

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号
特表2025-511829
(P2025-511829A)

(43)公表日 令和7年4月16日(2025.4.16)

(51)国際特許分類
G 0 6 F 17/16 (2006.01)

F I
G 0 6 F 17/16
G 0 6 F 17/16

テーマコード (参考)
5 B 0 5 6

D
M

審査請求 未請求 予備審査請求 未請求 (全32頁)

(21)出願番号	特願2024-559352(P2024-559352)	(71)出願人	500395107 アーム・リミテッド グレート・ブリテン及び北部アイルラン ド連合王国、イングランド、シー・ビー ー1、9エヌ・ジェイ、ケンブリッジ、 フルボーン・ロード 1 1 0
(86)(22)出願日	令和5年3月13日(2023.3.13)	(74)代理人	110000855 弁理士法人浅村特許事務所
(85)翻訳文提出日	令和6年10月4日(2024.10.4)	(72)発明者	グラセット、アルノー フィリップ クロ ード フランス共和国 0 6 4 1 0 ソフィア アンティボリス ビオット アベニュー ド ルーマニル 7 3 8 アーム フランス エ スエーエス宛
(86)国際出願番号	PCT/GB2023/050584	(72)発明者	ミラノヴィッチ、エレナ 最終頁に続く
(87)国際公開番号	WO2023/199014		
(87)国際公開日	令和5年10月19日(2023.10.19)		
(31)優先権主張番号	2205491.0		
(32)優先日	令和4年4月13日(2022.4.13)		
(33)優先権主張国・地域又は機関	英国(GB)		
(81)指定国・地域	AP(BW,CV,GH,GM,KE,LR,LS,MW,MZ ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW), EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES, FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV 最終頁に続く		

(54)【発明の名称】 アレイストレージに記憶されたデータ要素を処理するための技術

(57)【要約】

演算を実行する処理回路と、命令によって指定された演算を実行するように処理回路を制御するために命令をデコードする命令デコーダ回路と、データ要素を記憶するストレージ要素を備えるアレイストレージとを備える装置が提供される。アレイストレージは、演算を実行するときに処理回路にアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元である。命令デコーダ回路は、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、処理回路を制御して、1つ以上の識別されたベクトルのデータ要素をアレイストレージから宛先ストレージに移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用されたアレイストレージのストレージ要素を論理ゼロ値に設定するように構成されている。

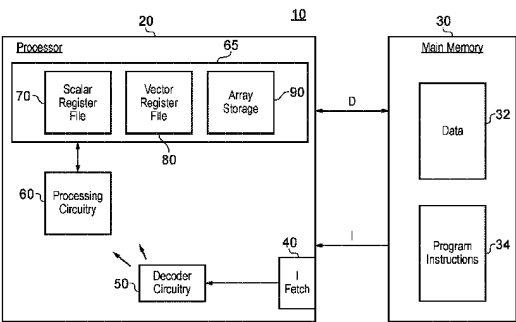


FIG. 1

【特許請求の範囲】

【請求項 1】

装置であって、

演算を実行するための処理回路と、

命令をデコードして、前記命令によって指定された前記演算を実行するように前記処理回路を制御する命令デコーダ回路と、

データ要素を記憶するためのストレージ要素を含むアレイストレージであって、前記アレイストレージは、前記演算を実行するときに前記処理回路にアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元である、アレイス

10

トレージと、を備え、
前記命令デコーダ回路は、前記アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、前記1つ以上の識別されたベクトルの前記データ要素を前記アレイストレージから宛先ストレージに移動させ、前記1つ以上の識別されたベクトルの前記データ要素を記憶するために使用された前記アレイストレージの前記ストレージ要素を論理ゼロ値に設定するように前記処理回路を制御するように構成されている、装置。

【請求項 2】

前記処理回路は、累積和演算の複数の反復を実行し、データ要素の前記所与の2次元アレイを使用して、前記累積和演算を実行するときに生成される累積和結果を維持するよう

20

に構成されており、前記累積和演算の所与の反復の後、データ要素の前記所与の2次元アレイ内のデータ要素の少なくとも1つの所与のベクトルが、最終累積和結果を記憶するように構成されている一方、データ要素の前記所与の2次元アレイ内のデータ要素の残りのベクトルが、中間累積和結果を記憶するように構成されており、
前記移動及びゼロ命令は、データ要素の前記少なくとも1つの所与のベクトルを識別するように構成されており、前記累積和演算の前記所与の反復の後に実行されて、前記処理回路に、前記少なくとも1つの所与のベクトルの前記最終累積和結果を前記アレイストレージから前記宛先ストレージに移動させ、前記少なくとも1つの所与のベクトルの前記最終累積和結果を記憶するために使用された前記アレイストレージの前記ストレージ要素を、後続の累積和演算で使用するためにそれらのストレージ要素を解放するためにクリアさ

30

【請求項 3】

累積和演算の前記複数の反復は、入力データ要素のアレイに対して有限インパルス応答（FIR）フィルタリング演算を実行するために使用される処理演算及び累積和演算であり、前記アレイストレージ内のデータ要素の前記所与の2次元アレイは、前記FIRフィルタリング演算の実行中に生成される出力データ要素のアレイを維持するために使用され

、
前記処理回路は、前記累積和演算の各反復中に、入力データ要素の単一のベクトルを処理し、出力データ要素の前記アレイの複数のベクトル内に累積するための出力データ要素を生成するように構成されている、請求項 2 に記載の装置。

40

【請求項 4】

データ要素の前記所与の2次元アレイは、データ要素の正方形2次元アレイであり、データ要素の前記正方形2次元アレイを形成する前記複数のベクトルは、第1のアレイ方向に配置された第1の複数のベクトルと、前記第1のアレイ方向に直交する第2のアレイ方向に配置された第2の複数のベクトルとを含み、前記移動及びゼロ命令の各インスタンスは、すべてが前記第1のアレイ方向に延びるか、又はすべてが前記第2のアレイ方向に延びるデータ要素の1つ以上のベクトルを識別するように構成されている、請求項 1～3 のいずれか一項に記載の装置。

【請求項 5】

前記処理回路は、データ要素の前記正方形2次元アレイに対して処理動作を実行するよ

50

うに構成されており、その間、前記処理回路は、前記第 1 のアレイ方向及び前記第 2 のアレイ方向の両方においてデータ要素のベクトルにアクセスすることを可能にされる、請求項 4 に記載の装置。

【請求項 6】

前記アレイストレージは、第 1 のアレイ方向に延びる複数のアレイベクトルレジスタを含み、前記処理回路は、1 つ以上の累積和演算を実行するように構成されており、各累積和演算は、前記アレイストレージの複数のアレイベクトルレジスタのグループ内で累積するための出力データを生成するように構成されており、

データ要素の前記所与の 2 次元アレイが、複数のアレイベクトルレジスタの前記グループ内に記憶された前記データ要素を含み、

前記 1 つ以上の累積和演算の実行の結果、前記移動及びゼロ命令によって識別された前記 1 つ以上の識別されたベクトル内に最終結果データが存在するようになると、前記処理回路に、前記 1 つ以上の識別されたベクトルの前記データ要素を前記アレイストレージから前記宛先ストレージに移動させ、前記 1 つ以上の識別されたベクトルの前記データ要素を記憶するために使用された複数のアレイベクトルレジスタの前記グループ内の各アレイベクトルレジスタを論理ゼロ値に設定するために、前記移動及びゼロ命令が実行される、請求項 1 ～ 5 のいずれか一項に記載の装置。

【請求項 7】

前記処理回路が前記 1 つ以上の累積和演算の実行を完了したとき、最終結果データが複数のアレイベクトルレジスタの前記グループ内の各アレイベクトルレジスタ内に存在し、前記移動及びゼロ命令の実行により、前記最終結果データが複数のアレイベクトルレジスタの前記グループから前記宛先ストレージに移動され、複数のアレイベクトルレジスタの前記グループ内の各アレイベクトルレジスタがゼロにクリアされる、請求項 6 に記載の装置。

【請求項 8】

複数のベクトルレジスタを含むベクトルレジスタファイルを更に備え、

前記移動及びゼロ命令は、前記宛先ストレージとして、前記ベクトルレジスタファイル内の 1 つ以上のベクトルレジスタを示すように構成されている、請求項 1 ～ 7 のいずれか一項に記載の装置。

【請求項 9】

前記移動及びゼロ命令は、前記宛先ストレージとして、前記 1 つ以上の識別されたベクトルの前記データ要素が記憶されるべきメモリ内の 1 つ以上の位置を示すように構成されている、請求項 1 ～ 7 のいずれか一項に記載の装置。

【請求項 10】

前記移動及びゼロ命令は、前記アレイストレージ内のデータ要素の前記所与の 2 次元アレイのデータ要素の前記 1 つ以上のベクトルを識別するために使用されるベクトル識別フィールドを含む、請求項 1 ～ 9 のいずれか一項に記載の装置。

【請求項 11】

前記ベクトル識別フィールドは、前記正方形 2 次元アレイを識別するための第 1 のサブフィールドと、前記 1 つ以上のベクトルを識別するために使用される 1 つ以上のライン識別子及びアレイ方向インジケーションを提供する第 2 のサブフィールドとを含む、請求項 4 を引用する請求項 10 に記載の装置。

【請求項 12】

前記移動及びゼロ命令は、前記 1 つ以上の識別されたベクトルのどのデータ要素が前記アレイストレージから前記宛先ストレージに移動されるべきかを識別し、それらの関連するストレージ要素を前記論理ゼロ値に設定させるために使用される述語情報を識別するための述語フィールドを含む、請求項 1 ～ 11 のいずれか一項に記載の装置。

【請求項 13】

前記移動及びゼロ命令が、前記 1 つ以上の識別されたベクトル内の各データ要素のサイズを識別するためのサイズフィールドを更に含む、請求項 12 に記載の装置。

【請求項 14】

装置のアレイストレージ内のデータ要素を処理する方法であって、

処理回路を使用して演算を実行することと、

命令デコーダ回路を使用して命令をデコードし、前記処理回路を制御して、前記命令によって指定された前記演算を実行することと、

前記アレイストレージであって、前記アレイストレージは、前記演算を実行するときに前記処理回路にアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元である、前記アレイストレージ内に、データ要素を記憶するためのストレージ要素を提供することと、

前記アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、前記1つ以上の識別されたベクトルの前記データ要素を前記アレイストレージから宛先ストレージに移動させ、前記1つ以上の識別されたベクトルの前記データ要素を記憶するために使用された前記アレイストレージの前記ストレージ要素を論理ゼロ値に設定するように前記処理回路を制御するように、前記命令デコーダ回路を使用することと、を含む、方法。

【請求項 15】

命令実行環境を提供するようにホストデータ処理装置を制御するためのコンピュータプログラムであって、

演算を実行する処理プログラムロジックと、

命令をデコードして、前記命令によって指定された前記演算を実行するように前記処理プログラムロジックを制御する、命令デコードプログラムロジックと、

データ要素を記憶するためのストレージ要素を含むアレイストレージをエミュレートするためのアレイストレージエミュレートプログラムロジックと、を備え、前記アレイストレージは、前記演算を実行するときに前記処理プログラムロジックにアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元であり、

前記命令デコードプログラムロジックは、前記アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、前記1つ以上の識別されたベクトルの前記データ要素を前記アレイストレージから宛先ストレージに移動させ、前記1つ以上の識別されたベクトルの前記データ要素を記憶するために使用された前記アレイストレージの前記ストレージ要素を論理ゼロ値に設定するように前記処理プログラムロジックを制御するように構成されている、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本技術は、データ処理の分野に関し、より具体的には、アレイストレージに記憶されたデータ要素の処理に関する。

【0002】

いくつかの現代のデータ処理システムは、データ処理動作を実行するときにデータ処理システムの処理回路によってアクセスされ得るデータ要素の1以上の2次元アレイを記憶するためのアレイストレージを提供することができる。これは、いくつかの異なるタイプの演算、例えば、累積和出力がデータ要素の2次元アレイ内に維持され得る累積和関数を含む演算を実行するための効率的な機構を提供することができる。

【0003】

しかしながら、そのようなアレイストレージの使用から実現され得る効率利得を最大限に利用するために、後続の動作に関連して使用するためにアレイストレージのリソースを解放するように、アレイストレージからデータ要素を転送するための効率的な機構を提供することが有益であろう。

【発明の概要】

【0004】

1つの例示的な構成では、装置が提供され、この装置は、演算を実行するための処理回路と、命令によって指定される演算を実行するために処理回路を制御するために命令をデコードするための命令デコーダ回路と、データ要素を記憶するためのストレージ要素を含むアレイストレージであって、アレイストレージは、演算を実行するときに処理回路にアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元である、アレイストレージと、を備え、命令デコーダ回路は、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、1つ以上の識別されたベクトルのデータ要素をアレイストレージから宛先ストレージに移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用されたアレイストレージのストレージ要素を論理ゼロ値に設定するように処理回路を制御するように構成されている。

10

【0005】

更なる例示的な構成では、装置のアレイストレージ内のデータ要素を処理する方法であって、処理回路を使用して演算を実行することと、命令デコーダ回路を使用して命令をデコードし、処理回路を制御して、命令によって指定された演算を実行することと、アレイストレージであって、アレイストレージは、演算を実行するときに処理回路にアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元である、アレイストレージ内に、データ要素を記憶するためのストレージ要素を提供することと、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、1つ以上の識別されたベクトルのデータ要素をアレイストレージから宛先ストレージに移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用されたアレイストレージのストレージ要素を論理ゼロ値に設定するように処理回路を制御するように、命令デコーダ回路を使用することと、を含む、方法が提供される。

