Analysis of power–accuracy trade-off in digital signal processing applications using low-power approximate adders

Celia Dharmaraj | Vinita Vasudevan | Nitin Chandrachoodan

Department of Electrical Engineering, Indian Institute of Technology, Madras, India

Correspondence
Celia Dharmaraj, Department of Electrical Engineering, Indian Institute of Technology, Madras, India.
Email: lee13d003@ee.iitm.ac.in

Abstract
In recent years, approximate circuit design targeting the error-tolerant applications has gained significance. In this study, the authors propose a metric that ranks a stand-alone approximate adder in terms of power savings obtained for a given mean error distance/mean square error (MSE). The authors demonstrate that this ranking of approximate adders can be used in applications that contain adder trees and registers. In applications that also have accurate multipliers interspersed with adders, the authors find that certain types of approximations in the adders result in more power-efficient implementations of multipliers. Besides power savings, the other metrics of interest are noise floor and mean error in filtering applications and the compression achieved for a given peak signal-to-noise ratio (PSNR) in image compression applications. The authors also show that for the same overall MSE, there is a trade-off between noise floor and mean error. This makes it possible to classify these adders based on whether they result in an increased noise floor or a mean error for the same overall MSE. Furthermore, the authors discuss the effect of using an approximate discrete cosine transform block to meet the reduced PSNR requirements, on the overall compression levels and the trade-offs involved in the process.

1 | INTRODUCTION

Approximate computing is widely used in the error-tolerant applications to obtain improvements in either power or speed while maintaining a required accuracy. As adders are the basic building blocks in these applications, a variety of approximate adders have been proposed in the literature. The approximate adders can be broadly classified as either low-power approximate adders (LP AAs) [1–7] or low-latency approximate adders (also categorized as accuracy configurable adders—[8] and references therein). Studies have indicated that LP AAs can result in significant power savings when used in applications [9–11]. The purpose of this study is to improve the power–accuracy trade-off in systems, considering LP AAs. All the existing LP AAs are two-part segmented adders. In an \((N, k)\)-bit LPAA, out of \(N\) bits, \(k\) lower-order bits of the sum are approximated using simpler hardware, thus resulting in savings in power.

For individual LP AAs, the power–accuracy trade-off can be measured using the following two figures of merit (FOMs):

1. Power per bit-normalized error distance (Power-NED) product [12]: Adders with lower Power-NED product are considered better.
2. Power saving per bit-NED (Power saving-NED) ratio [12]: Adders with higher Power saving-NED ratio are considered better.

These FOMs are based on the normalized power or power savings per approximate bit and the NED, both of which are obtained using experiments and Monte Carlo simulations. The NED is a measure of how rapidly the error grows with an increase in the number of approximate bits.

On comparing various LP AAs with respect to these two metrics, we found that adders considered ‘good’ with respect to one metric are not necessarily ‘as good’ with respect to the other metric. Moreover, the adders are not very accurate predictors of performance when it comes to a comparison of power savings for a given accuracy constraint (mean error distance [MED] or mean square error [MSE]). MED and MSE
are the main error metrics that capture the accuracy constraint (noise power, peak signal-to-noise ratio [PSNR]) in most signal processing applications. However, as MED is not a linear function of the NED, if we look at the power savings for a given MED, the ranking of adders could be very different from that obtained using the two existing FOMs.

Based on the same experimental curves as the two existing FOMs, we propose an alternate FOM that ranks a stand-alone adder in terms of power savings for a given MED/MSE. We show that for large values of MED, the power savings per bit is more indicative of the potential overall power savings, while for small and medium MEDs, both the NED and power savings per bit are equally important.

In the literature, stand-alone adders have been studied extensively and ranked in terms of various metrics (see e.g. Ref. [11]). However, these rankings do not in general reflect the true power savings when these adders are used in a larger system, and we have not seen any discussion on this aspect. Systems contain multiple adders, and in order to do a fair comparison, we need to find the optimal set of approximate bits for each adder such that the system specification is met. For this purpose, we have used an optimizer developed by us [13]. Once the optimal number of bits that can be approximated is known, it is possible to compare the potential power savings due to each type of adder. We have done this for FIR filter, IIR filter, Gaussian filter, and discrete cosine transform (DCT) module.

We demonstrate that the ranking obtained for stand-alone adders using the proposed FOM can be used in systems comprising primarily of adder trees (e.g. Gaussian filters, multiplier-less DCT, direct form I FIR filter) to assess power–accuracy trade-off. However, in systems that have interspersed multipliers and adders, such as an IIR filter, it becomes important to consider the effect of the approximate adder on the synthesized structure of the multiplier. Even if the multipliers are accurate, different approximations in the adder lead to different synthesized multiplier architectures, thus affecting the overall power savings.

Besides power savings, the other metrics that are of interest in filtering applications are the noise floor, distortion, and mean error. We show that there is a trade-off between the mean error and the noise floor, so that it is possible to classify the adders into two sets—one that gives a lower noise floor and distortion but higher mean error, and the other has no mean error, but a higher noise floor.

In image compression applications, most studies have focussed on power savings obtained for a particular PSNR [1,5,14]. A metric of interest is the amount of compression that can be obtained for a given PSNR. The JPEG compression algorithm has DCT followed by a quantization step, specified by a quality level Q (0–100). The quality level determines the degree of compression, with lower quality resulting in higher compression (lower number of retained coefficients). For a given Q, Snigdha et al. [15] found the PSNR degradation with DCT using AMA5 adders [1]. It is not clear what will happen if the compression requirements change at run time. Almurray et al. [16] studied the use of AMA1-4 [1] and InXA1-3 [6] adders in an approximate DCT block [17] and computed the PSNR for different levels of compression, with the number of approximate bits for all adders being fixed at 3 or 4 or 5 bits. They concluded that a reasonable trade-off between power savings, delay, and compression is to use InXA2 with 4 bits approximated.

