Wire Load Model for Rapid Power Consumption Evaluation in Early Design Stage of Via-Switch FPGA

Asuka Natsuhara†1 Takashi Imagawa†2 Hiroyuki Ochi†1,2

†1 Graduate School of Information Science and Engineering, Ritsumeikan University
†2 College of Information Science and Engineering, Ritsumeikan University
1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577 Japan
is0271pp@ed.ritsumei.ac.jp, takac-i@fc.ritsumei.ac.jp, ochi@cs.ritsumei.ac.jp

Abstract— This paper proposes a wire load model for via-switch FPGA to allow simulation-based power estimation before routing. Via-switch FPGA is expected to achieve a dramatic improvement in the area, delay, and power compared with conventional SRAM-based FPGA. To estimate the power consumption of an application circuit mapped on a via-switch FPGA, time-consuming routing process was needed before circuit simulation. Using the proposed post-placement simulation flow, runtime for power estimation is reduced by 63.8% on average compared with the conventional post-routing simulation flow, with 11.8% degradation of estimation error on average.

I. Introduction

The recent progress of manufacturing processes improved the scale, performance, and energy efficiency of circuits that application specific integrated circuits (ASICs) can realize. On the other hand, however, higher costs and longer time for development and fabrication of ASICs have become a serious problem. As an alternative for the ASICs, reconfigurable devices that allow the users to change the circuit configuration after fabrication are getting the attention. Among reconfigurable devices, FPGAs that hold configuration information in SRAM cells (SRAM-type FPGAs) have made remarkable progress, thanks to advances in CMOS process technology. However, SRAM-type FPGAs are far worse in area, delay, and power consumption than ASICs in exchange for their flexibility [1]. One of the factors is that the area of the SRAM cell used to hold configuration information is large.

To reduce the area, power and performance overhead of conventional FPGAs, a new reconfigurable device has been proposed that realizes the ON/OFF switching function between the wire segments by an element called an atom switch instead of an SRAM cell and a pass transistor [2]. Atom switch is a kind of a non-volatile resistive-change switch. It occupies a small footprint area and is integrated on back-end-of-line (BEoL) layers.

In [3], not only the routing resources of FPGA but also the truth table of LUT (Look-Up Table) that serves as the logic resource is realized using atom switches. Reference [4] proposes a 0-1-A-$\overline{A}$ LUT to further improve the area and delay of the LUT in [3]. Thanks to the fact that the atom switch has lower resistance and smaller parasitic capacitance than the pass transistor, one of the LUT input signals of the 0-1-A-$\overline{A}$ LUT is connected to the switch array, unlike conventional FPGAs in which all inputs are connected to the selection signals of the multiplexer (MUX). Reduction of power consumption as well as area and delay has been confirmed for LUT alone [5]. However, it has not been evaluated how much 0-1-A-$\overline{A}$ LUT contributes to low power consumption by implementing benchmark circuits of practical size on an FPGA with a large number of LUTs. To evaluate the power consumption of the entire FPGA, it is necessary to analyze power consumption not only of LUTs but also of wire segments. To evaluate power consumption considering parasitic capacitance of wire segments, however, required time-consuming post-routing circuit simulation.

To eliminate the routing process from power estimation, this paper proposes a wire load model for via-switch FPGA derived from actual placement and routing results of circuits and present the experimental results of power consumption analysis using the model. The standard deviation of the relative estimation error on the number of switches calculated by the proposed post-placement model is 15.0% in the worst case, while that of the post-synthesis model is 46.6%. Using the proposed post-placement simulation flow, runtime for power estimation is reduced by 63.8% on average compared with the conventional post-routing simulation flow, with 11.8% degradation of estimation error on average.

The next section reviews via-switch device and via-switch FPGA architecture, and power estimation method for conventional FPGAs. Section 3 proposes the wire load model that utilizes post-placement information, and Section 4 evaluates the accuracy and runtime of the proposed method. Section 5 concludes the paper.

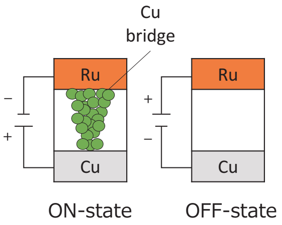


Fig. 1.: Atom switch.

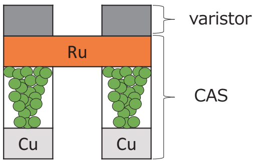


Fig. 2.: Via-switch.

