

(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
2013年2月7日(07.02.2013)

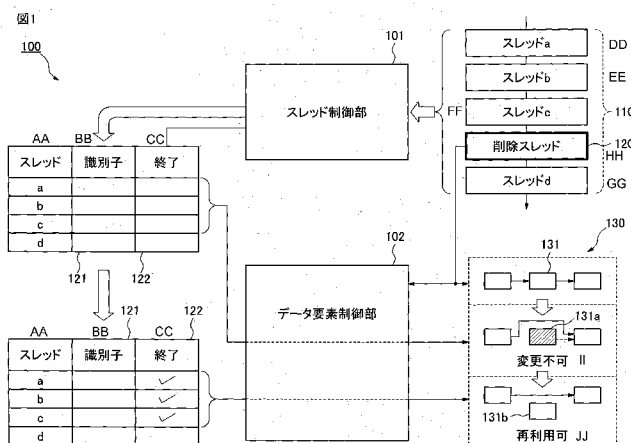


(10) 国際公開番号
WO 2013/018593 A1

- (51) 国際特許分類:
G06F 9/52 (2006.01) G06F 12/00 (2006.01)
 - (21) 国際出願番号: PCT/JP2012/068748
 - (22) 国際出願日: 2012年7月18日(18.07.2012)
 - (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 2011-167691 2011年7月29日(29.07.2011) JP
 - (71) 出願人 (米国を除く全ての指定国について): 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; および
 - (75) 発明者/出願人 (米国についてのみ): 堀川 隆 (HORIKAWA, Takashi) [JP/JP]; 〒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: 下坂 直樹 (SHIMOSAKA, Naoki); 〒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).
 - (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, 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, ML, MR, NE, SN, TD, TG).
- 添付公開書類:
— 国際調査報告 (条約第 21 条(3))

(54) Title: INFORMATION PROCESSING APPARATUS, INFORMATION PROCESSING SYSTEM, INFORMATION PROCESSING METHOD, AND CONTROL PROGRAM STORAGE MEDIUM

(54) 発明の名称: 情報処理装置、情報処理システム、情報処理方法および制御プログラム記録媒体



(57) Abstract: The present invention provides an information processing apparatus. The information processing apparatus is provided with: a thread control means which, upon initiation of a plurality of threads, grants identifiers to each of the threads, and, when each of the threads is to end, sends a notification of the ending of the thread accompanied with the identifier thereof; and a data element control means which, when a deletion thread for deleting a data element from list-structured data is executed, maintains a state wherein the content of the deleted data element cannot be changed until all the threads initiated before the deletion processing by the deletion thread have been confirmed as having ended from the notifications of the endings accompanied with the identifiers, and changes the deleted data element into a reusable state upon notification, accompanied with the identifiers, that all threads initiated before the deletion processing by the deletion thread have ended.

(57) 要約:

[続葉有]

- 101... THREAD CONTROL UNIT
- 102... DATA ELEMENT CONTROL UNIT
- AA... THREAD
- BB... IDENTIFIER
- CC... END
- DD... THREAD a
- EE... THREAD b
- FF... THREAD c
- GG... THREAD d
- HH... DELETION THREAD
- II... UNCHANGEABLE
- JJ... REUSABLE

WO 2013/018593 A1

本発明は、情報処理装置を提供する。係る情報処理装置は、複数のスレッドを開始するに際して、各スレッドに識別子を付与し、前記各スレッドの終了に際して、前記識別子を伴って終了を通知するスレッド制御手段と、リスト構造化されたデータからデータ要素を削除する削除スレッドが実行された場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にするデータ要素制御手段とを備える。

明細書

情報処理装置、情報処理システム、情報処理方法および制御プログラム記録媒体

5 技術分野

本発明は、複数のスレッドを並列的に実行する情報処理システムにおいて、リスト構造化されたデータからデータ要素を削除する情報処理技術に関する。

背景技術

- 10 上記技術分野において、クリティカルセクションを形成することなく、複数のスレッドからのアクセスを矛盾なく実施する方法として、`cmpxchg` 命令が知られている。`cmpxchg` 命令は、例えば、Intel 64 and IA-32 Architectures Software Developer's Manual Volume 2A: Instruction Set Reference, A-M(以下、非特許文献1と称する)に開示される。マルチプロセッサ命令の典型例である `cmpxchg` 命令は、CAS(Compare And
15 Swap)操作により、命令実行中に他の命令によりメモリが変更されないように制御する。

- Maged M. Michael, High Performance Dynamic Lock-Free Hash Tables and List-Based Sets, ACM SPAA'02(以下、非特許文献2と称する)は、上記CAS操作を
20 用いて、リスト構造化されたデータへのデータ要素の挿入または、リスト構造化されたデータからのデータ要素の削除において、正しい実行結果が得られるアルゴリズムを開示する。ここで、リスト構造とは、複数のデータ要素がポインタで接続された構造をいう。非特許文献2では、リスト構造化されたデータからのデータ要素の削除を、論理的な削除を実行するCAS操作と物理的な削除を実行するCAS操作とにより実行する。
25 そして、論理的に削除されたが物理的に削除されていないデータ要素を表わすマークビットを設けることにより、削除命令処理中に他の命令が介入することを防ぐ。ここで、データ要素の削除とは、データ要素が持つ情報を記録するために割り当てられていたメモリ領域が解放されて未使用状態となることである。解放されたメモリ領域は、再利用可能状態となる。

一方、米国特許第 5,442,758 号に開示されるRCU(Read-Copy Update)アルゴリズムは、リスト構造化されたデータからのデータ要素の削除を、非特許文献2に開示される削除に相当する削除フェーズと、削除したデータ要素を再利用可能状態にする再設定フェーズとにより実行する。

5

発明の概要

非特許文献2では、リスト構造化されたデータからデータ要素を削除する削除スレッドが正しく処理されることは保証される。しかしながら、削除スレッドの開始より前に開始された、削除されるデータ要素にアクセスするスレッドが正しく処理されることは保証されない。これは、データ要素に割り当てられていたメモリ領域は、物理的な削除が行われた後は再利用可能状態となるためである。ここで、再利用可能状態とは、そのメモリ領域に別の情報を書き込んでも、そのデータ要素が前に属していたリスト構造化されたデータを操作する別スレッドの処理に悪影響を及ぼさない状態を意味する。削除されたデータ要素が再利用可能状態となり、その内容が書き換えられると、削除スレッドの開始より前に開始された、削除されたデータ要素にアクセスするスレッドは、書き換えられたデータ要素にアクセスしてしまう。したがって、このスレッドは正しく処理されない恐れがあるという課題がある。

一方、米国特許第 5,442,758 号によれば、リストから削除した要素が再利用可能であって、再設定フェーズに移行可能か否かの判定が実施される。したがって、削除スレッドの開始より前に開始された、削除されるデータ要素にアクセスするスレッドの実行中に、データ要素の内容が書き換えられることは防ぐことができる。しかしながら、再設定フェーズに移行可能か否かの判定は、削除フェーズ完了後に全プロセッサがプロセス切替を行なったことを確認することによって実施される。したがって、削除されたデータ要素が実際に再利用可能状態になったとしても、そのデータ要素が再利用できるのは、削除されてからかなり後になってしまうという課題がある。

例えば、スレッド間通信において情報交換することにより、上記課題を解決することも考えられる。しかしながら、多くの削除スレッドと、データ要素にアクセスする多くの検索スレッドとが並列に実行している場合は、オーバーヘッドが大きくなると共に、管理が煩雑になってしまう。

30

本発明の目的は、上述の課題を解決する技術を提供することを主要な目的とする。

課題を解決するための手段

- 5 本発明の第1の情報処理装置は、複数のスレッドを開始するに際して、各スレッドに識別子を付与し、前記各スレッドの終了に際して、前記識別子を伴って終了を通知するスレッド制御手段と、リスト構造化されたデータからデータ要素を削除する削除スレッドが実行された場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にするデータ要素制御手段とを備える。

- 15 本発明の第1の情報処理システムは、リスト構造化されたデータの検索を行う検索スレッドと、リスト構造化されたデータからデータ要素を削除する削除スレッドとを実行する処理手段と、前記処理手段による複数のスレッドの実行を管理する処理状態管理手段とを備え、前記処理状態管理手段は、最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前のスレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの開始および終了を示す情報を含む順序番号管理データとを保持する状態保持手段と、開始されたスレッドに対して一意に増加する順序番号を付与する開始処理部と、順序番号に対応するスレッドが終了したことを順序番号管理データに反映する終了処理部とを含む開始終了処理手段とを備え、前記削除スレッドからの要求に回答して、前記状態保持手段から前記最大順序番号を取得してそれを返す最大順序番号取得部と、前記削除スレッドから取得した最大順序番号と前記状態保持手段に保持された最小順序番号とを比較する最小順序番号比較部とを含む終了判定手段とを備え、前記検索スレッドの検索処理の開始に際して前記開始処理部を呼び出し、前記検索スレッドの検索処理の終了に際して前記終了処理部を呼び出し、前記削除スレッドのデータ要素の削除処理の後に前記最大順序番号取得部と前記最小順序番号比較部とを呼び出すことにより、前記削除スレッドが削除したデータ要素が再利用可能かどうかの判定を実施する。

本発明の第1の情報処理方法は、複数のスレッドを開始するに際して、各スレッドに識別子を付与し、前記各スレッドの終了に際して、前記識別子を伴って終了を通知し、リスト構造化されたデータからデータ要素を削除する削除スレッドが実行される場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にする。

10

なお同目的は、上記の各構成を有する情報処理装置および情報処理方法を、コンピュータによって実現するコンピュータ・プログラム、およびそのコンピュータ・プログラムが格納されている、コンピュータ読み取り可能な記憶媒体によっても達成される。

15 発明の効果

本発明によれば、リスト構造化されたデータからのデータ要素の削除が実行中のスレッドに影響を与えることなく、削除されたデータ要素を迅速に再利用可能状態にすることができるといふ効果が得られる。

20 図面の簡単な説明

図1は、本発明の第1実施形態に係る情報処理システムの構成を示すブロック図である。

図2は、本発明の第2実施形態に係る情報処理システムの構成を示すブロック図である。

25 図3は、本発明の第2実施形態に係る情報処理システムの動作手順を示すシーケンス図である。

図4は、本発明の第2実施形態に係る処理状態管理部の構成を示すブロック図である。

図5は、本発明の第2実施形態で使用する変数の定義を示す図である。

30 図6は、本発明の第2実施形態で使用する操作の定義を示す図である。

図7Aは、本発明の第2実施形態で使用するデータ構成を示す図である。

図7Bは、本発明の第2実施形態で使用するデータ表記を示す図である。

図8Aは、本発明の第2実施形態に係る board 配列の構成例を示す図である。

図8Bは、本発明の第2実施形態に係る board 配列の他の構成例を示す図である。

5 図9は、本発明の第2実施形態に係る情報処理装置のハードウェア構成を示すブロック図である。

図10Aは、本発明の第2実施形態に係る情報処理装置のリスト検索スレッドの処理手順を示すフローチャートである。

10 図10Bは、本発明の第2実施形態に係る情報処理装置のリスト要素削除スレッドの処理手順を示すフローチャートである。

図11Aは、本発明の第2実施形態に係る開始処理の処理手順を示すフローチャートである。

図11Bは、本発明の第2実施形態に係る開始処理の処理手順を示すフローチャートである。

15 図12Aは、本発明の第2実施形態に係る終了処理の処理手順を示すフローチャートである。

図12Bは、本発明の第2実施形態に係る終了処理の処理手順を示すフローチャートである。

20 図12Cは、本発明の第2実施形態に係る終了処理の処理手順を示すフローチャートである。

図13は、本発明の第2実施形態に係る最大順序番号取得処理の処理手順を示すフローチャートである。

図14は、本発明の第2実施形態に係る最小番号比較処理の処理手順を示すフローチャートである。

25 図15は、本発明の第3実施形態に係る情報処理システムの構成を示すブロック図である。

発明を実施するための最良の形態

30 以下に、図面を参照して、本発明の実施の形態について例示的に詳しく説明する。

ただし、以下の実施の形態に記載されている構成要素は単なる例示であり、本発明の技術範囲をそれらだけに限定する趣旨のものではない。

第1実施形態

- 5 本発明の第1実施形態としての情報処理装置100について、図1を用いて説明する。情報処理装置100は、複数のスレッドを並列的に(並行して)実行するように構成されている。

図1に示すように、情報処理装置100は、スレッド制御部101と、データ要素制御部102と、を含む。スレッド制御部101は、複数のスレッド110を開始するに際して、各スレッドに識別子121を付与すると共に、各スレッドの終了に際して、識別子121を伴って終了122を通知する。データ要素制御部102は、リスト構造化されたデータ130からデータ要素131を削除する削除スレッド120が実行された場合に、以下のように動作する。すなわち、データ要素制御部102は、識別子121を伴った終了122の通知により削除スレッド120の削除処理以前に開始したすべてのスレッドa~cの終了が確認されるまで、削除されたデータ要素131の内容を変更できない状態131aに維持する。データ要素制御部102は、削除スレッド120の削除処理以前に開始したすべてのスレッドa~cの終了が識別子121を伴って通知されている場合、削除されたデータ要素131を再利用可能な状態131bにする。

- 20 スレッド制御部101は、以下の実施形態における開始終了処理部410および状態保持部430に相当し、データ要素制御部102は、同じく終了判定部420に相当する。

ここで、再利用可能な状態にするととは、データ要素が持つ情報を記録するために割り当てられていたメモリ領域を解放し、未使用な状態とすることである。

- 25 本実施形態によれば、リスト構造化されたデータからのデータ要素の削除が実行中のスレッドに影響を与えることなく、削除されたデータ要素を迅速に再利用可能状態にすることができる。

第2実施形態

- 30 次に、本発明の第2実施形態に係る情報処理システムについて説明する。本実施形

態では、情報処理システムは、各処理スレッドの開始および終了に応じて、処理状態を管理する処理状態管理部を備える。処理状態管理部は、リスト構造化されたデータからデータ要素を削除するリスト要素削除スレッドによる削除処理において、実行中のスレッドに影響を与えることなく、削除されたデータ要素を迅速に再利用可能な状態にする。

本実施形態によれば、処理状態管理部が複数の処理スレッドを管理することによって、リスト構造化されたデータからのデータ要素の削除が、実行中のスレッドに影響を与えることなく、削除されたデータ要素を迅速に再利用可能な状態にすることができ

情報処理システムの構成

図2は、本実施形態に係る情報処理システム200の構成を示すブロック図である。情報処理システム200は、記憶部220と、処理状態管理部210とを備える。

図2において、リスト要素削除スレッド201は、リスト構造化したデータからデータ要素を削除する処理を含むスレッドである。リスト検索スレッド202、203は、リスト構造化したデータにアクセスする処理を含むスレッドを代表している。情報処理システム200に含まれるどのプロセッサのどのCPU(central processing unit)が、これらスレッドを開始および実行してもよい。ただし、第2実施形態の以下の説明では、説明を簡潔にするため、1つのプロセッサが、並列的にスレッドを実行することを説明する。

処理状態管理部210は、上記各スレッド201~203の開始および終了に応じた処理を実行することによって、スレッドの処理状態を管理する(図4参照)。記憶部220は、リスト構造化したデータを含むデータを記憶する。なお、記憶部220は、例えば1つのプロセッサの主記憶(RAM(random access memory))であっても、ストレージ(ディスクなど)であってもよい。記憶部220は、また、複数のプロセッサに分散されるように構成されても、情報処理システム200内で共有されるように構成されてもよい。

情報処理システムの動作手順

図3は、本実施形態に係る情報処理システム200の動作手順を示すシーケンス図である。図3を参照して、情報処理システム200の動作手順の概略を説明する。なお、図3には、図2に示した各要素間のやりとりを記載したが、並列的に処理されるスレッドの数の制限はない。また、図3においては、全体の流れが明瞭になるように、細かな処理間のやりとりの表示は省略する。また、各スレッドの開始順は以下の通りである。すなわち、まず、リスト検索スレッド202が処理を開始され、次に、リスト要素削除スレッド201が処理を開始し、最後に、リスト検索スレッド203が処理を開始する。

