IEICE TRANS. FUNDAMENTALS, VOL.E91–A, NO.9 SEPTEMBER 2008

PAPER Special Section on Discrete Mathematics and Its Applications

Post-Silicon Clock-Timing Tuning Based on Statistical Estimation

Yuko HASHIZUME[†], ***Student Member***, **Yasuhiro TAKASHIMA**[†a)], ***and*** **Yuichi NAKAMURA**[††], ***Members***

SUMMARY In deep-submicron technologies, process variations can significantly affect the performance and yield of VLSI chips. As a countermeasure to the variations, post-silicon tuning has been proposed. Deskew, where the clock timing of flip-flops (FFs) is tuned by inserted programmable delay elements (PDEs) into the clock tree, is classified into this method. We propose a novel deskew method that decides the delay values of the elements by measuring a small amount of FFs' clock timing and presuming the rest of FFs' clock timings based on a statistical model. In addition, our proposed method can determine the discrete PDE delay value because the rewriting constraint satisfies the condition of total unimodularity.

key words: *post-silicon clock-timing tuning, deskew, programmable delay element (PDE), linear programming, totally unimodular*

1. Introduction

In deep-submicron technologies, especially under 45 or 35 nm, process variations may severely affect the performance and yield of VLSI chips [1]. As countermeasures, several methods have been proposed. These include *pre-silicon design* methods, such as SSTA [2], [3] or asynchronous circuit [4] which approach the problem from the stand point of the design or architecture of a chip, and *post-silicon tuning* which is executed after the fabrication of chips [5]–[8]. Between these methods, post-silicon tuning is more effective and practical because the tuned values are decided by observing the condition of the fabricated chip. Deskew, where delay values of the clock tree are tuned to satisfy the chip specifications, is one of the best methods for the post-silicon tuning (Fig. 1). Several approaches to deskew have been proposed [5]–[8]. However, all of them require measurement of many or all clock timings of all flip-flops (FFs). Thus, the test cost seems to be high.

In this paper, we propose a novel deskew method with a small amount of timing measurements. Our method consists of the estimation of the clock timings of FFs and the decision of the delay values of the clock tree. The number of measurements is greatly reduced because we assume that clock skew occurs on each edge of the H-tree. The estimation is based on statistical model. In addition, our proposed method can determine the discrete PDE delay value with Linear Programming (LP). Generally, this problem can be solved by Integer Linear Programming (ILP) but it is very time consuming. We propose a method that uses LP by rewriting the constraint as totally unimodular [9].

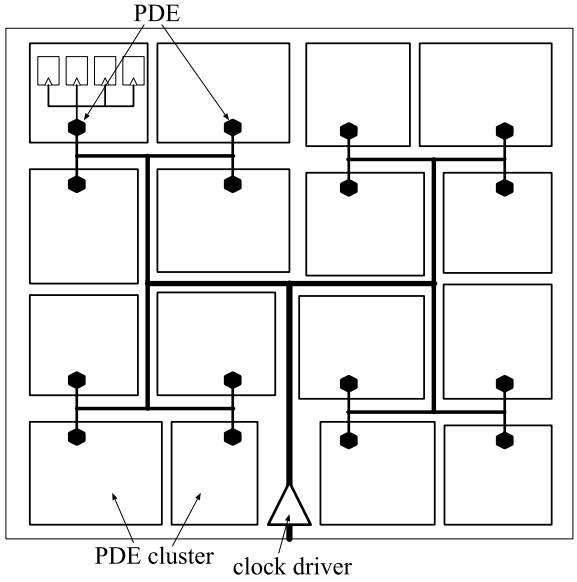

Fig. 1 Model of clock distribution.

Our experiments show that the yield after applying deskew is improved from 30.0% to 99.4% in a sample with measured arrival times of 0.3% of total FFs and tuning delays of 16 programmable delay elements (PDEs). This method can extremely reduce the test-cost compared with all the conventional methods, since the number of measured FFs is very small. Other experiments show that our proposed method can tune the delays of PDEs under wider variations.

