オペレーティング・システム 40 － 3
（１９８８. ９. ９）

二進木計算機ＣＯＲＡＬ６８Ｋの プログラム実行制御方式

高橋 義造
（徳島大学工学部）

松尾 賢二
（日本ＩＢＭ）

我々は大規模計算を高速に行うために二進木結合並列計算機Ｃｏｒａｌ68Kを開発し，基本的なプログラミング開発環境を整備した．この計算機は63台のプロセッサ要素をＤＭＡ回路によって二進木状に結合し，これをホスト計算機となるワークステーションに接続した疎結合の並列計算機である．二進木計算機は階層構造であることを特長としており，分割統治型の並列処理に有効である．Ｃｏｒａｌ６８Ｋではメッセージによるプログラム実行制御方式を採用した．この方法によれば各プロセッサはホスト計算機から次々に送られるシステムメッセージを受け取ってそに書かれているコマンドを処理することによって処理が行われるようになっている．この処理方式をサポートするためのプログラミング方式や初期化プログラムの機能などについて述べ，更にこの方法を推し進めたメッセージ駆動式のオペレーティングシステムを提案する．

Program Execution Control of Binary-Tree Parallel Computer Coral 68K

Yoshizo Takahashi
University of Tokushima
2-1 Minami jyousanjima-cho,
Tokushima 770, Japan

Kenji Matuo
IBM Japan
1623-14 Shimoturuma,
Yamato-shi 242, Japan

In order to practice parallel computing of large-scale problems, we have recently developed a binary-tree parallel computer Coral 68K which is constructed from 63 processor elements interconnected in a binary tree with DMA channels, and is connected to a workstation which serves as a host computer.

The hierarchical structure of the binary-tree machine fits the parallel processing by the divide-and-conquer method. The the program execution on the Coral 68K is controlled by the message. In this method the parallel procesing is performed by the processing elements which receive system messages broadcasted from the host computer and act in a way as described in the commands written in the message. In this paper the programming method and the initializing program which support this program execution control are described, and also the concept of the message-driven operating system developed from this method is proposed.

1．はじめに

比較的小型のプロセッサ要素を多数使用することによって大規模な計算を高速に処理することを目的とする並列計算機の研究が各方面で行われており，各種の方式が提案されると共に，実際にプロトタイプが試作され，並列アルゴリズムの開発に利用されたり，実規模の並列計算機を開発するための設計資料を得る目的に使用されている．我々は高並列度システムの適応する疎結合並列計算機について検討した結果，超ＬＳＩ化に適した二進木結合方式に着目し，先に小規模の実験機Ｃｏｒａｌ’８３を開発し並列アルゴリズムの研究に使用してきた[1]．その経験からこの方式の並列計算機が実際の問題の処理に有効であると確信し，実際の問題に使用できる実規模の二進木結合並列計算機Ｃｏｒａ１６８Ｋの開発に取り掛かり，昭和６２年６月に完成した．[2]

二進木計算機は階層構造を特長とし，探索問題など分割統治型の問題処理に有利である．しかし疎結合計算機では密結合計算機と違ってプロセッサ間通信に手間がかかり，プログラムの実行制御が難しい．この論文ではＣｏｒａ１６８Ｋで用いた並列処理プログラムの実行制御方式について述べる．そのために先ず二進木計算機の特性を簡単に説明し，Ｃｏｒａ１６８Ｋのハードウェアの構成について述べた後，Ｃｏｒａ１６８Ｋのプログラミング方式，これをサポートするための初期化プログラム，システムメッセージによる制御方式，プロセッサ間通信方式などについて説明する．さらにこの方式を発展させたメッセージ駆動方式のオペレーティングシステムの提案を行う．

2．二進木計算機の特性

疎結合並列計算機の代表的な結合方式にはトーラス結合，ハイパーキューブ結合，二進木結合がある．この３種類の結合方式の特性を平均接続数，平均通信距離，平均放送距離で示すと表１のようになる．[3]

表１ 疎結合並列計算機の結合方式の性能比較

結合方式	接続数	平均通信距離	放送距離
ﾄｰﾗｽ	4	$\sqrt{N}/2$	$\sqrt{N}$
二進木	3	$2\log_2 N$	$2\log_2 N$
超立方体	$\log_2 N$	$\log_2 N/2$	$\log_2 N$

