(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3691538号

(P3691538)

(45)発行日　平成17年9月7日(2005.9.7)　(24)登録日　平成17年6月24日(2005.6.24)

(51)Int.Cl.[7]	F I		
G06F 17/16	G06F	17/16	K
G06F 7/50	G06F	7/50	Z
G06F 7/52	G06F	7/52	310A

請求項の数 2　(全 14 頁)

(21)出願番号　特願平7-46763
(22)出願日　平成7年3月7日(1995.3.7)
(65)公開番号　特開平8-241302
(43)公開日　平成8年9月17日(1996.9.17)
審査請求日　平成12年12月27日(2000.12.27)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100095072
弁理士　岡田　光由
(74)代理人　100074848
弁理士　森田　寛
(72)発明者　黒田　浩二
神奈川県川崎市中原区上小田中１０１５番地　富士通株式会社内

審査官　鳥居　稔

最終頁に続く

(54)【発明の名称】ベクトルデータ加算方法及びベクトルデータ乗算方法

(57)【特許請求の範囲】

【請求項１】

ベクトルレジスタに格納されるｍビットのデータと、別のベクトルレジスタに格納されるｍビットのデータと、マスクレジスタに格納されるキャリーアウトデータとを入力とし、加算値となるｍビットデータを算出して加算結果用のベクトルレジスタに格納するとともに、その加算結果により生ずるキャリーアウトデータを別のマスクレジスタに格納するという機能を有する加算器を使って、被加数と加数との加算処理を実行するベクトルデータ加算方法であって、

（ｉ）ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとマスクレジスタｍｒ１に格納されるキャリーアウトデータとの加算結果をベクトルレジスタｖｒ３に格納するとともに、そのとき発生するキャリーアウトデータをマスクレジスタｍｒ２に格納しろというベクトル加算命令を、

「ＶＡＣ　ｖｒ１，ｖｒ２，ｍｒ１，ｖｒ３，ｍｒ２」

と表すならば、このベクトル加算命令の命令列で表される被加数と加数との加算命令を受け取り、

（ii）上記加算器を使って、上記命令列のベクトル加算命令をその命令列の順番に従って実行することで、被加数と加数とを加算することを、

特徴とするベクトルデータ加算方法。

【請求項２】

ベクトルレジスタに格納されるｍビットのデータと、別のベクトルレジスタに格納され

るｍビットのデータとを入力とし、乗算値となる２ｍビットデータを算出して、該乗算値の上位ｍビットデータか下位ｍビットデータのいずれか一方を選択して乗算結果用のベクトルレジスタに格納するという機能を有する乗算器と、ベクトルレジスタに格納されるｍビットのデータと、別のベクトルレジスタに格納されるｍビットのデータと、マスクレジスタに格納されるキャリーアウトデータとを入力とし、加算値となるｍビットデータを算出して加算結果用のベクトルレジスタに格納するとともに、その加算結果により生ずるキャリーアウトデータを別のマスクレジスタに格納するという機能を有する加算器とを使って、被乗数と乗数との乗算処理を実行するベクトルデータ乗算方法であって、

（ｉ）ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとの乗算結果の下位ｍビットデータ部分をベクトルレジスタｖｒ３に格納しろというベクトル乗算命令を、

「ＶＭＬ　ｖｒ１，ｖｒ２，ｖｒ３」

と表し、ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとの乗算結果の上位ｍビットデータ部分をベクトルレジスタｖｒ３に格納しろというベクトル乗算命令を、

「ＶＭＵ　ｖｒ１，ｖｒ２，ｖｒ３」

と表し、ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとマスクレジスタｍｒ１に格納されるキャリーアウトデータとの加算結果をベクトルレジスタｖｒ３に格納するとともに、そのとき発生するキャリーアウトデータをマスクレジスタｍｒ２に格納しろというベクトル加算命令を、

「ＶＡＣ　ｖｒ１，ｖｒ２，ｍｒ１，ｖｒ３，ｍｒ２」

と表すならば、第１番目にＶＭＬ命令、それに続いてＶＭＵ命令、それに続いてＶＭＬ命令、それに続いてＶＡＣ命令、それに続いてＶＭＬ命令、それに続いてＶＡＣ命令、それに続いてＶＭＵ命令、それに続いてＶＭＵ命令、それに続いてＶＡＣ命令、それに続いてＶＭＬ命令、それに続いてＶＡＣ命令、それに続いてＶＭＵ命令、最後にＶＡＣ命令という命令列で表される被乗数と乗数との乗算命令を受け取り、

（ii）上記命令列の乗算命令をその命令列の順番に従って実行するとともに、その実行にあたって、上記ＶＭＬ命令及び上記ＶＭＵ命令を実行するときは、上記乗算器を使ってその命令を実行し、上記ＶＡＣ命令を実行するときは、上記加算器を使ってその命令を実行することを、

特徴とするベクトルデータ乗算方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】

本発明は、ベクトル加算処理を高速に実行できるようにするベクトルデータ加算方法と、そのベクトルデータ加算方法を使って、ベクトル乗算処理を高速に実行できるようにするベクトルデータ乗算方法とに関する。

【０００２】

ベクトル処理装置では、ベクトル加算処理やベクトル乗算処理を実行する。このようなベクトル演算処理は高速に実行できるようにする必要がある。

【０００３】

【従来の技術】

従来のベクトル処理装置では、ベクトル加算処理を実行するときには、桁上げの発生を考慮して、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採っていた。

