

# VHDL と Verilog HDL の基礎概念と文法

## 第2章

本章では、ハードウェア記述言語 (HDL) の基本概念について解説する。そして、よく使用される VHDL と Verilog HDL の二つの HDL を対比しながら基本文法を説明する。ちょっとした違いを発見しながら読み進めると面白いだろう。

また、ソフトウェアで使用する C 言語や Java、コミュニケーションで使用する英語についても、正しく使用するために文法の知識は欠かせない。同じように HDL で設計する際も文法の知識が必要である。ここで、基本文法をしっかり押さえよう。(編集部)

三好 健文

### 1. ハードウェア記述言語の基本概念

プログラミング言語に多くの種類があるように、ハードウェア記述言語 (HDL) にもさまざまな種類があります。その中でもよく利用されるのが、VHDL と Verilog HDL です (図 1)。

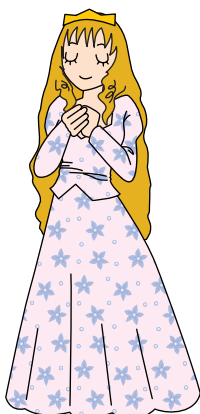
VHDL と Verilog HDL は、似たような概念に基づいた言語です。ただし、似たような概念でもそれぞれの言語で使用する言葉が違うので注意が必要です。両方の言語に共通する概念と、言語の特徴について説明します。

#### ● 構造の基本 — エンティティ/モジュール

どの言語にも基本的な構造があります。たとえば、C 言語では関数、Java ではクラスなどです。HDL では、与えられた入力に対して出力を生成するブロックが基本的な単位です (図 2)。このブロックを VHDL ではエンティティ、Verilog HDL ではモジュールと呼びます。



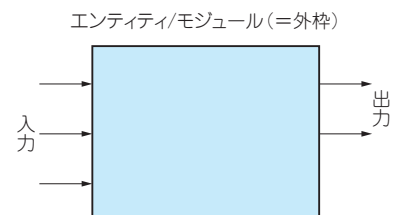
まじめなVHDLさん



お姫さまのVerilog HDLさん

◀ 図 1  
VHDL と Verilog HDL のイメージ  
VHDL はまじめなキャリア・ウーマン、Verilog HDL はドレスを着たお姫さまのイメージがあるといわれる。

▶ 図 2  
入力に対して出力を生成するブロックが基本単位



通常のプログラミング言語と HDL の大きな違いは、エンティティ/モジュールは、最初から最後まで与えられた入力に対する出力を生成し続けるということです。

C 言語などで関数を呼び出す場合、main プログラムからその関数内へ処理が移ります [図 3 (a)]。処理を終えると戻り値を呼び出し元に返し、main プログラムが再び動き始めます。つまり、main プログラムは、呼び出した関数の処理が完了するまで待たされます。これは、プログラム・カウンタが、プログラムを順々に呼び出して実行するからです。

一方、HDL で記述されたエンティティ/モジュールには、プログラム・カウンタのようなものではありません [図 3 (b)]。どの回路も同時に動作します。したがって、特定の入力を与えると出力を返すというよりも、入力されているデータに対して出力するデータを作り続けているというイメージになります。

#### ● 2 種類の基本処理方法 — 同時処理文と順次処理文

ハードウェア・モジュールでは、さまざまな処理が独立して動作します。そのため、同時並行的な処理を記述する必要があります。さらに、プログラミングを容易にするためには、同時並行的な処理に加え、条件分岐のような依存関係のある処理の記述が望まれます。これらの要求を満た