

SpecC技術にもとづく 分析的システム・レベル設計概論

石井忠俊



ここではSpecCを利用した組み込み機器やシステムLSIのシステム・レベル設計フローについて解説します。これまでベテラン設計者の頭のなかで行われていた仕様策定やシステム・レベル設計の作業手順を整理し、体系化しました。構造図や記述例を示しながら、一つ一つの設計タスクの作業内容を分析していきます。SpecC技術は、自動化ツールを使う場合のことも考慮されていますが、人手でシステム・レベル設計を行う際にも役に立ちます。(編集部)

組み込み機器や、それに搭載されるシステムLSIを開発する場合、扱うシステムが非常に複雑になるため、トップダウン設計手法やボトムアップ設計手法が採用されています(図1)。

トップダウン設計手法では、まずシステムの機能を抽象的な仕様として記述し、それをブレイクダウンしていくことによって設計作業が進みます。トップダウン設計手法は規模の大きい複雑なシステムの設計に向いています。一方、ボトムアップ設計手法では、すでに存在している電子部品、回路ブロック(マクロセル)、ソフトウェア部品を再利用し、必要なものをつなぎ合わせることによってシステムの機能を実現します。ボトムアップ設計手法は、短期間で、あるいは少ない設計リ

ソースでシステムを開発する場合に向いています。

近年、設計はどんどん複雑になり、回路やプログラムの規模が急激に大きくなっています。その一方で、複雑になったからといって、設計にかけられる期間が伸びてよいわけではありません。また、複雑になった分だけ人手を増やせるわけでもありません。そのため、通常はトップダウンとボトムアップの両方を混ぜ合わせた手法が採用されています。

そこで、今後、新規に組み込み機器やシステムLSIを開発する場合、トップダウンとボトムアップを上手にミックスし、しかも開発者にとって理解しやすく、設計対象にマッチした設計フローを確立することが重要になります。新しい設計フローを考える場合、現在の設計業務の内容を「設計探索」と「合成・コンパイル」からなる設計タスクに分解して考えます。そして、これらの設計タスクを整理し直す形で設計フローを構築します。ここで設計探索とは、仕様を実現するための設計案をいろいろと考え出し、そのうちのどれが一番よいものかを判断して選択することです。合成・コンパイルは、設計データをより詳細な設計データに変換し、最適化する作業です(必ずしも自動化ツールで自動合成するわけではない。人手で変換・最適化することも含めて、ここでは「合成・コンパイル」と呼んでいる)。

このほか、設計フローのなかに設計者の勘や経験、計算などによる設計判断を組み込んでいく必要もあります。

1 システム・レベル設計フローの全体像

本特集の第1章で述べたような問題を解決する手段の一つとして、ここではSpecCの設計フローを紹介します。SpecC技術による組み込みソフトウェアやLSIのシステム・レベル設計手法を図2に示しました。

