

LISP マシンの試作†

—アーキテクチャと LISP 言語の仕様—

瀧 和 男** 金 田 悠紀夫*** 前 川 禎 男***

本論文は試作した LISP マシンについてアーキテクチャに重点を置いて論じている。システムはミニコンピュータを入出力処理とバックアップに用いた計算機複合体であり、LISP プログラムの高速処理を行う LISP プロセッサモジュール、主記憶装置であるメモリモジュールがミニコンピュータのバスラインを介して接続されている。ミニコンピュータはシステム初期化、入出力処理の一部、システムのモニタリングを担当している。LISP プログラムの実行はミニコンピュータが行う入出力処理の部分を除きすべて LISP プロセッサモジュールにより行われる。

LISP プロセッサモジュールは4個のビットスライス形マイクロプロセッサを結合した演算部と、マイクロプログラム制御を行うシーケンサおよびマイクロプログラムメモリから構成される CCU を中核としており、外付けの高速ハードウェアスタック、マッピングメモリ、フィールド/ビット処理回路が付加されている。インタプリタはミニコンピュータ側の処理プログラムを除きすべてマイクロコード化されており、約1,400ステップとなっている。性能測定のため、当学会記号処理研究会が行った第2回 LISP コンテストの課題プログラムを実行し測定を行った結果、インタプリタによる実行時間ではコンテストに参加したいずれの LISP 処理系よりも高速であるという結果が得られた。

1. ま え が き

人工知能の研究等において広く用いられている LISP 言語には、この言語特有の記憶領域の動的割付、関数の再帰呼出、リスト処理など汎用計算機のアーキテクチャにとり負担の大きい処理が多く高速処理のための斬新なアーキテクチャの実現が強く期待されるようになった。今回開発した LISP マシンは、市販の LSI を用い、短期間にシステムを完成させることを念頭において以下の特徴を持たせた。

- (1) ビットスライスプロセッサエレメントと大容量 IC メモリチップの使用
- (2) マイクロプログラムシーケンサと書きかえ可能なマイクロプログラムメモリの実装によるマイクロプログラム制御の導入
- (3) 外付けの高速高機能ハードウェアスタック、マッピングメモリ、フィールド/ビット処理回路等の採用
- (4) 水平形マイクロプログラム命令を用いてインタプリタのマイクロプログラム化を行い高速化を図る。

2. LISP マシンシステムの構成

開発した LISP マシンシステムの構成図を図1に示す。システムの中核となる LISP プロセッサはプロセッサモジュールとメモリモジュール (32 ビット 64 k 語) から構成されミニコンピュータ (DEC 社 LSI-11) のバスに接続されている。ミニコンピュータはページモードでメモリモジュールにアクセスできるとともに独自の主記憶を持ち LISP プロセッサと並行して動作できる。仕事は2つあり一つは始動時や保守時にマスター CPU として、LISP プロセッサのプログラムロードや初期化を行う。もう一つは実行時に、LISP プロセッサからの割込を受け、入出力プログラムの一部を担当する。そのほか時間監視や、外部からの緊急要求の受け付けも行う。

プロセッサモジュールは、処理幅が 16 ビット、メモリモジュールとのデータ受け渡しは 32 ビット幅で行われる。すなわち主記憶1語に car 部と cdr 部 16 ビットずつを持ち同時に読み書きを行う。また書込はバイト単位でも可能である。

3. LISP プロセッサのハードウェア

LISP プログラムの高速処理に必要なとされるハードウェア機能としては、

- (1) 高速かつ強力なスタック操作機能
- (2) フィールド処理およびビット処理機能

† The Experimental LISP Machine—Its Architecture and the LISP Language Specification—by KAZUO TAKI (Omika Works, Hitachi, Ltd.), YUKIO KANEDA and SADA O MAEKAWA (Department of Systems Engineering, Kobe University).

