# A Study on Buffer Distribution for RRAM-based FPGA Routing Structures

Somayyeh Rahimian Omam, Xifan Tang, Pierre-Emmanuel Gaillardon, Giovanni De Micheli
Integrated System laboratory, EPFL
Lausanne, Switzerland

**Abstract – Compared to *Application-Specific Integrated Circuits* (ASICs), *Field Programmable Gate Arrays* (FPGAs) provide reconfigurablity at the cost of lower performance and higher power consumption. Exploiting a large number of programmable switches, routing structures are mainly responsible for the performance limitation. Hence, employing more efficient switches can drastically improve the performance and reduce the power consumption of the FPGA. *Resistive Random Access Memory* (RRAM)-based switches are one of the most promising candidates to improve the FPGA routing architecture thanks to their low *on*-resistance and non-volatility. The lower *RC* delay of RRAM-based routing multiplexers, as compared to CMOS-based routing structures encourages us to reconsider the buffer distribution in FPGAs. This paper proposes an approach to reduce the number of buffers in the routing path of RRAM-based FPGAs. Our architectural simulations show that the use of RRAM switches improves the critical path delay by 56% as compared to CMOS switches in standard FPGA circuits at 45-nm technology node while, at the same time, the area and power are reduced, respectively, by 17% and 9%. By adapting the buffering scheme, an extra bonus of 9% for delay reduction, 5% for power reduction and 16% for area reduction can be obtained, as compared to the conventional buffering approach for RRAM-based FPGAs.**

## I. Introduction

In the recent years, the market share of *Field Programmable Gate Arrays* (FPGAs) is increasing due to their large versatility. Nevertheless, FPGAs are still worse than *Application-Specific Integrated Circuits* (ASICs) in terms of computational density (area), delay, and power consumption. In conventional FPGA architectures, *Static Random Access Memories* (SRAMs) and transmission gates are used to form the programmable interconnects. The relatively low density of SRAM-based storage leads to significant silicon area utilization and consequently to longer routing paths and larger interconnect delays. Moreover, a large amount of power is consumed in SRAMs during standby due to the volatile nature of the memory circuits. The programmable interconnects occupy 50-90% of FPGA area and are responsible for 70-80% and 60-80% of the total delay and power consumption, respectively [1,2].

Replacing SRAM cells with *Non-Volatile Memories* (NVM) has been introduced as a promising approach to reduce the standby power, area and consequently delay of FPGAs [3-8]. *Resistive Random Access Memory* (RRAM) cells provide ultra-dense back-end-of-line integration with easy programing features, fast write time and low write energy as compared to other emerging NVMs [9-11]. These characteristics make RRAMs a promising emerging solution for next-generation FPGAs. A FPGA using RRAM-based configuration memory has been demonstrated in [4] and reduces the area by 40% and the *energy delay product* (EDP) by 28% as compared to conventional FPGAs. In addition to simply replacing the configuration memories, there have been several studies on RRAM-based routing structures [5-8]. RRAM switches are employed to form non-volatile multiplexers that are inserted in the routing paths. The small size and low *on*-resistance of RRAM cells guarantee a higher level of performances for RRAM-based routing structures.

Signal buffers are unavoidable parts of the routing paths in FPGAs and most of the previously proposed RRAM-based FPGA architectures [3-7] use straightforwardly the standard buffering scheme. However, the use of different switches is likely to impact the buffer allocation in RRAM-based FPGAs. To cope with this question, an adaptive buffer allocation method is proposed in [8], where the buffers are inserted on demand within the data path. While showing great promises, this method implies a serious change in the architecture and design methodology and employs additional RRAM switches to allocate the buffers. However, an intermediate path can be followed, consisting of optimizing the traditional FPGA buffering scheme to RRAM-based routing circuitries.