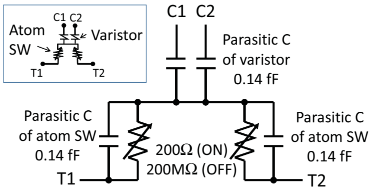


Fig. 3.: Equivalent circuit model of a via-switch [8].

II. Background

A. Atom switch

Atom switch is a nonvolatile switch device that utilizes formation of a copper bridge from metal ion and its extinction [2]. It is composed of a solid-electrolyte sandwiched between copper (Cu) and ruthenium (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 the switch turns OFF. Figure 1 shows the ON- and OFF-state of the atom switch.

To improve the OFF-state reliability of the device, Complementary Atom Switch (CAS) that consists of two atom switches connected in series with opposite direction has been proposed [6]. Via-switch with two varistors stacked on a CAS has also been proposed to allow writing multi-fanout routing to two-dimensional CAS array without using access transistors [7]. Figure 2 and 3 shows the structure of the via-switch and its equivalent circuit model, respectively.

Since the via switch has both state storage and a switch function, it replaces the programmable switch element used in the existing FPGA, that is, an SRAM cell for the state storage and a pass transistor for the switch function. The via-switch is superior in terms of the following four points when compared with the switch consisting of an SRAM element and a pass transistor.

Footprint area : The footprint area of a via-switch is $18F^2$ [7], where F is feature size, while that of conventional switch consisting of an SRAM element and a pass transistor is approximately 10x larger ($\simeq 200F^2$). In addition, via-switch requires metal layers only, and does not consume transistor layers.

ON-resistance : It is possible to achieve low on-resistance (e.g., 200 ohms) of an atom switch, hence that of a CAS with two atom switches connected in series is smaller than that of a pass transistor.

Parasitic Capacitance : The parasitic capacitance of an atom switch is 0.14fF, which is approximately 1/10 of a pass transistor.

Non-volatility : Unlike the SRAM, the via switch has non-volatility because the state of the bridge does not change even when the power is turned off.

B. Crossbar circuit

Achieving both area and programmability has been a challenge for reconfigurable devices. In conventional island-type FPGAs, switch blocks and connection blocks are used to realize programmable routing resources. In the via switch FPGA, a crossbar structure is used for this purpose. Figure 4 shows a crossbar circuit as an example. Via-switches are placed at each cross point of the crossbar, and a vertical and a horizontal track are connected by turning ON the switch at the crosspoint. Since the crossbar circuit in Fig. 4 has 6 × 6 tracks, 36 via-switches exist at the intersections. In the crossbar of the via-switch FPGA, at most one switch is allowed to be in the ON-state in the same row.

C. 0-1-A-$\overline{A}$ LUT

In the via-switch FPGA, 0-1-A-$\overline{A}$ LUT proposed in [4] is used as a programmable logic resource. As an example, Fig. 5 shows a 4-input 0-1-A-$\overline{A}$ LUT. This LUT has 4-bit inputs A, B, C, D, and 1-bit output X, and consists of an 8-row by 4-column switch matrix and an 8-input MUX. In the figure, the eight-input MUX is realized by seven two-input MUXs. A via-switch exists at each intersection of the left array part. Exactly one of the switches in each row is turned to an ON-state. Thereby, any one of 0, 1, A, and $\overline{A}$ can be selected as the value corresponding to each combination of the values of the inputs B, C, and D, and an arbitrary 4-input logic function can be realized by rewriting the states of via-switchs. In this research, a combination of a 0-1-A-$\overline{A}$ LUT and a DFF is used as a logic block (LB).

D. Power consumption analysis for FPGAs

There is an increasing demand for power saving in FPGAs because FPGA consumes more power than ASIC in exchange for flexibility, Power optimization for FPGA using power consumption analysis has been investigated. Power analysis of application circuits implemented on FPGA has been attempted at various stages of the design

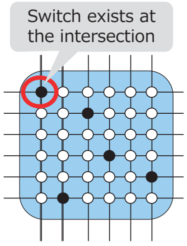


Fig. 4.: Crossbar circuit.

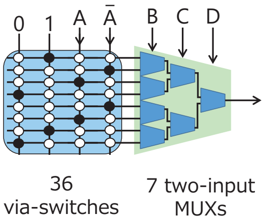


Fig. 5.: 0-1-A-$\overline{A}$ LUT.

