SYNOPSYS®

VCS 7.0が実現するSmart Verification

現在、機能検証には膨大な期間が費やされています。多少の相違はあるものの、多くの技術者は、設計期間全体の70%以上が機能検証に費やされているという見解で一致しています。設計者はチップのインプリメント作業よりも、その動作の検証とデバッグに多くの時間を費やしているわけです。一方、最近の半導体プロセスには以前よりも多額のコストが伴うため、検証結果の品質はさらに重視されるようになっています。つまり、回路設計の検証に時間をかけすぎると市場参入のチャンスを逃す原因になりかねないものの、かといって検証が不十分なままテープアウトを迎えるとコストの増大と販売後の製品へのバグの混入を招き、ひいては大損害をもたらすことになりかねません。

Smart Verification

EDA各社は、検証技術を向上させることにより検証のボトルネックを改善すべく努力を続けてきました。ワークステーションの性能が向上し、低価格になったことで、各企業は大規模なサーバ群を配備し、大掛かりなシミュレーションに対応できるようになりました。シミュレーション・アルゴリズムの発達およびコンピュータの処理能力の向上により、危機的状況は緩和されていますが、IC設計チームが直面している検証の難問に対処するためには、さらなる革新が必要とされています。つまり、現在の高性能シミュレーション・ファームをさらに先進の検証技術で補わなくてはなりません。現在、最先端の設計チームは、高性能テストベンチ自動生成、アサーション・ドリブン検証、コード・カバレッジ解析、フォーマル検証技術といった革新的な単体ツールを用いてシミュレーションを補完しています。しかし、検証の中心にあるのは依然としてHDLシミュレーションであり、これらの先進技術もHDLシミュレーションとの相互作用を必要とします。これらの技術すべてを同一のプラットフォームに統合し自動化を進めたもの、それがシノプシスのSmart Verificationです。Smart Verification手法は生産性/処理時間/品質の向上によって検証の所要時間を短縮するとともに、検証の信頼性の向上を可能にします。これらの先進技術をシミュレーションにとってのネイティブな機能として装備したSmart Verification手法こそが、最高の検証速度と自動化を保証する唯一の方法なのです。

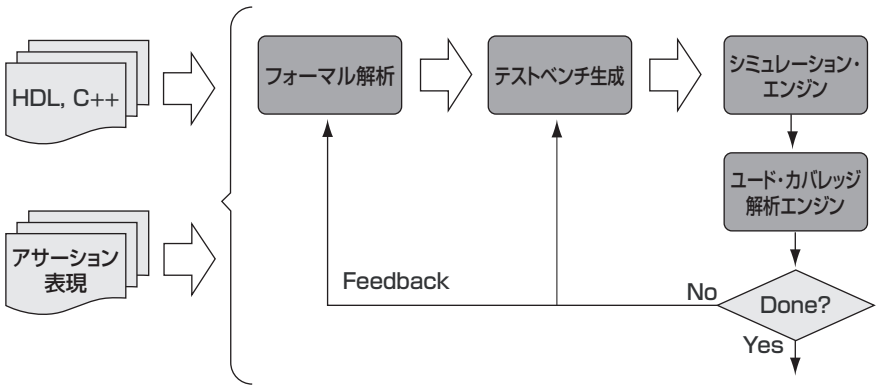

図1：Smart Verification

VCS 7.0概要

VCS 7.0は、世界中で実績のある高性能HDLシミュレーション・テクノロジを用いてSmart Verification環境を構築するためのオープン・プラットフォームを提供します。

VCS 7.0の新技術には、OpenVeraテストベンチ構文サブセットであるVeraLiteを用いたコード生成のネイティブ・サポート、アサーション・ドリブン検証を可能にするOpenVeraアサーション記述（OVA）を用いたコード生成のネイティブ・サポート、高いシミュレーション速度を実現するDirectCインターフェイス経由でのSystemC記述のネイティブ・サポート、内蔵のコード・カバレッジ解析機能Observed Coverageのネイティブ・サポートがあります。これらの技術はVCSにネイティブな機能として提供されているので、適用しやすく、最高のスピードを実現し、より短い時間でより多くの検証を可能にします。

