

電子機器製品の高品位設計を実現するESLツール S.E.R. studio

S.E.R. studio ESL Tool for Design of High-Quality Platforms for Electronic Devices

荒木 大 中村 敦資

■ ARAKI Dai

■ NAKAMURA Atsushi

システムLSIの集積度が高くなり構造が複雑になったため、その性能や特性がわかりにくくなり、アプリケーションを稼働させたとき予期しない不具合が発生するケースが増えている。

(株)インターデザイン・テクノロジーは、このような課題を解決するために、ESL (Electric System Level Design) ツールのS.E.R. studioを商品化した。このESLツールを活用した実機レスでの設計評価環境は、実機のハードウェア (HW) 及びソフトウェア (SW) を開発する前の上流設計の段階で、実機アーキテクチャの作り込み、設計の高品位化、及び設計結果の可視化などに対して非常に有効である。

Unexpected failures of electronic devices at an early stage of operation have been increasing due to designers' insufficient understanding of the performance and characteristics of system large-scale integrations (LSIs) installed in these devices along with the acceleration of integration density and structural complexity.

To overcome these problems, InterDesign Technologies, Inc. has developed S.E.R. studio, an electronic system level (ESL) design tool that realizes a virtual design environment in the upstream design phase to effectively optimize the hardware and software architectures, design a high-quality platform, and visualize the design performance.

1 まえがき

電子機器製品の中核を成す半導体の集積度は、非常に速いスピードで高まってきている。集積度が高くなるに従ってLSIの規模が大きくなり、CPUコアや周辺機能などの複数のモジュールから構成される電子システムを一つのチップ上にまとめた、システムLSIが主流になってきている。

マルチコア型プロセッサの普及などによりHW自身の性能向上が図られる一方で、マイコンやシステムLSIの構造や特性がいっそう複雑化しつつある。このため、電子機器製品の設計者にとって、共通部品あるいは汎用部品として利用するマイコン、システムLSI、OS (基本SW)、ミドルウェアなどHW及びSWのコンポーネントやプラットフォームの性能と特性がわかりにくくなり、電子機器製品を稼働させたとき予期しない性能のボトルネックや不具合が生じる可能性が増えている。設計工程の後段でこういった要求性能の未達が発生することは、設計コストを増大させ、また、タイムリーな商品化時期を逃して大きな機会損失につながる。

(株)インターデザイン・テクノロジーは、このような問題を解決するため、ESL (Electric System Level Design) ツールのS.E.R. studioを商品化した。ここでは、このツールの概要と特長などについて述べる。

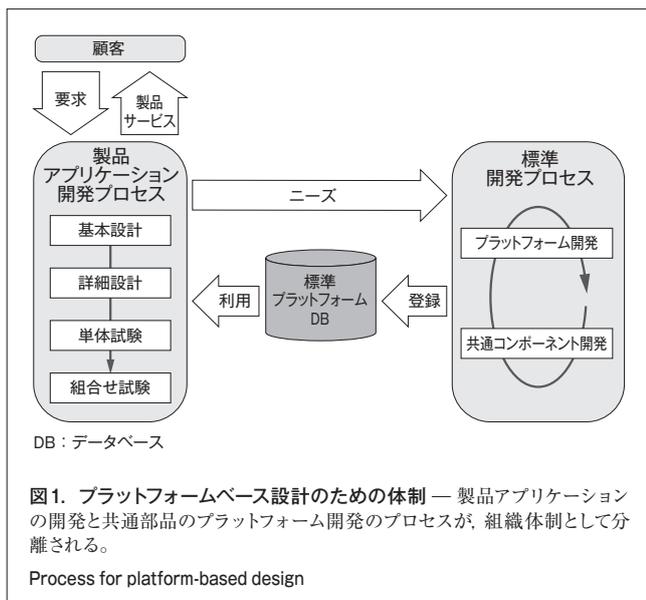
2 ESLを活用したHW/SWプラットフォームの開発

電子機器製品の量産開発及び派生品開発のコストダウンに向けて、HW及びSWの共通要素としてのプラットフォームをできるだけ整備して活用するプラットフォームベース設計が注目されている。ここでは、ESLツールのS.E.R. studioをプラットフォームの共通要素の蓄積や再利用に活用し、電子機器製品のHW及びSWの設計を容易化する方法について述べる。

