

氏 名	宮 脇 富 士 夫 みや わき ぶ じ お
学位の種類	工 学 博 士
学位記番号	論 工 博 第1201号
学位授与の日付	昭 和 54 年 7 月 23 日
学位授与の要件	学 位 規 則 第 5 条 第 2 項 該 当
学位論文題目	A P L 会 話 型 処 理 シ ス テ ム と そ の フ ァ ー ム ウ ェ ア 化 に関する研究

(主 査)
論文調査委員 教 授 萩 原 宏 教 授 大 野 豊 教 授 堂 下 修 司

論 文 内 容 の 要 旨

本論文は計算機のプログラミング言語の一つである APL の処理システムに関する研究をまとめたものである。APL は K. E. Iverson によって1957年に提案された言語を基にしている。この言語は種々の特長があり、簡潔ですぐれた記述能力があるため、会話型のプログラミング言語として注目されるようになり、IBM 社をはじめとして各所で実用に供されるようになった。

しかし、APL はプログラムの記述には極めてすぐれた能力を有しているが、その処理に当っては、実行中にダイナミックに解釈しなければならないところがあり、コンパイラ方式による処理は困難で、インタプリタ方式によって処理を行わざるを得ない。そのため処理効率が悪く実用上の難点となるが、手続き向きのプログラミング言語としてのすぐれた特長が注目されて、処理効率向上のための種々の研究が進められた。その方式の中心となるのが、本論文で取り上げている処理システムのファームウェア化である。

本論文は、まず、第1章で APL 言語の沿革について述べ、本研究の背景を明らかにしている。

第2章では処理方式やシステムの構造に影響を与える APL 言語の機能的特徴を取り上げて検討し、続いて言語処理の方式について検討して、中間表現インタプリタ方式が適していることを示し、更に、中間表現についての検討を行い、採用した中間表現の詳細を述べている。

第3章では実験に使用した二種の計算機のハードウェアの性能について述べ、続いて処理システムの構造を明らかにし、実行モードの処理部分と定義モードの処理部分を分けて、それぞれの部分を構成する主なサブシステムについて、処理の流れと機能の概略を述べている。本システムのプログラムの作成は階層構造の概念に従って行われており、使用頻度の高いサブシステムは特に機能単位として概念づけて、その抽出を行ない、その静的使用頻度を求めている。

第4章では外部データに対応する内部データは数値データおよび文字データのみとして、それぞれの内部形式およびその設定理由について述べ、他の処理システムで用いられているデータ形式とも比較している。また、本システム特有のデータについて、その形式および導入理由を明らかにしている。

第5章は本システムのアーキテクチャの主要部分である領域、表、スタックの詳細について述べている。

本システムでは、情報の性質によって格納場所を分離し、領域の構造を簡単にし、処理の統一をはかり、それによって処理プログラムが簡潔になることを示している。また、関数の独立性を考えることにより、名前表が簡潔になり、また、その管理も容易になり、名前表の有効利用が可能となることを示した。

第6章では本システムのインタプリタの動作について詳細に述べている。インタプリタは有限オートマトン方式で動作するものであり、有限個の内部状態をもっており、これを状態遷移図にまとめて、動作の全体を明確にし、また、インタプリタのアルゴリズムを示している。更に、代表的な APL 文を取りあげて処理過程を明らかにした。

第7章ではシステムの処理効率を向上させるためのファームウェア化について述べている。まず、効果的にファームウェア化するために、機能単位の動的使用頻度を測定し、また、インタプリタにおける処理時間を解析し、インタプリタ・メインルーチン、演算子処理ルーチンのファームウェア化が効果的であることを明らかにした。

第8章では前章で検討した結果を基にして HITAC-8350 においてファームウェア化した結果について述べている。まず、機能単位のファームウェア化について検討し、その効果は可成り大きな巾をもつが、機能全体をマイクロプログラムで作成したもので、5～17倍程度の処理速度の向上があることを示している。第2は構文解析およびコンパイル部分における効果であるが、それぞれ約6倍、約4倍の処理速度の向上が認められた。第3はインタプリタ・メインルーチン中でファームウェア化を行った部分について述べ、続いてインタプリタの処理を総合的にみた場合の効果について、代表的な APL 文についてインタプリタ・メインルーチンの処理時間を調べ、総合的效果として約6倍の速度向上となることを示している。最後に演算子処理ルーチンのファームウェア化について調べている。処理時間は種々の要因によって大幅に変わり、単純にその効果を判定することはできないが、特殊なものでは、100倍以上の効果が得られたものもある。

第9章は結論であって、本研究で作成したシステムの特徴的な事項、ファームウェア化によって得られた処理速度とファームウェア化の効果についてまとめ、本システム作成の経験から、ファームウェア技術、APL 処理システムについての今後に残された問題点を指摘している。

論文審査の結果の要旨

APL は計算機の会話型プログラミング言語として種々のすぐれた特長があるが、その処理に当ってはインタプリタ方式をとらざるを得ないため、処理効率が非常に悪く、実用上の難点となっている。本論文では、APL の会話型処理システムを作成して、種々の面から検討を行ない、効率向上のためファームウェア化すべき要点を明らかにし、その結果に基づいて処理システムを部分毎にファームウェア化して調べた処理速度向上の結果から、ファームウェア化の効果を確認したものであって、主な成果は次の通りである。

- (1) APL 言語の機能的特徴、言語処理の方式について検討し、中間表現インタプリタ方式が適当であることを明らかにし、中間表現として目的に適ったものを選定した。
- (2) APL の処理システムの構造について検討し、実行モードと定義モードの処理部分を分け、処理の

流れを明らかにした。

(3) 使用頻度の高いサブシステムを機能単位として抽出し、その静的使用頻度を求めた。

(4) 内部データとして数値データと文字データのみで充分であることを明らかにした。

(5) 処理システムで、情報の性質によって格納場所を分離し、領域の構造を簡単にし、処理の統一をはかることにより、処理プログラムが簡潔になることを示した。

(6) 関数の独立性を考慮することにより、名前表が簡潔になり、その管理も容易になり、名前表の有効利用が可能になることを示した。

(7) インタプリタを有限オートマトン方式で動作するものとして、状態遷移図にまとめて、動作の全容を明確にした。

(8) 処理システムのファームウェア化に当って、機能単位の動的使用頻度を測定し、インタプリタにおける処理時間を解析して、ファームウェア化が効果的である部分を明らかにした。

(9) ファームウェア化を行った結果、その効果に巾はあるが、小さいマイクロプログラムで5～17倍、大きなものでアルゴリズムを工夫したもので20～28倍、特殊なもので113～150倍の処理速度の向上を得た。

以上要するに、本論文は APL の処理システムについて検討を行い、処理方式を明確にし、ファームウェア化によって処理効率を向上させる上で効果のある点を明らかにし、実験によってその効果を確認したものであって、学術上、實際上寄与するところが少なくない。

よって、本論文は工学博士の学位論文として価値あるものと認める。