



人工知能向き高級言語マシンの方式研究

瀧, 和男

(Degree)

博士 (工学)

(Date of Degree)

1987-01-23

(Date of Publication)

2010-01-08

(Resource Type)

doctoral thesis

(Report Number)

乙1059

(URL)

<https://hdl.handle.net/20.500.14094/D2001059>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



氏名・（本籍）	たき 瀧	かず 和	お 男	（兵庫県）
学位の種類	工 学 博 士			
学位記番号	工博ろ第8号			
学位授与の要件	学位規則第5条第2項該当			
学位授与の日付	昭和62年1月23日			
学位論文題目	人工知能向き高級言語マシンの方式研究			

審 査 委 員	主査 教授	前 川 禎 男		
		教授 松 本 治 彌	教授 平 野 浩太郎	
		教授 藤 井 進		

論 文 内 容 の 要 旨

本論文では人工知能向き言語であるLISPおよびPrologを効率良く実行する2つの専用マシンの試作と評価をとおして、

- (1) LISP言語およびProlog言語の高速実行向きアーキテクチャとはいかなるものか。また両者に違いはあるか。
- (2) 実用マシンを指向したときのアーキテクチャ上の特徴は何か。また実験マシンに比べてのハードウェア量の増加はいかなる程度か。
- (3) インタプリタ向きに設計した専用マシンとコンパイルコードの実行向きに設計した専用マシンの性能差はどの程度か。
- (4) 同規模のハードウェアで構成された専用マシンと汎用マシンの性能差は、LISPやPrologに限った場合はいかなる程度となるか。

の4点に関する検討と考察を行ない、人工知能向き高級言語専用マシンを設計するに当たっての採用すべきアーキテクチャ、実現できる速度、ハードウェアコストに関する指針を与え、また汎用計算機上の処理系との性能差に関する目安を提示した。

人工知能向き高級言語専用マシンに対する要求は、高速化、メモリの大容量化、パーソナル化、対話型入出力装置の強化であり、実用化に当たってはこれらのすべてを満たす必要がある。またこれらはLISPマシンとPrologマシンに共通の要求である。高速化を実現するためには、言語とマシンハードウェアの間のセマンティックギャップを縮める必要があり、その方法として、

- 言語向きハードウェアアーキテクチャの設計
- マイクロプログラムによる最適化と適切な命令レベルの設定
- 最適化コンパイラの設計

の3種類の組合せが必要である。マシン実現のための基本方式のうち、言語の実行メカニズムとしては、設計の容易なマイクロプログラムによるインタプリタ方式と、最適化により高速化が図れるコンパイラ方式がある。またハードウェアの実現形態としてはパーソナル形の実用マシンに適するスタンドアロン方式と、設計コストが低いため実験マシンに適するバックエンド方式があり、目的に応じて使い分けることが必要である。

最初に試作した LISP マシンでは、LISP の高速実行向きアーキテクチャの研究を主目的として、設計工数の低いバックエンド方式を使用した。2台目の逐次型推論マシンでは、Prolog を拡張した言語の高速実行向きアーキテクチャを追求するとともに、マシンの実用性も重視してスタンドアロン方式を採用した。また両マシンとも最初の言語実行メカニズムには、設計の容易なマイクロプログラムによるインタプリタ方式を採用した。完成したマシン上で第2回 LISP コンテスト、第1回 Prolog コンテストの課題プログラムと若干の応用プログラムを実行させ、性能とアーキテクチャの評価を行なうとともに、LISP 向きと Prolog 向きのアーキテクチャの比較検討と、実験マシンと実用マシンの相異点に関する考察を行なった。

性能に関する結果は、両マシンともインタプリタ方式でありながら、汎用大型マシン上のコンパイルコードの実行時間と同等かそれ以上の性能を示した。高速性を解明するため、言語実行向きに導入したハードウェアの動特性を測定したところ、効果の大きかったハードウェア機能は LISP マシンでも逐次型推論マシンでもほぼ共通しており、

