A Survey of Majority Logic Designs in Emerging Nanotechnologies for Computing

Tingting Zhang, *Graduate Student Member, IEEE,* Honglan Jiang, *Member, IEEE,*
Weiqiang Liu, *Senior Member, IEEE,* Fabrizio Lombardi, *Life Fellow, IEEE,* Leibo Liu, *Senior Member, IEEE,*
Seok-Bum Ko, *Senior Member, IEEE,* and Jie Han, *Senior Member, IEEE*

***Abstract*—As Moore's law is coming to an end, research on technologies alternative to the complementary metal-oxide semiconductor (CMOS) has been extensively pursued over the last few decades. Many emerging nanotechnologies assemble circuits based on majority logic. It is generally known that majority logic is more expressive and hardware efficient than Boolean logic, however majority logic presents unique challenges at many levels such as arithmetic design and synthesis. With the rediscovery of majority logic as a computational primitive in the post-CMOS era, this paper briefly reviews recent studies from various perspectives to highlight accomplishments and open problems across many domains of majority logic design.**

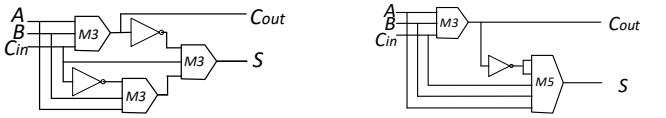
***Index Terms*—Majority logic, logic synthesis, approximate computing, nanotechnology, reliability.**

I. Introduction

MOORE'S law predicts that the number of transistors on a microchip will double approximately every eighteen months, as observed over the past 50 years for the development of integrated circuits. However, as the feature size of complementary metal-oxide semiconductor (CMOS) transistors continues to scale down, fundamental physical limits lead to many severe challenges, such as increased leakage currents, doping fluctuations, and expensive lithography [1]. In the post-CMOS era, new nanometer-scale technologies are being developed to improve upon or replace traditional CMOS technology to achieve a faster switching speed, higher integration density, and lower power dissipation.

A majority gate (or voter, MV) performs a voter function with an odd number of inputs, i.e., it outputs true (or false) if more than half of the inputs are true (or false). A 3-input MV implements the logic function, $F = M(A, B, C) = AB + BC + AC$, where A, B, and C are the inputs. Majority logic has been widely used to design fault-tolerant architectures, such as the N-modular redundancy for reliable system design. Unlike CMOS technology that relies on Boolean logic, a number of novel emerging technologies function on the principles of the majority or minority logic [2], such as quantum-dot cellular automata (QCA) [3], nanomagnetic logic (NML) [4], and spin-wave technology [5]. Majority logic is straightforward and thus efficient for implementations on a specific nanotechnology. It also benefits from its capability to realize functions using circuits with fewer gates, as shown in Fig. 1. In this paper, we review the use of majority logic for synthesis and arithmetic computation in different nanoscale technologies.

The remainder of this paper is organized as follows. Section II introduces majority logic-based nanometer-scale technologies. Majority logic synthesis methods are reviewed in Section III. Section IV discusses majority logic-based approximate arithmetic designs. Finally, Section V concludes this paper with a discussion of challenges and prospects.



MV: 3 $M3$s; INV: 2; D: 2. (a) Based on 3-input MV and NOT [6].

MV: 1 $M3$ and 1 $M5$; INV: 1; D: 2. (b) Based on 3/5-input MV and NOT [7].

Fig. 1: The full adders at the gate level using majority gates and inverters. It performs the addition of $A + B + C_{in}$ and produces a carry output, C_{out}, and a sum, S. $M3$: three input MV; $M5$: five input MV. The circuit characteristics are evaluated by the number of utilized MVs, the number of utilized inverters (INVs), and the critical path delay (D) from theoretical level.

II. Majority Logic-based Emerging Technologies

This section introduces several beyond-CMOS technologies using majority logic as the design primitive.

A. Quantum-dot Cellular Automata (QCA)

Instead of transistors, the basic element in QCA is a cell composed of four single tunnel structures with four quantum dots on the corners and two electrons [3], [11], [12]. The electrons are positioned on two different diagonals with a +1 or -1 polarization state for the logic value '1' or '0', respectively.

T. Zhang and J. Han are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: ttzhang@ualberta.ca, jhan8@ualberta.ca [Corresponding author]).

H. Jiang is with the Department of Micro-Nano Electronic, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: honglan@sjtu.edu.cn).

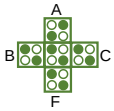
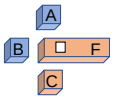
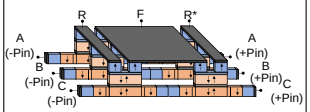
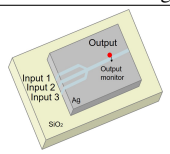
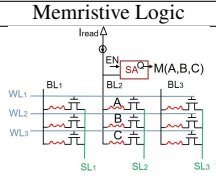
L. Liu is with the Institute of Microelectronics, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: liulb@tsinghua.edu.cn).

W. Liu is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: liuweiqiang@nuaa.edu.cn).

F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston MA 02115, USA (e-mail: lombardi@ece.neu.edu).

S. Ko is with the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon SK S7N 5C9, Canada (e-mail: seokbum.ko@usask.ca).

TABLE I: Majority Logic-based Nanotechnologies

Technology	Quantum-dot Cellular Automata	Nanomagnet Logic	Spin-based Technology	Plasmonic Technology	Memristive Logic
MV Circuits	 An MV in QCA [3]	 An MV in NML [4]	 An STT-MTJ-based MV [8]	 A plasmonic MV [9]	 A ReRAM-based R-V MV [10]
Basic Elements	QCA cells with four quantum dots and two electrons	Single domain nanomagnets	Magnetic tunnel junctions	Surface-plasmon-polariton waveguides	Memristors
Logic Values	Cells with electrons positioned on two different diagonals	Magnets with two stable states	The resistance states of MTJs	The phase of the SPP	The resistance states of memristors
Information Propagation	Electrostatic interaction among neighbor cells	Magnetostatic interaction among neighbor magnets	Electric currents to transport spin angular momentum	Phase-dependent interference of SPP waves	The charge passing through memristors
Features	High operating frequency Low power High circuit density	Low operating frequency Low power Non-volatility High circuit density	High endurance Low power Non-volatility High circuit density	High operating frequency Low power High circuit density	High operating frequency Low power Non-volatility High circuit density

Due to Coulomb repulsion, the polarization spreads from one cell to its neighbor cells without moving electrons between cells. A 3-input MV consists of four QCA cells that operate by electromagnetic interactions, as shown in Table I.

