(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5201641号
(P5201641)
(45)発行日　平成25年6月5日(2013.6.5)　(24)登録日　平成25年2月22日(2013.2.22)

(51)Int.Cl.　F I
G06F 17/16 (2006.01)　G06F 17/16 M

請求項の数 5　(全 15 頁)

(21)出願番号　特願2010-524899(P2010-524899)
(86)(22)出願日　平成20年7月28日(2008.7.28)
(65)公表番号　特表2010-539593(P2010-539593A)
(43)公表日　平成22年12月16日(2010.12.16)
(86)国際出願番号　PCT/US2008/071327
(87)国際公開番号　WO2009/035774
(87)国際公開日　平成21年3月19日(2009.3.19)
審査請求日　平成23年7月27日(2011.7.27)
(31)優先権主張番号　11/854,630
(32)優先日　平成19年9月13日(2007.9.13)
(33)優先権主張国　米国(US)

(73)特許権者　504199127
フリースケール　セミコンダクター　インコーポレイテッド
アメリカ合衆国　テキサス州　７８７３５　オースティン　ウィリアム　キャノン　ドライブ　ウエスト　６５０１
(74)代理人　100142907
弁理士　本田　淳
(72)発明者　モイヤー、ウィリアム　シー.
アメリカ合衆国　７８６２０　テキサス州　ドリッピング　スプリングス　メドウリッジ　ドライブ　１１１１

審査官　中野　裕二

最終頁に続く

(54)【発明の名称】重複オペランドを使用したＳＩＭＤの内積演算

(57)【特許請求の範囲】
【請求項１】
データ処理システムであって、
複数の汎用レジスタと、
１つ以上の命令を実行するためのプロセッサ回路と
を備え、前記１つ以上の命令は、少なくとも２つの内積を同時に実行するためのベクトル内積命令を含み、前記ベクトル内積命令は、前記複数の汎用レジスタから第１のソースレジスタと第２のソースレジスタとを特定し、前記第１のソースレジスタと前記第２のソースレジスタとの各々は、複数のベクトル要素を格納するためのものであり、
前記少なくとも２つの内積のうちの第１の内積は、前記第１のソースレジスタのベクトル要素の第１のサブセットと、前記第２のソースレジスタのベクトル要素の第１のサブセットとの間で実行され、
前記少なくとも２つの内積のうちの第２の内積は、前記第１のソースレジスタのベクトル要素の第２のサブセットと、前記第２のソースレジスタのベクトル要素の第２のサブセットとの間で実行され、
前記第２のソースレジスタの前記第１および第２のサブセットは異なっており、
前記第２のソースレジスタの前記第１および第２のサブセットの少なくとも２つのベクトル要素が重複している、データ処理システム。
【請求項２】
前記第１のソースレジスタの前記第１および第２のサブセットが同じサブセットである

、請求項１に記載のデータ処理システム。
【請求項３】
前記ベクトル内積命令は、前記第１のソースレジスタのどのベクトル要素を前記第１のソースレジスタのベクトル要素の第１のサブセットに含ませるべきかを少なくとも示すために使用するオフセットをさらに示す、請求項１に記載のデータ処理システム。
【請求項４】
複数の汎用レジスタと、
１つ以上の命令を実行するためのプロセッサ回路と
を備え、前記１つ以上の命令は、少なくとも２つの内積を同時に実行するためのベクトル内積命令を含み、前記ベクトル内積命令は、前記複数の汎用レジスタから第１のソースレジスタと第２のソースレジスタとを特定し、前記第１のソースレジスタと前記第２のソースレジスタとの各々は、複数のベクトル要素を格納するためのものであり、
前記少なくとも２つの内積のうちの第１の内積は、前記第１のソースレジスタの５つのベクトル要素の第１のサブセットと、前記第２のソースレジスタの５つのベクトル要素の第１のサブセットとの間で実行され、
前記少なくとも２つの内積のうちの第２の内積は、前記第１のソースレジスタの５つのベクトル要素の第２のサブセットと、前記第２のソースレジスタの５つのベクトル要素の第２のサブセットとの間で実行され、
前記第２のソースレジスタの前記第１および第２のサブセットの４つのベクトル要素が重複している、データ処理システム。
【請求項５】
同時内積演算を実行するための方法であって、
複数の汎用レジスタを提供すること、
１つ以上の命令を実行するためのプロセッサ回路を提供すること
を備え、前記１つ以上の命令は、少なくとも２つの内積を同時に実行するためのベクトル内積命令を含み、前記ベクトル内積命令は、前記複数の汎用レジスタから第１のソースレジスタと第２のソースレジスタとを特定し、前記第１のソースレジスタと前記第２のソースレジスタとの各々は、複数のベクトル要素を格納するためのものであり、
前記少なくとも２つの内積のうちの第１の内積は、前記第１のソースレジスタのベクトル要素の第１のサブセットと、前記第２のソースレジスタのベクトル要素の第１のサブセットとの間で実行され、
前記少なくとも２つの内積のうちの第２の内積は、前記第１のソースレジスタのベクトル要素の第２のサブセットと、前記第２のソースレジスタのベクトル要素の第２のサブセットとの間で実行され、
前記第２のソースレジスタの前記第１および第２のサブセットは異なっており、
前記第２のソースレジスタの前記第１および第２のサブセットの少なくとも２つのベクトル要素が重複している、方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、一般にデータ処理システムに関し、より具体的には、データ処理システム内で重複オペランドを用いたＳＩＭＤの内積演算に関する。
【背景技術】
【０００２】
ベクトルの複数要素に対する演算の並行実行を可能にすることにより、データ処理システムの性能向上を実現することができる。例えば、単一命令複数データ（ＳＩＭＤ）スカラプロセッサ（「ショートベクトル・マシン」とも称される）は、任意の既存スカラ汎用レジスタ（ＧＰＲ）を使用しつつ、限られたベクトル処理を許容する。例えば、３２個のスカラ６４ビットＧＰＲを有するデータ処理システムにおいて、各スカラレジスタは、２つの３２ビットベクトル要素、４つの１６ビットベクトル要素、または８つの８ビットベ

クトル要素を保持することが可能であり、それによって２つの３２ビットベクトル演算、４つの１６ビットベクトル演算、または８つの８ビットベクトル演算を実行可能である。

【発明の概要】

【発明が解決しようとする課題】

【０００３】

ＳＩＭＤアーキテクチャは、線形フィルタを広範囲に使用する画像処理や他のアルゴリズムなど、様々なアルゴリズムの性能を強化するのに適している。ただし、基底のハードウェアベクトルの次元で、これらのアルゴリズム内で処理される配列次元の効率的なマッピングが許容されないと、非効率が生まれる。

【図面の簡単な説明】

【０００４】

【図１】本発明の実施形態に係るデータ処理システムをブロック図形式で表す図である。

【図２】本発明の実施形態に係る図１のデータ処理システムによって実行可能なＳＩＭＤ内積命令を表す図である。

【図３】本発明の実施形態に係るＳＩＭＤ内積演算時における図１の実行ユニットとスカラ・レジスタ・ファイルとの一部をブロック図形式で表した図である。

【図４】本発明の実施形態に係るオフセットを使用したＳＩＭＤ内積演算時における図１の実行ユニットとスカラ・レジスタ・ファイルとの一部をブロック図形式で表した図である。

