

# A Highly-dense Mixed Grained Reconfigurable Architecture with Overlay Crossbar Interconnect using Via-switch

Junshi Hotate<sup>1,6</sup> Takashi Kishimoto<sup>1,6</sup> Toshiki Higashi<sup>1,6</sup> Hiroyuki Ochi<sup>1,6</sup> Ryutaro Doi<sup>2,6</sup>  
Munehiro Tada<sup>3,6</sup> Tadahiko Sugibayashi<sup>3,6</sup> Kazutoshi Wakabayashi<sup>3,6</sup>  
Hidetoshi Onodera<sup>4,6</sup> Yukio Mitsuyama<sup>5,6</sup> Masanori Hashimoto<sup>2,6</sup>

<sup>1</sup>Ritsumeikan University, <sup>2</sup>Osaka University, <sup>3</sup>NEC Corporation  
<sup>4</sup>Kyoto University, <sup>5</sup>Kochi University of Technology, <sup>6</sup>JST, CREST  
E-mail: nanocrest@gmail.com

## ABSTRACT

This paper proposes a highly-dense reconfigurable architecture that introduces via-switch device, which is a kind of resistive RAM and is used in crossbar switches. Since via-switch is implemented in BEOL layers only, the FEOL layer under the crossbar can be fully exploited for highly-dense logic blocks. The proposed architecture uses the FEOL layer for fine-grained look-up tables and coarse-grained arithmetic/memory units for better performance and highly wide applications. In a case study of application mapping, the proposed architecture reduces array area by 76% thanks to mixed grained logic structure and overlay 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 26X and the delay and energy in the interconnection are reduced by 90% and 93% at 0.5V operation.

## 1. INTRODUCTION

Reconfigurable devices, a representative of which is FPGA, are gaining their popularity as a means of integrated system implementation since NRE (Non-recurring Expense) cost of application specific integrated circuits (ASIC) 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[1].

The lower performance of FPGA originates from the low area efficiency and poor interconnect performance. Reference [2] reports 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 hence the interconnect delay becomes longer. In addition, the interconnection in FPGA consists of a number of programmable switches and buffers. Consequently, 80% of the circuit delay is occupied by interconnect delay[2].

To overcome the interconnect delay problem in FPGA, [3–7] introduced resistive RAM as not only configuration memory but also programmable switch for signal transmission. Besides, atom switch (a.k.a. nanobridge), which is a kind of RRAM and is integrated on BEOL (back end of line) layers, has been developed as a non-volatile programmable switch tailored for FPGA[8] and the first FPGA implementation with atom switch is fabricated and presented in [9]. The FPGA implementations with complementary atom switch (CAS) and their silicon results are reported in [11, 12].

However, all the above mentioned FPGA implementations with RRAM, atom switch and CAS require one or two access transistors for every programmable switch. If the access transistors can be eliminated, we can significantly improve the switch density and improve the interconnect delay and reduce energy for signal transmission. Motivated by this, via-switch has been developed[13], which consists of CAS and two varistors. The two varistors enable device selection for programming without the access transistors.

This paper proposes a highly-dense reconfigurable architecture that uses via-switch for crossbar implementation. We devise an interconnect structure that can exploit the small footprint and low re-

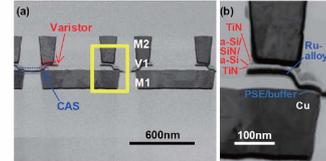


Figure 1: Cross-sectional TEM images of via-switch integrated in 65nm-node Cu-BEOL [13].

sistivity of via-switch. In addition, we devise a logic structure that consists of fine-grained look-up tables and coarse-grained arithmetic/memory units aiming to fully exploit the transistor area under the overlay crossbar for highly-dense logic blocks.

The remainder of this paper is organized as follows. Section 2 reviews via-switch. Section 3 presents the concept of the proposed reconfigurable architecture, followed by the proposed interconnect structure and logic structure in Section 4 and 5, respectively. Results on interconnect performance evaluation are shown in Section 6. Concluding remarks are given in Section 7.

