

論文 / 著書情報
Article / Book Information

Title	2-SAT Based Linear Time Optimum Two-Domain Clock Skew Scheduling in General-Synchronous Framework
Authors	Yukihide Kohira, Atsushi Takahashi
出典 / Citation	IEICE Trans. Fundamentals, Vol. E97-A, No. 12, pp. 2459-2466
発行日 / Pub. date	2014, 12
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 2014 Institute of Electronics, Information and Communication Engineers.

2-SAT Based Linear Time Optimum Two-Domain Clock Skew Scheduling in General-Synchronous Framework*

Yukihide KOHIRA^{†a)}, Member and Atsushi TAKAHASHI^{††}, Senior Member

SUMMARY Multi-domain clock skew scheduling in general-synchronous framework is an effective technique to improve the performance of sequential circuits by using practical clock distribution network. Although the upper bound of performance of a circuit increases as the number of clock domains increases in multi-domain clock skew scheduling, the improvement of the performance becomes smaller while the cost of clock distribution network increases much. In this paper, a linear time algorithm that finds an optimum two-domain clock skew schedule in general-synchronous framework is proposed. Experimental results on ISCAS89 benchmark circuits and artificial data show that optimum circuits are efficiently obtained by our method in short time.

key words: general-synchronous framework, multi-domain clock skew scheduling, two-domain clock skew scheduling, 2-SAT

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 conventional clock synchronous framework in which the simultaneous clock distribution to every register is assumed. We call the conventional clock synchronous framework *complete-synchronous framework (c-frame)*. The increases of the size and power consumption of a clock distribution circuit have become serious issues in c-frame. While, a clock synchronous framework [2]–[4], in which the clock is assumed to be distributed periodically to each register though not necessarily to all the registers simultaneously, is expected to give an essential solution. The clock synchronous framework without restriction of simultaneity has been discussed in the context of clock skew scheduling (CSS) [2], [3], useful-skew [5], semi-synchronous [6] and so on. In this paper, we call the clock synchronous framework *general-synchronous framework (g-frame)*. In g-frame, the quality of circuit such as the clock frequency, area, power consumption and peak power consumption is expected to be improved.

However, an unconstrained clock schedule with a large number of arbitrary clock delays cannot be realized reliably. Due to process variations, it is difficult to implement a clock

schedule with a large number of arbitrary clock delays. In practice, it is desirable that clock scheduling in g-frame is constrained to a limited number of clock delays. *Multi-domain clock skew scheduling (MDCSS)* has been proposed to meet this practical design requirement in [7]. Instead of assigning arbitrary number of clock delays, MDCSS restricts the number of feasible clock delays to a small number, called *clock domains*.

Many methods have been proposed to solve MDCSS problem. An algorithm based on simulated annealing has been proposed in [8], [9]. In [7], the authors formulated MDCSS problem as a mixed integer programming problem and solved it by a SAT-based algorithm. Although this method guarantees the optimality, its computational time is long because it takes long time to solve a SAT problem. In [10], a multi-level clustering algorithm has been proposed. The algorithm recursively merges half of the registers at each level until the total number of clusters is small enough. Compared to the work of [7], this algorithm is much faster. However, the algorithm is heuristic and it does not guarantee the optimality. An exact algorithm based on branch-and-bound search framework with greedy speeding up heuristics has been proposed in [11]. In [12], it is proved that MDCSS problem is NP-complete when the number of clock domains is $|V|/2$, where $|V|$ is the number of registers. Moreover, in [12], the algorithm has been proposed to obtain an optimum clock schedule in MDCSS problem. The time complexity of the algorithm proposed in [12] is $O((k-1)!|V||E|^k)$, where k is the number of clock domains, $|V|$ is the number of registers, and $|E|$ is the number of register pairs with signal propagations. It means that MDCSS problem can be solved in polynomial time if the number of clock domains is restricted to a small constant. Of course, the amount of improvement becomes smaller if the number of clock domains is restricted in g-frame. However, the performance of a circuit in g-frame is usually improved much compared to that in c-frame even if the number of clock domains is two. For example, in [13], it is shown that the power consumption of the clock tree in *two-domain clock skew scheduling (2DCSS)* is smaller than that in MDCSS and the clock period in 2DCSS is reduced by 10% compared with that in c-frame. Therefore, the fast 2DCSS algorithm is desired in practical circuit design to improve the circuit performance by a practical clock distribution network. However, the algorithm in [12] takes too much time since the time complexity of the algorithm in [12] is $O(|V||E|^2)$ when $k = 2$.

In this paper, we propose an optimum linear time algo-

Manuscript received March 17, 2014.

Manuscript revised June 30, 2014.

[†]The author is with the University of Aizu, Aizuwakamatsushi, 965-8580 Japan.

^{††}The author is with Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

*The preliminary version was presented at [1].

a) E-mail: kohira@u-aizu.ac.jp

DOI: 10.1587/transfun.E97.A.2459

algorithm that maximizes the performance under the constraint that the number of clock domains is restricted to two in g-frame. In our method, MDCSS problem is translated into 2-SAT problem. In our 2-SAT problem formulation, each variable corresponds to clock timing of each register and each set of four clauses corresponds to timing constraints for a register pair. Since 2-SAT problem formulation is obtained from MDCSS problem in $O(|V| + |E|)$ time in our proposed method and 2-SAT problem can be solved in $O(|V| + |E|)$ time [14], our proposed method is a linear time algorithm.

