

LSIデザインソリューション

～機能検証ソリューション～

森 義一 齋藤 早苗

「半導体チップの上に集積されるトランジスタの数は18～24ヶ月ごとに倍増する」というムーアの法則は依然として続いている。LSIの集積度の向上に伴い、LSIに搭載可能なシステムの規模が増大してきたが、LSIの設計能力は必ずしもこれに追従できているとは言えない。LSIの設計能力を如何に集積度の向上に追従させるかがLSI設計の大きな課題になっている。その一方で、システムLSIはデジタル家電では製品差別化のためのキーデバイスであり、その開発競争は激しいものになっている。

また、近年LSIに関するビジネスは欧米ベンチャー、韓国・台湾勢の台頭により大きな変貌を遂げ、従来の垂直統合型モデルに加え、製造専門メーカ、ファブレスLSIベンダ、IP (Intellectual Property) プロバイダらの登場と活躍で水平分業型モデルとして自らの専門領域を活かしたビジネス展開を行っている。

沖ネットワークエルエスアイ（以下、ONWと略す）は、このような急変するLSIビジネスの中で、沖電気のネットワークシステム部門LSIデザインセンタが、多くのLSIを開発してきた実績と技術力を基に、2002年10月にLSIデザインに関するソリューション提供を行う会社として設立されたものである。ネットワークシステム部門の中で活動してきたことにより、LSIデザインソリューションを提供する会社でありながらも、システム技術力とネットワーク技術力の優位性を持ち、システム設計とLSI設計を組み合わせたソリューションを提供できることを特徴としている。また、沖電気グループの一員として、沖電気およびグループ企業との連携により、LSIデザインソリューションのみに留まることなくLSIからボード、ソフトウェアまでの一貫したソリューションを提供することも大きな特徴である。

本稿では、システムLSI設計において最大の課題になっている機能検証に対して解説し、ONWが提供するソリューションについて述べる。

LSIデザインソリューション

他社製品との差別化を行うための有力な手段として改

めてシステムLSIが注目されている。このようなシステムLSIを用いた装置の開発では、装置の主要部分の設計そのものがLSI設計となっている。このLSI設計に関しては数々のスキルが必要とされる一方で、慢性的なりソース不足、迅速な開発に向けたIPの活用、次々に開発される新設計技術への対応、など数々の課題が存在している。

ONWは、このようなLSI設計に関してお客様が抱えている課題を解決するソリューションを提供する。提供しているソリューションは、LSI設計・検証、IP開発・販売、コンサルティングの三つの柱から成っている。表1に提供している主なソリューションの内容を示す。

表1 ONWが提供するLSIデザインソリューション

ソリューション	内容
LSI設計・検証	システムLSI受託設計 (仕様設計/検証、機能設計/検証、論理設計/検証)
IP開発・販売	無線LAN向けIP 各種マイコン周辺IP 通信LSI向けIP
コンサルティング	設計/検証コンサルティング (教育・トレーニング、設計環境構築、デザインレビューなど) 技術調査

LSI開発における機能検証の課題

表2にLSIの設計工程と各工程における検証内容を示す。LSI設計は一般に仕様の設計、機能設計、回路情報を作成する論理設計、そしてLSIのレイアウトを作成するレイアウト設計という段階を順次踏んで行われる。

LSIの場合、製品化以降の修正は極めて難しい。この段階でバグが見つかる場合によってはリスピン（再作）となるが、その結果さらなる開発費に加え、製品投入が遅れることとなる。これは、製品のライフサイクルが短期間になっている今日では致命的な問題になりかねない。また、設計の後工程で不具合が見つかった場合には、工程を遡る作業手戻りが発生し、設計工数、期間の面で損失

表2 LSI設計工程と検証内容

設計工程	検証内容
仕様設計	方式検証 性能検証
機能設計	機能検証 (RTLと仕様の一致性の検証)
論理設計	論理検証 (ゲートレベルとRTLの一致性の検証) 擬似配線長遅延タイミング検証
レイアウト設計	実配線長遅延タイミング検証 レイアウト検証