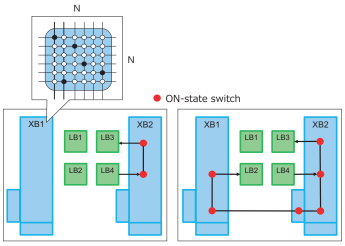


Fig. 6.: Architecture overview and routing examples.

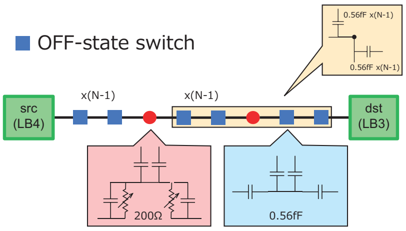


Fig. 7.: Equivalent circuit of the left side path of Fig. 6.

flow. At a low design level where the wiring length after placement and routing can be determined accurately, the accuracy increases but the execution time becomes longer. On the other hand, at high design levels such as RTL, accuracy would be lower but execution time is much shorter. Thus, there is a trade-off between accuracy and execution time of power analysis.

The existing methods for dynamic and static power analysis for FPGAs are briefly summarized below [9, 10]. Dynamic power estimation consists of two components, load capacitance estimation and switching activity estimation. In the former, when the result of placement and routing is not available, the method of using the estimated wire length has been widely used, and in the latter, several methods have been reported, including methods to calculate the average switching behavior using a specific power estimation formula for FPGAs, methods based on statistical model, and those using simulation results with random input vector.

Static power estimation methods have been proposed, such as an analytical method using transistor model parameters and a macro model construction method based on simulation.

Above methods are all for SRAM based FPGAs, and it is necessary to establish a power model for accurate and efficient analysis in via-switch FPGAs.

E. *Power analysis method when using placement and routing results*

An overview of the architecture assumed in this paper is shown in Fig. 6. LB1 to LB4 in the figure are logic blocks (LB) composed of the LUT in Fig. 5 etc., and XB1 to 2 in the figure represent the crossbar circuits in Fig. 4. One tile consists of four LBs and two XBs, and Fig. 6 shows a tile array of 1 × 2 tiles. Also, the black lines in the left side of Figure 6 shows a path connecting LB4 to LB3, and the black lines on the right side is the path connecting LB4 to LB3 and LB2. The red circles shows the ON-state via-switches in these paths.

Figure 7 shows the equivalent circuit of the path of fan-out 1 on the left side of Fig. 6. Note that each track in the path has one ON-state switch and $(N-1)$ OFF-state switches, where N is the number of the tracks of the crossbar. In power consumption analysis, a via-switch in the OFF-state contributes as a parasitic capacitance. As shown in Fig. 3, the parasitic capacitance per switch in the OFF-state is 0.56 fF, so the parasitic capacitance per track is $0.56(N-1)$ fF. Similarly, the equivalent circuit of the fan-out 2 path in the right side of Fig. 6 is as shown in Fig.8. In this way, if netlist is created using each element obtained by placement and routing results, power consumption can be calculated by performing HSPICE simulation.

In this study, we use 0-1-A-$\overline{A}$ LUTs with k=4, 5, and 6 with inverters whose sizes are optimized in terms of the energy-delay product (ED product). We set N = 100. We set the tile array size for each circuit so that wiring completes successfully.

F. *Power analysis using naive method and its problem*

As outlined in the previous section, we can obtain power analysis results by generating the net list and conducting circuit simulation using HSPICE. A power consumption analysis example are illustrated in Fig. 9. Power consumption analysis results for four relatively similar scale circuits, bigkey, des, ex5p, and s298 from the MCNC benchmark are shown in Fig. 9(a), (b), (c), and (d), respectively. This figure shows the breakdown of the power consumption of LB and wiring resources when each circuit is placed and routed on a via-switch FPGA with LUT size k being 4, 5, and 6. Here, the target process is the SOTB 65 nm process, the supply voltage 0.55 V, and the temperature 27 °C. Power consumption was analyzed by applying a random pattern of sufficient sequence length.