** (株)日立製作所大みか工場
*** 神戸大学工学部システム工学科

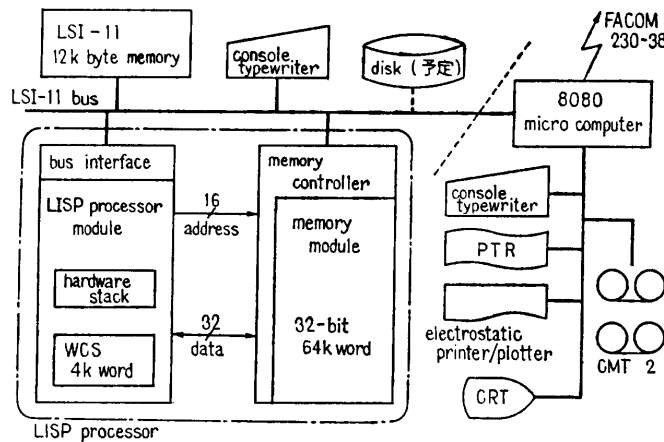


図 1 LISP マシンシステムのハードウェア構成
Fig. 1 The system configuration of the LISP machine.

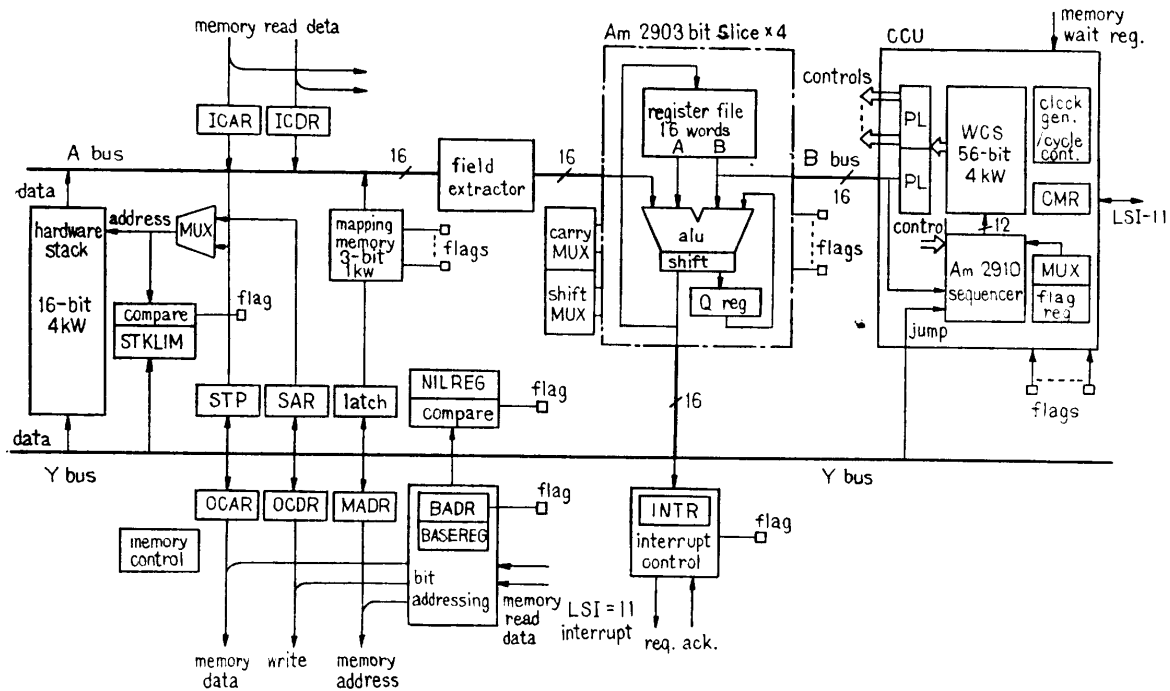


図 2 プロセッサモジュールのハードウェア構成
Fig. 2 The hardware structure of the LISP processor module.

