

# 並列処理機能を有するLispマシンの開発

飯田三郎 井戸隆明 中村博文

野中 修 北橋忠彦 橋 南信

豊橋技術科学大学

## 1. はじめに

関数型言語Lispは、従来から人工知能、知識工学等の研究に不可欠の言語である。だが、近年その適用領域は益々拡大しつつあり、Lisp処理系の高速度性・柔軟性の追求は、計算機ハードウェア・ソフトウェアの両面において重要な研究課題である。我々はLisp処理系の高速度性の観点から並列処理型Lispマシンの開発をすすめてきたが、現在、ハードウェアの稼働状態に入り、ソフトウェアによる試計算を行っている。ここでは、このLispマシンの現状について報告する。

このLispマシンは、アーキテクチャ面において従来のフォームウェアによる構成法とは異なり、非/イラン型の構成法を採用している。構成法の概要を以下に述べる。

(1) 並列処理性を、関数に引数を作用させる際の引数の並列評価に求め、並列演算可能なハードウェアをイテラティブとした。一方、関数の起動に対しては、関数に必要なホストの引数の評価終了後に関数を起動するブロー起動方式の手法を導入した。(1)

(2) ハードウェアの実現においては、ハードウェアに要求される各種機能をモジュールに分割し、各モジュールの並列動作により前項の(1)を実現した。モジュール間は、バスにより結合し、バスは、各引数内の関数コードから成る1ビットを駆逐する。

(3) 各モジュールを構成する際には、モジュールの構成法を2通りから選択し、各モジュール内には、数段のバッチファを設け、パイプライン処理を可能とした。この結果、ハードウェアの構成は、マイクロプロセッサ或いはフォームウェアによる実現方法によらずとも、比較的容易にワイヤードロシユクによる実現が可能となった。

(4) Lisp特有のセルを回収するガーベジ・コレクションについては、並列処理型ガーベジ・コレクションアルゴリズムは、基本的には参照カウント法にもとづいて行われる。

(5) Lispプログラムのフロント・エンド・プロセッサ上には存在するコンパイラにより、このシステム固有のオブジェクト・プログラムに変換し、これをシステム内に格納する方式を採用した。関数の起動は、各引数エントリオブジェクト・プログラム内の各引数にコピーするこにより行なわれる。

## 2. システム構成

ハードウェアの構成の概略を図1に示す。図内の四角の枠は、前項のモジュールに対応しているが、機能の大きさはモジュールは、さらに小さく、各モジュールから構成される。これらのモジュールは、ZのバスA-BU

