

## 同期回路設計環境を用いた準同期クロック木構成手法

石島誠一郎, 高橋篤司

東京工業大学 大学院 理工学研究科 集積システム専攻  
〒152-8552 東京都 目黒区 大岡山 2-12-1  
TEL : 03-5734-3572, FAX : 03-5734-2902  
E-mail : {ishijima, atushi}@lab.ss.titech.ac.jp

準同期回路では, 各レジスタを異なるタイミングで動作させることで, 動作周波数の向上や, クロック木の規模の削減などを実現する. 本稿では従来の設計環境を用い, 同期回路のクロック木を修正することで準同期回路のクロック木を実現する手法を提案する. 提案手法では, まず, 配置前のゲート遅延情報をもとにクロック木の概略構造を決定し, 配置情報を用いてクロック木の詳細構造を決定する. 次に, 配線後の遅延情報を用いてクロックスケジュールを決定し, クロック木にバッファを挿入することで準同期回路を実現する. 本手法をプロセッサの実設計に適用した結果, 容易に準同期回路のクロック木を構成でき, 回路の高速化を達成することができた.

準同期式回路, 設計手法, プロセッサ, クロック木

## Semi-Synchronous Clock Tree Construction Under Synchronous Circuit Design Environment

Seiichiro Ishijima and Atsushi Takahashi

Dept. of Communications and Integrated Systems,  
Graduate School of Science and Engineering, Tokyo Inst. of Tech.  
Ookayama, Meguro, Tokyo, 152-8552 Japan  
TEL : +81-3-5734-3572, FAX : +81-3-5734-2902  
E-mail : {ishijima, atushi}@lab.ss.titech.ac.jp

A circuit in which clock is not necessary distributed to all registers simultaneously, called a semi-synchronous circuit, leads to higher frequency or smaller clock tree of the circuit compared with a synchronous circuit. In this paper, we propose a design method to realize a clock tree of semi-synchronous circuit. The method constructs the clock tree making use of conventional design environment. First, it determines the outline of the clock tree structure by using the information of gate delay and determines the detail after the placement of the circuit. Next, it determines the clock schedule by using the information of delay including routing delay and realizes the semi-synchronous circuit by inserting buffers into the clock tree. We apply the proposed method to a micro processor design and find that it is easy to apply and that a faster circuit is obtained.

semi-synchronous circuit, design method, processor, clock tree

## 1 はじめに

近年、各レジスタへ必ずしも同時刻にクロックを分配しない設計方式、準同期式設計が提案され、その実用化へ向けた検討が進められている [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. 準同期式設計を用いることで、回路の動作周波数の向上やクロック木の大きさの削減などの効果を得ることができる。しかし、現在提案されている準同期回路の設計手法の多くは、現在の設計環境のもとで実際の設計に適用するのは困難である。そこで、できる限り従来の同期回路設計環境を用い、準同期回路を実現するために同期回路のクロック木に修正を加える手法 [9] を考える。

準同期回路では、クロック周期、レジスタ間の最大・最小遅延が与えられると、回路が正常に動作する各レジスタの動作タイミング（クロックスケジュール）を決定することができる。このとき、レジスタ間遅延を配置配線前のゲート遅延の情報だけを用いて見積もると、配線遅延の影響により、実際のレジスタ間遅延とは大きく異なる可能性がある。配置配線後のレジスタ間遅延の値を用いたとしても、クロックスケジュールを実現するためにクロック木の再構築や大規模な配置配線の修正を行うと、実際のレジスタ間遅延が当初の値より大きく変化する可能性がある。したがって、見積もりレジスタ間遅延を用いたり、クロックスケジュール実現のためにレイアウトを大きく変更すると回路が正常に動作しない可能性がある。そのため、クロックスケジュールを決定する場合には、配置配線後のレジスタ間遅延の値を用い、クロックスケジュールを実現するためのレイアウト修正は極力抑える必要がある。

