

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5143797号
(P5143797)

(45) 発行日 平成25年2月13日(2013.2.13)

(24) 登録日 平成24年11月30日(2012.11.30)

(51) Int.Cl. F 1
G 0 6 F 7/24 (2006.01) G 0 6 F 7/24 A

請求項の数 12 (全 16 頁)

<p>(21) 出願番号 特願2009-189602 (P2009-189602)</p> <p>(22) 出願日 平成21年8月18日 (2009.8.18)</p> <p>(65) 公開番号 特開2011-40009 (P2011-40009A)</p> <p>(43) 公開日 平成23年2月24日 (2011.2.24)</p> <p>審査請求日 平成24年7月27日 (2012.7.27)</p> <p>早期審査対象出願</p>	<p>(73) 特許権者 503002167 株式会社高速屋 神奈川県川崎市幸区新塚越201番地 ル リエ新川崎6-1</p> <p>(74) 代理人 100133570 弁理士 ▲徳▼永 民雄</p> <p>(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</p> <p>(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</p> <p>審査官 田中 友章</p> <p style="text-align: right;">最終頁に続く</p>
--	---

(54) 【発明の名称】 ビット列データソート装置、ソート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基準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の組毎に差分ビット位置順で各キーの当該差分ビット位置の下位nビットによるソートを行う下位nビットソート手段と、

前記基準キーと、前記下位nビットソート手段により差分ビット位置順で差分ビット位置毎にソートされたキーとを、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

10

20

を備え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 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の任意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 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、

前記同一の差分ビット位置を有するキーの前記ソート対象キー記憶手段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順次たどることにより、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 5】

請求項 4 に記載のビット列データソート装置において、

前記下位 n ビットソート手段は、親リンクが前記差分ビット位置表に格納された差分ビット位置毎に、前記差分ビット位置表の当該差分ビット位置に格納された親リンクと該親リンクの指す前記ソート対象キー記憶手段に記憶されたキーの当該差分ビット位置の下位 n ビットの組、及び該親リンクと同一の差分ビット位置を有する全てのキーのリンクと該リンクの指す前記ソート対象キー記憶手段に記憶されたキーの当該差分ビット位置の下位 n ビットの組をソート対象データとし、前記各組のキーの下位 n ビットをソートキーとして、前記ソート対象データをソートすることを前記差分ビット位置順に実行するものであり、

前記ソート済みキー出力手段は、差分ビット位置順に、前記下位 n ビットソート手段によりソートされたソート対象データの前記親リンクあるいはリンクの指す前記ソート対象キー記憶手段からキーを読み出してソート済みキーとして出力する、ことを特徴とするビット列データソート装置。

【請求項 6】

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基準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、

前記差分ビット位置計算ステップで求めた前記差分ビット位置毎に、前記分類対象キー

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の組毎に差分ビット位置順で各キーの当該差分ビット位置の下位 n ビットによるソートを行う下位 n ビットソートステップと、

前記基準キーと、前記下位 n ビットソートステップにおいて差分ビット位置順で差分ビット位置毎にソートされたキーとを、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、

を備えることを特徴とするビット列データソート方法。

10

【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 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の任意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。

20

【請求項 9】

請求項 8 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ソート対象キー記憶手段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順次たどることにより、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。

30

【請求項 10】

請求項 9 に記載のビット列データソート方法において、

前記下位 n ビットソートステップは、親リンクが前記差分ビット位置表に格納された差分ビット位置毎に、前記差分ビット位置表の当該差分ビット位置に格納された親リンクと該親リンクの指す前記ソート対象キー記憶手段に記憶されたキーの当該差分ビット位置の下位 n ビットの組、及び該親リンクと同一の差分ビット位置を有する全てのキーのリンクと該リンクの指す前記ソート対象キー記憶手段に記憶されたキーの当該差分ビット位置の下位 n ビットの組をソート対象データとし、前記各組のキーの下位 n ビットをソートキーとして、前記ソート対象データをソートすることを前記差分ビット位置順に実行し、

