

GHDL の使い勝手を試す



竹本 悟

無料で使えるLinux対応VHDLシミュレータ

ここでは、フリー・ソフトウェアとして提供されているVHDLシミュレータ「GHDL」を評価する。VHDL2002対応を目ざして開発が進められているため、現在はベータ版である。VHDL87には対応しており、FPGA設計用や学習用としては十分な機能を備えている。 (編集部)

VHDLが発表された当初、その規格が公開され、多くの大学で使われたことから、フリーのシミュレータが登場するのはそう遠くないと思われていました。事実、いろいろなフリー・シミュレータが登場してきました。しかし、残念なことに、ほとんどがVHDL規格のサブセット版でした。

最初のフルセットVHDLシミュレータは「savant」です。しかし、これは「純粋な」シミュレータでした。GUI (graphical user interface) や波形を表示する機能がなく、ユーザが波形を表示する方法を考えなければならなかったのです。また、構成が複雑で、だれもが簡単に使える代物ではありませんでした。

FreeHDLというプロジェクトがあります。ここではLinux版のVHDLシミュレータの開発が行われています。仕様書は存在していますが、まだ実用に耐えるレベルに到達していません。

このような中、突然フランスで登場したのが「GHDL」です。

GHDLの概要

GHDLはTristan Gingold氏が開発したVHDLシミュレータです。GNUのGPLに基づいて配付されています。したがって、基本的には無償、無保証です。

最初のバージョンが登場したのは2002年です。0.1から始まったバージョンは、本稿執筆時点で0.20になりました。

まだベータ版という位置づけで、いくつかの構文チェックに不備な点が残っているようです。しかし、ほぼフルセットに近い機能を持っています。

● VHDL87に対応

GHDLは、VHDL87(IEEE 1076-1987)に対応しています。VHDL2002の仕様を満足させるのがバージョン1.0の目標だそうですから、まだまだアップデートは続きそうです。履歴を見ると、ほぼ1~2ヵ月間隔でバージョンアップが行われています。

添付の標準ライブラリも商用のライブラリと同じものが備わっています。とくに算術演算ライブラリは古いものと新しいものを備え、ユーザがオプションで選択できるようになっています。ドキュメントには古い算術演算ライブラリの持つ問題が指摘されており、使用上の注意も書いてあります。

VITAL(VHDL Initiative toward ASIC Libraries)のサポートもあり、sdfのバックアノテーションの機能も含まれています。

GUIは含まれていません。そのため、シミュレーション結果を波形で観測する際には、波形表示ツールが必要になります。

● 実行可能コードを生成するコンパイラ

多くのHDLシミュレータと違い、GHDLはコンパイラです。VHDLのソース・コードから実行可能なバイナリ・コードを作ります。

VHDLは米国国防総省のお声掛けりとあって、その仕様はPascalやAdaによく似ています。当然PascalやAdaを使ってコンパイラを作れば、比較的簡単に実現できそうな

GHDLのインストール

COLUMN

● バイナリ・ファイルを使う場合

まず、バイナリ・ファイルをダウンロードし、解凍します。解凍されたディレクトリの中にバイナリの tar ファイルがあります。それを root で / の位置で解凍すると自動的に /usr/local の下に必要なファイルがインストールされます。

```
cd /
su
tar xvf ghdl-0.20-i686-pc-linux.tar
```

後はパスが /usr/local/bin に切つてあれば、すぐに GHDL を起動できます。

● ソース・ファイルを使う場合

ソース・ファイルからのインストールはちょっと複雑です。

まず、Ada のコンパイラである GNAT をインストールします。次に GCC のソース・コードを入手します。GHDL のバージョンによって、要求される GCC のバージョンが異なるので注意が必要です。

GCC のソース・コードの中に GHDL のソース・コードを組み込みます。そして --enable-languages=vhdl のオプションを付けて configure を実行します。

```
make CFLAGS="-O"
```

を実行させてコンパイルし、

```
make install
```

で GHDL コンパイラがインストールされます。

ここでは簡単にまとめましたが、実際にはけっこう複雑な作業になるため、できるだけバイナリを使うほうがよいでしょう。

気がします。GHDL は Ada を使って VHDL のコンパイラを実現しています。そして、オブジェクト・コードを実行することでシミュレーションを行っています。

● Linux のみのサポートだが Windows 対応にも期待

GHDL は、GHDL の Web サイト (<http://ghdl.free.fr/>) から入手できます。Linux 版のソース・コードとバイナリ・コードが提供されています。

ソース・コードがあるので、ほかの OS への移植も可能でしょうが、開発者によるサポートの予定はないようです。ただし、IVI のプロジェクトでは GHDL のサポートを表明しています。すでに MacOS への移植も試みられており、Solaris 版や Windows 版も期待されています。このプロジェクトでは、Verilog HDL シミュレータ「icarus」もサポートしているため、同じ環境でフリーの Verilog HDL シミュレータと VHDL シミュレータが動作することになります。

● バイナリを使ってインストール

筆者はソース・コードが要求する GCC のバージョンが合致しないのでバイナリを使用しました。バイナリを使うとインストールが非常に簡単です(上掲のコラム「GHDL のインストール」を参照)。

GHDL をソース・コードからコンパイルすることもできますが、非常にわかりにくい作業です。GCC と Ada を複合して作られているようで、GCC のほかに Ada のコンパ

イルである GNAT が必要です。コンパイル作業にある程度の経験がないと難しそうです。

なお、GHDL の G はマニュアルによると、特別な意味を持たないそうです。

GHDL を使ってみる

GHDL そのものの構造は非常にシンプルです。C コンパイラと同じように、VHDL ソース・コードを実行可能なオブジェクト・コードに変換します。シミュレーションは、そのオブジェクトを実行することによって実現されます。

ほかのプログラム言語と同じように、GHDL は VHDL ソース・コードをコンパイルし、リンクして実行モジュールを作り出します。VHDL の世界では、これらを「アナライズ」、「エラボレート」と呼んでおり、GHDL のコマンドもその呼び名で説明されています。

VHDL ソース・コードから実行モジュールを作り出す方法は、ソース・ファイルが一つの場合と複数の場合で異なります。

● ファイルが一つだけの場合

実際の設計では、ソース・ファイルが一つで済むケースなどはまれですが、GHDL の動作を理解するうえでは役に立ちます。

まず、アナライズを行います(図 1 の)。評価には、