

画像LSIのハードウェア/ソフトウェア 協調設計

Hardware–Software Codesign for Graphic LSIs

● 古手川博久 ● 蓮實直信

あらまし

デジタルカメラなどに搭載される画像LSIは高機能、高性能化が最も著しい分野の一つである。このようなLSIの開発においてチップコストを抑え、チップ面積と消費電力の低減を実現するためには、アプリケーションに適したアーキテクチャ設計が重要である。このアーキテクチャ設計を行うための手法としてESL(Electronic System Level)によるハードウェア/ソフトウェア(HW/SW)の協調設計が半導体ベンダやEDAベンダより提案されてきたが、モデルの開発コストなど様々な課題により実際の開発現場にはあまり浸透してこなかった。このような状況の中、富士通セミコンダクターでは、ASSPやASIC開発のお客様向けデザインサービスであるCedarサービスにおいて、ESLにおけるHW/SW協調検証によるアーキテクチャ設計技術をLSI開発に地道に適用し、オンチップバスのQoSとDDRメモリなどの外部メモリへのアクセス最適化が最も重要なポイントであることを見出した。そして、その最適化ポイントに着目しESLの課題に対する対策を検討した結果、ESLを使ったHW/SWの新しい協調設計ソリューションを構築でき、これまで以上に設計現場に適用されるようになってきた。

本稿では、この新しいアプローチの技術内容と適用効果を紹介するとともに、今後の展開についても説明する。

Abstract

The field of image processing LSIs that are mounted in devices such as digital cameras is one in which the increase in performance is very remarkable. It is important to have an architecture and design that is suitable for the application so as to lower the cost of developing chips for LSIs, and reduce their power consumption and the chip area. Semiconductor and EDA vendors have proposed hardware–software (HW/SW) codesign via Electronic System Level (ESL) as a way to develop chip architecture and designs. However, it has not been used much in actual development sites because of various issues such as the cost of developing models. Under these circumstances, at Fujitsu Semiconductor we found that in Cedar service, which is a design service for customers to develop ASIC, ASSP, and such like, the most important point is to steadily apply architecture and design technology to LSI development. This can be done by conducting HW/SW coverification in ESL to optimize access to QoS of On Chip Bus and external memory such as DDR. And, as a result of focusing on this point of optimization and investigating ways to tackle the issues with ESL, we have created a new HW/SW codesign solution that uses ESL. It has come to be used in design sites more than ever before. This paper introduces the technical aspects of this new approach and its effect, and also describes future developments.

まえがき

デジタルカメラをはじめとする画像LSI⁽¹⁾を搭載した製品のカタログで、CCDセンサやCMOSセンサの有効画素数が10 Mpixelなどと書かれているのを目にすることがよくある。この画素数は年々増加しており、一眼レフの最高位機種では、30 Gpixelを超えるものまで出てきている。また、様々な画像補正、画像効果、画像圧縮伸張など画像処理そのものが複雑化するだけでなく、Android OSを使用したグラフィカルなユーザインタフェースや様々な外部インタフェースなど機能も複雑になってきている。このため、画像LSIを構成するCPUコアやIPの高機能・高性能化およびRAM容量の増加による回路規模の増大、ひいてはLSIの電力の増大につながってきている。

このような回路規模、電力の増大に対し、デジタルカメラのようなモバイル製品は、バッテリー寿命やパッケージコストの観点などでチップ面積の最小化、低消費電力化といった、回路規模とはトレードオフの関係にある要求も満たさなければならぬ。これまで、このような要求に対しては、集積化、低電圧化技術により対応してきたが、最近の最先端のプロセステクノロジーではムーアの法則への追従が困難になりつつあるとともに、低電圧化の限界およびリーク電流、配線容量などの増大で低消費電力化についても改善が難しくなっている。そのため、上流工程であるアーキテクチャ設計段階でシステムのユースケースに則った性能、消費電力、チップ面積の最適化ができていのかどうかを判断するためのアーキテクチャ評価が重要となってきた。

しかし、上流でのアーキテクチャ評価は、実際の設計現場にはあまり浸透していないのが現実である。その原因として上流でアーキテクチャ評価を行う環境を構築するのが困難であることに加え、開発フローとしてアーキテクチャ評価が行えるようになっていないからである。一般的な画像LSIは、CPUコアやGPU (Graphics Processing Unit)、DSP (Digital Signal Processor) など様々なIP (Intellectual Property) を一つのチップに搭載したSoC (System-on-a-Chip) として開発されている。そのため、ソフトウェアを搭載して初めて画像処