抽象度の高度化によるテストベンチの生産性

OpenVeraをはじめとする抽象度の高いハードウェア検証言語（Hardware Verification Language = HVL）は、検証環境開発をスピードアップし、Verilog-HDLやVHDLまたは内製言語などのハードウェア記述言語（Hardware Description Language = HDL）に比べて検証の生産性をはるかに向上させることが実証されています。HVLは高品質なテストベンチの迅速な作成を支援します。

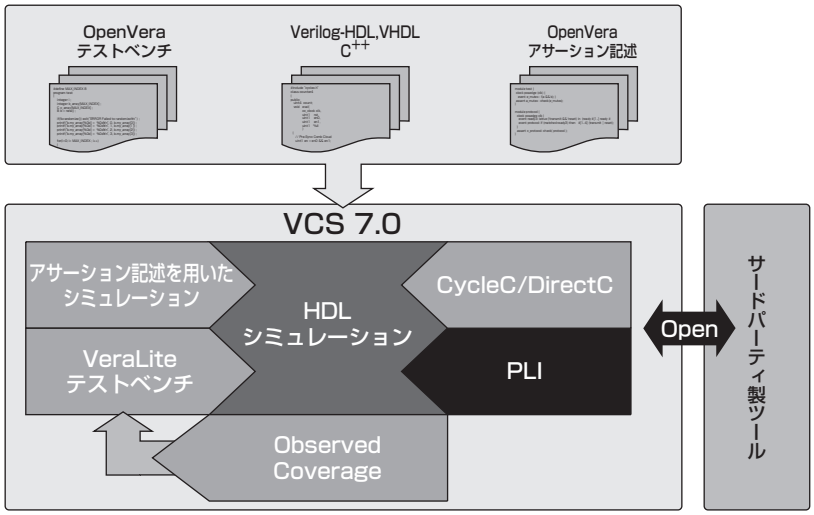

図2：Smart Verificationプラットフォーム VCS 7.0

設計検証のためのテストベンチ作成には、包括的なデータ構造と多機能な言語構文が必要です。OpenVera HVLは、HDL、C++、Javaの使いやすさと記述能力に、機能検証に的を絞った構文を組み合わせた、使いやすく、覚えやすい言語で、テストベンチ、アサーション、プロパティの記述開発に理想的です。OpenVeraは、検証環境とテストベンチの構築を短期間で実現します。

シノプシスのVCS 7.0は、OpenVeraテストベンチ構文のサブセットであるVeraLiteによるネイティブ・コード生成機能を提供します。VeraLiteを用いて、ユーザはVerilog-HDLよりも高い抽象度でテストベンチを記述でき、Verilog-HDLベースのテストベンチを容易に記述することができます。抽象度の高いテストベンチは一から作成するのではなく、セマフォ、リスト、メールボックスおよびクラスといった組込みプリミティブを用いて迅速に生成することができます。VCSはOpenVeraのネイティブ・コード生成を提供するので、全体的な検証性能が飛躍的に向上します。

VeraLiteによりVCSの機能が拡張され、Verilog-HDLを用いる設計者は、新しい環境への移行というリスクを負わずに、より高い抽象度での検証が可能になります。また、設計者の使い慣れたVCS環境で、Smart Verification手法のメリットを享受することができます。

●フロー・コントロール – コンカレンシー・コントロール ・fork-join all ・fork-join none ・fork-join any – シーケンシャル・コントロール ・case, repeat, for, while, break – タスクとファンクション ・リエントラント・タスク ・コール・バイ・バリュー ・コール・バイ・リファレンス	●ハイレベル・データタイプ –アレイ –ビットとビットベクタ –クラス –整数 –リスト –メールボックス –ストリング
●インタープロセス・コミュニケーター –イベント –セマフォ – syncs – triggers	●期待値 –floating – full – restricted

