Post-P&R Performance and Power Analysis for RRAM-Based FPGAs

Xifan Tang, *Member, IEEE*, Edouard Giacomin, *Student Member, IEEE*,
Giovanni De Micheli, *Fellow, IEEE*, and Pierre-Emmanuel Gaillardon, *Senior Member, IEEE*

***Abstract*—*Resistive random access memory* (RRAM)-based FPGAs are predicted to outperform conventional FPGAs architectures in area, delay, and power over a wide range of voltage operations, allowing novel energy-quality tradeoffs for reconfigurable computing. The opportunity lies in that RRAMs can realize the functionality of a *static random access memory* (SRAM) and a transmission gate in a unique device. However, most of predictive analyses shown in the state of the art are achieved by using analytical models. Unfortunately, while analytical models have been intensively refined for conventional FPGA architectures, their accuracy on RRAM-based FPGAs has not been carefully investigated. Consequently, misleading conclusions may be caused by using inaccurate analytical models. In this paper, we rely on electrical simulations and semi-custom design tools to perform detailed area and power comparison between SRAM-based and RRAM-based FPGAs. To enable accurate analysis, we develop a synthesizable Verilog generator for both SRAM-based and RRAM-based FPGAs and also enhance FPGA-SPICE to support most recent advanced RRAM-based circuits and FPGA architectures. The area analyses are based on full-chip layouts of SRAM-based and RRAM-based FPGAs, which are produced by a semi-custom design flow. We consider a full FPGA fabric, including core logic, configuring peripherals, and I/Os, which is more realistic than analytical models. The power analysis is based on SPICE simulation results by considering the 20 largest MCNC benchmarks. Simulation results identify that the target R_{HRS} of RRAM-based FPGAs should be at least 20 $M\Omega$ to guarantee energy improvements over SRAM-based FPGAs. Experimental results present that at nominal working voltage, RRAM-based FPGAs can improve up to 8% in area, on average 22% in delay and on average 16% in power, respectively, as compared to SRAM-based counterparts. Compared with SRAM-based FPGAs working at nominal voltage, near-V_t RRAM-based FPGAs can outperform close to two times in *energy-delay product* without delay overhead. As a result, RRAM-based FPGAs are more capable of trading-off energy and quality than the SRAM-based counterparts.**

***Index Terms*—Programmable logic arrays; resistive ram; simulation; system modeling; integrated circuit reliability.**

I. Introduction

Resistive *Random Access Memory* (RRAM) technology opens the opportunity in advancing FPGA technologies by bringing non-volatility and performance enhancements [1]–[11]. Major works focus on proposing novel programmable switches with the objective of replacing a *Static Random Access Memory* (SRAM) and a transmission-gate with a unique RRAM device [3]–[11]. With lower resistance than transistors and also smaller parasitic capacitances, RRAMs can bring remarkable improvements on the delay and power to routing multiplexers. Previous works predicted that these proposed RRAM FPGAs can improve area by 7-15%, delay by 45-58% and power by 20-58%, when compared to SRAM-based counterparts [3]–[11]. In particular, RRAM-based FPGAs operating in the near-V_t regime have demonstrated potentials in achieving both high performance levels, similar to an SRAM-based FPGA at nominal voltage, and low power levels comparable to a regular SRAM-based FPGA running at near-V_t regime [10], [11]. With promising performance and energy consumption at both near and regular V_t regime, RRAM-based FPGAs can provide a better range of operating voltages than SRAM-based counterparts, opening an opportunity for applying *Dynamic Voltage Scaling* (DVS) to FPGA fabrics [12]. For instance, operating voltage and energy consumption of FPGA-based computing system can be reduced significantly without sacrificing the throughputs and the fidelity of results [13].

However, most of the performance improvements have been assessed so far by using analytical area, delay, and power models [3]–[7], [10]. Even though analytical models have been intensively refined for conventional FPGA architectures, it is still difficult to guarantee accurate estimations. For example, it is reported that the Minimum Transistor Width Area model used in VPR suffers an overall prediction error variation of 93%, as compared to layout area [14], [15]. Similarly, the analytical FPGA power models can overestimate the total power consumption by at least 24%, as compared to electrical simulation results [16]–[19]. In addition, since these analytical models are exclusively designed to SRAM-based FPGAs, they are not general enough to capture the characteristics of RRAM-based FPGAs and other FPGAs based on emerging technologies. Therefore, by considering only analytical models, misleading conclusions can be drawn on the comparison between SRAM-based and RRAM-based FPGAs.

Manuscript received December 13, 2017; revised April 3, 2018 and May 27, 2018; accepted June 11, 2018. Date of publication June 14, 2018; date of current version September 11, 2018. This work was supported in part by the Swiss National Science Foundation under Grant 200021-146600, in part by the United States-Israel Binational Science Foundation under Grant 2016016, and by the Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Agreement FA8650-18-2-7855. This paper was recommended by Guest Editor M. Alioto. *(Corresponding author: Xifan Tang.)*

X. Tang, E. Giacomin, and P.-E. Gaillardon are with the Laboratory for NanoIntegrated Systems, Electrical and Computer Engineering Department, The University of Utah, Salt Lake City, UT 84112 USA (e-mail: xifan.tang@utah.edu).

G. De Micheli is with the Integrated Systems Laboratory, School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2018.2847600

2156-3357 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Compared to previous works, the contributions of this paper are:

1) We provide a realistic study of the area and power characteristics of RRAM-based FPGAs using full-chip layouts and electrical simulations. To enable accurate area analysis, we have developed a synthesizable Verilog generator for both SRAM-based and RRAM-based FPGAs, with which layouts of full FPGA fabrics can be derived by employing a semi-custom design flow. To enable an accurate power analysis, we enhance FPGA-SPICE [19] to output SPICE netlists modeling 4T1R-based RRAM-based circuits and FPGA architectures [8], [9].
2) We consider full FPGA fabrics in area evaluations and especially analyze the impact of their configuration peripherals, which is neglected in most of previous works.
3) We carefully examine the impact of the *off*-resistance of RRAMs R_{HRS} on the energy consumption.
4) We study the robustness of the RRAM-based FPGAs against RRAM and CMOS corner variations regarding to performance and energy consumption.

As a result, we have been able to draw the following conclusions:

1) Considering a commercial 40nm technology, averaged over twenty biggest MCNC benchmarks, experimental results show that RRAM-based FPGAs can improve up to 8% in area, on average 22% in delay and on average 18% in power respectively, as compared to SRAM-based counterparts. The area efficiency of RRAM-based FPGAs increases when a large routing channel width is considered.
2) Electrical simulations identify that the target R_{HRS} for RRAM-based FPGAs should be larger than $20M\Omega$ to guarantee energy improvements over SRAM-based FPGAs. When operating at near-V_t regime, RRAM-based FPGAs can improve *Energy-Delay Product* by close to 2× without delay overhead, as compared to SRAM-based FPGA operating at nominal working voltage.
3) When considering a 30% process variations on RRAM resistances and different CMOS process corners in a commercial 40nm technology, full-chip-level simulations show that the performance and energy consumption of the RRAM-based FPGAs shift within 3% and 8% respectively, demonstrating stable improvements over SRAM-based FPGAs.

The rest of this paper is organized as follows. Section II gives background knowledge about RRAM technology, SRAM-based and RRAM-based FPGA architectures. Section III shows our vision on RRAM-based FPGA architectures. Section IV introduces CAD flow based on FPGA-SPICE. Section V presents and analyses experimental results. Section VII concludes this paper.

II. Background

In this section, we first give a brief introduction on RRAM technologies (Section II-A). We then discuss the full SRAM-based FPGA architecture (Section II-B) and the limitation of associated CAD tools (Section II-C). Last but not least, we review recent works on RRAM-based FPGA architectures (Section II-D).

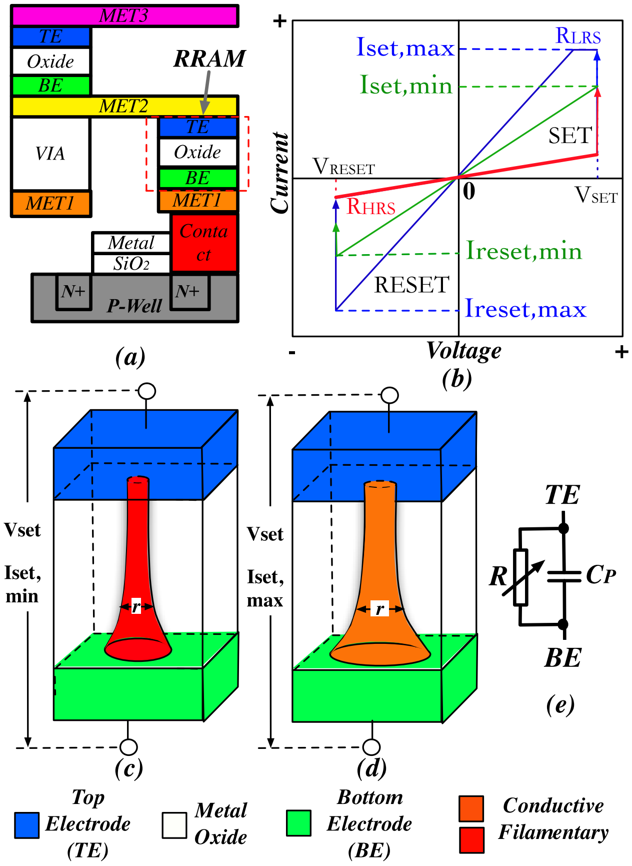

Fig. 1. (a) RRAM structure and BEoL integration; (b) RRAM I-V characterization; (c) RRAM structure with filaments controlled by $I_{set,min}$; (d) RRAM structure with filaments controlled by $I_{set,max}$; (e) Equivalent RC model.

