On the Implementation of Multi-bit Inexact Adder Cells and Application towards Image De-noising

Srikant Kumar Beura, Amol Arjun Jawale, Bishnulatpam Pushpa Devi, and Prabir Saha

Abstract—Inexact computing is an attractive concept for digital signal processing at the submicron regime. This paper proposes 2-bit inexact adder cell and further escalate to 4-bit, and 8-bit inexact adder and error metrics have been evaluated mathematically for such adder cells. The approximated design has been proposed through the simplification of the K-Maps, which leads to a substantial reduction in the propagation delay as well as energy consumption. The proposed design has been verified through the Cadence Spectre and performance parameters (such as delay, power consumption) have been evaluated through CMOS gpdk45 nm technology. Furthermore, the proposed design has been applied to image de-noising application where the performance of the images like Peak Signal to Noise Ratio (PSNR), Normalized Correlation Coefficient (NCC) and Structural Similarity Index (SSIM) has been analyzed through MATLAB, which offer the substantial improvement from its counterpart.

Index Terms—Delay, error metrics, image de-noising, inexact adder.

Original Research Paper
DOI: 10.7251/ELS2024033B

I. INTRODUCTION

Inexact computing emerged as a promising theory to reduce the net energy consumption of integrated circuits (IC) with a certain amount of accuracy [1]-[2]. However, the stringent accuracy is not required for the applications like image processing, stochastic signal processing, digital modulation technique such as delta modulation, etc., where propagation delay, transistor count, and power consumptions are the premier concern for the researchers in the present era [3]-[6]. From the last decade, a versatile amount of work [7]-[11] have been reported so far to show the improvements of the performance metrics along-with the accuracy enhancement in signal processing, requires the circuit component like delta modulation circuitry, which is an important encoding technique in digital signal processing, involves a trade-off between accuracy and power, can also be realized through the multiplier and adder as a block. For the digital implementation of such circuitry researchers are utilizing OR gate instead of XOR gate to achieve the efficient circuitry as a result the accuracy has been negotiated [7]-[9].

Moreover, digital signal processors are widely used to process the image and video information. Hence, high speed low power circuits with utmost accuracy for such a processor is the premier concern for the researchers in the present era [10]-[11]. From the last decade, a versatile amount of work [12]-[19] have been reported so far to show the improvements of the performance metrics along-with the accuracy enhancement. A research group headed by Zhu proposed an error-tolerant adder (ETA) in 2010 [12], and in the same year, lower part OF adder (LOA) has been proposed by Mahdiani et al. [13]. Moreover, the above-mentioned adders [7]-[8] have been applied to the FFT processor and soft computing applications respectively. However, both the applications are the error tolerant circuits; thus, error matrices would not be affected much, which are caused by the aforementioned adders. In 201, Gupta et al. [14] has designed a low power digital signal processing methodology, where they have proposed five approximate adder cells (AMA). However, all the approximations [9] use a large number of transistors leading to increase the circuit complexity.

Meanwhile, the researcher has optimized the transistor count for the design of an inexact adder (AXA) [15] through pass transistor logic, which also has been derived from ten transistors (10T) precise adder [20]. Without affecting the logic (pass transistor), another three different adder cells have been proposed by Almurib et al. [16] through the alteration of basic gates (InXA). Due to the application of the pass transistor logic transistor count and power consumption has been reduced but does not attain the other premiere parameters like accuracy and voltage swing. Therefore, to recover this voltage swing issue, pass transistor logic-based inexact adder has been proposed in the literature [17], but still, the accuracy has not been resolved so far. Dutt et al. have proposed four approximate full adder
Inexact adders can be designed in two methodologies based on the input length [21], which are inexact 1-bit adder and inexact multi-bit adder. The first methodology leads to less area, reduced power consumption, and small delay at the cost of diminution in accuracy. However, the second design methodology doesn’t rely on an approximation of a single bit, which allows the researchers to approximate a portion of an n-bit adder. This results in a significantly low error as compared to the adders designed by the first methodology [19]. The sum output of such block (multi-bit) is calculated by targeting the carry speculation mechanism. This design is based on the idea that when two random inputs are added, it rarely generates a long carry chain. Thereby, multi-bit approximated adders are the usual choice in the present era for the researches.

