システムＬＳＩ設計技術　107－18
(2002. 11. 28)

# 多相クロックを考慮したマルチサイクルパス解析

樋口　博之†

†（株）富士通研究所 CAD 研究部
〒 211-8588 川崎市中原区上小田中 4-1-1
E-mail: †higuchi@flab.fujitsu.co.jp

**あらまし**　本稿では，完全同期式順序回路に対するマルチサイクルパス解析手法を実際の設計に適用するため多相クロックなどの複雑なクロックを考慮する方法を提案する．本手法は，多相クロックやゲーティッドクロックを，基準となる単一クロックと各フリップフロップへのイネーブル信号に自動的に変換し，その後イネーブル端子のないフリップフロップに変換することにより複雑なクロックを考慮する．本手法をいくつかの設計例に適用した結果を示す．
**キーワード**　マルチサイクルパス，多相クロック，ゲーティッドクロック，含意操作，テスト生成

# Multi-Cycle Path Analysis Considering Multi-Phase Clocks

Hiroyuki HIGUCHI†

† Fujitsu Laboratories Ltd., CAD Laboratories
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588 Japan
E-mail: †higuchi@flab.fujitsu.co.jp

**Abstract**　This paper proposes a method for taking into account multi-phase clocks in implication-based multi-cycle path analysis. This method automatically translates multi-phase clocks and gated clocks into single clock and enable signals for driven flip-flops and then translate flip-flops with the enable inputs into those without the enable inputs. We show experimental results for some design examples to demonstrate the efficiency of our multi-cycle path analysis method.
**Key words**　multi-cycle path, multi-phase clock, gated clock, implication, ATPG

## 1. はじめに

近年，集積回路システムの高性能化の要求はますます高まっている．その一方で，テクノロジの微細化に伴い遅延見積りにおいてマージンを取るべき要因が増え，静的タイミング解析（STA）における遅延計算の悲観性はますます増大している．こうした背景から，STA によりトポロジカルな遅延のみを計算するだけでなく，回路の論理を考慮してフォールスパスやマルチサイクルパスをできるだけ多く見つけ出し，STA の悲観性をできるだけ少なくすることが高性能な回路を作ることにつながると認識されつつある．

また，テクノロジの微細化はゲート遅延に対する配線遅延を相対的に増大させるため，配置・配線をしてみないとタイミング要求が満たされているかどうか分からず，タイミング収束までの期間が増大するといういわゆるタイミングクロージャ問題も大きな問題となっている．近年この問題を解決するために論理合成やレイアウトあるいはそれらを融合したシステムでのタイミング最適化が行われるようになってきている [1]．これらのシステムによりタイミング最適化の能力は高められたが，その能力を十分に引き出すためにはできるだけ適切で必要十分なタイミング制約を与えることが重要である．フォールスパスやマルチサイクルパスのうちタイミング最適化前に確定できるものもそのようなタイミング制約の一つと考えられる．さらに，タイミング最適化によってもタイミングを満たせなかったパスがフォールスパスやマルチサイクルパスでないかを高速に判定するツールの開発が望まれている．実際，タイミング最適化後の STA でタイミング制約を満足しなかったパスについて詳しく調べるとフォールスパスやマルチサイクルパスであることが分かりタイミング制約を修正してからタイミング最適化をかけ直すということが繰り返される場合も少なくなく，それらのパスの自動解析の要求が高まっている．