二進木結合の特長として次の事項があげられる．

（１） 再帰的構造であり，構成が簡単である．

（２） 階層構造であるので探索問題，分割統治法による並列処理に適している．

（３） 平均および最大接続数が小さいのでプロセッサ要素の構造が簡単である．

（４） 平均通信距離及び平均放送距離が小さい．

（５） 接続が平面的であり，ＷＳＩ（ウェハー・スケール・インテグレーション）向きである．

これらの中で(3)(4)の性能はハイパーキューブに次ぐものであるが、(5)の特長はそれにはない特長である．

また一方では二進木結合の欠点として次の事項が指摘されている．

（１） プロセッサ要素によって通信負荷にアンバランスがある．

（２） 任意のプロセッサ要素間に複数の通信路がないので信頼性が劣る．

通信負荷のアンバランスは問題の分割法と割当法を選ぶことによって回避できる．また信頼性の問題は次のように考えると重要な欠点とは考えなくても良いことが分かる．すなわち根の部分が故障するとシステム全体が使用できなくなるが，葉の部分だとそのプロセッサ要素が使用できないだけである．Ｎ台のプロセッサ要素の内の１台が故障したとして，それが葉である確率は１／２であるが，根である確率は１／Ｎに過ぎない．従って１台のプロセッサ要素が確率ｆで故障したときに使用できなくなるプロセッサ要素の数の期待値はｆ・$\log_2 N$ と計算され，さほど大きいものではないことが解る．[4]

3．Ｃｏｒａ１６８Ｋの構成

Ｃｏｒａ１６８Ｋは図１に示すように二進木状に結合された６３台のプロセッサ要素と，その根に接続されたホスト計算機より構成されている．ホスト計算機は１６ビットのワークステーションで，プロセッサ要素で実行するプログラムの作成と管理，コンパイル，ローディングを行うほか，プロセッサ要素で実行中のプログラムの入出力サービス，ファイル管理，デバグなどを行う．ホスト計算機と木の根に当たるプロセッサ要素（以後ルートプロセッサと呼ぶ）の間は１６ビットの並列チャンネルによって接続されている．

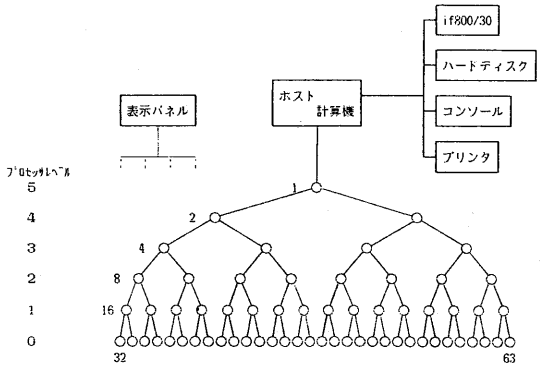

図1　Ｃｏｒａｌ６８Ｋの構成図

６３台のプロセッサ要素は相互にＤＭＡ方式による１６ビットのチャンネルで二進木状に結合し，各プロセッサ要素からは筐体表面の表示パネルと各シャーシに付属したテストパネルの制御信号を引き出している．表示パネルは実行中の各プロセッサ要素の動作状態をプログラマが直接監視して並列プログラムのデバッグや，動的な負荷のバランスの模様などを評価できるようにするために設けたものである．これは筐体の表面に二進木状に配置された合計126個のＬＥＤ（発光ダイオード）から構成され，各プロセッサ要素の動作状態およびプロセッサ要素相互間で行われている通信とその方向通信が一目で見えるようにした．ＬＥＤには各プロセッサ要素に対応して１個づつの赤色のものと，各接続回路毎に１個づつの２色（緑・橙）のものがある．赤のＬＥＤはプロセッサ要素からの命令によって駆動され，プログラムの実行状態を表示するのに使う．また２色ＬＥＤはプロセッサ要素間通信回路によって直接駆動され，ｔｏｐ方向から受信が行われる時には緑が，ｔｏｐ方向に送信が行われる時には橙色が一定時間点灯する．

