

Statistical Static Timing Analysis Technology

● Izumi Nitta ● Toshiyuki Shibuya ● Katsumi Homma

(Manuscript received April 19, 2007)

With CMOS technology scaling down to the nanometer realm, process variations have been increased. In particular, the increase of delay variations has seriously affected the design periods and timing yields. To estimate more accurately these delay variations, statistical static timing analysis (SSTA), which considers delay variations statistically, has been proposed. SSTA is expected to shorten the design turnaround time (TAT) and predict the timing yields. Research on practical applications of SSTA has already been conducted at Fujitsu Laboratories. We have developed SSTA tools for use in designs for processors and application specific integrated circuits (ASICs) in cooperation with Fujitsu and Fujitsu VLSI. This paper describes the delay variations and basic SSTA techniques and introduces SSTA applications to Fujitsu's processor and ASIC design flows.

1. Introduction

The scaling down of process technologies has increased the process variations. Among them, delay variations that impact the frequency performance of circuits have increased and now seriously affect the design turnaround time (TAT) and timing yields, which are the ratio of chips that achieve the target frequency.

In application specific integrated circuit (ASIC) designs, sufficient margins for delay variations are required to achieve the target frequency at the expected yield. In process technologies above 90 nm, the margins for delay variations are small enough and their impact on design can be eliminated. However, at 90 nm and below, the increased delay variations enlarge the margins in circuit design. This results in overestimations of circuit delay and makes design work difficult. Because of the overestimation, designers have to tune a design iteratively to achieve the target frequency performance, which lengthens the design TAT.

In high-performance microprocessor designs, excessive margins make it difficult to achieve the target performance. Therefore, nominal values are used at the design phase. After a batch of chips has been fabricated, the frequency selection process sorts them into several ranks according to the measured maximum frequency. Then, the chips are priced and shipped based on their ranks. In such designs, it is important to predict the timing yield at the design phase because, at this phase, we can consider the trade-offs between chip performance and yield.

Statistical static timing analysis (SSTA), which analyzes circuit delays statistically by considering delay variations, is attracting our interest as a solution to the above issues.

Fujitsu Laboratories started researching SSTA in 2003, and has applied SSTA technologies to processor and ASIC designs in cooperation with Fujitsu Limited and Fujitsu VLSI Limited.

This paper describes the delay variations caused by process variations and introduces the

basic SSTA techniques. It then reports on the SSTA research conducted by Fujitsu Laboratories and outlines examples of SSTA application to microprocessor and ASIC designs.

2. What are the delay variations?

The delay variations are due to various types of process variations. **Figure 1** shows examples of the major process variations. Figure 1 (a) shows the threshold voltage variations in transistors caused by density variations of impurities in the transistor material. The upper part of Figure 1 (b) shows pattern variations that occurred in the lithography process. These variations are caused by light interference between adjacent patterns. The lower part of Figure 1 (b) shows silicon-surface flatness variations caused by layout pattern density variations during the chemical mechanical planarization (CMP) process.

The variations described above can be classified as random variations or systematic variations. Random variations occur without regard to the locations and patterns of transistors within a chip; the variation in transistor threshold voltage, for example, is a random variation. Systematic variations, on the other hand, are related to the locations and patterns; some examples of these variations are exposure pattern variations and silicon-surface flatness variations.

The variations between the elements in the same chip are called within-die (WID) variations, and the variations between chips in the same wafer or in different wafers are called die-to-die (D2D) variations [Figure 1 (c)].

As technology advances, these process variations have been improved on the manufacturing side. For example, variations of exposure patterns have been improved by using better