## 2. 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 fanouts[13]. Two control lines connected to the varistors realize accurate one-by-one programming of each cross-point without select transistors. The via-switch can be integrated with a small footprint of  $18F^2$ . The following explains the device in detail.

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  $200\Omega$ [14], which is suitable for signal line switch. Turn-on is achieved within less than 2ns.

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  $85^\circ\text{C}$  are applied. The ON state is also reliable for more than 3000 hours at  $150^\circ\text{C}$ [10]. The cycling endurance is up to 10k cycles[15].

The varistor is introduced to provide two functionalities; (1) program line isolation during normal operation, and (2) program current supply during programming operation. The cross-section of via-switch fabricated in 65-nm node[13] 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.

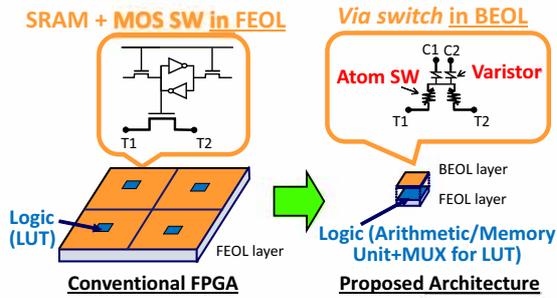


Figure 2: Concept of proposed reconfigurable architecture.

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 $\Omega$  in ON state and 200M $\Omega$  in OFF state. The varistor capacitance is also 0.14 fF. The small parasitic capacitance in addition to low on-resistance is the advantage of via-switch.

### 3. OVERVIEW OF PROPOSED ARCHITECTURE

With the via-switch explained in the previous section, we can implement a crossbar on BEOL layers without transistors, where the crossbar provides programmable interconnection and it is the most important component that determines the performance and integration density of the reconfigurable device. 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 5.

Using the crossbar and LUT memory with via-switch, the FEOL (front end of line) layers under the overlay crossbar and LUT memory can be fully used for logic implementation, as illustrated in Fig. 2. When we integrate the via-switch between M6 and M7 layers (as depicted in Fig. 3 above left), the metal layers of M1 to M4 can be used for the logic implementation.

The most significant advantage of the dense implementation is the shorter interconnection between the logic blocks. 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, the on-resistance of via-switch can be reduced to 400 $\Omega$ , which is 10X lower than the resistance of the smallest transistor. Thanks to these, the interconnect delay can be significantly reduced. To maximize the delay reduction effect, we devise a bidirectional interconnect structure with selective repeater insertion, which will be shown in the next section.

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. 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 whose basic elements are LUT and arithmetic/memory unit. This logical architecture design is discussed in Section 5.

### 4. INTERCONNECT STRUCTURE

Thanks to the  $18F^2$  ( $= 6F \times 3F$ ) via-switch which can be implemented on BEOL layers only, the crossbar can be implemented compactly. For example, a  $100 \times 100$  crossbar can be implemented in  $60\mu\text{m} \times 30\mu\text{m}$  in 65-nm node ( $F = 100\text{nm}$ ). Therefore, the interconnect resistance per crossbar is not high, and hence a repeater is not necessary for every crossbar. The via-switch is naturally capable of bidirectional signal transmission. Furthermore, 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

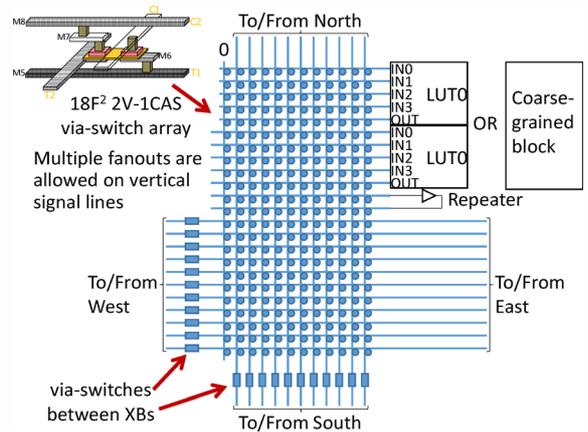


Figure 3: Proposed crossbar structure.

shown in Fig. 3. The crossbars are placed in a two-dimensional array. This crossbar includes a switch block, which is the bottom half of the crossbar, for the vertical and horizontal lines that are connected to the adjacent crossbars. 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 just an example. 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.

