Via-Switch FPGA: Highly Dense Mixed-Grained Reconfigurable Architecture With Overlay Via-Switch Crossbars

Hiroyuki Ochi, *Member, IEEE*, Kosei Yamaguchi, Tetsuaki Fujimoto, Junshi Hotate, Takashi Kishimoto, Toshiki Higashi, Takashi Imagawa, *Member, IEEE*, Ryutaro Doi, *Student Member, IEEE*, Munehiro Tada, *Senior Member, IEEE*, Tadahiko Sugibayashi, Wataru Takahashi, Kazutoshi Wakabayashi, *Member, IEEE*, Hidetoshi Onodera, *Fellow, IEEE*, Yukio Mitsuyama, *Member, IEEE*, Jaehoon Yu, *Member, IEEE*, and Masanori Hashimoto, *Senior Member, IEEE*

***Abstract*— This paper proposes a highly dense reconfigurable architecture that introduces via-switch device, which is a nonvolatile resistive-change switch and is used in crossbar switches. Via-switch is implemented in back-end-of-line layers only, and hence the front-end-of-line (FEoL) layers under the crossbar can be fully exploited for highly dense logic blocks. The proposed architecture uses the FEoL layers for fine-grained lookup tables and coarse-grained arithmetic/memory units for improving performance and compatibility with various applications. A case study of application mapping shows the proposed architecture can reduce the array area by 21.7%, thanks to the bidirectional interconnection. Thanks to $18F^2$ footprint and one order of magnitude lower resistivity of via-switch compared to MOS switch, the crossbar density is improved by up to 26× and the delay and energy in the interconnection are reduced by 90% and 94% at 0.5-V operation.**

***Index Terms*— Field programmable gate array (FPGA), nonvolatile resistive-change switch, reconfigurable architecture.**

I. Introduction

RECONFIGURABLE devices, a representative of which is field programmable gate array (FPGA), are gaining their popularity as a means of integrated system implementation since nonrecurring engineering cost of application specific integrated circuits (ASICs) is elevating as the fabrication technology becomes finer and finer. However, there remains a large gap between FPGA implementation and ASIC implementation in terms of performance and energy efficiency. For example, Kuon and Rose [1] report that for circuits implemented purely using the lookup table (LUT)-based logic elements, an FPGA implementation is approximately 35 times larger and between 3.4 and 4.6 times slower on average than a standard cell implementation.

The lower performance of FPGA originates from the low area efficiency and poor interconnect performance. Lin *et al*. [2] report that 78% of the chip area is consumed for routing and only 14% of the chip area is used for logic. Due to this, the chip size becomes larger, and consequently the interconnect delay becomes longer. In addition, the interconnection in FPGA consists of a number of programmable switches and buffers. As a result, 80% of the circuit delay is occupied by interconnect delay [2]. Recently, resistive RAM (RRAM) and magnetoresistive RAM are introduced as a storage of configuration information for nonvolatile operation and radiation-hard operation [3]–[5], but these works still use MOS switches and hence, the delay problem is unchanged.

To overcome the interconnect delay problem in FPGA, [6]–[9] introduced RRAM as not only configuration memory but also programmable switch for signal transmission. The speed improvement ranging from 45% to 58% is reported in [5]–[7]. Besides, atom switch (a.k.a. nanobridge), which is a nonvolatile resistive-change switch and is integrated on back-end-of-line (BEoL) layers, has been developed as a nonvolatile programmable switch tailored for FPGA [10] and the first FPGA implementation with atom switch is fabricated and presented in [11]. The ON/OFF resistance ratio of atom switch is over 10^5, and the ON-resistance can be reduced to 200 Ω. In addition, a complementary atom switch (CAS) consisting of two atom switches in series reduces the programming voltage to 2 V and improves OFF-state reliability [10], [12]. The FPGA implementations with CAS and their silicon results are reported in [13]–[15].

However, all the above mentioned FPGA implementations with RRAM, atom switch, and CAS require one or two access transistors for every programmable switch. Even though small-footprint nonvolatile programmable switches are implemented

Manuscript received August 31, 2017; revised December 6, 2017 and February 2, 2018; accepted February 17, 2018. Date of publication March 26, 2018; date of current version November 30, 2018. This work was supported by JST CREST under Grant JPMJCR1432. *(Corresponding author: Hiroyuki Ochi.)*

H. Ochi, K. Yamaguchi, T. Fujimoto, J. Hotate, T. Kishimoto, T. Higashi, and T. Imagawa are with Graduate School/College of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan (e-mail: nanocrest@gmail.com).

R. Doi, J. Yu, and M. Hashimoto are with the Department of Information Systems Engineering, Osaka University, Suita 565-0871, Japan.

M. Tada and T. Sugibayashi are with NEC Corporation, Tsukuba 305-8501, Japan.

W. Takahashi and K. Wakabayashi are with NEC Corporation, Kawasaki 211-8666, Japan.

H. Onodera is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.

Y. Mitsuyama is with the School of Systems Engineering, Kochi University of Technology, Kami 782-8502, Japan.

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

Digital Object Identifier 10.1109/TVLSI.2018.2812914

on BEoL layers, large area is consumed by access transistors. Consequently, the area per programmable switch is determined by the access transistors. On the other hand, in the crossbar implementation, the switch density is directly related to the interconnect delay since the dense crossbar enables shorter interconnection with lower resistance and smaller capacitance. If the access transistors can be eliminated, we can significantly improve the switch density and reduce the interconnect delay and energy for signal transmission. Motivated by this, via-switch has been developed [16], which consists of CAS and two varistors. The two varistors enable device selection for programming without the access transistors. 50 × 20 crossbar fabrication and its operation are reported in [17].

This paper proposes a highly dense reconfigurable architecture that uses via-switch for crossbar implementation. We devise a bidirectional interconnect structure that can exploit the small footprint and low resistivity of via-switch. In addition, we devise a logic structure that consists of fine-grained LUTs and coarse-grained arithmetic/memory units aiming to fully exploit the transistor area under the crossbar for highly dense logic blocks (LBs). A case study of application mapping shows the proposed architecture can reduce the array area by 21.7%, thanks to the bidirectional interconnection. Furthermore, due to $18F^2$ footprint and one order of magnitude lower resistivity of via-switch compared to MOS switch, the crossbar density is improved by up to 26× and the delay and energy in the interconnection are reduced by 90% and 94% at 0.5-V operation. Compared to a preliminary publication of this paper [18], we newly discuss the programming of via-switch-based crossbar and explain how the proposed via-switch with two varistors can solve sneak path problem with an induction-based proof. We perform layout aware area estimation, revise the logic structure according to the estimated area, and map larger designs as a case study. In addition, we detail the interconnect performance evaluation at various voltages and with several ON-resistance values of via-switch.

The remainder of this paper is organized as follows. Section II reviews the structure and characteristics of via-switch and static random access memory (SRAM)-based FPGA architecture. Section III presents the concept of the proposed reconfigurable architecture followed by the proposed interconnect structure in Section IV and the proposed logic structure in Section V. Results on interconnect performance evaluation are shown in Section VI. Concluding remarks are given in Section VII.

II. Background

A. Via-Switch

Via-switch is a nonvolatile and compact switch that consists of a CAS and two varistors (2V-1CAS), and it is developed to implement a crossbar switch that can accommodate multiple fan-outs [16]. Two control lines connected to the varistors realize accurate one-by-one programming of each cross-point without access transistors. The via-switch can be integrated with a small foot print of $18F^2$. The programming of via-switch crossbar will be explained in Section II-B. The following explains the device structure, functionality, and characteristics.

TABLE I

Atom Switch-Based Device Summary

	1T-1CAS[14]	DCAS (1V-1CAS)[22]	Via-switch (2V-1CAS)[17]
Select	MOS Tr.	Varistor	Two varistors
Area	Large	Very small ($12F^2$)	Very small ($18F^2$)
Fan-out	Possible	Impossible	Possible

Select: Device to select a switch to be programmed. Area: Footprint area per switch. Fan-out: Programmability of multiple ON-state switches on a line.

The atom switch is nonvolatile and rewritable solid-electrolyte switch, and it is composed of a solid-electrolyte sandwiched between Cu and Ru electrodes. By applying a positive voltage to the Cu electrode, a Cu bridge is formed in the solid-electrolyte and the switch turns ON. When a negative voltage is applied, the Cu atoms in the bridge are reverted to the Cu electrode and then the switch turns OFF. The transition between the low resistive (ON) state and high-resistive (OFF) state is repeatable and each state is nonvolatile. The ON-resistance of the atom switch can be tuned by a programming current, and it can be down to 200 Ω [19], which is suitable for signal line switch. Turn-ON is achieved within less than 2 ns.

The CAS consists of two two-terminal atom switches connected in series with opposite direction. This complementary connection improves the device reliability. The OFF-state persists for 10 years even when a dc voltage of 1 V and ambient temperature of 85 °C are applied. The ON-state is also reliable for more than 3000 h at 150 °C [12]. The cycling endurance is up to 10 000 cycles [20].

To program each CAS in a switch array as intended, program voltages must be applied selectively to the target CAS. For this purpose, [13] used an access transistor for each CAS. For pursuing further area reduction, [21] proposed diode-selected CAS in which a varistor (or bidirectional diode) is stacked on each CAS. The role of varistor is to provide two functionalities: 1) program line isolation during normal operation and 2) program current supply during programming operation. For 1), the dc resistance of the varistor should be very high, whereas it should be low for 2). Therefore, the varistor needs to have superior nonlinear (NL) characteristic of over 10^5. One-varistor-one-CAS (1V-1CAS) structure in [21], however, suffers from sneak path problem when multifan-out configuration is attempted (see Section IV-A). To overcome the problem, via-switch has been developed that consists of a 2V-1CAS [16]. Table I summarizes these CAS-based devices.

