A Ultra-Low-Power FPGA Based on Monolithically Integrated RRAMs

Pierre-Emmanuel Gaillardon, Xifan Tang, Jury Sandrini, Maxime Thammasack, Somayyeh Rahimian Omam, Davide Sacchetto, Yusuf Leblebici, Giovanni De Micheli
Swiss Federal Institute of Technology
Lausanne, Switzerland
pierre-emmanuel.gaillardon@epfl.ch

***Abstract*—Field Programmable Gate Arrays (FPGAs) rely heavily on complex routing architectures. The routing structures use programmable switches and account for a significant share in the total area, delay and power consumption numbers. With the ability of being monolithically integrated with CMOS chips, *Resistive Random Access Memories* (RRAMs) enable high-performance routing architectures through the replacement of *Static Random Access Memory* (SRAM)-based programming switches. Exploiting the very low *on*-resistance state achievable by RRAMs as well as the improved tolerance to power supply reduction, RRAM-based routing multiplexers can be used to significantly reduce the power consumption of FPGA systems with no performance compromises. By evaluating the opportunities of ultra-low-power RRAM-based FPGAs at the system level, we see an improvement of 12%, 26% and 81% in area, delay and power consumption at a mature technology node.**

I. Introduction

Static Random Access Memory (SRAM)-based *Field Programmable Gate Arrays* (FPGAs) are more flexible than *Application-Specific Integrated Circuits* (ASICs) at the cost of 20× bigger area, 4× longer delay, and 12× higher power consumption approximately [1]. The drawbacks of FPGAs lie in the expensive routing architecture, which accounts for about 70% of the area, 80% of the delay and 60% of the power of the entire chip [2]. Power consumption is a serious barrier for the distribution of FPGAs in a large set of consumer applications, i.e., *Ultra-Low Power* (ULP) *System-on-Chip* (SoCs). Previous works [3,4,5] demonstrate low-power SRAM-based FPGA designs where a low supply voltage is employed to save up to 50% of the power consumption. However, low-power SRAM-based FPGAs generally suffer from large delay degradation (up to 2×).

The recent development of *Resistive Random Access Memories* (RRAMs) opens opportunities in advancing the FPGA technology with high density, performance and excellent energy efficiency. Typically, around 40% of the transistor area in SRAM-based FPGAs is occupied by configuration memories. However, RRAMs can be monolithically fabricated within the *Back-End-of-Line* (BEoL) metal lines. This allows us to move the configuration memories onto the top of the transistors, thereby increasing the integration density, and to shorten the metal interconnections. Furthermore, SRAM-based FPGAs have to be configured every time the system is powered *on* due to the volatility of SRAMs. Overwhelming *Static Random Access Memories* (SRAMs) intrinsically, RRAMs hold storage when powered down and consume zero leakage power in sleep mode. Using RRAMs as standalone memories, FPGAs can benefit a ~50% power reduction from instant power-*on* and normal power-*off*, compared to SRAM-based counterparts [6]. Furthermore, RRAMs motivate the exploration of novel FPGA architectures whose routing structures are directly employing RRAMs in the data path. In the novel architectures, RRAMs play the role of both configurable memories and programmable switches. The *Low-Resistance State* (LRS) of RRAMs provides down to 75% lower *on*-resistance than pass transistors, and thus reduces the delay of critical path. Finally, the novel routing elements demonstrate very good properties under reduced power supply conditions. Indeed, RRAM-based multiplexers can operate at near-V_t with significant power reduction for almost no performance compromise. Such blocks can be exploited advantageously to design ultra-low-power FPGAs.

In this paper, we present (i) a ultra-low-power FPGA architecture working in the near-V_t regime and exploiting monolithically integrated RRAMs. To this purpose, we (ii) introduce a monolithic integration technique used to embed RRAMs within processed CMOS chips. We also (iii) describe in detail the circuit realization of non-volatile flip-flops and routing multiplexers. We finally (iv) run system-level benchmarking, demonstrating that the proposed approach can lead to a 12% area shrink, a 26% delay reduction and a 81% power improvements compared to a conventional FPGA design at similar technology node.

The remainder of the paper is organized as follows. In Section II, we report on the monolithic integration of resistive memories on top of already processed CMOS chips. In Section III, we introduce our RRAM-based FPGA architecture. In Section IV and Section V, we discuss in details the design of the two fundamental parts of the proposed architecture, respectively, the non-volatile routing circuits and the non-volatile flip-flops. In Section VI, we predict the performance of the proposed architecture through architectural simulations. In Section VII, we finally draw the conclusions of this paper.

