

A Practical Clock Tree Synthesis for Semi-Synchronous Circuits

Masahiko Toyonaga, Keiichi Kurokawa, Takuya Yasui

Corporate Semiconductor Development Division
Matsushita Electric Industrial Co., Ltd.
3-1-1, Yagumo-Nakamachi, Moriguchi
Osaka 570-8501 Japan
{toyo, kurokawa, yasui}@mrg.csdd.mei.co.jp

Atsushi Takahashi

Dept. of Electrical and Electronic Eng.
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro
Tokyo 152-8552 Japan
atushi@ss.titech.ac.jp

ABSTRACT

In this paper, we propose a new clock tree synthesis method for semi-synchronous circuits. A clock tree obtained by the proposed method is a multi-level multi-way clock tree such that a clock-input timing of each register is a multiple of a predefined unit delay and the length of interconnection from a parent node to its child is upper bounded. The clock trees are constructed for several practical circuits. The size of each clock tree is comparable to a zero skew clock tree. In order to assure the practical quality, they are examined under the five delay conditions, which cover various environmental and manufacturing conditions. As a result, they are proved stable under each condition and improve the clock speed up to 17.3 % against the zero skew clock trees.

Keywords

Semi-synchronous, clock-input timing, clock scheduling, environmental and manufacturing conditions, zero skew clock tree, various timing clock tree.

1. INTRODUCTION

The semiconductor manufacturing process technology has improved the scale, performance and power consumption of LSI circuits. To make a design of a large-scale circuit simple, most of designers adopt a *synchronous circuit* strategy of dividing a circuit into a clock part and a logic part. In designing a logic part, they assume a simultaneously distributed clock over the chip.

In DSM (Deep sub-micron) process such as 0.25-micrometer rule, the interconnection delay takes a large part of the signal propagation delay, and which makes a clock synthesis difficult to achieve the above assumption, as well as a logic synthesis difficult. To overcome this difficulty and to improve the circuit performance, a new design methodology is required.

A *semi-synchronous circuit* is one of solutions, that is a circuit with a global clock enabling and utilizing different clock-input timings of registers.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISPD 2000, San Diego, CA.
Copyright 2000 ACM 1-58113-191-7/00/0004...\$5.00

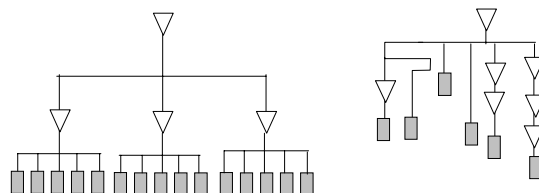
Fishburn [1] gave the necessary and sufficient condition that a circuit works with a clock period in terms of the maximum and minimum signal propagation delays between registers and their clock-input timings. He showed a possibility to improve the clock speed by tuning clock-input timings.

Takahashi et al. [2,3] interpreted the necessary and sufficient condition using a constraint graph, and introduced an algorithm that determines the shortest feasible clock period and clock-input timings for all registers. Albrecht et al. [4] proposed a different calculation method for a latch-based circuit. Neves and Friedman has extended it to design a robust circuit under the process variation [5]. Kourtev recently proposed a new logic optimization method that combines the re-timing technique and the semi-synchronous input-timing technique [6]. They showed that the synthesis methods for semi-synchronous circuits, but they did not specify the layout realization of the circuits.

In the synchronous framework as well as conventional framework, it is essential to control the clock-input timings. In a practical clock design, the deviation of clock-input timings, called *clock skew*, should be small under the environmental and manufacturing conditions.

In a usual type of clock circuit called *zero-skew clock tree* (ZSCT) as shown in figure 1(a), the clock skew would be relatively small under these conditions since the numbers of buffers from the clock source to registers are the same, and the interconnection delays are almost same [7,8,9,10].

On the other hand, in a clock tree as shown in figure 1(b) for the semi-synchronous framework, called a *various timing clock tree* (VTCT), the delays from the clock source to registers differ.



