

## 推論プロセッサ UNIREDI: 命令セットの評価の概要

1 Q-9

島田 健太郎 小池 汎平 田中英彦

E-mail: {shimada,koike,tanaka}@mtl.t.u-tokyo.ac.jp

東京大学 工学部

## 1 初めに

現在、我々は並列推論マシン PIE64[1]の開発を行なっている。PIE64は、対象言語として Committed Choice 型言語 Fleng を実行する。PIE64は64台の推論ユニット(IU)を2系統の自動負荷分散機能を持つ回線交換ネットワークで結合した構成を採る。各IUにはネットワークと接続しFleng向きの高度な通信/同期機能を提供する Network Interface Processor(NIP), IU内でのFlengの実行処理を行なう推論プロセッサ UNIREDI, 及びプロセス管理等を行なう管理プロセッサ(MP)として汎用プロセッサ SPARCがある。我々はこの推論プロセッサ UNIREDIについてこれまでアーキテクチャ、命令セットの設計を行ない[3], 更に多重コンテキスト処理などのアーキテクチャの評価を行なってきた[4]。ここでは、Flengの並列実行のために特に専用化された命令セットと、そのシミュレーションによる評価について述べる。評価結果では、デレファレンス命令などの専用命令の効果が充分あることを確認した。

## 2 推論プロセッサ UNIREDIの特徴

推論プロセッサ UNIREDIは並列マシンの要素プロセッサとして、Committed Choice 型言語 Fleng を効率良く実行するために設計された専用プロセッサである。その特徴としては次のようなことが挙げられる。

- 記号処理向きの機能としてタグ・アーキテクチャを採用、Flengの処理を効率化している。
- 命令バス、メモリ読み出しバス、メモリ書き込みバスの三つの分離したメモリバスを持ち、バンド幅の広い並列メモリアクセスを行なう。
- 最大四つのコンテキストによる多重コンテキスト処理によって動的なパイプライン充足率の向上を行なう。
- コプロセッサ・コマンドバスを持ち、NIP, MPとの間でのコマンド/リブライの通信プロトコルをサポートしており、これらのプロセッサと協調動作を行なう。
- Flengの処理を効率化する強力な命令セット[3]を持つ。

UNIREDIではすべての命令は1ワード(32ビット)長であり、パイプラインの1クロックで実行できる。またデータのワード長も同じく32ビットであり、ガーベジ・コレクションなどのマーク部2ビットの他、タグ部2ビット、データ部28ビット(ポインタ型)、或はタグ部6ビット、データ部24ビット(定数型)からなっている。

The Inference Processor UNIREDI: Overview of Evaluation of its Instruction Set Processor Architecture  
Kentarō SHIMADA, Hanpei KOIKE, and Hidehiko TANAKA  
Faculty of Engineering, University of Tokyo

dereference reg, Label :

```

L1:  mov      reg, sus      ; Sus.Regへコピー
      jump_on_not_tag reg, VAR, L2 ; VAR型でなければ終了
      load   [reg], %tmp   ; %tmpに参照先をロード
      jump_on_tag   %tmp, UDF, Label ; UDF型ならLabelへ
      mov    %tmp, reg     ; %tmpをregへ
      jump  L1            ; 繰り返し
L2:

```

図1: デレファレンス命令の分解

## 3 専用命令セット

次に UNIREDIの命令セットの中で特徴的な命令を幾つか説明する。その他の命令については[3]を参照されたい。

## 3.1 デレファレンス命令

UNIREDIの命令セットにおいて最も特徴的な命令としては、まず変数参照の手続きを行なうデレファレンス命令が挙げられる。これは指定したレジスタの内容が変数への参照(VAR型)である間その参照先を読み出し、VAR型でない時それが未束縛変数(UDF型)であるかどうかで条件ジャンプを行なう。デレファレンスした結果は、サスペンドの処理の効率化のために Suspension Register (sus: 汎用レジスタのR30)にコピーされる。図1にデレファレンス命令の動作を他の命令で分解して示す。実際にはこの他に、UDF型であった時の飛び先を Alternative Pointer (ap: 汎用レジスタのR28)で指定したり、変数への参照がリモートであったらNIPに対するDEREFコマンド[2]の発行に切り替わる機能がある。

