

Clock Period Minimization of Semi-Synchronous Circuits by Gate-Level Delay Insertion*

Tomoyuki YODA, Atsushi TAKAHASHI, and Yoji KAJITANI

Department of Electrical and Electronic Engineering,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
E-mail: {yodatomo, atushi, kajitani}@ss.titech.ac.jp

Abstract

A semi-synchronous circuit is a circuit in which every register is ticked by a clock periodically, but not necessarily simultaneously. A feature of semi-synchronous circuits is that the minimum delay between registers may be critical with respect to the clock period of the circuit. In this paper, we discuss a delay insertion method which makes such a semi-synchronous circuit faster. The maximum delay-to-register ratio of the cycles on the circuit gives a lower bound of the clock period. We show that this bound is achieved in the semi-synchronous framework by the proposed gate-level delay insertion method on the assumption that the delay of each element on the circuit is unique.

1 Introduction

Semi-synchronous circuits are expected to achieve a high-performance by removing the constraint of complete-synchronous circuits that every register is ticked by a clock simultaneously. Among various objectives in the synthesis of high-performance circuits, the clock period minimization is one of the most important subjects.

For given signal delays between registers, it is known that the minimum clock period in the semi-synchronous framework is obtained in polynomial time [2, 1, 6]. To achieve the optimal clock period, each register should be ticked by a clock at its own due clock-timing. A clock-tree synthesis algorithm that realizes a due clock-timing for each register was proposed in [5]. A clock-driven layout methodology that minimizes both the clock period and the clock-tree length was proposed in [7].

The purpose of these studies are in the improvement of the performance of a given circuit in the semi-synchronous framework. Although the performance of a circuit is improved more or less by these methods, a circuit should be synthesized taking account of the effect

of the semi-synchronous framework to make the best use of it. In the semi-synchronous framework, the increase of the minimum delay between registers may lead to the clock period minimization, which does not lead to the clock period minimization in the complete-synchronous framework.

In this paper, we discuss a delay insertion method which makes semi-synchronous circuits faster. The maximum delay-to-register ratio of the cycles on the circuit gives a lower bound of the clock period. We show that this bound is achieved in the semi-synchronous framework by the proposed gate-level delay insertion method if the delay of each element on the circuit is unique (That is, max delay = min delay for each element).

2 Preliminaries

In this paper, we consider a circuit consisting of *registers* and *gates*, and *wires* connecting them. Both registers and gates are referred to *elements*. A circuit is represented by the *circuit graph* G where a vertex $v \in V(G)$ represents an element and a directed edge $(u, v) \in E(G)$ does the signal propagation from the output of element u to the input of element v along the wire. The weight of an edge is the sum of the delay due to the corresponding wire and the delay due to the corresponding end element. We assume that each wire delay and element delay is unique. The circuit graph of the circuit shown in Fig. 1 is shown in Fig. 2. The clock-tree of the circuit is not represented in the circuit graph.

Let $V_r(G) = \{r_1, r_2, \dots, r_{n_r}\} \subset V(G)$ be the set of registers. A *register-path* from register r_i to register r_j on G is a directed path from r_i to r_j without other registers. Let $E_r(G)$ be the set of ordered register pairs (r_i, r_j) such that there is a register-path from r_i to r_j . A maximum (minimum) register-path from r_i to r_j is a register-path from r_i to r_j with the maximum (minimum) weight. The maximum (minimum) delay from r_i to r_j is the weight of a maximum (minimum) register-path from r_i to r_j , and denoted by $d_{\max}(r_i, r_j)$ ($d_{\min}(r_i, r_j)$). The delay between registers is not unique, because there are various paths between these registers, even if each wire delay and element delay is unique. For example, in

*This work is part of a project of CAD21 at Tokyo Institute of Technology, and partly supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan.

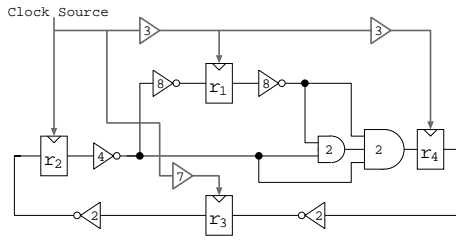


Figure 1: A semi-synchronous circuit (minimum clock period = 9)

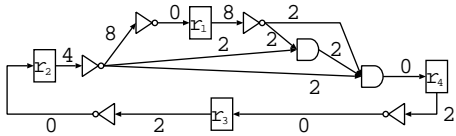


Figure 2: Circuit graph G

Fig. 2, $d_{\max}(r_2, r_4) = 8$, $d_{\min}(r_2, r_4) = 6$.

