

高速 LISP マシンとリスト処理プロセッサ EVAL II†

—アーキテクチャとハードウェア構成—

前川 博俊^{††} 齋藤 年史^{††} 土井 俊雄^{††}
西川 岳^{††} 安井 裕^{††}

本論文は、高速処理に重点をおいて試作した LISP マシンのアーキテクチャとハードウェア構成を中心に論じている。本マシンのおもな構成要素は、リスト構造のデータを処理するプロセッサ EVAL II、LISP 入出力関数の実行およびファイル等の外部記憶装置の管理を行う I/O プロセッサ、そして、リストセルを格納するメインメモリである。とくに EVAL II は、リスト処理向きアーキテクチャを備えることを目的として設計されたプロセッサであり、その実現への柔軟な対処のため、既製のビットスライス等のマイクロプロセッサはいっさい使用しなかった。EVAL II は、その扱うデータがおもにアドレスポインタであることから内部バスにアドレス/データ共通バスを備えている。EVAL II のリスト処理向き機能として、リストサーチを ALU 演算や分岐演算と並列に高速処理する CARCDR 演算機能や、リストセルや関数の型による処理先への分岐を効率よく実行するディスパッチ機能がある。また、分岐演算と他の演算の並列処理やパイプライン処理などを行っている。ベンチマークテストにより、LISP インタプリタでは、本マシンは超大型計算機上の LISP システムに匹敵する高速処理性能を示している。また、動特性の測定結果から本アーキテクチャが LISP マシンに適合することが確認されている。

1. はじめに

人工知能、知識工学等の研究に用いられてきたリスト処理言語 LISP¹⁾は、計算機言語の一つの基幹となってきた。LISP の応用への意欲の増大とともに、汎用大型計算機をはじめ各種計算機上で稼動する高速の処理系が相次いで製作され²⁾⁻⁵⁾、また、マイクロコンピュータやプロセッサ IC の普及、半導体集積回路技術の進歩により、LISP 専用マシンの開発も各所で行われている⁶⁾⁻²⁰⁾。

われわれは、LISP 処理系の速度向上への新しいアプローチとして、マルチプロセッサ構成の専用マシン—EVLIS マシン—を提案し、製作を続けている^{21), 22)}。EVLIS マシンは、インタプリタ¹⁾内の関数 `evlis` の第 1 引数の各要素を処理単位とし、この処理を複数台のプロセッサで並列に実行する。プロセッサへの処理の割付け、プロセッサ間の同期、リスト構造の書き換えを生ずる関数 (`rplaca` 等) の実行による副作用の正しい伝播など、並列化に伴う問題はすべて EVLIS マシンが処理する。したがって、LISP 1.5 の文法に従って書かれた、並列化を意識しないプログラムであって

も、マシンは自動的に並列処理を行う。

EVLIS マシンの開発においては、その構成要素である EVAL II と名づけたリスト処理専用プロセッサのアーキテクチャも、よりリスト処理向きなものとした。その結果、単体の EVAL II を使用して LISP マシンを構成した場合でも、従来の専用マシンをしのぐ性能が発揮されることが確かめられた。

本論文では、EVAL II のアーキテクチャと、これを 1 台使用して構成した LISP マシンのハードウェアの設計について論じる。

2. 設計方針

設計する LISP マシンは大学の一研究室の規模と技術で開発できる小型の実験機とするが、EVAL II ではとくにわれわれの意図するアーキテクチャを具備することに主眼を置く。EVAL II のアーキテクチャ設計上の留意点は次のとおりである。

- (1) メモリ上のデータの大部分がアドレスポインタであるため、アドレスバスとデータバスの区別をなくし、アドレスとデータの結合性を高める。
- (2) 高速処理と演算の柔軟性のため、ALU 演算は 3 アドレス方式とする。
- (3) ALU の演算ソースにマスカを設け、リストデータからのポインタ部の抽出などを容易にする。
- (4) 分岐演算が多用されるため、1 マイクロ命令

† Fast LISP Machine and List-Evaluation Processor EVAL II
—Processor Architecture and Hardware Configuration by
HIROTOSHI MAEGAWA, TOSHIFUMI SAITO, TOSHIO DOI,
TAKESHI NISHIKAWA and HIROSHI YASUI (Department of
Applied Physics, Faculty of Engineering, Osaka University).

