

UMLとC言語をベースとしたシステムLSI設計手法

上流設計の開発プロセスを見直し，UMLとC言語をベースとしたシステムLSI設計手法を開発しました。従来の手法と比べて，上流工程の設計期間を約1/3に短縮することが可能です。

概要

当社と株式会社富士通研究所は共同で，UML(Unified Modeling Language)とC言語(C/C++/SystemC)をベースとしたシステムLSI設計手法を開発しました。この新しい設計手法では，システムLSIの仕様設計からRTL設計までの上流設計における開発プロセスを全面的に見直し，設計効率を大幅に改善しています。これにより，従来の設計手法と比べて，上流工程の設計期間を約1/3に短縮することが可能になりました。

開発の背景

近年，システムとシステムLSIは，ますます大規模化・高機能化・複雑化しています。しかし，これに対して設計手法が追いつかず，このままでは入れ物(システムLSI)があっても中身を設計できなくなるという，強い危機感が生まれていました。また，最近の設計障害の多くは，上流工程におけるミスの混入(仕様抜け，あいまいな仕様，仕様の誤解)が原因であるという調査結果(JEITA * SLD研究会 先報告されています。このため，これまでの設計手法の改良方法(トランジスタレベル ゲートレベル RTL ビヘイビアレベルといった設計抽象度を上げる方法 = ボトムアップアプローチ)では問題を解決することができず，仕様設計のやり方を変えること(トップダウンアプローチ)が必要となってきました。

当社のソリューション

当社では前述のような問題を根本的に解決するため，ソフトウェア開発で普及しているオブジェクト指向の考え方をシステムおよびシステムLSIの設計に導入し，「開発プロセス」を全面的に見直しました。この新しい「開発プロセス」実現のため，システム，ハードウェア，ソフトウェアの仕様記述言語としてUMLを採用し，実装および検証のための言語としてC言語を採用しています。

図1に従来型システムLSI設計手法を示します。ここではもっとも典型的な例として，システム仕様書を元にシステムをハードウェアとソフトウェアに分割し，それぞれ詳細化(抽象度の高い記述より抽象度の低い記述に具体化していく過程)を行っていく例を示しています。

従来型システムLSI設計手法では，仕様書は自然言語(日本語や英語)で書かれることが普通でした。また，ハードウェアとソフトウェアの分割も，アーキテクトと呼ばれるエキスパートエンジニアの勘と経験に頼っていました。

図1の右側に描かれたソフトウェア詳細化のフローは，既に確立されたものがあります。しかし，左側に描かれたハードウェア詳細化のフローでは，仕様書レベルと実装レベルには依然として大きなギャップがあります。また，ハードウェアとソフトウェアを合わせたシステム検証はそれぞれが完成してからになるため，この段階でミスが見つかっても手遅れとなることが少なくありませんでした。

図2に新システムLSI設計手法を示します。この手法には，従来型システムLSI設計手法と比べて大きく変わっている点が2つあります。1つは仕様記述の言語として，UMLを採用している点です。これにより，多人数で同じ概念を共有することが容易となり，仕様に関連したミスが削減されます。もう1つは，ハードウェアとソフトウェアの分割からハードウェア詳細化にかけて，UMLとC++/SystemCを用いる新しい開発プロセス(作業手順)を導入した点です。これにより筋道の通った設計を行うことが可能となり，設計のやり直しが削減され，全体として設計期間が短縮化されます。また，ハードウェアとソフトウェアの協調検証環境が早い時期から利用できるうえ，上流工程で作成した検証環境を詳細化後の設計データにも適用できるなど，検証作業の効率化も図っています。

図3にUMLの例を示します。

UMLは言語(Language)という名前がついていますが，実際は図(Diagram)を基本とした表現方法です。図3ではUMLの代表的な3種類の図を例として取り上げています。

