An analytical framework for error modeling of approximate adders

Zhixi Yang\textsuperscript{a)}, Jun Yang, Kefei Xing, and Guang Yang

College of Mechantronics Engineering and Automation, National University of Defense Technology, Changsha 410073, People’s Republic of China

\textsuperscript{a}) nudtyzx@163.com

Abstract: Approximate computing has drawn recent attention as it can reduce power without significantly sacrificing the output quality for arithmetic operations in error-resilient applications. However, the error characteristics evaluation often depends on time consuming random simulation and estimation. In this paper, a formal framework of evaluating approximate adders is proposed based on a theoretical model. The proposed framework can accurately measure mean squared error (MSE) and mean absolute error (MAE) without using functional simulations. An approach has been adopted to estimate the final PSNR value in terms of MSE and MAE for DCT-IDCT image compression system. Experimental results have validated the accuracy and usefulness of the framework.

Keywords: approximate computing, error modeling and analysis, approximate adder, squared error, absolute error

Classification: Integrated circuits

References

[1] M. Samadi, \textit{et al.}: “Sage: Self-tuning approximation for graphics engines,” Proc. of the IEEE/ACM Int. Symp. on Microarchit. (2013) 13 (DOI: 10.1145/2540708.2540711).

[2] N. Zhu, \textit{et al.}: “Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. \textbf{18} (2010) 1225 (DOI: 10.1109/TVLSI.2009.2020591).

[3] A. B. Kahng and S. Kang: “Accuracy-configurable adder for approximate arithmetic designs,” Proc. of IEEE Design, Automation Conf. (DAC) (2012) 820 (DOI: 10.1145/2228360.2228509).

[4] D. Mohapatra, \textit{et al.}: “Design of voltage-scalable meta-functions for approximate computing,” Proc. of Design, Automation and Test in Europe Conference and Exhibition (DATE) (2011) 1 (DOI: 10.1109/DATA.2011.5763154).

[5] A. K. Verma, \textit{et al.}: “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” DATE (2008) 1250 (DOI: 10.1145/1403375.1403679).

[6] V. Gupta, \textit{et al.}: “Low-power digital signal processing using approximate adders,” IEEE Trans. CAD \textbf{32} (2013) 124 (DOI: 10.1109/TCAD.2012.2217962).

[7] V. Gupta, \textit{et al.}: “Impact: Imprecise adders for low-power approximate
computing.” Int. Symp. on Low Power Elec. and Designs (ISLPED) (2011) 409 (DOI: 10.1109/ISLPED.2011.5993675).
[8] H. R. Mahdiani, et al.: “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. 57 (2010) 850 (DOI: 10.1109/TCSI.2009.2027626).
[9] Z. Yang, et al.: “Transmission gate based approximate adders for inexact computing,” the IEEE/ACM Int. Symp. on Nanoscale Architectures (2015) 145 (DOI: 10.1109/NANOARCH.2015.7180603).
[10] C. Liu, et al.: “An analytical framework for evaluating the error characteristics of approximate adders,” IEEE Trans. Comput. 64 (2015) 1268 (DOI: 10.1109/TC.2014.2317180).
[11] C. Li, et al.: “Joint precision optimization and high level synthesis for approximate computing,” DAC (2015) 1 (DOI: 10.1145/2744769.2744863).
[12] L. Li and H. Zhou: “On error modeling and analysis of approximate adders,” Proc. of the IEEE/ACM Int. Conf. on Comp. Aided Des. (ICCAD) (2014) 511.
[13] C. Yu and M. Ciesielski: “A case study of analyzing imprecise adders using BDDs,” accepted by IEEE Annual Symp. on VLSI (2016).

1 Introduction

Approximate computing has become a promising technique to achieve large efficiencies in VLSI designs by sacrificing computational accuracy. The designs have proved to significantly save energy for error-tolerant applications, such as neural networks, multimedia, mining and recognition [1] while ensuring the errors are limited to a tolerable level.

As one of the fundamental components of arithmetic circuits, adders have attracted interest for approximate schemes. The approximate adders could be broadly categorized into two types. The first type is due to the impact of timing-induced errors when a n bit ripple carry adder (RCA) is divided into several k-bit sub-adders (k < n), e.g., [2, 3, 4, 5]. These designs cut the carry propagation chain to induce some errors to approximately calculate the results. Another type for reducing the critical path delay and power consumption is to approximate functional behavior by implementing a different Boolean function from a conventional adder, e.g., [6, 7, 8, 9].

