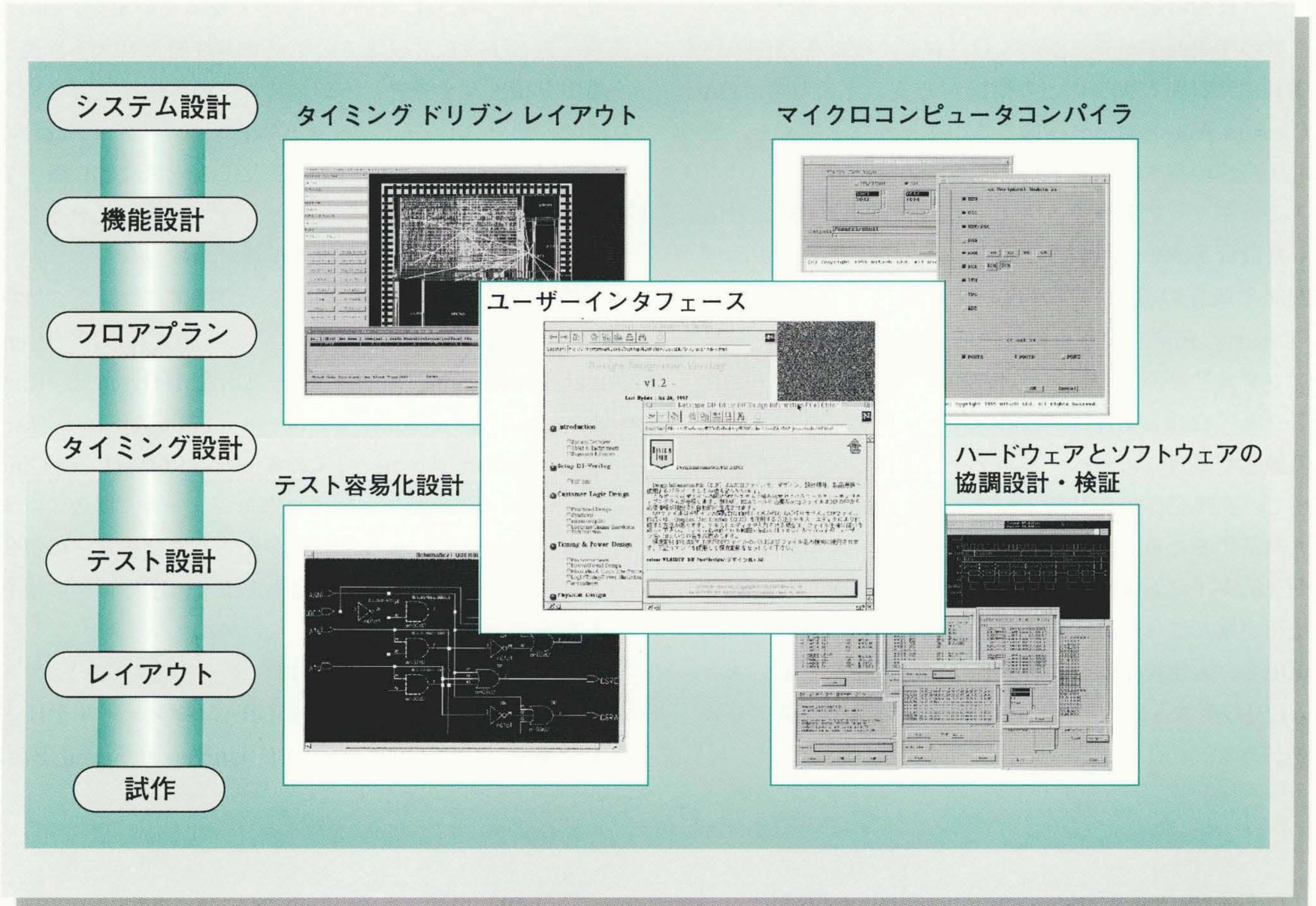


システムLSIのデザインフローと設計環境

Design Flow and Design Environment for System LSIs

高橋 強 *Tsuyoshi Takahashi* 野本和之 *Kazuyuki Nomoto*
鈴木 敬 *Kei Suzuki* 塩月八宏 *Yahiro Shiotsuki*



注：*Netscapeは、米国、日本およびその他の国における米国Netscape Communications Corp.の商標である。

システムLSIの設計環境

システムLSIのデザインフローの概要と各種設計ツールの表示画面を示す。ハードウェアとソフトウェア協調設計・検証、テスト容易化設計のインテグレーションなどが今後重要となる。

