

(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  
2017年8月31日(31.08.2017)



(10) 国際公開番号  
WO 2017/145317 A1

- (51) 国際特許分類:  
H03M 7/30 (2006.01) G06F 17/30 (2006.01)
- (21) 国際出願番号: PCT/JP2016/055583
- (22) 国際出願日: 2016年2月25日(25.02.2016)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人: 楽天株式会社(RAKUTEN, INC.) [JP/JP]; 〒1580094 東京都世田谷区玉川一丁目14番1号 Tokyo (JP).
- (72) 発明者: 金田 悠作(KANETA, Yusaku); 〒1580094 東京都世田谷区玉川一丁目14番1号楽天株式会社内 Tokyo (JP).
- (74) 代理人: 齋藤 拓也, 外(SAITO, Takuya et al.); 〒1000005 東京都千代田区丸の内1-7-12 サピアタワー 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,

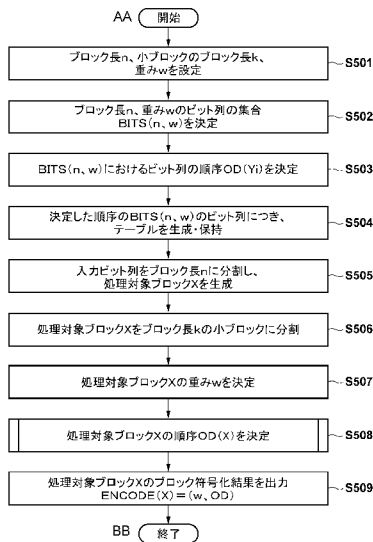
- BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW).
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[続葉有]

(54) Title: BLOCK ENCODING DEVICE, BLOCK DECODING DEVICE, INFORMATION PROCESSING DEVICE, PROGRAM, BLOCK ENCODING METHOD AND BLOCK DECODING METHOD

(54) 発明の名称: ブロック符号化装置、ブロック復号化装置、情報処理装置、プログラム、ブロック符号化方法及びブロック復号化方法

[図5]



- S501 Set block length n, block length k of small block, weight w
- S502 Determine set BITS(n, w) of bit strings having block length n and weight w
- S503 Determine orders OD(Y) of bit strings in BITS(n, w)
- S504 Generate and hold tables regarding BITS(n, w) bit strings of determined orders
- S505 Divide input bit string into block lengths n to generate block X to be processed
- S506 Divide block X to be processed into small blocks having block length k
- S507 Determine weight w of block X to be processed
- S508 Determine order OD(X) of block X to be processed
- S509 Output block encoding result of block X to be processed
- AA Start
- BB End

(57) Abstract: The purpose of the present invention is to provide an efficient encoding method and decoding method for block compression in order to implement RRR in a practical manner, and the present invention provides an RRR implementation that provides a high-speed and area-saving encoding method and decoding method, and that operates at high speed even with respect to a dense bit string. A block encoding device generates blocks to be processed by dividing an input bit string into a plurality of blocks having a first block length, determines a first weight of each block to be processed from the number of 1 values included in the bit string of the block to be processed, divides each block to be processed into small blocks having a second block length, determines a first order of the blocks to be processed in a set in which predetermined bit strings having the first block length and the first weight are arranged in a predetermined order on the basis of the weight and value of each of the small blocks, and generates block encoding data by converting the bit string of each block to be processed into the values of the determined first weight and first order.

(57) 要約:

[続葉有]

WO 2017/145317 A1



規則 4.17 に規定する申立て:

添付公開書類:

- 不利にならない開示又は新規性喪失の例外に  
関する申立て (規則 4.17(v))
- 国際調査報告 (条約第 21 条(3))

---

RRR の実用的な実装のために、ブロック圧縮に対する効率良い符号化法と復号化法の提供を目的とし、高速で省領域な符号化法と復号化法を与え、密なビット列に対しても高速に動作する RRR 実装を提供する。ブロック符号化装置は、入力ビット列を第 1 のブロック長の複数のブロックに分割して処理対象ブロックを生成し、処理対象ブロックのビット列に含まれる 1 の値の数から処理対象ブロックの第 1 の重みを決定し、前記処理対象ブロックを第 2 のブロック長の小ブロックに分割し、小ブロック毎の重みと値とに基づき前記第 1 のブロック長と前記第 1 の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第 1 の順序を決定し、前記処理対象ブロックのビット列を、前記決定された第 1 の重みと第 1 の順序との値に変換してブロック符号化データを生成する。

## 明 細 書

発明の名称：

ブロック符号化装置、ブロック復号化装置、情報処理装置、プログラム、  
ブロック符号化方法及びブロック復号化方法

### 技術分野

[0001] 本願発明は、ブロック符号化装置、ブロック復号化装置、情報処理装置、  
プログラム、ブロック符号化方法及びブロック復号化方法に関する。

### 背景技術

[0002] 大規模なデータを扱うアプリケーションでは、速度とともに作業領域の大きさも問題となる。作業領域が主記憶に収まらない場合、スワッピングが発生し、大幅な速度低下を招くこととなる。そのためデータ構造には、高速に検索を可能とするだけでなく、作業領域量が小さいことが求められている。

[0003] そこで近年、二次記憶装置（ハードディスク等）を参照することなく主記憶上での高速な大規模データ処理を目的に、データを情報理論的下限に近いサイズで表現したまま高速な操作を実現する簡潔データ構造、および、データをエントロピーに近いサイズに圧縮したまま高速な操作を実現する圧縮データ構造が理論・実用の両面から研究されている。

[0004] Rank/Select 辞書は、入力ビット列 $B[0..n-1]$  上の $\text{rank}_b(x, B)$  ( $B[0..x]$  中の $b \in \{0, 1\}$ の個数を求める操作) と $\text{select}_b(i, B)$  ( $B$ の先頭から $i$ 番目の $b \in \{0, 1\}$ の位置を求める操作) を実現するデータ構造である。Ramanらは、入力ビット列を圧縮したまま上述の操作を定数時間で実現するデータ構造 (RRR) を提案した(非特許文献1)。このデータ構造は、入力ビット列を長さ $\log(n)/2$ の固定長ブロックに分割し、各ブロックを重み(ブロック中の1の個数を表す固定長整数)と順序(同じ重みのブロック群に一意に割り振られた可変長整数)の組として符号化する。

[0005] このブロック符号化とその逆操作であるブロック復号化は、理論的には漸近的に $n$ より小さい補助領域を用いて定数時間で実現できる。一方、実用的に

はこの補助領域は無視できないほど大きい。したがって、効率良いブロック符号化・ブロック復号化の実現は、上述のデータ構造を実装する上で重要な課題である。

[0006] また、RRR の実用的な実装方法として、Navarro とProvidel の実装方法（以降、RRR-NP という）が提案されている（非特許文献2）。非特許文献2の提案手法では、RRR の実性能はブロック圧縮の実現方法に大きく依存している。

### 先行技術文献

#### 非特許文献

[0007] 非特許文献1 : R. Raman, V. Raman, S. R. Satti: Succinct indexable dictionaries with applications to encoding k-ary trees, prefix sums and multisets. ACM Transactions on Algorithms, Volume 3, Number 4,  
非特許文献2 : G. Navarro, E. Providel: Fast, small, simple rank/select on bitmaps, In Proc. 11th SEA, pp. 295{306, 2012.

### 発明の概要

#### 発明が解決しようとする課題

[0008] RRRの実用的な実装のために、ブロック圧縮に対する効率良い符号化法と復号化法の提供を目的とし、高速で省領域な符号化法と復号化法を与え、密なビット列に対しても高速に動作するRRR 実装を提供する。

#### 課題を解決するための手段

[0009] 上記課題を解決するための発明の実施形態の一つに対応するブロック符号化装置は、

1 以上のプロセッサと、

プログラムを記憶した1以上のメモリと

を備え、

前記プログラムは、前記1以上のプロセッサにより実行された場合に、前記1以上のプロセッサに

入力ビット列を第1のブロック長の複数のブロックに分割して処理対象ブロックを生成し、

処理対象ブロックのビット列に含まれる1の値の数から、処理対象ブロックの第1の重みを決定し、

前記処理対象ブロックを第2のブロック長の小ブロックに分割し、小ブロック毎の重みと値とに基づき、前記第1のブロック長と前記第1の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第1の順序を決定し、

前記処理対象ブロックのビット列を、前記決定された第1の重みと第1の順序との値に変換してブロック符号化データを生成する動作を実行させる。

[0010] 上記課題を解決するための発明の実施形態の一つに対応するブロック復号化装置は、

1以上のプロセッサと、

プログラムを記憶した1以上のメモリと、

を備え、

前記プログラムは、前記1以上のプロセッサにより実行された場合に、前記1以上のプロセッサに、第1の重みを有する第1のブロック長の処理対象ブロックのビット列をブロック符号化して生成されたブロック符号化データから前記処理対象ブロックのビット列を復号する動作を実行させ、

前記ブロック符号化データは、前記第1の重みと、前記第1のブロック長と前記第1の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第1の順序とを含み、

前記復号する動作には、

前記処理対象ブロックを第2のブロック長の小ブロックで分割した場合の、前記処理対象ブロックの最上位ビットを含む第1の小ブロックが有する第2の重みを決定し、

前記集合において、ビット列の第1の小ブロックにつき前記第2の重み

よりも小さい重みを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の第1の数を決定し、

前記集合において、前記第1の順序と前記第1の数との差分に基づき、前記処理対象ブロックの前記第1の小ブロックが有するビット列と、前記処理対象ブロックの前記第1の小ブロックと重みが同じで値の異なる第1の小ブロックを有するビット列の第2の数とを決定し、

前記第1の重みと前記第2の重みとの差分と、前記集合において、前記第1の順序と前記第1の数及び前記第2の数との差分とに基づいて、前記処理対象ブロックの第2の小ブロックが有するビット列を決定する動作が含まれる。

## 発明の効果

[0011] 本発明によれば、高速で省領域な符号化法と復号化法を与え、密なビット列に対しても高速に動作するRRR実装を提供することができる。これにより、主記憶内に収容するデータ量を増やして速度低下を防止し、データ処理の高速化を実現することができる。

