

EDAベンダとして早くから組み込みソフトウェアに参入 ARMアーキテクチャへの造詣深く ハード開発とソフト開発を ワンストップで提供

多くの組み込みシステムが高機能化や複雑化するなか、SoC (System-on-a-chip) などのカスタムチップの搭載が進んでいる。しかしいくらSoCを搭載しても、その上で動作するソフトウェアによっては思ったような性能を得られないことが多い。SoCというハードウェアと、ドライバやミドルウェア、アプリケーションなどソフトウェアの両輪が揃ってはじめて競争力ある製品を開発することができるからだ。メンター・グラフィックスは、EDA (Electronic Design Automation) ベンダながら、早くから組み込みソフトウェア業界に参入しており、ARMアーキテクチャへの造詣も深い。ここでは、ハードウェア開発ツールとソフトウェア開発ツールをワンストップで提供するメンター・グラフィックスの魅力聞いた。

全体をアーキテクトできる 人材の不足

組み込みシステムは、高度化、複雑化が進む一方、短期開発や低コスト開発などが要求されるなど、その開発環境は困難さを増している。従来、組み込みシステムは、ハードウェアとし

てマイコンやDSP (Digital Signal Processing) などの汎用的なチップを搭載し、多くの処理をソフトウェアで行うものが多かったが、システムの高度化や複雑化が進むにつれ、ASIC (Application Specific Integrated Circuit) やSoCなどのカスタムチップへのニーズが高まってきている。

しかし、SoCなどの開発には、多くのリスクが潜んでいる。特に限られた消費電力で高い性能を得るためのマルチコア化が進んでいるなか、性能解析の役割が重要になってきている。ある調査によると、2010年の組み込みシステムでの平均的な使用コア数は、1.9個とマルチコア化が着実に進んできている。処理を分散化でき、クロック数を上げなくても必要な処理が行えることで、システム全体の低消費電力化を図れるからだ。

「最適なパワーやコストなどを得られる最適化設計をしっかりと行わないと、工程が進んでから大きな手戻りが発生する恐れがあります」(牧野氏)。この手戻りは、大きなコスト負担をはじめ納期の遅れをもたらす。短期開発や開発コストの低減が求められるなか、プロジェクトそのものの行く末を危うくする要因ともなりかねない。

そこまでいなくても、ハードウェアのバグをソフトウェアで吸収するといったことは日常的に行われていることだ。「理想的には、ハードウェアはバグレスで開発しないとダメです」(牧野氏)。



メンター・グラフィックス・ジャパン株式会社
デザイン・クリエイション&シンセシス部
ビジネス・ディベロップメント・マネージャー

牧野 潔氏

「ハードウェア設計者とソフトウェア設計者が、うまく連携を取りながら開発することが大切ですが、全体をアーキテクトできる人材が不足しているのが現状です」(三橋氏) という。メンター・グラフィックスは、ハードウェアとソフトウェアのツールを提供しており、組み込みシステム開発をワンストップでサポートしている (図1)。

どう作るから 「何を創るか」の時代へ

牧野氏は、大きな手戻りになる要因として、①期待する動作ができていないこと、②基本的な機能や性能がでないこと、などが挙げられるという。

①の期待する動作ができていないことは、いわゆるバグであることが多い。前述のようにハードウェアのバグは大きな手戻りの要因になる。

②の基本的な機能や性能がでない要因はいくつかあるが、バス設計の不備もよくあることだろう。バス幅を広くすればスループット

は向上するが、その一方でダイサイズ (チップ面積) が大きくなってしまふ。ダイサイズはコストに直結するため、できるだけ小さくしたい。スループットとコストという相反する課題の最適化を図ることが求められる。

また、製品の差異化を図るための機能を搭載することもポイントとなる。「ネットワークに繋がるなどの基本的な機能は搭載して当たり前ですが、搭載したアプリケーションをいかにうまく快適に動作させるかも大切となります」(三橋氏) という。

「一般的に日本のメーカーは、どう作るかは得意なのですが、何を創るかは弱いことが多いようです。システムが大規模化するなか、職人芸的な動に頼る開発では、それらに対応することが難しくなっています」(牧野氏)。

メンター・グラフィックスでは、ESL (Electronic System Level) ソリューションの Vista™ Architect、機能検証ソリューションの Questa® といった多くのツールを提供している。そのなかで今回は Vista Architect を見てみよう。

より抽象度の高い設計ができる Vista Architect

Vista Architect は、TLM (Transaction Level Modeling) 2.0 ベースのアーキテクチャ設計/解析/検証ソリューションであり、システムが実現可能か否かを見定めることができるものだ (図2)。

