

NiosII マルチコアプロセッサのための システム設計レベルシミュレーション環境

志賀 瑞穂 北道 淳司

会津大学大学院コンピュータ理工学研究科

1. はじめに

近年、汎用プロセッサのみならず組込みプロセッサの分野においても、Intel Atom 330,D525[1]、NVIDIA Tegra(ARM11 MPCore)、Apple A4あるいはRUNESAS SH7200,SH7260シリーズ[2]などのマルチコア形式のプロセッサが開発されている。マルチコアプロセッサは、効率よく複数のタスクを並列に実行できるが、アプリケーションの設計において設計の正しさを保証する検証段階においては極めて多くの時間を必要とする。

我々の研究グループで開発を行っている Altera 社の FPGA ボード上で動作する NiosII[3] マルチコアプロセッサのためのシステム設計レベルでのシミュレーション検証環境を提案する。ターゲットデバイスとして FPGA を採用したのは、プロセッサアーキテクチャを最適化させるためにシステム構成を変更しやすいデバイスであるからである。本稿では、システム設計レベル言語 SystemC[4]を用い、NiosII プロセッサをモデリングし、その評価について述べる。

2. 提案シミュレーション手法

Altera 社が提供する FPGA 開発環境を用いたシステム設計環境の概要を図1に示す。

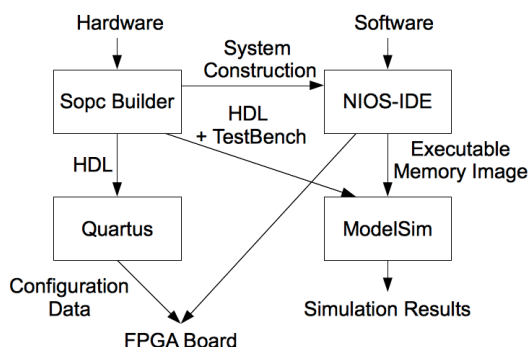


図1 Altera 社の設計環境の概要

ユーザは、SopcBuilder の GUI 環境を用いて、プロセッサコア、メモリ、その他 IP を選択し、バス

System Design Level Simulation Environment of Multi-core Processor NiosII

Mizuho Shiga and Junji Kitamichi

The University of Aizu, Graduate School of Computer Science and Engineering

構成を決定し、システムアーキテクチャを設計する。SopcBuilder はシステムの構成情報を合成系 QuartusII とソフトウェア開発系 NIOS-IDE に、構成情報とテストベンチをシミュレータ ModelSim に渡す。QuartusII において合成されたシステムの構成情報は FPGA ボードにダウンロードされる。NIOS-IDE においてプロセッサで動作するプログラムはコンパイルされ、命令メモリイメージが命令レベルシミュレータ (ISS)、RTL シミュレータ (ModelSim) あるいは FPGA ボードを用いた動作検証およびデバッグ環境に提供される。ModelSim は、VerilogHDL あるいは VHDL の RTL シミュレーションを高速に実行する動作検証環境を提供している。ただし、提供される NiosII プロセッサモデル (Standard と Fastest) は内部状態を観測することはできず、プロセッサと他のモジュール間のシグナルのみ観測できる。これは、プロセッサ内部の状態を観測しながらシステム開発およびデバッグを行う際に大きな障害となる。

本研究では、システム設計レベル言語の1つである SystemC を用いて、NiosII プロセッサをモデリングし、高速なシミュレーション環境を構築する。NiosII プロセッサの SystemC モデルは既に開発・市販されているが、プロセッサの内部および外部構成を自由に変更できるようにするため別途 SystemC モデルを開発した。

SystemC はシステム設計レベルから RT レベルまで様々なレベルにてモデリングを行うことができる。今回は1クロックに対して、1命令を実行するレベル (命令レベルと呼ぶ) にて、NiosII プロセッサをモデル化した。

NiosII プロセッサの仕様は以下の通りである。命令およびデータは 32bit 幅であり、32bit の PC (プログラムカウンタ)、32 個の RF (レジスタファイル)、CTL (コントロールレジスタ) などの構成要素からなる。途中の演算結果を保持する変数を適宜追加した。キャッシュおよび例外に対する処理は実装していない。動作を表すプロセスは、命令実行に相当するもの1つを用い、そのプロセスにおいて、キャッシュに対する命令などを除く 80 命令に対する処理を実現した。