A. RRAM Technology

Resistive Random Access Memories (RRAMs), a promising emerging memory technology [21], typically relies on a Metal-Oxide-Metal bitcell structure, which makes them compatible with *Back-End-of-the-Line* (BEoL) integration at a high density [22]. As depicted in Fig. 1(a), a RRAM can be freely fabricated anywhere between two metal layers on the top of transistors, e.g., between $MET1$ and $MET2$.

From a device perspective, a RRAM device typically consists of three layers: a *Top Electrode* (TE), a transition metal oxide material stack and a *Bottom Electrode* (BE), as highlighted in Fig. 1(a). Through a filamentary conduction mechanism in the metal oxide layer, RRAMs can be switched between two stable resistance states: the *High Resistance State* (HRS) and the *Low Resistance State* (LRS). The switching events between resistance states are triggered by applying a programming voltage across the TE and BE. The switching event from LRS to HRS is called **set** process, while the opposite one is called **reset** process. In terms of switching mechanisms, RRAMs can be broadly classified into two categories:

Bipolar Resistive Switching (BRS) and *Unipolar Resistive Switching* (URS). In this paper, we consider RRAM based on BRS only, which is a common choice for most RRAM-based circuits and systems [3]–[11]. Fig. 1(b) illustrates the I-V characteristics of a BRS RRAM. The minimum programming voltages required to trigger **set** and **reset** processes are defined as V_{set} and V_{reset}, respectively. The programming currents that are supplied during the set and reset processes are defined as I_{set} and I_{reset}, respectively. A current compliance on I_{set} is often enforced to avoid a permanent breakdown of the device, which is denoted by $I_{set,max}$ in Fig. 1(d). The programming current tunes the size of filaments, leading to a difference in the resistance of a RRAM in LRS, R_{LRS}. Take the examples in Fig. 1 (c) and (d), the filament highlighted in orange leads to a lower R_{LRS} than the filament highlighted in red. Fig. 1(e) depicts the equivalent RC model of a RRAM. Besides the configurable resistance R, a parasitic capacitance C_P induced by TE and BE should also be considered.

Thanks to BEoL compatibility and filament-based switching mechanism, the resistance and physical location of RRAMs can be tuned for different application demands, leading to a large design space for RRAM-based circuits and systems [1]–[11]. On the other side, the filamentary conduction property brings to RRAMs not only device-to-device variation but also cycle-to-cycle variability. Both device-to-device and cycle-to-cycle variations are reported to be well controlled between 10%-20% [23]–[25]. To be more robust in cycle-to-cycle variations, we can introduce program-verify strategy in programming RRAMs, similar to that of Flash memory [26]–[28]. More details about RRAM technology can be found in [22].

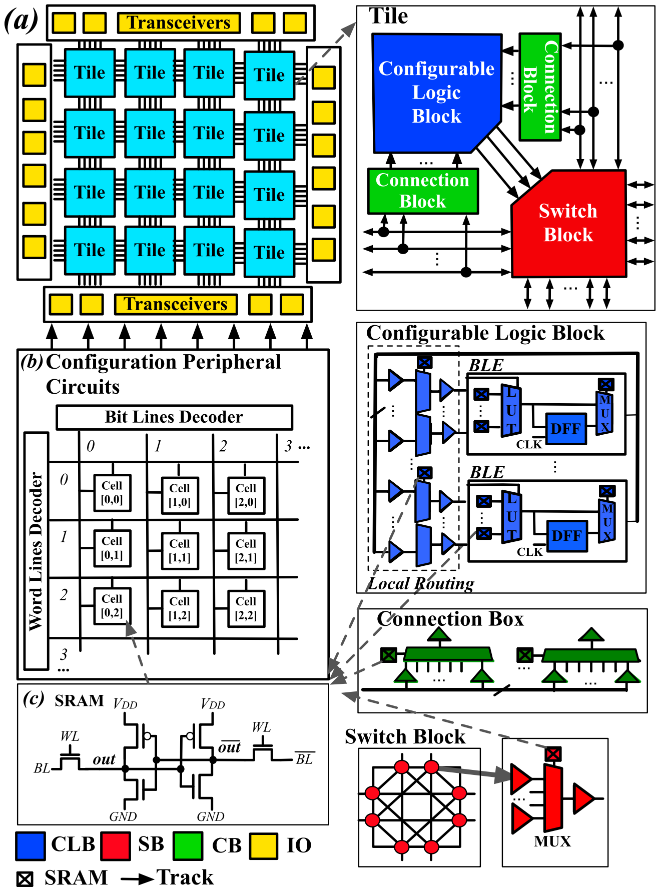

Fig. 2. Detailed full FPGA architecture: (a) Core logic and (b) Configuration peripheral circuits.

B. General FPGA Architecture

As shown in Fig. 2, a full FPGA fabric consists of two parts:

(a) The **core logic** is the hardcore of a FPGA that realizes logic functions. It consists of an array of tiles surrounded by IO blocks, as shown in Fig. 2(a). Each tile contains a *Configurable Logic Block* (CLB), a *Connection Block* (CB) and a *Switch Block* (SB) [29]. A CLB consists of *N Basic Logic Elements* (BLEs) and a local routing architecture providing inner-block interconnections. A BLE contains a *Look-Up Table* (LUT), a *Flip-Flop* (FF) and a 2:1 multiplexer, which selects either a combinational or a sequential output. SBs interconnect routing tracks between tiles, while CBs connect routing tracks to CLB input and output pins inside a tile. To accelerate arithmetic-intensive applications, commercial FPGAs [33]–[35] adopt various architectural enhancements, such as fracturable LUTs [32], hard carry chains and heterogeneous blocks. As we aim at capturing the difference between SRAM-based and RRAM-based FPGAs, we consider, without the loss of generality, the homogenous tile-based FPGA architecture shown in Fig. 2(a) in this paper.

(b) The **configuration peripheral circuits** aims at programming each SRAMs of LUTs and routing multiplexers belonging to the core logic. Most SRAM-based FPGAs consider scan-chains [33], [34], while non-volatile FPGAs typically use memory banks [35]. For a fair comparison to RRAM-based FPGAs, we consider the configuration peripheral circuits based on memory bank for SRAM FPGAs in this paper. As illustrated in Fig. 2(b), SRAM cells belonging to the same row share a BL signal while each column is controlled by a WL signal. All the BL and WL signals are controlled by two decoders. Each SRAM cell can be individually programmed when its associated BL and WL are enabled by manipulating the two decoders. Note that with efficient sharing BLs and WLs, n SRAMs only require $\sqrt{n}$ BLs and $\sqrt{n}$ WLs. Therefore, the area of configuration circuits based on memory bank can be quadric to the number of SRAMs.

C. Limitation of FPGA CAD Tools

Most FPGA architecture exploration CAD tools, such as *Versatile Placement and Routing* (VPR) [14], [29], are designed to provide fast area, delay and power analysis for the core logic part of FPGAs. Consequently, current FPGA CAD tools face two major challenges:

(a) Inaccuracy of analytical models. To enable fast analysis, most FPGA CAD tools rely on analytical models for area, delay and power estimations [14], [16], [17], [29], [37]. Even though analytical models have been intensively refined for conventional FPGA architectures, it is still difficult to guarantee accurate estimations. For example, it is reported that the *Minimum Width Transistor* (M.W.T.) area model used in VPR suffers an overall prediction error variation of almost 2×, as compared to layout area [14], [15], [29]. The analytical FPGA power models can overestimate the total power consumption by at least 24%, as compared to electrical simulation results [16]–[19].

(b) Incomplete FPGA fabrics. Only the core logic of FPGAs and even only the used CLBs, CBs and SBs are considered in the area, delay and power analysis. Note that the configuration circuit could lead to non-negligible area overhead due to the decoders and also routing area from intensive shared BLs and WLs. The unused CLBs, CBs and SBs may also lead to area and power overhead, and such problem could be significant when resource utilization rate of a FPGA is low.

In short, the two challenges can cause inaccurate conclusion on full FPGA fabric especially in area and power. Such limitation may become more serious when evaluating novel FPGA architectures, e.g., RRAM-based, the proposed architecture under exploration in this paper.

D. RRAM-Based Circuits and FPGA Architectures

Previous works [1]–[11] about RRAM-based circuits and FPGAs rely on two principles: (1) RRAMs can store configurations as SRAMs; (2) LRS and HRS can be exploited to propagate or block datapath signals. Chen [1] and Huang *et al.* [2] studied the FPGAs based on the first principle only, where RRAMs are used as standalone memories, in the place of SRAMs. Major works [3]–[11] apply both principles in FPGA architectures, where the combination of a SRAM and a transmission-gate is replaced by a unique RRAM device. Indeed, these proposed RRAM FPGAs are predicted to improve area by 7-15%, delay by 45-58% and power by 20-58% when compared to SRAM-based counterparts, thanks to the low R_{LRS} and high integration density of RRAMs.