B. Nanomagnet Logic (NML)

NML uses the two stable magnetization states of nanoscale magnets as logic values '0' and '1' [13], [14]. The magnets are placed close to each other, information is propagated using the magnetic interaction among neighboring magnets. An MV in NML is made of three input magnets surrounding a central element [4], as shown in Table I.

C. Spin-based Technology

Spin-based technology for MV implementation includes spin-wave technology [15], all spin logic [16], and spin-transfer torque technology [8]. Consider the spin-transfer torque technology as an example; a spin torque MV can be realized by magnetic tunnel junctions (MTJs). As one of the emerging spintronic technologies, an MTJ is made of two ferromagnetic layers separated by a thin insulating tunnel barrier: one free layer with a changeable magnetization and the other with a fixed magnetization. The resultant low or high resistance of the junction encodes a bit of information. An all-magnetic MV based on a spin-transfer torque MTJ (STT-MTJ), referred to as STTMAJ, can realize Boolean and non-Boolean functions without using transistors, as shown in Table I.

D. Plasmonic Technology

The plasmonic technology utilizes the propagation of interfacing surface-plasmon-polariton (SPP) waves between a dielectric and a metal. It localizes the electromagnetic energy in dimensions, thus overcoming the diffraction limitation of photonics. The wave nature of the computation has motivated the studies on implementing MVs with plasmons. In [9], the plasmonic-based logic utilizes the phase of the SPP wave as logic variables, which follow the majority roles. A three-input plasmonic MV layout is presented in Table I.

E. Memristive Logic

Memristive logic uses the resistance states of memristors as basic building blocks for logic operations. A memristor is a non-volatile memory element that can change its resistance state depending on the amount of charge that has passed through it with very low power consumption. Memristive logic has been recently exploited for in-memory computing by making logic gates executable in an array configuration [17]. The functionally complete Majority+NOT logic family is preferable for implementation [18]. An MV can be implemented in a Resistive Random Access Memory (ReRAM) array by utilizing V–R logic or R–V logic. A ReRAM-based R–V MV is presented in Table I.

Table I summarizes the basics for those aforementioned majority logic-based nanotechnologies. There are also some alternative technologies relying on minority logic, such as single electron tunneling [19], [20] and tunneling phase logic [21]. As complementary to majority logic, minority logic shares similar rules for circuit implementation.

III. Majority Logic Synthesis

In synthesis, a logic function is extracted from a high-level description to an optimized gate-level representation [22]. Various synthesis methods are based on logic primitives for circuit design using different technology or implementation platforms. For example, the use of NOR or NAND gates are common for CMOS technology [23]. Field programmable gate arrays (FPGAs) are built on reconfigurable lookup tables (LUTs) [24], [25]. For majority logic-based nanotechnologies, it is necessary to describe the logic functions by using primitives of MVs and inverters.

By synthesizing circuits using logic primitives of majority and NOT gates (such as the Majority-Invert-Graphs, or MIGs), and AND and NOT gates (such as the AND-OR-Invert-Graphs, or AOIGs), it has been shown that the majority logic (together with NOT) can achieve up to a 33% reduction in logic depth [26]. However, a direct transformation from Boolean logic to majority logic may likely not lead to an optimal gate-level implementation. Therefore, majority logic synthesis plays an important role to obtain an optimized netlist for the circuit to

be realized. Moreover, majority logic synthesis is applicable to CMOS-based applications such as FPGA-based technology mapping over the LUT structure [27].

This section briefly reviews majority logic synthesis methods for obtaining optimized circuit implementations.

A. Function-oriented Majority Logic Synthesis

1) Synthesis in Majority-of-3 (maj-3) Forms

The research in majority logic synthesis has been focused on three-input majority logic functions for QCA. Based on the geometric interpretation of Boolean functions, thirteen standard functions have been synthesized manually in [6], [28]. An automated method has been proposed in [29], but it cannot optimize the number of MVs and majority levels. Further improvement based on this approach has been studied in [30] to obtain the optimized majority logic function by introducing a metric to evaluate a majority network. However, these approaches cannot handle functions with more than three inputs. Various algorithms have been developed in the preprocessing and decomposition of circuits, and in converting each node into Majority-of-3 (maj-3) expressions [31]–[33].

Alternatively, a logic representation structure referred to as Majority-Invert-Graphs (MIG) has been recently proposed for optimizing majority logic expressions based on Boolean logic representation [26]. An MIG describes the maj-3 network through a ternary tree format by using majority nodes and regular/complemented edges. It is derived from an AOIG, by fixing one input as '0' or '1' to realize an AND or OR operation. A dedicated Boolean algebra is introduced to derive transformation rules for the optimization of the MIG representations. To further optimize both the number of MVs and majority levels, there are some studies on MIG-based optimization methods by utilizing the error masking property of majority functions [34], functional hashing [35], node-merging using Boolean satisfiability (SAT) [27], [36] or Boolean resubstitution [37], [38]. Fig. 2 (a)-(b) presents the optimized logic networks based on AOIG and MIG for a 4-bit adder, respectively. Moreover, there are some studies on logic synthesis based on an extended logic representation of MIG, referred to as XOR-Majority graph (XMG) [39], given exclusive-OR (XOR), MV, and inverters as primitives. Fig. 2 (c) shows the optimized logic network based on XMG for a 4-bit adder.

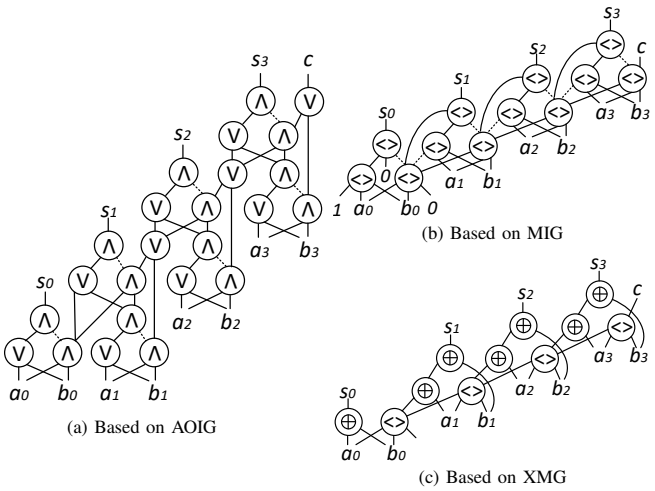


Fig. 2: Logic networks for a 4-bit adder [22]: $\bigvee$: OR; $\bigwedge$: AND; <>: Majority; $\bigoplus$: XOR; Dotted line: NOT. It performs the addition of $A(= a_3a_2a_1a_0) + B(= b_3b_2b_1b_0)$ and produces a carry output, c, and a sum, $S(= s_3s_2s_1s_0)$.

