A RRAM-based FPGA for Energy-efficient Edge Computing

Xifan Tang, Edouard Giacomin, Patsy Cadareanu, Ganesh Gore and Pierre-Emmanuel Gaillardon
Electrical and Computer Engineering, University of Utah, Salt Lake City, Utah, U.S.A.
xifan.tang@utah.edu

***Abstract*—The shift from centralized cloud to edge computing demands hardware systems with data processing capability at ultra-low power. Reconfigurable solutions such as *Field-Programmable Gate Arrays* (FPGAs) offer a high flexibility in terms of hardware implementation and are thus popular for use in many edge computing systems. However, breaking through the energy wall of FPGAs is a challenge, as low-power operation often requires compromising performances. In this paper, we study a low-power high-performance FPGA architecture exploiting *Resistive Random Access Memory* (RRAM) technology. To perform a comprehensive analysis, we introduce a novel design flow which can rapidly prototype FPGA fabrics from which accurate area, delay, and power results can be obtained. Based on full-chip layouts and SPICE simulations, we show that RRAM-based FPGAs can improve up to 8%/22%/16% in area/delay/power compared to SRAM-based counterparts at nominal voltage. Even when operated at a near-V_t supply, the proposed RRAM-based FPGA can improve the *Energy-Delay Product* by about 2 $\times$ without any delay overhead, when compared to an SRAM-based FPGA. In addition, Monte Carlo simulations showed that the proposed RRAM-based FPGA architecture stays robust under different CMOS process corners as well as under a 30% RRAM resistance standard deviation.**

***Index Terms*—Field-programmable gate arrays; Resistive memories; Low-power design**

I. Introduction

Advancements in *Artificial Intelligence* (AI) drive the use of edge computing for *Internet-of-Thing* (IoT) applications, which requires specialized hardware systems to be more capable in data processing under an ultra-low power budget [1]. Reconfigurable systems such as *Field-Programmable Gate Arrays* (FPGAs) have been a ubiquitous media in many edge computing systems, thanks to their flexibility in hardware implementation. However, energy efficiency has become a severe barrier for deploying FPGAs in a large set of IoT applications. To break the energy wall, two major challenges have to be resolved: (i) First, the programmable routing architecture which accounts for about 70% of the area, 80% of the delay, and 60% of the power of the whole chip [2], [3], is preventing them from achieving ultra-low energy efficiency; (ii) Second, FPGAs suffer from significant delay degradation at low voltages (up to 2$\times$). As such, low-power FPGAs are failing to meet the computing requirements on edge computing [4], [5].

Thanks to their non-volatile memory storage capabilities, their higher integration density, and their low power consumption, the *Resistive Random Access Memory* (RRAM) technology has opened the door to ultra-low-power FPGA technologies [6]–[12]. A RRAM device operates as a reconfigurable resistor which can be switched from a *High Resistance State* (HRS) to a *Low Resistance State* (LRS) and vice versa, based on a combination of programming voltage and current polarization. As a non-volatile memory technology, RRAM can guarantee zero leakage power for FPGAs when operating in sleep mode [6]. As shown in Fig. 1, this allows FPGAs to be fully switched off between operating periods without budgeting time and energy for wake-up. Besides, major works studied novel programmable switches in the purpose of replacing a *Static Random Access Memory* (SRAM) cell and a transmission-gate with a unique RRAM device [9]–[12]. Thanks to smaller parasitic resistance and capacitance, energy consumption of routing multiplexers can be significantly reduced by $4.7\times$ [15], [16]. As routing multiplexers are a dominant component in FPGA fabrics, RRAM-based FPGAs can potentially improve area by up to 15%, delay by up to 58% and power by up to 58%, when compared to their SRAM-based counterparts [9]–[12]. Previous works also proved that RRAM-based FPGAs are more energy efficient in the near-V_t regime without any performance loss, as the resistance of RRAM is independent from it voltage across [12].

In this paper, we study a low-power and high-performance FPGA architecture exploiting RRAM technology. To perform a comprehensive analysis, we introduce a novel design flow which can rapidly prototype FPGA fabrics, from which accurate area, delay, and power results can be obtained. Based on full-chip layouts and SPICE simulations, we show that RRAM-based FPGAs, when operating at nominal operating voltage, can improve by up to 8%/22%/16% the area, delay, and power respectively, when compared to their SRAM-based counterparts. When operating at reduced supply voltage regime, the proposed RRAM-based FPGAs can improve the *Energy-Delay Product* by about 2 $\times$ without any delay overhead, when compared to SRAM-based FPGAs operating at nominal voltage.

