

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2020年6月11日(11.06.2020)



(10) 国際公開番号

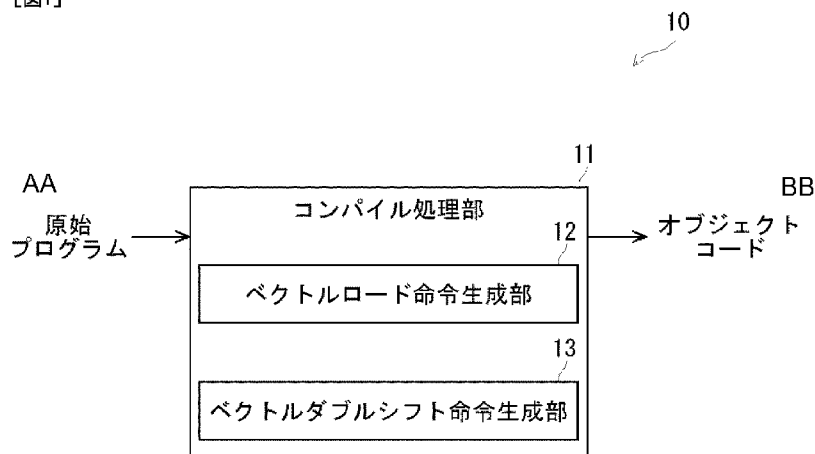
WO 2020/116025 A1

- (51) 国際特許分類:
G06F 8/41 (2018.01) G06F 17/16 (2006.01)
- (21) 国際出願番号: PCT/JP2019/040150
- (22) 国際出願日: 2019年10月11日(11.10.2019)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2018-229695 2018年12月7日(07.12.2018) JP
- (71) 出願人: 日本電気株式会社 (NEC CORPORATION) [JP/JP]; 〒1088001 東京都港区芝五丁目7番1号 Tokyo (JP).
- (72) 発明者: 増田 晃一(MASUDA Koichi); 〒1088001 東京都港区芝五丁目7番1号 日本電気株式会社内 Tokyo (JP).
- (74) 代理人: 家入 健(IEIRI Takeshi); 〒2210835 神奈川県横浜市神奈川区鶴屋町三丁目3番8 アサヒビルディング5階 響国際特許事務所 Kanagawa (JP).
- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): 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,

(54) Title: COMPILE DEVICE, COMPILE METHOD, AND NON-TRANSITORY COMPUTER-READABLE MEDIUM

(54) 発明の名称: コンパイル装置、コンパイル方法、及び非一時的なコンピュータ可読媒体

[図1]



- 11 Compile processing unit
12 Vector load command generating unit
13 Vector double shift command generating unit
AA Source program
BB Object code

(57) Abstract: A vector load command generating unit (12) of a compile device (10) generates a command for loading, from a memory into a first vector register, a "first data unit group", which is used as an element $a[i]$ during an iterative computation process, in a state of being packed on a word-by-word basis. Each data unit comprises $(1/2)^k$ words. The vector load command generating unit (12) generates a command for loading a second data unit group, which is used as an element $[i+2^k]$, into a second vector register. A vector shift double command generating

[続葉有]



WO 2020/116025 A1

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, TZ, UA,
UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (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), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類：

- 一 国際調査報告 (条約第21条(3))

unit (13) generates a command for causing a part of a data string, obtained by shifting a series of data string comprising data in the first vector register and the second vector register by $(1/2)^k$ words, to be stored in a third vector register in a state of being packed on a word-by-word basis.

(57) 要約：コンパイル装置 (10) のベクトルロード命令生成部 (12) は、繰り返し計算処理にて要素 a [i] として用いられる「第1データユニット群」をメモリから1ワード単位でパックした状態で第1ベクトルレジスタにロードする命令を生成する。各データユニットは、 $(1/2)^k$ ワードである。ベクトルロード命令生成部 (12) は、要素 [i + 2^k] として用いられる第2データユニット群を第2ベクトルレジスタにロードする命令を生成する。ベクトルシフトダブル命令生成部 (13) は、第1ベクトルレジスタ及び第2ベクトルレジスタのデータを一連のデータ列として $(1/2)^k$ ワード分シフトさせて得られたデータ列の一部を、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせる命令を生成する。

明 細 書

発明の名称：

コンパイル装置、コンパイル方法、及び非一時的なコンピュータ可読媒体
技術分野

[0001] 本開示は、コンパイル装置、コンパイル方法、及び制御プログラムに関する。

背景技術

[0002] ソースプログラム内に隣接するメモリアクセス命令が存在するときに重複したメモリアクセスを削減したオブジェクトコードに変換することによって、ベクトル計算機の実行性能を向上させる、コンパイル装置が提案されている（例えば、特許文献1）。特許文献1に開示されているコンパイル装置は、隣接するメモリアクセス命令を1つのメモリアクセス命令に変換する。さらに、該コンパイル装置は、該1つのメモリアクセス命令によってロードされた第1ベクトルレジスタのデータを、スライドさせて第2ベクトルレジスタに格納させる命令を生成する。そして、該コンパイル装置は、第1ベクトルレジスタのデータと第2のベクトルレジスタのデータとの演算を行う命令を生成する。

先行技術文献

特許文献

[0003] 特許文献1：特開2000-48009号公報

発明の概要

発明が解決しようとする課題

[0004] しかしながら、特許文献1のコンパイラ装置は、第1ベクトルレジスタのデータをスライドさせて第2ベクトルレジスタに格納させる命令を生成しているため、演算に用いる要素が不足する可能性がある（特許文献1の図4参照）。この場合、特許文献1のコンパイラ装置は、不足した要素を補うためロード/ストアや演算のための追加のベクトル命令、または、スカラ命令を生

成する必要がある。すなわち、特許文献1のコンパイラ装置は、ソースプログラムの最適化が不十分であり、プログラムを十分に高速化できていない可能性がある。

[0005] 本開示の目的は、プログラムの高速化を実現することができる、コンパイラ装置、コンパイル方法、及び制御プログラムを提供することにある。

課題を解決するための手段

[0006] 第1の態様にかかるコンパイラ装置は、 $(1/2)^k$ ワード型 (k は自然数)の配列Aの要素A[i]、要素A[i+1]、及び要素A[i+2^k]をオペランドとして含む算術計算を実行する計算処理を添え字*i* (*i*は0以上の整数)の値を1つずつずらしながら繰り返し実行するための原始プログラムを、オブジェクトコードに変換するコンパイル処理部を具備し、前記コンパイル処理部は、前記繰り返される計算処理において前記要素A[i]として用いられ且つ各データユニットが $(1/2)^k$ ワードである第1のデータユニット群をメモリから1ワード単位でパックした状態で第1ベクトルレジスタに第1ベクトルデータとしてロードするための第1ベクトルロード命令、及び、前記繰り返される計算処理において前記要素A[i+2^k]として用いられる第2のデータユニット群を前記メモリから1ワード単位でパックした状態で第2ベクトルレジスタに第2ベクトルデータとしてロードするための第2ベクトルロード命令を生成するベクトルロード命令生成部と、前記第1ベクトルデータ及び前記第2ベクトルデータを一連のデータ列として $(1/2)^k$ ワード分シフトさせて得られたデータ列の一部を、前記要素A[i+1]に対応する第3ベクトルデータとして、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令を生成するベクトルシフト命令生成部と、を含む。

[0007] 第2の態様にかかるコンパイル方法は、 $(1/2)^k$ ワード型 (k は自然数)の配列Aの要素A[i]、要素A[i+1]、及び要素A[i+2^k]をオペランドとして含む算術計算を実行する計算処理を添え字*i* (*i*は0以上の整数)の値を1つずつずらしながら繰り返し実行するための原始プログラム

を、オブジェクトコードに変換するコンパイル方法であって、前記繰り返される計算処理において前記要素A [i]として用いられ且つ各データユニットが $(1/2)^k$ ワードである第1のデータユニット群をメモリから1ワード単位でパックした状態で第1ベクトルレジスタに第1ベクトルデータとしてロードするための第1ベクトルロード命令、及び、前記繰り返される計算処理において前記要素A [i + 2^k]として用いられる第2のデータユニット群を前記メモリから1ワード単位でパックした状態で第2ベクトルレジスタに第2ベクトルデータとしてロードするための第2ベクトルロード命令を生成すること、及び、前記第1ベクトルデータ及び前記第2ベクトルデータを一連のデータ列として $(1/2)^k$ ワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 1]に対応する第3ベクトルデータとして、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令を生成すること、を含む。

[0008] 第3の態様にかかる制御プログラムは、 $(1/2)^k$ ワード型(kは自然数)の配列Aの要素A [i]、要素A [i + 1]、及び要素A [i + 2^k]をオペランドとして含む算術計算を実行する計算処理を添え字i(iは0以上の整数)の値を1つずつずらしながら繰り返し実行するための原始プログラムを、オブジェクトコードに変換するコンパイル処理をコンパイル装置に実行させる制御プログラムであって、前記コンパイル処理は、前記繰り返される計算処理において前記要素A [i]として用いられ且つ各データユニットが $(1/2)^k$ ワードである第1のデータユニット群をメモリから1ワード単位でパックした状態で第1ベクトルレジスタに第1ベクトルデータとしてロードするための第1ベクトルロード命令、及び、前記繰り返される計算処理において前記要素A [i + 2^k]として用いられる第2のデータユニット群を前記メモリから1ワード単位でパックした状態で第2ベクトルレジスタに第2ベクトルデータとしてロードするための第2ベクトルロード命令を生成すること、及び、前記第1ベクトルデータ及び前記第2ベクトルデータを一連のデータ列として $(1/2)^k$ ワード分シフトさせて得られたデータ列の一部を

、前記要素A [$i + 1$] に対応する第3ベクトルデータとして、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令を生成すること、を含む。

発明の効果

[0009] 本開示により、プログラムの高速化を実現することができる、コンパイル装置、コンパイル方法、及び制御プログラムを提供することができる。

図面の簡単な説明

- [0010] [図1]第1実施形態のコンパイル装置の一例を示すブロック図である。
- [図2]第2実施形態のコンパイル装置の一例を示すブロック図である。
- [図3]第2実施形態の原始プログラムの一例を示す図である。
- [図4]第2実施形態のコンパイル装置による、ベクトル化技法を適用可能か否かについての解析処理の一例を示すフローチャートである。
- [図5]第2実施形態のコンパイル装置によるベクトル化技法の一例を示すフローチャートである。
- [図6]第2実施形態のベクトル化技法によって生成されるオブジェクトコードの一例を示す図である。
- [図7]第2実施形態のベクトル化技法にて生成されるベクトルロード命令の説明に供する図である。
- [図8]第2実施形態のベクトル化技法にて生成されるベクトルシフトダブル命令の説明に供する図である。
- [図9]第2実施形態のベクトル化技法にて生成されるパックドベクトル演算命令の説明に供する図である。
- [図10]1 / 4ワード型の場合に、第2実施形態のベクトル化技法によって生成されるオブジェクトコードの一例を示す図である。
- [図11]1 / 4ワード型の場合に、第2実施形態のベクトル化技法にて生成されるベクトルロード命令の説明に供する図である。
- [図12]1 / 4ワード型の場合に、第2実施形態のベクトル化技法にて生成されるベクトルシフトダブル命令の説明に供する図である。