(a) Zero skew clock tree (b) Semi-synchronous clock tree

Figure 1 Zero-skew clock tree and semi-synchronous clock tree

There are two difficulties to realize VTCT. The first is the size of clock tree that tends to be larger than the ZSCT. The second is the stability of the clock-input timings under the environmental and manufacturing conditions. Xi and Dai [11] provided a faster

clock circuit by constructing a VTCT, which is obtained by modifying a ZSCT. Inoue et al. [12] showed that a smaller clock tree is obtained by setting the clock-input timings between the neighboring registers as small as possible. Chen et al. [13] proposed the associated skew problem in which registers of the same clock timings are clustered. They claimed their approaches provide smaller VTCT, though they did not examine the actual circuits [11-13]. Furthermore, there still remains the second difficulty in these approaches. The delay variation caused by interconnection becomes comparable to the gate delay in a DSM process. It is important to restrict it in constructing a VTCT.

In this paper, we propose a new clock tree synthesis method for semi-synchronous circuits. A clock tree obtained by the proposed method is a multi-level multi-way clock tree. First, using the method of simulated-annealing, the synthesis procedure assigns a clock-input timing to every register that is a multiple of a predefined unit delay and that satisfies the setup-hold constraint. Next, the synthesis procedure clusters the nodes of the same clock-input timing, and generates a parent node for each cluster. The clock-input timing of a parent node is set to that of the children minus the unit delay. The length of interconnection from a parent node to its child is upper bounded in order that the resistance of the interconnection might not affect the propagation delay.

The clock trees are constructed for several practical circuits. The size of each clock tree is comparable to a zero skew clock tree. In order to assure the practical quality, they are examined under the five delay conditions, which cover various environmental and manufacturing conditions. As a result, they are proved stable under each condition and improve the clock speed up to 17.3 % against the zero skew clock trees.

2. CONSTRAINTS OF SYNCHRONOUS CIRCUIT AND ITS VERIFICATION

Figure 2 illustrates a part of a synchronous circuit. Let Reg(i) and Reg(j) be the registers to which a clock with period T is inputted. Let S_i and S_j be the clock-input timings of Reg(i) and Reg(j), respectively. Let Cl be the combinational circuit between Reg(i) and Reg(j) with the signal propagation path $P_{max}(i,j)$ of the maximum delay $W_{max}(i,j)$ and $P_{min}(i,j)$ of the minimum delay $W_{min}(i,j)$.

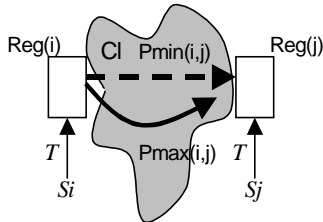


Figure 2 A synchronous circuit

A synchronous circuit works with period T if and only if the following two inequalities are satisfied for every register pair with signal propagation paths as in [1].

- (1) Setup constraint (no zero clocking constraint)

$$S_i - S_j \leq T - W_{max}(i, j)$$

- (2) Hold constraint (no double clocking constraint)

$$S_j - S_i \leq W_{min}(i, j)$$

In case of design based on the conventional zero-skew clock tree, since S_i and S_j are assumed to be equal, the inequalities can be rewritten as follows.

$$S_i - S_j = 0$$

$$0 \leq W_{min}(i, j) \leq W_{max}(i, j) \leq T_0$$

The clock period T_0 of such a circuit is constrained by $W_{max}(i,j)$. In case of semi-synchronous circuit, since S_i and S_j are not necessarily to be equal, the following inequality is obtained.

$$S_i - S_j + W_{max}(i,j) \leq T_s \leq W_{max}(i,j) \leq T_0$$

This shows that if $S_i - S_j$ is less than zero, then the clock period would be less than the maximum path delay of the circuit. The shorter clock period is achieved if clock-input timings are set appropriately. We call it **clock schedule optimization**.

Whether a circuit satisfies the setup-hold constraint after the clock schedule optimization would be assured visually by plotting a dot for each register pair with the signal propagation. The diagrams in figures 3 (1) and (2) correspond to the setup-hold verification, respectively, where the horizontal axis is S_j , and the vertical axes are $W_{max}+S_i-T$ (Setup) and S_i+W_{min} (Hold), respectively.

If there is no violations in the circuit, dots are plotted in the shaded area, otherwise, the plotted area would be move over the diagonal line in figure 3.

