大規模PLD+CPUコアの開発環境
―Niosのデザイン・フロー

Jason Krause

大規模PLDに搭載できるソフト・マクロのCPUコアが増えている．CPUに対する要求仕様がきびしくないアプリケーションにおいて，PLDの特徴である柔軟性を活かせることが大きなメリットである．しかし，CPUコアを含むハードウェア・システム（システムLSI）の設計は，決して容易ではない．このため，設計支援ツールが重要となる．また，CPUを使う場合は，そのソフトウェアの開発環境も忘れてはならない．本稿では，本誌2000年10月号の特集で取り上げたソフト・マクロのCPUである「Nios」を例に，大規模PLDによるシステムLSI開発の実際を説明する．（編集部）

はじめに

米国Altera社のExcalibur（エクスカリバー）は，設計者が自らプログラマブル・ロジック・デバイス（PLD）にエンベデッド・プロセッサを統合したシステムLSIを実現するためのソリューションです．アプリケーションによって，システムLSIのプロセッサに対する要求が異なるため，Excaliburファミリでは，複数の製品が提供されています．性能を重視し高性能のプロセッサが必要なアプリケーション用に，ARM9 ThumbおよびMIPS32 4Kの32ビットRISCプロセッサを内蔵したPLDがあります．これらの製品は，最大200MIPSで動作可能です．またPLDの柔軟性を重視し，50MIPS程度のパフォーマンスしか必要としないアプリケーションに対しては，32ビットRISCプロセッサのNiosがあります．

Niosは，HDLで記述された「ソフト・マクロ」です．PLDのロジック・エレメントを使用してプロセッサ回路が構成されます．Altera社のAPEXファミリのPLD構造に対して開発された，オリジナルのRISCアーキテクチャです（**表1**．詳細は本誌2000年10月号「プログラマブル・ロジックでシステムLSIを実現 ―― PLDに最適化されたソフト・マクロのCPUコアNios」を参照）．

Niosは，HDLベースのIPコアであり，アプリケーションのパフォーマンスとコスト要求に対して，ユーザがプロセッサ・コアのアーキテクチャとペリフェラルを最適化できます．また，PLDを使うことによって，自由に何度でも変更できます．しかし，プロセッサ・コア，ペリフェラル，またはオンチップ・バスの変更を短時間で実行できなければ，PLDの柔軟性という長所を生かすことができません．ソース・コードの変更を手作業で行うことはむずかしいため，使いやすい開発ツールが必要になります．そこでウィザード形式のグラフィカル・ユーザ・インターフェースをもつNios対応の開発ツールが用意されています．

本稿では，Niosのデザイン・フローとエンベデッド・プロセッサの開発において必要とされるソフトウェア開発ツールについて解説します．

〔表1〕Niosコアの特徴

項　目	特　徴
アーキテクチャ	Altera社オリジナルの32ビットRISC （16ビットも可能）
命令セット	16ビット固定長，1クロックに1命令を実行
アドレス・バス幅	設定可能（32ビット，4Gバイトまで）
汎用レジスタ	レジスタ・ファイル・サイズは512本の32ビット・レジスタまで設定可能 レジスタ・ウィンドウイング技術を使用
バレル・シフタ	長さを設定可能（31ビットまで）
乗算回路	高速乗算用の回路を追加可能

Nios開発環境

Niosの開発環境は，基本的に，三つの機能で構成されています．PLDベースのシステムLSI設計の場合，この三つすべてが必要です．一つでも欠けると，開発環境として不完全になり，せっかくの高性能プロセッサが使いにくいものになってしまいます．

(1) ハードウェア設計ツール

Niosシステムを設計するための，ウィザード形式の「MegaWizard」というツールです．グラフィカル・ユーザ・インターフェースにより，Niosコア，ペリフェラル，およびシステム・バスのパラメータを設定すると，HDLソース・コードが生成されます．

また，MegaWizardは，システム設計からプログラミング・ファイルをデバイスにダウンロードするまでのハードウェア設計フロー全体で使用するツールです．論理合成ツール（Exemplar Logic社のLeonardoSpectrumとSynopsys社のFPGA Express）を自動的に起動する機能をもちます．Quartus（APEXデバイスの開発ツール）メニューからもMegaWizardにアクセスできます．

