Accounting for Systematic Errors in Approximate Computing

Martin Bruestel, Akash Kumar
Institute for Computer Engineering, Processor Design Group
and Center for Advancing Electronics Dresden (CfAED)
Technische Universität Dresden, Germany
Email: martin.bruestel@tu-dresden.de, akash.kumar@tu-dresden.de

***Abstract*—Approximate computing is gaining more and more attention as potential solution to the problem of increasing energy demand in computing. Several recent works focus on the application of deterministic approximate computing to arithmetic computations. Circuits for addition and multiplication are simplified, trading exactness for energy and/or speed. Recent approximation techniques for adders focus on modifications of individual full adders' truth tables or shortening carry chains. While the resulting error is usually characterized with statistical measures over the range of possible input/output combinations, the actual adder is a static nonlinear system regarding arithmetic operations and signal processing. The resulting unexpected effects present a challenge for adopting approximate computing as a widespread and standard application-level optimization technique. This paper focuses on the deterministic effects of approximate multi-bit adders, which are especially evident for certain input data in an otherwise well specified systems, showing the necessity to look beyond purely statistical measures. We show which fundamental principles are violated depending on the chosen approximation scheme, and how this choice affects practical applications. This can serve as a basis for designers to make informed decisions about the use of approximate adders at the application level.**

I. Introduction

Lately, the domain of *inexact computing*, or *approximate computing* received a lot of attention [1], [2] as a possible solution to the post-Dennard scaling challenges [3] in terms of area, power and performance of digital circuits. While some techniques focus on implementing hardware with *probabilistic* or *non-deterministic* inexact behavior [4], many approaches modify existing circuits in a way where the results are not exact, but perfectly reproducible. This paper focuses on the effects of errors generated by these *deterministic* approximate computations, specifically on the effects of addition, since it plays a fundamental role in virtually all relevant data processing applications.

Since approximate computing is a highly application-dependent technique, it is not obvious which kind of degradation can be accepted, and if so, to what degree. To address that, several metrics have been developed [5], [1] for comparing different implementations regarding their suitability for different applications. We show that difficulties arise when applying these measures blindly for increasing computation efficiency.

In their treatment of quantization noise, the authors of [6] acknowledged that traditional analyses depend upon the noise source being independent of the input. This is not the case for deterministic approximation techniques in general, where the error always depends on the input. Their focus on spectral characteristics also cannot describe the effects of nonlinear signal processing. As Sec. IV-B shows, spectral properties cannot completely describe all effects related to employing deterministic approximation.

Up until now, the metrics developed for approximate computing focus solely on statistical properties, like *Error Rate*, *Mean Error Distance*, *Peak Signal-to-Noise Ratio*. While these are relevant for many quality metrics, deterministic approximate arithmetic functional units are actually nonlinear static systems, leading to the violation of fundamental assumptions of arithmetic operations. This paper focuses on addition as the basis of all arithmetic calculations.

Figs. 1 (f) through (i) show an example where the simple approximate addition of a horizontal and a vertical gray-scale gradient image from Fig. 1 (a) and (b) generates unwanted patterns. Perhaps acceptable in terms of mean error energy, the human brain is specialized at recognizing patterns, rendering these image sums unacceptable. This is a problem of the combination of chosen approximate adder and input signal. For more "irregular" inputs, the result might be acceptable.

II. Recently Proposed Adder Designs

Since multi-bit adders are the fundamental building block of any signal processing system, focus has been on improving area, delay or performance by applying approximate computation. The proposed techniques we consider here fall into two categories:

1) Simplifying the individual full adders, thereby modifying the associated truth tables [7], [8], [9].
2) Breaking the carry chain to decrease critical path length [10]. This technique can be augmented with optional correction logic.

See [11] for a detailed comparison of recently proposed approximate adder implementations and their *PDP* (*Power-Delay-Product*) values.

