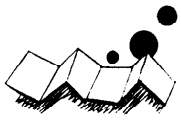
解　説

LISPマシン†



安　井　　裕††

1. まえがき

計算機利用の世界が，数値計算から情報処理という形態に拡大するにつれ，目的とする処理内容あるいはそれを記述する言語と，本来数値計算を対象として作られてきた従来からの計算機アーキテクチャとの間で非数値的な処理領域についての不整合が問題点となってきている．これらの問題の解決として，ここで述べるLISPマシンはもちろん，他の高級言語の名前，あるいは応用目的を名前として頭に戴いた専用計算機[18),63)]を製作しようとするのは当然の方向であり，すでに1960年代初めにバロース B 5000においてALGOL で成功を収めている．本稿では，記号処理システムの基盤となっているLISP言語システムに注目し，現在すでに稼動，あるいは開発中のLISPマシンについての紹介を行うとともに，幾つかのマシンをとりあげ，実例を介して特徴あるLISPマシンのアーキテクチャの解説を試みる．数多くある超大型機からマイクロプロセッサに至るマイクロプログラム可能でない単なる汎用マシン上にソフトウェアによって実現されたLISPシステムについては対象としない．また本号の編集上の主旨から，データフローマシン，リダクション・マシン等についても，ここでは積極的には取り上げないこととする．

LISP マシンは，次の世代の高性能パーソナルコンピュータの原型とも我々は期待するものであり，人工知能，記号処理というやや堅いイメージから，さらにDB，CAD，より進んだ OA 等への今後の身近な応用の世界への貢献が確信される．ここでまず，広く計算機アーキテクチャの各種アイディア，テクノロジの発達とともに歩んできたLISPマシンの開発経過を次章に追ってみたい．

† A Survey of LISP Machine by Hiroshi YASUI (Department of Applied Physics, Faculty of Engineering, Osaka University).

†† 大阪大学工学部応用物理学科

2. LISP マシンの開発経過とその背景

1960年 McCarthy によってもたらされた LISP[1)] は記号の高度で知能的な取扱いを目的とした言語として MIT の MAC プロジェクトを中心にその処理系の研究と応用が開始され，広く米国をはじめ各国でのAI 研究グループの間を中心にその道具としてのシステムの開発が盛んとなった．そして数年の間に IBM 7090を始めとして種々の計算機の上で多くの LISP の処理系[3),4)] が製作されている．これらの計算機は当時としては大型高性能のシステムであるが，主記憶も高々65K語程度，アクセスタイムも数マイクロ秒のものであり，現在から見ればマイコン，ミニコン級の計算機とみられるものである．そしてLISPの応用の意欲が成長するにつれてLISPのプログラムも大きくなり，加えて一般の計算機需要の増加とともに，その後の発達した大型計算機でもマルチプログラミングやTSS のもとでは，言語との不整合のみならず，使用可能メモリ領域の不足やメモリスワップの頻発につながり，メモリ量と処理性能の面でLISP処理系の問題点が現れてきた．すなわち，LISP はそのデータ構造とするリストの処理固有の動的な特性上，リストがメモリ領域を大量に消費し，かつリストの探索と生成，消滅にメモリを頻繁にアクセスするという量と処理時間において，ジョブの多重処理が通常となった身近なシステムでの利用環境では仕事が困難となってきた．(このことは今日でも我々大学間の計算センタにおける超大型機の上ですら，TSS 下では，その運用方式にも関係するが，レスポンスは大変遅くなり，閑散時間帯を狙って仕事をしなければならない状況にある．)そして，それ以来 LISP では，リスト処理速度の向上と大容量セルの獲得のための努力が始まる．

2.1 LISP マシンへの胎動

LISP の基本関数のなかの CAR，CDR をとりあげても，古典的に当時の IBM 7090 を例にすれば[1)] それぞれ機械語で7ステップのプログラムで実現されて

いる．もしこれらの基本関数や，さらに複雑な内容でよく使用される関数，スタック操作等の使用頻度の高い仕事が，1命令で，かつ理想的には1マシンサイクル内で実行できるならば，飛躍的な処理速度の向上が期待できる．しかしながら，1960年代では，我々が専用のアーキテクチャを提案しても自らハードウェアに手をつけることはまず不可能な時代であった．したがって，その頃いくつかあったリスト処理マシン，LISP マシンへのアプローチはいずれも，ハードを作るところまでは至っておらず，提案もしくはペーパマシンに終っている．リスト処理に対するマシン化について1963年カンサス大の Wigington[2] による研究があるが，論文では LISP 言語は直接対象になっておらず時期的にもハードウェアの製作は困難であった．しかし，B5000のアーキテクチャをスタートとしてその後連想記憶などアソシエイティブなユニットの導入[5]，スタックマシン，タグアーキテクチャやレジスタ割付等の研究[7]が続けられており，また種々の汎用機上の LISP システムの研究，使用経験等の蓄積が広く行われて LISP マシンの実現のための手段は培われていたといえる．そのような環境の中で，後述するように，意欲的に種々のアイディアを盛り込み，LISP マシンへの胎動期の典型ともいえる1971年に発表されたカーネギー・メロン大学の LISP マシン[6]がある．しかし製作には着手されず構想に終ったようである．

2.2 海外における LISP マシン

1973年ゼロックス社 PARC の Deutsch[8] がB 5000の影響を受けたアーキテクチャとして，タスクの制御と変数の束縛に1つのスタックを用い，LISP 向きの命令セットと，その命令が平均1バイトになる効率のよい命令語の LISP マシンの設計を示した．そして，当時の PDP-10 で働いていた BBN-LISP に比べて，オブジェクトコードの長さが 1/3 から 1/4 に，またコンパイラ自身も 1/3 になるという，コンパクトなオブジェクトコードを作り出す MicroLISP を示し，LISP マシン実現に影響を与えることになる．この時点ではマシンはまだでき上がっていない．しかし2年後に使用可能となる．1974年には Deutsch の LISP マシンや，その頃のカーネギー・メロン大の DEC PDP-11/40等のアーキテクチャの影響を受けた CONS[10] マシンと名付けられたアーキテクチャによる MIT の LISP マシンのプロジェクトが Greenblatt によって発表された[9]．これは，この頃から半導体集積回路技術の進歩が，専用機の製作を容易にしつつあり，またマイクロプログラム可能なマシンが，手近な道具として存在し得る世代となったことによる．彼らのシステムの狙いは

1 大きいプログラムの実行ができること．PDP-10の数メガ語に相当．

2 スタンドアロンで一人の利用者によるパーソナルなユーズ．午後の3時も，朝の3時のように使えること（いかに TSS 等の多重処理環境では使いものにならないかがよくうかがえて面白い表現である）．

3 手軽な値段で手に入れることができる．約7万ドル程度を期待（あるいはそれ以下に）．

4，5，…その他 PDP-10 の MACLISP との互換性での優位，ファイルシステム，コンパイルされたプログラムのメモリ効率の向上，コリカーシブであることや制御構造での CONNIVER 形式の踏襲，ディスプレイ指向のコンソールでの対話的利用等々である．

そして invisible ポインタや，CDR コーディングなどを含むアーキテクチャへの試行であった．この CONS と名付けられたプロセッサはさらにアドレス空間，マイクロ命令幅の増強等にみられる機能拡張が加えられ1978年 CADR マシン[23),28)]へと発展する．また一方では Deutsch によるゼロックスの LISP マシンは，ゼロックス PARC で作られた汎用のエミュレータ用マシン（マイクロプログラム可能）である Alto[33),42)] ハードウェア上で，ByteLISP[53] と名付けられた LISP について1974年から製作，1975年の中頃には徐々に利用できるようになった．さらに Inter LISP そのもののインプリメントと効率のよいインクレメンタルな自動ガベージ・コレクション（GC）が設計され，組み込まれることとなり，1977年の初めには，まず InterLISP の大きいプログラムが実行できるようになっている[62),53)]．しかし依然として実際の利用にはまだまだ低速で，とても仕事にならないことが判明する．そして同一の OP コードで処理速度，メモリ空間等の増強等を狙ってさらに高度化した新しいハードウェア Dorado[49]への開発が始まる．Dorado ができた当初は，彼らの予想に反して大変遅い状態であった．しかしさらに動作解析に役立つ統計情報の収集機能を追加し，その情報を分析考察することにより，アーキテクチャに改善を加えることに成功を見て，速度の評価は，プログラムの内容によって一概にはいえないが1980年には DEC KA-10 の5倍の速さで LISP プログラムを処理している報告ができるようになった．

