(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7401513号
(P7401513)

(45)発行日　令和5年12月19日(2023.12.19)

(24)登録日　令和5年12月11日(2023.12.11)

(51)国際特許分類
Ｇ０６Ｆ　17/16　(2006.01)

ＦＩ
Ｇ０６Ｆ　17/16　Ｐ

請求項の数　23　外国語出願　（全31頁）

(21)出願番号　特願2021-207147(P2021-207147)
(22)出願日　令和3年12月21日(2021.12.21)
(65)公開番号　特開2022-181161(P2022-181161A)
(43)公開日　令和4年12月7日(2022.12.7)
審査請求日　令和4年4月6日(2022.4.6)
(31)優先権主張番号　17/329,259
(32)優先日　令和3年5月25日(2021.5.25)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　502208397
グーグル エルエルシー
Ｇｏｏｇｌｅ ＬＬＣ
アメリカ合衆国 カリフォルニア州 ９４０４３ マウンテン ビュー アンフィシアター パークウェイ １６００
１６００ Ａｍｐｈｉｔｈｅａｔｒｅ Ｐａｒｋｗａｙ ９４０４３ Ｍｏｕｎｔａｉｎ Ｖｉｅｗ， ＣＡ Ｕ．Ｓ．Ａ．
(74)代理人　110001195
弁理士法人深見特許事務所
(72)発明者　レイナー・アルウィン・ポープ
アメリカ合衆国、９４０４３ カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、１６００

最終頁に続く

(54)【発明の名称】　ハードウェアにおけるスパース行列乗算

(57)【特許請求の範囲】
【請求項１】
システム入力行列とシステム入力ベクトルとの間の行列乗算を実行するように構成されたシステムであって、複数の乗算回路を含む回路の集合体を備えるスパースシャードを備え、前記スパースシャードは、
前記システム入力行列の複数の部分行列のいずれかであって予め定められた最大非ゼロ閾値以下の数の非ゼロ値を含むシャード入力行列を受けるように構成され、前記シャード入力行列は、前記スパースシャードの最大行列入力サイズを指定する予め定められた次元閾値以下の次元を有し、前記スパースシャードはさらに、
前記システム入力ベクトルの複数の部分ベクトルのいずれかであって複数のベクトル値を含むシャード入力ベクトルを受け、
前記乗算回路の各々について、前記シャード入力行列のそれぞれの非ゼロ値を受け、
前記複数の乗算回路によって、ベクトル値に前記シャード入力行列の前記それぞれの非ゼロ値を乗算した１つ以上の積を生成し、
前記スパースシャードへの出力として、前記１つ以上の積を用いて、前記シャード入力ベクトルを前記シャード入力行列に適用した積であるシャード出力ベクトルを生成するように構成されている、システム。
【請求項２】
前記シャード出力ベクトルの長さは１よりも大きい、請求項１に記載のシステム。
【請求項３】

前記スパースシャードは複数のスパースシャードのうちの１つであり、前記複数のスパースシャードは、
前記システム入力行列の部分行列である複数のシャード入力行列を受け、
前記システム入力ベクトルの部分ベクトルである複数のシャード入力ベクトルを受け、
前記複数のスパースシャードによって、前記システム入力ベクトルを前記システム入力行列に適用した積を表すシステム出力ベクトルを生成するように構成されている、請求項１に記載のシステム。
【請求項４】
前記複数のスパースシャードはシストリックアレイとして配列され、前記シストリックアレイは、前記シストリックアレイの列次元に沿ったスパースシャードの１つ以上のグループを含み、
前記システム出力ベクトルを生成するために、前記複数のスパースシャードはさらに、
前記シストリックアレイの前記列次元に沿ったグループごとに、前記グループ内の各スパースシャードのそれぞれのシャード出力ベクトルを合算してそれぞれの列出力ベクトルを生成し、
各グループの前記それぞれの列出力ベクトルを連結して前記システム出力ベクトルを生成するように構成されている、請求項３に記載のシステム。
【請求項５】
各乗算回路は、前記スパースシャードについての前記シャード入力行列のそれぞれからの前記それぞれの非ゼロ値を含むそれぞれのレジスタに結合される、請求項１に記載のシステム。
【請求項６】
前記複数の乗算回路における乗算回路の数は、前記予め定められた最大非ゼロ閾値と等しい、請求項５に記載のシステム。
【請求項７】
前記スパースシャードはクロスバー回路をさらに含み、前記スパースシャードはさらに、
前記クロスバー回路によって、前記シャード入力ベクトルの前記複数のベクトル値を受け、
前記複数の乗算回路の各々への入力として、前記クロスバー回路によって、前記複数のベクトル値のうちの１つのベクトル値を送るように構成されている、請求項５に記載のシステム。
【請求項８】
前記スパースシャードはさらに、
前記シャード入力行列の同じ列の非ゼロ値を、前記複数の乗算回路のうちの隣接する乗算回路のレジスタにロードするように構成されている、請求項７に記載のシステム。
【請求項９】
前記スパースシャードはさらに、前記シャード入力行列の各列に沿った非ゼロ値の位置を少なくとも指定する１つ以上の制御値を受けるように構成されており、
前記スパースシャードの前記クロスバー回路はさらに、
前記１つ以上の制御値を受け、
前記１つ以上の制御値に従って、前記シャード入力行列の同じ列に沿った非ゼロ値が乗算されるベクトル値を、隣接する乗算回路に送るように構成されている、請求項８に記載のシステム。
【請求項１０】
前記スパースシャードは複数の加算回路をさらに含み、
前記スパースシャードは１つ以上のセグメントマーカをさらに含み、各セグメントマーカは、前記セグメントマーカにロードされたそれぞれの制御値の値に基づいて、前記複数の加算回路のうちのそれぞれの加算回路への入力をゲート開閉するように構成されており、
前記スパースシャードはさらに、
前記１つ以上の制御値の少なくとも一部を前記１つ以上のセグメントマーカにロード

するように構成されており、前記シャード入力行列の１列目の非ゼロ値についての加算回路は、前記１列目とは異なる前記シャード入力行列の２列目の非ゼロ値を含む隣接する加算回路の入力を受けることからゲート開閉されており、前記スパースシャードはさらに、

前記複数の加算回路によって、前記１つ以上の積の１つ以上の合計を生成するように構成されており、前記１つ以上の合計の各々は、前記シャード入力行列の列の１つ以上の非ゼロ値に前記シャード入力ベクトルの１つ以上のそれぞれの値を乗算したそれぞれのセグメント化合計である、請求項９に記載のシステム。

【請求項１１】

前記複数の加算回路は並列セグメント化合計回路を形成し、１つ以上の前記セグメント化合計の各々は、セグメントマーカによってゲート開閉されていない隣接する加算回路への出力の合計である、請求項１０に記載のシステム。

【請求項１２】

前記クロスバー回路は第１のクロスバー回路であり、

前記スパースシャードは第２のクロスバー回路をさらに含み、前記第２のクロスバー回路は、

１つ以上の前記セグメント化合計を受け、

前記１つ以上の制御値に従って１つ以上の前記セグメント化合計を配列して、前記スパースシャードついて前記シャード出力ベクトルのそれぞれを生成するように構成されている、請求項１０に記載のシステム。

【請求項１３】

前記第２のクロスバー回路はベネシュネットワークを形成し、前記シャード入力行列は正方行列である、請求項１２に記載のシステム。

【請求項１４】

前記複数の部分行列は、第１の部分行列および第２の部分行列を含み、

前記第１の部分行列の要素数は、前記第２の部分行列の要素数とは異なる、請求項１～１３のいずれか１項に記載のシステム。

【請求項１５】

システム入力行列とシステム入力ベクトルとの間の行列乗算を実行する方法であって、

複数の乗算回路を含む回路の集合体を備えるスパースシャードが、前記システム入力行列の複数の部分行列のいずれかであって予め定められた最大非ゼロ閾値以下の数の非ゼロ値を含むシャード入力行列と、前記システム入力ベクトルの複数の部分ベクトルのいずれかであって複数のベクトル値を含むシャード入力ベクトルとを受けることを備え、前記シャード入力行列は、前記スパースシャードの最大行列入力サイズを指定する予め定められた次元閾値以下の次元を有し、前記方法はさらに、

前記乗算回路の各々について、前記シャード入力行列の前記非ゼロ値のそれぞれを受けることと、

前記スパースシャードの前記複数の乗算回路が、それぞれのベクトル値に前記シャード入力行列の前記非ゼロ値のそれぞれを乗算した１つ以上の積を生成することと、

前記スパースシャードが、前記スパースシャードへの出力として、前記１つ以上の積を用いて、前記シャード入力ベクトルを前記シャード入力行列に適用した積であるシャード出力ベクトルを生成することとを備える、方法。

【請求項１６】

前記スパースシャードが前記シャード出力ベクトルを生成することは、１よりも大きい長さの前記シャード出力ベクトルを生成することを含む、請求項１５に記載の方法。

【請求項１７】

前記スパースシャードは複数のスパースシャードのうちの１つであり、前記複数のスパースシャードは、

前記システム入力行列の部分行列である複数のシャード入力行列を受け、

前記システム入力ベクトルの部分ベクトルである複数のシャード入力ベクトルを受け、

前記複数のスパースシャードによって、前記システム入力ベクトルを前記システム入力

行列に適用した積を表すシステム出力ベクトルを生成するように構成されている、請求項１５に記載の方法。
【請求項１８】
前記複数のスパースシャードはシストリックアレイとして配列され、前記シストリックアレイは、前記シストリックアレイの列次元に沿ったスパースシャードの１つ以上のグループを含み、
前記システム出力ベクトルを生成することは、
前記シストリックアレイの前記列次元に沿ったグループごとに、前記グループ内の各スパースシャードのそれぞれのシャード出力ベクトルを合算してそれぞれの列出力ベクトルを生成することと、
各グループの前記それぞれの列出力ベクトルを連結して前記システム出力ベクトルを生成することとを含む、請求項１７に記載の方法。
【請求項１９】
各乗算回路は、前記スパースシャードについての前記シャード入力行列のそれぞれからのそれぞれの非ゼロ値を含むそれぞれのレジスタに結合される、請求項１５に記載の方法。
【請求項２０】
前記複数の乗算回路における乗算回路の数は、前記予め定められた最大非ゼロ閾値と等しい、請求項１９に記載の方法。
【請求項２１】
前記複数の部分行列は、第１の部分行列および第２の部分行列を含み、
前記第１の部分行列の要素数は、前記第２の部分行列の要素数とは異なる、請求項１５～２０のいずれか１項に記載の方法。
【請求項２２】
システム入力行列とシステム入力ベクトルとの間の行列乗算を実行するように構成されたシステムによって実行されると前記システムに動作を実行させる命令を格納した１つ以上のコンピュータプログラムであって、前記動作は、
複数の乗算回路を含む回路の集合体を備えるスパースシャードによって、前記システム入力行列の複数の部分行列のいずれかであって予め定められた最大非ゼロ閾値以下の数の非ゼロ値を含むシャード入力行列と、前記システム入力ベクトルの複数の部分ベクトルのいずれかであって複数のベクトル値を含むシャード入力ベクトルとを受けることを備え、
前記シャード入力行列は、前記スパースシャードの最大行列入力サイズを指定する予め定められた次元閾値以下の次元を有し、前記動作はさらに、
前記乗算回路の各々について、前記シャード入力行列のそれぞれの非ゼロ値を受けることと、
前記スパースシャードの前記複数の乗算回路によって、それぞれのベクトル値に前記それぞれの非ゼロ値を乗算した１つ以上の積を生成することと、
前記スパースシャードへの出力として、前記１つ以上の積を用いて、前記シャード入力ベクトルを前記シャード入力行列に適用した積であるシャード出力ベクトルを生成することとを備える、１つ以上のコンピュータプログラム。
【請求項２３】
前記複数の部分行列は、第１の部分行列および第２の部分行列を含み、
前記第１の部分行列の要素数は、前記第２の部分行列の要素数とは異なる、請求項２２に記載の１つ以上のコンピュータプログラム。
【発明の詳細な説明】
【背景技術】
【０００１】
背景
スパース行列は、行列の要素として非ゼロ値よりもゼロ値の方が割合が高い行列である。異なるスパース行列は、ゼロ値と非ゼロ値との割合に基づいてさまざまな程度のスパース性を有し得る。非ゼロ値よりもゼロ値の方が割合が高い行列は、非ゼロ値よりもゼロ値

