

論文 / 著書情報  
Article / Book Information

|                   |                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------|
| 論題(和文)            | 遅延挿入による準同期式回路のクロック周期最小化手法                                                                                         |
| Title(English)    | Clock Period Minimization Method of Semi-Synchronous Circuits by Delay Insertion                                  |
| 著者(和文)            | 小平行秀, 高橋篤司                                                                                                        |
| Authors(English)  | Yukihide Kohira, Atsushi Takahashi                                                                                |
| 出典(和文)            | 第17回 回路とシステム軽井沢ワークショップ 論文集, Vol. , No. , pp. 529-534                                                              |
| Citation(English) | Proc. the 17th Workshop on Circuits and Systems in Karuizawa, Vol. , No. , pp. 529-534                            |
| 発行日 / Pub. date   | 2004, 4                                                                                                           |
| URL               | <a href="http://search.ieice.org/">http://search.ieice.org/</a>                                                   |
| 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。<br>Copyright (c) 2004 Institute of Electronics, Information and Communication Engineers. |

# 遅延挿入による準同期式回路のクロック周期最小化手法

## Clock Period Minimization Method of Semi-Synchronous Circuits by Delay Insertion

小平 行秀<sup>†</sup>, 高橋 篤司<sup>†</sup>

<sup>†</sup> 東京工業大学大学院 理工学研究科 集積システム専攻

〒 152-8552 東京都目黒区大岡山 2-12-1

Yukihide KOHIRA<sup>†</sup>, Atsushi TAKAHASHI<sup>†</sup>

<sup>†</sup>Department of Communications and Integrated Systems, Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552 Japan

TEL:+81-3-5734-2665 FAX:+81-3-5734-2902

Email:<sup>†</sup>{kohira, atushi}@lab.ss.titech.ac.jp

### 1 はじめに

従来の集積回路設計では、人間の理解しやすさ、設計の容易さを第一に、回路の一つの実現方法である記憶素子に同時にクロックを分配する完全同期式回路を追求してきた。しかし、その前提の下での性能追求は限界に達しつつあるだけでなく、前提の維持のためのコスト（例えば、クロックバッファ数や消費電力）が非常に大きくなっている。一方、クロックを各記憶素子に同時に分配することを前提としない準同期式回路 [3] では、物理遅延を完全同期式回路よりも効率良く利用できる可能性があるため、クロック周期、クロック面積、消費電力、信頼性などさまざまな指標で計られる性能を、低コストで実現することが期待されている。

近年の準同期式回路に関する研究は、与えられた論理回路に対して、クロック回路をどの様に設計するかに重点が置かれていた。しかし、入力として与えられる論理回路は、従来の完全同期方式を前提に合成されており、準同期方式の下で性能が改善できる回路とは限らない。クロック周期の短縮のために、完全同期方式ではレジスタ間遅延の最大値を削減するように論理合成をするが、準同期方式では必ずしもレジスタ間遅延の最大値を削減する必要はない。そのため、準同期方式を考慮することで、高速化に伴う回路面積の増大などのコストを抑えた論理回路を得られる可能性が高い。

本研究では、任意のクロックスケジュールが設定でき、各素子が一意の遅延値を有しているという仮定の下で、回路に遅延を挿入することにより、準同期式回路で

のクロック周期の下界を達成できる手法を示す。また、既存手法 [5] と比べ、挿入する遅延量、計算時間が抑えられることを示す。

### 2 準備

回路を回路グラフ  $G = (V_g, A_g)$  として表現する。 $V_g = \{v_1, v_2, \dots, v_g\}$  はグラフの点の集合で、回路におけるレジスタ、論理ゲート、配線に対応する。 $A_g = \{a_1, a_2, \dots, a_p\}$  は、グラフの有向枝の集合で、回路における信号の伝搬に対応する。点  $v_i \in V$  は、遅延値  $d(v_i)$  を有している。本研究では、 $d(v_i) (1 \leq i \leq g)$  は一意の値に定まっているとする。また、レジスタの集合を  $V_r = \{r_1, r_2, \dots, r_n\}$  とする。レジスタの集合は、回路グラフの点の部分集合で、 $V_r \subset V_g$  である。図 1 左は、回路グラフの例で、 $a, b, c, d$  がレジスタ、点の中の数字が遅延値を表す。