20

【0006】

更なる例示的な構成では、命令実行環境を提供するようにホストデータ処理装置を制御するためのコンピュータプログラムであって、コンピュータプログラムは、演算を実行する処理プログラムロジックと、命令をデコードして、命令によって指定された演算を実行するように処理プログラムロジックを制御する、命令デコードプログラムロジックと、データ要素を記憶するためのストレージ要素を含むアレイストレージをエミュレートするためのアレイストレージエミュレートプログラムロジックと、を含み、アレイストレージは、演算を実行するときに処理プログラムロジックにアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元であり、命令デコードプログラムロジックは、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、1つ以上の識別されたベクトルのデータ要素をアレイストレージから宛先ストレージに移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用されたアレイストレージのストレージ要素を論理ゼロ値に設定するように処理プログラムロジックを制御するように構成されている、コンピュータプログラムが提供される。

30

40

【図面の簡単な説明】

【0007】

本技術は、添付の図面に示された本技術の例を参照して、例示としてのみ更に説明される。

【図1】例示的な一実装形態による装置のブロック図である。

【図2】ベクトルオペランドを記憶するためのベクトルレジスタと、データ要素の2Dア

50

レイを記憶するためのアレレジスタとを含む、装置内に提供され得るアーキテクチャレジスタの一例を示す図であり、アレレジスタの物理的実装形態の一例を含む。

【図 3 A】データ要素の所与の 2 D アレイが非正方形又は正方形のいずれかであり得る例を示す。

【図 3 B】データ要素の所与の 2 D アレイが非正方形又は正方形のいずれかであり得る例を示す。

【図 4 A】例示的な一実装形態による、アレイストレージ内の正方形 2 D アレイへのアクセスがどのように実行され得るかを概略的に示す。

【図 4 B】例示的な一実装形態による、アレイストレージ内の正方形 2 D アレイへのアクセスがどのように実行され得るかを概略的に示す。

【図 5 A】例示的な一実装形態による、移動及びゼロ命令内に提供され得るフィールドを概略的に示す。

【図 5 B】移動及びゼロ命令のベクトル識別フィールドを実現するために 1 つの特定の例示的な実装形態において使用され得るサブフィールドを概略的に示す。

【図 6】例示的な一実装形態による、移動及びゼロ命令がどのように処理され得るかを示す流れ図である。

【図 7】アレイストレージ内に提供されるデータ要素上で動作し得る 1 つの例示的命令シーケンスを示し、命令シーケンスは、本明細書に説明される移動及びゼロ命令のいくつかのインスタンスを含む。

【図 8】実行され得る有限インパルス応答 (FIR) フィルタリング演算を概略的に示す。

【図 9 A】例示的な一実装形態による、2 D 画像フィルタリング演算を実行するときにアレイストレージがどのように使用され得るかを示す図である。

【図 9 B】例示的な一実装形態による、2 D 画像フィルタリング演算を実行するときにアレイストレージがどのように使用され得るかを示す図である。

【図 9 C】例示的な一実装形態による、2 D 画像フィルタリング演算を実行するときにアレイストレージがどのように使用され得るかを示す図である。

【図 9 D】例示的な一実装形態による、2 D 画像フィルタリング演算を実行するときにアレイストレージがどのように使用され得るかを示す図である。

【図 10】アレイストレージ内に提供されるデータ要素上で動作し得る代替的な例示的命令シーケンスを示し、命令シーケンスは、本明細書に説明される移動及びゼロ命令のいくつかのインスタンスを含む。

【図 11】例示的な一実装形態による、ゼロベクトル命令がどのように処理され得るかを示す流れ図である。

【図 12】例示的な一実装形態による、ゼロベクトル命令内に提供され得るフィールドを概略的に示す図である。

【図 13】代替的な例示的な実装形態による、ゼロベクトル命令がどのように処理され得るかを示す流れ図である。

【図 14】使用され得るシミュレータ実装形態を示す。

【発明を実施するための形態】

【0008】

1 つの例示的な構成では、演算を実行するための処理回路と、命令によって指定された演算を実行するように処理回路を制御するために命令をデコードするための命令デコード回路とを有する装置が提供される。データ要素を記憶するためのストレージ要素を含むアレイストレージも提供される。アレイストレージは、演算を実行するときに処理回路にアクセス可能なデータ要素の少なくとも 1 つの 2 次元アレイを記憶するように構成されており、データ要素の各 2 次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは 1 次元である。

【0009】

先に述べたように、アレイストレージの使用は、ある種の演算、例えば累積和演算を実

10

20

30

40

50

行するための非常に効率的な機構を提供することができる。累積和演算は、累積和関数のみを実行することができるが、代替的に、累積和関数に加えて追加の処理を組み込むことができる（純粹に例として、累積和演算は、形式 $A = A + B * C$ の乗算累積和関数を実行することができる）。実現可能な潜在的な効率の利点を最大限に利用するために、データ要素がアレキストレージを使用して実行される計算の対象となくなるときにデータ要素をアレキストレージから移動させるための効率的な機構と、アレキストレージの関連するストレージ要素を解放して後続の計算に関連して使用できるようにするための効率的な機構とを有することが必要な場合がある。

【0010】

例示的な一実装形態によれば、そのようなプロセスの効率の大幅な改善を可能にすることができる移動及びゼロ命令が提供される。特に、例示的な一実装形態において、命令デコーダ回路は、アレキストレージ内のデータ要素の所与の2次元アレキのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令に応答して、1つ以上の識別されたベクトルのデータ要素をアレキストレージから宛先ストレージに移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用されたアレキストレージのストレージ要素を論理ゼロ値に設定するように処理回路を制御するように構成され得る。

【0011】

したがって、上記の技術によれば、実行されると、所与の2次元アレキ内のデータ要素の1つ以上の識別されたベクトル内のデータ要素をアレキストレージから移動させ、更に、それらのデータ要素を記憶していたアレキストレージの関連するストレージ要素を論理ゼロ値にクリアし、これにより、後続の計算で使用するためにそれらを準備する、単一の命令を指定することができる。

【0012】

これにより、性能を大幅に向上させることができる。特に、アレキストレージからデータ要素を移動させる動作、及び再使用のために関連するストレージ要素を準備する動作は、それ自体では有用な計算を実行せず、したがって、アレキストレージの使用に関連するオーバーヘッドと見なすことができる。単一の命令が、データ要素を移動させ、関連するストレージ要素を論理ゼロ値にクリアすることを可能にすることによって、このオーバーヘッドを大幅に低減することができる。

【0013】

特に、従来の既知の技術では、必要なデータ要素をアレキストレージから指定された宛先ストレージに移動するために少なくとも1つの移動命令を実行する必要がある、その後、論理ゼロ値の1つ以上のベクトルを1つ以上のソースベクトルレジスタからアレキストレージの関連するストレージ要素に移動するために1つ以上の追加の移動命令が必要であった。したがって、これは、1つずつ実行されなければならない依存する命令のシーケンスを生成する。例えば、アレキストレージからデータ要素の1つのベクトルを移動するために第1の移動命令が使用され、次に、論理ゼロ値のベクトルを指定されたソースベクトルレジスタからアレキストレージの関連するストレージ要素に移動するために第2の移動命令が使用される単純な場合のみを考慮すると、1つずつ実行される必要がある2つの依存する移動命令が明らかに存在し、この命令依存性は、新しい移動及びゼロ命令を使用することによって除去される。

【0014】

更に、組み合わせられた移動及びゼロ化動作を実行することに関連するハードウェアコストは、いくつかの実装形態では、データ要素のベクトルをアレキストレージから移動させるための標準的な移動動作だけを行うことに関連するハードウェアコストと同じであり得、したがって、関連するストレージ要素のゼロ化は、追加のハードウェアオーバーヘッドなしのために効果的に取得されることが分かっている。更に、組み合わせられた移動及びゼロ化動作の性能（実行速度）は、例示的な一実装形態において、単一の移動動作を単に実行することと同じであることが分かっている。

【0015】

更に、本技術を使用することにより、さもなければそれらの論理ゼロ値をアレイストレージへ移動するために使用される移動命令のためのソースオペランドとして使用される必要があるベクトルレジスタ内に論理ゼロ値を記憶する必要性が回避され、これにより、ベクトルレジスタファイル内の1つ以上のベクトルレジスタを解放する。

【0016】

更に、ゼロ化機能を実行するために別個の移動命令を必要としないので、性能が改善されることが分かる。

【0017】

そのようなアプローチは、アレイストレージの多くの例示的な使用事例において非常に有益であることが分かっている。例えば、アレイストレージは、多くの場合、累積和演算の多数の反復を実行するときに生成される結果を累積するために使用され、最終累積和結果が生成されるとき、それらは、典型的には、例えば、装置内に設けられた1つ以上のベクトルレジスタに移動されることによって、アレイストレージから移動される。累積和演算の実行においてアレイストレージが使用されているとき、それらのストレージ要素が最初に論理ゼロ値に設定される場合にのみ、新しい一連の累積和演算のためにそれらの最終累積和結果を記憶していたストレージ要素を再使用することが可能であり、本明細書で説明されるような移動及びゼロ命令の使用は、これが効率的に達成されることを可能にすることができる。

【0018】

したがって、例示的な実装形態では、処理回路は、累積和演算の複数の反復を実行し、データ要素の所与の2次元アレイを使用して、累積和演算を実行するときに生成される累積和結果を維持するように構成されてもよく、累積和演算の所与の反復の後、データ要素の所与の2次元アレイ内のデータ要素の少なくとも1つの所与のベクトルが、最終累積和結果を記憶するように構成されている一方、データ要素の所与の2次元アレイ内のデータ要素の残りのベクトルが、中間累積和結果を記憶するように構成されている。そのような実装形態では、移動及びゼロ命令は、データ要素の少なくとも1つの所与のベクトルを識別するように構成され得、累積和演算の所与の反復の後に実行されて、処理回路に、少なくとも1つの所与のベクトルの最終累積和結果をアレイストレージから宛先ストレージに移動させ、少なくとも1つの所与のベクトルの最終累積和結果を記憶するために使用されたアレイストレージのストレージ要素を、後続の累積和演算で使用するためにそれらのストレージ要素を解放するためにクリアさせ得る。

【0019】

上述した累積和演算は、単に累積和関数（例えば、形式 $A = A + B$ ）を実行することができるが、より一般的には、累積和関数に加えて何らかの追加の処理演算を含むこともできることに留意されたい。したがって、累積和演算は、処理演算結果値を生成するために実行される処理演算を含むことができ、処理演算結果値は、次いで、アレイストレージの関連するストレージ要素内の既存のデータ要素値と累積されて、アレイストレージのその関連するストレージ要素内に記憶される新しいデータ要素値を作成する。純粹に例として、上述の累積和演算は、乗算累積和演算（例えば、 $A = A + B * C$ の形式の）であってもよい。

【0020】

上述の累積和関数を利用することができ、そのためのアレイストレージの使用が効率的な実装技術を提供することができる様々なタイプのデータ処理動作が存在する。1つの特定の例示的な使用事例では、累積和演算の複数の反復は、入力データ要素のアレイに対して有限インパルス応答（FIR）フィルタリング演算を実行するために使用される処理演算及び累積和演算であり、アレイストレージ内のデータ要素の所与の2次元アレイは、FIRフィルタリング演算の実行中に生成される出力データ要素のアレイを維持するために使用され得る。処理回路は、累積和演算の各反復中に、入力データ要素の単一のベクトルを処理し、出力データ要素のアレイの複数のベクトル内に累積するための出力データ要素を生成するように構成され得る。

【0021】

入力データ要素と出力データ要素との間の対応は、実装形態に依存して変化し得る。例えば、入力データ要素の1つのベクトルは、出力データ要素の複数のベクトルに関連付けられてもよい。加えて、出力データ要素のこれらの複数のベクトルは、アレイストレージ内で水平方向及び垂直方向のいずれか又は両方に配置され得る（ベクトルが水平方向及び垂直方向の両方でアクセスされ得る実装形態をサポートするために、データ要素の2次元アレイは、典型的には、データ要素の2次元正方形アレイである）。更に、入力データ要素及び出力データ要素のサイズは異なってもよい。

【0022】

上述したような技術は、データ要素の正方形のアレイを使用して、スライディングウィンドウ技術によって実現されるFIRフィルタリングを計算するために外積アプローチを利用することができる。そのような技術は、典型的には、出力データ要素の正方形アレイのベクトルのうちのいくつかは、出力データ要素の他のベクトルの前に確定されることをもたらし、したがって、上述の移動及びゼロ命令の使用は、出力データ要素のそれらの確定されたベクトルがアレイストレージから移動されることを可能にすることができ、関連するストレージ要素は、出力データ要素の他のベクトルと関連して使用するために解放される。

10

【0023】

入力データ要素のアレイは、様々な形態をとることができるが、例示的な一実装形態では、画素値のアレイを表すことができる。しかしながら、本明細書で説明される技術は、例えば画像データを表さない場合があるデータ要素の他のアレイにも等しく適用可能である。

20

【0024】

いくつかの例示的な実装形態では、入力データ要素の行を処理することによって生成される出力データ要素の複数のベクトルは、出力データ要素の複数の「行」と呼ばれ得る。しかしながら、先に述べたように、出力データ要素のデータ要素の所与の正方形2Dアレイ（そのような正方形2Dアレイは、本明細書では正方形サブアレイとも呼ばれ得る）内に累積された出力データ要素の行は、アレイストレージ内で任意の所望の向きで記憶され得ることに留意されたい。例えば、行は、正方形サブアレイ内の水平ベクトルとして、又は正方形サブアレイ内の垂直ベクトルとして記憶されてもよく、したがって、「行」という用語は、本明細書では、アレイストレージ内のデータ要素の任意の特定の配向を暗示するものとして解釈されるべきではない。

