Non-Volatile Approximate Arithmetic Circuits using Scalable Hybrid Spin-CMOS Majority Gates

Honglan Jiang, *Member, IEEE,* Shaahin Angizi, *Member, IEEE,* Deliang Fan, *Member, IEEE,* Jie Han, *Senior Member, IEEE,* and Leibo Liu, *Senior Member, IEEE*

***Abstract*—In the nanoscale era, leakage/static power dissipation has become an inevitable and important issue for CMOS devices. To alleviate this issue, we propose to use spintronic devices with near-zero leakage power and non-volatility as key components in arithmetic circuits for error-resilient applications. To this end, spintronic threshold devices are first utilized to construct highly-scalable majority gates (MGs) based on spin-CMOS technology. These MGs are then used in the design of compressors for constructing multipliers and accumulators. For an MG-based compressor, the truth table of a conventional compressor is transformed to ensure that the outputs depend only on the number of input "1"s. To synthesize and optimize the MG-based circuits, a heuristic majority-inverter graph (HMIG) is further proposed for the design of an accurate and two approximate non-volatile 4-2 compressors (denoted as MG-EC, MG-AC1 and MG-AC2). Due to the high scalability of the MGs, approximate compressors with a larger number of inputs can be devised using the same method. Compared to previous designs, the proposed 4-2 compressors show shorter critical path delays and lower energy consumption; MG-AC1 and MG-AC2 also achieve a higher accuracy than state-of-the-art approximate designs. For achieving a similar image quality in image compression, the multiplier implementations using MG-AC1 and MG-AC2 result in more significant reductions in delay and energy than those using other approximate designs.**

***Index Terms*—Compressor, approximate computing, spin-CMOS majority gate, non-volatility, low leakage power, heuristic majority-inverter graph (HMIG).**

I. Introduction

With the scaling of CMOS technology, the increasing leakage (static) power consumption has become a severe issue to CMOS devices [1]. As an effective method, power gating has widely been considered to reduce leakage power [2]. However, sleep transistors and power-gating control circuits are required to switch off some parts of a system so they can be in a standby mode. As a result, the benefit of using power gating is reduced due to the power and area penalties introduced by the circuit, especially in deep-submicron technologies that suffer from more significant leakage power [3]. As a potential alternative to CMOS technology, spintronic devices possess promising complementary features including near-zero static power consumption, non-volatility, high integration density, and easy integration with CMOS processes [4].

As many arithmetic circuits in a large system do not need to work all the time, such as those in an Internet-of-Things (IoT) system, normally-off computing techniques have been developed for low-power designs. In such designs, the non-volatile elements play a key role [5], [6]. In a normally-off system, the non-volatile elements significantly improve the performance and energy efficiency by simplifying the boot-up step that is required for a conventional design after sleeping [7], [8]. Specifically, a large number of initialization operations are required to resume the operations in a conventional system from the sleeping mode, resulting in significant time and power loss. As non-volatile arithmetic circuits can retrieve the information after power-on, they can significantly simplify the resume process and save time and power dissipation. A fully non-volatile system can be powered on and off instantly without additional operations, resulting in no delay and very low power consumption [8]. A non-volatile processor has been explored to tolerate intermittent power supply [9]. Besides non-volatile memory, various non-volatile devices including flip-flops [7], adders [4], [10]–[15] and 4-2 compressors [16] have been proposed.

In addition, spintronic devices store the states in magnetic tunnel junctions (MTJs); thus, no additional register is required for pipelined designs [17]. Compared to a conventional CMOS-based system, spintronic circuits can improve the throughput with significantly lower hardware overhead.

Taking advantage of the inherent error tolerance of many applications such as multimedia processing and machine learning, approximate computing has been considered as a promising paradigm for improving the performance and/or energy efficiency of computing systems, with acceptable loss of accuracy [18]. Various CMOS-based approximate arithmetic circuits including adders, multipliers and dividers have been proposed [19]. As a key element in the partial product accumulation of a multiplier, compressors have been investigated for the design of approximate multipliers [16], [20]–[24].

To benefit from the synergy of spintronic devices and approximate computing, several approximate full adders (or 3-2 compressors) [11], [13], [16] and 4-2 compressors [16] have been developed using spin-CMOS technology. However, the existing spin-CMOS based approximate full adders show very low accuracy. The 4-2 compressors are usually designed by connecting two approximate (or accurate) full adders in series, resulting in a significantly low accuracy and long critical paths.

In this paper, non-volatile arithmetic circuits using spintronic device-based majority gates (MGs) are investigated for approximate computing. Although $(2m+1)$-input (m is a positive integer) scalable MGs have been presented in [16], only 3- and 5-input MGs are discussed and utilized in the circuit designs. Compared to the previous works, the new contributions are as follows.

- In addition to 3- and 5-input MGs, MGs with larger numbers of inputs such as 7 and 9 are exploited to achieve an efficient circuit design.

H. Jiang is with the Institute of Microelectronics, Tsinghua University, Beijing 100084, China. E-mail: jianghonglan@mail.tsinghua.edu.cn

S. Angizi and D. Fan are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA. E-mail: sangizi@asu.edu, dfan@asu.edu.

J. Han is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada. E-mail: jhan8@ualberta.ca.

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.

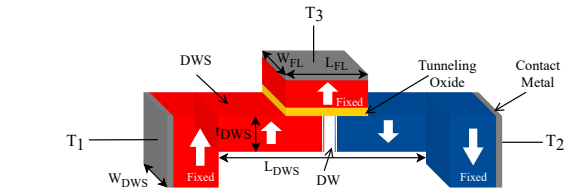
- To synthesize the compressor circuits based on the MGs with versatile inputs, the concept of a heuristic majority-inverter graph (HMIG) is proposed. Also, several transformation rules of $(2m+1)$-input MGs are introduced to simplify the HMIG.
- An accurate and two approximate 4-2 compressors (denoted as MG-EC, MG-AC1 and MG-AC2) are designed using the scalable MGs. Compared with the existing accurate designs, MG-EC consumes the lowest energy. MG-AC1 and MG-AC2 are more accurate and energy efficient than the other approximate designs. As the best design trade-off, MG-AC2 achieves 76.3% reduction in power dissipation, and 81.0% reduction in energy consumption with a 0.39% error rate, compared to the accurate design.
- The design methodology is generalized to compressors with larger number of inputs.
- Serving as the basic elements in non-volatile arithmetic circuits, the compressors are then used to construct multipliers and accumulators in an image processing application. For a similar processing result, MG-AC1 and MG-AC2 achieve more significant reductions in delay and energy than previous designs.

The remainder of this paper is organized as follows. Section II introduces the basic structure of spintronic threshold devices. Section III presents the design of scalable MGs based on hybrid spin-CMOS technology. The HMIG is described in Section IV for synthesizing the compressor designs using the scalable MGs. An accurate and two approximate 4-2 compressors are then obtained. In addition, a generalization of the proposed design approaches is discussed in this section. Section V evaluates the power dissipation and delay of MGs and compressors. Moreover, the accuracy of the compressors is assessed by using them in the multiplication and accumulation operations of an image processing application, presented in Section VI. The paper is concluded in Section VII.

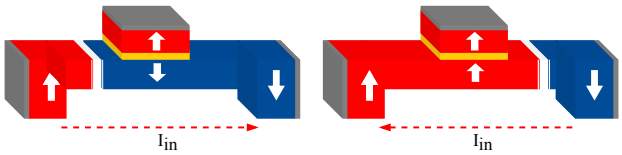
II. Spintronic Threshold Device Structure

Fig. 1(a) shows the structure of the utilized spintronic threshold device (spin-TD), consisting of a thin and short (50 nm × 20 nm × 2 nm) domain wall strip (DWS), a tunneling oxide layer and a small fixed magnetic layer (20 nm × 20 nm) with a fixed polarity [25]–[27]. The DWS is composed of a free magnetic domain in the middle and two fixed anti-parallel magnetic domains at the ends, where the polarity of the free magnetic domain can be changed by the current (larger than a critical current) between T_1 and T_2. Along the DWS, the magnetization transits from one direction to the opposite direction; the transition area is referred to as the domain wall (DW). When electrons pass through the DWS, they are spin-polarized and exert spin-transfer torque in and around the DW. The position of the DW depends on the direction and magnitude of the input current (I_{in}), as shown in Fig. 1(b). The free magnetic domain (Co/Ni) and the fixed magnetic layer (Co/Ni) with a tunneling oxide (MgO) deposited between them form a magnetic tunnel junction (MTJ), which is used to sense the state of the DWS [25], [28]. In this work, spin-TD is a perpendicularly magnetized CoNi/MgO heterostructure. We use PMA devices for both the DWS and MTJ. As the strip width is 20 nm, a Neel type DW is formed [29].

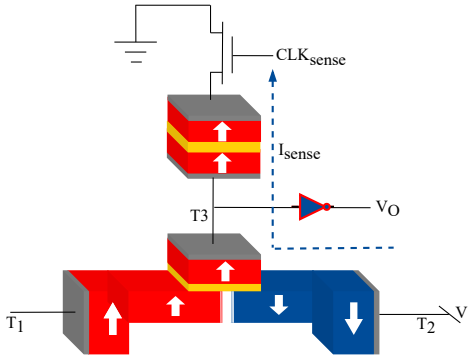
As shown in Fig. 1(b), the magnetization direction of the DWS is either anti-parallel (AP) or parallel (P) to the fixed layer on



(a) Spintronic threshold device (spin-TD).



(b) States of spin-TD determined by the input current.



(c) Spin-TD sense circuit

Fig. 1: A spintronic threshold device (spin-TD).

the top, resulting in a high and low resistance states for the MTJ, denoted as R_{AP} and R_P, respectively. Thus, the direction and relative value of the current passing through the DWS (compared with the critical current) can be detected by using a sense circuit, as shown in Fig. 1(c). There are two widely-explored methods to realize the sense operation in spin-based devices. First method is to leveraging the precharge sense amplifier (PCSA) [12] for resistive comparison. To generate the output, the sense voltage (V_{high}/V_{low}) corresponding to R_{AP}/R_P at one input of the sense amplifier is compared with a reference voltage that is typically set as the middle of high and low voltages through a reference resistance. Second method [16], [25] forms a resistive voltage divider between the spin device resistor (R_{AP}/R_P) and a fixed reference resistor typically implemented by an MTJ resistance (R_{ref}), on the top of Fig. 1(c). A transistor with a clock signal (CLK_{sense}) is used to separate the sense current (I_{sense}) and the input current (I_{in}). The sensing path goes from T_2 to T_3 and the ground. Considering R_{spinTD} as the spin-TD resistance (DWS+MTJ) and R_{ref} as the reference MTJ resistance (typically set as the average of R_{AP} and R_P), the voltage at T_3 is given by $V_{T_3} = \frac{R_{Ref}}{R_{Ref}+R_{spinTD}} \times V_{T_2}$. Then, the output CMOS inverter readily works as a thresholding logic to readout the spin-device current. Specifically, I_{sense} is low when R_{spinTD} is high; thus, the output voltage of the inverter V_O is high. Similarly, V_O is low when R_{spinTD} is low. In addition, the inverter works as a buffer to ensure a sufficient drive capability to pass the output signal to a succeeding stage. The choice between the two sensing schemes (PCSA and resistance divider) is a trade-off between speed and area-overhead. The resistance divider generates the

TABLE I: Device parameters used in simulation.

