelined, an accurate computation may take multiple clock cycles and could stall entire datapath, depending on the addend values.

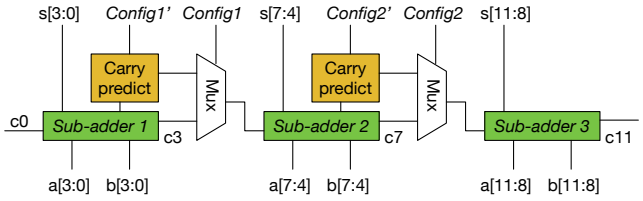

Fig. 2. Carry-prediction-based configurable adder.

The framework of carry-prediction-based methods [13], [14] is shown in Figure 2. These schemes start with an accurate adder design, which is formed by chaining a set of sub-adders. Each sub-adder comes with a fast but approximated carry prediction circuit. By selecting between the carry-out from sub-adder or carry prediction, the overall accuracy can be configured to different levels. Such an approach does not need error detection/correction circuitry. Moreover, the configuration of higher bits is independent of lower bits. This leads to fast convergence or graceful degradation in the progression of configurations. In GDA [13], the sub-adders are CRA designs while the carry-prediction circuit is similar to the carry look-ahead part of CLA. Further, its carry prediction can be configured to different accuracy levels. However, the complicated carry prediction induces large area overhead. The RAP-CLA scheme [14] uses CLA for its baseline where the carry-ahead of each bit is computed directly from the addends of all of its lower bits. Its carry prediction reuses a part of the look-ahead circuit rather than building extra dedicated prediction circuitry, and hence is more area-efficient than GDA. However, its baseline is much more expensive than GDA.

Our design is a carry-prediction-based approach. Its sub-adders are CRA instead of expensive CLA as in RAP-CLA. Its carry prediction also reuses part of the sub-adders rather than having dedicated prediction circuitry. As such, it avoids the disadvantages of both GDA and RAP-CLA. A comparison among the characteristics of these different techniques is provided in Table I.

TABLE I
COMPARISON OF CHARACTERISTICS FOR DIFFERENT TECHNIQUES.

Method	Baseline sub-adder	Error correction	Graceful degradation	Carry prediction
ACA [10]	Redundant CRA	Yes	No	No
GeAr [11]	Redundant CRA	Yes	No	No
Accurus [12]	Redundant CRA	Yes	Yes	No
GDA [13]	CRA	No	Yes	Stand-alone
RAP-CLA [14]	CLA	No	Yes	Reuse
SARA (ours)	CRA	No	Yes	Reuse

III. Simple Accuracy Reconfigurable Adder

A. Preliminaries

An N-bit adder operates on two addends $A = (a_N, a_{N-1}, ..., a_i, ..., a_1)$ and $B = (b_N, b_{N-1}, ..., b_i, ..., b_1)$. For bit i, its carry-in is c_{i-1} and its carry-out is c_i. Defining the carry generate bit $g_i = a_i \cdot b_i$ and propagate bit $p_i = a_i \oplus b_i$, the conventional full adder computes the sum s_i and carry c_i according to

$$s_i = p_i \oplus c_{i-1}, \tag{1}$$

$$c_i = g_i + p_i \cdot c_{i-1}. \tag{2}$$

A gate level schematic of conventional full adder is provided in Figure 3(a). A CRA is used to chain N bits of conventional full adders together.

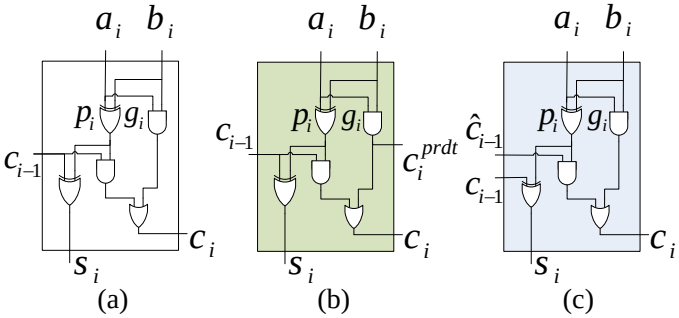

Fig. 3. (a) Conventional full adder; (b) Our carry-out selectable full adder; (c) Our carry-in configurable full adder.