40

前記ソート済みキー出力ステップは、差分ビット位置順に、前記下位 n ビットソートステップにおいてソートされたソート対象データの前記リンクの指す前記ソート対象キー記憶手段からキーを読み出してソート済みキーとして出力する、ことを特徴とするビット列データソート方法。

【請求項 11】

請求項 6 ~ 請求項 10 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。

50

【請求項 1 2】

請求項 6 ~ 請求項 1 0 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】

【0001】

本発明は、ビット列データのソート装置、ソート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。

10

【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データの検索に帰着されるということができる。

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データからなるキーのソート処理に帰着される。なお、以下においては、ビット列データを、単にキーということもある。

20

【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。

【0005】

図 1 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 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 の例示により、基数ソートの概念を説明する。

30

【0006】

図 1 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 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“0 0 0 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“1 1 1 1”、“0 0 1 1”、“1 0 1 0”、“1 1 1 0”が存在する。

【0007】

図 1 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。

40

【0008】

ビット位置毎の分類（1 ビット目）1 1 0 b では、キー“0 0 0 1”とキー“0 0 1 1”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 1 2 0 b しか得られず、0 ビット目の値 0 の組 1 2 0 a 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）1 1 0 c により 2 ビット目による分類が行われる。ビット位

50

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。

【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。

【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。

【先行技術文献】

【特許文献】

【0012】

【特許文献1】特開2002-116907号公報

【特許文献2】特開2005-316663号公報

【発明の概要】

【発明が解決しようとする課題】

【0013】

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、無効となる処理の発生を少なくした、効率的なソート手法を提供することである。

【課題を解決するための手段】

【0014】

本発明のソート処理によれば、基準となるビット列データである基準キーとソート対象であるビット列データであるソート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、ソートキーの差分ビット位置の下位nビット値によるソートを行うことにより、ソート済みキー列を取得する。ここで、ソートキーの差分ビット位置の下位nビット値とは、ソートキーの差分ビット位置がbとすると、差分ビット位置は0から始まることとしているので、ソートキーから上位b+1ビットを除く残りのビット列のビット値である。

【発明の効果】

【 0 0 1 5 】

本発明によれば、最初に差分ビット位置によりキーの分類を行うため、分類が行われな
い無効な処理の発生をなくすることができる。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。

【 図面の簡単な説明 】

【 0 0 1 6 】

【 図 1 】 従来の基数ソートの概念を説明する図である。

【 図 2 A 】 本発明の一実施の形態における差分ビット位置と下位ビット値によるソート処
理の概念を説明する図である。

【 図 2 B 】 本発明の一実施の形態におけるキーを差分ビット位置順に分類するためのデー
タ構造例を説明する図である。

10

【 図 2 C 】 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
例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 4 A 】 本発明の一実施の形態におけるソート処理の前段の処理フロー例を説明する図
である。

【 図 4 B 】 本発明の一実施の形態におけるソート処理の後段の処理フロー例を説明する図
である。

【 図 5 A 】 本発明の一実施の形態における各キーを差分ビット位置毎に分類する処理フロ
ー例を説明する図である。

20

【 図 5 B 】 本発明の一実施の形態における各差分ビット位置に分類されたキーをソートし
、ソート済みキーとして出力する処理フロー例を説明する図である。

【 発明を実施するための形態 】

【 0 0 1 7 】

以下、本発明を実施するための形態を、図 1 に示したキー列 1 0 0 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。

【 0 0 1 8 】

図 2 A 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と下位ビットのビット
値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 1 0 0 及びソート済
みキー列 1 3 0 は、図 1 に示すものと同じである。

