International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 10 | Oct 2018 www.irjet.net p-ISSN: 2395-0072

A Review of Approximate Adders for Energy-Efficient Digital Signal Processing

Richa Pathak[1]

[1]Student, Dept. of Electronic Engineering., Institute of Engineering and technology, Lucknow
Dr APJ Abdul Kalam Technical University Lucknow, Uttar Pradesh, India-226021

Abstract - *With the breakdown of Dennard's Scaling the power densities in today's ICs are rapidly reaching unmanageable levels. So there is a need of new sources of efficiency. Approximate computing has emerged as a promising approach to energy-efficient design of digital systems. Approximate computing relies on the ability of many systems and applications to tolerate some loss of quality in the computed result. This loss in quality can be exploited to build circuits with smaller area, lower power and higher performance. The demand of high speed and power as well as the fault tolerance nature of some emerging applications have motivated for the development of approximate adders. Adder plays an important role in determining the speed and power consumption of a digital signal processing (DSP) system. This paper reviews recent progress in the area, and also provides a comparative evaluation in terms of circuit characteristics.*

Key Words: Approximate Computing, Approximate Adders, DSP, Dennard's Scaling, Energy-efficient Design.

1. INTRODUCTION

This With the continuous scaling of physical dimension of CMOS to a few tens of nanometres, it is becoming increasingly difficult to improve circuit performance and/or to enhance power. Approximate computing has been advocated as a new emerging approach for saving area and power dissipation, as well as increasing performance at a limited loss in accuracy[3]. While in general computation errors are not desirable, applications such as multimedia processing, wireless communications, recognition, machine learning and data mining are tolerant to some error. Due to the statistical nature of digital signal processing, small errors in computation would not impose noticeable degradation in performance [22]. Basically, there are two types of methodologies for improving speed and energy efficiency by approximation. The first one is voltage-over-scaling (VOS) technique for CMOS circuits to save power [4,15,19]. While the VOS technique is applicable to most circuit for error-tolerant application the problem with VOS technique is that errors mostly occur at the timing-critical path associated with the most significant bits (MSBs), i.e., errors are often large. The second technique is based on redesigning a logic circuit into its an approximate version. An approximate redesign pertains to the functionalities of different logic circuit. Many approximate adder designs have been developed. One such design [2] achieves 60% power reduction for discrete cosine transform (DCT) computation without making any discernible difference to the images being processed. In realistic practice, accuracy requirements may vary for different applications. Approximately redesigned adders are reviewed and a comparative evaluation is performed in this paper.

2. REVIEW

Adders are used for calculating the addition of two binary numbers. The two most basic adders are ripple-carry adder (RCA) and carry-lookahead adder (CLA) [9, 21]. In an n bit RCA, n 1-bit full adders (FAs)are cascaded; the carry of each FA is propagated to the next FA, thus the delay of RCA grow in proportion to n (or O(n)). An n-bit CLA consists of n SPGs, which operate in parallel to produce the sum, generate ($g_i = a_i b_i$) and propagate ($p_i = a_i + b_i$) signal, and connected to a carry look ahead generator. For CLA, all carries are generated directly by the carry look ahead generator using only the g_i and p_i signals, so the delay of CLA is logarithmic in n (or O(log(n))), thus much shorter than that of RCA. However, CLA needs larger circuit area and higher power dissipation. The carry lookahead generator becomes very complex for large values of n. The area complexity of CLA is O(n log(n)) when the fan-in and fan out of the constituent gates are fixed [16]. Many approximation schemes have been proposed by reducing the critical path and hardware complexity of the accurate adder. An early methodology is based on a speculative operation [16, 23]. In an n-bit speculative adder, each sum bit is found by its previous k less significant bits (LSBs) (k<n). A speculative design makes an adder significantly faster than the conventional design. Segmented adders are proposed in [7, 19, 27]. An n-bit segmented adder is implemented by several smaller adders operating in parallel. In this, the carry propagation chain is truncated into shorter segments. Segmentation is also utilized in [1, 5, 8,10, 12, 25], but their carry input for each sub-adders are selected differently. This type of adder is referred to as a carry select adder. Another method for reducing the critical path delay and power dissipation of a conventional adder is by approximating the full adder [2, 17, 20, 24]; the approximate adder is usually applied to the LSBs of an accurate adder. So, we can say that approximate adders can be mainly divided into four types: Speculative adders, Segmented adders, Carry select adders, Approximate full adders.

