

タイミング検証機能を強化した論理シミュレータの開発

岡部 哲也 桜田 典世 小栗 澄男
吉田 宣郎 数馬 好和

三 菱 電 機 (株)

あらし 非同期式回路のタイミング検証を目的として、タイミングエラーが生じた際のエラー原因を報告する機能を強化した論理シミュレータを開発した。通常のシミュレーションの処理に加え、タイミングエラーの検出、ハザードの生成、伝播、消去の処理や各イベントが自分自身を発生させる原因となった入力パターンを保持する機能を追加している。以上の処理を加えたことによりシミュレーション速度の低下は避けられないが、前処理を施すことにより最小限に抑えることができた。3つの実回路を用いて解析した結果、通常のシミュレーションに比べて2倍弱の処理時間を要するにとどまり、実用的なツールであることが示された。

A Mixed Level Simulator with Enhanced
Timing Verification Capability

Tetsuya OKABE Noriyo SAKURADA Sumio OGURI Norio YOSHIDA Yoshikazu KAZUMA
MITSUBISHI ELECTRIC CORP.
5-1-1 OFUNA, KAMAKURA, KANAGAWA 247, JAPAN

Abstract We developed an advanced timing verification tool for asynchronous circuits on mixed level logic simulator "emsl" (Extended Mitsubishi Switch level Logic simulator). When a timing error is detected, emsl reports more detailed information than the other simulators. Special signal corresponding to hazard is used for this purpose and every event has the pointer to the input stimuli which generated the event itself. The preprocess of the circuit data realizes efficient timing verification. The application results show that the simulation with timing verification consumes at most double of the computation time of the simulation without timing verification.

1. はじめに

一般に論理回路は、同期式回路と非同期式回路に分類される。このうち、非同期式回路は、回路の動作がクロックで制御されないため、同期式回路と比較するとタイミングに関する問題が発生しやすい。このことが一因となり、大規模LSIの多くが同期式回路として設計されている。同期式回路を推奨し、非同期式回路を駆逐することは机上では簡単であるが、現実には、MSI、SSI等で設計された旧回路のLSI化や、設計者の受けた教育を反映して、多くの回路が非同期式回路を含む設計になっている。

この非同期式回路において用いられるタイミング検証ツールは、LSIの動作がクロックにより制御されていないため、同期式回路に対して用いられるパス解析を行うツール⁽²⁾を用いることができず、シミュレーションを利用した方式が用いられる。シミュレータによるタイミング検証には網羅性の問題があるが、実運用においては、テストに加えられる全入力パターンを用いて、タイミングに関する設計の可否判定を行っている。

多くのシミュレータは、シミュレーションプリミティブにタイミング制約の条件を加えることにより、タイミング検証機能を実現している。しかし、その機能はタイミングエラーの検出にとどまっているものが多く、設計者が回路をデバッグする上で十分な情報を出力しているとは言い難い。

このため、以前、当社では、エラーの原因を報告する機能を強化したタイミング検証ツールSIMCHK⁽³⁾を開発した。しかし、SIMCHKは、シミュレーション後に一括してタイミング検証を行う方法をとっていたため、全ての信号変化を記憶しなければならず、大量のディスク容量を必要とし、かつCPU時間を大量に消費（シミュレーションに要した時間の約3倍を必要）していた。そこで、SIMCHKの長所を受け継ぎ、かつ、シミュレーション実行中にタイミング検証を行う機能をホスト計算機上のMTV (Mitsubishi Timing verifier)⁽⁴⁾、EWS上のミックスレベルシミュレータemsl (Ext-ended Mitsubishi Switch-level Logic simulator) 上にそれぞれ実現した。

MTVは、FORTRANで書かれた論理シミュレータをベースに開発され、emslはC言語で書かれたミックスレベルシミュレータを

ベースに開発されている。タイミング検証機能はどちらも同じ考えの下に実現されているので、今回は後者のemslを中心に報告する。

2. ミックスレベルシミュレータemslの概要

まず、今回、タイミング検証機能を実現したミックスレベルシミュレータemslについて述べる。

シミュレーション対象レベル

スイッチレベル、ゲートレベル、機能レベル。機能記述言語は、SPD (Simulation Primitive Description) 言語。

シミュレーションアルゴリズム

全体のアルゴリズムは、イベントドリブン方式である。

双方向性のトランジスタ部分のシミュレーションは、bryantの提唱した手法⁽⁶⁾を用いている。

信号値

0 1 X Z

信号強度

5段階 (E, D, R, B, N)