†† 大阪大学工学部応用物理学科

内においてALU演算と分岐演算を並列に行う。

また、関数やリストセルの型による処理先への分岐を高速化するディスパッチ機能をもたせる。

- (5) リストをたぐる操作をALU演算や分岐演算と並列に行う機能(CARCDR演算機能)をもたせる。
- (6) 命令のフェッチと実行、メモリへの書き込み動作と他の演算等に対し、パイプライン処理を行う。
- (7) ハードウェア、ソフトウェアの両面にわたるデバッグ、統計データ収集のため、EVALIIの内部状態を直接アクセスできる診断機能をもたせる。

このようなアーキテクチャを実現するため、その処理の柔軟性と開発の容易性からEVALIIはマイクロプログラム制御を採用する。既製のビットスライスマイクロプロセッサファミリ等は、プログラムスタックが数ワード程度と浅く、バス構成もほとんどそのプロセッサに拘束され、また、リスト処理以外の機能にマイクロ命令の情報がとられる等の問題を有している。したがってEVALIIでは既製のマイクロプロセッサおよびその周辺LSIはまったく使用しなかった。

3. マシンの構成

マシンの全体構成を図1に示す。

EVALIIはリスト処理専用プロセッサであり、メインメモリ上のメールアドレスを経由して与えられたリストデータを評価し、値をメールアドレスに返す。

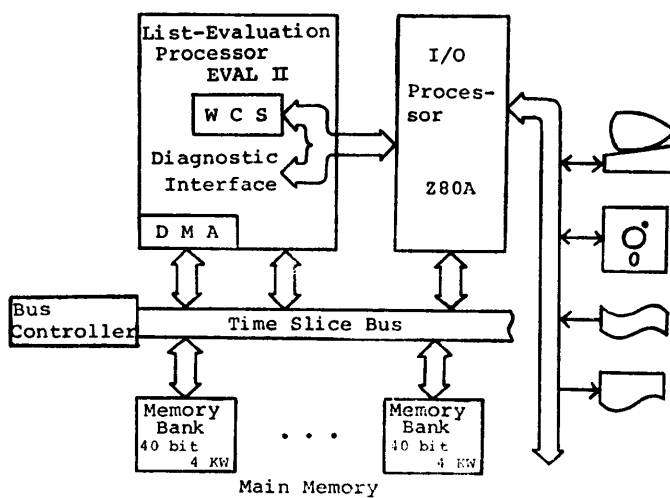


図1 LISPマシンの構成

Fig. 1 LISP machine configuration.

EVALIIの詳細は次章に述べる。

メインメモリは、1語40ビット、最大32k語の容量をもち、リードおよびライトアクセスタイムはそれぞれ350nsec、200nsecである。このメモリは4k語のバンク最大8個と、共通バス、コントローラからなる。メインメモリは2台まで実装可能で、2台のメインメモリは並列動作可能である。プロセッサ、バンク間のアクセスはメインメモリのバスを50nsecで時分割することにより多重化している。バンク間のバスの競合はバンク間の優先順位(50nsecごとに変化する)がより高いバンクへのアクセスを許可する方法で制御し、一つのバンクに対するプロセッサ間の競合はバンクごとに固定された優先順位により制御する。メインメモリはリストセル(car部、cdr部、各20ビット)の格納のほか、プロセッサ間通信用メールアドレスに使用する。

I/OプロセッサはCPUにZ80A・MPU(クロック4MHz)を使用し、64kバイトのローカルメモリをもつ。このプロセッサはマシンのマスタープロセッサとして働き、ファイルの管理、マイクロアセンブラによるプログラムの作成支援、EVALIIの起動および停止、LISP入出力関数の実行などを行う。また、EVALIIとの間に設けたバスを介して、WCSへのマイクロプログラムのロード、EVALIIの内部状態のモニタリングやデバッグの支援、統計データの収集を行う。