The contribution of this paper includes:

- We improve the time complexity of the algorithm for MDCSS problem where the number of clock domains is restricted to two in g-frame. The time complexity of our proposed method is $O(|V| + |E|)$ and it is faster than the existing method proposed in [12] whose time complexity is $O(|V||E|^2)$.
- Experimental results on ISCAS89 benchmarks and artificial data show the optimality and efficiency of our method.

The rest of the paper is organized as follows: in Sect. 2, we discuss g-frame and the formulation of MDCSS problem. We propose a linear time algorithm for MDCSS problem where the number of clock domains is restricted to two in Sect. 3. In Sect. 4, an improvement of the algorithm proposed in [12] is discussed to make a fair experimental comparison with the proposed method. Experimental results are presented and discussed in Sect. 5. The paper is concluded in Sect. 6.

2. Preliminaries

2.1 General-Synchronous Framework

A circuit in a clock synchronous framework works correctly with a clock period T if the following two types of constraints are satisfied for every register pair with signal propagations [2].

Setup (No-zero-clocking) constraints

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

Hold (No-double-clocking) constraints

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

where $S(a)$ ($S(b)$) is *clock timing* of a register a (b), which is defined as the difference in clock arrival time between a (b) and an arbitrarily chosen reference register, $D_{\max}(a, b)$ ($D_{\min}(a, b)$) is the maximum (minimum) delay from a to b (Fig. 1).

Since a clock ticks all registers simultaneously in c-frame, the clock period must be larger than or equal to the maximum of delays between register pairs. Let $T_C(G)$ be the minimum clock period of a circuit G in c-frame, which is equal to the maximum of delays between register pairs. On the other hand, in g-frame, circuits can work correctly

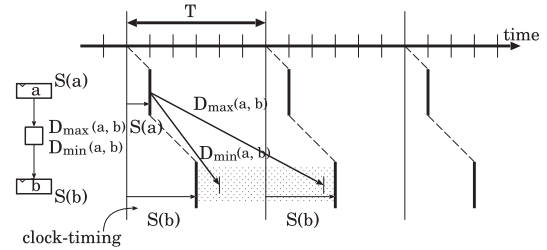


Fig. 1 Timing chart.

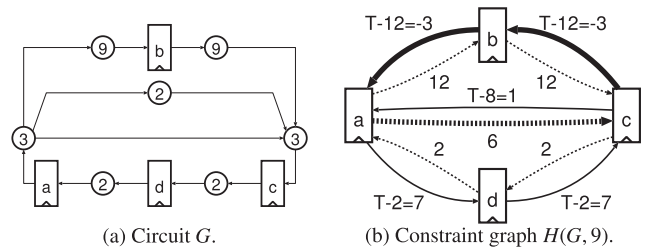


Fig. 2 An example of circuit and constraint graph.

with the clock period which is smaller than the maximum of delays between register pairs, if all register pairs with signal propagations satisfy two types of constraints.

Let $T_G(G)$ be the minimum clock period of a circuit G in g-frame under the assumption that the clock can be fed to each register at an arbitrarily designated timing. Hereafter, we simply call $T_G(G)$ the minimum clock period of G in g-frame. Note that $T_G(G) \leq T_C(G)$ since the clock timing of each register can also be set to the same in g-frame. $T_G(G)$ is determined by the *constraint graph* $H(G) = (V_r, E_r)$ for G , where vertex set V_r corresponds to registers in G and directed edge set E_r corresponds to two types of constraints [3], [4]. An edge in E_r 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 of Z-edges in $H(G)$ is set to t . Let the weight of a directed cycle in $H(G, t)$ be the sum of edge weights on the directed cycle. It is known that the minimum clock period $T_G(G)$ is the minimum t such that there is no cycle with negative weight in the constraint graph $H(G, t)$ [3], [4].

For example, the constraint graph $H(G, 9)$ of the circuit G shown in Fig. 2(a) is shown in Fig. 2(b). In Fig. 2(b), solid (dashed) lines correspond to Z-edges (D-edges). The constraint graph $H(G, T)$ has no cycle with negative weight when $T \geq 9$ and the weight of cycle (a, c, b, a) represented by bold lines in Fig. 2(b) is negative when $T < 9$. Therefore, the minimum clock period $T_G(G) = 9$.

2.2 Problem Definition

Given the number k of clock domains, the objective of MD-CSS is to decide the k domain values $\{s_1, s_2, \dots, s_k\}$ as well

as to assign each register one domain value such that the clock period T is minimized while the setup and hold constraints are satisfied. Since the clock timing is defined by the difference from that of an arbitrarily chosen reference register, without loss of generality, we assume that $s_1 = 0$ and $s_i \leq s_{i+1}$ ($1 \leq i \leq k-1$) in the rest of the paper. MDCSS problem is formally formulated as follows:

MDCSS

Input: maximum and minimum delays between register pairs $D_{\max}(a, b)$ and $D_{\min}(a, b)$ for $\forall(a, b) \in E_r$
Output: minimum clock period T_k and a clock schedule $S : \forall a \rightarrow \{s_1 (= 0), s_2, \dots, s_k\}$
Constraint: Satisfy hold and setup constraints.