2.1 プラットフォームベース設計

一般に、電子機器製品では新規品よりも派生品の開発の比重が圧倒的に高く、高生産性と高品質を両立させるために、徹底した設計の再利用性が要求されてきている。そこで、**図1**に示すように、製品アプリケーションの開発プロセスと、HW及びSWの共通要素としてのプラットフォームを開発する標準開発プロセスを分離した、組織体制作りが必要となってきた。製品アプリケーションの開発プロセスは、製品とサービスの顧客満足度向上や高付加価値性に注目する。標準開発プロセスは、製品アプリケーション開発の下支えとして、品質、性能、及び再利用性に重点を置き、高品位なプラットフォームや共通コンポーネントの整備に注力する。

このようなプラットフォームベース設計を電子機器製品の開発手法として定着させていくなかで、標準プラットフォームとして整備した共通コンポーネントを、製品アプリケーション開発のなかで効率的に活用できる設計支援環境として、ESLツ



ルに対する期待が高まっている。

2.2 ESL (システムレベル設計)

ESLは、電子機器製品の設計工程のできるだけ上流で製品仕様を明確に定義して検証を行う、フロントローディングを具現化させる有効な手段である。また、仕様の定義にC言語(プログラミング言語)が使用されることが多いため、“C言語ベース設計”とも呼ばれている。

ESLの設計手法の特徴は、仕様として定義した機能でHWとして実装する部分とSWとして実装する部分の分割を行う、設計詳細化ツールを用いる点である。また、HWの実装設計では、C言語ベースの設計からRTL(Register Transfer Level)の記述を自動的に生成する、動作合成ツールを活用する点も特徴の一つである。

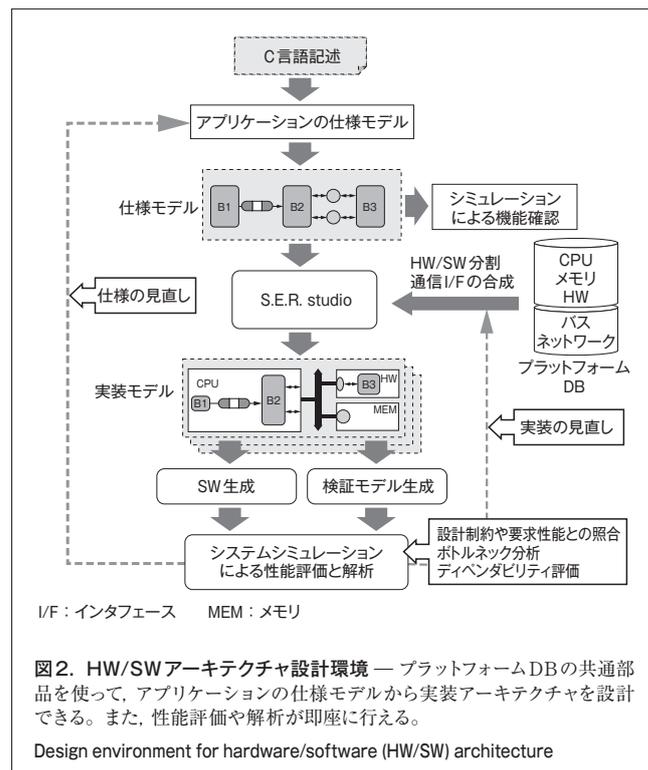
システムLSI設計の観点で見ると、従来はRTLから設計を開始することが普通であった。しかし、ESLの設計手法では、機能を実現する動作だけを記述したレベルから設計を開始するため、仕様を記述する量と検証時間を減少でき、開発期間を短縮できる。

2.3 HW/SWアーキテクチャの設計環境

ESLツールを使ったHW/SWアーキテクチャの設計環境を図2に示す。設計環境の特徴は次の二つである。

- (1) 設計詳細化ツール S.E.R. studioによって、プラットフォームデータベース(DB)に登録されたCPU、メモリ、バスなどの共通コンポーネントを使い、HWとSWの実装アーキテクチャを自由に設計できる。
- (2) システムシミュレータを使って、HWとSWの実装アーキテクチャの性能評価や解析が即座に行える。

設計環境に対する入力仕様は仕様モデルである。仕様モデルは、ビヘイビア(機能)と呼ぶC言語で記述した複数のプロセ



スでアプリケーションの機能仕様を定義し、図2のB1～B3の箱がビヘイビアを表している。ビヘイビアは並列に動作し、かつ、互いに通信しあう形で表現する。仕様モデルは、シミュレータを使って動作確認できる。