図3：VeraLite構文

アサーションによる高抽象化

アサーションは回路動作を明確に定義するための抽象度の高い表現方法で、ダイナミック・シミュレーションにおいて不適切なデザインの動作を監視したり、フォーマル検証においてデザインのプロパティを記述し、デザインの論理的な正しさを完全に検証することができます。OVAは、テスト対象のデバイスの複数のサイクルやモジュールにまたがる時間的動作を正確かつ簡潔に記述するために開発された言語です。一般的に使用されているVerilog-HDLやVHDLといったハードウェア記述言語は、手続き型構文によって1サイクル遷移のハードウェア動作をモデリングするために開発されたものです。このような実行モデルはマルチサイクルの遷移動作を定義するには不十分です。OVAでは、入出力動作、バス・プロトコル、関数を容易にわかりやすく表現することができます。OVAで生成されるアサーションはHDLよりも通常3〜5倍も簡潔で、アサーションの作成とデバッグに費やされる時間を大幅に短縮することが可能です。

```
    module protocol {
  clock posedge (clk) {
    event bus_ready3: istrue (!transmit || !reset)
     in (bus_ready #1 (#[0..] bus_ready)*[2]);
    event protocol: if (matched bus_ready3) then
    #[1..4] (transmit || reset);
   }
assert c_protocol:  check ( protocol );
}
```

OVAを用いたアサーション記述

仕様：*bus_ready* 信号が断続または継続的に3クロック・サイクル以上の間、真ならば、*transmit* 信号は *reset* 信号が入らない限り4クロック・サイクル以内に真にならなくてはいけない。

```
module protocol_checker ( clk, rst, clear,  bus_ready, transmit,  data_lost );
input   clk, rst, clear, bus_ready, transmit;
output data_lost;
reg          data_lost;
reg [3:0] cycles;
wire reset = rst || clear;
always @ (posedge clk or posedge reset)
   if (reset) begin
       cycles <= 0; data_lost <= 0;
          end
   else if (cycles == 8) begin
       data_lost <= 1; cycles <= 0;
       $display("Assertion Failure");
   end
   else if ((cycles > 0) && (cycles < 8) && transmit)
   begin
       cycles <= 0;
       $display("Assertion Success");
   end
   else if (bus_ready) || (cycles > 3)) begin
       cycles <= cycles + 1;   //waiting
   end
endmodule
```

Verilog-HDLを用いたアサーション記述

図4：アサーション+OVAとVerilog-HDLの比較

高性能なアサーション・ベースの検証を容易にするため、VCS 7.0ではOVAシミュレーションと監視のためのネイティブ・コード生成を提供します。VCSを使用することで、設計者は回路の目的の動作をOVAで記述し、作成した回路をシミュレーション時に継続的に監視することによって回路の意図に反した動作をおこすテストを検出することができます。VCSにおけるOVAサポートでは、アサーションをHDLコードに組み込むことも、別ファイルとして作成することも可能です。

ハイレベル C++ モデリング

一部の設計チームでは、C/C++のハイレベル・データ構造を利用してデザイン・ブロックをモデリングすることによって検証プロセスの生産性とスピードを向上させています。設計者がC++モデルをHDLシミュレーションへ容易に統合できるようにするため、シノプシスはVCS DirectCインターフェイスを開発しました。VCS DirectCインターフェイスはマイクロプロセッサ、画像圧縮、ネットワーク・アプリケーションなど、幅広い設計に使用され成功を収めています。他のVerilog-HDL C/C++ソリューションと異なり、C/C++モデルをHDLシミュレーションに統合するに当たっては、PLIについての知識を必要としません。DirectCインターフェイスはVerilog-HDLの機能性を拡張して、さらなる性能の向上を可能にします。