We focus on 2DCSS problem, in which the number of clock domains is restricted to two in g-frame. 2DCSS problem is defined as follows:

2DCSS

Input: maximum and minimum delays between register pairs $D_{\max}(a, b)$ and $D_{\min}(a, b)$ for $\forall(a, b) \in E_r$
Output: minimum clock period T_2 and a clock schedule $S : \forall a \rightarrow \{s_1 (= 0), s_2 (\geq 0)\}$
Constraint: Satisfy hold and setup constraints.

Moreover, the decision version of 2DCSS problem is defined as follows:

Decision problem of 2DCSS

Input: maximum and minimum delays between register pairs $D_{\max}(a, b)$ and $D_{\min}(a, b)$ for $\forall(a, b) \in E_r$, clock period T and clock timing $s_2 (\geq 0)$
Question: Does a clock schedule $S : \forall a \rightarrow \{s_1 (= 0), s_2 (\geq 0)\}$ exist?
Constraint: Satisfy hold and setup constraints.

2.3 2-SAT

In our method, the decision problem of 2DCSS problem is translated into 2-SAT problem. 2-SAT problem is the problem of determining whether a collection of clauses with two-valued variables can be assigned values satisfying all the clauses. 2-SAT problem is often described using a Boolean expression with a conjunction of disjunctions, where each disjunction has two literals that are either variables or the negations of variables. Hereinafter, we also use Boolean expressions. 2-SAT problem and value assignment of 2-SAT problem are defined as follows:

Step1 $T_U = T_C$, $T_L = \max\{\max_{(a,a) \in E_r}\{D_{\max}(a, a)\}, \max_{(a,b) \in E_r}\{(D_{\max}(a, b) - D_{\min}(a, b))\}\}$.
Step2 **while** $(T_U - T_L > \Delta T)$ **do**
 Solve the decision problem of 2DCSS problem with $T = \frac{T_U + T_L}{2}$ and $s_2 = \max\{0, D_{\min}^-, T_C - T\}$ by corresponding 2-SAT problem.
 if “yes” **then** $T_U = \frac{T_U + T_L}{2}$.
 else $T_L = \frac{T_U + T_L}{2}$.
end while
Step3 Determine the clock timing of each register with $T = T_U$ and $s_2 = \max\{0, D_{\min}^-, T - T_C\}$ by corresponding value assignment of 2-SAT problem.

Fig. 3 Outline of the proposed method.

2-SAT

Input: a set of Boolean variables $X = \{x_1, x_2, \dots, x_n\}$ and a collection of clauses $C(X) = \bigwedge_{i=1}^m c_i(X)$ (Each clause $c_i(X)$ has two literals.)
Question: Does a satisfying value assignment $t : X \rightarrow \{0, 1\}$ exist?

Value assignment of 2-SAT

Input: a set of Boolean variables $X = \{x_1, x_2, \dots, x_n\}$ and a collection of clauses $C(X) = \bigwedge_{i=1}^m c_i(X)$ (Each clause $c_i(X)$ has two literals.)
Output: a satisfying value assignment $t : X \rightarrow \{0, 1\}$ if exists

It is known that 2-SAT problem and value assignment of 2-SAT problem can be solved in $O(n + m)$ time, where n is the number of variables and m is the number of clauses [14].

3. Proposed Method

3.1 Outline of Our Method

The outline of the proposed method is shown in Fig. 3.

The proposed method determines the minimum clock period by a binary search on the clock period. The upper bound of the minimum clock period is given by the minimum clock period in c-frame and it corresponds to the maximum of delays between register pairs. The lower bound of the minimum clock period is given by the delay from a register to the same register and the difference of the maximum and minimum delays between a register pair. They are derived from setup and hold constraints and they are obtained in $O(|E_r|)$ time. In Fig. 3, the precision for the clock period is denoted by ΔT . If the decision problem of 2DCSS problem can be solved in $O(X)$ time, 2DCSS problem can be solved in $O(X \cdot \log \frac{T}{\Delta T})$ by a binary search on the clock period T . Since the range of clock period can be regarded as a constant, 2DCSS problem can be solved in $O(X)$. In our method, the decision problem of 2DCSS problem is trans-

lated into 2-SAT problem in $O(|V_r| + |E_r|)$ time and 2-SAT problem can be solved in $O(|V_r| + |E_r|)$ time. Therefore, our method solves 2DCSS problem in $O(|V_r| + |E_r|)$ time.

In the proposed method, s_2 is set to $\max\{0, D_{\min}^-, T_C - T\}$, where $D_{\min}^- = -\min_{(a,b) \in E_r} \{0, D_{\min}(a, b)\}$. If there are D-edges with negative weights, $D_{\min}^- > 0$. Otherwise, $D_{\min}^- = 0$. Domain values $\{s_1, s_2, \dots, s_k\}$ in MDCSS problem can be determined by the shortest path in the constraint graph for k clock domains [12]. In this paper, the number of clock domains is restricted to two. In this case, the shortest path in the constraint graph for two clock domains depends on the minimum negative edge in the constraint graph. In the constraint graph, weights of D-edges with negative weights and those of Z-edges whose maximum delays $D_{\max}$ are larger than T are negative. Therefore, the shortest path in the constraint graph for two clock domains is determined by $\min\{0, -D_{\min}^-, T - T_C\}$. Since we assume that $s_2 \geq 0$, we have $s_2 = \max\{0, D_{\min}^-, T_C - T\}$.