In this paper, we first analyze the effects of buffer allocation in RRAM-based FPGAs. Then, we adapt the traditional buffering scheme to RRAM-based routing structures, by reducing the number of buffers without sacrificing the system performance. Unlike traditional architectures that employ a buffer at the output of every routing switchbox, we omit some of the buffer and use unbuffered switchboxes. Architectural simulation results show that an RRAM-based FPGA can outperform its standard CMOS counterpart by 17%, 56% and 9% in area, delay and power respectively at a 45-nm technology node. When using the newly proposed buffering scheme, an extra bonus of 16%, 9% an 5% in area, delay and power respectively is granted to the RRAM-based architecture.

The rest of the paper is organized as follows. Section II presents a review of conventional and RRAM-based FPGA structures. Section III studies the effect of buffer allocation on FPGA performance. Section IV shows some architectural simulation results. Section V draws some conclusions.

## II. Background

The structure of a conventional island-style FPGA [2] consists of *Configurable Logic Blocks* (CLBs) implementing both combinational and sequential logic functions, *Connection Boxes* (CBs) connecting the CLBs to the routing channels, and *Switch Boxes* (SBs) providing the connections between the different routing channels. CBs and SBs consist in a large set of programmable multiplexers that are configured using programming bits. Conventionally, scan-chain SRAM cells are used to store the configuration data [2], which results in large area overhead for conventional FPGAs.

RRAM switches can be easily integrated into conventional FPGA circuits since their fabrication technology is fully compatible with standard CMOS process [9-11]. Therefore, new routing mutliplexer structures exploiting RRAMs to route the signals, i.e., combining the routing transmission-gate and the configuration memory, have been proposed in [6]. We report in Fig. 1 the structure of 4:1 RRAM-based routing multiplexer (Fig. 1-b) that can be employed to replace a traditional CMOS-based routing multiplexer (Fig. 1-a). Such a structure introduces RRAM in the signal datapath. The *on*-resistance of RRAM switches is generally lower than CMOS switches (<2kΩ for RRAM switches [6,7] and ~4kΩ for 45nm CMOS switch) which results in lower *RC* delay for RRAM-based

978-1-4799-8332-2/15/$31.00 ©2015 IEEE

routing paths.

Along with the configurable switches, buffers are the other important element of the FPGAs routing structure to optimize the *RC* delay of the routing paths. In conventional FPGA structures, after each SB, a buffer restores the signal and drives the next block. This *RC* delay reduction of the configurable switches in RRAM-based FPGAs affects the buffering needs. Thus, the buffer allocation should be reconsidered. In the next section, we study the effect of buffers on signal propagation delay and identify an optimum buffer repartition for RRAM-based FPGAs.

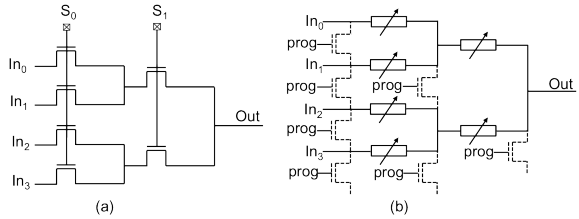


Fig. 1. SRAM and RRAM- based implementation of 4:1 multiplexer.

## III. Buffer Distribution in RRAM-based FPGA

In this section, we study the impact of the buffer distribution in the routing path of RRAM FPGAs, by identify the relation between the number of buffers and the signal delay. Then, we use circuit-level simulations to identify the optimum architecture.

### *A. Delay Calculation in Critical Path*

On a general basis, the insertion of buffers breaks the routing path into smaller segments and reduces the quadratic delay of the path [12] . 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.

Instead of using a buffer after each switchbox, we consider the use of a buffer after each $n$ switchboxes where $n>1$. The optimum number $n$ depends on the circuit characteristics. Fig. 2 shows the *RC* model for the critical path between two logic blocks where there are $N$ switchboxes and a buffer is assigned after each $n$ switch boxes. In this condition, the number of intermediate buffers is $N/n$-1.