This material is based on research sponsored by Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-18-2-7855. 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 Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) or the U.S. Government.
Pierre-Emmanuel Gaillardon, Ganesh Gore and Xifan Tang have financial interests in the company ReRouting LLC, which manufactures RRAM-based systems and provides engineering service.

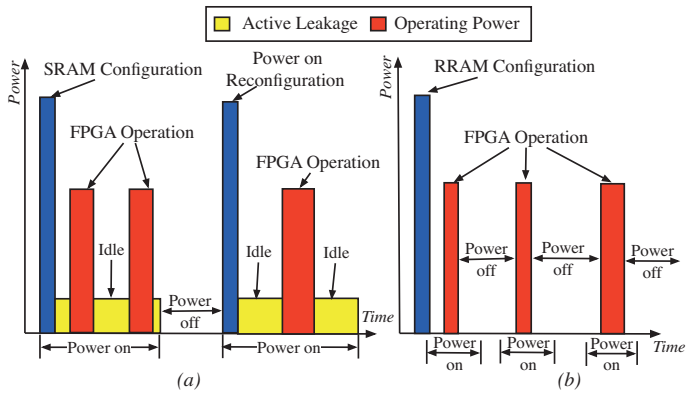

Fig. 1: Power consumption of (a) a SRAM-based FPGA and (b) a RRAM-based FPGA.

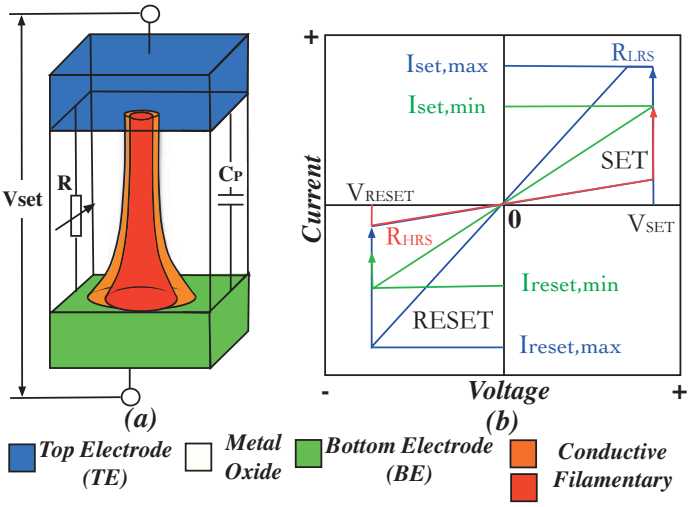

Fig. 2: RRAM structure: (a) Size of filaments inside a RRAM achieved by $I_{set,min}$ (red) or $I_{set,max}$ (orange); (b) I-V characteristics of a BRS RRAM.

Process corner analyses on a full FPGA fabric validated the robustness of the proposed RRAM-based FPGA architecture, leading to merely 2% shift on performance and 8% shift on energy consumption. Monto Carlo simulations showed that the proposed RRAM-based FPGAs can tolerate up to 30% three-sigma standard deviation on RRAM devices.

The rest of this paper is organized as follows: Section II introduces the necessary background knowledge about RRAM technology and FPGA architectures. Section III presents the architectural details of the proposed RRAM-based FPGAs. Section IV explains the fast prototyping tools developed for RRAM-based FPGAs. Section V presents a comprehensive architecture-level analysis. Section VI concludes this paper.

II. Background

In this section, we first review RRAM technology and then discuss about current state-of-the-art FPGA architectures.

A. RRAM Technology

RRAM is a promising emerging non-volatile memory technology [13], typically consisting of three layers: a *Top Electrode* (TE), a transition metal oxide material stack and a *Bottom Electrode* (BE), as seen in Fig. 2 (a) [14]. RRAM boasts low power consumption, high-speed operation, high-density integration, and CMOS process compatibility. The latter two of these benefits are due specifically to its compatibility with *Back-End-of-the-Line* (BEOL) processing such that RRAM can be fabricated anywhere between two metal layers, without occupying transistor area. The metal-oxide-metal structure facilitates an abrupt switching event in the oxide layer from insulating i.e., the *High Resistance State* (HRS) to conductive i.e., the *Low Resistance State* (LRS). This occurs by applying a programming voltage across the TE and BE after the initial formation of a conductive path in the oxide between the electrodes called the filament, as depicted in Fig. 2.

