An Analytical Framework for Evaluating the Error Characteristics of Approximate Adders

Cong Liu, Jie Han, *Member, IEEE,* and Fabrizio Lombardi, *Fellow, IEEE*

***Abstract*—Approximate adders have been considered as a potential alternative for error-tolerant applications to trade off some accuracy for gains in other circuit-based metrics, such as power, area and delay. Existing approximate adder designs have shown substantial advantages in improving many of these operational features. However, the error characteristics of the approximate adders still remain an issue that is not very well understood. A simulation-based method requires both programming efforts and a time-consuming execution for evaluating the effect of errors. This method becomes particularly expensive when dealing with various sizes and types of approximate adders. In this paper, a framework based on analytical models is proposed for evaluating the error characteristics of approximate adders. Error features such as the error rate and the mean error distance are obtained using this framework without developing functional models of the approximate adders for time-consuming simulation. As an example, the estimate of peak signal-to-noise ratios (PSNRs) in image processing is considered to show the potential application of the proposed analysis. This analytical framework provides an efficient method to evaluate various designs of approximate adders for meeting different figures of merit in error-tolerant applications.**

Keywords*—*Approximate computing, approximate adder, PSNR estimate, mean error distance, image processing.

I. Introduction

Approximate computing has become a promising technique to reduce the power, area and delay constraints in VLSI design, albeit at the expense of a loss in computational accuracy [1]. This technique is applicable to error-tolerant applications such as multimedia, mining and recognition [2]. Generally, there are two methodologies for reducing accuracy by approximation. The first methodology uses a voltage-over-scaling (VOS) technique for CMOS circuits to save power, while also introducing errors into the circuit [3]–[5]. The second methodology is based on redesigning a logic circuit into an approximate version. While the VOS technique is applicable to most circuits for error-tolerant applications, an approximate redesign requires to consider the different functionalities of logic circuits. As one of the simplest, but key components of arithmetic circuits, adders have attracted an extensive interest for redesigning and implementing approximate schemes. Approximate adders have been proposed by using a reduced number of transistors [6], [7] and by truncating the carry propagation chain for a speculation-based operation [8]–[12].

The approximate speculative designs achieve a better performance in terms of area, power and delay compared to conventional (exact) adders. New metrics and simulation-based approaches have been proposed to model and evaluate approximate adders according to specific computational features [2], [13]–[15]. Monte Carlo or exhaustive simulation approaches have been employed to acquire data for analysis. This class of approaches are however time-consuming and require building functional models of the approximate designs. To improve efficiency, a mathematical characterization of the arithmetic accuracy of approximate adders is then required for a better understanding of the design prior to a simulation-based evaluation.

In addition to generic metrics (such as the error rate), application specific measures (ASMs) such as the peak signal-to-noise ratio (PSNR) for image processing are well suited in practice. Without an approach to modeling the relationship between the generic metrics and the ASMs, extensive programming and simulation efforts are required to obtain the ASMs for assessing the impact and the potential of approximate computing in different applications. Therefore, an effective approach to obtain or estimate the ASMs from generic error metrics is needed; however, there are no formal methodologies or analytical approaches for these purposes in the technical literature.

In this paper, an analytical framework is proposed to assess the arithmetic accuracy, i.e. the error rate (ER) and mean error distance (MED), of approximate adders. Three types of approximate adders are considered and their error features are compared using the proposed analysis. The revealed error characteristics provide insights into the quality of an appropriate adder for achieving a desired operational accuracy. As an example of ASMs, the PSNR in image processing is considered. A model is presented for estimating the PSNR from the MED obtained from the proposed framework; experimental results show that the estimated PSNRs are very close to the PSNRs obtained by simulation. The utilization of the proposed framework to PSNR estimate provides an analytical approach for assessing and designing a feasible approximate image processing system based on approximate adders.

The major contributions of this paper are as follows.

- An analytical framework is developed for modeling and evaluating the error characteristics of three types of approximate adders found in the technical literature.
- A comparative study is performed for various approximate adders using different carry speculation schemes to gain an insight into the qualitative features of a design

C. Liu and J. Han are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2V4. (Email: cong4@ualberta.ca, jhan8@ualberta.ca)

F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, USA. (E-mail: lombardi@ece.neu.edu)

Manuscript received Oct. 22, 2013

with respect to several error metrics.
- An analytical approach is developed to model the relationship between the PSNR and the MED obtained from the proposed framework in approximate adder-based image processing. This approach can effectively estimate the PSNR from the approximate adders used in an image precessing application.

The organization of this paper is as follows. Section II reviews the approximate designs applicable to the framework proposed in this paper. Section III describes the analysis for modeling the error characteristics of the approximate adders. Discussion follows in Section IV. Section V presents the comparison of approximate adder designs using the proposed framework; as an application, the PSNR estimate in image processing is investigated. Conclusion is given in section VI.

II. Review of Existing Approximate Adders

A. The Speculative and Almost Correct Adder (ACA)

The so-called almost correct adder (ACA) [9] is based on the speculative adder design in [8]. The ACA utilizes insufficient information, i.e. k LSBs for predicting the sum of each bit in an n-bit adder ($n > k$). The same illustration (Fig. 1) as in [15] is used for the ACA (and the ESA in the following subsection). In Fig. 1, four bits (i.e. $k = 4$) are used to calculate each bit in the sum of an n-bit adder. The identical vertical rectangular blocks on the top denote the inputs, while the horizontal rectangles under them show the carry propagation paths for each sum bit. This design is based on the observation that the carry propagation chain is usually shorter than n, i.e. in practice, the truncation of the chain up to some length has a very low probability to be erroneous.

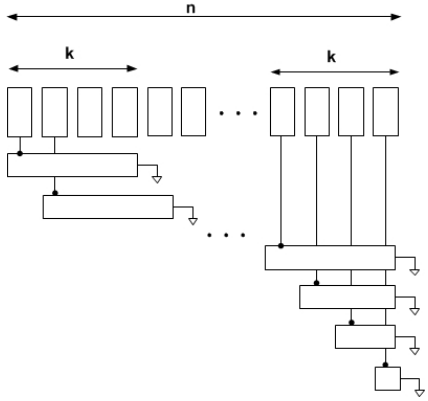


Fig. 1. The almost correct adder. n: the adder size; k: the maximum carry chain length.

B. The Equal Segmentation Adder (ESA)

A dynamic segmentation and error compensation (DSEC) scheme is presented in [5] for an approximate adder design. This approximate adder consists of several sub-adders of different sizes divided from an n-bit adder; each of the sub-adders operates in parallel and has a truncated carry input. For convenience, but with no loss in correctness, sub-adders of equal size are considered in this manuscript. Moreover the error compensation part [5] is neglected because the focus of this manuscript is on analyzing the approximate operation. Thus, a simplified DSEC adder referred to as an equal segmentation adder (ESA) is analyzed in this paper (Fig. 2).

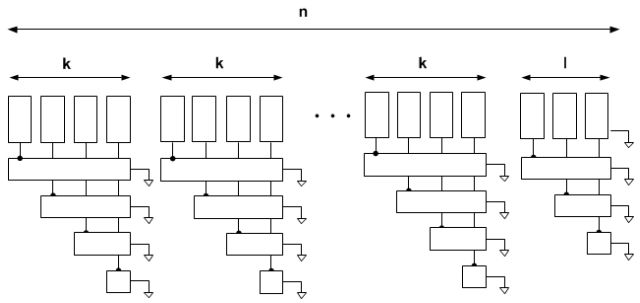


Fig. 2. The equal segmentation adder. n: the adder size; k: the maximum carry chain length; l: size of the first sub-adder ($l \leq k$).

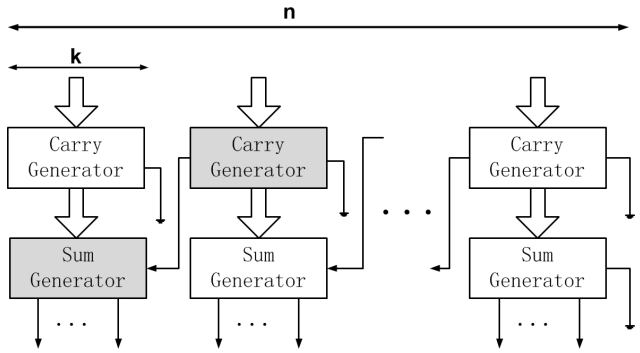


Fig. 3. Block Diagram of error-tolerant adder type II. n: the adder size; k: half of the maximum carry chain length

C. The Error-Tolerant Adder Type II (ETAII), the Speculative Carry Select Adder (SCSA) and the Accuracy-Configurable Approximate Adder

The ETAII is also based on the truncation of the carry propagation chain and the segmentation of a full-sized adder [12]. Compared to the ESA, the predicted carry input for each segmented k-bit sub-adder (or the sum generator in Fig. 3) is generated by k LSBs. The ETAII has an improved accuracy compared to the ESA, because it uses more information to predict the carry when the same k is used. In the so-called speculative carry select addition (SCSA) [10], an n-bit adder is first divided into $\lceil \frac{n}{k} \rceil$ sub-adders (also known as "window adders"); each sub-adder consists of two k-bit adders: adder0 and adder1(Fig. 4). The only difference between the two k-bit adders is the carry input; the carry of adder0 is "0" while it is "1" for adder1. The output of the ith sub-adder is selected from adder0 and adder1 based on the carry out signal generated by the $(i-1)$th sub-adder. The carry out of each sub-adder is generated based on the k-bit in the sub-adder rather than all previous bits. Therefore, the carry selection process is