However, the RRAM-based circuit designs in [3]–[7], [10], and [11] employ 2T(ransistor)1R(RAM) programming structure, which has been proved less realistic and efficient than the recently proposed 4T1R programming structure [8], [9]. Moreover, previous works [1]–[11] assessed area, delay and power with analytical models [29]. Since these analytical models are exclusively designed for SRAM-based FPGAs, they are not general enough to capture the characteristics of RRAM-based FPGAs and other FPGAs based on emerging technologies. Consequently, the area, delay and power advantages of RRAM-based FPGAs predicted by [3]–[7], [10], and [11] may be misleading. This paper aims at overcoming these limitations and employ full P&R and electrical simulations to compare SRAM-based and RRAM-based FPGAs. To the best of our knowledge, it is the first work in this research field.

III. General Vision of RRAM-Based FPGA

In this section, we describe the general vision of our RRAM-based FPGAs. The RRAM-based FPGA considered in this paper has no architectural difference with respect to the conventional SRAM-based FPGA shown in Fig. 2. The major difference lies in the circuit designs of its primitive modules, and its configuration peripheral circuits.

A. Circuit Designs of Primitive Modules

To achieve non-volatility, all the SRAM-based circuits in FPGA architectures are replaced with RRAM-based implementations. We apply two different strategies depending if we are replacing the SRAMs of routing multiplexers or LUTs.

1) The whole SRAM-based routing multiplexers are replaced by 4T1R-based counterparts, as illustrated in Fig. 3(a) and (c). We borrow the 4T1R-based routing multiplexer designs from [9], where both SRAMs and transmission-gates are replaced by 4T1R elements. Hence, RRAMs behave not only as memory cells but also as logic gates that propagate or block datapath signals. Thanks to the low R_{LRS} and efficiently sharing programming transistors, the 4T1R-based routing multiplexers can bring significant improvements in area, delay, power and especially in energy consumption [9]. More importantly, such replacement leads to the performance improvements without challenging the the endurance limit of RRAM devices. An actual programming operation for 4T1R-based multiplexers occurs infrequently, only during FPGA reconfiguration.
2) In LUTs, only the SRAMs are replaced by RRAM-based non-volatile SRAM topology, as illustrated in Fig. 3(b) and (d). Different from routing multiplexers, the *on/off* state of datapath transistors can be switched frequently during each operating cycle. Note that the data storage of SRAMs is changed only during reconfiguration, which has a low switching rate tolerable to RRAM endurance. Therefore, for LUTs, RRAMs are used to grant non-volatility to SRAMs, rather than to datapath transistors.

B. Configuration Peripheral Circuits

Different from previous works [1]–[11], in addition to the core logic, we also consider the effect of configuring peripherals and I/Os in the evaluation (see details in Section V-B). In our RRAM-based FPGA architecture, each 4T1R element [8] is accessed by BLs and WLs as well but requires two BLs and two WLs. Those lines are shared as shown in Fig. 2, where BLs and WLs of each 4T1R-based multiplexer and each RRAM of LUTs are divided into two groups:

1) Common BLs and WLs (highlighted blue in Fig. 3) that are shared by all the 4T1R elements (belonging to the multiplexer as well as the LUTs). Take the example in Fig. 3(b), (c) and (d), the two N-input 4T1R-based multiplexers share $BL[0...N-1]$ and $WL[0...N-1]$, and the NV SRAM share $BL[0]$ and $WL[0]$ with the multiplexers. Considering the different input size of multiplexers in FPGA architecture, the number of shared BLs and WLs is determined by the largest input size of multiplexers.
2) Independent BLs and WLs (highlighted red in Fig. 3), which are unique for each 4T1R-based element.

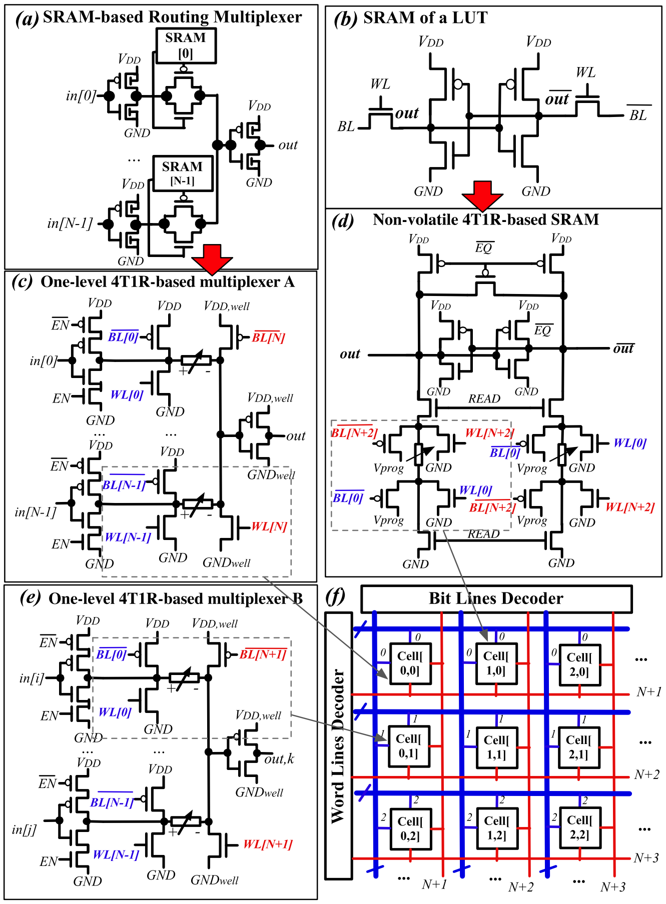

Fig. 3. Circuit designs of (a) SRAM-based routing multiplexer; (b) SRAM; (c) 4T1R-based routing multiplexer; (d) Non-volatile 4T1R-based SRAM; (e) Another 4T1R-based routing multiplexer sharing BLs and WLs; (f) Proposed *Bit Line* (BL) and *Word Line* (WL) sharing strategy for RRAM-based FPGAs.

As shown in Fig. 3(c) and (e), the programming transistors close to the output inverters in the two 4T1R-based multiplexer are controlled by two unique BLs and WLs, $(BL[N], WL[N])$ and $(BL[N+1], WL[N+1])$, respectively. Similarly, the NV SRAM in Fig. 3(d) has an unique pair of BL and WL, $BL[N+2], WL[N+2]$.

As such, each RRAM can be individually configured by addressing its unique set of BL and WL signals. As activating each set of BL and WL signals only allows the programming current to flow through a unique RRAM, the BL and WL sharing strategy in Fig. 3 can avoid parasitic programming.

Our RRAM-based FPGA architecture requires the number of BLs and WLs be linear to the number of NV SRAMs and 4T1R-based multiplexers. When compared to the SRAM-based FPGAs, whose number of BLs and WLs is the square root to the number of SRAMs, this could lead to large decoder circuits and potentially area overhead. However, our RRAM-based FPGA eliminates the use of SRAMs in routing multiplexers, bringing significant area reduction. Considering that routing multiplexers generally occupies more than 70% of the total area, the area overhead from the decoder circuits can be fully compensated by the 4T1R-based multiplexers. Overall, our RRAM-based FPGA will be as area efficient as its SRAM-based counterpart or even better, depending on the scale of routing architecture, which is validated by layout-level results in Section V-B.

IV. CAD Support and EDA Flow

To acquire post P&R results for RRAM-based FPGAs, we developed a new EDA flow, based on FPGA-SPICE [19]. As illustrated in Fig. 4, the new flow consists of two parts:

1) The traditional VPR-based FPGA EDA flow [14], where benchmark circuits are logic optimized by ABC [39] and then processed through activity estimation [20], packing, placement and routing of original VPR [14].
2) A novel EDA add-on capable of modeling a full FPGA fabric in Verilog and SPICE netlists based on the VPR results and the architecture description.

In order to have a seamless integration with HDL simulators and semi-custom backend tools, we develop a Verilog generator and integrate it into FPGA-SPICE. As illustrated in Fig. 4, the Verilog generator outputs structural Verilog netlists modeling the RRAM-based circuit designs and FPGA architectures described in Section III. All the structural Verilog netlists describing FPGA primitives, modules and full fabrics are synthesizable, which can be directly used for semi-custom design tools. For the top-level Verilog netlist, an associated testbench is automatically generated for full-chip-level verification purpose. Note that our Verilog generator is more capable than [31] in two aspects:

1) It can support transistor-level circuit designs and FPGA architectures based on RRAMs.
2) It can support one-level and two-level SRAM-based multiplexer designs, which are widely used in commercial FPGAs [33], [34].

In this paper, we exploit the FPGA-SPICE with Verilog auto-generation to achieve accurate area, delay and power results for both SRAM-based and RRAM-based FPGAs.