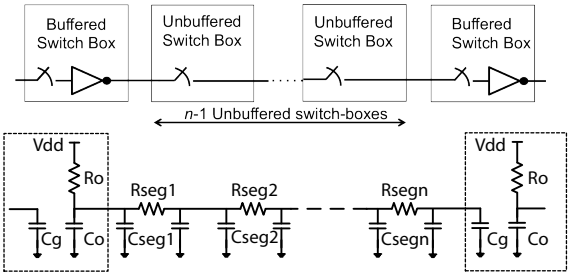


Fig. 2. Critical path between two buffered SB and its associated *RC* model.

Employing the Elmore approximation [12], the delay between two buffers can be expressed as:

$$\tau_s = R_o\left(C_o + C_g\right) + 2nR_oC_{seg} + nR_{seg}C_g + n^2R_{seg}C_{seg}, \quad (1)$$

where $R_o$ and $C_o$ stand for the output resistance and capacitance of the input buffer, and $R_{seg}$ and $C_{seg}$ denote the total resistance and capacitance of each segment, i.e., including the RRAM switches and the wiring parasitics between the buffered switchboxes. The delay between two logic blocks is:

$$\tau = \frac{N}{n}\tau_s + \left(\frac{N}{n} - 1\right)\tau_b, \quad (2)$$

where $\tau_b$ is the intrinsic delay of the buffers. By combining (1) and (2) and by identifying the minimum of the function, we derive that the propagation delay τ is minimized for an optimum $n$ such as:

$$n = \sqrt{\frac{R_o\left(C_o + C_g\right) + \tau_b}{R_{seg}C_{seg}}}. \quad (3)$$

Note that $n$ increases when $R_{seg}$ reduces. The resistance of a routing segment decreases as RRAM switches have smaller *on*-resistance as compared to SRAM-based structures. Therefore, $R_{seg}$ and $C_{seg}$ can be smaller in RRAM-based structures, which results in a higher $n$ and a lower number of buffers in the context of RRAM-based FPGAs. Interestingly, the number of unbuffered switches between two buffered switches $n$ is not dependent on the length of critical path which allows us to determine a unique $n$ for different critical paths.

### *B. Circuit Simulations Methodology*

We perform circuit-level simulations using Hspice to evaluate the effect of buffers on the circuit performance metrics and to identify the optimum number of buffers for the RRAM-based FPGA critical path. The performances of RRAM-based routing structures are compared to their SRAM-based counterparts. Electrical simulations are performed in a commercial 45-nm technology with $V_{dd}$ equal to 1V. However, the results are not technology dependent and similar improvement can be achieved using other technologies. The interconnect length and characteristics is estimated based on commercial 45-nm FPGA data.

We model the critical path of the FPGA between two logic blocks as per the following. In the reference architecture, the critical path between the logic blocks goes through $N$ buffered switchboxes. In the novel buffering scheme, we assume the same critical path but we distributed equally $n$ buffered switchboxes among $n$-1 unbuffered switchboxes, as illustrated in Fig. 3 for $n=3$.

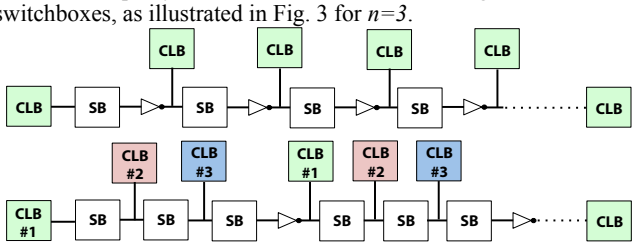


Fig. 3. Conventional and modified ($n=3$) buffering distributions.