II. RRAM-CMOS Monolithic Integration

As one of the most promising emerging *Non-Volatile* (NV) memories, the *Resistive Random Access Memory* (RRAM) technologies have been widely investigated [7]. RRAMs are two-terminal devices that typically consist of three layers: the top electrode, the switching metal oxide and the bottom

This work has been supported by the Swiss National Science Foundation under the project No. 200021-146600.

electrode. RRAMs can be programmed into two stable resistance state, a *Low Resistance State* (LRS) and a *High Resistance State* (HRS) respectively by modifying the conductivity of metal oxide. When a programming voltage is applied between the electrodes, the metal oxide sees a conductivity change which leads to the switch of the resistance states.

In this work, we employ a monolithic heterogeneous integration of RRAMs with standard CMOS technology by post-processing the BEoL of fully finished CMOS chips. The monolithic integration strategy consists in depositing a TaO_x thin film between the CMOS chip metal M6 and an intermediate metal layer between metals M5/M6, which is used to fabricate *Metal-Insulator-Metal* (MIM) capacitors. Fig. 1 shows a *Scanning Electron Micrograph* (SEM) of the MIM capacitor, which is electrically connected through VIAs to M6.

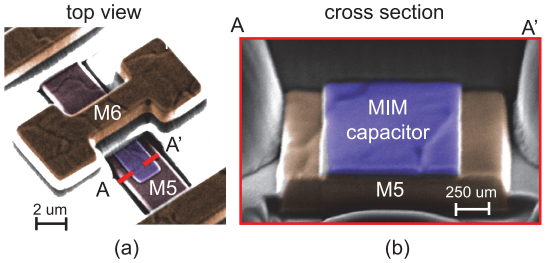

Fig. 1. (a) Scanning Electron Micrograph of the MIM capacitor. (b) FIB-SEM cross-section.

The main fabrication steps are shown in Fig. 2. After embedding the CMOS dies on a carrier wafer (Steps a-c), the chip passivation is opened down to M5 by wet etch and TaO_x is deposited by sputtering.

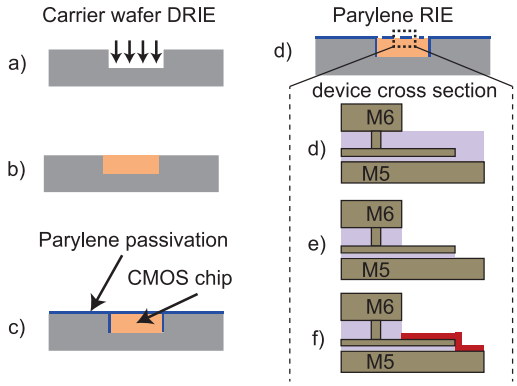

Fig. 2. Fabrication process steps for the RRAM-CMOS monolithic integration: a-Carrier wafer creation by Si deep reactive ion etch; b-CMOS chip assembly; c-Parylene coating; d-Parylene dry etch; e-Chip passivation opening; f-TaO_x sputtering.

DC electrical tests are reported in Fig. 3. After a voltage-forming step, low-voltage operation has been observed with a SET voltage of -1V and a RESET voltage of +1.3V. The 50nm-thick integrated RRAM has a LRS of 80Ω, and a HRS of 320Ω. Note that different resistance values, ratios, and programming voltages can be obtained by engineering the memory stack and the writing procedure [7].

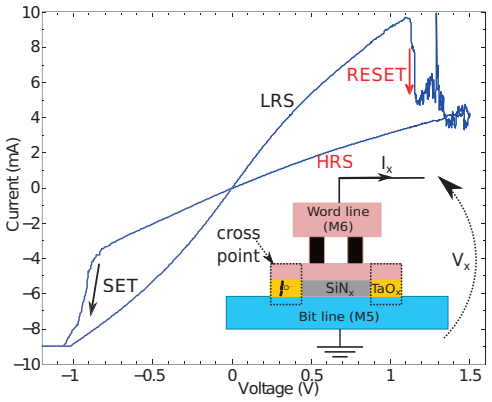

Fig. 3. Measured I-V curve of an integrated TaO_x-based RRAM device.

III. RRAM-Based FPGA Architecture

In this section, we discuss the overall architectural opportunities brought by RRAMs in the field of FPGAs. We first start with a description of the relevant previous art and we introduce our vision about the architecture.

