

1N-3

マルチグレイン並列処理用 シングルチップマルチプロセッサアーキテクチャ*

木村 啓二[†], 尾形 航[†], 岡本 雅巳[†], 笠原 博徳[†][†]早稲田大学理工学部電気電子情報工学科, † (株) 東芝

1 はじめに

集積回路の集積度の向上は留まるところを知らず、今後10年以内に1チップ上に10億トランジスタを集積できると言われている[1]。現在、このような大量のトランジスタを用いた次世代マイクロプロセッサとして、様々なアーキテクチャが提案されている。これらのアーキテクチャのうち、代表的なものとして以下のようなアプローチがある。一つは、回路の複雑化を前提として、スーパースカラ、VLIW、投機実行等を用いて性能を上げようとするもの[2][3]、もう一つは、DRAMを大量に搭載し、ベクトル演算を高速に行なうもの[4]、そして、DRAMとCPU複数個を1チップ上に混載するもの[1][5]等である。しかし、これらのアーキテクチャでは、命令レベルあるいはスレッドレベルのように単一の粒度の並列処理しか行なっておらず、特に命令レベル並列処理では並列性に限界があり、効果的な並列処理ないしスケラブルな性能向上が困難という問題点がある。

そこで、本論文では、まずFortranプログラムの並列処理を前提として、細粒度並列処理に加え、並列性の高いループイタレーションレベルの中粒度並列処理及び、スケジューリングオーバーヘッドを相対的に低く抑えることができるサブルーチンあるいはループ、基本ブロック間の粗粒度並列処理を組み合わせて使用することにより、高い実行性能を達成することができるマルチグレイン並列処理のシングルチップマルチプロセッサ上での適用と、それを可能とするアーキテクチャの評価の結果について述べる。

2 マルチグレイン並列処理

マルチグレイン並列処理とは、ループやサブルーチン等の粗粒度タスク間の並列処理を利用するマクロデータフロー処理[6][7]、ループレベルの並列処理である中粒度並列処理、基本ブロック内部のステートメントレベルの並列性を利用する近細粒度並列処理[8]とを階層的に組み合わせて、並列処理を効果的に行なう手法である。

2.1 マクロデータフロー処理

マクロデータフロー処理では、プログラムを以下に示す3種類の粗粒度タスク（マクロタスク（MT））[7]に分割する。

- BPA (Block of Pseudo Assignment statements)
基本ブロック及び基本ブロックを融合したブロック
- RB (Repetition Block)
最外側ナチュラルループ
- SB (Subroutine Block)
サブルーチン

*Single Chip Multiprocessor Architecture for Multigrain Parallel Processing

[†]Keiji KIMURA, Wataru OGATA, Hironori KASAHARA

[†]Waseda University, 3-4-1 Ohkubo Shinjuku-Ku, Tokyo 169

[†]Masami OKAMOTO

[†]TOSHIBA

MT生成後、コンパイラはBPA, RB, SB等のMT間のコントロールフローとデータ依存を解析し、それらを表したマクロフローグラフ(MFG)[6][9]を生成する。さらにMFGからMT間の並列性を再早実行可能条件解析[6][9]により引きだし、その結果をマクロタスクグラフ(MTG)[6][9]として出力する。その後、MTは条件分岐等の実行時不確定性が存在する場合にはダイナミックスケジューリングで、それ以外の場合にはスタティックスケジューリングにより各プロセスタスク(PC)に割り当てられ実行される。

2.2 中粒度並列処理

PCに割り当てられたMTがDoall可能なRBである場合、このRBはPC内のプロセッシングエレメント(PE)により、イタレーションレベルで実行される。

2.3 近細粒度並列処理

PCに割り当てられたMTが、BPAや中粒度並列処理を適用できないRBである場合、それらはステートメントレベルで並列化され、PC内のPEにより並列処理される。

ステートメントをPEに割り当てる際には、スケジューリング手法として、実用的なヒューリスティックアルゴリズムであるCP/DT/MISF法あるいはDT/CP法[9]を使用する。