デレファレンス命令にはこの他に、デレファレンスした結果がUDF型でなかった時にリストないしベクタへの参照(LST型, VEC型)であるか、或は即値で指定した定数であるかどうか調べたり、更にリスト、ベクタへの参照であればその先頭ワードを読みだす複合命令がある。これらは、結果がUDF型であった時の飛び先と、それ以外で指定した型/定数でなかった時の飛び先を命令中のオフセット値と先の Alternative Pointerの値とで別々に指定することが可能である。

## 3.2 エクセキューション命令

エクセキューション命令は、Flengの実行において Tail Recursionを行なうために特に設けられたジャンプ命令の一種である。これは通常のジャンプ命令の動作の他に Tail Recursionの中断を指示する Break Flagの検査、及びリモートな構造データのローカル・コピーを作るスクラッチ領域レジスタの初期化を行なう。Break Flagがセットされている時の飛び先は Break Trap Pointer (btp: 汎用レジスタのR29)で指定される。また、エクセキューション命令には更に前述のデレファレンス命令と

表1: サンプル・プログラムの命令実行割合

	nreverse 30	qsort 50	primes 100	8queen	best- path
総クロック数	5427	8049	41068	656714	270466
リダクション回数	496	380	726	38878	5704
サスペンド回数	29	125	103	594	794
性能 (KRPS)	913.9	469.5	176.8	592.0	210.9
コードサイズ*	300	460	1200	764	7940
総実行命令数	4858	7766	39352	648362	193462
Load/Store	39.1%	36.6%	5.0%	33.8%	28.6%
Bind	9.6%	4.9%	1.0%	2.7%	3.3%
Jump/Call	9.6%	8.0%	22.6%	7.9%	9.2%
Execute	10.8%	7.5%	1.5%	6.1%	3.5%
Dereference	4.9%	14.8%	2.5%	18.9%	15.1%
Allocate	10.8%	6.2%	1.0%	5.6%	6.3%
Arithmetics	0.0%	2.9%	49.6%	7.4%	3.9%

\*コードサイズの単位はByte

の複合命令が幾つか設けられている。

### 3.3 バインド命令

バインド命令は、未束縛変数に値の束縛を行なう命令である。並列環境下で動作するため、未束縛であることの検査と値の書き込みが不可分に行なわれる。また、その変数の束縛を持ってサスペンドしているゴールがあった時には、それを検出してアクティベートを指示する ACTIVATES コマンド [2] を NIP に向けて発行する。更にリモートな変数であった時には初めから NIP に向けて BIND コマンド [2] を送出する。

## 4 命令セットの評価

まず、幾つかのサンプル・プログラムでの命令の実行数の割合について、概要を示す。用いたサンプル・プログラムは nreverse 30 (長さ 30 のリストの naive reverse), qsort 50 (長さ 50 のリストの quick sort), primes 100 (100 までの素数の生成), 8 queen (8 クイーン問題), bestpath (36 ノード中の 1 ノードと他ノードの間の最短経路問題) である。ロード/ストア命令、バインド命令、条件ジャンプを含むジャンプ/コール命令、エクセキューション命令とその複合命令、デレファレンス命令とその複合命令、ヒープ領域のアロケート命令、算術演算命令のそれぞれの実行数の割合について、UNIREDI 単体でのシミュレーションにより測定した。結果を表 1 に掲げる。クロックを 10MHz (設計値) とした時の値である。UNIREDI はゴールの引数をレジスタに置くレジスタ・マシンとなっているため、ほとんどのプログラムでゴールの引数をメモリからレジスタへ読み書きするロード/ストア命令が多くなっている。また、エクセキューション命令、デレファレンス命令の実行割合もおおよそ 10~20% 弱あり、これらの専用命令を設けた意味も大きいことがわかる。