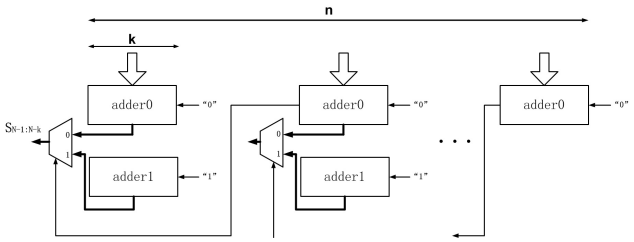


Fig. 4. The speculative carry selection adder. n: the adder size; k: the maximum carry chain length.

still approximate and faster than a traditional carry selection scheme. Even though the SCSA and the ETAII have different circuit implementations, they share a similar functionality if their sub-adders have the same length. The SCSA and the ETAII generate the same carry signal for each sub-adder (or the Sum Generator in the ETAII) even though by different circuit implementations. The accuracy-configurable approximate adder proposed in [11] can adjust the accuracy during runtime. For a given accuracy, the approximate configuration of the adder performs a similar function as the ETAII.

For establishing the error characteristics, adders with the same functionality are considered to be the same type. For example, an ETAII and a SCSA with the same k and n values generate the same output for the same inputs. Thus, they have the same error characteristics. However, characteristics related to a circuit implementation such as delay and power are not necessarily the same. Some of the approximate adders also have an error correction circuit that permits an additional accurate operation mode; only the approximate operation of each adder is considered in this paper.

III. Error Analysis

A. Preliminaries

1) Metrics: The *error distance* (ED) and the *mean error distance* (MED) are proposed in [13] to evaluate the arithmetic performance of approximate circuits. For an approximate adder, ED is defined as the absolute value of the difference between the accurate and approximate sums, i.e.,

$$ED = |S^{'} - S|, \tag{1}$$

where $S^{'}$ is the sum of the approximate adder and S is the sum of an accurate adder. MED is defined as the average ED for a given set of input vectors, i.e.,

$$MED = E[ED] = \sum_{i} ED_i P(ED_i), \tag{2}$$

where $P(ED_i)$ is the probability of ED_i. The *error rate* (ER) is defined as the percentage of erroneous outputs among all outputs [16], i.e.,

$$ER = \sum_{i} P(ED_i), \text{ for any } ED_i \neq 0. \tag{3}$$

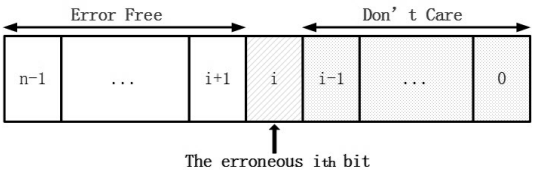


Fig. 5. Sub error set $\mathbf{\Pi_i}$ in the ACA

The above metrics (ER and MED) are of interest for evaluating the arithmetic performance of approximate adders. In the following section, an analytical method is presented to calculate these metrics for different types of adders.

2) Notation: The notation used in the error analysis throughout this paper is introduced next. We consider an n-bit approximate adder with inputs A, B and C_0, and an output S. A_i, B_i, S_i are the corresponding input and output bits at the ith position. C_i is the carry to be added to the ith bit. Let $p_i = P(C_i = 1) = P(A_{i-1}B_{i-1} = 1) + P(A_{i-1} \oplus B_{i-1} = 1, C_{i-1} = 1)$, for uniformly-distributed inputs, $p_i = \frac{1}{4} + \frac{1}{2}p_{i-1}$, which leads to

$$p_i = \frac{1}{2} + \frac{1}{2^i}(p_0 - \frac{1}{2}), \tag{4}$$

where p_0 is the probability that the initial carry bit is 1. Assume $p_0 = 0$, then

$$p_i = \frac{1}{2}(1 - \frac{1}{2^i}). \tag{5}$$

Let $\bar{X}_i$ and $\tilde{X}_i$ denote the events that the ith approximate sum bit is the same as or different from the ith exact sum bit, respectively, i.e., $\bar{X}_i = \{S_i = S_i^{'}\}, \tilde{X}_i = \{S_i \neq S_i^{'}\}$. An $\mathbf{X}$ vector consisting of $\bar{X}_i$'s or $\tilde{X}_i$'s is used to denote a set of outputs of the approximate adder compared to the accurate one. For example, for a 4-bit approximate adder, $\{\bar{X}_3\bar{X}_2\tilde{X}_1\}$ denotes a set of outputs in which S_1 is incorrect, both S_2 and S_3 are correct and S_0 could be either correct or incorrect.

B. ACA Error Characteristics

A *universal error set* is said to be formed by all possible error patterns of the ACA. To calculate the MED of an n-bit ACA, this error set is divided into n disjoint subsets and then the MED is calculated for each subset. The universal error set, denoted by $\mathbf{\Pi}$, is divided into the subsets of $\mathbf{\Pi_i}(i = 0, 1, ..., n-1)$, i.e.,

$$\mathbf{\Pi} = \cup \mathbf{\Pi_i}, \tag{6}$$

where $\mathbf{\Pi_i} = \{\bar{X}_{n-1}, \bar{X}_{n-2}, ..., \tilde{X}_i\}$. The error patterns in $\mathbf{\Pi_i}$ are those whose ith bit is erroneous, while the more significant bits are correct and the less significant bits are "don't cares", i.e., they can be either correct or erroneous, as shown in Fig. 5. Based on the error set division, the mean error distance of the approximate adder is calculated as:

$$MED = \sum_{i} E[|e_i|], \tag{7}$$

where $E[|e_i|]$ is the mean error distance of the subset $\mathbf{\Pi_i}$.

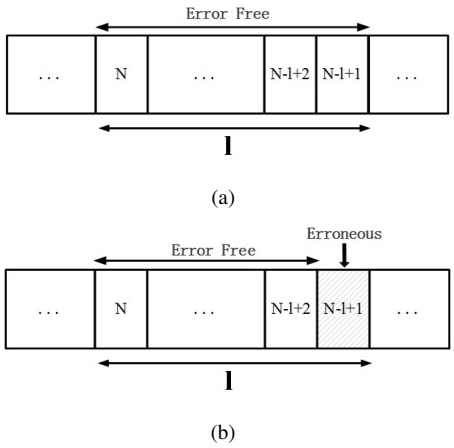


Fig. 6. (a) $\bar{P}_l$, (b) $\tilde{P}_l$.

In the subset $\mathbf{\Pi_i}$, the errors in the ith bit (i.e. $\pm 2^i$) are dominant. Moreover, some of the errors in the lower bits can cancel each other. Therefore, the errors in $\mathbf{\Pi_i}$ have on average a magnitude of approximately 2^i; so the MED of $\mathbf{\Pi_i}$ is calculated as

$$E[|e_i|] \approx 2^i q_i, \tag{8}$$

where q_i is the probability that the error patterns fall in $\mathbf{\Pi_i}$. The total mean error distance is then given by

$$MED = \sum_i 2^i q_i. \tag{9}$$

The error rate of the ACA is given by

$$ER = \sum_i q_i. \tag{10}$$

Next the calculation of q_i is presented. Let $\bar{P}_l$ be the probability that l consecutive bits in the approximate sum are correct (Fig. 6(a)) and $\tilde{P}_l$ be the probability that $l-1$ consecutive bits in the approximate sum are correct, but the next lower bit is erroneous (Fig. 6(b)), i.e.,

$$\bar{P}_l = P(\bar{X}_N, \bar{X}_{N-1}, ..., \bar{X}_{N-l+2}, \bar{X}_{N-l+1}), \tag{11}$$

$$\tilde{P}_l = P(\bar{X}_N, \bar{X}_{N-1}, ..., \bar{X}_{N-l+2}, \tilde{X}_{N-l+1}), \tag{12}$$

Furthermore, let $\bar{Q}_l$ denote the conditional probability that one approximate sum bit is correct given that $l-1$ consecutive lower bits in the approximate sum are correct, and $\tilde{Q}_l$ denote the conditional probability that one approximate sum bit is correct given that $l-2$ consecutive lower bits in the approximate sum are correct but the next lower bit is erroneous, i.e.,

$$\bar{Q}_l = P(\bar{X}_N | \bar{X}_{N-1}, ..., \bar{X}_{N-l+2}, \bar{X}_{N-l+1}), \tag{13}$$

$$\tilde{Q}_l = P(\bar{X}_N | \bar{X}_{N-1}, ..., \bar{X}_{N-l+2}, \tilde{X}_{N-l+1}), \tag{14}$$

Note that in these probabilities, only the length of the pattern is important, i.e., the exact index of each pattern, N, is less important.