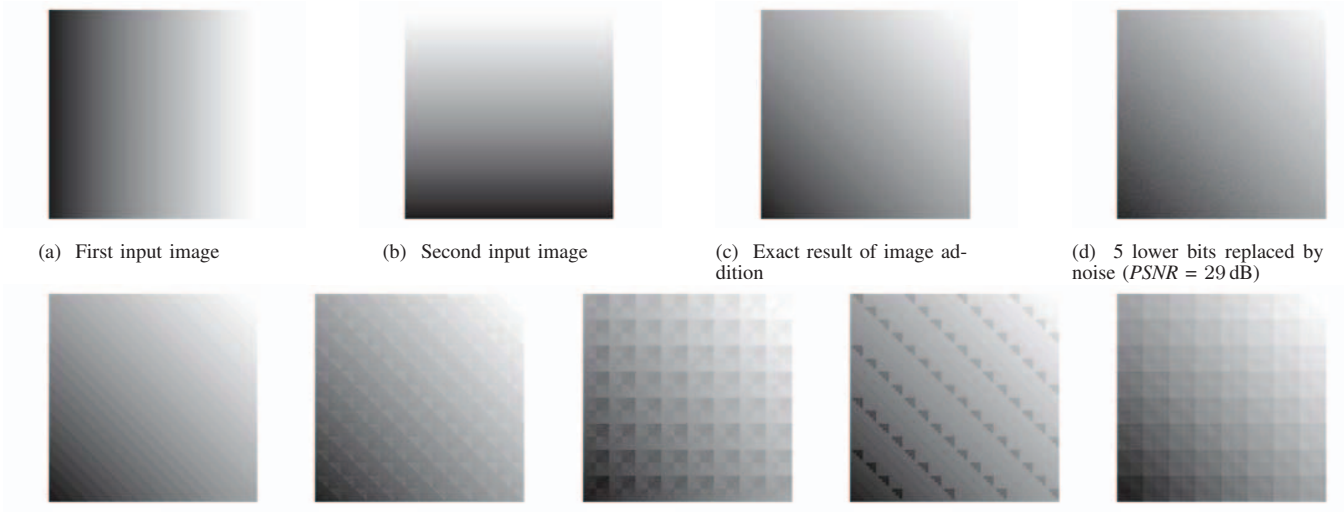
A. Truth-table modifying adders

This technique reduces circuit area and power of multi-bit adder circuits. A multi-bit adder is split into an exact and an inexact part, where the lower k significant bits are computed by approximate half- and full adders, and the higher significant $n-k$ bits by exact implementations. For these adders, all the relevant properties can be derived from the truth tables that result from the circuit simplifications. The truth tables used in this paper are reproduced in Table I. They are: *AXA1* from [7], *InXA1* - *InXA3* from [8], and *AMA1* from [9].

B. Adders with reduced carry chain length

This technique reduces the maximum carry chain length of multi-bit adders, decreasing critical path delay and allowing

978-3-9815370-8-6/17/$31.00



(a) First input image (b) Second input image (c) Exact result of image addition (d) 5 lower bits replaced by noise (*PSNR* = 29 dB)

(e) 5 lower bits truncated (*PSNR* = 29 dB) (f) 5 lower bits approx. by *InXA2* (*PSNR* = 33 dB) (g) 5 lower bits approx. by *InXA1* (*PSNR* = 27 dB) (h) approximated by *GeAr1* (*PSNR* = 28 dB) (i) 5 lower bits approx. by *InXA3* (*PSNR* = 33 dB)

Fig. 1. Image addition problem: the two 8-bit 256 by 256 gray-scale input images are added to produce the resulting image. Noisy addition, truncating bits, and using approximate adders produces visible artifacts.

faster circuit operation. One prominent example is *GeAr* [10], which provides a general configurable model for overlapping carry chain splitting adders. Here we cannot consider the individual truth tables because more than one adder is involved in a particular result bit. Instead, full range simulations of the specific configurations must be performed. For our simulations, we used one of the configurations that are proposed in [10] adapted to 8-bit values. The parameters are $N = 8$, $R = 2$, and $P = 2$, which represents 3 overlapped 4-bit adders with 2 bits of extra carry prediction. We refer to this configuration as *GeAr1*.

III. Consequence of violating basic addition properties

Depending on the specific chosen adder, following fundamental properties are violated by some, but not all, specific approximate adder implementations.

- A. Identity relation with neutral element: $x + 0 = x$
- B. Commutativity: $x + y = y + x$
- C. Associativity: $x + (y + z) = (x + y) + z$

Some of these characteristics can be checked by inspecting the modified truth table of the underlying full-adders. Using AMA1 from [9], e.g., will result in $1 + 0 = 0$, and $0 + 1 = 2$, violating properties A and B at the same time.

On the level of signal operations, other important characteristics are not guaranteed anymore:

- D. $E[x + y] = 0$ if $E[x] = 0$ and $E[y] = 0$
- E. Linearity

where E stands for the expected value, and can more informally be seen as the average value of a signal.

