

解説



機能メモリのアーキテクチャとその並列計算への応用

3. 機能メモリを用いた
計算機アーキテクチャ†

長沼次郎†† 小倉 武††

1. はじめに

コンピュータの応用分野の拡大と処理量の増大にとまらぬ、より高速なコンピュータへの要求が高まっている。コンピュータの高速化は、コンパイラ技術の進歩、デバイス技術の進歩とともに、パイプライン処理の導入、キャッシュメモリの導入、処理の専用ハードウェア化、並列処理化などのマシンアーキテクチャ技術の進歩によって達成されてきた。しかしながら、これらのコンピュータをメモリ技術の観点からみた場合、プロセッサとメモリを分離した従来のマシンアーキテクチャを踏襲した構成となっている。このような従来のマシンアーキテクチャでは、本来の情報・通信処理のメモリアクセスに内在する多くの並列性を活かした高速化を図ることが難しい¹⁾。

メモリアクセスに内在する並列性は、たとえばデータに対する記憶と並列処理の二つの機能を実現する連想メモリに代表される機能メモリを用いることにより、自然に引き出すことができる。このため、このような機能メモリをコンピュータの中心的な処理に用いることにより、メモリアクセスに内在する並列性を活用し、制御の簡単化とその飛躍的な高速化を図れる可能性がある。

一方、機能メモリの代表である連想メモリ技術に関しては、半導体集積回路技術を駆使した各種大容量連想メモリ LSI が実現され、その実用性が高まりつつある^{2),3)}。本稿では、機能メモリの代表として連想メモリを取り上げ、その特徴と計算機アーキテクチャにおける応用形態を整理するとともに、それらを用いた種々の計算機アーキテクチャを紹介し、その中での機能メモリ（連想メモ

り）の役割を考察する。また、連想メモリを活用した計算機アーキテクチャの実例として、4-kbit 連想メモリ LSI を用いて試作した Prolog マシンの概要を示す。

このような連想メモリに代表される機能メモリ技術をベースとする新たなコンピュータは、種々の情報・通信処理システムの構築にきわめて有用であり、今後の VLSI 技術の進展とともに、より効果的なものとなる。

2. 機能メモリと計算機アーキテクチャ

2.1 機能メモリの概要

機能メモリとは、書き込みおよび読み出し機能をもつ通常のメモリに対し、記憶機能と入出力との間に目的に応じて必要な機能を付加したメモリの総称である⁴⁾。付加した機能に着目して、現在までに実現されている主な機能メモリを表-1 に示す。詳細については、本特集号解説記事あるいは文献 5)~7) を参照されたい。ここでは、機能メモリの代表として連想メモリの概要を簡単に示す。

通常のメモリ (RAM) では、アドレスを用いてデータアクセスを行う。これに対し連想メモリで

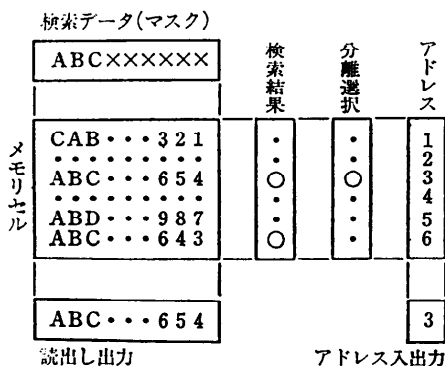
表-1 付加機能に着目した機能メモリの分類

付 加 機 能		品 種
アドレス生成論理の取込み	内容によるアクセス (内容がアクセスキー)	連想メモリ
	順序によるアクセス (順序がアクセスキー)	FIFO, スタック
	複数アドレスによる同時アクセス	マルチポートメモリ
	応用向き高速アクセス	画像メモリ, キャッシュ
データ変換論理の取込み	入出力順序の並び換え	時間スイッチ
	時間情報の位置情報へのマッピング	タイムメモリ
	データの高信頼化	誤り訂正機能付きメモリ
	データ変換	連想メモリ

† Computer Architectures Based on Functional Memory by Jiro NAGANUMA and Takeshi OGURA (NTT LSI Laboratories).
†† NTT LSI 研究所

は検索データを入力し、これと各ワードの記憶データの内容を照合、検索し、該当する内容をもつワードに対してアクセスを行う。連想メモリのもつ検索機能としては、検索データと記憶データとが一致しているか否かを照合する機能（一致検索機能）が最も一般的であるが、大小関係による検索機能⁸⁾、ベクトル距離によるあいまい検索機能⁹⁾などの関係検索機能も提案あるいは実現されている。

一致検索機能を有する連想メモリ概念図を図-1に示す。検索データと各ワードの記憶データとの照合が全ワード同時に実行され、一致検索が成立したワードに一致検出を示すフラグが立てられる。必要に応じて、一致検索が成立した複数のワードの中から一つのワードを選択してアクセスする。現在までに実現されている主なワードに対するアクセス機能としては、選択ワードの逐次書き込み／逐次読出し、選択／非選択ワードへの同時並列書き込み、選択／非選択ワードの不要化などがある。連想メモリ LSI の場合、RAM と比較すると小容量であるため、資源の有効利用の観点から、不要ワードの管理機能も重要なポイントとなる¹⁰⁾。このような連想メモリ LSI の検索機能、アクセス機能は、連想メモリ LSI のアーキテクチャを特徴付けるとともに、連想メモリ LSI を用いたシステムの全体性能を決定する要因となる。このため、どのような機能をもつ連想メモリ LSI を実現し、それをどのような処理に適用するのが大きなポイントとなる。