4. EVALIIのハードウェア

EVALIIのハードウェア構成を図2に、マイクロ命令のフォーマットを図3に示す。

4.1 内部バスとALU

内部バスは、各20ビットのAソースバス(ASB)、Bソースバス(BSB)、デスティネーションバス(DSB)からなる。

ALU演算は3アドレス方式をとり、演算として、論理積、論理和、排他的論理和、固定小数点加減算、右桁移動がある。演算の二つの入力はASBとBSBで与えられ、結果がDSBに送られる。ASBとDSBには演算対象となるレジスタやメモリをすべて接続した。BSBには使用頻度の高いと考えられる8個の汎用レジスタ(R0~R7)と5個の専用レジスタ

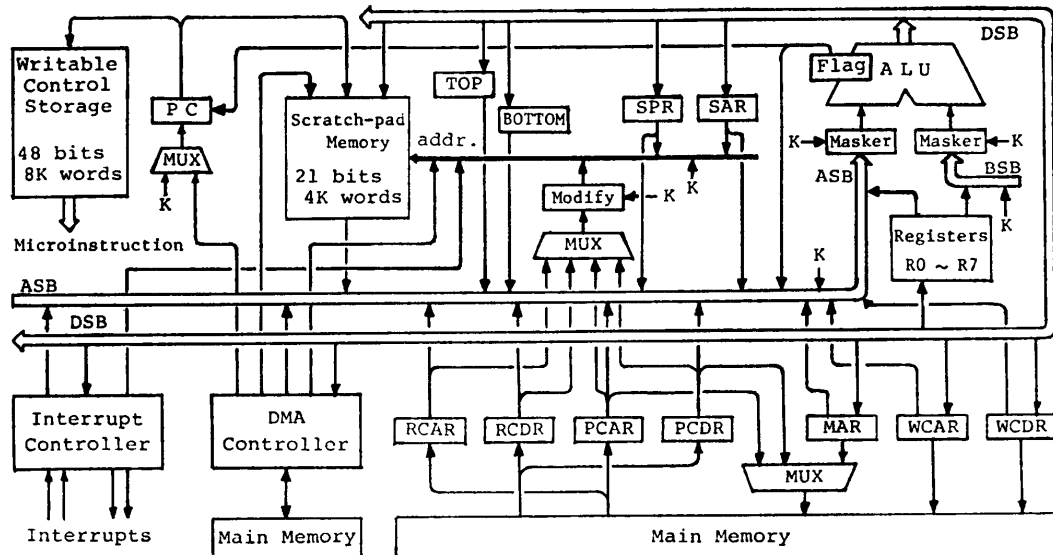
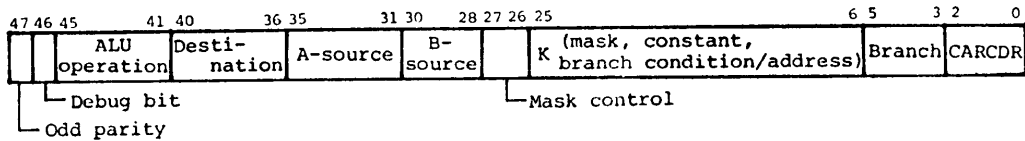


図 2 プロセッサ "EVAL II" のブロック図
Fig. 2 Processor "EVAL II" block diagram.



- Debug bit : used to set breakpoints, to measure execution frequencies, etc.
- ALU operation : specify a kind of ALU op. and flag controls
- Destination
- A-source } : specify operands of an ALU op.
- B-source
- Mask control : specify masking for A- and/or B-source
- K : specify a mask pattern, branch conditions and address, etc. (see Fig. 4)
- Branch : specify a kind of branches (see Fig. 4)
- CARCDR : specify a source and destinations of CARCDR-op.

図 3 マイクロ命令のフォーマット
Fig. 3 Microinstruction format.

