ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

A Three Level Inexact Computing Framework for Efficient and Accurate Digital Image Compression Using Approximate DCT

Dr. Anupama Jain
Associate Professor
Sagar Institute of Research & Technology, M.P., India
jain.anupama.vds@gmail.com

Dr. Tasneem Jahan
Assistant Professor
Sagar Institute of Research & Technology, M.P., India
tasneemjahanjahan@gmail.com

[1]***Abstract*— This paper proposes a new framework for digital image processing; it relies on inexact computing to address some of the challenges associated with the discrete cosine transform (DCT) compression. The proposed framework has three levels of processing; the first level uses approximate DCT for image compressing to eliminate all computational intensive floating-point multiplications and executing the DCT processing by integer additions and in some cases logical right/left shifts. The second level further reduces the amount of data (from the first level) that need to be processed by filtering those frequencies that cannot be detected by human senses. Finally, to reduce power consumption and delay, the third level introduces circuit level inexact adders to compute the DCT. For assessment, a set of standardized images are compressed using the proposed three-level framework. Different figures of merits (such as energy consumption, delay, power-signal-to-noise-ratio, average-difference, and absolute-maximum-difference) are compared to existing compression methods; an error analysis is also pursued confirming the simulation results. Results show very good improvements in reduction for energy and delay, while maintaining acceptable accuracy levels for image processing applications.**

Index Terms—Approximate computing, DCT, inexact computing, image compression
inexact computing, image compression.

I.INTRODUCTION

Today's amount of information that is computational and power computing systems usually process a significant intensive. Digital Signal Processing (DSP) systems are widely used to process image and video information, often under mobile/wireless environments. These DSP systems use image/video compression methods

and algorithms. However, the demands of power and performance remain very stringent. Compression methods are often utilized to alleviate such requirements. Image/video compression methods fall into two general categories: lossless and lossy.

The latter category is more hardware efficient but at the expense of quality of the final decompressed images/videos. For image processing, the Joint Photographic Experts Group (JPEG) method is the widely used lossy method while the Moving Picture Experts Group (MPEG) method is the widely used lossy method for video processing. Both standards use the Discrete Cosine Transform (DCT) algorithm as a basic processing step. Many different fast algorithms for DCT [1], [2] computation have been developed for image and video applications; however, as all these algorithms still need floating point multiplications; they are computationally intensive, requiring extensive hardware resources. To address these concerns, coefficients in many algorithms such as [3] can be scaled and approximated by integers such that floating-point multiplications can be replaced by integer multiplications [4],[5].

The resulting algorithms are significantly faster than the original versions and, therefore, they are extensively used in practical applications. Consequently, the design of good approximations of the DCT for implementation by narrower bus width and simpler arithmetic operations (such as shift and addition) has received considerable attention over the last few years [6]. An advantageous feature of image/video processing is its highly error-tolerant nature; human senses cannot often perceive degradation in performance, such as quality of visual and audio information. Therefore, imprecise computation can be used in many applications that tolerate some loss of precision and some degree of uncertainty, [7], [8], such as for example image/video processing. The introduction of inaccuracy at circuit level in the DCT computation targets specific figures of merit (such as power dissipation, delay and circuit complexity [9], [10], [11], [12], [13], [14]) and it is very challenging. This scheme targets low power and process tolerance is based on a logic/gate/transistor level redesign of an exact circuit. A logic synthesis approach [9] has been proposed to design circuits for implementing an inexact version of a given function by considering the so-called error rate (ER) as metric for error tolerance. Reduction in circuit complexity at transistor level of an adder circuit (such as by truncating the circuits at the lowest bit positions) provides a reduction in power dissipation higher than conventional low power design techniques [10]; in addition to the ER, new figures of merit for estimating the error in an inexact adder have been

