International Journal of VLSI design & Communication Systems (VLSICS)
Vol 13, No 1/2/3/4/5/6, December 2022

APPROXIMATE ARITHMETIC CIRCUIT DESIGN FOR ERROR RESILIENT APPLICATIONS

Viraj Joshi and Pravin Mane

Department of EEE, Bits Pilani, K.K. Birla Goa Campus, Goa, India

ABSTRACT

When the application context is ready to accept different levels of exactness in solutions and is supported by human perception quality, then the term 'Approximate Computing' tossed before one decade will become the first priority . Even though computer hardware and software are working to generate exact results, approximate results are preferred whenever an error is in predefined bound and adaptive. It will reduce power demand and critical path delay and improve other circuit metrics. When it comes to traditional arithmetic circuits, those generating correct results with limitations on performance are rapidly getting replaced by approximate arithmetic circuits which are the need of the hour, and so on about their design.

KEYWORDS

Approximate Computing (AC), Error Rate, Acceptance Probability, Mean Error Distance, Error Metrics, Circuit Metrics

1. INTRODUCTION

The need for approximate computing arose due to changing face of computing workloads as well as the requirement for fully efficient resources [1],[2]. Approximate computing is supporting various applications ranging from filtering, and convolutional neural networks to big data applications because of a few of the reasons. Due to psycho-visual limitations of a human being, a small error in the visual output is unnoticed by the user. The second one is the resilience to input noise so that the user can easily accept the noisy output (approximate output). Next is instead of one golden output for a given input, multiple solutions will go correct. One more is that during the approximation process if an error occurs at one stage will be getting nullified with equal and opposite error at other stages. Also, EDU (Error Detection unit) and ECU (Error Correction Unit) are actively working on error reduction. Other than that as silicon device scaling reached saturation level, to improve circuit efficiency approximate computing is the best choice [1].

A growing number of applications are designed to tolerate "noisy" real-world inputs and use statistical or probabilistic types of computations. These computations are trying to generate the results best matching the standard and are not traditional ones where an exact answer is required. These computations provide acceptable quality results and not to be perfect [1], [2]. And, here approximate computing is preferred. Actually, approximate computing is a concept in which inaccurate results are generated that are slightly deviating from exact results but not wrong or random. These results are good enough to produce output not fully correct but partially. This conflict in the case of AC gave birth to a trade-off between its basic parameters like quality and performance as shown in fig. 1[3].

DOI: 10.5121/vlsic.2022.13601

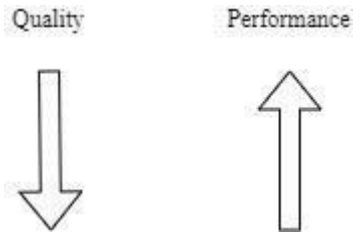

Figure 1. A trade-off in Approximate Computing

2. FROM ACCURATE TO APPROXIMATE

For any arithmetic circuit adder, subtractor and multiplier are the basic building blocks. Usually, an approximation can be added at one of these stages or at all. Fig.2 describes the basic concept of adding approximation in accurate system

An approximation is usually introduced either in construction or at the logic level or at the hardware description of arbitrary circuits. In other words, it is stated that approximation is achieved at four different levels [1]. First is the algorithm level where the actual algorithm is kept intact either by altering the inputs or the hyperparameters preferred in machine learning and called meta-learning. Another approach is the application level where algorithms are not intact and modified to achieve a level of approximation. Loop perforation is the one and in this, the loop iterations are managed by the user. The third approach is working at an architectural level. An approximation is achieved by doing modifications in an instruction set with error resilience bounded for each instruction. The last one works at the circuit level and is related to hardware circuits. Based on this a lot of work is available w.r.t. adders and multipliers categorized into deterministic and non-deterministic types.

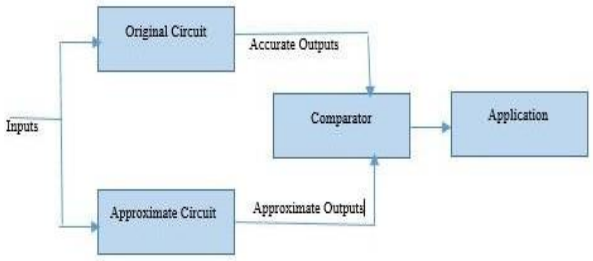

Figure 2. From Accurate to Approximate

3. WHY AC?

Though approximate computing is tossed and practiced for the last decade, it is still in trend for researchers due to two key parameters: power and reliability. Fig. 3 describes the concept of approximate computing. When approximation is introduced at a software level, better energy or power-efficient system is realized.

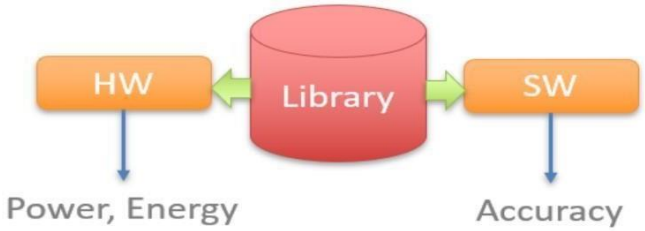

Figure 3. The idea of Approximate Computing

Approximate Computing is used in error-tolerant applications ranging from multimedia processing, machine learning, signal processing, scientific computing, and many more. There are several reasons that AC is preferred in these applications.