A. RRAM-Based FPGAs: State-of-the-Art

FPGA architecture can benefit from the *non-volatility* as well as the area and performance gains coming from the monolithic integration and the low *on*-resistance values achieved by RRAMs. Previous works [8,9,10,11] propose novel FPGA architecture based on two principles: (a) the replacement of the SRAMs in *Look-Up Tables* (LUTs) with RRAMs, and (b) the replacement of the SRAMs as well as the transmission gates in routing architecture with RRAMs. Fig. 4 illustrates the early RRAM-based architectures where bi-directional routing architecture is employed. Previous works only investigate operations under standard working voltages, leaving neat-V_t RRAM-based FPGAs an open question.

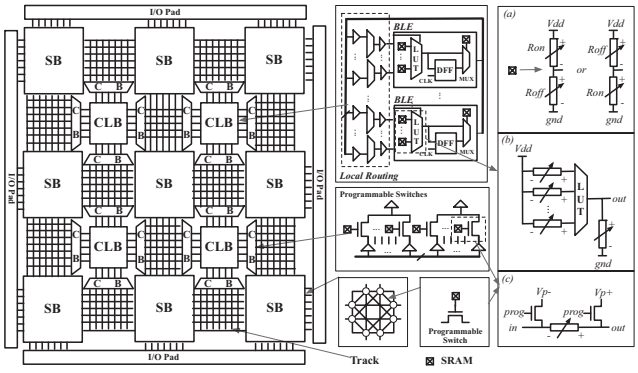

Fig. 4. Early RRAM-based FPGA architectures. (a) The SRAMs are replaced by RRAMs; (b) RRAM-based LUTs; (c) RRAM-based programmable switch.

B. RRAM-based FPGAs: Our Vision

The RRAM-based FPGA introduced in this work has no architectural difference with respect to a conventional SRAM-based FPGA. It remains an island-style FPGA where the cluster-based *Configurable Logic Blocks* (CLBs) are surrounded by *Switch Boxes* (SBs) and *Connection Blocks*

(CBs). The differences lie in the circuit design of those modules heavily relying on LUTs and multiplexers. Fig. 5 compares the circuit designs of LUT and multiplexer between a conventional SRAM-based FPGA and the RRAM-based FPGA introduced in this paper.

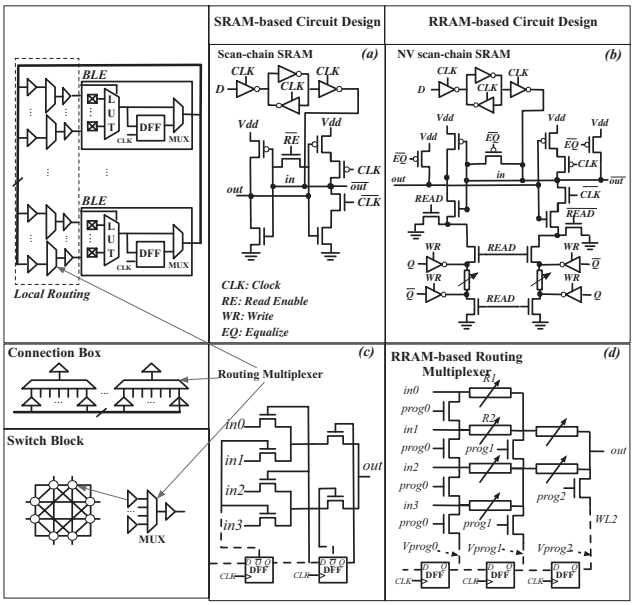

Fig. 5. Comparisons between a standard SRAM-based FPGA and the considered RRAM-based FPGA

In the proposed architecture, the logic elements exploit *Non-Volatile* (NV) LUTs. As other NV FPGAs, our structure does not need to be re-programmed during each power *on* and can benefit instant-*on* and normally-*off* properties [6]. Typically, a LUT consists of a bank of SRAMs and a multiplexer. The SRAM bank stores a truth table, which is decoded by the multiplexer, enabling LUT to realize any logic function. In this work, we replace the scan-chain SRAMs (Fig. 5-a) in LUTs with NV scan-chain SRAMs borrowed from previous work [14]. The multiplexers in LUTs are still implemented by transmission gates to avoid endurance limitations of the RRAMs. Compared to SRAM-based implementations, the proposed NV LUT has no difference in performance and power consumption because of the identical decoder implementation. Non-volatility is also introduced in the data path flip-flops using the same circuit elements. These FFs operate as standard volatile CMOS FF during regular operation but they are also capable to store on-demand non-volatile data before a sleep period. Data stored in the NV DFFs can then be restored during wake up. In these flip-flops, RRAMs are written only before the sleep period. These events have very low frequency and are compatible with the endurance capabilities of RRAMs. While supported by the presented architecture, instant-*on* and normally-*off* operation will not be evaluated in this paper. While the decoded paths of the LUT multiplexer change at runtime, the selected paths in the routing multiplexers, i.e., in BLE output selector, local routing, SBs and CBs, remain unchanged during the runtime. Therefore, RRAMs can be inserted in the data path of routing architecture without challenging the endurance. Fig. 5-d illustrates the RRAM-based multiplexer [13] that replaces the SRAM-based multiplexer shown in Fig. 5-c.