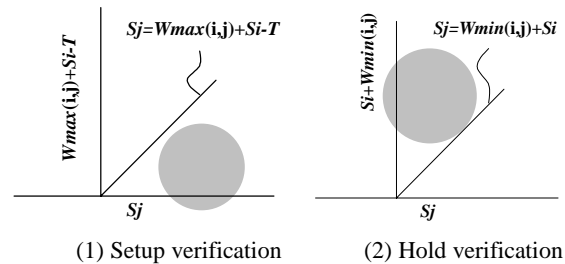


Figure 3 Verification diagrams for a synchronous circuit

2.1 The verification under the environmental and manufacturing variations

In order to assure the practical quality of LSI, we have to guarantee the functionality in several environments under voltages, temperatures, and process variations. This assurance is verified using a timing analysis tool by setting various gate delays and interconnection delays corresponding to the conditions.

A simple signal path delay model that we adopt here is

$$D_{path}(Cond) = D_{gate}(Cond) + D_{wire}(Cond),$$

where *Cond* represents a type of condition, D_{path} , D_{gate} and D_{wire} are the total signal path delay and the delays caused by the gates and the interconnections on the path, respectively. In the process rule larger than 0.5-micrometer when the process variation and the interconnection resistance are small, D_{wire} is

almost a constant since it mainly depends on the interconnection capacitance, which is constant under the voltage and temperature variation. It is sufficient to verify only three conditions according to the gate delay variation in those process rules.

$$\begin{aligned} D_{\text{path}}(\text{Typ}) &= D_{\text{gate}}(\text{Typ}) + D_{\text{wire}}, \\ D_{\text{path}}(\text{Min}) &= D_{\text{gate}}(\text{Min}) + D_{\text{wire}}, \\ D_{\text{path}}(\text{Max}) &= D_{\text{gate}}(\text{Max}) + D_{\text{wire}}, \end{aligned}$$

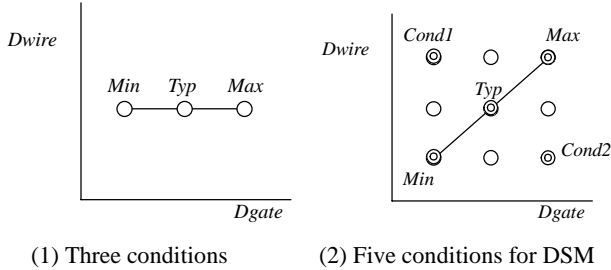
where Typ, Min and Max correspond to the conditions of typical, fastest, and slowest gate speed according under the corresponding voltages and temperatures, respectively.

Contrarily, in DSM process smaller than 0.5-micrometer where the process variation and the interconnection resistance are larger, the delay variation caused by interconnection becomes comparable to the gate delay. Therefore we have to consider extra three interconnection delay variations about Dwire.

As the result, the total number of conditions becomes nine as shown in figure 4(2), however, it is sufficient to examine only five conditions that cover whole nine conditions. Let us annotate these new two conditions as **Cond1** and **Cond2**.

$$\begin{aligned} D_{\text{path}}(\text{Min}) &= D_{\text{gate}}(\text{Min}) + D_{\text{wire}}(\text{Min}), \\ D_{\text{path}}(\text{Typ}) &= D_{\text{gate}}(\text{Typ}) + D_{\text{wire}}(\text{Typ}), \\ D_{\text{path}}(\text{Max}) &= D_{\text{gate}}(\text{Max}) + D_{\text{wire}}(\text{Max}), \\ D_{\text{path}}(\text{Cond1}) &= D_{\text{gate}}(\text{Min}) + D_{\text{wire}}(\text{Max}), \\ D_{\text{path}}(\text{Cond2}) &= D_{\text{gate}}(\text{Max}) + D_{\text{wire}}(\text{Min}). \end{aligned}$$

We verify the setup-hold constraint under these five conditions using verification diagrams shown in figure 3.



(1) Three conditions (2) Five conditions for DSM

Figure 4. Environmental and manufacturing five conditions

3. A NEW CLOCK TREE

We can design a circuit to satisfy the setup-hold constraint under the worst or the typical or the minimum conditions where the gate and interconnection delays change similarly. Contrarily the gate delay and wire delays change independently under Cond1 and Cond2 conditions. The delay differences between the logic and the clock become complicated, and we have to worry about the timing errors. Therefore we propose a clock tree where the clock-input timing is realized without the interconnection resistance as the practical clock design under the five conditions. In figure 5, we illustrate a new clock tree for the semi-synchronous circuit.