【０００４】

次に、１６０ビットの被加数と１６０ビットの加数との加算処理を例にとって、この従来技術を詳細に説明する。

加算器が６４ビット同士の加算処理を実行する場合には、従来では、図１３に示すように、６４ビットの３つのレジスタ（ｖｒ00，ｖｒ01，ｖｒ02）からなる被加数用のレジスタ

と、６４ビットの３つのレジスタ（ｖｒ03，ｖｒ04，ｖｒ05）からなる加数用のレジスタとを用意して、例えば、図１４に示す形式、すなわち、図１５に図式化する形式に従って、その被加数用のレジスタに１６０ビットの被加数を格納するとともに、加数用のレジスタに１６０ビットの加数を格納する。

【０００５】

そして、図１６に示すベクトル命令列を発行することで、１６０ビットの被加数と１６０ビットの加数との加算処理を実行する。ここで、

「ＶＡ　　ｖｒ１，ｖｒ２，ｖｒ３」

は、ベクトルレジスタｖｒ１とベクトルレジスタｖｒ２との加算結果をベクトルレジスタｖｒ３に格納しろというベクトル加算命令であり、

「ＶＳＲ　　ｖｒ１，ＳＣ，ｖｒ３」

は、ベクトルレジスタｖｒ１のデータをＳＣビット右シフトしてベクトルレジスタｖｒ３に格納しろというベクトルシフト命令であり、

「ＶＳＬ　　ｖｒ１，ＳＣ，ｖｒ３」

は、ベクトルレジスタｖｒ１のデータをＳＣビット左シフトしてベクトルレジスタｖｒ３に格納しろというベクトルシフト命令である。

【０００６】

すなわち、図１６に示すベクトル命令列に従い、先ず最初に、(1) のベクトル加算命令ＶＡに従って、ベクトルレジスタｖｒ02の被加数部分と、ベクトルレジスタｖｒ05の加数部分とを加算してベクトルレジスタｖｒ10に格納する。このとき、桁上げが発生する可能性があるので、続いて、(2) のベクトルシフト命令ＶＳＲに従って、ベクトルレジスタ10の格納データを６０ビット右シフトすることでその桁上げ値（キャリーアウトデータ）を取り出して、それをベクトルレジスタ15に格納する。

【０００７】

続いて、(3) のベクトル加算命令ＶＡに従って、ベクトルレジスタｖｒ01の被加数部分と、ベクトルレジスタｖｒ04の加数部分とを加算してベクトルレジスタｖｒ20に格納する。

【０００８】

続いて、(4) のベクトル加算命令ＶＡに従って、下位部分の加算処理により発生したキャリーアウトデータを加算すべく、ベクトルレジスタｖｒ15の格納するキャリーアウトデータと、ベクトルレジスタｖｒ20の格納データとを加算してベクトルレジスタｖｒ20に格納する。このとき、桁上げが発生する可能性があるので、続いて、(5) のベクトルシフト命令ＶＳＲに従って、ベクトルレジスタ20の格納データを６０ビット右シフトすることでそのキャリーアウトデータを取り出して、それをベクトルレジスタｖｒ25に格納する。

【０００９】

続いて、(6) のベクトル加算命令ＶＡに従って、ベクトルレジスタｖｒ00の被加数部分と、ベクトルレジスタｖｒ03の加数部分とを加算してベクトルレジスタｖｒ30に格納する。

【００１０】

続いて、(7) のベクトル加算命令ＶＡに従って、下位部分の加算処理により発生したキャリーアウトデータを加算すべく、ベクトルレジスタｖｒ25の格納するキャリーアウトデータと、ベクトルレジスタｖｒ30の格納データとを加算してベクトルレジスタｖｒ6 に格納する。

【００１１】

続いて、ベクトルレジスタｖｒ10に格納される６０ビットの有効データを取り出すべく、(8) のベクトルシフト命令ＶＳＬに従って、ベクトルレジスタ10の格納データを４ビット左シフトして、それをベクトルレジスタｖｒ10に格納し、(9) のベクトルシフト命令ＶＳＲに従って、そのベクトルレジスタｖｒ10の格納データを４ビット右シフトしてベクトルレジスタｖｒ8 に格納することで、上位４ビットにゼロ値を持つその６０ビットの有効データをベクトルレジスタｖｒ8 に格納する。

【００１２】

続いて、ベクトルレジスタｖｒ20に格納される６０ビットの有効データを取り出すべく、

(10)のベクトルシフト命令ＶＳＬに従って、ベクトルレジスタ20の格納データを４ビット左シフトして、それをベクトルレジスタｖｒ20に格納し、(11)のベクトルシフト命令ＶＳＲに従って、そのベクトルレジスタｖｒ20の格納データを４ビット右シフトしてベクトルレジスタｖｒ7 に格納することで、上位４ビットにゼロ値を持つその６０ビットの有効データをベクトルレジスタｖｒ7 に格納する。

【００１３】

このように、従来のベクトル処理装置では、ベクトル加算処理を実行するときには、桁上げの発生を考慮して、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採っていたのである。

【００１４】

一方、従来のベクトル処理装置の備える乗算器では、６４ビット×６４ビットのような入力仕様を持つ場合にあっても、ハードウェア量の削減を図るために、６４ビット×１６ビットのような少ないビット数の乗算機能を持つ構成を採っていた。