S.E.R. studioは、仕様モデルをHWで実装する部分とSWで実装する部分に分割する。また、プラットフォームDBに登録された共通部品を使って、HWのアーキテクチャ、CPUに搭載するSWのアーキテクチャ、CPUとHWを接続するバスやメモリのアーキテクチャといった実装の詳細を設計できる。

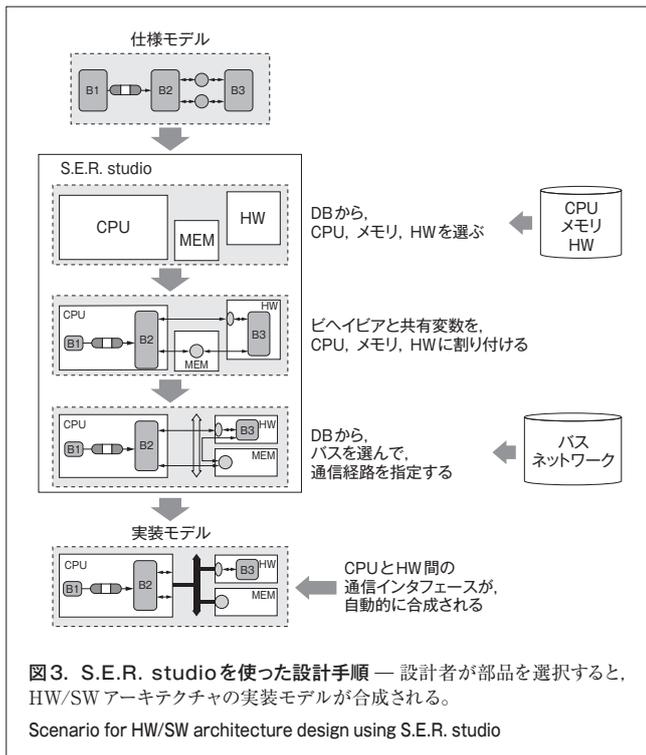
設計された実装モデルから、更に、組込みSWのソースコードと検証用の仮想HWモデルを自動生成でき、これらをシステムシミュレータで動作させることで性能評価及び解析が行える。

設計者は、実装設計の性能に問題がある場合、仕様モデルや実装方法を見直して、アプリケーションの詳細設計を自由に変更できる。

2.4 S.E.R. studio (設計詳細化ツール)

S.E.R. studioは、Specify-Explore-Refine設計手法⁽¹⁾に由来する名称で、JAXA (Japan Aerospace Exploration Agency: 宇宙航空研究開発機構)が開発したELEGANT (Electronic Design Guidance Tool for Space Use: 宇宙用電子機器設計支援システム)⁽²⁾の中核を成すESLツールとしてカリフォルニア大学で開発され、(株)インターデザイン・テクノロジーが商品化した。

S.E.R. studioを使った設計手順を図3に示す。最初の設計ステップでは、プラットフォームDBの部品表から必要なCPU、



メモリ、HWを設計者が選ぶ。

第2のステップで、仕様モデルとして入力したビヘイビアとビヘイビア間の通信変数を、設計者がCPU、メモリ、HWに割り付ける。S.E.R. studioは、仕様モデルの構造を解析してツリー構造として図示してくるので、設計者は入力したソースコード自身を見ることなく、ツールのGUI (Graphical User Interface) を使って、これらの選択を簡単にまちがいがなく指定できる。

第3のステップとして、CPU、メモリ、HWを接続するためのバスをプラットフォームDBに登録された部品から選んで、それぞれのコンポーネントの接続経路を設計者が指定する。

これらの入力設定が終了すると、指定されたHW/SW アーキテクチャの実装モデルが自動生成される。実装モデルの中で、仕様モデルの中に定義された機能が、CPUあるいはHWに分割して配置される。更に、それらの間で、バスやメモリなどを介したデータ送受信が実行される通信インターフェースの設計が自動的に合成される。

S.E.R. studioは、トランザクション精度モデルとピン精度通信モデルの2種類の実装モデルを自動生成する。

- (1) トランザクション精度モデル バスへのアクセスを関数メソッドの形で表現したモデルであり、高速にシミュレーションできる。
- (2) ピン精度通信モデル バスの信号線及びバスの信号の読み書きがビット単位で表現されたモデルになっており、HWやバスに関する高精度な性能評価を行いたい場合に用いる。また、設計データをHWの動作合成ツールに受け渡す際にも用いることができる。