A vital procedure for approximate computing is to fast and accurately evaluate the error into a computation and the evaluation often depends on Monte Carlo (MC) simulations. In the context of signal processing applications, relevant metric of assessing quality is quadratic error, e.g. SNR/PSNR, which is inversely related to mean squared error (MSE) and mean absolute error (MAE) [10]. To better understand the design prior to a simulation, mathematical models are required. Moreover, besides encompassing approximate arithmetic circuit, high level synthesis [11] has been an alternative method for approximate computing. Without of the aiding of theoretical error analysis, the error profiling or characterization [11] have to depend on simulations.

Hence a proper analytical model is essential to either guide early stage circuit design or give further insights for higher hierarchy, e.g. synthesis or architectural level. There are some progress in analyzing the error characteristics theoretically for
the first type adder (i.e., timing-induced approximation) as did in [10, 11, 12, 13], yet, the modeling for the second type still remains unsolved due to slightly complex error propagation mechanism from functional modification of carry out. The aim of this paper is to give a formal analytical approach for error in terms of MSE and MAE for arbitrary bit width RCA given arbitrary modified truth tables (Boolean functions). As far as we know, this is the first paper to show the error characterization for modified Boolean function based approximate adder. Then by adopting an analytical estimation between MSE, MAE and PSNR from [10], this paper could also estimate final PSNR of approximate adder based DCT-IDCT image compression system efficiently.

2 Error analysis and modeling

The paper aims to calculate MSE and MAE analytically for approximate addition shown in Fig. 1. The approximate addition uses same type of approximate adder (APA) for 1~kth bit. The major difficulty of APA based addition actually depends on fact that the error from lower bits would propagate to higher bits, e.g., if \( C_{1} \) is erroneous and used as an input for 2nd bit. Depending on this value, the output of 2nd bit would be either correct or not. The errors of adjacent bit positions are actually correlated generally, and the error would propagate and accumulate. If an arbitrary Boolean function is given for APA, the error metrics of interest would be hard to predict unless using MC simulations. This is the major reason that no progress has been made for this problem. There are some progress in bit-wise APA (i.e., using simple bit-wise logic operations), yet, no formal approach is given for both bitwise or non-bitwise APA. This section shows the detailed modeling for both types in terms of MSE and MAE.

2.1 General modeling

Consider a \( k \)-bit approximate RCA with inputs \( a, b \) first. Let \( a_i, b_i \) represent the input at \( i \)th position (starting from 1) and \( a = a_k a_{k-1} \ldots a_1, b = b_k b_{k-1} \ldots b_1 \). \( C_i \) is the carry out of \( i \)th bit. Assume the inputs are uniformly distributed and the total number of inputs is \( N = 4^k \). A \( n \)th input combination is marked as \( ab_n \) with \( a_{n,k} a_{n,k-1} \ldots a_{n,1} \) and \( b_{n,k} b_{n,k-1} \ldots b_{n,1} \), i.e.,

\[
ab_n = ab_{n,[k:1]} = a_{n,k}a_{n,k-1} \cdots a_{n,1} \\
b_{n,k}b_{n,k-1} \cdots b_{n,1}
\]  

Fig. 1. The type of approximate addition.
If the bit-width is increased by one, i.e., using \((k + 1)\)-bit for addition, one extra bit input combination for \((k + 1)\)th position should be added. This pair of input for \(a\) and \(b\) is called leading bits \(l_{k+1}\) for the specific input combination of \(n\) for lower \(k\) bits. Obviously, there are four combinations for \(l_{k+1}\), i.e., \(l_{k+1} = a_{k+1} \in L = \{0, 0, 1, 1\}\). Hence the combination of \((k + 1)\)-bit RCA can be denoted as:

\[
ab'_n = (l_{k+1}, ab_n) = a_{k+1}a_{n,k} \ldots a_{n,1} b_{k+1}b_{n,k} \ldots b_{n,1}
\]  

(2)