30

【 0 0 1 9 】

キー列 1 0 0 を構成する各キーは、差分ビット位置による分類 1 4 9 a において、最小値
キー 1 4 8 a、最小値キーとの差分ビット位置が 2 であるキー群 1 4 2 a 及び最小値キー
1 4 8 a との差分ビット位置が 0 であるキー群 1 4 0 a に分類される。図の例では、最小
値キー 1 4 8 a は “ 0 0 0 1 ” である。キー群 1 4 2 a には、キー “ 0 0 1 1 ” のみが含
まれており、キー群 1 4 0 a にはキー “ 1 1 1 1 ”、“ 1 0 1 0 ” 及び “ 1 1 1 0 ” が含
まれている。最小値キー 1 4 8 a は、キー列 1 0 0 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。

【 0 0 2 0 】

最小値キー 1 4 8 a は、図の点線の矢印 1 5 8 a に示すように、ソート済みキー列 1 3 0
のキー位置 1 3 0 a に格納される。また、キー群 1 4 2 a のキーは 1 つのみであるので、
図の点線の矢印 1 5 2 a に示すように、ソート済みキー列 1 3 0 のキー位置 1 3 0 b に格
納される。

40

【 0 0 2 1 】

最小値キー 1 4 8 a との差分ビット位置が 0 であるキー群 1 4 0 a にはキーが複数含まれ
ているので、図の点線の矢印の組 1 5 0 a に示すように、差分ビット位置の下位 n ビット
によるソート 1 4 9 b におけるソートの対象となる。差分ビット位置の下位 n ビットによ
るソート 1 4 9 b におけるソートでは、差分ビット位置による分類 1 4 9 a で分類に用い
られた差分ビット位置 0 の下位ビット、図 2 A の例では 1 ビット目から 3 ビット目までの
3 ビットを切り出してソートする。

50

【0022】

上述のソートにより、切り出された差分ビット位置の下位 n ビットはそれぞれ別々の組に完全に分類される。図 2 A の例では、ビット値が “ 0 1 0 ” の組 1 4 8 b、ビット値が “ 1 1 0 ” の組 1 4 8 c、ビット値が “ 1 1 1 ” の組 1 4 3 c に分類される。そして、点線の矢印 1 5 8 b、1 5 8 c、1 5 3 c にそれぞれ示すように、差分ビット位置までのビット値を付加して復元したキー “ 1 0 1 0 ”、“ 1 1 1 0 ”、“ 1 1 1 1 ” がキー位置 1 3 0 c、1 3 0 d、1 3 0 e に格納され、全体のソート処理が完了する。

【0023】

なお、差分ビット位置の下位 n ビットによるソート方法、図 2 A の例示ではソート 1 4 9 b におけるソート方法には限定はなく、任意のソート方法を用いることができる。

また、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ーとしているが、最大値キーを基準のキーとすることもできる。この場合には、図 2 A のソート対象キーの例では、最大値キーはキー “ 1 1 1 1 ” であり、差分ビット位置による分類の結果として、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 ”、差分ビット位置が 0 のキー “ 0 0 1 1 ” と “ 0 0 0 1 ” の 3 つの組が得られる。

【0024】

図 2 B 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 B 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。

【0025】

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されるものである。図 2 B の例示では、ソート対象の 5 つのキー 6 0 2 は、図 1 及び図 2 A 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。

【0026】

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、キー表 3 0 9 から最初に読み出されたキーの読出位置 6 0 1 が親リンク 6 1 2 として差分ビット位置 6 1 1 毎に格納されている。図 2 B 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 “ 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 ” のうち最初に読み出されるキー “ 1 1 1 1 ” の読出位置 P 1 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。なお、親リンクに設定される読出位置は、上述のように最初に読み出されるキーの読出位置に限ることはない。後述のリンク表 3 1 1 に格納されたリンク関係と併せて、最小値キーに対して同一の差分ビット位置を有するキーにアクセスすることができればよい。

【0027】

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。

【0028】

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 1 b で示すように、同一の差分ビット位置 0 を有する親リンクのキーの読出位置 P 1 に対して、その読出位置 P 1 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 1 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 を有する親リ

