(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2021-508125
(P2021-508125A)
(43)公表日　令和3年2月25日(2021.2.25)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/16	(2006.01)	G06F	17/16	M	5B056
G06F 15/80	(2006.01)	G06F	15/80		
G06F 7/57	(2006.01)	G06F	7/57		

審査請求 有　予備審査請求 未請求　（全 42 頁）

(21)出願番号　特願2020-536531 (P2020-536531)
(86)(22)出願日　平成30年10月19日 (2018.10.19)
(85)翻訳文提出日　令和2年8月5日 (2020.8.5)
(86)国際出願番号　PCT/CN2018/111077
(87)国際公開番号　WO2019/128404
(87)国際公開日　令和1年7月4日 (2019.7.4)
(31)優先権主張番号　201711499179.X
(32)優先日　平成29年12月29日 (2017.12.29)
(33)優先権主張国・地域又は機関
中国 (CN)

(71)出願人　504161984
ホアウェイ・テクノロジーズ・カンパニー・リミテッド
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング
(74)代理人　110000877
龍華国際特許業務法人

最終頁に続く

(54)【発明の名称】行列乗算器

(57)【要約】
本発明の実施形態は行列乗算器を開示し、データコンピューティング技術の分野に関し、それにより計算のために２つの行列をブロックに分割する。行列乗算器は、第１メモリ、第２メモリ、演算回路、およびコントローラを含み、ここで、演算回路、第１メモリ、および第２メモリはバスを使用してデータ通信を実行してよく、コントローラは、予め設定されたプログラムまたは命令に従って、第１行列および第２行列を、ブロックに分割されるように制御し、演算回路を、コントローラのブロック分割結果に基づいて、第１メモリおよび第２メモリの対応するブロックで乗算演算を実行するように制御するように構成される。行列乗算器は、２つの行列で乗算演算を実行するように構成されてよい。

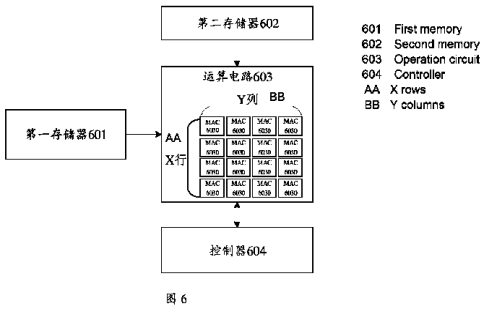


図 6

【特許請求の範囲】
【請求項１】
Ｍ＊Ｋ行列である第１行列を格納するように構成された第１メモリと、
Ｋ＊Ｎ行列である第２行列を格納するように構成された第２メモリと、
前記第１メモリおよび前記第２メモリに接続される演算回路と、
前記演算回路に接続されるコントローラと、を含む、
行列乗算器であって、
前記演算回路は、Ｘ行＊Ｙ列からなる演算ユニットを含み、各前記演算ユニットは、ベクトル乗算回路および加算回路を含み、前記行列乗算回路は、前記第１メモリによって送られる行ベクトルのデータおよび前記第２メモリによって送られる列ベクトルのデータを受信し、前記２つのベクトルを乗算するように構成され、前記加算回路は、前記２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、各演算ユニットの演算結果を取得するように構成され、
前記コントローラは、以下の動作、すなわち、
前記第１行列を、サイズがＸ＊Ｌであるサブブロックを単位とするブロックに分割し、同じサイズのＳ×Ｒ個のサブブロックを取得し、前記Ｓ×Ｒ個のサブブロックのうち第ｓ行第ｒ列におけるサブブロックはA_{sr}，ｓ＝（１，２，３，...，およびＳ）、およびｒ＝（１，２，３，...，およびＲ）で表される、動作と、
前記第２行列を、サイズがＬ＊Ｙであるサブブロックを単位とするブロックに分割し、同じサイズのＲ×Ｔ個のサブブロックを取得し、Ｒ×Ｔ個のサブブロックのうち第ｒ行第ｔ列におけるサブブロックは、B_{rt}，ｒ＝（１，２，３，...，およびＲ）、ｔ＝（１，２，３，...，およびＴ）で表される、動作と
を実行するように構成され、
前記コントローラは、さらに以下の動作、すなわち、
任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルにおける第ｙ列とを、Ｘ行＊Ｙ列からなる演算ユニットの第ｘ行第ｙ列において前記演算ユニットに入力し、それにより、処理を実行する動作を実行するように構成され、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，およびＹ）であり、前記任意のサブブロックA_{sr}におけるｒと、前記対応するサブブロックB_{rt}におけるｒとは同じ値を有する、
行列乗算器。
【請求項２】
前記コントローラは、以下の動作、すなわち、
前記任意のサブブロックA_{sr}の前記Ｘ個の行ベクトルにおける前記第ｘ行と、前記対応するサブブロックB_{rt}の前記Ｙ個の列ベクトルにおける前記第ｙ列とを、同じクロックサイクルにおいて並行して、Ｘ行＊Ｙ列からなる前記演算ユニットの第ｘ行第ｙ列において前記演算ユニットに入力し、それにより前記処理を実行する、動作
を実行するように具体的に構成される、請求項１に記載の行列乗算器。
【請求項３】
前記コントローラはさらに、前記任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる前記演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、前記コントローラはさらに、前記対応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる前記演算ユニットに対応する第ｙ行に連続的に入力するように同時に制御するように構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである、
請求項１または２に記載の行列乗算器。
【請求項４】
前記コントローラはさらに、ｓおよびｒの値を変更されないままにして、ｔの値を、少

なくとも２つの連続するサブブロック乗算計算サイクルにおいて変更されるように制御するように構成され、その結果、前記第１メモリは、前記少なくとも２つの連続するサブブロック乗算計算サイクル内で同じサブブロックA_{sr}を再使用し、前記サブブロック乗算計算サイクルは、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}上での行列乗算演算を完了させるように、Ｘ行＊Ｙ列の前記演算ユニットによって使用される時間である、

請求項１から３のいずれか一項に記載の行列乗算器。

【請求項５】

前記行列乗算器はさらに、前記演算回路に接続された第３メモリを含み、

前記コントローラは、前記ベクトル乗算回路および前記加算回路の演算結果を前記第３メモリに格納するように、Ｘ行＊Ｙ列の前記演算ユニットを制御するように構成される、

請求項１から４のいずれか一項に記載の行列乗算器。

【請求項６】

前記行列乗算器はさらに、前記第１メモリおよび前記第２メモリに接続される第４メモリと、前記第３メモリに接続される第５メモリとを含み、

前記コントローラはさらに、前記第１行列および前記第２行列の乗算演算を実行する前に、

前記第４メモリから、前記第１行列および前記第２行列のデータソースを、それぞれ前記第１メモリおよび前記第２メモリに移動させ、前記第３メモリから、前記計算結果を前記第５メモリに移動させるように、制御するように構成される、

請求項５に記載の行列乗算器。

【請求項７】

前記ベクトル乗算回路はＬ個の乗算器を含み、前記加算回路は入力数がＬ＋１である加算木を含む、

請求項１から６のいずれか一項に記載の行列乗算器。

【請求項８】

前記第１メモリ、前記第２メモリ、前記演算回路、および前記コントローラはバスインタフェースユニットを使用して接続される、

請求項１から７のいずれか一項に記載の行列乗算器。

【請求項９】

Ｓ＝

【数２２】

$$\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

および

Ｒ＝

【数２３】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、

Ｍ％Ｘ≠０のとき、計算は前記第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は前記第１行列の第（Ｋ＋１）行から第（Ｒ＊Ｙ－Ｋ）行まで実行されず、結果の値には０が割り当てられる、

請求項１から８のいずれか一項に記載の行列乗算器。

【請求項１０】

Ｒ＝

【数２４】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

および

Ｔ＝

【数２５】

$$\begin{cases} N/Y, N\%Y = 0 \\ [\frac{N}{Y}] + 1, N\%Y \neq 0 \end{cases}$$

であり、

Ｋ％Ｙ≠０のとき、計算は前記第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられ、Ｎ％Ｘ≠０のとき、計算は前記第１行列の第（Ｎ＋１）行から第（Ｔ＊Ｘ－Ｎ）行まで実行されず、結果の値には０が割り当てられる、

請求項１から８のいずれか一項に記載の行列乗算器。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明はコンピューティング技術の分野に関し、特に行列乗算器に関する。

【背景技術】

【０００２】

現在、２つの行列ＡおよびＢの積は、以下の２つの方式のいずれにおいて計算され得る。

【０００３】

方式１：計算がベクトルプロセッサを使用して実行される。

【０００４】

Ｃ＝Ａ＊Ｂであり、ベクトルプロセッサによって同時に計算できる要素の数はＭであることが前提とされる。図１を参照すると、ベクトルプロセッサは、行列Ａの第ｉ行（要素A_{i1}，A_{i2}，...，$A_{i(M-1)}$，およびA_{iM}を含む）におけるベクトルをソースレジスタＲｅｇ０にロードし、そして、行列Ｂの第ｊ列（要素B_{j1}，B_{j2}，...，$B_{j(M-1)}$，およびB_{jM}を含む）におけるベクトルをレジスタＲｅｇ１にロードし、その結果、Ｒｅｇ０およびＲｅｇ１に対応する要素の間の乗算が実装できる。最終的に、加算木を使用することによって累算演算が完了し、行列Ｃの第ｉ行第ｊ列におけるデータC_{ij}が、計算を通じて取得され、行列Ｃは複数回の計算を実行することによって取得され得る。

【０００５】

方式２：計算速度をさらに増加させるように、行列の乗算演算は、２次元計算アレイを使用して完了され得る。

【０００６】

例えば、２次元計算アレイはＮ＊Ｎシストリックアレイであり得る。方式１において、２個のＮ＊Ｎ行列の乗算演算を完了するには、Ｎ＾３個の乗算演算が必要である。ベクトルプロセッサは、各クロックサイクルにおいてＭ個の要素の間の乗算の計算が可能なので、１つの乗算演算が完了するために必要な期間はＮ＾３／Ｍ個のクロックサイクルである。方式２において、２個のＮ＊Ｎ行列の乗算演算を完了するには、Ｎ＾３個の乗算演算が必要である。シストリックアレイはＮ＾２個の演算ユニットを有するので、１つの行列演算を完了させるために必要な期間は、Ｎ＾３／Ｎ＾２＝Ｎ個のクロックサイクルである。

方式１および方式２の両者において、Ｎ＊Ｎ行列の乗算演算を完了させるためには長い時間がかかり、比較的固定されて柔軟性のないコンピューティングサイズをもたらす。

【発明の概要】

【０００７】

本発明の実施形態は、行列乗算器および関連するデバイスを提供し、それにより、行列乗算の最中の、柔軟性のない計算および低い効率という問題を解決する。

【０００８】

第１の態様によると、本発明の実施形態は行列乗算器を提供し、行列乗算器は、

Ｍ＊Ｋ行列である第１行列を格納するように構成された第１メモリと、

Ｋ＊Ｎ行列である第２行列を格納するように構成された第２メモリと、

第１メモリおよび第２メモリに接続される演算回路と、

演算回路に接続されたコントローラと、を含み、

演算回路はＸ行＊Ｙ列からなる演算ユニットを含み、各演算ユニットはベクトル乗算回路および加算回路を含み、行列乗算回路は、第１メモリによって送られる行ベクトルのデータおよび第２メモリによって送られる列ベクトルのデータを受信し、２つのベクトルを乗算するように構成され、加算回路は、２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、各演算ユニットの演算結果を取得するように構成される、演算回路と、

コントローラは以下の動作、すなわち、

第１行列を、サイズがＸ＊Ｌであるサブブロックを単位とするブロックに分割し、同じサイズのＳ×Ｒ個のサブブロックを取得し、Ｓ×Ｒ個のサブブロックのうち第ｓ行第ｒ列におけるサブブロックはA_{sr}，ｓ＝（１，２，３，...，およびＳ）、およびｒ＝（１，２，３，...，およびＲ）で表される、動作と、

第２行列を、サイズがＬ＊Ｙであるサブブロックを単位とするブロックに分割し、同じサイズのＲ×Ｔ個のサブブロックを取得し、Ｒ×Ｔ個のサブブロックのうち第ｒ行第ｔ列におけるサブブロックは、B_{rt}，ｒ＝（１，２，３，...，およびＲ）、ｔ＝（１，２，３，...，およびＴ）で表される、動作とを実行するように構成され、

コントローラは、さらに以下の動作、すなわち、

任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルにおける第ｙ列とを、Ｘ行＊Ｙ列からなる演算ユニットの第ｘ行第ｙ列において演算ユニットに入力し、それにより、処理を実行する動作を実行するように構成され、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，およびＹ）であり、任意のサブブロックA_{sr}におけるｒと、対応するサブブロックB_{rt}におけるｒとは同じ値を有する。

【０００９】

本発明の実施形態は、行列乗算器を提供し、ここで、行列乗算器は行列乗算ブロック分割方法、すなわち、ＭＮＫフラクタルを完了して、行列乗算器６０における内部コントローラ６０４の制御ロジックを使用することによって、乗算のために大きい行列を単位行列（具体的には、Ｘ＊Ｌ×Ｌ＊Ｙ行列）に分割するようにコントローラを使用する。コントローラ６０４の制御ロジックは、各クロックサイクルにおいて、単位行列乗算タスクを演算回路６０３に送り、その結果、データがパイプライン方式で実行され、Ｘ行＊Ｙ列の演算ユニットがフルロード状態で動作する。行列乗算の効率が増大し、ニューラルネットワークアルゴリズムを大幅に改善する適用効果が実現される。本発明のこの実施形態において提供される行列乗算器は、畳み込みニューラルネットワークにおける畳み込み演算およびＦＣ演算を実行し得る。

【００１０】

可能な実装において、コントローラは、以下の動作を実行するように具体的に構成される。

任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルにおける第ｙ列とを、同じクロックサイクルにおいて並行し

て、Ｘ行＊Ｙ列からなる演算ユニットの第ｘ行第ｙ列において演算ユニットに入力し、それにより演算を実行する。

【００１１】

可能な実装において、コントローラはさらに、任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、コントローラはさらに、対応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｙ行に連続的に入力するように同時に制御するように構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである。

【００１２】

可能な実装において、コントローラはさらに、ｓおよびｒの値を変更されないままにして、ｔの値を少なくとも２つの連続するサブブロック乗算計算サイクルにおいて変更されるように制御するように構成され、その結果、第１メモリは、少なくとも２つの連続するサブブロック乗算計算サイクル内で同じサブブロックA_{sr}を再使用し、サブブロック乗算計算サイクルは、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}上での行列乗算演算を完了させるようにＸ行＊Ｙ列の演算ユニットによって使用された時間である。