presented in [11]. This paper presents a new framework for approximate DCT image compression; this framework is based on inexact computing and consists of three levels. Level 1 consists of a multiplier-less DCT transformation, so involving only additions; Level 2 consists of high frequency component (coefficient) filtering; Level 3 consists of computation using inexact adders. Level 1 has been widely studied in the technical literature [16], [17], [18]; Level 2 is an intuitive technique to reduce the complexity of computation while attaining only a marginal degradation in image compression. Level 3 follows a circuit level technique by which inexact computation is pursued (albeit new and efficient inexact adder cells are utilized in this manuscript). Therefore, the contribution of this manuscript is found in the combined effects of these three levels. The proposed framework has been extensively analyzed and evaluated. Simulation and error analysis show a remarkable agreement in results for image compression as an application of inexact computing. Hereafter to avoid confusion the word "approximate" is used only for the DCT algorithms while the word "inexact" is used for circuits and design involving non-exact hardware for computing the DCT.

II. Literature review

For manuscript completeness, preliminaries to approximate DCT and a review of relevant topics are presented next.

A. Discrete Cosine Transform (DCT)

To obtain the ith and jth DCT transformed elements of an image block (represented by a matrix p of size N), the following equation is used:

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(x,y) cos\left[\frac{\pi(2x+1)i}{2N}\right] cos\left[\frac{\pi(2y+1)j}{2N}\right] (1)$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, \wedge u = 0 \\ 1, \wedge u > 0 \end{cases}$$

Where $p(x,y)$ $p(x,y)$ is the x, yth x, yth element of the image. This equation calculates one entry (i, jth) (i, jth) of the
Transformed image from the pixel values of the original image matrix. For the commonly used 8×8 8×8
Block for JPEG compression, *N* is equal to 8 and $x \wedge y$ $x \wedge y$ from 07 07. Therefore $D(I,j)$ $D(I,j)$ is also given
By the following equation:

$$D(i,j) = \frac{1}{4} C(i)C(j) \sum_{i=0}^{7}\sum_{j=0}^{7} p(x,y) cos\left[\frac{\pi(2x+1)i}{16}\right] cos\left[\frac{\pi(2y+1)j}{16}\right] (2)$$

For matrix calculations, the SCT matrix is obtained from the following:

$$T_{DCT}(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, \wedge i = 0 \\ \sqrt{\frac{2}{N}} cos\left[\frac{\pi(2j+1)i}{2N}\right], \wedge i > 0 \end{cases} (3)$$

$$T_{DCT}(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, \wedge i = 0 \\ \sqrt{\frac{2}{N}} cos\left[\frac{\pi(2j+1)i}{2N}\right], \wedge i > 0 \end{cases} (3)$$

So, DCT is computation intensive and may require floating-point operations for processing, unless an approximate algorithm is utilized.

B. Joint Photographic Experts Group (JPEG)

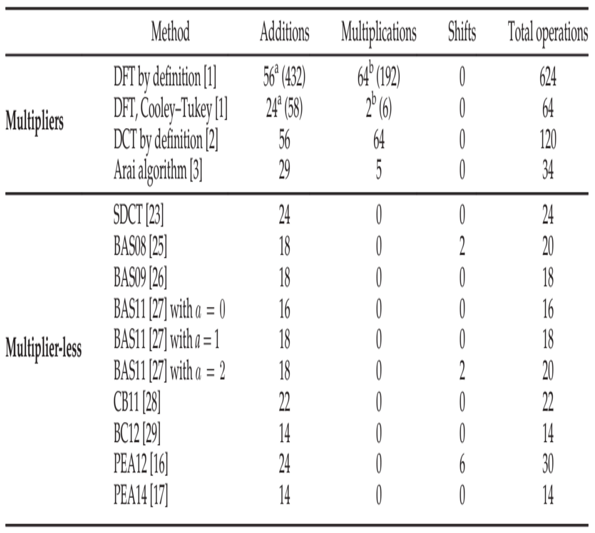
The JPEG processing is first initiated by transforming an image to the frequency domain using the DCT; this separates images into parts of differing frequencies. Then, the quantization is performed such that frequencies of lesser importance are discarded. This reflects the capability of humans to be reasonably good at seeing small differences in brightness over a relatively large area, but they usually cannot distinguish the exact strength of a rapidly varying brightness variation. The compression takes place during this quantization step in which each component in the frequency domain is divided by a constant, and then rounded to the nearest integer. This results in many high frequency components having very small or likely zero values, small values at best. The image is then retrieved during the decompression process that is performed using only the important frequencies that have been retained. For JPEG processing, the following steps must be executed:
1. An image (in color or grey scales) is first subdivided into blocks of kxk pixels (usually k = 8).