A. Identity Relation

Fig. 2 shows the outputs of different approximate 8-bit adders when one input is held at zero, which can be considered the transfer function of the static system given by $f(x) = x+0$.

Expecting this relation to hold for all adders, Fig. 2 shows that of the three chosen adders only *InXA3* exhibits correct behavior. We call this property *zero-input correctness*, which is important for systems employing negative feedback, typical for control systems, where a stable point is found by subtracting two signals and feeding back the resulting error. If the system responds incorrectly with non-zero values, stability cannot be reached.

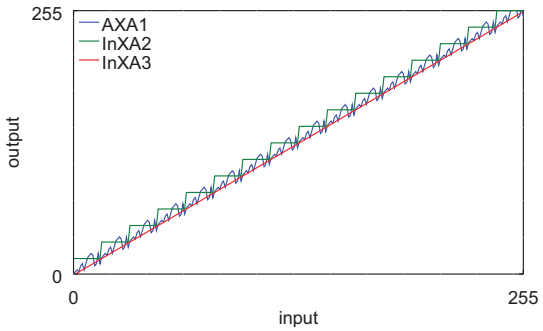


Fig. 2. Transfer Functions of different 8-bit inexact adders, where 4 bits are approximated, input 2 held at zero. In this example, *InXA3* exhibits correct behavior.

B. Commutativity

Adders like AMA1 [9] contain asymmetrical entries in the truth tables which always lead to violation of commutativity: If even the lowest result bit is not independent of addend order, the whole addition is non-commutative, which leads to grave restrictions for design automation tools: A synthesis tool cannot assume the two inputs to be equal, which reduces the degrees of freedom in layout optimization. To satisfy commutativity, following properties of a one-bit full-adder must hold:

1) $Sum(X, Y, C_{\text{in}}) = Sum(Y, X, C_{\text{in}})$
2) $C_{\text{out}}(X, Y, C_{\text{in}}) = C_{\text{out}}(Y, X, C_{\text{in}})$

for $C_{\text{in}} = [0, 1], X = 1, Y = 0$

Of the adders from Table I, only AMA1 violates commutativity.

TABLE I. Truth tables of recently proposed adder designs. Bold entries deviate from the correct truth table.

X	Y	C_{in}	Sum	C_{out}	Sum (AXA1)	C_{out} (AXA1)	Sum (InXA1)	C_{out} (InXA1)	Sum (InXA2)	C_{out} (InXA2)	Sum (InXA3)	C_{out} (InXA3)	Sum (AMA1)	C_{out} (AMA1)
0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0
0	0	1	1	0	1	0	1	**1**	1	0	1	0	1	0
0	1	0	1	0	**0**	**1**	1	0	1	0	1	0	**0**	**1**
0	1	1	0	1	**1**	**0**	0	1	**1**	1	0	1	0	1
1	0	0	1	0	**0**	**1**	1	0	1	0	1	0	**0**	0
1	0	1	0	1	**1**	**0**	0	1	**1**	1	0	1	0	1
1	1	0	0	1	0	1	0	**0**	0	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	**0**	1	1	1

C. Associativity

Associativity, is guaranteed by: $(x+y)+z = x+(y+z)$. This is important whenever operations are going to be reordered. Compilers frequently reorder the sequence of operations, both for sequential computer programs and during hardware synthesis. Different orderings will then output different sequences for the same input sequence of values. Thus, application-dependent resiliency must now be taken into account at one of the last steps in the design flow. All deterministic approximate adders violate associativity in the general sense.

D. Non-Zero Expectation Value

A different effect also follows from the aforementioned asymmetry: If positive and negative errors generated over all different input combination do not cancel each other out, any random input signal will lead to a non-zero expectation value in the output, producing a DC component. (See Fig. 3).

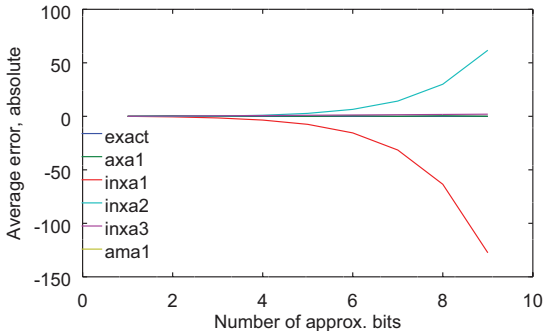


Fig. 3. DC component generated over all input codes for 8-bit numbers, showing the increase in average error for increasing number of approximate bits for different implementations.