【００１３】

可能な実装において、行列乗算器はさらに、演算回路に接続された第３メモリを含み、

コントローラは、ベクトル乗算回路および加算回路の演算結果を第３メモリに格納するように、Ｘ行＊Ｙ列の演算ユニットを制御するように構成される。

【００１４】

可能な実装において、行列乗算器はさらに、第１メモリおよび第２メモリに接続される第４メモリと、第３メモリに接続される第５メモリとを含み、

コントローラはさらに、第１行列および第２行列の乗算演算を実行する前に、

第４メモリから、第１行列および第２行列のデータソースを、それぞれ第１メモリおよび第２メモリに移動させ、第３メモリから、計算結果を第５メモリに移動させるように、制御するように構成される。

【００１５】

可能な実装において、ベクトル乗算回路はＬ個の乗算器を含み、加算回路は入力数がＬ＋１である加算木を含む。

【００１６】

可能な実装において、第１メモリ、第２メモリ、演算回路、およびコントローラはバスインタフェースユニットを使用して接続される。

【００１７】

可能な実装において、　Ｓ＝

【数１】

$$\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

およびＲ＝

【数２】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、

Ｍ％Ｘ≠０のとき、計算は第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は第１行列の第（Ｋ＋１）行から第（Ｒ＊Ｙ－Ｋ）行まで実行されず、結果の値には０が割り当てられる。

【００１８】

可能な実装において、Ｒ＝

【数３】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

およびＴ＝

【数４】

$$\begin{cases} N/Y, N\%Y = 0 \\ [\frac{N}{Y}] + 1, N\%Y \neq 0 \end{cases}$$

であり、

Ｋ％Ｙ≠０のとき、計算は第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられ、Ｎ％Ｘ≠０のとき、計算は第１行列の第（Ｎ＋１）行から第（Ｔ＊Ｘ－Ｎ）行まで実行されず、結果の値には０が割り当てられる。

【００１９】

可能な実装において、行列乗算器はさらに直接メモリアクセスユニットを含み、直接メモリアクセスユニットは、第１行列を第１メモリに移動する前に第１行列に行列転置演算を実行するか、または、第２行列を第２メモリに移動する前に第２行列に行列転置演算を実行するように構成される。

【００２０】

可能な実装において、コントローラは第１行列の任意のサブブロックを、行形式で第１メモリに格納されるように制御し、または、第２行列の任意のサブブロックを、行形式で第２メモリに格納されるように制御する。このようにして、サブブロックが素早く読み出されることができ、サブブロックはフレキシブルに、かつ素早く転置されることができる。

【００２１】

第２の態様によれば、本願は電子デバイスを提供し、電子デバイスは、

第１の態様の任意の実施例において提供されるセキュアな要素と、チップに結合された個別デバイスとを含み得る。

【００２２】

第３の態様によれば、本願はシステムオンチップを提供し、システムオンチップは第１の態様の任意の実施例において提供されたチップを含む。システムオンチップはチップを含み得、または、チップおよび別の個別デバイスを含み得る。

【図面の簡単な説明】

【００２３】

本発明の実施形態または背景における技術的解決方法をより明確に説明するように、以下に、本発明の実施形態または背景を説明するために必要とされる添付の図面を簡潔に説明する。

【００２４】

【図１】先行技術における、２つの行列の積を計算する処理の概略図である。

【００２５】

【図２】先行技術における、畳み込みカーネルを重み行列に変換する概略図である。

【００２６】

【図３】先行技術における、入力データを入力行列に変換する概略図である。

【００２７】
【図４】先行技術における、２つの行列の乗算演算を実行するための方法の概略図である。
【００２８】
【図５】先行技術における、ＴＰＵシストリックアレイの概略図である。
【００２９】
【図６】本発明の実施形態による、行列乗算アクセラレータの構造図である。
【００３０】
【図７】本発明の実施形態による、演算ユニット６０３０の構造図である。
【００３１】
【図８】本発明の実施形態による、行列をブロックに分割する概略図である。
【００３２】
【図９】本発明の実施形態による、特定の演算回路６０３の配線の概略図である。
【００３３】
【図１０】本発明の実施形態による、特定の演算回路６０３の配線の概略図である。
【００３４】
【図１１】本発明の実施形態による、ベースが４である行列乗算器の入力フォーマットを示す図である。
【００３５】
【図１２】Ｍ＝２、Ｎ＝２、およびＫ＝２のときの、Ｔ＝０の時点での行列乗算器のパイプライン実行の概略図である。
【００３６】
【図１３】Ｍ＝２、Ｎ＝２、およびＫ＝２のときの、Ｔ＝１の時点での行列乗算器のパイプライン実行の概略図である。
【００３７】
【図１４】Ｍ＝２、Ｎ＝２、およびＫ＝２のときの、Ｔ＝７の時点での行列乗算器のパイプライン実行の概略図である。
【００３８】
【図１５】Ｍ＝２、Ｂ＝２、およびＫ＝２のときの、Ｔ＝１１の時点での行列乗算器のパイプライン実行の概略図である。
【００３９】
【図１６】本発明の実施形態による、別の行列乗算器の構造図である。
【００４０】
【図１７】本発明の実施形態による、さらに別の行列乗算器の構造図である。
【００４１】
【図１８】本発明の実施形態による、命令非同期実行シーケンスの概略図である。
【発明を実施するための形態】
【００４２】
以下に、本発明の実施形態の添付の図面を参照して、本発明の実施形態を説明する。
【００４３】
本願の本明細書、特許請求の範囲、および添付の図面において、用語「第１」、「第２」、「第３」、「第４」および同様のものは、異なるオブジェクトとの間の区別を意図するものであり、特定の順序を意味するものではない。加えて、用語「を含む」、「を有する」およびそれらの任意の他の変形は、非限定的な含有を含めることを意図するものである。例えば、一連の段階またはユニットを含む処理、方法、システム、製品、およびデバイスは、列挙された段階またはユニットに限定されるものではなく、任意で、列挙されていない段階またはユニットをさらに含むか、任意で、当該処理、方法、製品、またはデバイスにもともと備わった別の段階またはユニットをさらに含む。
【００４４】
本明細書で「実施形態」に言及することは、実施形態を参照して説明される特定の特性

、構造、および特徴が、本願の少なくとも１つの実施形態に含まれ得ることを意味する。本明細書の様々な位置に示される語句は、必ずしも同じ実施形態を参照しなくともよく、独立した、または別の実施形態から排除された追加の実施形態ではない。本明細書において説明される実施形態は、別の実施形態と組み合わせされてよいことが、当業者には明確および暗示的に理解される。

【００４５】

本明細書で使用される「コンポーネント」、「モジュール」および「システム」などの用語は、コンピュータ関連のエンティティ、ハードウェア、ファームウェア、ハードウェアおよびソフトウェアの組み合わせ、ソフトウェア、または実行中のソフトウェアを意味するように使用される。例えば、コンポーネントは、プロセッサ上で動作する処理、プロセッサ、オブジェクト、実行可能なファイル、実行スレッド、プログラム、および／またはコンピュータであってよいが、それらに限定されるものではない。図に示されるように、コンピューティングデバイス上で動作するアプリケーションと、コンピューティングデバイスとの両者が、コンポーネントであってよい。１または複数のコンポーネントが、処理および／または実行スレッド内に存在してよく、コンポーネントは、１つのコンピュータに、および／または２つ以上のコンピュータの間で分散されて配置されてよい。加えて、これらのコンポーネントは、様々なデータ構造を格納する、様々なコンピュータ読み出し可能な媒体から実行されてよい。例えば、コンポーネントは、ローカルな、および／またはリモートな処理を使用することによって、例えば、１または複数のデータパケットを有する信号（例えば、ローカルシステムの、分散システムの、および／または、信号を使用することによって他のシステムとインタラクトするインターネットなどのネットワークにわたる、別のコンポーネントとインタラクトする２つのコンポーネントからのデータ）によって、交信してよい。

【００４６】

次に、解決される必要がある技術的問題と、本願の応用的シナリオが提供される。近年、画像分類、画像認識、音声認識、および他の関連する分野において、畳み込みニューラルネットワークが良好な性能であるので、畳み込みニューラルネットワークは、学界および産業界において研究および開発のホットスポットとなっている。畳み込みニューラルネットワークは主に、畳み込み演算および全結合（ｆｕｌｌｙ－ｃｏｎｎｅｃｔｅｄ、ＦＣ）演算を含む。畳み込み演算の演算量は、通常、ネットワークの全演算量の７０％より多くを占有することがある。

【００４７】

畳み込み演算は、行列乗算演算と厳格に同等ではない。しかしながら、畳み込み演算は、適切なデータ調整によって、行列乗算演算に変換され得る。通常、畳み込みニューラルネットワークには複数の畳み込みカーネルがある。畳み込みカーネルは３次元であり、３次元のデータを含む。方向ｘおよびｙは、データの長さおよび幅を表し、方向ｚは、データの深さとみなされ得る。畳み込みカーネルは実際にはフィルタ（ｆｉｌｔｅｒ）であり、主に画像から異なる特徴を取り出すように構成される。図２を参照すると、畳み込みカーネルは、実質的に、一連の重みの組み合わせである。Ｋ個の畳み込みカーネルがあると前提する。Ｋ個の畳み込みカーネルの同じ位置で方向ｚにＮ個の要素が取り出され、その結果、Ｎ＊Ｋの重み行列（ｗｅｉｇｈｔ　ｍａｔｒｉｘ）が取得できる。畳み込みカーネルは、行列乗算器の仕様（具体的には、行列乗算器によって計算できる行列の行の数および列の数）に基づいて重み行列の形態で行列乗算器のメモリに予め格納されてよく、その結果、畳み込みカーネルは、行列乗算器が行列乗算演算を実行するとき呼び出される。本発明の実施形態において、「＊」は「乗算」を表す。

【００４８】

図３を参照すると、畳み込みカーネルのストライド（ｓｔｒｉｄｅ）（本発明のこの実施形態において、ストライドは１である）に基づいて、行列乗算器は、方向ｚにおいてＭ個の入力点のＮ個のデータ、すなわち、合計でＭ＊Ｎ個のデータを取り出し得る。入力行列（ｉｎｐｕｔ　ｍａｔｒｉｘ）が形成され得る。行列乗算器は、入力行列と重み行列に

対して乗算演算を実行する必要がある。

【００４９】

ＦＣ演算は、実質的にベクトルと行列との乗算演算である。ＦＣ演算の入力はベクトル９２１６であり、ＦＣ演算は４０９６の点を必要とする。この場合、ＦＣ演算によって出力される点を取得するために、ベクトル９１２６と９２１６個の重みとに対して小数点乗算演算が実行される必要があり、４０９６の点すべてを取得するためには、ベクトル９２１６と９２１６×４０９６個の重みとに対して小数点乗算演算が実行される必要がある。

【００５０】

図４は、行列Ｃ＝Ａ＊Ｂの計算式を示し、ここで、ＡはＭ＊Ｋサイズの行列を表し、ＢはＫ＊Ｎサイズの行列を表す。本発明のこの実施形態において、Ｍ、Ｎ、およびＫはそれぞれ正の整数である。計算によって行列Ｃの１個のデータを取得するためには、小数点乗算演算は、行列Ａにおける１つの行ベクトルのデータと、行列Ｂにおける１つの列ベクトルの対応するデータとで実行される必要があり、そして累算が実行される。言い換えれば、計算によって行列Ｃの１個のデータを取得するためには、Ｎ個の乗算演算が実行される必要がある。この場合、計算によって行列Ｃを取得するためには、Ｍ＊Ｎ＊Ｋ個の乗算演算が実行される必要がある。

【００５１】

先行技術において、シストリックアレイコンピューティング方式、例えば、機械学習のためにＧｏｏｇｌｅによってカスタマイズされた専用チップ（ＡＳＩＣ）、Ｇｏｏｇｌｅ　ＴＰＵｖ１は、２５６×２５６　２－Ｄ　ＭＡＣアレイを使用することによって、行列乗算および畳み込み演算（図５に示されるように）に最適化されたシストリックアレイ設計を使用する。図の各セルは１つの乗算器である。乗算器が２つの行列の要素を乗算した後、計算によって取得された結果（部分和、すなわち、行列乗算における中間結果）が、図の下部の累算ユニットに伝送され、以前の関連する累算値に累算される。このようにして、データがフルロード状態で動作するとき、シストリックアレイは、各クロックサイクルにおいて１つの行列のサイズの中間値を累算する。前述の解決手段において、計算密度が低いので、行列乗算計算効率は比較的低い。加えて、畳み込み演算の最中に、シストリックアレイのコンピューティングサイズは比較的固定されているので、シストリックアレイの演算効率を増加させるように、入力および重みは多くの形態に転換される必要があり、柔軟性のない演算をもたらす。さらに、行列乗算の最中に、パイプライン実行効果を実現するように、データは大きなサイズを有する必要がある。例えば、小さい行列における２５６×２５６　２－Ｄシストリックアレイの計算効率は高くない。

【００５２】

加えて、関連する特許は、Ｍ＊Ｋ＊Ｎ　３－Ｄ　ＭＡＣアレイを実装する。ＴＰＵｖ１およびＮＶＤＬＡ　２－Ｄ　ＭＡＣアレイ解決方法と比較すると、行列乗算計算効率は大幅に増大する。本発明は新しいハードウェアアクセラレータアーキテクチャを提供し、その結果、新しいハードウェアアクセラレータアーキテクチャは［Ｎ×Ｎ］行列乗算演算を単一のクロックサイクルで完了することができる。ハードウェアアーキテクチャにおいて、処理エンジン（ＰＥ）に含まれる数の個数はＮ×Ｎ×Ｎであり、加算木に含まれる数の個数はＮ×Ｎである。加えて、大きい行列をより小さい行列に分割する計算方法も、また提供される。しかしながら、前述の解決手段において、ハードウェアによってサポートされるサイズとなるように、行列サイズが追加される必要がある。このことは、データ帯域幅を浪費し、計算効率を低減させる。行列が人為的に大きい行列および小さい行列に分割される場合、ソフトウェアプログラミングは複雑であり、また、関連するソフトウェアプログラミング量も格段に増大する。加えて、アクセラレータが単方向に周期的方式のみ行列の要素をロードすることができ、ソフトウェアは独立して行列を分割する必要があるので、計算モードは単一でありフレキシブルでない。さらに、行列Ａおよび行列Ｂのメモリがすべてのデータを収容できなくなると、繰り返しの読み出しが発生する。したがって、バッファサイズはサービスアルゴリズムに比較的強く依存しており、具体的には、アクセラレータは、密結合したオンチップメモリに強く従属している。