最初に書開始されたリスト検索スレッド202は、まず、ステップS301において、処理状態管理部210に対して自分の順番番号を要求する。処理状態管理部210は、ステップS303において、スレッドを管理する board 配列(図7A 参照)に新たなスレッドの開始を追加し、board 配列の先頭位置ポインタ(head: 以下では、最大順序番号とも呼ぶ)を+1する。そして、処理状態管理部210は、ステップS305において、その順番番号(head+1)をリスト検索スレッド202に返す。リスト検索スレッド202は、ステップS307においてリスト検索処理を開始し、記憶部220に記憶されたリスト構造化されたデータを検索する。

続いて、リスト要素削除スレッド201は、処理を開始し、ステップS309において、記憶部220のリスト構造化されたデータからデータ要素を削除する。かかる削除処理は、ステップS311のように、CAS操作を伴う非特許文献1または2に開示される操作により実施されてよい。ただし、この時点では、リスト要素削除スレッド201は、削除されたデータ要素の再利用は不可の状態、すなわち、内容が変更されないように維持をする。リスト要素削除スレッド201は、削除処理の完了時点であるステップS313において、処理状態管理部210に対して現在の最大順序番号を要求する。処理状態管理部210は、ステップS315において、現在の最大順序番号(本例では、head+1)をリスト要素削除スレッド201に返す。

リスト要素削除スレッド201は、次に、ステップS331において、処理状態管理部210に対して、受け取った最大順序番号(head+1)を送付すると共に、最大順序番号(head+1)を持つスレッドが終了したか否かを問い合わせる。処理状態管理部210は、

ステップS333において、最大順序番号(head+1)を持つスレッドの終了を board 配列内のデータに基づいて判断する。本例では、リスト検索スレッド202は、開始したばかりなのでまだ終了していない。そこで、処理状態管理部210は、ステップS335において、リスト要素削除スレッド201に対して最大順序番号(head+1)を持つスレッドの未終了を通知する。そのため、削除されたデータ要素は、再利用が不可のままであり、その内容が変更されないように維持される。

最後に開始したリスト検索スレッド203は、まず、ステップS321において、処理状態管理部210に対して自分の順番番号を要求する。処理状態管理部210は、ステップS323において、スレッドを管理する board 配列に新たなスレッドの開始を追加し、board 配列の先頭位置ポインタを+1(図3では、処理を明瞭にするため、head+2と示す)する。そして、処理状態管理部210は、ステップS325において、その順番番号(head+2)をリスト検索スレッド203に返す。リスト検索スレッド203は、ステップS327においてリスト検索処理を開始すると共に、記憶部220に記憶されたリスト構造化されたデータを検索する。なお、リスト検索スレッド203の開始のタイミングは、リスト要素削除スレッド201によるデータ要素の削除後であれば、図3に示すタイミングに限定されない。

リスト要素削除スレッド201は、ステップS331-S335を、所定の時間間隔で繰り返す。すなわち、リスト要素削除スレッド201は、所定の時間間隔で最大順序番号を処理状態管理部210に送付すると共に、リスト検索スレッド202までのすべてのスレッドが終了したか否かを問い合わせる。図3においては、リスト検索スレッド202が処理の終了を処理状態管理部210に対して報告しない限り、削除以前の全スレッドの終了状態にならないので、上記終了の報告までは削除したデータ要素の内容が維持される。

ステップS341において、リスト検索スレッド202は、処理を終了して、処理状態管理部210に処理の終了を報告する。処理状態管理部210は、ステップS343において、順序番号(head+1)を付与されたリスト検索スレッド202の終了処理を行なう。ステップS345において、リスト要素削除スレッド201は、最大順序番号(head+1)を送付する。

そして、リスト要素削除スレッド201は、処理状態管理部210に対して、リスト検索スレッド202までのすべてのスレッドが終了したか否かを問い合わせる。この時点では、処理状態管理部210は、ステップS347において、リスト検索スレッド202の終了を判定する(実際には、処理状態管理部210は、リスト要素削除スレッド201の削除以前に開始した全スレッドの終了を判定する)。そして、ステップS349において、処理状態管理部210は、リスト要素削除スレッド201に対して最大順序番号(head+1)を持つスレッドの終了を通知する。

リスト要素削除スレッド201は、最大順序番号(head+1)を持つスレッドの終了の通知を受けて、ステップS351において、記憶部220に対して削除したデータ要素の再利用可能を通知する。処理状態管理部210は、ステップS353に示すように、記憶部220に対し、リスト構造化されたデータからリスト要素削除スレッド201により削除されたデータ要素を、この時点で再利用可能にする。すなわち、処理状態管理部210は、削除されたデータ要素が持つ情報を記録するために割り当てられていたメモリ領域を解放し、未使用な状態にする。

このような動作手順により、リスト構造化されたデータからリスト要素削除スレッド201により削除されたデータ要素の内容は、リスト要素削除スレッド201の削除処理以前に開始された全スレッドが終了するまでは、維持される。また、対象のスレッドが終了すれば、直ちに削除されたデータ要素が再利用可能になる。なお、以上の説明では、リスト要素削除スレッド201の削除処理以前に開始された全スレッドの終了を条件としたが、リスト構造のデータにアクセスしないスレッドの終了は待つ必要はない。したがって、スレッドの処理内容による、さらなる制御も可能である。その場合には、削除されたデータ要素をより早く再利用可能にすることができる。

処理状態管理部の構成

図4は、本実施形態に係る処理状態管理部210の構成を示すブロック図である。

処理状態管理部210は、開始終了処理部410と、終了判定部420と、状態保持部430とを有する。開始終了処理部410は、開始処理部411と終了処理部412とを含

む。終了判定部420は、最大順序番号取得部421と最小順序番号比較部422とを含む。状態保持部430は、最新スレッドに付与された順序番号である最大順序番号431と、それ以前のスレッドがすべて終了しているスレッドの順序番号である最小順序番号432と、付与済および処理済の順序番号を管理する順序番号管理データ433とを保持する。

これらの機能構成部は、それぞれスレッド処理部から引数を持って呼び出され、処理後に戻り値をスレッド処理部に返す。その概略処理を、以下に説明する。

10 開始処理部411は、最大順序番号431と順序番号管理データ433とを参照および更新することにより、呼び出し元のスレッドに順序番号を付与する。終了処理部412は、呼び出し元のスレッドから順序番号を受け取り、最大順序番号431と最小順序番号432と順序番号管理データ433とを参照および更新すると共に、処理の完了した順序番号を最小順序番号432に反映させる。

15

最大順序番号取得部421は、呼び出された時点における最大順序番号431を呼び出し元のスレッドに渡す。最小順序番号比較部422は、呼び出し元のスレッドから渡された順序番号と最小順序番号432とを比較し、その順序番号に対応するスレッドが完了しているかどうかを呼び出し元のスレッドに通知する。

20

本実施形態では、リスト構造データの検索を行うリスト検索スレッドは、検索処理の開始時に開始処理部411を呼び出すことにより、検索処理に対応する順序番号を受け取る。リスト検索スレッドはまた、検索処理の終了時にその検索処理に対応する順序番号を引数として用いて、終了処理部412を呼び出す。

25

リスト構造からデータ要素を削除するリスト要素削除スレッドは、リスト構造データからデータ要素を物理的に削除するアトミックなメモリアクセスを実行した後に、最大順序番号取得部421を呼び出す。これにより、リスト要素削除スレッドは、その時点における最大順序番号431を取得する。その後、リスト要素削除スレッドは、その順序番号を引数として用いて最小順序番号比較部422を呼び出すことにより、その順序番号に対

30

応するスレッドまで完了しているか否かを調べる。これにより、リスト要素削除スレッドは、削除したデータ要素が再利用可能かどうかを判定する。

5 以上の構成により、リスト要素削除スレッドが削除したデータ要素を参照する可能性のあるリスト検索スレッドによる検索処理が終了したら、直ちに、そのデータ要素が再利用可能であることを認識することができるという効果が得られる。また、リスト検索処理完了の通知は、処理状態管理部210を経由して行われるので、リスト検索スレッドとリスト要素削除スレッドとが直接、通信を行う必要がない。したがって、システムのオーバーヘッドを抑えると共に、管理が煩雑になることを防止できるという効果が得られる。

10

以上、処理状態管理部210の動作手順の概略を上記に説明したが、以下に、より具体的な処理状態管理部210の動作手順を説明する。

変数の定義

15 図5は、本実施形態で使用する変数の定義500を示す図である。

本実施形態では、状態保持部430に格納されている最大順序番号431、最小順序番号432、順序番号管理データ433は、それぞれ、全スレッドからアクセス可能な変数である。510に示すように、それぞれの変数は、head、tail、board[SIZE]と表記される。また、配列である順序番号管理データの配列要素数を示す定数は、SIZEと表記される。また、520に示すように、スレッド別の変数として、スレッドに割り当てられる順序番号を格納する変数(myseqと表記)が用意される。さらに、530に示すように、一時記憶用の変数として、board配列要素もしくはその値をコピーする変数(state)と変数(newv)とが用意される。

20

25

操作の定義

図6は、本実施形態で使用する操作の定義600を示す図である。操作の定義600は、操作表記601とその操作内容602とを含む。

30 本実施形態では、基本操作として、以下の3操作が実行される。すなわち、変数vに

対し値“1”を不可分加算する操作610と、変数 v に対して `cmpxchg` 操作を不可分に実行する操作620と、変数 v に対応する配列のインデックス($v \% \text{SIZE}$)を求める操作630である。各々、`atomic_inc(&v)`、`atomic_cmpxchg(&v, o, n)`、`OFFSET(v)`と表記される。

5

データ構成

図7Aは、本実施形態で使用するデータ構成700を示す図である。

図7Aに示すように、順序番号710は、`board` 配列のインデックスに対応するOFFSET
10 値として使用される下位側ビット712と、`board` 配列に格納するTAG情報として使用される上位側ビット711とを含む。また、順序番号管理データ433に相当する`board` 配列720の各要素は、TAG情報724と、REUSE721と、PASSED722とを格納する。PASSED722は、当該配列要素に対応する順序番号のスレッドが未終了であるため
15 追い越されていることを示す1ビットのフラグである(詳細は後述する)。REUSE721は、追い越しが行われた配列要素に対応する順序番号のスレッドが終了したことを示す1ビットのフラグである(詳細は後述する)。723は、他の目的に使用可能なオプションビットであり、本実施形態では使用されない。ここで、`board` 配列720は、その配列があらかじめ決められたアドレス空間に限定される。新たな配列要素は、`board` 配列720のアドレス空間を巡回するように配置される。すなわち、処理状態管理部210は、スレッド
20 の開始順からアドレスを計算し、そのアドレス位置に実行中のスレッドの状態を保持する。

データ表記

図7Bは、本実施形態で使用するデータ表記730を示す図である。データ表記730
25 は、データ表記731に対応づけられたデータ内容732とデータの意味733とを含む。

`TAG(myseq)`および`TAG(board[i])`740は、順序番号(`myseq`)および`board` 配列要素
(`board[i]`, $i=0 \sim \text{SIZE}-1$ の値)から取得したTAG情報を示す。また、`REUSE(state)`75
0は、`board` 配列要素もしくはその値のコピーを格納する変数(`state`)から取得した
30 REUSEビットを示す。また、`PASSED(state)`760は、`board` 配列要素もしくはその値の

コピーを格納する変数 (state) から取得した PASSED ビットを示す。

board 配列

図8Aおよび図8Bとを参照して、図7Aに示した board 配列720の構成例を説明する。

- 5 なお、この2つの例はその特徴的な例であり、この例から board 配列720の構成と動作とが理解可能である。

10 図8Aは、本実施形態に係る board 配列720の構成例810を示す図である。図8Aは、OFFSET(head)が OFFSET(tail)を追い越さない状況における board 配列の一例である。ここで、t は TAG(head)である。

ここで、追い越しについて説明する。上述のように、board 配列720は、その配列があらかじめ決められたアドレス空間に限定される。そのため、終了処理が未完了の board 配列要素が増加すると、board 配列720のアドレス空間が、終了処理が未完了の board 配列要素によって占められてしまう。そこで、開始処理部411が呼び出されてインクリメントされた head に対応する board 配列要素は、board 配列720のアドレス空間を巡回するように配置される。このとき、OFFSET(head)が OFFSET(tail)を「追い越す」と称する。追い越しとは、開始処理部411が呼び出されてインクリメントされた head に対応する board 配列要素が、終了処理未完了の状態であったという事象を意味している。

20 OFFSET(head)が OFFSET(tail)を追い越した状況では、head と tail の差が SIZE 以上になる。この場合、board 配列の全要素が使用されている。OFFSET(head)が OFFSET(tail)を追い越した場合、その配列要素の PASSED ビットがセットされる。そして、その配列要素に対応する順序番号のスレッドの終了処理が完了した場合、当該配列要素の REUSE ビットがセットされる。

25 図8Aに示すとおり、追い越しのない状況では、board 配列で使用されている領域、すなわち、OFFSET(tail)から OFFSET(head)までの配列要素の REUSE ビットと PASSED ビットは、リセットされた状態(図では“0”と表記)である。また、TAG は t もしくは t+1 である。ここで TAG 値が t の場合、開始処理部411によって付与された順序番号が返却されていない、すなわち、その順序番号を引数とした終了処理部412が呼び出されて

30

いないことを意味している。一方、TAG 値が $t+1$ の場合、開始処理部 411 によって付与された順序番号が返却されている、すなわち、その順序番号を引数とした終了処理部 412 が呼び出されたことを意味している。ここで、tail の移動について説明する。tail と等しい順序番号を引数とした終了処理部 412 が呼び出されると、tail の順序番号に対応する board 配列要素 (board[OFFSET{tail}]) の TAG は、 t から $t+1$ に変更される。tail+1 の順序番号に対応する board 配列要素 (board[OFFSET{tail+1}])、図 8A では tail の一行下の TAG は、図 8A では $t+1$ となっており、既に終了処理が完了していることを示している。よって、終了処理部 412 により、tail は、その時点で終了処理の完了していない最も小さい番号である tail+2 の位置 (図 8A では「次の tail」と表記) に移動する。

図 8B は、本実施形態に係る board 配列の他の構成例 820 を示す図である。図 8B は、OFFSET(head) が OFFSET(tail) を追い越した状況、すなわち、head と tail の差が SIZE 以上になった場合を示す。この場合、board 配列の全要素が使用されている。ここで、board[OFFSET(head+1)] から配列最終要素 (board[SIZE-1]) において、TAG 値が t ($=\text{TAG}(\text{head})$) の場合、開始処理部 411 により付与された順序番号を引数とする終了処理が行なわれたことを示す。一方、TAG 値が $t-1$ の場合は、開始処理部 411 により付与された順序番号を引数とする終了処理が行なわれていないことを示す。

上述したように、追い越しとは、開始処理部 411 が呼び出されてインクリメントされた head に対応する board 配列要素が、終了処理未完了状態であったという事象を意味している。このとき、board 配列は、2 巡目以降の配置となる。この場合、開始処理部 411 は、その配列要素の PASSED ビットをセットし (図では“1”と表記)、head に 1 を加えたのち開始処理部 411 を再度実行する。この場合、その配列要素に対応する順序番号は付与されない。

また、配列先頭要素 (board[0]) から board[OFFSET(head)] までの領域では、TAG 値が $t+1$ ($=\text{TAG}(\text{head})+1$) の場合、開始処理部 411 により付与された順序番号を引数とする終了処理が行なわれたことを意味する。一方、TAG 値が t の場合は、開始処理部 411 により付与された順序番号を引数とする終了処理が行なわれていないことを意味

する。また、この領域でTAG値が $t-1$ 以下の場合、その配列要素に追い越し事象が発生したことを示している。その配列要素は、前記の操作により、PASSEDビットが設定された状態となっている。