By applying Equation (2) recursively, one can get

$$c_i = g_i + p_i g_{i-1} + \ldots + g_1 \prod_{k=2}^{i} p_k + c_0 \prod_{k=1}^{i} p_k. \tag{3}$$

This equation implies that c_i can be computed directly from g and p of all bits, without waiting for the c of its lower bits to be computed. This observation is the basis for CLA adder.

B. SARA: Simple Accuracy Reconfigurable Adder Design

In SARA, an N-bit adder is composed of K segments of L-bit sub-adders, where $K = \lceil N/L \rceil$ (see Figure 2). Each sub-adder is almost the same as a CRA except that the MSB (Most Significant Bit) of a sub-adder, which is bit i, provides a carry prediction as

$$c_i^{prdt} = g_i \tag{4}$$

For the LSB (Least Significant Bit) of the higher-bit sub-adder, which is bit $i+1$, its carry-out c_{i+1} can be computed using one of two options: either by the conventional $c_{i+1} = g_{i+1} + p_{i+1} \cdot c_i$, or by using the carry prediction as

$$c_{i+1} = g_{i+1} + p_{i+1} \cdot c_i^{prdt} = g_{i+1} + p_{i+1} \cdot g_i \tag{5}$$

The selection between the two options is realized using MUXes as in Figure 4 and the MUX selection result is denoted as $\hat{c}_i$. Comparing Equation (5) with (3), we can see that the carry prediction is a truncation-based approximation to carry computation[1]. Therefore, $\hat{c}_i$ can be configured to either accurate mode or approximation mode, i.e.,

$$\hat{c}_i \leftarrow \begin{cases} c_i^{prdt}, & \text{if approximation mode} \\ c_i, & \text{if accurate mode.} \end{cases} \tag{6}$$

It should be noted that the carry prediction c_i^{prdt} reuses g_i in an existing full adder instead of introducing an additional dedicated circuit as in [13] or Figure 2. This prediction scheme makes a very simple modification to the conventional full adder, as shown in Figure 3(b).

One can connect $\hat{c}_i$ to its higher bit $i+1$ to compute both carry c_{i+1} and sum s_{i+1} , as in GDA [13] and RAP-CLA [14]. We suggest an improvement over this approach by another simple change as in Figure 3(c), where s_{i+1} is based on c_i instead of $\hat{c}_i$. Applying this in SARA as in Figure 4, in the approximation mode, computing s_{j+1} from c_j can still limit the critical path to be between c_{i-1}^{prdt} and s_{j+1}, but has higher accuracy than computing s_{j+1} from $\hat{c}_j$. Compared to sum computation in GDA and RAP-CLA, this technique improves accuracy with almost no additional overhead. Compared to CRA, the overhead of SARA is merely the MUXes, which is almost the minimum possible for configurable adders.

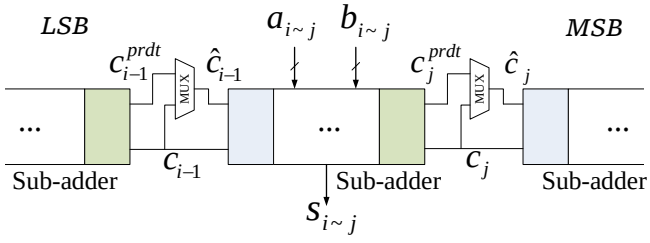

Fig. 4. Design of SARA.

[1]A similar approximation is used in static approximate adder design [7].