II. PROPOSED INEXACT ADDER ARCHITECTURES

Inexact adders can be designed in two methodologies based on the input length [21], which are inexact 1-bit adder and inexact multi-bit adder. The first methodology leads to less area, reduced power consumption, and small delay at the cost of diminution in accuracy. However, the second design methodology doesn’t rely on an approximation of a single bit, which allows the researchers to approximate a portion of an n-bit adder. This results in a significantly low error as compared to the adders designed by the first methodology [19]. The sum output of such block (multi-bit) is calculated by targeting the carry speculation mechanism. This design is based on the idea that when two random inputs are added, it rarely generates a long carry chain. Thereby, multi-bit approximated adders are the usual choice in the present era for the researches.

II.1 2-Bit Exact Adder

The 2-bit exact adder cell is designed by cascading two full adders for generating the outputs [11]. This design takes 5 input combinations \((A1, A0, B1, B0, Cin)\) and generates 3 outputs i.e. \(Cout1, Sum1,\) and \(Sum0\) which are given by eqn. 1, 2 and 3 respectively.

\[
\begin{align*}
Sum0 &= A0 \oplus B0 \oplus Cin \\
Sum1 &= A1 \oplus B1 \oplus Cout0 \\
Cout1 &= A1 \cdot B1 + (A1 \cdot B1) \cdot Cout0
\end{align*}
\]

Where \(Cout0\) is an intermediate output carry of the first full adder (taken from LSB side). To implement a full adder using static CMOS logic required 28 transistors, thus to implement 2-bit adder cell using same logic required 56 transistors. Due to the requirement of a large number of transistors, dynamic power consumption and node capacitances are increased, which leads to a cumulative increase in net energy consumption [10]. The output generated carry using the concatenation of two full adder cells requires 6 stages for output generation, which increases the propagation delay of the overall circuit.

II.2 2-Bit Inexact Adder

The proposed design (InEMA-1) has been shown in Fig. 1, which has been formulated through the 2-bit exact adder equations (1-3). The approximation has been made through the following steps:

**Step 1:** In this Fig. 1 lower significant adder has been replaced by an OR gate. In this architecture (Fig. 1), \(Cin\) and \(Cout\) have been ignored from the design, which offers the reduction of the carry chain from input to output. Therefore, due to the absence of \(Cin\), the first full adder (observation from the LSB side) can be replaced by a half adder with only two inputs \((A0, B0)\). However, to implement the half adder XOR and AND gates are required to produce the Sum and Carry, respectively.

**Step 2:** Basically, in adders, XOR gates tend to contribute to high delay, area, and power [17, 18]. Therefore, for approximating the half-adder, XOR gate of half adder is replaced with OR gate (both are having the equivalent outputs except for logic high inputs) as given by equation (4). Further, the requirement of the number of transistors for the circuit implementation is reduced (through the replacement of XOR gate with OR gate), which leads in the reduction of the dynamic power consumption and node capacitances of the overall circuit.

\[
Sum0 = A0 + B0
\]

**Step 3:** Through the assumption, neither \(Cin\) nor \(Cout\) is considered; thus, the truth table of a 2-bit adder with 5 inputs and 32 possible combinations can be reduced to 4 inputs and 16 possible combinations, which are shown in Fig. 2. In this Fig. (Fig. 2) the exact truth table for \(Sum1\) has been shown, which could be approximated through the interchanging of the bits marked with the help of arrow (Fig. 2). Through the approximation, the equation can be reduced to only an XOR gate, which has been given in eqn. 5.
Step 4: The carry out (Cout) of the half adder is generated by AND gate. Using this concept, as shown in Fig. 1 only AND gate is used in the last stage (observation from LSB side) to generate Cout1, which will be used as a carry-in for the higher bit position of exact adder stages. Moreover, an AND gate for Cout1 generation gives us only 2 errors in the Cout1 of the proposed 2-bit inexact adder design, which has been shown in Table I. In this table (Table I) erroneous output has been noted (✗), and correct output has been indicated by (√) for the 2-bit inexact adder. To implement the same using static CMOS technology requires only 24 transistors instead of 56.