30

【0025】

1つの例示的な構成では、データ要素の所与の2次元アレイは、データ要素の正方形2次元アレイであり、データ要素の正方形2次元アレイを形成する複数のベクトルは、第1のアレイ方向に配置された第1の複数のベクトルと、第1のアレイ方向に直交する第2のアレイ方向に配置された第2の複数のベクトルとを含み、移動及びゼロ命令の各インスタンスは、すべてが第1のアレイ方向に延びるか、又はすべてが第2のアレイ方向に延びるデータ要素の1つ以上のベクトルを識別するように構成されている。したがって、これは、アレイストレージから移動されるべきデータ要素の様々なベクトルがどのように識別されるかに関して大きな柔軟性を提供する。

40

【0026】

アレイストレージ内に記憶されるデータ要素の1つ以上の2次元アレイは、実装形態に応じて様々な形態をとることができる。1つの特定の例示的な実装形態では、処理回路は、データ要素の正方形の2次元アレイに対して処理動作を実行するように構成されており、その間、処理回路は、第1のアレイ方向及び第2のアレイ方向の両方においてデータ要素のベクトルにアクセスすることを可能にされる。

【0027】

例示的な一実装形態では、アレイストレージは、第1のアレイ方向に延びる複数のアレイベクトルレジスタを備えるように構成することができる。処理回路は、1つ以上の累積

50

和演算を実行するように構成されてもよく、各累積和演算は、アレイストレージの複数のアレイベクトルレジスタのグループ内で累積するための出力データを生成するように構成されている。したがって、そのような実装形態では、アレイストレージは、単一方向内に延びる複数の別々にアドレス指定可能なアレイベクトルレジスタを備えるものと見なされ、データ要素の前述の所与の2次元アレイは、上記で参照した複数のアレイベクトルレジスタのグループ内に記憶されたデータ要素を備えるものと見なされ得る。

【0028】

そのような例示的な実装形態では、1つ以上の累積和演算の実行の結果、移動及びゼロ命令によって識別された1つ以上の識別されたベクトル内に最終結果データが存在するようになると、処理回路に、1つ以上の識別されたベクトルのデータ要素をアレイストレージから宛先ストレージに移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用された複数のアレイベクトルレジスタのグループ内の各アレイベクトルレジスタを論理ゼロ値に設定するために、移動及びゼロ命令が実行され得る。

【0029】

1つの特定の例示的な実装形態では、処理回路が1つ以上の累積和演算の実行を完了したとき、最終結果データは、複数のアレイベクトルレジスタのグループ内の各アレイベクトルレジスタ内に存在する。移動及びゼロ命令の実行は、次いで、最終結果データを複数のアレイベクトルレジスタのグループから宛先ストレージに移動させ、複数のアレイベクトルレジスタのグループ内の各アレイベクトルレジスタに対してゼロにクリアさせることができる。次いで、これは、処理回路が、任意の所望の後続処理動作のためにアレイベクトルレジスタのグループからの1つ以上のアレイベクトルレジスタを再使用することを可能にする（したがって、例えば、それらのアレイベクトルレジスタのいずれか又はすべてを使用する後続累積和命令の実行は、アレイベクトルレジスタの内容がゼロにクリアされることに起因して、非累積変形形態が実行されることになる）。

【0030】

移動及びゼロ命令のために指定される宛先ストレージは、様々な形態をとることができる。例示的な一実装形態では、装置は、複数のベクトルレジスタを備えるベクトルレジスタファイルに更に提供することができ、移動及びゼロ命令は、宛先ストレージとして、ベクトルレジスタファイル内の1つ以上のベクトルレジスタを示すように構成され得る。1つ以上のベクトルレジスタを識別するために移動及びゼロ命令を構成することができる様々な方法がある。例えば、データ要素の単一ベクトルがアレイストレージから移動される単一ベクトルレジスタの場合、移動及びゼロ命令は、その単一ベクトルレジスタを決定するために使用される識別子を提供することができる。データ要素の複数のベクトルがアレイストレージから移動される複数のベクトルレジスタの場合、それらの複数のベクトルレジスタは、ベクトルレジスタの各々について別個の識別子情報を使用して明示的に識別され得るか、又は代替的に、1つのベクトルレジスタが命令によって識別され得、複数のベクトルレジスタ内の他のベクトルレジスタは暗黙である。例えば、複数のベクトルレジスタは、明示的に識別されたベクトルレジスタから始まる隣接するベクトルレジスタのシーケンスであり得るか、又は複数のベクトルレジスタは各々、一定のストライド値によって分離され得る。

【0031】

しかしながら、代替実装形態では、所望であれば、移動及びゼロ命令は、1つ以上の識別されたベクトルのデータ要素が記憶されるべきメモリ内の1つ以上の位置を宛先ストレージとして示すように構成され得る。そのような場合、移動及びゼロ命令は、代わりに記憶及びゼロ命令と呼ばれることがある。

【0032】

移動及びゼロ命令が、データ要素の1つ以上のベクトルが移動されるべきメモリ位置を識別するように構成され得る様々な方法がある。例えば、データ要素の単一ベクトルのみがメモリに移動される単一ベクトルの場合を考慮すると、移動及びゼロ命令は、メモリ内の位置を識別するように構成されてもよく、データ要素のベクトルは、次いで、その位置

によって識別される連続メモリアドレスに書き込まれる（この場合、位置は、例えば、第1のデータ要素のメモリアドレスであってもよい）。複数のベクトルが移動される場合、メモリ内の複数の別個の位置が命令によって識別されてもよく、データ要素のベクトルの各々は、指定された位置のうちの1つによって識別されるメモリアドレスのシーケンスに移動される。あるいは、1つの位置を命令によって指定することができ、他の位置は暗黙的であってもよい（例えば、これらの位置は、識別された位置から固定ストライド／オフセットでメモリアドレスを識別する）。

【0033】

アクセスされるアレイ方向及びアレイ内のそのアクセスされたベクトルに保持されるデータ要素の性質に応じて、単一のベクトルの個々の要素が別個のメモリ位置に記憶される場合もあり得る。しかしながら、単一のベクトル内の個々のデータ要素が別個のメモリ位置に記憶される必要がある状況では、通常、データ要素のベクトルは、やがてメモリに転送される前に、まずベクトルレジスタに移動される。

【0034】

移動されるべきデータ要素の1つ以上のベクトルが、移動及びゼロ命令によって識別され得るいくつかの方法がある。例示的な一実装形態では、移動及びゼロ命令は、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別するために使用されるベクトル識別フィールドを備え得る。例えば、データ要素の単一ベクトルが移動される場合、その単一ベクトルを識別するのに十分な識別子が提供され得る。データ要素の複数のベクトルが移動されるとき、ベクトル識別フィールドは、それらの複数のベクトルのそれぞれを明示的に識別するのに十分な情報を提供するために使用されてもよく、又は代替的に、データ要素の1つのベクトルが明示的に識別されてもよく、その場合、他のベクトル、例えば、隣接するベクトル、又は規則的に離間されたベクトル（ストライドアクセスと呼ばれることが多い）が暗黙的である。この後者の場合、移動されるべきベクトルの数を識別するために、ベクトル識別フィールドによって数が提供され得る。

【0035】

アクセスがいずれかのアレイ方向であり得る前述のアプローチでは、アクセスされているアレイ方向を識別するのに十分な情報を提供するために、ベクトル識別フィールドを使用することもできる。例えば、一実装形態では、ベクトル識別フィールドは、正方形2次元アレイを識別するための第1のサブフィールドと、1つ以上のベクトルを識別するために使用される1つ以上のライン識別子及びアレイ方向インジケーションを提供する第2のサブフィールドとを備え得る。

【0036】

例示的な一実装形態では、移動及びゼロ命令は、1つ以上の識別されたベクトルのどのデータ要素がアレイストレージから宛先ストレージに移動されるべきかを識別し、それらの関連するストレージ要素を論理ゼロ値に設定させるために使用される述語情報を識別するための述語フィールドを含み得る。これは、機能性が特定のベクトル内の特定のデータ要素に限定されることを可能にすることによって、更なる柔軟性を提供することができる。

【0037】

述語が使用されるいくつかの実装形態では、データ要素サイズは変更することができる場合があり、そのような場合、移動及びゼロ命令は、1つ以上の識別されたベクトル内の各データ要素のサイズを識別するためのサイズフィールドを備え得る。命令がこの追加情報を提供することを可能にすることによって、1つ以上の識別されたベクトル内の全データ要素のサブセットに関して移動及びゼロ動作が実行されることを可能にしながら、システム内で処理されている様々な異なるデータ要素サイズに対して命令が使用されることが可能である。

【0038】

本明細書で説明される別の技術によれば、アレイストレージ内のデータ要素のベクトル

10

20

30

40

50

をゼロにしかつそのようなアレイストレージを使用して累積和演算を実行するときに性能の改善を提供するために使用することもできる追加の新しい形態の命令が提供される。この技術によれば、演算を実行するための処理回路と、命令によって指定された演算を実行するように処理回路を制御するために命令をデコードするための命令デコーダ回路と、データ要素を記憶するためのストレージ要素を備えるアレイストレージとを有する装置が提供される。前に説明した技術のように、アレイストレージは、演算を実行するときに処理回路にアクセス可能なデータ要素の少なくとも1つの2次元アレイを記憶するように構成されており、データ要素の各2次元アレイは、データ要素の複数のベクトルを含み、各ベクトルは1次元である。この追加の技術によれば、命令デコーダ回路は、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の複数のベクトルを識別するゼロベクトル命令をデコードすることに対応して、データ要素の識別された複数のベクトルに対して演算するように構成された後続の累積和命令もデコードし、累積和命令によって指定された累積和演算の非累積変形形態を実行してアレイストレージ内の識別された複数のベクトルに記憶するための結果データ要素を生成するように処理回路を制御するように構成されている。

10

【0039】

上記のアプローチを使用することによって、ゼロベクトル命令は、命令デコーダ回路によるデコード時に、ゼロベクトル命令によって指定されるのと同じデータ要素の複数のベクトルを指定する後続の累積和命令と融合／マージされて、事実上、その累積和命令の非累積変形形態を作成することができる。命令符号化空間は典型的にはかなり制約されるので、そのようなアプローチは非常に有益であり得るが、アレイストレージ内の複数のベクトルに関して累積和演算を実行するためにいくつかの異なる累積和命令を提供することが望ましい場合があるが、それらの命令の非累積変形形態を提供することによって消費される命令符号化空間の量により、それらの非累積変形形態も提供しようとすることは非常に望ましくない場合があることが分かっている。本技術を使用することによって、命令の非累積変形形態を提供する必要がなく、代わりに、処理回路に非累積変形形態を実行させるために、ゼロベクトル命令を後続の累積和命令と組み合わせることによって、それらをシミュレートすることができる。

20

【0040】

例示的な一実装形態では、アレイストレージは、第1のアレイ方向に延びる複数のアレイベクトルレジスタを備えることができ、アレイストレージ内の識別された複数のベクトルは、アレイストレージの複数のアレイベクトルレジスタのグループによって提供される。そのような構成では、データ要素の所与の2次元アレイは、複数のアレイベクトルレジスタのグループ内に記憶されたデータ要素を備え得る。更に、後続の累積和命令は、データ要素の識別された複数のベクトルに対して（すなわち、ゼロベクトル命令によって指定される複数のアレイベクトルの同じグループに対して）実行されるべき累積和演算を含む処理演算を指定し得、ゼロベクトル命令は、処理演算の非累積変形形態の処理回路による実行を可能にするために、後続の累積和命令と組み合わせて使用され得る。

30

【0041】

例示的な一実装形態では、ゼロベクトル命令は、アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の複数のベクトルを識別するために使用されるベクトル識別フィールドを備え得る。

40

【0042】

先に説明した移動及びゼロ命令と同様に、ゼロベクトル命令は、必要に応じて、複数の識別されたベクトル内のどのストレージ要素が論理ゼロ値に設定されるべきかを識別するために使用される述語情報を識別する述語フィールドを含むことができる。そのようなアプローチは、実際には、いくつかのデータ要素が後続の累積和命令の非累積変形形態を受け、他のデータ要素が真の累積変形形態を受けることを可能にすることができる。また、必要に応じて、ゼロベクトル命令は、複数の識別されたベクトル内の各データ要素のサイズを識別するためのサイズフィールドを含んでもよい。

50

【0043】

次に、添付の図面を参照して、特定の例示的な実装形態を説明する。

【0044】

図1は、データ値32及びプログラム命令34を記憶するメモリ30に結合されたプロセッサ20を備える、データ処理システム10を概略的に示す。プロセッサ20は、メモリ30からプログラム命令34をフェッチし、フェッチされたプログラム命令を命令デコーダ回路50に供給するための命令フェッチユニット40を含む。デコーダ回路50は、フェッチされたプログラム命令をデコードし、処理回路60を制御するための制御信号を生成して、デコードされたベクトル命令によって指定されるようにレジスタストレージ65のストレージ要素内に保持されたデータ値について処理演算を実行する。図1に示すように、レジスタストレージ65は、複数の異なるブロックから形成されてもよい。例えば、命令によって指定することができる複数のスカラレジスタを含むスカラレジスタファイル70が提供されてもよく、同様に、命令によって指定することができる複数のベクトルレジスタを含むベクトルレジスタファイル80が提供されてもよい。

10

【0045】

また、図1に示すように、プロセッサ20は、アレイストレージ90にアクセスすることができる。図1に示される例では、アレイストレージ90はプロセッサ20の一部として提供されるが、これは必要条件ではない。様々な例では、アレイストレージは、以下のうちのいずれか1つ以上として実装することができる。すなわち、アーキテクチャ的にアドレス指定可能なレジスタ、アーキテクチャ的にアドレス指定不可能なレジスタ、スクラッチパッドメモリ、及びキャッシュである。