- 5 この PASSED ビットが“1”に設定された配列要素に対応する順序番号を引数とする終了処理は、REUSE ビットをセット(図では“1”と表記)することにより、その配列要素に対応する順序番号のスレッドの終了処理が完了していることを設定する。このような追い越し処理は、OFFSET(head)が OFFSET(tail)を2回以上追い越した状況、すなわち、head と tail の差が SIZE の2倍以上となった場合でも、同様に行なわれる。

10

情報処理装置のハードウェア構成

- 図9は、本実施形態に係る情報処理装置900のハードウェア構成を示すブロック図である。図9は、本実施形態の並列スレッド処理と処理状態管理部とを実行するハードウェア構成例を示す。ただし、図9の構成は本実施形態の一例に過ぎず、並列スレッド処理と処理状態管理部とが、これらのデータおよびプログラムを一部を共有しながら一部を分散する等、様々な形態が実現可能である。
- 15

- 図9において、CPU910-1~910-nは、演算制御用のプロセッサであり、プログラムを実行することにより情報処理装置900の各機能構成部を実現する。ROM(read only memory)920は、初期データおよびプログラムなどの固定データおよびプログラムを記憶する。通信制御部930は、ネットワークを介して他のプロセッサや通信端末などとデータを送受信する。
- 20

- RAM940は、CPU910-1~910-nが一時記憶のワークエリアとして使用するランダムアクセスメモリである。RAM940には、本実施形態の実現に必要なデータを記憶する領域が確保されている。431は、最大順序番号(head)である。432は、最小順序番号(tail)である。
- 25

- RAM940は、図2にも示したリスト検索スレッド202、203、およびリスト要素削除スレッド201を記憶する。941は、リスト検索スレッド202の順序番号(myseq)である。9
- 30

42は、リスト検索スレッド202の状態変数(state, newv)である。943は、リスト検索スレッド203の順序番号(myseq)である。944は、リスト検索スレッド203の状態変数(state, newv)である。945は、リスト要素削除スレッド201が取得した最大順序番号である。946は、リスト要素削除スレッド201が問い合わせた最小順序番号比較の結果
5 (終了/未終了)である。

ストレージ950には、データベースや各種のパラメータ、あるいは本実施形態の実現に必要な以下のデータまたはプログラムが記憶されている。610は、図6に定義された関数で表わされる操作 atomic_inc(&v)である。620は、図6に定義された関数で表わされる操作 atomic_cmpxchg(&v, o, n)である。630は、図6に定義された関数で表わされる操作 OFFSET(v)である。740は、図7Bに定義された関数で表わされるデータ TAG(myseq)および TAG(board[i])である。750は、図7Bに定義された関数あるいはポインタで表わされるデータ REUSE(state)である。760は、図7Bに定義された関数あるいはポインタで表わされるデータ PASSED(state)である。
10

15

ストレージ950には、以下のプログラムが格納される。951は、全体の処理を実行させる情報処理プログラムである。952は、情報処理プログラム951において使用されるリスト検索スレッドを実行するリスト検索処理プログラムである(図10A参照)。953は、情報処理プログラム951において使用されるリスト要素削除スレッドを実行するリスト要素削除処理プログラムである(図10B参照)。なお、図9には、2つのスレッドしか示さず、他のスレッドに対応するプログラムは省略している。
20

954は、処理状態管理を司る処理状態管理プログラムである。955は、処理状態管理プログラム954において、開始処理を実行する開始処理モジュールである(図11Aおよび図11B参照)。956は、処理状態管理プログラム954において、終了処理を実行する終了処理モジュールである(図12A乃至図12C参照)。957は、処理状態管理プログラム954において、最大順序番号取得処理を実行する最大順序番号取得処理モジュールである(図13参照)。958は、処理状態管理プログラム954において、最小順序番号比較処理を実行する最小順序番号比較処理モジュールである(図14参
30 照)。

図9においては、リスト検索スレッドおよびリスト要素削除スレッドと、各モジュールとが混在して処理されるように図示されている。しかし、図2に示すように、処理状態管理部210に関連するデータおよびプログラムを、リスト検索スレッドおよびリスト要素削除スレッドなどのデータ処理に関連するデータおよびプログラムと分離して、それぞれを独立して実行させてもよい。その場合には、例えば、プロセッサやCPUに役割分担させる構成となる。

なお、図9には、本実施形態に必須なデータやプログラムのみが示されており、OSなどの汎用のデータやプログラムは図示されていない。

入力インタフェース960は、各種入力機器からの入力データをインタフェースする。入力インタフェース960には、例えば、キーボード961や、ポインティングデバイス(PD)962や、記憶媒体963などが、接続可能である。また、出力インタフェース970は、処理データを出力する。出力インタフェース970には、例えば、表示部971やプリンタ972などが接続される。

図1に示す情報処理装置100のスレッド制御部101およびデータ要素制御部102は、コンピュータにより実現した場合、図9に例示するハードウェア構成を有する。図9に示す構成は、CPU(Central Processing Unit)910-1~910-n、ROM920、通信制御部930、RAM940、ストレージ950およびストレージ950に含まれるプログラムを備える。CPU910-1~910-nは、各種ソフトウェア・プログラム(コンピュータ・プログラム)を実行することにより、情報処理装置100の全体的な動作を司る。本実施形態および以下に示す他の実施形態において、CPU910-1~910-nが、RAM940等の記憶媒体を適宜参照しながら、情報処理装置100が備える各機能(各部)のソフトウェア・プログラムを実行する。

より具体的には、CPU910-1~910-nは、RAM940等の記憶媒体を適宜参照しながら、情報処理装置100が備える、図4に示す処理状態管理部210の機能を実施するソフトウェア・プログラムを実行することにより、開始終了処理部410、終了判

定部420および状態保持部430等のソフトウェア・プログラムを実行する。

リスト検索スレッドの処理手順

図10Aは、本実施形態に係る情報処理装置によるリスト検索スレッド202または203の処理手順を示すフローチャートである。

リスト検索スレッド202を実行する、例えばCPU910-1は、まず、開始処理部411を起動する。開始処理部411は、リスト検索処理を一意に識別する順序番号を取得する(ステップS1011)。続いて、CPU910-1は、リスト検索を行う(ステップS1013)。リスト検索が終了すると、CPU910-1は、終了処理部412を起動する。これにより、リスト検索スレッド202を実行するCPU910-1は、前記順序番号に対応付けられた処理が完了したことを処理状態管理部210に通知する(ステップS1015)。

リスト要素削除スレッドの処理手順

図10Bは、本実施形態に係る情報処理装置によるリスト要素削除スレッド201の処理手順を示すフローチャートである。

リスト要素削除スレッド203を実行する、例えばCPU910-1は、まず、リスト構造データからデータ要素を削除する(ステップS1021)。続いて、CPU910-1は、最大順序番号取得部421を起動する。最大順序番号取得部421は、その時点までに開始された検索処理に対応する順序番号の内、最大の順序番号を取得する(ステップS1023)。続いて、最大順序番号取得部421は、取得した最大順序番号を最小順序番号比較部422に通知する。最小順序番号比較部422は、取得した順序番号以下の番号に対応付けられた検索処理が完了しているかどうかを調べる(ステップS1025)。比較の結果、順序番号以下の番号に対応付けられた検索処理が完了していない場合、最小順序番号比較部422は、ステップS1025の最小順序番号比較処理を繰り返す。比較の結果、順序番号以下の番号に対応付けられた検索処理が完了している場合、最小順序番号比較部422は、削除したデータ要素を再利用可能状態にする(ステップS1027)。順序番号よりも小さい番号に対応付けられたすべての検索処理が完了していることは、削除したデータ要素を参照する可能性のあるスレッドが存在していない

ことを示す。

開始処理

図11Aおよび図11Bは、図10Aに示した本実施形態に係る開始処理S1011の処理手順を示すフローチャートである。開始処理においては、この処理内でのみ使用される、board 配列要素のコピーを格納する変数として state が使用される。なお、図中、“/X”は、Xの否定を示す。また、“actual”は、不可分な cmpxchg 操作からの戻り値を示す。

10 開始処理部411は、まず、最大順序番号(head)に対して値“1”を不可分加算し、加算前の値を変数 myseq に代入する(ステップS1101)。開始処理部411は、board 配列の OFFSET(myseq)番目の配列要素の値を state に代入する(ステップS1103)。開始処理部411は、変数 state の REUSE ビットが“0”であり、かつ state の TAG 情報と myseq の TAG 情報とが等しいという条件が成立しているかどうかを調べる(ステップS1105)。ステップS1105の条件が成立している場合、開始処理部411は、戻り値を myseq として動作を終了する(ステップS1107)。

一方、ステップS1105の条件が成立していない場合、開始処理部411は、state の REUSE ビットを調べる(ステップS1109)。その結果、REUSE ビットが“0”でない場合(ステップS1109の判断が yes の場合)、開始処理部411は、state の TAG 情報と myseq の TAG 情報とが等しいかどうかを調べる(ステップS1111)。これらの値が等しい場合、開始処理部411は、処理をステップS1101に戻す。また、両者が異なる場合、開始処理部411は、board 配列の OFFSET(myseq)番目の配列要素の値を、値 state から TAG(myseq)に変更する不可分な cmpxchg 操作を実施する(ステップS1113)。

25 そして、開始処理部411は、その cmpxchg 操作が成功したかどうかを調べる(ステップS1115)。cmpxchg 操作が成功した場合、開始処理部411は、ステップS1107以降の処理を実行する。一方、cmpxchg 操作が失敗した場合、開始処理部411は、cmpxchg 操作実行時において board 配列の OFFSET(myseq)番目の配列要素に格納されていた値を state に代入する(ステップS1123)。そして、開始処理部411は、ステップS1105以降の処理を実行する。

30

ステップS1109において、REUSEビットが“0”であった場合（ステップS1109の判断がnoの場合）、開始処理部411は、stateのPASSEDビットを調べる（ステップS1117）。その結果、PASSEDビットが“0”でない場合（ステップS1117の判断がnoの場合）、開始処理部411は、処理をステップS1101に戻す。一方、PASSEDビットが“0”の場合（ステップS1117の判断がyesの場合）、開始処理部411は、board配列のOFFSET(myseq)番目の配列要素の値を、値stateから、stateにPASSEDフラグを設定した値に変更する不可分なcmpxchg操作を実施する（ステップS1119）。そして、開始処理部411は、そのcmpxchg操作が成功したかどうかを調べる（ステップS1121）。cmpxchg操作が成功した場合、開始処理部411は、処理をステップS1101に戻す。一方、cmpxchg操作が失敗した場合、開始処理部411は、cmpxchg操作の実行時においてboard配列のOFFSET(myseq)番目の配列要素に格納されていた値をstateに代入する（ステップS1123）。そして、開始処理部411は、ステップS1105以降の処理を実行する。

15

終了処理

図12A乃至図12Cは、図10Aに示した本実施形態に係る終了処理S1015の処理手順を示すフローチャートである。終了処理においては、この処理内でのみ使用される、board配列要素のコピーを格納する変数としてstateおよびnewvが使用される。なお、図中、“/X”は、Xの否定を表す。また、“actual”は、不可分なcmpxchg操作からの戻り値を示す。

20

終了処理部412は、まず、board配列のOFFSET(myseq)番目の配列要素をstateに代入する（ステップS1201）。次に、終了処理部412は、stateのPASSEDビットを調べる（ステップS1203）。その結果、PASSEDビットが“0”の場合（ステップS1203の判断がyesの場合）、終了処理部412は、stateに“1”を加えた値をnewvに設定する（ステップS1205）。また、PASSEDビットが“0”でない場合（ステップS1203の判断がnoの場合）、終了処理部412は、stateに対してREUSEビットを設定した値をnewvに設定する（ステップS1207）。ステップS1203の判定がいずれの場合も、終了処理部412は、board配列のOFFSET(myseq)番目の配列要素の値を、値stateから、newv

30

に変更する不可分な cmpxchg 操作を実施する(ステップS1209)。そして、終了処理部412は、その cmpxchg 操作が成功したかどうかを調べる(ステップS1211)。

5 cmpxchg 操作が失敗した場合、終了処理部412は、cmpxchg 操作実行時において board 配列の OFFSET(myseq)番目の配列要素に格納されていた値を state に代入する(ステップS1213)。そして、終了処理部412は、ステップS1203以降の処理を実行する。一方、cmpxchg 操作が成功した場合、終了処理部412は、myseq と tail との値が等しいかどうかを調べる(ステップS1215)。両者の値が異なる場合、終了処理部412は、動作を終了する。両者の値が一致している場合、終了処理部412は、tail
10 に“1”を加えるため不可分な cmpxchg 操作を実施し(ステップS1217)、その cmpxchg 操作が成功したかどうかを調べる(ステップS1219)。cmpxchg 操作が失敗した場合、終了処理部412は、動作を終了する。

一方、ステップS1219の cmpxchg 操作が成功した場合、終了処理部412は、
15 myseq 値に“1”を加え(ステップS1221)、board 配列の OFFSET(myseq)番目の配列要素に格納されていた値を state に代入する(ステップS1223)。続いて、終了処理部412は、変数 state の REUSE ビットが“0”であり、かつ state の TAG 情報と myseq の TAG 情報とが等しいという条件が成立しているかどうかを調べる(ステップS1225)。その結果、ステップS1225の条件が成立している場合、終了処理部412は、動作を終了する。また、ステップS1225の条件が成立していない場合、終了処理部412は、
20 変数 state の REUSE ビットが“0”でなく、かつ、state の TAG 情報と myseq の TAG 情報とが異なっているという条件が成立しているかどうかを調べる(ステップS1227)。その結果、ステップS1227の条件が成立していない場合、終了処理部412は、ステップS1215以降の処理を実行する。

25

一方、ステップS1227の条件が成立している場合、終了処理部412は、board 配列の OFFSET(myseq)番目の配列要素の値を、値 state から、myseq の TAG 値に REUSE ビットと PASSED ビットとを設定した値に変更する、不可分な cmpxchg 操作を実施する(ステップS1229)。そして、終了処理部412は、その cmpxchg 操作が成功したかどうかを調べる(ステップS1231)。cmpxchg 操作が成功した場合、終了処理部412は、
30

処理をステップS1215に戻す。一方、cmpxchg 操作が失敗した場合、終了処理部412は、cmpxchg 操作実行時においてboard 配列の OFFSET(myseq)番目の配列要素に格納されていた値の TAG 値と myseq の TAG 情報を比較する(ステップS1233)。その結果、これらの値が異なる場合、終了処理部412は、処理をステップS1215に戻す。

5 これらの値が等しい場合、終了処理部412は、動作を終了する。

最大順序番号取得処理

図13は、図10Bに示した本実施形態に係る最大順序番号取得処理S1023の処理手順を示すフローチャートである。

10

最大順序番号取得部421は、起動すると、起動時における head の値を取得する。そして、最大順序番号取得部421は、取得した head の値を戻り値とし、動作を終了する(ステップS1301)。

15 最小番号比較処理

図14は、図10Bに示した本実施形態に係る最小番号比較処理S1025の処理手順を示すフローチャートである。

20 最小順序番号比較部422は、まず、引数として渡された myseq 値と起動時における tail の値を比較する(ステップS1401)。その結果、tail 値と myseq 値とが等しい、もしくは、tail 値の方が大きい場合、最小順序番号比較部422は、戻り値を真(true)として、動作を終了する(ステップS1403)。一方、tail 値の方が myseq 値より小さい場合、最小順序番号比較部422は、戻り値を偽(false)として、動作を終了する(ステップS1405)。

25

なお、順序番号を計算機で扱うことのできる整数値として表現することに関係する既存技術として、物理的な表現に関する数値のオーバフローを考慮に入れた大小比較方法が存在している。一例として、Linux kernel の source code では、

```
#define UINT_CMP_LT(a,b) (UINT_MAX/2 < (a)-(b))
```