そこで本稿では、クロックスケジュールを実現するためのレイアウト修正を極力抑えた準同期クロック木を構成するための手法を提案する。提案手法では、まず配置配線前のゲート遅延情報をもとに、各レジスタのクロックタイミングを見積もり、目標クロックスケジュールを実現しやすいクロック木の概略構造を決定し、配置後にクロック木の詳細構造を決定する。次に配線後の配線遅延を含めた詳細な遅延情報をもとにクロックスケジュールを決定し、小数のパッファ挿入や小規模なクロック木構造の変更によりそのクロックスケジュールを実現する。この手法では、配置配線後にクロック木の概略構造を変えず、レイアウトの修正量も少ないため、クロックスケジュールの実現に伴うレジスタ間遅延の変化を抑えることができ、正常に動作する準同期回路を得ることができる。

本手法を同期式設計されたパイプラインプロセッサに適用した結果、従来の設計環境を用いて容易に準同期回路を実現することができ、動作速度が 16% 向上することが確認された。

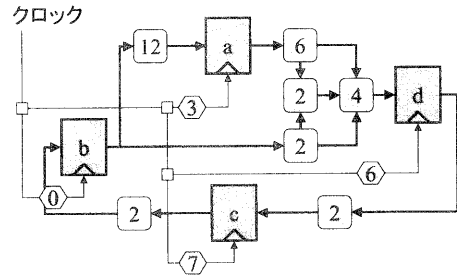


図 1: 準同期式回路

## 2 準同期式回路

本稿では、レジスタと組み合わせ回路からなる同期式回路を扱う。扱う同期式回路の例を図 1 に示す。図 1 において、レジスタ  $b$  からレジスタ  $a$  への信号遅延は 12 であり、これはこの回路のレジスタ間の最大信号遅延である。したがって、この回路の完全同期式における最小クロック周期は 12 となる。しかし、各レジスタにクロックは異なるタイミングで入力されている。クロックはレジスタ  $b$  を基準として  $a$ ,  $c$ ,  $d$  にそれぞれ 3, 7, 6 だけ遅れて入力されており、レジスタのセットアップ時間とホールド時間を 0 とすると、この回路はクロック周期 9 で動作する。このように、各レジスタにクロックを異なるタイミングで入力することを許容する回路を準同期式回路と呼ぶ。

各レジスタに入力するクロック時刻の基準時刻からの遅れをクロックタイミングと呼び、レジスタ  $v$  のクロックタイミングを  $s(v)$  で表す。図 1 では  $s(a) = 3$ ,  $s(b) = 0$  等となる。また、すべてのレジスタのクロックタイミングをまとめてクロックスケジュールと呼ぶ。

レジスタ  $u$  からレジスタ  $v$  への信号伝搬の最小（最大）遅延を  $d_{\min}(u, v)$  ( $d_{\max}(u, v)$ ) で表す。遅延グラフはレジスタを点とし、レジスタ間の信号伝搬を枝としたグラフで、回路のレジスタレベルの遅延情報を表す。各枝は最大遅延、最小遅延の二つのラベルを持つ。図 2 は図 1 の回路の遅延グラフである。例えば、レジスタ  $a$  からレジスタ  $d$  への最小遅延 10、最大遅延 12 の信号伝搬に対応し、遅延グラフは  $a$  から  $d$  へラベル 10, 12 の枝を持つ。

レジスタ  $u$  からレジスタ  $v$  へ信号が伝搬するとき、 $u, v$  のクロックタイミング  $s(u)$ ,  $s(v)$  に対しては、準同期式回路の動作条件として以下の制約が課せられる。

### Setup 制約

$$s(u) - s(v) \leq T - (d_{\max}(u, v) + \text{setup}(v))$$

### Hold 制約

$$s(v) - s(u) \leq d_{\min}(u, v) - \text{hold}(v)$$

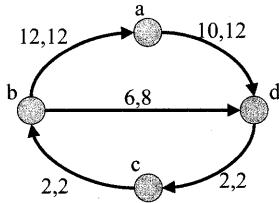


図 2: 回路のレジスタレベル表現

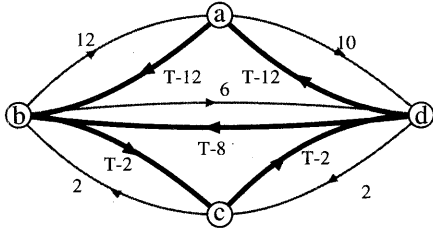


図 3: 制約グラフ  $G(V, E)$

この制約式で,  $setup(v)$ ,  $hold(v)$  はそれぞれレジスタ  $v$  のセットアップ時間, ホールド時間,  $T$  はクロック周期である.

