

Probabilistic Error Modeling for Approximate Adders

Sana Mazahir, Osman Hasan, Rehan Hafiz, Muhammad Shafique, and Jörg Henkel

Abstract—Approximate adders are widely being advocated as a means to achieve performance gain in error resilient applications. In this paper, a generic methodology for analytical modeling of probability of occurrence of error and the Probability Mass Function (PMF) of error value in a selected class of approximate adders is presented, which can serve as performance metrics for the comparative analysis of various adders and their configurations. The proposed model is applicable to approximate adders that comprise of sub-adder units of uniform as well as non-uniform lengths. Using a systematic methodology, we derive closed form expressions for the probability of error for a number of state-of-the-art high-performance approximate adders. The probabilistic analysis is carried out for arbitrary input distributions. It can be used to study the dependence of error statistics in an adder's output on its configuration and input distribution. Moreover, it is shown that by building upon the proposed error model, we can estimate the probability of error in circuits with multiple approximate adders. We also demonstrate that, using the proposed analysis, the comparative performance of different approximate adders can be correctly predicted in practical applications of image processing.

Index Terms—Approximate computing, Adders, Probability of error, Probability Mass Function (PMF), Image smoothing, Modeling, Arithmetic, Analysis

1 INTRODUCTION

As the computing systems, involving high complexity arithmetic, become increasingly embedded and mobile, the concern for energy efficiency, size and speed of these systems also accrues. A large number of such applications involve media processing, such as image, video and audio based applications designed for human interface. Other such computationally intensive applications include data mining and machine learning. A common feature in these applications is that they do not require the outcome to be fully precise, rather an approximate result is adequately acceptable. Approximate computing [1], [2] is an emerging trend in hardware and software design that exploits this inherent tolerance for inaccuracy for efficiency gain in terms of required hardware, speed and/or power.

Several functionally approximate designs for basic arithmetic blocks including adders [3], [4], [5], [6], [7], [8], [9], [10] and multipliers [11], [12], [13] have been proposed. The design of fast and efficient adders has attracted great attention in the approximate computing domain. Unlike the low-power approximate adders, most of these high-speed, low-latency approximate adders exploit the fact that the carry propagation chain for most input combinations is shorter than that for the worst case. Some prominent examples of these high-performance adders are: error tolerant adders (ETAs) [4], almost correct adder (ACA-I) [14], variable latency speculative adder (VLSA) [14], accuracy configurable adder (ACA-II) [3], gracefully-degrading adder (GDA) [5] and generic accuracy configurable adder (GeAr) [15]. Another type of approximate adders are low-power adders, presented in [16], [17], [18], [19].

In order to select an appropriate approximate adder for a given application, comparative performance of the available

designs has to be taken into consideration. The designs mentioned above have been evaluated and compared in terms of critical path delay, required hardware and power resources and error statistics. Traditionally, the error performance evaluation and comparison presented for these adders relies on computer simulations, which can only be executed exhaustively for small-sized adders. As the adders size grows, time and memory resources required for the exhaustive simulations render them infeasible or impossible. For deployment in any application, systematic quantification of the computational inaccuracy in these designs is indispensable. Unlike the existing statistical performance metrics that rely on Monte-Carlo simulations, the proposed analytical model for probability of error provides an accurate measure for accuracy comparison of a wide variety of approximate adders.

1.1 Motivational Studies

To understand the importance of mathematical analysis of approximation error, we present some studies to understand the limitations of computer simulations.

1.1.1 Limitations of Monte-Carlo Simulations

Monte-Carlo simulations are often performed to evaluate the suitability of a particular adder configuration. In comparison with mathematical analysis, they have following limitations:

- The simulation results do not provide any information about the causes of error or the conditions on inputs that lead to error.
- It is difficult to observe the effect of changes in adder configuration as every adder needs to be simulated separately, which requires considerable programming effort.
- They only provide an estimate about the long-term error statistics. Sometimes it may be required to configure the adder's accuracy according to application requirement. In such cases, it is important to know the conditions that lead to error. In [20], we demonstrated that knowing the mathematical model of error value leads to an enhanced design of accuracy configurable adders.

The mathematical analysis can provide a systematic understanding of the relationship between error statistics and the adder structure, which enables more intelligent and goal-oriented design.

• S. Mazahir and O. Hasan are with the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan.

E-mail: {sana.mazahir, osman.hasan}@seecs.nust.edu.pk

• R. Hafiz is with the Electrical Engineering Department, Information Technology University (ITU), Lahore, Pakistan.

Email: rehan.hafiz@itu.edu.pk

• M. Shafique is with Vienna University of Technology (TU Wien), Austria. Before, he was with Karlsruhe Institute of Technology (KIT), Germany.

E-mail:muhammad.shafique@tuwien.ac.at, muhammad.shafique@kit.edu

• J. Henkel is with Karlsruhe Institute of Technology (KIT), Germany.

E-mail:henkel@kit.edu

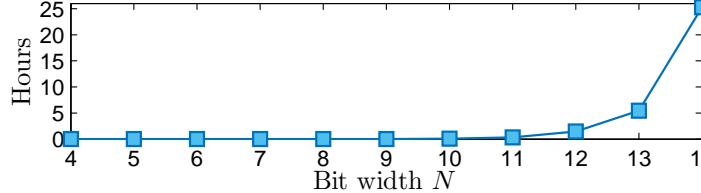


Fig. 1. Exhaustive simulation time for GeAr adders.

1.1.2 Limitations of Exhaustive Simulations

In order to compute the exact probability of error, we need to simulate the adder for all possible input combinations. The number of possible input combinations for an N -bit adder is 2^{2N} . Fig. 1 shows simulation time required to run these simulations for different adder lengths. The simulations were run on an Intel Core i7, 2.4 GHz processor. It can be seen that the exhaustive simulations become increasingly time-consuming with the increase in adder length. These simulations provide accurate comparison among different adder configurations. However, the required time makes them impractical as every single configuration will need to be simulated individually.

1.2 Novel Contributions

In order to have a scalable and accurate error performance evaluation of approximate adders without the need of simulations, in this paper, we propose a generic methodology for analytical modeling of probability of error and Probability Mass Function (PMF) of error value in the output of approximate adders. In the proposed methodology, the conditions on input that lead to error in the output are identified. These conditions are then systematically transformed into simpler, independent events and the probabilities of these events are analyzed in a general form for arbitrary input distributions. The probabilistic analysis is carried out using basic theorems of probability theory. The class of adders selected for the analysis include high-speed, low-latency approximate adders. These adders involve carry chain truncation and carry prediction between successive precise sub-adder units. A general model for this class of adders is presented for analysis that can be configured to any of the adders presented in [3], [4], [5], [14], [15] and the probability of error in their approximate sums can be computed using the proposed methodology. The accuracy achieved using the proposed approach has been validated by ensuring that the probability of error and PMF of error are in concordance with simulation results.

The main advantages of the proposed approach are:

- It facilitates comparison among arbitrary bit width approximate adders without the need of time-consuming simulations, which are always required to evaluate performance metrics used in existing works.
- Using the models developed by the proposed methodology, the error statistics can be expressed as functions of circuit parameters, like number of input bits, carry-chain length, number of overlapping prediction bits and number of sub-adders. Hence they can provide insights into the relationship between circuit specifications and error statistics, which can help the circuit designers choose circuit parameters according to application requirements.

Since in most practical applications, circuits with multiple adders are used to perform more complex arithmetic operations, we also present an analysis for cascade of adders. Once the probability of error in an approximate adder is computed using the above-discussed methodology, it is used to compute error probability in circuits using multiple, independent approximate adders. The results are validated through Monte-Carlo simulations and for image processing applications.

1.3 Paper Organization

The remainder of this article has been organized as follows: In Section 2, previous work on evaluation of accuracy of approximate adders is reviewed. Section 3 presents the adder model and explains the representation used throughout the paper. The methodology for probabilistic analysis is presented in Section 4. In Section 5, the usefulness of the proposed methodology is illustrated by applying it to the ACA-II [3], GeAr [15] and ETA-II [4] adders and our findings are compared with simulation results. Finally, Section 6 concludes the paper.

2 RELATED WORK

Approximate circuits are designed to compromise accuracy for other performance gains including reduced hardware cost, faster speed and lower power consumption. Performance metrics have been defined to qualify these circuits from all the above-mentioned perspectives. Performance metrics, such as minimum acceptable accuracy (MAA) [4], [15], accuracy of amplitude (ACC_{amp}) [3], [15], accuracy of information (ACC_{inf}) [3], [15], mean error distance (MED) [1], [15], [21], [22], [23], normalized error distance (NED) [1], [15], [21], error rate (ER) [22], [23], error significance (ES) [24] and maximum error magnitude [8], are used to quantify the computational accuracy of approximate circuits. *All of these metrics are estimations made on the basis of Monte-Carlo simulations.* Exhaustive simulations can only be performed for small sized circuits. As the size of circuits increases, such simulations lose their dependability for robust performance prediction. Furthermore, since different metrics are used for the evaluation of various adders found in existing works, fair comparison is not possible without simulating each of the configurations under consideration, which requires considerable programming effort..

In [25], error analysis is performed for arithmetic circuits with voltage over-scaling. The approach used is specific for arithmetic architectures for voltage over-scaled signal processing applications. In [26], a systematic methodology for Modeling and Analysis of Circuits for Approximate Computing (MACACO) has been proposed. Monte-Carlo simulations are used to compute metrics including error probability and error distribution. The methodology can be used to compare specific configurations of any approximate circuit but the error statistics and distributions obtained are numeric data that cannot be parametrized in terms of circuit specifications. Consequently, they cannot yield insights into its behavior dependencies and causalities.

In [3], the probability of error in the ACA-II adder has been approximated. However, in this model, circuit's components and intermediary outputs have been assumed to be independent in order to simplify the analysis. The model does provide some understanding of error dependence on adder configuration but the computed frequency of error is only an approximation, which may deviate significantly from its exact value as the effects of interdependencies in the adder components become more pronounced. Additionally, it cannot be used to find the PMF of error. In our previous work [15], an analytical model for the GeAr adder is presented, which only computes the probability of error. The GeAr model cannot be used for more general structures like some configurations of GDA and modified ETA-II. Therefore, a more generic methodology is required to understand the comparison among various configurations of this broader class of approximate adders.

In [22], error characteristics of the Error Segmentation Adder (ESA), Almost Correct Adder (ACA) and Error Tolerant Adder Type II (ETA-II) are approximated. This work considers the three adders separately but is not generalizable for more complex adder configurations. Furthermore, the approximation method is complicated by considering individual bits in

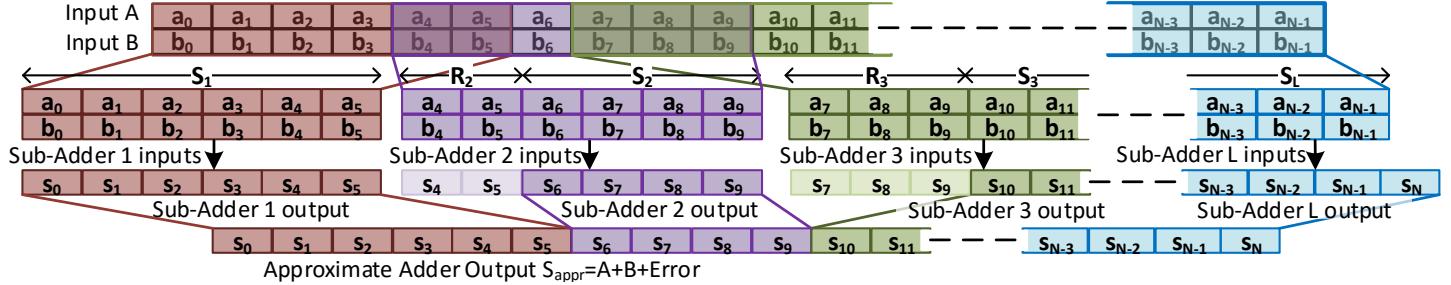


Fig. 2. Generic functional model for approximate adders. N -bit inputs are added using L sub-adders. By setting the values of S_1, S_2, \dots, S_L and R_2, \dots, R_L , the adder can be configured as ACA, ACA-II, GeAr, ETA-II, ETA-IIM, GDA etc.

