A Tuning Method of Programmable Delay Element with Two Values for Yield Improvement

Hayato Mashiko Yukihide Kohira
School of Computer Science, the University of Aizu
Aizu-Wakamatsu City, Fukushima, 965-8580 Japan
{m5161143,kohira}@u-aizu.ac.jp

Abstract—Due to progressing the process technology in LSI and increasing variations of wire and gate delays after fabrication, timing violations cause significant reduction in the yield of LSI chips. To recover the timing violations, programmable delay elements called PDEs are inserted into the clock tree before fabrication and their delays are tuned after fabrication. In this paper, we use PDEs with two delay values and propose a delay tuning method of the PDEs to improve the yield and to reduce the number of tests. Moreover, we evaluate the circuits obtained by the proposed method on commercial CAD tools. Experimental results show that the proposed method is effective.

I. Introduction

Due to progressing the process technology in LSI, the performance of LSI circuits has been advanced. Progressing the process technology increases the process variation. The process variation in fabrication causes the delay variation and the delay variation causes the timing violation. The yield is reduced significantly by the timing violation, and the cost of LSI chips has increased. In general, the addition of margins and the statistical static delay analysis have been applied to satisfy the timing constraints. However, since these methods need to estimate the delay variations before fabrication, they are not essential solutions in the situation in which the uncertainty is increasing in fabrication process.

Recently, deskew methods have been proposed to recover the timing violations. This method inserts the elements called Programmable Delay Elements (PDEs), whose delays can be tuned after fabrication, into the clock tree before fabrication. When some timing violations are found in test process, delays of PDEs are tuned to recover timing violations.

In [1], under assumptions that the delay of each PDE can be set to an arbitrary integer and delays between all register pairs are given, the delay tuning method formulated to integer linear programming (ILP) has been proposed. Since the formulation of ILP is totally unimodular, it can be solved in polynomial time. Therefore, the optimality of the delay timing method is guaranteed and it can be expected to improve the yield highly in short computational time. However, the structure of the PDE with arbitrary integer values is not considered in [1]. Furthermore, since measuring the exact delay for each register pair is required to apply this delay tuning method, the cost for this delay tuning method is high.

In [2], given PDEs with discrete values, a delay tuning method formulated to ILP and a heuristic method have been proposed. The optimality of the delay tuning method formulated to ILP is guaranteed and can be expected to improve the yield highly. However, the delay tuning method is unsuitable to large scale circuits because ILP typically takes long computation time. On the other hand, although the heuristic delay tuning method can be solved in short computation time, it cannot be expected to improve the yield enough since it does not guarantee the optimality. Moreover, when the PDEs with discrete values are constructed by only basic logic gates, the circuit size with the PDEs is larger and the power consumption in the circuit must be increased.

When PDEs with two delay values are constructed by only basic logic gates, the circuit size with the PDEs can be smaller. The delay tuning method of the PDEs can be obtained by applying 2-clustering method [3] at the given clock cycle time in polynomial time. However, the number of tests is considerable in this delay tuning method because whether the timing constraints are violated for any register pairs with signal propagations at all PDE delay tuning patterns must be checked by the path-delay-test [4].

In this paper, we use the PDEs with two delay values and propose a delay tuning method of the PDE to improve the yield and to reduce the number of tests. Moreover, we evaluate the circuits obtained by the proposed method on commercial CAD tools. Experimental results show that the proposed method is effective under Monte Carlo simulation which uses the delay information obtained by commercial CAD tools.

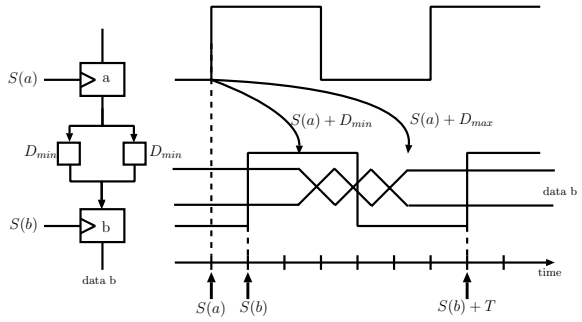

Fig. 1. Timing constraints in clock synchronous circuit.