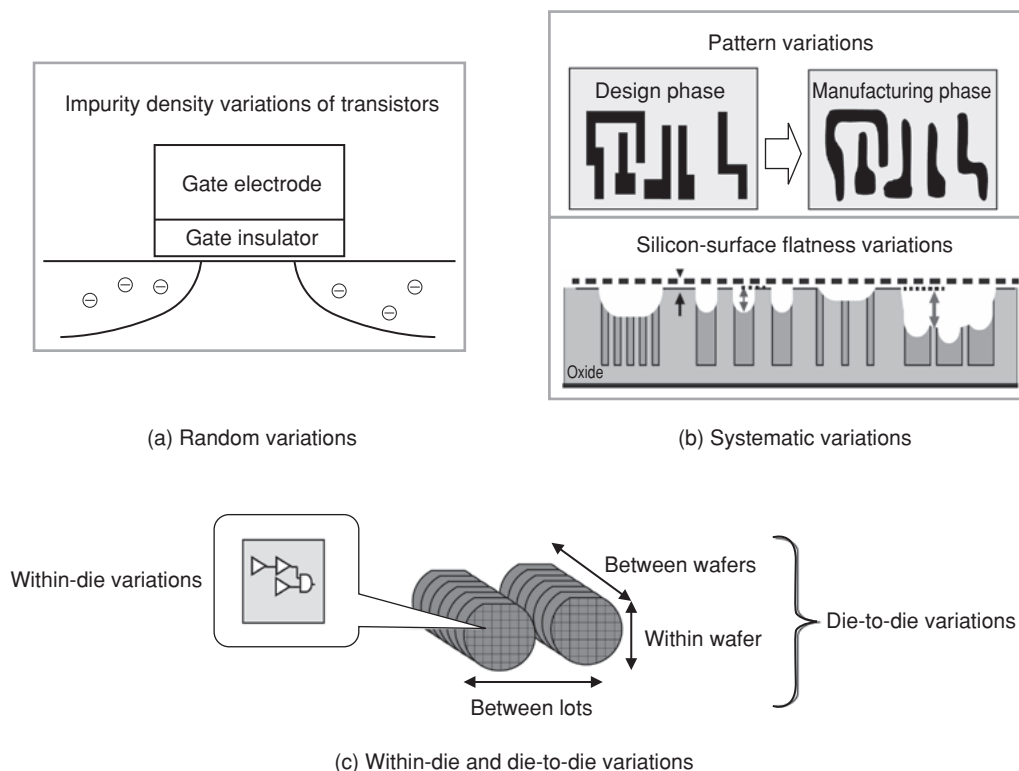


Figure 1
Examples of LSI process variations.

lithography exposure techniques. On the design side, one of the main techniques for assuring the expected yield has been to provide sufficient design margins. In nanometer technologies, because it will not be sufficient to reduce the process variations in the manufacturing phase, it will also be necessary to use “design for manufacturing” (DFM) methodologies that take process variations into consideration on the design side. As one of the DFM techniques, SSTA estimates circuit delay and frequency performance by considering delay variations statistically.

3. Basic techniques of SSTA

This section describes the basic SSTA techniques.

3.1 What is SSTA?

In the conventional design flow, static timing analysis (STA) is used to estimate the circuit delay and maximum frequency. To assure sufficient yield, STA analyzes corner cases, in which all the factors of the delay variations are at the worst-case or best-case corner values. In actual chips, however, the probability of all factors being at the corner values is very low. Therefore, STA estimates a delay that rarely occurs in actual processes; that is, it analyzes using excessive margins for delay variations.

The main concept of SSTA is to statistically consider the random variations of WID in order to analyze circuit delay more accurately. The simplest method of statistical calculation is Monte Carlo simulation. However, the computation time of this method increases drastically according to the number of variation factors and the circuit scale. For this reason, Monte Carlo simulation is not practical for analyzing actual designs. Therefore, many researchers have studied the basic SSTA method, and many of their results have been reported, starting from about 2000.^{1),2)} The basic SSTA method defines the random variations of the delay as random

variables and calculates the probability density function (PDF) of circuit delay. The method saves computation time while producing results equivalent to those of Monte Carlo simulation.

3.2 Basic SSTA operations

In SSTA processing, a circuit is expressed by a graph that represents the gates and interconnects as nodes. Traversing the graph, the PDF of the delay in each node is calculated using the statistical sum and max operations with the delay variations of the gates and interconnects as inputs.

Figure 2 shows the basic operations for two circuits with the interconnect delays ignored to simplify the calculations. Figure 2 (a) shows a circuit with two gates connected in series, and Figure 2 (b) shows a circuit in which two signals converge at the output pin of a gate. The delay of the series circuit in Figure 2 (a) is calculated using a statistical sum operation. With the delay PDF of these gates denoted as f_1 and f_2 , the delay PDF at the end point is the statistical sum of f_1 and f_2 . The statistical sum is calculated using convolution integration. However, when f_1 and f_2 have normal distributions, a simple formula can be used. In Figure 2 (a), when f_1 has a normal distribution with m_1 (average) and s_1 (3σ), and f_2 has a normal distribution with m_2 (average) and s_2 (3σ), f has a normal distribution with $m_1 + m_2$ (average) and $\sqrt{s_1^2 + s_2^2}$ (3σ). The value equivalent to 3σ of this distribution is $m_1 + m_2 + \sqrt{s_1^2 + s_2^2}$. In this case, when this delay is calculated using the conventional STA method, the delay is the sum of the worst-case values. When the worst-case values are equivalent to 3σ , the delay is $m_1 + m_2 + s_1 + s_2$. Therefore, the delay of the series circuit calculated with SSTA is smaller than that calculated with STA.

