特集

システムLSI新設計手法の導入

Introduction of New Design Methodologies for System LSI

あらまし

大規模かつ高性能なシステムLSIを短期間で設計するために導入した新設計手法を紹介する。HW/SW Co-design，すなわち，Hardware/Software Co-designはハードウェアとソフトウェアを同時に開発する設計手法であり，ソフトウェアを含めたシステム検証をハードウェアの試作前に行うことが可能になった。新論理検証手法では，スタティックタイミング解析による高速タイミング検証と，フォーマル・ベリフィケーションによる高速論理機能検証の結合により，論理検証期間の大幅な短縮を実現した。論理フロアプラン設計手法は，フロアプランに基づいた再最適化処理を行うことにより，実際の配置・配線後に近い予測配線容量での論理設計を可能にし，チップレイアウト後のタイミングエラー発生を最小限に抑えるので，最終的なタイミング調整のための作業を軽減した。また，RTL(Register Transfer Level)での消費電力解析と低消費電力合成手法は，短期間かつ最適な低消費電力設計を可能にした。

Abstract

This article describes the following new design methodologies we have introduced for designing large-scale, high-performance System LSI：HW/SW (Hardware/Software) Co-design：This enables simultaneous development of hardware and software. Using this methodology, system verification (including software verification) can be started long before the actual hardware is produced.

A Logic Verification methodology consisting of static timing analysis for high-speed timing verification and formal verification for high-speed logic function verification. This methodology can drastically reduce the logic verification time.

Logic Floorplan：This combines the logic floorplan and re-optimization, which enables logic design software to use an estimated wire capacitance that is closer to that of the actual layout. This helps avoid post-layout timing errors, and eliminates the need for ECO, which is conventionally required.

An RT Level Power Analysis combined with a Low-power Synthesis methodology that quickly provides an optimized low-power design.

石渡啓介（いしわた　けいすけ）

1982年千葉大学理学部数学科卒。同年富士通入社。以来論理LSIの開発，設計手法の開発に従事。1986年-1990年米国 Fujitsu Microelectronics，Inc. 勤務。
システムLSI事業部開発部

ま え が き

LSIの大規模化が急速に進み，搭載可能なゲート数が100万を超えた現在，チップ上に一つのシステムをすべて搭載することが夢ではなくなった。しかし，このようなシステムLSIを従来の設計手法で開発することはほとんど不可能である。それは主に二つの要因による。一つは回路規模そのものの増加であり，もう一つは高機能化による回路構成の複雑化である。このため今までと同じ設計のやり方を続けていれば，それはLSI開発期間の極端な増加，製品の品質低下を招くことは明白である。

システムLSI設計に立ちはだかるこの大きな問題を解決するための最も効果的な手段として，著者らは大規模LSI開発者に対し従来の設計手法からの脱皮を強く促す。すなわち，LSIの大規模化と複雑化の渦の中に巻き込まれないためには，それに対抗し得るような新設計手法の導入が必要不可欠である。そのような新しくかつ頑強な設計手法を早期に立ち上げ，提供し，サポートを行うことが著者らに課せられた使命である。

本稿では，すでに開発，提供中のこれら新設計手法とその重要性について紹介する。

HW/SW Co-design

HW/SW Co-designとは，ハードウェアとソフトウェアの協調設計を意味し，システムLSI設計において特に重要な役割を果たす設計手法である。従来別々に行われていたこれらの設計を同時進行的に遂行することにより，システム全体の設計をミスが少なくしかも短期間に行うことが可能になった。ここでは，そのうちで重要な役割を果たすハードウェア・ソフトウェア協調シミュレーションとエミュレーションについて説明する。

従来は，ソフトウェアを含めたシステムの機能検証をするためにはチップ等のハードウェアの試作完了を待たなければならなかった。しかし，ここに大きな問題が存在した。すなわち，実機によるシステム検証でソフトウェアやハードウェアのミスが発見された場合の，修正による開発期間およびコストの増加である。