20

【0046】

処理回路60は、例示的な一実装形態では、ベクトル処理回路及びスカラ処理回路の両方を含んでよい。スカラ処理とベクトル処理との間の一般的な区別は、以下の通りである。ベクトル処理は、単一ベクトル処理命令を、データベクトル内のそれぞれの位置に複数のデータ要素を有するデータベクトルのデータ要素に適用することを含む。本技術によれば、処理回路は、アレイストレージ90内に記憶されたデータ要素の2次元アレイ（サブアレイとも呼ばれることがある）内の複数のベクトルに対して演算を実行するためにベクトル処理を実行してもよい。スカラ処理は、データベクトルではなく、単一のデータ要素に対して効果的に演算を行う。ベクトル処理は、処理されるデータの多くの異なるインスタンスに処理演算が実行される場合に有用であり得る。ベクトル処理構成では、単一の命令を（データベクトルの）複数のデータ要素に同時に適用することができる。これにより、スカラ処理と比較して、データ処理の効率及びスループットを改善することができる。

30

【0047】

プロセッサ20は、アレイストレージ90に記憶されたデータ要素の2次元アレイを処理するように構成することができる。2次元アレイは、少なくともいくつかの実施例では、複数の方向においてデータ要素の1次元ベクトルとしてアクセスされてよい。例示的な一実装形態では、アレイストレージ90は、データ要素の1つ以上の2次元アレイを記憶するように構成されてもよく、データ要素の各2次元アレイは、メモリ内のデータ要素のより大きい又は更に高次元のアレイの正方形アレイ部分を形成してもよい。

40

【0048】

図2は、例示的な一実装形態において提供され得るプロセッサ20のアーキテクチャレジスタ65の一例を示す。（命令セットアーキテクチャ（ISA）において定義されるような）アーキテクチャレジスタは、処理回路60内のスカラ処理回路によって実行される処理動作のための汎用レジスタとして働くスカラ整数レジスタ100のセットを含み得る。例えば、一定の数の汎用レジスタ100、例えば、この例では31個のレジスタX0～X30が提供されてよい（スカラレジスタフィールドの32番目の符号化は、ハードウェアに提供されるレジスタに対応しない場合がある。なぜならば、これは、例えば、デフォルトで0の値を示すと見なされ得るか、又は汎用レジスタではなく専用タイプのレジスタであることを示すために使用することができるからである）。同じ物理ストレージにマッ

50

ピングされた異なるサイズのスカラレジスタにアクセスすることが可能であり得る。例えば、レジスタラベルX0～X30は、64ビットレジスタを指してもよいが、同じレジスタが32ビットレジスタとしてアクセスされてもよく（例えば、ハードウェアで提供される各64ビットレジスタの下位32ビットを使用してアクセスされる）、その場合、レジスタラベルW0～W30は、同じレジスタを参照するためにアセンブラコードで使用されてもよい。

【0049】

また、デコーダ50によってサポートされるISA内のプログラム命令による選択のために利用可能なアーキテクチャレジスタは、一定の数のベクトルレジスタ105（この例ではZ0～Z31とラベル付けされている）を含むことができる。もちろん、図2に示すスカラ／ベクトルレジスタの数を提供することは必須ではなく、他の例は、プログラム命令によって指定可能な異なる数のレジスタを提供してもよい。各ベクトルレジスタは、各データ要素が独立したデータ値を表し得る可変の数のデータ要素を含むベクトルオペランドを記憶し得る。ベクトル処理（SIMD）命令に応じて、処理回路は、レジスタに記憶されたベクトルオペランドに対してベクトル処理を実行して、結果を生成し得る。例えば、ベクトル処理は、結果ベクトルの要素の対応する結果を生成するために、1つ以上のオペランドベクトル内の要素の各レーンに対して対応する演算が実行される、レーンごとの演算を含み得る。ベクトル又はSIMD処理を実行するとき、各ベクトルレジスタは、あるベクトル長VLを有し得、ここで、ベクトル長は、所与のベクトルレジスタ内のビット数を指す。ベクトル処理モードで使用されるベクトル長VLは、所与のハードウェア実装形態に対して固定されてもよく、又は可変であってもよい。プロセッサ20によってサポートされるISAは、異なるプロセッサ実装形態が異なるサイズのベクトルレジスタを実装することを選択し得るように、可変ベクトル長をサポートし得るが、ISAは、そのプログラムを実行する所与のCPU上で実装されている特定のベクトル長にかかわらずコードが正しく機能し得るように命令が設計されるように、ベクトル長アグノスティックであり得る。

【0050】

ベクトルレジスタZ0～Z31は、アレイストレージ90内に記憶されたデータ要素の2次元アレイに対して処理回路60によって実行される処理演算及び累積和演算への入力を提供するベクトルオペランドを記憶するためのオペランドレジスタとしても機能し得る。ベクトルレジスタがそのような演算への入力を提供するために使用されるとき、ベクトルレジスタは、ベクトル演算に使用されるベクトル長VLと同じであり得るか、又は異なるベクトル長であることができるベクトル長MVLを有する。

【0051】

図2に示すように、アーキテクチャレジスタは、前述のアレイストレージ90を形成する一定の数NAのアレイレジスタ110、ZA0～ZA（NA－1）も含む。各アレイレジスタは、データ要素の単一の2Dアレイ、例えば、処理及び累積和演算の結果を記憶するためのレジスタストレージのセットと見なすことができる。しかしながら、処理演算及び累積和演算は、アレイレジスタを使用することができる唯一の演算ではない場合がある。アレイレジスタは、メモリ内のアレイ構造の行／列方向の転置を実行する際に、正方形アレイを記憶するために使用することもできる。プログラム命令がアレイレジスタ110のうちの1つを参照するとき、それは、アレイ識別子ZAiを使用して単一のエンティティとして参照されるが、いくつかのタイプの命令（例えば、データ転送命令）は、アレイの一部（例えば、要素の1つの水平／垂直グループ）を選択するインデックス値を定義することによってアレイのサブ部分を選択することもできる。

【0052】

実際には、アレイレジスタに対応するレジスタストレージの物理的実装形態は、図2にも示されているように、一定の数NRのアレイベクトルレジスタZAR0～ZAR（NR－1）を含むことができる。アレイレジスタストレージ110を形成するアレイベクトルレジスタZARは、SIMD処理やアレイ処理へのベクトル入力に使用されるベクトルレ

レジスタ $Z_0 \sim Z_{31}$ とは別個のレジスタセットであり得る。アレイベクトルレジスタ ZAR の各々は、ベクトル長 MVL を有し得、したがって、各アレイベクトルレジスタ ZAR は、可変の数のデータ要素に論理的に分割され得る長さ MVL の 1D ベクトルを記憶し得る。例えば、 MVL が 512 ビットである場合、これは、例えば、64 個の 8 ビット要素、32 個の 16 ビット要素、16 個の 32 ビット要素、8 個の 64 ビット要素、又は 4 個の 128 ビット要素のセットであり得る。これらのオプションのすべてが所与の実装形態においてサポートされる必要があるわけではないことが理解されるであろう。可変要素サイズをサポートすることによって、これは、異なる精度のデータ構造を伴う計算処理の柔軟性を提供する。データの 2D アレイを表すために、アレイベクトルレジスタ $ZAR_0 \sim ZAR(N_R - 1)$ のグループは、アレイレジスタ識別子 $ZA_0 \sim ZA(N_A - 1)$ のうちの所与の 1 つを割り当てられた単一のエンティティとして論理的に考えることができ、したがって、2D アレイは、アレイの 1 次元に対応する単一のベクトルレジスタ内に展開する要素と、複数のベクトルレジスタにわたってストライピングされるアレイの他の次元の要素とで形成される。

10

【0053】

水平方向における要素の数が垂直方向における要素の数と等しいデータの正方形アレイを記憶するようにアレイレジスタ ZA を構成することは、必須ではないが有用であり得る。これは、水平方向又は垂直方向のいずれかにおいてアレイレジスタ 110 を読み出す／書き込むためのサポートを提供することによって、アレイ構造をアレイレジスタ 110 とメモリとの間で転送する際にメモリにおけるアレイ構造の行／列次元を切り替えることができるアレイのオンザフライ転置をサポートするのに役立ち得る。水平方向又は垂直方向のいずれかで 2D アレイレジスタからデータを書き込む／読み出すためのサポートを提供することによって、これは、メモリとベクトルレジスタとの間でデータを転送するためのいくつかのギャザ／スキッターロード／ストア又は置換動作を用いて可能であるよりも高速に、一方向（例えば、行毎）にメモリからロードされたデータが、反対方向（例えば、列毎）にメモリに書き戻されることを可能にすることができる。

20

【0054】

上述したように、データ要素の所与の 2D アレイは、いくつかの例示的な実装形態では正方形アレイを形成し得るが、これは必要条件ではない。したがって、図 3A に示すように、例示的な一実装形態では、データ要素 115 の所与の 2D アレイは、非正方形アレイを形成することができる。代替として、図 3B に示されるように、データ要素 120 の所与の 2D アレイは、正方形アレイを形成してもよい。図 3A 及び図 3B の各々において、個々の正方形のボックスはデータ要素を示し、いくつかの実装形態では、データ要素サイズは変化し得る。いずれの例においても、データ要素の 2 次元アレイは、様々な方法で指定することができるが、例示的な一実装形態では、図 3A 及び図 3B に示すように、データ要素の所与の 2D アレイは、アレイベクトルレジスタ (ZAR) のシーケンスによって指定することができる。

30

【0055】

上述のように、処理回路 60 は、デコーダ回路 50 によってデコードされた命令の制御下で、スカラーレジスタ 70、ベクトルレジスタ 80 及び／又はアレイストレージ 90 にアクセスするように構成されている。ここで、この後者の構成の更なる詳細を図 4A を参照して説明するが、図 4A は、特にアレイストレージ内の正方形 2D アレイへのアクセスを考慮して、アレイストレージがどのようにアクセスされ得るかについての 1 つの例示的な例を単に提供する。

40

【0056】

示された実施例では、アレイストレージ 90 内の正方形の 2D アレイは、 $n \times n$ 個のストレージ要素／位置 200 のアレイ 205 として構成されており、 n は 1 よりも大きい整数である。本実施例では、 n は 16 であり、これは、ストレージ位置 200 へのアクセスの粒度が、水平アレイ方向又は垂直アレイ方向のいずれかで全ストレージの $1/16$ であることを意味する。

50

【0057】

処理回路の視点から、 $n \times n$ 位置のアレイは、第1の方向（例えば、描かれた場合の水平方向）の n 個の線形（1次元）ベクトル及び第2のアレイ方向（例えば、描かれた場合の垂直方向）の n 個の線形ベクトルとしてアクセス可能である。したがって、 $n \times n$ のストレージ位置は、処理回路60の視点から、それぞれ n のデータ要素からなる $2n$ 線形ベクトルとして構成されているか、又は少なくともアクセス可能である。

【0058】

ストレージ位置200のアレイは、少なくとも処理回路60と通信する及び任意選択的にデコード回路50と通信する制御回路250の制御下で、アクセス回路210、220、列選択回路230及び行選択回路240によってアクセス可能である。

10

【0059】

図4Bを参照すると、第1の方向（描かれている場合の水平又は「H」方向）における n 個の線形ベクトルは、「A1」として設計された例示的な正方形の2Dアレイの場合（以下で説明するように、アレイストレージ90内に設けられた2つ以上のこのような2Dアレイ、例えば、A0、A1、A2などが存在し得ることに留意されたい）、16個のデータ要素0～F（16進法において）のそれぞれであり、この実施例ではA1H0～A1H15として参照され得る。図4Bのアレイストレージ90 A1の256エントリ（ 16×16 エントリ）に記憶された同じ基礎データは、代わりに、A1V0～A1V15として、第2の方向（描かれている場合の垂直又は「V」方向）で参照されてもよい。例えば、データ要素260は、A1H0の項目Fであるが、A1V15の項目0として参照されることに留意されたい。なお、「H」及び「V」の使用は、アレイストレージ90を構成するデータ要素の記憶に関する任意の空間的又は物理的レイアウト要件を暗示せず、また、あらゆる例示的な適用においてアレイストレージ内の2Dアレイが行データ又は列データを記憶するかどうかに関係しない、ということに留意されたい。

20

【0060】

先に論じたように、アレイストレージ90の使用は、ある種の演算、例えば、アレイストレージ90内のデータ要素の所与の2次元アレイに対してそのような累積和演算の複数の反復が実行され得る累積和演算に関して性能を大幅に改善することができ、データ要素の2次元アレイは、それらの累積和演算を実行するときに結果を累積するために使用される。しかし、これらの累積和演算が完了すると、データ要素の結果として得られるベクトルをアレイストレージから移動させ、後続の累積和演算に使用できるようにアレイストレージ内の関連するストレージ要素を準備するための効率的な機構が望ましい。

30

【0061】

先に論じたように、例示的な一実装形態では、これは、アレイストレージ90内のデータ要素の所与の2次元アレイのデータ要素の1つ以上のベクトルを識別する移動及びゼロ命令を使用することによって達成される。そのような移動及びゼロ命令がデコードされると、処理回路60は、1つ以上の識別されたベクトルのデータ要素をアレイストレージから宛先ストレージ（例えば、ベクトルレジスタファイル80内の1つ以上のベクトルレジスタであることができる）に移動させ、1つ以上の識別されたベクトルのデータ要素を記憶するために使用されたアレイストレージのストレージ要素を論理ゼロ値に設定するように制御される。