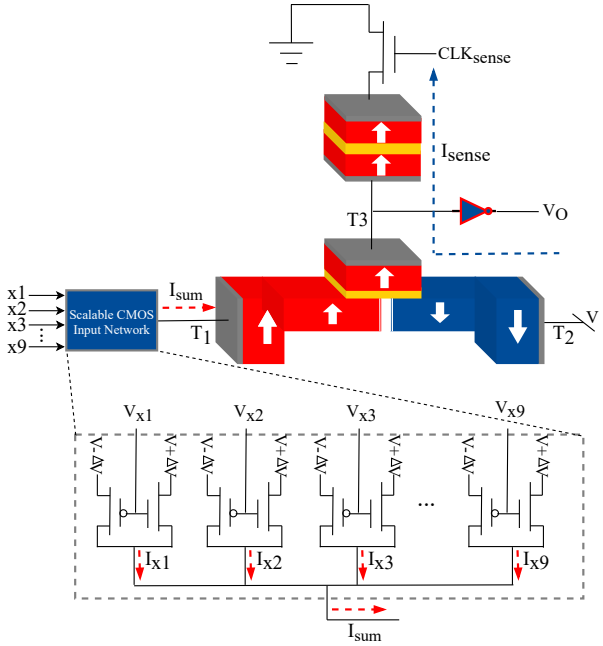
Symbol	Quantity	Values
α	Damping coefficient	0.02
K_u	Uniaxial anisotropy constant	3.5×10^5 J/m^3
M_s	Saturation magnetization	6.8×10^5 A/m
A_{ex}	Exchange stiffness	1.1×10^{-11} J/m
P	Polarization	0.6
t_{MgO}	MgO thickness of MTJ	1.5 nm
$(L.W.t)_{DWS}$	DWS dimension	$50\times20\times2$ nm^3

output using a smaller number of transistors (3T) than PCSA, but it is slower. On the other hand, PCSA is typically faster in terms of computation as it operates through the voltage comparison with a larger number of transistors (7T). In this work we aim for a smaller CMOS transistor overhead; thus, the resistance divider is used.

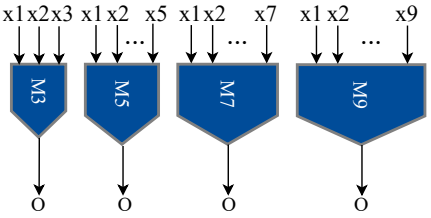
The minimum current magnitude required to switch the magnetization direction of the free layer is defined as the threshold of the spin-TD; it is determined by the critical current density and DW velocity. DW velocity is the speed at which the DW formed in DWS moves from one side to the other side. DW velocity almost linearly increases with the current density of DWS [25]. As per the simulation results from the object oriented micromagnetic framework (OOMMF) [30], with the device parameters shown in Table I, the magnetization of the DWS is fully switched within 1 ns, and the threshold (I_{th}) of the spin-TD is 30 uA corresponding to a DW velocity of $\sim$ 50 m/s. A detailed discussion for the fabrication and calibration of the utilized spin-TD has been shown in [16]. It is worth pointing out that, as clarified in [31], [32], when the DW's driving current is larger than the intrinsic deterministic threshold, thermal fluctuations impose a negligible impact on the DW velocity. Such effect depends almost linearly on the driving current, similar to a zero temperature case. In spin-TD, such driving current is way larger than the critical current of the DW motion. Therefore, the thermal noise effect on the spin-TD reliability is significantly smaller than the process variation of MTJs.

III. Hybrid Spin-CMOS Majority Gate

In a $(2m+1)$-input MG (m is a positive integer), the output is determined by the majority of its inputs, i.e., the output is asserted to be logic "1" only when more than m of the inputs are "1." Using the threshold characteristic of the spin-TD, a $(2m+1)$-input MG ($m=1, 2, \cdots$) is proposed, as shown in Fig. 2(a) [16]. The symbols of the MGs are shown in Fig. 2(b). The input port T_1 of the spin-TD is connected to a scalable CMOS input network consisting of $(2m+1)$ pairs of NMOS-PMOS input transistors. In this structure, all transistors work in deep triode region current sources (DTCS) by applying $V+\Delta V$ and $V-\Delta V$ to the source and drain, respectively. For example, three and five pairs of NMOS-PMOS input transistors are required to realize 3- and 5-input MGs, respectively. The proposed circuit works under the control of two clock signals ($CLK_{compute}$ and CLK_{sense}). We use CLK_{sense} to activate the sense path between T_2 and T_3, as explained earlier. In addition, we use the $CLK_{compute}$ to control the time interval in which the input voltages are applied to the input networks and accordingly current flows between T_1 and T_2 in each spin-TD. It could be considered as a synchronizing clock between different circuit levels. We did not show $CLK_{compute}$ for the sake of the simplicity in Fig. 2(a), however it readily synchronizes the output of each spin-TD device, through a transmission gate, fed to the inputs of the next level gate. A constant voltage V is connected to



(a) The circuit of the proposed scalable majority gate (MG).



(b) The symbols of the MGs.

Fig. 2: Scalable hybrid spin-CMOS implementation of $(2m+1)$-input MG.

the terminal T_2 of the spin-TD. In this design, $V = 500$ mV and $\Delta V = 50$ mV$\sim$ 110 mV (depending on the number of inputs), resulting in an ultra-small voltage drop and hence low power consumption. Moreover, the proposed scalable MG possesses the merits of near-zero static power and non-volatility, due to the use of the spin-TD.

During the compute phase when $CLK_{compute}$="1," voltages are applied at the gate of the input transistors, generating the input current flowing into (positive) or out of (negative) the spin-TD. The direction and magnitude of the total current at the intersection node are determined by the algebraic sum of the input currents (I_{x1}, I_{x2}, I_{x3}, $\cdots$). The sum current (I_{sum}) then controls the DW position in the DWS. In this design, the input transistors are properly sized to ensure that the current from each input branch is either $+30$ uA or -30 uA for a gate voltages of high ("1") or low ("0") respectively. Take the 3-input MG as an example, the input combination of (x1, x2, x3) = (1, 0, 1) leads to (I_{x1}, I_{x2}, I_{x3}) = ($+30$ uA, -30 uA, $+30$ uA), and an I_{sum} flowing into T_1 of $+30$ uA. As I_{sum} is larger than or equal to the threshold of the spin-TD, the DW moves towards T_1 and the bottom MTJ is in the high resistance state. To sense the resistance sate of the spin-TD, the CLK_{sense} is set to "1" in the sense phase. A voltage divider between the spin-TD's MTJ and the fixed reference MTJ is then formed, which results in a reliable output voltage by using an inverter.

Without modifying the device or circuit parameters, 5-, 7- and 9- input MGs can be obtained by changing the ΔV of a 3-input

MG. To avoid logic failure due to a large number of inputs and to increase the reliability, the number of inputs for an MG is limited to nine. Except for the majority function, other logic functions can be implemented by the scalable MGs with some constant inputs. For example, 2-input AND and OR gates can be obtained from a 3-input MG by setting one of the inputs to "0" and "1," respectively. 3-input AND and OR gates are obtained by setting two of the 5-input MG to "0" and "1," respectively.

A spin-based device like spin-TD is a basic element for constructing highly reliable, high-performance, non-volatile memory with the excellent endurance between 10^{14} and 10^{15} write cycles [33], [34] that is much higher than other non-volatile devices, e.g., ReRAM (10^{12} [35]) and PCM ($10^6 - 10^8$ [36]). Everspin's commercialized MRAM products, have been tested for 58 trillion cycles with no change in critical parameters and considered as "infinite endurance" cells [37]. While the other logic-in-memory counterpart designs based on resistive memories proposed to leverage ReRAM-based designs as a computing unit to fully implement logic operations, the spin-TD proposes an alternative way, not only taking advantage of a higher endurance memory/logic unit (spin-based device), but also providing a faster and more energy-efficient computation solution as will be discussed in the following sections.

IV. Approximate Compressor Designs

Compressors are commonly used for a fast accumulation of multiple signals; one of their most important applications is accumulating the partial products in a multiplier. Among the compressors with different numbers of inputs, the 5-input 4-2 compressor is the most widely used, except for the 3-2 compressor that is, in fact, a full adder (FA) [38]. As a compressor counts the number of "1"s in its inputs, its outputs can be independent of the permutation of the inputs, similar to an MG. Compared with a conventional logic gate, an MG can implement a more complex function, e.g., the carry output of an FA can directly be generated by using a 3-input MG instead of three AND gates and an OR gate (Cout = $x_0x_1 + x_1C_{in} + x_0C_{in}$). Moreover, MG can be the elementary element for building a digital logic circuit in various emerging nanotechnologies such as the magnetic logic, the quantum-dot cellular automata (QCA) and single-electron transistors. Using the 3-input MG, accurate and approximate FAs [39], [40], and approximate 4-2 compressors [40], [41] have been devised based on MTJ and QCA. In [39], three MGs and two inverters are utilized for constructing an accurate FA, whereas only an MG and an inverter are used in the approximate FA proposed in [40]. The approximate 4-2 compressors presented in [40], [41] consist of two MGs and one or two inverters, resulting in a large number of errors.

An MG with a larger number of inputs such as 5-, 7- or 9-input MG can implement much more complex functions. Moreover, as analyzed in Section III, increasing the number of inputs of the scalable MG does not need to modify the device parameters of the spin-TD in a 3-input MG. This indicates that, compared with a 3-input MG, a 5-, 7- or 9-input MG can implement more complex functions with only a marginally increased hardware overhead. Thus, by using the scalable MGs with variable inputs, a non-volatile compressor can be designed with an efficient hardware structure. In this paper, the compressors are designed by using majority functions implemented by MGs; the output signals are assigned with respect to the number of input "1"s.

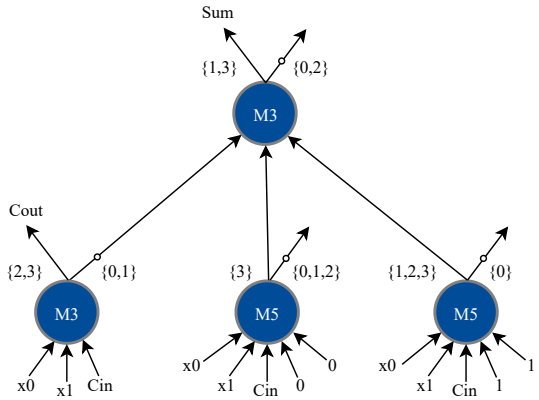


Fig. 3: A heuristic majority-inverter graph for the full adder.

To efficiently synthesize and optimize Boolean functions using exclusively 3-input majority gates and inverters, a logic representation structure denoted as the majority-inverter graph (MIG) has been presented [42]. In this structure, an MIG is derived from an AND/OR/Inverter graph (AOIG) and optimized using the transformation rules. In general, an AOIG uses AND gates, OR gates and inverters to implement a Boolean function. An MIG is obtained by replacing the AND and OR gates in the AOIG by 3-input MGs with a fixed input "0" and "1", respectively. Several transformation rules of 3-input MGs are then derived and used to optimize the MIG. As this methodology is not efficient for devising a circuit using MGs with more than 3 inputs, a heuristic MIG (HMIG) is proposed in this paper to assist the synthesis of compressors.

A. The Heuristic Majority-Inverter Graph

In an HMIG, a node denotes an MG, and its regular and complemented outputs are represented by an edge and an edge with a circle, respectively, as shown in Fig. 3. As the most important parameter, the possible number of "1"s in the primary inputs denoted as N_{in} plays a key role in the graph. The set of N_{in} triggering an edge is marked beside it, i.e., the edge is true when N_{in} belongs to the set marked beside it. For example in Fig. 3, the sets are $\{2,3\}$ and $\{0,1\}$ for each of the two output edges of the lower left M3.