The output delay of the multiple-input gate in Figure 2 (b) is calculated using a statistical max operation. It is generally difficult to calculate an accurate value for the statistical max. However, when the two random variables for f_1

and f_2 are independent of each other, an accurate solution can be obtained. Conversely, when f_1 and f_2 correlate with each other, it is difficult to obtain an accurate solution of the statistical max operation and only an approximation is possible by using the upper-bound or lower-bound of the PDF calculation or by using the moment matching technique.¹⁾ When f_1 and f_2 are independent, the result of the statistical max operation is known to have an upper bound and the value equivalent to 3σ is greater than the delay calculated with STA.

3.3 Path-based and block-based SSTA

The delay of an entire circuit can be analyzed by using the basic calculations shown above while traversing the graph. There are two types of graph analysis methods: path-based SSTA and block-based SSTA.

Figure 3 (a) shows the path-based SSTA. In this method, the delay PDF of each path is calculated individually, traversing from the source to the sink of the path. The advantage of

this method is that it accurately calculates the delay PDF of each path because it does not use statistical max operations to analyze sequential paths. Also, it can consider the correlations between paths easily. However, its computation time drastically increases with the circuit scale because the number of paths increases exponentially with the circuit scale.

Figure 3 (b) shows the block-based SSTA. In this method, all paths are analyzed simultaneously by traversing the graph, with the delay PDF of the entire circuit also being calculated at the end of the traversal. The advantage of this method is that it requires less computation time than the path-based method because more than one path can be analyzed simultaneously. However, the correlations between paths must be considered for the statistical max operation when multiple paths converge at a node. Therefore, there are trade-offs between accuracy and computation time.

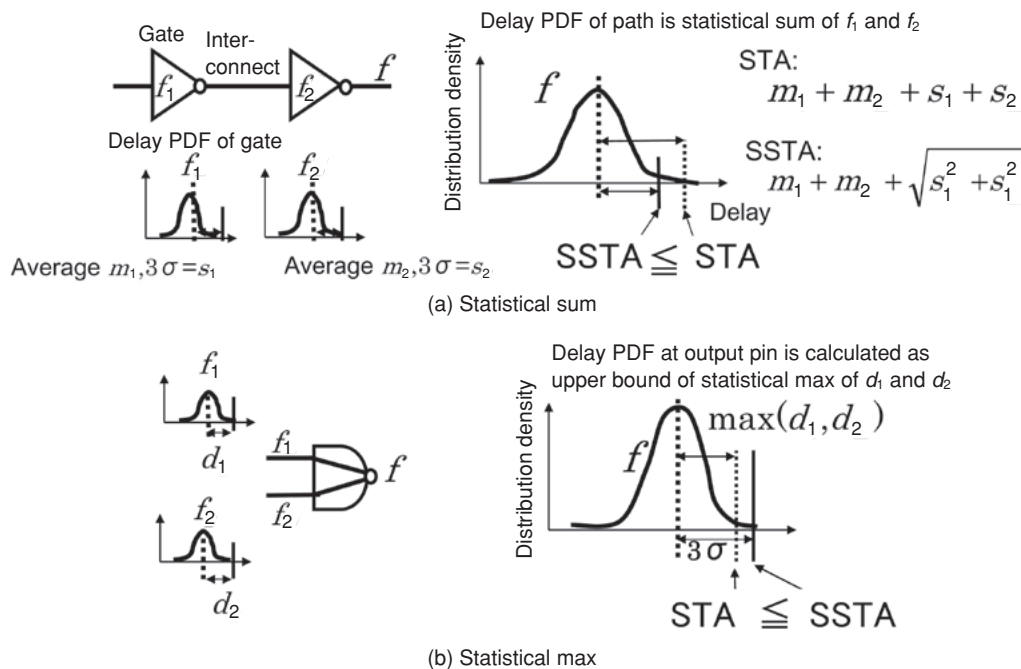


Figure 2
Basic operations of SSTA.

