

# ハード言語不要で、ソフト技術者でも 効率良く組込みFPGA開発 新機能の追加でさらに 使いやすくなったSDSoC開発環境



組込みシステムの中で、FPGAはハードウェアによる高速処理や大規模なカスタム機能の実現に不可欠なデバイスだ。以前は、FPGAの開発はHDLを使いこなす専門のハード技術者、組込みマイコンの開発はソフト技術者が分担して開発したり、チームとして協業することが多かった。しかし、ザイリンクス株式会社(以下ザイリンクス社)のSoCであるZynqのようにプロセッサとFPGAを高度に統合したデバイスが普及するにつれて、ソフト技術者自身でも最適なシステムを開発できる新しい開発環境が重要になっているという。

今回は、ソフト技術者とハード技術者の境界を越えて、幅広いユーザのFPGA活用を可能にするザイリンクス社のSDSoC開発環境と、その最新機能について紹介しよう。



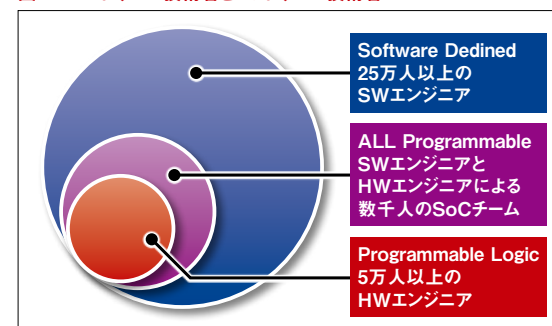
グローバルセールス アンド  
マーケット ツール メソドロジー  
アプリケーション部  
黒田 成一氏  
執筆：宮崎 仁

## 1. ソフト技術者が高性能のハードを開発できる時代になった

マイクロプロセッサは定型的なハードウェアを用いて、ソフトウェアによって任意機能を実現できる。一方、FPGAはハードウェアを自由にプログラミング可能で、高性能で自由度の高い機能を実現できる。ザイリンクス社では、このソフトウェアのプログラマブルとハードウェアのプログラマブルを統合して、あらゆる面でプログラマブルな新しいプラットフォームを実現するAll Programmableというコンセプトを提唱している。ARMコアとFPGAを統合したZynq-7000 All Programmable SoCは、その代表的なデバイスだ。

このAll Programmableを活用するには、専門のハードウェア技術者だけでなく、多くのソフトウェア技術者が協調してシステム開発を行う必要がある。HDLを使いこなすハードウェア技術者に比べて、C/C++に精通したソフトウェア技術者の数ははるかに多い。

図1 ハードウェア技術者とソフトウェア技術者



そのため、ザイリンクス社のFPGA開発環境Vivado Design Suiteでも、最新のHLx EditionではC言語ベースの高位合成(Vivado HLS)機能が標準搭載され、ソフトウェア技術者でもFPGAを使いこなせる環境が整えられている。

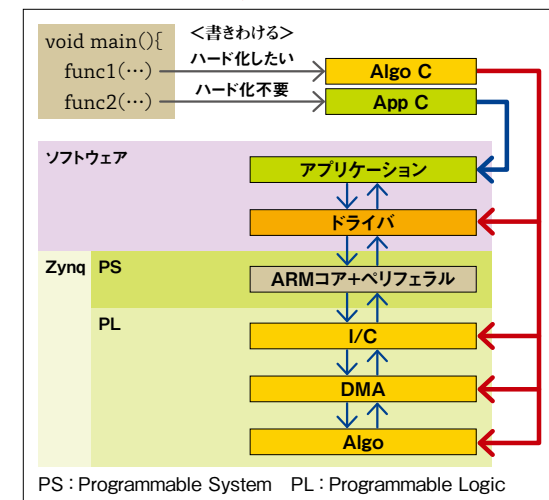
さらに、ザイリンクス社ではそこから一歩進んで、従来はハードウェアで実現していた機能も含めてソフトウェア主導でシステムを定義していくSoftware Definedを推進していこうとしている。プロセッサとFPGAを高度に統合したAll Programmable SoCのための新しい開発環境が、SDSoC(Software Defined SoC)だ。

SDSoCと従来のSoC開発フローとの大きな違いは、機能的にはすべてソフトウェアで定義したうえで、高速処理や高度な並列処理など性能面からハードウェア処理が求められる部分を、関数単位で自動的にハードウェア化するという点だ。

