Interconnect Delay Analysis for RRAM Crossbar based FPGA

Masanori Hashimoto[†] Yuki Nakazawa[†] Ryutaro Doi [†‡] Jaehoon Yu[†]
[†] Department of Information Systems Engineering, Osaka University
Email: {hasimoto, doi.ryutaro, yu.jaehoon}@ist.osaka-u.ac.jp
[‡] Research Fellow of Japan Society for the Promotion of Science

***Abstract*—FPGAs with novel RRAM-like nano-switches are under development for filling the gap between ASIC and FPGA. In these FPGAs, we need to analyze delay of signal interconnects that include several nano-switches with additional programming interconnects. This paper proposes simplified equivalent circuits for via-switch FPGAs, which enables analysis acceleration without loss of precision. Experimental results show that the proposed simplification increases the circuit simulation speed by 52x and 49x for single-fanout routes and multiple-fanout routes, respectively, on average while the calculation error is within 1.8% on average. When we further apply moment-based delay analysis called D2M to the simplified circuit, the overall average speed up reaches 2,500x and 600x for single-fanout routes and multiple-fanout routes, respectively.**

I. Introduction

Field programmable gate arrays (FPGAs) become more popular since the development cost of application specific integrated circuits (ASICs) is elevating due to the device miniaturization and larger scale integration. However, conventional FPGAs are still inferior to ASICs regarding operating speed, power consumption, and implementation area [1]. These drawbacks arise from a tremendous number of programmable switches that are included in FPGAs to acquire reconfigurability. In static random access memory (SRAM)-based FPGAs, which are the most widely used FPGAs, a programmable switch is composed of a transmission gate for switching and an SRAM cell to hold the on/off-state of the switch. These components consist of transistors, and hence the transmission gate has high resistance and large capacitance, and the SRAM cell having six transistors consumes large area. Therefore, SRAM-based programmable switches lead to the degradation of interconnect performance and area efficiency [2].

To overcome the drawbacks of conventional FPGAs, FPGAs that exploit resistive random access memories (RRAMs) as programmable switches instead of SRAM-based ones are widely studied [3]–[9]. In these RRAM based FPGAs, the crossbar, which has an RRAM switch at each intersection of horizontal signal wire and vertical signal wire, is responsible for signal routing. These RRAM-based FPGAs, however, require one or two access transistors per a programmable switch for switch programming. The access transistor is relatively large despite the small footprint of the RRAM-based switch, and hence it interferes with further area reduction. To eliminate access transistors, nonvolatile via-switches are actively developed [10], [11]. The via-switch consists of atom switches, which are a kind of nonvolatile RRAMs developed for application to FPGAs, and varistors in place of access transistors.

Timing verification is indispensable in digital circuit design. Especially, interconnect delay analysis is important when the wire delay is dominant in the overall delay. The interconnect delay analysis needs to extract parasitic resistance and capacitance values from the given layout and generate equivalent circuit models. Then, we can obtain delay values by simulating the generated equivalent circuit models with, e.g., HSPICE. However, such a circuit simulation based analysis is too time-consuming in general since there are a huge number of wires on a chip. Consequently, fast analysis techniques are demanded and developed accepting small accuracy degradation.

Via-switch FPGA, which uses via-switch for routing crossbar and is studied in [12], has a different interconnect structure from conventional SRAM FPGA, and the degree of freedom of wiring is higher due to bidirectional signaling and on-demand repeater insertion. Therefore, timing analysis for conventional FPGA, which is hardly disclosed though, is not applicable to via-switch FPGA. On the other hand, timing analysis methods for ASICs can cope with arbitrary wiring patterns and hence they are expected to analyze interconnects in via-switch FPGA as well. Besides, even with methods developed for ASICs, we may need to pay attention to the fact that via-switches having several hundred Ω exist in the wiring topology whereas the wire resistance and capacitance per unit length vary at most a few times in ASIC. Therefore, we need to develop a method to generate compact equivalent circuit models taking into account via-switches. On the other hand, via-switch FPGA is organized in an array structure, and hence simplification and speeding-up exploiting this regularity are expected.