● HW/SW コ・シミュレーション

HW/SW コ・シミュレーション，すなわち，ハードウェアとソフトウェアの協調シミュレーションは，このような問題を解決する。コ・シミュレーション環境の構成例を**図-1**に示す。ハードウェアの試作開始前にハードウェアとソフトウェアとの協調シミュレーションを行い，それぞれのデバッグを実行する。これによりソフトウェアのデバッグやハードウェアの仕様修正等をシステム開発の初期段階で行うことができる。つまり**図-2**に示されるように，システム全体の開発期間および開発コストを大幅に削減することが可能である。

富士通では，HW/SW コ・シミュレーションツールとして横河電機殿のVirtual ICE，およびガイオテクノロジー殿のAsim-Gをすでにサポート中である。また米Synopsys社

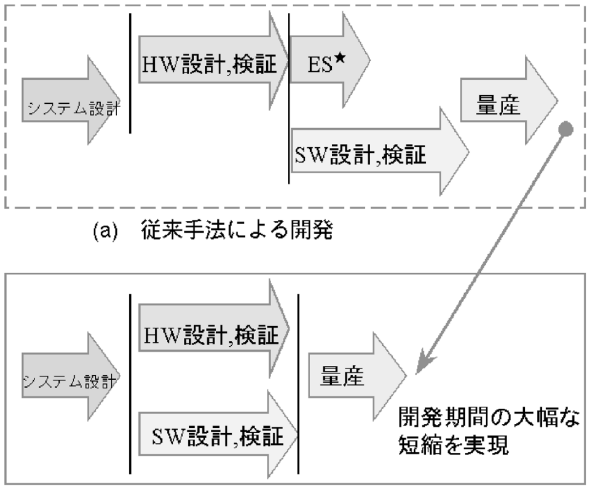

図-2 HW/SWコ・シミュレーションによる開発期間の短縮
Fig.2-Design time reduction with HW/SW co-simulation.

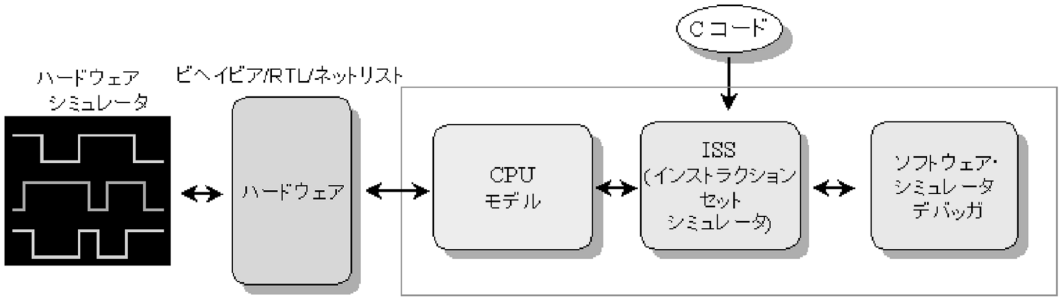

図-1 HW/SWコ・シミュレーション環境例
Fig.1-Example of HW/SW co-simulation configuration.

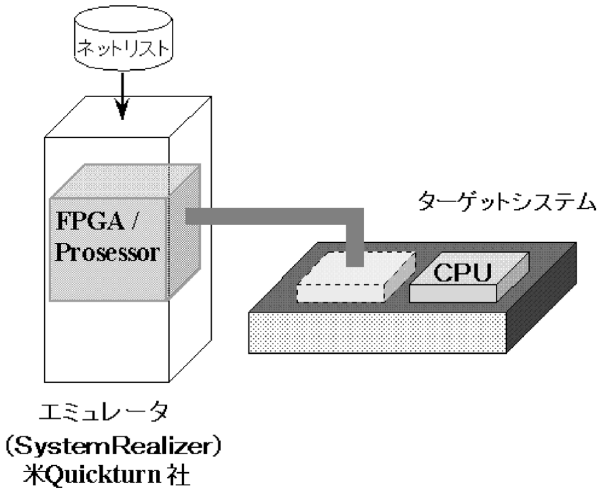

図-3　エミュレーション構成例
Fig.3-Example of emulation configuration.