4. SSTA research by Fujitsu Laboratories

In 2003, Fujitsu Laboratories started researching SSTA as a theme of DFM technology. At that time, many study results about SSTA techniques were reported, although there were no reports of an SSTA application to an actual design. Consequently, general researchers did not know about the effects and practical-application problems of SSTA.

We initiated research to determine the effects and problems of applying SSTA to Fujitsu processor and ASIC designs and developed the SSTA engine as a basic tool for this research. In this development, we considered it important to easily realize various SSTA techniques and incorporate these techniques into existing design flows. We therefore developed an SSTA application program interface (API) that implements the basic statistical operations described in the previous section and the path-based and block-based methods.

Next, in collaboration with Fujitsu Limited, we incorporated the SSTA engine as an SSTA tool for processor design and evaluated the effects of SSTA. Then, by using the technical knowledge acquired during our evaluation, we developed

and evaluated an SSTA tool for ASIC design in a joint project between Fujitsu Limited and Fujitsu VLSI Limited. As a result, we clarified the effects of introducing SSTA and identified practical-use problems in processor and ASIC designs. We also improved the tool for practical use, constructed a design flow, and launched a production run for chip design in the latter half of 2006.

5. Applying SSTA to processor design

The advantage of applying SSTA to processor design is that, because we can predict the timing yield with SSTA, we can consider the trade-off between circuit performance and timing yield during the design phase. To accurately predict the timing yield, SSTA must analyze the entire circuit. Also, the analysis must be completed within a practical amount of time. For these reasons, the SSTA tool for processor design uses the block-based SSTA method. The SSTA tool can statistically handle D2D variations as well as WID random variations, so the timing yield can be predicted more accurately. **Figure 4** shows the SSTA flow used by the tool to predict the timing yield of manufactured processors.³⁾ In this flow, we predict manually the WID random

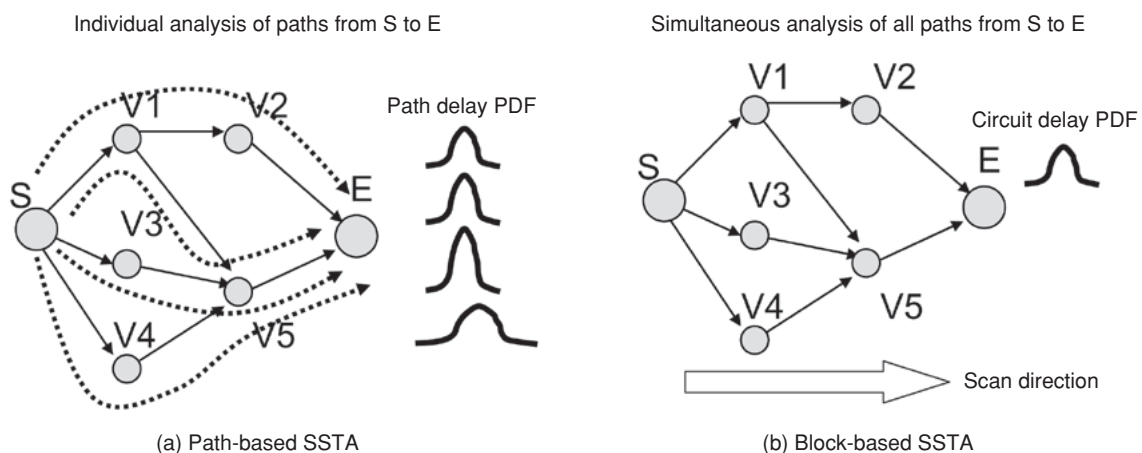


Figure 3
Path-based and block-based SSTA.

variations of the gates and interconnects that are inputs of the SSTA tool using the manufacturing data of older-generation technologies. For the D2D variations, we use the standard values of the existing technology of the target circuit because we do not know the actual values at the introduction of a new technology. However, once the manufacturing conditions are fixed and the operating frequencies of the chips' ring oscillators can be measured, we use the values of the D2D variations from the variations of the measured frequencies of the ring oscillators. Also, once we obtain the results of frequency selection, we can feed them back to the flow and correct the difference between the timing yields acquired by frequency selection and the SSTA-predicted values. As the manufacturing proceeds, the feedback operations conducted after manufacturing produce a system that improves the accuracy of prediction. For example, for one particular design, we confirmed that the yield errors were within 10%.