半導体開発は、回路図を CAD (Computer Aided Design) ツールなどで描いて作成する方式から、ソフトウェアの様にプログラミングで開発を行うトップダウン方式へと移行した。そこで用いられたのが、VHDL や Verilog などの HDL (Hardware Description Language) である。HDL によって RTL (Register Transfer Level) と呼ぶ論理回路を記述し、論理合成ツールで論理回路を生成していた。

TLM とは、RTL より抽象度の高いレベルの記述ができるものである。より抽象度を上げた上流設計により、少ない記述量で済むなど効率的な設計が可能となる。このレベルの設計環境を ESL と呼ぶ。



図1: メンター・グラフィックスでは、ハードウェア開発ツールとソフトウェア開発ツールを併せて提供している

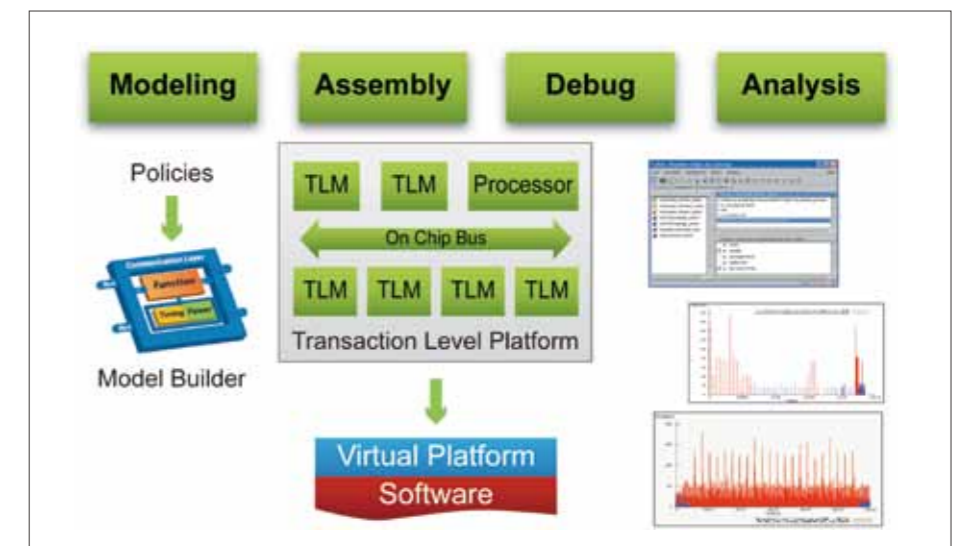


図2: TLM 2.0 ベースでより抽象度の高いハードウェア開発に対応した「Vista Architect」



メンター・グラフィックス・ジャパン株式会社
マーケティング部
ディレクター

三橋 明城 氏

TLMによる標準化が行われる前は、モデル間の互換性がなかったことから、モデル開発のコストを吸収することが難しかった。

ESL言語として、SystemC、SystemVerilog、SpecCなどがある。各言語はそれぞれ特長を持っているが近年はSystemCユーザが増えていく。

SystemCは、C++のクラスライブラリであり、ソフトウェア設計者にも馴染みやすい。SystemCは、RTLレベルの記述も可能であるが、より上流のTLMレベルの記述もできる。すなわち、レジスタ間のタイミングを考慮せず、アーキテクチャ設計に専念できるというわけだ。

ソフトウェアの先行開発とパフォーマンス解析が並行してできる

「Vista Architectの目的として、ソフトウェアの先行開発と性能解析が挙げられます」(牧野氏)という。

従来ソフトウェアは、サンプルチップが完成するまで待たされることが多かった。そのサンプルチップにハードウェアのバグが見つかった場合、前述のように大きな手戻りの要因になってしまう。

「特にマルチコアが進んだ現在では、より早期のソフトウェア開発や性能解析が求められます」(三橋氏)というように、ハードウェアとソフトウェアの双方でのデバッグやチュー

ニングが課題となる。

市販のマイコンではシミュレーション環境を利用できるが、カスタムチップだとそうはいかない。そこで求められるのが、高速なハードウェア・シミュレーション環境だ。以前はHDLモデルを用いていたことから、現実的な速度での検証はできなかった。

Vista Architectは、前述のとおり、TLM 2.0ベースのアーキテクチャ設計/解析/検証ソリューション環境であり、より上流での高速な検証が可能となる。SoCにおけるドライバ周りなど、よりハードウェアに近いソフトウェアをシミュレータで開発するには、高速のモデルが必須となる。Vista Architectは、そのためのツールである。

高速化のためには、モデルの抽象度を上げることがポイントだろう。「よくC設計といわれますが、Cは抽象度を上げるための手段であり、はじめからCありきではありません」(牧野氏)。