## 図面の簡単な説明

- [0012] [図1]発明の実施形態に対応する、システム全体の構成例を示す図。  
[図2A]発明の実施形態に対応する、クライアント101のハードウェア構成の一例を示すブロック図。  
[図2B]発明の実施形態に対応する、サーバ103のハードウェア構成の一例を示すブロック図。  
[図3]発明の実施形態に対応する、サーバ103の機能構成の一例を示すブロック図。  
[図4A]発明の実施形態に対応する、圧縮データ構造を適用可能な木構造で表現される情報を説明するための図。  
[図4B]発明の実施形態に対応する、圧縮データ構造を適用可能な木構造で表現される情報を説明するための図。  
[図4C]発明の実施形態に対応する、圧縮データ構造を適用可能な木構造で表

現される情報を説明するための図。

[図5]発明の実施形態に対応する、ブロック符号化処理の一例を示すフローチャート。

[図6]発明の実施形態に対応する、ブロック符号化処理の対象のビット列の配列順序を説明するための図。

[図7]発明の実施形態に対応する、ブロック符号化処理及びブロック復号化処理において参照する各テーブルを説明するための図。

[図8]発明の実施形態に対応する、順序OD (X) の決定処理の一例を示すフローチャート。

[図9]発明の実施形態に対応する、ブロック復号化処理の一例を示すフローチャート。

### 発明を実施するための形態

[0013] 以下、図面を用いて、本願発明に係る実施形態について説明する。なお、以下に示す実施形態は一例であり、これに限定するものではない。そのため、本願発明の特徴的構成を適用できれば他の構成であってもよい。説明する構成要素のうち同一機能を有するものには同じ符号を付し、その説明を省略する。以下では主に、XMLなどの半構造テキストフォーマットのデータとそのインデックスを配信し、受信側でそのデータを処理するシステムを説明する。

[0014] <システム構成>

図1は、本実施形態に対応する情報処理システムの全体構成を示すブロック図である。当該情報処理システムは、ユーザ端末であるクライアント101a、101b、101c（以下、これらを総称して「クライアント101」という。）及びサーバ103がインターネット102に接続され相互通信が可能に構成されている。サーバ103には、データベース104が接続されている。

[0015] クライアント101は、ユーザが操作するユーザ端末、或いは、情報処理装置である。クライアント101は、ユーザ操作に応じてサーバ103が提

供するサイトにアクセス可能である。クライアント101はサイト情報（ウェブページ情報）をサーバ103に要求する（「アクセス要求」ともいう）ことができる。当該要求に応じてサーバ103から送信されたサイト情報を受信すると、クライアント101は受信したサイト情報をディスプレイに表示する。

[0016] クライアント101は、例えば、ノートパソコン、デスクトップパソコン、携帯情報端末、携帯電話、スマートフォン、タブレット端末等を含む。クライアント101には、所謂インターネット・ブラウザのソフトウェア（プラグインでストリーミング・コンテンツを再生可能）がインストールされているものとする。

[0017] クライアント101は、無線LAN、LTE等の無線データ通信手段によってインターネット102に接続されている。尚、イーサネット（登録商標）等のネットワークケーブルを含むLANにより、ネットワーク102へアクセス可能に更に構成されていてもよい。インターネット102は、世界中のネットワークが互いに接続されたネットワークであるが、例えば、イントラネットのように特定の組織内においてのみ接続可能なネットワークであってもよい。

[0018] サーバ103はオンラインショッピングサイトのようなポータルサイトでXMLファイルを提供する情報処理装置であってもよい。クライアント101のユーザは当該ポータルサイトの登録ユーザであってもよい（以下、登録ユーザのことを単にユーザともいう。）。サーバ103は、各種サイト情報が登録されたデータベース104を管理し、クライアント101からのアクセスを受け付ける。

[0019] データベース104は、それぞれ所定のデータベース・ソフトウェアがインストールされた情報処理装置であり、各種データの管理を行う。データベース104は、所定のウェブサイトをインターネット102に公開するための文書ファイルの情報を管理する。文書ファイルには、HTMLファイル、PDFファイルの他、特定アプリケーションで利用されるファイル（例えば

、MSワードなど）が含まれる。データベース104は、随時更新され最新の情報が格納される。

[0020] 尚、本明細書では、便宜上、サーバ103及びデータベース104は、それぞれ物理的に独立した情報処理装置によって実現されるものとして説明するが、本発明の実施形態はこれに限られるものではない。例えば、これらが単一の情報処理装置によって実現されてもよい。その一方で、認証サーバ103等の各装置が複数台の情報処理装置により冗長構成、或いは分散構成されても良い。また、データベース104は、サーバ103とLAN等によって接続されているものとして説明するが、例えば、インターネット102や非図示のイントラネット経由でサーバ103と通信可能な形態をとってもよい。

[0021] <装置のハードウェア構成>

次に、本実施形態に対応する情報処理システムを構成する各装置のハードウェア構成について説明する。図2Aは、クライアント101のハードウェア構成の一例を示すブロック図である。上述した情報処理装置としての、データベース104も、同様或いは同等のハードウェア構成として構成しても良い。

[0022] 図2Bは、発明の実施形態に対応する情報処理装置としてのサーバ103のハードウェア構成の一例を示すブロック図である。CPU、ROM、RAM、外部記憶ドライブ、メディア、HD、指示入力装置、出力装置、I/F、システムバスの機能及び用途、更には、これらの関係は、図2Aを用いて説明するものと同様或いは同等である。図2Bでは、データベース210がシステムバス209Bに接続されていることに留意する。ここで、データベース210は、データベース104に相当する。以下、図2A及び図2Bの各構成要素についてまとめて説明する。

[0023] 図2A及び図2Bにおいて、CPU200A(200B)は、ハードディスク装置(以下、HDと呼ぶ)205A(205B)に格納されているアプリケーションプログラム、オペレーティングシステム(OS)や制御プログ

ラム等を実行し、RAM 202 A (202 B) にプログラムの実行に必要な情報、ファイル等を一時的に格納する制御を行う。CPU 200 A (200 B) は、単一のプロセッサであっても良いし、複数のプロセッサから構成されても良い。サーバ 103 から受信し、RAM 202 A に記憶されたサイト情報 (XML ファイル) を展開し、出力装置 207 A のディスプレイに表示可能な形式のデータに変換する。さらには、インターフェイス 208 A (208 B) を介した外部装置とのデータ送受信を制御するとともに、外部装置から受信したデータの分析・処理を実行し、また外部装置へ送信するデータ (処理要求やデータ要求を含む) の生成を行う。なお、後述する図 3 から図 8 を参照して説明する処理も CPU 200 A (200 B) が対応する処理プログラムを実行することにより装置全体を制御して実現される。

[0024] ROM 201 A (201 B) は、内部に基本 I/O プログラム、種々のアプリケーションプログラム等の各種データを記憶する。RAM 202 A (202 B) は各種データを一時記憶し、CPU 200 A (200 B) の主メモリ、ワークエリア等として機能する。

[0025] 外部記憶ドライブ 203 A (203 B) は、記録媒体へのアクセスを実現するための外部記憶ドライブであり、メディア (記録媒体) 204 A (204 B) に記憶されたプログラム等を本コンピュータシステムにロードすることができる。尚、メディア 204 A (204 B) は、例えば、フロッピー (登録商標) ディスク (FD)、CD-ROM、CD-R、CD-RW、PC カード、DVD、Blu-ray (登録商標)、IC メモリカード、MO、メモリスティック等を利用することができる。

[0026] 外部記憶装置 205 A (205 B) は、本実施形態では大容量メモリとして機能する HD (ハードディスク) を用いている。HD 205 A (205 B) には、アプリケーションプログラム、OS、制御プログラム、関連プログラム等が格納される。なお、ハードディスクの代わりに、フラッシュ (登録商標) メモリ等の不揮発性記憶装置を用いても良い。

[0027] 指示入力装置 206 A (206 B) は、キーボードやポインティングデバ

イス（マウス等）、タッチパネル等がこれに相当する。指示入力装置206A（206B）を用いて、ユーザは、クライアント101に対して、装置を制御するコマンド等を入力指示する。出力装置207A（207B）は、指示入力装置206A（206B）から入力したコマンドや、それに対するクライアント101やサーバ103の応答出力等を出力する。出力装置207A（207B）にはディスプレイ、スピーカ、ヘッドフォン端子等が含まれる。システムバス209A（209B）は、情報処理装置内のデータの流れを司る。

[0028] インターフェイス（以下、I/Fという）208A（208B）は、外部装置とのデータのやり取りを仲介する役割を果たす。例えば、クライアント101にとって、外部装置はサーバ103等であり、サーバ103にとって、外部装置はクライアント101等となる。具体的に、I/F208A（208B）は、無線通信モジュールを含むことができ、当該モジュールはアンテナシステム、RF送受信器、1つ以上の増幅器、同調器、1つ以上の発振器、デジタル信号プロセッサ、CODECチップセット、加入者識別モジュールカード、メモリなどを含む、周知の回路機構を含むことができる。また、有線接続のための有線通信モジュールを含むことができる。有線通信モジュールは1つ以上の外部ポートを介して他のデバイスとの通信を可能とする。また、データを処理する様々なソフトウェアコンポーネントを含むことができる。外部ポートは、イーサネット、USBやIEEE1394等を介して、直接的に、又はネットワークを介して間接的に他のデバイスと結合する。尚、以上の各装置と同等の機能を実現するソフトウェアにより、ハードウェア装置の代替として構成することもできる。

[0029] 本実施形態に対応する処理を実行するために対応するプログラムを動作させる度に、既にプログラムがインストールされているHD205A（205B）からRAM202A（202B）にロードするようにしてもよい。また、本実施形態に係るプログラムをROM201A（201B）に記録しておき、これをメモリマップの一部をなすように構成し、直接CPU200A（

200B)で実行することも可能である。さらに、メディア204A(204B)から対応プログラム及び関連データを直接RAM202A(202B)にロードして実行させることもできる。