というマクロ定義がある

30 (UINT_MAX は UINT 型で表現可能な数値の最大値)。本実施形態における順序番号

の比較操作として、このようなオーバーフローを考慮に入れた2つの整数値の大小比較方法を採用することにより、順序番号のオーバーフローを許容する。すなわち、物理的な表現ではオーバーフロー発生時に順序番号値は減少するが、論理的には一意に増加していると解釈できる順序番号を使用した処理状態管理も、本発明に含まれる。

5

第3実施形態

次に、本発明の第3実施形態に係る情報処理システムについて説明する。

情報処理システムの構成

10 図15は、本実施形態に係る情報処理システム1500の構成を示すブロック図である。

図15に示すように、プロセッサA1510およびプロセッサB1520は、同じ記憶部1530を共有して、処理を実行する。図15では、プロセッサA1510がリスト要素削除スレッド1511を実行している。同時に、プロセッサB1520が、リスト検索スレッド1521を並列的に実行している。

本実施形態に係る情報処理システムは、上記第2実施形態と比べると、各プロセッサが状態保持部を共有する一方で、それぞれが処理状態管理をする点で異なる。その他の構成および動作は、第2実施形態と同様であるため、詳しい説明を省略する。

20

かかる処理環境では、1つの処理状態管理部が、複数のプロセッサが処理する複数のスレッドの管理を行ったり、各スレッドからの問合せを処理したりすることは、オーバーヘッドが大きくなる。したがって、図15に示す情報処理システム1500においては、各プロセッサが、(状態保持部を除く)開始終了処理部410と終了判定部420とを有する処理状態管理部1512と1522とを備える。状態保持部1532は、プロセッサおよびスレッドにより共有されて使用されるため、リスト構造化されたデータ1531と共に記憶部1530に格納される。

本実施形態によれば、各プロセッサによる分散管理によって、リスト構造化されたデータからのデータ要素の削除が、実行中のスレッドに影響を与えることなく、削除された

30

データ要素を迅速に再利用可能状態にすることができる。

なお、各プロセッサが状態保持部を備えると共に、該状態保持部が常に同一情報を有するように構成されてもよい。

5

他の実施形態

以上、本発明の実施形態について詳述したが、それぞれの実施形態に含まれる別々の特徴を如何様に組み合わせたシステムまたは装置も、本発明の範疇に含まれる。

10

本発明によれば、複数のスレッドが並列的に複数のデータ要素を、リストなどの構造データに動的に挿入または削除することのできる情報処理システムにおいて、削除したデータ要素が再利用可能か否かの判定を速やかに、かつ、効率的に行なうことのできる他の用途に適用できる。

15

また、本発明は、複数の機器から構成されるシステムに適用されてもよいし、単体の装置に適用されてもよい。さらに、本発明は、実施形態の機能を実現する制御プログラムが、システムあるいは装置に直接あるいは遠隔から供給される場合にも適用可能である。したがって、本発明の機能をコンピュータで実現するために、コンピュータにインストールされる制御プログラム、あるいはその制御プログラムを格納した媒体、その制御プログラムをダウンロードさせるWWW(World Wide Web)サーバも、本発明の範疇に含まれる。

20

なお、上述した実施形態を例に説明した本発明は、上述した情報処理装置を、図9
25 に示すCPU910-1~910-nが実行する一例として、ソフトウェア・プログラムによって実現する場合について説明した。しかしながら、図1、図2、図4および図15に示す各ブロックに示す機能は、一部または全部を、ハードウェアとして実現してもよい。

また、上述した実施形態を例に説明した本発明は、上述した情報処理装置900に
30 対して、その説明において参照したフローチャート(図10Aないし図14)の機能を実現

可能なコンピュータ・プログラムを供給した後、そのコンピュータ・プログラムを、情報処理装置900のCPU910-1~910-nに読み出して実行することによって達成される。

- 5 また、係る供給されたコンピュータ・プログラムは、読み書き可能なメモリ(一時記憶媒体)またはハードディスク装置等のコンピュータ読み取り可能な記憶デバイスに格納すればよい。そして、このような場合において、本発明は、係るコンピュータ・プログラムを表すコード或いは係るコンピュータ・プログラムを格納した記憶媒体によって構成されると捉えることができる。

10

産業上の利用可能性

本発明は、例えば、複数のスレッドを並列的に実行する情報処理装置に適用できる。

- 15 実施形態の他の表現

上記の実施形態の一部または全部は、以下の付記のようにも記載されうるが、以下には限られない。

(付記1)

- 20 複数のスレッドを開始するに際して、各スレッドに識別子を付与し、前記各スレッドの終了に際して、前記識別子を伴って終了を通知するスレッド制御手段と、
- リスト構造化されたデータからデータ要素を削除する削除スレッドが実行された場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、前記削除スレッドの削除処理以前に開始したすべてのスレ
- 25 ッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にするデータ要素制御手段と
- を備える情報処理装置。

(付記2)

- 30 前記スレッド制御手段は、前記削除スレッドによる削除処理の完了時点における最

新スレッドの識別子を取得し、

前記データ要素制御手段は、前記最新スレッドおよびそれ以前に実行を開始したすべてのスレッドが終了したか否かの問い合わせに应答して、前記スレッド制御手段によって前記最新スレッドおよびそれ以前に実行を開始したすべてのスレッドの終了が前
5 記識別子を伴って通知されているか否かを調べ、前記終了が通知されている場合は、
削除されたデータ要素を再利用可能状態にする付記1に記載の情報処理装置。

(付記3)

前記スレッド制御手段は、前記削除スレッドを除く、前記リスト構造化されたデータを
アクセスする検索スレッドの開始に際して、識別子を付与し、前記検索スレッドの終了
10 に際して、前記識別子を伴って終了を通知する付記1または2に記載の情報処理装
置。

(付記4)

前記スレッド制御手段は、前記検索スレッドの検索処理の開始に際して、当該検索
スレッドに識別子を付与し、前記検索スレッドの終了に際して、当該検索スレッドから、
15 前記識別子を受け取る付記3に記載の情報処理システム。

(付記5)

前記スレッド制御手段は、

前記識別子に含まれる開始順に基づいてアドレスを計算し、そのアドレス位置に実
行中のスレッドの状態を保持する状態保持手段をさらに備え、

20 前記状態保持手段に格納される、前記識別子に対応するスレッドの終了を示す情
報を制御し、

前記データ要素制御手段は、前記状態保持手段に格納される、前記識別子に対応
するスレッドの終了を示す情報に基づいて、削除されたデータ要素が再利用可能かど
うかを決定する付記1乃至3のいずれか1項に記載の情報処理装置。

25 (付記6)

前記状態保持手段は、あらかじめ決められたアドレス空間を有し、スレッドの開始順
に前記アドレス空間を巡回するように各スレッドの状態を示す配列要素を保持し、

各配列要素は、

2 巡目以降において、終了が通知されていないスレッドに対応する配列要素に対し
30 て追い越しが行なわれたことを示すフラグと、

前記追い越しが行なわれた配列要素に対応するスレッドの終了が通知されたことを示すフラグとを含む

付記4に記載の情報処理装置。

(付記7)

5 前記スレッド制御手段は、

最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前のスレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの開始および終了を示す情報を含む順序番号管理データとを保持する状態保持手段と、

10 開始されたスレッドに対して一意に増加する順序番号を付与する開始処理部と、順序番号に対応するスレッドが終了したことを順序番号管理データに反映する終了処理部とを含む開始終了処理手段とを備え、

前記リスト構造化されたデータをアクセスする検索スレッドの検索処理の開始に際して前記開始処理部を呼び出し、前記検索スレッドの検索処理の終了に際して前記終了処理部を呼び出し、

15 前記データ要素制御手段は、

前記削除スレッドからの要求に応答して、前記状態保持手段から前記最大順序番号を取得してそれを返す最大順序番号取得部と、前記削除スレッドから取得した最大順序番号と前記状態保持手段に保持された最小順序番号とを比較する最小順序番号比較部とを含む終了判定手段とを備え、

前記削除スレッドのデータ要素の削除処理の後に前記最大順序番号取得部と前記最小順序番号比較部とを呼び出すことにより、前記削除スレッドが削除したデータ要素が再利用可能かどうかの判定を実施する付記1ないし5のいずれか1項記載の情報処理装置。

25 (付記8)

リスト構造化されたデータの検索を行う検索スレッドと、リスト構造化されたデータからデータ要素を削除する削除スレッドとを実行する処理手段と、

前記処理手段による複数のスレッドの実行を管理する処理状態管理手段とを備え、前記処理状態管理手段は、

30 最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前の

スレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの開始および終了を示す情報を含む順序番号管理データとを保持する状態保持手段と、

開始されたスレッドに対して一意に増加する順序番号を付与する開始処理部と、順序番号に対応するスレッドが終了したことを順序番号管理データに反映する終了処理部とを含む開始終了処理手段とを備え、

前記削除スレッドからの要求に应答して、前記状態保持手段から前記最大順序番号を取得してそれを返す最大順序番号取得部と、前記削除スレッドから取得した最大順序番号と前記状態保持手段に保持された最小順序番号とを比較する最小順序番号比較部とを含む終了判定手段とを備え、

前記検索スレッドの検索処理の開始に際して前記開始処理部を呼び出し、前記検索スレッドの検索処理の終了に際して前記終了処理部を呼び出し、

前記削除スレッドのデータ要素の削除処理の後に前記最大順序番号取得部と前記最小順序番号比較部とを呼び出すことにより、前記削除スレッドが削除したデータ要素が再利用可能かどうかの判定を実施する情報処理システム。

(付記9)

複数のスレッドを開始するに際して、各スレッドに識別子を付与し、

前記各スレッドの終了に際して、前記識別子を伴って終了を通知し、

リスト構造化されたデータからデータ要素を削除する削除スレッドが実行される場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、

前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にする情報処理方法。

(付記10)

最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前のスレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの開始および終了を示す情報を含む順序番号管理データとを、状態保持手段に保持し、

開始されたスレッドに対して一意に増加する順序番号を付与し、
スレッドからの終了の通知を受けて、順序番号に対応するスレッドが終了したことを
順序番号管理データに反映し、

リスト構造化されたデータからデータ要素を削除する削除スレッドからの要求に
5 応答して、前記状態保持手段から前記最大順序番号を取得してそれを返し、

前記削除スレッドから取得した最大順序番号と、前記最小順序番号とを比較して、
前記最大順序番号と前記最小順序番号とが一致する場合、または前記最大順序番
号が前記最小順序番号よりも大きい場合に、前記削除スレッドが削除したデータ要素
を再利用可能な状態にする請求項8記載の情報処理方法。

10 (付記11)

複数のスレッドを開始するに際して、各スレッドに識別子を付与する処理と、
前記各スレッドの終了に際して、前記識別子を伴って終了を通知する処理と、
リスト構造化されたデータからデータ要素を削除する削除スレッドが実行される場合
15 に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始
したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変
更できない状態に維持する処理と、

前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子
を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にする
処理と

20 をコンピュータに実行させる制御プログラムを記録する制御プログラム記録媒体。

以上、実施形態を参照して本願発明を説明したが、本願発明は上記実施形態に限
定されるものではない。本願発明の構成や詳細には、本願発明の範囲内で当業者
が理解しうる様々な変更をすることができる。

25

この出願は、2011年7月29日に出願された日本出願特願2011-167691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

30 100 情報処理システム

- 101 スレッド制御部
- 102 データ要素制御部
- 210 処理状態管理部
- 220 記憶部
- 5 410 処理状態管理部
- 420 終了判定部
- 430 状態保持部
- 411 開始処理部
- 412 終了処理部
- 10 421 最大順序番号取得部
- 422 最小順序番号比較部
- 431 最大順序番号
- 432 最小順序番号
- 433 順序番号管理データ

請求の範囲

[請求項1]

複数のスレッドを開始するに際して、各スレッドに識別子を付与し、前記各スレッドの終了に際して、前記識別子を伴って終了を通知するスレッド制御手段と、

- 5 リスト構造化されたデータからデータ要素を削除する削除スレッドが実行された場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を
- 10 再利用可能な状態にするデータ要素制御手段と
- を備える情報処理装置。

[請求項2]

- 前記スレッド制御手段は、前記削除スレッドによる削除処理の完了時点における最新スレッドの識別子を取得し、
- 15

- 前記データ要素制御手段は、前記最新スレッドおよびそれ以前に実行を開始したすべてのスレッドが終了したか否かの問い合わせに回答して、前記スレッド制御手段によって前記最新スレッドおよびそれ以前に実行を開始したすべてのスレッドの終了が前記識別子を伴って通知されているか否かを調べ、前記終了が通知されている場合は、
- 20 削除されたデータ要素を再利用可能状態にする請求項1に記載の情報処理装置。

[請求項3]

- 前記スレッド制御手段は、前記削除スレッドを除く、前記リスト構造化されたデータをアクセスする検索スレッドの開始に際して、識別子を付与し、前記検索スレッドの終了
- 25 に際して、前記識別子を伴って終了を通知する請求項1または2に記載の情報処理装置。

[請求項4]

- 前記スレッド制御手段は、
- 30 前記識別子に含まれる開始順に基づいてアドレスを計算し、そのアドレス位置に実

行中のスレッドの状態を保持する状態保持手段をさらに備え、

前記状態保持手段に格納される、前記識別子に対応するスレッドの終了を示す情報を制御し、

- 5 前記データ要素制御手段は、前記状態保持手段に格納される、前記識別子に対応するスレッドの終了を示す情報に基づいて、削除されたデータ要素が再利用可能かどうかを決定する請求項1乃至3のいずれか1項に記載の情報処理装置。

[請求項5]

- 10 前記状態保持手段は、あらかじめ決められたアドレス空間を有し、スレッドの開始順に前記アドレス空間を巡回するように各スレッドの状態を示す配列要素を保持し、

各配列要素は、

2巡目以降において、終了が通知されていないスレッドに対応する配列要素に対して追い越しが行なわれたことを示すフラグと、

- 15 前記追い越しが行なわれた配列要素に対応するスレッドの終了が通知されたことを示すフラグとを含む

請求項4に記載の情報処理装置。

[請求項6]

前記スレッド制御手段は、

- 20 最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前のスレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの開始および終了を示す情報を含む順序番号管理データとを保持する状態保持手段と、

- 25 開始されたスレッドに対して一意に増加する順序番号を付与する開始処理部と、順序番号に対応するスレッドが終了したことを順序番号管理データに反映する終了処理部とを含む開始終了処理手段とを備え、

前記リスト構造化されたデータをアクセスする検索スレッドの検索処理の開始に際して前記開始処理部を呼び出し、前記検索スレッドの検索処理の終了に際して前記終了処理部を呼び出し、

- 30 前記データ要素制御手段は、

前記削除スレッドからの要求に応答して、前記状態保持手段から前記最大順序番号を取得してそれを返す最大順序番号取得部と、前記削除スレッドから取得した最大順序番号と前記状態保持手段に保持された最小順序番号とを比較する最小順序番号比較部とを含む終了判定手段とを備え、

- 5 前記削除スレッドのデータ要素の削除処理の後に前記最大順序番号取得部と前記最小順序番号比較部とを呼び出すことにより、前記削除スレッドが削除したデータ要素が再利用可能かどうかの判定を実施する請求項1ないし5のいずれか1項記載の情報処理装置。

10 [請求項7]

リスト構造化されたデータの検索を行う検索スレッドと、リスト構造化されたデータからデータ要素を削除する削除スレッドとを実行する処理手段と、

前記処理手段による複数のスレッドの実行を管理する処理状態管理手段とを備え、前記処理状態管理手段は、

- 15 最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前のスレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの開始および終了を示す情報を含む順序番号管理データとを保持する状態保持手段と、

- 20 開始されたスレッドに対して一意に増加する順序番号を付与する開始処理部と、順序番号に対応するスレッドが終了したことを順序番号管理データに反映する終了処理部とを含む開始終了処理手段とを備え、

