

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3771682号
(P3771682)

(45) 発行日 平成18年4月26日(2006.4.26)

(24) 登録日 平成18年2月17日(2006.2.17)

(51) Int. Cl.

G06F 17/16 (2006.01)

F I

G06F 17/16

D

請求項の数 4 (全 11 頁)

(21) 出願番号	特願平9-217210	(73) 特許権者	000005223 富士通株式会社
(22) 出願日	平成9年8月12日(1997.8.12)		神奈川県川崎市中原区上小田中4丁目1番1号
(65) 公開番号	特開平11-66046	(74) 代理人	100083297 弁理士 山谷 皓榮
(43) 公開日	平成11年3月9日(1999.3.9)		100096530 弁理士 今村 辰夫
審査請求日	平成13年3月16日(2001.3.16)	(72) 発明者	持山 貴司 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	曾我 武史 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 ベクトル処理装置

(57) 【特許請求の範囲】

【請求項1】

主記憶手段と、ベクトル・レジスタ手段と、ベクトル演算手段を具備し、ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向にシフトして前記ベクトル・レジスタとは別のレジスタの異なる要素番号区分に格納するベクトル処理装置において

前記ベクトル・レジスタ手段を複数のブロックにより構成し、通常のベクトル演算においては各ブロックが独立に動作し、ベクトル総和演算等の要素間の演算のために各ブロック間に専用データバスを設け、この専用データバスを用いてブロック間で要素の転送を行うとともに、

ベクトル・アクセス・パイプとは独立に並行してベクトルスライドを実行するベクトルスライド手段を設け、ベクトル演算器と独立に、並行してベクトルスライド処理を実行することを特徴とするベクトル処理装置。

【請求項2】

前記要素番号が1つ増加又は減少する方向に要素を格納することを特徴とする請求項1記載のベクトル処理装置。

【請求項3】

主記憶手段と、ベクトル・レジスタ手段と、ベクトルマスクレジスタ手段と、ベクトル演算手段を具備するベクトル処理装置において、

前記ベクトル・レジスタ手段を複数のブロックにより構成し、通常のベクトル演算にお

いては各ブロックが独立に動作し、ベクトル総和演算等の要素間の演算のために各ブロック間に専用データベースを設け、この専用データベースを用いてブロック間で要素の転送を行うとともに、

ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向にシフトして前記ベクトル・レジスタとは別のベクトル・レジスタの異なる要素番号区分に格納するとき、読み出し元のベクトル・レジスタの値と書き込み先のベクトル・レジスタの元の値をマージし、要素番号の方向に移動したときのベクトルマスクレジスタの対応するビットが「1」または「0」のいずれか一方の特定値である要素についてのみ格納を行うベクトル制御手段を具備したことを特徴とするベクトル処理装置。

【請求項4】

10

主記憶手段と、ベクトル・レジスタ手段と、ベクトル演算手段と、ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向にシフトして前記ベクトル・レジスタとは別のレジスタの異なる要素番号区分に格納するベクトル処理装置において、

前記ベクトル・レジスタ手段を複数のブロックにより構成し、通常のベクトル演算においては各ブロックが独立に動作し、ベクトル総和演算等の要素間の演算のために各ブロック間に専用データベースを設け、この専用データベースを用いてブロック間で要素の転送を行うとともに、

ベクトル・アクセス・パイプとは独立にベクトル演算器と並行してベクトルスライドを実行するベクトルスライド手段を設け、ベクトル演算器と独立に、並行してベクトルスライド処理を実行することを特徴とするベクトル処理装置。

20

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はベクトル処理装置に係り、特に主記憶からベクトル・レジスタへのベクトルロード命令の実行回数を削減することによりベクトル演算の高速化をはかるものである。

【0002】

【従来の技術】

ベクトル処理装置においてベクトル演算を行う場合、図5に示す如く、主記憶100上のデータをベクトル部101のベクトル・レジスタ部110のベクトル・レジスタVR1、VR2にベクトルロードすることが必要である。主記憶100上のデータをベクトル・レジスタ部110のベクトル・レジスタVR1、VR2に格納する際に、ベクトル・アクセス・パイプ111を使用してベクトルロード命令を実行する。この場合、アドレスが1ワードずれた配列をベクトル・レジスタ110にロードする場合でも2回のベクトルロード命令を実行していた。