C. Usage of SARA

When $\hat{c}_i$ is configured to be c_i for all K sub-adders, SARA operates very much like the CRA, where the critical path is along N-bit full adders. If all $\hat{c}_i$ are selected to be c_i^{prdt}, the critical path is shortened to roughly L-bit full adders. This large delay reduction can be translated to power reduction by supply voltage scaling. There can be 2^{K-1} different configurations. For two configurations with the same critical path length, obviously we only need the one with higher accuracy. Therefore, there are K effective configurations, with critical path length of L-bit, $2L$-bit, ..., $K \cdot L \simeq N$-bit full adders.

D. SARA Error Analysis

When a sub-adder is in approximation mode, an error occurs when $c_i^{prdt} \neq c_i$. There is only one situation where this error may happen: when $c_{i-1} = 1, p_i = 1, c_i^{prdt} = 0$ and $c_i = 1$. Then the error rate, or probability of such error, is given by

$$\begin{aligned} ER_i^{prdt} &= P(c_i^{prdt} \neq c_i) = P(c_i^{prdt} = 0, c_i = 1) \\ &= P(c_{i-1} = 1, p_i = 1) = P(c_{i-1} = 1)P(p_i = 1) \end{aligned}$$

where P indicates probability and the last part assumes that c_{i-1} and p_i are independent of each other. Then, the error rate of MUX output $\hat{c}_i$ is

$$\widehat{ER}_i = P(\hat{c}_i \neq c_i) = \begin{cases} ER_i^{prdt}, & \text{if } \hat{c}_i \leftarrow c_i^{prdt} \\ 0, & \text{if } \hat{c}_i \leftarrow c_i. \end{cases} \tag{7}$$

Theorem 1. *If $\mathcal{I}$ is the set of bits with MUX at output, the expected error of SARA is*

$$\sum_{\forall i \in \mathcal{I}} \widehat{ER}_i \cdot P(p_{i+1} = 1) \cdot 2^{i+1}.$$

Proof Sketch: The expected error is derived based on the observation that an error produced at bit i can be directly propagated to the carry-in of bit $i+2$ in presence of propagate bit at bit $i+1$ and lead to an overall error magnitude of 2^{i+1}. The detailed proof is omitted due to space limitations.

Since $|\mathcal{I}| = K-1$, the error of the worst case approximation mode increases with the number of sub-adders, K. In addition, area overhead increases with K. On the other hand, a large K implies smaller L, and thus often facilitates shorter critical path and more power reduction. Therefore, K significantly affects the tradeoff among accuracy, power, delay and area.

IV. Delay-Adaptive Reconfiguration of SARA

Almost all previous works on accuracy configurable adder [10]–[14] reasonably assume that the accuracy configuration is decided by architecture/system level applications. We propose a self-configuration technique for the scenarios where the architecture/system level choice is either unclear or difficult. Simulation results show that SARA with the self-configuration outperforms several previous static approximate adder designs.

The main idea of self-configuration is based on the observation that the actual worst case path delay depends on addend values. Specifically, the actual path delay is large only when a carry is propagated through several consecutive bits. Any false propagate bit from the addends results in a shorter carry propagation chain. When the actual carry propagation chain

is short, there is no need to use approximation configuration, which is intended to cut carry chain shorter. We propose a Delay Adaptive Reconfiguration (**DAR**) technique: the output of a MUX in SARA is set to approximation mode only when a potentially long carry chain is detected. Compared to the constantly-approximate configuration, some errors for actual short carry chains are avoided, the actual long carry chain is cut shorter, and delay/power reduction can be still obtained.

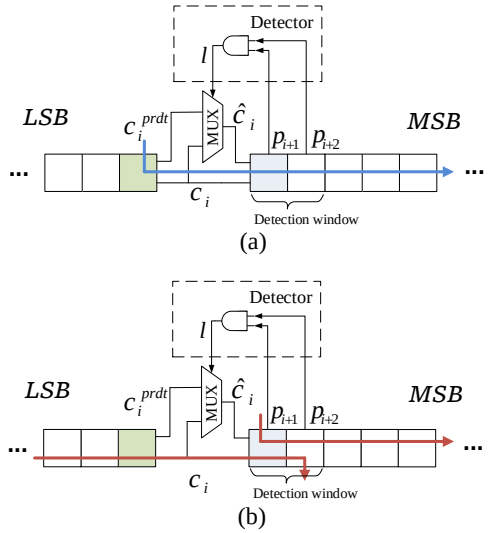

Fig. 5. Design and operation of delay-adaptive reconfiguration for SARA.