2.1 Speculative Adders

The carry chain is significantly shorter than n in most practical cases, [23] has proposed an almost correct adder (ACA) based on the speculative adder design of [16]. In an n-bit ACA, k LSBs are used to predict the carry for each sum bit (n > k), as shown in Fig. 1. Therefore, the critical path delay is reduced to O(log(k)) (for a parallel implementation such as CLA, the same below). As an example, four LSBs are used to calculate each carry bit in Fig. 1. As each carry bit needs a k-bit sub-carry generator in the design of [16], (n >k) k-bit sub-carry generators are required in an n-bit adder and thus, the hardware overhead is rather high. This issue is solved in [23] by sharing some components among the sub-carry generators. Moreover, a variable latency speculative adder (VLSA) is then proposed with an error detection and recovery scheme [23]. VLSA achieves a speedup of 1.5 on average compared to CLA.

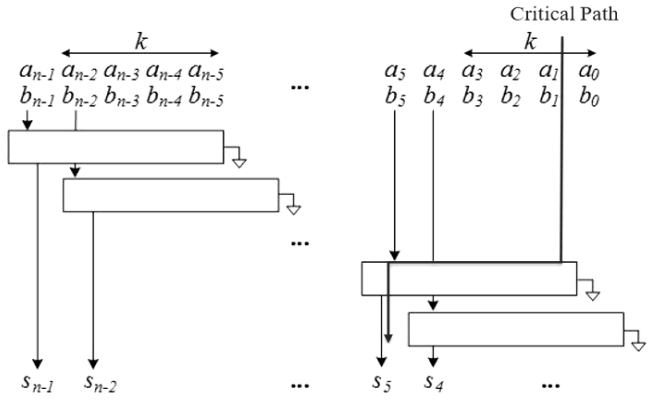

Fig 1: The Almost Correct Adder

2.2 Segmented Adders

2.2.1 The Equal Segmentation Adder (ESA)

A dynamic segmentation with error compensation (DSEC) is proposed in [19] to approximate adder. This scheme divides an n-bit adder into a number of smaller sub-adder; these sub-adders operate in parallel with fixed carry inputs. In this paper, the error compensation technique is ignored because the focus is on the approximate design, so the equal segmentation adder (ESA) (Fig. 2) is a simple structure of the DSEC adder. In Fig. 2, ⌈n k⌉ sub-adders are used, l is the size of the first sub-adder (l_ k), and k is the size of the other sub-adders. Hence, the delay of ESA is O(log(k)) and the hardware overhead is much less than ACA.

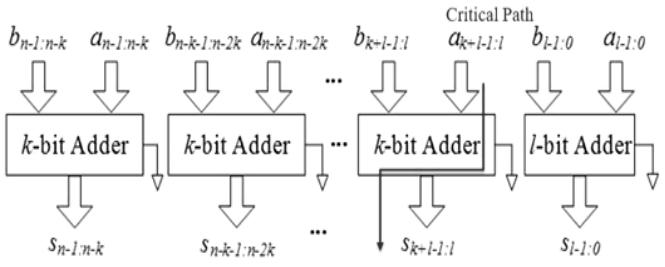

Fig 2: The n-bit equal segmentation adder (ESA) (l ≤ k)

2.2.2 The Error tolerant Adder Ⅱ type (ETA Ⅱ)

Another segmentation based approximate adder (ETAII) is proposed in [27]. Different from ESA, ETAII consist of carry generator and sum generator, as shown in Fig. 3 (n is the adder size; k is the size of the carry and sum generator). The carry signal from the previous carry generator is propagated in to the next sum generator. So, ETAII uses more information to predict the carry bit and it is more accurate as compared to ESA for the same value of k. Because the sub-adder in ESA produce both sum and carry, the circuit complexity of ETAII is almost same to ESA, however delay is larger (O(log(2k))). In addition to ETAII, several other error tolerant adders (ETAs) have been suggested by the same author in [26, 28, 29].