2) Synthesis in Majority-of-n (maj-n) Forms

When using MVs with a larger fan-in in majority logic synthesis can reduce the majority levels and the number of required MVs. Synthesis approaches for maj-3 networks can be generalized to exploit Majority-of-n (maj-n) networks [40]. However, it does not result in optimized circuits for maj-n networks. Therefore, there are some efforts on logic synthesis for maj-n networks. A heuristic-based majority logic synthesis method has been presented by using a library based on 3-input and 5-input MVs in [33]. [41] has proposed a synthesis flow for a maj-n network by converting each look-up table to an MV without decomposition. This approach has the drawback of using massive inversions without optimization. Chu et al. have presented a synthesis method to represent Boolean functions in maj-5 forms to optimize the number of MVs, followed by majority levels [42]. An improved approach has been studied in [43], which synthesizes networks composed of maj-3 and maj-5 by encoding majority logic functions into linear optimization models.

The efficiency of physical circuit implementations of MVs with different input sizes relies on the underlying technology; therefore, efficient decomposition of maj-n to maj-3 has been extensively explored. A sorting network with $O(nlog_n)$ [44] and the median selection algorithm with linear complexity [45] provide effective approaches to generate maj-3 networks with the asymptotic bound and the quasi-linear bound, respectively. To obtain a minimum MIG for small values of n, Testa et al. map monotone maj-n functions to maj-3 functions using binary decision diagrams (BDDs) and Shannon's expansion with the quadratic bound [46]. Chattopadhyay et al. further develop a constructive and linear decomposition approach [47].

B. Technology-aware Majority Logic Synthesis

The function-oriented majority logic synthesis is developed based on conventional metrics, i.e., the number of MVs and the majority levels. These approaches are technology-independent, so without considering physical implementations on a specific technology. Direct mapping into a circuit using basic building blocks may lead to inefficiency. Several optimization algorithms for multilevel majority logic synthesis have been developed based on the MIG by considering technology-aware metrics [22].

Since QCA has a relatively expensive implementation of inverters, logic synthesis algorithms to minimize or eliminate inversion have been developed based on the MIG [48], [49]. Testa et al. have rewritten the MIGs to an inversion-free network by moving all inverters to the primary inputs and also restricting the fan-out of each MIG node to a maximum value [49]. Moreover, an SAT-based method developed on MIGs has been proposed by considering constraints in fan-out

and implementation depth [50]. Torres et al. have presented an energy-aware model for logic synthesis for QCA based on an estimate of the energy dissipation [51]. The majority logic synthesis approach in [52] minimizes the number of wire-crossings in single-layer QCA circuits by analyzing the impact of input ordering on wire-crossing. [53] has proposed a cross-aware logic synthesis technique to exploit the inherent bit-level parallelism of ReRAM crossbar arrays.

IV. Majority Logic-based Approximate Arithmetic Circuit Designs

Approximate computing improves the efficiency of circuits and systems with a controlled loss of accuracy [54]. Although approximate designs based on Boolean logic can be utilized for majority logic-based nanotechnologies, a naive mapping does not fully leverage the properties of majority logic [55]. Hence, dedicated designs have been developed for majority logic-based arithmetic functions and, more recently, approximate circuits.

This section discusses majority logic-based designs for approximate arithmetic circuits, including adders, subtractors, compressors, and multipliers. Also, error-tolerant applications based on the reviewed designs are introduced.

A. Majority Logic-based Approximate Adders and Subtractors

A full adder (FA) performs the addition $A + B + C_{in}$ and produces a carry output, $C_{out} = M(A, B, C_{in})$, and a sum, $S = M(\overline{C}_{out}, M(A, B, \overline{C}_{in}), C_{in})$ [56], [57]. A full subtractor (FS) performs the subtraction denoted by $A - B - B_{in}$ and then produces one borrow output, $B_{out} = M(\overline{A}, B, B_{in})$, and one difference, $D = M(B_{out}, M(A, \overline{B}, B_{in}), \overline{B}_{in})$ [58].

1) Approximate Full Adders and Subtractors

Approximate FAs (AFAs) and approximate FSs (AFSs) have been developed based on the simplification of the truth table and logic functions. Various AFA designs [59], [60] are variants of the design in [61], denoted by AFA1, which accurately generates C_{out} and approximately computes S. Therefore, errors do not propagate to the more significant bits in a multi-bit adder. The AFA design in [62], denoted by AFA2, approximately generates both C_{out} and S; however, the errors introduced to C_{out} and S compensate each other. Labrado et al. presented an AFS design [61], which accurately computes B_{out} and generates an erroneous D in two input cases. As shown in Table II, compared with their exact counterparts, AFA and AFS designs save two MVs, one inverter and one unit of delay. AFA1 introduces a one-bit error when the exact result is 0, which is not included when computing MRED. Therefore, AFA1 and AFA2 share similar results in hardware and error metrics, except for the MRED.

2) Approximate Multi-bit Adders and Subtractors

Approximate 2-bit adders (A2As) have been proposed by cascading two AFAs [62], logic simplification based on the truth table [64], and shortening the carry chain [65]. Table II gives the results in hardware and error metrics of these A2A designs: (1) Zhang et al. have assembled A2As based on AFA1 and AFA2 [62]. The use of two different types of AFAs leads to more accurate A2A designs, denoted by A2A1 (whose lower significant bit is computed by AFA1) and A2A2 (whose lower significant bit is computed by AFA2). (2) The A2A design based on the truth table in [64] and denoted by A2A3, requires two more MVs than AFA-based A2As. However, it benefits from higher accuracy. (3) To shorten the critical path of multi-bit adders, two A2A designs with no carry out bit [65] and denoted by A2A4 and A2A5, have been proposed for multi-bit adders.

TABLE II: Hardware and Error Metrics of ML-based Approximate Adders and Subtractors

Designs		MV	INV	D	MRED	NMED	RMSE	ER
One-bit adders	**Exact [56]**	3	2	2	-	-	-	-
	AFA1 [61]	1	1	1	0.0416	0.083	0.25	0.25
	AFA2 [62]	1	1	1	0.1875	0.083	0.25	0.25
One-bit subtractors	**Exact [58]**	3	2	2	-	-	-	-
	AFS [61]	1	1	1	0.0416	0.083	0.25	0.25
Two-bit adders	**Exact [63]**	6	4	3	-	-	-	-
	A2A1 [62]	2	1	2	0.1977	0.0893	1	0.4375
	A2A2 [62]	2	2	1	0.2496	0.0893	1	0.4375
	A2A3 [64]	4	2	2	0.1912	0.0804	0.8292	0.5
	A2A4 [65]	6	1	3	0.2786	0.1429	1.3693	0.6563
	A2A5 [65]	6	2	4	0.1586	0.0714	0.7071	0.5
Four-bit subtractors	**Exact [58]**	12	12	5	-	-	-	-
	A4S1 [66]	7	6	5	0.6160	0.1250	2.6458	0.8125
	A4S2 [66]	5	4	3	0.9378	0.1821	3.5753	0.9023

The error metrics including the normalized mean error distance (NMED), the mean relative error distance (MRED), the root-mean-square error (RMSE), and the error rate (ER) [67] are utilized to assess the error characteristics of approximate designs.