Note that when $n \neq N$, i.e., when unbuffered SBs are used, an entering signal to a routing track can be either immediately followed by a buffer, or have several unbuffered SB to pass though before reaching a buffer. This has an impact on the delay performances. For example, in Fig. 3, a path starting from logic blocks #1 has different characteristics as compared to paths starting from logic blocks #2 and #3 even if they propagate through the same number of levels. However, in all our case studies, simulation results show negligible difference in delay and power consumed between these different structures. Hence, we only report in the following results where the signal enters at a buffered stage.

The SB structure are shown in Fig. 4. The traditional SB is depicted in Fig. 4-a. Each signal propagates though one switch followed by a line buffer. The switch can be either a transmission-gate (Fig. 4-a) for the CMOS reference architecture or an RRAM for the proposed architecture (Fig. 4-b). In an unbuffered switchbox, as shown in Fig. 4-c, the same structure is used but without any input/output buffers.

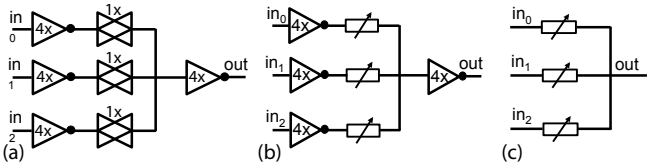


Fig. 4. One-track switchbox implementation – (a) CMOS-based buffered SB – (b) RRAM-based buffered SB – (c) RRAM-based unbuffered SB.

The RRAM cells are modeled by parasitic elements according to [13]. In our simulations, we consider two different RRAM technological options with $R_{on}$ of 1kΩ and 2 kΩ [7] respectively, to evaluate the impact of the *on*-resistance on the buffering scheme. The $R_{off}$ for both case is 1MΩ. For RRAM-based structure, the impact of the programming transistors are included in the simulation model.

### C. *Experimental Results*

In our first circuit simulation, we consider a path consisting of 10 switchboxes ($N$=10) and sweep $n$ to find the optimum number of buffers. Fig. 5 shows the delay evolution as a fonction of $n$ for SRAM and RRAM ($R_{on}$=1 kΩ and 2 kΩ) critical paths. As expected, the signal delay increases when the *on*-resistance of the RRAMs increases. The simulation results show that for SRAM-based FPGAs, the minimum delay is obtained by having one buffer after each switchbox, whereas for RRAM-based FPGAs, there is no need to dedicate a buffer after each switchbox. In this case study, using one buffer after every two switchboxes results in the lowest delay.

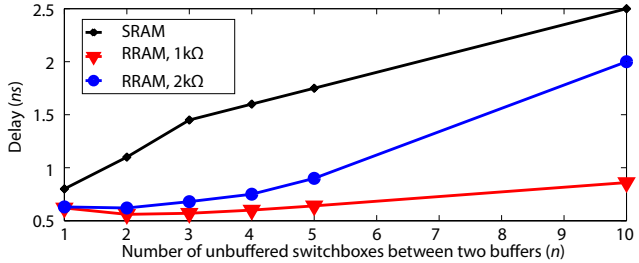


Fig. 5. Critical path delay for SRAM- and RRAM-based FPGAs for $N$=10 and $R_{on}$=1 kΩ and 2 kΩ.

In a second set of simulations, we vary the length of the critical path to evaluate its effect on the optimum number of buffers. We consider $N$=10 and $N$=20 with $R_{on}$=1kΩ. The experiment results are depicted in Fig. 6.

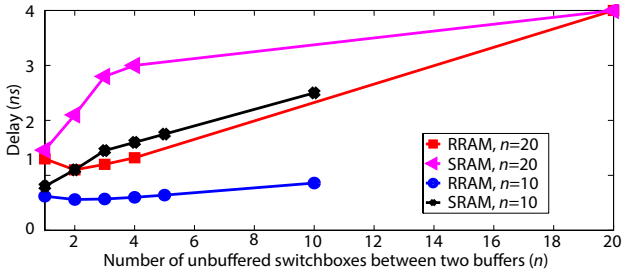


Fig. 6. Critical path delay for SRAM and RRAM based FPGAs for $N$=10 and $N$=20.