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. For avoiding a quadratic delay increase, we need to insert repeaters. This repeater insertion can be achieved by using the repeater below the LUT in Fig. 3. It should be noted that fewer and flexible repeater insertion makes it possible to take advantage of bidirectional signaling. In conventional FPGA, due to the larger number of buffers/repeaters, unidirectional signaling is widely adopted [16].

## 5. LOGIC STRUCTURE

### 5.1 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 two-dimensional array of the "unit tile" (Fig. 4(a)). The unit tile consists of four crossbar blocks (XBs), eight fine-grained logic blocks (LBs) and a coarse-grained arithmetic block (AB) or memory block (MB). Coarse-grained blocks (e.g., AB or MB) occupy larger area and pincounts than fine-grained components, and hence an AB or MB is connected to multiple XBs.

As mentioned in Section 3, the transistor area under the crossbar should be fully occupied by logic blocks for maximizing the area efficiency. To explore suitable logic blocks, the following discusses the logic area and switch area that are occupied by LB, XB and AB/MB. For this purpose, we define some parameters (Fig. 4(b)). 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 are the same.  $N_{local\_in}$  ( $N_{local\_out}$ ) is the total number of

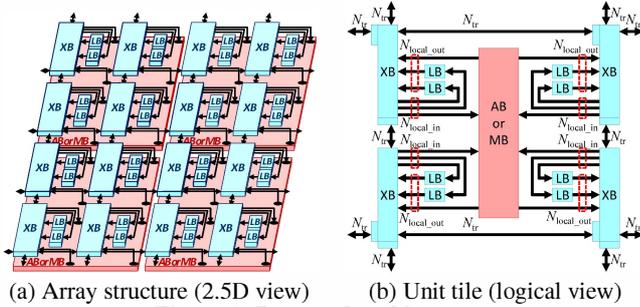


Figure 4: Proposed architecture.

Table 1: Comparison of LUT area.

		SRAM-based	CAS SW (0/1-type)	CAS SW (0/1/A/A-type)
4-LUT	Logic ( $F^2$ )	5,690	3,450	1,610
	Switch ( $F^2$ )	0	576	576

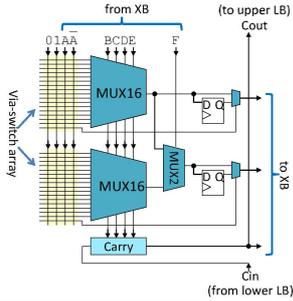


Figure 5: Proposed LB.

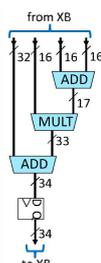


Figure 6: An example of AB (IAMA16).

local interconnects between an XB and the inputs (outputs) of the LBs and the AB/MB.

## 5.2 XB, LB and AB/MB

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

Next, we calculate the area of a LUT, which is the basic component of the LB. Conventional SRAM-based 4-input LUT consists of 16 SRAM cells and a 16-MUX. In [12], a 4-LUT is implemented using 32 CASs and 16-MUX (say 0/1-type LUT), and in [17], an improved 4-LUT architecture using 32 CASs and 8-MUX are proposed (say 0/1/A/A-type LUT). These areas are summarized in Table 1. Here, we assume that the area of a via-switch, an SRAM cell, and a  $k$ -MUX are  $18F^2$ ,  $140F^2$ , and  $230(k-1)F^2$ , respectively. We can see that the footprint area of 0/1/A/A-type LUT using via-switch is less than 1/3 of that of SRAM-based LUT.

