

逐次型推論マシン CHI-IIの性能評価

7L-1

幅田 伸一、新 淳、小長谷 明彦、横田 実
日本電気(株) C&Cシステム研究所

1. はじめに

通産省の第5世代コンピュータ・プロジェクトの一環として開発した逐次型推論マシンCHI-II (Co-operative High Performance Sequential Inference Machine) のアーキテクチャ性能評価について報告する。^{[1][2]} CHI-IIは、述語論理型言語を使用した知識ベースの高速実行を目的としており、大容量実メモリと高速インタプリタを特徴とする。性能評価には、ECRC (European Computer-industry Research Center) のベンチマークプログラムとCHI-II上に構築したDNA配列検索知識ベース・システムKNOAを使用した。性能評価の結果、CHI-IIアーキテクチャのマイクロ操作並列処理機能が処理速度向上に効果があることを確認した。また、処理速度を測定した結果、コンパイルしたコードの実行性能でSun4/280上のQuintus Prolog 2.3版の1.5倍、インタプリタの場合は6.8倍の結果を得た。

2. CHI-IIアーキテクチャの特徴

CHI-IIは、タグ付きデータ形式を採用したタグアーキテクチャ・マシンである。タグアーキテクチャは、データタイプを表すタグ部とデータ値を表すバリュール部の処理を並列実行することで、処理が高速化できる。CHI-IIでは、データ処理におけるタグ部とバリュール部の並列処理の他に、アドレス演算、データ処理、条件分岐制御の3種の処理を並列実行することにより、処理の高速化を実現している。プロセッサ部は、データ演算用算術論理演算器、アドレス計算用加減算器と多ポート(読み出し用4個、書き込み用2個)のレジスタファイルを備え(図1)、78ビット長の水平型マイクロ命令により最大5個のマイクロ

操作を並列実行できる。

3. マイクロ操作の並列実行化の効果

CHI-IIのマイクロ操作並列実行機能による処理速度向上の効果について、DNA配列検索知識ベース・システムKNOAを使用して評価した。評価方法は、KNOAを実行中のCHI-IIのマイクロアドレスを8メガ・ステップ収集し、マイクロアドレスから各マイクロ操作の実行回数を求めた。CHI-IIのマイクロ操作は、メモリアクセス、レジスタ間データ転送、条件分岐、タグ部操作の4つに分類している。これらの操作は、使用頻度が多い基本操作であり、かつ、メモリアクセス制御部、算術論理演算実行部、マイクロプログラム実行制御部の各ハードウェアが独立しているため並列実行化が容易な操作である。メモリアクセスの中には命令フェッチ、レジスタ間データ転送の中には算術論理演算、条件分岐の中にはタグ部とバリュール部の値を使用した多方向分岐を含んでいる。

収集したマイクロアドレスから求めたマイクロ操作を、1マイクロステップに1マイクロ操作しか実行できないと仮定した場合の処理に必要なマイクロステップ数を表しているのが図2の1である。メモリアクセスとレジスタ間データ転送がマイクロステップ数の約36%を占めている。この2つのマイクロ操作を並列実行可能にした場合が図2の2である。2つの処理を並列実行することで、全体のマイクロステップ数が約74%まで減少する。メモリアクセス、レジスタ間データ転送、条件分岐の3処理を並列実行した場合が図2の3である。上記3処理とタグ部操作処理を並列実行可能にした場合が図2の4である。上記4処理の並列実行に、タグ部の比較とバリュール部の比較を並列実

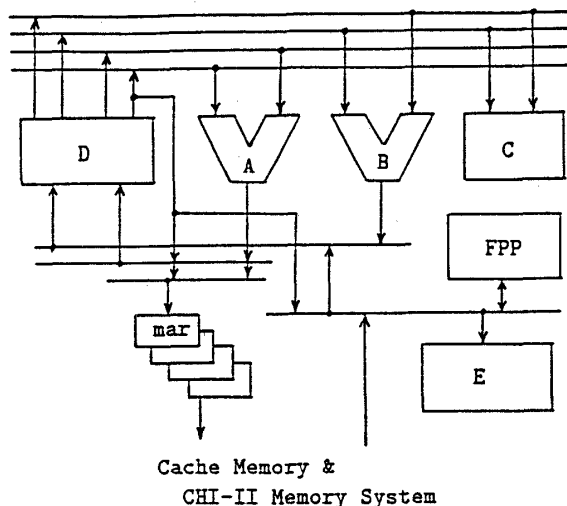


図1

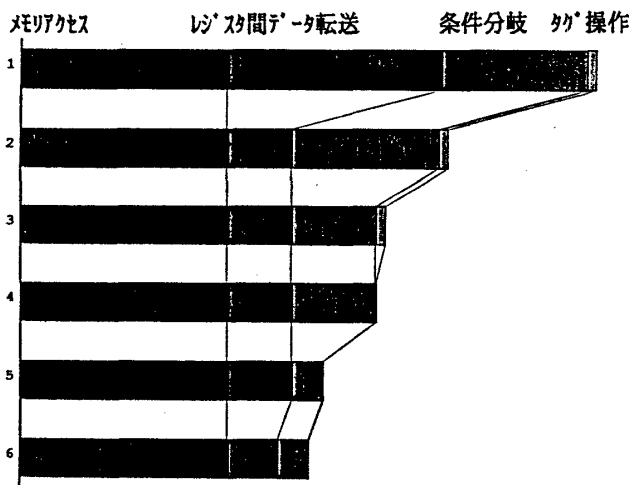


図2

Performance Evaluation of Sequential Inference Machine CHI-II

S.Habata, A.Konagaya, M.Yokota

C&C Systems Research Laboratories, NEC Corporation

行し、その結果で多方向分岐を可能にした場合が図2の5である。CHI-IIでは、タグアーキテクチャを採用した効果は、図2の4と図2の5でマイクロステップ数が減少した約15%と考えることが出来る。アドレス計算とデータ処理を並列実行可能にした場合が図2の6である。CHI-IIでは、マイクロ操作を並列実行化することによりマイクロステップ数が約50%に減少している。

