

ワイヤレスLAN用モデムICの デジタル設計

「実行可能な仕様書」を作成して再設計をなくす

ルーウィ ヴアレニヤ

ここでは、IEEE 802.11g ワイヤレスLAN用モデムICの開発プロジェクトを想定しながら、ベースバンド回路(デジタル回路)の設計の手順を紹介する。まず、「実行可能な仕様書(ゴールデン・テストベンチ)」を作成し、これに基づいてRTLコードを作成する。実行可能な仕様書があれば、設計の早い段階で、複数のモジュールによる接続検証を行える。また、論理合成によって生成したゲート・レベル・ネットリストを実行可能な仕様書に挿入して、ボトムアップ検証を行う。(編集部)

前章では、直交変調器の補正システムの構成と補正アルゴリズムを紹介しました。ここではデジタル部のRTL(register transfer level)コードを作成する方法について解説します。

●システム設計とデジタル設計の間には溝がある

従来のRTL設計は、多くの場合、図1(a)のようになっています。システム設計とデジタル設計が分離しており、「機能仕様書」がその間の唯一のインターフェースです。機能仕様書はたいてい未完成のドキュメントで、あいまいな部分が多く、細部まで記述されていません。

仕様書を作成するシステム設計者は、読む人がある程度の知識を持っていることを前提として書いています。読むほうのデジタル設計者がその知識のレベルを満たしていないと、誤解が生じやすくなります。

また、十分な知識があっても、仕様書のあいまいさによって、解釈の違いが発生します。デジタル設計者は自作のブロックを検証するとき、自分でテストベンチを作り、その誤解や勘違いをそのままテストベンチに組み込んでしまいます。結果的に、そういったミスは、ほかのグループのモジュールと組み合わせて検証を行うまで発見されませ

ん。場合によっては、設計のやり直しが発生し、開発スケジュールに影響を与えてしまいます。

●「実行可能な仕様書」で溝を埋める

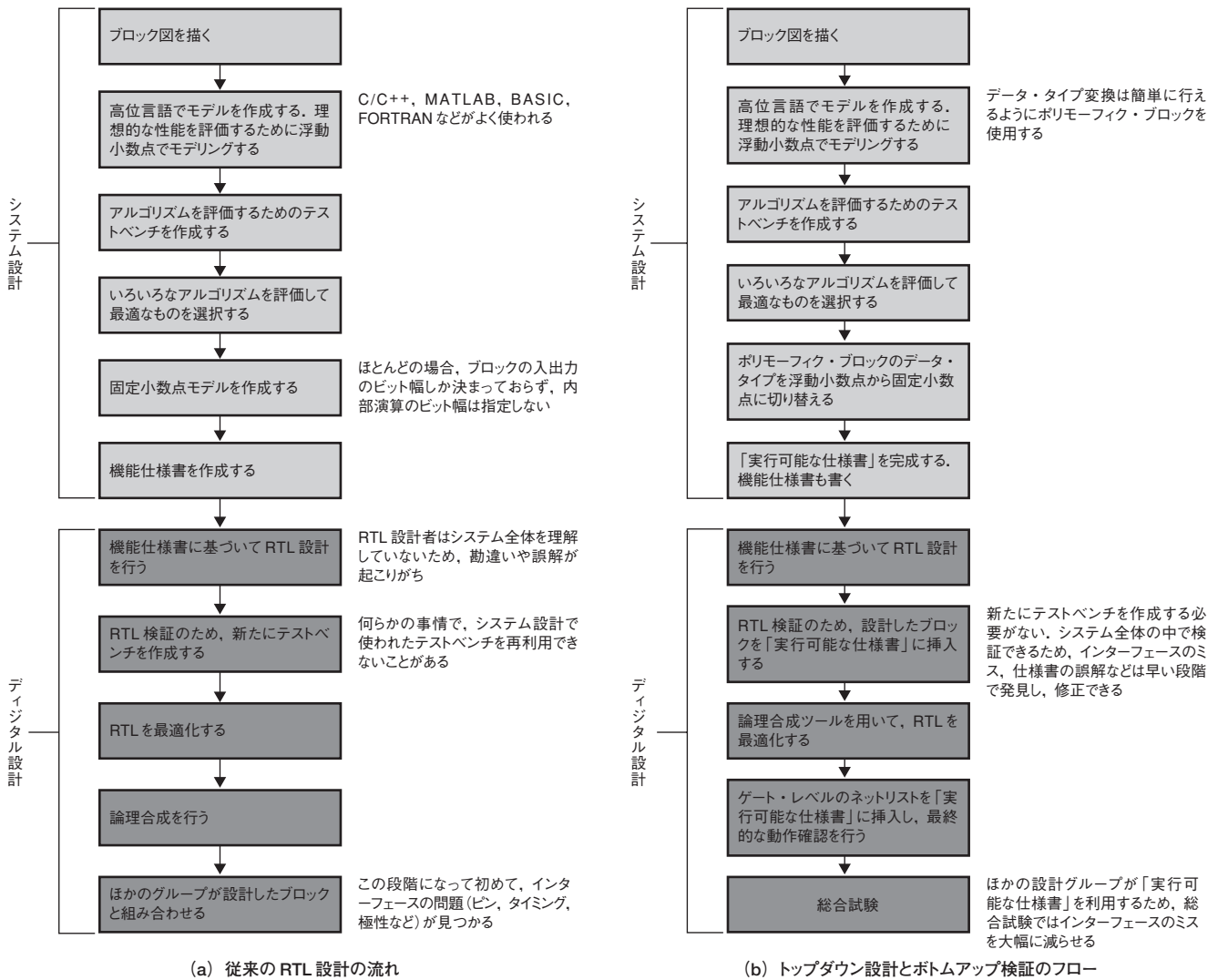
その弱点を補うため、「トップダウン設計・ボトムアップ検証」の手法が提案されました¹⁾。この手法のフローを図1(b)に示します。この手法は、「バーチャル・プロトタイプ」と呼ばれることもあります。簡単に説明すると、実物を製造する前に仮想的なビヘイビア・モデルを作ることです。例えば、デジタル設計グループはRTLのVerilog HDLコードやVHDLコードを、ソフトウェア開発グループはCコードを、アナログ・RF設計グループはSPICE ネットリストやVerilog-AMSコード、VHDL-AMSコードを提供して、一つの環境でコシミュレーション(協調シミュレーション)^{注1)}を実施し、機能的な検証を行います。各ブロックの動作を仮想システムの中で確認できるので、仕様に対する誤解やインターフェースの誤りのようなミスはすぐに見つかります。

