Can Emerging Computing Paradigms Help Enhancing Reliability Towards the End of Technology Roadmap?

Runsheng Wang[1*], Zuodong Zhang[1], Yawen Zhang[1], Yixuan Hu[1], Yanan Sun[2], Weikang Qian[3], and Ru Huang[1]
[1]Institute of Microelectronics, Peking University, Beijing, China
[2]Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China
[3]UM-SJTU Joint Institute and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China
[*]Email: r.wang@pku.edu.cn

2021 IEEE International Reliability Physics Symposium (IRPS) | 978-1-7281-6893-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/IRPS46558.2021.9405167

***Abstract*—With CMOS technology shrinking into nanoscale, the circuit design margin has become extremely tight due to the severer transistor aging and process variations. To relieve the circuit reliability problems, many design optimization methods have been proposed. In essence, all these methods trade off area/power /performance for reliability. In this paper, we present a new perspective to enhance design reliability: using emerging computing paradigms. As the preliminary attempts, three reliability-enhanced design flows based on approximate computing and/or stochastic computing are demonstrated. The results show that some emerging computing paradigms are inherently robust, or can trade off computing accuracy for reliability, which provides the designers with much more flexibility. It also indicates that emerging computing paradigms are very promising for circuit design with ultimately scaled CMOS and beyond CMOS devices.**

Keywords- reliability-enhanced design, NBTI, circuit reliability simulation, aging-aware STA, SSTA, stochastic computing, approximate computing, ReRAM crossbar, neural network.

I. INTRODUCTION

With CMOS technology continuously shrinking, the reliability issues have become more and more pronounced. Many kinds of non-ideal factors, such as process variations and transistor aging effects (especially, the negative bias temperature instability, NBTI), make the circuit design margin getting smaller [1-4]. To counteract the impact of aging and variations, the needed voltage/frequency guardband is getting larger, too. Thus, the benefits of technology scaling are smaller at advanced technology nodes.

To relieve the circuit design margin, solutions from the device level to the architecture level have been proposed, such as aging-aware voltage and frequency scaling [5, 6] and aging sensor [7]. These methods aim to avoid timing errors, which sacrifice speed for reliability. Other solutions like aging control gate [8] or gate resizing [9], can enhance the circuit reliability by reconstructing the circuit, but will bring additional overhead in area or power.

Recently, emerging computing paradigms and emerging devices are flourishing and attract more and more attention. It naturally raises a question that, whether it is possible to take advantage of emerging computing paradigms to help enhancing the circuit reliability. Therefore, as the first attempt, we present three design optimization methods to enhance circuit reliability as examples, which are all based on emerging computing paradigms.

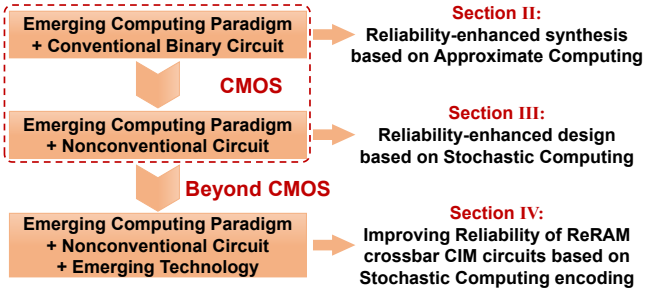

Figure 1. The overview of this paper.

The organization of this paper is shown in Fig. 1. The first example utilizes approximate computing to completely remove the guardband; the second example shows how stochastic computing (SC) inherently improve circuit reliability; the last example shows a variation-resistant ReRAM crossbar computing-in-memory (CIM) circuit based on stochastic coding. The results provide new insights into the reliability-enhanced design for nanoscale and emerging technologies.

II. RELIABILITY-ENHANCED SYNTHESIS BASED ON APPROXIMATE COMPUTING

Approximate computing is a promising emerging computing paradigm which has attracted a lot of attention in recent years [10]. It intentionally introduces some errors while ensuring the usability of the application, in exchange for smaller area and/or power. It has been demonstrated that approximate computing can improve energy efficiency in many applications that can tolerate some loss of accuracy, such as neural networks, data mining, and image processing [11]. However, most previous works on approximate computing focus on improving the energy efficiency without paying attention to the reliability of the approximate circuit. In Ref. [12], a reliability-enhanced design method was proposed, which truncates the low bits of adders to reduce the aged delay. However, this method is not applicable to other arithmetic units or function units, and the truncation is not the optimal approximation for adders [13].

This work is partly supported by the National Key R&D Program of China (2020YFB2205502), NSFC (61874005, 61927901) and the 111 Project (B18001).