4. マイクロ操作並列実行の評価

KNOAの場合、マイクロ操作を並列実行することで高速化できない処理がどの程度残っているかを求めたのが図3である。約26%のマイクロステップが1つのマイクロ操作しか実行していない。マイクロ操作の並列実行化できなかったマイクロステップの処理を調べた結果が図4である。ユニフィケーションにおけるデータタイプの検査とスタック上のフレームの読み出し操作が約58%を占めている。ユニフィケーションにおけるデータタイプの検査は、メモリアドレス計算、メモリからのデータ読み出し、読み出したデータのデータタイプ検査、データタイプに対応した処理の4フェーズからなる。マイクロ操作の並列実行化が出来ないのは、メモリからのデータ読み出しと読み出したデータのデータタイプ検査のフェーズで、並列実行化できなかったマイクロ操作の約34%を占める。これらのマイクロ操作を並列実行化するには、複合命令を導入し、複数の引数、または、リストの複数の要素を同時に処理する手段^[2]を積極的に導入する必要がある。スタック上のフレームの読み出しは、並列実行化できなかったマイクロ操作の約24%を占める。この処理は、メモリアクセスが制約となっており、メモリから一度に複数語のデータをアクセス可能にしなければ解決は不可能と考えられる。

5. メモリアクセス率

CHI-IIでは、命令/データ共有キャッシュメモリを採用している。この為、KNOAの場合、メモリアクセスが占める割合が約74%と高い値である。命令/データ共有キャッシュメモリの場合、命令フェッチがデータ・アクセスと衝突し、マイクロステップ数が増加する可能性が存

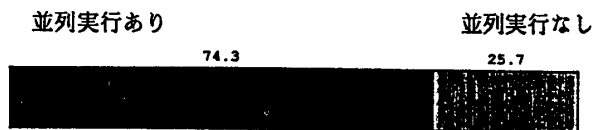


図3

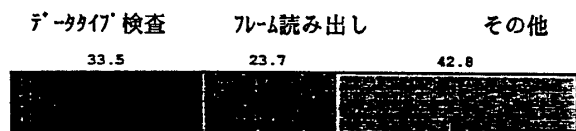


図4

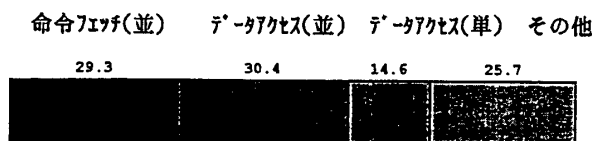


図5

在する。メモリアクセス操作が占める割合と他の操作と並列実行している割合を求めた結果が図5である。命令フェッチは全て他のマイクロ操作と並列実行している。命令フェッチと並列実行しているマイクロ操作は、メモリアクセス直前のアドレス計算が多く、命令/データ共有化の弊害がないことが確認できた。データアクセスで他のマイクロ操作と並列実行できない操作が全体の約15%存在するが、データアクセス単独のマイクロステップの約70%は、4節で述べたユニフィケーションにおけるデータタイプ検査とスタック上のフレーム読み出しのものである。したがって、キャッシュメモリは、命令/データ共有方式で問題はなく、一度にアクセスできるデータを複数語にする改善が必要だと分かった。

6. 処理速度

CHI-IIの実用性評価の1つとして処理速度を測定した結果を報告する。評価には、ECRC (European Computer-industry Research Center) のベンチマークプログラムを使用した。コンパイルしたコードの実行速度とインタプリタを使用した場合の実行速度を測定した。参考として、Sun4/280上のQuintus 2.3版の処理速度も測定した。測定した結果が表1である。インタプリタは、クローズコンパイル方式^[3]を採用した結果、Quintus Prolog の約6.8倍である。

7. まとめ

述語論理型言語で記述した知識ベースの高速実行を目指して研究開発して来た逐次型推論マシンCHI-IIの性能評価を行った結果、高速化の手段として採用したマイクロ操作の並列実行機能が処理の高速化に有効であることがわかった。さらに、実用性評価のため処理速度を測定した結果、コンパイルしたコードの実行速度で Quintus Prolog の約1.5倍、インタプリタの性能で約6.8倍であることを確認した。

参考文献:

- [1] R.Nakazaki, et al. "Design of a High-speed Prolog Machine (HPM)", Proc. of the 12th International Symposium on Computer Architecture, June 1985
- [2] S.Habata, et al, "Co-operative high Performance Sequential Inference Machine:CHI", Proc. of ICCD'87, 1987
- [3] 新 他、「逐次型推論マシンCHIにおける動的述語の実現と評価」、第39回情処全国大会

Table 1: Performance Measurement Result (KLIPS)

Benchmark Program	CHI		Quintus(v2.3) SUN4/280		CHI/Quintus	
	comp	interp	comp	interp	comp	interp
fibonacci	154.1	69.3	59.2	5.5	2.6	12.6
map	98.6	36.1	40.5	2.9	2.4	12.4
mham	102.5	31.7	75.4	6.5	1.4	4.9
mutest	99.3	32.2	65.6	5.8	1.5	5.6
qs	129.4	40.6	77.6	5.3	1.7	7.7
qu	180.0	58.4	119.8	6.7	1.5	8.7
query	74.4	28.1	65.1	5.7	1.1	4.9
differe	41.1	24.0	14.1	4.5	2.9	5.3
diff	116.0	39.3	69.3	5.3	1.7	7.4