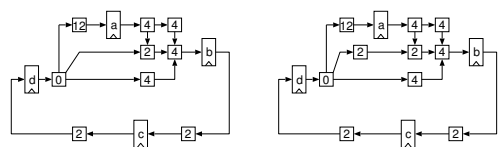


図 1: 回路グラフと遅延挿入

回路における遅延の挿入は、回路グラフにおいて 1 つの有向枝を 2 つの有向枝に切断して、その間に 1 つの点を挿入することに対応する。図 1 右の回路グラフは、図 1 左の回路グラフの  $d, b$  間の有向枝  $(0, 2)$  に遅延値 2



これは、完全同期式回路で任意量のリタイミングが可能であるという前提の下で、最適なりタイミングをした際の最小クロック周期と同じ値である。

これまでの準同期式回路の回路最適手法としては、遅延挿入による手法 [5] や、ゲートサイズの変更による手法 [4] などが提案されている。しかし [5] では、計算時間が長いので大規模回路では適応できず、余分な遅延挿入が多い。また、[4] は計算時間は短い、[5] で実現していた限界最小クロック周期を必ずしも実現できない。

**定理 2** ([5]) 論理回路を変更せず、任意のクロックスケジュールが設定可能な場合の準同期式回路の最小クロック周期  $T_S(G)$  が  $T_B(G)$  より大きいときは、負閉路内に  $D$  枝が含まれている。また、最小クロック周期が限界最小クロック周期  $T_B(G)$  であるとき、その負閉路は、 $Z$  枝のみで構成されている。

これにより、 $D$  枝に対応するパス中の配線に遅延を挿入することによって、クロック周期を改善できる。

## 5.2 遅延余裕

限界最小クロック周期を変化させない範囲での遅延挿入を行なうために、**遅延余裕**を定義する。

**定義 2** 許容総遅延, 遅延余裕

回路グラフ  $G$  中の閉路  $L$  に対して、

$$\text{許容総遅延} = T_B(G) \cdot (L \text{ のレジスタ数})$$

$$\text{遅延余裕} = (L \text{ の許容総遅延}) - (L \text{ の総遅延})$$

とする。

遅延挿入の結果、回路グラフ中の任意の閉路において、遅延の総和が許容総遅延を越えないならば、限界最小クロック周期は変化しない。

また、回路中の枝  $(v_i, v_j)$  の遅延余裕を枝  $(v_i, v_j)$  を通るすべての閉路の遅延余裕の最小値とする。枝に遅延余裕以下の遅延を挿入しても、全ての閉路に対して、許容総遅延を越えない。

既存手法 [5] では、制約グラフ  $G_{T=T_B(G)}$  上の負閉路  $C$  の  $D$  枝  $(r_i, r_j)$  に対応する  $G$  上の  $r_i$  から  $r_j$  への最小遅延のパスの各枝に対し順に、枝の遅延余裕と等しい量の遅延を挿入する。遅延余裕は、各枝への遅延挿入により変化するので、遅延挿入の毎に再計算する。

制約グラフにおける点の数を  $n$ 、有向枝の数を  $m$ 、回路グラフにおける点の数を  $g$ 、有向枝の数を  $p$  とすと、既存手法の計算時間は  $O(mgp^2)$  となる。

## 6 提案手法

既存手法のアルゴリズムでは、過剰な遅延挿入が多い。また、遅延余裕は 1 箇所に遅延を挿入すると他の遅延余裕が変わるため、回路データに戻り遅延余裕の計算をし直さなければならない。これは、回路データから閉路を見つけて計算しなければならないので、時間が掛かることになる。

そこで、あらかじめクロックスケジュールを決定し、そのクロックスケジュールにおいて制約を満たしていない部分回路に対して、遅延挿入をする手法を提案する。提案手法により、限界最小クロック周期を実現できる。また、既存手法 [5] に比べて、遅延余裕の計算が高速化でき、かつ、遅延の挿入量も抑えられる。

### 6.1 スケジューリング

クロック周期  $T = T_B(G)$  とした  $Z$  制約のみからなる制約グラフから、クロックスケジュールを定める。クロックスケジュールは、例えばあるレジスタのクロックタイミングを 0 とし、他のレジスタは  $Z$  制約のみからなる制約グラフ上のその点からの最短距離をクロックタイミングとすれば良い。

