社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－SLDM－127 (20)
2006／11／29

一般同期方式におけるレジスタ再配置によるレジスタ削減手法

小平 行秀† 高橋 篤司†

† 東京工業大学大学院 理工学研究科 集積システム専攻 〒152-8552 東京都目黒区大岡山 2-12-1-S3-58
E-mail: †{kohira,atsushi}@lab.ss.titech.ac.jp

あらまし クロックを各レジスタに任意のタイミングで入力できるという仮定の下で，回路の動作や構造を変更せず，レジスタの挿入位置のみを変更するレジスタ再配置を利用することで，回路が動作可能な最小クロック周期は減少することがある．しかし，その最小クロック周期が減少する場合，回路内のレジスタ数が増加する傾向にある．そこで，本稿では目標のクロック周期を達成しつつ，回路内のレジスタ数を減少させるレジスタ再配置手法を提案する．実験では，提案手法は多くの回路で実用的な時間内にレジスタ数を削減できることを示す．

キーワード 一般同期方式，レジスタ再配置，リタイミング

A Fast Register Relocation Method for Circuit Size Reduction in Generalized-Synchronous Framework

Yukihide KOHIRA† and Atsushi TAKAHASHI†

† Department of Communications and Integrated Systems, Tokyo Institute of Technology
2-12-1-S3-58 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan
E-mail: †{kohira,atsushi}@lab.ss.titech.ac.jp

Abstract Under the assumption that the clock can be inputted to each register at an arbitrary timing, the minimum feasible clock period might be reduced by register relocation which maintains the circuit behavior and topology. But if the minimum feasible clock period is reduced, then the number of registers tends to be increased. In this paper, we propose a gate-level register relocation method that reduces the number of registers while keeping the target clock period. In experiments, the proposed method reduces the number of registers in the practical time in most circuits.

Key words generalized-synchronous framework, register relocation, retiming

1. Introduction

The semiconductor manufacturing process technology has improved the scale, speed and power consumption of LSI circuits. However, increasing the ratio of the routing delay in the propagation delay bounds the amount of improvements in the complete-synchronous framework (c-frame) in which the simultaneous clock distribution to every register is assumed. The increases of the size and power consumption of a clock distribution circuit have become serious issues in c-frame. While, the generalized-synchronous framework (g-frame) [1]～[3], in which the clock is assumed to be distributed periodically to each individual register though not necessarily to all registers simultaneously, is expected to give an essential solution. By using g-frame, the improvements of quality of circuit such as the clock frequency, clock distribution circuit size, power consumption, and etc. are expected to be achieved.

The framework of synchronization by a global clock without restriction of simultaneity was discussed in the context of clock scheduling, useful-skew, semi-synchronous, and etc. In this paper, we call the framework g-frame to emphasize the framework includes c-frame. In the beginning of studies of g-frame, clock scheduling algorithms [1]～[4] and clock distribution circuit synthesis algorithms [5], [6] for given logic circuits were proposed. However, given logic circuits are synthesized for c-frame. Since the clock period might not be reduced in g-frame even if the maximum delay is reduced, the effort in c-frame might degrade the circuit performance in g-frame. So the optimization of circuit synthesis that takes

g-frame into account must be investigated.

As logic circuit modification methods that improve the performance in g-frame, delay insertion methods [7]~[9], a gate sizing method [10], a multi-clock cycle path method [11], and a register relocation method [12] are proposed.

In c-frame, the circuit modification in which registers are relocated while maintaining the circuit behavior and topology is called retiming [13]. But, in g-frame, retiming may be confused with the change of the clock input timing of a register. Therefore, in g-frame, we call it register relocation.

