

## シミュレーション高速化のためのゲート／トランジスタ回路からの標準HDL-RTLモデルの自動生成

本村哲朗† 新舎隆夫†† 小畑誠‡ 小島智‡

† (株)日立製作所 中央研究所

〒185 東京都国分寺市東恋ヶ窪1丁目280番地

†† (株)日立製作所 汎用コンピュータ事業部

〒259-13 神奈川県秦野市堀山下1番地

‡ (株)日立製作所 半導体事業部

〒185 東京都小平市上水本町5丁目20番1号

顧客サイトでのASIC論理シミュレーションの高速化のため、ゲート／Tr. レベルのモジュールから、Verilog HDL/VHDL記述のRTLモデルを生成するHILEXTの開発を行った。HILEXTは、ゲート／Tr. 回路をRTL論理に変換するRTL論理抽出に加え、信号共有論理のグループ化と順序論理のタイミング最適化により、シミュレーション高速化のためのRTLモデル最適化を行う。生成モデルの性能評価として、同一シミュレータ上のゲートTr. との性能比較とゲート／Tr. 専用シミュレータ上のゲート／Tr. との性能比較を実施し、ASIC 7モジュール平均で、前者は1.4倍、後者は5倍以上の高速化効果を確認した。

## Automatic Generation of RTL Model in Standard HDL from Gate/Transistor Circuits for High-Speed Simulation

Tetsuroo Honmura †, Takao Shins ha † †, Makoto Obata ‡, and Satoshi Kojima ‡

† Hitachi, Ltd. Central Research Laboratory

1-285, Higashi-koigakubo, Kokubunji-shi, Tokyo 185, Japan

† † Hitachi, Ltd. General Purpose Computer Division

1, Horiyamashita, Hadano-shi, Kanagawa, 259-13, Japan

‡ Hitachi, Ltd. Semiconductor & Integrated Circuits Division

20-1, Josuihon-cho, Schome, Kodaira-shi, Tokyo 187, Japan

For high-speed ASIC logic simulation on user's site, we develop HILEXT, Verilog HDL/VHDL RTL model generator from gate/transistor level modules. HILEXT not only extracts RTL logic from gate/transistor circuits but also optimizes RTL model for faster logic simulation by grouping signal-sharing logics and by timing optimization of sequential logics. We evaluate RTL model's speed-up effect by two performance comparison with gate/transistor performance 1) under same simulator platform with RTL model's one and 2) under special purpose gate/transistor simulator platform. As an average performance of 7 ASIC modules, We obtain 1.4 times performance in 1) and more than 5 times in 2).

## 1. はじめに

近年、ASIC設計では、ASICメーカーがCPUコア等のモジュール群を顧客に提供し、顧客はモジュールの組合せに顧客固有の設計論理を付加するという手法が主流となっている。この手法を適用する際の一つの課題は、モジュール群を顧客サイトの市販論理シミュレーション環境下で高速に動作させ、高速なASIC論理シミュレーション環境を提供することである。しかし、モジュール群は、ゲート/トランジスタ（以後Tr.と略す）回路で人手設計されたマクロ部を含む等の理由から、ゲート/Tr.回路をマスタとしており、シミュレーション速度が遅いという問題があった。

上記の課題を解決するために、我々は、自社言語で開発されたゲート・Tr.レベルのモジュールを、Verilog HDL/VHDL(以下、標準HDLと呼ぶ)記述の高速なRTLモデルに変換する(図1.1) HILEXT (High abstraction Level model generation system from gate-transistor circuits by function EXtractor and Timing optimizer)の開発を行った。標準HDLは、現在、大半の市販論理シミュレータで採用されているため、これにより、上記の課題は解決される。

