計算機アーキテクチャ　71－3
（1988. 7. 7）

データ駆動型シングルチッププロセッサ EMC－Rのアーキテクチャ

Architecture of a Dataflow Single Chip Processor EMC-R

坂井　修一　　山口　喜教　　平木　敬　　児玉　祐悦　　弓場　敏嗣

Shuichi SAKAI, Yoshinori YAMAGUCHI, Kei HIRAKI, Yuetsu KODAMA, Toshitsugu YUBA

電子技術総合研究所　電子計算機部

Computer Systems Division, Electrotechnical Laboratory

あらまし　データ駆動アーキテクチャの有効性は、数十台から百台の並列要素数をもつ実機で検証されつつある。われわれは千台規模の要素プロセッサをもつデータ駆動計算機EM-4を開発中であるが、この規模の並列計算機を構築するさいに、物理的な大きさと設計の複雑さをいかにおさえ、高速かつ低コストなマシンを実現するか、ということが大きな課題であり、要素プロセッサ１台を１チップ以下に実装することが必要となる。EM-4では、１個のＰＥは約５万ゲートのCMOSゲートアレイチップ上に実現されるが、本ＰＥはEMC-Rと呼ばれる。本稿では、EMC-Rのアーキテクチャを決定する際の基本戦略である、簡単化された待ち合せ機構、強連結枝モデルの導入によるＲＩＳＣ化と高速化、プロセッサ結合型網の採用などに関して述べ、これを活かすための命令セットアーキテクチャ、内部アーキテクチャに関して報告する。

Abstract

A dataflow machine is being proved to be effective, by real machines containing some tens or a hundred of processing elements(PE's). We are now developing a dataflow machine EM-4 containing one thousand PE's. To construct it, it is fairly significant to fabricate a PE (or PE's) on a single chip for the reason of the feasibility of size, design complexity, cost and operation speed. In EM-4, a PE is called EMC-R which is a 50,000-gate CMOS single chip processor. In this report, we will first explain the fundamental strategies such as a simplified matching mechanism, strongly connected arc model, an idea of a dataflow RISC and processor connected interconnection network. Then instruction set architecture and configuration architecture which exploit the strategies will be reported.

１．はじめに

データ駆動アーキテクチャの有効性は、数十台から百台の規模の並列要素数をもつ実機で検証されつつあり[1][2][3][4][5]、その成果は一部で実用化されている[6]。これらのうち、SIGMA-1は128台のＰＥをもつシステムが現在稼働中であり、250MIPS、170MFLOPSの実測性能を得た[7]。

われわれは、EM-3[4]とSIGMA-1[3][7]の試作経験をいかし、1000台規模の要素プロセッサをもつ記号処理用データ駆動計算機EM-4を開発中である[8][9][10][11]。

1000台規模の並列計算機を構築するさいに、物理的な規模の大きさと設計の複雑さをいかにおさえるか、ということが大きな課題であり、要素プロセッサ（ＰＥ）１台を１チップ以下か、多くとも２・３チップ以下の規模に実装することが、不可欠となる。

EM-4では、この理由のほかに、高速性、低コスト化などの点から、ＰＥをシングルチップ化している。本シングルチッププロセッサはEMC-R[9]と呼ばれ、約５万ゲート、入出力信号線数256のCMOSゲートアレイチップ上に実装される。

本稿では、EMC-Rのアーキテクチャに関して述べる。まず、EMC-R設計にあたっての基本的な考え方と設計方針に関して述べ（２）、つぎに本プロセッサのパケット形式・命令セットとその特徴を示す（３）。さらにEMC-Rの内部構成に関して各ユニットごとに説明し（４）、最後に本研究の今後の課題を整理する（５）。

２．設計方針

２．１　シングルチップ化の範囲

要素プロセッサのシングルチップ化にあたっては、プロセッサ本体だけでなく、主要な記憶要素や相互結合網の要素も含めた１チップ化が望ましい。今回はゲート数が約５万という制約から、記憶装置は、外部にＳＲＡＭをもたせることとし、チップ上にレジスタファイルとパケット・バッファを設け、メモリを階層化することにした。さらに相互結合網の要素機能としての通信機能は、チップ上で実現することにした。通信機能と演算機能は、同時に独立に動作する。

なお、EMC-Rでは、設計の容易さと高速化の目的から、ＲＩＳＣアーキテクチャを採用することにした。

２．２　待ち合せ処理の簡単化

データ駆動型計算機のＰＥをシングルチップ化する場合の問題点として、待ち合せ処理部のハードウェアが大きく、複雑なことがある。EM-4では、関数インスタンスごとに待ち合せメモリのセグメントを割り当て、セグメント内は静的な待ち合せを行なうことで、連想処理を用いない発火機構を実現した。これによって、簡単なハードウェアで待ち合せ処理部を構成することが可能になった[11]。

今回の実装では、セグメントは固定長とする。われわれは、本方式のほかに、セグメントを可変長とし、セグメント内に固定長のページを置く方式を検討したが、チップ内にページテーブルを格納する記憶領域をとることができず、またチップ外にこれを置くのはマッチングに時間がかかりすぎる、という欠点があるため、これを採用しなかった。

２．３　強連結枝モデルの導入による高速化

一般にデータ駆動計算機では、パケット転送・パケット処理のオーバヘッドが大きくなる危険がある。またEMC-Rにおい

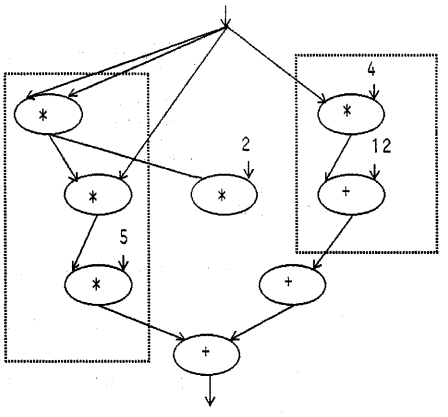
ては、通常のパケット処理の巡回パイプラインとＲＩＳＣアーキテクチャの整合性が悪いという問題がある。すなわち、従来の巡回パイプラインだけを処理方式としてゆるした場合、レジスタファイルを利用したＲＩＳＣ特有の高速な処理サイクルがいかせない。