IV. High-Performance Non-Volatile Routing Multiplexers

As a fundamental block of our architecture proposal, we describe in this section a routing multiplexer circuit that exploits RRAMs directly in the logic data path. We also comment on the stability of the multiplexer while employed with a reduced power supply.

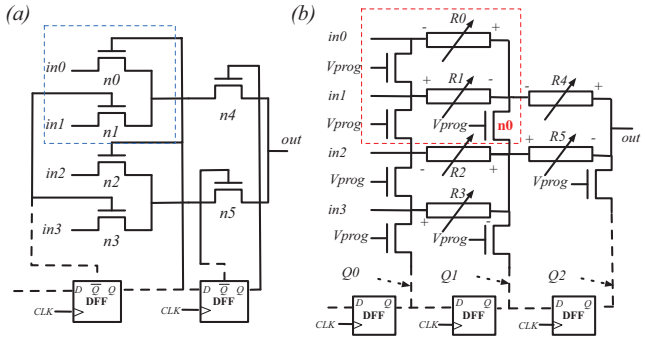

Fig. 6. 4-to-1 multiplexer structure exploiting (a) SRAMs and (b) RRAMs.

A. RRAM-Based Routing Multiplexer

RRAM-based routing multiplexers exploit the structure of multi-stage transmission-gate multiplexers, but use RRAMs instead of transistors for the data path routing. Fig. 6-a shows an example of SRAM-based 4 to 1 multiplexer built with three SRAM-based 2 to 1 multiplexers (one of them is highlighted in a rectangle). In a 2 to 1 multiplexer, the branches are always in different conduction state. Note that selection signals are permanently driven by SRAMs to ensure a constant path selection. Inspired by this structure, we propose a RRAM-based 2 to 1 multiplexer implemented with two RRAMs, as depicted in the rectangle box in Fig. 6-b. RRAM-based multiplexers take advantage of the *Bipolar Resistive Switching* (BRS) [7] in order to share programming transistors and achieve area-efficiency [13]. The positive terminal of the top device R_0 is connected the negative terminal of the bottom device R_1. The arrangement enables complementary programming of the two RRAMs, either R_0 in HRS, R_1 in LRS or R_0 in LRS, R_1 in HRS. Similar to SRAM-based implementation, when a RRAM is in HRS, the path is blocked and when a RRAM is in LRS, the path is propagating. The two RRAMs can be configured in one step by programming voltages using the scheme described in [13]. In Fig. 6-b, the 2 to 1 multiplexer brick is duplicated to realize a 4 to 1 routing multiplexer. Compared to the SRAM-based multiplexers, the RRAM-based multiplexers exhibit high performances accounted to the low *on*-resistance of the RRAMs introduced in the data path.

B. Energy/Delay Evolution at Near-V_T Power Supply

In conventional SRAM-based low-power FPGAs, a reduction of the supply voltage down to near/sub-V_t regime trades off power reduction with delay degradation. In RRAM-based FPGAs, the routing architectures exploit RRAMs in the data paths and perform differently when supply voltage changes. Fig. 7 compares the delay and power between a 32-input SRAM-based multiplexer and its RRAM-based counterpart when V_{dd} ranges from 0.4V to 1.8V. Both RRAM-

based and SRAM-based multiplexers reduce power but suffer from delay degradation when V_{dd} decreases. Generally, RRAM-based multiplexer consumes slightly more power than SRAM-based due to the low *on*-resistance of RRAMs in data paths. However, SRAM-based FPGA routing architecture suffers serious delay degradation when V_{dd} decreases. In contrast, RRAM-based FPGA routing architecture benefit the same power reduction but with very moderate delay degradation. The different trends in delay degradations are accounted to the low *on*-resistance of RRAMs which is achieved independently from V_{dd}, while the *on*-resistance of the transmission gates increases sharply when V_{dd} decreases. Furthermore, the parasitic capacitances brought by the programming transistors do not vary significantly until V_{dd} drops to sub-V_t regime. Therefore, the delay of RRAM-based multiplexer in near-V_t regime remains as they are at V_{dd}= 1.8V since its *RC* characteristic does not change. This opens large promises for FPGA power reduction with very limited delay compromises.