In both cases, the optimum number $n$ that minimizes the delay is $n$=2. It confirms the aforementioned expression (3), where $n$ is not dependent to $N$.

Circuit-level simulations confirm the interest of RRAM switches to improve the performances of the critical path by around 40%. The low *on*-resistance of RRAM switches reduces the *RC* delay of routing path and allows us to reduce the number of signal buffers. Reducing the number of buffers further enhance the performances of RRAM-based routing ressources.

Based on this fact, modified structures can be proposed for RRAM-based FPGAs where some of the conventional buffered multiplexers are replaced by unbuffered RRAM switches.

## IV. Architectural Simulations

In the previous section, we studied the effect of buffer distribution at the circuit level. In this section, we move to the architectural level and study the impact of the buffer allocation on the FPGA performance.

### A. *Methodology*

The architecture level simulations are done using the VTR flow [15]. The twenty largest MCNC benchmarks [16] are first synthesized by ABC [17]. Then, packing, placement, and routing are performed by VPR7 [15]. The island-type structure is considered and the technology parameters (area, delay and power) are extracted from a commercial 45nm technology.

The benchmarks are mapped on both standard CMOS SRAM-based and RRAM-based FPGAs. The different routing schemes that will be considered are depicted in Fig. 7. In Fig. 7-a, we consider a standard single driver routing scheme with channel length of 1. All the routing multiplexers are buffered. Note that other channel length can be used with no specific differences with the results. Fig. 7-b shows a modified routing scheme that removes half of the buffers. After each buffered multiplexer, a buffer is removed. We call this routing scheme $B_2$. Fig. 7-c removes two third of the buffers by using two unbuffered multiplexers between the buffered multiplexers. This scheme is called $B_3$.

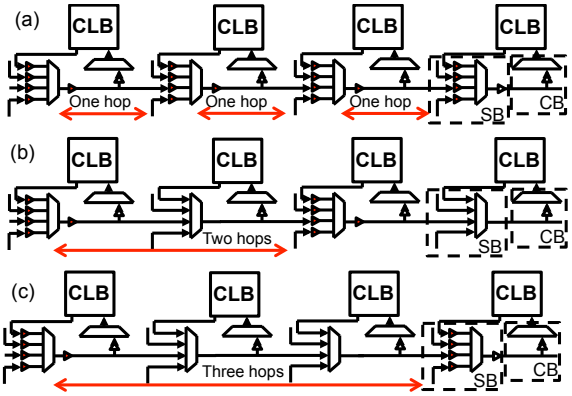


Fig. 7. Different FPGA routing buffer distributions: (a) conventional architecture; (b), (c) modified architecture $B_2$ and $B_3$. Hops between the buffered switchboxes are highlighted by the red arrows.

### B. *Simulation Results*

Fig. 8 shows area, critical path delay and power consumption for five different FPGA architectures exploiting: an SRAM-based conventional routing, an RRAM-based conventional routing, an SRAM-based $B_2$ routing, and RRAM-based $B_2$ and $B_3$ routings.

Comparing conventional structure $B_1$ for CMOS and RRAM-based circuits, we observe that the use of RRAM switches improves the critical path delay by 56% where the area and power consumption are reduced by 16.8% and 8.9% respectively.

In SRAM-based structures, $B_1$ demonstrates the best timing performance which means that reducing the number of buffers for these FPGAs is not a useful approach. This is in total coherence with the results obtained at the electrical-simulation level. Alternatively, in RRAM-based structures, $B_2$ shows better performance which validates the use of one unbuffered switchbox after every conventional buffered block. Averaged over the studied benchmarks, we conclude that a $B_1$ structure employed within a RRAM-based FPGA leads to an extra improvement of 15.9%, 8.6% and 5% in area, delay and power consumption respectively.