【００１５】

そして、入力仕様の６４ビット同士の乗算処理を実現するために、乗数を１６ビット単位に４分割し、６４ビット×１６ビット乗算機能を使って、それらの１６ビットの乗数部分と６４ビットの被乗数とを乗算することで部分積を求め、それらの部分積を１６ビットシフトしつつ加算して、その加算結果の示す６４ビット部分を乗算結果として出力するという構成を採っていた。

【００１６】

例えば、６４ビット同士の乗算処理の上位６４ビットが必要となる場合には、命令指示に従って、図１７に示すように、４分割した乗数を下位側から順番に選択して部分積を求め、それらの部分積を１６ビット左シフトしつつ加算していくことで乗算処理の上位６４ビットを得て出力していた。また、６４ビット同士の乗算処理の下位６４ビットが必要となる場合には、命令指示に従って、４分割した乗数を上位側から順番に選択して部分積を求め、それらの部分積を１６ビット右シフトしつつ加算していくことで乗算処理の下位６４ビットを得て出力していた。

【００１７】

【発明が解決しようとする課題】

しかしながら、従来技術のように、ベクトル加算処理を実行するときに、桁上げの発生を考慮して、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採っていると、命令数が多くなることから高速にベクトル加算処理を実行できないという問題点があった。

【００１８】

また、従来技術の乗算器では、内部で実行する乗算回数が多くなるとともに、内部に、部分積をシフトし加算する機能（ループ構成を使用している）を持たなくてはならないという問題点があった。

【００１９】

本発明はかかる事情に鑑みてなされたものであって、ベクトル加算処理を高速に実行できるようにするベクトルデータ加算方法の提供と、そのベクトルデータ加算方法を使って、ベクトル乗算処理を高速に実行できるようにするベクトルデータ乗算方法の提供とを目的とする。

【００２０】

【課題を解決するための手段】

図１に本発明の原理構成を図示する。

図中、１は本発明を実行するベクトル処理装置であって、ＣＰＵ１０と、ベクトル命令制御機構１１と、ベクトルレジスタ１２と、マスクレジスタ１３と、加算器１４と、乗算器１５とを備える。

【００２１】

このＣＰＵ１０は、ベクトル命令を発行する。ベクトル命令制御機構１１は、ベクトル命

令の実行を制御する。ベクトルレジスタ１２は、ベクトルデータを格納する。マスクレジスタ１３は、ベクトル処理で使用するマスクデータを格納する。加算器１４は、ベクトル加算命令を実行する。乗算器１５は、ベクトル乗算命令を実行する。

【００２２】

本発明を実現するために、加算器１４は、ベクトルオペランドの加数と被加数の他に、マスクレジスタ１３に格納されるキャリーアウトデータを入力するとともに、算出結果のキャリーアウトデータをマスクレジスタ１３へ出力する構成を採る。このとき、マスクオペランドの増加により命令で指定するレジスタ数が増える場合には、命令で指定するレジスタ数を抑えるべく、入力用のマスクレジスタと出力用のマスクレジスタとして同一のものを使用する構成を採ることが好ましい。

【００２３】

一方、本発明を実現するために、乗算器１５は、入力される２つのｍビットデータの乗算値となる２ｍビットデータを算出する機能を有するとともに、命令に応答して、乗算値の上位ｍビットデータか下位ｍビットデータのいずれか一方を選択して出力するセレクタを持つ構成を採る。

【００２６】

【作用】

本発明で用いる加算器１４は、ベクトルオペランドの被加数と加数の他に、マスクレジスタ１３に書き込まれるキャリーアウトデータを入力として加算処理を実行して、その加算結果により生ずるキャリーアウトデータをマスクレジスタ１３に書き込んでいく。

【００２７】

このように構成される加算器１４を使い、本発明を実行するベクトル処理装置１では、

（ｉ）ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとマスクレジスタｍｒ１に格納されるキャリーアウトデータとの加算結果をベクトルレジスタｖｒ３に格納するとともに、そのとき発生するキャリーアウトデータをマスクレジスタｍｒ２に格納しろというベクトル加算命令を、

「ＶＡＣ　ｖｒ１，ｖｒ２，ｍｒ１，ｖｒ３，ｍｒ２」

と表すならば、このベクトル加算命令の命令列で表される被加数と加数との加算命令を受け取り、

（ii）上述の構成を採る加算器１４を使って、この命令列のベクトル加算命令をその命令列の順番に従って実行することで、被加数と加数とを加算する。

このように、本発明によれば、キャリーアウトデータをマスクレジスタ１３に格納していく構成を採ることから、従来技術のように、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採る必要がなくなり、少ない命令数でもって高速にベクトル加算命令を実行できるようになる。

【００２８】

また、本発明で用いる乗算器１５は、入力される２つのｍビットデータの乗算値となる２ｍビットデータを算出する機能を有する。すなわち、従来の乗算器では、部分積を求め、それらをシフトしつつ加算することで、入力される２つのｍビットデータの乗算値となるｍビットデータを算出する構成を採っているのに対して、本発明で用いる乗算器１５では、部分積を求めることなく、直接、乗算値となる２ｍビットデータを算出する構成を採っている。例えば、６４ビットのデータと、６４ビットのデータとを乗算して、１２８ビットの乗算結果のデータを算出するのである。

【００２９】

これから、従来の乗算器では、内部で実行する乗算回数が多くなるとともに、内部に、部分積をシフトし加算する機能を持たなくてはならないという問題点があったが、本発明で用いる乗算器１５では、これを解決できることになる。

