

マイクロアーキテクチャの動的評価法

4L-6

川村 浩正

渡邊 和文

NTTヒューマンインタフェース研究所

1. はじめに

計算機アーキテクチャの評価方法として、ソフトウェアシミュレーションによる場合は、シミュレーションに多大な時間が必要となる。これに対してハードウェアを用いた評価方法は、短時間で評価結果を得ることができる。本報告では、ハードウェアによるデータの収集手段とソフトウェアの解析プログラムの組みあわせにより、マイクロプログラムを動的に解析し、計算機アーキテクチャを評価する方法について報告する。

また、この方法をLISPマシンELISに対して適用した結果について述べる。

2. 動的解析の方法

動的解析をするためのデータとして、制御記憶のアドレ스트レースを測定し、それに基づきマイクロ命令の時系列を求める。このマイクロ命令時系列から、資源や提供された制御・演算機能の利用率を求め、評価を行なう。

2.1 アドレストレースの測定法

アドレストレースの測定は、データ格納用として大規模な主記憶をもつ計算機(例えば、ELIS)を用い、被測定機で発生した制御記憶のアドレスをデータ格納用計算機の主記憶にハードウェアで書き込むことを行なう。これにより実時間で測定を行なうことができる。次に、データ格納用計算機上で制御記憶のアドレスとマイクロ命令の対応関係を用いて、個々のアドレスからマイクロ命令を求める。これをアドレストレースすべてについて行なえば、マイクロ命令の時系列が得られる。

2.2 解析方法

マイクロ命令は、ハードウェア資源の最小単位の操作命令の組み合わせからなり、その中に資源(レジスタ、メモリ等)の指定、および演算・操作、実行に関する制御情報が記述されている。この情報をマイクロ命令から抽出し、個々の資源とその操作の出現頻度を計数して、資源、および用意された機能の利用率を算出することにより、計

算機のマイクロアーキテクチャを詳細に評価することができる。

3. LISPマシンELISへの適用

3.1 データ収集

上で述べた評価方法をLISPマシンELIS[1]のアーキテクチャの評価に適用してみた。制御記憶のアドレストレースの測定では、データ収集用計算機としてもう1台ELISを用意し、試作した測定用基板を介して取りだした被測定機の制御記憶アドレスを、データ収集用ELISの主記憶に直接書き込むことにより実現した。測定は、関数の入口で開始し、出口で停止させ、対象とする関数プログラム中で実行されたすべてのマイクロ命令を収集した。今回は対象プログラムとして、TPU3[2]と呼ばれる実用プログラムに近い、比較的規模の大きいベンチマークプログラムを走行させた。これより得られたマイクロ命令の総数は、430万ステップであった。

3.2 測定結果

測定により得られた操作命令の利用率をTbl.1に示す。ELISは、マイクロプログラム制御のプロセッサであり、タグアーキテクチャ、大容量(32K語)ハードウェア・スタック、メモリ汎用レジスタ等を有している。ELISのデータバス構成をFig.1に示す。

(1) バス制御

バス制御が指定されたマイクロ命令は、全体の約5%である。内訳はタグ部とポイント部の分離演算のためのものが47%、残りがバイトデータ、2バイトデータの処理のためのものである。この例では、バイトデータ、2バイトデータの扱いが少ないが、文字列処理を伴うようなプログラムを実行させたときは、バイトデータの扱いが増えると考えられる。

(2) ALU演算

ALUを使用するマイクロ命令の割合は、全体の約93%(転送を含む)である。ELISでは、単純な論理演算、加減演算、シフト演算以外に、乗除演算機能を有している。しかし、この結果には、乗除演算機能を使用したマイクロ命令は

出現していない。

(3) 分岐制御

条件付き分岐命令は、全マイクロ命令の約45%である。タグを用いた条件分岐は、そのうちの58%であり、全体のマイクロ命令の1/4を占める。したがって、データ型による処理のディスパッチが頻繁に起きていることがわかる。