The switching event from LRS to HRS is called a set process, while the reverse event is called a reset process. In this paper, we consider RRAM based on *Bipolar Resistive Switching* (BRS) only, which is a common choice for most RRAM-based circuits and systems [6]–[12]. Fig. 2 (b) illustrates the I-V characteristics of a BRS RRAM. The minimum programming voltages required to trigger the set and reset processes are defined as V_{set} and V_{reset}, respectively. For fresh samples, a voltage larger than V_{set} is used to form the filament once and trigger the resistive switching behavior. The programming currents 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; this is denoted by $I_{set,max}$ in Fig. 2 (b). The programming current tunes the size of filaments, leading to a difference in the resistance of a RRAM in LRS, R_{LRS}. This is seen in Fig. 2 (a), where the filament highlighted in orange leads to a lower R_{LRS} than the filament highlighted in red. To read the data from the RRAM cell, a small read voltage, V_{read} is applied which doesn't affect the state of the memory cell.

Overall, FPGA architecture does not require particularly stringent RRAM parameters. RRAMs have the capacity for two functionalities in FPGAs: (1) as replacements for transmission gates in the data path of routing multiplexers, and (2) as standalone memories in flip-flops. The former requires the RRAM to have a high R_{off}/R_{on} ratio ($> 10^3$) to limit parasitic leakage and input crosstalk. The latter is not as rigid in its R_{on} and R_{off}/R_{on} ratio requirements as in regular memory applications. In this paper, we will consider a typical RRAM technology used in previous works [30], where $V_{set,max} = |V_{reset,max}| = 1.1$ V, $I_{set,max} = |I_{reset,max}| = 500\ \mu A$, lowest achievable $R_{LRS} = 2.2\ k\Omega$ and highest achievable $R_{RHS} = 20\ M\Omega$. Different from SRAM-based FPGAs, RRAM-based FPGAs do not need frequent reconfiguration even when their deployment require frequent power-off (see the example in Fig. 1). As a result, only a endurance of $\sim 10^4$ is required for RRAM write operations. Note that the relaxed demand on endurance allow the RRAMs to tolerate a higher R_{off}/R_{on} than typical range. However, RRAM-based FPGAs require long retention periods ($\sim$ 10 years @ $85°C$) because the programmed FPGAs need to hold their configurations, but fast programming is not required (write speed can be relaxed to 100 ns). The requirements explained here will be used in the rest of this paper.

More details about RRAM technology can be found in [14].

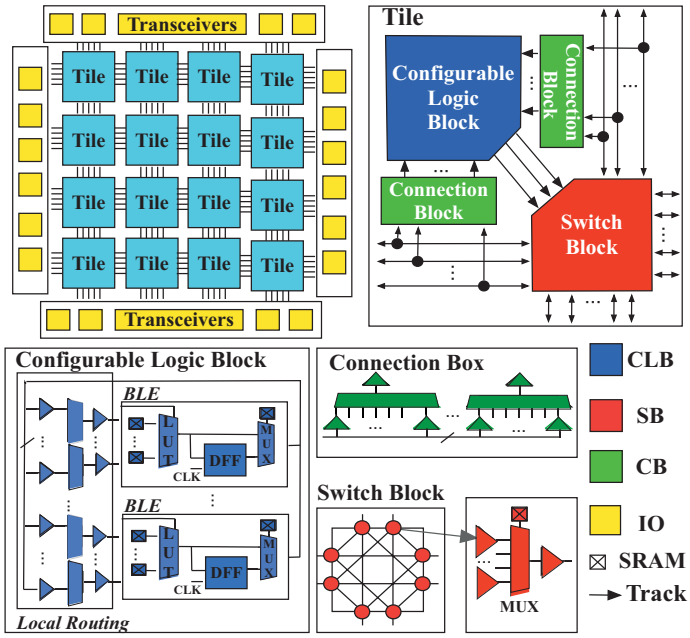

Fig. 3: General FPGA architecture.

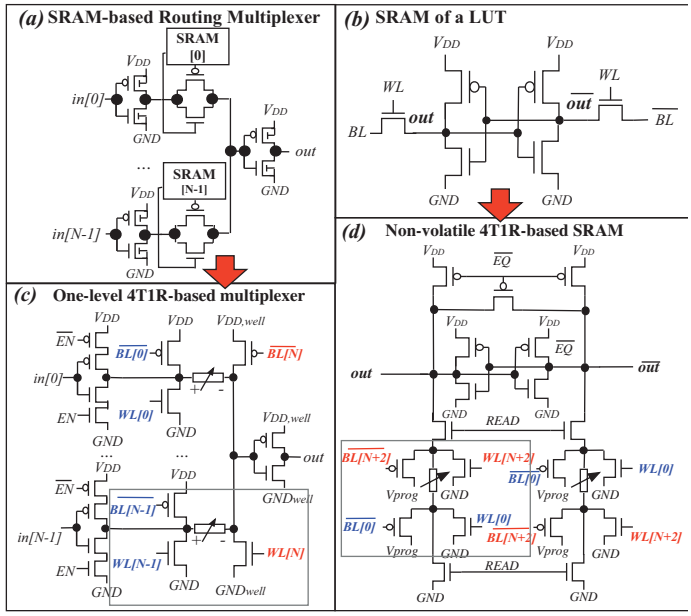

Fig. 4: Circuit designs of: (a) SRAM-based routing multiplexer; (b) 4T1R-based routing multiplexer; (c) 6T SRAM cell; (d) Non-volatile 4T1R-based SRAM.

