Probabilistic Error Analysis of Approximate Recursive Multipliers

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

Abstract—Approximate multipliers are gaining importance in energy-efficient computing and require careful error analysis. In this paper, we present the error probability analysis for recursive approximate multipliers with approximate partial products. Since these multipliers are constructed from smaller approximate multiplier building blocks, we propose to derive the error probability in an arbitrary bit-width multiplier from the probabilistic model of the basic building block and the probability distributions of inputs. The analysis is based on common features of recursive multipliers identified by carefully studying the behavioral model of state-of-the-art designs. By building further upon the analysis, Probability Mass Function (PMF) of error is computed by individually considering all possible error cases and their inter-dependencies. We further discuss the generalizations for approximate adder trees, signed multipliers, squarers and constant multipliers. The proposed analysis is validated by applying it to several state-of-the-art approximate multipliers and comparing with corresponding simulation results. The results show that the proposed analysis serves as an effective tool for predicting, evaluating and comparing the accuracy of various multipliers. Results show that for the majority of the recursive multipliers, we get accurate error performance evaluation. We also predict the multipliers’ performance in an image processing application to demonstrate its practical significance.

Index Terms—Approximate computing, Multipliers, Probability of error, Probabilistic analysis, Mathematical modeling, Arithmetic, Low power, Energy efficiency, Image processing.

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, machine learning and recognition. 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 accuracy for efficiency gain in terms of required hardware, speed and/or power.

Recently, low-power design of approximate logic circuits has been gaining a lot of interest [3]. Several techniques have been developed for the synthesis of energy-efficient hardware for computing applications [4], [5], [6], [7]. Since adders and multipliers are the most foundational building blocks in an arithmetic data path, several designs of functionally approximate low-power adders [4], [7], [8] and multipliers [7], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] have been proposed. In all these designs, computational accuracy is compromised to reduce power consumption. The requirements of computational accuracy and power consumption differ from application to application. Therefore, in order to deploy them in any application, it is necessary to quantify the approximation error in these circuits.

Traditionally, the computational accuracy in approximate adders and multipliers is evaluated using Monte-Carlo simulations. Most commonly used error metrics are Error Rate (ER) [13], [20], Mean Error Distance (MED) [20] and Mean Square Error (MSE) [11], [14]. All of these metrics are estimations derived from simulations, that are time consuming and require considerable programming effort. Also, the Monte-Carlo simulations evaluate the approximate circuits for randomly generated input combinations. As a result, it becomes difficult to relate error statistics with conditions on input and circuit architecture.

These limitations can be overcome by employing mathematical analysis for the performance evaluation and comparison of the approximate circuits. Owing to the importance of this mathematical analysis, recently, a number of efforts have been dedicated to develop methods for the accuracy analysis for various types of approximate modules [21], [22], [23], [24], [25]. A common conclusion from these works is that it is feasible to develop a generic analysis with reasonable complexity for components constructed from similar type of basic functional unit. In this regard, analysis of approximate multipliers has attracted relatively less attention. Therefore, in this paper, we present the probabilistic analysis of approximation error in low-power recursive approximate multipliers with approximate partial products. This represents a major class of low-power approximate multipliers, with a number of designs available to cover a wide range of operating conditions [7], [10], [11], [13], [14], [27].

Novel Contributions: In this paper, we present probabilistic error analysis for recursive approximate multipliers. The proposed multipliers under consideration are constructed from smaller approximate multiplier building blocks.

- The probability of error is analyzed for arbitrary-length approximate multipliers constructed from a given building block. The proposed method utilizes the probabilistic behavior of the constituent multiplier building blocks to compute the probability of error in larger multipliers.
- The analysis is generalized for any given input distributions.
- By building further upon the analysis, analysis for the Probability Mass Function (PMF) of error is developed.
- The analysis is applied to a number of state-of-the-art approximate multipliers and results are validated by comparing them with those obtained via simulations.
- The proposed PMF algorithm is also equipped to analyze multipliers constructed from a combination of approximate and precise building blocks.
- We also discuss generalization of the proposed analysis for its applicability to signed multipliers, squarers, constant multipliers and hybrid structures with approximations applied in partial product tree.
- The analysis is applied to predict the performance metrics in an image processing application.

It is important to note that the trends in existing literature [21], [22], [23], [24], [25] indicate that different analysis methods are required to analyze different types of approximations. Therefore, the proposed analysis is directly applicable to recursive low-power
multiplication [26, Type A] and their hybrid versions. A discussion towards the generalization of the proposed method is included to cover a larger design space for approximate multipliers.

Paper Organization: An overview of low-power approximate multipliers is given in Section 2. A general model for the recursive approximate multipliers is discussed in Section 3. In Section 4, we present the error probability and PMF analysis. In Section 5, analysis is compared with simulation result for several state-of-the-art approximate multipliers. Finally, some conclusions are drawn in Section 6.