[図13] 1 / 4ワード型の場合に、第2実施形態のベクトル化技法にて生成されるパックドベクトル演算命令の説明に供する図である。

[図14] コンパイル装置のハードウェア構成例を示す図である。

発明を実施するための形態

[0011] 以下、図面を参照しつつ、実施形態について説明する。なお、実施形態において、同一又は同等の要素には、同一の符号を付し、重複する説明は省略される。

[0012] <第1実施形態>

図1は、第1実施形態のコンパイル装置の一例を示すブロック図である。図1においてコンパイル装置10は、コンパイル処理部11を有している。コンパイル処理部11は、原始プログラム（ソースプログラム）を、オブジェクトコードに変換する。この原始プログラムは、 $(1/2)^k$ ワード型の配列aの要素a[i]、要素a[i+1]、及び要素a[i+2^k]をオペランドとして含む算術計算を実行する「計算処理」を添え字iの値を1つずつずらしながら繰り返し実行するためのプログラムである。kは自然数であり、添え字iは0以上の整数である。上記の算術計算は、要素a[i]と要素a[i+1]とが配列内で隣接しているので、配列の「隣接要素計算」を含んでいる。以下では、上記の繰り返し実行される計算処理を、「繰り返し計算処理」と呼ぶことがある。

[0013] コンパイル処理部11は、第1実施形態の「ベクトル化技法」を実行する。コンパイル処理部11は、ベクトルロード命令生成部12と、ベクトルシフトダブル命令生成部13とを含む。

[0014] ベクトルロード命令生成部12は、繰り返し計算処理にて要素a[i]として用いられる「第1データユニット群」をメモリ（図示せず）から1ワード単位でパックした状態で第1ベクトルレジスタにロードするための第1ベクトルロード命令を生成する。各データユニットは、 $(1/2)^k$ ワードである。さらに、ベクトルロード命令生成部12は、繰り返し計算処理にて要素a[i+2^k]として用いられる第2データユニット群をメモリ（図示せず）か

ら1ワード単位でパックした状態で第2ベクトルレジスタにロードするための第2ベクトルロード命令を生成する。以下では、上記の「第1データユニット群」及び「第2データユニット群」を、それぞれ、「第1ベクトルデータ」及び「第2ベクトルデータ」と呼ぶことがある。

[0015] ベクトルシフトダブル命令生成部13は、第1ベクトルデータ及び第2ベクトルデータを一連のデータ列として $(1/2) \times$ ワード分シフトさせて得られたデータ列の一部を、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせるベクトルシフトダブル命令を生成する。第3ベクトルレジスタにストアされる上記データ列の一部は、要素 $a[i+1]$ に対応する「第3ベクトルデータ」である。

[0016] 以上のように第1実施形態によれば、コンパイル装置10は、コンパイル処理部11を有している。コンパイル処理部11にてベクトルロード命令生成部12は、繰り返し計算処理にて要素 $a[i]$ として用いられる「第1データユニット群」をメモリ（図示せず）から1ワード単位でパックした状態で第1ベクトルレジスタにロードするための第1ベクトルロード命令を生成する。各データユニットは、 $(1/2) \times$ ワードである。さらに、ベクトルロード命令生成部12は、繰り返し計算処理にて要素 $[i+2^k]$ として用いられる第2データユニット群をメモリ（図示せず）から1ワード単位でパックした状態で第2ベクトルレジスタにロードするための第2ベクトルロード命令を生成する。

[0017] このコンパイル装置10の構成により、第1ベクトルレジスタ及び第2ベクトルレジスタにおける1ワード単位の各レジスタ領域を、パックされた2つのデータユニットによって埋めることができるので、レジスタのリソース利用効率を向上させることができる。

[0018] また、コンパイル処理部11にてベクトルシフトダブル命令生成部13は、第1ベクトルデータ及び第2ベクトルデータを一連のデータ列として $(1/2) \times$ ワード分シフトさせて得られたデータ列の一部を、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせるベクトルシフトダブル

命令を生成する。

[0019] このコンパイル装置 10 の構成により、第 1 ベクトルデータ及び第 2 ベクトルデータのそれぞれに含まれるデータユニットと同数のデータユニットを含む第 3 ベクトルデータを生成することができる。これにより、第 1 ベクトルデータ、第 2 ベクトルデータ、及び第 3 ベクトルデータにおいて、演算に用いる要素に不足が生じることがない。このため、上記の特許文献 1 のような不足した要素をロードする命令及び該命令によってロードされたデータを演算するスカラ命令を生成する必要がない。この結果として、プログラムの高速化を実現することができる。

[0020] <第 2 実施形態>

第 2 実施形態は、より具体的な実施形態に関する。

[0021] <コンパイル装置の構成例>

図 2 は、第 2 実施形態のコンパイル装置の一例を示すブロック図である。図 2 においてコンパイル装置 20 は、コンパイル処理部 21 を有する。コンパイル処理部 21 は、第 1 実施形態のコンパイル処理部 11 と同様に、原始プログラム（ソースプログラム）を、オブジェクトコードに変換する。コンパイル処理部 11 は、解析部 22 と、ベクトル化実行部 23 とを有している。

[0022] 解析部 22 は、ループ検出部 22 A と、ベクトル化可否判定部 22 B と、最適化適用可否判定部 22 C とを有している。

[0023] ループ検出部 22 A は、原始プログラムにおけるループを検出し、検出されたループ内に上記の様な「計算処理」が含まれているか否かを判定する。

[0024] ベクトル化可否判定部 22 B は、ループ検出部 22 A にて検出されたループ内に上記の様な「計算処理」が含まれる場合、ループをベクトル化できるか否かを判定する。「ベクトル化できる条件」としては、例えば、従来の条件、つまり、「ループ内の配列及び変数についての定義及び参照関係にベクトル化を阻害する依存関係がないこと」を用いることができる。

[0025] 最適化適用可否判定部 22 C は、上記の「計算処理」に対して、第 2 実施

形態の「ベクトル化技法」を適用できるか否かを判定する。例えば、最適化適用可否判定部 2 2 C は、次の第 1 条件、第 2 条件、及び第 3 条件のすべてを満たす場合、上記の「計算処理」に対して、第 2 実施形態の「ベクトル化技法」を適用できると判定する。

(第 1 条件) 配列が $(1/2)^k$ ワード型であること。

(第 2 条件) 1 回の繰り返しにおいて、つまり、上記「計算処理」において、計算対象となる配列要素がすべて隣接していること。

(第 3 条件) 1 回の繰り返しにおいて、つまり、上記「計算処理」において、計算に用いられる配列要素の数が「 $m \times 2^k + 1$ 」個であること。ただし、 m は、任意の自然数である。

[0026] 原始プログラムが例えば図 3 に示すプログラムである場合には、ループ検出部 2 2 A 及びベクトル化可否判定部 2 2 B の判定は、肯定的な判定となる。さらに、原始プログラムが例えば図 1 に示すようなプログラムである場合には、最適化適用可否判定部 2 2 C の判定も、肯定的な判定となる。すなわち、図 3 に示すプログラムは、配列が半ワード型 ($k = 1$) であり、計算対象である配列要素の添え字が、 $[i]$ 、 $[i + 1]$ 、 $[i + 2]$ 、 $[i + 3]$ 、 $[i + 4]$ と 1 つずつずれている。さらに、図 3 に示すプログラムは、1 回の繰り返しにおいて計算に用いられる配列要素の数が 5 ($= 2 \times 2^1 + 1$) 要素である。このため、図 3 に示すプログラムは、上記の第 1 条件、第 2 条件、及び第 3 条件のすべてを満たす。図 3 は、第 2 実施形態の原始プログラムの一例を示す図である。

[0027] ベクトル化実行部 2 3 は、ループ検出部 2 2 A、ベクトル化可否判定部 2 2 B、及び最適化適用可否判定部 2 2 C にて肯定的な判定がなされた原始プログラムに対して、第 2 実施形態の「ベクトル化技法」を実行する。

[0028] ベクトル化実行部 2 3 は、ベクトル長算出命令生成部 2 3 A と、ベクトルロード命令生成部 1 2 と、ベクトルシフトダブル命令生成部 1 3 と、パックドベクトル演算命令生成部 2 3 B と、ベクトルストア命令生成部 2 3 C とを有している。

[0029] ベクトル長算出命令生成部23Aは、「ループ長N」を取得して $N \times (1/2)^k$ を右論理シフト演算によって計算するための、SRL(Shift Right Logical)命令を生成する。また、ベクトル長算出命令生成部23Aは、システムにて許容されている「最大ベクトル長」を取得するための、SMVL(Store Max Vector Length)命令を生成する。さらに、ベクトル長算出命令生成部23Aは、SRL命令によって計算した値と「最大ベクトル長」とを比較して小さい方を選択するための、MIN(Minimum)命令、及び、選択された値を「ベクトル長」とするための、LVL(Load Vector Length)命令を生成する。図3のプログラムの場合、ベクトル長算出命令生成部23Aは、ループ長nを取得し、 $n \times (1/2)^1$ と最大ベクトル長とを比較して小さい方をベクトル長とする命令群を生成することになる。

[0030] ベクトルロード命令生成部12は、 2^k 個のデータユニット(つまり、1ワードのデータユニット)を1つのパックドデータとしてそれぞれ異なるベクトルレジスタに1ワードベクトルロードするための、 $(m+1)$ 個のVLD(Vector Load)命令を生成する。これらの $(m+1)$ 個のVLD命令は、メモリ(図示せず)からデータユニットのロードを開始する位置、つまり、「ベクトルロード開始位置」が互いに 2^k 要素(つまり、1ワード分の要素数)ずつずれている。

[0031] ベクトルシフトダブル命令生成部13は、「ベクトルロード開始位置」が最も近いデータユニット群の各ペアに対して「所定シフト量」分シフトするための、 $(2^k - 1)$ 個のVSRD(Vector Shift Double Right)命令を生成する。上記の「ベクトルロード開始位置」が最も近いデータユニット群の各ペアは、ベクトルロード開始位置が1ワード分の要素数だけずれた、データユニット群の各ペアである。 $(2^k - 1)$ 個のVSRD命令についての「所定シフト量」は、それぞれ、 $l \times (1/2)^k$ ワード(l は、1から $(2^k - 1)$ までの自然数)である。

[0032] VSRD命令は、次のような命令である。

「VSRD %v0, %v1, %v2, S」

ここで、%v 0は、演算結果を書き込む1ワードベクトルレジスタを意味する。%v 1, %v 2は、演算のオペランドとなるベクトルレジスタを意味する。Sは、シフト量を指定する0から(2ワード-1)までのスカラ値である。すなわち、VSRD命令によれば、オペランドである2つのベクトルレジスタを合わせて2ワードデータ(つまり一連のデータ列)とし、該2ワードデータがSだけ右シフト演算され、得られた2ワードデータの下位側の1ワードの値が演算結果としてベクトルレジスタに書き込まれる。

[0033] パックドベクトル演算命令生成部23Bは、ベクトルロードされたベクトルデータ及びベクトルシフトダブルによって生成されたベクトルデータに対して演算を行うための、PPOP(Packed Vector Operation)命令を生成する。ここで、具体的には、生成されるPPOP命令は、上記の算術計算の態様に応じた命令となる。例えば、図3に示す原始プログラムの例では、算術計算が要素間の加算であるので、生成されるPPOP命令は、具体的には、PVADD(Packed Vector ADD)命令となる。

