

解説

4. マイクロプログラム技術の応用



4.2 専用マシンにおけるマイクロプログラム技術†

山本昌弘†† 中崎良成†† 中田登志之††

1. はじめに

特定の応用分野向けに構造を特化することにより汎用計算機より高速化して、性能・価格比を向上させた専用マシンが各種開発されている。専用マシンにおいて、高速化や応用分野に専用化するために、マイクロプログラム技術が広く用いられている¹⁾。本稿では、専用マシンにおいてマイクロプログラム技術が導入されている背景、狙い、利用分野、マイクロプログラム・アーキテクチャおよび効果について、具体例をもとに概説する。

2. 専用マシンでのマイクロプログラミングの概要

(1) 専用マシン

専用マシンは IBM 370, VAX 780, ACOS 2000, インテル 8086 などの汎用計算機、汎用マイコンと対比させるもので、特定の応用分野に要求される処理に限定した構造をもった計算機であり、現在、非常に広範囲にわたって開発されている。専用マシンは汎用マシンと比べてとき、特定用途の処理にのみ限定することによって汎用マシンより高速化、高機能化、低価格化、拡張性、柔軟性などを徹底的に追求しており、表-1 は主な専用マシンを示している。

(2) 専用マシンでのマイクロプログラミングの利用形態

専用マシンは特定用途の処理に専用化した構造をとっており、マイクロプログラミングがきめ細かな制御、柔軟性、特定処理への専用化、設計・検査・修正の容易さなどの特徴を備えていることから専用マシン

で幅広く用いられている。表-1 はおのおの専用マシンでのマイクロプログラムの利用状況を要約したものである。詳細は以降で論じるが、専用マシンの導入の狙いや特徴の主なものとして以下の点があげられる。

- 要求される処理・アルゴリズムを高速に実行する
- ハードウェアでは実現困難な複雑な処理・アルゴリズムをソフトウェアよりも効率よく実現する
- 処理・アルゴリズムの変更・拡張を容易にする
- 高速化を実現するため 100 ビット程度の水平マイクロアーキテクチャが多い
- 専門家向けマイクロアーキテクチャである

3. AI マシンのマイクロプログラミング

AI マシンは各種エキスパートシステムを始めとする知識処理システムを開発あるいは利用するために必要な高性能かつ優れた開発環境を提供する専用マシンである。AI マシンは知識処理システムを記述するための核言語になっている知識処理指向高級言語や知識表現モデルを処理対象にしている。知識処理指向高級言語としては、関数型言語 LISP, 述語論理型言語 PROLOG, オブジェクト指向言語 Smalltalk, また知識表現モデルとしては、プロダクションシステム, セマンティックネットワークなどが代表的なものである。これらの高級言語を高速に処理する LISP マシン, PROLOG マシン, Smalltalk マシンや知識表現モデルを実行するプロダクションマシン²⁾が一般に呼ばれている AI マシンである。後者のプロダクションマシンにはマイクロプログラム技術として特に注目するものは少ないので、前者の 3 種の AI マシンについて述べることにする。

AI マシンには以下の特性が求められる。

- 知識処理システム構築に適するマシンであること
- 知識表現モデルおよび言語の進歩にともなって

† Microprogram Architecture for Specialized Computers by Masahiro YAMAMOTO, Ryosei NAKAZAKI and Toshiyuki NAKATA (C & C Systems Laboratories NEC Corporation).
†† 日本電気(株) C&C システム研究所