準同期式回路が動作するためには, すべての信号伝搬に対して上記の式が成り立たなければならない.

各レジスタのクロックタイミングが任意に設定可能な場合に回路が動作する最小のクロック周期は, すべてのレジスタ間の最大遅延と最小遅延を元にして作られる制約グラフ  $G(V, E)$  から得られる. 点  $v \in V$  はレジスタに対応し, 枝  $(u, v) \in E$  は上記の2つの制約に対応する. 各枝は, Setup制約 (Hold制約) に対応し, その重みは  $w(u, v)$  は  $T - (d_{max}(u, v) + setup(v))$  ( $d_{min} - hold(v)$ ) である. また, クロック周期  $t$  のときの制約グラフを  $G_{T=t}(V, E)$  と表す. 図 3 は図 1 に示す回路の制約グラフである. ただし, この例では  $setup(v) = hold(v) = 0$  と仮定している.

遅延  $d_{max}(u, v)$ ,  $d_{min}(u, v)$ , セットアップ時間  $setup(v)$ , ホールド時間  $hold(v)$  を定数として与えられ, 各レジスタのクロックタイミングを任意に設定できるとき, 準同期式回路がクロック周期  $t$  で動作するかどうかは  $G_{T=t}(V, E)$  中に負閉路があるかどうかで判定できる. ここで, 負閉路とは閉路中のすべての枝重みの合計が負になる閉路を示す. すべての制約を満たすことが可能な最小のクロック周期を  $T(G)$  とおくと, 次の定理が成り立つ [11, 2].

**定理 1** 準同期式回路の最小クロック周期  $T(G)$  は  $G_{T=t}(V, E)$  が負閉路を持たない最小のクロック周期  $t$  である.

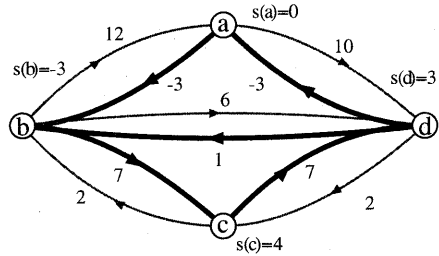


図 4: 制約グラフ  $G_{T=9}(V, E)$  とクロックタイミング

最小クロック周期は一意に定まり, 多項式時間で求めることができる.  $G_{T=t}(V, E)$  が負閉路を持たないときすべての制約を満たすクロックスケジュールが存在し, それを許容スケジュールと呼ぶ. 許容スケジュールは制約グラフを用いて得ることができる. クロック周期  $t$  における許容スケジュールは, 例えば, 次のように決定することができる.

1. 制約グラフ  $G_{T=t}(V, E)$  の任意の点  $v$  を選ぶ.
2. 点  $v$  から各点までの  $G_{T=t}(V, E)$  での最短パスを計算し, その重み和をその点のクロックタイミングとする.

図 1 の回路に対して, 最小クロック周期  $T = 9$  における許容スケジュールをこの方法で決定すると, 図 4 のようになる. ただし, 周期  $t$  における許容スケジュールは一般に一意に定まるわけではないことに注意されたい.

各レジスタ  $v$  に対して範囲  $r_s(v) = [s_{min}(v), s_{max}(v)]$  が与えられ, 各レジスタ  $v$  のクロックタイミングを  $r_s(v)$  から任意に選択したとき, クロック周期  $t$  ですべての制約をみたすならば  $r_s$  をクロック周期  $t$  における独立スケジュール可能範囲と呼び,  $r_s(v)$  を  $v$  のスケジュール可能範囲と呼ぶ. クロック周期  $t$  における独立スケジュール可能範囲は, 例えば, 次のように決定することができる: クロック周期  $t$  における許容スケジュールを  $s$  とする; 各レジスタ  $v$  に対し

$$u(v) = 1/2 \min_{(u,v) \in E} (s(u) + w(u, v) - s(v))$$

$$l(v) = 1/2 \min_{(v,u) \in E} (s(v) + w(v, u) - s(u))$$

とし,  $r_s(v) = [s(v) - l(v), s(v) + u(v)]$  とする.  $r_s$  はクロック周期  $t$  における独立スケジュール可能範囲となる. ただし, 許容スケジュール  $s$  に対し, 独立スケジュール可能範囲が一意に定まるわけではないことに注意されたい.