Ｃｏｒａｌ６８Ｋの構成要素と仕様を表２にまとめて示す．

各プロセッサ要素にはｔｏｐ，ｌｅｆｔ，ｒｉｇｈｔの３方向に対してそれぞれ独立した接続ポートがある．各接続ポートにはステータス回路，割り込み回路，およびデータ転送回路がある．ステータス回路には３ビットのステータス信号のための出力および入力信号線があり，それぞれ相手のプロセッサ要素のステータス回路の入力信号線と出力信号線に交差して接続される．

表２　Ｃｏｒａｌ６８Ｋの仕様

要素	項目	仕様
ﾎｽﾄ計算機	ﾓﾃﾞﾙ	ｼｬｰﾌﾟIX-5
	CPU	MC68000(10MHZ)
	ﾒﾓﾘ	2MB
	ﾃﾞｨｽｸ	67MB
	OS	UNIX SYSTEM V
ﾌﾟﾛｾｯｻ要素	台数	63
	CPU	MC68000(10MHZ)
	ROM	16KB
	RAM	512KB
ﾃﾞｰﾀ転送速度	ﾎｽﾄ-PE間	37KB/秒
	PE-PE間	2.0MB/秒

ステータス信号はオペレータに各プロセッサ要素の状態を知らせるためと，隣どうしのプロセッサ要素間で割り込み原因の通知，データの要求信号などの短い情報を直接伝達するために使用する．

データ転送回路にはＤＭＡ方式を採用した．今回開発したデータ転送回路は１６本の双方向データ線と５本の制御線で構成する．３方向の接続ポートのデータ転送回路は１個のＤＭＡコントローラMC68450によって制御し，バーストモードで１６ビットデータの転送行う．このモードのデータ転送中はバスが占有されるので３方向同時にデータ転送を行うことはできないが２ＭＢ／秒の高速転送が行われる．送受信の方向はバス方向信号によって決定されるが，この信号はｌｅｆｔ，ｒｉｇｈｔの方向には出力信号，ｔｏｐ方向には入力信号となっている．従ってｔｏｐ方向のプロセッサ要素が常にバス方向の決定の主導権をとるのでバス調停は不要になる．一方が受信のとき相手側が送信するというようにプロセッサ間通信を協調的に行うようにプログラムを書けば通信方向がぶつかる恐れはない．そうでない場合，例えば相手側が受信していない時に送信したいような時には，割込み回路を使って相手側に割り込みをかけステータス信号で受信要求を知らせるようにする．

４．　Ｃｏｒａｌ６８Ｋのプログラミング方式

４．１　並列処理方式

ある問題を並列処理する方式には機能分割，デ

ータ分割，処理分割，などの方式があるが，問題の性質と使用する並列計算機のアーキテクチャに適した方式を選択しなければならない．二進木計算機は階層構造を持つことを特長としており，このことから分割統治型の処理分割を行うのに適している．また二進木計算機はトップダウンまたはボトムアップのトリーパイプラインとしても使用できるからパイプラインによる処理分割にも適しているということができる．

分割統治型の並列処理はソーティングの場合に適当である．二進木計算機でソーティングを行う方法を図２に示す．すなわちホスト計算機から与えられたデータをルートプロセッサが２等分し，左右のノードプロセッサに送り，ノードプロセッサはこれを２等分して下に送る．そしてリーフプロセッサに届いたデータはそれぞれ並列にソートされ，その結果は上のノードプロセッサに送られ，各ノードプロセッサは左右から送られたデータをマージし結果をさらに上のプロセッサに送る．その結果最後にホスト計算機にソートされたデータがえられることになる．

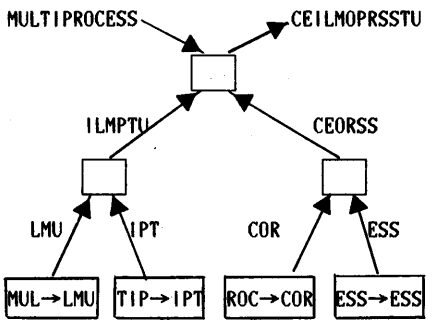

図２　二進木計算機によるソーティング

通常のパイプライン処理には１次元に配置されたプロセッサ要素を用いるが，二進木状に結合されたプロセッサ要素のルートプロセッサからデータを入力して処理の一部を行って左右のノードに送り，リーフプロセッサから最終の計算結果を取り出すトップダウンパイプラインや，リーフプロセッサからデータを入力してルートプロセッサから最終結果をとりだすボトムアップパイプラインの処理方式が有効な場合がある．素数計算では図３に示すようにリーフプロセッサが発生した自然数の列を上のノードに送り，ノードプロセッサがこれを篩いにかけてさらに上に送るようにするボトムアップ式のパイプライン処理が考えられる．