表-1 専用マシンとマイクロプログラミングの特徴

専用マシンの分野	μ -p 利用の主な目的	μ -p アーキテクチャの特徴	具体例	その他
AI マシン	<ul style="list-style-type: none"> ●ソフトのファーム化による高速化 ●開発の柔軟性 ●複数言語処理 	<ul style="list-style-type: none"> ●タグ分岐 ●スタック操作のメモリ制御機能 ●1語 100 ビット, 64K語 	<ul style="list-style-type: none"> ●Symbolics, CADR ●CHI, PSI ●ELIS ●Explorer 	<ul style="list-style-type: none"> ●大容量 μ-p メモリの chip 内組み込み ●GC 機能 ●RISC アーキテクチャとの競合
CAD マシン	<ul style="list-style-type: none"> ●アルゴリズムのファームウェア化 ●複数応用への対応 	<ul style="list-style-type: none"> ●水平型マイクロ命令 ●低レベル並列処理 	<ul style="list-style-type: none"> ●MegaLogician ●Compute Engine ●MARS 	<ul style="list-style-type: none"> ●マルチプロセッサ化による高速化 ●高級言語をマイクロプログラムへ変換
専用マイコン	<ul style="list-style-type: none"> ●設計・製造・検査容易さ ●共通化 ●きめ細かな制御 	<ul style="list-style-type: none"> ●10~40 bit\times0.5~2 KW ●垂直型マイクロ命令 	<ul style="list-style-type: none"> ●μCOM 48 ●HMCS 45 ●T 3444 ●μPD 78312 	<ul style="list-style-type: none"> ●ゲーム, 計測器, 家電, 電卓などの制御
信号処理	<ul style="list-style-type: none"> ●低レベル並列処理による高速化 ●アルゴリズムのファームウェア化 	<ul style="list-style-type: none"> ●乗算器と ALU の垂直接続 ●パイプライン処理 	<ul style="list-style-type: none"> ●μPD 77230 ●TMS 32 DC 25 ●DSP 32 	<ul style="list-style-type: none"> ●浮動小数点演算機能の採用
数値処理	<ul style="list-style-type: none"> ●応用分野の拡張 ●開発の容易性 	<ul style="list-style-type: none"> ●水平マイクロ命令 100 ビット 	<ul style="list-style-type: none"> ●GF 11 ●Warp 	<ul style="list-style-type: none"> ●マルチプロセッサ構成

AI マシンの改良, 拡充を進めるためには専用性ととも改良, 拡充を容易にできる柔軟性をもつこと

●マンマシンインタフェースが優れているインタラクティブなマシンであること

このような AI マシンにおける内部処理の特徴は以下のものがあげられる。

●文字列マッチング, 変数結合処理などに必要な内部表現データとのパターンマッチング機能

●変数結合情報および処理遷移を保存するために必要なスタックの高速操作と動的なメモリ管理

●インデクシング, レジスタ/スタック操作の最適化を実施するためのコンパイルコードの生成

前記の特徴をもつ AI マシンを実現するために, マイクロプログラムは以下の点で適している。

●言語処理方式, マシンアーキテクチャの改良, 拡充に柔軟に対応

●使用頻度の高い関数, 組み込み述語およびメモリ内の不用領域回収などシステム処理の高速化

マイクロプログラムで実現している知識処理指向の専用マシンの中から, 1970年代後半に開発された初期の LISP マシンとして注目される MIT の CADR³⁾, WAM⁴⁾ 形式のアーキテクチャに基づいて述語論理型言語指向の専用マシンを実現した ICOT と日本電気の CHI⁵⁾, オブジェクト指向言語用の専用チップをマイクロプログラムメモリ内蔵で実現した日立の AI

32^{6),7)}を以下に示す。ほかにも多くの研究開発および製品化が行われているが, それらに関しては別の資料^{8),9)}を参照いただきたい。

3.1 CADR マシン

CADR マシンはビットマップディスプレイをベースにしたウィンドウシステムを実現しているスタンドアロン形式のマシンである。CADR は MACLISP で記述したプログラムを DEC KA-10 の3倍の性能で実行できる。CADR プロセッサのハードウェアは図-1に示すように, ALU, シフト/マスク回路による32ビットのデータ操作機構を核にして, 32語と1K語のレジスタファイルおよび1K語のプッシュダウンリストメモリをプロセッサ内データ保持用に備えている。

これらのデータ操作ハードウェアは48ビット \times 16K語の制御メモリに格納されたマイクロプログラムで制御される。マイクロプログラムのアドレス決定用に, ディスパッチメモリ, マイクロアドレススタックが利用される。図-2にCADRマシンのマイクロ命令形式を示す。マイクロ命令としては二つのソースデータ操作と一つの格納先指定が可能で, 以下の4種類にわけられる。