In this work, we investigate interconnect delay analysis suitable for via-switch FPGA. We first focus on generating equivalent circuit model taking into account the regular structure and via-switch existence and present a compact model generation method. Then, we apply a moment-based delay analysis method developed for ASICs and evaluate the accuracy and computational time for actual routing patterns in a filter application.

The remainder of this paper is organized as follows. Section II explains the structure of via-switch FPGA and its detailed equivalent circuit model. Section III discusses the

2159-3477/18/$31.00 ©2018 IEEE
DOI 10.1109/ISVLSI.2018.00101

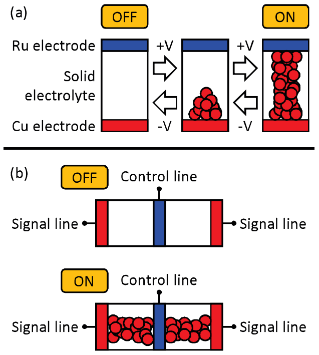


Fig. 1. Structure and operation of (a) atom switch and (b) CAS.

simplification of equivalent circuit model, and Section IV shows the accuracy and computational times for actual wiring patterns. Concluding remarks are given in Section V.

II. Via-switch FPGA

A. Via-switch

The via-switch is a nonvolatile, rewritable, and compact switch that is developed to implement a crossbar switch by Banno et al. [10], and it is composed of atom switches and varistors. Here, we explain the device structure, functionality, and characteristics in the following.

The atom switch consists of a solid electrolyte sandwiched between copper (Cu) and ruthenium (Ru) electrodes as shown in Figure 1(a). By applying a positive voltage to the Cu electrode, a Cu bridge is formed in the solid electrolyte, and the switch turns on. On the other hand, when a negative voltage is applied, Cu atoms in the bridge are reverted to the Cu electrode, and then the switch turns off. The switching between on-state and off-state is repeatable, and each state is nonvolatile. For improving the device reliability, the complementary atom switch (CAS) is devised, where it consists of two atom switches connected in series with opposite direction as shown in Figure 1(b). In the programming of CAS, a pair of signal line and control line supply a programming voltage to each atom switch, and two atom switches are programmed sequentially. During normal operation, on the other hand, only signal lines are used for routing [9].

Figure 2 shows the structure of via-switch, and the varistor is connected to the control terminal of CAS. When a voltage higher than the threshold value (programming voltage) is applied between the signal and control lines, the varistor supplies programming current to an atom switch. On the other hand, the varistor isolates the control lines from the signal lines during normal operation [10]. Figure 2 shows two-varistor-one-CAS (2V-1CAS) structure, which is adopted via-switch FPGA proposed in [12]. This 2V-1CAS structure enables multiple fanouts.

Here, we summarize contribution of via-switch to FPGAs. The footprint, on-resistance, and capacitance are 18 F^2, 200 Ω, and 0.14 fF respectively [10], [12]. Thanks to these characteristics, the area efficiency and performance of via-switch FPGA are dramatically improved compared to SRAM-based one. Ochi et al. report that the crossbar density is improved by up to 26x, and in this case, the delay and energy in the interconnection are reduced by 90% or more [12].

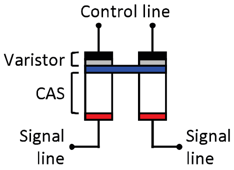
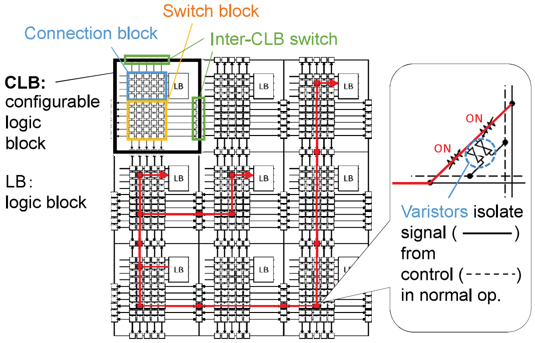


Fig. 2. Structure of 2V-1CAS via-switch.