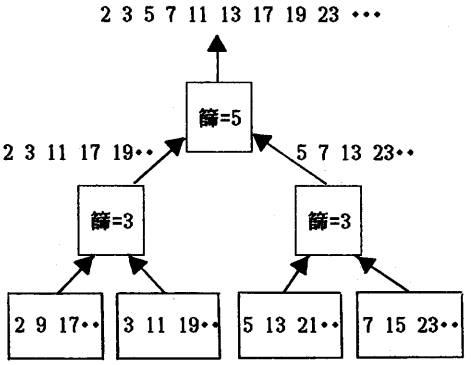

図３　ボトムアップパイプラインによる素数計算

Ｃｏｒａ１６８Ｋでは各プロセッサは上から送られるデータを処理しその結果を左右のプロセッサ要素に送って残りの処理をさせるのと，左右から送られる結果を集計して上に送るのを基本動作としている．したがってＣｏｒａ１６８Ｋの各プロセッサ要素のプログラムはできるだけこの３方向の通信だけで全ての処理ができるように工夫して書くようにする．例えば同じ情報を全てのプロセッサ要素に伝える放送はルートプロセッサが左右に送った情報を他のプロセッサ要素が受け取ってさらに左右に送るようにする．

４．２　プログラミング環境

Ｃｏｒａ１６８Ｋではプログラムを開発するためにホスト計算機のＵＮＩＸ－Ｓｙｓｔｅｍ Ｖのソフトウェア開発環境をそのまま利用する外，二進木計算機の機能を活用するためのサブルーチンを用意し，また初期化プログラムとメッセージプロセッサをＲＯＭに書き込んで使用するようにした．

Ｃｏｒａ１６８Ｋを使用するにはホスト計算機用と，プロセッサ要素用の２種類のプログラムを作る必要がある．プロセッサ要素ごとに違うプログラムを作ることもできるが，通常は同じプログラムのコピーを全プロセッサ要素にロードし，実行時にプログラムの中でプロセッサ番号などを調べて各要素で違う処理をさせるようにしている．こうするとメモリは多少無駄になるがプログラムのデバグが容易になる利点がある．どちらのプロ

グラムもＣ言語で書き，ホスト計算機でコンパイルする．ホスト計算機用のプログラムには表３に示すようなランタイムサブルーチンを使用するが，この中でｌｏａｄルーチンは全プロセッサ要素に同じプログラムを指定した番地からロードするためのものであり，ｇｏルーチンは全プロセッサ要素が指定した番地から実行することを指令するためのものである．特定のプロセッサ要素にだけプログラムをロードしたり，実行させるには６に述べるシステムメッセージを使用する．

表３　ホスト計算機用ランタイムサブルーチン

ルーチン名	機能
crlopen()	ﾌﾟﾛｾｯｻ要素のｵｰﾌﾟﾝ
crlclose()	〃　　ｸﾛｰｽﾞ
crlputh(var)	2ﾊﾞｲﾄﾃﾞｰﾀの送信
crlgeth(&var)	〃　　受信
crlputi(var)	4ﾊﾞｲﾄﾃﾞｰﾀの送信
crlgeti(&var)	〃　　受信
crlputbk(&var,size)	ﾌﾞﾛｯｸﾃﾞｰﾀの送信
crlgetbk(&var,size)	〃　　受信
load(filename,adr)	ﾌﾟﾛｸﾞﾗﾑのﾛｰﾄﾞ
go(adr)	ﾌﾟﾛｸﾞﾗﾑの実行