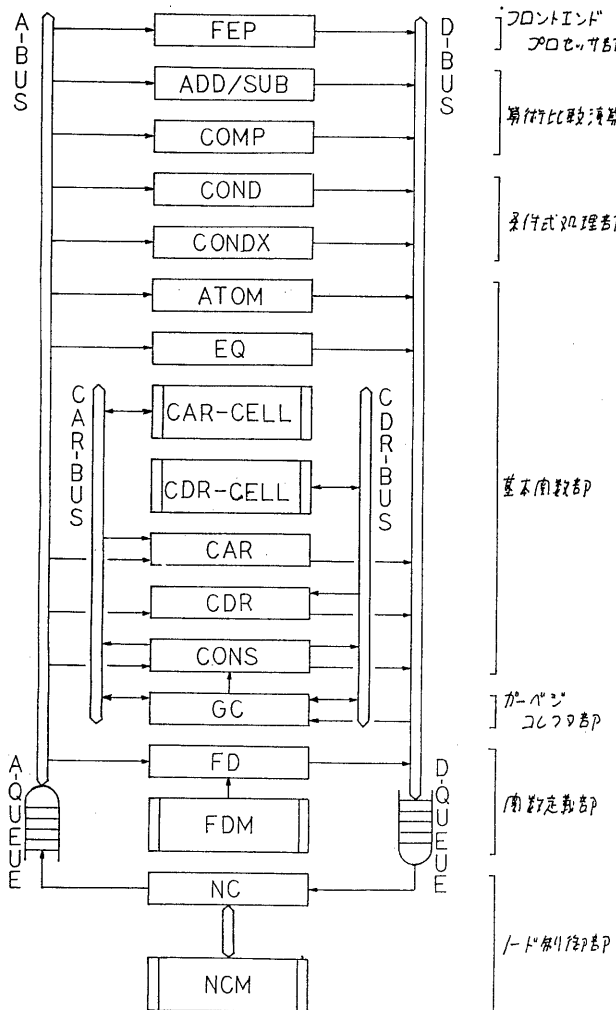


図1. 11ビットコンピュータの構成

S及ωD-BUSに接続されている。A-BUS及ωD-BUSは、10ビットの転送に用いられるが、そのうち128ビットのビット中を有し、このビット長の10ビットを1回転送することが可能である。図中の矢印は、データの転送方向を示す。

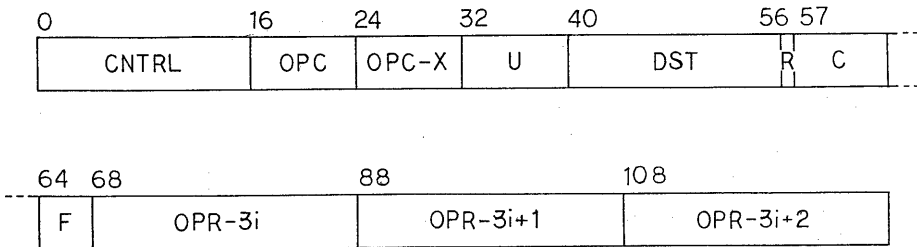
A-BUS及ωD-BUS上での10ビットの形式を図2に示す。2つの10ビット上の10ビットは、同一の形式で表

現される。  
 図3にプログラムのALのオブジェクト・プログラムを示す。コンパイラは、10ビットの形式に適合したオブジェクト・プログラムを生成するが、関数は一意にコード化され、OPC部で表現される。  
 以下、図1, 2, 3を用い、計算の流れの概略について述べる。詳細は、参考文献(2)を参照されたい。

(1)関数は引数を用いる際、オペラの引数の評価の終了した10ビットは、1-ドレトリックメモリNCMに貯えられているが、オペラの引数の評価の終了した10ビットは、1-ドレトリックNCにより、A-QUEUEへ送られる。この際、10ビットは、関数をOPCに、変引数をOPRに保持される。

(2)A-BUSに接続された各モジュールは、10ビットのOPC部をデコードし、該当モジュールは、この10ビットを取り込む。その後該当モジュールは、10ビットの情報をもとに、モジュールに課せられた機能

を行う。  
 (2-1) 10ビットが図2の基本演算部或いは算術比較演算部で取り込まれた場合、入力10ビットのOPR部を変引数とし、該当演算(car, cdr, cons等)を行い、演算結果を10ビットのOPR部に乗せωD-BUSへ出力する。この際、入力10ビットのDST及ωC部はそのままコピーし出力するとともに、



i=0,1,2,...

②. ノット の 形式

LIST OF SOURCE PROGRAM

```
(DE EQUAL (X Y)
  (COND ( (ATOM X) (EQ X Y) )
        ( (ATOM Y) NIL )
        ( (EQUAL (CAR X) (CAR Y))
          (EQUAL (CDR X) (CDR Y)) )
        ( T NIL ) ) )
```

