

デジタル信号処理とFPGAの橋渡しをするツールを利用

森田 康

Appendix

ここでは、本特集の設計事例で利用されている米国Xilinx社の「System Generator for DSP」について解説する。System Generator for DSPは、MATLAB/Simulink上でFPGA用の機能マクロを利用したり、SimulinkのモデルをFPGA用のHDLデータやテストベンチに変換するためのツールである。MATLAB/Simulinkのシミュレータ機能と実ハードウェア(FPGA)をリンクさせて、協調シミュレーションを行う機能も備えている。(編集部)

米国Xilinx社のSystem Generator for DSP(以下、System Generator)は、FPGA開発専用の高位設計ツールです。従来の

FPGA設計は、仕様書をもとにHDLでコーディングするところから作業が始まります。一方、System Generatorによる設計では、システムをブロック線図形式で記述するところから始まります。

デジタル信号処理システムや通信システムのブロック線図の入力やシミュレーションには、米国The MathWorks社の「MATLAB/Simulink」がよく利用されています。System GeneratorはこのMATLAB/Simulinkとともに使用します。ユーザから見ると、System GeneratorはMATLAB/Simulinkの拡張機能のように扱えます。

System Generatorの特徴は、ブロック線図のモデルから等価

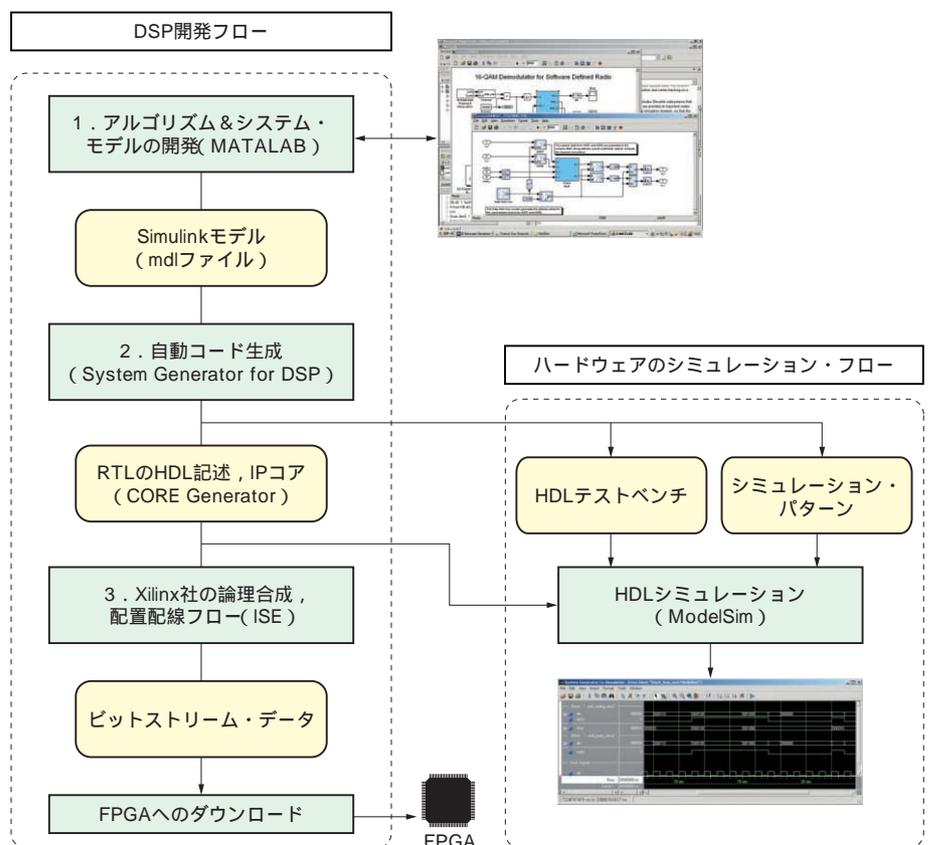


図1 System Generatorによる設計フロー

MATLAB/Simulinkを用いてブロック線図表現でアルゴリズムを記述する。アルゴリズムの動作に問題がないことを確認した後、ブロック線図の構成要素をSystem Generatorが提供するXilinx Blocksetを使ってモデリングする。再度シミュレーションを行い、量子化の悪影響が出ていないかなどをチェックする。動作に問題がなければ、このモデルと等価なHDLコードを生成する。