30

【0003】

例えば下記のDOLoop10

```
DO 10 I = 1, 100
    A(I) = B(I) + B(I + 1)
10 CONTINUE
```

を実行する場合のように、同一配列Bで1ワードずれた場合でも、2回主記憶アクセスを行うことが必要であった。

40

【0004】

なお上記DOLoop10のプログラムは、下記の演算を行うことを示すものである。

$$\begin{aligned} A(1) &= B(1) + B(2) && \dots (1) \\ A(2) &= B(2) + B(3) && \dots (2) \\ A(3) &= B(3) + B(4) && \dots (3) \\ &\vdots && \\ &\vdots && \\ &\vdots && \\ A(100) &= B(100) + B(101) && \dots (100) \end{aligned}$$

50

このプログラムを実行するとき、第1回目のベクトルロード命令により右側の項のうちの1番目の要素 $B(1)$ 、 $B(2)$ ・・・ $B(100)$ を主記憶100からベクトル・レジスタ部110のベクトル・レジスタ $VR1$ に格納する。それから第2回目のベクトルロード命令により2番目の要素 $B(2)$ 、 $B(3)$ ・・・ $B(101)$ を主記憶100からベクトル・レジスタ $VR2$ に格納する。

【0005】

そしてベクトル演算器112により上記各式の演算を行い、その演算結果をベクトル・レジスタ $VR3$ に順次格納する。

なお、図5において102はスカラ部、113はベクトル命令制御部である。

【0006】

【発明が解決しようとする課題】

前記の如くベクトル演算を行うとき、ベクトル・レジスタ $VR2$ に格納される各要素 $B(2)$ 、 $B(3)$ ・・・ $B(101)$ が、ベクトル・レジスタ $VR1$ に格納されている各要素 $B(1)$ 、 $B(2)$ ・・・ $B(100)$ と、主記憶100上で1ワードしかずれていない配置の場合であっても、ベクトル・アクセス・パイプ111が主記憶に対するアクセスを2回行うことが必要であり、このため演算速度が遅くなり、性能低下の原因となっていた。

【0007】

即ち、前記DOLoop10の命令を実行するとき、ベクトル命令制御部113はベクトル・アクセス・パイプ111に対し、第1回目のベクトルロード命令を発行する。これによりベクトル・アクセス・パイプ111が主記憶100から配列Bの要素(I)($I=1\sim 100$)を読み出してベクトル・レジスタ $VR1$ に格納する。このとき格納される対象は、 $B(1)\sim B(100)$ の100要素であり $VR1(1)\sim VR1(100)$ に格納される。

【0008】

次に第2回目のベクトルロード命令が発行され、ベクトル・アクセス・パイプ111は $B(I+1)$ を読み出し、ベクトル・レジスタ $VR2$ に格納する。このとき格納される対象は $B(2)\sim B(101)$ の100要素であり、 $VR2(1)\sim VR2(100)$ にそれぞれ格納される。

【0009】

引続き、ベクトル・レジスタ $VR1$ と $VR2$ が読み出され、ベクトル演算器112でベクトル加算が実行され、結果がベクトル・レジスタ $VR3$ に格納される。そしてこの $VR3$ に格納された結果を主記憶100上の配列A(I)に格納して、このDOLoopの処理が終了する。

【0010】

一般にベクトル・レジスタは主記憶から遠く、ベクトル・アクセス・パイプの性能は低い。このため2回ベクトルロードを実行することは大きいオーバーヘッドとなり、性能低下の原因となる。

【0011】

したがって本発明の目的はこのようなアクセス回数を減少して演算速度の向上を図るベクトル処理装置を提供することである。

【0012】

【課題を解決するための手段】

前記目的を達成するための本発明の関連技術の構成を図1に示す。図1において、1は主記憶、2はベクトル部、3はスカラ部、10はベクトル・レジスタ部、11はベクトル・アクセス・パイプ、12はベクトル演算器、13はベクトル命令制御部である。ベクトル命令制御部13は、図1に示す如く、例えば配列レジスタ番号 $VR1$ に格納された要素を配列レジスタ番号 $VR2$ にスライドすることができる。前記本発明の目的は、下記の(1)～(4)に記載された構成により達成される。

