Optimizing Power-Accuracy trade-off in Approximate Adders

Celia D, Vinita Vasudevan and Nitin Chandrachoodan
Department Electrical Engineering, Indian Institute of Technology Madras
Chennai, India 600036
Email: {ee13d003,vinita,nitin}@ee.iitm.ac.in

***Abstract*—Approximate circuit design has gained significance in recent years targeting applications like media processing where full accuracy is not required. In this paper, we propose an approximate adder in which the approximate part of the sum is obtained by finding a single optimal level that minimises the mean error distance. Therefore hardware needed for the approximate part computation can be removed, which effectively results in very low power consumption. We compare the proposed adder with various approximate adders in the literature in terms of power and accuracy metrics. The power savings of our adder is shown to be 17% to 55% more than power savings of the existing approximate adders over a significant range of accuracy values. Further, in an image addition application, this adder is shown to provide the best trade-off between PSNR and power.**

***Index Terms*—approximate adder, low power, accuracy, architectural approximation**

I. Introduction

Many signal processing blocks, especially those meant for video and speech, are error tolerant which makes it possible to use inaccurate arithmetic units. This is exploited in systems to save power and area as well as to reduce the delay. Approximation is mainly done using voltage over-scaling and architectural approximation [1], [2]. In voltage over-scaling, supply voltage is scaled down leading to power savings, but causing increased delays. This results in possible timing violations and hence inaccurate results. In architectural approximation, the functionality of the circuit is approximated and simplified so that even at nominal supply voltages, the results are not accurate. The simplification in functionality results in reduced logic density and achieves savings in power, delay and area.

Several approximate adders have been proposed and studied in the literature, each representing a trade-off between power and accuracy. One possibility is to segment the adder into two parts. The upper part of the sum containing its most significant bits (MSBs) are obtained using accurate adders. Approximate logic is used to compute the lower part of the sum containing the remaining least significant bits (LSBs). In such approximate adders, for a given number of lower-order bits being approximated, the power consumed by the accurate upper part is almost the same. Power savings in the lower portion is typically due to reduction in the switching activity due to use of simpler gates.

The simplest of these adders is the truncation adder, where the lower part is set to zero. Since there is no hardware requirement for the lower part, it has the largest savings in both power and area. However, it also results in significant errors [3]. The other two-part segmented approximate adders namely approximate mirror adder [3], lower-part OR adder (LOA) [4], error tolerant adder (ETA) [5] and inexact adder [6], obtain better accuracy by introducing limited computations for approximating the less significant part of the output. However, this results in additional costs in area and power.

There are other approximate adders, such as [7], [8] which do not split the output into approximate lower part and accurate upper part. Instead, the adder is divided into many subadders and carry is predicted. Here we can trade off power and accuracy by varying the size of subadders. However, these adders perform worse than the two-part segmented adders in terms of power-accuracy trade-off [9]. In this paper, therefore we focus on segmented adders with two parts – an accurate upper part and an inaccurate lower part.

It is possible to match the power savings of the truncation adder if the approximate bits are set to a fixed value. In this paper, we propose to set it to a fixed value L, that minimizes the mean error distance (MED). The value is chosen so that it is optimal for all inputs that have a symmetric probability mass function (PMF). If the input PMF is not symmetric, we show that it is close to optimal as long as the number of approximate bits is not too large. We quantify the power savings of various two-part adders interms of the power savings per bit and the MED. We have used the proposed adder in an image addition application to demonstrate its effectiveness.

Section II contains a discussion of the power consumed by various approximate adders proposed in the literature. This is followed by a theoretical analysis leading to the choice of the fixed value L. Simulation results and power accuracy trade-off for various adders are discussed in detail in Section IV. Section V has the conclusions.

II. Comparison of previously proposed adders

A. Notation and Previously proposed adders

Consider an approximate adder with N-bit inputs A, B and $N+1$ bit sum S. Let k be the number of bits in the lower part of the sum which are approximated. The input A, with binary representation $a_{N-1}a_{N-2}...a_k a_{k-1}...a_0$, is denoted as the concatenation $A_H A_L$, where $A_H = a_{N-1}a_{N-2}...a_k$ is the upper part and $A_L = a_{k-1}...a_0$ is the lower part. The input B is denoted as $B_H B_L$ in a similar way. The adder output S has

978-3-9819263-0-9/DATE18/©2018 EDAA

binary representation $s_N s_{N-1}...s_k s_{k-1}...s_0$ and is denoted as $S_H S_L$, where $S_H = s_N s_{N-1}...s_k$ is the upper part and $S_L = s_{k-1}...s_0$ is the lower part. $c_{k-1}s_{k-1}s_{k-2}...s_0$ denotes the approximate sum of A_L and B_L. Here, c_{k-1} denotes the carry bit to the upper part.