- スタックアクセスの高速化機能
- ガーベジコレクション用のビットを含むタグアーキテクチャとタグ値にもとづく多方向分岐機能 (LISP マシンではマッピングメモリで実現)
- 多様な条件分岐機能
- スタック以外のメモリアクセスの高速化機能
- マイクロ命令レベルでの並列制御の機能

の5つであった。また LISP 向きと Prolog 向きで異なるのは

- LISP ではハードウェアスタックが効果的なのに対し、Prolog ではスタック上でのデータアクセスの分散とスタック本数の多さからキャッシュメモリの方が適すること。
- Prolog では実行管理情報が LISP より多いことからレジスタファイルを多く用いる傾向にあること。またフレームバッファの実現やコンパイルコードの実行でもレジスタファイルを多用すること。

の2点であった。この中で Prolog の応用プログラムの実行でキャッシュヒット率が96%以上あったことや分岐命令の頻度が LISP マシンで40%、逐次型推論マシンで70%もあるなどの興味深いデータが得られた。また逐次型推論マシンで実用化のために導入された機能としては、

- ページングによる大容量メモリの管理と効率的再配置の機能
- マルチプロセスのサポート機能
- 入出力バスの制御と割込み機能
- メモリエラー訂正とエラー検出および保守とデバッグの機能

があり、これらを実現するためのハードウェアはマシン全体の約40%を占め、ハードウェアの追加とファームウェアの複雑化による性能低下は20%余りとなった。

次にそれぞれのマシンに対して言語向きの中間言語命令とそれをターゲットとするコンパイラを設計した。LISPマシン上のコンパイラはスタック上で、逐次型推論マシン上のコンパイラはレジスタ上でデータの受渡しを行なう方式とし、各々最適化処理を行なわせた。先に述べたものと同じ評価用プログラムにより実行時間を測定した結果、インタプリタに比べLISPマシンでは1.74から2.86倍、逐次型推論マシンでは1.74から3.36倍の高速化が達成された。またLISPマシンではマイクロコードへのコンパイラを検討し、中間言語へのコンパイラに比べてさらに2.8倍から3.2倍高速化されることを確認した。両処理系とも、もっとも実用レベルに近い評価用プログラムでは、インタプリタとコンパイルコードの実行時間比が約2倍となった。コンパイルコード実行のために、命令の先読み回路や、オペランドを切出してスタックやレジスタファイルの指定に使う機能を導入すると、さらに1.5倍程度の高速化が可能と考えられた。したがってインタプリタの高速実行向きに設計した専用マシンとコンパイルコード実行向きに設計した専用マシンの性能差は、LISPでもPrologでも約3倍程度になると考えられる。この比率は汎用マシン上のインタプリタとコンパイルコードの実行時間比に比べて十分小さいものであり、もしもインタプリティブな実行を必要とする応用がある場合には、マイクロプログラム化インタプリタ方式による専用マシンでも速度的に見て十分に利用可能な範囲にあるといえる。

最後に両専用マシンのコンパイルコードの処理速度を、CPUの実現方式が(多段実行パイプラインを持たない点で)近いVAX 11/785とDEC-2060の2つのシステム上の処理系と比較した。各々の処理時間を各マシンのCPUサイクルタイムで正規化したときの専用マシンと汎用マシンの性能差は、LISPとProlog、VAXとDEC-2060によらず6倍前後という値が得られた。すなわちCPUのサイクルタイムと実現方式が同じような、専用マシンと汎用マシンを比較した場合、LISPやPrologに関する性能差は6倍前後であるという目安を得た。この値は専用マシンのハードウェアをコンパイルコード実行用にさらに最適化すれば開く傾向にあり、汎用マシン上のコンパイラにおける最適化を頑張れば縮む傾向にあるが、どちらも1.5倍以内と考えている。但し、多段実行パイプラインを備えた汎用超大型計算機のような実現方式をとる場合には、本研究で有効性の確かめられた言語実行メカニズムのハードウェアサポート、マイクロプログラムレベルでの最適化の2つが多段パイプラインハードウェア中に十分取込むことが容易でなく、6倍の高速化を達成することは難しいと考えるものである。