が発生してしまう。したがって、設計の早い段階からバグを発見・修正することが重要であり、検証という作業が極めて重要視される。

検証とは、各工程においてそれぞれの設計結果の内容を確認する作業であり、同時に前工程の出力結果との一致性を確認する作業でもある。仕様設計の出力は仕様であり、機能設計の出力は仕様に基づきHDL (Hardware Description Language) を使用してRTL (Register Transfer Level) で機能を記述したコードとなる。論理設計ではRTLに基づいて、面積・タイミング等の制約を満たして作成された回路情報 (ゲートレベル) が、そしてレイアウト設計ではレイアウトそのものが出力となる。したがって各工程では仕様とRTLとの間、RTLとゲートレベル (ネットリスト) との間、ゲートレベルとレイアウトとの間で一致性が検証されなければならない。論理設計以降のRTLとゲートレベル、ゲートレベルとレイアウトとの一致性検証はEDAツールにより行うことが可能であるが、機能設計工程での仕様とRTLとの一致性検証は仕様記述そのものが形式化されていない今日ではEDAツールを用いて行うことはできない。そこで機能設計段階で如何に仕様とRTLとの一致を確認するかという機能検証が重要な課題になっている。

それでは機能検証は、どのように行われるのであろうか？ 仕様が確定すると、仕様に基づき各設計者が設計を行うが、この時に必ずと言っていいほど、仕様の誤解釈などでデザインに何らかの誤りが入る。デザインに誤りが発生する要因としては、仕様書自体に含まれる誤り (仕様ミス、記述誤り)、設計者による仕様の誤解釈、RTLモデリングの際に発生する誤り (記述ミス等) がある。

機能検証は、これらの誤りを検出することにより仕様との一致性を確認する作業である。仕様との一致性を見るための手段として最も一般的なのがシミュレーションによる確認である。シミュレーションは被試験対象を擬

似的に動作させ、その結果が期待した動作であるか否かを確認する作業である。シミュレーションでは、実行する試験の内容を可能な限り漏れなく作成し、着実に実行する必要がある。

システムLSIの機能検証では、規模の増大と機能の複雑化により検証項目・検証工数も急激に増加している。設計・検証の効率を改善するために再利用可能なIPを活用することも一般化したが、IP間の接続や組み合わせた場合の動作検証は依然として残ることになる。

また、最も難しい課題は極めて稀な条件で発生する事象や検証項目から漏れた項目などを、検証作業の中で如何に動作確認するかということである。この課題に向けたEDAツールや検証手法も提案されているが、一般的に新たな言語の習得などが必要で導入の壁が高く十分に活用されていない。

以下に、機能検証の課題をまとめる。

- 急増する機能検証のリソースの確保
- 第三者観点からの仕様と検証項目のチェック
- 検証項目漏れ・作業漏れなど人為的ミスの排除
- シミュレーション実行時の確実な動作チェック
- 新しい検証手法の導入と活用
- プロトタイプを活用したソフトも含めた検証

機能検証ソリューション

ONWは、ネットワークシステムのLSI設計技術と経験を基に上述した機能検証の課題に対してソリューションを提供する。

ONWとしては主として以下の四つの機能検証ソリューションを提供している。

- (1) 検証作業のアウトソーシング
- (2) 第三者観点からの検証サービス
- (3) 最新の検証手法を活用した高度な検証
- (4) プロトタイピングボードを活用した協調検証

以下、各項目について説明する。

(1) 検証作業のアウトソーシング

これは膨大な工数を要する検証作業の一部を請負、作業を行うものである。従来から設計受託会社等で行われているものと同一である。機能検証のアウトソーシングにより、お客様は社内リソースを競争優位性の源泉に集中させ、有効に活用することができる。

(2) 第三者観点からの検証サービス

前章で設計者による仕様の誤解釈や記述ミスの防止、検

証項目の網羅性が重要であることを指摘した。特に設計担当者と検証担当者が同一の場合には、仕様の誤解釈に気が付かず、問題が見過ごされ、後日バグとなって現れる事が多い。

第三者観点からの検証サービスは、ONWが仕様の段階から検証に参画することで、検証の観点から仕様の内容をチェックし、検証項目の策定、シミュレーションによる検証を行うものである。

図1に第三者観点からの検証サービス作業の流れを示す。お客様からは検証対象の仕様書とRTLコードを提供いただき、ONWにて検証項目をリストアップし、その検証を実施して問題が検出された場合にはお客様へ問題点を報告するものである。