This shows that e.g. *AMA1*, while violating commutativity, produces no DC component when applied to random input. We performed experiments where the addition of two sinusoidal signals led to the generation of a DC component in addition to the approximation error. While the approximation error manifests as quasi-random high frequency distortions, the DC component may be a real problem in signal processing systems that would normally not generate a DC component. Additional effort might be required to filter out this component, which competes with the original goal of reduced effort computation.

E. Linearity

Linear systems respond to any sum of inputs with the sum of the individual inputs' response. This is known as the *superposition principle*. Generally, an adder is considered to be a linear system, such that:

$f(a*X+b*y) = a*f(X)+b*f(Y)$

While no real adder can satisfy this property completely due to quantization effects, approximate adders reduce the threshold at which this becomes evident in applications. As an example, repeat the addition of all possible inputs to a constant value for the *GeAr1* adder. In contrast to Sec. III-A the constant input is not 0 but 255. For this experiment the value 255 is the worst case input, guaranteed to trigger the maximum error resulting from the simplified carry chain. The resulting input-output relation is shown in Fig. 4. All considered approximate adders are nonlinear, since the only way to obtain correct results for all input codes is to perform an exact addition.

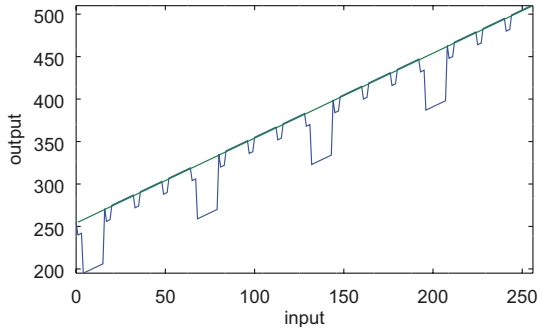


Fig. 4. Output of adding the constant value 255 to the input range (0 to 255) of the *GeAr1* adder. For reference, the straight line shows the exact result.

F. Discussion

The effects of approximating addition are similar to the general problem of limited precision for computing in general. However, while usually values are quantized with sufficient dynamic range to abstract away from these problems, introducing deterministic errors in the lower significant bits shifts these effects from the noise range into the signal range. While *probabilistic* approximate adders add uncorrelated error into the system – manifesting as a 6 dB per bit *Signal-to-Noise Ratio* decrease – the use of *deterministic* approximate adders requires careful consideration of all effects associated with nonlinear systems. This does not preclude the fact that for a certain combination of approximate systems and input signals, the generated error does indeed exhibit noise-like behavior.

IV. Comparison of Deterministic Approximation, Truncation, Noise

A. Image Addition

To compare the "acceptability" subjectively, Figs. 1 (d) - (i) show different image addition results with modifications in the lower 5 bits of the result, except for Fig. 1(h), for which the *GeAr1* configuration was again used. Of all the results the *InXA* results were chosen because the *InXA1* image sum has a similar *PSNR* value as the noise and truncation sum with 5 bits of approximation. Choosing a horizontal and a vertical gray-scale gradient as inputs ensures using the whole operational range of the adder.

Fig. 1 shows that statistical measures fail to capture application-specific error criticality. While the images in Figs. 1 (d), (e), (g), (h) exhibit similar *PSNR* values, the result of the noisy addition is most similar to the exact result, and in print almost indistinguishable. Figs. 1 (c), (f) have worse subjective quality than Fig. 1(d), despite having higher PSNR values, typically indicating *better* quality. Truncation is perceived as "poor quality", while the approximate versions generate visible artifacts. In the context of images, the generated systematic error competes with the image content, while the random error is generated in the "noise domain" of the application. We did not consider any more sophisticated image quality measures, and only some example adders. The important point here is that the generation of errors happens in different "domains" for different quality reduction techniques. The noisy image is easily accepted by a viewer, while for the *GeAr1* result the artifacts may be unacceptable.

B. IIR Filtering

As a second example we chose a second order digital high-pass filter, a fundamental block in signal processing applications. Because it is a feedback system, substitution of linear for nonlinear operations is expected to lead to stability problems.

We ran simulations on all the truth-table modifying adders, where we applied a 2nd high pass filter (Direct Form I) using the different approximate adders for the summing nodes. The multiplication was assumed to be exact, input and coefficients were quantized to 8 bits. We used dirac impulses, unit steps and mixtures thereof as input signals. In the response, we looked for two things: change of the form of impulse/step response and stability. The resulting approximate filters are nonlinear systems, thus spectral analysis and transfer functions generally cannot be used for characterization. We inferred (input-specific) stability from constant-input responses: if the signal does not remain constant, limit cycles exist. In an audio applications, these are prone to be audible.