Let the truncated carry propagation length be k; $\tilde{P}_l (l \leq k)$ is calculated first. Further denote the accurate sum as S and the inaccurate sum as $S^{'}$. As $S^{'}_{N-l+1}$ is the most significant erroneous bit, i.e., all higher bits in the sum are correct, the input carry, $C^{'}_{N-l+1-k}$, to calculate $S^{'}_{N-l+1}$, must propagate all the way to $S^{'}_{N-l+1}$, i.e., $P_i = A_i \oplus B_i = 1$, for $i = N-l-k+2, N-l-k+3, ..., N-l$. Also, $P_{N-l+1} = 0$, because, otherwise, the wrongly estimated carry would propagate to $S^{'}_{N-l+2}$. As $P_{N-l+1} = 0$, it is sufficient to show that $S^{'}_{N-l+2}, S^{'}_{N-l+3}, ..., S^{'}_{N}$ are correct, because $P_{N-l+1} = 0$ prevents the error from propagating to the higher $k-1$ bits. Therefore ,

$$\begin{aligned}\tilde{P}_l = P(&P_{N-l-k+2}, ..., P_{N-l} = 1, P_{N-l+1} = 0,\\ &C^{'}_{N-l-k+2} \neq C_{N-l-k+2}) = \tfrac{1}{2^{k+1}}, \quad l \geq 2.\end{aligned} \tag{15}$$

Let $\tilde{P}_1 = P(\tilde{X}_N)$, then

$$\begin{aligned}\tilde{P}_1 = P(&P_{N-k+1}, ..., P_{N-2} = 1, P_{N-1} = 0,\\ &C^{'}_{N-k+1} \neq C_{N-k+1}) = \tfrac{1}{2^k}.\end{aligned} \tag{16}$$

Thus,

$$\tilde{P}_l = \begin{cases} = \frac{1}{2^{k+1}}, l \geq 2. \\ \frac{1}{2^k}, l = 1. \end{cases} \tag{17}$$

Since $\bar{P}_1 = \tilde{P}_2 + \bar{P}_2 = \tilde{P}_2 + \tilde{P}_3 + \bar{P}_3 = ... = \tilde{P}_2 + \tilde{P}_3 + ... + \tilde{P}_l + \bar{P}_l$, then

$$\bar{P}_l = \bar{P}_1 - \sum_{i=2}^{l} \tilde{P}_i, \tag{18}$$

where $\bar{P}_1 = 1 - \tilde{P}_1$. For $l \leq k$,

$$\tilde{Q}_l = \frac{\tilde{P}_l}{\bar{P}_{l-1}}, l \leq k. \tag{19}$$

For $l > k$, since only k bits are used to calculate the current sum, the bits that are less significant than these k bits, have no influence on $\tilde{Q}_l$. Thus,

$$\tilde{Q}_l = \bar{Q}_k = \frac{\bar{P}_k}{\bar{P}_{k-1}}, l > k. \tag{20}$$

From all of the above, we obtain

$$\tilde{Q}_l = \begin{cases} \frac{\tilde{P}_l}{\bar{P}_{l-1}} = 1, k \geq l > 2, \\ \frac{\tilde{P}_2}{\bar{P}_1} = \frac{1}{2}, l = 2, \\ \frac{\bar{P}_k}{\bar{P}_{k-1}} = \frac{2^{k+1}-k-1}{2^{k+1}-k}, l > k. \end{cases} \tag{21}$$

This leads to

$$\begin{aligned} q_i &\approx P(\bar{X}_N, \bar{X}_{N-1}, ..., \bar{X}_{i+1}, \tilde{X}_i) \\ &= P(\bar{X}_N | \bar{X}_{N-1}, ..., \bar{X}_{i+1}, \tilde{X}_i) P(\bar{X}_{N-1}, ..., \bar{X}_{i+1}, \tilde{X}_i) \\ &= P(\bar{X}_N | \bar{X}_{N-1}, ..., \bar{X}_{i+1}, \tilde{X}_i) P(\bar{X}_{N-1} | \bar{X}_{N-2}, ..., \bar{X}_{i+1}, \tilde{X}_i) \\ &\quad P(\bar{X}_{N-2}, ..., \bar{X}_{i+1}, \tilde{X}_i) \\ &= P(\bar{X}_N | \bar{X}_{N-1}, ..., \bar{X}_{i+1}, \tilde{X}_i) P(\bar{X}_{N-1} | \bar{X}_{N-2}, ..., \bar{X}_{i+1}, \tilde{X}_i) \\ &\quad ... P(\bar{X}_{i+1} | \tilde{X}_i) P(\tilde{X}_i) \\ &= \tilde{Q}_{N-i+1} \tilde{Q}_{N-i} ... \tilde{Q}_2 \tilde{P}_1. \end{aligned} \tag{22}$$

Hence, (17), (21) and (22) can be used to calculate q_i in the ACA error analysis.

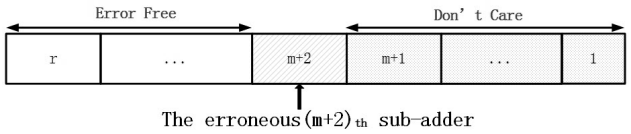


Fig. 7. An example of the error in ESA dominated by the $(m+2)$th sub-adder.

C. ESA Error Characteristics

Consider an n-bit ESA divided into r ($r = \lceil \frac{n}{k} \rceil - 1$) sub-adders of equal size k and 1 sub-adder of size $l = n - kr$. Thus there are $(r+1)$ sub-adders in total. Since the lowest sub-adder (i.e. the first sub-adder) is always error-free, only the higher r sub-adders that can be erroneous are considered. Initially the error in the $(m+2)$th sub-adder ($m = 0, 1, ..., r-1$) is considered; this is always 2^{mk+l}, as introduced by the wrong estimate of the input carry to this sub-adder.

When the speculative carry into each sub-adder is truncated to 0, the error rate of the $(m+2)$th sub-adder is given by

$$\begin{aligned} &P(error = -2^{mk+l}) = P(C^{'}_{mk+l} < C_{mk+l}) \\ &= p_{mk+l} \approx \tfrac{1}{2}. \end{aligned} \tag{23}$$

The error rate of each sub-adder (except for the first one) is approximately $\frac{1}{2}$. All sub-adders are independent, because there is no connection between them; so, the error rate of the entire adder is

$$ER = 1 - (1 - \frac{1}{2})^r = 1 - (\frac{1}{2})^r. \tag{24}$$

An approximate method is then introduced to calculate the mean error distance of the ESA. Since the errors in a lower sub-adder are significantly smaller than those in a higher sub-adder, the error magnitude of the approximate adder is dominated by the highest erroneous sub-adder. For example, for 8-bit sub-adders, the error in the third sub-adder is 256 times greater than the error in the second sub-adder. Therefore, if a sub-adder is erroneous, the errors in the lower sub-adders become insignificant and thus, they can be ignored. Hence, for an ESA with $(r+1)$ sub-adders, the error magnitudes can be approximately divided into r levels, i.e. $\{2^l, 2^{k+l}, ..., 2^{(r-1)k+l}\}$. Fig. 7 shows a case when the error is dominated by the $(m+2)$th sub-adder, and the error probability is given by

$$P(error = -2^{mk+l}) \approx (\tfrac{1}{2})^{r-m-1} \times \tfrac{1}{2}. \tag{25}$$

The mean error distance is therefore given by

$$\begin{aligned} &MED = \sum_{m=0}^{r-1} 2^{mk+l} (\tfrac{1}{2})^{r-m-1} \times \tfrac{1}{2} \\ &= \tfrac{2^{kr+r}-1}{2^{k+1}-1} \times 2^{l-r} \approx 2^{n-k-1}. \end{aligned} \tag{26}$$

D. ETAII Error Characteristics

Similar to the analysis of the ESA, the ETAII uses the same partition scheme and its error analysis starts with the evaluation of the error rate in the $(m+2)$th sub-adder. The $(m+1)$th sub-adder generates the approximate carry, $C^{'}_{mk+l}$, to the $(m+2)$th sub-adder based on the assumption that the input carry to itself, $C^{'}_{(m-1)k+l}$, is 0. Thus, when the exact carry $C_{(m-1)k+l}$ is 1 and propagates through the $(m+1)$th sub-adder, the carry generated by the $(m+1)$th sub-adder is erroneous; thus, this results in an error in the $(m+2)$th sub-adder. Hence, the error rate of the $(m+2)$th sub-adder is given by

$$\begin{aligned} &P(error = -2^{mk+l}) = P(C^{'}_{mk+l} < C_{mk+l}) \\ &= (\tfrac{1}{2})^k p_{mk+l} = (\tfrac{1}{2})^{k+1}(1 - \tfrac{1}{2^{mk+l}}) \approx \tfrac{1}{2^{k+1}}. \end{aligned} \tag{27}$$