【発明を実施するための形態】

【０００５】

本発明を添付の図に一例として示すが、これらは本発明を制限するものでない。添付の図において、同様の参照符号は同様の要素を表す。図内の要素は、簡潔化と明瞭化のために示されるものであり、必ずしも正しいスケールで描かれてはいない。

【０００６】

利用可能なデータ並列が多いので、ＳＩＭＤ演算は、ベクトル×行列演算や行列×行列演算の性能向上を見込むことができる。これらの演算は、画像処理アルゴリズムなど、様々なアルゴリズムで広く使用されている。例えば、現在の画像処理アルゴリズムおよび他のアルゴリズムは、特徴認識プロセスの一部として線形フィルタを広く使用する。ただし、基底のハードウェアベクトルの次元で、これらのアルゴリズム内で処理される配列または行列次元の効率的なマッピングが許容されないと、非効率が生まれることがある。例えば、現在利用可能なあるＳＩＭＤアーキテクチャは、８バイトのベクトル（各々が８バイトのベクトル要素）をサポートする。このアーキテクチャでは、上位４つのベクトル要素と下位４つのベクトル要素とで独立した内積演算の実行を処理する命令が提供される。これは、４×４の行列と４要素ベクトルでの演算では非常に効率的だが、画像処理アルゴリズムで一般に使用される３×３および５×５の行列での演算では効率が低下する。

【０００７】

例えば、多くの画像処理アルゴリズムで、演算は、あるピクセルとその直近の８個（つまり３×３行列を伴う）または近隣の２４個（つまり５×５の行列を伴う）のピクセルで実行される。５×５の線形フィルタを使用した場合には、３×３の線形フィルタと比較して、人為的な影響の少ない良好な結果が生成されるのが一般的だが、効率が悪く、必要な計算が多い。そのため、本発明の一実施形態では、現在利用可能なＳＩＭＤアーキテクチャを使用して、５×５の行列演算の効率が３×３の行列演算の効率と同等またはそれを上回るような改良型ＳＩＭＤ内積命令が提供される。一実施形態において、この効率改善は、ＳＩＭＤアーキテクチャ内で２つの追加８ビット×８ビット乗算器だけを犠牲にして実現することができる。

【０００８】

本明細書で使用されているとおり、「バス」という用語は、データ、アドレス、コントロール、またはステータスなど１つ以上の各種情報を転送する目的で使用され得る複数の信号または導体を意味する目的で使用される。本明細書で述べられている導体は、単一の

導体、複数の導体、一方向性導体、または双方向性導体であることに関連して例示または記載されている場合がある。ただし、実施形態が異なれば、導体の実装形態も異なる場合がある。例えば、双方向性導体ではなく、個別の一方向性の導体を使用することがあり、その逆もあり得る。また、複数の導体に代えて、連続方式または時間多重化方式で多重信号を転送する単一の導体を使用してもよい。同様に、多重信号を伝達する単一の導体を、これらの信号のサブセットを伝達する各種導体に分離してもよい。そのため、信号を伝送するための数々のオプションが存在する。

【０００９】

図１は、本発明の一実施形態によるデータ処理システム１０をブロック図形式で表している。データ処理システム１０は、プロセッサ１４と、メモリ１２と、入出力装置（Ｉ／Ｏ）１６と、他の周辺機器１８と、システムバス２０とを含む。メモリ１２は導体２２を介してシステムバス２０に双方向に連結され、Ｉ／Ｏ１６は導体２４を介してシステムバス２０に双方向に連結され、他の周辺機器１８は導体２６を介してシステムバス２０に双方向に連結され、プロセッサ１４は導体５８を介してシステムバス２０に双方向に連結されている。一実施形態において、他の周辺機器１８は１つ以上の周辺機器を含んでもよく、各々は、汎用非同期送受信回路（ＵＡＲＴ）、リアル・タイム・クロック（ＲＴＣ）、キーボードコントローラ、他のメモリなど、任意の種類の周辺機器であってよい。他の周辺機器１８の一部または全部は、導体６２を介してデータ処理システム１０に外部情報を通信してもよい。Ｉ／Ｏ回路１６は、例えば導体６０を介してデータ処理システム１０に外部情報を送受信する任意の種類のＩ／Ｏ回路を含んでもよい。メモリ１２は、読み取り専用メモリ（ＲＯＭ）、ランダム・アクセス・メモリ（ＲＡＭ）、不揮発性メモリ（フラッシュなど）等、任意の種類のメモリであってよい。データ処理システム１０は、図示されている以外の要素を含んでもよく、含む要素が図示されている要素より多くても少なくてもよい。例えば、データ処理システム１０は、任意の数のメモリまたはプロセッサを含んでもよい。

【００１０】

プロセッサ１４は、例えば、マイクロプロセッサ、マイクロコントローラ、デジタル信号プロセッサなど、任意の種類であってよい。一実施形態において、プロセッサ１４は、プロセッサコアまたはプロセッサ回路と呼ばれることもある。別の実施形態において、プロセッサ１４は、マルチプロセッサデータ処理システムにおける多数のプロセッサの１つであってもよい。さらに、図示されてはいないものの、プロセッサ１４はパイプライン型プロセッサであってもよい。図１に示す一実施形態において、プロセッサ１４は、制御ユニット２８と、命令ユニット３０と、実行ユニット３２と、スカラ・レジスタ・ファイル３４と、バス・インタフェース・ユニット（ＢＩＵ）３６と、ロード／ストアユニット３８とを含む。制御ユニット２８は、導体４０を介して命令ユニット３０に、導体４２を介して実行ユニット３２に、導体４６を介してスカラ・レジスタ・ファイル３４に、そして導体４８を介してロード／ストアユニット３８に双方向に連結される。実行ユニット３２は、導体４４を介してスカラ・レジスタ・ファイル３４に双方向に連結され、スカラ・レジスタ・ファイル３４は、導体５０を介してロード／ストアユニット３８に双方向に連結される。ＢＩＵ３６は、導体５４を介して命令ユニット３０に、導体５２を介してロード／ストアユニット３８に双方向に連結される。プロセッサ１４は、導体５８に連結されている導体５６を介して、システムバス２０と双方向に通信することができる。なお、プロセッサ１４は、例示されているよりも多くの回路を含んでもよく、追加回路は導体５８に連結されてもよい。すなわち、導体５６は、導体５８の全体または一部を介してシステムバス２０と通信してもよい。なお、プロセッサ１４の全体または一部は、処理回路と呼ばれることもある。

【００１１】

演算時、命令ユニット３０は、ＢＩＵ３６とシステムバス２０とを介して、メモリ１２などのメモリから命令をフェッチし、制御ユニット２８との間で制御情報を送受信する。命令ユニット３０は、従来技術において公知の任意の種類の命令ユニットであってよく、