[0034] ベクトルストア命令生成部23Cは、PPOP命令による演算結果を、2^k要素のパックドデータとして1ワードベクトルストアするための、VST(Vector Store)命令を生成する。

[0035] <コンパイル装置の動作例>

以上の構成を有するコンパイル装置20の処理動作の一例について説明する。

[0036] <第2実施形態のベクトル化技法を適用可能か否かについての解析処理>

図4は、第2実施形態のコンパイル装置による、ベクトル化技法を適用可能か否かについての解析処理の一例を示すフローチャートである。

[0037] コンパイル装置20において解析部22は、原始プログラムにループが存在するか否かを判定する(ステップS101)。

[0038] 原始プログラムにループが存在する場合(ステップS101YES)、解析部22は、検出したループ内に計算処理が含まれているか否かを判定する(ステップS102)。

- [0039] ループ内に計算処理が含まれている場合（ステップS103）、解析部22は、ループをベクトル化できるか否かを判定する（ステップS104）。上記の通り、「ベクトル化できる条件」としては、例えば、従来の条件、つまり、「ループ内の配列及び変数についての定義及び参照関係にベクトル化を阻害する依存関係がないこと」を用いることができる。
- [0040] ベクトル化ができると判定した場合（ステップS104YES）、解析部22は、計算処理の配列の型が（1／2）^kワード型であるか否かを判定する（ステップS105）。
- [0041] 計算処理の配列の型が（1／2）^kワード型である場合（ステップS105YES）、解析部22は、計算処理において計算対象となる配列要素がすべて隣接しているか否かを判定する（ステップS106）。
- [0042] 計算処理において計算対象となる配列要素がすべて隣接している場合（ステップS106YES）、解析部22は、計算処理において計算に用いられる配列要素の数が「 $m \times 2^k + 1$ 」個であるか否かを判定する（ステップS107）。
- [0043] 計算処理において計算に用いられる配列要素の数が「 $m \times 2^k + 1$ 」個である場合（ステップS107YES）、解析部22は、原始プログラムに対して第2実施形態のベクトル化技法を適用することを決定する（ステップS108）。
- [0044] なお、ステップS101からステップS107のいずれかにおいて否定的な判定が為された場合、解析部22は、例えば従来の最適化を適用することを決定する（ステップS109）。
- [0045] 〈第2実施形態のベクトル化技法〉
図5は、第2実施形態のコンパイル装置によるベクトル化技法の一例を示すフローチャートである。図5の処理フローは、解析部22によって原始プログラムに対して第2実施形態のベクトル化技法を適用することを決定された場合にスタートする。
- [0046] コンパイル装置20においてベクトル化実行部23は、ループ長Nを取得

して $N \times (1/2)^k$ を右論理シフト演算によって計算するための、SRL命令を生成する（ステップS201）。原始プログラムが図3に示すプログラムの場合、図6に示されるオブジェクトコードの部分P6-1の1行目に示される命令が生成されることになる。図6は、第2実施形態のベクトル化技法によって生成されるオブジェクトコードの一例を示す図である。

[0047] ベクトル化実行部23は、システムにて許容されている「最大ベクトル長」を取得するための、SMVL命令を生成する（ステップS202）。

[0048] ベクトル化実行部23は、SRL命令によって計算した値と「最大ベクトル長」とを比較して小さい方を選択するための、MIN命令、及び、選択された値を「ベクトル長」とするための、LVL命令を生成する（ステップS203）。原始プログラムが図3に示すプログラムの場合、ステップS201からステップS203によって、図6に示されるオブジェクトコードの部分P6-1が生成されることになる。

[0049] ベクトル化実行部23は、 2^k 個のデータユニットを1つのパックドデータとしてそれぞれ異なるベクトルレジスタに1ワードベクトルロードするための、 $(m+1)$ 個のVLD命令を生成する（ステップS204）。これらの $(m+1)$ 個のVLD命令は、メモリ（図示せず）からデータユニットのロードを開始する位置が互いに 2^k 要素（つまり、1ワード分の要素数）ずつずれている。原始プログラムが図3に示すプログラムの場合、図6に示されるオブジェクトコードの部分P6-2、P6-5が生成されることになる。図3に示すプログラムでは配列が半ワード型（つまり、 $k=1$ ）であり且つ要素の数が5個（つまり、 $m=2$ ）であるので、3つのVLD命令が生成されている。また、ロード命令の対象は、 $a[i]$ 、 $a[i+2]$ 、及び $a[i+4]$ であり、これらのロード開始位置は互いに2要素ずつずれている。例えば、図6に示されるオブジェクトコードの部分P6-2が実行されると、図7に示されるような処理が実行されることになる。すなわち、「VLD vreg1, a[i]」が実行されることによって、複数のデータユニット $a[0] \sim a[11]$ が、ベクトルロード開始位置を $a[0]$ とし且つ2つ

のデータユニットが1つにパックされた状態で、ベクトルレジスタ $vreg1$ にロードされる。すなわち、ベクトルレジスタ $vreg1$ の1ワード単位の各レジスタ領域の上位領域（Upper領域）にデータユニット $a[0]$, $a[2]$, $a[4]$, $a[6]$, $a[8]$, $a[10]$ がロードされ、下位領域（Lower領域）にデータユニット $a[1]$, $a[3]$, $a[5]$, $a[7]$, $a[9]$, $a[11]$ がロードされている。また、「 $VLD\ vreg1, a[i+2]$ 」が実行されることによって、複数のデータユニット $a[2]$ ~ $a[13]$ が、ベクトルロード開始位置を $a[2]$ とし且つ2つのデータユニットが1つにパックされた状態で、ベクトルレジスタ $vreg2$ にロードされる。図7は、第2実施形態のベクトル化技法にて生成されるベクトルロード命令の説明に供する図である。

[0050] 図5の説明に戻り、ベクトル化実行部23は、「ベクトルロード開始位置」が最も近いデータユニット群の各ペアに対して「所定シフト量」分シフトするための、 $(2k-1)$ 個の $VSRD$ 命令を生成する（ステップS205）。 $(2k-1)$ 個の $VSRD$ 命令についての「所定シフト量」は、それぞれ、 $l \times (1/2)^k$ ワード (l は、1から $(2k-1)$ までの自然数) である。原始プログラムが図3に示すプログラムの場合、図6に示されるオブジェクトコードの部分P6-3, P6-6が生成されることになる。すなわち、 $a[i]$ 及び $a[i+2]$ のペアについて、部分P6-3が生成され、 $a[i+2]$ 、及び $a[i+4]$ のペアについて、部分P6-6が生成される。図3に示すプログラムでは配列が半ワード型（つまり、 $k=1$ ）であるので、各ペアについて1つの $VSRD$ 命令が生成され、所定シフト量は $1/2$ ワードとなる。例えば、図6に示されるオブジェクトコードの部分P6-3が実行されると、図8に示されるような処理が実行されることになる。すなわち、 $VSRD$ 命令のオペランドであるベクトルレジスタ $vreg1$, $vreg2$ を合わせて2ワードデータとし、該2ワードデータが $1/2$ ワードだけ右シフト演算され、得られた2ワードデータの下位側の1ワードの値がベクトルレジスタ $vreg3$ に書き込まれる。このベクトルレジスタ $vreg3$

に書き込まれた複数のデータユニットは、 $a[i+1]$ に相当する。図8は、第2実施形態のベクトル化技法にて生成されるベクトルシフトダブル命令の説明に供する図である。

[0051] 図5の説明に戻り、ベクトル化実行部23は、ベクトルロードされたベクトルデータ及びベクトルシフトダブルによって生成されたベクトルデータに対して演算を行うための、P V O P命令を生成する(ステップS206)。原始プログラムが図3に示すプログラムの場合、図6に示されるオブジェクトコードの部分P6-4, P6-7が生成されることになる。すなわち、図3に示すプログラムの計算式には加算演算子が4つ含まれるので、4つのP V O P命令が生成されている。図6に示されるオブジェクトコードの部分P6-4, P6-7が実行されると、図9に示すように、ベクトルレジスタ $vreg1, vreg2, vreg3, vreg6, vreg7$ の1ワード単位の各レジスタ領域における上位領域及び下位領域について別々に加算処理が行われる。図9は、第2実施形態のベクトル化技法にて生成されるパックドベクトル演算命令の説明に供する図である。

[0052] 図5の説明に戻り、ベクトル化実行部23は、P V O P命令による演算結果を、 2^k 要素のパックドデータとして1ワードベクトルストアするための、V S T命令を生成する(ステップS207)。図9に示されたP V O P命令による演算結果を見ると、各レジスタ領域における上位領域及び下位領域のそれぞれには、上記の「繰り返し計算処理」の各計算処理において得られる計算結果が保持されることがわかる。

[0053] なお、以上の説明では、1/2ワード型のケースを例にとって説明を行ったが、当然のことながらこれに限定されるものではなく、1/4ワード型、1/8ワード型についても同様に、上記の「ベクトル化技法」と適用することができる。

[0054] 例えば、1/4ワード型の場合、図3に示す原始プログラムは、図10に示すオブジェクトコードに変換される。図10は、1/4ワード型の場合に、第2実施形態のベクトル化技法によって生成されるオブジェクトコードの

一例を示す図である。

[0055] 具体的には、ステップS204では、配列が1/4ワード型（つまり、 $k=2$ ）であり且つ要素の数が5個（つまり、 $m=1$ ）であるので、2つのVLD命令が生成される。そして、ロード命令の対象は、 $a[i]$ 及び $a[i+4]$ であり、これらのロード開始位置は互いに4要素ずれている。これらの2つのVLD命令が実行されると、図11に示されるような処理が実行されることになる。すなわち、「VLD vreg1, a[i]」が実行されることによって、複数のデータユニット $a[0] \sim a[23]$ が、ベクトルロード開始位置を $a[0]$ とし且つ4つのデータユニットが1つにパックされた状態で、ベクトルレジスタvreg1にロードされる。ここで、図11のベクトルレジスタvreg1における各行は、1ワード単位の「レジスタ領域」に対応し、各行における4つの列のそれぞれは1/4ワードに対応する「部分領域」に対応する。また、「VLD vreg1, a[i+4]」が実行されることによって、複数のデータユニット $a[4] \sim a[27]$ が、ベクトルロード開始位置を $a[4]$ とし且つ4つのデータユニットが1つにパックされた状態で、ベクトルレジスタvreg2にロードされる。図11は、1/4ワード型の場合に、第2実施形態のベクトル化技法にて生成されるベクトルロード命令の説明に供する図である。

[0056] そして、ステップS205では、 $a[i]$ 及び $a[i+4]$ のペアについて、「所定シフト量」がそれぞれ1/4ワード、2/4ワード、3/4ワードである、3つのVSRD命令が生成される。 $a[i]$ 及び $a[i+4]$ のペアについて「所定シフト量」が3/4ワードであるVSRD命令「VSRD vreg3, vreg2, vreg1, 48」が実行されると、VSRD命令のオペランドであるベクトルレジスタvreg1, vreg2を合わせて2ワードデータとされる。そして、該2ワードデータが3/4ワードだけ右シフト演算され、得られた2ワードデータの下位側の1ワードの値がベクトルレジスタvreg3に書き込まれる。このベクトルレジスタvreg3に書き込まれた複数のデータユニットは、 $a[i+1]$ に相当する。図1