- 25 前記削除スレッドからの要求に応答して、前記状態保持手段から前記最大順序番号を取得してそれを返す最大順序番号取得部と、前記削除スレッドから取得した最大順序番号と前記状態保持手段に保持された最小順序番号とを比較する最小順序番号比較部とを含む終了判定手段とを備え、

前記検索スレッドの検索処理の開始に際して前記開始処理部を呼び出し、前記検索スレッドの検索処理の終了に際して前記終了処理部を呼び出し、

- 30 前記削除スレッドのデータ要素の削除処理の後に前記最大順序番号取得部と前記最小順序番号比較部とを呼び出すことにより、前記削除スレッドが削除したデータ要素が再利用可能かどうかの判定を実施する情報処理システム。

[請求項8]

複数のスレッドを開始するに際して、各スレッドに識別子を付与し、

前記各スレッドの終了に際して、前記識別子を伴って終了を通知し、

- 5 リスト構造化されたデータからデータ要素を削除する削除スレッドが実行される場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持し、

- 10 前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にする情報処理方法。

[請求項9]

- 15 最後に開始されたスレッドに付与された識別子である最大順序番号と、それ以前の
スレッドがすべて終了しているスレッドの順序番号である最小順序番号と、各スレッドの
開始および終了を示す情報を含む順序番号管理データとを、状態保持手段に保持
し、

開始されたスレッドに対して一意に増加する順序番号を付与し、

- 20 スレッドからの終了の通知を受けて、順序番号に対応するスレッドが終了したことを
順序番号管理データに反映し、

リスト構造化されたデータからデータ要素を削除する削除スレッドからの要求に
応答して、前記状態保持手段から前記最大順序番号を取得してそれを返し、

- 25 前記削除スレッドから取得した最大順序番号と、前記最小順序番号とを比較して、
前記最大順序番号と前記最小順序番号とが一致する場合、または前記最大順序番
号が前記最小順序番号よりも大きい場合に、前記削除スレッドが削除したデータ要素
を再利用可能な状態にする請求項8記載の情報処理方法。

[請求項10]

複数のスレッドを開始するに際して、各スレッドに識別子を付与する処理と、

- 30 前記各スレッドの終了に際して、前記識別子を伴って終了を通知する処理と、

リスト構造化されたデータからデータ要素を削除する削除スレッドが実行される場合に、前記識別子を伴った終了の通知により前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が確認されるまで、前記削除されたデータ要素の内容を変更できない状態に維持する処理と、

- 5 前記削除スレッドの削除処理以前に開始したすべてのスレッドの終了が前記識別子を伴って通知されている場合、前記削除されたデータ要素を再利用可能な状態にする処理と