遅延値

単一、標準 (立ち上がり、立ち下がり)、Pin to Pin (立ち上がり、立ち下がり)。

Pin to Pin遅延とは、入力ピン、出力ピンの各ペアー毎の遅延値のことをいう。通常、素子のファンイン数、ファンアウト数等から計算された遅延値 (仮想遅延) および、レイアウト終了後、抽出した配線容量をもとに算出した遅延値 (実遅延値) がPin to Pin遅延値となる。Pin to Pin遅延値を用いたシミュレーションを行うことにより、遅延に関する模擬精度を高め、タイミング検証をより有効なものとするができる。

3. タイミング検証機能

3.1 タイミングチェック対象

emslのタイミングチェックの対象は次の2種類である。

1) フリップフロップ、ラッチ、メモリなどの記憶素子

一般のシミュレータと同様にフリップフロップ、ラッチ、メモリなどの記憶素子に対してタイミング検証を行う。このタイミング検証のユーザインターフェースは、完全にセルのレベル（ゲートアレイのマクロセル、セルベース方式のスタンダードセル）において実現されている。すなわち、フリップフロップ、ラッチが複数のシミュレーションプリミティブに展開されているにもかかわらず、ユーザはシミュレーションプリミティブのレベルを意識することなく、セルに対してタイミング制約の指定をすることができ、かつ、エラーレポート中の素子名もセルのレベルの名前で出力することができる。

2) ゲートのみから構成されるフィードバックループ

ゲートのみから構成されるフィードバックループに対しては、タイミングチェックの対象となるループの最大ゲート段数を指定することにより、シミュレータは回路中よりループを自動検出し、タイミング検証を行う。

3. 2 検出するタイミングエラー

検出するタイミングエラーの種類を以下に示す。記憶素子に対してと、ループに対してでは若干異なったチェックを行っている。

1) 記憶素子に対するチェック

- a) 指定した2ピン間の信号変化の時間差のチェック（図1）

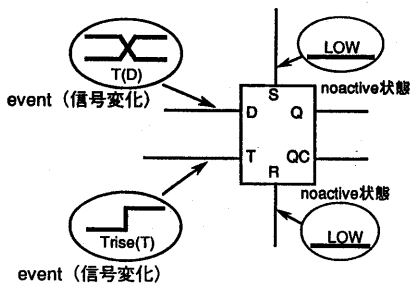


図1 リレーションチェック（セットアップタイム）

このチェックは、セットアップタイムチェック、ホールドタイムチェックを含んで、さらに、クロックピンとセットピン、リセットピンやメモリのピン間に到着する信号変化の時間差に関するチェックも行う。

- b) 指定したピンの信号値のチェック
指定したピンに信号変化が生じた際、指定した他のピンが指定した信号値の場合、エラーとする。例えば、フリップフロップでクロックにエッジ信号が伝播したときにデータピンが信号値不定である時にエラーとするなどに用いる。
- c) スパイクのチェック
指定したピンにスパイクが伝播したときにエラーとする。
- d) ハザードのチェック
指定したピンにハザードが伝播したときにエラーとする。

ここでいうハザードとは、実際のLSIの素子の遅延のばらつきを考慮したスパイク発生警告信号である。すなわち、相異なる入力ピンに伝播する信号変化の時間差が非常に小さく、遅延のばらつきによっては、スパイクを発生する可能性のある場合、ハザード信号を発生させる。詳しくは、4.4で述べる。

2) ループに対するタイミングチェック

- a) ループに入力される2つの信号線の信号変化時刻の差位のチェック
ループに入力される2つの信号線の信号変化の時間差が指定した時間より小さい場合には、ループ内の状態が安定する前に新たな変化が生じたときとみなしてエラーとする。
- b) スパイクのチェック
ループに入力される信号線にスパイクが伝播したときにエラーとする。
- c) ハザードのチェック
ループに入力される信号線にハザードが伝播したときにエラーとする。
- d) 発振チェック

ループを構成するゲートのうち、ループを構成する信号線に関し、入力の反転値を出力する状態の素子（NAND、NOR、INV、EOR）が奇数個ある場合、ループが発振する状態なのでエラーとする。

3. 3 エラーのレベル

テストの使用や実使用などの状況を考えると、全てのピンに同時に入力変化が与えられることはなく、必ず多少の時間のずれが生じる。このピン間スキューが存在することによって、設計した回路が誤動作する場合が生じる。ems1はピン間スキューまで考慮したタイミング検証が可能であり、以下のようなエラーのレベルで区別している。

