

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4921453号  
(P4921453)

(45) 発行日 平成24年4月25日(2012.4.25)

(24) 登録日 平成24年2月10日(2012.2.10)

(51) Int.Cl.

F I

G 0 6 F 7/24 (2006.01)

G 0 6 F 7/24

A

請求項の数 10 (全 23 頁)

(21) 出願番号	特願2008-325692 (P2008-325692)	(73) 特許権者	506235616
(22) 出願日	平成20年12月22日(2008.12.22)		株式会社エスグランツ
(65) 公開番号	特開2010-146472 (P2010-146472A)		千葉県千葉市美浜区高洲三丁目5番3棟1
(43) 公開日	平成22年7月1日(2010.7.1)		210号
審査請求日	平成23年12月15日(2011.12.15)	(74) 代理人	100133570
早期審査対象出願			弁理士 ▲徳▼永 民雄
		(72) 発明者	新庄 敏男
			千葉県千葉市美浜区高洲三丁目5番3棟1
			210号 株式会社エスグランツ内
		(72) 発明者	園分 光裕
			千葉県千葉市美浜区高洲三丁目5番3棟1
			210号 株式会社エスグランツ内
		審査官	田中 友章
			最終頁に続く

(54) 【発明の名称】 ビット列データソート装置、方法及びプログラム

(57) 【特許請求の範囲】

【請求項1】

ビット列で表されるソート対象のキーのソート処理を行うビット列データソート装置において、

前記ソート対象のキーを記憶するソート対象キー記憶手段と、

前記ソート対象キー記憶手段に記憶された前記ソート処理を構成する分類処理の分類対象である前記キーと、該分類対象であるキーのうち基準となるキーと、のビット列比較により、最初に異なるビット値となる差分ビット位置を求める差分ビット位置計算手段と、

前記差分ビット位置計算手段で求めた差分ビット位置毎に該差分ビット位置を有するキーの識別情報を記憶する分類対象キー差分ビット位置記憶手段と、

前記差分ビット位置計算手段で求めた前記差分ビット位置毎に該差分ビット位置を有するキーの識別情報を前記分類対象キー差分ビット位置記憶手段に書き込むことにより、前記分類対象であるキーを同一の差分ビット位置を有する組に分類する差分ビット位置分類手段と、

前記差分ビット位置分類手段で分類された同一の差分ビット位置を有するキーが1つの場合はそのキーを、同一の差分ビット位置を有するキーが複数の場合はそのうちの次の分類処理において基準となるキーをソート済みキーとして出力するソート済みキー出力手段と、

前記差分ビット位置計算手段、前記差分ビット位置分類手段及び前記ソート済みキー出力手段を制御する制御手段と、

10

20

を備え、

前記制御手段は、

前記差分ビット位置計算手段が、前記差分ビット位置計算手段の最初の処理における前記分類対象をソート対象のキー全体とし、以後、前記差分ビット位置分類手段で分類された同一の差分ビット位置を有するキーを複数含むキーの組を前記分類対象として前記差分ビット位置をもとめることを繰り返し、

前記差分ビット位置分類手段が、前記差分ビット位置計算手段で求められた差分ビット位置毎に同一の差分ビット位置を有する組に分類することを繰り返し、

前記ソート済みキー出力手段は、前記差分ビット位置分類手段が前記差分ビット位置毎に同一の差分ビット位置を有する組に分類するたびに、該同一の差分ビット位置を有するキーが1つの場合のキー及び前記基準となるキーをソート済みキーとして出力するように、前記差分ビット位置計算手段、前記前記差分ビット位置分類手段及び前記ソート済みキー出力手段を制御し、

前記分類対象であるキーのうち基準となるキーを、分類対象であるキーのうち最小の値をとる最小値キーあるいは最大の値をとる最大値キーとすること  
を特徴とするビット列データソート装置。

【請求項2】

請求項1に記載のビット列データソート装置において、

前記キーを識別する情報は、該キーが記憶された前記ソート対象キー記憶手段における該キーのアドレスである読出位置であり、

前記分類対象キー差分ビット位置記憶手段は、前記分類対象であるキーと該分類対象であるキーのうち基準となるキーである前記最小値キーあるいは最大値キーとの差分ビット位置毎に、該最小値キーあるいは最大値キーに対する同一の差分ビット位置を有するキーのうち、最小あるいは最大のキーの前記読出位置である親リンクを格納する差分ビット位置表と、前記読出位置毎に該読出位置のキーと前記同一の差分ビット位置を有するキーの読出位置であるリンクを格納するリンク表から構成されることを特徴とするビット列データソート装置。

【請求項3】

請求項2に記載のビット列データソート装置において、

前記同一の差分ビット位置を有するキーの前記キー表の読出位置をリンク表の読出位置とするリンクは、該リンクをリンク表の読出位置とし、さらに該読出位置に設定されたリンクをたどることにより、前記同一の差分ビット位置を有する全てのキーの前記ソート対象キー記憶手段の読出位置を参照するために設定されており、該同一の差分ビット位置を有するキーのうちの最後のキーの読出位置のリンクには、該最後のキーの読出位置が格納されていることを特徴とするビット列データソート装置。

【請求項4】

請求項3に記載のビット列データソート装置において、

前記制御手段は、前記差分ビット位置分類手段が、前記差分ビット位置表の差分ビット位置の降順あるいは昇順に、該差分ビット位置に格納された前記親リンクを読出位置とするキーと同一の差分ビット位置を有する組を分類することを繰り返すように、制御することを特徴とするビット列データソート装置。

【請求項5】

ビット列で表されるソート対象のキーのソート処理を行うビット列データソート方法において、

前記ソート対象のキーを記憶するソート対象キー記憶手段に前記ソート対象のキーを記憶するソート対象キー記憶ステップと、

前記ソート対象キー記憶手段に記憶された前記ソート処理を構成する分類処理の分類対象である前記キーと、該分類対象であるキーのうち基準となるキーと、のビット列比較により、最初に異なるビット値となる差分ビット位置を求める差分ビット位置計算ステップと、

10

20

30

40

50

前記差分ビット位置計算ステップで求めた差分ビット位置毎に、分類対象キー差分ビット位置記憶手段に該差分ビット位置を有するキーの識別情報を書き込むことにより、前記分類対象であるキーを同一の差分ビット位置を有する組に分類する差分ビット位置分類ステップと、

前記差分ビット位置分類ステップで分類された同一の差分ビット位置を有するキーが1つの場合はそのキーを、同一の差分ビット位置を有するキーが複数の場合はそのうちの次の分類処理において基準となるキーをソート済みキーとして出力するソート済みキー出力ステップと、

前記差分ビット位置計算ステップ、前記差分ビット位置分類ステップ及び前記ソート済みキー出力ステップを制御する制御ステップと、

を備え、

前記制御ステップは、

最初の処理における前記分類対象をソート対象のキー全体として前記差分ビット位置計算ステップを実行し、以後、前記差分ビット位置分類ステップで分類された同一の差分ビット位置を有するキーを複数含むキーの組を前記分類対象として前記差分ビット位置計算ステップを繰り返し、

前記差分ビット位置計算ステップで求められた差分ビット位置毎に同一の差分ビット位置を有する組に分類する前記差分ビット位置分類ステップを繰り返し、

前記差分ビット位置分類ステップが前記差分ビット位置毎に同一の差分ビット位置を有する組に分類するたびに、前記ソート済みキー出力ステップが、該同一の差分ビット位置を有するキーが1つの場合のキー及び前記基準となるキーをソート済みキーとして出力するように、前記差分ビット位置計算ステップ、前記前記差分ビット位置分類ステップ及び前記ソート済みキー出力ステップを制御するものであり、

前記分類対象であるキーのうち基準となるキーを、分類対象であるキーのうち最小の値をとる最小値キーあるいは最大の値をとる最大値キーとする、

ことを特徴とするビット列データソート方法。

#### 【請求項6】

請求項5に記載のビット列データソート方法において、

前記キーを識別する情報は、該キーが記憶された前記ソート対象キー記憶手段における該キーのアドレスである読出位置であり、

前記分類対象キー差分ビット位置記憶手段は、前記分類対象であるキーと該分類対象であるキーのうち基準となるキーである前記最小値キーあるいは最大値キーとの差分ビット位置毎に、該最小値キーあるいは最大値キーに対する同一の差分ビット位置を有するキーのうち、最小あるいは最大のキーの前記読出位置である親リンクを格納する差分ビット位置表と、前記読出位置毎に該読出位置のキーと前記同一の差分ビット位置を有するキーの読出位置であるリンクを格納するリンク表から構成されることを特徴とするビット列データソート方法。

#### 【請求項7】

請求項6に記載のビット列データソート方法において、

前記同一の差分ビット位置を有するキーの前記キー表の読出位置をリンク表の読出位置とするリンクは、該リンクをリンク表の読出位置とし、さらに該読出位置に設定されたリンクをたどることにより、前記同一の差分ビット位置を有する全てのキーの前記ソート対象キー記憶手段の読出位置を参照するために設定されており、該同一の差分ビット位置を有するキーのうちの最後のキーの読出位置のリンクには、該最後のキーの読出位置が格納されていることを特徴とするビット列データソート方法。

#### 【請求項8】

請求項7に記載のビット列データソート方法において、

前記制御ステップは、前記差分ビット位置分類ステップが、前記差分ビット位置表の差分ビット位置の降順あるいは昇順に、該差分ビット位置に格納された前記親リンクを読出位置とするキーと同一の差分ビット位置を有する組を分類することを繰り返すように、制

10

20

30

40

50

御することを特徴とするビット列データソート方法。

【請求項 9】

請求項 5 ～ 請求項 8 のいずれか 1 項に記載のビット列データソート方法をコンピュータに実行させるプログラム。

【請求項 10】

請求項 5 ～ 請求項 8 のいずれか 1 項に記載のビット列データソート方法をコンピュータに実行させるプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、ビット列で表されるキーデータのソート装置、方法及びプログラムに関する。