組合せ回路からフォールスパスを検出する方法は従来から数

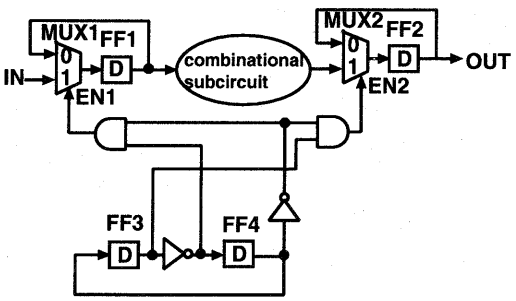


図 1　例題の回路

多く提案されている [2]～[5]．一方，順序回路からマルチサイクルパスを検出する方法に関する研究はそれほど多くない [6]～[8]．我々はマルチサイクルを自動的に高速に検出する方法を提案している [9]．その方法では単一クロックにより駆動される完全同期式順序回路のみを対象としていた．しかし，実際の ASIC の設計では多相クロックやゲーティッドクロックなどが存在するため，提案していたマルチサイクルパス解析手法の枠組み内でこれらを考慮する方法を考案した．本手法は，多相クロックやゲーティッドクロックなどのクロック木の情報を論理回路に自動変換し，変換した論理回路に対して含意操作に基づくマルチサイクルパス解析を行うものである．その方法を実装し，いくつかの設計例に適用した結果について報告する．

## 2. 準　　備

### 2.1 マルチサイクルパス解析

論理回路のパスとは外部入力あるいはフリップフロップ (FF) の出力を始点とし，途中に FF を経由せずに，外部出力あるいは FF の入力に至る経路をいう．パスの始点をソース，パスの終端をシンクと呼ぶ．

あるパスを信号がソースからシンクまで 1 クロックで到達する必要がないときそのパスをマルチサイクルパスという．特に最大 $k$ クロックで信号がシンクまで到達すればよいパスを $k$ サイクルパスという．

ある FF 対 $(A, B)$ について，$A$ をソース，$B$ をシンクとする全てのパスがマルチサイクルパスであるとき $(A, B)$ をマルチサイクル FF 対と呼ぶ．特に，FF 対間のパスのうち少なくとも一つは $k$ サイクルパスで，それ以外のパスは全て $k$ サイクル以上のパスであるとき，その FF 対を $k$ サイクル FF 対と呼ぶ．

### 2.2 マルチサイクルパスの例

マルチサイクルパスの例として図 1 の論理回路を考える．$FF_1, FF_2, FF_3, FF_4$ は D-FF である．$IN, OUT$ はそれぞれ外部入力，外部出力である．一般的には，$FF_1$ と $FF_2$ はそれぞれ複数ビットからなるレジスタであってもよい．同様に，$IN, OUT$ もそれぞれ複数ビットであってもよい．$FF_3$ と $FF_4$ は，初期状態を $(FF_3, FF_4) = (0,0)$ とすると，$(0,0) \rightarrow (0,1) \rightarrow (1,1) \rightarrow (1,0) \rightarrow (0,0)$ と繰り返し変化する周期 4 のカウンタをなす．$FF_1$ は $(FF_3, FF_4) = (0,0)$ の次のクロックでのみデータを取り込む．一方，$FF_2$ は，$FF_1$ で取り込まれたデータを楕円部分で処理し，$(FF_3, FF_4) = (1,0)$ の次のクロックでのみデータを取り込む．カウンタ $(FF_3, FF_4)$ は $(0,0)$ から $(1,0)$ となるのに 3 クロック必要であるので，3 サイクル FF 対であると分かる．

### 2.3 完全同期式順序回路に対するマルチサイクルパス解析

我々が提案している単一クロックで駆動される完全同期式順序回路に対するマルチサイクルパス解析は ATPG などで用いられる含意操作 (implication) を用いて高速にマルチサイクル FF 対を求めるものである [9]．マルチサイクルパスとはソース FF での値変化がシンク FF に伝搬するのに 2 クロック以上かかるということであるから，時刻 $t+1$ にソース FF の値が変化したときその変化が時刻 $t+2$ ではまだシンク FF に伝搬しないということである．我々の手法では時刻 $t+1$ でソース FF の値が変化したという条件から $t+2$ でシンク FF の値が変化するという条件が含意操作により得られるかどうかを調べる．

図 1 の回路を例にとり簡単に説明する．FF 対 $(FF_1, FF_2)$ がマルチサイクルパスかどうかを調べる場合を考える．値変化は実際には時刻 $t$ での $FF_1$ の値 $FF_1(t)$ と時刻 $t+1$ での $FF_2$ の値 $FF_2(t)$ により場合わけしてそれぞれ解析する．ここでは $(FF_1(t), FF_2(t+1)) = (0,0)$ の場合を考える．それ以外の 3 つの場合も同様である．

