

# MOS LSI を対象とするシミュレーションエンジン†

小田原 豪太郎<sup>††</sup> 冨田 昌 宏<sup>††</sup>

論理装置において MOS LSI がますます多用されるに至り、MOS 素子の双方向性、トライステート出力等の特徴を利用した回路が一般的となっている。開発期間短縮のため、論理設計段階で MOS 素子の動作、タイミングを短時間で正しく評価する必要がある。従来の MOS 素子評価手法には、遅延の扱いやアルゴリズムの複雑さの点で問題があった。一方、検証の高速化のために専用ハードウェアによるシミュレーションエンジンが開発されてきた。しかし、その前処理において、多くが単純なゲート単位の回路分割手法をとり、ユニット間通信量の増大を招いていた。本論文では双方向バス等を多用する論理装置のタイミング検証を対象とし、(1)遅延・信号強度・不定値を正確に評価し、かつイベント法に適した MOS 素子評価アルゴリズム、(2)実行結果を利用した機能ブロック単位の回路分割手法による処理効率向上、の2点の特徴を持つシミュレーションエンジンについて述べる。MOS 素子について直前の信号値評価の結果から信号源となった端子を記憶し、素子単位の評価を行いつつパス全体について矛盾のない信号値・信号強度評価を行うことが可能である。前処理において、実行結果を利用した機能ブロック単位の分割手法により、ユニット間通信量の削減と負荷の均一化を図った。実験により提案した各項目の有効性を確認した。

## 1. はじめに

論理装置を構成する要素として MOS LSI はますます重要な位置を占めるに至っている。論理装置の早期開発のため、これら MOS LSI を短期間で誤りなく設計することが不可欠の条件となっている。また、回路の高速化が進み、タイミングに関しても厳密な検証を行うことが求められている。

よって論理設計を支援するシステムには以下の2点が必要となる。

- (1) 短時間でタイミングエラーの検証が可能。
- (2) MOS 回路の正確な評価。

(1)の要求を満たすため、論理設計において多くの時間を要する論理、タイミングのシミュレーションを専用ハードウェアにより高速化するシミュレーションエンジンが開発されてきた<sup>1)~4)</sup>。従来のエンジンは、LSI 設計に必要な対話処理を前提とする場合、エンジン上で実行するために必要な前処理等に要する時間が問題となっていた<sup>5)</sup>。また、従来実現されてきたエンジンの多くは、回路分割手法がエンジン上の処理効率に与える影響について考慮していなかった。

(2)については、MOS 回路を論理レベルで正しく評価するため、種々のモデル化手法が提案されてきた<sup>6)~13)</sup>。以下の2種類に分類される。

- i) 素子単位の評価を行う手法<sup>6)~9)</sup>

既存のゲートレベルシミュレータの素子モデル拡張により実現可能である。信号伝播方向の決定手法に問題があり、ループを含む回路について誤った結果を導く場合があった<sup>14)</sup>。

- ii) スイッチレベルで互いに導通状態にあるパス単位で評価を行う手法<sup>10)~13)</sup>

MOS 素子評価の正確さ、種々の回路への対応の点で優れている。しかし、アルゴリズムが比較的複雑となり、かつタイミング評価の点で問題がある。

ASIC に代表される LSI を用いたシステムを、Quick Turn-Around-Time (QTAT) で誤りなく設計するため、遅延、信号強度、MOS 素子の評価が可能で応答時間の短いシミュレーションエンジンが必要である。

そこで本論文では、イベント法を用いたゲートレベルシミュレータ上での実現に適した MOS 素子評価手法を提案する。信号伝播の方向を考慮し、動作・遅延の正確な評価を行うとともに不要な X 発生を抑える手法に特徴を持つ。本手法をハードウェア上で実現し、処理効率を高める回路分割手法に特徴を持つ TASSE システム<sup>15)</sup>についても述べ、有効性を評価する。

## 2. TASSE: Timing and Strength Simulation Engine

MOS LSI を含むシステムの検証に対する要求を満たすため、TASSE は以下に示す特徴を持つ。

- (1) ハードウェア化に適した素子単位の評価により、正確な遅延・信号強度・不定値の評価を行う

† A Simulation Engine for MOS LSI by GOTARO ODAWARA and MASAHIRO TOMITA (Department of Precision Engineering, Faculty of Engineering, University of Tokyo).

†† 東京大学工学部精密機械工学科

MOS 素子評価アルゴリズム.

(2) 通信量削減と負荷の均一化を同時に実現する, 実行結果を利用した機能ブロック単位の回路分割手法.

以下, 仕様の根拠とする, 並列処理台数と処理速度に対する評価を示した後, ソフトウェア・ハードウェアの仕様と構成について述べる.

2.1 並列処理台数と処理速度

従来, シミュレーションエンジンに対しては, 特定の回路とパターンを用いて並列処理効率に対する評価が行われてきた<sup>1), 16)</sup>. 本論文では, 並列処理台数  $n$  とオーバヘッドの関係について定式化を行う. 特に, 回路分割手法による通信量の差を明らかにする.

1 シミュレーション時刻の処理に必要な時間  $T_{ex}$  は次の式で表される.

$$T_{ex} = T_{alt} + \max_{i=1 \dots n} (T_{proc} + T_{cov}) \quad (1)$$

ただし,  $T_{alt}$  は時刻切替えに要するクロック数を,  $T_{proc}$  はシミュレーション実行時間を,  $T_{cov}$  は通信オーバヘッドを表す.  $T_{alt}$ ,  $T_{cov}$  の存在と  $T_{proc}$  のユニット間のばらつきが, 並列処理に伴うオーバヘッドの3要素となる.

ここでは,  $T_{cov}$  評価のためユニット間通信への依存度を評価する. 通信依存度は, 1時刻当り並列処理ユニット間で通信されるイベント数:  $P_{ext}$  と全イベント数:  $P_{all}$  の比で定義される. ランダムなゲート単位分割を行うと, 通信依存度は  $(n-1)/n$  となる. 機能ブ

ロックを利用した分割を行うと, 各ユニットから外部に出力される信号線本数  $p$  は Rent の法則<sup>17)</sup>に従い, 以下の式で示されるとする.

$$p = aG^b \quad (a \doteq 2, b \doteq 0.6) \quad (2)$$

ここで,  $G$  はユニット当りのゲート数を示す.  $a, b$  は経験的に定められた値である. 全ゲート数を  $A$  で表すと, 通信依存度に関して以下の式が導かれる.

$$\frac{P_{ext}}{P_{all}} = \frac{a(n-1)}{n^b \cdot A^{1-b}} \quad (3)$$

機能的分割法と, ゲート単位のランダム分割法のそれぞれを適用した場合の, 通信依存度に対する評価結果を図1に示す. また, バスによる通信方式を採用した場合に, ユニット間通信に要する時間と実行処理に要する時間が等しくなる  $P_{ext}/P_{all}$  を, バス容量として表す.