これらの独自に設計されたプロセッサではなく商品化された既製のマイクロプログラム可能なマシンによるLISPマシン開発の方向がある．1977年ユタ大学のGriss等によるバロースB 1726計算機でのStandard LISPをサポートするLISPマシンである[24),26)]．これはマイクロプログラムによってつくられたMBALMと名付けられたスタックマシンで，ポーリッシュ・ストリングとしてのByteコードを実行するLISPインタプリタをベースにしている．また彼らはこのLISPスタックマシンを他の計算機上に作成するポータブルな言語BILを用いて，ブートストラッピングによりZ 80からCRAY-1に及ぶいくつかの計算機上にMBALMの改訂版MTLISPを実現させている．そしてREDUCEのサブセットをZ 80やLSI-11で使用できるようにしている[34),35)]．

1978年には構想の段階ではあるが機能分散型のマルチプロセッサによるLISPマシンともいえるイリノイ大学のMSLマシンが提案された[27)]．これはコンパイラによる目的コードの実行ではなく，LISPのDirect Execution machineを目的としている．トランスレーション，エクスキューション，メモリの3つの名前のついた3台のプロセッサが非同期で疎結合された形式で構成されたものである．

ポール・サバチエ大学（フランス）の直接実行型のLISPマシンM3L (LISP Like Languageのマシン)[48)]は通常の方式での線型に連なった中間言語でなく，LISP形式のリスト構造で表現して直接実行する．各種高級言語でつくられたソーステキストをツリー構造の中間言語に変換し実行することを目的としている．アーキテクチャとしては，各レジスタに接続された16K (90ビット)のスタックや，マイクロプログラムカウンタとエスケープコードのスタックにより構成されるマイクロコントロール・ユニットに特徴がある．また1981年BBNから直接実行による高級言語マシンJERICHOが発表されInterLISPとPascalをサポートしている[65)]．同年，メキシコ国立大学ではZ 80を5台並べ，マルチプロセッサでpure LISPの範囲内で並列処理を行うAHRマシン[59)]の開発を進めている．

そして今日的なアプローチとしては，LISPマシンのワンチップ化の設計がMITで進められ，製作はゼロックスPARC等の協力を得て，Scheme-79と名付けられた1チップマシンができ上がっている[38),57)]．これはアーキテクチャとしてCONS, CADRマシンの延長線上のものではない．

米国ではすでにCADRの商品化が行われ1980年からLMI (LISP Machine Inc.)社のCADRマシン[66)]，Symbolics社のLM-2が発売されており[55)]，LM-2の改訂強化版のSymbolics 3600が本年から発売されようとしている[61)]．またLMI社からは，やはり強化版のLambdaマシン[68)]が発表されている．Lambdaは，標準では32ビットマシンであるがオプションで40ビットマシンとなり仮想アドレスは32ビットとなる．ゼロックスのDoradoとその下位機種Dolphinが市販されそうな話題もあるが，81年からDolphinをベースにした1100 Scientific Information Processor[67)]と名付けられたパーソナルなInterLISPの環境を提供するマシンが発売されている．

2.3 わが国における LISP マシン

わが国では，1974年7月に本学会記号処理シンポジウムが蓼科で行われ，主として商用汎用機上に作成されたLISPの処理系のほとんどが報告されている[11)]．これはその後のわが国におけるLISPマシンをも含めたLISPシステムのセル容量，処理速度等の向上に大きな影響を与える結果[14),30),37)]となった会合であった．

ここでは電子技術総合研究所においてHP 2100でボブロウスタックをベースにつくられたBマシン[13)]上でLISPマシンを作製中の報告がある[12),16)]．そして76年には同所でつくられた汎用エミュレーションのための計算機ACE[15)]上にさらに拡張されたLISPを実現している．また80年からは通産省大型プロジェクト「パターン情報処理システム」において開発された16ビットのマイクロプロセッサPULCE[25)]を用いてパーソナルマシンとしてのLISPマシンの開発が行われている[45)]．1976年パーソナルな利用を目標に当時一般に関心の高かったインテル8080を用いたLISPマシンALPS[17),36)]が青山学院大学で作られ，その経験を基礎に80年から，ビットスライスのマイクロプロセッサで新たに開発した80ビットのマイクロ命令幅をもつLISP Function Unit LFUによりALPS-IIへと発展中である[70),72)]．1978年には理化学研究所において，各種興味あるアーキテクチャにさらにハッシング・ハードウェア[19),22),51)]を加え，SISDの範囲内でできるだけ並列処理を盛り込み，REDUCE等による数式処理を目指してFLATSマシン[20),72)]と名付けられた世界的にも最大規模のLISPマシンの開発が始まっている．

同年，慶応大学ではリスト処理とGCにそれぞれ対等のCPUを分担配置させ，I/Oプロセッサとともに3台のプロセッサ[21]によるLISPの並列処理システムが開発されている[32]．また神戸大学では，ビットスライス・マイクロプロセッサ Am 2900 による LISP マシンの製作が行われ，当時の大型機上の LISP と処理速度を競っている[29),41]．

並列 GC のアルゴリズムの提案とその LISP への実装を目的として，電電公社武蔵野通研において，GC用ハードウェアと TOSBAC 40 L 2台による LISP マシンを構成し，実時間 GC の実験が報告されている[31),47]．さらに同所では LIPQ，TAO などの会話型 LISP 処理系製作，使用等の豊富な経験の上に立った，バックエンド・プロセッサとして位置するメガセル級の LISP マシン-ELIS の開発が 1980 年から始められ[50]，現在すでにハードウェアのデバッグが済み，その上で働く 64 ビット幅のマイクロプログラムの開発を急いでいる[71),72]．また LSI 化を念頭においた設計となっている．そしてやはり80年からデータフロー制御による LISP の並列処理マシンの検討を始めている[56]．

1979年京都大学ではマイクロプログラム可能な専用ハードウェアによる LISP 向きマイクロ命令の設計を行った LISP マシン NK 3 において，LISP プログラムの実行結果の報告[39]があり，同年大阪大学では筆者らによるリスト処理そのものの並列処理を狙った複数台の EVAL プロセッサによる EVLIS マシンが提案され[43]，自作のプロセッサ EVAL Ⅱ[69),73] の2台により82年に一部稼動を始めている．EVLIS マシンは取り扱う LISP として pureLISP に限定していない．

1981年，国産のマイクロプログラム可能な市販のミニコン NEAC-MS-50 による LISP マシンのプロジェクトとして AIM 1.0 が東北大学で開発され，汎用大型機上の LISP システムの処理速度に近い性能を示している[64]．計算機メーカが開発中の LISP マシンとしては，メモリネックの解消を狙って，メモリと cpu を一体化した VLSI 志向のロジックインメモリのセルを階層的に接ぐ，セルラ・アーキテクチャの LISP マシン（日本電気・中央研究所）[62]，LISP の諸環境でのスタックの多用に備えて，高速仮想スタックを中心とした LISP マシン（富士通研究所）[74]がありそれぞれ今後の成果が期待される．

3. LISP マシンのアーキテクチャとマシン例

以上に述べた各マシンは，アーキテクチャにおいてそれぞれ独自の工夫や主張をもっており，その詳細にははなはだ興味ある内容が多くすべてに触れるべきであるが，それらのうちには他稿[18),40),44),46),58),63),75]にも述べられているものもあるので，ここでは以下に述べるマシンを例に LISP マシンのアーキテクチャの一端を示す．

3.1 カーネギー・メロン大学の LISP マシン[6]

LISP のインタプリタのマイクロコード化を狙ったプロセッサの設計について述べており構想に終ったが，LISP の処理に有効な多くのアイディアを実現しようとしているのでまずはじめに LISP マシンの原型としてここに紹介する．

命令フェッチ／データ・フェッチの比を最小にすることに主眼をおき当時の PDP-10 の InterLISP に比較して1桁以上の速度の向上を目的としていた．その設計における主な特徴は，

1） プロセッサは比較的大容量のスクラッチパッドメモリと使用目的の異るキャッシュメモリ（64語）を2つもち，キャッシュの使用目的は1つは LISP プログラムおよびデータ用，もう1つはマイクロコード用にあてられている．

2） 1レジスタに1引数で関数に引数を渡す等，スクラッチパッド的に使用されるレジスタ群．

3） バス構成は2つのデータ・バスが並列に取り扱える．

4） 各バスにはマトリクス状のハードウェアのスイッチを備えており1語内のビット列を制御し，バイト単位の取扱いを効率的に行う．

5） コンパクト GC とインクレメンタルな並列 GC の採用．

6） タイプタグを使用し，データに基づいたオペレーションを行うタグアーキテクチャ．

7） コンパクト・リスト構造（CDR コーディング）．

8） プロパティ・ディスクリプタ・テーブルをプロセッサ内にもち，P-リストサーチの速度を向上させ，データメモリのフェッチ回数を半減させる．

9） Structure-Access Vector (SAV) の採用により CADDR 等の CAR，CDR の連続は，**図-1** のようにエンコードして，専用のカウンタとシフトレジスタでデコードすることにより特別処理される．

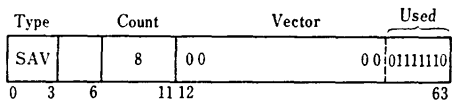


図-1 (CADDR(CDDDDR(CAR L)))の SAV エンコード

などが採用されている.

