(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2012-128559
(P2012-128559A)
(43)公開日　平成24年7月5日(2012.7.5)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/16	(2006.01)	G06F	17/16	A	5B013
G06F 9/34	(2006.01)	G06F	9/34	350B	5B033
G06F 9/38	(2006.01)	G06F	9/38	310X	5B056

審査請求 未請求　請求項の数 5　ＯＬ　（全 17 頁）

(21)出願番号　特願2010-278041 (P2010-278041)
(22)出願日　平成22年12月14日 (2010.12.14)

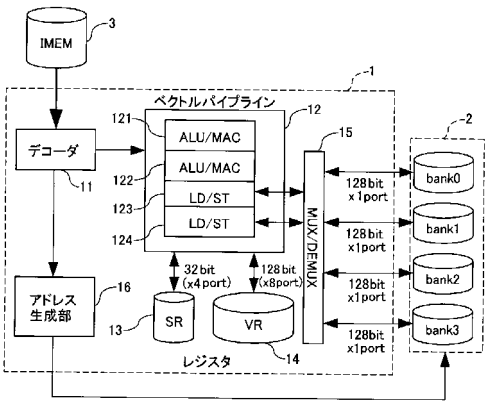
(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100099759
弁理士　青木　篤
(74)代理人　100119987
弁理士　伊坪　公一
(74)代理人　100081330
弁理士　樋口　外治
(74)代理人　100114177
弁理士　小林　龍
(72)発明者　都市　雅彦
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

最終頁に続く

(54)【発明の名称】演算処理装置

(57)【要約】
【課題】ストライド命令の後続命令に対してもチェイニングを行って性能を向上することができる演算処理装置の提供を図る。
【解決手段】同時アクセス可能な複数のメモリブロックｂａｎｋ０～ｂａｎｋ３を有するデータメモリ２との間でデータを遣り取りする複数のベクトルパイプライン１２１～１２４を有する演算処理装置であって、前記データメモリに対するストライドアクセスを、基本パターンのデータサイズを決める第１パラメータと、該基本パターンにおける有効なデータ数を決める第２パラメータで規定する。
【選択図】図３

図3



【特許請求の範囲】
【請求項１】
同時アクセス可能な複数のメモリブロックを有するデータメモリとの間でデータを遣り取りする複数のベクトルパイプラインを有する演算処理装置であって、
前記データメモリに対するストライドアクセスを、基本パターンのデータサイズを決める第１パラメータと、該基本パターンにおける有効なデータ数を決める第２パラメータで規定することを特徴とする演算処理装置。
【請求項２】
前記第１パラメータをＤＳＴとし、前記第２パラメータをＣＮＴとし、前記同時アクセス可能なメモリブロックの数をＮとするとき、ＤＳＴおよびＣＮＴは、ＣＮＴ×Ｎ≧ＤＳＴを満たす整数として規定されることを特徴とする請求項１に記載の演算処理装置。
【請求項３】
さらに、第１ビット幅を有する第１レジスタを有し、
前記ストライドアクセスにより同時にアクセスされるメモリブロックの数は、前記第１ビット幅に従って規定されることを特徴とする請求項１または２に記載の演算処理装置。
【請求項４】
前記ストライドアクセスは、ストライドロード／ストア命令によるアクセスであり、
デコードした命令が前記ストライドロード／ストア命令のとき、該ストライドロード／ストア命令の先行命令がロード／ストア命令の場合には、当該先行命令が完了した時点で、前記ストライドロード／ストア命令を前記ベクトルパイプラインへ発行するようになっていることを特徴とする請求項１～３のいずれか１項に記載の演算処理装置。
【請求項５】
さらに、アドレス生成部を有し、
該アドレス生成部は、前記第１および第２パラメータにより規定される前記基本パターンおよびベースアドレスを受け取って、前記複数のメモリブロックに対するアドレス信号をそれぞれ生成することを特徴とする請求項１～４のいずれか１項に記載の演算処理装置。
【発明の詳細な説明】
【技術分野】
【０００１】
この出願で言及する実施例は、演算処理装置に関する。
【背景技術】
【０００２】
従来、配列データに対する計算（ベクトル演算）を１命令で処理可能な演算処理装置（プロセッサ）として、ベクトルプロセッサが利用されている。このようなベクトルプロセッサは、気象予測や流体解析といった科学技術計算に適用されているが、近年、携帯端末のソフトウェア無線（ＳＤＲ：Software Defined Radio）への適用も考えられている。
【０００３】
ベクトルプロセッサは、複数の演算器に対して連続的にデータを投入することで、高い演算スループットを得ることでき、１サイクルで処理可能なデータ数を増やす様々な工夫も行われている。
【０００４】
ところで、従来、ベクトルプロセッサ（演算処理装置）としては、様々なものが提案されている。
【先行技術文献】
【特許文献】
【０００５】
【特許文献１】特開２０００－２５９６０９号公報
【特許文献２】米国特許第６５９１３４５号明細書
【発明の概要】

【発明が解決しようとする課題】
【０００６】
前述したように、ベクトルプロセッサは、一般に、［レイテンシ＋データ数／演算器数］サイクルで処理を終えることができ、特に、メモリレイテンシによる性能低下を緩和できる特徴に注目して研究が行われてきた。
【０００７】
ところで、組み込み用途では、消費電力が大きくて性能予測が難しいキャッシュメモリを用いることなく、高速なローカルメモリを採用することがある。この場合、大容量の外部メモリとローカルメモリ間の通信はプログラマが責任を持つことになるが、データメモリを固定レイテンシでアクセスできるため、パイプラインストールなどの制御を少なくすることができ、回路の単純化や占有面積の低減を図ることができる。
【０００８】
しかしながら、ストライドアクセスの飛び飛びのアドレスでは、実際に使用するデータサイズよりも広い範囲をアクセスすることになり、連続アクセスのように１サイクルでメモリアクセスが完了しない。そのため、後続命令とのチェイニングができず、性能劣化の原因になることがある。
【０００９】
また、データをどこまで転送したかを管理して、処理も遅らせることも考えられるが、制御が複雑になるといった問題がある。
【課題を解決するための手段】
【００１０】
一実施形態によれば、同時アクセス可能な複数のメモリブロックを有するデータメモリとの間でデータを遣り取りする複数のベクトルパイプラインを有する演算処理装置が提供される。
【００１１】
前記データメモリに対するストライドアクセスを、基本パターンのデータサイズを決める第１パラメータと、該基本パターンにおける有効なデータ数を決める第２パラメータで規定する。
【発明の効果】
【００１２】
開示の演算処理装置は、ストライド命令の後続命令に対してもチェイニングを行って性能向上を図ることができるという効果を奏する。
【図面の簡単な説明】
【００１３】
【図１】演算処理装置の一例における命令実行を説明するためのタイムチャートである。
【図２】ベクトル命令のチェイニングを説明するためのタイムチャートである。
【図３】本実施例の演算処理装置の構成例を示すブロック図である。
【図４】図３の演算処理装置におけるデータメモリの読み出しシーケンスの一例を説明するための図である。
【図５】図３の演算処理装置におけるデータメモリのアドレス割り当てを説明するための図である。
【図６】図３の演算処理装置におけるスカラレジスタを説明するための図である。
【図７】図３の演算処理装置におけるベクトルレジスタを説明するための図である。
【図８】本実施例の演算処理装置におけるシーケンシャルアクセスの一例を説明するためのタイムチャートである。
【図９】本実施例の演算処理装置によるストライドアクセスの一例を説明するための図である。
【図１０】図９のストライドアクセスを説明するためのタイムチャートである。
【図１１】本実施例の演算処理装置によるストライドアクセスの他の例を説明するための図である。