の方が割合が低い行列よりもスパース性が高いと言われる。
【０００２】
ニューラルネットワークは、受けた入力に対する出力を予測するための１つ以上の非線形演算層を含む機械学習モデルである。入力層および出力層に加えて、１つ以上の隠れ層を含むニューラルネットワークもある。各隠れ層の出力は、ニューラルネットワークの別の隠れ層または出力層に入力され得る。ニューラルネットワークの各層は、当該層の１つ以上のモデルパラメータの値に従って、受けた入力からそれぞれの出力を生成することができる。モデルパラメータは、ニューラルネットワークが正確な出力を生成するように訓練アルゴリズムを通して求められる重みまたはバイアスであってもよい。ニューラルネットワークの層のモデルパラメータ値は、行列またはテンソルの要素として表すことができる。
【発明の概要】
【課題を解決するための手段】
【０００３】
簡単な概要
本開示の局面は、ハードウェアにおけるスパース行列密ベクトル乗算に関する。
【０００４】
本開示の一局面は、複数の乗算回路を含むスパースシャードを含むシステムを提供し、上記スパースシャードは、予め定められた最大非ゼロ閾値以下の数の非ゼロ値を含むシャード入力行列を受け、複数のベクトル値を含むシャード入力ベクトルを受け、上記乗算回路の各々について、上記シャード入力行列のそれぞれの非ゼロ値を受け、上記複数の乗算回路によって、ベクトル値に上記シャード入力行列の上記それぞれの非ゼロ値を乗算した１つ以上の積を生成し、上記スパースシャードへの出力として、上記１つ以上の積を用いて、上記シャード入力ベクトルを上記シャード入力行列に適用した積であるシャード出力ベクトルを生成するように構成されている。
【０００５】
本開示の別の局面は、複数のスパースシャードを含むシステムによって実行されると上記システムに動作を実行させる命令を格納した１つ以上の非一時的なコンピュータ読取可能記憶媒体を提供し、上記動作は、複数の乗算回路を含むスパースシャードによって、予め定められた最大非ゼロ閾値以下の数の非ゼロ値を含むシャード入力行列と、複数のベクトル値を含むシャード入力ベクトルとを受けることと、上記乗算回路の各々について、上記シャード入力行列のそれぞれの非ゼロ値を受けることと、上記スパースシャードの上記複数の乗算回路によって、それぞれのベクトル値に上記それぞれの非ゼロ値を乗算した１つ以上の積を生成することと、上記スパースシャードへの出力として、上記１つ以上の積を用いて、上記シャード入力ベクトルを上記シャード入力行列に適用した積であるシャード出力ベクトルを生成することとを含む。
【０００６】
本開示の別の局面は、方法を提供し、上記方法は、複数の乗算回路を含むスパースシャードが、予め定められた最大非ゼロ閾値以下の数の非ゼロ値を含むシャード入力行列と、複数のベクトル値を含むシャード入力ベクトルとを受けることと、上記乗算回路の各々について、上記シャード入力行列の上記非ゼロ値のそれぞれを受けることと、上記スパースシャードの上記複数の乗算回路が、それぞれのベクトル値に上記シャード入力行列の上記非ゼロ値のそれぞれを乗算した１つ以上の積を生成することと、上記スパースシャードが、上記スパースシャードへの出力として、上記１つ以上の積を用いて、上記シャード入力ベクトルを上記シャード入力行列に適用した積であるシャード出力ベクトルを生成することとを含む。
【０００７】
上述のおよび他の局面の各々は、任意に以下の特徴のうちの１つ以上を単独でまたは組み合わせて含み得る。一実現例は、以下の特徴のすべてを組み合わせて含み得る。
【０００８】

上記シャード出力ベクトルの長さは１よりも大きい。
上記スパースシャードは複数のスパースシャードのうちの１つであり、上記複数のスパースシャードは、システム入力行列の部分行列である複数のシャード入力行列を受け、システム入力ベクトルの部分ベクトルである複数のシャード入力ベクトルを受け、上記複数のスパースシャードによって、上記システム入力ベクトルを上記システム入力行列に適用した積を表すシステム出力ベクトルを生成するように構成されている。
【０００９】
上記複数のスパースシャードはシストリックアレイとして配列され、上記シストリックアレイは、上記シストリックアレイの列次元に沿ったスパースシャードの１つ以上のグループを含み、上記システム出力ベクトルを生成するために、上記１つ以上のプロセッサはさらに、上記シストリックアレイの上記列次元に沿ったグループごとに、上記グループ内の各スパースシャードのそれぞれのシャード出力ベクトルを合算してそれぞれの列出力ベクトルを生成し、各グループの上記それぞれの列出力ベクトルを連結して上記システム出力ベクトルを生成するように構成されている。
【００１０】
各乗算回路は、上記スパースシャードについての上記シャード入力行列のそれぞれからの上記それぞれの非ゼロ値を含むそれぞれのレジスタに結合される。
【００１１】
上記複数の乗算回路における乗算回路の数は、上記予め定められた最大非ゼロ閾値と等しい。
【００１２】
上記スパースシャードはクロスバー回路をさらに含み、上記スパースシャードはさらに、上記クロスバー回路によって、上記シャード入力ベクトルの上記複数のベクトル値を受け、上記複数の乗算回路の各々への入力として、上記クロスバー回路によって、上記複数のベクトル値のうちの１つのベクトル値を送るように構成されている。
【００１３】
上記スパースシャードはさらに、上記シャード入力行列の同じ列の非ゼロ値を、上記複数の乗算回路のうちの隣接する乗算回路のレジスタにロードするように構成されている。
【００１４】
上記スパースシャードはさらに、上記シャード入力行列の各列に沿った非ゼロ値の位置を少なくとも指定する１つ以上の制御値を受けるように構成されており、上記スパースシャードの上記クロスバー回路はさらに、上記１つ以上の制御値を受け、上記１つ以上の制御値に従って、上記シャード入力行列の同じ列に沿った非ゼロ値が乗算されるベクトル値を、隣接する乗算回路に送るように構成されている。
【００１５】
上記スパースシャードは複数の加算回路をさらに含み、上記スパースシャードは１つ以上のセグメントマーカをさらに含み、各セグメントマーカは、上記セグメントマーカにロードされたそれぞれの制御値の値に基づいて、上記複数の加算回路のうちのそれぞれの加算回路への入力をゲート開閉するように構成されており、上記スパースシャードはさらに、上記１つ以上の制御値の少なくとも一部を上記１つ以上のセグメントマーカにロードするように構成されており、上記シャード入力行列の１列目の非ゼロ値についての加算回路は、上記１列目とは異なる上記シャード入力行列の２列目の非ゼロ値を含む隣接する加算回路の入力を受けないようにゲート開閉されており、上記スパースシャードはさらに、上記複数の加算回路によって、上記１つ以上の積の１つ以上の合計を生成するように構成されており、上記１つ以上の合計の各々は、上記シャード入力行列の列の１つ以上の非ゼロ値に上記シャード入力ベクトルの１つ以上のそれぞれの値を乗算したそれぞれのセグメント化合計である。
【００１６】
上記複数の加算回路は並列セグメント化合計回路を形成し、１つ以上の上記セグメント化合計の各々は、セグメントマーカによってゲート開閉されていない隣接する加算回路へ

の出力の合計である。
【００１７】
上記クロスバー回路は第１のクロスバー回路であり、上記スパースシャードは第２のクロスバー回路をさらに含み、上記第２のクロスバー回路は、１つ以上の上記セグメント化合計を受け、上記１つ以上の制御値に従って１つ以上の上記セグメント化合計を配列して、上記スパースシャードついて上記シャード出力ベクトルのそれぞれを生成するように構成されている。
【００１８】
上記第２のクロスバー回路はベネシュネットワークを形成してもよく、上記シャード入力行列は正方行列である。
【００１９】
本開示の別の局面は、１つ以上のプロセッサと、上記１つ以上のプロセッサによって実行されると上記１つ以上のプロセッサに動作を実行させる命令を格納した１つ以上のメモリデバイスとを含むシステムを提供し、上記動作は、ゼロ値および非ゼロ値を含む入力行列を受けることと、上記入力行列を複数の部分行列に区分することとを含み、各部分行列の非ゼロ値の数は予め定められた最大非ゼロ閾値以下であり、各部分行列の次元は予め定められた次元閾値以下である。
【００２０】
上記動作はさらに、部分行列ごとに、上記部分行列の各列に沿った非ゼロ値の位置を指定する１つ以上のそれぞれの制御値を生成することを含み得る。
【００２１】
上記動作はさらに、各部分行列および各部分行列についての上記１つ以上のそれぞれの制御値を、各部分行列および上記部分行列についてのそれぞれの制御値を処理するように構成された複数のスパースシャードに送ることを含み得る。上記複数のスパースシャードの各々は、部分行列と、上記部分行列についての上記１つ以上のそれぞれの制御値と、入力ベクトルの少なくとも一部とを受け、上記部分行列と上記入力ベクトルの上記一部との積を表すそれぞれの出力シャードベクトルを生成するように構成されている。上記入力行列を区分することは、上記行列を、上記複数のスパースシャードにおけるスパースシャードの数と等しい数の部分行列に区分することを含む。
【００２２】
本開示の他の局面は、対応するシステム、装置、および１つ以上の非一時的なコンピュータ読取可能記憶媒体に格納されたコンピュータプログラムを含む。
【図面の簡単な説明】
【００２３】
【図１】本開示の局面に係る、スパースシャードのアレイを含むシステムの一例のブロック図である。
【図２】本開示の局面に係る、スパースシャードの一例のブロック図である。
【図３Ａ】スパースシャードのシャード入力行列の一例を示す図である。
【図３Ｂ】シャード入力行列の一例の非ゼロ値のベクトルを示す図である。
【図３Ｃ】シャード入力行列の一例についての制御値のベクトルを示す図である。
【図４】シャード入力行列と、シャード入力ベクトルと、シャード入力行列についての制御値とを受けるスパースシャードによる行列乗算の一例を示す図である。
【図５】本開示の局面に係る、スパースシャード上のスパース行列の部分行列にシステム入力ベクトルを乗算するためのプロセスの一例のフロー図である。
【図６】複数のスパースシャードからのシステム入力ベクトルとシステム入力行列との積を表すシステム出力ベクトルを生成するプロセスの一例のフロー図である。
【図７Ａ】システム入力行列の一例およびシステム入力ベクトルを示す図である。
【図７Ｂ】システム入力行列およびシステム入力ベクトルの区分を示す図である。
【図７Ｃ】区分されたシステム入力行列、およびシステム入力行列とシステム入力ベクトルとを乗算した積を表すシステム出力ベクトルを示す図である。

【図８Ａ】本開示の局面に係る、シャード入力行列の１つ以上の制御値を用いてスパースシャードを構成するためのプロセスの一例のフロー図である。
【図８Ｂ】図８Ａのプロセスに従って構成されたスパースシャードを用いて行列－ベクトル乗算を実行するためのプロセスの一例のフロー図である。
【図９】本開示の局面に係る、システム入力行列から部分行列を生成するためのプロセスの一例のフロー図である。
【図１０】本開示の局面に係る、スパース行列乗算システムを実現するコンピューティング環境の一例のブロック図である。
【発明を実施するための形態】
【００２４】
詳細な説明
概要
本開示の局面は、スパース行列密ベクトル乗算のために構成された１つ以上の集積回路を含むシステムに関する。複数のスパースシャード（sparse shard）のシステムは、スパースシャードごとに、入力システム行列およびベクトルの部分行列および部分ベクトルを受けることができる。各スパースシャードは、集積回路の少なくとも一部であってもよく、乗算回路および加算回路などの複数の演算ユニットを実現することができる。各スパースシャードは、最大非ゼロ閾値以下の数の非ゼロ値を有する部分行列を受けるように構成されており、この最大非ゼロ閾値は、システムが、たとえば本明細書に記載されているようにスパースシャードと任意に１つ以上の他のコンポーネントとを含むチップとして実現される場合に、予め定めることができる。
【００２５】
各スパースシャードはさらに、本明細書に記載されている制御値などのメタデータを受けることができ、スパースシャードはこのメタデータを用いて、異なる入力を各乗算回路または加算回路に導くことができ、あるユニットからの出力が別のユニットに渡されるときにゲート開閉することができる。受けたメタデータに従ってスパースシャードを構成することによって、スパースシャードは、予め定められた次元閾値までの任意のサイズの入力部分行列を効率的に処理し、対応する長さのベクトルとして積を入力に出力することができる。
【００２６】
部分行列を受ける一部として、各スパースシャードは、部分行列の各列における非ゼロ値の位置を表す１つ以上の制御値を受けることができる。この１つ以上の制御値を用いて、スパースシャードは、スパースシャードが部分行列と部分ベクトルとの積を表すシャード出力ベクトルを生成するように、部分行列および部分ベクトルの個々の値をどのように乗算、加算、および配列すべきかを調整するように構成され得る。システムはさらに、たとえばシステム入力ベクトルにシステム入力行列を乗算することによって、各シャードのシャード出力ベクトルから、システム入力ベクトルをシステム入力行列に適用した積を表すシステム出力ベクトルを生成するように構成され得る。
【００２７】
また、本開示の局面は、スパースシャードのアレイによる処理のためにスパース行列を前処理するためのシステムに関する。１つ以上のプロセッサのシステムは、入力行列を複数の部分行列に区分するように構成され得、各部分行列の次元は、スパースシャードの最大行列入力サイズを指定する予め定められた次元閾値以下である。区分の一部として、システムは、部分行列のうちのいずれかが、予め定められた最大非ゼロ閾値よりも大きい数の非ゼロ値を含むか否かを識別し、それに応じて、識別した部分行列と同じ行または列に沿って部分行列を再区分することができる。システムは、部分行列の次元が次元閾値内にあり、部分行列の非ゼロ値の数が予め定められた非ゼロ閾値以下であるように、スパースシャードごとに部分行列を生成するまでこのプロセスを繰り返すことができる。
【００２８】
本開示の局面に従って実現されるシステムは、スパース行列とベクトルとの反復乗算を

伴う作業負荷をより効率的に実行することができる。たとえば、本開示の局面に係るシステムオンチップ（ＳｏＣ：system-on-a-chip）を実現するデバイスは、少なくとも、最終的な積に寄与しない行列のゼロ要素の冗長な「ゼロ乗算」計算が省略されるので、従来のアプローチよりも少ない処理サイクルで、スパース行列にベクトルを乗算した積を生成することができる。他のアプローチでは、スパース比が高い大きな行列では性能損失が増大する可能性があるが、本明細書に記載されているように実現されるシステムは、少なくとも、入力行列のスパース比が高くなるにつれて省略される冗長計算の割合が高くなるので、これらの行列を用いてさらに効率的に積を計算することができる。