理システムとしての動作が実現されている。よって、システムとしての性能や消費電力を評価するためには、SoCに加えソフトウェアが必要となってくる。しかし、現状の開発フローにおいてソフトウェアを開発するのは、SoCのES (Engineering Sample) ができた後に実機ボードで行われているのが一般的であり、アーキテクチャ設計段階でソフトウェアは存在しない。そのため、上流工程でSoCとしてのアーキテクチャ評価ができない。

富士通セミコンダクターでは、このアーキテクチャ評価ができない課題に対してESL (Electronic System Level) の技術を活用したハードウェア/ソフトウェア (HW/SW) 協調設計フローを構築した。

本稿では、このHW/SW協調設計について説明する。

ESLの概要と課題

まず、本題に入る前に、ESLおよびモデルの抽象度について説明するとともに、これまでESLの適用を阻害していた課題について説明する。

● ESLとは

ESLは、C、C++をベースとしたSystemCなどのハードウェア記述言語によりSoCをモデリングして、ソフトウェア先行開発やアーキテクチャ評価を行うコンピュータ上で仮想的にシミュレーションを行う環境のことである。⁽²⁾ EDA (Electronic Design Automation) 業界では、SystemCなどの言語で書かれたハードウェア記述を高位合成によってRTL (Register Transfer Level) を生成することもESLと言う場合があるが、本稿では、前者のシミュレーション環境のことを指す。

● モデルの抽象度

このように、ESLは、ソフトウェア先行開発とアーキテクチャ評価を行うという二つの目的に利用されるが、それぞれの目的に応じてモデリング方法が違ってくる。

ソフトウェア開発を行う場合は、より実機に近いシミュレーション性能を要求されるため、抽象度の高いモデリングをしなければならない。次に、アーキテクチャ評価を行う場合は、データの読み書きを実際のハードウェアの通信プロトコルに近い形で行う必要があり、抽象度の低いモデリングが必要となる。その用途に応じたモデリング

方法をSystemCの標準化団体であるOSCI (Open SystemC Initiative) がTLM (Transaction Level Modeling) として標準化している。TLMでは、ソフトウェア開発に適した抽象度の高いモデルをLT (Loosely-Timed) モデルと呼び、アーキテクチャ評価に適した抽象度の低いモデルをAT (Approximately-Timed) モデルと呼ぶ。

● ESLの課題

このように、ESLの環境は用途に応じてLT, ATの二つの抽象度の環境を準備しなければならないため環境構築にはそれなりの工数とコストが必要になってくる。また、アーキテクチャ評価に使う抽象度の低いモデルの開発はRTL開発と同等の工数が必要であり、高位合成などでRTLを生成できない場合、2重開発のように見えてしまう。SoCの開発においては、RTLが既に存在するIPを使用することが多いことから、この課題が顕著となる。更に、他社から導入したIPやレガシーなIPの場合、モデリングに必要な仕様が不明瞭であり、実質的にモデリングが不可能な場合が多い。そのため、実際の設計現場への導入障壁が高くなり受け入れられなかったのである。

アーキテクチャ設計の評価ポイント

前章で述べた課題を克服するために、富士通セミコンダクターは、これまでの経験を生かしアーキテクチャ設計を行う上で重要な評価ポイントを整理し対策を講じた。本章では、その評価ポイントについて述べる。

これまでのLSIの開発において、IP単体では要求

性能を満たしているが、SoCとして組み上げた後要求性能を満たせないという障害が発生することが多々あった。

そのような障害に対し、エミュレーションなどで問題が発生する条件を再現させながら分析した結果、IPに関連するオンチップバス、DDRなどの外部メモリコントローラが、最適にデータをさばけるようになっていなかったり、データフロー的に競合する別のIPが邪魔をしてしまったりすることが性能向上の妨げにつながっていると分かった。つまり、オンチップバスやDDRなどの外部メモリコントローラがSoCのデータ処理として最適化できていなかったことが性能向上の妨げの原因であり、性能改善を行う上でその最適化が重要であるということが見えてきたのである。

