

論文 / 著書情報
Article / Book Information

Title	Gate-Level Register Relocation in Generalized Synchronous Framework for Clock Period Minimization
Authors	Yukihide Kohira, Atsushi Takahashi
Citation	IEICE Trans. Fundamentals, Vol. E90-A, No. 4, pp. 800-807
Pub. date	2007, 4
URL	http://search.ieice.org/
Copyright	(c) 2007 Institute of Electronics, Information and Communication Engineers

Gate-Level Register Relocation in Generalized Synchronous Framework for Clock Period Minimization*

Yukihide KOHIRA^{†a)}, Student Member and Atsushi TAKAHASHI[†], Member

SUMMARY Under the assumption that clock can be inputted to each register at an arbitrary timing, the minimum feasible clock period can be determined if delays between registers are given. This minimum feasible clock period might be reduced by register relocation maintaining the circuit behavior and topology. In this paper, we propose a gate-level register relocation method to reduce the minimum feasible clock period. The proposed method is a greedy local circuit modification method. We prove that the proposed method achieves the clock period achieved by retiming with delay decomposition, if the delay of each element in the circuit is unique. Experiments show that the computation time of the proposed method and the number of registers of a circuit obtained by the proposed method are smaller than those obtained by the retiming method in the conventional synchronous framework.

key words: register relocation, retiming, clock period minimization, generalized synchronous framework

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 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) [3]–[6], 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 the clock frequency, clock distribution circuit size, peak 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 [3]–[6] and clock distribution circuit synthesis algorithms [7],

[8] for given logic circuits were proposed. However, given logic circuits are synthesized for c-frame. In order to improve the clock period in c-frame, a circuit is synthesized so that the maximum delay between registers is as small as possible. However, in g-frame, the clock period might not be reduced 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 [9]–[11], a gate sizing method [12], a multi-clock cycle path method [13], and a register relocation method [14] are proposed.

In c-frame, the circuit modification in which registers are relocated while maintaining the circuit behavior and topology is called retiming [15]. 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 [14], 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 this paper, we propose a gate-level register relocation method in g-frame for the clock period minimization. The proposed method is a greedy local circuit modification method in order to improve the minimum clock period in g-frame. It is known that retiming with delay decomposition achieves a lower bound of the minimum feasible clock period in c-frame. We prove that the proposed method achieves this lower bound without delay decomposition and changing the timing of I/O. Moreover, experiments show that the computation time of the proposed method and the number of registers of circuits obtained by the proposed method are smaller than those obtained by the retiming method in c-frame [15] in most circuits.

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 *circuit graph* $G = (V_g, E_g)$, where V_g is the vertex set corresponding to elements in the circuit. V_g consists of two vertex sets V_d and

Manuscript received June 26, 2006.

Manuscript revised October 24, 2006.

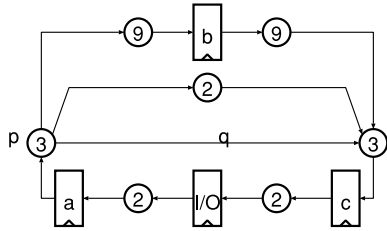
Final manuscript received December 15, 2006.

[†]The authors are with the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

*The preliminary version was presented at [1], [2].

a) E-mail: kohira@lab.ss.titech.ac.jp

DOI: 10.1093/ietfec/e90–a.4.800


 Fig. 1 Circuit graph G .

V_r . Delay vertex set V_d corresponds to elements with delay. Register vertex set V_r corresponds to registers. In this paper, we assume that each element has a unique and non-negative delay. Let $d(v)$ be the weight of $v \in V_d$ which corresponds to the delay of the corresponding element, and $d(v) = 0$ where $v \in V_r$. In this paper, we assume that the delay, setup time, and hold time of a register are 0 for simplicity. If they are not 0, they can be represented by using delay vertices. E_g is the directed edge set corresponding to signal propagations in the circuit. Let $d(e) = d(u)$, where $e = (u, v) \in E_g$. This means that an edge weight is equal to the weight of the head vertex of the edge. Let $D(P) = \sum_{k=1}^i d(e_k)$, where $P = (e_1, e_2, \dots, e_i)(e_1, e_2, \dots, e_i \in E_g)$ is a path in G . Let $D_{\max}(a, b)$ be the maximum delay from a vertex a to a vertex b without going through registers. A path P from a to b through no register is said to be a D_{\max} path from a to b if $D(P) = D_{\max}(a, b)$. Similarly, D_{\min} path is defined. The cycle weight $D(C)$ of a cycle C in G is defined as the sum of edge weights on C .