3.2 CONS[9]/CADR マシン[23),28)]と Scheme-79[38),57)]

CONS マシンの設計者 Knight はその論文[10]の冒頭において, ゼロックスの Alto と, カーネギー・メロンの DEC PDP-11/40 のシステムに影響をうけたことを述べている. CONS の強化版 CADR マシンではメモリに作られたスタックの先頭部分を格納している 1K 語のAスクラッチパッドメモリと, 情報へのアクセスの並列性を向上させるためAスクラッチパッドメモリの最初の 32 語のコピーをもっているMスクラッチパッドメモリをソースにするA, M 2つのバスがあり, キャッシュのように働き主記憶へのアクセスの大部分をこれらのスタックにおき換えマシンの高速化を計っている. また 16K語の WCS を有し, 32 語のマイクロサブルーチンのリターンスタックと, 2 K語のディスパッチメモリによって, プログラムコントロールの高速化を, 1命令の先読みと命令の実行をオーバラップさせることとともに行っている.

LISP ノードは, Qと名付けられた語を2語使って構成されるが, 連続したN要素のリストに対してはCDR コーディングによりN個の連続した領域に形成させる. LISP に対して特徴的なメモリ管理のアイディアとして, 関数 CONS は引数が3個あり, 第3引数は CONS して作られるノードの領域を指定することができる. 1つの領域は連続する頁で構成され, 1頁は 64 個のQからなる. 各領域はその使用目的に応じて, データ・タイプにこだわることなく関係する独自のフリー・ストアレージ・リストを構成し仮想記憶のページングのコントロールも含めて, データに局所性をもたらしており, これらの領域のもつ具体例としては昇順に並んだフリーポインタ, フリーページのリストあるいはフリーノードのリスト等がある. これらのリストは, ある一定の用途のものを記憶するための領域として選択的に利用させたり, 大容量の作業用のメモリとして分離させることができる. そして利用者はある領域に対してリードオンリイの宣言をしたり, 他の領域と関係なく, GC を選択的に行ったりして GC の速度向上に役立てることができる. このマシンの特徴ある機能を支えるものに invisible ポインタがある. これは間接アドレスとよく似た機能をもつもので, 間接アドレスが命令語に間接指定を行うのに対して, 図-2のデータ・タイプで指定されるように, データ語に間接指定がなされる. これの応用としては CDR コーディングされた状態において関数 RPLACD を行った結果, 圧縮形式のリストの線型状態が乱された場合においてもインビジブルポインタを用いることにより, 関数 EQ 等における同値性が保存される. その他, 変数の束縛, GC 等特徴ある LISP マシンとしての各種機能の実現に役立ち, CONS/CADR マシンの性能向上に寄与している.

CONS/CADR マシンが線型の命令列であるのに対して Scheme-79 はリスト構造をもつ S-コードと称する機械語で実行される. データ, プログラム, スタックが一様にヒープ上にリスト構造として置かれ, リスト構造やリンクされたレコードの作成等メモリ関連の仕事や, CAR, CDR, CONS, EQ 等の関数は直接実行できる機械語の OP コードとして準備されている.

2レベルのマイクロプログラムで, state machine の実現に PLA 方式を取り入れたアーキテクチャとなっている.

3.3 Dorado[49),60)]

ここでは Dorado をとりあげ, 多くの LISP マシンで命令実行の高速化のために工夫がなされているパイプライン処理と, LISP プログラムの実行時に全実行命令中約 40% を占め高い使用頻度[70]となる分岐命令等のマイクロ命令の設計の一部について紹介する.

1マイクロ命令内で行える処理の並列度をあげることはデータバスすなわちオペランドソースとなるバスの種類を増すこととなり, また並行して処理する機能

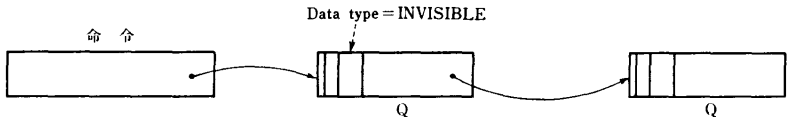
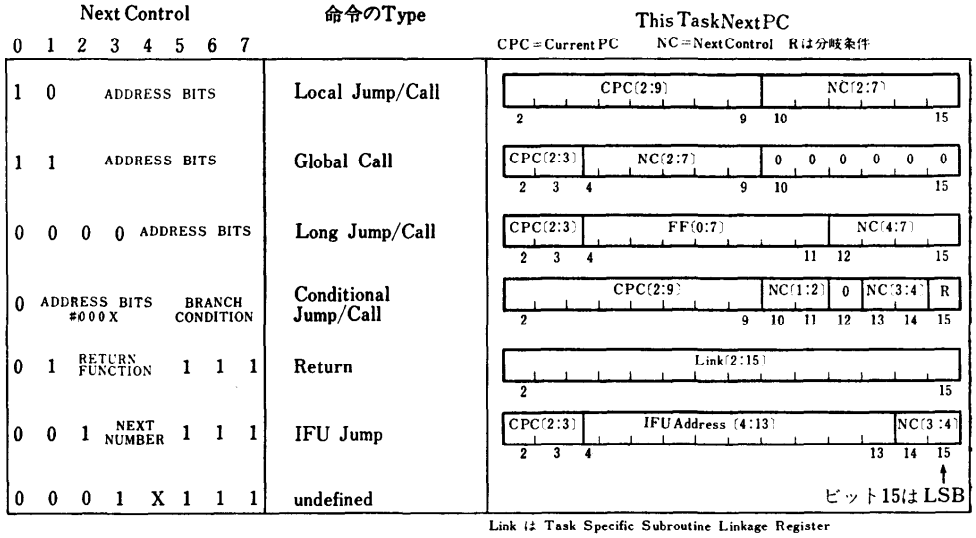


図-2 インビジブル・ポインタによる間接指定

RAddress	ALUOP	BSelect	Load Control	ASelect	Block	F F	Next Control

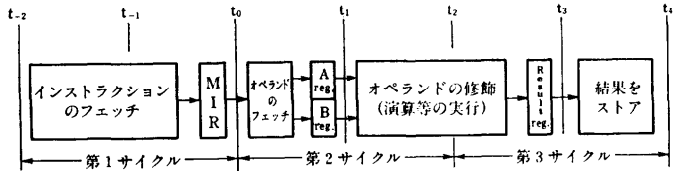
MIR の各フィールド

	ビット数	
RAddress	4	256 個の汎用レジスタ・バンク RM のアドレス.
ALUOP	4	ALU オペレーションまたはシフタのコントロール.
BSelect	3	Bバス上のソースの選択，定数も含む.
LoadControl	3	RM とタスク識別レジスタTへのロードの制御.
ASelect	3	Aバス上のソースの選択，メモリ参照の開始.
Block	1	次のアドレスとして I/O タスクか，タスク0のためのスタックオペレーションかの選択．タスク0はエミュレーションの実行.
FF	8	指定された関数機能の受場所.
NextControl	8	NextPC の取扱いの指定.