Algorithm 1 presents the process of constructing an HMIG for an MG-based compressor. The input arguments are the numbers of the input and output signals for the circuit, and the sets of N_{in} that trigger each output signal, denoted as $O_i = \{n_{i,1}, n_{i,2}, \cdots, n_{i,k_i}\}$, where k_i is the number of elements in O_i. We aim to use as few MGs as possible; thus, the number of inputs for an MG at the first stage (on the bottom of a graph) is equal to or larger than the number of the primary inputs (line 1 in Algorithm 1). The redundant inputs are connected to "0"s and "1"s to generate various majority functions, as shown in line 9. Based on the current MG set G_{org}, a majority function f_i is then derived to implement the O_i, where more MGs, edges and stages can be required. When an f_i can be found based on the current G_{org}, the utilized MGs and edges are recorded as G_{sel} and E_{sel}, see lines 12 to 15. Otherwise, G_{org} is expanded by appending MGs with a larger number of inputs, see line 17. This process is repeated until all majority functions are constructed for all the output signals. To implement more complex majority functions, the output edges of one stage are selected as the inputs of the MGs at the other stages. For simplicity, 3-input and 5-input MGs

are usually utilized in the stages higher than the first. Except for the first stage, the set of N_{in} for an MG output edge is determined by both the majority function and the sets for the input edges. For example, the output edge of a 3-input MG has a set consisting of the elements belonging to at least two sets of its input edges, while it is the set with elements belonging to at least three input sets for a 5-input MG. Note that the connections of the edges and MGs are not limited to neighboring stages.

Algorithm 1 Algorithm of constructing a heuristic majority-inverter graph for an MG-based compressor.

Input: n_{in}; *// the number of input signals.*
n_{out}; *// the number of output signals.*
$O_i = \{n_{i,1}, n_{i,2}, \cdots, n_{i,k_i}\}$; *// the set of N_{in} triggering the output O_i, $i = 1, 2, \cdots, n_{out}$.*

Output: $HMIG = \{G_o, E_o\}$; *// G_o is the required MGs, E_o is the directed edges networking MGs.*

// Initialization

1: $j = \lceil n_{in} \rceil$; *// $\lceil n_{in} \rceil$ is the nearest odd number equal to or larger than n_{in}.*
2: $G_{org} = \{\}$; $G = \{\}$; $E = \{\}$;
3: **for** $i = 1$ to n_{out} **do**
4: $Flag = 0$;
5: **while** $Flag = 0$ **do**
6: **if** $j = n_{in}$ **then**
7: $G_{add} = \{M_j, \overline{M_j}\}$; *// M_j is an MG with j inputs; $\overline{M_j}$ is the complement of M_j.*
8: **else**
9: $G_{add} = \{M_{j,0}, \overline{M_{j,0}}, M_{j,1}, \overline{M_{j,1}}\}$; *// $M_{j,0}$ and $M_{j,1}$ are the M_j with 0s and 1s for the redundant input ports, respectively.*
10: **end if**
11: $G_{org} = \{G_{org}, G_{add}\}$
12: **if** $O_i = f_i(G_{org})$ **then** *// f_i is a majority function implemented by using MGs*
13: $G = \{G, G_{sel}\}$; *// G_{sel} contains the MGs for implementing function f_i*
14: $E = \{E, E_{sel}\}$; *// E_{sel} contains the directed edges connecting the MGs in G_{sel}*
15: $Flag = 1$;
16: **else**
17: $j = j + 2$;
18: **end if**
19: **end while**
20: **end for**
21: Removing the repeated MGs and edges in G and E.
22: Searching for the MGs and edges satisfying the transformation rules;
23: Simplifying the G and E using the transformation rules;
24: $HMIG = \{G_o, E_o\}$; *// G_o and E_o are the resultant MGs and edges after performing simplification.*

Consequently, an output of a compressor is implemented by constructing an edge with a set of N_{in} that is equal to its triggering condition. Generally, by using majority functions, a set of consecutive numbers, e.g., $\{2,3\}$ or $\{0,1\}$, is easier to be detected than that of non-consecutive numbers, e.g, $\{1,3\}$. To simplify the circuit, thus, the numbers in the sets of N_{in} triggering the output signals of a compressor should be made consecutive if possible. In some cases, a small approximation to ensure a set of consecutive elements for an output can save significant hardware resources.

As the set of N_{in} for an output varies with the function to be implemented, various majority functions can be required in an HMIG, besides the regular MG with the same number of inputs as the primary inputs. A straightforward way to implement a different majority function is using a larger MG and setting its redundant inputs to "0"s or "1"s. Let M_{2m+1} be the majority operator of $(2m+1)$ inputs, x_i be a primary input ($i = 1, 2, \cdots, n$), and $\overline{M_{2m+1}}$ be the complement of M_{2m+1}. The following inferences are fundamental to constructing an HMIG.

$$M_{2m+1}(x_1, \cdots, x_n, 0, \cdots, 0) = 1 \Rightarrow N_{in} \in \{m+1, \cdots, n\}, \quad (1)$$

and

$$M_{2m+1}(x_1, \cdots, x_n, 1, \cdots, 1) = 1 \Rightarrow N_{in} \in \{n-m, \cdots, n\}, \quad (2)$$

where $(n-1)/2 \le m \le n-1$. (1) shows that whether the number of "1"s in the n primary inputs belongs to $\{m+1, \cdots, n\}$ can be detected by using a $(2m+1)$-input MG with $(2m-n+1)$ input "0"s. Likewise, $M_{2m+1}(x_1, \cdots, x_n, 1, \cdots, 1)$ recognizes the input cases with an $N_{in} \in \{n-m, \cdots, n\}$. When $m = n-1$, $M_{2m+1}(x_1, \cdots, x_n, 0, \cdots, 0)$ and $M_{2m+1}(x_1, \cdots, x_n, 1, \cdots, 1)$ implement an n-input AND and OR gates, respectively. Similarly, we can have

$$\overline{M_{2m+1}(x_1, \cdots, x_n, 0, \cdots, 0)} = 1 \Rightarrow N_{in} \in \{0, \cdots, m\}, \quad (3)$$

and

$$\overline{M_{2m+1}(x_1, \cdots, x_n, 1, \cdots, 1)} = 1 \Rightarrow N_{in} \in \{0, \cdots, n-m-1\}, \quad (4)$$

Thus, the N_{in} belongs to a set of any consecutive numbers can be detected by using the following features.

$$M_3\Big(M_{2m_2+1}(x_1, \cdots, x_n, 0, \cdots, 0), \overline{M_{2m_1+1}(x_1, \cdots, x_n, 1, \cdots, 1)}, 0\Big) = 1 \Rightarrow N_{in} \in \{m_2+1, \cdots, n-m_1-1\}, \quad (5)$$

$$M_3\Big(\overline{M_{2m_2+1}(x_1, \cdots, x_n, 0, \cdots, 0)}, M_{2m_1+1}(x_1, \cdots, x_n, 1, \cdots, 1), 0\Big) = 1 \Rightarrow N_{in} \in \{n-m_1, \cdots, m_2\}, \quad (6)$$

where $(n-1)/2 \le m_1, m_2 \le n-1$. $m_2 \le n-m_1-2$ for (5), and $m_2 \ge n-m_1$ for (6). To this end, an edge with a set of arbitrary elements can be obtained by using (5) and (6) with the help of the AND and OR operations based on (1) and (2).

Take the FA with three inputs (i.e., x_0, x_1, C_{in}) as an example, its output signals, sum (Sum) and carry-out (Cout) are given by: Cout = 1 if $N_{in} \in \{2,3\}$, and Sum = 1 if $N_{in} \in \{1,3\}$, i.e., $O_{Cout} = \{2,3\}$ and $O_{Sum} = \{1,3\}$ in Algorithm 1. By using Algorithm 1 without the simplification (lines 22 and 23), an HMIG is constructed for the FA, as shown in Fig. 3. By exploiting (1) to (4), an M_3 and two M_5 in the first stage generate the universal sets of N_{in} for the three primary inputs. Here, the universal sets mean that any possible set can be obtained from them via proper connections. For example, the set $\{1,3\}$ triggering the Sum signal is achieved by using an M_3. Finally, two 3-input MGs and two 5-input MGs are required to implement the FA. However, this HMIG can be simplified by using the transformation rules of MGs introduced next.

To simplify an HMIG, some basic transformation rules for 3-input MGs [42] are extended to $(2m+1)$-input MGs in this paper. They are

Commutativity

$$\begin{aligned} M_{2m+1}(x_1,x_2,\cdots,x_{2m+1}) &= M_{2m+1}(x_2,x_1,\cdots,x_{2m+1}) \\ &= \\ &\cdots \\ &= M_{2m+1}(x_{2m+1},x_{2m},\cdots,x_1), \end{aligned} \tag{7}$$

Majority

$$\begin{aligned} &M_{2m+1}(x_1,\cdots,x_i,\cdots,x_{2m},\overline{x_i}) \\ &\quad = M_{2m-1}(x_1,\cdots,x_{i-1},x_{i+1},\cdots,x_{2m}), \end{aligned} \tag{8}$$

Inverter Propagation

$$M_{2m+1}(\overline{x_1},\overline{x_2},\cdots,\overline{x_{2m+1}}) = \overline{M_{2m+1}(x_1,x_2,\cdots,x_{2m+1})}, \tag{9}$$

where $\overline{x_i}$ is the complement of x_i.

To further exploit an HMIG, some complex majority operations can be simplified by using the following transformation rules.

$$\begin{aligned} &M_3\Big(M^Z, \\ &\quad M_{2m+1}(x_1,\cdots,x_n,0,\cdots,0), M_{2m+1}(x_1,\cdots,x_n,1,\cdots,1)\Big), \\ &= M_{2m+1}\Big(x_1,\cdots,x_n,M^Z,\cdots,M^Z\Big) \end{aligned} \tag{10}$$

where M^Z is a majority function detecting an arbitrary set of N_{in} denoted as Z, i.e., $M^Z = 1$ when $N_{in} \in Z$.

Proof: As per (1) and (2), the outputs of $M_{2m+1}(x_1,\cdots,x_n,0,\cdots,0)$ and $M_{2m+1}(x_1,\cdots,x_n,1,\cdots,1)$ are true when N_{in} is in $\{m+1,\cdots,n\}$ and $\{n-m\cdots,n\}$, respectively. The intersection of two of the three sets, Z, $\{m+1,\cdots,n\}$ and $\{n-m,\cdots,n\}$, is $(\{n-m,\cdots,m\}\cap Z)\cup\{m+1,\cdots,n\}$, which triggers the output of the M_3. On the right hand side of (10), the output of M_{2m+1} is true for $N_{in} \in \{m+1,\cdots,n\}$ regardless of M^Z. When $N_{in} \in Z$, the number of "1"s in the inputs of the M_{2m+1} is $N_{in}+2m-n+1$ due to $M^Z = 1$; so the output of M_{2m+1} is also true for the cases of $(N_{in}+2m-n+1 \geq m+1)\cap(N_{in}\in Z) \Rightarrow N_{in} \in (\{n-m,\cdots,n\}\cap Z)$. Hence, the M_{2m+1} on the right hand side of (10) detects whether the N_{in} is in $(\{n-m,\cdots,m\}\cap Z)\cup\{m+1,\cdots,n\}$, which implements the same function as the four MGs on the left hand side.

In (10), the majority function using four MGs (an M_3, an M^Z and two M_{2m+1}) on the left hand side can be implemented by using the two MGs (an M_{2m+1} and an M^Z) on the right hand side, which saves two MGs. Hence, the three MGs on the right of Fig. 3 can be replaced by an M_5, as shown in Fig. 4. Finally, an FA can be implemented by a 3-input MG and a 5-input MG.

Similarly, the following transformation equation can be achieved.

$$\begin{aligned} &M_3\Big(M_{2m_2+1}(x_1,\cdots,x_n,1,\cdots,1), \\ &\quad M_{2m_1+1}(x_1,\cdots,x_n,0,\cdots,0), \overline{M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)}\Big), \\ &= M_{2m_2+1}\Big(x_1,\cdots,x_n,\overline{M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)},\cdots, \\ &\quad \overline{M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)},1,\cdots,1\Big) \end{aligned} \tag{11}$$

where $(n-1)/2 \leq m_1, m_2 \leq n-1$, $m_2 \geq m_1$, the numbers of input "1"s and $\overline{M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)}$ for the M_{2m_2+1} on the right hand side of (11) are (m_2-m_1) and (m_2+m_1-n+1), respectively.