1) To ensure functional correctness of FPGAs, all the Verilog and SPICE netlists are verified to deliver the same outputs as pre-VPR netlists under random input vectors, as shown in Fig. 4(b).
2) To perform area analysis for full FPGA fabric, we employ semi-custom design tool Cadence Innovus [42] to generate full-chip layouts, as shown in Fig. 4(c). Note that in addition to area results, the full-chip layout can be directly used for fabrication purpose, enabling fast prototyping for both SRAM-based and RRAM-based FPGAs.
3) To perform delay analysis, we run SPICE simulations for each component in a FPGA, i.e., LUTs, FFs and multiplexers. The timing results are back-annotated to the timing analysis engine in VPR to estimate accurate critical path delays.
4) For accurate power analysis, we enhance FPGA-SPICE to support the most recent RRAM-based circuit designs [8], [9]. SPICE netlists and associated testbenches are automatically generated for each component in full FPGA fabrics, by considering their actual loads and signal activities in the architecture context and benchmark circuits. As shown in Fig. 4(d), HSPICE [41] is

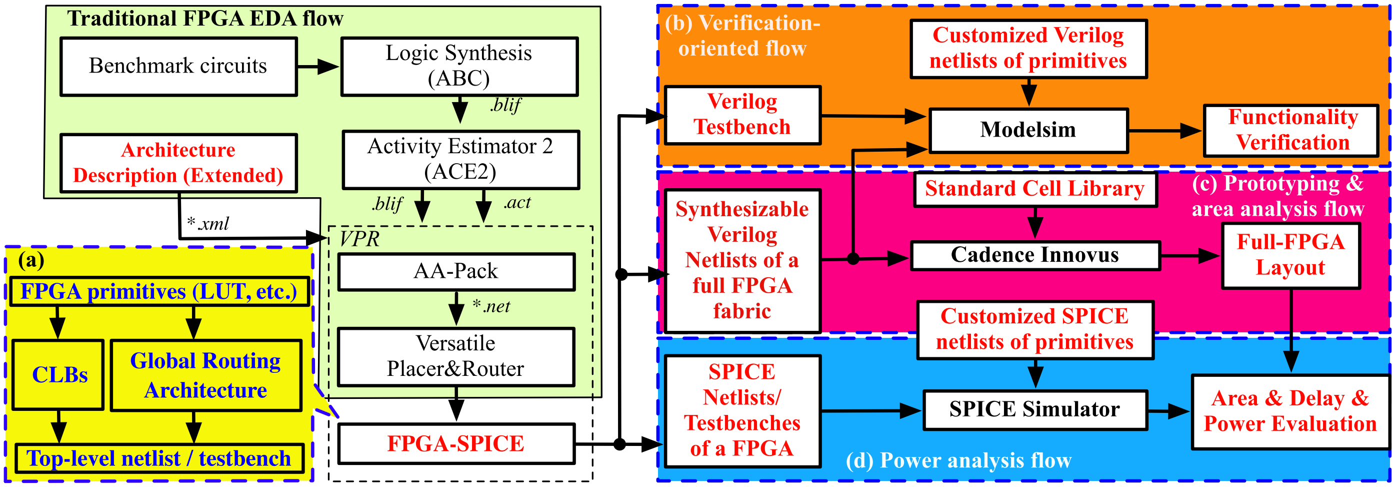
Traditional FPGA EDA flow
Benchmark circuits
Logic Synthesis (ABC)
.blif
Architecture Description (Extended)
Activity Estimator 2 (ACE2)
.blif
.act
*.xml
VPR
AA-Pack
*.net
Versatile Placer&Router
FPGA-SPICE
(a)
FPGA primitives (LUT, etc.)
CLBs
Global Routing Architecture
Top-level netlist / testbench
(b) Verification-oriented flow
Customized Verilog netlists of primitives
Verilog Testbench
Modelsim
Functionality Verification
Standard Cell Library
(c) Prototyping & area analysis flow
Synthesizable Verilog Netlists of a full FPGA fabric
Cadence Innovus
Full-FPGA Layout
Customized SPICE netlists of primitives
SPICE Netlists/ Testbenches of a FPGA
SPICE Simulator
Area & Delay & Power Evaluation
(d) Power analysis flow

Fig. 4. Proposed EDA flows based on Verilog Generator for accurate area, delay and power analysis.

employed to perform power analysis and total power consumption is achieved by summing up the power results extracted from each HSPICE simulation.

Note that area, delay and power results achieved by the novel EDA flow are more accurate and realistic than the analytical models in VPR, and can be used as baseline to evaluate the accuracy of analytical models.

V. Experimental Results

In this section, we analyze the area, delay and power results achieved by the developed EDA flow in Fig. 4.

A. Methodology

We exploit the novel EDA flow in Fig. 4 to compare the area, delay and power of SRAM-based and RRAM-based FPGAs. The twenty largest MCNC benchmarks [40] are selected as the input of the EDA flow. All the experiments are run on a 64-bit RedHat Linux server with 28 Intel Xeon Processors and 256GB memory.

To be fair in comparison, both SRAM-based and RRAM-based FPGAs employ a CLB architecture with forty inputs pins ($I = 40$). Each CLB consists of ten BLEs ($N = 10$), each of which contain a 6-input LUT ($K = 6$) [37]. Similar to commercial FPGAs [33], [34], we consider unidirectional routing architectures [36] with three types of wire lengths. In each routing channel, 30% of routing tracks are built with length-1 wires $L = 1$, another 30% of routing tracks are built with length-2 wires ($L = 2$) and the rest 40% of routing tracks are built with length-4 wires ($L = 4$). Each routing track can be connected to other three routing tracks from adjacent channels ($Fs = 3$). Each CLB input pin can be connected to 15% of the routing tracks in a channel ($F_{c,in} = 0.15$), while each CLB output pin can reach 10% of the routing tracks ($F_{c,out} = 0.10$).

Both SRAM-based and RRAM-based FPGAs are built with a commercial 40nm technology. To guarantee the best overall performance, multiplexers in local routing architecture and CBs adopt a two-level structure while the others are built with a one-level structure [36], [37]. All the RRAM-based multiplexers adopt a one-level structure and RRAMs are placed between the first and the second metal layer, for best overall performance [9]. The datapath circuits and the 4T1R programming structures are built with standard logic transistors ($W/L = 140nm/40nm$). Transmission gates are implemented by a pair of minimum-width n-type and p-type logic transistor. Input and output inverters are sized to 3× minimum width in order to resist the parasitics of metal wires.

We consider a RRAM technology [46], [47] with programming voltages $V_{set} = |V_{reset}| = 1.1V$ and a maximum current compliance of $I_{set} = |I_{reset}| = 500\mu A$. The lowest achievable R_{LRS} of a RRAM is $2.2k\Omega$ while the R_{HRS} is swept from $10M\Omega$ to $100M\Omega$ in order to identify the required property for achieving a good energy efficiency. The considered range of R_{HRS} can be achieved by applying different programming conditions, e.g., programming current, which has been validated by experimental measurements in [48]. The Stanford RRAM compact model [38] is used to model the considered RRAM technology. In this work, we assume that the RRAMs are co-embedded in the *MET2* and *MET3* vias, leading to a feature size of $130nm \times 130nm$ in the considered 40nm technology. Previous works have shown successful integration of RRAM technologies in the metal vias, which are similar in dimension to the considered commercial 40nm technology [44]–[46]. To accurately include the parasitic effects from the co-integration, we add a parasitic capacitance of $13.2aF$ to the RRAM SPICE model, which is estimated by considering the height and the dimension of metal vias in the commercial 40nm technology.

B. Area Characteristics

1) Full-Chip Layouts: Fig. 5 presents the full-chip layouts of SRAM-based and RRAM-based FPGAs, both of which including core logics, configuring peripherals and IOs. Note that both FPGAs contain a channel width of 300, which is similar to commercial FPGAs [33], [34]. For sake of the

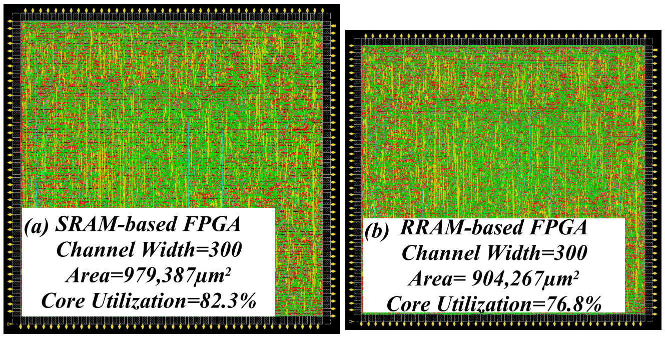

Fig. 5. Full-chip layouts (Channel width is set to 300) of FPGAs configured by BL and WL decoders: (a) SRAM-based and (b) RRAM-based.

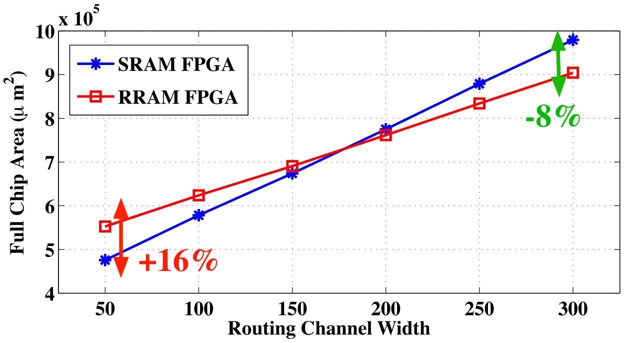

Fig. 7. Full-chip area of SRAM-based and RRAM-based FPGAs with different routing channel width.

