

並列回路シミュレーションマシン Cenju

5 P-1

中田 登志之, †田辺 記生, 梶原 信樹, 松下 智,

‡小野塚 裕美, †浅野 由裕, 小池 誠彦

日本電気(株) C&C システム研究所, †日本電気(株) 超 LSI CAD 技術本部,

‡日本電気技術情報システム開発(株)

1 はじめに

VLSI の開発に不可欠な回路シミュレーションは倍精度浮動小数点演算を多用するため、VLSI 回路の規模が大きくなるにつれて、そのシミュレーション時間の増加が問題になってきている。

我々はモジュール分割に基づく並列回路シミュレーションのアルゴリズムを提案し、その有効性を4台構成のプロトタイプシステムで確認し、更に実際に64台構成のマルチプロセッサシステム Cenju を開発した。^{[1][2]} 本稿では Cenju のマシンアーキテクチャ、並列アルゴリズム及び評価結果について報告する。

2 モジュール分割に基づく並列アルゴリズム

過渡解析における従来の直接法では、1つの大規模・非線形・ランダムスパースな常微分方程式を解く。一方本方式では、シミュレーション対象となる回路を

並列実行フェーズ1 複数の非線形な小規模な部分回路の常微分方程式群の疑似解の求解、

単一実行フェーズ 接続ネットワークに相当する境界変数に関する線形方程式の求解、

並列実行フェーズ2 線形計算による、境界変数の解を用いた部分回路ごとの真の解の求解

の3段階を繰り返すことにより、実行する。^[1]

処理を3段階に分けたことにより、従来必要でなかった計算部分が生じるが、並列処理可能な部分が増し、並列処理の効果が期待できる。ただし、単一実行フェーズでの逐次処理が非並列な処理として残り、部分回路の分割数が多くなった場合に無視できなくなる可能性が存在する。

本方式で高速化を実現するためには、単一実行フェーズ処理時間の短縮及び並列実行フェーズでの各々のプロセッサの負荷の均衡化をはかることが重要となる。

3 並列回路シミュレーションマシン Cenju

前節で述べたアルゴリズムにおいて、主なプロセッサ間通信は

1. 並列実行フェーズ-1の終わりで部分回路を担当するプロセッサから接続ネットワークを担当するプロセッサへの通信における、n台のプロセッサから1台のプロセッサへの書き込み。
2. 単一実行部の終わりで、接続ネットワークを担当するプロセッサから、各部分回路を担当するプロセッサに分配する1台のプロセッサからn台のプロセッサへの書き込み。

が発生する。従って、並列回路シミュレータ Cenju を構成するにあたっては、1)分散共有メモリ方式の採用、並びに2)グローバル階層ネットワーク-ローカルバス方式の採用を基本方針とした。

図1に Cenju のシステム構成図を示す。

Cenju は、64台のプロセッサから構成される。各プロセッサは8台毎に1本のクラスタバスで結合される。これら8本

Cenju: A Multiprocessor System for Modular Circuit Simulation

Toshiyuki NAKATA, Norio TANABE, Nobuki KAJIHARA, Satoshi MATSUSHITA,

Hiroimi ONOZUKA, †Yoshihiro ASANO, and Nobuhiko KOIKE

NEC Corporation and ‡NEC Scientific Information System Development Ltd.

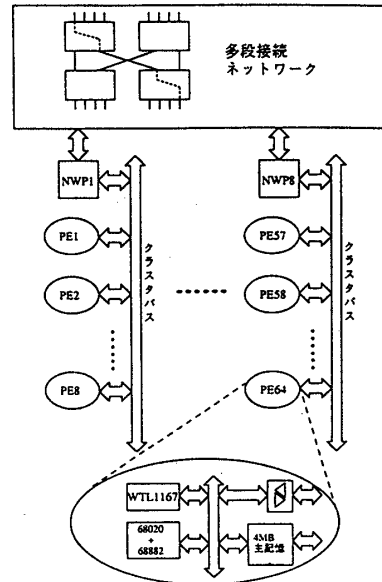


図1: Cenju のシステム構成

のクラスタバス間は蓄積型多段結合網で結合される。プロセッサ(以後 PE)には、MC68020 (20MHz), MC68882 並びに分散共有メモリを実現するために2ポート化した4MBのメモリを用いることとし、更に浮動小数点演算機能を強化するために、Weitek社の浮動小数点演算器(WTL1167)(ピーク1.6MFLOPS(倍精度))を搭載した。