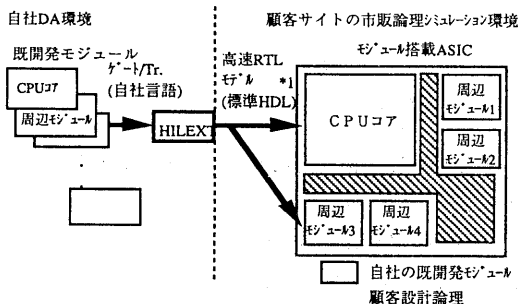


図1.1 HILEXTの目的 \*1:Verilog HDL/VHDL

これまで、ゲート/Tr.レベルの回路から上位レベルのモデルを生成するアプローチとして、トランジスタ回路からのゲートモデルの生成[1,2,3,4]、FF明示のゲート回路からのRTLモデルの生成[5]、およびTr.回路からのRTLモデルの生成[6]等が、行われてきた。RTLモデルの生成のうち、文献[5]では、

演算式等上位の記述レベルの生成まで行うが、詳細な人手指定情報が必要である。文献[6]では、シミュレーション高速化のため評価時間削減に有効な真理値表でRTLモデルの生成を行うが、真理値表を用いるためシミュレーション時のメモリ量が大きくなる。

HILEXTは、FF不明示のゲート/Tr.混在回路から文献[6]と同等レベルのブール式とif文からなるRTLモデルの生成をほぼ全自動で行い、同時にシミュレーション高速化のために、イベント処理時間削減に有効なRTLモデル最適化を行う。

本稿では、RTLモデル最適化方式を中核に、HILEXTのRTLモデル生成方式について報告する。以下、第2章と第3章でHILEXTのシステム構成とRTLモデル生成方式の概要を述べ、第4章でRTLモデル最適化方式を述べ、第5章で評価を述べる。最後に第6章でまとめを述べる。

## 2. HILEXTのシステム構成

HILEXTのシステム構成を図2.1に示し、入出力を以下に述べる。

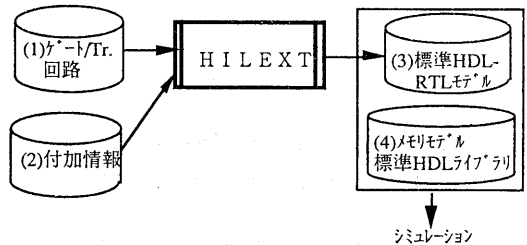


図2.1 HILEXTのシステム構成

### (1) ゲート/Tr. 回路

これは、自社言語で記述された既開発のゲート/Tr.レベルの回路である。

### (2) 付加情報

これは、モデルの機能実現、高速化、およびタイミング仕様実現を目的に、回路情報以外に人手で入力する情報である。

機能実現に必要な付加情報は、ディレイ利用回路セル名と論理強度利用回路セル名である。前者は、ワンショット発生回路のようにディレイ0では機能の実現

が不可能なセルの名称であり、後者は、出力の論理強度が通常より弱いセルの名称である。HILEXTは、0ディレイで論理強度が同一のモデルの生成を基本とするが、これらを利用してディレイと論理強度を考慮したモデルの生成も行う。

高速化に必要な付加情報は、クロック信号名である。これは、複数相の互いに重複のないクロックの信号名であり、第4章で述べるRTLモデル最適化で用いる。

タイミング仕様実現に必要な付加情報は、出力ディレイと入力タイミングチェック仕様である。前者は、出力信号に付加するディレイであり、後者は、入力信号に対するセットアップタイム違反等のタイミングチェック仕様である。HILEXTは、これらのタイミング情報を付加したモデルを生成する。

### (3) 標準HDL-RTLモデル

これは、HILEXTが生成する標準HDL記述のRTLモデルである。

### (4) メモリモデル標準HDLライブラリ

これは、自社言語に含まれるROM等のメモリ機能を標準HDLで記述したライブラリであり、システムが提供する。モデル使用時には標準HDL-RTLモデルとこのライブラリを合わせて用いる。