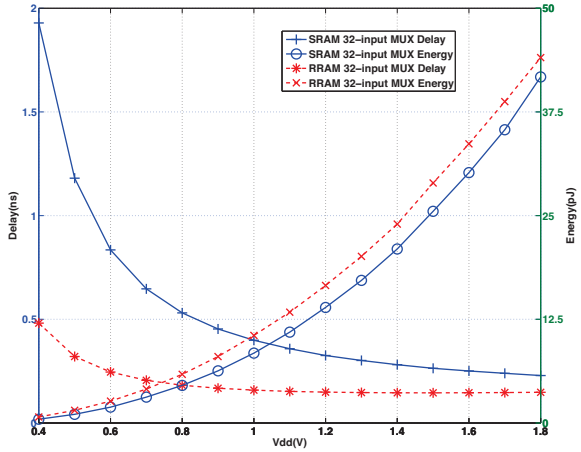

Fig. 7. Energy and delay evaluation of 32-input SRAM-based and RRAM-based multiplexers under V_{dd} reductions (Mature low-power 0.18µm technology is assumed. Nevertheless, similar conclusions can be draw with other technology nodes.)

C. Impact on the Routing Buffering

In standard FPGA systems, the routing architecture employs a large number of buffers. As illustrated in Fig. 8-a, a buffer is employed after each routing multiplexer. On a general basis, the insertion of buffers breaks the routing path into smaller segments and reduces the quadratic delay of the path. In exchange, the intrinsic delay of the buffers is added to the path. Hence, over employing buffers can degrade the delay. Besides, reducing the number of buffers in routing structure can reduce the power consumption and area overhead. Replacing CMOS-based MUXs with RRAM-based MUXs reduces the path delay. Hence, the number of required buffers can intuitively be reduced in RRAM-based structures. Therefore, we propose to reduce the number of routing buffers in the global routing structures [15]. Fig. 8-b shows a modified routing scheme that employs buffers only every two routing multiplexers. This routing scheme leads to the best performance results in our context of RRAM-based FPGAs [15].

To summarize, RRAM-based routing multiplexers lead to three positive effects on the FPGA architecture: 1-it reduces the data path delay; 2-it tolerates large power supply reduction; and 3-it allows us to reduce the number of buffers within the global routing architecture.

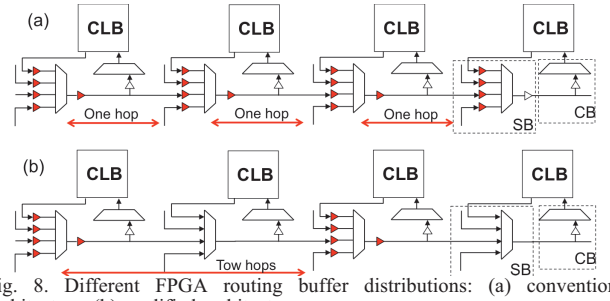

Fig. 8. Different FPGA routing buffer distributions: (a) conventional architecture; (b) modified architecture.

V. Near-V_T Non-Volatile Flip-Flop Design

In the presented architecture, the LUTs are driven by a collection of non-volatile flip-flops, providing large drive strength and endurance together with non-volatility. We present the flip-flop circuit design in this section and comment on its robustness at near-V_t power-supply.

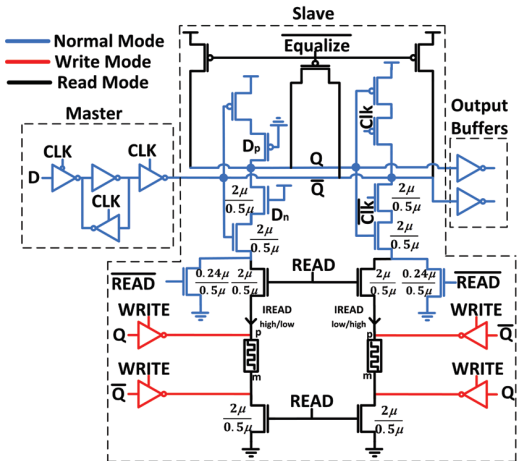

Fig. 9. Sub-V_t optimized non-volatile Flip-Flop design using complementary RRAMs.