The proposed clock tree is a multi-level multi-way clock. The root and internal nodes of the tree correspond to the drive buffers, and the leaves correspond to the registers. A drive buffer drives other drive buffers and/or registers. The clock-input timing of each node represents the clock delay from the clock source to the node minus the target clock delay.

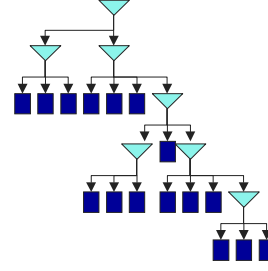


Figure 5. A new clock tree

3.1 Optimization of the discrete clock schedule

We use the method of simulated annealing [14] to obtain a discrete clock scheduling solution. A multiple of the unit delay value of Ut, which is defined by the **signal slew rate** or **fan-out constraints** in the circuit, is assigned to every register.

In simulated annealing, we evaluate the clock schedule by the following cost function.

$$\begin{aligned} COST &= A \times \text{Max}_{\{i,j\}} \{S_i - S_j + W_{\text{max}}(i,j) - T\} \\ &+ B \times \sum_{\{i,j\}} \{S_i - S_j + W_{\text{max}}(i,j) - T\} \\ &+ C \times \sum_{\{i,j\}} \{S_j - S_i - W_{\text{min}}(i,j)\} \\ &+ D \times \sum_j (S_j - S_0)^2 \end{aligned}$$

where T, {i,j}, j and S0 denote the minimum clock period that is calculated by the clock scheduling method in which continuous clock timings are allowed [2], all the register pairs with signal propagation, all the registers and the target clock delay, respectively. The objective is to obtain a clock schedule with shorter clock period that consists of discrete clock timings satisfying equations (1) and (2). The first term denotes the difference of the target clock period T and an achieved clock. The second and third terms are the sum of the setup violations and the sum of the hold violations, respectively. The fourth term objective is to obtain a akin tree to ZSCT that takes practical size [15]. The parameters of A, B, C and D are constant values.

3.2 A clustering based clock tree synthesis

The clock synthesis procedure clusters registers, which are assigned the maximum clock-input timing at the first step and generates a parent node for each cluster. The clock-input timing of the parent node of a cluster is set to be equal to the clock-input timing of children in the cluster minus Ut. The delay of Ut from the parent node to children is controlled by the gate delay of parent buffer and its driving performance and the total capacitance of children's input-pins and interconnection to them.

In the clustering phase, first, the pair of registers with the minimum estimated interconnection length is selected as a seed of a cluster, and then the location of a parent node of the cluster is determined. Next, the cluster is enlarged by adding registers according to the distance from the parent node until the estimated delay reaches Ut.

In the enlargement of cluster, the maximum length of interconnections from the parent node to a child as well as the

wiring capacitance and input capacitance of registers is limited to reduce the propagation delay caused by the interconnection resistance. The capacitance per unit length of interconnection is estimated statistically, since a shorter interconnection would not be important to the delay and a longer wire would be well estimated statistically.

Note that a single child cluster is formed if the distance of every pair of registers is larger than twice the limit of interconnection. In case that no register can be added to the cluster even if the delay does not reach U_t , the delay U_t is achieved by adjusting the topology of interconnection and by inserting a redundant interconnection.

The procedure above that clusters the leaves and internal nodes of the same clock-input timing is repeated until all the nodes are clustered from the maximum to the minimum clock-input timing. The variation of clock-input timing realized on this tree under the various conditions is small since the variation of interconnection resistance is negligible.

In Fig. 6, a light circle denotes the register with maximum clock-input timing $n \cdot U_t$. The black circles and triangles denote nodes assigned $(n-1) \cdot U_t$. Figure 6(b) shows clusters of clock-input timing of $n \cdot U_t$. The clustering is repeated as shown in figures 6(c) and (d).