In [12], a mixed integer linear programming (MILP) formulation and a heuristic algorithm of the register relocation in g-frame are proposed. The objective of these algorithms is the clock period minimization or the tolerance maximization to clock signal delay variations. But since the computation time of these algorithms is too long, these algorithms cannot be applied to circuits with thousands of gates. In [14], the register relocation method for the clock period minimization in g-frame is proposed. We call this method P-CR (Period reduction by Cone Relocation). P-CR is guaranteed to achieve the lower bound of the minimum clock period by register relocation in g-frame if the delay of each element is unique. Since the computation time of P-CR is short, P-CR can be applied to large circuits. But the number of registers obtained by P-CR tends to be increased, since P-CR only applies cone relocations to reduce the minimum clock period in g-frame even if the numbers of registers are increased.

In this paper, we propose a gate-level register relocation method to reduce the number of registers in g-frame. We call this method S-CR (Size reduction by Cone Relocation). S-CR is a greedy local circuit modification method in g-frame. A register relocation that reduces the number of registers most is applied iteratively if the target clock period is kept. In experiments, we use P&S-CR in which S-CR is used as post processing of P-CR. The number of registers by P&S-CR is smaller than that by P-CR in most cases. Although there are cases that the number of registers by P&S-CR is not optimal, an optimal circuit is obtained when the method by MILP outputs a circuit. The computation time of P&S-CR is less than 10 minutes for most circuits with thousands of gates.

2. Preliminaries

In this paper, we consider a circuit consisting of registers and gates, and wires connecting them. We call them elements. A circuit is represented by the graph $G = (V, E)$, where vertex set V corresponds to elements in the circuit and directed edge set E corresponds to signal propagations in the circuit.

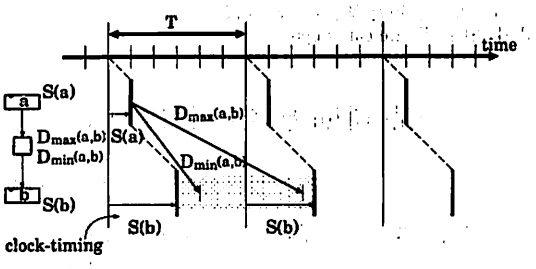


Fig. 1 Timing chart.

2.1 Generalized-Synchronous Framework

In c-frame, the clock timing of a register is the same as those of the other registers. C-frame, in which the clock timings of registers are assumed to be equal, is a kind of g-frame. In g-frame [1]~[4], the clock timing of a register may be different from other registers. The clock timing $S(r)$ of register r is defined as the difference in the clock arrival time between r and an arbitrary chosen reference register.

A circuit works correctly with a clock period T if the following two types of constraints are satisfied for every register pair with signal propagations (Fig. 1) [1].

Setup (No-Zero-Clocking) Constraints

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

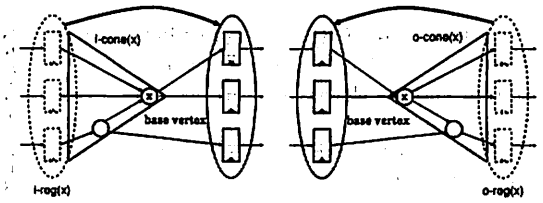
Hold (No-Double-Clocking) Constraints

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

where $D_{\max}(a, b)(D_{\min}(a, b))$ is the maximum (minimum) delay from a register a to a register b.

Since c-frame has the premise that a clock ticks all the register simultaneously, the clock period must be larger than the maximum delay between registers. On the other hand, in g-frame, the circuit can work correctly with the clock period which is smaller than the maximum delay between registers, if all the register pair with the signal path satisfies two types of constraints.

Let $T_S(G)$ be the minimum clock period of a circuit G in g-frame under the assumption that the clock can be inputted to each register at an arbitrary timing. Hereafter, we simply call $T_S(G)$ the minimum clock period of G in g-frame. $T_S(G)$ is determined by the *constraint graph* $H(G) = (R, A)$ for G, where vertex set R corresponds to registers in G and directed edge set A corresponds to two types of constraints. An edge in A from a register a to a register b with weight $D_{\min}(a, b)$, called the D-edge, corresponds to the Hold constraint, and an edge from a register b to a register a with weight $T - D_{\max}(a, b)$, called the Z-edge, corresponds to the Setup constraint. Let $H(G, t)$ be the constraint graph in which the clock period T is set to t. Let the weight of a



(a) Forward (f-reloc(x)). (b) Backward (b-reloc(x)).