Theorem 1: If a feasible clock schedule exists at the clock period T , there is a feasible clock schedule with $s_2 = \max\{0, D_{\min}^-, T_C - T\}$ in 2DCSS, where $D_{\min}^- = -\min_{(a,b) \in E_r} \{0, D_{\min}(a, b)\}$.

3.2 Translation from 2DCSS to 2-SAT

In this sub-section, we discuss the translation from the decision problem of 2DCSS problem to 2-SAT problem.

A Boolean variable x_a in 2-SAT problem corresponds to clock timing $S(a)$ of a register a . $x_a = 0$ ($x_a = 1$), if and only if $S(a) = 0$ ($S(a) = s_2$).

Each set of four clauses corresponds to timing constraints of a register pair. A collection of clauses $C(X)$ is defined as follows:

$$C(X) = \bigwedge_{(a,b) \in E_r} (c_0^{(a,b)} \wedge c_1^{(a,b)} \wedge c_2^{(a,b)} \wedge c_3^{(a,b)}).$$

The definition of each clause is shown in Table 1. Each clause is defined by the timing constraints of a register pair. For example, if hold and setup constraints of a register pair (a, b) are satisfied when $(S(a), S(b)) = (0, 0)$, $c_0^{(a,b)} = 1$. Otherwise, $c_0^{(a,b)} = x_a \vee x_b$.

If $(x_a, x_b) = (0, 0)$ and if hold and/or setup constraints of a register pair (a, b) are violated when $(S(a), S(b)) = (0, 0)$, $c_0^{(a,b)} = x_a \vee x_b = 0$. It means that $(x_a, x_b) = (0, 0)$ is not a solution of value assignment of 2-SAT problem when hold and/or setup constraints of a register pair (a, b) are violated and when $(S(a), S(b)) = (0, 0)$. On the other hand, $c_0^{(a,b)} = x_a \vee x_b = 1$ when $(x_a, x_b) = (0, 1)$, $(1, 0)$ and $(1, 1)$. It means that even if hold and/or setup constraints of a register pair (a, b) are violated when $(S(a), S(b)) = (0, 0)$, $(x_a, x_b) = (0, 1)$, $(1, 0)$ and $(1, 1)$ are allowed as solutions of value assignment of 2-SAT problem.

If a satisfying value assignment exists, a clock schedule can be obtained by the satisfying value assignment. From the definition of 2-SAT problem, if the value assignment satisfies all clauses, the corresponding clock schedule satisfies

Table 1 Definition of clause.

x_a	x_b	$S(a)$	$S(b)$	timing constraint for (a, b)	
				satisfied	violated
0	0	0	0	$c_0^{(a,b)} = 1$	$c_0^{(a,b)} = x_a \vee x_b$
0	1	0	s_2	$c_1^{(a,b)} = 1$	$c_1^{(a,b)} = x_a \vee \overline{x_b}$
1	0	s_2	0	$c_2^{(a,b)} = 1$	$c_2^{(a,b)} = \overline{x_a} \vee x_b$
1	1	s_2	s_2	$c_3^{(a,b)} = 1$	$c_3^{(a,b)} = \overline{x_a} \vee \overline{x_b}$

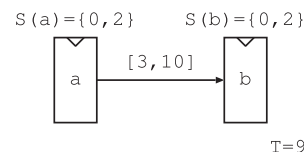


Fig. 4 An example which has a feasible clock schedule $(S(a), S(b)) = (0, 2)$.

the timing constraints.

For example, suppose the circuit such that $D_{\min}(a, b) = 3$, $D_{\max}(a, b) = 10$, $s_2 = 2$ and clock period $T = 9$ as shown in Fig. 4. In this case, setup constraints are violated when $(S(a), S(b)) = (0, 0)$, $(2, 0)$ and $(2, 2)$. Consequently, we have

$$\begin{aligned} C(X) &= c_0^{(a,b)} \wedge c_1^{(a,b)} \wedge c_2^{(a,b)} \wedge c_3^{(a,b)} \\ &= (x_a \vee x_b) \wedge (\overline{x_a} \vee x_b) \wedge (\overline{x_a} \vee \overline{x_b}). \end{aligned}$$

Since this collection is satisfied only if $(x_a, x_b) = (0, 1)$, the feasible clock schedule $(S(a), S(b)) = (0, 2)$ is obtained by solving the value assignment of 2-SAT.

The number of Boolean variables $|X|$ in 2-SAT problem is equal to the number of registers $|V_r|$. The number of clauses in 2-SAT problem is at most $4 \cdot |E_r|$. Moreover, since the time complexity of check of the timing constraint for a register pair is constant, 2DCSS problem can be translated into 2-SAT problem in $O(|V_r| + |E_r|)$ time and 2-SAT problem can be solved in $O(|V_r| + |E_r|)$ time. Therefore, 2DCSS problem can be solved in $O(|V_r| + |E_r|)$ time. The algorithm for 2-SAT problem proposed in [14] and examples are described in Appendix.