In the ETAII, the first and second sub-adders are always error-free, so there are totally $(r-1)$ sub-adders that can be erroneous. Hence, the ER of the ETAII is given by

$$ER \approx 1 - (1 - \frac{1}{2^{k+1}})^{r-1}. \tag{28}$$

Similarly as for the ESA, the error in the ETAII is approximately divided among the $(r-1)$ levels. Hence

$$P(error = -2^{mk+l}) = (1 - \frac{1}{2^{k+1}})^{r-m-1} \times \frac{1}{2^{k+1}}, \tag{29}$$

and

$$\begin{aligned} &MED = \sum_{m=1}^{r-1} 2^{mk+l} (1 - \tfrac{1}{2^{k+1}})^{r-m-1} \times \tfrac{1}{2^{k+1}} \\ &= \tfrac{2^{kr-k+l} - 2^l (1 - \frac{1}{2^{k+1}})^{r-1}}{2^{k+1} - 2 + 2^{-k}} \end{aligned} . \tag{30}$$

(30) can be simplified by ignoring $-2+2^{-k}$ in the denominator and $2^l(1 - \frac{1}{2^{k+1}})^{r-1}$ in the numerator, hence

$$MED \approx 2^{n-2k-1}. \tag{31}$$

E. Error Characteristics Under Different Carry Estimation Methods

All approximate adders considered so far use a fixed carry (with a value of 0) estimate such that the carry propagation chain is truncated. For example, for $k = 10$ in the ACA, an assumed input carry $C^{'}_1 = 0$ is used to calculate $S^{'}_{10}$. This assumption may lead to under-estimated results and the average error is non-zero. A straightforward solution to avoid the non-zero average error encountered in a method using a fixed carry is to use a random carry. If the input operands are uniformly distributed, one of the less significant input bits can be used as the random carry signal [14]. For example, $C^{'}_1 = A_0$ or $C^{'}_1 = B_0$ can be used to estimate C_1 in the previous ACA example. Hereafter, these two methods are referred to as the *fixed carry estimate* and *1-LSB carry estimate*, respectively.

In the previous section, the error characteristics of approximate adders with a fixed carry estimate have been discussed. Next, the error characteristics under the 1-LSB carry estimate case are considered.

1) ACA: A detailed solution under the 1-LSB carry estimate case is not provided, because it can be obtained in a manner similar to the fixed carry case. When 1-LSB is used to estimate the carry, (17) and (21) are replaced by (32) and (33), respectively, while (22) remains the same.

$$\begin{array}{l}\tilde{P}_l = P(q_{N-l-k+2}, ..., q_{N-l} = 1, q_{N-l+1} = 0, \\ C^{'}_{N-l-k+2} \neq C_{N-l-k+2}) = \frac{1}{3}\frac{1}{2^k}(1 + \frac{1}{2^{2l-1}}), \ \ l \leq k\end{array} \tag{32}$$

$$\tilde{Q}_l = \begin{cases} \frac{\tilde{P}_l}{\tilde{P}_{l-1}} = \frac{2^{2l-1}+1}{4(2^{2l-3}+1)}, k \geq l \geq 2 \\ \frac{\tilde{P}_k}{\tilde{P}_{k-1}} = \frac{1}{4}\frac{9\times 2^{3k-1}-(6k+4)2^{2k-2}-1}{9\times 2^{3k-3}-(6k-2)2^{2k-4}-1}, l > k \end{cases} \tag{33}$$

After q_i is calculated by (22), (9) and (10) can still be used to obtain the MED and ER in the 1-LSB carry estimate case.

2) ESA: The error rate of each sub-adder in the 1-LSB carry estimate case is half of the error rate in the fixed carry case (as discussed later in this manuscript). Thus, the procedure in the previous section can be utilized by changing the error rate of each sub-adder at the beginning. If 1-LSB is used to estimate the truncated carry, (23) is changed to:

$$\begin{array}{l}P(error = 2^{mk+l}) = P(C^{'}_{mk+l} > C_{mk+l}) \\ = P(A_{mk+l-1} = 1, B_{mk+l-1} = C_{mk+l-1} = 0) \\ = \frac{1}{4}(1 - p_{mk+l}) \approx \frac{1}{8},\end{array} \tag{34}$$

and

$$P(error = -2^{mk+l}) = P(C^{'}_{mk+l} < C_{mk+l}) \approx \frac{1}{8}. \tag{35}$$

The mean error distance is then given by

$$\begin{array}{l}MED = \sum_{m=0}^{r-1} 2^{mk+l}(\frac{3}{4})^{r-m-1} \times \frac{1}{4} \\ = \frac{2^{kr}-(\frac{3}{4})^r}{2^{k+2}-3} \times 2^l \approx 2^{n-k-2}.\end{array} \tag{36}$$

The ER is:

$$ER = 1 - (1 - \frac{1}{4})^r = 1 - (\frac{3}{4})^r. \tag{37}$$

3) ETAII: If 1-LSB is used to estimate the truncated carry, the analysis of the ETAII is very similar to the ESA, so the ER and MED for the ETAII are given as follows:

$$ER \approx 1 - (1 - \frac{1}{2^{k+2}})^{r-1}, \tag{38}$$

and

$$\begin{array}{l}MED = \sum_{m=1}^{r-1} 2^{mk+l}(1 - \frac{1}{2^{k+2}})^{r-m-1} \times \frac{1}{2^{k+2}} \\ \approx 2^{n-2k-2}.\end{array} \tag{39}$$

F. Monte Carlo Simulation

1) Error Analysis for Monte Carlo Simulation: Assume the accurate MED is μ and $\hat{\mu}_T$ is an estimate for MED obtained by averaging EDs from T iterations of Monte Carlo simulation. This can be modeled as a Monte Carlo integration approach [17]. The variance of $\hat{\mu}_T$ is

$$var(\hat{\mu}_T) = \frac{v}{T}, \tag{40}$$

where v is the variance of EDs given by

$$v = \sum_i (ED_i)^2 P(ED_i) - \mu^2. \tag{41}$$

For large T, by the Law of Large Numbers,

$$\hat{\mu}_T \sim N(\mu, \frac{v}{T}). \tag{42}$$

For a given confidence level, a parameter z_c can be determined to show the corresponding confidence interval. Therefore the error of $\hat{\mu}_T$ becomes

$$e = \frac{z_c}{\mu}\sqrt{\frac{v}{T}}. \tag{43}$$

For a confidence level of 95%, $z_c = 1.96$. Therefore, for $T = 1,000,000$, the error is

$$e = \frac{0.00196\sqrt{v}}{\mu} = 0.00196CV, \tag{44}$$

where $CV = \sqrt{v}/\mu$, the coefficient of variation.

(44) will be used in the following to analyze the error for Monte Carlo simulations of three approximate designs with a confidence level of 95%.

Considering (25), (26) and (41), the variance of the EDs of ESA for a fixed carry estimate is given by

$$\begin{array}{l}v = \sum_{m=0}^{r-1} (2^{mk+l})^2(\frac{1}{2})^{r-m-1} \times \frac{1}{2} - (2^{n-k-1})^2 \\ \approx 2^{2n-2k-2}.\end{array} \tag{45}$$

By taking into consideration (26) and (45), the percentage error is 0.196% by (44), i.e., we are 95% confident that a simulated MED is within 0.196% of the true MED. Similarly, the variance of EDs of ESA for the 1-LSB carry estimate is

$$v \approx 3 \times 4^{n-k-2}. \tag{46}$$

The corresponding error is 0.34%.

For ETAII in the fixed carry estimate case, based on (29), (31) and (41), the variance of EDs is

$$\begin{array}{l}v = \sum_{m=1}^{r-1} (2^{mk+l})^2(1 - \frac{1}{2^{k+1}})^{r-m-1} \times \frac{1}{2^{k+1}} - (2^{n-2k-1})^2 \\ \approx 2^{2n-3k-1}.\end{array} \tag{47}$$

The corresponding error for a confidence level of 95% and 1,000,000 iterations of simulation is

$$e = 0.00196\sqrt{2^{k+1}}. \tag{48}$$

In the simulation, the smallest and largest k for ETAII is 4 and 10, thus the error is in the range of 1.11%-8.87%. For ETAII in the 1-LSB carry estimate case, the variance can be obtained in a similar way as in the fixed carry estimate case:

$$v \approx 2^{2n-3k-2}. \tag{49}$$

The corresponding error is

$$e = 0.00196\sqrt{2^{k+2}}. \tag{50}$$

TABLE I. CV (I.E. $\sqrt{v}/\mu$) VALUES FOR ACA

	fixed carry estimate	1-LSB carry estimate
k=6	6.8	7.7
k=7	9.7	11.0
k=8	13.6	15.5
k=9	19.6	21.9
k=10	27.0	31.4
k=11	39.0	43.7
k=12	53.7	63.7

TABLE II. ERRORS OF MONTE CARLO SIMULATION FOR ACA

	fixed carry estimate	1-LSB carry estimate
k=6	1.3%	1.5%
k=7	1.9%	2.2%
k=8	2.7%	3.0%
k=9	3.8%	4.3%
k=10	5.3%	6.1%
k=11	7.7%	8.6%
k=12	10.5%	12.5%

The error is in the range of 1.57%-12.54% for k between 4 and 10.

Since it is difficult to get the theoretical variance for ACA, simulated variances are used. The CVs (i.e. $\sqrt{v}/\mu$) for different k and different carry estimate methods are presented in Table I. According to Table I and (44), the errors of ACA are in the ranges of 1.3%-10.5% and 1.5%-12.5% for the fixed and 1-LSB carry estimates, as shown in Table II.