40

【0062】

図5Aは、例示的な一実装形態による、移動及びゼロ命令内に提供され得るフィールドを概略的に示す。オペコードフィールド305は、命令を移動及びゼロ命令として識別するために使用される。いくつかの例示的な実装形態では、提供される移動及びゼロ命令の異なる変形形態があり得、したがって、移動及びゼロ命令を識別し得る2つ以上の異なるオペコードがあり得る。1つの特定の例として、アレイストレージ内の識別されたベクトルをベクトルレジスタファイル80内のターゲットベクトルレジスタに移動させるために使用されるときに移動及びゼロ命令のために定義される1つの変形形態が存在してもよく、アレイストレージ内の1つ以上のベクトルがメモリに移動されるときには、異なる変形

50

形態が提供されてもよい（この後者の場合、命令は、例えば、記憶及びゼロ命令と呼ばれてもよい）。

【0063】

ベクトル識別フィールド310も、移動動作を受けるアレイストレージ内の1つ以上のベクトルを識別するために提供される。いくつかの例では、単一のベクトルのみが識別され得るが、他の例では、複数のベクトルがこのフィールドによって識別され得る。後者の場合、それらの複数のベクトルは、例示的な一実装形態ではすべて独立して識別され得るが、別の例示的な実装形態では、複数のベクトルは、例えば、第1のベクトルのインジケーション及び移動されるべきベクトルの数のインジケーションから推論され得る。

【0064】

図5Aに示すように、宛先ストレージ識別フィールド315も、ベクトルが移動されるべき宛先ストレージを識別するために移動及びゼロ命令内に提供される。例示的な一実装形態では、このフィールドは、ベクトルレジスタファイル80内の1つ以上のベクトルレジスタを識別するために使用され、複数のそのようなベクトルレジスタが識別される例では、それらは、複数のベクトルがベクトル識別フィールドによって識別される方法と類似の方法で識別され得る（例えば、第1のベクトルレジスタが識別され得、他のベクトルレジスタは、宛先ストレージを形成するために必要とされるベクトルレジスタの数の知識に基づいて暗黙的である）。アレイストレージ内の1つ以上のベクトルがメモリに移動される代替実装形態では、宛先ストレージ識別フィールド315内に提供される情報は、1つ以上のベクトルのデータ要素が記憶されるべきメモリ内の位置を識別するように構成され得る。これは、例えば、メモリ内の必要な位置を識別するためにその内容が使用される1つ以上のレジスタを識別することを含むことができる。

【0065】

必要に応じて、1つ以上の任意選択の追加フィールド320を命令300内に設けることができる。例えば、述語フィールドを使用して、1つ以上の識別されたベクトル内のどのデータ要素が移動及びゼロ動作を受けるべきかを制御するために使用される述語情報を識別することができる。これは、特定のデータ要素に関して動作が適用されることを可能にするが、他のデータ要素に関しては適用されないことによって柔軟性を提供する。別の例として、データ要素サイズインジケーションが命令内に提供されてもよく、それによって、そのデータ要素サイズが固定されていないベクトルに関して命令が適用されることを可能にする。

【0066】

例示的な一実装形態では、移動及びゼロ命令は、第1のアレイ方向に延びるアレイベクトルレジスタに対して動作するように構成され得、それらの実装形態では、ベクトル識別フィールド310内で水平方向と垂直方向の両方が符号化されることを可能にする必要はない。しかしながら、アレイストレージ90内の2Dアレイが水平方向又は垂直方向のいずれかでアクセスされ得る例では（1つのそのような実装形態では、2Dアレイは正方形アレイである）、ベクトル識別フィールドは、図5Bに示す形態をとり得る。特に、このベクトル識別フィールド310'は、アクセスされるアレイストレージ90内の所与の正方形2Dアレイを識別するために使用される第1のサブフィールド312と、2つの部分313及び314から形成される第2のサブフィールドとを含むことができる。第1の部分313は、正方形2Dアレイ内のデータ要素の1つ以上のラインを識別するための1つ以上のライン識別子を提供し、第2の部分314は、アレイ方向インジケーションを提供し、したがって、ライン識別子によって識別されるデータ要素のラインが水平方向又は垂直方向に延びるかどうかに関する決定を可能にする。第1の部分313と第2の部分314との組み合わせは、正方形2Dアレイ内の1つ以上のベクトルが識別されることを可能にすることが理解されるであろう。

【0067】

図6は、移動及びゼロ命令の動作を示す流れ図である。ステップ350において、デコーダ回路50によって移動及びゼロ命令が遭遇したと判定された場合、ステップ355に

10

20

30

40

50

において、移動及びゼロ命令のベクトル識別フィールド 310 に提供された情報から、アレイストレージ内のデータ要素の 1 つ以上のベクトルが識別される。更に、ステップ 360 において、移動及びゼロ命令の宛先ストレージ識別フィールド 315 から、使用される宛先ストレージが識別される。先に述べたように、このステップは、通常、ベクトルレジスタファイル 80 内の 1 つ以上のベクトルレジスタを、アレイストレージから移動されたデータ要素の宛先として識別させることができるが、代替的に、いくつかの実装形態では、識別された宛先ストレージがメモリ内の 1 つ以上の位置の形態をとる場合もある。例示的な一実装形態では、ステップ 355 及び 360 はデコーダ回路 50 によって実行され得るが、代替的な実装形態では、処理回路 60 は、デコーダ回路 50 によって提供される情報に基づいてこれらの決定ステップを実行し得る。

10

【0068】

ステップ 365 において、処理回路 60 は、データ要素の各識別されたベクトルを宛先ストレージに移動させ、次いで、アレイストレージの関連するストレージ要素、すなわち、この時点で宛先ストレージに移動されているデータ要素を記憶するために使用されたストレージ要素をゼロに設定するために使用される。

【0069】

図 7 は、例示的な一実装形態においてアレイストレージ 90 上で動作し得る例示的な命令シーケンスを示す。図 7 の例に示されるように、一連のデータ処理命令（この例では 3 つの命令）は、アレイストレージ内に提供された所与の 2D アレイ内で処理演算及び累積和演算を実行するために実行され得る。これらの複数の命令の実行中に、結果は所与の 2D アレイ内に累積される。

20

【0070】

この例では、第 3 のデータ処理命令が完了すると、所与の 2D アレイ内の第 1 の垂直ベクトルが最終累積和結果を記憶するのに対して、この段階における所与の 2D アレイ内の他の垂直ベクトルは中間累積和結果のみを記憶すると仮定する。第 1 の垂直ベクトルが最終累積和結果を記憶すると仮定すると、後続の処理演算及び累積和演算で使用するためにその第 1 の垂直ベクトルのストレージ要素を解放するように、それらの結果をアレイストレージから移動させることが有用であろう。

【0071】

図 7 に示されるように、これは、移動及びゼロ命令を実行し、垂直ベクトル 1 を識別し、その垂直ベクトルの内容が移動されるべき宛先ベクトルレジスタを定義することによって達成され、この例では、そのベクトルレジスタはレジスタ Z_i と呼ばれる。この命令の実行により、垂直ベクトル 1 内の最終累積和結果が、識別されたベクトルレジスタ内に移動され、所与の 2D アレイ内の関連するストレージ要素（すなわち、垂直ベクトル 1 を実現するストレージ要素）が論理ゼロ値にクリアされる。この単一命令の実行の結果として、最終累積和結果がアレイストレージから移動されるだけでなく、基礎となるストレージ要素も準備されて、後続の処理演算及び累積和演算における再使用のために直ちに利用可能になる。特に、それらの内容をゼロにクリアすることによって、それらは、後続の命令によって生成される新しい累積和結果の宛先として直ちに指定され始めることができる。

30

【0072】

したがって、図 7 に示すように、後続のデータ処理命令 4 が実行されるとき、これは 2D アレイに累積することができ、必要に応じて、垂直ベクトル 1 を再使用することができる。そのデータ処理命令が実行されると、垂直ベクトル 2 はここで最終累積和結果を保持すると仮定され、したがって、所与の 2D アレイ内の垂直ベクトル 2 の内容をアレイストレージから宛先ベクトルレジスタ（この例ではレジスタ Z_{i+x} ）へ移動するために、追加の移動及びゼロ命令を実行することができる。この場合も、この命令の実行により、識別されたベクトルの内容がアレイストレージから移動させられ、対応するストレージ要素が論理ゼロ値にクリアされ、したがって、後続の処理演算及び累積和演算で使用するためにこれらのストレージ要素を解放する。したがって、図 7 に示すように、データ処理命令の後続の反復を実行することができ、必要に応じて、垂直列 2 を再使用することができる

40

50

。

【0073】

結果を累積するためにアレistroレージ90内の所与の2Dアレイを使用して実行され得る様々なタイプの演算があり、その所与の2Dアレイ内のベクトルのすべてが必ずしも同時に最終累積和結果を保持するわけではない。そのような状況では、再使用のために2Dアレイ内のリソースを解放するように、図7に例として概略的に示されたアプローチを採用することが有用であり得る。そのようなアプローチの1つの例示的な使用事例は、スライディングウィンドウアプローチを使用して2D有限インパルス応答(FIR)フィルタリングを実行するときである。そのようなアプローチが図8に概略的に示されており、入力画像400が考慮されている。特に、FIRフィルタリング演算は、対応する出力画像430を生成するために入力画像に適用され、出力画像430内の各画素415、420は、対応するフィルタリング演算407、412の結果として生成される。これらのフィルタリング演算の各々に対して、複数の入力画素が考慮され、出力画素に対する値を生成するために、フィルタリング係数がこれらの複数の入力画素に関して適用される。

10

【0074】

図8に示された例では、各出力画素は、入力画素の3×3アレイを考慮することによって生成されると仮定される。したがって、入力画素405の第1の3×3アレイがフィルタ演算407に提供され、ここで、画素415の出力値を生成するために、フィルタ係数の対応するアレイを使用してフィルタリングを実行する。同様に、入力画素410の第2の3×3アレイは、出力画素420のための値を生成するために、フィルタ係数の対応するセットを使用してフィルタ演算412にかけられる。アレイ410は、アレイ405に対して1画素位置だけ右にシフトされ、上述のプロセスが繰り返されるにつれて、3×3画素のスライディングウィンドウが、各フィルタ演算への入力として働くように抽出されることが認められるであろう。行の終わりに達すると、プロセスは、入力画像の左側に戻ることができるが、画像内の1つ下の行から開始し、再び画像の左から右に進む。したがって、図8に示される構成において、最初に「水平」方向に移動するスライディングウィンドウが効果的に存在し、スライディングウィンドウは、各出力画像画素を計算するときに使用される入力画像画素の3×3アレイを抽出する入力画像にわたって移動することが理解されるであろう。

20

【0075】

図9A～図9Dは、アレistroレージ90内の正方形2Dアレイを使用して、その2Dアレイ内に保持されたデータに対して外積累積和演算を実行することによって、そのような2D画像フィルタリング演算をどのように効率的に実行できるかを示す。図9A～図9Dに示す例では、スライディングウィンドウは、最初に「垂直」方向に、すなわち図8の例に示された方向に対して直交する方向に移動することに留意されたい。したがって、図9A～図9Dに示される入力画像440は、その通常の視線方向に対して横にあると考えることができる。

30

【0076】

図9A～図9Dに示されるように、入力画像440の1つの列(又は画像の一部)が一度に処理され、フィルタ係数のベクトルを使用してフィルタリング演算を受ける。フィルタ係数のベクトル内の各ブロック(例えば、図9Aのブロック465を参照)は、係数の3×3アレイからの3つのフィルタ係数を表す。パディング要素(例えば、図9Aの要素467を参照)は、ゼロ又は未定義値に対応し、示された実装形態の単なるアーチファクトである。具体的には、図示された実装形態では、使用される命令は、結果ごとに4つの乗算及び累積和を実行することができるが、図示された例示的な実装形態では、3つの乗算及び累積和を実行することのみが必要とされる。

40

【0077】

図示された例では、係数の4つのセット(例えば、3×3係数の4つのセットの1つの行から取られた図9Aの4つのブロック468を参照)が、提供された入力ベクトルに対する4つの出力ベクトルを計算するために入力として使用される((3+1)係数の1つ

50

のセットが、4つの出力ベクトルを生成するために使用される4つの乗算演算の各々において使用される)。加えて、図9A～図9Dに示されたプロセスは、命令ごとに係数の各 3×3 アレイからの3つの係数の1つの列のみを使用し、したがって、4つのベクトルの任意の特定のセット内で最終累積和結果を生成するために3つの命令を要する。

【0078】

プロセスが進行中であり、定常状態にあるとき、図9C及び図9Dを参照して後述するように、プロセスは、4つの出力ベクトルの3つのセットに対して同時に動作している（例えば、図9Cに示された出力ベクトル475、485、495の3つのセットを参照）。

【0079】

図9Aに示されるように、入力画像440の第1の列470が係数のベクトル460を使用して処理されるとき、これは累積和結果を4つのベクトル475内に記憶させる。図9Bに示されるように、入力画像440の第2の列480が係数のベクトル482を使用して処理されるとき、これは、4つのベクトル475及び4つのベクトル485の両方の中に累積和結果をポピュレートする。次に、第3列490が、係数のベクトル492を使用して、図9Cに示されるように処理されるとき、これは、累積和結果を4つのベクトル475、4つのベクトル485、及び4つのベクトル495内にポピュレートさせる。図8の前述の説明から、この時点で、入力画像の最初の3つの列内のすべての画素が処理されることになり、したがって、4つの水平ベクトル475の内容が、出力画像の列の第1のグループに対する最終累積和結果を表すことが理解されよう。したがって、図9Cに示されるように、これらの4つのレジスタ475の内容は、それらの内容を宛先ストレージ、例えばベクトルレジスタファイル80内の4つのベクトルレジスタに移動させ、2Dアレイ450内のこれらの4つのベクトル475を形成するストレージ要素をクリアしてそれらが後続の累積和演算に利用可能になるようにするために、前述の移動及びゼロ命令を受けることができる。