The rest of this paper is organized as follow: Sect. 2 covers previous works on deskew. Section 3 describes our proposed deskew method and discussion about the experimental results. Finally, Sect. 4 summarizes the results and mentions future work concludes.

2. Previous Works

For deskew, PDEs are prefabricated with delay parameters left free on the clock distribution tree (See Fig. 1). PDEs can be constructed using delay generators and registers to control which generators are active [5], [6]. The PDEs can control the arrival times of FFs whose clock paths run the

Manuscript received December 18, 2007.

[†]The authors are with the Faculty of Environmental Engineering, the University of Kitakyushu, Kitakyushu-shi, 808-0135 Japan.

[††]The author is with the System IP Core Research Lab., NEC, Kawasaki-shi, 211-8666 Japan.

a) E-mail: takasima@env.kitakyu-u.ac.jp

DOI: 10.1093/ietfec/e91–a.9.2322

PDEs. The clock deskew problem is to determine the delay of each PDE to satisfy the chip specification by means of the on-chip timing analysis after fabrication.

Tam et al. implemented [5] a clock deskew method by clustering FFs, tuning PDE delay with fuses in the post-process test, and tuning PDE delay with switches on the scan-chain. This framework, especially its use of fuses and scan-chain switches, is suitable for the current chip design flow and is used in the Itanium2. The units of PDE delay value for fuses and scan-chains are 35 ps and 9 ps, respectively. The method needs genetic Algorithm (GA) in order to apply the measurement of the clock delay and tuning. In Tam et al's GA, the calculation of fitness is based on the post-process test such as the ratio of successes to total trials. Thus, it requires considerable time. Murakawa et al. [6] and Susa et al. [8] proposed a method similar to Tam et al.'s. The methods try to reduce the number of tests by GA. However, it is too expensive to evaluate the clock skew in real chips.

Tsai et al. [7] proposes a method that calculates the slack of each FF-FF path, inserts PDEs into the clock inputs of FFs with small slack, and determines PDE delay using LP. This method is based on the fact that the effect of process variations can cause FFs with small slack to fail. It is a theoretically valid but an impractical approach because PDEs are inserted at every FF with severe timing, so the number of PDEs may become quite large.

Overall, the test cost of all conventional methods is very high, since they have to observe the all FFs. We introduce a novel clock deskew system that models the process variations and estimates arrival clock times of FFs, to achieve the low-test cost deskew method.

3. Proposed Deskew Methods

In this paper, we assumed that chip defects are caused only by variations in clock skew. Furthermore, H-tree clock distribution is assumed to model the problem strictly. However, the method can be easily extended to general clock tree systems (See Fig. 2). For FF_i, let t_i and ε_i be the arrival clock time determined in the course of design and the amount of clock skew due to process variations, respectively. The sum of t_i and ε_i is the clock timing of the *post-silicon* of FF_i, denoted as T_i. Let $\Delta_{ii'}$ be the path delay between FF_i and $FF_{i'}$.

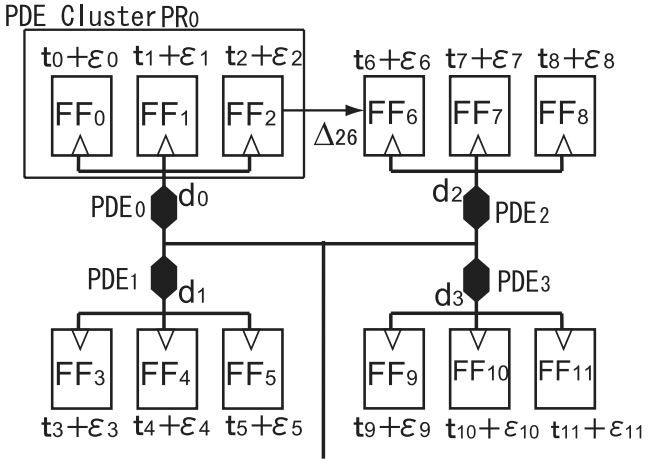

Fig. 2 H-tree clock distribution with PDEs.