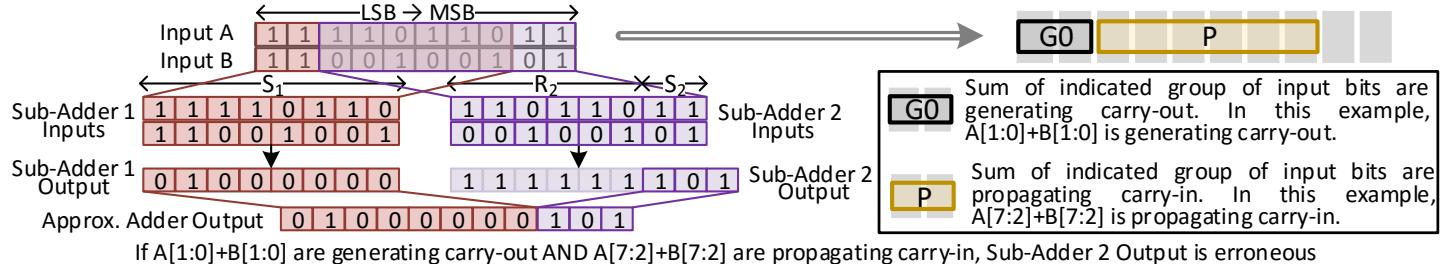


Fig. 3. An example of addition using an approximate adder with two sub-adders. Error in the approximate adder is due to error in Sub-Adder 2. This occurs when all the prediction bit pairs of Sub-Adder 2 are propagating carry-in (the pairs of bits at positions indicated by R_2 are unequal) and the less significant bit pairs, that are not input to Sub-Adder 2, are generating carry-out. Therefore, the event E_2 (error in Sub-Adder 2) is represented as simultaneous carry-out generation (G_0) and carry-in propagation (P) events at the relevant bit locations.

case of ACA, which is not necessary as we show in this paper that only the location of carry chain truncation is required to predict the error value. In case of error value analysis of ESA and ETA-II, only the error in the most significant sub-adder unit is considered at a time, while ignoring the less significant ones, which may not be always negligible.

It is important to note that the mathematical analysis is not only a tool for comparison of different adders, but it also helps understand the adders' behavior, which in turn helps in designing more enhanced adder structures. In our work in [20], we demonstrated that the knowledge of the mathematical model of approximation error leads to more efficient design of accuracy configurable adders.

3 ADDER MODEL

Most of the approximate adders, presented in the literature, are designed by exploiting the fact that the occurrence of long carry propagations is very rare. This implies that errors induced by breaking the carry chain are very infrequent. In many designs, carry chain is truncated by dividing the input bits among multiple disjoint or overlapping sub-adders, so that the longest carry propagation chain is now equal to the size of these sub-adders.

Fig. 2 shows a general model of an N -bit approximate adder consisting of L precise sub-adders. The i^{th} sub-adder takes $S_i + R_i$ bits as input. The number of bits used to predict carry-in is R_i . The number of sum bits is S_i . The output of first sub-adder is always precise. Hence, S_1 is equal to the length of first sub-adder and $R_1 = 0$.

Our adder model can be configured to many approximate adder designs, such as ACA-II [3], GeAr [15], GDA [5] and ETA-II [4], by specifying the values of L, R_2, R_3, \dots, R_L and S_1, S_2, \dots, S_L . R and S can remain constant in all sub-adders, like ACA-II [3], GeAr [15] and ETA-II [4] or they can change from sub-adder to sub-adder as in the modified ETA-II in [4] and GDA [5].

Error in the i^{th} sub-adder's output occurs when its R_i prediction bits fail to correctly predict the carry-in for the higher S_i sum bits. The error in any of the L sub-adders

can lead to error in final output. This introduces error at the respective bit positions of S_{appr} and thus the corresponding probability of error can be defined as the expected relative frequency of incorrect adder outputs:

$$\Pr[\text{Error}] = \Pr[S_{appr} \neq A + B] \quad (1)$$

where A and B are N -bit adder inputs, having probability distributions $p_A(\cdot)$ and $p_B(\cdot)$. An example is shown in Fig. 3. The output of Sub-Adder 2 is erroneous because there is a carry-out generated by the sum $A[1 : 0] + B[1 : 0]$, which would have been propagated to $A[9 : 8] + B[9 : 8]$ (as the sum $A[7 : 2] + B[7 : 2]$ is propagating carry-in i.e., $A[i] \oplus B[i] = 1$ for $i = 2, \dots, 7$), if the carry-chain had not been broken. In Fig. 3, "G0" represents the event that input combination is such that $A[1 : 0] + B[1 : 0]$ generates a carry-out. Similarly, "P" represents the event that $A[7 : 2] + B[7 : 2]$ propagates carry-in. Error in Sub-Adder 2 occurs when both of these events occur simultaneously. The representation of events in Fig. 3 will be used throughout this article. The rectangle labeled as G0 represents that there is a carry-out generated from the sum of inputs at the indicated bit locations and P represents that the sums of bits at the indicated bit locations are all propagating carry-in.

Since the occurrence of error is due to the broken carry chain, a deficiency is created in the sum by an amount equal to the weight of that bit position due to the missed addition of carry bit. Thus, we found that in case of an error, the adder output value is always going to be less than the precise sum. In case of simultaneous error in more than one sub-adder, the error value is the sum of all the carries that are discarded because of the broken carry chain. Due to this type of behavior, the error can have certain specific values. As a result, the metrics such as ACC_{inf} , ACC_{amp} , MSE , MED and NED may not be adequately informative since they average different forms of magnitude of error (absolute error or hamming distance) over all inputs, that mostly yield the correct output. For the example given in Fig. 3, the value of error is 256. However, since the output will be correct for most of the inputs, averaging the error over all inputs will yield a small error value. Therefore, for this type of adders, it is more pertinent to use probability of

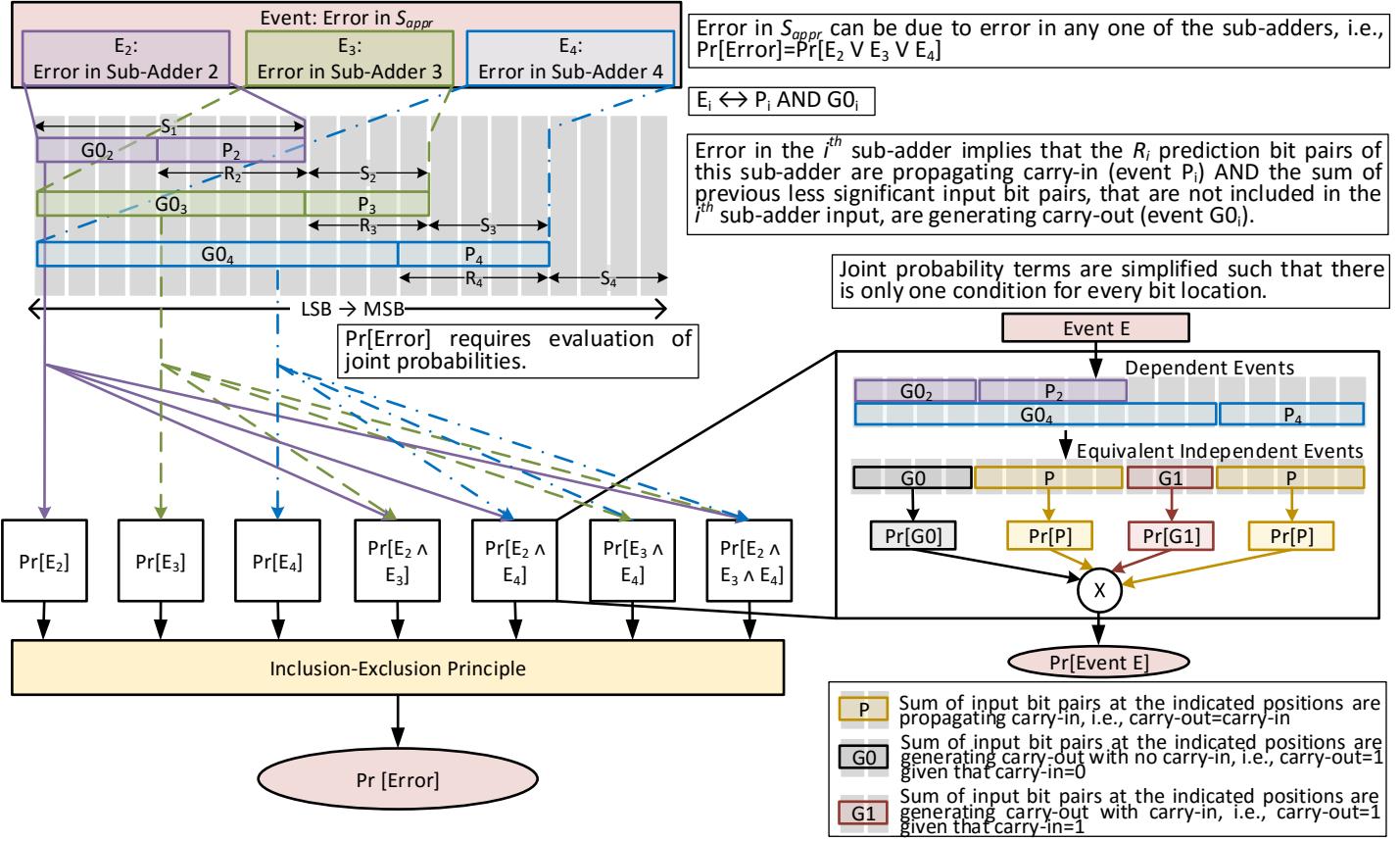


Fig. 4. Methodology for the probability of error analysis.

error and PMF of error value as performance metrics. Once the PMF of error is found, other metrics like *MED*, *MSE*, *NED* etc. can also be calculated, if required.

4 METHODOLOGY FOR ANALYTICAL MODELING

In this section, we develop a generic methodology for error probability analysis of the class of adders represented by the model in Fig. 2. The proposed methodology, depicted in Fig. 4, primarily has the following components:

- 1) The first step is to identify the errors in the intermediary logical elements of the approximate adder that contribute to error in the output and then mathematically relate the occurrence of such errors with $\Pr[\text{Error}]$, defined in Eq. (1). In this step, $\Pr[\text{Error}]$ will be expressed as the sum of probabilities of joint carry-in propagation and carry-out generation events for specified groups of input bits. This is explained in Sub-section 4.1.
- 2) The next step is to find each joint probability term. For this, we propose a method to transform these joint probabilities into products of probabilities of independent carry-in propagation and carry-out generation events. This method is given in Sub-section 4.2.
- 3) The probabilities of carry-in propagation and carry-out generation events are derived in generalized form for arbitrary input distributions, which will be used to compute each product term. The derivations are given in Sub-section 4.3.
- 4) By building upon the analysis, the analysis for the PMF of error value is presented in Sub-section 4.4.
- 5) In Sub-section 4.5, the proposed analysis is used to estimate the error statistics in circuits with multiple approximate adders.

4.1 Identifying Events leading to Approximation Error

The N -bit approximate adder, given in Fig. 2, is constructed using L sub-adders. The i^{th} sub-adder, for $i = 1, 2, \dots, L$, is

a $R_i + S_i$ -bit precise adder. The output of Sub-adder 1 is always correct as there is no loss of accuracy due to carry-chain truncation. However, the outputs of Sub-adder 2 to Sub-adder L can be erroneous since addition with the carry generated by the previous less significant bits has been eliminated. Since the output sum is obtained by gathering the outputs of all the sub-adders, as shown in Fig. 2, error in any sub-adder's output can contribute to error in the final output.