【0080】

したがって、例として、図9Dに示されるように、入力画像440の第4の列500が、係数のベクトル460を使用して処理されるとき、これは、累積和結果を4つのベクトル485、4つのベクトル495、及び4つのベクトル475内にポピュレートさせることができる（これは、これらのベクトル内のストレージ要素が前の移動及びゼロ命令によって論理ゼロ値にクリアされているため、ここで再使用することができる）。また、図9Dに示されるように、4つのベクトル485は、ここで、出力画像の列の第2のグループを表す最終累積和結果を記憶する（この時点で、第2、第3、及び第4の入力列の各々が処理されているからである）。したがって、それらの4つの水平ベクトル485内に記憶されたデータ要素は、アレイからベクトルレジスタファイルのベクトルレジスタに移動され得、次いで、基礎となるストレージ要素は、後続の累積和演算におけるそれらの再使用を可能にするためにクリアされる。

【0081】

図9A～図9Dでは、移動及びゼロ命令を受けた任意のベクトルが直ちに再使用されるが（これはプログラミングをより容易にすることができるため）、クリアされたベクトルを直ちに再使用する必要はなく、代わりに、処理は、必要に応じて、ベクトルのグループ475、485、495より下のベクトルを消費し続けることができ、アレイ450の底部に到達したときに始めに戻るだけである。

【0082】

例示的な一実装形態では、アレイストレージ90内の所与の正方形2Dアレイは、水平方向又は垂直方向のいずれかにおいてアクセスされ得る。しかしながら、いくつかの実装形態では、ベクトルが方向のうちの1つでしかアクセスされないアレイストレージ90内の2Dアレイを使用して実行することができる特定の処理動作がある。したがって、例として、前に説明した図2に戻って参照すると、アレイを通して第1のアレイ方向に延びるアレイベクトルレジスタZARを具体的に識別するいくつかの処理命令が存在し得る。こ

10

20

30

40

50

これらの命令は、関連する処理演算及び累積和演算が実行される複数のZARレジスタを指定することによって、特定の処理演算及び累積和演算の非常に効率的な実行を可能にすることができる。しかし、一連のこれらの命令が実行されると、識別されたアレイベクトルレジスタZARのすべては、通常、最終累積和結果を含むが、これらの結果がアレイストレージから移動され、これらのアレイベクトルレジスタを形成するストレージ要素の現在の内容が論理ゼロ値にクリアされるまで、後続の処理演算及び累積和演算のためにこれらのレジスタを再使用することは不可能である。

【0083】

図10は、そのような状況において性能を大幅に向上させるために、前述の移動及びゼロ命令をどのように使用することができるかを概略的に示す。具体的には、図10に示すように、3つのアレイベクトルレジスタZAR2、ZAR3、及びZAR4がゼロに初期化され、次いで、処理演算及び累積和演算を実行するために、上記のタイプの一連のデータ処理命令が実行され、累積和結果が上記の3つのアレイベクトルレジスタ内に維持されると仮定する。必要とされる一連のデータ処理命令が完了すると（この例では、2つのそのようなデータ処理命令が実行されると仮定される）、上記の3つのアレイベクトルレジスタのすべてが最終累積和結果を記憶する。したがって、前述の移動及びゼロ命令を使用して、それらの3つのアレイベクトルレジスタを、そのデータ要素が宛先ストレージに移動されるべきベクトルとして指定することができ、宛先ストレージとして使用されるストレージを識別することもでき、この例では、ベクトルレジスタファイル80内の3つの隣接するベクトルレジスタが使用されると仮定する。したがって、移動及びゼロ命令の実行は、累積和結果のすべてをアレイストレージからベクトルレジスタファイルの識別されたベクトルレジスタに移動させ、また、それらの3つのアレイベクトルレジスタを形成するストレージ要素を論理ゼロ値にクリアさせる。したがって、図10に示すように、プロセスは、次に、同じ一連のアレイベクトルレジスタZAR2、ZAR3、及びZAR4にやはり累積される後続のデータ処理命令のシーケンスを実行するように直ちに進むことができる。これは、非常に効率的な実装形態を提供する。

【0084】

本明細書で説明する別の技術によれば、命令の追加の新たな形態（本明細書ではゼロベクトル命令と呼ばれる）が提供され、このゼロベクトル命令は、アレイストレージ内のデータ要素のベクトルをゼロにし、そのようなアレイストレージを使用して累積和演算を実行するときに（1つ以上のベクトルレジスタからアレイストレージの所望のベクトルにゼロをパスするために移動命令を使用する必要がある実装形態と比較して）性能の改善をもたらすために使用することもできる。この追加の技術によれば、命令デコード回路50は、そのようなゼロベクトル命令（アレイストレージ内のデータ要素の所与の2次元アレイのデータ要素の複数のベクトルを識別するように構成されている）のデコードに応答して、データ要素の識別された複数のベクトルに対して動作するように構成されている後続の累積和命令もデコードするように構成されている。次いで、処理回路は、識別された複数のベクトルのデータ要素を記憶するために使用されるアレイストレージのストレージ要素を論理ゼロ値に設定し、次いで、累積和命令によって指定された累積和演算を実行して、アレイストレージ内の識別された複数のベクトルに記憶するための結果データ要素を生成するようにされる。

【0085】

上記のアプローチを使用することによって、ゼロベクトル命令は、命令デコード回路によるデコード時に、ゼロベクトル命令によって指定されるのと同じデータ要素の複数のベクトルを指定する後続の累積和命令とマージされて、事実上、その累積和命令の非累積変形形態を作成することができる。これは、命令符号化空間が通常は非常に重要であり、アレイストレージ内のデータ要素の複数のベクトルに対して動作するように定義され得る様々な累積和命令の非累積変形形態を指定することが可能でない場合があるので、非常に有益であり得る。

10

20

30

40

50

【0086】

図11は、例示的な一実装形態によるそのようなゼロベクトル命令の処理を示す流れ図である。ステップ520において、デコーダ回路50がゼロベクトル命令に遭遇すると、ステップ525において、アレイストレージ内のデータ要素の複数のベクトルが、ゼロベクトル命令のベクトル識別フィールドを参照して識別される。例えば、複数のアレイベクトルレジスタZARをベクトル識別フィールドにおいて識別することができる。次に、ステップ530において、デコーダ回路は、デコードされるべき次の命令が、ゼロベクトル命令によって識別されるものと同じベクトルに対して動作している累積和命令であるかどうかを判定する。

【0087】

そうでない場合、ステップ535において、処理回路は、ステップ525において決定された識別された複数のベクトルのデータ要素を記憶するために使用されるアレイストレージのストレージ要素を論理ゼロ値に設定するように制御され、その後、処理は単に次の命令の実行を続ける。

【0088】

しかしながら、ステップ530において、次の命令が、ゼロベクトル命令によって識別されたものと同じベクトルに対して動作している累積和命令であると判定された場合、デコーダは、2つの命令を効果的に融合し、ステップ540において、識別された複数のベクトルの各々に記憶するための結果を生成するために、累積和命令によって指定された累積和演算の非累積変形形態を実行するように処理回路を制御する（通常、これは、処理演算と後続の累積の両方を含む）。前述のように、そのようなアプローチによって、アレイストレージ内の複数のベクトルに対して動作するように構成された任意の累積和命令の非累積変形形態を具体的に符号化する必要はない。なぜならば、そのような非累積変形形態は、ゼロベクトル命令と、それに続く必要な累積和命令との使用を通じて、上記の融合手順によって効果的に実行することができるからである（それによって、累積和命令によって定義された処理動作を実行するが、累積和関数は効果的に無効化される）。

【0089】

図12は、例示的な一実装形態におけるゼロベクトル命令内に提供され得るフィールドを概略的に示す図である。特に、ゼロベクトル命令550は、命令が実際にゼロベクトル命令であることを識別するためにそのビットが使用されるオペコードフィールド555を含む。更に、アレイストレージ内の複数のベクトルを識別するために使用されるベクトル識別フィールド560が提供される。このフィールド内の情報は、概して、図5Aの移動及びゼロ命令300のベクトル識別フィールド310を説明するときに前述した形態をとることができるが、例示的な一実装形態では、後続の累積和命令は、第1のアレイ方向に延びるアレイベクトルレジスタに対して動作するように構成された命令であり、したがって、典型的には、水平方向及び垂直方向の両方がベクトル識別フィールド560内で符号化されることを可能にする必要はない。

【0090】

必要に応じて、ボックス565によって示されるように、移動及びゼロ命令の例を参照して前述したような述語情報フィールド及びデータ要素サイズフィールドなど、いくつかの任意選択の追加フィールドが設けられてよい。

【0091】

図13は、別の例示的な実装形態によるゼロベクトル命令の処理を示す流れ図である。ステップ570において、デコーダ回路50がゼロベクトル命令に遭遇すると、ステップ575において、アレイストレージ内のデータ要素の複数のベクトルが、ゼロベクトル命令のベクトル識別フィールドを参照して識別される。例えば、複数のアレイベクトルレジスタZARをベクトル識別フィールドにおいて識別することができる。

【0092】

次いでステップ580において、処理回路は、ステップ575において決定された識別された複数のベクトルのデータ要素を記憶するために使用されるアレイストレージのスト

10

20

30

40

50

レージ要素を論理ゼロ値に設定するように制御され、その後、処理は単に次の命令の実行を続ける。

【0093】

この実装形態においても、ゼロベクトル命令を後続の累積和命令と組み合わせるために融合が行われない場合、著しい利益を依然として達成することができる。特に、複数の移動命令を実行する必要はなく、各移動命令は、ゼロのベクトルをベクトルレジスタファイルのベクトルレジスタからアレイストレージの識別されたベクトルに移動させる。更に、そのようなゼロ化機能は、（ゼロの）移動ベクトル機能を実行しなければならないよりも、ハードウェアにおいて構築するのがより簡単かつより安価である。更に、移動命令の使用に基づく上記の実装形態が採用された場合に必要とされるはずの論理ゼロ値を保持するためにベクトルレジスタファイル内に1つ以上のベクトルレジスタを予約する必要がないので、追加の節約がある。

10

【0094】

図14は、使用され得るシミュレータ実装形態を示す。前述の例は、当該技法をサポートする特定の処理ハードウェアを動作させるための装置及び方法の観点で本発明を実施するが、本明細書に記載の例による命令実行環境を提供することも可能であり、命令実行環境は、コンピュータプログラムの使用により実施される。このようなコンピュータプログラムは、コンピュータプログラムがハードウェアアーキテクチャのソフトウェアベースの実装形態を提供する限り、シミュレータと称されることがよくある。様々なシミュレータコンピュータプログラムは、エミュレータ、仮想マシン、モデル、及び動的バイナリトランスレータを含むバイナリトランスレータを含む。典型的に、シミュレータの実装形態は、ホストプロセッサ615において実行してもよく、任意選択でホストオペレーティングシステム610を実行し、シミュレータプログラム605をサポートする。いくつかの構成では、ハードウェアと提供された命令実行環境との間に複数層のシミュレーションがあり得、かつ／又は、複数の異なる命令実行環境が同じホストプロセッサ上に提供され得る。歴史的に、強力なプロセッサが、合理的な速度で実行されるシミュレータ実装形態を提供するのに必要とされてきたが、このようなアプローチは、互換性又は再使用の理由から別のプロセッサにネイティブなコードを実行することが望まれるようなときなど一定の状況では、正当化され得る。例えば、シミュレータ実装形態は、ホストプロセッサハードウェアによってサポートされていない追加の機能性を有する命令実行環境を提供し得るか、又は典型的には異なるハードウェアアーキテクチャに関連付けられた命令実行環境を提供し得る。シミュレーションの概要は、「Some Efficient Architecture Simulation Techniques」、Robert Bedichek、1990年冬、USENIX Conference、53～63頁に記載されている。

20

30

【0095】

例が、特定のハードウェア構築又は特徴を参照して前述されている程度に、シミュレートされた実装形態では、同等の機能が、好適なソフトウェア構築又は特徴によって提供され得る。例えば、特定の回路は、シミュレートされた実装形態では、コンピュータプログラムロジックとして提供され得る。同様に、レジスタ又はキャッシュなどのメモリハードウェアは、シミュレートされた実装形態では、ソフトウェアデータ構造として提供され得る。また、ハードウェア装置10内のメモリ30にアクセスするために使用される物理アドレス空間は、シミュレートされたアドレス空間としてエミュレートされ得、シミュレートされたアドレス空間は、ホストオペレーティングシステム610によって使用される仮想アドレス空間に、シミュレータ605によってマッピングされている。前述の例で言及されているハードウェア要素のうち1つ以上がホストハードウェア（例えば、ホストプロセッサ615）に存在する構成では、いくつかのシミュレートされた実装形態は、好適である場合、ホストハードウェアを使用し得る。

40

【0096】

シミュレータプログラム605は、（非一時的媒体であり得る）コンピュータ読み取り

50

可能な記憶媒体に記憶され得、（アプリケーション、オペレーティングシステム、及びハイパーバイザを含み得る）ターゲットコード 600 への仮想ハードウェアインタフェース（命令実行環境）を提供し、仮想ハードウェアインタフェースは、シミュレータプログラム 605 によってモデル化されたハードウェアアーキテクチャのハードウェアインタフェースと同じである。したがって、ターゲットコード 600 のプログラム命令は、シミュレータプログラム 605 を使用して命令実行環境内から実行され得、このため、上で考察される装置 10 のハードウェア特徴を実際には有しないホストコンピュータ 615 は、これらの特徴をエミュレートすることができる。シミュレータプログラムは、処理回路 60 の動作をエミュレートする処理プログラムロジック 620 と、命令デコード回路 50 の動作をエミュレートする命令デコードプログラムロジック 625 と、アレキストレージ 90 をエミュレートするためにデータ構造を維持するアレキストレージエミュレートプログラムロジック 622 とを含み得る。したがって、本明細書で説明される技術は、図 14 の例では、シミュレータプログラム 605 によってソフトウェアで実行され得る。