EMC-Rでは、この問題の解決をひとつの目的として、強連結枝モデルの導入を行なった[9]。これは、データ駆動図式上に強連結ブロックと呼ばれる不可分ブロックを指定し、これを同一プロセッサ内のレジスタファイルの上でパケットをつくらずに処理するものである。強連結枝モデルの導入によって、通常の巡回パイプラインの他に、強連結ブロック処理用の短いパイプラインを並立させたことになる。

強連結ブロック内の命令の実行順序制御方式として、カラーのないデータ駆動方式と、制御駆動方式が考えられる。EMC-Rでは、ハードウェアの簡単さ、デバグ容易性、副作用をもたらす操作の効率化などの点から、制御駆動方式を採用した。

なお、本モデルの導入によって、高速化だけではなく、資源管理や優先処理が可能となった。

図１に、強連結ブロックの例として、 多項式 $5x^3+2x^2+4x+12$を計算するデータ駆動図式を示す。図で、破線で囲った部分が強連結ブロックである。



$5x^3+2x^2+4x+12$を計算するプログラム（破線内強連結ブロック）
図１　強連結ブロックの例

２．４　相互結合網

相互結合網の構成として、通常のオメガ網のような多段構成の網を採用した場合、網のハードウェア量がO(NlogN)（Ｎ：ＰＥ数）となり、N=1000では現実的でない。そこでEM-4では、プロセッサ結合型オメガ網（図２）を採用した。[12]。本結合網は、総ハードウェア量がO(N)、ノード間距離がたかだかO(logN)、ノードのポート数が定数、ルーティングが容易、内部で２階層の階層構成が可能、などの優れた性質をもつ。

図２で、長方形であらわされている部分が、EMC-Rのデータ交換部である。

EM-4の相互結合網のもうひとつの特徴は、網自体が関数レベルの動的負荷分散機構をもつことである。
すなわち、EM-4では、裏循環路（図２の破線）と呼ばれる経路に特殊パケットを巡回させることで、均等で効率のよい負荷分散を実現している（４．２）[12]。

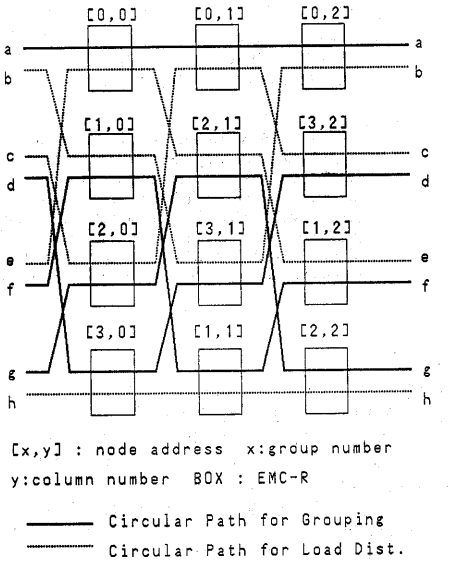


図２　プロセッサ結合型オメガ網

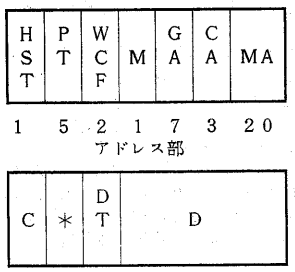
３．命令セットアーキテクチャ

３．１　ＲＩＳＣ化

さきにも述べたとおり、EMC-RではＲＩＳＣアーキテクチャを採用した。その内容を列挙する。
(1)パケット長は２語の固定長とする。
(2)命令は、即値データをもたないものが１語、もつものが２語の長さをもつ。
(3)命令数は、２８である。
(4)メモリの連続番地読みだし（書き込み）と分岐命令を除くすべての命令は１クロックで実行される。
(5)１６ワードのレジスタファイルをもつ。レジスタファイルの大きさは、ゲート数の制限からくるものである。
(6)遅延分岐機構をもつ。
(7)演算ユニットの用いるメモリのアドレッシングモードは一種類だけである。

３．２　パケット形式

本マシンの典型的なパケットの形式を図３に示す。パケットはアドレス部とデータ部からなり、各部の大きさはそれぞれ１語（３９ビット）である。



数字はフィールドの大きさ（ビット）
図３　パケット形式

３．２．１　アドレス部

(1)ＨＳＴ：当該パケットがホストプロセッサに送られる（１）か、ＰＥに送られるかを表わす。ホストに送られるのは、トレースをしている場合の全パケット、エラーレポートのパケット、最終結果の入ったパケットなどである。
(2)ＰＴ：パケット型。通常はＮＯＲＭ型であるが、関数起動

・終了、エラー処理、メモリ読みだし・書き込み、擬結果の処理、構造体の処理などで、特殊な型のパケットを用いる。特殊パケットは、システム初期化のときに自由に設定することができる。
(3)ＷＣＦ：待ち合せ条件フラグ。待ち合せの型を示す。１入力型、即値待ち型、２入力左型、２入力右型の４種類がある。
(4)Ｍ：モード。２ノード以上の大きさをもつ強連結ブロックを発火させるパケットであるときに１、１ノードのブロック（通常のデータフローノード）を発火させるパケットであるときに０である。
(5)（ＧＡ，ＣＡ，ＭＡ）：行先アドレス。ＧＡは行先ＰＥのグループ番号、ＣＡは行先ＰＥのグループ内メンバ番号、ＭＡはメモリアドレスである。ＰＴがＮＯＲＭ型で、かつＷＣＦが２入力型のパケットでは、ＭＡは待ち合せ番地となる。

３．２．２　データ部

(1)Ｃ：キャンセルビット。Ｃ＝１のとき、このパケットは無効となる。
(2)ＤＴ：データ型。EMC-Rで用いられるデータ型は、整数、アイデンティファイア、ＮＩＬ、リスト、ユーザ定義構造体（２つ）、擬結果、無定義の８種類である。
(3)Ｄ：データ。