1. Need for energy-efficient systems: Calculations needed for approximate computing are less, accordingly less hardware requirement and so it is cost-effective.
2. In the case of big data applications, the database is carrying huge data. Out of that AC is working on the data which is used to generate actual output. This is time-efficient. Also, real-time computations are done so that resultant data values are used to generate approximate statistical values.
3. Need for an hour is a performance as compared to accuracy. In those situations, AC is preferred.
4. To reduce common dependencies like 'carry chain propagation' in the case of arithmetic circuits. It minimizes dependency chains to reduce calculations as well as limit the bit length of operands.
5. Parallelization: When calculations are lengthy, then the serialization of data or calculations is a big problem. This concept manages time constraints in a powerful way.
6. Ease of use: By putting in very small efforts approximate computing is showing maximum adoption for a wide data range.
7. Suitable for inexact model: When models are naturally inexact then approximate computing perfectly suits this application domain.
8. Limitations on supply voltage usage are strict or voltage scaling techniques reach to saturation, AC is having full scope in such situations .
9. Deteriorating Moore's law in practise pushes towards AC.
10. Using Multiple Inexact Program Versions or Inexact or Faulty Hardware will indirectly supports AC.

4. REQUIREMENTS OF AC

It must be a result-oriented application-specific system.

While applying approximations safety and quality properties must be distinguished. For a successful approximate computing system, hardware and software co-design is an important factor.

The measure of the approximate system decides the trade-off between generality and potential efficiency. Every approximate model must have probabilistic and statistical error analysis characteristics.

5. BACKGROUND

5.1. Review of Approximate Adders

Vaibhav Gupta et.al. [4] proposed various inexact full adders with the reduction in a number of transistors in basic full adder cells and utilized them in the construction of multibit approximate adders used in signal processing applications showing up to 69 % power saving without much degradation in the quality of output.

Arnab Raha et.al. [5] designed Reconfigurable Adder/subtractor Block (RAB) for video encoding which controls the degree of approximation (DA) across different videos maintaining PSNR degradation within 1% to 10% while achieving up to 38% energy saving compared to exact encoders.

Bharath Srinivas Prabakaran et.al. [6] proposed different FPGA-based multi-bit architectures for performing approximate addition. The outcome of the paper is in the achievement of gains in the area of 50%, latency of 38%, and power-delay product of 53% as compared to 16-bit exact adders. The Register Transfer Level and behavioral codes of these approximate modules are open-source and can be used for further fueling the research in designing approximate adders. Sunil Dutt et.al. [7] proposed energy-efficient, high performance Approximate Full Adders (AFAs), by using the idea of breaking the carry chain subject to low error rate in adder design and constructed N-bit approximate adder which has shown 46.31% and 28.57% improvement in power and area respectively with respect to ripple carry adder. In order to improve Error Distance (ED) and Error Rate (ER), the concept of carry lifetime, error detection, and correction which provide bit-width aware constant delay is used.

Mahmoud Masadeh et.al. [8] proposed a method to consider the type and position of approximate full adders used in an approximate multiplier as crucial to designing the target design module. The algorithm suggested in the article is Pareto-space exploration technique. The significance of this technique is to mark most optimal approximate designs.

5.2. Review of Approximate Multipliers

Weiqiang Liu et.al. [9] designed two approximate Radix-4 Booth Encoders and two 4:2 compressors which are used to design 4 approximate redundant binary multipliers. The proposed Radix-4 Booth encoders help in generating approximate partial products and the 4:2 compressors replicate a reduction in the number of partial products. The work can be extended by designing new approximate Booth Encoders and 4:2 compressors and applying them to the different multiplier algorithms.

Muhammad Shafique et.al. [10] presented a survey of different techniques used for designing power-/energy efficient approximate arithmetic components like adders and multipliers. The author proposed a method to bridge the gap between the logic layer and architecture layer for enabling cross-layer approximate computing. The design space exploration technique developed by them can be used for building multiple approximate components with improved performance metrics.

K. Bhardwaj et.al. [11] designed a new power-area efficient Approximate Wallace Tree Multiplier (AWTM) with a bit-width aware approximate multiplication algorithm. The carry-in prediction method is employed augmented with hardware efficient carry-in precomputation.