システムの性能は、バス帯域やバスへ繋がるモデルなどによって変わってしまう。Vista Architectは、これらのタイミングを中心とした性能解析ができる。

TLM 2.0には、LT (Loosely Timed) とAT (Approximately Timed) と呼ぶ2つのタイミングモードがある。LTはタイミング精度が粗く、ATはタイミング精度が比較的細かいモードであり、ソフトウェア先行開発にはLTを用い、性能解析にはATを用いる。

「Vista Architectは、LTとATを動的に切り替えることができます。これにより、OSをブー

トするまでLTで高速に走らせ、その後ATに切り替えて、詳細な性能解析をするなど、効率的なソフトウェア開発環境を実現できます(図3)」(牧野氏)。

LTとATを動的に切り替えられるツールは、Vista Architectだけだという。「LTを効果的に用いることでソフトウェアのチューニングも先行して実施できます」(三橋氏)という。

豊富な組み込みソフトウェア開発ソリューションを持つ

次にメンター・グラフィックスの組み込みソフトウェア開発ソリューションを見てみよう。

メンター・グラフィックスは、1995年にリアルタイムOS (RTOS) のVRTXやMicrotecコンパイラ、XRAYデバッガなど持っていたマイクロテックリサーチを買収したのを皮切りに、組み込みソフトウェア業界へ参入した。メンター・グラフィックス本体としても、すでに17年の実績を持つことになる。

その後2002年には、携帯電話など多くのデバイスへの搭載実績があるNucleus® RTOSなどを持ったアクセラレイテッド・テクノロジーを買収した。「Nucleus RTOSは、ロイヤリティ不要、世界的なサポート体制も備えているなど多くの特長を持っており、全世界で23億台以上もの携帯電話に採用されています」(三橋氏)。

さらに、ユーザインタフェース開発ツールであるInflexion® UIの提供、Embedded Alleyの買収により、Android™やLinux®などへの対

応を進めてきた。また、オープンソースの開発ツール群を提供してきたCodeSourceryを買収するなど、組み込みソフトウェア製品のラインアップ強化を図っている。

ここでは、多くの組み込みソフトウェア開発ツールのなかから、Sourcery™ CodeBenchとInflexion UIを見てみよう。

GNU標準化の方向性を決定する著名なエンジニアが多く在籍

Sourcery CodeBenchは、GNUツールチェーン(GCC)ベースのC/C++開発環境であり、Eclipseをフレームワークとして機能を拡張できるものだ(図4)。

Sourcery CodeBenchを開発したCodeSourceryは、プロセッサベンダなどへのチューニングサービスをビジネスにしてきた高いスキルを備えており、メンター・グラフィックスになってからも同様のビジネスを継続している。「CodeSourceryは、GNU標準化の方向性決定などに関わる著名なエンジニアが多く、そのエンジニアをメンター・グラフィックスが抱えることで、組み込み業界に対して多彩なソリューションを提供できるのも強みのひとつです」(三橋氏)。

Sourcery CodeBenchは、Vista Architectと連携する。ソフトウェアエンジニアは、ターゲットが実ボードでもVista Architectのモデルでも、同じように扱うことが可能となり、仮想モデルと実機間をシームレスに移行したソフトウェア開発ができる。

「Sourcery CodeBenchは、さまざまな機能を備えたツールとしてワールドワイドでの高い実績があります」(牧野氏)。

組み込みシステムのユーザインタフェースを簡単に作成

Inflexion UIは、組み込みシステムのユーザインタフェースを簡単に作成できるツールである(図5)。

ユーザインタフェースは、システムの差異化に大きなポイントとなる部分である。ネット接続などの基本機能では、差異化を図ることはできないからだ。

ユーザインタフェースでは、意匠が大切となることから、デザイナーが手軽に使えるツールが求められている。「Inflexion UIは、アニメーションなどの動きを含めたグラフィックスをデザイナーが手軽に作れるのが特長です。さらに、ランタイムエンジンまで自動生成されることがInflexion UIの最大の特長です」(三橋氏)。

いままでは、ユーザインタフェースのデザインを受けて、ソフトウェアエンジニアがそれを表現していた。「デザイナーとソフトウェアエンジニアの意思疎通の度合いによって、作り直しなどに時間が掛かっていたものが、Inflexion UIを用いることで短縮できることがメリットです」(三橋氏)。

さらに、Vista Architectとの連携も可能であり、システム開発の早い段階からユーザインタフェースを検証することもできる。

今後も組み込み市場へのアプローチを強めていく

ARM社との関係は、ARM7の頃からだという。「ARMとの付き合いは、15年を超えており、非常に長くお付き合いいただいています」(三橋氏)という。

現在、メンター・グラフィックスではCortex-M3とA9のユーザ数が増えており、さらにCortex-R4やR5も出てきている。またCortex-A7やA15についても、Vista Architectに統合される形で提供されているという。