### 6.2 遅延挿入

$Z$  制約から求められたクロックスケジュールでは、必ずしも  $D$  制約が満たされていないので、 $D$  制約が満たされるように、遅延の挿入をする。

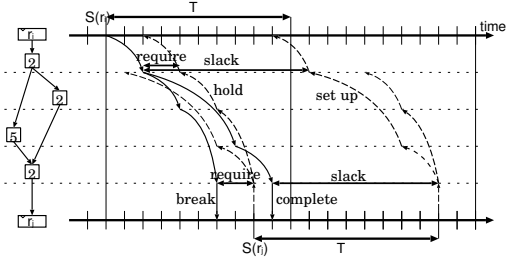


図 5: break, complete, hold, setup, slack, require

各枝  $(u, v) \in A_g$  に対して、最も早く信号が到着する時間  $break(u, v)$ 、最も早く正確な信号に確定する時間  $complete(u, v)$ 、 $D$  制約を満たすために、最も早い信号が到着できる最も早い時間  $hold(u, v)$ 、 $Z$  制約を満たすに、正確な信号にならなければならない最も遅い時間  $setup(u, v)$  を定義する。またその時間から、挿入できる遅延量である**遅延余裕量**  $slack(u, v)$ 、挿入しなければならない遅延量である**遅延要求量**  $require(u, v)$  を定

義する (図 5).

**定義 3** 枝  $(u, v) \in A_g (u, v \in V_g)$  に対し,

$$\begin{aligned} break(u, v) &= \min_{(t, u) \in A_g} \{break(t, u) + d(u)\} \\ complete(u, v) &= \max_{(t, u) \in A_g} \{complete(t, u) + d(u)\} \\ hold(u, v) &= \max_{(v, w) \in A_g} \{hold(v, w) - d(v)\} \\ setup(u, v) &= \min_{(v, w) \in A_g} \{setup(v, w) - d(v)\} \\ slack(u, v) &= setup(u, v) - complete(u, v) \\ require(u, v) &= hold(u, v) - break(u, v) \end{aligned}$$

とする. ただし,  $u = r_i \in V_r$  のとき,

$$\begin{aligned} break(r_i, v) &= s(r_i) \\ complete(r_i, v) &= s(r_i) \end{aligned}$$

とし,  $v = r_j \in V_r$  のとき,

$$\begin{aligned} hold(u, r_j) &= s(r_j) \\ setup(u, r_j) &= s(r_j) + T \end{aligned}$$

とする.

$(u, v), (v, w) \in A_g, v \notin V_r$  において,  $break(v, w) = break(u, v) + d(v)$  のとき,  $(v, w)$  の  $break$  は  $(u, v)$  によって定まる, と呼ぶ. また, パス  $P = (e_1, \dots, e_n) (e_1 = (r_i, v_1), e_k = (v_{k-1}, v_k) (2 \leq k \leq n), r_i \in V_r, v_k \notin V_r (1 \leq k \leq n-1))$  において,  $e_k$  の  $break$  が  $e_{k-1}$  によって定まる  $(2 \leq k \leq n)$  のとき, パス  $P$  は  $e_n$  の  $break$  を定めるパス, と言う.  $e_n$  の  $break$  は,  $break(e_n) = S(r_i) + \sum_{k=1}^{n-1} d(v_k)$  となる. また,  $complete, hold, setup$  についても同様に定義する.

**定理 3** クロック周期  $T$  で, 与えられたクロックスケジュール  $S$  が  $Z$  制約を満たすための必要十分条件は, すべての枝  $e$  に対して  $slack(e) \geq 0$  であることである.

*Proof.* ある枝  $e_s$  に対し,  $slack(e_s) < 0$  と仮定する.  $e_s$  の  $complete$  を定めるパスを  $(e_0^I, \dots, e_{s-1}^I)$  とする. ただし,  $e_0^I = (r_a, v_1^I)$ ,  $e_k^I = (v_k^I, v_{k+1}^I) (1 \leq k \leq s-1)$ ,  $v_k^I$  の遅延値を  $d_k^I (1 \leq k \leq s)$  とする. また  $e_s$  の  $setup$  を定めるパスを  $(e_1^O, \dots, e_t^O)$  とする. ただし,  $e_k^O = (v_k^O, v_{k+1}^O) (1 \leq k \leq t-1)$ ,  $e_t^O = (v_1^O, r_b)$ ,  $v_k^O$  の遅延値を  $d_k^O (1 \leq k \leq t)$  とする (図 6).