2 AN OVERVIEW OF APPROXIMATE MULTIPLIERS

The approximate multipliers in existing works can be broadly classified into following two types:

- **Type A:** Multipliers with approximate partial products, that are added using precise adders. Larger multipliers are constructed from smaller multiplier building blocks. Kulkarni et al. [13] presented an approximate $2 \times 2$ multiplier. This $2 \times 2$ multiplier is used to generate partial products, that are added using precise components. Using $2 \times 2$ approximate multipliers for an $N$-bit multiplier generates $N^2/4$ partial products instead of $N^2$, thereby reducing the addition stages. Another approximate $2 \times 2$ multiplier designs are proposed by Shafique et al. [7] and Rehman et al. [10]. Lin et al. [27] proposed using a $4 \times 4$ approximate $4 : 2$ compressor based multiplier to generate partial products for larger multipliers. These multipliers are capable of providing a wide range of operation conditions with respect to area, power, speed and error statistics and thus can be considered as a major class of low-power multipliers.

- **Type B:** Multipliers with precise partial products and approximate adders or compressors for their addition. Momeni et al. [14] and Maheshwari et al. [11] proposed approximate Wallace tree multipliers constructed using approximate $4 : 2$ compressors. The approximate $4 : 2$ compressors simplify the logic by representing 3-bit result in 2 bits (sum and carry). As a result, adding stages in tree multiplier are reduced. Bhardwaj et al. [17] proposed an approximate Wallace tree multiplier in which carry prediction is employed in the addition of partial products. Liu et. al. [9] employed an approximate adder with limited carry propagation to reduce power consumption. Another example is the Broken-Booth multipliers presented by Farschi et al. [18]. Lin et al. [27] presented an approximate multiplier constructed from a combination of approximate counters and carry predictor in the addition of partial products.

A more detailed survey of approximate multipliers can be found in [10], [26]. Recently, an estimation based analysis method for Wallace tree multipliers with approximate Full Adders (FAs), which is a Type B multiplier, was presented in [25]. In this paper, we present a general analysis for Type A multipliers.

3 MULTIPLIER MODEL

Fig. 1 shows the multiplier model under consideration for analysis. An $N \times N$ approximate multiplier is constructed from $M \times M$ low-power approximate multiplier units, where $N = 2^k M$ ($k = 0, 1, 2, 3, \ldots, \log_2(N/M)$). As illustrated in Fig. 1, the $N \times N$ approximate multiplier is constructed from four $N/2 \times N/2$ approximate multipliers, each of which is in turn constructed from four $N/4 \times N/4$ approximate multipliers and so on. The additions of the approximate partial products at every stage are precise. This $2^k M \times 2^k M$ multiplier topology in Fig. 1 is most commonly found in existing works [7], [10], [13], [27]. For the ease of discussion and without loss of generality, in the next section, we will first present analysis for this model. Later, we will develop it for the more general $k M \times k M$ multiplier, shown in Fig. 3.

In this paper, we will apply the proposed analysis on following two types of building blocks found in literature:

- $2 \times 2$ approximate multipliers.
- $4 \times 4$ multipliers utilizing approximate $4 : 2$ compressors.

The $2 \times 2$ building blocks based approximate multipliers belong to the Type II (Section 2) while those constructed from $4 \times 4$ units can be considered as a hybrid of Types I and II, as they involve both the approximate compressors and approximate partial products generation. Therefore, considering the model in Fig. 1 with $M = 2$ and $M = 4$ will cover a wide variety of approximate multipliers.

It is important to note that since this addition is precise, the structure of the adder tree does not affect the error statistics; rather the errors in all the partial products will be simply added to give the cumulative error. Extension of the proposed analysis for hybrid multipliers, with approximations in the partial product accumulation, will be discussed later in Section 4.6.1.

4 ANALYSIS OF ERROR PROBABILITY

Consider the $N \times N$ multiplier in Fig. 1. It takes two $N$-bit inputs $A$ and $B$ and produces $2N$-bit output $A \times B$, where $\times$ represents approximate multiplication. The probability of error in the multiplier output is defined as follows:

$$
Pr[\text{Error}] = Pr[A \times B \neq A \times B]
$$

In Section 4.2, for the ease of discussion, we will first derive the probability of error for the model in Fig. 1, in which $N = 2^k M$. However, in a more general multiplier based on $M \times M$ units, $N = k M$. Therefore, by building further upon the analysis in Section 4.2, we will analyze $Pr[\text{Error}]$ in the general $k M \times k M$ multiplier case in Section 4.3.

4.1 Assumptions on Inputs and Multiplier Architecture

In order to simplify and generalize the analysis, we make following assumptions:

- The inputs $A$ and $B$ are independent. In Sections 4.2 and 4.3, we first present the analysis for uniformly distributed inputs. In Section 4.4, the analysis will be developed further for a general probability distribution of the inputs.
- For a given design, all the partial products are generated using identical approximate multipliers.
- Multiplier building blocks are constructed in such a way that occurrence of error is identically related to both the inputs, i.e., $A = B = B \times A$.