をコンピュータに実行させる制御プログラムを記録する制御プログラム記録媒体。

図1

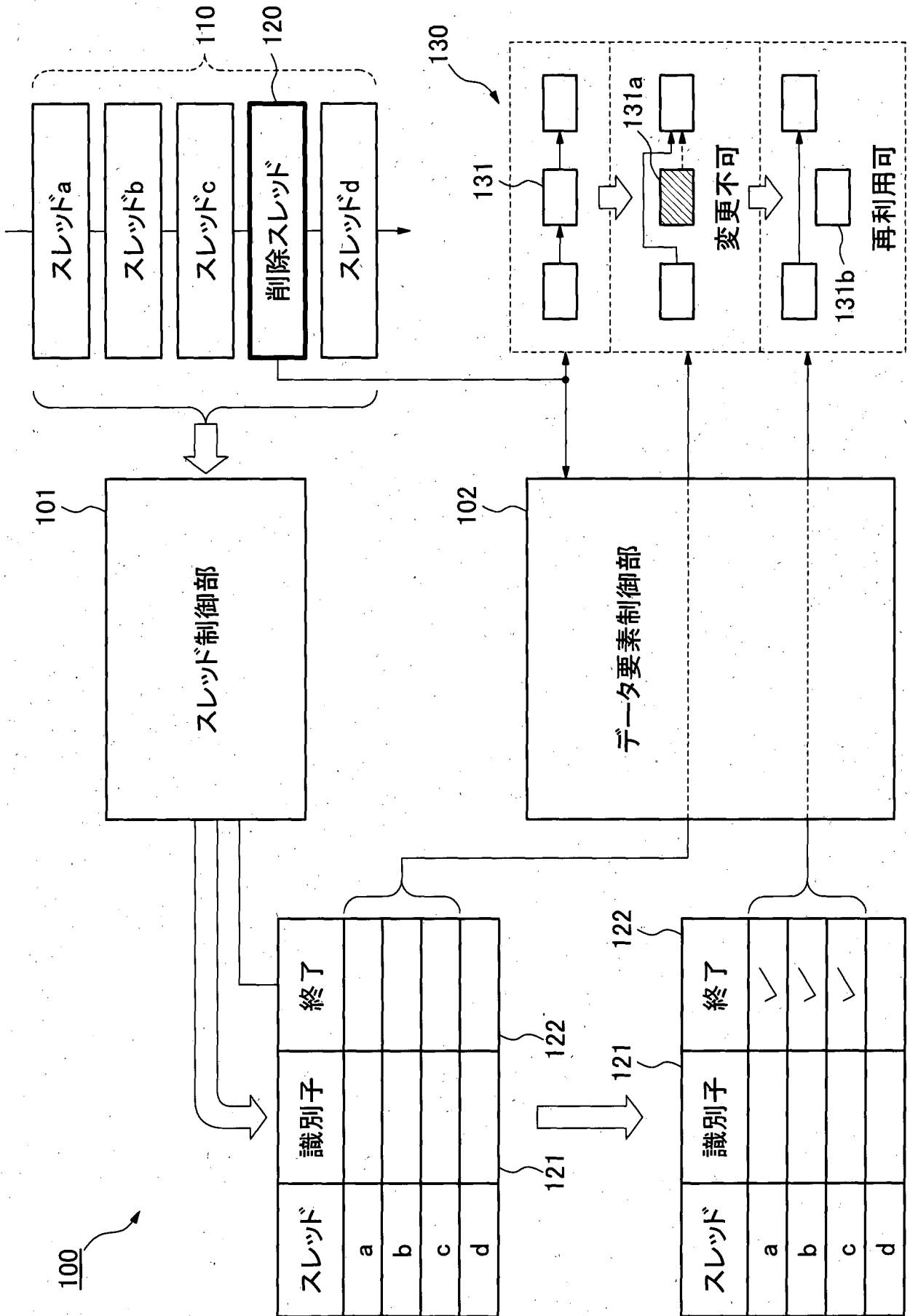


図2

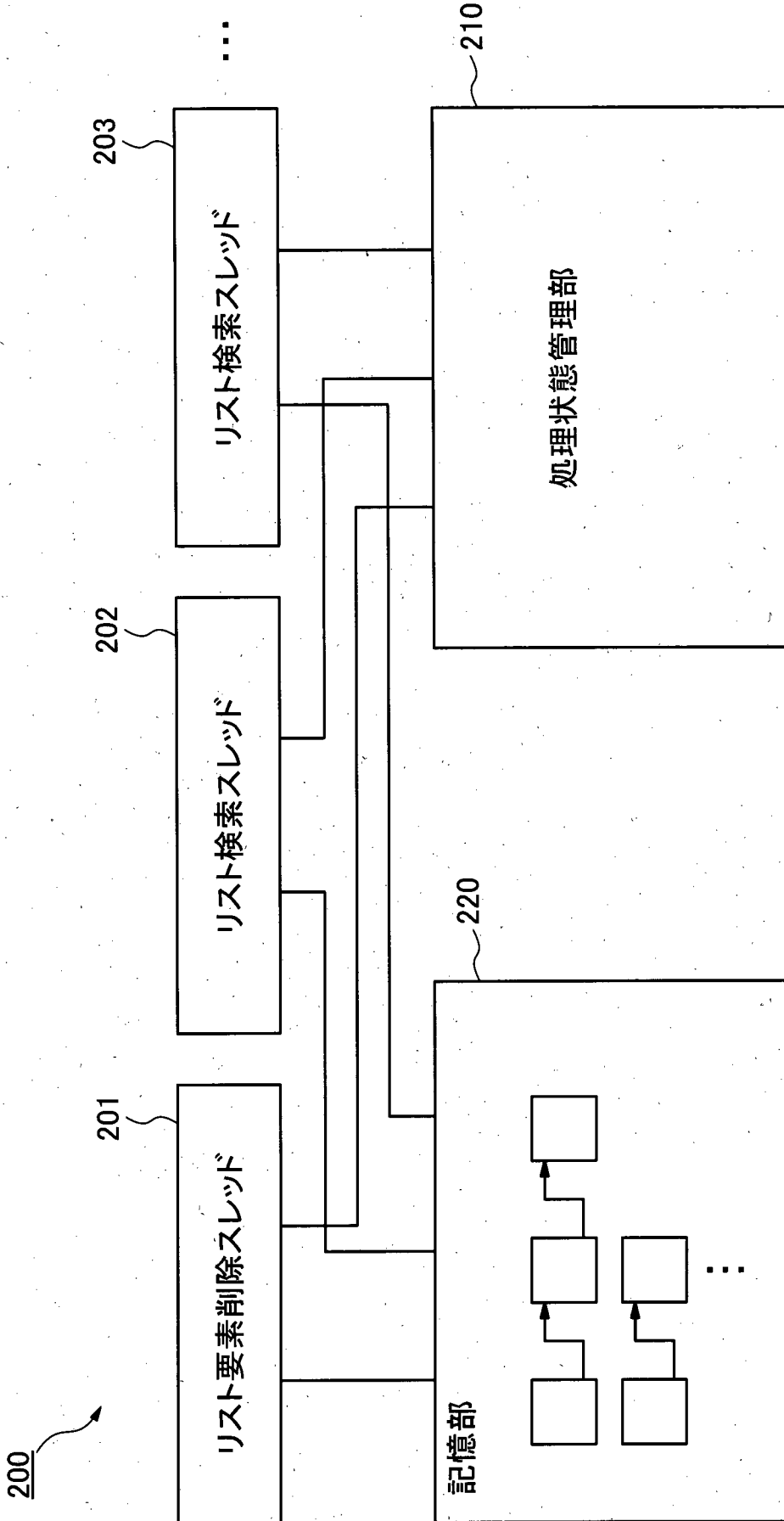


図3

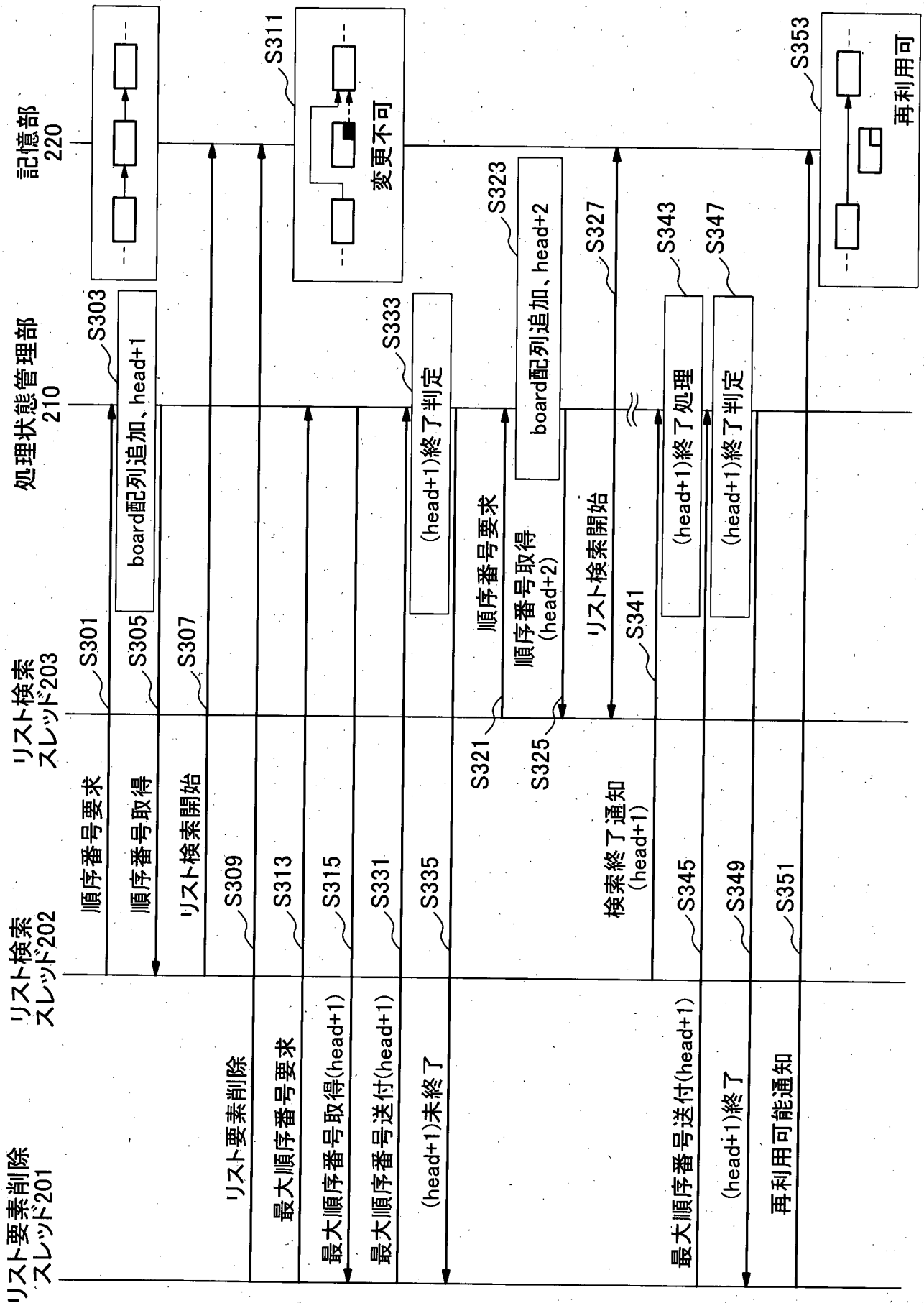


図4

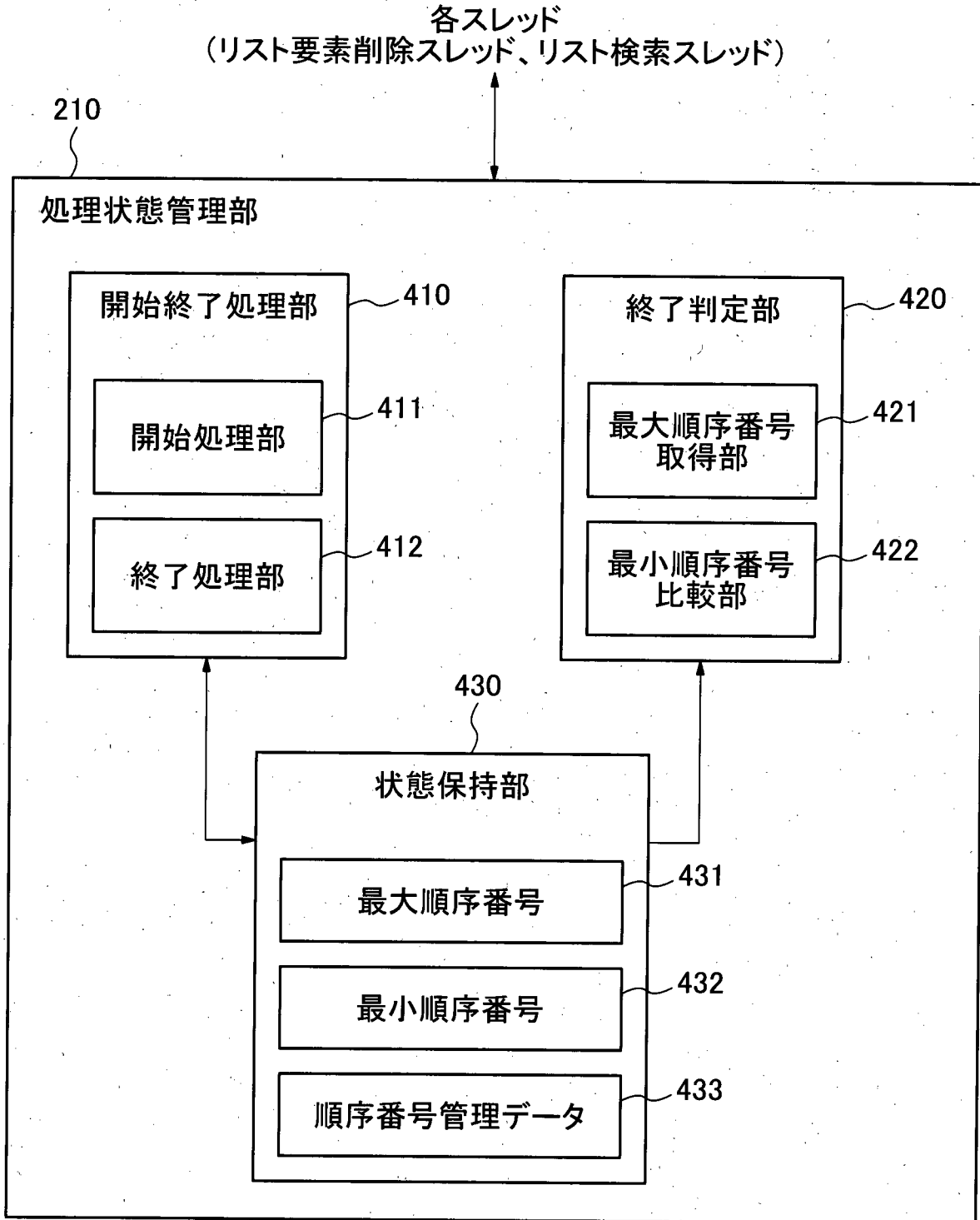


図5

500

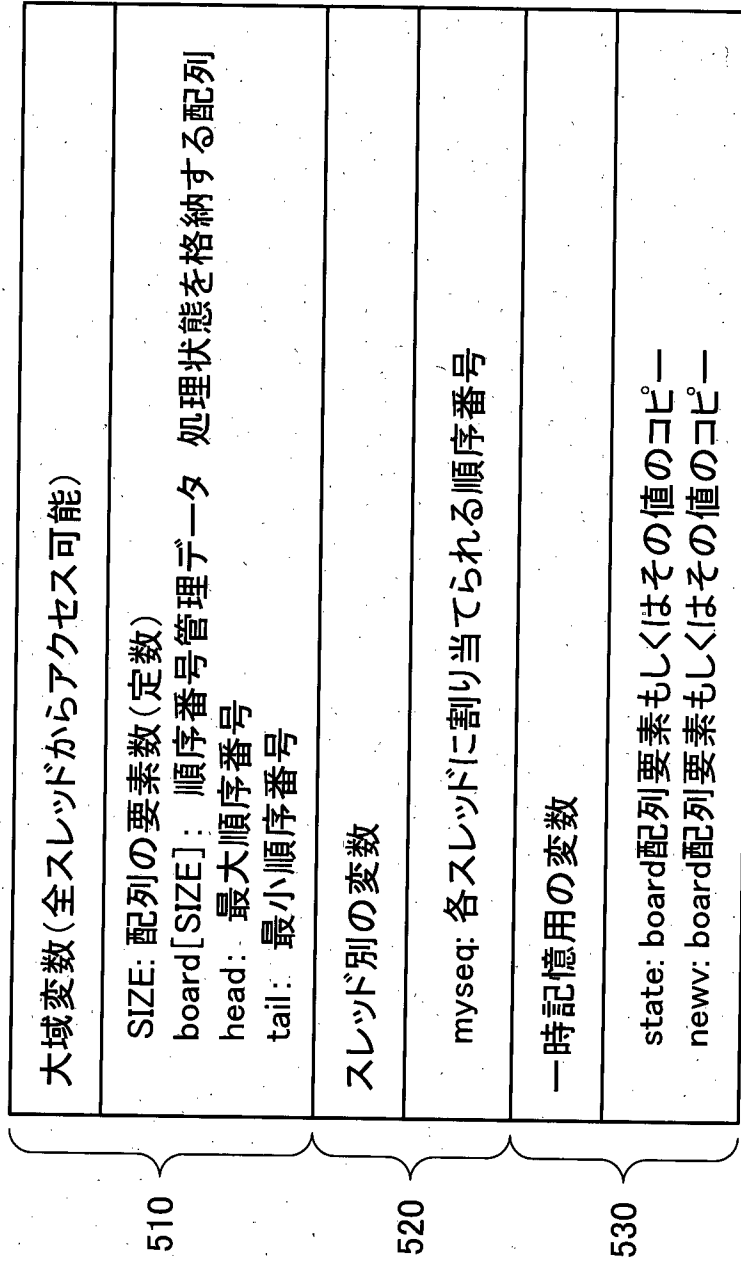


図6

600

601

602

操作表記	操作内容
atomic_inc(&v):	変数vに対し値1を不可分加算する 引数は変数vのアドレス、戻り値は加算後の値
atomic_cmpxchg(&v,o,n):	変数vに対してcmpxchg操作を不可分に実行する 引数は定数vのアドレス、書き換え前の値o、書き換え後の値n メモリから読み込んだvの値がoと一致した場合のみ、vにnを設定する 戻り値は、メモリから読み込んだvの値
OFFSET(v):	変数vに対する配列のインデックス(v%SIZE)を求める

610

620

630

図7B

730

731

732

733

データ表記	データ内容	データの意味
TAG(myseq), TAG(board[i]):	順序番号(myseq)およびboard配列要素(board[i], i=0~SIZE-1の値)から取得したTAG情報	
REUSE(state):	board配列要素もしくはその値のコピーを格納する変数(state)から取得したREUSEビット	1: 追い越し終了 0: その他
PASSED(state):	board配列要素もしくはその値のコピーを格納する変数(state)から取得したPASSEDビット	1: 追い越し有り 0: 追い越し無し

740

750

760

図8A

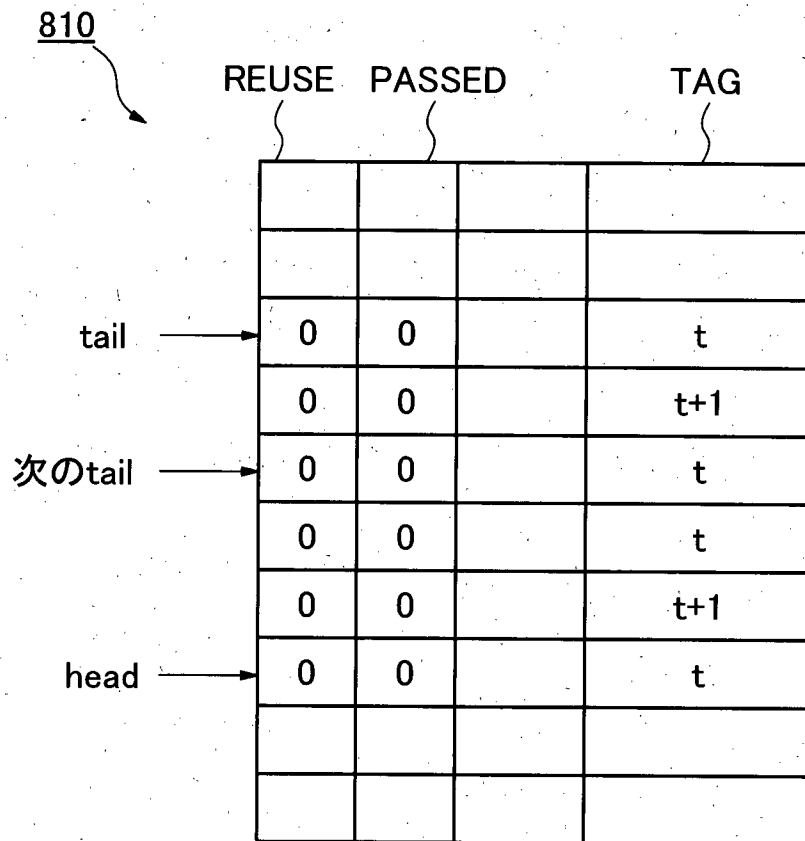


図8B

820

	REUSE	PASSED	TAG
	0	0	t
	0	0	t+1
tail	0	1	t-1
	1	1	-
	0	1	t-1
	0	0	t
head	0	0	t
	0	0	t-1
	0	0	t
	0	0	t