【００３０】

しかるに、乗算器１５に入力されるデータがｍビット構成であるときには、加算器１４に入力されるデータもｍビット構成を採るので、乗算器１５が２ｍビットのデータを出力

したのでは整合性を保てない。これから、本発明で用いる乗算器１５では、命令に応答して、乗算値の上位ｍビットデータか下位ｍビットデータのいずれか一方を選択して出力するセレクタを持つ構成を採ることで、これに対処している。

このように構成される乗算器１５及び加算器１４を使い、本発明を実行するベクトル処理装置１では、

（ｉ）ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとの乗算結果の下位ｍビットデータ部分をベクトルレジスタｖｒ３に格納しろというベクトル乗算命令を、

「ＶＭＬ　ｖｒ１，ｖｒ２，ｖｒ３」

と表し、ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとの乗算結果の上位ｍビットデータ部分をベクトルレジスタｖｒ３に格納しろというベクトル乗算命令を、

「ＶＭＵ　ｖｒ１，ｖｒ２，ｖｒ３」

と表し、ベクトルレジスタｖｒ１に格納されるｍビットデータとベクトルレジスタｖｒ２に格納されるｍビットデータとマスクレジスタｍｒ１に格納されるキャリーアウトデータとの加算結果をベクトルレジスタｖｒ３に格納するとともに、そのとき発生するキャリーアウトデータをマスクレジスタｍｒ２に格納しろというベクトル加算命令を、

「ＶＡＣ　ｖｒ１，ｖｒ２，ｍｒ１，ｖｒ３，ｍｒ２」

と表すならば、第１番目にＶＭＬ命令、それに続いてＶＭＵ命令、それに続いてＶＭＬ命令、それに続いてＶＡＣ命令、それに続いてＶＭＬ命令、それに続いてＶＡＣ命令、それに続いてＶＭＵ命令、それに続いてＶＭＵ命令、それに続いてＶＡＣ命令、それに続いてＶＭＬ命令、それに続いてＶＡＣ命令、それに続いてＶＭＵ命令、最後にＶＡＣ命令という命令列で表される被乗数と乗数との乗算命令を受け取り、

（ii）この命令列の乗算命令をその命令列の順番に従って実行するとともに、その実行にあたって、ＶＭＬ命令及びＶＭＵ命令を実行するときは、上述の構成を採る乗算器１５を使ってその命令を実行し、ＶＡＣ命令を実行するときは、上述の構成を採る加算器１４を使ってその命令を実行する。

このように、本発明によれば、ベクトル乗算命令を実行するにあたって実行が要求されることになるベクトル加算命令の実行にあたって、キャリーアウトデータをマスクレジスタ１３に格納していく構成を採ることから、従来技術のように、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採る必要がなくなることで、少ない命令数でもって高速にベクトル加算命令を実行できるようになり、これにより、少ない命令数でもって高速にベクトル乗算命令を実行できるようになる。

【００３１】

【実施例】

以下、実施例に従って本発明を詳細に説明する。

図１で説明したように、本発明で用いる加算器１４は、ベクトルオペランドの被加数と加数の他に、マスクレジスタ１３に書き込まれるキャリーアウトデータを入力として加算処理を実行して、その加算結果により生ずるキャリーアウトデータをマスクレジスタ１３に書き込む構成を採っている。

【００３２】

すなわち、図２に示すように、ベクトルレジスタ１２から読み込む被加数と、ベクトルレジスタ１２から読み込む加数と、マスクレジスタ１３から読み込むキャリーアウトデータとを入力して加算処理を実行して、その加算値をベクトルレジスタ１２に書き込むとともに、その加算処理により生じたキャリーアウトデータをマスクレジスタ１３に書き込む構成を採るのである。

【００３３】

マスクレジスタ１３から読み込むキャリーアウトデータは、１ビットのデータであることから、この加算処理は簡単なハードウェア構成により実現できることになる。

【００３４】

一方、図１で説明したように、本発明で用いる乗算器１５は、入力される２つのｍビットデータの乗算値となる２ｍビットデータを算出する機能を有するとともに、命令に応答して、乗算値の上位ｍビットデータか下位ｍビットデータのいずれか一方を選択して出力するセレクタを持つ構成を採る。

【００３５】

すなわち、図３に示すように、ベクトルレジスタ１２から読み込むｍビットの被乗数と、ベクトルレジスタ１２から読み込むｍビットの乗数とを入力として乗算処理を実行して、その乗算値の２ｍビットのデータをラッチし、命令に応答して、その乗算値の上位ｍビットデータか下位ｍビットデータのいずれか一方を選択して出力するセレクタを持つ構成を採るのである。

【００３６】

次に、上述の構成を採る加算器１４を用いて実行される本発明によるベクトル加算処理について、１６０ビットの被加数と１６０ビットの加数との加算処理を例にして説明する。

加算器１４が６４ビット同士の加算処理を実行する場合には、図１３で示したように、６４ビットの３つのレジスタ（ｖｒ00，ｖｒ01，ｖｒ02）からなる被加数用のレジスタと、６４ビットの３つのレジスタ（ｖｒ03，ｖｒ04，ｖｒ05）からなる加数用のレジスタとを用意して、図４に示す形式、すなわち、図５に図式化する形式に従って、その被加数用のレジスタに１６０ビットの被加数を格納するとともに、加数用のレジスタに１６０ビットの加数を格納する。