Switch block
Connection block
Inter-CLB switch
CLB: configurable logic block
LB: logic block
LB
ON
ON
Varistors isolate signal (———) from control (------) in normal op.

Fig. 3. Structure of via-switch FPGA.

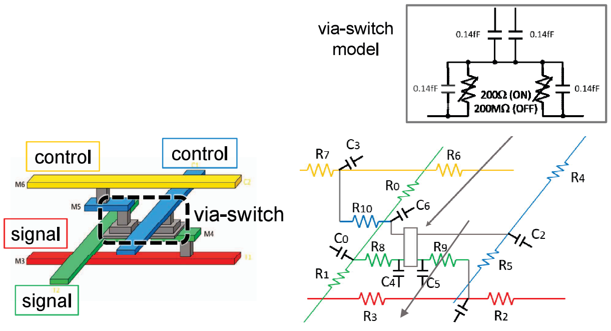
B. Via-switch FPGA structure

The via-switch FPGA consists of an array of configurable logic blocks (CLBs), and each CLB is composed of the logic block and crossbar where a via-switch is placed at each crossbar intersection of signal lines as shown in Figure 3 [12]. The via-switch in the crossbar is responsible for connection and disconnection between the horizontal and vertical signal lines. Besides, the top half of the crossbar provides a function of input and output multiplexers to the logic block and corresponds to connection block in conventional FPGAs. On the other hand, the bottom half of the crossbar, which corresponds to switch block, routes global interconnections. The logic block organizes combinational and sequential circuits.

Figure 3 also illustrates a signal routing with red lines. The routing starts from the bottom-left LB and ends at three LBs. The small red boxes correspond to on-state via-switches. Within CLB, the on-state via-switch connects the vertical and horizontal signal lines. Also, for making a connection to the adjacent CLBs, inter-CLB switches are turned on. The right figure details the via-switch that have two control lines in addition to two signal lines, whereas the control lines are omitted in the left figure. Two varistors are located between the red signal route and control lines. The control lines are used only for via-switch programming. After the programming, the varistors isolate the control lines from the signal route.

C. Equivalent circuit of interconnect in via-switch FPGA

Figure 4(a) depicts the three-dimensional structure of the intersection with a via-switch. There are two orthogonal signal lines and two orthogonal control lines. Figure 4(b) shows an equivalent circuit of the intersection, where the same color



(a) 3D structure (b) Equivalent circuit

Fig. 4. Via-switch structure and equivalent circuit.

indicates the same wire in Figure 4. The via-switch circuit model is found in the top. The two variable resistors are atom switches, and the top two capacitors correspond to the varistors [12]. Other two capacitors represent the input capacitors of off-state atom switches. When the atom switches are on, this input capacitors can be ignored. The on-resistance value of each via-switch depends on the amount of programming current, but in this work, it is assumed to be 400 Ω, which means 200Ω for each atom switch. The off-state via-switch is 400 MΩ.

By connecting this equivalent circuit model, we can construct an equivalent circuit model for each routing. In [12], interconnect delay and energy performances are evaluated using the constructed circuit model and a circuit simulator. On the other hand, there are many circuit elements in the constructed model, and hence its circuit simulation needs long CPU time, which is not suitable for timing verification and optimization in the design phase.