The long carry chain detection and SARA-DAR design are shown in Figure 5(a). When the MUX is switched to accurate mode by any false propagate bit in the detection window, the actual carry chain is retained by the position of false propagate bit. To obtain a shorter carry chain in the accurate mode, the detection window for the MUX at bit i in MSB should start from bit $i+1$. In this example, we use p_{i+1} and p_{i+2} to detect if there is a carry propagation across two sub-adders, and configure the MUX according to

$$\hat{c}_i \leftarrow \begin{cases} c_i^{prdt}, & \text{if } p_{i+1} \cdot p_{i+2} \text{ is true} \\ c_i, & \text{otherwise.} \end{cases} \tag{8}$$

In approximation mode, the effective carry chain is represented by the blue line in Figure 5(a) and its length is no greater than $L+1$ bits. When the MUX is set to accurate mode, the carry chain is indicated by the red lines in Figure 5(b) and their lengths can be restrained to within $L+2$ bits. Note that the detection overhead here is almost the minimum possible, i.e., only one NAND gate for configuring each MUX.

In Figure 5, we use a 2-bit detection window, which can be generalized to W bits. Then, the error rate for MUX at bit i becomes

$$\widehat{ER}_i^{dar} = ER_i^{prdt} \cdot \prod_{j=1}^{W} P(p_{i+j} = 1) \tag{9}$$

The detection window size W decides the tradeoff between accuracy and the effective carry chain length in accurate mode, which is $L+W$. When W increases, the error rate decreases while the critical path length in accurate mode increases.

V. Experimental Results

A. Experiment Setup and Evaluation

Our SARA, SARA-DAR and several previous designs are synthesized to 32-bit adders by Synopsys Design Compiler using the Nangate 45nm Open Cell Library. The synthesized circuits are placed and routed by Cadence Encounter. The default supply voltage level is 1.25V. To make fair comparisons across architectures, we described all designs by structural modeling in Verilog to reduce the impact of synthesis and optimization. In addition, we set the same supply voltage and no delay constraint on all designs for the same reason.

The evaluation of accuracy configurable adder designs can be subtle and therefore is worth some discussion.

1) **Area**: In the literature, the area sometimes refers to the part of the circuit working in a certain mode, e.g., the circuit for the accurate part is not included in area estimation when evaluating approximation mode. We report the routed layout area of each entire design.
2) **Delay**: Some configurable adders, such as ACA [10] and GeAr [11], implement error correction with pipelining, which sometimes takes multiple clock cycles to determine the complete result. The delay or performance evaluation of such designs is much more complicated than unpipelined designs. Our work is focused on unpipelined implementation, although it can be pipelined. Thus, the reported delay is the maximum combinational logic path delay obtained from Synopsys PrimeTime with consideration of wire delay.
3) **Power**: The power dissipation is estimated by Synopsys PrimeTime considering both static and dynamic power.
4) **Accuracy**: We use PSNR (Peak Signal-to-Noise Ratio), where errors are treated as noise, as a composite accuracy metric for considering both error magnitude and error rate. In addition, the worst case error, which is equivalent to the maximum error magnitude [8], and error rate are also reported. Each error result is from 100K-run Matlab-based Monte Carlo simulation assuming uniform distribution of addends.
5) **Tradeoff**: The tradeoff among the above factors is complex and is difficult to capture in a simple picture. To this end, we use composite metrics including power-delay product (PDP) and iso-delay power.