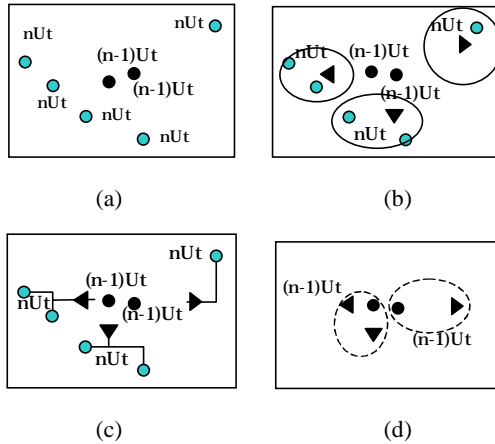


Figure 6 Process to construct the clock tree

4. EXPERIMENTS

4.1 The experimental conditions

4.1.1 Overview of the benchmark circuits

Table 1 shows the overview of the benchmark circuits. These circuits are sub blocks of an image processing LSI that has been manufactured in our company by 0.25micronmeter process on 1.8 Volt. The clock speed specification is 12.3 nanosecond (ns).

Table 1. The experiment circuit

Circuits	# of Cells	# of Regs.
C1	6536	1226
C2	21657	5363
C3	24646	7672

4.1.2 The input/output

The input/outputs for our clock synthesis are followings.

Input: net list, signal path delay, and cell placement.

Output: clock net list, clock buffer placement, clock interconnections.

4.1.3 The parameters of Simulated-annealing

The parameters of simulated-annealing are as follows.

Cooling parameter : 0.95

Start temperature : 10.0

The end temperature is defined as the temperature when the cost improvement converges within $\pm 0.1\%$.

4.1.4 The unit delay value U_t and the range of clock timings

We specify the interconnection limit as the length corresponding to the resistance that causes 5% delay of U_t in a cluster. In the experiments, we set U_t to 0.6 ns and use the corresponding buffer.

The target clock delay is 1.2 ns. The range of clock-input timing for registers is restricted from -0.6ns to 2.4 ns, which follows the target of the continuous clock schedule obtained by [2].

4.1.5 The five conditions

As a matter of fact, the five conditions are different in each silicon foundry. According to semiconductor data sheets published on the Web, Dgate (Min) / Dgate (Typ) is ranged from 0.5 to 0.7, Dgate (Max) / Dgate (Typ) is ranged from 1.6 to 1.8 in 0.25~0.35 micrometer process [16]. According those values, we select the environmental conditions as shown in table 2 for our experiments.

Dwire (Max) / Dwire (Typ) is set smaller than Dgate (Max) / Dgate (Typ) since an interconnection is effected mainly by the process variation.

Table 2. Five operation conditions

Condition	Min	Typ	Max	Cond1	Cond2
Dgate/Dgate(Typ)	0.5	1.0	2.0	0.5	2.0
Dwire/Dwire(Typ)	0.5	1.0	1.5	1.5	0.5

4.2 The experimental result

4.2.1 The benchmark system

Our clock synthesis generates the clock net lists, clock buffer placements and their interconnection topologies. They are linked to a major vendor's layout tool to complete designs. The parasitic extraction (LPE) and Standard Delay Format (SDF) files are prepared by using the vendor tool. The verification diagrams are obtained by modifying SDF files according to the five conditions.

4.2.2 The performance

We compared our clock tree and the previous VTCT obtained by [2]. We worried about our discrete timing strategy to be worse, as its solution space is limited. However, it achieved almost the same improvement level, i.e., the continuous clock schedule improved 2.1 to 19.2 % of clock speed, while our clock improved 1.2 to 17.3 %. The CPU time was about 1.5 hours in UA1 (200 MHz Work Station by the Sun-Microsystems Inc.). Note that the

improvement of the circuit C2 is small since C2 contains a signal propagation path from the register to the same register (a loop) with large propagation delay which invalidates the semi-synchronous technique.