従来技術において公知のとおりに動作するため、本明細書では詳しくは説明しない。命令ユニット３０は制御ユニット２８に命令を提供し、制御ユニット２８は、受信したこれらの命令の実行を、例えば実行ユニット３２やロード／ストアユニット３８を通じて制御する。実行ユニット３２およびロード／ストアユニット３８はともに、必要に応じて、スカラ・レジスタ・ファイル３４と直接的に、または制御ユニット２８を介して通信することができる。例えば、制御ユニット２８は、ロード／ストアユニット３８とＢＩＵ３６とを介して、命令を実行するとき必要に応じて（メモリ１２などの）メモリからスカラ・レジスタ・ファイル３４内のレジスタにデータをロードすることができるとともに、命令を実行するとき必要に応じて、スカラ・レジスタ・ファイル３４内のレジスタから（メモリ１２などの）メモリにデータをストアすることができる。例えば、一実施形態では、ロード／ストアユニット３８は、制御ユニット２８から導体４８を介して提供された制御情報に基づき、導体５０を介してスカラ・レジスタ・ファイル３４と直接通信する（それによってデータを読み書きする）ことができる。実行ユニット３２は、スカラ・レジスタ・ファイル３４内に記憶（ストア）されたデータを使用して、算術、論理、シフト、または他の演算を実行することができ、制御ユニット２８を経由して命令ユニット３０から受信した命令を実行するために、必要に応じてスカラ・レジスタ・ファイル３４内のレジスタに結果をストアすることができる。実行ユニット３２は、例えば、算術論理ユニット（ＡＬＵ）や浮動小数点ユニット等を含んでもよく、これらのユニットは、例えば、乗算器、加算器、アキュムレータ、中間結果用の記憶装置等を含んでもよい。

【００１２】

スカラ・レジスタ・ファイル３４は、Ｎ個（Ｎは１以上の任意の整数）の汎用レジスタ（ＧＰＲ）を含む。一実施形態において、スカラ・レジスタ・ファイル３４は、３２個の６４ビット・レジスタを含む。本明細書で使用されるスカラレジスタは、１つの１次元マップを持ち、したがって１行のデータだけを保持するレジスタ（１×Ｍビットレジスタなど）を表す。Ｍは１以上の任意の整数を取り得る。一実施形態において、Ｍは６４であり、したがって各レジスタは、６４ビット量を格納することができる。スカラ・レジスタ・ファイル３４は、導体４６を介して、制御ユニット２８との間で制御情報またはデータを送受信することができる。

【００１３】

プロセッサ１４の動作は一般に当業者であれば理解し得る。そのため、本明細書では、プロセッサ１４について、図２から図４を参照して記載されている様々な実施形態を理解する上で必要な部分を除き、さらに詳しくは記載しない。また、スカラ汎用レジスタファイルに格納されているオペランドを有するデータ処理システムの既存設計は、本明細書に記載されている内積命令を実行するために、必要に応じて変更してもよいという点に注意されたい。さらに、スカラ・レジスタ・ファイルを使用してもよいことから、現在の既存デザインを、本明細書に記載される命令を許容するように改変してもよいという点にも注意されたい（ただし、本明細書に記載されている実施形態は、任意の種類のレジスタファイルで使用してもよく、スカラ・レジスタ・ファイルだけに限定されないという点に注意されたい）。

【００１４】

図２は、図１のプロセッサ１４などの処理回路によって実行され得る内積命令を表している。例えば、この命令は、この命令を制御ユニット２８に適宜提供する命令ユニット３０によってフェッチされる。そのため、制御ユニット２８は、後でさらに詳述するように、受信した命令を実行するように必要に応じてロード／ストアユニット３８と実行ユニット３２とに対して指示し、データをストアするために必要に応じてスカラ・レジスタ・ファイル３４を使用することができる。なお、本明細書で使用されるベクトル要素（またはレジスタ要素）は、最大でもスカラＧＰＲのサイズまでの要素を表すが、ＧＰＲのサイズより小さいこともある。例えば、スカラ・レジスタ・ファイル３４が６４ビット・レジスタ（Ｍ＝６４）を含む場合には、ベクトル要素が６４ビット以下のサイズということもある。例えば、１つの６４ビットＧＰＲが８つのベクトル要素を保持できるように、ベクト

ル要素が１バイト（８ビット）ということもある。また、１つの６４ビットＧＰＲが４つのベクトル要素を保持できるように、ベクトル要素がハーフワード（１６ビット）ということもある。同様に、１つの６４ビットＧＰＲが２つの要素を保持できるように、ベクトル要素がワード（３２ビット）ということもある。また、本明細書で使用されているとおり、バイトは「ｂ」、ハーフワードは「ｈ」、ワードは「ｗ」で表されるという点に注意されたい（なお、代替実施形態において、ワードまたはハーフワードの定義が異なる場合もあり、例えば、ワードが３２ビットではなく１６ビットを表すことがあるが、本明細書では、説明を容易にするために、ワードは３２ビットを表す）。

【００１５】

図２は、内積命令ｅｖｄｏｔｐ５ｂ［ａ］を示す。この命令は、２つの同時５バイトベクトル内積を実行する。一実施形態において、ｅｖｄｏｔｐ５ｂ［ａ］は、宛先レジスタ（ｒＤ）と、２つのソースレジスタ（ｒＡおよびｒＢ）と、オフセットとを指定することができる３２ビット命令である。命令の末尾に「ａ」がある場合（ｅｖｄｏｔ５ｂａなど）には蓄積を表し、命令の末尾に「ａ」がない場合（ｅｖｄｏｔ５ｂなど）には蓄積を表さない。図２の実施形態に示すとおり、ｅｖｄｏｔｐ５ｂ［ａ］命令は、演算コード、宛先レジスタ（ｒＤ）、２つのソースレジスタ（ｒＡおよびｒＢ）、オフセット、Ａビット、サブ演算コードなど、様々なフィールドを含む。なお、代替実施形態において、フィールドの配置が異なってもよく、異なる数のビットを使用して、命令と、図２に示される配置以外の様々なフィールドの各々とを定義してもよい。

【００１６】

図２の命令は、２つの５バイト同時内積演算を実行する。宛先ｒＤのワードごとに、ｒＡにおける５バイトペアの符号付き整数ベクトル要素と、ｒＢにおける符号無し整数ベクトル要素とが乗算され、５つの１６ビットの中間積を生成する。これらの中間積は、３２ビットまで符号拡張が可能で、合算されて２つの和を生成する。命令に「ａ」がない場合には、蓄積が実行されないため、中間積の２つの合計の各々はｒＤの対応ワードに配置される。すなわち、２つの合計のうちの一方は、ビット位置０～３１など、ｒＤの第１のワード要素に格納されるのに対し、２つの命令結果の他方は、ビット位置３２～６３など、ｒＤの第２のワード要素に格納される。命令に「ａ」がある場合には、蓄積が実行される。この場合、２つの和の各々は、アキュムレータ（ＡＣＣ１またはＡＣＣ２）の対応ワードに追加されて、ｒＤの対応ワードに格納される。ｒＤでの結果も、その後アキュムレータに配置される。