×: マスクされていることを示す

図-1 連想メモリ概念図

2.2 計算機アーキテクチャにおける機能メモリの応用形態

機能メモリ（連想メモリ）の最も単純な応用は、アドレス変換などを行うテーブル検索である。しかし、連想メモリ LSI の大容量化、高機能化にともない、単純なテーブル検索にとどまらず種々の応用が可能となってきた。これらの連想メモリの応用分野としては、以下に示すように多くの提案がなされており、知識情報処理向きの VLSI アーキテクチャの一候補としても期待されている^{11), 12)}。

- (1) 仮想記憶、キャッシュなどにおけるアドレス変換テーブル¹³⁾
- (2) データフローマシンにおける制御用メモリ^{14), 15)}
- (3) データベースマシンのデータメモリ^{16), 17)}
- (4) Lisp マシンにおけるセル格納用メモリ、メソッドテーブル^{18), 19)}
- (5) Prolog マシンにおける各種高機能スタックなど^{20)~25)}
- (6) 図形処理用データメモリ²⁶⁾
- (7) 画像処理用フレームメモリ^{27), 28)}
- (8) 人工知能用マシンのデータメモリ²⁹⁾

これらの各種提案における連想メモリの応用技術を連想メモリの応用形態および連想メモリに付与すべき機能の観点から整理したものが表-2である。

単純なテーブル検索においては、連想メモリの代替技術としてハッシュ技法が知られている。ハードウェアによるハッシュ機構を用いた場合、一致検索、データ読出しに関しては、連想メモリと同程度の速度性能が得られる。一方、連想メモリでは、選ばれた複数のワードのデータを同時に書き換えること（並列書き込み機能）が可能のため、この種の動作の速度性能では連想メモリが優位に立つ。

データフローマシン、Lisp マシンなどでは、複数の検索フィールドに対して、マスクデータを用いて検索対象となるフィールドを切り替えながら逐次検索を行う処理が不可欠である（マスク検索、タグ検索）。連想メモリの機能としては、マスク検索機能、並列書き込み機能とともに、指定したフィールドのデータのみを書き換えられる部分書き込み機能が有用となる。データベースマシ

表-2 連想メモリの種類々の応用形態

連想メモリ装置の応用形態			代替技術・備考
テーブル検索	単純検索	検索フィールドが単一	重複データの格納・無 → 代替技術 → ハッシュ
			重複データの格納・有 → 並列書き込み機能が有用
	マスク検索	複数個の検索フィールドをもつ、マスク機能、部分書き込み機能が必要不可欠となる	
	タグ検索	マスク検索の一種、タグフィールドで検索したのち、選択されたデータに対して、種々の処理を行う	代替技術 → なし (従来プロセッサ+ソフト) でシリアル処理
ワード並列ビット直列演算	大小比較、加算などの処理をワード並列ビット直列演算で実現する。処理時間がワード数に依存しないため、大量データの一括処理に適している		
複合処理	タグ検索+ワード並列ビット直列演算などの複合処理を行う。最も一般的な処理形態である		

ン、Prolog マシン、図形処理用データメモリなどでは、タグ検索+ワード並列ビット直列演算などを行う複合処理の機能が不可欠になる。

ワード並列ビット直列演算とは、連想メモリに格納されているデータに対して大小比較、最大値/最小値検索、ソーティング、加算などの処理を行うものであり、連想メモリのもつ“並列処理”機能の中核を成すものであり、その有用性は古くから指摘されている^{30),31)}。連想メモリを用いるとあらゆる種類の算術論理演算が実現できる。ワード並列ビット直列演算では、その時間がビット数にのみ依存しワード数には依存しないため、大量のデータの一括処理に適している。Mワード×Nビットのデータに対して従来手法ではO(M)の手順が必要であった処理が連想メモリを用いるとO(1)、O(N)あるいはO(N²)の手順で実現できる。一般に、M≫Nであり、連想メモリ適用の効果は大きい³²⁾。

3. 機能メモリを用いた計算機アーキテクチャ

本章では、機能メモリ（連想メモリ）の実際的な応用技術を明らかにすることを目的とし、これまでに提案あるいは開発された機能メモリを用いた計算機アーキテクチャの主なものを紹介する。

3.1 汎用マシン

汎用マシンにおける最も重要な連想メモリの利用は、仮想記憶におけるアドレス変換テーブル、キャッシュメモリの制御である¹³⁾。ここでは、仮想記憶の制御への連想メモリの利用を紹介する。用途は異なるがキャッシュの場合も原理的には同様である。仮想記憶は、主メモリの容量よりはるかに大きなアドレス空間をユーザに提供する方式

である。ユーザに見える大きなアドレス空間を仮想アドレス（virtual address）といい、実際の主メモリのアドレス空間を実アドレス（real address）という。この仮想アドレスから実アドレスへのマッピングは、マシン全体の性能を左右するため、高速化が不可欠である。

アドレスマッピングの原理を図-2に示す。アドレス空間は、1KB~4KBの固定長のページ（実メモリではページ枠）に分割する。ページ番号pとページ枠番号p'の対応が、仮想アドレスから実アドレスへのマッピングを与える。仮想アドレスは、ページ番号pとページ内のアドレスのオフセット（offset）を示す変位dにより、 $v=(p, d)$ で表される。ここで、仮想アドレス空間におけるページ番号pと実アドレスのページ枠p'をペアにしてアドレス変換テーブル（連想メモリ）に格納しておき、ページ番号pを検索データとして連想メモリをサーチし、対応するページ枠p'を得る。このページ枠p'と変位dを合わせて、実アドレス $r=(p', d)$ を得る。