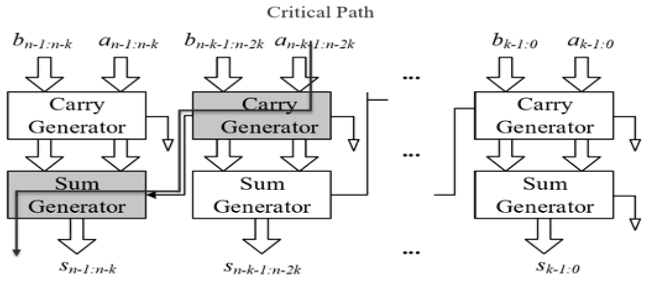

Fig 3: The n-bit error-tolerant adder type II (ETAII)

2.2.3 Accuracy Configurable Adder (ACA)

An accuracy-configurable approximate adder (ACA) is proposed in [7]. As accuracy can be configured at run time by changing the circuit structure, a tradeoff of accuracy against performance and power can be achieved. In a n-bit adder, [n/k1] 2k-bit sub-adder are required. Each sub adder adds 2k consecutive bits with an overlap of k bits, and all 2k-bit sub adders operate parallely to reduce the delay up to O(log(2k)). In each sub-adder the half most significant sum bit are selected as the partial sum. An error detection and correction (EDC) circuit is used to correct the error generated by each sub-adder.

2.2.3.1 Simple Accuracy Reconfigurable Adder (SARA)

A new carry-prediction based accuracy configurable adder design: simple accuracy reconfigurable adder (SARA) is proposed in [30]. It is a simple design with significantly less area than CLA. SARA inherits the advantages of all previous carry-prediction-based approaches: no error correction overhead, no data stall, and allowing graceful degradation. Compared to GDA, SARA incurs 50% less power-delay product (PDP) and can reach the same peak signal-to-noise ratio (PSNR). Moreover, SARA demonstrates remarkably good accuracy-power-delay tradeoff.

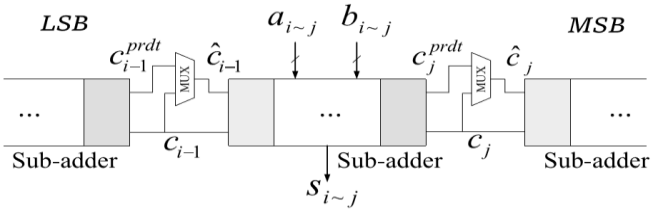

Fig 4: SARA

2.3 Carry Select Adders

In the carry select adder, several signals are used. There are different types of adders.

2.3.1 The Speculation Carry Select Adder (SCSA)

The Speculative Carry Select Adder is proposed in [1]. An n-bit SCSA consist of m =⌈n/k⌉ sub-adders (window adders). Each sub-adder is made of two k-bit adders: adder0 and adder1, as shown in Fig. 5. Adder0 has carry-in 0 while carry-in of adder1 is 1 the carry-out of adder0 is connected to a multiplexer to select the addition result as part of final result. SCSA and ETAII have the same accuracy for the same parameter k, because the same function is used to predict the carry for every sum bit. Compared with ETAII, SCSA use an additional adder and multiplexer in each block and hence, the circuit of SCSA is more complex than ETAII.

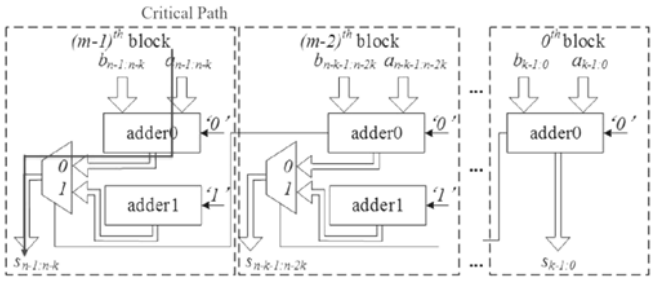

Fig 5: The n-bit speculative carry selection adder (SCSA)

2.3.2 The Carry Skip Adder (CSA)

Similar to SCSA, an n-bit carry skip adder (CSA) [8] is divided into ⌈n/k⌉ blocks, but each block consists of a sub-carry generator and a sub-adder. The carry-in of the $(i+1)^{th}$ sub-adder is determined by the propagate signals of the i^{th} block: the carry-in is the carry-out of the (i-1)the sub-carry generator when all the propagate signal are true (P_i = 1),otherwise it is the carry-out of the i^{th} sub-carry generator. Therefore, the critical path delay is O(log(2k)). This scheme improves the carry prediction accuracy.