As demonstrated in Fig. 9, we can obtain valuable re-

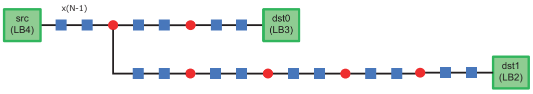
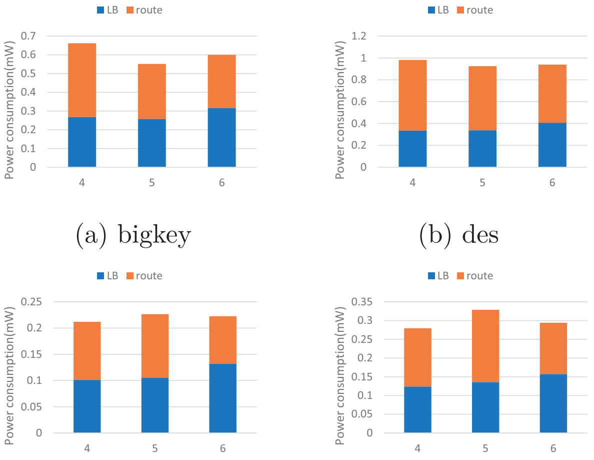


Fig. 8.: Equivalent circuit of the right side path of Fig. 6.



(a) bigkey

(b) des

(c) ex5p

(d) s298

Fig. 9.: Power analysis result using conventional method.

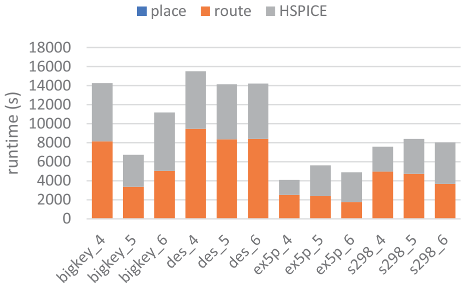
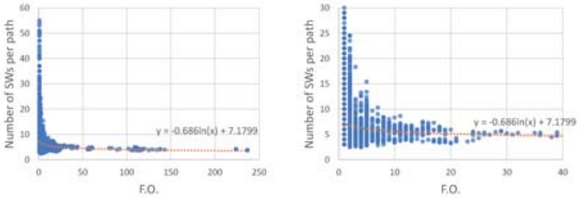


Fig. 10.: Runtime breakdown of conventional method. On average, placement, routing, and circuit simulation processes occupy 0.06%, 54.68%, and 45.26%, respectively.



(a) Overall view.

(b) Close-up view.

Fig. 11.: Relation bwtween F.O. and #switch_path.

sults, including dependency of optimal k and circuits, breakdown of the power consumption, and so on, from the conventional analysis method. As shown in Fig. 10, however, it takes a long runtime using the above method. Especially, the routing process dominates the runtime.

III. Proposed Wire Load Model

To speed up the power analysis of the circuit implemented on the via-switch FPGA, we propose a wire load model to estimate the wiring capacitance without performing the routing process that takes long runtime.

To derive a wire load model, we first examine the relationship between fanout of nets (F.O.) and the number of switches on the nets (#swich_net) using the same benchmark circuit as in the previous section, namely bigkey, des, ex5p, and s298. In this experiment, we mapped each of the four circuits to LUTs of $k = 6$, and performed placement and routing to a via-switch FPGA to determine #swich_net. Figure 11(a) shows the distribution map in which the relationship between F.O. and #switch_net of all four circuits are plotted, and its enlarged view is shown in Fig. 11. The horizontal and vertical axis show F.O. and #switch_path, respectively, where #switch_path is defined as #switch_path = #swich_net/F.O. From these distribution maps, it can be seen that there is a negative correlation between #switch_path and F.O. We introduce a single-variable constant-logarithmic regression model to represent this relationship, which is plotted by an orange line in these figures.

As can be seen from Fig. 11(b), however, there is still a large variation in #switch_path for F.O.< 20, which seems difficult to model with a single variable F.O. To improve the estimation accuracy, we introduce a new variable BOX, which is defined as a half perimeter of a bounding box that surrounds each net which is extracted from the post-placement layout. Note that runtime for placement is negligible compared with those for routing and circuit simulation as demonstrated in Fig. 10.

Since #switch_net is expected to increase monotonically as both BOX and F.O. variables increse, we introduce a two-variable linear regression model #switch_net = a F.O + b BOX + c to estimade #switch_net. Figure 12 exemplifies the relation between BOX and #switch_path for F.O.=1, from which we can observe a positive corelation between BOX and #switch_path. We determined parameters of the two-variable linear regression model as follows.

$$a = 3.287,\ b = 1.600,\ c = -3.918\ (\text{F.O.} \leq 10)$$
$$a = 3.713,\ b = 3.714,\ c = -37.527\ (\text{F.O.} > 10)$$

IV. Experimental Results

Table I shows the total switch count estimated by the proposed one-variable constant-logarithmic regression