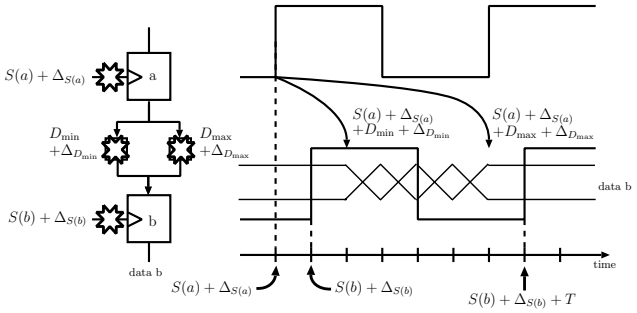

Fig. 2. Timing constraints with delay variations in clock synchronous circuit.

II. Preliminaries

A. Timing Constraint

To work clock synchronous circuits correctly, timing constraints between register pairs with signal propagation must be satisfied (Fig. 1) [5]. Suppose that signal is propagated from register a to register b. The timing constraints consist of hold constraint and setup constraint as follows:

Hold Constraint:

$$S(b) - S(a) \quad \leq \quad D_{\min}(a, b)$$

Setup Constraint:

$$S(a) - S(b) \quad \leq \quad T - D_{\max}(a, b),$$

where T is the clock period, $D_{\max}(a, b)$ ($D_{\min}(a, b)$) is the maximum (minimum) delay from register a to register b, and $S(a)$ ($S(b)$) is the clock timing of register a (b). Especially, violating the hold (setup) constraint is described as *hold* (*setup*) *violation* and violating the hold constraint or the setup constraint is described as *timing violation*.

Wire and gate delays are varied after fabrication (Fig. 2). Thus, delays between register pairs and clock timings are varied and the timing constraints in the clock synchronous circuit with delay variations are transformed as follows:

$$\begin{aligned} (S(b) \quad &+ \quad \Delta_{S(b)}) - (S(a) + \Delta_{S(a)}) \\ &\leq \quad D_{\min}(a, b) + \Delta_{D_{\min}} \qquad (1) \\ (S(a) \quad &+ \quad \Delta_{S(a)}) - (S(b) + \Delta_{S(b)}) \\ &\leq \quad T - (D_{\max}(a, b) + \Delta_{D_{\max}}), \qquad (2) \end{aligned}$$

where $\Delta_{\max}$ ($\Delta_{\min}$) is the delay variation of maximum (minimum) delay from register a to register b, and $\Delta_{S(a)}$ ($\Delta_{S(b)}$) is the delay variation of clock timing $S(a)$ ($S(b)$). By converting eq. (1) and eq. (2) to eq. (3) and eq. (4), all delay variations are regarded as the delays between register pairs.

$$\begin{aligned} S(b) - S(a) \quad &\leq \quad D'_{\min}(a, b) \qquad (3) \\ S(a) - S(b) \quad &\leq \quad T - D'_{\max}(a, b), \qquad (4) \end{aligned}$$

where

$$\begin{aligned} D'_{\min}(a, b) \quad &= \quad D_{\min} - (\Delta_{S(a)} + \Delta_{S(b)} - \Delta_{D_{\min}}) \\ D'_{\max}(a, b) \quad &= \quad D_{\max} - (\Delta_{S(a)} - \Delta_{S(b)} + \Delta_{D_{\max}}). \end{aligned}$$

Hereinafter, we discuss that all delays are varied between register pairs.