Their result has shown up to 99.85% accuracy, up to 24% reduction in latency, up to 41.96% reduction in power, and up to 34.49% reduction in area. In complex data path designs, errors in an approximation of individual circuits accumulate and are not accepted beyond the error limit. Parag Kulkarni et. al. [12] proposes a design of approximate multiplier showing power savings of 31.78% − 45.4% when compared with an exact multiplier design at the cost of average error of 1.39% -3.32%. Also, it shows improvement in SNR 2X- 8X times than the voltage-over scaling-based power error trade-off method for the multiplier for the same power savings. Chia-Hao Lin et. al. [13] proposes a counter design to effectively reduce partial product stages in Wallace tree multiplier. As compared to the normal Wallace tree multiplier this design is 10.74% power efficient. Also showing 9.8% low latency w.r.t. normal Wallace tree. But error rates for proposed designs are in the range of 0.2% to 13.76% in comparison with Wallace tree multiplier. It is approaching but not exactly 100% with an accuracy of about 99%. With the error detection and correction unit, it is showing more promising results like for 32-bit length, power savings are 10% over Wallace tree multiplier with 6% faster response time. Burdon of adding this error correction and detection unit is less than 6% on the overall system.

Amir Momeni et.al.[14] discusses the design of two proposed 4-2 compressors used in the approximate multiplier and their analysis. Four different schemes have been proposed for these compressors and it has been utilized in Dadda multiplier. These two compressors and four multipliers are simulated using HSPICE for different CMOS feature sizes viz. 32, 22, 16nm. Finally, eight different modes of approximate multipliers are applied to image processing applications and results are compared for error metrics using visual studio.

Sana Mazahir et.al. [15] developed an adaptive error detection and correction scheme in order to restrict the size of errors in data path design. The guidelines were developed for designing flexible approximate adders and multipliers taking into account their integration in data path designs.

6. RESEARCH GAPS AND SCOPE FOR ADDER AND MULTIPLIER DESIGN

After going through the literature survey for approximate adder and multiplier following gaps are noticed where there is a scope for research.

Analysis of error metrics linked with circuit metrics. With an upper bound on error parameters, approximate circuits will be designed with less degradation in area and power consumption and considerable speed improvement.

Operand size is a major issue in arithmetic circuits from hardware and software points of view. Error metric and circuit metric analysis are strongly dependent on operand sizes. Their dependence on operand size can be considerably reduced using approximate circuits. Approximate tree adders and multipliers can be designed for larger operand lengths by introducing approximation techniques.

Signed arithmetic circuits have their own challenges in interpreting results because of their representation in a binary world. After a literature survey, limited work has been reported on signed approximate arithmetic circuits. There seems a scope for approximation in arithmetic circuits with signed operands.

When approximation is introduced, by default error will be present. Error detection (EDU) and correction (ECU) units will be designed for adaptive bound-on on error parameters so that circuits can be used in a multitude of applications. Depending on circuit metric and error metric analysis, the mode of operation is decided for a particular application. Although there will be

degradation in terms of accuracy; as per requirement accurate or approximate mode is selected based on EDU and ECU reports.

Multiplier is the heart of almost all arithmetic and logical circuits and so of many electronic systems. It is the most complex part for design because of the repeated addition of partial products with carry. The scope is to introduce a new technique for partial product generation and addition.

7. PROPOSED DESIGN

This is the proposed idea for adding two numbers without carry propagation. Throughout addition from LSB to MSB, no carry is considered and propagated.

Fig. 4 indicates the schematic of this technique for the addition of two numbers (of any bit length). Let us start with the important features of this technique.

No carry is considered while adding numbers, in short, carry independent technique.

As carry is not waving from LSB to MSB, the carry chain propagation problem does not exist. Non-existence of carry chain propagation problem fastens the arithmetic operations by relaxing the limit on the power requirement.

Which in turn is the best-suited method for battery-operated appliances.

The method itself is self-corrective that suppress the separate need of error detection and correction unit.

Attainable accuracy with the help of circuit parameters makes this method flexible for exact as well as approximate applications.

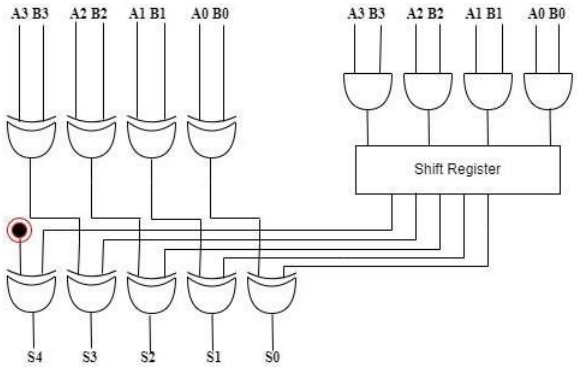

Figure 4. Schematic of the proposed method for addition

The proposed method is slightly different because of the absence of carry propagation, but it also gives accurate results too. Though the number of steps required to generate correct results is more than conventional then also it is energy efficient as well as faster because carry chain propagation (conventional method) consumes more energy and lowers the speed. Evaluation metrics are focusing more on it.

7.1. Quality or Results Metrics

In energy-efficient applications, approximate circuits are playing a vital role. Basically, each approximate circuit is analyzed on the basis of circuit metrics and error metrics. After getting combined results, application level is decided.