このようなテーブルに連想メモリを用いない場合、ページ番号pをアドレスとしてページ枠番号

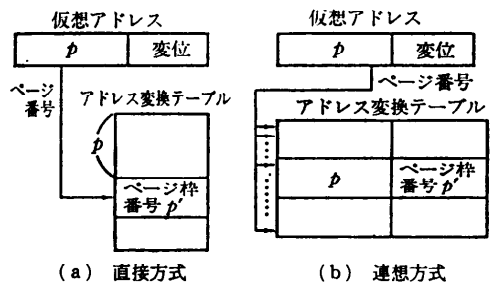


図-2 仮想記憶におけるアドレスマッピングの原理 (文献13)

P' を格納するようなアドレス変換テーブル (通常のメモリ) を用いる。しかし、これらのアドレス変換テーブルは有効な仮想アドレス空間の大きさに比例して大きくなる。このため、実際には、通常のメモリで構成された2レベル程度のページング方式を採用し、その際のオーバーヘッドを小容量の連想メモリで構成されたアドレス変換テーブルのキャッシュ: TLB (Translation Lookaside Buffer) を用いて吸収する構成が採られている。この場合、アドレス変換テーブルは実際に割り当てられた仮想アドレス空間にほぼ比例する。

3.2 データフローマシン

データフローマシンは、逐次計算モデルをベースにしたフォンノイマンマシンのボトルネックを解消し、飛躍的な高速化を達成する並列マシンとして、1974年に Dennis により提案された³³⁾。データフローの計算モデルは関数型言語と親和性が良く、プログラマは並列性を明示しなくても処理に内在する並列性が最大限に引き出されるという特徴をもっている。データフローの計算モデルにおける計算は図-3 (a) のように表現される。データフローグラフで、ノード (以下アクタと呼ぶ) は加算、掛算などの演算を表し、アーク (以下リンクと呼ぶ) はデータの入(出)力線を示す。データフローグラフにおける計算は入力リンクに

データを表すトークンを置くことによって始まる。計算を実行する規則は、以下の二つである。

- (1) 入力リンクのすべてにトークンがそろったらアクタを発火する (演算を実行する)
- (2) アクタを発火すると入力リンクからトークンを取り除き、演算結果を示すトークンを出力リンクに置く

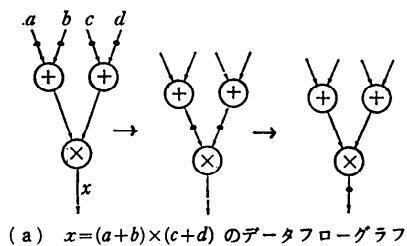
以上の規則にしたがってアクタは次々発火し図-3 (a) に示すように計算が進んでいく。

このような計算モデルを実現するハードウェアの概念図を図-3 (b) に示す。上記のような実行制御を実現するためには、入力トークンが到着するたびに、各アクタとその入力リンクにトークンが到着したかどうかを検索し、次にどのアクタが発火できるかどうかを検出する機能が必要である (発火制御)。このような発火制御は、データフローグラフを連想メモリに格納することにより、連想メモリのフィールド検索機能で容易に実現できる。このため、連想メモリを用いたデータフローマシンが数多く提案されている³⁴⁾。しかし、実際には、まだ当時、大容量の連想メモリが実現されていなかったため、トークンの到着を示すフラッグ部分のみ連想メモリを用いたり、ハッシュなどの代替手法を用いて実現されていた。

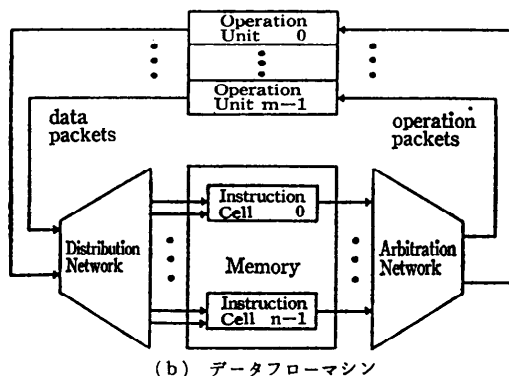
3.3 データベースマシン

連想メモリは本来データベースを操作対象とするものであるから、データベース本体の処理、その制御などに高い整合性があり、連想メモリを利用したデータベースマシンも数多く提案、試作されている¹⁶⁾。しかし、大容量の連想メモリ LSI が開発される以前のものには、データベース本体と言うよりは、ディレクトリの管理、ディスクとメモリとの間でのファイルのインデックスの管理などの制御に連想メモリを適用したものが多かった。しかし、連想メモリ LSI の大容量化にともない、データベースの本体の処理に適用されるようになってきた。

中野らは、データベースのメモリ常駐を前提に置き、連想メモリを演算器に使い、関係代数演算の超高速処理を目指した処理方式を提案している¹⁷⁾。階層型並列構成を図-4 に示す。連想メモリは大容量化してきたが、まだメモリ常駐は期待できない。このため、関係代数の個別演算のたびに、演算対象の属性データのメモリから読み出し、



(a) $x = (a+b) \times (c+d)$ のデータフローグラフ



(b) データフローマシン

図-3 データフローグラフとデータフローマシンの概念図 (文献 33)

連想メモリへの書き込み、処理結果のメモリ上での生成などを行う。関係演算の並列処理を引き出すため、図-4に示すように、連想メモリ、データベース格納用メモリなどを分割し、複数ユニット構成とし、各ユニットをSIMDモードで動作させる。このようなユニットレベルのマクロな並列性、連想メモリのタグ検索+ワード並列ビット直列演算などのマイクロな超並列性 (massive parallel) の階層型並列アーキテクチャにより、シミュレーションによると、ハッシュ方式と比較して、Join演算において2桁程度の性能向上が可能であると報告されている。