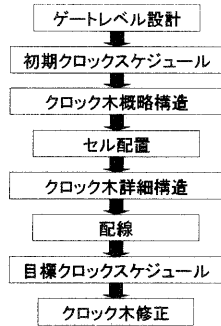


図 5: 設計フロー

### 3 準同期クロック木構成の流れ

提案する設計手法の流れを図 5 に示す。

#### 3.1 初期クロックスケジュール

配置配線前にクロック木概略構造を決定するために、まずゲート遅延を用いて初期クロックスケジュールを決定する。

チップ外部の回路との信号の受け渡しタイミングを任意に設定することは現状では困難なため、外部と信号の受け渡しをするレジスタのクロックは、外部クロックと同タイミングでなければならない。したがって、外部と信号をやりとりするレジスタを 1 つの仮想的なレジスタへ縮退して遅延計算を行い、その仮想的なレジスタに対してスケジューリングを行うことで、レジスタ群への共通なクロックタイミングを決定する。また、レジスタファイル等のマクロセルのように、レジスタ毎にクロックタイミングを変更できないレジスタ群に対しても同様にレジスタ群の縮退を行い遅延計算、スケジューリングを行う。

レジスタ間遅延の最小値と最大値は、ゲートレベルまで設計された完全同期回路のネットリストとセルライブラリから生成された SDF ファイルから回路遅延グラフを構築し、回路遅延グラフにおける対応するレジスタ間の最長、最短パス長により求める [9]。

目標クロック周期を達成することのできるクロックスケジュールは多様である。本手法では、同期回路のクロック木に修正を加えて準同期回路とする。したがって、ほとんどのレジスタのタイミングが等しいスケジュールならば、一部のみの修正で実現が可能である。そこで、各レジスタの目標タイミングを 0 とし、目標との差分が小さいスケジュールを求めることを考える。タイミングの修正は、レイアウトをできる限り変更しな

いよう、クロック木にバッファ等の遅延素子を挿入することのできる限り実現する。そのため、あるレジスタのタイミングを正に修正することは、遅延素子の挿入で実現できるため容易であるが、負のタイミングへの修正は、クロック木において遅延を減少させることは困難であるため、困難である。他のレジスタのタイミングを増加させることで、相対的に負のタイミングを実現できるが、クロック木の構造によっては必要な修正量が大きくなる。したがって、実現困難な負のクロックタイミングとなるレジスタは少なくするか、または、修正が容易なようにあらかじめクロック木の概略構造を定めなければならない。

本手法では、まず、各レジスタの目標タイミングをすべて 0 とし、コストを目標タイミングとの差の総和としたコスト最小クロックスケジュールを求める。次に、得られたスケジュールにおいて、クロックタイミングが負となるレジスタのタイミングを 0 に固定した上でコスト最小クロックスケジュールを求める。このときのコストの増分を考慮し、増分が少なければ、そのレジスタのタイミングを 0 とし、増分が大きければ、負のタイミングを実現する際の修正量を抑えるようにクロック木の概略構造を工夫する。ここで用いるコスト最小スケジューリング問題は、高速に解けることが知られている。[10]。

— コスト最小スケジューリング問題 —

入力 レジスタ間遅延, 目標クロック周期  $T$ , 目標クロックタイミング  $o(v)$

出力 クロックタイミング  $s(v)$

制約 Setup 制約, Hold 制約

目標  $\sum |s(v) - o(v)|$  の最小化

#### 3.2 クロック木概略構造

ゲート遅延情報から求めたクロックスケジュールをもとに、クロック木概略構造を決定する。負のタイミングをもつレジスタがあるとき、クロックスケジュールを少ない修正量で実現しなければならない。