III. Simplifying circuit model for delay analysis

Figure 5 shows a procedure of interconnect delay analysis. After the application mapping, we extract resistance and capacitance values from the physical layout and construct an equivalent circuit model. Then, we analyze the equivalent circuit model to derive the propagation delay times from the source node to the destination nodes. Here, there is a trade-off between the complexity of the equivalent circuit model and the accuracy of the circuit representation in general. Also, the complexity of the model affects the computational time of delay analysis. Therefore, it is important to construct a sufficiently accurate equivalent circuit model for efficient delay analysis. This section discusses such a circuit model construction for delay analysis of via-switch FPGA. As for the delay analysis, a number of publications are available, and there are several well-known methods, such as PRIMA [13] and D2M [14]. Depending on the required accuracy and allowable CPU time, one of them should be selected.

A. Control line elimination

We first eliminate the control lines that are necessary for via-switch programming but unnecessary for signal routing. We leave the two varistor capacitances of 0.14 fF in Figure 4(b), remove the blue and yellow wire resistances and related capacitances of control lines and connect the upper varistor capacitance terminals to ground. Due to the impedance of control lines, the varistor capacitances may work as smaller capacitances during the dynamic signal transition, but the conservative calculation is preferable in timing analysis, and hence this treatment is supposed to be reasonable.

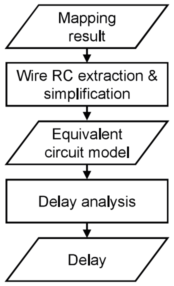


Fig. 5. Delay analysis procedure.

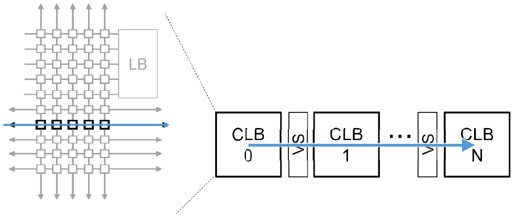


Fig. 6. Routing that propagates a signal through CLBs without bents.

TABLE I
Impact of control line elimination.

Routing distance (#CLBs)	Delay [ps]		Error [%]
	w/ ctl. lines	w/o ctl. lines	
1	15.1	15.1	0.0
3	50.6	50.6	0.0
5	92.6	92.6	0.0
10	228	228	0.0
30	1190	1190	0.0

We experimentally evaluate the impact of this control line elimination. Figure 6 shows the routing under evaluation, where a logic signal is propagating straight through several CLBs and inter-CLB via-switches. The detailed conditions are as follows, where the same setup is used throughout the paper.

- Crossbar size: 163×96
- Via-switch resistance: 400 Ω (ON) / 400 MΩ (OFF)
- Driver resistance: 1 kΩ
- Other signals: grounded

Table I shows the delay times before and after the control line elimination. We can see that the control line elimination does not degrade the accuracy of the circuit representation. In the following, the control lines are eliminated.

B. Simplification inside CLB without bents

Next, the signal line simplification is discussed. We take a strategy that applies the simplification to the route within a CLB, and the simplified CLB models and inter-CLB switches are connected to compose an entire equivalent circuit for delay analysis. This subsection focuses on the within-CLB routing without bents in Figure 7. The following discussion supposes

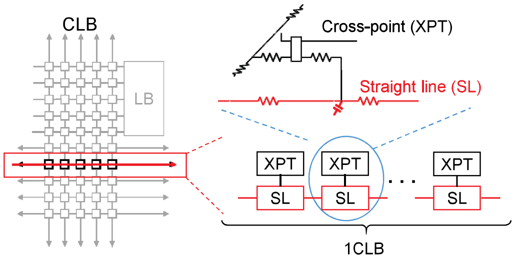


Fig. 7. Within-CLB routing without bents.

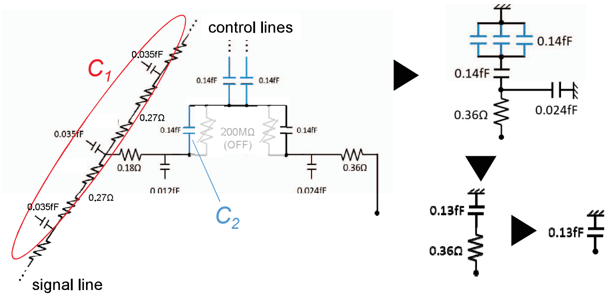


Fig. 8. Equivalent circuit simplification at Step 1.

horizontal direction, but the same discussion can be done for the vertical direction.