The cross section of via-switch fabricated in 65-nm node [16] is shown in Fig. 1. For achieving NL characteristics of over 10^5, a novel nitrogen-modulated TiN/a-Si/SiN/a-Si/TiN varistor is introduced and it is stacked on the CAS in [16]. The measured characteristics of via-switch is shown in Fig. 2. Set and reset programmings of the atom switch through the varistor are confirmed. In the via-switch programming, the electrical connection/disconnection between T1 and T2 is demonstrated in Fig. 2(a). In the ON-state, high current between T1 and T2 is observed, which can be used for signal transfer. The ON/OFF current ratio of the CAS is 4.6×10^5. Fig. 2 shows the leakage current between T1 and C1 or C2. The electrical separation through the varistor is also

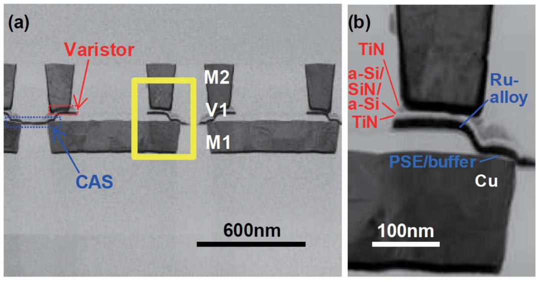


Fig. 1. Cross-sectional TEM images of via-switch integrated in 65-nm node Cu-BEoL [16]. (a) Low-magnification views. (b) High-magnification views.

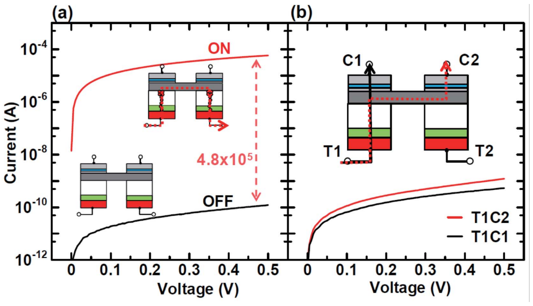


Fig. 2. Via-switch current characteristics [16]. (a) ON-state characteristics and OFF-state characteristics. ON/OFF-current ratio is 4.8×10^5. (b) OFF-state leakage current between T1 and C1 or C2.

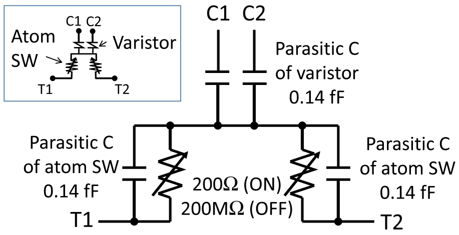
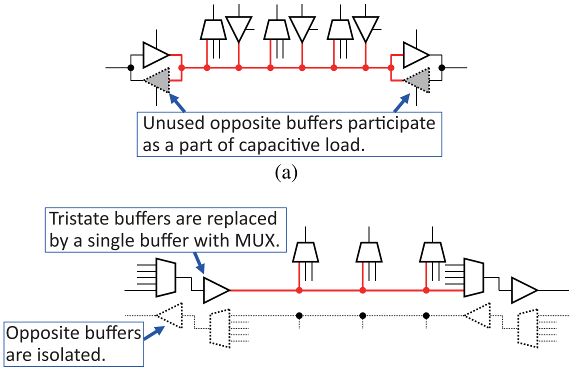

Atom SW
C1 C2
Varistor
T1
T2
C1 C2
Parasitic C of varistor 0.14 fF
Parasitic C of atom SW 0.14 fF
200Ω (ON)
200MΩ (OFF)
Parasitic C of atom SW 0.14 fF
T1
T2

Fig. 3. Equivalent circuit model of 2V-1CAS via-switch in normal operation.

demonstrated. The NL characteristics of a Si/SiN/a-Si varistor is improved with a novel triple layered SiN in [17].

Fig. 3 shows an equivalent circuit model of 2V-1CAS via-switch in normal operation. The parasitic capacitance of an atom switch is 0.14 fF, and it is connected in parallel to the variable resistor whose value is 200 Ω in ON-state and 200M Ω in OFF-state. In the normal operation, the varistor works as a capacitor to the control line. The varistor capacitance is also 0.14 fF. The small parasitic capacitance in addition to low ON-resistance is the advantage of via-switch. It should be noted that the varistor is still under development and the current varistor [17] cannot provide 5-mA programming current which makes the ON-resistance of atom switch 200 Ω. On the other hand, currently, similar selector devices are widely studied and recent improvements are remarkable. We therefore evaluate the potential performance supposing the ON-resistance of an atom switch is 200 Ω in the following. For estimating the impact of ON-resistance on performance, on the other hand, we will evaluate interconnection delay and energy in the range of 200 to 1200 Ω in Section VI, where the current varistor [17] can provide 250 μA and achieve ON-resistance of 1 kΩ.


Unused opposite buffers participate as a part of capacitive load.
Tristate buffers are replaced by a single buffer with MUX.
Opposite buffers are isolated.

(a)

(b)

Fig. 4. Bidirectional and unidirectional wire segment. (a) Bidirectional wire segment with tristate buffers. (b) Unidirectional wire segment with single driver [22].

B. SRAM-Based FPGA Architecture

FPGA is a well-established class of reconfigurable integrated circuit. Typically, it consists of programmable LBs, programmable I/O blocks, and programmable routing resources. The LBs are arranged in an array, and routing channels run between the blocks in island architecture. The LB typically consists of LUTs and optional D-flip-flops (D-FFs). The k-input LUT (k-LUT) consists of 2^k SRAM elements and a 2^k-to-1 multiplexer. Coarser grained LBs such as multiplier-accumulator (MAC) unit and block memory are also embedded in the array. At each cross-point of routing channels, there is a switch block (SB) to allow connections between wire segments. At the edge of each LB, there is a connection block to allow connection between wire segments and ports of the blocks. Configuration of routing resources is also retained by on-chip SRAM elements.

SRAM-based FPGAs in early times use bidirectional wire segments with tristate buffers as in Fig. 4(a). Bidirectional interconnect is more flexible than unidirectional counterpart. For example, when only 30% tracks of a channel is occupied by one direction, the remaining 70% tracks can be used for opposite direction in case of bidirectional architecture. This architecture, however, has been reported to be inefficient in terms of area, delay, and area-delay product compared with unidirectional wire with single (nontristate) buffer as in Fig. 4(b) [22]. The main reason is that at least 50% of tristate buffers in the bidirectional interconnect are unused once configured while these buffers participate as a part of load capacitance, degrading area, power, and delay. Moreover, using single (nontristate) buffer improves the performance since regular buffer provides better drive strength. Note that the above argument assumes that the programmable "switch" is implemented by CMOS devices (e.g., pass gate, tristate buffer, and/or MUX), which typically suffer from parasitic capacitance (≈1 fF/Tr. in 65-nm technology).

III. Overview of Proposed Architecture

With the via-switch explained in Section II-A, we can implement a crossbar on BEoL layers without transistors, where the crossbar provides programmable

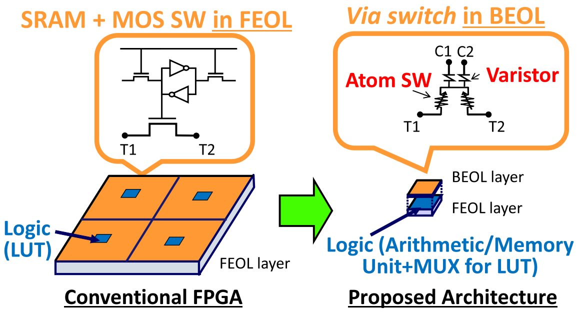


Fig. 5. Concept of proposed reconfigurable architecture.

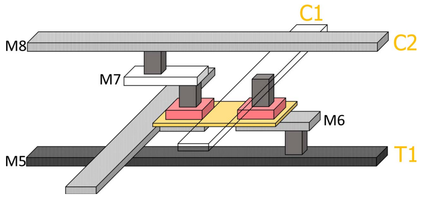


Fig. 6. 3-D view of $18F^2$ 2V-1CAS via-switch implemented between M6 and M7 layers.

interconnection and it is the most important component that determines the performance and integration density of the reconfigurable device. A peripheral circuit for programming will be discussed in Section IV-A. In addition to the crossbar, the memory for LUT can be implemented with the via-switch, where the LUT structure will be presented in Section V.

Using the crossbar and LUT memory with via-switch, the front-end-of-line (FEoL) layers under the overlay crossbar and LUT memory can be fully used for logic implementation, as illustrated in Fig. 5. When we integrate the via-switch between M6 and M7 layers as depicted in Fig. 6, the metal layers of M1 to M4 can be used for the logic implementation. In conventional FPGA, a 6T SRAM cell and a pass gate, which can be a single transistor or complementary CMOS transistors, are necessary for each intersection in the crossbar. Other SRAM cells are used for storing LUT values. In this case, the most of transistor area is consumed by the crossbar implementation, and a small portion of the area is used for logic implementation. Comparing the conventional FPGA, we can improve the logic density significantly.

The most significant advantage of the dense implementation is the shorter interconnection between the LBs. In recent advanced technologies, the interconnect delay is much larger than the gate delay, and hence the shorter interconnection is expected to provide considerable performance improvement. Furthermore, as mentioned in Section II-A, the ON-resistance of CAS can be reduced to 400 Ω (200 Ω for each atom switch), which is much lower than the resistance of the smallest transistor. Thanks to these, the interconnect delay is expected to be significantly reduced. To maximize the delay reduction effect, we devise a bidirectional interconnect structure (Fig. 7) with selective repeater insertion, which will be shown in Section IV-C. The amount of performance improvement will be experimentally shown in Section VI.

On the other hand, we have observed that, even using the crossbar with via-switch, the crossbar area on the BEoL layers is larger than the logic (LUT multiplexer), and the transistor area could be unused. It is clearly wasteful that the transistor area remains unused. The most area-efficient implementation should fully use the transistor area under the overlay crossbar for improving the performance and enriching the functionality. Motivated by this, we adopt a mixed grained reconfigurable architecture (MGRA) whose basic elements include both fine-grained logic resources (e.g., LUT) and coarse-grained counterparts (e.g., arithmetic/memory unit). This logical architecture design is discussed in Section V.

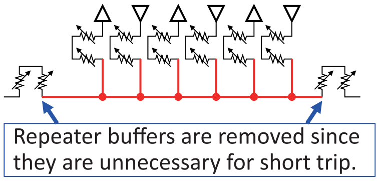

Repeater buffers are removed since they are unnecessary for short trip.

Fig. 7. Bidirectional wire segment with CAS.