2.3.3 The Gracefully-Degrading Accuracy-Configurable Adder (GDA)

An accuracy-configurable adder, also referred as the gracefully degrading accuracy-configurable adder (GDA), is presented in [25]. Control signal is used to configure the accuracy of GDA by selecting the accurate carry-in using a multiplexer for each sub-adder. The delay of GDA is determined by the carry propagation and control signals to multiplexers.

2.3.4 The Carry Speculative Adder

The carry speculative adder (CSPA) in [12] has one sum generator, two internal carry generators (one with carry-0 and one with carry-1) and one carry predictor in each block. The output of the i^{th} carry predictor is used to select carry signal for the $(i + 1)^{th}$ sum generator. l input bits (rather than k; l < k) in a block are used in carry predictor. Therefore, the hardware overhead is reduced significantly as compared to SCSA.

2.3.5 The Generate Signal Exploited Carry Speculation Adder

In [5], the generate signal are used for carry speculation. GSCA has a similar structure of CSA. The only difference is the carry selection; the carry-in for the $(i + 1)^{th}$ sub-adder is selected by its own propagate signals. The carry-in is the most significant generate signal g_i k-1of the i^{th} block if $P_i = 1$, else it is the carry-out of the i^{th} sub-carry generator. The critical path delay of GCSA is O(log(2k)) due to the carry propagation. This carry selection scheme effectively controls the maximal relative error.

2.4 Approximate Full Adders

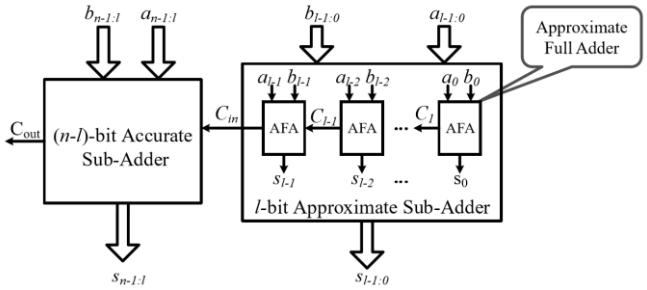

Fig 6: The n-bit approximate adder using approximate full adders

2.4.1 The Lower-Part-OR Adder

LOA [17] divide a n-bit adder into a (n-l)-bit more significant sub-adder and an l-bit less significant sub-adder. For the less significant sub-adder, its input is simply processed by using OR gate (as a simple approximate full adder). The more significant (n-l)-bit sub-adder is accurate adder. An extra AND gate is used to generate the carry-in signal for the more significant sub-adder. The critical path of LOA is from the AND gate to the most significant sum bit of the accurate adder, i.e., O(log(n-l)). LOA has been used in a recently suggested approximate floating-point adder [14].

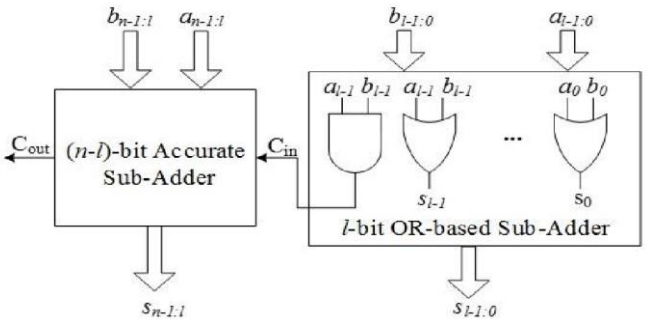

Fig 7: The n-bit lower-part-OR adder (LOA)

2.4.2 Approximate Mirror Adders (AMAs)

In [2], five AMAs are suggested by decreasing the number of transistors and the internal node capacitance of the mirror adder (MA). The AMAs adder cells are then used in LSBs of a multiple-bit adder. However, the critical paths of AMA1-4 are longer than LOA because the carry propagate through every bit. As for AMA5, the carry-out is one of the inputs; thus, no carry propagation exist in the LSBs of an approximate multiple bit adder.

2.4.3 Approximate Full Adders using Pass Transistors

Three approximate adders (AXAs) based on XOR/XNOR gates and multiplexer have been proposed in [24]. Many approximate complementary pass transistor logic (CPL) adders have been suggested by reducing the number of transistors in the accurate CPL adder [20].Significant area and power savings have been obtained for all types of approximate design.