2. Then from left to right and top to bottom, the DCT is applied to each block.
3. This generates kxk coefficients (so 64 for k = 8) that are then quantized to reduce the magnitudes.
4. The resulting array of compressed blocks represents the compressed image, i.e., the stored or transmitted image. 5. To retrieve the image, the compressed image (array of blocks) is decompressed using Inverse DCT (IDCT).

C INEXACT ADDITION AND APPROXIMATE DCT

Arithmetic circuits are well suited to inexact computing; addition has been extensively analyzed in the technical literature and is one of the fundamental arithmetic operations in many applications of inexact computing [31]. A reduction in circuit complexity at transistor level of an adder circuit usually provides a good reduction in power dissipation, often higher than conventional low power design techniques [10]. Inexact adder designs have been evaluated in [12]: inexact operation has been introduced by either replacing the accurate cell of a modular adder design with an approximate cell of lower circuit complexity, or by modifying the generation and propagation of the carry in the addition process. In [14], three new inexact adder cell designs have been presented (denoted as InXA1, InXA2 and InXA3); these cells have both electrical and error features that are very favorable for approximate computing. These adder cells as shown in Table1 have the following advantageous features over previous designs [10], [13]: (i) a very small number of transistors; (ii) a very small number of erroneous outputs at the two outputs (i.e., Sum and Carry); (iii) smaller switching capacitances (expressed in Cgn gate capacitance of minimum size NMOS), thus incurring in a substantial reduction in both delay and energy dissipation (Table 3) (and their product as combined metric). Metrics such as delay, energy dissipated and EDP (energy delay product) of the inexact cells for both average and worst

TABLE 1
Approximate DCT Methods Applied to Image Compression; Number of Operations Required to Calculate the DCT for an 8x8 Block Size



	Method	Additions	Multiplications	Shifts	Total operations
Multipliers	DFT by definition [1]	56^a (432)	64^b (192)	0	624
	DFT, Cooley–Tukey [1]	24^a (58)	2^b (6)	0	64
	DCT by definition [2]	56	64	0	120
	Arai algorithm [3]	29	5	0	34
Multiplier-less	SDCT [23]	24	0	0	24
	BAS08 [25]	18	0	2	20
	BAS09 [26]	18	0	0	18
	BAS11 [27] with $a = 0$	16	0	0	16
	BAS11 [27] with $a = 1$	18	0	0	18
	BAS11 [27] with $a = 2$	18	0	2	20
	CB11 [28]	22	0	0	22
	BC12 [29]	14	0	0	14
	PEA12 [16]	24	0	6	30
	PEA14 [17]	14	0	0	14

cases are presented in Table 1. Among the inexact cells, InXA1 has the least average and worst case delays while InXA2 incurs in the least average and worst case power dissipations and least average EDP. The average and worst case delays and energy dissipation of the adder cells have been determined by exhaustive simulation. For each input signal, the delay is measured when the output reaches 90 percent of the maximum value while the energy dissipated is measured in all transistors during the time when the output reaches 90 percent. As per these advantages, InXA1 and InXA2 based adders are considered for the DCT application as treated next.

III. PROPOSED APPROXIMATE FRAMEWORK

This paper presents a new image compression framework that consists of three levels of approximation as follows.
Level 1 is the multiplier-less DCT transformation, Level 2 is the high frequency filtration, Level 3 is the inexact computation. Levels 1 and 3 were explained in previous sections. Although high frequency filtration (Level 2) is not a new concept, it is appropriate to describe it for sake of completeness because it contributes to the proposed framework for reducing its execution time and energy. Therefore, instead of performing the quantization process on all resulting DCT transformation coefficients, the process is only performed on the set of coefficients for the low frequency components of the transformed block.