【００２９】

ニューラルネットワークの実行または訓練などの一部の作業負荷は、行列乗算の実行に大きく依存している。ハードウェアアクセラレータまたはその他のデバイスは、行列乗算などの特定の演算を効率的に実行することができるが、処理条件が制限されていることが多い。たとえば、デバイスは、行列のゼロ値と非ゼロ値との予め定義されたスパース比を要する場合があったり、処理する入力のサイズが厳しく制限されている場合がある。行列乗算は、多くのニューラルネットワーク作業負荷にとってユビキタスなタイプの計算であるが、処理条件が制限されているアクセラレータは、さまざまなスパース比の行列の行列乗算を伴う作業負荷など、アクセラレータがサポートできる作業負荷の種類の数が制限されている。

【００３０】

本開示の局面は、異なるサイズおよびスパース比の行列を柔軟に処理することができるスパース行列乗算のためのシステムを提供する。スパース行列乗算のために構成されたスパースシャードのシステムのスパースシャードは、複数の列にわたって所与の入力部分行列の非ゼロ値の位置を追跡し続けながら、入力行列のゼロ値を破棄することができる。スパースシャードは、予め定められた最大非ゼロ閾値と同数のレジスタおよび乗算回路のみで構成され、入力部分行列の非ゼロ値のみをメモリに格納することができる。その結果、スパースシャードは、フルサイズの行列を格納するのと比較して、より効率的に、より少ないリソースで、入力データを格納して処理することができる。

【００３１】

さらに、スパースシャードは、非ゼロ値と、それらに入力ベクトルの値を乗算した積とを、隣接する乗算回路および加算回路に沿ってそれぞれ配列するように構成され得る。スパースシャードは、これらの隣接する回路の出力を組み合わせることによって入力行列の列ごとにセグメント化合計を効率的に生成し、このセグメント化合計を再配列してシャード出力ベクトルを生成するように構成され得る。スパースシャードは、非ゼロ値の順序を保存して、正確に非ゼロ値を加算してこれに入力ベクトルの対応する値を乗算し、入力部分行列と入力ベクトルとの積としてシャード出力ベクトルを正確に生成することができる。スパースシャードは、入力行列の形状を変更するための前処理演算を必要とせずに、入力行列に対して乗算を実行することができる。

【００３２】

また、このシステムは、複数のスパースシャードによって入力ベクトルと乗算するためにスパース行列を前処理するように構成された１つ以上のプロセッサを含み得る。このシステムは、利用可能なスパースシャードの数に応じて異なる数の部分行列を生成すること、および異なる最大非ゼロ閾値についての部分行列を生成することなど、スパースシャードの異なる構成をサポートすることができる。

【００３３】

本明細書に記載されている本開示の局面は、１つ以上のプロセッサと複数のスパースシャードとを含むシステムによって実現することができる。このシステムは、たとえば、コンピューティングプラットフォームのデータセンター内のサーバコンピューティングデバイスなどのコンピューティングデバイスにインストールされたチップとして実現することができる。

【００３４】

システムの例

図１は、スパースシャード１０１Ａ～Ｐのアレイ１０１を含むシステム１００の一例のブロック図である。アレイ１０１は、行列入力に対して行列乗算を実行するように構成されたスパース行列乗算システム１００などのスパース行列乗算システムの少なくとも一部であってもよい。

【００３５】

図２を参照して本明細書により詳細に記載されているように、スパースシャードは、算術演算を実行するように、算術演算を実行する回路間に入力および出力を配列するように、ならびに／または他の回路間の入力および出力をゲート開閉するように構成された回路の集合体である。たとえば、スパースシャードは、スパース行列の矩形シャードまたは部分と、ベクトルの線形シャードまたは片との間の行列乗算を実行するように構成され得る。本明細書により詳細に記載されているように、スパースシャードは、予め定められた最大閾値までの任意の数の非ゼロ値を有する、スパース行列のさまざまな異なるシャードまたは部を受けることができる。

【００３６】

回路の集合体は、各ゲートの状態（すなわち、開もしくは閉）、計算されるオペランド、ならびに／または各回路への入力および各回路からの出力の配列を制御するように、本明細書に記載されているように構成され得る。この構成は、スパースシャードが入力として受けるように構成されているシャード入力行列の次元に、ならびに／またはシャード入力行列内の非ゼロ値の配列および数に、少なくとも部分的に依存し得る。各スパースシャードは、スパース行列にベクトルを乗算するように構成されているシステムに対する個々の回路コンポーネントであってもよい。

【００３７】

各スパースシャード１０１Ａ～Ｐは、アレイ１０１内の２つ以上の他のスパースシャードと通信するように構成されている。スパースシャードは、そのすぐ隣のスパースシャード、たとえば、スパースシャード１０１Ａ～Ｐの矩形配列によって定義される次元に沿った当該スパースシャードの直前または直後のスパースシャードと通信することができる。スパースシャード間の接続は、たとえば、スパースシャードをその近隣のスパースシャードに物理的に接続するバスまたは１つ以上の回路相互接続を介して実現することができる。各スパースシャード１０１Ａ～Ｐは、行列の少なくとも一部およびベクトルの少なくとも一部を受け、入力行列とベクトルとを乗算した積を表す出力ベクトルを生成するように構成された１つ以上の回路として実現することができる。いくつかの例では、スパースシャード１０１Ａ～Ｐはシストリックアレイとして編成されるが、さまざまな実現例において、アレイ１０１は一般にスパースシャード１０１Ａ～Ｐの矩形配列に従って構成または配列される。

【００３８】

アレイ１０１は、回路基板または他の材料の上に複数のコンポーネントおよび集積回路を実現するシステムオンチップの少なくとも一部であってもよい。アレイ１０１は、コンピューティングデバイスの一部としてインストールされ、メモリ、プロセッサ、ネットワークコンポーネント、および／または周辺機器など、デバイスの他のコンポーネントとやり取りするように構成され得る。たとえば、アレイ１０１は、システム１００の一部として実現された１つ以上のメモリデバイスから、システム入力ベクトル１０５およびシステム入力行列１１０を受けることができる。システム１００は、出力として、システム入力ベクトル１０５とシステム入力行列１１０とを乗算した積を表すシステム出力ベクトル１１５を生成することができる。

【００３９】

いくつかの例では、システム出力ベクトル１１５は、システム１００を実現する他のデバイスまたはデバイスのコンポーネントへの入力として供給され得る。たとえば、システム出力ベクトル１１５が、モデルパラメータ値にニューラルネットワークについてのあるベクトル入力を乗算した積である場合は、システム出力ベクトル１１５は、出力ベクトル

１１５の活性化関数を計算するように構成された１つ以上のプロセッサへの入力として供給され得る。

【００４０】

システム１００は、前処理エンジン１５０からシステム入力ベクトル１０５およびシステム入力行列１１０を受けることができる。前処理エンジン１５０は、システム１００を実現するデバイスであってもなくてもよい１つ以上のコンピューティングデバイス上に実現することができる。システム入力行列１１０は、一例として、ニューラルネットワークについてのモデルパラメータ値の少なくとも一部を表す値を含み得る。システム入力行列１１０は、多次元配列またはテンソルなどのより複雑なデータ構造の一部であってもよい。システム１００は、３次元テンソルまたは行列などのより大きなデータ構造の少なくとも一部に対応する各行列を受け、この行列にシステム入力ベクトル１０５を乗算した対応する出力を生成するように構成され得る。システム入力ベクトル１０５は、たとえば、そのモデルパラメータ値がシステム入力行列１１０の値によって少なくとも部分的に表される訓練済みニューラルネットワークへの入力であってもよい。

【００４１】

前処理エンジン１５０は、システム入力行列１１０を処理して１つ以上の制御値１１１および行列パーティションデータ１１２を生成するように構成され得る。前処理エンジン１５０は、システム入力行列１１０を、スパース行列を格納するためのさまざまな異なるフォーマットで、たとえばその完全な行－列形式（すべての非ゼロ値およびゼロ値を有する）で受けることができる。他の例として、前処理エンジン１５０は、圧縮されたスパース列フォーマット、座標リストフォーマット、またはスパース行列を格納するためのさまざまな他のフォーマットのうちのいずれかに従って、システム入力行列１１０を受けることができる。

【００４２】

前処理エンジン１５０がシステム入力行列１１０をその完全な形で受けるいくつかの例では、前処理エンジン１５０は、システム入力行列１１０を、メモリに格納するのに適していると予め定められたフォーマットに変換するように構成されている。たとえば、前処理エンジン１５０は、行列からゼロ値を除去し、行列内の非ゼロ値の元の位置に対する非ゼロ値の位置を追跡するための制御値を生成してから、この予め定められたフォーマットに変換することができる。

【００４３】

制御値は、図２を参照して本明細書により詳細に記載されているように、システム入力ベクトル１０５の部分ベクトルを受けてこれにシステム入力行列１１０の部分行列を乗算するための各スパースシャードを構成するために使用される。パーティションデータ１１２は、どのようにシステム入力行列１１０を部分行列に区分すべきかを指定するデータである。各部分行列はそれぞれのスパースシャードにおいて入力として受けられ、パーティションデータ１１２は、アレイ１０１内のスパースシャードと同数の部分行列についての区分を指定する。

【００４４】

前処理エンジン１５０は、ベクトルパーティションデータ１０６を生成するように構成され得る。図５Ａ～図５Ｃを参照してより詳細に記載されているように、同じ行または列に沿ったスパースシャードは、同じ部分ベクトルを、それぞれの部分行列と乗算するための入力として受けることができる。

【００４５】

システム入力行列１１０およびシステム入力ベクトル１０５はシステム１００の左側および右側に沿って供給されるものとして示されているが、システム１００に入力データを供給するバスの正確な位置決めおよび方向付けは実現例によって異なり得る。たとえば、システム１００と同じチップ上の他のコンポーネントの位置に基づいて、システム１００に入力を供給してシステム１００から出力を受けるためのバスまたは回路相互接続は、それらの他のコンポーネントの位置を考慮してさまざまに方向付けられ得るかまたは位置決

めされ得る。
【００４６】
図２は、本開示の局面に係る、スパースシャード２００の一例のブロック図である。たとえば、システム１００のスパースシャード１０１Ａ～Ｐの各々は、スパースシャード２００を参照して本明細書に記載されているように実現することができる。
【００４７】
スパースシャード２００は、シャード入力ベクトル２０５およびシャード入力行列２１０を受けるように構成されている。シャード入力ベクトル２０５は、１つ以上のベクトル値を含み、１×Ｒの最大次元を有し得る。シャード入力行列は、１つ以上のゼロ値および１つ以上の非ゼロ値を含み、Ｒ×Ｃの最大次元を有する。Ｒ（行）およびＣ（列）は、予め定められた、スパースシャードが入力として受けることができるベクトル／行列の最大入力サイズに対応する、次元閾値である。異なる実現例では、スパースシャードは異なる次元ＲおよびＣについて構成され得る。ＲおよびＣは互いに等しくても異なっていてもよく、たとえばスパースシャードが処理するように構成されている異なる作業負荷のデータの性質に応じて、異なる次元に異なるスパースシャードを実現することができる。スパースシャードのアレイのスパースシャードは、同じ最大次元閾値内の入力を受けるように構成され得る。
【００４８】
スパースシャード２００は、スパースシャード２００に関連付けられたメモリ内の予め定められたアドレス範囲内のシャード入力ベクトル２０５およびシャード入力行列２１０を受けるように構成され得る。たとえば、スパースシャード２００は、結合されたメモリ内の第１のアドレス範囲からシャード入力ベクトル２０５を自動的に取り出し、メモリ内の同じまたは異なるアドレス範囲からシャード入力行列２１０を取り出すように構成されている。スパースシャード２００を有するシステム１００を実現するデバイスまたはデバイスのコンポーネントは、システムによって実現される１つ以上のスパースシャードの各々に対応するメモリ内の場所にシャード入力行列およびベクトルを送るように構成され得る。たとえば、前処理エンジン１５０は、システム入力行列および／またはシステム入力ベクトルを処理生成した後、個々のシャード入力行列およびシャード入力ベクトルを、スパースシャード２００を含む各スパースシャードに対応するアドレス範囲に格納するように構成され得る。
【００４９】
最大次元閾値に加えて、スパースシャード２００は、予め定められた最大非ゼロ閾値以下の非ゼロカウントを有するシャード入力行列を受けるように構成されている。次元閾値と同様に、最大非ゼロ閾値は、スパースシャード２００およびその対応するアレイの実現例によって異なる値に設定され得る。たとえば、システムが処理するように構成される作業負荷のデータが一般に、スパース比が高い処理行列を含む場合、スパースシャードのシステムは、比較的高い最大非ゼロ閾値で構成され得る。本明細書に記載されているように、スパースシャードの乗算回路および加算回路の数は、その最大非ゼロ閾値に対応し、したがって、最大非ゼロ閾値が相対的に低いスパースシャードは、最大非ゼロ閾値が相対的に高いスパースシャードと比べて、少ない回路で構築することができる。
【００５０】
スパースシャード２００は、クロスバー２１５および乗算回路２２０を含み得る。クロスバー２１５は、最大でＲ個のベクトル値を受け、その値をＮ個の乗算回路２２０に分散させるように構成され、Ｎはスパースシャード２００についての最大非ゼロ閾値と等しい。乗算回路２２０Ａ～Ｃ、Ｎが示されているが、異なる実現例ではスパースシャード２００はより多いまたはより少ない乗算回路を含み得ると理解される。
【００５１】
クロスバー２１５は、入力を受けてその入力を１つ以上の宛先に渡すように構成された任意の１つまたは複数の回路として実現することができ、この宛先は乗算回路２２０などの他の回路であってもよい。乗算回路２２０は、２つのオペランド間のハードウェア乗算