左端のユースケース図は，システムにどのような機能があるか，またそれがどのように使われるかということを表現します。人の形を

図1 従来型システムLSI設計手法

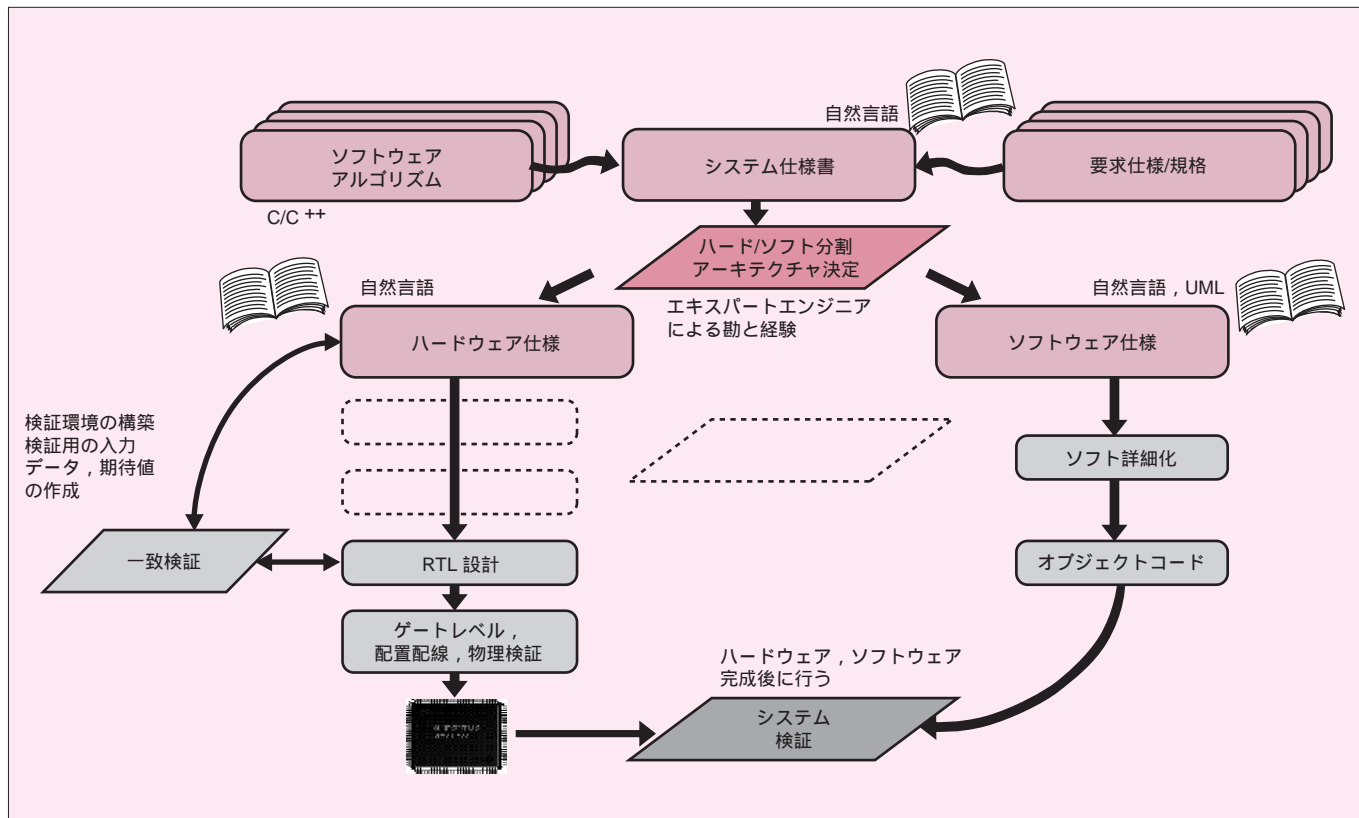