2) Simulation Results: Fig. 8 and Fig. 9 shows the simulation results and the analytical results for the ACA, ESA and ETAII. The functional models of both accurate and approximate adders are implemented in Matlab. 1,000,000 random input combinations are used to find the MED and ER values.

The simulation and the analytical MEDs are well matched especially for ESA for which the theoretical and simulated curves overlap; there are mainly two sources of discrepancy. The first source is due to the simulation method, i.e. Monte Carlo simulation is not exhaustive. The second source is caused by the approximation used in the analytical framework. As shown in the figures, the MED drops exponentially as k is increased: the MED drops approximately to half of its previous value when k is increased by 1 for all three approximate adders. The difference between MED values with different k values for the same adder is very large; therefore, the small discrepancy between the analytical and simulated results is rather negligible, i.e., the discrepancy will not result in an incorrect assessment of k.

The simulation results for the error rate (ER) are shown in Fig. 9. For the ACA, ER decreases to half of its previous value when the value of k is increased by 1. For any k larger than 9, the ER drops below 2% for both 1-LSB and fixed carry estimates. Due to the design, the ER of the ESA is nearly constant for a value of k in a range between 11 and 15; it starts to decrease significantly when k is 16. In general, the ESA has a very high error rate, even though it has a very small MED. The ETAII can significantly reduce ER. When $k = 6$, for example, the ER of the ETAII with 1-LSB carry estimate is below 2%.

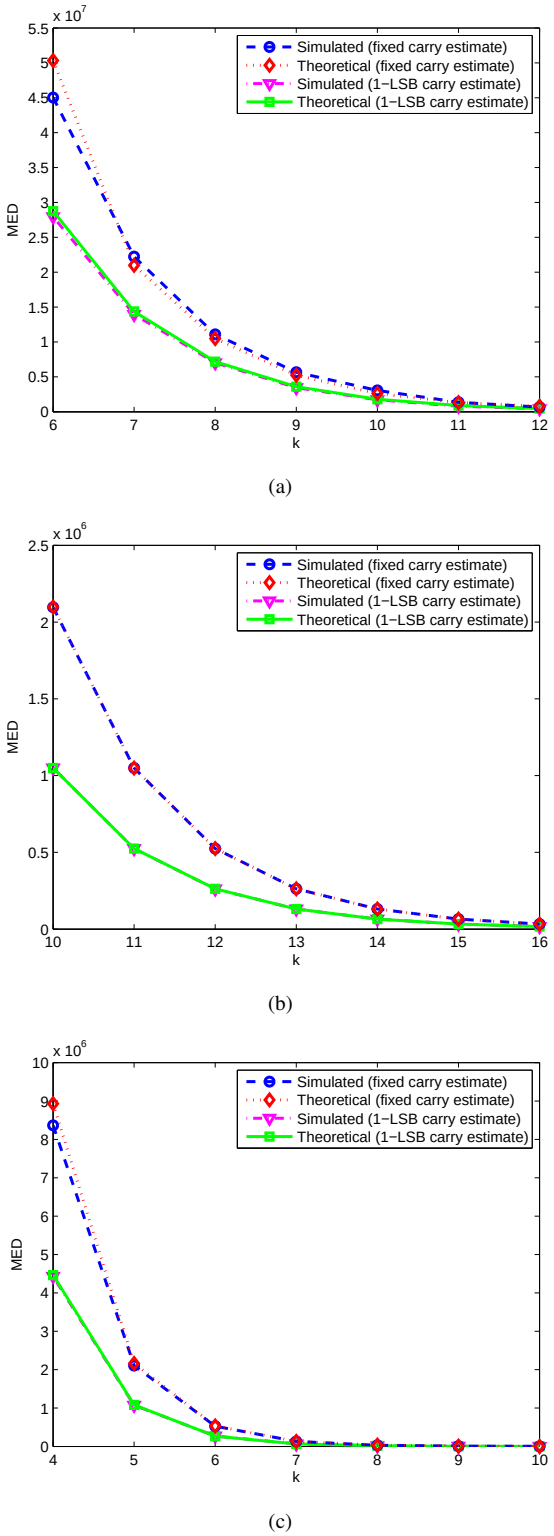
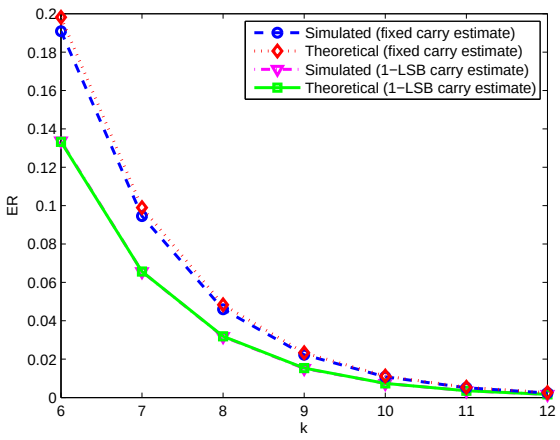
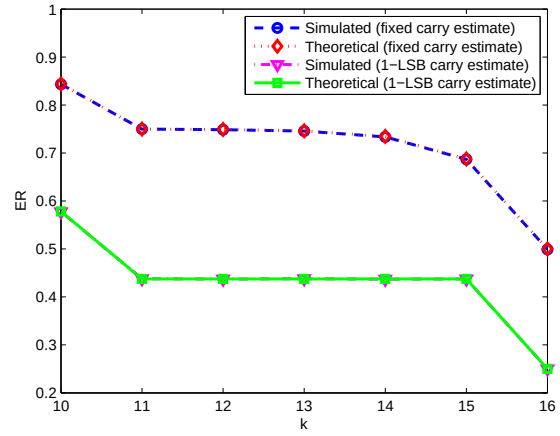


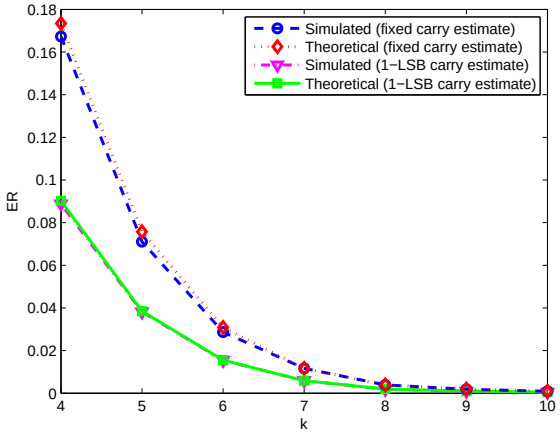
Fig. 8. Simulated and theoretical MED for a 32-bit (a) ACA, (b) ESA, and (c) ETAII.



(a)



(b)



(c)

Fig. 9. Simulated and theoretical ER for a 32-bit (a) ACA, (b) ESA, and (c) ETAII.

IV. Discussion

In this section, different features as related to carry estimate are analyzed for the various approximate adders.

A. Generalizing Carry Estimation Methods

The 1-LSB carry estimate generates rather symmetrical and centralized errors, while a fixed carry tends to generate more biased errors. Another significant advantage of 1-LSB carry estimate is that it reduces the MED. As shown in the simulation results (Fig. 8), the MED of 1-LSB carry estimate is approximately half of the MED of a fixed carry for all three types of approximate adder. Consider the carry estimate accuracy (EA), i.e. the probability that the estimated carry is equal to the exact carry. If the carry is fixed to 0, then $EA = P(C_i = 0) = 1 - p_i \approx \frac{1}{2}$. For the 1-LSB carry estimate, $EA = P(C_i = A_{i-1}) = P(C_i = 1|A_{i-1} = 1)P(A_{i-1} = 1) + P(C_i = 0|A_{i-1} = 0)P(A_{i-1} = 0) = \frac{3}{4}$. It is also intuitively true that the 1-LSB carry estimate method has a higher accuracy than a fixed carry estimate, because it uses more information (when $A_{i-1} = 1$, C_i is more likely to be 1). It is then evident that the use of the 1-LSB to estimate carry reduces the ER as well as the MED.

The approximation in the estimate of the carry signals is a significant issue for an approximate adder design. Intuitively, the utilization of more less significant bits (LSBs) results in a better performance to predict the carry signal. Hereinafter, "k-LSB carry estimate" refers to using the k bits in both A and B that are less significant than the current index to estimate the current carry. The fixed carry approach fails to use the LSBs in the estimate process, while the 1-LSB carry estimate method uses only 1 LSB. For the ETAII and SCSA, if the Sum Generator is considered as a sub-adder, then the Carry Generator (Fig. 3) is the circuitry for the k-LSB carry estimate. Thus, the ETAII uses more LSBs for the carry estimate than the ESA and this increases the accuracy. Nevertheless, the use of more LSBs incurs a larger area overhead and longer delay; a trade-off must be made between the number of LSBs for carry prediction and circuit performance.