3.4 Lisp マシン

実用的な Lisp システムにおいて、実時間の塵集め (以下 GC と呼ぶ) は不可欠な機能である。通常のメモリを用いた典型的な GC は、まず、(1)すべての活性化しているデータ (セル) をトレースし、マーキングを行う、次に、(2)すべてのマークがついていないセルを塵として回収する、の二つのフェーズで実行される。しかし、この方法はメモリのサイズに依存しているため、メモリ容量が大きくなった場合、本来の処理の低下を招く。このため、J.G. Bonar らは、セルの格納に連想メモリを用いた実時間処理向けの Lisp システムを提案している¹⁸⁾。Lisp の処理過程で不要になったセルは、連想メモリのタグ検索+選択ワードの不要化 (ガーベジフラグをセットする) により、フリーなセルとして再利用される。このように、セルを連想メモリに格納することにより、連想メモリの不要ワード管理機能で、実時間 GC が実現できる。この方法はメモリのサイズに依存し

ないため、メモリ容量が大きくなっても本来の処理の低下はほとんどない。

一方、井田は、Common Lisp 上のオブジェクト指向機構である Common Loops の高速化のために、連想メモリを用いたハードウェア支援機構を開発した¹⁹⁾。D-Eval と呼ばれる Common Loops のインタプリタの中の、(a)クラスの優先順位表と (b)メソッドキャッシュに連想メモリを用いる。これらのワード形式を図-5に示す。これにより、メソッド選択時におけるクラス間の階層関係の判定とメソッドの選択が連想メモリの検索により高速化される。このハードウェア支援機構は、VMEバスを使用した MC 68020 ベースの Common Loops システムに対する機能ボードとして動作する。その 4-kbit 連想メモリ LSI³⁵⁾を用いたボードの構成を図-6に示す。連想メモリボードアクセスの高速化のため、アドレスバス上に検索情報などをのせている。連想メモリボードのサイクルは 4 MHz であり、ホストからは 250 nsec のメモリのように見える。全体として、連想メモリの適用により、ソフトウェアだけの場合の 10 倍以上の高速化が図られると報告されている。

3.5 Prolog マシン

Prolog の処理には多くの連想処理が内在している。連想メモリを用いれば、このような連想処理を、より高速に、より効率的に実現できる可能性がある。現在までに提案、実現されているプログラム格納、高機能スタックへ連想メモリを用いた Prolog マシンおよび専用ハードウェアを紹介する。

Syracuse Unification Machine (SUM) は、ユニフィケーションの高速化のための専用ハードウェア

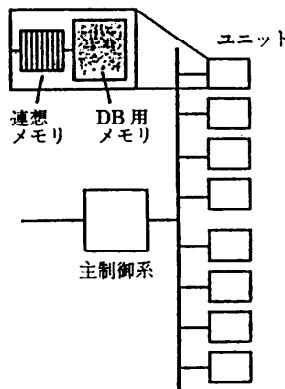


図-4 階層型並列構成のデータベースマシンの構成 (文献 17)

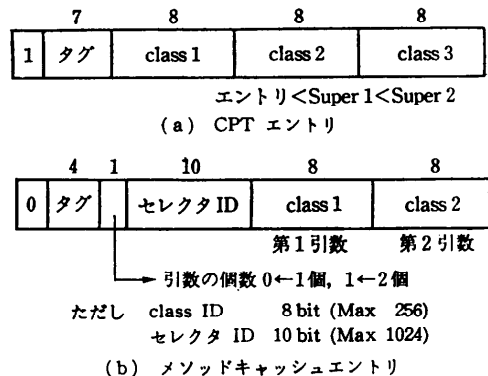


図-5 クラスの優先順位表 (CPT) とメソッドキャッシュのワード形式 (文献 19)

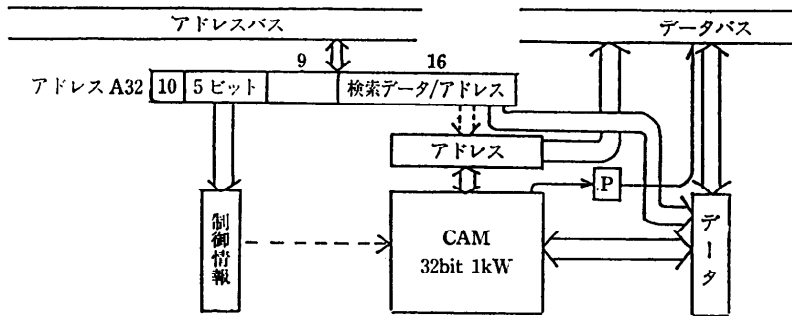


図-6 試作した連想メモリボード (文献 19)

アである²¹⁾。SUM は、連想メモリ、制御用ハードウェア、ホストマシンとのインタフェース部から構成されている。初期の SUM においては、バインド情報を連想メモリに格納し、バインド情報の高速アクセスと、専用ハードウェアの適用により、ユニフィケーション処理の高速化を図っている。また、シフトレジスタを用いて連続領域の空ワード管理を行う連想メモリ LSI (8w×3b, 128w×17b) を試作している。この連想メモリでは外部で使ったワード数をカウントし、それに基づいて連想メモリ内部のシフトレジスタを操作することにより、連続領域の空ワード管理を実現している。その構成を図-7 に示す。バックトラック時のバインド情報の消去は、格納したバインド情報の個数を記憶しておくもうひとつのスタックを用いて、このシフトレジスタのシフト量を制御することにより実現している (通常 RAM を用いた場合、変数セルのアドレス計算、トレイルスタックを用いた逐次的な undo 処理が必要である)。SUM ではその後、Prolog の節の格納および構造体の格納にも連想メモリを適用し、さらに高速化を図っている²²⁾。