An example of a circuit graph is shown in Fig. 1. In the circuit graph shown in Fig. 1, $\{a, b, c, I/O\}$ is the register vertex set, and the figure in each delay vertex represents its weight.

2.1 Generalized Synchronous Framework

In the *complete-synchronous framework* (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 the *generalized synchronous framework* (g-frame). In g-frame [3]–[6], the clock timing of a register may be different from other registers. The clock timing $S(r)$ of a 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. 2) [3].

Setup (No-Zero-Clocking) Constraints

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

Hold (No-Double-Clocking) Constraints

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

Since c-frame has the premise that a clock ticks all the

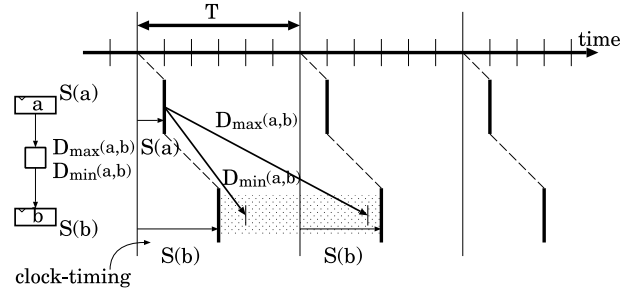


Fig. 2 Timing chart.

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 . $T_S(G)$ is determined by the *constraint graph* $H(V_r(G), E_r(G))$ for G defined as follows. The vertex set $V_r(G)$ corresponds to register vertex set V_r in G . The directed edge set $E_r(G)$ corresponds to two types of constraints. An edge from a register a to a register b with weight $D_{\min}(a, b)$, called 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 Z-edge, corresponds to the Setup constraint. Hereafter, we simply refer to $H(V_r(G), E_r(G))$ as $H(G)$. The weights of Z-edges are the functions of the clock period T . Let $H(G, t)$ be the constraint graph $H(G)$ in which the clock period T is set to t . Let the weight $W(P)$ of a directed path P in $H(G)$ be the sum of edge weights on P and the weight $W(C)$ of a directed cycle C in $H(G)$ be the sum of edge weights on C . We refer to a cycle C whose weight $W(C)$ is positive, 0, and negative as *positive-cycle*, *zero-cycle*, and *negative-cycle*, respectively. It is known that $T_S(G)$ is determined as in the following theorem.

Theorem 1 ([5]): $T_S(G)$ is the minimum t such that there is no negative-cycle in the constraint graph $H(G, t)$.

A cycle C that is zero in $H(G, T_S(G))$ and is negative in $H(G, t)$ where $t < T_S(G)$ determines the minimum clock period $T_S(G)$ in g-frame. Therefore, we call such cycle C *critical-cycle*. In this paper, since we assume that each element has a non-negative delay, no critical-cycle consists of only D-edges.

For example, $D_{\max}(a, b)$ in G shown in Fig. 1 is 12, which is the maximum delay between registers. Therefore, the minimum clock period in c-frame is 12. The constraint graph $H(G)$ for G shown in Fig. 1 is shown in Fig. 3(a). Since $H(G, 9)$ shown in Fig. 3(b) includes no negative-cycle, the weight of cycle (a, c, b, a) in $H(G, 9)$ is zero, and that in $H(G, t)$ where $t < 9$ is negative, then cycle (a, c, b, a) is critical and $T_S(G)$ is 9. In the following, Z-edge, D-edge, and

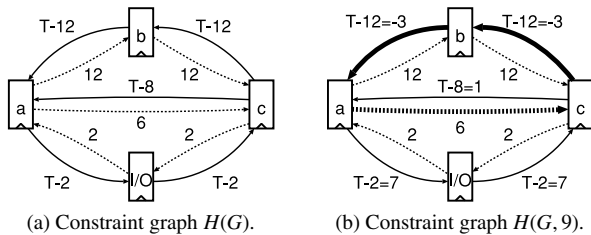


Fig. 3 Constraint graph.