まず，$(FF_1(t), FF_2(t+1)) = (0,0)$ を割り当てる．次に，ソースの $FF_1$ が時刻 $t+1$ で信号が変化する場合を調べるため $FF_1(t+1)$ に $\overline{FF_1(t)} = 1$ を割り当てる．その後含意操作を行い既に割り当てられた値から一意的に決められる値を割り当てていく．

含意操作を行った後の値割り当ては図 2 のとおりである．図中，丸印の付いた値は含意操作前から割り当てられている値，それ以外の値は含意操作で得られた値を示す．図 2 より，$FF_2(t+2) = 0$ が得られているので，$FF_2$ は $FF_1$ が立ち上がりの信号変化が起った次のクロックでどのような入力と回路の状態のもとでも値が変化しないことが分かる．すなわち，$(FF_1(t), FF_2(t+1)) = (0,0)$ の場合には $(FF_1, FF_2)$ はマルチサイクルと判定できる．以下同様に $(0,1), (1,0), (1,1)$ の場合を順に調べ全ての場合でマルチサイクル条件が満たされればこの FF 対はマルチサイクルパスであると判定できる．含意操作の結果，もしマルチサイクルパス条件が満たされず，かつ，矛盾も生じ無い場合には，ATPG を用い $FF_2$ で値が変化する入力パタンが存在しないかどうかを調べ，存在した場合その FF 対をシングルサイクルと判定する．

## 3. 複雑なクロックを考慮したマルチサイクルパス解析

本章ではマルチサイクルパス解析を実際の設計に適用するため，どのように複雑なクロックや FF のイネーブル信号を考慮するかについて説明する．

### 3.1 クロックの扱い

マルチサイクルパス解析は 1 つのマスタクロックとそのマスタクロックから作られるスレーブクロックにより駆動される部分について行う．このひとまとまりのクロックのグループをク

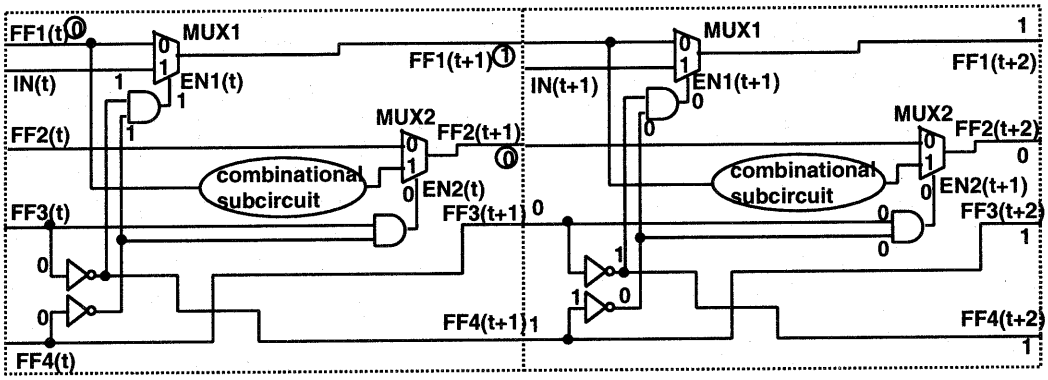


図 2　含意操作の結果

ロックグループとよぶ．従って，マスタクロックが複数ある場合は，対象以外のクロックグループに駆動される FF はブラックボックスとみなす．本手法ではブラックボックスはその入力と出力が外部出力および外部入力と等価であるとして処理する．回路全体についてマルチサイクルパス解析したい場合は，クロックグループごとに 1 つずつ行う．同期していない FF をブラックボックス化するのである程度解析の悲観性が存在する可能性があるが，マルチサイクルパスであると判定されたパスは必ずマルチサイクルパスであることは保証される．