Here, we assume that all PDEs are inserted into the same level edges of the H-tree. The number of PDEs can be defined by the level of the H-tree which is the number of edges from the source. For example, if the level is three, the number of PDEs is $2^3 = 8$. Figure 1 shows an example of the case of level 4. An FF belongs to only one *PDE cluster*. If FF_i belongs to the cluster of PDE_j, denoted as PR_j, the arrival clock time of FF_i is delayed by the delay value d_j of PDE_j. Without loss of generality, we assume $d_j \geq 0$. Figure 2 shows a one-layer H-tree with four PDEs and 12 FFs. In this example, FF_1 has the arrival clock time t_1 and the clock skew ε_1. The path between FF_2 and FF_6 has the path delay Δ_{26}. The cluster of PDE_0, denoted as PR_0, consists of FF_0, FF_1 and FF_2, whose arrival clock times are all delayed d_0.

We propose a deskew method that calculates the delay of each PDE using a feasibility check of LP. We utilize a *path-delay* constraint, which is intended to remove timing violations in the setup and hold constraints with PDE delays.

Our final target is the proposal of the deskew method which can achieve large yield by small measured FFs. To discuss the final target, we describe our deskew method using the model where the arrival clock time of all FFs are measured as the preparation of the final target. Then, we extend the method to handle the model where the arrival clock times of some FFs are measured.

1) Problem Definition for Model of All Measured Arrival Clock Times (AMA)

Input: Set of FFs, arrival clock times of all FFs, and set of paths with their path delay
Output: Each PDE delay with the discrete value
Constraints: Hold constraint and setup constraint

2) Problem Definition for Model of Some Measured Arrival Clock Times (SMA)

Input: Set of FFs, arrival clock times of some FFs, set of paths with their path delay and the variation of the delay of each H-tree edges
Output: Each PDE delay with the discrete value
Constraint: Hold constraint and setup constraint

In the rest of this section, we introduce the formulation of these problems.

3.1 Problem AMA

3.1.1 Formulation

We formulate the problem AMA as a feasibility check of LP with the path delay constraint. The constraint of LP consists of the setup-time constraint and the hold-time constraint for the path between $FF_i \in PR_j$ and $FF_{i'} \in PR_{j'}$. That is

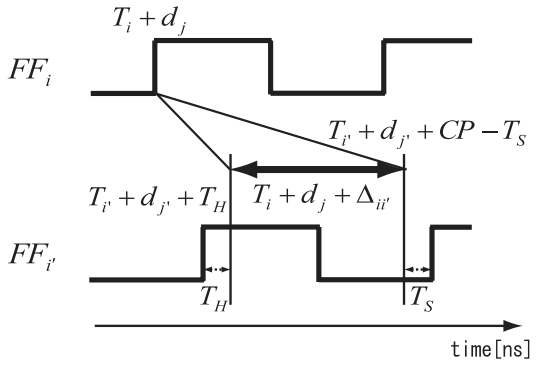

Fig. 3 Timing chart under the path-delay constraint.

$$\left(T_i + d_j\right) + \Delta_{ii'} \leq \left(T_{i'} + d_{j'}\right) + CP - T_S, \text{ and}$$
$$\left(T_i + d_j\right) + \Delta_{ii'} \geq \left(T_{i'} + d_{j'}\right) + T_H$$

respectively, where T_S is the setup time, T_H is the hold time, and CP is the clock period. These constraints are merged into

$$T_H - \Delta_{ii'} \leq \left(T_i + d_j\right) - \left(T'_i + d_{j'}\right) \leq CP - T_S - \Delta_{ii'}, \quad (1)$$