## V. Conclusion

RRAMs offer new opportunities to form efficient FPGA routing structures. Thanks to their reduced *RC* delay as compared to conventional transistors, the use of RRAM-based routing structures allow us to reduce the number of intermediate buffers and improve power, area and even the delay in RRAM-based FPGAs. In this paper, the effect of buffers on routing paths of RRAM-based FPGAs is investigated. Analytical expressions are presented to determine the optimum number of buffers for each defined path and circuit level simulations are performed to validate these analytical

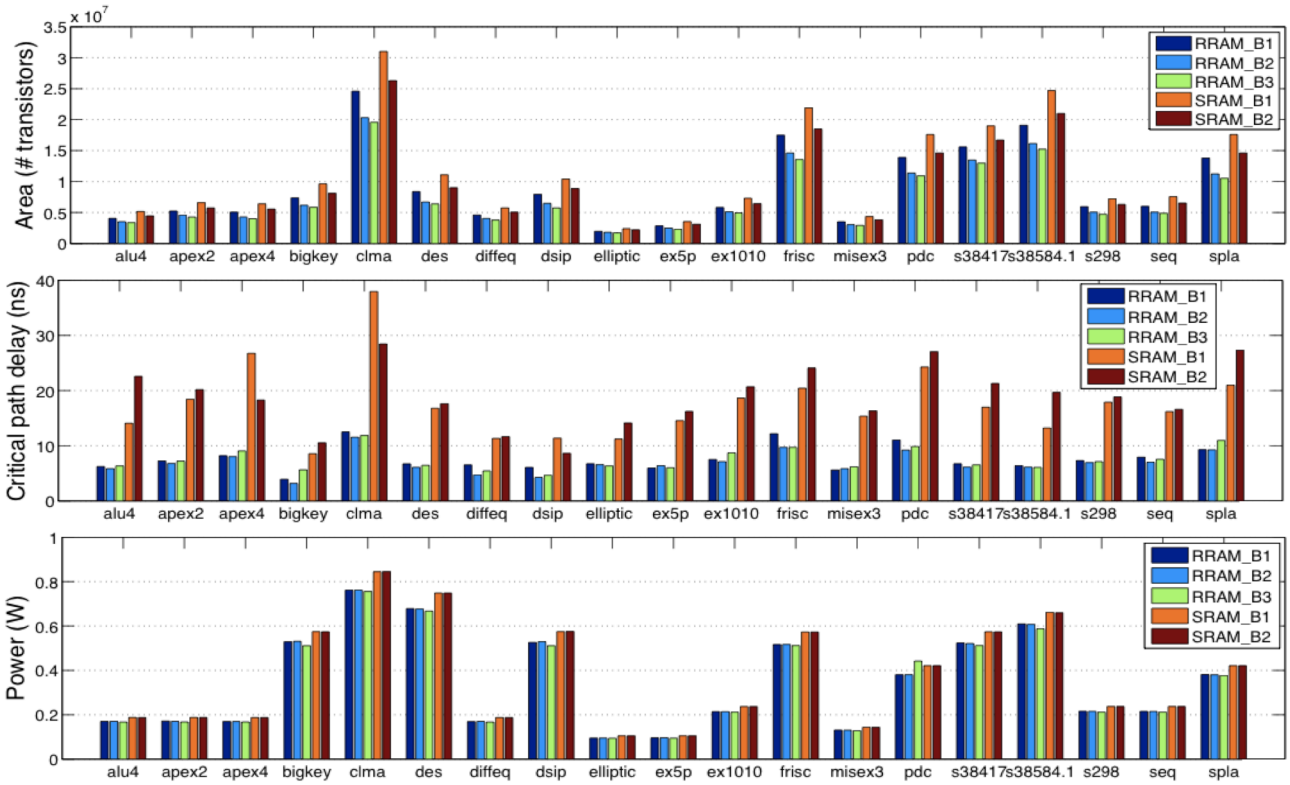


Fig. 8. Architectural simulation results (top-Area, middle-Critical Path Delay and bottom-Power) for the MCNC benchmarks- The different buffer distribution schemes $B_1$ to $B_3$ are considered for both SRAM- and RRAM-based architectures.

expressions.