Here, the difference of the SAT formulation of our method and that proposed in [7] is discussed. In our proposed method, a Boolean variable corresponds to the clock timing for each register. Therefore, the number of Boolean variables $|X|$ in 2-SAT problem is equal to the number of registers $|V_r|$. On the other hand, in the SAT formulation proposed in [7], a Boolean variable is defined for each pair between a clock domain and a register and it represents an assignment of the clock domain for each register. Therefore, the number of Boolean variables $|X|$ in the SAT formulation proposed in [7] is equal to $k \cdot |V_r|$, where k is the number of clock domains. In the SAT formulation, since a disjunction has more than two literals, the formulation is SAT problem but not 2-SAT problem. Since SAT problem is NP-hard, its computational time cannot be expected to be short.

3.3 Enhancement of Our Method

The proposed method can be enhanced for the following

generalized 2DCSS problem easily by changing 0 in $S(a)$, s_2 in $S(a)$, 0 in $S(b)$ and s_2 in $S(b)$ to s_1^a , s_2^a , s_1^b and s_2^b in Table 1, respectively. The proposed method can be applied to various problems such as a 2DCSS problem with large delay variations and a post-silicon skew tuning problem with two values.

Generalized 2DCSS

Input: maximum and minimum delays between register pairs $D_{\max}(a, b)$ and $D_{\min}(a, b)$ for $\forall(a, b) \in E$ and clock period T

Output: a clock schedule $S : \forall a \rightarrow \{s_1^a, s_2^a\}$ if exists

Constraint: Satisfy hold and setup constraints.

4. Improved Method of Existing Method

Since the algorithm proposed in [12] focuses on MDCSS problem, it is not efficient for 2DCSS. To make a fair comparison with the proposed method, an enhancement of the algorithm proposed in [12] for 2DCSS is discussed.

The time complexity of the existing method is $O((k-1)!|V_r||E_r|^k)$. Since there are at most $O((k-1)!|E_r|^{k-1})$ candidates of domain values $\{s_1, s_2, \dots, s_k\}$ with k clock domains and the feasibility for each candidate can be checked by the modified Bellman-Ford (BF) algorithm [15] in $O(|V_r||E_r|)$ time, the time complexity of the existing method is $O((k-1)!|V_r||E_r|^k)$. Although the number of candidates whose feasibilities are checked is restricted by two pruning techniques in [12], its time complexity is still $O((k-1)!|V_r||E_r|^k)$. When $k = 2$, the time complexity of the existing method is $O(|V_r||E_r|^2)$.

As mentioned in Theorem 1, s_2 can be determined uniquely in 2DCSS problem. Consequently, the existing method can omit the enumeration of clock schedule candidates. We implemented the existing method omitting the enumeration and refer to this method as *improved BF method* hereafter. Note that the time complexity of the improved BF method is $O(|V_r||E_r|)$ time due to the modified BF algorithm.

5. Experimental Results

We implemented the proposed algorithm and the improved BF method in C++, which were compiled by gcc4.3.2, and executed on a PC with 6 GB memory by using Intel core i7-940 of 2.93 GHz. Note that only one core is used for our experiments. We obtained the same data of ISCAS89 benchmarks as those in [12] from the authors of [10], and used these benchmarks for comparisons among [12], improved BF method, and the proposed method. We also used artificial data and ISCAS89 benchmarks which are implemented in a commercial FPGA. However, since the number of registers and the number of register pairs with signal propagations in obtained s953 and s35932 are different from those shown in [10] and in [12], we ignore the results of these two

circuits.

The results of remaining 29 circuits in ISCAS89 benchmarks obtained from authors of [10], are shown in Table 2. In this experiment, primary inputs are regarded as one register and primary outputs are also regarded as another. The number of registers, the number of register pairs with signal propagations, the minimum clock period in g-frame and the minimum clock period in 2DCSS are represented by $|V_r|$, $|E_r|/2$, T_G and T_2 , respectively. The results of Existing [12] are directly copied from [12]. The results show that the minimum clock period obtained by our method is always the same as that obtained by the existing method and the improved BF method. Therefore, our method obtains an optimum solution as well as the existing method and the improved BF method. Although we cannot compare the execution time because the precisions of execution time shown in [12] are not clear, the execution time of the proposed method and the improved BF method is almost the same.

The results on artificial data are shown in Table 3. We made two types: one denoted by random is made randomly, and another denoted by worst is made so that the shortest path has $|V_r|$ vertices. In each type, each FF has ten signal propagations. If the shortest path has $|V_r|$ vertices, the execution time of the improved BF method is expected to be long since the number of iterations in the modified BF algorithm [15] is $|V_r|$. The results show that the execution time of the proposed method and the improved BF method is almost the same in random since the number of iterations in the modified BF algorithm is small. However, the proposed method is much faster than the improved BF method in the cases of worst examples. The results show the effectiveness of the proposed method.