３．３　命令形式

本マシンの命令は、命令コードの命令をレジスタファイル上のデータ（あるいはパケットデータ）にほどこし、結果をレジスタファイルに格納する（またはパケット出力する）操作を行なうことを指示する。

命令形式は、即値データをもつか否か、パケットを出力するか否かによって、４種類に分類される。なおEM-4では、メモリ読み書きの命令と分岐命令は、単独ではパケットを出すことができない。

EM-4では、１命令につきたかだか１個しかパケットを出力しない。複数個のパケットを出力する場合は、当該命令とＭＫＰＫＴ（Make Packet）命令を強連結枝で接続する。１命令が複数個パケットを出力する場合に比較して、本方式では、効率を損ねることがなく、かつ制御が簡単化されている。

３．３．１　即値データがなくパケットを出力しない場合

図４(a)に示される形式である。
(1)ＯＰ：命令コード。
(2)ＡＵＸ：命令コードの補助フィールド。命令の細かな性質を規定したり、場合によっては、即値データを格納するのに用いられる。
(3)ＴＲＣ：トレースの指示。
(4)Ｍ：モード指定。Ｍが１のとき、本命令で当該強連結ブロックを終了する。
(5)Ｆ：命令フォーマット。０のとき即値なし、１のとき即値あり。ここでは０。
(6)Ｒ０，Ｒ１：当命令のオペランドの入っているレジスタの番地。
(7)ＯＵＴ：パケット出力があるか否かを示すフィールド。この場合は０（出力なし）。
(8)Ｒ２：当命令の結果を格納するレジスタの番地。
(9)ＢＣ：当命令が分岐命令だったときの、分岐時の行先指定の条件。
(10)ＤＰＬ：当命令が分岐命令だったときの、分岐先の番地。

３．３．２　即値データがなくパケットを出力する場合

図４(b)に示される形式である。ＯＰ、ＡＵＸ、ＴＲＣ、Ｍ、Ｆ（この場合０）、Ｒ０、Ｒ１、ＯＵＴ（この場合１）は３．３．１と意味が同じである。結果はパケットとして出力され、レジスタファイルには入らない。

(1)ＷＣＦ：出力パケットのＷＣＦフィールド。すなわち、パケットの待ち合せ条件フラグとなる。
(2)Ｍ２：出力パケットのＭフィールド。すなわち、パケットのモードとなる。
(3)ＣＡ：出力パケットのＣＡフィールド。すなわち、パケットの行先ＰＥのメンバ番号となる。
(4)ＤＰＬ：出力パケットのＭＡの階８ビットとなる。

３．３．３　即値データがありパケットを出力しない場合

図４(c)に示される形式である。３．３．２と異なるのは、即値データのある語が加わった点である。
(1)ＬＲ：当即値データが左オペランドである（０）か、右オペランドである（１）かを示す。
(2)ＤＴ：データ型。
(3)Ｄ：データ。

３．３．４　即値データがありパケットを出力する場合

図４(d)に示される形式である。３．３．２と異なるのは、即値データのある語が加わった点である。この語の意味は、３．３．３と同じである。

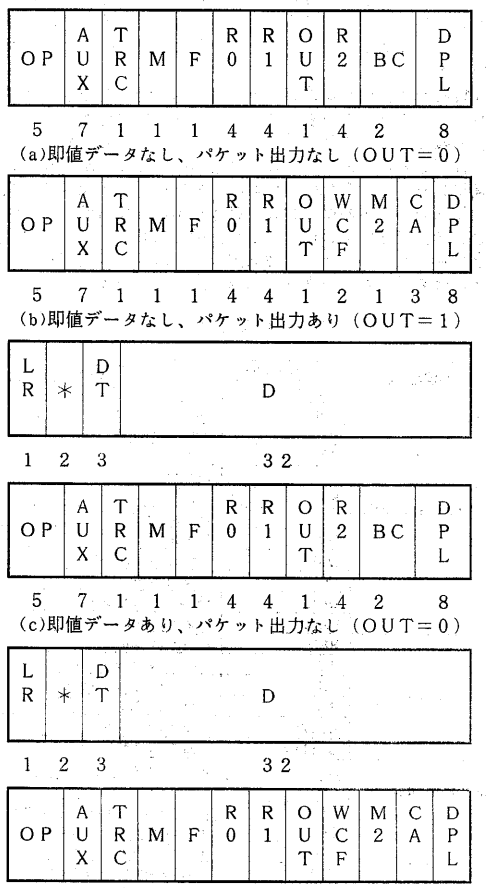


図４　命令形式

3．4　命令セットとその特徴

表1に、EM-4の命令セットを示す。

EM-4は、全部で２８個の命令をもつ。これらの命令は、メモリ連続読みだし（書きこみ）の命令と分岐命令以外はすべて１クロックで実行される。命令のうちのあるものは、ＡＵＸフィールドによって、細かな実行規則を与えられる。

たとえば、ＳＨＦ命令は、ＡＵＸフィールドの先頭ビットが０のとき左論理シフト、１のとき右論理シフトを実行する。ＢＴＹＰＥ命令では、ＡＵＸフィールドは、比較する型をあらわす即値データと解釈される。すなわち、（ＢＴＹＰＥ　ＡＵＸ＝０）はＬＩＳＰのＮＵＬＬ、（ＢＴＹＰＥ　ＡＵＸ＝１）はＮＵＭＢＥＲＰ、（ＢＴＹＰＥ　ＡＵＸ＝３）はＡＴＯＭ、（ＢＴＹＰＥ　ＡＵＸ＝４）はＬＩＳＴＰとなる。

なお、EMC-Rでは、マイクロプログラム制御を用いない。これは、マイクロプログラムを格納するメモリを、(1)チップ内にとることはゲート数の制約から困難であり、(2)チップ外にとることはピン数の制約から困難であることによる。