【００３７】

そして、図６に示すベクトル命令列を発行することで、１６０ビットの被加数と１６０ビットの加数との加算処理を実行する。ここで、

「ＶＡＣ　ｖｒ１，ｖｒ２，ｍｒ１，ｖｒ３，ｍｒ２」

は、ベクトルレジスタｖｒ１とベクトルレジスタｖｒ２とマスクレジスタｍｒ１との加算結果をベクトルレジスタｖｒ３に格納するとともに、そのとき発生するキャリーアウトデータをマスクレジスタｍｒ２に格納しろというベクトル加算命令である。

【００３８】

すなわち、図６に示すベクトル命令列に従い、先ず最初に、(1) のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ02の被加数部分と、ベクトルレジスタｖｒ05の加数部分と、初期値としてゼロ値を格納するマスクレジスタｍｒ00の格納データとを加算してベクトルレジスタｖｒ08に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ01に格納する。

【００３９】

続いて、(2) のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ01の被加数部分と、ベクトルレジスタｖｒ04の加数部分と、マスクレジスタｍｒ01に格納されるキャリーアウトデータとを加算してベクトルレジスタｖｒ07に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ02に格納する。

【００４０】

最後に、(3) のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ00の被加数部分と、ベクトルレジスタｖｒ03の加数部分と、マスクレジスタｍｒ02に格納されるキャリーアウトデータとを加算してベクトルレジスタｖｒ06に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ00に格納する。

【００４１】

このように、上述の構成を採る加算器１４を用いて実行される本発明によるベクトル加算処理では、図７に示すように、マスクレジスタｍｒ00,01,02を使いつつ、３個のベクトル加算命令を発行することで、１６０ビットの被加数と１６０ビットの加数との加算値を算出できることになる。これに対して、従来技術に従っていると、図１３で説明したように、１１個のベクトル加算命令／ベクトルシフト命令を発行しなければならない。

【００４２】

次に、上述の構成を採る乗算器１５を用いて実行される本発明によるベクトル乗算処理について説明する。

図８に示すように、４倍精度データでは、１１２ビットの仮数を持っている。これから、４倍精度の乗算処理では、乗算結果の仮数を求めるために、図９に示すオペランドの乗算処理を実行する必要がある。

【００４３】

これから、上述の構成を採る乗算器１５を用いて実行される本発明によるベクトル乗算処理について、１２８ビットの被乗数と１２８ビットの乗数との乗算処理を例にして説明する。

【００４４】

乗算器１５が６４ビット同士の乗算処理を実行する場合には、図１０に示すように、１２８ビットの被乗数用のレジスタ（上位６４ビット部分を“０１”、下位６４ビット部分を“０２”で表してある）と、１２８ビットの乗数用のレジスタ（上位６４ビット部分を“０３”、下位６４ビット部分を“０４”で表してある）とを用意して、その被乗数レジスタに１２８ビットの被乗数（上位６４ビット部分をＡ１、下位６４ビット部分をＡ２で表してある）を格納するとともに、その乗数レジスタに１２８ビットの乗数（上位６４ビット部分をＢ１、下位６４ビット部分をＢ２で表してある）を格納する。

【００４５】

そして、図１１に示すベクトル命令列を発行することで、図１２に図式化する乗算過程に従いつつ、１２８ビットの被乗数と１２８ビットの乗数との乗算処理を実行する。ここで、

「ＶＭＬ　　ｖｒ１，ｖｒ２，ｖｒ３」

は、ベクトルレジスタｖｒ１とベクトルレジスタｖｒ２との乗算結果の下位６４ビットをベクトルレジスタｖｒ３に格納しろというベクトル乗算命令であり、

「ＶＭＵ　　ｖｒ１，ｖｒ２，ｖｒ３」

は、ベクトルレジスタｖｒ１とベクトルレジスタｖｒ２との乗算結果の上位６４ビットをベクトルレジスタｖｒ３に格納しろというベクトル乗算命令であり、

「ＶＡＣ　　ｖｒ１，ｖｒ２，ｍｒ１，ｖｒ３，ｍｒ２」

は、ベクトルレジスタｖｒ１とベクトルレジスタｖｒ２とマスクレジスタｍｒ１との加算結果をベクトルレジスタｖｒ３に格納するとともに、そのとき発生するキャリーアウトデータをマスクレジスタｍｒ２に格納しろというベクトル加算命令である。

【００４６】

すなわち、図１１に示すベクトル命令列に従い、先ず最初に、(1) のベクトル乗算命令ＶＭＬに従って、ベクトルレジスタｖｒ02の被乗数部分Ａ２と、ベクトルレジスタｖｒ04の乗数部分Ｂ２とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の下位６４ビットのＡ２Ｂ２Ｌをベクトルレジスタｖｒ23に格納する。

【００４７】

続いて、(2) のベクトル乗算命令ＶＭＵに従って、ベクトルレジスタｖｒ02の被乗数部分Ａ２と、ベクトルレジスタｖｒ04の乗数部分Ｂ２とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の上位６４ビットのＡ２Ｂ２Ｕをベクトルレジスタｖｒ05に格納する。

【００４８】

続いて、(3) のベクトル乗算命令ＶＭＬに従って、ベクトルレジスタｖｒ01の被乗数部分Ａ１と、ベクトルレジスタｖｒ04の乗数部分Ｂ２とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の下位６４ビットのＡ１Ｂ２Ｌをベクトルレジスタｖｒ06に格納する。

