

第1章

システム・レベル設計 とはなにか

木下常雄



ここでは、組み込み機器およびシステムLSIの開発者を対象としたシステム・レベル設計手法の概要について解説します。エレクトロニクスの世界では、設計対象が複雑になりすぎ、従来の手法で手に負えない状況がやってくると、「設計抽象度の引き上げ」が発生します。たとえば、組み込みソフトウェアのプログラミングはアセンブリ言語からC言語に移行しました。デジタルLSIの設計は回路図入力からRTLのHDL入力に変わりました。そして再び、設計抽象度の引き上げが起こるきざしが見えてきました。今度の転換は、組み込みソフトウェア開発者と回路設計者の双方を巻き込んだ大がかりなものになります。マイクロプロセッサを内蔵するシステムLSIとそれを搭載した組み込み機器の台頭が、設計手法の転換を促します。(編集部)

ふだん、なにげなく使っている「システム」という単語はたいへん便利な言葉です。辞書を引くと「組織、体系、順序、方式」とあります。「これがわが家のシステムです」と家内から言われて、毎朝出がけに1,000円札1枚をもらって通勤するわが身も、この「システム」という言葉にどっぷりつかっています。

一方、設計の世界でも、「システム」という言葉が氾らんしているように思います。電子機器の設計者は部品を組み上げ

て「システム」を開発しますし、半導体設計者はマクロ・セルやIPコアを組み合わせて「システムLSI」を構築します。ソフトウェア開発者にとっては、ソフトウェア・モジュールを統合したものが、すなわち「システム」です。さて、設計技術者にとっての「システム」とは、いったいなんなのでしょう(図1)。

●システム・レベルのカバー範囲は広い

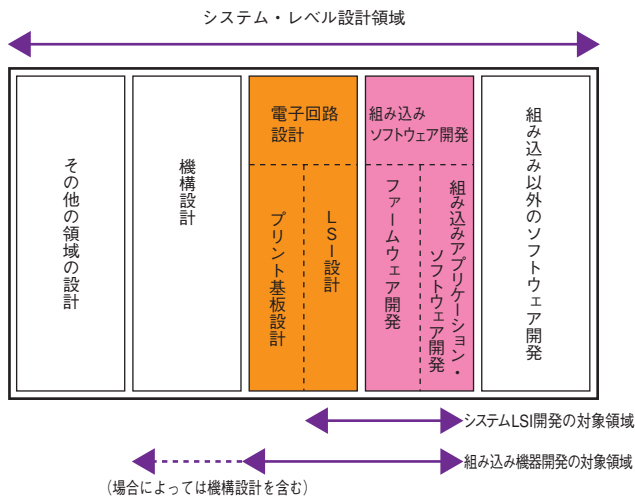
今年(2000年)の1月末に、横浜で開催されたASP-DAC (Asia and South Pacific Design Automation Conference) のパネル・ディスカッションでは、“One Language or More? (How Can We Design an SoC at a System Level?)” というテーマが熱っぽく論じられました。その冒頭で、司会者のGary Smith氏(半導体調査会社であるGartnerGroup社、Dataquestのアナリスト)が「システム・レベル設計」のカバー範囲を図で説明しました。このときの図を筆者なりに整理して作り直したのが図2です。

これまで、LSIの設計にたずさわってこられた方にとっては、システムLSIの対象領域が予想以上に広い範囲をカバーしていることに驚かれるかもしれません。しかし、高性能なプロセッサや大容量メモリを搭載したシステムLSIの開発で



〔図1〕設計技術者にとっての「システム」とは?

機器設計者、半導体設計者、ソフトウェア開発者、それぞれの世界にシステムのイメージがある。



〔図2〕 システム・レベル設計の領域

2000年1月に開催されたASP-DACでGary Smith氏が示したシステム・レベル設計の図を整理して作り直した。従来はLSI設計、プリント基板設計、ソフトウェア開発など、個別に開発手法が検討されてきた。システム・レベル設計では、これらを横断的に取り扱う。本特集では、おもに電子回路設計と組み込みソフトウェア開発の連携が必要となる組み込み機器、およびシステムLSIの開発手法に焦点をあてる。