No bents in the CLB means that there are no on-state via-switches on the route. We first discuss how to treat off-state via-switches on the route of interest. Figure 7 shows that each intersection with a via-switch consists of straight lines (SL) and cross-points (XPT), where SL corresponds to the wire on the routing and XPT represents the off-state via-switch and the wires beyond the switch. For a propagating signal on the routing consisting of SLs, XPTs are distributed loads. We simplify this distributed RC tree with the following two steps.

1) Step1: Replacing XPT as capacitance: We first replace each XPT as a capacitance. Figure 8 explains this replacement. The right bottom terminal in the left figure is connected to SL, and hence we need to investigate how this left figure behaves as a load from this terminal. Now, the atom switches are off, and hence the switch resistive impedance is thought to be much higher than that of the parallel capacitance C_2. Next, we focus on C_1, which is the sum of the capacitances beyond the off-state atom switches. Here, the vertical signal wire is connected to 163 via-switches, and then C_1 is much larger than C_2. When two capacitances are connected in series and one is much larger than the other, the total impedance is determined by the smaller capacitance. Therefore, we eliminate the circuitry beyond C_2 and obtain the circuit in the top right figure. Next, we replace four capacitances as a single capacitance. Here, the time constant of this RC circuit is smaller than 0.1 fs, and hence the resistance can be ignored taking into account ordinary signal switching speed. Thus, we can replace an XPT as a capacitance of 0.13 fF as shown in Fig. 9.

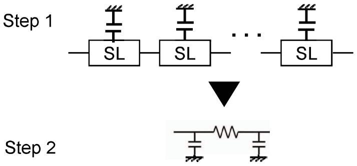


Fig. 9. Two-step simplification of equivalent circuit.

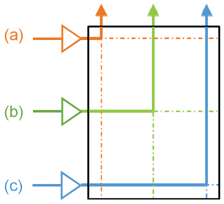


Fig. 10. Different left-to-top routes within a CLB.

2) Step2: Replacing RC tree with π circuit: An RC ladder circuit is often replaced with a CRC π circuit, which is shown in Fig. 9. The number of π stages is determined by the required accuracy. Here, the total resistance of the RC ladder circuit is smaller than the on via-switch impedance of 400 Ω, and hence a single-π circuit is thought to be enough. In this case, the number of RC elements is reduced from 1,536 to 3.

C. Simplification inside CLB with bents

Next, we discuss the simplification of CLB that includes an on-state via-switch on the signal route of interest.

We first examine the impact of on-state via-switch location on the propagation delay. If the impact is very small, we can provide the same circuit model irrelevant to the location of the on-state via-switch. Let us suppose three routes depicted in Fig. 10, where (a) and (c) are the shortest and longest routes, respectively, and (b) is the route that consists of the middle horizontal line and the middle vertical line. We attached a driver whose output impedance was 1 kΩ and evaluated the delay from the driver input to the CLB output with a circuit simulator. Table II shows the result. We can see that the delay difference due to the location of the on-state via-switch is within 2% and quite limited. This is because the total capacitance of the route is identical independent of the on-state via-switch since the capacitances connected to the dotted lines in addition to those connected to the solid lines must be charged and discharged. This result suggests that the location-independent model is accurate enough in most cases, which simplifies the equivalent circuit construction for delay analysis.

According to the above observation, we construct an equivalent circuit model of a CLB that includes a single bent as shown in Fig. 11. The horizontal and vertical lines are represented as two-π RC models whose middle nodes correspond to the horizontally and vertically middle locations of the crossbar, respectively. Here, for realizing route (b) in Fig. 10, two-π model is selected while one-π model is chosen for a straight line without bents in Section III-B. Then, we attach an on-state via-switch model, which is also expressed as a two-π RC model, between these two middle nodes. Note that this model is independent of the signal direction, and hence the same

TABLE II
DELAY DIFFERENCE DEPENDING ON LOCATION OF ON-STATE VIA-SWITCH.