Error in the i^{th} sub-adder (for $i \geq 2$) occurs when the R_i prediction bits of the sub-adder's input are all propagating carry-in and the previous less significant bits of adder's input, that are not input to this sub-adder, are generating carry-out. The error occurs because the generated carry-out is not propagated due to the broken carry chain between sub-adders. Let E_2, E_3, \dots, E_L represent events associated with the occurrence of error in Sub-adders 2, 3, ..., L respectively. The error events are defined such that $E_i = 1$ if the i^{th} sub-adder output is erroneous and $E_i = 0$ otherwise. Since any one of these events can contribute to error in the final output S_{appr} , the event associated with an error in S_{appr} is expressed as the union of these events. Furthermore, since two or more of these events can occur simultaneously, these events are not mutually exclusive. The probability of the union of such events can be evaluated using the inclusion-exclusion principle [27],

$$\begin{aligned} \Pr[\text{Error}] &= \Pr [E_2 \vee E_3 \vee \dots \vee E_L] = \sum_{i=2}^L \Pr [E_i] \\ &\quad - \sum_{2 \leq i < j \leq L} \Pr [E_i \wedge E_j] + \sum_{2 \leq i < j < k \leq L} \Pr [E_i \wedge E_j \wedge E_k] - \dots + (-1)^L \Pr \left[\bigwedge_{i=2}^L E_i \right] \end{aligned} \quad (2)$$

Each term in Eq. (2) is going to be evaluated separately in the next sub-sections. For larger number of sub-adders, as the analysis becomes increasingly difficult due to large number of

Algorithm 1 Probability of Error Evaluation

```

1: Input  $L, S_1, S_2, \dots, S_L$  and  $R_2, R_3, \dots, R_L$ .
2: Find the power set  $P_S = \mathbb{P}(S)$ , where  $S = \{2, 3, \dots, L\}$ .
3: Initialize  $\text{Pr}_{err} = 0$ .
4: for each  $i \in \{1, 2, \dots, L - 1\}$  do
5:   for every  $j^{th}$  ( $1 \leq j \leq \binom{L-1}{i}$ ),  $i$ -sized element  $P_j^{(i)} \in P_S$  do
6:     Run Algorithm 2 with inputs  $P_j^{(i)}, S_1, S_2, \dots, S_L$  and
       $R_2, R_3, \dots, R_L$ . to find the probability of intersection  $\text{Pr}_{int} =$ 
       $\bigwedge_{k \in P_j^{(i)}} E_k$ .
7:     Update  $\text{Pr}_{err} = \text{Pr}_{err} + (-1)^{i+1} \text{Pr}_{int}$ .
8:   end for
9: end for
10: Output  $\text{Pr}_{err}$ .

```

joint probability terms, Algorithm 1 is presented so that the inclusion-exclusion principle can be automated, if desired.

4.2 Evaluation of Joint Probabilities

4.2.1 Probability of Error in Individual Sub-Adders

Every event E_2, E_3, \dots, E_L in Eq. (2) is due to the *co-occurrence* of a *carry-out generation* event and a *carry-in propagation* event, as explained in Section 3. For the i^{th} sub-adder ($i \geq 2$), the carry-in propagation event is defined for its R_i prediction bits and the carry-out generation event is defined for the previous less significant bits that are not included in the input of this sub-adder. Let $G0_i$ and P_i represent the carry-out generation (with carry-in equal to zero) and carry-in propagation events that lead to error in the output of this i^{th} sub-adder, respectively. Hence, probability of error in this i^{th} sub-adder has been formulated as follows:

$$\text{Pr}[E_i] = \text{Pr}[G0_i \wedge P_i] \quad (3)$$

Since for Sub-adder i ($i \geq 2$), P_i and $G0_i$ represent conditions on disjoint groups of bits of the adder's inputs, they can be modeled as independent events. Thus, the probability of intersection of independent events is equal to product of probabilities of individual events.

$$\text{Pr}[E_i] = \text{Pr}[G0_i] \text{Pr}[P_i] \quad (4)$$

It should be noted here that the input bits are perfectly independent if they are uniformly distributed and the probabilities of events P_i and $G0_i$ can be parametrized in terms of number of bits being added. In case of non-uniformly distributed inputs, these probabilities depend on both the number and position of bits and the events $G0_i$ and P_i may not be independent. This will be explained in Sub-section 4.3. In order to keep the analysis simple, we shall use the formulation in Eq. (4) and hence treat the corresponding groups of bits as independent.

Now each of the term in the first sum, i.e., $\sum_{i=2}^L \text{Pr}[E_i]$, in Eq. (2) has been represented using Eq. (4). The probabilities $\text{Pr}[G0_i]$ and $\text{Pr}[P_i]$ are going to be derived in general form in Sub-section 4.3 and results will be substituted back in Eq. (2).

4.2.2 Probability of Error in Multiple Sub-Adders

When two or more of the events E_2, E_3, \dots, E_L occur simultaneously, carry-in propagation and carry-out generation events for the respective events, as defined in Sub-section 4.1, also co-occur. All the possible co-occurrences of the error events are represented by the joint probability terms in Eq. (2). Each intersection term in Eq. (2) is expressed as the *intersection* of carry-out generation and carry-in propagation events defined for *each of the operands of intersection* according to Eq. (3). For instance, $\text{Pr}[E_i \wedge E_j \wedge E_k]$ represents probability of co-occurrence of error in Sub-adders i, j and k . Since, according to Eq. (3), each of events E_i, E_j and E_k is an intersection of

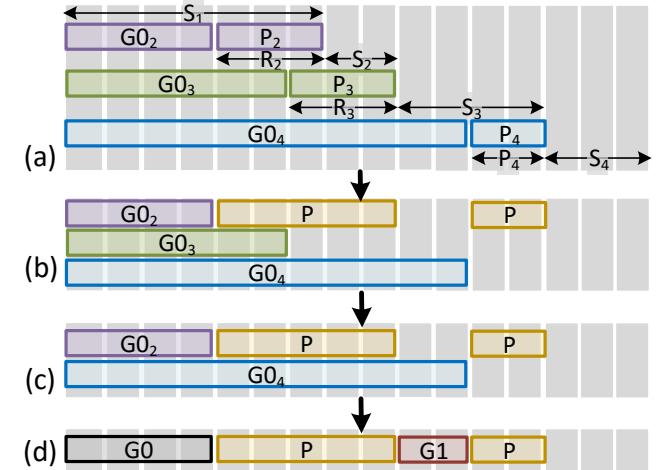


Fig. 5. Transformation of dependent events into equivalent independent events

carry-in propagation and carry-out generation events, the joint probability can be expressed as follows.

$$\begin{aligned} \text{Pr}[E_i \wedge E_j \wedge E_k] \\ = \text{Pr}[(G0_i \wedge P_i) \wedge (G0_j \wedge P_j) \wedge (G0_k \wedge P_k)] \end{aligned} \quad (5)$$

Now $G0_i, P_i, G0_j, P_j, G0_k$ and P_k are not all mutually independent when all of them are simultaneously involved in intersection operation in Eq. (5). For instance, $G0_j$ and P_i may be defined for overlapping groups of bits for a particular configuration of adder in Fig. 2. Due to the overlap, multiple conditions are found to be imposed on the same bits of adder inputs. This means that the events in the intersection operation in Eq. (5) are not mutually independent.

We found that the overlap in the events of carry-in propagation and carry-out generation is such that these events can be redefined or transformed into such equivalent events that are mutually independent. The equivalent events can be either carry-in propagation (P), carry-out generation with no carry-in ($G0$) or carry-out generation with carry-in equal to one ($G1$). In the proposed methodology, we have formulated a method for the transformation of dependent or overlapping events such that every input bit contributes to only one event. The concept can be explained by considering the example in Fig. 5. Consider the event $(E_2 \wedge E_3 \wedge E_4) = (P_2 \wedge G0_2 \wedge P_3 \wedge G0_3 \wedge P_4 \wedge G0_4)$ in Fig. 5(a). The groups of generating and propagating sums are indicated. It can be seen that same bits are contributing to multiple events. The events are transformed as follows:

- 1) The three propagation events can be replaced by one propagation event, such that the new event imposes the propagation condition on all the bits involved in individual events. This is shown in Fig. 5(b). Hence, the number of events in the intersection operation are reduced, i.e., $(E_2 \wedge E_3 \wedge E_4) = (P \wedge G0_2 \wedge G0_3 \wedge G0_4)$.
- 2) Now consider $G0_2$ and $G0_3$. Since there is a carry at the 5^{th} bit position (due to $G0_2$) and next bits are propagating, then there will be a carry-out at the 7^{th} bit position, which means that $G0_3$ is implied by $(G0_2 \wedge P)$. Therefore, $G0_3$ can be eliminated, as shown in Fig. 5 (c). The joint event becomes $(E_2 \wedge E_3 \wedge E_4) = (P \wedge G0_2 \wedge G0_4)$.
- 3) Now, $G0_4$ implies a third carry at 12^{th} bit position. However, $P \wedge G0_2$ implies that there is a carry-out at the 10^{th} bit location, which will also contribute to the sum at this bit location. Hence, $G0_4$ can be redefined as an event $G1$, i.e., a carry-out generation event at the 12^{th} bit location with a carry-in at 10^{th} bit location. The joint event becomes $(E_2 \wedge E_3 \wedge E_4) = (P \wedge G0 \wedge G1)$, where $G0 = G0_2$. Now,

since every bit contributes to a single event, the probability can be evaluated as follows:

$$\begin{aligned} \Pr[E_2 \wedge E_3 \wedge E_4] &= \Pr[P \wedge G0 \wedge G1] \\ &= \Pr[P] \Pr[G0] \Pr[G1] \end{aligned} \quad (6)$$

Algorithm 2 computes the probability of intersection events employing the above discussed concept. First, all the propagation bits are collected and eliminated from carry-generation event sets (steps 2-6). Then the carry-generation events are iteratively simplified into $G0$ and $G1$ events (steps 7-17). The probabilities $\Pr[P]$, $\Pr[G0]$ and $\Pr[G1]$ used in Algorithm 2, for specific groups of input bits, are derived in Sub-section 4.3. Algorithm 2 also computes the error value V , which will be used to find the PMF in Sub-section 4.4 (steps 14, 18-23).

Algorithm 2 Joint Probability $\Pr[\wedge_{i \in I} E_i]$

```

1: Input the set of sub-adder indices  $I$  in intersection operation,
    $S_1, S_2, \dots, S_L$  and  $R_2, R_3, \dots, R_L$ .
2: For  $i^{th}$  sub-adder ( $2 \leq i \leq L$ ), find the set of prediction bits:  $r_i = \{x : \sum_{k=1}^{i-1} S_k - R_i \leq x \leq \sum_{k=1}^{i-1} S_k - 1\}$ 
3: Find union of all the sets of prediction bit positions  $R_U = \bigcup_{i \in I} r_i$ .
4: Set  $\Pr_{int} = \Pr[P; R_U]$ .
5: Find the sets for carry-out generating bits  $g_i = \{0, 1, 2, \dots, \sum_{k=1}^{i-1} S_k\} - r_i$ , for all  $i \in I$ .
6: Update every  $g_i = g_i - R_U$ .
7: Find intersection  $G_I = \bigcap_{i \in I} g_i$ .
8: Initialize the set of generation events  $S_G = \{G_I\}$ .
9: Update  $\Pr_{int} = \Pr_{int} \times \Pr[G0; G_I]$ .
10: Find difference set  $G_D = \{g_i - G_I : i \in I \wedge g_i - G_I \neq \emptyset\}$ .
11: for every element of  $G_D$  do
12:   Update  $G_I = \bigcap_{g_d \in G_D} g_d$ .
13:   Update  $\Pr_{int} = \Pr_{int} \times \Pr[G1; G_I]$ .
14:   Add  $G_I$  to the set  $S_G$ .
15:   Update  $G_D = \{g_d - G_I : g_d \in G_D \wedge g_d - G_I \neq \emptyset\}$ 
16: end for
17: Output  $\Pr_{int}$ .
18: For error value, find  $s_i = \sum_{k=1}^i S_k$ .
19: Initialize error value  $V = 0$ .
20: for every element  $g$  of  $S_G$  do
21:   Update  $V = V + 2^s$ , where  $s$  is found as follows: for  $d_i = s_i - \max(g)$ ,  $s = s_i$  for which  $d_i$  has the smallest positive value.
22: end for
23: Output error value  $V$ .
```

4.3 Probabilities of Carry-in Propagation and Carry-out Generation Events

We found in Sub-section 4.2 that all the terms in Eq. (2) can be represented as product of probabilities of carry-in propagation and carry-out generation by addition of specific groups of bits of N -bit inputs A and B . In this section, we first derive the PMFs of groups of bits of inputs A and B for given p_A and p_B , which are then used to derive the probabilities of events P , $G0$ and $G1$. Analysis for the special case of uniformly distributed inputs is also presented to demonstrate the application of the proposed method.