を実行するためのさまざまな異なる技術のうちのいずれかに従って実現することができる。乗算回路の第１のオペランドは、クロスバー２１５が受けたベクトル値であってもよい。乗算回路の第２のオペランドは、シャード入力行列２１０からの非ゼロ値であってもよい。各非ゼロ値は、それぞれの乗算回路２２０Ａ～Ｃ、Ｎのそれぞれのレジスタ２２１Ａ～Ｃ、Ｎにロードされる。最大非ゼロ閾値と同数の乗算回路を有することにより、非ゼロ閾値内のシャード入力行列ごとに利用可能な乗算回路が提供される。各乗算回路は、そのそれぞれのレジスタに格納された非ゼロ値に、クロスバー２１５が受けたベクトル値を乗算する。

【００５２】

また、スパースシャード２００は加算回路２２５を含み得る。各加算回路２２５Ａ～Ｃ、Ｎは、対応する乗算回路から入力を受けるように構成されている。各加算回路の間にセグメントマーカがある。加算回路２２５Ａ～Ｃ、Ｎおよびセグメントマーカ２２６Ａ～Ｃ、Ｎ－１が図２に示されているが、乗算回路２２０と同様に、加算回路およびセグメントマーカの数は実現例によって異なり得ると理解される。

【００５３】

加算回路は、２つのオペランドのハードウェア追加のための任意の技術を用いて実現することができる。加算回路からの第１のオペランドは、乗算回路から受けた積であってもよい。たとえば、乗算回路２２０Ａは、ベクトル値と非ゼロ値とを乗算した積を加算回路２２５Ａに渡す。セグメントマーカは、ゲート入力値に応じて、隣接する加算回路間の入力をゲート開閉するように構成された回路または他のハードウェアコンポーネントである。

【００５４】

制御値２３０は、シャード入力行列２１０およびシャード入力ベクトル２０５とともに入力として受けられ、クロスバー２１５、セグメントマーカ２２６、および／またはクロスバー２３５のうちの１つ以上を構成するために使用され得る。制御値２３０は、シャード入力行列２１０のそれぞれの非ゼロ値にそれぞれ対応する値のシーケンスであってもよい。制御値２３０は、１などの第１のタイプの値を含み得、これらの値は、シャード入力行列のそれぞれの列の第１の非ゼロ値である非ゼロ値に対応し得る。制御値２３０は、０などの第２のタイプの値を含み得、これらの値は、シャード入力行列の同じ列の１つ以上の他の非ゼロ値の後ろに位置する非ゼロ値に対応し得る。また、制御値２３０は、図３Ａ～図３Ｃを参照して本明細書に記載されているように、シャード入力行列２１０およびシャード入力ベクトル２０５を処理するようにスパースシャードを構成するために本明細書に記載されているように使用され得る値の１つ以上のベクトルを含み得る。図４およびその対応する本明細書中の記載は、スパースシャードを用いる行列乗算の一例を示す。

【００５５】

いくつかの実現例では、クロスバー２１５、２３５は、スパースシャード２００の異なる最大次元閾値を利用するように実現することができる。たとえば、次元ＲおよびＣが等しいまたはほぼ等しい場合は、クロスバー２３５は、たとえばベネシュネットワークとして、正方形またはほぼ正方形の入力に対するクロスバー再配列のための任意の技術に従って実現することができる。

【００５６】

図３Ａ～図３Ｃは、シャード入力行列３００Ａ、シャード入力行列内の非ゼロ値のベクトル３００Ｂ、およびシャード入力行列３００Ａに対応する制御値のベクトル３００Ｃの一例を示す。また、図３Ｃは追加の制御ベクトル３０５Ｃおよび３１０Ｃを示している。

【００５７】

図３Ａは、スパースシャードのシャード入力行列３００Ａの一例を示す図である。分かりやすくするために、非ゼロ値は影付きセルとして示されている。分かりやすくするために、シャード入力行列３００Ａの列および行に沿って、かつベクトル３００ＢおよびＣに沿って、インデックスが与えられている。たとえば、行列３００Ａにおいて、行２、列４（２，４）の値は１である。

【００５８】

図３Ｂは、シャード入力行列３００Ａの一例の非ゼロ値のベクトル３００Ｂを示す図である。ベクトル３００Ｂの非ゼロ値は、シャード入力行列３００Ａの非ゼロ値を左から右に読み取ったときの出現順序に対応するが、正確な読み取り順は実現例ごとに異なり、たとえば右から左であってもよい。

【００５９】

図３Ｃは、シャード入力行列３００Ａの一例についての制御値のベクトル３００Ｃ、３０５Ｃ、および３１０Ｃを示す図である。いくつかの例では、ベクトル３００Ｃ、３０５Ｃ、および３１０Ｃは、予め定められた順序に従って同じベクトルの一部であってもよく、スパースシャードは、この予め定められた順序を用いて、本明細書に記載されているように、制御値のベクトルを受け、ベクトル３００Ｃ、３０５Ｃ、および３１０Ｃに従ってスパースシャードの異なる成分を構成するように構成され得る。

【００６０】

ベクトル３００Ｃは、スパースシャードのセグメントマーカを構成するために使用される制御値に対応する。ベクトル３００Ｃにおける１の値の制御値（この例ではビット）は、行列３００Ａの新しい列の開始に対応する。インデックス０の値は、制御値のベクトル３００Ｃの開始として、自動的に１に設定され得る。いくつかの実現例では、開始制御値は、処理スパースシャードによって省略されて定数と仮定され得る。ハードウェア実現例は、この値が１であることが既知であるという事実を利用することによって、その回路を簡素化することができる。ベクトル３００Ｃにおけるインデックス１の値も、行列３００Ａの次の列の最初の非ゼロ値であるベクトル３００Ｂにおけるインデックス１の値に対応するように、１に設定される。ベクトル３００Ｃにおけるインデックス２の値は、次の列の最初の非ゼロ値ではないベクトル３００Ｂにおける非ゼロ値に対応するので、０に設定される。

【００６１】

別の例として、ベクトル３００Ｃにおけるインデックス３のビットは、対応する非ゼロ値（ベクトル３００Ｂのインデックス３の値１）が行列３００Ａの次の列（具体的には列２）の最初の非ゼロ値であるので、１に設定される。ベクトル３００Ｃにおけるシーケンスは、すべての列のすべての非ゼロ値が表されるまで、この説明したパターンに従う。

【００６２】

ベクトル３０５Ｃは、スパースシャードの入力クロスバーを構成するための制御値に対応する。ベクトル３０５Ｃは、シャード入力行列３００Ａの非ゼロ値ごとに、部分行列３００Ａ内の非ゼロ値の「ｙ」座標を指定する。この例では、「ｙ」次元はスパース行列３００Ａを垂直方向に上下するが、他の例では「ｙ」次元はたとえば水平に定義されるなど、異なって定義され得る。たとえば、ベクトル３０５Ｃにおける要素ゼロの値「３」は、行列３００Ａの１列目の非ゼロ値「１」の「ｙ」座標に対応する。別の例として、ベクトル３０５Ｃにおける要素６の値「４」は、行列３００Ａの４列目の一番下の値「１」に対応する。

【００６３】

ベクトル３１０Ｃは、スパースシャードの加算回路によって生成された合計が、シャード出力ベクトルを生成するために出力クロスバーによってどのように配列されるかを構成するための制御値に対応する。

【００６４】

行列乗算の数学的定義により、スパース行列の位置（ｘ，ｙ）における各値に、位置ｙにおける入力ベクトルの値が乗算される。この乗算の結果は位置ｘにおける出力に加算される。入力クロスバーは、ベクトル３０５Ｃを用いて、同じ列の非ゼロ値をスパースシャード内の隣接する乗算回路に配列する。出力クロスバーは、ベクトル３１０Ｃを用いて、スパースシャードによってシャード入力行列とシャード入力ベクトルとを乗算した積を表すシャード出力ベクトルにおいて、計算した合計を正しい順序で配列する。

【００６５】

図２に戻って、クロスバー２１５は、制御値２３０を受け、シャード入力ベクトルにシ

ャード入力行列が乗算されると、対応するシャード入力行列の列に一致する１つ以上の乗算回路への入力として受けられるようにシャード入力ベクトル２０５についての各値を配列するように構成され得る。

【００６６】

クロスバー２３５は、加算回路から１つ以上の合計を受け、受けた合計を再配列して、入力シャード行列と入力シャードベクトルとの乗算に対応する正しい出力シャードベクトルを得るように構成され得る。クロスバー２１５と同様に、クロスバー２３５を１つ以上の回路として実現するためのさまざまな異なる技術のうちのいずれかを適用することができる。

【００６７】

本明細書に記載されているように、セグメントマーカは、セグメントマーカのゲート入力値に応じて、隣接する加算回路間の入力をゲート開閉するように構成されている。たとえば、セグメントマーカが１の値の制御値を受けた場合は、セグメントマーカは、セグメントマーカに隣接する第１の加算回路からの出力が、セグメントマーカに隣接する第２の加算回路への入力として渡されることを防止することができる。セグメントマーカが０の値の制御値を受けた場合は、セグメントマーカは、第１の加算回路からの出力を第２の加算回路に（または、実現例によってはその逆に）渡す。このセグメントマーカの構成は、異なる列の非ゼロ値の合計とは別に、同じ列の非ゼロ値に対応する合計のみを加算することに相当する。図２を参照して本明細書に記載されているように、この１つ以上の合計は、クロスバー２３５に渡され、シャード出力ベクトル２４０を生成するように再配列され得る。

【００６８】

クロスバー２３５は、入力を破棄するか受けるかを判断し、その入力をシャード出力ベクトル２４０内のその正しい位置に一致するように再配列するように構成され得る。図２に示されるように、各加算回路はクロスバー２３５に出力を渡すことができる（クロスバー２３５への矢印で示す）。ある加算回路の後ろの次のセグメントマーカがゲート開閉されていない場合は、隣接する加算回路間のランニングサムが終了していないので、クロスバー２３５はその加算回路への出力を破棄することができる。次のセグメントマーカがゲート開閉されている場合（または、最後の加算回路２２５Ｎの場合はセグメントマーカがない場合）は、その列についてのランニングサムは終了しており、クロスバー２３５は、出力シャードベクトル２４０の一部になるべき入力としてランニングサムを受ける。どの合計を無視し、どの合計をシャード出力ベクトル２４０の一部として含めるかを把握することによって、クロスバー２３５は、最大次元Ｃまで、列和を正確に追跡してさまざまな長さの出力ベクトルを生成することができる。

【００６９】

図４は、シャード入力行列４１０と、シャード入力ベクトル４０５と、シャード入力行列４１０についての制御値４３０とを受けるスパースシャード４００の計算の一例を示す。

【００７０】

シャード入力ベクトル４０５およびシャード入力行列４１０の値の一例について考える。

【００７１】

［１　３　２］（ベクトル４０５）

［０　３　０　２　０　４　１　０　０］（行列４１０）

この例における行列４１０についての対応する制御ベクトル４３０～４３２は、

［１　０　１　１］（制御ベクトル４３０）

［１　２　０　１］（制御ベクトル４３１）

［０　０　１　２］（制御ベクトル４３２）

である。

【００７２】

説明し易くするために、乗算回路４０Ａ～Ｄを乗算器Ａ～Ｄと略し、セグメントマーカ４３Ａ～ＣをセグメントマーカＡ～Ｃと略し、加算回路４２Ａ～Ｄを加算器Ａ～Ｄと略す

ることにする。

【００７３】

行列４１０の非ゼロ値に基づいて、乗算器Ａには値２がロードされ、乗算器Ｂには値１がロードされ、乗算器Ｃには値３がロードされ、乗算器Ｄには値４がロードされる。なお、この例では、スパースシャードは４つの乗算器Ａ～Ｄおよび４つの加算器Ａ～Ｄのみを含む。

【００７４】

クロスバー４１５は、ベクトル値１、３、および２を有するベクトル４０５を受ける。シャード入力ベクトル４０５からシャード出力ベクトル４４０へのデータの経路を示すために、破線４５Ａ、実線４５Ｂ、および点線４５Ｃが示されている。クロスバー４１５は制御ベクトル４３１を受け、制御ベクトル４３１の各値はシャード入力行列４１０のそれぞれの非ゼロ値の「ｙ」座標に対応する。制御ベクトル４３１の値は、１、２、０、および１を含む。なお、スパース入力行列４１０は次元３×３を有するので、「ｙ」座標の値は０から２の範囲である。制御ベクトル４３１は、乗算器Ａ～Ｄの各々について、シャード入力ベクトル４０５の値のうちのどの値をどの乗算器に送るべきかを指定する。