図2 新システムLSI設計手法

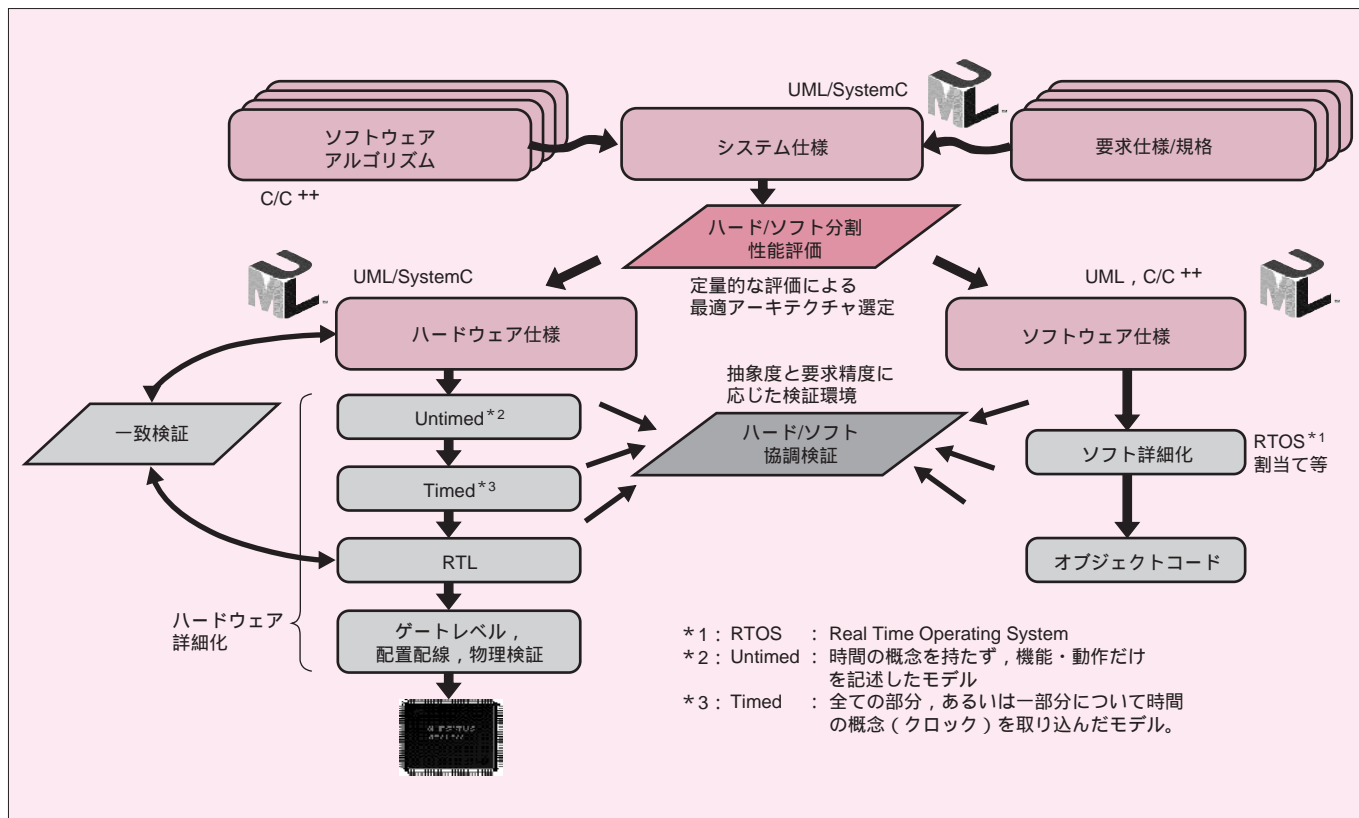
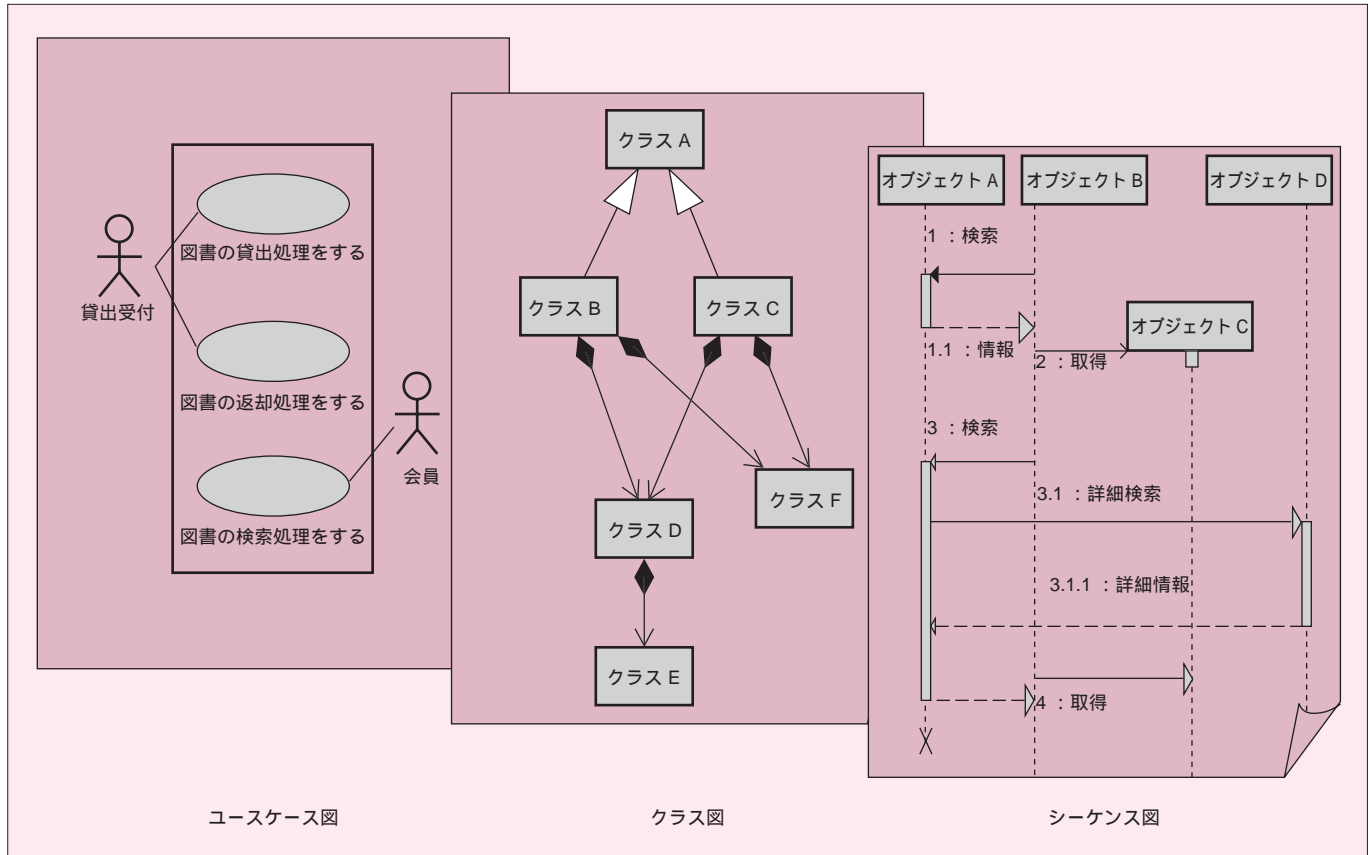