[0030] 図3は、本実施形態に係るサーバ103の機能ブロック図である。ここに示す各ブロックは、サーバ103が備えるROM201A、RAM202B、HD205Bに保持された所定のプログラムがサーバ103のCPU200Bにより実行されることで実現される各種の制御及び処理機能を表している。

[0031] ブロック符号化部301は、入力ビット列をブロック長 $n$ のビット列からなるブロックに分割し、各ブロックを重み $w$ と順序 $O$ との値に変換するブロック符号化処理を行う。ブロック符号化データはRAM202Bのワークエリアに格納される。ブロック符号化処理の詳細は図5から図7を参照して後述する。

[0032] ブロック復号化部302は、ブロック符号化部301におけるブロック符号化処理により得られ、RAM202Bのワークエリアに格納されているブロック符号化データを復号し、元のビット列を復元するブロック復号化処理を行う。ブロック復号化処理の詳細は図7から図8を参照して後述する。

[0033] テーブル生成部303は、ブロック符号化部301及びブロック復号化部302において実行されるブロック符号化処理及びブロック復号化処理において参照されるテーブルの情報を生成する。テーブル情報の生成については図5を参照して後述する。

[0034] テーブル保持部304は、ブロック符号化部301及びブロック復号化部302において実行されるブロック符号化処理及びブロック復号化処理において参照されるテーブルの情報を保持する。テーブル情報及びその利用方法の詳細については図5から図8を参照して後述する。

[0035] 次に発明の実施形態に対応する圧縮データ構造を適用可能な木構造で表現される情報について説明する。図4Aから図4CはXMLやJSON等の木構造で表現されるデータの一例を示す図である。図4Aは、半構造テキスト

データの一例を示す図である。図4Aに示す半構造テキストデータは、JSON形式のデータを例示している。半構造テキストデータは、一般的に複数のノードを含む木構造と対応することが知られている。半構造テキストデータでは、複数のノードのそれぞれは半構造テキストデータのうちそれぞれの始点から終点までの文字列により表される。例えば、この始点や終点がシンボル（図4Aの例ではそれぞれ「{」と「}」）により表されてもよいし、ほかの何らかの構文上の規則によって始点と終点が示されていてもよい。また、あるノードの文字列の中に、そのノードの子ノードの文字列が入れ子構造で含まれていてもよい。また子ノードの文字列に孫ノードなどのさらに下位のノードの文字列が含まれていてもよい。図4Aにおける丸付き数字および破線は半構造テキストデータを説明するための符号および線であり、丸付き数字は破線を介して接続される括弧の位置に対応するノードの番号を示している。

[0036] 図4Bは図4Aの半構造テキストデータに対応する木構造を示す図である。番号0のノードは最上位のノードであり、その子ノードは番号1および4のノードである。また番号1のノードの子ノードは番号2であり、番号2の子ノードは番号3のノードである。番号4のノードの子ノードは番号5および6のノードであり、番号6の子ノードは番号7のノードである。

[0037] 図4Cは図4Bに示した木構造を括弧木表現（BP表現）及びビット列として示したテーブルである。このビット列は本実施形態における処理対象のデータであり、ノード位置、ビット位置、BP表現はそのビット列を説明するためのものである。

[0038] 図4Cに示すビット列は、XMLやJSON等の木構造で表現されるテキストデータに基づいて生成することができる。BP表現の行の「（」「）」（以下では単に「括弧」と記載した場合、この括弧木表現の括弧を示す）はそれぞれテキストデータの各ノードの始点と終点を示す。したがって、ビット列の「1」「0」はそれぞれ対応するノードの始点、終点を表す。なお図4Cではビット位置は0から始まる番号としている。一方、本実施形態では

入力ビット列を固定長  $n$  の処理対象ブロックに分割するが、図 4 C では 1 ブロックを 8 ビットとし、ブロック番号は 0 から始まる番号としている。1 ブロックに含まれるビット数は 8 以外の 2 のべき乗の数であってもよいし、それ以外の任意の数であっても良い。以下では、ブロックのビット数のことをブロック長と呼ぶ。

[0039] 図 4 C の B P 表現における開き括弧とその開き括弧に対応する閉じ括弧とはノードを示している。図 4 C のノード位置はそのビット位置を始点とするノードの番号を示し、このノード位置は図 4 B の木構造のノード番号に対応する。ここで、図 4 C に示すビット列では左から右にビット位置が大きくなっているが、RAM や CPU のレジスタに格納される物理的なブロックのデータでは、ビット位置の番号が大きくなるにつれて RAM やレジスタの上位の桁のビットに順に配置されるようにビットが配置されている。図 4 C に示すビット列では、右になるほど RAM やレジスタの上位の桁に格納される。

[0040] <ブロック符号化処理>

以下、本実施形態に対応するバイナリデータのブロック符号化処理について説明する。ブロック符号化処理は、本実施形態においてはブロック長  $n$  の任意のビット列で構成される処理対象ブロック  $X$  を、当該ブロック  $X$  に含まれる重み  $w$  と、処理対象ブロックの順序  $OD$  との値に変換する処理を言う。数式では以下のように表現する。

$$\text{ENCODE}(X) = (w, OD) \quad (\text{式 1})$$

ここで、重みとはビット列内に含まれる「1」の個数をいう。また、順序とは重み  $w$ 、ブロック長  $n$  のビット列の集合 ( $BITS(n, w)$ ) において処理対象ブロック  $X$  が位置する順番を示す値をいう。

[0041] 図 5 は、ブロック符号化処理の一例を示すフローチャートである。当該フローチャートに対応する処理は、ブロック符号化部 301、テーブル生成部 303 及びテーブル保持部 304 により実行される。

[0042] まず、S501 では、テーブル生成部 303 が処理対象ブロックのブロッ

ク長 $n$ 、処理対象ブロックを構成する各小ブロックのブロック長 $k$ 、及び処理対象ブロックの重み $w$ を設定する。本実施形態では、ブロック長 $n$ の処理対象ブロックは、ブロック長 $k$ の小ブロックにより等分される。重み $w$ は、ブロック長 $n$ のバイナリデータの全体についての重みであって、小ブロック毎の重みは限定しない。重み $w$ の範囲は $0 \leq w \leq n$ となる。例えば重み $w$ が2の場合、処理対象ブロック内には「1」が2個含まれることになる。

[0043] 次に、S502では、テーブル生成部303がS501で決定されたブロック長 $n$ 、重み $w$ のバイナリデータにつき、存在し得るビット列の集合 $BITS(n, w)$ を決定する。このとき $BITS(n, w)$ に含まれるビット列の個数を $l$ とする。 $l$ は以下の式により求めることができる。

$$l = C(n, w) \quad (\text{式2})$$

ここで、 $C(n, w)$ は、 $n$ 個の中から $w$ 個を選択する組み合わせ ${}_n C_w$ を表す。

[0044] 続くS503では、テーブル生成部303がS502で決定した $BITS(n, w)$ におけるバイナリデータの順序 $OD(Y_i)$ を決定する。ここで、 $Y_i$ は、 $BITS(n, w)$ に含まれる任意のビット列を示し、 $i$ は0以上 $l$ 未満の整数で表される( $0 \leq i < l$ )。このとき同時に $0 \leq OD(Y_i) < l$ も成立する。次にS504では、テーブル生成部303が、S503で決定された順序の $BITS(n, w)$ ビット列について、ブロック符号化処理において参照すべきテーブルを作成し、テーブル保持部304に保持する。

[0045] 次に、S505では、ブロック符号化部301が入力ビット列をS501で設定されたブロック長 $n$ のブロックに分割し処理対象ブロック $X$ を生成する。続くS506では、ブロック符号化部301が処理対象ブロック $X$ をブロック長 $k$ の小ブロックに分割する。

[0046] 続くS507では、ブロック符号化部301が、処理対象ブロック $X$ のビット列に含まれる「1」の数に基づき重み $w$ を決定する。続くS508では、ブロック符号化部301が、テーブル保持部304が保持するテーブルを参照して処理対象ブロック $X$ の順序 $OD(X)$ を決定する。 $OD(X)$ は、

S 5 0 3で決定されたB I T S ( n , w ) におけるバイナリデータの順序O D ( Y<sub>i</sub> ) のうち、X = Y<sub>i</sub>となる順序O D ( Y<sub>i</sub> ) を特定すればよい。またXの順序O D ( X ) は、O D ( X ) > O D ( Y<sub>i</sub> ) を満たすB I T S ( n , w ) 内のY<sub>i</sub>個数に相当する。S 5 0 9では、ブロック符号化部3 0 1がO D ( X ) と重みwに基づき、処理対象ブロックXのブロック符号化結果 ( E N C O D E ( X ) = ( w , O D ) ) を出力する。

[0047] 上記ステップのうちS 5 0 1からS 5 0 4までの処理は、入力ビット列が入力される以前に想定されるn、k、wの任意の組み合わせについて予め実行されていてもよい。

[0048] 以下、図5のフローチャートに従ったブロック符号化処理の具体例につき、図6及び図7を参照して説明する。図6は、ブロック符号化処理の対象となるビット列の配列順序を説明するための図であり、図7は、ブロック符号化処理及びブロック復号化処理において参照する各テーブルを説明するための図である。

[0049] まず、S 5 0 1における処理につき、以下の具体例ではブロック長n = 6、小ブロックのブロック長k = 3、重みw = 2とする。なお、これらの数値はあくまで例示のために用いるものであって、より大きな値を用いても良い。

[0050] S 5 0 2における処理につき、ブロック長nを6とした場合に存在し得るビット列Y<sub>i</sub>のパターンB I T S ( 6 , 2 ) は図6の左表6 0 1に示すとおりである。6ビットの処理対象ブロックに「1」が2個含まれるようなビット列の個数lは、 $l = C ( 6 , 2 ) = 1 5$ となる。ビット列Y<sub>i</sub>は、Y<sub>0</sub>からY<sub>14</sub>までが存在し、図6の左表6 0 1では「1」が最下位ビット側から存在する順序でデータを昇順で示している。当該順序はビット列Y<sub>i</sub>の大きさ順に相当する。表の左脇に示す0から14までの数字は、表に示す各バイナリデータに対して割り当てられる順序を示している。ここでは、「1」の位置は6ビットのブロック全体について判定されている。