In order to compare power savings and compression in DCT across adders, we used the optimizer [13] to find the optimal number of approximate bits of each adder in the DCT block for a given PSNR. This enables us to identify three different PSNR regimes for DCT with corresponding clear guidelines for the choice of adder in each regime, which we have not seen in any previous work. The purpose of this study is to explore various power–accuracy trade-offs and obtain a framework that will help choose the best approximate adder in a particular technology for an application.

Our main contributions can be summarized as follows:

1. We propose an alternate FOM that can be used to rank stand-alone approximate adders in terms of power savings for a given error when implemented in a given technology
2. We demonstrate that it is possible to use this FOM in applications that primarily contain adder trees and registers to assess power–accuracy trade-off
3. If the architecture has multipliers interspersed with adders, we demonstrate that it is important to consider the effect of the approximate adder on the synthesized structure of the multiplier. This alters the overall power savings significantly. We propose an alternate guideline for choice of the adder
4. In filtering applications, there is a trade-off between the noise floor and mean error for the same overall MSE. Based on this, we classify LPAAs based on whether they cause an increased noise floor or a larger mean error
5. In image compression applications, we provide guidelines for the choice of approximate adder to get better trade-off between PSNR and level of compression

This study is organized as follows: In Section 2, we provide a background on LPAAs and notation used. In Section 3, we describe the various metrics used to evaluate the power saving and error of LPAAs. In Section 4, we explain how to identify the energy-efficient LPAAs. We discuss on the power–accuracy trade-off obtained in various systems in Section 5. The effects of approximate adders on output qualities of various applications are discussed in Section 6, and finally, Section 7 concludes the article.

2 | NOTATION AND BACKGROUND

Consider an accurate N-bit adder with inputs A, B and sum S. Let the inputs be denoted as a_{N-1}, a_{N-2}...a_0 and b_{N-1}, b_{N-2}...b_0, respectively, in binary representation. The adder output S has binary representation s_{N}, s_{N-1}...s_0.

In the case of an LPA, the adder is divided into two parts. The lower k bits of the sum are approximated and the upper N-k bits of the sum are computed accurately. The input A is
divided into $A_H$ and $A_L$, where $A_H = a_{N-1}a_{N-2}...a_k$ denotes the upper part containing $N-k$ bits and $A_L = a_{k-1}...a_0$ denotes the lower part containing $k$ bits. The input $B$ is denoted as $B_BB_B$ in a similar way. Let the approximate sum be denoted as $\hat{S}$ with a binary representation $\hat{s}_N\hat{s}_{N-1}...\hat{s}_k\hat{s}_{k-1}...\hat{s}_0$. $\hat{S}$ is also denoted as $\hat{S}_H\hat{S}_L$, where $\hat{S}_H = \hat{s}_N\hat{s}_{N-1}...\hat{s}_k$ is the upper-part sum and $\hat{S}_L = \hat{s}_{k-1}...\hat{s}_0$ is the lower-part sum. $\hat{c}_{k-1}$ denotes the approximate carry bit to the upper part, obtained using $A_L$ and $B_L$. Figure 1 illustrates this notation. The upper part of the sum denoted by $\hat{S}_H$ is obtained by adding the corresponding bits of the inputs and the approximate carry $\hat{c}_{k-1}$ using an accurate sub-adder. Some kind of approximate logic is used to compute the lower part of the sum $\hat{S}_L$ and the approximate carry $\hat{c}_{k-1}$.

2.1 Existing LPAAs

LPAAs proposed in the literature can be broadly classified into two categories based on whether or not they require carry propagation for evaluation of the lower part of the approximate sum. LPAAs that do not require carry propagation include Truncation adder ($\hat{S}_L = 0$ and $\hat{c}_{k-1} = 0$), lower-part-OR adder (LOA) [3], ETA-I and ETA-III adders [4], AMA5 adder [1], the InXA1 adder [6], and the median adder (MA) [7]. AMA1-4 [1], AXA1-3 [2], InXA2-3 [6], and TGA1-2 [5] require carry propagation for evaluation of the lower part of the approximate sum.

3 EVALUATION OF LPAAs IN TERMS OF ERROR AND POWER METRICS

3.1 Error metrics

In an LPA, since the upper-part sum is obtained using an accurate adder, the error occurs only due to the approximate computation of lower-part sum. Hence, the error is the difference between the accurate lower-part sum and the approximate lower-part sum. If the number of approximate bits is $k$, the error can be written as follows:

$$e = \sum_{i=0}^{k-1} (a_i + b_i)2^i - \hat{c}_{k-1}2^k - \sum_{i=0}^{k-1} \hat{s}_i2^i. \quad (1)$$

The error metrics that are used in the literature are MSE, MED, NED, mean relative error distance (MRED), and error rate [12,18,19]. Of these, the most widely used metrics for signal and image processing applications are MSE, MED, MRED, and NED. The definition of each of these metrics is given in Table 1 for $(N,k)$-bit approximate adders. The metrics are obtained by averaging over $n$ input pairs. $\hat{S}_i$ denotes the accurate sum of the $i^{th}$ input pair, while $\hat{S}_i$ denotes its approximate sum. For some cases, we can get analytical expressions for the metrics.