また, 式(1)における  $T_{alt}$  と  $T_{proc}$  のばらつきによるオーバヘッドの評価を行い, 前者は  $O(n)$ , 後者は  $O(n^{0.5} (\log n)^{0.7})$  のオーダーで増加することが明らかとなった. しかし, これら要因による処理時間増加の和は, 2万ゲート規模回路に対して8台のユニットを適用した場合に40%以下であった.

これらの結果より, 次の3点が明らかとなった.

- i) 機能的分割はランダム分割に比べ, ユニット間通信量を1/6~1/15に削減する効果がある.
- ii)  $n \leq 8$  ならば, バスによる通信方式で問題はない.

iii) 対象回路規模に応じて  $n$  を決定する必要がある.

そこで, 本システムではプロトタイプとして4台のユニットによる並列処理をとり, ハードウェア量の少ないバス通信方式を採用した.

2.2 仕様

2.2.1 ソフトウェア仕様

本システムでは, 市販のEWSと同様の階層的設計支援環境を提供する. デジタイザを用いて対話的・図形的に回路情報, テストパターン等を入力する. 回路分割・リンク処理によりエンジン上で実行可能なモデルを作成する. これを入力としてシミュレーションを実行し, 結果をリアルタイムで画面表示する. ここでは, 特徴的な回路分割手法について述べる.

本システムでは, 処理ユニット間の通信量削減と, 各ユニットへの負荷均一化を同時に満たす回路分割手法を提案する. 以下の2点に特徴をもつ.

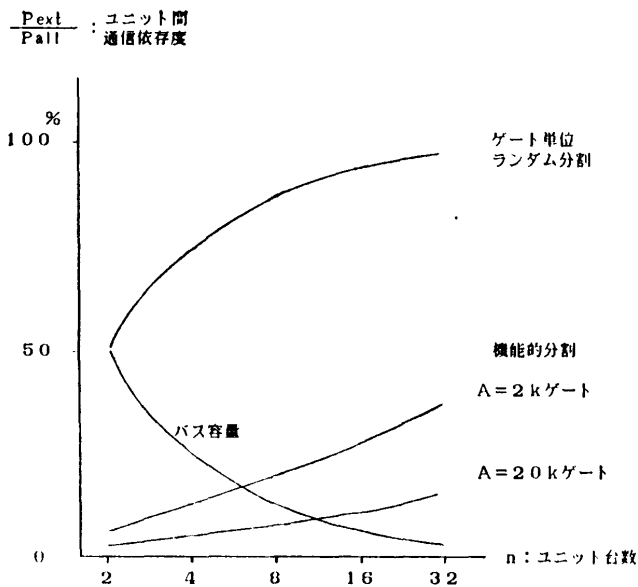


図1 ユニット間通信への依存度  
Fig. 1 Inter-unit communication ratio.

(1) ブロックに基づくトップダウン分割

設計情報に含まれる、各機能ブロックの階層関係を利用して、トップダウンに分割を行う。ブロックを単位とすることにより、ユニット間通信量を減少させる。また、計算量低減によりソフトウェア処理時間を短縮し、同一機能ブロックのデータを共有することによりメモリ量を削減する。

(2) 並列処理効率の向上

論理シミュレーションは、通常、1回路に対して何度も繰り返し行われる。よって、前回の実行結果から、各プロセッサで発生したイベント数を抽出し、回路分割において考慮することにより、並列処理効率の向上を図る。

2.2.2 ハードウェア仕様

LSI の論理およびタイミングチェックを対象とするため、ゲートレベルでイベント法によるシミュレーションを行う。MOS 素子について、素子単位で正確に評価する手法を提案し、ゲートレベルシミュレーションエンジン上での実現を容易とする。

イベント法を用いた場合、負荷のばらつき発生が問題となる。しかし、通常イベント発生率は数 % 以下であり<sup>18)</sup>、2.1 節における評価結果より、イベント法による処理速度向上効果がばらつきによる速度低下を上回ると判断した。以下に仕様を示す。

(1) シミュレーション対象

ゲート、メモリ、双方向性 MOS 素子が基本要素として扱われる。

(2) 容量

数万ゲート規模の ASIC とそれを含む回路のシミュレーションのため、ゲート素子、MOS 素子合計で 64,000 素子と 1 Mbyte までのメモリを評価する。

(3) 速度

イベント法に基づく処理を行い、1 MHz のクロックで 100 万イベント/秒の速度を得る。数万ゲートの回路について、数千パターンのシミュレーションを数秒で終わるとの基準より、処理速度を決定した。クロックについてはエンジン製作上の都合により決定した。

(4) 信号値

以下の状態値および信号強度の組合せにより表現する。

- 状態値 : 4 値 (0, 1, X, Contention: 競合).
- 信号強度 : 4 段階 (D: 駆動, R: 抵抗負荷, B: 大容量性ハイインピーダンス, F: 小容量性ハイインピーダンス) × 2 (確定/不定フラグ).

蓄積される電荷容量によってハイインピーダンス状態を区別することにより、プリチャージ等のダイナミックな回路のモデル化を容易とする。各信号強度について、確定/不定フラグを用意し、ゲート入力状態が不定のトランジスタの影響の有無を示す。

(5) 遅延

立ち上がり、立ち下がり遅延を与える。

2.3 構成

図 2 にシステム構成を示す。TASSE は 4 台の並列処理ユニットから構成される TASSE エンジンと、前処理、制御、結果解析を行う支援ソフトウェアの二つの部分より成る。