TABLE I

: Estimated total switch count and relative error

circuit	estimation method	$k = 4$	$k = 5$	$k = 6$
bigkey	post-routing	21983	14325	17637
	post-synthesis	22546 2.6%	14254 -0.5%	17914 1.6%
	post-placement	22878 4.1%	15305 6.8%	18582 5.4%
des	post-routing	28376	26024	25520
	post-synthesis	27314 -3.7%	24744 -4.9%	24233 -5.0%
	post-placement	29290 3.2%	27304 4.9%	27297 7.0%
ex5p	post-routing	16324	12474	9941
	post-synthesis	15142 -7.2%	12094 -3.0%	10260 3.2%
	post-placement	15892 -2.6%	12072 -3.2%	9821 -1.2%
s298	post-routing	18240	16580	14314
	post-synthesis	16579 -9.1%	15533 -6.3%	14246 -0.5%
	post-placement	16900 -7.3%	15495 -6.5%	13657 -4.6%

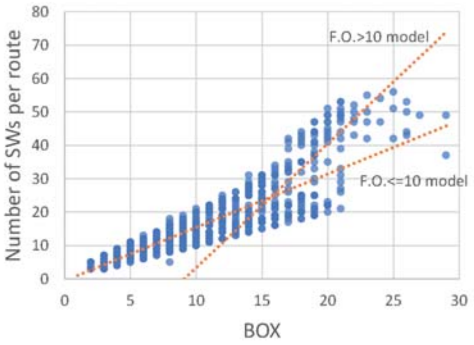


Fig. 12.: Relation between BOX and #switch_path when F.O.=1.

model (post-synthesis) and two-variable linear regression model (post-placement) with actual total switch count (post-routing) for comparison. We used four circuits (bigkey, des, ex5p, s298) and three LUT sizes $k = 4$, 5, and 6 for the experiment. As shown in Table I, the estimation error of the total switch count ranges from -9.1% to 3.2% for the post-synthesis model, and from -7.3% to 7.0% for the post-placement model. Table II shows the standard deviation of the relative estimation error of the proposed models. As can be seen from Table II, the post-placement model is superior in accuracy over the post-synthesis model.

Figure 13 shows the estimated power consumption by the proposed post-placement model in comparison with those using post-routing (actual) #switch. The estimation error is 47.0% in the worst case. Note that the post-placement model successfully identifies an architecture parameter k that achieves the minimum power for each circuit.

TABLE II

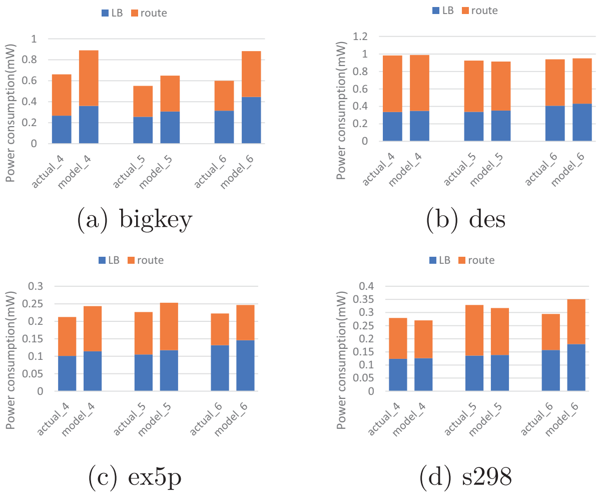
: Standard deviation of the estimation error

circuit	estimation method	$k = 4$	$k = 5$	$k = 6$
bigkey	post_synthesis	43.0%	27.3%	46.6%
	post_placement	12.7%	14.2%	15.0%
des	post_synthesis	14.5%	16.8%	17.5%
	post_placement	9.6%	8.1%	7.2%
ex5p	post_synthesis	9.6%	10.8%	8.7%
	post_placement	6.2%	6.2%	6.8%
s298	post_synthesis	13.0%	14.8%	14.5%
	post_placement	5.9%	5.7%	4.9%

Figure 14 shows the comparison of the runtime for estimating the power consumption of via-switch FPGA with $k = 4$ using the proposed post-placement circuit simulation flow (model) with the post-routing (actual) simulation flow. The execution time is significantly reduced in all four circuits, and 71% in the best case.