Fig. 2 Cone relocation.

directed cycle in $H(G,t)$ be the sum of edge weights on the directed cycle. It is known that the minimum clock period in g-frame is determined as in the following theorem.

[Theorem 1] ([3]) $T_S(G)$ is the minimum t such that there is no cycle with negative weight in the constraint graph $H(G,t)$.

A cycle whose weight is zero in $H(G,T_S(G))$ and is negative in $H(G,t)$ where $t < T_S(G)$ is said to be critical and determines the minimum clock period $T_S(G)$ in g-frame.

2.2 Register Relocation

Register relocation method is a circuit modification method in g-frame. Let i-cone(x), the input cone of a vertex x in G, be the set of vertices of G from which a signal propagates to x without go through registers in G. Let i-reg(x) be the set of input registers of i-cone(x). An edge (u,v) in G is called an output of the i-cone(x) if u is in i-cone(x) and v is not in i-cone(x).

The *forward cone relocation* of a vertex x (f-reloc(x)) is a modification of G in which all i-reg(x) are removed and which a register is inserted to each output of i-cone(x) (Fig. 2 (a)). Similarly, the backward cone relocation of x (b-reloc(x)) is defined (Fig. 2 (b)). In f-reloc(x) or b-reloc(x), x is called the *base vertex*. Since i-reg(x) and outputs of i-cone(x) (o-reg(x) and inputs of o-cone(x)) in forward (backward) cone relocation can be determined by depth first search in G, the time complexity of a cone relocation is $O(|E|)$ if a base vertex x is given.

In a cone relocation, we can consider that a register is relocated along a path in the circuit with duplication when the path branches and with merging when the path converges. A cone relocation is an enhancement of the well-known register relocation [13] of a vertex which we call *vertex relocation*. A cone relocation can be defined as the set of vertex relocations.

The minimum clock period of a circuit in g-frame is changed by register relocation. Let the *lowest clock period* T_L of a circuit be the minimum of the minimum clock periods of all circuits which are obtained by register relocations.

3. Proposed Method

Let the size gain of a cone relocation be the difference in the number of registers before and after the cone relocation. Let the period slack of a cone relocation be the difference between the target clock period and the minimum clock period in g-frame after the cone relocation. By a cone relocation with positive size gain and non-negative period slack, the number of registers is reduced while the clock period constraint is satisfied.

Procedure P_CR($G, T_{tar}(G)$)
Input : circuit G, target clock period $T_{tar}(G)$
Output : circuit G obtained by cone relocations
Step 1 : Determine $T_S(G)$ from the constraint graph. If $T_S(G) \leqq T_{tar}(G)$, then output G and terminate.
Step 2 : Obtain a cone relocation with the maximum size gain among cone relocations which break critical cycles.
Step 3 : Obtain G by the cone relocation, and go to Step 1.

Fig. 3 P-CR [14]

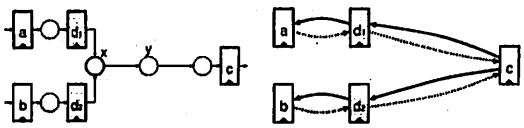
In P-CR (Period reduction by Cone Relocation) [14], a cone relocation that breaks a critical cycle in $H(G)$ is applied iteratively to reduce the clock period of G until the lowest minimum clock period T_L or the given target clock period is achieved (see Fig. 3). If there are some cone relocations that break critical cycles, P-CR applies a cone relocation with the maximum size gain among them. It is shown that T_L is achieved by P-CR under the assumption that the delay of each element is unique. But the number of registers in the obtained circuit tends to be increased since P-CR only applies cone relocations to reduce the minimum clock period in g-frame even if their size gains are negative.