Note that these assumptions are made after carefully studying the available multipliers of this type. These multipliers [7], [10], [13], [14], [27] provide a wide range of operating conditions in terms of speed, area, power and error statistics [10], which means that they can be considered as a major class of approximate multipliers.
4.2 Probability of Error in a $2^k \times M \times 2^k \times M$ Multiplier

We first consider the most commonly found multiplier architecture based on the model in Fig. 1. In this case, the bit-width $N$ is $2^k$ times $M$. Let $E_1$, $E_2$, $E_3$, and $E_4$ be the random variables (RVs) representing the binary events associated with the occurrence of error in $N$-bit partial products $PP_1$, $PP_2$, $PP_3$, and $PP_4$, respectively, as shown in Fig. 2. The RV $E_1$ is $1$ if there is error in the partial product $PP_1$, and $E_2 = 0$ otherwise. Error in the $N \times N$ multiplier occurs when there is an error in any of the four partial products, except the cases when simultaneous errors in multiple partial products cancel one another to result in zero error. Let $\kappa$ be the probability of occurrence of all $N$-bit input combinations that result in zero error due to cancellation, even though some individual partial products are erroneous. The probability of error $P[E]$ in an $N \times N$ approximate multiplier, constructed from $N/2 \times N/2$ building blocks, is represented as follows:

$$P(E) = \sum_{N/2} \rho(N,M)$$

(2)

Now, we discuss the evaluation of two parts of the above equation.

4.2.1 Evaluation of $\rho(N,M)$

Consider the probability of error in any one of the partial products in Fig. 2. For example,

$$Pr[E_i] = Pr[A_1 \times B_i \neq A_1 \times B_i] = \rho(N/2, M)$$

(3)

Note that the cancellation cases will be dealt with by using $\kappa$ at the end, so we are using $Pr[E_i] = \rho(N/2, M)$. Since $A_1$ and $B_1$ are independent, they make equivalent contributions to the occurrence of error $E_1$, i.e., both $A_1$ and $B_1$ must simultaneously satisfy the conditions required for the occurrence of error. Mathematically, this can be stated as follows: Let $\Theta$ be the set of all $N/2$-bit values such that the product $A_1 \times B_1$ is erroneous if $A_1 \in \Theta$ and $B_1 \in \Theta$, i.e.,

$$Pr[E_1] = Pr[A_1 \in \Theta \land B_1 \in \Theta] = Pr[A_1 \in \Theta] Pr[B_1 \in \Theta]$$

(4)

From Eq. (3) and Eq. (4), we can write

$$Pr[E_1] = \rho(N/2, M) = \sqrt{\rho(N/2, M) \cdot \rho(N/2, M)}$$

(5)

In order to explain the modeling in Eq. 4, we consider the following examples: in Kulkarni’s multiplier [13], Rehman’s [10] and Lin’s [28] multipliers discussed above. For others [7, 11], it can be considered to be a good approximation, as it represents most of the error cases. For example, in Shafiique’s multiplier [7], we can only define an approximate set $\Theta$, i.e., $\Theta = \{112, 012\}$. This definition represents most of the error cases for this multiplier (see Fig. 5b). As a result, for this type of multipliers, Eq. 4 will lead to an estimated error probability, which can be considered a tight lower bound, as evident from the results in Fig. 7.

Now consider Fig. 2. We can see that every $N/2$-bit component of the first factor $(A_2, A_3)$ is multiplied by every $N/2$-bit component of the second factor $(B_2, B_3)$. Therefore, error will occur if at least one of the $N/2$-bit components of both the factors simultaneously satisfy the condition for the occurrence of error, i.e.,

$$\rho(N,M) = Pr[E_1 = 0 \land E_2 = 0 \land E_3 = 0 \land E_4 = 0]$$

(6)

The expression in Eq. (7) is a recursive one. As a base case, the probability of error $\rho_{M}$ in the $M \times M$ basic building block will be used, which is known from its truth table.

4.2.2 Evaluation of $\kappa$

The error cancellation in two partial products leads to zero error if the two errors have the same magnitude and opposite polarities and there is no error in other partial products. Considering the existing designs for this class of multipliers, following observations are made:

- In most designs, errors in $M \times M$ units always have the same polarity [10], [11], [13], [27]. In these cases, it is straightforward to see that $\kappa = 0$, as there are no cancellations.
- Consider the designs where errors can possibly have opposite polarity [14]. For a multiplier with four partial products, as shown in Fig. 2, cancellation will lead to zero cumulative error if $PP_2$ and $PP_3$ have errors of equal magnitude and opposite signs and there is no error in $PP_1$ and $PP_4$. Therefore, $E_2 \land E_3 \rightarrow A_1, B_2, A_2, B_1 \in \Theta \rightarrow E_4 \land E_5$. Therefore, $Pr[E_1 = 0 \land E_2 = 0 \land E_3 = 0 \land E_4 = 0] = 0$, which implies $\kappa = 0$.

Note that in this work, our aim is to develop a generic analysis for this type of multipliers, that is simple enough for fast application. The available multipliers have been shown to cover a wide design space with respect to speed, power, area and error characteristics, so the analysis has been developed by keeping their common characteristics in view. Theoretically, it may be possible to have more complex functional models. However, we do not find any such models with structural (speed/area) benefits in the existing literature. Therefore, for the existing models, we will use $\kappa = 0$, which is demonstrated to give fairly accurate results in Section 5.