In the following, we assume that the weight of each edge can be increased independently by delay insertion to the circuit. Moreover, we assume that the clock input timing of each register is controlled as we design.

3 A lower bound of the clock period of a circuit

The *delay-to-register ratio* of a directed cycle L in G is defined as

$$\frac{\text{sum of edge weights over } L}{\text{number of registers in } L}$$

Clearly, the delay-to-register ratio of any directed cycle on G gives a lower bound of the clock period of G . Let $T_B(G)$ be the maximum delay-to-register ratio of the directed cycles on G . $T_B(G)$ also gives a “lower Bound” of the clock period, and plays an important role in the following discussion. Notice that $T_B(G)$ can be obtained in polynomial time [3]. Notice also that no circuit G works with clock period less than $T_B(G)$, even if retiming techniques or semi-synchronous techniques are applied, unless delays of elements are reduced or the number of registers in a cycle is increased.

In the following, we discuss a lower bound of the clock period of a circuit in the complete-synchronous framework and the semi-synchronous framework, respectively.

3.1 Complete-synchronous circuits

Since a complete-synchronous circuit has the premise that a clock ticks all the registers simultaneously, the maximum delay of any path between registers must be smaller than the clock period. Thus, a lower bound of the clock period of a complete-synchronous circuit is given as $\max_{(r_i, r_j) \in E_r(G)} (d_{\max}(r_i, r_j))$. To reduce this value, retiming relocates registers of a given circuit while preserving its functionality. Since it does not change the delay-to-register ratio of any cycle, a lower bound of the

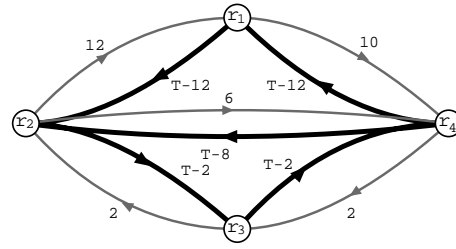


Figure 3: Constraint graph $G_{cg}(G, T)$ of G

clock period achieved by retiming is given by $T_B(G)$ [4].

3.2 Semi-synchronous circuits

In semi-synchronous circuits, the clock input timing of a register may be different from other registers. The *clock-timing* $s(r_i)$ of register r_i is defined as the difference in clock arrival time between r_i and an arbitrary chosen reference register.

We assume that a circuit works correctly with clock period T if the following two types of constraints are satisfied [2].

No-Double-Clocking Constraints

$$s(r_j) - s(r_i) \leq d_{\min}(r_i, r_j) \quad \text{for } \forall (r_i, r_j) \in E_r(G)$$

No-Zero-Clocking Constraints

$$s(r_i) - s(r_j) \leq T - d_{\max}(r_i, r_j) \quad \text{for } \forall (r_i, r_j) \in E_r(G)$$

These two types of constraints must be satisfied in complete-synchronous circuits, too. However no-double-clocking constraints in the complete-synchronous framework could be ignored, since $s(r_i) = s(r_j) = 0$ and $d_{\min}(r_i, r_j) \geq 0$ are considered to hold.

We define the *constraint graph* $G_{cg}(G, T)$ as follows: a vertex corresponds to a register in $V_r(G)$, and an edge corresponds to either type of constraints. An edge which corresponds to the no-double (no-zero, respectively) clocking constraint is called D-edge (Z-edge, respectively) and the weight of the edge is $d_{\min}(u, v)$ ($T - d_{\max}(u, v)$, respectively). The sets of D-edges and Z-edges are denoted by $E_d(G)$ and $E_z(G)$, respectively. For example, the constraint graph $G_{cg}(G, T)$ for the circuit graph shown in Fig. 2 is shown in Fig. 3.

It is known that the circuit works with clock period T in the semi-synchronous framework if and only if the constraint graph contains no negative weight directed cycles [6]. Note that if the circuit works with clock period T , then the circuit works with T' such that $T' \geq T$. Let $T_S(G)$ be the minimum T such that $G_{cg}(G, T)$ contains no negative weight directed cycle. This gives a lower bound of the clock period of G in semi-synchronous framework. We can determine the minimum clock period $T_S(G)$ and the clock-timing of each register in polynomial time [2, 1, 6].