B. Delay Tuning of PDEs

In the delay tuning methods for PDEs [1,2,6,7], the delays of PDEs inserted into the clock tree before fabrication are tuned to recover the timing violations caused by the delay variations after fabrication. The timing constraints in the clock synchronous circuit with delay variations considering delays of PDEs are as follows:

$$\begin{aligned} S'(b) - S'(a) \quad &\leq \quad D'_{\min}(a, b) \qquad (5) \\ S'(a) - S'(b) \quad &\leq \quad T - D'_{\max}(a, b) \qquad (6) \end{aligned}$$

where $S'(a) = S(a) + d_a, S'(b) = S(b) + d_b$ and $d_a(d_b)$ is the delay of PDE inserted for register a (b).

If the hold constraint is violated, the left side in eq. (5) must be decreased to recover it. Therefore, the delay of PDE inserted for register a is increased or the delay of PDE inserted for register b is decreased. Similarly, if the setup constraint is violated, the left side in eq. (6) must be decreased to recover it. Therefore, the delay of PDE inserted for register a is decreased or the delay of PDE inserted for register b is increased.

C. 2-clustering Method [3]

In clock skew scheduling [3], the set of registers with same clock timing is called *cluster* and partitioning registers to clusters is called *clustering*.

TABLE I
DEFINITION OF MAXTERM.

x_a	x_b	$S(a)$	$S(b)$	Timing constraint of (a,b) Satisfaction	Violation
0	0	0	0	$c_1^{(a,b)} = 1$	$c_1^{(a,b)} = x_a \vee x_b$
0	1	0	d	$c_2^{(a,b)} = 1$	$c_2^{(a,b)} = x_a \vee \overline{x_b}$
1	0	d	0	$c_3^{(a,b)} = 1$	$c_3^{(a,b)} = \overline{x_a} \vee x_b$
1	1	d	d	$c_4^{(a,b)} = 1$	$c_4^{(a,b)} = \overline{x_a} \vee \overline{x_b}$

The 2-clustering problem in which the number of cluster is equal to or less than two is defined as follows:

2-clustering Problem

Inputs: Delays between register pairs, clock period T and clock skew d ($d \geq 0$)

Output: The clock timing for each register

Constraint: All timing constraints are satisfied at the given clock period T.

In 2-clustering method, the 2-clustering problem is converted to 2-SAT problem and it is solved by a 2-SAT solver. 2-SAT problem assigns logic variable x_i ($1 \leq \forall j \leq n$) in the set of logic variables $X = \{x_1, x_2, \ldots, x_n\}$ to 0 or 1 so that a logic function $C(X) = c_1(X) \wedge c_2(X) \wedge \cdots \wedge c_m(X)$ equals to 1. Since 2-SAT problem can be solved in polynomial time [8], the 2-clustering problem can also be solved in polynomial time.

The logic variable x_a in 2-SAT problem corresponds to the clock timing $S(a)$ of register a in the 2-clustering problem and $S(a) = 0$ ($S(a) = d$) if and only if $x_a = 0$ ($x_a = 1$). For each register pair with signal propagation in the 2-clustering problem, four maxterms $c_k^{(a,b)}$ are defined by whether or not the timing constraint is satisfied at the clock timings as Table I. The logic function is defined by $C(X) = \bigwedge(c_1^{(a,b)} \wedge c_2^{(a,b)} \wedge c_3^{(a,b)} \wedge c_4^{(a,b)})$.

III. Proposed Method

A. PDE Structure

In this paper, PDEs with two delay values 0 and d are constructed by only basic logic gates. The PDE structure is shown in Fig. 3. A certain delay d is realized by buffer chains connecting buffers in series and the control to select whether or not the delay d is added is realized by a multiplexer.

B. Design Flow for Circuit with PDEs

The design flow for a circuit with PDEs is shown in Fig. 4. At first, logic synthesis is applied to the circuit. Next, in the circuit in gate-level representation, a PDE in gate-level representation is inserted to each path from the clock source to each register. Furthermore, placement, global routing and detailed routing are applied to the circuit with PDEs. In this paper, the yields of the circuits with PDEs are estimated by Monte Carlo simulation which uses the delay information obtained by commercial CAD tools.