【００７５】

たとえば、制御ベクトル４３１の最初の値は１であり、シャード入力行列４１０の最初の非ゼロ値の「ｙ」座標に対応する。「ｙ」座標１は（ゼロの後の）２番目の座標であるので、クロスバー４１５は、シャード入力ベクトル４０５の２番目の値を第１の乗算器（ここでは乗算器Ａ）に導く。制御ベクトル４３１の２番目の値は２であり、シャード入力行列４１０の２番目の非ゼロ値の「ｙ」座標に対応する。クロスバー４０５は、次に、シャード入力ベクトル４０５の３番目の値を乗算器Ｂに導くように構成され得る。別の例として、制御ベクトル４３１の３番目の値は０であり、０の「ｙ」座標を有する次の非ゼロ値に対応する。クロスバー４１５は、シャード入力ベクトル４０５の最初の値を乗算器Ｃに導く。

【００７６】

行列４１０の１列目について、クロスバー４１５は値３を乗算器Ａに導き、値２を乗算器Ｂに導く。行列４１０の２列目について、クロスバー４１５は値１を乗算器Ｃに導く。行列４１０の最後の３列目について、クロスバー４１５は値３を乗算器Ｄに導く。乗算器Ａ～Ｄの積は、乗算器Ａは６（３×２）、乗算器Ｂは２（２×１）、乗算器Ｃは３（１×３）、乗算器Ｄは１２（３×４）である。

【００７７】

次に、乗算器Ａ～Ｄが計算した積を加算器Ａ～Ｄが受ける。加算器Ａは、乗算器Ａからの積、つまり６を受ける。最初の制御値（１）は破棄される。加算器Ａの前には加算器がないので、加算器Ａはセグメントマーカ Ａに合計を渡す。制御値４３０の２番目の値はゼロであるので、セグメントマーカ Ａはゲート開閉されていない。加算器Ｂは、加算器Ａから現在の合計（６）を受け、それを乗算器Ｂの積（２）に加算する。セグメントマーカ Ｂはゲート開閉されているので、加算器Ｂの出力（８）はクロスバー４２０に渡される（破線４５Ａで示す）。セグメントマーカ Ｂは加算器Ｂからの出力をゲート開閉するので、加算器Ｃは乗算器Ｃの積（３）にゼロを加算する。セグメントマーカ Ｃはゲート開閉されているので、加算器Ｃの出力（３）はクロスバー４２０に渡される（点線４５Ｃで示す）。最後に、加算器Ｄは、乗算器Ｄの積（１２）を受け、これはスパースシャード内の最後の加算器であるので、その出力（１２）をクロスバー４２０に自動的に渡す（実線４５Ｂで示す）。

【００７８】

クロスバー４２０は、受けた合計８、３、および１２を、出力シャードベクトル４４０を出力するための正しい順序に従って再配列する。クロスバー４２０は、値０、０、２、および１を有する制御ベクトル４３２を受ける。図３Ａ～図３Ｃを参照して本明細書に記載されているように、出力クロスバー４２０についての制御ベクトルの値は、非ゼロ値の「ｘ」座標位置に対応する。「ｙ」座標と同様に、この値はこの例では０から２の範囲で

ある。制御ベクトル４３２の最初の２つの値は０である。したがって、クロスバー４２０は、受けた第１の合計を出力シャードベクトル４４０の第１の要素に導く。クロスバー４３２の０の次の値は２である。クロスバー４３２は、加算器Ａ～Ｄから受けた第２の合計を出力ベクトル４４０の第２の要素に導き（線４５Ｃで示す）、第３の合計を要素に導く（線４５Ｂで示す）ように構成されている。いくつかの例では、クロスバー４２０は、たとえば最初の２つがゼロであるベクトル４３２に示されるような、ベクトル４３２の連続する重複制御値をスキップするように構成されている。いくつかの例では、連続する重複制御値をスキップするのではなく、クロスバー４２０は、受けた入力合計に対して包含的ＯＲ演算を実行し、この包含的ＯＲ演算の結果を、連続する重複制御値に対応する出力ベクトル４４０の位置に出力するように構成されている。

【００７９】

たとえば、ライン４６は、出力クロスバー４２０への加算器Ａの潜在的な入力ソースを示す。セグメントマーカ４３Ａはゼロの値を有するので、クロスバー４２０への加算器Ａの出力は抑制され、たとえばマスクされるかまたはゼロに設定される。代わりに、加算器Ａの出力はセグメントマーカ４３Ａを通って加算器Ｂに渡される。いくつかの例では、出力クロスバー４２０が制御値４３２を受けると、出力クロスバー４２０は、（ライン４６を通して）受けた第１の合計（ゼロの値を有する）と、（ライン４５Ａを通して）受けた第２の合計（加算器Ｂからの８の値を有する）とに対して包含的ＯＲ演算を実行する。クロスバー４２０は、非ゼロオペランドを出力するために包含的ＯＲ演算を実行するように構成され得る。クロスバー４２０は、受けた合計に対して包含的ＯＲ演算（たとえば、０ＯＲ　８）を実行した後に８を出力し、その結果を出力ベクトル４４０内の最初の位置に渡す。いくつかの例では、出力クロスバー４２０は加算器Ａ～Ｄのうちの少なくともいくつかから個々の出力を受けてそれらを合計することができる。

【００８０】

加算回路は、セグメントを定義する連続した数値範囲を追加するためのさまざまな異なる回路構成のうちのいずれかに従って実現することができ、各セグメントは、対応するスパースシャードによって処理されるシャード入力行列のそれぞれの列内の値に対応する。たとえば、スパースシャード２００または４００の加算回路は、各列の非ゼロ値にシャード入力ベクトルのそれぞれの値を乗算することによって積の順次セグメント化合計を実行するための、（たとえば、スパースシャード２００または４００によって示されるような）１つ以上の順次セグメント化合計回路として実現することができる。いくつかの実現例では、加算回路は、並列セグメント化合計回路として合計ツリーを実行するように構成され得る。個々の加算回路は、本明細書に記載されているように、対応する入力を並列に加算し、その合計を、任意の介在するセグメントマーカのゲート値に従って、出力クロスバーおよび／または隣接する加算回路に渡すように構成され得る。

【００８１】

並列セグメント化合計回路は、特に合計している項の数が多い場合に、順次セグメント化合計回路と比べて回路の待ち時間を短縮することができる。セグメントマーカをゲート開閉するための制御値によって、並列セグメント化合計が可能になる。なぜなら、少なくとも、合計する値の範囲を、異なるセグメント内の値に対応する加算回路間の入力をゲート開閉するとともに同じセグメント内の追加される値に対応する加算回路間の入力を可能にする、セグメントマーカのゲート値に従って追跡することができるからである。

【００８２】

いくつかの実現例では、出力クロスバー４２０は、各加算器によって計算された個々の合計を受け、その合計を加算し、制御ベクトル４３２を用いて、その合計を出力シャードベクトル４４０の対応する要素に導くように構成されている。２つ以上の合計が同じ要素に導かれる場合、たとえば、制御ベクトル４３２がベクトル４３２の最初のゼロのように同じ値の重複を含む場合は、出力クロスバーは、図４の線４５Ａで示すように１つの合計を受けるのではなく、出力シャードベクトル４４０内の同じ要素に導かれるように、受けた各合計を加算するように構成されている。

【００８３】
方法の例
図５は、本開示の局面に係る、スパースシャード上のスパース行列の部分行列にシステム入力ベクトルを乗算するためのプロセス５００の一例のフロー図である。説明し易くするために、スパース行列の部分行列をシャード入力行列と呼ぶ。図９は、本明細書において、入力スパース行列を複数の部分行列に区分するためのプロセスの一例を示す。たとえば、図２のスパースシャード２００などのスパースシャードがプロセス５００を実行する。
【００８４】
ブロック５１０に従って、スパースシャードはシャード入力行列を受ける。シャード入力行列は、予め定められた次元閾値内にあり、予め定められた最大非ゼロ閾値以下の非ゼロ値カウントを有する。
【００８５】
ブロック５２０に従って、スパースシャードは、複数のベクトル値を含むシャード入力ベクトルを受ける。シャード入力ベクトルは、システム入力ベクトルの部分ベクトルであり、これは、図１および図９を参照して本明細書に記載されているように、前処理エンジンによってシステムに対する前処理入力の一部として生成することができる。
【００８６】
ブロック５３０に従って、スパースシャードは、それぞれのベクトル値にそれぞれの非ゼロ値を乗算した１つ以上の積を生成する。図２および４を参照して本明細書に記載されているように、スパースシャードは、シャード入力行列内の非ゼロ値の位置に対応する制御値で構成され得る。この制御値に基づいて、スパースシャードは、部分ベクトルの受信ベクトル値を、入力シャード行列の同じ列に沿った非ゼロ値が格納されている、対応する隣接する乗算回路に導くように構成され得る。本明細書に記載の図８Ａは、シャード入力行列についての制御値を用いてスパースシャードを構成するためのプロセスの一例を示す。
【００８７】
ブロック５４０に従って、スパースシャードは、この１つ以上の積の１つ以上の合計を生成する。図２を参照して本明細書に記載されているように、スパースシャードは、対応する乗算回路から入力を受けるように構成された複数の加算回路を含む。加算回路はさらに、セグメントマーカに隣接する回路をゲート開閉するように設定された当該マーカに達するまで、隣接する加算回路に沿ってこれらの入力を加算するように構成されている。最大でセグメントマーカまでの加算回路入力の合計は、たとえば図２のクロスバー２３５などのクロスバーに渡され得る。
【００８８】
スパースシャードは、この１つ以上の合計から、シャード入力ベクトルにシャード入力行列を乗算した積であるシャード出力ベクトルを生成する。スパースシャードは、乗算回路によって生成された積のオペランドが第１のクロスバーによってどのように順序付けられたかに応じて、たとえば第２のクロスバーを通して、受けた合計を再配列することができる。
【００８９】
図６は、複数のスパースシャードからのシステム入力ベクトルとシステム入力行列との積を表すシステム出力ベクトルを生成するプロセス６００の一例のフロー図である。図１のスパース行列乗算システム１００などのスパースシャードのシステムが、プロセス６００を実行することができる。
【００９０】
ブロック６１０に従って、スパースシャードのアレイの列次元に沿ったスパースシャードのグループごとに、グループ内の各スパースシャードのシャード出力ベクトルを加算してそのグループの列出力ベクトルを生成する。後述する図７Ｃは、それぞれのシャード入力行列に従うスパースシャードのグループ化の一例を示す。スパースシャードのグループが沿って形成される次元は、実現例によって異なり得る。たとえば、システム入力行列およびシステム入力ベクトルがスパースシャードのアレイに供給される方向に応じて、グル

ープは、列とは対照的に、アレイの行に沿うことができる。

【００９１】

ブロック６２０に従って、システムは、各列出力ベクトルを連結してシステム出力ベクトルを生成する。システム出力ベクトルは、システム入力行列にシステム入力ベクトルを乗算した積である。いくつかの実現例では、図９を参照して本明細書により詳細に記載されているように、システムは、行列の列が置換されるシステム入力行列を受けて、たとえば、非ゼロ値の出現をスパースシャードに割り当てられた部分行列により均等に分散させることができる。それらの実現例では、システムは、連結されたシステム出力ベクトルの要素を再配列して元の順列を逆にするように構成され得る。システムは、システム入力行列、制御値、およびシステム入力行列の区分を定義するデータを受ける一部として、並び替えを定義するデータを受けることができる。

【００９２】

図７Ａ～図７Ｃおよび対応する記載は、スパース行列７００の一例とベクトル７５０との間の乗算の一例を示す。説明のために、この乗算は、４×４アレイの１６個のスパースシャードを有するシステムに対して実行されるものとして記載されている。

【００９３】

図７Ａは、システム入力行列７００の一例およびシステム入力ベクトル７５０を示す図である。この図では、システム入力行列７００は整数値で示されているが、要素はたとえば浮動小数点値などの他の値であってもよいと理解される。また、図７Ａ～図７Ｃでは、システム入力行列７００を含むさまざまな行列の非ゼロ値の要素は影付きセルとして示されている。

【００９４】

図７Ｂは、システム入力行列７００およびシステム入力ベクトル７５０の区分を示す図である。この例では、システム入力行列７００は、４×４アレイのスパースシャードごとに１つの部分行列があるように、１６個の部分行列７００Ａ～Ｐに区分されている。システム入力ベクトル７５０は、４×４アレイのスパースシャードの列ごとに１つの部分ベクトルがあるように、４つの部分ベクトル７５０Ａ～Ｄに区分されている。

【００９５】

部分ベクトルおよび部分行列を１６個のスパースシャード（スパースシャードＡ～Ｐと呼ぶ）にマッピングする一例は、以下の表１に示す通りである。

【００９６】

【表１】

スパースシャード	入力部分ベクトル	入力部分行列
A	７５０A	７００A
B	７５０A	７００B
C	７５０A	７００C
D	７５０A	７００D
E	７５０B	７００E
F	７５０B	７００F
G	７５０B	７００G
H	７５０B	７００H
I	７５０C	７００I
J	７５０C	７００J
K	７５０C	７００K
L	７５０C	７００L
M	７５０D	７００M
N	７５０D	７００N
O	７５０D	７００O
P	７５０D	７００P

