JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A ROUNDING-BASED APPROXIMATE MULTIPLIER FOR HIGH-SPEED ENERGY-EFFICIENT DIGITAL SIGNAL PROCESSING

K.PAWAN[1],Dr.B.NAGESHWARRAO[2] ,Dr.T.VAMSHI[3]

[1]M.Tech Student, Talla Padmavathi College of Engineering, Somidi, Kazipet, Telangana, 506003
[2]Assoc Professor, Talla Padmavathi College of Engineering Student, ,Somidi, Kazipet, Telangana, 506003
[3]Assoc Professor, Talla Padmavathi College of Engineering,Somidi, Kazipet, Telangana, 506003
[1]kandagatlapawan@gmail.com, *[2]nagesh.south@gmail.com, [3]vamshi22g@gmail.com*

Abstract

An energy-efficient and fast approximate multiplier is presented in this study. Operands are rounded up to the next exponent of two, which is what we're going for here. At the cost of a little inaccuracy, the computationally costly part of the multiplier is removed, which improves both speed and energy consumption. Multiplications that are signed or unsigned can both benefit from the new method. One for unsigned operations and two for signed operations are included in our three implementations of the approximation multiplier. Performance comparisons are carried out utilising different design parameters for the suggested multiplier in order to evaluate its efficiency. It is also examined in terms of image processing applications, such as sharpening and smoothing of images. In order to expand The RoBA multiplier is utilised in the FIR Filter convolution process.

Key words: Accuracy, approximate computing, energy efficient, error analysis, high speed, multiplier.

1. Introduction

Energy Almost all electronic systems, especially portable ones like smart phones, tablets, and other gadgets, have a primary design requirement of minimising complexity. Minimizing the performance (speed) penalty while yet achieving this minimization is greatly sought [1]. These portable devices use signal processing techniques (DSP) blocks in order to perform multimedia applications. The arithmetic logic unit is the heart of these blocks, where multiplications account for the majority of all arithmetic operations [2]. So boosting the performance and power/energy efficiency of multipliers is critical to increasing CPU efficiency.

Most of those DSP cores employ image processing and analysis algorithms that result in images or films that can be consumed by humans. Because of this, we can use approximations to speed up and save energy. This stems from the fact that humans have a restricted ability to perceive images and videos. Additionally, there are applications where the accuracy of the mathematical operations is not important to the system's functionality, such as video compression (see [3], [4]). To

make compromises between accuracy, speed, and power/energy usage, the designer can employ approximation computing.It is possible to apply the estimate to the arithmetic units at several design abstraction levels, such as the circuit, logic and architecture levels, along with algorithm and software layers. One or more of several strategies, such as the use of voltage over scaled or over clocking and function approximations methods (such as changing a circuit's Boolean function), may be used to approximate the data, such as tolerating some timing violations. [4], [5]] A number of approximate math building blocks, such like adders and multipliers, have been proposed in the domain of link prediction approaches.

A low-power, high-speed approximate multiplier for DSP applications that is also error-resistant is the emphasis of this study. Using rounded input values and changing the standard multiplication strategy at the algorithm level, the suggested approximative multiplier is both efficient and effective. Rounding-based approximate multiplier (RoBA) is what we name this. For both signed & unsigned multiplications, three optimal designs are shown in the suggested multiplication technique. By analyzing the delays, performance / energy consumptions, EDPs and areas of these structures with some approximated and accurate (exact) multipliers, the efficiency of these structures are evaluated. The following is a summary of the paper's contributions: We'll show you a novel approach to RoBA multiplication by changing the usual multiplication strategy, and then go over the proposed approximation multiplication scheme's hardware implementation in three different architectures for sign and unsigned computations. (2).

2. Literature survey

2.1 Low-power digital signal processing using approximate adders.

Portable multimedia devices utilising a variety of signal processing methods and designs necessitate low power consumption. Humans can usually glean important information from somewhat incorrect outputs in most multimedia programmes. So we don't have to worry about producing numerical outputs that are perfectly accurate. Previous studies in this area have used algorithmic and architectural strategies to reduce the faults that come from voltage overscaling in order to utilise error resiliency. Logic minimising wastage at the circuit level is proposed in this study as an alternate way to take benefit of the relaxation in numerical accuracy. By providing a variety of inaccurate or approximated full adder units with lower latency at the transistor level, we show this principle and then use them to develop approximate multi-bit adders. To further enable voltage scaling, our solutions result in shorter critical routes, which reduce switching capacitance. In order to show the effectiveness of our method, we construct and test video / data compression algorithm architectures based on the suggested approximation arithmetic units. These approximate adders also have basic mathematical models for inaccuracy and power consumption. These approximate adders are also shown to be useful in the following dsp designs (discrete cosine transform and finite impulse response filter) under special quality restrictions, as shown in the following examples The suggested approximate adders can save up to 69 percent of the power compared to current implementations employing precise adders, according to simulation data.