10

【0097】

本出願において、「～ように構成された（configured to...）」という用語は、装置の要素が、定義された動作を行うことが可能である構成を有することを意味するために使用される。この文脈において、「構成」とは、ハードウェア又はソフトウェアの配置又は相互接続の方法を意味する。例えば、装置は、定義された動作を提供する専用ハードウェアを有し得るか、又はプロセッサ若しくは他の処理デバイスが、機能を実施するようにプログラムされ得る。「ように構成された」は、装置要素が、定義された動作を提供するために何らかの変更がなされる必要があることを意味しない。

20

【0098】

本明細書では例示的な例について添付の図面を参照して詳細に説明してきたが、理解されるように、本発明はそれらと寸分違わない構成に制限されるのではなく、また、添付の特許請求の範囲に定める本発明の範囲及び趣旨から逸脱することなく、当業者によって様々な変更、追加、及び修正を行うことができる。例えば、従属請求項の特徴の様々な組み合わせは、本発明の範囲から逸脱することなく、独立請求項の特徴でなされ得る。

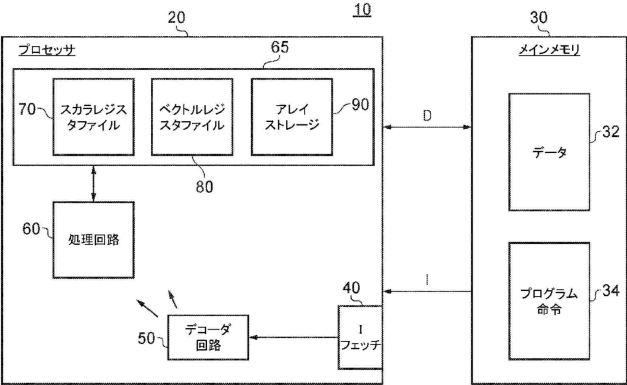
30

40

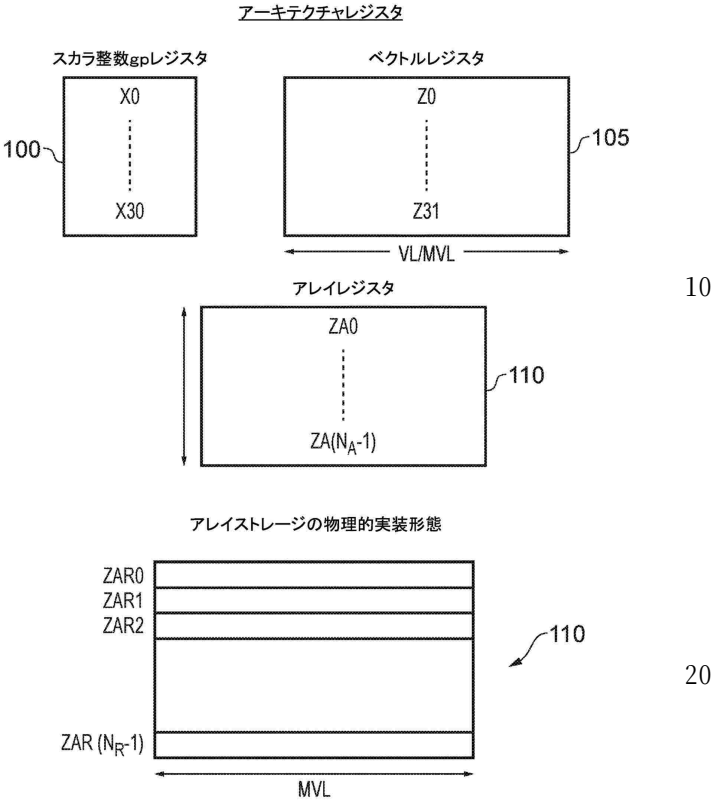
50

【図面】

【図 1】

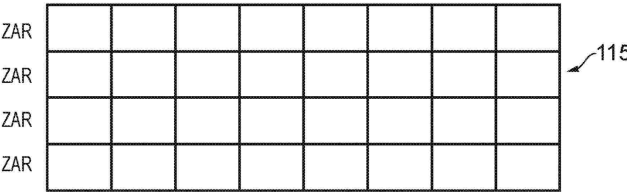


【図 2】



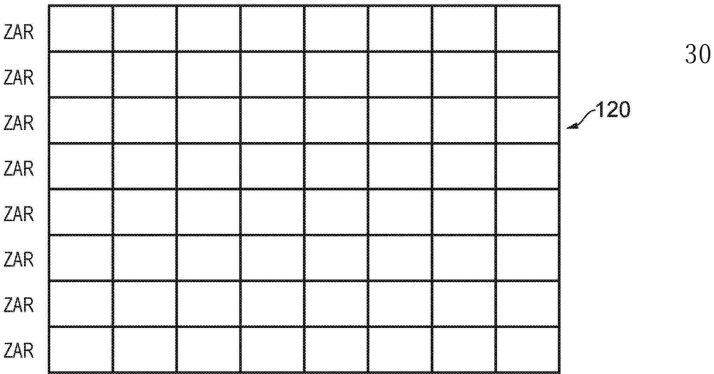
【図 3 A】

データ要素の非正方形2Dアレイ



【図 3 B】

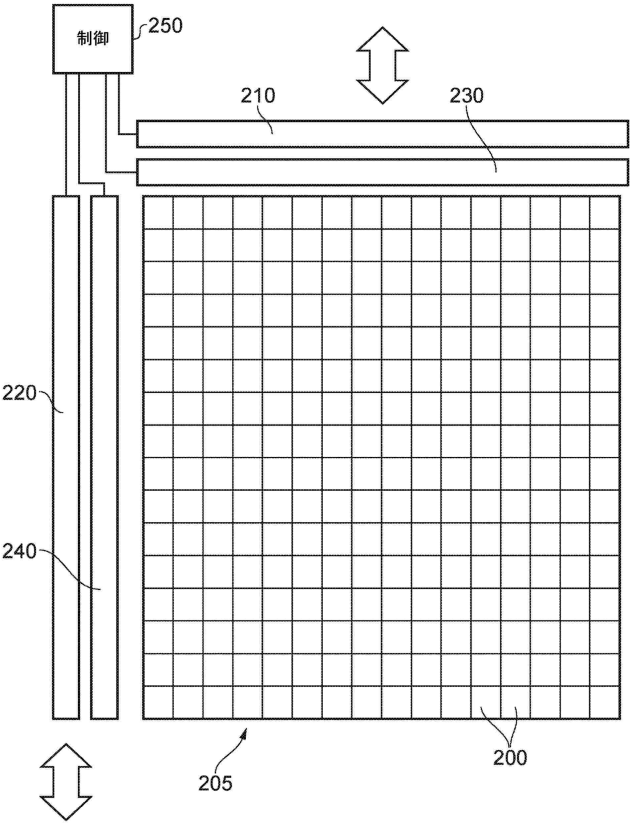
データ要素の正方形2Dアレイ



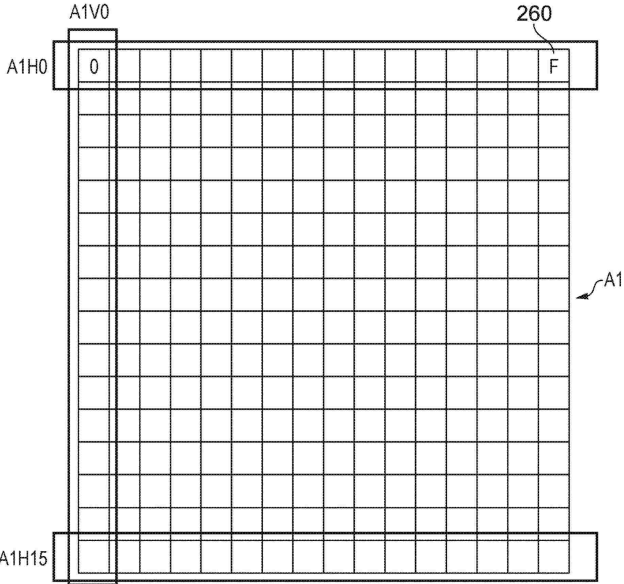
40

50

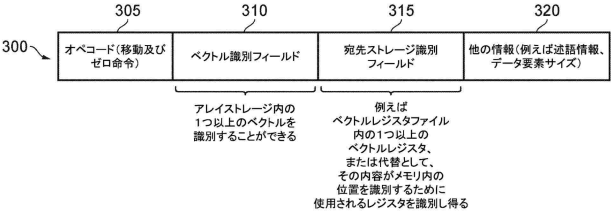
【図 4 A】



【図 4 B】

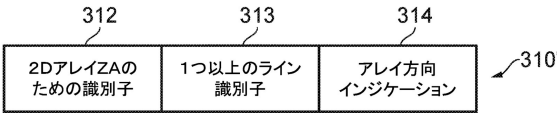


【図 5 A】



【図 5 B】

ベクトル識別フィールドの例



10

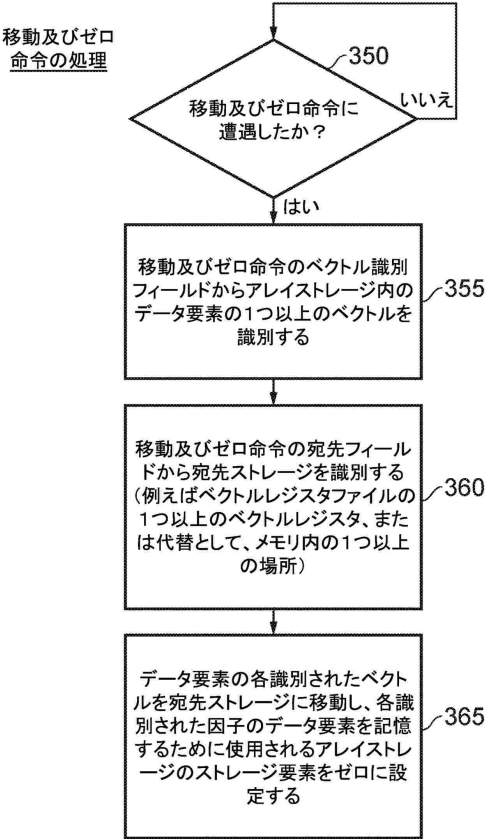
20

30

40

50

【図 6】



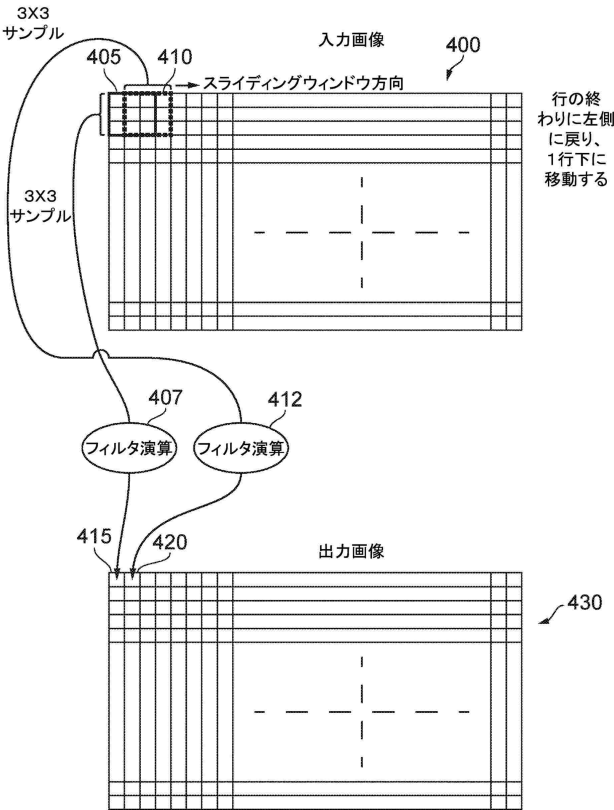
【図 7】

アレイドレージ上で動作する命令シーケンスの例

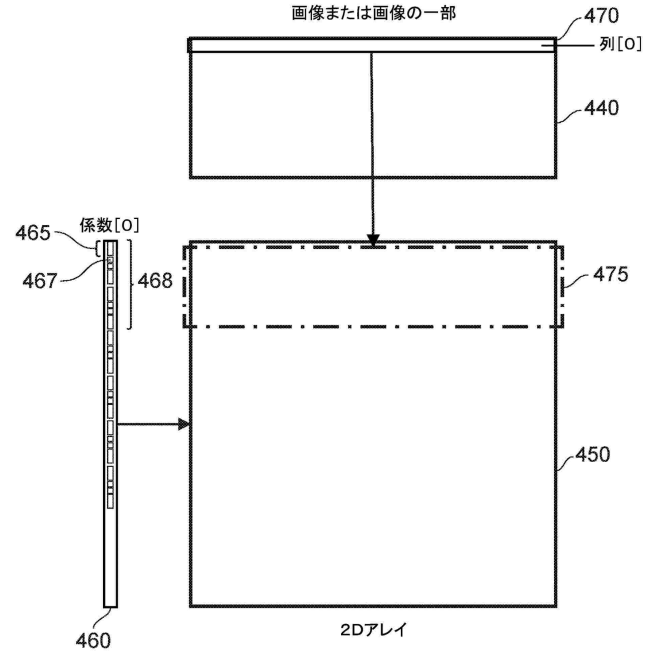
- 1) データ処理命令1 (2Dアレイドレージに累積和する)
- 2) データ処理命令2 (2Dアレイドレージに累積和する)
- 3) データ処理命令3 (2Dアレイドレージに累積和する)
- 4) 移動及びゼロ命令 (垂直ベクトル1、ベクトルレジスタ Z_i)
— 垂直ベクトル1は最終累積和結果を保持し、
移動及びゼロ命令の実行はこれらの最終累積和結果を Z_i へ移動し、垂直ベクトル1のためのストレージ要素をクリアする
- 5) データ処理命令4 (2Dアレイドレージに累積和する—
垂直列1を再使用することができる)
- 6) 移動及びゼロ命令 (垂直ベクトル2、ベクトルレジスタ Z_{i+x})
— 垂直ベクトル2は最終累積和結果を保持し、移動及びゼロ命令の実行はこれらの最終累積和結果を Z_{i+x} へ移動し、垂直ベクトル2のためのストレージ要素をクリアする
- 7) データ処理命令5 (2Dアレイドレージに累積和する—
垂直列2を再使用することができる)