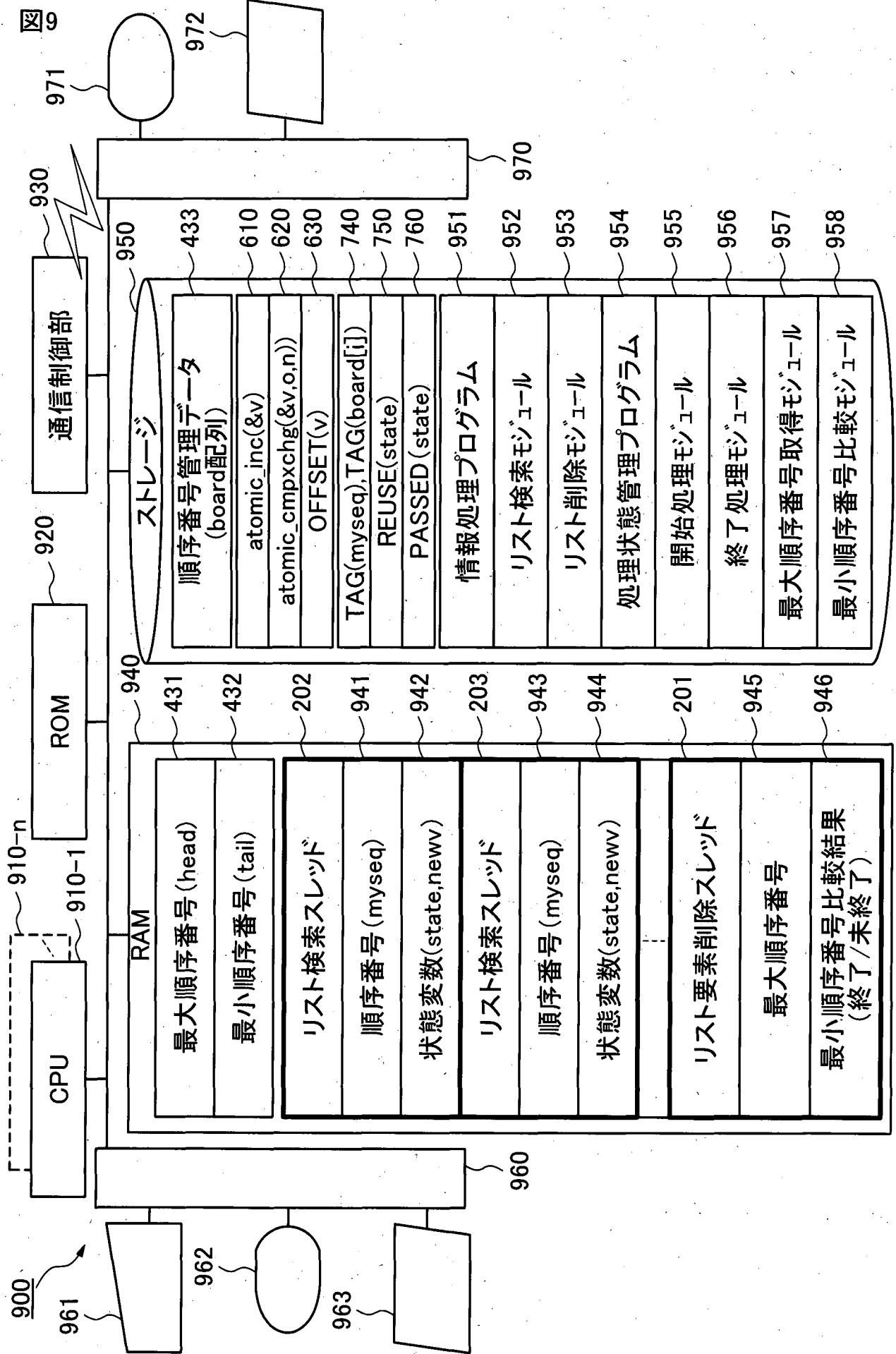


図9

図10A

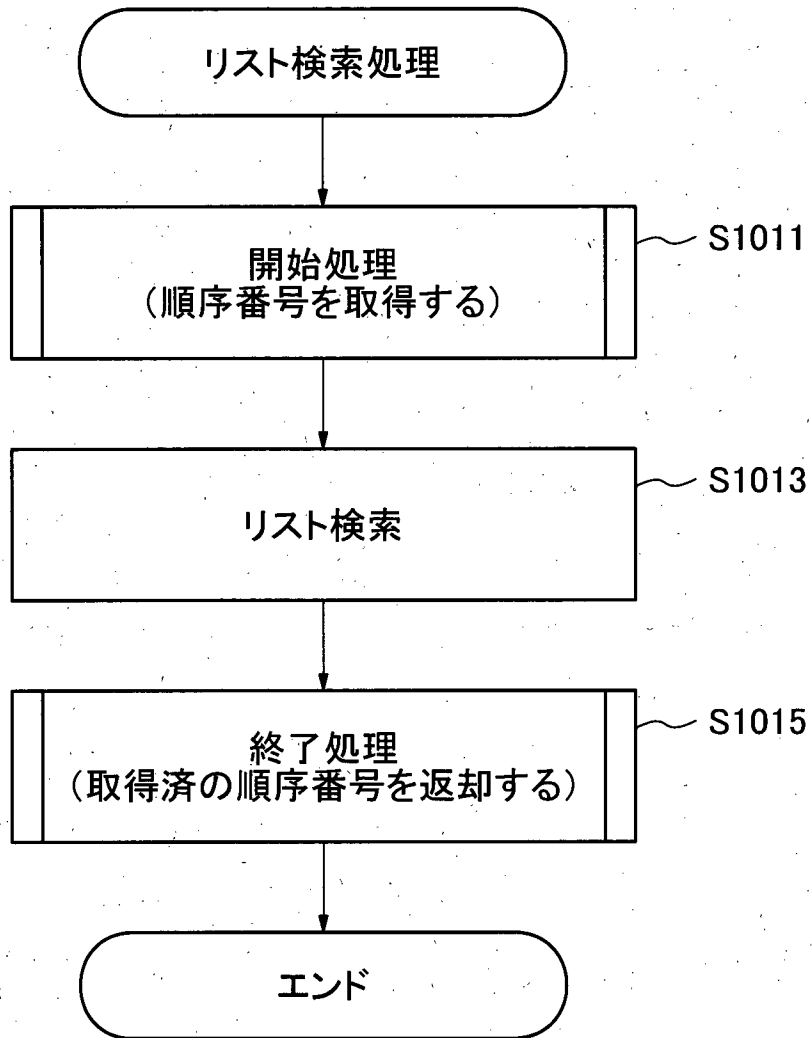


図10B

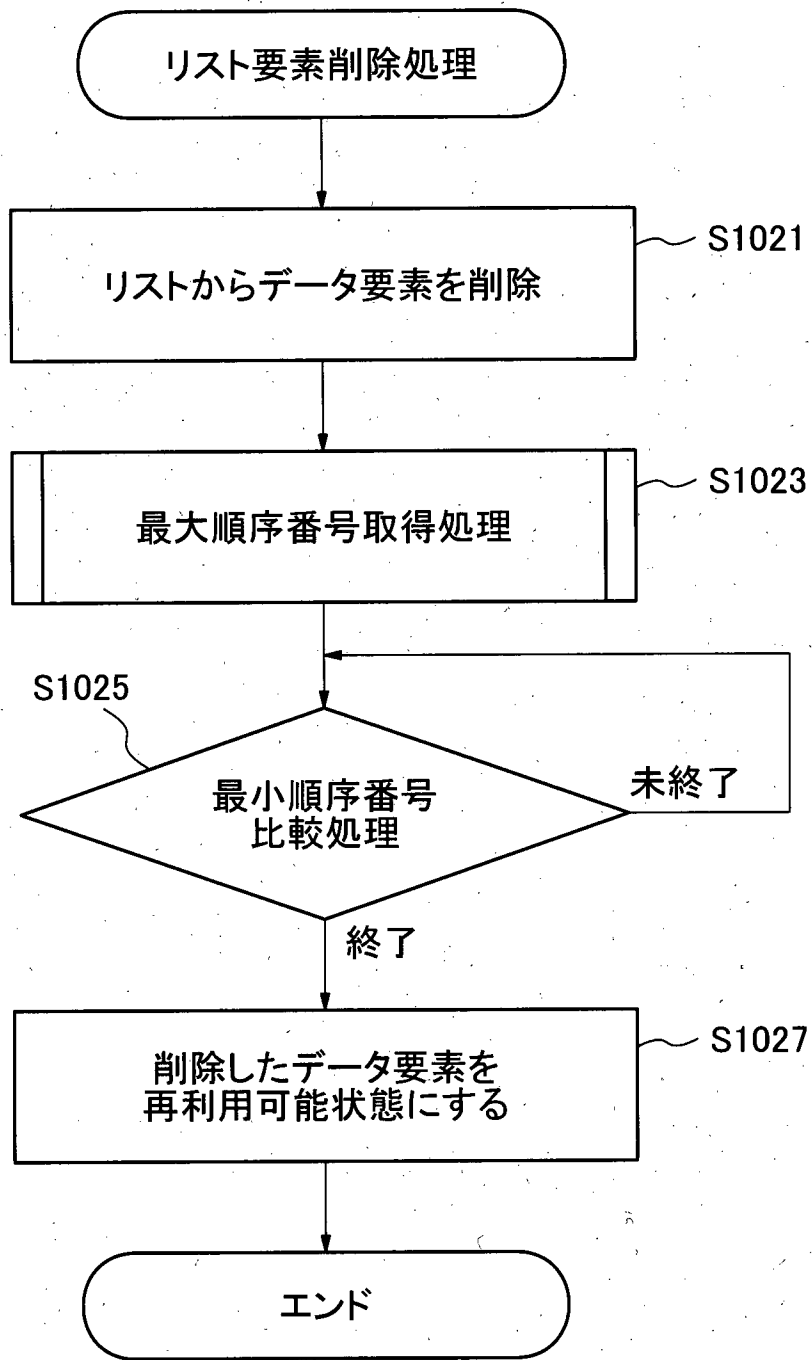
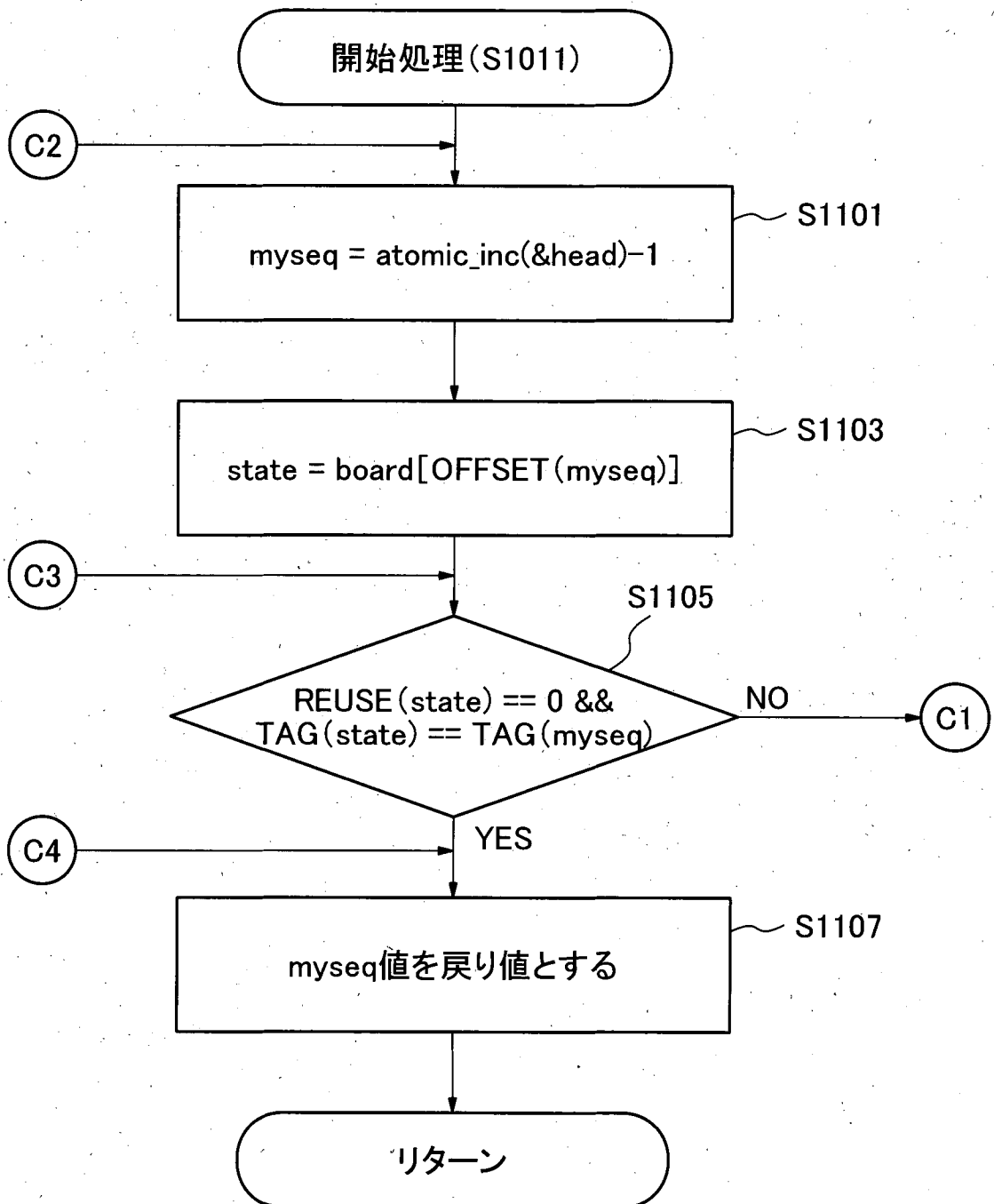


図11A



111B

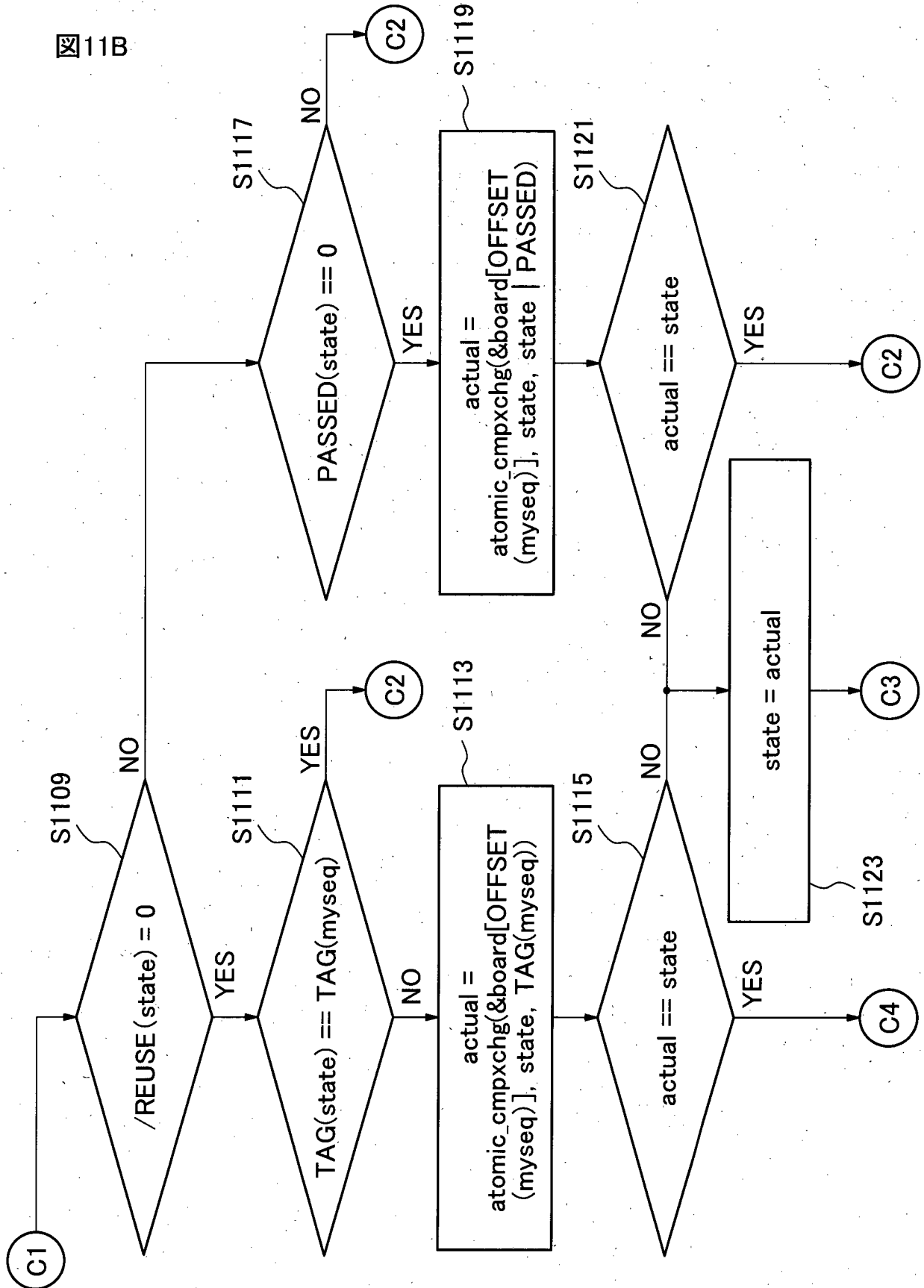


図12A

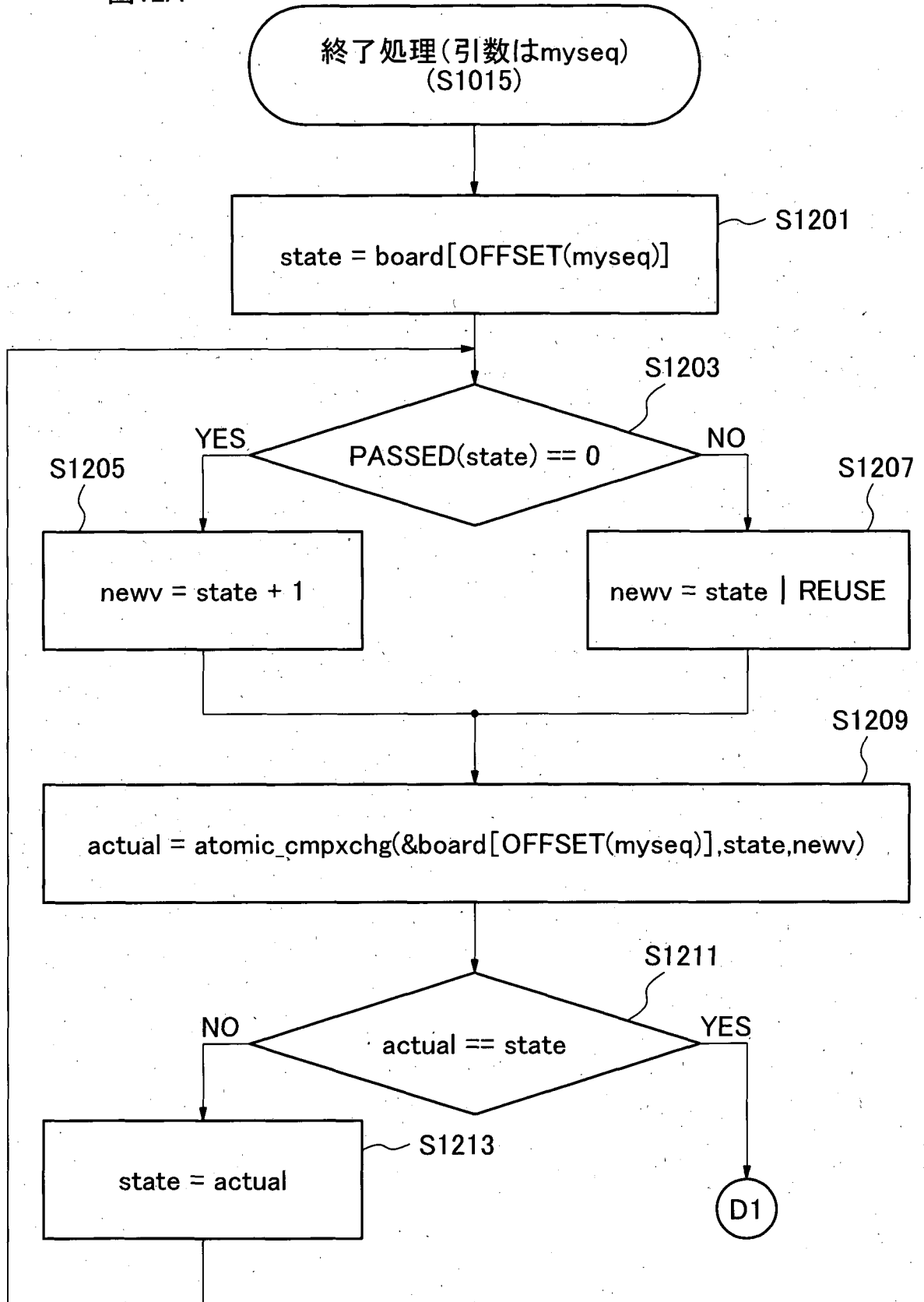
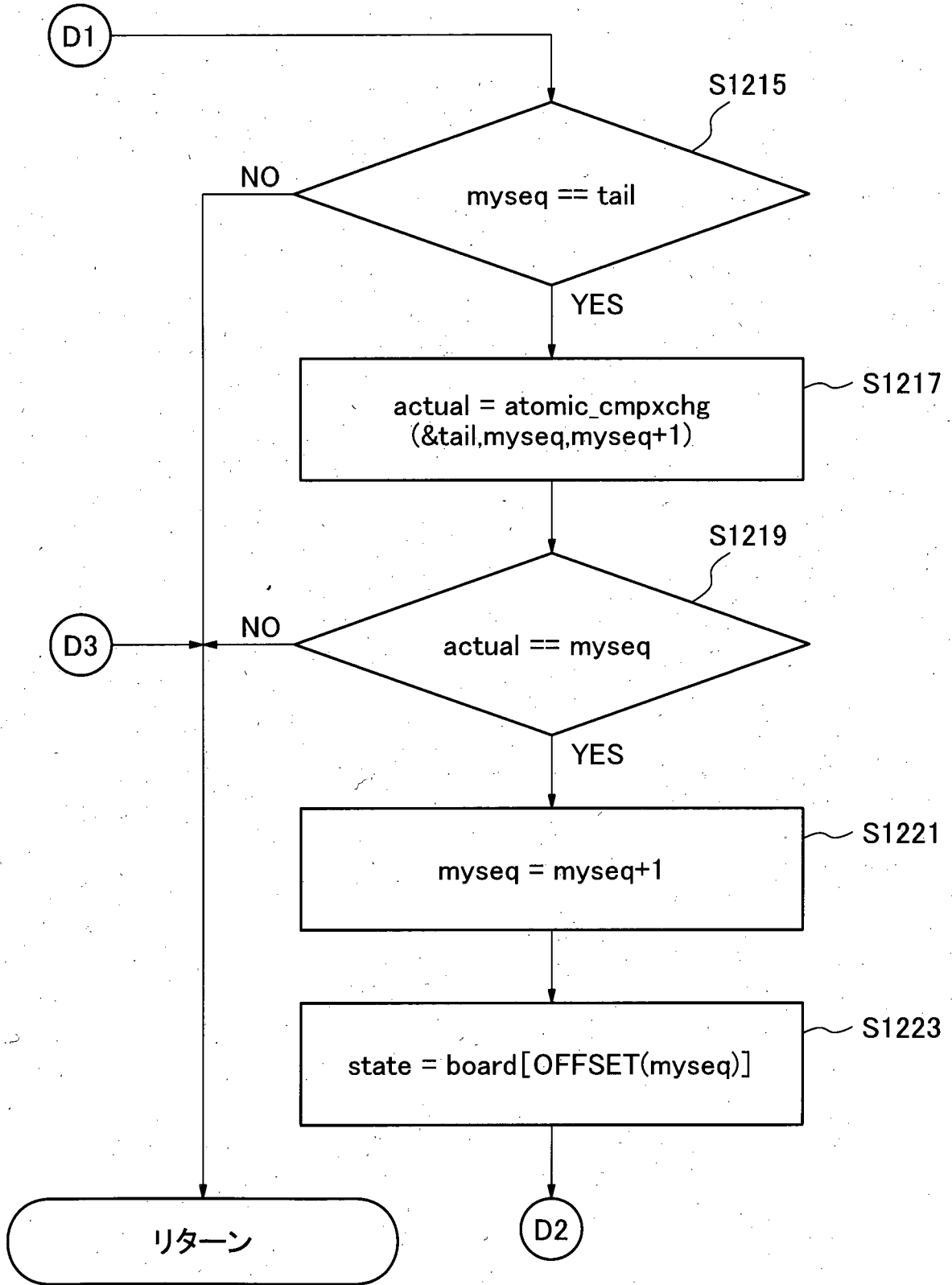