以下に本マシンに特徴的な命令について説明する。

(1)分岐命令

EM-4では、実装の複雑さの点から、ＳＷＩＴＣＨ命令の実装を省略し、条件分岐はすべて制御の流れを変更するＢＲＡＮＣＨ命令で実現することにした。したがって、この部分だけは、純粋なデータ駆動計算機（命令レベル）としての仕様を破っている。表１のＢＥＱ、ＢＧＥ、ＢＤ、ＢＧＴ、ＢＴＹＰＥ、ＢＴＹＰＥ２がＢＲＡＮＣＨ命令である。このうち、ＢＴＹＰＥ２は２つのオペランドの型比較を同時に行なう命令である。

分岐命令は、判定結果がジャンプを引き起こす場合には２クロック、そうでない場合には１クロックで実行される。なお、分岐はすべて遅延分岐である。　ＳＷＩＴＣＨ命令は、ＢＲＡＮＣＨ命令とＭＫＰＫＴ命令の組合せで実現されるが、強連結ブロックを有効に用いれば、分岐命令の出現頻度が減り、従来のデータ駆動方式より効率があがる場合が多いと予想される。図５にその例を示した。図５(a)は、ＳＷＩＴＣＨ命令を用いたプログラム（従来方式）、図５(b)は、ＢＲＡＮＣＨ命令を用いたプログラム（EM-4方式）である。

表１　命令セット

KIND	INSTRUCTION	ACTION
Arithmetic and Logic	ADD	integer add
	SUB	integer subtract
	MUL	integer multiply
	INC	increment
	DEC	decrement
	SHF	logic shift
	AND	bitwise AND
	OR	bitwise OR
	EOR	bitwise exclusive OR
	NOT	bitwise NOT
	DIV1	element of division
	EXPD	data expand for division
Branch	BEQ	branch by equality
	BGT	branch by greaterness
	BGE	branch by greaterness or equality
	BD	branch by data
	BTYPE	branch by data type
	BTYPE2	branch by 2 data types
Memory or Register Read or Write	L	load from memory
	S	store to memeory
	LS	load and store from/to memory
	LDR	load directly from register
	LIR	load indirectly from register
Others	GET	send packet for data fetch
	MKPKT	make packet by two operands
	SENDOP	send error operand
	SENDIAR	send error instruction address
	SETDT	set data type

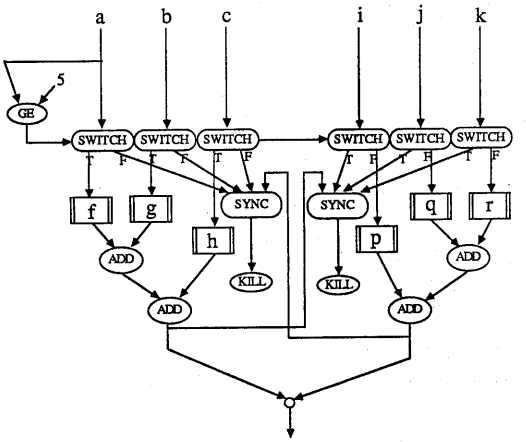
図５(a)では、ＳＷＩＴＣＨ命令が計６個あり、また、テスト命令ＧＥの結果は、これらすべての命令の入力になっている。命令ＧＥからのトークン供給の速さには限界があり、ここが全体の処理時間を決定する。また、関数の終了処理のために残留トークンの処理を行なう必要があり（図中の２つのＳＹＮＣ命令）、オーバヘッドが生じる。

それに比較して図５(b)では、テストとデータ分配がすべて同一強連結ブロックに含まれているため、この部分では、通常のデータ駆動型の待ち合せを行なわない。そのため、処理が高速化されている。また、この例では、残留トークンを生じないため、関数の終了処理のオーバヘッドがなくなる。

(2)ＭＫＰＫＴ

メモリ読みだし（書き込み）および分岐命令をのぞくすべての命令は、結果をパケットとして出力することができる。さらにEM-4では、パケット生成専用のＭＫＰＫＴ命令が用意されている。

ＭＫＰＫＴ命令は、第１オペランドで示される番地へ第２オペランドのデータを送りつける命令であり、ＡＵＸフィールドで、パケット型（ＰＴ）を指定することができる。ＭＫＰＫＴ命令を連続して用いることで、データのコピー・分配



(a)ＳＷＩＴＣＨ命令による実現

```
        (a=R0, b=R1, c=R2, i=R3, j=R4, k=R5)
        - - -
        - - -
        [BGE (R3:'5) (THN)]
        [NOP]
        [MKPKT (SEG:R3) <CALL_P single>]
        [MKPKT (SEG:R4) <CALL_Q single>]
        [MKPKT (SEG:R5) <CALL_R single> end_block]
THN     [NOP]
        [MKPKT (SEG:R0) <CALL_F single>]
        [MKPKT (SEG:R1) <CALL_G single>]
        [MKPKT (SEG:R2) <CALL_H single> end_block]

CALL_F  {GCALL 'f (L10 left)}
CALL_G  {GCALL 'g (L10 right)}
CALL_H  {GCALL 'h (L20 right)}
L10     (ADD (L20 left))
L20     (ADD (- - -))

CALL_P  {GCALL 'p (L110 left)}
CALL_Q  {GCALL 'q (L110 right)}
CALL_Q  {GCALL 'r (L120 right)}
L110    (ADD (L120 left))
L120    (ADD (- - -))
```

(b)ＢＲＡＮＣＨ命令による実現

```
(if (a>=5) then f(a)+g(b)+h(c)
           else p(i)+q(j)+r(k))
```

図５　分岐命令の実現

をすることができるが、これは、SIGMA-1のＤＩＳＴ命令（１命令に複数の行先をゆるす）と同じクロック数で実現され、後者より柔軟なデータ流生成が可能となる。その際、第１オペランドのレジスタとしては、ＳＥＧ（当該セグメント番号の入ったレジスタ）が指定される。

