

並列回路シミュレータの実装と評価*

3C-10

江橋 浩一†

徳吉 隆宏†

鈴木 響太郎†

天野 英晴†

慶應義塾大学 理工学部†
(株)日立製作所 水戸工場†

1 はじめに

VLSI技術の発達により大規模な電子回路シミュレーションの高速化の必要性はますます高まっており、並列処理はそのうち最も有望な手法の一つである。多大な計算を要する電子回路の過渡解析は、各時間ステップ毎に非線形連立方程式をニュートン法などの反復法を用いて解くが、その際反復の各ステップで一般疎行列を係数に持つ連立方程式を解く必要が生じ、この部分が最も時間を必要とする処理である。

直接法に関しては広く用いられているSPICEの並列化[1]が行なわれ、さらに静的スケジューリングを用いたより高度な手法[2]も提案されている。しかし、直接法は本質的に対角化の過程で並列性の減少を伴うため、大規模な問題を多数のプロセッサにより処理する場合は並列性を維持することが難しい。

これに対し反復法は、問題サイズに比例する並列性を維持することが可能なことから、大規模並列計算機に向いている。しかし、電子回路の方程式は係数の比が極端に大きいため、通常反復法では収束性がきわめて悪い。このため、前処理により収束性を保ちつつ並列処理を行なう方法が提案されている[3]。本報告では、回路の各節点に仮想的に容量を付加することにより高い安定性を実現する緩和法を[4]、並列計算機へ実装することを試みる。

2 緩和法のアルゴリズム

電子回路の過渡解析では、各時間ステップで、各節点の電圧ベクトル v に対して以下の非線形連立方程式を解く必要が生ずる。

$$F(v) = 0$$

ここで、 J をヤコビ行列

$$J = \frac{\partial F(v)}{\partial v}$$

とすると、ニュートン・ラブソン法は、以下の式の反復となる。

$$J\tilde{v} = F(v_k)$$

ここで、 $\tilde{v} = v_k - v_{k+1}$ である。

J は、一般的には対角優位にならないため、収束させるために各節点に仮想容量を付加し、常微分方程式に変換する。

$$C \frac{d\tilde{v}}{dt} = -J\tilde{v} + F(v_k)$$

この方程式を下に示すようにバックワード・オイラー法で解くことにより、ニュートン・ラブソン法で解くべき連立方程式の解を得ることができる。

$$(C + hJ)\tilde{v}^{m+1} = C\tilde{v}^m + hF(v_k)$$

この方法は、仮想容量の導入により連立方程式を連立常微分方程式に帰着させて解くため、直接法に比べ計算量が多くなるが、 C と h を適切に設定すれば、逐次処理で直接法を上回る性能を示すことが報告されている[4]。さらに、全てのステップが反復法により実現されるため、高い並列性を維持することができる。

3 並列計算機 ATTEMPT-0

アルゴリズムの並列化は、実装するハードウェアに大きく依存する。そこで、実験に使用したATTEMPT-0の構成について述べる。

ATTEMPT-0[5]は図1に示すように最大20枚のPUボードをFuturebusを用いて接続することが可能なバス結合型並列計算機である。

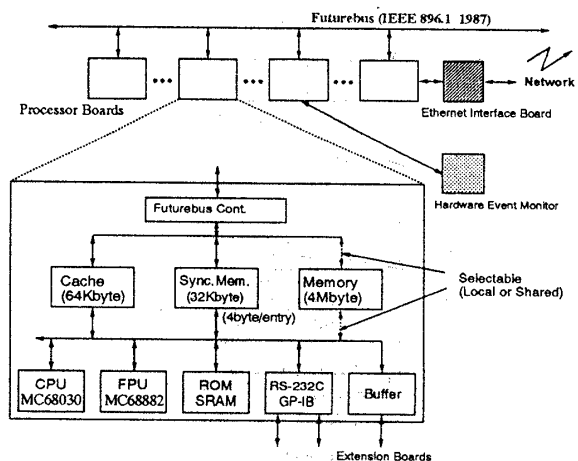


図1: ATTEMPT-0の構成

各ボードにはMC68030と4Mbyteのローカルメモリ、64Kbyteのプライベート・キャッシュが搭載されている。

*An implementation and an evaluation of a parallel circuit simulator

†Kouichi Ebashi, Hitachi, Ltd, Mito Works

†Takahiro TOKUYOSHI, Kyotaro SUZUKI, Hideharu AMANO, Keio University

また、シンクロナイザという特殊なメモリを持つ。PUボード上の拡張用コネクタにより、Ethernetボードや計測に使用するハードウェア・イベント・モニタを取り付けることができる。アドレス空間のうち、共有メモリとシンクロナイザは全PUで共有する。