Lastly, we apply the proposed method to ISCAS89 benchmarks implemented in a commercial FPGA. We use Xilinx Spartan3AN as a target device and Xilinx ISE Design Suite 12.4 as a CAD tool. In this experiment, primary inputs and primary outputs are ignored. In 8 circuits among 48 circuits in ISCAS89 benchmarks, the lack of memory occurred in the extraction of maximum and minimum delays between register pairs. Therefore, we apply the proposed method to the remaining 40 circuits. The results are shown in Table 4. In Table 4, the number of clock domains in g-frame at clock period T_G is represented by k_G . It is obtained by clustering method [16] to minimize the number of clock domains in the clock schedule with clock timing range which is determined by the clock scheduling method [17]. Note that k_G is not necessarily the minimum number of clock domains in g-frame at clock period T_G . The execution time of this experiment is almost the same as that in the first experiment. The minimum clock period obtained by our method is less than that in c-frame in all circuits. However, there is room of improvement since the minimum clock period obtained by our method is larger than that in g-frame. Since our method guarantees the optimality, the circuit obtained by our method achieves the minimum clock period in 2DCSS. Therefore, the number of clock domains must be increased to improve the minimum clock period.

Table 2 Result on ISCAS89 benchmarks.

Design	$ V_r $	$ E_r /2$	T_G	Existing [12]		Improved BF		Proposed	
				T_2/T_G	time[s]	T_2/T_G	time[s]	T_2/T_G	time[s]
s27	5	21	5.06	1.14	<0.1	1.14	<0.01	1.14	<0.01
s208	10	70	9.91	1.00	<0.1	1.00	<0.01	1.00	<0.01
s298	16	86	10.79	1.05	<0.1	1.05	<0.01	1.05	<0.01
s344	17	121	13.15	1.09	<0.1	1.09	<0.01	1.09	<0.01
s349	17	121	13.51	1.09	<0.1	1.09	<0.01	1.09	<0.01
s382	23	175	9.63	1.20	<0.1	1.20	<0.01	1.20	<0.01
s386	8	129	9.61	1.04	<0.1	1.04	<0.01	1.04	<0.01
s400	23	175	9.89	1.17	<0.1	1.17	<0.01	1.17	<0.01
s420	18	146	21.13	1.00	<0.1	1.00	<0.01	1.00	<0.01
s444	23	175	8.10	1.34	<0.1	1.34	<0.01	1.34	<0.01
s510	8	103	14.29	1.00	<0.1	1.00	<0.01	1.00	<0.01
s526	23	167	11.22	1.05	<0.1	1.05	<0.01	1.05	<0.01
s526n	23	167	11.31	1.05	<0.1	1.05	<0.01	1.05	<0.01
s641	21	486	29.51	1.00	<0.1	1.00	<0.01	1.00	<0.01
s713	21	486	30.58	1.00	<0.1	1.00	<0.01	1.00	<0.01
s820	7	213	16.74	1.00	<0.1	1.00	<0.01	1.00	<0.01
s832	7	213	16.22	1.00	<0.1	1.00	<0.01	1.00	<0.01
s838	34	298	44.66	1.00	<0.1	1.00	<0.01	1.00	<0.01
s1196	19	365	22.28	1.00	<0.1	1.00	<0.01	1.00	<0.01
s1238	19	365	24.33	1.00	<0.1	1.00	<0.01	1.00	<0.01
s1423	76	2235	73.13	1.03	<0.1	1.03	<0.01	1.03	<0.01
s1488	8	266	23.18	1.00	<0.1	1.00	<0.01	1.00	<0.01
s1494	8	266	23.85	1.00	<0.1	1.00	<0.01	1.00	<0.01
s5378	165	2180	22.88	1.13	<0.1	1.13	0.01	1.13	0.01
s9234	140	2226	33.76	1.01	<0.1	1.01	<0.01	1.01	<0.01
s13207	471	3885	53.36	1.03	<0.1	1.03	0.02	1.03	0.01
s15850	565	16375	85.27	1.08	<0.1	1.08	0.07	1.08	0.02
s38417	1465	31980	86.19	1.00	<0.1	1.00	0.01	1.00	0.01
s38584	1451	17900	286.62	1.00	<0.1	1.00	0.03	1.00	0.02

$|V_r|$ the number of registers
 $|E_r|/2$ the number of register pairs with signal propagations
 T_G the minimum clock period in g-frame
 T_2 the minimum clock period in 2DCSS

Table 3 Result on artificial data.

Type	$ V_r $	$ E_r /2$	Improved BF time[s]	Proposed time[s]
random	10000	100000	0.10	0.10
	20000	200000	0.18	0.15
	40000	400000	0.36	0.33
worst	10000	100000	19.71	0.18
	20000	200000	90.24	0.24
	40000	400000	367.97	0.51

6. Conclusion

In this paper, we proposed an optimum linear time algorithm that maximizes the performance under the constraint that the number of clock domains is restricted to two in general-synchronous framework. In our method, the problem is translated into 2-SAT problem. Experimental results on ISCAS89 benchmarks and artificial data confirmed the optimality and efficiency of our method.

In our future work, we will enhance our proposed algorithm to multi-domain clock skew scheduling in general-synchronous framework.

Acknowledgements

This work was partly supported by the Nakajima Foundation

and JSPS KAKENHI Grant-in-Aid for Young Scientists (B) 26730029.