また、本命令によって、関数起動の際の引き数の受渡しや、戻り値の受渡しが実現される。

(3)ＧＥＴ命令

ＧＥＴ命令は、第１オペランドで示される番地Ａへ戻り番地（図４(b)のＣＡとＤＰＬで指定される）を送りつける命令である。これは、番地Ａのデータになんらかの操作を加えて、結果を戻り番地に送り返す、という操作を行なうときに用いられる。ＭＫＰＫＴ同様、ＡＵＸフィールドで、パケット型（ＰＴ）を指定することができる。

具体的には、ＣＡＲ、ＣＤＲなどの構造体処理、擬結果関数の起動、他ＰＥのメモリ読みだし、ホストとの交信などに用いる。

３．５　マクロ命令

EM-4では、複雑な命令は、より単純な命令の列からなる強連結ブロックの形で実現される。これをマクロ命令と呼ぶ。以下にＭＳＹＮＣマクロ命令を例にとって説明する。

ＭＳＹＮＣ(Multiple SYNC)マクロ命令は、強連結ブロックの入口などで、多数（３以上無限個）のオペランド待ち合せを行なう場合に用いる。図６に、ＭＳＹＮＣの内容を示した。ＭＳＹＮＣは、２入力のデータ待ち合せと、局所的なカウンタの更新によって実現される。すなわち、データが到着し、オペランドのうちあらかじめ定められたデータとのマッチングに成功したとき、当マクロ命令が起動される。当マクロ命令では最初に、対応するメモリ番地に２つのデータを格納し（２行目のＳ命令）、カウンタを更新する（６行目のＡＮＤ命令から８行目のＡＤＤ命令）。その結果、すべてのオペランドの到着が確認された場合、オペランドデータをレジスタファイルにロードして（MLOAD）、以下の演算を実行する。まだ到着していないオペランドがある場合は、強連結モードを解除し（１１行目のＮＯＰ）、未到着データの到着を待つ。

このように、EMC-Rでは、複雑な手順を要する操作が、強連結ブロックを用いたマクロ化によって、単純化され、しばしば高速化される。

なお、マクロ命令としては、ＭＳＹＮＣの他にＣＯＮＳなどの構造体操作マクロ命令、ＧＣＡＬＬなどの関数制御マクロ命令などがある。

３．６　特殊パケットの処理方式

EM-4には、通常の待ち合せを行なうパケット以外に、セグメント番号をアドレス部にもたない特殊なパケットが存在する。表２にEM-4のパケット型の構成例を示したが、このうちＮＯＲＭ型以外のパケット型がこれにあたる。

特殊パケットは、通常ＭＫＰＫＴ命令またはＧＥＴ命令で生成される。

特殊パケットとして、関数制御関係のパケット（ＭＬＰＥ、ＧＥＴＧＳＥＧ、ＳＥＴＳＥＧ、ＫＩＬＬ、ＡＣＴＩＶＡＴＥ）、構造体操作のパケット（ＲＳ、ＲＳ２、ＩＲＥＦ、ＤＲＥＦ）、エラー処理のためのパケット（ＥＲＲＯＲ０、ＥＲＲＯＲ１、ＥＲＲＯＲ２）、メモリとレジスタの読み書きを行なうパケット（Ｒ、Ｗ、ＲＲ、ＷＲ）などを考えている。

特殊パケットの処理は、強連結ブロック（モニタ）の呼び出しの形で実現される。具体的には、特殊パケットが到着すると、パケット型に対応した番地の命令が（待ち合せなしに）読みだし・実行され、以下この番地に続く強連結ブロックの命令がつぎつぎに実行される、という方式である。

本方式によって、あるパケット型に対応する処理が、通常のプログラムと同様の形式で書けるため、特殊パケットの意味付けと処理内容の柔軟さがもたらされる。図７にＲパケットの処理ルーチンを示す。Ｒパケットは、そのアドレス部のメモリデータを読みだし、データ部のアドレスにこれを送りつけることを命じる特殊パケットである。

３．７　プログラム例

EM-4におけるプログラムの例として、図８にAPPEND関数、図９にフィボナッチ関数のアセンブラによる記述を示した。両方とも、型判定と分岐の部分は、ひとつの強連結ブロック内で行なわれており、高速化・簡単化されている。また、従来のデータ駆動図式（図は略）と比較して、処理の隘路となる２入力データ駆動命令の数（即値をもつものを除く）が、APPENDでは８個から４個に、フィボナッチでは４個から２個になっており、処理効率が向上していると判断される。

４．内部アーキテクチャ

４．１　全体構成

EMC-Rの全体構成を、図１０に示す。EMC-Rは、ＳＵ、ＩＢＵ、ＦＭＵ、ＥＸＵ、ＭＣＵの５つのユニットとメンテナンス回路、および外付けのＳＲＡＭからなる。ＳＵは２．４で述べた結合網の要素機能を実現するスイッチであり、他のユニットと独立・並列に動作する。ＩＢＵは入力バッファであり、チップ上に数十語のバッファを持ち、さらに外付けＳＲＡＭ上に退避領域を持つ。ＦＭＵは待ち合せと命令フェッチ

```
        [JMP (MSYMC)]
        [S (OAR:R0) WORDS=2]
        - - -
        - - -
        - - -
MSYNC   [AND (OAR:'FFFFFFF8) R3]
        [L (R3:NL) R4 WORDS=1]
        [ADD (R4:'2) R4]
        [BEQ (R4:'xxx) (MLOAD)]
        [S (R3:R4) WORDS=1]
        [NOP end_block]
MLOAD   [INC (R3:NL) R3]
        [L (R3:NL) R4 WORDS=xxx]
        - - -
        - - -
        - - -

(xxx represents the number of sync data)
```

図６　多数待ち合せの実現（ＭＳＹＮＣ）

表２　特殊パケット（例）

KIND	PACKET TYPE	MEANING
Normal	NORM	normal data packet
Function Control	MLPE	search minimum loaded PE
	GETGSEG	get operand segment number
	SETSEG	bind operand segment to template
	KILL	kill function
	ACTIVATE	activate pseudo function
Structure	RS	read structure(CAR, CDR)
	RS2	read both of a CAR part and a CDR part
	IREF	increment reference counter
	DREF	decrement reference counter
Memory or Register Read of Write	R	read memory
	W	write memory
	RR	read register
	WR	write register
Maintenance	ERROR0	error operand 0
	ERROR1	error operand 1
	ERROR2	error instruction address

```
[LD (MR:R2) WORDS=1]
[MKPKT (R0:R2) end_block]
```