978-1-7281-6893-7/21/$31.00 ©2021 IEEE

Thus, we have proposed a reliability-enhanced design method based on approximate logic synthesis (ALS) [14], as shown in Fig. 2. The design flow has two key parts: a reliability simulation flow supporting statistical static timing analysis (SSTA), and an approximate logic synthesis flow which can effectively reduce the circuit delay. After the forward reliability simulation flow, the aged path failure rates (PFRs) of the critical paths can be estimated by aging-aware SSTA. If the largest PFR is higher than the given threshold, the timing errors will destroy the functionality of the circuit. In this case, the netlist needs to be processed backward by approximate logic synthesis. Then, the reliability of the approximate circuit is checked again. This procedure is repeated until the reliability requirement is satisfied.

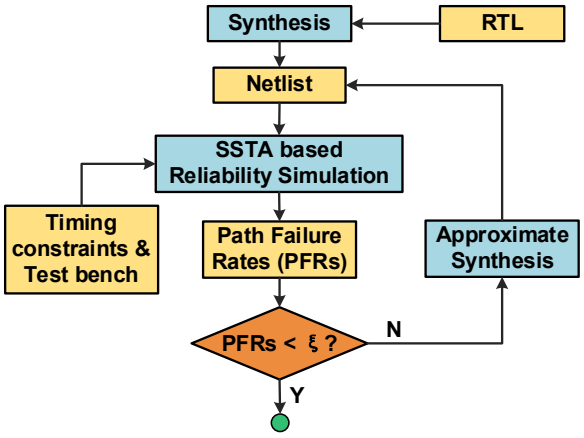

Figure 2. Flow diagram of the proposed reliability-enhanced design flow based on approximate synthesis.

A. *SSTA-based Reliability Simulation Flow*

The goal of reliability simulation is to analyze the timing after aging. In digital applications, NBTI dominates the transistor aging [15-17]. NBTI in digital circuits depends on the working frequency and the duty factor (DF) of each transistor. Therefore, as shown in Fig. 3, the reliability simulation flow is divided into two parts: workload analysis which calculates the degradation of transistors, and timing analysis after aging which calculates the path delay distributions.

The most accurate method to calculate the degradation of transistors is the SPICE-level simulation of the whole netlist with application programming interface (API) like Synopsys MOSRA, Cadence RelXpert, TSMC model interface (TMI), or the CMC open model interface (OMI). However, the SPICE-level simulation of VLSI circuits is not practical. Therefore, the proposed flow divides the workload analysis into two steps: the first step selects the *N*-worst paths of the circuit and uses gate-level simulation to obtain the input waveform of these paths; the second step only simulates the netlists of paths and calculates the DF of each transistor. After obtaining the DF of each transistor, a long-term aging model together with a variation model [14] is used to calculate the NBTI degradation and process variations.

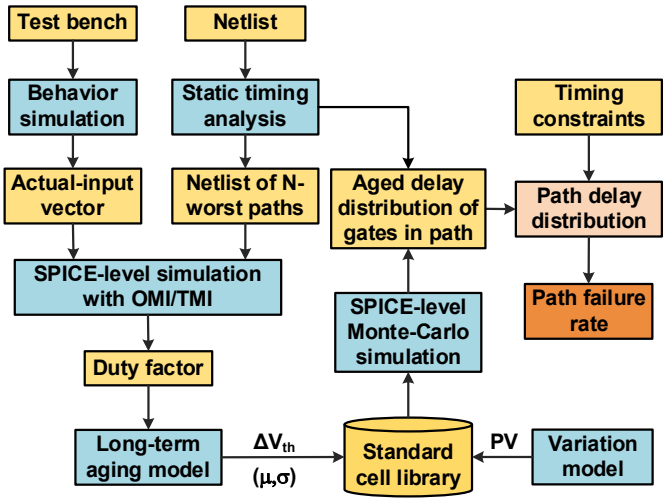

Figure 3. Flow diagram of the proposed SSTA-based reliability simulation flow.

For timing analysis, the SPICE-level Monte Carlo simulation is used to calculate the delay distribution of each gate and then calculate the path failure rate according to the working frequency. Although SPICE-level simulation is time-consuming, considering that the number of paths will not be large, the simulation time is acceptable.

B. *Delay-driven Approximate Logic Synthesis*

In the proposed reliability-enhanced design flow, the approximate logic synthesis is to find the optimal approximate local change (ALC), which can reduce the delay significantly and have the least error impact on the application. Our synthesis algorithm works on the AND-inverter graph (AIG) representation, and the basic procedure is presented in Fig. 4. In AIG representation, the circuit delay is proportional to the depth of AIG. To reduce the delay, the depth of the AIG needs to be reduced. The subgraph containing all critical paths is called the critical graph, and a cut on all critical paths is called a critical cut. For example, the cut with nodes 8 and 9 in Fig.4 is a critical cut. The ALS algorithm aims at finding the optimal critical cut in the critical graph, which has the minimal error impact on the circuit.