SDSoCは、多くのソフトウェア技術者にとってなじみ深いeclipseベースの統合開発環境だ。C/C++やOpenCLによって普通にソース・コードを記述でき、そのままビルドすれば、所望のプログラムを生成できる。そして、ソース・コードの中で開発者がワンクリックで選択した関数を、FPGA上のハードウェアとして自動的に合成してくれる。関数の呼び出し側から隠蔽された局所的な仮想マシンと見なすこともできる。

さらに、SDSoCはソース・コードに記述された機能をハードウェアとして合成するだけでなく、FPGAとプロセッサ・コアの間のデータ転送回路やDMA機能なども合わせて合成してくれる。また、それらを使用するためのソフトウェア・ドライバも自動生成してく

図2 SDSoCによるハードウェアの自動生成



れる。すなわち、ソフトウェア技術者から見れば、高速処理可能なライブラリ関数を呼び出すのと同じ感覚で、高性能のハードウェアを使いこなすことができる。

SDSoCを活用することによって、ソフトウェア技術者も高性能のAll Programmable SoCを最適に活用していくことができるようになった。

## 2. 新機能の追加でさらに使いやすくなったSDSoC

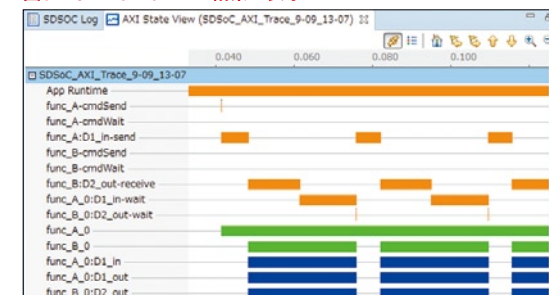
ザイリンクス社では、従来のZynq-7000 All Programmable SoCをさらに進化させた次世代デバイスとして、Zynq UltraScale+ MPSoCの出荷を始めている。それに対応して、開発環境SDSoCもより使いやすく機能強化された新バージョンに進化している。

今回追加された主な新機能としては、最新デバイスであるZynq UltraScale+ MPSoCへの対応、イベント・トレース機能に対応した強力なシステムレベル・プロファイリング・ツール、コンパイル時間の短縮があるという。

システムレベル・プロファイリング・ツールは、開発中のシステムの内部構成をファームウェアやハードウェアのレベルで可視化できる。さらに、ビルド時にハードウェアのイベント・トレース・モジュールを挿入することによって、複数の関数を同時にトレースして、タイムラインで図示することができる。

これを活用すれば、関数の並列化の様子やそれぞれのパフォーマンスを一目で確認できるので、システム

図3 イベント・トレース結果の表示



のボトルネックを迅速に発見できる。さらに、ソース・コードを修正してトレース結果を比較することによって、修正によるパフォーマンス改善の効果も確認することができる。

このシステムレベル・プロファイリング・ツールの活用と、コンパイル時間の短縮によって、開発の生産性は大きく向上するという。

## 3. これからのソフトウェア技術者は並列化の意識が重要

SDSoCを使えば、HDLなどのハードウェア言語を知らないソフトウェア技術者でも、高性能の組み込みシステムを簡単に設計できる。選択した関数だけを高速化してくれる簡単ツールとして、ブラックボックス的に利用することも可能だ。

ただし、実際になるべく効率の良いシステムを開発するためには、ソース・コードの記述時に処理時間がボトルネックになりそうな部分を見つけ、並列化を意識して記述しておくことが重要だ。元のソース・コードが処理を直列的に逐次実行するように記述されていれば、生成されたハードウェアも直列的な回路になり、高速化の効果が上げにくい。

ソース・コードのレベルで、並列実行可能な部分やパイプライン処理が可能な部分を抽出し、それに適した構造で記述を行えば、SDSoCは効率の良い並列ハードウェアを生成してくれる。

従来から、ハードウェア技術者はHDLを学ぶことで、このような並列化や回路アーキテクチャを意識した記述を身に付けてきた。ソフトウェア技術者がそれらを身に付けようとするれば、ハードウェアを一から学ぶことになり、敷居が高かった。

SDSoCではHDLを学ぶ必要はなく、C/C++の範囲で並列処理を実現するハードウェアのアーキテクチャを意識するだけで良い。しかも、新機能のシステムレベル・プロファイリング・ツールを活用すれば、ソース・コードの記述時に意図したことが実際にどれだけ効果を上げられるかを、即座に確認することができる。それを繰り返せば、組み込み開発におけるセンス、スキルの向上も望める。多くのソフトウェア技術者にとって、SDSoCは自らのスキルを磨き仕事の幅を広げるチャンスと言えるだろう。

図4 並列化を意識したソース・コードの構造