Step 5: It is observed from the Fig.1 that XOR gate and AND gate have been generating Sum1 and Cout1, respectively. By using Boolean algebra, the expression for Cout1 and Sum1 can be altered, which is shown in eqn. 6 and 7. Now, Sum1 requires two NOR gates and one AND gate for the implementation of this expression and the output of the AND gate can be directly taken as Cout1. Therefore, Fig. 3(a) shows the reduced gate level diagram and Fig. 3(b) shows the transistor level diagram of the modified proposed 2-bit inexact adder (InEMA-1), where it requires only 20 transistors for the implementation rather than 24 transistors (Fig. 1).

\[
\text{Sum1} = (A1 + B1) + (A1 \cdot B1) \quad (6)
\]

\[
\text{Cout1} = A1 \cdot B1 \quad (7)
\]

II.3 4-Bit Inexact Adder

4-bit inexact adder (InEMA-2) design has been given in Fig. 4. In this figure (Fig. 4) A [3:0] and B [3:0] are the input functions, whereas Cout3 and Sum [3:0] are the required outputs. The design has been implemented through the cascade combination of two InEMA-1s. One extra AND gate is used in the MSB position to generate the carry out (as a carry-in) for the next stage (if required). In this design, the intermediate carry generation and propagation are also avoided. Therefore, the carry chain is avoided to reduce the carry propagation delay of the circuit. Furthermore, the number of transistors requirement for the implementation of the proposed InEMA-2 can be reduced significantly.
Similar Boolean expressions (eqn. 6, 7) are also applicable for the proposed InEMA-2. As seen from the above diagram (Fig. 4), Cout3 and Sum3 are generated by AND gate and XOR gate, respectively. Furthermore, the transistor count can be reduced by a similar fashion like Fig. 3. Fig. 5(a) and Fig. 5(b) depict the resulting transistor level diagram and the block diagram of the modified InEMA-2, respectively.

II.3 8-Bit Inexact Adder

Fast adders like Carry Look Ahead Adders (CLAs) are mostly used in digital systems. However, CLAs require larger circuitry and dissipate more power as compared to ripple carry adders (RCA). Therefore, 8-bit inexact RCA (InEMA-3) has been proposed and shown in Fig. 6. In this Fig. 6, InEMA-2 adder cell has been put in the LSB side of the 8-bit inexact adder.

Inexact computing gives immense opportunity to introduce error for an error-resilient application. In binary representation, the weight of each bit position is increasing from right to left. So as to reduce the error distance (explained in section 3.1) InEMA-2 replaces four single bit precise full adders in the LSBs of the exact 8-bit RCA instead of MSBs. Due to the use of proposed inexact adders in 8-bit RCA, it is convincingly reducing transistor count and power dissipation. Precise 8-bit RCA requires 224 transistors (using static CMOS logic implementation), whereas the proposed InEMA-3 requires only 150 transistors using the same logic.
III. RESULT AND DISCUSSION

III.1 Error Analysis

Gate level design of the proposed adder as well as the reported adder so far [13]–[17], [19] has been coded with Matlab for the calculation of the error matrices. All the possible input combinations between 0 to $2^N - 1$, where $N$ stands for the number of inputs have been considered for the variations, and the output has been observed. The output of the inexact adder has been compared with the output of the exact adder, and the error metrics have been examined. These error matrices can give us an idea about the accuracy of the proposed circuit. The following performance metrics for error analysis (as defined in [16]) have been evaluated for the comparison:

1. Error Distance (ED):
   For these inexact designs, a metric has been used to evaluate the inexactness with respect to the exact result; the so-called error distance has been proposed in [23] as a figure of merit for inexact computing. For a given input, the error distance $(ED)$ is defined as the arithmetic difference between the exact result $(E)$ and the inexact result $(I)$.
   \[
   ED(E, I) = |E - I| = |\sum_i E[i] \times 2^i - \sum_j I[j] \times 2^j| \quad (8)
   \]
   Where $i$ and $j$ are the indices for the bits in $E$ and $I$, respectively.