2.3.1 ソフトウェア構成

論理情報入力システムによって対話的に入力された回路情報は、各ブロック単位で外部端子の内部接続を示すバイナリデータにコンパイルされる。

次に、回路をシミュレーションエンジン上の並列処理に適するよう、四つの部分回路に分割する。リンク

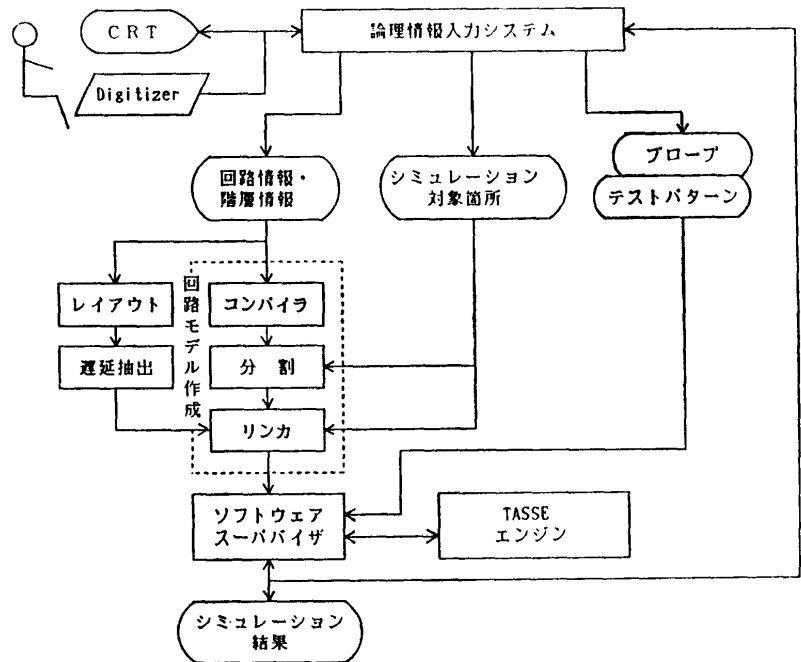


図 2 システム構成 Fig. 2 System configuration.

は分割結果と論理接続情報、配線遅延情報をもとに、シミュレーションモデルを作成する。スーパーバイザは、シミュレーション実行制御等の対話処理を司る。ソフトウェア処理において、各ブロックについての内部接続データを同一機能のブロック間で共有することにより、必要メモリ量を削減する。

回路分割手法について以下に述べる。

階層化された回路情報におけるブロックを、ゲート数制限を満たす限りそのまま同一プロセッサに割り付けることにより、プロセッサ間で分断される信号線本数を削減する。

また、負荷の均一化については、前処理段階だけで評価を行うことは困難であると同時に処理時間増加の要因となる。一方、シミュレーションは、1回だけで終わることはなく、通常、同一回路に対して誤りが発見されなくなるまで繰り返し実行する。よって、初回か2回目以降かにより、異なる二つの分割手法を適用する。

#### i) 初回

初めてシミュレーションを行う場合は、ゲート数とブロック間の結線本数を評価する。

#### ii) 2回目以降

ゲート数・結線本数だけでなく、イベント数をも考慮した分割を行う。

ii) について以下に述べる。前回のシミュレーション結果より、各機能ブロック：Bにおいて発生したイベント数を抽出する。シミュレーション実行中に、各ゲートに生じたイベント数を積算し、各機能ブロックごとにまとめることにより、抽出を行う。抽出されたイベント数を利用し、各ユニットにイベントが均等に割り当てられるように回路分割を行う。ただし、回路やテストパターンの変更に対応するため、ゲート数の均一化についても同時に考慮する。イベント数とゲート数の均一化のため、各ブロックについて以下の式により決定される割付けコスト： $CE_B$ を求め、その和が均一となるように各ユニットに割り付ける。

$$CE_B = W_{E/T} \cdot E_B + (1 - W_{E/T}) \cdot G_B \quad (4)$$

$E_B$  は発生したイベント数、 $G_B$  はブロック B の内部ゲート数を示す。 $W_{E/T}$  は 0~1 の値をとり、イベント数に対する重みづけを表す。テストパターンや回路に対する変更が少ない場合は、 $W_{E/T}=1$  としてイベント数を重視した分割を行い、変更が大きい場合にはゲート数を重視した分割を行う。

アルゴリズムを以下に示す。

- Step 1) 各ブロックについて割付けコスト  $CE_B$  を計算
- Step 2) ブロックを  $CE_B$  の大きいものから順に並べ、“Unassigned Block List (UBL)” とする。
- Step 3) 既割付けコストの最も小さいプロセッサ  $P_{Emin}$  を見つける。
- Step 4) UBL より 1 ブロックを取り出し、そのプロセッサに割り付ける。割付けによりゲート数がプロセッサに定められたゲート制限を越えると、ブロックを再分割し、UBL に再登録の後 Step 3) へ。
- Step 5) そのブロックと最も多くの接続を持つプロセッサ  $P_{Cmax}$  を見いだす。
- Step 6)  $P_{Emin}$  と  $P_{Cmax}$  から、ゲート数制限と、すでにそのプロセッサに割り付けられたブロックとの結線本数を考慮して割付け対象プロセッサを決定する。
- Step 7) UBL が空でなければ Step 3) へ。
- Step 8) ブロック内部の未割付けゲートに対し Step 3)~Step 6) と同様の手法により割付けを行う。

### 2.3.2 ハードウェア構成

TASSE エンジンのハードウェア構成を図 3 に示す。以下の四つの部分より成り立つ。

#### (1) 処理ユニット

4 台の処理ユニットがゲート素子や MOS 素子について並列してシミュレーションを実行する。処理ユニット内の構成を図 4 に示す。処理ユニット内は 7 段のパイプラインにより構成され、1 クロックごとにゲート評価を行う。Zycad LE<sup>4)</sup> は 1 ゲートの評価に 11

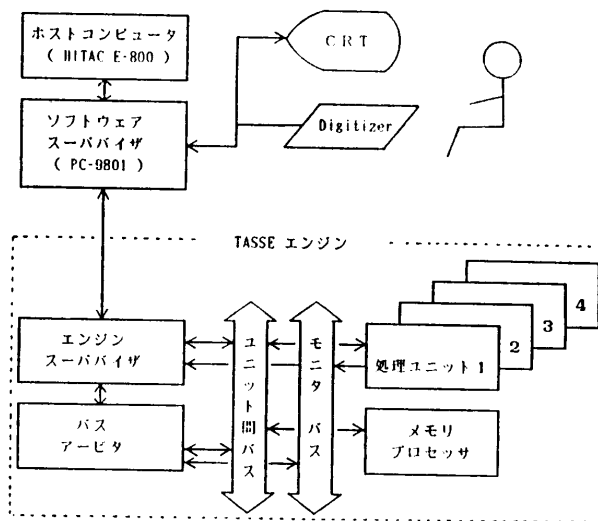


図 3 TASSE エンジン構成

Fig. 3 TASSE engine configuration.

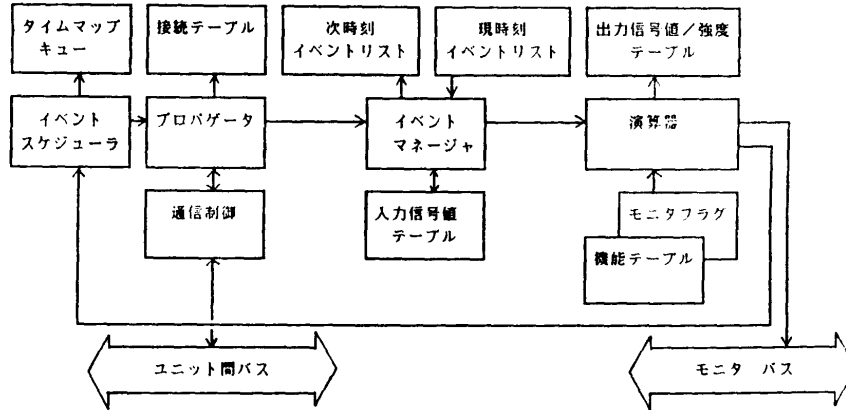


図 4 処理ユニット内構成  
Fig. 4 Processing unit.

