

プロローグ 開発手法のパラダイム・シフト

ARMコアFPGAを使って完全C/C++言語で設計する時代が到来

丸山 としひろ

Toshihiro Maruyama

来栖川 智久

Tomohisa Kurusugawa

横山 雅一

Masakazu Yokoyama

1. SDSoCならアルゴリズム開発に専念できる

SDSoCは、ARMプロセッサとFPGAを1チップに集積しているZynqというデバイスをターゲットとした統合開発環境です。

C/C++アプリケーション・プログラムの最適化コンパイラを備えています。プロファイル機能でソフトウェア・アプリケーションのボトルネックを特定できます。また、各種見積もり機能があり、ハードウェア化した場合の効果を事前に確認できます。

アクセラレータ開発(ソフトウェアのハードウェア化)に従来必要だった作業を、ツールが自動的に行ってくれます。ユーザはC/C++レベルでのアルゴリズム開発に専念できるという大きなメリットがあります。

例えば、以下のようなことは、全てSDSoCが自動的に行ってくれます。

- Cコードからの高位合成
 - メモリとアクセラレータ間のデータ転送を行うデータ・ムーバの追加
 - 各ハードウェア・モジュール間の接続
 - アクセラレータ、データ・ムーバのドライバ作成
- ターゲットとなるプログラムはLinux, FreeRTOS, ベアメタルのアプリケーション・プログラムです。

プロファイリング機能として、ソフトウェア処理時間、ハードウェア処理時間、リソース利用率、消費電力の見積もりなど統合的なパフォーマンス解析が可能です。

(丸山 としひろ)

2. FPGA設計者の視点…SDSoCはハードウェア技術者にとってのフロンティア

ソフトウェアの開発環境と比べて、著しく貧しいのが、ハードウェア設計環境です。集約すると、2つの課題があると筆者は考えます。

- 課題1: アルゴリズムのハードウェア化に工数がかかりすぎる

FPGAをシステムを中心に据え置いたシステムの開発では、アプリケーションの要求が固まると、まず、ソフトウェアでアルゴリズムを検証します。その後、

検証済みのアルゴリズムをハードウェア化するのですが、この作業には、ソフトウェア開発の何倍も、場合によっては何十倍もの工数が必要でした。例えば、以下のような理由があります。

- プログラミング言語のコードをVHDLやVerilog HDLといったハードウェア設計言語(HDL)に変換しなければならない。その際、情報処理系のアルゴリズムは、そのままハードウェア化しにくい場合が多い。
- Cプログラムではいとも簡単にfopen()やprintf()などの標準関数を用いることができる。HDLには標準的な入出力ライブラリは存在せず、全て自前でそろえなければならない。
- HDL設計では、クロック単位の動作を考えなければならない。動作クロックが上がらない場合にはパイプライン化したり、内部回路を書き換えて遅延を解消する工夫を行う。シミュレーションによる検証には多大な時間を要する。

- 課題2: 分業体制では待ち時間が長い

ハードウェア設計者とソフトウェア設計者が分かれている場合、おのおのの機能開発を行うため、開発工程で幾つかのすり合わせや待ち時間が発生します(図1)。

- ハードウェア設計者によるFPGA開発が終わるまで、ソフトウェア設計者が開発を待たされる。
- ソフトウェア開発の際に不具合が見つかったら、ハードウェア開発者にフィードバックを行うことになるが、アルゴリズム検証からやり直しになるため、多大な時間を要することになる。

最終製品であるハードウェアの完成は数週間から数か月を要するものもあり、完成までの期間はソフトウェア開発者は業務を進めることが極めて困難です。

- 開発環境は改善しているものの万全ではない

このような課題に対して「プログラム・コードからHDLの自動生成(高位合成)」や、「FPGA搭載評価ボードとセットのソフトウェア・プラットフォーム(BSP:基本ソフトウェア・パッケージ)」などが提供されています。

筆者も過去にC言語系からの高位合成では4言語ほど使っていました。しかし複数のツール間を行き来する必要があるなど、必ずしも使い勝手が良いとは言えませんでした(図2)。