V. Conclusion

In this paper, we proposed a wire load model for via-switch FPGA derived from actual placement-and-routing results of circuits and present the experimental results of power estimation accuracy and runtime using the model. The standard deviation of the relative estimation error on the number of switches calculated by the proposed post-placement model is 15.0% in the worst case, while that of the post-synthesis model is 46.6%. Using the proposed post-placement simulation flow, runtime for power estimation is reduced by 63.8% on average compared with the conventional post-routing simulation flow, with 11.8% degradation of estimation error on average. As a fu-



(a) bigkey

(b) des

(c) ex5p

(d) s298

Fig. 13.: Comparison of estimated power

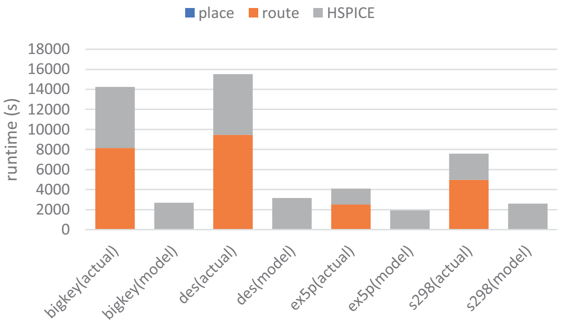


Fig. 14.: Comparison of runtime

ture work, it is desirable to extend the applicability of the model to the via-switch FPGAs including arithmetic blocks (or DSP blocks). By quantitative analysis of the power consumption of the entire circuit implemented on a via-switch FPGA using our model, further development such as effective power optimization method is expected.

Acknowledgement

This work was supported by JST CREST under Grant JPMJCR1432. This work was also supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol.26, no.2, pp.203–215, 2007.

[2] N. Banno, T. Sakamoto, M. Tada, M. Miyamura, Y. Yabe, Y. Saito, S. Ishida, K. Okamoto, H. Hada, N. Kasai, N. Iguchi, and M. Aono, “Reliable solid-electrolyte crossbar switch for programmable logic device,” Proceedings of Symposium on VLSI Technology, pp.115–116, 2010.

[3] M. Miyamura, M. Tada, T. Sakamoto, N. Banno, K. Okamoto, N. Iguchi, and H. Hada, “First demonstration of logic mapping on nonvolatile programmable cell using complementary atom switch,” Proceedings of IEEE International Electron Devices Meeting (IEDM), pp.10.6.1–10.6.4, 2012.

[4] T. Higashi and H. Ochi, “Area-efficient LUT-like programmable logic using atom switch and its mapping algorithm,” Proceedings of 15th International Symposium on Communications and Information Technologies (ISCIT), pp.201–204, 2015.

[5] A. Natsuhara, T. Imagawa, and H. Ochi, “A study on the power efficiency of via-switch oriented programmable logic 0-1-A-$\overline{A}$ LUT,” IEICE Tech. Rep., vol.117, no.377, pp.107–112, 2018. (in Japanese).

[6] M. Miyamura, T. Sakamoto, Y. Tsuji, M. Tada, N. Banno, K. Okamoto, N. Iguchi, and H. Hada, “0.5-V highly power-efficient programmable logic using nonvolatile configuration switch in BEOL,” Proceedings of International Symposium on Field-Programmable Gate Arrays (FPGA), pp.236–239, 2015.

[7] N. Banno, M. Tada, K. Okamoto, N. Iguchi, T. Sakamoto, M. Miyamura, Y. Tsuji, H. Hada, H. Ochi, H. Onodera, M. Hashimoto, and T. Sugibayashi, “A novel two-varistors (a-Si/SiN/a-Si) selected complementary atom switch (2V-1CAS) for nonvolatile crossbar switch with multiple fan-outs,” Proceedings of IEEE International Electron Devices Meeting (IEDM), pp.2.5.1–2.5.4, Dec. 2015.

[8] J. Hotate, T. Kishimoto, T. Higashi, H. Ochi, R. Doi, M. Tada, T. Sugibayashi, K. Wakabayashi, H. Onodera, Y.Mitsuyama, and M. Hashimoto, “A highly-dense mixed grained reconfigurable architecture with overlay crossbar interconnect using Via-switch,” Proceedings of International Conference on Field Programmable Logic and Applications (FPL), pp.272–275, 2016.

[9] A. Rahman and V. Polavarapuv, “Evaluation of low-leakage design techniques for field programmable gate arrays,” Proceedings of International Symposium on Field-Programmable Gate Arrays (FPGA), pp.23–30, 2004.

[10] F. Li, Y. Lin, L. He, and J. Cong, “Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics,” Proceedings of International Symposium on Field-Programmable Gate Arrays (FPGA), pp.42–50, 2004.