## 3. RTLモデル生成方式の概要

RTLモデル生成処理は、ゲート抽出、FF抽出、およびRTL変換の3つからなり、RTL変換は、RTL記述変換とRTLモデル最適化の処理を行う。機能的には、ゲート抽出、FF抽出、およびRTL記述変換のRTL論理抽出とRTLモデル最適化に分かれる。これらの処理を、表3.1に示し、以下に述べる。

### (1) RTL論理抽出

これは、ゲート/Tr.回路からRTL論理を抽出する処理であり、以下に述べる3つの変換からなる。

ゲート抽出は、文献[1,3,4]等と同様にTr.の回路種と回路接続形式等の回路形を認識して、ルールに基づきTr.群をゲートに変換する処理である。

FF抽出は、ゲート抽出と同様に回路形を認識して、ルールに基づきゲート群をFFに変換する処理である。

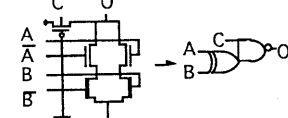
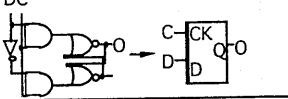
RTL記述変換は、組合せ回路をブール式に、順序

回路をif文に変換する処理である。

### (2) RTLモデル最適化

これは、RTLモデルのシミュレーション性能向上を図るために、信号共有論理のグループ化と順序論理のタイミング最適化を行う処理であり、第4章にて詳述する。

表3.1 RTLモデル生成の処理

No.	処理	内容
1	RTL論理抽出	ゲートTr.回路からRTL論理を抽出
1	ゲート抽出	Tr.群→ゲート変換をルールベースに処理 
2	FF抽出	ゲート群→FFをルールベースに処理 
3	RTL記述変換	組合せ回路→ブール式、順序回路→if文に変換
2	RTLモデル最適化	信号共有論理のグループ化と順序論理のタイミング最適化によりRTLモデルを最適化

## 4. RTLモデル最適化方式

以下では、第4.1節でRTLモデル最適化の考え方を述べ、以降の節で最適化方式を述べる。

### 4.1 RTLモデル最適化の考え方

RTLモデルが稼働する標準HDLシミュレータは、イベントドリブン方式であり、この方式の論理シミュレーション処理には、イベント伝搬処理と各要素の出力値を求める評価処理とがある。前者が処理の大半を占めており、これを削減するためにRTLモデルの最適化を行う。この最適化は、従来から同じ目的で行われてきた論理シミュレータの改良とは異なり、標準HDLの構文で許され、モジュール外部への応答が正しい範囲で行う論理表現の変形である。以下の2点に着目して、最適化を行った。

#### (1) 信号共有論理のグループ化

イベント数削減に最も有効な方法として、信号共有

論理をグループ化し、共有信号のイベント削減又は共通化を図ることを考えた。

グループ化の最も極端な方法は、コンパイル方式と同様に、全ての回路をグループ化することであるが、この方法はリセット動作等の非同期動作も含まれる対象回路には使えない。デジタル回路は、組合せ論理と順序論理に分かれており、組合せ論理は入力に変化するままに動作し、順序論理は動作タイミングが特定の信号で規定されるという違いがあるため、組合せ論理と順序論理に分けてグループ化を検討した。

組合せ論理は、入力の変化に応じて即座に動作し、イベントを順次伝搬していくため、イベント伝搬経路上論理のグループ化により対処することにした。但し、グループ化単位は、グループ化による評価時間の増大を考慮して検討する必要がある。詳細は、第4.2節で述べる。

順序論理は、大半の時間、コントロール信号の変化により動作タイミングが規定され、かつこの信号を共有する順序論理は多数あるため、コントロール信号共有順序論理のグループ化により対処することにした。詳細は、第4.2節で述べる。