Generally, the result of addition with \(ab'_n\) can be expressed as:

\[
\begin{align*}
\sum_{i=1}^{k+1} 2^{i-1} S_i &= C_{k+1}^{\text{Sum}_{k+1}} \sum_{i=1}^{k} 2^{i-1} \sum_{j=1}^{i} \\
\sum_{i=1}^{k+1} 2^{i-1} S_i' &= C_{k+1}'^{\text{Sum}_{k+1}'} \sum_{i=1}^{k} 2^{i-1} \sum_{j=1}^{i} 
\end{align*}
\]

(3)

where \(\sum_{i=1}^{k+1} 2^{i-1} S_i\) and \(\sum_{i=1}^{k+1} 2^{i-1} S_i'\) means the result for accurate and approximate addition with corresponding input combination, respectively.

The \(i\)th bit accurate addition can be expressed in a decimal format:

\[
S_i = \sum_{i=1}^{k+1} 2^{i-1} S_i
\]

(4)

similarly, \(S_i'\) means the approximate result in decimal for \(i\)th bit.

If the addition is accurate, Eq. (3) can be written as:

\[
\sum_{i=1}^{k+1} 2^{i-1} S_i = C_{k+1}^{\text{Sum}_{k+1}} \sum_{i=1}^{k} 2^{i-1} \sum_{j=1}^{i} \\
\sum_{i=1}^{k+1} 2^{i-1} S_i' = C_{k+1}'^{\text{Sum}_{k+1}'} \sum_{i=1}^{k} 2^{i-1} \sum_{j=1}^{i}
\]

which means for the accurate addition, each bit position can be seen as independent by setting carry in as 0 and the final result is the summation of \(S_i\) with corresponding weight.

However, for the approximate adder based addition, Eq. (5) is no longer correct due to the carry out of a certain bit position could be erroneous. The incorrect carry signal could further induce error for the higher bits, hence adjacent bits become dependent. Yet, it can still be expressed as following:

\[
r_{ab_n}' = 2^k S_{k+1}^{\text{Sum}_{k+1}} + r_{ab_n}' + \delta
\]

(6)

where \(S_{k+1}^{\text{Sum}_{k+1}}\) means the decimal result for \((k + 1)\)th bit given the condition that the carry in (i.e., \(C_{k}'\)) is 0 with certain \(l_{k+1}\). \(\delta\) means the difference due to the flipping error of \(C_{k}'\) and is defined as:

\[
\delta = \begin{cases} 
0, & C_{k}' = 0 \\
2^k (d - 1), & C_{k}' = 1 
\end{cases}
\]

(7)

where \(d\) is defined as:

\[
d = S_{k+1}^{\text{Sum}_{k+1}} \quad C_{k+1}' = 0 - S_{k+1}^{\text{Sum}_{k+1}} \quad C_{k+1}' = 1.
\]

2.2 Error characteristics

According to the general models in Section 2.1, the error between approximate and accurate addition is obtained based on Eq. (5) and (6):

\[
e_{ab_n} = r_{ab_n}' - r_{ab_n} = 2^k S_{k+1}^{\text{Sum}_{k+1}} + r_{ab_n}' + \delta - 2^k S_{k+1} - r_{ab_n}
\]

(8)

where \(S_{k+1}\) actually equals to \(S_{k+1}^{\text{Sum}_{k+1}}\). Also \(r_{ab_n}' - r_{ab_n} = e_{ab_n}\). Hence Eq. (8) is written as:
\[ e_{ab_n} = 2^k(S'_{k+1}|l_{k+1}, c'_k = 0 - S_{k+1}|l_{k+1}, c_k = 0) + e_{ab_n} + \delta \]  

When \( C'_k = 0 \), Eq. (9) is written as:

\[ e_{ab'_n} = 2^k(S'_{k+1}|l_{k+1}, c'_k = 0 - S_{k+1}|l_{k+1}, c_k = 0) + e_{ab_n} \]  

When \( C'_k = 1 \), Eq. (9) is written as:

\[ e_{ab'_n} = 2^k(S'_{k+1}|l_{k+1}, c'_k = 0 - S_{k+1}|l_{k+1}, c_k = 0) + e_{ab_n} + \delta \]

\[ = 2^k(S'_{k+1}|l_{k+1}, c'_k = 1 - d) - (S_{k+1}|l_{k+1}, c_k = 1 - 1]) + e_{ab_n} + 2^k(d - 1) \]  

\[ = 2^k(S'_{k+1}|l_{k+1}, c'_k = 1) - S_{k+1}|l_{k+1}, c_k = 1) + e_{ab_n}. \]

