

PROGRESS REVIEW • OPEN ACCESS

Via-switch FPGA with transistor-free programmability enabling energy-efficient near-memory parallel computation

To cite this article: Masanori Hashimoto *et al* 2022 *Jpn. J. Appl. Phys.* **61** SM0804

View the [article online](#) for updates and enhancements.

You may also like

- [Implementation and verification of different ECC mitigation designs for BRAMs in flash-based FPGAs](#)
Zhen-Lei Yang, , Xiao-Hui Wang et al.
- [A new FPGA architecture suitable for DSP applications](#)
Liyun Wang, , Jinmei Lai et al.
- [A software solution to estimate the SEU-induced soft error rate for systems implemented on SRAM-based FPGAs](#)
Zhongming Wang, , Zhibin Yao et al.



Via-switch FPGA with transistor-free programmability enabling energy-efficient near-memory parallel computation

Masanori Hashimoto^{1*}, Xu Bai², Naoki Banno², Munehiro Tada², Toshitsugu Sakamoto², Jaehoon Yu³, Ryutaro Doi⁴, Hidetoshi Onodera⁵, Takashi Imagawa⁶, Hiroyuki Ochi⁷, Kazutoshi Wakabayashi⁸, Yukio Mitsuyama⁹, and Tadahiko Sugibayashi^{2*}

¹Dept. Communications and Computer Engineering, Kyoto University, Kyoto, Kyoto 606-8501, Japan

²NanoBridge Semiconductor, Inc., Tsukuba, Ibaraki 305-0047, Japan

³AI Computing Research Unit, Tokyo Institute of Technology, Yokohama, Kanagawa 226-8502, Japan

⁴Dept. Information Systems Engineering, Osaka University, Suita, Osaka 565-0871, Japan

⁵Faculty of Informatics, Osaka Gakuin University, Suita, Osaka 564-8511, Japan

⁶Department of Computer Science, Meiji University, Kawasaki, Kanagawa 214-8571, Japan

⁷College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan

⁸d.lab, The University of Tokyo, Bunkyo, Tokyo 113-8656, Japan

⁹School of Systems Engineering, Kochi University of Technology, Kami, Kochi 782-8502, Japan

*E-mail: hashimoto@i.kyoto-u.ac.jp; sugibayashi@nanobridgesemi.com

Received January 11, 2022; revised April 5, 2022; accepted April 27, 2022; published online June 23, 2022

We are developing field-programmable gate arrays (FPGAs) with a new non-volatile switch called via-switch. In via-switch FPGAs (VS-FPGAs), the via-switches required for reconfiguration are placed in the routing layer so that the entire transistor layer can be utilized for computing, and higher implementation density can be achieved compared to conventional SRAM FPGAs. Furthermore, since arithmetic units and memories for computing can be placed under the via-switch crossbar for routing, large-scale parallel operations can be realized where the memory and the arithmetic unit are adjacent to each other. These features enable operation with high energy efficiency. This article reports 65 nm prototype fabrication results and predicted the performance of the VS-FPGA designed for AI applications. We also present the developed application mapping flow and crossbar programming method. The VS-FPGA closes the gap between FPGA and application-specific integrated circuits (ASIC) with the performance advantage of the via-switch and via-switch copy scheme for FPGA-to-ASIC migration, contributing to the expansion of the FPGA usage. © 2022 The Author(s). Published on behalf of The Japan Society of Applied Physics by IOP Publishing Ltd

1. Introduction

Expectations for reconfigurable circuits, such as FPGAs, are increasing due to the rising cost of ASIC development. Figure 1 compares FPGA and ASIC in terms of five aspects; time to market, energy efficiency, performance, unit cost, and non-recurring engineering (NRE) cost. FPGA is superior in terms of short time to market and low NRE cost. FPGAs are a suitable platform for implementing up-to-date machine learning algorithms and state-of-the-art AI applications, including inference engines in embedded systems and training accelerators in cloud systems. However, ASIC is better in energy efficiency, performance and unit cost. Due to the large number of switch circuits on FPGAs that switch wiring connections to achieve reconfiguration, signal delay, power consumption, and circuit area are large.^{1,2)} An online database³⁾ compares the energy efficiency of neural network (NN) accelerators reported in the literature. It shows that the gap in energy efficiency between FPGA and ASIC is more than 10×. This large gap limits the FPGA usage.

We are developing an FPGA that replaces SRAM-based switches, which are the performance bottleneck of conventional FPGAs, with via-switches, a type of non-volatile nano-scale switch. Figure 2 highlights the feature of via-switch FPGA (VS-FPGA). In the conventional SRAM FPGA, SRAM cell and switch are used for providing programmability, and they consume most of the chip area. On the other hand, in the VS-FPGA, programmability is attained by the via-switch, which locates in back-end-of-line (BEOL) interconnect layers. This

switch is non-volatile, and then one via-switch has both functions of memory and switch. Therefore, the VS-FPGA does not spend silicon area for programmability. The entire silicon area can be used for computing as logic and memory. This article summarizes the current status of VS-FPGA development and reports the expected performance improvement.

The rest of this article is organized as follows. Section 2 explains the structure and features of via-switch, and Sect. 3 demonstrates the advantage of the VS-FPGA with a prototype chip fabricated in a 65 nm process. Section 4 discusses the FPGA architecture suitable for via-switch and AI applications. Application mapping flow and via-switch crossbar programming are described in Sects. 5 and 6, respectively. Section 7 introduces via-switch copy scheme that reduces the cost of mass fabrication, and finally Sect. 8 concludes the discussion.