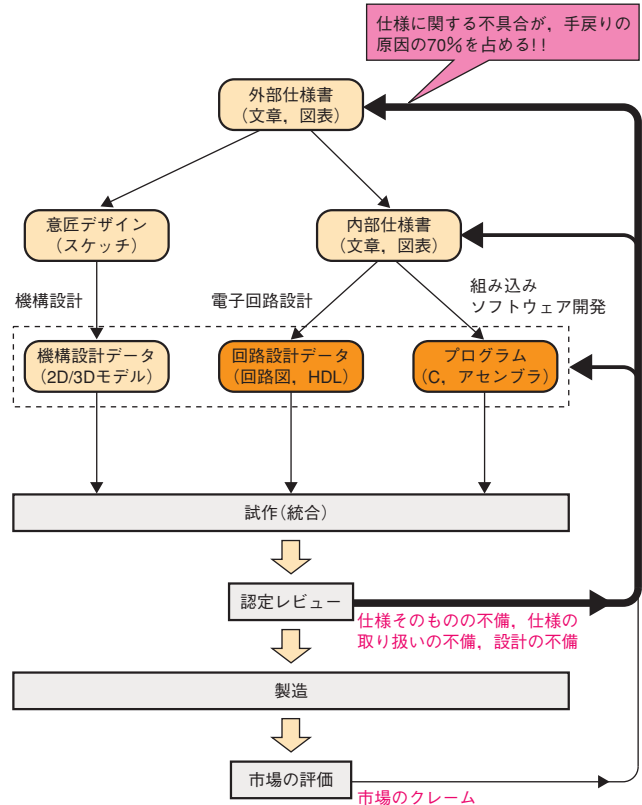
は、組み込みソフトウェアを切り離して考えることはできません。一方、組み込み機器などを開発している機器設計者は、プリント基板や筐体などにも配慮する必要があります。

ここに示されているシステム・レベルに該当する製品としては、ハード・ディスク装置、DVD装置、CD-ROM装置、メモリ・カードなどの記憶装置、プリンタ装置、通信機器、AV機器、情報家電などが挙げられます。図2からわかるように、システム・レベルにはソフトウェアもハードウェアも含まれています。読者のみなさんがどちらの領域に属している技術者であっても、ご自分がシステム・レベル設計の対象者であることを納得していただきたいと思います。

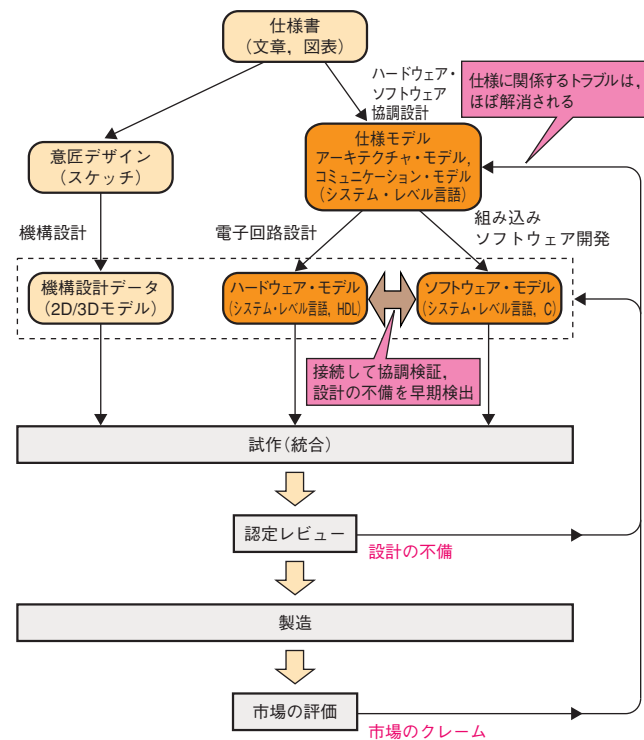
●仕様決め、ハード設計、ソフト開発が混然と進む

組み込み機器やシステムLSIを開発する際の「システム・レベル設計」の範囲が明確になったところで、次に、これらの設計フローを考えてみましょう。従来、よく行われている組み込み機器の設計フローを図3(a)に示しました。システムLSIの開発では、左端の機構設計の部分が存在しないことになります。また、設計対象(あるいは会社の業態)によっては、組み込みソフトウェア開発のみ、あるいは電子回路設計のみで業務が閉じている場合もあるかと思えます。

図3(a)のフローにしたがって、まず要求仕様をきっちり固め、その後、ソフトウェアの仕様分析、機能抽出、タスク設計、あるいはハードウェア(電子回路)のアルゴリズム設計、アーキテクチャ設計、ビヘイビア設計、RTL設計というように、上流から下流へ向かって整然と作業を進めるのが理想で



(a) 従来の設計フロー



(b) システム・レベル言語を導入した場合の設計フロー

〔図3〕 システム・レベルの設計フロー

(a)では、設計の手戻り作業が多く発生する。仕様書を表示する言語(自然言語)と設計言語が異なっており、設計言語もハードウェアとソフトウェアで異なっている。仕様の内容について、誤解が生じやすい。(b)では、設計の手戻り作業が少なくなる。仕様書を表示する言語、ハードウェアの設計言語、ソフトウェアの設計言語が共通になるので、仕様に関する誤解が生じにくい。