Let $X = A[k_2 : k_1]$ and $Y = B[k_2 : k_1]$, for $0 \leq k_1 \leq k_2 \leq N - 1$, be the n -bit ($n = k_2 - k_1 + 1$) groups of bits of A and B , respectively. In order to find the probability of carry-in propagation and carry-out generation by the addition of X and Y , we first need the probability distributions of X and Y . If $p_A(k)$ is the PMF of A , then $p_X(k)$ can be found by summing $p_A(k)$ over all the values of A for which $X = k$. For example, if $A = [a_6, a_5, a_4, a_3, a_2, a_1, a_0]$ and $X = [a_3, a_2]$, then $p_X(k)$ is given as follows:

$$p_X(k) = \sum_{\substack{a_6, a_5, a_4, \\ a_1, a_0 \in \{0, 1\}}} p_A \left(\begin{matrix} 2^6 a_6 + 2^5 a_5 + 2^4 a_4 \\ + 2^2 k + 2 a_1 + a_0 \end{matrix} \right) \quad (7)$$

for $k = 0, 1, 2, 3$. The above equation can also be written as a double summation:

$$p_X(k) = \sum_{i=0}^{2^3-1} \sum_{j=0}^{2^2-1} p_A(2^4 i + 2^2 k + j) \quad (8)$$

This relationship is generalized as follows:

$$p_X(k) = \sum_{i=0}^{2^{N-1-k_2}-1} \left(\sum_{j=0}^{2^{k_1}-1} p_A(2^{k_2+1} i + 2^{k_1} k + j) \right) \quad (9)$$

for $0 \leq k \leq 2^{k_2-k_1+1} - 1$. Similarly, $p_Y(\cdot)$ can be evaluated from $p_B(\cdot)$. Assuming that the two adder inputs A and B , and hence X and Y , are independent, the PMF of the sum $Z = X + Y$ is given by the convolution of p_X and p_Y .

$$p_Z(k) = p_X(k) * p_Y(k), \text{ for } 0 \leq k \leq 2^{k_2-k_1+2} - 2 \quad (10)$$

If A and B are uniformly distributed between 0 and $2^n - 1$, then X and Y are uniformly distributed between 0 and $2^n - 1$, i.e.,

$$p_X(k; n) = p_Y(k; n) = \begin{cases} \frac{1}{2^n}, & \text{for } 0 \leq k \leq 2^n - 1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The PMF of sum is given as follows:

$$p_Z(k; n) = p_X(k; n) * p_Y(k; n) \quad (12a)$$

$$p_Z(k; n) = \begin{cases} \frac{k+1}{2^{2n}}, & 0 \leq k \leq 2^n - 1 \\ \frac{2^{n+1} - k - 1}{2^{2n}}, & 2^n - 1 < k \leq 2^{n+1} - 2 \\ 0, & \text{otherwise} \end{cases} \quad (12b)$$

Note that in case of uniformly distributed inputs, the distributions p_X and p_Y only depend on the number of bits $n = k_2 - k_1 + 1$, while in case of non-uniformly distributed inputs, bit locations (k_1, k_2) are also significant.

4.3.1 Probability of Event P

Carry-in is propagated by an n -bit addition when all n pairs of the input bits are propagating. This happens when the values of bits in every pair of addition arguments are unequal. When every pair of bits satisfy this condition, the carry-out will be equal to carry-in and the sum will be equal to $2^n - 1$. Hence, the probability of carry-in propagation by adding X and Y is given as follows:

$$\Pr[P; k_1, k_2] = p_Z(2^n - 1) \quad (13)$$

where $Z = X + Y$ and $p_Z(k) = p_X(k) * p_Y(k)$. $\Pr[P; k_1, k_2]$ denotes the probability of propagation event for given k_1, k_2 , indicating that p_X and p_Y depend upon the bit locations k_1, k_2 .

In case of uniformly distributed inputs, probability of carry-in propagation is evaluated by using $k = 2^n - 1$ in Eq. (12). In this case, since all the bits are identically distributed, it only depends upon the number of bits n . Hence,

$$\Pr[P; n] = \frac{1}{2^n} \quad (14)$$

4.3.2 Probabilities of Events $G0$ and $G1$

$G0$ denotes the event for carry-out generation provided that carry-in for n -bit addition is zero, i.e.,

$$\Pr[G0; k_1, k_2] = \Pr[\text{Carry-out} = 1 | \text{Carry-in} = 0] \quad (15)$$

Since carry-out is one when the sum Z cannot be accommodated in n bits, $\Pr[G0]$ is given as follows:

$$\Pr[G0; k_1, k_2] = \Pr[Z > 2^n - 1] = \sum_{k=2^n}^{2^{n+1}-2} p_Z(k) \quad (16)$$

For uniformly distributed inputs, $\Pr[G0; n]$ is found by using $p_Z(\cdot)$ from Eq. (12) in Eq. (16):

$$\Pr[G0; n] = \sum_{k=2^n}^{2^{n+1}-2} \frac{2^{n+1} - k - 1}{2^{2n}} = \frac{2^n - 1}{2^{n+1}} \quad (17)$$

Similarly, $G1$ denotes the event for carry-out generation provided that carry-in for n -bit addition is one.

$$\Pr[G1; k_1, k_2] = \Pr[\text{Carry-out} = 1 | \text{Carry-in} = 1] \quad (18)$$

$$\begin{aligned} \Pr[G1; k_1, k_2] &= \Pr[X + Y + 1 > 2^n - 1] \\ &= \Pr[Z + 1 > 2^n - 1] = \Pr[Z > 2^n - 2] \end{aligned} \quad (19)$$

$$\Pr[G1; k_1, k_2] = \sum_{k=2^{n-1}}^{2^{n+1}-2} p_Z(k) \quad (20)$$

For uniformly distributed inputs, Eq. (12) is used in Eq. (20) to get $\Pr[G1; n]$ as follows:

$$\begin{aligned} \Pr[G1; n] &= \frac{1}{2^n} + \sum_{k=2^n}^{2^{n+1}-2} \frac{2^{n+1} - k - 1}{2^{2n}} \\ &= \frac{1}{2^n} + \frac{2^n - 1}{2^{n+1}} \end{aligned} \quad (21)$$

Analytical model for any configuration of approximate adder in Fig. 2 has been developed by first representing the $\Pr[\text{Error}]$ by the inclusion-exclusion principle as given in Eq. (2), then representing each term as product of probabilities using the approach in Sub-section 4.2 and then each of the probabilities in the products will be substituted with one of the probabilities $\Pr[P]$, $\Pr[G0]$ and $\Pr[G1]$, derived above. Note that the proposed analysis is exact for uniformly distributed inputs, while approximate for non-uniformly distributed inputs. This is because expressing the joint probabilities as products of probabilities implies that the respective groups of input bits are independent, which may not be true for non-uniformly distributed inputs. However, simulation results in Section 5 show that it is a good approximation.

4.4 Probability Distribution of Error Value

In Sub-sections 4.1–4.3, we found the overall probability of error in an approximate adder output. Now, we find the PMF of error value, aiming at understanding the distribution of this overall probability among all the possible error values. This is done by individually considering all possible error cases. In every case, error value is found by identifying the sub-adders with erroneous outputs and the weights of carry bits that are discarded due to the carry chain truncation. The corresponding probabilities are computed using the same concepts that we developed in Sub-sections 4.1–4.3.

As explained in Section 3, the error in an approximate adder is due to the missed addition of carry bit(s) due to the broken carry chain. Consequently, the approximate sum is deficient by an amount equal to the weight of this bit location. Here, we define the error value $V = S_{\text{prec}} - S_{\text{app}}.$. For example, in case of an approximate adder with two sub-adders, the error value can be either 2^{S_1} or 0. The error value distribution p_V is given as follows:

$$p_V(x) = \begin{cases} \Pr[E_2], & x = 2^{S_1} \\ 1 - \Pr[E_2], & x = 0 \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

In case of an approximate adder with 3 sub-adders, there are three possible mutually exclusive error events: $(E_2 \wedge \overline{E}_3)$, $(\overline{E}_2 \wedge E_3)$ and $(E_2 \wedge E_3)$. The error value V in each case depends on the number and positions of carry-out generation events. If S_2 partially overlaps with R_3 , p_V is given as follows:

$$p_V(x) = \begin{cases} \Pr[E_2 \wedge \overline{E}_3], & x = 2^{S_1} \\ \Pr[\overline{E}_2 \wedge E_3], & x = 2^{S_1+S_2} \\ \Pr[E_2 \wedge E_3], & x = 2^{S_1} + 2^{S_1+S_2} \\ 1 - \Pr[E_2 \vee E_3], & x = 0 \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

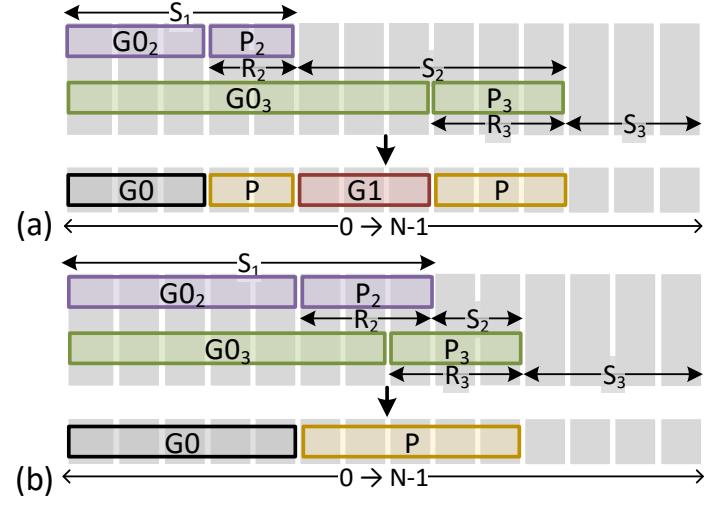


Fig. 6. Error value in case of simultaneous error in multiple sub-adders depends upon the number of carry-generation events. The events leading to $(E_2 \wedge E_3)$ are shown for two different approximate adders having 3 sub-adders (a) If S_2 partially overlaps with R_3 , there are two carry-out generation events. Note that $G1 = G0 \vee P$, but in order to keep the analysis simple, we only consider $G1$ as a generation event, as $\Pr[G0] > \Pr[P]$ for more than one bit. (b) If S_2 completely overlaps with R_3 , there is only one carry-out generation event.

When both Sub-adders 2 and 3 are erroneous, there are two carry-out generation events, as shown in Fig. 6(a). Hence, the cumulative error is due to two missed additions of carry bits at locations S_1 and $S_1 + S_2$. If S_2 completely overlaps with R_3 , simultaneous error in both the Sub-adders 2 and 3 implies only one missed addition of carry bit at location S_1 , as shown in Fig. 6(b). Hence p_V is given as follows:

$$p_V(x) = \begin{cases} \Pr[E_2 \wedge \overline{E}_3] + \Pr[E_2 \wedge E_3], & x = 2^{S_1} \\ \Pr[\overline{E}_2 \wedge E_3], & x = 2^{S_1+S_2} \\ 1 - \Pr[E_2 \vee E_3], & x = 0 \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

These PMFs are verified in Section 5.

The examples discussed above suggest that in order to find the distribution of error for an approximate adder with L sub-adders, we need to find the probability of all the possible mutually exclusive error events, i.e., $\Pr \left[\bigwedge_{i \in I} E_i \wedge \bigwedge_{j \in S-I} \overline{E}_j \right]$, where $S = \{2, 3, \dots, L\}$, $I \in \mathbb{P}(S)$, and the corresponding error values. The probabilities of the form $\Pr[E_2 \wedge E_3 \wedge \overline{E}_4 \wedge \overline{E}_5]$ (for $L = 5$) are evaluated using the law of total probability [27] as follows:

$$\begin{aligned} &\Pr[E_2 \wedge E_3] = \\ &\Pr[E_2 \wedge E_3 \wedge (\overline{E}_4 \wedge \overline{E}_5)] + \Pr[E_2 \wedge E_3 \wedge (\overline{E}_4 \wedge \overline{E}_5)] \\ &\implies \Pr[E_2 \wedge E_3 \wedge (\overline{E}_4 \wedge \overline{E}_5)] = \\ &\Pr[E_2 \wedge E_3] - \Pr[E_2 \wedge E_3 \wedge (E_4 \vee E_5)] \end{aligned} \quad (25)$$

where $\Pr[E_2 \wedge E_3 \wedge (E_4 \vee E_5)] = \Pr[E_2 \wedge E_3 \wedge E_4] + \Pr[E_2 \wedge E_3 \wedge E_5] - \Pr[E_2 \wedge E_3 \wedge E_4 \wedge E_5]$, which contains only joint probabilities that can be evaluated using the method in Sub-section 4.2.2, along with Algorithm 2. Using this approach, the distribution of error values in an N -bit approximate adder with L sub-adders is found and compared with simulation results in Section 5. In case of large L , this analysis can be automated using Algorithms 3 and 4.