1) LOGICエラー

LOGICエラーは、ピン間スキューが存在しない場合でも発生するタイミングエラーである。

2) SKEWエラー

SKEWエラーは、ピン間スキューを考慮すると、2つの信号変化の伝播する時刻が変動し、タイミング制約を違反するエラーである。

3) CAUTION

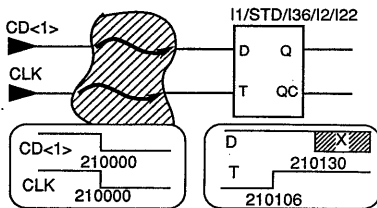
2つの信号変化の原因となる外部入力ピンの情報が失われてしまった場合において、2つの信号にピン間スキューを仮定した場合に発生するタイミングエラーである。

3. 4 詳細なエラーメッセージ

ems1は、タイミングエラーを検出すると詳細なレポートを出力し、設計者のデバッグの負担を軽減する。以下にその出力する内容を挙げる。

* Simulation time = 210130 Element is I1/STD/136/12/122
HOLD

— SKEW ERROR —
Pin T changed 0->1 at 210106
— External pin CLK changed 0 at 210000
Pin D changed 1->X at 210130
— External pin CD<1> changed 0 at 210000

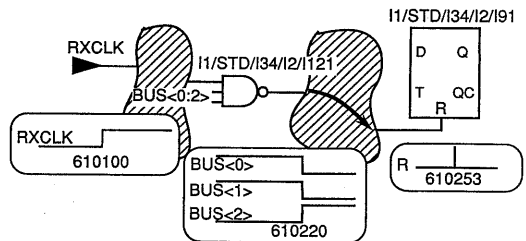


(a) ホールドタイムエラー

* Simulation time = 610253 Element is I1/STD/134/12/191
HAZARD

— LOGIC ERROR —

Hazard was propagated to pin R
Hazard was generated at the element I1/STD/134/12/121
Signal I1/STD/134/12/BUS<0> changed 1->0 at 610220
— External pin RXCLK changed 1 at 610100
Signal I1/STD/134/12/BUS<1> changed 1->0 at 610220
— External pin RXCLK changed 1 at 610100
Signal I1/STD/134/12/BUS<2> changed 0->1 at 610220
— External pin RXCLK changed 1 at 610100



(b) ハザードエラー

図2 タイミングエラー出力例

4. 実現方式

4.1 タイミング検証システム構成

まず、タイミング検証実行時の e m s l 全体フローについて述べる。図3にタイミング検証時の概略構成図を示す。図3の斜線部がタイミング検証時に必要になる処理、ファイルである。

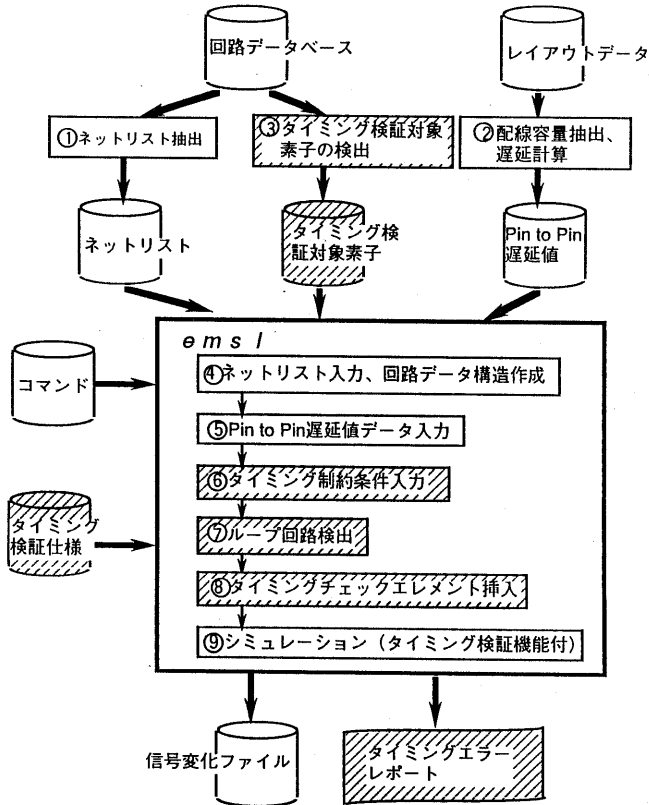


図3 e m s l タイミング検証の概略構成図

シミュレータ起動前の処理

- ① 回路データベースより、ネットリストを抽出する。
- ② レイアウトデータから配線容量を抽出し、これからPin to Pin遅延値を算出し、ファイルに格納する。
- ③ 回路データベースより、タイミング検証対象セル、モジュールの検索を行い、論理接続関係を抽出する。