4.3 Probability of Error in $kM \times kM$ Multipliers

In general, a $kM$-bit ($k = 1, 2, 3, \ldots$) multiplier can be constructed by adding $(N/M)^2$ partial products, each generated by an $M \times M$ approximate multiplier unit, as shown in Fig. 3. In order to evaluate
the probability of error $p(N, M)$ in this case, we use the same approach as in Section 4.2. Since error in any of the $(N/M)^2$ partial products can lead to error, $p(N, M)$ is defined as follows:

$$p(N, M) = \Pr \left[ E_1 \lor E_2 \lor \ldots \lor E_{(N/M)^2} \right]$$

(8)

Let $A_1, A_2, \ldots, A_{N/M}$ and $B_1, B_2, \ldots, B_{N/M}$ be the $M$-bit components of the multiplier inputs $A$ and $B$, respectively. Since at least one of the $M$-bit components of each multiplier inputs must simultaneously satisfy the condition for the occurrence of error, $p(N, M)$ is defined as follows:

$$p(N, M) = \Pr \left[ \left( A_1 \in \Theta \right) \lor \left( A_2 \in \Theta \right) \lor \ldots \lor \left( A_{N/M} \in \Theta \right) \right] \land \left( B_1 \in \Theta \right) \lor \left( B_2 \in \Theta \right) \lor \ldots \lor \left( B_{N/M} \in \Theta \right)$$

(9)

In this case, $\Theta$ is the set of all $M$-bit values of the $M \times M$ multiplier inputs that lead to the occurrence of error. Therefore, following the same reasoning used in Section 4.2, $\Pr[A_1 \in \Theta] = \Pr[A_2 \in \Theta] = \ldots = \Pr[A_{N/M} \in \Theta] = \Pr[B_1 \in \Theta] = \Pr[B_2 \in \Theta] = \ldots = \Pr[B_{N/M} \in \Theta] = \sqrt{p_{\text{err}}}$ where $p_{\text{err}}$ is known from the truth table of $M \times M$ approximate multiplier unit. By using the inclusion-exclusion principle [29], $p(N, M)$ is found as follows:

$$p(N, M) = \left( \sum_{i=1}^{N/M} \Pr \left[ A_i \in \Theta \right] - \sum_{1 \leq i < j \leq N/M} \Pr \left[ (A_i \in \Theta) \land (A_j \in \Theta) \right] \right) \times \left( \sum_{i=1}^{N/M} \Pr \left[ B_i \in \Theta \right] - \sum_{1 \leq i < j \leq N/M} \Pr \left[ (B_i \in \Theta) \land (B_j \in \Theta) \right] \right) \ldots$$

(10)

$$+ (-1)^{N/M-1} \Pr \left[ \bigwedge_{1 \leq i \leq N/M} \left( A_i \in \Theta \right) \right] \times \left( \sum_{i=1}^{N/M} \Pr \left[ B_i \in \Theta \right] - \sum_{1 \leq i < j \leq N/M} \Pr \left[ (B_i \in \Theta) \land (B_j \in \Theta) \right] \right)$$

If $A$ and $B$ are uniformly distributed, then all the individual bits of $A$ and $B$ are independent and uniformly distributed. Therefore, the joint probabilities in Eq. (10) are equal to the products of individual events. Consequently, Eq. (10) is evaluated as follows:

$$p(N, M) = \left( \sum_{k=1}^{N/M} \left( -1 \right)^{k+1} \left( \frac{N}{M} \right)_k \left( \sqrt{p_{\text{err}}} \right)^k \right)^2$$

(11)

where $\rho_\Theta$ is the probability of error in $M \times M$ multiplier and $\left( \frac{N}{M} \right)_k$ is the binomial coefficient. Since $2^M$ $M$-bit multipliers are special cases of $k$-bit multipliers, so Eq. (11) is applicable to all cases.

### 4.4 Error Probability Analysis for a Given Input Distribution

If the multiplier inputs $A$ and $B$ are uniformly distributed, then every bit in the binary representation of $A$ and $B$ is $0$ or $1$ with equal probability. This property does not hold if the inputs have a non-uniform distribution. This means that the probabilities $\Pr[A_k \in \Theta]$ and $\Pr[B_k \in \Theta]$ may not be equal for all $1 \leq k \leq N/M$.