2. Via-switch

A via-switch is a stacked device consisting of non-volatile atom switches with a high on-off ratio and varistors for selection and can be integrated into a wiring layer with a small footprint.⁴⁾ Figure 3 shows the via-switch whose varistor is composed of a-Si/SiN_x/a-Si.^{5,6)} The atom switch has a structure in which a solid electrolyte is sandwiched between copper and ruthenium electrodes, and the formation and disappearance of cross-links by copper ions can be reversibly repeated by applying a voltage between the electrodes.⁷⁾ See (Ref. 6) for the details of the via-switch device structure and characteristics.



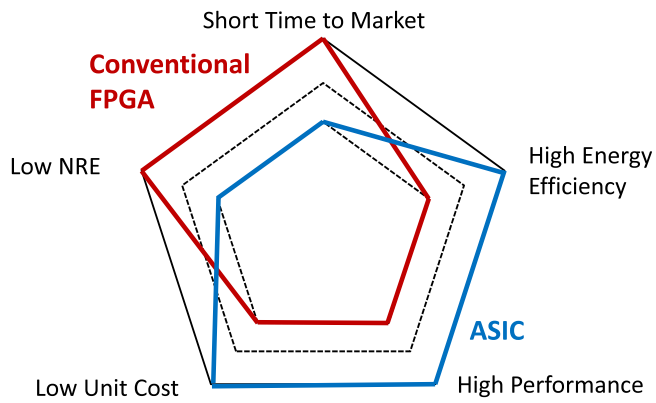


Fig. 1. (Color online) Conventional FPGA versus ASIC.

The via-switch is a device in which a varistor is connected to the control terminal of a complementary atom switch (CAS) consisting of two-atom switches (right figure of Fig. 4). Here, two atom switches are connected in series for achieving high reliability,⁸⁾ and two varistors are introduced for achieving multiple fanouts in the crossbar.^{4,9)} The right figure depicts the programming of the lower atom switch. To provide current, we give a high voltage to the vertical signal line in red and a ground voltage to the horizontal control line in blue. We program two-atom switches sequentially. The programming sequence will be discussed in Sect. 6.

3. Prototype chip demonstration

To demonstrate the feasibility of VS-FPGA and potential performance improvement, a small-scale VS-FPGA is fabricated.^{10–12)} Figure 5 shows the die photo of the fabricated chip and the interconnect structure. The via-switches are located between M4 and M5 layers. The FEOL and M1 to M4 metal layers are processed in a commercial 65 nm CMOS process, and the via-switch, and M5 to M7 metals are processed by ourselves. The footprint of the via-switch is $8 \times 6 F^2$, where F is the minimum unit defined by technology. If we use four metal layers for via-switch, it can be reduced 18 F^2 .⁹⁾ The fabricated FPGA consists of 6×6 logic cells, and each logic cell is configured with a basic logic element consisting of two sets of 4-input lookup tables (LUTs) and a D flip-flop. In the logic cells, via-switches are used for the

crossbar for signal routing and the memory of the LUTs, as illustrated in the left figure of Fig. 4.

Figure 6 shows the area compared with the previously designed SRAM FPGA,¹³⁾ demonstrating that the same functionality can be achieved with only 8.3% of the area compared to the SRAM FPGA. References 11,12 also show the performance comparison between VS-FPGA and atom-switch FPGA (AS-FPGA), where the AS-FPGA uses two-atom switches and one access transistor for each cross point in the crossbar.¹³⁾ The measurement results show that the VS-FPGA has 51% to 58% less energy-delay product than AS-FPGA. Note that the AS-FPGA has 60% less active power and 3x faster operation compared with the SRAM FPGA.¹⁴⁾

It should be noted that Refs. 15–18 introduced RRAM as not only configuration memory but also programmable switches for signal transmission and estimated the performance improvement. However, their silicon chip implementations have not been presented.

4. FPGA architecture exploration

This Sect. discusses the architecture dedicated to VS-FPGA and its extension to AI applications.

4.1. Baseline structure

VS-FPGAs have a structure of CLBs (Configurable Logic Blocks) as shown in the left figure of Fig. 4, and each CLB consists of a crossbar circuit and a logic block with via-switches at the intersections of vertical and horizontal wires. Each CLB consists of a crossbar circuit and logic blocks with via-switches placed at the intersections of vertical and horizontal wires.⁹⁾ This baseline structure originates from the AS-FPGA.¹³⁾ On the other hand, in the VS-FPGA, the transistor area under the crossbar can be fully used for computing, whereas many access transistors exist under the crossbar in the AS-FPGA. Therefore, the VS-FPGA can accommodate SRAM and arithmetic units in addition to LUTs for implementing memories, combinational circuits, and sequential circuits.

We also explored a highly dense reconfigurable architecture that uses via-switch for crossbar implementation. We devised a bidirectional interconnect structure that can exploit the small footprint and low resistivity of via-switch.⁹⁾ As the interconnect length becomes shorter compared with SRAM FPGA, the number of required repeaters is small, and hence the bidirectional interconnect becomes possible. Reference 9

