

システム・レベル言語 によるLSI開発

枝 隆司



ここでは、LSI 開発者向けに、SpecC 言語をハードウェア記述言語として利用した場合の記述例を紹介し、SpecC 技術の設計階層はシステム設計のために考案されたものであるため、既存のLSIの設計階層とは一致しません。そこで、ここでは、従来のLSI設計のスタイルに合わせて、ゲート・レベル、RTL(register transfer level)、ビヘイビア・レベルなどの回路動作を表現したSpecC記述を紹介し、高位のモデルを低位のモデルに置き換える作業は人手でも行えますし、自動化ツール(ビヘイビア合成ツールなど)を使用して自動合成することも可能です。(編集部)

SpecC 言語は、仕様策定からシステム設計、インプリメンテーション設計(ソフトウェア設計、ハードウェア設計)へと続くフローを一貫してサポートする言語として登場しました。

設計フローの一貫性は、生産性の向上に寄与します。SpecC 言語の開発にあたっては、仕様設計やシステム記述に適した言語にするために、言語として必要な機能や概念を調査したといいます。さまざまなプログラミング原理を取り入れることになりましたが、言語に必要な特性を並列性、状態遷移、構造/ビヘイビア階層、例外処理、タイミング、通信、同期、データ処理、再利用などに絞り込みました。もちろんC言語が備えている構文は、すべてサポートしています。

C言語(または、その拡張)を使用して設計する最大の利点は、世の中に豊富な設計資産が存在している点です。また、とにかく「動くプロトタイプ」が早くほしいのであれば、HDLよりもC言語が向いていると思います。さらに、忘れてはならない点は、C言語でプログラムできる設計者(ソフトウェア開発者を含む)の数が多ということです。

LSI設計者にとって気になる点は、C言語をベースにしたSpecC言語を使って、回路の動作を正確に表現できるのかという点だと思います。LSI設計では回路図やハードウェア記述言語(HDL)による設計が一般的です。HDLを使用してい

る設計者であっても、具体的なコンポーネント間の接続や動作を頭に思い描いて設計しているのが現状だと思います。

1 SpecCとLSI設計フロー

図1は、SpecC言語を用いた場合の設計フローです。システム全体をSpecC言語で記述して、少しずつ詳細な記述に変換していきます。システムは途中でハードウェアとソフトウェアに分割され、それぞれの設計作業が進んでいきます。最後の段階で、分割されたハードウェアとソフトウェアが統合され、システムが完成します。

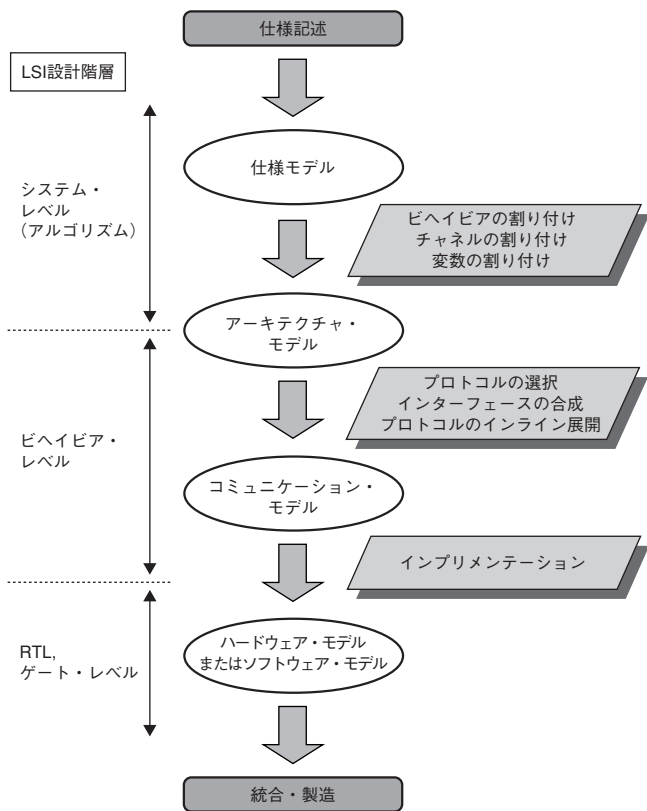
SpecC技術では、システムを表現するために四つのモデルを定義しています。もともとはシステム記述を前提にしたフローですが、LSI開発のフローとして考えることも可能です。この章では図1のSpecC設計フローをLSI設計にあてはめた場合を例に説明していきます。