A. High Frequency Filtration

Filtering the high frequencies generates an image that is hardly distinguished by the human eye (as only sensitive to low frequency contents).
This feature can be used to compress an image. As outlined earlier, a DCT transforms the image in the frequency domain such that it is possible to ignore those coefficients that encode the high frequency components (so not sensitive to the human eye) while retain the other coefficients. Different numbers of retained coefficients are considered when applied to image compression applications; it has been demonstrated that just 0.34-24.26 percent out of 92112 DCT coefficients are sufficient in high speed face recognition applications [34], [35]. Examples for 8 X8 image blocks are as follows:
Image compression using a supporting vector machine in which only the first 8–16 coefficients are considered [36], An image reconstruction method based on three coefficients only as proposed in [37],
Evaluation and assessment of various image compression methods employing only 10 coefficients as in [25], [26].

B. Approximate DCT Implementation

Unlike the implementations of approximate DCT approaches found in Table 1, next all required calculations (addition and subtraction) are implemented at bit level using the corresponding logic functions. The length of all operators is given by 32-bits and implementations are simulated by MATLAB using their Boolean logical functions.

Selected approximate DCT approaches are simulated for the Lena image; the results are plotted in Fig. 1 in which the Power Signal to Noise Ratio (PSNR) of all methods is plotted against the number of Retained Coefficients (RC) used in the quantization stage of the compression. The PSNR is calculated from the Mean Square Error (MSE) as follows:

- Mean Square Error (MSE):

$$MSE = \frac{1}{m \times n}\sum_{x=1}^{m}\sum_{y=1}^{n}$$

Where $p_{j,k}$ $p_{j,k}$ is the accurate pixel value at row x x and column y y of the image, $\acute{p}_{x,y}$ $\acute{p}_{x,y}$ is the approximate value of the same pixel, m m and n n are the size of the image (rows and columns respectively).

- Peak Signal to Noise Ratio (PSNR):

$$PSNR = 10log$$

The results show that, except for the non-orthogonal SDCT method compression using CB11 generally produces the best outcome in terms of PSNR. Three types of behavior are observed. Increasing output quality with an increase of the number of retained coefficients (RC). This occurs for CB11, BAS08, BAS09, BAS11(a = 0 and a = 1), an almost constant PSNR by increasing the RC. This occurs for BC12 and PEA14, degradation in output quality with an increase of RC. This occurs for both BAS11(a = 2) and PEA12.Two additional measures are used for a better insight on the resulting quality, i.e., the Average Difference (AD) and the Maximum Absolute Difference (MD). These metrics are defined as

- Average Difference (AD):

$$AD = \frac{1}{m \times n}\sum_{j=1}^{m}\sum_{k=1}^{n}(p_{x,y} - \acute{p}_{x,y})(7)$$

- Maximum Absolute Difference (MD):

$$MD = \max_{m,n}\{|p_{x,y} - \acute{p}_{x,y}|\}(8)$$

Figs. 2 shows the resulting AD and MD for all methods; the average difference between the uncompressed and inexact-compressed images become smaller as RC increases except for BAS11(a =2) and PEA12 (further confirming the PSNR results

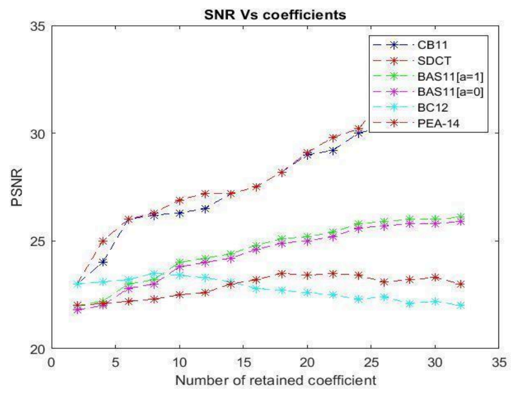


Fig. 1. Compression of an image using approximate DCT and bit-level exact computing.