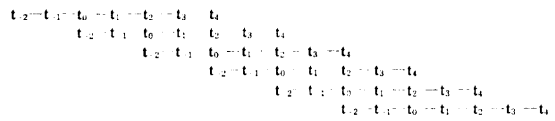


NextAddress の形成

(a) Dorado のマイクロインストラクション・レジスタ (MIR) と各フィールドの機能



(b) 命令のパイプライン



(c) 命令のパイプラインとそのタイミング

図-3 Dorado のマイクロ命令

の指定等により命令の各種フィールドの増加をきたし命令幅を広くとることになって WCS の容量の増大を招く．Dorado ではまず図-3(a)において NextControl のアドレスを小さくするためにマイクロストアを頁に分割しており次にフェッチされる命令のアドレス NextPC はカレント頁（現在実行中の頁）内の数ビットの相対アドレスとなり，そして branch, call, return, 他の頁への移動は NextControl の8ビットのアドレスの中にエンコードされた type フィールドで指定している．プロセッサは図-3(b)のマイクロサイクルの開始にあたり，まず type フィールドをデコードしてカレント頁 No. や，リターン・リンク等の情報に基づき NextPC を計算する．そしてこれらに要する時間はパイプラインで吸収し，また，マイクロ命令のアドレスをなるべく頁内に収めることをアセンブラによって解決している．

パイプライン方式では，条件分岐命令の実行に際し，条件判断の結果によって次の命令のフェッチの変更と実行のタイミングが問題になる．Dorado では，分岐は8分岐条件の1つを選択することになっており，判断結果は NextPC の下位ビットを修飾変更する．修飾は下位のビットに対して判断結果との論理和が命令フェッチサイクルの半分でなされるが，判断結果がアドレスの一部としてアドレスの再生成が行われるのではなく，チップセレクトとして働き遅れなく所定のサイクル内で高速に次の命令をフェッチすることができる．そのためこれもアセンブラによってあらかじめ分岐先アドレスは条件が偽のときはマイクロストアの偶数番地に，真のときはその1つ上の奇数アドレスに分岐先を置いている．

その他の工夫として，Dorado は各マイクロサイクルでプロセッサの機能を制御するために Function Field FF の8ビットで示すプロセッサのオペレーションのエンコードを行っている．このフィールドはマイクロ命令実行サイクルの始め（t_0〜t_1）で，デコードされ，I/O バスの制御，メモリや命令フェッチユニットのステイトの読出しや，それらへのステイトのセット，ある語の任意フィールドからの抽出その他等の比較的頻繁に使用されないプロセスが実行される．またこの FF は8ビットの定数アドレス，あるいはマイクロアドレスを表すビットの一部としても使用される．

そして各サイクルで FF の指定するオペレーションは1つであるが，パイプライン上にある多くのオペレーションに必要なデータのパスがあわせて存在するときは，並列に多数の機能が実行されることになる．さらにディスプレイ，ディスク等入出力装置に関連してタスクの切換等のタスク・レベルでの制御では2ステージ，キャッシュ・仮想記憶に伴うメモリ管理7ステージとそれぞれ強力なパイプライン化がなされて高速化に貢献しているが原論文に譲り割愛する．

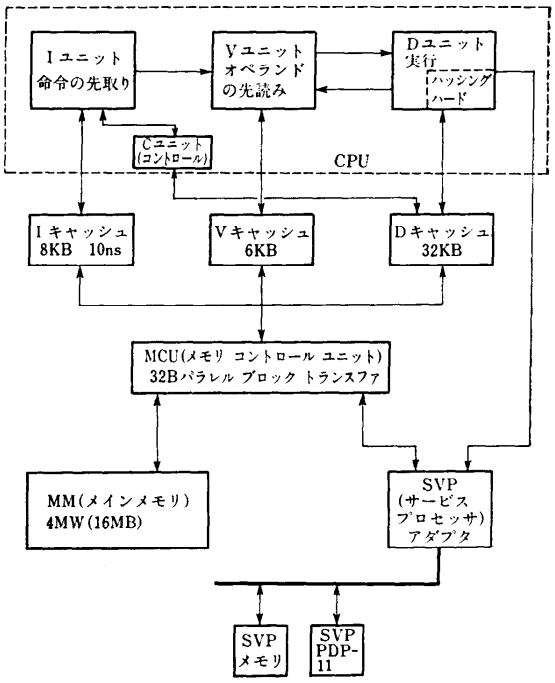


図-4 FLATS マシン構成図

3.4 FLATS[72]

FLATS は LISP マシンとして 10 MIPS の処理能力を設計目標としており，中大型機のバックエンドプロセッサとして位置づけられ，本特集号の出る頃には稼働を予定されている．マシン構成を図-4に示す．特徴あるハードウェアとして並列ハッシング・ハードウェア[19),51)]があり，データ空間に取られたハッシュ表とともに働くハッシング番地生成機構，ハッシュ操作制御機構から成り立っている．仮想アドレス空間を，コンパイルド・コードの入る I 空間とデータの入る D 空間に分けておりそれぞれアドレスは24ビットである．命令は1語4バイトで最初の1バイトが OP コードで残りの3バイトで3アドレス形式の命令体形をもち，128個の汎用レジスタ，127個のローカルス

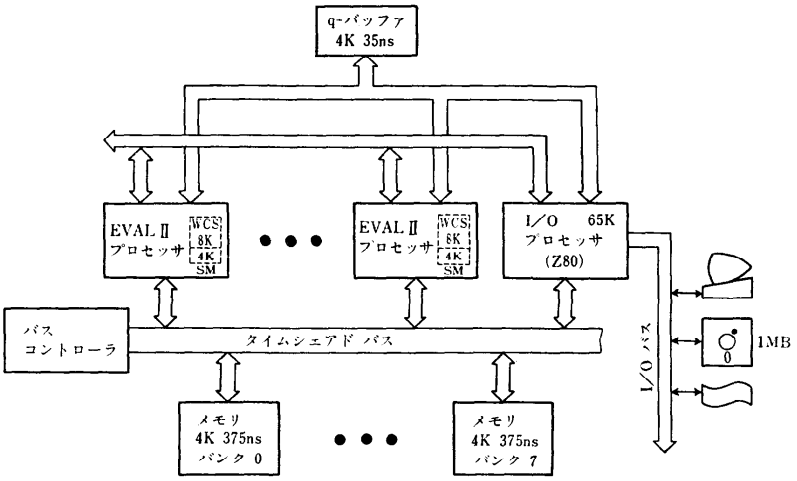


図-5 EVLIS マシンの構成図

タック・フレームレジスタをアクセスし，またその働きとしては0，1，2，3の各アドレス方式を含んでいる．各レジスタは，3つのコピーをもつレジスタとしてVキャッシュ上において，3並列読出しポートとともに実現しプログラム実行の高速化を狙っている．そして基本関数のハードウェア実行機構が働くときは2マシンサイクルで終了し，また3ステージのパイプライン，アドバンスコントロールを行って命令実行時間の消去を計っている．CDR コーディングとともに，リバース CONS をハードで準備しており，実行時データ型検出，ハードウェアスタック，メインメモリ空間のセグメンテーション，配列の境界検査機構と，汎用レジスタマシン，スタックマシン，タグアーキテクチャ等の諸技術を強力に取り入れている．そして，24 ビット×24 ビットの乗算を 30 ns で行うハードウェアを内蔵させているのもシステム全体への積極的なハードウェア化の姿勢の現れであり数式処理の目的からすれば特筆すべきであろう．

3.5 EVLIS マシン[43),73)]

このマシンは，LISP インタプリタの中の関数 EVLIS の内容が第1引数の要素の数に分割されたリスト処理のタスクとして並列に処理でき，かつインタプリタ実行中に頻繁に使用されることに着目して，EVLIS の引数評価を複数台の EVAL プロセッサで実行させている．処理の対象となるプログラムは，pureLISP に限定せずリスト書換えを含む副作用を伴う関数の使用も許している．製作する前に行った EVAL プロセッサの台数に関するシミュレーションでは，台数に比例して並列処理効果の認められる仕事もあるが，TPU[37)]等で代表されるプログラムでは3台程度が適している．各プロセッサは対等であって，各プロセッサの実行対象となる発生したタスクと，それらの各プロセッサへの取込み配分等のタスク管理の時間的なオーバヘッドは，ハードウェアで準備された**図-5** の q-バッファと名付けた FIFO スタックを介して行うことで減少させている．最初に製作した EVAL プロセッサの改訂版である EVALⅡ[69)]は，q-バッファへのアクセス等の並列処理用機能に加えて単体の性能としても LISP マシンとして十分な処理速度が得られるよう設計されている（**図-6**）．