2は、1/4ワード型の場合に、第2実施形態のベクトル化技法にて生成されるベクトルシフトダブル命令の説明に供する図である。

[0057] そして、ステップS206では、ベクトルロードされたベクトルデータ及びベクトルシフトダブルによって生成されたベクトルデータに対して演算を行うための、PPOP命令が生成される。例えば、 $a[i]$ と $a[i+4]$ とのベクトル加算演算は、図13に示すように、上記の「部分領域」単位で加算されることになる。図13は、1/4ワード型の場合に、第2実施形態のベクトル化技法にて生成されるパックドベクトル演算命令の説明に供する図である。

[0058] このように、ベクトル化実行部23（ベクトルシフトダブル命令生成部13）は、配列aが1/4ワード型である場合、上記一連のデータ列（つまり、上記の2ワードデータ）を3/4ワード分シフトさせて得られたデータ列の一部を、第3ベクトルデータとして第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令と、上記一連のデータ列を2/4ワード分シフトさせて得られたデータ列の一部を、要素A[i+2]に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにストアさせる第2ベクトルシフトダブル命令と、上記一連のデータ列を1/4ワード分シフトさせて得られたデータ列の一部を、要素A[i+3]に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにストアさせる第3ベクトルシフトダブル命令と、を生成している。

[0059] 配列aが1/8ワード型の場合も同様である。すなわち、ベクトルシフトダブル命令生成部13は、配列aが1/8ワード型である場合、上記一連のデータ列（つまり、上記の2ワードデータ）を7/8ワード分シフトさせて得られたデータ列の一部を、第3ベクトルデータとして第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令と、上記一連のデータ列を6/8ワード分シフトさせて得られたデータ列の一部を、要素A[i+2]に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第

4ベクトルレジスタにストアさせる第2ベクトルシフトダブル命令と、上記一連のデータ列を5/8ワード分シフトさせて得られたデータ列の一部を、要素A [i + 3] に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにストアさせる第3ベクトルシフトダブル命令と、上記一連のデータ列を4/8ワード分シフトさせて得られたデータ列の一部を、要素A [i + 4] に対応する第5ベクトルデータとして、1ワード単位でパックした状態で第5ベクトルレジスタにストアさせる第4ベクトルシフトダブル命令と、上記一連のデータ列を3/8ワード分シフトさせて得られたデータ列の一部を、要素A [i + 5] に対応する第6ベクトルデータとして、1ワード単位でパックした状態で第6ベクトルレジスタにストアさせる第5ベクトルシフトダブル命令と、上記一連のデータ列を2/8ワード分シフトさせて得られたデータ列の一部を、要素A [i + 6] に対応する第7ベクトルデータとして、1ワード単位でパックした状態で第7ベクトルレジスタにストアさせる第6ベクトルシフトダブル命令と、上記一連のデータ列を1/8ワード分シフトさせて得られたデータ列の一部を、要素A [i + 7] に対応する第8ベクトルデータとして、1ワード単位でパックした状態で第8ベクトルレジスタにストアさせる第7ベクトルシフトダブル命令と、を生成している。

[0060] <他の実施形態>

図14は、コンパイル装置のハードウェア構成例を示す図である。図14においてコンパイル装置100は、プロセッサ101と、メモリ102とを有している。第1実施形態及び第2実施形態のコンパイル装置10, 20のコンパイル処理部11, 21は、プロセッサ101がメモリ102に記憶されたプログラムを読み込んで実行することにより実現されてもよい。プログラムは、様々なタイプの非一時的なコンピュータ可読媒体 (non-transitory computer readable medium) を用いて格納され、コンパイル装置10, 20に供給することができる。また、プログラムは、様々なタイプの一時的なコンピュータ可読媒体 (transitory computer readable medium) によってコン

パイル装置 10, 20 に供給されてもよい。

[0061] 以上、実施の形態を参照して本願発明を説明したが、本願発明は上記によって限定されるものではない。本願発明の構成や詳細には、発明のScope内で当業者が理解し得る様々な変更をすることができる。

[0062] この出願は、2018年12月7日に提出された日本出願特願2018-229695を基礎とする優先権を主張し、その開示の全てをここに取り込む。

符号の説明

- [0063] 10, 20 コンパイル装置
11, 21 コンパイル処理部
12 ベクトルロード命令生成部
13 ベクトルシフトダブル命令生成部
22 解析部
22A ループ検出部
22B ベクトル化可否判定部
22C 最適化適用可否判定部
23 ベクトル化実行部
23A ベクトル長算出命令生成部
23B パックドベクトル演算命令生成部
23C ベクトルストア命令生成部

請求の範囲

[請求項1]

(1 / 2)^kワード型 (kは自然数) の配列Aの要素A [i]、要素A [i + 1]、及び要素 [i + 2^k] をオペランドとして含む算術計算を実行する計算処理を添え字 i (i は0以上の整数) の値を1つずつずらしながら繰り返し実行するための原始プログラムを、オブジェクトコードに変換するコンパイル処理手段を具備し、

前記コンパイル処理手段は、

前記繰り返される計算処理において前記要素A [i]として用いられ且つ各データユニットが (1 / 2)^kワードである第1のデータユニット群をメモリから1ワード単位でパックした状態で第1ベクトルレジスタに第1ベクトルデータとしてロードするための第1ベクトルロード命令、及び、前記繰り返される計算処理において前記要素A [i + 2^k]として用いられる第2のデータユニット群を前記メモリから1ワード単位でパックした状態で第2ベクトルレジスタに第2ベクトルデータとしてロードするための第2ベクトルロード命令を生成するベクトルロード命令生成手段と、

前記第1ベクトルデータ及び前記第2ベクトルデータを一連のデータ列として (1 / 2)^kワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 1]に対応する第3ベクトルデータとして、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令を生成するベクトルシフト命令生成手段と、

を含む、

コンパイル装置。

[請求項2]

前記コンパイル処理手段は、前記ベクトルロード命令生成手段が前記第1ベクトルロード命令及び前記第2ベクトルロード命令を生成する処理を開始し且つ前記ベクトルシフト命令生成手段が前記第1ベクトルシフトダブル命令を生成する処理を開始する条件が満たされてい

るか否かを判定する条件判定手段をさらに含み、

前記条件は、各計算処理において計算対象となるすべての要素の添え字が連続しているという条件を含む、

請求項1記載のコンパイル装置。

[請求項3] 前記コンパイル処理手段は、前記計算処理が N (N は2以上の自然数)回繰り返される場合、前記 $N \times (1/2)^k$ 及びシステムにて許容されている最大ベクトル長のうちで小さい方をベクトル長として選択するためのベクトル長算出命令を生成するベクトル長算出命令生成手段をさらに具備し、

前記ベクトルロード命令生成手段は、それぞれ前記ベクトル長を有する前記第1ベクトルデータ及び前記第2ベクトルデータをロードするための前記第1ベクトルロード命令及び前記第2ベクトルロード命令を生成する、

請求項1又は2に記載のコンパイル装置。

[請求項4] 前記ベクトルシフト命令生成手段は、前記配列 A が $1/2$ ワード型である場合、前記一連のデータ列を $1/2$ ワード分シフトさせて得られたデータ列の一部を、前記第3ベクトルデータとして前記第3ベクトルレジスタにストアさせる前記第1ベクトルシフトダブル命令を生成する、

請求項1から3のいずれか1項に記載のコンパイル装置。

[請求項5] 前記ベクトルシフト命令生成手段は、前記配列 A が $1/4$ ワード型である場合、

前記一連のデータ列を $1/4$ ワード分シフトさせて得られたデータ列の一部を、前記第3ベクトルデータとして前記第3ベクトルレジスタにストアさせる前記第1ベクトルシフトダブル命令と、

前記一連のデータ列を $2/4$ ワード分シフトさせて得られたデータ列の一部を、前記要素 $A[i+2]$ に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにスト

アさせる第2ベクトルシフトダブル命令と、

前記一連のデータ列を1/4ワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 3] に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにストアさせる第3ベクトルシフトダブル命令と、

を生成する、

請求項1から3のいずれか1項に記載のコンパイル装置。

[請求項6]

前記ベクトルシフト命令生成手段は、前記配列Aが1/8ワード型である場合、

前記一連のデータ列を7/8ワード分シフトさせて得られたデータ列の一部を、前記第3ベクトルデータとして前記第3ベクトルレジスタにストアさせる前記第1ベクトルシフトダブル命令と、

前記一連のデータ列を6/8ワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 2] に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにストアさせる第2ベクトルシフトダブル命令と、

前記一連のデータ列を5/8ワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 3] に対応する第4ベクトルデータとして、1ワード単位でパックした状態で第4ベクトルレジスタにストアさせる第3ベクトルシフトダブル命令と、

前記一連のデータ列を4/8ワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 4] に対応する第5ベクトルデータとして、1ワード単位でパックした状態で第5ベクトルレジスタにストアさせる第4ベクトルシフトダブル命令と、

前記一連のデータ列を3/8ワード分シフトさせて得られたデータ列の一部を、前記要素A [i + 5] に対応する第6ベクトルデータとして、1ワード単位でパックした状態で第6ベクトルレジスタにストアさせる第5ベクトルシフトダブル命令と、

前記一連のデータ列を $2/8$ ワード分シフトさせて得られたデータ列の一部を、前記要素A [$i + 6$] に対応する第7ベクトルデータとして、1ワード単位でパックした状態で第7ベクトルレジスタにストアさせる第6ベクトルシフトダブル命令と、

前記一連のデータ列を $1/8$ ワード分シフトさせて得られたデータ列の一部を、前記要素A [$i + 7$] に対応する第8ベクトルデータとして、1ワード単位でパックした状態で第8ベクトルレジスタにストアさせる第7ベクトルシフトダブル命令と、

を生成する、

請求項1から3のいずれか1項に記載のコンパイル装置。

[請求項7]

($1/2$)^kワード型 (kは自然数) の配列Aの要素A [i]、要素A [$i + 1$]、及び要素A [$i + 2^k$] をオペランドとして含む算術計算を実行する計算処理を添え字 i (i は0以上の整数) の値を1つつずらしながら繰り返し実行するための原始プログラムを、オブジェクトコードに変換するコンパイル方法であって、

前記繰り返される計算処理において前記要素A [i] として用いられ且つ各データユニットが ($1/2$)^kワードである第1のデータユニット群をメモリから1ワード単位でパックした状態で第1ベクトルレジスタに第1ベクトルデータとしてロードするための第1ベクトルロード命令、及び、前記繰り返される計算処理において前記要素A [$i + 2^k$] として用いられる第2のデータユニット群を前記メモリから1ワード単位でパックした状態で第2ベクトルレジスタに第2ベクトルデータとしてロードするための第2ベクトルロード命令を生成すること、及び、

前記第1ベクトルデータ及び前記第2ベクトルデータを一連のデータ列として ($1/2$)^kワード分シフトさせて得られたデータ列の一部を、前記要素A [$i + 1$] に対応する第3ベクトルデータとして、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせ

る第1ベクトルシフトダブル命令を生成すること、を含む、
コンパイル方法。

[請求項8]

(1/2)^kワード型(kは自然数)の配列Aの要素A[i]、要素A[i+1]、及び要素A[i+2^k]をオペランドとして含む算術計算を実行する計算処理を添え字i(iは0以上の整数)の値を1つずつずらしながら繰り返し実行するための原始プログラムを、オブジェクトコードに変換するコンパイル処理をコンパイル装置に実行させる制御プログラムであって、