【図１２】図１１のストライドアクセスを説明するためのタイムチャートである。
【図１３】本実施例の演算処理装置によるストライドアクセスのさらに他の例を説明するための図である。
【図１４】図１３のストライドアクセスを説明するためのタイムチャートである。
【図１５】本実施例の演算処理装置におけるストライドアクセスの動作を規定するパラメータを説明するための図である。
【図１６】本実施例の演算処理装置におけるアドレス生成部の一例を示すブロック図である。
【発明を実施するための形態】
【００１４】
まず、本実施例の演算処理装置を詳述する前に、演算処理装置の一例における命令の実行、並びに、ベクトル命令のチェイニングを、図１および図２を参照して説明する。
【００１５】
図１は、演算処理装置の一例における命令実行を説明するためのタイムチャートである。ここで、演算処理装置（ベクトルプロセッサ）は、配列データに対するベクトル演算を１命令で処理可能なプロセッサであり、演算器に対して連続的にデータを投入することで高い演算スループットを得るようになっている。
【００１６】
また、ベクトルプロセッサは、並列に動作可能な複数の演算器を有し、連続した配列データに対しては、［スタートアップ（レイテンシ）＋データ数／演算器数］サイクルで処理するようになっている。
【００１７】
また、同時動作可能な複数のベクトルパイプラインを設け、命令を並列に実行することで、さらなる性能向上を図ることも行われている。
【００１８】
具体的に、図１に示されるように、例えば、８個の１６ビット演算器を有するベクトルプロセッサは、６４要素の配列データに対して演算を行う場合、スタートアップを４にすると、４＋６４／８＝１２サイクルで演算を終えることができる。なお、スタートアップは、全てのパイプラインにデータが流れるまでの時間（サイクル）に対応する。
【００１９】
ここで、各演算器では、命令のフェッチ（fetch）、デコード（decode）、レジスタからの読み出し（reg. read）、実行（execute）およびライトバック（writeback）の５つの処理が行われる。
【００２０】
なお、図１の各ブロック中の『０..７』，『８..１５』，…，『５６..６３』は、６４要素の配列データにおいて、各演算器で１サイクルごとに処理される８要素のデータを示している。
【００２１】
図２は、ベクトル命令のチェイニングを説明するためのタイムチャートである。図２において、参照符号ｖｌｄｈは、ハーフワード（halfword）データをデータメモリから読み出してレジスタに格納させる命令であり、また、ｖａｄｄｈは、ベクトルレジスタ同士の加算を行わせる命令である。
【００２２】
具体的に、図２の命令『vldh sr1 vr0』は、ｓｒ１が示すアドレスから連続した領域にメモリアクセスを行わせ、ｖｒ０～ｖｒ６３へ格納させる命令である。なお、『vaddh vr0 vr64 vr128』は、先行するｖｌｄｈとデータ依存があるが、５サイクル目にはｖａｄｄｈ命令で一番先に行う演算に必要なデータが参照可能になるため、このタイミングからｖａｄｄｈの演算を開始することができる。
【００２３】
それ以降、ｖａｄｄｈは、先行のｖｌｄｈ命令を追いかけるかのように、メモリから読

み出されるデータを次々と処理し、５サイクル目以降、それぞれのベクトルパイプラインでｖｌｄｈとｖａｄｄｈは同時に動作する。

【００２４】

このようなベクトル命令間の並列動作をチェイニングというが、ｖｌｄｈ命令により読み出されるデータサイズと、ｖａｄｄｈ命令により必要とされるデータサイズが同じであれば、データ依存関係を満たして正しく演算することが可能になる。

【００２５】

ところで、ベクトルプロセッサは、配列データの演算といった連続データを扱うことに向いており、連続したメモリアドレスに対するアクセスをシーケンシャルアクセスという。

【００２６】

しかしながら、連続したデータだけではなく、より複雑なデータパターンに対する演算を行いたいという要求がある。これは、例えば、ソフトウェア無線（ＳＤＲ：Software Defined Radio）への適用といったものが考えられ、一定の間隔で並んだデータを抽出して演算を行うといった処理である。

【００２７】

その場合、メモリ上に並んだデータから必要なデータだけを取り出して、ロード／ストアする命令が必要になる。一般に、このようなアクセスパターンは、ストライドアクセスと呼ばれている。

【００２８】

しかしながら、ストライドアクセス（ストライド命令）の飛び飛びのアドレスでは、実際に使用するデータサイズよりも広い範囲をアクセスすることになり、連続アクセスのように１サイクルでメモリアクセスが完了しない。

【００２９】

そのため、後続命令とのチェイニングができず、性能劣化の原因になることがある。また、データをどこまで転送したかを管理して、処理も遅らせることも考えられるが、制御が複雑になる。

【００３０】

以下、演算処理装置の実施例を、添付図面を参照して詳述する。図３は、本実施例の演算処理装置の構成例を示すブロック図である。図３において、参照符号１は演算処理装置、２はデータメモリ、そして、３は命令メモリ（ＩＭＥＭ）を示す。

【００３１】

演算処理装置１は、デコーダ（デコードロジック）１１、ベクトルパイプライン部１２、スカラーレジスタ（ＳＲ）１３、ベクトルレジスタ（ＶＲ）１４、マルチプレクサ・デマルチプレクサ（MUX/DEMUX）１５、および、アドレス生成部１６を有する。

【００３２】

ベクトルパイプライン部１２は、４本のパイプラインを有する。その内、２本がＡＬＵ，乗算，論理演算などの演算命令を実行するベクトルパイプライン１２１，１２２である。また、残りの２本がロード・ストア（Load/Store）などの転送命令を実行するベクトルパイプライン１２３，１２４である。