のEAGLEiをサポート準備中である。

● **エミュレーション**

多くのFPGA(Field Programmable Gate Array)や専用プロセッサチップにチップ等の回路論理をマッピングし，それを開発対象システムに接続することにより，インサーキット・エミュレーションを行う。これにより，ハードウェア試作前にシステム検証を行うことが可能となった。エミュレーションの構成例を**図-3**に示す。エミュレータはFPGAまたは専用プロセッサチップで構成され，その出力をケーブルを通して開発対象システムのチップが搭載されるソケットに接続する。

エミュレータを使うメリットは二つある。一つはハードウェア試作前にソフトウェアとともにシステム検証をすることができることで，二つ目は超高速論理シミュレーションとしての役割である。通常のゲートレベルのシミュレータの4桁から6桁ほど高速なので論理検証期間を大幅に削減できる。またチップに与えられる入力信号は開発対象システムが発生するので，システム検証に必要な膨大なテストパターンの作成が不要なことも大きなメリットである。

エミュレータとして，米Quickturn社のSystemRealizerを現在サポートしている。

論　理　検　証

ここでは論理検証を広い意味で定義し，論理機能検証とタイミング検証を含むものとする。これらの手法の従来の問題点とそれらを解決するための新手法を以下に説明する。

従来は，回路の機能を検証する手法として論理シミュレーションが用いられてきた。これは，回路が正しく機能するかどうかの論理機能検証とタイミング的に正しく動作するかどうかのタイミング検証を同時に実行するものであった。ところが，システムLSIのような大規模回路ではシミュレーションの実行時間やテストパターン作成時間が指数関数的に増加するという問題があった。また，テストパターンの品質によってテストカバレッジ(注1)が大きく左右されるという品質上の問題もあった。

これらの問題を解決するために以下のような新論理検証手法を導入した。

● **スタティックタイミング解析**

従来，論理シミュレーションで行っていたタイミング検証部分をスタティックタイミング解析ツールで行う。これは，テストパターンを用いず静的にタイミング検証を行う手法である。大きな特長として，検証時間が論理シミュレーションに比べて圧倒的に短いことが挙げられる。実例として，等価ゲート数50万ゲートの回路データに対して論理シミュレーションで12日間かかっていたものがスタティックタイミング解析では4時間で検証できた。

また，スタティックタイミング検証はテストパターンを必要としないため，テストパターン作成に要していた時間を節約することができる。

このように設計期間の短縮が可能になることに加えて，スタティックタイミング解析によってタイミング検証の品質向上が可能になった。まず，網羅的な検証が可能となったことが挙げられる。従来の論理シミュレーションではテストパターンの品質がそのままカバレッジに影響を与えたが，スタティックタイミング解析はタイミング問題のほぼすべてを検出可能である。また，チップ内に存在する遅延値のばらつきも的確に反映し高品質な検証が可能となった。

現在，スタティックタイミング解析ツールとして米Synopsys社のMotiveおよび内製ツールのOptingをサポート中である。

● **フォーマル・ベリフィケーション**

フォーマル・ベリフィケーションは，二つの設計データの論理機能が等価であることを検証する手法である。

具体的には，元となるRTL(注2)やネットリスト(注3)等の設計データを論理シミュレーションではじめに論理機能検証しておく。その後の各設計段階でタイミング調整等のなんらかの理由で論理の変更なしに回路の変更をした

（注1）故障検出率。
（注2） Register Transfer Levelの略語で，レジスタ間の動作で表す設計レベルのこと。
（注3）ゲートの結線情報データ。

際，元のデータと論理機能的に等価であることをこのフォーマル・ベリフィケーションで検証可能である。

この手法の効果は二つある。まず，従来の論理シミュレーションに比べて処理にかかる時間がはるかに短い。実例として論理シミュレーションで12日間かかっていた上述の等価ゲート数50万ゲートの回路データに対して12時間で検証できた。つぎに論理機能検証の網羅性である。従来の論理シミュレーションではタイミング検証と同様，テストパターンが検証カバレッジを左右するが，本手法は理論上100%の検証が可能である。

フォーマル・ベリフィケーションツールとして，米Chrysalis社のDesignVERIFYerを現在サポート中である。

論理フロアプランと再最適化

ここでは，設計初期段階からフロアプラン(注4)等を併用しながらタイミング設計を行い，レイアウト後の最終検証でタイミング問題が発生することを防ぐ設計フローを紹介する。