2.2 Energy-efficient approximate multiplication for digital signal processing and classification applications.

There has been an increasing demand for energy-constrained devices to enable a variety of DSP and classification applications. Fixed-point arithmetic is commonly used in these applications to do huge matrix multiplications, but they also tolerate small amounts of computational error. As a result, multiplication efficiency must be improved. Here, we provide a set of multiplier topologies that, at design time, may balance computational precision with energy consumption. It consumes 58% fewer energy/op compared to a precise multiplier with an average computing error of 1%. Finally, we showed that

quality of DSP and also the reliability of text categorization are unaffected by a tiny computational error.University of Wisconsin-Electrical Madison's and Computer Engineering Department, in Madison, Wisconsin HADI ASGHRI MOGHADAM Wisconsin-Madison Department of Electronics And communication Department, Madison. An aerospace engineering department at the University of Wisconsin–Madison Department of Communications and Information Technology, Daegu Gyeongbuk Institute of Science and Technology UW-Electrical Madison's and Computer Engineering Department has Nam Sung Kim as an instructor.

3.1 Background and motivation

When it comes to the fundamentals of multiplication, it's merely adding an integers to itself a certain number of times. To arrive at a result, a number (multiplicand) is multiplied by yet another number (multiplier) (product). There are a number of ways that pupils learn to multiply at elementary school. It's then amplified by each of the multiplier's digits, starting with the Least Significant Digit on the right (LSD). The error values (partial products) are stacked one on top of the other and offset by one digit in order to align numerals of the same weight. The total of all the sub-products yields the final product. Despite the fact that many people think of multiplying in base 10, this approach can be used to any base, even binary. Flow of information for the fundamental multiplication method is shown in Figure 1.1. One number is represented by each black dot.

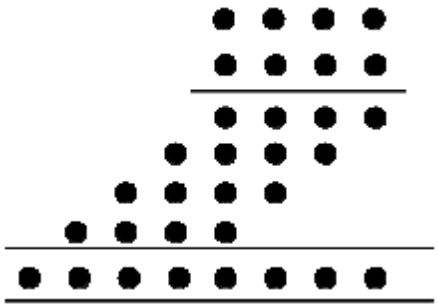


Figure 1: basic Multiplication

We presume that MSB stands for the digit's sign in this example. In digital electronics, multiplication is a straightforward process. The conventional algorithm for multiplying two binary values is where it got its start. The product of binary factors is calculated using the addition + shift left operations in this algorithm. The algorithm depicted in figure 3.2 can be deriving from the foregoing procedure for any type of multiplication.

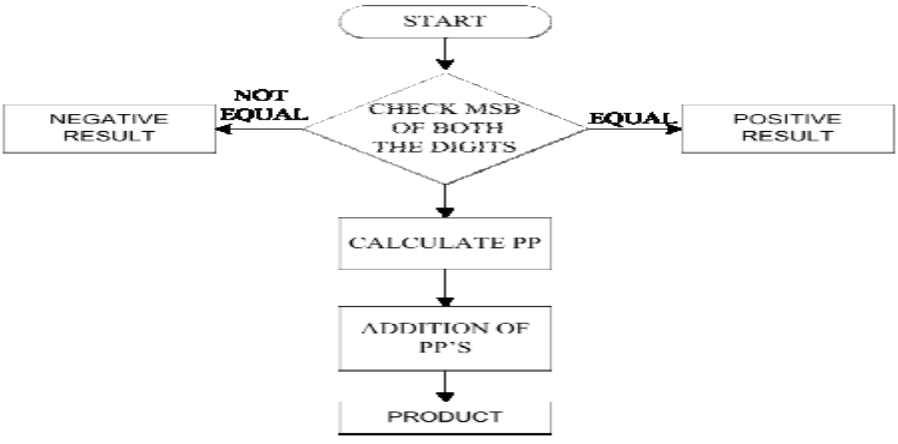


Figure.2 Signed Multiplication Algorithm

3.2 Binary Multiplication

The set [0, 1] is the only set of digits available in the binary numbers. If you multiply a binary code by a binary blob bit, the outcome is either 0 or the original binary value. As a result, the intermediate partial-products can be formed quickly and easily. It takes a lot of effort to add up these partial products for binary multipliers. Creating the input bits one at a moment and summating them since they are formed is a sensible technique. Because it takes at least one additional machine cycle to sum each successive partial-product, this technique is often done by programming on processors without a hardware multiplier. Direct hardware implementation of multipliers is an option if this solution isn't fast enough for your application. Signed and unsigned integers are the two major types of binary multiplication. Bit shifts and bit additions combine the two numbers, multiplicand and multiplier, to get a digit multiplication result. Unsigned multiplication requires the addition of up to n shifting copies of the multiplicand, using the bit representations of x = xn-1.....x1 x0 and y = yn-1.....y1y0. Part of the product is generated, part of the product is reduced, and then the final addition is made.