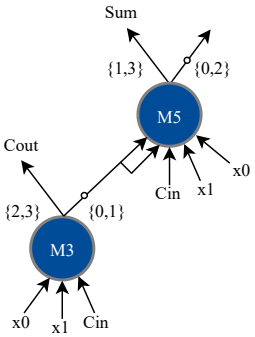


Fig. 4: A simplified HMIG for the accurate full adder.

Proof: The formulas (2), (1) and (4) show that $M_{2m_2+1}(x_1,\cdots,x_n,1,\cdots,1)$, $M_{2m_1+1}(x_1,\cdots,x_n,0,\cdots,0)$ and $M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)$ are true if N_{in} is in $\{n-m_2,\cdots,n\}$, $\{m_1+1,\cdots,n\}$ and $\{0,\cdots,n-m_1-1\}$, respectively. Thus, the output of the M_3 is true when N_{in} belongs to the intersection of two of these three sets, which is $\{n-m_2,\cdots,n\}\cap(\{0,\cdots,n-m_1-1\}\cup\{m_1+1,\cdots,n\}) = \{n-m_2,\cdots,n-m_1-1\}\cup\{m_1+1,\cdots,n\}$. On the right hand side of (11), when $\overline{M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)} = 0$ ($N_{in} \in \{n-m_1,\cdots,n\}$), the output of M_{2m_2+1} is true for $N_{in}+m_2-m_1 \geq m_2+1 \Rightarrow N_{in} \in \{m_1+1,\cdots,n\}$; in this case, the output of M_{2m_2+1} is true when $N_{in} \in \{n-m_1,\cdots,n\}\cap\{m_1+1,\cdots,n\} = \{m_1+1,\cdots,n\}$. When $\overline{M_{2m_1+1}(x_1,\cdots,x_n,1,\cdots,1)} = 1$ ($N_{in} \in \{0,\cdots,n-m_1-1\}$), the number of "1"s in the inputs of the M_{2m_2+1} is $N_{in}+2m_2-n+1$; the output of M_{2m_2+1} is true for the cases of $(N_{in}+2m_2-n+1 \geq m_2+1)\cap(N_{in}\in\{0,\cdots,n-m_1-1\}) \Rightarrow N_{in} \in \{n-m_2,\cdots,n-m_1-1\}$. Overall, the M_{2m_2+1} on the right hand side of (11) recognizes the inputs with an N_{in} belonging to $\{n-m_2,\cdots,n-m_1-1\}\cup\{m_1+1,\cdots,n\}$. Thus, the equation (11) holds.

To synthesize a function, various HMIGs can be generated by using different types of MGs and connections. The selection and connection of MGs also determine whether a transformation rule can be applied for a further simplification. An area-optimized design is obtained by comparing the total number of required MGs, while the one with the minimum number of stages results in a delay-optimized design.

B. 4-2 Compressors

A conventional implementation of a 4-2 compressor consists of two cascaded FAs [38]. Its truth table is shown in Table II (the bits on the left side of the arrows), where the inputs and Sum output have a weight of $2^0 = 1$, and the Carry and Cout have a weight of $2^1 = 2$ in the binary representation. The hardware overhead of an accurate 4-2 compressor is nearly twice of that of an FA. For an efficient design, the truth table of a 4-2 compressor is transformed by exploiting the commutativity of the signals with the same weights, to ensure an HMIG can be constructed.

As per Algorithm 1, the truth table of the compressor is modified so that the output signals are determined by the number of "1"s in its inputs (N_{in}). Specifically, Cout is aimed to be implemented using a 5-input MG to reduce the critical path. To this end, the Cout is flipped for some input combinations to ensure that Cout is "1" when the inputs have three or more "1"s, and

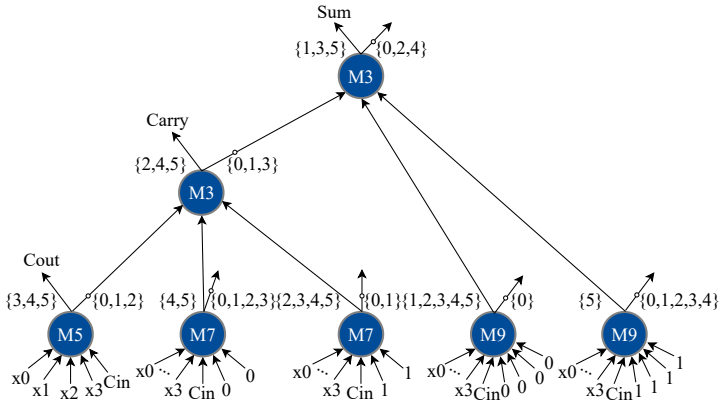


Fig. 5: A heuristic majority-inverter graph for the 4:2 compressor.

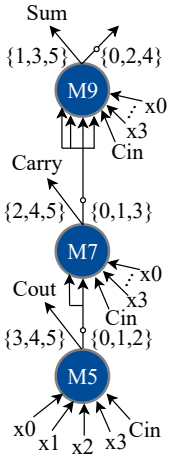


Fig. 6: The simplified HMIG for the 4-2 compressor.

TABLE II: The truth table of an accurate and approximate 4-2 compressors.

Inputs					Outputs		
Cin	x3	x2	x1	x0	Sum→Sum′	Carry	Cout
0	0	0	0	0	0→1 ✘	0	0
0	0	0	0	1	1	0	0
0	0	0	1	0	1	0	0
0	0	0	1	1	0	1	0
0	0	1	0	0	1	0	0
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	1	1	1→0	0→1
0	1	0	0	0	1	0	0
0	1	0	0	1	0	0→1	1→0
0	1	0	1	0	0	0→1	1→0
0	1	0	1	1	1	1→0	0→1
0	1	1	0	0	0	0→1	1→0
0	1	1	0	1	1	1→0	0→1
0	1	1	1	0	1	1→0	0→1
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	0
1	0	0	0	1	0	0→1	1→0
1	0	0	1	0	0	0→1	1→0
1	0	0	1	1	1	1→0	0→1
1	0	1	0	0	0	0→1	1→0
1	0	1	0	1	1	1→0	0→1
1	0	1	1	0	1	1→0	0→1
1	0	1	1	1	0	1	1
1	1	0	0	0	0	0→1	1→0
1	1	0	0	1	1	0	1
1	1	0	1	0	1	0	1
1	1	0	1	1	0	1	1
1	1	1	0	0	1	0	1
1	1	1	0	1	0	1	1
1	1	1	1	0	0	1	1
1	1	1	1	1	1→0 ✘	1	1

Cout is "0" otherwise, as indicated by the right arrows in the rightmost column in Table II. To keep an accurate computation result, the output Carry is also flipped at the input combinations for which Cout is flipped. Consequently, the outputs of an accurate 4-2 compressor are given by

$$\text{Cout} = \begin{cases} 1 & \text{if } N_{in} \in \{3,4,5\} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

$$\text{Carry} = \begin{cases} 1 & \text{if } N_{in} \in \{2,4,5\} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

and

$$\text{Sum} = \begin{cases} 1 & \text{if } N_{in} \in \{1,3,5\} \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

The HMIG for the 4-2 compressor is then constructed, as shown in Fig. 5. A 5-input, two 7-input and two 9-input MGs are used at the first stage to realize the functions detecting various N_{in}. As per (12), the output signal Cout can be implemented by

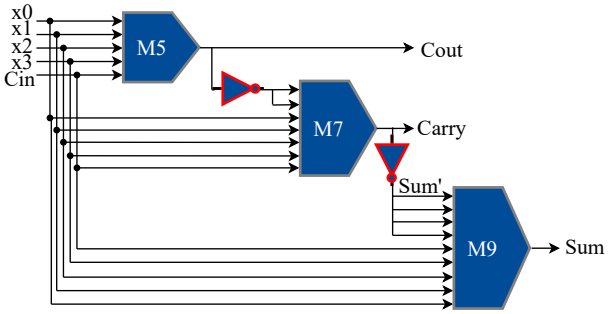


Fig. 7: The accurate (MG-EC) and approximate (MG-AC1) MG-based 4-2 compressor.

a 5-input MG. To implement Carry, the edges resulting in an intersection containing as many as possible elements in $\{2,4,5\}$ are considered as the inputs of the MGs at the next stage. A 3-input MG is then used for generating the Carry. Additional two 9-input MGs and a 3-input MG are required for implementing the Sum. As can be seen, the two 7-input MGs and 9-input MGs connected to the 3-input MGs can be simplified by using (10). Finally, the simplified HMIG shown in Fig. 6 is obtained for the 4-2 compressor. Thus, a 4-2 compressor can be implemented by using three MGs, as shown in Fig. 7.

To reduce the complexity of a 4-2 compressor, the Sum can be approximated by the inverted Carry, as shown in Table II and Fig. 7. The Sum′ is then given by

$$\text{Sum}' = \begin{cases} 1 & \text{if } N_{in} \in \{0,1,3\} \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

In this case, two errors occur when the inputs are '00000' and '11111.'

As shown in Fig. 7, three and two MGs are required for the accurate and approximate 4-2 compressors, denoted as MG-EC and MG-AC1, respectively. The critical path of MG-EC consists of three MGs and two inverters, while it is two MGs and two inverters for the MG-AC1. Considering the two errors in Table II, the probability that an error occurs, i.e., the error rate (ER), is $0.5^5 + 0.5^5 = 6.25\%$ if "0" and "1" are equally likely to occur to each input ($P(0) = 0.5$). However, under this assumption, the probabilities of being "0" and "1" for each input are 0.75 ($P(0) = 0.75$) and 0.25 respectively, when the compressor is used for the partial product accumulation of a multiplier. In this case, the ER of the approximate compressor is $0.75^5 + 0.25^5 = 23.83\%$.

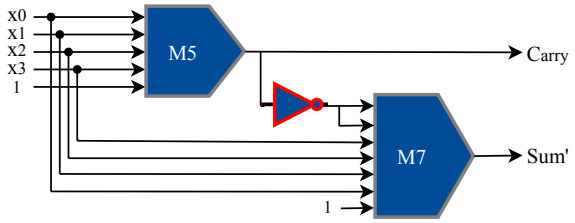


Fig. 8: The approximate 4-2 compressor design MG-AC2.

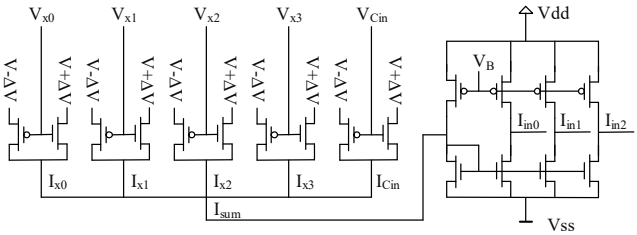


Fig. 9: Current generation and replication circuit using a current mirror.

Here, we assumed the inputs of the compressor are mutually independent, which is not exactly the case when compressors are cascaded in the partial product accumulation. However, compared to the inputs, the partial products of a multiplier have a higher probability to be "0" due to the AND operation. This indicates that the proposed approximate 4-2 compressor would result in more errors in the implementation of a multiplier.

As a partial product in a multiplier has a higher probability to be "0" than "1," the case '00000' is more likely to occur than '11111.' Also, the relative error compared to the accurate output for the case '00000' is larger than that for '11111.' Thus, to guarantee a high accuracy, the error due to the inputs of all zeros should be avoided. This can be achieved by reducing the number of the compressor inputs.