B. Related Works on FPGA Architecture

Fig. 3 illustrates a fundamental FPGA fabric, which is built with an array of repeatable tiles surrounded by IO blocks. Each tile consists of a *Configurable Logic Block* (CLB), two *Connection Blocks* (CBs), and a *Switch Block* (SB) [18]. CBs connect routing tracks to the CLB input pins, while SBs provide inter-tile interconnection between the CLB output pins and the routing tracks. In each CLB, there are a number of *Basic Logic Elements* (BLEs) which are interconnected by a dense local routing architecture. Each 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. Based on different application need, commercial FPGAs may adopt fracturable LUTs and hard carry chains in CLBs, and also replace columns of tiles with heterogeneous blocks [19]–[21]. In this paper, we aim to capture the difference between SRAM-based and RRAM-based FPGAs. Without loss of generality, our evaluations consider the homogeneous tile-based FPGA architecture shown in Fig. 3.

III. Proposed RRAM-based FPGA Architecture

The RRAM-based FPGA proposed here has no main architectural difference with respect to the general FPGA depicted in Fig. 3. To achieve non-volatility, SRAM-based primitive blocks are replaced by RRAM-based circuits, as illustrated in Fig. 4. To leverage the performance of RRAM-based circuits, we apply two different strategies when replacing the SRAMs of routing multiplexers or LUTs.

a) ***Routing multiplexer****:* We borrow the 4T1R-based routing multiplexer designs from [15] to replace the SRAM-based routing multiplexers, as illustrated in Fig. 4 (a) and (c). By replacing both SRAMs and transmission-gates, 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 by efficiently sharing programming transistors, the 4T1R-based routing multiplexers outperform SRAM-based counterparts at nominal voltage by 28% in area, 34% in delay, 30% in power. When operating at a near-V_T supply voltage a 4.7× energy consumption benefit was acquired [15]. Note that the endurance limit of RRAM devices will not be challenged by such replacement, due to the fact that programming operation for 4T1R-based multiplexers occurs only during FPGA reconfiguration, which is infrequent.

b) ***LUTs****:* The multiplexers in LUTs are still implemented by CMOS transistors and only the SRAMs of LUTs are replaced by RRAM-based non-volatile SRAM circuitry, as illustrated in Fig. 4 (d). There are two reasons why RRAM-use is avoided in the datapath of LUTs: (1) RRAMs in the datapath will be frequently switched between two resistance states. Compared to CMOS transistors, RRAM programming is typically much slower $>$ 10ns and thus drastically limiting the operating speed of LUTs. (2) the frequent switching of the RRAM LUTs is far beyond RRAM endurance. Therefore, RRAMs are used in only SRAMs of LUTs to grant non-volatility.

IV. OpenFPGA: An Opensource FPGA IP Generator

To conduct a comprehensive analysis on the proposed FPGAs, we adapt the open-source tool OpenFPGA [22], which is an FPGA IP generator designed for SRAM-based FPGAs. As shown in Fig. 5, our design flow adds SPICE and Verilog backends to the traditional VPR-based FPGA EDA flow [23]. In this paper, we extend this idea to support the proposed RRAM-based FPGA architecture. Rather than using the analytical results produced by Yosys [24] and VPR [23], our flow enables more realistic area, delay and power analyses drawn by:

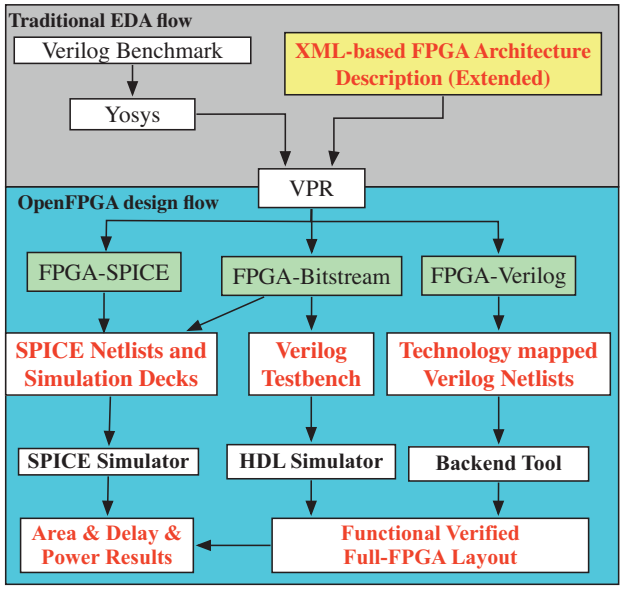

Fig. 5: OpenFPGA flow adapted for RRAM-based and SRAM-based FPGA architecture evaluation.

