特集

ソフト&ハード，両方知っててあたりまえ！

Introduction

ソフトウェア・ハードウェア・ボーダレス技術者の時代へ

編集部

ハードウェア技術者とソフトウェア技術者の間には，高い壁があるように見えます．それは知識の壁であり，設計文化の壁であり，コミュニケーションの壁です．それぞれの技術者がそれぞれの世界に閉じこもってしまい，コミュニケーションがとりにくくなっているのではないでしょうか．コミュニケーションの不足は相手に対する不信感を生みます．「ハードウェア開発のスケジュールの遅れを，ソフトウェアの開発期間の短縮でカバーさせられている」，「ソフトウェア技術者に仕様書を渡しても，なかなかハードウェア側の意向をくんでくれない」．設計現場からこのような声が頻繁に聞こえてくるようになりました（図1）．

電子機器の開発現場では，ある時期にハードウェアの設計グループとソフトウェアの設計グループが分かれました．それぞれの技術者がそれぞれの分野で専門性を高め，設計効率を上げる努力を積み重ねてきました．そして，それはおおむねうまくいっていました．しかし，最近になって，こうした構図がくずれ始めています．理由はいくつかありますが，ハードウェア技術者にはソフトウェアの知識が，ソフトウェア技術者にはハードウェアの知識が求められるようになってきました（図2，図3）．すなわち，ボーダレスで考えられる技術者が必要とされています．以下では，こうした変化の背景となっている動向を簡単に紹介します．

●システム・アーキテクトが足りない

たとえば，交換機や伝送装置，ネットワーク装置などでは，性能と信頼性が重要になります．そのため，専用LSI（ASIC）の開発が欠かせません．しかし，最近では，性能や信頼性に加えて，TAT（turn-around-time）やネットワークを利用した保守の容易さなども求められるようになっています．多数の専用LSIを設計すると，どうしても開発期間が長くなりがちです．そのため，これまで必要な機能を専用LSIで実現する割合が高かった機器でも，その機能をできるだけソフトウェアで実現していこう，という方向に向かっています．また，一般に，ソフトウェアの変更によって機能を調整できる構成をとる電子機器が増えています．

こうした機器の開発では，性能やコスト，消費電力，保守性など，さまざまな制約条件を考慮しながら，最適なアーキテクチャ（ハードウェアとソフトウェアのバランス）を追求することが求められます．ハードウェアとソフトウェア，双方のよい点――ハードウェアの性能，ソフトウェアの柔軟性・保守性――を引き出すことが，設計技術者の腕の見せどころと言えます．とくに大手電機メーカでは，ハードウェアとソフトウェアの両方の知識をもったシステム・アーキテクトの確保が大きな課題になっています．

〔図1〕ハード技術者とソフト技術者のコミュニケーションがとれない

もともとハードウェア技術者とソフトウェア技術者の間に壁はなかった．設計対象が複雑になり，組織が大きくなるにつれて，ハードウェア技術者とソフトウェア技術者が分かれていった．それぞれの技術者が専門化した結果，いつのまにか円滑なコミュニケーションがとれなくなってきた．

●ハード技術者とソフト技術者の協調と相互理解が必要

多くのシステムLSIには，CPUコア，メモリ・コア，周辺論理などが1チップに集積されます．従来，汎用LSIであるマイコン，メモリ，専用LSIであるASICは別々のチップでした．そのため，マイコン上で実行する組み込みソフトウェアの開発とASIC（周辺論理）の開発は，その作業フローが明確に分かれていました．しかし，システムLSIの開発では，事情が異なります．それぞれの作業が相互に影響し合います．つまり，ハードウェア技術者とソフトウェア技術者の間の協調作業が不可欠になります．協調作業を円滑に進めるためには，互いに相手の事情を理解しておくことが重要です．

たとえば，システムLSIを開発する際に，ハードウェア・ソフトウェア協調検証（コベリフィケーション）を行って，開発期間を短縮しようという試みがあります．これは，ハードウェアのシミュレーション・モデルをソフトウェア・デバッグのための仮想プロトタイピングとして利用し，ソフトウェア・プログラムをハードウェア・デバッグのためのテストベンチとして利用しようという考え方です．実際のチップを製造する前に，仮想的にハードウェアとソフトウェアの統合テストを行えます．