【００１７】

一実施形態において、ｒＡの５つのベクトル要素は、同じものが両方の同時内積演算で使用されるのに対し、ｒＢの５つのベクトル要素は、２つの異なるサブセットが２つの同時内積演算に使用される。すなわち、この実施形態においては、ｒＢの５つのベクトル要素の第１のサブセットが２つの同時内積演算のうちの一方に使用され、ｒＢの５つのベクトル要素の第２のサブセットが２つの同時内積演算の他方に使用される。一実施形態において、第１のサブセットおよび第２のサブセットは、重複しているベクトル要素を含む。例えば、図３を参照して後述するとおり、ｒＡでは最初の５つのベクトル要素が同時内積演算の両方に使用できるのに対し、ｒＢの第１のサブセットは、ｒＢの最初の５つのベクトル要素を含むことができ、ｒＢの第２のサブセットは、ｒＢの第２から第６のベクトル要素をｒＢに含むことができる。なお、この実施形態においては、同時内積演算に使用されるｒＢのベクトル要素の２つのサブセットが、ｒＢ内で１要素ずつ相互にシフトされるだけである。そのため、ｒＢの第１および第２のサブセットにおける５つのベクトル要素のうち、４つは重複する。一実施形態において、２つの同時内積演算に使用されるオペランドは、ｒＡの係数セットとｒＢのデータサンプルとから２つの出力値を計算して、５×５のフィルタリング演算を支援するように選択されてもよい。一実施形態において、第１のピクセル値Ｘ（ｒＢの第３のベクトル要素に対応する第１のピクセル値Ｘ）と同じ行にある直近の近隣値が、１つの計算内積に関与してもよく、その一方で、第２のピクセル値Ｙ（ｒＢの第４の要素に対応する第２のピクセル値Ｙ）の直近の近隣値が、第２の同時内

積演算に関与して、２つの独立出力値を生成してもよい。この場合、ＸとＹとの隣接値が重複するため、ｒＢの重複するベクトル要素は、同時内積計算に関与している。

【００１８】

オフセットフィールドは、ｒＡのどの５つのベクトル要素が内積演算のために選択されるべきかを表す。すなわち、オフセットがゼロである場合やオフセットが存在しない場合には、（図３の実施例に示すとおり）第１の５つのベクトル要素が用いられる。ただし、オフセットが２である場合には、（図４の実施例に示すとおり）第３から第７のベクトル要素が使用される。２というオフセット（オフセット＝２）を使用する図４を参照して記載されるとおり、ｒＡの第３から第７のベクトル要素が両方の同時内積演算に使用されるのに対し、ｒＢの第１のサブセットは、ｒＢの第３から第７のベクトル要素を含み、ｒＢの第２のサブセットは、ｒＢの第４から第８のベクトル要素を含む。なお、図３と図４との実施例において、ｒＡの５つのベクトル要素とｒＢの第１のサブセットの５つのベクトル要素とは、同じベクトル要素位置（すなわち図３の第１の５つのベクトル要素および図４の第３から第７までのベクトル要素）に対応する。ただし、代替実施形態において、ｒＢ用の追加オフセットフィールドを使用するなど、このことが当てはまらない場合もある。代替実施形態においては、ｒＡのベクトル要素およびｒＢのベクトル要素用の独立オフセットフィールドが指定されてもよい。加えて、代替実施形態が提供するｒＢの要素の第１および第２のサブセットの重複の度合は異なってもよい。

【００１９】

例示されている実施形態において、ｒＡ、ｒＢ、およびｒＤの各々は、スカラ・レジスタ・ファイル３４の内の６４ビット・レジスタの１つである。また、図２の実施形態において、ソースレジスタｒＡは５つの符号付き整数要素を提供し、ソースレジスタｒＢは５つの符号無し整数要素を提供する。ただし、代替実施形態においては、ｒＡとｒＢとの各々が符号付きまたは符号無しベクトル要素を格納することができ、ｒＡとｒＢとの各々が分数または整数ベクトル要素を格納することができるという点に注意されたい。そのため、様々な演算コードまたはサブ演算コードエンコーディングを使用して、ｒＡとｒＢとの各々が符号付きなのか符号無しなのか、あるいは分数なのか整数なのかを表してもよい。あるいは、ｅｖｄｏｔｐ５ｂ［ａ］命令の追加フィールドを使用して、ｒＡとｒＢとの各々が符号付きなのか符号無しなのか、あるいは分数なのか整数なのかを表してもよい。また、様々な演算コードまたはサブ演算コードエンコーディングあるいは追加フィールドを使用して、中間積が剰余積なのか飽和積なのか、あるいは中間積の和が実行されるのか差が実行されるのかを表してもよい。また、代替実施形態においては、オフセットフィールドが存在しなくてもよく、その場合には、ｒＡの第１の５つのベクトル要素が常に使用される。さらに別の実施形態では、第２のオフセットフィールドが提供されて、ｒＢのどの要素が選択されるべきかを表してもよい。また、命令のフィールドの明示に使用される命令フォーマットは様々であってよいという点に注意されたい。例えば、様々なオプションの組み合せを提供する様々な命令を一斉に使用したり、命令内の追加フィールドを使用して、ユーザが様々なオプションを選択できるようにしたりしてもよい。ｅｖｄｏｔｐ５ｂ［ａ］の様々な演算例について、図３および図４を参照して以下説明する。

【００２０】

図３は、実行ユニット３２およびスカラ・レジスタ・ファイル３４の一部と、（蓄積が実行され、オフセットフィールドが存在しないか、ゼロとして提供される）ｅｖｄｏｔｐ５ｂａ命令の演算を表すデータフロー例とを示す。図３は、ソースレジスタｒＡを表すレジスタ６６と、ソースレジスタｒＢを表すレジスタ６４と、第１のワードであるＷＯＲＤ１および第２のワードであるＷＯＲＤ２を有する宛先レジスタｒＤを表すレジスタ９４とを含む。図３はまた、中間積を格納するための記憶位置６８および７０と、内積を格納するための記憶位置８６とを含む。図３はまた、第１のワードであるＡＣＣ１と第２のワードであるＡＣＣ２とを含むアキュムレータ８８も含む。図３はまた、２つの同時内積演算の一方を実行するのに使用される乗算器７１～７５および加算器８２と、２つの同時内積演算の他方の実行に使用される乗算器７６～８０および加算器８４とを含む。図３はまた

、内積結果の各々にアキュムレータ８８の値を加算してｒＤを更新する目的で使用される加算器９０，９２を含む。なお、ｒＤが更新されると、アキュムレータ８８も、更新されたｒＤの値で更新される。