References

- [1] Y. Kohira and A. Takahashi, "2-sat based linear time optimum two-domain clock skew scheduling," ASP-DAC, pp.173–178, 2014.
- [2] J.P. Fishburn, "Clock skew optimization," IEEE Trans. Comput., vol.39, no.7, pp.945–951, 1990.
- [3] R.B. Deoker and S.S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," ISCAS, pp.407–410, 1994.
- [4] A. Takahashi and Y. Kajitani, "Performance and reliability driven clock scheduling of sequential logic circuits," ASP-DAC, pp.37–43, 1997.
- [5] J. Xi and W. Dai, "Useful-skew clock skew routing with gate sizing for low power design," DAC, pp.383–388, 1996.
- [6] A. Takahashi, K. Inoue, and Y. Kajitani, "Clock routing driven layout methodology for semi-synchronous circuit design," ICCAD, pp.260–265, 1997.
- [7] K. Ravindran, A. Kuehlmann, and E. Sentovich, "Multi-domain clock skew scheduling," ICCAD, pp.801–808, 2003.
- [8] M. Toyonaga, K. Kurokawa, T. Yasui, and A. Takahashi, "A practical clock tree synthesis for semi-synchronous circuits," ISPD, pp.159–164, 2000.
- [9] K. Kurokawa, T. Yasui, M. Toyonaga, and A. Takahashi, "A practical clock tree synthesis for semi-synchronous circuits," IEICE Trans. Fundamentals, vol.E84-A, no.11, pp.2705–2713, Nov. 2001.
- [10] J. Casanova and J. Cortadella, "Multi-level clustering for clock skew optimization," ICCAD, pp.547–554, 2009.

Table 4 Experimental results in which the proposed method is applied to circuits implemented in Xilinx Spartan3AN.

Design			k_G	clock period [ps]					Proposed time[s]
	$ V_r $	$ E_r /2$		T_C	T_G	T_G/T_C	T_2	T_2/T_C	
s27	3	7	2	3451	3091	0.90	3091	0.90	<0.01
s208.1	8	36	6	4687	2383	0.51	3407	0.73	<0.01
s298	14	70	5	6185	4147	0.67	4825	0.78	<0.01
s344	15	89	3	6299	5187	0.82	5224	0.83	<0.01
s349	15	89	4	5976	4607	0.77	5140	0.86	<0.01
s382	21	146	6	5312	4759	0.90	5295	1.00	<0.01
s386	6	36	4	5490	5369	0.98	5451	0.99	<0.01
s400	21	146	7	5679	5122	0.90	5625	0.99	<0.01
s420.1	16	136	11	6640	3324	0.50	5382	0.81	<0.01
s444	21	146	7	5583	4564	0.82	5312	0.95	<0.01
s499	22	484	5	8824	8650	0.98	8733	0.99	<0.01
s510	6	36	2	7189	7158	1.00	7158	1.00	<0.01
s526	21	144	7	5868	4998	0.85	5775	0.98	<0.01
s526n	21	144	6	5824	5230	0.90	5789	0.99	<0.01
s635	32	528	16	11487	3672	0.32	9013	0.78	<0.01
s641	19	115	3	10526	9661	0.92	9793	0.93	<0.01
s713	19	115	3	10633	10082	0.95	10196	0.96	<0.01
s820	5	25	2	7818	7701	0.99	7701	0.99	<0.01
s832	5	25	3	7508	7473	1.00	7474	1.00	<0.01
s838.1	32	528	18	10039	3760	0.37	7681	0.77	<0.01
s938	32	528	20	9264	4170	0.45	7355	0.79	<0.01
s953	26	156	5	7540	7429	0.99	7536	1.00	<0.01
s967	26	156	2	7665	7646	1.00	7646	1.00	<0.01
s991	19	71	10	17399	4601	0.26	10089	0.58	<0.01
s1196	12	20	5	9157	4128	0.45	6150	0.67	<0.01
s1238	12	20	7	9036	2839	0.31	6392	0.71	<0.01
s1269	37	288	12	14908	10270	0.69	12209	0.82	<0.01
s1423	74	1764	5	20431	16739	0.82	18266	0.89	<0.01
s1488	6	36	3	8762	8719	1.00	8743	1.00	<0.01
s1494	6	36	4	8847	8642	0.98	8794	0.99	<0.01
s1512	57	513	3	8953	8579	0.96	8619	0.96	<0.01
prolog	132	549	13	11522	8343	0.72	10726	0.93	<0.01
s3271	116	896	37	11952	5934	0.50	9740	0.81	0.01
s3330	132	549	12	10122	7341	0.73	8408	0.83	<0.01
s3384	183	1827	5	18863	15923	0.84	16589	0.88	0.01
s5378	178	1144	34	8432	6452	0.77	8339	0.99	<0.01
s9234	160	2000	10	11983	9810	0.82	10552	0.88	<0.01
s9234.1	143	1884	12	11593	9478	0.82	10912	0.94	<0.01
s13207.1	621	3330	8	14755	11335	0.77	12592	0.85	0.03
s38584	1434	15387	4	16718	15343	0.92	15582	0.93	0.07
ave.					0.77		0.89		

$|V_r|$ the number of registers
 $|E_r|/2$ the number of register pairs with signal propagations
 k_G the number of clock domains in g-frame at clock period T_G
 T_C the minimum clock period in c-frame
 T_G the minimum clock period in g-frame
 T_2 the minimum clock period in 2DCSS