6. Applying SSTA to ASIC design

ASIC design uses SSTA for "timing sign-off," which is done at the last stage of timing verification in the design flow to confirm that the circuit frequency acquired by timing analysis flow is within the target frequency. In timing sign-off, all paths must satisfy their timing constraints. Therefore, our SSTA tool uses a path-based algorithm that can accurately calculate path delay. We also devised a new technique for alleviating the timing constraint of each path as compared with the conventional SSTA technique and incorporated it into the SSTA tool.⁴⁾ To apply our SSTA tools to Fujitsu's design flow, the following two conditions must be satisfied:

- 1) If the STA check results are replaced by the SSTA check results, a yield equivalent to the conventional yield must be assured.
- 2) The designer's workload must be minimized.

First, to assure a yield equivalent to the conventional yield, we must improve the accuracy of the SSTA tool to prevent the estimation of optimistic delays. Errors in the outputs of our SSTA tool are unavoidable because of the

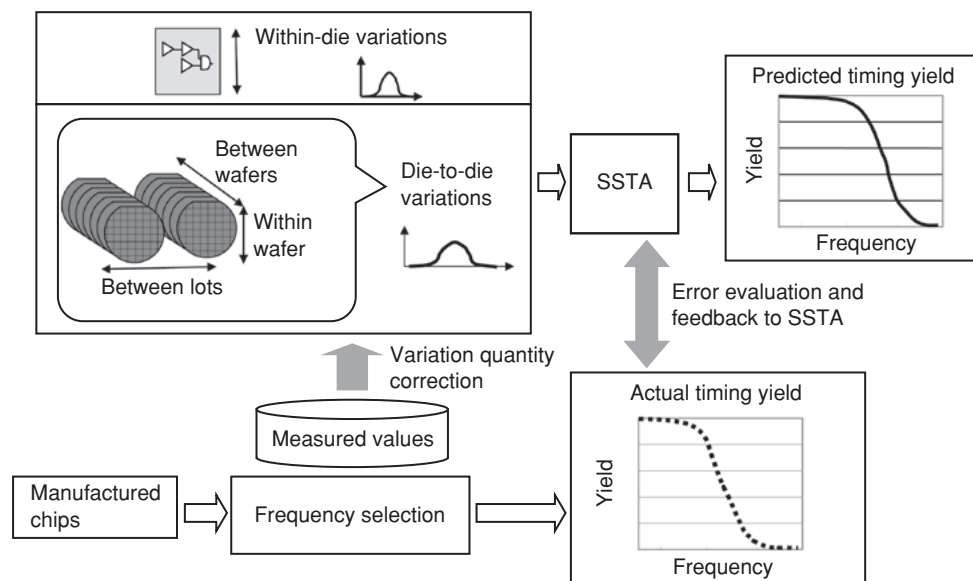


Figure 4
SSTA flow for processor design.

errors in the models, algorithms, and inputs. Especially, input gate delay variations are difficult to accurately estimate because some of the factors of the delay variations are difficult to analyze. Therefore, our SSTA tool only extracts random variations among the gate delay variations and processes them statistically. Regarding the other delay variation factors, the SSTA tool uses the worst-case conditions in the same way as the conventional STA. Therefore, the risks of optimistic delay estimations are avoided and yields at least as good as those of the conventional flow are assured.

Secondly, to minimize the designer's workload, the additional load due to statistical information handling and the additional time due to SSTA processing must be reduced. Therefore, we propose that the SSTA tool analyzes only the critical paths that are detected by the conventional STA tool. **Figure 5** compares this new design flow with the conventional design flow.

If the sign-off processing using the conventional STA tool indicates the NG state, the developed SSTA tool is used for postprocessing. The critical path list detected by the STA tool and the delay variation information are both input to the SSTA tool. The SSTA results are added to

the input critical path list, which makes it easier for the designer to confirm the SSTA results.

In this SSTA design flow, the SSTA tool analyzes only the critical paths, which prevents an increase in processing time. Moreover, we adopted a yield assurance check technique⁵⁾ that determines how many critical paths must be analyzed to assure the expected timing yield. Therefore, a timing yield equivalent to the conventional yield can be assured using this SSTA design flow.

Figure 6 shows an example of applying this SSTA design flow to a 90 nm technology circuit. The graph shows the timing yield distribution calculated from the circuit delay probability distribution using SSTA. The frequency equivalent to a yield of 3σ is 221 MHz. Compared with the conventional STA estimation frequency of 209 MHz, this SSTA estimation frequency is an improvement of about 5.7%. This frequency improvement reduces the time needed for timing optimization: whereas the conventional design flow for this circuit took one month, the new design flow only took 20 days. Therefore, the SSTA design flow can reduce the design TAT.