表１

【００９７】

図７Ｃは、区分されたシステム入力行列７００、およびシステム入力行列とシステム入力ベクトル７５０とを乗算した積を表すシステム出力ベクトル７７０を示す図である。図７Ｃは、列７０５Ａ～Ｄに沿ってグループ化されている、区分された行列を示している。図６を参照して本明細書に記載されているように、システムは、スパースシャードのアレイの列ごとにシャード出力ベクトルを加算し、列出力ベクトルを生成することができる。図７Ｃにおいて、列出力ベクトル７１０Ａ～Ｄは列７０５Ａ～Ｄに対応する。システムは、列出力ベクトル７１０Ａ～Ｄを連結して、システム入力行列にシステム入力ベクトルを乗算した積としてシステム出力ベクトルを生成することができる。

【００９８】

図８Ａ～図８Ｂは、本開示の局面に係る、入力行列の１つ以上の制御値を用いてスパースシャードを構成し、行列乗算を実行するためのプロセス８００Ａ～Ｂの一例のフロー図である。

【００９９】
図８Ａは、本開示の局面に係る、シャード入力行列の１つ以上の制御値を用いてスパースシャードを構成するためのプロセス８００Ａの一例のフロー図である。たとえば図２のスパースシャード２００などのスパースシャードが、プロセス８００Ａを実行することができる。
【０１００】
ブロック８１０に従って、スパースシャードは、シャード入力行列の各列に沿った非ゼロ値の位置を指定する１つ以上の制御値を受ける。
【０１０１】
ブロック８２０に従って、スパースシャードは、シャード入力行列の非ゼロ値を乗算回路のレジスタにロードする。図２を参照して本明細書に記載されているように、スパースシャードは、システムについて予め定められた最大非ゼロ閾値と等しい数の乗算回路を実現することができる。スパースシャードは、左から右などの予め決められた読み取り方向に沿ってシャード入力行列を読み取ったときの出現順序で非ゼロ値をロードすることができる。
【０１０２】
ブロック８３０に従って、スパースシャードは、この１つ以上の制御値をスパースシャードのクロスバーにロードする。図２のクロスバー８１５などの第１のクロスバーは、制御値を受け、行列－ベクトル乗算の一部としてベクトル値が乗算される、対応するシャード入力行列の列に一致する１つ以上の乗算回路への入力として受けられるようにシャード入力ベクトルの各値を配列するように構成され得る。図２のクロスバー２３５などの第２のクロスバーは、加算回路から１つ以上の合計を受け、受けた合計を再配列して、入力シャード行列と入力シャードベクトルとの乗算に対応する正しい出力シャードベクトルを得るように構成され得る。
【０１０３】
ブロック８４０に従って、スパースシャードは、この１つ以上の制御値を、この制御値の値に基づいて加算回路への入力をゲート開閉するように構成された１つ以上のセグメントマーカにロードする。
【０１０４】
図８Ｂは、図８Ａのプロセス８００Ａに従って構成されたスパースシャードを用いて行列－ベクトル乗算を実行するためのプロセスの一例のフロー図である。
【０１０５】
ブロック８５０に従って、スパースシャードは、入力シャード行列の非ゼロ値を受けてロードする。
【０１０６】
ブロック８６０に従って、スパースシャードは、シャード入力ベクトルのベクトル値を受け、これらのベクトル値をシャード入力行列の同じ列に沿った非ゼロ値とともに乗算回路に送る。
【０１０７】
ブロック８７０に従って、スパースシャードは、セグメントマーカによってゲート開閉されていない隣接する加算回路から１つ以上のセグメント化合計を生成する。図２および図８Ａを参照して本明細書に記載されているように、スパースシャードは、セグメントマーカによってゲート開閉されていない加算回路間の合計を集約し、シャード入力行列の異なる列からの計算を表す加算回路をゲート開閉するように制御値を用いてセグメントマーカを構成することができる。スパースシャードがアクティブゲートビットを有するセグメントマーカに達すると、スパースシャードは、セグメント化合計を再配列してシャード出力ベクトルを生成するように構成された第２のクロスバーにセグメント化合計を渡す。
【０１０８】
ブロック８８０に従って、スパースシャードは、この１つ以上のセグメント化合計からシャード出力ベクトルを生成する。図２および図８Ａを参照して本明細書に記載されてい

るように、クロスバーは、スパースシャードの加算回路から１つ以上のセグメント化合計を受けるように、制御値を用いて構成され得る。クロスバーはさらに、セグメント化合計を再配列して、シャード入力行列にシャード入力ベクトルを乗算した積を表す正しいシャード出力ベクトルを生成するように構成され得る。

【０１０９】

図９は、本開示の局面に係る、システム入力行列から部分行列を生成するためのプロセス９００の一例のフロー図である。１つ以上の場所にある１つ以上のプロセッサが、プロセス９００を実行することができる。たとえば、図１の前処理エンジン１５０などの前処理エンジンが、プロセス９００を実行することができる。

【０１１０】

ブロック９１０に従って、前処理エンジンはシステム入力行列を受ける。システム入力行列は、たとえば、図７Ａのシステム入力行列７００で示されるような行列であってもよい。

【０１１１】

ブロック９２０に従って、前処理エンジンは、システム入力行列を複数の候補部分行列に区分する。区分の一部として、前処理エンジンは、予め定められた次元閾値を指定するパラメータと、部分行列を受けるシステムによって実現されるスパースシャードの数を示す１つ以上のパラメータ値とを受けることができる。たとえば、前処理エンジンは、８行×８列の次元閾値内で（４×４アレイのスパースシャードについて）１６個の候補部分行列を生成するように構成され得る。いくつかの例では、前処理エンジンは、たとえば、異なる次元閾値および／またはスパースシャードの構成を有する異なるシステムにわたる前処理入力のために、更新されたパラメータ値を受けることができる。

【０１１２】

いくつかの実現例では、前処理エンジンがシステム入力行列を区分する前に、前処理エンジンはシステム入力行列の列を置換して、非ゼロ値を候補部分行列に均等に分散させる。たとえば、非ゼロ値が、予め定められた許容差を超えて入力行列の一方側よりも他方側に頻繁に出現する場合、前処理エンジンは、非ゼロ値の出現がより広がることにより、区分後にスパースシャードにより均等に分散するように、入力行列の列の順序を変更するように構成され得る。

【０１１３】

前処理エンジンがこの順序付けを実行する場合、前処理エンジンは、順序付けを表すデータを、追加入力として、たとえばスパースシャードのアレイを有するシステムに送られるパーティションデータの一部として、システムに渡す。システムは、出力ベクトルを、システム入力行列の列が置換される前のシステム入力行列とシステム入力ベクトルとを乗算した出力と一致させるために、順序付けに従ってシステム出力ベクトルの要素を並べ替えるように構成され得る。

【０１１４】

システム入力行列の列を置換すると、システム入力行列がスパースシャードのシステムによって処理される全体速度を向上させることができる。たとえば、列を置換することによって、特に、いくつかの例では、一部のスパースシャードがゼロ値のみを有するシャード入力行列を受け得、他のシャードが非ゼロ値のみを有するシャード入力行列、または非ゼロ閾値までの数の非ゼロ値を有するシャード入力行列を受け得る場合に、各スパースシャードをより効率的に使用することができる。

【０１１５】

ブロック９３０に従って、前処理エンジンは、予め定められた非ゼロ閾値よりも大きい非ゼロ値カウントを有する候補部分行列があるか否かを判定する。予め定められた非ゼロ閾値よりも大きい非ゼロ値カウントを有する候補部分行列があると前処理エンジンが判定した場合は、ブロック９４０に従って、前処理エンジンは、候補部分行列と同じ行または列に沿って部分行列を再区分する。

【０１１６】

図７Ｂに示されるように、部分行列は、入力行列におけるそれらの値の位置に基づいて、列および行に沿って編成され得る。たとえば、部分行列７００Ｊが非ゼロ閾値よりも高い非ゼロ値カウントを含むと前処理エンジンが判定した場合は、前処理エンジンは、部分行列７００Ｊの行（部分行列７００Ｉ、７００Ｋ、および７００Ｌを含む）に沿って、ならびに／または部分行列７００Ｊの列（部分行列７００Ｂ、７００Ｆ、および７００Ｎを含む）に沿って、部分行列を再区分することができる。再区分を実行する際、前処理エンジンは、予め定められた次元閾値を使用し、候補部分行列の数が変わらないように再区分を実行する。たとえば、前処理は、候補部分行列を分割し、決定された候補部分行列の行／列に沿って部分行列の行／列を再分散させることができる。

【０１１７】

ブロック９３０および９４０に従って、部分行列を再区分した後、前処理エンジンは、非ゼロ閾値よりも大きい非ゼロ値カウントを有する候補部分行列があるか否かを再び判定する。前処理エンジンは、最大非ゼロ閾値を超える非ゼロ値カウントを有する候補部分行列がないと判定するまで、ブロック９３０および９４０に従って判定および再区分を繰り返し、ブロック９５０に進むことができる。

【０１１８】

ブロック９５０に示されるように、前処理エンジンはシステム入力ベクトルを区分する。システム入力ベクトルの各部分ベクトルは、スパースシャードのアレイを供給するバスがどのように配列されるかに応じて、スパースシャードの行または列の各々に入力される。前処理エンジンは、ベクトル次元に、受けたスパースシャードの行列を乗算できるように、たとえば数学的に有効な行列乗算のための有効な次元を乗算できるように、ベクトルを区分する。

【０１１９】

たとえば図７Ｂ～図７Ｃ、および表１に示されるように、ベクトル７５０は部分ベクトル７５０Ａ～Ｄに区分され、これらの各々が入力として１つ以上のスパースシャードに渡される。また、図７Ｂでは、各部分ベクトル７５０Ａ～Ｄの各次元は、対応する部分行列７００Ａ～Ｐと乗算するための正しい次元を有している。たとえば、部分ベクトル７５０Ｂは１×２（行×列）であり、部分行列７００Ｅ～Ｈの各々は２行を有しているため、部分ベクトル７５０Ｂと部分行列７００Ｅ～Ｈとの間の有効な行列乗算が可能である。

【０１２０】

ブロック９６０に示されるように、前処理エンジンは、候補部分行列ごとに制御値を生成する。図３Ａ～図３Ｃを参照して記載されているように、前処理エンジンは、行列の列ごとに開始非ゼロ値を示す制御値のベクトルを生成することができる。前処理エンジンは、候補部分行列ごとにこの生成を繰り返して、部分行列ごとに対応する制御値を生成する。

【０１２１】

ブロック９７０に従って、前処理エンジンは制御値および候補部分行列を出力する。前処理エンジンは、制御値と、システム入力行列の区分を指定するデータとを、たとえば図１に示されるようなシステム１００に出力することができる。

【０１２２】

<u>コンピューティング環境の例</u>

図１０は、本開示の局面に係る、スパース行列乗算システム１００および前処理エンジン１５０を実現するコンピューティング環境の一例のブロック図である。前処理エンジン１５０は、サーバコンピューティングデバイス１０１５などにおいて、１つ以上の場所に１つ以上のプロセッサを有する１つ以上のデバイス上に実現することができる。ユーザコンピューティングデバイス１０１２およびサーバコンピューティングデバイス１０１５は、ネットワーク１０６０を介して１つ以上のストレージデバイス１０３０に通信可能に結合され得る。ストレージデバイス（複数可）１０３０は、揮発性メモリと不揮発性メモリとの組み合わせであってもよく、コンピューティングデバイス１０１２、１０１５と同じまたは異なる物理的位置にあってもよい。たとえば、ストレージデバイス（複数可）１０３０は、ハードドライブ、ソリッドステートドライブ、テープドライブ、光ストレージ、

メモリカード、ＲＯＭ、ＲＡＭ、ＤＶＤ、ＣＤ－ＲＯＭ、書き込み可能メモリ、および読み取り専用メモリなどの、情報を格納することができる任意の種類の非一時的なコンピュータ読取可能媒体を含み得る。

【０１２３】

サーバコンピューティングデバイス１０１５は、１つ以上のプロセッサ１０１３およびメモリ１０１４を含み得る。メモリ１０１４は、プロセッサ（複数可）１０１３によって実行され得る命令１０２１を含む、プロセッサ（複数可）１０１３がアクセス可能な情報を格納し得る。また、メモリ１０１４は、プロセッサ（複数可）１０１３が取り出す、操作する、または格納することができるデータ１０２３を含み得る。メモリ１０１４は、揮発性メモリおよび不揮発性メモリなどの、プロセッサ（複数可）１０１３がアクセス可能な情報を格納することができる非一時的なコンピュータ読取可能媒体の一種であってもよい。プロセッサ（複数可）１０１３は、１つ以上の中央処理装置（ＣＰＵ）、グラフィックス処理ユニット（ＧＰＵ）、フィールドプログラマブルゲートアレイ（ＦＧＰＡ）、および／またはテンソル処理ユニット（ＴＰＵ）などの特定用途向け集積回路（ＡＳＩＣ）を含み得る。

【０１２４】

サーバコンピューティングデバイス１０１５は、スパース行列乗算システム１００を、たとえばシステムオンチップとして、ハードウェア内に実現することができる。システム１００は、サーバコンピューティングデバイス１０１５にスロットインされたまたはインストールされた物理的チップの一部として実現することができる。システム１００は、サーバコンピューティングデバイス１０１５の他のコンポーネントと通信するように構成されている。