このとき,  $complete(e_s) = S(r_a) + \sum_{k=1}^s d_k^I$  で,  $setup(e_s) = S(r_b) + T - \sum_{k=1}^t d_k^O$  である. すると,

$$slack(e_s) = setup(e_s) - complete(e_s)$$

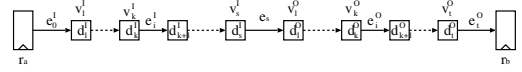


図 6:  $complete, setup$  を定めるパスの例

$$\begin{aligned} &= S(r_b) - S(r_a) + T - \left( \sum_{k=1}^s d_k^I + \sum_{k=1}^t d_k^O \right) \\ &\leq S(r_b) - S(r_a) + T - d_{max}(r_a, r_b) \end{aligned}$$

である. ここで,  $slack(e_s) < 0$  より,

$$S(r_a) - S(r_b) > T - d_{max}(r_a, r_b)$$

となり,  $(r_a, r_b)$  は  $Z$  制約を満たさない. 逆も同様. ■

以上より,  $slack(e) > 0$  の枝  $e$  に最大  $slack(e)$  まで遅延を挿入しても,  $Z$  制約を違反しない.

次の定理は, 定理 3 と同様に証明できる.

**定理 4** クロック周期  $T$  で, 与えられたクロックスケジュール  $S$  が  $D$  制約を満たすための必要十分条件は, すべての枝  $e$  に対して  $require(e) \leq 0$  であることである.

以上より, すべての枝  $e$  が  $require(e) \leq 0$  となるように, 遅延を挿入しなければ  $D$  制約を満たさない.

$D$  制約を満たさないとき, 枝  $(u, v)$  間に挿入する遅延量は,  $\min\{slack(u, v), require(u, v)\}$  とすればよい. これは, 遅延要求量の遅延を挿入すれば,  $D$  制約を満足するのに十分であり, 遅延余裕量より大きい遅延を挿入すると,  $Z$  制約を違反するからである.

**定義 4** クリティカルパス

クロック周期  $T$  で, クロックスケジュール  $S$  が与えられたとき, レジスタ  $r_i$  から  $r_j$  へのパス  $P$  は, 遅延の総和が  $S(r_j) - S(r_i) + T$  であるならば, クリティカルパスと呼ばれる.

この定義において, 全てのレジスタのクロックタイミングが等しく,  $T$  が完全同期式回路の最小クロック周期ならば, レジスタ間の最大遅延が最大であるパスはクリティカルパスとなる. 従ってこの定義は, 完全同期式回路のクリティカルパスを一般化した定義である.

**補題 1** クロック周期  $T$  において  $Z$  制約を満たすスケジュール  $S$  が与えられたとき, ある枝  $e_s$  に対し  $slack(e_s) = 0$  ならば,  $e_s$  はあるクリティカルパス上にある.

*Proof.*  $e_s$  の *complete* を定めるパス,  $e_s$  の *setup* を定めるパスを定理 3 の証明と同様に定義する.

このとき,  $\text{slack}(e_s)$  は,

$$\text{slack}(e_s) = S(r_b) - S(r_a) + T - \left( \sum_{k=1}^s d_k^I + \sum_{k=1}^t d_k^O \right)$$

であり,  $\text{slack}(e_s) = 0$  より,

$$\sum_{k=1}^s d_k^I + \sum_{k=1}^t d_k^O = S(r_b) - S(r_a) + T$$

を得る. したがって,  $(e_0^I, \dots, e_{s-1}^I, e_s, e_1^O, \dots, e_t^O)$  はクリティカルパスであり,  $e_s$  はクリティカルパス上にある. ■

### 6.3 提案手法のアルゴリズム

入力: 回路グラフ  $G$

出力: 遅延挿入後の回路グラフ  $G'$

**Step 1**: 限界最小クロック周期  $T_B(G)$  において  $Z$  制約を満たすクロックスケジュールを求める.

**Step 2**:  $D$  制約を満たしていないレジスタ対  $(r_i, r_j)$  が存在する限り以下を繰り返す.  $r_i$  から  $r_j$  への最小遅延を与えるパス上の  $\text{require}(e) > 0$  の枝  $e$  に対し,  $\min\{\text{slack}(e), \text{require}(e)\}$  の遅延を挿入する. ただし, 遅延の挿入毎に  $\text{slack}$ ,  $\text{require}$  を再計算する.

**Step 3**: 遅延挿入後の回路グラフを  $G'$  として出力し, 終了.

提案手法は,  $D$  制約を満たさないレジスタ対  $(r_i, r_j)$  に対し,  $r_i$  から  $r_j$  への最小遅延を与えるパス中の枝に遅延を挿入する. 従って, そのパスの 1 つ以上の枝に遅延余裕があることを証明すれば良い.