プロセッサ要素のメモリのアドレス空間を図４のように割り当てた．先頭の１６ＫバイトはＲＯＭエリアで，0x10000番地からの５１２ＫバイトがＲＡＭエリアになる．ＲＯＭには割り込み処理プログラムやメッセージプロセッサが書き込まれており，これがＲＡＭエリアの先頭の８Ｋバイトにあるシステムエリアを管理する．ホスト計算機からロードするプログラムは 0x12000番地以上のユーザエリアに格納する．したがってプログラムをコンパイルしてリンクする時にロードアドレスを指定する必要がある．0x100000以降の番地はＤＭＡコントローラやその他のデバイスのアドレスに使用する．

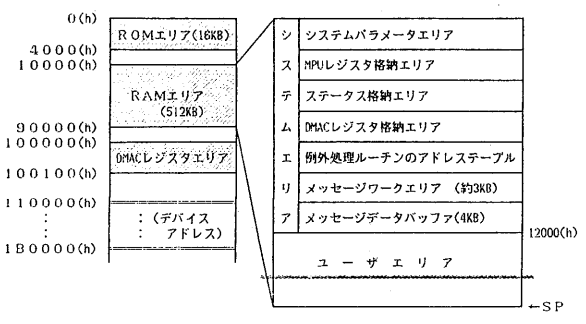

図４　プロセッサ要素のアドレス空間

プロセッサ要素用のプログラムのために，割り込み，ステータス入出力，データ転送のための表４に示すようなランタイムサブルーチンを用意した．ここで　ｄｉｒではｔｏｐ，ｌｅｆｔ，ｒｉｇｈｔ の方向を指定する．離れたプロセッサ要素に転送する場合には途中のプロセッサ要素で中継する必要があるが，これはユーザプログラムで処理するか，あるいは別に開発するＯＳを使用しなければならない．

表４　プロセッサ要素用ランタイムサブルーチン

ルーチン名	機能
intrpt(dir)	dir方向に割り込む
insts(dir)	dir方向のｽﾃｰﾀｽ入力
outsts(dir,data,cnt)	〃　ｽﾃｰﾀｽ出力
puth(var,dir)	dir方向の2ﾊﾞｲﾄ送信
geth(&var,dir)	〃　受信
puti(var,dir)	dir方向の4ﾊﾞｲﾄ送信
geti(&var,dir)	〃　受信
putbk(&var,size,dir)	dir方向のﾌﾞﾛｯｸ送信
getbk(&var,size,dir)	〃　受信

５．　初期化プログラム

Ｃｏｒａ１６８Ｋは６３台のプロセッサ要素によって完全木が構成できるようになっているが，必ずしも完全木である必要はない．あるプロセッサ要素が故障の時はそのサブトリーを除いた不完全木でも動作するように考えられている．プロセッサ要素の結合状態を調べてその構成を決めるのが初期化プログラムである．この初期化プログラムはメッセージプロセッサ，テストプログラム，割り込み処理プログラムなどと共に各プロセッサ要素のＲＯＭに書き込まれている．

各プロセッサ要素は電源投入またはリセット割り込みで先ずテストプログラムを実行し，メモリテストと接続ポートの診断を行い，各プロセッサ要素は３方向の接続ポートに他のプロセッサ要素が接続されているかどうかを調べる．次に初期化

プログラムがシステムパラメタを初期化し，それが終わるとメッセージプロセッサの実行に移る．

メッセージプロセッサはｔｏｐ方向よりシステムメッセージが送られてくるのを待ち，これを受け取るとその中に書いてあるコマンドに応じた処理を行い，処理が終了すると再びメッセージ待ちの状態になる．これを状態遷移図に表したのが図５である．またメッセージ処理中に割り込みスイッチが押されると処理を中断してメッセージ待ちになり，「継続」のシステムメッセージを受け取ると中断した所から処理を継続するようになっている．