From both Eq. (10) and (11), the relation between original error of \( k \)-bit addition with error of \( (k + 1) \)-bit addition can be easily understood, i.e., the error induced by adding an extra leading bit can be seen as independent from error induced by the rest of bits. Thus both can be simplified to:

\[ e_{ab'_n} = 2^k(S'_{k+1}|l_{k+1} - S_{k+1}|l_{k+1}) + e_{ab_n} = e_{l_{k+1}} + e_{ab_n} \]  

where \( e_{l_{k+1}} \) means the difference between approximate and accurate addition with identical \( l_{k+1} \) given same carry in, i.e., \( C_k = C'_k. \)

### 2.3 Squared error characteristics

According to Eq. (12), the squared error can be expressed as:

\[ E_{ab'_n} = (e_{l_{k+1}} + e_{ab_n})^2 = e_{l_{k+1}}^2 + e_{ab_n}^2 + 2e_{l_{k+1}}e_{ab_n} \]  

As shown in Section 2.1, the total number of inputs for \( ab_n \) is \( N = 4^k \) and by adding the leading bits the total number becomes \( N' = 4^{k+1} = 4N \) since there are four different combinations for \( l_{k+1} \). Hence the total squared error for \( (k + 1) \)-bit addition is:

\[ \sum_{n=1}^{N'} e_{ab'_n} = \sum_{n=1}^{N'} (e_{l_{k+1}}^2 + e_{ab_n}^2 + 2e_{l_{k+1}}e_{ab_n}) \]

\[ = 4 \sum_{n=1}^{N'} e_{ab_n}^2 + \sum_{n=1}^{N'} e_{l_{k+1}}^2 + 2 \sum_{n=1}^{N'} e_{l_{k+1}}e_{ab_n} \]  

Then the mean squared error (MSE) becomes:

\[ \frac{\sum_{n=1}^{N'} e_{ab'_n}^2}{N'} = \frac{4 \sum_{n=1}^{N'} e_{ab_n}^2}{N'} + \frac{\sum_{n=1}^{N'} e_{l_{k+1}}^2}{N'} + \frac{2 \sum_{n=1}^{N'} e_{l_{k+1}}e_{ab_n}}{N'} \]

\[ = \frac{4 \sum_{n=1}^{N'} e_{ab_n}^2}{4N} + \frac{\sum_{n=1}^{N'} e_{l_{k+1}}^2}{N'} + \frac{2 \sum_{n=1}^{N'} e_{l_{k+1}}e_{ab_n}}{N'} \]

\[ \Rightarrow \]

\[ MSE_{k+1} = MSE_k + M_1 + M_2 \]

As per Eq. (15), there is a recursive relation between the MSE of \( k \)-bit and \( (k + 1) \)-bit adder. As per Eq. (12), \( e_{l_{k+1}} = 2^k(S'_{k+1}|l_{k+1} - S_{k+1}|l_{k+1}) = 2^k\Delta S'_{k+1} \). Let \( \Delta S' \in [-3, 3] \) and \( \Delta S'_{k+1} \in \Delta S' \) and \( \Delta S'_{k+1} \in \mathbb{Z} \).
We now show how to calculate $M_1$. First, the probability of $\Delta S'_{k+1}$ equals to a certain value in $\Delta S'$ is expressed as:

$$p_{\Delta S'_{k+1}} = \sum_{j=0}^{1} p(C'_k = s) p(\Delta S'_{k+1} | s)$$  \hspace{1cm} (16)$$

where $p(\Delta S'_{k+1} | s)$ means the probability of $\Delta S'_{k+1}$ equals to $\Delta S'$ given $s$ as carry in. Then $M_1$ can be written as:

$$M_1 = \sum_{\Delta S'_{k+1} = -3}^{3} p_{\Delta S'_{k+1}} 4^k \Delta S'^2_{k+1}$$  \hspace{1cm} (17)$$

where $p(C'_k = s)$ can be calculated recursively as:

$$p(C'_k = s) = \sum_{j=0}^{1} p(C'_{k-1} = t) p(c'_j | t)$$  \hspace{1cm} (18)$$

where $p(c'_j | t)$ actually means the probability of carry out being $s$ given carry in $t$ for the $k$th bit approximate adder. The value can be obtained from a modified truth table directly.