2. Error Rate (ER):
   ER is characterized as the level of incorrect yields among all yields.
   \[
   ER = \frac{\text{Total Number of Erroneous Output}}{\text{Total Number of Outputs}} \times 100 \quad (9)
   \]

3. Total Error Distance (TED):
   It is the absolute sum of error distance.
   \[
   TED = |\sum ED| \quad (10)
   \]

4. Mean Error Distance (MED):
   MED is the average for a set of outputs.
   \[
   MED = \frac{\text{Total Error Distance}}{\text{Total Number of Outputs}} \quad (11)
   \]

5. Normalized Mean Error Distance (NMED):
   NMED is the normalized value of MED.

Fig. 7. (a) Error Rate, (b) MED, (c) NMED of proposed & existing inexact adders
In this context, the existing single bit inexact adders [13]–[17] are concatenated to make 2-bit inexact adder for the error analysis with the InEMA-1 adder cell. The error analysis results of the InEMA-1 adder cell with other inexact designs are provided in Fig. 7.

The error rate of adder cells has been clustered into several groups shown in Fig. 7(a). From the Fig. 7(a) it has been observed that error rate of the [14], [15], [16], [17] are falls in one category (marked in yellow in Fig. 7(a)) which has been considered for the comparison. The error rate which is shown in blue cluster (Fig. 7(a)) is not considered for comparison due to the higher error rate. The error rate of the proposed InEMA-1 (marked in red in Fig. 7(a)) has been reduced by ~12.5% compared with the best reported architectures [14], [15], [16], [17].

MED and NMED have been calculated for the existing and proposed InEMA-1 adder. As shown in Fig. 7 (b) MED of the proposed design (marked in red in Fig. 7(b)) have been reduced by 50% from the best-reported architectures [14], [17] (marked in yellow in Fig. 7(b)) and NMED of the proposed InEMA-1 (marked in red in Fig. 7(c)) have been reduced by 12.5% from the existing best design [13] (marked in yellow in Fig. 7(c)). The blue clusters [Fig. 7(b) and 7 (c)] have not considered for comparison due to the higher MED and NMED respectively.

### III.1.2 Error Analysis of 4-Bit Inexact Adder (InEMA-2)

In 4-bit configuration, the error analysis has been performed between the proposed InEMA-2 adder and the existing inexact adders. Existing 1-bit inexact adders are concatenated to form a 4-bit inexact adder, and then error analysis is performed in MATLAB. Fig. 8 shows the analysis of error metrics between the proposed InEMA-2 adders and the cascaded 4-bit inexact designs.

From Fig. 8(a), it has been observed that error rate of [15], [16], [17] are falls in one category (marked in yellow in Fig. 8(a)), which has been considered for the comparison. The error rate of the proposed InEMA-2 (marked in red in Fig. 8(a)) has been shown ~6.5% improvement as compared with the best reported architectures [15], [16], [17].

Similarly, Fig 8 (b, c) has been shown that the proposed InEMA-2 (marked in red in Fig. 8(b, c)) have an appreciable reduction of MED and NMED over [17] (marked in yellow in Fig. 8(b, c)) respectively. Like as Fig 7 the blue clusters cells have not considered for comparison due to the higher error matrices.

### III.1.3 Error Analysis of 8-Bit Inexact Adder (InEMA-3)

Four number of exact single bit Full Adders in the LSBs of the precise 8-bit RCA is being replaced by the proposed InEMA-2 adder cell to form InEMA-3 as discussed in section 2.4. Existing single bit inexact adders are being concatenated to form 4-bit inexact adder. Similarly, four precise single-bit full adders in the LSBs of the 8-bit exact RCA are replaced by the concatenated 4-bit inexact adders to form 8-bit inexact RCA. The simulation for the proposed 8-bit inexact RCA and existing 8-bit inexact RCAs are presented in this context. The results are shown in Fig. 9.