Route	(a)	(b)	(c)
Delay [ps]	40.9	41.7	42.3
Diff. from (b) [%]	-1.81	0	1.53

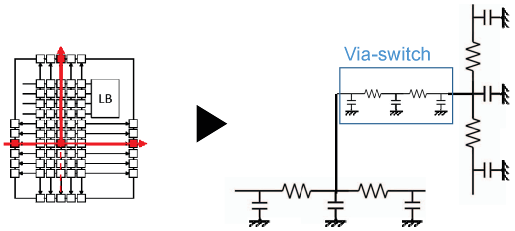


Fig. 11. Within-CLB routing with a bent and its equivalent circuit.

model can be used for all the signaling directions. Similarly, by inserting the via-switch model into the bent location, we can construct a within-CLB equivalent circuit model that has two bents on the route of interest.

Now, given the within-CLB models with and without bents, we can construct an equivalent circuit model for arbitrary routes by connecting the within-CLB models. The next section evaluates the accuracy of delay computation.

IV. Experimental results

This section evaluates the accuracy degradation and the speed-up of delay analysis for routing patterns found in an actual mapping result. Figure 12 shows an FIR (Finite Impulse Response) filter mapped to a 10×5 CLB array [12]. The crossbar size, via-switch resistance, and driver resistance are the same as Section III-A. We assume that the interconnects except the route of our interest are grounded.

A. Routes of single fanout

We first compare the delay times and CPU times before and after the circuit simplification discussed in the previous section. Here, the single-fanout routes are evaluated. Figure 13 shows the delay comparisons between the original circuit and simplified circuit, where each circuit is simulated by HSPICE. We can see that the delay computed with the simplified circuit is highly correlated with that of the original circuit. The averages of absolute and relative errors are 4.1 ps and 1.2 %, respectively. Figure 14 shows the histogram of the simulation speed-up ratio thanks to the circuit simplification. We can get 40x to 80x simulation speed-up in most cases, and the average and the maximum are 52x and 124x, respectively.

Next, we apply D2M [14] to the simplified circuit to compute the propagation delay aiming at further speed-up. We compare the delay and CPU times of the original circuit and HSPICE with those of the simplified circuit and D2M. Figure 15 shows the delay comparisons, which also includes Elmore delay results. We can see that the delay computed with the simplified circuit and D2M is highly correlated with that of the original circuit and HSPICE, whereas Elmore delay model overestimates the delay, as we expected. The averages of D2M absolute and relative errors are 4.4 ps and 1.2 %, respectively. Note that only fanout-1 interconnects are analyzed whereas D2M may lose the accuracy for interconnects with branches. Our near future work investigates the accuracy of D2M for multiple-fanout interconnects. Figure 16 shows the histogram of the overall speed-up ratio. The average and the maximum speed-up ratios are 2,500x and 6,600x, respectively.

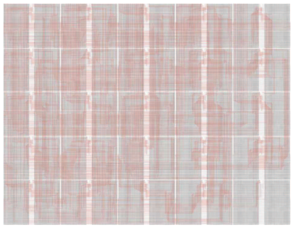


Fig. 12. Mapping result.

B. Routings of multiple fanouts

We carried out the similar evaluations to multiple-fanout routes. Figures 17 and 18 shows the delay and CPU time comparisons between the original circuit and the simplified circuit. We can see that the accuracy degradation is almost invisible and the average error is 1.8%. The average simulation speed-up is 49x.