従来のレイアウト前の論理設計では，配線容量(注5)として経験的データから求められた仮配線容量テーブル(注6)を用いていた。ところがこの手法には，レイアウト後に行う検証でタイミングエラーが多出してしまうという欠点があった。この問題を解決するために以下の手法が導入された。

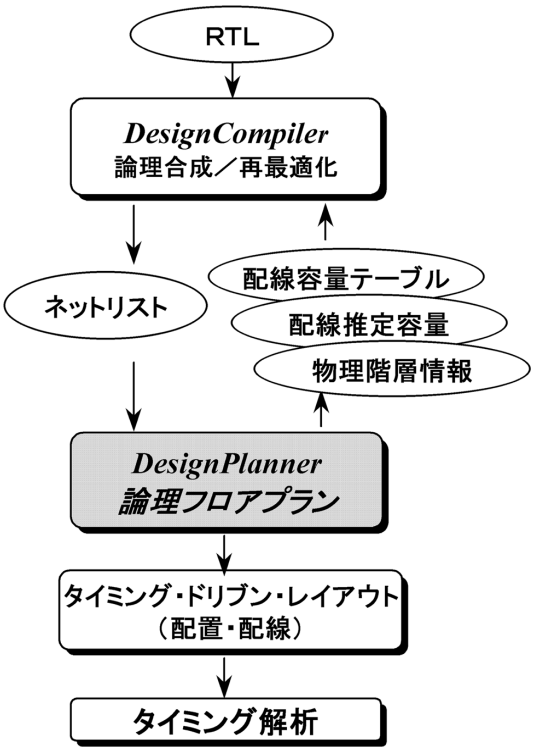

図-4　論理フロアプランと再最適化フロー
Fig.4-Design flow with logical floorplanning and re-optimization.

（注4）マクロやセルのチップ上の概略配置を決める処理。
（注5）金属配線のキャパシタンス。
（注6）経験値から統計的に求められた配線容量テーブルで，セル数と出力分岐数に応じた推定配線容量値が記述されている。

レイアウトの前処理としてのいわゆる物理フロアプランに対して，それよりも早期の論理設計段階でフロアプランを使用し，より正確なタイミング設計を行うことを目的としたものを，ここでは論理フロアプランと呼ぶ。

論理フロアプランは，大規模LSIでは今後非常に重要な設計手法になると著者らは考える。それはレイアウト後のタイミングエラーを最小限に抑え，チップ設計期間の短縮を実現するからである。

論理フロアプランを用いた設計フローを**図-4**に示す。以下，設計手順を簡単に説明する。前述の仮配線容量テーブルを用いた論理合成によって生成された原ネットリストを論理フロアプランツールに入力し，チップ上の相対的なマクロやコアの位置を決める。さらに暫定的なセルの自動配置も行う。論理フロアプランツールは，このフロアプラン結果から物理階層毎の配線容量テーブルとセル配置結果に基づいた配線容量を自動的に生成する。これらのファイルを論理合成ツールに入力し，そこで検出されるタイミングエラーを再最適化処理により自動的に修正する。

ここで注目すべきことは，この再最適化処理で使用される配線容量テーブルあるいは配線容量が実際のレイアウトのベースとなるフロアプラン結果をもとにしているため，従来の仮配線テーブルに比べてはるかに高精度，つまり実際のレイアウト後の値に近いことである。

論理フロアプランツールとして米Cadence社のDesignPlannerを，そしてこれにリンクして実行する再最適化のための論理合成ツールとして，米Synopsys社のDesignCompilerをサポート中である。

低消費電力設計

携帯機器等の民生機器では消費電力を抑えることが至上命題である。ところがシステムLSIでは回路の大規模化・高速化に伴い，消費電力は更に増加する傾向にあり，低消費電力設計手法は特に注目されてきている。

従来は低消費電力設計を実現するような手法は特に存在していなかった。このため，最終検証の段階で規定消費電力を超えていることが判明し，設計のやり直しまたはシステムのスペックダウン等を招くことが多かった。