4.5 Circuits with Multiple Approximate Adders

It was observed via simulations that if the approximate adders are cascaded, then the individual bits in the output of adders

Algorithm 3 PMF of Error Value

```

1: Input  $L, S_1, S_2, \dots, S_L$  and  $R_2, R_3, \dots, R_L$ .
2: Find the power set  $P_S = \mathbb{P}(S), S = \{2, 3, \dots, L\}$ .
3: Initialize arrays  $V = []$  and  $p_V = []$ .
4: for each  $i \in \{1, 2, \dots, L - 1\}$  do
5:   for every  $j^{th}$  ( $1 \leq j \leq \binom{L-1}{i}$ ),  $i$ -sized element  $P_j^{(i)} \in P_S$  do
6:     Find  $\Pr_i = \Pr \left[ \bigwedge_{m \in P_j^{(i)}} E_m \wedge \bigwedge_{n \in S - P_j^{(i)}} \overline{E_n} \right]$  using Algorithm 4,
      with inputs  $L, P_j^{(i)}, S_1, S_2, \dots, S_L$  and  $R_2, R_3, \dots, R_L$ .
7:     Run Algorithm 2 with inputs  $P_j^{(i)}, S_1, S_2, \dots, S_L$  and
       $R_2, R_3, \dots, R_L$  to get the error value  $v$ .
8:     if  $v \in V$  then
9:       Add  $\Pr_i$  to  $p_V(v)$ .
10:    else
11:      Append  $v$  to  $V$  and  $\Pr_i$  to  $p_V$ .
12:    end if
13:  end for
14: end for
15: Set  $p_V(0) = 1 - \sum p_V$ .
16: Output  $V$  and  $p_V$ .

```

Algorithm 4 Evaluation of $\Pr \left[(\bigwedge_{i \in I} E_i) (\bigwedge_{j \in J} \overline{E_j}) \right]$

```

1: Input  $L$ , the set of sub-adder indices  $I, S_1, S_2, \dots, S_L$  and
    $R_2, R_3, \dots, R_L$ .
2: Find the set difference  $J = \{2, 3, \dots, L\} - I$ .
3: Find  $P_1 = \Pr[\bigwedge_{i \in I} E_i]$  using Algorithm 2, with inputs  $I, S_1, S_2, \dots, S_L$  and  $R_2, R_3, \dots, R_L$ .
4: Find the power set  $P_J$  of set  $J$ .
5: Initialize  $P_2 = 0$ .
6: for  $j \in \{1, 2, \dots, \#J\}$  ( $\#$  denotes cardinality of a set) do
7:   for every  $j$ -sized element  $P_j^{(j)}$  of  $P_J$  do
8:     Find  $P'_2 = \Pr \left[ \left( \bigwedge_{i \in I} E_i \right) \left( \bigwedge_{j \in P_j^{(j)}} E_j \right) \right]$ , using Algorithm 2,
      with inputs  $I \cup P_j^{(j)}, S_1, S_2, \dots, S_L$  and  $R_2, R_3, \dots, R_L$ .
9:     Update  $P_2 = P_2 + (-1)^{j+1} P'_2$ .
10:  end for
11: end for
12: Find  $\Pr_{IJ} = P_1 - P_2$ .
13: Output  $\Pr_{IJ}$ .

```

still remain approximately uniformly distributed. This is because sum from every single full adder can be 1 or 0 with equal probability. They are not exactly uniformly distributed because of the occurrence of error. However, since the occurrence of error is infrequent in most adder configurations, we found that the distribution of output bits is going to be very close to the uniform distribution. Thus the probability of error models remain approximately the same for every approximate adder and they can be applied by considering each adder as an independent component.

Consider a circuit with M identically constructed, N -bit approximate adders, connected together to perform a more complex arithmetic operation involving the addition of $M + 1$ N -bit numbers. We further assume that all the $M + 1$ inputs are independent. If the probability of error in each adder is ρ , then the probability of error in the final addition is approximated using the *inclusion-exclusion principle* as follows:

$$\Pr[\text{Error}] \approx \sum_{m=1}^M \binom{M}{m} (-1)^{m+1} \rho^m \quad (26)$$

Since all the $M + 1$ inputs are independent, the PMF of cumulative error is estimated by the convolution of the PMFs of error in individual adders [27].

$$p_{V_M} \approx \underbrace{p_V * p_V * \dots * p_V}_{M \text{ convolutions}} \quad (27)$$

Two examples of circuits with multiple approximate adders are shown in Fig. 7. The exact configuration in which the adders are connected may differ from application to application. The configuration in Fig. 7(a) can be used in filter-

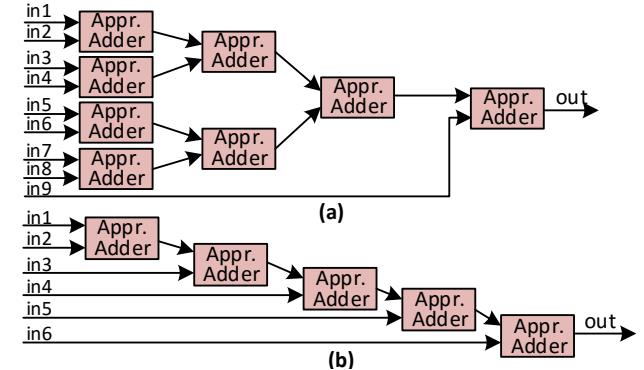


Fig. 7. Circuits with multiple approximate adders.

ing applications. Multiple approximate additions can also be performed in pipelined and sequential circuits, which may be functionally similar to configuration in Fig. 7(b). Both types of configurations are evaluated in Section 5.

5 CASE STUDIES

In this section, the proposed analysis is applied to the ACA-II [3], GeAr [15] and ETAIIM [4] adders. The results are validated by comparing them with simulation results. Performance evaluation for cascades of approximate adders is also carried out and compared with simulation results.

5.1 Probability of Error in Accuracy Configurable Adder (ACA-II)

The generic adder model, given in Fig. 2, is configured to ACA-II by setting $S_1 = 2k, R_2 = R_3 = \dots = R_L = R = k$ and $S_2 = S_3 = \dots = S_L = S = k$, where k is the half of carry-chain length. The general expression for probability of error in the output of ACA-II with three sub-adders is given by,

$$\Pr[\text{Error}; k] = \Pr[E_2] + \Pr[E_3] - \Pr[E_2 \wedge E_3] \quad (28)$$

Now the probability of error in the i^{th} sub-adder of ACA-II can be obtained by using Eq. (4) as follows:

$$\Pr[E_i] = \Pr[P_i; (i-1)k, ik-1] \Pr[G0_i; 0, (i-1)k-1] \quad (29)$$

where $\Pr[P; k_1, k_2]$ and $\Pr[G0; k_1, k_2]$ are found using Eqs. (13) and (16), respectively. Similarly, using Eq. (5), we get

$$\Pr[E_2 \wedge E_3] = \Pr[P_2 \wedge G0_2 \wedge P_3 \wedge G0_3] \quad (30)$$

In order to handle the joint probability term in Eq. (28), we transform the dependent events into independent events according to the proposed methodology, as shown in Fig. 8. Now the probability of the joint event is given as follows:

$$\Pr[E_2 \wedge E_3] = \Pr[P; k, 3k-1] \Pr[G0; 0, k-1] \quad (31)$$

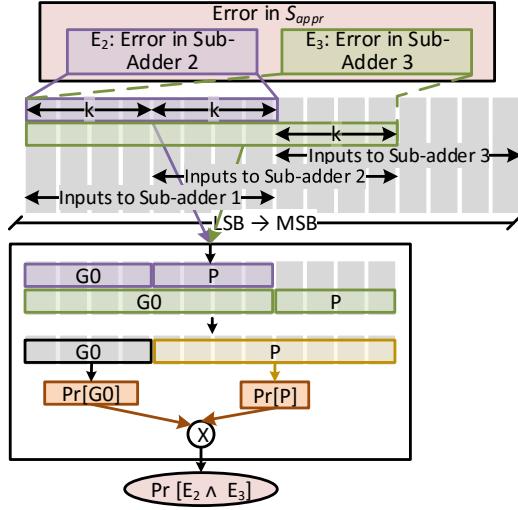
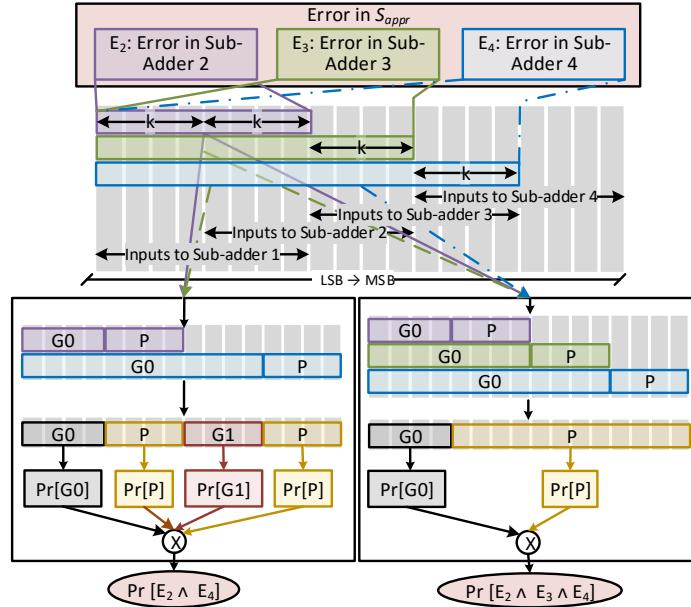
For uniformly distributed inputs, $\Pr[P]$ and $\Pr[G0]$ only depend on the number of bits. Therefore, the probability of error comes out to be as follows:

$$\Pr[\text{Error}; k] = \frac{1}{2^k} \frac{2^k - 1}{2^{k+1}} + \frac{1}{2^k} \frac{2^{2k} - 1}{2^{2k+1}} - \frac{1}{2^{2k}} \frac{2^k - 1}{2^{k+1}} \quad (32)$$

Similarly, an equivalent expression for $\Pr[\text{Error}]$ can be derived for the ACA-II adder with four sub-adders,

$$\begin{aligned} \Pr[\text{Error}] &= \Pr[E_2] + \Pr[E_3] + \Pr[E_4] \\ &\quad - \Pr[E_2 \wedge E_3] - \Pr[E_3 \wedge E_4] \\ &\quad - \Pr[E_2 \wedge E_4] + \Pr[E_2 \wedge E_3 \wedge E_4] \end{aligned} \quad (33)$$

The transformation of dependent events into equivalent independent events for the intersection terms ($E_2 \wedge E_4$) and

Fig. 8. Derivation of $\Pr[E_2 \wedge E_3]$ for ACA-II with three sub-adders.Fig. 9. Derivation of $\Pr[E_2 \wedge E_4]$ and $\Pr[E_2 \wedge E_3 \wedge E_4]$ for ACA-II with four sub-adders.

$(E_2 \wedge E_3 \wedge E_4)$ is illustrated in Fig. 9. For uniformly distributed inputs, $\Pr[Error]$ comes out to be as follows:

$$\begin{aligned} \Pr[Error; k] &= \frac{1}{2^k} \frac{2^k - 1}{2^{k+1}} + \frac{1}{2^k} \frac{2^{2k} - 1}{2^{2k+1}} + \frac{1}{2^k} \frac{2^{3k} - 1}{2^{3k+1}} \\ &- \frac{1}{2^{2k}} \frac{2^k - 1}{2^{k+1}} - \frac{1}{2^{2k}} \frac{2^{2k} - 1}{2^{2k+1}} - \frac{1}{2^{2k}} \frac{2^k - 1}{2^{k+1}} \left(\frac{2^k - 1}{2^{k+1}} + \frac{1}{2^k} \right) \\ &+ \frac{1}{2^{3k}} \frac{2^k - 1}{2^{k+1}} \end{aligned} \quad (34)$$

Fig. 10 shows a comparison of the proposed analysis with simulation results. It can be seen that the proposed analysis is in perfect agreement with the results obtained via exhaustive simulations.

5.1.1 Comparison with Approximate ACA-II Model

Fig. 10 also shows comparison with the probability of error model presented in [3]. This model assumes that the sub-adder outputs are independent of one another. The assumption simplifies the analysis but the model can compute only approximate probability of error. We found that the difference between exact probability of error and that predicted by the model in [3] increases as number of sub-adders increases from three to

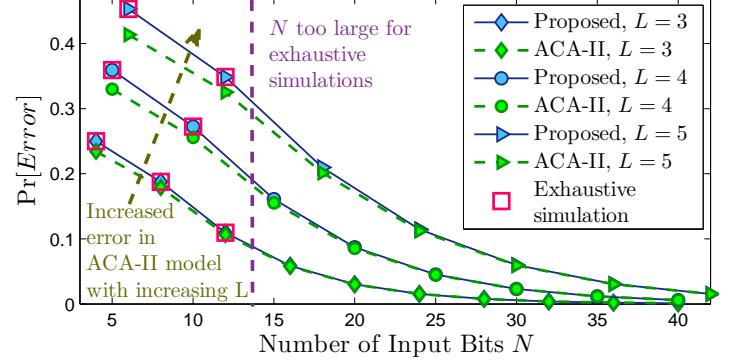


Fig. 10. Comparison of proposed model with exhaustive simulations and approximate model for the ACA-II adder in [3].

five. Also with increasing number of input bits N , we found that this difference decreases and the exact probability of error approaches the approximation given in [3]. This is because of the fact that with increasing N , if the number of sub-adders is kept constant, then the number of prediction bits increases in each sub-adder that in turn increases the probability of correct carry prediction and the outputs of different sub-adders become less *inter-dependent*.