このような第三者観点からの検証を行うことで、仕様の解釈誤りや検証項目漏れなどの人為的ミスを削減することができ、検証の品質向上を図ることができる。

(3) 最新の検証手法を活用した高度な検証

機能検証の課題に対してはEDAベンダからさまざまなツールや手法が発表されており、機能検証の効率・品質の向上に効果を上げている。特に、90年代末に登場した、ランダム検証、アサーション・ベース検証などの手法は機能検証の方法論にも大きな影響を及ぼすものとなっている。

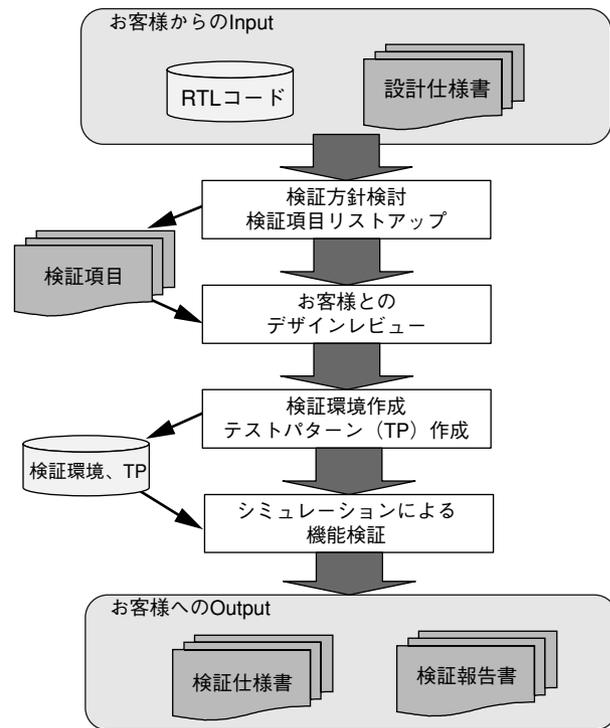


図1 第三者観点からの検証サービスの流れ

*1) OpenVeraはSynopsys Inc.の登録商標です。

表3 最新の検証手法

検証手法	内容
アサーション・ベース検証	検証専用言語により仕様を記述し、シミュレーション実行中に動作をモニタする、またはフォーマル検証によりRTLと仕様の一致性を確認する
ランダム検証	入力パターンをランダムに自動生成することにより、検証の網羅性を高め、コーナーケース・バグをなくす
機能カバレッジ観測	仕様で定義された動作を全て試験しているか観測し、検証漏れをなくす

表3に最新の検証手法とその特徴を示す。これらの手法は、シミュレーション実行中のモデルの振る舞いの監視や大量の入力パターンの生成など従来の検証手法ではカバーできない検証も実現する。このような設計手法が登場したのは、検証が人手に依存する部分が極めて多く、テストデータの不足やチェック漏れなどの問題が起きていたことによる。これらの手法は、ツールでできる作業はできるだけツール側に行わせ、パターンの生成や各種のチェック、テストの網羅率の測定などに人為的な誤りが入るのを防いで検証の品質を向上させるものである。

以下、具体的に最新の検証技術について説明する。

①アサーション・ベース検証

アサーションとは「本来成立すべき条件」のことを指している。たとえば、LSI自体や内部ブロックなどの通信は定められたプロトコルに基づき行われるが、このプロトコルにおいて「成立する(しない)条件」を明確化して記述したものがアサーションである。

アサーションが着目される背景には、専用記述言語、および処理系が開発されたこと、モデルと仕様との間の等価性を数学的に証明するフォーマル検証ツールの入力としてアサーションを活用し、モデルを静的に検証することが可能になったことが大きい。

アサーション・ベース検証は、モデルが満たすべき仕様あるいは違反事項の条件をアサーションとして専用の言語を用いて記述し、モデルとともにシミュレーションを行うことで振る舞いの正当性確認や異常動作検出などを行う。インタフェースのプロトコルや内部回路の制御を行うステートマシンの動作が指定した通りに振舞うか否かをシミュレーション実行中に監視する。図2に検証環境におけるアサーションの位置付けを示す。

アサーションを記述する言語としては、TemporalE、OpenVera^{*1)} Assertionなどが現在既に使用されている。さらに将来に向けて業界標準を策定する団体である