【００３３】

各ベクトルパイプライン１２１～１２４は、例えば、１６ビットの演算器を８個ずつ有し、それぞれ毎サイクル、１６ビット演算を８並列で演算することができる。データメモリ２は、４つのバンク（メモリブロック）ｂａｎｋ０～ｂａｎｋ３を有し、マルチプレクサ・デマルチプレクサ１５を介してベクトルパイプライン１２３，１２４に接続されている。

【００３４】

アドレス生成部１６は、デコーダ１１の出力に応じて、例えば、ストライドロードストア命令（ストライドLoad/Store命令）のときに、図８～図１４を参照して説明する処理を行うために、データメモリ２に対するアドレス信号等を生成する。

【００３５】
なお、スカラーレジスタ１３は、例えば、３２ビット幅のレジスタで４つのエントリを有し、また、ベクトルレジスタ１４、例えば、１２８ビット幅のレジスタで８つのエントリを有している。
【００３６】
図４は、図３の演算処理装置におけるデータメモリの読み出しシーケンスの一例を説明するための図である。
【００３７】
図４に示されるように、データメモリ２の読み出しシーケンスは、アドレス（address）が入力されてから、次のクロック（clock）でデータ（read data）が読み出される。すなわち、データメモリは、レイテンシが１のローカルメモリとなっている。なお、命令メモリもレイテンシが１のローカルメモリとなっている。
【００３８】
図５は、図３の演算処理装置におけるデータメモリのアドレス割り当てを説明するための図である。データメモリ２は、４つのバンクｂａｎｋ０～ｂａｎｋ３を有し、各バンクｂａｎｋ０～ｂａｎｋ３は、例えば、それぞれ１２８ビット幅の読み出し／書き込み共用のアクセスポートを１つ有する。なお、アドレスは、例えば、バンクインタリーブ方式で割り振られている。
【００３９】
具体的に、バイトアドレス０～１５の１６バイト（１２８ビット）のデータは、データメモリ２のバンクｂａｎｋ０（dmem-bank0）に格納（転送）され、また、バイトアドレス１６～３１の１６バイトのデータは、バンクｂａｎｋ１（dmem-bank1）に格納される。
【００４０】
さらに、バイトアドレス３２～４７の１６バイトのデータは、データメモリ２のバンクｂａｎｋ２（dmem-bank2）に格納され、また、バイトアドレス４８～６３の１６バイトのデータは、バンクｂａｎｋ３（dmem-bank3）に格納される。そして、バイトアドレス６４～７９の１６バイトのデータは、再びバンクｂａｎｋ０（dmem-bank0）に格納され、同様の処理が繰り返される。
【００４１】
従って、或るデータにアクセスしたい場合、各バンクメモリの物理アドレスは、次のように求めることができる。

バンクメモリの物理アドレス
＝（データのバイトアドレス）÷（各バンクのラインサイズ　×　バンク数）
＝（データのバイトアドレス）÷（１６　×　４）

【００４２】
図６は、図３の演算処理装置におけるスカラーレジスタを説明するための図であり、また、図７は、図３の演算処理装置におけるベクトルレジスタを説明するための図である。
【００４３】
図６に示されるように、スカラーレジスタ（ＳＲ）１３は、例えば、３２ビット幅のレジスタであり、例えば、アドレス（address）等のデータが格納される。
【００４４】
図７に示されるように、ベクトルレジスタ（ＶＲ）１４は、例えば、１２８ビット幅のレジスタであり、例えば、１６ビットデータの各要素が８個ずつ格納される。すなわち、ベクトルレジスタ１４の各エントリには、それぞれ要素０～７，要素８～１５，要素１６～２３，…が格納される。
【００４５】
ここで、前に、図１を参照して説明したように、ベクトルプロセッサ（各演算器）では、命令のフェッチ（fetch）、デコード（decode）、レジスタからの読み出し（reg. read）、実行（execute）およびライトバック（writeback）の各ステージの処理が行われる。
【００４６】

なお、図３に示すデコーダ１１は、例えば、フェッチした命令をデコードし、１サイクルに１命令ずつベクトルパイプラインに投入する。なお、各命令で演算するデータ数は、例えば、ベクトルレングス（Vector Length：ＶＬ）という制御レジスタで管理される。

【００４７】

図８は、本実施例の演算処理装置におけるシーケンシャルアクセスの一例を説明するためのタイムチャートである。ここで、データメモリ２の各バンクｂａｎｋ０～ｂａｎｋ３には、図５を参照して説明した１６バイトのデータがそれぞれ格納されている。

【００４８】

すなわち、図８に示されるように、実行（演算）ステージでは、アドレスＡ，Ａ＋１，…に従ってデータメモリ２をアクセスしたデータを演算し、そして、ライトバックステージでその演算結果をベクトルレジスタ１４の各エントリにライトバックされる。

【００４９】

ここで、アドレスＡによりアクセスされる各バンクのバイトアドレスは、例えば、サイクル１でｂａｎｋ０のバイトアドレス０～１５、また、サイクル２でｂａｎｋ１のバイトアドレス１６～３１となる。そして、例えば、サイクル３でｂａｎｋ２のバイトアドレス３２～４７、また、サイクル４でｂａｎｋ３のバイトアドレス３２～６３となる。

【００５０】

さらに、次のアドレスＡ＋１によりアクセスされる各バンクのバイトアドレスは、例えば、サイクル５でｂａｎｋ０のバイトアドレス６４～７９、また、サイクル６でｂａｎｋ１のバイトアドレス８０～９５となる。そして、例えば、サイクル７でｂａｎｋ２のバイトアドレス９６～１１１、また、サイクル８でｂａｎｋ３のバイトアドレス１１２～１２７となる。

【００５１】

なお、ライトバックステージでは、例えば、サイクル２からベクトルレジスタ１４のエントリ０，１，２…に対して、データメモリ２において前のサイクルでアクセスされた１２８ビット（１６バイト）のデータの格納（ライトバック）が行われる。

【００５２】

このように、シーケンシャルアクセスを行うベクトルロード命令では、各バンクｂａｎｋ０～ｂａｎｋ３のデータ幅（１２８ビット）と転送先のベクトルレジスタ１４のデータ幅が同じであるため、毎サイクル１バンクずつアクセスすればよいことになる。

【００５３】

次に、命令をデコードしたときに、ストライドロードストア命令（ストライドLoad/Store命令）であった場合の動作を説明する。命令をデコードしたときに、ストライドLoad/Store命令であった場合、命令発行制御部は、先行命令でLoad/Store命令が実行されているか否かを確認する。