The constraint (1) means that the signal from FF_i has to arrive within the period indicated by the solid arrow in Fig. 3. An example of the PDE delay effect is shown in Fig. 4. Figures (a) and (b) show clock timing without and with PDE delay and their timing charts, respectively. In these figures, the clock period and the path-delay between FF A and FF B are 2 ns and 1.8 ns, respectively. Both the setup time and the hold time are 0.1 ns. On the left side of these figures, the clock timing of FF A and FF B are 1 ns and 0.5 ns, respectively. Thus, the signal from FF A arrives at FF B at 2.8 ns. However, since the signal arrives after the next clock timing of FF B, FF B cannot receive it. To adjust the clock skew, the PDE delay of FF B is set to 0.5 ns, then the next clock timing of FF B becomes 3 ns. As a result, FF B can receive the signal from FF A to FF B.

In practical use, PDE delay values are not continuous but discrete. Thus, we need to transform constraint (1) into an integral constraint. We assume that the PDE delay value is the integral multiplication of the delay value of one delay element p and rewrite $d_j = px_j$. Thus, Eq. (1) becomes

$$A \leq px_j - px_{j'} \leq B, \quad (2)$$
$$\text{where } \begin{cases} A = T_H - (T_i - T_{i'} + \Delta_{ii'}) \\ B = CP - T_S - (T_i - T_{i'} + \Delta_{ii'}) \end{cases}.$$

All sides divided by p is Eq. (3).

$$A/p \leq x_j - x_{j'} \leq B/p \quad (3)$$

Equation (3) can be rewritten as (4) because the variables x_i and $x_{i'}$ are integers.

$$\lceil A/p \rceil \leq x_j - x_{j'} \leq \lfloor B/p \rfloor \quad (4)$$

In each constraint, there are only two variables such that one coefficient is 1 and the other coefficient is -1. Thus, the coefficient matrix is totally unimodular. Furthermore, the constraint vector is integral. Therefore, we obtain an integral solution by not ILP but LP [9].

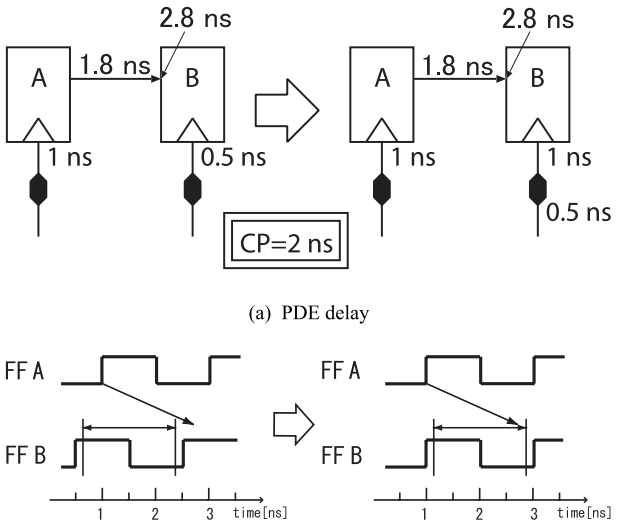

Fig. 4 Example of PDE delay value under the path-delay constraint.

3.1.2 Experimental Results

We conducted an experiment to confirm the ability of our LP-based deskew method for AMA. The computation environment was as follows: the CPU was an AMD Opteron 265 with 3 GB of memory and CentOS 4.3 OS. We used the commercial LP calculation system CPLEX 10.0 [10], as the LP solver. Since the computation times are always several seconds, they are not listed in the tables.
The benchmarks were

1) Number of layers in the H-tree: 5.
2) Number of FFs: 20480.
3) Number of paths per FF: 5.
4) Number of PDEs: between 2 and 1024.
5) Clock Period is 2 ns, i.e., the clock frequency is 500 MHz.
6) Hold-time T_H and setup-time T_S: 0.1 ns.
7) Path delay Δ: a Gaussian distribution with $(\mu_\Delta, \sigma_\Delta)$ = (1.1, 0.16). In practical, each path delay could be given since path delay can be calculated by STA.
8) Clock skew occurs on each edges of the H-tree. We assume that every clock skews of edges are independent. Thus, the clock skew ε of each FF is the sum of the skew of edges.
9) Variation in each edge holds a Gaussian distribution, with an average of 0 over all edges, and a deviation proportional to the length of the edge, is 0.067 ns on the edge connected to the root. The standard deviation from clock source to FF is 0.119 ns. According to [11], this value is appropriate.
10) The resolution of PDE delay value is assumed to be 75 ps × 3 bits.