- 演算時に利用する ALU 型
- バイト, ビット操作するための BYTE 型
- 条件分岐, call/return などの分岐制御に利用す

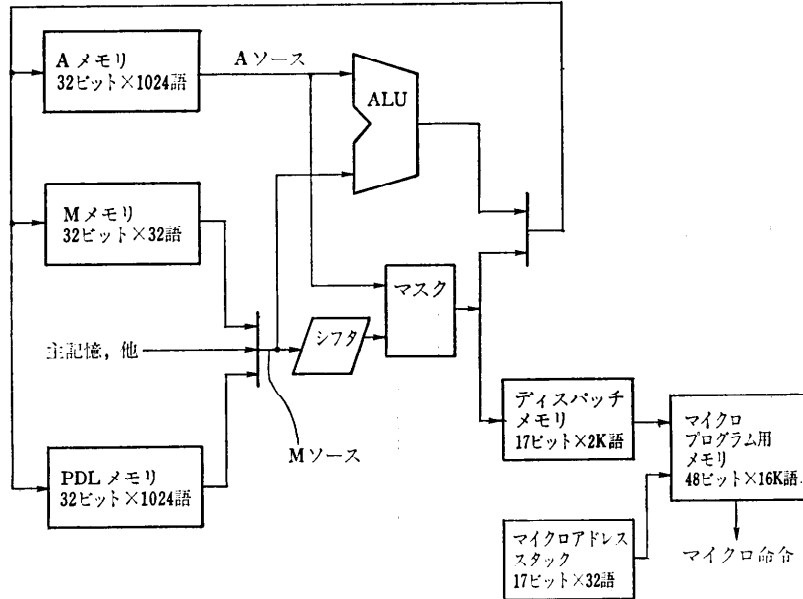


図-1 CADR マシンのプロセッサ構成

47	42	32	26	0
A ソース 指定		M ソース 指定	ALU マイクロ形式	
			JUMP ”	
			DISPATCH ”	
			BYTE ”	

マイクロ命令タイプ(ALU, JUMP, DISPATCH, BYTE)
評価, クロック制御
他

図-2 CADR マシンのマイクロ命令形式

る JUMP 型

- ディスパッチメモリを用いて任意のマイクロアドレスへの分岐を可能にする DISPATCH 型

3.2 CHI マシン¹⁰⁾

CHI マシンは入出力関連装置がホストプロセッサによって制御されるバックエンド形式のマシンで、マシンの基本的な内部アーキテクチャは、述語論理型言語のユニフィケーション操作を記述できる程度の高さを有するとともにコンパイラによる最適化を十分にできる程度に基本処理要素に分解した機械語命令を備えることが特徴である。

CHI ハードウェアは図-3 に示すように、データ演算用およびアドレス演算用の2種類の ALU、タグの比較あるいは操作を実行する引数比較・操作器、マルチポートの読み出し・書き込み可能なレジスタファイルをデータ処理の基本機能として備えている。データ

処理単位である1語は8ビットのタグ関係情報と32ビットの値部で構成される。

これらのデータ操作用ハードウェアは78ビットX16K語の制御メモリに格納されたマイクロプログラムで制御される。図-4はCHIのマイクロ命令形式を示す。マイクロプログラムのアドレス決定用に1個のタグ情報で分岐先を決定するタグ分岐メモリ、2個のタグ情報を同時に参照して分岐先を決定するタグディスパッチメモリを備えている。この分岐機能は1マシンサイクル内でタグを解析できるので、タグアーキテクチャを採用しているCHIの高速化に有効である。CHIの高速化のために、マイクロ命令において、以下の処理が同時に並列処理可能である。

- 主記憶からの命令先取り用の命令読み出し
- 主記憶アドレス用のレジスタ更新
- レジスタファイル内のスタック制御ポインタ更新
- レジスタファイル内のデータ演算・操作
- マイクロ分岐