【０１２５】

命令１０２１は１つ以上の命令を含み得、この命令は、プロセッサ（複数可）１０１３によって実行されると、この１つ以上のプロセッサに、この命令によって定義された動作を実行させる。命令１０２１は、プロセッサ（複数可）１０１３によって直接処理するためのオブジェクトコード形式で、または、解釈可能なスクリプトもしくはオンデマンドで解釈されるか事前にコンパイルされる独立したソースコードモジュールの集合体を含む他の形式で、格納され得る。命令１０２１は、本開示の局面と一致するスパースシャード４００を実現するための命令を含み得る。プリプロセッサエンジン１０５は、プロセッサ（複数可）１０１３を用いて、および／またはサーバコンピューティングデバイス１０１５から離れて位置する他のプロセッサを用いて実行され得る。

【０１２６】

データ１０２３は、命令１０２１に従ってプロセッサ（複数可）１０１３によって取り出され、格納され、または修正され得る。データ１０２３は、コンピュータレジスタ内に、複数の異なるフィールドおよびレコードを有するテーブルとしてリレーショナルもしくは非リレーショナルデータベース内に、またはＪＳＯＮ、ＹＡＭＬ、ｐｒｏｔｏ、もしくはＸＭＬ文書として、格納され得る。また、データ１０２３は、２進値、ＡＳＣＩＩまたはユニコードなどであるがこれらに限定されないコンピュータ読取可能形式にフォーマットされ得る。さらに、データ１０２３は、数字、説明文、プロプライエタリコード、ポインタ、他のネットワーク場所を含む他のメモリに格納されたデータの参照などの、関連情報を識別するのに十分な情報、または関連データを計算する機能によって使用される情報を含み得る。

【０１２７】

ユーザコンピューティングデバイス１０１２も、１つ以上のプロセッサ１０１６と、メモリ１０１７と、命令１０１８と、データ１０１９とを有して、サーバコンピューティングデバイス１０１５と同様に構成され得る。また、ユーザコンピューティングデバイス１０１２は、ユーザ出力１０２６およびユーザ入力１０２４を含み得る。ユーザ入力１０２４は、キーボード、マウス、機械的アクチュエータ、ソフトアクチュエータ、タッチスクリーン、マイクロフォン、およびセンサなどの、ユーザから入力を受けるための任意の適

切なメカニズムまたは技術を含み得る。

【０１２８】

サーバコンピューティングデバイス１０１５は、ユーザコンピューティングデバイス１０１２にデータを送信するように構成され得、ユーザコンピューティングデバイス１０１２は、ユーザ出力１０２６の一部として実現されたディスプレイに受信データの少なくとも一部を表示するように構成され得る。また、ユーザ出力１０２６を用いて、ユーザコンピューティングデバイス１０１２とサーバコンピューティングデバイス１０１５との間のインターフェースを表示することができる。ユーザ出力１０２６は、これに代えてまたはこれに加えて、１つ以上のスピーカ、トランスデューサまたは他の音声出力、ユーザコンピューティングデバイス１０１２のプラットフォームユーザに非視覚情報および非可聴情報を提供するハプティックインターフェースまたは他の触覚フィードバックを含み得る。

【０１２９】

図１０は、プロセッサ１０１３、１０１６およびメモリ１０１４、１０１７がコンピューティングデバイス１０１５、１０１２の内部にあると示しているが、プロセッサ１０１３、１０１６およびメモリ１０１４、１０１７を含む本明細書に記載されているコンポーネントは、異なる物理的位置で動作することができ、かつ同じコンピューティングデバイスの内部にない、複数のプロセッサおよびメモリを含み得る。たとえば、命令１０２１、１０１８およびデータ１０２３、１０１９の一部は、取り外し可能なＳＤカードに格納され、他の部分は読み取り専用のコンピュータチップ内に格納されてもよい。命令およびデータの一部またはすべてが、プロセッサ１０１３、１０１６から物理的に離れているがプロセッサ１０１３、１０１６がアクセス可能な場所に格納されてもよい。同様に、プロセッサ１０１３、１０１６は、同時および／または順次動作を実行することができるプロセッサの集合体を含み得る。コンピューティングデバイス１０１５、１０１２の各々は、タイミング情報を提供する１つ以上の内部クロックを含み得て、このタイミング情報は、コンピューティングデバイス１０１５、１０１２によって実行される動作およびプログラムの時間測定に使用され得る。

【０１３０】

サーバコンピューティングデバイス１０１５は、データの処理要求をユーザコンピューティングデバイス１０１２から受けるように構成され得る。たとえば、環境１０００は、プラットフォームサービスを公開するさまざまなユーザインターフェースおよび／またはＡＰＩを介して、さまざまなサービスをユーザに提供するように構成されたコンピューティングプラットフォームの一部であってもよい。サービスを実行する一部として、サーバコンピューティングデバイス１０１５は、システム１００を用いて受信データを処理し得る。たとえば、サービスが機械学習モデルを訓練している場合は、サーバコンピューティングデバイス１０１５は、システム１００を用いて、機械学習モデルを訓練する一部として乗算演算を実行するように構成され得る。

【０１３１】

デバイス１０１２、１０１５は、ネットワーク１０６０を介した直接および間接通信が可能であってもよい。デバイス１０１５、１０１２は、情報を送受信するための開始接続を受け入れ得るリスニングソケットをセットアップしてもよい。ネットワーク１０６０自体が、インターネット、ワールドワイドウェブ、イントラネット、仮想プライベートネットワーク、ワイドエリアネットワーク、ローカルネットワーク、および１つ以上の企業に独自の通信プロトコルを用いるプライベートネットワークなど、さまざまな構成およびプロトコルを含み得る。ネットワーク１０６０はさまざまな近距離および長距離接続をサポートし得る。この近距離および長距離接続は、２．４０２ＧＨｚ～２．４８０ＧＨｚ（一般にブルートゥース（登録商標）規格に関連する）、２．４ＧＨｚおよび５ＧＨｚ（一般にＷｉ－Ｆｉ（登録商標）通信プロトコルに関連する）などの異なる帯域幅にわたって行われてもよく、または無線ブロードバンド通信用のＬＴＥ（登録商標）規格などのさまざまな通信規格で行われてもよい。また、ネットワーク１０６０は、これに加えてまたはこれに代えて、さまざまなタイプのイーサネット（登録商標）接続を介するなどして、デバ

イス１０１２、１０１５間の有線接続をサポートし得る。

【０１３２】

図１０には１つのサーバコンピューティングデバイス１０１５およびユーザコンピューティングデバイス１０１２が示されているが、本開示の局面は、順次もしくは並列処理のためのパラダイムにおいて、または複数の装置の分散ネットワークを介してなど、さまざまな異なる構成および量のコンピューティングデバイスに従って実現することができると理解される。いくつかの実現例では、本開示の局面は、１つのデバイス、およびその任意の組み合わせに対して実行され得る。さらに、前処理エンジンおよびスパース行列乗算システム１００は同じサーバコンピューティングデバイス１０１５上に実現されるものとして示されているが、いくつかの実現例では、前処理エンジン１５０は、サーバコンピューティングデバイス１０１５とは別の、１つ以上のサーバコンピューティングデバイスおよび／またはユーザコンピューティングデバイス１０１２上に実現される。

【０１３３】

本開示の局面は、デジタル回路、コンピュータ読取可能記憶媒体において、１つ以上のコンピュータプログラム、または上述のうちの１つ以上の組み合わせとして実現することができる。コンピュータ読取可能記憶媒体は、たとえば、コンピューティングデバイスによって実行可能であり有形ストレージデバイスに格納される１つ以上の命令として、非一時的であってもよい。

【０１３４】

本明細書において、「～ように構成される」という表現は、コンピュータシステム、ハードウェア、またはコンピュータプログラム、エンジン、もしくはモジュールの一部に関連する異なる文脈で使用されている。システムが１つ以上の動作を実行するように構成されると言う場合、これは、動作時に、システムにこの１つ以上の動作を実行させる適切なソフトウェア、ファームウェア、および／またはハードウェアが、システムにインストールされていることを意味する。あるハードウェアが１つ以上の動作を実行するように構成されると言う場合、これは、動作時に、入力を受け、この入力に従ってこの１つ以上の動作に対応する出力を生成する１つ以上の回路を、ハードウェアが含むことを意味する。コンピュータプログラム、エンジン、またはモジュールが１つ以上の動作を実行するように構成されると言う場合、これは、１つ以上のコンピュータによって実行されるとこの１つ以上のコンピュータにこの１つ以上の動作を実行させる１つ以上のプログラム命令を、コンピュータプログラムが含むことを意味する。

【０１３５】

図面に示されて請求項に記載されている動作は特定の順序で示されているが、これらの動作は示されている順序とは異なる順序で実行可能であること、また、一部の動作は省略可能であり、複数回実行可能であり、ならびに／または他の動作と並列におよび／もしくは同時に実行可能であることが理解される。さらに、異なる動作を実行するように構成された異なるシステムコンポーネントの分離は、コンポーネントの分離を要するものと理解されるべきではない。記載されているコンポーネント、モジュール、プログラム、およびエンジンは、１つのシステムとして統合されてもよく、または複数のシステムの一部であってもよい。

【０１３６】

特に明記しない限り、上述の代替例は互いに排他的ではなく、特有の利点を達成するためにさまざまな組み合わせで実現され得る。上記特徴のこれらのおよび他の変形例および組み合わせを、請求項によって定義される主題から逸脱することなく利用することができるので、実施例の上述の記載は、請求項によって定義される主題を限定するものとして解釈されるべきではなく、例示するものとして解釈されるべきである。加えて、本明細書に記載されている実施例の提供、ならびに、「などの」および「含む」などと表現されている節は、請求項の主題を特定の実施例に限定するものとして解釈されるべきではなく、むしろ、実施例は、多数の可能な実現例のうちの１つのみを示すことを意図している。さらに、さまざまな図における同一の参照番号は同一または同様の要素を識別し得る。