Figure 2a shows log$_2$MED as a function of the number of approximate bits $k$ for various approximate adders. From the figure, it is seen that the slope is the same for all adders, and it is very close to one. Hence, $\text{MED} \approx c \times 2^k$, where $c$ is different for each adder. By definition, $c = \text{NED}$. The NED is a measure of how rapidly the MED grows with every additional approximate bit. A low value of NED indicates that error grows slowly and consequently a larger number of approximate bits can be used for the same MED. Table 2 shows the values of NED for various approximate adders. It is seen that Truncation adder has the highest NED and TGA2 adder has the least NED.

In applications like signal processing and image processing, MSE is used in noise power and PSNR computations. The variation in root mean square error (RMSE) with the number of approximate bits in various 16-bit approximate adders is shown in Figure 2b. It shows a similar trend as that for MED. Also, for a given number of approximate bits, Truncation adder has the largest MSE and TGA2 has the least value of MSE. A similar trend is also seen for MRED in Figure 2c.

3.2 Metrics to evaluate the power consumption

The total power savings is computed as the difference between the power consumed by an $N$-bit accurate adder ($P_{\text{acc}}$) and the power consumed by an $(N,k)$-bit approximate adder ($P_A$). It is given by

$$P_{1A} = P_{\text{acc}} - P_A. \quad (2)$$

The percentage power savings of an approximate adder $A$ is given by

\begin{table}[h]
\centering
\caption{Definition of various error metrics}
\begin{tabular}{|c|c|}
\hline
Error metric & Definition \\
\hline
MED & $\frac{1}{n} \sum_{i=0}^{n-1} |S_i - \hat{S}_i|$ \\
NED & $\frac{\text{MED}}{2}$ \\
MRED & $\frac{1}{n} \left( \sum_{i=0}^{n-1} \frac{|S_i - \hat{S}_i|}{k} \right)$ \\
MSE & $\frac{1}{n} \sum_{i=0}^{n-1} (S_i - \hat{S}_i)^2$ \\
\hline
\end{tabular}
\end{table}
We implemented each approximate adder using Faraday 55 and 28 nm technologies, and then simulated the adder using $10^6$ uniform random inputs to obtain the power consumption. The synthesis tool used was Cadence Genus.

The percentage power savings of various LP AAs is shown in Figure 3 as a function of $k$. From the figure, it is seen that for all adders, the percentage power savings increase linearly with $k$. The Truncation and Median adders give the most power savings irrespective of the number of bits approximated as the lower-part sum is a constant for both these adders. The next best adder in terms of power savings is AMA5, where the power consumed is due to toggles in the input. Also as expected, LP AAs that do not have carry propagation for the approximate lower-part sum are more power efficient than the ones that do.

Let $P_{b,acc}$ be the power consumed per bit by the $N$-bit accurate adder. The normalized power saving per bit of an $(N, k)$-bit approximate adder $A$ can be obtained as

$$P_{mb,A} = \frac{P_{b,acc} - P_{b,acc} \times \frac{k}{N}}{P_{b,acc} \times \frac{k}{N}}.$$  

This is used as a metric to evaluate the power savings possible with various approximate adders. As mentioned, this is a technology-dependent number. However, it will always be equal to one for Truncation and Median adder as no hardware is required to compute the approximate sum. The values for other approximate adders depend on the hardware required for the lower part of the sum. For Faraday 55 and 28 nm technologies, the value of $P_{mb,A}$ for various approximate adders is tabulated in Table 3. The value obtained for LOA matches well with the value reported in [12].

Table 4 summarizes the notation used to describe the power metrics.

### Measures for power–accuracy trade-off

The trade-off in LP AAs is between power savings and error. Two FOMs that have been used in the literature to capture this trade-off are the power per bit-NED product [12] and power saving per bit-NED ratio [12]. The power-NED product is given by the product of normalized power consumed per approximate bit.
TABLE 3 Normalized power saving per approximate bit of various adders obtained for Faraday 55 nm (28 nm) technology

| Adder         | \( P_{\text{nsb}}/A \) | Adder         | \( P_{\text{nsb}}/A \) | Adder         | \( P_{\text{nsb}}/A \) |
|---------------|-----------------|---------------|-----------------|---------------|-----------------|
| Trunc, MA     | 1 (1)           | ETA-I         | 0.58 (0.47)     | TGA1          | 0.31 (0.20)     |
| AMA5          | 0.88 (0.85)     | TGA2          | 0.42 (0.17)     | AXA2          | 0.18 (0.10)     |
| LOA           | 0.72 (0.71)     | InXA3         | 0.35 (0.22)     | AXA3          | 0.08 (0)        |
| InXA1         | 0.63 (0.59)     | AXA1          | 0.33 (0.27)     | InXA2         | 0.02 (0)        |

Abbreviations: LOA, lower-part-OR adder; MA, median adder.