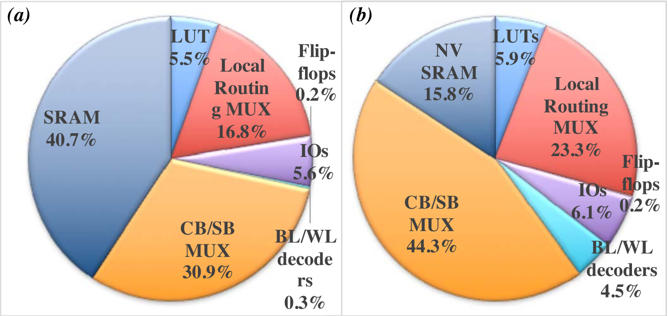

Fig. 6. Area breakdown (Channel width is set to 300) of FPGAs configured by BL and WL decoders: (a) SRAM-based and (b) RRAM-based.

capability of our workstation, we consider a CLB array size of 5×5 which are surrounded by 160 I/O pads. Note that the achieved area results with a 5×5 CLB array can be representative because large FPGAs can be regarded as an assembly of the small CLB arrays. The full-chip layout comparison shows that RRAM-based FPGAs can as area efficient (with a 8% improvement) as SRAM-based FPGA when considering full configuring peripherals.

2) Area Breakdown: Fig. 6 compares the area breakdown of SRAM-based and RRAM-based FPGA chips when channel width is set to 300. Routing multiplexers, including local routing architecture, CBs and SBs, occupy 46-70% of the total area, which are the major contributors in both FPGAs, while LUTs and FFs stand only up to 6% in the total area. More than 40% of the total area is consumed by SRAMs in the SRAM-based FPGA, while only 15% of the total area is consumed by NV SRAMs in the RRAM-based FPGA. This area difference is due to 4T1R-based multiplexers eliminating the use of SRAMs and that NV SRAMs only occur in LUTs. As a result, the SRAM-based FPGA contains 180,470 SRAMs, while the RRAM-based FPGA reduces the number to only 16,160 NV SRAMs. This contributes to the RRAM-based FPGA requiring 8% less total area than the SRAM-based FPGA.

3) Impact of Routing Channel Width W: Routing channel width W, representing the number of routing tracks per FPGA routing channel, is a key factor impacting the area of FPGA architectures. In this part, we swept the routing channel width from 50 to 300 with a step of 50 for both SRAM-based and RRAM-based FPGA architectures. Fig. 7 compares the full-chip area of SRAM-based and RRAM-based FPGAs by considering different routing channel widths. A large W increases the capacity of FPGAs but requires many routing multiplexers. As 4T1R-based multiplexers are more area efficient than their SRAM-based counterparts [9], the area benefits of RRAM-based FPGAs can be significant when a large W is considered, which is typical in modern FPGA architectures. In contrast, a small routing channel width requires a small number of 4T1R-based multiplexers. The area improvement can be overshadowed by :

- the overhead of NV SRAMs. As depicted in Fig. 3(d), a NV SRAM requires more transistors than a volatile SRAM.
- the overhead of the configuration circuits of our RRAM-based FPGAs. As illustrated in Fig. 3(f), the SRAM-based FPGAs requires the number of BL/WL lines to be the square root to the number of SRAMs, while the RRAM-based FPGAs requires the number of BLs and WLs be linear to the number of NV SRAMs and RRAM-based multiplexers. This results in that the configuration circuits of our RRAM-based FPGAs are larger in area than the SRAM-based counterparts, which is validated in Fig. 6.

When a small routing channel width is applied, the number of RRAM-based multiplexers is small and their area benefits are not enough to compensate the area overhead from NV SRAMs and configuration circuits. When routing channel width increases, the area benefits of RRAM-based multiplexers becomes large enough to compensate the area overhead, and lead to area improvements. Therefore, in Fig. 7, we see area overheads from the proposed RRAM-based FPGAs when $W \leq 150$, while when $W \geq 150$, the proposed RRAM-based FPGAs become more area efficient than SRAM-based FPGA. We believe that significant area reduction can be achieved when the routing channel width is larger than 300.

C. Power Characteristics

1) Impact of R_{HRS}: As explained in [11], the R_{HRS} can influence the power consumption of RRAM-based routing elements. We evaluate in Fig. 8 the impact of R_{HRS} on the average energy consumption of the considered FPGA architectures implementing in MCNC big20 benchmarks by using FPGA-SPICE. Indeed, a low R_{HRS} leads to a high leakage power

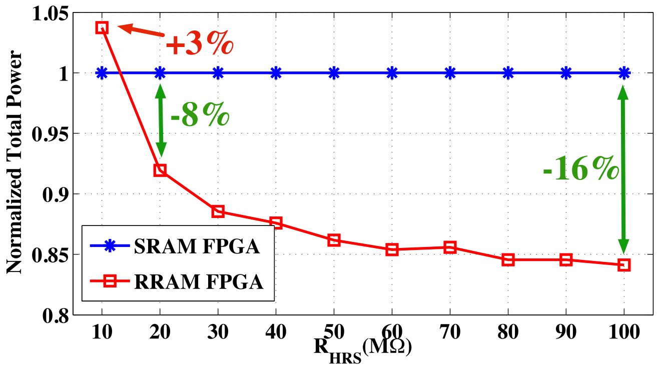

Fig. 8. Normalized power consumption of RRAM-based FPGAs with different R_{HRS} (Channel width W is set to 300).

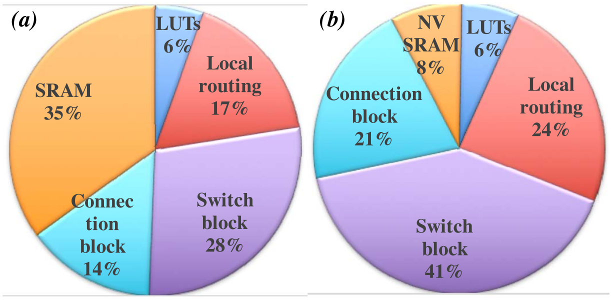

Fig. 9. Leakage power breakdown (Channel width W is set to 300) of (a) SRAM-based and (b) RRAM-based FPGAs.

and it causes 3% overheads in total power when $R_{HRS} = 10M\Omega$, as shown in Fig. 8. When $R_{HRS} \geq 20M\Omega$, RRAM-based FPGAs become more power efficient than SRAM-based FPGAs. Note that the power reduction is achieved with performance improvement. However, the leakage power overhead can be mitigated by not only an increase in R_{HRS} but also non-volatility. RRAM-based FPGAs can be fully powered down during a long idle period and then instantly turned on for operation. As a result, leakage power of RRAM-based FPGA only occurs during standard operation time, which is typically along with high dynamic power consumption. The dynamic power of 4T1R-based multiplexers is 20% smaller than CMOS multiplexers, leaving more budget for leakage power [9]. All the factors contribute to the fact that lower bound of R_{HRS} of RRAM-based FPGAs ($\geq 20M\Omega$) can be much lower than the *off*-resistance of transmission gates in SRAM-based FPGA (In the considered technology, it is $\sim 500M\Omega$), to guarantee similar energy efficiency. In the rest of this paper, we consider $R_{HRS} = 20M\Omega$ for RRAM-based FPGAs.

2) Power Breakdown: Fig. 9 compares the leakage power breakdown between RRAM-based and SRAM-based FPGAs. In general, routing multiplexers consumes 69% of the total leakage power, while LUTs and SRAMs only consumes up to 41% of the total. Due to the heavy use of SRAMs, 35% of the leakage power is consumed by SRAMs in SRAM-based FPGA, which are consistent with those reported by commercial products [43]. Differently, in RRAM-based FPGA, only 5% is required by NV SRAMs, because they are only used in LUTs. 4T1R-based multiplexers eliminate the use of SRAMs, significantly reducing the weight of SRAM leakage power. This also leaves more leakage power budget to multiplexers, relaxing the lower bound of R_{HRS}. Fig. 10 compares the dynamic power breakdown between RRAM-based and SRAM-based FPGAs. We see that over 70% of the total power is consumed by routing multiplexers, while only 28.8% is consumed by LUTs. By removing the SRAMs in routing multiplexers, the power share of SRAMs is reduced from 23% (SRAM-based FPGA) to 5% in RRAM-based FPGA.

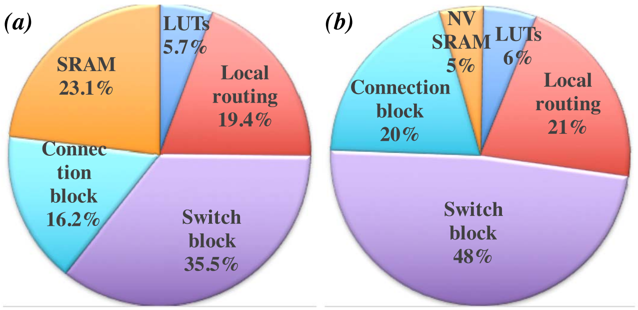

Fig. 10. Total power breakdown (Channel width W is set to 300) of (a) SRAM-based and (b) RRAM-based FPGAs.