3. COMPARATIVE EVALUATION

3.1 Delay and Area Comparison

Analysis of delay and circuit complexity of approximate adders is presented in Table 1.

3.2 Features Comparison

Comparison of characteristics for different techniques is presented in Table 2. [30]

3.3 Accuracy Comparison

16-bit approximate adders with an equivalent 8- bit carry propagation are compared in terms of error rate (ER) and mean relative error distance (MERD). ETAII, ACAA and SCSA have the same error characteristics. The carry select adders (CSA, CSPA, CCA, GCSA) and the speculative adder (ACA) are very accurate with small values of ER and MRED (except for CSPA). The approximate full adder (LOA) has a moderate MRED but very large ER. The segmented adders (ESA, ETAII, ACAA) are not very accurate. The truncated adder (TruA) is the least accurate in terms of ER among the equivalent designs.

Error rate (ER) is the probability of producing an incorrect result and MRED (mean relative error distance (RED) is used to evaluate the mean relative difference between an approximate result and the accurate result.

Table 1 : Analysis of delay and circuit complexity

Adder Type		Adder Name	Delay	Critical area
Conventional Adders		RCA	$O(n)$	$O(n)$
		CLA	$O(log(n))$	$O(nlog(n))$
Approximate Adders	Speculative Adders	ACA	$O(\log(k))$	$O((n-k)\log(k))$
	Segmented Adders	ESA	$O(\log(k))$	$O(nlog(k))$
		ETA II	$O(\log(k))$	$O(nlog(k))$
		ACAA	$O(\log(k))$	$O((n-k)\log(k))$
	Carry Select Adders	SCSA	$t_{adder} + t_{mux}$	$A_{adder}+A_{mux}$
		CSA	$O(\log(k))$	$A_{adder}+A_{carry}$
		CSPA	$t_{adder} + t_{mux}$	$A_{adder}+A_{mux}+A_{carry}$
		CCA	$t_{adder} + t_{mux}$	$A_{adder}+A_{mux}$
		GCSA	$O(\log(k))$	$O(nlog(k))$
	Approximate Full Adders	LOA	$O(\log(n-1))$	$A_{loa}+(l \times A_{OR})$

$t\ adder$: $(\log(k))$ A_{adder}: $O(n\log(k))$ A_{loa}: $O((n-l)\log(n-l))$

Table 2 : Comparison of characteristics for different techniques

Method	Baseline sub-adder	Error Correction	Graceful degradation	Carry Prediction
ACA	Redundant CRA	Yes	No	No
GeAr	Redundant CRA	Yes	No	No
Accurus	Redundant CRA	Yes	Yes	No
GDA	CRA	No	Yes	Stand-alone
RAP-CLA	CLA	No	Yes	Reuse
SARA	CRA	No	Yes	Reuse