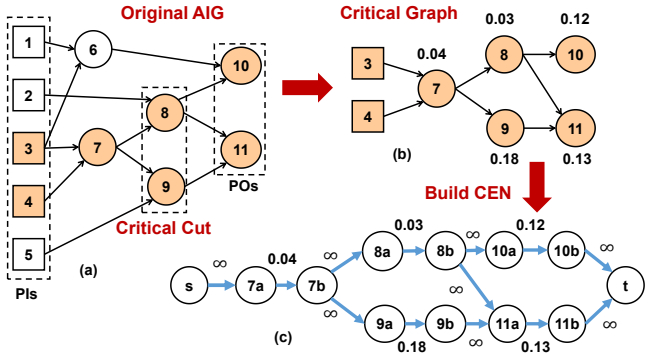

Figure 4. Illustration of delay-driven approximate logic synthesis algorithm.

For a large AIG graph, enumerating all sets of ALCs and critical cuts is too time-consuming. Therefore, to reduce the complexity, the algorithm transforms the optimization problem into a network flow problem. First, simulate the error impact of each ALC. Then, map the original critical graph into

a critical error network (CEN). Finally, obtain the minimum cut of the CEN by solving a maximum flow problem on CEN. The minimum cut corresponds to a good critical cut. After the good critical cut is found, the ALCs on the cut are applied to simplify the AIG. Finally, the simplified AIG is mapped into a gate-level netlist.

C. Results and Discussions

To examine the effectiveness of the proposed design flow, some ISCAS'85 benchmark circuits are tested. We use Synopsys Design Compiler to synthesize the benchmark circuits. Synopsys PrimeTime is used to perform fresh timing analysis and report the critical paths. The open-source Nangate 15 nm standard cell library [18] is employed to obtain the delay and area of the circuit.

Table I shows the results of different benchmark circuits. It can be seen that with conventional aging-aware design, an additional guardband of reducing 6%~10% frequency should be added for the resistance of 10-year aging. Note that this additional aging guardband is at the similar amount with the impacts of process variations. However, if using the proposed reliability-enhanced approximate (REA) design, a small sacrifice of some accuracy can completely eliminate aging guardband (i.e., zero additional guardband). Especially when performed with arithmetic circuits without controller (e.g., ALU4 and APEX6), the error rate caused by approximation is less than 0.4%, while the required aging guardband of the original circuit is about 6.5%~7.5%. Note that, a small error rate (<5%) is acceptable for most error-tolerant applications.

TABLE I. RESULT ON DIFFERENT BENCHMARK CIRCUIT

Bench	*Gates*	*I/Os*	*Aging guardband*	*Error rate @ Zero guardband*
C1355	546	41/32	8.16%	5.95%
C1908	880	33/25	9.04%	2.10%
C3540	1669	50/22	8.15%	3.32%
C5315	2307	178/123	6.55%	1.41%
ALU4	681	14/8	6.49%	0.33%
APEX6	452	135/99	7.27%	0.28%

For a case study, the image compression application is chosen to demonstrate the system reliability enhancement of the proposed design flow. The compression algorithms include discrete cosine transformation (DCT) and inverse discrete cosine transformation (IDCT). We use an 8-bit multiplier and a 16-bit adder as the arithmetic units. Fig. 5 shows the output image processed by different circuits. After 10 years of aging, the image quality of the original circuit is greatly reduced due to timing errors. Because timing errors are more likely to occur on longer paths, that is, the more significant bits of the adder, it will seriously affect the computing result. In contrast, for the circuit with the proposed REA design, although its initial PSNR decreases slightly (less than 1 dB), its performance after aging does not decrease due to the shortening of the critical path. The results indicate that, the proposed design flow can convert the timing violations that seriously affect the circuit functions into the deliberately induced errors that have negligible impact in practical applications.

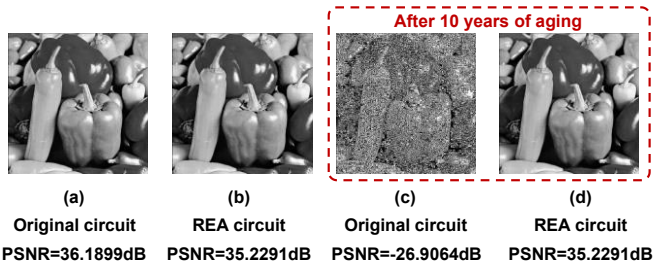

Figure 5. Images processed by the (a) fresh original circuit, (b) fresh REA circuit, (c) aged original circuit, and (d) aged REA circuit.