【００５４】

Load/Store命令が実行中であれば、そのLoad/Store命令が完了するまでベクトルパイプラインへの発行を待つ。そして、先行のLoad/Store命令が完了した時点で、ストライドLoad/Store命令がベクトルパイプラインへ発行される。

【００５５】

ストライドLoad/Store命令を実行中は、後続の命令がLoad/Store命令であれば、発行部はベクトルパイプラインへの命令発行を、そのストライドLoad/Storeが完了するまで発行を待ち合わせる。

【００５６】

ベクトルパイプラインにストライドLoad/Store命令が投入されると、ベクトル命令は、引数であるスカラーレジスタ（ＳＲ）１３を読み出し、アクセスデータパターン（ｓｒ０～ｓｒ３１）を読み出す。

【００５７】

アクセスデータパターンは、ディスタンス（distance）およびカウント（count）という２つのパラメータで決まる基本パターン（ストライドパターン）の繰り返しとして規定

される。ここで、ディスタンスは、基本パターンのデータサイズを決めるパラメータであり、また、カウントは、有効なデータ数を決めるパラメータである。

【００５８】

図９は、本実施例の演算処理装置によるストライドアクセスの一例を説明するための図であり、また、図１０は、図９のストライドアクセスを説明するためのタイムチャートである。なお、図９および図１０は、distance＝４，count＝２のバイトデータ（distance＝２，count＝１のハーフデータ）のストライドLoad/Store命令を説明するためのものである。

【００５９】

distance＝４でcount＝２の場合、すなわち、基本パターンのデータサイズが４バイトで、有効なデータ数が２バイトの場合、図９に示されるように、データメモリ２のバンクｂａｎｋ０～ｂａｎｋ３からベクトルレジスタ（ＶＲ）１４へデータが転送される。

【００６０】

すなわち、ｂａｎｋ０のバイトアドレス０～３中のアドレス０，１のデータ、バイトアドレス４～７中のアドレス４，５のデータ、…、ｂａｎｋ１のバイトアドレス２８～３１中のアドレス２８，２９のデータが、ＶＲのエントリに格納される。

【００６１】

ここで、バイトアドレス０～３において、最初のアドレス０がベースアドレス（基底アドレス）になり、また、バイトアドレス４～７において、最初のアドレス４がベースアドレスになる。

【００６２】

次に、ｂａｎｋ２のバイトアドレス３２～３５中のアドレス３２，３３のデータ、バイトアドレス３６～３９中のアドレス３６，３７のデータ、…、ｂａｎｋ３のバイトアドレス６０～６３中のアドレス６０，６１のデータが、ＶＲのエントリに格納される。

【００６３】

さらに、ｂａｎｋ０のバイトアドレス６４～６７中のアドレス６４，６５のデータ、バイトアドレス６８～７１中のアドレス６８，６９のデータ、…、ｂａｎｋ１のバイトアドレス９２～９５中のアドレス９２，９３のデータが、ＶＲのエントリに格納される。

【００６４】

このように、ロード命令（Load）では、データメモリ２上のデータから２バイトおきのデータを抽出し、それらのデータを整列させてベクトルレジスタ１４へ転送（格納）する。

【００６５】

ここで、ベクトルレジスタ１４のデータ幅は１２８ビットなので、図１０に示されるように、２つのバンクを同時にアクセスすることで、データメモリ２における１２８ビットのデータを扱うようになっている。

【００６６】

すなわち、サイクル１では、アドレスＡによるｂａｎｋ０のバイトアドレス０～１５、および、ｂａｎｋ１のバイトアドレス１６～３１を同時にアクセスする。また、サイクル２では、アドレスＡによるｂａｎｋ２のバイトアドレス３２～４７、および、ｂａｎｋ３のバイトアドレス３２～６３を同時にアクセスする。

【００６７】

さらに、サイクル３では、アドレスＡ＋１によるｂａｎｋ０のバイトアドレス６４～７９、および、ｂａｎｋ１のバイトアドレス８０～９５を同時にアクセスする。そして、サイクル４では、アドレスＡ＋１によるｂａｎｋ２のバイトアドレス９６～１１１、および、ｂａｎｋ３のバイトアドレス１１２～１２７を同時にアクセスする。

【００６８】

なお、ライトバックステージでは、例えば、サイクル２からベクトルレジスタ１４のエントリ０，１，２…に対して、データメモリ２において前のサイクルでアクセスされた２つのバンクからの１２８ビットのデータ転送が行われる。

【００６９】
従って、distance＝４でcount＝２のバイトデータのストライドLoad/Store命令では、２つバンクを同時アクセスすることにより、転送先ベクトルレジスタ１４のデータ幅と同じ１２８ビットにすることができる。
【００７０】
なお、有効なデータのバイトアドレスは、必ず連番になるので、ベースアドレスまたはベースアドレス＋１のどちらを演算子として使用するかを決めればよいことになる。
【００７１】
このように、本実施例によれば、毎サイクルで、ベクトルパイプラインのデータ幅と同じデータ転送を可能にすることができる。これにより、ストライドロード／ストア命令（ストライドLoad/Store命令）の後続命令に対しても、チェイニングを行うことが可能になり、演算処理装置の性能向上を図ることができる。
【００７２】
なお、命令をデコードしたとき、ストライドLoad/Store命令であった場合、このストライドLoad/Store命令の先行命令がLoad/Store命令の場合には、その先行命令が完了した時点で、ストライドLoad/Store命令をベクトルパイプラインへ発行するようになっている。これは、後述する他のストライドアクセスの例でも同様である。
【００７３】
図１１は、本実施例の演算処理装置によるストライドアクセスの他の例を説明するための図であり、また、図１２は、図１１のストライドアクセスを説明するためのタイムチャートである。なお、図１１および図１２は、distance＝６，count＝２のバイトデータ（distance＝３，count＝１のハーフデータ）のストライドLoad/Store命令を説明するためのものである。
【００７４】
distance＝６でcount＝２の場合、すなわち、基本パターンのデータサイズが６バイトで、有効なデータ数が２バイトの場合、図１１に示されるように、データメモリ２のバンクｂａｎｋ０～ｂａｎｋ３からベクトルレジスタ（ＶＲ）１４へデータが転送される。
【００７５】
すなわち、ｂａｎｋ０のバイトアドレス０～５中のアドレス０，１のデータ、バイトアドレス６～１１中のアドレス６，７のデータ、…、ｂａｎｋ２のバイトアドレス４２～４７中のアドレス４２，４３のデータが、ＶＲのエントリに格納される。
【００７６】
次に、ｂａｎｋ３のバイトアドレス４８～５３中のアドレス４８，４９のデータ、バイトアドレス５４～５９中のアドレス５４，５５のデータ、…、ｂａｎｋ１のバイトアドレス９０～９５中のアドレス９０，９１のデータが、ＶＲのエントリに格納される。
【００７７】
さらに、ｂａｎｋ２のバイトアドレス９６～１０１中のアドレス９６，９７のデータ、バイトアドレス１０２～１７中のアドレス１０２，１０３のデータ、…、ｂａｎｋ０のバイトアドレス１３８～１４３中のアドレス１３８，１３９のデータが、ＶＲのエントリに格納される。
【００７８】
ここで、ベクトルレジスタ１４のデータ幅は１２８ビットなので、図１２に示されるように、３つのバンクを同時にアクセスすることで、データメモリ２における１２８ビットのデータを扱うようになっている。
【００７９】
すなわち、サイクル１では、アドレスＡによるｂａｎｋ０のバイトアドレス０～１５、ｂａｎｋ１のバイトアドレス１６～３１、および、アドレスＡによるｂａｎｋ２のバイトアドレス３２～４７を同時にアクセスする。
【００８０】
また、サイクル２では、アドレスＡによるｂａｎｋ３のバイトアドレス３２～６３、ア