We also evaluated the delay using D2M and Elmore delay model. Figure 19 shows the delay comparison. We can see that the accuracy of D2M is not as good as the single-fanout case of Figure 15 and the average error is 8.5%, which is due to several outliers. When the routing length is much different for different sink terminals, D2M suffers from delay overestimation of the route to the nearest sink. On the other hand, most of the routes are still analyzed well by D2M. Figure 18 shows the histogram of speed-up ratio, where the average speed-up is 600x. We observe that larger-fanout wires tend to have smaller speed-up ratios. There are two groups in the histogram, where the 400x group corresponds to large-fanout routes, such as eight and ten.

The above results indicate that we conclude that we should construct a delay analysis framework that screens some outlier routes for analyzing them with more sophisticated method, such as PRIMA, and computes the delays of remaining majority routes with D2M.

V. Conclusion

This paper investigated the interconnect delay analysis for via-switch FPGA mainly focusing on the equivalent circuit simplification. The off-state via-switches are replaced with capacitances, and the distributed RC ladder within a CLB is simplified as a CRC one-π model or CRC two-π models connected with CRC π model of via-switch. Experimental results for actual fanout-1 routing patterns show that the proposed circuit simplification attains 52x and 49x speed-up for single- and multiple-fanout routes on average, respectively, within 1.8% average error. Combination of moment-based delay analysis of D2m and the circuit simplification achieves

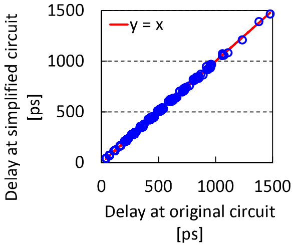


Fig. 13. Delay comparison between original and simplified circuits (single fanout).

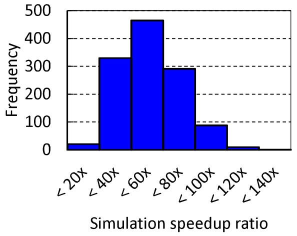


Fig. 14. CPU time comparison between original and simplified circuits (single fanout).

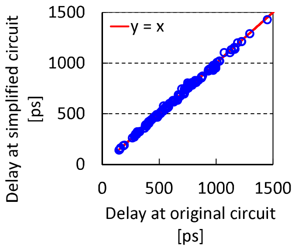


Fig. 17. Delay comparison between original and simplified circuits (multiple fanouts).

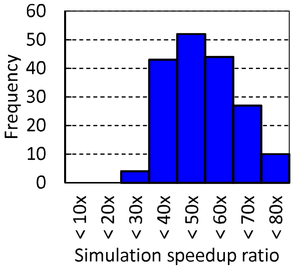


Fig. 18. CPU time comparison between original and simplified circuits (multiple fanouts).

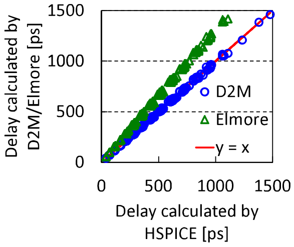


Fig. 15. Delay comparison between HSPICE with original circuit and D2M/Elmore with simplified circuit (single fanout).

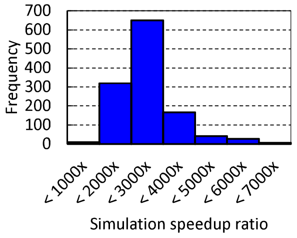


Fig. 16. CPU time comparison between HSPICE with original circuit and D2M with simplified circuit (single fanout).

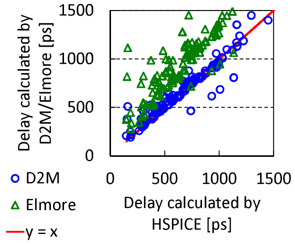


Fig. 19. Delay comparison between HSPICE with original circuit and D2M/Elmore with simplified circuit (multiple fanouts).

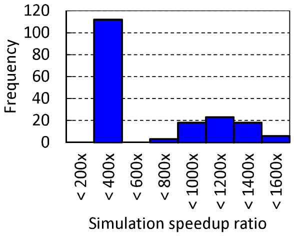


Fig. 20. CPU time comparison between HSPICE with original circuit and D2M with simplified circuit (multiple fanouts).