論文審査の結果の要旨

人工知能の研究は、人間の高度の頭脳活動を機械に代行させることを意図して推進されてきており、現在では、自然言語処理、エキスパートシステム等の実用例をはじめとして、多くの分野で研究が行なわれている。

この人工知能研究の流れの中で、LISP 言語は当初から記述言語として使用され、その研究の発展を支えてきた。また、Prolog 言語は探索問題の記述や不完全データ構造の取扱いなどの点で、LISP より強力な言語として注目されている。

これらの言語は実行時のデータ型判定などの動的処理やリスト操作によるメモリ使用のために、汎用計算機上で使用した場合、実行速度が遅く、メモリ容量の不足も問題となる。

人工知能向き高級言語マシンは、これらの問題を解決するための専用計算機であり、LISP 言語を高速で実行するのが LISP マシンで、Prolog 言語用に考えられたのが Prolog マシンである。

本論文は、これらの人工知能向き高級言語マシンの方式について研究したものであり、従来、これらの専用マシンを設計しようとした場合、選択すべき方式上の明確な指針がなく、ある規模のハードウェアを用意したときに得られる処理速度の目安もなかったところ、本論文はこの問題に対して、一応の解決を与えたものである。

本論文は、LISP 向きおよび Prolog を拡張した言語向きの二つの専用マシンの試作と評価を行うことにより、

- (1) LISP 言語および Prolog 言語の高速実行向きアーキテクチャはどうあるべきか、また、両者に差が存在するか。
- (2) 実用計算機を指向したとき、実験用試作計算機に比して、アーキテクチャの特徴は何か、また、実験マシンに比べてハードウェアの量の増加はどれ位か。
- (3) インタプリタ向きに設計した専用マシンとコンパイルコードの実行向きに設計した専用マシンの性能差はどの程度か。
- (4) 同規模のハードウェアで構成された専用マシンと汎用マシンの性能差はどの程度か。

について、検討と考察を行い、設計上のアーキテクチャに関する指針と、性能やハードウェア・コストに関する数値的な目安を与えている。

論文構成としては、全体で6章から成るが、第2章では、人工知能向き高級言語用専用マシンに対する要求項目と、専用マシン実現のための基本的方式を示し、マイクロプログラムによるインタプリタ方式と、コンパイル方式の長所、欠点、また、スタンドアロン方式とバックエンド方式の各々の特徴について説明している。

第3章では、LISP 言語の高速実行向きアーキテクチャの研究を主目的に、バックエンド方式とマイクロプログラム化インタプリタ方式による LISP マシンを試作し、その性能とハードウェア・アーキテクチャに関する評価を加え、考察のためのデータを提供している。この試作機は極めてすぐれた性能を示し、この種の計算機の設計の指針を与えた。

第4章では、Prolog 言語を拡張した言語を指向した専用マシンについて、高速化と実用化の両者を目的に、スタンドアロン方式とマイクロプログラム化インタプリタ方式で設計し、試作、評価した結果をのべている。この研究は、わが国の第5世代コンピュータ・プロジェクトの一環として、(社)新世代コンピュータ技術開発機構において行なわれたものである。この評価結果から、LISP 向きと Prolog 向きアーキテクチャについて、また、実用マシンのためのアーキテクチャとハードウェア・コストについて考察している。

第5章では、これら二つの専用マシンに対して、中間言語命令とコンパイラを設計し、両マシンにおける性能を評価している。また、LISP マシンについては直接マイクロコード生成型のコンパイラについても検討を加えている。また、二つの処理系の比較から、インタプリタ向きに設計した専用マシンと、コンパイルコード向きに設計した専用マシンの性能差、同規模の汎用マシン上の処理系との性能差について考察している。

以上、本論文は、優れた性能を示した試作例を通じて、人工知能向き高級言語マシンの方式設計について研究し、この種の計算機的设计上、重要な知見を得、計算機工学上価値ある集積をしたもので、工学上寄与するところ大である。

よって学位申請者瀧 和男は工学博士の学位を得る資格があると認める。