TABLE 4 Notation used in describing various power metrics

| Notation | Definition |
|----------|------------|
| \( P_{\text{acc}} \) | Power consumed by an \( N \)-bit accurate adder |
| \( P_{\text{b,acc}} \) | Power per bit of an \( N \)-bit accurate adder, given by \( P_{\text{acc}}/N \) |
| \( P_A \) | Power consumed by an \( (N, k) \)-bit approximate adder \( A \) |
| \( P_{s,A} \) | Power saving of adder \( A \), given by \( P_{\text{acc}} - P_A \) |
| \( \% P_{s,A} \) | Percentage power saving, given by \( (P_{\text{acc}}/P_A) \times 100\% \) |
| \( P_{\text{nsb},A} \) | Normalized power saving of adder \( A \), given by \( P_{s,A}/P_{\text{b,acc}} \) |
| \( P_{\text{nsb},A} \) | Normalized power saving per bit, given by \( P_{\text{nsb},A}/P_{\text{b,acc}k} \) |

\((P_{\text{nsb},A})\) and NED. The smaller the product, the better is the adder in terms of trade-off between power and accuracy. Power saving-NED ratio is given by the ratio of normalized power saving per approximate bit \((P_{\text{nsb},A})\) and NED. Adders that have a larger value of the ratio are better as they have larger power savings and a lower rate of growth of the MED.

Tables 5 and 6 list the values of these two FOMs for various approximate adders in 55 and 28 nm technologies, and the following conclusions can be drawn from these tables.

1. Based on the power per bit-NED product, the Truncation and Median adders are considered to be the best adders. A drawback is that it is not possible to distinguish between them. Both adders do not perform well in terms of power savings-NED ratio, with the Truncation adder performing very poor.

2. Although the exact ordering is different in the two metrics, LOA, AMA5, TGA2, and ETA-I are highly ranked in both metrics.

3. By both metrics, TGA1 and TGA2 are ranked higher than InXA1 and other inexact adders, though TGA requires additional hardware for carry propagation.

We emphasize that these are technology-dependent numbers and the actual ordering depends on the technology used. However, we do expect LOA, AMA5, and ETA-I to perform well in all technologies as (a) they do not require carry propagation in order to compute the approximate sum, reducing hardware requirement and (b) they have a relatively small value of NED. TGA2 has a relatively low value of power savings, but a very low value of NED, while InXA1 has a relatively high value of NED and power savings.

\( 4 \mid \text{PROPOSED MEASURE FOR COMPARISON OF APPROXIMATE ADDERS} \)

In most of the signal processing applications, the accuracy metric is MED/MSE or PSNR, and we are interested in maximizing power savings for a given MED/MSE. Figure 4a is a plot of the percentage power savings as a function of log \( \text{MED} \), which is obtained from Figures 2a and 3. It is seen that MA gives the largest power savings for a considerable range of MED, which is not reflected by both the existing FOMs. This is followed by Truncation, AMA5, and LOA. Truncation actually performs better than LOA for moderate MEDs, and both ETA-I and InXA1 perform better than TGA2, which is contrary to the ranking obtained using the two FOMs. The actual ranking of adders also depends on the MED. For low MEDs, for example, ETA-I is better than Truncation, while for moderate MEDs, it becomes worse. Also, as expected intuitively, all LPAAIs that do not require carry propagation for evaluation of the approximate sum perform better, that is, they have larger power savings for a given MED.

The reason the two FOMs in the literature do not predict the power savings for a given MED well is because the MED is a non-linear function of the NED. If \( k_A \) is the number of approximate bits in adder \( A \) for a given MED, the total power savings per bit-NED ratio can be written as follows:

\[
\frac{P_{\text{nsb},A}}{\text{NED}} = \frac{P_{s,A}}{\text{MED}} \times \frac{2^{k_A}}{k_A}.
\]

As seen from the equation above, the two ratios \( P_{\text{nsb},A}/\text{NED} \) and \( P_{s,A}/(\text{MED} \times k_A) \) are not equivalent due to the non-linear dependence of MED on the NED.
This can be taken into account as follows. The total normalized power savings of an adder $A$ for a given MED can be written as follows:

$$P_{\text{norm},A} = P_{\text{norm},A_1} \times k_A = P_{\text{norm},A_2}(\log_2 \text{MED} - \log_2 \text{NED}_A).$$ (6)

Therefore, the power savings of adder $A_1$ with respect to adder $A_2$ for a given MED can be written as follows:

$$\frac{P_{\text{ns},A_1}}{P_{\text{ns},A_2}} = \frac{P_{\text{ns},A_2} - \log_2 \text{MED}}{P_{\text{ns},A_2} - \log_2 \text{NED}_A}. $$ (7)

For the Truncation adder, $P_{\text{ns},\text{trunc}} = \text{NED} = 1$, giving $P_{\text{ns},\text{trunc}} = \log_2 \text{MED}$. Using this, our proposed FOM, denoted $FOM_A(MED)$, can be used to rank all adders relative to Truncation adder as follows:

$$FOM_A(MED) = \frac{P_{\text{ns},A}}{P_{\text{ns},\text{trunc}}} = \frac{1 - \log_2 \text{NED}_A}{1 - \log_2 \text{MED}}.$$ (8)

The MA is always better than Truncation since $P_{\text{ns},\text{MA}} = 1$ and $\text{NED} = 0.33$. The ranking of other adders relative to Truncation depends on the power savings per bit, NED and MED. Based on Equation (8), we conclude that for large values of the MED, the adders can be more or less ranked using $P_{\text{ns}}$. This is validated in Figure 4a. For small values of the MED, both NED and $P_{\text{ns}}$ need to be taken into account in order to get the correct ranking. For example, AMA5 is better than Truncation for low and moderate MED due to its lower value of NED.