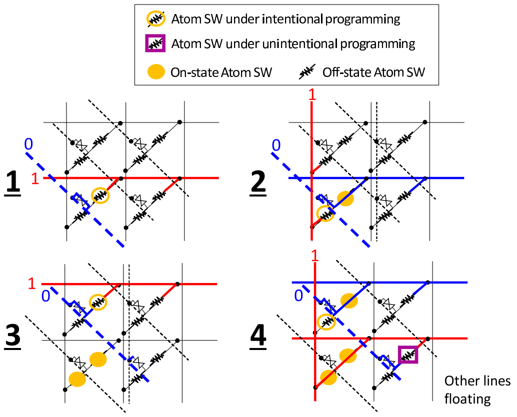

Atom SW under intentional programming
Atom SW under unintentional programming
On-state Atom SW
Off-state Atom SW
1
0
1
2
1
0
3
1
0
4
1
0
Other lines floating

Fig. 8. Unintentional programming problem of 1V-1CAS via-switch [21] for multiple fan-outs due to sneak path (2 × 2 crossbar example).

IV. Interconnect Structure

A. Crossbar Programming

When using varistor for device selection in programming, we need to pay attention to eliminate a sneak path problem. To explain the sneak path problem, a previous work in which via-switch is implemented with a CAS and a single varistor (1V-1CAS) [21] is given below as an example. A similar 1D2R device with one diode and two variable resistors is recently reported in [23]. Fig. 8 shows the crossbar structure in [21], where signal lines are aligned horizontally and vertically, and control lines are routed diagonally. This crossbar structure can accept the programming in which at most one intersection can be turned ON for every horizontal signal line and for every vertical signal line, which means multiple fan-outs are not allowed. Let us explain what happens if we try to turn ON multiple 1V-1CAS via-switches on a vertical signal line. Fig. 8 illustrates the programming steps, where an atom switch is turned ON at each step. Positive voltage (noted as "1") is given to one of the signal lines and ground voltage ("0") is given to one of the control lines. Other lines are floated.

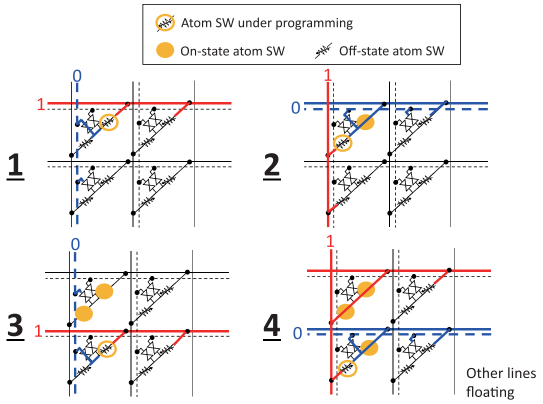


Fig. 9. 2V-1CAS via-switch programming for multiple fan-outs (2 × 2 crossbar example).

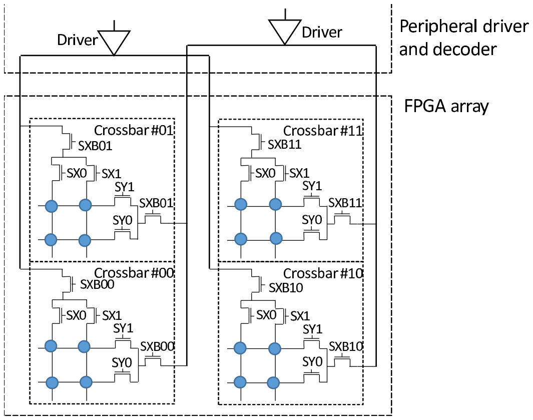


Fig. 10. Programming circuitry. Four 2 × 2 crossbars are placed in an array. Only signal lines are depicted for simplicity.

At steps 1 and 2, the two atom switches composing the via-switch located at the bottom left is turned ON. Next, we try to turn ON the via-switch at the top left. One of the atom switches is turned ON at step 3. However, there is a conflict in the via-switch at the bottom right at step 4. The atom switch marked by a purple rectangle is under programming unintentionally since the positive voltage is provided through the ON via-switch at the bottom left. Such an unintentional signal path that is generated by ON switches is called a sneak path, and the unintentional programming due to the sneak path is sneak path problem.

Struggling to solve the sneak path problem, it is found that 1V-1CAS via-switch is not capable of multiple fan-outs. Therefore, to overcome the sneak path problem, via-switch has been developed that consists of a 2V-1CAS [16], which was introduced in Section II-A. The crossbar structure with the 2V-1CAS via-switch and its programming steps are illustrated in Fig. 9. Two control lines are routed vertically and horizontally. When programming an atom switch in a crossbar, one of the signal lines is connected to high voltage and one of the control lines is connected to ground voltage. The two atom switches composing the CAS at the top left are turned ON at steps 1 and 2. At steps 3 and 4, the CAS at the bottom left are successfully turned ON without any unintentional programming. We can see that no sneak path problem is found in this crossbar structure. We also confirm that these ON CAS switches can be turned OFF. This programming operation is verified with a fabricated 50 × 20 crossbar [17]. Here, it should be noted that the multiple fan-outs, i.e., multiple ON via-switches can be allowed in one direction. For example, in the example of Fig. 9, the multiple ON via-switches are allowed in the vertical direction. In the horizontal direction, only a single via-switch can be turned ON. If we try to turn ON multiple via-switches on the horizontal signal line, the sneak path problem arises. We prove mathematically that this programming constraint can avoid speak path problem in Section IV-B.

A possible structure of programming circuit is illustrated in Fig. 10, where four 2×2 crossbars are placed in an array. For each signal and control wire in a crossbar, pass transistors are attached to provide programming voltage and current. Here, for simplifying the figure, only signal wires are depicted. Similar structure is necessary to provide the voltages to control lines. The programming voltage and current are provided by the drivers located at the peripheral of the entire array. The enable signals for the pass transistors are also provided from the decoder at the peripheral.

The drivers at the peripheral need high-voltage (e.g., 3.3 V) transistors, but the minimum number of drivers for the entire array are two; one for high-programming voltage and the other for ground voltage. On the other hand, the pass transistors in each crossbar can be implemented with middle-voltage (e.g., 1.8 V) transistors since the number of programming is limited to up to 10 000 due to the endurance of atom switch [20] and consequently the stress time of the pass transistors is very short. Similar discussion is applicable to the input transistors of the datapath. The reliability issue originating from the programming is important and careful investigation is necessary, but it should be noted that the similar programming circuit is already developed for atom switch FPGA and successful programming is demonstrated with fabricated chips [15]. When the pass transistors are implemented with NMOS accepting Vth drop, additional well separation is not necessary and hence the area overhead due to well separation inside the crossbar can be eliminated. During the programming, only the programming circuits are active and the core supply voltage is not given to the datapath logic, and hence the disturbance due to programming current is not an issue. In addition to the above, we need to provide control signals that select pass transistors to be ON. The control signal is decoded and distributed to all the crossbars. To select the crossbar under programming, one-hot select signal is routed to each crossbar. These decoders are also located at the peripheral similar to [15] and hence the area overhead is limited.

As mentioned in Section II-A, 2 ns is necessary to program a via-switch. The total programming time is 2 ns multiplied by the number of via-switches to be ON when serial programming scheme is adopted. If this programming time is not acceptable, the parallel programming should be chosen. In this case, the number of drivers needs to be increased. The number of parallel programming could be limited by the total amount of simultaneous programming current.

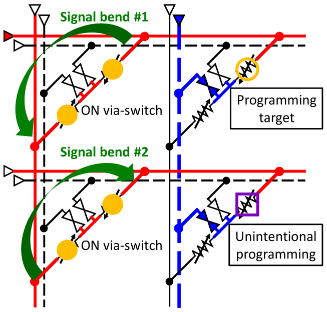


Fig. 11. Multiple bends of programming signal causes sneak path problem.

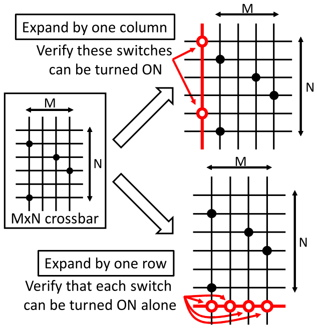

Expand by one column
Verify these switches can be turned ON
M
N
MxN crossbar
Expand by one row
Verify that each switch can be turned ON alone

Fig. 12. Crossbar expansion supposed in induction-based proof.

B. Proof of Sneak Path Avoidance

We mentioned that there is no sneak path problem in the crossbar structure with the 2V-1CAS via-switch under a constraint that the multiple ON via-switches are allowed only in one direction. Following the procedure below, we prove it using induction. Besides, we hereafter refer to this constraint as programming constraint.

1) Prove that sneak path problem does not arise in the programming of 1×1 crossbar (i.e., single via-switch). This is self-evident.
2) Assume that sneak path problem does not arise at each programming step for any configuration patterns of $M \times N$ crossbar.
   a) Prove that sneak path problem does not arise at each programming step for any configuration patterns in $(M + 1) \times N$ crossbar.
   b) Prove that sneak path problem does not arise at each programming step for any configuration patterns in $M \times (N + 1)$ crossbar.
3) Once procedures 1 and 2 are proved, there is no sneak path problem in the programming of any-sized crossbar.

The atom switch at the intersection, when a positive voltage is given to the signal line and a ground voltage is given to the control line, is turned ON. When the number of such intersections is two or more, sneak path problem arises. Here, for the proof, we focus on the number of bends of the programming signal given to the signal line. When the number of bends of programming signal given to the signal line is two or more, there are two or more intersections where the corresponding atom switches become ON as shown in Fig. 11. In other words, the programming signal must be bent twice or more by ON via-switches to cause sneak path problem. We will show that the number of signal bends is at most one in procedure 2. The following assumes that the multiple ON via-switches are allowed only in the vertical direction without losing generality since the crossbar has a symmetrical structure. Also, turning ON and OFF a via-switch is a symmetrical operation, and by swapping the voltages given to the signal line and control line, the same proof can be easily achieved. Therefore, the following proof only considers the situation in which via-switches are turned ON.