III. RELIABILITY-ENHANCED DESIGN BASED ON STOCHASTIC COMPUTING

Apart from approximate computing, stochastic computing (SC) is also an attractive emerging computing paradigm. SC processes data as the probability of 1 appearing in a bitstream, so the circuit in SC is not the conventional binary circuit. Many arithmetic operations can be implemented with simple logic gates [19-21]. In addition to low power and high area-efficiency, SC exhibits high fault-tolerance because of its operations with probability instead of binary numbers. It raises the concern of whether the reliability of SC circuits (SCCs) is enhanced as compared with that of conventional binary circuits.

Therefore, using the reliability simulation flow presented in Section II-A, the reliability of SC circuits in practical applications is investigated and compared with that of binary circuits. The Robert cross edge detector [22] for image recognition is chosen as the benchmark application. Using the correlated input sequence, the absolute value subtractor can be implemented by an XOR gate in SC, while the corresponding binary circuit requires two 8-bit absolute value subtractors and an 8-bit adder. For a fair comparison, the bit stream generator (BSG) is included in the SC circuit.

A. Workload Comparison

The workload distributions under different input images are shown in Fig. 6. It is noted that the whole netlist is used for SPICE-level simulation to get the duty factors of all the internal nodes. It can be seen that the DFs of the internal nodes are mostly around 50% in the SC circuit, while the DFs are widely distributed in the binary circuit and many nodes' DFs are close to 1. Since the binary circuit is much more complex, the internal nodes are more likely to be biased [23]. However, in the coding format of SC, each bit is equally weighted and randomized, so the switching activity is relatively high and DF is usually not too large.

High DF means the transistor is in the stress state for most of the time, so the traps accumulate, which results in larger ΔV_{th}. If the gates with high DFs locate on the critical paths, the circuit delay will increase significantly.

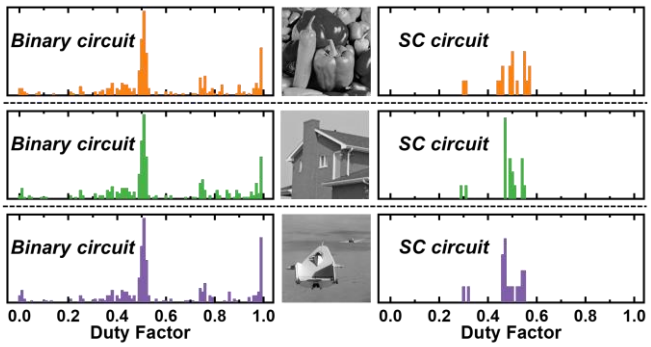

Figure 6. Workload comparison between the binary circuit and the SC circuit under different input images. The Y-axis in each figure is the normalized occurrence.

B. *Performance Degradation of Circuits*

After workload analysis, the path delay and path failure rate after aging can be estimated. The results show that the path failure rate of the SC circuit (including BSG) is lower than that of the binary circuit [Fig. 7(a)]. To quantitatively evaluate the image quality, the s-component of structure similarity index (MSSIM) is used to evaluate the quality of the output image. As shown in Fig. 7(b), the quality of the image processed by SCC is higher even at the same path failure rate, which means that the computing paradigm is highly robust. The above results prove that the reliability enhancement of SC originates from the low degradation of the SC circuit and the robustness of the coding format.

The images processed by different circuits is shown in Fig. 8. It can be seen that the image processed by the binary circuit loses most of the information after 10 years of aging [Figs. 8(a-c)]. While the quality of the image processed by the SCC is barely reduced [Figs. 8(e-g)]. If using the conventional aging-aware circuit design method and setting the optimization goal to maintain the performance (e.g. MSSIM > 0.9) at the end of life [Fig. 7(c)], the binary circuit needs to deploy a very large aging guardband of 16% [Fig. 8(d)]. In contrast, the MSSIM of the SC circuit is above 0.9 without any aging guardband. Thus, a part of the precision can be sacrificed in exchange for speed. The precision of SC is determined by the bitstream length (BSL). As shown in Fig. 8(h), after reducing BSL from 256 to 64 bits, the SC circuit still has the required performance, while reducing the total delay by 4 times.

The performance of the different circuits under different working conditions is shown in Fig. 9. Because NBTI is sensitive to the temperature, a slight increase in the temperature will cause a large increase of ΔV_{th}, which will significantly reduce the performance of the binary circuit. For the SCC, due to the low DF, the performance loss is less. Large process variations will increase the uncertainty, and make the path failure rate higher. However, the performance of SCC is not as obvious as the binary circuit because of its coding format and lower degradation. The results prove that the SCC is insensitive to the working conditions, which will reduce the design complexity.