A. RRAM-Based Non-Volatile Flip-Flop

We introduce non-volatility in a CMOS flip-flop structure without disturbing its standard behavior. For this purpose, we consider a conventional master-slave flip-flop realized in CMOS technology and we enhance it with non-volatile storage [14]. The non-volatile data storage is realized thanks to two RRAM devices inserted in the sink of the cross-coupled inverter pair in the slave latch. The resulting circuit is shown in Fig. 9. The two RRAM devices are always used in a complementary fashion, i.e., one device is programmed to the HRS, while the other is programmed to the LRS. Dedicated programming circuits for the RRAMs are inserted in the structure as well. They are highlighted in red color in Fig. 9.

Similarly, specific restore on wake-up circuits are shown in black color. In standard mode, the flip-flop operates as a regular flip-flop circuit, storing volatile information in the slave latch. Before turning *off* the system, the volatile information can be saved in the RRAM thanks to the inserted write circuits. When the circuit is turned *on* again, the non-volatile

information is transferred to the volatile latch using the wake-up circuits. Restoring the information consists in pre-charging and equalizing the nodes of the latch to V_{dd} and then starting the race condition that depends on the discharge of the internal nodes in the RRAM branches. For more details about the flip-flop architecture and principles, we refer the reader to [14] for more details.

B. Near-V_T / Sub-V_T Optimizations and Robustness

In the proposed flip-flop circuit, a correct read operation depends on the modulation of the discharge current by the complementary RRAM devices. A good control of the discharge current requires a good matching between the two CMOS branches. Therefore, we insert two always-*on* dummy transistors into the simpler non-clocked inverter to mimic the tri-state inverter in the other pull-down branch. In addition, all transistor pairs are upsized to further improve the matching. Applying circuit-level optimizations improves the robustness of the cell in both regular operations and in the near/sub-V_t regimes. As shown in [14], such structure is capable of tolerating variations in both CMOS and RRAM devices up to a 10% standard deviation from the nominal value of the parameters down to a very aggressive sub-V_t power supply of 0.4V.

VI. System-Level Impact

In this section, we attempt to forecast the gain brought by RRAMs in the field of ultra-low-power, yet high-performance, FPGAs.

A. Methodology

The architecture level simulations are done using the VTR flow [16]. The twenty largest MCNC benchmarks [17] are first synthesized by ABC [18]. Then, packing, placement, and routing are performed by VPR7 [16]. The island-type structure is considered and the technology parameters (area, delay and power) are extracted from a commercial mature 0.18µm technology. A mature technology is preferred here to ensure good reliability of operations in sub/near-V_t regimes. The benchmarks are mapped on both standard CMOS SRAM-based and RRAM-based FPGAs. We consider for the RRAM-based structure an optimized routing scheme as discussed earlier. Both architectures consider a standard single driver routing scheme with channel length of 1. Note that other channel length can be used with no specific differences with the results. We assume CBs with $F_{c,in}$=0.15 and $F_{c,out}$=0.10. The SBs use a *Wilton* pattern with F_s=3. We evaluate the performances of the architectures in both nominal (1.8V) and near-V_t (1.2V) conditions.

B. Architectural Benchmarking Results

Fig. 10-top illustrates the area comparison between the two FPGA architectures under the two considered voltage conditions. Compared to the standard FPGA architectures, the RRAM-based FPGA saves 12% area on average thanks to the monolithic integration of the memories, as well as the associated simplification of the routing scheme.

Fig. 10-middle illustrates the delay comparison between the four FPGA architectures. When V_{dd} drops from 1.8V to 1.2V, the standard FPGA architecture sees a 38% increase in its critical path delay, resulting from the degradation of driving current that transistors can provide. Compared to the standard FPGA architecture at V_{dd}=1.8V, the RRAM-based FPGA reduces by 10% on average the delay. This comes from the high performance of the RRAM-based routing architecture. When investigating a more energy efficient sub-V_t, we note that the RRAM-based routing architectures can still produce high performance at V_{dd}=1.2V and even compensate the delay degradation in logic elements, leading to overall performance gain of 26%. Such a result is of extreme interest as it shows that a near-V_t RRAM-based FPGA is able to over perform a regular CMOS architecture working at nominal voltage.

