重点企画

ハードウェア/ソフトウェア・コデザイン

ASIC設計現場におけるアプローチ

河合高志

1．ハードウェア/ソフトウェア・コデザインはなぜ必要か

●システム・オン・チップの時代に向けて

ディープ・サブミクロン技術によって，LSIの集積度と動作速度は飛躍的に増加してきています．これまでのASICは，高度なシステムを構成する際に，CPUに制御されて専用の処理を行うような，周辺チップ的な使われ方がされてきました．しかし，最近では，従来CPUが行っていた処理までもASICに取り込めるようになっています．すなわちCPUコアとユーザが設計したロジックを一つのチップに搭載したり，CPUコアとDSPコアを搭載したマルチメディア用のチップのような，システム・オン・シリコン・チップ(System on silicon chip)が開発され始めています．

●従来のＡＳＩＣとシステム・オン・シリコン・チップとは何が違うのか

従来の，比較的単純なＡＳＩＣでは，ASIC単体の機能を検証するだけで十分でした．ハードウェアの設計仕様どおりに動作していることさえ確認できればよいということです．システム・レベルでみても，それにより大きな問題が発生することはありませんでした．

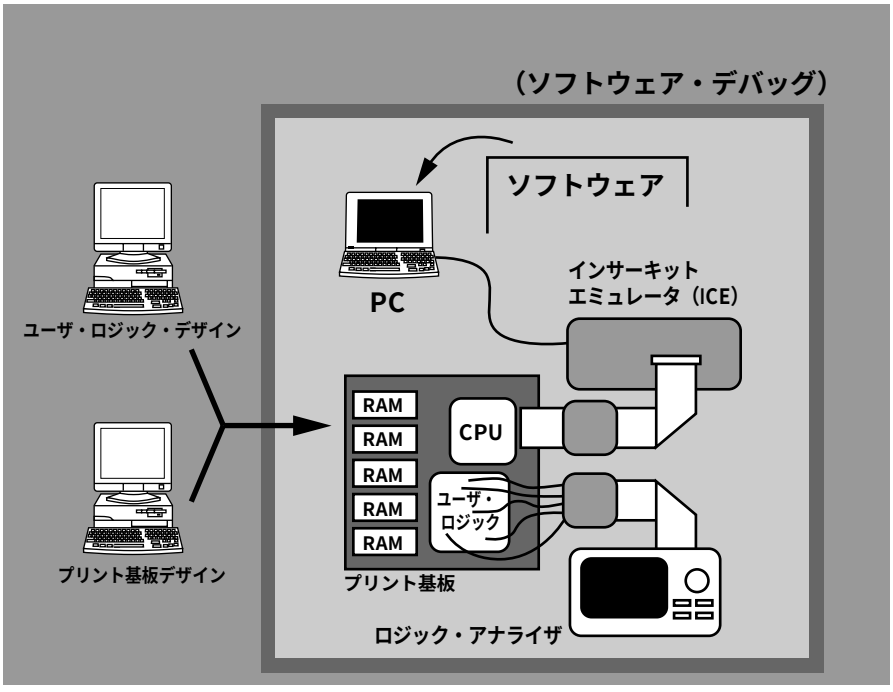

図1　従来のCPUコア搭載ASICの開発 /デバック環境

しかし，CPUに制御された大規模ASICや，CPUまたはDSPを内蔵したシステム・オン・シリコン・チップにおいては，ハードウェアとこれを制御するためのソフトウェアを同時に検証する必要があります(**図1**)．ハードウェアの動作がソフトウェアにより制御されることから，ハードウェアのみでは正しい動作をするかどうかの検証ができないので，ソフトウェアと組み合わせる必要があります．しかもこれは，設計時のシミュレーション段階で行われていなければなりません．

HDLと論理合成を利用したトップダウン設計はかなり普及してきていますが，

この技術だけでは今後要求されるであろう高度なASICの開発は困難になってきています．トップダウン設計の技術に加えて，ハードウェアとソフトウェアのコデザイン(協調設計)技術が求められています(**図2**)．

●従来の開発フローにおける問題点

従来の開発フローでは，製品の仕様が決定された後は，ハードウェアとソフトウェアは独自に開発され，試作品ができた段階で，ソフトウェアのデバッグが始まるのが一般的でした．