Table III shows the truth table of C_{i+1} given 1-LSB, i.e., A_i and B_i, where "U" means "unknown". More LSBs must be used to determine the unknown values. Without the information provided by the additional LSBs, the unknown values have approximately a probability of 0.5 to be either '1' or '0'. Therefore, the best EA using 1-LSB information is $\frac{2+2/2}{4} = \frac{3}{4}$. The 1-LSB estimated carry, i.e., $C'_{i+1} = A_i$, uses a logic function based on 1-LSB information for achieving an EA of $\frac{3}{4}$. Consider the case in which k LSBs are used to estimate the carry C'_m. If the propagates in these k positions are all 1's, i.e., $P_i = A_i \oplus B_i = 1, i = m-1, m-2, ..., m-k$, C'_{m-k} is required to determine C'_m. Therefore, without the information of C'_{m-k}, there are 2^k unknown entries in the truth table of $C_m{}'$ based on previous k LSBs. In the k-LSB carry estimate method, these entries are arbitrarily assigned with certain values (1 or 0), which on average can successfully estimate half (i.e. 2^{k-1}) of the unknown entries. There are totally 2^{2k} input combinations, with 2^k unknown entries and

TABLE III. TRUTH TABLE OF C_{i+1} GIVEN A_iB_i

A_iB_i	C_{i+1}
00	0
01	U
10	U
11	1

$2^{2k} - 2^k$ known entries in the truth table. Thus, the highest EA based on k LSBs is $\frac{2^{2k}-2^{k-1}}{2^{2k}} = 1 - 2^{-k-1}$. However, the utilization of more LSBs for carry prediction increases both the delay and the circuit complexity (i.e. the number of transistors). To reduce complexity, additional "approximate" carry estimates based on k LSBs can be derived by slightly changing some entries in the truth table. Two examples are provided by using 2 LSBs, as given in (51) and (52), with EA being 0.8125 and 0.875, respectively. Note that (52) achieves the highest EA of the 2-LSB carry estimate while (51) has a lower EA with a simpler logic implementation compared to (52).

$$C_{i+1} = A_iB_i + A_iA_{i-1}. \tag{51}$$

$$C_{i+1} = A_iB_i + A_{i-1}(A_i \oplus B_i). \tag{52}$$

B. Comparison Among Different Approximate Adders

The ACA has an area complexity of $O(n \log\log n) \approx O(n)$, and a delay complexity of $O(\log k)$ [9], where n is the size of the adder and k is the maximum carry chain length. The ESA has the same complexity as the ACA if a parallel adder structure such as carry look-ahead (CLA) is used to implement each sub-adder in the ESA. However for the same k, the ACA has a more complicated structure than the ESA and thus, it has a larger area overhead and probably a longer delay. As an example consider $k = 10$ and the 1-LSB carry estimate case. The simulated MEDs of the ACA and the ESA are 1.54×10^6 and 1.05×10^6 respectively. In terms of MED, the ESA is better than the ACA. If area and delay are taken into consideration, the ESA is certainly a better scheme, because it has a smaller MED as well as smaller area and delay. However, the ACA has a smaller ER: the ER of the ACA is 0.0074 while the ER of the ESA is 0.5775, which is 78 times of the ER of the ACA.

In the ESA, $\lceil \frac{n}{k} \rceil$ sub-adders have to be implemented. The area and delay complexities are $\lceil \frac{n}{k} \rceil A(k)$ and $\tau(k)$, respectively, where $A(k)$ and $\tau(k)$ are the area and delay complexities of a k-bit adder. In the ETAII, the area complexity is $(2\lceil \frac{n}{k} \rceil - 2)A(k)$ because every sub-adder is duplicated except for the first and last sub-adders. The delay complexity of the ETAII is $2\tau(k)$ because the critical path contains two k-bit sub-adders in series. The SCSA has the same error characteristics as the ETAII, but it has a different circuit implementation. The delay of the SCSA is $\tau(k)$ with the delay of a multiplexer. Compared to ETAII, the last k-bit sub-adder still needs duplication and $(\lceil \frac{n}{k} \rceil - 1)$ k-bit multiplexers are needed (they are not used in the ETAII). Therefore, the SCSA is approximately two times faster than the ETAII at the cost of an increased area overhead.

TABLE IV. COMPARISON BETWEEN THE ADDERS

	ACA	ESA	ETAII	SCSA
k	10	10	5	10
Delay	O(logk)	O(logk)	O(log2k)	O(logk)
Area	O(nlog(log(n)))	O(n)	O(n)	O(n)
MED ($\times 10^3$)	1540	1050	1074	1.074
ER (%)	0.74	57.8	3.81	0.05

For comparison purposes, choose $k = 10$ for the ESA and $k = 5$ for the ETAII with a 1-LSB carry estimate, because these two implementations have relatively the same delay (even though the ETAII requires more area). The MED of the ESA is 1.05×10^6, while the MED of the ETAII is 1.07×10^6; the ERs are 0.5775 and 0.0381 for the ESA and the ETAII, respectively. Therefore for the compared schemes, the ESA and the ETAII have relatively similar MED, but the ETAII has a significantly smaller ER. Hence, the ETAII tends to generate large error magnitudes, because it has a similar MED as the ESA, but a very small ER.

Consider $k = 10$ for the ACA, the ESA and the SCSA, and $k = 5$ for the ETAII as further examples. These three adders have similar critical path delays. The ESA has the least area overhead, but the largest ER. The ACA has the smallest ER, but it has the largest MED. According to [11], the ACA occupies 36% more area but it incurs in a 30% smaller delay compared to ETAIIM (a modified version of the ETAII), with both adders having the same carry propagation length. The ETAII has a significantly reduced MED compared to the ACA and an acceptable ER of 3.81%. The SCSA is faster than the ETAII at the cost of an increase in area (due to the additional multiplexers). If the design objective is an extremely fast approximate adder, then the SCSA is the best choice among these four approximate adders, because it achieves a similar MED to other approximate adders with a smaller k (i.e. a shorter critical path delay). The SCSA ($k = 10$) has approximately the same delay as the ETAII ($k = 5$). However, the SCSA with $k = 10$ has an extremely small MED and ER: the ER is only 0.05%, while the MED is 10^{-3} of the ETAII ($k = 5$), as shown in Table IV.

V. APPLICATION: IMAGE QUALITY EVALUATION USING PSNRS

A. Peak Signal-to-noise Ratio (PSNR)

PSNR is widely used in many DSP applications (such as image processing) as an important figure of merit. In image processing, if $\mathbf{I}$ is the noise-free image and $\mathbf{K}$ is the noisy image, the PSNR is defined as [18]:

$$PSNR = 20\log(MAX_I/\sqrt{MSE_K}), \tag{53}$$

where MAX_I is the maximum possible pixel value of image $\mathbf{I}$ and MSE is the mean squared error defined as

$$MSE_K = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[\mathbf{I}(i,j) - \mathbf{K}(i,j)]^2 \tag{54}$$

When approximate circuits are used for image processing, $\mathbf{I}$ can be the resulting image using an exact computation while

$\mathbf{K}$ is the image obtained by approximate computing. For a good agreement with a precisely processed image, the PSNR of a noisy image should be very large - usually larger than a threshold P, i.e.

$$PSNR > P,$$

or

$$\sqrt{MSE_K} < C, \tag{55}$$

where $C = 10^{\log(MAX_I) - \frac{P}{20}}$.

For (54), define an error matrix as $\mathbf{E} = \mathbf{K} - \mathbf{I}$; then the MSE and MED of image $\mathbf{K}$ are

$$MSE_K = E(\mathbf{E} \circ \mathbf{E}), \tag{56}$$

$$MED_K = E(|\mathbf{E}|). \tag{57}$$

In (56) and (57), $E(\mathbf{X})$ denotes the average value of all the elements in matrix $\mathbf{X}$, the "$\circ$" operation obtains the element-wise product of two matrices, which is also known as the *Hadamard product* [19], and $|\mathbf{X}|$ obtains the absolute value of each element in $\mathbf{X}$.