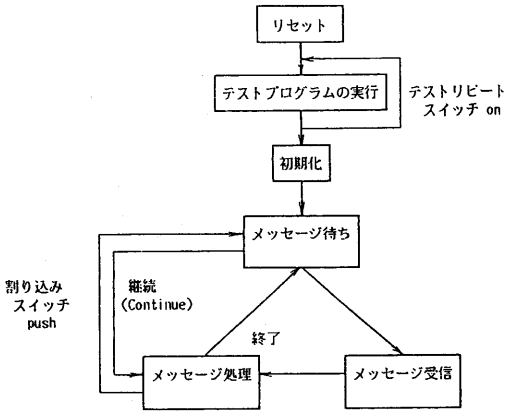

図５　プロセッサ要素の状態遷移図

初期化プログラムでは次のようなそれぞれ３２ビットのシステムパラメタを計算し，ＲＡＭのシステムパラメタエリアにかき込む．

*PNUM　:ﾌﾟﾛｾｯｻ番号
*HGT　:二進木の高さ
*PLVL　:ﾌﾟﾛｾｯｻﾚﾍﾞﾙ
*PDPTH　:ﾌﾟﾛｾｯｻの深さ
*LEFTN　:LEFT 方向に連結されているﾌﾟﾛｾｯｻ数
*RIGHTN　:RIGHT方向に連結されているﾌﾟﾛｾｯｻ数
*SUBN　:自分自身を含む部分木のﾌﾟﾛｾｯｻ数
*UPLIST,*LPLIST　:ﾌﾟﾛｾｯｻﾘｽﾄ

*UPLISTと*LPLISTは連結して６４ビットになり，１のビットの位置に対応するプロセッサ要素が接続されていることを表し，０の時は接続されていないことを表す．第０ビット（ＬＳＢ）が１の時には６３台の全てのプロセッサ要素が接続されていることを示す．

まずテストプログラムでは各プロセッサ要素は３方向のステータス出力を７にセットした後に各方向のステータス入力を調べ，これが７かどうかで，相手側のプロセッサ要素が接続されているかどうかを調べる．ｔｏｐ方向が接続されていないプロセッサ要素は自分をルートプロセッサと見なし＊ＰＮＵＭを１とし，ｌｅｆｔ方向に２を送り，ｒｉｇｈｔ方向に３を送る．ルートプロセッサ以外の全てのプロセッサはｔｏｐから送られる数値を自分の＊ＰＮＵＭに書き，それを２倍してｌｅｆｔに送り，さらに１を加えた値をｒｉｇｈｔに送る．このようにしてルートプロセッサに連結されているすべてのプロセッサ要素の番号が幅順(breadthfirst order)に設定される．また各プロセッサ要素はプロセッサ番号を２で割った値をプロセッサの深さとして＊ＰＤＰＴＨにかき込む．

ｌｅｆｔ方向が接続されていないプロセッサ要素では＊ＬＥＦＴＮを０にし，ｒｉｇｈｔ方向が接続されていないプロセッサ要素は＊ＲＩＧＨＴＮを０にする．その他のプロセッサ要素ではｌｅｆｔから送られる数を＊ＬＥＦＴＮに書き，ｒｉｇｈｔから送られる数を＊ＲＩＧＨＴＮに書く．そして＊ＬＥＦＴＮ＋＊ＲＩＧＨＴＮ＋１を計算して＊ＳＵＢＮに書きこの値をｔｏｐに送る．

＊ＳＵＢＮが１のプロセッサ要素（すなわちリーフプロセッサ）は自分の深さをｔｏｐ方向に報告する．その他のプロセッサ要素は左右から送られて来る深さを比較し，大きい方をｔｏｐに送る．最後にルートプロセッサが最大と判断した深さを二進木の高さであると判断しこの値を＊ＨＧＴに書き同時に他のプロセッサ要素に放送する．この値を受け取ったプロセッサ要素はこれを＊ＨＧＴに書くと同時に＊ＨＧＴ－＊ＰＤＰＴＨを計算し，＊ＰＬＶＬにかき込む．最後にルートプロセッサがリーフプロセッサからの報告をもとにプロセッサリストをつくり，各プロセッサ要素に放送する．

各プロセッサ要素には通常同じプログラムのコピーをローディングし実行するが，各プロセッサ要素ごとに異なる動作をさせるにはプログラム中でこのシステムパラメタを調べて分岐するようなプログラムを書く．例えばｔｏｐから送られたデータを全プログラムに放送する様なときには次のように＊ＬＥＦＴＮ，＊ＲＩＧＨＴＮを調べて転送するかどうかをきめる．

```
geti(&data,TOP);
```

```
if(*LEFTN) puti(data,LEFT);
if(*RIGHTN) puti(data,RIGHT);
```