In a simple truncation adder, $S_L = 0$ and $c_{k-1} = 0$. In this adder there is no circuit to calculate the lower part of the output sum, and the power consumption is only due to the upper part. Consequently, it has the lowest power consumption amongst all the approximate adders in the literature. In the approximate mirror adder 5 (AMA5) proposed in [3], the lower part of the result is set as $S_L = A_L$ and the carry is set as $c_{k-1} = b_{k-1}$. The AMA5 adder has higher power consumption than truncation adder, due to the toggles in the lower part that come from one of the inputs and also because of the carry propagating to the upper part of the sum. In the LOA approximate adder [4], $S_L = A_L$ OR B_L and $c_{k-1} = a_{k-1}$ AND b_{k-1}. The power consumed by LOA adder is more than that of truncation and AMA5 adders due to the OR gates used in computation of S_L and the AND gate for c_{k-1}. In the ETA approximate adder proposed in [5], the lower part of the inputs are added from left to right until the point at which both input bits are logic 1. Beyond this point all the sum bits are set to logic 1. Here $c_{k-1} = 0$. The hardware required and hence the power consumed is higher than the LOA adder because it needs extra gates for the detection logic and setting of the sum bit to the appropriate value. In InXA2 adder proposed in [6], the i-th bit of the lower part sum is set as $s_i = (a_i$ XOR $b_i)$ OR c_{i-1}, where c_{i-1} is the carry obtained on adding the i LSBs of the two inputs. This adder has a relatively large power consumption owing to the hardware needed for computation.

B. Accuracy Metrics

To measure the accuracy and quality of approximate arithmetic circuits, the metrics proposed in the literature are Mean Error Distance (MED), Normalized Mean Error Distance (NMED and NED) and Mean Relative Error Distance (MRED) [8], [10], [11].

MED is the average absolute error between the accurate and approximate outputs. NMED is normalization of MED by $2^{N+1} - 2$, which is the maximum sum possible in an N-bit adder. It is an indication of the significance of the MED in adders of various lengths. NED is obtained by normalizing the MED by 2^k and is an indication of how rapidly the MED grows with every additional approximate bit. MRED is a relative error metric and is an indicator of the percentage error across all values of the sum. The ranking of adders in terms of the MRED and NMED show the same trend [9]. As NMED is just the MED divided by a constant value, optimization with respect to either MED or NMED will give the same result.

Since the primary error metric is the MED, we use it for further analysis and optimization.

III. Proposed Median Approximate Adder (MA)

Our focus in this paper is to try and minimise the MED, while aiming to achieve the low power consumption of the truncation adder. To do this, we need to have fixed values of S_L and c_{k-1} so that, like the truncation adder, there is no hardware to find the lower part of the output. The values of S_L and c_{k-1} are to be chosen such that the MED is minimized.

For purposes of analysis, in this section, A_H, A_L, B_H and B_L refer to the corresponding decimal representation. In all cases, we assume that both inputs have N bits and k bits of the sum are approximated. The accurate sum therefore, is given by $(A_H + B_H)2^k + (A_L + B_L)$.

Assume that $Z = A_L + B_L$ is approximated by a fixed value L. Therefore, the approximate sum is $(A_H + B_H)2^k + L$. The error distance (ED), which is the absolute value of the error, is therefore given by $|Z - L|$. The goal is to find L so that $E\{|Z - L|\}$ is minimised. The solution to this minimization problem is well known, namely, the value of L that minimises $E\{|Z - L|\}$ is the median of the distribution of Z [12]. For a discrete random variable, the median is defined as follows.
Definition: The median of the random variable Z is defined to be any number M_Z that satisfies the relationship $P(Z \leq M_Z) \geq 1/2$ and $P(Z \geq M_Z) \geq 1/2$.
Hence $L = M_Z$. However, the value of M_Z depends on the probability distribution (PMF) of Z. We now consider various cases of input distributions.