Table 1 The nondefective chip rate of all timing measured model (Nondefective rate before applying deskew: 30.0%).

Number of PDE	Nondefective rate
2	58.4%
4	93.0%
8	98.2%
16	99.6%
32	100.0%
64	100.0%
128	100.0%
256	100.0%
512	100.0%
1024	100.0%

We applied our method to 500 randomly generated benchmarks in the conditions described above. Assuming that defects are caused only by clock skew, there are 150 nondefective benchmarks (30.0%) before deskew is applied to the 500 benchmarks. The results are shown in Table 1.

From the result, as the number of PDEs increases, the nondefective chip rates after applying deskew increases. In this case, while the nondefective chip rate before applying deskew is 30.0%, the nondefective chip rates after applying deskew increase to 98.2% with 8 PDEs. We thus conclude that our method is effective for improving the yield, when timings of all FFs can be measured.

3.2 Problem SMA

The discussions in the previous sub-section indicate the deskew method would improve the yield, if the defective chips are only caused by the variation of the clock skew. However, since the proposed method for AMA needs the observation of the timings of all FFs, the effectiveness of the method for the practical use is the same as all the conventional methods with the expensive test-cost. Thus, we propose the method for the problem for some measured arrival clock times model, called SMA.

3.2.1 Estimation

To reduce the test-cost and the number of the observed FFs, we employ a following estimation method. First, we measure some arrival clock times of FFs. Then, we estimate the arrival clock times of the rest FFs from the measured arrival clock times. When we estimate FF_p's clock timing from measured FF_o's clock timing T_o, let Vr be the nearest point to leaves among common points of FF_i and FF_j in the clock tree (Fig. 5). Let $N(\mu_{r-o}, \sigma^2_{r-o})$ and $N(\mu_{r-p}, \sigma^2_{r-p})$ be the variation from r to FF_o and FF_p, respectively. Since the variation from r to FF_o is $N(\mu_{r-o}, \sigma^2_{r-o})$, the clock arrival time at r is calculated as $N(T_o - \mu_{r-o}, \sigma^2_{r-o})$. Then, the estimated clock arrival time at FF_p is $N(T_o\mu_{r-o} + \mu_{r-p}, \sigma^2_{r-o} + \sigma^2_{r-p})$ since the variation from r to FF_p is $N(\mu_{r-p}, \sigma^2_{r-p})$.

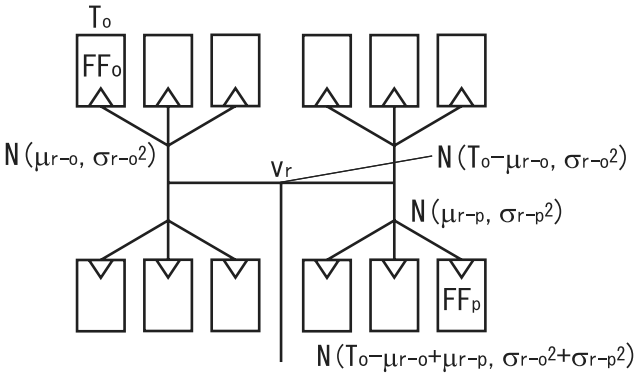

Fig. 5 Presumption model.