プロセス技術の進展により、チップに搭載できるゲート規模が飛躍的に増大し、時代は正にシステムLSIの時代に入った。大規模なシステムLSIを短時間で設計するには、設計生産性の抜本的改革が求められる。既存の設計資産の再利用であるIP(Intellectual Property)化や、ハードウェアとソフトウェアの分割・協調設計、検証技術による設計期間の短縮が重要である。一方、チップ設計でも、より大規模で高速動作のシステムLSIを効率的に設計する必要がある。高速に論理・タイミング検証す

るために、サイクルベースシミュレーションや静的タイミング検証、論理等価検証(形式検証)技術が進んでいる。また、低電力化のための設計・検証環境と、多種多様な部品を搭載するシステムLSIのテスト設計をトータルに改革することも大切である。

これらの新しい設計技術を従来の設計環境にデザインキットとして盛り込み、新しい統合設計環境を構築し、提供していく。

1 はじめに

プロセス技術の進展により、ハーフミクロン、クォータミクロンからサブクォータミクロンの時代に突入し、チップに搭載できるゲート規模も飛躍的に増大してきたにもかかわらず、そのチップに搭載する論理機能の設計力は延びてきていない。SEMATECH, Inc.によれば、プロセス技術の進展を示すチップ当たりの論理トランジスタ数の年成長率が58%であるのに対し、設計生産性の年成長率は21%と少ないとの指摘がある。そのため、DA (Design Automation) 技術でこのギャップを埋めて、設計生産性の抜本的改革を行う必要がある。

このギャップを埋めるとともに、システムLSIに対応した設計手法の改革が重要となっている。搭載する論理機能を既存の設計財産の再利用で補うIP (Intellectual Property) 化や、より自由度の増えるハードウェアとソフトウェア分割での最適設計技術、ハードウェア・ソフトウェア設計の先行協調設計・検証技術による設計期間の短縮が重要である。一方、チップ設計でも、より大規模で、高速動作のシステムLSIを高速に論理・タイミング検証するために、サイクルベースシミュレーションや静的タイミング検証、および論理等価検証(形式検証)技術が進んでいる。また、低電力化のための設計・検証環境と、大規模でかつ多種多様な部品を搭載するシステムLSIのテスト設計のためのトータル環境を改革する必要がある。

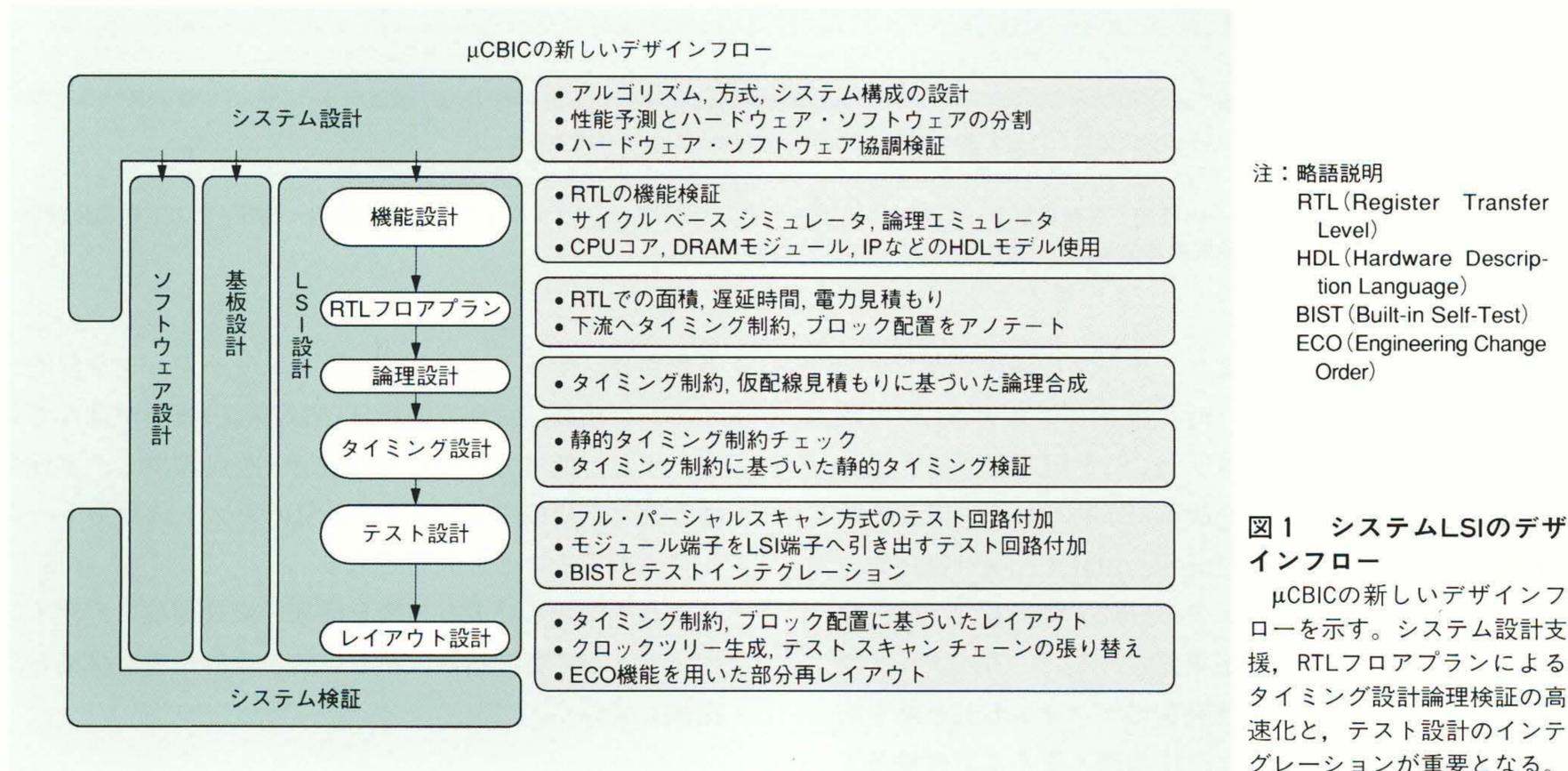
ここでは、日立製作所のシステムLSI製品を代表するμCBIC (Micro Cell Based IC) をベースとした新デザインフローや設計環境、特にシステムLSIの設計・検証のために新しく取り入れる主要技術について述べる。