Fig. 10-bottom illustrates the power comparison between the four testcases. Both the standard and RRAM-based near-V_t FPGA architectures reduce on average by 81% the power consumption. This is accounted directly to the reduction of V_{dd}. At the same V_{dd}, RRAM-based and standard FPGAs have almost no difference in power consumption because of the similar switching capacitances in the data paths. In the logic elements, RRAM-based and standard FPGAs have similar switching capacitances because they share similar circuit topologies. In the RRAM routing architectures, the switch capacitances come from the programming transistors, while in the standard routing architecture, they come from the pass transistors. The number of programming transistors in a RRAM-based multiplexer roughly equals to the number of pass transistors in a standard one. Therefore, the switch capacitances in routing architectures are similar.

VII. Conclusion

In this paper, we described a non-volatile ultra-low-power FPGA architecture exploiting monolithically integrated RRAMs. We demonstrate RRAM-based circuit designs and FPGA architecture that do not target on certain special RRAM but are rather general. RRAM-based flip-flop design topologies are developed based on conventional master-slave flip-flop. Optimization topology is also proposed to ensure robustness and reliability against process variation and near/sub-V_t operations. We demonstrate RRAM-based multiplexer designs showing improved performance / power consumption operating points. Indeed, RRAM-based multiplexers stay high-performance with significant power reduction even in near/sub-V_t regime. Beside non-volatility, near-V_t FPGA employing proposed RRAM-based circuits designs demonstrate overwhelming improvements compared to SRAM-based conventional FPGA working at nominal voltage with on average 12%, 26% and 81% improvements in area, delay and power consumption respectively.

References

[1] I. Kuon and J. Rose, "Quantifying and Exploring the Gap Between FPGAs and ASICs," Springer, 2009.

[2] M. Lin, A. El Gamal, Y.-C. Lu and S. Wong "Performance Benefits of Monolithically Stacked 3-D FPGA," IEEE TCAD, vol. 26, no. 2, pp. 216-229, 2007.

[3] L. Cheng, P. Wong, F. Li, Y. Lin and L. He, "Device and Architecture Cooptimization for FPGA Power Reduction," DAC Tech. Dig., 2005.

[4] T. Tuan, A. Rahman, S. Das, S. Trimberger and S. Kao, "A 90-nm Low-Power FPGA for Battery-Powered Applications," IEEE TCAD, vol. 25, no. 2, pp. 296-300, 2007.