The basic circuit metrics include critical path delay (latency), power, and area [LPA]for regular circuits, and one additional constraint called ‘error’ is added for approximate circuits [16], [18], [23].

It is already proved that a large approximation is tolerated in multipliers rather than adders because complex computations are more prone to errors in addition as compared to the multiplication process with the same operand length [23].

7.1.1. Error Metrics

Various approaches are used to perform error analysis of approximate circuits using MATLAB or PYTHON. Few papers suggested data-based error analysis with random inputs to cover maximum inputs for that particular bit length and analysis [19]. This is the most widely used method for error analysis as the number of iterations/ trials is increased then analysis results approach towards accuracy.

Another approach is probabilistic error mode for predicting the possibility of error in the output [15]. This method works on bit position error probability so that the prediction is expected to be closer to reality.

The important error parameters considered for comparative analysis are Error Rate (ER), Mean Error Distance (MED), Average Error, Average Hamming Distance, Mean Relative Error Distance, Normalized Mean Error Distance, Acceptance Probability, etc.

Error rate and Acceptance Probability play a vital role to decide the level of approximation for a particular application.

In this article, the first approach is used for error analysis which considers random inputs.

The number of steps is deciding the mode of operation of a circuit. Circuit switches from approximate to exact mode of operation when a number of steps vary from 5 to 128 progressively.

The number of steps is nothing but the repeated process of logical EX-ORing and ANDing of the operands till getting the correct result.

Table I describes the behavior of a 128-bit length adder for a various number of steps generating different error rates and acceptance probabilities.

Table 1. Error Analysis for 128 Bit Length.

Size	Trial	Number of Steps	Error Rate	Acceptance Probability
128	1000	5	88.4	85
128	1000	9	10.6	98.6
128	1000	13	0.9	100
128	1000	17	0	100
128	1000	21	0	100
128	1000	41	0	100
128	1000	53	0	100
128	1000	65	0	100
128	1000	97	0	100
128	1000	129	0	100

----------Approximate Mode

----------Accurate (Exact) Mode

Fig.5 explains about the acceptance probability variation w.r.t. bit length for fixed number of trials, say 1000. As the number of steps varied from 5,9,13,17 to 21, acceptance probability increased by 15.0%. It is because of an increase in accuracy which leads the results towards exactness.

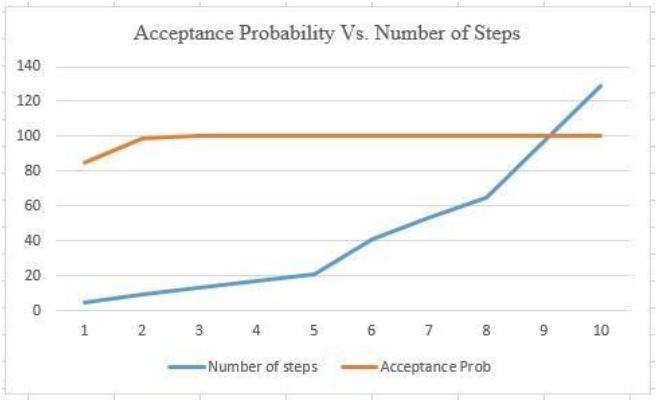

Figure 5. Acceptance Probability versus Bit Length

While in the case of response in fig.6, when the number of steps is varied from 9 to 17 through 13, the error rate is reduced by 98.98% because of an increase in accuracy. Improvement in error rate (accuracy) is at the cost of increased delay and power demand.

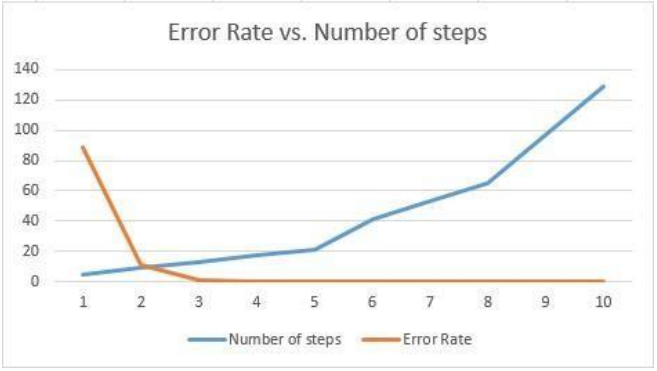

Figure 6. Error Rate versus Bit Length

7.1.2. Circuit Metrics

Any design is synthesized for basic circuit metrics like area, power dissipation, and critical path delay (latency). Along with that compound metrics like power-delay product (PDP), area-delay product (ADP), and energy-delay product (EDP) can also be studied.

Electronic Design Automation (EDA) tool is used for circuit design and performance evaluation. For evaluation of resource parameters, different process technologies and software compilers are available. Few of them are Cadence RTL compiler, predictive technology model (PMT) those are used for different values of foundries. The Monte Carlo algorithm is used as an assessment method for PVT analysis of a circuit.

Table 2 is showing circuit metrics analysis using the cadence–genus tool for synthesis.