4. Proposed system

4.1 Introduction to Approximate multiplier.

Countless low-power gadgets, including research and commercial computers of all sizes and types, will soon produce an unprecedented amount of fresh data, and we are only at the cusp of a massive data flood. There is an expansion in the computing footprint of a variety of applications that try to obtain deep insight from enormous amounts of organised and unstructured data, while traditional workloads like transactional & databases processing continue to increase slightly Traditional computing implies an accuracy that is not required for the processing of the majority of these data types. Despite this, cognitive applications are still being run on special purpose (and accelerator) systems that are extremely precise and built with reliability in mind. With the goal of achieving considerable increases in computer throughput while still keeping acceptable quality outcomes, approximate computing attempts to loosen these restrictions.

Approximate computing research aims to determine what levels of approximation are possible in the multiple elements of the structure stack (from algorithms to circuits & semi-conductor devices) so the social competence are reasonable, albeit potentially different from those procured using precise computation. Some academics have focused on improving just one layer of something like the system stack, which has resulted in improvements in power or processing conditions. A primary goal of this study was to determine if the benefits of combining several approximation approaches covering more than one tier of the technology stack might be generalised to other application domains.

Our demonstration focused on three types of approximation: skipping calculations, approximations of mathematical calculations themself, and approximate solution of interaction between computational units themselves. We examined loop perforation, lower arithmetic precision, and synchronisation relaxation on behalf of each category. Despite their high computational costs, the applications we chose have the opportunity to have a big impact on our daily lives if they become more widely available and affordable. Dsp, automation, and machine learning were all represented in our work. It was possible to reduce execution time by 50% on average while maintaining acceptable quality of outcomes for all the investigated apps,according to our findings across all the examined applications. Aside from these advantages, we were also able to reduce computations to 10-16 bits instead of 32 or even 64, which might have a substantial impact on performance and energy consumption. We were able to lower the execution time of the parallel programs we studied by 50% by eliminating some of the synchronisation overheads.Finally, our findings show that when these strategies are used together, the benefits are even greater. Thus, the many approaches do not dramatically diminish one another's efficacy when used together. Rethinking the special purpose processor architecture is necessary because the benefits of approximated

computing extend beyond a narrow set of applications, and thus motivates a rethinking of the multi purpose processor.

4.2 Multiplication Algorithm of RoBA Multiplier

. An approximation multiplier is based on the fact that when the integers are 2 to the power n, it is much easier to operate on them (2n). To better understand how the approximation multiplier works, let's use Ar and Br to symbolise the rounded input numbers of A and B. It is possible to rewrite the multiplication A by B as

$$A \times B = (Ar - A) \times (Br - B) + Ar \times B + Br \times A - Ar \times Br \ldots\ldots(1)$$

The shift operation can be used to perform the multiplication operation of Ar Br, Ar B, and Br A. This equation has a complicated hardware implementation because of the fact that (Ar A) = (Br B). There is normally a minimal impact on the final outcome, which is based on changes in rounded values, from the magnitude of this phrase. As a result, we believe that removing this portion of (1) will make multiplication easier. As a result, the following formula is employed to multiply:

$$A \times B \sim= Ar \times B + Br \times A - Ar \times Br \ldots\ldots\ldots\ldots(2)$$

In other words, three shifts and two addition/subtraction procedures are all that are required to multiply. The 2n-closest values of A and B should be found using this strategy. 2p and 2p1 are the two nearest values for A (or B) when the number of A (or B) is equivalent to the 3 2p2 (where p is indeed an unspecified positive number larger than one). If p is greater than 2, the suggested multiplier's accuracy will be the same, but if p is greater than 2, the proposed multiplier's hardware implementation will be smaller, therefore it is selected in this research. Rounding up and down can be simplified by using 3 p 2 2 as a rounding factor, and simpler logic expressions can be produced if they are employed in this manner..