In procedure 2-a, we have to prove that there is no sneak path problem in any possible configuration patterns newly added by the horizontal one-column crossbar extension. Here, the number of ON via-switches in each row is zero or one because the multiple ON via-switches in the horizontal direction are not allowed. Therefore, what we need to clarify is that for each row with no ON via-switches, the switches located at the expanded column can be turned ON without sneak path problem. The upper right of Fig. 12 illustrates this example. For programming a via-switch, we must turn ON the two atom switches, i.e., the upper atom switch connected to the horizontal signal line and the lower atom switch connected to the vertical signal line as shown in Fig. 9. When the upper atom switch is under programming, a positive voltage is provided to the horizontal signal line (e.g., step 1 in Fig. 9). Here, all other switches on the same horizontal line are OFF due to the programming constraint, and hence the signal given to the signal line never bends. On the other hand, when the lower atom switch is under programming, we use the vertical signal line (e.g., step 2 in Fig. 9). In this case, the programming signal can be bent depending on whether there are ON via-switches on the same vertical line. However, the signal never bends further because the multiple ON via-switches in the horizontal direction are not allowed. From the above, we can conclude that the total number of signal bends is at most one and hence, no sneak path problem arises in procedure 2-a.

Next, let us prove the procedure 2-b. The crossbar is expanded by one row in the vertical direction. Here, only one via-switch on the expanded row can be turned ON because the multiple ON via-switches in the horizontal direction are not allowed. Therefore, what we should verify is that any one of the switches on the expanded row can be turned ON, which is illustrated in the lower right of Fig. 12. When the upper atom switch is under programming, the programming signal given to the horizontal signal line never bends similar to the proof of procedure 2-a. When the lower atom switch is under programming, the programming signal bends at most once. From the above, the total number of signal bends is one or less, and hence no sneak path problem arises in procedure 2-b.

In summary, we have proved procedures 1 and 2, and consequently, we reach the entire proof that there is no sneak path problem in the programming of any sized crossbar under the programming constraint that multiple ON via-switches are allowed only in one direction.

C. Proposed Interconnect Structure

Thanks to the $18F^2$ $(= 6F \times 3F)$ via-switch which can be implemented on BEoL layers only, the crossbar can be

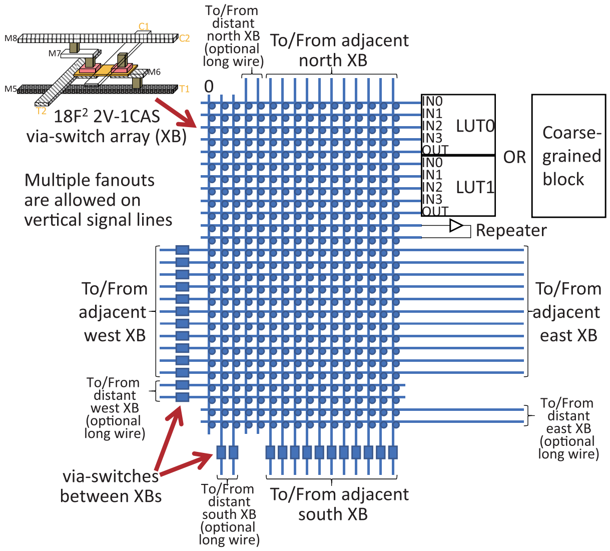

M8
C1
C2
M7
M6
M5
T1
T2
18F² 2V-1CAS
via-switch array (XB)
To/From distant north XB (optional long wire)
To/From adjacent north XB
0
IN0
IN1
IN2
IN3
OUT
LUT0
IN0
IN1
IN2
IN3
OUT
LUT1
OR
Coarse-grained block
Repeater
Multiple fanouts are allowed on vertical signal lines
To/From adjacent west XB
To/From adjacent east XB
To/From distant west XB (optional long wire)
To/From distant east XB (optional long wire)
via-switches between XBs
To/From distant south XB (optional long wire)
To/From adjacent south XB

Fig. 13. Proposed crossbar structure. At each intersection, 2V-1CAS via-switch is located. On vertical signal lines, multiple fan-outs are allowed. Control lines are omitted.

implemented compactly. For example, a 100 × 100 crossbar can be implemented in 60 μm × 30 μm in 65-nm node ($F = 100$ nm). Therefore, the interconnect resistance per crossbar is not high, and hence a repeater is not necessary for every crossbar. In addition, the via-switch is naturally capable of bidirectional signal transmission. If the signal lines are bidirectional, the routing efficiency per signal line improves and consequently the number of necessary signal lines can be reduced.

Taking these into account, we have devised the crossbar structure shown in Fig. 13. The crossbars are placed in a 2-D array. This crossbar includes a SB, which is the bottom half of the crossbar, for the vertical and horizontal lines that are connected to the adjacent crossbars. As an architecture option, some of the vertical and horizontal lines may be connected to distant crossbars to form longer wire segments, such as quad-length wires or hex-length wires as seen in commercial FPGAs. In addition, the crossbar includes input and output multiplexers to LUT, arithmetic/memory unit, and repeater, which are often called connection box and are located at the top half of the crossbar. Here, two 4-input LUTs and a repeater are depicted, but they are just an example. The interconnect structure is independent of the number of LUTs, the number of LUT inputs and the number of repeaters, and they can be changed. The signal lines that are connected to the adjacent crossbars are bidirectional, while the lines connected to LUT, coarse-grained unit, and repeater are unidirectional. The signal lines can be connected to either the next crossbars for short connection or distant crossbars for long connection. The number of signal lines and their connections need to be determined taking into account the routability and interconnect delay. This crossbar accepts the multiple ON via-switches on the vertical signal line. Thanks to this, the same signal can be given to different LUTs. More importantly, the signals coming from the east/west crossbars can be transferred to the LUT inputs, and the LUT output signals can be delivered to the east/west crossbars. Besides, via-switches are inserted between the crossbars and they are responsible for signal connection and isolation. Thanks to them, we can eliminate cross-coupled tristate buffers shown in Fig. 4(a) even for bidirectional signaling. For enabling us to provide a fixed value to LUT, coarse-grained unit and repeater, a vertical line of "0" is added.

An important feature of this crossbar structure is the existence of the repeater. The proposed interconnect structure can provide a long wire by connecting crossbars with ON via-switches. Compared to conventional FPGA, the resistance of such a long wire is much lower, but still it increases as the length becomes larger. In this case, the interconnect delay starts to be proportional to the length squared. For avoiding such a quadratic delay increase, we need to insert repeaters. This repeater insertion can be achieved by using the repeater below the LUT in Fig. 13. It should be noted that fewer and flexible repeater insertion makes it possible to take advantage of bidirectional signaling in the proposed interconnect structure. In conventional FPGA, the necessity of larger number of buffers/repeaters is another reason why unidirectional signaling is widely adopted [25].

V. Logic Structure

A. Unit Tile

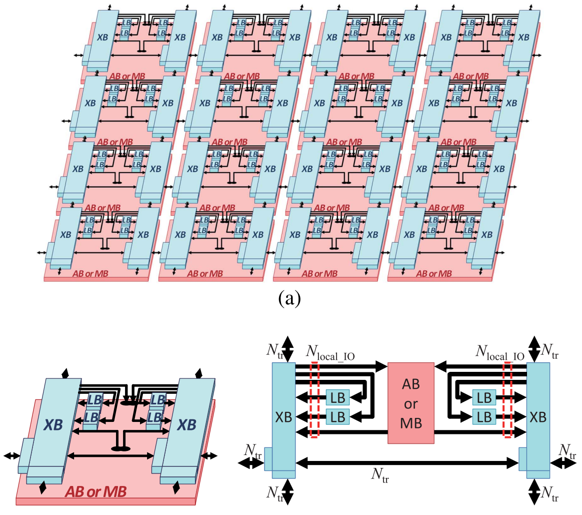
The programmable logic resource of the proposed architecture consists of both fine-grained blocks (e.g., LUT) and coarse-grained blocks (e.g., multiplier and memory) similar to modern commercial FPGAs. The proposed architecture is a 2-D array of the "unit tile" [Fig. 14(a)]. The unit tile [Fig. 14(b)] consists of two crossbar blocks (XBs), four fine-grained LBs and a coarse-grained arithmetic block (AB) or memory block (MB). Coarse-grained blocks (e.g., AB or MB) occupy larger area and pin counts than fine-grained components, and hence an AB or MB is connected to multiple XBs.

As mentioned in Section III, the transistor area under the crossbar should be fully occupied by LBs for maximizing the area efficiency. To explore suitable LBs, the following discusses the logic area and switch area that are occupied by LB, X-band AB/MB. For this purpose, we define some parameters [Fig. 14(c)]. Let N_{tr} be the number of tracks between two adjacent XBs, where for simplicity, we assume that the numbers of vertical and horizontal tracks is the same. $N_{\mathrm{local\_IO}}$ is the total number of local interconnects between an X-band the IO pins of the LBs and the AB/MB.

B. XB, LB, and AB/MB

The number of switches in an XB is given by $(N_{\mathrm{local\_IO}} + N_{\mathrm{tr}}) \times N_{\mathrm{tr}}$, and then the XB area is $(N_{\mathrm{local\_IO}} + N_{\mathrm{tr}}) \times N_{\mathrm{tr}} \times 18F^2$.

Next, we calculate the area of an LUT, which is the basic component of the LB. Conventional SRAM-based 4-input LUT consists of 16 SRAM cells and a 16-to-1 multiplexer (16-MUX). In [14], a 4-LUT is implemented using 32 CASs and a 16-MUX (say 0/1-type LUT), and in [26], an improved 4-LUT architecture using 32 CASs and an 8-MUX are



(a)

(b)

(c)

Fig. 14. Proposed architecture. (a) Array structure. (b) Unit tile (2.5-D view). (c) Unit tile (logical view).