EVALⅡプロセッサは，高速のスクラッチパッドメモリ（SM）を備えており，マイクロプログラムのリターンアドレス，データの退避のためのスタック，ディスパッチ・テーブルおよび割込みのためのアドレステーブル等，作業領域として利用される．SM はアドレス・レジスタ（SPR，SAR），マイクロ命令（**図-7**）のKフィールド，により任意にアドレス指定が行え，またスタックポインタ・レジスタ（SPR）は自動的に増減を行いスタックとして利用するときの push，pop の管理を容易にしている．また Top-Bottom の境界が指定でき，境界を越えたときは割込みが発生しスタック管理のオーバヘッドを少なくしている．

リストをたぐることの多い LISP のために，メモリから読み込んだデータをそのまま次のアドレスとして使用できるようにした PCAR，PCDR のレジスタをもっており，CAR，CDR 演算をメモリアドレスレジ

スタ (MAR), WCAR, WCDR と異なるパスを通して主記憶 (MM) にアクセス可能であり，命令内での MM へのメモリアクセスが競合するときは PCAR, PCDR からのアクセスが優先される．

リスト処理ではオペランドとしてポインタが頻繁に取り扱われることを考慮してバス上ではデータとアドレスを区別せず，1つのデータとして取り扱っている．ALU 演算は3アドレス方式であり，またデータ・タイプや関数のタイプに対する分岐，多方向分岐のためにデータ中の指定フィールドをマスカにより抽出しその値によって対応する処理ルーチンへの分岐を高速に処理するディスパッチ機能を有している．マイクロ命令は分岐のためのフィールドを設けてあり他の演算と並行処理がなされる．マイクロ命令のフェッチと実行は当然パイプライン化されている．

マシンのハードウェアとソフトウェアのデバッグ，LISP プログラム実行時の動特性等の諸情報を収集するために，EVAL Ⅱの各レジスタ，バス，カウンタ等の要所に埋め込まれたシフトレジスタがありそれらを直列につないだ診断用のインタフェースが準備されている．このインタフェースから I/O プロセッサの制御によって，バスやレジスタ，フラグ等の状態を随時監視でき，またそれらに I/O プロセッサを介して外部より任意の値や，状態をセットできる．

EVAL Ⅱ単体の LISP マシンとしての処理速度は，プロセッサのサイクルタイムが 200 ns における TPU-6[37] で 3926 ms (シャローバインド)，5848 ms (a-list) であり，ACOS 900 における 10867 ms (a-

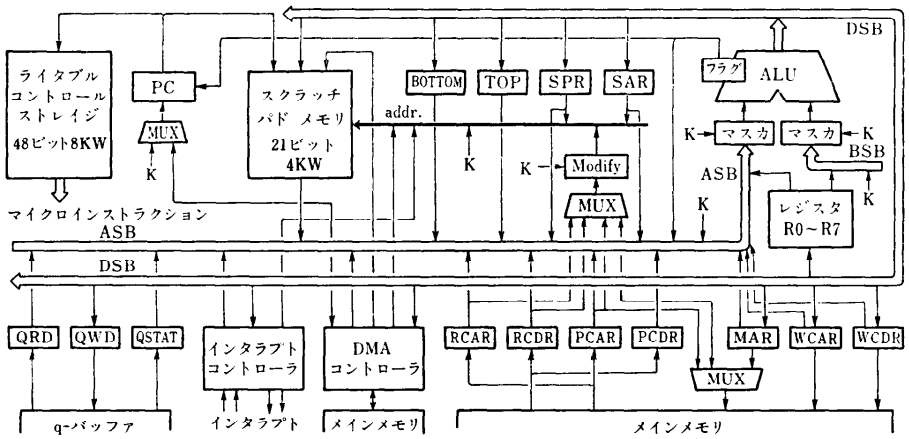
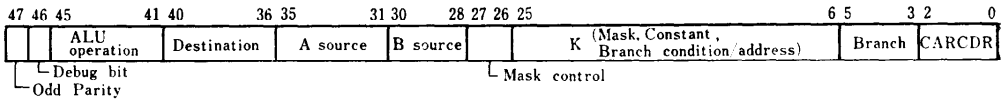


図-6 EVAL II のブロック図



各フィールド	機　能
○Odd parity	
○Debug bit	診断用カウンタで計数され，ブレークポイントとして，また診断用インタフェースを介して I/O プロセッサにより情報の収集を行う．
○ALU	ALU 演算の指定．
○Destination	演算結果の格納先の指定，MM アクセスの起動．
○A source	Aソースバス上のリソースの指定，q-バッファ リードの起動．
○B source	Bソースバス上のリソースの指定，Mask control フィールドが3のときは LISP 用定数*T*等や，オール1を選択．
○Mask control	A, B のソースバスを選択しマスクを指定．マスクパターンはKフィールドで与える．
○K	Branch フィールドとの関連により，(1) Jump, Call, Return, (2) デイスパッチ，(3) SM アドレッシング，(4) マスクパターンの4通りに利用される．(1)～(3) のときはマスクパターンの指定は2ビットで，それぞれタグ部4ビット，ポインタ部16ビット分のマスクを担当する．
○Branch	Jump, Call, Return, Dispatch の指定．
○CARCDR	PCAR, PCDR レジスタの内容をアドレスとして MM リードを行う．

図-7 EVAL II のマイクロ命令

表-1 海外 LISP

開発／発表年	1973 (1979)	1974	1977	1978	1978	1978	1979	1978 [0] 1979 [1]
マシン名	Alto	CONS	MBALM	CADR	MLS	M 3 L	Scheme-79	Dorado (0, 1)
機関名	Xerox PARC	MIT	Utah 大学	MIT	Illinois 大学	Paul Sabatier 大学	MIT	Xerox PARC
プロセッサ	Alto プロセッサ (170 ns)	実験用プロセッサ+PDP−11/10	B 1726/B 1800 167 ns	CONSの強化版+PDP−11	トランスレーション (TP), エキスキューション (EP), メモリ (MP) の3プロセッサ	AMD 2900 ファミリ 500 ns	1チップ LISP マシン	特製プロセッサ ECL 10K ファミリ 50〜60 ns
主記憶	64 K (16 bit) (256 k)	64 K (32 bit) 1 M max	128 KB 350 KB	64 K (32 bit) アドレス 22 bit	2.2 MW RAM 256 KW ROM	64 K (40 bit) アドレス 16 bit から 24 bit へ拡張	32 K (32 bit)	256 KB
仮想記憶	——	4 M (32 bit) アドレス22bit (当初)	——	16 M (32 bit) アドレス 24 bit	——	——	アドレス 24 bit	4 M (16 bit)
制御記憶	1 K (32 bit) RAM (2 K) 1 K (32 bit) ROM 170 ns (3 K)	4 K (42 bit)	4 KB —	16 K (48 bit)	——	4 K (32 bit) RAM, 256 (100 bit) ROM		4 K (34 bit) 16 K max.
バッファ, キャッシュ		32 (32 bit) 256 (32 bit)	— 4 KB	1 K (32 bit) Aスクラッチパッド 32 (32 bit) Bスクラッチパッド 2 K (17 bit) デスパッチメモリ	——	——		4 K×1, 8 K×1, 32 K×1
スタック		1 K (32 bit) pdl バッファ 200 ns	32 (24 bit) マイクロアドレススタック	32 (15 bit) μサブルーチンスタック 1 K (32 bit) pdl	1 K (32 bit)	16 K (90 bit) 各レジスタに直結		(64(16 bit)/スタック)×4
レジスタ	64 (16 bit) (32 R レジスタ+8×32 S レジスタ)	——	16 (48 bit) スクラッチパッドレジスタ	主として制御用, アドレス関係用として各bit 幅に亘り多数	——	16 (16 bit), 256 (16 bit), 5 (16 bit), 1 (8 bit), 1 (6 bit)	主として 10 レジスタを 6 EVAL 用, GC 用に共用	256 (16 bit) 汎用, その他
GC方式	ビットマップ, (レファレンスカウンタによるインクレメンタルなダイナミック GC)	インクレメンタル, コンパクト GC	ビットマップによる	CONSに同じ	リアルタイム		フルスケールバージョンでフル GC	インクレメンタル, レファレンスカウント
その他特徴	Interlisp, Bytelisp 200 ns/命令 2,5 MB ディスク 3 Mbit/S 通信機能 定数 256 (16 bit) PROM ディープバインド	Maclisp PDP KA-10 上の Maclisp の3倍の速度 10 M bit/S でパケットによるネットワーク可	Funarg のサポート clever ポインタ 14,000 op コード/秒 KA-10 の 3.2 倍 (MIL コード) REDUCE ではヒープが65 Kのためスワップが問題 CDC 7600 90 kop コード/秒 CRAY-1 200 kop コード/秒	Maclisp コンパイラオブジェクトは 16 bit のオーダコード 診断用インタフェース MIT で23台稼動 (1981年9月現在)	機能分散型マルチプロセッサ CDR コーディング GPSS シミュレーション: CAR 1.8 μs CDR 2.5 μs CONS 7.0 μs COND 4.5 μs EQ 3.0 μs	高級言語の直接実行 シミュレーション: CAR 2 μs CDR 2 μs CONS 10 μs ATOM 3 μs RPLACA 4 μs	機械語はリスト形式の S コード 1 MHz のクロックで 128 K のリストノード (=1 M ヒープ) を6秒以下でコレクション 24ビットデータフィールド・7ビットタイプ・1ビットストアレジアロケータ用	Interlisp-D ディープバインド DEC-KA-10 の5倍 80 MB ディスク コンパイラオブジェクトは Byte コード メインメモリ・バンド幅 533 Mbit/s