In this paper, we propose a register relocation method that reduces the number of registers while keeping the target clock period. We call this proposed method S-CR (Size reduction by Cone Relocation). In S-CR, among the cone relocations with non-negative period slack, a cone relocation with maximum size gain is applied iteratively until there exists no cone relocation with positive size gain and non-negative period slack.

Note that the size gain of a cone relocation is obtained by depth first search in G in $O(|E|)$ time. While, in order to obtain the period slack, the constraint graph must be updated which takes $O(|E| \cdot |R|)$ time and the minimum clock period in g-frame must be computed which takes $O(k \cdot |R| + k^2 \cdot |A|)$ time [4], where k is the maximum number of edges in a shortest trail in the constraint graph which can be regarded as a small constant in most cases. Since $|R|$ and $|A|$ are smaller than $|E|$ in general, the time complexity of the period slack computation depends on updating the constraint graph and which is larger than that of the size gain computa-

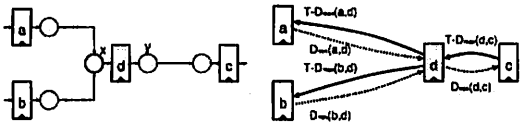
Procedure S_CR($G, T_{tar}(G)$)
Input : circuit G, target clock period $T_{tar}(G)$
Output : circuit G obtained by cone relocations
Step 1 : Compute the size gains of all cone relocations of G and sort cone relocations with positive size gains in non-increasing order.
Step 2 : Compute the period slack of a cone relocation according to the size gain order. If a cone relocation with non-negative period slack is found, then obtain G by the cone relocation and go to Step 1. Otherwise, output circuit G and terminate.

Fig. 4 S-CR



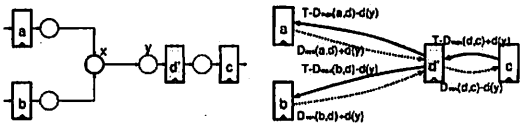
(a) circuit. (b) constraint graph.

Fig. 5 A part of original circuit.



(a) circuit. (b) constraint graph.

Fig. 6 After f-reloc(x).



(a) circuit. (b) constraint graph.

Fig. 7 After f-reloc(y).

tion. Therefore, the number of the period slack computations should be small enough to handle a circuit with thousands of gates.

In S-CR shown in Fig. 4, first, the size gains of all cone relocations are computed and cone relocations with positive size gain are sorted in non-increasing order. Then, S-CR computes the period slack of a cone relocation according to the size gain order until a cone relocation with non-negative period slack is found. If a cone relocation with non-negative period slack is found, then it is applied to the circuit and S-CR repeats from the size gain computation. Otherwise, S-CR terminates and outputs the circuit.

In order to save the computation time, when the delay of each element is unique, the cone relocations of a vertex with single incoming and single outgoing edges such as buffers or inverters are excluded since these size gains are zero or there exist other cone relocations with the same size gain and period slack. For example, the circuit and constraint graph obtained from the circuit shown in Fig. 5 (a) by f-reloc(x) and f-reloc(y) are shown in Fig. 6 and Fig. 7, respectively. It is not necessary to apply f-reloc(y), since the size gain and period slack of f-reloc(y) are same as those of f-reloc(x).

Procedure P&S_CR($G, T_{tar}(G)$) **or** P&S_CR(G)
Input : circuit G, (target clock period $T_{tar}(G)$)
Output : circuit obtained by cone relocations
Step 1 : If $T_{tar}(G)$ is not given, determine $T_L(G)$ from the constraint graph consisting of only Z-edges ([8]), and set $T_{tar}(G)$ to $T_L(G)$.
Step 2 : $G :=$ P_CR($G, T_{tar}(G)$).
Step 3 : $G :=$ S_CR($G, T_{tar}(G)$), output circuit G and terminate.