SpecC設計フローは大きく別けて三つのステップに分割されています。すなわち、仕様設計、機能設計、実装設計です。



〔図1〕 トップダウン設計とボトムアップ設計

トップダウン設計では、まず全体像をイメージし、それを具体化したり、パーツに分解して設計する。ボトムアップ設計では、すでにできあがっているパーツを組み合わせる。

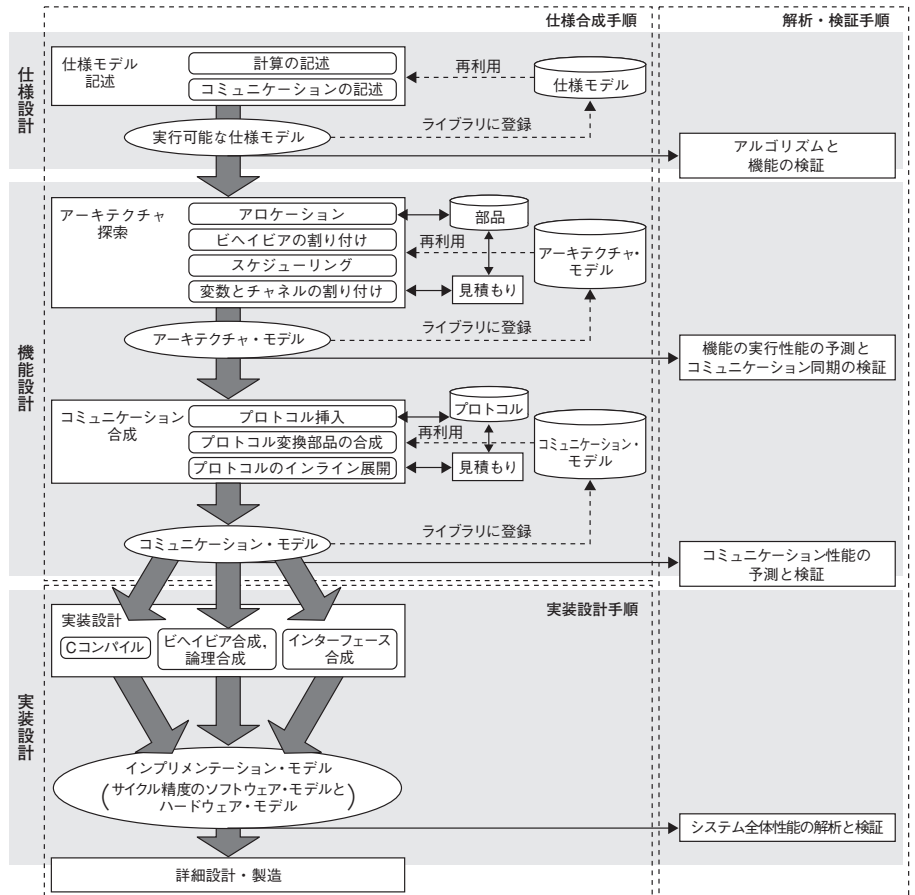
【図2】SpecC 設計フローの詳細

SpecC 設計フローは、「仕様設計」、「機能設計」、「実装設計」の3ステップからなる。最初の二つのステップは、これまで分けてとり上げられることが少なかった。SpecC 技術はおもにこの二つのステップに重点を置いている。

仕様設計では、設計対象である組み込みシステムなどの動作だけを記述する。ここでは製品を設計するうえで必要となるブロック分けやタスク分割、実装に必要な部品などの情報は含まれない。

機能設計は、大きく二つのステップに分かれる。最初のステップは「アーキテクチャ探索」と呼ばれ、設計したい製品にもっともよく合うアーキテクチャを試行錯誤しながら見つけだす。このステップでは、仕様モデルに部品情報やブロック分けの情報が書き足される。仕様設計で作成した動作だけの仕様を分割し、設計制約を満たすように部品やブロックへ割り付け。こうしてできあがった設計データをアーキテクチャ・モデルと呼ぶ。

2番目のステップでは、部品間のコミュニケーションを整理するために「コミュニケーション合成」を行う。部品間やタスク間のコミュニケーションの動作はアーキテクチャによって変化する。その結果を、実際に設計するコミュニケーションのアーキテクチャであるバスやメモリ・コントローラ、グローバル変数、デバイス・ドライバといったものにまとめ上げる。



最初の二つのステップは、現在の設計フローでは区分けして取り上げられることが少なく、読者のみなさんにとっては目新しいものかもしれません。SpecC 技術では、おもにこの二つのステップに重点を置いています。

(1) 仕様設計

SpecC 設計フローは、仕様設計から始まります。このステップでは実行可能な仕様記述を作成します。「実行可能」とは、記述をそのまま動作(シミュレーション)させて、実行状況を見ることができるとい意味です。この仕様には、設計したいシステムの機能のほか、性能、消費電力、使用リソース量 (LSI ならばゲート数やマスクの面積、ソフトウェアならばコード・サイズやデータ・サイズ) などのシステムの設計制約についても記述します。

仕様設計では、設計対象の動作だけを記述します。この段階で、具体的な設計内容を思わせるような情報、たとえば、使用する部品、設計する上で必要なブロックの切り分け、タスク分割などの情報は含めるべきではありません。

(2) 機能設計①：アーキテクチャ探索

次の機能設計は、二つのステップに分かれます。すなわち、アーキテクチャ探索とコミュニケーション合成です。

最初のアーキテクチャ探索では、設計したい製品にもっとも

よく合うアーキテクチャを試行錯誤しながら見つけだします。具体的には、前の工程で作成した仕様記述に部品情報やブロック分けの情報を書き足します。仕様設計で作成した動作だけの仕様記述を設計制約を満たすように部品 (LSI の機能ブロック、ソフトウェア・モジュールに相当) へ分割し、割り付ける作業を実施します。できあがった設計データをアーキテクチャ・モデルと呼びます。

アーキテクチャ探索は、「アロケーション」、「割り付け」、「スケジューリング」の三つの工程からなります。ここでアロケーションとは、システムを構成する部品の型と数を決める設計タスクのことです。たとえば、プロセッサ、DSP、ASIC、バス、リアルタイム OS、リアルタイム OS 上のプロセスなどのシステム部品を使ってシステムを構成します。割り付けとは、システム機能の構成要素(ビヘイビアと呼ぶ)を前の設計タスクでアロケーションした各システム部品へ分割・分配することです。スケジューリングとは、各システム部品内で構成要素の実行順序を決め、逐次実行できるようにすることです。

(3) 機能設計②：コミュニケーション合成

次に部品間のコミュニケーションを整理するために、コミュニケーション合成を実施します。部品間やプロセス間のコミュニケーション動作は、アーキテクチャによって変化しま