#### (2) 順序論理のタイミング最適化

次に着目したのは、順序論理の起動タイミングである。順序論理の大半は、ほとんどの時間クロックに同期して動作し、クロック立上り前にデータは到着しているため、シミュレーションは、クロックの立上り時で充分である。しかし、実際の回路は、ゲート規模の小さいレベルタイプ（コントロールが1レベルの区間、データが変化すればそれに応じてデータを取り込むタイプ）の構成をしているため、シミュレーション時には、入力データの変化時にも無駄な起動を繰り返す。そこで、この無駄な起動を削減するレベル順序論理の起動タイミング最適化を行うことにした。詳細は第4.3節で述べる。

#### 4.2 イベント伝搬経路上論理のグループ化

これは、組合せ論理を対象とした最適化である。この最適化は、以下に述べる、組合せ論理単体のグループ化と、組合せ論理と順序論理のグループ化からなる。

##### (1) 組合せ論理単体のグループ化

イベント伝搬経路上の組合せ論理をグループ化するとイベント数は削減できるが、逆に評価時間が大きくなるというトレードオフがある。このトレードオフを考慮し、最適なグループ化単位を決定するのが、この最適化の課題である。

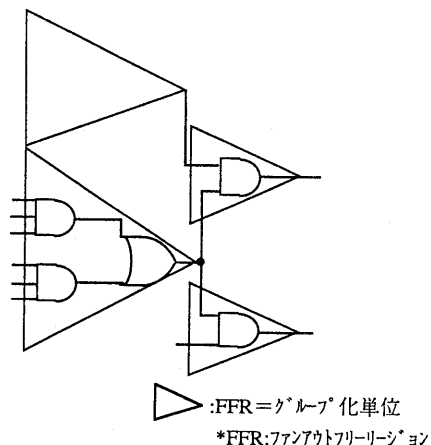


図4.1 組合せ回路単体のグループ化

グループ化単位として、図4.1に示すように、ファンアウトフリーゾーン（以後、FFRと略す）を選択した。ここで、FFRとは、リージョン内のいずれか一つのゲートのみをファンアウト先とするゲート群の集合である。FFRの選択理由は、実用回路のFFR内のゲート数は約3[7]であり、かつFFR内は全てファンアウトが単一であるという性質による。この性質により、評価のオーバーヘッドは高々ブール演算子2回分（3-グループ化前にも評価が必要なイベント入力ゲートの評価分1）の評価であり、又、図4.2に示すように、グループ内のファンアウトが複数ケースと比べ入力イベントに無関係なゲートが少なく評価が無駄になる可能性が少なくなるからである。

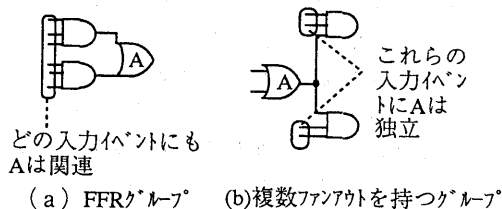


図4.2 グループ構成の比較

F F R内のゲート数3、ゲートの入力本数3の標準モデルについて、イベント伝搬時間をブール演算子評価時間の5倍（ある標準HDLシミュレータの実測結果）、入力イベントが発生した時の中間イベント発生率を50%と仮定した時、高速化効果は1.7倍という机上評価結果を得て、この方式を採用した。

#### (2) 組合せ論理と順序論理のグループ化

このグループ化は、組合せ論理単体のグループ化と同じ考え方を順序論理まで拡張し、順序論理を最終段とするF F Rをグループ化単位とする。すなわち、図4.2に示すように、組合せ論理単体のF F Rのファンアウト先が順序論理のデータ入力のみである時に、F F Rの評価を順序論理のif文の中で行うようにグループ化する。

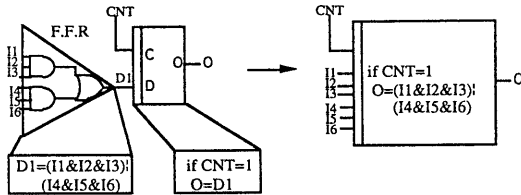


図4.3 組合せ論理と順序論理のグループ化

これにより、順序論理の入力D1に発生するイベント削除に加え、入力I1～I6のいずれかにイベント発生時行われていたブール演算子評価が、CNTが1になった時のみとなるため、ブール演算子評価回数削減も図れる。反面、CNTが0の時のif文の空評価は、グループ化前にはD1イベント発生時のみだったが、入力I1～I6のイベント発生時となるとというオーバーヘッドがある。