【0013】

10

20

30

40

50

(1) 本発明のベクトル処理装置では、主記憶手段と、ベクトル・レジスタ手段と、ベクトル演算手段を具備し、ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向にシフトして前記ベクトル・レジスタとは別のレジスタの異なる要素番号区分に格納するベクトル処理装置において、前記ベクトル・レジスタ手段を複数のブロックにより構成し、通常のベクトル演算においては各ブロックが独立に動作し、ベクトル総和演算等の要素間の演算のために各ブロック間に専用データバスを設け、この専用データバスを用いてブロック間で要素の転送を行うとともに、ベクトル・アクセス・パイプとは独立に並行してベクトルスライドを実行するベクトルスライド手段を設け、ベクトル演算器と独立に、並行してベクトルスライド処理を実行することを特徴とする。

【0014】

(2) 本発明のベクトル処理装置では、前記(1)において、前記要素番号が1つ増加又は減少する方向に要素を格納することを特徴とする。

【0015】

(3) 本発明のベクトル処理装置では、主記憶手段と、ベクトル・レジスタ手段と、ベクトルマスクレジスタ手段と、ベクトル演算手段を具備するベクトル処理装置において、前記ベクトル・レジスタ手段を複数のブロックにより構成し、通常のベクトル演算においては各ブロックが独立に動作し、ベクトル総和演算等の要素間の演算のために各ブロック間に専用データバスを設け、この専用データバスを用いてブロック間で要素の転送を行うとともに、ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向にシフトして前記ベクトル・レジスタとは別のベクトル・レジスタの異なる要素番号区分に格納するとき、読み出し元のベクトル・レジスタの値と書き込み先のベクトル・レジスタの元の値をマージし、要素番号の方向に移動したときのベクトルマスクレジスタの対応するビットが「1」または「0」のいずれか一方の特定値である要素についてのみ格納を行うベクトル制御手段を具備したことを特徴とする。

【0016】

(4) 本発明のベクトル処理装置では、主記憶手段と、ベクトル・レジスタ手段と、ベクトル演算手段と、ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向にシフトして前記ベクトル・レジスタとは別のレジスタの異なる要素番号区分に格納するベクトル処理装置において、前記ベクトル・レジスタ手段を複数のブロックにより構成し、通常のベクトル演算においては各ブロックが独立に動作し、ベクトル総和演算等の要素間の演算のために各ブロック間に専用データバスを設け、この専用データバスを用いてブロック間で要素の転送を行うとともに、ベクトル・アクセス・パイプとは独立にベクトル演算器と並行してベクトルスライドを実行するベクトルスライド手段を設け、ベクトル演算器と独立に、並行してベクトルスライド処理を実行することを特徴とする。

【0017】

そしてこれにより次の如き作用効果を奏する。

(1) ベクトル・レジスタに保持されている要素を読み出し、これを要素番号の方向に移動してベクトル・レジスタの異なる要素番号区分に格納するベクトルスライド処理を行うことにより、ベクトルロード命令の回数が従来の2回から1回に削減され、高速ベクトル処理が実現できる。

【0018】

(2) マスクレジスタを使用する場合でもベクトルロード命令の回数が削減され、高速ベクトル処理が実現できる。

【0019】

(3) 要素番号が1つ増加又は減少する方向に要素をスライド処理により格納するので、1ワードずれた配列を1回のベクトルロード命令により可能となり、ベクトルロード命令の回数を削減し、高速ベクトル処理ができる。しかもスライド処理制御のためのハード機構が簡単になり、制御も容易なものとなる。

【0020】

(4) ベクトル・アクセス・パイプを使用せずにベクトル・アクセス・パイプとは独立

10

20

30

40

50

に並行してベクトルスライドを実行することができるので、メモリアクセスを行いながらスライド処理を行うことができ、高速ベクトル処理ができる。

【 0 0 2 1 】

(5) ベクトル・レジスタ手段が複数のブロックにより構成されているときでも、従来から設けられている専用データバスを用いて要素の転送が可能となり、転送用の特別のピンを設ける必要なく、ベクトルロード命令の回数を削減するとともに、高速ベクトル処理ができる。