The error rate of adder cells has been grouped into three groups as shown in Fig. 9(a). From the Fig. 9(a) it has been observed that error rate of the [15], [16], [17] are falls in one category (marked in yellow in Fig. 9(a)) which has been considered for the comparison. The error rate of the proposed InEMA-3 (marked in red in Fig. 9(a)) has been improved by ~7.5% as compared with the best reported architectures [15], [16], [17].
The circuit parameters like power and delay are extracted in Cadence Spectre using gpdk45 nm technology. For the analysis of multi-bit approximation, we have to create exact models of the existing adder circuits. As discussed in section 3.1.3, the proposed InEMA-3 adder cell is utilized for the analysis of performance parameters and compared with its counterpart.

Similarly, from Fig. 9 (b,c) it has been observed that [13], [19] (marked in yellow in Fig. 9(b, c)) has least MED and NMED. Thereby, the comparison has been done with the proposed InEMA-3 (marked in red in Fig. 9 (b, c)) and observed the betterment from its counterpart.

### III.2 Power, Delay and Power Delay Product (PDP) Calculation

Moreover, as a circuit design prospect, the proposed design has been implemented in Cadence. The designs have been taken from different references and the same has been implemented in the same environment for the fair comparison.
the proposed architecture has least propagation delay followed by [19] (marked in yellow in Fig. 10 (b)). Furthermore, Fig. 10(c) depicted that the proposed InEMA-3 RCA (marked in red in Fig. 10 (c)) has been shown the lowest PDP (0.279fJ) followed by [13] (0.339fJ) (marked in yellow in Fig. 10 (c)).

IV. APPLICATION OF INEXACT ADDERS IN IMAGE DE-NOISING AND IMAGE ADDITION

The performance of the inexact adder cells can be evaluated through image processing applications like image de-noising, image compression, image addition, image sharpening, etc. Therefore, to check the improvement of the results, the researcher have reported in their research [14], [16]–[18]. Researchers in [16] has added Lena and Tulip images to generate a new image, where, the inexact adder [14] has been applied. Moreover, image sharpening application has been performed in [17], where, multiplication operation has been carried out by carry save adder followed by RCA. The researchers of [14], [18] performed image compression and decompression with the help discrete cosine transformation (DCT) and inverse DCT respectively. The hardware realization of such (DCT and IDCT) requires adder. The researchers [12], [16] have replaced the reported inexact adder [12], [16] to check the effectiveness of the same. In this article, the analysis of image de-noising has been carried out in depth and image addition figure has been provided for comparison. To check the improvement of the proposed inexact multi-bit adder, the existing multi-bit inexact [19] has been considered for the comparison.

Fig. 11. 3x3 Kernel for weighted averaging filter

To verify the improvement of the proposed inexact adder cell, the image de-noising application has been used. In this task, a noisy image has been generated by adding salt and pepper noise with a density function of 0.03 to the original 128 ×128 size Lena image. To remove the noise, a weighted averaging filter of 3×3 kernel (shown in Fig. 11) has been utilized, which has been convolved with the noisy image.

![Original Image](image1.jpg) ![Noisy Image](image2.jpg) ![Proposed](image3.jpg) ![Apx-1 [19]](image4.jpg) ![Apx-2 [19]](image5.jpg) ![Apx-3 [19]](image6.jpg)

Fig. 12. Image De-noising Results

To perform the convolution operation adders and multipliers are required, which can be obtained through the proposed inexact adder and exact multiplier [24] respectively. The 16-bit precise adder in the convolution operation has been replaced by the proposed InEMA-3 adder. InEMA-3 is an 8-bit adder, which has been extended to 16-bit; one 8-bit exact RCA has been added to the MSB side of the InEMA-3. Finally, the following parameters like peak signal to noise ratio (PSNR), normalized correlation coefficient (NCC), structural similarity index metric (SSIM): have been evaluated and compared with best-reported results reported so far [19].