この結果を踏まえ、現在富士通セミコンダクターの画像系LSIのオンチップバスとして多く使われているAXI (AMBA3) のインターコネクトバスと、外部メモリコントローラの中でも様々なコマンド制御によりRead/Writeレイテンシの最適化を行う必要のあるDDRメモリコントローラを想定して具体的な評価ポイントを整理しまとめたのが表-1である。

HW/SW協調設計フロー

表-1では、評価ポイントとそれを評価するために必要な設計環境の抽象度や入力情報についてもまとめた。これを考慮して新たに構築したHW/SW協調設計フローを図-1に示す。本章では、その設計フローを構成する各環境について説明する。

表-1 性能評価の評価ポイント

評価ポイント		必要条件	抽象度
オンチップバス	トポロジー	システム動作	AT
	動作周波数		
	FIFO段数	データフロー	CA
	アービトレーション		
	プライオリティ制御		
DDRメモリコントローラ	ポート数	システム動作	AT
	動作周波数		
	FIFO段数	データフロー	CA
	アービトレーション		
	コマンド制御		

まず、表-1に示した設計環境の抽象度を見ると、これまでESLでのモデル抽象度ではあまり耳にしなかったCA (Cycle Accurate) という抽象度が示されている。この抽象度は、RTLと同等の抽象度でATより更に低い抽象度であり実際のハードウェアの動作をより忠実に表現する。これまでの富士通セミコンダクターのアーキテクチャ評価の経験から、AXIのリクエスト要求が競合したときのバスにおけるアービトレーションや、DDRメモリコントローラにおけるメモリチャネルの帯域幅を有効に使うためのコマンド制御（リフレッシュ制御、プリチャージ隠蔽のためのオーダリング制御）などは、CAの抽象度でないとな実際の動作と大きく異なり、この動作の違いがアーキテクチャ評価結果を大きく歪ませる原因となることが分かっている。

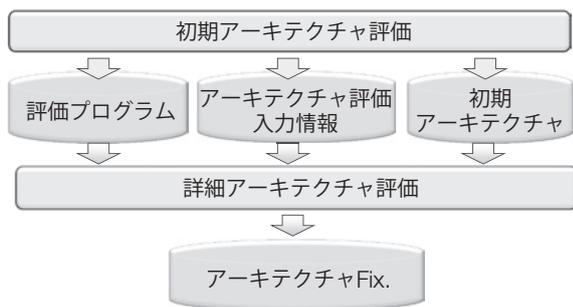


図-1 HW/SW協調設計フロー

ただし、表-1の各評価ポイントでの設計環境の抽象度は、全てCAの抽象度でなくてはならないというわけではない。前で述べたように抽象度が低くなるとシミュレーション速度が遅くなることからSoCの全構成に近い形でソフトウェアを実行するのは不可能である。そこで、ATの抽象度でベースのアーキテクチャの決定と次の評価を行うために必要な情報を取得する「初期アーキテクチャ評価環境」とCAの抽象度で最終的にアーキテクチャを決定する「詳細アーキテクチャ評価環境」の二つの評価環境を組み合わせる設計フローを構築した。

● 初期アーキテクチャ評価環境

初期アーキテクチャ評価環境は、表-1のATの抽象度で評価しなければならない評価ポイントの評価する環境である。図-2に示したとおりこの環境の特徴は、周辺ペリフェラルなどの性能評価であり重要ではないが、システムとして動作させるために必要なハードウェアは、LT抽象度のモデルで構成し、性能評価に重要なCPUコアやAXIバスなどのハードウェアのみAT抽象度のモデルとして構成した。また、GPUなどの社外から導入したIPやお客様のロジックなど、富士通セミコンダクターでモデリングできないものはFPGAにマッピングし、FPGAとPC上のESL間には高速に接続できるSCE-MIの技術を利用した。このように、シミュレ