10

20

30

40

50

リンクのキー“1111”と同一の差分ビット位置0を有するキー“1110”のキー表309の読出位置P4が格納されている。

【0029】

そして、図の点線の矢印674cで示すように、リンク表311の読出位置P4に、同一の差分ビット位置0を有するキー“1010”のキー表309の読出位置P3が格納されている。また、図の点線の矢印673cで示すように、リンク表311の読出位置P3には、同一の値のキー表309の読出位置P3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。

【0030】

また、差分ビット位置表310からリンク表311への点線の矢印672bで示すように、同一の差分ビット位置2を有する親リンクのキーの読出位置P2に対して、リンク表311の読出位置P2には、リンク表311の読出位置P2と同一の値のキー表309の読出位置P2が格納されている。これは、キー表309の読出位置P2のキー“0011”が最小値キー“0001”との差分ビット位置が2であるキーの唯一のものであることを示している。

【0031】

上述の図2Bに示す差分ビット位置表310とリンク表311の状態は、図2Aに示す差分ビット位置による分類149a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となり、上述のキー表309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310とリンク表311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309の読出位置601はキーの識別情報に相当する。

【0032】

図2C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。

図に示すように、ビット列データソート装置200は、差分ビット位置計算手段210、差分ビット位置分類手段220、下位nビットソート手段225、及びソート済みキー出力手段230を含む。以下、各手段の動作の概要について、図2Bの例示を参照して説明する。

【0033】

差分ビット位置計算手段210は、キー表309に記憶された全てのキー602のうち、基準キー以外のキーについて、基準キーに対する差分ビット位置を計算する。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P0に格納された最小値キー“0001”である。ソート済みキー出力手段230は、最小値キー“0001”をソート済みキー表313に出力する。

【0034】

差分ビット位置分類手段220は、差分ビット位置計算手段210の計算結果に基づいて、差分ビット位置表310とリンク表311の書き込みを行う。その結果、キーの識別情報をキー表309の読出位置601として、差分ビット位置が0の組(P3、P4、P1)と差分ビット位置が2の組(P2)に分類される。

【0035】

下位nビットソート手段225は、差分ビット位置分類手段220により分類された各差分ビット位置の組について、差分ビット位置の下位のnビットによりソートを実行する。

ソート済みキー出力手段230は、下位nビットソート手段225によるソート結果に基づいて、ソート済みのキーをソート済みキー表313に出力する。

【0036】

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302及びキャッシュメモリ303を少なくとも

10

20

30

40

50

も備えたデータ処理装置301によりデータ格納装置308を用いて実施される。キー表309、差分ビット位置表310、リンク表311、ソート済みキー表313を有するデータ格納装置308は、主記憶装置305または外部記憶装置306で実現することができ、あるいは通信装置307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。

【0037】

図3の例示では、主記憶装置305、外部記憶装置306及び通信装置307が一本のバス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、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

10

【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。

【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2Aあるいは図2Bの例示を適宜用いて

20

【0040】

図4Aは、本発明の一実施の形態におけるソート処理の前段の処理フロー例を説明する図である。この前段の処理は、ソートキーを差分ビット位置毎に分類するとともに、基準キーである最小値キーをソート済みキー表に設定するものである。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2Bに示すようにキー表309の読出位置P0には最小値キーが記憶されているものとする。

【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」が図示しない一時記憶領域であることを省略して述べることもある。

30

上記の設定により、図2B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、図示しない一時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。

40

【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差分ビット位置によ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

50

て、差分ビット位置によりソートキーを分類し、ステップS 4 0 3に戻る。

ステップS 4 0 6の処理においては、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5 Aを参照して説明する。

【0045】

一方、ステップS 4 0 4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S 4 0 7に分岐し、ステップS 4 0 2で得た最小値キーを、ソート済みキー表に設定し、図4 Bに示すステップS 4 0 8以下の処理に進む。以上の処理により、図2 Bの例示のように、差分ビット位置表3 1 0とリンク表3 1 1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3 1 3に格納される。