3.2.2 Formulation

We also formulate this problem as a feasibility check of LP with the path-delay constraint. Since the estimated clock timing is obtained as a Gaussian distribution, difference between the estimated clock timings of a pair of FFs is also a Gaussian distribution, i.e. $N\left(\mu_i - \mu_{i'}, \sigma_i^2 + \sigma_{i'}^2\right)$, Gaussian distribution, three standard deviations from the mean account for about 99.7% of the set. Using this property, we decide that the range of $N\left(\mu_i - \mu_{i'}, \sigma_i^2 + \sigma_{i'}^2\right)$ is

$$\left[\mu_i - \mu_{i'} - 3\sqrt{\sigma_i^2 + \sigma_{i'}^2}, \mu_i - \mu_{i'} + 3\sqrt{\sigma_i^2 + \sigma_{i'}^2}\right].$$

We rewrite constraint (1) for the path between $FF_i \in PR_j$ and $FF_{i'} \in PR_{j'}$ as

$$\begin{cases} T_H - \Delta_{ii'} - \left(\mu_i - \mu_{i'} - 3\sqrt{\sigma_i^2 + \sigma_{i'}^2}\right) \le d_j - d_{j'}, \\ d_j - d_{j'} \le CP - T_S - \Delta_{ii'} - \left(\mu_i - \mu_{i'} + 3\sqrt{\sigma_i^2 + \sigma_{i'}^2}\right). \end{cases}$$

To handle the discrete PDE delay value, the constraints are transformed as

$$\lceil A/p \rceil \le x_j - x_{j'} \le \lfloor B/p \rfloor,$$

where

$$\begin{cases} A = T_H - \Delta_{ii'} - \left(\mu_i - \mu_{i'} - 3\sqrt{\sigma_i^2 + \sigma_{i'}^2}\right) \\ B = CP - T_S - \Delta_{ii'} - \left(\mu_i - \mu_{i'} + 3\sqrt{\sigma_i^2 + \sigma_{i'}^2}\right) \end{cases}.$$

3.2.3 Experimental Results

We conducted an experiment to confirm the effectiveness of our proposed deskew method. The computation environment was the same as that for the previous experiment for AMA. Again, the computation times are not shown. We apply to all benchmarks in spite of their nondefectiveness before applying our deskew method.

The experimental results are shown in Table 2. The number of measured FFs is from 4 (a measured FF for 256 leaves of the clock tree) to 1024 (a measured FF per leaf) for

Table 2 The nondefective chip rate of some measured timing model (Nondefective chip rate before applying deskew: 30.0%).

#observed FFs	1024	512	256	128	64	32	16	8	4
#PDE	ratio	ratio	ratio	Ratio	ratio	ratio	ratio	Ratio	Ratio
2	55.0%	51.2%	47.0%	40.6%	37.2%	30.8%	30.2%	30.0%	30.0%
4	90.6%	88.8%	85.2%	79.0%	70.0%	43.6%	34.6%	30.0%	30.0%
8	97.8%	97.2%	96.4%	95.2%	93.4%	73.6%	50.0%	30.0%	30.0%
16	99.6%	99.6%	99.2%	99.4%	99.4%	97.0%	87.2%	30.2%	30.0%
32	100.0%	100.0%	99.8%	99.8%	100.0%	99.8%	94.4%	32.0%	30.0%
64	100.0%	100.0%	99.8%	99.8%	100.0%	99.8%	98.6%	45.8%	30.0%
128	100.0%	100.0%	99.8%	99.8%	99.8%	99.8%	98.8%	79.6%	30.0%
256	100.0%	100.0%	99.8%	99.8%	99.6%	99.6%	99.6%	93.4%	31.2%
512	100.0%	100.0%	99.8%	99.8%	99.6%	99.4%	99.2%	94.2%	67.8%
1024	100.0%	99.8%	99.8%	99.8%	99.4%	99.4%	99.2%	94.4%	86.8%

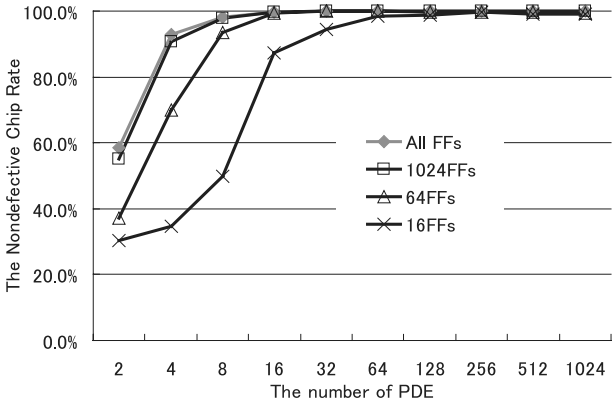

Fig. 6 Comparisons among the number of measured FFs.