2.5 システムシミュレーション

S.E.R. studioを使って設計した実装モデルから生成される組込みSWのソースコードを動作させるためには、リアルタイムOSやミドルウェアなど組込みSWのプログラムコンポーネントを含めたSW環境一式を、シミュレータで動作させることになる。

これらの組込みSWと、S.E.R. studioで作られる仮想HWモデルの協調シミュレーションを行うための環境の一つとして、CoMET[®](注1)がある。CoMET[®]は、CPUサイクルの粒度でモデリングされたCPUモデルによる、高精度かつ高速なHW/SW協調シミュレーション環境である。ユーザーが自由にHWモデルを作成してシミュレーション環境を構築することも可能で、対応CPUについても十分なバリエーションを持っている。

そのほかに、S.E.R. studioのバックエンドで利用可能なシステムシミュレータには、VisualSpecTM for Embedded⁽³⁾もあり、利用目的に合った性能評価環境を選ぶことができる。

システムシミュレーションから得られる情報の主要な項目には、次のようなものがある。

- (1) SW実行部分のレイテンシ(注2)の評価 ターゲットCPU上でSW部分を実行した場合の演算量と演算時間を解析できる。
- (2) HW/SW間の通信量の分析 HWとSWを分割したモデルのシミュレーションによって、HWとSW間のバス通信量の分布や経時変化を分析できる。

3 マイコンプラットフォーム検討への適用

CPUコアを複数搭載して演算処理性能を高めたマルチコアプロセッサは、一つのCPUコアの動作周波数と同等でありながら、CPUコア複数個分の処理性能を発揮できるのが特長である。しかし、マルチコア型マイコンでは、搭載される入出力インターフェース(I/O)などの周辺HWの点数も増えるとともに、内部バスやメモリの構成と構造も複雑化する。このために、アプリケーションの設計から見たマイコンの性能や特性がわかりにくくなり、アプリケーションをマイコン上で稼働させたときに、予期しない性能のボトルネックや不具合が生じる可能性が増えている。

ある特定用途向けのマルチコア型マイコンの基本設計で、HW/SWアーキテクチャ設計環境を使って、基本設計中のマイコンの性能評価を行った。基本設計中のマイコンは、まだ内部構成の実現方式案が複数あったので、S.E.R. studioを使ってそれぞれの仮想設計を行い、マイコンの設計要件や要求に関する性能をCoMET[®]を使って計測し、各案の相互比較を行った。