Fig. 8 P&S-CR

4. Experiments

We implement P-CR [14], and P&S-CR shown in Fig. 8 in which S-CR is used as post processing of P-CR on a PC with 3.06GHz/512K Intel Pentium-4 CPU, 512MB RAM and gcc3.5.5 of C++. Moreover, we implement S-VR(MILP) which is modified from the MILP formulation for clock period minimization [12] to that for register minimization. MILP is solved by CPLEX 9.0.0 [15]. We perform these methods on the ISCAS89 benchmark suite. In experiments, NOT gate delay is set to 1, NAND and NOR gate delay are set to 2, AND and OR gate delay are set to 3, and routing and register delays are set to 0.

First, target clock periods are set to the lowest clock period T_L. In 22 circuit among 48 ISCAS89 benchmark circuits, the minimum clock period in g-frame is not decreased by the register relocation because original circuits are optimal. The results of the other 26 circuits are shown in Table 1. In the table, T_C and T_S represent the minimum clock period in c-frame and g-frame, respectively. Moreover, #CR of size, period, and appl. represent the number of size gain computations of cone relocations, period slack computations of cone relocations, and applied cone relocations, respectively. Among 26 circuits shown in Table 1, the solutions of 21 circuits cannot be obtained by S-VR(MILP) because of the lack of memory. The optimality of P&S-CR is not guaranteed, but an optimal circuit is obtained when S-VR(MILP) outputs a circuit. Moreover, the number of registers by P&S-CR is less than that by P-CR in most circuits. The computation time of P&S-CR becomes longer than that of P-CR in all circuits, but the computation time of P&S-CR is practical in the most circuits. Table 1 shows S-CR saves the number of period slack computations. If both the size gain and pe-

Table 1 Experiment results. Target clock periods are set to the lowest clock period T_L. T_C and T_S represent the minimum clock period in c-frame and in g-frame, respectively. #CR of size, period, and appl. represent the number of size gain computations, period slack computations, and applied cone relocations, respectively.

	original					S-VR(MILP)		P&S-CR								
								P-CR [14]				S-CR				
										#CR				#CR		
model	#gate	T_C	T_S	#reg	T_L	#reg	time[s]	#reg	time[s]	size	appl.	#reg	time[s]	size	period	appl.
s298	119	18	12.0	14	10.00	17	7823.23	17	0.01	4	2	17	0.01	226	6	0
s344	160	37	34.0	15	19.00	22	1400.93	26	0.05	14	11	22	0.10	696	42	2
s349	161	37	34.0	15	19.00	22	1965.21	26	0.05	14	11	22	0.10	716	42	2
s382	158	18	12.0	21	11.25	23	72180.79	25	0.03	10	4	23	0.17	786	40	2
s400	164	18	12.0	21	11.25	23	37356.44	27	0.03	10	4	23	0.17	1004	42	3
s444	181	20	13.0	21	11.67	N/A	N/A	35	0.08	17	8	28	0.28	1352	58	4
s499	152	23	19.0	22	11.50	N/A	N/A	109	0.70	936	90	76	3.80	4328	750	14
s526	193	18	12.0	21	11.00	N/A	N/A	22	0.01	2	1	22	0.01	396	3	0
s526n	194	18	12.0	21	11.00	N/A	N/A	22	0.01	2	1	22	0.01	394	3	0
s635	286	162	158.0	32	88.50	N/A	N/A	76	1.95	101	53	61	2.03	4655	180	14
s991	519	117	110.0	19	109.00	N/A	N/A	20	0.01	2	1	20	0.77	810	3	0
s1269	569	70	61.0	37	39.34	N/A	N/A	90	2.76	103	50	80	2.06	4183	109	2
s1423	657	164	156.0	74	146.00	N/A	N/A	81	0.80	8	7	80	1.03	2036	27	1
s1512	780	54	43.0	57	40.50	N/A	N/A	61	0.11	2	2	61	0.29	1078	6	0
s3271	1572	58	34.0	116	27.72	N/A	N/A	199	6.07	159	54	174	27.49	23196	369	11
s3330	1789	66	40.0	133	32.00	N/A	N/A	147	0.93	20	8	74	3.29	11280	48	7
s3384	1685	168	154.0	183	75.50	N/A	N/A	292	13.27	214	76	222	29.37	39974	279	25
s4863	2342	144	129.0	104	75.00	N/A	N/A	219	20.50	281	103	144	182.40	80483	1406	35
s6669	3080	231	197.0	239	56.50	N/A	N/A	975	588.67	4344	798	450	11160.09	535228	23181	154
s9234	5597	107	72.0	228	63.00	N/A	N/A	240	2.30	12	4	231	83.15	54778	221	7
s9234.1	5597	107	72.0	211	63.00	N/A	N/A	223	2.23	12	4	223	135.20	100265	321	0
prolog	1601	68	40.0	136	31.00	N/A	N/A	154	0.89	20	8	78	16.76	74599	123	30
s13207	7951	106	76.0	669	75.00	N/A	N/A	670	1.40	2	1	611	656.00	1101415	497	35
s15850	9772	141	104.0	597	78.00	N/A	N/A	643	39.66	34	14	603	357.20	188929	195	22
s15850.1	9772	141	124.0	534	103.00	N/A	N/A	544	15.80	13	7	541	78.76	29656	40	1
s38417	22179	85	61.0	1636	60.00	N/A	N/A	1638	10.22	2	1	1602	245.40	185943	24	12