A. Uniformly distributed inputs

This is the most common assumption for the PMF of the inputs. Assume that A and B have a uniform PMF with values in the range $[0, 2^N - 1]$. It is obvious that A_L and B_L will also have a uniform PMF with values within a range $[0, 2^k - 1]$. Since the PMF of $Z = A_L + B_L$ is the convolution of the PMFs of A_L and B_L, it is a triangular distribution with values between 0 and $2^{k+1} - 2$. Since it is a symmetric PMF, the median value is the midway point, namely, $2^k - 1$. If the binary representation of L is $c_{k-1}s_{k-1}s_{k-2}\cdots s_0$, then $c_{k-1} = 0$ and $s_{k-1}s_{k-2}\cdots s_0 = 11\cdots 1$.

B. Inputs have a symmetric distribution

Here, it is assumed that the PMFs of both inputs A and B are symmetric, i.e. $P(A = Q) = P(A = 2^N - 1 - Q)$ and $P(B = Q) = P(B = 2^N - 1 - Q)$, where $0 <= Q <= 2^N - 1$. In such a setting, we claim that the distribution of A_L and B_L are also symmetric, i.e. $P(A_L = Q) = P(A_L = 2^k - 1 - Q)$ and $P(B_L = Q) = P(B_L = 2^k - 1 - Q)$, where $0 <= Q <= 2^k - 1$. A proof for this claim is as follows.

Divide the range $[0, 2^N - 1]$ into 2^{N-k} bins, each of size 2^k. The i-th bin is given by $[i2^k, (i + 1)2^k - 1]$, where $0 \leq i \leq 2^{N-k} - 1$. In each bin, the most significant $N - k$ bits have a fixed value. For $0 \leq Q \leq 2^k - 1$, we have

$$P(A_L = Q) = \sum_{i=0}^{2^{N-k}-1} P(A = 2^k i + Q).$$

Hence,

$$P(A_L = 2^k - 1 - Q) = \sum_{i=0}^{2^{N-k}-1} P(A = 2^k i + 2^k - 1 - Q).$$

Since the PMF of A is symmetric, we have

$$\begin{aligned} & P(A_L = 2^k - 1 - Q) \\ &= \sum_{i=0}^{2^{N-k}-1} P(A = 2^N - 1 - (2^k i + 2^k - 1 - Q)) \\ &= \sum_{i=0}^{2^{N-k}-1} P(A = 2^k(2^{N-k} - 1 - i) + Q) = P(A_L = Q). \end{aligned}$$

The proof for the symmetry of B_L is same as the above.

If the PMFs of A_L and B_L are symmetric, the PMF of their sum (which is the convolution of the PMFs of the two inputs) is also symmetric. Since the range of Z is 0 to $2^{k+1} - 2$ and Z is symmetric, the median of Z is $2^k - 1$.

C. PMF of the inputs are arbitrary

In most of the applications seen in the literature, the number of approximate bits rarely exceeds $N/2$. As before, consider the division of the range $[0, 2^N - 1]$ into 2^{N-k} bins, each of size 2^k. For small k values, if the distribution of the 2^k values within each bin is approximately uniform, then the distributions of A_L and B_L are also approximately uniform. This means that the PMF of $Z = A_L + B_L$ is approximately triangular. Therefore, in the setting of small k, the median of Z is closely approximated by $2^k - 1$.

Since most applications are likely to satisfy one of the three cases, the proposed median adder (MA) uses $L = 2^k - 1$. However, if the PMFs of the inputs are known, it is possible to derive the PMFs of the lower k bits, which can then be convolved to obtain the PMF of the lower part sum. In this case, the median can be obtained exactly and L can be set to this value.

D. Accuracy metrics for Median adder for uniformly distributed inputs

An expression for MED of the median adder ($E\{|Z - L|\}$) with uniformly distributed inputs can be derived as follows.

$$\begin{aligned} MED &= \sum_{i=0}^{2^k-1} iP(|Z - L| = i) \\ &= 2\sum_{i=0}^{2^k-1} i\frac{2^k - i}{2^{2k}} = \frac{2^k}{3} - \frac{2^{-k}}{3}. \end{aligned} \tag{1}$$

The expressions for NMED and NED are obtained using suitable normalization factors as follows.

$$NMED = \frac{2^k - 2^{-k}}{3 \cdot (2^{N+1} - 2)} \text{ and } NED = \frac{1 - 2^{-2k}}{3}.$$

The expression for MED and NED of the truncation adder are $2^k - 1$ and $1 - 2^{-k}$ respectively. Clearly, MA is much better than truncation in terms of both metrics, while having the same power savings.