次に、UNIREDI の命令セットの中で最も特徴的なこのデレファレンス/エクセキューション命令の効果を確かめるために、これらの命令の使用範囲を順に変えて性能の変化を測定した。即ち、デレファレンス命令の処理を図 1 に示したように分解して実現した場合、基本的な dereference (derf) 命令のみを用いた場合、複合命令の内、タグを検査する dereference\_and\_check\_list (dfcl), dereference\_and\_check\_vector (dfcv) 及び定数と

表2: デレファレンス/エクセキューション命令の効果

使用した 命令範囲	none	+derf	+dfcl, dfcv, dfcc	+exel	+dcll, dcvl, exll	total
nreverse 30	608.3	670.5	726.0	768.4	913.9	-
	-	10.6	9.2	7.0	24.0	50.7
qsort 50	367.0	403.4	438.1	457.3	469.5	-
	-	9.9	9.5	5.2	3.3	27.9
primes 100	166.1	173.0	174.7	175.5	176.8	-
	-	4.1	1.0	0.5	0.8	6.4
8 queen	454.8	504.6	539.4	558.7	592.0	-
	-	11.0	7.7	4.2	7.3	30.2
best- path	173.7	193.3	201.4	202.9	210.9	-
	-	11.3	4.7	0.9	4.6	21.4

\*上段は性能 (KRPS), 下段は向上比 (%)

比較する dereference\_and\_check\_constant (dfcc) 命令を用いた場合、エクセキューション (exec) 命令との複合命令として execute\_on\_list (exel: exec + dfcl) 命令も用いた場合、最後に構造データへの参照であればその先頭ワードを読み出す dereference\_and\_check\_list\_load\_car (dcll), dereference\_and\_check\_vector\_load\_top (dcvl) 命令及び execute\_on\_list\_load\_car (exll: exec + dcll) 命令を用いた場合について調べた。結果を表 2 に示す。表 2 では、算術演算命令の実行割合が多い primes 100 以外ではそれぞれおおよそ 3~10% 程度の割合で性能が向上しており、またデレファレンス/エクセキューション命令全体で約 20~50% の性能向上が得られている。これから、デレファレンス/エクセキューション命令とその複合命令を設けたことによつて、充分効果を得ることが出来たとと言える。

## 5 終わりに

並列推論マシン PIE64 の推論プロセッサ UNIREDI の命令セットの評価について述べた。UNIREDI は最終的には CMOS ゲートアレイ (富士通製 1.2μ, SOG タイプ) として実現される。ゲート数は約 42400, 端子数 256 (信号線数 212) である。今後の課題としては、より大規模なプログラムによる性能評価、実チップでの評価、また PIE64 における 64 台での並列動作の評価が挙げられる。

なお、本研究は文部省科研費補助金 (特別研究員奨励費) No.03001269 による。

## 参考文献

- [1] Koike, H. and Tanaka, H.: "Multi-Context Processing and Data Balancing Mechanism of the Parallel Inference Machine PIE64" Proc. of FGCS'88, ICOT, 1988.
- [2] 清水, 小池, 田中: "PIE64 のネットワーク・インターフェース・プロセッサ LSI の詳細", 情報処理学会 計算機アーキテクチャ研究会 87-5, 1991 年 3 月
- [3] 島田, 下山, 清水, 小池, 田中: "推論プロセッサ UNIREDI の命令セット" 情報処理学会 計算機アーキテクチャ研究会 79-5, 1989 年 11 月
- [4] 島田, 小池, 田中: "推論プロセッサ UNIREDI: シミュレーションによる性能評価" 並列処理シンポジウム JSPP '91, 情報処理学会 1991 年 5 月