クロックを要しているが、ハードウェア構成の工夫により処理効率向上を図った。

タイムマッピングキューは、イベントスケジューリングに際し種々の遅延を効率よく扱うために用いられている。プロパゲータは、ある素子に生じたイベントを同一信号線で接続する他の素子の入力に伝播する。イベントマネージャは、次時刻に評価されるべきイベントをリストに格納するとともに、現時刻で評価すべきイベントを取り出す。演算器は、ゲートおよび MOS 素子の評価を行い、その出力に信号値変化が生じたか否かをチェックする。プローブとして指定された素子に生じたイベントは、モニタバスを介してホストコンピュータに送られる。

(2) メモリプロセッサ

メモリプロセッサは、ROM/RAM のような記憶素子や、PLA のような array logic の評価を行う。

(3) エンジンスーパーバイザ

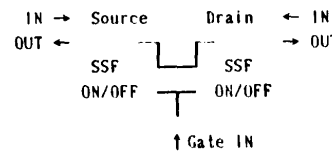
エンジンスーパーバイザはハードウェアを制御する。ソフトウェアによるスーパーバイザとの通信によるデータ授受を行う。

3. MOS 素子評価手法

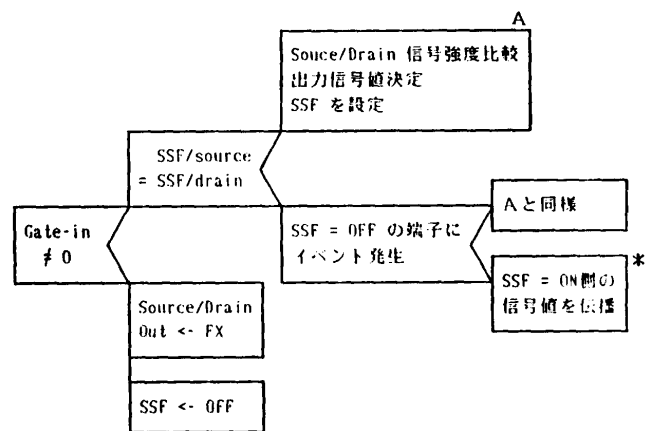
本論文では、素子単位の評価に基づき仮想素子を用いずに MOS 素子評価を行うアルゴリズムを提案する。必要十分な X 評価を行うことにより、過小・過大のいずれにも属さない正確な評価を可能とする。素子単位評価に基づくため、専用ハードウェア上での実現に適する。

3.1 MOS モデルと評価手法

図 5 a) に本手法で適用する MOS モデルを示す。Source/Drain については入力/出力の信号値を別々に保持する。前回の評価による信号伝播方向を Signal Source Flag (SSF) として保持し、イベントが生じた際の伝播方向判断に用いることを特徴とする。MOS 素子、トライステート出力等が接続するトライステートラインは素子と同様に扱われ、各端子出



a) MOS 素子モデル



b) MOS 素子評価アルゴリズム

図 5 MOS 素子モデルと評価アルゴリズム  
Fig. 5 MOS transistor model and evaluation algorithm.

力からネット全体の信号値・信号強度を求める。

図5 b)に MOS 素子の評価手法を示す。Gate 端子入力が ON で Source/Drain の SSF が互いに等しい場合は両端子間の信号強度比較によって伝播方向を決定する。SSF が ON, すなわち前回に信号値伝播の源となった端子側だけにイベントを生じた場合、たとえば信号強度が他方の端子より低い場合でもその信号値を他方へ伝える。

図6にトリステートラインの評価手法を示す。通常、信号強度の最も高い端子の信号値が、他の端子に伝播される。トリステートラインについても SSF を用意し、SSF/ON の端子にのみイベントが生じた場合には、そのまま伝播する。

MOS 回路遅延はトリステートラインに与える。ただし、前回より低い信号強度のイベントが発生した場合は0ディレイで伝播し、他の素子・信号線に伝播された前回の信号値を打ち消す。次に、各トリステートラインにスケジュールされたこの打ち消しイベントの評価によって、各信号線の値が更新される。これは、実際の回路において電流源側から値が決定される現象に対応している。

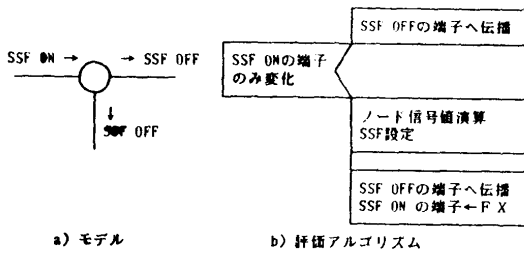


図6 トリステートラインモデルと評価アルゴリズム  
Fig. 6 Tri-state line model and evaluation algorithm.

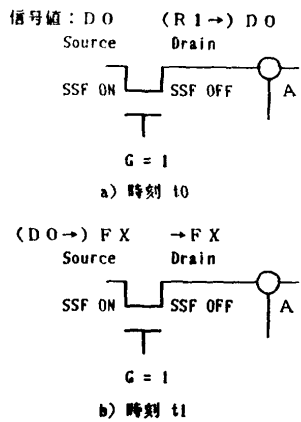


図7 MOS 素子評価例  
Fig. 7 Evaluation example.