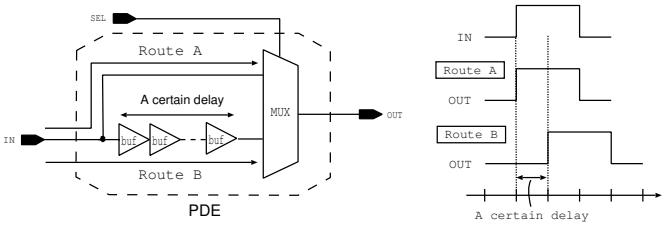

Fig. 3. PDE structure.

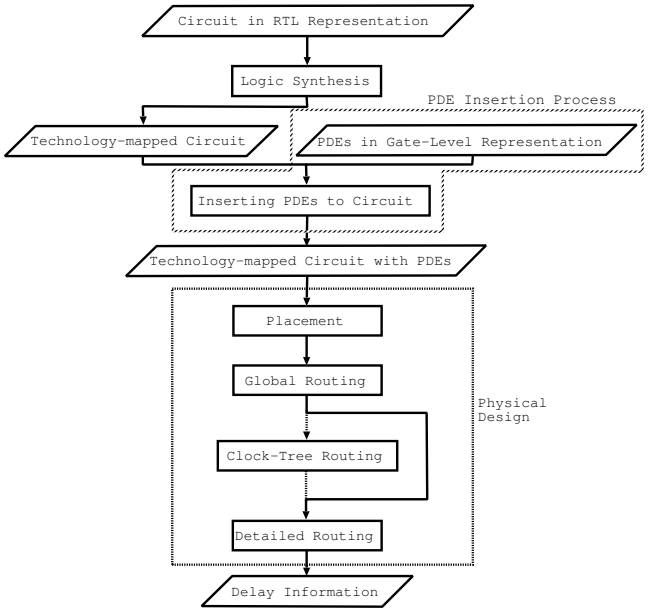

Fig. 4. Design flow for a circuit with PDEs.

C. Proposed PDE Delay Tuning Method

To apply the 2-clustering method directly, it is required to check whether or not there are timing violations by the path-delay-test at all PDE delay tuning patterns for any register pairs with signal propagations. The number of path-delay-tests for this method is $4P + 2S$, where P (S) is the number of signal propagations between different registers (a register).

The flow chart of the proposed PDE delay tuning method is shown in Fig. 5. In the proposed method, the 2-clustering method is firstly applied by using the timing constraint information assumed that all timing constraints are satisfied at all PDE delay tuning patterns such as Table II. Each PDE is tuned to 0 or d by the 2-clustering method. After applying the 2-clustering

TABLE II
INITIALIZED TIMING CONSTRAINT INFORMATION. OK? MEANS THAT THE TIMING CONSTRAINT IS ASSUMED TO BE SATISFIED.

$S^{'}(a)$	$S^{'}(b)$	Timing constraint of (a, b)
0	0	OK?
0	d	OK?
d	0	OK?
d	d	OK?

method, the timing constraints are checked by the path-delay-test for the register pairs assumed that the timing constraint is satisfied. In this paper, we define the number of the path-delay-tests as the number of PDE delay tuning patterns for register pairs assumed the satisfaction of timing constraint and applied the path-delay-test. If the all timing constraints are satisfied, the proposed method is finished as a success. Otherwise, the timing constraint information is updated. Next, the following failure conditions of PDE delay tuning are confirmed.

Failure conditions of the PDE delay tuning

- A timing constraint for signal propagation between a register is violated.
- Timing constraints at all PDE delay tuning patterns for a register pair are violated.
- Timing constraint information for no register pairs is updated.