**Theorem 1**: For a special case, if $C'_k(ab, s, C'_{k-1} = 0) = C'_k(ab, s, C'_{k-1} = 1)$, i.e., for the $k$th bit, the approximate carry out is same for identical inputs no matter what the carry in is, $C'_k$ can then be seen independent from the rest of the bits. The probability of $C'_k$ being either 0 or 1 equals to the probability of a single cell which can be obtained from the modified truth table.

**Proof**: Since $C'_k(ab, s, C'_{k-1} = 0) = C'_k(ab, s, C'_{k-1} = 1)$, then $p(c'_j | 0) = p(c'_j | 1) = p(c'_j)$. As per Eq. (18), $p(C'_k = s) = [p(C'_{k-1} = 0) + p(C'_{k-1} = 1)] p(c'_j) = p(c'_j)$.

According to Theorem 1, on this condition, Eq. (16) becomes:

$$p_{\Delta S'_{k+1}} = \sum_{j=0}^{1} p(c'_j) p(\Delta S'_{k+1} | c'_j)$$  \hspace{1cm} (19)$$

At last, let's consider $M_2$. Note $e_{l_{k+1}}$ and $e_{abn}$ are indeed correlated with $C'_k$, hence $M_2$ can be reformulated as:

$$N'M_2 = 2 \sum_{l_{k+1} \in L} \sum_{s=0}^{1} e_{l_{k+1}|C'_{k+1} = s} \left( \sum_{abn,s} e_{abn,s} \right)$$  \hspace{1cm} (20)$$

where $\sum e_{abn,s}$ denotes the summation of the errors induced by $abn$ with $C'_k = s$. This term actually should be obtained when adding leading bits for $(k-1)$-bit width addition. Hence for ease of representation, $e_{abu,q}$ is shown to show the calculation:

$$\sum e_{abu,q} = \sum_{j=0}^{1} \left[ N_{q|s} (\sum_{abn,s} e_{abn,s}) + 2^k N_s \sum_{l_{k+1} \in L} \Delta S'_{k+1|q|s} \right]$$  \hspace{1cm} (21)$$

where $N_{q|s}$ means the number given $s$ and $q$ as carry in and carry out for the $(k+1)$th bit adder and obviously $\sum_{s=0}^{1} \sum_{q=0}^{1} N_{q|s} = 8$. $\Delta S'_{k+1|q|s}$ represents the difference between $S'_{k+1}$ and $S_{k+1}$ with carry out $q$ given carry in $s$. $N_s$ is the number of outputs with carry out as $s$ for $k$th bit and is defined as:

$$N_s = 4^k p(C'_k = s)$$  \hspace{1cm} (22)$$
Hence, based on Eq. (15)∼(22), the MSE of any given bit-width can be calculated. The Algorithm 1 is shown to formalize the process.

**Algorithm 1: Calculation of MSE for K-bit adder**

**Input:** A Modified Truth Table of Approximate Adder and bit-width K  
**Output:** MSE$_1$∼MSE$_K$

**Initialization:** $k = 0$, $p(C_0 = 0) = 1$, $N_0 = 1$ and $N_1 = 0$. Calculate $\Delta S'_1$, $p_{\Delta S'_1} = p(\Delta S'_1|0)$, $\sum e_{ab,s} = \sum \Delta S'_{1,0}$, $N_{q,s}$ ($q, s \in \{0, 1\}$), $\sum e_{ab,s} = \sum \Delta S'_{1,0}$, $k = k + 1$.

**while**($k < K$)

**Step 1:**
Calculate Eq. (18), Eq. (16) (or Eq. (19) if the Truth Table meets Theorem 1), Eq. (17), Eq. (20) and Eq. (15) by order.