ドレスＡ＋１によるｂａｎｋ０のバイトアドレス６４～７９、および、ｂａｎｋ１のバイトアドレス８０～９５を同時にアクセスする。

【００８１】

さらに、サイクル３では、アドレスＡ＋１によるｂａｎｋ２のバイトアドレス９６～１１１、ｂａｎｋ３のバイトアドレス１１２～１２７、および、アドレスＡ＋２によるｂａｎｋ０のバイトアドレス１２８～１４２を同時にアクセスする。

【００８２】

なお、ライトバックステージでは、例えば、サイクル２からベクトルレジスタ１４のエントリ０，１，２…に対して、データメモリ２において前のサイクルでアクセスされた３つのバンクからの１２８ビットのデータ転送が行われる。

【００８３】

従って、distance＝６でcount＝２のバイトデータのストライドLoad/Store命令では、３つバンクを同時アクセスすることにより、転送先のベクトルレジスタ１４のデータ幅と同じ１２８ビットにすることができる。

【００８４】

このように、本実施例によれば、毎サイクルで、ベクトルパイプラインのデータ幅と同じデータ転送が可能になり、ストライドLoad/Store命令の後続命令に対しても、チェイニングを行うことが可能になり、演算処理装置の性能向上を図ることができる。

【００８５】

図１３は、本実施例の演算処理装置によるストライドアクセスのさらに他の例を説明するための図であり、また、図１４は、図１３のストライドアクセスを説明するためのタイムチャートである。なお、図１３および図１４は、distance＝５，count＝２のバイトデータのストライドLoad/Store命令を説明するためのものである。

【００８６】

distance＝５でcount＝２の場合、すなわち、基本パターンのデータサイズが５バイトで、有効なデータ数が２バイトの場合、図１３に示されるように、データメモリ２のバンクｂａｎｋ０～ｂａｎｋ３からベクトルレジスタ（ＶＲ）１４へデータが転送される。

【００８７】

すなわち、ｂａｎｋ０のバイトアドレス０～４中のアドレス０～２のデータ、バイトアドレス５～９中のアドレス５～７のデータ、…、ｂａｎｋ１のバイトアドレス２５～２９中のアドレス２５のデータが、ＶＲのエントリに格納される。

【００８８】

次に、ｂａｎｋ１のバイトアドレス２５～２９中のアドレス２６，２７のデータ、バイトアドレス３０～ｂａｎｋ２のバイトアドレス３４中のアドレス３０～３２のデータ、…、ｂａｎｋ３のバイトアドレス５０～５４中のアドレス５０，５１のデータが、ＶＲのエントリに格納される。

【００８９】

さらに、ｂａｎｋ３のバイトアドレス５０～５４中のアドレス５２のデータ、バイトアドレス５５～５９中のアドレス５５～５７のデータ、…、ｂａｎｋ０のバイトアドレス７５～７９中のアドレス７５～７７のデータが、ＶＲのエントリに格納される。

【００９０】

ここで、ベクトルレジスタ１４のデータ幅は１２８ビットなので、図１４に示されるように、２つのバンクを同時にアクセスすることで、データメモリ２における１２８ビットのデータを扱うようになっている。

【００９１】

すなわち、サイクル１では、アドレスＡによるｂａｎｋ０のバイトアドレス０～１５、および、ｂａｎｋ１のバイトアドレス１６～３１を同時にアクセスする。また、サイクル２では、アドレスＡによるｂａｎｋ２のバイトアドレス３２～４７、および、ｂａｎｋ３のバイトアドレス３２～６３を同時にアクセスする。

【００９２】

さらに、サイクル３では、アドレスＡ＋１によるｂａｎｋ０のバイトアドレス６４～７９、および、ｂａｎｋ１のバイトアドレス８０～９５を同時にアクセスする。そして、サイクル４では、アドレスＡ＋１によるｂａｎｋ２のバイトアドレス９６～１１１、および、ｂａｎｋ３のバイトアドレス１１２～１２７を同時にアクセスする。

【００９３】

なお、ライトバックステージでは、例えば、サイクル２からベクトルレジスタ１４のエントリ０，１，２…に対して、データメモリ２において前のサイクルでアクセスされた２つのバンクからの１２８ビットのデータ転送が行われる。

【００９４】

従って、distance＝５でcount＝２のバイトデータのストライドLoad/Store命令では、２つバンクを同時アクセスすることにより、転送先ベクトルレジスタ１４のデータ幅と同じ１２８ビットにすることができる。

【００９５】

このように、本実施例によれば、毎サイクルで、ベクトルパイプラインのデータ幅と同じデータ転送が可能になり、ストライドLoad/Store命令の後続命令に対しても、チェイニングを行うことが可能になり、演算処理装置の性能向上を図ることができる。

【００９６】

以上、詳述したように、ディスタンスおよびカウントの２つのパラメータで規定される基本パターン（ストライドパターン）に従って、同時に２または３つのバンクをアクセスすることにより、転送先ベクトルレジスタのデータ幅と同じにすることができる。

【００９７】

図１５は、本実施例の演算処理装置におけるストライドアクセスの動作を規定するパラメータを説明するための図である。図１５において、distance（ディスタンス）は、基本パターンのデータサイズを決めるパラメータであり、また、count（カウント）は、有効なデータ数を決めるパラメータである。

【００９８】

また、図１５において、『ｏ』は、データメモリ２から１サイクルで取り込んでベクトルレジスタ１４に格納することができる場合を示し、また、『ｘ』は、パイプラインをストールせせる場合を示している。

