

# 誰でも半導体チップを作れる世界へ

秋田 純一

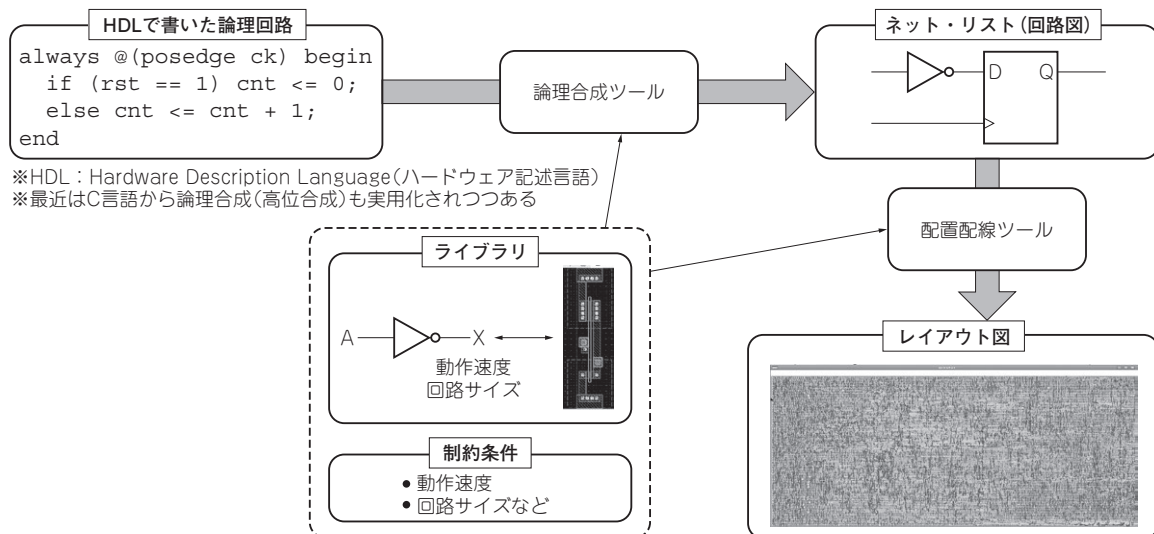


図1 半導体チップを作る大まかな流れ

オープンソースといえば、今ではLinuxをはじめ、ソフトウェアでは、ごく普通に使われています。オープンソースとは「中身や設計図が公開されている」という意味ですから、ソフトウェアに限った話ではなく、ハードウェアでも可能な概念です。実際に最近は使われる場面も増えてきました。

オープンソースの利点は、何より「全て自分で使える」ということです。つまり、情報提供側が明日から「もう公開しない、もう更新しない」と言われても、続きは自分で、できる訳です。

ここでは、オープンソースのプロセッサ・アーキテクチャである「RISC-V<sup>リストアップ</sup>」を、オープンソースな設計ツール・ライブラリを使って、シリコン・チップの設計データまで変換する手順を紹介します。誰でもCPUを作れる時代になるかもしれません。

## 半導体チップ作りの流れ

### ● プログラムのように論理回路を記述

デジタル回路の設計方法の1つは、「論理回路」の

教科書にあるような、ブール代数、カルノー図、状態遷移表などを駆使するものです。しかし、プロセッサのような大規模な回路をこの方法で設計するのは相当大変です。さらに、出来上がった回路の検証も、それ以上に大変です。そこで、HDL (Hardware Description Language) と呼ばれる言語でプログラムのように論理回路を記述し、それをコンパイラが実行バイナリを作るような調子で、実際の電子回路まで変換する設計方法が、近年はよく使われます。HDLとしてはVerilog HDLやVHDLが有名です。さらに、よりプログラミング言語に近いCなどの言語で回路の動作を記述し、それを電子回路にまで変換する高位合成と呼ばれる手順も、最近は実用レベルになりつつあります。

### ● 論理回路を電子回路に変換する

HDLで記述した論理回路は、図1のような流れで電子回路に変換されます。まず、HDLで記述された論理回路の動作を、論理回路の最小単位である論理ゲートやフリップフロップで構成された論理回路で実現されるように変換します(論理合成)。それぞれの