### 3.2　複雑なクロックを考慮するための基本的方針

ネットリストとクロックの情報をもとにクロック木の解析を行う．クロックの情報は解析すべきクロックグループあるいはマスタクロックの指定と各クロックの周期などの情報である．クロックの解析により得られるゲーティッドクロック，多相クロックをそれらのクロックにより駆動される FF のイネーブル信号に変換する．これらの信号を $S_{gc}$,$S_{mp}$ とする．これらのイネーブル信号を生成するにはそのための論理回路を合成する必要がある．FF の本来のイネーブル信号を $S_{en}$ とすると，ゲーティッドクロック，多相クロックは図 3 の (a) から (b) のような回路変換により単一クロックの完全同期式順序回路となる．単一クロックになれば，我々が提案しているマルチサイクルパス解析が直接適用できるようになる．我々のマルチサイクルパス解析プログラムではこれらの変換を全て自動で行う．

以下ではゲーティッドクロック，多相クロックを FF のイネーブル信号に変換する方法について説明する．

### 3.3　ゲーティッドクロック

ゲーティッドクロックの解析は以下の手順で行う．

（1）　ネットリストからクロックグループの各クロック（クロック指定された端子）をソースとしてその推移的ファンアウト（transitive fanout）を組合わせ論理で表せないノード（FF, RAM など）に至るまで次々と取り出す．論理のないノードの CK 端子に直接つながる端子をローカルクロックと呼ぶ．

（2）　クロックとクロック木の side input に論理変数を割り当てる．ローカルクロック $LC_i$ での論理関数 $f_i$ を求める．ローカルクロック $LC_i$ のもとになるクロックに割り当てた論理変数を $x_c$ としたとき，ゲーティッドクロックにより駆動 FF へのクロックをイネーブルにする条件を論理関数で表すと，

$$CkEn\,(i) = f_i|_{x_c} \cdot \overline{f_i|_{\overline{x_c}}}$$

となる（ただし $LC_i$ とそのソースのクロックに位相差がない場合）．本手法ではこの論理関数を BDD により計算する [10]．この式の $f_i|_{x_c}$ の項は High パルスが消えない条件に関係する項であり，$\overline{f_i|_{\overline{x_c}}}$ の項は Low パルスが消えない条件を表す．

（3）　ステップ 2 の論理関数を計算する回路を作るとその出力がゲーティッドクロックに対応するイネーブル信号となる．この信号が 1 のときその次のクロックが FF に供給される．この論理回路はステップ 2 で計算した $CkEn$ の BDD から自動合成する．

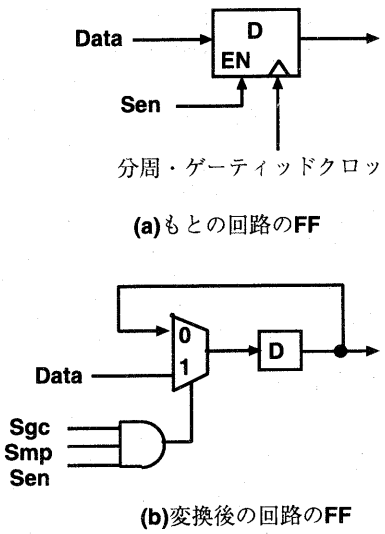


図 3　複雑なクロックやイネーブル信号の完全同期式回路への変換方針

### 3.4　周期の異なるクロック

解析対象のクロックグループで周期の異なるクロックがある場合はそれらのクロック周期の最大公約数 $T_{GCD}$ を求めそのクロック周期の仮想的なクロックとクロック数を数えるカウンタを用いて全ての FF が駆動されるようにする．例えば図 4 のよ