【００９９】

図１５に示されるように、例えば、パラメータdistance（ディスタンス）をＤＳＴとし、パラメータcount（カウント）をＣＮＴとし、同時アクセス可能なバンクの数をＮとすると、ＤＳＴおよびＣＮＴは、次の式を満たす整数として定義することができる。

$$CNT \times N \geqq DST$$

【０１００】

図１６は、本実施例の演算処理装置におけるアドレス生成部の一例を示すブロック図であり、図８～図１４を参照して説明したデータメモリ２に対するストライドアクセスのためのアドレス信号等を生成するアドレス生成部１６の一例を示すものである。ここで、アドレス生成部１６は、ベースアドレス、ストライドパターン（基本パターン）およびオペコードを受け取って、データメモリ２に対する各種の信号を生成する。

【０１０１】

図１６に示されるように、アドレス生成部１６は、データメモリ２の各バンクｂａｎｋ０～ｂａｎｋ３に対するアドレスＡｄｄ０～Ａｄｄ３、チップセレクト信号ＣＳ０～ＣＳ３、および、ライトイネーブル信号ＷＥ０～ＷＥ３を生成する。

【０１０２】

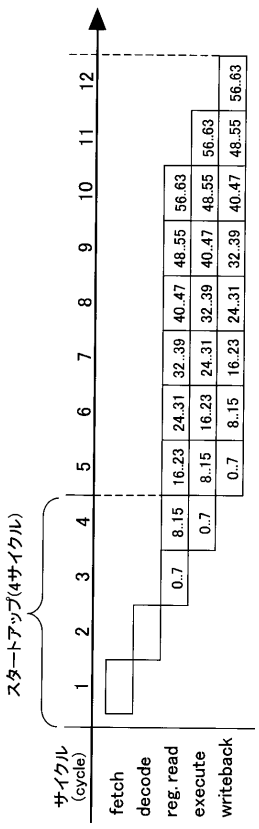
アドレス生成部１６は、セレクタ１６１～１６５、フリップフロップ１６６、インクリメント回路１６７、および、制御回路１６８を有する。セレクタ１６１～１６３は、それぞれｂａｎｋ０～ｂａｎｋ３に対するアドレスを選択するもので、ベースアドレス或いはそのベースアドレス（またはフリップフロップ１６６の出力）にインクリメント回路１６７により１を加算したアドレスを選択して出力する。

【０１０３】
制御回路１６８は、２つのパラメータ distance および count により規定されるストライドパターンおよびオペコードを受け取って、ｂａｎｋ０～ｂａｎｋ３に対するチップセレクト信号ＣＳ０～ＣＳ３およびライトイネーブル信号ＷＥ０～ＷＥ３を生成する。なお、制御回路１６８は、各セレクタ１６１～１６５に対する制御信号も生成する。
【０１０４】
以上、詳述した実施例において、データメモリ２は、４つのバンクｂａｎｋ０～ｂａｎｋ３を有するものに限定されず、また、演算処理装置１が同時アクセス可能であれば、バンク以外の複数のメモリブロックを有するものでもよい。さらに、データメモリ２に対するストライドアクセスは、ストライドLoad/Store命令に限定されるものでもない。
【０１０５】
以上の実施例を含む実施形態に関し、さらに、以下の付記を開示する。
（付記１）
同時アクセス可能な複数のメモリブロックを有するデータメモリとの間でデータを遣り取りする複数のベクトルパイプラインを有する演算処理装置であって、
前記データメモリに対するストライドアクセスを、基本パターンのデータサイズを決める第１パラメータと、該基本パターンにおける有効なデータ数を決める第２パラメータで規定することを特徴とする演算処理装置。
【０１０６】
（付記２）
前記第１パラメータをＤＳＴとし、前記第２パラメータをＣＮＴとし、前記同時アクセス可能なメモリブロックの数をＮとするとき、ＤＳＴおよびＣＮＴは、ＣＮＴ×Ｎ≧ＤＳＴを満たす整数として規定されることを特徴とする付記１に記載の演算処理装置。
【０１０７】
（付記３）
さらに、第１ビット幅を有する第１レジスタを有し、
前記ストライドアクセスにより同時にアクセスされるメモリブロックの数は、前記第１ビット幅に従って規定されることを特徴とする付記１または２に記載の演算処理装置。
【０１０８】
（付記４）
前記第１レジスタは、前記第１ビット幅の複数のエントリを有するベクトルレジスタであることを特徴とする付記２または３に記載の演算処理装置。
【０１０９】
（付記５）
前記ストライドアクセスは、ストライドロード／ストア命令によるアクセスであり、
デコードした命令が前記ストライドロード／ストア命令のとき、該ストライドロード／ストア命令の先行命令がロード／ストア命令の場合には、当該先行命令が完了した時点で、前記ストライドロード／ストア命令を前記ベクトルパイプラインへ発行するようになっていることを特徴とする付記１～４のいずれか１項に記載の演算処理装置。
【０１１０】
（付記６）
さらに、アドレス生成部を有し、
該アドレス生成部は、前記第１および第２パラメータにより規定される前記基本パターンおよびベースアドレスを受け取って、前記複数のメモリブロックに対するアドレス信号をそれぞれ生成することを特徴とする付記１～５のいずれか１項に記載の演算処理装置。
【０１１１】
（付記７）
前記アドレス生成部は、
前記基本パターンにおける有効なデータのアドレスを、前記ベースアドレスをインクリメントして生成することを特徴とする付記６に記載の演算処理装置。

【０１１２】
（付記８）
前記メモリブロックは、前記データメモリにおけるバンクであることを特徴とする付記１～７のいずれか１項に記載の演算処理装置。
【符号の説明】
【０１１３】
１　演算処理装置（ベクトルプロセッサ）
２　データメモリ
３　命令メモリ（ＩＭＥＭ）
１１　デコーダ（デコードロジック）
１２　ベクトルパイプライン部
１３　スカラーレジスタ（ＳＲ）
１４　ベクトルレジスタ（ＶＲ）
１５　マルチプレクサ・デマルチプレクサ（MUX/DEMUX）
１６　アドレス生成部
ｂａｎｋ０～ｂａｎｋ３　バンク（メモリブロック）
ＣＮＴ　カウント（count：第２パラメータ）
ＤＳＴ　ディスタンス（distance：第１パラメータ）
Ｎ　同時アクセス可能なバンク（メモリブロック）の数