【背景技術】

【0002】

近年、社会の情報化が進展し、大規模なデータベースが各所で利用されるようになってきている。このような大規模なデータベースからレコードを検索するには、各レコードの記憶されたアドレスと対応づけられたレコード内の項目をインデックスキーとして検索をし、所望のレコードを探し出すことが通例である。また、全文検索における文字列も、文書のインデックスキーと見なすことができる。

20

【0003】

そして、それらのインデックスキーはビット列で表現されることから、データベースの検索はビット列の検索に帰着されるということが出来る。

一方、データベースに関連した処理として、データベース中のレコードのインデックスキーによるソート処理が行われている。このソート処理もビット列のソート処理に帰着される。

【0004】

ソートの手法は各種のものが開発されており、下記特許文献 1 には、クイックソート、ラディックソート（基数ソート）等が紹介されている。また、特許文献 2 にも基数ソートが記載されている。

30

【0005】

図 1 A に示すのは、従来の基数ソートの概念を説明する図である。基数ソートによれば、図 1 A に例示するソート対象である 4 ビットのビット列であるキーは、0 ビット目から 3 ビット目に至る各ビット位置におけるビット値による分類を繰り返すことにより、ソートが実行される。以下、図 1 A の例示により、基数ソートの概念を説明する。

【0006】

図 1 A には、ソート対象であるキーからなるキー列 1 0 0 が示されている。図 1 A の例示では、キー列 1 0 0 に含まれるキーの存在するキーの位置であるキー位置 1 0 1 が 1 0 0 a（以下、キー位置 1 0 0 a のように表記する。）である記憶領域にはキー“1 1 1 1”が存在する。また、キー位置 1 0 0 b、1 0 0 c 1 0 0 d、1 0 0 e には、それぞれキー“0 0 1 1”、“1 0 1 0”、“0 0 0 1”、“1 1 1 0”が存在する。

40

【0007】

図に示すように、まずビット位置毎の分類（0 ビット目）1 1 0 a によりキー列 1 0 0 に含まれるキーの 0 ビット目による分類が行われる。その結果、キー“0 0 0 1”とキー“0 0 1 1”からなる 0 ビット目の値 0 の組 1 2 0 a と、キー“1 1 1 1”、キー“1 0 1 0”、キー“1 1 1 0”からなる値 1 の組 1 2 1 a が得られる。次に値 0 の組 1 2 0 a はビット位置毎の分類（1 ビット目）1 1 0 b により、次に値 1 の組 1 2 1 a はビット位置毎の分類（1 ビット目）1 1 0 e によりそれぞれ 1 ビット目の値による分類が行われる。

【0008】

50

ビット位置毎の分類（１ビット目）１１０ｂでは、キー“０００１”とキー“００１１”の１ビット目が共に０であることから、１ビット目の値０の組１２０ｂしか得られず、０ビット目の値０の組１２０ａと同じキーが２ビット目による分類の対象となり、ビット位置毎の分類（２ビット目）１１０ｃにより２ビット目による分類が行われる。ビット位置毎の分類（２ビット目）１１０ｃでは、２ビット目の値０の組１２０ｄとして１つのキー“０００１”が得られるので、最小値であるキーを格納する、ソート済みキー列１３０のキーを格納する位置であるキー位置１３１が１３０ａ（以下、キー位置１３０ａのように表記する。）である記憶領域に格納される。同様に、２ビット目の値１の組１２１ｄとして１つのキー“００１１”が得られるので、ソート済みキー列１３０の最小値の次の値のキーを格納するキー位置１３０ｂに格納される。なお、ソート済みキー列には、キー位置の符号順に小さいほうのキーから格納されるものとする。

10

#### 【０００９】

一方、ビット位置毎の分類（１ビット目）１１０ｅの分類では、１ビット目の値０の組１２０ｅとして１つのキー“１０１０”が得られるので、ソート済みキー列１３０の次の格納位置であるキー位置１３０ｃに格納される。また、１ビット目の値１の組１２１ｅとして、キー“１１１１”とキー“１１１０”の組が得られ、ビット位置毎の分類（２ビット目）１１０ｆの分類で、２ビット目の値に基づく分類が行われる。

#### 【００１０】

ビット位置毎の分類（２ビット目）１１０ｆでは、キー“１１１１”とキー“１１１０”の２ビット目が共に１であることから、１ビット目の値１の組１２１ｆしか得られず、１ビット目の値１の組１２１ｅと同じキーが３ビット目による分類の対象となり、ビット位置毎の分類（３ビット目）１１０ｇにより３ビット目の値による分類が行われる。ビット位置毎の分類（３ビット目）１１０ｇでは、３ビット目の値０の組１２０ｇとして１つのキー“１１１０”が得られるので、ソート済みキー列１３０の次の格納位置であるキー位置１３０ｄに格納される。同様に、３ビット目の値１の組１２１ｇとして１つのキー“１１１１”が得られるので、ソート済みキー列１３０の次の格納位置であるキー位置１３０ｅに格納される。

20

#### 【００１１】

以上の処理により、キー列１００のキーは、ソート済みキー列１３０のキー位置１３０ａ～１３０ｅにソートされて格納される。しかし、上述の基数ソート方法によりソートを実行する場合には、図１Ａに示すビット位置毎の分類（１ビット目）１１０ｂやビット位置毎の分類（２ビット目）１１０ｆにみられるように、分類が行われない無効な処理が発生する。

30

【特許文献１】特開２００２－１１６９０７号公報

【特許文献２】特開２００５－３１６６６３号公報

【発明の開示】

【発明が解決しようとする課題】

#### 【００１２】

そこで本発明の解決しようとする課題は、ビット列データのソート処理において、無効となる処理が発生しない、効率的なソート手法を提供することである。

40

【課題を解決するための手段】

#### 【００１３】

本発明のソート処理によれば、基準となるキーとソート対象であるビット列からなるキーのビット列比較を行い、最初に異なるビット値となるビット位置である差分ビット位置を求め、差分ビット位置によるソート対象のキーの分類を行い、同一の差分ビット位置に分類された複数のキーのうちの基準となるキーについての差分ビット位置による分類をさらに繰り返すことにより、ソート済みキー列を取得する。

【発明の効果】

#### 【００１４】

50

本発明によれば、差分ビット位置によりキーのソートを行うため、分類が行われない無効な処理は発生しない。したがって、効率的なソート処理を実現することができる。

【発明を実施するための最良の形態】

【0015】

以下、本発明を実施するための最良の形態を、図1Aに示したキー列100に含まれるキーと同一のキーをソート対象のキーとして例示して説明する。

【0016】

図1Bに示すのは、本発明の一実施の形態における差分ビット位置によるソート処理の概念を説明する図である。ソート対象のキー列100及びソート済みキー列130は、図1Aに示すものと同じである。

10

【0017】

キー列100を構成する各キーは、差分ビット位置による分類149aにおいて、最小値キー148a、最小値キーとの差分ビット位置が2であるキー群142a及び最小値キー148aとの差分ビット位置が0であるキー群140aに分類される。図の例では、最小値キー142aは“0001”である。キー群142aには、キー“0011”のみが含まれており、キー群140aにはキー“1010”、“1111”及び“1110”が含まれている。最小値キー148aは、キー列100からキーを差分ビット位置によるソート処理のための入力バッファに読み込むときに求めることができる。

【0018】

最小値キー148aは、図の点線の矢印158aに示すように、ソート済みキー列130のキー位置130aに格納される。また、キー群142aのキーは1つのみであるので、図の点線の矢印152aに示すように、ソート済みキー列130のキー位置130bに格納される。

20

【0019】

最小値キー148aとの差分ビット位置が0であるキー群140aにはキーが複数含まれているので、図の点線の矢印の組150aに示すように、差分ビット位置による分類149bにおいて、キー群140a内の最小値キー148bと、最小値キー148bとの差分ビット位置が1であるキー群141b、に分類される。図の例では、最小値キー148bは“1010”である。キー群141bには、キー“1111”とキー“1110”が含まれている。

30

【0020】

最小値キー148bは、図の点線の矢印158bに示すように、ソート済みキー列130のキー位置130cに格納される。

一方、最小値キー148bとの差分ビット位置が1であるキー群141bにはキーが複数含まれているので、図の点線の矢印の組151bに示すように、差分ビット位置による分類149cにおいて、キー群141b内の最小値キー148cと、最小値キー148cとの差分ビット位置が3であるキー群143c、に分類される。図の例では、最小値キー148cは“1110”である。キー群143cにはキー“1111”のみが含まれている。

40

【0021】

最小値キー148cは、図の点線の矢印158cに示すように、ソート済みキー列130のキー位置130dに格納される。また、キー群143cのキーは1つのみであるので、図の点線の矢印153cに示すように、ソート済みキー列130のキー位置130eに格納される。

【0022】

以上の処理により、キー列100のキーは、ソート済みキー列130のキー位置130a～130eに昇順で格納され、ソート処理が完了する。そして、本実施の態様によるソート処理は、差分ビット位置に基づいて分類を繰り返すので、分類処理毎に必ずキーの分類が実行される。なお、上述の例では、最小値キーを、差分ビット位置を計算する基準のキ

50

ーとしているが、最大値キーを、差分ビット位置を計算する基準のキーとすることも可能である。この場合には、図 1 B のソート対象キーの例では、最大値キーがキー “ 1 1 1 1 ” であり、最初の差分ビット位置による分類処理の結果として、最大値キー、差分ビット位置が 3 のキー “ 1 1 1 0 ”、差分ビット位置が 1 のキー “ 1 0 1 0 ” 及び差分ビット位置が 0 のキー “ 0 0 1 1 ” と “ 0 0 0 1 ” の組が得られる。

#### 【 0 0 2 3 】

次に、図 1 B を参照してその概念を説明した本実施の形態における差分ビット位置によるソート処理を実現するための機能ブロック構成例について説明する。本実施の形態における差分ビット位置によるソート処理は、図 1 B に例示した差分ビット位置による分類 10  
1 4 9 a、1 4 9 b、1 4 9 c 等の処理を実行する機能ブロックを、ソート対象のあらゆるキーの組み合わせに対して用意すれば実現可能であることは明らかである。しかしそれでは資源の無駄遣いであり、その実現手法は現実的でない。そこで本実施の形態においては、以下に説明する工夫を行っている。