To further simplify a 4-2 compressor, the carry input (Cin) is usually ignored. The outputs of a 4-2 compressor are also reduced by 1 bit, so the simplified 4-2 compressor has 4 inputs and 2 outputs, referred to as MG-AC2. In this design, the Carry output is "1" when $N_{in} \in \{2,3,4\}$. This function, therefore, can be implemented by a 5-input MG with a constant input "1." The Sum$'$ is "1" when $N_{in} \in \{1,3,4\}$. Fig. 8 is then obtained by simplifying the HMIG using (11). In this case, the approximate and accurate results are 3 and 4 in decimal representation, respectively, when all inputs are "1"s. Thus, the ER is $0.5^4 = 6.25\%$ when $P(0) = 0.5$, whereas it is $0.25^4 = 0.39\%$ for $P(0) = 0.75$. Note that MG-AC2 works as an accurate FA when one of the inputs is set to "0."

Due to the use of 7- and 9-input MGs, the critical path of the proposed compressors and the ERs of the MG-ACs are reduced compared to the designs using only 3-input and 5-input MGs [16]. However, the number of input transistors increases with the inputs of the scalable MG, which may penalize the overall hardware overhead. As shown in Fig. 2, a pair of NMOS-PMOS transistors are required for each input of the MG, which means 21 and 12 pairs of transistors are demanded for the MGs in MG-EC and MG-AC1/2, respectively. As the MGs in the circuits share the same inputs, five pairs of transistors and a current mirror are used in the proposed designs to generate the sum of the primary input currents for the Spin-TDs. Thus, the numbers of transistors used in the proposed designs are reduced. The current generation and replication circuit for the three MGs in Fig. 7 is shown in Fig. 9. $V_{x0} \sim V_{x3}$ and V_{Cin} are the input voltages of a 4-2 compressor, and I_{ini} $(i = 0,1,2)$ is the current input to a spin-TD of an MG.

C. Generalization

Although not frequently used, compressors with a larger number of inputs may result in a lower critical path and power consumption in the accumulation of some specific numbers of inputs, e.g., a well-designed 6-input compressor can be more efficient than a 4-2 compressor when accumulating 6 inputs. Using the same design approaches, i.e., applying the commutativity of the signals with the same weights and proper approximations to make sure the outputs depend only on the N_{in}, approximate compressors with a different number of inputs can be devised based on the HMIG. Also, emerging technologies such as memristors [43] can be used for constructing the scalable MGs and the approximate compressors.

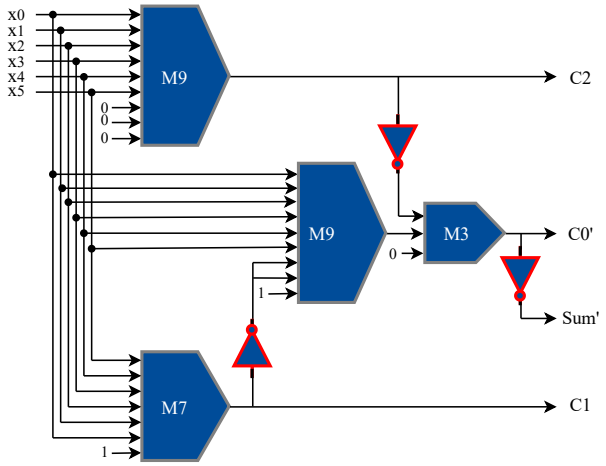


Fig. 10: The approximate 6-input compressor design 1.

For instance, an approximate 6-input compressor can be devised by following the same process. Table III shows the transformed truth table for an approximate 6-input compressor, where the C0, C1 and C2 are the carry outputs with a weight of $2^1 = 2$. In this design, C2 is "1" for the input cases of $N_{in} \in \{4,5\}$; C1 is "1" if $N_{in} \in \{3,4,5,6\}$; C0$'$ is "1" for the input cases with two or four "1"s; Sum$'$ is the inverted C0$'$. Two errors could occur for the input cases of '000000' and '111111.' The ERs are 3.13% and 17.82% for $P(0) = 0.5$ and $P(0) = 0.75$, respectively. To obtain a circuit implementation, an HMIG is then constructed and simplified by using the transformation rules. Finally, four MGs are utilized for implementing the approximate 6-input compressor, as shown in Fig. 10. In the HMIG for this design, only 3-input MGs are considered for the second and third stages and simplified by using (11). The critical path consists of three MGs and two inverters. By using a 5-input MG at the second stage of the HMIG, the critical path can be reduced by an MG; however, five MGs are required in total, as shown in Fig. 11. Thus, the circuit designs with a small area and a short critical path can be respectively obtained for the approximate 6-input compressor with a truth table shown in Table III.

Similar to the 4-2 compressor, an output bit of the 6-input compressor can be ignored, as shown in Table IV. In this design, the C1 is "1" when $N_{in} \in \{4,5,6\}$, which can be implemented by using a 7-input MG with a constant input "1." The C0$'$ is "1" for

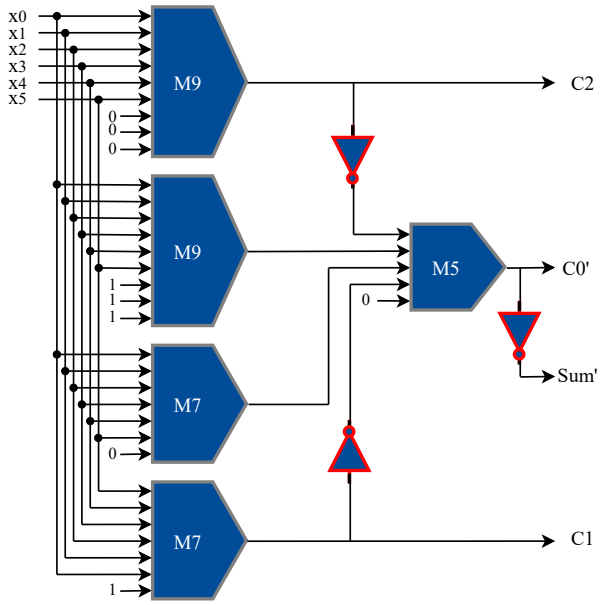


Fig. 11: The approximate 6-input compressor design 2.

TABLE III: The transformed truth table for the proposed approximate 6-input compressor design with three carry outputs.

N_{in}	Sum→Sum′	C0→ C0′	C1	C2
0	0→1 ✘	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	0	1	0
4	0	1	1	0
5	1	0	1	1
6	0→1 ✘	1→0 ✘	1	1

the input cases of $N_{in} \in \{2,3,4,5,6\}$, generated by a 9-input MG with three input "1"s. As the Sum′ is unchanged from Table III, the same circuit structure as in Fig. 11 is utilized to compute it. Using the inverter propagation rule shown in (9), Fig. 12 is obtained. Two MGs and an inverter are on the critical path. Errors occur for the input cases of '000000' and '111111.' This design exhibits the same error characteristics and uses the same hardware resources as the one shown in Table III; however, the number of outputs is reduced by one bit, which would simplify the subsequent circuits that process the outputs.

D. Reliability

Usually, MG is widely used in fault-tolerant architectures to improve the reliability of a system, e.g., a triple/N-tuple modular redundancy system [44]. Similarly, the reliability of a computation element constructed by using MGs can also be improved compared to using conventional logic gates [45]. In [45], the reliability of three 3-input MG-based and an XOR-based FAs are analyzed and simulated at the gate level. Overall, the MG-based FAs are more reliable than the XOR-based FA with a same gate error probability ε. For an MG with a larger number

TABLE IV: The transformed truth table for the proposed approximate 6-input compressor design with two carry outputs.

N_{in}	Sum→Sum′	C0→ C0′	C1
0	0→1 ✘	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	1	1
5	1	1	1
6	0→1 ✘	1	1

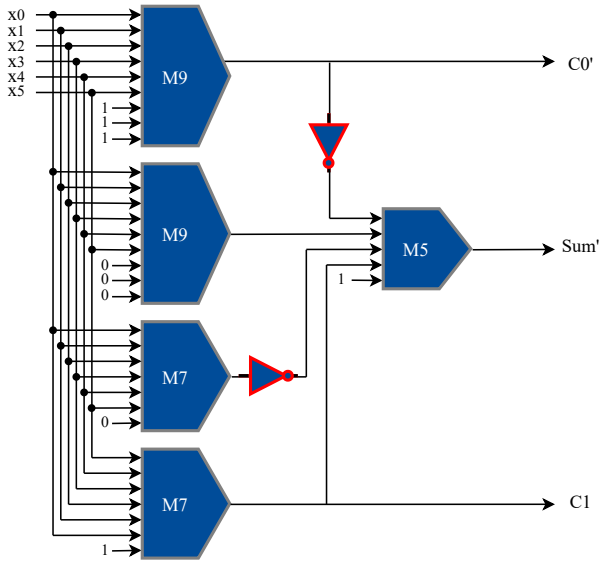


Fig. 12: The approximate 6-input compressor design 3.

of inputs such as 9-input MG, it generates a less reliable output than 3-input MGs to implement the same function when the ε is small [44]. With the increase of the ε, the output reliability of a 9-input MG-based circuit becomes higher than that based on the 3-input MG. However, not only the gate error probability but also the reliability of the inputs, wires and crossovers, as well as the number of utilized gates and their connection ways affect the output reliability of a circuit. A detailed discussion on the reliability of the scalable MG-based computation elements is not within the scope of this work.

V. Evaluation

A. Evaluation Framework

To evaluate the circuit characteristics of the proposed MG and compressor designs, a comprehensive simulation framework is developed that involves device, circuit and application level simulations. At the device level, the DW motion dynamics is benchmarked with experimental data from [29] using OOMMF [30]. The MTJ is modeled in Verilog-A and calibrated with the experimental data in [46]. Here, the electron transport and magnetization dynamics of the MTJ is modeled by using the non-equilibrium Green's function (NEGF) and the Landau-Lifshitz-Gilbert (LLG) equation, respectively. For the circuit level simulation, a Verilog-A model of a 3T-spin-TD is developed by using the device-level models to co-simulate with the interface CMOS circuits in Cadence Spectre and SPICE. In SPICE, the 45 nm North Carolina State University product development kit (PDK) library [47] is used to verify the proposed designs and acquire the circuit measurements of delay and power dissipation. Finally, a widely-used image compression algorithm for discrete cosine transform (DCT) and inverse DCT (IDCT), is implemented using approximate compressors to show their accuracy at the application level.

The simulation results at the device level have been analyzed in [16]. In this paper, the evaluation is focused on the circuit and application levels. The circuit level simulation results for the scalable MG and compressor designs are discussed in Sections V-B and V-C, respectively. Section VI simulates the compressor designs at the application level.

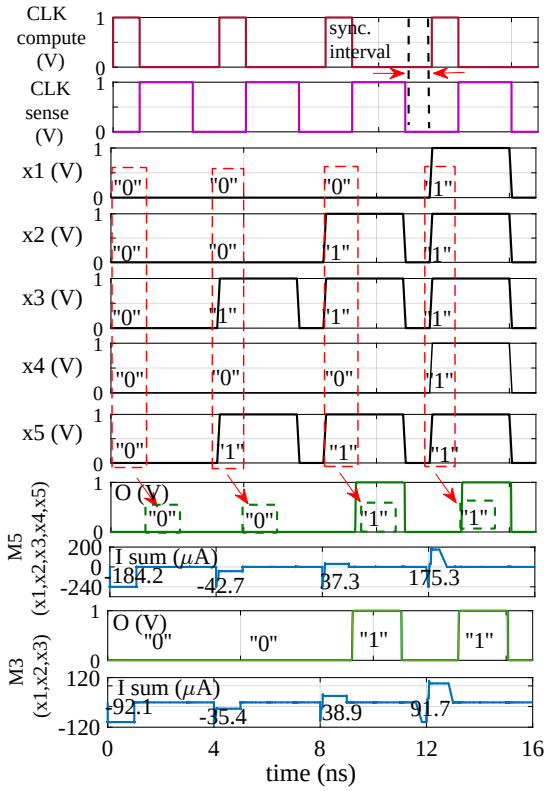


Fig. 13: Transient analysis of the MG with 3 and 5 inputs.