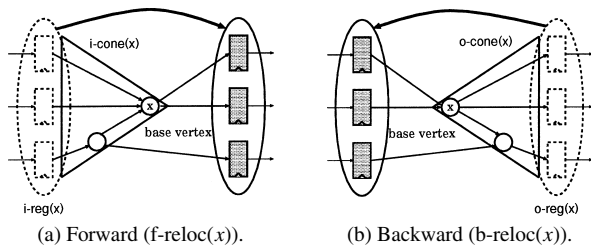


Fig. 4 Cone relocation.

edges in critical-cycles of a constraint graph are drawn by solid lines, dotted lines, and bold lines in figures, respectively.

If the clock timings of some registers are requested to be equal, such constraints can be represented by contracting the corresponding vertices into one vertex in the constraint graph. In this paper, inputs and outputs of a circuit are contracted into one vertex (*I/O register*) in the constraint graph because input and output timings of circuits are assumed to be equal in general.

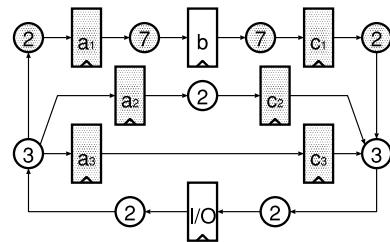
2.2 Register Relocation

Register relocation method is a circuit modification method in g-frame. We propose *cone relocation* which is a kind of register relocation.

Let $i\text{-cone}(x)$, the input cone of a delay 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 $i\text{-reg}(x)$ be the set of input registers of $i\text{-cone}(x)$. An edge (u, v) in G is called an output of the $i\text{-cone}(x)$ if u is in $i\text{-cone}(x)$ and v is not in $i\text{-cone}(x)$.

The *forward cone relocation* of a delay vertex x (f-reloc(x)) is a modification of G in which all $i\text{-reg}(x)$ are removed and which a register is inserted to each output of $i\text{-cone}(x)$ (Fig. 4(a)). Similarly, the *backward cone relocation* of x (b-reloc(x)) is defined (Fig. 4(b)). In f-reloc(x) or b-reloc(x), x is called the *base vertex*.

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 of a vertex [15] which we call *vertex relocation*. A cone relocation can be defined as the set of vertex relocations.

Fig. 5 Circuit graph G' obtained from G shown in Fig. 1 by retiming.

2.3 Retiming

Register relocation methods in c-frame are called *retiming* [15]. The number of registers in a cycle remains same by a register relocation. If registers can be relocated to any edge in G and if each element can be decomposed into two elements with arbitrary delays whose delay sum is equal to the delay of the decomposed element, then the minimum clock period achieved by retiming is the maximum of the delay sum over the number of registers of a cycle among cycles in G . This minimum clock period of G achieved by retiming is called the *lowest clock period* $T_L(G)$.

Definition 1 ([9]): The lowest clock period $T_L(G)$ is defined as

$$T_L(G) = \max_{C \in \text{cycles in } G} \frac{D(C)}{N(C)},$$

where $N(C)$ is the number of registers in a directed cycle C in G .

In G shown in Fig. 1, the lowest clock period $T_L(G)$ is 7, since the delay sum and the number of registers of cycle $(a, b, c, I/O, a)$ are 28 and 4, respectively. The circuit graph G' obtained from G shown in Fig. 1 by retiming is shown in Fig. 5. In the following, the decomposed delay elements and registers inserted by register relocation are drawn by shade in figures. The circuit corresponding to G' in c-frame achieves the lowest clock period since the maximum delay between registers is 7. While, the minimum clock period of G in g-frame is 9, which is larger than lowest clock period.

Although the retiming method achieves the lowest clock period, the assumption that each element can be decomposed into two elements with arbitrary delays is not practical. If the delay cannot be decomposed into two delays, retiming does not always obtain a circuit that achieves the lowest clock period.

3. Cone Relocation in Generalized Synchronous Framework

As mentioned in the previous section, g-frame does not always achieve the lowest clock period. On the other hand, the retiming method achieves the lowest clock period if an element can be decomposed, but does not always achieves the lowest clock period otherwise. Therefore, we propose

among vertices at which P_1 and P_2 merge (branch) is called junction vertex of r of C .

A cone relocation of a vertex x that relocates a register r is called a cone relocation of r . A D-Z register (Z-D register) in a critical-cycle C is called a target register of C . A cone relocation of a target register r of a critical-cycle C whose base vertex is a junction vertex of r of C is called a *target cone relocation* of r of C . Note that a junction vertex of r of C must be a delay vertex. C has a chance to become non critical if a target cone relocation of r of C is performed.