LIST OF OBJECT PROGRAM

< EQUAL >

	OPC	OPCX	U	DST	R	C	F	OPR0	OPR1
GC								#OPR0(1)	
COND	00		1		0		0	#OPR0	#OPR1
CONDX			1	1	0	F	0		
ATOM			0	0	0	OPR0	0	#OPR0	
EQ			0		0		0	#OPR0	#OPR1
GC									#OPR1(1)
COND	02		1		0		0	#OPR0	#OPR1
CONDX			1	1	0	F	0		
ATOM			0	0	0	OPR0	0	#OPR1	
GC								#OPR0(-1)	#OPR1(-1)
NOP			0		1		0	NIL	
GC								#OPR0(1)	#OPR1(1)
COND	04		1		0		0	#OPR0	#OPR1
CONDX			1	1	0	F	0		
EQUAL			2	1	0	OPR0	0		
CAR			0	0	0	OPR0	0	#OPR0	
CAR			0	0	0	OPR1	0	#OPR1	
EQUAL			2		0		0		
CDR			0	0	0	OPR0	0	#OPR0	
CDR			0	0	0	OPR1	0	#OPR1	
GC								#OPR0(-1)	#OPR1(-1)
NOP			0		1		0	NIL	

③. アドレス フォーム

R部のビットを立てる。

(2-2) パケットが図2の関数定義部で取り込まれた場合。関数定義部は、オプジェクト・プログラムを保持するメモリ-FDM及び制御部FDより構成される。関数定義部が関数を起動するためには、入力パケットのOPC及びOPC-X部を使用する。OPC-Xは、条件式の述部及び値部に対し順番にコンパイラがくりあげた一連番号であり、OPC-Xにより実行すべき述部或いは値部が定めらる。関数定義部は、入力パケットのOPC及びOPC-Xをキーとして、オプジェクト・プログラムの実行すべき述部或いは値部を関数定義メモリより順次読み出し、入力パケットのOPR部に保持される実行引数をオプジェクト・プログラム内の所引数(図3では井で示される)にコピーし、D-BUSへパケットを送出する。実行数のコピー数は、パケットのOPC部をGCとしてカーベジ・コレクタへ通知される。

(2-3) パケットが図2の条件式処理部で取り込まれた場合。条件式を取り扱うためコンパイラは、条件式の述部のオプジェクト・プログラムに対し、乗分に2つのパケットを付加する。これらのパケットのOPC部はCONDX及びCONDである。CONDXは、条件式述部の評価結果である真偽を実引数として送り、これをパケットのF部の1ビットに及び、CONDへ伝える。CONDは、条件式の評価に必要な実行数をすべて保持すると共に、A-BUSから取り込んだパケットのF部の1ビットを参照し、真の場合は $OPC-X \leftarrow OPC-X + 1$ とし、実行数を条件式述部に伝える。偽の場合は、 $OPC-X \leftarrow OPC-X + 2$ とし実行数を次の条件式述部に伝える。

(3) D-BUSへ送出手続きのパケットは、D-QUEUEを經由して、ノード

制御に取り込まれる。D-QUEUEは、実行引数や積数のパケットから構成される場合の整合を行う。

(4) 関数に引数を伴う場合には、必要なすべての引数の評価を行なわなければならないが、個々の引数の評価終了は、断片的に一定ではない。この引数の評価の未終了の状態を、パケットとして保持する必要が生じることが、これらのパケットの相互関係は、本構造として表現される。この未終了の状態は、実行引数の未確定状態のパケットに対応し、アークは、評価終了後の実行引数の格納位置を示すものと考えられる。図2のパケットの形式に於いて、未確定引数の数はU部で、アークはDST及びC部で表現される。ノード制御は、上記のパケットを保持するメモリNCMと制御部NCより構成される。ノード制御の機能は、次のとおりである。

(4-1) 取り込んだパケットのR部が立っている場合は、このパケットのDSTをもとにNCMからノードを呼び出す。その後、パケットのOPRに存在する実行引数を同じパケットのC部の指示に従い、ノードと結合する。ノードのU部に対し $U \leftarrow U - 1$ の演算を行い $U = 0$ ならばパケットとしてA-QUEUEへ送出手続き。Uが0ならば、未確定の引数があるためもとNCMへ格納する。