B. Results of Tradeoff for Different Configurations

We now compare the following accuracy configurable adders:

- GDA [13]: We use the same design as in [13], where each sub-adder has 4 bits. This design can be configured by choosing accurate or predicted carry-out for each sub-adder. The carry prediction at each segment can also be configured to different accuracy levels by using different number of lower-bit addends.
- RAP-CLA [14]: We implement four different designs with carry prediction bit-width from 1 bit to 4 bits, which is reflected in the name. For example, RAP-CLA2 means that each carry prediction is from its 2 lower bits. As in [14], each design can be configured to either only one approximation mode or accurate mode.

- SARA: This is our proposed design and we evaluate sub-adder bit-width of 4 bits and 8 bits, referred to as SARA4 and SARA8, respectively.

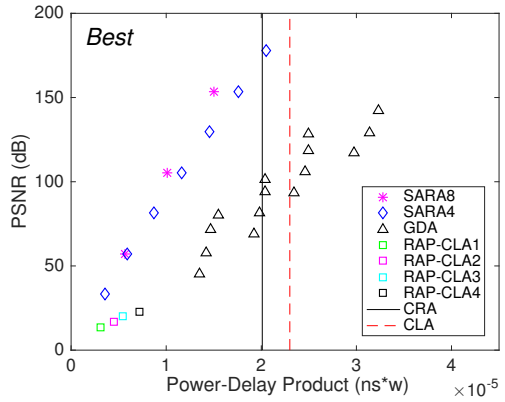

Fig. 6. PSNR vs. power-delay product.

The main result is shown in Figure 6, where each point is from one configuration of one design. The computation accuracy is evaluated by PSNR while the conventional design objectives are characterized by PDP. A design and configuration is ideal if it has large PSNR but low PDP, i.e., the northwest corner of the figure. The PDPs of two classic accurate designs, CRA and CLA, are indicated by the two vertical lines as their PSNR is near infinity. Evidently, the best solutions are from SARA4 and SARA8. At 100 dB PSNR, the PDP of SARA4 and SARA8 is about a half of GDA or CRA. The solutions from RAP-CLA, the latest previous work, are also largely dominated by SARA in PSNR-PDP tradeoff.

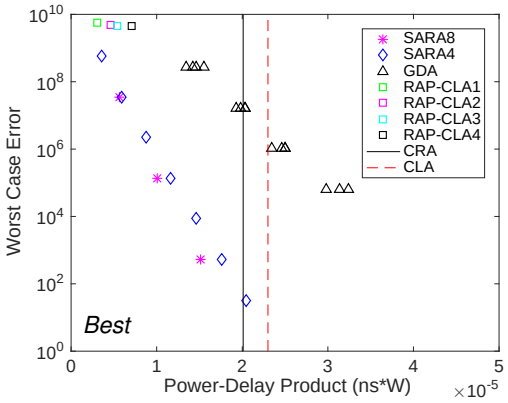

Fig. 7. The worst case error vs. power-delay product.

C. Results of Tradeoff for Delay-Adaptive Reconfiguration

We now evaluate the SARA-DAR design, where the configuration decision has already been made. Here, the natural choice is to additionally compare the design with static approximate adders, where no configuration is needed. Static approximate adder designs including ETAII [6], FICTS [8] and AFICTS [8] are implemented in the experiment. Seven SARA-DAR designs are obtained based on seven configurations of SARA4. That is, if a MUX at bit i is configured to accurate carry in SARA4, bit i of corresponding SARA-DAR is hard-wired to accurate carry without using MUX. When bit j in SARA4 is in approximation mode, bit j of corresponding SARA-DAR uses the delay-adaptive reconfiguration.

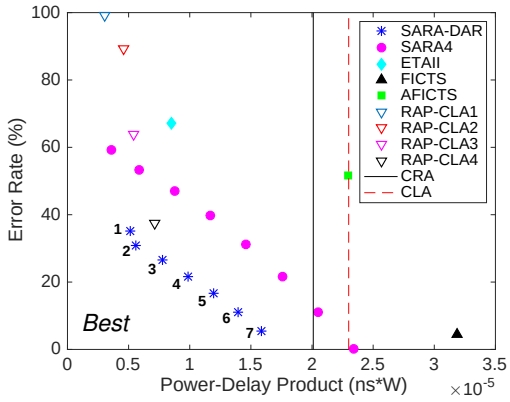

Fig. 8. Error rate vs. power-delay product.

