

# 第1章

## 特集

# みんながシステム設計を 気にする理由

### 組み込みソフトとSOCの システム設計最前線

### ——システムLSI開発の視点から

山崎正実

ここでは、システムLSI開発の視点で、システム・レベル設計の現状について解説する。マイコンと周辺回路を1チップに集積したシステムLSIを開発する場合、まずハードウェアとソフトウェアの分割を行う。このときハードウェア処理とソフトウェア処理の境界部分を最適化(チューニング)する必要があるのだが、この作業が一筋縄ではいかない。また、システム・レベルの設計言語や記述表現が続々登場しており、それらの違いや特徴がわかりにくい。現在の状況は、1990年代のHDL導入期よりも混沌としているという。(編集部)

数年前、ハードウェア・ソフトウェア・コデザイン(協調設計)ということばが脚光を浴び始めたころ、仕事上の相棒であるソフトウェア技術者といっしょにEDAベンダの説明会に行ったことがあります。この会社のツールは、システム仕様を実行可能なC言語で記述し、アーキテクチャ設計や性能/コスト評価を行い、ソフトウェア・プログラムとハードウェア記述(HDLコード)を生成するというものでした。ハードウェア、特にASICの設計者であ

る筆者にとって、それは興味深い内容でした。しかし、ソフトウェア技術者である相棒の失望は、それはそれは大きなものでした。なぜでしょう？

#### ●「システム」ということばに惑わされるな！

相棒は通信・伝送装置(例えばSONET、WDM機器)の組み込みソフトウェア開発者です。彼にとってのシステムとは通信・伝送装置であり、ハードウェアとは数枚～数十枚のプリント基板で構成されたものです。そして、そのプリント基板には何十ものLSIが実装されています。彼にとってLSIは単なる部品です。彼のチームが開発するソフトウェアはリアルタイムOSの上で動作し、コード量は数十万ステップに及びます。EDAベンダの示すコデザイン・ツールに入力する実行可能な仕様記述を見て、彼は自分たちの仕様とあまりにもかけ離れたしろものであることを理解しました。その仕様記述は、彼らの最終成果物であるプログラムとほとんど変わらないレベルのものでした(図1)。

彼は、日ごろ、約束どおりの仕様でしあがってこないハードウェアのために苦勞していました。そのため、実際のハードウェアの設計データを使ってソフトウェア・プログラムの動作を確認できる(いわゆるハードウェア・ソフトウェア協調検証)という点に興味がありました。ところが、シミュレーションの実行速度がソフトウェア開発用のシミュレータと比べて3けた以上遅いということがわかりました。今まで数分で終わっていたシミュレーションに、何日もかかるというのではお話しになりません。加えてソフトウェア開発ツールと比べると、導入コストがべらぼうに高い(数百万～数千万円)という問題があることもわかってきました。



〔図1〕話がかみ合わないEDAベンダとソフトウェア技術者  
システムの定義はまちまち。EDAベンダにとってのシステムはシステムLSI。でも、ソフトウェア技術者にとっては、あるときは伝送装置、またあるときはもっと巨大なネットワークそのものだったりする。

大げさなことばやわかっているようで意味不明のことばのことを英語で“big word”と言いますが、まさに「システム」ということばはbig wordの一つです。ふだん何げなく使っていますが、開発する対象によって「システム」ということばの定義が異なる場合が非常に多いのです。前述の相棒の例も、そうしたすれ違いから起こった悲劇です。ですから、「システムなんか」なるものを議論するときは、システムとは何かを決めておく必要があります。

本稿ではシステム・レベル設計について議論しますが、取り扱う対象を「システムLSI」に限定します。「システムLSI」ということばもなかなか一筋縄ではいかないのですが、ここではアプリケーション・プログラムが動作するマイコンとメモリ、MPEGデコーダなどのメディア処理やモデム、プロトコル終端などの通信用DSPや専用ハードウェアなどから構成されるLSIを考えます。