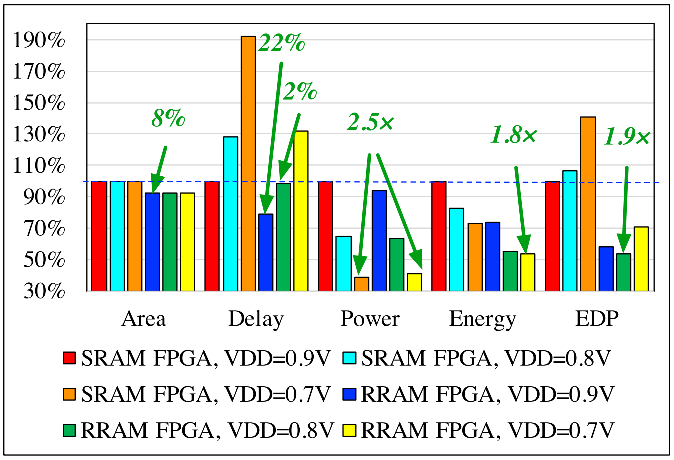

Fig. 11. Area, delay and energy comparison between SRAM-based and RRAM-based FPGAs operating at nominal and near-V_t regime.

D. Near-V_t Opportunity

As predicted in [8]–[11], near-V_t RRAM-based circuits and FPGAs can achieve both high-performance and low-power when compared to CMOS counterparts working at nominal voltage. This is due to the resistance of RRAMs being independent from working voltage, unlike transistors whose equivalent resistance degrades seriously at near-V_t regime. Fig. 11 compares the overall performance of SRAM-based and RRAM-based FPGAs operating at both nominal and near-V_t regime. When operating at nominal voltage ($V_{DD} = 0.9V$), RRAM-based FPGA can improve delay by 22% over its SRAM-based counterpart. Even when V_{DD} is reduced to near-V_t regime, i.e., $0.8V$, RRAM-based FPGA remains the same performance-level as the SRAM-based FPGA at nominal voltage. And the near-V_t RRAM-based FPGA benefits from significant energy reduction, leading to a $2-2.3\times$ improvement on energy. Note that the energy of RRAM-based FPGA operating at $V_{DD}=0.9V$ is similar to the best SRAM-based FPGA ($V_{DD} = 0.7V$). In terms of *Energy-Delay Product* (EDP), SRAM-based FPGA at nominal voltage is the best, while RRAM-based FPGA at $V_{DD}=0.8V$ is the best with a close to 2× improvement compared to the best SRAM-based FPGA.

E. Comparison to Analytical Models

While staying in a similar range, it is important to note that the predicted performance gain shown in this paper is smaller than previous works [1]–[11]. However, the present paper considers full FPGA fabric (core and periphery) and employs semi-custom design flow and electrical simulations in evaluations, which delivers more realistic results:

1) For the area evaluation, we consider the full parasitics of RRAM-based circuits and also include configuration circuits and I/Os in full-chip layouts, which are ignored in previous works [1]–[11]. Fig. 5 shows that interconnecting metal wires is the dominant factor. Analytical models only focus on the area consumed by standard cells, which only occupy 76.8% of the total area in Fig. 5. Fig. 6 shows that configuration circuits and I/Os can consume a non-negligible 10.6% of the total area of RRAM-based FPGAs. Analytical models only focus on the core logic, leading to a significant accuracy loss in the area estimation.
2) For the power evaluation, we consider electrical simulation results to approach practical operating activities, which cannot be accurately captured by analytical models. Fig. 8 shows that the choice of R_{HRS} could lead to significant power difference for RRAM-based FPGAs, which was ignored by the analytical models in previous works [1]–[10]. This could lead to an error of $> 16\%$ for the analytical power results. Fig. 9 and Fig. 10 show that SRAMs and NV SRAMs consume 5-35% of the total power, whose impact has also not been explicitly studied in most analytical models [1]–[10].

F. Impact of RRAM Variations

RRAMs are known to exhibit extensive variations in resistance, due to its stochastic nature in filamentary conducting [44]–[46]. In this paper, we focus on studying the robustness of the proposed RRAM-based FPGAs against the corner variations from both CMOS and RRAM technologies. In electrical simulations, we consider all the possible combinations of the following corner variations:

1) Three CMOS corner cases, namely *Fast-Fast* (FF), *Typical-Typical* (TT) and *Slow-Slow* (SS), which are provided natively in the considered commercial 40nm technology.
2) Three RRAM corner cases with variations on R_{LRS} and R_{HRS}, called *Best*, *Typical* and *Worst*, as detailed in Table I. In the *Typical* case, we consider the R_{HRS} and R_{LRS} as reported in the experimental results [46], [47]. In the *Best* and the *Worst* cases, we apply up to 30% variations to R_{LRS} and R_{HRS}, which are typical values reported experimentally [44]–[46]. The *Best* case puts us in the high-performance and low-energy corner, while the *Worst* case represents the low-performance and high-energy corner.

TABLE I

DETAILED R_{LRS} AND R_{HRS} VARIATIONS FOR THE DIFFERENT RRAM CORNER CASES.

RRAM corners	R_{LRS}	R_{HRS}
Best	$3.7k\Omega$	$26M\Omega$
Typical	$4.8k\Omega$	$20M\Omega$
Worst	$6.3k\Omega$	$14M\Omega$

In total, we performed 9 corner analyses on the RRAM-based FPGA operating at nominal V_{DD} and studied their impacts on the delay and energy.

Fig. 12(a) and (b) depicts the shift on FPGA delay and energy impacted when considering both CMOS and RRAM variations, in the case of benchmark *s298*. The simulation results presented that CMOS variations could cause a 20% delay degradation and a 50% energy degradation respectively to RRAM-based FPGAs, while RRAM variations leads to limited impacts with less than 3% in delay shift and up to 8% in energy shift. Such stability on performance and energy can be explained as follows:

1) The impact of RRAM variations is limited on RRAM-based multiplexer. As illustrated in Fig. 3(c), RRAM-based multiplexers consist of considerable transistors on the datapaths, such as driving inverters and programming transistors. Therefore, the resistance of RRAMs has a limited impact on the delay and energy characteristics. In our SPICE simulations, when considering a standalone RRAM-based multiplexer, the delay shift could be as large as 10%, but it is rather smaller than the RRAM variations.
2) The proposed FPGA architectures include considerable pure CMOS circuits, such as LUTs and FFs, which further reduce the impact of RRAM variations on performance and power.

Note that the corner analyses present the pessimistic results under RRAM variations. In practice, the variations of RRAMs are dominated by the cycle-to-cycle variation and each RRAM ends up having an independent variation that would be better capture by Monte-Carlo simulations. A decrease in R_{LRS} results in delay improvement for RRAM-based multiplexers, while an increase in R_{LRS} causes delay degradation; A decrease in R_{HRS} leads to energy overheads in the RRAM-based multiplexers, while an increase in R_{HRS} would lead to energy savings. Therefore, at architecture-level, the variation on delay and energy may be fully mitigated. To be illustrative, we performed a 100-run Monte-Carlo SPICE simulation for full FPGA fabrics considering the same benchmark as the corner analyses, and the resulting delay and energy distributions are shown in Fig. 13.

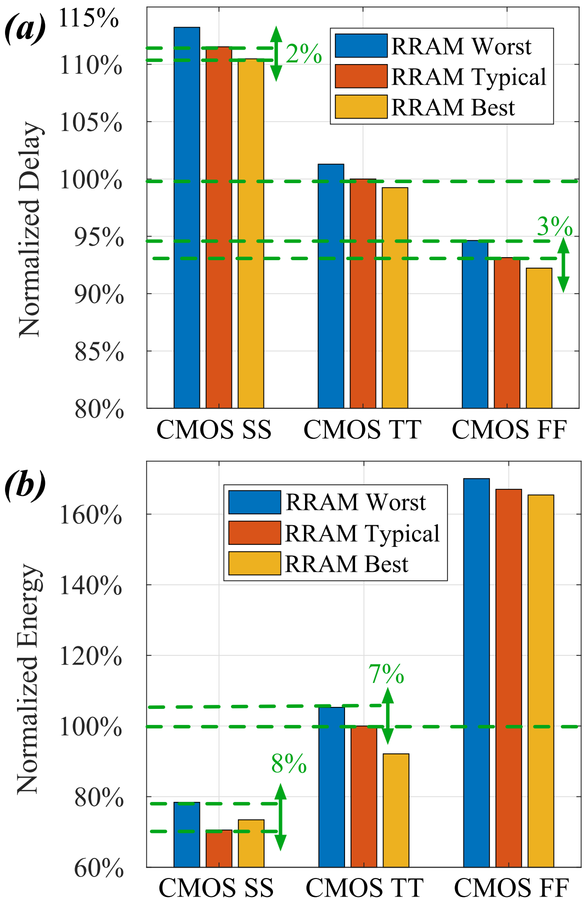

Fig. 12. Case study on benchmark *s298*: Impact of RRAM and CMOS corners on the RRAM-based FPGA operating at $V_{DD} = 0.9V$: (a) delay and (b) energy.

By presenting both the pessimistic case with a corner analysis and time-consuming Monte-Carlo simulations, we show the robustness of proposed RRAM-based FPGA to both CMOS and RRAM process variations, indicating that they can bring reliable delay and energy advantages over SRAM-based counterparts.