図7に以上の手法による評価例を示す。図7 a)において Drain 側の信号値が R1 の値をとるとき、Source 側の信号値が D0 に変化したとする。信号強度比較により Source 側が信号源と判断され、Source 側の SSF が ON にセットされる。次に図7 b)において Source 側が最も低い信号強度である FX に変化したとする。これは図5 b)の\*で示した処理を実行する条件に相当し、SSF/ON である Source 側の信号値 FX をそのまま Drain 側に伝播する。次に Drain に接続する信号線に相当するノードAの評価を行う。F (小容量性ハイインピーダンス) よりも高い信号強度を持つと判断されると、その値が Drain 側にイベントとして返される。

以上のように、SSF の導入によって Source/Drain 各端子間の単純な信号強度比較によって招く誤った信号伝播を防ぐことができる。

### 3.2 不定値 X の扱い

#### (1) 信号競合による X 発生

MOS 素子やトリステートラインにおいて一時的に1と0が競合し、不定値 X を生じることがある。特

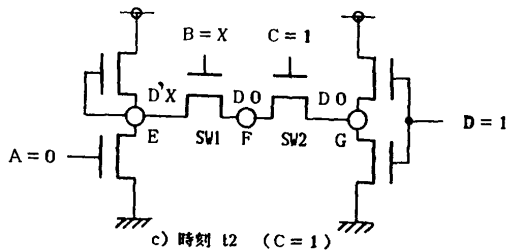
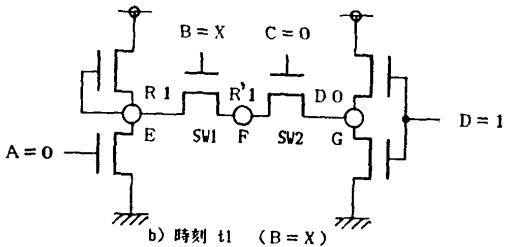
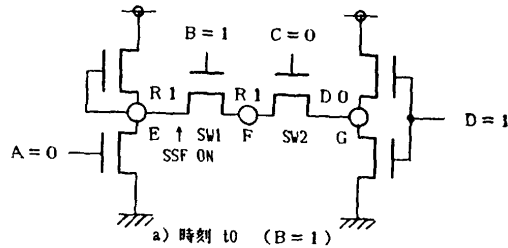


図8 ゲート端子入力 X の評価  
Fig. 8 Evaluation of X on MOS gate terminal.

に CMOS 回路では MOS トランジスタの ON/OFF に伴う瞬間的競合は不可避である。この不定値の処理を誤ると、回路のほとんどの部分に X が伝播することにより、意味のないシミュレーション結果となる。

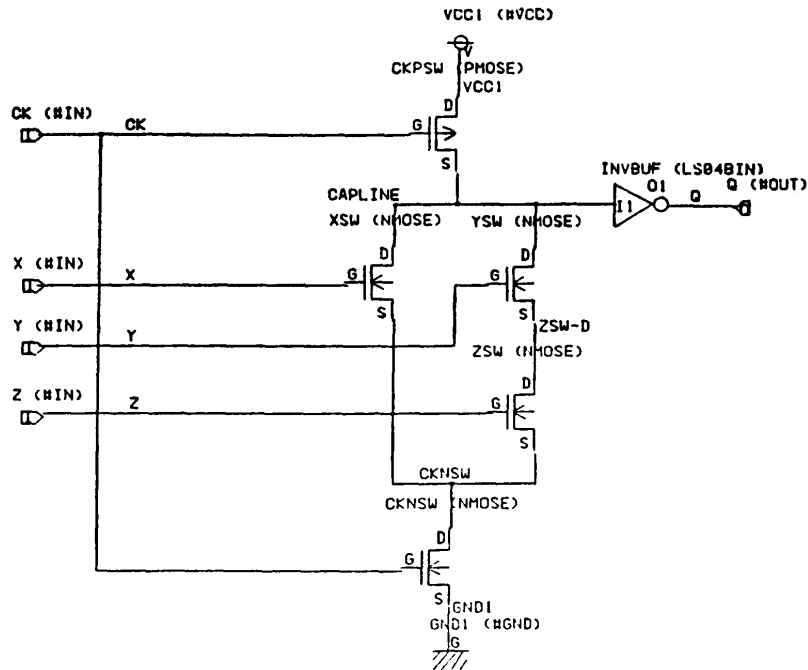
TASSE では、このような状態を示すために新しい

表 1 ソフトウェアステートメント数  
Table 1 Software statement counts.

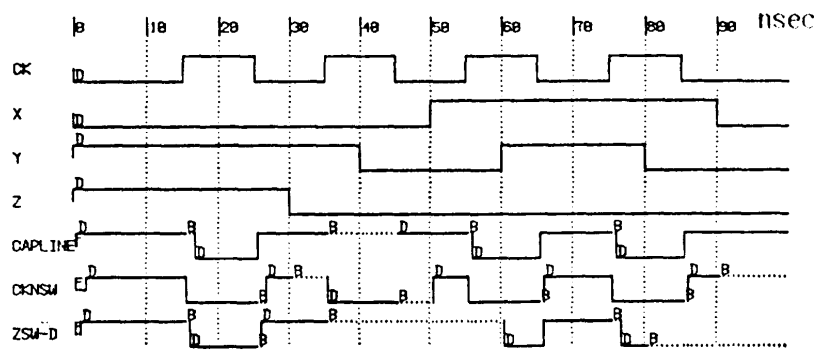
プロセス	言語, ステートメント数	ホストコンピュータ
回路モデル変換	Pascal 8,400 行	32 bit ミニコンピュータ HITAC E-800 (0.4MIPS)
論理情報入力, スーパーバイザ	Pascal 30,000 行	PC 9801/VM 2

表 2 MOS 回路例  
Table 2 MOS circuit examples.

回路名	トランジスタ数
1. CMOS ドミノ回路	7
2. レシオ型ラッチ	10
3. クロックドラッチ	12
4. マルチプレクサ	12
5. 直列スイッチ	10
6. 双方向バス通信	50
7.ブリッジ回路	6
8. NAND <sup>14)</sup>	3
9. 閉ループ回路 <sup>14)</sup>	4
10. 抵抗性直列スイッチ <sup>14)</sup>	3
11. インバータスイッチ <sup>14)</sup>	4



a) CMOS ドミノ回路



b) TASSE による評価結果

図 9 CMOS ドミノ回路と評価結果

Fig. 9 CMOS domino circuit and simulation result.

状態値 C: Contention (競合) を導入している。実際の回路では、ノードの充放電に要する遅延時間内の一時的信号競合は問題を生じず、連続的に値が変化する。よって、遅延時間内に C が他の値に置き換えられた場合には、その一時的な C を無視する。遅延時間を越える C が直接ゲート入力等に接続されると X と同じく不定値として扱われる。

#### (2) MOS 素子ゲート入力の X

MOS 素子ゲート入力に X が発生した場合、Source/Drain の値を一意的に決定するのは困難である。ゲート入力に X であるすべての素子 TXi について 0/1 の 2 通りを考え、一致しない箇所を X とみなすのが最も正確である。しかし、 $O(2^n)$  の評価を必要とし、現実的でない。Bryant は 0/1 の 2 通りについて、それぞれを信号値とするノードの個数を最大とするように TXi の値を定め、不一致箇所を X としている<sup>14)</sup>。