IV.1 Peak Signal to Noise Ratio (PSNR)

The peak signal to noise ratio (PSNR), is used as a quality metric between two images and mathematical formula of the same is shown in eqn. 13, where R is the maximum fluctuation in the input image data type and mean square error (MSE) represents the cumulative squared error between the original image and the reconstructed image. This ratio is used as a quality measurement between the original and a reconstructed image. The image quality is directly proportional to the value of PSNR. Fig. 12 shows the results of the image de-noising application which have been carried out through existing [19] and proposed adders. The result offers a significant amount of improvement in PSNR, which has shown in Fig. 13.

\[ PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right) \]  
(13)

Fig. 13. PSNR values for the proposed and existing multi-bit adders

IV.2 Normalized Correlation Coefficient (NCC)

Normalized Correlation Coefficient (NCC) is also a quality metric, used to measure matching of images, where area-based spatial filtering technique for correlation has been utilized [25]. The mathematical formulation for the same is given in eqn. 14.
where 'A' is the exact image, whereas 'B' is the distorted image, complement of A and B is the mean value of the exact and distorted images respectively. The NCC values for the proposed and existing multi-bit adders are shown in Fig. 14.

\[
NCC = \frac{\sum_m \sum_n (A_{mn} - \mu_A) (B_{mn} - \mu_B)}{\sqrt{\sum_m \sum_n (A_{mn} - \mu_A)^2} \sqrt{\sum_m \sum_n (B_{mn} - \mu_B)^2}}
\]  
(14)

**IV.4 Image Addition**

In this task, Apple and Lena images are considered for the image addition. Both the images are of 128x128 sizes. The addition operation performed here is pixel by pixel, where each of the pixels contains a decimal value in between 0 to 255. This decimal value is converted to 8-bit binary and given to the exact and inexact 8-bit RCA for addition. The summation result is again converted back to decimal. The exact addition and inexact addition result has been provided in Fig. 16.

**V. Conclusions**

In this manuscript, multi-bit adders viz. 2-bit and 4-bit approximation technique has been demonstrated. Moreover, the 4-bit inexact adder has been utilized in higher dimension adder (8-bit inexact) to express the advantages of such reported approach. Error metrics and circuit performance parameters have been calculated for further applications. Furthermore, in application prospect, the reported adder has been utilized in image de-noising application and examines the improvement from its counterpart. It would be a welcome approach for the researcher, to check betterment of the proposed adder cells in discrete signal processing applications.

**References**

[1] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in 2013 1ST IEEE EUROPEAN TEST SYMPOSIUM (ETS), Avignon, France, 2013, pp. 1–6, doi: 10.1109/ETS.2013.6569370.

[2] K. Roy and A. Raghunathan, “Approximate Computing: An Energy-Efficient Computing Technique for Error Resilient Applications,” in 2015 IEEE Computer Society Annual Symposium on VLSI, 2015, pp. 473–475, doi: 10.1109/ISVLSI.2015.130.

[3] S. Mittal, “A Survey of Techniques for Approximate Computing,” ACM Comput Surv, vol. 48, no. 4, pp. 62:1–62:33, Mar. 2016, doi: 10.1145/2893356.

[4] V. K. Chippa, S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Approximate computing: An integrated hardware approach,” in 2013 Asilomar Conference on Signals, Systems and Computers, 2013, pp. 111–117, doi: 10.1109/ACSSC.2013.6810241.

[5] N. Kouvavas, “Operations on delta-modulated signals and their application in the realization of digital filters,” Radio Electron. Eng., vol. 48, no. 9, pp. 431–438, Sep. 1978, doi: 10.1049/ree.1978.0062.

[6] S. H. Nawab, A. V. Oppenheim, A. P. Chandrakasan, J. M. Winograd, and J. T. Ludwig, “Approximate Signal Processing,” p. 24, 1997.