マシン比較

1980	1981	1981	1981	1981	1981	1981	1982
Dolphin	Symbolics LM-2	LMI の CADR マシン	AHR マシン	Symbolics 3600	1100 Scientific Information Processor	JERICHO	Lambda プロセッサ
Xerox PARC	Symbolics 社	Lisp Machine Inc. (LMI)	メキシコ国立大学	Symbolics 社	Xerox EOS	BBN	LMI
特製プロセッサ TTL	MIT CADR の商品化 180 ns	左に同じ 180 ns	Z80 5台＋ディストリビュータ	LM-2 の強化版	Dolphin をベースにしたプロセッサ 200 ns	AMD 2903, 2910	LMI LISP マシンの最新版
256 KB	256 K (32 bit)	128 K (32 bit) 2 M max.	8 K (32 bit) プログラム用 (ノード) 64 K (32 bit) リストとアトム用 16 K (32 bit) 変数用	64K (36 bit), 256 K max.	576 K (16 bit)	512 KB, 16 MB 720 ns	1 MB 32 bit/1 W
4 M (16 bit)	CADR に同じ	CADR に同じ	——	256 M (36 bit)	4 M (32 bit)	アドレス 22 bit	アドレス 24 bit, 32 bit max.
	12 K (48 bit)	12 K (48 bit) 16 K max.	——			4 K (64 bit) RAM 16 K max. 2 K (64 bit) PROM	
		1 K (32 bit) Aスクラッチパッド 32 (32 bit) Mスクラッチパッド 2 K (24 bit) ディスパッチメモリ	——			——	40 bit マシンのときにもマイクロプログラム変更不要
		32 μ サブルーチンスタック 1 K (32 bit) pdl スタック	4 K (19 bit) FIFO 55 ns			——	
			Z 80			256 (32 bit) スクラッチパッド 16 (16 bit) シフトコントロール	
インクレメンタル, レファレンスカウント			ノルマル・シリアル(第1版) パラレル・インクレメンタル (第2版)		インクレメンタル, レファレンス・カウント		
Interlisp-D ディープバインド DEC KA-10 に同等 Xerox 社内等で 150 台稼動中 Dorado の低コスト版	MIT CADR マシンの LSI 版 80 MB ディスク ネットワーク MIT Chasonet, Xerox Ethnet 1981年発売 $ 99,000 カラーグラフィック 576×454 ピクセル DEC 2060 ×1/2	MIT で 20 台以上稼動 左に同じ $ 80,000 DEC 2060 ×1/2	並列 purelisp マシン 1981年8月虫とり中 メモリ プログラム用アクセスタイム 55 ns リストとアトム用アクセスタイム 150 ns 変数用アクセスタイム 150 ns	Zetalisp, Maclisp VAX 11/780 の2倍 DEC-20/60 に同等 LM-2 の3倍 67 MB ウィンチェスタディスク 1 MIPS 1982年3月発売予定 約6万ドル	ディープバインディング CDR コーディング Interlisp-D, Mycin ディスプレイ 17″ ビットマップ 1024×808 ピクセル 23 MB ウィンチェスタディスク KL-10 に同等以上 1982年発売納期 6カ月 59,719 ドル	ディスプレイ 1024×1024 ピクセル 195 MB ウィンチェスタディスク 価格 Delphin 以下?	約2倍の性能向上 300 MB ディスク Lambda プロセッサ5万ドル Western Digital の Nu マシンを必要とする 1.5万ドル 2台以上の Lambda の接続可 オプションで 40 bit マシン

表-2 国内 LISP

開発／発表年	1974	1976	1978	1978	1978	1978
マシン名	Bマシン	ACE システム	LISP マシン	LISP マシン	FLAT マシン	並列 GC マシン
機関名	電子技術総合研究所	電子技術総合研究所	慶応大学	神戸大学	理化学研究所	武蔵野電気通信研究所
プロセッサ	HP 2100	自家製プロセッサモジュール 250 ns +NOVA 800	PM/I (500 ns) +PM/II (600 ns) +NOVA (28 K (24 bit))	Am 2903, 2910 300 ns +LSI-11	自家製 ECL 50 ns	TOSBAC-40L×2 +GC 用ハード (200 ns 以下)
主記憶	4 K (16 bit)	2^{16}(16 bit) 800 ns	16 K (16 bit)	64 K (32 bit) 675 ns	4 M (32 bit) 400 ns	TOSBAC-40L のメモリ
仮想記憶	—	—	—	—	—	—
制御記憶	1 K (24 bit)	約 4 K (32 bit) を主記憶におきマイクロキャッシュで使用	1 K (24 bit)^{I} 1 K (24 bit)^{II}	4 K (56 bit) 150 ns	1 K (150 bit), 256 (50 bit) 50 ns	256 (12 bit)
バッファ，キャッシュ等	—	256 (32 bit) マイクロ用 16 (32 bit) レジスタ拡張用 2 K (32 bit) 一般データ用	—	—	8 KB Iキャッシュ， 6 KB Vキャッシュ 32 KB Dキャッシュ	—
スタック		16 (16 bit)×2 汎用 8 (16 bit) ×1 マイクロ用	16 レベル^{I}, 16 レベル^{II}	4 K (16 bit) 70 ns	1 K (32 bit)	1 K (16 bit)
レジスタ	2 スタックポインタ，2 変数バインド用 4 ワーキング	16 (16 bit) 主レジスタ 9 (16 bit) マスクレジスタ	16^{I} (24 汎用，8 特殊)^{II}	16 (16 bit) レジスタファイル	128 (32 bit) 汎用 127 (32 bit) ローカル・フレーム	16 (16 bit) レジスタファイル
GC 方式	—		ダイキストラの並列アルゴリズム	ビットアドレッシング回路とビットテーブルによる	ビットベクトルとビット操作用ハードによるコンパクト GC	提案アルゴリズム：走査要求フラッグを用いる刻印アルゴリズムによる並列 GC 用ハード
その他特徴	コルーチン Bobrow スタックのファームウェア化 L-1.6 (TOSBAC 5600) と同等のパフォーマンス	シフタ：0〜15 ビット，左，右，回転 単長倍長を 1 マシンサイクル Bobrow スタック	インタプリタプロセッサ，ストアレッジ管理プロセッサ，I/O プロセッサによる密結合機能分散型マルチプロセッサ構成	Fast Lisp シャローバインド，ディープバインド スタック間転送 1 マイクロ命令サイクル マッピングメモリ回路 1 K (36 bit) {インプット：主記憶アドレス アウトプット：利用種別のコード 3 ビット} TPU-6 6728 ms	ハッシングハードウェア CDR コーディング，RCONS 10 Mips 目標 数式処理用マシンのハードウェア化	LIPQ に近い実験用 LISP TPU-6 1010秒 (並列 GC) 1050秒 (通常 GC) GC アルゴリズム実験システム