６．システムメッセージによる実行制御

初期化が終わると，あるいは操作パネルの割り込みスイッチが押されると各プロセッサ要素はメッセージ待ちの状態になる．このときにはプロセッサ要素は表示パネルの赤のＬＥＤをつける．ｔｏｐ方向からシステムメッセージが送られるとこのＬＥＤを消し，宛先を調べて必要があればこれをｌｅｆｔ，ｒｉｇｈｔ方向にに中継した後，宛先に自分が含まれている場合にはメッセージを解釈して処理を行う．

システムメッセージは２０バイトの固定長のデータで図６のようにコマンド部，宛先部，アドレス部，サイズ部の４つの部分よりなっている．

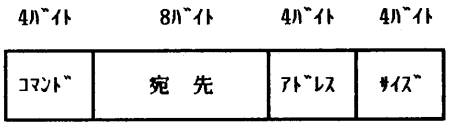

図６　システムメッセージの形式

コマンド部にはコマンドのコードが書き込まれる．コマンドには次の種類があり，その宛先部のプロセッサリストで指定されたプロセッサ要素ではそれぞれ以下に述べる処理を行う．

（１）初期化コマンド：先に述べた初期化と同じ処理，すなわちシステムパラメタの設定を行う．

（２）ロードコマンド：このシステムメッセージに続いてｔｏｐから送られてくる，サイズ部で指定されるサイズのプログラムをアドレス部で指定されるアドレスから始まるメモリ領域に書き込み，ｌｅｆｔ，ｒｉｇｈｔ方向にプロセッサ要素が接続されていればこれにプログラムを転送する．

（３）実行コマンド：アドレス部で指定されたアドレスからプログラムを実行する．

（４）メモリダンプコマンド：アドレス部とサイズ部で指定された範囲のメモリの内容をｔｏｐに送る．このデータは図５中に示した４ＫＢのメッセージデータバッファを使って中継されるので４ＫＢより大きい場合には何度かに分けて転送する．

（５）メモリライトコマンド：このメッセージに続いて送られてくるデータを指定された範囲のメモリ領域に書き込む．４ＫＢより大きい領域の場合は何回かに分けて転送する．

（６）レジスタダンプコマンド：ＭＰＵのレジスタをｔｏｐに送る．

（７）テストコマンド：リセット時に行うのと同じプロセッサ要素のテストを行う．

（８）継続コマンド：割り込みスイッチによって中断されたプログラムのＭＰＵのレジスタ，出力ステータス，ＤＭＡＣのレジスタをシステムエリアから回復し，プログラムを継続して実行する．

宛先部にはこのシステムメッセージをとどける宛先のプロセッサ要素がプロセッサリストの形式で書かれる．各プロセッサ要素はこのメッセージをｌｅｆｔ，またはｒｉｇｈｔのプロセッサ要素に中継するに当たってその方向に宛先のプロセッサ要素があることを確かめてから送るようにしなければならない．

全プロセッサに同じプログラムのコピーをローディングする場合にはホストから宛先のプロセッサリストを0x0000000000000001にしたロードコマンドに続いてプログラムを送ればよい．これはホスト用のランタイムルーチンｌｏａｄで行われる．プロセッサ要素ごとに違うプログラムをローディングするにはレベルの低いプロセッサ要素から順次ローディングを行うようにする必要がある．また全てのプロセッサ要素のローディングが終わってから実行コマンドを送るようにしなければならない．

プロセッサ要素はメッセージ処理の実行にはいるとそれを終了するか，あるいは割り込みスイッチで処理を中断されるまではシステムメッセージを受け付けないので，実行コマンドを送る時期に注意する必要がある．ホストからはやはりレベルの低いプロセッサ要素から順番に実行コマンドを送らなければならない．

７．プロセッサ間通信

二進木計算機ではプロセッサ要素間の通信経路が一意的に決まるのでメッセージを中継する場合のルーティングは簡単になる．プロセッサ番号ｐのプロセッサ要素が任意の方向から受け取ったメッセージの宛先をｑとすると中継方向ｄｉｒ（ｑ）は次の計算で求められる．

```
dir(q)=[q=p→own;
        q<p→top;
        q=2p→left;
```

```
    q=2p+1→right;
    t  →dir(q/2);
  ]
```