In f-reloc(x) or b-reloc(x), all registers in i-reg(x) or o-reg(x) are called *removed registers*. In this paper, the cone relocations such that removed registers contain I/O register are prohibited in order to keep the communication data from/to environment. A target register r is said to be *legal* if there is a target cone relocation of r of a critical-cycle C such that no I/O register is contained in removed registers by the target register relocation of r of C .

The critical graph of a constraint graph is the subgraph of the constraint graph that consists of edges in all critical-cycles. A critical-cycle C is said to be *maximal* if no critical-cycle is obtained from C by removing a target register r of C and adding a path connecting two removed registers by a cone relocation of r in the critical graph.

The proposed method is guaranteed to achieve the lowest clock period by applying a target cone relocation of a legal target register of a maximal critical critical-cycle iteratively. In the following, we will show that the proposed method achieves the lowest clock period.

3.2 Proposed Method

The proposed method is shown in the following.

Inputs : circuit G

Outputs : circuit obtained by target cone relocations

Step 1 : Determine $T_L(G)$ from the constraint graph consisting of Z-edges ([10]).

Step 2 : Determine $T_S(G)$ from the constraint graph. If $T_S(G) = T_L(G)$, then output circuit G and terminate.

Step 3 : Choose a legal target register r in a maximal critical-cycle C .

Step 4 : Obtain G by applying a target cone relocation of r of C and go to Step 2.

Hereinafter, we show the validity of the proposed method. At first, we show that there exists a target register in a critical-cycle.

Lemma 1 ([9]): If $T_S(G) > T_L(G)$, a critical-cycle contains at least one D-edge.

A critical-cycle contains at least one Z-edge from the assumption that no critical-cycle consists of only D-edges. From Lemma 1 and the fact mentioned above, a critical-cycle contains at least one Z-D register and D-Z register if $T_S(G) > T_L(G)$. So we have the following theorem.

Theorem 2: If $T_S(G) > T_L(G)$, a critical-cycle contains a target register.

If $T_S(G) = T_L(G)$, a critical-cycle which consists of only Z-edges exists in the constraint graph [9]. So, if $T_S(G) = T_L(G)$, the proposed method cannot improve the minimum clock period in g-frame since no target register exists on the critical-cycle.

Theorem 3: If a critical-cycle contains a target register, there exists a target cone relocation of the target register of the critical-cycle.

Proof. We will show that a junction vertex of the target register d of the critical-cycle C is a delay vertex. Let d be a D-Z register of $C = (a, d, b, \dots, a)$, x be a junction vertex of d . P_1 be a D_{\min} path from a to d via x , and P_2 be a D_{\max} path from b to d via x . If $a \neq b$, then x is neither a nor b but a delay vertex since internal vertices of P_1 and P_2 are not registers. Otherwise, P_1 and P_2 form a critical-cycle $C = (a, d, a)$. If x is register a , we have that $D_{\max}(a, d) = D_{\min}(a, d)$ since the delay of a vertex is unique and $P_1 = P_2$. Then, we have $W(C) = T$. But this fact contradicts the assumption that C is critical ($W(C) = 0$). Therefore, x is a delay vertex.

The case when d is a Z-D register can be proved similarly. \square

By using the following lemma, we will show that there exists a maximal critical-cycle in the critical graph.

Lemma 2: If a critical-cycle C_1 of $H(G)$ contains a target register d , a cycle C_2 obtained from C_1 by replacing d with f_1 is critical, where f_1 is a removed register by a target cone relocation of d of C_1 .

Proof. Assume that d is a D-Z register of $C_1 = (b, d, a, P_1, b)$ and x is a junction vertex of d of C_1 . f_1 is removed by b-reloc(x) (See Fig. 10). Since C_1 is critical, we have that

$$W(C_1) = W(P_1) + D_{\min}(b, d) + T_S(G) - D_{\max}(a, d) = 0.$$

Since x is a junction vertex and the delay of each element is unique, we have that $D_{\min}(x, d) = D_{\max}(x, d)$. So we have that