シミュレータ内処理

- ④ ネットリストを入力し、e m s l 内部に回路データ構造を作る。
- ⑤ Pin to Pin遅延値ファイルを入力し、各素子に対して、遅延値のテーブルを作る。

⑥ タイミング制約を入力し、タイミング制約条件を e m s l 内部表現に変換する。

⑦ ループ検出ルーチンが内部に作った回路データ構造を探索して、ループを検出する。

⑧ ③で作成したファイルを入力し、⑦で検出したループ回路とともに、タイミングチェックエレメントを付加する。タイミングチェックエレメントは4.2でのべる。

⑨ テストパターンを入力しながら、シミュレーションを実行し、タイミングエラーを検出するとエラーレポートを出力する。

4.2 タイミングエラーの検出方法

e m s l では、シミュレーションプリミティブにタイミング制約を加える方法をとらず、タイミングチェックの対象とする素子、ループなどに対し、シミュレータ内部で新たにタイミングチェックエレメントを付加する方法(図4)をとった。このエレメントは見かけ上は他の一般の素子と同一で、素子の機能として課せられたタイミング制約のチェックを行うものである。この方式を用いる利点は以下が考えられる。

- ・エラーの検出に関して、既存のシミュレータに手を加える必要が無い。
- ・シミュレーションプリミティブのレベルの制約を受けずに、任意のレベルでタイミング検証を行うことができる。
- ・ループに対しても同一の方法でタイミングエラーの検出が行える。

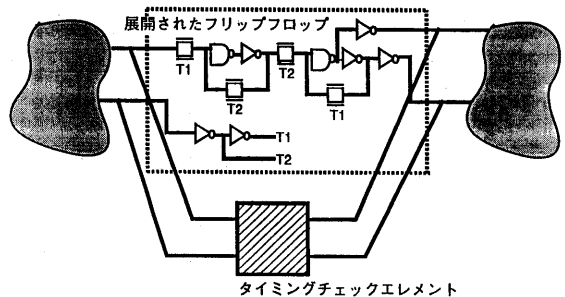


図4 タイミングチェックエレメントの付加

4.3 外部入力変化情報の伝達

各イベントは、そのイベントの発生の原因となった外部入力変化情報を保持する。この情報は、エラーレポートの出力、SKEWエラーチェックの際に参照される。この情報の伝達は以下のようにして行われる。

- ・外部入力の信号変化のたびに、その変化情報をテーブルとして記憶する。
- ・イベントは、そのイベントの発生の原因となった外部入力変化テーブルへのポインタテーブルをリスト構造としてもつ。
- ・素子の入力ピンにイベントが伝播し、出力信号が変化した場合、入力ピンのイベントが持っている外部入力変化テーブルへのポインタの値を出力イベントにコピーすることにより、外部入力変化情報を伝播する。

ポインタテーブルをリストとしてもつことにより、複数の入力ピンに同時に信号変化が生じ、新たに発生したイベントが複数の外部入力の原因による場合でも、情報が欠落することなく、伝播することができる。(図5)

なお、外部入力変化のテーブルは、イベントから参照されている回数を記憶しておき、参照回数が0になるとそのデータ領域を解放する。

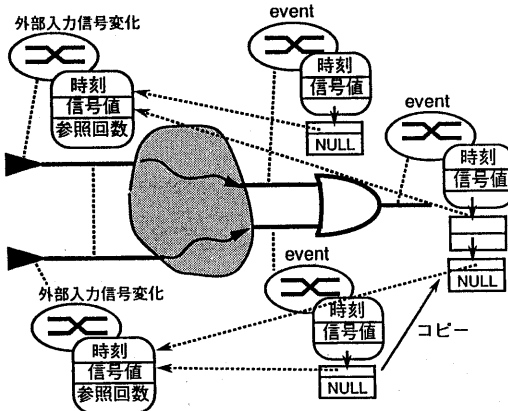


図5 イベントの発生と外部入力変化情報の伝達

4.4 ハザードの発生、伝播、消滅

ANDゲートを用いてハザードの発生、伝播、消滅(図6)を説明する。ANDゲートは、静的0ハザード(0→1→0)のみ発生する可能性

がある。ANDゲートがこのハザードを発生するのは、立ち上がり信号の直前に、立ち下がり信号が伝播し、かつ、信号変化のない入力ピンの信号値が1の場合である。ems1では、各素子の入力ピンのテーブルに信号変化の履歴(図7)をもち、ゲートの出力評価時に、各ピンの信号変化履歴を参照し、指定した時間幅内に発生した立ち上がり、立ち下がりの信号変化数、および、信号変化を起こさなかったピンに関しては、そのピンの信号値を調べることによってハザードの発生を調べている。