#### 【 0 0 2 4 】

図 2 A は、本発明の一実施の形態における差分ビット位置によるソート処理で用いるデータ構造を説明するものである。図 2 A に示すように、キー表 3 0 9、差分ビット位置表 3 1 0、リンク表 3 1 1、ソート済みキー表 3 1 3 が用いられる。

#### 【 0 0 2 5 】

キー表 3 0 9 はソート要求があったときにソート対象のキー 6 0 2 が読み込まれて設定 20  
されるものである。図 2 A の例示では、ソート対象の 5 つのキー 6 0 2 は、図 1 A 及び図 1 B 示すキー列 1 0 0 に含まれるものが読み込まれている。キー表 3 0 9 の読出位置 6 0 1 の先頭は P 0 であり、以下、P 1、P 2、P 3、P 4 と続いている。読出位置 P 0 には、最小値キー “ 0 0 0 1 ” が設定されているが、別領域に設定しておくことも可能である。点線の矢印 6 8 9 c で示すように、最小値キーはソート済みキー表 3 1 3 の所定の位置に格納される。

#### 【 0 0 2 6 】

差分ビット位置表 3 1 0 は、最小値キーに対する同一の差分ビット位置を有するキーのうち、最小のキーの読出位置 6 0 1 を親リンク 6 1 2 として差分ビット位置 6 1 1 毎に格納するものである。図 2 A の例示では、点線の矢印 6 6 0 a で示すように、最小値キー “ 30  
0 0 0 1 ” との差分ビット位置が 0 であるキー “ 1 1 1 1 ”、“ 1 0 1 0 ”、“ 1 1 1 0 ” のうち最小のキー “ 1 0 1 0 ” の読出位置 P 3 が差分ビット位置表 3 1 0 の差分ビット位置 6 1 1 が 0 である位置に格納されている。また、点線の矢印 6 6 2 a で示すように、最小値キー “ 0 0 0 1 ” との差分ビット位置が 2 であるキーはキー “ 0 0 1 1 ” だけであるので、キー “ 0 0 1 1 ” の読出位置 P 2 が差分ビット位置表 3 1 0 の差分ビット位置 6 1 1 が 2 である位置に格納されている。

#### 【 0 0 2 7 】

リンク表 3 1 1 は、最小値キーに対する同一の差分ビット位置を有するキーにアクセスするためのものである。図に示すように、キーの読出位置 6 2 1 に対して、同一の差分 40  
ビット位置を有するキーの読出位置を示すリンク 6 2 2 が格納されている。

#### 【 0 0 2 8 】

差分ビット位置表 3 1 0 からリンク表 3 1 1 への点線の矢印 6 7 3 b で示すように、同一の差分ビット位置 0 を有するキーの最小値キーの読出位置 P 3 に対して、その読出位置 P 3 を読出位置 6 2 1 とするリンク表 3 1 1 の読出位置（以下、リンク表 3 1 1 の読出位置 P 3 のようにいう。）に、最小値キー “ 0 0 0 1 ” に対して差分ビット位置 0 を有するキーの最小値キー “ 1 0 1 0 ” と同一の差分ビット位置 0 を有するキー “ 1 1 1 0 ” のキー表 3 0 9 の読出位置 P 4 が格納されている。

#### 【 0 0 2 9 】

そして、図の点線の矢印 6 7 4 c で示すように、リンク表 3 1 1 の読出位置 P 4 に同一の 50

差分ビット位置 0 を有するキー “ 1 1 1 1 ” のキー表 3 0 9 の読出位置 P 1 が格納されている。また、図の点線の矢印 6 7 1 c で示すように、リンク表 3 1 1 の読出位置 P 1 には、同一の値のキー表 3 0 9 の読出位置 P 1 が格納されている。これにより、それ以上同一の差分ビット位置 0 を有するキーは存在しないことを表している。

【 0 0 3 0 】

また、差分ビット位置表 3 1 0 からリンク表 3 1 1 への点線の矢印 6 7 2 b で示すように、同一の差分ビット位置 2 を有するキーの最小値キーの読出位置 P 2 に対して、リンク表 3 1 1 の読出位置 P 2 には、リンク表 3 1 1 の読出位置 P 2 と同一の値のキー表 3 0 9 の読出位置 P 2 が格納されている。これは、キー表 3 0 9 の読出位置 P 2 のキー “ 0 0 1 1 ” が最小値キー “ 0 0 0 1 ” との差分ビット位置が 2 であるキーの最小のキーであるとともに、最後のものでもあることを示している。

10

【 0 0 3 1 】

上述の図 2 A に示す差分ビット位置表 3 1 0 とリンク表 3 1 1 の状態は、図 1 B に示す差分ビット位置による分類 1 4 9 a が実行されて得られるものである。

以上の説明から明らかなとおり、上述のキー表 3 0 9 は請求項 1 に係る発明のソート対象キー記憶手段の一実施例である。また、差分ビット位置表 3 1 0 とリンク表 3 1 1 で分類対象キー差分ビット位置記憶手段の一実施例を構成しており、キー表 3 0 9 の読出位置 6 0 1 はキーの識別情報に相当する。

【 0 0 3 2 】

図 2 B は、本発明の一実施の形態におけるビット列データソート装置の機能ブロック構成を説明する図である。

20

図に示すように、ビット列データソート装置 2 0 0 は、差分ビット位置計算手段 2 1 0、差分ビット位置分類手段 2 2 0、ソート済みキー出力手段 2 3 0、及び制御手段 2 4 0 を含む。以下、各手段の動作の概要について、図 2 A の例示を参照して説明する。

【 0 0 3 3 】

差分ビット位置計算手段 2 1 0 は、キー表 3 0 9 に記憶されたキー 6 0 2 のうちから、差分ビット位置による分類処理の対象となるキーの差分ビット位置を計算する。制御手段 2 4 0 は最初の分類処理の対象をキー表 3 0 9 に記憶された全てのキーとする。この場合、差分ビット位置を計算する基準となるキーは、読出位置 P 0 に格納された最小値キー “ 0 0 0 1 ” である。ソート済みキー出力手段 2 3 0 は、最小値キー “ 0 0 0 1 ” をソート済みキー表 3 1 3 に出力する。

30

【 0 0 3 4 】

差分ビット位置分類手段 2 2 0 は、差分ビット位置計算手段 2 1 0 の計算結果に基づいて、差分ビット位置表 3 1 0 とリンク表 3 1 1 の書き込みを行う。その結果、キーの識別情報をキー表 3 0 9 の読出位置 6 0 1 として、差分ビット位置が 0 の組 ( P 3、P 4、P 1 ) と差分ビット位置が 2 の組 ( P 2 ) に分類される。

【 0 0 3 5 】

制御手段 2 4 0 は、分類後に複数のキーが含まれる組について、さらに差分ビット位置計算手段 2 1 0 による差分ビット位置の計算と差分ビット位置分類手段 2 2 0 による分類処理が繰り返されるように制御する。

40

【 0 0 3 6 】

図 3 は、本発明を実施するためのハードウェア構成例を説明する図である。

本発明によるソート処理は中央処理装置 3 0 2 及びキャッシュメモリ 3 0 3 を少なくとも備えたデータ処理装置 3 0 1 によりデータ格納装置 3 0 8 を用いて実施される。キー表 3 0 9、差分ビット位置表 3 1 0、リンク表 3 1 1、ソート済みキー表 3 1 3 を有するデータ格納装置 3 0 8 は、主記憶装置 3 0 5 または外部記憶装置 3 0 6 で実現することができ、あるいは通信装置 3 0 7 を介して接続された遠方に配置された装置を用いることも可能である。

【 0 0 3 7 】

図 3 の例示では、主記憶装置 3 0 5、外部記憶装置 3 0 6 及び通信装置 3 0 7 が一本の

50



バス 304 によりデータ処理装置 301 に接続されているが、接続方法はこれに限るものではない。また、主記憶装置 305 をデータ処理装置 301 内のものとすることもできるし、キー表 309 は外部記憶装置 306 に、他の表は主記憶装置 305 に持つなど、使用可能なハードウェア環境、ソート対象キーの集合の大きさ等に応じて適宜ハードウェア構成を選択できることは明らかである。

【0038】

また、特に図示されてはいないが、処理の途中で得られた各種の値を後の処理で用いるためにそれぞれの処理に応じた主記憶装置 305 の一時記憶領域が用いられることは当然である。そして、以下の説明においては、一次記憶領域に格納されるあるいは設定される値を一時記憶領域の名前で呼ぶことがある。

10

【0039】

次に、本発明の一実施の形態におけるソート処理について、図 4A 及び図 4B を参照して詳細に説明する。また、その説明において、図 1B あるいは図 2A の例示を適宜用いて説明する。

【0040】

図 4A は、本発明の一実施の形態におけるソート処理の初段の分類処理とソート済みキーの出力処理の処理フローを説明する図である。キー表にはソート対象のキーが記憶されており、キー表の先頭の読出位置には最小値キーが記憶されていることを前提とする。すなわち、図 1B に示すキー列 100 に含まれるキーは図 2A に示すようにキー表 309 に記憶されているものとする。

20

【0041】

まず、ステップ S401 において、キー表の先頭位置をキー表の読出位置（以下、単に読出位置ということがある。）に設定する。ここでいう「キー表の読出位置」は、先に述べた「処理の途中で得られた各種の値を後の処理で用いるためにそれぞれの処理に応じた一時記憶領域」の 1 つである。以下の説明では、「キー表の先頭位置をキー表の読出位置に設定する。」のように、「キー表の読出位置」が図示しない一次記憶領域であることを省略して述べることもある。

上記の設定により、図 2A の例示では、キー表の読出位置に P0 が設定される。

【0042】

次にステップ S402 において、キー表より、先頭位置の指すキーを最小値キーとして取り出す。ここでいう「最小値キーとして取り出す」は、図示しない一次記憶領域である最小値キーに設定する、の意味である。以下においても、同様な表記を用いる場合がある。