【 0 0 2 2 】

(6) 専用のベクトルスライド手段を設けたので、ベクトル演算手段とは独立に並行してスライド処理を実行することができるので、スライド処理とベクトル演算とを同時に行うことができ、高速ベクトル処理ができる。

10

【 0 0 2 3 】

【 発明の実施の形態 】

本発明の関連技術を図 1 にもとづき説明する。図 1 において、1 は主記憶、2 はベクトル部、3 はスカラ部、10 はベクトル・レジスタ部、11 はベクトル・アクセス・パイプ、12 はベクトル演算器、13 はベクトル命令制御部である。

【 0 0 2 4 】

主記憶 1 は、ベクトル処理装置を動作する各種のデータが記憶されるものであって、ベクトル演算されるべき配列の各要素が格納されたり、演算結果が格納されたり、ベクトル部 2 が実行すべき命令等が格納されるものである。

20

【 0 0 2 5 】

ベクトル部 2 は、主記憶 1 に格納されたデータにベクトル演算を行い、その演算結果を主記憶 1 に格納するものであって、ベクトル・レジスタ部 10、ベクトル・アクセス・パイプ 11、ベクトル演算器 12、ベクトル命令制御部 13 等を具備している。

【 0 0 2 6 】

スカラ部 3 は主記憶 1 から命令を読み出し、これを解読してスカラ命令かベクトル命令かを識別し、スカラ命令であればこれを実行し、ベクトル命令であればこれをベクトル命令制御部 13 に送出する。

【 0 0 2 7 】

ベクトル・レジスタ部 10 は、ベクトル演算器 12 でベクトル演算されるべき配列要素及びベクトル演算結果得られた配列要素を一時保持するものであり、主記憶 1 から読み出された配列 B の各要素 B (1)、B (2) ・ ・ ・ B (1 0 1) が格納されるベクトル・レジスタ V R 1 と、ベクトル・レジスタ V R 1 の要素の一部 B (2)、B (3) ・ ・ ・ B (1 0 1) がスライド格納されるベクトル・レジスタ V R 2 と、ベクトル・レジスタ V R 1 とベクトル・レジスタ V R 2 との各要素のベクトル演算結果が格納されるベクトル・レジスタ V R 3 等を具備している。

30

【 0 0 2 8 】

ベクトル・アクセス・パイプ 11 は、主記憶 1 に対してベクトル要素をロードしたり、ストアするものであって、ロードの場合にはアクセス先のアドレス計算を行ったり、主記憶 1 に対してロード要求を発行し、これにより主記憶 1 から読み出された各要素を取り出してベクトル・レジスタ部 10 に送出する。またストアの場合はアクセス先のアドレス計算を行い、主記憶 1 に対しストア要求を発行し、ストアすべき各要素を主記憶 1 に送出してこれらをストアするものである。

40

【 0 0 2 9 】

ベクトル演算器 12 は、ベクトル・レジスタ V R 1 と V R 2 の同一の要素番号の要素をそれぞれ読み出して演算を行い、演算結果をベクトル・レジスタ V R 3 の同一要素番号の区分に格納処理を行ったり、本発明の特徴とするスライド処理を行うものである。

【 0 0 3 0 】

このスライド処理は、ベクトル・レジスタ V R 1 から要素 B (1)、B (2) ・ ・ ・ B (1 0 1) を読み出し、要素を、その番号が 1 つ減る方向の要素番号としてベクトル・レ

50

ジスタVR2に格納するものである。即ちベクトル・レジスタVR1の要素番号2の要素B(2)は、 $2 - 1 = 1$ つまりベクトル・レジスタVR2の要素番号1の要素として格納され、ベクトル・レジスタVR1の要素番号3の要素B(3)はベクトル・レジスタVR2の要素番号2の要素として格納され、ベクトル・レジスタVR1の要素番号101の要素B(101)はベクトル・レジスタVR2の要素番号100の要素として格納される。これにより図1に示すベクトル・レジスタVR1の要素が、矢印に示す如く、ベクトル・レジスタVR2の1つ減る方向に要素番号がずらして格納されることになる。