In addition to the exact step response, Fig. 5 shows the responses for the following modifications of the lower 4 bits: truncated, noisy, and approximated. The chosen approximate responses are representative for all considered adders. Noisy addition had the expected effect of adding noise to the response, without influencing filter characteristics. Using truncated addition changed the filter characteristics by a comparatively small amount, while the response remain stable. *AXA1* changed the characteristics *and* introduced instability, while *InXA3* also changed characteristics, without introducing instability. While no relationship was observed between stability, number of approximated bits, input signal and approximation type, it demonstrates that using such components requires careful case-by-case analyses when employing approximation. Choosing a deterministic approximate with sufficient *SNR* can still introduce unwanted zero-input limit cycles.

V. Conclusion

We showed that using deterministic approximate adders can have unforeseen effects on the acceptability of an application result, not only the reduction of some quality measure. This cannot be captured only by the statistical metrics that have been proposed up to now, and is an obstacle for widespread adoption of approximate computing.

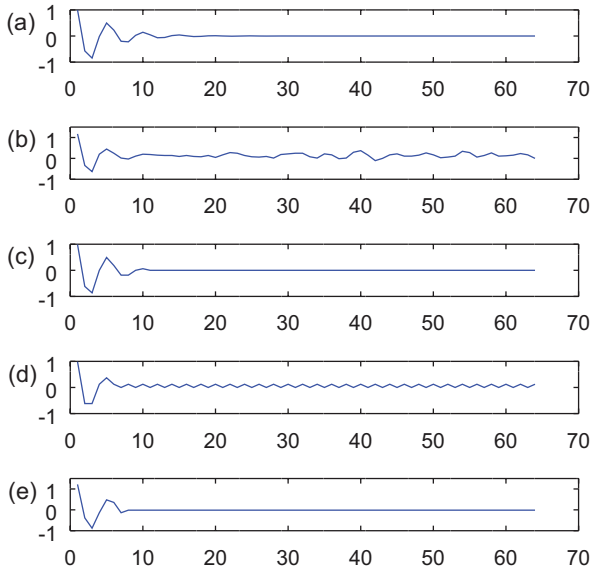


Fig. 5. Step responses of different adders inside the same second order IIR filter. (a) exact response, (b) noisy addition, (c) truncated addition, (d) *AXA1*-approximated, (e) *InXA3*-approximated. The shape of the initial sinusoidal peaks relates to the filter characteristics, while the flat region shows the steady-state behavior.

The susceptibility to these problems depends on the application and/or the input data. If it is known on which fundamental assumptions an application depends, it will be much easier to make the optimum choice of approximation technique. In the presented examples, the generation of patterns was a criteria for evaluating image addition, while stability was one of the defining properties of IIR filtering. None of these characteristics relate to any of the existing statistical measures. Error, resulting from deterministic approximation, is not guaranteed to exhibit noise-like behavior. Especially regular inputs may generate artifacts that compete with information in an application domain.

References

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *ETS*, 2013, pp. 1–6.

[2] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate Computing: A Survey," *IEEE Des. Test*, vol. 33, no. 1, pp. 8–22, 2016.

[3] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *ISCA*, Jun. 2011, pp. 365–376.

[4] K. Palem and A. Lingamneni, "What to Do About the End of Moore's Law, Probably!" in *DAC*, 2012, pp. 924–929.

[5] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1760–1771, 2013.

[6] B. Barrois, K. Parashar, and O. Sentieys, "Leveraging Power Spectral Density for Scalable System-Level Accuracy Evaluation," in *DATE*, 2016.

[7] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate XOR/XNOR-based adders for inexact computing," in *IEEE-NANO*, 2013, pp. 690–693.

[8] H. A. F. Almurib, T. N. Kumar, and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in *DATE*, 2016, pp. 660–665.

[9] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-Power Digital Signal Processing Using Approximate Adders," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[10] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A Low Latency Generic Accuracy Configurable Adder," in *DAC*, 2015, pp. 86:1–86:6.

[11] H. Jiang, J. Han, and F. Lombardi, "A Comparative Review and Evaluation of Approximate Adders," in *GLSVLSI*, 2015, pp. 343–348.