タ (値が NIL, *T*, 全ビットが1のものなど) のみを接続し、回路が複雑になるのを避け、また、命令の Bソースフィールドを短縮した。

演算の入力はマイクロ命令の Kフィールドに指定した 20 ビットのパターンでマスクができる。入力データの上位 4 ビットと下位 16 ビットのマスクはリストセルのタグ部とポインタ部の抽出に頻繁に使用するため、Kフィールドの特別の 2 ビットをこれらのマスク指定にあて、Kフィールドのその他の部分を分岐先情報等に利用可能とした (4.4 節および図 4 参照)。ALU への定数入力、全ビットが 1 の専用レジスタに対する定数パターンのマスク指定により行う。

演算の結果に対し、零、負、キャリのフラグが立

ち、キャリは ALU 演算の第 3 番目の入力とすることができる。

4.2 スクラッチパッドメモリ

汎用性をもつ記憶領域として、8 個の汎用レジスタのほかに、1 語 21 ビット (うち 1 ビットが偶数パリティ)、4 k 語構成のスクラッチパッドメモリ (SM) をもち、これにはアクセスタイム 35 nsec のスタティックメモリを使用している。SM のおもな用途は、マイクロプログラムの戻り番地やデータを一時保持するスタック、多方向分岐演算命令で用いるディスパッチテーブル、割込み (他のプロセッサからの通信や、パリティエラーなどによる) 時の分岐先番地のテーブルである。SM 内の番地を指定するものとして、スタッ

クポインタレジスタ (SPR), SM 用アドレスレジスタ (SAR), マイクロ命令の K フィールド, 多方向分岐演算命令のディスパッチ番地, 割込みのベクタがある。

SPR はスタック内の最新データの格納位置を指している。SPR の値はスタックへのアクセス時に自動増減するが, この値を変化させずにスタックをアクセスするモードも設けている。スタックのオーバフロー, アンダフローは境界値レジスタ (TOP, BOTTOM) と SPR の比較により検出され, 割込みの要因となる。

SAR と K フィールドは SM 内の任意の番地へのアクセスに使用される。

SM へのデータの実際書込みと ALU 演算等はパイプライン化してあり, 書込みは次の命令サイクルの後半で行う (サイクルの前半ではパリティ生成と書込みの準備を行う)。したがって, SM への書込みを行う命令の直後に SM の読出しを行う命令を実行する場合, 両アクセス番地の比較を行い, 等しいときは迂回路を通して正しいデータを読み出している。

スタックのオーバフロー時におけるスタック内容の一部の退避や, EVLIS マシンでの EVAL II の状態の保存を効率よく行うために, SM とメインメモリの間での DMA 転送機構 (DMA コントローラは現在実装していない) を設ける。SM のアクセスは SM をサイクルスチールして行い, 2 語単位で転送する。メインメモリとの転送は, 時分割バスに接続された独

立なバスにより行う。

4.3 メインメモリアクセスと CARCDR 演算機能

メインメモリへのアクセスは, レジスタ MAR に格納された番地に対して行われる。書込みはレジスタ WCAR, WCDR の一方または両方の内容で行い, 読出しはレジスタ RCAR, RCDR, PCAR, PCDR のすべてまたは前 2 者に行う。アクセスの起動は, 演算結果を MAR, WCAR, WCDR に格納する命令で指定され, アクセス中, EVAL II は他の処理を続けることが可能である。

CARCDR 演算機能は, リストをたぐる操作を効率よく行う目的で設けた機能であり, PCAR, PCDR の内容を番地として直接メインメモリの内容を読み出す。アクセスの起動はマイクロ命令中の CARCDR フィールドの指定で行われ, 他の演算と同時並行的にメインメモリのアクセスが可能である。

4.4 マイクロ命令とシーケンス・コントロール

マイクロ命令格納用のコントロールストア (WCS) は 1 語 48 ビット (うち 1 ビットが奇数パリティ), 8 k 語の構成で, アクセスタイム 35 nsec のスタティックメモリを使用している。命令の基本サイクルタイムは 100 nsec であるが, ALU 演算が加減算のとき, および, スクラッチパッドメモリの読出しのときには, 各 50 nsec 延長される。メインメモリへのアクセスはパイプライン処理されるが, アクセス時, メインメモ

