

SystemVerilog言語の開発歴史

SystemVerilogは、2005年にIEEE Std. 1800として標準化された言語です。その前身はハードウェア設計言語であるVerilog HDLです。Verilog HDLはもともとGateway Design Automationという会社が自社シミュレータで動作する言語として開発されました。1990年代前半は設計言語としてはVHDLとVerilogの言語競争が激しく、言語を広める戦略として取られた策が、独自言語からオープンなパブリックドメインへの移行でした。さらに1995年にはIEEE Std. 1364として標準化されました。ModelSimはもともとVHDL言語のシミュレータですがこの頃からVerilogもサポートを開始し、両方の言語を混在させてシミュレーションすることが可能なSKS – Single Kernel Simulationテクノロジーをいち早く完成させました。その後、IEEE Std. 1364は2001年に大きな改訂を、2005年にマイナーな改訂をしています。

IEEEでは半導体開発に関わる言語やライブラリフォーマットなどを標準化していますが、当時のIEEEでは標準化プロセスが長期化する傾向にあり、半導体やEDA技術の進化に追従できるかが大きな懸念であり課題でした。そこで当時のOVI – Open Verilog InternationalとVHDL Internationalの2つの組織を併合し、IEEE標準化プロセスを加速（アクセラレート）させる目的でAccelleraという標準化団体が生まれました。その後SPIRIT ConsortiumやOpen SystemC Initiativeも併合し新たにAccellera Systems Initiativeと改名し、現在に至るまで多くの標準化に関する実績を残しています。

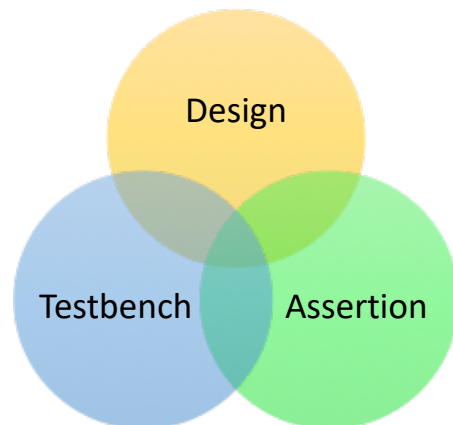
Accelleraは半導体の微細化が進み、設計の大規模化、複雑化、機能検証の難度化が大きな課題になるとの予測から新たなプロジェクトを開始しました。SoC – System on Chip時代、すなわちチップ上にシステムを構築できる時代に突入した背景から、Verilogを改めSystemVerilogという名称にしました。標準化にあたっては既存言語の寄贈も募り、当時あったSUPERLOGやVeraといった言語もプロジェクトで共有され、さらにオブジェクト指向の要素も導入され、標準の原型としています。



SystemVerilog言語の構成

SystemVerilogは大きく設計構文、アサーション構文、テストベンチ構文から構成されています。設計構文の主な特徴には、簡便なバス接続、インタフェース、多次元配列や連想配列、列挙型のサポート、外部Cモデルとの双方向呼出しなどがあります。またアサーション構文の主な特徴にはイミディエートとコンカレントのサポート、信号間の時相的な関係や演算子によるプロパティ表現と検証ツールへのディレクティブなどがあります。そしてテストベンチ構文では、テストのランダム化とその際の制約指定、機能カバレッジとその統計的メソッド、クラスによるオブジェクト指向の構文サポートなどがあります。特にアサーション構文やテストベンチ構文は機能検証の品質と生産性向上を目的としたもので、RTLよりも上位のトランザクションレベルのテスト生成や、標準の検証メソッドロジであるUVMを構築する上で欠かせない構文です。

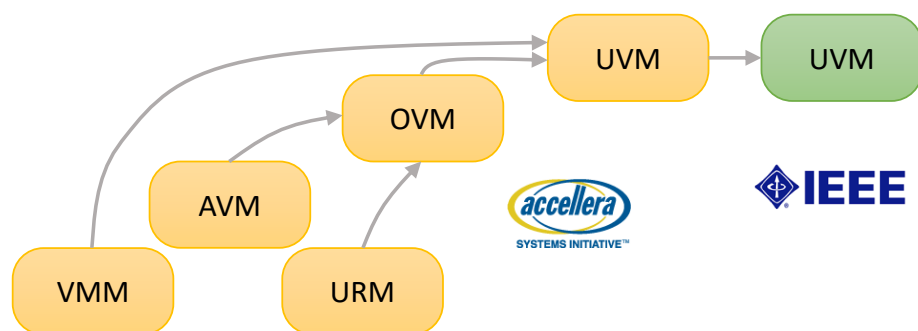
SystemVerilogの設計構文、アサーション構文、クラスによるオブジェクト指向記述はModelSim DEやQuesta Core、Questa Prime/Ultraでサポートされています。またランダムテストやその制約設定、機能カバレッジ、UVM検証メソッドロジなどはQuesta Prime/Ultraでサポートされています。またModelSim PEでもSystemVerilogの設計構文とクラスによるオブジェクト指向記述はサポートされています。