【図面】

【図１】

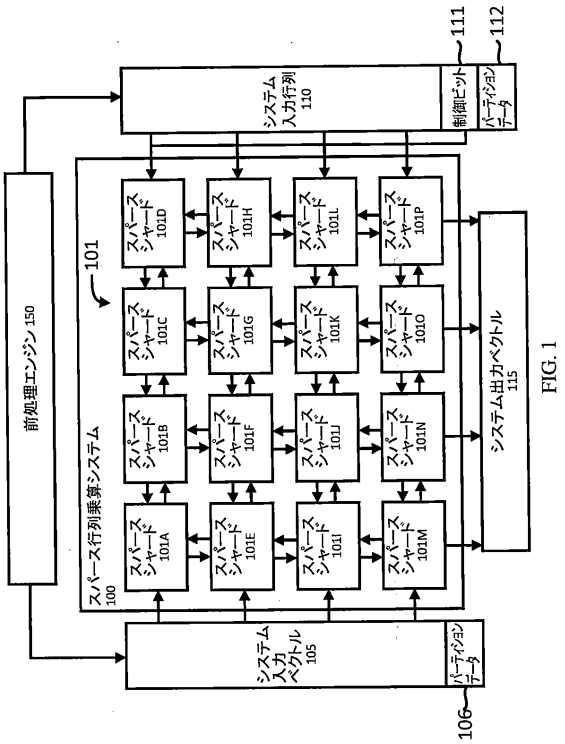

FIG. 1

【図２】

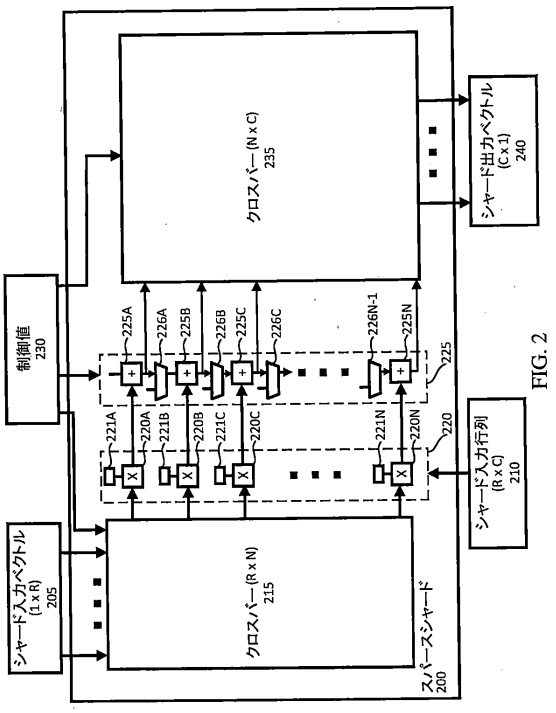

FIG. 2

【図３Ａ】

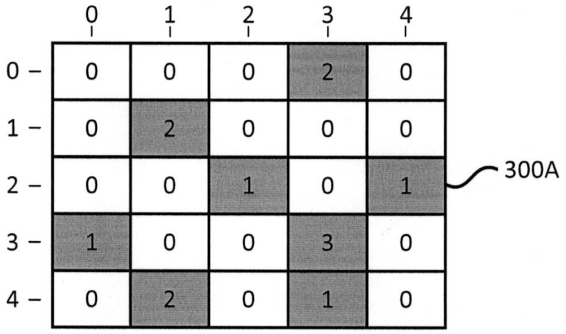

FIG. 3A

【図３Ｂ】

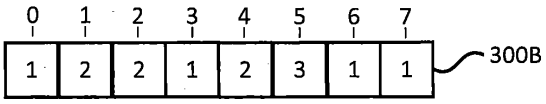

FIG. 3B

【図３Ｃ】

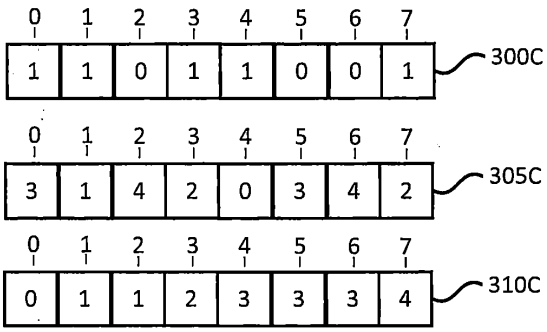

FIG. 3C

【図４】

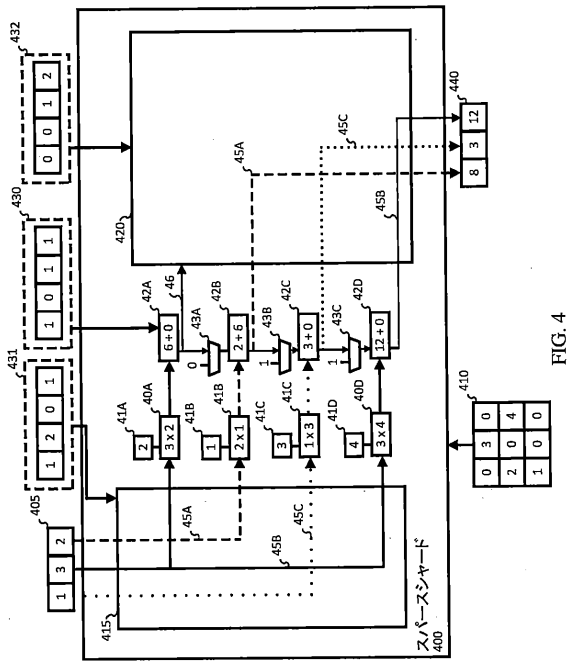

FIG. 4

【図５】

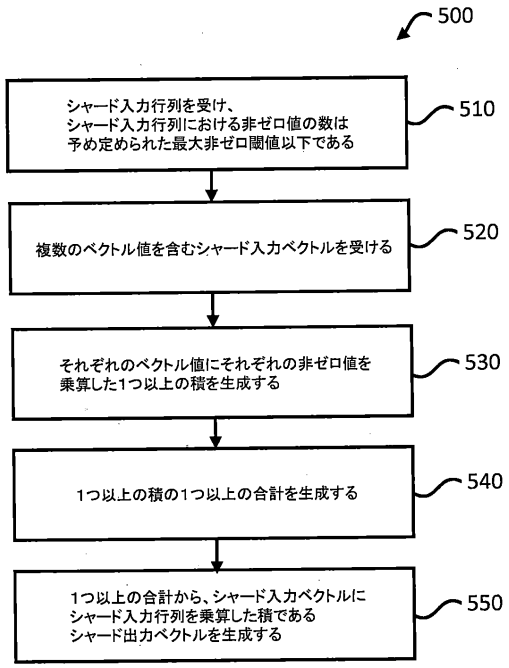

FIG. 5

【図６】

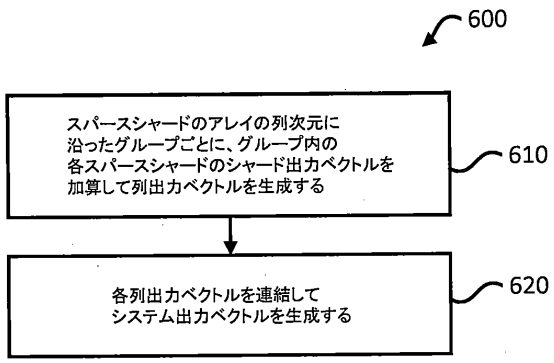

FIG. 6

【図７Ａ】

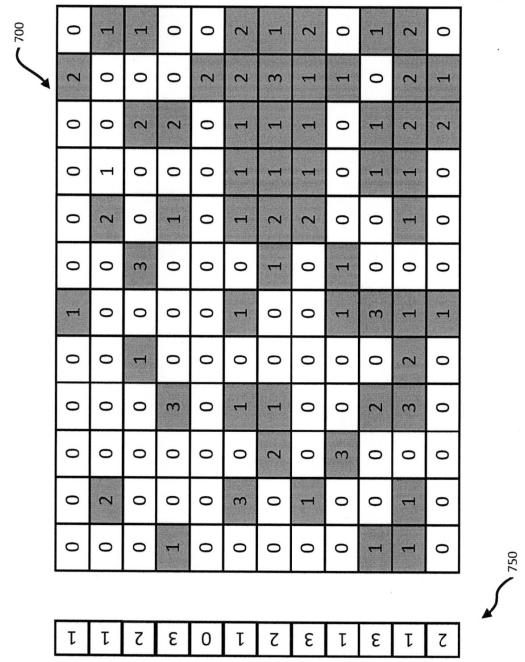

FIG. 7A

【図７Ｂ】

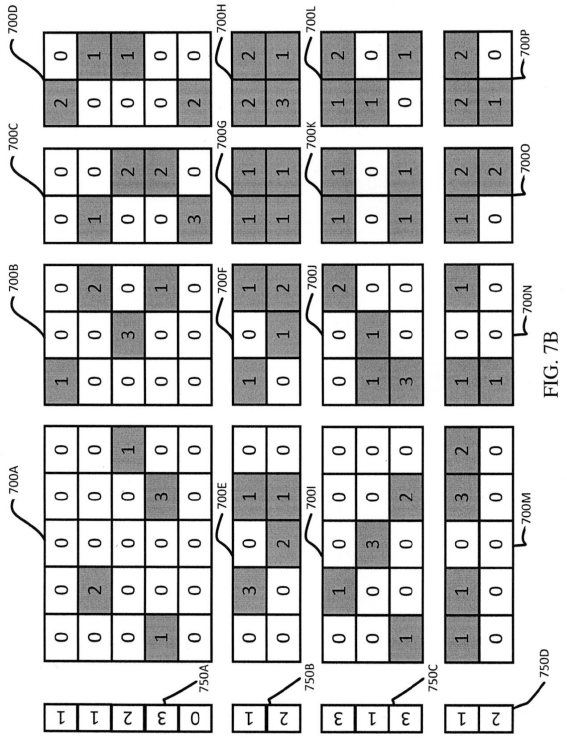

FIG. 7B

【図７Ｃ】

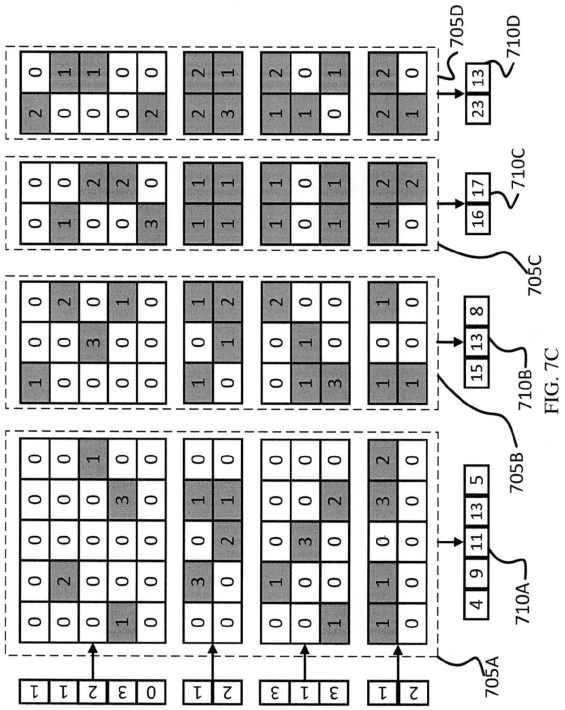

FIG. 7C

【図８Ａ】

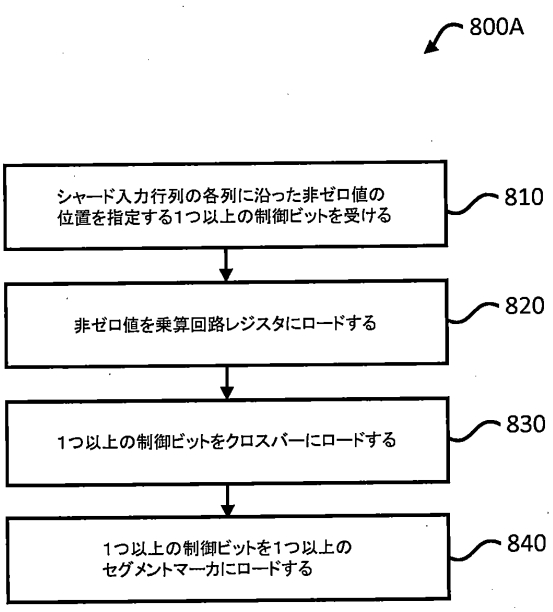

FIG. 8A

【図８Ｂ】

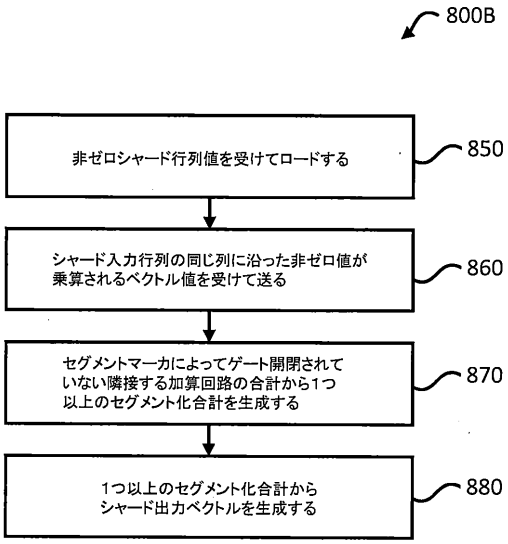

FIG. 8B

【図９】

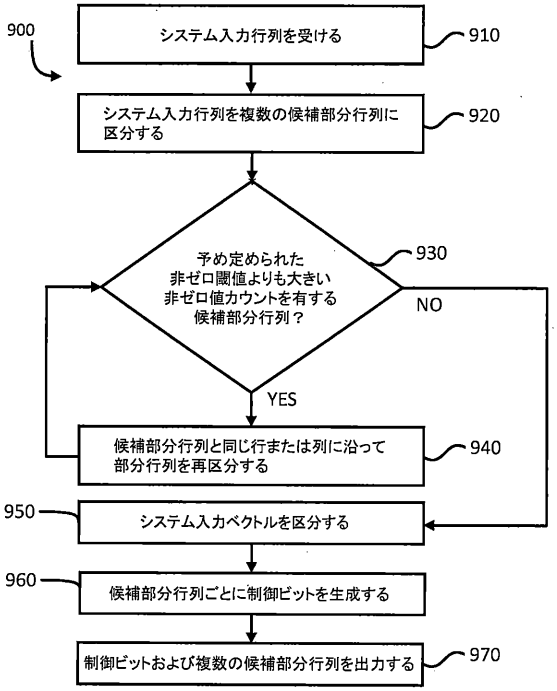

FIG. 9

【図１０】

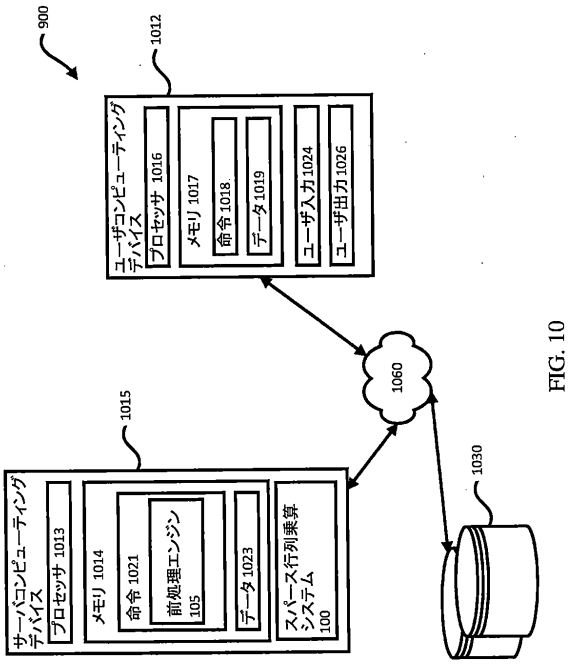

FIG. 10

フロントページの続き

審査官　坂東 博司

(56)参考文献　YAN MINGYU ET AL，HyGCN: A GCN Accelerator with Hybrid Architecture，2020 IEEE INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE COMPUTER ARCHITECTURE (HPCA)，米国，IEEE，2020年02月22日，pages 15-29，［平成5年1月5日検索］，インターネット ＜URL：https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9065592＞
HE XIN ET AL，Sparse-TPU adapting systolic arrays for sparse matrices，PROCEEDINGS OF THE 34TH ACM INTERNATIONAL CONFERENCE ON SUPERCOMPUTING, ACMPUB27，米国，2020年06月29日，pages 1-12，［平成5年1月5日検索］，インターネット ＜URL：https://tnm.engin.umich.edu/wp-content/uploads/sites/353/2020/08/2020.6.sparse-tpu_ics2020.pdf＞
QIN ERIC ET AL，SIGMA: A Sparse and Irregular GEMM Accelerator with Flexible Interconnects for DNN Training，2020 IEEE INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE COMPUTER ARCHITECTURE (HPCA)，米国，IEEE，2020年02月22日，pages 58-70，［online］，［平成5年1月5日検索］，インターネット ＜URL：https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9065523＞

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１７／１６