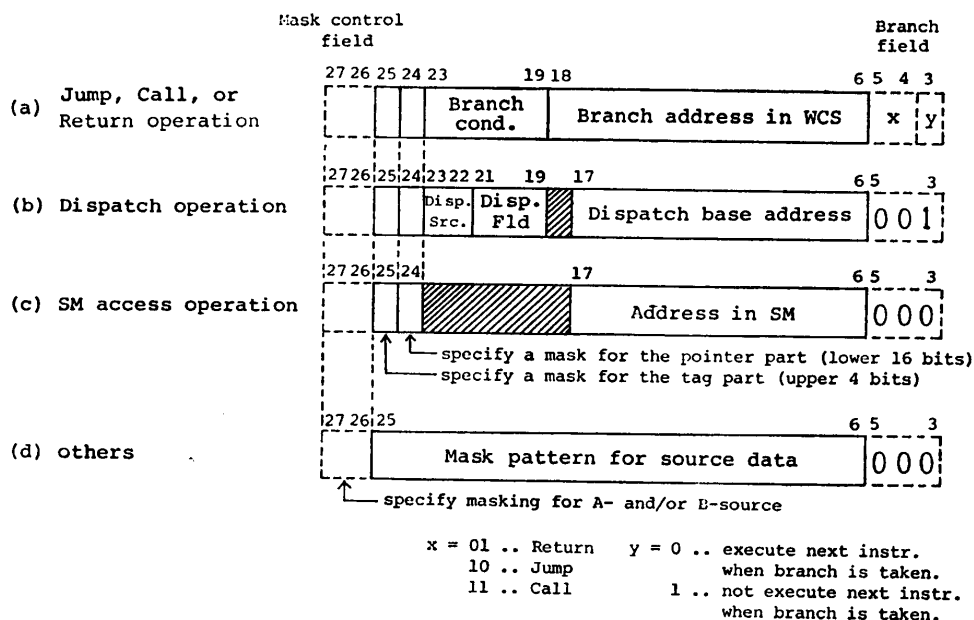


図 4 K フィールドの構成

Fig. 4 K-field construction.

りがビジーであればサイクルの延長がおこる。

コントロールの分岐はマイクロ命令の Branch フィールドと K フィールドの指定により行う。図 4 にこれらのフィールドの機能をまとめて示す。図 4 (a) の 3 種の分岐演算のときは、ALU のフラグに対する 5 条件 (正, 零, 負, キャリ有, 無) の組合せで分岐の成否を決定する。マイクロ命令のフェッチと実行をパイプライン化しているため、分岐成功時、フェッチ済みの次命令を実行するか、または、その命令の実行を抑制するかの選択を可能とした。

図 4 (b) は、16 あるいは 256 方向への多方向分岐演算 (ディスパッチ演算) の場合を示す。分岐先番地の導出は、図 5 の機構により、1 命令サイクル内で実行される。

4.5 診断用インタフェース

ハードウェア、ソフトウェアの両面にわたるデバッグ、保守、動特性等の統計データの収集を容易にするため、EVAL II の内部状態を I/O プロセッサから直接アクセス可能な診断用インタフェースを設けた。少ない結線数でこれを実現するため、EVAL II の内部にシフトレジスタを埋め込み、情報の転送を行っている (図 6)。診断用カウンタ (16 ビット) はマイクロ命令のデバッグビットを命令実行時に計数しており、オーバーフローすると EVAL II のクロックを停止させる。このカウンタとデバッグビットの組合せにより、プログラムのブレイクポイントの設定、動特性の測定が行える。