By discarding some of the primary input bits and the carry in bit, Chu et al. have used two approximate 4-bit adders (A4As) to cut off the carry chain and reduce the hardware when building large adders [66]. Approximate 4-bit subtractors (A4Ss) have been designed for constructing unsigned restoring array dividers [68]: A4S1 correctly generates the borrow output and A4S2 decreases the accuracy in the borrow propagation to alleviate the issue of a long delay. The comparison results in Table II indicate that A4S1 with $NMED = 0.1250$ and A4S2 $NMED = 0.1821$ save five and seven MVs, respectively. They are respectively preferred for the computation of the most and less significant bits. Approximate multi-bit designs can be assembled by cascading the aforementioned designs [64], i.e., implementing the least significant bits (LSBs) by approximate circuits, whereas computing the most significant bits (MSBs) by exact counterparts.

B. Majority Logic-based Approximate Multipliers

For an unsigned multiplier, let $A = a_{n-1}a_{n-2}...a_1a_0$ be the multiplicand and $B = b_{n-1}b_{n-2}...b_1b_0$ be the multiplier. Let pp_{ij} denote the partial product (PP) bit in the ith row and the jth column, where $0 \leq i, j \leq n-1$. The PPs are first generated as $pp_{ij} = M(a_j, b_i, 0)$ [63]. For the signed multiplication, the Booth algorithm is widely utilized to reduce the number of PPs by recoding the multiplier [69]. A radix-2^k Booth multiplier groups the multiplier bits into sets of $k+1$ adjacent bits. Then, the multiplicand is weighted by $\pm 2^{k-1}$, $\pm 2^{k-2}$, ..., or 0 to generate PPs. The higher radix the Booth algorithm is, the fewer PPs are generated, thereby reducing the complexity of PP accumulation.

For both unsigned and signed multipliers, the PPs are then accumulated by using different circuit structures, such as a carry save adder array [70], the Wallace tree [71], or the Dadda tree [72]. In general, FAs, half adders, and 4:2 compressors are used

TABLE III: Hardware and Error Metrics of ML-based Approximate 4:2 Compressors

Approximate Designs		MV	INV	D	MRED	NMED	RMSE	ER (%)
Based on AFAs	AC1 [64]	2	2	2	0.1625	0.075	0.6124	37.5
	AC2 [64]	2	2	2	0.1953	0.075	0.6124	37.5
	AC3 [64]	2	2	2	0.1495	0.075	0.6124	37.5
Based on the Truth Table	AC4 [76]	2	2	2	0.2344	0.1	0.7906	43.75
	AC5 [77]	1	0	1	0.2604	0.1	0.7071	50
Based on Output Reduction	AC6 [64]	4	0	2	0.2339	0.0875	0.7071	40.63
	AC7 [78]	1	0	1	0.2031	0.125	0.7071	50

to compress the PP array into two rows. Lastly, the remaining two rows are added to generate the final result.

1) Approximate Partial Product Generation

For unsigned array multipliers, the authors in [64] have proposed a majority logic-based approximate 2×2 multiplier with a 3-bit output and a compensation bit, where each bit is generated by one MVs. The compensation bit is useful to compensate for the accuracy loss in larger multipliers. The more compensation bit is disregarded, the less accurate the result is.

For radix-4 Booth multipliers, approximate PP generators with single-sided or double-sided errors have been developed based on different implementation of encoding schemes [73], [74]. Compared with the radix-4 Booth algorithm, radix-8 and radix-16 Booth algorithms suffer from a long carry propagation to generate odd multiples of the multiplicand. Therefore, approximate recoding adders (ARAs) without carry propagation have been considered for computing odd multiples [75]. For calculating $3A$, $5A$ and $7A$, 2-bit, 3-bit, and 4-bit ARAs have been designed to compute the sum of A and $2A$, the sum of A and $4A$, and the sum of $-A$ and $8A$, respectively.

2) Approximate Partial Product Accumulation

For the PP compression, AFAs and approximate 4:2 compressors (ACs) have been used in Wallace and Dadda trees [64]. A 4:2 compressor takes five inputs: four primary inputs and one carry input from the lower bit accumulation. It produces three outputs: two primary outputs ($Carry$ and Sum) and one carry output (C_{out}). It is commonly implemented by cascading two FAs. A comprehensive comparison of ACs is provided in Table III. The existing ACs are designed using three criteria: using AFAs, changing truth tables and using output reduction: (1) AFAs can be used to replace the exact counterparts [64]. AC1 and AC2 are designed by cascading two AFA1s and two AFA2s, respectively; AC3 uses AFA1 for the lower part computation and AFA2 for the higher part computation. AC3 produces the smallest MRED. For higher accuracy, only the less significant FA is substituted by an AFA [79]. (2) The entries in a truth table can be changed by introducing errors to reduce circuit complexity [76], [77]. Compared with AC4 [76], although with an increase of 11.1% in MRED and 14.2% in ER, AC5 [77] improves the accuracy in RMSE by 11.3% with a reduction of one MV, two inverters, and one delay unit. (3) To further decrease the hardware complexity, two ACs are designed by reducing the number of outputs [64], [78]. These designs use two output bits (rather than three) for the result, so C_{out} is omitted. Different from AC6 [64], AC7 [78] uses only four input operands and one MV.

Therefore, among the AC designs with five input operands and three outputs, AC3 [64] and AC5 [77] are superior in terms of accuracy and hardware. The AC6 [64] benefits from its characteristics of the fewer number of outputs, which can decrease the complexity of PP reduction and compression design. The AC7 [78] with only two outputs is specially designed for four input operands.

Instead of using 4-2 compressors, there are some designs of approximate compressors with an arbitrary number of inputs for PP reduction [66], [80], [81]. More interestingly, an algorithm of constructing a heuristic MIG for exact and approximate compressors using maj-n logic networks has been developed in [81]. Moreover, the low power truncation technique [78] and complementary strategies for the errors based on a probabilistic analysis [74] can be utilized to achieve a superior trade-off between accuracy and hardware efficiency.

C. Applications

Majority logic-based approximate arithmetic units have been applied in error tolerant applications.