3.3 AI32

AI32は最新のVLSI技術を利用することにより、コプロセッサ形式の知識処理指向32ビット高速プロセッサを1チップで実現している。Smalltalkを主な処理対象言語にしているが、LISP, PROLOG系の言語処理も可能にしつつある。

チップ内のプロセッサ構成は図-5に示すように各

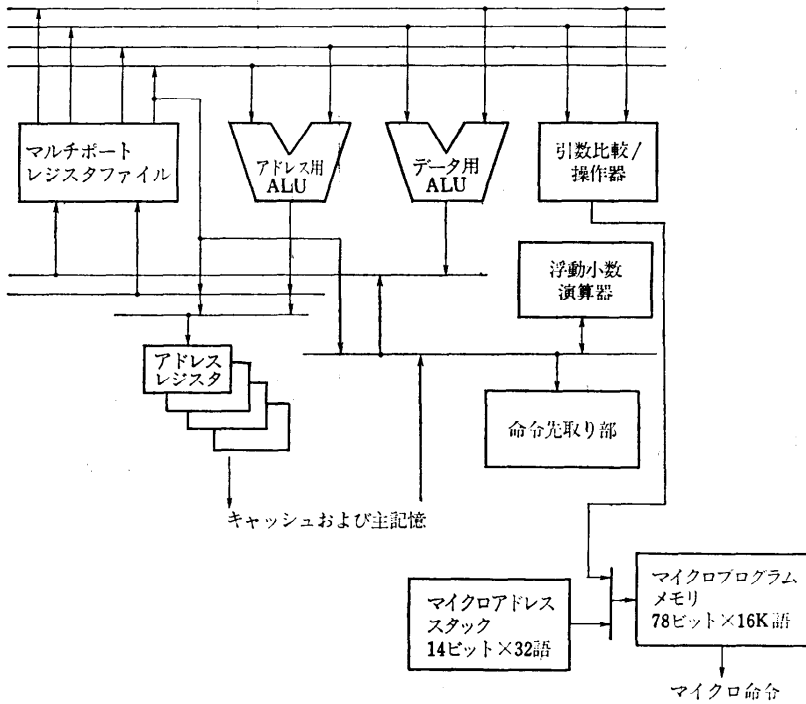


図-3 CHI のハードウェア構成

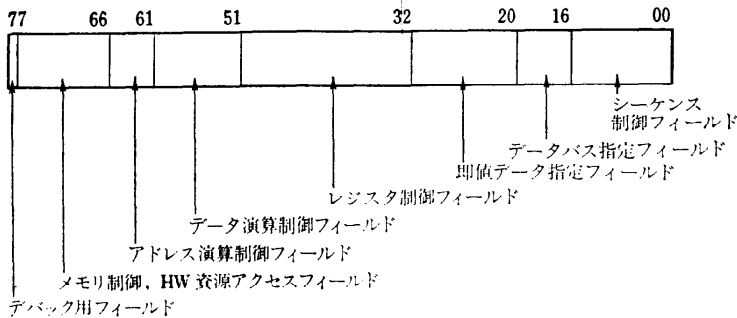


図-4 CHI のマイクロ命令形式

種のスタックの一時記憶として利用できる2種のレジスタファイル、データ演算およびタグフィールドの高速操作と検査を実施するALU、シフトおよびマスクを備えている。そのほか、内部スタックのオーバフロー検出機能、バスを介して結合しているホストプロセッサとの通信機能を備えている。

AI32では、これらのハードウェアを64ビット×4K語のEPROMに格納されたマイクロプログラムで制御している。図-6はAI32のマイクロ命令形式を示す。マイクロプログラムのシーケンス決定機能の

一つとしてタグ分岐を実現しているほか、マイクロアドレススタックも内蔵している。また、プロセッサの汎用性を高めるために、命令先取りはハードウェアではなく、マイクロプログラムで制御している。

AI32はAI指向の言語に共通のタグ操作、スタック操作の基本的なハードウェアを備え、言語間での違いはマイクロプログラムで吸収することが大きな特徴である。