- (3) 多様な条件ジャンプ機能
 - (4) 高速なメモリアクセス機能
 - (5) パイプライン処理や並行処理機能
- などを上げることができる。

本 LISP マシンは上述の5つの機能を念頭に置いて、市販のビットスライスプロセッサエレメント (ALU) とマイクロプログラムシーケンサ LSI を用いて設計している。LISP プロセッサ部のハードウェア構成を図 2 に示す。高速かつ高機能のハードウェアス

タック、フィールド抽出回路、ビットアドレッシング回路、マイクロプログラムレベルでのマルチウェイジャンプ、間接ジャンプ機能、豊富な条件テスト回路、メモリ番地を3ビットコードに変換するマッピングメモリ回路が ALU のバスに接続されており、マイクロプログラム用の4k語の WCS とシーケンサから構成されるコンピュータコントロールユニット (CCU) の制御下に置かれている。

スタックは4k語 (1語16ビット) の固定長であ

り、内部バス構成は A バス、B バス、Y バスの各 16 ビットの 3 バス構成である。

A バスがデータのソースバスであり、Y バスがデスティネーションバス、また B バスにはマイクロプログラム中の定数が与えられる。ALU での演算は、

- (i) ALU レジスタどうし
- (ii) ALU レジスタと B バスデータ
- (iii) ALU レジスタと A バスデータ
- (iv) A バスデータと B バスデータ

との間でそれぞれ可能である。またマイクロプログラムシーケンサは Y バスとも結ばれていて、演算結果の値が示す番地へジャンプすることができる。このことによりマルチウェイジャンプやスタックに保存しておいた番地へのリターンが許されマイクロプログラムレベルでの再帰呼出しを可能にしている。また ALU での演算結果や、ビットアドレッシング回路からの信号はフラグレジスタを介してマイクロプログラムシーケンサに結ばれ、条件ジャンプに利用される。

3.1 ビットスライスプロセッサエレメント

ALU は Advanced Micro Devices 社の 4 ビットスライスである Am 2903 を 4 個使用している。その特徴は、(i) チップ上に 16 語のレジスタファイルを持ち、任意の 2 つをデータソースとして独立に指定できる。(ii) ALU に対するソースデータの 2 つをともに外部から与えることができる (A バスと B バス) 等である。(ii) の機能は ALU をレジスタと切離して利用するため便利であり、A バスソースとマイクロプログラム上の定数との間の演算に利用している。最小サイクルタイムは 100 nsec 程度である。zero, negative, carry, overflow のフラグ用の出力端子を備えており条件ジャンプのテストに用いている。

3.2 コンピュータコントロールユニット

マイクロプログラム制御部 CCU は Am 2910 マイクロプログラムシーケンサ、WCS、PL (パイプライン) レジスタ、CMR、フラグレジスタ等から成り立っている。1 レベルのパイプライン制御を行っており、ALU の実行中に次のマイクロ命令の読出しを並行して行う。

Am 2910 は 4,096 語の WCS に対するアドレス付けが可能で、 μ -PC (マイクロプログラムカウンタ) の他にサブルーチン用の 5 段のスタック、ループカウンタを持っている。

WCS は 1 語 56 ビットを 4,096 語実装しており素子はアクセスタイム 150 nsec の 4 k ビット MOS メ

モリである。

CMR は LSI-11 から読書きできる 8 ビットのレジスタである。LISP プロセッサの実行、停止、初期化の制御に用いられ、また attention 0,1 の各ビットはフラグレジスタに結ばれていて、LSI-11 から LISP プロセッサに対して何らかの要求があることを示す。

3.3 ハードウェアスタック

70 nsec の高速メモリによる 4 k 語の固定長スタックである。スタックへのアクセスは 2 つのスタックポインタレジスタ STP と SAR により、SAR は主にスタック内部へのアクセスに用いる。いずれもカウンタタイプのレジスタで、スタックをソースとしたときの自動減少、デスティネーションとしたときの自動増加のモードを有する。これらにより ALU と STP、SAR 間のデータ転送の回数を減らすことができる。STP、SAR を用いることにより、スタックからのデータの読出し、書込みと ALU での演算処理が 1 マイクロサイクルで可能である。またスタックからスタックへのデータ転送を 1 マイクロサイクルで完了できる。LISP インタプリータでは関数の呼出しやリターンのとき、また引数の移動の際などスタックに対するデータ転送は多く、これらの機能の活用によりマイクロプログラムのステップ数を減少させることができる。