【００４９】

続いて、(4) のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ05に格納されるＡ２Ｂ２Ｕと、ベクトルレジスタｖｒ06に格納されるＡ１Ｂ２Ｌと、初期値としてゼロ値を格納するマスクレジスタｍｒ00の格納データとを加算してベクトルレジスタｖｒ07に格

納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ01に格納する。

【００５０】

続いて、(5) のベクトル乗算命令ＶＭＬに従って、ベクトルレジスタｖｒ02の被乗数部分Ａ２と、ベクトルレジスタｖｒ03の乗数部分Ｂ１とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の下位６４ビットのＡ２Ｂ１Ｌをベクトルレジスタｖｒ08に格納する。

【００５１】

続いて、(6) のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ07の格納データと、ベクトルレジスタｖｒ08に格納されるＡ２Ｂ１Ｌと、初期値としてゼロ値を格納するマスクレジスタｍｒ00の格納データとを加算してベクトルレジスタｖｒ22に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ02に格納する。

【００５２】

続いて、(7) のベクトル乗算命令ＶＭＵに従って、ベクトルレジスタｖｒ01の被乗数部分Ａ１と、ベクトルレジスタｖｒ04の乗数部分Ｂ２とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の上位６４ビットのＡ１Ｂ２Ｕをベクトルレジスタｖｒ10に格納する。

【００５３】

続いて、(8) のベクトル乗算命令ＶＭＵに従って、ベクトルレジスタｖｒ02の被乗数部分Ａ２と、ベクトルレジスタｖｒ03の乗数部分Ｂ１とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の上位６４ビットのＡ２Ｂ１Ｕをベクトルレジスタｖｒ11に格納する。

【００５４】

続いて、(9) のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ10に格納されるＡ１Ｂ２Ｕと、ベクトルレジスタｖｒ11に格納されるＡ２Ｂ１Ｕと、マスクレジスタｍｒ01に格納されるキャリーアウトデータとを加算してベクトルレジスタｖｒ12に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ00に格納する。

【００５５】

続いて、(10)のベクトル乗算命令ＶＭＬに従って、ベクトルレジスタｖｒ01の被乗数部分Ａ１と、ベクトルレジスタｖｒ03の乗数部分Ｂ１とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の下位６４ビットのＡ１Ｂ１Ｌをベクトルレジスタｖｒ13に格納する。

【００５６】

続いて、(11)のベクトル加算命令ＶＡＣに従って、ベクトルレジスタｖｒ12の格納データと、ベクトルレジスタｖｒ13に格納されるＡ１Ｂ１Ｌと、マスクレジスタｍｒ02に格納されるキャリーアウトデータとを加算してベクトルレジスタｖｒ21に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ03に格納する。

【００５７】

続いて、(12)のベクトル乗算命令ＶＭＵに従って、ベクトルレジスタｖｒ01の被乗数部分Ａ１と、ベクトルレジスタｖｒ03の乗数部分Ｂ１とを乗算して、乗算器１５のセレクタを制御することで出力されるその乗算結果の上位６４ビットのＡ１Ｂ１Ｕをベクトルレジスタｖｒ15に格納する。

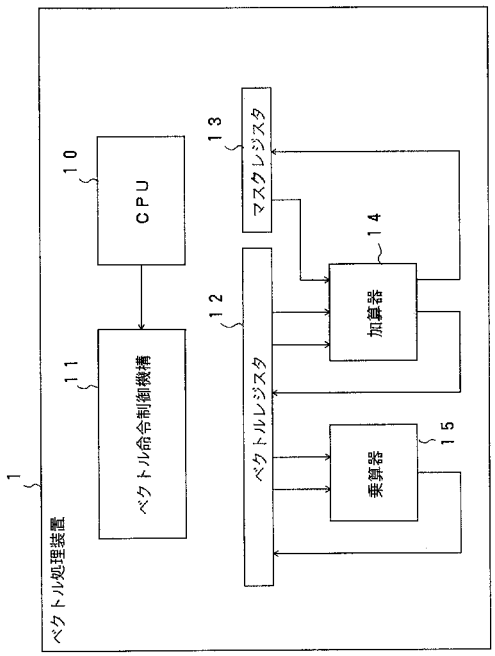
【００５８】

最後に、(13)のベクトル加算命令ＶＡＣに従って、初期値としてゼロ値を格納するマスクレジスタｖｒ00の格納データと、ベクトルレジスタｖｒ15に格納されるＡ１Ｂ１Ｕと、マスクレジスタｍｒ03に格納されるキャリーアウトデータとを加算してベクトルレジスタｖｒ20に格納するとともに、このとき発生する桁上げ値のキャリーアウトデータをマスクレジスタｍｒ00に格納する。