ハザードの伝播、消滅に関しては、ANDゲートに入力される他のすべての入力ピンの信号値が1であればハザードは伝播し、そうでなければ消滅する。

また、実際のLSI回路においては、各素子の立ち上がり、立ち下がり遅延の差によって、ハザードが消滅する場合が考えられるが、今回はそれは考慮していない。ただし、それはより安全側のシミュレーションであると考えられる。

OR、EORについても同様な方法でハザードの発生、伝播、消滅を扱うことができ、複合ゲートについても、AND、ORの論理の組み合わせなので、上記の処理の組み合わせでハザードの処理を実現している。

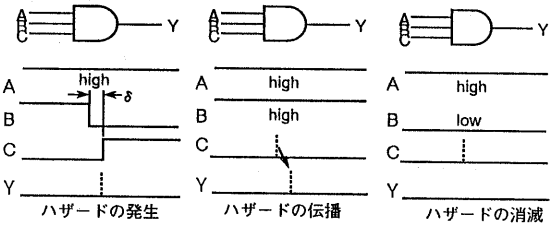


図6 ハザードの発生、伝播、消滅

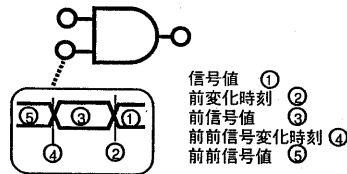


図7 ems1が持つ入力ピンの信号変化の履歴

4. 5 処理速度への配慮

以上のように、タイミング検証モードでは、通常のシミュレーションに対して処理が増えるため、実行速度の低下が懸念される。今回、特にハザードに関する処理時間を削減するために、次のような前処理を行っている。

一般に、ハザードの伝播によってタイミングエラーとなるのは、フリップフロップのセット、リセット、クロックピンが主な対象であり、すべてのピンがハザードの伝播によってエラーとなるわけではない。このため、ems1はシミュレーション前にハザードをエラーとするピンを調べ、そのピンを起点として回路のバックトレースを行い、ハザードの発生、伝播、消滅の処理を行う範囲を限定した。(図8)

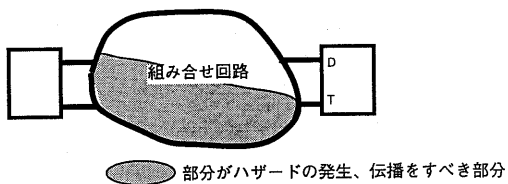


図8 ハザードの処理を行う範囲(概念図)

5. 速度評価および解析

本プログラムの性能を評価するために、3つの回路に対して、タイミング検証モードと通常のシミュレーションモードでのems1の処理時間を測定した。これら回路は、いずれも、実際に開発されたLSIあるいは、開発中のLSIである。また、設定したタイミング制約も、実回路に適用する条件を用い、実使用時を意識した速度評価を行った。

表1に結果を示す。タイミング検証モードで

表1 実行結果 (TVはタイミング検証モードを示す)

	モード	素子数	CPU時間	CPU比
回路A	TV	6081	245s	1.67
	非TV		147s	
回路B	TV	14646	3839s	1.26
	非TV		3035s	
回路C	TV	3412	4499s	2.02
	非TV		2222s	

(Sun4/260C 128Mbyte)

表2 タイミングチェックエレメントの割合

	素子数	タイミングチェックエレメント数	素子評価回数に占めるタイミングチェックエレメントの割合
回路A	6081	35 (0.5%)	0.6%
回路B	14646	616 (4.0%)	7.4%
回路C	3412	618 (15.3%)	41.9%