【0031】

なお、図1に示す例は、ベクトル・レジスタVR1から要素番号が1つ減る方向にずらしてベクトル・レジスタVR2に格納されるものであるが、スライド処理はこれに限定されるものではなく、スライドされる量は1のみに限定されるものではなく2以上の任意の整数が選択されるものであり、また方向も減る方向のみならず要素番号の増加する方向にスライドすることもできる。

10

【0032】

ベクトル命令制御部13は、スカラ部3からベクトル命令を受け取ったときこれを解読して、その内容に応じてベクトル・レジスタ10、ベクトル・アクセス・パイプ11、ベクトル演算器12等に対し選択的に制御指示を与えるものである。

【0033】

次に、図1に示す本発明の関連技術に対する動作を、前記D0ループ10の命令を実行する場合について説明する。

20

【0034】

図示省略したコンパイラが、前記D0ループ10の命令全体を認識することにより、これがベクトル命令であり、その演算に必要な要素は配列BのB(1)~B(101)であること、そしてこれらの要素B(1)~B(101)により前記(1)式~(100)式の演算を行うこと、またベクトル・レジスタVR2に格納する要素は、ベクトル・レジスタVR1に格納された要素を1つ減る方向にスライドしてVR2に格納すればよいことを認識し、これらを実行する命令即ち後述するステップ1~ステップ3を実行する命令を作成し、主記憶1に格納する。

【0035】

(ステップ1)要素B(I) $I = 1 \sim 101$ を主記憶1から取り出し、ベクトル・レジスタVR1に格納する。

30

【0036】

(ステップ2)ベクトル・レジスタVR1から要素B(1)~B(101)を読み出し、要素を要素番号が1つ減る方向にずらしてベクトル・レジスタVR2に格納する。従って、ベクトル・レジスタVR2の要素番号1にはB(2)が格納され、要素番号2にはB(3)が格納され・・・要素番号(I)にはB(I+1)が格納され・・・要素番号100にはB(101)が格納される。これにより主記憶1から要素B(2)~B(101)をVR2に格納した場合と同じ結果が得られる。

【0037】

(ステップ3)ベクトル・レジスタVR1とVR2をベクトル加算($I = 1 \sim 100$)し、演算結果をベクトル・レジスタVR3に格納する。

40

【0038】

従って、コンパイラが作成した命令をスカラ部3が主記憶1から取り出し、これがベクトル命令であることが解読されると、これらの命令はベクトル命令制御部13に送出される。

【0039】

まず前記ステップ1を実行するベクトルロードがベクトル命令制御部13からベクトル・アクセス・パイプ11に送出され、ベクトル・アクセス・パイプ11が主記憶1から配列Bの要素B(I)、($I = 1 \sim 101$)を読み出してベクトル・レジスタVR1に格納する。このとき格納される対象はB(1)~B(101)の101要素であり、ベクトル

50

・レジスタVR1(1)~VR1(101)に格納される。

【0040】

次にベクトル命令制御部13は、前記ステップ2を実行するベクトルスライド命令をベクトル演算器12に送出する。これによりベクトル演算器12は、ベクトル・レジスタVR1から要素B(1)~B(101)を読み出し、その要素番号が1つ減る方向にずらしてベクトル・レジスタVR2に格納する。

【0041】

これにより、ベクトル・レジスタVR1の要素番号2に格納されていた要素B(2)はベクトル・レジスタVR2の要素番号1に格納され、VR1の要素番号3に格納されていた要素B(3)はVR2の要素番号2に格納され、VR1の要素番号101に格納されていた要素B(101)はVR2の要素番号100に格納される。即ち、VR1の要素番号Iに格納されていた要素B(I)は、VR2の要素番号(I-1)に格納される。かくしてVR1に格納された各データは1要素分上位方向にシフトしてVR2に格納される。

10

【0042】

このようにして、図1に示す如く、VR1の要素B(2)~B(101)がVR2にスライドされることになる。なお、VR1の要素番号1に格納されていた要素B(1)は、最初にVR1から読み出されたとき、そのスライド先がVR2には存在しないので、この読み出された要素B(1)は捨てられることになるが、VR1にはそのまま要素B(1)が残っているので、演算には何等の影響もない。