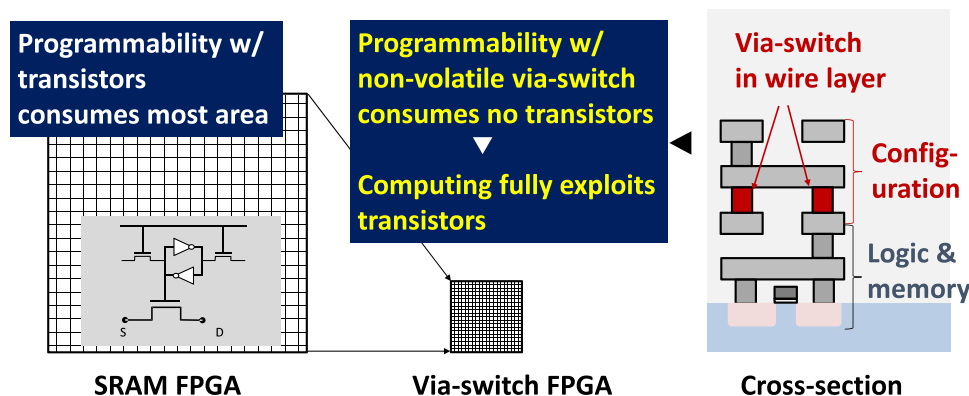


Fig. 2. (Color online) Overview of VS-FPGA. © [2020] IEEE. Reprinted, with permission, from (Ref. 10).

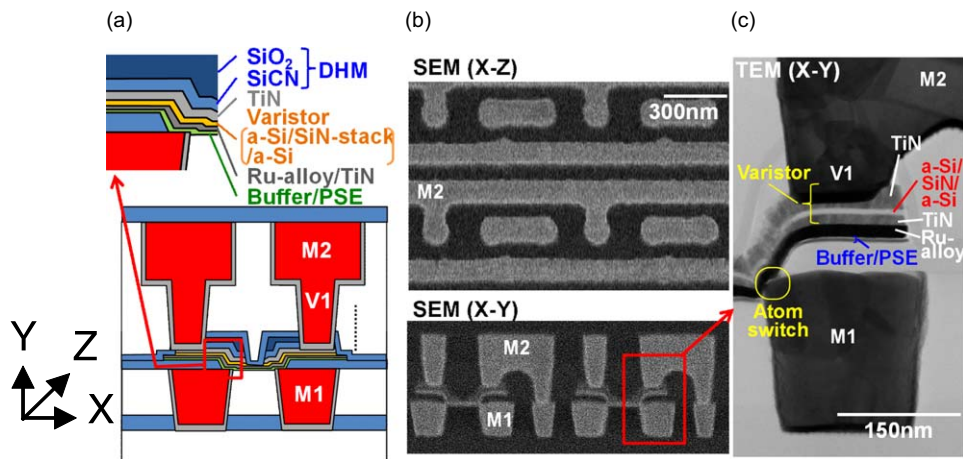


Fig. 3. (Color online) Via-switch. (a) a cross-sectional illustration of via-switch, (b) SEM images, and (c) TEM image of fabricated via-switch. © [2016] IEEE. Reprinted, with permission, from (Ref. 5).

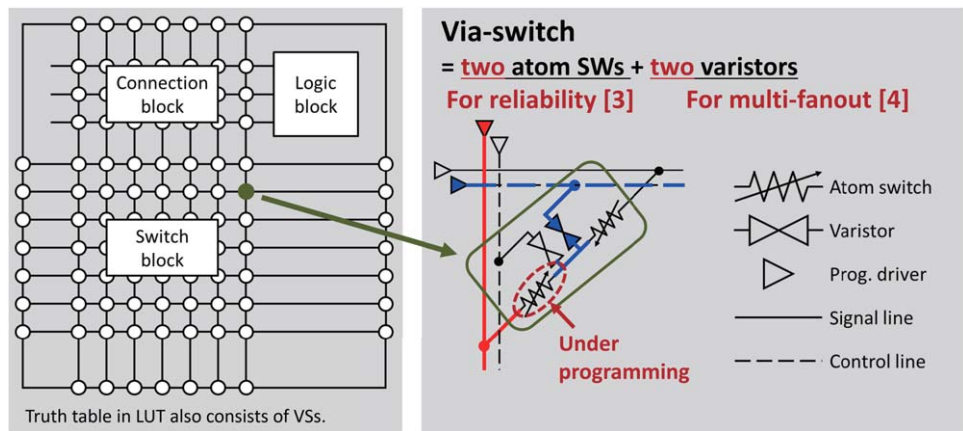


Fig. 4. (Color online) Via-switch crossbar structure.

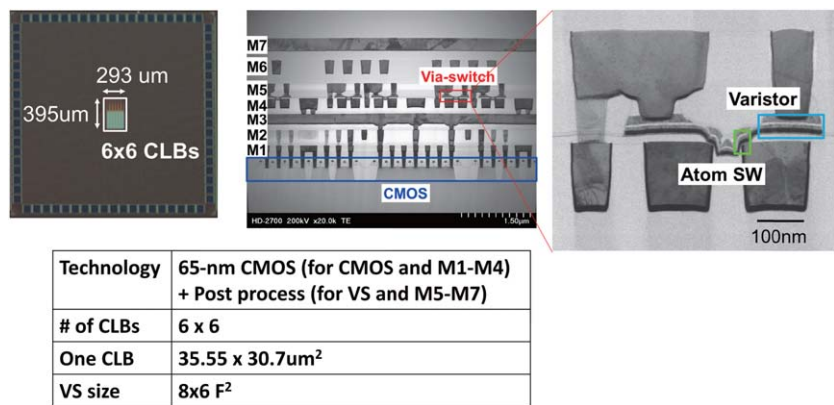


Fig. 5. (Color online) Die micrograph, TEM images, and specification of VS-FPGA. © [2020] IEEE. Reprinted, with permission, from (Ref. 10).

also presents a structure that enables selective repeater insertion for long wires.