図12B



12C

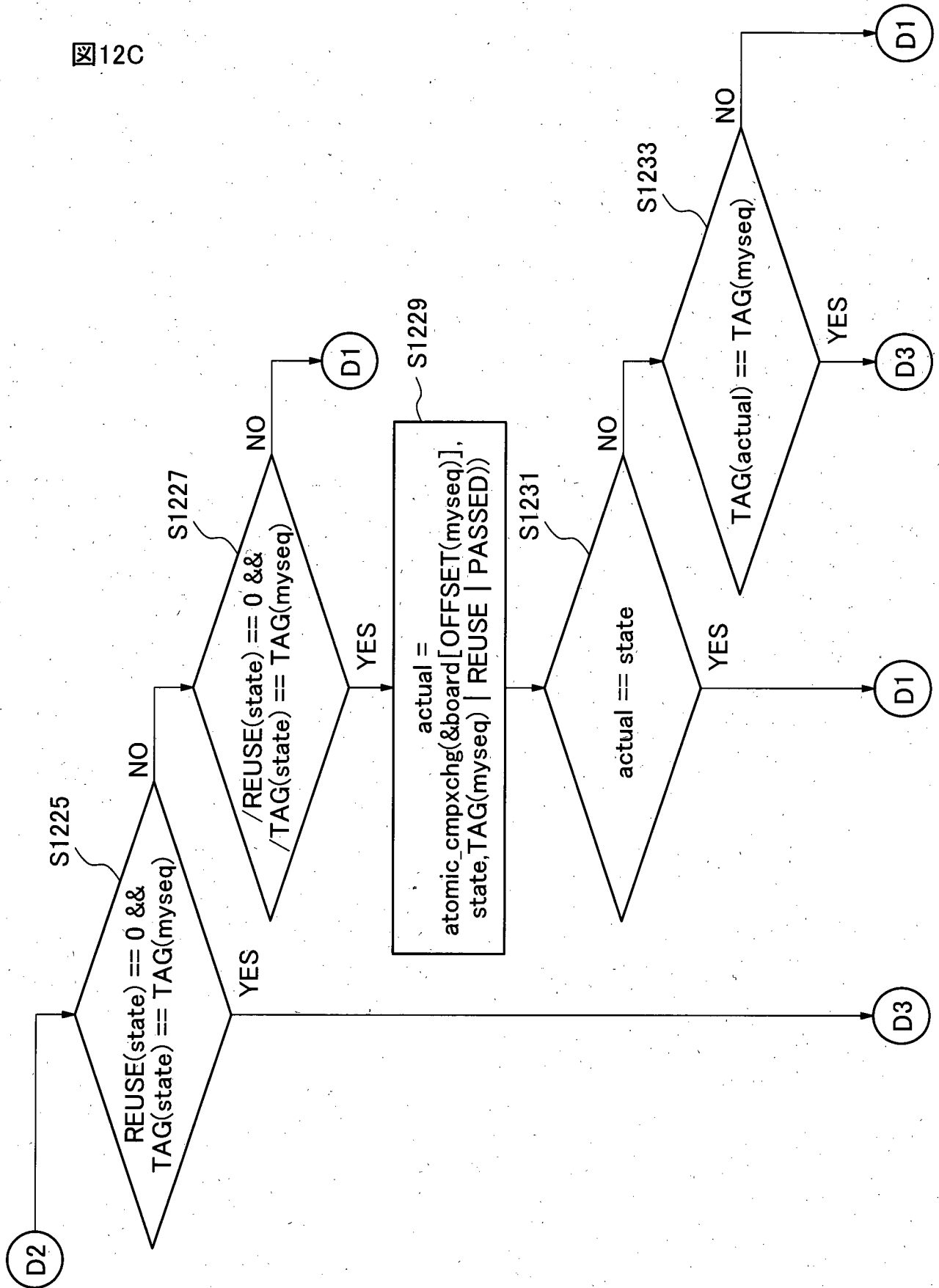


図13

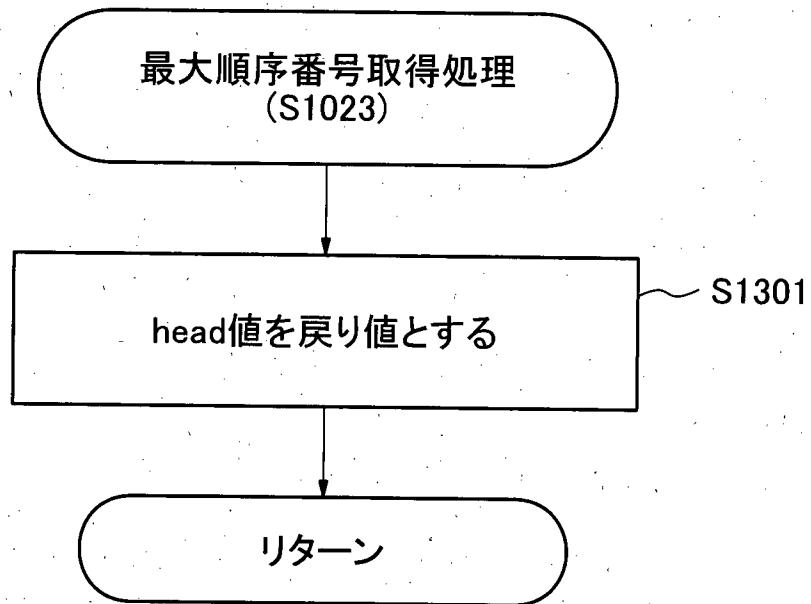


図14

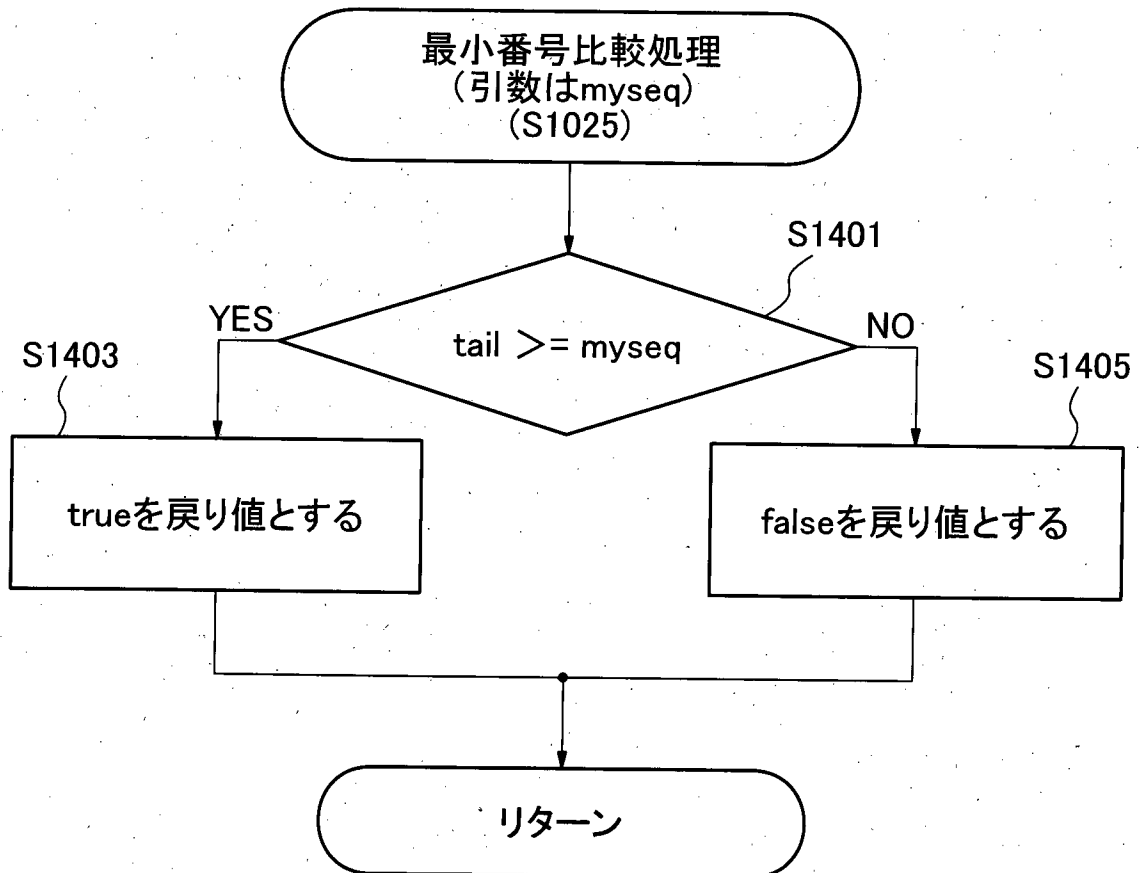
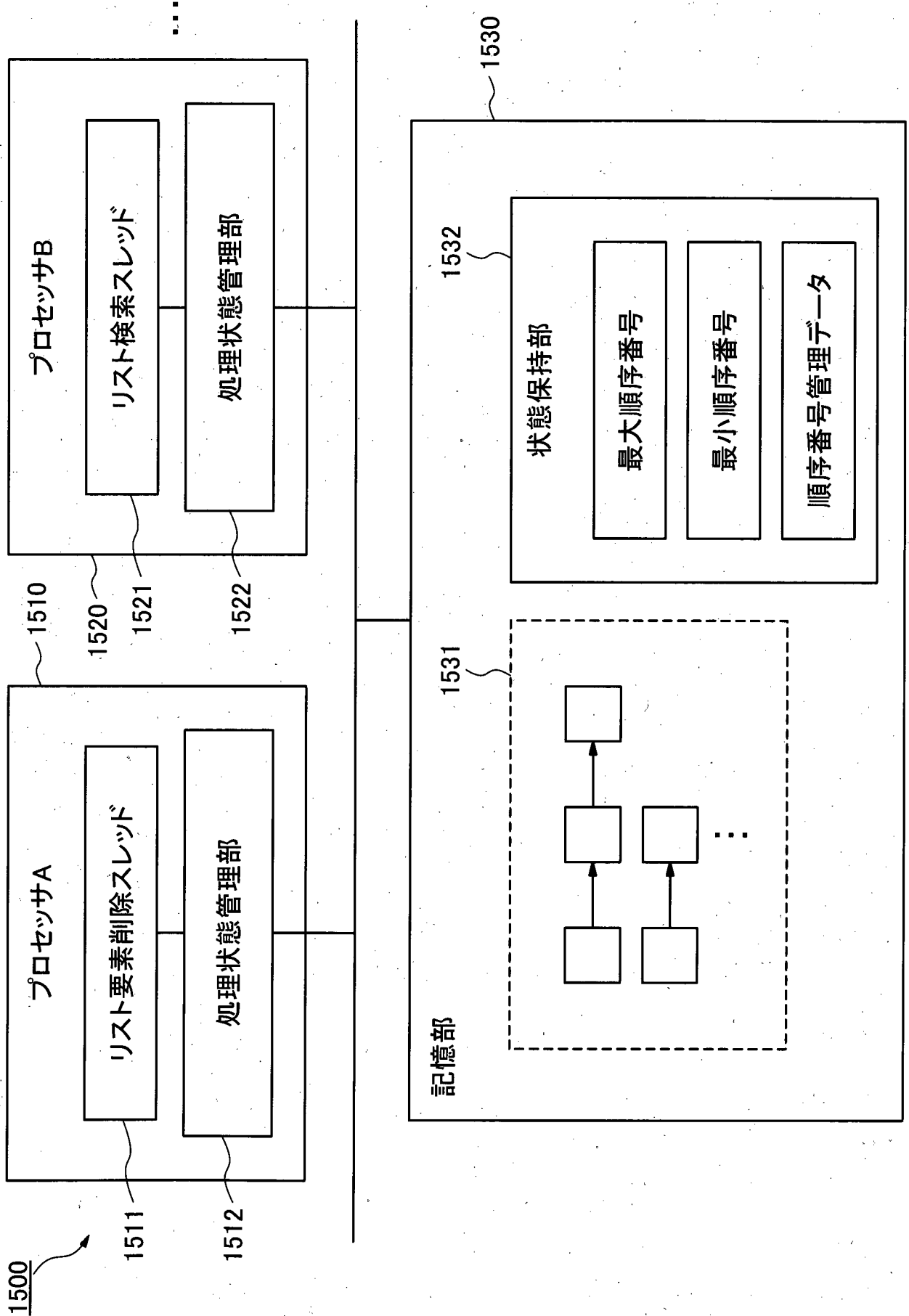


図15



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/068748

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/52(2006.01) i, G06F12/00(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)

G06F9/52, G06F12/00

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

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2012
Kokai Jitsuyo Shinan Koho	1971-2012	Toroku Jitsuyo Shinan Koho	1994-2012

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.
Y A	JP 2010-33554 A (International Business Machines Corp.), 12 February 2010 (12.02.2010), paragraphs [0002] to [0013]; fig. 5 to 7 & US 2010/0023559 A1 & KR 10-2010-0012805 A	1-3, 8, 10 4-7, 9
Y A	JP 2002-259146 A (Matsushita Electric Industrial Co., Ltd.), 13 September 2002 (13.09.2002), paragraphs [0010] to [0023] & US 2002/0049719 A1 & EP 1164485 A2 & CN 1325069 A	1-3, 8, 10 4-7, 9

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
13 August, 2012 (13.08.12)

Date of mailing of the international search report
21 August, 2012 (21.08.12)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))
 Int.Cl. G06F9/52(2006.01)i, G06F12/00(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. G06F9/52, G06F12/00

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 1971-2012年
 日本国実用新案登録公報 1996-2012年
 日本国登録実用新案公報 1994-2012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
Y A	JP 2010-33554 A (インターナショナル・ビジネス・マシーンズ・コーポレーション) 2010.02.12, 【0002】-【0013】、図5-7 & US 2010/0023559 A1 & KR 10-2010-0012805 A	1-3、8、 10 4-7、9
Y A	JP 2002-259146 A (松下電器産業株式会社) 2002.09.13, 【0010】-【0023】 & US 2002/0049719 A1 & EP 1164485 A2 & CN 1325069 A	1-3、8、 10 4-7、9

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
 13.08.2012

国際調査報告の発送日
 21.08.2012

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)
 田中 幸雄
 電話番号 03-3581-1101 内線 3545