3 評価対象アーキテクチャ

本章では、評価対象となるアーキテクチャのうちの2例について、その概要を述べる。これらのアーキテクチャの評価を、クロックレベルの精密なシミュレータを用いてシミュレーションすることにより行なう。

3.1 共通仕様

評価対象のアーキテクチャは、32bit固定命令長、ロード/ストアアーキテクチャのシンプルなシングルイシューRISCアーキテクチャのCPUを1チップ上に4個搭載するものと仮定する。今回はシングルチップマルチプロセッサのみについて評価を行なうものとする。

3.2 OSCAR型アーキテクチャ

OSCAR型アーキテクチャとは、図1に示すマルチプロセッサシステムOSCARを基に構成されたアーキテクチャである。

OSCAR型アーキテクチャは、CPU、データ転送ユニット(DTU)、ローカルプログラムメモリ(LPM)、ローカルデータメモリ(LDM)、そしてデュアルポートメモリで構成された分散共有メモリ(DSM)を持つプロセッシングエレメント(PE)と、集中共有メモリ(CSM)とを複数バスで接続する構成となる。LPM, LDMは自PEのCPUからのみアクセスでき、DSMは他PEのCPUからも直接アクセスできるものとする。また、DTUはCSMとDSMとの間のデータ転送を、CPUの処理と並列に実行することができるものとする。今回は、PE間バスの本数を3としてシミュレーションを行なった。

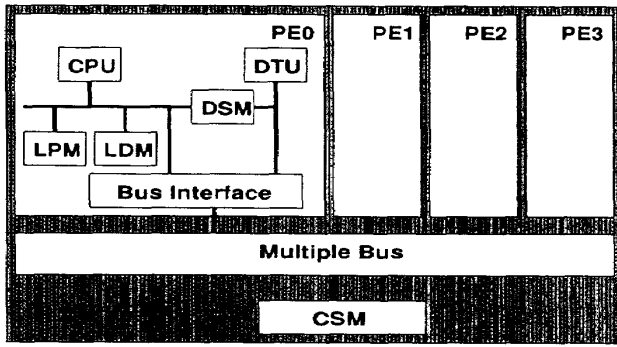


図 1: OSCAR 型アーキテクチャ

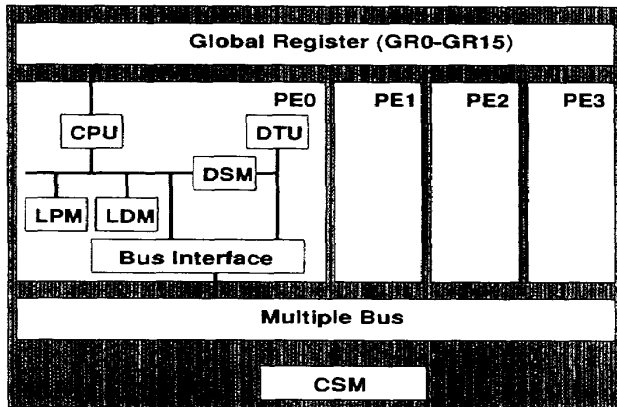


図 2: OSCAR + グローバルレジスタ型アーキテクチャ

3.3 OSCAR + グローバルレジスタ型アーキテクチャ

OSCAR + グローバルレジスタ型アーキテクチャとは、3.2 節で述べた OSCAR 型アーキテクチャの各 CPU から同時にアクセスできる共有グローバルレジスタ (GR) を 16 本持つアーキテクチャである (図 2)。

GR は、今回のシミュレーションでは、近細粒度並列処理を行なう際にステートメント間のデータ転送及び同期のために使用した。ステートメント間のデータ転送及び同期は、OSCAR 型では DSM を経由して行なうが、GR を利用することによって、コストの小さいデータ転送を行なうことができる。GR のステートメント間データ転送に対する割り当ては、一般のグラフカラーリング技術を用い、コンパイラが自動的に決定する。