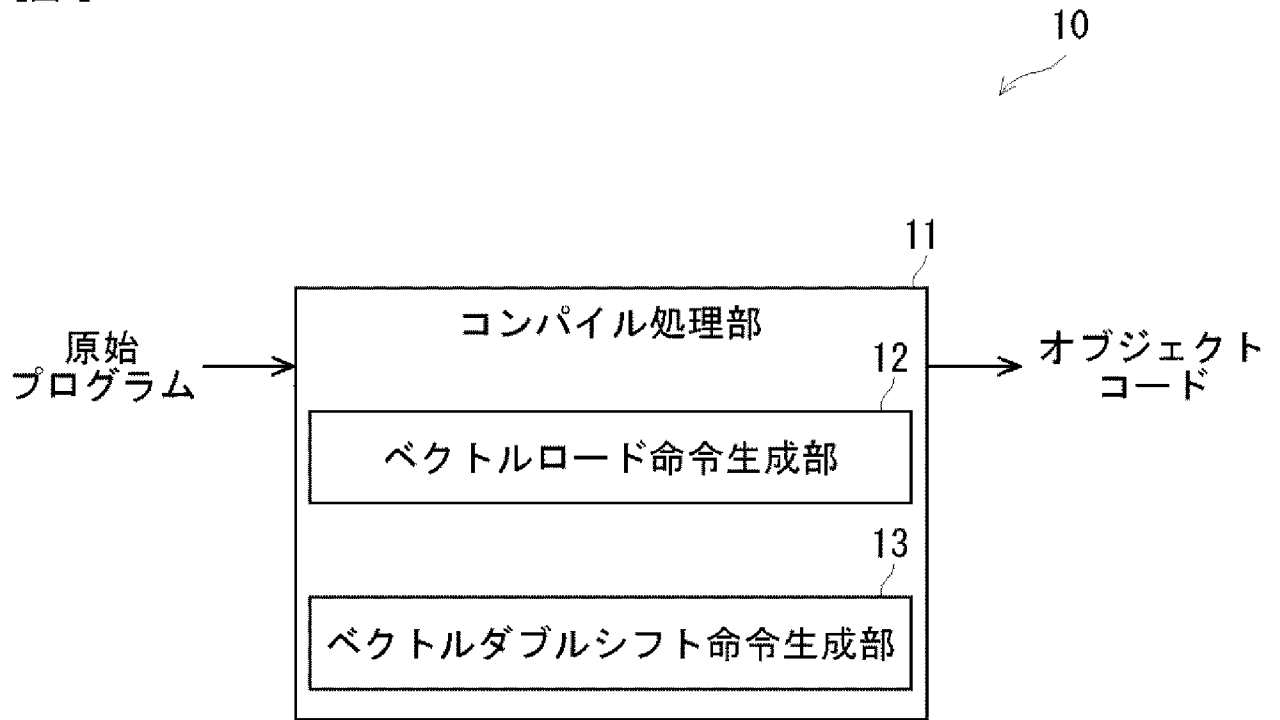
前記コンパイル処理は、

前記繰り返される計算処理において前記要素A[i]として用いられ且つ各データユニットが(1/2)^kワードである第1のデータユニット群をメモリから1ワード単位でパックした状態で第1ベクトルレジスタに第1ベクトルデータとしてロードするための第1ベクトルロード命令、及び、前記繰り返される計算処理において前記要素A[i+2^k]として用いられる第2のデータユニット群を前記メモリから1ワード単位でパックした状態で第2ベクトルレジスタに第2ベクトルデータとしてロードするための第2ベクトルロード命令を生成すること、及び、

前記第1ベクトルデータ及び前記第2ベクトルデータを一連のデータ列として(1/2)^kワード分シフトさせて得られたデータ列の一部を、前記要素A[i+1]に対応する第3ベクトルデータとして、1ワード単位でパックした状態で第3ベクトルレジスタにストアさせる第1ベクトルシフトダブル命令を生成すること、を含む、

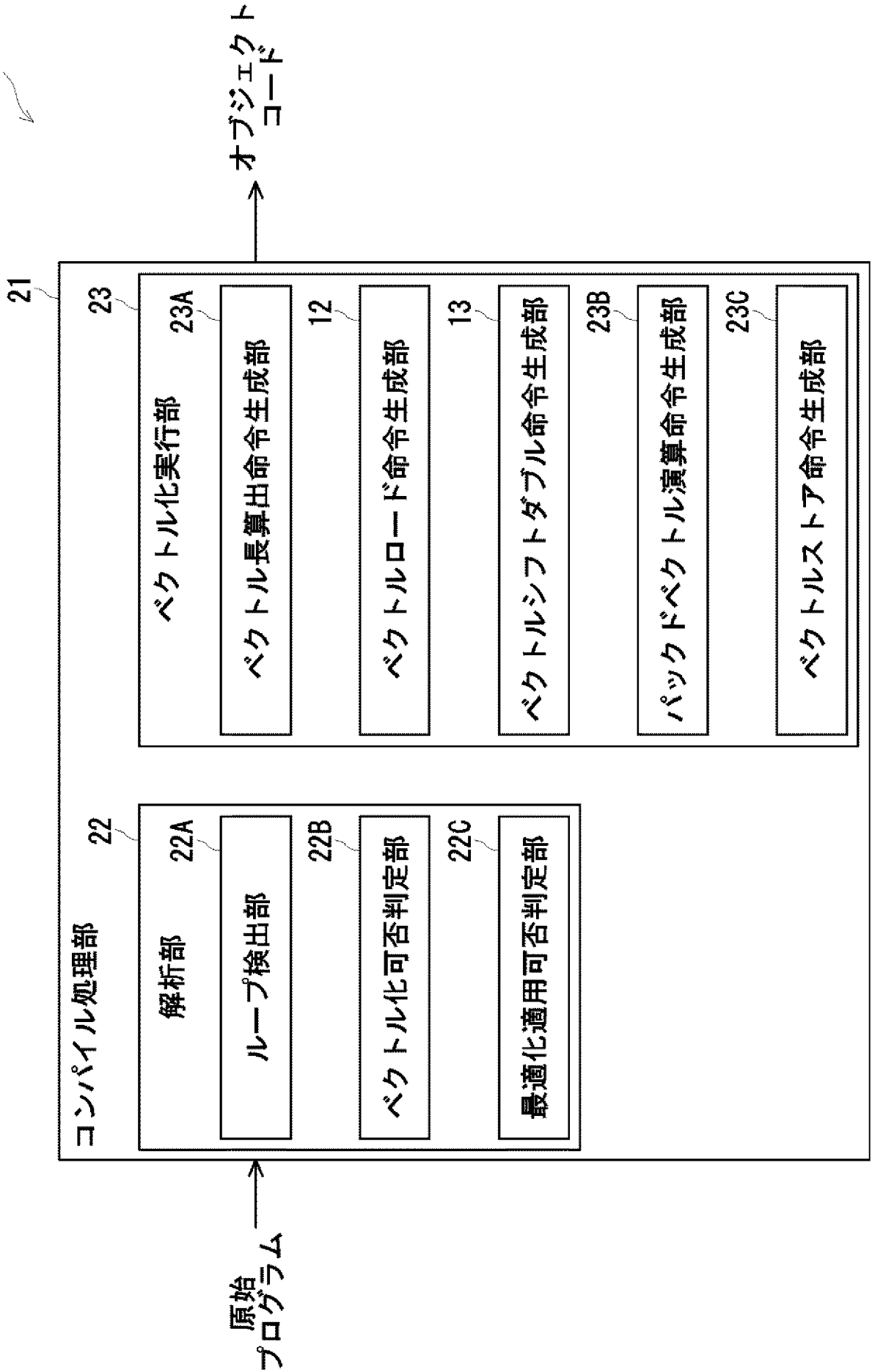
制御プログラムを格納する非一時的なコンピュータ可読媒体。

[図1]



[図2]