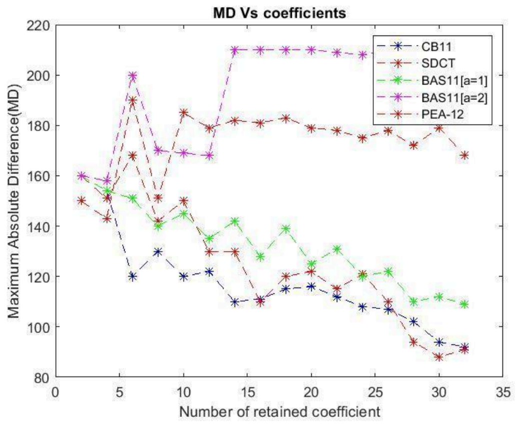


Fig. 2. Maximum absolute difference (MD) for compression of an image using approximate DCT

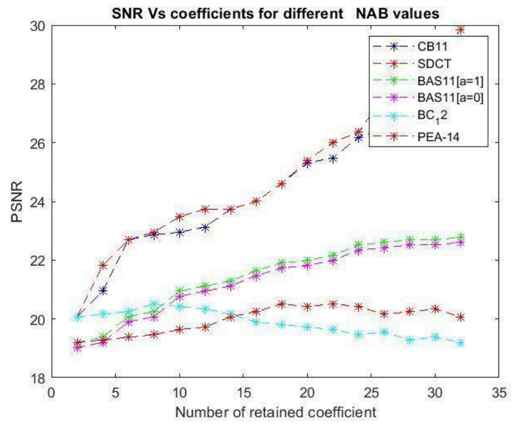


Fig. 3. Approximate DCT compression of an image using inexact adders with different NAB values; (Number of Approximate bits) NAB =3

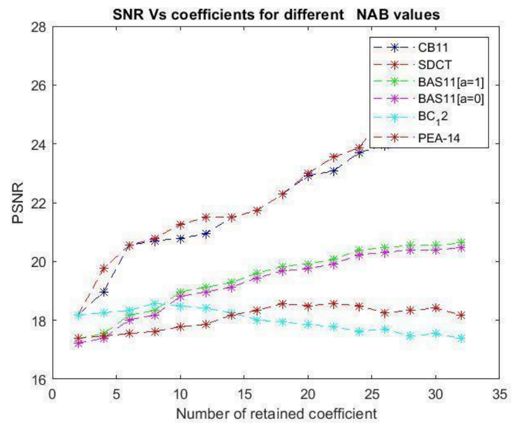


Fig. 4. Approximate DCT compression of an image using inexact adders with different NAB values; NAB =4

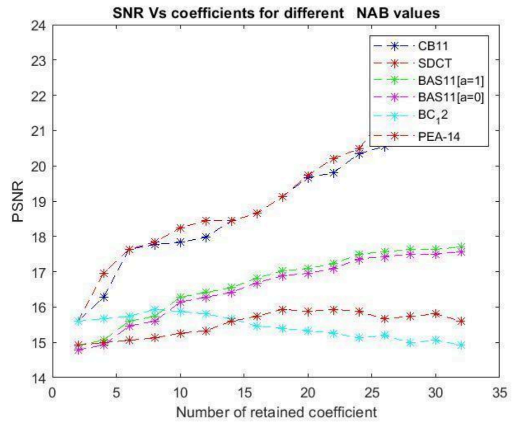


Fig. 5. Approximate DCT compression of an image using inexact adders with different NAB values; NAB =5

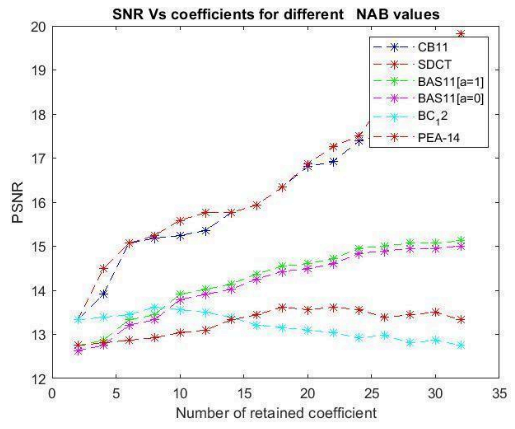


Fig. 6. Approximate DCT compression of an image using inexact adders with different NAB values; NAB =6 in Fig.1).