Even if only one of the failure conditions is satisfied about the timing constraint information, it is guaranteed that the circuit cannot be recovered by PDE delay tuning. Therefore, the proposed method is finished as a failure. Otherwise, the 2-clustering method is applied by using the updated timing constraint information. The proposed method repeats the timing constraint information update and 2-clustering method until the success or failure is determined.

An example of the 2-clustering method to the circuit with PDEs and two register pairs (a, b) and (b, c) is shown in Fig. 6. At first, the timing constraint information for the register pairs (a, b) and (b, c) is assumed that all timing constraints are satisfied, and the 2-clustering method is applied by using the timing constraint information. After applying the 2-clustering method and tuning delays of PDEs, it is found that the timing constraint of the register pair (a, b) is satisfied but that of (b, c) is not. These results are updated to the timing constraint information and the 2-clustering method is applied again. After that, when the timing violation is recovered for the register pair (b, c), tuning PDE delays are succeeded.

D. Comparative PDE Delay Tuning Method

If the delays of the PDEs can be set to 0 or d, the PDE delay tuning is uniquely determined to recover timing violations in the circuit with the PDEs according to the violated constraint. Suppose that signal is propagated from register a to register b. For the hold violation, the delay of PDE inserted to register a is tuned to d and the delay of PDE inserted to register b is tuned to 0 so that the signal propagation $S(a)+D^{'}_{\min}(a, b)$ may be later than the rise of clock in the current period $S(b)$. Similarly, for the setup violation, the delay of PDE inserted to register a is tuned to 0 and the delay of PDE inserted to register b is tuned to d so that the signal propagation $S(a) + D^{'}_{\max}(a, b)$ may be faster than the rise of clock in the next period $S(b) + T$. To compare with the proposed PDE delay tuning method in the experiments, the comparative PDE delay tuning method tunes the delays for each register pair with timing violation at once (Fig. 7).

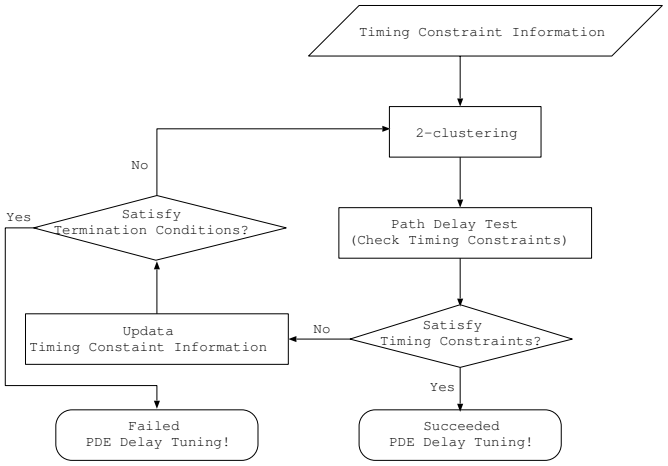

Fig. 5. Flow chart of the proposed PDE delay tuning method.

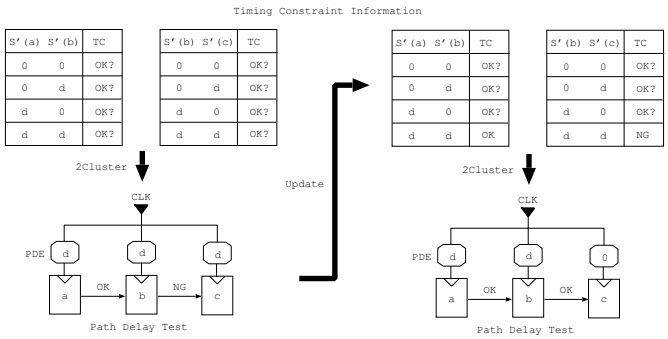

Fig. 6. An example of the proposed PDE delay tuning method.