この手法の特徴は、以下のとおりです。

1) 実行可能な仕様書を作成する

「実行可能な仕様書(executable specification)」は、ゴールデン・テストベンチとも呼ばれます。抽象度の高い設

注1: コシミュレーション(co-simulation)は、複数の異なるシミュレータを接続し、それぞれのモデルを協調動作させながら動作を模擬する技術である。例えば、複数の言語が混在するシミュレーションを実現できる。ほとんどのコシミュレーションでは、プロセス間通信(IPC: inter-process communication)が利用されている。プロセス間通信では二つのプログラムが同時に走っており、OS(Windows, Linuxなど)経由で相互に通信を行う。そのため、処理のオーバーヘッドが大きいという問題がある。一方、本稿で使用している信号処理システム開発環境「SPW」の場合、一つのプログラムしか走っていない。そのため、一般にプロセス間通信を利用する場合よりもシミュレーション速度が速くなる。



〔図1〕システム・レベルとRTLの開発

従来の設計手法では、各設計グループが独立に作業を進める。また、紙に書かれた「機能仕様書」が各設計グループの間の唯一のインターフェースである。この方法では、最後にモジュールを組み合わせる段階で仕様の誤解やインターフェースのミスが見つかる。一方、トップダウン設計とボトムアップ検証を組み合わせた手法では、「実行可能な仕様書」を利用することにより、モジュールの接続検証を早い段階で実現できる。そのため、前述のような問題を解消できる。

計言語(C/C++やSystemCなど)で書かれているコンピュータ・プログラムです。設計するブロックのビヘイビア・モデルを利用している(つまり、詳細な設計情報が含まれていない)ため、シミュレーションの実行速度は速くなります。すべての設計グループは、このゴールデン・テストベンチを利用して自分が作ったブロックの動作を検証し

ます。

2) 実行可能な仕様書をリファインする^{注2}

各ブロックの詳細設計の進捗に合わせて、ビヘイビア・モデルを更新していきます。詳細化したブロックをゴールデン・テストベンチにどんどん組み込んでいくと、シミュレーションの実行速度は遅くなります。そこで、できるだけ検証しているブロックのみを詳細なレベルにして、ほかのブロックにはビヘイビア・モデルを利用します。

ゴールデン・テストベンチは、米国CoWare社の「SPW (Signal Processing Workstation)」で作成しました(図2) SPWは信号処理システムを開発するためのEDAツールの

注2：リファインとは「洗練」、「磨きをかける」という意味。ここでは詳細設計が必要とされる信号遅延や入出力端子などをビヘイビア・モデルに反映させ、そのモデルの設計抽象度を下げていくことを指す。例えば、目標の動作周波数に間に合わせるために、パイプライン構成を導入したり、ラッチを挿入するなどの作業である。