2,500x and 600x speed-up for single- and multiple-fanout routes on average, respectively. These results reveal that most of the routes can be processed with the proposed simplification and D2M and more sophisticated delay analysis methods should analyze the other routes.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP17J10008 and JST CREST Grant Number JPMJCR1432, Japan.

References

[1] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, Feb 2007.

[2] M. Lin, A. E. Gamal, Y. C. Lu, and S. Wong, "Performance Benefits of Monolithically Stacked 3-D FPGA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 216–229, Feb 2007.

[3] P. E. Gaillardon, D. Sacchetto, G. B. Beneventi, M. H. B. Jamaa, L. Perniola, F. Clermidy, I. O ' Connor, and G. D. Micheli, "Design and Architectural Assessment of 3-D Resistive Memory Technologies in FPGAs," *IEEE Transactions on Nanotechnology*, vol. 12, no. 1, pp. 40–50, Jan 2013.

[4] S. Tanachutiwat, M. Liu, and W. Wang, "FPGA Based on Integration of CMOS and RRAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 11, pp. 2023–2032, Nov 2011.

[5] J. Cong and B. Xiao, "FPGA-RPI: A Novel FPGA Architecture With RRAM-Based Programmable Interconnects," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 864–877, April 2014.

[6] X. Tang, P. E. Gaillardon, and G. D. Micheli, "A high-performance low-power near-Vt RRAM-based FPGA," in *2014 International Conference on Field-Programmable Technology (FPT)*, Dec 2014, pp. 207–214.

[7] Y. Y. Liauw, Z. Zhang, W. Kim, A. E. Gamal, and S. S. Wong, "Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory," in *2012 IEEE International Solid-State Circuits Conference*, Feb 2012, pp. 406–408.

[8] K. Okamoto, M. Tada, T. Sakamoto, M. Miyamura, N. Banno, N. Iguchi, and H. Hada, "Conducting mechanism of atom switch with polymer solid-electrolyte," in *2011 International Electron Devices Meeting*, Dec 2011, pp. 12.2.1–12.2.4.

[9] M. Miyamura, T. Sakamoto, M. Tada, N. Banno, K. Okamoto, N. Iguchi, and H. Hada, "Low-power programmable-logic cell arrays using nonvolatile complementary atom switch," in *Fifteenth International Symposium on Quality Electronic Design*, March 2014, pp. 330–334.

[10] 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," in *2015 IEEE International Electron Devices Meeting (IEDM)*, Dec 2015, pp. 2.5.1–2.5.4.

[11] N. Banno, M. Tilda, K. Okamoto, N. Iguchi, T. Sakamoto, H. Hada, H. Ochi, H. Onodera, M. Hashimoto, and T. Sugibayashi, "50x20 crossbar switch block (CSB) with two-varistors (a-Si/SiN/a-Si) selected complementary atom switch for a highly-dense reconfigurable logic," in *2016 IEEE International Electron Devices Meeting (IEDM)*, Dec 2016, pp. 16.4.1–16.4.4.

[12] H. Ochi, K. Yamaguchi, T. Fujimoto, J. Hotate, T. Kishimoto, T. Higashi, T. Imagawa, R. Doi, M. Tada, T. Sugibayashi, W. Takahashi, K. Wakabayashi, H. Onodera, Y. Mitsuyama, J. Yu, and M. Hashimoto, "Via-Switch FPGA: Highly-Dense Mixed-Grained Reconfigurable Architecture with Overlay Via-Switch Crossbars," *IEEE Transactions on VLSI Systems*, in press.

[13] A. Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 645-654, Aug 1998.

[14] C. J. Alpert, A. Devgan and C. V. Kashyap, "RC delay metrics for performance optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 5, pp. 571-582, May 2001.