IV. EXPERIMENTS

To confirm the effects of the proposed method, we apply the proposed method and the comparative method to 26 circuits in ISCAS89 benchmarks in which the time for yield estimation is not long. We compare the yields of the circuits applying between the proposed method and the comparative method. In this paper, we focus on the

Step1: Detect all timing violations and these violated constraints.

Step2: For any timing violations of signal propagation from register a to register b, apply the following procedure.

Step2-1: If the hold violation is found, the delay of PDE inserted to register a is tuned to d.

Step2-2: If the setup violation is found, the delay of PDE inserted to register b is tuned to d.

Step3: Apply path-delay-tests to the register pairs including the registers tuned to d. If no timing violations exist, PDE delay tuning is succeed. Otherwise, PDE delay tuning is failed.

Fig. 7. Comparative PDE delay tuning method.

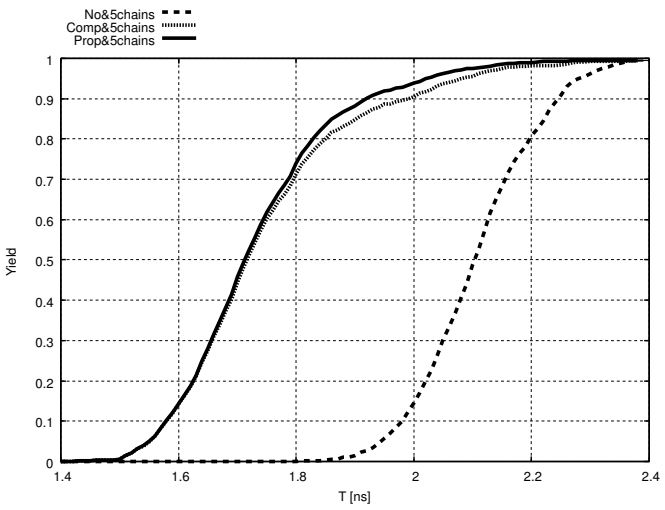

Fig. 8. Relations between clock period and yield in s298.

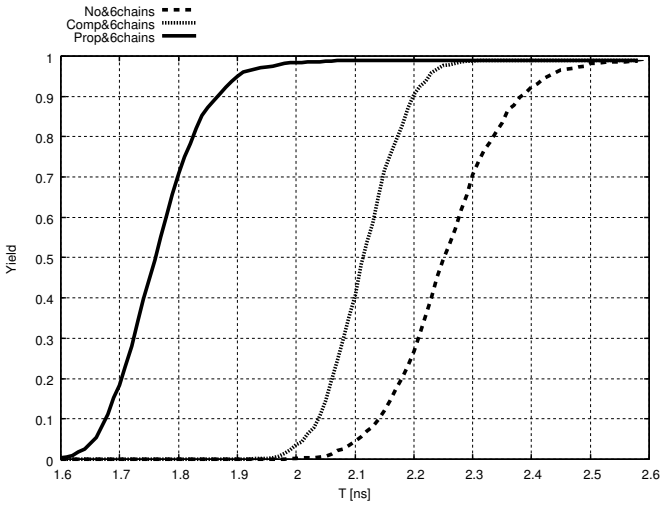

Fig. 9. Relations between clock period and yield in s382.

timing violation. The yield is defined by the ratio of the number of circuits without timing violation to the total number of circuits. The yields are estimated by applying the Monte Calro simulation to the circuits at 1000 times. In Monte Carlo simulation, the delay of each wire or each gate is varied in Gaussian distribution modeled by its minimum and maximum delay in Rohm $0.18\mu m$ library.

To implement the proposed method, the PDEs increasing the number of buffer chains one by one are prepared and are inserted to the circuits according to the design flow for each circuit. Furthermore, the clock period is increased by $0.01ns$ and the yields are estimated by each method at the clock period. Cadence Encounter is used for placement and routing in the design flow.