$$W(P_1) + D_{\min}(b, x) + T_S - D_{\max}(a, x) = 0. \quad (1)$$

Since $D_{\min}(x, f_1) \leq D_{\max}(x, f_1)$, we have that

$$W(C_2) = W(P_1) + D_{\min}(b, f_1) + T_S(G) - D_{\max}(a, f_1) \leq 0,$$

where cycle $C_2 = (f_1, a, P_1, b, f_1)$. Since the minimum clock

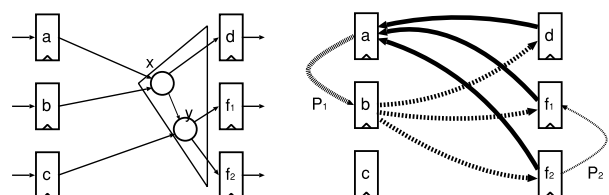


Fig. 10 A part of a circuit graph and a part of the constraint graph before b-reloc(x).

period is $T_S(G)$, the constraint graph $H(G, T_S(G))$ has no negative-cycle. So $W(C_2) = 0$ and C_2 is also critical.

The case when d is a Z-D register can be proved similarly. \square

Theorem 4: A maximal critical-cycle exists in a strongly connected component of the critical graph of $H(G)$.

Proof. If $T_S(G) = T_L(G)$, a critical-cycle which consists of only Z-edges exists in $H(G)$ [9]. Since a critical-cycle which consists of only Z-edges contains no target register, the critical-cycle is a maximal critical-cycle.

If $T_S(G) > T_L(G)$, a critical-cycle contains a target register from Theorem 2. Let d be a D-Z register of a critical-cycle $C_1 = (d, a, P_1, b, d)$ which is not maximal. Since C_1 is not maximal, without loss of generality, we assume that removed registers f_1 and f_2 by a target cone relocation of d of C_1 have a path $P_2 = (f_2, \dots, f_1)$ on the critical graph (Fig. 10). From Lemma 2, cycles (f_1, a, P_1, b, f_1) and (f_2, a, P_1, b, f_2) are also critical. Then cycle $(a, P_1, b, f_2, P_2, f_1, a)$ is critical and this critical-cycle has more registers than C_1 . Therefore, if a critical-cycle C is not maximal, then there exists a critical-cycle that has more registers than C . If a strongly connected component of the critical graph contains no maximal critical-cycle, then its size is infinite. But since the size of a strongly connected component is finite, a maximal critical-cycle exists in the strongly connected component.

The case when d is a Z-D register can be proved similarly. \square

A strongly connected component of the critical graph is found in linear time [6]. A maximal critical-cycle is found by the procedure shown in the proof in polynomial time.

Next, we show that there exists a legal target register on each critical-cycle. From Lemma 2 and Theorem 2, we can prove the following theorem.

Theorem 5: If $T_S(G) > T_L(G)$, a critical-cycle contains at least one legal target register.

Proof. From Theorem 2, a critical-cycle contains a target register if $T_S(G) > T_L(G)$. Actually, there are at least two target registers. We will show that at least one of two target registers of a critical-cycle such that the path from one to another which consists of only Z-edges is legal.

Assume that target registers a and b of a critical-cycle $C_1 = (a, P_1, b, \dots, a)$ are not legal, where path P_1 consists of only Z-edges. From Lemma 2, cycle $C_2 = (I/O, P_1, b, \dots, I/O)$ is also critical, since I/O register is relocated by a target register relocation of a of C_1 . Similarly, cycle $C_3 = (a, P_1, I/O, \dots, a)$ is also critical. Therefore, since C_1 , C_2 , and C_3 are contained by a strongly connected component of the critical graph, cycle $C_4 = (I/O, P_1, I/O)$ is critical. This contradicts Theorem 2, since C_4 is a critical-cycle which consists of only Z-edges. \square

At last, we show that the minimum clock period in g-frame can be improved by the lowest minimum clock period.

Theorem 6: If $T_S(G) > T_L(G)$, the number of zero-cycles

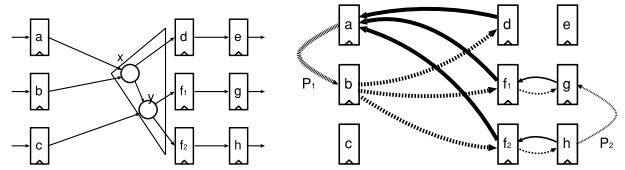


Fig. 11 A part of a circuit graph G_p and a part of the constraint graph $H(G_p)$ before b-reloc(x).