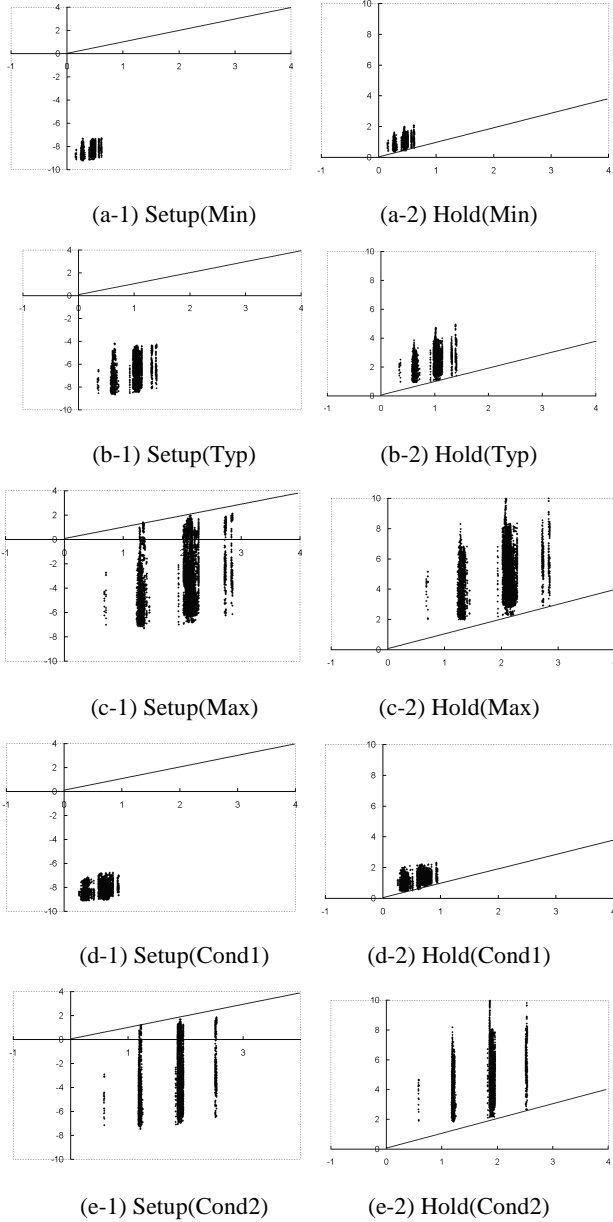


Figure 7 Verification results over the five conditions.

Table 3. Comparison of performance (unit: ns)

Circuits	Sync.	Prev. Semi-Sync	Our	Improved(%)	
				Semi	Our
C1	11.956	9.747	9.89	18.5	17.3
C2	11.911	11.66	11.67	2.1	1.2
C3	11.808	9.545	9.91	19.2	16.1

4.2.3 Verifications in the five conditions

Figures 7 (a) to (e) illustrate verification diagrams for circuit C1 under the five conditions. The horizontal and vertical axes are as the same as in figure 3.

As we selected discrete clock delay strategy, there are stripes of shaded area. They show that the clock tree satisfies the equations of (1) and (2).

To evaluate the validity of our clock tree structure, we implemented the semi-synchronous clock tree controlled only by the wire length. Fig.8 shows the verification results in Cond1. It is found that many timing violations concerned with Hold.

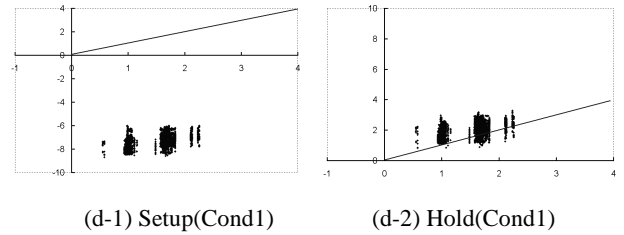


Figure 8 Verification results of the clock tree controlled by interconnection delay

Table 4. Comparison of clock tree for the benchmarks

Circuit	#of Buffers		Wire length(mm)	
	0-Skew	Ours	0-Skew	Ours
C1	46	47	45.47	46.92
C2	204	211	195.88	208.37
C3	280	281	260.70	263.11

Table 4 shows the comparison of the number of buffers and the wiring length of our clock circuit and a zero skew clock circuit. The differences of both indices are less than 7%.

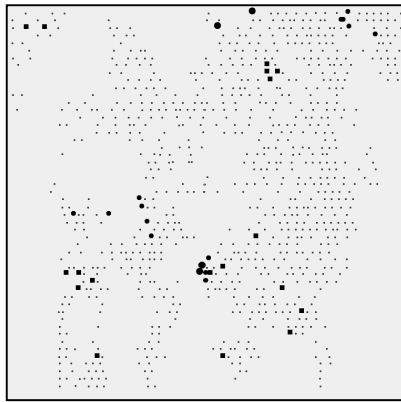
A scheduling result and a clock tree layout based on our procedure are shown in figures 9(a) and (b), respectively. The big black circle, the small black circle, the dot, and the black square are corresponding to registers with clock delay 1.2ns, 1.8ns, 2.4ns, and 3.0ns, respectively.