マシン比較

1979	1979	1980	1980	1980	1981
NK 3	EVLIS マシン	ELIS	パーソナル LISP マシン	ALPS-II	AIM-1.0
京都大学	大阪大学	武蔵野電気通信研究所	電子技術総合研究所	青山学院大学	東北大学
自家製プロセッサ 350 ns +INTERDATA 8/32	自家製 EVAL-II×2 100 ns +Z 80	Am 2900 180 ns +PDP11/60	PULCE プロセッサ	自家製 LFU+i 8086 300 ns	NEAC MS-50
36 K (8 byte) セル INTERDATA のメモリ共用	16 K (40 bit) 375 ns	512 K (64 bit) 550 ns	128 K (32 bit)	128 K (44 bit) 512 K max.	64 K (32 bit) セル 0.9 μs
—	—	16 M (64 bit)	—	—	—
4 K (42 bit) 100 ns	8 K (48 bit) スクラッチパッド 100 ns EVAL-II 1台につき	16 K(64 bit) 80 ns	14 K (32 bit) RAM +2 K (32 bit) ROM	4 K (80 bit)	2 K (64 bit)
4 (32 bit)×4 ブロック	4 K (21 bit) 35 ns q-バッファ	—	8 (32 bit)×1 レジスタファイル	2 (44 bit) バルクメモリ用バッファ	4 K (16 bit)
メモリ上のスタックの先頭数 W 〔2 K (4 byte)〕	4 K (21 bit) 100 ns EVAL-II 1台につき	32 K (32 bit) 60 ns, 3 (14 bit) スタックポインタ	8 (32 bit)×2 4 レベル (32 bit) スタック	4 K (20 bit)	スタック命令あり
WR 0〜15, INR 0〜7, μR 0〜7.	8 (20 bit), CAR, CDR 各1(20 bit)	4 (64 bit) 多目的レジスタ, 32 (32 bit) レジスタファイル	29 (16 bit) 汎用 7 (16 bit) マスク, その他8	32 (20 bit) ジェネリックレジスタ	7 (16 bit) 汎用 12 (20 bit), 2 (16 bit), 7 (8 bit)
LISP 1.5 に同じ	全プロセッサによる並列 GC			—	マーキング
ディープバインド タグアーキテクチャ シフト 1ステップで実行 (350+50 ×シフト数) ns 使用メモリ領域 36 K (8 byte) フリーセル 2 K (4 B) スタック 4 K W フルワード 0.5 K (16 B) アトムヘッダ 1 K (4 B) ハッシュテーブル	LISP 1.5 の複数台の EVAL プロセッサによる並列処理 ディープバインド, シャロー(単体用) TPU-6 3926 ms ACOS 900 の約3倍 高速診断用インタフェース (16カ所)	ディープバインド M 200 の2〜3倍 (コンパイラ) 1.5 倍 (インタプリタ) TAO LISP (DEC 11/60) の6倍, 20倍を目標 0.3 MLips (インタプリタ)	ミドル・バインディング スタックマシン 機能分散 LISP 言語と 1:1 のマクロ命令 7000 ゲート/チップ LSI RS 232 C	スタック: 絶対アドレス, ポインタ相対アドレス指定. スタック制御レジスタ SCR の採用 TARAI-4 51 ms(最速版) TARAI-4 155 ms ((ハンド)コンパイル版)	ディープバインド 関数タイプ: MSUBR ACOS 900 に近い処理速度

list) より十分高速であり，調整が進みサイクルタイムが 100 ns のときにはさらにその値の 0.64～0.74 倍に短くなるものと推定される．現在2台の EVAL-Ⅱにより稼動しつつあるが，シミュレーション[43]の値との比較等の詳細は別途報告される予定である．

EVLIS マシンにおける GC は，各プロセッサからの割込みを動機として，I/O プロセッサの制御により，全プロセッサの仕事を中断して，各プロセッサで並列に GC を担当する．

シングルプロセッサによる LISP マシンは近年その完成度を高めつつあり，さらに飛躍的な処理能力の向上実現にはマルチプロセッサによる並列処理に可能性を求めることになろう．データ・フローマシン，リダクションマシンも含め今後の並列処理による LISP マシンの発展に期待したい。

結びにあたり本稿に述べた内外の LISP マシンのアーキテクチャの諸元，機能等について一部ではあるが**表-1**，**表-2** に示す．なお，不備な部分も多々あるが，原論文はもちろん，すでにあげた解説，書物等もあわせて参考に，それらを補っていただければ幸いである．

4. おわりに

LISP マシンは計算機開発環境の著しい発展とともに研究対象としての試作機から実用機へと高性能を追求しながら早いテンポで成長を続けてきており，一面ではマシンあるいはプロセッサのチップ化によりさらに量産化が行われる時代を迎えようとしている．そしてバック・エンドプロセッサとしての高性能大規模 LISP マシンへの道を歩むかたわら，スタンドアロンの CAD, OA 等の専用マシンとしての高性能パーソナルコンピュータとなった LISP マシンが普遍する世代を実現しつつある．半導体回路技術の高度化，および新素子の開発は広く一般にメモリをも含めた計算機のスピードをますます高速なものとするであろう．そしてその上に立って，さらに経済性とともに高速高性能を追求するためには，よりすぐれたデータ構造，言語および新しいアーキテクチャへの休むことのない挑戦は続くものと考える．

以上内容が現状の一部報告にとどまったが，本稿が記号処理と LISP マシンあるいはコンピュータアーキテクチャに対する新たな関心を読者に起していただくきっかけにでもなれば筆者の喜びといたします．終りに臨み，文献の収集に際しご協力をいただいた日本電気株式会社中央研究所山本昌弘氏，東京芝浦電気株式会社総合研究所黒川利明氏，および bit 誌への原稿[75]をお送りいただいた理化学研究所井田哲雄氏にお礼を申し上げる．

参考文献

1) McCarthy, J., et al.: LISP 1.5 Programmer's Manual, The M. I. T. Press (1962).
2) Wigington, R. L.: A Machine Organization for a General Purpose List Processor, IEEE Trans. Comput., pp. 707-714 (1963).
3) Berkeley, E. C. and Bobrow, G. ed.: The Programming Language LISP: Its Operation and Applications, M. I. T. Press (1964).
4) 安井 裕，立花道明，山本昌弘：NEAC 2206 のための LISP について，情報処理学会昭和 41 年度全国大会予稿集，pp. 36-37 (1966).
5) Bashkow, T. R., Kroft, D. and Sasson, A.: Study of a Computer for Direct Execution of List Processing Language, Columbia Univ., Rept. TR-103 (1968).
6) Barbacci, M., Goldberg, H. and Knudsen, M.: C. ai-- A LISP Processor for C. ai, Carnegie-Mellon Univ., CMU-CS-71-103 (1971).
7) Feustel, E.: On The Advantages of Tagged Architecture, IEEE Trans. Comput. Vol. C-22 No. 7, pp. 644-656 (1973).
8) Deutsch, L. P.: A LISP Machine with Very Compact Programs, Proc. 3rd IJCAI, Stanford, pp. 697-703 (1973).
9) Greenblatt, R.: The LISP Machine, MIT AI Memo No. 79 (1974).
10) Knight, T.: CONS, MIT AI Memo No. 80 (1974).
11) 記号処理シンポジウム報告集，情報処理学会プログラムシンポジウム委員会 (1974).
12) 島田俊夫，山口喜教，坂村 健：LISP と LISP マシン，記号処理シンポジウム報告集，pp. 21-27 (1974).
13) 島田俊夫，山口喜教，坂村 健：LISP マシンとその評価，情報処理学会計算機アーキテクチャ研究会資料 74-7 (1974).
14) 中西正和：LISP コンテスト，bit, Vol. 7, No. 3, pp. 69-74 (1975).
15) 山口喜教，島田俊夫，守屋朋子，坂村 健：ACE 上の LISP マシン，電子通信学会技報 EC 76-13, pp. 67-75 (1976).
16) 島田俊夫，山口喜教，坂村 健：LISP マシンとその評価，電子通信学会論文誌，Vol. J 59-D No. 6, pp. 406-413 (1976).
17) 井田昌之：LISP マシン ALPS/I，昭和 51 年度記号処理研究会委員会報告集，pp. 142-165 (1977).
18) 島田俊夫，坂村 健，山口喜教：高級言語マ

