

SmartSpiceの Verilog-Aパーサの解説

はじめに

近年、アナログ/ミックスド・シグナル設計において、Verilog-AMSハードウェア記述言語 (Verilog-A) が広く用いられるようになりました。これに伴い、多くのEDAベンダがVerilog-Aに対応したシミュレーション・ツールを提供しており、SIMUCADのSmartSpiceも、数年前にVerilog-Aをサポートしました。最近、SmartSpiceのVerilog-Aパーサに、設計シミュレーションのパフォーマンスやプロジェクト管理に関する改良が加えられました。本稿では、その改良点を詳しく紹介します。本稿により、SmartSpiceのVerilog-Aパーサをより深く理解し、活用できるようになります。

ファイル管理の改良

SmartSpiceのVerilog-Aパーサは、ソース・コードを処理した後、複数のファイルを生成します。以前は、生成ファイルがプロジェクト・ファイルと同じ場所に保存されていたため、入力ファイルと出力ファイルが混在し、プロジェクト管理上不都合な場合があります。この点を解決するため、SmartSpiceのVerilog-Aパーサが改良されました。改良後のVerilog-Aパーサは、プロジェクト・ファイル・ディレクトリの配下にディレクトリ階層を作成し、すべての結果ファイルをこのディレクトリに書き出します。ディレクトリ階層は、図1のように作成されます。

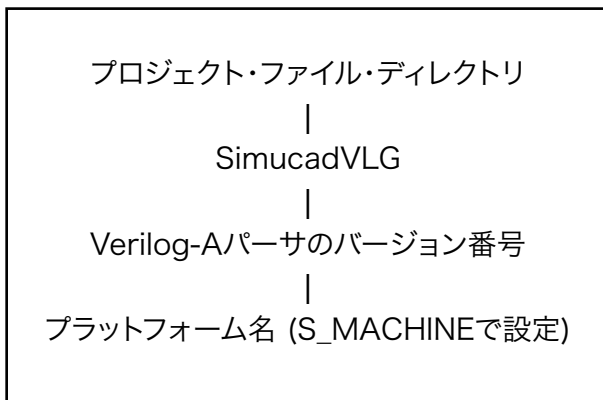


図1: ディレクトリ階層のパターン

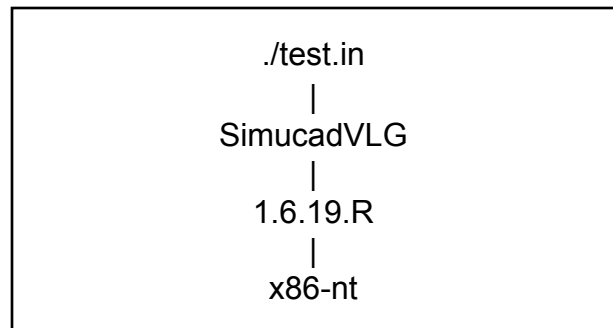


図2: ディレクトリの例

たとえば、プロジェクト・ファイルが ./test.in、Verilog-Aパーサのバージョン番号が 1.6.19.R、プラットフォーム名が x86-nt の場合、Verilog-Aパーサは、図2のようなディレクトリ階層を作成します。このディレクトリ階層では、入力ファイルと出力ファイルが別々であるため、Verilog-Aパーサによって生成された結果ファイルを容易に管理できます。

Verilog-Aパーサ処理結果の再利用

SmartSpiceにVerilog-Aコードのファイルが読み込まれると (SCIモード以外の場合)、Verilog-Aパーサがファイルを処理し、SmartSpiceのシミュレーションに使用される共有ライブラリを生成します。この共有ライブラリの生成には、長い時間がかかる場合があります。Verilog-Aファイルが変更されていない場合、シミュレーションのたびにこの処理を行う必要はなく、時間の無駄になります。そこで、SmartSpiceが改良され、共有ライブラリを再利用できるようになりました。Verilog-Aソース・ファイルとその従属ファイルが変更されていない場合、過去にそのファイルから生成された共有ライブラリを、Verilog-Aパーサで前処理することなく再利用できます。したがって、Verilog-AファイルをVerilog-Aパーサで一度処理しておけば、その処理結果をSmartSpiceで何度も再利用でき、処理時間を短縮できます。

SmartSpiceの出力メッセージにより、ファイルがVerilog-Aパーサで処理されたか、

以前の処理結果が再利用されたかは簡単に分かります。たとえば、次のメッセージは、

```
(VERILOGA):Compiling file ..... 'D:\test_case\laplace.va'.
```