●4段階でモデルをリファインするSpecC設計フロー

SpecCのモデルには、図1のように、仕様モデル、アーキテクチャ・モデル、コミュニケーション・モデル、ハードウェア(ソフトウェア)モデルがあります。仕様モデルがもっとも抽象度が高く、ハードウェア・モデルは実際に製造可能なモデルです。上位のモデルから下位のモデルへと、手作業で置き換える(リファインする)こともできますし、自動化ツールを使用して下位のモデルを自動合成することもあります。

(1)仕様モデル

仕様モデルは、ターゲットとなるLSIの機能をモデル化した仕様記述で、コンピュータで実行することができます。通常、仕様は紙に書いたドキュメントで提供されることが多いと思いますが、仕様モデルは、いわば仕様に忠実に動作するプログラムと言えるでしょう。実装方法の詳細は記述されていませんが、性能や消費電力、回路規模などの制約を記述す



【図1】 SpecC設計フロー

SpecC 技術では、システムを表現するために四つのモデルが定義されている。それらの階層は、既存のLSI 設計階層とは必ずしも一致していない。

ることができます。仕様記述では、ライブラリに登録されている既存のC コードを組み込むこともあります。

(2) アーキテクチャ・モデル

仕様記述からターゲット・アーキテクチャへのマッピングは、大きく分けて「アーキテクチャの探索」と「コミュニケーション合成」の二つの作業があります。まずアーキテクチャの探索について説明します。アーキテクチャの探索では、アロケーション、割り付け(パーティショニング)、スケジューリングの作業を行います。

アロケーションでは、専用のプロセッサやメモリ、それらを接続するバスなどの種類や数量を決定します。割り付けでは、LSI を構成している個々の機能(ビヘイビア)をライブラリに登録されている機能ブロックに割り付けたり、専用プロセッサで実行するかなどを決定します。また、コミュニケーション・チャンネルを抽象度の高いバス・プロトコルなどに割り当てたり、変数をレジスタやメモリに置き換えたりします。スケジューリングでは、実行の順番を決定します。より最適なアーキテクチャを探し出す作業なので、何度も繰り返し作業が行われるフェーズです。これらの作業が終わると、仕様モデルはアーキテクチャ・モデルになります。

(3) コミュニケーション・モデル

アーキテクチャ・モデルは、コミュニケーション合成を行うことにより、コミュニケーション・モデルになります。

コミュニケーション合成では、ビヘイビア間で行われている大まかな“コミュニケーション”を具体的な“通信回路”に置き換えます。たとえば、通信プロトコルが割り当てられたり、インターフェース回路が合成されたり、合成可能な通信プロトコルがインライン展開されたりします。

コミュニケーション合成の後、ビヘイビア間はバスやハンドシェイク・プロトコルによってデータのやり取りが行われます。これらのやり取りは、その動作を検証するため、バス・ファンクショナル・モデルになることが多いようです。このモデルをもとに、実装設計が行われます。

(4) ハードウェア・モデル、ソフトウェア・モデル

ここでは、本来、ハードウェアとソフトウェアに分割する作業が行われます。ソフトウェアは純粋なC コードになり、特定のプロセッサ用にコンパイルされます。ハードウェアの部分は合成可能なVHDL や Verilog-HDL になります。各機能ブロックはプロセッサやメモリ、アプリケーションに特化した専用回路などに割り当てられます。これらの作業が終了すると、全体のシステムはクロック・サイクルの精度をもつソフトウェア・モデルとハードウェア・モデルになります。

●ハードウェア記述言語としてのSpecC

SpecC 言語は、仕様決めから実装設計まで、ハードウェア記述に必須である並行実行や階層、状態遷移、データ処理などを表現しやすいように拡張されています。拡張にあたっては、記述からあいまいさを排除し、誤りを防ぐ目的で、さまざまな専用の構文を用意しています。

上述した四つのSpecC モデルと、現在よく知られているデジタルLSI の設計階層は、必ずしも1対1に対応していません。そのため、LSI 設計者がSpecC 言語を利用する場合、最初はとまどうかもしれません(図2)。ここでは、既存の設計スタイルに合わせて、SpecC 記述を紹介していきます。

現在のLSI 設計階層は、以下の四つのレベルからなります。

- システム・レベル
- ビヘイビア・レベル
- RTL(register transfer level)
- ゲート・レベル

SpecC 言語のハードウェア・モデルは、クロック・サイクルの精度をもつRTL とゲート・レベルに相当します。一方、SpecC 言語のアーキテクチャ・モデルとコミュニケーション・モデルは、現在の設計フローでは明確に規定されていない部分です。あえて分類するならば、ビヘイビア・レベルの記述からRTL へ書き換えていく過程の抽象度に対応しています。仕様モデルは、純粋なアルゴリズム設計の段階です(図1)。