For conducting a case study in Section 5.3, 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 FFs, and a dedicated carry chain as depicted in Fig. 5. The logic area and switch area of this LB are estimated as  $10,905F^2$  and  $2,448F^2$ , respectively.

As for AB, we can adopt various kind of arithmetic circuits including multipliers and multiply-accumulators (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 5.3, IAMA16 in Fig. 6 is used for AB. The logic area consumed by an IAMA16 is  $338,300F^2$ .

## 5.3 Mapping Experiments

We have developed a dedicated design flow for the proposed architecture, and implemented a design “CConv”. “CConv” is a front-end circuit for image sensor including RGB-YUV conver-

Table 2: Required logic resource for implementing CConv.

Target	AB	LB	Unit tile array size
Mixed-grained	14	76	$4 \times 4$ (16 ABs and 128 LBs)
Fine-grained only	—	512	$8 \times 8$ (512 LBs)

sion. The design described in C is compiled to RTL using Cyber Work Bench[18], and the RTL is further compiled to netlist. The required logic resource is shown in Table 2. In this table, the row “mixed-grained” shows the technology mapping result for the proposed mixed-grained architecture, while the row “fine-grained only” shows the result using only fine-grained resources for comparison. Since each unit tile of the proposed architecture has one AB and eight LBs,  $4 \times 4$  unit tile array is sufficient to implement the mixed-grained netlist, while  $8 \times 8$  unit tile array is needed for mapping the fine-grained netlist of 512 LBs, assuming that each unit tile has eight LBs. The netlist is then placed and routed to the target array.

Let  $N_{tr}^T[i]$ ,  $N_{tr}^B[i]$ ,  $N_{tr}^L[i]$ , and  $N_{tr}^R[i]$  be the numbers of occupied tracks of the  $i$ -th 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  $\max_i(N_{tr}^T[i]) + \max_i(N_{tr}^B[i])$ , and that of horizontal tracks is  $\max_i(N_{tr}^L[i]) + \max_i(N_{tr}^R[i])$ . Especially, if we assume that number of tracks for four directions are the same (i.e., symmetric array),  $N_{tr} = 2 \cdot \max(\max_i(N_{tr}^T[i]), \max_i(N_{tr}^B[i]), \max_i(N_{tr}^L[i]), \max_i(N_{tr}^R[i]))$ . 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  $\max_i(N_{tr}^T[i] + N_{tr}^B[i])$  and  $\max_i(N_{tr}^L[i] + N_{tr}^R[i])$ , respectively. In case of the symmetric array,  $N_{tr} = \max(\max_i(N_{tr}^T[i] + N_{tr}^B[i]), \max_i(N_{tr}^L[i] + N_{tr}^R[i]))$ .

From the routed layout of “CConv” for the mixed-grained array, we found that there is a channel that propagates 44 signals and all of them go to the same direction. This means that  $N_{tr} = 44 \times 2 = 88$  tracks are needed if the tracks are unidirectional, while only  $N_{tr} = 44$  tracks are sufficient if they are bidirectional. From the routed layout of “CConv” for the fine-grained array,  $N_{tr} = 34 \times 2 = 68$  tracks are needed for unidirectional interconnect architecture, while  $N_{tr} = 36$  tracks are needed for bidirectional interconnect architecture. As observed above, the routing demand of opposite directions are averaged using the bidirectional tracks, resulting in considerable reduction of  $N_{tr}$ .

Table 3 summarizes the chip area for the unit tile array that is capable of implementing “CConv”. In this table, the mapping results for two types of logic architectures (FGRA and MGRA) and two types of interconnect architectures (unidirectional and bidirectional) are shown. MGRA consists of the proposed unit tile (one AB, eight LBs and four XBs per unit tile as shown in Fig. 4(b)), while FGRA consists of the unit tile without AB or MB (eight LBs and four XBs per unit tile). The column “BEOL area” (“FEOL area”) shows the area occupancy for the BEOL (FEOL) layers. The column “Tile area” shows the physical dimension of a unit tile, which is given by  $\max(\text{Total}(B), \text{Total}(F))/0.8$ , assuming that 20% of chip area is used for power/ground rails. The column “Array area” lists the chip size needed to implement “CConv” circuit, which is given by a product of “Tile area” and “Array size”, where “Array size” is given by Table 2.