【００２１】

演算時に、ｒＡ（レジスタ６６）は８つのベクトル要素ａ０～ａ７を格納し、ｒＢ（レジスタ６４）は８つのベクトル要素ｂ０～ｂ７を格納する。図３の実施例はオフセットを指定しないか、オフセットとしてゼロを使用するため、同時内積演算の両方がｒＡ（ａ０～ａ４）の第１の５つのベクトル要素を使用する。そのため、２つの同時内積演算の一方はｒＢの第１の５つのベクトル要素（ｂ０～ｂ４）を使用するのに対し、２つの同時内積演算の他方は、ｒＢの次の５つのベクトル要素（ｂ１～ｂ５）を使用する。ｂ１～ｂ５は、ｂ０～ｂ４に対して１要素シフトしたものである。なお、ｂ０～ｂ４は、ｒＢの第１のサブセットと呼ばれることがあり、ｂ１～ｂ５はｒＢの第２のサブセットと呼ばれることがある。記憶位置７０は、ａ０～ａ４およびｂ０～ｂ４という積の対に対応する５つの中間積を格納する。すなわち、乗算器７６はａ０をｂ０倍して、結果を記憶位置７０の第１のフィールドに格納し、乗算器７７はａ１をｂ１倍して、結果を記憶位置７０の第２のフィールドに格納し、乗算器７８はａ２をｂ２倍して、結果を記憶位置７０の第３のフィールドに格納し、乗算器７９はａ３をｂ３倍して、結果を記憶位置７０の第４のフィールドに格納し、乗算器８０はａ４をｂ４倍して、結果を記憶位置７０の第５のフィールドに格納する。次に、これら５つの中間積は加算器８４によって合計され、得られた和は記憶位置８６の第１のワードに格納される。記憶位置６８は、ａ０～ａ４およびｂ１～ｂ５という積の対に対応する５つの中間積を格納する。すなわち、乗算器７１はａ０をｂ１倍して、結果を記憶位置６８の第１のフィールドに格納し、乗算器７２はａ１をｂ２倍して、結果を記憶位置６８の第２のフィールドに格納し、乗算器７３はａ２をｂ３倍して、結果を記憶位置６８の第３のフィールドに格納し、乗算器７４はａ３をｂ４倍して、結果を記憶位置６８の第４のフィールドに格納し、乗算器７５はａ４をｂ５倍して、結果を記憶位置６８の第５のフィールドに格納する。次に、これらの５つの中間積は加算器８２によって合計され、得られた和は記憶位置８６の第２のワードに格納される。

【００２２】

そのため、レジスタ８６は、第１の結果用の乗算器７６～８０および加算器８４と、第２の結果用の乗算器７１～７５および加算器８２とを使用して同時に実行された２つの内積結果を格納するという点に注意されたい。その後加算器９０を使用して、（ａ０～ａ４とｂ０～ｂ４との内積から得られた）第１の内積結果を、アキュムレータ８８の第１のワードに格納されている対応アキュムレータ値ＡＣＣ１に加算する。得られた和は、ｒＤ（レジスタ９４）の第１の対応ワードＷＯＲＤ１に格納される。同様に、加算器９２を（加算器９０によって実行される加算と同時に）使用して、（ａ０～ａ４とｂ１～ｂ５との内積から得られた）第２の内積結果を、アキュムレータ８８の第２のワードに格納されている対応アキュムレータ値ＡＣＣ２に加算する。得られた和は、ｒＤの第２の対応ワードであるＷＯＲＤ２に格納される。その後、ｒＤに格納された値をアキュムレータ８８に格納して、ＡＣＣ１とＡＣＣ２との値を新しい結果で更新することができる。

【００２３】

なお、蓄積が実行されない実施形態においては、図３のレジスタ８６がｒＤを表す。ｒＤは、２つの同時内積演算の結果を直接格納する。

図４は、実行ユニット３２およびスカラ・レジスタ・ファイル３４の一部と、（蓄積が実行され、オフセットフィールドが２に設定される）ｅｖｄｏｔｐ５ｂａ命令の演算を表す別のデータフロー例とを示す。図４の演算は、図３の演算と類似しており、同様の数字は同様の要素を表す。図４のデータフローは、オフセットとして（ｒＡにおけるベクトル要素のオフセットに対応する）２が使用されることを除き、図３のデータフローと類似している。すなわち、ａ０～ａ４およびｂ０～ｂ４と、ａ０～ａ４およびｂ１～ｂ４との同時内積を実行する図３の実施例とは異なり、図４の実施例は、ａ２～ａ６およびｂ２～ｂ６と、ａ２～ａ６およびｂ３～ｂ７との同時内積を実行する。すなわち、使用されている

ｒＡの５つのベクトル要素がａ０から２要素分オフセットされることに注意されたい。そのため、図３の実施例の対応サブセットと比較して、ｒＢの第１のサブセットの５つのベクトル要素と、ｒＢの第２のサブセットの５つのベクトル要素とは、２要素分オフセットされている。なお、図３の実施例に示すとおり、ｒＢの第１のサブセット（ｂ２－ｂ６）と比較して、ｒＢの第２のサブセット（ｂ３－ｂ７）は、ｒＢの５つのベクトル要素のうちの４が２つの同時内積演算で重複するように１要素分シフトしている。

【００２４】

図４の実施例を参照すると、記憶位置７０は、ａ２～ａ６およびｂ２～ｂ６という積の対に対応する５つの中間積を格納する。すなわち、乗算器７６はａ２をｂ２倍して、結果を記憶位置７０の第１のフィールドに格納し、乗算器７７はａ３をｂ３倍して、結果を記憶位置７０の第２のフィールドに格納し、乗算器７８はａ４をｂ４倍して、結果を記憶位置７０の第３のフィールドに格納し、乗算器７９はａ５をｂ５倍して、結果を記憶位置７０の第４のフィールドに格納し、乗算器８０はａ６をｂ６倍して、結果を記憶位置７０の第５のフィールドに格納する。次に、これら５つの中間積は加算器８４によって合計され、得られた和は記憶位置８６の第１のワードに格納される。記憶位置６８は、ａ２～ａ６およびｂ３～ｂ７という積の対に対応する５つの中間積を格納する。すなわち、乗算器７１はａ２をｂ３倍して、結果を記憶位置６８の第１のフィールドに格納し、乗算器７２はａ３をｂ４倍して、結果を記憶位置６８の第２のフィールドに格納し、乗算器７３はａ４をｂ５倍して、結果を記憶位置６８の第３のフィールドに格納し、乗算器７４はａ５をｂ６倍して、結果を記憶位置６８の第４のフィールドに格納し、乗算器７５はａ６をｂ７倍して、結果を記憶位置６８の第５のフィールドに格納する。次に、これらの５つの中間積は加算器８２によって合計され、得られた和は記憶位置８６の第２のワードに格納される。

【００２５】

そのため、レジスタ８６は、第１の結果用の乗算器７６～８０および加算器８４と、第２の結果用の乗算器７１～７５および加算器８２とを使用して同時に実行された２つの内積結果を格納するという点に注意されたい。その後加算器９０を使用して、（ａ２～ａ６とｂ２～ｂ６との内積から得られた）第１の内積結果を、アキュムレータ８８の第１のワードに格納されている対応アキュムレータ値ＡＣＣ１に加算する。得られた和は、ｒＤ（レジスタ９４）の第１の対応ワードＷＯＲＤ１に格納される。同様に、加算器９２を（加算器９０によって実行される加算と同時に）使用して、（ａ２～ａ６とｂ３～ｂ７との内積から得られた）第２の内積結果を、アキュムレータ８８の第２のワードに格納されている対応アキュムレータ値ＡＣＣ２に加算する。得られた和は、ｒＤの第２の対応ワードであるＷＯＲＤ２に格納される。その後、ｒＤに格納された値をアキュムレータ８８に格納して、ＡＣＣ１とＡＣＣ２との値を新しい結果で更新することができる。