[0051] これに対し、本実施形態では、S 5 0 3における処理にて15個のビット

列 $Y_i$ の並びをブロック長 $k=3$ の小ブロック( $Y_{i-0}$ 、 $Y_{i-1}$ )毎の値に基づいて入れ替え、順序 $OD(Y_i)$ を決定する。図6の右表602はブロックの順序を入れ替えた結果を示している。ここでは0番目から8番目までの順序は不変であるが、9から12番目の順序が入れ替わっている。右表602では処理対象ブロックの順序は、最上位ビット側から小ブロック単位のビットが有する重み及び値に基づき昇順で決定されている。ここでは重み $w$ に基づく順序は $w=0, 1, 2$ の順となる。また、値に基づく順序は例えば重み1ならば「001」、「010」、「100」であり、重み2であれば「011」、「101」、「110」となる。図6の例では、「000」という重み0のグループと、「001」、「010」及び「100」という重み1のグループと、「011」、「101」、「110」という重み2のグループとが生成されるので、これらの順に順序が決定される。

[0052] 図6では、処理対象ブロックのブロック長 $n$ が6ビットの場合を示したが、上述の順序決定方法は、ブロック長 $n$ 、小ブロック長 $k$ 、重み $w$ が変わった場合も同様に適用可能である。例えば、ブロック長 $n$ がより長くなった場合であっても、最上位ビット側に位置する小ブロックから順に重みが小さく、値の小さい順序で昇順に配列される。

[0053] 続くS504では、S503で決定された順序のビット列について、ブロック符号化処理にて参照されるテーブルをテーブル生成部303が生成する。テーブル生成部303は図7に示すような5種類のテーブルを生成することができる。

[0054] 図7に示す第1のテーブル(T1)701は、最上位ビット側の小ブロック $Y_{i-0}$ につき、より小さい重みを有する $Y_i$ の個数をブロック $Y_{i-0}$ の重み $w$ ごとに登録する。まず、重み0よりも小さい重みはないため、T1<sub>0</sub>の先頭のエントリの値は0となる。次に重み1より小さい重み0を有するビット列 $Y_i$ は3個あるため、重み1のためのT1<sub>0</sub>のエントリの値は3となる。同様に重み2より小さい重み0及び重み1を有するビット列 $Y_i$ は12個あるため、重み2のためのT1<sub>0</sub>のエントリの値は12となる。また、図7の例では重み3

は存在しないので値として無効であるが、仮に重み 3 が存在する場合には、対応する値が登録される。

[0055] 図 7 では、ブロック長  $n = 6$ 、小ブロック長  $k = 3$  の場合について記載したが、これよりも長いブロック長について T 1 を生成する場合は、最上位ビット側から  $n/k - 1$  個分の小ブロックのそれぞれについて、より小さい重みを有する  $Y_i$  の個数を重み  $w$  ごとに登録する。各値は以下の式より求めることができる。

$$T1[x][y][z] = \sum_{v \text{ in } [0..z]} C(k, v) \times C(k \times x, y - v), \quad x \text{ in } [0..n/k], \quad y \text{ in } [0..w], \quad z \text{ in } [0..k] \quad (\text{式 3})$$

[0056] 次に、第 2 のテーブル (T 2) 7 0 2 は、重み  $w$  の値に基づく小ブロック (小ブロック長  $k$ ) 内での組み合わせ数を登録する。ここでは、重み  $w = 2$  であるので、小ブロック内に存在し得る「1」の数は、「0」、「1」、「2」のいずれかである。また、小ブロック長  $k = 3$  である。これに基づき T 2 の先頭のエンタリは  $C(3, 0)$  であり、小ブロックを構成する 3 ビットのブロックにおいて「1」が存在しないことを示す。このような組み合わせの数は「000」の 1 通りしか存在しないので、登録される値は「 $C(3, 0) = 1$ 」となる。次に、小ブロックを構成する 3 ビットのブロックにおいて「1」が 1 つ存在する場合、「001」、「010」、「100」の 3 通りがあるので、登録される値は「 $C(3, 1) = 3$ 」となる。同様に、小ブロックを構成する 3 ビットのブロックにおいて「1」が 2 つ存在する場合、「011」、「101」、「110」の 3 通りがあるので「 $C(3, 2) = 3$ 」となる。

[0057] また、発明の実施形態はこの例に限定されることなく、T 2 には、ブロック長  $n$ 、小ブロック長  $k$  及び重み  $w$  の大きさに合わせた値が登録される。各値は以下の式より求めることができる。

$$C(k \times x, y) \quad (x \text{ in } [0..n/k], \quad y \text{ in } [0..w]) \quad (\text{式 4})$$

[0058] 次に第 3 のテーブル (T 3) 7 0 3 は、小ブロック長  $k$  について、重みによって異なるビットパターンに割り当てられる順序の値を登録する。T 3 に

登録される同一重みにおける順序としてビット列 $X$ の順序を $OD_{sw}(X)$ で表す。ここでは $k=3$ の場合を説明するが、例えば、重み0と重み3の場合は「000」、「111」しか存在しないため値を登録する必要はない、或いは $OD_{sw}(000)=0$ 、 $OD_{sw}(111)=0$ を登録しておく。また、重み1及び重み2の場合は3ビットのブロック内で「1」を取り得る位置に応じてビットパターンが異なる。重み1の場合「001」、「010」、「100」の3パターンが存在し、重み2の場合「011」、「101」、「110」の3パターンがやはり存在する。これらはその順序で「0」、「1」、「2」の値が割り当てられる。例えば、 $OD_{sw}(001)=0$ 、 $OD_{sw}(010)=1$ 、 $OD_{sw}(100)=2$ となる。また、 $OD_{sw}(011)=0$ 、 $OD_{sw}(101)=1$ 、 $OD_{sw}(110)=2$ となる。

[0059] この順序 $OD_{sw}(X)$ の値は、当該順序が割り当てられたビット列の小ブロックよりも前に存在し、かつ、重みが同じ小ブロックの数を示している。例えば、「001」に割り当てられる順序は「0」であるが、これは「001」の小ブロックの前には同一重みの小ブロックが存在しないことを意味する。また、「100」に割り当てられる順序は「2」であるが、これは「100」の小ブロックの前には同一重みの小ブロックが2つ（「001」と「0101」のこと）存在することを意味する。また、発明の実施形態はこの例に限定されることなく、T3には、小ブロック長 $k$ 及び重み $w$ の大きさに合わせた値が登録され、同一の重みの場合、順序は値の大きさに基づき昇順で決定される。

[0060] 次に第4のテーブルT4（704）は、小ブロック長 $k$ のビット列につき、同一重み内で与えられる順位と重みの値との組み合わせに対応するビット列を登録する。図7では $k=3$ のビット列 $X$ についての組み合わせに対応するビット列と、その登録順序が関連づけて登録されており、本実施形態ではこのビット列を $BIT_{s_{k=3}}(w, OD_{sw})$ で表すこととする。この場合、重み $w$ は0から3の値を取ることができ、存在し得るビット列のパターンは8通りである。各ビット列は、重みの値（0から3）と、同一重み内で割り当

てられた順序  $OD_{s_w}$  (0から2) とで一意に特定可能となっている。例えば、重み0の場合、 $BIT_{s_{k=3}}(0, 0) = 000$ となる。また、重み1の場合、取り得るビット列は「001」、「010」、「100」のいずれかとなるが、このうちから1つのビット列を順序の値に基づいて特定する。例えば順序  $OD_{s_w} = 1$  の場合、 $BIT_{s_{k=3}}(1, 1) = 010$ となる。ここで同一重みごとにビット列Xに与えられる順序  $OD_{s_w}$  は、テーブルT3と同様にビット列の値の大きさの順になる。

[0061] また、図7には示していないが、重みwの値に基づき同一重みで順序  $OD_{s_w}$  が先頭のビット列をテーブルT4内で特定するためのテーブルT4における登録順序を示す値が、テーブルT4に予め登録されているか、又はテーブル保持部304により保持されていても良い。当該値は、 $BIT_{s_{k=3}}(w, OD_{s_w})$  につき  $OD_w = \Sigma(3, w-1)$  で求めることができる(但し、 $w=0$  の場合は0)。例えば、 $k=3$  の場合、取り得る重みは0から3であるので、重み0の場合は、 $OD_{w=0} = 0$ 、重み1の場合  $OD_{w=1} = C(3, 0) = 1$ 、重み2の場合  $OD_{w=2} = C(3, 0) + C(3, 1) = 4$ 、重み3の場合  $OD_{w=3} = C(3, 0) + C(3, 1) + C(3, 2) = 7$  となる。このうち、重みwが2の場合は  $OD_{w=2} = 4$  だから、テーブルT4の登録順序が「4」の位置から重み2に対応するビット列が登録されていることが分かる。図7では、登録順序が4から6の「011」から「110」までが重み2に対応する。そして、当該同一重みにおける先頭エントリ「011」から順序  $OD_{s_w}$  の値に従ってビット列を特定することができる。例えば、順序  $OD_{s_w} = 2$  であれば、先頭から2番目のエントリ(110)が特定すべきビット列となる。

[0062] 以上をまとめると、例えば  $(w, OD_{s_w}) = (2, 1)$  の場合、テーブルT4の登録順序を  $OD_{w=2} + OD_{s_w} = 4 + 1 = 5$  と求めることで、 $BIT_{s_{k=3}}(2, 1) = 101$  を選択することができる。

[0063] 次に、第5のテーブルT5(705)は、小ブロック長kのビット列が有する重みの値を登録する。図7では  $k=3$  のビット列Xについて示しており、本実施形態では順序を  $W_{k=3}(X)$  で表すとする。例えば、重み0のビット列

「000」は $W_{k=3}(000) = 0$ であり、重み2のビット列「101」は $W_{k=3}(101) = 2$ となる。このようにして生成したテーブルT1からT5は、テーブル保持部304が保持する。