スタックリミットレジスタ (STKLIM) にはあらかじめ値をセットしておき、この値を越えてスタックが伸びたときはフラグがセットされて、スタックオーバーフローが簡単に調べられる。

3.4 フィールド抽出回路とマッピングメモリ

フィールド抽出回路は A バスと ALU との間に入りデータのマスクングとシフトを同時に行う。データの素通りを合せて 4 種類の固定パターンを有する。

マッピングメモリは 3 ビット \times 1 k 語の書きかえ可能なメモリで、主記憶アドレスを入力として 3 ビットのコードを出力する。すなわち主記憶を 64 語ごとに 1,024 区画に分割し、各区画に利用種別を表わす 3 ビットのタグを付けたと考えることができる。図 3 が LISP マシンのメモリ割当であるが、マッピングメモリによりアドレスから利用種別のコードがただちに得られる。このコードやフィールド抽出回路の出力を定数 (アドレス) と加算することによりインデックスジャンプのような形のマルチウェイジャンプが実現できる。

マッピングメモリの出力は A バスの他にデコーダを

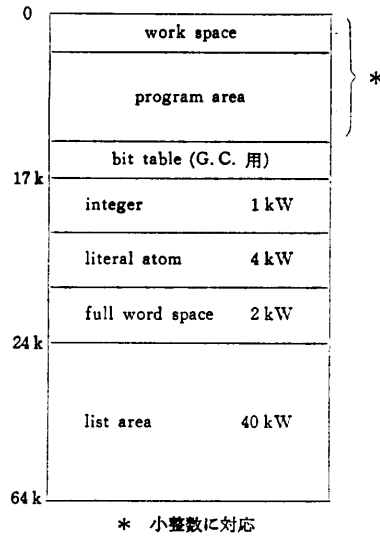


図 3 メモリ割当

Fig. 3 The memory allocation.

ととしてフラグレジスタに接続されておりリスト、文字アトム、数値などの判定が簡単に行える。

3.5 メモリ動作とその他のレジスタ

メモリとのデータの受け渡しは ICAR (Input Car Register), ICDR (Input Cdr Register), OCAR (Output Car Register), OCDR (Output Cdr Register) を通して行う。MADR (Memory Address Register) にアドレスを書込むとメモリ動作は自動的に始まる。アクセスタイム 375 nsec, サイクルタイムは 675 nsec である。メモリ動作が進行している間の 1 サイクルはメモリに関係のあるレジスタを除いて普通の ALU オペレーションが可能である。メモリオペレーションは read と read while write の 2 種で, read while write はメモリセルの内容の読出しと同時に別のデータの書込が行われるモードである。

また LSI-11 に割込を発生させるためのレジスタとして INTR がある。INTR にデータを書込むと LSI-11 に対して割込みが発生しそのデータが割込ベクタアドレスとして送られる。受け付けられるとフラグレジスタの iack フラグがセットされる。

3.6 ビットアドレッシング回路

主記憶上のビットテーブルに対してビットテストとビットセットを行う。ガーベージコレクション時にスタックを用いたマーキングを行うので 1 語に対して 1 ビットのマークビットを用意すると 64 k ビットすなわち 2 k 語のビットテーブルとなる。カーベージコレクションルーチンではテーブルをすべてクリアしたあと BASEREG にテーブルの先頭アドレスを書込む。

以後マーキングアルゴリズムに移り, 例えば $8,000_{(16)}$ 番地のリスト要素に対してマークの有無が知りたければ BADR に TEST 指定で $8,000_{(16)}$ を書込む。するとメモリ参照が起り, 2 つ後のマイクロ命令サイクルでフラグレジスタにマークの有無が現れる。ビットセットの場合にも同様に SET 指定で BADR に書込めば対応ビットがセットされる。