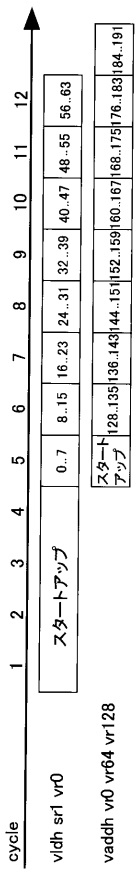
【図１】

図1



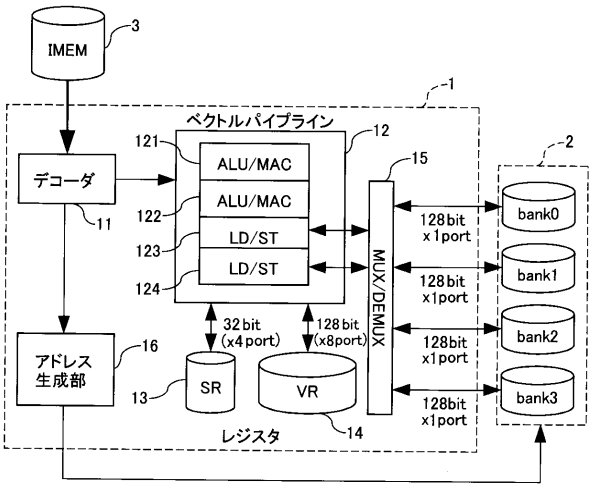
【図２】

図2



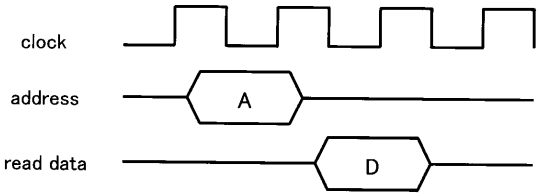
【図３】

図3



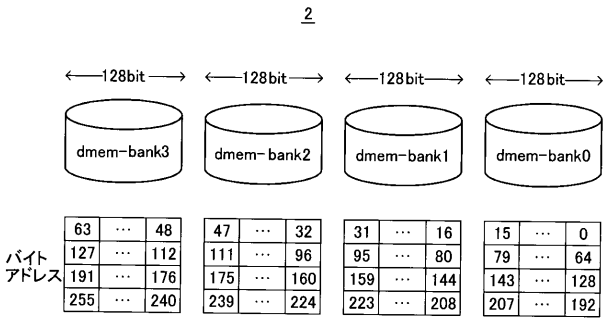
【図４】

図4



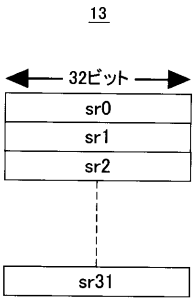
【図５】

図5



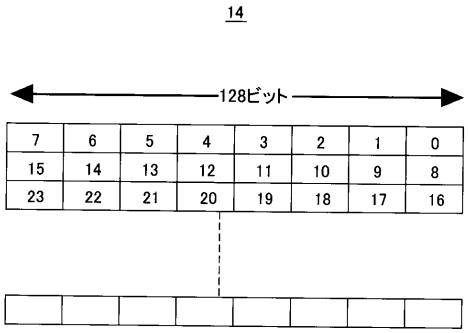
【図６】

図6



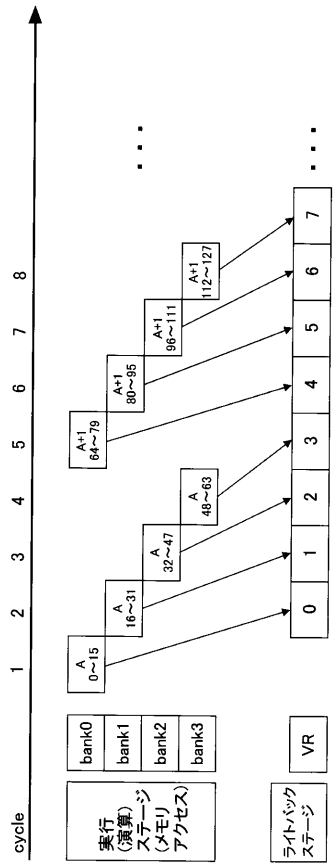
【図７】

図7



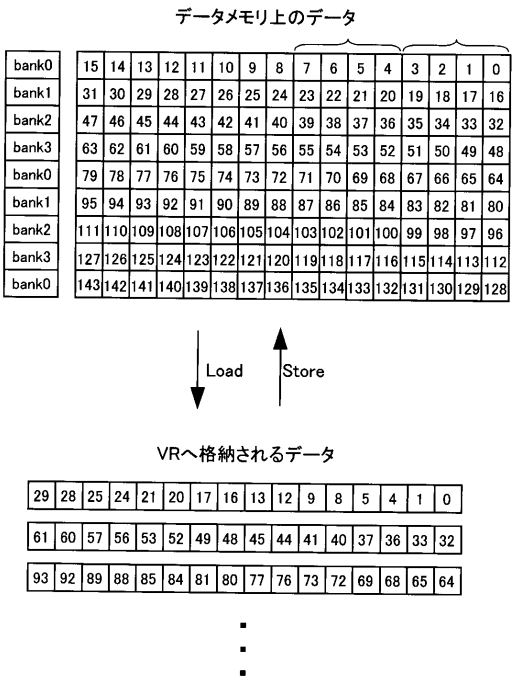
【図８】

図8



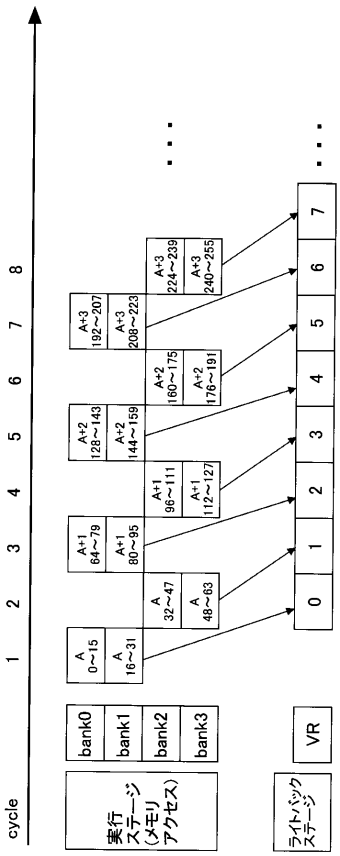
【図９】

図9



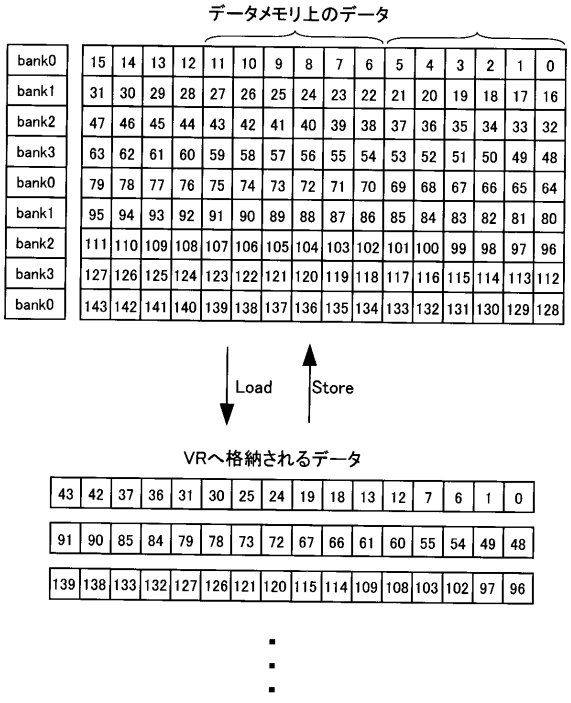
【図１０】

図10