Let the $M$-bit multiplier inputs $A$ and $B$ be distributed according to PMFs $p_A(x)$ and $p_B(x)$, respectively. Let $\Pr[A_k \in \Theta] = \delta_A k$ and $\Pr[B_k \in \Theta] = \delta_B k$, for $1 \leq k \leq N/M$. In order to find $\delta_A k$, we need to sum $p_A(x)$ over all values of $A$ whose binary representation have a sequence in $\Theta$ at the $k$th $M$-bit component. Essentially, we need to derive the probabilities of certain bit sequences from the PMF of the decimal representation of inputs. For example, consider a 6-bit binary input $A = [a_5, a_4, a_3, a_2, a_1, a_0]$. The PMF $p_A(x)$ gives the distribution of decimal values of $A$. For $M = 2$ and $\Theta = \{112\}$, it is straightforward to see that $\delta_{A,2} = \Pr[A_2 \in \Theta] = \Pr[[a_5, a_2, a_2] = 112]$ can be found as follows:

$$p_A[a_5, a_4, a_3, a_2, a_1, a_0] = \sum_{h=0}^{2^M-1} p_A(2^h a_5 + 2^{h+1} a_4 + 2^{h+2} a_3 + 2^{h+3} a_2 + 2^{h+4} a_1 + 2^{h+5} a_0)$$

(12)

Eq. (12) basically permanently assigns the bits $[a_5, a_2, a_2] = 112$ and sums the PMF $p_A(.)$ over the decimal equivalents of all the possible combinations of $[a_5, a_4, a_3, a_1, a_0]$. This relationship is generalized as follows:

$$\delta_{A,k} = \sum_{y_0 \in \Theta} \sum_{h=0}^{2^{N-k} \cdot M-1} \sum_{h=0}^{2^{N-k} \cdot M-1} p_A(2^h y_0 + 2^{h+1} M y_{10} + i)$$

(13)

where $y_{10}$ is the decimal equivalent of the sequence $y \in \Theta$. Similarly, $\delta_{B,k}$ can be found from $p_B(.)$. Joint probabilities, like $\Pr[A_1 \in \Theta \land A_2 \in \Theta]$, can be found by permanently assigning $[a_5, a_2, a_1, a_0] = 11112$ and following the same procedure. Although the joint probabilities in the inclusion-exclusion principle in Eq. (10) can all be separately evaluated, here we estimate the events $A_1, A_2 \in \Theta$, for $1 \leq k \leq N/M$, as independent to simplify the analysis.

Therefore, the joint probabilities are expressed as products of $\delta_{A,k}$, for $1 \leq k \leq N/M$. Similar estimation is also made for input $B$. Therefore, using $\delta_{A,k}$ and $\delta_{B,k}$, probability of error $p(N, M)$ is found by using Eq. (10) as follows:

$$p(N, M) = \left( \sum_{i=1}^{N/M} \delta_{A,i} - \sum_{1 \leq i < j \leq N/M} \delta_{A,i} \delta_{A,j} \right) \ldots$$

(14)

$$+ \left( \sum_{i=1}^{N/M} \delta_{B,i} - \sum_{1 \leq i < j \leq N/M} \delta_{B,i} \delta_{B,j} \right) \ldots$$

In case of uniformly distributed inputs, Eq. (2) can be simplified to get Eq. (11) by setting $\delta_{A,k} = \delta_{B,k} = \sqrt{p_{\text{err}}}$, for all $1 \leq k \leq N/M$.

### 4.5 Probability Distribution of Error

In Section 4, we found the overall probability of error in an approximate multiplier 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 conditions on the factors of partial products. In every case, error value is found by identifying the partial products with erroneous outputs, along with their weights, which are determined by their respective left shifts. The corresponding probabilities are computed using the same concepts that we developed in Sections 4.1–4.4.

The error in the product $A \times B$ is the sum of errors in all the $(N/M)^2$ partial products. It can be observed from Figs. 1, 2 and 3 that the partial products have common factors, which means that they are highly interdependent. As a result, general and accurate analysis of PMF of error value is a very difficult problem. Here, we first consider an accurate method for $M \times M$ multipliers with only one possible error value (Case I). For more complex models, a generic method for PMF estimation is presented (Case II). This means that Case II can cover all the multipliers and Case I is an accurate method for some special cases.

#### 4.5.1 Case I

In this case, the error in $M \times M$ multiplier unit can only attain a value $V_M$. The PMF is computed using Algorithm 1, which exhaustively computes the probability of all the possible combinations for which the condition in Eq. (9) is true.
Algorithm 1 PMF of Approximation Error

1: Input $N, M, \delta A_1, \ldots, \delta A_{N/M}, \delta B_1, \ldots, \delta B_{N/M}$ and $V_M$.
2: Input $X = (x, y) : x \in F_A(x) \land y \in F_B(y) \land x \neq y$.
4: Define the set of factor indices $F = \{1, 2, \ldots, N/M\}$.
5: Find the power set $P_F = \mathcal{P}(F)$.
6: for every $i^{th}$ element $F_B \in P_F$ do
7: for every $j^{th}$ element $F_B \in P_F$ do
8: Find the set of pairs $S = \{(x, y) : x \in F_A(x) \land y \in F_B(y)\}$.
9: Find error value $v_e = \sum_{(x, y) \in S} \delta A \cdot \delta B \cdot \delta (x+y-2M)$.
10: Find the probability of error value $p_e = \prod_{m \in F_A} (1 - \delta A_m) \prod_{n \in F_B} (1 - \delta B_n)$
11: if $v_e \in v$ then
12: Update array value $p_{\nu} (v_e) = p_{\nu} (v_e) + p_e$.
13: else
14: Append $v_e$ to array $v$ and $p_{\nu}$ to array $p(v)$.
15: end if
16: end for
17: end for
18: Append 0 to array $v$ and $(1 - \sum p(v))$ to array $p(v)$.
19: Output arrays $v$ and $p(v)$.