B. Evaluation of Majority Gates

Fig. 13 presents the waveforms of a transient analysis for the scalable MGs with 3 and 5 inputs. In this simulation, the full computation period for an MG is considered as 3 ns. A pulse width of 1 ns is used for $CLK_{compute}$ to complete the computation; a 2 ns pulse (CLK_{sense}) signal is employed for a proper implementation of sensing. A 1 ns synchronization interval is considered after sensing to make the circuit synchronizable in connection with other circuits.

During the computation phase when $CLK_{compute}$= "1," the input voltages are applied to the scalable CMOS input network for 1 ns leading to the generation of I_{sum} as a trigger for the DW. The transient current analysis of I_{sum} is plotted in Fig. 13. The purpose of this simulation is to observe the current alternations in the intersection point affected by the inputs. Considering M_3 as an example, for the input combinations of '000,' '001,' '011,' and '111,' four distinct I_{sum} (-92.1, -35.4, 38.9, 91.7) uA are obtained, which are consistent with the discussion in Section III. Thus, the DW position can be determined after this phase. During the sense phase, CLK_{sense} is activated for 2 ns to read the output voltage (V_O); the corresponding logic values are "0," "0," "1" and "1" for the 3-input MG.

Table V shows the power consumption and delay of the proposed MGs with various inputs. Clearly, the power consumption of the MG increases with the number of inputs. Power must be supplied to keep data in CMOS-based circuit, whereas it can be cut-off in the non-volatile designs with near-zero static power. Due to the non-volatility, the MGs can achieve higher speed and throughput using pipeline techniques without any additional clock control circuit. A fully pipelined design can be realized by alternately applying two clock signals on neighboring stages. Hence, the throughput of the proposed MGs can be considerably increased to one output set per 1 ns, which results in an equivalent 1 ns delay. A larger current injection to the MGs could lead to a higher computation speed, but it also leads to a higher power consumption. The device count can offer a fairly accurate estimation of the area since the proposed hybrid design is more compact than a CMOS implementation [15]. Furthermore, an embedded buffer can be presumed for spin-TDs due to their non-volatility characteristic; however, such a buffer should be inserted after every other logic gates working at different operational phases in a CMOS design.

TABLE V: Circuit measurements of the proposed MGs.

MG Designs	Device count §	Power (uW) †	Delay (ns) ‡
3-input	9T+2M+1D	22.1	3
5-input	13T+2M+1D	27.5	3
7-input	17T+2M+1D	32.5	3
9-input	21T+2M+1D	39.4	3

§ T: MOS Transistor; M: MTJ; D: DWS. † Total power including dynamic and static powers. ‡ Total gate delay not considering the pipeline.

C. Evaluation of Compressor Designs

With promising features, spintronic devices have widely been investigated as an innovative memory candidate; relatively less work has been done for logic design. However, several non-volatile accurate and approximate FAs have been proposed using MTJ and DWS recently [12], [13], [15], [16]. Based on the logic-in-memory architecture, a nonvolatile FA is fabricated using 4 MTJs and 34 transistors [15]. In this design, an input stored in the MTJ is added with the two external inputs by using a regular CMOS logic tree; the outputs are then obtained by the read operation using the PCSA. Similarly, [13] presents an accurate and two approximate FAs consisting of MTJs and transistors. The two approximate FAs are designed by removing some transistors in the logic tree and reducing the supply voltage of the accurate one, respectively. In [12] and [16], FAs are implemented using the current-mode MGs based on the structure shown in Fig. 4. The difference occurs in the sense circuit, i.e., the PCSA is used to sense the output voltage of the FA in [12], whereas the sense circuit of the design in [16] is implemented by a resistive voltage divider. Following a traditional design approach, two approximate 4-2 compressors have also been devised in [16] by using non-volatile accurate and approximate FAs, referred to as Design I and Design II. To show the efficiency of the proposed designs, the accurate and approximate 4-2 compressors consisting of FAs are compared. The designs in [13] are not considered in the comparison because their accuracy is very low compared with the other designs.

Table VI shows the accuracy and circuit characteristics of accurate and approximate 4-2 compressor designs. In this table, the bias is calculated as the average value of all possible errors; the mean error distance (MED) is the mean value of all absolute errors. Table VI shows that the MEDs of the approximate compressors have the same values as their ERs; this is because the error magnitudes of these designs are always 1. Note that the device count here does not show a strict comparison of the considered designs; however, it reveals a rough comparison tendency in circuit area. Thus, the proposed designs can result in smaller areas than the other designs except for Design I [16]. When a same MG is used, the proposed MG-EC shows a shorter critical path than the other accurate 4-2 compressors; MG-AC1 and MG-AC2 have shorter critical paths than the other approximate designs. Also, the proposed designs show lower ERs and smaller biases and MEDs for $P(0) = 0.75$ than the other approximate 4-2 compressors; so they are significantly more accurate. When $P(0) = 0.5$, the biases

TABLE VI: Comparison of accurate and approximate 4-2 compressor designs.

Design	Device count [§]	Critical path	$P(0)=0.5$			$P(0)=0.75$		
			ER (%)	Bias (%)	MED (%)	ER (%)	Bias (%)	MED (%)
MTJ [15] [*]	68T+8M	NA	0	0	0	0	0	0
HSM DWM [12] [†]	46T+8M+4D	4MG+2INV	0	0	0	0	0	0
LPM DWM [12] [†]	46T+8M+4D	4MG+2INV	0	0	0	0	0	0
DWM [16] [‡]	44T+8M+4D	4MG+2INV	0	0	0	0	0	0
Design I [16] [¶]	22T+4M+2D	2MG+2INV	37.5	0	37.5	50.8	44.1	50.8
Design II [16] [¶]	33T+6M+3D	3MG+2INV	25.0	0	25.0	34.4	28.9	34.4
MG-EC	33T+6M+3D	3MG+2INV	0	0	0	0	0	0
MG-AC1	26T+4M+2D	2MG+2INV	6.25	0	6.25	23.8	23.6	23.8
MG-AC2	28T+4M+2D	2MG+1INV	6.25	-6.25	6.25	0.39	-0.39	0.39

[*] Accurate compressor consisting of two MTJ-based FAs in [15]. [†] HSM DWM and LPM DWM are the accurate 4-2 compressors consisting of DWM-based high-speed and low-power mode FAs in [12]. [‡] Accurate compressor implemented by two cascaded spin-TD based FAs in [16]. [¶] Design I is the approximate 4-2 compressor constructed by two approximate FAs. Design II is the approximate 4-2 compressor consisting of an accurate and an approximate FAs. [§] T: MOS Transistor; M: MTJ; D: DWS.

are zero for MG-AC1, Design I [16] and Design II [16]. The MEDs of the proposed compressors are smaller than those of the other approximate designs for both $P(0)=0.5$ and $P(0)=0.75$.

Table VII shows the power and delay measurements of the compressors, where two CMOS-based designs [15], [48] are also considered. In the design of CMOS [15], two accurate FAs designed for a low static power consumption are utilized. The FA is a classic mirror adder with a dynamic current mode logic under the control of a clock to cut off the static current flow from the supply voltage to the ground [49]; thus, 42 transistors are used. Therefore, 84 transistors are required for a 4-2 compressor. The compressor denoted as CMOS [48] is implemented by using the newly designed XOR-XNOR module that enables an ultra low supply voltage, resulting in a lower power dissipation and higher performance than a conventional design. CMOS [48] consists of 72 transistors.

Table VII reveals that the static power dissipation of the MG-based 4-2 compressors is zero, so they are effective in normally-off systems for low-power operations. CMOS [15] also shows a very low static power due to the use of the dynamic current mode logic. In general, the CMOS-based 4-2 compressors show smaller delay but larger static power than the non-volatile designs; however, as larger numbers of MOS transistors are required for the CMOS-based designs, they tend to have larger areas. Among the accurate non-volatile designs, MG-EC is the most efficient with the lowest power dissipation and a relatively small delay. MTJ [15] consumes the highest power with the largest delay because of its complex logic-in-memory function. The accurate compressor consisting of high-speed mode (HSM) DWM-based FAs shows the highest speed but consumes a significantly large dynamic power and energy, whereas the one consisting of low-power mode (LPM) DWM-based FAs dissipates a lower power with a larger delay. The dynamic power consumed by DWM [16] and Design II [16] is quite high. With a higher accuracy, MG-AC1 and MG-AC2 are significantly more power and energy efficient than the other approximate designs. To give an idea of how read and write operations contribute to the delay and dynamic power of the presented 4-2 compressor designs, we measured them separately in the form of (write, read). In terms of the dynamic power consumption, the reported results for MG-EC, MG-AC1, and MG-AC2 are (58.62, 14.78) uW, (32.33, 11.96) uW, and (30.62, 9.67) uW, respectively. As for delay, (2, 2) ns, (1, 2) ns, and (1, 2) ns are achieved, respectively. As per these measurements, the proposed spin-based designs are more energy-efficient in read- than in write-sensitive applications.

Fig. 14 shows a comprehensive comparison of non-volatile 4-2 compressors considering both error and circuit measurements. The MTJ [15] and HSM DWM [12] are not included in the figure due to their very large power and energy consumption. For a more straightforward comparison, the absolute values for the bias are considered and the values for each metric are normalized by its maximum absolute value. As the MED and ER of each design share the same value, only the ER is shown in this comparison. Fig. 14 shows that MG-AC2 reaches the best trade off between accuracy and hardware overhead, except that it has the largest bias for $P(0)=0.5$. Thus, MG-AC2 is well-suited for the partial product accumulation in a multiplier. With small values of error and circuit measurements, MG-AC1 performs well for $P(0)=0.5$. MG-EC is the most energy efficient accurate design.

In summary, the proposed approximate 4-2 compressors outperform the other designs in terms of accuracy and/or hardware overhead. These advantages are obtained due to the use of the

TABLE VII: Power and delay measurements of accurate and approximate 4-2 compressor designs.

Designs	Type	Dynamic power (uW)	Static power (nW)	Delay (ns)	Energy (pJ)
CMOS [15]	accurate	65.4	0.22	0.22	0.015
CMOS [48]	accurate	39.4	2.5	0.13	0.005
MTJ [15]	accurate	4,200	0.00	20.4	85.7
HSM DWM [12]	accurate	2,728	0.00	2.54	6.93
LPM DWM [12]	accurate	170	0.00	3.70	0.63
DWM [16]	accurate	99.2	0.00	9.00	0.89
Design I [16]	approximate	57.8	0.00	3.00	0.17
Design II [16]	approximate	84.5	0.00	4.00	0.34
MG-EC	accurate	73.4	0.00	4.00	0.29
MG-AC1	approximate	44.3	0.00	3.00	0.13
MG-AC2	approximate	40.3	0.00	3.00	0.12

The Delay includes the time consumed by compute and sense operations leveraging pipeline technique in [16].

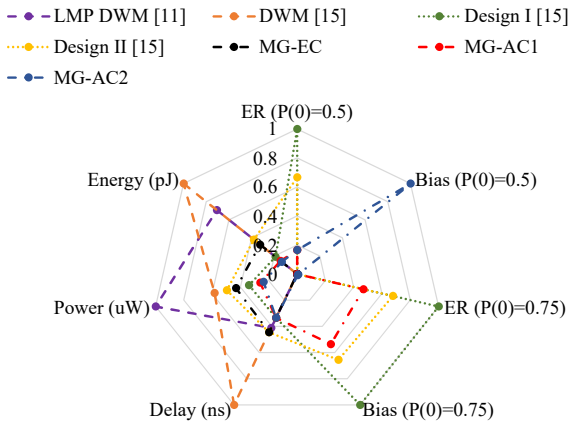


Fig. 14: A comprehensive comparison of 4-2 compressors.

novel design principles. Specifically, an equivalent transformation and approximation is performed on the truth table of an accurate compressor to avoid the use of the conventional design method by cascading several FAs. This reduces the critical path and prevents the error propagation between FAs. Moreover, the use of the HMIGs and MGs with five or more inputs simplifies the circuit of a compressor.