⑵ ソフトウェア設計ツール

Niosのソフトウェア開発ツールはGNUです．エンベデッド・プロセッサ向けのソフトウェア開発ツールとしては，GNUPro（米国RedHat社）が広く使われているため，NiosでもGNUProが採用されました．Nios用のC言語コンパイラ，アセンブラ，リンカは，米国のRedHat社（旧Cygnus社）により開発されました．

⑶ デバッグ・ツール

Niosのデバッグ・ツールは二つあります．一つはRedHat社のInsight Debugger（インサイト・デバッガ）です．これはWindowsベースのソフトウェア・デバッガです．GUIでメモリ編集，ステップ実行，ブレークポイント，ウォッチポイントなどの機能をサポートします．

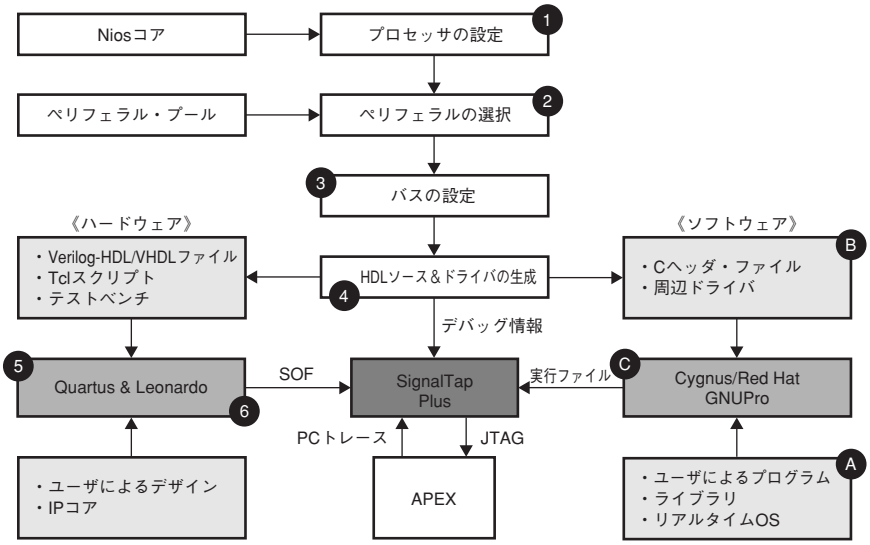
二つ目は，Altera社のSignalTap Plusハードウェア・デバッグ・ツールです．エンベデッド・ロジック・アナライザのSignalTap Plusは，PLD内部の信号および基板上の信号を取り込むことができます．Quartusを使って，信号データを波形フォーマットで見たり，内部信号と外部信号を同期化して解析できます．

Niosデザイン・フロー概要

先に述べたように開発環境には三つのツールがあります．エンベデッド・プロセッサ・システムのデザイン・フローは，一般にハードウェアとソフトウェアの二つに分けられます．デバッグもそれぞれについて行います．**図1**において，左側がハードウェア・フロー，右側がソフトウェア・フローです．

通常，ハードウェアの設計を先に行います．まず，Niosシステムでは，MegaWizardで設計者がNiosコアをコンフィグレーションします（**図1**の①）．次に，ライブラリからペリフェラルを選択して（メモリ・インターフェースを含む），パラメータを設定します（②）．カスタム・ペリフェラル（ユーザが設計する周辺回路）の場合，このステップで，ペリフェラルのインターフェースを定義して，ファンクションを接続するためのソケットを作ります．最後に，システムのメモリ・マップ，割り込み番号，割り込みベクタ・テーブル，およびリセット・ベクタを設定します（③）．これでMegaWizardがプロセッサ・コア，ペリフェラル，およびシステム・バス（ペリフェラル・バス・モジュール（PBM）という）のHDLソース・ファイルとペリフェラルのドライバ・ソフトウェアを自動的に生成します（④）．

この段階で，カスタム・ペリフェラルのHDLソース・ファイルとMegaWizardが生成したHDLソース・ファイルを統合して，チップ全体を論理合成します（⑤）．もちろん，論理合成する前に機能シミュレーションを実施することが可能です．MegaWizardがHDLシミュレーション用のテストベンチを出力します．



〔図1〕Niosデザイン・フロー