5.2 Probability of Error in Generic Accuracy Configurable (GeAr) Adder

The generic adder, given in Fig. 3, can be configured to represent the GeAr adder [15] by considering $R_2 = R_3 = \dots = R_L = R$, $S_2 = S_3 = \dots = S_L = S$ and $S_1 = S + R$. In GeAr model, R and S may or may not be equal, but they are same for all sub-adders. For three sub-adders, the probability of error can be represented using the inclusion-exclusion principle, given in Eq. (2), as follows:

$$\Pr[Error] = \Pr[E_2] + \Pr[E_3] - \Pr[E_2 \wedge E_3] \quad (35)$$

$\Pr[E_i]$ for given R and S is given as follows:

$$\begin{aligned} \Pr[E_i] &= \Pr[P; (i-1)S, (i-1)S + R - 1] \\ &\times \Pr[G0; 0, (i-1)S - 1] \end{aligned} \quad (36)$$

The event represented by the intersection term in Eq. (35) was found to be different for GeAr configurations with $R \geq S$ and $R < S$. After the transformation of events, according to the proposed methodology, the general expressions for the GeAr adder with three sub-adders are given as follows: For $R \geq S$,

$$\begin{aligned} \Pr[Error; S, R] &= \Pr[P; S, S + R - 1] \Pr[G0; 0, S - 1] \\ &+ \Pr[P; 2S, 2S + R - 1] \Pr[G0; 0, 2S - 1] \\ &- \Pr[P; S, 2S + R - 1] \Pr[G0; 0, S - 1] \end{aligned} \quad (37)$$

For uniformly distributed inputs, $\Pr[Error]$ is a function of S and R :

$$\begin{aligned} \Pr[Error; S, R] &= \frac{1}{2^R} \frac{2^S - 1}{2^{S+1}} + \frac{1}{2^R} \frac{2^{2S} - 1}{2^{2S+1}} \\ &- \frac{1}{2^{S+R}} \frac{2^S - 1}{2^{S+1}} \end{aligned} \quad (38)$$

Similarly for $R < S$,

$$\begin{aligned} \Pr[Error; S, R] &= \Pr[P; S, S + R - 1] \Pr[G0; 0, S - 1] \\ &+ \Pr[P; 2S, 2S + R - 1] \Pr[G0; 0, 2S - 1] \\ &- \Pr[P; S, 2S + R - 1] \Pr[G0; 0, S - 1] \\ &\times \Pr[P; 2S, 2S + R - 1] \Pr[G1; S + R, 2S - 1] \end{aligned} \quad (39)$$

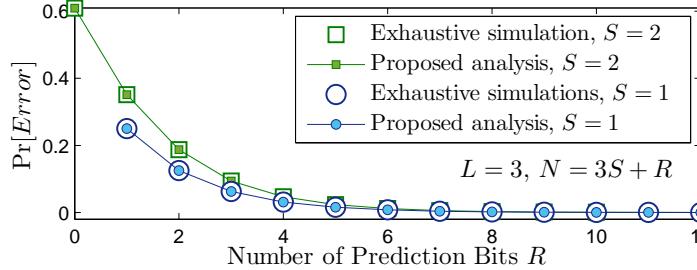


Fig. 11. Comparison of proposed model with exhaustive simulations for the GeAr adder in [15].

For uniformly distributed inputs, $\text{Pr}[\text{Error}]$ is given as follows:

$$\begin{aligned} \text{Pr}[\text{Error}; S, R] = & \frac{1}{2^R} \frac{2^S - 1}{2^{S+1}} + \frac{1}{2^R} \frac{2^{2S} - 1}{2^{2S+1}} \\ & - \frac{1}{2^{2R}} \frac{2^S - 1}{2^{S+1}} \left(\frac{2^{S-R} - 1}{2^{S-R+1}} + \frac{1}{2^{S-R}} \right) \end{aligned} \quad (40)$$

Fig. 11 shows a comparison of analysis and simulation results for uniformly distributed inputs, which validates the proposed analysis. Table 1 shows results for various input distributions for ACA-II and GeAr adders. It can be seen that the analysis and simulation results are in good agreement. PMFs for GeAr adders are evaluated using the method described in Sub-section 4.4. For $L = 3$, PMFs are given by Eqs. (24) and (23). Results for inputs with uniform and geometric distributions are shown in Fig. 12, which show that the analysis and simulation results are in good agreement.

5.3 Probability of Error in Modified Error Tolerant Adder-II (ETAIIM)

In some approximate adders, including ETAIIM and some configurations of GDA, the number of prediction and sum bits differ from sub-adder to sub-adder. The lengths of individual sub-adders may also vary. These cases are not covered by the GeAr model. The approximate adder model in Section 3 is a more general model, so the proposed methodology can also be applied to compute the $\text{Pr}[\text{Error}]$ in ETAIIM and GDA.

In this case study, we consider a structure similar to the ETAIIM [4] with three sub-adders, i.e., $L = 3$. In this design, the number of prediction bits in the 3rd sub-adder is larger, so the probability of error in more significant bits is smaller. Considering the case with $S_2 = S_3 = S$ and $R_3 = 2R_2$, the $\text{Pr}[\text{Error}]$ for uniformly distributed inputs is evaluated as follows:

$$\begin{aligned} \text{Pr}[\text{Error}; S, R] = & \frac{1}{2^{R_2}} \frac{2^S - 1}{2^{S+1}} + \frac{1}{2^{R_3}} \frac{2^{N-S-R_3} - 1}{2^{N-S-R_3+1}} \\ & - \frac{1}{2^{R_3}} \frac{2^S - 1}{2^{S+1}} \end{aligned} \quad (41)$$

Table 1 shows the results for ETAIIM with various input distributions.

5.4 Probability of Error in Circuits with Multiple Approximate Adders

Using the approximation in Eq. (26), the probability of error in case of multiple additions is evaluated for several approximate adders. In all the simulations, the approximate adders are connected together in the form of a cascade, similar to the one shown in Fig. 7(b). The results of analysis are compared with those of Monte-Carlo simulations in Fig. 13(a). For estimation of probability using the Monte-Carlo simulations, 100000 randomly generated input combinations were used. In every considered case for the comparison, all the approximate adders

TABLE 1
Comparison of probability of error computed using the proposed analysis with Monte-Carlo simulation results.

Adder	Input Distribution	Pr[Error]	
		Analysis	Monte-Carlo
ACA-II $N=12, k=4, L=2$	Uniform	0.0293	0.0293
	Gaussian $\mu = 2040, \sigma = 295.6$	0.0293	0.0290
	Geometric $p = 0.01$	0.0164	0.0168
ACA-II $N=16, k=4, L=3$	Uniform	0.0586	0.0586
	Gaussian $\mu = 32760, \sigma = 4729$	0.0586	0.0587
	Geometric $p = 0.0005$	0.04910	0.0498
ACA-II $N=20, k=4, L=4$	Uniform	0.0870	0.0870
	Gaussian $\mu = 5 \times 10^5, \sigma = 7 \times 10^4$	0.0870	0.0853
	Geometric $p = 0.00005$	0.0699	0.0703
GeAr N=16, $R=7, S_3=3, L=3$	Uniform	0.0068	0.0067
	Geometric $p = 0.0005$	0.0043	0.0043
GeAr N=16, $R=1, S_5=5, L=3$	Uniform	0.4352	0.4354
	Geometric $p = 0.0005$	0.3918	0.3857
GeAr N=16, $R=4, S_3=3, L=4$	Uniform	0.0814	0.0813
	Geometric $p = 0.0005$	0.0607	0.0603
GeAr N=16, $R=0, S_4=4, L=4$	Uniform	0.8501	0.8501
	Geometric $p = 0.0005$	0.7584	0.7591
ETA-II-M, $N=16, L=3, S_1=8, S_2=S_3=4, R_2=4, R_3=8$	Uniform	0.0293	0.0293
	Geometric $p = 0.0005$	0.0292	0.0291
	Gaussian $\mu = 32760, \sigma = 4729$	0.0292	0.0290

are identical and all inputs are independent and uniformly distributed. It can be seen that the simulation results are in good agreement with the proposed analysis.

Using Eq. (27), PMF of error in case of cascaded adders was approximated. Mean Error Distance (MED), which is one of the commonly used performance metrics in approximate computing, is found using p_V as follows:

$$MED = \sum_{v=-\infty}^{\infty} |v| p_V(v) \quad (42)$$

The computed MED is compared with simulation results in Fig. 13(b).

5.5 Observations and Discussions

5.5.1 Accuracy of Analysis

The expressions for probability of error derived for uniformly distributed inputs using the proposed methodology are accurate in the sense that they precisely tell the fraction of input combinations for which the adder output will be inaccurate. This is confirmed by Figs. 10 and 11, in which the proposed analysis is compared with the results of exhaustive simulations. The exhaustive simulations are performed by evaluating all 2^{2N} input combinations. In case of non-uniformly distributed inputs, we introduced some approximations to keep the analysis simple, as explained in Sub-section 4.3. Comparisons in Table 1 show that the analytical and simulation results match well for both types of input distributions.

5.5.2 Complexity of Analysis

We observed that in case of large number of sub-adders, the derivation of exact probability of error using the proposed methodology becomes significantly complex. This is due to increase in the number of terms in Eq. (2). The number of terms in Eq. (2) in case of L sub-adders is $\sum_{i=1}^{L-1} \binom{L-1}{i}$. Therefore, for large L , Algorithms 1 and 2 can be used to automate/program the analysis, if required.

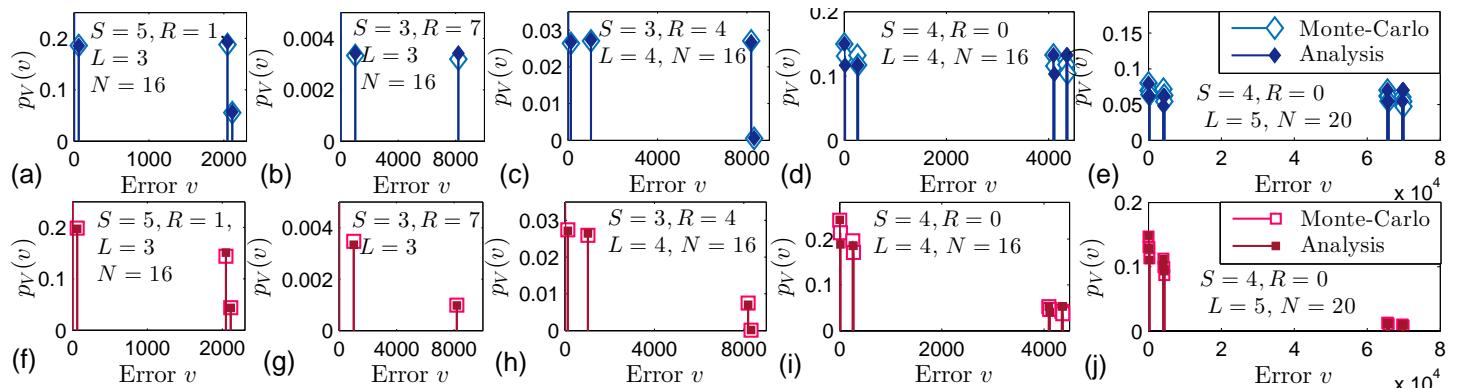


Fig. 12. Comparison of analysis and simulation results for PMFs of various configurations of GeAr adders. (a)-(e) Results with uniformly distributed inputs. (f)-(j) Results with geometrically distributed inputs.