Case studies of image processing include addition, filtering, multiplication, sharpening, smoothing and compression. The output quality is commonly evaluated by the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM). The greater the PSNR and SSIM values are, the less distortion occurs. The use of approximate 8-bit adders for addition in image filtering approximately results in a PSNR value of 44 dB and an SSIM value of 0.95, with a reduction of 40% in area-and-delay product (ADP) [65]. The approximate 8×8 multiplier in [78] provides a PSNR value of around 40 dB for image multiplication, with a saving of approximately 70% in ADP.

NN applications include classification using a multilayer perceptron and joint face detection and alignment using a multi-task convolution NN (MTCNN). Two often employed measurements are the true positive rate (TPR) and the false positive rate (FPR). For the joint face detection and alignment using the Face Detection Data Set and Benchmark dataset, approximate 8×8 multipliers in inference reduce the number of multiply-and-accumulate units by 5.1%, with a saving of approximately 18.9% in ADP and an accuracy loss of 7.4% in TPR, compared to the use of exact multipliers [74].

V. Conclusions, Challenges and Prospects

Emerging nano-meter technologies have shown great potentials as alternatives to conventional CMOS technology for their high density and low energy.

For those majority logic-based technologies, including QCA, NML, memristive logic, spin-based technology, and plasmonic technology, logic synthesis by leveraging the characteristics of majority logic plays a crucial role in circuit design. However, the costs and performances of circuits constructed by MVs with different inputs and inverters highly depend on a physical implementation. For example, the number of utilized gates and the wire crossings can be reduced when realizing a function in maj-n networks, but this may be at the cost of an increased gate area. Therefore, to obtain an optimized circuit implementation, an automatic logic synthesis approach with a comprehensive consideration from functional, physical, and geometric levels is worthy of further studies.

Approximate circuits provide opportunities for majority logic-based nanotechnologies to improve hardware efficiency. Designs of majority logic-based approximate arithmetic circuits are briefly reviewed, including adders, subtractors, compressors, and multipliers. Compared to addition, subtraction, and multiplication, division requires more complex computing structures. majority logic-based approximate 4-bit subtractors have been developed for unsigned restoring array dividers in [68]. However, more efficient designs could be obtained by exploiting the unique structures of various types of dividers. Moreover, there is a lack of research on common computational blocks such as those for fast Fourier transform and multiply–accumulate units.

Finally, most current approximate arithmetic designs are developed independently from the physical characteristics of nanotechnologies. Compared with conventional Boolean logic, the utilization of majority logic as a primitive can improve the reliability of computational circuits [82]–[84]. However, defects may appear in the manufacturing process and modify the correct logic function [85], [86]. The impact of different types of defects on the effectiveness of approximate arithmetic circuits awaits further investigation. It will also be interesting to see if approximate computing can provide effective mitigation strategies for the reliability issues that have been long-lasting and plaguing nanotechnologies.

Acknowledgment

This work was supported by NSERC (RES0048688, RES0051374 and RES0054326), and NSFC (62022041 and 62104127). The research of Fabrizio Lombardi was supported by NSF under CCF-1953961 and 1812467 grants. T. Zhang was supported by a Ph.D. scholarship from the China Scholarship Council (CSC).

References

[1] R. H. Dennard, J. Cai, and A. Kumar, "A perspective on today's scaling challenges and possible future directions," in *Handbook of Thin Film Deposition*. Elsevier, 2018, pp. 3–18.

[2] B. Parhami, D. Abedi, and G. Jaberipur, "Majority-logic, its applications, and atomic-scale embodiments," *Comput. Electr. Eng.*, vol. 83, p. 106562, 2020.

[3] A. Imre, G. Csaba, L. Ji, A. Orlov, G. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.

[4] M. Vacca, M. Graziano, and M. Zamboni, "Majority voter full characterization for nanomagnet logic circuits," *IEEE TNANO*, vol. 11, no. 5, pp. 940–947, 2012.

[5] A. Mahmoud, F. Ciubotaru, F. Vanderveken, A. V. Chumak, S. Hamdioui, C. Adelmann, and S. Cotofana, "Introduction to spin wave computing," *J. Appl. Phys.*, vol. 128, no. 16, p. 161101, 2020.

[6] R. Zhang, K. Walus, W. Wang, and G. A. Jullien, "A method of majority logic reduction for quantum cellular automata," *IEEE TNANO*, vol. 3, no. 4, pp. 443–450, 2004.

[7] M. R. Azghadi, O. Kavehei, and K. Navi, "A novel design for quantum-dot cellular automata cells and full adders," *J. Appl. Sci.*, vol. 7, no. 22, pp. 3460–3468, 2007.

[8] V. Jamshidi, "A VLSI majority-logic device based on spin transfer torque mechanism for brain-inspired computing architecture," *IEEE TVLSI*, vol. 28, no. 8, pp. 1858–1866, 2020.

[9] S. Dutta, O. Zografos, S. Gurunarayanan, I. Radu, B. Soree, F. Catthoor, and A. Naeemi, "Proposal for nanoscale cascaded plasmonic majority gates for non-Boolean computation," *Sci. Rep.*, vol. 7, no. 1, p. 17866, 2017.

[10] J. Reuben, "Binary addition in resistance switching memory array by sensing majority," *Micromachines*, vol. 11, no. 5, p. 496, 2020.

[11] V. Pudi and K. Sridharan, "Low complexity design of ripple carry and brent-kung adders in QCA," *IEEE TNANO*, vol. 11, no. 1, pp. 105–119, 2011.

[12] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, "QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata," *IEEE TNANO*, vol. 3, no. 1, pp. 26–31, 2004.

[13] G. Csaba, A. Imre, G. H. Bernstein, W. Porod, and V. Metlushko, "Nanocomputing by field-coupled nanomagnets," *IEEE TNANO*, vol. 1, no. 4, pp. 209–213, 2002.

[14] M. T. Alam, M. J. Siddiq, G. H. Bernstein, M. Niemier, W. Porod, and X. S. Hu, "On-chip clocking for nanomagnet logic devices," *IEEE TNANO*, vol. 9, no. 3, pp. 348–351, 2010.

[15] O. Zografos, B. Sorée, A. Vaysset, S. Cosemans, L. Amaru, P.-E. Gaillardon, G. De Micheli, R. Lauwereins, S. Sayan, P. Raghavan *et al.*, "Design and benchmarking of hybrid CMOS-spin wave device circuits compared to 10nm CMOS," in *IEEE-NANO*. IEEE, 2015, pp. 686–689.

[16] M. G. Mankalale and S. S. Sapatnekar, "Optimized standard cells for all-spin logic," *ACM JETC*, vol. 13, no. 2, pp. 1–22, 2016.

[17] J. Reuben, D. Fey, and C. Wenger, "A modeling methodology for resistive RAM based on stanford-pku model with extended multilevel capability," *IEEE TNANO*, vol. 18, pp. 647–656, 2019.