Appendix



なHDLコードを生成することにあります。このとき、FPGAのハード・マクロやIPコア、そのほかのリソースを有効活用したHDLコードを生成するという特徴があります(下掲のコラム「System GeneratorはVirtex-4搭載XtremeDSPスライスに対応」を参照)。

● MATLAB/SimulinkとFPGA開発環境をリンク

System Generator を利用したFPGA設計のフローを図1に示します。

まず、MATLAB/Simulinkを用いてブロック線図表現でアルゴリズムを記述します。これがリファレンス・モデルとなります。

次に、動作検証によってアルゴリズムの動作に問題がないことを確認した後、FPGAに実装するモデルをSystem Generatorが提供するXilinx Blockset(FPGA向けの機能ブロック・ライブラリ)のブロックで構築していきます。このとき、リファレンス・モデルが浮動小数点モデルであっても、一般にFPGAに実装するモデルは固定小数点で作成する必要があります。そのため、再度シミュレーションを行い、量子化の悪影響が出ていないかなどをチェックします。

固定小数点モデルの動作に問題がなければ、System Generatorの機能を使ってこのモデルと等価なHDLコードを生成します。その際、一部のブロックは「CORE Generator」というXilinx

社のIPコア生成ツールを使って生成します。また、HDLコードを生成するときに、テストベンチを生成することも可能です。

● アルゴリズム検討や実装設計に使われている

System Generatorはデジタル信号処理アルゴリズムの検討に使われる場合もあれば、実際のハードウェア設計(部分設計やシステム全体の設計)に使われる場合もあります(右掲のコラム「Simulinkモデルが『実行可能な仕様書』になる」を参照)。

1) アルゴリズムの検討

アルゴリズムを検討する際には、MATLAB/Simulinkでシステム・レベルのシミュレーションを行った後、System Generatorのリソース評価機能を利用して実装時の回路規模やメモリ容量を見積もります。また、ハードウェア協調シミュレーション(いわゆるコシミュレーション)の機能を備えており、これを利用して検証の効率化を図ることができます。協調シミュレーションとは、MATLAB/Simulinkのシミュレーション機能を外部の実ハードウェア(FPGA)と連携動作させて、アルゴリズムの中の一部のブロックの処理を実ハードウェアに肩代わりさせる手法です。こうすることで、シミュレーションの動作と実ハードウェアの動作を容易に比較できます。また、シミュレーション速度の向上も期待できます(実ハードウェアが一種のエミュレータとして機能する)。



System GeneratorはVirtex-4搭載XtremeDSPに対応

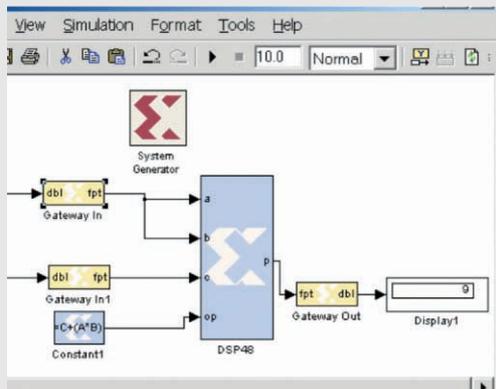
Virtex-4では、デジタル信号処理システムの実装に有効な「XtremeDSPスライス」が搭載されています。このスライスは、I/Oレジスタ、18×18ビットの符号付き乗算器、3入力加減算器、Op-mode付きマルチプレクサを内蔵しています。ここでOp-modeとは、

Xtreme DSPスライスの内部リソース間の接続を動的に切り替える機能(いわば、非常に小さなダイナミック・コンフィギュレーション機能)です。また、XtremeDSPスライスどうしを直結する専用ポート、配線リソースも用意されています。

XtremeDSPスライスのI/Oレジスタは、ほかのハードウェア・リソースを消費せずに、データシート上で示される最高速で動作します。また、フィルタ構造を実現する際に、性能低下の原因となるアダー・ツリー構造ではなく、アダー・チェーン構造(積和演算器をカスケードにつなげた構造)を構成できます。

今回、XtremeDSPスライスにOp-mode付きマルチプレクサを導入したことにより、一つのスライスで40種類以上の機能を実現できるようになりました。また、その設定を1クロック周期の間に動的に変更することも可能です。

本誌付属のDVD-ROMに収録されているSystem Generator for DSP v6.3には、XtremeDSPスライスに対応したブロックが用意されています。Op-modeを入力する専用ポートとモード指定のための定数をConstantブロックで与えることで、Op-modeをブロックの外部入力として利用できるようになります(図A-1)。



図A-1 Op-mode対応DSP48ブロック