【0046】

図4 Bは、本発明の一実施の形態におけるソート処理の後段の処理フロー例を説明する図である。この後段の処理は、順次、差分ビット位置毎に分類されたソートキーを差分ビット位置の下位nビットによりソートし、ソート済みキーとしてソート済みキー表に出力するものである。

【0047】

ステップS 4 0 8においては、図示しない一時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 B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。

【0048】

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。

【0049】

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時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時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。

【0050】

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 3 aに進む。

【0051】

ステップS 4 1 3 aでは、リンク表より、ステップS 4 1 0で読み出した親リンク（キー表の読出位置）の指すリンク列に含まれるキーをソートし、ソート済みキー表に設定してステップS 4 1 6に進む。ステップS 4 1 3 aの処理の詳細は、後に図5 Bを参照して説明する。また、親リンクの指すリンク列に含まれるキーについても後に説明する。

【0052】

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。

【0053】

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。一方、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

10

20

30

40

50

合でも、差分ビット位置の大きいキーほど基準キーの値に近いのであるから、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当業者に明らかである。

【0054】

図5Aは、図4Aに示すステップS406の処理の詳細を示すものであり、本発明の一実施の形態における各キーを差分ビット位置毎に分類する処理フロー例を説明する図である。本実施の形態においては、差分ビット位置表とリンク表を生成することにより、分類対象の各キーは差分ビット位置毎に分類される。

【0055】

図5Aに示すように、ステップS501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4Aに示すステップS405で設定されるものであり、図5Aに示す分類処理の対象のキーである。

10

【0056】

次にステップS502で差分ビット位置表より、ステップS501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S503で親リンクは設定済みであるか判定する。

【0057】

親リンクが設定済みでなければ、ステップS504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S505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。

20

なお、ステップS504の処理におけるソートキーの読出位置は、図4Aに示すステップS405で設定されるものである。

【0058】

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、最初に読み出された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S504の処理は、まさに最初に読み出された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。

【0059】

一旦親リンクが設定済みとなった後、すなわちステップS503での判定が親リンクは設定済みになると、ステップS510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S511で、次リンクと親リンクは同一か判定する。

30

【0060】

ステップS511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S512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S513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。

【0061】

一方、ステップS511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S514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S515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S510で得た次リンクを設定して処理を終了する。

40

【0062】

上述のステップS511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。

50

【 0 0 6 3 】

分類済みのキーが2つ以上のときは、ステップS 5 1 4とステップS 5 1 5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。

【 0 0 6 4 】

上述の説明から明らかとなっており、同一の差分ビット位置を有するソートキーの読出位置は、差分ビット位置表に格納された親リンクからリンク表の読出位置とリンクをたどることにより順次全てアクセスすることができる。そこで、以下の説明において、この親リンクからたどることのできる親リンクも含めた読出位置の列を親リンクの指すリンク列といい、親リンクの指すリンク列に含まれる読出位置のキー表に記憶されたキーを、親リンクの指すリンク列に含まれるキーということがある。

10

【 0 0 6 5 】

図5 Bは、図4 Bに示すステップS 4 1 3 aの処理である、本発明の一実施の形態における各差分ビット位置に分類されたキーをソートし、ソート済みキーとして出力する処理フロー例を説明する図である。本実施の形態においては、リンク表に基づいて親リンクの指すリンク列に含まれるキーを取り出してソートし、ソート済みキー表に設定する。

【 0 0 6 6 】

図に示すように、ステップS 5 4 1において、リンク表より、親リンクの指すリンク列のリンクを取り出す。親リンクは、図4 Bに示すステップS 4 1 0で読み出したものである。

20

【 0 0 6 7 】

ステップS 5 4 2に進み、キー表よりステップS 5 4 1で取り出したリンクの指すキーを読み出し、リンクとともにキーの差分ビット位置の下位nビットをソートキーとしてソート対象データに設定する。