with $N = 4, M = 2$, $(A_1 \in \Theta) \land (A_2 \in \Theta)$ implies one of the three mutually exclusive events: $(A_1 \in \Theta) \land (A_2 \notin \Theta)$, $(A_1 \notin \Theta) \land (A_2 \in \Theta)$ or $(A_1 \notin \Theta) \land (A_2 \notin \Theta)$. Similarly, conditions on input $B$ will also be imposed simultaneously. The probability of an example of these jointly occurring events is given below:

$$P \left[ \left( A_1 \in \Theta \right) \land \left( A_2 \notin \Theta \right) \right] \times P \left[ \left( B_1 \in \Theta \right) \land \left( B_2 \notin \Theta \right) \right] = \delta A_1(1 - \delta A_2)\delta B_2(1 - \delta B_1)$$

The error in this example will occur in the partial product $A_1 \times B_2$. If the magnitude of error in the $M \times M$ unit is $V_M$, then the magnitude of error in this left-shifted partial product is $\rho^M V_M$. Algorithm 1 finds the PMF of error by considering every possible mutually exclusive event that results in error in the approximate adder and computing its probability and its magnitude. The outputs include an array of error magnitudes and an array containing the corresponding probabilities. It is also equipped to find the PMF of multipliers built from a combination of approximate and precise multiplier units. This can be accomplished by using the input $X$ in the proposed algorithm, that includes all the pairs $(A_i, B_j)$ that are multiplied using precise multipliers. Note that this method will compute the exact PMF for multipliers satisfying the assumptions in Section 4.1 and an estimated PMF otherwise.

4.5.2 Case II

Generally, the error values in $N \times M$ multipliers may differ for different inputs with PMF $p_{PMF}(v)$, where $p_{PMF}$ is the conditional PMF of error given that it is non-zero. The PMF of error is $N \times N$ multiplier can be estimated by using the law of total probability. This involves adding the PMFs of error for the two mutually exclusive classes as described above. For example, the PMF of error for the condition in Eq. (15) is given as follows:

$$p_{\nu} (v | A_1, B_2 \in \Theta, A_2, B_1 \notin \Theta) \approx p_{\nu} \left( \frac{v}{2^{M}} \right) \delta A_1(1 - \delta A_2) \delta B_2(1 - \delta B_1)$$

In cases where more than one partial product is erroneous, the PMF can be estimated by convolving the PMFs of individual partial products. For example,

$$p_{\nu} (v | A_1, A_2, B_2 \in \Theta, B_1 \notin \Theta) \approx p_{\nu} \left( \frac{v}{2^{M}} \right) \times p_{\nu} \left( \frac{v}{2^{M}} \right) \delta A_1 \delta A_2 \delta B_2(1 - \delta B_1)$$

Note that since the partial products are not independent, convolution leads to an approximate result. The PMF of error for the $N \times N$ multiplier will be found by adding the PMFs similar to those in Eqs. (15) and (17) for all the possible conditions on $A$ and $B$.

4.6 Towards Generalization of the Analysis

4.6.1 Approximations in Partial Product Tree

The class of multipliers under consideration can be used in hybrid structures where approximate Full Adders (FAs) or compressors are used in a Wallace tree to add the approximate partial products [10]. For the error analysis of a Wallace tree multiplier using approximate FAs, an estimation based method is presented in [25]. Since the errors from all the $M \times M$ units are additive and the errors from FAs are also additive, and the two approximations are done independent of each other, the PMF analysis for this general case can be done by convolving the PMF from the proposed analysis (representing errors in partial products) with the PMF obtained from the method in [25] (representing errors in Wallace tree structure). This will also require an evaluation of the probability distributions of all the bits of partial products that are input to the FAs. This distribution can be computed by using the truth table of $M \times M$ units, along with input distributions. Similar extensions of the proposed analysis can also be done for multipliers using other types of approximate adders for partial product summation.

4.6.2 Signed Multipliers

A signed multiplier can be designed as a modification of the general model in Fig. 3 by using the sign extension elimination method [30, Section 5.9.2]. By carefully observing the changes in functional model, we see that for some partial products in a signed multiplier, the logic designed for unsigned multipliers can be reused. For other partial products, we can either use precise logic circuits or design suitable approximate $M \times M$ modules. Fig. 4 illustrates the similar and different parts of signed and unsigned multipliers for a $8 \times 8$ multiplier, considering $2 \times 2$ approximate partial product generators. As explained in Section 4.2, Algorithm 1 is capable of analyzing hybrid multipliers by using the optional input set $X$. Therefore, we will consider similar hybrid structures for signed multiplication in the proposed analysis.