From this table it is observed that the proposed MGRA with bidirectional interconnect achieves 33% reduction from 127,642 to 85,108  $\mu\text{m}^2$  in Array area compared to the FGRA with bidirectional interconnect. This is due to the dramatic reduction of Array size using ABs, although MGRA’s Tile area is twice as large as FGRA’s. Table 3 also shows that the proposed MGRA with bidirectional interconnect achieves 51% reduction from 174,989 to 85,108  $\mu\text{m}^2$  in Array area compared to MGRA with unidirectional interconnect. While the MGRA with unidirectional interconnect

Table 3: Mapping result of “CConv”.

Architecture		Parameters			BEOL area ( $F^2$ )			FEOL area ( $F^2$ )			Area ( $\mu\text{m}^2$ )		
Logic	Interconnect	$N_{\text{tr}}$	$N_{\text{local in}}$	$N_{\text{local out}}$	XB	LB	Total(B)	LB	AB	Total(F)	Tile area	Array size	Array area
FGRA	unidir.	68	12	6	421k	20k	441k	87k	0	87k	5,508	$8 \times 8$	352,512
FGRA	bidir.	36	12	6	140k	20k	160k	87k	0	87k	1,994	$8 \times 8$	127,642
MGRA	unidir.	88	32	15	855k	20k	875k	87k	338k	426k	10,937	$4 \times 4$	174,989
MGRA	bidir.	44	32	15	288k	20k	308k	87k	338k	426k	5,319	$4 \times 4$	85,108

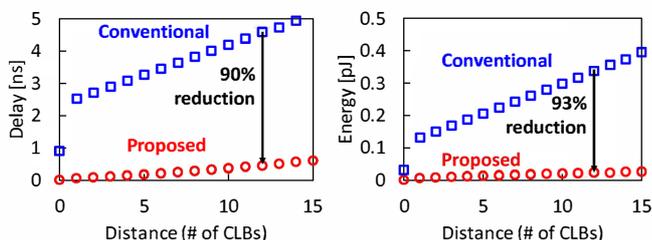


Figure 7: Comparison between proposed and conventional architecture. Repeaters are inserted. Crossbar size is  $91 \times 44$ . Supply voltage is 0.5V.

architecture demands much larger BEOL layer area ( $875kF^2$ ) than FEOL layer area ( $426kF^2$ ), the area demand of BEOL layers of the proposed unit tile ( $308kF^2$ ) is close to that of FEOL layers, resulting in better area efficiency. In total, compared to FGRA with unidirectional interconnect, which is the straightforward implementation referring to current SRAM-based FPGA, the proposed MGRA with bidirectional interconnect reduced Array area by 76% ( $352,512$  to  $85,108\mu\text{m}^2$ ).

## 6. INTERCONNECT DELAY AND ENERGY EVALUATION

We constructed a circuit model of  $91 \times 44$  crossbar ( $273F \times 264F = 27.3\mu\text{m} \times 26.4\mu\text{m}$ ) using the equivalent circuit model of the 2V-1CAS via-switch. Here, the  $91 \times 44$  crossbar corresponds to the proposed MGRA with “bidir.” interconnection in Table 3. Then, by connecting the crossbar circuit models with inter-crossbar 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. Then, we performed the circuit simulation 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 them.

We compared the interconnect delay and energy between the proposed architecture and a conventional architecture. The conventional architecture corresponds to SRAM-based FPGA. The crossbar 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. The crossbar size is estimated by the number of transistors, and the transistor-level netlist of the crossbar is generated. Here, the  $91 \times 44$  crossbar size is  $163.8\mu\text{m} \times 114.4\mu\text{m}$ , and it is 26 times larger compared to the via-switch crossbar.