クラスタ間結合網を設計するに当たっては、共有メモリアクセスを実現するために、ハードウェアと性能とのトレードオフを考慮して、クラスタ間データ書き込みをハードウェアで実現する一方、クラスタ間データ読み出しをハードウェアとソフトウェアの分担作業で実現することとした。^[2] このため、クラスタ間のデータ読み出しは遅くなる。

回路シミュレーションではこの点を考慮して、プロセッサ間通信を行う場合は、可能なかぎり Producer - Consumer モデルに基づき、データの生成者がデータの消費者にデータを転送するように工夫している。

4 並列度の評価

表1に本回路シミュレーションの評価で用いた回路の諸元、並びにプロセッサ数を変化させて実行させたときの実行時間と速度向上比を示す。(但し回路2の場合は、容量の関係で8台未満では実行できなかった。8台の時速度向上を1台の時の6.25倍として計算している。) いずれもダイナミックRAMの制御回路である。

回路1, 回路2共に速度向上としては15-16倍程度の速度向上が得られた。並列化しにくい直接法を用いてダイナミックRAMの制御回路という実用的な回路に対して、この様な値が得られたことは大きな意義があると思われる。

当初の予想に反して回路1でと回路2であまり速度向上に差がなかった理由としては、

1. 回路2の部分回路数が191個とPE台数64の3倍程度となっており、その分接続ネットワークの行列自体が大きくなったこと。

表 1: 並列性の評価

	回路 1	回路 2
Tr. 数	1688	6974
分割数	64	191
1	13 hr 44min (1.0)	...
2	7 hr 36min (1.81)	...
4	4 hr 07min (3.33)	...
8	2 hr 19min (5.93)	10hr 35min (6.25)
16	1 hr 26min (9.58)	6hr 35min (9.94)
32	1 hr 26min (10.56)	4hr 47min (13.82)
64	0 hr 56min (14.7)	4hr 11min (15.81)

```

for k:= 1 to n do
  if(PeContainsPivotRow){
    Broadcast akj (j=k to n)...(a)
  }
  Barrier(); /* 全 PE での同期 */
  for All i in k+1 to n which concerns this PE do ... (b)
    aik := aik/akk
    for j:= k+1 to n do
      aij = aij - aik * akj
    end
  end
  Barrier(); /* 全 PE での同期 */
end.

```

図 2: LU 分解の並列化アルゴリズム

2. 回路 2 の DRAM のモデルの特性のために、接続ネットワークの行列の非零要素率が 20% 程度と予想していたより大きくなったことがあげられる。このために、回路 2 では 1 台で実行する場合、逐次部分の全体に占める率が約 4% 弱になると推定される。

5 逐次部分の並列化 -LU 分解の並列化

前節では、部分回路数が増加すると逐次部分のオーバーヘッドが大きいたことが判明した。本システムの逐次部分の処理の大半は境界解に関する線形方程式の求解処理における LU 分解の部分である。スパース行列の LU 分解の並列化で得られる並列性は高々数倍という評価が一般的である。しかし、我々は本来なら逐次処理となってしまう部分を高速化する事によって得られる効果は大きいと考え、LU 分解の部分の並列化して評価することにした。

5.1 LU 分解の並列化のアルゴリズム

LU 分解の並列化の方法としては、外積法^[3]の LU 分解に対して、各行をプロセッサに割り付ける方法を採用した。

説明を簡単にするために、行列を密行列の場合として、行毎に並列化したアルゴリズムを図 2 に示す。

並列効果を妨げる要因としては a) のオーバーヘッド並びに b) に起因する PE 間の負荷のばら付きによるものが存在する。更に、回路シミュレーションの他の部分と比べるとこの部分での並列性の粒度は比較的細かいものとなり、同期のオーバーヘッドなどが目立つことになる。