(4-2) 取り込んだパケットが関数定義部の出力である場合は、コンパイラが本構造を保持しているため、ノード制御はD-QUEUEから順次パケットを取り出し、次の制御を行う。Uが0のパケットは、NCMの未確定領域に格納する。U=0のパケットは、A-QUEUEへ送出手続き。未確定の状態に於いてのパケットに対しては、Uの値の如何にかかわらず、パケットの

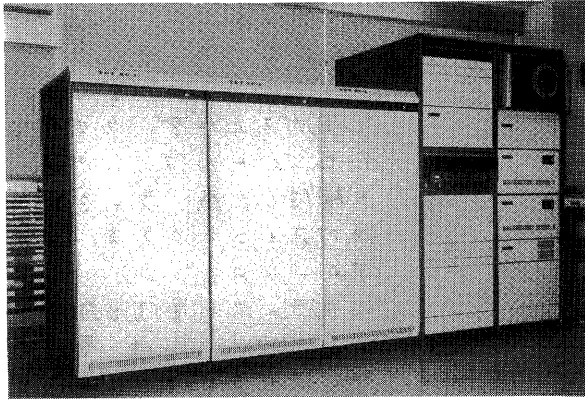


写真1

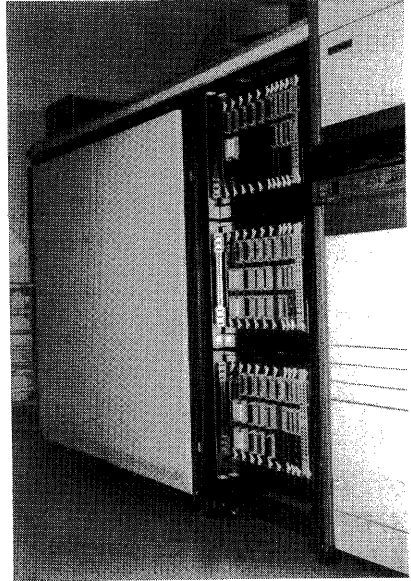


写真2

DSTとNCM上の格納位置をともに、本構造をNCM上で再構成する。

以上の計算の流れの概略であるが、その他に

(5)ガーベジ・コレクション部は、上記の計算と並行してガーベジ・セルの回収を行う。回収のアルゴリズムは、基本的に、参照カウント法に基づいているが、カウントを増減するための情報には、D-BUSより入手する。そのため、基本関数部は、入力した変引数とコピーしたD-BUS上へ出力するとともに、関数定数部は、変引数のコピー数をガーベジ・コレクション部へ通知する。

(6)フロント・エンド・フロセ、サ部は、計算の開始時に該当関数を起動し、終了時に返す受け取る。

### 3.11-ドウェア

ハードウェアの製作にあたり、基本的には我々の有る技術の範囲内で、実現可能な方式を採用することをとした。

#### (1)素子

使用素子は、使用経験及び入手の容易さを考慮して、論理素子としては74LSシリーズ相当品のTTL、記憶素子としては64KDRAMを用いた。

#### (2)実装

基板は、約20×30cm<sup>2</sup>の大27のものを使用した。この上には、16ピンDIP接ぎで、96個のICを搭載可能である。この基板は、前面に40ピット、7-ポルを6本まで接続可能であり、後面では80ピンコネクタ2個により、160本の布線が可能である。この4の基板は、幅60cm、奥行80cm、高さ120cmの筐体3台に収納される(写真1)。筐体は、上・中・下段の3つのシャーシで構成され(写真2)、1つのシャーシには、9枚の基板が収納可能である。シャーシの収納にあたり、2は、1つのシャーシに3枚の基板を構成し、この4の3枚を、上・中・下段に配置した。