TABLE II
COMPARISON OF LUT AREA

		SRAM-based	CAS SW (0/1-type)	CAS SW (0/1/A/$\bar{A}$-type)
4-LUT	Logic (μm^2)	56.90	34.50	16.10
	Switch (μm^2)	0	5.76	5.76
5-LUT	Logic (μm^2)	116.10	71.30	34.50
	Switch (μm^2)	0	11.52	11.52

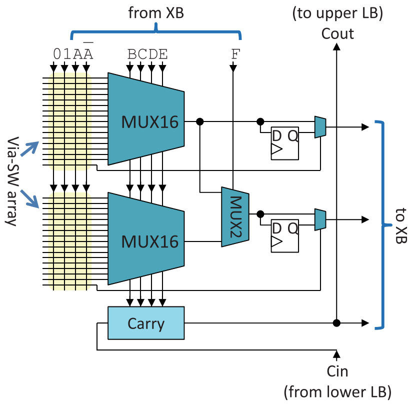


Fig. 15. Proposed LB.

proposed (say 0/1/A/$\bar{A}$-type LUT). These areas are summarized in Table II. Here, we assume that the area of a via-switch, an SRAM cell, and a k-MUX are 0.18, 1.40, and 2.30($k - 1$) μm^2, respectively. We can see that the footprint area of 0/1/A/$\bar{A}$-type LUT using via-switch is less than 1/3 of that of SRAM-based LUT.

For conducting a case study in Section V-C, we assume that the LB of the proposed architecture is similar to conventional SRAM-based FPGAs. The LB consists of a 6-LUT that can be divided to two 5-LUTs, optional output D-FFs, and a dedicated carry chain as depicted in Fig. 15, The logic area and switch area of this LB are estimated as 109.05 and 24.48 μm^2, respectively.

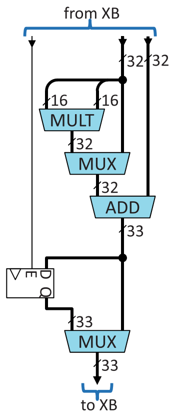


Fig. 16. Example of AB (IMA16).

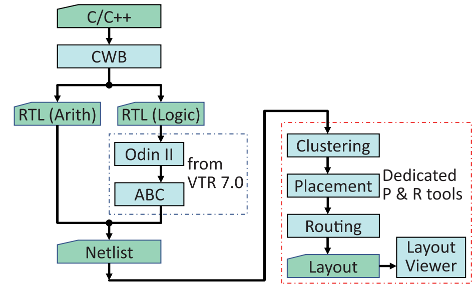


Fig. 17. Developed tool chain for our proposed architecture.

As for AB, we can adopt various kinds of arithmetic circuits including multipliers and MACs with various word sizes of input and output. MB can be single-port or dual-port SRAM macro with various word sizes and word counts. It is noteworthy that AB/MB requires very few number of switches and consumes only logic area. For the case study of Section V-C, IMA16 in Fig. 16 is used for AB. The logic area consumed by an IMA16 is 4255.52 μm^2.

C. Mapping Experiments

To demonstrate the effectiveness of bidirectional interconnect, we have developed a dedicated design flow as shown in Fig. 17. The design described in C++ is compiled to register-transfer level (RTL) using Cyber Work Bench [27], and the RTL is further compiled to technology-dependent netlist using Odin II and ABC included in VTR 7.0 [28]. The netlist is then placed and routed to the target array using a dedicated tool for our architecture. Before placement, a simple clustering tool is used to cluster an LUT and its output D-FF to form a LB. Note that clustering of multiple LBs for a tile is not necessary since each LB has direct access to XB in our architecture. For placement, simulated annealing-based program is used. Compared with VTR placer, range limiter is modified to get better solution for target MGRA [29]. For routing, we developed a dedicated program based on negotiation-based algorithm. Although the base algorithm is similar to that of the router in VTR 7.0, we could not use it because of the restriction to avoid sneak path problem as noted in IV-A, namely, while the multiple ON via-switches are allowed in a vertical signal line in a XB [Fig. 18(a)], multiple ON via-switches on a horizontal signal line in an XB are not allowed [Fig. 18(b)].

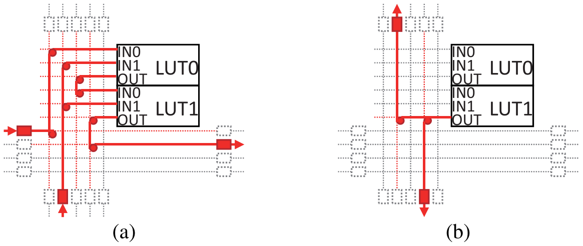


Fig. 18. Routing constraint of via-switch crossbar. (a) Multiple ON via-switches in a vertical line are allowed. (b) Multiple ON via-switches in a horizontal line are prohibited.

As for sample circuits, we used "fir_10tap_14bit," "fir_13tap_14bit," and "conv_5 × 5". The first two are finite impulse response (FIR) filter circuits that consist of multipliers, adders, and logics for saturation operation. "conv_5 × 5" is a module for applying convolution operation of 5 × 5 cofactor matrix to input image.

To compare the chip area of MGRAs with unidirectional and bidirectional interconnect structure, it is important to consider the N_{tr} demanded for implementing the same circuit. Let $N_{\mathrm{tr}}^{T}[i]$, $N_{\mathrm{tr}}^{B}[i]$, $N_{\mathrm{tr}}^{L}[i]$, and $N_{\mathrm{tr}}^{R}[i]$ be the numbers of occupied tracks of the ith XB in the array for the signals propagating toward top, bottom, left, and right directions, respectively. If the routing tracks consist of unidirectional interconnects only, and assuming that all XB have the same number of tracks (i.e., homogeneous array), the number of required vertical tracks is $\max_i(N_{\mathrm{tr}}^{T}[i]) + \max_i(N_{\mathrm{tr}}^{B}[i])$, and that of horizontal tracks is $\max_i(N_{\mathrm{tr}}^{L}[i]) + \max_i(N_{\mathrm{tr}}^{R}[i])$. Especially, if we assume that number of tracks for four directions are the same (i.e., symmetric array), $N_{\mathrm{tr}} = 2 \cdot \max(\max_i(N_{\mathrm{tr}}^{T}[i]), \max_i(N_{\mathrm{tr}}^{B}[i]), \max_i(N_{\mathrm{tr}}^{L}[i])$, and $\max_i(N_{\mathrm{tr}}^{R}[i]))$. Note that N_{tr} for unidirectional interconnect structure is dominated by the most demanded signal direction. On the other hand, if all the routing tracks are bidirectional (i.e., signal direction of each track is reconfigurable) and homogeneous array is assumed, the required vertical and horizontal tracks are $\max_i(N_{\mathrm{tr}}^{T}[i] + N_{\mathrm{tr}}^{B}[i])$ and $\max_i(N_{\mathrm{tr}}^{L}[i] + N_{\mathrm{tr}}^{R}[i])$, respectively. In case of the symmetric array, $N_{\mathrm{tr}} = \max(\max_i(N_{\mathrm{tr}}^{T}[i] + N_{\mathrm{tr}}^{B}[i]), \max_i(N_{\mathrm{tr}}^{L}[i] + N_{\mathrm{tr}}^{R}[i]))$. Thanks to the bidirectional interconnect structure, the routing demand inside a vertical channel (north-bound and south-bound tracks) and a horizontal channel (east-bound and west-bound tracks) are averaged, resulting smaller N_{tr} compared with unidirectional interconnect structure.

Fig. 19 shows the layout view of "fir_13tap_14bit" on the proposed MGRA of 5 × 5 array with bidirectional interconnect structure. Table III summarizes the chip area for the unit tile array that is capable of implementing the circuits. In this table, the mapping results of three circuits ("fir_10tap_14bit," "fir_13tap_14bit," and "conv_5 × 5") for two types of interconnect architectures (unidirectional and bidirectional) are shown. The columns N_{AB} and N_{LB} represent the required number of ABs and LBs, respectively, for mapping the circuits. The column N_{tr} represents the minimum number of tracks for successful routing. While the proposed architecture of bidirectional interconnects with $N_{\mathrm{tr}} = 96$ is capable of implementing "fir_13tap_14bit," $N_{\mathrm{tr}} = 112$ is needed in the case of unidirectional interconnects due to unbalanced routing demands. The column $N_{\mathrm{local\_IO}}$ represents the actual value obtained as follows. The XB of the left-hand side of Fig. 14(c) is connected to six inputs and three outputs of each of two LUTs as well as 49 (out of 65) inputs for the AB are connected. The XB of the right-hand side of Fig. 14(c) is connected to six inputs and three outputs of each of the other two LUTs as well as 16 inputs and 33 outputs for the AB are connected. As a result, both XBs have $N_{\mathrm{local\_IO}} = 67$ connections. The columns "BEoL area" and "FEoL area" show the area occupancy for the BEoL layers and FEoL layers, respectively. The column "Tile area" shows the physical dimension of a unit tile, which is given by $\max(\mathrm{Total(B)}, \mathrm{Total(F)})/0.8$, assuming that 20% of chip area is used for power/ground rails. The column "Array area" lists the chip size needed to implement the circuit, which is given by a product of "Tile area" and "Array size," where "Array size" is the minimum square that has at least N_{AB} ABs and N_{LB} LBs.

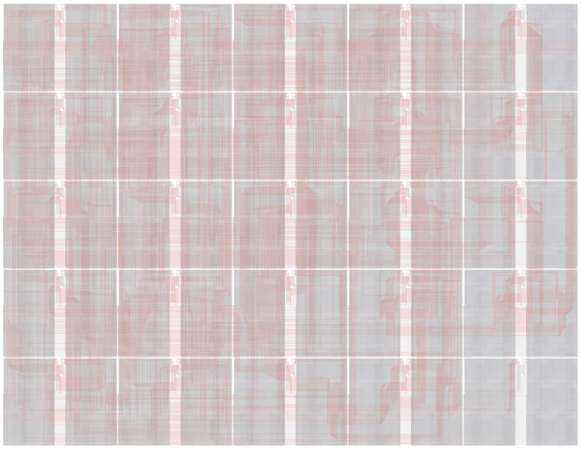


Fig. 19. Layout view of fir_13tap_14bit on the proposed MGRA of 5×5 array with bidirectional interconnect structure.

From the result of "fir_13tap_14bit," it is observed that the proposed MGRA with bidirectional interconnect achieves 21.7% reduction from 228 602 to 179 094 $\mu\mathrm{m}^2$ in array area compared to that with unidirectional interconnect. While the MGRA with unidirectional interconnect architecture demands larger BEoL layer area (7315$\mu\mathrm{m}^2$) than FEoL layer area (4692 $\mu\mathrm{m}^2$), the area demand of BEoL layers of the proposed unit tile (5731 $\mu\mathrm{m}^2$) is close to that of FEoL layers, resulting in better area efficiency. In the case of "conv_5 × 5," it is also observed that N_{tr} is decreased from 68 to 62. Unfortunately, this improvement does not contribute to array area reduction since BEoL layer area is smaller than FEoL.