Fig. 8 and Fig. 9 are experimental results for the two circuits s298 and s382, respectively. The relations between clock period and yield by no PDE delay tuning, the comparative method, and the proposed method are reprensented by no&5chains (no&6chains), comp&5chains (comp&6chains) and prop&5chains (prop&6chains), respectively. In Fig. 8, there is no distinct difference between comp&5chains and prop&5chains because many timing violations in s298 are recovered by the comparative method. In Fig. 9, there is the distinct difference between comp&6chains and prop&6chains because many timing violations in s382 are recovered by the proposed method but not the comparative method.

The experimental results for the 26 circuits are shown in Table III. The minimum clock periods where the yields are over 50% in the proposed method are improved by 13% compared to those in no PDE delay tuning on average. Additionally, the yields in the proposed method are improved compared to those in no PDE delay tuning and the comparative method. Furthermore, the numbers of path-delay-tests for the proposed method are improved by 45.6% compared to those for the 2-clustering method on average.

V. Summary and Conclusions

In this paper, we use PDEs with two delay values and propose a delay tuning method of the PDEs to improve the yield and to reduce the number of tests. Experimental results show that the yields are highly improved with the low path-delay-test cost by using the PDEs with two delay values and applying the proposed PDE delay tuning method.

In the future works, how to decide the number of buffers in the PDE with high yield improvement and decrease the number of the PDEs to keep the yield will be considered.

TABLE III
EXPERIMENTAL RESULTS.

circuit	$\#FF$	P	S	$\#buf$	$T0$	$T1$	$T2$	$\frac{T2}{T0}$	$Y0$	$Y1$	$Y2$	$Y2-Y0$	$\#L$	$\#T$	$\#T_{all}$	$\frac{\#T}{\#T_{all}}$
s27	3	4	3	3	1.07	1.00	1.00	0.93	0.104	0.570	0.579	0.475	7	14.007	22	0.637
s208.1	8	28	8	7	1.62	1.22	1.22	0.75	0	0.541	0.548	0.548	8	108.195	128	0.845
s298	14	56	14	5	2.11	1.72	1.72	0.82	0	0.512	0.525	0.525	9	150.096	252	0.596
s344	15	74	15	5	2.29	2.00	2.00	0.87	0	0.516	0.549	0.549	10	246.486	326	0.756
s349	15	74	15	5	2.29	2.00	2.00	0.87	0	0.547	0.567	0.567	10	249.615	326	0.766
s382	21	131	15	6	2.20	2.12	1.77	0.80	0	0	0.552	0.552	12	321.057	554	0.580
s386	6	30	6	1	2.29	2.25	2.25	0.98	0.363	0.549	0.555	0.192	8	46.015	132	0.349
s400	21	131	15	6	2.26	2.18	1.80	0.80	0	0	0.505	0.505	13	329.366	554	0.595
s420.1	16	120	16	7	2.06	2.05	1.67	0.81	0	0.276	0.561	0.561	11	373.808	512	0.730
s444	21	131	15	7	2.45	2.30	1.99	0.81	0	0	0.524	0.524	13	349.467	554	0.631
s499	22	462	22	2	2.67	2.60	2.57	0.96	0.190	0.402	0.515	0.325	11	711.411	1892	0.376
s510	6	30	6	2	2.26	2.24	2.22	0.98	0.356	0.453	0.521	0.165	9	43.585	132	0.330
s526	21	123	21	4	2.15	1.92	1.92	0.89	0.011	0.514	0.544	0.533	9	315.237	534	0.590
s526n	21	123	21	5	2.12	1.94	1.93	0.91	0.011	0.494	0.546	0.535	14	340.805	534	0.638
s635	32	496	32	9	6.38	6.33	5.76	0.90	0	0.034	0.502	0.502	54	1231.090	2048	0.601
s641	19	100	15	7	4.02	3.92	3.57	0.89	0	0.207	0.521	0.521	9	191.876	430	0.446
s713	19	100	15	10	4.76	4.75	4.06	0.85	0	0	0.506	0.506	8	204.665	430	0.476
s820	5	20	5	3	3.97	3.95	3.95	0.99	0.473	0.500	0.504	0.031	6	25.721	90	0.286
s832	5	20	5	7	4.02	4.01	4.01	1.00	0.475	0.505	0.508	0.033	6	26.704	90	0.297
s838.1	32	496	32	8	2.94	2.94	2.26	0.77	0	0.042	0.530	0.530	17	1234.820	2048	0.603
s938	32	496	32	9	2.91	2.91	2.24	0.77	0	0.056	0.506	0.506	20	1295.180	2048	0.632
s953	26	150	6	3	2.54	2.44	2.35	0.93	0.002	0.089	0.549	0.547	10	330.718	612	0.540
s967	26	150	6	7	2.82	2.73	2.72	0.96	0.216	0.485	0.510	0.294	11	224.976	612	0.368
s1196	12	20	0	7	2.67	2.66	2.09	0.78	0	0.008	0.500	0.500	10	39.231	80	0.490
s1238	12	20	0	7	2.59	2.59	2.02	0.78	0	0.011	0.511	0.511	10	40.385	80	0.505
s1269	37	260	28	6	5.09	4.57	4.57	0.90	0	0.519	0.519	0.519	6	514.552	1096	0.469
ave.								0.87				0.447				0.544