【００２６】

なお、蓄積が実行されない実施形態においては、図４のレジスタ８６がｒＤを表す。ｒＤは、２つの同時内積演算の結果を直接格納する。

なお、ベクトル要素を８つずつ格納するレジスタを使用して効率的な３×３または４×４の行列演算を提供しているシステムでは、ソースレジスタの４要素の対応する互いに素な（すなわち重複しない）セットに対して１対の内積演算が実行される場合には、８つの加算器だけが提供されるのが一般的である。ただし、このようなシステムは、５×５の行列演算（あるいは８つのベクトル要素を格納するレジスタと適合しない他の次元）では非効率的である。そのため、本明細書に記載のとおり、２つの追加乗算器（例えば７５および８０）とソースレジスタｒＢの重複サブセットを使用することにより、５×５の内積演算の効率が改善されることがある点に注意されたい。これらの種類の演算は、５×５の行列演算に大きく依存するアプリケーションで特に有用となり得る。他の次元の行列演算に大きく依存することのある他の種類の演算においては、少数の追加乗算器があり、ａ０～ａＮおよびｂ０～ｂＮと、ａ０～ａＮおよびｂ１～ｂ（Ｎ＋１）との同時内積を実行できるｅｖｄｏｔｐ５ｂ［ａ］命令と同様の命令を使用することができる。また、代替実施形

態において、ｒＡおよびｒＢの様々なサブセットを使用することができる。例えば、図３で提供されているサブセット例ｒＡとｒＢとではなく、ｒＡの第１のサブセットおよびｒＢの第１のサブセットと、ｒＡの第２のサブセットおよびｒＢの第２のサブセットとの同時内積を実行してもよく、ｒＡのサブセットの各々は、同じサブセットであっても、重複する要素を有する異なるサブセットであってもよく、ｒＢのサブセットの各々も、同じサブセットであっても、重複する要素を有する異なるサブセットであってもよい。また、ｒＢの要素にオフセットが追加されてもよい。例えば図３では、ｒＢに対してオフセットとして２が指定された場合、ｒＢの第１のサブセットがｂ０～ｂ４で、第２のサブセットがｂ１～ｂ５となるのではなく、２つの内積演算の一方の第１のサブセットがｂ２～ｂ６で、他方の内積演算の第２のサブセットがｂ３～ｂ７となる。また、重複指定子が、ｒＢの要素に追加されて、ｒＢの第２のサブセットがｒＢの第１のサブセットに対してどれだけのベクトル要素分シフトするかを表してもよい。例えば図３では、ｒＢに対して２という重複指定子が指定された場合、ｒＢの第１のサブセットがｂ０～ｂ４で、第２のサブセットがｂ１～ｂ５となるのではなく、２つの内積演算の一方の第１のサブセットがｂ０～ｂ４で、他方の内積演算の第２のサブセットがｂ２～ｂ６となる。

【００２７】

以上により、重複するオペランドを２つの同時内積演算で使用する場合に、５つのベクトル要素の同時内積演算が２つ実行される５×５の内積演算など、一部の行列演算の効率を改善する命令と回路とが提供されることが理解されるべきである。すなわち、８つのベクトル要素レジスタを使用した効率的な３×３や４×４の行列演算の場合には、８つの乗算器だけがシステムで提供されるのが一般的であるが、２つの追加乗算器の存在と、重複するオペランドの使用とにより、８つのベクトル要素レジスタを使用したさらに効率的な５×５の行列演算が実現される。

【００２８】

記載されているｅｖｄｏｔｐ５ｂ［ａ］命令の変形に加え、本発明の実施形態は、１対の４×４内積計算を伴う従来の内積演算を実行するために、他の公知の命令をサポートしてもよい。４×４内積演算時には、追加乗算器が使用されないため、不要なエネルギー消費を防ぐために電源を落としたり、ゲート制御を行ったりしてもよい。追加乗算器と、関連する加算回路との有効化を、実行される演算の種類に基づいて条件化することによって、データ処理システムの電源消費を最適化してもよい。あるいは、追加乗算器の出力が実行中の命令の一部として必要でない場合には、この出力を、ゼロなど所定の出力値に強制的に制限してもよい。

【００２９】

一実施形態において、データ処理システムは、複数の汎用レジスタと、１つ以上の命令を実行するためのプロセッサ回路とを備える。１つ以上の命令は、少なくとも２つの内積を同時に実行するためのベクトル内積命令を含む。ベクトル内積命令は、複数の汎用レジスタから第１のソースレジスタと第２のソースレジスタとを特定する。第１のソースレジスタと第２のソースレジスタとの各々は、複数のベクトル要素を格納するためのものである。少なくとも２つの内積のうちの第１の内積は、第１のソースレジスタのベクトル要素の第１のサブセットと、第２のソースレジスタのベクトル要素の第１のサブセットとの間で実行される。少なくとも２つの内積のうちの第２の内積は、第１のソースレジスタのベクトル要素の第２のサブセットと、第２のソースレジスタのベクトル要素の第２のサブセットとの間で実行される。第２のソースレジスタのベクトル要素の第１および第２のサブセットは異なっており、第２のソースレジスタの第１および第２のサブセットの少なくとも２つのベクトル要素は重複している。

【００３０】

さらなる実施形態において、ベクトル内積命令は、第１の内積の結果と第２の内積の結果とを格納するための宛先レジスタをさらに特定する。

さらに別の実施形態において、プロセッサ回路はアキュムレータをさらに含み、ベクトル内積命令は、第１の内積の結果とアキュムレータの第１の値との和と、第２の内積の結

果とアキュムレータの第２の値との和とを格納するための宛先レジスタをさらに特定する。

【００３１】

さらに別の実施形態において、第１のソースレジスタの第１および第２のサブセットは同じサブセットである。

さらに別の実施形態において、第１のソースレジスタのベクトル要素の第１のサブセットは、第２のソースレジスタのベクトル要素の第１のサブセットと同じベクトル要素位置に対応している。

【００３２】

さらに別の実施形態において、ベクトル内積命令は、第１のソースレジスタのどのベクトル要素を第１のソースレジスタのベクトル要素の第１のサブセットに含ませるべきかを少なくとも示す際に使用するオフセットをさらに示す。さらにまた別の実施形態において、ベクトル内積命令は、第２のソースレジスタのどのベクトル要素を第２のソースレジスタのベクトル要素の第１のサブセットに含ませるべきかを少なくとも示す際に使用する第２のオフセットをさらに示す。

【００３３】

さらに別の実施形態において、ベクトル内積命令は、第２のソースレジスタのどのベクトル要素を第２のソースレジスタのベクトル要素の第１のサブセットに含ませるべきかを少なくとも示す際に使用するオフセットをさらに示す。

【００３４】

別の実施形態において、データ処理システムは、複数の汎用レジスタと、１つ以上の命令を実行するためのプロセッサ回路とを備える。１つ以上の命令は、少なくとも２つの内積を同時に実行するためのベクトル内積命令を含む。ベクトル内積命令は、複数の汎用レジスタから第１のソースレジスタと第２のソースレジスタとを特定する。第１のソースレジスタと第２のソースレジスタとの各々は、複数のベクトル要素を格納するためのものである。少なくとも２つの内積のうちの第１の内積は、第１のソースレジスタの５つのベクトル要素の第１のサブセットと、第２のソースレジスタの５つのベクトル要素の第１サブセットとの間で実行される。少なくとも２つの内積のうちの第２の内積は、第１のソースレジスタの５つのベクトル要素の第２のサブセットと、第２のソースレジスタの５つのベクトル要素の第２サブセットとの間で実行される。第２のソースレジスタの第１および第２のサブセットの４つのベクトル要素が重複している。