シン，情報処理，Vol. 18. No. 4, pp. 386-394 (1977).
19) 後藤英一，井田哲雄：ハッシング・プロセッサ，情報処理，Vol. 18, No. 4, pp. 395-401 (1977).
20) 後藤英一，井田哲雄，相馬 嵩：記号処理向き計算機 FLATS の構想，情報処理学会研究会記号処理 1-1, pp. 1-9 (1977).
21) Tokoro, M. et al.: PM/II-Multiprocessor oriented byte-sliced LSI processor modules, Proc. NCC '77, pp. 217-225 (1977).
22) Ida, T. and Goto, E.: Performance of a parallel hash hardware with key Deletion, Proc. IFIP Congress, pp. 643-647 (1977).
23) Bawden, A. et al.: LISP Machine Progress Report. by the Lisp Machine Group, MIT AI Memo, No. 444 (1977).
24) Griss, M. L. and Swanson, M. R.: MBALM/1700: A Microprogrammed LISP Machine for the Burroughs B1726, Proc. of MICRO-10, ACM, pp. 15-25 (1977).
25) Iizuka, H. and Hara, Y.: Development of a high-performance universal computing element-PULCE, AFIPS Proc. NCC, pp. 1255-1264 (1978).
26) Fitch, J. and Marti, J.: SLISP-A Standard LISP Implementation for the Burroughs 1700, Utah Symbolic Computation Group. Operating Note, No. 37 (1978).
27) Williams, R.: A Multiprocessing System for The Direct Execution of LISP, Proc. 4th work slop on Computer Architecture, pp. 35-41 (1978).
28) Steele, G. L. Jr. and Moon, D. A.: CADR, MIT AI Memo No. ??? (1978).
29) 瀧 和男，金田悠紀夫，前川禎男：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 (1978).
30) 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).
31) 日比野靖：並列ガーベジコレクションアルゴリズムと LISP への適用，電子通信学会技報，EC 78-32 (1978).
32) 薄 隆，田丸喜一郎，所真理雄：マルチマイクロプロセッサによる LISP マシン，電子通信学会技報，EC 78-31 (1978).
33) Deutsch, L. P.: Experience with a Microprogrammed Interlisp System, Proc. of MICRO-11, pp. 128-129 (1978).
34) Griss, M. L. and Kessler, R. R.: REDUCE/1700: A Micro-coded Algebra System Proc. of MICRO-11, pp. 130-138 (1978).
35) Griss, M. L. and Kessler, R. R.: A Microprogrammed Implementation of Standard LISP and REDUCE on the Burroughs B 1700/B 1800 Computer, UCP-70 (1979).
36) 井田昌之，小林茂男，山方宏修：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).
37) 竹内郁雄：第2回 LISP コンテスト，情報処理，Vol. 20, No. 3, pp. 192-199 (1979).
38) Steele, G. L. Jr. and Sussman, G. J.: Design of LISP-Based Processors or, SCHEME: A Dielectric LISP or, Finite Memories Considered Harmful or, LAMBDA: The Ultimate Opcode, MIT AI Memo No. 514 (1979).
39) 長尾 真他：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).
40) 松崎 稔：LISP マシン開発の現状を探る，日経エレクトロニクス．3月19日，pp. 62-79 (1979).
41) 瀧 和男，金田悠紀男，前川禎男：LISP マシンの試作，情報処理学会論文誌，Vol. 20. No. 6, pp. 481-493 (1979).
42) Thacker, C. P., et al.: Alto: A personal computer, Xerox PARC report, CSL-79-11 (1979).
43) 安井 裕，斎藤年史，三石彰純，宮崎洋一：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資料記号処理 10-1 (1979).
44) 井田哲雄，後藤英一：数式処理プロセッサ，情報処理，Vol. 21, No. 1, pp. 19-27 (1980).
45) 山口喜教：パーソナル LISP マシンの開発，情報処理学会研究会資料計算機アーキテクチャ，37-9 (1980).
46) 坂村 健：スーパー・パーソナル・コンピュータのアーキテクチャ，bit, Vol. 12, No. 8, pp. 44-52 (1980).
47) Hibino, Y.: A Practical Parallel Garbage Collection Algorithm and Its Implementation, Proc. of 7th Symposium on Computer Architecture, pp. 113-120 (1980).
48) Sansonnet, J. P., Castan, M. and Percebois, C.: M 3 L: A List-Directed Architecture, Proc. of the 7th Symposium on Computer Architecture, pp. 105-112 (1980).
49) Lampson, B. W. and Pier, K. A.: A Processor for a High-Performance Personal Computer, Proc. of the 7th Symposium on Computer Architecture, pp. 146-160 (1980).
50) 日比野靖，渡辺和夫，大里延康：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).
51) 井田哲雄：ハッシング・ハードウェアの試作とその応用，情報処理学会研究会資料記号処理 12-16 (1980).
52) Masinter, L. M. and Deutsch, L. P.: Local

Optimization in a Compiler for Stack-based Lisp Machines, Conf. Record of 1980 LISP Conference, Stanford, pp. 223-230 (1980).
53) Deutsch, L. P.: ByteLisp and its Alto Implementation, Conf. Record of 1980 LISP Conference, Stanford, pp. 231-242 (1980).
54) Burton, R. R., et al.: Overview and Status of DoradoLisp, Conf. Record of 1980 LISP Conference, Stanford, pp. 243-247 (1980).
55) Marshall, M.: LISP Computers go commercial, Electronics, Vol. 53, No. 25, pp. 89-90 (1980).
56) 雨宮真人，長谷川隆三，三上博英：リスト処理向きデータフローマシンの検討，情報処理学会研究会資料記号処理 13-3 (1980).
57) Sussman, G. J., et al.: Scheme-79-Lisp on a Chip, IEEE Computer, Vol. 14, No. 7, pp. 10-21 (1981).
58) Yamamoto, M.: A Survey of High-Level Language Machine in Japan, IEEE Computer, Vol. 14, No. 7, pp. 68-78 (1981).
59) Guzmán, A.: A Heterarchical Multi-Micro processor LISP Machine, Computer Architecture for Pattern Analysis and Image Database Management, pp. 309-317 (1981).
60) Clark, D. W., Lampson, B. W. and Pier, K. A.: The Memory System of a High-Performance Personal Computer, IEEE Trans. Comput., Vol. C-30, No. 10, pp. 715-733 (1981).
61) Waller, L.: Lisp language gets special machine, Electronics, Vol. 54, No. 17, pp. 40-41 (1981).
62) 小長谷明彦，山本昌弘：リダクションマシンの構想について，電子通信学会技報 EC 81-33 (1981).
63) 箱崎勝也，山本昌弘：高級言語マシンの実際，産報出版，p. 174 (1981).
64) 伊藤貴康，庄内 亨，岸本光弘：マイクロプログラミング方式による LISP 処理系とその評価（I），電子通信学会技報 EC 81-68 (1981).
65) Greenfeld, N. R.: JERICHO: A Professional's Personal Computer System, Proc. of 8th Symposium on Computer Architecture, pp. 218-226 (1981).
66) LMI LISP Machine Specification, LMI (1981).
67) The 1100 Scientific Information Processor Fact Sheet, Xerox (1981).
68) LMI Lambda Machine Overview, LMI (1982).
69) 前川博俊他：試作 EVLIS マシンの EVALII と開発支援機能，情報処理学会研究会資料記号処理 17-1 (1982).
70) 佐藤 衛，井田昌之，間野浩太郎：会話型人工知能専用機 ALPS/II のLisp 処理機構，情報処理学会研究会資料記号処理 17-2 (1982).
71) 大里延康，渡辺和文：Lisp マシン ELIS の開発環境，情報処理学会研究会資料記号処理 17-3 (1982).
72) 理研シンポジウム：リスプ・マシン報告集，理化学研究所，p. 42 (1982).
73) 安井 裕，斎藤年史：EVLIS マシンの試作について，理化学研究所，理研シンポジウム：リスプ・マシン報告集，pp. 29-32 (1982).
74) 服部 彰，篠木 剛，品川明雄，林 弘：高速リスト処理に適したアーキテクチャについて，情報処理学会研究会資料記号処理 18-9 (1982).
75) 井田哲雄：LISP と LISP マシン：その現状と展望，bit，Vol. 14，8月号 (1982).

（昭和57年6月14日受付）