$\#FF$: The number of registers
P: The number of signal propagations between different registers
S: The number of signal propagations between a register
$\#buf$: The number of buffer chains such that the minimum clock period where the yield is over 50% in the proposed method is best
$T0$: The minimum clock period where the yield is over 50% in no PDE delay tuning
$T1$: The minimum clock period where the yield is over 50% in the comparative method
$T2$: The minimum clock period where the yield is over 50% in the proposed method
$Y0$: The yield at $T2$ in no PDE delay tuning
$Y1$: The yield at $T2$ in the comparative method
$Y2$: The yield at $T2$ in the proposed method
$\#L$: The maximum number of loops in the proposed method at $T2$
$\#T$: The average of the numbers of path-delay-tests at $T2$ for the proposed method
$\#T_{all}$: The number of path-delay-tests for the 2-clustering method ($4P + 2S$)

ACKNOWLEDGEMENTS

This work is supported by VLSI Design and Education Center(VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Inc, Rohm Corporation and Toppan Printing Corporation.

REFERENCES

[1] Y. Hashizume, Y. Takashima, and Y. Nakamura, "Post-silicon clock-timing tuning based on statistical estimation," IEICE Trans. Fundamentals, vol.E91-A, no.9, pp.2322–2327, 2008.

[2] M. Kaneko and L. Jian, "Post-silicon skew tuning algorithm utilizing setup and hold timing tests," Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp.125–128, 2012.

[3] Y. Kohira and A. Takahashi, "An optimum 2-clustering method in general-synchronous framework," Proceedings of the 25th Workshop on Circuits and Systems, pp.178–183, 2012 (Japanese).

[4] A. Krstic and K.-T. Cheng, Delay Fault Testing for VLSI Circuits, Kluwer Academic Publishers, 1998.

[5] J.P. Fishburn, "Clock skew optimization," IEEE Tranc. on Computers, vol.39, no.7, pp.945–951, 1990.

[6] S. Tam, U. Desai, and R. Lymaye, "Clock generation and distribution for the third generation itanium processor," Proc. of Symp. on VLSI Circuits, pp.9–12, 2003.

[7] K. Nagaraj and S. Kundu, "An automatic post-silicon clock tuning system for improving system performance based on tester measurements," IEEE International Test Conference, pp.1–8, 2008.

[8] B. Aspvall, M.F. Plass, and R.E. Tarjan, "A linear-time algorithm for testing the truth of certain quantified boolean formulas," Information Processing Letters, vol.8, no.3, pp.121–123, 1979.