30

【0043】

ステップ S403 に進み、読出位置に次の読出位置を設定する。次にステップ S404 において、全てのキーは処理済みか、すなわち、全てのキーについての初段における分類処理が終了したか判定する。

【0044】

全てのキーが処理済みでなければ、ステップ S405 に進み、キー表より、読出位置の指すキーをソートキーとして取り出すとともに、読出位置をソートキーの読出位置に設定する。そしてステップ S406 において、ソートキーと最小値キーの差分ビット位置を得て、差分ビット位置によりソートキーを分類し、ステップ S403 に戻る。

40

ステップ S406 の処理においては、初段における分類処理として差分ビット位置表とリンク表への設定が行われる。その処理の詳細については、後に図 5A 及び図 5B を参照して説明する。

【0045】

一方、ステップ S404 で全てのキーが処理済みであると判定されると、ステップ S407 に分岐し、ステップ S402 で得た最小値キーを、ソート済みキー表に設定し、図 4B に示すステップ S408 以下の処理に進む。以上の処理により、図 2A の例示のように、差分ビット位置表 310 とリンク表 311 の各エントリの値が設定される。また、最小値キー“0001”がソート済みキー表 313 に格納される。

50

## 【 0 0 4 6 】

図 4 B は、本発明の一実施の形態におけるソート処理の初段以降の分類処理とソート済みキーの出力処理の処理フローを説明する図である。

## 【 0 0 4 7 】

ステップ S 4 0 8 においては、図示しない一次記憶領域である差分ビット位置に、初期値として差分ビット位置表の差分ビット位置、すなわちソート対象のキーが有する可能性のある差分ビット位置の最大値を設定する。図 2 A の例示では、差分ビット位置の最大値として 3 が設定される。

## 【 0 0 4 8 】

ステップ S 4 0 9 において、差分ビット位置表の全ての差分ビット位置のエントリについて処理済みであるか判定し、処理済みであればソート処理を終了し、処理済みでなければステップ S 4 1 0 に進む。

10

## 【 0 0 4 9 】

ステップ S 4 1 0 では、差分ビット位置表より、差分ビット位置の指すエントリから親リンクを読み出す。差分ビット位置の指すエントリに親リンクが格納されていれば、この読み出された親リンクは一次記憶領域である親リンクに設定される。すなわち、本実施の態様の説明における「親リンクを読み出す」等の表記は、親リンクを読み出して図示しない一次記憶領域である親リンクに設定することを意味することがある。親リンク以外のものに対しても同様である。

ステップ S 4 1 1 では、親リンクは設定済みであるか判定する。設定済みでなければステップ S 4 1 6 で差分ビット位置から値を 1 減らしてステップ S 4 0 9 に戻り、一方、設定済みであればステップ S 4 1 2 に進む。

20

## 【 0 0 5 0 】

ステップ S 4 1 2 では、キー表より、ステップ S 4 1 0 で設定した親リンク（キー表の読出位置）の指すキーをソート済みキーとして取り出し、その取り出したソート済みキーをソート済みキー表に設定する。先に説明したように、差分ビット位置表にその読出位置が格納されているキーは、同一の差分ビット位置に分類されたキーのうちの最小値キーであるので、ステップ S 4 1 2 でソート済みキー表に設定する。

## 【 0 0 5 1 】

次にステップ S 4 1 3 で、リンク表より、ステップ S 4 1 0 で設定した親リンクの指すリンクを次リンクとして読み出し、ステップ S 4 1 4 で、差分ビット位置表の、差分ビット位置の指す親リンクを削除してステップ S 4 1 5 に進む。ここで差分ビット位置表の差分ビット位置の指す親リンクを削除するのは、キー表の読出位置が親リンクである最小値キーの分類処理はステップ S 4 1 2 で完了しており、もはや差分ビット位置表上のこの親リンクは必要がないことと、この親リンクが差分ビット位置表に残っていることによる後続の処理への影響を避けるためである。

30

## 【 0 0 5 2 】

ステップ S 4 1 5 では、ステップ S 4 1 3 で読み出した次リンクとステップ S 4 1 0 で設定した親リンクが同一であるか判定する。

ステップ S 4 1 5 で次リンクと親リンクが同一であると判定されると、先に説明したステップ S 4 1 6 を経由してステップ S 4 0 9 に戻る。

40

上述のステップ S 4 0 9 ~ ステップ S 4 1 6 のループ処理により、差分ビット位置の降順でソート済みキーの取り出しとソート済みキー表への格納が行われる。より大きい差分ビット位置を有するキーの値がより最小値キーの値に近いので、差分ビット位置の降順でソート済みキーを取り出すことにより、昇順でソート済みキーをソート済みキー表へ格納することができる。

## 【 0 0 5 3 】

ステップ S 4 1 5 で次リンクと親リンクが同一であると判定されるのは、親リンクが格納された差分ビット位置を有するキーが親リンクをキー表の読出位置とするキーしか存在しない場合である。図 2 A の例示では、リンク表 3 1 1 の読出位置 6 2 1 が P 2 でリンク

50

6 2 2 が P 2 の場合である。また、図 1 B の例示では、キー群 1 4 2 a がキー “ 0 0 1 1 ” のみで構成され、キー群 1 4 2 a についての分類処理は行われない場合に相当する。そして、この場合は差分ビット位置表に親リンクが格納された差分ビット位置についての分類処理が終了したことを意味するので、差分ビット位置から値 1 を減らし、次の差分ビット位置についてのステップ S 4 0 9 ~ S 4 1 5 の処理に進む。図 1 B の例示では、差分ビット位置 0 のキー群 1 4 0 a についての処理が行われる。

【 0 0 5 4 】

一方、ステップ S 4 1 5 で次リンクと親リンクが同一でないと判定されると、ステップ S 4 1 7 に進み、同一の差分ビット位置（ステップ S 4 0 8 またはステップ S 4 1 6 で設定されている。）を有する複数のキーを、下位の差分ビット位置によりさらに分類する。ステップ S 4 1 7 の処理の詳細については、後に図 5 C を参照して説明する。

10

【 0 0 5 5 】

次リンクと親リンクが同一でないと判定される場合は、図 2 A の例示では、リンク表 3 1 1 の読出位置 6 2 1 が P 3 でリンク 6 2 2 が P 4 の場合である。図 1 B の例示では、キー群 1 4 0 a の複数のキーについて、差分ビット位置による分類 1 4 9 b、1 4 9 c が実行される。

ステップ S 4 1 7 の処理では、差分ビット位置表とリンク表が再設定され、ステップ S 4 0 8 に戻る。そして、再設定された差分ビット位置表の親リンクの設定された差分ビット位置についての処理が繰り返される。

【 0 0 5 6 】

20

上述の処理を、ステップ S 4 0 9 において、差分ビット位置表の全ての差分ビット位置のエントリについて処理済みである、と判定されるまで繰り返し、その判定が得られるとソート対象のキーのソートが完了しているので処理を終了する。

図 4 B に例示する処理は、差分ビット位置を計算する基準となるキーを最小のキーとし、差分ビット位置の降順で処理を行ってソート済みキーを昇順で取り出すものである。しかし、先に述べたように、差分ビット位置を計算する基準となるキーを最大のキーとする場合には、基準キーに対して同一の差分ビット位置を有するキーのうち最大のキーの読出位置を親リンクとし、差分ビット位置の降順で処理を行ってソート済みキーを降順で取り出すことができることは、本明細書及び図面の記載によって当業者に明らかである。

30

【 0 0 5 7 】

次に図 5 A 及び図 5 B を参照して本発明の一実施の形態におけるキーを分類する処理を説明する。図 5 A 及び図 5 B に記載するものは、図 4 A に示すステップ S 4 0 6 の処理、及び後述の図 5 C に示すステップ S 5 2 4 の処理の詳細な処理フローである。

図 5 A は、本発明の一実施の形態におけるキーを分類する処理の前段の処理フローを説明する図である。

【 0 0 5 8 】

図に示すように、ステップ S 5 0 1 で、ソートキーに設定されているキーと最小値キーに設定されているキーをビット列として比較し、上位 0 ビット目からみた最初の不一致ビットのビット位置を、差分ビット位置に設定する。ソートキーは、図 4 A に示すステップ S 4 0 5、あるいは後述の図 5 C に示すステップ S 5 2 2 で設定されるものであり、図 5 A 及び図 5 B に示す分類処理の対象のキーである。

40

【 0 0 5 9 】

次にステップ S 5 0 2 で差分ビット位置表より、ステップ S 5 0 1 で設定した差分ビット位置の指す親リンクを読み出し、ステップ S 5 0 3 で親リンクは設定済みであるか判定する。親リンクが設定済みでなければ、ステップ S 5 0 4 に進み、差分ビット位置表の、差分ビット位置の指す親リンクにソートキーの読出位置を設定し、ステップ S 5 0 5 において、リンク表の、ソートキーの読出位置の指すリンクに、ソートキーの読出位置を設定して処理を終了する。

【 0 0 6 0 】

50

差分ビット位置表には、先に述べたように同一の差分ビット位置を有するキーのうち、最小のキーの読出位置が親リンクとして格納されるが、上記ステップ S 5 0 4 の処理は、最初のソートキーを仮の最小値キーとして、ソートキーの読出位置を差分ビット位置表の差分ビット位置の指す親リンクに設定するものである。一旦親リンクが設定済みとなった後、すなわちステップ S 5 0 3 での判定が親リンクは設定済みになると、図 5 B に示す後段の処理に進み、ソートキーと最小値キーの大小比較による最小値キーの更新が行われ、最終的には同一の差分ビット位置を有するキーのうち、最小のキーの読出位置が親リンクとして差分ビット位置表に格納される。

【 0 0 6 1 】

なお、ステップ S 5 0 4 の処理におけるソートキーの読出位置は、図 4 A に示すステップ S 4 0 5 あるいは後述の図 5 C に示すステップ S 5 2 3 で設定されるものである。