riod slack are computed in S-CR, then the computation time becomes much longer, since the number of cone relocations with positive size gain is much small in general.

Next, P-CR and P&S-CR are applied to s344, s3330, and s3384 in order to observe the relations between the target clock period and the number of registers. The results of s344, s3330, and s3384 are shown in Fig. 9, 10, and 11, respectively. #reg-optimal is a circuit which has the minimum number of registers without the clock period constraint obtained by solving the minimum-cost flow problem corresponding to the circuit [13]. In s344 and s3330, P&S-CR achieves the lower bound of the number of registers if the target clock periods are set to large enough. Moreover, the number of registers by P&S-CR becomes smaller than that by P-CR at most target clock periods. On the other hand, P&S-CR cannot achieve the lower bound of the number of registers in s3384, and the number of registers cannot be reduced enough at some target clock periods. These facts imply that P&S-CR is heuristic.

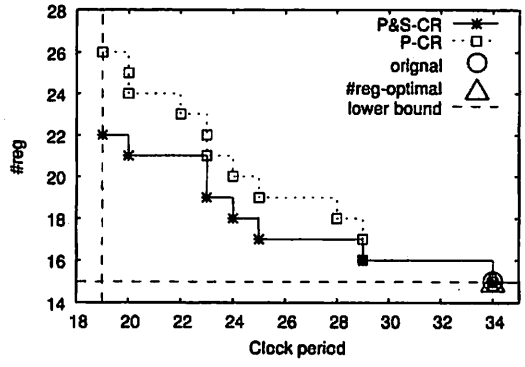


Fig. 9 Result of benchmark s344.

5. Conclusions

In this paper, we propose a register relocation method that reduces the number of registers while keeping the target clock period in the generalized-synchronous framework. The proposed method is a fast greedy local circuit modification

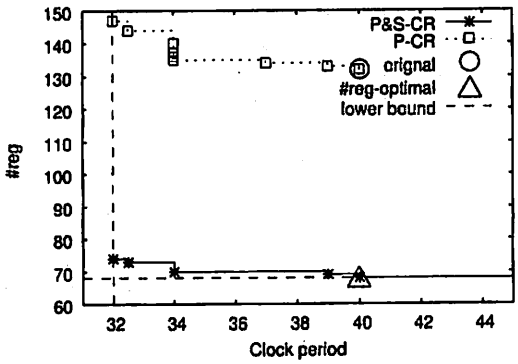


Fig. 10 Result of benchmark s3330.