2 システムLSIのデザインフロー

従来のデザインフローはチップ設計に対するものであり、システムLSIに対応したものとなっていない。そのため、システム設計段階でのシステム性能、コスト、消費電力見積もりを考慮し、システム機能・性能検証を充実させ、チップ(ハードウェア)設計、ソフトウェア設計の再設計をなくす必要がある。また、チップ設計に限っても、より大規模で高速な動作と、低消費電力化をさらに効率的に設計するための技術革新が求められる。

設計生産性のいっそうの向上を目指すには、既設計財産の再利用IPも重要である。IPの品ぞろえは、CPU (Central Processing Unit) コアや、DRAM (Dynamic Random Access Memory) モジュール、SRAM (Static Random Access Memory) モジュール、ROM (Read-Only Memory) モジュール、アナログモジュール、周辺インタフェースモジュールなど多種多様であり、またその提供元は半導体ベンダである場合もあるし、IPベンダの場合もある。

日立製作所は、SuperHやH8シリーズのマイコン(マイクロコンピュータ)を、さらにDRAMモジュールをIPとして準備している。また、USB (Universal Serial Bus)



やIEEE1394などの周辺インタフェースモジュールの整備も推進している。

μCBICの新しいデザインフローを図1に示す。

2.1 システム設計

目標システム仕様を達成するためのアルゴリズム、方式、構成を設計する。LSIのハードウェアの基本設計だけでなく、デバイスドライバなどのソフトウェアの基本設計も行う。性能、コスト、消費電力を考慮して、ハードウェアとソフトウェアのどちらで実現するか意思決定の後、「ハードウェア・ソフトウェア協調シミュレーション」により、ハードウェアとソフトウェアを同一の設計環境でシステム検証する。

2.2 機能設計

ハードウェアで実現する機能をRTLで記述し、シミュレーションで検証する。既存機能であるCPUやDRAM、SRAM、ROMなどのモジュールは、日立製作所が提供するモデルを用いて検証する。システムLSIが大規模なので高速性を重視し、この段階ではタイミングは考慮しないで、論理機能の検証だけを行う。検証ツールは、サイクルベースシミュレータや論理エミュレータを使用する。

RTLの機能検証後にRTLフロアプランを行い、ブロック面積、遅延時間、消費電力について見積もる。また、この段階で決めたブロック配置やブロックモジュール間の遅延時間の見積もりをタイミング制約として、下流の設計である論理合成やレイアウト設計に渡す。