**Step 2:**
Calculate Eq. (22) and Eq. (21). $\sum e_{ab,s} = \sum e_{ab,s}$, ($s = q$), $k = k + 1$.

end

The initialization first defines several parameters for the special 0th bit. In fact, 0th bit only gives a fixed 0 as carry in for the 1st bit. Based on the pre-defined values, $\Delta S'_1$, $p_{\Delta S'_1} = p(\Delta S'_1|0)$ and MSE$_1$ are calculated. Then $N_{q,s}$ and $\sum e_{ab,s}$ are calculated according to the modified truth table compared to the accurate one. These values are only calculated once and used for higher bits since same type of APA is used for addition. Within the loop, Step 1 shows the general process of calculating MSE for rest of bits. Step 2 calculates $\sum e_{ab,s}$ as $\sum e_{ab,s}$ (note $s = q$) and $N_i$ for the next iteration. Hence the complexity of proposed approach is approximately $O(K)$ while the naive simulation is $O(4^K)$, which substantially reduces complexity.

### 2.4 Absolute error characteristics

The previous section describes the modeling of MSE. Another metric used frequently is the mean error distance (MED) [10] which is essentially the mean absolute error (MAE). The result could be obtained recursively as did for MSE. Assume an error matrix for the $k$-bit addition denoted as $error\_mat_k$ which is defined as:

$$error\_mat_k = \begin{pmatrix} \sum e^+_{ab,0} & \sum e^+_{ab,1} \\ \sum e^-_{ab,0} & \sum e^-_{ab,1} \end{pmatrix}$$  \hspace{1cm} (23)$$

$\sum e^+_{ab,s}$ represents the sum of positive errors when $C'_k = s$ and $\sum e^-_{ab,s}$ is similarly defined and $\sum e^+_{ab,s} + \sum e^-_{ab,s} = \sum e_{ab,s}$ which is defined in Eq. (20). When adding the leading bits $l_{k+1}$ for $(k + 1)$th bit, Eq. (21) could be rewritten as:
\[
\begin{align*}
\sum e_{ab_{n;0}}^{+} &= \sum N_{q,s}^{k} \sum e_{ab_{n;0}}^{+} + \sum_{p} \max \left(0, \sum e_{ab_{n;0}} - 2^{k}N_{s}\Delta S'_{n+1,q,s}\right) \\
\sum e_{ab_{n;1}}^{-} &= \sum N_{q,s}^{k} \sum e_{ab_{n;1}}^{-} + \sum_{p} \min \left(0, \sum e_{ab_{n;1}} + 2^{k}N_{s}\Delta S'_{n+1,q,s}\right)
\end{align*}
\tag{24}
\]

where \(N_{q,s}^{k}\) means number of accurate cells with \(q\) given \(s\) in the truth table and \(W\) is the condition when \(l_{k+1} = L\) and \(\Delta S'_{n+1,q,s}\) ≠ 0. Obviously, the \((k + 1)\)th error matrix can now be represented as:

\[
error_{mat_{k+1}} = \begin{pmatrix}
\sum e_{ab_{n;0}}^{+} & \sum e_{ab_{n;1}}^{+} \\
\sum e_{ab_{n;0}}^{-} & \sum e_{ab_{n;1}}^{-}
\end{pmatrix}
\tag{25}
\]

Then the MAE is calculated as:

\[
MAE_{k+1} = \sum |error_{mat_{k+1}}| / 4^{k+1}
\tag{26}
\]

Actually the error matrix contains only the summation of positive and negative errors, yet no descriptions on detailed error values and corresponding number of errors. Hence, the above model can only accurately calculate MAE for “unbiased” approximate adder cell. The biasing is defined as:

\[
\begin{align*}
|\Delta S_{k+1}|[0] + |\Delta S'_{k+1}|[1] &= \|\Delta S'_{k+1}[0] - |\Delta S'_{k+1}|[1]|| \\
|\Delta S'_{k+1}|[0] &\neq |\Delta S'_{k+1}|[1](\neq 0)
\end{align*}
\tag{27}
\]

which indicates that the biasing means \(\Delta S'_{k+1}[0]\) and \(\Delta S'_{k+1}[1]\) have opposite signs and non-equal, non-zero absolute values. Although the proposed model can only accurately calculate MAE for unbiased adder, it still is useful to approximately estimate MAE for biasing adders. Algorithm 2 is shown to formalize the calculation process.