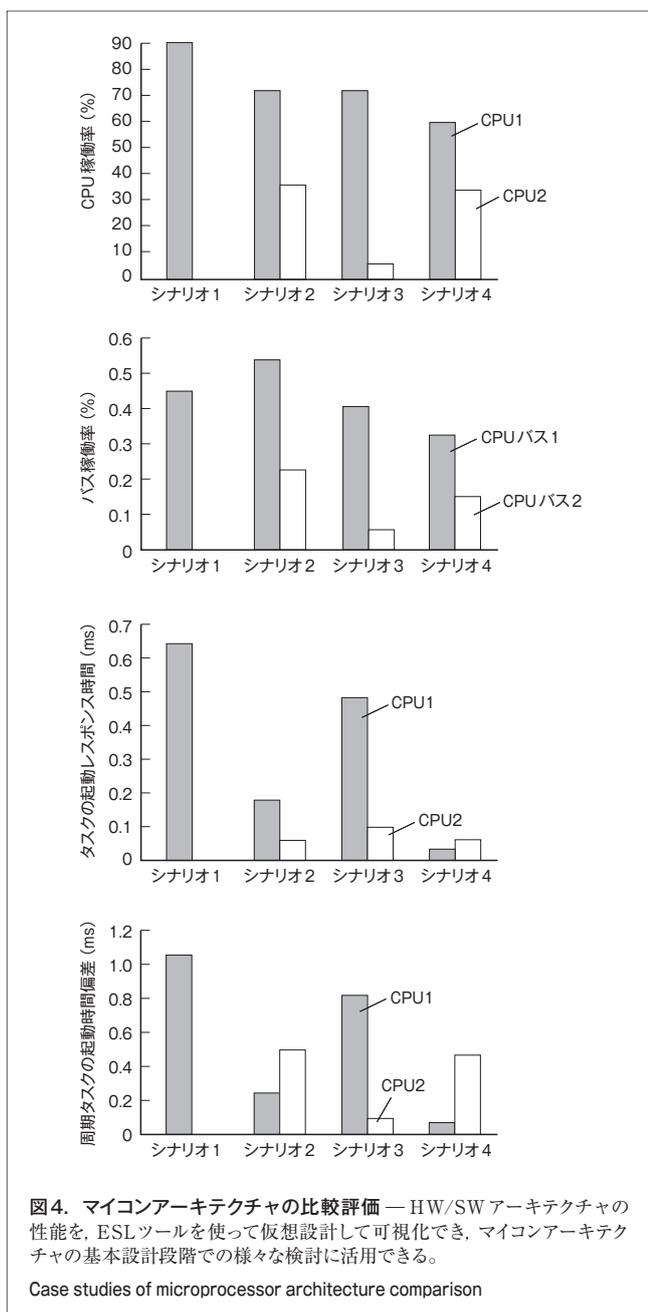
マイコンのアーキテクチャ設計のかなめとなる部分は、複数

(注1) CoMETは、米国VaST Systems Technology社の登録商標。

(注2) プロセッサでの命令実行が完了するのに要するクロック・サイクル数。

のCPUコアを接続するためのバスアーキテクチャ、共有メモリアーキテクチャ、I/O、タイマなど周辺HWの配置であった。そこで、CPUコアの数、バスアーキテクチャ、共有メモリアーキテクチャ、周辺HWの配置を変えた、4種類の検討シナリオを用意した。

更に、性能評価のための複数のタスクと、タスク間通信によって構成される模擬アプリケーションを使って、仮想設計した4種類のマイコンを比較した。比較作業は、模擬アプリケーションのタスクを複数のCPUに分散して配置し、バスアーキテクチャや共有メモリアーキテクチャの違いによる性能や挙動の差異を計測する方法で実施した。



複数のタスクで構成されたRTOS (Real Time Operating System) を使って稼働する組込みSWを用い検討したマイコンアーキテクチャ間での性能差の比較例を図4に示す。シナリオ1はCPUコアが一つの場合で、シナリオ2~4はCPUコアが二つの場合である。CPUコアが増えると全般に性能は向上するが、アーキテクチャの違いでソフトウェアのレスポンスや実行タイミングにずれや揺らぎといった特性の差が出ることがわかる。

このような形で、HW/SWアーキテクチャの性能を、ESLツールを使って仮想設計して可視化することができ、マイコンアーキテクチャの基本設計段階での様々な検討に活用できた。

4 あとがき

電子機器製品の設計開発で、“フロントローディング”が設計の生産性向上に必須の事項と認識されつつある。この実現には、上流設計工程の段階で、設計対象システムのHW/SWアーキテクチャの実装に対する検討を、短期間で効率よく最適な設計に追い込んでおくことが重要である。

ESLツールを活用した実機レスでの設計評価環境は、実機のHWとSWを開発する前の上流設計の段階で、実装アーキテクチャの作り込み、設計の高品位化、設計結果の可視化などに対する有効な手段である。しかし、ツール環境あるいはプラットフォームDBの準備や整備のためにコストと期間を必要とするので、派生品の開発が繰り返し行われるような製品をターゲットとしないと、投資対効果が出にくい点が課題である。

今後とも、各種製品設計への適用や試行の積み重ねを通して活用技術を蓄積するとともに、設計ツール自身に対する機能拡張や性能向上にもつなげていく。

文献

- (1) Gerstlauer, A., et al. "Specify-Explore-Refine(SER): from Specification to Implementation". Proceedings of the 45th Design Automation Conference. Anaheim, CA, USA, 2008-06. ACM/IEEE, 2008, p.585 - 591.
- (2) 神武直彦, ほか. “宇宙用電子機器設計の挑戦—システムレベル設計環境ELEGANT”. STARCフォーラム/シンポジウム2008. <http://www.starc.jp>, (参照2009-07-07).
- (3) 石井正悟, ほか. 組込みソフトウェア向け仮想ボードシミュレータの有効性検証. 東芝レビュー. 63, 1, 2008. p.63 - 67.



荒木 大 ARAKI Dai, Ph.D.

(株) インターデザイン・テクノロジー エンベデッド・ソリューションテクノロジー事業部 技術開発グループ部長、工博。システムレベル設計ツールの企画・開発に従事。IEEE、情報処理学会会員。InterDesign Technologies Inc.



中村 敦資 NAKAMURA Atsushi

(株) インターデザイン・テクノロジー エンベデッド・ソリューションテクノロジー事業部 技術開発グループ シニアエンジニア。システムレベル設計ツールの開発及びエンジニアリングサポートに従事。情報処理学会会員。InterDesign Technologies Inc.