3. 評価

SystemC を用いて NiosII プロセッサの命令レベルのモデルは 981 行となった (デバッグ用のコードも一部含む)。

乱数により初期化された配列に対するバブルソートプログラムを用いてシミュレーションに要した時間を計測する。シミュレーションは、リセットからバブルソートが終了するまでである。これには、スタックポインタの初期化などに加え、命令プログラム領域に格納されている配列の初期値をデータ用領域にコピーする命令群の実行も含まれている。

用いた評価環境は以下の通りである。計算機は Xeon 3GHz 4CPU x2, 16G Mem, Linux 2.6.18 である。SystemC は、OSCI のリファレンスコンパイラ Ver. 2.2, g++ 4.1.2, コンパイラオプションは -g を用いており最適化は行っていない。SystemC には、クロック、リセット、および CPU とメモリ間のすべてのチャネルの値を VCD 形式で保存させる内容がテストベンチに含まれている。比較として同じ計算機環境にて、QuartusII ver10.1, ModelSim Starter Edition 6.6C を用いた .SopcBuiler で、NiosII プロセッサ (Economy) とメモリ 1 つからなるシステムを作成し、NIOS-IDE でプログラムをコンパイルし、ModelSim でシミュレーションを行った。提案モデルの場合は、NIOS-IDE で得られた実行形式のファイルから初期メモリイメージを作成し、メモリの初期化を行う記述に変換し、システムのシミュレーションを行わせた。双方のシミュレーション時間 (モデルを動作させる期間) は同じである。双方の実行時間とその比 (ModelSim の実行時間と提案モデルを用いた場合の実行時間の比率) を表 1 に示す。ModelSim および SystemC は、それぞれ ModelSim および提案モデルを用いてシミュレーションを実行した時間である。それぞれコンパイルに要した時間については省略する。

n	命令数	ModelSim (msec)	SystemC (msec.)	比
4	591	1,821.778	7	260/1
8	1,475	4,187.071	11	380/1
16	4,491	9,873.113	23	429/1
32	18,483	36,983.801	88	420/1
64	69,263	135,406.000	316	428/1
128	268,643	531,423.563	1,226	433/1
256	269,219	532,375.723	1,222	435/1

表 1 シミュレーション実行時間の比較

2 つのシミュレータの実行時間はソートする要素数のおよそ二乗に比例して大きくなる。提案モデルのシミュレーションに要する時間は ModelSim を用いた場合に比べて、約 1/430 の実行時間でシミュレーションを終了できる。

NIOS-IDE の環境では、提案モデルと同じくサイクルレベルシミュレータも用意されているが、これを用いた場合、プロセッサとその周辺回路も含めてシミュレーションが行えず、我々の研究グループで開発を行っているマルチコアプロセッサに適用できないため、本研究でも比較実験は行わなかった。

ModelSim を用いた NiosII プロセッサのシミュレーションでは各モジュール間の信号値は可読であるが、内部レジスタの値をトレースすることができない。これはマルチコアプロセッサの動作を確認する上で大きな問題である。提案モデルは、内部の全レジスタのダンプをとることができるので、今後マルチコアプロセッサシステムの開発およびデバッグに活用できる。

4. まとめ

SystemC を用いて NiosII プロセッサの命令レベルのモデルを作成し、CAD ベンダが配布する標準環境でのシミュレータとのシミュレーション実行時間の比較を行った。結果約 430 倍の高速化を実現した。

本報告では、命令レベルのモデルの設計開発とその評価を行った。今後サイクルアキュレートレベル (クロックレベル) に拡張すること、割り込みなど未実装機能を実現すること、および提案するマルチコアプロセッサ用にシミュレーションモデルを開発しデバッグ用の環境を整備することが今後の課題である。

参考文献

- [1] Intel Corporate, "Intel Atom Processor 330 Series Datasheet", <http://download.intel.com/design/processor/datashts/322844.pdf>.
- [2] Renesas Technology Corporate, "SH7205 Group Hardware Manual: Renesas 32-Bit RISC Microcomputer SuperH RISC engine Family/ SH7200 Series", <http://documentation.renesas.com/eng/products/mpumcu/rej09b0372sh7205hm.pdf>.
- [3] ALTERA, "Nios II Processor Reference Handbook", http://www.altera.co.jp/literature/hb/nios2/n2cpu_nii5v1.pdf. 2010.
- [4] IEEE, "IEEE Standard SystemC Language Reference Manual", <http://standards.ieee.org/getieee/1666/>, 2006.