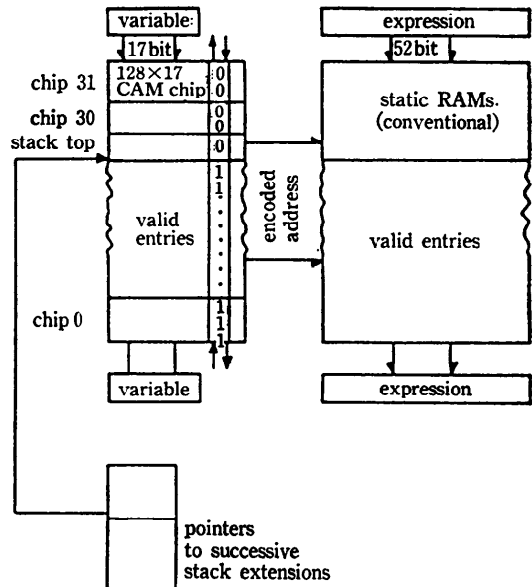


図-7 SUM における Binding Agent (文献 21)

FAIM-1 においては、Prolog における節の選択とユニフィケーションの高速化のための専用ハードウェアサポートのために、連想メモリ概念を拡張した Pattern Addressable Memory (PAM) を提案している²³⁾。その構成を図-8 に示す。この PAM にヘッド部を S 式として格納し、節の選択を行う。また、PAM を用いてユニフィケーションのサブセット (モックユニフィケーション: 変数を don't care で扱う) をサポートする。PAM を用いた節の選択は、文字列マッチングを行うため、マッチング可能な文字列の長さ按比例する処理時間が必要であるが、同一述語名が多数存在す

ようなデータベースの場合は有効である。

ASCA は筆者らが開発した連想メモリ適用と階層的なパイプライン化により処理の高速化と制御の簡便化を図った Prolog マシンである²⁰⁾。ASCA は、節の起動を行う述語処理部と引数のユニフィケーションを行う引数処理部の二つの処理部からなる。節の実行制御を行うコントロール情報とユニフィケーション時に必要なバインド情報を連想メモリに格納する。ASCA のユニフィケーションにおける連想メモリ適用手法を図-9 に示す³⁶⁾。ASCA においては、(1)引数の評価 (dereference) におけるバインド情報の検索、(2)バックトラック処理におけるバインド情報の消去、(3)成功時、TRO (Tail Recursion Optimization)³⁷⁾ 処理時にもなうバインド情報 (Local variable に対応する) の消去などにおいて、連想

行う引数処理部の二つの処理部のそれぞれに連想メモリが必要である。述語処理部と引数処理部の連想メモリに必要な機能は異なるが、開発効率、互換性を考慮して、4-kbit 連想メモリ LSI を中心とした同一のハードウェア構成とした。述語処理部と引数処理部のそれぞれに必要な機能をマイクロプログラム制御で実現した⁴⁴⁾。

各処理部のハードウェア構成と命令フォーマットを図-10 に示す。図-10 に示すように、本試作ハードウェアは、連想処理部 (Associative Part) と逐次処理部 (Sequential Part) および制御部 (Control Part) とからなる。各部は水平型マイクロ命令で並列に動作する。連想処理部は、連想メモリのアレイと RAM, RALU, およびレジスタファイルとから構成される。RAM には連想メモリと結合されたデータを、レジスタファイルにはマスクデータやキーデータを格納する。RALU は、マスクデータやキーデータとともに、シーケンス制御のための各種フラッグを生成する。逐次処理部は連想処理以外の処理を高速に実行するため、各種応用に応じた構成とする。連想処理部と逐次処理部のインタフェースは固定しており、種々の逐次処理部が結合できる構成としている。このような構成を採ることにより、連想メモリのもつ高速処理性能を引き出し、かつ種々の情報処理に柔軟に対処できる。

試作した Prolog マシンの写真を図-11 に、その諸元を表-3 に示す。試作した Prolog マシンでは、述語処理部、引数処理部のそれぞれを 144 ビットの水平型マイクロ命令で制御する。マイクロプログラムの開発はホストプロセッサ上のマイクロアセンブラを用いて行う。ホストプロセッサ上にはマイクロプログラムとハードウェアのデバッグのためのハードウェアモニタを搭載しており、連想メモリ、RAM、レジスタなどの内容を表示、設定などが行える。これらの環境を用いて、述語処理部、引数処理部のそれぞれ約 1000 行のマイクロインタプリタを開発した。開発したマイクロインタプリタでは、Pure Prolog と cut, fail, 算術演算などの組み込み述語をサポートしている。マシンサイクル 200 ns で、インタプリタで 108KLIPS の高性能を達成している。コンパイラを仮定すると、さらに数倍の高速化が図れる。