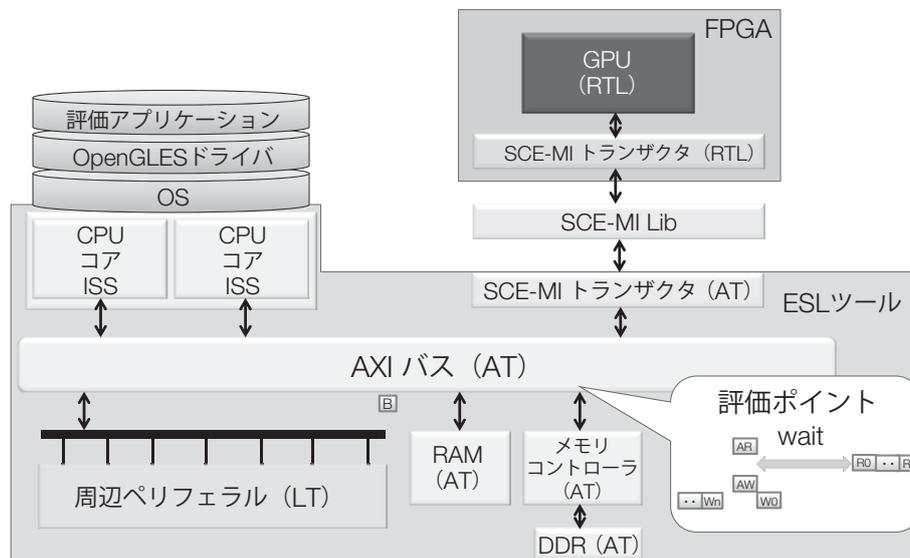


図-2 初期アーキテクチャ評価環境

ション性能を劣化させるAT抽象度のモデルを必要最小限にし、モデリングが困難なIPをFPGAにマッピングし接続することで、アプリケーションに近い評価用ソフトウェアを使いシステムのアーキテクチャ評価をシミュレーション性能ならびに開発工数の観点でリーズナブルに行えるようにした。

この環境では、AXIのトポロジーや、DDRに必要なメモリ容量、DDRメモリコントローラのインタフェース仕様など詳細アーキテクチャのベースを決めるとともに、詳細アーキテクチャ評価に必要な、性能評価のポイントとなる性能ボトルネック箇所を決定し、そのボトルネックを評価するのに必要なデータフローや、それに関連するマスターのトランザクション動作など、詳細アーキテクチャ評価環境での評価に必要な情報を取得できる。

● 詳細アーキテクチャ評価環境

初期アーキテクチャ評価環境で評価した結果、性能ボトルネックとそれに関連するデータフローから重要なマスターのIPとスレーブのIPを見出し、そのIPのRTLを、CAの抽象度を保ったままCモデルに変換するCarbonizeという技術を持ったEDAツールで変換する。この変換されたCモデルでRTLより5～10倍高速なシミュレーション環境を構築することができる。しかし、回路規模やマイクロアーキテクチャ的にCモデルへの変換が困難な場合や、そのIPを実際に動作させボトルネックとなる

データフローを実現するために、実際のアプリケーション動作の実時間として数十秒かかるものは、RTLより5～10倍高速とはいえ実際の運用には耐えられない。そのような場合は、図-3に示した汎用トランザクション生成モデルを活用することで解決する。この汎用トランザクション生成モデルは、トランザクション生成の制御をC言語で柔軟に行うことができるため、ボトルネックとなるトランザクションを簡単に生成することができ、実際のIPを接続するより簡単にシステムとしての性能ボトルネックの状態を表現できる。

この詳細アーキテクチャ評価環境で評価することにより、AXIバスのFIFO段数、アービトレーションの決定、DDRメモリコントローラのコマンドキュー段数の調整やコマンド制御方式の決定などの最終調整をRTLの全体が組み上がる前に短期間に行うことが可能となった。

適用効果

この新しいESLの技術を活用したHW/SW協調設計フローにより、これまでエミュレーションや実機評価といったフェーズでしか検出できなかった性能問題をアーキテクチャ設計段階で検出できるようになった。