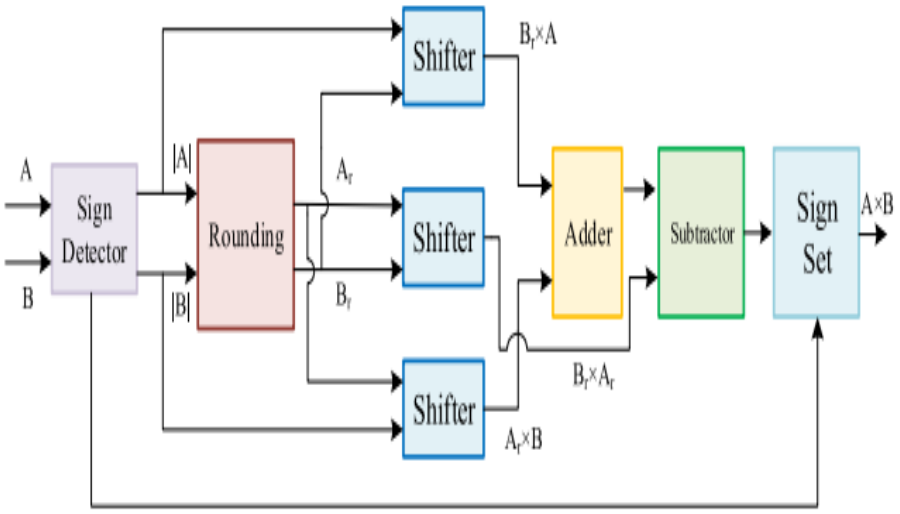


Fig. 3. Block diagram for the hardware implementation of the proposed multiplier

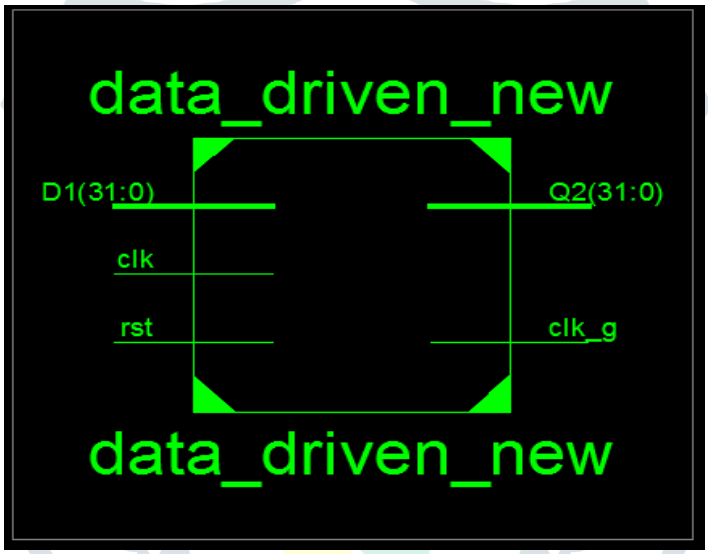
The lone exception is the number three, for which the new agreements multiplier uses two as the closest possible value. The RoBA multiplier's ultimate output may be bigger or smaller than with the precise result dependent on the amplitudes of Ar & Br compared to the magnitudes of A and B, correspondingly. You should keep in mind that, if one of the variable is less

than its rounded value and the other operand is greater than its adjusted value, then the estimated result will be greater than the size result.. As a result of (Ar A) (Br B) having a negative multiplication result, this is the case. Given that (1) and (2) are not equal, the approximate result exceeds the exact value by a factor of 2. Similarly, the approximated result will indeed be lower than the optimal solution when both A & B are larger or smaller than Ar & Br. RoBA multiplier's advantage only exists for positive inputs since in the two-complement representation of rounded values of inverting input are not 2n. That's why it's recommended to first identify both inputs' absolute values and the output sign based on their signs, and then perform the multiplication operation on unsigned numbers before applying an appropriate sign to it at its end. Next, the physical implementation approximate multiplier is discussed.

