

マルチコアプロセッサを用いた並列論理シミュレーション手法

A Parallel Logic Simulation method based on Multi-core Processor

トウ ブンチク 竹内 勇矢 豊永 昌彦 村岡 道明

Wenzhu Dou Yuya Takeuchi Masahiko Toyonaga Michiaki Muraoka

高知大学大学院 理学専攻 (情報科学分野)

1. まえがき

論理回路の大規模化に伴い、論理シミュレーションにかかる時間が増大し、高速化が望まれている。本研究の先行研究[1]では、ソフトウェアのプロファイリングを行うことにより、実行時間を簡単に見積もる手法を提案した。本研究では、その見積もり手法を用いて、論理シミュレーションアルゴリズムに適用し、マルチコアプロセッサを用いた並列化アルゴリズムを提案することを目的とする。

2. ソフトウェアの実行時間を考慮したソフトウェアの並列化手法

本手法は、図1に示す5ステップより構成される。

- (1) 対象アルゴリズムの時間精度付きモデル[2]を生成する。
- (2) (1)で生成した時間精度付きモデルのシミュレーションとプロファイリングを行う。

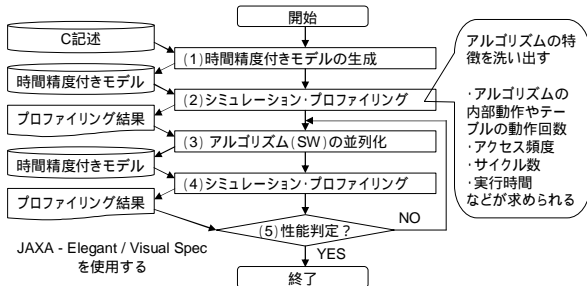


図1. 並列化手法のフローチャート

- (3) アルゴリズム (ソフトウェア) の並列化により高速化を図る。
- (4) 並列化アルゴリズムのシミュレーションとプロファイリングを行う。
- (5) 性能判定: 評価した結果が性能を満たせば終了となるが、ビヘイビアの並列化が適切でない場合には、(3)に戻る。

3. 適用対象 - 論理シミュレーションアルゴリズム

適用対象とする論理シミュレーションアルゴリズムは論理回路の伝搬遅延時間を考えず、論理機能を検証するレベルソート法を使用する。

今回用いた論理シミュレーションアルゴリズムの処理手順を以下に示す。

- ネットリストファイルを読み込む。
- 論理回路の中の組み合わせ回路部分の論理ゲート段数を数える。
- テストベクタを設定する。
- 入力ピンに繋がる段から順に段内のすべての論理ゲートを演算する。
- 論理ゲートの演算結果をシミュレーション結果として保存する。

4. 並列論理シミュレーションアルゴリズム

並列化手法 (図1) を論理シミュレーションアルゴリズムに適用した結果、実行時間の8割以上を占めた論理演算処理がボトルネック部分であることが判明した。

論理演算処理部について、マルチコアプロセッサを前提とした論理シミュレーションアルゴリズムの並列化により高速化を考える。並列プログラム間では、データの転送時

間がなくなると、並列効果が顕著になると考えられる。そのため、相互にデータ通信がないロジックコンによる論理回路の分割方法を用いて、高速化を行う。

マルチコアプロセッサのコア数は限られるため、複数のロジックコンを一つコーングループにまとめ、それを一つコアプロセッサに対応させる。並列の各コーングループの処理時間を均等化させるために「ロジックコンの畳み込み法」を提案した。また、コーングループの中の重複ゲート演算を削除し、さらなる高速化方法を提案した。

5. 評価および考察

図2のロファイリング条件は以下とする。

- ・評価データ: 組合せ回路 (4bit adder x 4, 16bit adder) 順序回路 (8bit マイコン)
- ・シミュレーションパターン数: 10000
- ・使用シミュレータ: Elegant / Visual Spec (version 4.1.6)
- ・ターゲットプロセッサ: ARM946E-S
- ・コンパイラ: arm-elf-gcc
- ・クロック周波数 (最大): 200MHz

図2では、論理演算処理の並列の比較結果を示す。

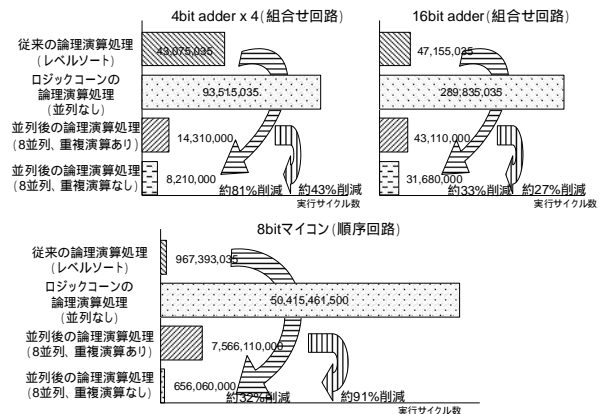


図2. 論理演算処理の並列の比較結果

本並列アルゴリズムを組合せ回路 (4bit adder x 4 と 16bit adder) と順序回路 (8bit マイコン) に適用し評価した結果、両者の回路とも従来のレベルソート法のアルゴリズムよりソフトウェア実行時間を30%から80%削減できる (8並列の場合) 見通しが得られた。

6. まとめと今後の課題

本手法を用いることにより洗い出した論理シミュレーションアルゴリズムのボトルネック部分である論理演算処理について、並列化方法を提案し高速化の可能性を見積もった。今後の課題として、大規模な論理回路や多種の回路による評価を行うことにより、本並列アルゴリズムの高速性を実証したいと考える。

参考文献

- [1] 松永惇弥, 村岡道明, 荒木大, “ソフトウェア並列化を考慮したハードウェア/ソフトウェア分割手法の評価”, 電子情報通信学会技術研究報告, vol.109, No.393, VLD2009-71, pp13-18 2010年1月
- [2] M. Muraoka, et al., “Software Execution Time Back-annotation Method for High Speed Hardware-Software Co-simulation”, Proc. of SASIMI2004 pp. 169-175, October 2004