Let d and μ denote the ED and MED of the approximate adder and σ^2 represent the mean squared error distance (MSED):

$$\sigma^2 = MSED = E[d^2]. \tag{58}$$

The MED and MSED of the approximate adders are obtained by the assumption that the inputs are uniformly distributed. In the analysis in this manuscript, the pixel values of an image are assumed to be sufficiently random, i.e. if an image is processed by an approximate addition for each pixel, the mean and mean squared value of the error matrix are considered to be the same as the MED and the MSED of the approximate adder. Hereinafter, μ and σ^2 denote the MED and MSED of the approximate adder, as well as the mean and mean squared values of the error matrix if a resulting image is obtained by one approximate addition of each pixel. If the image $\mathbf{K}$ is processed by a approximate addition operations, the mean squared error, MSE_K, is given by

$$MSE_K = E[(\sum_{i=1}^{a} \mathbf{E_i}) \circ (\sum_{i=1}^{a} \mathbf{E_i})], \tag{59}$$

where $\mathbf{E_i}$ is the error matrix of the ith addition. Assume each $\mathbf{E_i}$ is independent, then (59) becomes:

$$\begin{aligned} MSE_K &= E[\sum_{i=1}^{a} \mathbf{E_i} \circ \mathbf{E_i} + 2 \sum_{1 \leqslant i < j \leqslant a} \mathbf{E_i} \circ \mathbf{E_j}] \\ &\leqslant a\sigma^2 + 2 \sum_{1 \leqslant i < j \leqslant a} E(|\mathbf{E_i}|)E(|\mathbf{E_j}|) \\ &\approx a\sigma^2 + a(a-1)\mu^2. \end{aligned} \tag{60}$$

For the ESA and the ETAII, assume the relationship between $\sqrt{MSE}$ (i.e. σ) and MED (i.e. μ) is given by $\sigma = f(\mu)$ (the function $f(\cdot)$ will be derived next), (60) can be converted to

$$\sqrt{MSE_K} \approx \sqrt{af(\mu)^2 + a(a-1)\mu^2} \leqslant C \tag{61}$$

Based on (53) and (61), the PSNR can be estimated by

$$PSNR \approx 20log(MAX_I / \sqrt{af(\mu)^2 + a(a-1)\mu^2}) \tag{62}$$

The solution of (61) gives the maximum value of MED for deriving the parameter k in the ESA or the ETAII using the corresponding equations (i.e. (26) and (30)) given in section III. Hence, this analytical framework can be used to select the proper approximate adder type and parameter (i.e. carry propagation length k) for image processing applications, instead of building functional models of the approximate adder and running time-consuming simulations with different parameters.

B. Relationship between μ and σ

1) ESA: The relationship between μ and σ is analyzed for the fixed carry case. Similar to the MED calculation in (26), the MSE of the ESA is calculated as:

$$\sigma^2 = \sum_{m=0}^{r-1} (2^{mk+l})^2 (\tfrac{1}{2})^{r-m-1} \times \tfrac{1}{2} \approx 2^{2n-2k-1}. \tag{63}$$

Based on (26) and (63), the relationship between μ and σ for the ESA is given by:

$$\sigma = \sqrt{2}\mu. \tag{64}$$

2) ETAII: Similar to the analysis for the ESA, the MED and MSE for the ETAII are found as $\mu \approx 2^{n-2k-1}, \sigma^2 \approx 2^{2n-3k-1}$. The relationship between μ and σ is:

$$\sigma = (2^{n+1}\mu^3)^{\frac{1}{4}}. \tag{65}$$

The simulated relationship between μ and σ and the analytical functions in (64) and (65) are plotted as in Fig. 10 for $n = 16$. For the ESA, the simulated curve and the analytical results match very closely, as shown by the nearly perfect overlapping curves. For the ETAII, there is a slight discrepancy in the simulated and analytical plots. This discrepancy is due to the approximation when calculating MED and MSE as outlined previously.

C. Estimate of the PSNR for Image Processing

Three image processing algorithms are evaluated next: image sharpening, point detection and arithmetic mean filter.

If $\mathbf{I}$ is the original image and $\mathbf{S}$ is the processed image, the sharpening algorithm [20] is performed as

$$\begin{aligned} &\mathbf{S}(x,y) = 2\mathbf{I}(x,y) - \\ &\tfrac{1}{273} \sum_{i=-2}^{2} \sum_{j=-2}^{2} \mathbf{G}(j+3, j+3)\mathbf{I}(x-i)(y-j), \end{aligned} \tag{66}$$

where $\mathbf{G}$ is a matrix given by:

$$\mathbf{G} = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}. \tag{67}$$

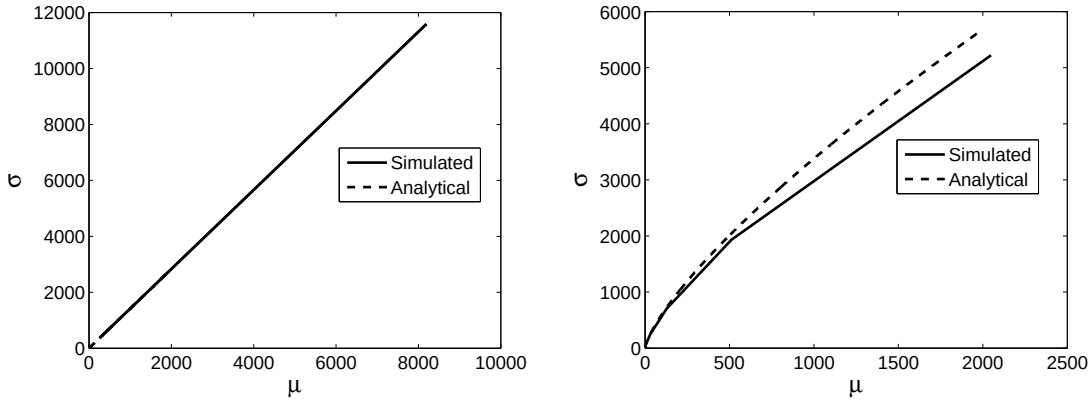


Fig. 10. Simulated and analytical $\sigma - \mu$ relationships for (a) ESA, and (b) ETAII.

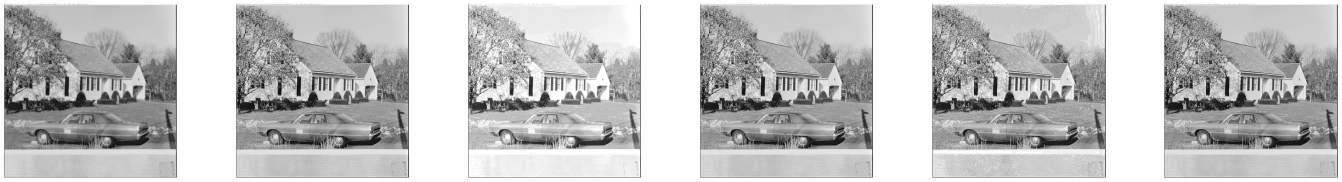


Fig. 11. Image sharpening: (a) original image, (b) using exact adders, (c) using the ESA with k=6 and PSNR=18.1dB, (d) using the ESA with k=10 and PSNR=41.4dB, (e) using the ETAII with k=4 and PSNR=27.4dB and (f) using the ETAII with k=7 and PSNR=56.8dB.

Fig. 12. Point detection: (a) original image, (b) using exact adders, (c) using the ESA with k=7 and PSNR=4.4dB, (d) using the ESA with k=10 and PSNR=24.6dB, (e) using the ETAII with k=4 and PSNR=10.5dB and (f) using the ETAII with k=5 and PSNR=19.9dB.

Fig. 13. Arithmetic mean filter: (a) original image with noise, (b) using exact adders, (c) using the ESA with k=5 and PSNR=10.5dB, (d) using the ESA with k=8 and PSNR=27.2dB, (e) using the ETAII with k=3 and PSNR=16.9dB and (f) using the ETAII with k=5 and PSNR=36.3dB.

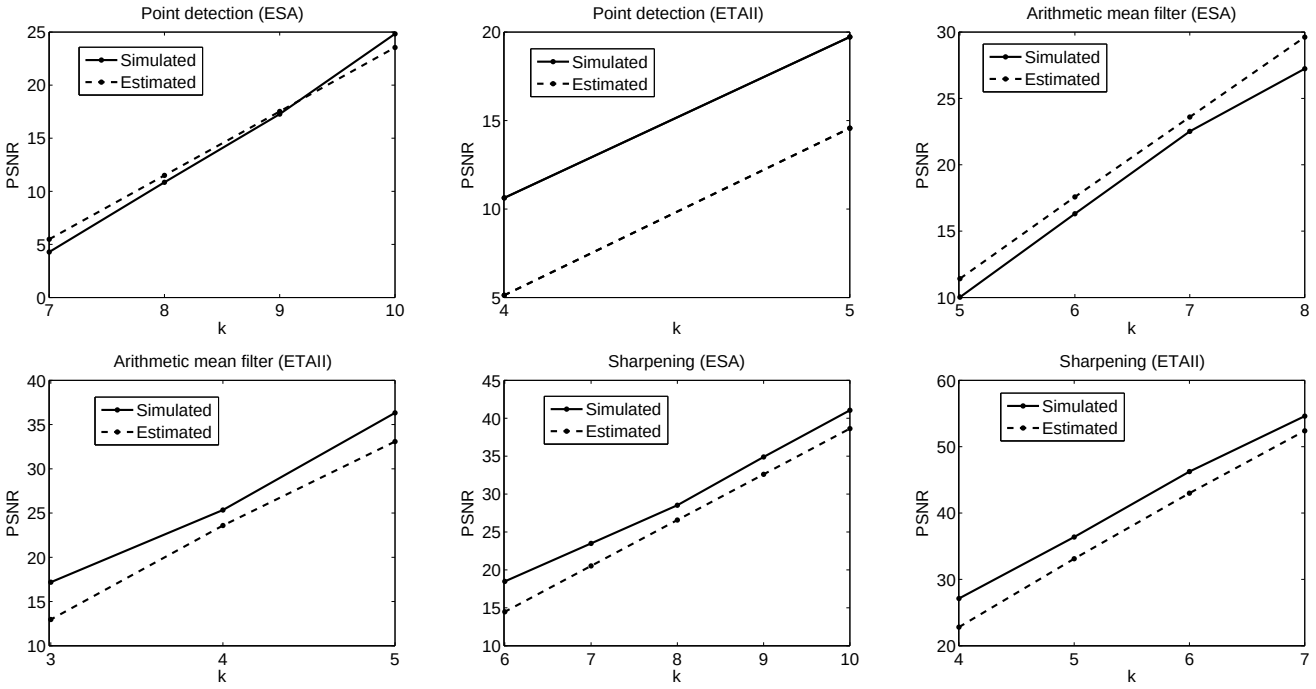


Fig. 14. Simulated and estimated PSNR (dB).