3.7 マイクロ命令と 4 つのジャンプ命令

ALU, マイクロプログラムシーケンサ, スタック, 外付けのレジスタ類を制御するために 56 ビットのマイクロコードを使用する。マイクロ命令は 2 つのタイプに分かれ, タイプ 1 は ALU に対して 16 ビットの定数を与えて定数演算できるタイプ, タイプ 2 はフラグをテストして条件ジャンプのできるタイプである。ALU の動作についてはいずれのタイプも同様に指定できる。

このようにマイクロ命令を 56 ビットと長くとったため, 本システムでは 4 種類のジャンプ命令を用意することが可能となった。

- (i) 無条件ジャンプ命令: タイプ 2 の命令で, 他 ALU 演算と並行オペレーションが可能である。
- (ii) 条件ジャンプ命令: タイプ 2 の命令で条件フラグで示された条件が満たされたときジャンプする。他は (i) と同様。
- (iii) インデックスジャンプ命令: タイプ 1 の命令で, マイクロ命令中の定数にマッピングメモリまたはフィールド抽出回路からの出力データを ALU で加算した番地へジャンプする命令。この命令を用い出力データによって異なる飛び先番地にジャンプするマルチウェイジャンプを実現している。
- (iv) 間接ジャンプ命令: タイプ 1 の命令で ALU での演算結果の番地へジャンプする命令。

これらの命令群はインタプリタの高速化に大きく寄与している。マイクロ命令の実行サイクルは基本的に 300 nsec である。

4. LISP 言語の仕様

本処理系で現在用意しているデータ型としては, 小整数, 整数, 文字アトム, リスト要素の 4 種類がある。これらの表現を図 4 で示す。

文字アトムをリストの形で表現せずに連続した領域に情報を置くことにより, 属性のチェック, 属性値の

データ型	語数	表	現
小整数	0	ポインタの値そのもの, $-8192 \leq n \leq 8191$	
整数	1	31 30	0
		sign	(2の補数表現で整数エリアに格納)
文字アトム	4	31	16 15 12 11 0
		関数本体	属性 属性別飛び先
		flag	引数個数 top-level-value
		property-list	print-name
		未使用	未使用
リスト要素	1	31	16 15 0
		cdr	car

図 4 データ型とその表現

Fig. 4 Data types and their representations on the main memory.

取出しを容易にした。また変数の値は old-value をスタック上の退避する shallow-binding を採用している。自由変数の取出しについては速度の向上が期待できる。主記憶上の領域をデータ型によって分割し、マッピングメモリ回路により高速に型判別を行っている。

関数型は、EXPR, FEXPR とシステム関数用としてマイクロコードで記述された MSUBR, MRSUBR, MRFSUBR を用意した。MSUBR だけは再帰呼出しを許さない関数型である。

MSUBR: CAR, CDR, CONS, EQ, NULL, ATOM, ADD 1, etc.

MRSUBR: EVAL, APPLY, EVLIS, EQUAL, MAPCON, etc.

MRFSUBR: COND, PROG, AND, OR, LIST, DE, SETQ, etc.

5. ハードウェア、ソフトウェアの製作

本システムのハードウェアサイズはおおよそつぎのとおりである。LISP プロセッサ部分は 17 cm×22 cm の万能基板 15 枚を用いた構成で電源を含め 3 段の 19 インチラックに収納している。内分けは、プロセッサモジュールが 7 枚、メモリモジュールが 8 枚で、はんだ付けとワイヤラッピングにより配線を行っている。総 IC 個数は約 600 個である。

LSI-11 側ではタイマ割込回路、8080 マイクロコンピュータインタフェース回路、バス拡張回路を約 70 個の IC を用いて 3 枚の基板上に構成し、LSI-11 の筐体内に収納している。