VCS 7.0ではCycleCとのVCS DirectCインターフェイス機能を拡張することにより、Verilog-HDL 環境でのC/C++コードのシミュレーションの大幅な高速化を可能にしています。CycleCは同期RTL設計のためのC++コーディングスタイルであり、これによりシミュレーション処理速度を10倍まで向上させ、メモリ使用量を4分の1にまで削減することが可能です。CycleCでは、同期ブロックをCycleCのコーディング・ガイドラインに従ってRTL C++で作成し、DirectCインターフェイス経由で円滑にVCSシミュレーション・エンジンに統合することができます。VCS 7.0のCycleCは、コードの品質と一貫性を保つため、スタイル・チェッカが提供されています。CycleCのコードは、PLIのオーバーヘッドを生じさせずにHDLコードとともにネイティブにシミュレーションされるので、既存のVerilog-HDLシミュレーション環境で最高の処理速度と検証容量が得られます。また、VCS 7.0にはCycleCベースのコードをVerilog-HDL RTLに変換して標準のRTL合成フローで使用できるようにするためのユーティリティも備わっています。

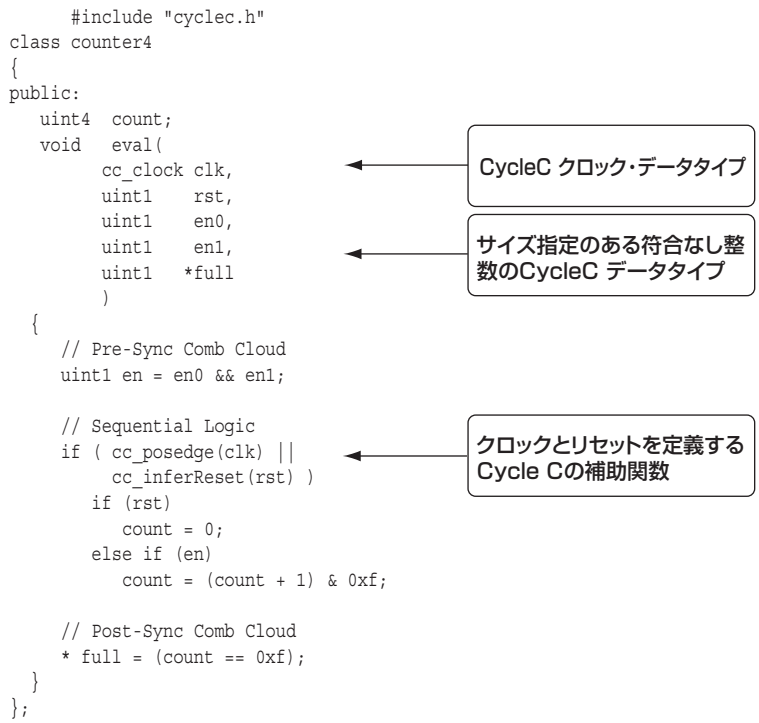

図5：CycleCによるカウンタの記述例

VCS 7.0は、抽象度の高いC/C++モデルをVerilog-HDLの中で容易に使用できるようにします。既存のRTL検証/インプリメンテーション方式を活かしながら、高い抽象度での設計モデルの高速シミュレーションを可能にします。

検証の収束を制御するカバレッジ測定

コード・カバレッジ解析では、検証環境の品質が測定されます。コード・カバレッジ解析なしでは、回路のどの程度が実際に検証されているのかを判断できません。テストベンチの効率はコード・カバレッジ解析によって判断することができます。これによって設計者は、十分に検証されていない回路部分に対してテストを調整したり焦点を変えたりすることができます。また、コード・カバレッジ解析を実行することにより、既に検証済みの領域や機能に試行を繰り返さずに済むため、シミュレーション期間を無駄に費やすことがなくなります。コード・カバレッジ測定は、検証の終了を知るために役立ちます。

従来のシミュレーション・カバレッジ解析ツールはVerilog-HDL PLIなどの低速APIを通じたシミュレーション動作の監視に頼っていました。この方式では通常2〜10倍もシミュレーション速度が低下し、Smart Verification技術を用いることができません。VCSは、コンパイラ内部でカバレッジ・データのネイティブ生成を行うことでこの障害を取り除きました。このことにより、カバレッジ解析がさらに簡潔になり、シミュレーションのオーバーヘッドが削減され、設計チームは特にリグレッション・テストにおいて、カバレッジ測定をより頻繁に利用できるようになりました。