It is easy to see the following lemma.

Lemma 1 $T_S(G) \geq T_B(G)$

The constraint graph $G_{cg}(G, T_B(G))$ is denoted by $G_{cg}(G)$ for simplicity. If $T_S(G) > T_B(G)$, there are

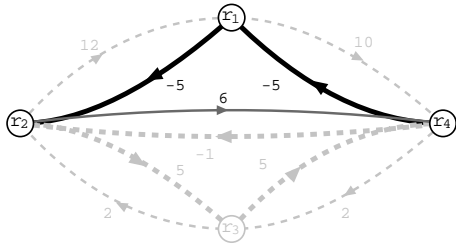


Figure 4: A negative cycle on $G_{cg}(G)$ ($T_B(G) = 7$, $T_S(G) = 9$)

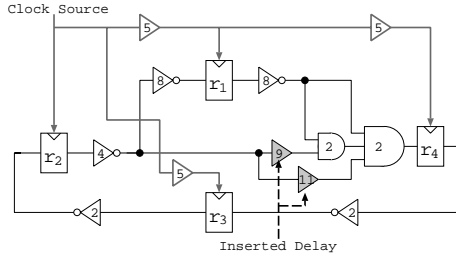


Figure 5: Circuit after delay insertion ($T_S(G) = 7$)

negative cycles in $G_{cg}(G)$ (Fig. 4).

4 Delay Insertion

We show that a circuit which works with clock period $T_B(G)$ is obtained from G by delay insertion. An example of the obtained circuit is shown in Fig. 5.

The Z-edge $e_z(r_j, r_i)$ of the constraint graph corresponds to a maximum register-path from r_i to r_j on G . Similarly, The D-edge $e_d(r_i, r_j)$ corresponds to a minimum register-path from r_i to r_j . Note that the direction of the D-edge is the same as that of a corresponding minimum register-path, but the direction of the Z-edge is opposite to that of a corresponding maximum register-path. So, a directed cycle on the constraint graph does not always correspond to the directed cycle on the circuit graph. The circuit shown in Fig. 6 corresponds to the cycle shown in Fig. 4.

Lemma 2 *If $T_S(G) > T_B(G)$, all negative cycles on $G_{cg}(G)$ have a D-edge.*

Because all negative cycles on $G_{cg}(G)$ have D-edges, we may improve the clock period by delay insertion on the circuit corresponding to D-edges. But we must not change the maximum delay-to-register ratio of the cycles in the circuit, because we cannot decrease it by delay insertion.

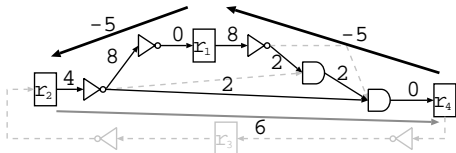


Figure 6: Circuit corresponding to the cycle on $G_{cg}(G)$

Algorithm 1

Inputs : Circuit graph G , Negative cycle C on $G_{cg}(G)$

Output : Circuit graph G' after delay insertion

Step 1: Find an unmarked D-edge (r_i, r_j) on C , and mark the edge.

Step 2: Let e_1, e_2, \dots, e_n be the edges on a minimum register-path from r_i to r_j on G . Insert the delay with the same amount of delay-slack of each edge from e_1 to e_n . Here the delay-slack of e_l is calculated after delay insertion of e_{l-1} , because the delay-slack of each edge is changed by delay insertion to another edge.

Step 3: If all D-edges on C are marked, then output the resultant circuit G' else go to Step 1.

Figure 7: Delay insertion to negative cycles

The *allowable-delay* of a cycle L on G is defined as $T_B(G) \times (\text{number of registers in } L)$.

As a result of the delay insertion, if the weight of any cycle on the circuit graph is at most the allowable-delay of the cycle, the delay-to-register ratio remains same as the original circuit.

The *cycle-slack* of L on G is defined as

$$(\text{allowable-delay of } L) - (\text{total delay of } L).$$

The *delay-slack* of an edge (v_i, v_j) on the circuit graph is the minimum of cycle-slacks of all cycles that contain (v_i, v_j) .