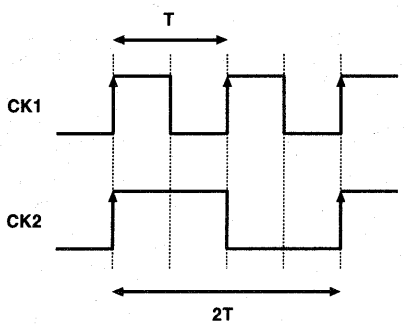


図 4　周期の異なるクロック

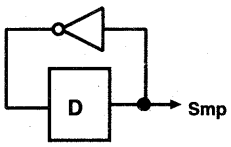


図 5　図 4 の $CK_2$ と等価なイネーブル信号 $S_{mp}$ の生成回路

うに，同期しているが周期が T と 2T の 2 つのクロック $CK_1$ と $CK_2$ がある場合を考える．この場合 $T_{GCD} = T$ であるのでクロック $CK_1$ を仮想的に基準クロックとして用いる．$CK_1$ が 2 クロックで $CK_2$ の 1 クロック分なので，クロック $CK_2$ が駆動していた FF については，あらたにクロック $CK_1$ が駆動するようにし，クロック $CK_1$ のクロック数を 2 つ数えるカウンタ，すなわち 0,1 を繰り返し出力するカウンタの値が 0 の場合のみ FF をイネーブルにするようにする．$CK_2$ に駆動される FF を $CK_1$ で駆動するためのイネーブル信号 $S_{mp}$ を生成する回路は図 5 のようになる．

各クロックの周期は外部から入力する．$T_{GCD}$ となるクロックは必ずしもマスタクロックとは限らない．

### 3.5　位相の異なるクロック

位相の異なるクロックもカウンタを利用する．クロックグループの各クロックの周期が全て同じ T であるとし，そのグループ内のある基準クロックからの位相差をそれぞれ $m_1T/n_1, m_2T/n_2, \ldots$（$n_i, m_j$ は自然数,$n_i \geqq m_i$）とすると，まず $n_1, n_2, \ldots$ の最小公倍数 $n_{LCM}$ を求めそのクロック周期 $T/n_{LCM}$ の仮想的クロックとクロック数 0 からを $n_{LCM} - 1$ まで数えるカウンタを用いて全ての FF が駆動されるようにする．例えば図 6 のように，位相が半周期ずれた周期 $T$ のクロック $CK_1, CK_2$ を考える．T/2 ずれているので周期 T/2 の仮想的クロック $CK_0$ を基準クロックとして用いる．またカウンタは周期 2 のカウンタを用いる．$CK_1$ は位相差 0 なのでカウンタの値が 0 のときにそのクロックが駆動する FF がイネーブルになるようにする．一方 $CK_2$ は $T/2$ ずれているのでカウンタの値が 1 のときそのクロックが駆動する FF がイネーブルになるようにする．$CK_1, CK_2$ それぞれに駆動される FF を $CK_0$ で駆動するためのイネーブル信号 $S_{mp}1, S_{mp}2$ を生成する回路は図 7 のようになる．

各クロックの位相差は外部から入力する．クロック木での否

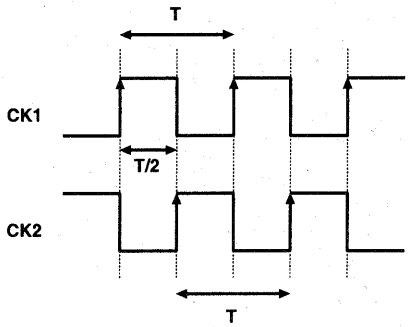


図 6　位相の異なるクロック

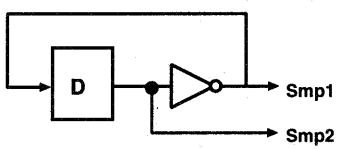


図 7　図 6 の $CK_1, CK_2$ と等価なイネーブル信号 $S_{mp}1, S_{mp}2$ の生成回路

定ゲートなどで反転される位相は 3.3 節のステップ 2 の関数 $f_i$ から求めることができる．同じクロックグループで周期も位相差も異なる場合はこれらの方法を組合わせて変換する．

### 3.6　3 サイクル以上のマルチサイクルパスの解析

以上のような方法で多相クロックを扱う場合，例えば基準クロックに対して周期が 2 倍のクロックに駆動される 2 つの FF の間のパスは明らかに 2 サイクル以上のパスであり，自明でないマルチサイクルを検出するには 3 サイクル以上のパスが存在するかを調べる必要がある．3 サイクル以上のマルチサイクルパスの解析は基本的には図 2 のように 2 時刻展開するかわりに 3 時刻以上時間展開することにより行える．$n$ サイクル以上のマルチサイクルパスを見つけるには以下のような性質が成り立つかを調べることになる：