【００５３】
したがって、本願において解決されるべき技術的問題は、効率的で、フレキシブルで、低いエネルギー方式で、ハードウェアを使用することによる畳み込みニューラルネットワークにおいて、多数のデータ演算をいかに実行するかである。
【００５４】
本発明のこの実施形態において提供される行列乗算器は、機械学習、ディープラーニング、および畳み込みニューラルネットワークなどの分野に適用されてよく、または、デジタル画像処理およびデジタル信号処理などの分野に適用されてよく、または、行列乗算演算に関連する他の分野に適用されてよいことが、理解されることができる。
【００５５】
前述の分析に基づいて、本願は、行列乗算アクセラレータを提供し、本願において提供される技術的問題を具体的に解析および解決する。図６は、本発明の実施形態による行列乗算アクセラレータの構造図である。図６に示されるように、行列乗算器６０は第１メモリ６０１、第２メモリ６０２、演算回路６０３、およびコントローラ６０４を含む。演算回路６０３は、バスを使用して、第１メモリ６０１、第２メモリ６０２、およびコントローラ６０４とデータ通信を実行し得る。演算回路６０３は、第１メモリ６０１および第２メモリ６０２から行列データを取り出し、ベクトル乗算および加算演算を実行するように構成される。コントローラ６０４は、ベクトル演算を完了するように、予め設定されたプログラムまたは命令に従って、演算回路６０３を制御するように構成される。第１メモリ６０１は第１行列を格納するように構成される。
【００５６】
第１行列はＭ＊Ｋ行列である。行列ａが第１行列である場合、第１行列ａの第ｉ行第ｊ列の要素はa_{ij}と表されてよく、ここで、ｉ＝（１，２，３，...，およびＭ）であり、ｊ＝（１，２，３，...，およびＫ）である。
【００５７】
本発明のこの実施形態において説明される第１メモリ６０１、および、以下に説明される関連する行列乗算器の第２メモリ６０２、第３メモリ６０６、および内部メモリは、それぞれ、レジスタ、ランダムアクセスメモリ（ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ、略してＲＡＭ）、静的ランダムアクセスメモリ、フラッシュメモリ、または別の読み出しおよび書き込み可能メモリであってよい。本願において、第１行列、第２行列および演算結果のデータ型はそれぞれ、ｉｎｔ８、ｆｐ１６、またはｆｐ３２などの型であってよい。
【００５８】
第２メモリ６０２は第２行列を格納するように構成され、第２行列はＫ＊Ｎ行列である。行列ｂが第２行列である場合、第２行列ｂの第ｊ行第ｇ列の要素はB_{jg}と表されてよく、ここで、ｊ＝（１，２，３，...，およびＫ）であり、ｇ＝（１，２，３，...，およびＮ）である。
【００５９】
本明細書では、Ｍ、Ｋ、Ｎ、Ｘ、およびＹはそれぞれ、０より大きい整数である。Ｍ、Ｎ、およびＫのうちいずれか２つのパラメータが等しくてもよく、等しくなくてもよい。代替的に、Ｍ、Ｎ、およびＫは、等しくてもよく、等しくなくてもよい。ＸおよびＹは、等しくてもよく、等しくなくてもよい。これは、本願を具体的に限定するものではない。
【００６０】
演算回路６０３は、Ｘ行＊Ｙ列の演算ユニット６０３０（乗算累算ユニットＭＡＣと称されてよい）を含み得る。各演算ユニットは、独立してベクトル乗算演算を実行し得る。図６において、演算回路６０３が４＊４演算ユニット６０３１を含む例が図に使用され、すなわち、Ｘ＝４およびＹ＝４である。演算ユニット６０３０は、それぞれ第１メモリ６０１によって送られた行ベクトルと、第２メモリ６０２によって送られた列ベクトルとを受信し、行ベクトルと列ベクトルとのベクトル乗算演算を実行するように使用される、２つの入力を提供される。具体的には、１つの演算回路６０３０はベクトル乗算回路および

加算回路を含み、ここで、行列乗算回路は第１メモリ６０１によって送られる行ベクトルのデータと、第２メモリ６０２によって送られる列ベクトルのデータとを受信し、２つのベクトルを乗算するように構成され、加算回路は、２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、演算ユニットの演算結果を取得するように構成される。

【００６１】

図７は、演算ユニット６０３０の構造図である。可能な実装において、ベクトル乗算回路はＬ個（例えば、Ｌ＝４）の乗算器を含む。加算回路は、入力数がＬ＋１である加算木を含み、具体的には、加算木はＬ個の乗算結果を累算し、異なるクロックサイクルの演算ユニットの計算結果を累算するように構成される。任意で、行列乗算器６０は第３メモリ６０５をさらに含み、第３メモリ６０５はベクトル乗算回路および加算回路の演算結果を格納し、異なるクロックサイクルの演算結果を格納するように構成される。本願における第３メモリ６０５は、Ｘ＊Ｙ個のストレージユニットを含み得、各ストレージユニットは、対応する演算ユニットが演算を実行するたびに、取得された演算結果を格納するように構成されることが、理解され得る。代替的に、各演算ユニットは、第３メモリ６０５の指定されたストレージスペースに対応し、ストレージスペースは、演算ユニットが演算を実行するたびに取得された演算結果を格納するように使用される。

【００６２】

コントローラ６０４は、第１行列および第２行列の積を計算するように、以下の動作を実行し得る。

【００６３】

コントローラ６０４は、第１行列を、サイズがＸ＊Ｌであるサブブロックを単位とするブロックに分割し、同じサイズのＳ×Ｒ個のサブブロックを取得し、ここで、Ｓ×Ｒ個のサブブロックのうち第ｓ行第ｒ列におけるサブブロックは、A_{sr}と表され、ｓ＝（１，２，３，...，およびＳ）、ｒ＝（１，２，３，...，およびＲ）である。すなわち、本願の行列乗算器６０に関して、行列乗算器６０に含まれるＸ行＊Ｙ列の行列データは、生産または送出の後に固定され、対応する乗算回路における数Ｌの乗算器もまた固定される。したがって、行列演算の最中、第１行列および第２行列はフラクタルである必要があり、すなわち、ブロックに分割される必要がある。分割方式は、第１行列を、Ｘ＊Ｌサブブロックを単位として使用したブロックに分割することである。本発明のこの実施形態において、ブロック分割の目的は、大きい行列を、行列乗算器のサイズに準拠した多くの小さい行列に分割し、そして、特定のシーケンスの小さい行列を計算し、関連する小さい行列の値を累算し、最終的に行列乗算結果を取得することである。このようにして、フレキシブルな計算が実行でき、後の再使用およびマルチレベルのキャッシングを円滑化し、計算効率がさらに増大することができ、データ移動帯域幅およびエネルギー消費を低減することができる。

【００６４】

第１行列がＭ＊Ｋ行列であり、第１行列が整数個のＸ＊Ｌサブブロックで正確に分割できない場合が存在し得ることに、留意すべきである。したがって、Ｍ／ＸまたはＫ／Ｌが整数でないとき、演算は要素０をパッディングする方式で実行されてよい。代替的に、対応する位置で全く計算が実行されず、結果の値に０が割り当てられる。具体的には、

【数５】

$$S=\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

および

【数６】

$$R=\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、Ｍ％Ｘ≠０のとき、計算は第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は第１行列の第（Ｋ＋１）行から第（Ｒ＊Ｙ－Ｋ）行まで実行されず、結果の値には０が割り当てられる。言い換えれば、演算ユニットは対応する行および列において実体的乗算計算を実行せず、処理のために、演算が実行されたが結果が０であるとみなす。このようにして、対応する演算ユニットの読み出しおよび演算電力消費は低減され得る。

【００６５】

対応して、コントローラ６０４は、第２行列を、サイズがＬ＊Ｙであるサブブロックを単位とするブロックに分割し、同じサイズのＲ×Ｔ個のサブブロックを取得し、ここで、Ｒ×Ｔ個のサブブロックのうち第ｒ行第ｔ列におけるサブブロックは、B_{rt}と表され、ｒ＝（１，２，３，...，およびＲ）、ｔ＝（１，２，３，...，およびＴ）である。コントローラ６０４が第１行列を、演算回路６０３の仕様に従ってブロックに分割されるように制御した後、第２行列もまた、第１行列に合致する方式で対応して分割されることを必要とし、そうでなければ、行列乗算計算は実行され得ない。

【００６６】

第２行列がＫ＊Ｎ行列であり、第２行列が整数個のＬ＊Ｙサブブロックで正確に分割できない場合が存在し得ることに、留意すべきである。したがって、Ｋ／ＬまたはＮ／Ｙが整数でないとき、演算は要素０をパッディングする方式で実行されてよい。代替的に、対応する位置で全く計算が実行されず、結果の値に０が割り当てられる。具体的には、

【数７】

$$R=\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

および

【数８】

$$T=\begin{cases} N/Y, N\%Y = 0 \\ [\frac{N}{Y}] + 1, N\%Y \neq 0 \end{cases}$$

であり、Ｋ％Ｙ≠０のとき、計算は第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられ、Ｎ％Ｘ≠０のとき、計算は第１行列の第（Ｎ＋１）行から第（Ｔ＊Ｘ－Ｎ）行まで実行されず、結果の値には０が割り当てられる。言い換えれば、演算ユニットは対応する行および列において実体的乗算計算を実行せず、処理のために、演算が実行されたが結果が０であるとみなす。このようにして、対応する演算ユニットの読み出しおよび演算電力消費は低減され得る。

【００６７】

固定された仕様に従って、第１行列および第２行列が別々にブロックに分割された後、２つの行列は、サブブロックの間の行列乗算演算を実行するように、演算回路６０３に入力されてよい。具体的な計算処理において、コントローラ６０４は、任意のサブブロックA_{sr}のＸ個の行ベクトルのうちの第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルのうちの第ｙ列とが、Ｘ行＊Ｙ列からなる演算ユニットにおける第ｘ行第ｙ列の演算ユニットに入力され、それにより演算を実行するように制御し得、ここで、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，Ｙ）であり、任意のサブブロックA_{sr}におけるｒおよび対応するサブブロックB_{rt}におけるｒは同じ値を有する。サブブ

ロックA_{sr}の行ベクトルおよびサブブロックB_{rt}の列ベクトルが演算ユニットに入力される前に、第１行列および第２行列はブロックに分割されており、すなわちフラクタルである。したがって、特定のシーケンスにおいて、演算回路６０３にサブブロックA_{sr}および対応するサブブロックB_{rt}を入力する複数の実装があってよい。

【００６８】

可能な実装において、サブブロックA_{sr}および対応するサブブロックB_{rt}におけるｓまたはｔの値のシーケンスにおいて連続的に演算が実行されてよい。図８に示されるように、例えば、第１行列はＭ＊Ｋ行列であり、第２行列はＫ＊Ｎ行列である。Ｍ＝１２、Ｋ＝６、Ｎ＝１２、Ｘ＝４、Ｙ＝４、およびＬ＝３であると前提される。第１行列および第２行列がブロックに分割された後、Ｓ＝３、Ｒ＝２、およびＴ＝３であるとわかる。この場合、第１行列

【数９】

$$A=\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{33} \end{bmatrix}$$

および第２行列

【数１０】

$$B=\begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

がブロック分割後に取得され、ここで、ＡはＸ＊Ｌ行列、すなわち４＊３行列を表し、Ｂの各要素は実際、Ｌ＊Ｙ行列、すなわち３×４行列である。

【００６９】

Ｃ＝Ａ＊Ｂ＝

【数１１】

$$\begin{bmatrix} A_{11}B_{11}+A_{12}B_{21} & A_{11}B_{12}+A_{12}B_{22} & A_{11}B_{13}+A_{12}B_{23} \\ A_{21}B_{11}+A_{22}B_{21} & A_{21}B_{12}+A_{22}B_{22} & A_{21}B_{13}+A_{22}B_{23} \\ A_{31}B_{11}+A_{32}B_{21} & A_{31}B_{12}+A_{32}B_{22} & A_{31}B_{13}+A_{32}B_{23} \end{bmatrix}$$

【００７０】

第１行列および第２行列の乗算演算において、行列乗算演算が、任意の１つのサブブロックA_{sr}、すなわち、第１行列の各サブブロックA_{sr}と、第２行列の対応するサブブロックB_{rt}とで実行されることが必要である。行列乗算計算がシーケンスで最初に実行される特定のシーケンスおよび特定のサブブロックを決定する、複数の実装があり得る。

【００７１】

方式１：行列乗算シーケンスにおいて、例えば、サブブロックはサブブロックA_{11}およびサブブロックB_{11}であってよい。A_{11}のすべての行ベクトルと、対応するB_{11}のすべての列ベクトルとが、サブブロック乗算計算サイクル（第１のラウンドとして理解されてよい）に入力され、それにより、演算を実行する。A_{12}のすべての行ベクトルおよび対応するB_{21}のすべての列ベクトルでの演算が、第２サブブロック乗算計算サイクル（第２のラウンドとして理解されてよい）において実行される。このようにして、演算ユニットが累算を実行した後、結果行列Ｃの第１行第１列における結果点C_{11}の値が取得され得る。同様に、結果行列Ｃのすべての位置における結果点が取得され得る。実際、

$C_{11} =$

【数１２】

$$A_{11}B_{11} + A_{12}B_{21}$$

ここで、

【数１３】

$$A_{11} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

【数１４】

$$A_{12} = \begin{bmatrix} a_{14} & a_{15} & a_{16} \\ a_{24} & a_{25} & a_{26} \\ a_{34} & a_{35} & a_{36} \\ a_{44} & a_{45} & a_{46} \end{bmatrix}$$

【数１５】

$$B_{11} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix}$$

【数１６】

$$B_{12} = \begin{bmatrix} b_{14} & b_{15} & b_{16} \\ b_{24} & b_{25} & b_{26} \\ b_{34} & b_{35} & b_{36} \\ b_{44} & b_{45} & b_{46} \end{bmatrix}$$

【００７２】

言い換えれば、C_{11}は実際、４＊４行列である。したがって、行列計算ルールによれば、最終的に取得される行列ＣはＭ＊Ｎ結果行列であり、すなわち、１２＊１２結果行列である。

【００７３】

方式２：１つのサブブロックが具体的ルールによって再使用される。本発明のこの実施形態は、第１行列の１つのサブブロックA_{sr}および第２行列の対応するサブブロックB_{rt}に、サブブロックでの行列乗算演算の実行をもたらすサブブロック再使用方式を提供する。具体的には、コントローラ６０４はさらに、ｓおよびｒの値を変更されないままに

して、ｔの値を、少なくとも２つの連続するサブブロック乗算計算サイクルにおいて変更されるように制御するように構成され、その結果、第１メモリは、少なくとも２つの連続するサブブロック乗算計算サイクル内で同じサブブロックA_{sr}を再使用し、ここで、サブブロック乗算計算サイクルは、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}での行列乗算演算を完了するように、Ｘ行＊Ｙ列の演算ユニットによって使用される時間である。

【００７４】

例えば、Ｍ＝１２、Ｋ＝６、Ｎ＝１２、Ｘ＝４、Ｙ＝４、およびＬ＝３を前提とする前述した実施形態において、A_{11}のすべての行ベクトルおよび対応するサブブロックB_{11}のすべての列ベクトルが、サブブロック乗算計算サイクル（第１のラウンドとして理解されてよい）に入力され、それにより演算を実行する。第２サブブロック乗算計算サイクル（第２のラウンドとして理解されてよい）において、ｓおよびｒの値は変更されないままであり、しかし、ｔの値は変更される必要があり、具体的には、A_{11}のすべての行ベクトルおよび別の対応するサブブロックB_{12}のすべての列ベクトルで演算が実行される。任意で、第３のサブブロック乗算計算サイクル（第３ラウンドとして理解されてよい）において、A_{11}のすべての行ベクトルおよびさらに別の対応するサブブロックB_{13}のすべての列ベクトルで演算が実行される。このようにして、第１メモリのA_{11}は、複数の連続するサブブロック乗算計算サイクルで繰り返し使用されることができ、その結果、読み出しおよび書き込みオーバーヘッドが低減され、データ移動帯域幅が低減される。