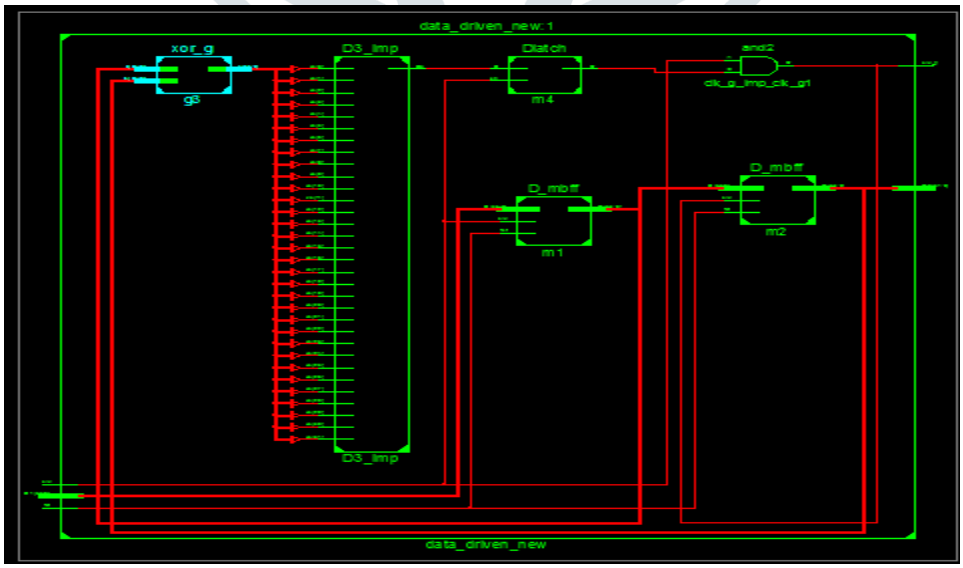
.

5. Results

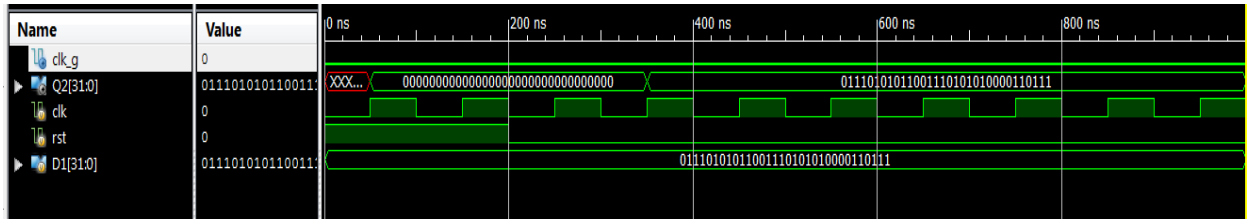
Entity diagram:



RTL schematic:



Simulation results:



6. Conclusion

This work proposes a high-speed, low-energy approximate multiplier named RoBA. Based on 2n rounding of the inputs, the proposed multiplier had good accuracy. It was possible to reduce speed and energy usage by skipping the computationally costly phase of multiplication in exchange for a slight inaccuracy. Signed & unsigned multiplications could both benefit from the method provided here. One for unsigned and two for signed operations were explored among three embedded applications of the approximation multiplier. The suggested multipliers were compared against precise and approximate multipliers with varied design parameters in order to determine their efficiency. The RoBA multiplication architectures were found to outperform the related approximation (exact) multipliers in the vast majority of instances. Both sharpening and smoothing applications in image processing were tested to see how well the proposed approximate multiplier method worked. The results showed that the images had the same visual quality as those produced by mathematically correct multiplication methods.

7. References

[1] M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2012.

[2] 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.

[3] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[4] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. Int. Conf. Comput.-Aided Design, Nov. 2011, pp. 667–673.

[5] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, "New approximate multiplier for low power digital signal processing," in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADS), Oct. 2013, pp. 25–30.

[6] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in Proc. 24th Int. Conf. VLSI Design, Jan. 2011, pp. 346–351.

[7] D. R. Kelly, B. J. Phillips, and S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf. Design Archit. Signal Image Process., 2009, pp. 97–104.

[8] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.

[9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[10] K. Bhardwaj and P. S. Mane, "ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-

chip," in Proc. 8th Int. Workshop Reconfigurable Commun.-Centric Syst.-Chip, 2013, pp. 1–6.

[11] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient approximate wallace tree multiplier for error-resilient systems," in Proc. 15th Int. Symp. Quality Electron. Design (ISQED), 2014, pp. 263–269.

[12] J. N. Mitchell, "Computer multiplication and division using binary logarithms," IRE Trans. Electron. Comput., vol. EC-11, no. 4, pp. 512–517, Aug. 1962.

[13] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," IEEE Trans. Comput., vol. 55, no. 12, pp. 1523–1535, Dec. 2006.

[14] Nangate 45nm Open Cell Library, accessed on 2010. [Online]. Available: http://www.nangate.com/

[15] H. R. Myler and A. R. Weeks, The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.

[16] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[17] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 418–425.

[18] C.-H. Lin and I.-C. Lin, "High accuracy approximate multiplier with error correction," in Proc. 31st Int. Conf. Comput. Design (ICCD), 2013, pp. 33–38.

[19] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proc. 49th Design Autom. Conf. (DAC), Jun. 2012, pp. 820–825.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[21] 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, Sep. 2013.