4.2. Extension to AI applications

AI applications often rely on NN computation. Meanwhile, NN computation is computation-intensive, and hence developing hardware dedicated for NN computation is widely studied. Besides, four issues should be considered in the hardware design. The first issue is that novel network structures are released very frequently, and designers need to catch up with those in a short time. The second issue is that large memory is necessary for network information, and then many memory-

saving algorithms are under development. Thirdly, tremendous multiply-accumulate (MAC) operations are necessary, and hence massive parallel computation is indispensable. Fourth, huge memory accesses are necessary to read input data and store output data, limiting parallel memory access and consequent parallel computation. The first and second issues can be mitigated by using FPGA as the design platform since FPGA enables a shorter design and delivery time. The FPGA architecture should resolve the remaining third and fourth issues.

Assuming an application that requires large-scale parallelization of product-or-accumulate operations such as

© 2022 The Author(s). Published on behalf of
SM0804-3 The Japan Society of Applied Physics by IOP Publishing Ltd

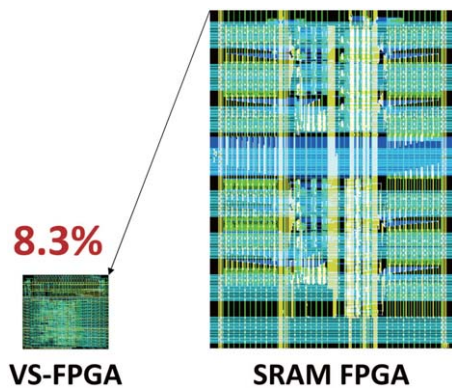


Fig. 6. (Color online) Area comparison between SRAM FPGA and VS-FPGA.

convolutional NN (CNN), we take a strategy that tiles arithmetic logic and SRAM uniformly, as shown Fig. 7. The weights in the NN are provided from the adjacent SRAMs to the arithmetic logic. Also, local data tracks facilitate systolic array organization. With this strategy, we can enable near-memory massively parallel computing.

To quantitatively predict the performance in the next subsection, we designed an FPGA architecture that exploits via-switch crossbars for AI applications, which is depicted in Fig. 8.¹⁰ There are two types of CLB, which are SRAM_CLB and Arith_CLB, and they are uniformly tiled for achieving local data movement between memories and arithmetic units, as suggested in Fig. 7. Via-switch consumes no front end of the line (FEOL), and hence SRAM and arithmetic circuits are packed under crossbars. To balance the BEOL crossbar area and FEOL arithmetic circuit area, two Arith_CLBs contain one arithmetic circuit block. For efficient systolic array implementation, a local data track between adjacent Arith_CLBs is equipped in the FEOL layer. These features eliminate long-distance communication and enable near-memory computing for higher energy efficiency and smaller latency. LUT is mainly responsible for finite-state machine implementation. In addition to popular arithmetic operations, such as MAC, in AI applications, word-size multiplexers (MUXes) can be implemented in Arith_CLB, which reduces LUT usage and improves compatibility with high-level synthesis, which will be discussed in Sect. 5. We implemented the SRAM_CLB and Arith_CLB with Verilog and laid them out with commercial physical synthesis flow using several custom cells in 65 nm technology.

4.3. Performance prediction

Figure 9 shows the estimated performance, where the x-axis is the computation density defined as the number of

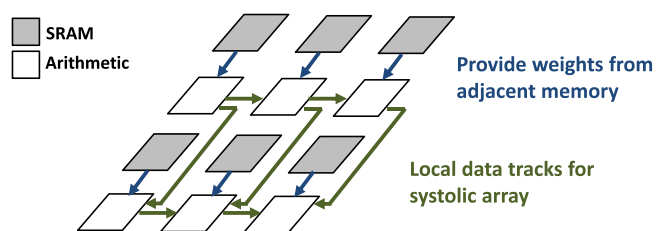


Fig. 7. (Color online) Strategy for near-memory computing. SRAMs and arithmetic units are tiled uniformly for enabling local weight supply and systolic array organization.

operations per second per area, and the y-axis is the energy efficiency. For comparison, we also implemented multiplexer-based SRAM_CLB and Arith_CLB composing a MUX-FPGA. Both the axes are normalized by those of the MUX-FPGA in the 65 nm process. See (Refs. 9,10,19) for the circuit modeling and performance estimation. We can see the VS-FPGA in 65 nm attains $5\times$ energy efficiency improvement and $29\times$ computation density improvement compared with the 65 nm MUX-FPGA. When scaling to 28 nm, the energy efficiency and density improve further. In 7 nm,²⁰ an additional $11\times$ energy efficiency improvement and $54\times$ density improvement are expected. The VS-FPGA can improve the performance as the technology node advances.

5. Application mapping flow

We have developed an automated application mapping flow in Fig. 10 for the above-mentioned architecture, which takes C/C++ as input and realizes the application through behavioral synthesis (CyberWorkBench (CWB)²¹), logic synthesis (ABC²²), and place and route. We have developed a small-area LUT that selects signals from A and not-A in addition to 0, and 1 using the via-switch feature,⁹ and developed a delay-optimized mapping method for the LUT.²³ Furthermore, we developed a method for proper placement of blocks with different granularity, namely, arithmetic operators and LUTs.²⁴ We have also studied a placement method that takes into account the carry signal between arithmetic units²⁵ and a routing delay analysis method suitable for VS-FPGAs.²⁶

Using the above design flow, we evaluated the effect of area reduction by bidirectional wiring, targeting imaging applications, and found that the area could be reduced by up to 21.7% by reducing the number of required wiring tracks.⁹

One of the significant differences between VS-FPGAs and existing SRAM FPGAs is that the number of ABs (arithmetic blocks) is relatively larger than that of LBs (logic blocks). To efficiently utilize the computing resources on VS-FPGAs, it is necessary to use AB for control circuits such as MUXes and counter circuits, which were conventionally implemented using LBs. Furthermore, when MUXes are implemented with many LBs, wiring with a large fan-out is generated, resulting in a signal delay increase. We have developed an algorithm to extract elemental circuits that contribute significantly to the reduction of the number of CLBs and fan-outs by mapping them to ABs.²⁷ Technology mapping using the proposed algorithm on benchmark circuits such as tensor multiplication circuits and FFTs for machine learning shows that the number of CLBs can be reduced by 30% to 50%. At the same time, the maximum number of fan-outs can be reduced by 12% to 87%.