Resource parameters chosen are area, power, and delay (latency). The circuit is analyzed under three different operating modes viz. full approximate mode (0.5 * bit-length), half approximate mode (0.75* bit-length), and accurate mode (1 * bit–length).

While in the case of response in fig.7, when the number of steps is varied from 9 to 17 through 13, the error rate is reduced by 98.98% because of an increase in accuracy. Improvement in error rate (accuracy) is at the cost of increased delay and power demand.

Table 2. Circuit Metrics Analysis

Bit Length	Resource Parameter	Number of steps		
		(0.5)*bit length	(0.75)*bit length	Equal to bit length
8	Area (unit)^2	215.1	233.2	263.3
	Power (mW)	2.466	6.368	6.356
	Delay(nS)	5.429	5.710	5.922
16	Area (unit)^2	650.1	732.906	776.340
	Power (mW)	4.2	12.079	12.081
	Delay(nS)	5.96	6.5	7.004
32	Area (unit)^2	2170.3	2558.844	2639.556
	Power (mW)	8.808	26.143	26.159
	Delay(nS)	7.046	8.147	9.172
64	Area (unit)^2	7867.3	9373.536	9522.99
	Power (mW)	17.139	51.246	51.299
	Delay(nS)	9.202	11.372	13.502
128	Area (unit)^2	29264.598	35249.94	36140.166
	Power (mW)	104.035	104.566	104.773
	Delay(nS)	13.602	17.941	22.231
256	Area (unit)^2	111244.392	136385.838	144509.706
	Power (mW)	210.029	212.010	212.937
	Delay(nS)	22.497	31.201	39.702

Fig.7 explaining about area demand w.r.t. bit length variation. When a circuit operates in an approximate mode, the area occupied is less because of the less number of hardware involved in generating results. The circuit occupies 29.90% more area when operated in full exact mode from full approximate mode in case of 256 bit-length.

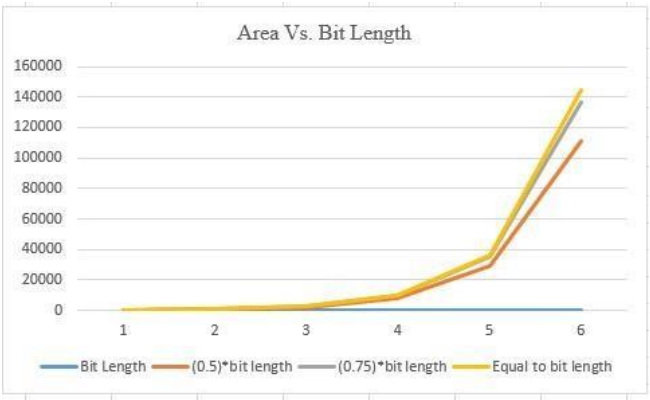

Figure 7. Area versus Bit Length

As mapped in fig.8, power demand is plotted versus bit length. When the number of steps is reduced (approximate mode), power (energy) demand is reduced. This circuit demands 1.38% more power when switching from approximate mode to exact mode for 256 bit-length. This means when we are moving from exact to approximate mode circuit becomes more energy efficient.

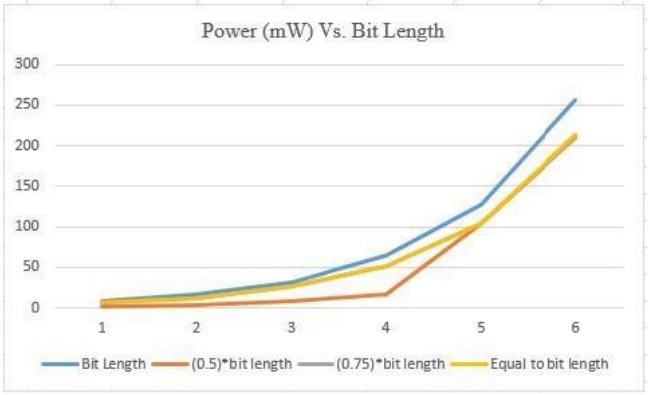

Figure 8. Power (Energy) versus Bit Length

Fig. 9 is all about Delay offered w.r.t. bit length. Delay increases by 9.08% when the circuit switches from approximate to the accurate mode for 256 bit-length.

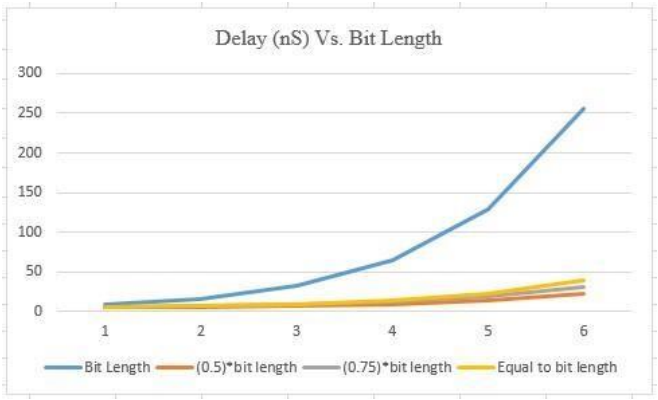

Figure 9. Delay Vs. Bit Length

Overall it is concluded that the circuit becomes more energy-efficient as well as faster at the cost of accuracy when switching from an accurate to an approximate mode of operation.