【００７５】

方式１および方式２において、サブブロック乗算計算サイクルにおける、第１行列のサブブロックA_{sr}および第２行列の対応するサブブロックB_{rt}に関する計算ルールは、第１行列における任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルのうちの第ｙ列とが、Ｘ行＊Ｙ列からなる演算ユニットにおける第ｘ行第ｙ列の演算ユニットに入力され、それにより演算を実行するように制御し得、ここで、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，Ｙ）であり、任意のサブブロックA_{sr}におけるｒおよび対応するサブブロックB_{rt}におけるｒは同じ値を有する。すなわち、サブブロックA_{sr}の任意の行ベクトルと、第２行列の対応するサブブロックB_{rt}の任意の列ベクトルは、計算のために、Ｘ行＊Ｙ列からなる演算ユニットにおける指定された演算ユニットに入力される。例えば、A_{11}の第２行ベクトル

【数１７】

$$\begin{bmatrix} a_{21} & a_{22} & a_{23} \end{bmatrix}$$

および、第２行列の対応するサブブロックB_{11}における第３列ベクトル

【数１８】

$$\begin{bmatrix} b_{13} \\ b_{23} \\ b_{33} \end{bmatrix}$$

は、演算のために、Ｘ行＊Ｙ列からなる演算ユニットにおける第２行第３列に対応する演算ユニットに入力される、などである。

【００７６】

図６で示された演算回路６０３における演算ユニットの構成方式によると、図９は、本発明の実施形態による特定の演算回路６０３の配線の概略図である。

【００７７】

ＢＵＦＡは第１行列の第１メモリ６０１であり、ＢＵＦＢは第２行列の第２メモリ６０２であり、ＢＵＦＣは各演算ユニット６０３０の計算結果を格納する第３メモリ６０５で

あり、演算回路６０３はＸ行＊Ｙ列（Ｘ＝４、Ｙ＝４と前提する）からなる演算ユニット、すなわち図のＭＡＣ　ＧＲＰ　Ｒ００Ｃ００からＭＡＣ　ＧＲＰ　Ｒ０３Ｃ０３を含む。加えて、各演算ユニットＭＡＣ　ＧＲＰは、Ｘ＊Ｌ行列の１つの行ベクトルとＬ＊Ｙ行列の１つの列ベクトルで乗算演算を実行し得る。

【００７８】

本発明のこの実施形態において、演算回路６０３は、フラクタル行列乗算ユニットと称されてよく、３－Ｄ　ＭＡＣアレイ（ＭＡＣ　Ｃｕｂｅ）およびアキュムレータ（Ａｃｃｕｍｕｌａｔｏｒ）を含み、以下のようなフラクタル行列乗算命令を実行するように構成される。Ｃ＝Ａ＊ＢまたはＣ＝Ａ＊Ｂ＋Ｃ、ここで、Ａ／Ｂ／Ｃは２次元行列である。Ａのサイズは（Ｍ＊ベース）×（Ｋ＊ベース）、Ｂのサイズは（Ｋ＊ベース）×（Ｎ＊ベース）、Ｃのサイズは（Ｍ＊ベース）×（Ｎ×ベース）である。ベースは演算回路６０３の基本サイズであり、すなわちＸ＊Ｙ、例えば、８＊８、１６＊１６、および３２＊３２である。前述のＣ＝Ａ＊ＢまたはＣ＝Ａ＊Ｂ＋Ｃ計算演算は、ＭＮＫ行列乗算（および累算）と称される。実際の実行処理において、コントローラは、特定のシーケンスの組み合わせ（上で説明された方式１または方式２）におけるフラクタル方式でＭＮＫ行列乗算を完了させるように、大きい行列を、ベースサイズの基本行列に分割されるように制御する。

【００７９】

フラクタル行列乗算ユニットの具体的なアーキテクチャは、図７に示される（ベース＝４と前提する）。例えば、図７において、ＭＡＣグループはＮ＊Ｎ（４＊４）からなる乗算累算グループであり、Ｎ（４）個の乗算ユニット、および、入力数がＮ＋１（５）である累算木を含む。行列乗算に関して、乗算アキュムレータが、１つの行に１つの列を乗算して累算（すなわち、結果行列の１つの要素）を実行する演算を、実行してよい。図９において、４×４乗算累算グループの全体があり、すなわち、完全な４×４＊４×４の行列乗算演算が同時に計算され得る。

【００８０】

図９の配線の概略図において、演算回路６０３は、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}での行列乗算計算を、同じクロックサイクルにおいて完了することのサポートとなり得ることが、理解されることができる。サブブロックA_{sr}のＸ個の行ベクトルのすべてと、対応するサブブロックB_{rt}のＹ個の列ベクトルのすべてとが、図９の配線方式において対応するＢＵＦＡおよびＢＵＦＢから同時に、対応する演算ユニット６０３０に到達し得るので、コントローラ６０４は、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}での乗算計算を１クロックサイクルで完了するように、および、次のクロックサイクルにおいて、別のサブブロックA_{sr}および対応するサブブロックB_{rt}での乗算計算を完了するか、または、同じサブブロックA_{sr}および対応する別のサブブロックB_{rt}での行列乗算計算を完了するように、演算回路６０３を制御し得る。

【００８１】

図１０は、本発明の実施形態による特定の演算回路６０３の配線の概略図である。演算回路６０３において、図１０に対応して、シストリックアレイ構造が提供される。具体的に、コントローラ６０４はさらに、任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、コントローラはさらに、対応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｙ行に連続的に入力するように同時に制御するようにさらに構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである。

【００８２】

すなわち、各演算ユニット６０３０（乗算アキュムレータ）をフルに利用するように、本発明のこの実施形態におけるフラクタル行列乗算ユニットは、シストリックアレイ構造

を有してよい。ＴＰＵｖ１構造からの差は、（ＴＰＵｖ１におけるデータの個数は１であるが）各シストリック伝送におけるデータの個数がＬであることにある。したがって、データ処理の並行性の程度は、ＴＰＵｖ１におけるシストリックアレイのそれより高い。

【００８３】

シストリックアレイアーキテクチャに基づいて、図１０に対応する配線構造において、ＢＵＦＡ／Ｂが、それぞれ、第１行列／第２行列をバッファリングするように構成されるメモリである。図１０において、第１行列バッファ（ＢＵＦＡ）は、各クロックサイクルにおいて、行列Ａにおける単位行列をＸ個の行に分割し、同じ行のＬ個の要素をシストリックアレイの演算ユニットに順次送る。同様に、第２行列バッファ（ＢＵＦＢ）は、各クロックサイクルにおいて、第２行列における単位行列をＹ個の列に分割し、同じ列のＬ個の要素をシストリックアレイに順次送る。具体的な時間シーケンスは以下の通りである。

【００８４】

ＢＵＦＣは、「Ａ＊Ｂ＋Ｃ」計算における「Ｃ」（オフセット）行列を格納するバッファ（Ｌ０バッファまたはバッファレジスタを使用して構築されてよい）であり、また、行列乗算の中間値がＢＵＦＣに格納されてもよい。乗算アキュムレータが乗算を完了した後、累算木は、乗算後に取得されたＬ個の中間値と、ＢＵＦＣに格納された１つのオフセットまたは中間値を累算する。

【００８５】

Ｍ＝２、Ｎ＝２、およびＫ＝２（すなわち８×８＊８×８行列乗算が使用される例。行列乗算器６０のコントローラ６０３は、図１１のフォーマットで行列乗算を分割し、全体で８個の４×４単位行列演算を取得する。ＭＮＫ行列乗算演算に関して、分割シーケンスのための多くの可能性があり、分割シーケンスのルールは、ＭＮＫ行列乗算演算が、方式１および方式２のシーケンスにおいて実行されてよいことである。方式２において再使用するデータの最大数倍のポリシーを使用することによって、データを読み出すための消費電力を低減し得ることが、理解されてよい。ＭＮＫフラクタル分割が実行された後、図１２から図１５に示されるように、コントローラ６０３の制御ロジックは８クロックサイクルに８個のフラクタルをシストリックアレイに入力する。図１２は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝０におけるフラクタル行列乗算器のパイプライン実行を示し、図１３は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝１における行列乗算器のパイプライン実行を示し、図１４は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝７におけるフラクタル行列乗算器のパイプライン実行を示し、図１５は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝１１におけるフラクタル行列乗算器のパイプライン実行を示す。シストリックアレイは、Ｔ＝６のとき、すなわち、第７クロックサイクルにおいてフルロード状態で動作を開始することがわかる。最後の６クロックサイクルにおいて、単位行列がシストリックアレイから出力され、行列全体の乗算演算もまた完了する。

【００８６】

任意で、図１６を参照すると、行列乗算器６０は、命令発送ユニット６０６、命令フェッチユニット６０７、直接メモリアクセスユニット６０８、ベクトルユニット６０９、スカラユニット６１０、およびバスインタフェースユニット６１１をさらに含んでよい。さらに、本発明のこの実施形態において提供される行列乗算器６０は、コプロセッサとして使用され、中央演算処理装置（Ｃｅｎｔｒａｌ Ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ，略してＣＰＵ）８０上に載置されてよく、ＣＰＵは行列乗算器６０に計算タスクを割り当てる。具体的に、ＣＰＵ８０は第１行列、第２行列、および外部メモリ７０への関連する命令を格納し得る。行列乗算器６０は、第１行列、第２行列、および外部メモリ７０における関連する命令を読み出すことによって、行列乗算演算を完了し得る。外部メモリ７０は、具体的には、ダブルデータレートシンクロナスダイナミックランダムアクセスメモリ（Ｄｏｕｂｌｅ Ｄａｔａ Ｒａｔｅ Ｓｙｎｃｈｒｏｎｏｕｓ Ｄｙｎａｍｉｃ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ、略してＤＤＲ）、または別の読み出しおよび書き込み可能メモリであってよい。外部メモリは、行列乗算器６０からプライベートなメモリであってよい。具体的に、第１メモリ６０１、第２メモリ６０２、第３メモリ６０５、およ

び外部メモリ７０は一般的に、オンチップメモリ（Ｏｎ－Ｃｈｉｐ　Ｂｕｆｆｅｒ）である。

【００８７】

１．ベクトルユニット６０９（Ｖｅｃｔｏｒ　Ｕｎｉｔ）は、様々な種類のマルチパラレルコンピューティングデバイス（例えば、浮動小数点乗算、浮動小数点加算、浮動小数点値比較）を含み、ここで、コンピューティングデバイスは、ＳＩＭＤ（Ｓｉｎｇｌｅ　Ｉｎｓｔｒｕｃｔｉｏｎ　ｍｕｌｔｉｐｌｅ　ｄａｔａ）命令を実行するように構成され、統一されたバッファ（Ｕｎｉｆｉｅｄ　Ｂｕｆｆｅｒ）およびＬ０Ｃバッファのために移動する直接のデータに責任を負う。

【００８８】

２．スカラユニット６１０（Ｓｃａｌａｒ　Ｕｎｉｔ）は、様々な種類の整数基本演算デバイス（例えば、加算、乗算、比較、およびシフト）を含む。

【００８９】

３．直接メモリアクセスユニット（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ　Ｕｎｉｔ，ＤＭＡ　Ｕｎｉｔ）は、各ストレージユニットにデータを移動するように、例えば、Ｌ１　ＲＡＭからＬ０　ＲＡＭへデータを移動するように構成される。本発明のこの実施形態における直接メモリアクセスユニットが、行列乗算器の外部メモリまたは内部メモリから、乗算演算に関与する行列データを移動するとき、直接メモリアクセスユニットは、行列がブロックに分割された後に取得された結果を格納する必要がある。例えば、２＊２行列に関して、第１行列の第１行第１列のサブブロックA_{11}＝

【数１９】

$$\begin{bmatrix} A0 & A1 \\ A2 & A3 \end{bmatrix}$$

がサブブロックのユニットに格納され、Ａ０、Ａ１、Ａ２およびＡ３が１つの行に格納される、などである。このようにして、第１行列または第２行列が、対応する第１メモリへと移動されてよいとき、または、第２行列が、対応する第２メモリに移動されてよいときは、ストレージは、前述の方式で実行されてよい。演算ユニットが読み出しの実行を必要とするとき、演算ユニットもまた、前述のストレージシーケンスにおいて読み出しを実行してよく、それにより、計算を円滑化する。行ベクトルが列ベクトルに転置される必要があるとき、転置はフレキシブルに、そして素早く実行されてよい。

【００９０】

４．命令フェッチユニット６０７（Ｉｎｓｔｒｕｃｔｉｏｎ　Ｆｅｔｃｈ　Ｕｎｉｔ，ＩＦＵ）は、内部でＰＣ（プログラムカウンタ）およびＩＭ（命令メモリ）に統合され、メインメモリからバスインタフェースユニット（ＢＩＵ）６１１を使用して命令をフェッチし、実行手順を復号および制御する。

【００９１】

５．命令発送ユニット６０６（Ｄｉｓｐａｔｃｈ　Ｕｎｉｔ）は、命令フェッチユニットによって伝送された命令を構文解析し、命令に対応するタイプ命令を４つのパイプラインユニットに提示し、ここで、パイプラインユニットは図１６のスカラユニット（Ｓｃａｌａｒ　Ｕｎｉｔ）、ダイレクトメモリアクセス（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ，ＤＭＡ）ユニット、ベクトルユニット（Ｖｅｃｔｏｒ　Ｕｎｉｔ）、およびフラクタル行列乗算ユニットである。命令発送ユニットが、４つのパイプラインの間の順序立てた実行を制御するためのメカニズムがある。

【００９２】

パイプラインユニットには２つの型、すなわち非同期実行（Ｐｏｓｔｅｄ　Ｅｘｅｃｕｔｉｏｎ）および同期実行があることに留意すべきである。すべてのタイプ命令は順序保持方式で伝送される。違いは、非同期実行ユニットによる命令の実行は非同期的に終了し、同期実行ユニットによる命令の実行は同期的に終了する、ということにある。スカラユ

ニット（Ｓｃａｌａｒ　Ｕｎｉｔ）は同期実行ユニットであり、フラクタル行列乗算ユニット（Ｆｒａｃｔａｌ　Ｍａｔ　Ｍｕｌｔ　Ｕｎｉｔ）、ＤＭＡユニット、およびベクトルユニット（Ｖｅｃｔｏｒ　Ｕｎｉｔ）は非同期実行ユニットである。