[0064] 次に、ブロック符号化部301が処理対象ブロックXのBITS(6, 2)における順序OD(X)を決定するS508における処理を、まず概念的に説明する。具体的に、ブロック符号化部301は、BITS(6, 2)においてS503で決定した順序で処理対象ブロックXよりも前に位置するビット列 $Y_i$ の個数を計算する。このとき、処理対象ブロックXを構成する小ブロックを $X_0$ 及び $X_1$ で表し、BITS(6, 2)に含まれる処理対象ブロックX以外の任意のビット列 $Y_i$ の小ブロックを $Y_{i-0}$ と $Y_{i-1}$ とで表すこととする。

[0065] まず、小ブロック $X_0$ で $OD(X_0) > OD(Y_{i-0})$ を満たす $Y_i$ の個数を計算する。例えば図6右表602に示されるビット列のうち、処理対象ブロックXを順序「12」の「011000」とすると、 $X_0$ は「011」であり、図6右表の例で $OD(X_0) > OD(Y_{i-0})$ を満たす $Y_i$ の個数は12個になる。そして、 $X_0 = Y_{i-0}$ を満たす他の $Y_i$ は存在しないため「12」がそのまま処理対象ブロックXの順序となる。

[0066] また、処理対象ブロックXを順序「10」の「100010」とすると、 $X_0$ は「100」であり、図6右表602の例で $OD(X_0) > OD(Y_{i-0})$ を満たす $Y_i$ の個数は9個になる。そして、 $X_0 = Y_{i-0}$ を満たす $Y_0$ は3つ存在し、そのうち $OD(X_1) > OD(Y_{i-1})$ を満たす $Y_i$ の個数を求めると $Y_i$ の個数は1であるから、 $9 + 1$ により、BITS(6, 2)において処理対象ブロックXよりも小さいビット列 $Y_i$ の個数は10となる。よって、ブロック符号化部301は処理対象ブロックXの順序OD(X)を「10」と求めることができる。

[0067] このように、本実施形態における処理対象ブロックXの順序OD(X)は、最上位ビット側から小ブロック毎に重み、及び値の順で決定される。次に、上記の概念に基づいて実行される、図7の5つのテーブルT1からT5のいずれかを参照したS508における順序OD(X)の決定処理の一例を図

8のフローチャートを参照して更に説明する。

[0068] 以下では、順序「10」が割り当てられた $Y_{10} = 100010$ を処理対象ブロックXとして処理する場合を説明する。まず、S801では、ブロック符号化部301は処理対象ブロックXの各小ブロックの位置を表すパラメータ $h$ を0に初期化する。 $h$ は、 $0 \leq h < n/k - 1$ の範囲を取る整数値である。図6のビット列602の例では、 $h$ は0、1の値をとる。また、S507で決定した処理対象ブロックXの重み $w$ の値で、パラメータ $w'$ を初期化する。図6の例では、重み $w = 2$ であるから、パラメータ $w'$ は2に初期化される。

[0069] 次にS802にて、ブロック符号化部301は小ブロック $X_h$ の値から重み $w_h$ を算出する。ここで $X_0 = Y_{10-0}$ は「100」であるから重み $w_h = 1$ となる。このとき図7のテーブルT5を参照しても良い。続くS803では、重み $w_h$ が0であるか否かを判定する。もし、重み $w_h$ が0の場合（S803で「YES」）、S804に移行して $h$ を1つ更新して次の小ブロックについてS802の処理を行う。一方、重み $w_h$ が0でなかった場合（S803で「NO」）、S805に移行する。図6の例では、 $w_h = 1$ で、0ではないからS805に移行する。

[0070] S805では、パラメータ $w'$ の値を $w' - w_h$ の値で更新する。これにより処理対象ブロックXに対して割り当てられた重みの値が、各小ブロックが有する重みにより減算されていく。図6の例では、 $2 - 1 = 1$ で $w'$ の値が1に更新される。続くS806では、 $w'$ が0と一致するか否かを判定する。もし、 $w'$ が0と一致する場合（S806で「YES」）、S814に移行する。S814では、テーブルT3より重み $w_h$ と同一重みを有する値の内で $X_h$ の順序 $OD_h(X_h)$ を決定し、S815に移行する。一方、 $w'$ が0と一致しない場合（S806で「NO」）、S807に移行する。図6では、 $w' = 1$ で0ではないから、S807に移行する。

[0071] S807では、ブロック符号化部301が、小ブロック $X_h$ が有する重み $w_h$ よりも小さい重みの小ブロック $Y_{i-h}$ を有する $Y_i$ の数( $N_h$ )を、テーブル

$T_{1_h}$ から取得する。ここでテーブル $T_{1_h}$ は、小ブロックの位置 $h$ 毎に用意されているものとし ( $0 \leq h < n/k - 1$ )、図7のテーブル701は $h=0$ の場合を示している。但し、図7では添え字 $h$ を省略している。図6のビット列602の例では、 $X_0 = Y_{10-0}$ は重み $w_h = 1$ であり、重み1よりも小さい重み、即ち重み0を有するビット列 $Y_i$ の個数として、 $h=0$ のテーブル $T_{1_0}$ から「3」を得る。続くS808では、小ブロック $X_h$ の値と、重み $w_h$ とに基づき、テーブルT3より、同一重み内での $X_h$ の順序 $OD_{sw}(X_h)$ を決定する。図6の例では、「100」の順序 $OD_{sw}(100) = 2$ と分かる。その後S809にて、重みが $w_h$ で処理対象ブロックの順序 $OD(X)$ よりも小さい順序 $OD(Y_i)$ である $Y_i$ の数 ( $N_h'$ ) を以下の式で算出する。

$$N_h' = OD_{sw}(X_h) \times C(k(n/k - h - 1), w') \quad (\text{式5})$$

[0072] 図6の例では、 $OD_{sw}(100) = 2$ 、 $k(n/k - h - 1) = 3$  ( $6/3 - 0 - 1$ ) = 3、 $w' = 1$ であるから、 $N_h' = 2 \times C(3, 1) = 2 \times 3 = 6$ となる。なお、 $C(k(n/k - h - 1), w')$ の値は、テーブルT2に予め登録されている値を用いることができる。

この計算により図6の例では、 $Y_{10-0} = 「100」$ より前に存在する重み1で $Y_{i-0}$ が「001」、「010」であるビット列 $Y_i$ の個数が判明した。

[0073] 次に、S810で小ブロック $X_h$ の順序 $OD_h(X_h)$ を以下の式に基づき算出する。

$$OD_h(X_h) = N_h + N_h' \quad (\text{式6})$$

図6の例では、 $N_h = 3$ 、 $N_h' = 6$ であったから、 $OD_0(X_0) = 9$ と求めることができる。続くS811では $h$ を1つ更新し、次のS812にて次の小ブロックが処理対象ブロックにおける未処理の最後の小ブロックであるか否かを判定する。最後の小ブロックで無かった場合 (S812で「NO」)、S802に戻って以上の処理を繰り返す。最後の小ブロックであった場合 (S812で「YES」)、S813に移行する。図6の例では、 $h=1$ となり、最後の小ブロックとなるので、S813に移行する。

[0074] S813では、S805で計算された最新の重み $w'$ の値に基づき、テーブ

ルT3より、同一重み内で $X_h$ の順序 $OD_h(X_h)$ を決定しS815に移行する。図6の例では、ビット列「010」の小ブロック $X_1$ の順序 $OD_1(010)$ は、テーブルT3より「1」と特定される。次に、S815では、S810と、S813又はS814とで各小ブロック $X_h$ について計算された順序 $OD_h(X_h)$ を合計して、処理対象ブロック $X$ の順序 $OD(X)$ を算出する。図6の例では、 $OD_0(X_0) + OD_1(X_1) = 9 + 1 = 10$ であるから、処理対象ブロック $X(100010)$ の順序 $OD(X) = 10$ と求めることができる。ブロック符号化の場合にはこの順序の値の他に重み $w$ の値を付随する。上述の例では $w = 2$ である。よって、S509にてブロック符号化部301は、 $ENCODE(X) = (2, 10)$ とブロック符号化を行うことができる。

[0075] なお、ブロック長 $n$ 及び小ブロック長 $k$ は、符号側と復号側との間において既知の値であれば符号化する必要がないが、既知でない場合はそれらの値も同時に符号化される。本実施形態では $n$ 、 $k$ の値が既知であるとして説明している。

[0076] <ブロック復号化処理>

以下、本実施形態に対応するブロック復号化部302による、ブロック符号化データを復号するブロック復号化処理について説明する。ブロック復号化処理は、本実施形態においてはブロック符号化結果の重み $w$ と処理対象ブロック $X$ の順序 $OD$ との値から、元のブロック長 $n$ のビット列で構成される処理対象ブロック $X$ を復号する処理をいう。数式では以下のように表現する。

$$DECODE(w, OD) = X \quad (\text{式7})$$

[0077] 以下、本実施形態におけるブロック復号化処理の具体例を、図6及び図7を参照しながら図9のフローチャートに基づきより詳細に説明する。当該フローチャートに対応する処理は、ブロック復号化部302、テーブル生成部303及びテーブル保持部304により実行される。

[0078] まず、S901にて、ブロック復号化部302は小ブロックの位置を表す

パラメータ  $h$  を 0 に初期化する。  $h$  は、  $0 \leq h < n/k - 1$  の範囲を取る整数値である。

[0079] 次に S 9 0 2 にて、ブロック復号化部 3 0 2 は小ブロック  $X_h$  の重み  $w_h$  を算出する。ここでの算出式は以下の通りである。

$$w_h = \text{MAX}\{v \text{ in } [0..k] \mid T1[n/k-h-1][w][v] \leq OD\} \quad (\text{式 } 8)$$

[0080] より具体的に、例えば  $h$  を初期化した直後 ( $h = 0$ ) であれば、ここでは復号対象の先頭の小ブロック  $X_0$  の重み  $w_0$  を算出する。図 6 のようにブロック長が 6 ビット、小ブロック長が 3 ビットの場合、小ブロックは  $X_0$  と  $X_1$  の 2 つがあるが、まず、 $X_0$  の重み  $w_0 = \text{weight}(X_0)$  を計算する。その場合 (式 8) の計算式は以下のように示される。

$$w_0 = \text{MAX}\{j \text{ in } [0..k] \mid T1[j] \leq 10\} = 1 \quad (\text{k は小ブロックのブロック長を表す}) \quad (\text{式 } 8-1)$$