【 0 0 6 8 】

図2 Bに示す例で、差分ビット位置が0の場合には、親リンクはP 1、親リンクの指すリンク列のリンクは、P 1、P 4、P 3であるから、ソート対象データには、(P 1、“1 1 1”)、(P 4、“1 1 0”)、(P 3、“0 1 0”)の3つのデータが設定される。ソートキーは、それぞれ差分ビット位置の下位3ビットである、“1 1 1”、“1 1 0”、“0 1 0”、である。

30

【 0 0 6 9 】

次に、ステップS 5 4 3においてソート対象データをソートする。上述の例で昇順にソートすると、ソート結果のデータの並びは、(P 3、“0 1 0”)、(P 4、“1 1 0”)、(P 1、“1 1 1”)となる。

【 0 0 7 0 】

最後にステップS 5 4 4で、ソート順に、ソート対象データのリンクを取り出し、キー表より、リンクの指すキーを読み出し、ソート済みキー表に順次設定し、処理を終了する。

40

【 0 0 7 1 】

上述の例では、まず、リンクP 3が取り出され、キー表より読出位置P 3のキー“1 0 1 0”が読み出されてソート済みキー表に設定される。次に、リンクP 4が取り出され、キー表より読出位置P 4のキー“1 0 1 0”が読み出されてソート済みキー表に設定される。最後に、リンクP 1が取り出され、キー表より読出位置P 1のキー“1 1 1 1”が読み出されてソート済みキー表に設定される。

【 0 0 7 2 】

以上本発明を実施するため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。

例えば、図5 Bに示すソート処理では、キー表よりステップS 5 4 1で取り出したリンク

50

の指すキーを読み出し、リンクとともにキーの差分ビット位置の下位 n ビットをソートキーとしてソート対象データに設定することにより、ソートキーのサイズを小さくしているが、キー自体をソート対象データに設定し、ソートキーをキーの差分ビット位置の下位 n ビットとしてソートしても同一のソート結果が得られることは明らかである。

【 0 0 7 3 】

また、リンクをソート対象データに含めず、差分ビット位置までの上位のビット値を記憶しておき、ソート後にキーの差分ビット位置の下位 n ビットと合成してキーを復元することが可能であることも明らかである。処理は複雑になるが、リンクに要するビット数と差分ビット位置の大きさとの関係により、ソート対象データを、リンクを含めたものとするかしないかを切り替えることも可能である。

10

差分ビット位置が大きい場合には、先頭から差分ビット位置までのビットをソート対象から除くことにより、ソート対象データのデータサイズを縮小することができる。

【 0 0 7 4 】

さらに、本発明のビット列データソート装置が、図 2 B に例示するデータ構造を有するデータを記憶する記憶手段と図 4 A 及び図 4 B 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。

20

【 符号の説明 】

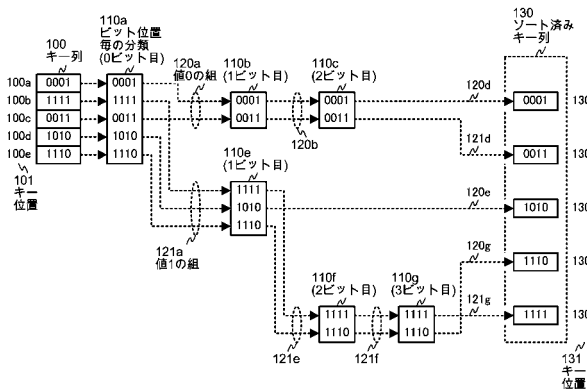
【 0 0 7 5 】

- 1 0 0 キー列
- 1 3 0 ソート済みキー列
- 1 4 9 a 差分ビット位置による分類
- 1 4 9 b 差分ビット位置の下位 n ビットによるソート
- 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 2 5 下位 n ビットソート手段
- 2 3 0 ソート済みキー出力手段
- 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

