

マルチコアを用いた高速並列論理シミュレーション手法の研究

Research of High-speed parallel Simulation Method Using Multi-core Processor

竹内 勇矢 豊永 昌彦 村岡 道明

Yuya Takeuchi Masahiko Toyonaga Michiaki Muraoka

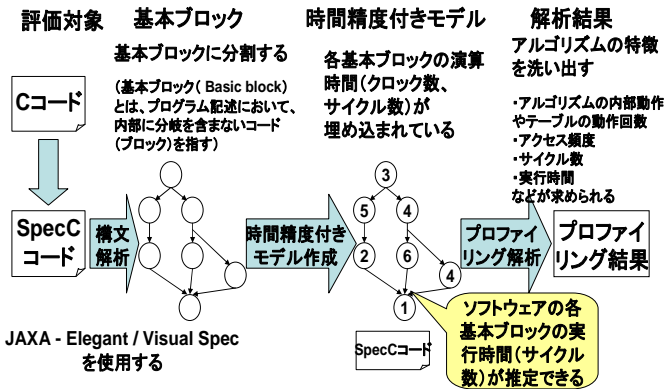
高知大学大学院 理学専攻(情報科学分野)

1. まえがき

近年、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論理シミュレーションアルゴリズムに適用し、効率の良い並列化アルゴリズムを提案する。

3. ソフトウェアの実行時間見積もり手法

本見積もり手法では、Visual Spec を用いて時間精度付きモデル[2]を作成し、プロファイリングすることで実行時間を求める。プロファイリングの手順を図1に示す。



本見積もり手法を用いてアルゴリズムをプロファイルし、ボトルネック部分を洗い出して並列化する。

2. 論理シミュレーション手法

適用対象とする論理シミュレーション手法は並列化が難しいイベント管理や伝搬遅延時間を考慮する必要が無いレベルソート法が並列化に向くと考え採用した。

4. 並列シミュレーション手法

本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をプロファイリングした結果、実行時間の8割以上を占める論理演算処理がボトルネック部分だと判明した。

本研究では、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並列化を行い、高速化を実現する。また、並列プログラム間のデータ通信は並列化による高速化を阻害するため、相互にデータ通信が発生しないファンアウトコンを用いて論理回路を分割し、並列処理を行った。

しかしながら、マルチコアプロセッサのコア数は限られるため、複数のファンアウトコンをグループ化してコア数以下のコーングループを作成し、それらとコアを1対1で対応させた。また、並列効果を向上させるため各コーングループの処理時間が均等になるようにコーングループを作成した。

5. 論理素子の複合化

さらなる高速化手法として論理素子の複合化[3]を提案する。論理素子の複合化とは、複数の論理素子を1つのマクロとして表現し、ルックアップテーブル(LUT)方式に置き換える手法である。複数の論理素子の演算結果を1回のテーブルアクセスで求められるため実行時間を短縮できる。

6. 評価および考察

本並列論理シミュレーションと市販の高速シミュレータとの比較を行った。評価環境を以下に示す。

・CPU: Intel®core™i7 3.07GHz

・シミュレータ

–商用 sim: 市販の高速シミュレータ (ModelSim SE 10.2c)

–並列 sim: 提案する並列論理シミュレーション (40 並列)

※並列論理シミュレーションの実行時間は実行時間見積もり手法を用いて求めた。

表1では評価に用いる回路情報を示し、図2では2つのシミュレータの性能比較を示す。

表1 評価用回路の情報

回路名	inst数	FF数	論理段数
cpu	2,148	173	56
cpux20	42,599	3,460	56
cpux40	85,179	6,920	56
ldpc_encoder	78,806	0	12

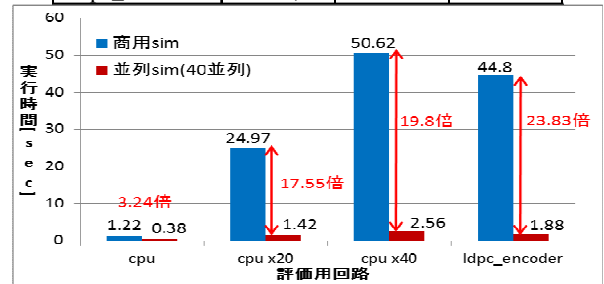


図2. 性能比較

本並列アルゴリズムにより順序回路(cpu系)では3~19.8倍、組み合わせ回路(ldpc_encoder)では23.83倍の高速化率を得られた。回路規模が大きい論理回路で20倍程度の高速化率が得られると推測される。以上の結果から、本並列アルゴリズムは論理シミュレーションの高速化に有用であることが示された。

7. まとめと今後の課題

本見積もり手法を用いることにより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を洗い出し、論理演算処理について並列化方法を提案した。また、高速演算手法として論理素子の複合化を提案し、さらなる高速化を目指した。これらの高速化手法を適応した並列論理シミュレーションを開発し、市販の高速シミュレータと比較した結果、順序回路及び組み合わせ回路ともに20倍程度の高速化率が得られた。今後の課題として、より大規模な論理回路を用いて評価することやメニーコア・プロセッサを想定し100並列以上で評価すること及び並列化によるオーバーヘッドを削減することでさらなる高速化が期待できる。

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
- [3] 竹内勇矢, 豊永昌彦, 村岡道明, “マルチコアを用いた高速並列論理シミュレーション”, 電子情報通信学会技術研究報告 Vokl.2015-SLDM-169 NO.12 pp-49-54, 2015年1月29日