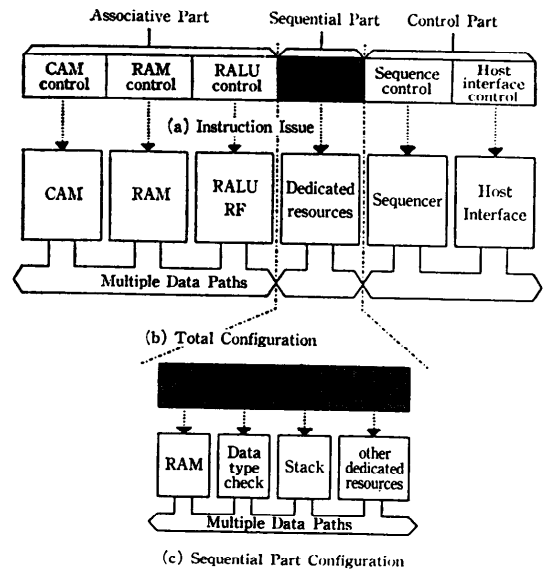


図-10 各処理部のハードウェア構成と命令フォーマット

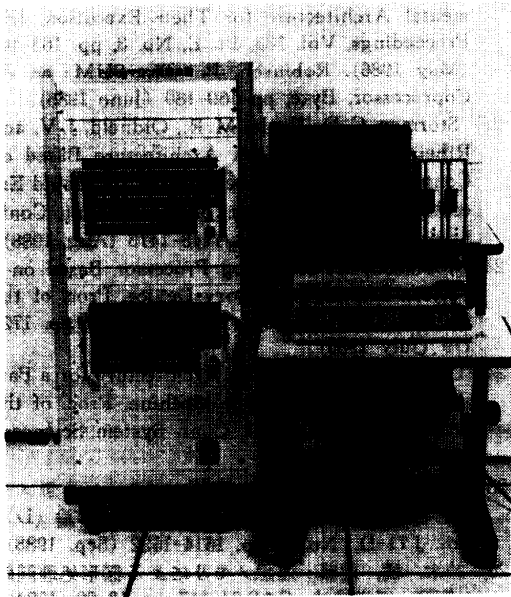
表-3 試作 Prolog マシンのハードウェア諸元

バス構成	フィールド分割・マルチウェイ 32ビット
メモリ構成	連想メモリ 32ビット×16kW (128チップ)×2
	RAM 32ビット×16kW×2 ×2
	マイクロプログラムメモリ
	144ビット×4kW ×2
	ディスクパッチメモリ 14ビット×1kW ×2
ボード構成	B4判 13枚
	連想メモリ 8枚 (16チップ/枚)×2
	そのほか 5枚 ×2
部品点数	約2,000個
マシンサイクル	200ns

4.2 試作ハードウェア上での各種応用実験

4-kbit 連想メモリ LSI を用いて試作された Prolog マシンは、各処理部単体で、汎用的な大容量連想プロセッサシステムとして使用することができる。先に述べたマイクロアセンブラやハードウェアモニタなどを用いて、連想メモリを用いたアルゴリズムの実験が容易に行える。ここでは、本マシン上で試された Prolog 以外の 3 例の応用実験を紹介する。

大久保らは、広義なユニフィケーションを高速に実行するための専用ハードウェア: Union Find Memory と、そのハードウェアアルゴリズムを提案している⁴⁵⁾。このアルゴリズムは連想メモリを用いても高速に実現できるため、試作ハードウェア上でマイクロプログラムによって実現した⁴⁶⁾。このアルゴリズムは連想メモリの並列書き込み機



(a) Frontal view



(b) CAM board

図-11 試作した Prolog Machine (ASCA)

能を活用しており、より複雑な構造体のユニフィケーションに有効である。一方、Union Find 問題を高速に実行できるため、VLSI のレイアウト処理などへの応用も期待される。

安浦らは、機能メモリ型並列プロセッサアーキテクチャを提案し、その上で論理シミュレーションを機能メモリの並列処理を用いて高速に実行できることを示した⁴⁷⁾。またそのアルゴリズムを試作ハードウェア上にマイクロプログラムで実装し、評価実験を行った。その結果、専用論理シミュレーションマシンの性能を凌ぐ高速処理が実現できる可能性を示した。また、石浦によって、このような機能メモリの並列処理を活用して、故障シミュレーションを線系時間で解く研究も行われている⁴⁸⁾。

また、筆者らは、連想メモリのもつ超並列性に着目し、局所表現型ニューラルネットワークの高速なアクセラレータを実現するハードウェアアルゴリズムを提案し、性能評価を行った。ニューラルネットワークの実現に必要な減算、絶対値、2乗、しきい値関数の演算、総和など各種演算が連想メモリのタグ検索+ワード並列ビット直列演算で実現されているため、その処理性能はニューロン数に比例して向上する。シナプス総数で 10^6 個程度で、専用チップの性能に匹敵する性能を得ている⁴⁹⁾。

5. おわりに

本稿では、機能メモリの代表として連想メモリを取り上げ、その特徴と計算機アーキテクチャにおける応用形態を整理するとともに、それらを用いた種々の計算機アーキテクチャを紹介し、その中での機能メモリ（連想メモリ）の役割を考察した。また、連想メモリを活用した計算機アーキテクチャの実際例として、4-kbit 連想メモリ LSI を用いて試作した Prolog マシンの概要を示し、その上での各種の応用実験を紹介した。

最近では、連想メモリ LSI チップ^{50), 51)} やそれを用いたボード^{52)~54)} およびその開発環境が市販されるにいたり、その実用性が高まりつつある。本稿で示したような機能メモリを用いた計算機アーキテクチャやその実験環境を容易に実現できるようになってきている。

本稿で示した機能メモリを用いた計算機アーキテクチャは、種々の情報処理システムの構築にきわめて有用であり、今後の VLSI 技術の進展とともに、より効果的なものとなる。従来のコンピュータの枠組に捕らわれないこれらの試みが、集積回路技術に基づく新たな高機能メモリ技術を喚起することを期待している。

参考文献

- 1) 田中：非ノイマン型コンピュータ [I][II], 信学誌, Vol. 68, No. 7-8, pp. 773-778, 869-874 (July, Aug. 1985).
- 2) Ogura, T., Yamada, J., Yamada, S. and Tan-no, M.: A 20 Kb CMOS Associative Memory LSI for Artificial Intelligence Machines, IEEE J. Solid-State Circuits, Vol. SC-24, No. 4, pp. 1014-1020 (Aug. 1989).
- 3) Yamagata, T., Mihara, M., Hamamoto, T., Kobayashi, T. and Yamada, M.: A 288-kbit Fully