10

【 0 0 6 2 】

図 5 B は、本発明の一実施の形態におけるキーを分類する処理の後段の処理フローを説明する図である。

【 0 0 6 3 】

図に示すように、ステップ S 5 0 6 で、キー表より、親リンクの指すキーを、親ソートキーとして読み出し、ステップ S 5 0 7 において、ソートキーは親ソートキーより小さいか判定する。

【 0 0 6 4 】

ステップ S 5 0 7 でソートキーは親ソートキーより小さいと判定されると、ステップ S 5 0 8 に進み、差分ビット位置表の、差分ビット位置の指す親リンクにソートキーの読出位置を設定し、ステップ S 5 0 9 でリンク表の、ソートキーの読出位置の指すリンクに、ステップ S 5 0 2 で得た親リンクを設定して処理を終了する。

20

【 0 0 6 5 】

上述のステップ S 5 0 8 の処理は、差分ビット位置表の差分ビット位置の指す親リンクに同一の差分ビット位置を有するキーのうちで最小のキーの読出位置を設定するために、より小さいキーの読出位置で親リンクを更新するものである。そして、ステップ S 5 0 9 の処理は、差分ビット位置表に新たに設定されたソートキーの読出位置の指すリンク表のリンクに、差分ビット位置表に設定されていた親リンクを設定することで、リンク表に、同一の差分ビット位置を有するキーのキー表における読出位置をたどるための情報を保持させるものである。

30

【 0 0 6 6 】

一方、ステップ S 5 0 7 でソートキーは親ソートキーより小さくないと判定されると、ステップ S 5 1 0 に進み、リンク表より、親リンクの指すリンクを次リンクとして読み出し、ステップ S 5 1 1 で、次リンクと親リンクは同一か判定する。

【 0 0 6 7 】

ステップ S 5 1 1 で次リンクと親リンクは同一であると判定されると、ステップ S 5 1 2 において、リンク表の親リンクの指すリンクに、ソートキーの読出位置を設定する。そして、ステップ S 5 1 3 において、リンク表のソートキーの読出位置の指すリンクに、ソートキーの読出位置を設定して処理を終了する。

40

【 0 0 6 8 】

一方、ステップ S 5 1 1 で次リンクと親リンクは同一でないと判定されると、ステップ S 5 1 4 に進み、リンク表の親リンクの指すリンクに、ソートキーの読出位置を設定する。そして、ステップ S 5 1 5 において、リンク表のソートキーの読出位置の指すリンクに、ステップ S 5 1 0 で得た次リンクを設定して処理を終了する。

【 0 0 6 9 】

上述のステップ S 5 1 1 の判定は、同一の差分ビット位置を有するキーのうちで分類済みのキーが 1 つのみであったかの判定と同じである。分類済みのキーが 1 つのみであればこのキーの読出位置の指すリンク表のリンクにはその読出位置が格納されている。そこで、それをソートキーの読出位置で更新し、ソートキーの読出位置の指すリンク表のリンク

50

にソートキーの読出位置を設定してソートキーが、その時点での最後の分類済みのキーであることを示す。分類済みのキーが2つ以上のときは、ステップS514とステップS515の処理により、リンク表で表現される親リンクと次リンクのリンク関係の間に、ソートキーの読出位置を挿入する。

【0070】

次に図5Cを参照して、図4Bに示すステップS417の処理である、本発明の一実施の形態における同一の差分ビット位置を有する複数のキーをさらに分類する処理の詳細な処理フローを説明する。

図5Cは、本発明の一実施の形態における同一の差分ビット位置を有する複数のキーをさらに分類する処理の詳細な処理フローを説明する図である。

10

【0071】

図に示すように、ステップS521において、最小値キーにソート済みキーを設定する。ソート済みキーは、図4Bに示すステップS412で得たものであり、後記ステップS524においてソートキーの差分ビット位置を計算するための基準となるキーである。

【0072】

ステップS522に進み、キー表より、次リンクの指すキーをソートキーとして取り出すとともに、次リンクをソートキーの読出位置に設定する。そして、ステップS523において、リンク表より次リンクの指すリンクを読み出して退避エリアに退避する。ステップS522における次リンクは、図4Bに示すステップS413において、リンク表より読み出された親リンクの指すリンクである。図2Aの例の差分ビット位置0を有するキーの分類処理であれば、親リンクがP3、次リンクがP4、そして次リンクの指すリンクであって退避エリアに退避されるリンクはP1である。

20

【0073】

ステップS523において次リンクの指すリンクを退避するのは、次のステップS524において、ステップS522で設定したソートキーの、ステップS521で設定した最小値キーに対する差分ビット位置に基づく分類処理が行われ、その結果としてリンク表の次リンクの指すリンクの値が書き換えられるので、あらかじめ退避しておいて、ステップS524以後の、次リンクの指すリンクを用いた処理を正常に行うためである。

【0074】

ステップS524では、図5A及び図5Bを参照して詳細に説明した、ソートキーと最小値キーの差分ビット位置を得て差分ビット位置によりソートキーを分類する処理が実行される。

30

【0075】

そして、ステップS525において、次リンクと退避リンクは一致するか判定される。退避リンクが次リンクと一致すれば、同一差分ビット位置を有する分類対象となるキーは残っていないことから処理を終了する。

【0076】

一方、ステップS525において、退避リンクが次リンクと一致しないと判定されると、ステップS526において退避リンクを次リンクに設定してステップS522に戻り、新しく設定された次リンクをキー表の読出位置とするキーを分類対象のソートキーとして分類処理を継続する。

40

【0077】

上述のステップS522～S526のループ処理を、退避リンクが次リンクと一致するまで、すなわち分類処理の対象となるキーがなくなるまで繰り返し、分類処理の対象となるキーがなくなると処理を終了する。

【0078】

以上、本発明の最良の実施形態に詳細に説明した。以下においては、本発明の理解をさらに容易にするために、図1B及び図2Aの例示についての処理を、図6A～図7を参照して説明する。

50

## 【 0 0 7 9 】

図 6 A は、本発明の一実施の形態におけるキーを昇順に取り出す初段の処理のデータの流れを説明する図である。

図 6 A の ( a ) に示すのは、キー列 1 0 0 に含まれるキーをキー表 3 0 9 に設定するデータの流れである。

## 【 0 0 8 0 】

キー位置 1 0 0 a のキー “ 1 1 1 1 ” は、点線の矢印 6 5 1 a で示すように、キー表 3 0 9 の読出位置 P 1 に設定される。同様に、キー位置 1 0 0 b のキー “ 0 0 1 1 ” は、点線の矢印 6 5 1 b で示すように、キー表 3 0 9 の読出位置 P 2 に設定され、キー位置 1 0 0 c のキー “ 1 0 1 0 ” は、点線の矢印 6 5 1 c で示すように、キー表 3 0 9 の読出位置 P 3 に設定され、キー位置 1 0 0 e のキー “ 1 1 1 0 ” は、点線の矢印 6 5 1 e で示すように、キー表 3 0 9 の読出位置 P 4 に設定される。キー位置 1 0 0 d のキー “ 0 0 0 1 ” は、点線の矢印 6 5 1 d で示すように、最小値キーとしてキー表 3 0 9 の読出位置 P 0 に設定される。

## 【 0 0 8 1 】

図 6 A の ( b ) に示すのは、初段の分類処理で最小値キーがソート済みキーとして取り出され、最小値キーを基準のキーとした差分ビット位置によりソート対象のキーを分類した状態である。先に図 2 A に示したものと類似のものである。キー表 3 0 9 の先頭の読出位置 P 0 に設定された最小値キー “ 0 0 0 1 ” は、矢印 6 8 9 c に示すように、ソート済みキー表 3 1 3 の先頭の書込位置 6 4 0 a に設定される。

## 【 0 0 8 2 】

また、図の点線の矢印 6 6 0 a に示すように、差分ビット位置が 0 である 3 つのキーのうち最小のキーである “ 1 0 1 0 ” の読出位置 P 3 が差分ビット位置表 3 1 0 の差分ビット位置 0 に親リンクとして格納され、差分ビット位置が 2 である 1 つのキーであって最小のキーである “ 0 0 1 1 ” の読出位置 P 2 が差分ビット位置表 3 1 0 の差分ビット位置 2 に親リンクとして格納される。

## 【 0 0 8 3 】

リンク表 3 1 1 には、点線の矢印 6 7 3 b 及び 6 7 2 b に示すように、差分ビット位置表 3 1 0 に設定された親リンクの読出位置 6 2 1 に、同一の差分ビット位置を有するキーの読出位置が格納されている。すなわち、親リンク P 3 が指す読出位置 6 2 1 にはリンク 6 2 2 として P 4 が、親リンク P 2 が指す読出位置 6 2 1 にはリンク 6 2 2 として P 2 が格納されている。

## 【 0 0 8 4 】

点線の矢印 6 7 4 c に示すように、リンク P 4 が指す読出位置 6 2 1 にはリンク 6 2 2 として P 1 が格納され、リンク P 1 が指す読出位置 6 2 1 には、同一の読出位置 P 1 がリンク 6 2 2 として格納されている。これは、同一の差分ビット位置 0 を有するキーがそれ以上存在しないことを表している。すなわち、キー表 3 0 9 の読出位置 P 1 のキー “ 1 1 1 1 ” がリンク表 3 1 1 において同一差分ビット位置のキーの最後のキーであることを示している。

## 【 0 0 8 5 】

読出位置 6 2 1 が P 2 のリンク 6 2 2 に P 2 が格納されているのは、キー表 3 0 9 の読出位置 P 2 のキー “ 0 0 1 1 ” がリンク表 3 1 1 において同一差分ビット位置のキーの唯一のキーであることに対応している。

図 6 A の ( b ) に示す差分ビット位置表 3 1 0 とリンク表 3 1 1 の設定については、後に図 7 を参照して詳細に説明する。

## 【 0 0 8 6 】

図 6 A の ( c ) に示すのは、初段の分類処理で分類されたキー群のうち、差分ビット位置 6 1 1 が 2 である親リンク P 2 を読出位置とするキー “ 0 0 1 1 ” を含むキー群 ( 実際には 1 つのキーしか含まない。 ) の最小値キー “ 0 0 1 1 ” をソート済みキーとしてソート済みキー表 3 1 3 に格納する流れを説明する図である。