After complete analysis, it is concluded that this kind of approximate adder is more energy efficient and faster in its approximate mode of operation with a bounded compromise in accuracy.

8. APPLICATION DOMAIN OF AC

AC itself is covering a broad range of applications from data processing, filtering, and neural networks to big data applications and image processing. Each application domain is having a specific reason for incorporating approximation in either input or in methodology or in logic. Here are a few examples of these application domains which work on the principle of error resilience [16],[21].

Image Processing: In the case of image or signal processing applications, the inputs are accepted from discrete sensors or cameras that are already noisy in nature as compared to the analog sensor outputs. Still users unnoticed noise in input as well as output because of perceptual limitations.

AC makes use of this particular property for generating imperfect outputs with good acceptance probability [16], [17].

Computer Graphics: In case of Computer Graphics applications, the most important parameters are speed (frame rate) as well as computational resources especially needed for gaming. If these requirements are not meeting, then compromise in the accuracy of output for available frame rate so that erroneous (approximate) output can be accepted [16], [17].

Cloud Computing or Big Data Applications: These are falling under machine learning area. This category of applications is handling a huge data that is collected as well as processed from/to various resources or users and because of it, there is a huge possibility of having errors in the output for which the system and users are trained. This quality strongly supports approximate computing [16], [17].

Computer Vision: Mostly gaming and dynamic image processing applications are related to computer vision which covers motion tracking, pose estimation, and object detection by writing algorithms for it. Not all algorithms are optimal to achieve the required accuracy and generate exact digital images. Users are also not able to read exact images allowing them to add approximation in the output [17], [18].

Biometric Security: Biometric applications are a deadly combination of image processing, computer vision, and machine learning. These are data-rich applications including face recognition, fingerprinting, iris scanning, biometric signatures, etc. Biometric details are dedicated and unique w.r.t. individual. If the details vary minutely (or in a controlled manner) then also it will maintain its uniqueness w.r.t. that individual and where exactly it shows the scope for approximation [16],[17].

9. APPLICATIONS

Although AC is very promising still it will not be the first choice every time because of a few limitations like a proper selection of mode as well as approximation technique, also approximate code or data portion because all these parameters are application-oriented and cause quality loss with the same choice every time [21].

Usually, error rates less than 10% and PSNR values greater than 30dB are acceptable respectively in error-resilient and image processing applications respectively [21], [22]. When image processing applications are discussed, approximate multipliers with a lower error rate and higher acceptance probability outperform over the same circuit with a higher error rate for applications viz. image sharpening, image smoothening, brightness and contrast control, image compression [23]. But same thumb rule is not applicable for approximate adders as for complex operations, these adders do not give expected results.

The operand length and an approximation technique are the most important factors on which evaluation reports are dependent. After proper evaluation of the arithmetic circuit, its application domain is decided.

In case of image processing applications, the combination of approximate adder and multiplier is preferred to reconstruct a good quality processed image. Using either approximate adder or multiplier is not a good option to get the same result.

Applications like image sharpening are tolerating more approximation in addition process as compared to JPEG compression and face detection and alignment.

Approximate adders are mainly designed for reducing critical path delay by lowering error rates and improving circuit performance. But in the case of approximate adders as the carries are ignored or reduced throughout addition will be prone to generate single-sided errors that will become prominent in repeated or iterative addition. When approximate full adders are used at LSBs produce high ER but low power dissipation. So while selecting an approximate adder for any application the best trade-off should be achieved between error metrics and circuit metrics. Unsigned multipliers show improvement in circuit area by truncating part of PPs or part of LSBs in input operands at the cost of degraded accuracy. The tradeoff is between area and accuracy [4]. Few are showing a trade-off between energy and accuracy [4]. For signed multipliers, the booth multiplier shows promising results due to good error compensation as compared to TBMs. Other than that MRED[Mean Relative Error Distance] is a measure of accuracy in the case of multiplication applications like face detection.

10. CASE STUDY

To illustrate the applicability of the proposed technique to real-time use, it is applied to benchmark images as shown in fig.10. It is tested by implementing image processing applications like brightness and contrast control. Further same adder technique is used in approximate multipliers for partial product addition and showcasing the corresponding effect on chosen standard images.

As mentioned earlier, the application-level approximation technique is used where the approximation is introduced by managing the loop iterations.

We can apply the proposed adder in any of the machine learning algorithms where addition and subtraction are heavily performed. Output generated by an accurate adder is considered a golden reference for application.

Image b is the original image with brightness and contrast setting limits where it is showing the minimum and maximum range for brightness and contrast variation.

In this case study, the value of image parameters says brightness and contrast changes automatically after the application of proposed adder.