【0043】

20

引き続き、前記ステップ3を実行する命令がベクトル命令制御部13からベクトル演算器12に送出される。これによりベクトル・レジスタVR1とVR2が読み出されてベクトル加算が実行され、その演算結果がベクトル・レジスタVR3に格納される。

【0044】

それからベクトル命令制御部13は、ベクトル・アクセス・パイプ11に対し演算結果格納命令を送出する。これによりベクトル・アクセス・パイプ11は、ベクトル・レジスタVR3に格納された演算結果を主記憶1上の配列A(I)に格納し、これにより前記D0ループ10のD0ループの処理が終了する。

【0045】

このようにして、本発明の関連技術では、図5に説明した従来例ではベクトルロード命令が2回発行されたのに比べて、その実行に長時間を必要とするベクトルロード命令の回数が1回済み、1回だけその回数を少なくすることができるため、オーバヘッドが小さく、高い性能を実現することができる。

30

【0046】

図2に本発明の第1の実施の形態を説明する。図2において図1と同記号は同一部分を示し、14はベクトルスライド部である。

【0047】

ベクトルスライド部14は、ベクトル命令制御部13からの指示にもとづき、前記ステップ2等で説明したベクトルスライド処理を行うものである。このベクトルスライド部14は、ベクトル演算器12とは独立に、並行してベクトルスライド処理を実行することができる。

40

【0048】

従って、ベクトル演算器12で演算を行うことと並行してベクトルスライドを実行することができるので、ベクトル演算を高性能に行うことができる。

【0049】

本発明の第2の実施の形態を図3にもとづき説明する。

ベクトル処理装置は、一般に複数のベクトル・レジスタ部と、複数のベクトル演算部で構成される。図3において、他図と同一記号は同一部を示し、ベクトル処理装置は、複数のベクトル・レジスタ・演算器ブロック(以下ブロックという)21、22、23、24で構成される。これらの各ブロックは同一構成であり、ブロック22に代表的に例示され

50

るように、ベクトル・レジスタ部 10 とベクトル演算器 12 を具備している。

【0050】

ベクトル・レジスタ部 10 は、図 1 に示す如く、複数のベクトル・レジスタ VR 1、VR 2・・・を具備し、ベクトル演算器 12 は、ベクトル加算器 12 - 1、ベクトル乗算器 12 - 2、ベクトル除算器 12 - 3、全ブロックのベクトル・レジスタの要素の総和を求める総和演算器 12 - 4、前記の如きスライド処理を行うスライド機構 12 - 5 等を具備している。

【0051】

各ブロック 22 に例示されるように、ブロック毎にベクトル・レジスタ部 10 とベクトル演算器 12 とは接続され、通常のベクトル演算においては、各ブロック毎に並行して独立に動作可能であり、ブロック間のデータ移動は通常発生しない。

10

【0052】

ところでベクトル処理装置では、ベクトル総和演算等の各ブロックの要素間の演算を必要とする一部の特殊な命令を実行するとき、ブロック間をデータ移動するための、総和演算用データバスと称される特別なデータバス B が設けられている。

【0053】

またベクトル・レジスタの要素番号も、例えば 1 ~ 100 はブロック 21 に、101 ~ 200 はブロック 22 というように、同一ブロック内に連続して配置すると、特定のブロックのみ動作状態となって負荷が集中する欠点が存在するのでブロック 21、22、23、24 に要素番号 1、2、3、4 を配置するという、いわゆるインタリーブ方式が採用される。

20

【0054】

したがって前記スライド処理を行うとき、要素がブロック間にデータ移動することが発生する。このスライド処理専用のバスを設けることはそれだけピン数が多くなるので好ましいことではない。本発明では、このため、ベクトルスライド命令の場合も、総和演算等のためのデータバス B を共用するように構成したので、ブロック間のデータバスのサイズを増加することなく、ベクトルスライド命令を実現するハードウェアを構成することが可能となった。