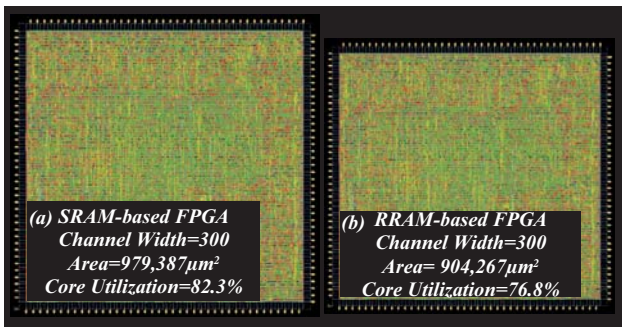

Fig. 6: Full-chip layouts (Channel width is set to 300) of FPGAs: (a) SRAM-based and (b) RRAM-based.

*a) **Full-chip layout generation**:* Thanks to FPGA-Verilog, we can employ a semi-custom design flow to prototype RRAM-based FPGA architectures. Using this design flow, fabrication-ready layout of a medium-sized FPGA fabric can be achieved in less than 24 hours [22]. Accurate area analysis can be performed by industrial physical design tools. In addition, the layout can be fully verified by Verilog testbenches which are automatically generated by OpenFPGA.

b) Full-fabric SPICE simulation: We enhanced FPGA-SPICE [25] to output SPICE netlists for the full fabric as well as each component in a FPGA, i.e., LUTs, FFs and multiplexers. Accurate timing results are extracted from SPICE simulations and then back-annotated to the timing analysis engine in VPR to estimate accurate critical path delays. By loading the bitstream to full-fabric SPICE simulation, accurate power analysis can be achieved for FPGAs configured to different benchmarks.

V. Experimental Results

In this section, we first introduce our experimental methodology and then perform a comprehensive analysis on the area, delay and power of the proposed FPGAs.

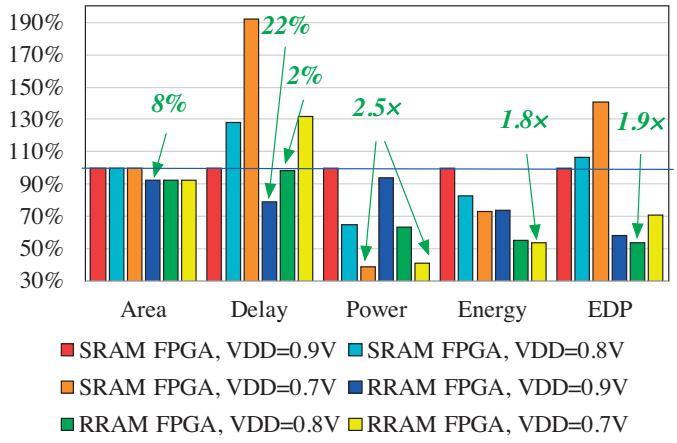

Fig. 7: Area, delay and energy comparison between SRAM-based and RRAM-based FPGAs operating at nominal and near-V_t regime.

A. Evaluation Methodology

To provide a fair comparison, both SRAM-based and RRAM-based FPGAs employ a popular and well-optimized FPGA architecture using a commercial 40 nm technology modeled by the VTR project [23][1]. To guarantee the best overall performance, CMOS multiplexers in local routing architecture and CBs adopt a two-level structure while the others are built with a one-level structure [26]. All the RRAM-based multiplexers adopt a one-level structure for optimal performance. In our analysis, RRAMs are placed between the first and the second metal layer to be close to the transistors and minimize interconnect parasitics [27]. We exploit the OpenFPGA flow in Fig. 5 to compare the area, delay and power of SRAM-based and RRAM-based FPGAs. The twenty largest MCNC benchmarks [28] are selected as the input of the EDA flow. Full fabric layout is implemented using Cadence Innovus 17.1, while delay and power analysis are performed by using Synopsys HSPICE 2017.03.