5. CONCLUSION

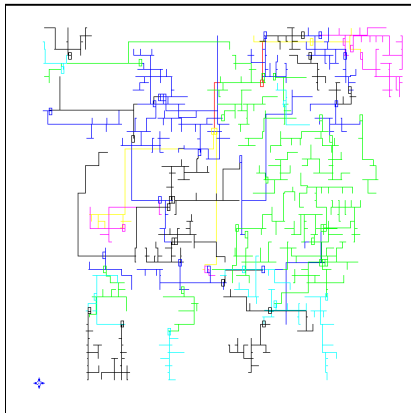
We proposed a new clock tree with synthesis method that produces appropriate clock timings of registers under the five environmental and manufacturing conditions for LSI quality assurance. This clock tree generates a discrete clock delay to each register under the limitations of the maximum wiring length that endures the influence of the environmental and manufacturing variations. The experimental results for practical three circuits showed that they satisfy the five conditions of quality assurance, while their performances were improved up to 17.3% compared to the zero-skew based circuits. As the future work, we are planning to develop the effective cost function to improve the EMI problem in our framework.

6. ACKNOWLEDGMENTS

We thank to Prof. Yoji Kajitani and Mr. Tomoyuki Yoda of Tokyo Institute of Technology for their useful suggestions, and Mr. Yoshifumi Okamoto and Mr. Shiro Tsuji of Matsushita Electric Industrial Co., Ltd. for their supports.



(a) Clock delay distributions



(b) Interconnections

Figure 9 Clock delay and interconnections of the clock tree

7. REFERENCES

- [1] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. on Computers*, Vol.39, pp945-951,1990.
- [2] A. Takahashi, Y. Kajitani, "Performance and reliability driven clock scheduling of sequential logic circuits," In Proc. of Asia and South Pacific Design Automation Conference, pp37-42, 1997.
- [3] A. Takahashi, K. Inoue, Y. Kajitani, "Clock-tree routing realizing a clock-schedule for semi-synchronous circuits," In Proc. ICCAD, pp.260-265, 1997.
- [4] C. Albrecht, B. Korte, J. Schietke, and J. Vygen, "Cycle time and slack optimization for VLSI-chips," In Proc. ICCAD, pp232-238, 1999.
- [5] J. L. Neves, and E. G. Friedman, "Optimal clock skew scheduling Tolerant to Process Variations," In Proc. 33rd Design Automation Conference, pp.623-628, 1996.
- [6] X. Liu, M. C. Capaefthymiou, and E. G. Friedman, "Maximizing performance by retiming and clock skew scheduling," In Proc. 36th Design Automation Conference, pp.231-236, 1999.
- [7] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock routing for high-performance ICs," In Proc. 27th Design Automation Conference, pp.573-579, 1990.
- [8] T. H. Chao, Y. C. Hsu, J. M. Ho, K. D. Boese, and A. B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Trans. On Circuits and Systems*, vol. 39, no. 11, pp. 799-814, 1992.
- [9] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," In Proc. 30th Design Automation Conference, pp.612-616, 1993.
- [10] D.J.-H.Huang, Andrew B.Kahng and Chung-Wen Albert Tsao, "On the bounded-skew clock and steiner routing problems," In Proc. 32nd Design Automation Conference, pp.508-513, 1995.
- [11] J. G. Xi and W. M. Dai, "Useful-skew clock routing with gate sizing for low power design," In Proc. 33rd Design Automation Conference, pp.383-388, 1996.
- [12] K. Inoue, W. Takahashi, A. Takahashi, Y. Kajitani "Schedule-Clock-Tree Routing for Semi-Synchronous Circuits," *IEICE Transactions on Fundamentals*, Vol.E82-A, No.11, PP. 2431-2439, 1999.
- [13] Y. Chen, A. B. Kahng, G. Qu, and A. Zelikovsky, "The Associative-Skew Clock Routing Problem," In Proc. ICCAD, pp.168-172, 1999.
- [14] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by simulated annealing," *Science*, 220, pp.671-680, 1983.
- [15] I. S. Kourtev, E. G. Friedman, "Clock Skew Scheduling for Improved Reliability via Quadratic Programming," In Proc. ICCAD, pp.239-243, 1999.
- [16] <http://www.bgs.nu/sdw/> "Semiconductor datasheets on the Web".