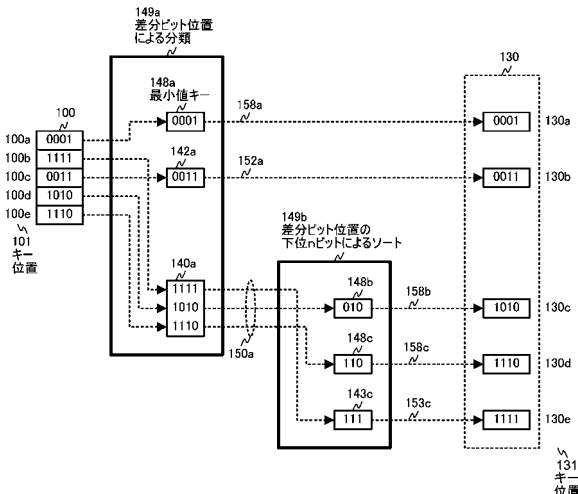
30

40

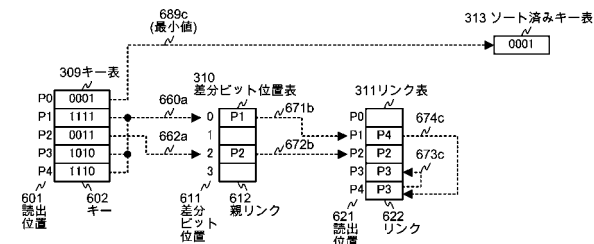
【図1】



【図2A】



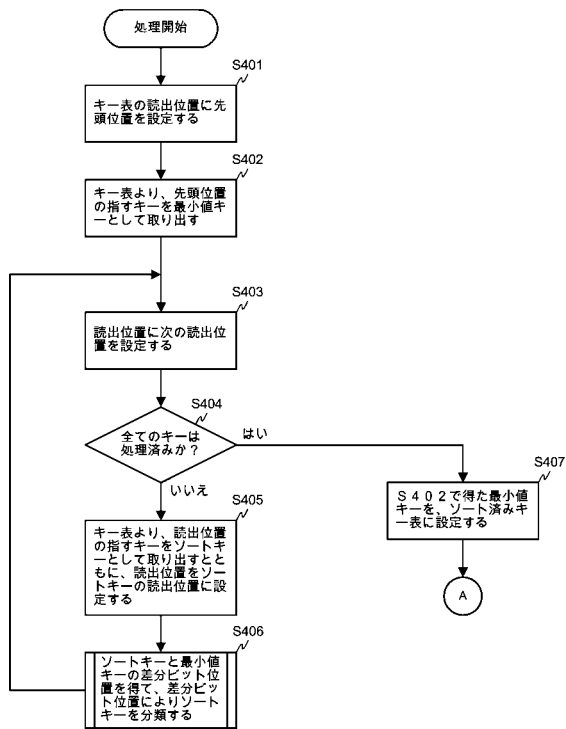
【図2B】



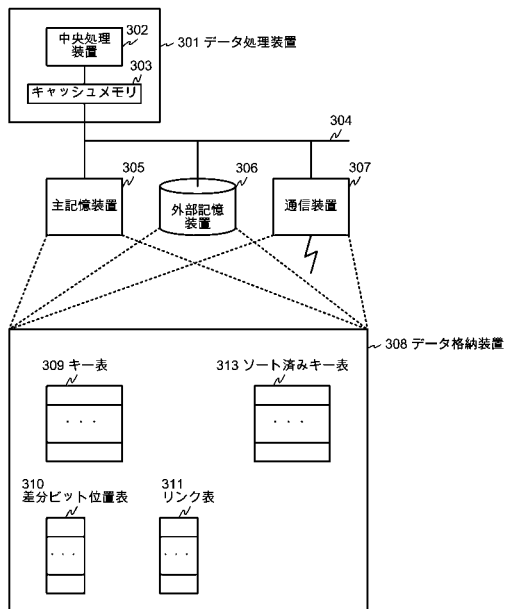
【図2C】



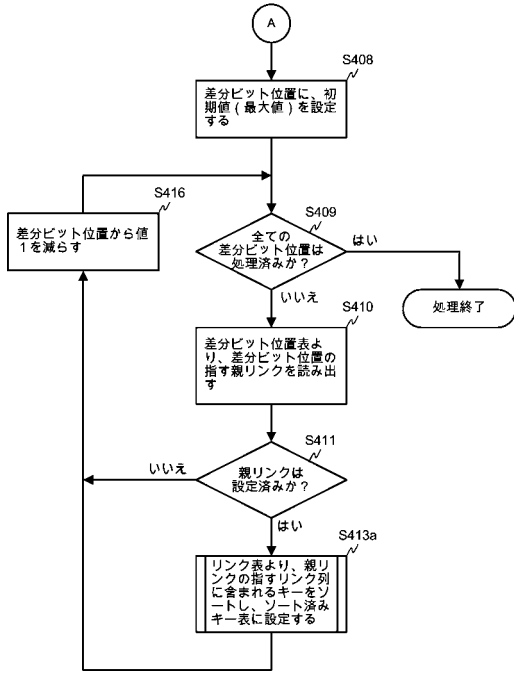
【図4A】