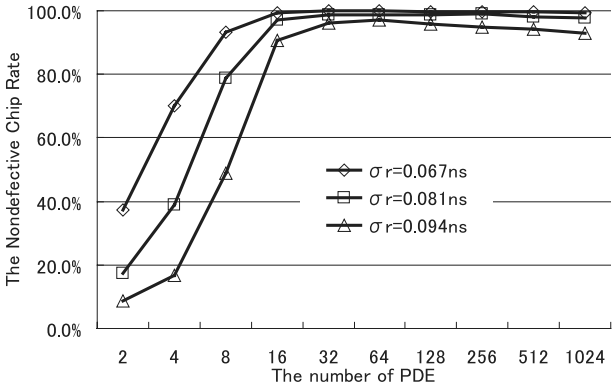

Fig. 7 Comparisons of several variations.

the same benchmarks as that of AMA.

From the results, in general, as the number of PDEs increases, the nondefective chips rate increases. In addition, as the number of measured FFs increases, the nondefective chips rate increases. The exceptions are caused by the error between the estimated and real values of the arrival times of the FFs. However, the ratio of exception is quite small.

3.3 Comparison of Experimental Results

Figure 6 shows the relationship between the number of PDE and the nondefective chip rate over the different number of measured FFs. We select the cases of measured all FFs, 1024FFs, 64FFs, and 16FFs. The ratio of measured all FFs and measured 1024 FFs are almost same. The nondefective chips are increased from 30.0% to 99.4% after applying our proposed method with measured 64 FFs (0.3% of all FFs) and tuning 16 PDEs. This is comparable to the nondefective chip rate after applying deskew with measured all FFs (20480 FFs), which is 99.6%. Even the ratio over measured 16 FFs is comparable to the ratio of measured all FFs with more than 16 PDEs. Therefore, the results confirm that our proposed method can significantly decrease the number of FFs measured the clock timings with remaining its effectiveness.

Figure 7 shows that relationship between the number of PDE and the nondefective chip rate over the different standard deviation on the edge connected to the root with the 64 FFs measuring. We select the cases of 0.094, 0.081 and 0.067 ns. For each case, the nondefective chips without applying deskew are 7.8%, 14.6% and 30.0%, respectively. As the variation is larger, the number of nondefective chips after applying deskew becomes lower. However, for all variation, the nondefective chips improve to about 90% after applying deskew with 16 PDEs. Thus, the results confirm that our proposed method is also effective to larger process variations. This can relax the design constraint. As a result, the turn-around-time (TAT) may become much shorter.

Figure 7 also shows that the number of nondefective chips after applying deskew with more than 16 PDEs becomes lower as the number of inserting PDEs is larger. In this paper, we estimated the clock timing using the Gaussian distribution, where its mean is μ and its standard deviation is σ. The distribution has the property that about 99.7% phenomenon exists between $\mu-3\sigma$ and $\mu+3\sigma$. When the difference between the clock timings of a pair of FFs is out of the estimated range for the constraint of LP, it causes inaccuracy. As the number of constraints increases as according to

the number of inserting PDEs, the number of the departures also increases. Thus the nondefective chip rate decreases. However, since the nondefective chip rates with 16 PDEs are more than 90%, we do not have any advantage using much PDEs.