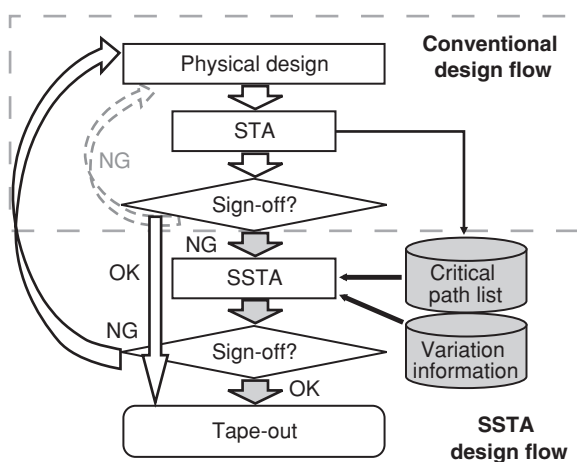


Figure 5
SSTA and conventional ASIC design flows.

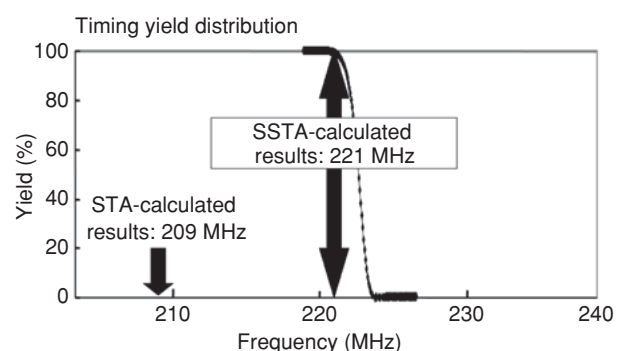


Figure 6
Results of design flow applied to ASIC design.

7. Conclusion

This paper described the basic techniques of SSTA and its application to solve problems that occur due to increases in delay variations. It also described the use and effectiveness of SSTA in the design of Fujitsu processors and ASICs.

In the future, we will improve the correlation with actual yields by adding more examples of applying SSTA to processor and ASIC designs and provide solutions for improving design efficiency by feeding back SSTA results.

Several vendors have released SSTA tools since late 2006. In the future, we will be able to quickly apply these tools to Fujitsu's design flows by using the technical knowledge acquired from our SSTA applications.



Izumi Nitta, *Fujitsu Laboratories Ltd.*
Ms. Nitta joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1991, where she has been engaged in research and development of VLSI physical design analysis and optimization. She is a member of the Information Processing Society (IPSJ) of Japan.

E-mail: nitta.izumi@jp.fujitsu.com



Toshiyuki Shibuya, *Fujitsu Laboratories Ltd.*
Mr. Shibuya joined Fujitsu Laboratories Ltd., Kawasaki, in 1985, where he has been engaged in research and development of VLSI physical design analysis and optimization, and parallel computing. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

E-mail: shibu@jp.fujitsu.com

References

- 1) S. Tsukiyama: Statistical Timing Analysis: A Survey. (in Japanese), The 18th Workshop on Circuits and Systems in Karuizawa, 2005, p.533-538.
- 2) A. Srivastava et al.: Statistical Analysis and Optimization for VLSI: Timing and Power. Springer, 2005.
- 3) H. Komatsu et al.: Statistical Timing Analysis and its Application to Microprocessor Design. (in Japanese), IPSJ Symposium Series, Vol. 2006, No.7, p.1-6.
- 4) I. Nitta et al.: A Study of the Model and the Accuracy of Statistical Timing Analysis. (in Japanese), *IEICE Technical Report*, VLD 2005-71, **105**, 442, p.61-66 (2005).
- 5) K. Homma, et al.: A Study of the Execution Path Number and the Accuracy of Path-Based Statistical Timing Analysis. (in Japanese) *IEICE Technical Report*, CPSY2005-74, **105**, 669, p.61-66 (2006).



Katsumi Homma, *Fujitsu Laboratories Ltd.*

Mr. Homma received B.S. degree in Physics from Hirosaki University in 1985 and M.S. degree in Mathematics from Niigata University in 1988, respectively. He joined Fujitsu Ltd. in 1992 and moved to Fujitsu Laboratories in 1994. His current research interest is VLSI physical design and optimization. He is a member of

the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

E-mail: km.homma@jp.fujitsu.com