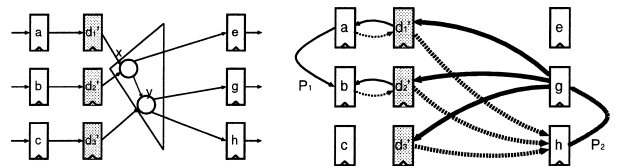


Fig. 12 A part of a circuit graph G'_p and a part of the constraint graph $H(G'_p)$ after b-reloc(x).

in the constraint graph $H(G, T_S(G))$ is reduced by a cone relocation of a legal target register in a maximal critical-cycle.

Proof. From above Theorems 4 and 5, a maximal critical-cycle contains at least one legal target register if $T_S(G) > T_L(G)$. Let d be legal and a D-Z register of a maximal critical-cycle $C_1 = (d, a, P_1, b, d)$ and x be a junction vertex of d of C_1 . Let $C_2 = (d'_3, h, P_2, g, d'_3)$ be a cycle in $H(G'_p)$ obtained by the b-reloc(x) (See Figs. 11 and 12). Assume that C_2 is critical and y is a junction vertex of d'_3 of C_2 . Since C_1 and C_2 are critical, similar to Eq. (1), we have that

$$W(P_1) + D_{\min}(b, x) + T_S - D_{\max}(a, x) = 0 \quad (2)$$

and

$$W(P_2) + D_{\min}(y, h) + T_S - D_{\max}(y, g) = 0. \quad (3)$$

Here, we focus on cycle $C = (a, P_1, b, f_2, h, P_2, g, f_1, a)$ in $H(G_p)$. $W(C)$ in $H(G_p, T_S)$ is as follows.

$$\begin{aligned} W(C) &= W(P_1) + D_{\min}(b, f_2) + D_{\min}(f_2, h) \\ &\quad + W(P_2) + T_S - D_{\max}(f_1, g) \\ &\quad + T_S - D_{\max}(a, f_1) \\ &= W(P_1) + D_{\min}(b, x) + T_S - D_{\max}(a, x) \\ &\quad + W(P_2) + D_{\min}(y, h) + T_S - D_{\max}(y, g) \\ &\quad + D_{\min}(x, y) - D_{\max}(x, y) \end{aligned}$$

From Eqs. (2) and (3), we have that $W(C) = 0$ and C is critical. This fact contradicts C_1 is maximal. Thus C_2 is not critical. The other type of non-critical-cycle also remains non-critical. Thus, no zero-cycle is formed by the proposed cone relocation method in the constraint graph $H(G'_p, T_S(G))$ after the cone relocation.

Since C_1 takes over a positive-cycle by a target cone relocation of d of C_1 , the number of zero-cycles in $H(G'_p, T_S(G))$ after a cone relocation is smaller than that in $H(G_p, T_S(G))$ before the target cone relocation.

The case of a Z-D register can be proved similarly. \square

Table 2 Results. Figures in () of the proposed method are achieved ratios of the retiming method by MILP [15]. The results of s13207, s15850, s38417 by the retiming method by MILP cannot be obtained because of the lack of memory, and that of s15850.1 cannot be obtained within a day.