図3 UMLの例



した記号(アクター)は、人間でなく物でも構いません。

中央のクラス図は、システムで使われるクラス(オブジェクトの集まり)の関連付けを表わしたものです(クラスAはクラスBとクラスCからできているなど)。

右端のシーケンス図は、オブジェクト間の相互やりとり(メッセージの交換など)を時間軸を基準として表現したものです。ハードウェア設計をする際によく用いられるタイミング図(時間軸を横方向にとることが多い)とも似ています。

今回開発したシステムLSI設計手法は、単にUMLやC言語を導入したり、動作合成ツールを導入したことが特長なのではありません。本設計手法の優れた点は、ソフトウェア開発で用いられているオブジェクト指向開発プロセスをシステムLSI設計に拡張し、仕様設計からRTL設計に至る一貫した「開発プロセス」を定義したことです。またさらに、ソフトウェア用であったUMLで、システムやハードウェアを記述できるような仕組み(モデリング手法、検証手法、性能評価手法)を開発した点にあります。

当社では、本設計手法に関する教育プログラムも開発しており、既にハードウェア/ソフトウェア技術者を対象に、UML/C/C++/SystemCの各言語教育と設計手法教育を本格的に始めています。また、本設計手法を用いたシステムLSIの開発も開始しており、年内には製品開発を終える見込みです。

プラットフォームベース設計との連携

システムLSIを効率よく設計する手法として、プラットフォームベース設計手法があります。この手法は、複数の製品の開発を行う際に、繰り返し使われる部分をプラットフォーム(共通の土台、ひな型)として作り、それ以外の部分、すなわち製品として特長を持たせる部分の設計に注力することで、短期間によりよい製品を開発しようというものです。