[0081] 図 7 の例では、テーブル  $T1$  に登録されている重みのうち、重み 0 の場合  $T1[0] = 0$ 、重み 1 の場合  $T1[1] = 3$ 、重み 2 の場合  $T1[2] = 12$  で、順序  $OD(X) = 10$  以下の値を有するのは  $T1[1]$  であるから、重みは 1 と決定される。

[0082] また、発明の実施形態はこの例に限定されることなく、重みの算出処理は、ブロック長  $n$ 、小ブロック長  $k$  及び重み  $w$  の大きさに応じて実行することができる。

[0083] 次に、S 9 0 3 では、算出した重み  $w_h$  が 0 であるか否かを判定する。重みが 0 の場合 (S 9 0 3 で「YES」) には次の小ブロックを処理対象とすべく S 9 0 4 にて  $h$  の値を 1 つ更新して S 9 0 2 に戻って、再度重みの計算を行う。重みが 0 でない場合 (S 9 0 3 で「NO」)、S 9 0 5 に移行する。

[0084] S 9 0 5 では小ブロック  $X_h$  の順序  $OD_h = OD(X_h)$  を計算する。ここでの算出式は以下の通りである。

$$OD_h = (OD - T1[n/k-h-1][w][w_h]) / T2[n/k-h-1][w-w_h] \quad (\text{式 } 9)$$

[0085] より具体的に、ここでは小ブロック  $X_0$  は重み 1 であるから、取り得る値は

「001」、「010」、「100」のいずれかであり、このいずれであるかを順序OD(X)に基づき決定する。小ブロックX<sub>0</sub>の順序OD<sub>0</sub>は、復号対象のブロックが有する順序OD(X) = 10から重み0のビット列の数(C(3, 1) = 3)を除いた数に含まれる、重み1のビットパターンの組み合わせ数(C(3, 1) = 3)で求められる。このとき計算結果の小数点以下は切り捨てる。その場合、式9は以下のように示される。

$$OD_0 = (OD(X) - T1[w_0]) / C(3, 1) = (10 - 3) / 3 = 2$$

(式9-1)

[0086] 次に、S906では、S902及びS906で算出された重みw<sub>h</sub>、OD<sub>h</sub>に基づき、小ブロックX<sub>h</sub>のビット列を決定する。S905の計算から、小ブロックX<sub>0</sub>は、重さ1のビット列において順序2のパターンであることが分かっている。よって、図7のテーブルT4との関連で(w, OD<sub>s\_w</sub>) = (1, 2)につき、テーブルT4の登録順序はOD<sub>w=1</sub> + OD<sub>s\_w</sub> = 1 + 2 = 3と分かるから、テーブルT4からBIT<sub>s<sub>k=3</sub></sub>(1, 2) = 100が選択され、小ブロックX<sub>0</sub>のビットパターンを100と決定することができる。

[0087] 次に、S907にて、次の小ブロックを処理対象とすべくhの値を1つ更新し、S908でhがn/k-1と一致したかどうか、即ち、最下位ビットを含む小ブロックが処理対象となったか否かを判定する。最下位ビットを含む小ブロックが処理対象となった場合(S908で「YES」)はS909に移行し、それ以外の場合(S908で「NO」)はS902に戻って処理を継続する。

[0088] S909では、それまでの小ブロックに割り当てられた重みをブロック符号化データに含まれるブロックX全体の重みwから減算して重みw<sub>h</sub>(h = n/k - 1)を求める。また、それまでの小ブロックに割り当てられた順序をブロック符号化データに含まれるブロックX全体の順序OD(X)から減算して順序OD<sub>h</sub>(h = n/k - 1)を求める。

[0089] 例えば図6のビット列602の例では、小ブロックX<sub>1</sub>の重みw<sub>1</sub>を算出する。重みw<sub>1</sub>は、w<sub>1</sub> = w - w<sub>0</sub> = 1で算出することができる。また、小ブロッ

ク $X_1$ の順位 $OD_1=OD(X_1)$ を計算する。小ブロック $X_1$ も重み1であるから、取り得る値は「001」、「010」、「100」のいずれかであり、このいずれであるかを順序 $OD(X)$ に基づき決定する。小ブロック $X_1$ の順序 $OD_1$ は、復号対象のブロックが有する順位10から重み0のビット列の数(3)と、順序2よりも前のビットパターン「001」、「010」を有するビット列の数( $OD_0 \times C(3,1)$ )とを引いた数で求められる。これを式で表すと以下のようなになる。

$$OD_1 = (OD - T1[w_0] - OD_0 \times C(3,1)) = 10 - 3 - 2 \times 3 = 1$$

(式10)

[0090] よって、重さ1のビットパターンにおいて順序1のパターンであるから、 $(w, OD_{sw}) = (1, 1)$ につき、テーブルT4の登録順序は $OD_{w=1} + OD_{sw} = 1 + 1 = 2$ と分かるから、テーブルT4から $BIT_{k=3}(1, 1) = 010$ が選択され、 $X_1$ のビットパターンを010と決定することができる。

[0091] 以上により、復号結果として得られるブロック $X = 「100010」$ と決定することができる。

[0092] 以上により、所定のブロック長 $n$ 、小ブロック長 $k$ 、重さ $w$ の処理対象ブロックについて、ブロック符号化及びブロック復号化を行うことができる。このような演算は $k = \lg(n)$ と設定した場合、 $n^3 + o(n^3)$ ビット領域の表を用いて、符号化： $O(n/\lg(n))$ 時間、復号化： $O(n \lg \lg(n)/\lg(n))$ 時間で計算が可能である。具体的に、ブロック長 $n=64$ 、 $k$ を8の場合と16の場合とで長さ $2^{28}$ のランダムなビット列につき、1の個数を $0.5 \times 2^{28}$ とした密な入力ビット列と、1の個数を $0.01 \times 2^{28}$ とした疎な入力ビット列とについて処理した場合、小ブロックに分割しない既存の手法に比べてRANK/SELECT操作について20%程度高速化することができる。これにより、密な入力ビット列に対するクエリ性能の高速化が可能であり、より安定したクエリ性能を提供することができる。

[0093] [その他の実施形態]

本発明は上記実施の形態に制限されるものではなく、本発明の精神及び範

困から離脱することなく、様々な変更及び変形が可能である。従って、本発明の範囲を公にするために、以下の請求項を添付する。また本発明に係る情報処理装置は、1以上のコンピュータを該情報処理装置として機能させるプログラムによっても実現可能である。該プログラムは、コンピュータが読み取り可能な（非一時的）記録媒体に記録されることにより、あるいは電気通信回線を通じて、提供／配布することができる。

## 請求の範囲

### [請求項1]

1以上のプロセッサと、  
プログラムを記憶した1以上のメモリと  
を備えるブロック符号化装置であって、  
前記プログラムは、前記1以上のプロセッサにより実行された場合に、前記1以上のプロセッサに  
入力ビット列を第1のブロック長の複数のブロックに分割して処理対象ブロックを生成し、  
処理対象ブロックのビット列に含まれる1の値の数から、処理対象ブロックの第1の重みを決定し、  
前記処理対象ブロックを第2のブロック長の小ブロックに分割し、小ブロック毎の重みと値とに基づき、前記第1のブロック長と前記第1の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第1の順序を決定し、  
前記処理対象ブロックのビット列を、前記決定された第1の重みと第1の順序との値に変換してブロック符号化データを生成する動作を実行させることを特徴とするブロック符号化装置。

### [請求項2]

前記所定の順序は、前記集合に含まれる前記所定のビット列を前記第2のブロック長の小ブロックに分割した場合に、最上位ビットを含む第1の小ブロック側から順に小ブロック毎の重み及び値がより小さいビット列がより先頭側に位置するように決定されていることを特徴とする請求項1に記載のブロック符号化装置。

### [請求項3]

前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記第1の順序を決定させる場合に、  
前記第1のブロック長、前記第1の重みを有するビット列が前記第2のブロック長の小ブロック内で取り得る重みの大きさごとに、前記集合において、ビット列の前記第1の小ブロックについてより小さい重みを有するビット列の数を登録する第1のテーブルを参照させ、

前記集合において、ビット列の前記第1の小ブロックにつき前記処理対象ブロックの前記第1の小ブロックのビット列が有する重みよりも小さい重みを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の第1の数を算出させる動作を実行させることを特徴とする請求項2に記載のブロック符号化装置。

[請求項4] 前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記第1の順序を決定させる場合に、

前記第2のブロック長の小ブロックが取り得る重みの大きさに前記小ブロックが取り得るビット列のパターン数を登録する第2のテーブルと、前記第2のブロック長の小ブロックが取り得る重みの大きさに前記小ブロックが取り得るビット列のパターン同士の間で定められる順序を登録する第3のテーブルとを参照させ、

前記集合において、前記処理対象ブロックの前記第1の小ブロックのビット列と重みが同一で値の異なるビット列の第1の小ブロックを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の第2の数を算出させる動作を実行させることを特徴とする請求項3に記載のブロック符号化装置。

[請求項5] 前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記第1の順序を決定させる場合に、

前記第3のテーブルを参照させ、前記集合において、前記処理対象ブロックの前記第1の小ブロックと同一の値のビット列の第1の小ブロックを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の第3の数を算出させ、

前記第1の数、前記第2の数、前記第3の数を加算して前記第1の順序を決定する

動作を実行させることを特徴とする請求項4に記載のブロック符号化

装置。

[請求項6]

1以上のプロセッサと、

プログラムを記憶した1以上のメモリと、

を備えるブロック復号化装置であって、

前記プログラムは、前記1以上のプロセッサにより実行された場合に、前記1以上のプロセッサに、第1の重みを有する第1のブロック長の処理対象ブロックのビット列をブロック符号化して生成されたブロック符号化データから前記処理対象ブロックのビット列を復号する動作を実行させ、