#### (3)電源

電源は、各筐体内に設置された+5Vの単一電源を供給する。基板への電源は、基板の後面のピンを8ピンコネクタに、それぞれ+5V及びGNDに接

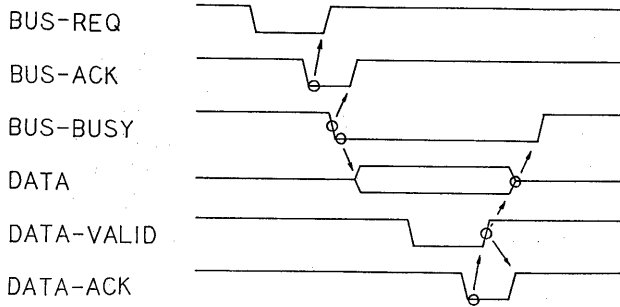


図4. データの転送

既し、供給される。

(4) 布線

基板内の布線は、ラッピングにより行い、上のモジュール内の基板内の布線には、フロッタ・ケーブルを用いた。

(5) バス

バスと基板との接続には、基板の後面のピンを用いられる。従って、バスは、筐体の裏面に布線される。筐体内のバスの経路には、布線のツイステッド・ペア線より成るケーブルを用いた。A-BUSとD-BUSの総延長は約6mであり、バスの両終端にはツイステッドペア線の整合を行う。

(6) データの転送

データの転送は、バスの総延長を考慮し、バス・シフト方式による非同期転送により、行なわれる。(図4)

モジュールが、バスを用いてデータを転送する際には、次の手順に従う。

(6-1) モジュールは、バス・アドレスに対し、バスの使用許可を要求する。(BUS-REQ)

(6-2) バス・アドレスは、バスの未使用状態を確認した後、バス使用の要求を出さるモジュールに対し、バスの使用許可を与える。(BUS-ACK)

(6-3) 使用許可権を獲得したモジュール

は、バスをビジー状態にした後(BUS-BUSY)、データをバス上に乗せる。

(6-4) バス上にデータを乗せたモジュールは、データを補償した後、データが有効であることを示す。(DATA-VALID)

(6-5) データの受信を指示されたモジュールは、受信可能状態であることを示す。データを取り込み、受信完了を送信側に知らせる。(DATA-ACK)

ACK)

(6-6) DATA-ACKにより、受信完了を示す。モジュールは、DATA-VALID線、DATA線、BUS-BUSY線の順序で、この状態を初期状態に戻す。

(7) エラール出

データには、8ビットごとに、11011ビットを付加した。エラール出は、バス上のデータに付し行なわれる。エラールが発生した場合、このエラール回路は、FERRに通知するとともに、エラールを表示する。

(8) モジュールの作成

モジュールを作成する際には、各モジュールに共通する機能に対して同一の作成手法を採用して、システム全体を見過しよく構成する方針をとった。この結果、各モジュールのバッドアンプ構成法、制御回路等、かなり共通化された。

(9) ドキュメンテーション

回路の修正、後の保守のために、ドキュメンテーションは、重要な問題である。モジュールに対し、

- ・モジュールの機能図
  - ・ワイヤング・チャート
  - ・基板のピン接続図
  - ・IC配置図
  - ・回路台図
- 等を作成した。

## 4. ヴィクトリア

このシステムにおいて、Lisp の諸機能は、ハードウェアにより実現されるため、ソフトウェアの担い役割は、非常に軽減される。以下に述べるソフトウェアは、FEP 上に存在してプログラムの実行を支援する。

### (1) コンパイラ

Lisp のプログラムを、このシステム固有のオプジェクト・プログラムに変換する。変換の例は、図 3 に示した。

### (2) ローダー

オプジェクト・プログラムを、システム内の固数定義メモリ FDM にロードする機能も有する。

### (3) 形式の入出力プログラム

プログラムの実行時に、形式の入出力を行う。

又、このシステムにおいて、FEP 上からハードウェアの診断が可能であり。

### (4) ハードウェア診断プログラム

も開発された。上記 4 種類のプログラムは、ハードウェアのテストに際して、ソフトウェアの作成されたが、今後、これを一つのシステム・ソフトウェアとしてまとめたい予定である。

一方、ガーベジ・コレクションのソフトウェアは、ガーベジ・コレクションのアルゴリズムとその実装を研究の対象として、開発を行う予定である。

## 5. まとめ

以上、我が国が現在開発中の並列処理型 Lisp マシンの概要について報告した。現状は、ハードウェアが実装可能となり、ソフトウェアによる試計算を行っているが、今後は、