Both the power-NED product and power-savings-NED ratio ranks InXA1 ($A_1$) lower than TGA2 ($A_2$) for the technology and libraries used. However, Figure 4a indicates that InXA1 is better than TGA2 over a wide range of MED. Using the values of NED and $P_{\text{ns}}$ in Tables 2 and 3, we get $\frac{P_{\text{ns},\text{InXA1}}}{P_{\text{ns},\text{TGA2}}} = 1.5 \times \frac{1 + \sqrt{\text{MED}}}{1 + \sqrt{\text{MED}}}$, which implies $P_{\text{ns,InXA1}}$ is greater than $P_{\text{ns,TGA2}}$ as long as $\log_2 \text{MED} > 3$, which is approximately what is seen in the figure.

Figure 4b is a plot of the percentage power savings as a function of $\log_2(\text{MSE})$, which is obtained from Figures 2b and 3. The trend is similar to that seen in Figure 4a. In image processing applications, the measure used is PSNR. Figure 4c shows the variation in percentage power savings with PSNR for an image addition example. Various values of PSNR are obtained by varying $k_A$. The trend is similar to that seen in Figure 4a,b.

Figure 5 shows the variation in percentage power savings with MED and RMSE in various 16-bit approximate adders and also the variation in percentage power savings with PSNR for image addition in 28 nm technology. It can be seen that the ranking of some LPAAs in 28 nm technology is different from that in 55 nm technology. Based on the power–accuracy trade-off obtained with 55 and 28 nm technologies, it is seen that the approximate adders that do not require carry propagation for computation of the approximate lower-part sum perform better in terms of power savings. Studies in Refs [1,9,10,12] also concluded at a similar results. Therefore, we focussed on Truncation, AMA5, LOA, ETA-I, InXA1, and MA to build larger approximate image and signal processing systems.

5 | POWER–ACCURACY TRADE-OFF IN APPROXIMATE SYSTEMS

When accurate adders are replaced with approximate adders in a system, the overall MSE of the system is contributed by the error generated in each approximate adder. We compute the mean and MSE of each approximate adder using parameterized error models detailed in Ref. [13]. An optimization
framework discussed in Ref. [13] is used to fix the number of approximate bits in each adder for a given accuracy constraint. We first obtain the optimum number of approximate bits for each adder in the system, for a given MSE using the optimization framework. This is done for FIR, IIR, and Gaussian filters and an $8 \times 8$ DCT module. These are typically the benchmarks that have been used in the literature. Using these optimal sets of approximate bits, we implement the circuits in Verilog and synthesize them using Faraday 55 and 28 nm technologies to obtain gate-level netlists in each case. The synthesized netlists along with standard delay format file generated by Synopsys DC are simulated with $10^5$ uniform random inputs for FIR and IIR filter and standard images for Gaussian filter and DCT computation.

5.1 | FIR filter

We used the direct form I realization of an 18-tap low-pass FIR filter, as shown in Figure 6, with accurate adders being replaced with approximate adders. It has 17 adders connected in five levels. In our implementation, we have assumed that the input of the filter has 10 fractional bits and the filter

![Figure 6](image-url) Direct form I structure of an 18-tap low-pass FIR filter

![Figure 7](image-url) Percentage power savings versus approximation noise power at the output of an 18-tap FIR filter implemented using various approximate adders implemented using 55 nm technology. (a) Power savings in adder portion of the FIR filter (55 nm) and (b) Power savings in the entire FIR filter (55 nm)
coefficients and multipliers' outputs and adders' outputs have 15 fractional bits of precision.

We define the output noise power as $10 \times \log_{10} \text{MSE}$. Figure 7a shows the percentage power savings in the adder portion of the circuit versus output noise power. As expected, FIR filter implemented using MA and AMA5 adders give maximum power savings. Since direct form I realization of an FIR filter has multipliers in the first level of the circuit, the power consumed by the multipliers do not depend upon the type of approximate adders used in the design. Hence, the trend obtained while considering the percentage power savings obtained in the entire FIR filter as in Figure 7b is same as that when the adder portion is considered as in Figure 7a. Figure 8a,b shows the results in 28 nm technology, where similar behaviour is seen.

We also note that the ranking of adders in terms of power savings for a given MSE is the same as that of a stand-alone adder for almost all cases, indicating that evaluation of a stand-alone adder is a reasonable predictor of power savings if the approximate part of the circuit is primarily composed of adder trees. From Figure 7a, we see that LOA gives more power savings than Truncation. This is because, at higher levels in the tree, the input static probabilities of the lower-order bits of the LOA become close to 1 due to the repeated logical OR operation of the lower-order bits. When the input static probabilities are close to 1, the power savings of LOA increases significantly as shown in Figure 9. This can be explained from the values of $(P_{\text{nsb}}, NED_A)$, which are $(0.85, 0.25)$ and $(0.95, 0.1)$ for LOA with $P_i = 0.75$ and $0.9375$, respectively. From Equation (7), we see that $P_{\text{nsb}}$ with $P_i = 0.75$ and $P_i = 0.9375$ are greater than $P_{\text{nsb}}$ with $P_i = 0.5$ as long as $\log_2 \text{MED}$ is greater than 0.6 and $-6$, respectively.

5.2 Gaussian filter

An approximate $3 \times 3$ Gaussian filter used to blur an image is considered. Images represented using 8-bit pixels are used as input images. So the inputs to the filter are assumed to have eight fractional bits, that is, Q1.8 in fixed point format. The following $3 \times 3$ Gaussian smoothing kernel is used to perform 2D convolution with the image matrix.

$$G = \frac{1}{16} \times \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Each adder is replaced by approximate adder, and multiplication is achieved using the shift operations. Hence, the circuit structure here is basically an adder tree.