(4) メモリアクセス制御

メモリアクセスをふくむマイクロ命令は、全体の約9%である。理想的には、30%までメモリアクセスをふくむマイクロ命令を増やしても性能のネックにはならないような設計がされており、その状態からみれば、3割程度の負荷状態である。

また、メモリアクセスを含むマイクロ命令中半数以上は、タグによりアクセス制限されたものが使用されている。

(5) スタック操作

スタック操作を行なうマイクロ命令は、全体の45%である。したがって、マイクロ命令の半分

近くが、なんらかの形でスタックをアクセスしていることになる。

4. おわりに

計算機アーキテクチャを詳細に評価する手法として、マイクロプログラムの動的な特性を測定し、資源、および操作の利用率を算出して、比較する方法を述べた。そして、これをELISに適用することにより、詳細なアーキテクチャの評価ができることがわかった。今後は、他のプログラムでの解析を加え、総合的な評価を行なっていきたい。

参考文献

- [1]渡辺、石川、山田、日比野: 知能処理用ワークステーションELIS、研究実用化報告、Vol.37、No.2、pp.129-136、1988
- [2]C-L. Chang and R. C-T. Lee., Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1973

機能	ALU		Path		Stack		条件分岐		メモリ	
Total(%)	93		5		45		45		9	
内訳(%)	算術	97	tag	47	sp	11	tag	58	tag付read	69
	シフト	3	byte	53	内容	89	演算結果	8	read	27
	乗除	0	/half				状態	34	write	4

Tbl.1 各機能の出現割合

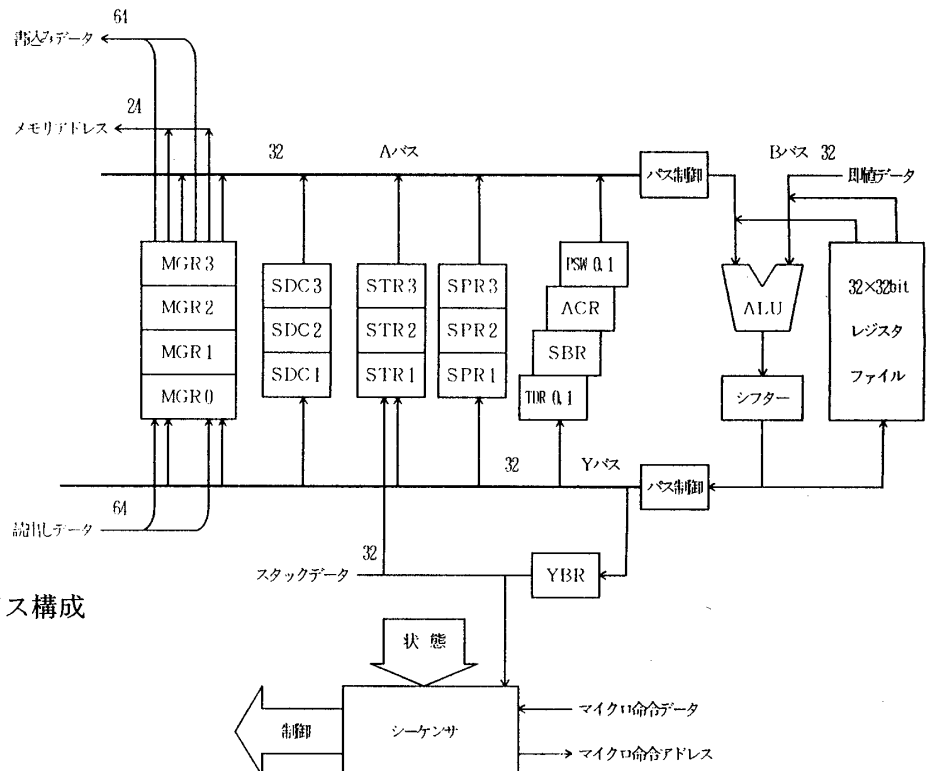


Fig.1 データバス構成