The point detector operation [21] is given by

$$\begin{aligned}\mathbf{S}(x,y) &= 8\mathbf{I}(x,y) - \mathbf{I}(x-1,y-1) - \mathbf{I}(x-1,y)\\ &-\mathbf{I}(x-1,y+1) - \mathbf{I}(x,y-1) - \mathbf{I}(x,y+1)\\ &-\mathbf{I}(x+1,y-1) - \mathbf{I}(x+1,y) - \mathbf{I}(x+1,y+1).\end{aligned} \tag{68}$$

A 3×3 arithmetic mean filter [21] is also implemented as

$$\mathbf{S}(x,y) = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1}\mathbf{I}(x+i,y+j). \tag{69}$$

For all three algorithms, the exact additions are replaced by approximate additions using the ESA and ETAII. All other operations (i.e. multiplication, division and subtraction) are performed accurately. The mean squared error is estimated using (61), in which the number of approximate additions, a, is 25, 8 and 9, respectively for image sharpening, point detection and arithmetic mean filter applications. The PSNR is then given by (53).

In the sharpening algorithm, 16-bit approximate adders are used, because the maximum possible sum is 255×273, i.e. approximately $2^{16} - 1$. 12-bit approximate adders are used for the point detection and arithmetic filer computations. The analytical $\sigma - \mu$ plots shown in Fig. 10 are used for the PSNR.

Six images are selected for the three algorithms and the corresponding PSNR values are obtained; three of them are shown in Figs. 11, 12 and 13. The images are selected such that they are quite "typical" images to be processed, i.e., they show features commonly found in multimedia applications. For the same algorithm and approximate adder, the six images have PSNR values that are very close; this indicates that the PSNR is not strongly correlated to an image, hence the estimated PSNR can be readily applied to them. For further analyzing this feature, define the *relative discrepancy* (RD) as the maximum difference between the PSNR values of the six images divided by their mean PSNR value. The RD values are shown in Table V; the largest RD is 13.2%, however in most cases the RD is below 5%, i.e., at an acceptable level for this type of applications.

TABLE V. RELATIVE DISCREPANCY.

Point detection									
	ESA				ETAII				
k	7	8	9	10	4	5			
RD (%)	13.2	4.8	2.1	1.4	10.5	5.2			
Arithmetic mean filter									
	ESA				ETAII				
k	5	6	7	8	3	4	5		
RD (%)	9.9	5.0	2.7	1.7	5.0	2.0	0.8		
Sharpening									
	ESA					ETAII			
k	6	7	8	9	10	4	5	6	7
RD (%)	5.7	5.0	3.4	1.7	0.9	2.1	1.7	2.6	4.1

The simulated and estimated PSNR values are shown in Fig. 14. The best match between simulated and estimated PSNR values is achieved by the ESA-based point detection application, while the worst occurs for the ETAII-based point detection. In the worst case, the maximum PSNR is about 20dB, that is generally considered to be low. For most cases, the estimate tends to be less accurate when the PSNR values are too small ($< 20dB$). The interesting cases are the ones whose

PSNR values are larger than 20dB, otherwise the approximate results are not acceptable. For these cases, the estimate is shown to be accurate in this PSNR range. Moreover, the difference between estimated and simulated results is within a 3dB margin; so for a given image processing application and approximate adder, the error in the estimate appears to be almost constant. Hence, the simulation results show that with the proposed methodology, the MED is a good indicator for the PSNR. The estimate procedure proposed in this paper can be used to evaluate the performance of approximate adders in image processing applications.

VI. Conclusion

In this paper, an analytical framework has been proposed for characterizing approximate adder designs. This framework consists of models for the evaluation of three different types of approximate adders targeting several error metrics. Time-consuming simulation can then be avoided by using the proposed analytical models. Design criteria with respect to error characteristics in the operations of these approximate adders have been provided based on the analysis. As an example of the application of the framework, the PSNR in image processing has been evaluated using the proposed framework. The estimated PSNR can then be utilized for selecting the proper scheme of an approximate adder. Extensive simulation results show that there is a good agreement between the analytical outcomes of the proposed framework and the simulation results for three different computational algorithms commonly used in image processing. The PSNR estimate method proposed in this paper shows that there is a close relationship between MED and PSNR, while ER is less important. This may provide insights in the design of approximate arithmetic circuits for error-tolerant applications.

References

[1] J. Han and M. Orshansky, "Approximate Computing: An Emerging Paradigm For Energy-Efficient Design," in *ETS'13, Proc. of the 18th IEEE European Test Symposium*, Avignon, France, May 2013.

[2] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "Macaco: Modeling and analysis of circuits for approximate computing," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2010, pp. 667–673.

[3] R. Hegde and N. Shanbhag, "Soft digital signal processing," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 6, pp. 813–823, Dec 2001.

[4] Y. Liu, T. Zhang, and K. Parhi, "Computation error analysis in digital signal processing systems with overscaled supply voltage," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 517–526, April 2010.

[5] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," *2011 Design, Automation & Test in Europe*, pp. 1–6, Mar. 2011.

[6] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 1, pp. 124–137, 2013.

[7] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate xor/xnor-based adders for inexact computing," in *IEEE International Conference on Nanotechnology*. Beijing, China: IEEE, 2013.

[8] S.-L. Lu, "Speeding up processing with approximation circuits," *Computer*, vol. 37, no. 3, pp. 67–73, 2004.

[9] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proceedings of the conference on Design, automation and test in Europe*. ACM, 2008, pp. 1250–1255.

[10] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, 2012, pp. 1257–1262.

[11] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 820–825.

[12] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Integrated Circuits, ISIC'09. Proceedings of the 2009 12th International Symposium on*. IEEE, 2009, pp. 69–72.

[13] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *Computers, IEEE Transactions on*, vol. 62, no. 9, pp. 1760–1771, 2013.

[14] J. Huang, J. Lach, and G. Robins, "A methodology for energy-quality tradeoff using imprecise hardware," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 504–509.

[15] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in *Proceedings of the International Conference on Computer-Aided Design*. ACM, 2012, pp. 728–735.

[16] M. Breuer, "Intelligible test techniques to support error-tolerance," in *Test Symposium, 2004. 13th Asian*, 2004, pp. 386–393.

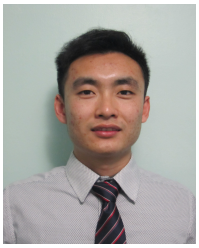
[17] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.

[18] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*. IEEE, Aug. 2010, pp. 2366–2369.

[19] R. A. Horn, "The hadamard product," in *Proc. Symp. Appl. Math*, vol. 40, 1990, pp. 87–169.

[20] M. S. Lau, K.-V. Ling, and Y.-C. Chu, "Energy-aware probabilistic multiplier: design and analysis," in *Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*. ACM, 2009, pp. 281–290.

[21] H. R. Myler and A. R. Weeks, *The pocket handbook of image processing algorithms in C*. PTR Prentice Hall, 1993.



Cong Liu received the B.S. degree in automation from Tsinghua University, Beijing, China, in 2012. Since September 2012, he has been a graduate student in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. His current research interest is approximate computing.

Jie Han (S'02-M'05) received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China, in 1999 and the Ph.D. degree from Delft University of Technology, The Netherlands, in 2004. He is currently an assistant professor in the Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, Canada. His research interests include reliability and fault tolerance, nanoelectronic circuits and systems, and novel computational models for nanoscale and biological applications. Dr. Han was nominated for the 2006 Christiaan Huygens Prize of Science by the Royal Dutch Academy of Science (Koninklijke Nederlandse Akademie van Wetenschappen (KNAW) Christiaan Huygens Wetenschapsprijs). His work was recognized by the 125th anniversary issue of Science, for developing theory of fault-tolerant nanocircuits. He served as a Technical Program Chair and General Chair in IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012 and 2013, respectively. He also served as a Technical Program Committee Member in several other international symposia and conferences.

Fabrizio Lombardi (M'81-SM'02-F'09) graduated in 1977 from the University of Essex (UK) with a B.Sc. (Hons.) in Electronic Engineering. In 1977 he joined the Microwave Research Unit at University College London, where he received the Master in Microwaves and Modern Optics (1978), the Diploma in Microwave Engineering (1978) and the Ph.D. from the University of London (1982). He is currently the holder of the International Test Conference (ITC) Endowed Chair Professorship at Northeastern University, Boston. His research interests are bio-inspired and nano manufacturing/computing, VLSI design, testing, and fault/defect tolerance of digital systems. He has extensively published in these areas and coauthored/edited seven books.