VI. Discussion

In this paper, we focused only on the energy efficiency achieved by the FPGAs during operating period. When considering non-volatility, the energy improvement of RRAM-based FPGAs could go beyond 2×. As depicted in Fig. 14, compared to SRAM-based FPGAs, RRAM-based FPGAs can be simply powered off during sleep mode and be instantly powered on when needed. As a result, two types of power consumption can be eliminated when using RRAM-based FPGA: (1) the operational leakage power, highlighted yellow in Fig. 14(a) and (2) the reconfiguration power each time FPGAs are powered on, highlighted blue in Fig. 14(a). In practice, the two types of power consumption could dominate the total power. Considering some *Internet of Things* (IoTs) applications, the energy improvement of RRAM-based FPGAs can be magnified by 10×, as over 90% of total power is consumed by the operational leakage power.

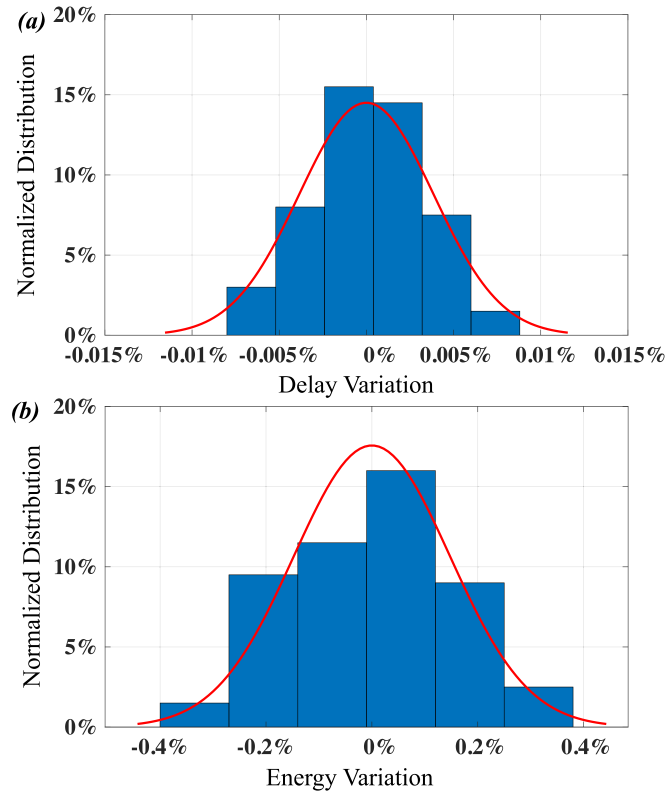

Fig. 13. Monte-Carlo results for benchmark *s298*: impact of RRAM variation on (a) delay and (b) energy.

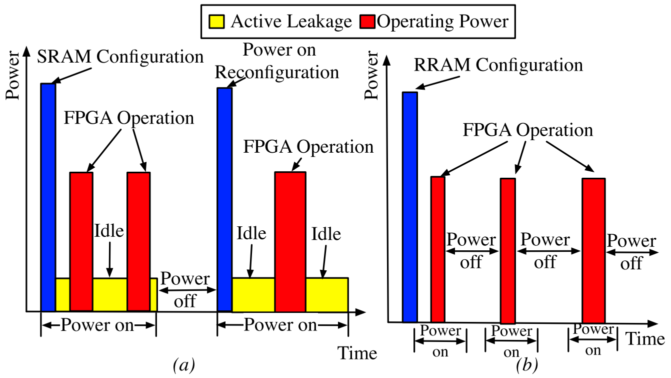

Fig. 14. Power consumption of (a) a SRAM-based FPGA and (b) a RRAM-based FPGA.

In Fig. 11 and Fig. 12, we presented the large V_{DD} space of RRAM-based FPGAs in energy-quality trade-off and their robustness against variations when compared to the SRAM counterparts. Such features can be exploited in approximate computing applications, where *Dynamic Voltage Scaling* (DVS) techniques can be applied to achieve low energy consumption with limited quality/performance degradation [12], [13].

VII. Conclusion

In this paper, we have developed a Verilog generator to model full FPGA fabrics, in order to enable a realistic study on the area and power characteristics of RRAM-based

FPGAs with full-chip layouts and electrical simulations. We enhance FPGA-SPICE to output SPICE netlists modeling recent advanced RRAM-based circuits and FPGA architectures. Accurate power analysis is performed by using SPICE simulators. Considering a commercial 40nm technology, averaged over the twenty largest MCNC benchmarks, experimental results show that RRAM-based FPGAs can improve up to 8% in area, on average 22% in delay and on average 16% in power respectively, as compared to SRAM-based counterparts. In particular, the area efficiency of RRAM-based FPGAs becomes significant when large channel width is applied. Electrical simulations show that R_{HRS} of RRAM-based FPGAs should be at least $20M\Omega$ to achieve the same power efficiency of SRAM-based FPGAs. Especially, when compared to the SRAM-based FPGAs working at nominal voltage, near-V_t RRAM-based FPGAs can outperform close to 2× in *Energy-Delay Product* without delay overhead. Validated by realistic post P&R results, RRAM-based FPGAs are more capable in trading-off energy and quality than SRAM-based counterparts.

Acknowledgment

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL and DARPA or the U.S. Government.

References

[1] Y.-C. Chen, W. Wang, H. Li, and W. Zhang, "Non-volatile 3D stacking RRAM-based FPGA," in *Proc. IEEE FPL*, Oslo, Norway, Aug. 2012, pp. 367–372.

[2] K. Huang, R. Zhao, W. He, and Y. Lian, "High-density and high-reliability nonvolatile field-programmable gate array with stacked 1D2R RRAM array," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 1, pp. 139–150, Jan. 2016.

[3] S. Tanachutiwat, M. Liu, and W. Wang, "FPGA based on integration of CMOS and RRAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 11, pp. 2023–2032, Nov. 2011.

[4] J. Cong and B. Xiao, "FPGA-RPI: A novel FPGA architecture with RRAM-based programmable interconnects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 4, pp. 864–877, Apr. 2014.

[5] P.-E. Gaillardon, M. H. Ben-Jamaa, G. B. Beneventi, F. Clermidy, and L. Perniola, "Emerging memory technologies for reconfigurable routing in FPGA architecture," in *Proc. IEEE ICECS*, Dec. 2010, pp. 62–65.

[6] P.-E. Gaillardon, D. Sacchetto, S. Bobba, Y. Leblebici, and G. De Micheli, "GMS: Generic memristive structure for non-volatile FPGAs," in *Proc. IEEE/IFIP VLSI-SoC*, Oct. 2012, pp. 94–98.

[7] P. Gaillardon *et al.*, "Design and architectural assessment of 3-D resistive memory technologies in FPGAs," *IEEE Trans. Nanotechnol.*, vol. 12, no. 1, pp. 40–50, Jan. 2013.

[8] X. Tang, G. Kim, P.-E. Gaillardon, and G. De Micheli, "A study on the programming structures for RRAM-based FPGA architectures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 4, pp. 503–516, Apr. 2016.

[9] X. Tang, E. Giacomin, G. De Micheli, and P.-E. Gaillardon, "Circuit designs of high-performance and low-power RRAM-based multiplexers based on 4T(ransistor)1R(RAM) programming structure," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 5, pp. 1173–1186, May 2017.

[10] X. Tang, P.-E. Gaillardon, and G. De Micheli, "A high-performance low-power near-V_t RRAM-based FPGA," in *Proc. IEEE ICFPT*, Dec. 2014, pp. 207–214.

[11] X. Tang, P.-E. Gaillardon, and G. De Micheli, "Accurate power analysis for near-V_t RRAM-based FPGA," in *Proc. IEEE FPL*, London, U.K., Sep. 2015, pp. 1–4.

[12] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 89–102, 2001.

[13] M. Alioto, "Energy-quality scalable adaptive VLSI circuits and systems beyond approximate computing," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 127–132.

[14] J. Rose *et al.*, "The VTR project: Architecture and CAD for FPGAs from verilog to routing," in *Proc. FPGA*, 2012, pp. 77–86.

[15] F. F. Khan and A. Ye, "An evaluation on the accuracy of the minimum width transistor area models in ranking the layout area of FPGA architectures," in *Proc. IEEE FPL*, Lausanne, Switzerland, Aug./Sep. 2016, pp. 1–11.

[16] K. K. W. Poon, S. J. E. Wilton, and A. Yan, "A detailed power model for field-programmable gate arrays," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 10, no. 2, pp. 279–302, 2005.

[17] J. B. Goeders and S. J. E. Wilton, "VersaPower: Power estimation for diverse FPGA architectures," in *Proc. IEEE ICFPT*, Dec. 2012, pp. 229–234.

[18] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 11, pp. 1712–1724, Nov. 2005.

[19] X. Tang, P.-E. Gaillardon, and G. De Micheli, "FPGA-SPICE: A simulation-based power estimation framework for FPGAs," in *Proc. IEEE ICCD*, New York, NY, USA, Oct. 2015, pp. 696–703.

[20] J. Lamoureux and S. J. E. Wilton, "Activity estimation for field-programmable gate arrays," in *Proc. IEEE FPL*, Aug. 2006, pp. 1–8.

[21] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class-memory," *J. Res. Develop.*, vol. 52, nos. 4–5, pp. 449–464, Jul./Sep. 2008.

[22] H.-S. P. Wong *et al.*, "Metal-oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012.