図７　Ｒパケットの処理

をつかさどる。待ち合せ記憶と命令記憶はＳＲＡＭ上にとられる。ＥＸＵは演算処理部である。ＳＲＡＭのバスはＭＣＵでマルチプレクスして各ユニットに用いられる。

通常の処理は、待ち合せ・命令フェッチ・実行の順で各ユニットをパケットが巡回することで実現される（図１０の矢印つきの実線）が、強連結処理は、ＥＸＵ内のレジスタファイル上で待ち合せなしに実現される（矢印つきの破線）。

以下に各ユニットの説明を行なう。

４．２　ＳＵ

ＳＵは、３入力３出力のパケット交換スイッチであり、入力ポートにバッファをもつ（蓄積型）。ＳＵの内部構成を図１１に示す。

相互結合網としてプロセッサ結合型オメガ網（図２）を用いた場合、ストアアンドフォワードデッドロックを防止する機構が必要になる。本マシンでは、各入力ポートに３つのバッファを設け（図１１の各ＩＰ参照）、パケットが第０段のＳＵに到達するたびに、これまでよりひとつ上位のバッファを用いる、という方式を採用することで、この問題を回避した[12]。本方式によってストアアンドフォワードデッドロックが回避されるのは、網の論理的構成がループフリーになることによる。

さらに、ＳＵには、２．４で述べた負荷分散のためにパケットの書換えを行なうことが必要である。これは、ＭＬＰＥという特殊パケットを、網の裏循環路上のＰＥが書き換えていくことで実現される。ＭＬＰＥパケットのフォーマットを図１２に示す。

ＭＬＰＥパケットがＮＯＲＭ型パケットと異なる点は、ＭＡ部に当該パケットの通ったＰＥの負荷量の最小値が入っている（ＭＬフィールド）点と、データ部の２９～２０ビットに最小負荷ＰＥ番号が入っている（ＭＰフィールド）点である。

ＭＬＰＥパケットの書換えは、図１１中のＰＲＣ（パケット書換え部）で行なわれる。ＰＲＣでは、ＭＬＰＥパケットが到着したのと同時に、ＭＬフィールドを書き換える必要があるかどうかを判断し、必要がある場合は自ＰＥの負荷量を

```
        [BTYPE (R0) 'NIL (L100)]
        [NOP]
        [BTYPE (R0) 'LIST (L200)]
        [NOP]
        [MKPKT (SEG:R0) <ERROR single> end_block]
L100    [MKPKT (SEG:R1) <RETVAL right> end_block]
L200    [MKPKT (SEG:R1) <APND right>]
        [L (R0:NL) R5 WORDS=2]
        [MKPKT (SEG:R5) <CNS left>]
        [MKPKT (SEG:R6) <APND left> end_block]

APND    {GCALL 'append (CNS right)}
CNS     {CONS (RETVAL right)}

RETVAL  [MKPKT]
        [KILL end_block]
ERROR   - - -
```

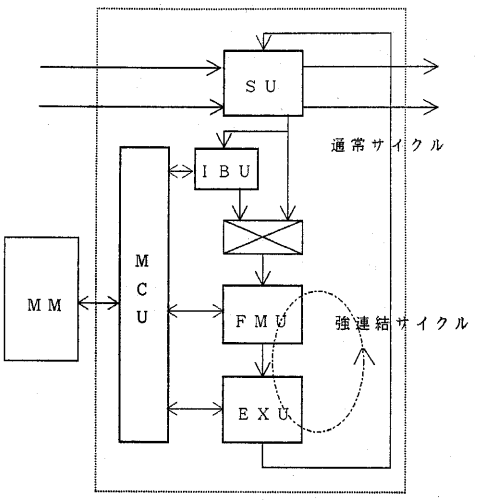
図８　ａｐｐｅｎｄプログラム

```
        [BD (R0:'1) (L100)]
        [SUB (R0:'1) R1]
        [BD (R0:'2) (L100)]
        [SUB (R0:'2) R2]
        [MKPKT (SEG:R1) <L10 single>]
        [MKPKT (SEG:R2) <L20 single> end_block]
L100    [MKPKT (SEG:'1) <RETVAL right> end_block]

L10     {GCALL* 'FIB  (L30 left)}
L20     {GCALL* 'FIB  (L30 right)}
L30     (ADD    (RETVAL right))

RETVAL  [MKPKT]
        [KILL end_block]
```

図９　フィボナッチ数をもとめるプログラム



SU: Switching Unit　MCU: Memory Control Unit
IBU: Input Buffer Unit　FMU: Fetch and Matching Unit　EXU: Execution Unit　MM: Memory Module)
DOTTED RECTANGLE: EMC-R

図１０　EMC-Rの内部構成

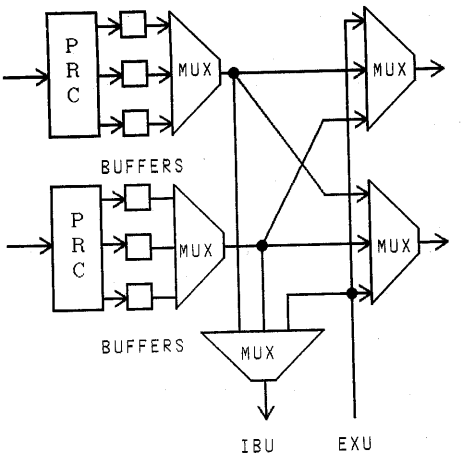
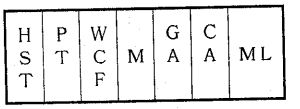