本手法では、以下の 2 点によってゲート入力の X を素子単位で正しく評価する。

i) SSF を利用することにより不定信号強度の伝播方向を限定する。

ii) 可能性のある最高の信号強度を持つ状態値を保持しつつ伝播することにより、他の信号によって強度が確定される場合を正しく評価する。

図 8 に示す例について述べる。

#### i) 時刻 $t_0$

$B=1$  であり、SW 1 の Source/Drain に関する信号強度比較の結果、信号値 R 1 が Source 側から Drain 側へ伝播される。SW 2 は OFF なので FX を出力し、ノード F に対する評価の結果、信号値は R 1 と判断される。

#### ii) 時刻 $t_1$

$B=X$  に変化したとする。SW 1 に関して SSF/ON 側の信号値 R 1 がゲート入力不定の MOS 素子を通ることにより、R'1 なる信号値としてノード F に伝えられる。R' は最高で R/最低 F の信号強度を意味する。SW 2 の状態に変化はなく、ノード F の信号値は R'1 と判断される。

#### iii) 時刻 $t_2$

$C=1$  に変化したとする。ノード F に対する評価の結果、不定信号強度を持つ信号値 R'1 は、より高い信号強度を持つ信号値 D 0 に吸収され、D 0 がノード F の新しい信号値と判断される。この値は、信号源でない SW 1 の Drain に伝播される。SW 1 の Source: R 1 と Drain: D 0 の強

度比較により、不定強度を含む信号値 D'0 が SW 1 の Source 側へ伝播される。ノード E に R 1 と D'0 が入力される。最高の信号強度は D' である。状態値については、不定信号強度を含む信号値: D'0 と含まない信号値: R 1 が異なる値をとるため、X と判断される。

## 4. 実 験

性能評価項目と実験による評価結果を示す。

ソフトウェア各部のステートメント数を表 1 に示す。ハードウェアは、通常の LS-TTL IC と、64 k SRAM (120 ns)、一つのゲートアレイで構成されている。

各処理ユニットは、400 個の LS-TTL パッケージ、600 kbyte の SRAM とゲートアレイを含む。エンジンスーパーバイザとバスアービタは、それぞれ 100 個程度の LS-TTL IC により構成される。クロックは、現在 1 MHz のメインクロックと、ロジック制御、メモリ書き込みのために 2 相の 2 MHz サブクロックを用いている。

### 4.1 評価項目

次の 3 項目について本システムを評価した。

TASSEによる  
遅延評価

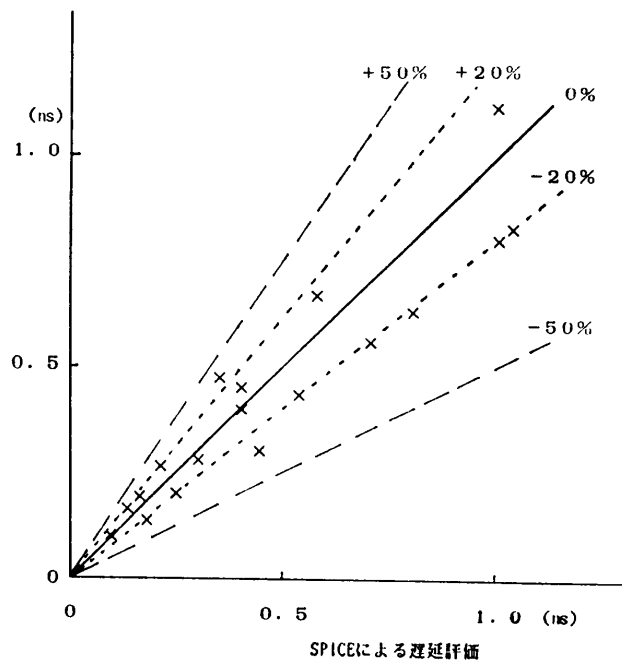


図 10 SPICE, TASSE 遅延時間評価結果

Fig. 10 Delay time estimation result of SPICE and TASSE.



- (1) MOS 素子評価の正当性
- (2) 処理速度
- (3) バス通信

4.2 実験結果

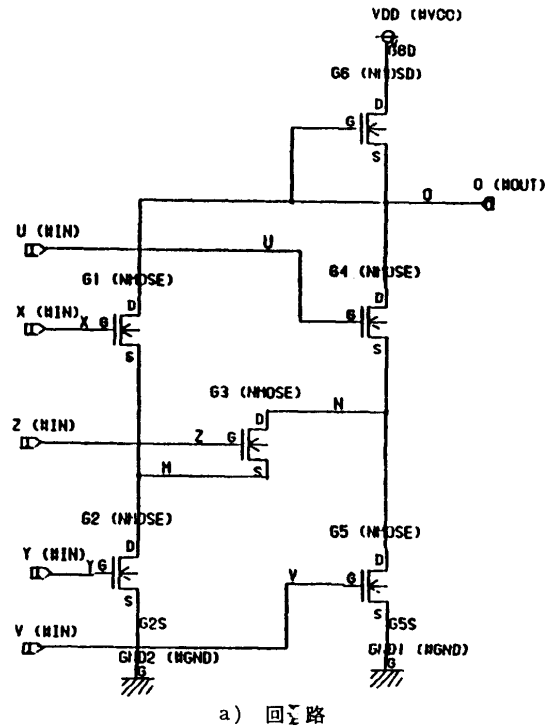
4.2.1 MOS 素子評価

詳細遅延を扱うシミュレータ<sup>13)</sup>や、遅延モデル<sup>19)</sup>の評価には、回路レベルシミュレータ SPICE<sup>20)</sup>を、比較対象に用いることが一般的である。よって、表 2 に示す MOS トランジスタを用いた典型的な回路例について SPICE による結果との比較を行った。CMOS ドミノ回路は、32 ビットマイクロプロセッサに多用されており<sup>21)</sup>、電荷を利用したダイナミック動作に対する評価実験のために選んだ。また、Bryant が文献<sup>14)</sup>で示した、ループ回路、X 評価、評価順序が問題となる各例題 (Fig. 2, 3, 6, 7) についても実験を行った。遅延と信号強度評価、X 評価についての実験結果と計算量に対する考察を示す。