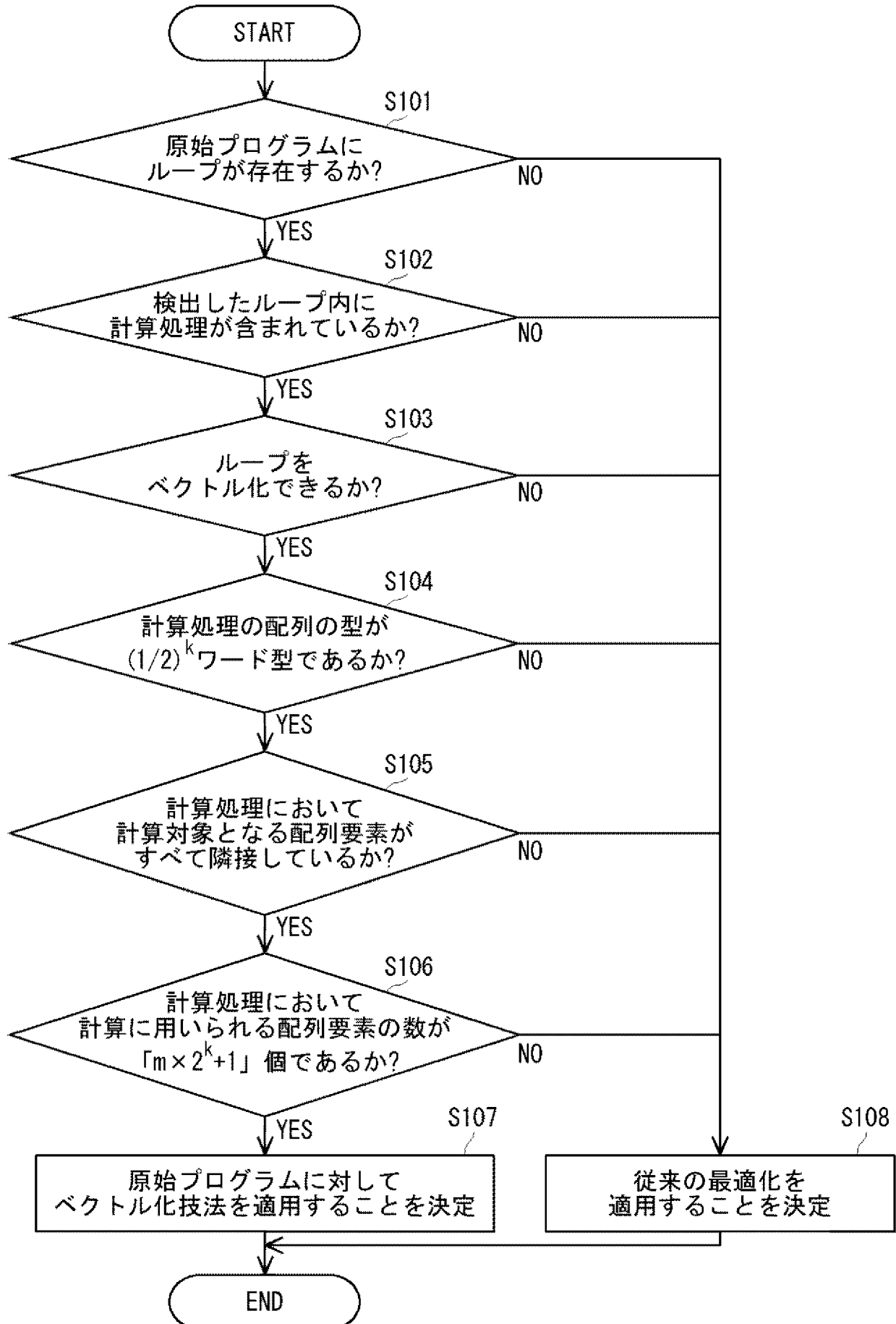
20



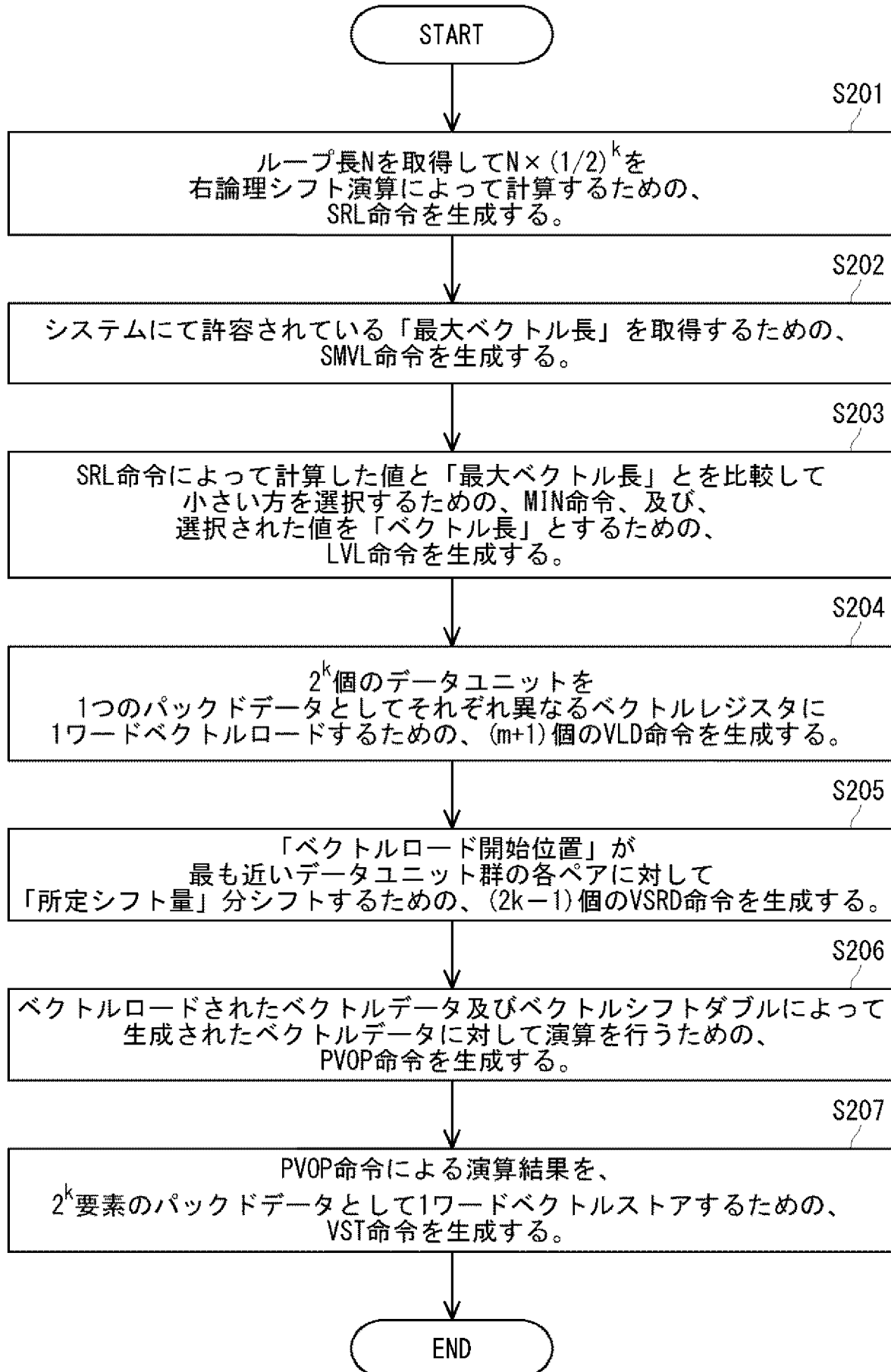
[図3]

```
void stencil(float *a, float *b) {  
    for (int i=0, i<n, i++)  
        b[i]=a[i]+a[i+1]+a[i+2]+a[i+3]+a[i+4]  
}
```

[図4]



[図5]

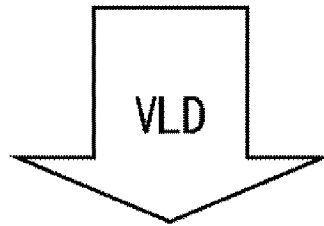
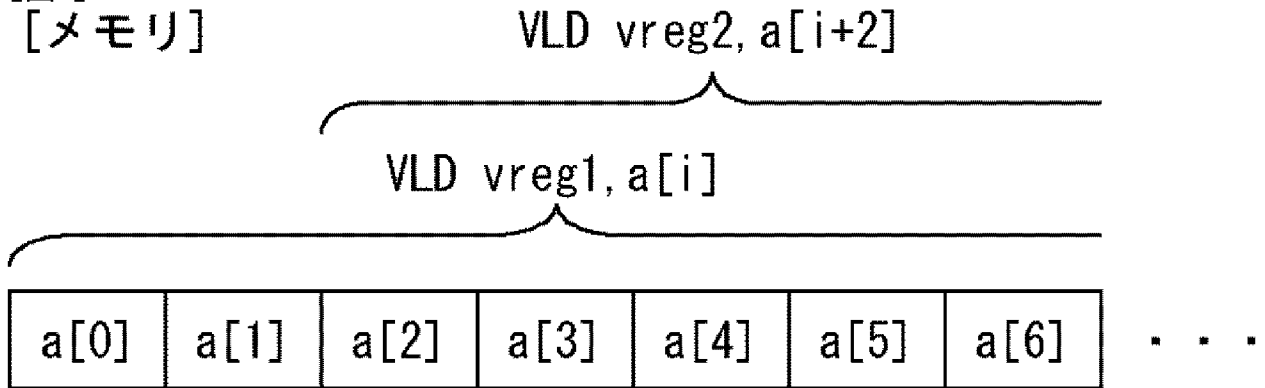


[図6]

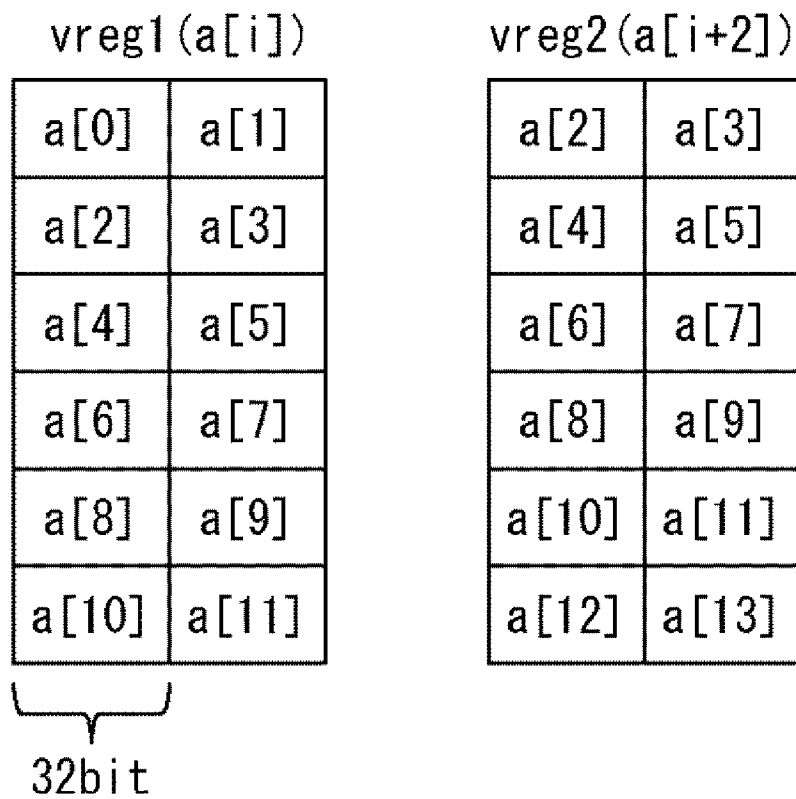
SRL	sreg1, n,	1	}	P6-1
SMVL	sreg2			
MIN	sreg3, sreg1, sreg2			
LVL	sreg3			
VLD	vreg1, a[i]		}	P6-2
VLD	vreg2, a[i+2]			
VSRD	vreg3, vreg2, vreg1,	32	}	P6-3
PVADD	vreg4, vreg2, vreg1		}	P6-4
PVADD	vreg5, vreg4, vreg3			
VLD	vreg6, a[i+4]		}	P6-5
VSRD	vreg7, vreg6, vreg2,	32	}	P6-6
PVADD	vreg8, vreg6, vreg5		}	P6-7
PVADD	vreg9, vreg8, vreg7			
VST	vreg9, b[i]		}	P6-8

[図7]

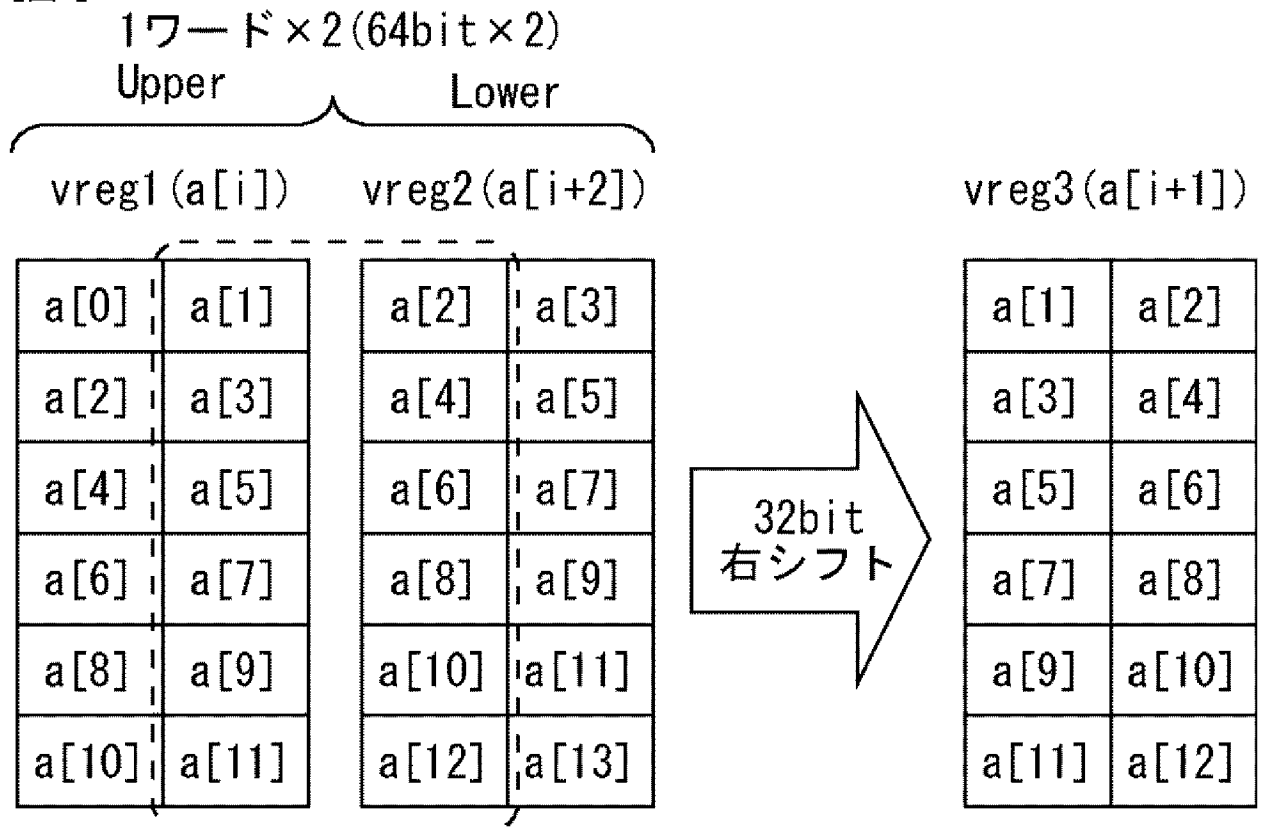
[メモリ]