## 【 0 0 8 7 】

親リンク 6 1 2 は、差分ビット位置表 3 1 0 の差分ビット位置 6 1 1 の降順で取り出される。図の例の場合、矢印 6 8 2 で示す差分ビット位置 6 1 1 が 2 であるエントリの親リンク P 2 が取り出され、図の矢印 6 8 2 b に示すように、キー表 3 0 9 の親リンク P 2 の指す読出位置 6 0 1 からキー “ 0 0 1 1 ” が読み出され、矢印 6 8 2 c に示すようにソート済みキー表 3 1 3 の書込位置 6 4 0 b に書き出される。

## 【 0 0 8 8 】

一方、矢印 6 7 2 b に示す親リンク P 2 に対応するリンク表 3 1 1 の読出位置 6 2 1 に格納されたリンク 6 2 2 は親リンク 6 1 2 と同一の P 2 であるから、差分ビット位置 2 に関するさらなる分類は行われない。そこで、差分ビット位置表 3 1 0 の差分ビット位置 2 に格納されていた親リンク P 2 は、以降の処理の妨げにならないように削除される。

10

## 【 0 0 8 9 】

図 6 B は、本発明の一実施の形態におけるキーを昇順に取り出す中段の処理のデータの流れを説明する図である。

図 6 B の ( d ) に示すのは、図 6 A の ( c ) に示す差分ビット位置 2 についての処理に続いて、図の矢印 6 8 0 a で示すように差分ビット位置 0 についての分類処理を開始するときの、差分ビット位置表 3 1 0 とリンク表 3 1 1 の状態である。この状態は、差分ビット位置表の差分ビット位置 2 のエントリから親リンク P 2 が削除されている点を除いて、図 6 A の ( b ) に示すものと同様である。ただし、リンク表 3 1 1 の読出位置 P 2 のエントリの値は不必要なので省略している。

20

## 【 0 0 9 0 】

図 6 B の ( e ) に示すのは、図 6 A の ( b ) に示す最小値キー “ 0 0 0 1 ” を基準のキーとしたときの差分ビット位置が 0 であるキーの分類処理の流れを示すものである。図 1 B の例示では、差分ビット位置による分類 1 4 9 b に相当する。

## 【 0 0 9 1 】

矢印 6 8 0 a で示す差分ビット表 3 1 0 の差分ビット位置 0 の親リンク P 3 が取り出され、点線の矢印 6 8 3 b で示す P 3 を読出位置とするキー表 3 0 9 からの点線の矢印 6 8 3 d に示すように、キー表 3 0 9 の読出位置 P 3 に格納されたキー “ 1 0 1 0 ”、すなわち親リンク P 3 の指すキーが図 4 B に示すステップ S 4 1 2 でソート済みキーとして取り出されて一時記憶領域であるソート済みキー 6 5 0 に設定され、さらにソート済みキー表 3 1 3 の書込位置 6 4 0 c に書き出される。

30

## 【 0 0 9 2 】

一方、ソート済みキー 6 5 0 に設定された、先の分類対象キーのうちの最小値キーであるキー “ 0 0 0 1 ” について同一の差分ビット位置 0 を有するキーのうちの最小の値のキーであるキー “ 1 0 1 0 ” と、同一の差分ビット位置 0 を有するキー “ 1 1 1 1 ” と “ 1 1 1 0 ” との差分ビット位置 1 が計算され、図の点線の矢印 6 6 1 a が示すように、小さいほうのキーの読出位置である P 4 が差分ビット位置表 3 1 0 の差分ビット位置 6 2 1 が 1 であるエントリに親リンク 6 1 2 として格納される。

## 【 0 0 9 3 】

さらに矢印 6 7 4 b に示すように、親リンク P 4 の指すリンク表 3 1 1 のリンク 6 2 2 には読出位置 P 1 が格納され、その読出位置 P 1 が指すリンク表 3 1 1 のリンクには、矢印 6 7 1 c で示すように同じ P 1 が格納されている。

40

## 【 0 0 9 4 】

図 6 B の ( f ) に示すのは、図 6 B の ( e ) に示す最小値キー “ 1 0 1 0 ” を基準のキーとしたときの差分ビット位置が 3 であるキーの分類処理の流れを示すものである。図 1 B の例示では、差分ビット位置による分類 1 4 9 c に相当する。

## 【 0 0 9 5 】

矢印 6 8 1 a で示す差分ビット表 3 1 0 の差分ビット位置 1 の親リンク P 4 が取り出され、点線の矢印 6 8 4 b で示す P 4 を読出位置とするキー表 3 0 9 からの点線の矢印 6 8 4 d に示すように、キー表 3 0 9 の読出位置 P 4 に格納されたキー “ 1 1 1 0 ”、すなわ

50

ち親リンク P 4 の指すキーが図 4 B に示すステップ S 4 1 2 でソート済みキーとして取り出されて一次記憶領域であるソート済みキー 6 5 0 に設定され、さらにソート済みキー表 3 1 3 の書込位置 6 4 0 d に書き出される。

【 0 0 9 6 】

一方、ソート済みキー 6 5 0 に設定された、先の分類対象キーのうちの最小値キーであるキー “ 1 0 1 0 ” について同一の差分ビット位置 1 を有するキーのうち最小の値のキーであるキー “ 1 1 1 0 ” と、同一の差分ビット位置 1 を有するキー “ 1 1 1 1 ” との差分ビット位置 3 が計算され、図の点線の矢印 6 6 2 a が示すように、小さいほう、すなわち唯一のキーの読出位置である P 1 が差分ビット位置表 3 1 0 の差分ビット位置 6 2 1 が 3 であるエントリに親リンク 6 1 2 として格納される。

10

さらに矢印 6 7 1 b に示すように、親リンク P 1 の指すリンク表 3 1 1 のリンク 6 2 2 には親リンクと同じ値の P 1 が格納される。

【 0 0 9 7 】

図 6 C は、本発明の一実施の形態におけるキーを昇順に取り出す終段の処理のデータの流れを説明する図である。すなわち図 6 B の ( f ) に示す処理により、差分ビット位置表 3 1 0 には、差分ビット位置 3 のエントリの親リンク P 1 だけが格納されている。

【 0 0 9 8 】

差分ビット位置表 3 1 0 の差分ビット位置 6 1 1 の降順で親リンク 6 1 2 が取り出されるので、図の例の場合、矢印 6 8 3 a で示す差分ビット位置 6 1 1 が 3 であるエントリの親リンク P 1 が取り出され、図の矢印 6 8 1 b に示すように、キー表 3 0 9 の親リンク P 1 の指す読出位置 6 0 1 からキー “ 1 1 1 1 ” が読み出され、矢印 6 8 1 c に示すようにソート済みキー表 3 1 3 の書込位置 6 4 0 e に書き出される。

20

【 0 0 9 9 】

一方、矢印 6 7 1 b に示す親リンク P 1 に対応するリンク表 3 1 1 の読出位置 6 2 1 に格納されたリンク 6 2 2 は親リンク 6 1 2 と同一の P 1 であるから、差分ビット位置 3 に関するさらなる分類は行われぬ。そこで、差分ビット位置表 3 1 0 の差分ビット位置 3 に格納されていた親リンク P 1 は、削除される。

すると、差分ビット位置表 3 1 0 の全ての親リンクは削除され、全ての差分ビット位置についての処理は完了するので、すべての処理を終了する。

【 0 1 0 0 】

30

図 7 は、初段の分類処理のデータの流れ、すなわち、図 6 A の ( b ) に示す差分ビット位置表 3 1 0 とリンク表 3 1 1 の設定を説明する図である。

図 7 の ( a ) に示すのは、キー表 3 0 9 にキー 6 0 2 が設定され、差分ビット位置表 3 1 0 とリンク表 3 1 1 には何も設定されていない、初期状態である。

【 0 1 0 1 】

なお、読出位置 P 2 に格納されるキー “ 0 0 1 1 ” は省略されている。キー “ 0 0 1 1 ” については、最小値キー “ 0 0 0 1 ” に対する差分ビット位置が 2 であるキーが唯一のものである。したがって、図 5 A に示すステップ S 5 0 4 とステップ S 5 0 5 で、差分ビット位置表とリンク表 3 1 1 の読出位置 P 2 に関する設定が一度で完成するので、その設定の理解が容易なものであるから省略した。

40

【 0 1 0 2 】

図 7 の ( b ) に示すのは、キー表 3 0 9 の読出位置 P 1 に格納されたキー “ 1 1 1 1 ” に関する処理のデータの流れである。点線 6 8 0 d と 6 8 1 d で示すように、読出位置 P 0 に格納された最小値キー “ 0 0 0 1 ” とのビット列比較により差分ビット位置 0 が計算され、点線の矢印 6 6 0 a に示すように、差分ビット位置表 3 1 0 の差分ビット位置 0 に読出位置 P 1 が設定される。また、点線の矢印 6 7 1 b で示すように、リンク表 3 1 1 の読出位置 P 1 に、P 1 が設定される。

【 0 1 0 3 】

図 7 の ( c ) に示すのは、キー表 3 0 9 の読出位置 P 3 に格納されたキー “ 1 0 1 0 ” に関する処理のデータの流れである。点線 6 8 0 d と 6 8 3 d で示すように、読出位置 P

50



0に格納された最小値キー“0001”とのビット列比較により差分ビット位置0が計算される。そして、差分ビット位置表310の差分ビット位置0に設定されている親リンクP1を読出位置とするキー“1111”との大小比較が図5Bに示すステップS507で行われる。読出位置P3のキーの方が小さいので、点線の矢印660aに示すように、差分ビット位置表310の差分ビット位置0のエントリは読出位置P3に更新される。また、点線の矢印673bで示すように、リンク表311の読出位置P3に、P1が設定される。この設定により、矢印671cで示すように読出位置P3のキーと読出位置P1のキーの差分ビット位置が同一であるというリンク関係が保持される。