尚 LU 分解に適したプロセッサ数と全体の回路シミュレーションに適したプロセッサ数は異なる事も予想される。そこで、この LU 分解での同期処理は、通常のプロセッサ間同期とは異なるルーチンを用いた。

5.2 性能評価

LU 分解の部分だけを抽出して回路 1 と回路 2 に関して測定した所、プロセッサ数が 6-8 で速度向上は飽和しており、最大で、1 台の時の 3.6 倍、逐次版に比べて 2.9 倍程度の速度向上が得られた^[4]。文献^[4]では、この結果から LU 分解を並列に実行した場合、全体のシミュレーション時間においては、回路 2

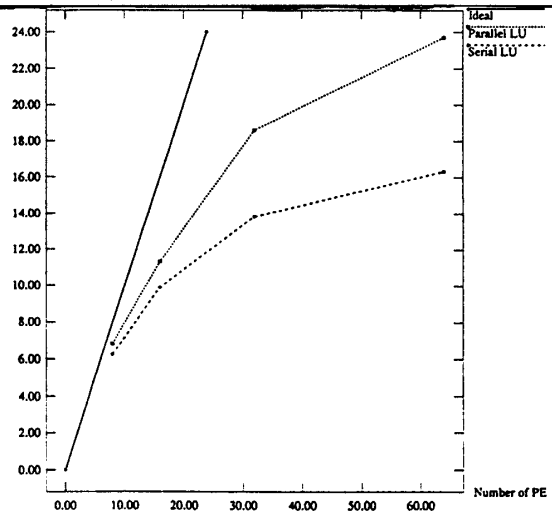


図 3: LU 分解を並列化したことによる速度向上

では 1 台に比べて 27.8 倍程度の速度向上が得られると予測した。その後、実際にシミュレーションプログラムに LU 分解の並列アルゴリズムをリンクし、回路 2 に関して測定を行なった結果を図 3 に示す。現在、64 台構成で 1 台の時の時点で 23.8 倍の速度向上が得られている。今後よりチューニングを行なっていく予定である。

6 終わりに

回路シミュレーションを効率よく実現するモジュール分割アルゴリズムに基づく並列回路シミュレーションマシン Cenju の構成、システムソフトウェアの概要および性能評価について述べた。

Cenju は回路シミュレーション以外にも粗い粒度の並列性を有する他の応用に適用することが可能である。我々は現在、故障シミュレーション^[5]、並列 LSI ルータ^[6]等の電気系 CAD の応用や、モンテカルロ型プラズマシミュレーション、ニューラルネットワークシミュレーション^[7]にも Cenju を適用して評価を開始している。

今後はこれらの評価を元に Cenju のアーキテクチャをより汎用な並列マシンに適したものに発展させていきたいと考えている。

謝辞

本研究の機会を与えて頂き、また有益な示唆を頂いた当社半導体事業グループ柳川技師長、超 LSI CAD 技(本)伊藤本部長、佐々木本部長代理、晴山部長、黒部部長、北村部長代理、C&C システム研究所石黒所長、森野所長代理、大野主管研究員に深謝致します。

参考文献

- [1] Nakata T., et al. *A Multiprocessor System for Modular Circuit Simulation*, Proc. ICCAD 87, pp364-367, 1987
- [2] 松下 他: 並列シミュレーションマシン, 情報第 37 回全国大会予稿集, pp97-98, 1988
- [3] 津田孝夫: 数値処理プログラミング, 岩波講座ソフトウェア科学 9:1988
- [4] 中田 他: 並列回路シミュレーションマシン Cenju, 情報処理, Vol. 31, No.5, pp.593-601, May, 1990
- [5] 中田 他: 統合 DA 用専用並列マシン MAN-YO における並列故障シミュレーション, 情報第 36 回全国大会予稿集, pp.1911-1912, (March 1988)
- [6] 山内 他: 並列シミュレーションマシン Cenju 上の LSI ルータの評価, 情報第 40 回全国大会予稿集, March 1990)
- [7] 田地野 他: 並列シミュレーションマシン Cenju におけるニューラルネットワークシミュレータ, 情報第 40 回全国大会予稿集, March 1990)