**定理 5** クロック周期  $T$  において  $Z$  制約を満たすスケジュールが与えられ, そのスケジュールが  $D$  制約を違反するとき,  $D$  制約を違反するレジスタ間の最小遅延を与えるパス中の 1 つ以上の枝に遅延余裕がある.

*Proof.* クロック周期  $T$  において  $Z$  制約を満たすスケジュールが与えられ, そのスケジュールにおいて  $(r_a, r_b)$  が  $D$  制約を違反するとする. このとき,

$$S(r_b) > S(r_a) + d_{\min}(r_a, r_b) \quad (1)$$

となる.

$(r_a, r_b)$  の最小遅延を与えるパス  $P = (e_1, \dots, e_{c+1})$  中の全ての枝に遅延余裕がないと仮定する. ただし,  $e_1 =$

$(r_a, v_1), e_i = (v_{i-1}, v_i) (2 \leq i \leq c), e_{c+1} = (v_c, r_b)$ ,  $v_i$  の遅延値を  $d_i (1 \leq i \leq c)$  とする. 各遅延が一意的値を有しているとの仮定から,

$$d_{\min}(r_a, r_b) = \sum_{k=1}^c d_k \quad (2)$$

となる.

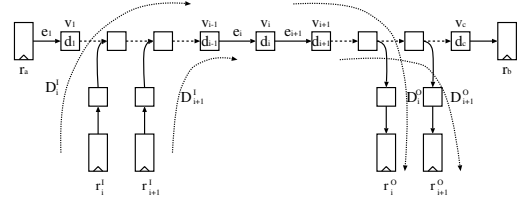


図 7:  $c+1$  本の枝がすべてクリティカルパス上にある.

補題 1 より  $e_i$  が属するクリティカルパスが存在する. そのパスを  $p_i$  とする. ただし,  $p_i$  は,  $r_i^I$  から  $r_i^O$  までのパスであるとし,  $r_i^I$  から  $d_{i-1}$  を含めないで  $v_{i-1}$  までの遅延値の総和を  $D_i^I$ ,  $d_i$  を含めず  $v_i$  以降から  $r_i^O$  までの遅延値の総和を  $D_i^O$  とする (図 7).

$p_i$  がクリティカルパスより,

$$S(r_i^O) = S(r_i^I) - T + D_i^I + d_{i-1} + d_i + D_i^O$$

という関係がある.

これより,  $p_1, p_2$  は,

$$S(r_1^O) = S(r_1^I) - T + D_1^I + d_0 + d_1 + D_1^O$$

$$S(r_2^O) = S(r_2^I) - T + D_2^I + d_1 + d_2 + D_2^O$$

であり, パス  $(r_2^I, D_2^I, d_1, D_1^O, r_1^O)$  による  $Z$  制約より,

$$S(r_2^I) \leq S(r_1^O) + T - D_1^O - d_1 - D_2^I$$

となる. 上記の 3 式の両辺を足すと,

$$S(r_2^O) \leq S(r_1^I) - T + D_1^I + d_0 + d_1 + d_2 + D_2^O$$

を得る. ここで,  $D_1^I + d_0 + d_1 + d_2 + D_2^O$  は,  $(r_1^I, r_2^O)$  間の遅延の総和である.

以下同様に  $p_{c+1}$  までこの操作を続けると,

$$S(r_{c+1}^O) \leq S(r_1^I) - T + D_1^I + \sum_{k=0}^{c+1} d_k + D_{c+1}^O \quad (3)$$

を得る.  $e_1$  の始点は  $r_a$  なので  $S(r_1^I) = S(r_a)$ ,  $D_1^I = 0$ ,  $d_0 = 0$  となり,  $e_{c+1}$  の終点は  $r_b$  なので  $S(r_{c+1}^O) = S(r_b)$ ,  $D_{c+1}^O = 0$ ,  $d_{c+1} = 0$  となる. これより, 式 (3) は

$$S(r_b) \leq S(r_a) - T + \sum_{k=1}^c d_k$$

となる。さらに式 (2) より、

$$\begin{aligned} S(r_b) &\leq S(r_a) - T + d_{\min}(r_a, r_b) \\ &< S(r_a) + d_{\min}(r_a, r_b) \end{aligned}$$

となる。これは、式 (1) と矛盾し、 $(r_a, r_b)$  が  $D$  制約を違反するという仮定を満たさない。よって、最小遅延を与えるパス中のいずれかの枝に遅延余裕が存在する。■