【0055】

本発明の第 3 の実施の形態を図 4 により説明する。前記の如くベクトルスライド命令により、ベクトル・レジスタ VR 1 の要素が読み出されて要素番号が、例えば 1 つ減少する方向にスライドされる。即ち VR 1 (I + 1) の要素を VR 2 (I) に書き込む。第 3 の実施の形態では、マスクレジスタ MR 3 を設け、このとき対応するマスクレジスタ MR 3 (I) の値が例えば「1」の要素にはスライドした書き込みを行うが、「0」の要素には書き込みを行わない。したがって書き込みの行われなかった要素番号には、旧データがそのまま保持されるものとなる。

30

このようにして、スライド処理のときにマスク制御を行うことが可能となる。

【0056】

本発明の第 4 の実施の形態を説明する。第 4 の実施の形態では、ベクトル・アクセス・パイプを使用せず、ベクトル・アクセス・パイプとは独立に並行してベクトルスライド処理を実行するものである。これにより主記憶に対するメモリアクセスを行いながらスライド処理を行うことができ、高速ベクトル処理が可能となる。

40

【0057】

前記説明ではスライド量を要素番号が 1 つ減る方向にした場合について説明したが、本発明は勿論これに限定されるものではなく、増える方向にスライドしてもよく、スライド量も 2 以上にすることができる。

【0058】

【発明の効果】

本発明によれば主記憶からの複数回のデータの読み出しを一回のベクトルロード命令の実行により行うことができるので、従来に比較してその実行に時間がかゝるベクトルロード

50

命令の回数を減少することができ、ベクトルロードのオーバーヘッドが削減され高速なベクトル処理が実現できる。

【図面の簡単な説明】

【図 1】本発明の関連技術である。

【図 2】本発明の第 1 の実施の形態である。

【図 3】本発明の第 2 の実施の形態である。

【図 4】本発明の第 3 の実施の形態である。

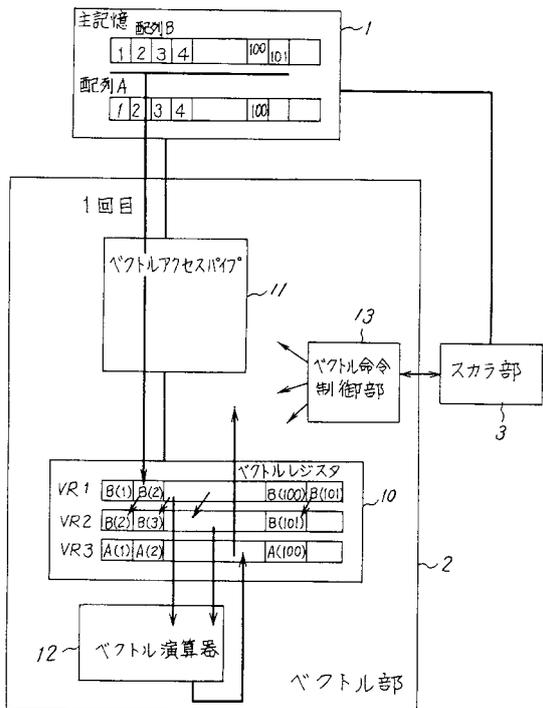
【図 5】従来例である。

【符号の説明】

- 1 主記憶
- 2 ベクトル部
- 3 スカラ部
- 10 ベクトル・レジスタ部
- 11 ベクトル・アクセス・パイプ
- 12 ベクトル演算器
- 13 ベクトル命令制御部