2.3 タイミング設計

タイミング検証では、論理合成結果の論理ブロック自体と、論理ブロックとCPU、DRAM、SRAM、ROMなどのモジュール間で、タイミング制約を満たしているかどうかの検証を行う。タイミング検証では、静的タイミング検証ツールを用いて全パスのタイミング検証を高速に行う。ディレイタイミング検証のためのテストパターン設計は不要である。

2.4 テスト設計

大規模で多種多様なモジュールを搭載するシステムLSIのテスト設計では、単一テスト技術だけでなく、各種のテスト設計技術を統合させる必要がある。フルスキャン・パーシャルスキャンで論理ブロックにテスト回路を付加したり、モジュールを単体でテストできるように、モジュール端子の信号をLSI端子に引き出すためのテスト回路を付加する。面積・遅延オーバーヘッドとテスト設計工数・テストパターン長のトレードオフを考慮し、各

種のテスト設計技術を組み合わせて使用する。

3 システム設計・検証

3.1 システムLSIの設計と検証

システムLSIの登場により、これまでシステムを構成する1部品であったLSIは、システムの大半を集積する中核になる。システム設計をLSIの開発という観点で見ると、従来はシステムの仕様を明確化してからLSIの仕様を決め、開発を行っていた。一方、システムLSIの場合は、システムの仕様を決める作業とLSIの仕様を決定する作業の多くを並行して行うことになる(図2参照)。

システム設計に対する設計支援ツールのニーズは、(1)システム仕様設計(アルゴリズムや方式の検討、プロトタイプングによるシステム構成の決定、性能予測とハードウェア・ソフトウェア分割)、(2)ハードウェア検証、(3)ソフトウェア検証とソフトウェアの性能チューニングの三つに大別できる。

3.2 システム設計支援ツール

システムレベルの設計を支援するCADは、広義にはESDA(Electronic System Design Automation)と呼ばれている。ESDAに分類されるツールは、(1)ブロック図入力やFSM(Finite State Machine)入力などの設計エントリツール、(2)デジタル信号処理アルゴリズムを設計し、データパスやDSP(Digital Signal Processor)のプログラムを開発するシステム、(3)待ち行列などのモデル

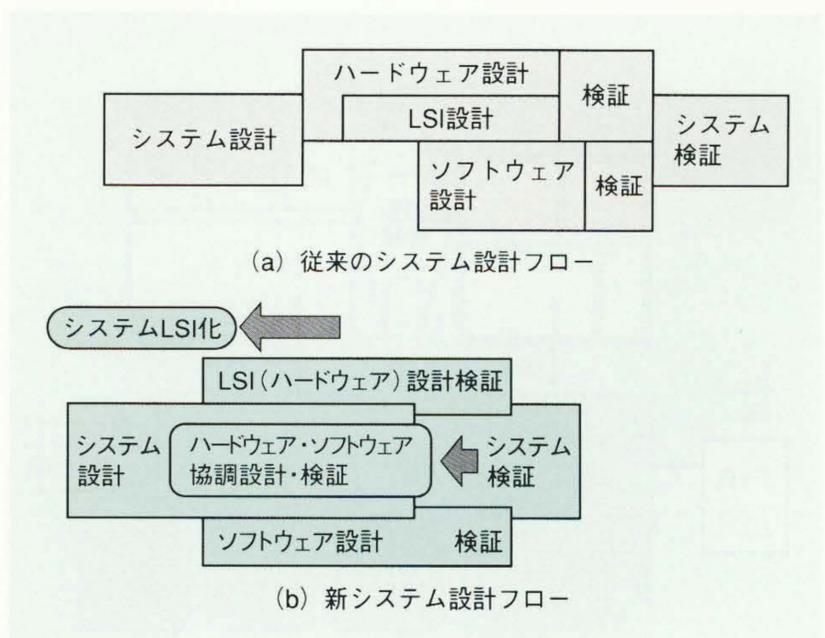


図2 システム設計フローの変化

従来の設計フロー(a)では、ハードウェアとソフトウェアの設計・検証が完了してからシステム検証を行っていたので、システム検証の段階で発見されるバグが大きな設計遅延となる。新しい設計フロー(b)では、ハードウェア・ソフトウェア協調設計・検証により、両方を並行設計するとともに、システム検証も並行して、全体の設計期間を短縮する。