[7] D. Zrlic and N. Pjevalica, “Stochastic signal processing using delta-sigma modulation,” in Proceedings of the 5th Biannual World Automation Congress, 2002, vol. 14, pp. 653–658, doi: 10.1109/WAC.2002.1049509.
[8] W. J. Poppelbaum, C. Afuso, and J. W. Esch, “Stochastic computing elements and systems,” in Proceedings of the November 14-16, 1967, fall joint computer conference on - AFIPS ‘67 (Fall), Anaheim, California, 1967, p. 635, doi: 10.1145/1465611.1465696.
[9] B. R. Gaines, “Stochastic Computing Systems,” in Advances in Information Systems Science: Volume 2, J. T. Tou, Ed. Boston, MA: Springer US, 1969, pp. 37–172.
[10] V. Gupta, D. Mohapatra, S. P. Park, A. Raghu Nathan, and K. Roy, “IMPACT: Imprecise adders for low-power approximate computing,” in IEEE/ACM International Symposium on Low Power Electronics and Design, Fukuoka, Japan, 2011, pp. 409–414, doi: 10.1109/ISLPED.2011.5993675.
[11] Y. Kim, Y. Zhang, and P. Li, “Energy Efficient Approximate Arithmetic for Error Resilient Neuromorphic Computing,” IEEE Trans. Very Large Scale Integr. VLSI Syst., vol. 23, no. 11, pp. 2733–2737, Nov. 2015, doi: 10.1109/TVLSI.2014.2365458.
[12] Ning Zhu, Wang Ling Goh, Weija Zhang, Kiat Seng Yeo, and Zhi Hui Kong, “Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing,” IEEE Trans. Very Large Scale Integr. VLSI Syst., vol. 18, no. 8, pp. 1225–1229, Aug. 2010, doi: 10.1109/TVLSI.2009.2020591.
[13] H. R. Mahdian, 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. Regul. Pap., vol. 57, no. 4, pp. 850–862, Apr. 2010, doi: 10.1109/TCSII.2009.2027626.
[14] V. Gupta, D. Mohapatra, A. Raghu Nathan, and K. Roy, “Low-Power Digital Signal Processing Using Approximate Adders,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013, doi: 10.1109/TCAD.2012.2217962.
[15] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, “Approximate XOR/XNOR-based adders for inexact computing,” in 2013 13th IEEE International Conference on Nanootechnology (IEEE-NANO 2013), Beijing, China, 2013, pp. 690–693, doi: 10.1109/NANO.2013.6720793
[16] H. A. F. Almurith, T. N. Kumar, and F. Lombardi, “Inexact Designs for Approximate Low Power Addition by Cell Replacement,” in Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016, pp. 660–665, doi: 10.3850/9783981537079_0042.
[17] Z. Yang, J. Han, and F. Lombardi, “Transmission gate-based approximate adders for inexact computing,” in Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH’15), Boston, MA, USA, 2015, pp. 145–150, doi: 10.1109/NANOARCH.2015.7180603.
[18] S. Dutt, S. Nandi, and G. Trivedi, “Analysis and Design of Adders for Approximate Computing,” ACM Trans Embed Comput Syst, vol. 17, no. 2, pp. 40:1–40:28, Dec. 2017, doi: 10.1145/3131274.
[19] S. Tajasob, M. Rezaalipour, and M. Dehyadegari, “Designing energy-efficient imprecise adders with multi-bit approximation,” Microelectron. J., vol. 89, pp. 41–55, Jul. 2019, doi: 10.1016/j.mejo.2019.04.002.
[20] J.-F. Lin, Y.-T. Hwang, M.-H. Sheu, and C.-C. Ho, “A Novel High-Speed and Energy Efficient 10-Transistor Full Adder Design,” IEEE Trans. Circuits Syst. Regul. Pap., vol. 54, no. 5, pp. 1050–1059, May 2007, doi: 10.1109/TCSII.2007.895509.
[21] Y. Wu, Y. Li, X. Ge, Y. Gao, and W. Qian, “An Efficient Method for Calculating the Error Statistics of Block-Based Approximate Adders,” IEEE Trans. Comput., vol. 68, no. 1, pp. 21–38, Jan. 2019, doi: 10.1109/TC.2018.2859960.
[22] S. Venkatachalame and S. Ko, “Design of Power and Area Efficient Approximate Multipliers,” IEEE Trans. Very Large Scale Integr. VLSI Syst., vol. 25, no. 5, pp. 1782–1786, May 2017, doi: 10.1109/TVLSI.2016.2643639.
[23] 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, doi: 10.1109/TC.2012.146.
[24] A. Habibi and P. A. Wintz, “Fast Multipliers,” IEEE Trans. Comput., vol. C–19, no. 2, pp. 153–157, Feb. 1970, doi: 10.1109/T-C.1970.222881.
[25] B. Zitová and J. Flusser, “Image registration methods: a survey,” Image Vis. Comput., vol. 21, no. 11, pp. 977–1000, Oct. 2003, doi: 10.1016/S0262-8856(03)00137-9.