

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の提案

Proposal of Parallel Logic Simulation Method based on Multi-core Processor

トウ プンチュク 村岡 道明
Wenzhu Dou Michiaki Muraoka
高知大学 理学部(情報科学コース)

1. まえがき

近年、組み込みプロセッサは数多くの分野で、例えば携帯電話、ゲーム機、車載機器などの低消費電力かつ高性能な組み込みシステムに用いられている。従来では、ターゲットプロセッサ上でのソフトウェアの実行時間を見積るのが困難であったが、先行研究では、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を簡単に試行する手法を提案した。本研究では、その手法を用いて、論理シミュレーションアルゴリズムに適用し、効率の良いソフトウェアの並列化方法を提案する。

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) 時間精度付きモデルの生成: C記述されたアルゴリズムを対象として、JAXAが開発したElegant/Visual Specというツールを使って、ターゲットプロセッサとクロック周波数を決定し、時間精度付きモデルを生成する。
- (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を行い、ターゲットプロセッサにおけるアルゴリズムの内部動作やテーブルの動作回数、アクセス頻度、サイクル数、実行時間等を求める。

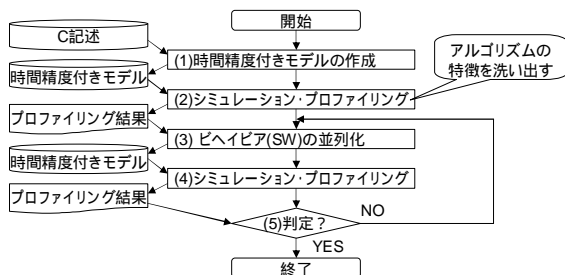


図1. 並列化手法のフローチャート

- (3) ビヘイビア(ソフトウェア)の並列化: (2)の性能結果から、マルチコアを前提とした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)で作成した並列ソフトウェアの時間精度付きモデルを生成し、(2)と同様に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)

今回用いた論理シミュレーションプログラムは以下の流れで実行する。まず、ネットリストファイルを読み込んで、組み合わせ回路の論理ゲート段数を数える。次に、テスト

ベクタを設定し、ゲート論理を段ごとに計算する。最後に、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. プロファイリング

今回、Elegant/Visual Spec (version 4.1.6) 上で時間精度付きモデルを生成し、ターゲットプロセッサとしてARM946E-Sを用い、そのコンパイラとしてarm-elf-gccを使用した。クロック周波数は最大を使い、200MHzである。本手法を論理シミュレーションアルゴリズムに適用し、4bit adderと16bit adderの評価データを用い、シミュレーションおよびプロファイリングした結果を表1に示す。

表1. 並列化前の実行時間の評価結果

	ファンクション名	行数	呼び出し回数	4bit adder		16bit adder	
				処理最小サイクル数(1回)	累積サイクル数(各ファンクションが占める割合)	処理最小サイクル数(1回)	累積サイクル数(各ファンクションが占める割合)
ネットリストを読み込む	load inst	58	1	40,297	40,297(0.49%)	166,791.5	166,791.5(0.27%)
	load net	27	1	18,356	18,356(0.22%)	81,427	81,427(0.13%)
	load pin	37	1	45,968.5	45,968.5(0.56%)	209,434.5	209,434.5(0.34%)
	load sn	25	1	29,928	29,928(0.36%)	147,511	147,511(0.24%)
入力ベクタを作成する	set vector	41	1	518,373	518,373(6.28%)	1,719,057	1,719,057(2.75%)
論理ゲート段数を数える	critical path	39	1	26,534	26,534(0.32%)	101,342	101,342(0.16%)
ゲート論理を段ごとに計算する	hostLogicCal2	100	1000	6,524	6,524,000(79.02%)	56,666	56,666,000(90.72%)

その結果、論理シミュレーションアルゴリズムの中で大部分の実行時間を占めているゲート論理を段ごとに計算する処理がボトルネック部分であることが判明した。

4. 並列化手法の検討

プロファイリングの結果に基づいて、論理シミュレーションアルゴリズムの並列化を検討し、以下に提案した。

提案1: 回路データの分割による並列化。

提案2: 回路データ設定処理の並列化。

提案3: ボトルネック部分(ゲート演算)の並列化。

提案4: 回路の並列化方法 - ブール微分法による分割。

5. まとめと今後の課題

上記の並列化手法の提案について、提案1、2、3は効果が小さいことがわかった。提案4については、ブール微分法による分割が考えられ、高速化が期待できることがわかった。また、他の回路の並列化方法として、「プログラマブルロジックアレイ(PLA)による分割」と「ロジックコンyonによる分割」が考えられる。

参考文献

- [1] 松永惇弥, 村岡道明, " タイミングを考慮したハードウェア/ソフトウェア分割手法の評価 ", デザインガイア 2009, pp31-36 2009年12月