4 実装

今回の実装では、各PUに電流の計算及びヤコビ行列の計算を素子単位に割り当て、連立方程式の解法では節点単位に割り当て並列化を行なった。

電流の計算は、各PUが素子の両端の節点に保持されている電圧からその素子に流れる電流を計算し、その結果を素子の両端の節点に格納する。ヤコビ行列の計算は、マスタがヤコビ行列の各要素の代入位置を計算し、スレーブに渡している。スレーブは、マスタから渡された各要素の代入位置へ計算したヤコビ行列を格納する。

回路の各種情報は共有データであるが、更新されることがないので、ローカルメモリにコピーし共有メモリに対するアクセスを低減し、アクセス競合を防いでいる。

同期は、バリア同期を用いている。また収束の判定は、マスタが行い、その判定が終るまでスレーブは待たされている。

5 評価

● 評価の方針

同期や通信は、しばしば速度の向上を妨げる要因となる。そこで、台数効果と共に、PU間の通信時間とその内訳及び、通信時間の全体に占める割合を測定した。

● 評価回路

評価に用いたシミュレーションの回路は、TTLのNAND回路を4つ縦列につないだものである。回路の内部節点数は28個、素子数は28個である。

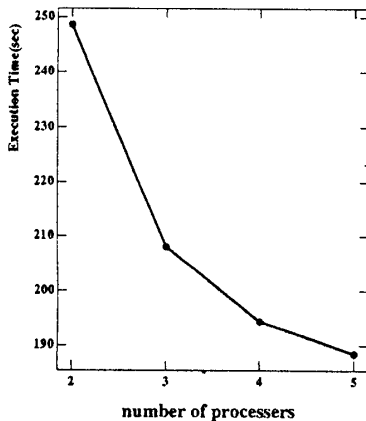


図 2: 実行速度

● 検討

本実装では、マスタでは実際の計算は行なわず収束判定を行なう役割のみを果たす。したがって、2PUでの実行は1PUに比べて速度は向上しない。2PUでの実行を基準にすれば図2のようにPU数が増加するごとに速度が向上している。

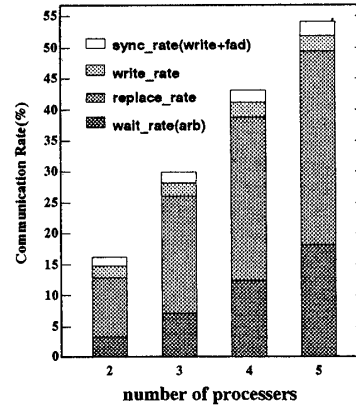


図 3: 通信時間

図3は、PU数別の通信時間とその内訳を示している。この図によると、バス待ちが5PU時で通信時間の3分の1を占めており、バス待ちによるネックは問題になっている。また、通信時間の実行時間全体に占める割合は、5PU時で55%弱であった。これは、各PUが共有メモリ上にある電圧、電流、ヤコビ行列を頻繁に書き換えられるためである。

今後の課題は、ヤコビ行列を分割して各PUのローカルメモリ上に置くことにより、回路シミュレータの通信時間を低く抑えて台数効果を引き出したい。

6 まとめ

本研究では、電子回路シミュレータを高速化するため並列化を行ない、ATTEMPT-0上で実装及び評価を行なった。評価の結果、電子回路のシミュレーション時間をPU台数の増加と共に短縮出来ることが確かめられた。今後、このシミュレータを大規模な回路へ適用するために、並列処理アルゴリズム及び同期について検討していく予定である。

参考文献

- [1] S.McGrogan, R.Olson and N.Toda, "Parallelizing Large Existing Programs - Methodology and Experiences," Proc. of Spring COMPCON pp.458-466, Mar. 1986.
- [2] 前川, 田村, Premchaiswadi, 笠原, 成田, "近細粒度タスクを用いた電子回路シミュレーションの並列処理," JSPP92 論文集, pp.453-460, 1992.
- [3] 須田礼仁, "前処理つき Jacobi 法による並列回路 simulation," 情処研報 Vol.93, No.72 (93-HPC-48-10), Aug. 1993.
- [4] M.Tanaka and M.Asai, "Relaxation based Algorithm by Variable Virtual Capacitors for Analysis of Large Stiff Nonlinear Networks," ISCAS 87, pp.617-620. 1987.
- [5] 鳥居, 竹本, 天野, 小椋, "バス結合型並列計算機の交信用メモリの性能評価," 情報処理学会論文誌 Vol.33 No.3 (1992).