すなわち先ずｐとｑをくらべ，同じなら自分に宛てられたものであるから中継は不要であり，ｑ＜ｐならｔｏｐ方向に，ｑが２ｐならｌｅｆｔ方向に，ｑが２ｐ＋１ならｒｉｇｈｔ方向に中継する．それ以外の場合にはｑをｑ／２で置き換えて同じ計算をすればよい．なお放送の場合には中継方向の計算は不要で，メッセージが送られてきた方向以外の２方向に中継すればよい．

Ｃｏｒａｌ６８Ｋではプロセッサ要素間のデータ転送にＤＭＡ転送方式をとっているためデータ転送をブロック単位で行う必要がある．このために不定長のデータを送るには先ずデータサイズを送ってからデータ本体を送らなければならないので通信時間が余計にかかる．システムメッセージを２０バイトに固定したように，アプリケーションプログラムではできるだけ固定長のデータを用いるのが望ましい．いくつかの並列計算機に採用されているように，プロセッサ要素間をＦＩＦＯメモリで結合すれば通信速度はデータ長に無関係であるからこのような手間は不要であるが，ＦＩＦＯの記憶容量の制限から一度に転送できるデータ量が小さくなるという問題点が起こる．

８．メッセージ駆動式オペレーティングシステム

Ｃｏｒａｌ６８Ｋに現在インプリメントされているプログラムではメッセージプロセッサはシステムメッセージの処理中には新たなシステムメッセージを受け取れないようになっている．それはアプリケーションプログラムで交換するデータとシステムメッセージが区別できないからである．しかし割り込みによっていつでもシステムメッセージを受け取れるようにしたメッセージ駆動式オペレーティングシステムを考えることができる．すなわちプロセッサ要素間で交換するデータはすべてシステムメッセージの形式を取り，割り込み処理プログラムがメッセージを解読してその内容によって処理を行うか，あるいはアプリケーションプログラムに渡すようにするのである．

このためにはメッセージの様式を変更して発信者のプロセッサ番号とプロセス番号をかき込む送信元部を設け，また次の様なコマンドをシステムメッセージを追加する必要がある．

（９）ショートデータ転送コマンド：アドレス部で指定されるプロセスにサイズ部の４バイトデータを渡す．

（１０）ロングデータ転送コマンド：アドレス部で指定されるプロセスに，このメッセージに続いて送られるサイズ部で指定される長さのデータを渡す．

なお各プロセッサ要素で実行されるプロセス間でデータの交換をするには次のようなランタイムサブルーチンを使う．

```
sendi(var,plist,process_id);
recvi(&var,plist,process_id);
sendbk(&var,size,plist,process_id);
recvbk(&var,size,plist,process_id);
```

ここにｐｌｉｓｔは宛先のプロセッサのプロセッサリストで，ｐｒｏｃｅｓｓ＿ｉｄはそこで動作しているプロセスのプロセス番号である．ｓｅｎｄｉ，ｒｅｃｖｉはショートデータのメッセージを作って送り出したり，受け取ったりする仕事をし，ｓｅｎｄｂｋ，ｒｅｃｖｂｋはロングデータのメッセージを扱う．

この外にホスト計算機に対して入出力のサービスを依頼する特別なメッセージなども必要になるであろう．

９．まとめ

二進木結合の特長を生かしたプログラムの実行制御方式を説明し，これにもとづいたメッセージ駆動式オペレーティングシステムの提案を行った．今後このＯＳをインプリメントし評価したいと考えている．

参考文献

1) Takahashi,Y.,et al.: "Efficiency of Parallel Computation on the Binary-Tree Machine CORAL'83", Journ. of Information Processing, Vol.8, No.4,pp.288-299 (Mar. 1986)

2)遠藤俊雄，松尾賢二，白方新洋，樋谷一，高橋義造：２進木マシンＣｏｒａｌ６８Ｋのシステム構成と性能評価，情処研報Vol.87,No.45,87-CA-66-5 (1987-7)

3)高橋義造：並列処理のためのプロセッサ結合方式，情報処理，23巻、3号， pp.201-209 (1982-3)

4)Takahashi,Y.:"Fault Tolerance in Processor Networks"，第25回情報処理学会全国大会、1F-6 (1982-10)