遅延素子の挿入により負のタイミングを実現するためには、クロックソースから負のタイミングをもつレジスタまでの経路上の分岐点で、そのレジスタへ向かわない側に遅延素子を挿入しなければならない。例えば、図 6 の (a) と (b) では、同じクロックスケジュールを実現するが、(a) のクロック木構造では挿入遅延量が合計 12 必要なのに対し、(b) のクロック木構造では遅延量が合計 6 である。すなわち、修正量を少なくするためには、負のタイミングをもつレジスタまでの

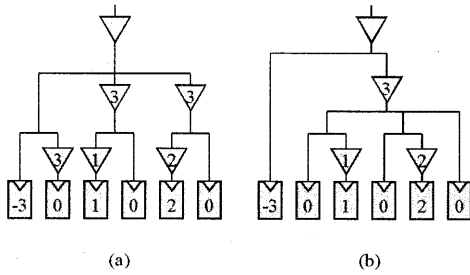


図 6: クロック木と挿入遅延量

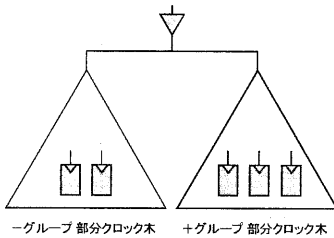


図 7: クロック木の概略構造

経路上の分岐点を少なくすればよい。

本手法では、負のタイミングをもつレジスタ群（-グループ）とそれ以外のレジスタ群（+グループ）に分割し、それぞれのレジスタ群をクロックソースから独立に駆動するクロック木の概略構造を採用する（図 7 参照）。各レジスタ群の中で再帰的に負のタイミングをもつレジスタ群とそれ以外のレジスタ群に分割する概略構造としてもよいが、負のタイミングをもつレジスタは少ないと考えられるので、再帰的に分割することは行わない。

### 3.3 クロック木の詳細構造

クロック木の概略構造に基づく配置後のクロック木合成は、レイアウトツールを用いて以下の手順で合成する。

1. ネットリストを修正し、レジスタを +グループ、-グループに分類する。
2. グループごとに等遅延のクロック木を合成し、2つの部分クロック木を得る。
3. 2つの部分クロック木を接続し、クロック木とする。

### 3.4 目標クロックスケジュール

配置配線後、初期スケジュールを得たときの制約を考慮し、配線遅延も含めた遅延情報をもとに、目標クロックスケジュールを決定する。

### 3.5 クロック木修正

まず、各レジスタ  $v$  の目標タイミング  $s(v)$  およびスケジュール可能範囲  $[s_{\min}(v), s_{\max}(v)]$  から、以下の手順で遅延素子の挿入位置および挿入量を決定する。

1. +グループに属す  $s(v) > 0$  であるレジスタのクロック入力の直前に遅延量  $s(v)$  の遅延素子を挿入する。
2. +グループの根に遅延量  $D = -\min s(v)$  の遅延素子を挿入する。
3. -グループに属すレジスタ  $v$  のクロック入力の直前に遅延量  $\max(s_{\min}(v) + D, 0)$  の遅延素子を挿入する。
4. 各分岐点に対して、その分岐点から各子分岐点までの配線上に挿入された遅延量の最小値を  $d$  とする。 $d$  が正ならば、遅延量  $d$  の遅延素子を親分岐点との配線上に挿入し、各子分岐点までの配線上の遅延素子の遅延量を  $d$  だけ減らす。

上記の手順では、連続的な遅延量が実現できると仮定しているが、実際に実現できる遅延素子の遅延量は離散的であるため、各レジスタのスケジュール可能範囲から実際の遅延素子の挿入で実現できるタイミングを選択する。また、実際にはクロックタイミングを変更するレジスタの数が少ないため、ステップ 4 で挿入遅延量の修正を行えることはほとんどない。

次に、完全同期回路のレイアウト中のクロック木に遅延素子としてバッファを挿入し、準同期回路とする。

バッファ挿入は、まず、クロック木のネットリストを手動で書き換え、次に、レイアウトツールへ完全同期回路のレイアウトと修正後のネットリストを入力し、バッファ挿入にともなうレイアウト修正を自動で行うことで実現する。