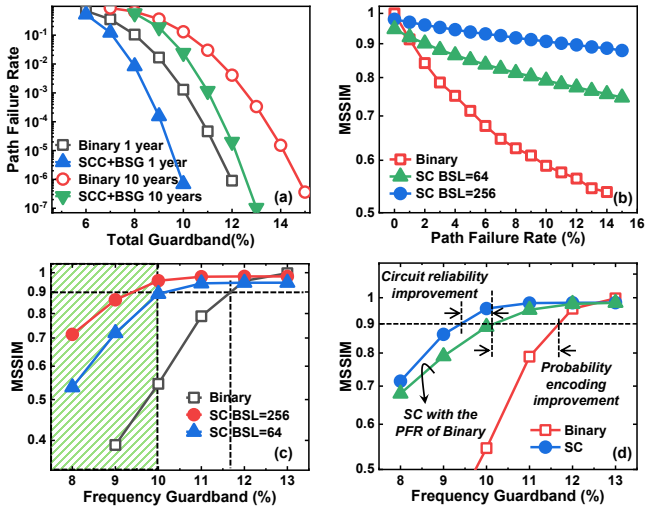

Figure 7. (a) The relationship between path failure rate and total guardband under different degradation times. (b) The relationship between MSSIM (quality index of image) and path failure rate. (c) The relationship between MSSIM and total guardband. (d) The origin of the inherent reliability improvement of SC.

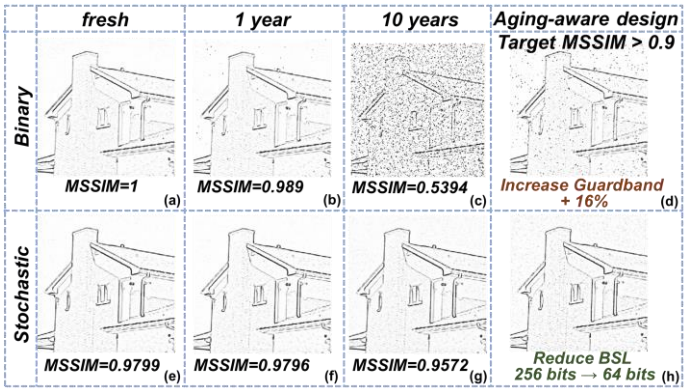

Figure 8. Images processed by the binary circuit (a-c) and the SC circuit (e-g) after 1 year and 10 years of aging. Using aging-aware design optimization to ensure the quality of images in the binary circuit (d) and the SC circuit (h).

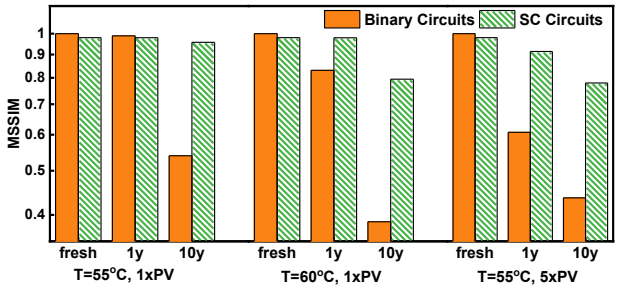

Figure 9. The quality (MSSIM) of images processed by the binary circuit or the SC circuit under different aging times, for different process variations (PV) and/or temperatures.

C. *The Origins of the Reliability Enhancement*

It has been mentioned that the reliability enhancement comes from both the low degradation of circuits and the fault-tolerance of the probability encoding format. To figure out the proportion of the two parts, the quality of images processed by SC circuits with the path failure rate of binary circuits is shown in Fig. 7(d). The result shows that the improvement of the circuit reliability also accounts for a considerable proportion,

which was not expected before. Besides, since the reliability of circuits is independent of BSL, the proportion of circuit reliability enhancement will be larger when BSL is shorter.

IV. Reliability-enhanced ReRAM Crossbar CIM based on Stochastic Computing Encoding

Neural networks have shown great promise for a wide range of applications, including image classification and speech recognition. Deep learning also created the demand for energy-efficient hardware accelerators. However, neural networks contain a huge number of matrix-vector multiplication operations. Their performance is limited by the traditional von Neumann architecture. Computing-in-memory is proposed to reduce the data movement between processor and memory. Resistive random access memory (ReRAM) crossbar is a promising candidate for CIM architecture, which is faster and consumes lower power than CMOS-based accelerator [24-26].

However, ReRAM suffers from the resistance variation problem, due to imperfect or immature fabrication process and stochastic filament-based switching [27]. The second reason is its inherent mechanism, which cannot be solved by process optimization. The resistance variation affects the precision of the synaptic weights, and can significantly degrade the accuracy of neural networks [28]. To overcome the impact of resistance variation, some off-device training methods were presented in [29, 30], but it also suffers from significant accuracy loss under large variations. Furthermore, the use of multi-level cell (MLC) makes the problem even worse.