これらの問題を回避するために以下のような手法を導入した。

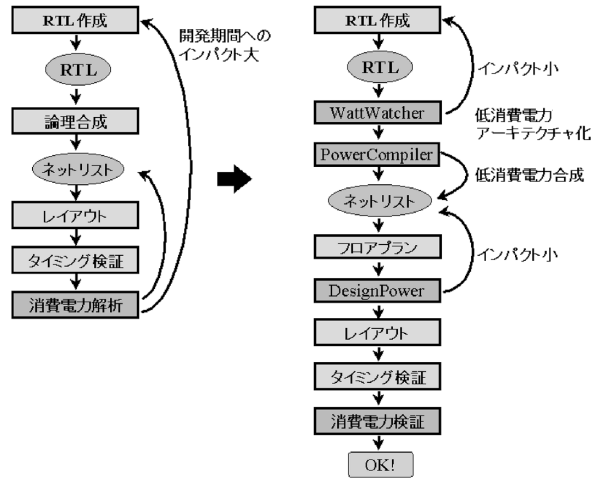

図-5 低消費電力設計フローの比較
Fig.5-Comparison of low-power design flow.

● RTL消費電力推定

RTLによる上流言語設計段階から消費電力を推定し，より消費電力の少ないアーキテクチャを選択した論理設計を可能とする。

低消費電力設計手法の新旧比較を**図-5**に示す。作成されたRTLをRTL消費電力推定ツールに入力し消費電力値を推定する。これによるメリットは最小の時間で低消費電力設計の繰返しが可能なことである。また，設計初期段階からLSIの消費電力を見積もれるため，最終段階で問題が発覚するようなことを未然に防ぐことが可能になった。

RTL消費電力推定ツールとして，米Sente社のWatt-Watcherをサポート中である。

● 低消費電力合成

従来，論理合成ツールに与えることが可能な制約条件はエリア(注7)とタイミングのみであったが，この新手法により，さらに消費電力パラメータを制約条件に加えることが可能になった。つまり，あるデザインがエリアやスピードよりも低消費電力であることを要求される場合には，この手法によって消費電力パラメータに優先度を与えて論理合成し，同じRTLから結果として，より低消費電力のデータをネットリストの形で得ることが可能である。

低消費電力合成推定ツールとして，米Synopsys社のPowerCompilerをサポート中である。

今後の展望

システムLSI時代の本格化に対応して，今後とも更なる新設計手法の導入が不可欠である。富士通では，大きく分けて以下の二つの分野に力を入れて設計手法・設計環境の開発を進めている。

第一に，より上位言語レベルにおける設計環境の充実である。本稿ですでに紹介したHW/SW Co-design環境を軸として，設計の更に上位レベルからの自動化を図る。具体的には，C，C++言語を用いた上位言語設計に新手法を導入し，従来は人の手によってしかできなかった部分をも自動化していく。現在，C，C++言語やビヘイビア言語レベルでのパーティショニングや論理合成等の技術開発に注力し準備を進めている。

第二に，タイミング・ドリブン・レイアウト手法である。すでに一部導入されてはいるが，まだ実用化されているとは言い難い。求められているのは，レイアウト後にタイミングエラーがなく，しかもチップ面積を最小化できるような自動レイアウトシステムである。この要求に応えるべく，上位タイミング制約をもとにしたタイミングドリブン手法およびコンパクション(注8)手法などを早期に提供できるよう現在準備中である。

むすび

止まることを知らない回路の大規模化と複雑化の中で生き抜くためには，これに対抗し得る新たな設計手法・設計システムの導入が必須である。このシステムLSI時代において，もし旧来の設計手法をそのまま使ったとしたら開発期間の長大化と製品の低品質化はおろか，開発そのものが収束しない事態にもなりかねない。

本稿では，富士通で既に開発・提供を開始している新設計手法についてその主なもの，すなわち，HW/SW Co-design，新論理検証手法，論理フロアプランと再最適化，そして低消費電力設計手法を紹介した。しかし，設計データの大規模化は予想以上のスピードで設計システムの変革を要求している。この中で著者らがすべきことの意義は大きい。大規模化・高機能化する設計データに対して短期間でしかも高品質な開発を可能にする新設計手法を，早期かつ確実に提供していくのが著者らの使命と考えている。

（注7）回路がチップ上に占める面積。

（注8）レイアウト結果の面積をさらに縮小すること。