をシミュレーションするシステムシミュレータなど幅広く存在する。

その中で、CPUコアを内蔵するシステムLSIの設計を支援するハードウェア・ソフトウェア協調設計が注目されている。ハードウェア・ソフトウェア協調設計の主な目的は、(1)設計の早い段階でシステムの仕様を明確化して、ハードウェアとソフトウェアのトレードオフを最適化することと、(2)ハードウェアとソフトウェアの開発を同時進行させることによって設計期間を短縮することにある。現在、後者を目的として、ソフトウェアを実行する命令セットシミュレータとハードウェアを実行する論理シミュレータを並行して動作させる「ハードウェア・ソフトウェア協調シミュレータ」が普及しはじめている。

協調シミュレータ上の命令セットシミュレータも、CPUの汎用レジスタの表示やブレイクポイントの設定など、通常の組込み用CPUの開発環境として提供されている命令シミュレータと同様の機能を持っている。論理シミュレータ上での波形観測と合わせて、ソフトウェアとハードウェアの同時デバッグが可能になる(図3参照)。

ハードウェア・ソフトウェアの協調設計・検証は、システムの設計手法に関する新しいアプローチであり、一般的には設計対象のアプリケーションと設計手法により、ツールに対するニーズが異なってくる。

協調シミュレータを用いたシステム検証はCPUコアを内蔵するシステムLSIの設計に有効なツールである

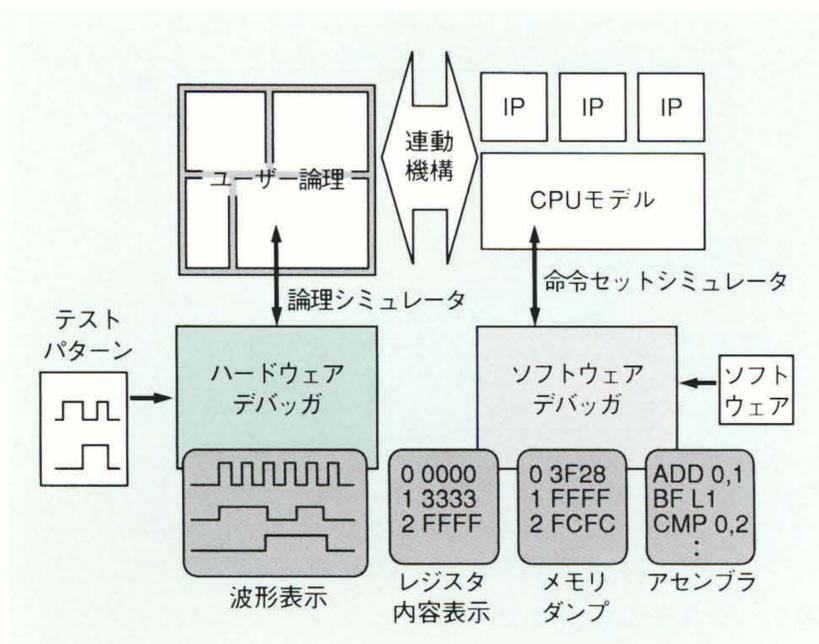


図3 ハードウェア・ソフトウェア協調検証ツール

ユーザー論理は論理シミュレータで、ソフトウェアは命令セットシミュレータで、連動機構の下でシステム全体をそれぞれシミュレーションする。ハードウェアとソフトウェアおのおのに適したデバッグを用意している。

が、これだけで上記のニーズをすべて満たせるわけではない。ハードウェア検証には、特にバスサイクルに関して、タイミングが高精度なプロセッサモデルが必要である。ソフトウェアのデバッグには、高速性やGUI(Graphical User Interface)、ICE(In-Circuit Emulator)などの実機デバッグ環境とのインタフェースの使い勝手が重要である。さらに、ソフトウェアのチューニングには、バスアクセスやキャッシュ動作、またはプログラム実行のプロファイル機能が大切である。

3.3 CPUコアモデルと開発ツール

EDA(Electronic Design Automation)のベンダでは、ハードウェア・ソフトウェア協調シミュレータだけでなく、ハードウェア・ソフトウェア協調設計ツールも製品化しはじめている。しかし、協調設計ツールが対象とするシステムレベルの設計はアプリケーションごとに設計の観点異なるため、汎用的に使えるツールは今のところ無い。また、協調設計ツールは設計手法に対する影響度が大きいいため、徐々に浸透していくことになるものと予想する。