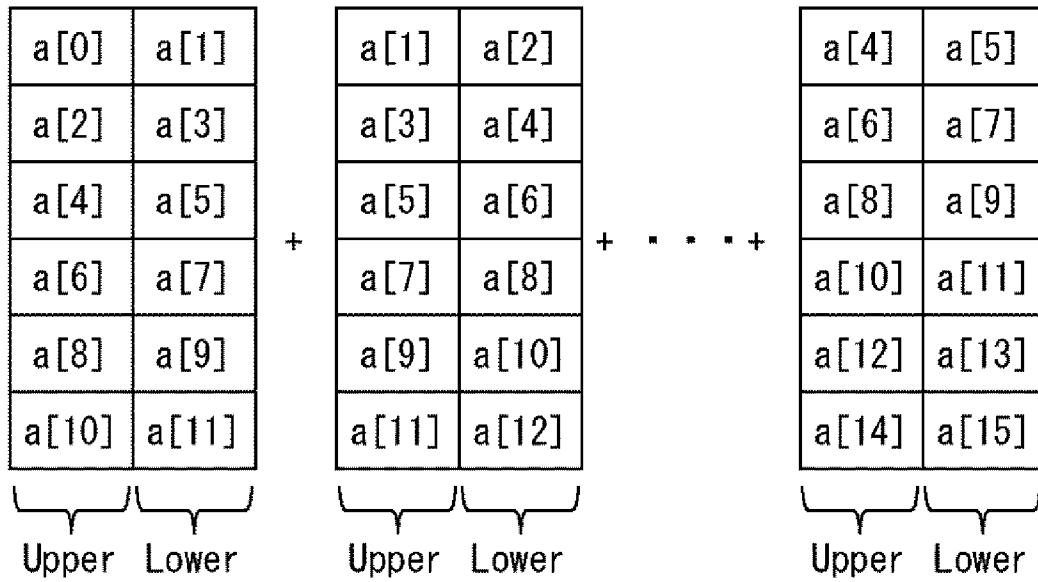
[レジスタ]



[図8]



[図9]



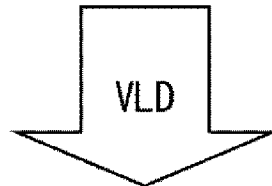
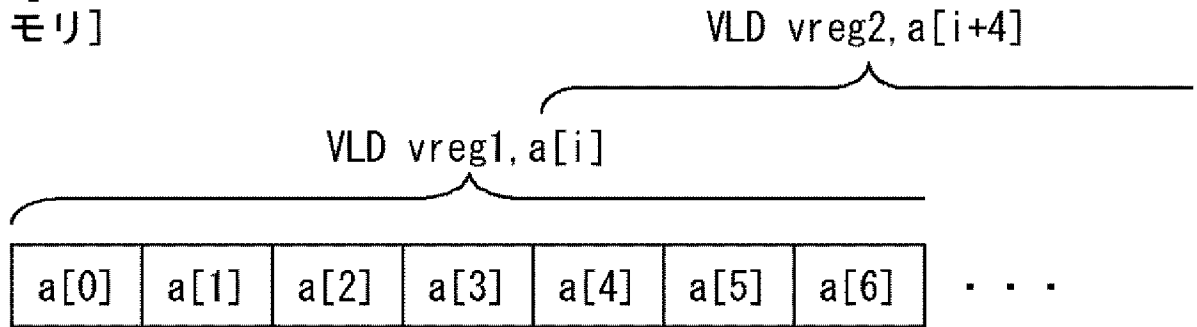
=	a[0]+a[1]+a[2]+a[3]+a[4]	a[1]+a[2]+a[3]+a[4]+a[5]
	a[2]+a[3]+a[4]+a[5]+a[6]	a[3]+a[4]+a[5]+a[6]+a[7]
	a[4]+a[5]+a[6]+a[7]+a[8]	a[5]+a[6]+a[7]+a[8]+a[9]
	a[6]+a[7]+a[8]+a[9]+a[10]	a[7]+a[8]+a[9]+a[10]+a[11]
	a[8]+a[9]+a[10]+a[11]+a[12]	a[9]+a[10]+a[11]+a[12]+a[13]
	a[10]+a[11]+a[12]+a[13]+a[14]	a[11]+a[12]+a[13]+a[14]+a[15]
	} Upper Lower	

[図10]

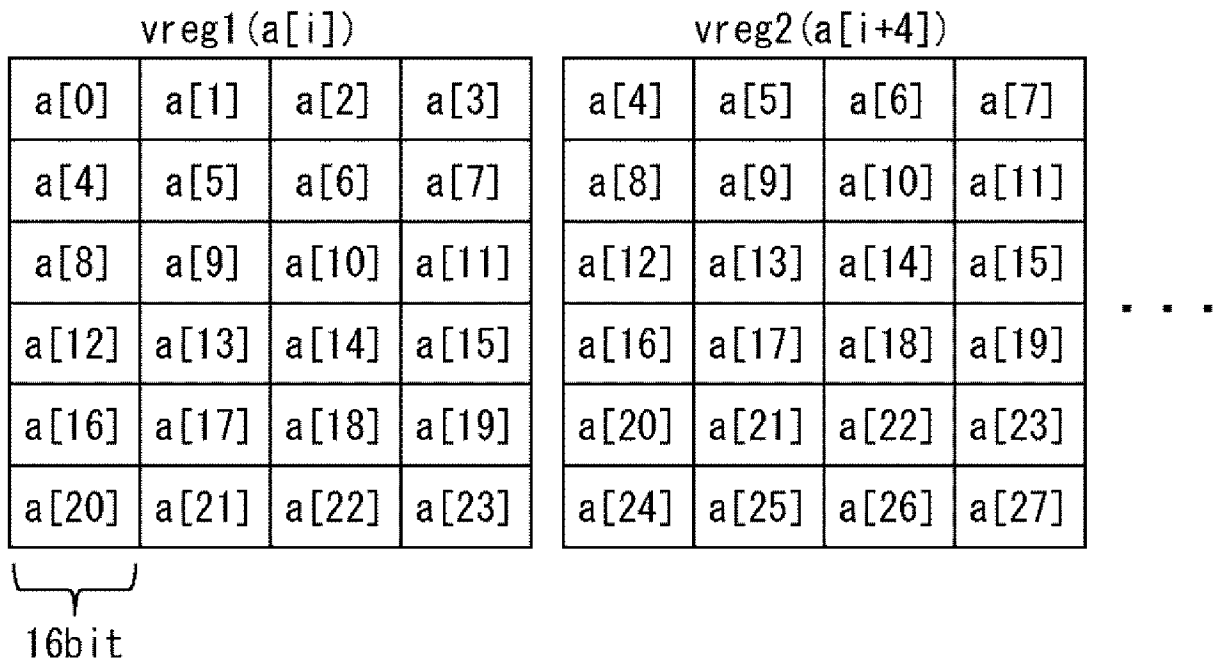
SRL	sreg1, n,	2
SMVL	sreg2	
MIN	sreg3, sreg1, sreg2	
LVL	sreg3	
VLD	vreg1, a[i]	
VLD	vreg2, a[i+4]	
VSRD	vreg3, vreg2, vreg1,	48
PVADD	vreg4, vreg2, vreg1	
PVADD	vreg5, vreg4, vreg3	
VSRD	vreg6, vreg2, vreg1,	32
PVADD	vreg7, vreg6, vreg5	
VSRD	vreg8, vreg2, vreg1,	16
PVADD	vreg9, vreg8, vreg7	
VST	vreg9, b[i]	

[図11]

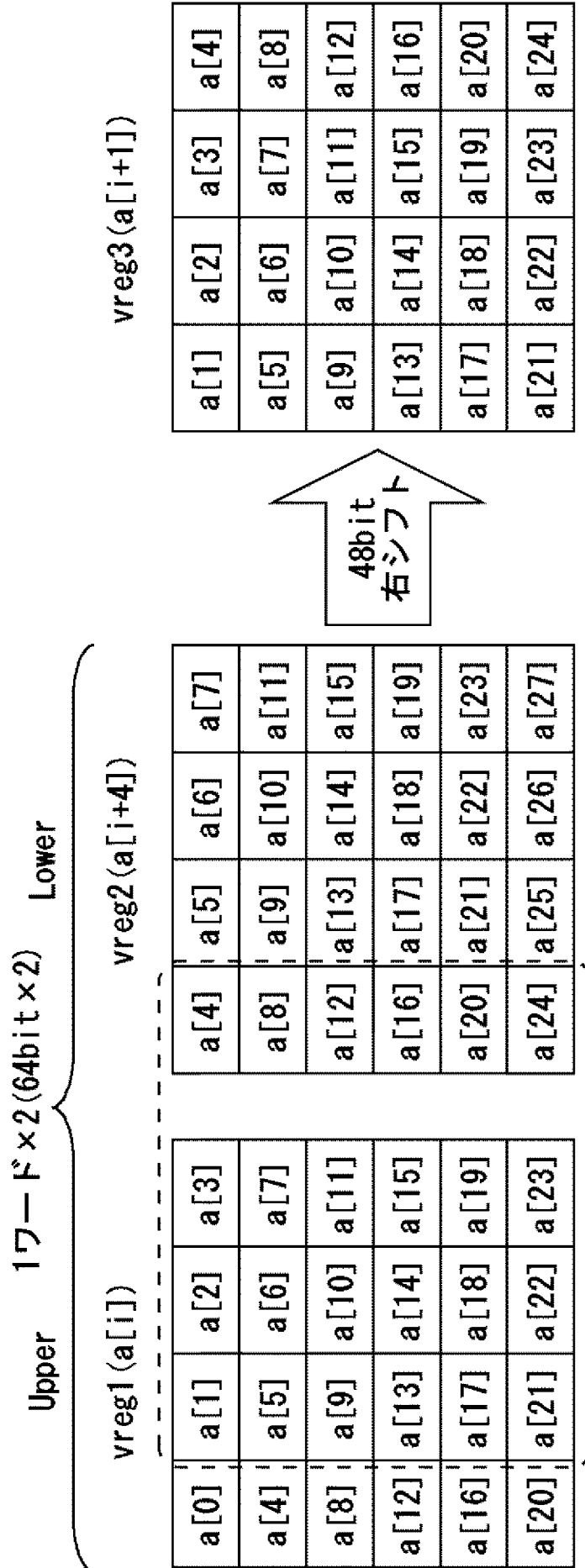
[メモリ]