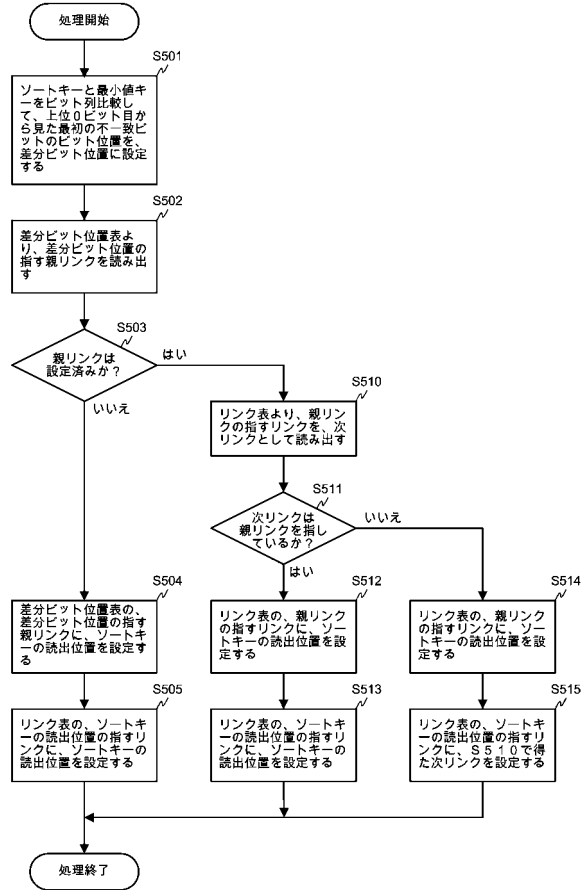
【図3】



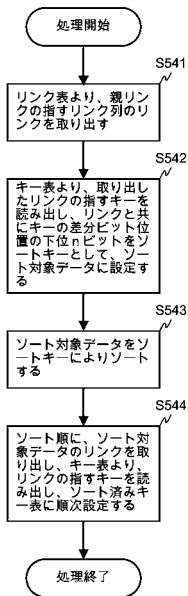
【図4B】



【図5A】



【図5B】



フロントページの続き

- (56)参考文献 米国特許第4595995 (US, A)
特開平07-302187 (JP, A)
特開2010-092088 (JP, A)
特開平04-127321 (JP, A)
国際公開第2010/073471 (WO, A1)

- (58)調査した分野(Int.Cl., DB名)
G06F 7/24