現在のハードウェア・ソフトウェア協調検証ツールは，検証速度や使い勝手などに課題があります．このような制約のあるツールをうまく使いこなすには，ハードウェアとソフトウェア双方の設計や開発ツールに関する知識が必要になります．

●ハードもソフトもC/C++で設計

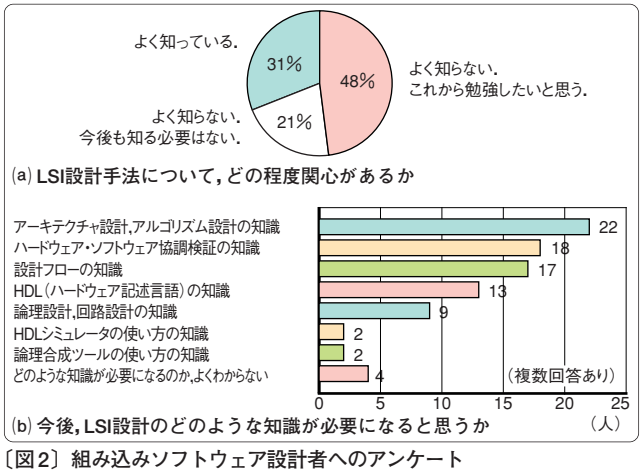
システム・レベル設計を支援するシステム・レベル設計言語の標準化が昨年（1999年）の後半からスタートしています．これは仕様設計，回路設計，組み込みソフトウェア設計を共通の言語で行おうという試みです．システム・レベル設計言語の世界では，C/C++（あるいは，その拡張言語）を利用する手法が優勢です．ハードウェア技術者であっても，C/C++を自由に使いこなしていく必要があります．

●米国での訴訟をきっかけに検証エンジニアを配置

HDLによるLSI設計ではテストベンチ（検証環境）を作成します．テストベンチは，回路設計というよりも，ソフトウェア・プログラミングに近い作業です（ハードウェア特有の制約がない）．実際，ソフトウェア技術者が担当したほうが，品質のよいテストベンチができあがる，という声もあります．また，通信や画像，音声などを扱うLSIの開発では，アプリケーションに合わせた高度な検証環境を構築できるかどうかで，検証効率に差が出てきます．

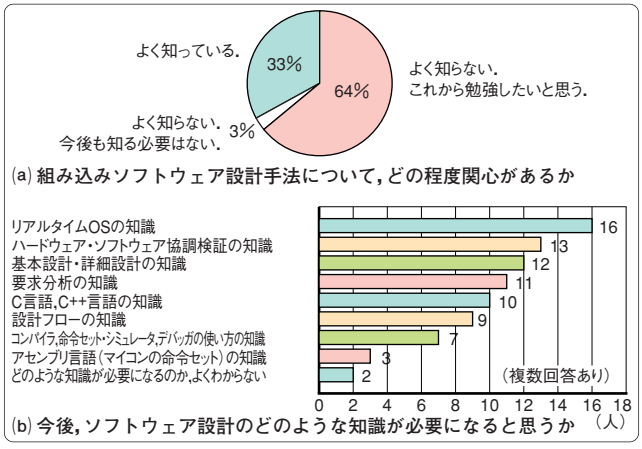
米国では，こうしたLSIの検証業務を“検証エンジニア”と呼ばれる専門家が担当しています．検証エンジニアには，LSI設計の知識はもちろん，ソフトウェア・プログラミング，およびコンピュータやネットワークのシステム構築の知識が必要になります．国内でも，最近になって，少しずつ検証エンジニアの導入が始まっているようです．ある国内の大手企業では，コンピュータ機器の不具合に対する訴訟が米国で起こり，それをきっかけに設計部門の検証体制の見直しを行うことになったそうです．その一貫として，検証専任のエンジニアを配置し，デザイン・レビューに参加させるようになったと言います．

Design Wave Magazine編集部　中山俊一



〔図2〕組み込みソフトウェア設計者へのアンケート

2000年7月に開催された第3回組み込みシステム開発技術展（ESEC）で調査した．組み込みソフトウェア開発者，および組み込みソフトウェアとLSIの両方を開発している技術者（合計56人）の回答を合わせた．データの提供は組み込みソフトウェア開発ツール・ベンダのキャッツ．



〔図3〕LSI設計者へのアンケート

図2と同じように，ESECで調査した．LSI設計者，および組み込みソフトウェアとLSIの両方を開発している技術者（合計29人）の回答を合わせた．データの提供はキャッツ．