遅延の挿入により、各枝の *slack*, *require* は増加することはない。従って、提案手法により、遅延の挿入によって置き換えられた枝に対し、遅延を挿入することはない。また、それらの枝は *slack* = 0 か、*require* ≤ 0 となるので、このアルゴリズムは、回路グラフにおける高々枝数回繰り返すと終了する。よって、提案手法により、 $D$  制約違反を解消するだけの遅延を挿入することができる。

#### 6.4 計算時間

既存手法と同じく、制約グラフの点の数を  $n$ 、有向枝の数を  $m$ 、回路グラフの点の数を  $g$ 、枝の数を  $p$  とする。Step1 において、限界最小クロック周期を求める計算時間は  $O(mn)$  であり [2]、スケジューリングはベルマン・フォード法の計算時間である  $O(mn)$  である。Step2 の *break*, *complete*, *setup*, *hold* は、ダイクストラ法の応用である。これを  $p$  回繰り返すので、計算時間は  $O(g^2p)$  である。これによりこのアルゴリズムの計算時間は  $O(g^2p)$  となる。

### 7 実験結果

提案手法を既存手法 [5] で用いられている LGSynth91 に適用する実験を行った。各ゲートの遅延値は 1 とし、レジスタと配線の遅延は 0 とした。実験した 24 つの回路のうち 6 つの回路でクロック周期の改善が見られた。改善が見られた回路に対する結果を表 7 に示す。ただし、*gate* は回路のゲート数、*MD* は遅延挿入前のレジスタ間最大遅延、*SS* は遅延挿入前の準同期式回路の最小クロック周期、*Fin* は遅延挿入後の準同期式回路の最小クロック周期、*prev* は既存手法の遅延挿入量 [5]、*prop* は提案手法の遅延挿入量である。

既存手法と提案手法で s344 と s349 のレジスタ間最大遅延と遅延挿入後の準同期式回路の最小クロック周期は、左側が既存手法、右側が提案手法の値である。それぞれ値が異なるのは、提案手法では入力をレジスタとしたのに対して、既存手法ではレジスタとしていないからである。

実験の結果、挿入する遅延量は既存手法に比べ平均で 62% 減少した。

表 1: 実験結果

| model | gate | MD    | SS | Fin   | prev | prop |
|-------|------|-------|----|-------|------|------|
| s298  | 119  | 9     | 6  | 5.33  | 78   | 14   |
| s344  | 160  | 20/19 | 17 | 14/10 | 225  | 121  |
| s349  | 161  | 20/19 | 17 | 14/10 | 225  | 121  |
| s444  | 181  | 11    | 7  | 6.58  | 57   | 19   |
| s526  | 193  | 9     | 6  | 5.5   | 110  | 12   |
| s1423 | 657  | 59    | 54 | 53    | 5987 | 3779 |

### 8 まとめと今後の課題

本研究では、遅延挿入により準同期式回路のクロック周期最小化手法を提案すると共に、既存手法と計算時間の比較を行なった。それにより、遅延余裕量より少ない遅延量を挿入することにより、従来より少ない遅延挿入量で限界最小クロック周期を実現でき、計算時間が  $\frac{22}{9}$  倍程度の高速化が可能であることを示した。また、計算機実験により、提案手法の有効性を示した。

今後の課題は、遅延余裕量と遅延要求量から、実際にどこに遅延を挿入すれば、挿入する遅延量や計算時間が最小となるか、どのようにスケジュールを与えれば、挿入する遅延量が最小となるのかを考察することである。また、遅延値が一意でない実際の回路に対して、遅延を挿入する手法を提案することである。

#### 参考文献

- [1] J.P Fishburn. Clock skew optimization. *IEEE Trans. on Computers*, Vol. 39, No. 7, pp. 945–951, 1990.
- [2] R.M Karp. A characterization of minimum cycle mean in a digraph. *Discrete Mathematics*, Vol. 23, pp. 309–311, 1978.
- [3] A Takahashi and Y Kajitani. Performance and reliability driven clock scheduling of sequential logic circuits. In *ASP-DAC'97*, pp. 37–43, 1997.
- [4] 安井卓也, 黒川圭一, 豊永昌彦, 高橋篤司. クロックタイミング余裕度を考慮した遅延修正による回路最適化手法. DA シンポジウム 2002, pp. 259–264, 2002.
- [5] 依田友幸, 高橋篤司, 梶谷洋司. ゲートレベルの遅延挿入による準同期式回路のクロック周期の最小化. DA シンポジウム 1998, pp. 233–238, 1998.