前記ブロック符号化データは、前記第1の重みと、前記第1のブロック長と前記第1の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第1の順序とを含み、

前記復号する動作には、

前記処理対象ブロックを第2のブロック長の小ブロックで分割した場合の、前記処理対象ブロックの最上位ビットを含む第1の小ブロックが有する第2の重みを決定し、

前記集合において、ビット列の第1の小ブロックにつき前記第2の重みよりも小さい重みを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の第1の数を決定し、

前記集合において、前記第1の順序と前記第1の数との差分に基づき、前記処理対象ブロックの前記第1の小ブロックが有するビット列と、前記処理対象ブロックの前記第1の小ブロックと重みが同じで値の異なる第1の小ブロックを有するビット列の第2の数とを決定し、

前記第1の重みと前記第2の重みとの差分と、前記集合において、前記第1の順序と前記第1の数及び前記第2の数との差分とに基づいて、前記処理対象ブロックの第2の小ブロックが有するビット列を決定する

動作が含まれる、ことを特徴とするブロック復号化装置。

[請求項7] 前記所定の順序は、前記集合に含まれる前記所定のビット列を前記第2のブロック長の小ブロックに分割した場合に、最上位ビットを含む第1の小ブロック側から順に小ブロック毎の重み及び値がより小さいビット列がより先頭側に位置するように決定されていることを特徴とする請求項6に記載のブロック復号化装置。

[請求項8] 前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記処理対象ブロックの前記第1の小ブロックの第2の重みを決定させる場合に、

前記第1のブロック長、前記第1の重みのビット列が前記第2のブロック長の小ブロック内で取り得る重みの大きさごとに、前記集合において、ビット列の前記第1の小ブロックについてより小さい重みを有するビット列の数を登録する第1のテーブルを参照させ、

前記第1のテーブルに登録されている重みのうち、前記第1の重みよりも値が小さく、かつ、該登録されている重みと関連づけられているビット列の数が前記第1の順序の数を超えない重みを選択し、

前記1以上のプロセッサに前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の前記第1の数を決定させる場合に、

前記第1のテーブルにおいて前記第2の重みと関連づけられているビット列の数を選擇する

動作を実行させることを特徴とする請求項7に記載のブロック復号化装置。

[請求項9] 前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記処理対象ブロックの前記第1の小ブロックが有するビット列を決定させる場合に、

前記第2のブロック長の小ブロックが取り得る重みの大きさごとに前記小ブロックが取り得るビット列のパターン数を登録する第2の

テーブルと、前記第2のブロック長の小ブロックが取り得る重みの大きさと、前記小ブロックが取り得るビット列のパターン同士の間で定められる順序との組み合わせに対応するビット列を登録する第3のテーブルとを参照させ、

前記第1の順序と前記第1の数との差分と、前記第2のテーブルにおいて前記第2の重みと関連づけて登録されたパターン数とに基づき、前記第3のテーブルにおいてビット列を選択するための第2の順序を算出し、前記第3のテーブルにおいて前記第2の重みと前記第2の順序との組み合わせに対応するビット列を前記第1の小ブロックが有するビット列に決定する

動作を実行させることを特徴とする請求項8に記載のブロック復号化装置。

[請求項10]

前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記第2の数を決定させる場合に、

前記第1の順序と前記第1の数との差分と、前記第2のテーブルにおいて前記第2の重みと関連づけて登録されたパターン数とに基づき、前記集合において、前記処理対象ブロックの前記第1の小ブロックと重みが同じで値の異なる第1の小ブロックを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の数を算出させる

動作を実行させることを特徴とする請求項9に記載のブロック復号化装置。

[請求項11]

前記プログラムは、前記1以上のプロセッサにより実行され、前記1以上のプロセッサに前記処理対象ブロックの第2の小ブロックが有するビット列を決定させる場合に、

前記第1の順序と前記第1の数及び前記第2の数との差分に基づき、前記第3のテーブルにおいてビット列を選択するための第3の順序を算出し、前記第3のテーブルにおいて前記第1の重みと前記第2

の重みとの差分が表す重みと、前記第3の順序との組み合わせに対応するビット列を前記第2の小ブロックが有するビット列に決定する動作を実行させることを特徴とする請求項10に記載のブロック復号化装置。

[請求項12] 請求項1乃至5のいずれか1項に記載のブロック符号化装置と、請求項6乃至11のいずれか1項に記載のブロック復号化装置とを備えることを特徴とする情報処理装置。

[請求項13] コンピュータに、請求項1乃至5のいずれか1項に記載のブロック符号化装置の1以上のプロセッサの動作を実行させるためのプログラム。

[請求項14] コンピュータに、請求項6乃至11のいずれか1項に記載のブロック復号化装置の1以上のプロセッサの動作を実行させるためのプログラム。

[請求項15] 1以上のプロセッサと、プログラムを記憶した1以上のメモリとを備えるブロック符号化装置が、前記1以上のプロセッサが前記プログラムを実行することにより実行するブロック符号化方法であって、

入力ビット列を第1のブロック長の複数のブロックに分割して処理対象ブロックを生成することと、

処理対象ブロックのビット列に含まれる1の値の数から、処理対象ブロックの第1の重みを決定することと、

前記処理対象ブロックを第2のブロック長の小ブロックに分割し、小ブロック毎の重みと値とに基づき、前記第1のブロック長と前記第1の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第1の順序を決定することと、

前記処理対象ブロックのビット列を、前記決定された第1の重みと第1の順序との値に変換してブロック符号化データを生成することと

を含む、ことを特徴とするブロック符号化方法。

## [請求項16]

1以上のプロセッサと、プログラムを記憶した1以上のメモリとを備えるブロック復号化装置が、前記1以上のプロセッサが前記プログラムを実行することにより、第1の重みを有する第1のブロック長の処理対象ブロックのビット列をブロック符号化して生成されたブロック符号化データから前記処理対象ブロックのビット列を復号するブロック復号化方法であって、

前記ブロック符号化データは、前記第1の重みと、前記第1のブロック長と前記第1の重みとを有する所定のビット列を所定の順序で配列した集合における前記処理対象ブロックの第1の順序とを含み、

前記方法は、

前記処理対象ブロックを第2のブロック長の小ブロックで分割した場合の、前記処理対象ブロックの最上位ビットを含む第1の小ブロックが有する第2の重みを決定することと、

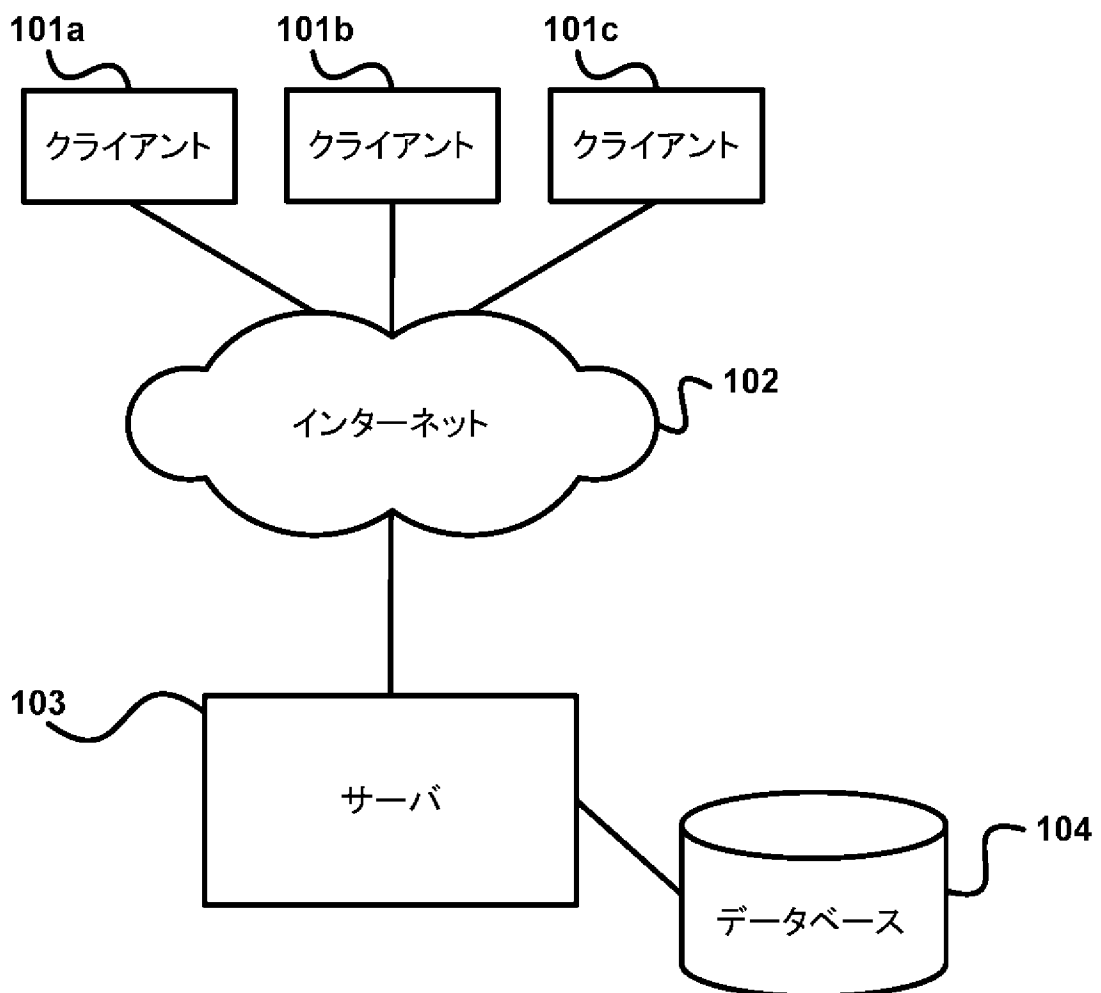
前記集合において、ビット列の第1の小ブロックにつき前記第2の重みよりも小さい重みを有し、かつ、前記所定の順序において前記処理対象ブロックよりも前に位置するビット列の第1の数を決定することと、