model	original				retiming by MILP			proposed					
	#gate	T_C	T_S	#FF	T_R	#FF	time[s]	T_L	([%])	#FF	([%])	time[s]	([%])
$T_R > T_L$ (12 circuits)													
s344	160	37	34.0	15	20	27	0.12	19.00	(95.00)	26	(96.30)	0.06	(50.00)
s349	161	37	34.0	15	20	27	0.13	19.00	(95.00)	26	(96.30)	0.06	(46.15)
s382	158	18	12.0	21	12	29	0.14	11.25	(93.75)	25	(86.21)	0.03	(21.43)
s400	164	18	12.0	21	12	29	0.17	11.25	(93.75)	27	(93.10)	0.04	(23.53)
s444	181	20	13.0	21	13	29	0.33	11.67	(89.75)	35	(120.69)	0.09	(40.91)
s499	152	23	19.0	22	12	89	0.10	11.50	(95.83)	109	(122.47)	0.74	(740.00)
s635	286	162	158.0	32	89	63	1.57	88.50	(99.44)	76	(120.63)	2.10	(133.76)
s1269	569	70	61.0	37	40	123	5.88	39.34	(98.34)	90	(73.17)	3.10	(52.72)
s1512	780	54	43.0	57	41	72	25.72	40.50	(98.78)	61	(84.72)	0.11	(0.43)
s3271	1572	58	34.0	116	28	185	8.80	27.72	(98.98)	199	(107.57)	6.47	(73.52)
s3384	1685	168	154.0	183	76	183	111.10	75.50	(99.34)	292	(159.56)	0.04	(0.04)
s6669	3080	231	197.0	239	58	448	133.49	56.50	(97.41)	975	(217.63)	612.49	(458.83)
$T_R = T_L$ (10 circuits)													
s298	119	18	12.0	14	10	47	0.07	10.00	(100.00)	17	(36.17)	0.01	(14.29)
s526	193	18	12.0	21	11	63	0.26	11.00	(100.00)	22	(34.92)	0.01	(3.85)
s526n	194	18	12.0	21	11	63	0.25	11.00	(100.00)	22	(34.92)	0.01	(4.00)
s991	519	117	110.0	19	109	26	7.63	109.00	(100.00)	20	(76.92)	0.01	(0.13)
s1423	657	164	156.0	74	146	87	193.59	146.00	(100.00)	81	(93.10)	0.88	(0.45)
s3330	1789	66	40.0	133	32	123	7.34	32.00	(100.00)	147	(119.51)	1.00	(13.62)
s4863	2342	144	129.0	104	75	159	253.55	75.00	(100.00)	219	(137.74)	21.91	(8.64)
s9234	5597	107	72.0	228	63	263	445.65	63.00	(100.00)	240	(91.25)	2.37	(0.53)
s9234.1	5597	107	72.0	211	63	255	444.04	63.00	(100.00)	223	(87.45)	2.36	(0.53)
prolog	1601	68	40.0	136	31	144	6.04	31.00	(100.00)	154	(106.94)	0.97	(16.06)
T_R cannot be obtained. (4 circuits)													
s13207	7951	106	76.0	669	N.A.	N.A.	N.A.	75.00	(—)	670	(—)	1.43	(—)
s15850	9772	141	104.0	597	N.A.	N.A.	N.A.	78.00	(—)	643	(—)	41.46	(—)
s15850.1	9772	141	124.0	534	N.A.	N.A.	N.A.	103.00	(—)	544	(—)	16.59	(—)
s38417	22179	85	61.0	1636	N.A.	N.A.	N.A.	60.00	(—)	1638	(—)	9.74	(—)

By repeating a target cone relocation, all zero-cycles in $H(G, t)$, where t equals to the minimum clock period, become positive if the minimum clock period is larger than the lowest clock period, the minimum clock period is reduced, and the lowest clock period is achieved.

A legal target register on a maximal critical-cycle is chosen at Step 3 in the proposed method. Actually, the proposed method chooses the legal target register on a maximal critical-cycle so that the number of registers of the circuit obtained by the target cone relocation is small in order to minimize the number of registers of the obtained circuit.

4. Experiments

We implement retiming method in c-frame using a MILP formulation [15] and the proposed cone relocation method in g-frame in a PC with 3.06 GHz/512 K Intel Pentium-4 CPU, 512 MB RAM and gcc3.5.5 of C++. MILP is solved by CPLEX 9.0.0 [16]. We perform these methods on the IS-CAS89 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.

In 22 circuit among 48 ISCAS89 benchmark circuits, the lowest clock period $T_L(G)$ is equal to the minimum

clock period in g-frame $T_S(G)$. We do not apply the retiming method by MILP and the proposed method to these 22 circuits since they are optimal. We apply the retiming method by MILP and the proposed method to the other 26 circuits. Note that the retiming method by MILP cannot always achieve the lowest clock period since each delay element cannot be decomposed into two delays.

The results are shown in Table 2. T_C means the minimum clock period of the original circuit in c-frame. The results of s13207, s15850, and s38417 by the retiming method by MILP cannot be obtained because of the lack of memory, and that of s15850.1 cannot be obtained within a day. In 12 circuits among the other 22 circuits, the lowest clock period $T_L(G)$ achieved by the proposed method is smaller than the minimum clock period $T_R(G)$ achieved by the retiming method by MILP. In 6 circuits among these 12 circuits, the number of registers of circuits obtained by the proposed method is smaller than that by retiming method by MILP. In most circuits, the computation time of the proposed method is smaller than that of the retiming method by MILP.