【００９３】

可能な実装において、直接メモリアクセスユニットに関して、本発明のこの実施形態は、構成可能なオンフライト行列転置機能を提供する。例えば、第１行列のブロック行列がメモリ（例えば、行列乗算器の外部メモリ）から、別のメモリ（第１メモリなどの、行列乗算器の内部メモリ）に移動されるとき、直接メモリアクセスユニットは、当該移動の最中に行列転置演算を実行し、転置行列の順序で転置行列を格納する。行列転置は、ニューラルネットワークトレーニングプロセスの必須の演算フェーズである。移動後の転置の実行のための共通命令と比較すると、本発明のこの実施形態における構成可能なオンフライト行列転置のための移動命令は、よりフレキシブルであり、また、ソフトウェアはより容易に、およびより簡潔にされる。詳細は以下の表に示される。

【００９４】

共通命令：構成可能なオンフライト行列転置機能のための命令。

【表１】

LOAD_L0	X2 , {X1}	LOAD_L0_to_L1.Trans {X4}, {X1}
Transpose	X3 , X2	
STORE_L1	{X4}, X3	

【００９５】

共通移動命令が、構成可能なオンフライト行列転置機能のための命令と比較される。構成可能なオンフライト行列転置機能をサポートすることによって、同じ命令が、異なるパラメータへの、より多くの応用的シナリオをサポートし得る。フラクタル行列乗算プロセッサアーキテクチャに適用可能な、構成可能なオンフライト行列転置方法が設計される。

【００９６】

図１７を参照すると、データ再使用を円滑化し、消費電力を低減し、密結合されたオンチップメモリへの依存を低減するように、本発明の実施形態はさらに、マルチレベルバッファを使用するストレージ構造を提供する。すべての演算ユニットは、統一されたバッファ（Ｕｎｉｆｉｅｄ　Ｂｕｆｆｅｒ）を使用することによって、相互作用データを読み出し／書き込みしてよい。行列乗算器内には、２つのレベルの専用バッファＬ１およびＬ０がある。Ｌ１バッファおよび統一されたバッファは通常、ダイレクトメモリアクセスＤＭＡユニットを使用して、外部格納空間とデータを交換する。外部格納空間は複数レベルのストレージユニットを含む。例えば、行列乗算器は複数レベルのバッファを含み、Ｌ０からＬ１へ、そしてＬ２バッファへと、容量が次第に増加し、帯域幅が次第に減少し、遅延が次第に増加し、消費電力オーバーヘッドが次第に増加する。Ｌ０は、最も内側のレベルのバッファであり、ＭＮＫ乗算命令の３つの行列「第１行列」「第２行列」および「結果行列」をバッファリングするように構成されてよい。Ｌ０は計算に近いので、帯域幅および遅延に関する要件はもっとも高く、データ再使用の可能性は最大である。性能を改善して消費電力を低減させるように、Ａ　Ｄトリガ（ＤＦＦ）がＬ０を構築するために使用されてよい。フラクタル命令のソースおよび宛先オペランドは、Ｌ１（図１７の第５メモリ６１２および第４メモリ６１３）から来る。実行の最中に、データはＬ０（例えば、図１７の第１メモリ６０１および第２メモリ６０２）を使用することによって再使用される。上記のフラクタル命令のようなソフトウェアは、Ｌ１を使用することによってデータを再使用し得る。マルチレベルバッファにおけるデータ再使用は、フラクタル命令を実行するシーケンスおよびフラクタル命令の上のソフトウェアを制御するシーケンスを使用することによって実装され得る。加えて、マルチレベルバッファのデータを再使用することによって、各バッファのデータのデータ移動時間もまた隠され得る。以下の表の例は、データ再使用と、バッファの複数のレベルの間の移動とを説明し得る。

【００９７】

以下の２つの行列：Ａ＝

【数２０】

$$\begin{bmatrix} A0 & A1 \\ A2 & A3 \end{bmatrix}$$

およびＢ＝

【数２１】

$$\begin{bmatrix} A0 & A1 \\ A2 & A3 \end{bmatrix}$$

があり、２つの行列のデータ移動ステップが以下の表に示されると前提する。

【表２】

時点	Ｌ１からの読み出し	Ｌ０への格納	計算
1	Ａ０，Ｂ０		
2	Ｂ１	Ａ０，Ｂ０	Ａ０＊Ｂ０
3	Ａ２	Ａ０，Ｂ０，Ｂ１	Ａ０＊Ｂ１
4	Ａ１	Ａ０，Ａ２，Ｂ０，Ｂ１	Ａ２＊Ｂ０
5	Ｂ２	Ａ１，Ａ２，Ｂ０，Ｂ１	Ａ２＊Ｂ１
6	Ｂ３	Ａ１，Ａ２，Ｂ１，Ｂ２	Ａ１＊Ｂ２
7	Ａ３	Ａ１，Ａ２，Ｂ２，Ｂ３	Ａ１＊Ｂ３
8		Ａ２，Ａ３，Ｂ２，Ｂ３	Ａ３＊Ｂ２
9		Ａ２，Ａ３，Ｂ２，Ｂ３	Ａ３＊Ｂ３

【００９８】

時点１において、コントローラ６０はＬ１バッファから行列のＡ０およびＢ０部分を読み出し、Ａ０およびＢ０部分をＬ０に格納する。

【００９９】

時点２において、Ａ０およびＢ０フラクタル行列は、Ｌ０から読み出され、演算に関与することができる。同時に、ハードウェアはＬ１からＢ１フラクタルを読み出し、Ｂ１フラクタルをＬ０に格納し、次の動作に関する準備を行う。加えて、データの読み出し時間もまた、計算によって隠される。この場合、ハードウェアは２つのフラクタル行列の両者を読み出す必要はなく、Ｂ１行列のみを読み出す。「Ａ０＊Ｂ１」が時点３において行列のために計算されるとき、時点１において格納されたデータＡ０が再使用される。前述のリストを参照すると、データが各時間単位において再使用されることが、後の計算においてわかることができる。

【０１００】

本発明のこの実施形態は、Ｌ１およびＬ０の間のデータの移動に限定されるものではないことに留意すべきである。Ｌ２（例えば、外部メモリ７０１および外部メモリ７０２）からＬ１バッファへとデータを移動する最中に、データもまた、帯域幅を低減させてエネルギー消費を最適化するように、再使用されてよい。本発明のこの実施形態において、行

列分割方式および移動シーケンスは、限定されるものではない。データ再使用は、各時間単位にデータ移動を実現するように、データ移動の最中に最大化されるべきであり、フラクタル行列計算はフルロード状態で実行される。

【０１０１】

本発明のこの実施形態において、マルチレベルバッファ構造、行列フラクタルデータ再使用、フラクタル命令を実行するシーケンス、およびフラクタル命令の上のソフトウェアを制御するシーケンスを使用することによって、マルチレベルバッファにおけるデータ再使用が実現でき、密結合するオンチップメモリへの依存が低減され、エネルギー効率が最適化され、ソフトウェアプログラミングの複雑性が低減される。

【０１０２】

本発明のこの実施形態において、行列で乗算演算を実行するための命令を実行するシーケンスは、２つの方式を含む：命令同期実行および命令非同期実行である。

【０１０３】

本発明のこの実施形態において、フラクタル行列乗算命令が実行される前に、例えば、行列サイズの計算、行列データの読み出し、および宛先アドレスの計算といった、一連の制御準備およびデータ準備が必要とされる。プロセッサの命令実行ポリシーが同期実行である場合、具体的には、すべての命令がシーケンスにコミット（ｃｏｍｍｉｔ）される必要がある場合、関連づけられていない命令が終了するまで、命令の実行が開始しないという可能性が非常に高い。このことは、大きくそして不要である、性能の損失をもたらすことがある。以下の手順は、命令同期実行シーケンスである：アドレス計算→制御準備→行列０の読み出し→行列０の乗算→アドレス計算→制御準備→行列１の読み出し→行列１の乗算。

【０１０４】

前述の実行シーケンスにおいて、第２の時間の制御準備、アドレス計算、行列１のデータの読み出しは、行列０の乗算の終了に依存せず、そのような追加の時間は、不要な待ち時間をもたらすことがある。この問題を解決するように、本発明のこの実施形態において、ハードウェア命令発送ユニット６０６は、マルチチャネル順序保持方式で伝送を実行し、それにより、異なる型の命令が同時にかつ順次実行されることを可能にすることを保証する。前述の例において、制御準備およびアドレス計算が順序保持方式によってスカラチャネル上で実行され、行列読み出しおよび格納が順序保持方式によってデータ移動チャネル上で実行され、行列乗算計算もまた、順序保持方式によって行列演算チャネル上で実行される。チャネルは、オーバーラップしてよいが、順序保持はされず、互いに従属した命令が、待ちフラグ（Ｗａｉｔ　Ｆｌａｇ）を設定することによって同期されてよい。命令非同期実行ポリシーを使用することによって、命令は並列で実行され得、これにより、ますます実行効率が増大する。前述の同期実行シーケンスの例において、非同期実行ポリシーが使用される場合、効果は図１８に示される。命令非同期実行シーケンスにおいて、命令は順序保持されず、依存関係を有する関連する命令が、ソフトウェアによって追加された待ち命令を使用することによって同期されてよい。フラクタル行列乗算の制御準備オーバーヘッドは、この非同期実行方式を使用することによって隠され得る。フラクタル行列乗算プログラミング方式に適用可能な非同期実行方式が設計される。

【０１０５】

行列乗算器が提供され、ここで、行列乗算器は行列乗算ブロック分割方法、すなわち、ＭＮＫフラクタルを完了して、行列乗算器６０における内部コントローラ６０４の制御ロジックを使用することによって、乗算のために大きい行列を単位行列（具体的には、Ｘ＊Ｌ×Ｌ＊Ｙ行列）に分割するようにコントローラを使用する。コントローラ６０４の制御ロジックは、各クロックサイクルにおいて、単位行列乗算タスクを演算回路６０３に送り、その結果、データがパイプライン方式で実行され、Ｘ行＊Ｙ列の演算ユニットがフルロード状態で動作する。行列乗算の効率が増大し、ニューラルネットワークアルゴリズムを大幅に改善する適用効果が実現される。本発明のこの実施形態において提供される行列乗算器は、畳み込みニューラルネットワークにおける畳み込み演算およびＦＣ演算を実行し

得る。

【０１０６】

前述した実施形態のすべてまたは一部は、ソフトウェア、ハードウェア、ファームウェア、またはそれらの組み合わせによって実装されてよい。実施形態を実装するようにソフトウェアプログラムが使用されるとき、実施形態は、完全に、または部分的に、コンピュータプログラム製品の形態で実装されてよい。コンピュータプログラム製品は、１または複数のコンピュータ命令を含む。コンピュータプログラム命令がコンピュータ上でロードおよび実行されるとき、本願の実施形態による手順または機能が、すべてまたは部分的に生成される。コンピュータは汎用コンピュータ、専用コンピュータ、コンピュータネットワーク、または他のプログラマブルな装置であってよい。コンピュータ命令は、コンピュータ可読記憶媒体に格納されてよく、または、コンピュータ可読記憶媒体から別のコンピュータ可読記憶媒体に伝送されてもよい。例えば、コンピュータ命令は、ウェブサイト、コンピュータ、サーバ、またはデータセンタから、別のウェブサイト、コンピュータ、サーバ、またはデータセンタに、有線（例えば、同軸ケーブル、光ファイバ、またはデジタル加入者線（Ｄｉｇｉｔａｌ　Ｓｕｂｓｃｒｉｂｅｒ　Ｌｉｎｅ，略してＤＳＬ））または無線（例えば、赤外線、無線、およびマイクロ波、または同様のものなど）方式で、伝送されてよい。コンピュータ可読記憶媒体は、コンピュータによってアクセス可能な任意の可用媒体、または、１または複数の可用媒体を統合する、サーバまたはデータセンタなどのデータストレージデバイスであってよい。可用媒体は、磁気媒体（例えば、フロッピー（登録商標）ディスク、ハードディスク、または磁気テープ）、光学媒体（例えば、ＤＶＤ）、半導体媒体（例えば、ソリッドステートドライブ（Ｓｏｌｉｄ　Ｓｔａｔｅ　Ｄｉｓｋ，略してＳＳＤ））、または同様のものであってよい。

【０１０７】

本願は、実施形態を参照して説明されてきたが、保護を請求する本願の実装の処理において、当業者は、添付の図面、開示された内容、および添付の特許請求の範囲を閲覧することによって、開示された実施形態の別の変形を理解および実装し得る。請求項において、「含む（ｃｏｍｐｒｉｓｉｎｇ）」は、別のコンポーネントまたは別の工程を排除せず、「１つの（ａ）」または「１つ（ｏｎｅ）」は、複数という意味を排除しない。単一のプロセッサまたは別のユニットが、請求項において列挙される複数の機能を実装してもよい。いくつかの測定値が互いに異なる従属請求項に記録されるが、これは、これらの測定値がより良い効果を作り出すように組み合わせられ得ないことを意味しない。

【０１０８】

本願は具体的特徴およびそれらの実施形態を参照して説明されてきたが、明らかに、様々な修正および組み合わせが、本願の思想および範囲から逸脱することなく、行われてよい。対応して、明細書および添付の図面は単に、添付の特許請求の範囲によって画定された本願の例としての説明にすぎず、本願の範囲を含める修正、変形、組み合わせ、または均等物の、いずれかまたはすべてとみなされる。明らかに、当業者は、本願の思想および範囲から逸脱することなく、本願の様々な修正および変形を行うことができる。以下の特許請求の範囲およびそれらの均等技術によって画定される保護の範囲内に属するならば、本願は、本願へのこれらの修正および変形を含めることを意図する。

【図１】

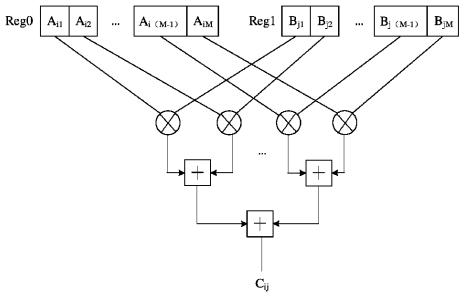
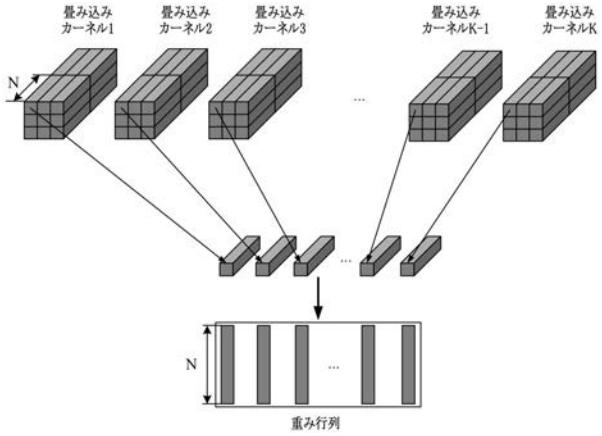
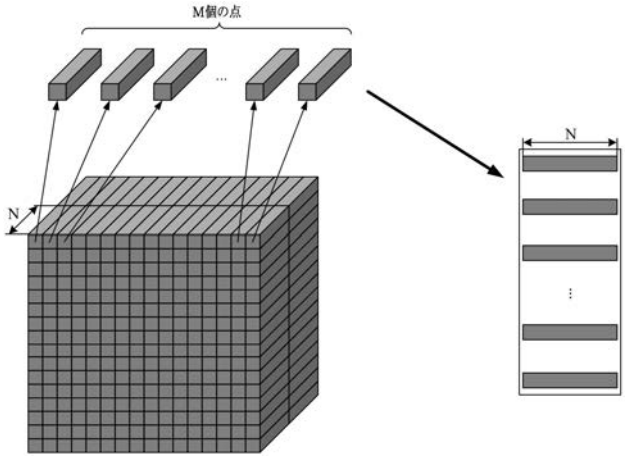