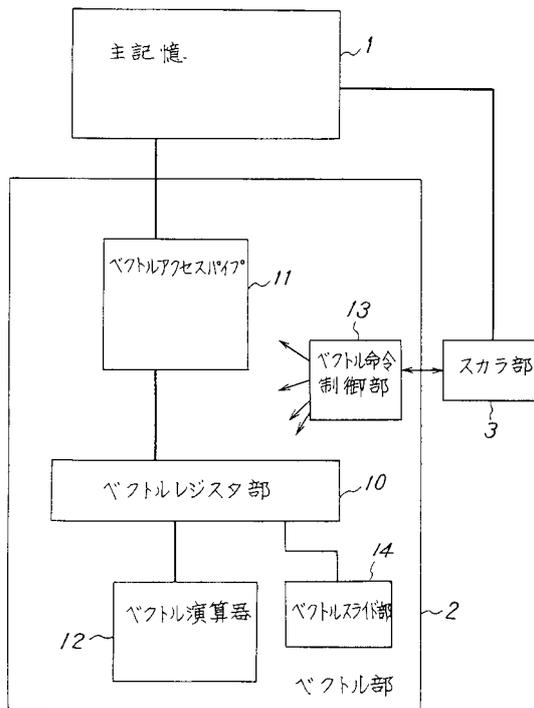
【図 1】

本発明の関連技術



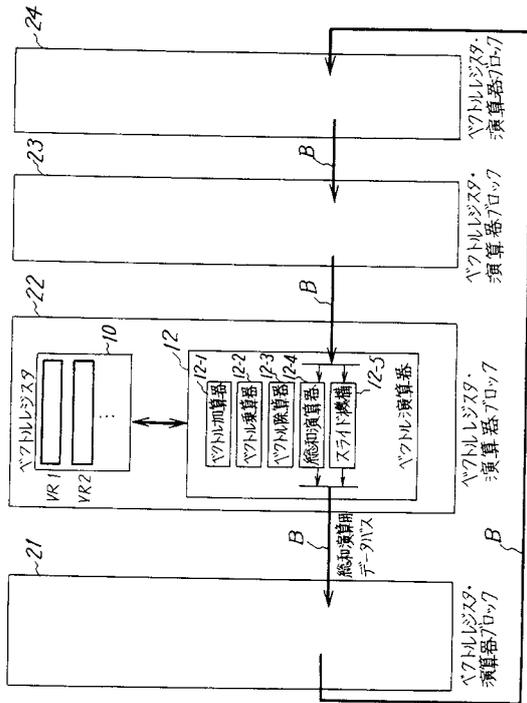
【図 2】

本発明の第 1 の実施の形態



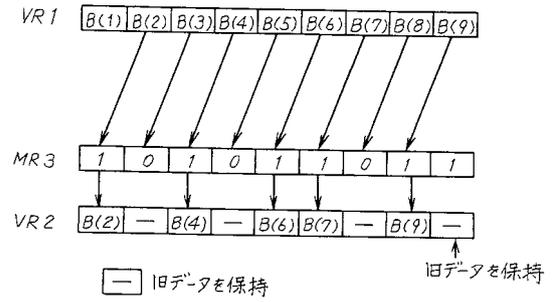
【 図 3 】

本発明の第 2 の実施の形態



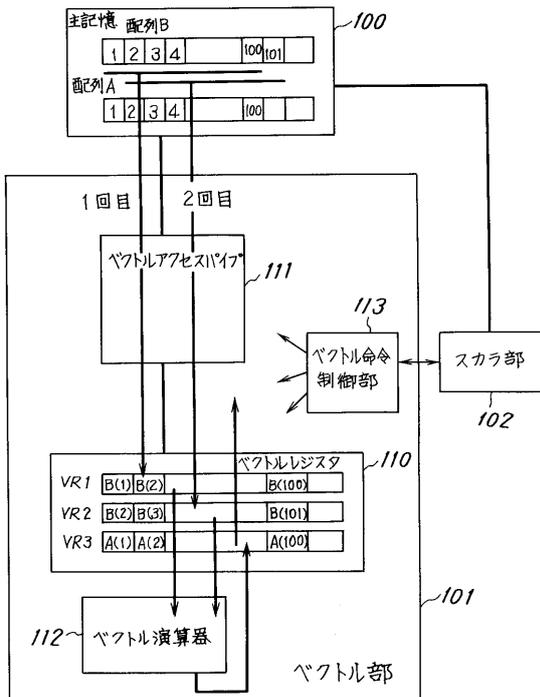
【 図 4 】

本発明の第 3 の実施の形態



【 図 5 】

従来例



フロントページの続き

審査官 鳥居 稔

- (56)参考文献 特開平09 - 034878 (JP, A)
特開昭61 - 100862 (JP, A)
特開昭63 - 153668 (JP, A)

- (58)調査した分野(Int.Cl. , DB名)
G06F 17/16
G06F 12/06