### 3.6 動作確認

完成したレイアウトから遅延情報を抽出し、配線遅延を含めた動作シミュレーションを行うとともに、全てのレジスタについて Setup 制約、Hold 制約を満たすことを確認する。

グループ	レジスタ数	候補 1	候補 2	候補 3	候補 4	目標スケジュール
A	1	-	*0	-	*0	0
B	1	-	-	*0	*0	-
C	1	+	+	+	+	+
D	1	+	+	+	+	+
E	14	+	+	+	+	0
F	18	0	+	0	+	0
G	52	0	0	+	+	0
H	1	0	0	+	+	+
I	1390	0	0	0	0	0

表 1: スケジュール結果

## 4 実験結果

完全同期回路としてゲートレベルまで設計されたプロセッサに対し、ゲート遅延情報による評価を行った結果、最小クロック周期は 14.33ns であり、任意のクロックスケジュールが実現可能としたときの準同期回路の最小クロック周期は、12.73ns であった。配置配線後の配線遅延を含めた遅延評価を行った結果、最小クロック周期は 16.35ns であり、任意のクロックスケジュールが実現可能としたときの準同期回路の最小クロック周期は、13.27ns であった。

ゲート遅延情報をもとに目標クロック周期を 12.73ns として初期スケジュールを決定し、配置配線後の遅延情報をもとに目標クロック周期を 14.31ns として目標クロックスケジュールを決定した。初期スケジュールを決定する際に用いた初期スケジュール候補と目標スケジュールを表 1 に示す。ただし、簡単のため各スケジュールにおけるスケジュール可能範囲の特徴によりレジスタを 9 つのグループに分けて示す。表中で "+", "0", "-" は、それぞれスケジュール可能範囲の下限が正、範囲が 0 を含む、上限が負であることを意味し、"\*0" はスケジュールを 0 に固定したことを意味する。なお、グループ B に属すレジスタ MAU は、外部と接続する 201 個のレジスタと内部メモリとして用いた 2048 個のレジスタを縮退した仮想レジスタである。負タイミングのレジスタ数とクロック木の修正量を考慮し、候補 2 を初期スケジュールとした。したがって、-グループのレジスタ数は、MAU に対応する 2249 であり、+グループのレジスタ数は、1478 である。クロック木の概略構造は、図 8 に示すように決定した。

配線後の遅延情報を用いた目標クロック周期 14.31ns の目標スケジュールにおいて、スケジュール可能範囲に 0 を含まないレジスタのクロックタイミングを遅延要素挿入により修正しなければならない。それらレジ

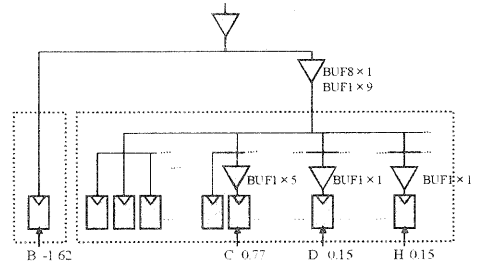


図 8: クロック木の概略構造とバッファ挿入結果

スタのスケジュール可能範囲と実際のバッファ挿入により実現したクロックタイミングを表 2 に示す。表中で BUF1, BUF8 は、バッファの種類を示す。BUF8 の駆動能力は、BUF1 に比べて大きく、面積も大きい。+グループのレジスタのタイミングを修正するためのレジスタの直前へのバッファ挿入には、BUF1 を用いた。これは、BUF1 の面積が小さく、また、入力容量がレジスタの入力容量と等しく、配線遅延への影響を抑えられるためである。負タイミングを実現するための+グループの根へのバッファ挿入には、BUF1 と BUF8 を用いた。+グループの根には BUF8 が用いられていた。そのため、そのバッファの直前に BUF1 を、さらにその前に BUF8 を挿入することで配線遅延への影響を抑えた。図 8 にクロック木へのバッファ挿入結果を示す。レイアウト修正後の遅延評価により、設計した準同期回路の最小クロック周期は 14.21ns であることが確認された。