図 1

【図２】



【図３】

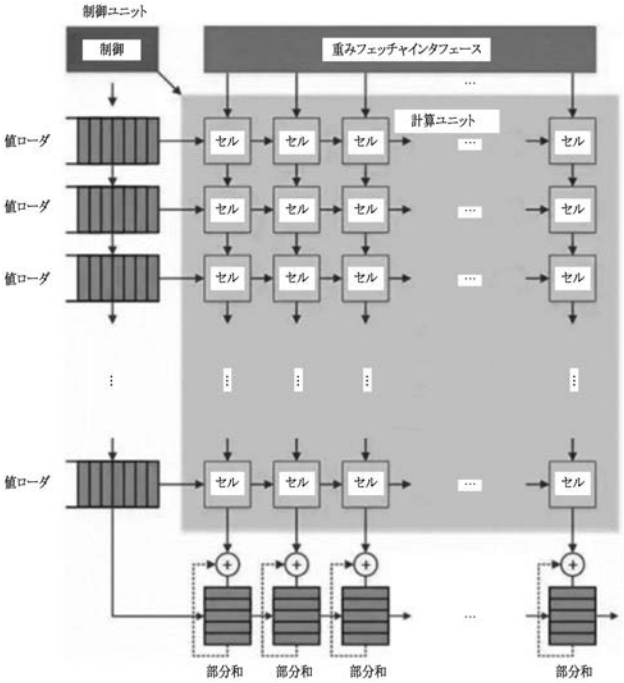


【図４】

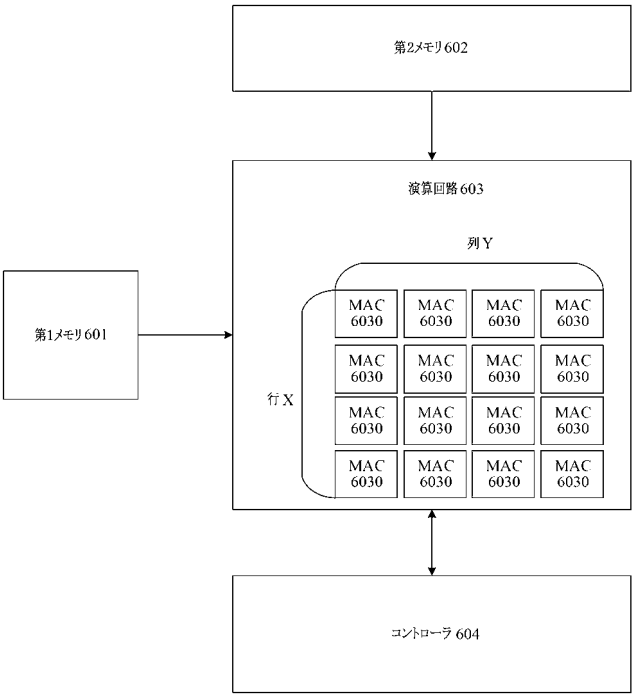
$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MN} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1K} \\ B_{21} & B_{22} & \cdots & B_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N1} & B_{N2} & \cdots & B_{NK} \end{bmatrix} = \begin{bmatrix} \sum_{j}^{N} A_{1j}B_{j1} & \sum_{j}^{N} A_{1j}B_{j2} & \cdots & \sum_{j}^{N} A_{1j}B_{jK} \\ \sum_{j}^{N} A_{2j}B_{j1} & \sum_{j}^{N} A_{2j}B_{j2} & \cdots & \sum_{j}^{N} A_{2j}B_{jK} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j}^{N} A_{Mj}B_{j1} & \sum_{j}^{N} A_{Mj}B_{j2} & \cdots & \sum_{j}^{N} A_{Mj}B_{jK} \end{bmatrix}$$

行列A　　行列B　　行列C

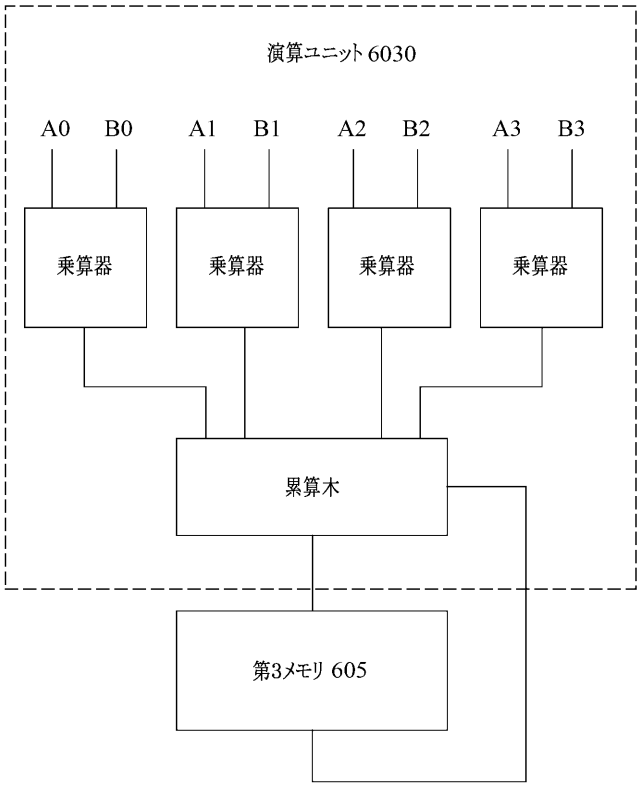
【図５】



【図６】



【図７】



【図８】

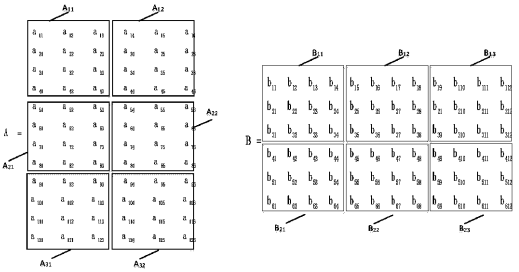


図 8

【図９】

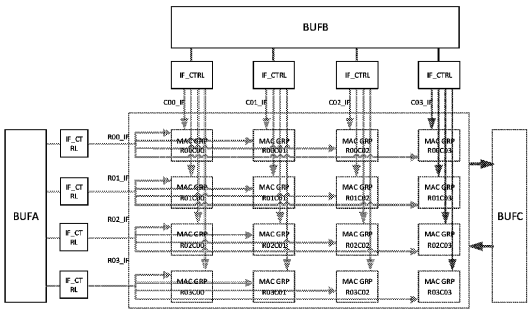


図 9

【図１０】

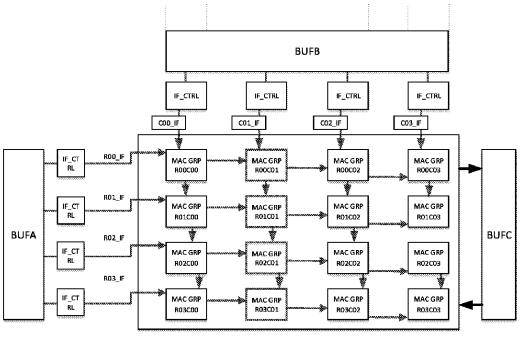


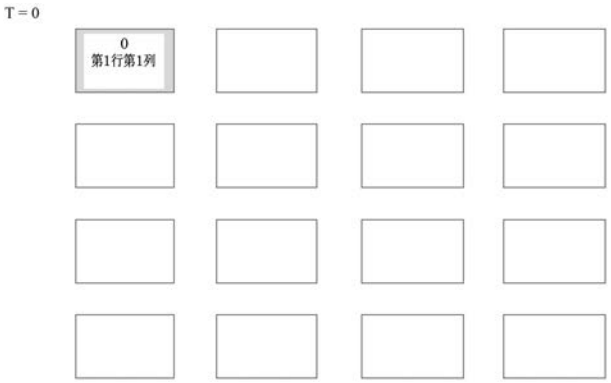
図 10

【図１１】

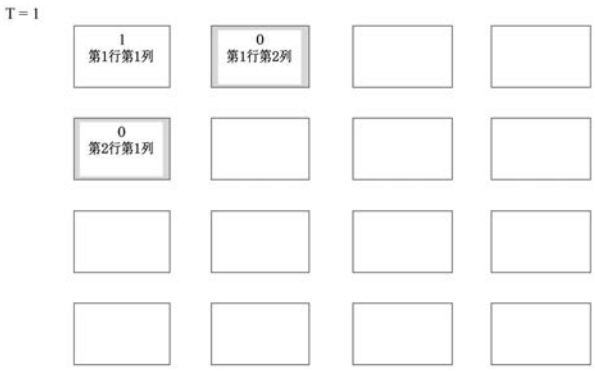
クロックサイクル	0	1	2	3	4	5	6
BUFA_0	[0,R0]	[1,R0]	[2,R0]	[3,R0]			
BUFA_1		[0,R1]	[1,R1]	[2,R1]	[3,R1]		
BUFA_2			[0,R2]	[1,R2]	[2,R2]	[3,R2]	
BUFA_3				[0,R3]	[1,R3]	[2,R3]	[3,R3]

クロックサイクル	0	1	2	3	4	5	6
BUFB_0	[0,C0]	[1,C0]	[2,C0]	[3,C0]			
BUFB_1		[0,C1]	[1,C1]	[2,C1]	[3,C1]		
BUFB_2			[0,C2]	[1,C2]	[2,C2]	[3,C2]	
BUFB_3				[0,C3]	[1,C3]	[2,C3]	[3,C3]