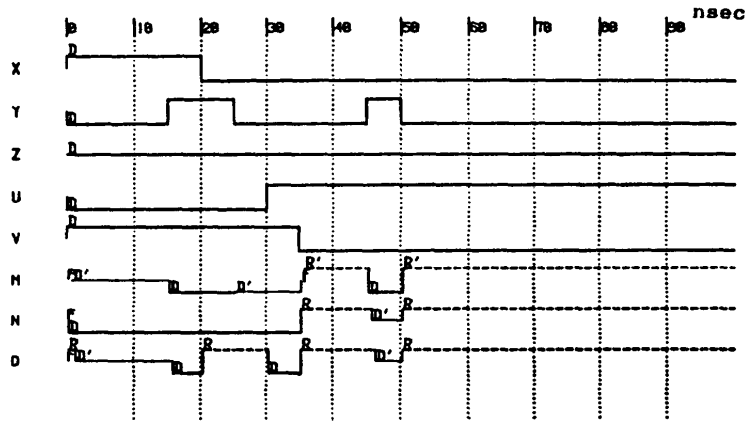
CMOS ドミノ回路について、図 9 a) に回路図を、図 9 b) に CRT 上に表示された TASSE の結果を示す。SPICE による結果との比較により、論理機能が正しく反映されていることが確認された。プリチャージされるトライステートライン CAPLINE についてタイミング評価検討の結果、SPICE で 1.8 ns, 1.1 ns と評価された遅延が、TASSE では 1.8 ns, 1.2 ns と評価され 10% 程度の誤差で遅延評価が可能となった。

各回路に対する SPICE と TASSE による遅延評価結果の関係を図 10 に示す。80~90% の例について 20% 以内の誤差で遅延評価可能であることが確認された。20% 以上の誤差については、各ノードに接続する静電容量、駆動源インピーダンスの動的変化を考慮しないことが誤差要因と考える。

また、ループを含むブリッジ回路に対する、不定値入力評価結果を示す。回路図を図 11 a) に、シミュレーション結果を図 11 b) に示す。太い線は確定値、細い線は不定信号強度を含む値を示す。双方向に動作



a) 回路



b) 結果

図 11 不定値入力に対するシミュレーション結果 (ブリッジ回路)  
Fig. 11 Simulation result of bridge circuit with X input.

表 3 ソフトウェア処理時間  
Table 3 Software processing time.

	回路例 1	回路例 2	回路例 3	回路例 4
ゲート数	60,500	9,200	4,300	2,200
コンパイル (sec)	26	105	71	36
分割	8	8	5	5
リンク	306	50	21	9
計	340	163	97	50

するトランジスタ G3 の入力 Z に不定信号値を与えた結果、出力 O について以下の正しい結果を得た。

①状態値 0 (時刻 15~20, 30~35 ns)  
 $X=Y=1$ , または  $U=V=1$  の条件による.

②状態値 1 (時刻 20~30, 35~45 ns)  
 $X=U=0$ , または  $Y=V=0$  の条件による.

③不定値 X (時刻 0~15, 45~50 ns)  
 $X=V=1$  かつ  $Y=U=0$ , または  $X=V=0$  かつ  $Y=U=1$  の条件が成立する場合, 出力 O について D 0 と R 1 の 2 通りが考えられる. よって, 状態値を不定とするのが妥当である. 実験結果は D' X を示しており, 正しい結果といえる.

また, Bryant が示した各例題についても, 正しい結果を得ることを確認した.

次に, 計算量について, MOSSIM<sup>10), 11)</sup> と比較・評価する. MOSSIM では, 1 イベントにつきゲートレベルシミュレータの 2~3 倍の処理時間を要していた. TASSE では反復動作を含まない素子単位の評価を適用している. よって, MOS 素子は一般のゲートと同じ時間で処理できる. X 評価について, MOSSIM ではバス単位で 2 通りの処理を行う必要がある. TASSE では, フラグによって判別しているのので 1 回の処理で済む.

以上の結果より, TASSE の MOS 評価手法が, 正確かつ効率よく MOS のタイミング, 論理動作を評価できることが示された.

4.2.2 処理速度

ソフトウェア処理時間に関する結果を表 3 に示す. 回路例 1 は, カウンタとレジスタファイルより成る回路, 回路例 2~4 は TASSE エンジン処理プロセッサ回路全体および規模の異なる部分回路を示す.

回路情報コンパイルに要する時間については, すべての部分回路を回路図面情報からバイナリファイルへ変換した場合の値を示す. 分割は短時間で行われ, リンク処理時間についても回路規模に対してリニアオーダで変化している.

ハードウェア処理速度と回路分割パラメータとの関係の評価した. 図 12 にイベント係数比率  $W_{E/T}$  の変化に対するハードウェア処理速度 (図 12 a) と稼働率 (図 12 b) の変化を示す. それぞれについて, テストパラメータの変更がない場合 (1k ステップ) と, ステップ数を半分に短縮した場合について示す.  $W_{E/T}$

= 0.5 の場合, テストパラメータの変更の有無にかかわらず, 処理速度を約 16%, 稼働率を約 8% 向上させる. よって, イベント数考慮による負荷の均一化が, 並列処理効率向上に効果のあることが示された.

4.2.3 ユニット間通信

並列処理プロセッサ間の通信によるオーバーヘッドを評価した. 表 4 に回路例 2 について, 複数のユニットで共有される信号線, バスを介して通信されるイベント数, バス待ち状態に陥るイベント数を示す. バス待

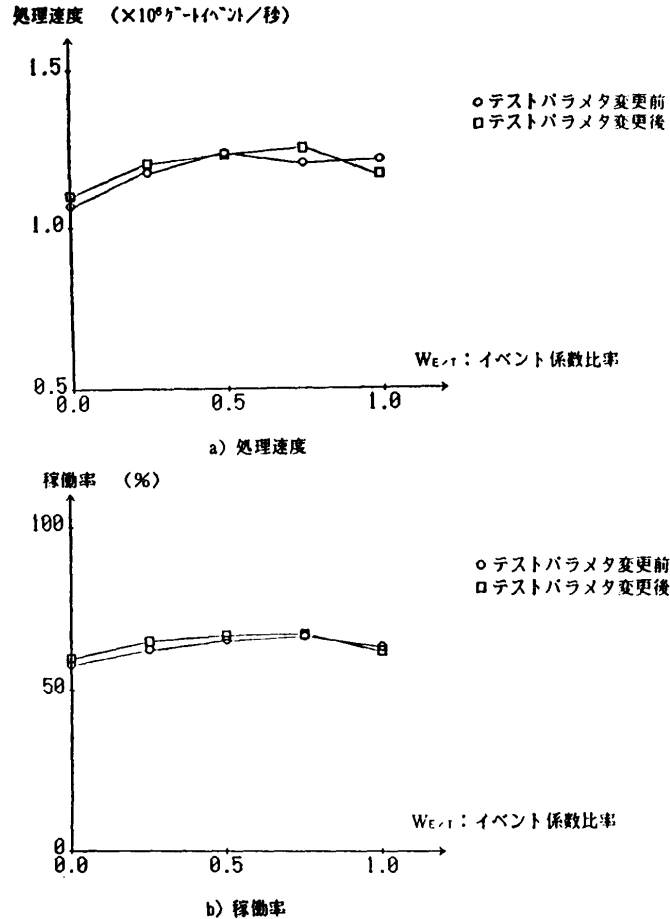


図 12 イベント係数比率による処理速度と稼働率の変化  
 Fig. 12 Effects of  $W_{E/T}$  on processing rate and processing efficiency.