診断用インタフェースに直結していないレジスタやスクラッチパッドメモリへのアクセスは、WCS 上の

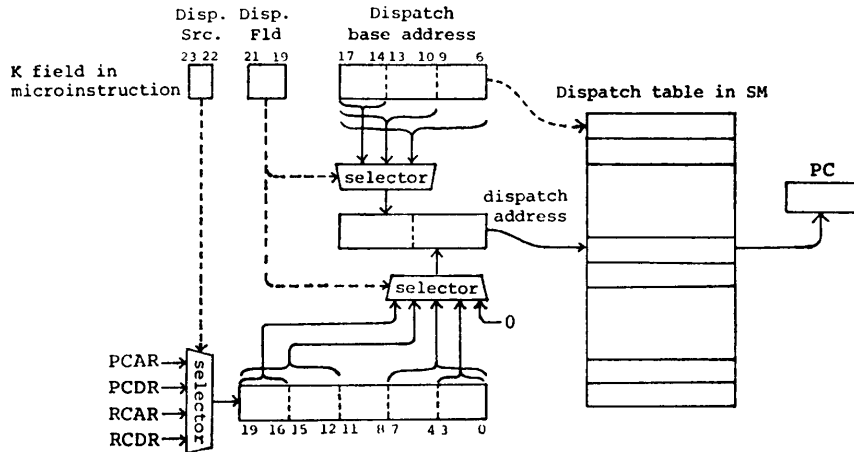


図 5 ディスパッチ機能による分岐アドレスの生成

Fig. 5 Branch address generation on dispatch facility.

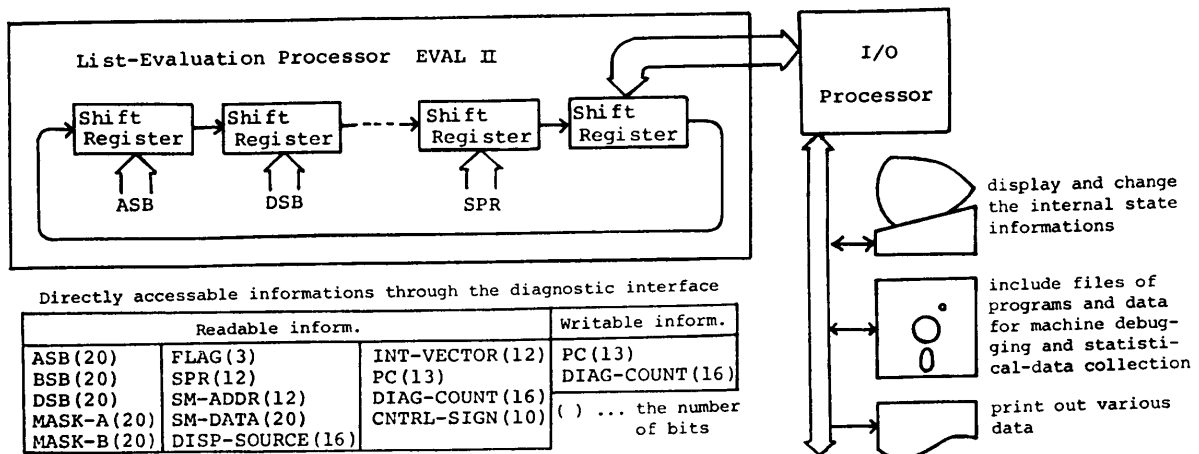


図 6 EVAL II の診断用インタフェースと内部モニタ機能

Fig. 6 EVAL II diagnostic interface and monitoring function.

命令の実行 (I/O プロセッサが起動) により行う。読出しの場合は、ASB, BSB に情報を取り出し、診断用インタフェースを通してアクセスする。書込みの場合は、マイクロ命令の K フィールドを用いてデータを入力する。

これら、直接、間接にアクセスする情報のモニタや変更指示は CRT 端末で統括して行う。これにより、ハードウェア調整の際にマシンの状態チェックが容易になり、調整時間の短縮に効果があった。また、インタプリタの作成、デバッグは、CRT 端末とプリンタのみで行うことができた。

5. おわりに

リスト処理向きのアーキテクチャをもち、高速処理を達成することを目標に、LISP マシンを開発した。このマシンでは、CARCDR 演算機能をはじめ、分岐演算と他の演算の同時実行やパイプライン処理などの並列処理化を図った。また、このアーキテクチャの実現と高速化達成のため、リスト処理専用プロセッサ EVALII のすべてを論理 IC 等の組合せで構成した。

本マシンの処理速度は、ベンチマークプログラムに対し、超大型計算機 (ACOS-1000, M 200-H) 上にインプリメントされた LISP インタプリタに匹敵しており、設計方針をほぼ満足する専用マシンとして稼動している。また、動特性の測定から、アーキテクチャが LISP マシンに適合することを確認した。これらについては別稿に述べる予定である。