SystemVerilogベースの検証メソッドロジ : UVM

AccelleraではSystemVerilogを用いた検証メソッドロジであるUVM – Universal Verification Methodologyを標準化しました。むしろ検証メソッドロジを標準化することを強く意識しながらSystemVerilogが策定されたと言っても過言ではありません。この標準化は従来の言語仕様を策定する作業とは異なり、検証メソッドロジで使用されるクラスライブラリを開発し標準化しなくてはならず、Accelleraにとっても大きな挑戦でした。

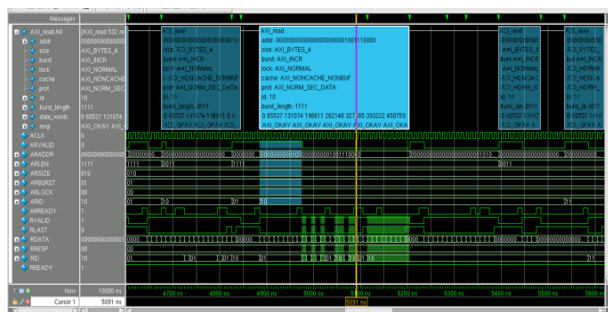
UVMが目的とするところは、ブロックレベルからチップレベルまで、1プロジェクトから他のプロジェクトへ、1企業から他社へと、テストベンチの再利用性を高め、業界全体で検証効率を引き上げることでした。バスファブリックやインターフェースなどのプロトコルは高い流用性が求められ、それまではEDAベンダーごとに異なる検証メソッドロジをベースにしていたのですが、UVMの出現によりその業界地図が一気に変わりました。現在はUVMを中心に検証IPが流通しています。



UVMはシノプシスがVMM、メンターがAVM、ケイデンスがURMをAccelleraに寄贈する形で標準化されました。特にAVMは早くからオープンソースで提供され、その後の標準化の方向に大きな影響を与えています。UVMはIEEE Std. 1800.2として2017年に標準化されています。

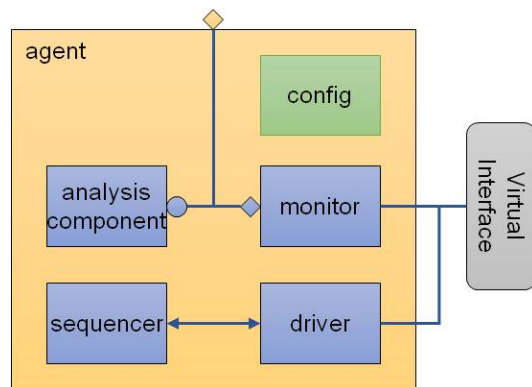
UVMランザクションレベル

UVMではランザクションレベルのテストを扱います。例えば右の波形ビューワにある水色の大きな箱は、AXI_readのランザクション実行を示します。アドレスやデータ、その他制御信号などの詳細はアトリビュートとして確認することができます。RTLのビットレベルの信号を扱うのに対して、テスト作成もデバッグも格段に効率が上がります。またテストをランダム生成し、想定できないシナリオも自動的に生成させ、カバレッジによりテストの完全性を測定します。



UVMベースの検証IP

UVMではエージェントと呼ばれるコンポーネントを基本に検証IPを構成します。バーチャルインターフェースで接続する信号からモニターやドライバを介してランザクションレベルへと抽象度を上げます。これにより高い抽象度でカバレッジ測定用の解析や、よりシナリオに近いシーケンスによるテストを実現します。エージェントはデザインを駆動するアクティブモードと、モニターを行うパッシブモードにコンフィギュレーションし、テストベンチ全体を構成します。



IEEE Standard LRMの入手

IEEEから発行されているLRM – Language Reference Manualは、IEEE非会員は購入しなくてはなりません。AccelleraがスポンサーであるIEEE Get Programでカバーされる標準規格は、無償で入手することが可能です。ウェブサイト – www.accellera.org – の Downloads → IEEE Standards から、ここで紹介しているSystemVerilogやUVMなどのリファレンスをダウンロードすることができます。