4.6.3 Squarer

It is straightforward to modify the proposed analysis for this case by setting $A = B$ in Eq. (9). The probability of error in an $N \times N$ squarer, constructed from $M \times M$ approximate multipliers comes out to be as follows:

$$P_{error} (N, M) = \sum_{k=1}^{N/M} (-1)^{k+1} \left( \frac{n}{k} \right) \rho^{k} = \sqrt{\rho(N, M)}$$

4.6.4 Multiplication with a Constant

If an $N$-bit random variable $A$ is multiplied by a constant $B$, then $P_{error} (B_1 \in \Theta) \lor (B_2 \in \Theta) \lor \ldots \lor (B_{N/M} \in \Theta)$ in Eq. (9) is either 1 or 0, depending upon the value of the constant $B$. Let $\xi = \{X | X_1 \in \Theta \lor \ldots \lor X_{N/M} \in \Theta\}$ be the set containing all the constants that lead to error in $A \times B$ for a given $N \times N$ approximate multiplier. Therefore, the probability of error in $A \times B$ is given as follows:

$$P_{error} (N, \xi) = \begin{cases} \sqrt{\rho(N, M)}, & \text{if } B \in \xi \\ 0, & \text{otherwise} \end{cases}$$
The PMF of error can be computed by using Algorithm 1 with inputs $\delta_{B,K} = 0$ or 1. For example, if $B = 11000110111_2$ and $\Theta = \{112\}_2$, then $\delta_{B,6}, \delta_{B,1} = 1, \delta_{B,5}, \delta_{B,4}, \delta_{B,3}, \delta_{B,2} = 0$. This is useful in DSP applications, where the most common arithmetic operation is the evaluation of the sum of products, using constant multipliers such as those in digital filters.

## 5 Results and Discussions

In this section, we apply our analysis to several state-of-the-art approximate multipliers. The results of the analysis are validated by comparing them with those obtained from simulation of these multipliers. For small-sized multipliers, we have run exhaustive simulations by evaluating the multiplier for all possible inputs. For larger multipliers, Monte-Carlo (MC) simulations are used.

### 5.1 Multipliers used for Evaluation

Fig. 5 shows two instances of $2 \times 2$ approximate multiplier building blocks, proposed by Kulkarni et al. [13] and Shafique et al. [7], along with their K-Maps and $\rho_{ST}, VM$ values. Other $2 \times 2$ multipliers are proposed by Rehman et al. [10]. We have selected ApproxMul$_{1}$, ApproxMul$_{4}$, and ApproxMul$_{6}$, for evaluation. Fig. 6(a) shows a $4 \times 4$ approximate multiplier constructed using a $4 \times 2$ compressor. Two $4 \times 2$ compressors, proposed by Lin et al. [27] and Momeni et al. [14] are shown in Figs. 6(b) and 6(c) respectively, along with their $PM$ and $VM$ values. In addition, we will present results with $4 \times 2$ compressors given in [12]. Four designs from [12] are selected, namely the High Accuracy Compressors (HAC1 and HAC2) and Medium Accuracy Compressors (MAC1 and MAC2). Note that HAC1 [12] and Lin’s compressor have the same functional model with different structural implementations, which means their error statistics are the same.

### 5.2 An Overview of Results

Table 1 shows a summary of results for various configurations of all the eleven multipliers considered in evaluation. Error Rate (ER) is the found as $\rho(N, M)$ and MED is found from the PMF as $\sum_{v \in V} |v| p_{V}(v)$ where $V$ is all of the elements. All the multipliers are evaluated for uniform, geometric, and/or Gaussian distributions. 2. We observe that:

- The analysis results match well with MC simulations in case of Kulkarni’s, Rehman’s (ApproxMul$_{4}$ and ApproxMul$_{6}$), Ma’s (HAC1 and HAC2) and Lin’s multipliers.

- In case of Shafique’s, Ma’s (MAC1 and MAC2) and Momeni’s multipliers, the analysis results are close to MC simulation results. These results are explored in detail in the next sections.

### 5.3 Error Probability Results

Figs. 7 shows the comparison of analysis and simulation results for the four multipliers under consideration with uniformly distributed inputs. It can be clearly seen that the simulation results are in good agreement with the proposed analysis. In case of Kulkarni’s multiplier in Fig. 7a and Lin’s multiplier in Fig. 7b, the results of analysis and those obtained from exhaustive simulations are a perfect match. However, in case of Shafique’s $2 \times 2$ and Momeni’s $4 \times 4$ multipliers, the analytical and simulation results show slight differences. This is because of the assumptions made in Section 4 to condition the distribution, i.e., $p_{N}(z) = p_{M}(z) = p_{V}(z)$ (Lin’s architecture) exactly satisfy these assumptions, we obtain accurate probability of error. But these assumptions do not completely reflect the behaviors of Shafique’s and Momeni’s multipliers; rather in these cases, they are approximations. However, it is evident from the results that the error incurred in the probability of error analysis is very small. This is due to the fact that the assumptions are true for most of the error cases for these multipliers. Therefore, our proposed analysis can estimate the probability of error with reasonable accuracy. Results for non-uniform distributions are given in Table 1.