協調シミュレーションでキーとなるのは、CPUコアモデルである。CPUコアモデルの開発での課題は、シミュレーションの高速性と実際のチップとの等価性である。現在、SuperHマイコンコアのモデルでサポートしており^{*)}、モデルの開発には、各社との技術協力によって等価性の維持に努めている。

4 機能・タイミング検証

システムLSIの時代を迎えて論理規模は膨大なものとなり、その論理規模やテストパターン長に相乗的に比例する機能検証とタイミング検証は、従来のイベントドリブンシミュレーションでは限界にきている。この打開策として、機能検証とタイミング検証を分離し、それぞれに適した技術を適用して総合的に処理時間を低減する手法が進められている(図4参照)。

4.1 機能検証

システムLSIの機能検証は、ユーザー論理とCPU、DRAM、IPなどのモジュール混在で行う。大規模なため

*) 1998年6月時点で、横河電機株式会社の仮想ICE、メンター・グラフィックス・ジャパン株式会社のSeamless CVE、ガイオ・テクノロジー株式会社のXASS-V、CARDtools Systems社のNitroVPなどのCPUコアモデルがある(順不同)。

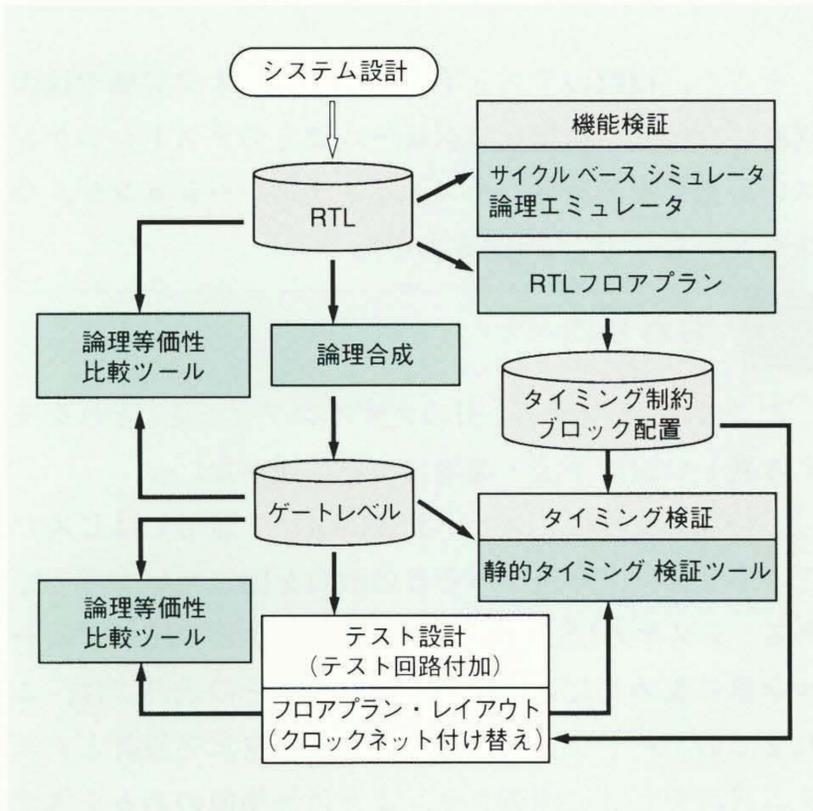


図4 機能・タイミング検証手法

RTLの機能検証はサイクルベースシミュレータや論理エミュレータで、ゲートレベルのタイミング検証は静的タイミング検証ツールでそれぞれ行う。両レベルの論理等価性検証や、下流のテスト・レイアウトでのネットリスト変更前後での論理等価性は、形式検証ツールで検証する。

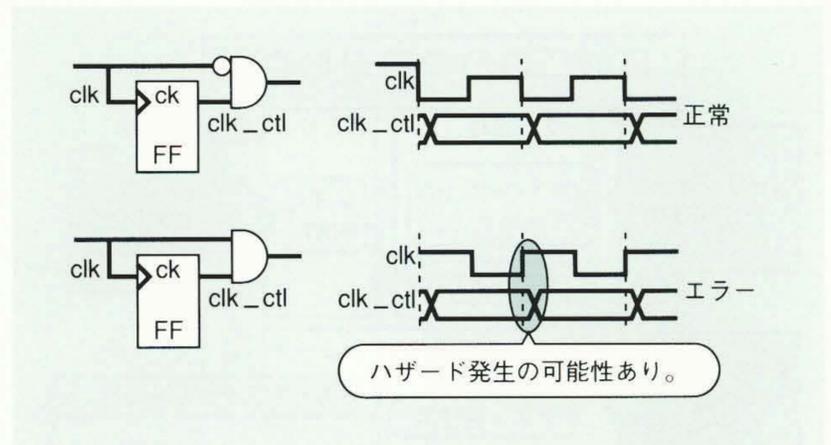
に高速性を重視し、この段階ではゲート遅延時間やセットアップ・ホールド時間のタイミングを無視したサイクルベースシミュレータや、論理エミュレータを使用する。これにより、処理速度は、従来のイベントドリブンシミュレータに比べて、数十から数千倍の高速化が見込まれる。