図１１　ＳＵの構成



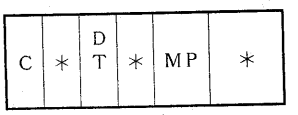


図１２　ＭＬＰＥパケット

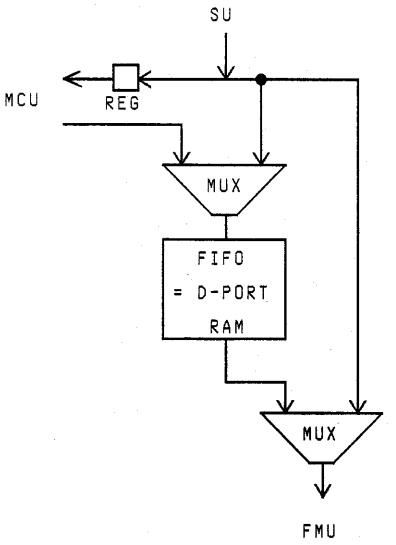
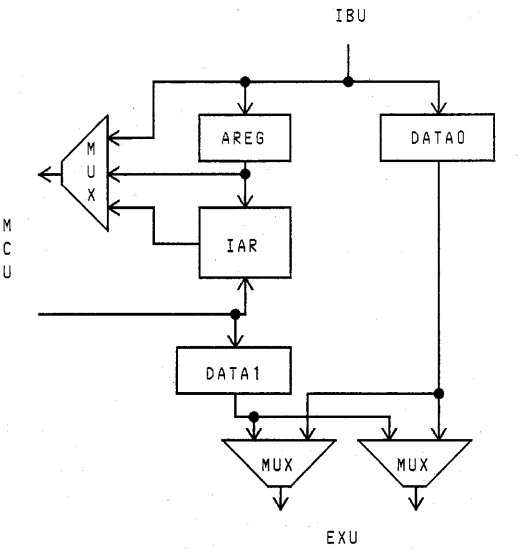
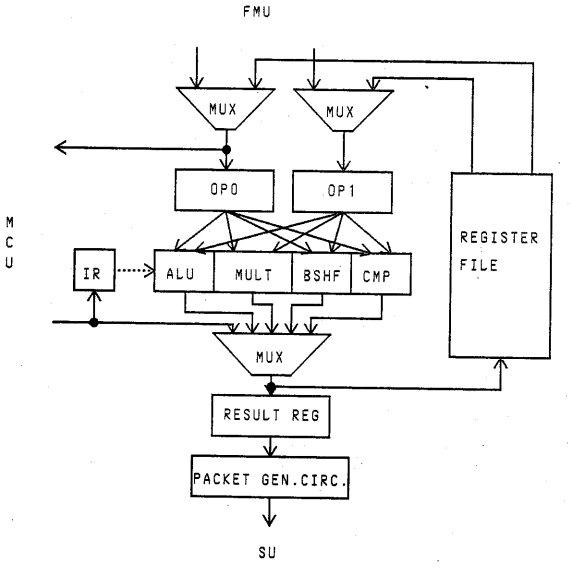


図１３　ＩＢＵの構成



AREG:Address Part Reg. DATA0:Data Part Reg.
IAR:Instruction Address Reg.
DATA1:Partner Data Reg.

図１４　ＦＭＵの構成



IR:Instruction Register MULT:Multiplier
BSHF:Barrel Shifter CMP:Comparator

図１５　ＥＸＵの構成

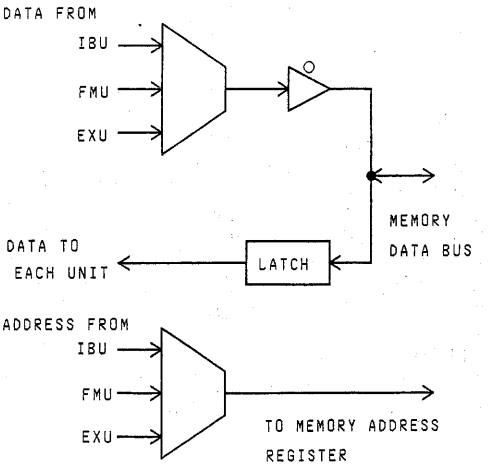


図１６　ＭＣＵの構成

ＭＬフィールドに書き込む。この場合、パケットのデータ部の到着と同時に、ＭＰフィールドを当該ＰＥ番号に書き換える。

以上の書換えは、パケット転送時間以内に行なわれるため、ＭＬＰＥパケットはＮＯＲＭ型パケットとまったく同じ速さで、網内を巡回する。

なお、EM-4の関数レベル負荷分散方式として、ＭＬＰＥを用いる方式以外に、ユーザプログラムやコンパイラが関数の分配を決める静的な方式、乱数分配方式、ローテート方式などが用意されている。

４．３　ＩＢＵ

ＩＢＵは、実行待ちのパケットのバッファである。チップ内で２ポートＲＡＭを使ってＦＩＦＯバッファを実現している（図１３）。このバッファは１６語の容量があり、あふれたときは、外部にあるＳＲＡＭ上にパケットを退避する。外部の退避領域がいっぱいになったときには、デッドロックが発生する危険が大きいため、致命的なエラーとみなして、エラー処理の手続きを実行する。

４．４　ＦＭＵ

ＦＭＵは、マッチングと命令フェッチを行なうユニットであり、EMC-Rの制御の中心である。図１４にその内部構成を示

す。FMUは、命令アドレスレジスタ、パケットデータレジスタ、マッチング用のデータレジスタ、アドレスのマルチプレクサ、オペランドデータのマルチプレクサ、および制御回路からなる。制御回路は、本マシン内の種々のパイプラインを実現するためのシーケンシングをつかさどる。命令は、パケットによって発火する場合と強連結ブロック内の発火規則にしたがって発火する場合があり、その調停が必要である。

FMUの処理するパイプラインとして、(1)1入力データ駆動命令のパイプライン、(2)2入力データ駆動命令のパイプライン、(3)特殊パケット処理のパイプライン（最初の1命令の読みだしまでが通常のパイプラインと異なる）、(4)強連結ブロック内のパイプラインなどがある。