Figure 7 shows the performance comparison result at 0.5V operation. The reduction ratios of the delay and energy from SRAM-based FPGA are 90% and 93%, respectively. This improvement can contribute to filling the gap between FPGA and ASIC.

## 7. CONCLUSION

This paper proposed a reconfigurable architecture that can exploit the advantages of via-switch in terms of small footprint, BEOL only integration and low on-resistance. An example of application mapping result shows that the proposed mixed grained logic structure with bidirectional interconnect achieved 76% array area reduction compared to fine grained logic structure with unidirectional interconnect. Evaluation results on crossbar performance show that the proposed interconnect structure can achieve

26X higher integration density and reduce interconnect delay and energy by 90% and 93% at 0.5V operation compared to conventional transistor-based crossbars.

## 8. REFERENCES

- [1] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” *IEEE Trans. CAD*, vol.26, no.2, pp.203–215, 2007.
- [2] M. Lin et al., “Performance benefits of monolithically stacked 3-D FPGA,” *IEEE Trans. CAD*, vol.26, no.2, pp.216–229, 2007.
- [3] P. Gaillardon et al., “Design and architectural assessment of 3-D resistive memory technologies in FPGAs,” *IEEE Trans. Nanotechnology*, vol.12, no.1, pp.40–50, 2013.
- [4] S. Tanachutiwat et al., “FPGA based on integration of CMOS and RRAM,” *IEEE Trans. VLSI Systems*, vol.19, no.11, pp.2023–2032, 2010.
- [5] J. Cong and B. Xiao, “FPGA-RPI: A novel FPGA architecture with RRAM-based programmable interconnects,” *IEEE Trans. VLSI Systems*, vol.22, no.4, pp.864–877, 2014.
- [6] X. Tang et al., “A high-performance low-power near-Vt RRAM-based FPGA,” *Proc. ICFPT*, pp.207–215, 2014.
- [7] Y.Y. Liao et al., “Non-volatile 3D-FPGA with monolithically stacked RRAM based configuration memory,” *Dig. ISSCC*, pp.406–408, 2012.
- [8] M. Tada et al., “Polymer solid-electrolyte (PSE) switch embedded on CMOS for nonvolatile crossbar switch,” *IEEE Trans. Electron Devices*, vol.58, no.12, pp.4398–4405, 2011.
- [9] M. Miyamura et al., “Programmable cell array using rewritable solid-electrolyte switch integrated in 90nm CMOS,” *Dig. ISSCC*, pp.228–229, 2011.
- [10] 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, 2012.
- [11] M. Miyamura et al., “0.5-V highly power-efficient programmable logic using nonvolatile configuration switch in BEOL,” *Proc. FPGA*, pp.236–239, 2015.
- [12] M. Miyamura et al., “Low-power programmable-logic cell arrays using nonvolatile complementary atom switch,” *Proc. ISQED* pp.330–334, 2014.
- [13] 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,” *Dig. IEDM*, pp.32–35, 2015.
- [14] M. Tada et al., “Nonvolatile crossbar switch using TiOx/TaSiOy solid-electrolyte,” *IEEE Trans. Electron Devices*, vol.57, no.8, pp.1987–1995, 2010.
- [15] 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, 2013.
- [16] D. Lewis et al., “The Stratix routing and logic architecture,” *Proc. FPGA*, pp.12–20, 2003.
- [17] T. Higashi and H. Ochi, “Area-efficient LUT-like programmable logic using atom switch and its mapping algorithm,” *Proc. ISGIT*, pp.201–204, 2015.
- [18] K. Wakabayashi and T. Okamoto, “C-based SoC design flow and EDA tools: An ASIC and system vendor perspective,” *IEEE Trans. CAD*, vol.19, no.12, pp.1507–1522, 2000.