VCS 7.0では、内蔵のコード・カバレッジ解析エンジンがさらに改善されています。VCS 7.0では、新しい高度なObserved Coverage（OBC）測定機能を組み込むことによって、内蔵のライン、トグル、有限ステートマシン、ステートメントの各カバレッジ解析機能が拡張されています。OBCはシミュレーション/スティミュラスを解析して、スティミュラスの動作がユーザ指定の観測点（通常はデザイン・ブロックの出力ピン）に伝搬しない（つまり観測できない）原因となっている回路内の問題点を特定します。期待される観測点にスティミュラスが伝搬しないということは、回路設計またはテストベンチに潜在的な問題点があるということになります。

OBCは、従来のコード・カバレッジ解析技術を超越した解析結果を提供し、従来のカバレッジ測定機能では見つけられなかった問題点を検出します。OBCは、テストそのものの品質を分析し、テストの改善とバグ検出を的確に行うための指標を提供します。

フォーマル解析による検証ソリューションの拡張

制約条件によって制御される擬似ランダム・スティミュラス生成およびシミュレーション・カバレッジ解析により、テストベンチの効率を大幅に向上させることができます。Smart Verificationは、回路デザインのフォーマル解析に基づいた、高カバレッジを達成するための一連のテスト（アダプティブ・テスト）生成を提供することで、このプロセスをさらに効率化します。

Smart Verificationは、所定の回路動作を仕様ごとに、スタティックに実証するためのプロパティ検証技術をサポートしています。この場合は、テストベンチの代わりに、期待される回路動作を示すプロパティを作成します。一例として、“acknowledge must follow request within a window of 10 cycles”というプロパティがあります。プロパティの記述には、OVAなどのアサーション言語が最も適しています。フォーマル・エンジンはOVAプロパティをスタティックに解析し、プロパティを実証または反証するために、回路内の違反をチェックします。すべてのプロパティが完全に検証されるので、ダイナミック・シミュレーションでプロパティを検証する必要はありません。これにより、ダイナミック・シミュレーションの負荷が大幅に削減され、検証の信頼性が増大します。

アダプティブ・テスト生成は、回路デザインと検証環境を解析して、検証実行時に達成され得るカバレッジを実際にカバーする一連のテストを自動的に生成する機能です。これは、フォーマル・エンジンを使用してデザイン・ブロックの算術解析を実行し、回路の状態空間をカバーする目的のテストを生成することによって実現されます。さらに、不到達解析では、いずれの状況においても到達し得ない状態が特定されます。カバレッジの増加と不到達情報により、検証エンジニアは、回路に対する検証のカバレッジ収束達成において、高い信頼性を得ることができます。

Smart Verification手法は、ダイナミック・シミュレーションとフォーマル検証の長所を併せ持つことにより、カバレッジの目標値を迅速に達成します。Smart Verificationでは、最初に制約条件によって制御されるランダム・スティミュラス生成を実行してから、カバレッジの増加率を監視します。この率が減少または平坦になった時点で、ダイナミック・シミュレーション環境は停止し、制御をフォーマル解析エンジンに渡します。フォーマル解析エンジンでは、回路に関する情報を使用して未テストの部分に的を絞った新しいテストを作成することにより、カバレッジの範囲を増加させ、検出が困難なバグを特定します。フォーマル検証エンジンを使用したアダプティブ・テスト生成とプロパティ・チェックの組み合わせにより、エラーのない回路ブロックの作成が可能になります。これはプロジェクト全体の検証期間の大幅な短縮につながります。

VCSに緊密に統合されたフォーマル検証技術により、検出が困難なバグを特定するための検証機能が強化され、回路の品質が向上します。

まとめ

Smart Verification手法は検証の課題を克服するための要となるソリューションです。VCS 7.0は、発展を続けるシミュレーション技術を改革する大きな一歩です。VCS 7.0は検証の処理速度/効率/信頼性を向上させるSmart Verification技術をVerilog-HDL設計者に提供します。

VCS 7.0はVeraLiteテストベンチ、OpenVera Assertions、CycleC、Observed Coverage技術をネイティブ・サポートし、従来のHDLシミュレーションの概念を全く新しいものにします!

日本シノプシス株式会社

http://www.synopsys.co.jp