6. Via-switch programming

VS-FPGAs use a crossbar circuit with via-switches at the intersections of vertical and horizontal signal wires to switch wiring connections. In this structure, a shared signal wiring is used for programming the switches.

Figure 11 shows the programming procedure for via-switches. In each step, one atom switch is rewritten to the ON state. A pair of intersecting signal and control lines are used to rewrite the switch, and the rewriting driver gives high

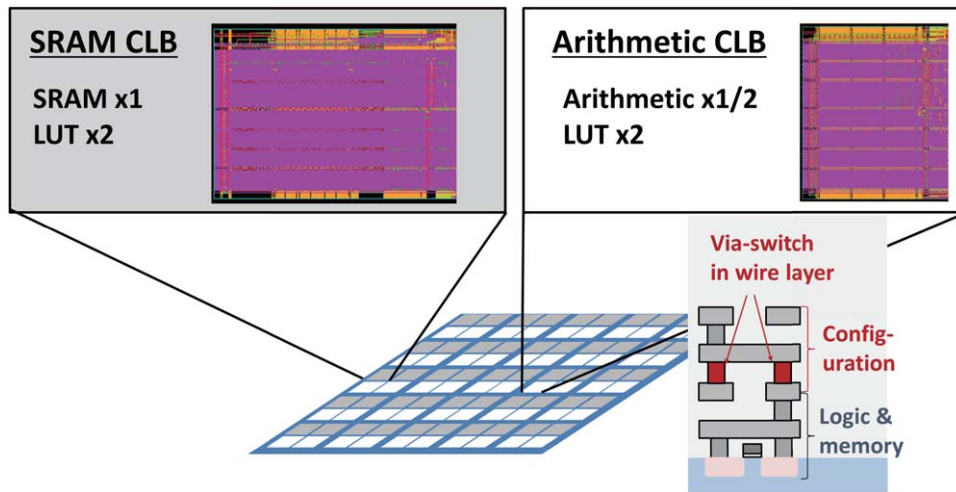


Fig. 8. (Color online) VS-FPGA architecture for AI applications.

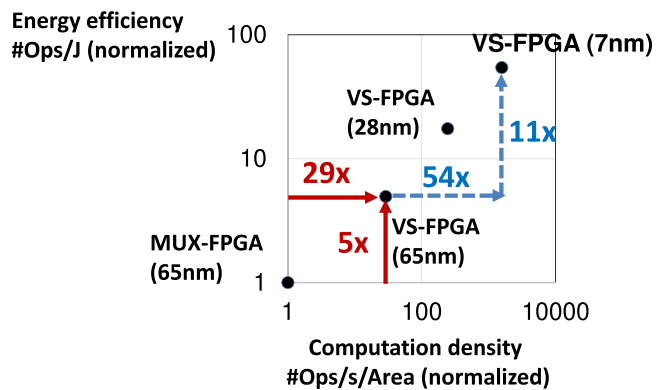


Fig. 9. (Color online) Comparison of computation density and energy efficiency between 65 nm MUX-FPGA, 65 nm VS-FPGA, 28 nm VS-FPGA, and 7 nm VS-FPGA. © [2020] IEEE. Reprinted, with permission, from (Ref. 10).

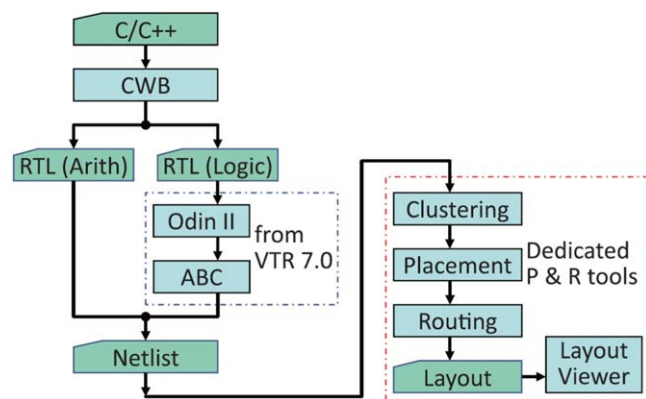


Fig. 10. (Color online) Developed application mapping flow for VS-FPGA. © [2018] IEEE. Reprinted, with permission, from (Ref. 9).

potential to the signal line and low potential to the control line. Steps (1) and (2) show that the two atom switches that compose the lower-left via-switch can be turned on correctly. Subsequent steps (3) and (4) program the upper-left via-switch to the ON state. However, the programming of the lower-right via-switch in the next step (5) cannot be performed correctly. This is because the programming signal is routed to the upper-right through the lower-left and upper-left via-switches that are already in the ON state, and the