4 シミュレーションによる性能評価

本章では、3.2, 3.3 節で述べたアーキテクチャのシミュレータを用いた性能評価の結果について述べる。

性能評価に使用したプログラムは、航空宇宙技術研究所において数値風洞の開発の際に性能評価の対象とした CFD プログラム「NS3D」の一部を取り出し、更に配列サイズを縮小したものである。このプログラムは、全部で 4 つの DO ループから構成され、そのうちのひとつが Doall 処理可能である。このプログラムに、中粒度並列処理のみを行なったもの、中粒度並列処理に加えシーケンシャルループのボディ部に近細粒度並列処理を行なったもの、さらに、近細粒度並列処理時のステートメント間データ転送に共有グローバルレジスタ (GR) を使用するもので比較を行なった。PE 数 1, 2, 4 で評価した結果を、それぞれ PE 数 1 での実行時間を基準とした時の速度向上比として表 1 に示す。ただし、PE 数 1 での実行時間は全手法で同一である。

コンパイル手法	PE 数		
	1	2	4
中粒度	1.000	1.055	1.092
中粒度+近細粒度	1.000	1.723	2.840
中粒度+近細粒度+GR	1.000	1.729	2.916

表 1: コンパイル手法による速度向上比の違い

表 1 より、中粒度並列処理のみを行なった場合、PE 数 4 で速度向上比が 1.092 に過ぎなかったものが、中粒度並列処理に近細粒度並列処理を組み合わせたマルチグレイン並列処理では 2.84 倍になり、マルチグレイン並列処理がプログラムの持つ並列性を有効に引き出すことがわかる。一方、共有グローバルレジスタを用いても、同一 PE 数の中粒度+近細粒度並列処理を行なったものの 3% 弱しか性能が向上しない。これは、3 本の PE 間バスのデータ転送能力が PE 数に対して充分高いためであると考えられる。

5 まとめ

本稿では、シングルチップマルチプロセッサではアーキテクチャの工夫により、細粒度並列処理だけではなくループ並列化、粗粒度並列化を組み合わせたマルチグレイン並列化が可能であることを示した。

今後は、スーパースカラ、VLIW 等の命令レベル並列処理プロセッサとマルチグレインシングルチップマルチプロセッサの性能比較、ハードウェアキャッシュとソフトウェア制御キャッシュの性能比較を行なうと共に、より高い実行性能を引き出すために、無同期近細粒度並列処理を効率良く行なうためのアーキテクチャの開発を行なう予定である。

本研究の一部は、通産省次世代情報処理基盤技術開発事業並列分散分野マルチプロセッサコンピューティング領域研究の一環として行なわれた。

最後に、CFD プログラム NS3D を御提供戴いた航空宇宙技術研究所の皆様へ感謝いたします。

参考文献

- [1] Hammond et al. "a single-chip multiprocessor". *Computer*, Vol. 30, No. 9, 1997.
- [2] Patt, Patel, Evers, Friendly, and Stark. "one billion transistors, one uniprocessor, one chip". *Computer*, Vol. 30, No. 9, 1997.
- [3] Lipasti and Sben. "superspeculative microarchitecture for beyond ad 2000". *Computer*, Vol. 30, No. 9, 1997.
- [4] Kozyrakis et al. "scalable processors in the billion-transistor era, iram". *Computer*, Vol. 30, No. 9, 1997.
- [5] 岩下, 宮嶋, 村上. "「ppram_m^R」アーキテクチャの概要". 情報研報, Vol. 96, No. 80, 1996.
- [6] 本多, 岩田, 笠原. "fortran プログラム粗粒度タスク間の並列性検出法". 信学論 (D-I), Vol. J73-D-I, No. 12, 1990.
- [7] 笠原, 合田, 吉田, 岡本, 本多. "fortran マクロデータフロー処理のマクロタスク生成手法". 信学論, Vol. J75-D-I, No. 8, 1992.
- [8] Kasahara, Honda, and Narita. "a multigrain parallelizing compilation scheme for oscar". *Proc. 4th Workshop on Lang. And Compilers for Parallel Computing*, Aug 1991.
- [9] 笠原. 並列処理技術. コロナ社, 1991.