IV. Experimental results

In this section, we compare the proposed adder with other approximate adders in terms of power savings, MED and the peak signal-to-noise ratio (PSNR). All the approximate circuits are designed using Verilog and synthesized using Cadence Genus for 55nm technology. The synthesized netlist along with Standard Delay Format (SDF) file generated by Genus is simulated with 10^6 uniform random input pairs at a frequency of operation of an accurate ripple carry adder. A full adder's input pin capacitance is set as the output load capacitance.

TABLE I: Power saving per approximate bit of various adders.

Adder (A)	Trunc/MA	AMA5	LOA	ETA	InXA2
$P_{sb,A}$	1	0.88	0.72	0.57	0.01

A. Power savings and MED

Fig. 1a shows the variation in power savings normalized by the power of an accurate 1-bit full adder, as a function of the number of approximate bits (k) in various approximate adders. For k approximate bits, both the truncation and MA adders discard k full adders and set the lower part to a constant. Therefore, the normalized power savings for those two adders is k. For any other adder A, the normalized power savings can be roughly expressed as $P_{sb,A} \cdot k$, where $P_{sb,A}$ is the normalized power saving per approximate bit. Table I shows the value of $P_{sb,A}$ for various approximate adders. These values are obtained from the slopes of the curves in Fig. 1a. This is same as the power saving per bit described in [11]. We note that the value of $P_{sb,A}$ depends on the hardware complexity involved in the approximation of the lower part of the adder. Hence these values are low for ETA and InXA2 adders which have more hardware.

Fig. 1b shows the variation in MED with the number of approximate bits for all the adders. As seen from the figure, for all adders, $\log_2$ MED is approximately linear in k with a slope that approaches one as k becomes larger ($k \geq 3$). This means $MED \approx c\ 2^k$ in this range. Further, as seen from the figure, $c = 1$ for the truncation adder. For all other adders, $c < 1$. Therefore, for a given MED, the truncation adder has the least number of approximate bits. MA for example, can have $\log_2 3$ more approximate bits than truncation for the same MED. We also note that for a given k, MED of the MA is only marginally higher than the other inexact adders.

B. Power-accuracy tradeoff

The important trade-off in approximate hardware systems is the power-accuracy trade-off i.e. which of the adders meets an accuracy constraint with highest possible power saving. Using the plots in Figs. 1a and 1b, we obtain Fig. 1c, which is a plot of the normalized power savings as a function of $\log_2$(MED). For a considerable range of accuracies, the MA meets the accuracy constraint of a given MED with the highest power saving and hence offers the best trade-off between power and accuracy as compared to all the other approximate adders.

We analyse the power savings of all the approximate adders with repect to the truncation adder. For the truncation adder, $P_{sb,\text{trunc}} = 1$ and the number of approximate bits for a given MED is $k_{\text{trunc}} \approx \log_2 \text{MED}$. Therefore, the normalized power savings $P_{s,\text{trunc}} \approx \log_2 \text{MED}$. Let c_A denote the number of additional approximate bits possible for the same MED in other adders. Therefore, the power savings for a given MED

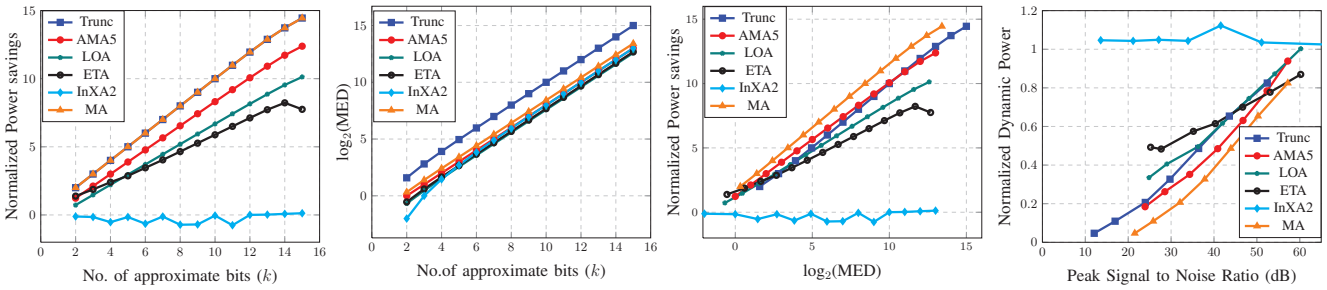

Fig. 1: (a)-(b) Variation in Normalized Power savings and Mean Error Distance with the number of approximate bits in various 16-bit approximate adders; (c) Normalized Power savings vs Mean Error Distance for 16-bit approximate adders; (d) Normalized Power vs PSNR for image addition using approximate adders, with number of approximate bits varying from 1 to 7.