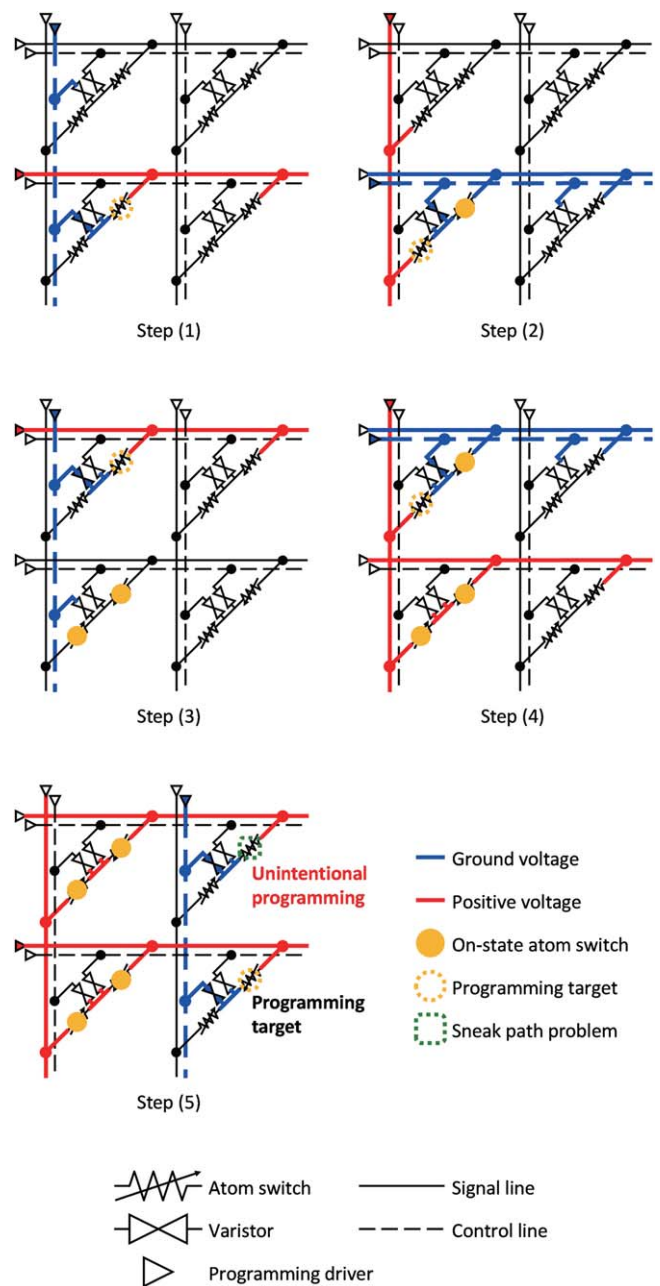


Fig. 11. (Color online) Via-switch crossbar programming and sneak-path problem. © [2018] IEEE. Reprinted, with permission, from (Ref. 28).

© 2022 The Author(s). Published on behalf of

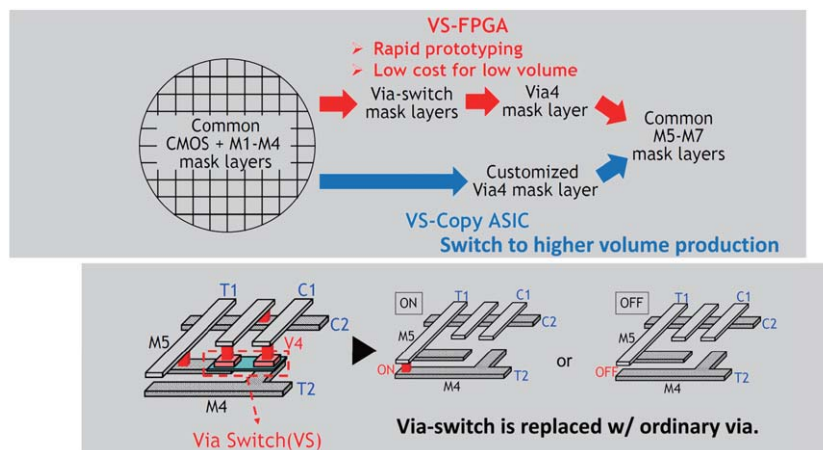


Fig. 12. (Color online) VS copy scheme for low-cost migration from VS-FPGA to ASIC.

voltage is applied to the unintended atom switch. This phenomenon of rewriting a switch other than the targeted one due to the detour of the programming signal is called the sneak-path problem.

Conventionally, the sneak-path problem was avoided by a constraint that allowed multiple switches to be turned on only in either the vertical or horizontal direction.⁹⁾ In other words, the sneak-path problem was avoided at the expense of wiring flexibility. Later, a detailed analysis of the conditions under which the sneak-path problem occurs proved that there exists a programming order of via-switches that avoids the sneak-path problem for any configuration without loops.²⁸⁾ We have also proposed a partial reconfiguration method that achieves the minimum number of switch programming steps while avoiding the sneak-path problem.²⁹⁾ This method contributes to extending the via-switch lifetime and fast reconfiguring of the VS-FPGA.

7. Discussion

As discussed in Sects. 2–4, the VS-FPGA improves the performance and energy efficiency. We here discuss the remaining weak point in Fig. 1; the unit cost.

We have proposed a VS-copy scheme to overcome the unit cost issue that commonly applies to FPGA. This VS-copy scheme enables rapid and low-cost migration from VS-FPGA to ASIC for applications that have been verified with VS-FPGAs. Figure 12 illustrates the scheme that ON via-switches are replaced with regular metal vias, and all the masks except for the V4 layer are the same as those in the VS-FPGA, which enables low-cost ASIC conversion. We confirmed that the chip manufactured by replacing the hard vias works as expected.