(1)で述べた標準モデルについて、if文評価時間をブール演算子評価時間の4倍（ある標準HDLシミュレータの実測値）と仮定すると、入力I1～I6がCNT=1となる前に3回以上変化すると高速化効果が期待できる。現状、重複のない異相のクロックでコントロールCNTと入力I1～I6が変化することを想定しており、CNT=1となる前に入力I1～I6に複数個のイベントが起る可能性は高いと考え、この方式も採用した。但し、入力I1～I6が3回以上変化して同等程度なので大きな効果は望めない。

#### 4.3 コントロール共有順序論理のグループ化

このグループ化は、図4.4に示すように、コントロール信号CNTを共有する順序論理をグループ化する。この時、コントロール信号CNTをグループの共通入力とし、if文も共通化する。

これにより、CNTイベント伝搬数が1/グループ化順序論理数となり、CNTが変化するたびに順序論理毎に必要なif文の評価回数も1/グループ化順序論理数となるため、大きな高速化効果が期待できる。

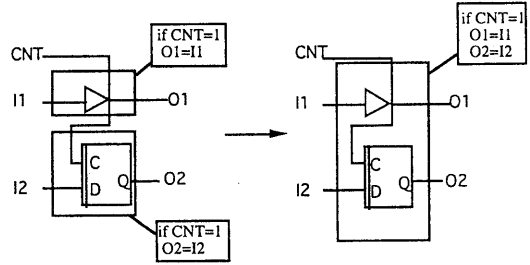


図4.4 コントロール共有順序論理のグループ化

#### 4.4 レベル順序論理の起動タイミング最適化

これは、レベルタイプ順序論理の入力データの変化に対する無駄な起動を削減する最適化であり、複数相の重複のないクロックに同期する順序論理を前提にしている。

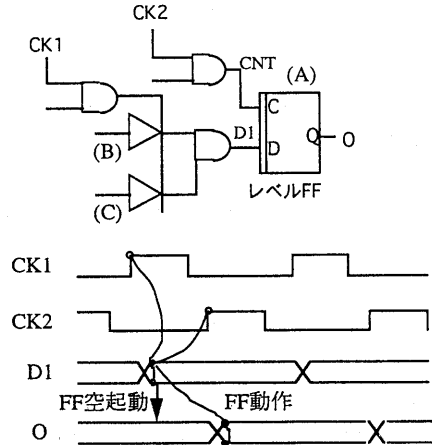


図4.5 レベル順序論理の動作

図4.5に示すように、レベルFF(A)はCK2

に同期しており、入力データD1は、トライステート素子(B)(C)がクロックCK1に同期しているため、CK1に同期している。しかし、(A)はレベルタイプであるため、タイムチャートに示すように、D1変化時にも空起動される。このように、データ取込みクロックと入力データ作成クロックが異なる時には、無駄な起動が起り、シミュレーション時にも、D1イベント伝搬と(A)をモデル化したときのif文の空評価が無駄になる。

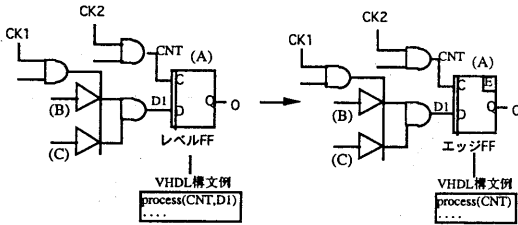


図4.6 レベル順序論理の起動タイミング最適化

上記の条件が満たされている時には、図4.6に示すように、(A)をエッジタイプのFFに変換する。具体的には、VHDL構文例に示すように、センシティブティリストから入力データ信号を除去する。又、順序論理のクロックの判定は、クロックCK2=0を仮に伝搬させコントロールCNTが常に0になる順序論理をCK2同期とすることにより行う。