Figure 10a–c shows the percentage power savings obtained by the filter implemented using various approximate adders for three different input images namely Cameraman, Lena, and
Figure 11 shows the same plots in 28 nm technology. We note that the performance trend is very similar to that obtained for FIR filter, which also consisted of an adder tree. Based on this, we draw the following conclusions:

1. If the static probability at the output of an adder is same as that of the input, the FOM can be used directly. This is true for AMA5 and InXA2 adders, for example.

2. If the static probability is different and power savings for a given MED is similar, it is necessary to take into account increased/decreased power saving due to change in static probability.

5.3 | IIR filter

The IIR filter is a direct form II realization of a fourth-order low-pass Butterworth filter obtained by cascading two second-order filters, as shown in Figure 12 with 10 multipliers and eight adders. The adder outputs are multiplied with filter coefficients, and the multiplier outputs are added using approximate adders. In our implementation, we have assumed that the filter inputs are normalized and have 10 fractional bits of precision, while the filter coefficients and multipliers' outputs have 15 fractional bits of precision.

Figure 13a shows the power savings obtained by just the approximate adders for various noise power constraints at the output. Performance of AMA5, LOA, and MA is very similar, and Truncation, InXA1, and ETA-I forms another group. Although the gaps differ, this ranking is not inconsistent with what we obtain for the adder tree. However, when we consider the total percentage power savings, including power consumed by multipliers, shown in Figure 13b, it is seen that...
1. If MA or Truncation adders are used, the percentage power savings changes only marginally when the multiplier power is added.

2. Power savings obtained with LOA and AMA5 drop drastically, with a much larger drop for AMA5.

3. Almost no power savings is obtained with InXA1.

In the IIR filter, one of the inputs to every multiplier comes from an approximate adder. In case of Truncation and MA, out of $N$ fractional bits, $k$ bits are fixed to a constant value of either all 0’s or 1’s. As a result, the synthesis tool simplifies the computation logic used in the multiplier, resulting in smaller multipliers and additional power savings, even though the multipliers are accurate. This is not possible with the other approximate adders.

InX1 and AMA5 give very low power savings compared to the other adders. In both these cases, the static probability at the output of the adder is 0.5, resulting in a

**FIGURE 13** Percentage power savings versus approximation noise power at the output of a fourth-order IIR filter implemented using various approximate adders in 55 nm and 28 nm technologies. (a) Power savings in adder portion of the IIR filter (55 nm), (b) Power savings in the entire IIR filter (55 nm), (c) Power savings in adder portion of the IIR filter (28 nm) and (d) Power savings in the entire IIR filter (28 nm).

**FIGURE 14** Percentage power savings versus noise power at the output of an 8 x 8 discrete cosine transform module implemented using various approximate adders in 55 nm technology with the input images as (a) Cameraman and (b) Lena and (c) Peppers respectively at 55 nm technology. PSNR, peak signal-to-noise ratio.
high toggle rate. This also leads to additional power consumed by the multipliers, reducing the overall power savings. Although the use of LOA and ETA also do not lead to any simplification in multiplier logic, the static probability at the output of these adders is higher, resulting in a lower toggle rate, and consequently, a lower dynamic power consumption in the multipliers. The overall power savings reduces, but it does not drop as drastically as the circuits using AMA5 and InXA1. Similar results are obtained for 28 nm technology as shown in Figure 13c,d.

In conclusion, for circuits that have adder trees, the power savings can be predicted reasonably well using the results for the stand-alone adder. However, if the circuits have interspersed adders and multipliers, it becomes important to take into account the effect of the approximation on the multiplier. In general, an adder whose static probability is 0.5 at its output will result in higher power consumption in the interspersed multipliers.

5.4 | DCT module

Several approximate DCT matrices were studied for image compression in Ref. [16] to reduce power, delay, and error. Among the approximate DCT matrices studied, they conclude that the approximate DCT as presented in [17] gives the best trade-off between power and PSNR. Hence, we consider the implementation of $8 \times 8$ DCT using the transform matrix presented in [17], which is a multiplier-less transformation matrix with entries 0, 1, and $-1$. The circuit consists of 288 adders spread over six levels. In our implementation, the inputs to the DCT algorithm are assumed to have eight fractional bits of precision, that is, Q1.8 in fixed point format.

For various input images such as Cameraman, Lena, and Peppers, $8 \times 8$ DCT was performed with approximation optimized for various noise power values. The percentage power savings obtained is plotted in Figures 14 and 15 for 55 and 28 nm technologies, respectively. MA gives the best trade-off and there is not much to choose between LOA, AMA5, Truncation, and ETA-I adders, but using InXA1 adder is always the worst. LOA once again gains in power savings due to higher static probability and the presence of a few multipliers at the output.

6 | EFFECT OF APPROXIMATE ADDERS ON THE OUTPUT QUALITY

When approximate adders are used in filtering and image compression applications, the other metrics of interest are mean error and noise floor at the output of the filters and the actual compression achieved in image compression.
The effect of using approximate adders in signal processing applications like FIR and IIR filters in the form of distortion in the frequency spectrum of the filtered signal is shown in this section. The effect in both FIR and IIR filters is similar. Therefore, we present the results only for the FIR filter.

A signal containing the sum of sine waves of frequencies 100 and 400 Hz and random noise is fed as input to an 18-tap FIR filter. The filter is a low-pass filter whose pass band is from 0 to 200 Hz and stop band from 300 Hz with at least 30 dB attenuation. The frequency spectrum of the filter output obtained using accurate adders is shown in Figure 16a. It has the low-pass signal at 100 Hz with 40 dB magnitude. Figures 16 and 17 show the output spectrum when various approximate adders are used. Filters implemented using Truncation, ETA-I, and InXAI adders have large DC component of approximately 30 dB at their outputs. This is expected, as the error in these adders is always positive resulting in a large mean error. However, other than the DC shift, the output obtained is closest to that of the accurate filter. All other adders have a significantly higher noise floor. This can be explained as follows.