グループ	レジスタ	可能範囲 ( <i>ns</i> )	実現タイミング ( <i>ns</i> )	BUF1(個)	BUF8(個)
B	MAU	-1.90 ~ -1.61	-1.62	9	1
C	DCHU/MA_DATA_REG[0]	0.75 ~ 1.13	0.77	5	0
D	DCHU/MA_DATA_REG[31]	0.07 ~ 1.01	0.15	1	0
H	CU/OV_Q_REG	0.08 ~ 0.74	0.15	1	0

表 2: タイミング実現結果

## 5 結論

本稿において、クロック木修正により準同期回路を実現するためのクロック木構成手法を提案した。また、提案手法を同期式で設計されたプロセッサに対し適用し、クロック木を修正することで、準同期回路を実現することができた。しかし、準同期化したプロセッサは、同期回路として遅延の最適化が十分にされているとは言い難いため、今後十分に最適化された回路を用いて実験する必要がある。また、本提案手法は、容易に準同期回路を実現できる設計法であるが、今後配線遅延の比率がより大きくなると適用が難しくなる可能性がある。また、回路の性能をより向上するために、準同期式設計の考え方を実際の設計環境にできる限り取り入れる必要があると考える。

## 謝辞

本研究を進めるにあたり、終始適切な御助言と暖かい御指導を下された東京工業大学 梶谷洋司教授、実設計に対する御指導を下された畔上謙吾氏、レジスタスケジュールの計算プログラムを作成して下さった依田友幸氏、共同でチップ設計に携わった大戸友博氏、内海哲章氏に深く感謝致します。実験で用いたチップの試作は東京大学大規模集積システム設計教育研究センターを通し、ルーム(株)および凸版印刷(株)の協力で行われたものである。また、本研究はCAD21プロジェクトの一部である。

## 参考文献

- [1] A. Takahashi, K. Inoue and Y. Kajitani: "Clock-tree routing realizing a clock-schedule for semi-synchronous circuits", Proc. 1997 ICCAD, pp. 260-265 (1997).
- [2] A. Takahashi and Y. Kajitani: "Performance and reliability driven clock scheduling of sequential logic circuits", Proc. ASP-DAC '97, pp. 37-42 (1997).
- [3] A. Takahashi, W. Takahashi and Y. Kajitani: "Clock-routing driven layout methodology for semi-synchronous circuit design", Proc. TAU '97, pp. 63-66 (1997).
- [4] K. Inoue, W. Takahashi, A. Takahashi and Y. Kajitani: "Schedule-clock-tree routing for semi-synchronous circuits", IEICE Transactions on Fundamentals, **E82-A**, 11, pp. 2431-2439 (1999).
- [5] T. Yoda and A. Takahashi: "Clock period minimization of semi-synchronous circuits by gate-level delay insertion", IEICE Transactions on Fundamentals, **E82-A**, 11, pp. 2383-2389 (1999).
- [6] M. Toyonaga, K. Kurokawa, T. Yasui and A. Takahashi: "A practical clock tree synthesis for semi-synchronous circuits", Proc. ISPD '00 (2000).
- [7] 東昌秋, 斉藤誠, 高橋篤司: "スケジュール可能範囲を考慮したクロック木合成手法", SLDM 2000-SLDM-97 (2000-79), 情報処理学会研究報告 (2000).
- [8] 斉藤誠, 東昌秋, 高橋篤司: "クロック木構成を考慮したクラスタ分割による高速クロックスケジューリング手法", VLD 2000-100 (100-437), 電子情報通信学会技術研究報告 (2000).
- [9] 大戸友博, 石島誠一郎, 内海哲章, 畔上謙吾, 高橋篤司: "準同期式設計法を用いたプロセッサ設計", VLD 2000-101 (100-437), 電子情報通信学会技術研究報告 (2000).
- [10] T. Yoda and A. Takahashi: "Clock schedule design for minimum realization cost", IEICE Transactions on Fundamentals, **E83-A**, 12, p. to appear (2000).
- [11] E. L. Lawler: "Combinatorial Optimization, Networks and Matroids", Holt, Rinehart and Winston, New York (1976).