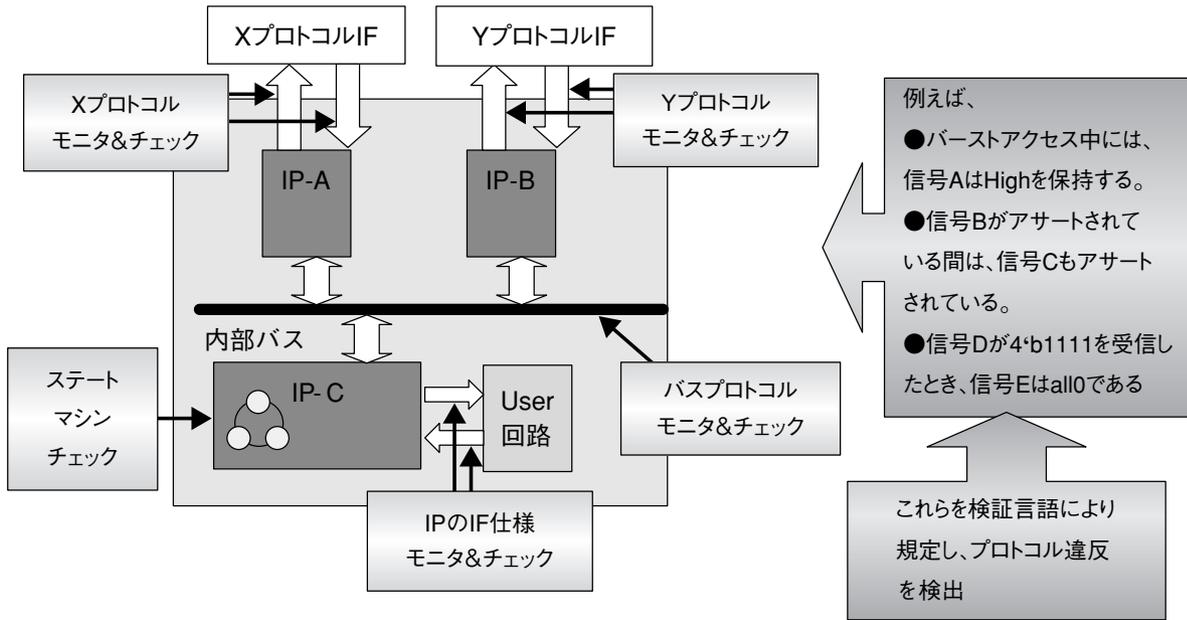


図2 検証環境におけるアサーションの位置付け

Accelleraにおいて、PSL (Property Specification Language)、SVA (SystemVerilog Assertion) などの専用言語の標準化作業が進められている。

専用言語を用いることで、チェックすべき内容をチェック方法まで指示することなく、容易に記述を行うことができる。また、シミュレーション実行中に常時監視が行われることで、意図に違反した動作を大量のシミュレーションの中でも見つけることが容易になる。

②ランダム検証

シミュレーションは、被試験対象に対し、入力パターン (Stimulus) を与え、その応答を見ることで行われる。一般に、機能検証は、検証項目をリストアップして項目に沿ったパターンを用意することで行われる。

この時のパターンは、これまで人手あるいは何らかのジェネレータで作成していたが、この方法ではパターン作成者のスキルに依存することになり、検証漏れなども起きかねない。そこで、パターン生成をツール側でランダムに行うテストベンチ生成ツールが90年代後半より登場してきた。検証技術者はランダムパターンの発生方法を、制約条件を与えて制御する。具体的なツールとしてはSpecmanElite^{*2)}、Vera^{*3)} などがある。これらのツールで使用する言語はC++などの言語が持つ機能に加えランダム検証を効果的に行うための機能を持たせた専用言語 (e言語、OpenVera言語など) が使用されており、検証対象および検証手法に応じて専用言語を用いて検証

*2) SpecmanEliteはVerisity Design Inc.の商標です。 *3) VeraはSynopsys Inc.の登録商標です。

環境を構築するものである。

このランダム検証手法は、検証の網羅性を高め、設計者や検証技術者による把握が難しく極めて稀な条件でしか発生しないバグであるコーナーケース・バグなどを検出するために開発されたものである。

③機能カバレッジ観測

機能カバレッジとは、被試験対象で定義した状態なり、振る舞いがシミュレーション途中で発生したか否かをモニタして結果をレポートする機能である。ランダム検証のような場合は、意図的なパターンの生成が困難であるので、この機能を用いて仕様で定義された機能がシミュレーションで満たされたか否かを監視させ、その結果を見て検証の充分性を判断するものである。