Let the set of adder output nodes be denoted by \( A \). For each adder node \( n \in A \), let \( T_{n,o} \) denote the transfer function from the node \( n \) to an output node \( o \in O \) computed as in Ref. [20]. Let \( T_{n,o}(e^{j\omega}) \) and \( T_{n,o}(e^{j\omega}) = (1/2\pi)\int_{-\pi}^{\pi} |T_{n,o}(e^{j\omega})|^2 d\omega \) for each \( n \in A \). For a given set of number of approximate bits for all the

**TABLE 7** Power savings and average number of approximate bits out of maximum possible 8 bits for a PSNR of 35 dB with Cameraman image as input (55 nm)

| Adder \((A)\) | %Power saving | Average \(k\) |
|----------------|--------------|---------------|
| MA             | 52.6         | 4.0           |
| LOA            | 41.2         | 4.1           |
| AMAS           | 35.3         | 4.3           |
| Trunc          | 31.1         | 2.8           |

Abbreviations: LOA, lower-part-OR adder; MA, median adder; PSNR, peak signal-to-noise ratio.

Need to fill in the rest of the table.
approximate adders, the mean square error $MSE_o$ at any primary output $o$ can be calculated \[21\] as

$$MSE_o = \left( \sum_{n \in A} T \mu_n \times \mu_n \right)^2 + \sum_{n \in A} \left( T \sigma_n^2 \times \sigma_n^2 \right),$$

where $\mu_n$ and $\sigma_n^2$ are the mean and variance of error introduced in the approximate adder output node $n$. For a given $MSE$, the adders that have large mean error will have lower variance of error that corresponds to lower noise floor. Hence, Truncation, ETA-I, and InXA1 adders having large mean error will have lower noise floor.

There is clearly a trade-off between the mean error and the noise floor. If the system can tolerate a mean error, it is possible to get a much lower noise floor for the same overall $MSE$ constraint using certain approximate adders.

### 6.2 Image compression

In the JPEG standard, a DCT step is followed by quantization. The amount of quantization (called quality level) determines the compression achieved. Let $Q_n$ denote a quantization step that has quality level $n$. $Q_n$ is given by

$$Q_n = \begin{cases} 
\left( \frac{100 - n}{50} \right) Q_{50}, & n > 50 \\
\left( \frac{50}{n} \right) Q_{50}, & n < 50 
\end{cases}$$

where $Q_{50}$ is the JPEG standard quantization matrix. The lower the quantization quality level, the larger the compression and the lower the PSNR. If the DCT block is accurate, a low PSNR can only be obtained by lowering $Q_n$. However, if the
DCT block is implemented using approximate adders, a low PSNR will be obtained even at the highest quality level. This is counterproductive as it does not result in compression.

Using the approximate DCT block, we obtained the overall PSNR as a function of the quality level for three cases. In each case, the level of approximation in the DCT block was determined so that it met a particular PSNR specification at the highest quality level ($Q_n = 100$). In the first case, the optimizer was run for a target PSNR of 35 dB using various approximate adders in DCT without quantization (i.e. quantization quality level of 100). Figure 18 shows the variation in PSNR as the quantization quality level $Q_n$ is decreased from 100 to 10. It is seen that any of the approximate adders can be used with the same $Q_n$ as that used with accurate adders. Table 7 contains the average number of approximate bits and the power savings obtained at this PSNR using various approximate adders. It is seen that we can get up to 52% more power savings using MAs than the one implemented using accurate adders, while achieving the same PSNR and compression.

In the second case, the DCT block is approximated for a PSNR of 27 dB, at quantization quality level of 100. Figure 19 shows the variation in PSNR as $Q_n$ is decreased from 100 to 10. When approximate adders are used, it is seen that the same PSNR is obtained even when the quantization quality level is reduced. As long as the requirement is less than 27 dB, we can use the same quality level as with an accurate DCT block, resulting in the same compression.

In the third case, the optimizer is run for a lower PSNR of 30 dB using various approximate adders in the DCT block with $Q_n = 100$. Figure 20 shows the variation of PSNR as $Q_n$ is decreased from 100 to 10. Table 8 contains the percentage power savings obtained at this PSNR and also the quantization quality level possible for various approximate adders and average number of DCT coefficients that are retained. It is seen that to achieve the target PSNR of 30 dB, different adder configurations provide different trade-offs between the power savings obtained and the amount of compression (determined by $Q_n$). The accurate adder gives the best compression for a given PSNR. If both PSNR and quality level are fixed, it is best to use accurate adders. However, if PSNR is fixed and quality level is relaxed, approximate adders can be used with significant power savings. MA, AMA5, and Truncation adders are closest to accurate adder.

### Table 8: Power savings, quantization quality level $Q_n$ and the average number of RC for a PSNR of 30 dB with Cameraman image as input

| Adder (A) | %Power saving | $Q_n$ | RC |
|----------|---------------|------|----|
| Accurate | -             | 70   | 9  |
| MA       | 60.5          | 80   | 16 |
| LOA      | 55.5          | 90   | 23 |
| AMA5     | 48.6          | 80   | 13 |
| Trunc    | 46.3          | 80   | 17 |

Abbreviations: LOA, lower-part-OR adder; MA, median adder; PSNR, peak signal-to-noise ratio; RC, retained coefficients.