この最適化により、D1イベント伝搬時間が不要となり、評価時間もD1イベント時のif評価時間が不要となるため、大きな高速化効果が期待できる。但し、このような典型的な同期回路がどれだけ回路中に含まれるかで効果は決まり、例えば、(A)の前段順序論理にリセットピンがあれば、リセットによるデータ作成タイミングはクロック同期ではないのでこの最適化は実施できない。

## 5. 評価

### 5.1 評価方法

評価は、ゲート/Tr.比性能を評価尺度として、シミュレータ性能比に依存しない技術的な評価と実用上の評価の2つの観点で行った。

技術的な評価は、RTLモデル生成方式のみの高速化効果を評価するために行った。この評価を行うため

には、RTLシミュレータとゲート/Tr.専用シミュレータのシミュレータ性能依存を排除しなければならない。そこで、同一シミュレータ上のゲートTr.との性能比較を実施した。具体的には、VHDLで、元のゲート/Tr.記述と1対1で、素子個々はRTLモデルと同じブール式とif文によりモデル化した0ディレイゲート/Tr.モデルを生成し、このモデルに対する、RTLモデルの性能比をVHDLシミュレータ上で実測した。

実用上の評価は、ASIC顧客への提供する場合のRTLモデルのゲート/Tr.記述に対する有効性を評価するために行った。仮に、ゲート/Tr.記述を提供するとすれば遅延付ゲート/Tr.記述であり、顧客はゲート/Tr.専用シミュレータ上でシミュレーションする。そこで、ゲート/Tr.専用シミュレータ上のゲート/Tr.との性能比較を実施した。具体的には、ゲート/Tr.専用シミュレータでの遅延付ゲート/Tr.の性能に対する、このシミュレータに接続されている標準HDLシミュレータ上でのRTLモデルの性能比を実測した。

上記の2つの性能評価用のベンチマーク回路を表5.1に示す。これは、全て自社のASIC用モジュールである。

表5.1 ベンチマーク回路

回路名 (仮名称)	ゲート規模(KG) (4Tr.=1ゲート換算)
モジュールA	9.9
モジュールB	3.1
モジュールC	0.6
モジュールD	0.3
モジュールE	1.3
モジュールF	1.0
モジュールG	1.0

### 5.2 同一シミュレータ上のゲートTr.との性能比較

第5.1節で述べた方法により得たVHDLゲート/Tr.比性能を、表5.2に示す。

モジュールA以外は、ほぼ全て10倍以上、平均14.4倍の高速化効果を得ている。モジュールAの性

能が低いのは、後で述べるように、モジュールA特有に組合せ論理単体のグループ化効果が低いためである。

表5.2 VHDLゲート/Tr.比性能

回路名 (仮名称)	VHDL RTLモデルの 対VHDLゲートTr.比性能
モジュールA	4.3
モジュールB	19.2
モジュールC	28.8
モジュールD	13.7
モジュールE	11.6
モジュールF	9.9
モジュールG	13.5
平均	14.4

次に、最適化の高速化効果内訳を述べる。表5.3は、3回路についての最適化の高速化効果内訳である。RTL論理抽出は、いずれも2倍程度の高速化効果がある。RTLモデル最適化の内訳を見ると、回路依存のある最適化と予想に反して効果のない最適化がある。これらについて、以下に述べる。

表5.3 最適化の高速化効果内訳

回路名	ベース (VHDL ゲートTr.)	RTL 論理抽出	RTLゲート最適化			
			バッド伝搬経路上 論理のグループ化	組合せ論理 と順序論理	制御共有 順序論理の グループ化	レベル順序論 理の起動タイ ミング最適化
モジュール A	1	×2.0 2.0	×2.1 4.1	×1.0 4.2	×1.0 4.2	×1.0 4.3
モジュール B	1	×2.0 2.0	×6.2 12.3	×1.0 12.5	×1.4 18.1	×1.1 19.2
モジュール E	1	×2.0 2.0	×4.4 8.6	×1.0 8.9	×1.3 11.6	×1.0 11.6