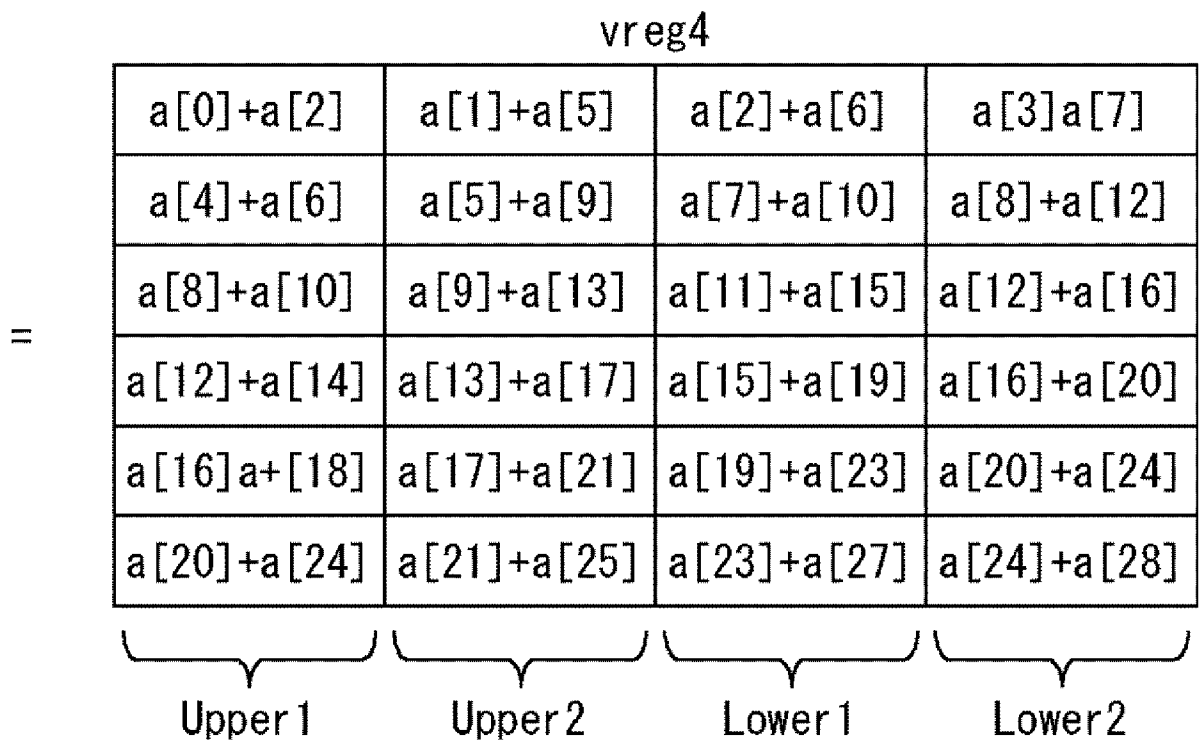
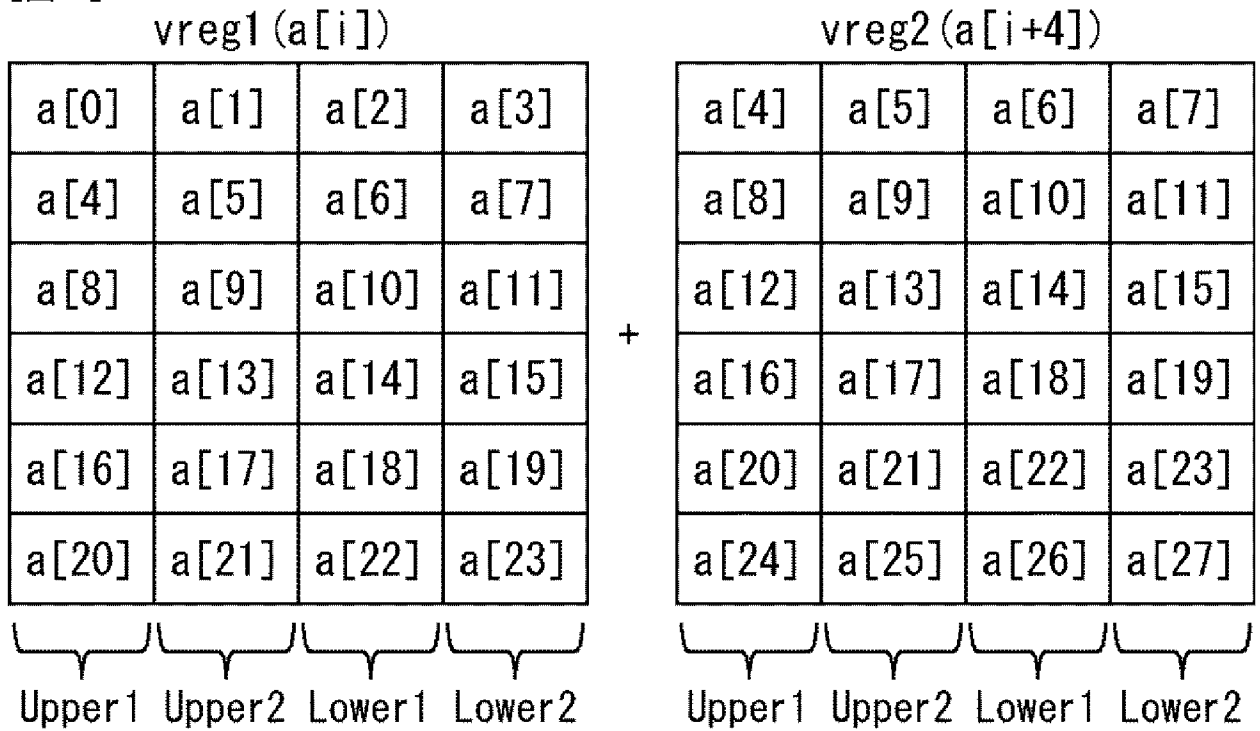
[レジスタ]



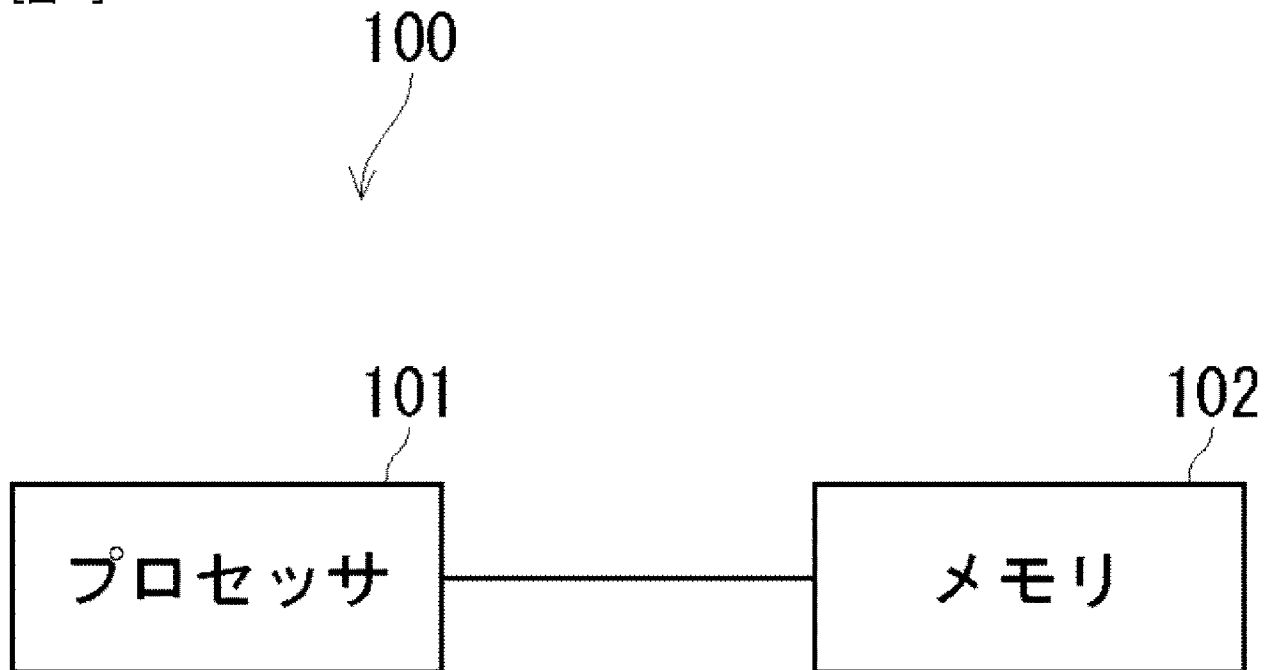
[図12]



[図13]



[図14]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2019/040150

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. G06F8/41 (2018.01) i, G06F17/16 (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)

Int. Cl. G06F8/41, G06F17/16

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996

Published unexamined utility model applications of Japan 1971-2019

Registered utility model specifications of Japan 1996-2019

Published registered utility model applications of Japan 1994-2019

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2018-49461 A (NEC CORP.) 29 March 2018, abstract, fig. 2, 3 (Family: none)	1-8
A	JP 9-62654 A (KOFU NIPPON DENKI KK) 07 March 1997, entire text, all drawings (Family: none)	1-8
A	JP 2014-526758 A (INTEL CORP.) 06 October 2014, entire text, all drawings & US 2014/0195778 A1, entire text, all drawings & CN 103827814 A & KR 10-2014-0057363 A	1-8
A	JP 2012-128790 A (FUJITSU LTD.) 05 July 2012, entire text, all drawings (Family: none)	1-8

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:	"I" 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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"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	

Date of the actual completion of the international search
29.11.2019

Date of mailing of the international search report
17.12.2019

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC）） Int.Cl. G06F8/41(2018.01)i, G06F17/16(2006.01)i		
B. 調査を行った分野 調査を行った最小限資料（国際特許分類（IPC）） Int.Cl. G06F8/41, G06F17/16		
最小限資料以外の資料で調査を行った分野に含まれるもの 日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-2019年 日本国実用新案登録公報 1996-2019年 日本国登録実用新案公報 1994-2019年		
国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2018-49461 A（日本電気株式会社）2018.03.29, 要約, [図2] - [図3] （ファミリーなし）	1-8
A	JP 9-62654 A（甲府日本電気株式会社）1997.03.07, 全文, 全図 （ファミリーなし）	1-8
A	JP 2014-526758 A（インテル・コーポレーション）2014.10.06, 全文, 全図 & US 2014/0195778 A1, 全文, 全図 & CN 103827814 A & KR 10-2014-0057363 A	1-8
☑ C欄の続きにも文献が列挙されている。 ☐ パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」特に関連のある文献ではなく、一般的技術水準を示すもの 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す） 「O」口頭による開示、使用、展示等に言及する文献 「P」国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」同一パテントファミリー文献		
国際調査を完了した日 29.11.2019	国際調査報告の発送日 17.12.2019	
国際調査機関の名称及びあて先 日本国特許庁（ISA/J P） 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官（権限のある職員） 井上 宏一 電話番号 03-3581-1101 内線 3544	5B 4177

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2012-128790 A (富士通株式会社) 2012.07.05, 全文, 全図 (ファミリーなし)	1-8