### 5.4 Error PMF Results

The PMFs of approximation error are computed for Kulkarni’s ($M = 2, \rho_M = 1/16, V_M = 2$), Lin’s ($M = 4, \rho_M = 1/256, V_M = 16$), Rehman’s ApproxMul$_{4}$ ($M = 2, \rho_M = 1/16, V_M = 8$) and Shafique’s ($M = 2, \rho_M = 3/16, V_M = 1$) multipliers using Algorithm 1 and for Momeni’s ($M = 4, \rho_M = 100/256$, $V_M$) multiplier.

2. Standard geometric RVs can be infinite valued. However, since $N$-bit inputs are limited between 0 and $2^{N-1} - 1$, the distributions used here are truncated to the range $[0, 2^{N-1} - 1]$. 3. Geometric RVs are generated by summing several uniform RVs, and Gaussian PMF is the convolution of these uniform PMFs.
using the method based on the law of total probability, as described in Section 4.5.2. The results of the proposed analysis are compared with those obtained from simulation of the respective multipliers. Histograms are used to compute the PMFs from simulation results.

It can be seen from Fig. 8 that the analysis and simulation results match perfectly for Kulkarni’s and Lin’s multipliers. In case of Shafique’s and Momeni’s multiplier, we observe approximation error in the computed PMFs. This is again because of the assumptions made to simplify the analysis in Section 4. The effect of these assumptions is clearly understood by closely observing the results for Shafique’s multiplier in Figs. 8c,d. We see that the larger probabilities are computed fairly accurately, while error values with very small probabilities are neglected by the proposed algorithm. Since these assumptions represent most but not all of the error cases, the estimated probabilities of the infrequently occurring error values are set to zero by Algorithm 1. Similar type of approximation error in the computed PMF is also observed in case of Momeni’s multiplier, shown in Figs. 8h. This is due to the convolution operation, which ignores the interdependence of partial products. The PMF results with geometric and Gaussian distributions are shown in Fig. 9, where similar trends can be observed.

5.5 Results for Hybrid Multipliers

Algorithm 1 is equipped to compute PMF for multipliers that are constructed from a combination of precise and approximate multiplier units. Fig. 10a shows the results for $8 \times 8$ multipliers constructed from a combination of Kulkarni’s and precise $2 \times 2$ multipliers. It can be seen that the probability of error and maximum error value is decreased because the more significant partial products are evaluated using precise components. This type of analysis is important because in many multipliers, approximation is only done for bits with lower significance. This information is also important in the design of accuracy configurable multipliers, as it can predict the redistribution of error value as a result of partial error correction. Recently, in [31], Consolidated Error Correction (CEC) was proposed for circuits with multiple adders. Using the same concept, these error distributions can be utilized to optimize error recovery schemes.

As explained in Section 4.6.2, hybrid multipliers can also be used to design signed multipliers, as illustrated in Fig. 4. Here, we evaluate $8 \times 8$ signed multipliers that are designed by employing $2 \times 2$ Kulkarni’s multipliers for the partial products that are same as in an unsigned multiplier and precise $2 \times 2$ multipliers for the rest of partial products. Results are shown in Fig. 10b.

5.6 Application: Blending of Images

In this section, we demonstrate the usefulness of the proposed analysis in some practical signal processing applications. Blending two or more images is an application of image editing used to create special effects. The Multiply Mode is one of the commonly used modes of blending, in which two images are multiplied pixel-by-pixel. Fig. 11 shows the results obtained by blending two images using fixed-point precise and approximate multipliers. Since the inputs to the multipliers are coming from two different images, they can be considered as independent and random. The difference...
images show that the multiplication results are not 100% accurate. Conventionally, this inaccuracy is quantified by a number of metrics, including error rate (ER), mean error distance (MED), average normalized error distance (NED), means square error (MSE), and peak signal-to-noise ratio (PSNR). Conventionally, they are evaluated by simulating image processing applications, which not only requires considerable programming effort but can also yield significantly different results for different images. The proposed analysis is useful in evaluating average performance of an approximate multiplier. Table 2 shows a comparison of the metrics evaluated from the images and the proposed analysis. The results show that the proposed analysis predicts the multipliers’ performance with reasonable accuracy. It should be noted that since we aim to evaluate the average performance, the analysis in this case has been performed by assuming uniformly distributed inputs.

6 Conclusion
In this paper, we have presented probabilistic analysis of approximation error in a class of low-power approximate multipliers. The probability of error and PMF of error are analyzed for arbitrarily length multipliers for a general input distribution. The results of the analysis are found to be in good agreement with those obtained from simulations. The proposed analysis is also tailored for square constant multipliers and hybrids of precise and approximate multipliers. Some interesting observations are made regarding the relationship between the functional models of approximate multiplier units and the input distribution, which can serve as a design guideline towards building optimal low-power multipliers. The proposed analysis is also used to compute the performance metrics that are conventionally used to predict and compare multipliers’ performance in practical image processing applications.

References