### 7 CONCLUSIONS

We have performed a comprehensive comparison of the existing LPAAs in terms of power–accuracy trade-off and proposed a FOM to rank stand-alone approximate adders in terms of power savings for a given error. We have also demonstrated that this ranking of adders also holds good in applications, such as FIR and Gaussian filters, that primarily contain adder trees and registers. However, in applications, such as IIR filters, that also have multipliers interspersed with the adders, we show that some adders result in more power-efficient implementations of the multipliers, hence resulting in additional power savings.

The output quality of signal processing applications is determined not only by a single accuracy constraint but also by other factors, such as the noise floor, distortion, or compression level. Therefore, the choice of approximate adder and the number of approximate bits need to account for these factors as well.

### ORCID

Celia Dharmaraj [https://orcid.org/0000-0002-7272-2112]

### REFERENCES

1. Gupta, V., et al.: Low-power digital signal processing using approximate adders. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. 32, 1 (2013)
2. Yang, Z., et al.: Approximate xor/xnor-based adders for inexact computing. In: 2013 13th IEEE International Conference on Nano-technology. IEEE-NAANO 2013, Beijing, pp. 90–963. (2013). https://doi.org/10.1109/NAANO.2013.6720793
3. Mahdian, H.R., et al.: Bio-inspired Imprecise computational blocks for efficient VLSI implementation of soft-computing applications. IEEE Trans. Circ. Syst. I: Reg. Pap. 57, 4 (2010)
4. Zhu, N., 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. Syst. 18, 1225–1229 (2010)
5. Yang, Z., Han, J., Lombardi, F.: Transmission gate-based approximate adders for inexact computing. In: Proceedings of the 2015 IEEE ACM International Symposium on Nanoscale Architectures (NANOARCH ’15), Boston, MA, pp. 145–150. (2015). https://doi.org/10.1109/NANOARCH.2015.7180603
6. Almirut, H.A.F., Kumar, T.N., Lombardi, F.: Inexact designs for approximate low power addition by cell replacement. In: Design, Automation and Test in Europe (DATE), Dresden, pp. 660-665. (2016)
7. Celia, D., Vasudevan, V., Chandrasekhoor, N.: Optimizing power-accuracy trade-off in approximate adders. In: Design, Automation Test in Europe Conference Exhibition (DATE), Dresden, pp. 1488–1491. (2018). https://doi.org/10.23919/DATE.2018.8342248
8. Xu, W., Sapatnekar, S.S., Hu, J: A simple yet efficient accuracy-configurable adder design. IEEE Trans. Very Large Scale Integr. Syst. 26, 1112–1125 (June 2018)
9. Paltrinieri, A., et al.: On the effect of approximate-computing in motion estimation. J. Low Power Electron. 15, 40–50 (03 2019)
10. Jiang, H., Han, J., Lombardi, F.: A comparative review and evaluation of approximate adders. In: Proc. of the Great Lakes Symposium on VLSI. GLSVLSI (2015)
11. Jiang, H., et al.: A review, classification, and comparative evaluation of approximate arithmetic circuits. J. Emerg. Technol. Comput. Syst. 13, 1–34 (2017)
12. Liang, J., Han, J., Lombardi, F.: New metrics for the reliability of approximate and probabilistic adders. IEEE Trans. Comput. 62, 1760–1771 (2013)
13. Dharmaraj, C., Vasudevan, V., Chandrachoodan, N.: Optimization of DSP applications using parameterized error models for low power approximate adders. arXiv e-prints, p. arXiv:2001.02102 (Jan 2020)

14. Pashaieifar, M., et al.: A theoretical framework for quality estimation and optimization of dsp applications using low-power approximate adders. IEEE Trans. Circ. Syst.: Reg. Pap. 66, 327–340 (Jan 2019)

15. Snigdha, F.S., et al.: Optimal design of JPEG hardware under the approximate computing paradigm. In: 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, pp. 1–6. (2016). https://doi.org/10.1145/2897937.2898057

16. Almurib, H.A.F., Kumar, T.N., Lombardi, F.: Approximate dct image compression using inexact computing. IEEE Trans. Comput. 67, 149–159 (Feb 2018)

17. Bouguezel, S., Ahmad, M.O., Swamy, M.N.S.: 8x8 transform for image compression. In: 2009 International Conference on Microelectronics (ICM), pp. 74–77. 2009

18. Kang, A.B., Kang, S.: Accuracy-configurable adder for approximate arithmetic designs. In: DAC Design Automation Conference 2012, San Francisco, CA, pp. 820–825. (2012). https://doi.org/10.1145/2228360.2228509

19. Liu, C., Han, J., Lombardi, F.: A low-power, high-performance approximate multiplier with configurable partial error recovery. In: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, pp. 1–4. 2014

20. Mason, S., Zimmermann, H.: Electronic circuits, signals, and systems. New York, NY: Wiley; (1960)

21. Menard, D., Rocher, R., Sentieys, O: Analytical fixed-point accuracy evaluation in linear time-invariant systems. IEEE Trans. Circ. Syst.: Reg. Pap. 55, 3197–3208 (Nov 2008)

How to cite this article: Dharmaraj C, Vasudevan V, Chandrachoodan N. Analysis of power–accuracy trade-off in digital signal processing applications using low-power approximate adders. IET Comput. Digit. Tech. 2021;15:97–111. https://doi.org/10.1049/edt2.12006