### (1) ハードウェアのチューニング

を行う。

(2) 現システムの動作特性を解析し、より高速処理可能なシステムに拡張することを目指す。

(3) イマン型計算機上の Lisp 処理系と比較し、その得失を明確に、

(4) ガーベジ・コレクションのアルゴリズムを、ガーベジ・コレクション上のインパリティとして、比較・検討を行う予定である。

最後に、Dijkstra の提率した階層進(?)による自然対数の基底計算のアルゴリズム<sup>(3)</sup>は、このシステムによる計算結果の一部を APPENDIX に示す。なおこの図に誤りがあるが、これは、我々のハードウェアのバグであることである。おそれなくとも願うた。

## [参考文献]

- (1) 両宮「データフロー・パーキテック」に「2」コンピュータ・ソフトウェア、Vol. 1, No. 1, p. 42
- (2) 信学技報, Vol. 83, No. 283, EC83-51
- (3) 数学セミナー, Vol. 21, No. 4, p. 85

APPENDIX

```

(DE PRINT E ()
  (PROG (A)
    (SETQ A (CONS 1 NIL))
    (RPLACD A A)
    (PRINI "e=2.")
  LOOP
    (SETQ A (F A 2))
    (PRINI (CAR A))
    (SETQ A (CDR A))
    (GO LOOP)
  ))
(DE F (L N)
  (PROG (X Y Z)
    (SETQ X (CAR L))
    (RETURN
      (COND ((ATOM X)
        (COND ((EQ (QUOTIENT (TIMES X 10) N)
          (QUOTIENT (PLUS (TIMES X 10) 9) N))
          (CONS (QUOTIENT (TIMES X 10) N) (CONS L NIL)) )
        (T
          (SETQ Y (F (CDR L) (ADD1 N)))
          (CONS (QUOTIENT (PLUS (TIMES X 10) (CAR Y)) N)
            (CONS (REMAINDER (PLUS (TIMES X 10) (CAR Y)) N)
              (CDR Y))) ) ) )
      (T
        (SETQ Z (F (CDR X) (ADD1 N)))
        (F (CONS (REMAINDER (PLUS (TIMES (CAR X) 10) (CAR Z)) N)
          (CDR Z))
          N) ) )
    )))

```

e=2.71828182845904523536028747135266249775724709369995	50
95749669676277240766303535475945713821785251664274	100
27466391932003059921817413596629043572900334295260	150
59563073813232862794349076323382988075319525101901	200
157383418793070215400891499348841675092447614606680	250
82264800168477411853742345442437107539077744992069	300
55170276183860626133138458300075204493382656029760	350
67371132007093287091274437470472306969772093101416	400
92836819025515108657463772111252389784425056953696	450
77078544996996794686445490598793163688923009879312	500
77361782154249992295763514822082698951936680331825	550
28869398496465105820939239829488793320362509443117	600
30123819706841614039701983767932068328237646480429	650
53118023287825098194558153017567173613320698112509	700
96181881593041690351598888519345807273866738589422	750
87922849989208680582574927961048419844436346324496	800
84875602336248270419786232090021609902353043699418	850
49146314093431738143640546253152096183690888707016	900
76839642437814059271456354906130310720851038375051	950
01157477041718986106873969655212671546889570350354	1000
02123407849819334321068170121005627880235193033224	1050
74501585390473041995777709350366041699732972508868	1100
76966403555707162268447162560798826517871341951246	1150
65201030592123667719432527867539855894489697096409	1200
75459185695638023637016211204774272283648961342251	1250
64450781824423529486363721417402388934412479635743	1300
70263755294448337998016125492278509257782562092622	1350
64832627793338656648162772516401910590049164499828	1400
93150566047258027786318641551956532442586982946959	1450
30801915298721172556347546396447910145904090586298	1500