Figure 13 shows a comparison between the VS-FPGA and the ASIC. Traditionally, FPGAs were inferior to ASICs in terms of energy efficiency, performance, and chip cost, as discussed in Sect. 1. However, the via-switch brings energy efficiency and performance closer to ASICs, and the VS copy scheme improves chip cost. Thus, the VS-FPGA closes the gap between FPGA and ASIC, which is expected to enhance FPGA-based computing in quality and quantity.

8. Conclusion

This article has introduced the VS-FPGA under development and reported the status of our development to date. The high

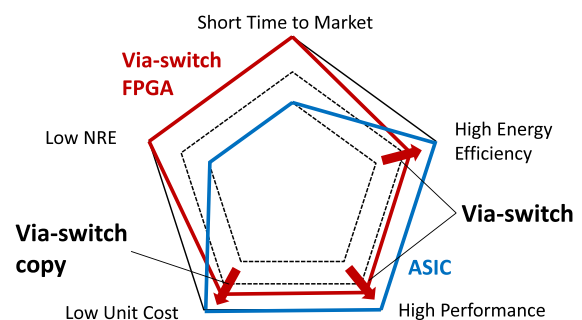


Fig. 13. (Color online) VS-FPGA versus ASIC.

energy efficiency of FPGAs is a demand of society, and VS-FPGAs can be the best design platform that quickly supports the latest AI algorithms and provides high energy efficient execution. Moreover, dedicated datapaths can be implemented according to the required performance to achieve computational acceleration with minimal overhead.

VS-FPGAs are also highly compatible with intermittent sensor operation using the non-volatility of via-switches and have a wide range of applications from the cloud to the edge and terminals. The combination of advanced CMOS technology and via-switches is promising and will be a fundamental underlying technology to support computing in decades.

The urgent future work is to fabricate the FPGA architecture for AI applications, evaluate its operation in actual devices and reveal its superiority to conventional FPGAs. Mid-term future work includes improving the varistor performance since its on-current limits the on-resistance of the via-switch, and its off-current determines the leakage power. One promising direction is to apply chalcogenide material to the varistor formation.³⁰⁾ Another important future work is to develop a test methodology for VS-FPGA shipment. Preliminary work is presented in references,^{31,32)} but its application to fabricated chips is kept as future work.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR1432, Japan. The authors thank all the contributors including students, researchers and engineers who are involved in this project.