Image c is the resultant image generated by an accurate adder (when the number of steps=1*bitlength). It is exactly matching with the original one as brightness and contrast are maintained the same as the original image.

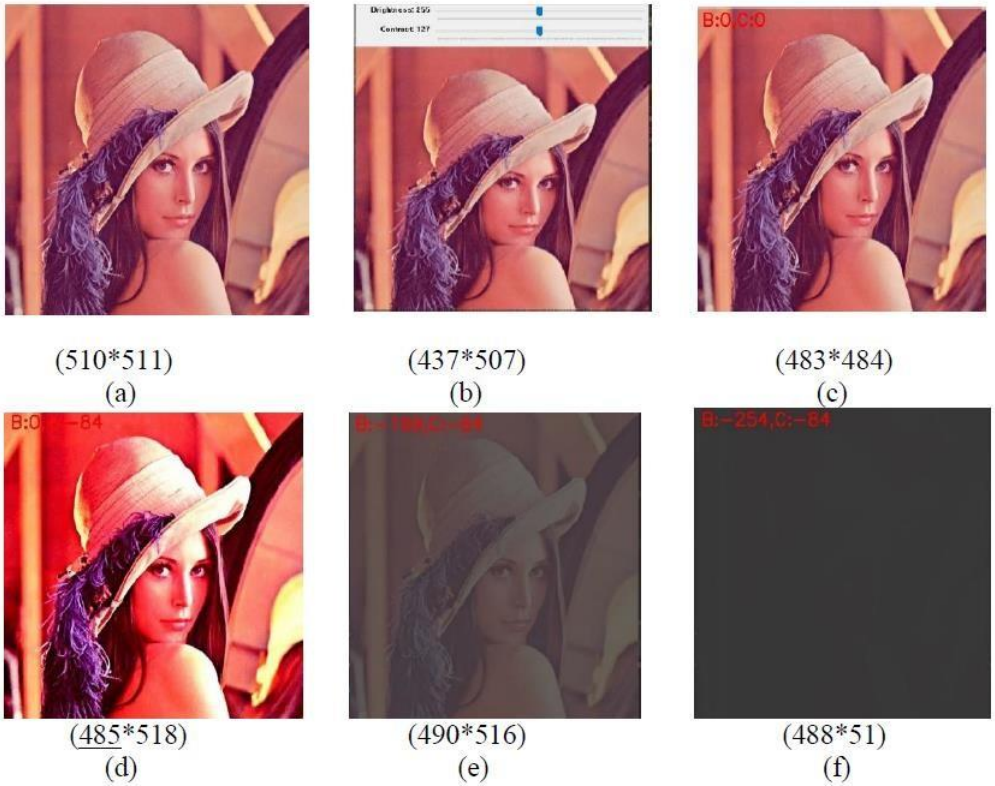

Figure 10. Original image (a) and output images using brightness and contrast control with accurate and approximate adder

Image d (when the number of steps=0.75*bit-length),e (when the number of steps=0.50*bitlength),f(when the number of steps=0.35*bit-length)are the outputs generated by an approximate adder when approximation increased progressively through the loop perforation technique. Brightness and contrast value changes automatically which deviates the output image from the original one. As per the required quality of the output, the level of approximation is used. Along with this practical application approach, through circuit metric analysis it is already proved that how proposed adder significantly reduced hardware resource consumption.

11. CONCLUSION, CHALLENGES, AND PROSPECTS

This article is an overall review of approximate circuits also proposing an energy-efficient approximate adder. It covers basics, related work, approaches for approximation and challenges, evaluation, applications, and future scope.

The purpose of designing Approximate Arithmetic adders is basically for improvement in power efficiency as well as critical path delay. But sometimes due to approximation techniques used the errors become prominent in complex applications, even though the error rates are low. Therefore, adders are showing less scope for approximation than multipliers. Approximate adders are always showing conflicts between circuit and error metrics. This conflict is cross verified with the help of a proposed technique.

On the other hand, approximate multipliers with lower error rates are showing promising results in complex applications. While selecting any multiplier tradeoff between accuracy and circuit metrics viz. area and energy is taken to be into an account.

In case of hardware platform, approximate computing has been proven by fabricating a silicon chip with approximate processors for recognition and data mining applications.

Nowadays approximate computing is quite settled and the major goal in it is automation. Due to this researchers are designing approximate computing systems for partial or total cognition to decide the implication of approximation. A literature survey for image processing applications shows that almost all applications are implemented for static systems and not for dynamic and there it seems a big scope for research work. Because run time error quality monitoring system (dynamic) is more complex, costly, and maintenance prone than compared to a static system. But these systems are more beneficial for working.

According to ITRS roadmap, to chase Moore's law, the semiconductor industry is progressing with 10nm microchips for production and looking for 7nm in the upcoming 5 years. But it increases the heavy Burdon on design constraints like supply voltage, frequency, yield along with operational faults. These hardware faults might propagate in software that can deflect the actual output. While applying approximate computing, we have to take care of these issues. Otherwise, approximate computing is really a good hope for green energy.