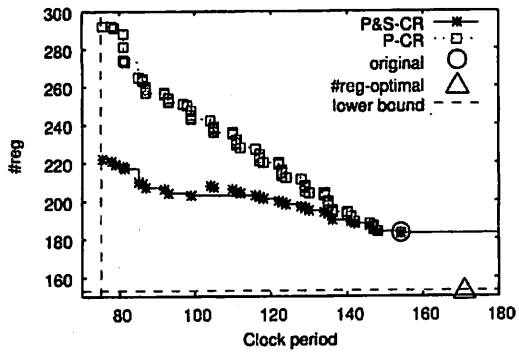


Fig. 11 Result of benchmark s3384.

method in the generalized-synchronous framework in order to reduce the number of registers until the minimum clock period in g-frame achieves the given target clock period. Experiments show that the number of registers is reduced by the proposed method, and the computation time is practical in most circuits.

Since the proposed method is heuristic, there are some cases that the number of registers cannot be reduced enough or the computation time is large. Improvements of the proposed method in register size and in computation time are in our future works. Although the clock distribution circuit synthesis that realizes the clock schedule is not in the scope of this paper, the enhancement of the proposed method to take the clock distribution circuit synthesis into account is necessary to improve the final quality of the circuit.

文　献

[1] J. Fishburn, "Clock skew optimization," IEEE Trans. on Computers, vol.39, no.7, pp.945-951, 1990.

[2] R.B. Deoker and S.S. Sapatneker, "A Graph-Theoretic Approach to Clock Skew Optimization," ISCAS, pp.407-410, 1994.

[3] A. Takahashi and Y. Kajitani, "Performance and reliability driven clock scheduling of sequential logic circuits," ASP-DAC'97, pp.37-43, 1997.

[4] A. Takahashi, "Practical Fast Clock-Schedule Design Algorithms," IEICE Trans. Fundamentals, vol.E89-A, no.4, pp.1005-1011, April 2006.

[5] K. Inoue, W. Takahashi, A. Takahashi, and Y. Kajitani, "Schedule-Clock-Tree Routing for Semi-Synchronous Circuits," IEICE Trans. Fundamentals, vol.E82-A, no.11, pp.2431-2439, 2002.

[6] S. Ishijima, T. Utsumi, T. Oto, and A. Takahashi, "A semi-synchronous circuit design method by clock tree modification," IEICE Trans. Fundamentals, vol.E85-A, no.12, pp.2596-2602, Nov. 2002.

[7] T. Yoda and A. Takahashi, "Clock period minimization of semi-synchronous circuits by gate-level delay insertion," IEICE Trans. Fundamentals, vol.E82-A, no.11, pp.2383-2389, Nov. 1999.

[8] Y. Kohira and A. Takahashi, "Clock Period Minimization Method of Semi-Synchronous Circuits by Delay Insertion," IEICE Trans. Fundamentals, vol.E88-A, no.4, pp.892-898, 2005.

[9] B. Taskin and I.S. Kourtev, "Delay Insertion Method in Clock Scheduling," IEEE trans. CAD, vol.25, no.4, pp.651-663, 2006.

[10] T. Yasui, K. Kurokawa, M. Toyonaga, and A. Takahashi, "A circuit optimization method by the register path modification in consideration of the range of feasible clock timing," DA Symposium 2002, pp.259-264, 2002. In Japanese.

[11] B.A. Rosdi and A. Takahashi, "Low Area Pipelined Circuits by Multi-clock Cycle Path and Clock Scheduling," ASP-DAC 2006, pp.260-265, 2006.

[12] X. Liu and M.C. Papaefthymiou, "Retiming and Clock Scheduling for Digital Circuit Optimization," IEEE trans. CAD, vol.21, no.2, pp.184-203, 2002.

[13] C.E. Leiserson and J.B. Saxe, "Retiming Synchronous Circuitry," Algorithmica, vol.6, no.1, pp.5-35, 1991.

[14] Y. Kohira and A. Takahashi, "Clock Period Minimization Method of Semi-Synchronous Circuits by Register Relocation," 19th Workshop on Circuits and Sustems in Karuizawa, pp.259-264, 2006. (In Japanese).

[15] ILOG, CPLEX, http://www.ilog.com/.