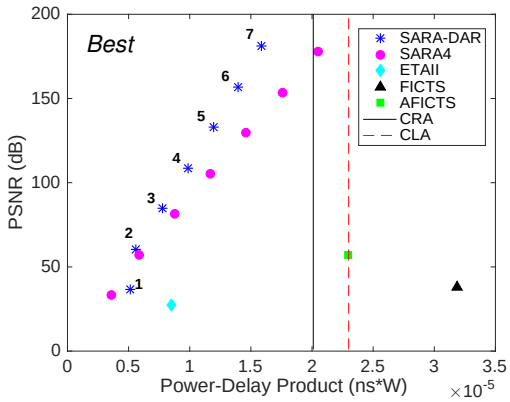

Fig. 9. PSNR vs. power-delay product.

Figures 8 and 9 show the tradeoff between error rate and PSNR, respectively, and the PDP. The error rate of SARA4 is mostly lower than RAP-CLA. By using delay-adaptive reconfiguration, SARA-DAR often further reduces error rate and PDP, and also greatly outperforms the static approximate adders in both error rate and PDP. In Figure 9, SARA-DAR also demonstrates the overall best PSNR-PDP tradeoff.

D. Results of Iso-delay Power and Area

Although power-delay product results have been shown in Sections V-B and V-C, the tradeoff between power and delay is still unclear. The power-delay tradeoff can be obtained by different accuracy configurations or varying supply voltages. Different combinations of configurations and voltages may lead to overwhelming volume of results, which are difficult to interpret, especially when implication to accuracy is involved at the same time. Thus, we indicate the tradeoff by investigating the iso-delay power, which is the power of each circuit tuned to the same critical path delay (0.82ns) by voltage scaling. The results are shown in Figure 10. In general, SARA4, SARA8 and SARA-DAR can achieve much

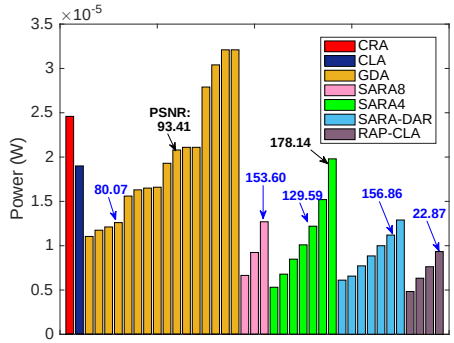

Fig. 10. Iso-delay power comparison. The numbers are PSNR.

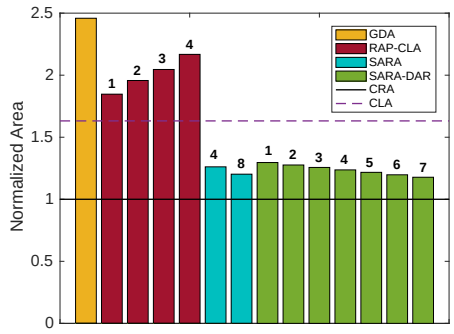

Fig. 11. Area comparison.

lower power than CRA. Although GDA and RAP-CLA seem to provide low power, their PSNR is lower than our designs.

Last but not least, we compare area of these designs in Figure 11. As expected, GDA and RAP-CLA have greater area than CLA while area of our designs is significantly smaller than CLA. On average, the area of SARA is 39% smaller than that of RAP-CLA and 50% smaller than GDA.

E. Results of DCT Computation in Image Processing

The different adder designs are applied to DCT computation [15] in JPEG image compression. We replace the adders in the DCT circuit [15] with one type of approximate adder among GDA, RAP-CLA, SARA4, SARA8 and SARA-DAR. For low (high) frequency components of images, we configure the adders to more accurate (approximate) mode, because the human eye is more sensitive to low frequency errors.