以上のような最新の検証手法を取り入れた検証環境 (テストベンチ) を図3に示す。

このような検証環境は、被試験対象に対して大量のテストパターンを適用でき、かつ各種のチェックもツール側で行わせることができ、網羅的な検証を行うには有効なものである。しかし、テストベンチ構築にはツール専用の言語、あるいはC++等の言語を用いる必要があり、従来のHDLベースでの検証環境のスキルだけでは対応が難しいという課題がある。また、アサーションの記述にしても専用言語の習得が不可欠である。

したがって、このような検証手法を用いて、より効果

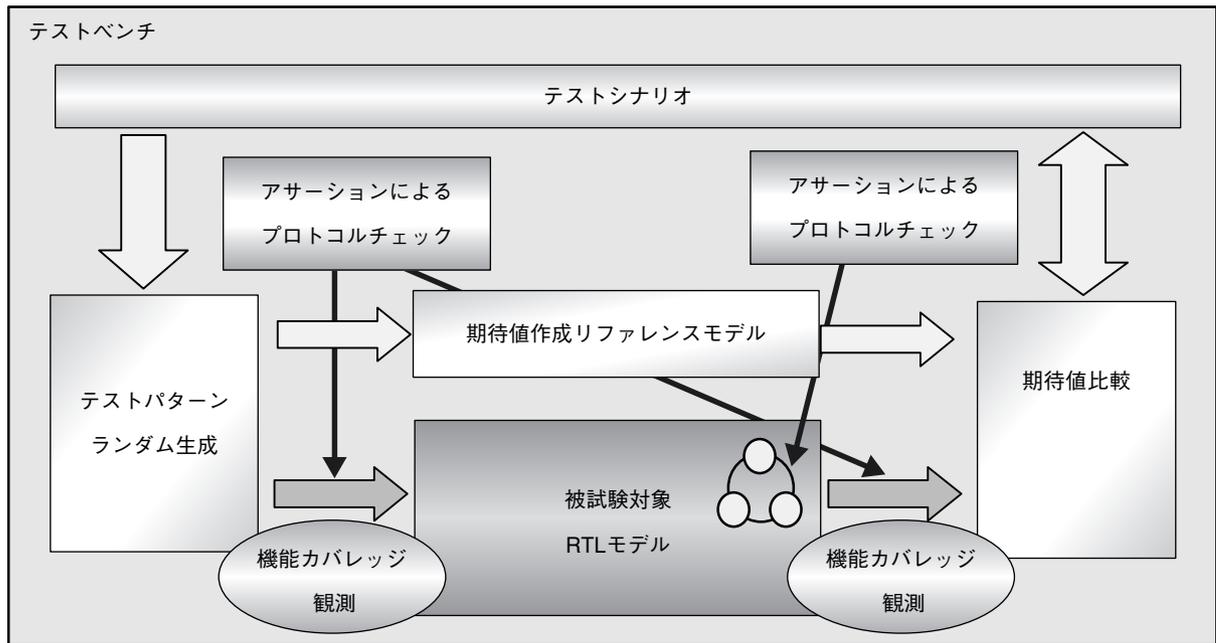


図3 最新の検証手法を取り入れたテストベンチ

的な検証を行いたいと考えているお客様でも、上述のような課題等により最新の検証環境を適用していない場合が少なくない。ONWでは、このようなお客様に対して、高度な検証の効果が得られるようにお客様の要望に応じたレベルでのソリューションを提供している。

(4) プロトタイプボードを活用した検証

機能検証では、シミュレーションでの検証に留まらず、RTLコードをFPGAに実装し、ハードウェアのプロトタイプとして利用する場合がある。たとえば、ドライバソフトなど一部ソフトウェアも含めた検証や実際の通信を行う通信系の検証がある。また、音声・画像などメディア処理では、大量のデータ処理を行うために高速シミュレーションが必要となる場合もある。

ONWでは、主に μ PLAT^{*4)}プロトタイプボード¹⁾を活用して μ PLATと組み合わせたプロトタイプ評価システムを開発して提供している。

このプロトタイプボードの利用方法としては、システムバスに接続されるIPの評価に加え、 μ PLATベースのシステムLSIのハードウェア・ソフトウェアの協調検証がある。プロトタイプボードを活用することで、LSIとしての検証に加え、ソフトウェアに関するさまざまなデバイスドライバの開発やリアルタイムOSの移植、通信ミドルウェアなどのソフトウェアのデバッグが可能となる。この結果、システムLSIには不可欠なハード