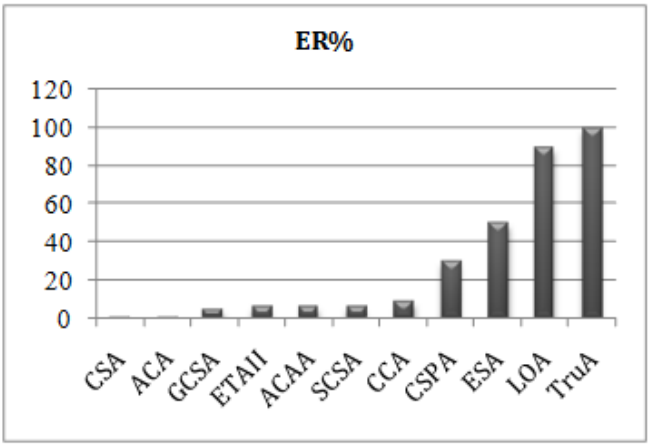

Chart -1: Error Rate of Approximate Adders

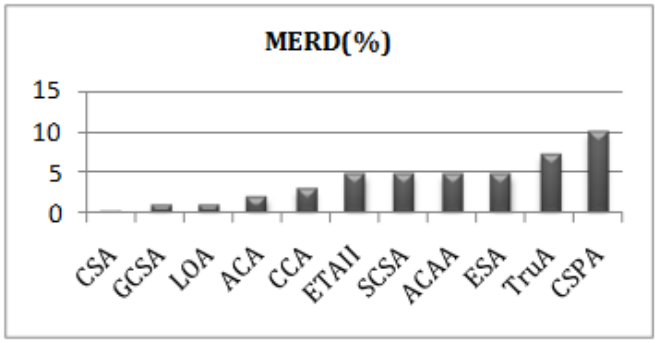

Chart -2: Mean Relative Error Distance of Approximate Adders

3.4 Hardware Comparison

Delay and power of 16-bit equivalent approximate adders is compared in chart-3 and chart-4. The carry select adders (CSA, CSPA, CCA, GCSA) tend to consume large power at a relatively high performance. The speculative adder (ACA) is very fast but very power consuming. The approximate full adder (LOA) is slow, but it consumes a low power and area. The segmented adders (ESA, ETAII, ACAA) are power and area efficient. The truncated adder (TruA) is very power and area efficient, but with a relatively long delay.

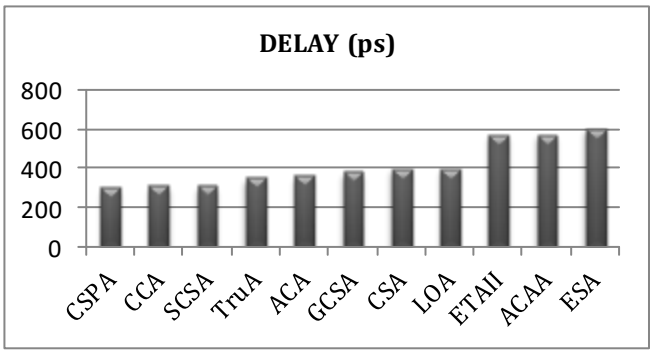

Chart -3: Delay of Approximate Adders

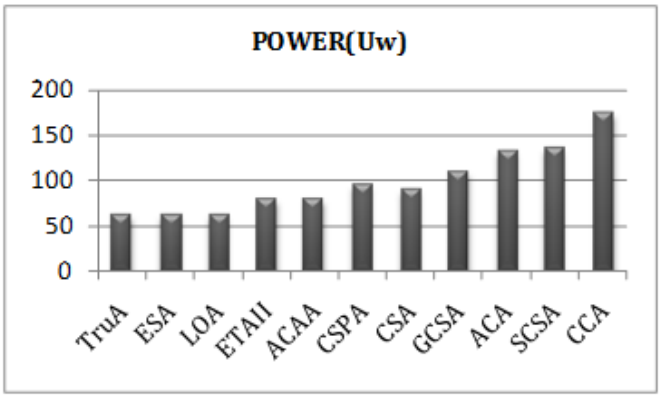

Chart -4: Power of Approximate Adders

4. CONCLUSION

After reviewing all approximate adders we come to a conclusion that (ESA) is the most hardware efficient design, but it has the lowest accuracy in terms of error rate (ER) and mean relative error distance (MRED). The error-tolerant adder type II (ETAII), the speculative carry select adder (SCSA) and the accuracy-configurable approximate adder (ACAA) are equally accurate however ETATII incurs the lowest power-delay product (PDP) among them. SARA has considerable lower area overhead than almost all the previous works The most power consuming scheme with a moderate accuracy is the almost correct adder (ACA). The lower-part-OR adder (LOA) is the slowest, but it is highly efficient in power dissipation.

REFERENCES

[1] K. Du, P. Varman, and K. Mohanram. High performance reliable variable latency carry select addition. In DATE, pages 1257{1262, 2012.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. Low-power digital signal processing using approximate adders. IEEE Trans. CAD, 32(1):124{137, 2013.

[3] J. Han and M. Orshansky. Approximate Computing: An Emerging Paradigm for Energy efficient Design. In ETS, pages 1{6, Avignon, France, 2013.

[4] R. Hegde and N. Shanbhag. Soft digital signal processing. IEEE Trans. VLSI Syst., 9(6):813{823, 2001.

[5] J. Hu and W. Qian. A new approximate adder with low relative error and correct sign calculation. In DATE, in press, 2015.

[6] J. Huang, J. Lach, and G. Robins. A methodology for energy-quality tradeof using imprecise hardware. In DAC, pages 504{509, 2012.