- Parallel Content Addressable Memory Using Stacked Capacitor Cell Structure, Proc. of the Custom Integrated Circuits Conference (CICC), pp. 10.3.1-4 (May 1991).
- 4) 山田, 小倉: 機能メモリ, 信学誌, Vol. 73, No. 4, pp. 392-397 (Apr. 1990).
 - 5) 飯塚: 論理メモリ, 情報処理, Vol. 16, No. 4, pp. 275-285 (Apr. 1975).
 - 6) 奥川: 連想メモリとその応用, bit, Vol. 15, No. 4, pp. 318-329 (Apr. 1983).
 - 7) 小倉, 山田: 連想メモリ, 情報処理, Vol. 27, No. 6, pp. 593-600 (June 1986).
 - 8) Ramamoorthy, C.V., Turner, J.J. and Wah, B.W.: A Design of a Fast Cellular Associative Memory for Ordered Retrieval, IEEE Trans. Comput., Vol. 27, No. 9, pp. 800-815 (Sep. 1978).
 - 9) 市川, 坂村, 諸隈, 相磯: 連想プロセッサ ARES, 信学論 (D), Vol. J61-D, No. 10, pp. 743-750 (Oct. 1978).
 - 10) 小倉, 山田: 連想メモリ LSI の現状と今後, 信学誌, Vol. 69, No. 7, pp. 745-751 (July. 1986).
 - 11) 内田: 人工知能向き VLSI アーキテクチャ, 昭和 58 年電気四学会連大, 32-6 (1983).
 - 12) Deering, M.F.: Architecture for AI, BYTE, pp. 193-205 (Apr. 1985).
 - 13) 奥川: 32 ビット MPU のキャッシュメモリと仮想メモリ, bit, Vol. 20, No. 5, pp. 535-547 (May 1988).
 - 14) Amamiya, M., Hasagawa, R., Nakamura, O. and Mikami, H.: A List-Processing-Oriented Data Flow Machine Architecture, Proc. of the 1982, National Computer Conference, AFIPS, pp. 143-151 (1982).
 - 15) 山名, 丸島, 草野, 村岡: 並列処理システム-晴-の要素プロセッサ構成, 情報処理学会, 計算機アーキテクチャ研究会, 88-CA-69, pp. 1-8 (Mar. 1988).
 - 16) 市川, 平川: 連想処理とデータベース, 情報処理, Vol. 23, No. 8, pp. 748-756 (Aug. 1982).
 - 17) Nakano, R. and Kiyama, M.: MACH: Much Faster Associative Machine, Proc. of the 5th International Workshop on Database Machines (IWDM '87) pp. 482-495 (1987). 中野, 木山: 連想メモリを用いた関係演算処理方式, 情報処理学会, 第 34 回全国大会論文集, 3C-1, pp. 459-460 (1987).
 - 18) Bonar J.G. and Levitan, S.P.: Real-Time LISP Using Content Addressable Memory, Proc. of the 10th Int. Conf. on Parallel Processing, pp. 112-117 (1981).
 - 19) 井田: Common Loops のためのあるメッセージ送信機構, 情報処理学会, 記号処理研究会, 39-6 (Nov. 1986).
 - 20) Naganuma, J., Ogura, T., Yamada, S. and Kimura, T.: High-Speed CAM Based Architecture for a Prolog Machine (ASCA), IEEE Trans. Comput., Vol. 37, No. 11, pp. 1375-1383 (Nov. 1988).
 - 21) Oldfield, J.V.: Logic Programs and an Experimental Architecture for Their Execution, IEE Proceedings, Vol. 133, Pt. E, No. 3, pp. 163-167 (May 1986). Robinson, P.: The SUM: an AI Coprocessor, Byte, pp. 169-180 (June 1985).
 - 22) Stormon, C.D., Brule, M.R., Oldfield, J.V. and Ribeiro, J.C.D.F.: An Architecture Based on Content-Addressable Memory for the Rapid Execution of Prolog, Proc. of the 5th Int. Conf. Logic Programming, pp. 1448-1473 (Aug. 1988).
 - 23) Robinson, I.: A Prolog Processor Based on a Pattern Matching Memory Device, Proc. of the 3rd Int. Conf. on Logic Programming, pp. 172-179 (July 1986).
 - 24) Chu, Y. and Itano, K.: Architecture for a Parallel Associative Prolog Machine, Proc. of the 19th Annu. Hawaii Int. Conf. System Sciences, pp. 9-23 (Jan. 1986).
 - 25) 安田, 五十嵐, 阿刀田, 斎藤: パターン連想メモリとその論理型言語処理系への応用, 信学論 (D), Vol. J71-D, No. 9, pp. 1614-1622 (Sep. 1988).
 - 26) 鈴木, 橋, 佐藤: 連想メモリによる図形処理問題の解法, 信学技法, CAS 84-117, pp. 13-20 (1984).
 - 27) Jones, S.R., Jalowiecki, I.P., Hedge, S.J. and Lea, R.M.: 9-kbit Associative Memory for High-Speed Parallel Processing Applications, IEEE J. Solid-State Circuits, Vol. SC-23, No. 2, pp. 543-548 (Apr. 1988).
 - 28) Storer, P.R. and Duller, A.W.G.: An Associative Processor Array as a Part of a Heterogeneous Vision Architecture, Proc. of the 14th Annu. Hawaii Int. Conf. on System Sciences, Vol. 1, pp. 260-268 (Jan. 1991).
 - 29) Higuchi, T., Furuya, T., Kusumoto, H., Handa, K. and Kokubu, A.: The Prototype of a Semantic Network Machine IXM, Proc. of the 1989 Int. Conf. on Parallel Processing, Vol. 1, pp. 217-224 (1989). Higuchi, T., Furuya, T., Kusumoto, H., Handa, K., Takahashi, N., Nishiyama, H. and Kokubu, A.: IXM 2: A Parallel Associative Processor, Proc. of the 18th Int. Symp. Comput. Architecture, pp. 22-31 (1991).
 - 30) Estrin, G. and Fuller, R.H.: Some Applications for Content Addressable Memory, Proc. of the FJCC, pp. 495-508 (1963).
 - 31) McKeever, B.T.: Associative Memory Structure, Proc. of the FJCC, pp. 371-388 (1965).
 - 32) 小倉, 山田, 長沼: 大容量連想メモリ LSI の構成とその応用手法, 信学技法, CAS 84-192, pp. 17-24 (1985).
 - 33) Dennis, J.B. and Misunas, D.P.: A Preliminary Architecture for a Basic Data Flow Processor, Proc. of the 2nd Int'l Conf. Comput. Architecture, pp. 126-132 (Jan. 1974).
 - 34) 島田: データフローマシン, 情報処理, Vol. 28, No. 1, pp. 85-93 (Jan. 1987).
 - 35) Ogura, T., Yamada, S. and Nikaido, T.: A 4-kbit Associative Memory LSI, IEEE J. Solid-State Circuits, Vol. SC-20, No. 6, pp. 1277-1282 (Dec. 1985).