VI. Interconnect Performance Evaluation

This section evaluates the interconnect delay and the energy for signal transmission. Section VI-A evaluates the performance improvement, thanks to the bidirectional signal transmission and selective repeater insertion, which are the features of the proposed architecture. In section VI-B, we compare the performance of the proposed and conventional architectures.

TABLE III
MAPPING RESULT OF CIRCUITS

Circuit			Architecture			BEoL area ($\mu\mathrm{m}^2$)			FEoL area ($\mu\mathrm{m}^2$)			Area ($\mu\mathrm{m}^2$)			
Name	N_{AB}	N_{LB}	Interconnect	N_{tr}	$N_{\mathrm{local\_IO}}$	XB	LB	Total(B)	LB	AB	Total(F)	Tile area	Array size	Array area	Ratio
fir_10tap_14bit	18	73	unidir	96	67	5,633	98	5,731	436	4,256	4,692	7,164	5 × 5	179,103	
			bidir	86	67	4,737	98	4,835	436	4,256	4,692	6,044	5 × 5	151,094	−15.6%
fir_13tap_14bit	24	87	unidir	112	67	7,217	98	7,315	436	4,256	4,692	9,144	5 × 5	228,602	
			bidir	96	67	5,633	98	5,731	436	4,256	4,692	7,164	5 × 5	179,094	−21.7%
conv_5x5	89	421	unidir	68	67	3,305	98	3,403	436	4,256	4,692	5,865	11 × 11	709,665	
			bidir	62	67	2,879	98	2,977	436	4,256	4,692	5,865	11 × 11	709,665	−0.0%

A. Performance Improvement, Thanks to Bidirectional Signal Transmission and Selective Repeater Insertion

First, we evaluate the dependence of the delay and energy on the crossbar size aiming to show that the smaller crossbar, thanks to bidirectional signal, can contribute to higher performance. We constructed circuit models of 86 × 153 crossbar ($516F \times 459F = 51.6\ \mu\mathrm{m} \times 45.9\ \mu\mathrm{m}$) and 96 × 163 crossbar ($576F \times 489F = 57.6\ \mu\mathrm{m} \times 48.9\ \mu\mathrm{m}$) using the equivalent circuit model of the 2V-1CAS via-switch in Fig. 3. Here, the 86 × 153 crossbar corresponds to the proposed MGRA with bidirectional interconnection, and the 96 × 163 crossbar is the MGRA with unidirectional interconnection, where both of the configurations are found in Table III. The circuit models include wire resistance and capacitance of the signal lines. Then, by connecting the crossbar circuit models with intercrossbar via-switches, we generated the transistor-level netlists of the tiled crossbars. We also connected the LUTs and repeaters to the netlist of the tiled crossbars. A 65-nm thin-BOX fully depleted silicon on insulator transistor [24], which is suitable for low-voltage operation, is assumed. Then, we performed the circuit simulation by HSPICE and evaluated the propagation delay from the LUT output of the source LB to the LUT input of the destination LB by changing the number of XBs between the source LB and the destination LB. Here, the signal propagation along the longer edge of the crossbar is assumed.

Fig. 20 shows the interconnect delay and the energy per signal transition. In this evaluation, no repeaters are inserted. We can see that the interconnect delay depends on the crossbar size and it decreases by 11% when the crossbar size is reduced from 96 × 163 to 86 × 153. The energy per signal transition also depends on the crossbar size and it decreases by 10%. Here, these reduction ratios are evaluated at the distance of seven XBs, since the average distance in the mapping results of fir_10tap_14bit is roughly seven XBs. The crossbar size reduction, thanks to bidirectional interconnection, is effective for interconnect delay and energy reduction.

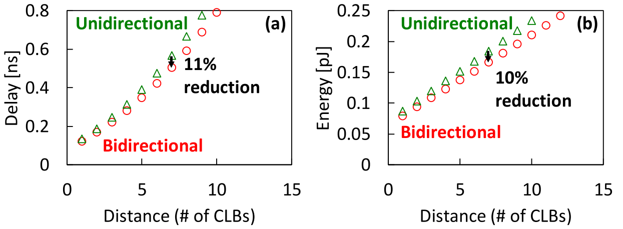


Fig. 20. (a) Interconnect delay. (b) Energy per signal transmission in the proposed interconnect structure. Two crossbar sizes of 86 × 153 and 96 × 163 are evaluated. Repeaters are not inserted. Supply voltage is 1.0 V.

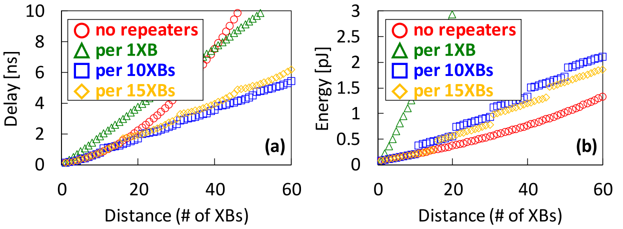


Fig. 21. (a) Interconnect delay. (b) Energy per signal transmission in the proposed interconnect structure. Crossbar size is 86 × 153. Supply voltage is 1.0 V.

Next, we evaluated the impact of repeater insertion. Fig. 21 shows the relations of interconnect delay and energy to the distance between the source LB and the destination LB. The repeater size is determined to balance the propagation delay and energy, and its size is 8×. By inserting repeaters, the delay becomes proportional to the distance, as we expected. We can also see that the frequent repeater insertion (per 1 XB in Fig. 21) leads to the significant increase in both delay and energy. Therefore, it should be avoided to insert repeaters frequently in via-switch-based FPGA. In the case of short-distance transmission, no repeater insertion is needed for minimizing the delay and energy. When the distance is more than 14 XBs, the repeater insertion per 10 XBs achieved the minimum delay. On the other hand, the insertion per 15 XBs reduces the energy with small delay increase. We can insert repeaters in the proposed interconnect structure according to the timing constraint.

In the mapping result of fir_10tap_14bit described in Section V-C, the most frequent distance from the source to the destination was six XBs and the ratio of the interconnections whose distance is longer than 14 XBs was only 2.3%. For 97.7% of interconnections, no repeaters are needed for delay minimization. For larger designs, longer interconnection will appear, but its frequency is expected to be still not high since such tendency is observed with Rent's rule (e.g., [31]). Therefore, the number of repeaters is expected to be small. In addition, by introducing long wires, which can be easily accommodated as shown in Fig. 13, we further reduce the number of repeater insertion. Thus, the flexible repeater insertion well fits the proposed interconnect structure.

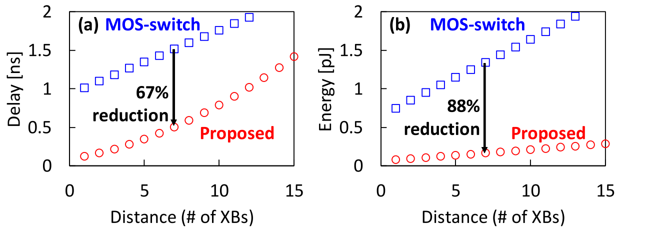


Fig. 22. (a) Delay and (b) energy comparison between the proposed and MOS-switch architectures. Repeaters are not inserted. Crossbar size is 86 × 153. Supply voltage is 1.0 V.

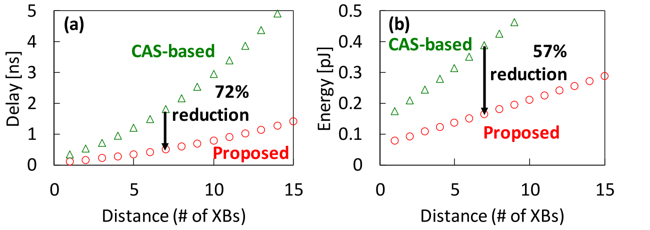


Fig. 23. (a) Delay and (b) energy comparison between the proposed and CAS-based architectures. Repeaters are not inserted. Crossbar size is 86×153. Supply voltage is 1.0 V.

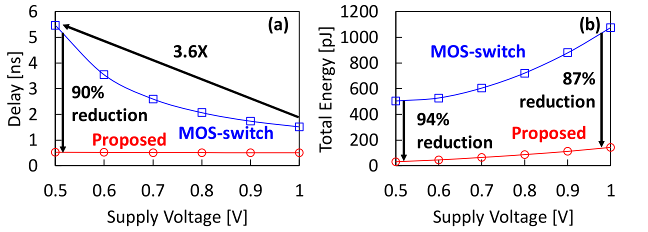


Fig. 24. (a) Interconnect delay. (b) Total energy in the proposed and MOS-switch architectures when supply voltage is varied. Repeaters are not inserted. Crossbar size is 86 × 153.

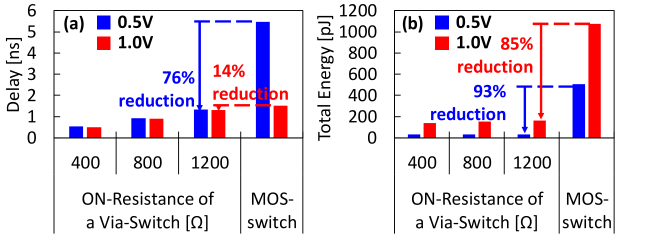


Fig. 25. (a) Interconnect delay. (b) Total energy in the proposed and MOS-switch architectures when the ON-resistance of via-switch is varied. Repeaters are not inserted. Crossbar size is 86×153.