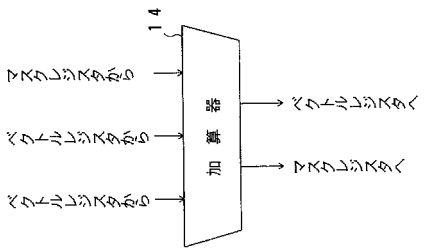
【００５９】
このように、上述の構成を採る乗算器１５を用いて実行される本発明によるベクトル乗算処理では、６４ビット同士の乗算処理により求まる１２８ビットの乗算結果の上位６４ビットか下位６４ビットのいずれかを取り出しながら、上述の構成を採る加算器１４を用いて実行される本発明によるベクトル加算処理を用いつつ、１２８ビットの被乗数と１２８ビットの乗数との乗算値を算出していくのである。
【００６０】
なお、この構成にあって、マスクレジスタｍｒ00やベクトルレジスタｖｒ00には、ゼロ値を格納しておく必要はなく、そのようなレジスタ番号が指定されるときには、ゼロ値の入力指定があったと見なしていく構成を採ってもよい。また、ｍｒ00へ書き込むキャリーアウトデータは、実際には後で使用するものではない。これから、そのようなレジスタ番号が指定されるときには、実際の書込処理を行わないことで、元のデータを壊さないようにする構成を採ってもよい。また、ベクトル加算命令ＶＡＣでは、５個のレジスタを指定しなければならないが、入力と出力とでマスクレジスタを共通にすれば、４個のレジスタの指定で済むことになる。
【００６１】
【発明の効果】
以上説明したように、本発明のベクトルデータ加算方法によれば、キャリーアウトデータをマスクレジスタに格納していく構成を採ることから、従来技術のように、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採る必要がなくなり、少ない命令数でもって高速にベクトル加算命令を実行できるようになる。
【００６２】
また、本発明のベクトルデータ乗算方法によれば、ベクトル乗算命令を実行するにあたって実行が要求されることになるベクトル加算命令の実行にあたって、キャリーアウトデータをマスクレジスタに格納していく構成を採ることから、従来技術のように、ベクトルシフト命令を実行しながらベクトル加算命令を実行していくという構成を採る必要がなくなることで、少ない命令数でもって高速にベクトル加算命令を実行できるようになり、これにより、少ない命令数でもって高速にベクトル乗算命令を実行できるようになる。
【図面の簡単な説明】
【図１】　本発明を実行するベクトル処理装置の装置構成図である。
【図２】　本発明で用いる加算器の説明図である。
【図３】　本発明で用いる乗算器の説明図である。
【図４】被加数及び加数の格納処理の説明図である。
【図５】被加数及び加数の格納処理の説明図である。
【図６】本発明で発行するベクトル加算命令の説明図である。
【図７】本発明の加算処理の説明図である。
【図８】４倍精度データのデータフォーマットの説明図である。
【図９】４倍精度乗算処理のオペランドの説明図である。
【図１０】被乗数及び乗数の格納処理の説明図である。
【図１１】本発明で発行するベクトル乗算命令の説明図である。
【図１２】本発明の乗算処理の説明図である。
【図１３】従来技術の説明図である。
【図１４】従来技術の説明図である。
【図１５】従来技術の説明図である。
【図１６】従来技術の説明図である。
【図１７】従来技術の説明図である。
【符号の説明】
１　　ベクトル処理装置
１０　ＣＰＵ
１１　ベクトル命令制御機構

１２　ベクトルレジスタ
１３　マスクレジスタ
１４　加算器
１５　乗算器

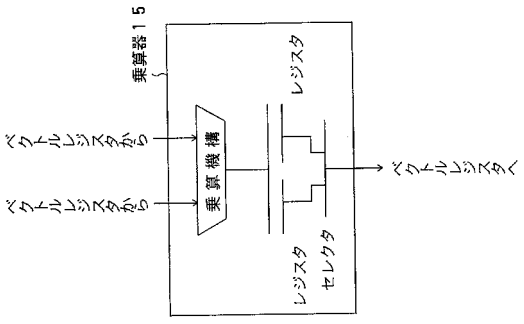
【図１】



【図２】



【図３】



【図４】

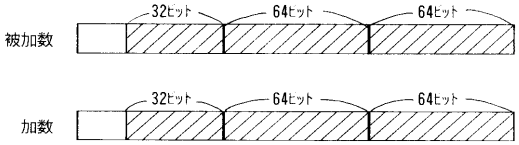
被加数及び加数の格納処理の説明図

vr00：bit00～31　“0”
vr00：bit32～63　被加数のビット00～31
vr01：bit00～63　被加数のビット32～95
vr02：bit00～63　被加数のビット96～159

vr03：bit00～31　“0”
vr03：bit32～63　加数のビット00～31
vr04：bit00～63　加数のビット32～95
vr05：bit00～63　加数のビット96～159

【図５】

被加数及び加数の格納処理の説明図



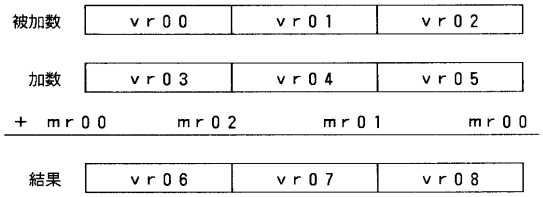
【図６】

本発明で発行するベクトル加算命令の説明図

(1) VAC　vr02, vr05, mr00, vr08, mr01
(2) VAC　vr01, vr04, mr01, vr07, mr02
(3) VAC　vr00, vr03, mr02, vr06, mr00