[18] J. Reuben, "Rediscovering majority logic in the post-CMOS era: A perspective from in-memory computing," *J. Low Power Electron. Appl.*, vol. 10, no. 3, p. 28, 2020.

[19] T. Oya, T. Asai, T. Fukui, and Y. Amemiya, "A majority-logic device using an irreversible single-electron box," *IEEE TNANO*, vol. 2, no. 1, pp. 15–22, 2003.

[20] M. H. Sulieman and V. Beiu, "On single-electron technology full adders," *IEEE TNANO*, vol. 4, no. 6, pp. 669–680, 2005.

[21] H. A. Fahmy and R. A. Kiehl, "Complete logic family using tunneling-phase-logic devices," in *ICM*. IEEE, 1999, pp. 153–156.

[22] E. Testa, M. Soeken, L. G. Amar, and G. De Micheli, "Logic synthesis for established and emerging computing," *Proc. IEEE*, vol. 107, no. 1, pp. 165–184, 2018.

[23] L. Hellerman, "A catalog of three-variable or-invert and and-invert logical circuits," *IEEE Trans. Electron. Comput.*, no. 3, pp. 198–223, 1963.

[24] J. Cong and K. Minkovich, "Optimality study of logic synthesis for lut-based FPGAs," in *ACM/SIGDA FPGA*, 2006, pp. 33–40.

[25] W. L. Neto, L. Amarú, V. Possani, P. Vuillod, J. Luo, A. Mishchenko, and P.-E. Gaillardon, "Improving LUT-based optimization for ASICs," in *DAC*, 2022, pp. 421–426.

[26] L. Amaru, P.-E. Gaillardon, and G. De Micheli, "Majority-inverter graph: A new paradigm for logic optimization," *IEEE TCAD*, vol. 35, no. 5, pp. 806–819, 2015.

[27] W. J. Haaswijk, M. Soeken, L. Amaru, P.-E. Gaillardon, and G. De Micheli, "LUT mapping and optimization for majority-inverter graphs," in *IWLS*, no. CONF, 2016, pp. 1–8.

[28] K. Walus, G. Schulhof, G. Jullien, R. Zhang, and W. Wang, "Circuit design based on majority gates for applications with quantum-dot cellular automata," in *ACSSC*, vol. 2. IEEE, 2004, pp. 1354–1357.

[29] R. Zhang, P. Gupta, and N. K. Jha, "Synthesis of majority and minority networks and its applications to QCA, TPL and SET based nanotechnologies," in *CONF*. IEEE, 2005, pp. 229–234.

[30] K. Kong, Y. Shang, and R. Lu, "An optimized majority logic synthesis methodology for quantum-dot cellular automata," *IEEE TNANO*, vol. 9, no. 2, pp. 170–183, 2009.

[31] P. Wang, M. Niamat, S. Vemuru, M. Alam, and T. Killian, "Comprehensive majority/minority logic synthesis method," in *IEEE-NANO*. IEEE, 2013, pp. 694–697.

[32] P. Wang, M. Y. Niamat, S. R. Vemuru, M. Alam, and T. Killian, "Synthesis of majority/minority logic networks," *IEEE TNANO*, vol. 14, no. 3, pp. 473–483, 2015.

[33] V. K. Mishra and H. Thapliyal, "Heuristic based majority/minority logic synthesis for emerging technologies," in *VLSID*. IEEE, 2017, pp. 295–300.

[34] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "Boolean logic optimization in majority-inverter graphs," in *DAC*, 2015, pp. 1–6.

[35] M. Soeken, L. G. Amarù, P.-E. Gaillardon, and G. De Micheli, "Optimizing majority-inverter graphs with functional hashing," in *DATE*. IEEE, 2016, pp. 1030–1035.

[36] M. Soeken, L. G. Amaru, P.-E. Gaillardon, and G. De Micheli, "Exact synthesis of majority-inverter graphs and its applications," *IEEE TCAD*, vol. 36, no. 11, pp. 1842–1855, 2017.

[37] H. Riener, E. Testa, W. Haaswijk, A. Mishchenko, L. Amarù, G. De Micheli, and M. Soeken, "Scalable generic logic synthesis: One approach to rule them all," in *DAC*, 2019, pp. 1–6.

[38] H. Riener, E. Testa, L. Amaru, M. Soeken, and G. De Micheli, "Size optimization of migs with an application to QCA and stmg technologies," in *Nanoarch*, 2018, pp. 157–162.

[39] W. Haaswijk, M. Soeken, L. Amarú, P.-E. Gaillardon, and G. De Micheli, "A novel basis for logic rewriting," in *ASP-DAC*. Ieee, 2017, pp. 151–156.

[40] L. Amarú, P.-E. Gaillardon, A. Chattopadhyay, and G. De Micheli, "A sound and complete axiomatization of majority-n logic," *IEEE TC*, vol. 65, no. 9, pp. 2889–2895, 2015.

[41] A. Neutzling, F. S. Marranghello, J. M. Matos, A. Reis, and R. P. Ribas, "maj-n logic synthesis for emerging technology," *IEEE TCAD*, vol. 39, no. 3, pp. 747–751, 2019.

[42] Z. Chu, W. Haaswijk, M. Soeken, Y. Xia, L. Wang, and G. De Micheli, "Exact synthesis of Boolean functions in majority-of-five forms," in *ISCAS*. IEEE, 2019, pp. 1–5.

[43] E. C. Ferraz and A. C. R. Da Silva, "Linear optimization models for maj-3 and maj-5 exact synthesis," *IEEE Access*, vol. 10, pp. 130 518–130 531, 2022.

[44] M. Ajtai, J. Komlós, and E. Szemerédi, "An $o(nlogn)$ sorting network," in *Proc. ACM STOC*, 1983, pp. 1–9.

[45] D. Dor and U. Zwick, "Selecting the median," *SIAM J. Computing*, vol. 28, no. 5, pp. 1722–1758, 1999.

[46] E. Testa, M. Soeken, L. G. Amarù, W. Haaswijk, and G. De Micheli, "Mapping monotone boolean functions into majority," *IEEE TC*, vol. 68, no. 5, pp. 791–797, 2018.

[47] A. Chattopadhyay, D. Bhattacharjee, and S. Maitra, "Improved linear decomposition of majority and threshold boolean functions," *IEEE TCAD*, 2023.

[48] E. Testa, M. Soeken, O. Zografos, L. Amaru, P. Raghavan, R. Lauwereins, P.-E. Gaillardon, and G. De Micheli, "Inversion optimization in majority-inverter graphs," in *NANOARCH*. Ieee, 2016, pp. 15–20.

[49] E. Testa, O. Zografos, M. Soeken, A. Vaysset, M. Manfrini, R. Lauwereins, and G. De Micheli, "Inverter propagation and fan-out constraints for beyond-CMOS majority-based technologies," in *ISVLSI*. IEEE, 2017, pp. 164–169.