B. Performance Comparison Between Proposed and Conventional Architectures

Here, we compare the interconnect delay and energy between the proposed architecture and two conventional architectures. The first conventional architecture corresponds to MOS-switch FPGA. The fully dense crossbar that has the same routing flexibility with Fig. 13 is implemented with complementary pass gates and SRAM cells, and the crossbars are connected by back-to-back tristate buffers with SRAM cells for enabling bidirectional signaling. Here, the minimum transistors are used for pass gates while 4× tristate buffers are used for intercrossbar connection to minimize the propagation delay. Referring an industrial 65-nm cell library, the crossbar size is estimated by the number of transistors, and the transistor-level netlist of the crossbar, which include wire resistance and capacitance, is generated. Here, the 86 × 153 crossbar size is 223.6 μm × 275.4 μm, and it is 26 times larger compared to the via-switch crossbar. This means the proposed architecture can achieve 26× higher crossbar density. The second conventional architecture corresponds to CAS-based FPGA [30], in which the crossbar is implemented with CAS switches with access transistors. We estimated the crossbar size from the information in [30], and the 86 × 153 crossbar size is 271.8 μm × 127.0 μm. Due to the access transistor, the crossbar size is 15× larger, which demonstrates the superiority of the via-switch.

First, we compare the performance by changing the signal transmission distance. Figs. 22 and 23 show the comparisons between the proposed and MOS-switch architectures and between the proposed and CAS-based architectures, respectively. We can see that the proposed architecture attains significant delay and energy reduction. The reduction ratios of delay and energy from the MOS-switch FPGA are 67% and 88%, respectively. Compared to the CAS-based FPGA, the slope of delay increase to distance is much smaller, which is due to 15× density improvement. The proposed architecture reduces interconnect delay and energy by 72% and 57%, respectively, compared to the CAS-based FPGA.

So far, the supply voltage is fixed at 1.0 V. Next, we sweep the supply voltage in the range from 0.5 to 1.0 V to evaluate the performance at low-voltage operation. Fig. 24 shows the evaluation result. Here, the signal transmission with seven XBs distance is assumed, where please remind that seven XBs is the average distance in the mapping result of fir_10tap_14bit. The total energy, which is depicted in Fig. 24(b), is the sum of all the interconnect energies in fir_10tap_14bit assuming that a single pulse propagates through each interconnect. Fig. 24(a) shows that the interconnect delay reduction ratio from the MOS-switch FPGA becomes higher as the supply voltage becomes lower. At 0.5 V, the ratio of interconnect delay reduction reaches 90% because the ON-resistance of the via-switch is independent of the supply voltage while that of conventional transistor switch depends on the supply voltage. The interconnect delay increasing ratio from 1.0 to 0.5 V of the proposed architecture is only 1.1×, whereas that of the MOS-switch architecture is 3.6×. We can also see that the energy reduction ratio become higher with voltage decrease [Fig. 24(b)]. Focusing on 0.5-V operation, the energy reduction is 94%. These evaluation results show that the proposed architecture can achieve high performance even at low-supply voltage.

Next, we evaluate the impact of ON-resistance of via-switch by increasing it from 400 to 1200 Ω (from 200 to 600 Ω for each atom switch) since, as mentioned Section II-A, the varistor in via-switch is under development for increasing programming current. Evaluation results are shown in Fig. 25. The signal transmission distance is seven XBs. We can see from Fig. 25(a) that the delay increases in proportion to the ON-resistance. However, even when the ON-resistance rises to 1200 Ω, the delay reduction from the MOS-switch architecture is still achievable, especially large reduction of 76% is attained

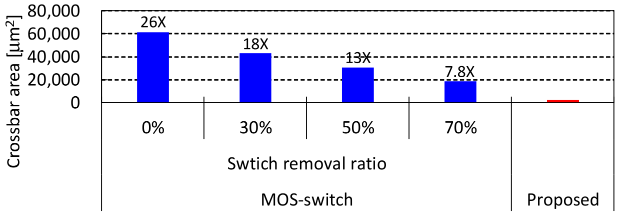


Fig. 26. Crossbar area variation in terms of switch removal.

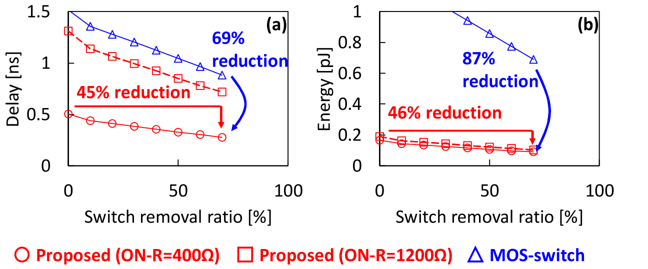


Fig. 27. (a) Interconnect delay. (b) Energy in the proposed and MOS-switch architectures when the thinning ratio of intersection switches is varied. Repeaters are not inserted. Crossbar size is 86 × 153.

at 0.5-V operation. On the other hand, the total energy has almost no changes even when the ON-resistance rises, and it is reduced nearly by one order of magnitude from the MOS-switch one at both 1.0- and 0.5-V operations.

All the above evaluations suppose dense crossbars where via-switches are placed at all the intersections. On the other hand, sparser (or depopulated) switch box, which have fewer intersection switches, are often used in conventional FPGAs. The sparsened switch box may degrade the routing flexibility but can reduce the area [32]. In the via-switch FPGA, the sparse crossbar decreases the number of via-switches connected to each signal wire and consequently reduces the load capacitance of each signal wire, whereas the crossbar area is unchanged. In case of conventional FPGA, the crossbar area is also reduced since the number of transistors is reduced.

We evaluate the impact of crossbar sparseness on interconnect performance. In the evaluation, we sweep the ratio of switch removal in the range from 0% (i.e., fully dense crossbar) to 70%. If the switch removal ratio is 70%, each horizontal wire segment has $0.3N_{tr}$ switches, and each input–output ports of LBs and ABs has $0.3N_{tr}$ switches. The latter can be expressed as $F_{c,\text{in}} = F_{c,\text{out}} = 0.3$ using the terminology of island-style FPGA. Fig. 26 shows the crossbar area. Even when the removal ratio reaches 70%, the via-switch crossbar is still 7.8× smaller than MOS-switch one. Fig. 27 shows the interconnection delay and energy, where the crossbar size is 86 × 153, the supply voltage is 1.0 V, and the signal transmission distance is seven XBs. As shown in Fig. 27, both delay and energy decrease according to via-switch removal. When the removal ratio is 70% and ON-resistance of via-switch is 400 Ω, the reduction ratios of delay and energy from the full matrix crossbar are 45% and 46%, respectively. We can also see that the delay and energy of via-switch FPGA remain much smaller than those of MOS-switch FPGA even with the switch removal, and the reduction ratios of delay and energy are 69% and 87%, respectively.

The above results indicate that the proposed architecture achieves significant performance improvement compared with the conventional architectures. In particular, it is more significant at the low-voltage operation, and the interconnect delay and energy are reduced by one order of magnitude or more at 0.5-V operation. This improvement can contribute to filling the gap between FPGA and ASIC.

VII. Conclusion

This paper proposed a reconfigurable architecture that could exploit the advantages of via-switch in terms of small footprint, BEoL only integration and low ON-resistance. The overlay interconnect structure adopts bidirectional signaling for minimizing the crossbar size and improving the performance, and it enables flexible repeater insertion. The underlay logic structure is designed so as to fully exploit the FEoL area under the crossbar, and mixed-grained logics consisting of LUT and arithmetic/memory unit are adopted for improving the compatibility to various applications. An example of application mapping result shows that the proposed mixed grained logic structure with bidirectional interconnect achieved 21.7% array area reduction compared to that with unidirectional interconnect. Evaluation results on crossbar performance show that the proposed interconnect structure can achieve up to 26× higher crossbar integration density and reduce interconnect delay and energy by 90% and 94% at 0.5-V operation, compared to conventional transistor-based crossbars. Future works include the consideration of via-switch ON-resistance variability in performance estimation and design tools.

References

[1] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 203–215, Feb. 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 Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 216–229, Feb. 2007.

[3] R. Rajaei, “Radiation-hardened design of nonvolatile MRAM-based FPGA,” *IEEE Trans. Magn.*, vol. 52, no. 10, Oct. 2016, Art. no. 3402010.

[4] W. Zhao, E. Belhaire, V. Javerliac, C. Chappert, and B. Dieny, “Evaluation of a non-volatile FPGA based on MRAM technology,” in *Proc. ICICDT*, 2006, pp. 1–4.

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

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

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

[8] X. Tang, P.-E. Gaillardon, and G. de Micheli, “A high-performance low-power near-Vt RRAM-based FPGA,” in *Proc. ICFPT*, 2014, pp. 207–215.

[9] Y. Y. Liauw, Z. Zhang, W. Kim, A. El Gamal, and S. S. Wong, “Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory,” in *ISSCC Dig. Tech. Papers*, Feb. 2012, pp. 406–408.

[10] 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*, vol. 58, no. 12, pp. 4398–4405, Dec. 2011.

[11] M. Miyamura *et al.*, “Programmable cell array using rewritable solid-electrolyte switch integrated in 90 nm CMOS,” in *ISSCC Dig. Tech. Papers*, Feb. 2011, pp. 228–229.

[12] M. Tada *et al.*, “Improved off-state reliability of nonvolatile resistive switch with low programming voltage,” *IEEE Trans. Electron Devices*, vol. 59, no. 9, pp. 2357–2362, Sep. 2012.

[13] M. Miyamura *et al.*, "0.5-V highly power-efficient programmable logic using nonvolatile configuration switch in BEOL," in *Proc. FPGA*, 2015, pp. 236–239.

[14] M. Miyamura *et al.*, "Low-power programmable-logic cell arrays using nonvolatile complementary atom switch," in *Proc. ISQED*, 2014, pp. 330–334.

[15] X. Bai *et al.*, "A low-power Cu atom switch programmable logic fabricated in a 40 nm-node CMOS technology," in *Symp. VLSI Technol. Dig.*, Jun. 2017, pp. T28–T29.

[16] 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," in *IEDM Tech. Dig.*, Dec. 2015, pp. 32–35.

[17] N. Banno *et al.*, "50×20 crossbar switch block (CSB) with two-varistors (a-Si/SiN/a-Si) selected complementary atom switch for a highly-dense reconfigurable logic," in *IEDM Tech. Dig.*, Dec. 2016, pp. 16.4.1–16.4.4.

[18] J. Hotate *et al.*, "A highly-dense mixed grained reconfigurable architecture with overlay crossbar interconnect using via-switch," in *Proc. FPL*, 2016, pp. 1–4.