> 「時刻 $t+1$ にソース FF の値が変化し時刻 $t+n-1$ までその値が変化しない場合に時刻 $t+2$ から $t+n$ までシンク FF の値が変化しない.」

ここで注意すべき点はこの性質は $n-1$ サイクル以上のパスであることが分かっているという仮定のもとで $n$ サイクルのパスであることを示すものであることである．従って実際には $n$ サイクル以上のパスかどうかを調べるには，2 サイクル以上のパスでないか，3 サイクル以上のパスでないか，ということを $n$ サイクルまで順に調べる必要がある．もちろん定式化として「時刻 $t+1$ にソース FF の値が変化するとき時刻 $t+1$ から $t+n$ までシンク FF の値が変化しない」かどうかを調べるとすることもできるが順に調べる場合と比べ手間は削減されない．

## 4.　実験結果

多相クロックなどを考慮したマルチサイクルパス解析プログラムの性能を評価するためいくつかの設計例に対してマルチ

表 1　回路の諸元

| 回路名 | セル数 | 外部入力数 | 外部出力数 | FF 数 | クロックの数 master | クロックの数 slave |
|---|---|---|---|---|---|---|
| 回路 1 | 36,489 | 101 | 87 | 8,672 | 2 | 2 |
| 回路 2 | 33,522 | 209 | 277 | 5,586 | 3 | 12 |

表 2　実験結果

| 回路 | clock group | FF 数 | black box 数 | 対象 FF 対数 | 乱数 sim 後 FF 対数 | MC FF 対数 | 未定 FF 対数 | 検出率 (%) | CPU 時間 (秒) |
|---|---|---|---|---|---|---|---|---|---|
| 回路 1 | A | 7,586 | 2,401 | 91,045 | 10,499 | 3,801 | 916 | 99% | 786 |
| | B | 1,022 | 14,169 | 13,964 | 13,964 | 357 | 204 | 99% | 344 |
| 回路 2 | A | 4,964 | 1,396 | 249,411 | 143,533 | 4,279 | 45,443 | 82 % | 31,770 |
| | B | 643 | 11,773 | 7,491 | 7,445 | 8 | 264 | 97% | 884 |
| | C | 9 | 13,074 | 53 | 53 | 18 | 0 | 100% | 33 |

サイクルパスを検出する実験を行った．実験を行ったマシンは UltraSPARC-III（750MHz, 主メモリ 5GB）で，使用コンパイラは gcc version 2.95.3，コンパイラオプションは "-O2" である．実験では，ATPG の最大バックトラック数は 50 に設定した．また，implication に基づく解析の前に乱数シミュレーションによりマルチサイクル FF 対でないと分かった FF 対を削除した．実験で用いた回路の諸元を表 1 に示す．表中「クロックの数」は「master」がマスタクロックの数,「slave」がスレーブクロックの数を示す．

マルチサイクルパス検出の結果を表 2 に示す．この結果は FF のイネーブル信号，ゲーティッドクロック，多相クロック全てを考慮した場合のものである．表 2 中,「clock group」は解析対象のクロックグループ名を示す。「blackbox 数」は解析中ブラックボックスとして扱ったセルの数である．ブラックボックスとして扱うセルは RAM などの他，3.1 節で述べたように対象クロックグループ以外のクロックに駆動される FF などがある.「対象 FF 対数」は，回路の FF 対のうちパスが存在する FF 対の数である.「乱数 sim 後 FF 対数」は乱数シミュレーションによりマルチサイクルパスでないと分かった FF 対を削除したあとの対象 FF 対数である.「MC FF 対数」はマルチサイクルパス解析の結果マルチサイクルパス FF 対と判定された FF 対の数である.「未定 FF 対数」は最大バックトラック数以内で解析が終了しなかった FF 対の数である．検出率は対象 FF 対のうちマルチサイクルパスかシングルサイクルパスかが同定できた FF 対数の割合を百分率で表したものである.「CPU 時間」はファイルの読み込みや乱数シミュレーションの時間なども含めたマルチサイクルパス解析全体に要した CPU 時間を秒で示したものである．