一方、ソフトウェアについては、OSやアプリケーション・プログラムまで含めて考えると、話が大きくなりすぎます。開発対象によって事情は異なりますが、BIOSやドライバ・ソフトウェア、信号処理用ソフトウェアくらいが妥当ではないでしょうか。つまり、アプリケーション・プログラムやOSを実装するための基盤となるソフトウェアに限定して考えていきます(図2)。

●システム・レベル設計が無視できなくなった理由

従来、組み込み機器では、マイコン(CPU)、メモリ、専用LSI(ASIC)などの構成要素が別々のチップとして実現されていました。機能の複雑化に伴ってシステムが巨大になり、しだいにハードウェア開発とソフトウェア開発の分業と専門化が進みました。各開発グループはそれぞれの分野で設計効率の向上を追求し、それに伴ってお互いの関係は疎になっていきました。

この間、CPUとASICの間のインターフェース仕様は、基本的にあまり変わりませんでした。1冊のインターフェース仕様書が、ハードウェアとソフトウェアを結ぶ懸け橋になりました。CPUとASICの間のインターフェース仕様がほとんど変わらなかった理由は、主に二つあると思います。第1に、ソフトウェアが巨大化したことにより、ソフトウェア・アーキテクチャの変更を伴うインターフェース仕様の変更が敬遠されたためです。第2に、汎用のマイコンを使う関係で、インターフェース仕様がCPUバスという一種の標準規格に縛られていたためです。

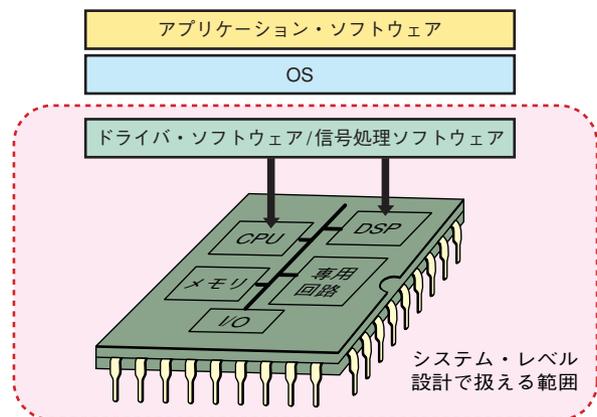
LSI技術の進歩は早く、いまや制御用プロセッサ、デ

ータ処理用プロセッサ、データパス回路、ソフトウェアといった組み込み機器の構成要素を一つのチップに集約できるまでになりました。この技術が、携帯電話やデジタル・カメラなど、小型のハイテク機器を実現するためのキー・テクノロジーになっています。

CPUについては、ひと昔前の汎用大型コンピュータ(メインフレーム)の性能をはるかに上回るものを簡単に入手できるようになりました。そのため、従来はハードウェアでしか処理できなかった機能をソフトウェアで実現できるようになってきました。

機器メーカーに対する市場の要求は多様化し、かつめまぐるしく変化しています。開発期間(time to market)を短縮することと、市場要求の変化に追従できる柔軟なシステムを開発することが要求されています。その一方で、ASICの開発期間は延びる傾向にあります。これは、回路規模が大きくなったためと、微細化の影響によりレイアウト設計時にタイミングを収束させることが難しくなったためです。例えば数百万ゲートのASICでは、論理検証が完了してからサンプル・チップを入手するまで、半年近くもかかっているのが現状です。半年もたつと市場の風向きが変わってしまいます。相当うまく開発していかないと、完成したときには機能や性能が陳腐化していたということになりかねません。

この点で、FPGAを利用する方法は有効です。ただし、FPGAを乱用すると、コストが上昇します。ASICを使いながらこの問題を解決するには、プログラマブルな部分(CPU)をASICに組み込む必要があります。機能をなるべくソフトウェア制御で実装し、ソフトウェアの追加や変更によって仕様の変化に対応するのです。こうする



【図2】システム・レベル設計で扱える範囲

現状のシステム・レベル設計で扱うソフトウェアは、BIOSやドライバ・ソフトウェア、信号処理用ソフトウェアくらいの大きさにするのが妥当。