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した。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。

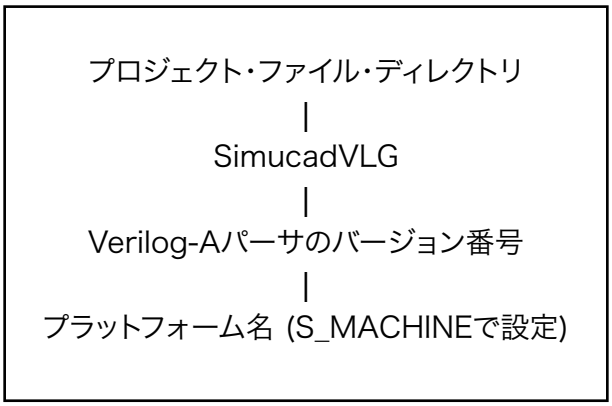

図1: ディレクトリ階層のパターン

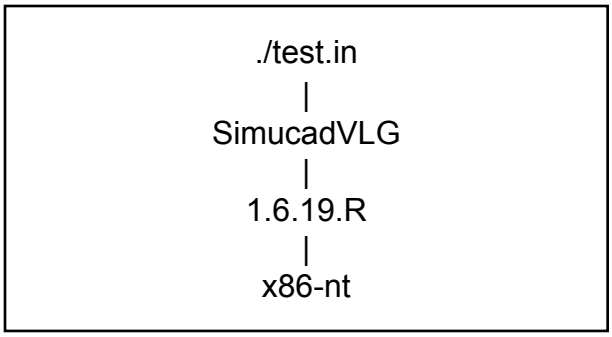

図2: ディレクトリの例

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できます。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ことができるようになります。