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

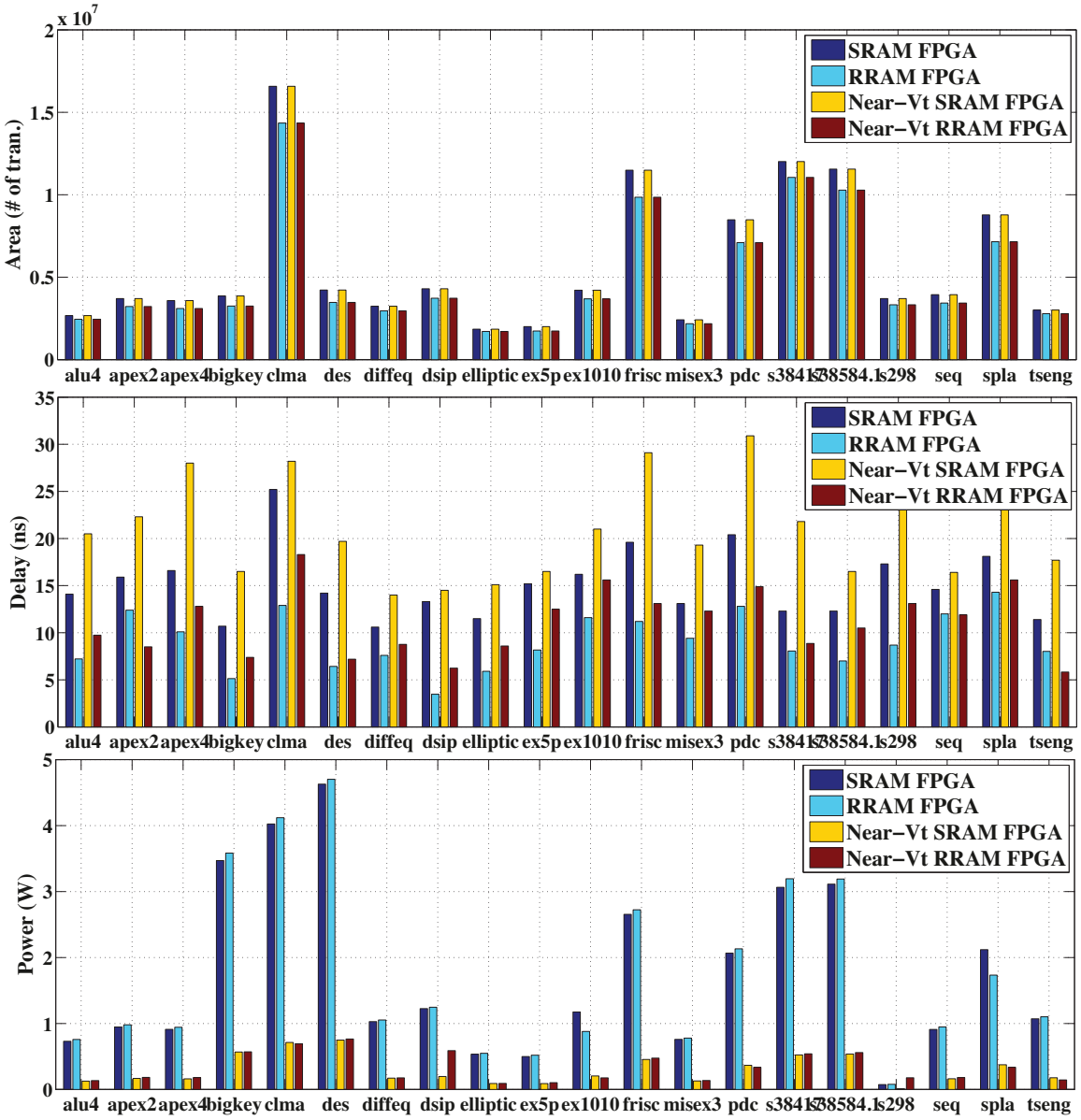

Fig. 10. Area (top), critical path delay (middle) and power (bottom) comparison for 20 biggest MCNC benchmarks implemented in standard CMOS architecture at V_{DD}=1.8V, RRAM-based architecture at V_{DD}=1.8V, standard CMOS architecture at V_{DD}=1.2V and RRAM-based architecture at V_{DD}=1.2V.

[6] O. Turkyilmaz *et al.*, "RRAM-based FPGA for Normally Off, Instantly On Applications," Nanoarch Tech. Dig. 2012.

[7] H.-S. P. Wong *et al.*, "Metal-Oxide RRAM," Proc. of the IEEE, vol. 100, no. 6, pp. 1951-1970, 2012.

[8] Y. Chen *et al.*, "Non-Volatile 3D Stacking RRAM-based FPGA," FPL Tech. Dig., 2012.

[9] S. Tanachutiwat *et al.*, "FPGA Based on Integration of CMOS and RRAM," IEEE TVLSI, vol. 19, no. 11, pp. 2023-2032, 2010.

[10] P.-E. Gaillardon *et al.*, "Emerging Memory Technologies for Reconfigurable Routing in FPGA Architecture," ICECS Tech. Dig., 2010.

[11] P.-E. Gaillardon *et al.*, "Design and Architectural Assessment of 3-D Resistive Memory Technologies in FPGAs," IEEE TNANO, vol. 12, no. 1, pp. 40-50, 2013.

[12] J. Cong, B. Xiao, "FPGA-RPI: A Novel FPGA Architecture With RRAM-Based Programmable Interconnects," IEEE TVLSI, vol. 22, no. 4, pp. 864-877, 2014.

[13] P.-E. Gaillardon *et al.*, "GMS: Generic Memristive Structure for Non-Volatile FPGAs," IEEE/IFIP VLSI-SoC Tech. Dig., 2012.

[14] I. Kazi *et al.*, "Energy/Reliability Trade-Offs in Low-Voltage RRAM-based Non-Volatile Flip-Flop Design, vol. 61, no. 11, pp. 3155-3164, 2014.

[15] S. Rahimian, X. Tang, P.-E. Gaillardon and G. De Micheli, "A Study on Buffer Distribution for RRAM-based FPGA Structure," *submitted*, LASCAS Tech. Dig., 2015.

[16] J. Rose *et al.*, "The VTR Project: Architecture and CAD for FPGAs from Verilog to Routing," *FPGA Tech. Dig.*, 2012.

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

[18] University of California in Berkeley, ABC: A System for Sequential Synthesis and Verification, Available online. http://www.eecs.berkeley.edu/~alanmi/abc/