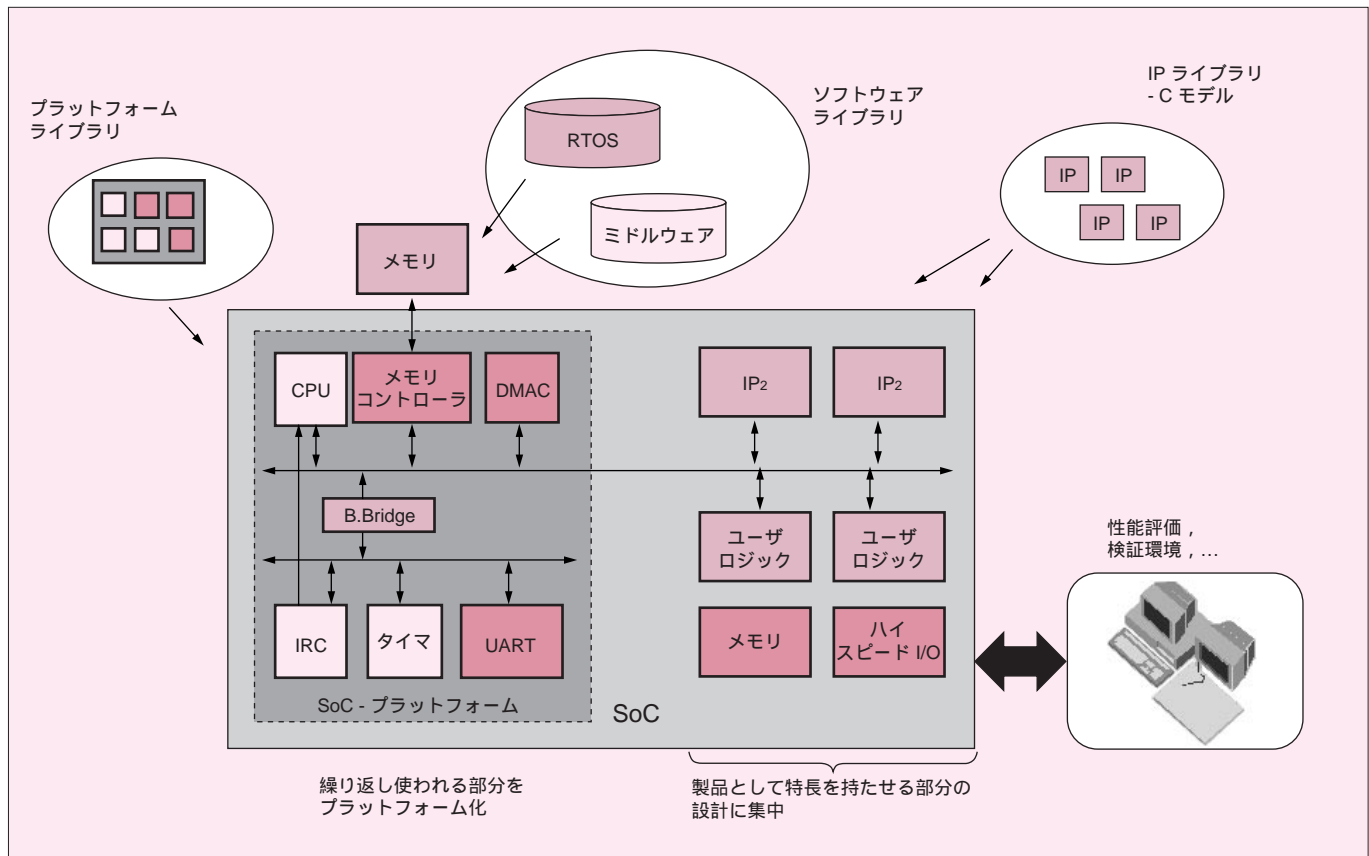
図4にプラットフォームベース設計手法を示します。

プラットフォームベース設計という概念そのものは新しいものではありませんが、従来と比べて設計に必要な4つの構成要素(プラットフォーム、IP、モデルウェア、検証環境等)が明確になり、またプラットフォームやIPをつなぐバスの標準化、検証環境の充実などで、より手軽に高機能なSoCを開発することが可能となっています。

当社では、このプラットフォームベース設計手法に、前述のUML/C言語ベース設計手法で開発された要素技術を盛り込み、さらなる設計効率の改善を図っていきます。

* JEITA：日本電子情報技術産業協会

図4 プラットフォームベース設計手法



[参考文献]

- (1)富士通/富士通研究所プレスリリース
「UMLとC言語をベースとしたシステムLSI設計手法を開発」
2002.4.16
- (2)朱強他, DAシンポジウム2002「システム設計技術」:
UMLを用いたシステムレベル設計手法の提案(B2-4)