[7] B. Kahng and S. Kang. Accuracy-configurable adder for approximate arithmetic designs. In DAC, pages 820{825, 2012.

[8] Y. Kim, Y. Zhang, and P. Li. An energy efficient approximate adder with carry skip for error resilient neuromorphic vlsi systems. In ICCAD, pages 130{137, 2013.

[9] Koren. Computer Arithmetic Algorithms, 2nd edition. A K Peters/CRC Press, Natick, 2001.

[10] L. Li and H. Zhou. On error modeling and analysis of approximate adders. In ICCAD, pages 511{518, 2014.

[11] Liang, J. Han, and F. Lombardi. New metrics for the reliability of approximate and probabilistic adders. IEEE Trans. Comput., 62(9):1760{1771, 2013.

[12] Lin, Y. Yang, and C. Lin. High-performance low-power carry speculative addition with variable latency. IEEE Trans. VLSI Syst., in press, 2014.

[13] C. Liu, J. Han, and F. Lombardi. An analytical framework for evaluating the error characteristics of approximate adders. IEEE Trans. Comput., 2014.

[14] W. Liu, L. Chen, C. Wang, M. ONeill, and F. Lombardi. Inexact oating-point adder for dynamic image processing. In IEEE-NANO, pages 239{243, 2014.

[15] Y. Liu, T. Zhang, and K. Parhi. Computation error analysis in digital signal processing systems with over scaled supply voltage. IEEE Trans. VLSI Syst., 18(4):517{526, 2010.

[16] S.-L. Lu. Speeding up processing with approximation circuits. Computer, 37(3):67{73, 2004.

[17] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas. Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft - Computing Applications.IEEE Trans. Circuits Syst., 57(4):850{862, 2010.

[18] Miao, K. He, A. Gerstlauer, and M. Orshansky. Modeling and synthesis of quality-energy optimal approximate adders. In ICCAD, pages 728{735, 2012.

[19] D. Mohapatra, V. Chippa, A. Raghunathan, and K. Roy. Design of voltage-scalable meta-functions for approximate computing. In DATE, pages 1{6, 2011.

[20] D. Nanu, R. P. K., D. Sowkarthiga, and K. S. A. Ameen. Approximate adder design using cpl logic for image compression. International Journal of Innovative Research and Development, 3(4):362{370, 2014.

[21] B. Parhami. Computer Arithmetic: Algorithms and Hardware Designs, 2nd edition. Oxford University Press, New York, 2010.

[22] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan. Macaco: Modeling and analysis of circuits for approximate computing. In ICCAD, pages 667{673, 2010.

[23] K. Verma, P. Brisk, and P. Ienne. Variable latency speculative addition: A new paradigm for arithmetic circuit design. In DATE, pages 1250{1255, 2008.

[24] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi. Approximate XOR/XNOR-based adders for inexact computing. In IEEE-NANO, pages 690{693, 2013.

[25] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu. On reconfiguration-oriented approximate adder design and its application. In ICCAD, pages 48{54, 2013.

[26] N. Zhu, W. L. Goh, G. Wang, and K. S. Yeo. Enhanced low-power high-speed adder for error-tolerant application. In SOCC, pages 323{327, 2010.

[27] N. Zhu, W. L. Goh, and K. S. Yeo. An enhanced lowpower high-speed adder for error-tolerant application. In ISIC 2009, pages 69{72, 2009.

[28] N. Zhu, W. L. Goh, and K. S. Yeo. Ultra low-power high speed exible Probabilistic Adder for Error-Tolerant Applications. In SOCC, pages 393{396, 2011.

[29] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong. Design of Low-Power High-Speed Truncation Error-Tolerant Adder and Its Application in Digital Signal Processing. IEEE Trans. VLSI Syst., 18(8):1225{1229, 2010.

[30] Xu Wenbin, Sapatnekar S. Sachin and Hu Jiang, A simple yet efficient accuracy configurable adder design, IEEE Trans. on VLSI Systems, Jan 2018.

BIOGRAPHY

Richa Pathak received the B. Tech degree in electronics and communication engineering from SRMCEM, affiliated to Dr. APJ AKTU, Lucknow, India and is currently working towards her M. Tech degree in Microelectronics with research interest in Digital VLSI and Approximate Computing from Institute of Engineering and Technology, Lucknow, Uttar Pradesh.