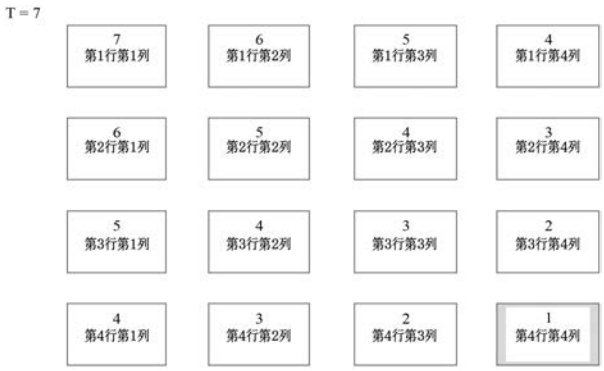
【図１２】



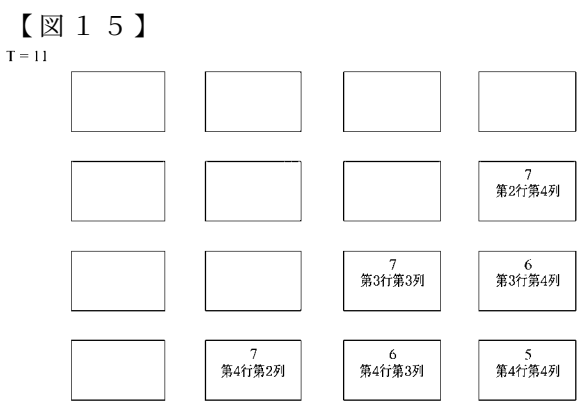
【図１３】



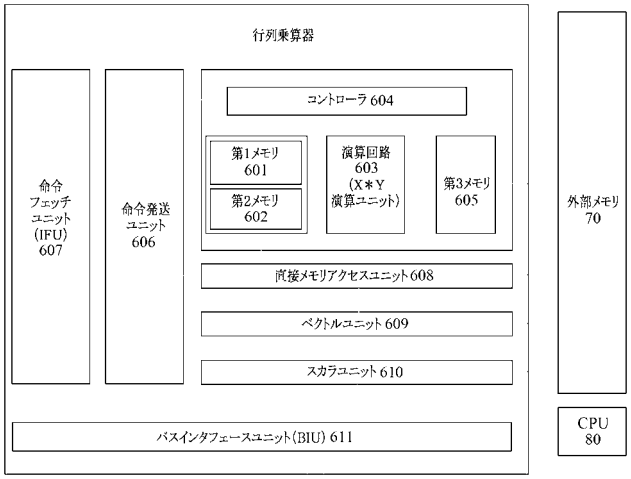
【図１４】



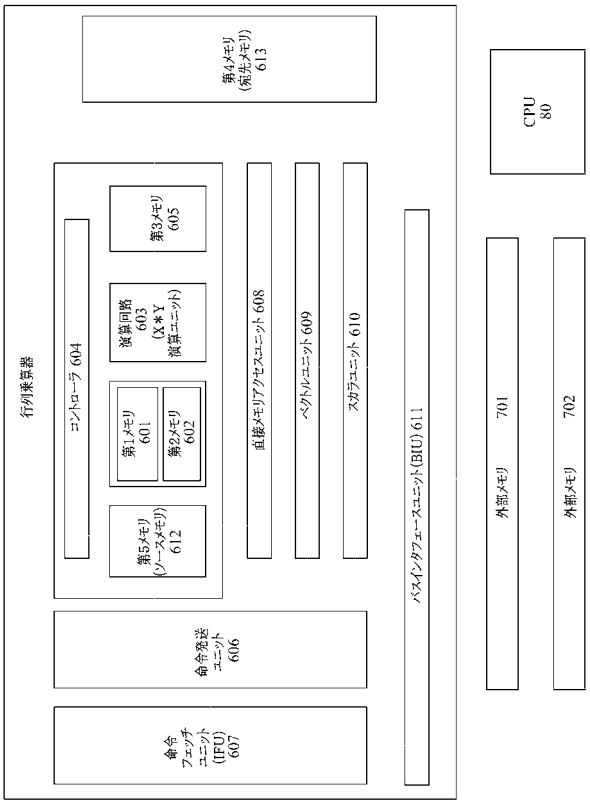
【図１５】



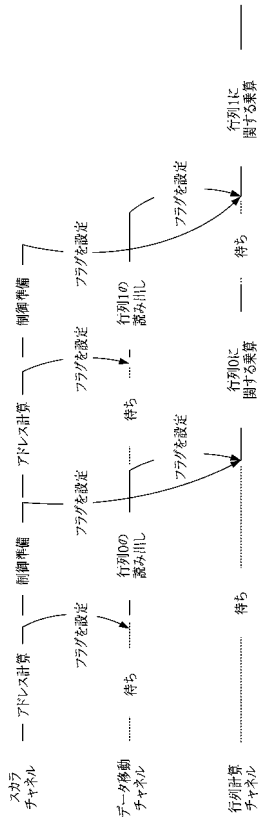
【図１６】



【図１７】



【図１８】



【手続補正書】
【提出日】令和2年8月5日(2020.8.5)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
Ｍ＊Ｋ行列である第１行列を格納するように構成された第１メモリと、
Ｋ＊Ｎ行列である第２行列を格納するように構成された第２メモリと、
前記第１メモリおよび前記第２メモリに接続される演算回路と、
前記演算回路に接続されるコントローラと、を含む、
行列乗算器であって、
前記演算回路は、Ｘ行＊Ｙ列からなる演算ユニットを含み、各前記演算ユニットは、ベクトル乗算回路および加算回路を含み、前記ベクトル乗算回路は、前記第１メモリによって送られる行ベクトルのデータおよび前記第２メモリによって送られる列ベクトルのデータを受信し、前記２つのベクトルを乗算するように構成され、前記加算回路は、前記２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、各演算ユニットの演算結果を取得するように構成され、
前記コントローラは、以下の動作、すなわち、
前記第１行列を、サイズがＸ＊Ｌであるサブブロックを単位とするブロックに分割し、同じサイズのＳ×Ｒ個のサブブロックを取得し、前記Ｓ×Ｒ個のサブブロックのうち第ｓ行第ｒ列におけるサブブロックはA_{sr}，ｓ＝（１，２，３，...，およびＳ）、およびｒ＝（１，２，３，...，およびＲ）で表される、動作と、
前記第２行列を、サイズがＬ＊Ｙであるサブブロックを単位とするブロックに分割し、同じサイズのＲ×Ｔ個のサブブロックを取得し、Ｒ×Ｔ個のサブブロックのうち第ｒ行第ｔ列におけるサブブロックは、B_{rt}，ｒ＝（１，２，３，...，およびＲ）、ｔ＝（１，２，３，...，およびＴ）で表される、動作と
を実行するように構成され、
前記コントローラは、さらに以下の動作、すなわち、
任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルにおける第ｙ列とを、Ｘ行＊Ｙ列からなる演算ユニットの第ｘ行第ｙ列において前記演算ユニットに入力し、それにより、処理を実行する動作を実行するように構成され、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，およびＹ）であり、前記任意のサブブロックA_{sr}におけるｒと、前記対応するサブブロックB_{rt}におけるｒとは同じ値を有する、
行列乗算器。
【請求項２】
前記コントローラは、以下の動作、すなわち、
前記任意のサブブロックA_{sr}の前記Ｘ個の行ベクトルにおける前記第ｘ行と、前記対応するサブブロックB_{rt}の前記Ｙ個の列ベクトルにおける前記第ｙ列とを、同じクロックサイクルにおいて並行して、Ｘ行＊Ｙ列からなる前記演算ユニットの第ｘ行第ｙ列において前記演算ユニットに入力し、それにより前記処理を実行する、動作
を実行するように具体的に構成される、請求項１に記載の行列乗算器。
【請求項３】
前記コントローラはさらに、前記任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる前記演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、前記コントローラはさらに、前記対

応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる前記演算ユニットに対応する第ｙ列に連続的に入力するように同時に制御するように構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである、

請求項１または２に記載の行列乗算器。

【請求項４】

前記コントローラはさらに、ｓおよびｒの値を変更されないままにして、ｔの値を、少なくとも２つの連続するサブブロック乗算計算サイクルにおいて変更されるように制御するように構成され、その結果、前記第１メモリは、前記少なくとも２つの連続するサブブロック乗算計算サイクル内で同じサブブロックA_{sr}を再使用し、前記サブブロック乗算計算サイクルは、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}上での行列乗算演算を完了させるように、Ｘ行＊Ｙ列の前記演算ユニットによって使用される時間である、

請求項１から３のいずれか一項に記載の行列乗算器。

【請求項５】

前記行列乗算器はさらに、前記演算回路に接続された第３メモリを含み、

前記コントローラは、前記ベクトル乗算回路および前記加算回路の演算結果を前記第３メモリに格納するように、Ｘ行＊Ｙ列の前記演算ユニットを制御するように構成される、

請求項１から４のいずれか一項に記載の行列乗算器。

【請求項６】

前記行列乗算器はさらに、前記第１メモリおよび前記第２メモリに接続される第４メモリと、前記第３メモリに接続される第５メモリとを含み、

前記コントローラはさらに、前記第１行列および前記第２行列の乗算演算を実行する前に、

前記第４メモリから、前記第１行列および前記第２行列のデータソースを、それぞれ前記第１メモリおよび前記第２メモリに移動させ、前記第３メモリから、前記計算結果を前記第５メモリに移動させるように、制御するように構成される、

請求項５に記載の行列乗算器。

【請求項７】

前記ベクトル乗算回路はＬ個の乗算器を含み、前記加算回路は入力数がＬ＋１である加算木を含む、

請求項１から６のいずれか一項に記載の行列乗算器。

【請求項８】

前記第１メモリ、前記第２メモリ、前記演算回路、および前記コントローラはバスインタフェースユニットを使用して接続される、

請求項１から７のいずれか一項に記載の行列乗算器。

【請求項９】

Ｓ＝

【数２２】

$$\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

および

Ｒ＝

【数２３】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、
Ｍ％Ｘ≠０のとき、計算は前記第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は前記第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられる、
請求項１から８のいずれか一項に記載の行列乗算器。
【請求項１０】
Ｒ＝
【数２４】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

および
Ｔ＝
【数２５】

$$\begin{cases} N/Y, N\%Y = 0 \\ [\frac{N}{Y}] + 1, N\%Y \neq 0 \end{cases}$$

であり、
Ｋ％Ｙ≠０のとき、計算は前記第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられ、Ｎ％Ｘ≠０のとき、計算は前記第１行列の第（Ｎ＋１）行から第（Ｔ＊Ｘ－Ｎ）行まで実行されず、結果の値には０が割り当てられる、
請求項１から８のいずれか一項に記載の行列乗算器。
【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】０００８
【補正方法】変更
【補正の内容】
【０００８】
第１の態様によると、本発明の実施形態は行列乗算器を提供し、行列乗算器は、
Ｍ＊Ｋ行列である第１行列を格納するように構成された第１メモリと、
Ｋ＊Ｎ行列である第２行列を格納するように構成された第２メモリと、
第１メモリおよび第２メモリに接続される演算回路と、
演算回路に接続されたコントローラと、を含み、
演算回路はＸ行＊Ｙ列からなる演算ユニットを含み、各演算ユニットはベクトル乗算回路および加算回路を含み、ベクトル乗算回路は、第１メモリによって送られる行ベクトルのデータおよび第２メモリによって送られる列ベクトルのデータを受信し、２つのベクトルを乗算するように構成され、加算回路は、２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、各演算ユニットの演算結果を取得するように構成される、演算回路と、
コントローラは以下の動作、すなわち、
第１行列を、サイズがＸ＊Ｌであるサブブロックを単位とするブロックに分割し、同じサイズのＳ×Ｒ個のサブブロックを取得し、Ｓ×Ｒ個のサブブロックのうち第ｓ行第ｒ列におけるサブブロックはA_{sr}，ｓ＝（１，２，３，...，およびＳ）、およびｒ＝（１，２，３，...，およびＲ）で表される、動作と、
第２行列を、サイズがＬ＊Ｙであるサブブロックを単位とするブロックに分割し、同じサイズのＲ×Ｔ個のサブブロックを取得し、Ｒ×Ｔ個のサブブロックのうち第ｒ行第ｔ列

におけるサブブロックは、B_{rt}，ｒ＝（１，２，３，...，およびＲ）、ｔ＝（１，２，３，...，およびＴ）で表される、動作とを実行するように構成され、

コントローラは、さらに以下の動作、すなわち、

任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルにおける第ｙ列とを、Ｘ行＊Ｙ列からなる演算ユニットの第ｘ行第ｙ列において演算ユニットに入力し、それにより、処理を実行する動作を実行するように構成され、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，およびＹ）であり、任意のサブブロックA_{sr}におけるｒと、対応するサブブロックB_{rt}におけるｒとは同じ値を有する。

【手続補正３】
【補正対象書類名】明細書
【補正対象項目名】００１１
【補正方法】変更
【補正の内容】
【００１１】

可能な実装において、コントローラはさらに、任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、コントローラはさらに、対応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｙ列に連続的に入力するように同時に制御するように構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである。

【手続補正４】
【補正対象書類名】明細書
【補正対象項目名】００１７
【補正方法】変更
【補正の内容】
【００１７】

可能な実装において、　Ｓ＝

【数１】

$$\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

およびＲ＝

【数２】

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、

Ｍ％Ｘ≠０のとき、計算は第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられる。

【手続補正５】
【補正対象書類名】明細書
【補正対象項目名】００５５
【補正方法】変更
【補正の内容】

【００５５】
前述の分析に基づいて、本願は、行列乗算アクセラレータを提供し、本願において提供される技術的問題を具体的に解析および解決する。図６は、本発明の実施形態による行列乗算器６０の構造図である。図６に示されるように、行列乗算器６０は第１メモリ６０１、第２メモリ６０２、演算回路６０３、およびコントローラ６０４を含む。演算回路６０３は、バスを使用して、第１メモリ６０１、第２メモリ６０２、およびコントローラ６０４とデータ通信を実行し得る。演算回路６０３は、第１メモリ６０１および第２メモリ６０２から行列データを取り出し、ベクトル乗算および加算演算を実行するように構成される。コントローラ６０４は、ベクトル演算を完了するように、予め設定されたプログラムまたは命令に従って、演算回路６０３を制御するように構成される。第１メモリ６０１は第１行列を格納するように構成される。
【手続補正６】
【補正対象書類名】明細書
【補正対象項目名】００５７
【補正方法】変更
【補正の内容】
【００５７】
本発明のこの実施形態において説明される第１メモリ６０１、および、以下に説明される関連する行列乗算器の第２メモリ６０２、第３メモリ６０５、および内部メモリは、それぞれ、レジスタ、ランダムアクセスメモリ（ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ、略してＲＡＭ）、静的ランダムアクセスメモリ、フラッシュメモリ、または別の読み出しおよび書き込み可能メモリであってよい。本願において、第１行列、第２行列および演算結果のデータ型はそれぞれ、ｉｎｔ８、ｆｐ１６、またはｆｐ３２などの型であってよい。
【手続補正７】
【補正対象書類名】明細書
【補正対象項目名】００６０
【補正方法】変更
【補正の内容】
【００６０】
演算回路６０３は、Ｘ行＊Ｙ列の演算ユニット６０３０（乗算累算ユニットＭＡＣと称されてよい）を含み得る。各演算ユニットは、独立してベクトル乗算演算を実行し得る。図６において、演算回路６０３が４＊４演算ユニット６０３０を含む例が図に使用され、すなわち、Ｘ＝４およびＹ＝４である。演算ユニット６０３０は、それぞれ第１メモリ６０１によって送られた行ベクトルと、第２メモリ６０２によって送られた列ベクトルとを受信し、行ベクトルと列ベクトルとのベクトル乗算演算を実行するように使用される、２つの入力を提供される。具体的には、１つの演算回路６０３０はベクトル乗算回路および加算回路を含み、ここで、ベクトル乗算回路は第１メモリ６０１によって送られる行ベクトルのデータと、第２メモリ６０２によって送られる列ベクトルのデータとを受信し、２つのベクトルを乗算するように構成され、加算回路は、２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、演算ユニット６０３０の演算結果を取得するように構成される。
【手続補正８】
【補正対象書類名】明細書
【補正対象項目名】００６４
【補正方法】変更
【補正の内容】
【００６４】
第１行列がＭ＊Ｋ行列であり、第１行列が整数個のＸ＊Ｌサブブロックで正確に分割できない場合が存在し得ることに、留意すべきである。したがって、Ｍ／ＸまたはＫ／Ｌが

整数でないとき、演算は要素０をパッディングする方式で実行されてよい。代替的に、対応する位置で全く計算が実行されず、結果の値に０が割り当てられる。具体的には、

【数５】

$$S=\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

および

【数６】

$$R=\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、Ｍ％Ｘ≠０のとき、計算は第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられる。言い換えれば、演算ユニットは対応する行および列において実体的乗算計算を実行せず、処理のために、演算が実行されたが結果が０であるとみなす。このようにして、対応する演算ユニットの読み出しおよび演算電力消費は低減され得る。

【手続補正９】
【補正対象書類名】明細書
【補正対象項目名】００７１
【補正方法】変更
【補正の内容】
【００７１】

方式１：行列乗算シーケンスにおいて、例えば、サブブロックはサブブロックA_{11}およびサブブロックB_{11}であってよい。A_{11}のすべての行ベクトルと、対応するB_{11}のすべての列ベクトルとが、サブブロック乗算計算サイクル（第１のラウンドとして理解されてよい）に入力され、それにより、演算を実行する。A_{12}のすべての行ベクトルおよび対応するB_{21}のすべての列ベクトルでの演算が、第２サブブロック乗算計算サイクル（第２のラウンドとして理解されてよい）において実行される。このようにして、演算ユニットが累算を実行した後、結果行列Ｃの第１行第１列における結果点C_{11}の値が取得され得る。同様に、結果行列Ｃのすべての位置における結果点が取得され得る。実際、C_{11}＝

【数１２】

$$A_{11}B_{11} + A_{12}B_{21}$$

ここで、

【数１３】

$$A_{11} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

【数１４】

$$A_{12}=\begin{bmatrix} a_{14} & a_{15} & a_{16} \\ a_{24} & a_{25} & a_{26} \\ a_{34} & a_{35} & a_{36} \\ a_{44} & a_{45} & a_{46} \end{bmatrix}$$

【数１５】

$$B_{11}=\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix}$$

【数１６】

$$B_{12}=\begin{bmatrix} b_{14} & b_{15} & b_{16} \\ b_{24} & b_{25} & b_{26} \\ b_{34} & b_{35} & b_{36} \\ b_{44} & b_{45} & b_{46} \end{bmatrix}$$

【手続補正１０】
【補正対象書類名】明細書
【補正対象項目名】００８１
【補正方法】変更
【補正の内容】
【００８１】
図１０は、本発明の実施形態による特定の演算回路６０３の配線の概略図である。演算回路６０３において、図１０に対応して、シストリックアレイ構造が提供される。具体的に、コントローラ６０４はさらに、任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、コントローラ６０４はさらに、対応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる演算ユニットに対応する第ｙ列に連続的に入力するように同時に制御するようにさらに構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである。
【手続補正１１】
【補正対象書類名】明細書
【補正対象項目名】００８３
【補正方法】変更
【補正の内容】
【００８３】
シストリックアレイアーキテクチャに基づいて、図１０に対応する配線構造において、

ＢＵＦＡ／Ｂが、それぞれ、第１行列／第２行列をバッファリングするように構成されるメモリである。図１０において、第１行列バッファ（ＢＵＦＡ）は、各クロックサイクルにおいて、第１行列における単位行列をＸ個の行に分割し、同じ行のＬ個の要素をシストリックアレイの演算ユニットに順次送る。同様に、第２行列バッファ（ＢＵＦＢ）は、各クロックサイクルにおいて、第２行列における単位行列をＹ個の列に分割し、同じ列のＬ個の要素をシストリックアレイに順次送る。具体的な時間シーケンスは以下の通りである。