【0104】

図7の(d)に示すのは、キー表309の読出位置P4に格納されたキー“1110”に関する処理のデータの流れである。点線680dと684dで示すように、読出位置P0に格納された最小値キー“0001”とのビット列比較により差分ビット位置0が計算される。そして、差分ビット位置表310の差分ビット位置0に設定されている親リンクP3を読出位置とするキー“1010”との大小比較が図5Bに示すステップS507で行われる。読出位置P4のキーの方が大きいので、点線の矢印660aに示すように、差分ビット位置表310の差分ビット位置0の読出位置P3は更新されない。しかし、点線の矢印673bで示すように、リンク表311の読出位置P3のエントリはP4に更新される。そして、矢印674cで示すように、リンク表311の読出位置P4にP1が設定される。この設定により、矢印674cと矢印671cで示すように、読出位置P3のキー、読出位置P4のキー及び読出位置P1のキーの差分ビット位置が同一であるというリンク関係が保持される。

以上のデータの流れ(読出位置P2については省略)により、初段の分類処理による差分ビット位置表310とリンク表311の設定が行われる。

【0105】

以上本発明を実施するための最良の形態について詳細に説明したが、本発明の実施の形態はそれに限ることなく種々の変形が可能であることは当業者に明らかである。

また、本発明のビット列データソート装置が、図2Aに例示するデータ構造を有する記憶手段と図4A及び図4Bに示す処理をコンピュータに実行させるプログラムによりコンピュータ上に構築可能なことは明らかである。

したがって、上記プログラム、及びプログラムを記録したコンピュータ読み取り可能な記録媒体は、本発明の実施の形態に含まれる。

以上詳細に説明した本発明が提供する新しいビット列データソート手法を用いることにより、より高速なビット列データのソートを行うことが可能となる。

【図面の簡単な説明】

【0106】

【図1A】従来の基数ソートの概念を説明する図である。

【図1B】本発明の一実施の形態における差分ビット位置によるソート処理の概念を説明する図である。

【図2A】本発明の一実施の形態における差分ビット位置によるソート処理で用いるデータ構造を説明する図である。

【図2B】本発明の一実施の形態におけるビット列データソート装置の機能ブロック構成を説明する図である。

【図3】本発明を実施するためのハードウェア構成例を説明する図である。

【図4A】本発明の一実施の形態におけるソート処理の初段の分類処理とソート済みキーの出力処理の処理フローを説明する図である。

【図4B】本発明の一実施の形態におけるソート処理の初段以降の分類処理とソート済みキーの出力処理の処理フローを説明する図である。

【図5A】本発明の一実施の形態におけるキーを分類する処理の前段の処理フローを説明する図である。

【図5B】本発明の一実施の形態におけるキーを分類する処理の後段の処理フローを説明

する図である。

【図 5 C】本発明の一実施の形態における同一の差分ビット位置を有する複数のキーをさらに分類する処理の処理フローを説明する図である。

【図 6 A】本発明の一実施の形態におけるキーを昇順に取り出す初段の処理のデータの流れを説明する図である。

【図 6 B】本発明の一実施の形態におけるキーを昇順に取り出す中段の処理のデータの流れを説明する図である。

【図 6 C】本発明の一実施の形態におけるキーを昇順に取り出す終段の処理のデータの流れを説明する図である。

【図 7】初段の分類処理のデータの流れを説明する図である。

10

【符号の説明】

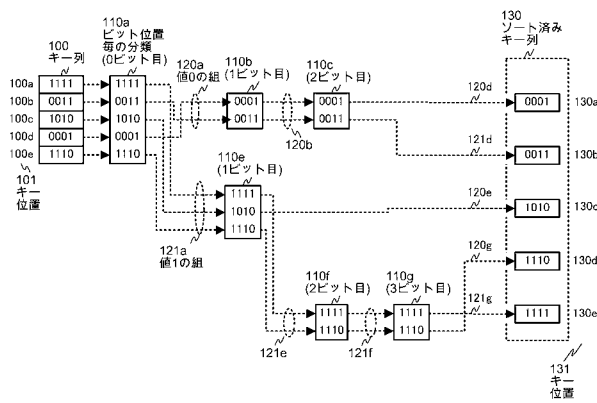
【 0 1 0 7 】

1 0 0	キー列
1 3 0	ソート済みキー列
1 4 9 a、1 4 9 b、1 4 9 c	差分ビット位置による分類
2 0 0	ビット列データソート装置
2 1 0	差分ビット位置計算手段
2 2 0	差分ビット位置分類手段
2 3 0	ソート済みキー出力手段
2 4 0	制御手段
3 0 1	データ処理装置
3 0 2	中央処理装置
3 0 3	キャッシュメモリ
3 0 4	バス
3 0 5	主記憶装置
3 0 6	外部記憶装置
3 0 7	通信装置
3 0 8	データ格納装置
3 0 9	キー表
3 1 0	差分ビット位置表
3 1 1	リンク表
3 1 3	ソート済みキー表