B. Layout Area Comparison

Fig. 6 compares the full-chip layouts of SRAM-based and RRAM-based FPGAs, both of which include programmable fabric, configuring peripherals and I/Os. Note that both FPGA fabric adopt a routing channel width of 300, being similar to commercial FPGAs [19], [20]. For sake of the capability of our workstation (256GB memory), we considered an array size of 5×5 for the programmable fabrics and 160 I/O pads. Considering that FPGAs are assembled by repeated tiles, we believe that a 5×5 fabric is representative to draw general conclusions. The full-chip layouts show that RRAM-based FPGAs consume 8% smaller area than the SRAM-based FPGA counterpart. This is mainly due to the BEOL integration of RRAM and design optimizations in RRAM-based routing multiplexers.

[1]Available at https://github.com/verilog-to-routing/vtr-verilog-to-routing/blob/master/vtr_flow/arch/timing/k6_N10_40nm.xml

C. Delay and Energy efficiency

Fig. 7 compares the delay and energy of the proposed RRAM and SRAM-based FPGAs. When operating at the same nominal $V_{DD} = 0.9V$, RRAM-based FPGAs improve on average 22% in delay and 16% in power against their SRAM-based counterparts. This performance gain comes from the delay and power efficiency of RRAM-based routing multiplexers. More opportunities lie in the near-V_t regime for RRAM-based FPGAs. When V_{DD} is reduced to near-V_t regime, i.e., $0.8V$, RRAM-based FPGA remains at the same performance-level as the SRAM-based FPGA at nominal voltage, while achieving an $1.8\times$ energy reduction. This is due to the resistance of RRAMs being independent from the working voltage, unlike transistors whose equivalent resistance degrades seriously at near-V_t regime. This is an important feature of RRAM-based FPGAs, showing their strong potential in edge computing applications.

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	$6.3k\Omega$	$14M\Omega$

D. Impact of RRAM Variations

Process variation is a major challenge for RRAM-based circuits, considering the stochastic nature of filamentary conduction mechanism of RRAMs [29]. In this section, we used electrical simulations to evaluate the robustness of the proposed RRAM-based FPGAs under both CMOS and RRAM variations. For the CMOS technology, we consider three process corners provided by the considered commercial 40nm technology: *Fast-Fast* (FF), *Typical-Typical* (TT), and *Slow-Slow* (SS). For the RRAM technology, three process corners called *Best*, *Typical* and *Worst* are developed by assuming variations on R_{LRS} and R_{HRS}. As detailed in Table I, for both corner cases and monte carlo simulations, we considered a typical three-sigma standard deviation of 30% the nominal resistance, as experimentally reported in [29]. In the *Typical* case, nominal R_{LRS} and R_{HRS} are considered as introduced in Section II-A. The Best case assumes the high-performance and low-leakage corner, while the Worst case assumes the low-performance and high-leakage corner.

a) Corner Analyses: In this part, we focused on studying the impact of process corners on the FPGA delay and energy. To be representative without losing generality, we showcased the MCNC *s298* benchmark. As shown in Fig. 8 (a) and (b), variations on CMOS can negatively impact RRAM-based FPGAs with serious degradation (up to 20% in delay and 50% in energy). However, RRAM variations have limited impacts where delay is only impacted by $< 3\%$ and the energy shift is within 8%. This can be explained from two aspects: (a) the impact of RRAM variations is limited on RRAM-based circuits. As illustrated in Fig. 4 (a) and (c), RRAM circuits

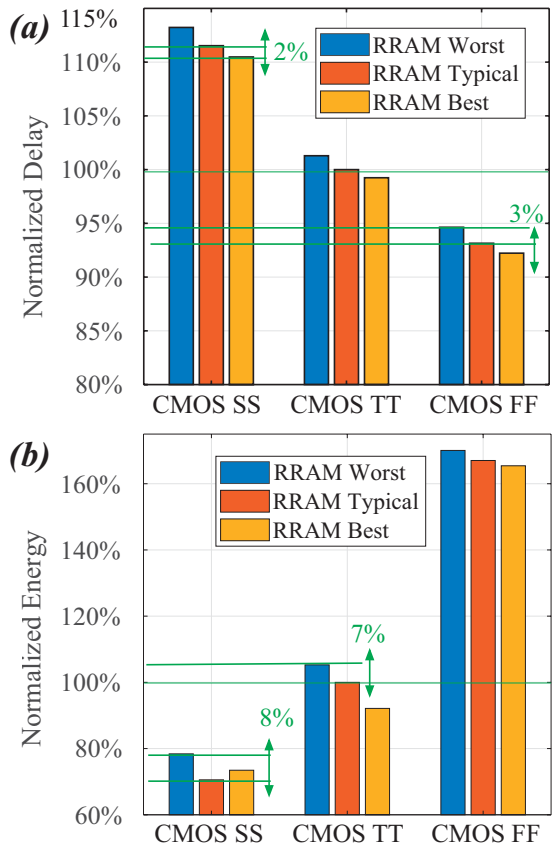

Fig. 8: Case study on benchmark *s298*: Impact of RRAM and CMOS corners on the RRAM-based FPGA operating at $V_{DD} = 0.9V$: (a) delay and (b) energy.