【手続補正１２】
【補正対象書類名】明細書
【補正対象項目名】００８５
【補正方法】変更
【補正の内容】
【００８５】
Ｍ＝２、Ｎ＝２、およびＫ＝２（すなわち８×８＊８×８行列乗算が使用される例。行列乗算器６０のコントローラ６０４は、図１１のフォーマットで行列乗算を分割し、全体で８個の４×４単位行列演算を取得する。ＭＮＫ行列乗算演算に関して、分割シーケンスのための多くの可能性があり、分割シーケンスのルールは、ＭＮＫ行列乗算演算が、方式１または方式２のシーケンスにおいて実行されてよいことである。方式２において再使用するデータの最大数倍のポリシーを使用することによって、データを読み出すための消費電力を低減し得ることが、理解されてよい。ＭＮＫフラクタル分割が実行された後、図１２から図１５に示されるように、コントローラ６０４の制御ロジックは８クロックサイクルに８個のフラクタルをシストリックアレイに入力する。図１２は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝０におけるフラクタル行列乗算器のパイプライン実行を示し、図１３は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝１におけるフラクタル行列乗算器のパイプライン実行を示し、図１４は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝７におけるフラクタル行列乗算器のパイプライン実行を示し、図１５は、Ｍ＝２、Ｎ＝２およびＫ＝２のときの、時点Ｔ＝１１におけるフラクタル行列乗算器のパイプライン実行を示す。シストリックアレイは、Ｔ＝６のとき、すなわち、第７クロックサイクルにおいてフルロード状態で動作を開始することがわかる。最後の６クロックサイクルにおいて、単位行列がシストリックアレイから出力され、行列全体の乗算演算もまた完了する。

【手続補正１３】
【補正対象書類名】明細書
【補正対象項目名】００８９
【補正方法】変更
【補正の内容】
【００８９】
３．直接メモリアクセスユニット（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ　Ｕｎｉｔ，ＤＭＡ　Ｕｎｉｔ）は、各ストレージユニットにデータを移動するように、例えば、Ｌ１　ＲＡＭからＬ０　ＲＡＭへデータを移動するように構成される。本発明のこの実施形態における直接メモリアクセスユニットが、行列乗算器の外部メモリまたは内部メモリから、乗算演算に関与する行列データを移動するとき、直接メモリアクセスユニットは、行列がブロックに分割された後に取得された結果を格納する必要がある。例えば、２＊２行列に関して、第１行列の第１行第１列のサブブロックA_{11}＝
【数１９】

$$\begin{bmatrix} A0 & A1 \\ A2 & A3 \end{bmatrix}$$

がサブブロックのユニットに格納され、Ａ０、Ａ１、Ａ２およびＡ３が１つの行に格納

される、などである。このようにして、第１行列が、対応する第１メモリへと移動されてよいとき、または、第２行列が、対応する第２メモリに移動されてよいときは、ストレージは、前述の方式で実行されてよい。演算ユニットが読み出しの実行を必要とするとき、演算ユニットもまた、前述のストレージシーケンスにおいて読み出しを実行してよく、それにより、計算を円滑化する。行ベクトルが列ベクトルに転置される必要があるとき、転置はフレキシブルに、そして素早く実行されてよい。

【手続補正１４】
【補正対象書類名】明細書
【補正対象項目名】００９８
【補正方法】変更
【補正の内容】
【００９８】
時点１において、コントローラ６０４はＬ１バッファから行列のＡ０およびＢ０部分を読み出し、Ａ０およびＢ０部分をＬ０に格納する。

【手続補正１５】
【補正対象書類名】明細書
【補正対象項目名】０１０８
【補正方法】変更
【補正の内容】
【０１０８】
本願は具体的特徴およびそれらの実施形態を参照して説明されてきたが、明らかに、様々な修正および組み合わせが、本願の範囲から逸脱することなく、行われてよい。対応して、明細書および添付の図面は単に、添付の特許請求の範囲によって画定された本願の例としての説明にすぎず、本願の範囲を含める修正、変形、組み合わせ、または均等物の、いずれかまたはすべてとみなされる。明らかに、当業者は、本願の思想および範囲から逸脱することなく、本願の様々な修正および変形を行うことができる。以下の特許請求の範囲およびそれらの均等技術によって画定される保護の範囲内に属するならば、本願は、本願へのこれらの修正および変形を含めることを意図する。

（項目１）
Ｍ＊Ｋ行列である第１行列を格納するように構成された第１メモリと、
Ｋ＊Ｎ行列である第２行列を格納するように構成された第２メモリと、
上記第１メモリおよび上記第２メモリに接続される演算回路と、
上記演算回路に接続されるコントローラと、を含む、
行列乗算器であって、
上記演算回路は、Ｘ行＊Ｙ列からなる演算ユニットを含み、各上記演算ユニットは、ベクトル乗算回路および加算回路を含み、上記行列乗算回路は、上記第１メモリによって送られる行ベクトルのデータおよび上記第２メモリによって送られる列ベクトルのデータを受信し、上記２つのベクトルを乗算するように構成され、上記加算回路は、上記２つのベクトルの乗算によって取得された結果を加算し、同一の演算ユニットの計算結果を累算し、各演算ユニットの演算結果を取得するように構成され、
上記コントローラは、以下の動作、すなわち、
上記第１行列を、サイズがＸ＊Ｌであるサブブロックを単位とするブロックに分割し、同じサイズのＳ×Ｒ個のサブブロックを取得し、上記Ｓ×Ｒ個のサブブロックのうち第ｓ行第ｒ列におけるサブブロックはA_{sr}，ｓ＝（１，２，３，...，およびＳ）、およびｒ＝（１，２，３，...，およびＲ）で表される、動作と、
上記第２行列を、サイズがＬ＊Ｙであるサブブロックを単位とするブロックに分割し、同じサイズのＲ×Ｔ個のサブブロックを取得し、Ｒ×Ｔ個のサブブロックのうち第ｒ行第ｔ列におけるサブブロックは、B_{rt}，ｒ＝（１，２，３，...，およびＲ）、ｔ＝（１，２，３，...，およびＴ）で表される、動作と
を実行するように構成され、

上記コントローラは、さらに以下の動作、すなわち、
任意のサブブロックA_{sr}のＸ個の行ベクトルにおける第ｘ行と、対応するサブブロックB_{rt}のＹ個の列ベクトルにおける第ｙ列とを、Ｘ行＊Ｙ列からなる演算ユニットの第ｘ行第ｙ列において上記演算ユニットに入力し、それにより、処理を実行する動作を実行するように構成され、ｘ＝（１，２，３，...，およびＸ）、ｙ＝（１，２，３，...，およびＹ）であり、上記任意のサブブロックA_{sr}におけるｒと、上記対応するサブブロックB_{rt}におけるｒとは同じ値を有する、
行列乗算器。
（項目２）
上記コントローラは、以下の動作、すなわち、
上記任意のサブブロックA_{sr}の上記Ｘ個の行ベクトルにおける上記第ｘ行と、上記対応するサブブロックB_{rt}の上記Ｙ個の列ベクトルにおける上記第ｙ列とを、同じクロックサイクルにおいて並行して、Ｘ行＊Ｙ列からなる上記演算ユニットの第ｘ行第ｙ列において上記演算ユニットに入力し、それにより上記処理を実行する、動作
を実行するように具体的に構成される、項目１に記載の行列乗算器。
（項目３）
上記コントローラはさらに、上記任意のサブブロックA_{sr}の行ベクトルを、ｘ個の行番号の昇順で、Ｘ行＊Ｙ列からなる上記演算ユニットに対応する第ｘ行に連続的に入力するように制御するように構成され、近接する行ベクトルが同じ列で異なる行の演算ユニットに入る時点の間の差は１クロックサイクルであり、上記コントローラはさらに、上記対応するサブブロックB_{rt}の列ベクトルを、ｙ個の列番号の昇順で、Ｘ行＊Ｙ列からなる上記演算ユニットに対応する第ｙ行に連続的に入力するように同時に制御するように構成され、近接する列ベクトルが同じ行で異なる列の演算ユニットに入る時点の間の差は１クロックサイクルである、
項目１または２に記載の行列乗算器。
（項目４）
上記コントローラはさらに、ｓおよびｒの値を変更されないままにして、ｔの値を、少なくとも２つの連続するサブブロック乗算計算サイクルにおいて変更されるように制御するように構成され、その結果、上記第１メモリは、上記少なくとも２つの連続するサブブロック乗算計算サイクル内で同じサブブロックA_{sr}を再使用し、上記サブブロック乗算計算サイクルは、１つのサブブロックA_{sr}および対応するサブブロックB_{rt}上での行列乗算演算を完了させるように、Ｘ行＊Ｙ列の上記演算ユニットによって使用される時間である、
項目１から３のいずれか一項に記載の行列乗算器。
（項目５）
上記行列乗算器はさらに、上記演算回路に接続された第３メモリを含み、
上記コントローラは、上記ベクトル乗算回路および上記加算回路の演算結果を上記第３メモリに格納するように、Ｘ行＊Ｙ列の上記演算ユニットを制御するように構成される、
項目１から４のいずれか一項に記載の行列乗算器。
（項目６）
上記行列乗算器はさらに、上記第１メモリおよび上記第２メモリに接続される第４メモリと、上記第３メモリに接続される第５メモリとを含み、
上記コントローラはさらに、上記第１行列および上記第２行列の乗算演算を実行する前に、
上記第４メモリから、上記第１行列および上記第２行列のデータソースを、それぞれ上記第１メモリおよび上記第２メモリに移動させ、上記第３メモリから、上記計算結果を上記第５メモリに移動させるように、制御するように構成される、
項目５に記載の行列乗算器。
（項目７）
上記ベクトル乗算回路はＬ個の乗算器を含み、上記加算回路は入力数がＬ＋１である加

算木を含む、
項目１から６のいずれか一項に記載の行列乗算器。
（項目８）
上記第１メモリ、上記第２メモリ、上記演算回路、および上記コントローラはバスインタフェースユニットを使用して接続される、
項目１から７のいずれか一項に記載の行列乗算器。
（項目９）
Ｓ＝
（数２２）

$$\begin{cases} M/X, M\%X = 0 \\ [\frac{M}{X}] + 1, M\%X \neq 0 \end{cases}$$

および
Ｒ＝
（数２３）

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

であり、
Ｍ％Ｘ≠０のとき、計算は上記第１行列の第（Ｍ＋１）行から第（Ｓ＊Ｘ－Ｍ）行まで実行されず、結果の値には０が割り当てられ、Ｋ％Ｙ≠０のとき、計算は上記第１行列の第（Ｋ＋１）行から第（Ｒ＊Ｙ－Ｋ）行まで実行されず、結果の値には０が割り当てられる、
項目１から８のいずれか一項に記載の行列乗算器。
（項目１０）
Ｒ＝
（数２４）

$$\begin{cases} K/L, K\%L = 0 \\ [\frac{K}{L}] + 1, K\%L \neq 0 \end{cases}$$

および
Ｔ＝
（数２５）

$$\begin{cases} N/Y, N\%Y = 0 \\ [\frac{N}{Y}] + 1, N\%Y \neq 0 \end{cases}$$

であり、
Ｋ％Ｙ≠０のとき、計算は上記第１行列の第（Ｋ＋１）列から第（Ｒ＊Ｙ－Ｋ）列まで実行されず、結果の値には０が割り当てられ、Ｎ％Ｘ≠０のとき、計算は上記第１行列の第（Ｎ＋１）行から第（Ｔ＊Ｘ－Ｎ）行まで実行されず、結果の値には０が割り当てられる、
項目１から８のいずれか一項に記載の行列乗算器。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2018/111077

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/16(2006.01)i

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)

G06F17; G06F7

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)

WPI, EPODOC, CNPAT, CNKI: 矩阵乘法器, 存储器, 计算, MATRIX MULTIPLICATION DEVICE, MEMORY, CALCULATING

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 101086699 A (ZHEJIANG UNIVERSITY) 12 December 2007 (2007-12-12) entire document	1-10
A	CN 107315574 A (BEIJING CAMBRICON TECHNOLOGY CO., LTD.) 03 November 2017 (2017-11-03) entire document	1-10
A	CN 103902509 A (RONGCHENG DINGTONG ELECTRONIC INFORMATION TECHNOLOGY CO., LTD.) 02 July 2014 (2014-07-02) entire document	1-10

☐ 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
22 November 2018	**06 December 2018**

Name and mailing address of the ISA/CN	Authorized officer
State Intellectual Property Office of the P. R. China (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2018/111077

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	101086699	A	12 December 2007	CN	100465876	C	04 March 2009
CN	107315574	A	03 November 2017	WO	2017185389	A1	02 November 2017
CN	103902509	A	02 July 2014	None			

Form PCT/ISA/210 (patent family annex) (January 2015)

国际检索报告

国际申请号
PCT/CN2018/111077

A. **主题的分类**

G06F 17/16(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. **检索领域**

检索的最低限度文献(标明分类系统和分类号)

G06F17; G06F7

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

WPI, EPODOC, CNPAT, CNKI：矩阵乘法器，存储器，计算，MATRIX MULTIPLICATION DEVICE,MEMORY,CALCULATING

C. **相关文件**

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 101086699 A (浙江大学) 2007年 12月 12日 (2007-12-12) 全文	1-10
A	CN 107315574 A (北京中科寒武纪科技有限公司) 2017年 11月 3日 (2017-11-03) 全文	1-10
A	CN 103902509 A (荣成市鼎通电子信息科技有限公司) 2014年 7月 2日 (2014-07-02) 全文	1-10

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2018年 11月 22日	2018年 12月 6日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 林亮亮 电话号码 86-(010)-62411890

表 PCT/ISA/210 (第2页) (2015年1月)

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2018/111077

检索报告引用的专利文件			公布日（年/月/日）	同族专利			公布日（年/月/日）
CN	101086699	A	2007年 12月 12日	CN	100465876	C	2009年 3月 4日
CN	107315574	A	2017年 11月 3日	WO	2017185389	A1	2017年 11月 2日
CN	103902509	A	2014年 7月 2日		无		

表 PCT/ISA/210（同族专利附件）（2015年1月）

フロントページの続き

(81)指定国・地域 AP(BW,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,MC,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,CO,CR,CU,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,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,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT

(72)発明者 リウ、フ
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング ホアウェイ・テクノロジーズ・カンパニー・リミテッド内
(72)発明者 リアオ、ヘン
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング ホアウェイ・テクノロジーズ・カンパニー・リミテッド内
(72)発明者 トゥ、ジアジン
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング ホアウェイ・テクノロジーズ・カンパニー・リミテッド内
(72)発明者 ユアン、ホンフイ
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング ホアウェイ・テクノロジーズ・カンパニー・リミテッド内
(72)発明者 ラム、ホウファン
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング ホアウェイ・テクノロジーズ・カンパニー・リミテッド内
(72)発明者 チュー、ファン
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング ホアウェイ・テクノロジーズ・カンパニー・リミテッド内

Ｆターム(参考) 5B056 AA05 BB71