【００３５】

別の実施形態のさらなる実施形態において、ベクトル内積命令は、第１の内積の結果と第２の内積の結果とを格納するための宛先レジスタをさらに特定する。

他の実施形態のさらに別の実施形態において、プロセッサ回路はアキュムレータをさらに含み、ベクトル内積命令は、第１の内積の結果とアキュムレータの第１の値との和と、第２の内積の結果とアキュムレータの第２の値との和とを格納するための宛先レジスタをさらに特定する。

【００３６】

他の実施形態のさらに別の実施形態において、第１のソースレジスタの第１および第２のサブセットは同じサブセットである。

他の実施形態のさらに別の実施形態において、第１のソースレジスタのベクトル要素の第１のサブセットは、第２のソースレジスタのベクトル要素の第１のサブセットと同じベクトル要素位置に対応している。

【００３７】

他の実施形態のさらに別の実施形態において、ベクトル内積命令によって特定される第１および第２のソースレジスタの各々は、８つのベクトル要素を格納するためのものであり、プロセッサ回路は１０個の乗算器を備え、そのうちの５つは、第１の内積を実行するためのものであり、残りの５つは、第２の内積を実行するためのものである。

【００３８】

別の実施形態のさらに別の実施形態において、ベクトル内積命令は、第１または第２のソースレジスタのどのベクトル要素が、第１または第２のソースレジスタのベクトル要素の第１のサブセットに含まれるべきかを少なくとも示す際に使用するオフセットをさらに示す。

【００３９】

さらに別の実施形態において、同時内積演算を実行するための方法は、複数の汎用レジスタを提供すること、および１つ以上の命令を実行するためのプロセッサ回路を提供することを備える。１つ以上の命令は、少なくとも２つの内積を同時に実行するためのベクトル内積命令を含む。ベクトル内積命令は、複数の汎用レジスタから第１のソースレジスタと第２のソースレジスタとを特定する。第１のソースレジスタと第２のソースレジスタとの各々は、複数のベクトル要素を格納するためのものである。少なくとも２つの内積のうちの第１の内積は、第１のソースレジスタのベクトル要素の第１のサブセットと、第２のソースレジスタのベクトル要素の第１のサブセットとの間で実行される。少なくとも２つの内積のうちの第２の内積は、第１のソースレジスタのベクトル要素の第２のサブセットと、第２のソースレジスタのベクトル要素の第２のサブセットとの間で実行される。第２のソースレジスタの第１および第２のサブセットは異なっており、第２のソースレジスタの第１および第２のサブセットの少なくとも２つのベクトル要素は重複している。

【００４０】

さらに別の実施形態のさらなる実施形態において、ベクトル内積命令は、第１の内積の結果と第２の内積の結果とを格納するための宛先レジスタをさらに特定する。

さらに別の実施形態のさらに別の実施形態において、プロセッサ回路はアキュムレータをさらに含み、ベクトル内積命令は、第１の内積の結果とアキュムレータの第１の値との和と、第２の内積の結果とアキュムレータの第２の値との和とを格納するための宛先レジスタをさらに特定する。

【００４１】

さらに別の実施形態のさらに別の実施形態において、第１のソースレジスタの第１および第２のサブセットは同じサブセットである。

さらに別の実施形態のさらに別の実施形態において、ベクトル内積命令は、第１または第２のソースレジスタのどのベクトル要素が、第１または第２のソースレジスタのベクトル要素の第１のサブセットに含まれるべきかを少なくとも示す際に使用するオフセットをさらに示す。

【００４２】

本発明を実装する装置は、ほとんどの場合、当業者に公知の電子部品と回路とから成るので、回路の詳細は、本発明の基底概念の理解と認識のため、および本発明の教示の混乱または逸脱を防ぐために上記のとおり必要とみなされる程度以上には説明しない。

【００４３】

上記実施形態の一部は、適宜様々な情報処理システムを使用して実装してもよい。例えば、図１およびその説明は、例示的な情報処理ＳＩＭＤアーキテクチャについて記載しているものの、この例示的なアーキテクチャは、本発明の各種態様について述べる際の有用な参考情報を提供する目的でのみ提示されている。当然のことながら、アーキテクチャに関する説明は、単純化されており、本発明に従って使用できる多くの様々な種類の適切なアーキテクチャの１つにすぎない。当業者であれば、論理ブロック間の境界が単なる例示であり、代替実施形態において、論理ブロックまたは回路要素を統合したり、各種論理ブロックまたは回路要素に割り当てる機能を変更したりしてもよいことを認識し得る。

【００４４】

このように、本明細書に描写されているアーキテクチャは単なる例示であり、実際、同じ機能を実現するアーキテクチャが他にも数多く実装可能であることを理解すべきである。抽象的であるが確かな意味で、同じ機能を実現するための構成部品のいかなる配置も所望の機能が実現されるように効果的に「関連付け」られる。それ故、特定の機能を実現するために結合されている本明細書のいずれか２つの構成部品は、アーキテクチャまたは中

間構成部品に関係なく、所望の機能が実現されるように相互に「関連付け」られているものとみなすことができる。同様に、そのように関連付けられているいずれか２つのコンポーネントは、所望の機能を実現するために、相互に「動作可能に接続」または「動作可能に連結」されているとみなすことができる。

【００４５】

また、例えば一実施形態において、システム１０の図示要素は、単一の集積回路上または同一装置内に存在する回路である。あるいは、システム１０が、任意の数の別個の集積回路や、相互に接続されている別個の装置を含んでもよい。例えば、メモリ１２は、プロセッサ１４と同じ集積回路上または別個の集積回路上に位置してもよく、システム１０の他の要素とは別の周辺機器またはスレーブ内に位置してもよい。他の周辺機器１８および入出力回路１６も、別個の集積回路または装置に位置してもよい。また、例えばシステム１０またはその一部が、物理回路の、あるいは物理回路に変換可能な論理表現のソフト表示またはコード表示であってもよい。そのため、システム１０は、任意の適切な種類のハードウェア記述言語で具現化されてもよい。

【００４６】

当業者は、上述の演算の機能間の境界が単なる例示にすぎないことを認識し得る。複数の演算の機能は単一の演算に統合されてもよく、単一の演算の機能が追加演算に分散されてもよい。さらに、代替実施形態は、特定の演算に関する複数の事例を含んでもよく、演算の順序は、他の各種実施形態で変更してもよい。

【００４７】

一実施形態において、システム１０は、パーソナル・コンピュータ・システムなどのコンピュータシステムである。他の実施形態が、異なる種類のコンピュータシステムを含んでもよい。コンピュータシステムは、１人以上のユーザに独立した計算能力を与えるように設計され得る情報処理システムである。コンピュータシステムは、メインフレーム、ミニコンピュータ、サーバ、ワークステーション、パーソナルコンピュータ、ノートパッド、パーソナル携帯情報端末、電子ゲーム機、自動車および他の組み込みシステム、携帯電話および他の各種無線機器を含むがこれらに限定されない数々の形態であってよい。典型的なコンピュータシステムは、少なくとも１つの処理ユニットと、関連付けられているメモリと、いくつかの入出力（Ｉ／Ｏ）機器を含む。