[50] E. Testa, M. Soeken, O. Zografos, F. Catthoor, and G. De Micheli, "Exact synthesis for logic synthesis applications with complex constraints," in *IWLS*, no. CONF, 2017, pp. 1–7.

[51] F. S. Torres, R. Wille, P. Niemann, and R. Drechsler, "An energy-aware model for the logic synthesis of quantum-dot cellular automata," *IEEE TCAD*, vol. 37, no. 12, pp. 3031–3041, 2018.

[52] R. K. Nath, B. Sen, and B. K. Sikdar, "Optimal synthesis of QCA logic circuit eliminating wire-crossings," *IET Circ. Devices Syst.*, vol. 11, no. 3, pp. 201–208, 2017.

[53] D. Bhattacharjee, L. Amaŕu, and A. Chattopadhyay, "Technology-aware logic synthesis for ReRAM based in-memory computing," in *DATE*. IEEE, 2018, pp. 1435–1440.

[54] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *ETS*. IEEE, 2013, pp. 1–6.

[55] T. Zhang, H. Jiang, W. Liu, F. Lombardi, L. Liu, S.-B. Ko, and J. Han, "Majority logic-based approximate multipliers for error-tolerant applications," in *Approximate Computing*. Springer, 2022, pp. 119–146.

[56] H. Cho and E. E. Swartzlander, "Adder designs and analyses for quantum-dot cellular automata," *IEEE TNANO*, vol. 6, no. 3, pp. 374–383, 2007.

[57] S. Perri and P. Corsonello, "New methodology for the design of efficient binary addition circuits in QCA," *IEEE TNANO*, vol. 11, no. 6, pp. 1192–1200, 2012.

[58] J. Maharaj and S. Muthurathinam, "Efficient majority logic subtractor design using multilayer crossover in quantum-dot cellular automata," *J. Nanophotonics*, vol. 14, no. 3, pp. 036 011–036 011, 2020.

[59] B. Aravinth and L. J. A. Marcilin, "Implementation of coplanar approximate adders in QCA," in *WiSPNET*. IEEE, 2016, pp. 680–684.

[60] M. Parameshwara and N. Maroof, "An area-efficient majority logic-based approximate adders with low delay for error-resilient applications," *Circuits, Syst. Signal Process.*, pp. 1–21, 2022.

[61] C. Labrado, H. Thapliyal, and F. Lombardi, "Design of majority logic based approximate arithmetic circuits," in *ISCAS*. IEEE, 2017, pp. 1–4.

[62] T. Zhang, W. Liu, E. McLarnon, M. O'Neill, and F. Lombardi, "Design of majority logic (ML) based approximate full adders," in *ISCAS*. IEEE, 2018, pp. 1–5.

[63] H. Cho and E. E. Swartzlander, "Adder and multiplier design in quantum-dot cellular automata," *IEEE TC*, vol. 58, no. 6, pp. 721–727, 2009.

[64] W. Liu, T. Zhang, E. McLarnon, M. O'Neill, P. Montuschi, and F. Lombardi, "Design and analysis of majority logic-based approximate adders and multipliers," *IEEE TETC*, vol. 9, no. 3, pp. 1609–1624, 2019.

[65] S. Perri, F. Spagnolo, F. Frustaci, and P. Corsonello, "Accuracy improved low-energy multi-bit approximate adders in QCA," *IEEE TCASII*, vol. 26, no. 7, pp. 3456–3460, 2021.

[66] Z. Chu, C. Shang, T. Zhang, Y. Xia, L. Wang, and W. Liu, "Efficient design of majority-logic-based approximate arithmetic circuits," *IEEE TVLSI*, vol. 30, no. 12, pp. 1827–1839, 2022.

[67] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE TC*, vol. 62, no. 9, pp. 1760–1771, 2012.

[68] C. Shang and Z. Chu, "Design of majority logic based 4-bit approximate subtractors and its application in divider," in *ASICON*. IEEE, 2021, pp. 1–4.

[69] W.-C. Yeh and C.-W. Jen, "High-speed Booth encoded parallel multiplier design," *IEEE TC*, vol. 49, no. 7, pp. 692–701, 2000.

[70] P. Behrooz, "Computer arithmetic: Algorithms and hardware designs," *Oxford University Press*, vol. 19, pp. 512 583–512 585, 2000.

[71] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE TC*, no. 1, pp. 14–17, 1964.

[72] L. Dadda, "Some schemes for parallel multipliers," *ALTA FREQUENZA*, vol. 34, pp. 349–356, 1965.

[73] T. Zhang, W. Liu, J. Han, and F. Lombardi, "Design and analysis of majority logic based approximate radix-4 Booth encoders," in *NANOARCH*. IEEE, 2019, pp. 1–6.

[74] T. Zhang, H. Jiang, H. Mo, W. Liu, F. Lombardi, L. Liu, and J. Han, "Design of majority logic-based approximate Booth multipliers for error-tolerant applications," *IEEE TNANO*, vol. 21, pp. 81–89, 2022.

[75] T. Zhang, H. Jiang, W. Liu, F. Lombardi, L. Liu, and J. Han, "Majority logic-based approximate recording adders for high-radix booth multipliers," in *IEEE NANO*. IEEE, 2022, pp. 519–522.

[76] M. H. Moaiyeri, F. Sabetzadeh, and S. Angizi, "An efficient majority-based compressor for approximate computing in the nano era," *Microsyst. Technol.*, vol. 24, no. 3, pp. 1589–1601, 2018.

[77] M. Taheri, A. Arasteh, S. Mohammadyan, A. Panahi, and K. Navi, "A novel majority based imprecise 4: 2 compressor with respect to the current and future vlsi industry," *Microprocess. Microsyst.*, vol. 73, p. 102962, 2020.

[78] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE TCASI*, vol. 66, no. 11, pp. 4200–4208, 2019.

[79] S. Angizi, H. Jiang, R. F. DeMara, J. Han, and D. Fan, "Majority-based spin-cmos primitives for approximate computing," *IEEE TNANO*, vol. 17, no. 4, pp. 795–806, 2018.

[80] Y. Zhang, C. He, T. Zhang, J. Han, and G. Xie, "Design and analysis of a majority logic based imprecise 6–2 compressor for approximate multipliers," in *IEEE-NANO*. IEEE, 2022, pp. 291–294.

[81] H. Jiang, S. Angizi, D. Fan, J. Han, and L. Liu, "Non-volatile approximate arithmetic circuits using scalable hybrid spin-CMOS majority gates," *IEEE TCASI*, vol. 68, no. 3, pp. 1217–1230, 2021.