- 36) 長沼, 小倉, 山田, 木村: 連想メモリを用いた Prolog マシン (ASCA), 信学技法, CPSY 86-52, pp. 1-12 (1987).
- 37) Warren, D. H. D.: An Abstract Prolog Instruction Set, Tech. Note 309, Artif. Intell. Center, SRI International (Oct. 1983).
- 38) 長沼, 小倉: 超 OR 並列推論のための基本アーキテクチャと負荷分散アルゴリズム, 信学技法, CPSY 91-45, pp. 23-30 (1991).
- 39) Naganuma, J. and Ogura, T.: An Associative Processor for Logic Programming Languages, Proc. of the 14th Annu. Hawaii Int. Conf. on System Sciences, Vol. 1, pp. 229-236 (Jan. 1991).
- 40) Wade, J. P. and Sodini, C. G.: A Ternary Content Addressable Search Engine, IEEE J. Solid-State Circuits, Vol. SC-24, No. 4, pp. 1003-1013 (Aug. 1989).
- 41) Yamada, H., Murata, Y., Ikeda, R., Motohashi, K. and Takahashi, K.: Realtime String Search Engine LSI for 800-Mbit/sec LANs, Proc. of the Custom Integrated Circuits Conference (CICC), pp. 21.6.1-4 (1988).
- 42) Motomura, M., Toyoura, J., Hirata, K., Ooka, H., Yamada, H. and Enomoto, T.: A 1.2 M-Transistor 33 MHz 20-bit Dictionary Search Processor for a Machine Translation System, 1990 ISSCC Dig. Tech. Paper (Feb. 1990).
- 43) Okabayashi, I., Kotani, H. and Kadota, H.: A Proposed Structure of 4 Mbit Content-Addressable and Sorting Memory, Proc. of the VLSI '90 Symp. (1990).
- 44) 長沼, 小倉: 連想メモリを用いた Prolog マシンの実現とその評価, 信学論 (D-I), Vol. J 73-D-I, No. 11, pp. 856-863 (Nov. 1990).
- 45) 大久保, 安浦, 高木, 矢島: UNION-FIND メモリを利用した単一化操作について, 信学技法, PRL 85-48, pp. 31-40 (Dec. 1985).
- 46) 大久保, 長沼: 連想メモリを用いた単一化の ASCA 上での実現と評価, 情報処理学会, 第 33 回全国大会論文集, 4B-6 (Oct. 1986).
- 47) 安浦, 渡辺, 左達, 田丸: 機能メモリ型並列プロセッサ FMPP 上での論理シミュレーション, 信学技法, CPSY 90-94, pp. 41-48 (Jan. 1991).
- 48) 石浦, 矢島: 連想記憶を用いた線形時間故障シミュレーション, Proc. of the 4th KARUIZAWA Workshop on Circuits and Systems, pp. 63-68 (Apr. 1991).
- 49) 小倉, 長沼: 連想プロセッサ上での局所表現型ニューラルネットワークのハードウェアアルゴリズムとその評価, 信学技法, CPSY 91-44, pp. 15-22 (1991).
- 50) AMD 社 Am 99 C 10 データシート (Nov. 1988).
- 51) CRC 32256 and Coherent Processor Product Announcement, Coherent Research, Inc. (1990).
- 52) Wilnai, D.: The SM 4 K-GPX Board—A Core of a CAM Development System, Proc. of the Electro, pp. 612-615 (May 1990).
- 53) Amitai, Z.: Address Filtering FDDI LAN Bridges (The CAM Solution), Proc. of the Elector pp. 616-620 (May 1990).
- 54) Stormon, C. D.: The Coherent Processor—An Associative Processor for AI and Databases, Proc. of the Electro, pp. 621-625 (May 1990).
(平成 3 年 7 月 22 日受付)



長沼 次郎 (正会員)

昭和 31 年生。昭和 56 年徳島大学工学部電気卒業。同年電電公社武蔵野電気通信研究所入社。以来、連想メモリを用いた Prolog マシンとその並列化および論理型言語の応用手法の研究に従事。現在、NTT LSI 研究所設計システム研究部主任研究員。電子情報通信学会会員。



小倉 武 (正会員)

昭和 28 年生。昭和 51 年大阪大学工学部電子卒業。昭和 53 年同大学院修士課程修了。同年電電公社武蔵野電気通信研究所入社。以来、連想メモリ LSI およびそれを応用した情報処理装置の研究に従事。現在、NTT LSI 研究所設計システム研究部主幹研究員。工学博士。IEEE, 電子情報通信学会各会員。