表 2 から，実回路に対しても実用的な時間で解析が行えていることが分かる．処理時間は回路の規模だけでなく対象 FF 対の数，乱数シミュレーション後の FF 対の数およびクロック木の複雑さにも大きく依存する．回路 2 のクロックグループ A に時間がかかっているのは回路 1 と比較して対象 FF 対数が多く，かつ，乱数シミュレーションでもその数が半分程度にしかならないためであると思われる．

マルチサイクルパスの検出能力は回路 2 の A を除いて 97%以上ありこれらに対しては十分な検出能力がある。回路 2 の A は 82%とやや低いが必要であれば回路全体からある程度の数のマルチサイクルパスを検出するという目的では十分な検出能力であると思われる。最大バックトラック数の制限を緩めることにより検出能力を高めることが可能である．ただし処理時間はさらに必要となる．検証に利用する場合には対象の FF 対を絞りこんで最大バックトラック数を大きくするという利用法が考えられる．

## 5. おわりに

本報告書では，我々が提案しているマルチサイクルパス解析手法を実際の設計に適用するために多相クロックなど複雑なクロックを考慮する方法を提案し，いくつかの設計例を用いて評価した結果について述べた．実験により本手法はクロックの複雑な回路に対しても実際的な時間で解析を行え，かつマルチサイクルパスの検出能力についても十分利用できる程度のものであることが分かった．今後はまずさらに多くの設計例についてマルチサイクルパス検出の実験を行う．また，外部からの信号のタイミングの情報を何らかの形で入力し，それも利用して解析できるような方法を開発することも課題である．これらと同時に，検出したマルチサイクルパスがタイミング最適化の実行時間などをどれだけ短縮できるかという実験も行っていく必要がある．さらに本解析システムの枠組みでフォールスパス検出が行えないか，また STA による遅延情報が存在する場合にはより正確な解析が行えるようなフレームワークを作っていくことも行っていきたい．

### 文　献


[1] 田代尚美, 今野正. Performance driven layout システムの概要. In *DA シンポジウム 2002*, pages 71–74, 2002.

[2] D. Brand and V. S. Iyengar. Timing analysis using functional relationships. In *Proceedings of IEEE International Conference on CAD-86*, pages 126–129, November 1986.

[3] H.-C. Chen and D. H.-C Du. Path sensitization in critical path problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 12(2):196–207, February 1993.

[4] S. Devadas, K. Keutzer, and S. Malik. Computation of floating mode delay in combinational circuits: Theory and algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 12(12):1913–1923, December 1993.

[5] P. C. McGeer, A. Saldanha, R. K. Brayton, and A. Sangiovanni-Vincentelli. Delay models and exact timing analysis. In T. Sasao, editor, *Logic Synthesis and Optimization*, pages 167–189. Kluwer Academic Publishers, 1993.

[6] A. P. Gupta and D. P. Siewiorek. Automated multi-cycle symbolic timing verification of microprocessor-based designs. In *Proceedings of the 31th ACM/IEEE Design Automation Conference*, pages 113–119, 1994.

[7] K. Nakamura, S. Kimura, and K. Watanabe. Timing verification of sequential logic circuits based on controlled multiclock path analysis. *IEICE Trans. on Fundamentals*, E81-A(12):2515–2520, December 1998.

[8] 中村 一博, 丸岡 新治, 木村 晋二, 渡邊 勝正. 充足可能性判定手法に基づいたマルチクロックパス解析. In 信学技報 *VLD99-82,ICD99-211,FTS99-60*, pages 55–62, November 1999.

[9] H. Higuchi. An implication-based method to detect multicycle paths in large sequential circuits. In *Proceedings of the 39th ACM/IEEE Design Automation Conference*, pages 164–169, June 2002.

[10] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, August 1986.