Architectural simulations for the twenty largest MCNC benchmarks show that exploiting RRAM-routing multiplexers can reduce the delay by 56% as compared to SRAM-based architecture, while the power consumption and area are also reduced by 8.9% and 16.8%. Reducing the numbers of buffers in the routing ressources by half leads to additional improvements of 8.6%, 5%, and 15.6%, respectively, for delay, area, and power for RRAM-based FPGAs. In SRAM-based structures reducing the number of buffers degrades the performance of the circuit due to high resistance of the CMOS switches.

## Acknowledgments

This work has been partly supported by the ERC senior grant NanoSys ERC-2009-AdG-246810 and the Swiss National Science Foundation under the project No. 200021- 146600.

## References

[1] M. Lin, A. El Gamal, “A Low-Power Field-Programmable Gate Array Routing Fabric,” *IEEE TVLSI*, 17(10):1481-1494, 2009.

[2] V. Betz, J. Rose, A. Marquardt, “Architecture and CAD for Deep-Submicron FPGAs,” Kluwer Publishing Group, 1999.

[3] M. Liu, W. Wang, “rFGA: CMOS-nano hybrid FPGA using RRAM components,” *Nanoarch Tech. Dig.*, 2008.

[4] Y. L. Young, Z. Zhiping, K. Wanki, A. El Gamal, S. S. Wong, "Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory," *ISSCC Tech. Dig.*, 2012.

[5] P.-E. Gaillardon, M. H. Ben Jamaa, G. Betti Beneventi, F. Clermidy L. Perniola, “Emerging Memory Technologies for Reconfigurable Routing in FPGA Architecture,” *ICECS Tech. Dig.*, 2010.

[6] P.-E. Gaillardon, D. Sacchetto, S. Bobba, Y. Leblebici, G. De Micheli, “GMS: Generic memristive structure for non-volatile FPGAs,” *VLSI-SoC Tech. Dig.*, 2012.

[7] P.-E. Gaillardon *et al.*, “Design and Architectural Assessment of 3-D Resistive Memory Technologies in FPGAs,” *IEEE TNANO*, 12(1): 40-50, 2013.

[8] J. Cong, X. Bingjun, “FPGA-RPI: A Novel FPGA Architecture With RRAM-Based Programmable Interconnects,” *IEEE TVLSI*, 22(4):864-877, 2014.

[9] S.-S. Sheu *et al.*, “A 4Mb Embedded SLC Resistive-RAM Macro with 7.2ns Read-Write Random-Access Time and 160ns MLC-Access Capability,” *ISSCC Tech. Dig.*, 2011.

[10] C. Meng-Fan *et al.*, “A High-Speed 7.2-ns Read-Write Random Access 4-Mb Embedded Resistive RAM (ReRAM) Macro Using Process-Variation-Tolerant Current-Mode Read Schemes,” *IEEE JSSCC*, 48(3):878-891, 2013.

[11] R. Fackenthal *et al.*, “A 16Gb ReRAM with 200MB/s Write and 1GB/s Read in 27nm Technology,” *ISSCC Tech. Dig.*, 2014.

[12] W. C. Elmore, “The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers,” *J. of Appl. Phys.*, 19(1):55-63, 1948.

[13] P. Huang *et al.*, “A Physics-Based Compact Model of Metal-Oxide-Based RRAM DC and AC Operations,” *IEEE TED*, 60(12):4090-4097, 2013.

[14] S. Onkaraiah *et al.*, “Using OxRRAM Memories for Improving Communications of Reconfigurable FPGA Architectures,” *Nanoarch Tech. Dig.*, 2011.

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

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

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