REFERENCES

[1] M. Ammar Ben Khadra, "An introduction to approximate computing", 10 Dec. 2017
[2] Source: Anand Raghunathan, Writer: Emil Venere, 'Approximate Computing' improves efficiency, saves energy, December 17, 2013
[3] Approximate Computing: Wikipedia.
[4] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124–137, Jan 2013.
[5] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 3, pp. 846–857, March 2016.
[6] B. S. Prabakaran, S. Rehman, M. A. Hanif, S. Ullah, G. Mazaheri, A. Kumar, and M. Shafique, "Demas: An efficient design methodology for building approximate adders for fpga-based systems," in 2018 Design, Automation Test in Europe Conference Exhibition (DATE), March 2018, pp. 917–920.
[7] S. Dutt, S. Nandi, and G. Trivedi, "Accuracy enhancement of equal segment based approximate adders," IET Computers Digital Techniques, vol. 12, no. 5, pp. 206–215, 2018.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
[8] M. Masadeh, O. Hasan, and S. Tahar, "Comparative study of approximate multipliers," CoRR, vol. abs/1803.06587, 2018. [Online]. Available: http://arxiv.org/abs/1803.06587
[9] W. Liu, T. Cao, P. Yin, Y. Zhu, C. Wang, E. E. Swartzlander, and F. Lombardi, "Design and analysis of approximate redundant binary multipliers," IEEE Transactions on Computers, vol. 68, no. 6, pp. 804–819, June 2019.
[10] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, "Invited: Cross-layer approximate computing: From logic to architectures," in 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), June 2016, 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 Fifteenth International Symposium on Quality Electronic Design, March 2014, pp. 263–269.
[12] Kulkarni, P., Gupta, P., & Ercegovac, M. (2011). Trading Accuracy for Power with an Underdesigned Multiplier Architecture. In 2011 24th Internatioal Conference on VLSI Design
[13] Chia-Hou Lin, Ing-Chao Lin, "High Accuracy Approximate Multiplier with Error Correction", in IEEE 31st International Conference on Computer Design (ICCD), Oct. 2013
[14] Amir Momeni, Jie Han, Paolo Montuschi, Fabrizio Lombardi, "Design and Analysis of Approximate Compressors for Multiplication", IEEE Transactions on Computers, Feb., 2014, pp. 984 – 994.

[15] Sana Mazahir, Osman Hasan, Rehan Hafiz, Muhammad Shafique, Jorg Henkel, "ProbabilisticError Modeling for Approximate Adders", IEEE Transactions on Computers, Vol. 66, No. 3, March 2017, pp. 515-530.
[16] S. Reda, M. Shafique, "Approximate Circuits-Methodologies and CAD", Springer Nature Switzerland, Springer, 2019.
[17] Naman Maheshwari , Approximate Computing and its Applications, Aug. 6, 2016
[18] Yi Wu, You Li, Xiangxuan Ge, Yuan Gao, Weikang Qian, "An Efficient Method for Calculating the Error Statistics of Block-Based Approximate Adders", Ieee Transactions On Computers, Vol. 68, No. 1, January 2019, PP. 21-38.
[19] Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, Massoud Pedram, "RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder", Ieee Transactions On Circuits And Systems—Ii: Express Briefs, Vol. 65, No. 8, August 2018, pp. 1089-1093.
[20] Sparsh Mittal, "A Survey Of Techniques for Approximate Computing", Article in ACM Computing Surveys · March 2016, pp1-34.
[21] Abbas Rahimi, Amirali Ghofrani, Kwang-Ting Cheng, Luca Benini, and Rajesh K Gupta, "Approximate associative memristive memory for energy-efficient GPUs", In Design, Automation & Test in Europe,2015, pp. 1497–1502.
[22] Honglan Jiang, Cong Liu, Liebo Liu, Faling to predictbrizio Lombardi, Jie, han, "A Review, Classification and Comparative Evaluation of Approximate Arithmetic Circuits", ACM Journal on Emerging Technologies in Computing Systems, Vol. 13, No. 4, Article 60, Pub. date: July 2017, pp. 60:1-60:37.
[23] Honglan Jiang, Francisco Javier Hernandez Santiago, Hai Mo, Leibo Liu, Jie Han, "Approximate Arithmetic Circuits: A Survey, Characterization, and Recent Applications", Proceedings of the IEEE, Volume: 108,Issue: 12, Dec. 2020, pp. 2108 – 2135.

AUTHORS

Viraj Joshi received her B.E. degree from Shivaji University and M.E. degree from Bharti Vidyapeeth. She is full time research schoar at Bits Pilani, K.K. Birla, Goa campus in EEE department. Her research interest includes Approximate arithmetic circuits.

Dr. Pravin Mane received his B.E. degree from RIT, Sakhrale , M. Tech. from IIT, Roorkee and Ph.D. from Bits Pilani, K.K. Birla, Goa Campus. Currently he is working as assistant professor in Bit Pilani, K.K. Birla, Goa campus. His area of interest is VLSI design.