Technical Reports on Information and
Computer Science from Kochi
Vol. 5 (2013), No. 3

並列化アルゴリズムによる論理シミュレーションの高速化手法の提案

Proposal of Acceleration Method for Logic Simulation based on Parallel Algorithm

竹内　勇矢*　トウ　文竹**　村岡　道明*
Yuya Takeuchi　Wenzhu Dou　Michiaki Muraoka
高知大学　理学部(情報科学コース)*　高知大学大学院　理学専攻（情報科学分野）**

1.まえがき

近年、半導体技術の進歩により回路の大規模化に伴い論理シミュレーションの演算時間が増大し、回路設計の長期化につながっているため、論理シミュレーションの高速化が要望されるようになった。

先行研究で提案されたソフトウェア実行時間を考慮した並列化手法[1]を用いて、論理シミュレーションアルゴリズムに適用し高速化を目指す。

2.並列化手法

並列化手法は次の 5 ステップにより構成される。

(1)時間精度付きモデルの作成：ターゲットプロセッサとクロック周波数を決定しソフトウェア時間精度付きモデルを作成する。

(2)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：ターゲットプロセッサにおいてのアルゴリズムの内部動作やアクセス頻度、サイクル数、実行時間を求める。

(3)アルゴリズムの並列化：(2)の性能評価結果からアルゴリズムのボトルネック部分の並列化により高速化を図る。

(4)シミュレーションとプロファイリング：(3)で作成した並列アルゴリズムを(2)と同様に時間精度付きモデルを生成し実行結果を求める。

(5)性能評価：評価した結果が性能を満たせていれば終了となるがそうでないなら(3)に戻る。

3.アルゴリズム(論理シミュレーション)

論理シミュレーションの処理手順を以下に示す。

(1)ネットリストファイルを読み込む。

(2)組み合わせ回路の論理ゲート段数を求める。

(3)テストベクタを設定する。

(4)入力ピンにつながる段から順に段内のすべての論理ゲートを演算する。

(5)出力ピンの演算結果がシミュレーション結果として保存される。

上記のアルゴリズムに並列化手法を適用させた結果、ボトルネック部分は(4)の処理だとわかった。(4)の処理を並列化させ、評価を行ったが並列度の低い回路の高速化率は向上しなかった。並列度の低い回路の並列化を行うと分割した回路間にデータ通信が発生し、並列化を行ったとしても待ち時間が増大するためである。そのため、本研究ではデータ通信が発生しない並列手法を提案する。

4.提案する並列手法

本分割方法は、AND/OR プレーンを利用した技術であり、図 1 に示すように任意の組み合わせ回路を AND プレーンと OR プレーンで表現することができる。

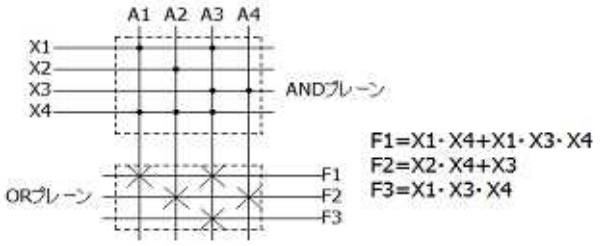

図 1 AND/OR プレーン

本研究では、OR プレーンの出力ごとに論理演算の並列化を行い、高速化率を調査した。OR プレーンの出力ごとに回路分割を行っても、各出力に対応の論理演算が格納されているためデータ通信は発生しない。

二つ目は割り付け方法である。本研究の分割方法では出力の個数に対応して論理演算が定義される。一般の回路ではマルチコアに搭載されているプロセッサ数よりも出力の個数の方が大きくなる。そのため、マルチコア上で動作する並列シミュレーションソフトウェアに論理演算を割り付ける処理が必要となる。

5.結果

本研究のシミュレーションやプロファイリングを行う環境を以下に示す。

・シミュレータ：Elegant/Visual SpecC(version4.1.6)
・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：250MHz
・シミュレーション回数：1000 回

表 1 プロファイリング結果(シミュレーション時間、msec)

	論理シミュレーション	本並列手法適用(8 並列)	高速化率
16bit adder	552.8	436.1	1.27
4bit adder x16	1954.2	462.4	4.23
H8 マイコン	11271.4	7763.6	1.45

表 1 は、今回提案した処理を加えた並列論理シミュレーションの評価結果である。並列度の低い回路である 16bit adder の高速化率を約 1.3 倍向上でき、順序回路である H8 マイコンでは約 1.5 倍向上できる見通しを得た。また、並列度の高い回路である 4bit adder x16 では、高速化率が約 4.2 倍と高い数値を得られた。

6.まとめと今後の課題

本研究では、データ通信が発生しない分割方法を提案した。この方法による並列論理シミュレーションを並列度の高い回路と低い回路、順序回路の 3 つパターンで評価を行った。その結果、並列度の低い回路では約 1.3 倍、順序回路では約 1.5 倍程度の高速化を得られ、並列度の高い回路では高速化率が約 4.2 倍と高い数値を得られた。これらの結果から、本研究で提案した並列手法の有効性を示すことができた。

本研究からいくつか課題を見つけることができた。1 つ目がマルチコアへの割り付け方法の自動化。本研究では、出力の割り付けは手動で行っている。大規模回路に対応させるために自動化にする。2 つ目が論理式コード以外での並列化。これには、ロジックコーンや積和標準形などが考えられる。3 つ目が論理式の簡単化を行い積項数を減らすこと。4 つ目が回路を真理値表で表現し高速に演算。以上の 4 つを今後の課題として取り組み更なる高速化を目指す。

参考文献

[1]中田有哉，村岡道明，”ソフトウェアアルゴリズムの並列化手法の研究 ” 高知の情報科学第 3 巻 No.3、2011 年 3 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