凡例 個別効果  
全体効果

回路依存が顕著なのは、組合せ論理単体のグループ化とコントロール共有順序論理のグループ化である。いずれもモジュールAが他の回路より効果が小さい。これは、モジュールAが、当初想定していた通常のクロック同期より複雑なコントロールタイミングを持っているためであり、一般的にあてはまる傾向ではない。

予想に反して効果のない最適化は、レベル順序論理

の起動タイミング最適化である。原因はこれらの回路の中の順序論理が最適化される率が予想に反して低い(モジュールA 2%, モジュールB 14%, モジュールC 6%)ためであり、最適化率の低い原因は、現在調査中である。

### 5.3 ゲート/Tr.専用シミュレータ上のゲート/Tr.との性能比較

第5.1節で述べた方法により得たゲート/Tr.専用シミュレータ上のゲート/Tr.との性能比を、表5.4に示す。シミュレータ、回路によって結果は大きく異なるが、総じて2~20倍、平均約5倍以上の性能を得ており、有効性を確認できた。

表5.4 ゲート/Tr.専用シミュレータ上のゲート/Tr.との性能比

回路名 (仮名称)	VHDL接続シミュレータ	Verilog接続シミュレータ
モジュールA	5.2	4.5
モジュールB	6.5	4.6
モジュールC	14.3	9.8
モジュールD	29.4	1.9
モジュールE	20.1	4.8
モジュールF	2.4	2.9
モジュールG	3.1	4.6
平均	11.6	4.7

## 6. まとめ

顧客サイトでのASIC論理シミュレーションの高速化のため、ゲート/Tr.レベルのモジュールから、標準HDL記述のRTLモデルを生成するHILEXTの開発を行った。HILEXTは、ゲート/Tr.回路をRTL論理に変換するRTL論理抽出に加え、シミュレーション高速化のため、信号共有論理のグループ化と順序論理のタイミング最適化によるRTLモデル最適化を行う点に特徴がある。HILEXT生成モデルの評価として、同一シミュレータ上のゲートTr.との性能比較(技術的評価)とゲート/Tr.専用シミュレータ上のゲート/Tr.との性能比較を実施し、ASIC7モジュールに対して、前者は平均14倍、後者は平均5倍以上の高速化効果を確認した。

## 謝辞

本研究に関し設計者の立場からご討論頂き、又、ベンチマーク回路をご提供頂いた日立製作所半導体事業部の平地和春、内田覚、および前田武志の3氏、並びに半導体事業部ASIC設計部の諸氏に感謝致します。

## 参考文献

- [1] M.Boehner, "LOGEX-An Automatic Logic Extractor from Transistor to Gate Level for CMOS Technology", 25th DAC, pp.518-522, 1988.
- [2] R.E.Bryant, "Extraction of Gate Level Models from Transistor Circuits by Four-Valued Symbolic Analysis", ICCAD, pp.350-353, 1991.
- [3] D.T.Blaauw et al., "Functional Abstraction of Logic Gates For Switch-Level Simulation", EDAC, pp.329-333,1991.
- [4] SpiceネットリストからVerilogネットリストを抽出するEDAソフトウェア:日系エレクトロニクス、No.631, p212 (平7-3-13)。
- [5] 大村他:機能情報抽出を用いたCPUの動作検証、信学論, vol.J76A, No.9, pp.187-194 (平5-9)。
- [6] D.T.Blaauw et al., "Automatic Generation of Behavioral Models from Switch-Level Description", 26th DAC, pp.179-184,1989.
- [7] Z.Barzai et al., "HSS-A High-Speed Simulator", IEEE Trans.CAD, vol.CAD-6, No.4, pp.601-617, July 1987.