[23] Y. Wu, J. Liang, S. Yu, X. Guan, and H.-S. P. Wong, "Resistive switching random access memory—Materials, device, interconnects, and scaling considerations," in *Proc. IEEE Int. Integr. Rel. Workshop Final Rep.*, Oct. 2012, pp. 16–21.

[24] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S. P. Wong, "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling," in *IEDM Tech. Dig.*, Dec. 2012, pp. 10.4.1–10.4.4.

[25] F. M. Puglisi, P. Pavan, L. Larcher, and A. Padovani, "Analysis of RTN and cycling variability in HfO_2 RRAM devices in LRS," in *Proc. ESSDERC*, Sep. 2014, pp. 246–249.

[26] A. Levisse, B. Giraud, J. P. Noël, M. Moreau, and J. M. Portal, "SneakPath compensation circuit for programming and read operations in RRAM-based crosspoint architectures," in *Proc. IEEE 15th Non-Volatile Memory Technol. Symp.*, Oct. 2015, pp. 1–4.

[27] H. Aziza, M. Bocquet, M. Moreau, and J.-M. Portal, "A built-in self-test structure (BIST) for resistive RAMs characterization: Application to bipolar OxRRAM," *Solid-State Electron.*, vol. 103, pp. 73–78, Jan. 2015.

[28] F. M. Puglisi, C. Wenger, and P. Pavan, "A novel program-verify algorithm for multi-bit operation in HfO_2 RRAM," *IEEE Electron Device Lett.*, vol. 36, no. 10, pp. 1030–1032, Oct. 2015.

[29] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Norwell, MA, USA: Kluwer, 1999.

[30] I. Kazi *et al.*, "Energy/reliability trade-offs in low-voltage ReRAM-based non-volatile flip-flop design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 11, pp. 3155–3164, Nov. 2014.

[31] J. H. Kim and J. H. Anderson, "Synthesizable FPGA fabrics targetable by the verilog-to-routing (VTR) CAD flow," in *Proc. IEEE FPL*, London, U.K., Sep. 2015, pp. 1–8.

[32] M. Hutton *et al.*, "Improving FPGA performance and area using an adaptive logic module," in *Proc. FPL*, 2004, pp. 135–144.

[33] Altera Corporation. (Jul. 2015). *Stratix 10 TX FPGA Advance Information Brief*. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/stratix-10/S10_TX_FPGA_AIB.pdf

[34] Xilinx Corporation. (May 2015). *Virtex-7 User Guide DS180 (V1.17)*. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[35] J. Greene *et al.*, "A 65 nm flash-based FPGA fabric optimized for low cost and power," in *Proc. 19th ACM/SIGDA Int. Symp. Field Program. Gate Arrays (FPGA)*, New York, NY, USA, 2001, pp. 87–96.

[36] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in FPGA interconnect," in *Proc. IEEE ICFPT*, Dec. 2004, pp. 41–48.

[37] C. Chiasson and V. Betz, "Should FPGAs abandon the pass-gate?" in *Proc. FPL*, Sep. 2013, pp. 1–8.

[38] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, and H.-S. P. Wong, "Verilog-A compact model for oxide-based resistive random access memory (RRAM)," in *Proc. SISPAD*, Sep. 2014, pp. 41–44.

[39] University of California in Berkeley. *ABC: A System for Squential Synthesis and Verification*. Accessed: Jul. 17, 2016. [Online]. Available: http://www.eecs.berkeley.edu/~alanmi/abc

[40] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," in *Proc. MCNC*, Jan. 1991, pp. 1–45.

[41] Synopsys Inc. *HSPICE: The Gold Standard for Accurate Circuit Simulation*. Accessed: 2018. [Online]. Available: https://www.synopsys.com/content/dam/synopsys/verification/datasheets/hspice-ds.pdf

[42] *Innovus Implementation System: Meet PPA and TAT Requirements At Advanced Nodes*, Cadence Des. Syst., San Jose, CA, USA, 2018.

[43] T. Tuan and B. Lai, "Leakage power analysis of a 90 nm FPGA," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2003, pp. 57–60.

[44] Z. Wei *et al.*, "Distribution projecting the reliability for 40 nm ReRAM and beyond based on stochastic differential equation," in *IEDM Tech. Dig.*, Washington, DC, USA, Dec. 2015, pp. 7.7.1–7.7.4.

[45] C.-F. Lee, H.-J. Lin, C.-W. Lien, Y.-D. Chih, and J. Chang, "A 1.4 Mb 40-nm embedded ReRAM macro with 0.07 μm^2 bit cell, 2.7 mA/100 MHz low-power read and hybrid write verify for high endurance application," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Seoul, South Korea, Nov. 2017, pp. 9–12.

[46] H. Y. Lee, *et al.*, "Low power and high speed bipolar switching with a thin reactive Ti buffer layer in robust HfO_2 based RRAM," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2008, pp. 1–4.

[47] M. Thammasack, G. De Micheli, and P.-E. Gaillardon, "Effect of O^{2-} migration in Pt/HfO_2/Ti/Pt structure," *J. Electroceram.*, vol. 39, nos. 1–4, pp. 137–142, 2017.

[48] A. Grossi *et al.*, "Experimental investigation of 4-kb RRAM arrays programming conditions suitable for TCAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8306516

Xifan Tang (S'13–M'17) received the B.Sc. degree in microelectronics from Fudan University, Shanghai, China, in 2011, and the M.Sc. degree in electrical engineering and the Ph.D. degree in computer science from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2013 and 2017, respectively. He is currently a Post-Doctoral Researcher with The University of Utah. His current research interests include computer-aided design for programmable architecture and emerging technologies. He was a recipient of the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad.

Edouard Giacomin received the M.Sc. degree in electrical and computer engineering from CPE-Lyon, France, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering with The University of Utah. His research interests include programmable architectures, low power digital designs, and emerging technologies.

Giovanni De Micheli (F'94) received the Nuclear Engineer degree from the Politecnico di Milano in 1979, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley in 1980 and 1983, respectively. He was a Professor of electrical engineering with Stanford University. He is currently a Professor and the Director of the Institute of Electrical Engineering and the Integrated Systems Centre, EPF Lausanne, Switzerland. He is a Program Leader of the Nano-Tera.ch Program.

He has authored the book entitled *Synthesis and Optimization of Digital Circuits* (McGraw-Hill, 1994), and co-authored or co-edited eight other books and over 800 technical articles. His research interests include several aspects of design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips, 3-D integration, heterogeneous platform design, including electrical components and biosensors, and data processing of biomedical information. He is a fellow of the ACM, a member of the Academia Europaea, and an International Honorary Member of the American Academy of Arts and Sciences. He is a member of the Scientific Advisory Board of imec (Leuven, B), CfAED (Dresden, D), and STMicroelectronics. His citation h-index is 93 according to Google Scholar.

Dr. De Micheli was a recipient of the 2016 IEEE/CS Harry Goode Award for seminal contributions to design and design tools of networks on chips, the 2016 EDAA Lifetime Achievement Award, the 2012 IEEE/CAS Mac Van Valkenburg Award for contributions to theory, practice, and experimentation in design methods and tools, and the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems, the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000, the D. Pederson Award for the best paper on the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS in 1987 and 2018, and several best paper awards, including DAC in 1983 and 1993, DATE in 2005, Nanoarch in 2010 and 2012, and Mobihealth in 2016.

He has been serving the IEEE in several capacities, namely, the Division 1 Director from 2008–2009, a Co-Founder and the President Elect of the IEEE Council on EDA from 2005 to 2007, the President of the IEEE CAS Society in 2003, and the Editor in Chief of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS from 1997 to 2001. He has been the chair of several conferences, including Memocode in 2014, DATE in 2010, pHealth in 2006, VLSI SOC in 2006, DAC in 2000, and ICCD in 1989.

Pierre-Emmanuel Gaillardon (S'10–M'11–SM'16) received the Electrical Engineering degree from CPE-Lyon, France, in 2008, the M.Sc. degree in electrical engineering from INSA Lyon, France, in 2008, and the Ph.D. degree in electrical engineering from CEA-LETI, Grenoble, France, and the University of Lyon, France, in 2011. He is currently an Assistant Professor with the Electrical and Computer Engineering Department and an Adjunct Assistant Professor with the School of Computing, The University of Utah, Salt Lake City, UT, USA, where he leads the Laboratory for NanoIntegrated Systems. Prior to joining The University of Utah, he was a Research Associate with the Swiss Federal Institute of Technology, Lausanne, Switzerland, within the Laboratory of Integrated Systems and a Visiting Research Associate with Stanford University, Palo Alto, CA, USA. Previously, he was a Research Assistant with CEA-LETI. He was a recipient of the C-Innov 2011 Best Thesis Award, the Nanoarch 2012 Best Paper Award, the BSF 2017 Prof. Pazy Memorial Research Award, the 2017 NSF CAREER Award, and the 2018 IEEE CEDA Pederson Award.

His research activities and interests include the development of novel computing systems exploiting emerging device technologies and novel EDA techniques. He has been serving as a TPC member for many conferences, including DATE'15-19, DAC'16-18, and Nanoarch'12-17. He is a reviewer for several journals and funding agencies. He served as a Topic Co-Chair of Emerging Technologies for Future Memories for DATE'17-19.