- [11] Y. Zhi, H. Zhou, and X. Zeng, "A practical method for multi-domain clock skew optimization," ASP-DAC, pp.521–526, 2011.
- [12] L. Li, Y. Lu, and H. Zhou, "Optimal multi-domain clock skew scheduling," DAC, pp.152–157, 2011.
- [13] Y. Kohira and A. Takahashi, "An evaluation of clock tree based on clustering in general-synchronous framework," Proc. Fundamentals Conf. 2010 IEICE Soc., A-3-1, Sept. 2010.
- [14] B. Aspvall, M.F. Plass, and R.E. Tarjan, "A linear-time algorithm for testing the truth of certain quantified boolean formulas," Inf. Process. Lett., vol.8, no.3, pp.121–123, 1979.
- [15] J.P. Fishburn, "Solving a system of difference constraints with variables restricted to a finite set," Inf. Process. Lett., vol.82, no.3, pp.143–144, 2002.
- [16] Y. Kohira and A. Takahashi, "Optimal register merging method after register relocation in semi-synchronous," SASIMI, pp.134–140,

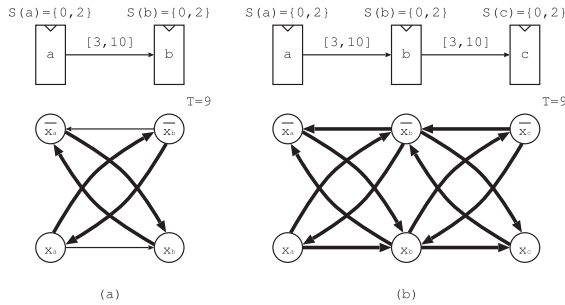
- 2006.
- [17] A. Takahashi, "Practical fast clock-schedule design algorithms," IEICE Trans. Fundamentals, vol.E89-A, no.4, pp.1005–1011, April 2006.

Appendix

2-SAT problem can be solved by using the concept of the strongly connected components in graph theory [14]. A directed graph is said to be strongly connected if there is a path between all pairs of vertices. A strongly connected component in a directed graph is a maximal strongly connected subgraph. For each variable x_i , two vertices named x_i and

Table A·1 The relation of a clause for timing violation and directed edges added into the graph.

x_a	x_b	clause	directed edges
0	0	$c_0^{(a,b)} = x_a \vee x_b$	$(\overline{x_a}, x_b), (\overline{x_b}, x_a)$
0	1	$c_1^{(a,b)} = x_a \vee \overline{x_b}$	$(\overline{x_a}, \overline{x_b}), (x_b, x_a)$
1	0	$c_2^{(a,b)} = \overline{x_a} \vee x_b$	$(x_a, x_b), (\overline{x_b}, \overline{x_a})$
1	1	$c_3^{(a,b)} = \overline{x_a} \vee \overline{x_b}$	$(x_a, \overline{x_b}), (x_b, \overline{x_a})$

**Fig. A·1** Examples: (a) Feasible clock schedule exists. (b) Feasible clock schedule does not exist.

$\overline{x_i}$ are added in the graph. For each clause $(u \vee v)$, two directed edge $(\overline{u}, v)$ and $(\overline{v}, u)$ are added. The relation between a clause for timing violation and directed edges added into the graph is shown in Table A·1. 2-SAT problem is satisfiable if and only if no vertices x_i and $\overline{x_i}$ belong to the same strongly connected component in the graph.

At first, suppose the circuit such that $D_{\min}(a, b) = 3$, $D_{\max}(a, b) = 10$, $s_2 = 2$ and clock period $T = 9$, which is shown in shown in Fig. 4 and shown in the top of Fig. A·1(a) again. In this case, timing constraints are violated when $(S(a), S(b)) = (0, 0), (2, 0)$ and $(2, 2)$. Therefore, edges $(\overline{x_a}, x_b), (\overline{x_b}, x_a), (x_a, x_b), (\overline{x_b}, \overline{x_a}), (x_a, \overline{x_b})$ and $(x_b, \overline{x_a})$ are added into the graph (see the figure shown in the bottom of Fig. A·1(a)). In Fig. A·1(a), bold lines represent the edges in strongly connected components. In this case, both x_a and x_b belong to the different strongly connected components from $\overline{x_a}$ and $\overline{x_b}$ in the graph. Therefore, we can assign a satisfying value assignment of 2-SAT problem and we obtain a feasible clock schedule $(S(a), S(b)) = (0, 2)$.

Next, we consider the circuit shown in the top of Fig. A·1(b). In this case, all vertices belong to the same strongly connected component in the graph shown in the bottom of Fig. A·1(b). Therefore, no satisfying value assignment of 2-SAT problem exists.



of ACM, IEEE and IPSJ.

Yukihide Kohira received his B.E., M.E., and D.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2003, 2005, and 2007, respectively. He had been a researcher of Department of Communications and Integrated Systems in Tokyo Institute of Technology from 2007 to 2009. He is currently with the School of Computer Science and Engineering, the University of Aizu, as an associate professor. His research interests are in VLSI design automation and combinational algorithms. He is a member



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 as an associate professor from 1997 to 2009, and from 2012. He had been with the Osaka University as an associate professor from 2009 to 2012. He visited University of California, Los Angeles, U.S.A., as a visiting scholar from 2002 to 2003. He is currently with Department of Communications and Computer Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology, as an associate professor since 2013. His research interests are in VLSI layout design and combinational algorithms. He is a member of IEEE, ACM and IPSJ.