contain a considerable amount of CMOS transistors on their datapaths. As a result, the resistance of RRAMs stay as a small factor in the delay and energy characteristics; (b) the proposed FPGA architectures still employs many pure CMOS circuits, such as in LUTs and FFs, which are not impacted by RRAM variations.

b) Monte Carlo Analysis: In practice, corner cases may rarely happen but each RRAM ends up having an independent variation. To capture such cycle-to-cycle variation, we performed a 100-run Monte-Carlo SPICE simulation on the same *s298* benchmark used for our corner analyses. Fig. 9 illustrates the resulting delay and energy distributions, indicating that at the architectural-level, the variation on delay and energy may be fully mitigated. Some routing multiplexers may benefit from performance improvements from a decrease in R_{LRS}, while others may degrade due to an increase in R_{LRS}. Similarly, some routing multiplexers may suffer from an energy overhead from a decrease in R_{HRS}, while others may benefit from an energy reduction due to an increase in R_{HRS}.

VI. Conclusion

In this paper, we studied a low-power and high-performance FPGA architecture exploiting RRAM technology. To perform a comprehensive analysis, we modified the OpenFPGA flow to support RRAM-based FPGA architectures. Based on full-

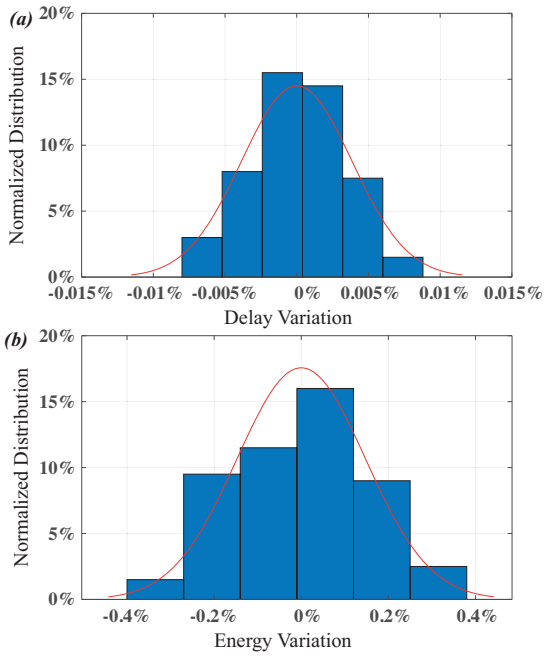

Fig. 9: Monto-Carlo results for benchmark *s298*: distribution of (a) delay and (b) energy under the impact from RRAM variation.

chip layouts and SPICE simulations, we showed that RRAM-based FPGAs, when operating at nominal operating voltage can improve up to 8%/22%/16% in area/delay/power, when compared to their SRAM-based counterparts. When operated close to the near-V_t regime, the proposed RRAM-based FPGAs can outperform by about 2× in energy consumption without delay overhead, against an SRAM-based FPGA operating at nominal voltage. Worse case process corner analysis on a full FPGA fabric validated the robustness of the proposed RRAM-based FPGA architecture, resulting to merely 2% shift on performance and 8% shift on energy consumption. Monto Carlo simulations presented that proposed RRAM-based FPGA can tolerate up to 30% variation on RRAM devices.

REFERENCES

[1] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile Edge Computing: A Survey," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450-465, Feb. 2018.

[2] M. Lin, A. E. Gamal, Y. C. Lu, and S. Wong, *Performance Benefits of Monolithically Stacked 3-d FPGA*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 2, pp. 216–229, Feb 2007.

[3] I. Kuon and J. Rose, *Quantifying and Exploring the Gap Between FPGAsand ASICs*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[4] T. Tuan, A. Rahman, S. Das, S. Trimberger, and S. Kao, *A 90-nm Low-power FPGA for Battery-powered Applications*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 2, pp. 296–300, Feb 2007.

[5] B. H. Calhoun, J. F. Ryan, S. Khanna, M. Putic, and J. Lach, *Flexible Circuits and Architectures for Ultralow Power*, Proceedings of the IEEE, vol. 98, no. 2, pp. 267–282, Feb 2010.