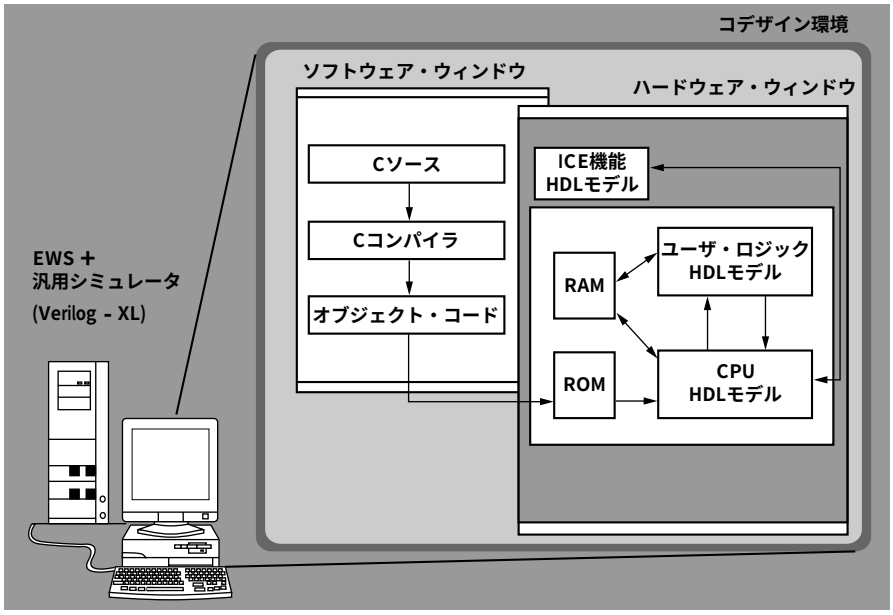

図2　ハードウェア/ソフトウェア・コデザインによる環境の例（CPUコア搭載型ASICの開発）

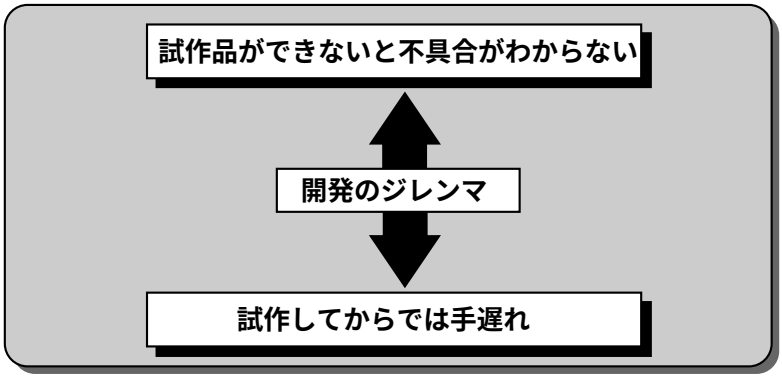

図3　システム開発の問題点とコデザインの重要性

しかしながらこのような方法をとる場合，以下のような問題が発生してしまいます(**図3**)．

⑴試作品ができるまで最終的なソフトウェアのデバッグができない．

⑵試作品ができあがった後でハードウェアやソフトウェアの不具合が発見される．

ハードウェアにせよ，ソフトウェアにせよ，不具合の修正は，開発の初期段階ほど容易です．後になるとむだな時間が増えるだけでなく，不具合自体が複雑になり，解決が困難になります．

まずはじめに試作品を作って評価し，不具合を修正して再度試作するというような手順で開発をしていると，不具合の発見が開発の後期になるため，納期の遅延と開発コストの増大が無視できないほど大きくなってしまいます．また，製品のライフ・サイクルが短く，競争が激しい市場では，製品の投入時期が遅くなるということは致命的です．

多くの企業では，ハードウェアの開発とソフトウェアの開発は別々のグループで行われています．ハードウェアとソフトウェアの開発では，開発手法，開発環境，技術用語，常識，技術文化が異なっており，共有できる開発環境はほとんどありません．困ったことに，システムの性能に関わる不具合やハードウェアの再設計につながる欠陥は，ハードウェアとソフトウェアが密接に結びついた境界領域で起こることが多いのです(**図4**)．

既存のハードウェアやソフトウェアの開発環境で，論理的なバグのないソフトウェアと，仕様どおりのハードウェアを設計したとしても，これらを組み合わせて期待した性能が出せるかどうかは，実際の試作品を作ってみるまでわからないのが現状です．