4．5　EXU

EXUは演算実行部である。EXUの内部には、命令レジスタ、オペランドレジスタ、レジスタファイル、出力パケット生成回路、結果レジスタ、加減算器、乗算器、バレルシフタ、比較器、いくつかのデータマルチプレクサ、および制御回路がある。命令実行時には、オペランドレジスタの内容に演算がほどこされ、結果はパケットデータレジスタに入れられて外部に出力されるが、レジスタファイルに格納される。それと同時に、つぎに実行される命令で使われるデータが、レジスタファイル（またはFMUのデータレジスタ）からオペランドレジスタにロードされる（図15）。

なお、レジスタファイル内には、最初のあきオペランド・セグメント（フリーチェインでつながれているセグメント群の先頭のセグメント）の先頭番地をさすアドレスレジスタ、あき構造体領域の先頭をさすアドレスレジスタ、現在実行中の命令の属するセグメントを示すレジスタ（SEG）、時間的にもっとも遅く到着したパケットのアドレス部を格納しておくレジスタなどの専用レジスタがある。

4．6　MCU

MCUは、各ユニットからの外付けメモリへのアクセス競合を調停するユニットである（図16）。メモリのデータバスとアドレスバスは、IBU（パケット退避）、FMU（マッチングと命令フェッチ）、EXU（メモリ読み書き命令）の3つのユニットが使用する。3者の優先順位は、EXU、IBU、FMUの順であり、また、直前のクロックでメモリを使用していたユニットは、原則としてつぎのクロックでも他ユニットに優先してこれを使用する権利がある。ただし、EXUからパケット出力中にIBUがパケット退避要求を出したときは、デッドロック状態になっている可能性があるため、FMUはつぎの命令の読みだしを中止して、IBUにメモリを渡す。

MCUは、3つのユニットからのデータバスのマルチプレクサ、アドレスバスのマルチプレクサ（外付けのメモリアドレスレジスタの前でマルチプレクスする）、および調停回路からなる（図16）。

4．7　外付けメモリ

外付けメモリ（図10ではMM）は、1バンクのSRAMである。容量は、約1．2MBを実装する予定であり、これは約5MBまで拡張することができる。

外付けメモリの内容は、処理待ちパケットの退避領域、特殊パケット処理ルーチンの入っている領域、待ち合せの領域、通常の命令コードの格納される領域、構造体の格納される領域、その他の作業領域である。

前2者以外の領域のロケーションと大きさは、システムの初期化のときに自由に決めることができる。

4．8　メンテナンス回路

EMC-Rには、以上の各ユニットのほかに、これらを保守し、エラー処理を行なうために、メンテナンス回路と呼ばれる回路が存在する。この回路が起動されると、通常の処理とは別種のクロックが働いて、全レジスタおよび外付けメモリの読み書きが本来とは別の経路を用いて行なわれる。メンテナンス回路の詳細は別途報告することにする。

5．おわりに

データ駆動計算機EM-4のシングルチップ型要素プロセッサであるEMC-Rについて、その設計の基本方針、命令セットアーキテクチャ、内部アーキテクチャについて述べた。本マシンで用いている種々のパイプライン方式について、関数制御機構について、構造体の格納・操作機構について、およびエラー処理方式とメンテナンス機構については別途報告することにする。本マシンの今後の課題として、EMC-Rの設計・試作、シミュレーションによるEM-4の予備的な性能測定、100台規模のパイロットシステムの試作による性能測定と有効性の検証、高級言語の開発、強連結ブロックを考慮したコンパイラなどソフトウェアの開発、スケジューリング問題を含む最適化手法の確立、ベンチマークプログラムの設定などがある。また、より大規模なチップを想定した、要素プロセッサのアーキテクチャ、マシン全体のアーキテクチャの考察がつぎの課題である。

謝辞　本研究を遂行するにあたり御指導、御討論いただいた柏木電総研次長、田村電子計算機部長ならびに計算機方式研究室の同僚諸氏に感謝いたします。

参考文献

[1]Arvind, Dertouzos, M.L., and Iannucci, R.A. :A Multiprocessor Emulation Facility, MIT-LCS Technical Report 302(Sep. 1983).

[2]Gurd, J.R., Kirkham, C.C. and Watson, I. :The Manchester Prototype Dataflow Computer, CACM, Vol.28, No.1, pp.34-52(Jan. 1985).

[3]Hiraki, K., Sekiguchi, S., and Shimada, T. :System Architecture of a Dataflow Supercomputer, TENCON87, Seoul(Aug. 1987).

[4]Yamaguchi, Y., Toda, K., Herath, J. and Yuba, T.: EM-3:A Lisp-Based Data-Driven Machine, Proc. Int. Conf. FGCS., pp.524-532(1984).

[5]Amamiya, M., Takesue, M., Hasegawa, R. and Mikami, H.:Implementation and Evaluation of A List-Processing-Oriented Data Flow Machine, Proc. of the 13th Annu. Symp. on Comp. Arch., pp.10-19(June 1986).

[6]菊池、本田、能美、中田、天満：データフローコンピュータNEDIPSのアーキテクチャ、データフローワークショップ予稿集、電子情報通信学会、pp.49-56(1986).

[7]関口、平木、島田：科学技術計算用データ駆動計算機SIGMA-1全体システムの評価（その2）、第36回情処全大6B-2(Mar. 1988).

[8]坂井、山口、平木、弓場：データ駆動計算機EM-4のアーキテクチャ、　情処計算機アーキテクチャ研究会66-7(July 1987).

[9]坂井、山口、平木、弓場：データ駆動型シングルチッププロセッサEMC-Rにおける強連結枝モデルの導入、データフローワークショップ予稿集、電子情報通信学会、pp.231-238(1987).

[10]坂井、山口、平木、児玉、弓場：データ駆動計算機EM-4における要素プロセッサのシングルチップ化の検討、第37回情処全大発表予定(Sept. 1988).

[11]山口、坂井、平木、児玉、弓場：データ駆動計算機EM-4における待ち合せ機構、第37回情処全大発表予定(Sept. 1988).

[12]坂井、山口、平木、弓場：プロセッサ結合型データ駆動計算機における網構成の諸問題、第34回情処全大2Q-5(Mar. 1987).