の実行は、通常シミュレーションの1.3から2倍くらいの時間を要することがわかる。次に回路Cのタイミング検証モードの実行速度が他の2つの回路に対して遅くなった原因を調べる。まず、図9に、シミュレーション結果より得た2入力nandゲート、ラッチ、タイミングチェックエレメントの平均的な処理時間の比を示す(インバータの処理時間を1とする)。これより、タイミングチェックエレメントの評価時間は、1シミュレーションプリミティブと同等であることがわかる。また、表2には、全素子数に対するタイミングチェックエレメントの個数、および、シミュレーションにおける全素子評価回数に対するタイミングチェックエレメント評価回数の比を示す。タイミング検証モードでの実行速度が遅い回路Cは、他の2つに比べて、回路中のフリップフロップ、ラッチが多く、これにより、素子の評価回数が増大し、処理速度が他の回路に比べて遅くなっていることがわかる。また、回路Aは、回路中のラッチ、フリップフロップの割合は非常に小さいが、ハザード関係の処理に時間がかかり、実行速度が回路Bに比べて遅くなっていることがわかっている。ハザード関係の処理が多くなっている理由は、回路の特徴やテストパターンによるものと考えられる。

このように、タイミング検証モードの速度は、回路中に占めるラッチ、フリップフロップの割合やテストパターンなどにより変化するが、ほぼ通常シミュレーションの2倍弱の時間を必要とするといえる。

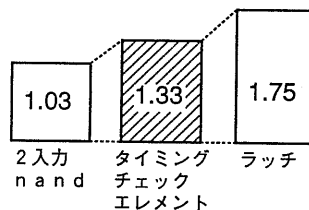


図9 各素子の評価時間

(インバータを1とする)

次に、前処理により、ハザードの処理範囲を限定した効果をみる。図10、11に回路Aにおける回路中の全素子に対するハザードの処理を行うべき素子数、および、前処理を行った場合と行わなかった場合のシミュレーション速度の比較を示す。前処理を行うことにより、ハザード処理を行うべき素子数を約4割ほど削減できた。また、シミュレーション時間に関しては、シミュレーション全体でかかるCPU時間は約3割削減でき、また、タイミング検証処理に要する時間で比較すると約2分の1にすることができ、十分効果が現れているといえる。

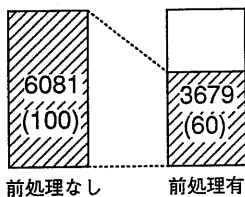


図10 ハザード処理を行う素子数 (回路A)

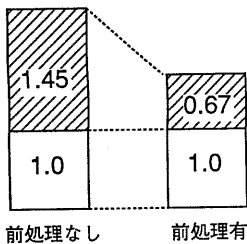


図11 シミュレーション時間 (回路A)

▨ はタイミング検証処理に要する時間

6. まとめ

ミックスレベルシミュレータems1上にタイミング検証機能を実現した。多くのシミュレータが実現している機能に加え、タイミングエラーが生じた際に、エラーの原因となった外部入力の信号変化の情報、スパイク、ハザードの発生情報を出力することができる。このため、ハザードの発生、伝播、消滅の処理、外部入力変化情報の伝達など、通常のシミュレーションに対し、処理が増加しているが、実使用条件下での測定結果では、通常シミュレーションの高々2倍ほどの時間でタイミング検証が行えることが実証された。SIMCHKと比較すると、タイミング検証用にディスクを使用する必要がない点に加え、シミュレータの速度改善と相ま

って、タイミング検証に要するCPU時間を大幅に改善(10~15倍程度)できた。なお、MTV、ems1は、これまでに約100品種のLSIの開発において利用され、その有効性が確認されている。

参考文献)

- 1) 松下: タイミング検証、情報処理 Vol.25 No.10 pp.1056-1061(1984)
- 2) J.K.Ousterhout: "Switch-level timing verifier for digital MOS VLSI", IEEE Trans. Computer-Aided Design, Vol.CAD-4, pp.336-349, July 1985
- 3) 荻原、松下、村井: 順序回路のタイミングベリファイア(SIMCHK)、情報処理学会研究報告、電子装置設計技術10-3, pp.1-20(1981)
- 4) 吉田、鶴崎、長谷川、小野: タイミングエラーが生じた外部入力パターンを出力するタイミング検証システム、電子情報通信学会春季全国大会(1989) vol.1 p262.
- 5) 南谷: 非同期式回路の理論 [I] ~ [IV], 電子通信学会誌 vol.63, No.6, No.7, No.8, No.9
- 6) R.E.Bryant, "An algorithm for MOS Logic Simulation" Lambda Magazine, Vol.1, No.3, pp.46-53(1980)