ファイル laplace.va が Verilog-A パーサによって処理されたことを示しています。一方、次のメッセージは、

```
(VERILOGA): Using existing model(s) from 'D:\test_case\laplace.va'.
```

以前の処理結果が再利用できたため、ファイル laplace.va を Verilog-A パーサで処理する必要がなかったことを示しています。

Verilog-A パーサの単独実行

以前の Verilog-A パーサは、独立したプログラムではなく、SmartSpice によってのみ呼び出し可能な、単なるモジュールでした。また、Verilog-A をバージョンアップするたびに SmartSpice 本体の再インストールが必要でした。現在の SmartSpice では、Verilog-A パーサが独立したプログラムとなり、SmartSpice から呼び出す以外に、単独でも実行できます。その結果、SmartSpice を使用せずに、パーサだけを実行して Verilog-A ファイルを処理できるようになりました。SmartSpice は、パーサの単独実行により生成された共有ライブラリを、シミュレーションで直接使用できます。また、Verilog-A パーサをバージョンアップする際も、SmartSpice 本体に変更を加えることなく、パーサのみをインストールできるようになり、メンテナンス性が向上しました。さらに、単独実行の Verilog-A パーサは、1 つのコマンドで複数のファイルを処理できます。たとえば、次のシェル・コマンドを使用すると、

```
> veriloga -l *.va -vcc
```

現在のワーク・ディレクトリのすべての Verilog-A ファイルを処理でき、パワフルかつシンプルな処理が可能です。

コマンドラインから Verilog-A パーサを実行する際に、次の特記事項があります。まず、オプションの "-l" は常に設定することを推奨します。Verilog-A パーサが生成するログ・ファイルには、有用な情報、特にエラーのデバッグ作業に役立つ情報が含まれるためです。次に、Windows において環境変数 "LIBVLG_PATH" を設定していない場合、オプション "-libvlgpath" を使用して Windows の C コンパイラが libVLG ライブラリを見つけるためのパスを設定する必要があります。さらに、オプション "-f" は、すべてのメッセージにおいて、入力ファイルの名前の代わりに表示されるファイル名を指定するために使用できます。たとえば、次のコマンドを使用すると、

```
>veriloga -l .\test\example1\example.va -f example.va -cc
```

すべてのメッセージにおいて、.\test\example1\example.va の代わりに example.va が使用されるようになります。このオプションを使用することで、メッセージを短く読みやすくなります。ただし、コマンドラインに複数の入力ファイルがある場合は、このコマンドを使用できません。

SmartSpice と同様に、Verilog-A パーサでも、コマンドラインで "-V" オプションを使用することで、バージョンを指定して起動できます。たとえば、次のシェル・コマンドを使用すると、

```
>veriloga -V 1.6.17.R *.va -vcc
```

Verilog-A パーサのバージョン 1.6.17.R を使用してファイルが処理されます。指定されたバージョンがインストールされていない場合、次のメッセージが表示されます。

```
"Command line -V "1.6.17.R" not found.
Exiting
```

```
Available versions for this platform
are: 1.6.15.R and 1.6.13.R.
```

Verilog-A ファイルの処理を SmartSpice から実行した場合で、SmartSpice ModelLib コンフィギュレーション・ファイルで指定された libVLG ライブラリのバージョンが、インストールされている Verilog-A パーサのバージョンと一致しないとき、このメッセージは SmartSpice の出力ウィンドウにも表示されます。バージョンを一致させることは重要です。SmartSpice が読み込んだ libVLG ライブラリのバージョンと、Verilog-A パーサのバージョンが一致することは必要条件です。libVLG ライブラリのバージョンは、ModelLib コンフィギュレーション・ファイルに指定されています。通常は、SmartSpice が正しいバージョンの Verilog-A パーサを必ず使用するため問題になりませんが、あらかじめ Verilog-A パーサを単独実行して Verilog-A ファイルを処理し、後でその生成結果を SmartSpice で直接再利用する場合、前処理に使用した Verilog-A パーサのバージョンが、SmartSpice が使用する libVLG ライブラリのバージョンと一致するかどうか、ユーザが確認する必要があります。バージョンが一致しない場合、SmartSpice は正しいバージョンのパーサを呼び出して Verilog-A ファイルを処理するため、Verilog-A パーサ処理結果の再利用のメリットが失われてしまいます。

まとめ

本稿では、SmartSpice の Verilog-A パーサに最近追加された改良点を紹介しました。これらの改良により、プロジェクト管理が容易となり、シミュレーション・パフォーマンスが向上し、フレキシブルな使用が可能になることが分かりました。本稿の内容を理解することで、ツールをさらに活用し、より効率的に作業を仕上げるができるようになります。