VI. Image Processing Application

For an application level evaluation, the approximate 4-2 compressors are utilized to implement the DCT and IDCT operations in an image compression algorithm. The basic computation in DCT-IDCT is 8×8 matrix multiplication. In the simulation, the inputs are in 16-bit 2's complement format, so 16×16 signed multipliers and 32-bit accumulators are used. The approximate compressors are utilized in the multipliers and accumulators to show their accuracy regarding inputs with $P(0) = 0.75$ and $P(0) = 0.5$, respectively.

In the multiplier, some least significant columns of the partial products (LSPPs) are accumulated by approximate 4-2 compressors, while the remaining most significant columns are accurately processed. Two Dadda tree-based partial product accumulation structures have been constructed by using approximate 4-2 compressors in [20], resulting in two approximate multipliers. The Dadda tree structure of Multiplier 1 in [20] is utilized here for the partial product accumulation using MG-AC1 with 5 inputs. For MG-AC2 with 4 inputs, the Dadda tree of Multiplier 2 in [20] is used. Table VIII shows the error characteristics of the 16×16 signed approximate multipliers, where the numbers in the header are the numbers of LSPPs accumulated by the corresponding compressor designs. The NMED is the MED normalized by the maximum output of an accurate design. The MRED is the mean of the absolute relative error distance calculated by $\left|\frac{\text{ED}}{M}\right|$, where $\text{ED} = |M' - M|$, and M' and M are the approximate and accurate results, respectively. As per Table VIII, the MG-AC2-based multiplier achieves significantly smaller errors than the other designs. With smaller values of ER, NMED and MRED, the multipliers using MG-AC1 and MG-AC2 are more accurate than those using the other designs.

Table IX shows the peak signal-to-noise ratios (PSNRs) and structural similarity index measures (SSIMs) of the DCT-IDCT results using approximate compressors in the multipliers. Also, the resultant images are shown in Fig. 15, where the numbers following the design names are the numbers of LSPPs for the multipliers using the corresponding compressors. Table IX shows that using the proposed approximate 4-2 compressors produces more accurate results with higher PSNRs and SSIMs than the other designs for a same value of LSPPs. Among the DCT-IDCT results, the ones processed by the multipliers using MG-AC2 show the highest values of PSNR and SSIM. To achieve a nearly accurate result, the maximum number of LSPPs for Design I [16], Design II [16] and MG-AC1 is 12, whereas it is 15 for MG-AC2. This indicates that more LSPPs can be approximated by using MG-AC2 than the other approximate compressors, due to its significantly lower ERs and error biases for the inputs with $P(0) = 0.75$.

In addition, the delay and energy reductions due to the use of approximate compressors in 16×16 signed multipliers compared to an accurate DWM [16]-based multiplier are shown in Fig. 16. Note that the current generation and replication circuit

TABLE VIII: Error characteristics of the 16×16 signed multipliers using approximate compressors.

Design	Metric	12	13	14	15	16
Design I [16]	ER (%)	99.97	99.98	99.99	100.0	100.0
	NMED (10^{-5})	0.55	1.09	2.40	4.68	10.2
	MRED (10^{-3})	0.66	1.30	3.00	6.20	13.8
Design II [16]	ER (%)	99.86	99.94	99.98	99.99	99.99
	NMED (10^{-5})	0.46	0.91	1.91	3.96	7.82
	MRED (10^{-3})	0.57	1.20	2.80	5.60	12.2
MG-AC1	ER (%)	96.10	97.34	98.30	98.84	99.24
	NMED (10^{-5})	0.33	0.70	1.70	3.47	6.71
	MRED (10^{-3})	0.40	0.94	2.40	4.90	11.0
MG-AC2	ER (%)	13.70	17.09	21.78	26.47	34.86
	NMED (10^{-5})	0.02	0.04	0.10	0.24	0.70
	MRED (10^{-3})	0.04	0.09	0.38	0.50	1.60

The numbers in the header row are the numbers of the least significant columns of partial products in the multiplier accumulated by using the corresponding approximate compressors in the first column.

TABLE IX: PSNRs (dB) and SSIMs of the DCT-IDCT results by using approximate multipliers.

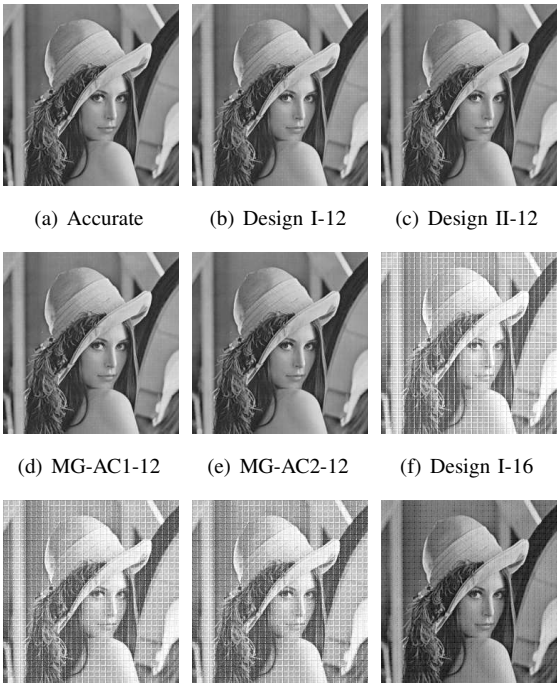
Design	Metric	12	13	14	15	16
Design I [16]	PSNR	34.45	27.94	21.42	16.56	11.94
	SSIM	0.912	0.751	0.500	0.329	0.203
Design II [16]	PSNR	35.00	28.56	21.43	16.20	13.02
	SSIM	0.920	0.773	0.5114	0.320	0.232
MG-AC1	PSNR	37.31	30.03	22.13	16.68	13.25
	SSIM	0.948	0.814	0.543	0.336	0.237
MG-AC2	PSNR	49.58	46.55	40.83	33.02	27.43
	SSIM	0.996	0.992	0.976	0.893	0.762

The numbers in the header row are the numbers of the LSPPs in the multiplier accumulated by using the corresponding approximate compressors in the first column.

presented in Fig. 9 is used to efficiently implement the proposed compressors and multipliers. The most significant bits of the partial products in the approximate multipliers are accumulated by the MG-EC. The results show that the accurate multiplier using MG-EC reduces the energy consumption and delay by more than 30 ns and 45 uJ respectively, compared to that using DWM [16]. With descending PSNRs in the figure, the PSNRs of the resultant images can be approximately divided into four levels, i.e., around 50 dB (high), around 35 dB (medium), around 30 dB (medium-low), and around 13 dB (low). Among the implementations with high PSNRs, the one using MG-AC2-13 achieves the highest delay and energy reductions; MG-AC1-12 results in more hardware improvements in the cases with medium PSNRs. The implementation using MG-AC2-16 saves the most time and energy in the designs with medium-low PSNRs.

In the matrix multiplications of DCT-IDCT, accumulators are required to sum the final products of 16×16 multipliers. Although not as frequently used as in a multiplier, compressors can be utilized to implement an accumulator with a high speed. Here, approximate compressors are used in the accumulators to further assess their accuracy for the case that each input is 50% to be "0" ($P(0) = 0.5$). Similar to multipliers, some least significant columns of the inputs (LSIs) are accumulated by the approximate 4-2 compressors. The PSNRs and SSIMs of the DCT-IDCT results are reported in Table X.

With the highest values of PSNR and SSIM, the images processed by MG-AC1 show the highest quality among the considered designs for a same number of LSIs. MG-AC2 also shows a relatively high accuracy than Design I [16] and Design II [16] in the accumulator. These are consistent with the ER results for $P(0) = 0.5$, as shown in Table VI. With the same ER for $P(0) = 0.5$, MG-AC2 shows lower PSNRs and SSIMs than MG-AC1 due to its higher error bias. The error generated



(a) Accurate (b) Design I-12 (c) Design II-12

(d) MG-AC1-12 (e) MG-AC2-12 (f) Design I-16

(g) Design II-16 (h) MG-AC1-16 (i) MG-AC2-16

Fig. 15: DCT-IDCT results using approximate multipliers based on different 4-2 compressors.

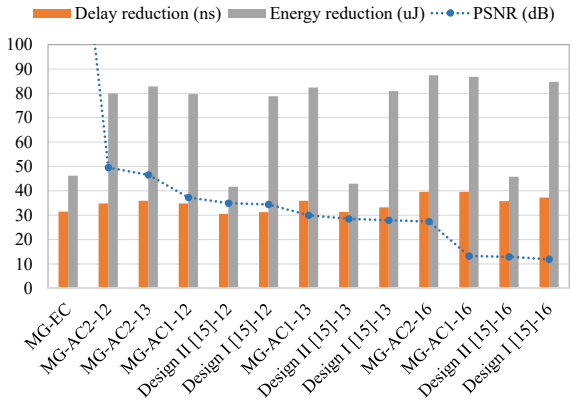


Fig. 16: A comparison of DCT-IDCT implementations using 4-2 compressors in the multipliers.

by MG-AC2 is single-sided (i.e., -1), while both -1 (for input '11111') and +1 (for input '00000') errors are possible for MG-AC1; thus, errors can be partially compensated by using MG-AC1. To achieve a close accuracy to the accurate design, the maximum numbers of approximate LSIs in the accumulators are 17, 17, 18 and 17 for using Design I, Design II, MG-

TABLE X: PSNRs (dB) and SSIMs of the DCT-IDCT results by using approximate accumulators.

Design	Metric	16	17	18	19	20
Design I [16]	PSNR	40.35	34.51	28.26	22.03	15.92
	SSIM	0.963	0.875	0.677	0.429	0.221
Design II [16]	PSNR	41.94	36.31	30.11	23.81	17.53
	SSIM	0.973	0.911	0.743	0.495	0.267
MG-AC1	PSNR	45.19	39.65	33.08	26.41	19.83
	SSIM	0.986	0.951	0.837	0.622	0.382
MG-AC2	PSNR	42.33	36.20	29.31	22.67	16.56
	SSIM	0.984	0.939	0.796	0.557	0.330

The numbers in the header row are the numbers of the least significant columns of inputs accumulated by using the corresponding approximate compressors in the first column.

AC1 and MG-AC2, respectively. Using approximate compressors in accumulators, the advantage of the MG-AC2 over the other designs is less significant than the case when they are used in multipliers. This is because different input distributions lead to different error characteristics (e.g., ERs and error biases) for the considered approximate 4-2 compressors.

VII. Conclusion

In this paper, highly-scalable MGs using spin-TD are introduced to enable the design of non-volatile arithmetic circuits. To synthesize the MG-based compressor circuits, the HMIG is proposed; the transformation rules for $(2m+1)$-input MGs are further presented for the simplification of HMIGs. An accurate and two approximate non-volatile 4-2 compressors are then designed with near-zero static power, low dynamic power and high performance. Moreover, the proposed methodology is generalized for designing approximate compressors with a larger number of inputs. The circuit simulation results show that the MG-EC are more hardware efficient than the existing accurate 4-2 compressors; MG-ACs achieve significant reductions in power dissipation, delay and energy consumption with smaller errors, compared to existing approximate designs. MG-AC2 performs better in the partial product accumulation of multipliers than in the accumulator, whereas MG-AC1 are more suitable for implementing an accumulator. To reach a similar image quality in DCT-IDCT (measured in PSNR and SSIM), MG-ACs achieve the largest reductions in delay and energy compared with the other approximate designs.