「Vista Architectは、ハードウェアプラットフォームを作成するエンジニアに向けたものです。今後は、低コスト化を図った派生製品を充実させ、組み込みソフトウェアエンジニアが使いやすいようにしていきます」(牧野氏)。

「組み込みシステムは、ソフトウェア開発のコストが圧倒的に大きくなっています。メンター・グラフィックスは、その改善に向けたツールをさらに拡充させていきます」(三橋氏)。

組み込み市場へのアプローチを強めていくのが、メンター・グラフィックスの会社としての方針である。その一環として、Sourcery CodeBenchとVista Architectの開発メンバーと一緒に動き始めているという。

最後に牧野氏は「組み込みシステムの開発は、ハードウェアとソフトウェアの各ツールをワンストップ化していかないと、今後は成り立ちません。メンター・グラフィックスは、お客様のご要望に合わせたツールを併せてサポートしていきます」と結んだ。

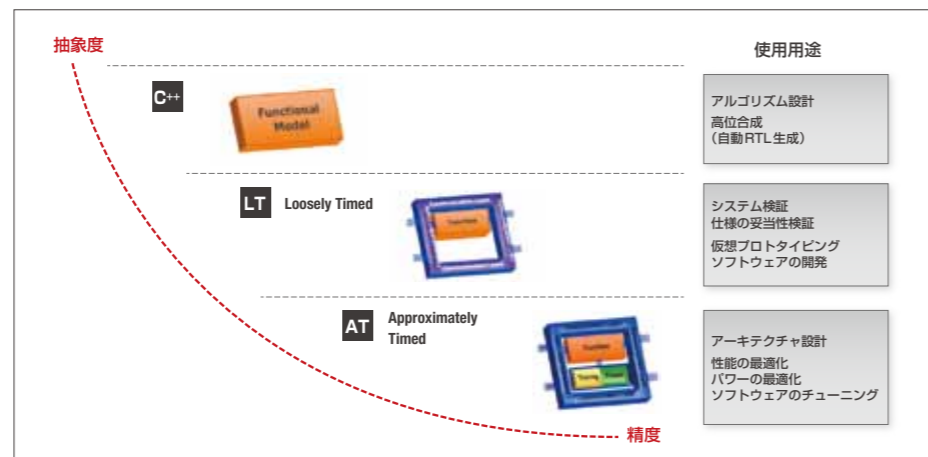


図3: スケラブル・モデリング・メソッド

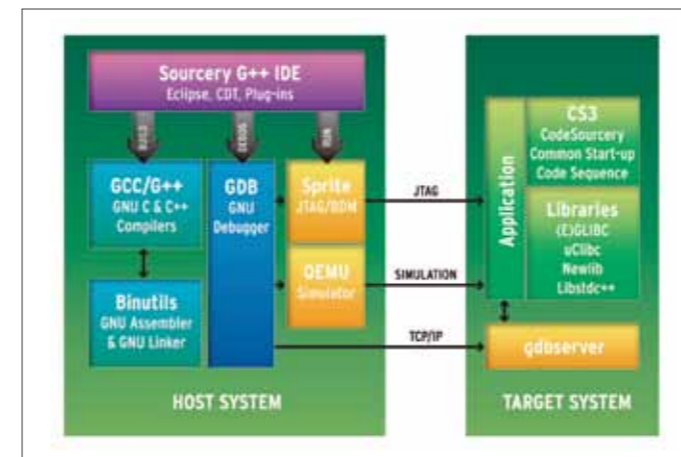


図4: GNUベースの包括的なソフトウェア開発環境である「Sourcery CodeBench」

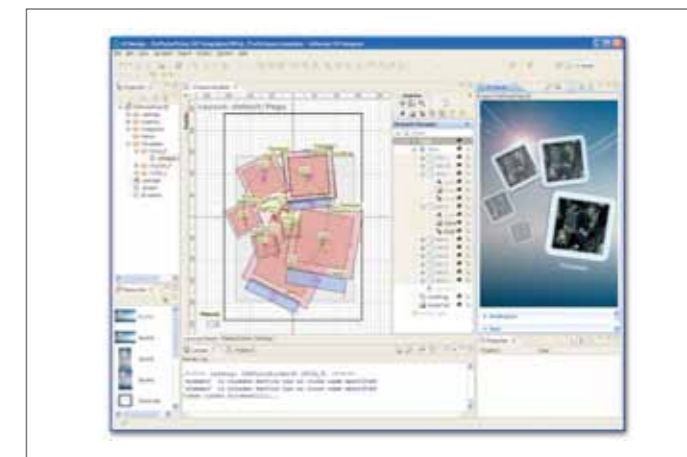


図5: 差異化に直結するユーザインタフェースを簡単に開発できる「Inflexion UI」