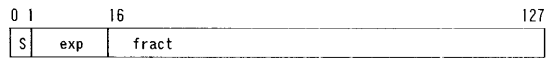
【図７】

本発明の加算処理の説明図



【図８】

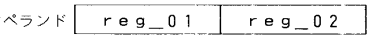
4倍精度データのデータフォーマットの説明図



bit00　：符号s
bit01～bit15　：指数e
bit16～bit127：仮数f

【図９】

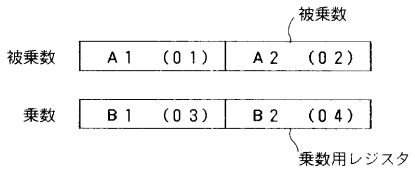
4倍精度乗算処理のオペランドの説明図



reg_01　bit 0～bit14　全て零
　　bit15　隠れビット
　　bit16～bit63　仮数の上位48ビット
reg_02　bit00～bit63　仮数の下位64ビット

【図１０】

被乗数及び乗数の格納処理の説明図



【図１１】

本発明で発行するベクトル乗算命令の説明図

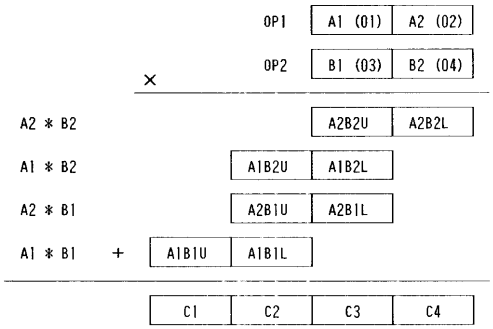
```
                                                    (C4)
!A2B2L
!A2B2U
!A1B2L
                                    (C3)
!A2B1L
!
!A1B2U
!A2B1U
                                                    (C2)
!A1B1L
!
                                                    (C1)
!A1B1U
!
```

```
vr02(A2), vr04(B2), vr23
vr02(A2), vr04(B2), vr05
vr01(A1), vr04(B2), vr06
vr05, vr06, mr00, vr07, mr01
vr02(A2), vr03(B1), vr08
vr07, vr08, mr00, vr22, mr02
vr01(A1), vr04(B2), vr10
vr02(A2), vr03(B1), vr11
vr10, vr11, mr01, vr12, mr00
vr01(A1), vr03(B1), vr13
vr12, vr13, mr02, vr21, mr03
vr01(A1), vr03(B1), vr15
vr00, vr15, mr03, vr20, mr00
```

```
(1) VML
(2) VMU
(3) VML
(4) VAC
(5) VML
(6) VAC
(7) VMU
(8) VMU
(9) VAC
(10) VML
(11) VAC
(12) VMU
(13) VAC
```

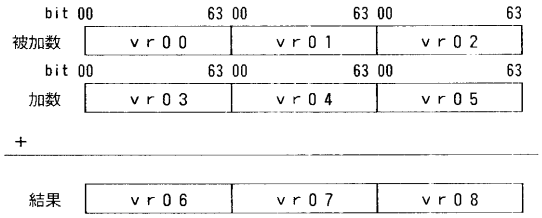
【図１２】

本発明の乗算処理の説明図



【図１３】

従来技術の説明図



【図１４】

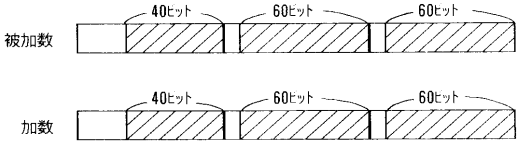
従来技術の説明図

vr00：bit00～23 “0”
vr00：bit24～63 被加数のビット00～39
vr01：bit00～03 “0”
vr01：bit04～63 被加数のビット40～99
vr02：bit00～03 “0”
vr02：bit04～63 被加数のビット100～159

vr03：bit00～23 “0”
vr03：bit24～63 加数のビット00～39
vr04：bit00～03 “0”
vr04：bit04～63 加数のビット40～99
vr05：bit00～03 “0”
vr05：bit04～63 加数のビット100～159

【図１５】

従来技術の説明図



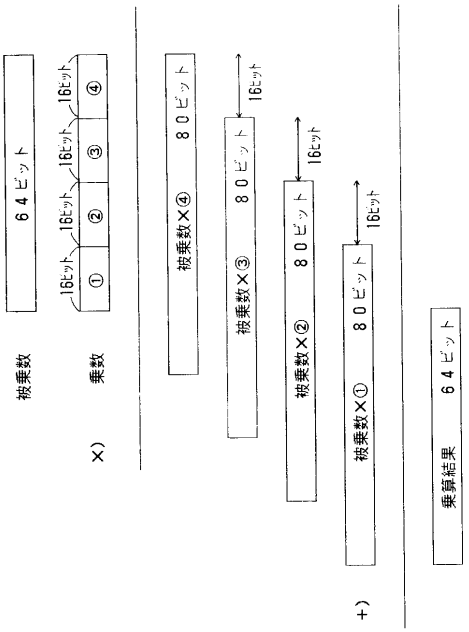
【図１６】

従来技術の説明図

(1) VA vr02, vr05, vr10
(2) VSR vr10, 60, vr15
(3) VA vr01, vr04, vr20
(4) VA vr15, vr20, vr20
(5) VSR vr20, 60, vr25
(6) VA vr00, vr03, vr30
(7) VA vr25, vr30, vr6
(8) VSL vr10, 4, vr10
(9) VSR vr10, 4, vr8
(10) VSL vr20, 4, vr20
(11) VSR vr20, 4, vr7

【図１７】

従来技術の説明図



フロントページの続き

(56)参考文献 特開平０５－０２０３５２（ＪＰ，Ａ）
特開昭６０－１３８６３７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 17/16
G06F 7/50-54