The process first forms the error matrix for the 1st bit from modified truth table and calculates several parameters which will be used in Eq. (24). Within the loop, general process of calculating MAE for rest of bits is shown. The complexity of MAE estimation is approximately \(O(K)\). Hence, algorithm 1 and 2 form the analytical framework for analyzing the error characteristics of approximate adder cell based addition.

**Algorithm2: Calculation of MAE for K-bit adder**

**Input:** A Modified Truth Table and bit-width \(K\)

**Output:** \(MAE_{1} \sim MAE_{K}\)

\(p(C'_{0} = 0) = 1\). Form \(error_{mat_{1}}\). Calculate \(N_{q,s}^{k}\) and \(MAE_{1}, k = 1\).

\(\text{while}(k < K)\)

Calculate Eq. (18), Eq. (24), Eq. (25) and Eq. (26).

\(k = k + 1\).

\(\text{end}\)
3 Experiments and results

3.1 Calculation example

Before validation, we will use an example to show the calculation procedure. The approximate mirror adder (AMA1) is chosen to be used for analysis. Table I shows the truth table of AMA1 as in [6].

Table I. Truth table for AMA1

| a | b | C_{in} | C_{out}&Sum | C'_{out}&Sum | ΔS |
|---|---|--------|-------------|-------------|----|
| 0 | 0 | 0      | 00          | 01          | 1  |
| 0 | 0 | 1      | 01          | 01          | 0  |
| 0 | 1 | 0      | 01          | 10          | 1  |
| 0 | 1 | 1      | 10          | 10          | 0  |
| 1 | 0 | 0      | 01          | 01          | 0  |
| 1 | 0 | 1      | 10          | 10          | 0  |
| 1 | 1 | 0      | 10          | 10          | 0  |
| 1 | 1 | 1      | 11          | 10          | −1 |

MSE is calculated first. According to Algorithm 1, during initialization, \( N_{000} = 2, N_{011} = 1, N_{100} = 2, N_{111} = 3 \); \( ΔS' = \{ 1, 0, −1 \} \), \( ΔS'_1 = 1 \) and \( p_{ΔS'_1} = p(ΔS'_1|0) = \frac{1}{2} \) are calculated based on Table I. Then \( MSE_1 = M_1 = \frac{1}{2} \times 1 = 0.5 \). \( \sum e_{abs,0} = \sum e_{abs,1} = 1 \). \( k \) then becomes 1.

Within the loop with \( k = 1 \) when following step 1, several parameters are obtained: \( p(C'_1 = 0) = p(c'_0|0) = \frac{1}{2} \), \( p(C'_1 = 1) = p(c'_1|0) = \frac{1}{2} \) which also means \( C'_1 \) has both 0 and 1. Based on Table I, we could find \( ΔS'_2 = \{ 1, 0, −1 \} \). As per Eq. (16), \( p_1 = \frac{1}{2} \times 1 + \frac{1}{2} \times 0 = \frac{1}{2} \) while \( p_{−1} = \frac{1}{2} \times 0 + \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} \) for 2nd bit. Eq. (17) then becomes \( M_1 = \frac{1}{2} \times 4 + \frac{1}{8} \times 4 = 1.5 \). Since \( \sum e_{abs,0} = 1 \) and \( \sum e_{abs,1} = 1 \), as per Eq. (20), \( M_2 = 2 \times \frac{2^1+2^2(−2)^1+1}{16} = 0.25 \). Hence \( MSE_2 = MSE_1 + M_1 + M_2 = 0.5 + 1.5 + 0.25 = 2.25 \).

Then as per Step 2, according to Eq. (22), we can obtain \( N_0 = 2 \) and \( N_1 = 2 \). As per Eq. (21),

\[
\sum e_{abs} = N_{d=0} \sum e_{abs,0} + 2^1 \times N_0 \sum ΔS'_{1,d=0} + N_{d=1} \sum e_{abs,1} + 2^1 \times N_1 \sum ΔS'_{1,d=1}
\]

based on the truth table. \( \sum ΔS'_{1,d=0} = 1 \), \( \sum ΔS'_{1,d=1} = 1 \), \( \sum ΔS'_{1,d=0} = 0 \), \( \sum ΔS'_{1,d=1} = −1 \). \( \sum e_{abs,0} = 2 \times 1 + 2 \times 2 \times 1 + 1 \times 1 + 2 \times 0 = 7 \), similarly \( \sum e_{abs,1} = 5 \). Then use the result to update \( \sum e_{abs} \) (note \( q = s \)). The MSE for other bit-width can be calculated recursively.