【図 8】

スライディングウィンドウアプローチを使用するFIRフィルタリング



【図 9 A】



10

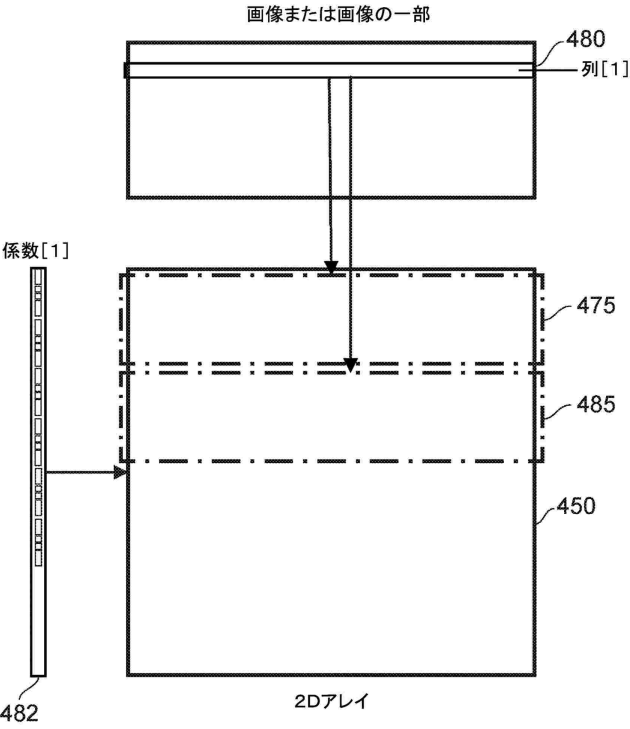
20

30

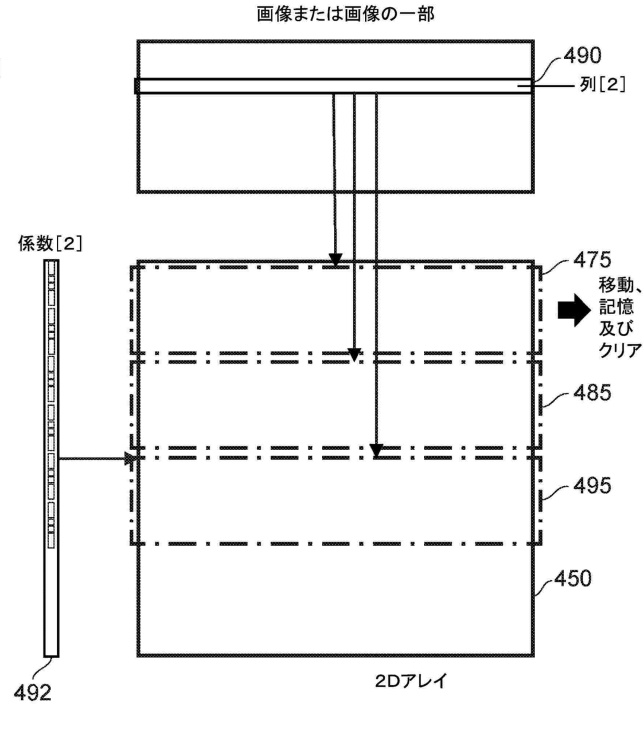
40

50

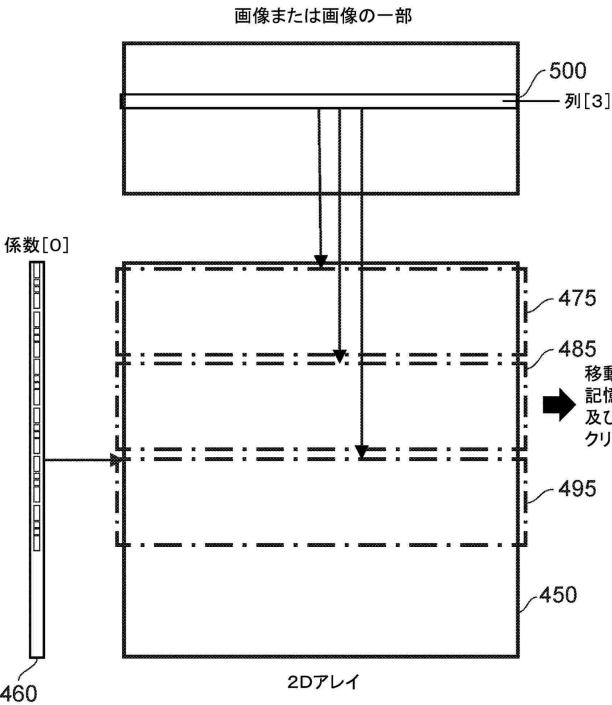
【図 9 B】



【図 9 C】



【図 9 D】



【図 1 0】

アレイドレジスタ上で動作する命令シーケンスの代替例

- 0に初期化されたZAR 2、ZAR 3及びZAR 4を仮定する
- 1) データ処理命令 (ZAR 2、ZAR 3及びZAR 4に累積和する)
 - 2) データ処理命令 (ZAR 2、ZAR 3及びZAR 4に累積和する)
 - 3) 移動及びゼロ命令 (ZAR 2、ZAR 3、ZAR 4へ Z_i 、 Z_{i+1} 、 Z_{i+2})
— すべての累積和結果をベクトルレジスタファイル内のベクトルレジスタに移動し、
ZAR 2、ZAR 3及びZAR 4をクリアする
 - 4) データ処理命令 (ZAR 2、ZAR 3、及びZAR 4に累積和する)
実行は、ZAR 2、ZAR 3、及びZAR 4のそれぞれにおいて生成されかつ記憶された結果を用いて、実行される定義された演算の非累積和変形である有効性を引き起こす。
 - 5) データ処理命令 (ZAR 2、ZAR 3及びZAR 4に累積和する)

10

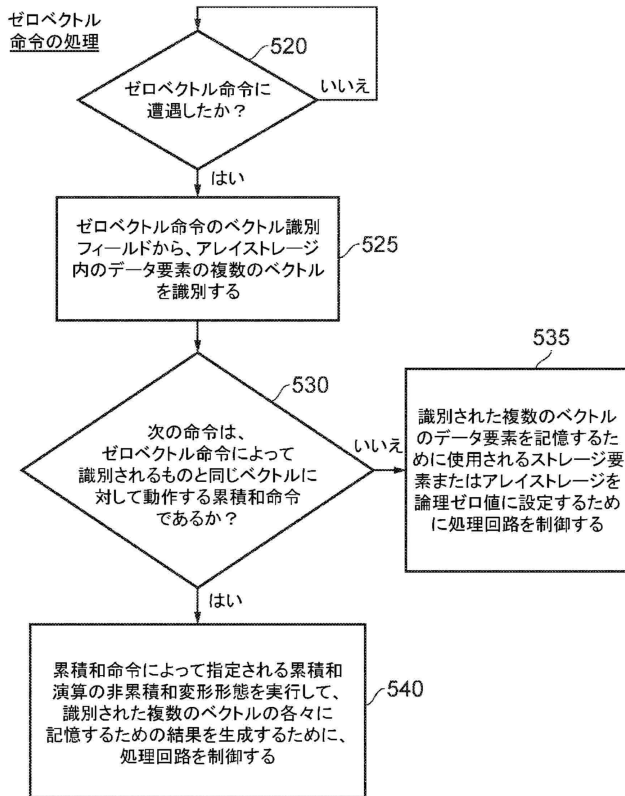
20

30

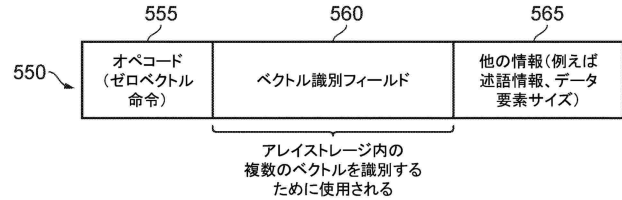
40

50

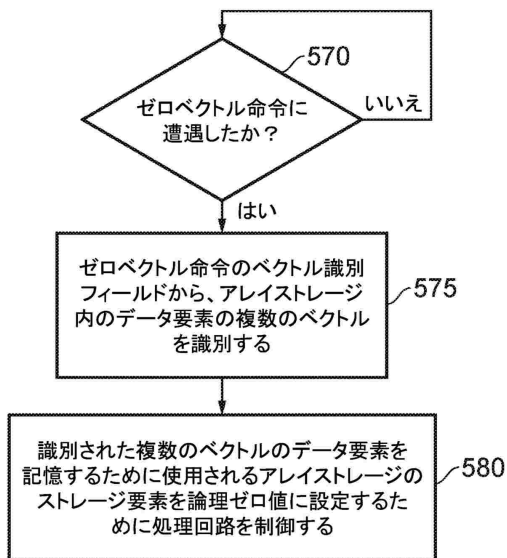
【図 1 1】



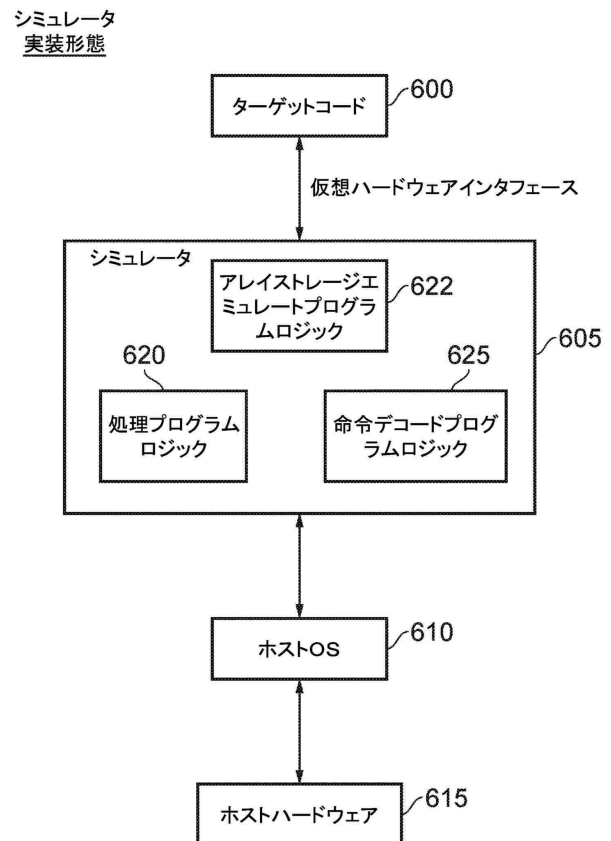
【図 12】



【図 13】



【図 14】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2023/050584

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F9/30 ADD. According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2021/042261 A1 (MILANOVIC JELENA [FR] ET AL) 11 February 2021 (2021-02-11) the whole document -----	1-15
Y	WO 2005/013084 A2 (CRADLE TECHNOLOGIES INC [US]) 10 February 2005 (2005-02-10) page 17, line 4 - line 30 -----	1-15
A	US 3 544 965 A (PACKARD ROGER E) 1 December 1970 (1970-12-01) the whole document -----	1, 14, 15
A	EP 3 629 154 A2 (INTEL CORP [US]) 1 April 2020 (2020-04-01) the whole document -----	1-15
A	EP 3 929 736 A1 (INTEL CORP [US]) 29 December 2021 (2021-12-29) the whole document -----	1-15
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
6 June 2023		14/06/2023
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Moraiti, Marina

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2023/050584

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2021042261 A1	11-02-2021	CN 113490914 A	08-10-2021
		EP 3931688 A1	05-01-2022
		IL 285752 A	31-10-2021
		JP 2022543332 A	12-10-2022
		KR 20220038579 A	29-03-2022
		TW 202107279 A	16-02-2021
		US 2021042261 A1	11-02-2021
		WO 2021023957 A1	11-02-2021
WO 2005013084 A2	10-02-2005	US 2005027773 A1	03-02-2005
		US 2005027901 A1	03-02-2005
		US 2005027969 A1	03-02-2005
		US 2005235026 A1	20-10-2005
		WO 2005013084 A2	10-02-2005
US 3544965 A	01-12-1970	NONE	
EP 3629154 A2	01-04-2020	CN 110955454 A	03-04-2020
		CN 114237713 A	25-03-2022
		CN 114625418 A	14-06-2022
		EP 3629154 A2	01-04-2020
		EP 4141661 A1	01-03-2023
		EP 4177738 A1	10-05-2023
		US 2020104135 A1	02-04-2020
		US 2021318874 A1	14-10-2021
		US 2022100505 A1	31-03-2022
EP 3929736 A1	29-12-2021	CN 113849224 A	28-12-2021
		EP 3929736 A1	29-12-2021
		US 2021406018 A1	30-12-2021

フロントページの続き

,MC,ME,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,
ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,C
O,CR,CU,CV,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,I
R,IS,IT,JM,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,MG,MK,MN,MW,MX
,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,
SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW
フランス共和国 0 6 4 1 0 ソフィア アンティポリス ビオット アベニュー ド ルーマニル 7
3 8 アーム フランス エスエーエス宛
F ターム (参考) 5B056 BB28 BB32 BB71 FF10