The image processing results are demonstrated in Figure 12. SARA-DAR has the highest PSNR (39.45dB) among all configurable adders, which is close to the quality of accurate adder. Comparing SARA8 with GDA, they have similar PSNR and similar delay, but SARA8 has less power consumption according to the analysis in the previous section. SARA-DAR achieves better image quality than SARA4, but might result in more power due to additional logics for self-configuration.

VI. Conclusion

A simple accuracy reconfigurable adder (SARA) design is proposed. It has significantly lower power-delay product than the latest previous work when comparing at the same accuracy level. In addition, SARA has considerable lower area overhead than almost all the previous works. The accuracy-power-delay efficiency is further improved by a delay-adaptive reconfiguration technique.

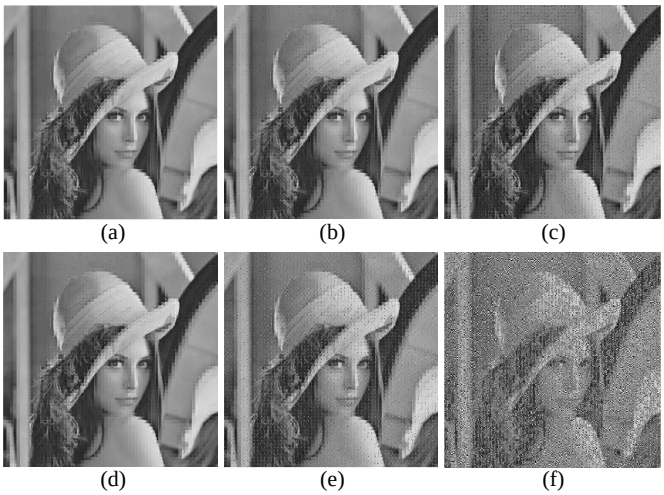

Fig. 12. Image quality comparison: (a) accurate adder, PSNR: 39.85dB; (b) SARA4, PSNR: 38.32dB; (c) SARA8, PSNR: 35.33dB; (d) SARA-DAR, PSNR: 39.45dB; (e) GDA, PSNR: 34.53dB; (f) RAP-CLA, PSNR: 33.38dB.

Acknowledgment

This project is partially supported by NSF(CCF-1255193, CCF-1525749 and CCF-1525925). The authors would like to thank Dr. Duncan M. Walker from Texas A&M University for helpful discussions.

References

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *IEEE European Test Symposium*, pp. 1–6, 2013.

[2] V. Chippa, A. Raghunathan, K. Roy, and S. Chakradhar, "Dynamic effort scaling: Managing the quality-efficiency tradeoff," in *DAC*, pp. 603–608, 2011.

[3] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *TCAD*, vol. 32, no. 1, pp. 124–137, 2013.

[4] S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits," in *DATE*, pp. 1367–1372, 2013.

[5] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *DATE*, pp. 1257–1262, 2012.

[6] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *ISIC*, pp. 69–72, 2009.

[7] J. Hu and W. Qian, "A new approximate adder with low relative error and correct sign calculation," in *DATE*, pp. 1449–1454, 2015.

[8] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in *ICCAD*, pp. 728–735, 2012.

[9] S. Mazahir, O. Hasan, R. Hafiz, M. Shafique, and J. Henkel, "An area-efficient consolidated configurable error correction for approximate hardware accelerators," in *DAC*, pp. 1–6, 2016.

[10] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *DAC*, pp. 820–825, 2012.

[11] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in *DAC*, pp. 1–6, 2015.

[12] V. Benara and S. Purini, "Accurus: A fast convergence technique for accuracy configurable approximate adder circuits," in *IEEE Computer Society Annual Symposium on VLSI*, pp. 577–582, 2016.

[13] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *ICCAD*, pp. 48–54, 2013.

[14] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, accepted.

[15] M.-T. Sun, T.-C. Chen, and A. M. Gottlieb, "VLSI implementation of a 16*16 discrete cosine transform," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 4, pp. 610–617, 1989.