*4) μ PLATは沖電気工業(株)の登録商標です。

ウェア、ソフトウェアを一体化したシステムとして検証が実現できる。

機能検証ソリューションの効果

ONWの機能検証ソリューションは、会社設立以前からの活動も含め約2年以上の実績を有している。この間、お客様の設計したモデルを検証した結果、いくつもの問題を発見してお客様のLSIの品質向上に貢献してきた。

既に市場に投入された製品に関する追加検証を行うことで新たな問題を発見し、さらなる高品質化に貢献することができた。また、サインオフ前に問題を検出することで最終的なエンドユーザへの影響を最小限に留めることができたなど、お客様にとってもメリットを享受していただいている。

また、アサーション・ベース検証やランダム検証など新しい検証手法は有効な方法としては認識されるようになったが、前述の通り一般的に導入を行う場合には専門言語のスキルなどが障害になっている。

ONWでは、既に導入済みのお客様に対しては作業の一部を分担するアウトソーシングとして、またこれからの導入を検討されているお客様に対しては、導入に当たったのコンサルティングや教育・トレーニング等のサポートを実施し、新しい検証手法の活用に向けた機能検証ソリューションの提供を行っている。

将来への展望

システムLSI開発において最大の課題となっている機能検証について解説し、ONWが提供するソリューションについて述べた。

大規模システムLSI設計を効率化するために、従来のHDLによるRTL設計に加え、より抽象度の高いレベルからの設計を可能とするためにC++言語をハードウェア設計向けに拡張したSystemC^{*5)}言語をシステム設計に活用する動きがある。また、RTL設計においても従来の設計を対象としたVerilog^{*6)}-HDLから、設計と検証を対象とした新しい言語SystemVerilogの標準化がAccelleraで進められている。このような標準化の動きを見ても、従来の設計に重点が置かれていた時代から、設計と検証の両方が重視される時代への流れを見ることができる。

ONWでは、このようなSystemC言語を用いたシステム設計段階での検証、また次々に登場する検証用専用言語や新たな検証手法への対応を行うことで、今後も、高水準の検証技術を確認し、お客様に対して効率的かつ高品質の機能検証ソリューションを提供していく。◆◆

参考文献

1) 稲葉：μPLATのプロトタイプング技術、沖テクニカルレビュー196号、Vol.70 No.4、pp.52-55、2003年

● 筆者紹介

森義一：Yoshikazu Mori.株式会社沖ネットワークエルエスアイ
デザインソリューション本部

齋藤早苗：Sanae Saitou.株式会社沖ネットワークエルエスアイ
デザインソリューション本部

TIPS

【基本用語解説】

EDA ツール (Electronic Design Automation Tool)

電気回路の設計・検証を行うツール。

HDL (Hardware Description Language)

ハードウェア記述言語。LSI やシステムなどの設計データを記述する。代表的な言語に Verilog-HDL、VHDL がある。

PSL (Property Specification Language)

アサーションを記述するための専用言語。Accellera にて標準化が進められている。IBM 社が開発した Sugar 言語をベースにしている。

RTL (Register Transfer Level)

HDL による記述レベルの一つ。LSI の機能をレジスタ (フリップフロップ) 間の組み合わせ回路で表し、レジスタ間の転送という形で機能を表現する方法のこと。

Verilog-HDL

代表的な HDL の一つ。元々は Verilog-XL シミュレータ用に開発された言語。現在は IEEE 1076 として標準化されている。

アサーション

アサーションはデザインの中で「成立すべき条件」を表現したもの。アサーションのチェックを行うことで、デザインが仕様を満足させるか否かを判断する。アサーションの記述には PSL 等の専用言語が使用される。

ゲートレベル

HDL による記述レベルの一つ。LSI の回路情報をライブラリのゲートとその接続関係で示す。ネットリストとも呼ばれる。

システム LSI

従来のボードや装置に相当する機能 (CPU、メモリ、各種周辺など) をシングルチップに集積した LSI、SoC (System on Chip) とも呼ばれる。

Accellera

EDA 分野で HDL などの標準化を進める団体。Verilog-HDL、VHDL の標準化を進めていた団体が合併し、現在は SystemVerilog や PSL などの標準化を進めている。

*5) SystemCはOpen SystemC Initiativeの登録商標です。 *6) VerilogはCadence Design Systemsの登録商標です。