[82] J. Han, E. R. Boykin, H. Chen, J. Liang, and J. A. Fortes, "On the reliability of computational structures using majority logic," *IEEE TNANO*, vol. 10, no. 5, pp. 1099–1112, 2011.

[83] S. Roy and V. Beiu, "Majority multiplexing-economical redundant fault-tolerant designs for nanoarchitectures," *IEEE TNANO*, vol. 4, no. 4, pp. 441–451, 2005.

[84] W. Ibrahim, V. Beiu, and M. H. Sulieman, "On the reliability of majority gates full adders," *IEEE TNANO*, vol. 7, no. 1, pp. 56–67, 2008.

[85] J. Han and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," *IEEE TNANO*, vol. 1, no. 4, pp. 201–208, 2002.

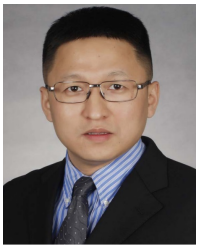
[86] M. Momenzadeh, J. Huang, M. B. Tahoori, and F. Lombardi, "On the evaluation of scaling of QCA devices in the presence of defects at manufacturing," *IEEE TNANO*, vol. 4, no. 6, pp. 740–743, 2005.

Tingting Zhang (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in the College of Electronic and Information Engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2016 and 2019, respectively. She is working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada, since Sep. 2019. Her research interests include approximate computing, Ising computing, combinatorial optimization and nanoelectronic circuits and systems.

Honglan Jiang (Member, IEEE) received the B.Sc. and Master's degrees in instrument science and technology from the Harbin Institute of Technology, Harbin, China, in 2011 and 2013, respectively, and the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 2018. From 2018 to 2021, she was a Postdoctoral Fellow with the School of Integrated Circuits, Tsinghua University, Beijing, China. She is currently an associate professor in the Department of Micro-Nano Electronic, Shanghai Jiao Tong University, Shanghai, China. Her research interests include approximate computing, reconfigurable computing, and stochastic computing.

Weiqiang Liu (Senior Member, IEEE) received the B.Sc. degree in information engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2006, and the Ph.D. degree in electronic engineering from Queen's University Belfast (QUB), Belfast, U.K., in 2012. He is currently a Professor and the Vice Dean of the College of Electronic and Information Engineering and the College of Integrated Circuits, NUAA. He has published two research books and over 200 leading journals and conference papers (over 80 IEEE and ACM journals, including eight invited articles). His research interests include energy efficient and secure computing integrated circuits and systems. He is also an Tutorial Organizer and a Speaker of DAC 2022, DATE 2022, IEEE ISCAS 2021, and COINS 2021. He is a member of IEEE NTC AdCom and CASCOM/VSA Technical Committee of IEEE CAS Society. He has been awarded the Prestigious Excellent Young Scholar Award by the National Natural Science Foundation of China in 2020 and Young Scientist Award by Fok Ying Tung Education Foundation, Ministry of Education, China, in 2022. He has been listed in the Stanford University's list of the top 2% scientists in the world. He is the Program Co-Chair of IEEE ARITH 2020 and ACM/IEEE NANOARCH 2022 and also a Technical Program Committee Member of a number of IEEE/ACM conferences, including DAC, DATE, ARITH, and ISCAS. He is the VP-Elect of TA of the IEEE Nanotechnology Council (NTC). He serves as an Associate Editor for IEEE Transactions on Circuits and Systems—I: Regular Papers, IEEE Transactions on Emerging Topics in Computing, and IEEE Transactions on Computers, a Steering Committee Member for IEEE Transactions on Very Large Scale Integration (VLSI) Systems, and the Guest Editor for Proceedings of the IEEE.

Fabrizio Lombardi (Life Fellow, IEEE) received the B.S. degree (Hons.) in electronic engineering from the University of Essex, U.K., in 1977, the master's degree in microwaves and modern optics, the Diploma degree in microwave engineering from the Microwave Research Unit, University College London, in 1978, and the Ph.D. degree from the University of London in 1982. He is currently the International Test Conference (ITC) Endowed Chair Professorship with Northeastern University, Boston, USA. His research interests are bio-inspired and nano manufacturing/computing, VLSI design, testing, and fault/defect tolerance of digital systems.



Leibo Liu (Senior Member, IEEE) received the B.S. degree in electronic engineering and the Ph.D. degree from the Institute of Microelectronics, Tsinghua University, Beijing, China, in 1999 and 2004, respectively. He is currently a Full Professor with the School of Integrated Circuits, Tsinghua University. His current research interests include reconfigurable computing, mobile computing, and very large-scale integration digital signal processing.

Seok-Bum Ko (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of Rhode Island, Kingston, RI, USA, in 2002. He worked as a member of technical staff at the Korea Telecom Research and Development Group, South Korea, from 1993 to 1998. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada. His research interests include computer arithmetic, computer architecture, deep learning processor architecture, efficient hardware implementation of compute-intensive applications, and biomedical engineering. He is a Senior Member of the IEEE Circuits and Systems Society and an Associate Editor of the IEEE Transactions on Very Large Scale Integration (VLSI) Systems, IEEE Access, and IET Computers and Digital Techniques. He was an Associate Editor of the IEEE Transactions on Circuits and Systems—I: Regular Papers from 2020 to 2021.

Jie Han (Senior Member, IEEE) received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from the Delft University of Technology, The Netherlands, in 2004. He is currently a Professor and Director of computer engineering with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include approximate computing, stochastic computing, reliability and fault tolerance, nanoelectronic circuits and systems, novel computational models for nanoscale, and biological applications. He was a recipient of the Best Paper Award at the International Symposium on Nanoscale Architectures (NANOARCH) 2015 and the Design, Automation and Test in Europe Conference (DATE) 2023; and the Best Paper Nominations at the Great Lakes Symposium on VLSI (GLSVLSI) 2015, NANOARCH 2016, the International Symposium on Quality Electronic Design (ISQED) 2018, and DATE 2022. He was nominated for the 2006 Christiaan Huygens Prize of Science by the Royal Dutch Academy of Science. His work was recognized by Science, for developing a theory of fault-tolerant nanocircuits in 2005. He served as the General Chair for NANOARCH 2021, GLSVLSI 2017, and the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT) 2013; and the Technical Program Committee Chair for NANOARCH 2022, GLSVLSI 2016, DFT 2012, and the Symposium on Stochastic and Approximate Computing for Signal Processing and Machine Learning in 2017. He serves (or has served) as an Associate Editor of the IEEE Transactions on Emerging Topics in Computing, the IEEE Transactions on Nanotechnology, the IEEE Circuits and Systems Magazine, the IEEE Open Journal of the Computer Society, Microelectronics Reliability, and the Journal of Electronic Testing: Test and Application (Springer Nature).