References

[1] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.

[2] B. H. Calhoun, F. A. Honore, and A. P. Chandrakasan, "A leakage reduction methodology for distributed MTCMOS," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 5, pp. 818–826, 2004.

[3] K. Shi and D. Howard, "Challenges in sleep transistor design and implementation in low-power designs," in *DAC*, 2006, pp. 113–116.

[4] Y. Gang, W. Zhao, J.-O. Klein, C. Chappert, and P. Mazoyer, "A high-reliability, low-power magnetic full adder," *IEEE Transactions on Magnetics*, vol. 47, no. 11, pp. 4611–4616, 2011.

[5] T. Nakada, T. Shimizu, and H. Nakamura, "Normally-off computing for iot systems," in *ISOCC*, 2015, pp. 147–148.

[6] M. Hayashikoshi, Y. Sato, H. Ueki, H. Kawai, and T. Shimizu, "Normally-off MCU architecture for low-power sensor node," in *ASP-DAC*, 2014, pp. 12–16.

[7] A. Roohi and R. F. DeMara, "NV-clustering: Normally-off computing using non-volatile datapaths," *IEEE Transactions on Computers*, vol. 67, no. 7, pp. 949–959, 2018.

[8] T. Hanyu, T. Endoh, D. Suzuki, H. Koike, Y. Ma, N. Onizawa, M. Natsui, S. Ikeda, and H. Ohno, "Standby-power-free integrated circuits using MTJ-based VLSI computing," *Proceedings of the IEEE*, vol. 104, no. 10, pp. 1844–1863, 2016.

[9] K. Ma, Y. Zheng, S. Li, K. Swaminathan, X. Li, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan, "Architecture exploration for ambient energy harvesting nonvolatile processors," in *HPCA*, 2015, pp. 526–537.

[10] E. Deng, Y. Wang, Z. Wang, J.-O. Klein, B. Dieny, G. Prenat, and W. Zhao, "Robust magnetic full-adder with voltage sensing 2T/2MTJ cell," in *NANOARCH*, 2015, pp. 27–32.

[11] H. Cai, Y. Wang, L. A. Naviner, Z. Wang, and W. Zhao, "Approximate computing in MOS/spintronic non-volatile full-adder," in *NANOARCH*, 2016, pp. 203–208.

[12] A. Roohi, R. Zand, and R. F. DeMara, "A tunable majority gate-based full adder using current-induced domain wall nanomagnets," *IEEE Transactions on Magnetics*, vol. 52, no. 8, pp. 1–7, 2016.

[13] H. Cai, Y. Wang, L. A. D. B. Naviner, and W. Zhao, "Robust ultra-low power non-volatile logic-in-memory circuits in FD-SOI technology," *IEEE Transactions on Circuits & Systems I Regular Papers*, vol. 64, no. 4, pp. 847–857, 2017.

[14] A. Roohi, R. Zand, D. Fan, and R. F. DeMara, "Voltage-based concatenatable full adder using spin hall effect switching," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[15] S. Matsunaga, J. Hayakawa, S. Ikeda, K. Miura, H. Hasegawa, T. Endoh, H. Ohno, and T. Hanyu, "Fabrication of a nonvolatile full adder based on logic-in-memory architecture using magnetic tunnel junctions," *Applied Physics Express*, vol. 1, no. 9, p. 091301, 2008.

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

[17] E. Deng, Y. Zhang, W. Kang, B. Dieny, J.-O. Klein, G. Prenat, and W. Zhao, "Synchronous 8-bit non-volatile full-adder based on spin transfer torque magnetic tunnel junction," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 7, pp. 1757–1765, 2015.

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

[19] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proceedings of the IEEE*, 2020.

[20] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015.

[21] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4: 2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1352–1361, 2017.

[22] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, no. 99, pp. 1–14, 2018.

[23] M. S. Ansari, H. Jiang, B. F. Cockburn, and J. Han, "Low-power approximate multipliers using encoded partial products and approximate compressors," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 404–416, 2018.

[24] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 11, pp. 4200–4208, 2019.

[25] X. Fong, Y. Kim, K. Yogendra, D. Fan, A. Sengupta, A. Raghunathan, and K. Roy, "Spin-transfer torque devices for logic and memory: Prospects and perspectives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 1–22, 2016.

[26] S. Gu, E. H.-M. Sha, Q. Zhuge, Y. Chen, and J. Hu, "Area and performance co-optimization for domain wall memory in application-specific embedded systems," in *DAC*, 2015, p. 20.

[27] W. Zhao, D. Ravelosona, J. Klein, and C. Chappert, "Domain wall shift register-based reconfigurable logic," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 2966–2969, 2011.

[28] S. Fukami, M. Yamanouchi, S. Ikeda, and H. Ohno, "Depinning probability of a magnetic domain wall in nanowires by spin-polarized currents," *Nature communications*, vol. 4, no. 1, pp. 1–7, 2013.

[29] S. Fukami, M. Yamanouchi, K.-J. Kim, T. Suzuki, N. Sakimura, D. Chiba, S. Ikeda, T. Sugibayashi, N. Kasai, T. Ono *et al.*, "20-nm magnetic domain wall motion memory with ultralow-power operation," in *IEDM*, 2013, pp. 3–5.

[30] M. Donahue and D. Porter, "OOMMF user's guide, version 1.0," Interagency Report NISTIR 6376, National Institute of Standards and Technology, Gaithersburg, MD, Sept 1999.

[31] E. Martinez, "The stochastic nature of the domain wall motion along high perpendicular anisotropy strips with surface roughness," *Journal of Physics: Condensed Matter*, vol. 24, no. 2, p. 024206, 2011.

[32] S. Angizi, Z. He, N. Bagherzadeh, and D. Fan, "Design and evaluation of a spintronic in-memory processing platform for non-volatile data encryption," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[33] J. Kan, C. Park, C. Ching, J. Ahn, L. Xue, R. Wang, A. Kontos, S. Liang, M. Bangar, H. Chen *et al.*, "Systematic validation of 2x nm diameter perpendicular mtj arrays and mgo barrier for sub-10 nm embedded stt-mram with practically unlimited endurance," in *IEDM*, 2016, pp. 27–4.

[34] S. Tehrani, "Status and prospect for mram technology," in *HCS*, 2010, pp. 1–23.

[35] C.-W. Hsu, I.-T. Wang, C.-L. Lo, M.-C. Chiang, W.-Y. Jang, C.-H. Lin, and T.-H. Hou, "Self-rectifying bipolar tao x/tio 2 rram with superior endurance over 10 12 cycles for 3d high-density storage-class memory," in *Symposium on VLSI Technology*, 2013, pp. T166–T167.

[36] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali, "Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling," in *MICRO*, 2009, pp. 14–23.

[37] (2011) Eeverspin. Accessed August 7, 2017. [Online]. Available: https://www.everspin.com/mram-technology-attributes

[38] M. D. Ercegovac and T. Lang, *Digital Arithmetic*. Elsevier Science & Technology, 2003.

[39] S. B. Mamaghani, M. H. Moaiyeri, and G. Jaberipur, "Design of an efficient fully nonvolatile and radiation-hardened majority-based magnetic full adder using FinFET/MTJ," *Microelectronics Journal*, vol. 103, p. 104864, 2020.

[40] 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 Transactions on Emerging Topics in Computing*, 2019.

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

[42] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization," in *DAC*, 2014, pp. 1–6.

[43] G. S. Rose, J. Rajendran, H. Manem, R. Karri, and R. E. Pino, "Leveraging memristive systems in the construction of digital logic circuits," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 2033–2049, 2011.

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

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

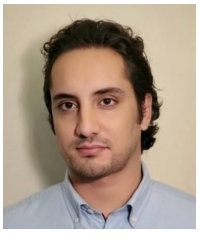
[46] X. Fong, S. K. Gupta, N. N. Mojumder, S. H. Choday, C. Augustine, and K. Roy, "KNACK: A hybrid spin-charge mixed-mode simulator for evaluating different genres of spin-transfer torque mram bit-cells," in *SISPAD*, 2011, pp. 51–54.

[47] (2011) NCSU EDA FreePDK45. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45:Contents

[48] C.-H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 10, pp. 1985–1997, 2004.

[49] M. W. Allam and M. I. Elmasry, "Dynamic current mode logic (DyCML): A new low-power high-performance logic style," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 550–558, 2001.

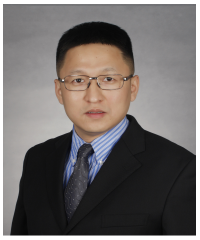
Honglan Jiang (S'14-M'18) received the B.Sc. and Master degrees in instrument science and technology from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in 2011 and 2013, respectively. In 2018, she received the Ph.D. degree in integrated circuits and systems from the University of Alberta, Edmonton, AB, Canada. She is currently a postdoctoral fellow in the Institute of Microelectronics, Tsinghua University, Beijing, China. Her research interests include approximate computing, reconfigurable computing and stochastic computing.



Shaahin Angizi (S'15) is currently a Ph.D. candidate in Electrical Engineering at the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. His primary research interests include ultra-low power in-memory computing based on volatile & non-volatile memories, brain inspired (neuromorphic) computing, and accelerator design for big data applications such as deep neural network, bioinformatics, etc. He is the recipient of Best Presentation Award of Ph.D. Forum at IEEE/ACM DAC in 2018, two Best Paper Awards of IEEE ISVLSI in 2017 and 2018, and Best Paper Award of ACM GLSVLSI in 2019. He is a student member of IEEE.

Deliang Fan (M'15) is currently an Assistant Professor in the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. Before joining ASU in 2019, he was an assistant professor in Department of ECE at University of Central Florida, Orlando, FL, USA. He received his M.S. and Ph.D. degrees, under the supervision of Prof. Kaushik Roy, in ECE from Purdue University, West Lafayette, IN, USA, in 2012 and 2015, respectively. His primary research interests include Energy Efficient and High Performance Big Data Processing-In-Memory Circuit, Architecture and Algorithm, with applications in Deep Neural Network, Data Encryption, Graph Processing and Bioinformatics Acceleration-in-Memory system; Brain-inspired (Neuromorphic) and Boolean Computing Using Emerging Nanoscale Devices like Spintronics and Memristors; security of AI system. He has authored and co-authored 100+ peer-reviewed international journal/conference papers. He served as the TPC member of DAC, ICCAD, DATE, GLSVLSI, ISVLSI, ASP-DAC, ISQED, etc. He also served as the area technical chair of GLSVLSI, ISQED, etc.

Jie Han (S'02-M'05-SM'16) 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 in the Department of Electrical and Computer Engineering at the 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. Dr. Han was a recipient of the Best Paper Award at the International Symposium on Nanoscale Architectures (NanoArch) 2015 and Best Paper Nominations at the 25th Great Lakes Symposium on VLSI (GLSVLSI) 2015, NanoArch 2016 and the 19th International Symposium on Quality Electronic Design (ISQED) 2018. 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 (2005). He is currently an Associate Editor for the IEEE Transactions on Emerging Topics in Computing (TETC), the IEEE Transactions on Nanotechnology, the IEEE Circuits and Systems Magazine, the IEEE Open Journal of the Computer Society and Microelectronics Reliability (Elsevier Journal). He served as a General Chair for GLSVLSI 2017 and the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT) 2013, and a Technical Program Committee Chair for GLSVLSI 2016, DFT 2012 and the Symposium on Stochastic & Approximate Computing for Signal Processing and Machine Learning, 2017.



Leibo Liu (M'10-SM'17) received the B.S. degree in electronic engineering and the Ph.D. degree with the Institute of Microelectronics, both from Tsinghua University, Beijing, China, in 1999 and 2004, respectively. He is currently a Full Professor with the Institute of Microelectronics, Tsinghua University. His current research interests include reconfigurable computing, mobile computing, and very large-scale integration digital signal processing.