



FLASH内蔵エントリ版で チップ内電子工作 私のスペシャルIC製作! FPGA MAX 10の研究

⑧ Verilog HDLによるRTL記述入門
論理設計の道具を自分のものにしよう

圓山 宗智 Munetomo Maruyama

今月号ではハードウェア記述言語の代表格、Verilog HDLによるRTL記述方法の基礎を説明します。本稿の内容が理解できれば、ほとんどの論理回路は問題なく記述できると思います。

ハードウェア記述言語 「Verilog HDL」とは

● Verilog HDLとは

論理回路を抽象的に記述するためのハードウェア記述言語としては、Verilog HDLとVHDLが有名です。歴史的には、まずVHDLが米国国防総省によって開発されました。VHDLは記述量が多いのですが、明確に機能仕様を定義できる特徴があります。その後、論理シミュレータと一体化した言語としてVerilogが開発されました。Verilogは記述量がVHDLよりも少なく手軽にコーディングできますが、論理シミュレータ側の視点で定義された言語であり、物理的な論理回路を確実に記述するには多少の注意が必要です。

Verilogは歴史的に複数のバージョンが定義されてきました。今月号ではVerilog 1995とVerilog 2001を説明します。Verilog 2001はVerilog 1995の改良版です。現在では、さらに改良され多くの機能が加えられたSystem Verilogも普及しています。System Verilogを使った機能検証方法については、本連載で改めて触れる予定です。

● Verilog 1995とVerilog 2001

基本的に、アルテラ社が提供する開発環境Quartus PrimeやModelSim-AlteraではVerilog 1995もVerilog 2001も共に使えます。もちろんSystem Verilogもサポートしています。

本稿で解説するVerilog 1995とVerilog 2001の間は、さほど大きな差はありませんが、Verilog 2001のほうが組み合わせ回路の記述方法が楽になる利点があります。Verilog 1995型の記述とVerilog 2001型の記述は、一つのRTL記述の中で混在可能です。Verilog 1995

とVerilog 2001の記述方法で意識すべき差異は、この後に説明します。

論理機能のモジュール構造記述

● モジュール構造と階層構造の記述

論理機能はモジュールという単位で記述します。リスト1(a)のように、論理機能はmodule文とendmodule文で挟み、入出力信号と内部機能を記述します。モジュールは一般的に階層構造をもちます。例えば、一つのLSIの中にCPUやDMACが内蔵されている場合、LSIという階層の下にCPUとDMACがあります。さらにCPUの下にも、例えばDATAPATHとCONTROLという階層が置かれるのが通常でしょう。

● インスタンス化

あるモジュールの下の階層として、別のモジュールを配置する場合の記述方法をリスト1(b)に示します。ここで重要な概念があります。一つのモジュールは、あちこちで使いまわされることが一般的です。例えば、デュアル・コアをもつLSIの場合は同じCPUが2個あるでしょうし、2入力NANDゲートをモジュールとして定義すれば、LSI内で同じものが大量に使いまわされます。

このとき、ある階層内に置く個々の下位階層モジュールを区別するため、その階層内で固有となるインスタンス名を付けて、その下位階層モジュールを置きます。これをモジュールのインスタンス化(instantiation)といいます。インスタンス化というのは「実体化・具体化」という意味です。module文に記述するモジュール名は、定義名にすぎないことに注意してください。原則として各モジュールは必ずその上位階層でインスタンス化されます。

それではLSIやFPGAの外部入出力信号が置かれる階層(LSIやFPGAにとっての最上位階層)はどうなるのでしょうか？ これもシミュレーション時にはインスタンス化されます。

後述しますが、LSIやFPGA全体を論理シミュレー