The delay insertion less than or equal to the delay-slack of an edge (v_i, v_j) doesn't change the maximum delay-to-register ratio.

We insert the delays to the edges on the register-path corresponding to a D-edge by Algorithm 1 shown in Fig. 7.

Theorem 1 *Let G' be the graph obtained by Algorithm 1. The weight of cycle C on $G_{cg}(G', T_B(G))$ becomes non-negative or at least one minimum register-path corresponding to a D-edge in C on G becomes non minimum register-path on G' . That is, a delay is inserted on at least one edge on G .*

If the delay between elements is not unique, the delay of a minimum register-path corresponding to a D-edge has a certain range. Theorem 1 does not always hold in such a case.

The clock period of a semi-synchronous circuit is decreased to a lower bound of clock period $T_B(G)$ by Algorithm 2 shown in Fig. 8.

When Algorithm 2 terminates, clearly $T_S(G') = T_B(G)$. So, we prove only that Algorithm 2 terminates in polynomial time.

Theorem 2 *Algorithm 2 terminates in polynomial time.*

Algorithm 2

Inputs : Circuit graph G
Outputs : Circuit graph G' after delay insertion

Step 1: Calculate $T_B(G)$, and let $G' = G$.

Step 2: Calculate $T_S(G')$.

Step 3: If $T_S(G') = T_B(G)$, output G' and terminate.

Step 4: Find the negative cycle C on $G_{cg}(G', T_B(G))$, and let G' be the circuit obtained by Algorithm 1 with input G' and C , and go to Step 2.

Figure 8: Delay insertion algorithm

Table 1: Experimental results : clock period reduction

circuit	Gates	MD	Initial	Final	ID
s1423	657	59	54	53	5987
s298	119	9	6	5.33	78
s344	160	20	17	14	225
s349	161	20	17	14	225
s444	181	11	7	6.58	57
s526	193	9	6	5.5	110

Proof: The number of times of repetition of Step 4 is at most the number of edges on G since the delay-slack of at least one edge becomes zero at each repetition. \square

5 Experimental Results

Algorithm 2 is applied to benchmark circuits in L-GSynth91. We assume that each gate has unit delay, and routing delays are zero. The clock period is reduced in 6 out of 24 circuits. The minimum clock periods before delay insertion of the other 18 circuits in semi-synchronous framework are equal to the maximum delay-to-register ratio of each circuit. The results of improved circuits are shown in Table 1. Here, the number of gates is denoted by Gate, the maximum delay-to-register ratio is denoted by MD, the clock period of the semi-synchronous circuit before delay insertion is denoted by Initial, the clock period after delay insertion is denoted by Final, and the amount of inserted delay is denoted by ID.

6 Conclusions

We prove that the minimum clock period of a circuit in semi-synchronous framework achieves the maximum delay-to-register ratio by delay insertion if the delay of each edge is unique.

As future works, the reduction of the amount of the inserted delay in Algorithm 2, more practical delay assumption (the delay of each edge is not unique), delay realization method such as detour of routing wire or delay element insertion, and the combination of retiming

and delay insertion technique to minimize the area and clock period, should be investigated.

References

- [1] R.B. Deokar and S.S. Sapatnekar. A graph-theoretic approach to clock skew optimization. In *Proc. ISCAS'94*, pp. 407–410, 1994.
- [2] J.P. Fishburn. Clock skew optimization. *IEEE Trans. on Computers*, Vol. 39, pp. 945–951, 1990.
- [3] R.M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, Vol. 23, pp. 309–311, 1978.
- [4] M.C. Papaefthymiou. Understanding retiming through maximum average-delay cycles. *Math. System Theory*, Vol. 27, pp. 65–84, 1994.
- [5] A. Takahashi, K. Inoue, and Y. Kajitani. Clock-tree routing realizing a clock-schedule for semi-synchronous circuits. In *Proc. 1997 ICCAD*, pp. 260–265, 1997.
- [6] A. Takahashi and Y. Kajitani. Performance and reliability driven clock scheduling of sequential logic circuits. In *Proc. ASP-DAC '97*, pp. 37–42, 1997.
- [7] A. Takahashi, W. Takahashi, and Y. Kajitani. Clock-routing driven layout methodology for semi-synchronous circuit design. In *Proc. TAU '97*, pp. 63–66, 1997.