can be written as $P_{s,\mathrm{A}} = P_{sb,A}(k_{\mathrm{trunc}} + c_A)$. For the MA, $P_{sb,MA} = 1$ and $c_{\mathrm{MA}} \approx \log_2 3$ as seen from equation (1). Therefore, MA always performs better than truncation. For other approximate adders,

$$\begin{aligned} P_{s,A} &= P_{sb,A} \cdot (k_{\mathrm{trunc}} + c_A) \\ &= P_{s,\mathrm{trunc}} + P_{sb,A} \cdot c_A - (1 - P_{sb,A}) \log_2 \mathrm{MED}. \end{aligned} \quad (2)$$

As seen in Table I, $P_{sb,A} < 1$ for all adders other than truncation and MA. As MED increases, the third term in (2) becomes greater than the second, so that $P_{s,A}$ becomes less than $P_{s,\mathrm{trunc}}$. Fig. 1c shows that truncation adder performs better than (a) ETA beyond $\log_2 \mathrm{MED} \approx 3$, (b) LOA beyond $\log_2 \mathrm{MED} \approx 4$ and (c) AMA5 beyond $\log_2 \mathrm{MED} \approx 10$. Although ETA and LOA have a larger c_A for a given MED, the difference is not large enough to result in a reduction of power. This is because both of them have a significant amount of hardware to compute the lower part sum, resulting in a lower $P_{sb,A}$. AMA5 effects a power saving compared to truncation for most cases because its $P_{sb,A}$ is high as seen in Table I. Since the MA adder has no additional hardware, the power savings due to the increase in c_A are fully realised.

Both the power savings and MED depend only upon the number of approximate bits k. So, for any adder size N, the power savings versus MED trade-off remains the same.

C. Application: Image addition

The approximate adders are used in a simple image processing application, namely image addition. The objective is to compare the power consumed by various adders for a given peak signal to noise ratio (PSNR). Two images (Cameraman and Rice) each of size 256x256 with each pixel represented using 8 bits are chosen as input images. Fig. 1d shows the dynamic power consumption as a function of PSNR for various approximate adders, with number of approximate bits varying from 1 to 7. From the figure, we see that MA once again gives the best trade-off between dynamic power and PSNR as compared to all the other approximate adders.

V. Conclusion

We have proposed the median adder, which approximates the lower k bits of the output to the fixed value $2^k - 1$. The median adder results in power consumption as low as that of a truncation adder, but provides significantly better accuracy. When compared to other similar adders, simulation results show that this adder gives the best trade-off between power and accuracy. We also used this adder in a simple image processing application and showed that the median adder gives better PSNR and power savings when compared to other similar adders in the literature.

References

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proceedings of the 18th IEEE European Test Symposium (ETS)*, 2013.

[2] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in *IEEE/ACM ICCAD*, pp. 667–673, 2011.

[3] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, 2013.

[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired AMA5 computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 57, pp. 850–862, 4 2010.

[5] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, pp. 1225–1229, 8 2010.

[6] H. A. F. Almurib, T. N. Kumar, and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in *Design, Automation and Test in Europe (DATE)*, 2016.

[7] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *Design, Automation and Test in Europe (DATE)*, 2011.

[8] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Design Automation Conference (DAC)*, 2012.

[9] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *Proc. of the Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 343–348, 2015.

[10] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Design, Automation and Test in Europe (DATE)*, 2014.

[11] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on Computers*, vol. 62, pp. 1760–1771, 9 2013.

[12] S. K. Bar-Lev, B. Boukai, and P. Enis, "On the mean squared error, the mean absolute error and the like," *Communications in Statistics - Theory and Methods*, vol. 28, no. 8, pp. 1813–1822, 1999.