[19] M. Tada, T. Sakamoto, N. Banno, M. Aono, H. Hada, and N. Kasai, "Nonvolatile crossbar switch using $\mathrm{TiO}_x/\mathrm{TaSiO}_y$ Solid-electrolyte," *IEEE Trans. Electron Devices*, vol. 57, no. 8, pp. 1987–1995, Aug. 2010.

[20] M. Tada *et al.*, "Improved ON-state reliability of atom switch using alloy electrodes," *IEEE Trans. Electron Devices*, vol. 60, no. 10, pp. 3534–3540, Oct. 2013.

[21] K. Okamoto *et al.*, "Bidirectional TaO-diode-selected, complementary atom switch (DCAS) for area-efficient, nonvolatile crossbar switch block," in *Symp. VLSI Technol. Dig.*, 2013, pp. T242–T243.

[22] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in FPGA interconnect," in *Proc. IEEE Int. Conf. Field-Program. Technol. (FPT)*, Dec. 2004, pp. 41–48.

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

[24] H. Makiyama *et al.*, "Ultralow-voltage operation of Silicon-on-Thin-BOX (SOTB) 2Mbit SRAM down to 0.37 V utilizing adaptive back bias," in *Symp. VLSI Circuits Dig.*, Jun. 2013, pp. T212–T213.

[25] D. Lewis *et al.*, "The StratixΠ routing and logic architecture," in *Proc. FPGA*, 2003, pp. 12–20.

[26] T. Higashi and H. Ochi, "Area-efficient LUT-like programmable logic using atom switch and its mapping algorithm," in *Proc. ISCIT*, 2015, pp. 201–204.

[27] K. Wakabayashi and T. Okamoto, "C-based SoC design flow and EDA tools: An ASIC and system vendor perspective," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 12, pp. 1507–1522, Dec. 2000.

[28] J. Luu *et al.*, "VTR 7.0: Next generation architecture and CAD system for FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 6:1–6:30, Jul. 2014.

[29] 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.*, vol. E99-A, no. 12, pp. 2328–2334, 2016.

[30] M. Miyamura *et al.*, "First demonstration of logic mapping on nonvolatile programmable cell using complementary atom switch," in *IEDM Tech. Dig.*, Dec. 2012, pp. 10.6.1–10.6.4.

[31] D. Stroobandt, *A Priori Wire Length Estimates for Digital Design*. New York, NY, USA: Springer, 2001.

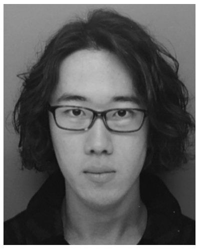
[32] G. Lemieux and D. Lewis, *Design of Interconnection Networks for Programmable Logic*. Norwell, MA, USA: Kluwer, 2004.

Hiroyuki Ochi (M'04) received the B.E., M.E., and Ph.D. degrees in engineering from Kyoto University, Kyoto, Japan, in 1989, 1991, and 1994, respectively.

From 1994 to 2004, he was an Associate Professor at Hiroshima City University, Hiroshima, Japan. From 2004 to 2013, he was an Associate Professor at Kyoto University. Since 2013, he has been a Professor at the College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Japan. His current research interests include reconfigurable architecture and low-power VLSI design.

Dr. Ochi is a member of the Association for Computing Machinery (ACM), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Information Processing Society of Japan (IPSJ).

Kosei Yamaguchi received the B.E. degree from Ritsumeikan University, Kusatsu, Japan, in 2017, where he is currently working toward the M.S. degree at the Graduate School of Information Science and Engineering.



Tetsuaki Fujimoto received the B.E. degree from Ritsumeikan University, Kusatsu, Japan, in 2017, where he is currently working toward the M.S. degree at the Graduate School of Information Science and Engineering.

Junshi Hotate received the B.E. degree from Ritsumeikan University, Kusatsu, Japan, in 2016, where he is currently working toward the M.S. degree at the Graduate School of Information Science and Engineering.

Takashi Kishimoto received the B.E. degree from Ritsumeikan University, Kusatsu, Japan, in 2015 and the M.E. degree from the Graduate School of Information Science and Engineering, Ritsumeikan University, in 2017.

He is currently with Honda Automobile R&D Center, Tochigi, Japan.

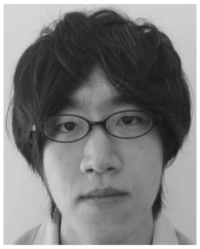
Toshiki Higashi received the B.E. degree from Ritsumeikan University, Kusatsu, Japan, in 2015 and the M.E. degree from the Graduate School of Information Science and Engineering, Ritsumeikan University, in 2017.

He is currently with MegaChips Corporation, Makuhari, Japan.

Takashi Imagawa (S'08–M'15) received the B.E. degree in electrical and electronic engineering, and the M.S. and Ph.D. degrees in communications and computer engineering from Kyoto University, Kyoto, Japan, in 2008, 2010, and 2015, respectively.

From 2010 to 2013, he was a JSPS Research Fellow. From 2015 to 2017, he was a Postdoctoral Research Fellow at the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. In 2017, he joined the College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Japan, as an Assistant Professor. His current research interests include VLSI design methodology and reconfigurable architectures.

Dr. Imagawa is a member of the Information Processing Society of Japan (IPSJ) and the Institute of Electronics, Information and Communication Engineers (IEICE).



Ryutaro Doi (S'14) received the B.E. and M.E. degrees in information systems engineering from Osaka University, Suita, Japan, in 2015 and 2017, respectively. He is currently working toward the Ph.D. degree at the Department of Information Systems Engineering, Osaka University.

His current research interest includes reconfigurable architecture and its test.

Munehiro Tada (M'07–SM'10) received the M.S. (honors) and Ph.D. degrees from Keio University, Tokyo, Japan.

In 1999, he joined NEC Corporation, Tsukuba, Japan. He was a Visiting Scholar at Stanford University, Stanford, CA, USA, in 2008. From 2011 to 2014, he was with the Low-Power Electronics Association and Project, Tsukuba. He is currently a Principal Researcher with NEC Corporation. His current research interests include ultralow power device, circuit, system, and applications of emerging technologies.

Tadahiko Sugibayashi received the B.S. and M.S. degrees in material science from Osaka University, Osaka, Japan, in 1984 and 1986, respectively.

In 1986, he joined NEC Corporation, Tsukuba, Japan, where he was involved in memory LSI design. He is currently involved in the development of ultralow power circuits utilizing next-generation nonvolatile devices. He is a Senior Manager of Green Platforms Laboratories, NEC Corporation.

Mr. Sugibayashi is a member of the Institute of Electronics, Information and Communication Engineers, Japan.

Wataru Takahashi received the B.E. and M.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 1996 and 1998, respectively.

In 1998, he joined NEC Corporation, Kawasaki, Japan, where he has been involved in the research and development high-level synthesis for VLSI.

Kazutoshi Wakabayashi (M'93) received the B.E., M.E., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan.

From 1993 to 1994, he was a Visiting Researcher at Stanford University, Stanford, CA, USA. In 1986, he joined NEC Corporation, Kawasaki, Japan. He is currently a Senior Principal Researcher at the Central Research Laboratory and a Senior Expert at the Embedded System Solution Division, NEC Corporation. He has been involved in the research and development of VLSI CAD systems.

Dr. Wakabayashi served on executive committee or organizing committee of some international conference including the ASP-DAC'09 General Chair, the CODES+ISSS'09 Co-Technical Program Chair, and the Editor-in-Chief of ICCAD and DAC. He has also served on the program committees for several conferences including DAC, ICCAD, DATE, ASP-DAC, ISSS, SASIMI, and so on.

Hidetoshi Onodera (M'87–SM'12–F'18) received the B.E., M.E., and Dr. Eng. degrees in electronic engineering from Kyoto University, Kyoto, Japan.

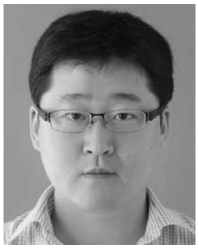
In 1983, he joined the Department of Electronics, Kyoto University, where he is currently a Professor at the Department of Communications and Computer Engineering, Graduate School of Informatics. His current research interests include design technologies for digital, analog, and RF LSIs, with a particular emphasis on low-power design, design for manufacturability, and design for dependability.

Dr. Onodera served as a Program Chair and a General Chair of ICCAD and ASP-DAC. He was the Chairman of the Information Processing Society of Japan SIG-System LSI Design Methodology (IPSJ SIG-SLDM), the Institute of Electronics, Information and Communication Engineers (IEICE) Technical Group on VLSI Design Technologies, the IEEE SSCS Kansai Chapter, the IEEE CASS Kansai Chapter, and IEEE Kansai Section. He served as the Editor-in-Chief of the *IEICE Transactions on Electronics* and *IPSJ Transactions on SLDM*.

Yukio Mitsuyama (S'98–M'03) received the B.E., M.E., and Ph.D. degrees in information systems engineering from Osaka University, Osaka, Japan, in 1998, 2000, and 2010, respectively.

He was an Assistant Professor with the Graduate School of Engineering, Osaka University. He is currently an Associate Professor with the School of System Engineering, Kochi University of Technology, Kami, Japan. His current research interests include reconfigurable architecture and its VLSI design.

Dr. Mitsuyama is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and the Information Processing Society of Japan (IPSJ).



Jaehoon Yu (S'11–M'13) received the B.E. degree in electrical and electronic engineering and the M.S. degree in communications and computer engineering from Kyoto University, Kyoto, Japan, in 2005 and 2007, respectively, and the Ph.D. degree in information systems engineering from Osaka University, Osaka, Japan, in 2013.

He is currently an Assistant Professor at the Department of Information Systems Engineering, Osaka University.

Dr. Yu is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).

Masanori Hashimoto (S'00–A'01–M'03–SM'11) received the B.E., M.E., and Ph.D. degrees in communications and computer engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively.

He is currently a Professor at the Department of Information Systems Engineering, Osaka University. His current research interests include design for manufacturability and reliability, timing and power integrity analysis, reconfigurable computing, soft error characterization, and low-power circuit design.

Dr. Hashimoto was on the technical program committees of international conferences including DAC, ICCAD, ITC, Symposium on VLSI Circuits, ASP-DAC, and DATE. He serves/served as an Associate Editor for IEEE TVLSI, TCAS-I, and ACM TODAES.