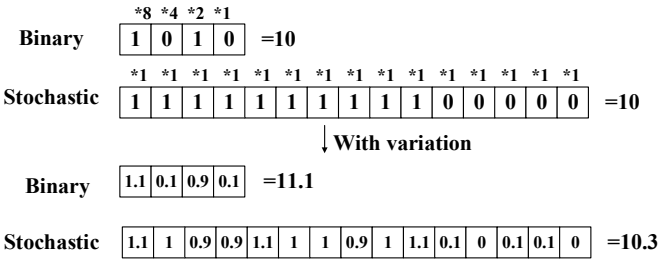

Figure 10. Stochastic coding, or unary coding, is variation-resistant due to the equivalent significance of each bit.

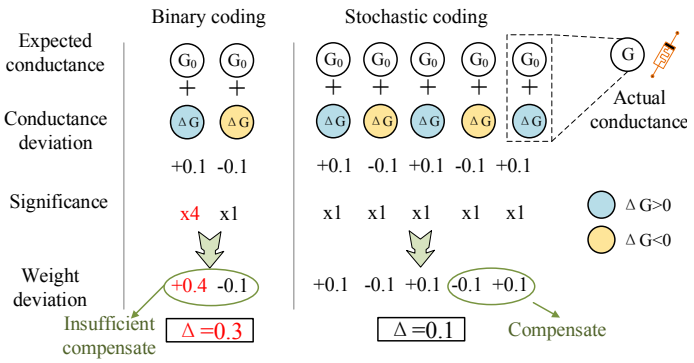

Figure 11. Stochastic coding has a lower deviation than the corresponding binary coding for representing the same value of weight.

It has been mentioned in Section III that the coding format of SC is fault-tolerant. Since each bit in the stochastic coding has the same significance, which is also known as unary coding [31], the stochastic coding has a strong tolerance to bit-flip errors. As shown in Fig. 10, stochastic coding also has a strong tolerance to variations of ReRAM resistance (or conductance). Thus, a new solution to enhance the reliability of the ReRAM-based crossbar circuit based on stochastic coding has been proposed [32].

A. Stochastic Coding with Single-level Cell and Multi-level Cell of ReRAMs

In binary coding, as different bits at different positions have different significance, the most significant bits (MSBs) can amplify the weight variation. In contrast, the stochastic coding has all the bits of the same significance. However, the stochastic coding needs more cells for representing the same range of data as the binary coding. For example, $(2^N - 1)$ bits are needed in the stochastic coding with single-level cells to represent the same precision of the N-bit binary coding. The decimal number 10, expressed in the binary coding as "1010", is represented as "111111111100000" in the stochastic coding.

To further increase the efficiency of the ReRAM crossbar accelerator, the 2-bit MLCs are widely used in neural network architectures [25]. MLCs reduce the bits needed to represent data. When k-bit MLCs are used, only $\frac{n}{k}$ and $\frac{2^n-1}{2^k-1}$ of cells are needed in binary coding and stochastic coding, respectively. For example, the decimal number 10 can be expressed in the binary coding as "22" with 2-bit MLCs, and be expressed in the stochastic coding as "33310". Using MLC causes more benefit for stochastic coding than for binary coding.

Fig. 11 shows why stochastic coding is more variation-resistant than binary coding when using MLCs. Assuming 2-bit MLCs are used and the variation of each cell is uniform, the variations of different cells can compensate in stochastic coding due to the equivalent significance of each cell. In contrast, in binary coding, the compensation is insufficient: the variations of MSB cells have a greater impact on the value.

As the results shown in Ref. [32], even under a relatively large variation level of σ=0.8, using stochastic coding to represent the weight can directly improve the accuracy by 30% compared with that using binary coding, for small neural networks such as MLP or LeNet. However, for some deep neural networks such as Vgg16, the accuracy improvement is not significant by only changing the coding format. Therefore, we also propose a stochastic coding assisted optimal mapping method to further reduce the variation impact, as will be discussed in the next subsection.

B. Stochastic Coding Assisted Variation-aware Optimal Mapping and Architecture

As all the bits in the stochastic coding have the same significance, the order of these bits does not change the represented value. When using MLCs, there is more than one way to encode the data. For example, the decimal number 10 can be represented in multiple forms, such as "33310", "33220", or "22222". Since each ReRAM unit has a different variation, different mapping methods will lead to different final weight variations. If the mapping method is not appropriate, it is difficult to fully compensate for the variations between different cells.

In this case, to make the best use of the coding flexibility brought by stochastic coding and find the optimal mapping way, a variation-aware optimal mapping has been proposed, which can greatly reduce the weight variations, even for deep neural networks with large datasets. More details can be found in Ref. [33].