[6] O. Turkyilmaz *et al.*, *RRAM-based FPGA for "Normally off, Instantly on" Applications*, 2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Amsterdam, 2012, pp. 101-108.

[7] Y. Chen, W. Wang, H. Li and W. Zhang, *Non-volatile 3D Stacking RRAM-based FPGA*, 22nd International Conference on Field Programmable Logic and Applications (FPL), Oslo, 2012, pp. 367-372.

[8] K. Huang, R. Zhao, W. He and Y. Lian, *High-Density and High-Reliability Nonvolatile Field-Programmable Gate Array With Stacked 1D2R RRAM Array*, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 1, pp. 139-150, Jan. 2016.

[9] S. Tanachutiwat, M. Liu and W. Wang, *FPGA Based on Integration of CMOS and RRAM*, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 11, pp. 2023-2032, Nov. 2011.

[10] J. Cong and B. Xiao, *FPGA-RPI: A Novel FPGA Architecture With RRAM-Based Programmable Interconnects*, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 4, pp. 864-877, April 2014.

[11] P.-E. Gaillardon, D. Sacchetto, S. Bobba, Y. Leblebici and G. De Micheli, *GMS: Generic Memristive Structure for Non-volatile FPGAs*, IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC), Santa Cruz, CA, USA, 2012, pp. 94-98.

[12] X. Tang, P.-E. Gaillardon and G. De Micheli, *A high-performance Low-power Near-Vt RRAM-based FPGA*, International Conference on Field-Programmable Technology (FPT), Shanghai, 2014, pp. 207-214.

[13] G. Burr *et al.*, *Overview of Candidate Device Technologies for Storage-Class Memory,* IBM Journal of Research and Development, vol. 52, p. 449-464 2008, July 2008

[14] H. S. P. Wong *et al.*, *Metal-Oxide RRAM*, Proceedings of the IEEE, vol. 100, no. 6, pp. 1951–1970, June 2012.

[15] 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*, in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 5, pp. 1173-1186, May 2017.

[16] X. Tang, E. Giacomin, G. De Micheli, and P.-E. Gaillardon. 2017. Physical Design Considerations of One-level RRAM-based Routing Multiplexers. In Proceedings of the 2017 ACM on International Symposium on Physical Design (ISPD '17). ACM, New York, NY, USA, 47-54.

[17] X. Tang, G. Kim, P.-E. Gaillardon and G. De Micheli, *A Study on the Programming Structures for RRAM-Based FPGA Architectures*, in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 63, no. 4, pp. 503-516, April 2016.

[18] V. Betz *et al.*, *Architecture and CAD for Deep-Submicron FPGAs*, Kluwer Academic Publishers, 1999.

[19] Altera Corporation, *Stratix 10 Advance Information Brief*, July 2015.

[20] Xilinx Corporation, *Virtex-7 User Guide DS180 (v1.17)*, May 2015.

[21] M. Hutton *et al.*, *Improving FPGA Performance and Area Using an Adaptive Logic Module*, FPL, 2004, pp. 135-144.

[22] X. Tang, E. Giacomin, A. Alacchi, B. Chauviere and P. Gaillardon, *OpenFPGA: An Opensource Framework Enabling Rapid Prototyping of Customizable FPGAs*, 29th International Conference on Field Programmable Logic and Applications (FPL), Barcelona, Spain, 2019, pp. 367-374.

[23] J. Luu *et al.*, *VTR 7.0: Next generation architecture and CAD system for FPGAs*, ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 7, no. 2, June 2014.

[24] Yosys Open Synthesis Suite, available at https://github.com/YosysHQ/yosys

[25] X. Tang, E. Giacomin, G. D. Micheli and P. Gaillardon, *FPGA-SPICE: A Simulation-Based Architecture Evaluation Framework for FPGAs*, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 3, pp. 637-650, March 2019.

[26] C. Chiasson and V. Betz, *COFFE: Fully-automated transistor sizing for FPGAs*, International Conference on Field-Programmable Technology (FPT), Kyoto, 2013, pp. 34-41.

[27] E. Giacomin and P.-E. Gaillardon, *A Resistive Random Access Memory Addon for the NCSU FreePDK 45 nm*," in IEEE Transactions on Nanotechnology, vol. 18, pp. 68-72, 2019.

[28] S. Yang, *Logic Synthesis and Optimization Benchmarks User Guide Version 3.0*, MCNC, Jan. 1991.

[29] A. Grossi *et al.*, *Experimental Investigation of 4-kb RRAM Arrays Programming Conditions Suitable for TCAM*, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 12, pp. 2599-2607, Dec. 2018.

[30] 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*, IEEE International Electron Devices Meeting, San Francisco, CA, 2008, pp. 1-4.