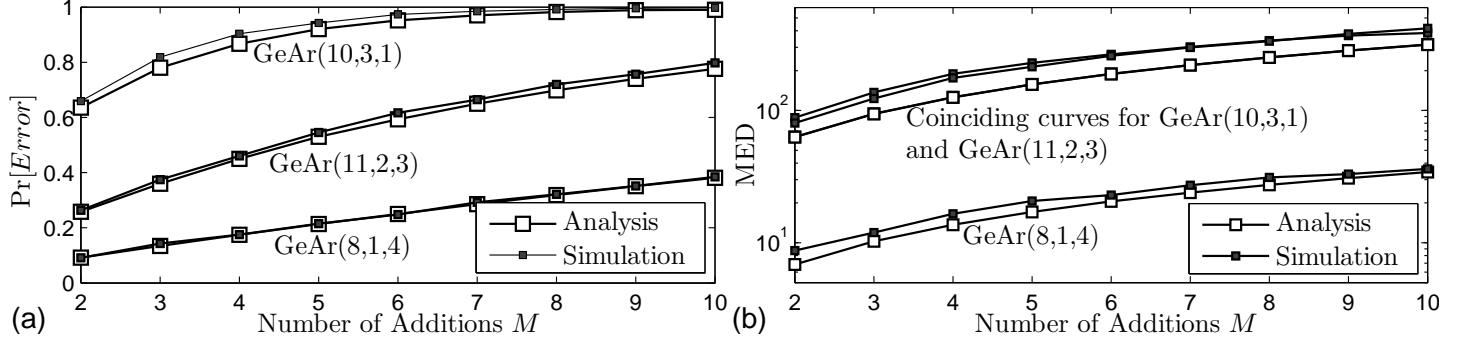


Fig. 13. Comparison of analysis and simulation results for of circuits with M adders using GeAr(N, S, R) adders.

5.5.3 Effects of Input Distribution

Table 1 shows the comparison of results with various input distributions. It can be seen that the results with Gaussian distribution are very close to those with uniform distribution. Similar behavior was observed for PMFs as well. For most input distributions, we found that the $\text{Pr}[Error]$ and PMF are not significantly affected, except in case of steeply decreasing geometric distributions. The results are shown in Table 1 and Fig 12. This is due to the fact that the probabilistic behavior of these adders depends upon the lower significance bits, which still remain approximately uniformly distributed. The input distribution largely affects the more significant bits, that are not responsible for error in these approximate adders. The geometric distributions used in Table 1 and Fig 12 are so rapidly decaying that it significantly affects the distribution of propagation bits of last sub-adder.

5.5.4 PMF of Approximation Error

The PMFs in Fig. 12 show that the approximation error in this class of adders can have certain specific values, which depend on the locations of carry chain truncation. As explained in Sections 3 and 4, the error in the output is equal to the sum of the weights of all the carry bits that are discarded due to the carry chain truncation and are not correctly predicted by the propagating bits in the respective sub-adders. As a result, the PMFs of approximation error are found to consist of widely separated impulses. The location of these impulses indicate the source of error, i.e., the sub-adder that yields the sum bits with the same weight.

5.5.5 Predicting Trade-offs among Accuracy, Speed and Area-on-chip

Figs. 14 and 15 show various trends of probability of error that can be observed using the derived models. In Fig. 14(a), $\text{Pr}[Error]$ has been plotted against number of prediction bits R for various values of N . This can provide useful information regarding the additional area-on-chip required to achieve

a certain constraint on accuracy. If all the sub-adders are implemented as Ripple Carry Adders (RCAs), number of full adders increase by $(L - 1)R$ as compared to a precise adder. The GeAr configurations were synthesized on a XILINX Virtex 6 XC6VLX75T FPGA and the area, in terms on FPGA look-up tables (LUTs), was found. The results are plotted in 14(b), which shows that the required number of LUTs increases approximately linearly with number of prediction bits, for a fixed N and L . The $\text{Pr}[Error]$ decreases with increasing R for fixed values of N and L . However, the curves for various values of N are found to overlap which means that there is no significant effect on $\text{Pr}[Error]$ with changing N . This happens because an increase in N with constant L and R means that S has been increased. It can be observed from Eqs. (17) and (21) that probability of carry-out generation rapidly approaches 0.5 as the number of bits n increases. From Eqs. (38) and (38), we also found that each of the $\text{Pr}[P]$ terms depends only upon R and every $\text{Pr}[G_0]$ and $\text{Pr}[G_1]$ term approaches 0.5 with increasing S . Therefore, for constant R and L , change in N or S has very little effect on $\text{Pr}[Error]$.

In Fig. 15, we have plotted $\text{Pr}[Error]$ against the carry-chain length $R + S$ that yields insights into the trade-off between speed and accuracy of the approximate adder. The delay of a sub-adder is equal to the delay of a $R + S$ -bit precise adder. The delay only depends upon $R + S$ and is not affected by N or L . It can be observed that $\text{Pr}[Error]$ decreases with increasing carry-chain length and to achieve same $\text{Pr}[Error]$, required carry-chain length $R + S$ increases with increasing N . This is because $\text{Pr}[Error]$ depends pre-dominantly upon R , as observed in Fig. 14(a). For constant R and L , if N increases, then S , and hence $R + S$ would increase. For constant N and L , increase in carry-chain length is a consequence of increased R that reduces the $\text{Pr}[Error]$. Moreover, by increasing R for same number of sub-adders, overlap between the inputs of successive sub-adders increases, which in turn requires larger area-on-chip.

It is important to note that we have predicted the trends

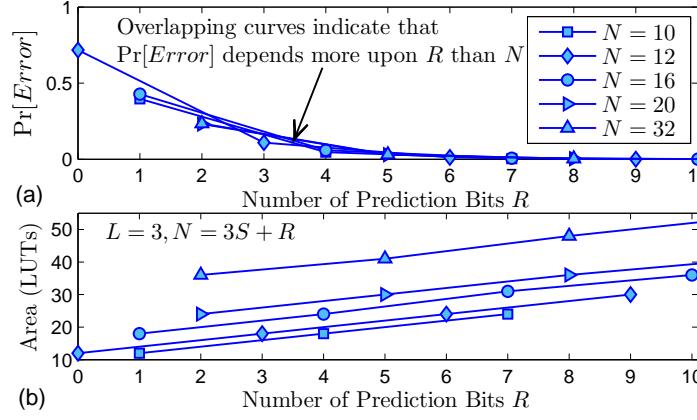


Fig. 14. (a) Probability of error in the GeAr adder, and (b) area on FPGA, with increasing number of prediction bits R .

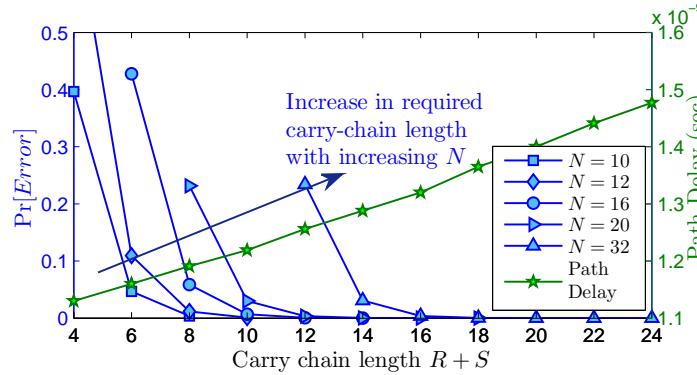


Fig. 15. Probability of error in the GeAr adder and the critical path delay with increasing carry-chain length. $L = 3, N = 3S + R$.

pertaining to trade-offs among accuracy, speed and area-on-chip using *analytical models* instead of performing time consuming Monte-Carlo simulations. Also it is significantly noteworthy that now we can mathematically explain how various circuit parameters are contributing towards causing inaccuracy in the adder's output that can provide greater level of control over the design of such circuits.

5.6 Predicting Error Performance in Image Processing

In order to illustrate the applicability of the proposed performance analysis to a practical problem, we have implemented image processing applications using several GeAr adders and computed the Error Rate (ER) and MED in these applications. Applications include Gaussian smoothing of noisy image, using a 3x3 and a 5x5 Gaussian kernel, and the Sum of Absolute Differences (SAD), using a 16-pixel kernel. For each case, the application was simulated using precise arithmetic and GeAr adders. In each case, the ER was found as the fraction of erroneous pixels and MED as the average error. The frequency of error was then computed using the analysis, described in Sub-sections 5.2 and 4.5, since both the applications involve multiple additions. It should be noted that the multiplications in Gaussian smoothing and differences in SAD are evaluated using precise components while the additions are performed using the approximate adders. The adders are connected together in configurations similar to the one in Fig. 7(a).

Table 2 shows results for the ER and MED in case of the Gaussian smoothing using 3x3 and 5x5 kernels, respectively. Comparisons are performed for several configurations of the GeAr adder, which are specified by N , L , R and S . We see that the results of the proposed analysis are close to the experimental results in all cases. Images obtained in case of Gaussian smoothing using 3x3 kernel are shown in Fig. 16.

TABLE 2
Comparison of probability of error computed using the proposed analysis with image processing results.

Adder	Evaluuated from	ER	MED ($\times 10^3$)
Gaussian Smoothing using 3x3 kernel. $M = 8, N = 16$			
GeAr $L = 3$ $R = 1, S = 5$	Image	0.9432	3.861
Analysis		0.9880	4.0944
GeAr $L = 3$ $R = 4, S = 4$	Image	0.3668	1.247
Analysis		0.3831	1.018
GeAr $L = 3$ $R = 7, S = 3$	Image	0.0391	0.197
Analysis		0.0534	0.245
GeAr $L = 3$ $R = 10, S = 2$	Image	0.00476	0.0498
Analysis		0.00584	0.0698
Gaussian Smoothing using 3x3 kernel. $M = 8, N = 8$			
GeAr $L = 4$ $R = 0, S = 2$	Image	0.9667	0.321
Analysis		0.999	0.254
GeAr $L = 3$ $R = 2, S = 2$	Image	0.691	0.071
Analysis		0.810	0.059
GeAr $L = 4$ $R = 4, S = 1$	Image	0.289	0.0311
Analysis		0.3189	0.028
GeAr $L = 3$ $R = 5, S = 1$	Image	0.051	0.0213
Analysis		0.118	0.0117
Gaussian Smoothing using 5x5 kernel. $M = 24, N = 16$			
GeAr $L = 3$ $R = 4, S = 4$	Image	0.7804	3.110
Analysis		0.7652	2.962
GeAr $L = 3$ $R = 7, S = 3$	Image	0.1801	0.678
Analysis		0.1518	0.815
GeAr $L = 4$ $R = 4, S = 3$	Image	0.8381	5.577
Analysis		0.8696	6.0724
GeAr $L = 4$ $R = 8, S = 2$	Image	0.0899	0.789
Analysis		0.1003	0.845
GeAr $L = 5$ $R = 6, S = 2$	Image	0.4042	3.471
Analysis		0.4340	3.042

Difference images are also shown along with the results to show that ER decreases with increasing R and the results in Table 2 show that it can be predicted with reasonable accuracy using the proposed analysis. Similar observation is made in Fig. 17 for the SAD application. Thus, we conclude that the proposed analysis can be applied for predicting the relative error performance in practical applications employing approximate adders.

Based on the case studies, presented in this section, we have observed that the proposed methodology provides an accurate and parametrized probability of error analysis. The accuracy enhances the reliability prediction of approximate adders and parametrization helps understanding the dependence of accuracy on various adder parameters, like size of adder and carry-chain length. These insights can be useful to the circuits designers in building more predictable and reliable circuits.