Among 10 circuits such that $T_R(G) = T_L(G)$, the computation time of the proposed method is smaller than that of the retiming method by MILP in all circuits, and the number of registers of circuits obtained by the proposed method is smaller than that by the retiming method by MILP in 7 cir-

cuits. The number of registers of a circuit by the proposed method is larger than that by the retiming method by MILP in 3 circuits. This means that the proposed method is not optimal in terms of the number of registers. Since the proposed method is a greedy local circuit modification method in order to improve the minimum clock period in g-frame until the minimum clock period in g-frame is equal to the lowest clock period, the computation time and the number of registers become large if the amount of improvement from the initial minimum clock period is large.

Naturally, the results such as the number of improved circuits, the improved ratio of clock period and so on depend on the delay of each gate. For example, if routing and register delays are set to 0 and the other gate delays are set to 1, then in 28 circuit among 48 ISCAS89 benchmark circuits, the lowest clock period $T_L(G)$ is equal to the minimum clock period in g-frame $T_S(G)$. Moreover, in 10 circuits among the other 20 circuits, the lowest clock period $T_L(G)$ achieved by the proposed method is smaller than the minimum clock period $T_R(G)$ achieved by the retiming method by MILP.

5. Conclusions

In this paper, we propose a gate-level register relocation method in the generalized synchronous framework in order to achieve the lowest clock period without delay decomposition under the assumption that the delay of each element is unique. We prove that the proposed method achieves the lowest clock period. Moreover, experiments show that the computation time of the proposed method and the number of registers of circuits by the proposed method are smaller than those of the retiming method by MILP in most circuits.

In the future work, since the proposed method is a greedy local circuit modification method, it is not optimal in terms of the number of registers. So we try to improve the proposed method in order to improve the number of registers and the computation time. Moreover, we need to handle more practical delay model. Although the lowest clock period is not always achieved when we adopt more practical delay model, we think the basic idea of the proposed method can be applied even if we adopt more practical delay model.

References

- [1] E. Kamibayashi, Y. Kohira, and A. Takahashi, "Circuit modification method of semi-synchronous circuits with retiming," IEICE Technical Report, VLD2004-146, 2005.
- [2] Y. Kohira and A. Takahashi, "Clock period minimization method of semi-synchronous circuits by register relocation," 19th Workshop on Circuits and Systems in Karuizawa, pp.259–264, 2006.
- [3] J. Fishburn, "Clock skew optimization," IEEE Trans. Comput., vol.39, no.7, pp.945–951, 1990.
- [4] R.B. Deokar and S.S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," ISCAS, pp.407–410, 1994.
- [5] A. Takahashi and Y. Kajitani, "Performance and reliability driven clock scheduling of sequential logic circuits," ASP-DAC'97, pp.37–43, 1997.
- [6] A. Takahashi, "Practical fast clock-schedule design algorithms," IEICE Trans. Fundamentals, vol.E89-A, no.4, pp.1005–1011, April 2006.
- [7] 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, Nov. 2002.
- [8] 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, Dec. 2002.
- [9] 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.
- [10] 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, April 2005.
- [11] B. Taskin and I.S. Kourtev, "Delay insertion method in clock scheduling," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.25, no.4, pp.651–663, 2006.
- [12] 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.
- [13] 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.
- [14] X. Liu and M.C. Papaefthymiou, "Retiming and clock scheduling for digital circuit optimization," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.21, no.2, pp.184–203, 2002.
- [15] C.E. Leiserson and J.B. Saxe, "Retiming synchronous circuitry," Algorithmica, vol.6, no.1, pp.5–35, 1991.
- [16] ILOG, CPLEX, <http://www.ilog.com/>



Yukihide Kohira received his B.E. and M.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2003 and 2005, respectively. He is currently a D.E. student of Department of Communications and Integrated Systems in Tokyo Institute of Technology. His research interests are in VLSI design automation and combinatorial algorithms. He is a student member of IEEE.



Atsushi Takahashi received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He had been with the Tokyo Institute of Technology as a research associate from 1991 to 1997 and has been an associate professor since 1997. He visited University of California, Los Angeles, U.S.A., as a visiting scholar from 2001 to 2002. He is currently with Department of Communications and Integrated Systems, Graduate School of Science and Engineering, Tokyo Institute of Technology. His research interests are in VLSI layout design and combinatorial algorithms. He is a member of IEEE and IPSJ.