AXIなどのプロトコルでは、リクエストとデータが突き放した制御がされているため、各単発のト

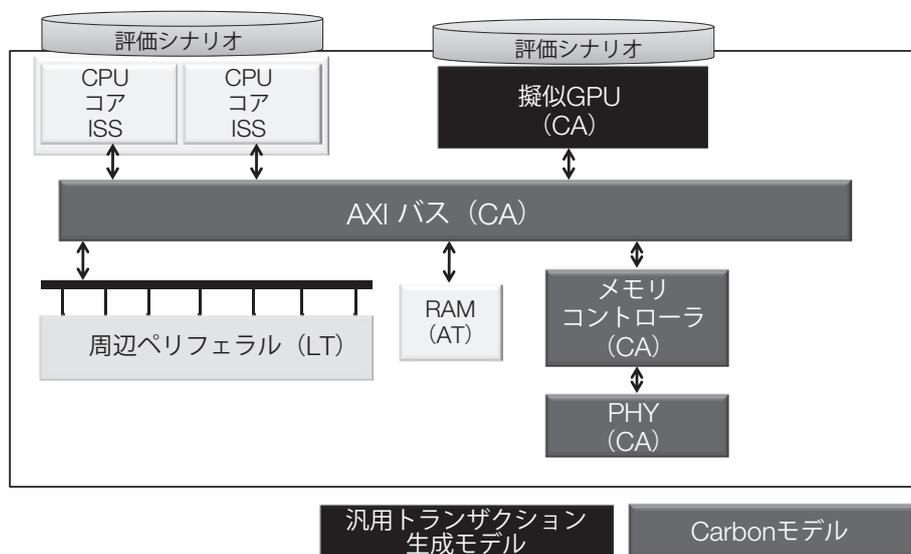


図-3 詳細アーキテクチャ評価環境

ランザクションによるスループット・レイテンシーの確認のみでは問題は見えずらく、画像処理のように複数画像フレームの処理が積み重なって初めて性能問題として顕著化してくる場合がある。このような問題に対して、初期アーキテクチャ評価により実時間で10秒程度の長い時間のシミュレーションを行い、複数フレーム処理としての問題傾向をつかみ、その問題箇所を詳細アーキテクチャ評価環境でシミュレーションすることで、性能問題を起している本質的原因をアーキテクチャ仕様策定段階で短時間に見つけ回避できるようになった。

具体的な事例として、レスポンスは悪いが、IP単体で見たときは性能面で問題にならないレガシーIPに対し、そのIPとデータフローとして競合する大量のデータ処理を行うIPを動作させたところ、時間経過とともにレスポンスが悪い点が徐々に影響し、最終的に性能劣化に至るといった問題を検出し、仕様段階でこの性能問題を解決し一発完動に導いたケースがある。

今後の課題

上記のように、具体的成果を出し始めたESLの技術を活用したHW/SW協調設計フローだが、現時点ではまだ次のような課題がある。

(1) 高速インタフェース系IPの外界のモデリング

USB, PCI Expressなどのような高速インタフェース系のマクロは、SoCの外の世界と様々なユースケースで接続される。そのため、そのユースケースを考慮した外界をモデリングすることは不可能に近い。現時点では、そのSoCの使われ方のワーストケースとなるパターンを想定し、前述の汎用ランザクション生成モデルにより模擬して

いる。しかし、現実のユースケースよりランザクション量を多くし過ぎることによる性能の過多、回路規模の増大や消費電力の増大につながってしまったり、逆にランザクション量を少なくし過ぎてしまうと性能劣化を引き起こしてしまったりすることにつながる懸念される。

(2) 低消費電力設計へのフィードバック

今回紹介したHW/SW協調設計フローによるアーキテクチャ設計の最適化により、無駄な性能を出さないという観点では低消費電力化につながると考えているが、実際に電力値としてどれだけ最適化されたかの定量的な可視化、またRTL実装やレイアウト設計などにつなげる設計データとしてのフィードバックができていない。

今後は、上記の二つの課題を解決するためのHW/SW協調設計フローの改善を行っていきたい。

む す び

本稿で説明したHW/SW協調設計フローは、画像LSIだけではなく、様々なASSP, ASICに適用されるようになってきた。今後も、ASICのお客様や富士通セミコンダクターの設計者からの貴重なフィードバックを生かし、より良い上流設計フローを構築していきたいと考えている。

参考文献

- (1) 金子浩美：デジタルカメラから携帯電話までを陰で支える画像処理LSI。
<http://i.impressrd.jp/files/images/bn/pdf/im200602-108-solution.pdf>
- (2) 大村正之ほか：C/C++によるVLSI設計。共立出版、2003。

著者紹介



古手川博久 (こてがわ ひろひさ)

富士通セミコンダクター (株)
開発本部SoCソリューション統括部第一設計技術部 所属
現在、上流設計検証環境開発、および上流設計検証支援に従事。



蓮實直信 (はすみ なおのぶ)

富士通セミコンダクター (株)
開発本部SoCソリューション統括部第一設計技術部 所属
現在、上流設計検証環境開発、および上流設計検証支援に従事。