今回試作したマシンのハードウェアの規模は、プロセッサ EVALII (1台) が基板 10 枚、IC 約 620 個、I/O プロセッサが基板 4 枚、IC 約 270 個、メインメモリが基板 12 枚、IC 約 690 個である。

謝辞 最後に、マシンの実装において協力いただいた大寺信行氏をはじめ研究室の諸兄に感謝する。本研究の一部は文部省科学研究費による。

参 考 文 献

- 1) McCarthy, J. et al.: *LISP 1.5 Programmer's Manual*, MIT Press, Cambridge, Mass. (1966).
- 2) 情報処理学会プログラミング・シンポジウム委員会: 記号処理シンポジウム報告集, pp. 176-219 (1974).
- 3) 竹内郁雄: 第二回 Lisp コンテスト, 情報処理, Vol. 20, No. 3, pp. 192-199 (1979).
- 4) Kurokawa, T.: *Fast Interpreter of LISP 1.9*, *J. Inf. Process.*, Vol. 2, No. 2, pp. 81-88 (1979).
- 5) 近山 隆: プログラミング・ツールとしての Lisp 処理系, 情報処理学会第 23 回全国大会論文集, pp. 225-226 (1981).
- 6) Greenblatt, R.: *The LISP Machine*, MIT AI Lab., Working Paper, 79 (1974).
- 7) Knight, T.: *CONS*, MIT AI Lab., Working Paper, 80 (1974).
- 8) 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, Vol. J-59-D, No. 6, pp. 406-413 (1976).
- 9) 後藤, 井田, 相馬: 記号処理向き計算機 FLATS の構想, 情報処理学会研究会資料, 記号処理 1-1 (1977).
- 10) Steele, G. L. Jr. and Moon, D. A.: *CADR*, MIT AI Lab., Working Paper, ??? (1978).
- 11) 薄, 田丸, 所: マルチプロセッサによる LISP マシン, 電子通信学会技報, EC 78-31 (1978).
- 12) Williams, R.: *A Multiprocessing System for the Direct Execution of LISP*, *Proc. 4th Workshop on Computer Architecture*, pp. 35-41 (1978).
- 13) 井田, 間野: マイクロプロセッサを用いた Lisp マシン ALPS/I, 情報処理学会論文誌, Vol. 20, No. 2, pp. 113-121 (1979).
- 14) 長尾, 中島, 伊藤, 川口, 三田村: LISP マシン NK 3 のアーキテクチャとその性能評価, 情報処理学会研究会資料, 記号処理 7-4 (1979).
- 15) 瀧, 金田, 前川: LISP マシンの試作, 情報処理学会論文誌, Vol. 20, No. 6, pp. 481-493 (1979).
- 16) 日比野, 渡辺, 大里: LISP マシン ELIS の基本設計, 情報処理学会研究会資料, 記号処理 12-15 (1980).
- 17) Burton, R. R. et al.: *Overview and Status of Dolado LISP*, *Conf. Record of 1980 LISP Conference, Stanford*, pp. 243-247 (1980).
- 18) Hibino, Y.: *A Practical Parallel Garbage Collection Algorithm and Its Implementation*, *Proc. of 7th Symposium on Computer Architecture*, pp. 113-120 (1980).
- 19) Sussman, G. J. et al.: *Scheme-79-Lisp on a Chip*, *IEEE Comput.*, Vol. 14, No. 7, pp. 10-21 (1981).
- 20) Guzmán, A.: *A Heterarchical Multi-Micro Processor LISP Machine*, *Proc. Conf. on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 309-317 (1981).
- 21) 安井, 斎藤, 三石, 宮崎: LISP での並列処理における動的特性と EVLIS マシンの構成, 情報処理学会研究会, 記号処理 10-4 (1979).
- 22) 前川, 土井, 西川, 斎藤, 安井: 試作 EVLIS マシンの EVALII と開発支援機能, 情報処理学会研究会, 記号処理 17-1 (1982).

(昭和 57 年 9 月 27 日受付)

(昭和 58 年 4 月 19 日採録)