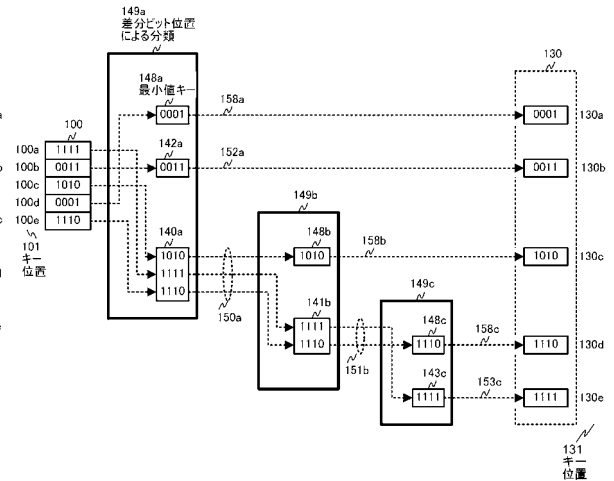
20

30

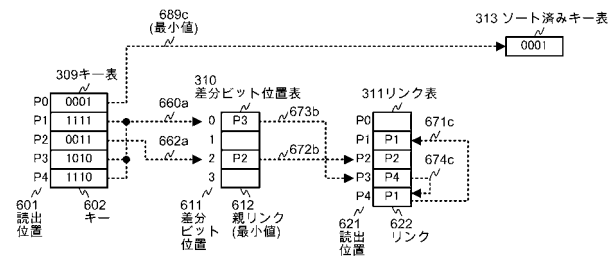
【図 1 A】



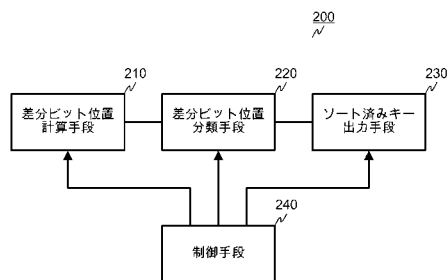
【図 1 B】



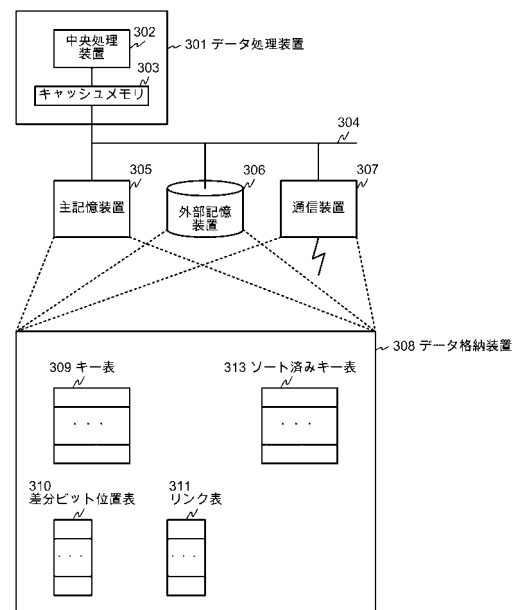
【図 2 A】



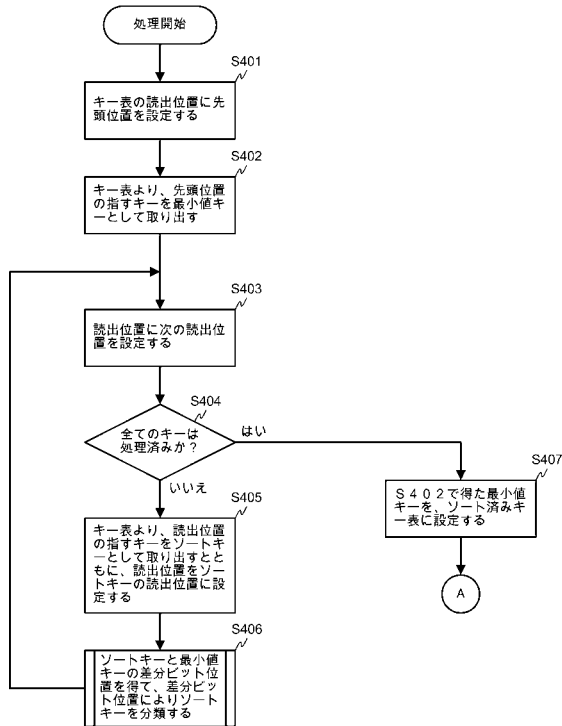
【図 2 B】



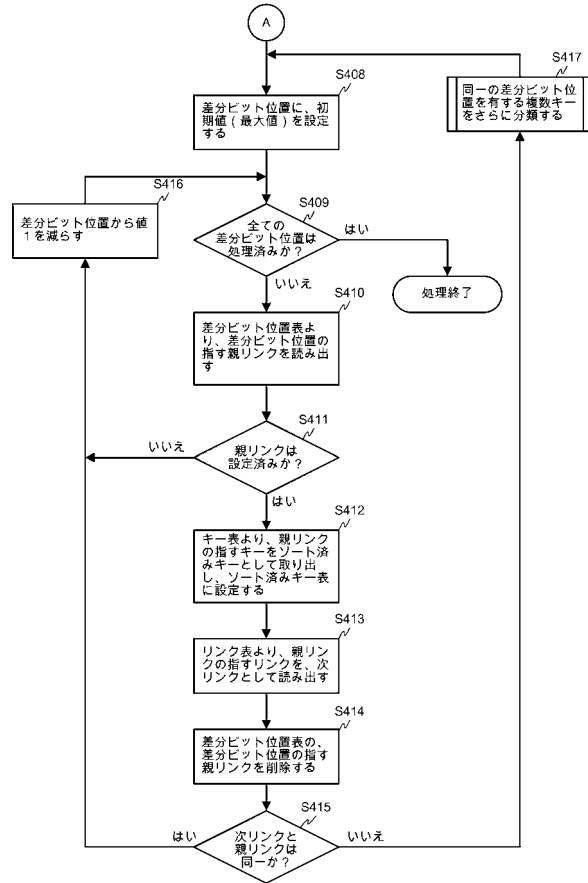
【図 3】



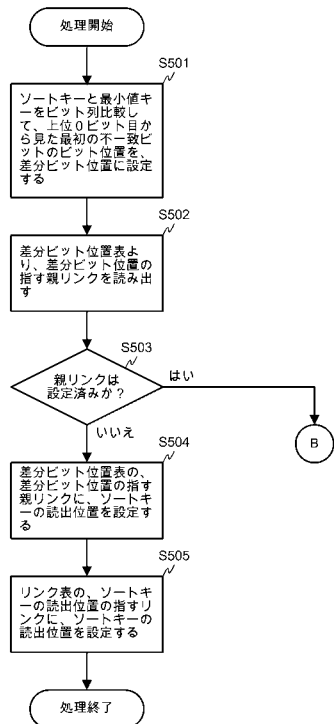
【図 4 A】



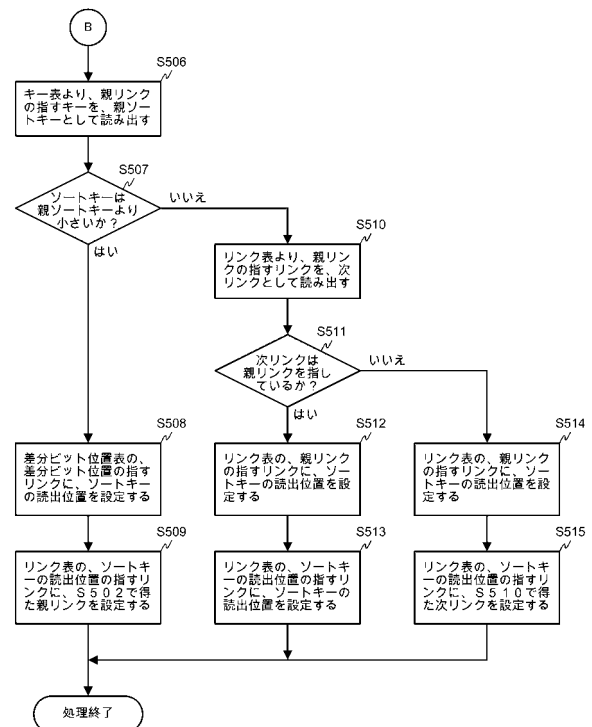
【図 4 B】



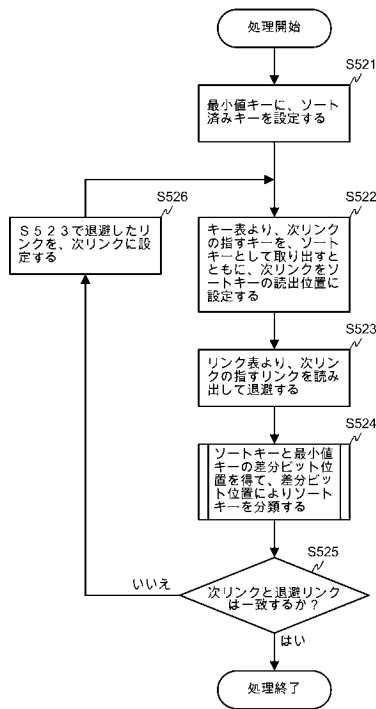
【図 5 A】



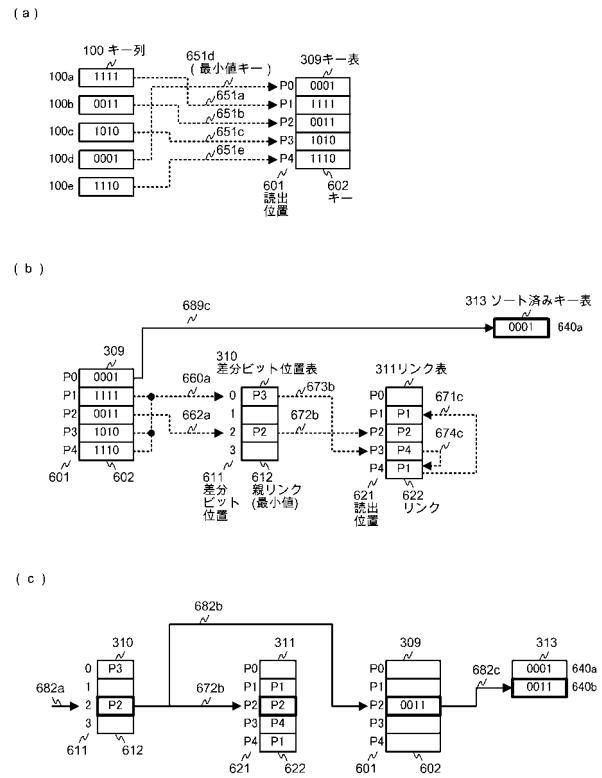
【図 5 B】



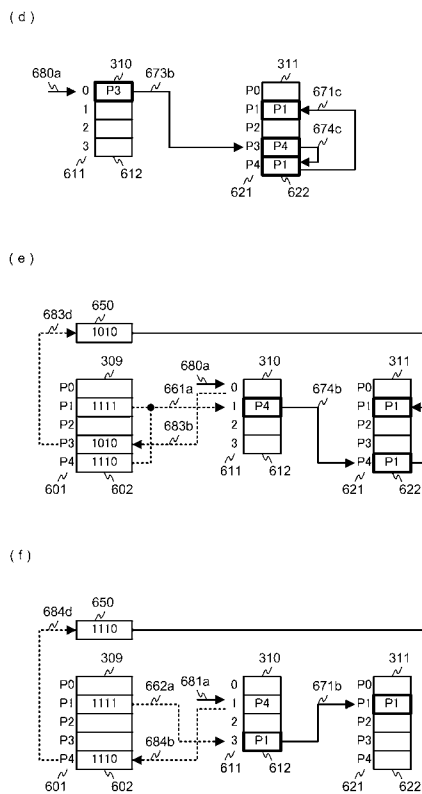
【図 5 C】



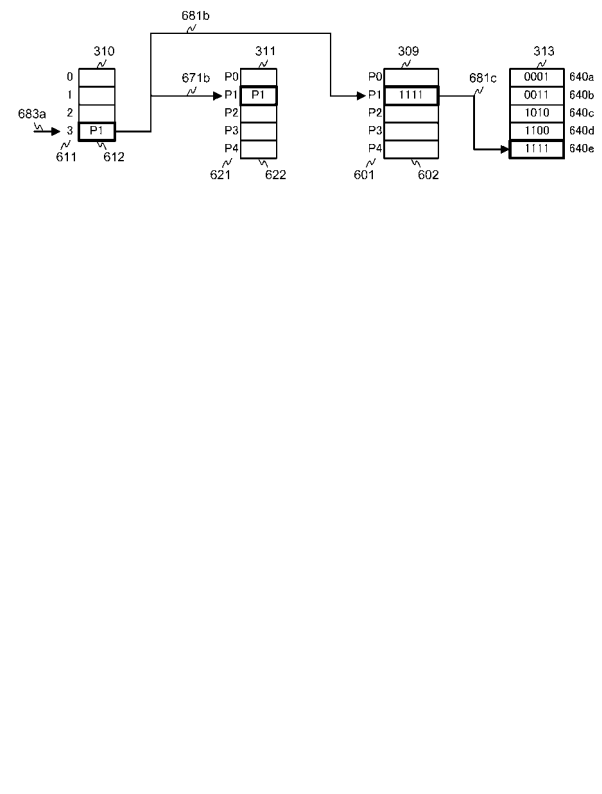
【図 6 A】



【図 6 B】

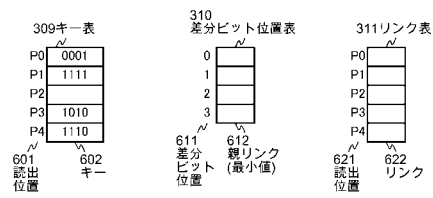


【図 6 C】

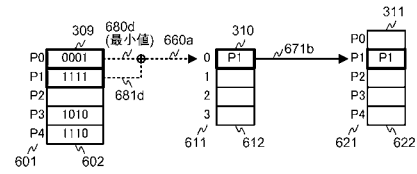


## 【図 7】

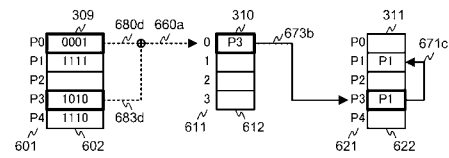
( a )



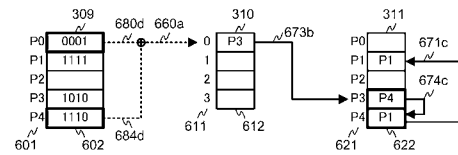
( b )



( c )



( d )



---

フロントページの続き

(56)参考文献 特開2007-122302(JP,A)  
特開平04-032924(JP,A)  
米国特許第5396622(US,A)  
米国特許第5440734(US,A)

(58)調査した分野(Int.Cl., DB名)  
G06F 7/24  
G06F 17/30