ORCID iDs

Masanori Hashimoto  <https://orcid.org/0000-0002-0377-2108>

Xu Bai  <https://orcid.org/0000-0002-7478-8705>

Naoki Banno  <https://orcid.org/0000-0003-0052-2434>

Munehiro Tada  <https://orcid.org/0000-0002-1015-2222>

Jaehoon Yu  <https://orcid.org/0000-0001-6639-7694>

Hidetoshi Onodera  <https://orcid.org/0000-0001-5198-0668>

Takashi Imagawa  <https://orcid.org/0000-0002-1131-0800>

Hiroyuki Ochi  <https://orcid.org/0000-0002-9075-6711>

Yukio Mitsuyama  <https://orcid.org/0000-0001-8151-0085>

- 1) I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **26**, 203 (2007).
- 2) M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3-D FPGA," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **26**, 216 (2007).
- 3) K. Guo, W. Li, K. Zhong, Z. Zhu, S. Zeng, S. Han, Y. Xie, P. Debacker, M. Verhelst, and Y. Wang, Neural Netw. Accel. Comparison Available: <https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/>.
- 4) N. Banno et al., "A novel two-varistors (a-Si/SiN/a-Si) selected complementary atom switch (2V-1CAS) for nonvolatile crossbar switch with multiple fan-outs," Technical Digest of IEEE Int. Electron Devices Meeting (IEDM), p. 32, 2015.
- 5) N. Banno, M. Tada, K. Okamoto, N. Iguchi, T. Sakamoto, H. Hada, H. Ochi, H. Onodera, M. Hashimoto, and T. Sugibayashi, "50 × 20 Crossbar Switch Block (CSB) with Two-Varistors (a-Si/SiN/a-Si) Selected Complementary Atom Switch for a Highly-Dense Reconfigurable Logic," Technical Digest of IEEE Int. Electron Devices Meeting (IEDM), 2016, 10.1109/IEDM.2016.7838431.
- 6) N. Banno, K. Okamoto, N. Iguchi, H. Ochi, H. Onodera, M. Hashimoto, T. Sugibayashi, T. Sakamoto, and M. Tada, "Low-power crossbar switch with two-varistors selected complementary atom switch (2V-1CAS; Via-Switch) for nonvolatile FPGA," *IEEE Trans. Electron Devices* **66**, 3331 (2019).
- 7) M. Tada, K. Okamoto, T. Sakamoto, M. Miyamura, N. Banno, and H. Hada, "Polymer solid-electrolyte switch embedded on CMOS for nonvolatile crossbar switch," *IEEE Trans. Electron Devices* **58**, 4398 (2011).
- 8) M. Tada, T. Sakamoto, M. Miyamura, N. Banno, K. Okamoto, N. Iguchi, and H. Hada, "Improved off-state reliability of nonvolatile resistive switch with low programming voltage," *IEEE Trans. Electron Devices* **59**, 2357 (2012).
- 9) H. Ochi et al., "Via-switch FPGA: highly-dense mixed-grained reconfigurable architecture with overlay via-switch crossbars," *IEEE Trans. VLSI Syst.* **26**, 2723 (2018).
- 10) M. Hashimoto et al., "Via-Switch FPGA: 65 nm CMOS implementation and architecture extension for AI applications," Technical Digest of Int. Solid-State Circuits Conf. (ISSCC), p. 502, 2020.
- 11) X. Bai et al., "1.5× energy-efficient and 1.4× operation-speed via-switch FPGA with rapid and low-cost ASIC migration by via-switch copy," Technical Digest of VLSI Symp. on Technology, 2020, 10.1109/VLSITechnology18217.2020.9265046.
- 12) X. Bai et al., "Via-Switch FPGA: 65nm CMOS Implementation and Evaluation," *IEEE J. Solid-State Circuits* accepted.
- 13) M. Miyamura, M. Tada, T. Sakamoto, N. Banno, K. Okamoto, N. Iguchi, and H. Hada, "First demonstration of logic mapping on nonvolatile programmable cell using complementary atom switch," Technical Digest of IEEE Int. Electron Devices Meeting (IEDM) 2012, 10.1109/IEDM.2012.6479020.
- 14) M. Miyamura, T. Sakamoto, M. Tada, N. Banno, K. Okamoto, N. Iguchi, and H. Hada, "Low-power programmable-logic cell arrays using nonvolatile complementary atom switch," Proc. Int. Symp. on Quality Electronic Design (ISQED), p. 330, 2014, 10.1109/ISQED.2014.6783344.
- 15) S. Tanachutiwat, M. Liu, and W. Wang, "FPGA based on integration of CMOS and RRAM," *IEEE Trans. Very Large Scale Integr. Syst.* **19**, 203 (2011).
- 16) P.-E. Gaillardon, D. Sacchetto, G. B. Beneventi, M. H. B. Jamaa, L. Perniola, F. Clermidy, I. O'Connor, and G. De Micheli, "Design and architectural assessment of 3-D resistive memory technologies in FPGAs," *IEEE Trans. Nanotechnol.* **12**, 40 (2013).
- 17) J. Cong and B. Xiao, "FPGA-RPI: a novel FPGA architecture with RRAM-based programmable interconnects," *IEEE Trans. Very Large Scale Integr. Syst.* **22**, 864 (2014).
- 18) X. Tang, P.-E. Gaillardon, and G. De Micheli, "A high-performance low-power near-Vt RRAM-Based FPGA," Proc. Int. Conf. Field-Programmable Technol. (FPT), p. 207, 2014.
- 19) T. Higuchi, T. Ishihara, and H. Onodera, "Performance modeling of VIA-switch FPGA for device-circuit-architecture co-optimization," Proc. IEEE Int. System-on-Chip Conf. (SOCC), p. 112, 2018, 10.1109/SOCC.2018.8618503.
- 20) L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: A 7-Nm finFET Predictive Process Design Kit," *Microelectron. J.* **53**, 105 (2016).
- 21) "CyberWorkBench™," 2022. [Online]. Available: <https://www.nec.com/en/global/prod/cwb/index.html>.
- 22) "ABC: System for Sequential Logic Synthesis and Formal Verification," 2022. [Online]. Available: <https://github.com/berkeley-abc/abc>.
- 23) T. Higashi and H. Ochi, "Area-efficient LUT-like Programmable Logic Using Atom Switch and its Delay-optimal Mapping Algorithm," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E100-A**, 1418 (2017).
- 24) T. Kishimoto, W. Takahashi, K. Wakabayashi, and H. Ochi, "Range limiter using connection bounding box for SA-based placement of mixed-grained reconfigurable architecture," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E99-A**, 2328 (2016).
- 25) K. Honda, T. Imagawa, and H. Ochi, "Placement algorithm for mixed-grained reconfigurable architecture with dedicated carry chain," Proc. Int. System-on-Chip Conf. (SOCC), 2017, 10.1109/SOCC.2017.8226012.
- 26) M. Hashimoto, Y. Nakazawa, R. Doi, and J. Yu, "Interconnect Delay Analysis for RRAM crossbar based FPGA," Proc. IEEE Computer Society Annual Symp. on VLSI (ISVLSI), 2018, 10.1109/ISVLSI.2018.00101.
- 27) T. Imagawa, J. Yu, M. Hashimoto, and H. Ochi, "MUX granularity-oriented iterative technology mapping for implementing compute-intensive applications on via-switch FPGA," Proc. Design, Automation and Test in Europe Conf. (DATE), 2021, 10.23919/DATE51398.2021.9474202.
- 28) R. Doi, J. Yu, and M. Hashimoto, "Sneak path free reconfiguration of via-switch crossbars based FPGA," Proc. Int. Conf. Computer-Aided Design (ICCAD), 2018, 10.1145/3240765.3240849.
- 29) R. Doi, J. Yu, and M. Hashimoto, "Sneak Path Free Reconfiguration with Minimized Programming Steps for Via-Switch Crossbar Based FPGA," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **39**, 2572 (2020).
- 30) H. Numata, N. Banno, K. Okamoto, N. Iguchi, H. Hada, M. Hashimoto, T. Sugibayashi, T. Sakamoto, and M. Tada, "Characterization of chalcogenide selectors for crossbar switch used in nonvolatile FPGA," Proc. Silicon Nanoelectronics Workshop, 2019, 10.23919/SNW.2019.8782960.
- 31) R. Doi, X. Bai, T. Sakamoto, and M. Hashimoto, "Fault diagnosis of via-switch crossbar in non-volatile FPGA," Proc. Design, Automation and Test in Europe Conf. (DATE), 2020, 10.23919/DATE48585.2020.9116217.
- 32) R. Doi, X. Bai, T. Sakamoto, and M. Hashimoto, "A fault detection and diagnosis method for via-switch crossbar in non-volatile FPGA," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **103-A**, 1447 (2020).