現状の技術では、RTL段階の機能検証はシミュレーションや論理エミュレータに頼らざるをえないが、論理合成前後の論理等価性の検証やテスト回路付加前後、レイアウトでのテストスキャンチェーン張り替えとクロックツリー生成前後の論理等価性の確認では、論理等価性検証(形式検証)ツールを使用する。論理等価性検証ツールにより、従来のシミュレーションによる結果比較で数日要したものを、数時間の検証で済ませている。

4.2 タイミング検証

タイミング検証では、静的タイミング検証ツールを用いてタイミング検証を高速に行う。

静的タイミング検証ツールでは、単相クロック同期回路であれば、フォールスパスやマルチサイクルパスの問題を除けば、大きな問題はない。しかし、現実のシステムLSIでは完全な同期設計が行われず、部分的に非同期回路が使われたり、消費電力を考慮したゲーテッドクロックやマルチクロックが多用される。このような場合、



注：略語説明 clk(クロック信号), ck(クロック端子) FF(フリップフロップ), ctl(クロック制御信号)

図5 STA制約チェックの例

ゲーテッドクロックのクロック制御信号(clk_ctl)がクロック信号と同相で変化しないことをチェックする。

現状の静的タイミング検証ツールでは、的確な検証が行われず、従来のイベントドリブンシミュレータを使ったタイミング検証が必要となる。

この問題を解決するものが、静的タイミング制約チェック機能である。従来、マイクロプロセッサの設計には静的タイミング検証ツールを適用しており、この経験を通して得た技術やノウハウを静的タイミング制約(STA制約)として整理し、STA制約チェッカとともに、μCBICへの展開を始めている。STA制約は、静的タイミング検証ツールの不足機能を補う位置づけであり、ゲーテッドクロック、マルチクロック、フリップフロップのセット・リセット端子でのポテンシャルハザードのチェックや、クロック分周回路の構成に関する制約のチェックを行う(図5参照)。論理設計の早期段階でSTA制約チェックを行い、エラー部を修正することにより、下流設計がスムーズに行える。

5 テスト容易化設計環境

システムLSIの時代を迎え、テスト容易化設計にもさらに一歩進んだ対応が求められている。特に、(1)システムLSI実現の背景にあるLSIの大規模化への対応、(2)マイコンコアや標準インタフェースなど内蔵するモジュールのテスト、(3)内蔵するモジュールの構成により、テスト手法を組み合わせた最適設計を行うためのテストインテグレーションの3点が重要になっている(図6参照)。

