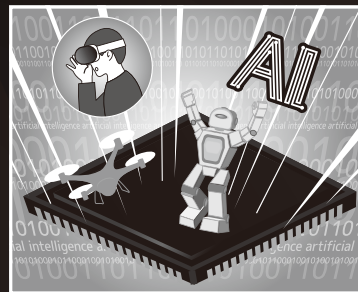


2nm時代到来!

AI, RISC-V, ベクトル・プロセッサ…

新時代の プロセッサ開発技術



第5回 回路の動作を数学で確かめるフォーマル検証

草刈 憲

● 連載概要

本連載では電子システムのアプリケーションを支える屋台骨のプロセッサについて、市場と技術の最新動向を解説します。半導体の中でも、その最先端を走るプロセッサを支える技術を知ることにより、半導体、ひいては電子システムのトレンドを把握することができ、ソフトウェアを開発する立場でも、より多面的にハードウェアからソフトウェアまで含めたシステムを捉えることができるようになると思います。

● 今回のテーマ

今回はプロセッサのフォーマル検証に焦点を当てます。プロセッサの性能を上げるための工夫（例えばストールを解消するためのキャッシュ、分岐予測など）を増やしていくと、設計は複雑化し、検証も複雑化します。結果として検証期間が長期化しがちです。

せっかく魅力のあるプロセッサを設計しても、検証が終わらなければ世に出すことができません。組み込みエンジニア目線でも、ハードウェアの検証が完全ではないタイミングで同時並行にソフトウェアの検証を行うというケースにおいては、書いたソフトウェア・コードが悪いのか、ハードウェアの不具合なのかの切り分けも難しく、デバッグが非常に困難になります。そうした事態は避けなければなりません。

高い性能を求めつつも、必要な検証を如何に限られた時間内で完遂できるかが検証技術者の腕の見せ所となります。この難題の解決策として、網羅的な検証を行うことができる、フォーマル検証に注目します。

● フォーマル検証…回路の動作を数学的に検証

フォーマル検証は、HDL (Hardware Description Language : ハードウェア記述言語) で記述された回路を数式化し、その回路に期待する動きを定義したアサーション (SVA : System Verilog Assertion で記述) を用いて検証する手法です。テスト・パターンは不要で、アサーションのみで検証できます。ハードウェアの内部状態の組み合わせのバグは、人手によるテスト・パターンでの検出が難しいのですが、フォーマル

検証では数学的に証明を行うので、ここでアサーションが証明成功となればバグがないことを証明できます。ツール側が全てのパターンを網羅してくれるからです。

一方、シミュレーション検証の場合はテスト・エンジニアがテスト・パターンを作って検証を進めていきます。コード・カバレッジなどの手法を用いて検証の網羅性を測るのが一般的ですが、パターンを作るのはツールではなく人間の責任となり、完全に網羅できたことを示すのは難しく、膨大な工数がかかります。

このような特徴から、フォーマル検証は面で検証できると言われます（これに対してシミュレーション検証は線の検証の集合というイメージ）。

フォーマル検証の強みを 最短経路問題でイメージ

網羅的に検証できるとはどういうことか、中学受験などで目にする、最短経路の問題を使って説明します。

● 例にする最短経路問題

図1のようにAからBまで最短経路を通っていく道順があるとします。経路は全部で35通りあります。それぞれの経路にRoute0～Route34という番号を付けます。図1をベースに、経路の番号を入力するとその経路を通るようにスイッチが制御される回路（のようなもの）を考えてみます。この回路の要求仕様は、AからBまで7サイクルで到達する（レイテンシ=7）とします。図2は、Route6を選択した際の例となります。コントローラが正しく動けばAからBまでパル

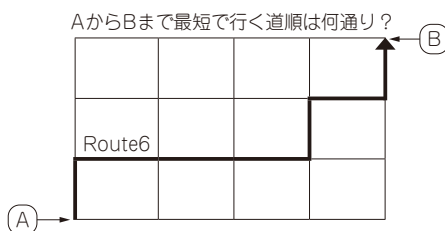


図1 AからBまでの最短経路を見つける問題