Then MAE is calculated. From Table I, AMA1 is clearly an unbiased adder cell, hence Algorithm2 can give accurate results in terms of MAE. According to the truth table, \( error_{mat} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \) and \( N_0^2 = 1, N_1^2 = 1, N_0^3 = 1 \) and \( N_1^3 = 2 \). As per Eq. (18), \( N_0 = 2 \) and \( N_1 = 2 \). As per Eq. (24), \( \sum e_{abs,0}^+ = 1 \times 1 + 1 \times 1 + 1 + 2 \times 2 = 7 \); \( \sum e_{abs,0}^- = 1 \times 1 + 2 \times 2 \times 1 + 1 \times 1 + 2 \times 0 = 8 \); \( \sum e_{abs,1}^+ = 0 + 0 + min(0, 1 + 2 \times 2) + min(0, 1 + 2 \times 2) = 0 \); \( \sum e_{abs,1}^- = 0 + 0 + min(0, 1 + 2 \times 2) + min(0, 1 −
The error matrix becomes $\text{error} \_\text{mat}_2 = \begin{pmatrix} 7 & 8 \\ 0 & 3 \end{pmatrix}$ with $\text{MAE}_2 = 1.125$. MAE for other bit widths can be calculated similarly.

### 3.2 Validation

This section shows the validation for the proposed framework. Designs of [6] denoted as AMA1~AMA5 are used for validation. According to [6], AMA1~5 are unbiased adder cells, hence Algorithm 2 can measure MAE accurately.

Fig. 2 presents the simulated and theoretical results for chosen designs by varying the bit-width (i.e., $K$) of an RCA from 1~16. Due to space limitation, AMA1, AMA4 and 5 are chosen. Based on their Boolean functions, AMA1 and 4 are non-bitwise design (i.e., the errors of adjacent bit positions are correlated) while AMA5 is bit wise design. Matlab models are implemented for simulation and the results are obtained by using exhaustive input combinations (i.e., $4^K$) for each bit.
width. Obviously, the results from simulation and theoretical analysis are accurately matched for both bitwise and non-bitwise type.

3.3 Application: Image compression evaluation using PSNR

[10] has modeled the relation between PSNR and MSE, MAE as:

$$PSNR \approx 20\log(\frac{MAX_I}{\sqrt{aMSE + a^2MAD^2}})$$  (28)

where $a$ means the number of approximate addition operations processed for each pixel in an image and $MAX_I$ means the maximal possible pixel value of original image. AMA1$\sim$AMA5 are implemented in a DCT-IDCT image compression system with a Lenna image [6]. In the system, the multiplication is converted to shifts and additions then $a$ is 64 for this DCT-IDCT system. For addition, the lower 7 and 8 LSBs are approximately calculated with corresponding AMAs.

As Fig. 3 shows, the adopted approach could estimate AMA2$\sim$5 accurately for 7 LSB case while AMA1 for 8 LSB case. If the desired PSNR value is set to be higher than 30 dB, the approximation length for AMA1 should be no larger than lower 8 LSBs while AMA2, 4, 5 should be no larger than lower 7 LSBs. In such case, the maximal PSNR error margin is within 4.2 dB; so the given MSE and MED combination is a good estimator for the PSNR. The application also shows the usefulness of the proposed framework.

4 Conclusion

In this paper, an analytical framework has been proposed for characterizing approximate adder designs. The framework consists of mathematical models for the evaluation of arbitrary approximate adder characterized by modified truth table targeting mean squared error (MSE) and mean absolute error (MAE). Time-consuming simulation can then be avoided by employing the proposed analytical models. Extensive simulation results verify the correctness of the proposed model with adder designs from previous literatures. A estimation approach for DCT-IDCT image compression system shows that the final PSNR is closely related to both MSE and MAE. Hence the proposed framework may provide insights of designing approximate arithmetic based error-tolerant applications.