5.1 LSIの大規模化への対応

論理規模の増大は、テストに必要なテストパターン長の増大を招き、テストへの負荷が大きくなっていく。日立製作所は、従来、ASICのテスト容易化設計手法とし

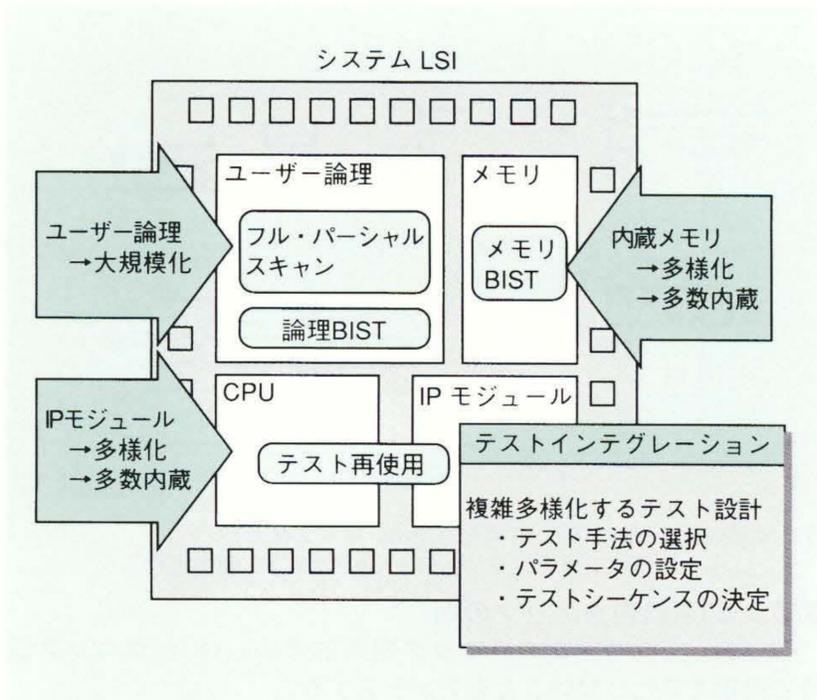


図6 システムLSIテスト設計の課題

ユーザー論理の大規模化やIPモジュール・内蔵メモリの多種多様化に伴い、それらの構成に応じたテスト手法を組み合わせ、面積・遅延オーバーヘッドとテスト設計工数・テストパターン長のトレードオフの最適設計を行うテストインテグレーションが重要になっている。

て、スキャンパス方式自動テスト設計や、 μ CBICのモジュールテスト設計を行ってきた。今後は、パーシャルスキャンによるオーバーヘッドの低減と、LSIにみずからテストを行わせるBIST方式の活用が重要になると考え、開発を進めている。

5.2 内蔵モジュールのテスト

日立製作所は、 μ CBICとして、H8SやSH-3などのCPUコアと、周辺モジュールを搭載したASICをすでに製品化している。テスト時には、内蔵したモジュールの端子を、LSIの外部端子に引き出すテスト回路を自動付加する。また、設計済みのモジュール単体のテストパターンを、LSIの外部端子からのテストパターンに再編集し、内蔵モジュールのテスト設計を完全に自動化している。

また、内蔵メモリモジュール(DRAM, SRAM, ROM)のテストについては、LSI内部でテストを自動化する「メモリBIST」の活用を進めている。

5.3 テストインテグレーション

システムLSIの各モジュールについては、その特徴に応じて最適なテスト容易化設計手法を選択する必要がある。一般に、採用できるテスト手法は1種類ではなく、複数の手法やパラメータから最適と思われるものを選択しなければならない。テスト容易化設計では論理ネット上でテスト回路の追加を行うので、あとからテスト方針を変えると、論理設計まで戻る必要が生じて現実的では

ない。

そこで、複数のテスト手法のトレードオフ評価や採否判断のガイド、複雑なモジュールごとのテストシーケンスの最適化などを行うテストインテグレーションが、今後重要になってくる(図6参照)。

6 おわりに

ここでは、システムLSIのデザインフローと、それを支える新しい設計手法・環境について述べた。

この設計環境はシステムLSIに向けて適用しはじめたところであり、今後いっそうの改良を図っていく考えである。システムLSIは、CPUやDRAMなどのIPモジュールを単に集めただけのものではなく、その設計には、これまでのハードウェア設計に、ソフトウェア設計とシステム設計を互いに協調させ、より付加価値のあるシステムとして社会に送り出していくことが重要である。そのために、システムメーカーといっそう緊密な情報交換を行っていく考えである。

参考文献

- 1) 日経エレクトロニクス1997年9月1日号, No.697, pp.67~85

執筆者紹介



高橋 強

1970年日立製作所入社, 半導体事業本部 システムLSI事業部 技術開発センター 設計技術開発部 所属
現在, マイコン・ASIC用設計環境開発に従事
E-mail: takahats@cm.musashi.hitachi.co.jp



鈴木 敬

1989年日立製作所入社, 中央研究所 システムLSI研究室 所属
現在, システムLSIの設計・検証技術の研究に従事
工学博士
電子情報通信学会会員, 情報処理学会会員, 電気学会会員, IEEE会員, ACM会員
E-mail: ksuzuki@crl.hitachi.co.jp



野本和之

1980年日立製作所入社, 半導体事業本部 システムLSI事業部 技術開発センター 設計技術開発部 所属
現在, テスト容易化設計環境開発に従事
E-mail: nomotok@cm.musashi.hitachi.co.jp



塩月八宏

1982年日立製作所入社, 半導体事業本部 システムLSI事業部 技術開発センター 設計技術開発部 所属
現在, ASIC用設計環境開発・適用サポートに従事
E-mail: shiotsuk@cm.musashi.hitachi.co.jp