表 4 バス通信 (回路例 2)  
 Table 4 Bus communication (Example 2).

ユニット間 共有信号線 本数	ユニット間 非共有信号線 本数	バス通信イベント数	
		全イベント数	バス待ちクロック数
1,330	6,950	5,400	220
		74,000	105,400

ちの割合は 0.2% であった。階層構造を利用した分割によってユニット間で共有される信号線数は低く抑えられ、バス交信がボトルネックとはならないことが示された。

各回路例に対し、実際のユニット間通信への依存度を 2.1 節で求めた評価式で求めた値と比較した。テストパターンによって異なるが、誤差はゲート単位分割に関して 20% 程度、機能ブロック単位分割に対して 50% 程度であることを確認した。

以上の実験結果より、以下の 2 点が示された。

(1) 本手法により正確かつ効率のよい MOS タイミング、信号強度評価が可能となった。ただし、動的な負荷容量・駆動インピーダンスの変化による影響の評価についてはさらに検討を要する。

(2) 実行結果を利用した機能ブロック単位の分割により、通信量削減と並列処理効率向上を同時に実現した。

## 5. むすび

われわれは、直前の信号値伝播方向を保持することにより、MOS トランジスタを含む回路のタイミング、信号強度、不定値 X 等を正しく評価する手法を提案した。さらに、QTAT で対話的に回路の詳細なタイミング評価を行うため、提案した手法を用いたシミュレーションエンジン TASSE を開発した。また、通信量削減と負荷の均一化のため、実行結果を利用した機能ブロック単位の回路分割手法を提案した。

実験により、MOS 素子評価の正当性と、回路分割手法の効果を確認した。

今後の課題として、回路パラメータの動的変化への対応、CPU を含んだ回路のシミュレーションへの適用、シミュレーション用パターン評価手法の開発等が挙げられる。

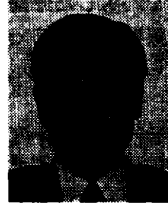
## 参 考 文 献

- 1) Pfister, G. F.: The Yorktown Simulation Engine: Introduction, *Proc. 19th Design Automation Conference*, pp. 51-54 (1982).
- 2) Sasaki, T. et al.: HAL; A Block Level Hardware Logic Simulator, *Proc. 20th Design Automation Conference*, pp. 150-156 (1983).
- 3) Takasaki, S. et al.: A Mixed Level Hardware Logic Simulation System, *Proc. 23rd Design Automation Conference*, pp. 581-587 (1986).
- 4) Zycad Corp: LE 1000 Series Logic Evaluator/General Description Manual (1984).
- 5) Blank, T.: A Survey of Hardware Accelerators Used in Computer-Aided Design, *IEEE Design & Test of Computers*, Vol. 1, No. 7, pp. 21-39 (1984).
- 6) Sherwood, W.: A MOS Modeling Technique for 4-state true-value Hierarchical Logic Simulation or Karnough Knowledge, *Proc. 18th Design Automation Conference*, pp. 775-781 (1981).
- 7) McDermott, R. M.: Transmission Gate Modeling in an Existing Three-Value Simulator, *Proc. 19th Design Automation Conference*, pp. 678-681 (1981).
- 8) Hayes, J. P.: A Logic Design Theory for VLSI, *Proc. 2nd Caltech Conference*, pp. 455-476 (1981).
- 9) Kawai, M. et al.: An Experimental MOS Fault Simulation Program CSASIM, *Proc. 21st Design Automation Conference*, pp. 2-9 (1984).
- 10) Bryant, R. E.: A Switch-Level Model and Simulator for MOS Digital Systems, *IEEE Trans. Comput.*, Vol. C-33, No. 2, pp. 160-177 (1984).
- 11) Bryant, R. E.: MOSSIM: A Switch-Level Model Simulator for MOS LSI, *Proc. 18th Design Automation Conference*, pp. 786-790 (1981).
- 12) Bryant, R. E. et al.: COSMOS: A Compiled Simulator for MOS Circuits, *Proc. 24th Design Automation Conference*, pp. 9-16 (1987).
- 13) Adler, D.: SIMMOS: A Multiple-Delay Switch-Level Simulator, *Proc. 23rd Design Automation Conference*, pp. 159-163 (1986).
- 14) Bryant, R. E.: A Survey of Switch-Level Algorithms, *IEEE Design & Test of Computers*, Vol. 4, No. 4, pp. 26-40 (1987).
- 15) 富田昌宏ほか: C-MOS ゲートアレイを対象としたシミュレーションエンジン, 第 34 回情報処理学会全国大会論文集, pp. 2005-2006 (1987).
- 16) 小池ほか: HAL: 論理シミュレーションマシンの評価, 情報処理, Vol. 26, No. 11, pp. 1331-1341 (1985).
- 17) Landman, B. S. and Russo, R. L.: On a Pin versus Block Relationship for Partitions of Logic Graphs, *IEEE Trans. Comput.*, Vol. C-20, No. 12, pp. 1469-1479 (1971).
- 18) Bailey, M. L. et al.: An Empirical Study of On-Chip Parallelism, *Proc. 25th Design Automation Conference*, pp. 160-165 (1988).
- 19) Chang, F. C. et al.: An Accurate and Efficient Gate Level Delay Calculator for MOS Circuits, *Proc. 25th Design Automation Conference*, pp. 282-287 (1988).
- 20) Nagel, L. W.: SPICE 2: A Computer Program to Simulate Semiconductor Circuits, Memorandum No. ERL-M 520, Univ. of California, Berkeley, May 9 (1975).
- 21) Chen, H. Y. et al.: Performance Optimization for Domino CMOS Circuit Modules, *Proc. International Conference on Computer Design: VLSI in Computers & Processors*, pp. 522-525 (1987).

(昭和 63 年 5 月 30 日受付)  
(平成 元年 1 月 17 日採録)

**小田原豪太郎** (正会員)

1944年生. 1967年東京大学工学部精密機械工学科卒業. 1972年東京大学工学部精密機械工学科講師. 1985年より同大学同学科教授. 工学博士. 専門は情報処理機器, CAD/CAM/CAT. 電子情報通信学会, ACM, IEEE, プリント回路学会各会員.

**富田 昌宏** (正会員)

1960年生. 1983年東京大学工学部精密機械工学科卒業. 1986年東京大学工学部精密機械工学科助手. 工学博士. 研究テーマは論理装置のCAD, 特に論理設計の支援. IEEE, プリント回路学会各会員.