4. Conclusion

We proposed a novel deskew method that can decide delay values from measuring a small amount of FFs' clock timings and estimating the rest of FF's clock timings based on statistical model. The PDE delays are determined by linear programming. Our method was able to determine the discrete PDE delay value because the rewriting constraint satisfies the condition of total unimodularity. Our experiments showed that the yield after applying deskew was improved from 30.0% to 99.4% for a sample when the arrival times for 0.3% of total FFs are measured and the delays of 16 programmable delay elements are tuned. The results confirmed that our proposed method can significantly decrease the number of FFs measured the clock timings, while it remains effectiveness of the deskew method with small number of PDEs. Furthermore, we tested under the several variations. The results confirmed that our proposed method is also effective to larger process variations. This would lead that the design constraint is relaxed and TAT becomes shorter, resultantly.

In the future work, we plan to extend our method to more realistic situations, such as applying clock trees other than the H-tree. Ultimately, we aim to design a chip that autonomously obtains PDE delays.

References

[1] L. Scheffer, S. Nassif, A. Strojwas, B. Koenemann, and N.S. Nagaraj, "Design for manufacturing in the sub-64 nm era," Tutorial on DAC 2005, 2005.
[2] J. Singh and S. Sapatnekar, "Statistical timing analysis with correlated non-Gaussian parameters using independent component analysis," Proc. DAC 2006, pp.155–160, 2006.
[3] W.S. Wang, V. Kreinovich, and M. Orshansky, "Statistical timing base on incomplete probabilistic descriptions of parameter uncertainty," Proc. DAC 2006, pp.161–166, 2006.
[4] CHAINworks, http://www.silistix.com/
[5] S. Tam, U. Desai, and R. Limaye, "Clock generation and distribution for the third generation itanium processor," Proc. Symp. on VLSI Circuits, pp.9–12, 2003.
[6] M. Murakawa, E. Takahashi, T. Susa, and T. Higuchi, "Post-fabrication clock timing adjustment for digital LSIs with genetic algorithms ensuring timing margins," MIRAI Project, 2004.
[7] J.L. Tsai, D.H. Baik, C.C.P. Chen, and K.K. Saluja, "A yield improvement methodology using pre- and post-silicon statistical clock scheduling," Proc. SMC, pp.3670–3674, 2004.
[8] T. Susa, M. Murakawa, E. Takahashi, T. Furuya, T. Higuchi, S. Furuichi, Y. Ueda, and A. Wada, "Speed enhancement technique for the post-fabrication clock-timing adjustment of digital LSIs," Proc. SASIMI, pp.166–173, 2007.
[9] W.J. Cook, W.H. Cunningham, W.R. Pullayblank, and A. Schrijver, Combinatorial Optimization, John Willy & Sons, 1998.
[10] http://www.ilog.com/produtcs/cplex
[11] M. Hashimoto, T. Yamamoto, and H. Onodera, "Statistical analysis of clock skew variation in H-tree structure," IEICE Trans. Fundamentals, vol.E88-A, no.12, pp.3375–3381, Dec. 2005.

Yuko Hashizume received her B.E. degree in information engineering from the University of Kitakyushu in 2006, she has been a master course student in the University of Kitakyushu.

Yasuhiro Takashima received his B.E., M.E. and D.E. degree in electrical and electronic engineering from the Tokyo Intitute of Technology in 1993, 1995, and 1998, respectively. He was a research associate in the School of Information Science at the Japan Advanced Institute of Science and Technology from 1998 to 2003. He was with the Kitakyushu Foundation for the Advancement of Industry, Science and Technology as an invited researcher from 2003 to 2005. Since 2005, he has been an associate professor of the University of Kitakyushu. His research interests include combinatorial algorithms for VLSI design.

Yuichi Nakamura received his B.E. degree in information engineering and M.E. degree in electrical engineering from the Tokyo Institute of Technology in 1986 and 1988, respectively. He received D.E. degree from the Graduate School of Information, Production, and Systems, Waseda University, in 2007. He is currently a principal researcher at System IP Core Labs., NEC Corp. His research interests include the System LSI design and their algorithms.