Fig. 2 shows that the MD between the uncompressed and inexact-compressed image pixels is reduced as more retained coefficients are used, the exceptions are PEA12 and BAS11 (a =2). This further confirms the previous results.
Fig. 4 depicts the compressed Lena image using the most accurate CB11 method for three RC values, i.e., 4, 10 and 16 retained coefficients. This figure also shows for comparison purpose the exact DCT compression results with RC = 16.

C. *Approximate DCT Using Inexact Computing*

Consider next the approximate DCT compression of Lena using inexact adders; as previously, the value of the NAB is increased from 3 to 6. The PSNR results are shown in Fig. 3,4,5,6 versus RC; the PSNR of the compressed images (a measure of quality) is plotted by executing all approximate DCT methods using only one inexact adder (for example the top row uses AMA1 as the inexact adder). Each column plots the quality of the compressed images by executing all approximate DCT methods with only inexact adders (for a NAB value). For example the left most column are for NAB = 3. As expected, the PSNR deteriorates as the NAB increases (an acceptable level of PSNR is reached at a NAB value of 4). 4.4 Truncation Truncation is one of the possible inexact computing techniques that may be utilized; the results of using truncation are also shown. The use of inexact adders results in more accurate results (truncation is performed at values of 3 and 4 bits).

IV. CONCLUSION

This paper has presented a new approach for compressing images by approximate compression using the Discrete Cosine Transform (DCT) algorithm. The proposed approach consists of a 3-level framework by which initially a multiplier-less DCT transformation (so involving only additions and shift operations) is executed; this level is followed by a high frequency component (coefficient) filtering and computation using inexact adders. It has been shown that using 8 x 8 image blocks each level contributes to an approximation in the compression process, while still generating at the end a very high quality image. This manuscript has confirmed that the combined effects of these three levels are well understood; simulation and error analysis have shown a remarkable agreement in results for image compression as an application of inexact computing.
As the proposed framework has been proved to be effective for a DCT method combining approximation at all three proposed levels, the following specific findings have been found and confirmed in this manuscript by simulation and error analysis. Among all approximate DCT methods, CB11 produces the best quality compression (highest PSNR values) when using exact 16 bits adders (Fig. 1). Other image manipulation quality measures (AD and MD) confirmed the PSNR results. (Figs. 1 and 2). Methods BAS08, BAS11 with a = 0 and BAS11 with a = 1 are the next best methods. Among all inexact adders discussed [14], it has been found that InXA2 performs the best. When inexact adders are utilized to implement approximate DCT JPEG compression, non-truncation based methods produces better results than the corresponding truncation schemes, especially when considering higher NABs. (Figs. 3,4,5and 6). The results for the DCT computed by using inexact adders are consistent when different images were used. In

general acceptable compression can be obtained with NAB values up to 4. Then it has been shown that the quality of the results decreases substantially when larger NAB values are used. On average using 4 image benchmarks, the BC12 and PEA14 methods take the least execution time and energy to compress an image compared to compressing an image using an exact adder. As for the best PSNR as a metric of image quality, the approximate DCT method CB11 produces the highest value; however if both reductions in execution time and energy are considered, then BAS09 is the best approximate DCT method.

REFERENCES

[1].R. E. Blahut, Fast Algorithms for Digital Signal Processing. Reading, MA, USA: Addison-Wesley, 2005

[2.]V. Britanak, P. Yip and K. R. Rao, Discrete Cosine and Sine Trans forms. New York, NY, USA: Academic, 2007 [3].Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for im- ages," IEICE Trans., vol. E-71, no. 11, pp. 1095–1097,2008.