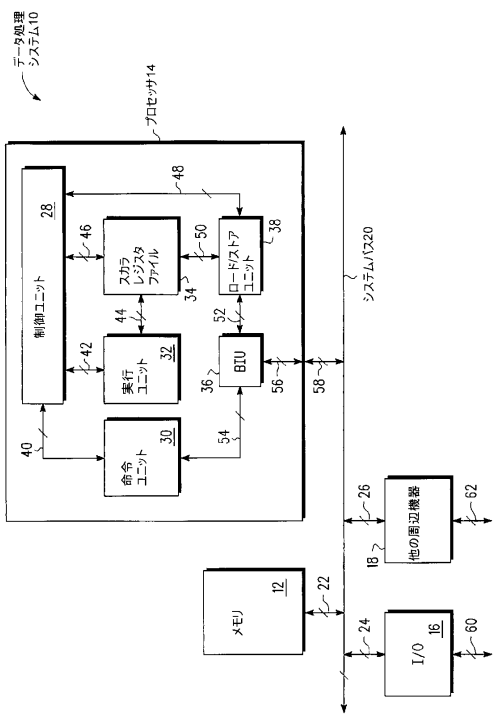
【００４８】

本発明は、本明細書において特定の実施形態を参照して記載されているものの、請求項に定める本発明の範囲を逸脱しなければ、各種変形および変更を行うことができる。例えば、他のＳＩＭＤアーキテクチャを使用したり、異なるベクトル要素のサブセットを定義したりしてもよい。したがって、仕様および図は、限定的な意味ではなく、例示的な意味で考慮されるべきであり、このような変形はすべて本発明の範囲内に含まれるものと意図される。本明細書において特定の実施形態に関して記載されているいかなる利点、効果、または課題への解決策も、任意またはすべての請求項の必須の、あるいは本質的な特徴または要素として解釈されることを意図するものではない。

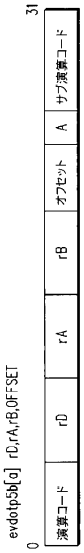
【００４９】

本明細書で使用されている「連結」という用語は、直接連結または機械式連結に限定されることを意図するものではない。本明細書で使用されている「１つの」という用語は、１つまたはそれ以上と定義される。特に明記しない限り、「第１」および「第２」などの用語は、それらの語句が修飾する要素を区別する目的で使用されている。そのため、これらの用語は、そのような要素の時間的またはその他の優先順位を表すことを必ずしも意図するものではない。

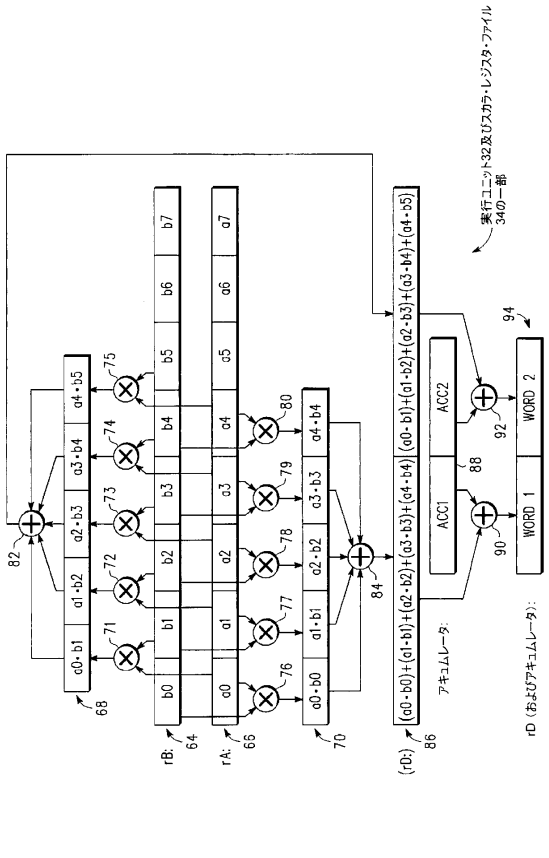
【図１】


データ処理システム10
プロセッサ14
制御ユニット 28
スカラレジスタファイル 34
ロード/ストアユニット 38
実行ユニット 32
BIU 36
命令ユニット 30
システムバス20
メモリ 12
I/O 16
他の周辺機器 62

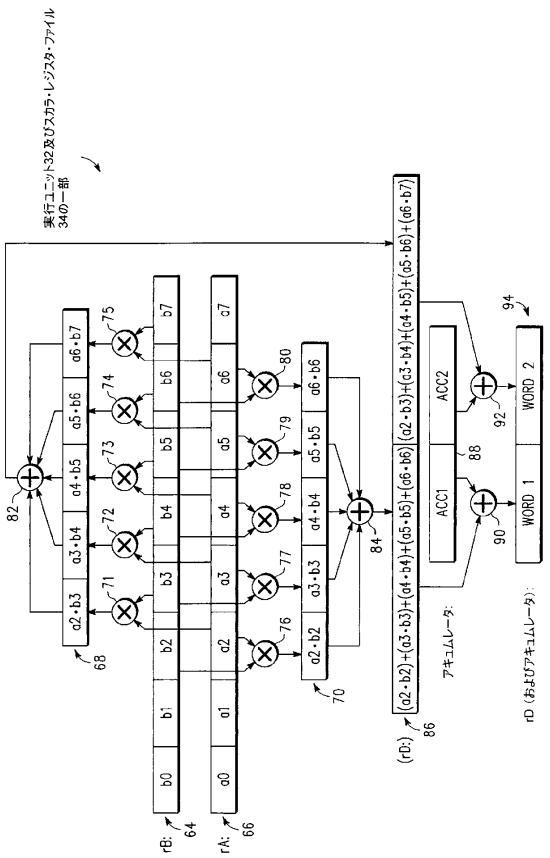
【図２】


31
0
演算コード
rD
rA
rB
オフセット
A
サブ演算コード
evdotp5b[a] rD,rA,rB,OFFSET

【図３】


実行ユニット32及びスカラ・レジスタ・ファイル
340の一部
rB:
rA:
(rD:)
rD (およびアキュムレータ):
アキュムレータ:
ACC1
ACC2
WORD 1
WORD 2

【図４】


実行ユニット32及びスカラ・レジスタ・ファイル
340の一部
rB:
rA:
(rD:)
rD (およびアキュムレータ):
アキュムレータ:
ACC1
ACC2
WORD 1
WORD 2

フロントページの続き

(56)参考文献 特開２００１－２５６１９９（ＪＰ，Ａ）
特開平１０－１２４４８４（ＪＰ，Ａ）
特開平３－７５８６８（ＪＰ，Ａ）
特開２００１－２９０６３３（ＪＰ，Ａ）
特開２０００－３２２２３５（ＪＰ，Ａ）
特開２００２－２２９９７０（ＪＰ，Ａ）
特開平５－２６７９９２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 17/16