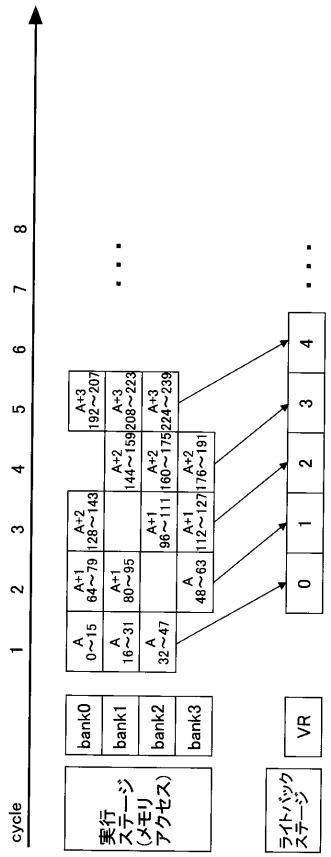
【図１１】

図11



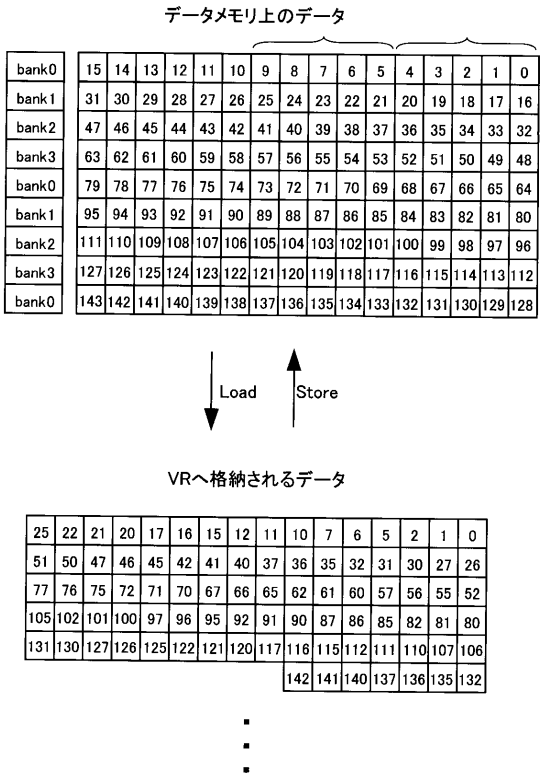
【図１２】

図12



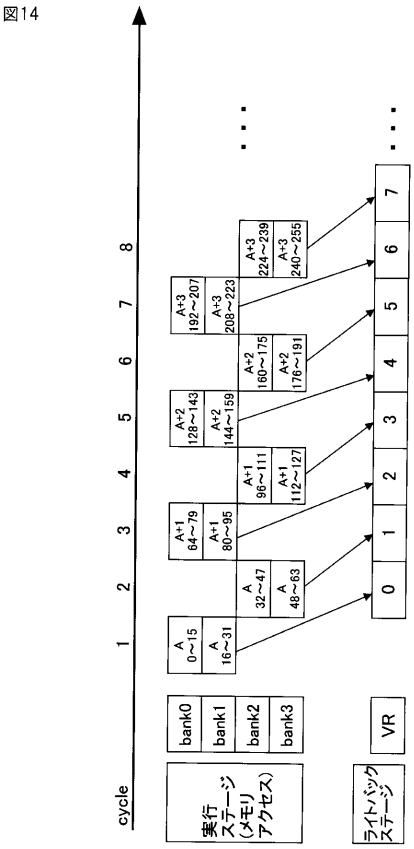
【図１３】

図13



【図１４】

図14



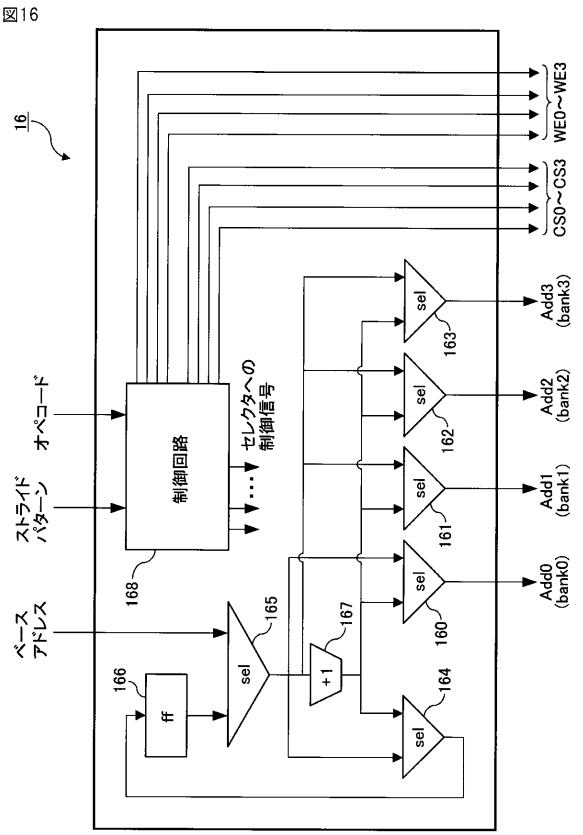
【図１５】

図15

		DST(distance：ディスタンス)							
		1	2	3	4	5	6	7	8
CNT (count) (カウント)	1	o	o	o	o	x	x	x	x
	2	x	o	o	o	o	o	o	o
	3	x	x	o	o	o	o	o	o
	4	x	x	x	o	o	o	o	o
	5	x	x	x	x	o	o	o	o
	6	x	x	x	x	x	o	o	o
	7	x	x	x	x	x	x	o	o
	8	x	x	x	x	x	x	x	o

【図１６】

図16



フロントページの続き

Ｆターム(参考) 5B013 AA20
5B033 AA04 DB12
5B056 AA05 BB32 DD12