TABLE II. ACCURACY OF DIFFERENT IMPLEMENTATION AT σ= 1 AMONG FOUR COMBINATIONS OF NEURAL NETWORK AND DATASET. ALL IMPLEMENTATIONS USE FOUR 2-BIT MLCS TO REPRESENT A WEIGHT

Neural Network	ResNet18	Vgg16	ResNet18		Vgg16	
Dataset	CIFAR10	CIFAR10	ImageNet		ImageNet	
Accuracy / %	Top1	Top1	Top1	Top5	Top1	Top5
Ideal	94.30	93.66	69.68	89.07	73.36	91.52
Binary computing	10.83	10.59	0.11	0.51	0.11	0.49
Stochastic computing	94.22	92.77	62.69	84.62	67.57	88.08
Stochastic – Binary	83.39	82.18	62.58	84.11	67.46	87.60
Loss (Ideal - Stochastic)	0.08	0.89	6.99	4.45	5.79	3.43

In addition, the architecture of the ReRAM crossbar accelerator is changed accordingly. The traditional binary ReRAM crossbar accelerator uses ADCs to convert current into digital values, and then uses a shift module and an adder to calculate the final value [25]. The proposed architecture uses stochastic coding to represent the weights, so it does not need the shift module after the ADC. Therefore, the proposed SC-based architecture also reduces the hardware overhead.

C. Results and Discussions

The proposed method is evaluated with two neural networks on two datasets. The accuracy of various methods for four different combinations of neural networks and datasets is listed in Table II. We set the device variation σ as 1, which is a very large level. If the proposed method can guarantee a small accuracy loss under this extreme case, the accuracy loss will also be smaller for σ<1. The "Ideal" row gives the ideal accuracy with floating-point weights and no variation. The "Binary computing" method is the traditional binary computing architecture. The "Stochastic computing" method is the SC-based architecture with the stochastic coding assisted variation-aware optimal mapping. All these methods use four 2-bit MLCs to represent a weight, so the hardware costs of these methods are the same.

The results show that stochastic coding assisted optimal mapping method can improve accuracy significantly. The proposed method has at least 62.58% higher accuracy than the traditional binary computing architecture. It is worth noting that the accuracy loss in stochastic computing mostly originated from the lower representation precision of the stochastic coding itself. In addition, the energy/area-efficiency can be further improved if the MLC has more levels [33].

V. CONCLUSIONS

In this paper, three reliability-enhanced design methods are demonstrated, which are all based on emerging computing paradigms. The results show that some emerging computing paradigms can inherently enhance reliability, which can be used with the unreliable emerging device to build a dependable system. It should be noted that these emerging computing paradigms typically target at applications that do not require absolute computational accuracy, so the accuracy constraint can be relaxed in exchange for power, area, speed, and/or reliability. The results also indicate that the cross-layer design framework is urgently needed in advanced technology nodes and beyond CMOS devices.

ACKNOWLEDGMENT

R. Wang would like to thank former students, Zhe Zhang and Shaofeng Guo, for the input.

REFERENCES

[1] R. Huang et al., "Variability-and reliability-aware design for 16/14nm and beyond technology," in 2017 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2017, pp. 12.4.1-12.4.4.

[2] Z. Ji, H. Chen et al., "Design for reliability with the advanced integrated circuit (IC) technology: challenges and opportunities," Sci. China Inf. Sci., vol. 62, no. 12, p. 226401, Dec. 2019.

[3] R. Wang et al., "Too Noisy at the Bottom? —Random Telegraph Noise (RTN) in Advanced Logic Devices and Circuits," 2018 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2018, pp. 17.2.1-17.2.4, doi: 10.1109/IEDM.2018.8614594.

[4] R. Wang, Z. Yu, J. Zhang, Z. Sun, Z. Zhang and R. Huang, "Understanding Hot Carrier Degradation and Variation in FinFET Technology," 2020 IEEE 15th International Conference on Solid-State & Integrated Circuit Technology (ICSICT), Kunming, 2020, pp. 1-4, doi: 10.1109/ICSICT49897.2020.9278158.

[5] Y. Cao et al., "Cross-Layer Modeling and Simulation of Circuit Reliability," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 1, pp. 8-23, Jan. 2014.

[6] H. Amrouch, B. Khaleghi, A. Gerstlauer and J. Henkel, "Reliability-aware design to suppress aging," 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2016, pp. 1-6, doi: 10.1145/2897937.2898082.

[7] J. P. Kulkarni et al., "A 409 GOPS/W Adaptive and Resilient Domino Register File in 22 nm Tri-Gate CMOS Featuring In-Situ Timing Margin and Error Detection for Tolerance to Within-Die Variation, Voltage Droop, Temperature and Aging," in IEEE Journal of Solid-State Circuits, vol. 51, no. 1, pp. 117-129, Jan. 2016.