[4].D. Tun and S. Lee, "On the fixed-point-error analysis of several fast DCT algorithms," IEEE Trans. Circuits Syst. Vid. Technol., vol. 3, no. 1, pp. 27–41, Feb. 2009.

[5].C. Hsu and J. Yao, "Comparative performance of fast cosine trans form with fixed-point roundoff error analysis," IEEE Trans. Signal Process., vol. 42, no. 5, pp. 1256–1259, May2010.

[6].J. Liang and T. D. Tran, "Fast multiplierless approximations of the fast DCT algorithms," IEEE Trans. Circuits Syst. Vid. Technol., vol. 3, no. 1, pp. 27–41, Feb. 1993.DCT with the lifting scheme," IEEE Trans. Signal Process., vol. 49, no. 12, pp. 3032–3044, Dec. 2011.

[7].Y. Dote and S.J. Ovaska, "Industrial applications of soft comput ing: A review," Proc. IEEE, vol. 89, no. 9, pp. 1243–1265,Sep.2013.

[8].K. V. Palem, "Energy aware computing through probabilistic switching: A study of limits," IEEE Trans. Comput., vol. 54, no. 9, pp. 1123–1137, Sep. 2015.

[9].D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proc. Des. Automat. Test Europe,2020,pp.957–960.

[10] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Des. Integrated CircuitsSyst.,vol.32,no.1,pp.124–137,Jan.2016.

[11] 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. 2018.

[12] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in Proc. ACM/IEEE Great Lakes Symp. VLSI, May 2015, pp. 343–348.

[13] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate xor/xnor-based adders for inexact computing," in Proc. IEEE Int. Conf. Nanotechnology, Aug. 2013,pp.690—693.

[14] H. A. F. Almurib, T. Nandha Kumar and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in Proc. Conf. Des. Autom. Test Europe, Mar. 2016,pp.660–665.

[15] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power DCT architecture," in Proc. Des.Automat.Test Europe, 2017, pp. 1–6.

[16] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, and N. Rajapaksha, "Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing," Meas. Sci. Technol., vol. 23, no. 11, pp. 1–15,Nov. 2022.

[17] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya, "Improved 8-point approximate DCT for image and video compression requiring only 14 additions," IEEE Trans. Circuits Syst. I Regular Papers, vol. 61, no. 6, pp. 1727–1740, Jun. 2014.

[18] N. Merhav and B. Vasudev, "A multiplication-free approximate algorithm for the inverse discrete cosine transform," in Proc. Int. Conf. Image Process., Oct. 24-28, 1999,pp.759–763.

[19] C. Loeffler, A. Lightenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications, "in Proc. IEEE Int. Conf. Acoustics Speech Signal Process., Feb. 2019, pp. 988–991.

[20] P. Duhamel and H. H'Mida, "New S DCT algorithms suitable for VLSI implementation," in Proc. IEEE Int. Conf. Acoustics Speech Signal Process., 2017, pp. 1805–1808.

[21] E. Feig and S. Winograd, "On the multiplicative complexity of discrete cosine transform," IEEE Trans. Inform. Theory, vol. 38, no. 4, pp. 1387–1391, Jul. 2022.

[22] V. Lecuire, L. Makkaoui, and J.-M. Moureaux, "Fast zonal DCT for energy conservation in wireless image sensor networks," Electron. Lett., vol. 48, no. 2, pp. 125–127, Jan. 19,2022.

[23] T.I. Haweel, "A new square wave transform based on the DCT," Signal Process., vol. 82, pp. 2309–2319, 2001.

[24] K. Lengwehasatit and A. Ortega, "Scalable variable complexity approximate forward DCT," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 11, pp. 1236–1248,Nov.2014.

[25] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low complexity 8 x 8 transform for image compression," Electron. Lett., vol. 44, no. 21, pp. 1249–1250, 2018.