6 CONCLUSION

In this paper, a generic methodology for probability of error analysis of approximate adders is presented. The methodology can be applied to calculate exact probability of occurrence of error and the PMF of error in any configuration of the adder model presented for given input distributions and hence these configurations can be reliably compared without the need of exhaustive or Monte-Carlo simulations. The analytical models also yield insights into the dependence of error performance on circuit parameters. Models for some configurations are verified through exhaustive simulations and simulation results are shown to be in perfect agreement with the analysis. The proposed methodology can serve as a useful tool for predicting comparative error performance of various configurations. Comparative performance of some GeAr adder configurations

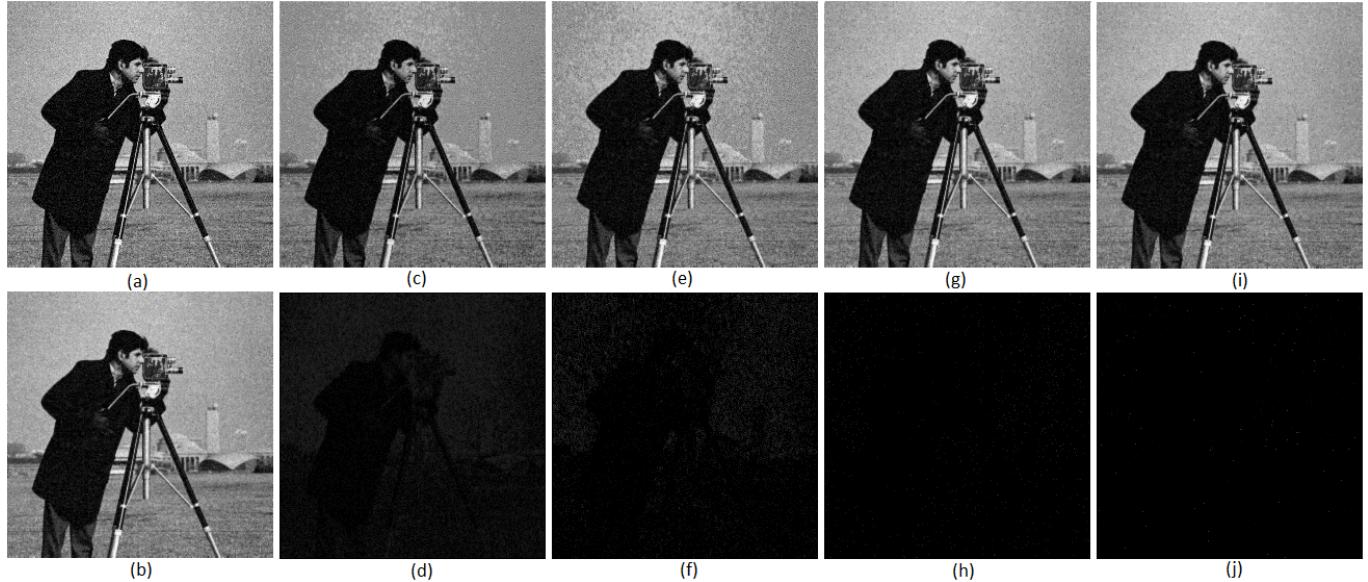


Fig. 16. Image smoothing using 3x3 kernel: (a) original image with noise.(b) Image filtered using precise arithmetic. Images filtered using 16-bit GeAr adders with three sub-adders and difference images showing the erroneous pixels. (c) Filtered image and (d) difference image for $R = 1, S = 5$. (e) Filtered image and (f) difference image for $R = 4, S = 4$. (g) Filtered image and (h) difference image for $R = 7, S = 3$. (i) Filtered image and (j) difference image for $R = 10, S = 2$.

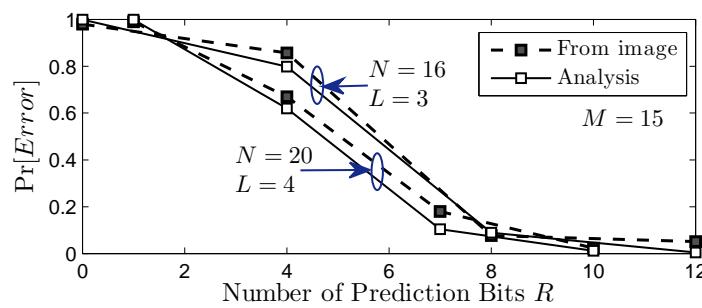


Fig. 17. Comparison of $\text{Pr}[\text{Error}]$ predicted by proposed analysis with SAD results, using a 16-pixel kernel.

in a practical application of Gaussian smoothing of an image is also predicted and verified.

REFERENCES

- [1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *18th IEEE European Test Symposium (ETS)*. IEEE, 2013, pp. 1–6.
- [2] S.-L. Lu, "Speeding up processing with approximation circuits," *Computer*, vol. 37, no. 3, pp. 67–73, 2004.
- [3] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proc. 49th Annual Design Automation Conference*. ACM, 2012, pp. 820–825.
- [4] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *in Proc. of 12th International Symposium on Integrated Circuits*. IEEE, 2009, pp. 69–72.
- [5] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *Proc. International Conference on Computer-Aided Design*. IEEE Press, 2013, pp. 48–54.
- [6] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *Proc. Design, Automation & Test in Europe*. IEEE, 2012, pp. 1257–1262.
- [7] S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-and-simplify: a unified design paradigm for approximate and quality configurable circuits," in *Proc. Design, Automation and Test in Europe*. EDA Consortium, 2013, pp. 1367–1372.
- [8] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in *Proc. International Conference on Computer-Aided Design*. ACM, 2012, pp. 728–735.
- [9] J. Miao, A. Gerstlauer, and M. Orshansky, "Approximate logic synthesis under general error magnitude and frequency constraints," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2013, pp. 779–786.
- [10] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate XOR/XNOR-based adders for inexact computing," in *Proc. 13th IEEE Conference on Nanotechnology*. IEEE, 2013, pp. 690–693.
- [11] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," *Journal of Low Power Electronics*, vol. 7, no. 4, pp. 490–501, 2011.
- [12] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Proc. conference on Design, Automation & Test in Europe*. EDA, 2014, p. 95.
- [13] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power-and area-efficient approximate wallace tree multiplier for error-resilient systems," in *15th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2014, pp. 263–269.
- [14] A. K. Verma, P. Brisk, and P. Ilenne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proc. Design, automation and test in Europe*. ACM, 2008, pp. 1250–1255.
- [15] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in *Proc. 52nd Annual Design Automation Conference*. ACM, 2015, p. 86.
- [16] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IM-PACT: imprecise adders for low-power approximate computing," in *Proc. 17th IEEE/ACM international symposium on Low-power electronics and design*. IEEE Press, 2011, pp. 409–414.
- [17] C. Lin, Y.-M. Yang, and C.-C. Lin, "High-performance low-power carry speculative addition with variable latency," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1591–1603, 2015.
- [18] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, 2013.
- [19] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 8, pp. 1225–1229, 2010.
- [20] S. Mazahir, O. Hasan, R. Hafiz, M. Shafique, and J. Henkel, "An area-efficient consolidated configurable error correction for approximate hardware accelerators," in *Proc. 53rd Annual Design Automation Conference*. ACM, 2016, p. 96.
- [21] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, 2013.
- [22] C. Liu, J. Han, and F. Lombardi, "An analytical framework for evaluating the error characteristics of approximate adders," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1268–1281, 2015.
- [23] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *Proc. 25th edition on Great Lakes Symposium on VLSI*. ACM, 2015, pp. 343–348.
- [24] I. Chong, H.-Y. Cheong, and A. Ortega, "New quality metric for multimedia compression using faulty hardware," in *Intl Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006.
- [25] Y. Liu, T. Zhang, and K. K. Parhi, "Computation error analysis in digital signal processing systems with overscaled supply voltage,"

- IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 4, pp. 517–526, 2010.
- [26] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, “MACACO: Modeling and analysis of circuits for approximate computing,” in *Proc. International Conference on Computer-Aided Design*. IEEE Press, 2011, pp. 667–673.
- [27] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*. Pearson/Prentice Hall, 2008.



Sana Mazahir received the B.E. degree (President's Gold Medal) in electrical engineering and the masters degree in electrical engineering (DSP and communications) from the College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), Pakistan, in 2013 and 2015, respectively. In 2015, she worked as a Research Assistant with the System Analysis and Verification Laboratory at NUST. In 2016, she was a Visiting Student at King Abdullah University of Science and Technology, Saudi Arabia, for six months. Her research interests include signal construction and processing for wireless communications and stochastic modeling and analysis of systems and approximate computing.



Muhammad Shafique (M11, SM16) is a full professor at Vienna University of Technology (TU Wien), Austria, where he is directing the Chair for Computer Architecture and Robust, Energy-Efficient Technologies (CARE-Tech). He was a senior research group leader at Karlsruhe Institute of Technology (KIT), Germany for more than 5 years after receiving his Ph.D. in Computer Science from KIT in January 2011. Before, he was with Streaming Networks Pvt. Ltd. where he was involved in research and development of video coding algorithms and optimizations for VLIW-based Processors for three years. His research interests are in power/energy-efficient, reliable, and adaptive computing systems covering various design abstractions of the hardware and software stacks (like micro-architecture, architecture, run-time system, and compiler), embedded systems, emerging technologies and computing paradigms, and their applications in Internet-of-things, Cyber-Physical Systems, and ICT for Developing Countries (ICTD).

Dr. Shafique received 2015 ACM/SIGDA Outstanding New Faculty Award, six gold medals, and several best paper awards and nominations at prestigious conferences like CODES+ISSS 2015, 2014, and 2011, DATE 2008, DAC 2016, and ICCAD 2009, Best Master Thesis Award, DAC 2014 Designer Track Best Poster Award and SS14 Best Lecturer Award. He is the TPC co-Chair of ESTIMedia 2015 and 2016, and has served on the TPC of several IEEE/ACM conferences like ICCAD, DATE, CASES, and ASPDAC. He is a senior member of the IEEE and IEEE Signal Processing Society (SPS). He holds one US patent.



Osman Hasan received the B.Eng. (Hons.) degree from the N-W.F.P University of Engineering and Technology, Pakistan, in 1997, and the M.Eng. and Ph.D. degrees from Concordia University, Montreal, Canada, in 2001 and 2008, respectively. He served as an ASIC design Engineer from 2001 to 2003 at LSI Logic Corporation in Ottawa, Canada and as a Research Associate at Concordia University, Montreal, Canada, for 18 months after his doctoral degree. Currently, he is an Assistant Professor at the School of Electrical Engineering and Computer Science, National University of Science and Technology (NUST), Islamabad, Pakistan. He is the founder and director of System Analysis and Verification (SAVe) Lab at NUST, which mainly focuses on the design and formal verification of embedded systems. He has received several awards and distinctions, including the Pakistans Higher Education Commissions Best University Teacher (2010) and Best Young Researcher Award (2011) and the Presidents gold medal for the best teacher of the University from NUST in 2015. Dr. Hasan is a Senior member of IEEE, member of Association for Automated Reasoning (AAR) and member of the Pakistan Engineering Council.



Jörg Henkel (M'95–SM'01–F'15) is currently with Karlsruhe Institute of Technology (KIT), Germany, where he is directing the Chair for Embedded Systems CES. Before, he was a Senior Research Staff Member at NEC Laboratories in Princeton, NJ. He received his PhD from Braunschweig University with Summa cum Laude. Prof. Henkel has/is organizing various embedded systems and low power ACM/IEEE conferences/symposia as General Chair and Program Chair and was a Guest Editor on these topics in various Journals like the IEEE Computer Magazine. He was Program Chair of CODES01, RSP02, ISLPED06, SIPS08, CASES09, Estimedia11, VLSI Design12, ICCAD12, PATMOS13, NOCS14 and served as General Chair for CODES02, ISLPED09, Estimedia12, ICCAD13 and ESWeek16. He is/has been a steering committee member of major conferences in the embedded systems field like at IC-CAD, ESWeek, ISLPED, Codes+ISSS, CASES and is/has been an editorial board member of various journals like the IEEE TVLSI, IEEE TCAD, IEEE TMSCS, ACM TCPS, JOLPE etc. In recent years, Prof. Henkel has given around ten keynotes at various international conferences primarily with focus on embedded systems dependability. He has given full/half-day tutorials at leading conferences like DAC, ICCAD, DATE, etc. Prof. Henkel received the 2008 DATE Best Paper Award, the 2009 IEEE/ACM William J. McCalla ICCAD Best Paper Award, the Codes+ISSS 2015, 2014, and 2011 Best Paper Awards, and the MaXentric Technologies AHS 2011 Best Paper Award as well as the DATE 2013 Best IP Award and the DAC 2014 Designer Track Best Poster Award. He is the Chairman of the IEEE Computer Society, Germany Section, and was the Editor-in-Chief of the ACM Transactions on Embedded Computing Systems (ACM TECS) for two consecutive terms. He is an initiator and the coordinator of the German Research Foundations (DFG) program on Dependable Embedded Systems (SPP 1500). He is the site coordinator (Karlsruhe site) of the Three- University Collaborative Research Center on Invasive Computing (DFG TR89). He is the Editor-in-Chief of the IEEE Design & Test Magazine since January 2016. He holds ten US patent and is a Fellow of the IEEE.



Rehan Hafiz received his Ph.D. degree in EE from The University of Manchester, UK in 2008. He is currently at Information Technology University (ITU) Lahore, as an Associate Professor in EE. Earlier he served as an Assistant Professor at the School of Electrical Engineering & Computer Science at National University of Sciences & Technology (NUST) from 2008 till 2015. He founded and directed the Vision Image & Signal Processing (ViSPro) Lab that focused on vision system design, development of power efficient architectures, design of approximate computing based hardware accelerators techniques, FPGA based and H/W/SW co-design, multi-projector and immersive display technologies and applied image and video processing. He holds several patents in US, South Korean and Pakistan patent office. He has published several articles related to custom processor design, approximate computing, application specific processor designing, video stabilization & multi projector rendering.