[8] S. Arasu et al., "Controlling Aging in Timing-Critical Paths," in IEEE Design & Test, vol. 33, no. 4, pp. 82-91, Aug. 2016.

[9] M. Ebrahimi et al., "Aging-aware logic synthesis," in 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, 2013, pp. 61–68.

[10] J. Han et al., "Approximate computing: An emerging paradigm for energy-efficient design," 2013 18th IEEE European Test Symposium (ETS), Avignon, 2013, pp. 1-6.

[11] M. Shafique et al., "Invited - Cross-layer approximate computing: from logic to architectures," in Proceedings of the 53rd Annual Design Automation Conference on - DAC '16, Austin, Texas, 2016, pp. 1–6.

[12] H. Amrouch et al., "Towards Aging-Induced Approximations," in Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17, Austin, TX, USA, 2017, pp. 1–6.

[13] V. Gupta et al., "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.

[14] Z. Zhang, R. Wang, et al. "Reliability-Enhanced Circuit Design Flow Based on Approximate Logic Synthesis," Proceedings of the 2020 on Great Lakes Symposium on VLSI, 2020, pp. 71-76.

[15] S. Guo et al., "Towards reliability-aware circuit design in nanoscale FinFET technology: — New-generation aging model and circuit reliability simulator," in 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Irvine, CA, 2017, pp. 780–785.

[16] Z. Yu, J. Zhang, R. Wang, S. Guo, C. Liu and R. Huang, "New insights into the hot carrier degradation (HCD) in FinFET: New observations, unified compact model, and impacts on circuit reliability," 2017 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2017, pp. 7.2.1-7.2.4, doi: 10.1109/IEDM.2017.8268344.

[17] S. Guo et al., "Investigation on NBTI-induced dynamic variability in nanoscale CMOS devices: Modeling, experimental evidence, and impact on circuits," Microelectronics Reliability 81 (2018): 101-111.

[18] M. Martins et al., "Open cell library in 15nm FreePDK technology," Proceedings of the 2015 Symposium on International Symposium on Physical Design, 2015, pp. 171-178.

[19] J.P. Hayes, "Introduction to stochastic computing and its challenges," 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, 2015, pp. 1-3.

[20] A. Alaghi, W. Qian and J. P. Hayes, "The Promise and Challenge of Stochastic Computing," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 8, pp. 1515-1531, Aug. 2018.

[21] Y. Zhang et al., "Design guidelines of stochastic computing based on FinFET: A technology-circuit perspective," 2017 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2017, pp. 6.6.1-6.6.4.

[22] A. Alaghi, Cheng Li and J. P. Hayes, "Stochastic circuits for real-time image-processing applications," 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2013, pp. 1-6.

[23] Z. Zhang, R. Wang, Z. Zhang, Y. Zhang, S. Guo and R. Huang, "Circuit Reliability Comparison Between Stochastic Computing and Binary Computing," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 12, pp. 3342-3346, Dec. 2020.

[24] P. Chi et al., "PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), pp. 27-39, June 2016.

[25] A. Shafiee et al., "ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars," ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), pp. 14-26, June 2016.

[26] L. Song, X. Qian, H. Li and Y. Chen, "Pipelayer: a pipelined ReRAMbased accelerator for deep learning," IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 541-552, February 2017.

[27] H. Li et al., "Variation-aware, reliability-emphasized design and optimization of RRAM using SPICE model," Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1425-1430, March 2015.

[28] Y. Long, T. Na, S. Mukhopadhyay, "ReRAM-based processing-inmemory architecture for recurrent neural network acceleration," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 12, pp. 2781-2794, December 2018.

[29] L. Chen et al., "Accelerator-friendly neural-network training learning variations and defects in RRAM crossbar," Design, Automation & Test in Europe Conference & Exhibition (DATE) , pp. 19-24, March 2017

[30] Y. Long, X. She and S. Mukhopadhyay, "Design of reliable DNN accelerator with un-reliable ReRAM," Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1769-1774, March 2019.

[31] W. Qian, R. Wang, Y. Wang, M. Riedel and R. Huang, "A Survey of Computation-Driven Data Encoding," 2019 IEEE International Workshop on Signal Processing Systems (SiPS), Nanjing, China, 2019, pp. 7-12, doi: 10.1109/SiPS47522.2019.9020519.

[32] C. Ma, Y. Sun, W. Qian, Z. Meng, R. Yang and L. Jiang, "Go Unary: A Novel Synapse Coding and Mapping Scheme for Reliable ReRAM-based Neuromorphic Computing," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2020, pp. 1432-1437.

[33] Y. Sun et al., "Unary Coding and Variation-Aware Optimal Mapping Scheme for Reliable ReRAM-based Neuromorphic Computing," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, to be published.