ソフトウェアとしては LISP プロセッサ上で働くマ

イクロコード、LSI-11 側のモニタプログラム、マイクロアセンブラが上げられる。それぞれのサイズはおおよそ次のようになる。

- (1) マイクロコード (組込関数 55 種実装時)
 - (i) 総ステップ数: 1,369 ステップ
 - (ii) インタプリータ: 539 ステップ
 - (iii) READ, PRINT: 183 ステップ
 - (iv) ガベージコレクション: 118 ステップ
 - (v) システム関数群: 212 ステップ
- (2) LSI-11 モニタ: 3.3k 語 (16 ビット語)
- (3) マイクロアセンブラ: 1,300 ステートメント (FORTRAN)

となっている。マイクロコードの部分がコンパクトな形で実現されているのが特徴である。

6. 結 論

本システムは計画の立案、基本設計に約 1 年 4 カ月要しており、ハードウェア、ソフトウェアの基本設計から詳細設計、製作におよそ 4 名の人員で当り約 7 カ月で完成させた。昭和 54 年 2 月現在システムは安定した状態で稼動しており種々の測定を行っている。

一例として本学会記号処理研究会が主催した第 2 回 LISP コンテストで課題とされたプログラムを実行し他のコンテストに参加したシステムとの比較を行った。インタプリータとしては本 LISP マシンはコンテストに参加したいずれの処理系よりも実行時間は短かく、超大型機上で実現されている LISP 処理系よりも高速処理が実現されていることが判明した。

この結果は当初の予想を越えるものであり、興味深い。高速性が実現できた理由としては、

- (1) LISP インタプリータ向きの水平形で強力なマイクロ命令を用意し、各種のパイプライン制御、並行制御を積極的に取入れた。
- (2) 高速のスタックを外付けにし、高機能のスタックオペレーションを準備した。
- (3) 特殊なハードウェアの外付け回路を準備した。
- (4) 実装上可能なかぎり高速の IC を使い、しかもボトルネックが発生しないようバランスのとれたシステム設計につとめた。
- (5) インタプリータの完全マイクロコード化と高速化を図った。

などを上げることができる。

なおインタプリータ構造および詳細なシステム分析

と評価は別の機会にゆずることにする。

謝辞 本 LISP マシンの開発に当り、有益な御助言と、多くの参考資料を提供していただいた電子技術総合研究所電子計算機部の島田俊夫氏、山口喜教氏に深く感謝いたします。また本研究に参加し献身的な努力をばらしてくれた小林康博、滝本博道、多田光弘の各君に深謝します。

参 考 文 献

- 1) 瀧, 金田, 前川: LISP マシンの試作, 情報処理学会研究会資料, 計算機アーキテクチャ 32-3 (Sep. 1978).
- 2) 山口, 島田: 仮想計算機による LISP プログラムの動的特性, 電子通信学会論文誌 D, J61-D, 8 (1978).
- 3) 黒川利明: LISP のデータ表現, 情報処理, Vol. 17, No. 2 (1976).
- 4) 中西正和: LISP 入門, 近代科学社 (1977).
- 5) 竹内郁雄: LISP 処理系コンテストの結果, 情報処理学会研究会資料, 記号処理 5-3 (1978).
- 6) 島田, 山口, 坂村: LISP マシンとその評価, 電子通信学会論文誌 D, J59-D, 6 (1976).
- 7) Teitelman, W.: INTERLISP Reference Manual, Xerox (Feb. 1974).
- 8) LISP 1.9 User's Manual, EPICS-5-ON-2, ETL.
- 9) Greenblatt, R.: The LISP Machine, MIT AI Lab. Working Paper 79 (Nov. 1974).
- 10) Knight Tom: CONS, MIT AI Lab. Working Paper 80 (Nov. 1974).
- 11) Bawden, Aian et al.: LISP Machine Progress Report, MIT AI Lab. Memo No. 444 (Aug. 1977).
- 12) The Am 2900 Family Data Book, Advanced Micro Devices, Inc.

(昭和 54 年 5 月 21 日受付)

(昭和 54 年 6 月 21 日採録)