前記集合において、前記第1の順序と前記第1の数との差分に基づき、前記処理対象ブロックの前記第1の小ブロックが有するビット列と、前記処理対象ブロックの前記第1の小ブロックと重みが同じで値の異なる第1の小ブロックを有するビット列の第2の数とを決定することと、

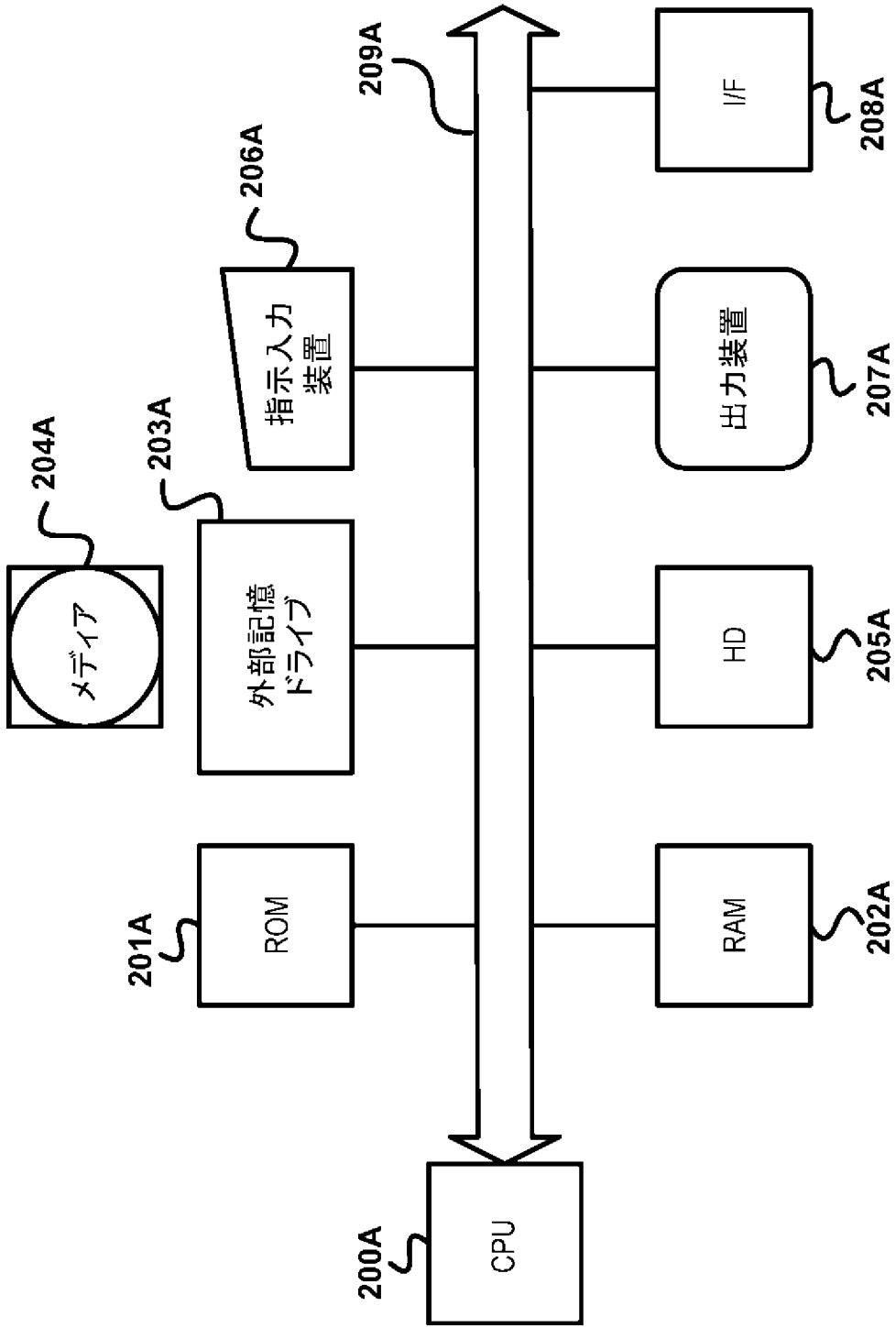
前記第1の重みと前記第2の重みとの差分と、前記集合において、前記第1の順序と前記第1の数及び前記第2の数との差分とに基づいて、前記処理対象ブロックの第2の小ブロックが有するビット列を決定することと

を含む、ことを特徴とするブロック復号化方法。

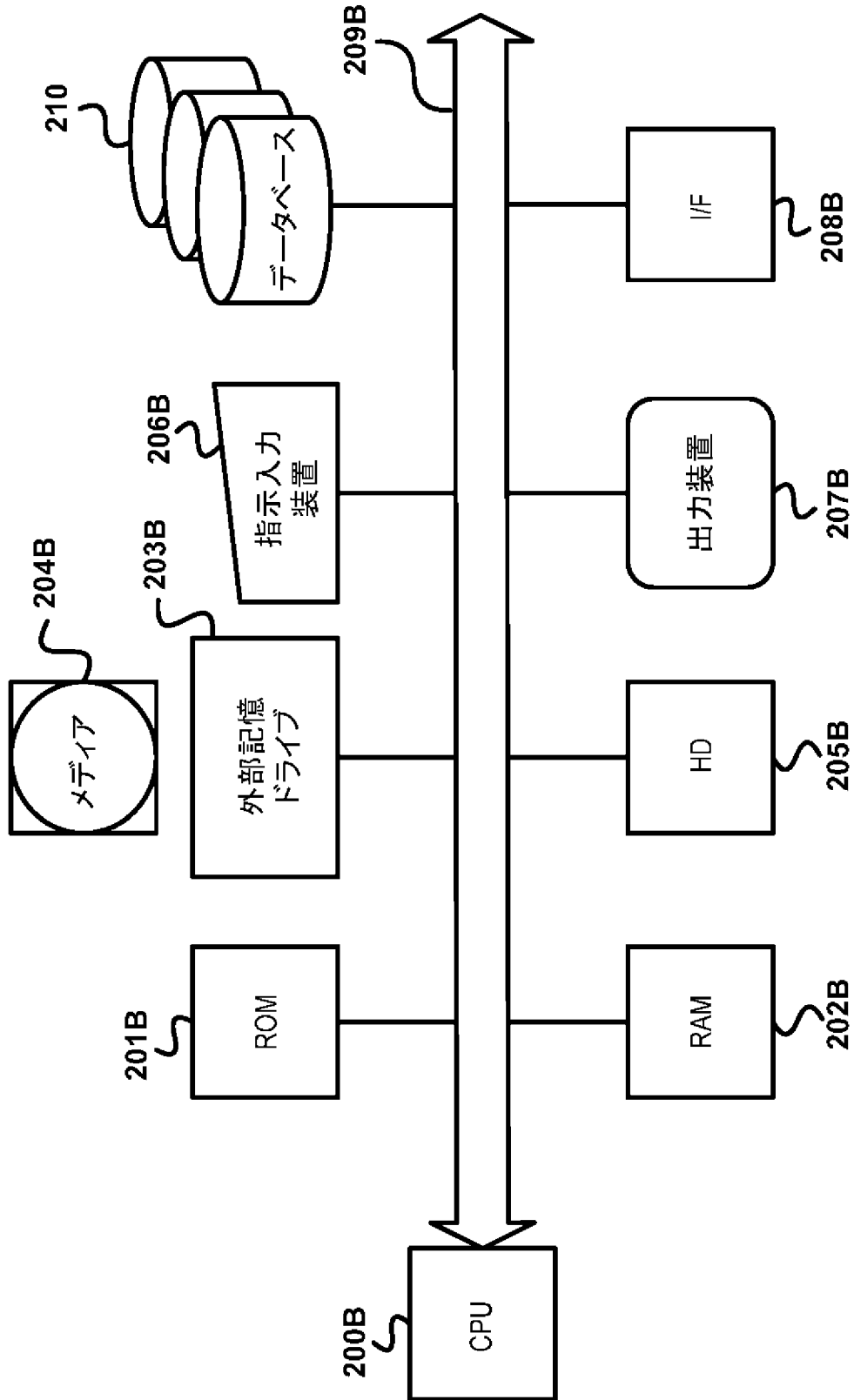
[図1]



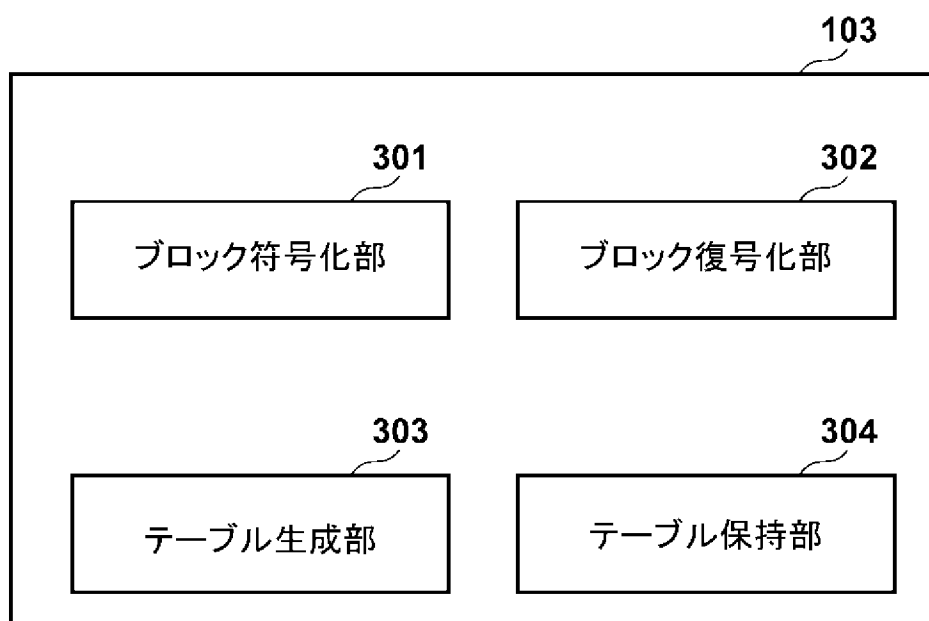
[図2A]



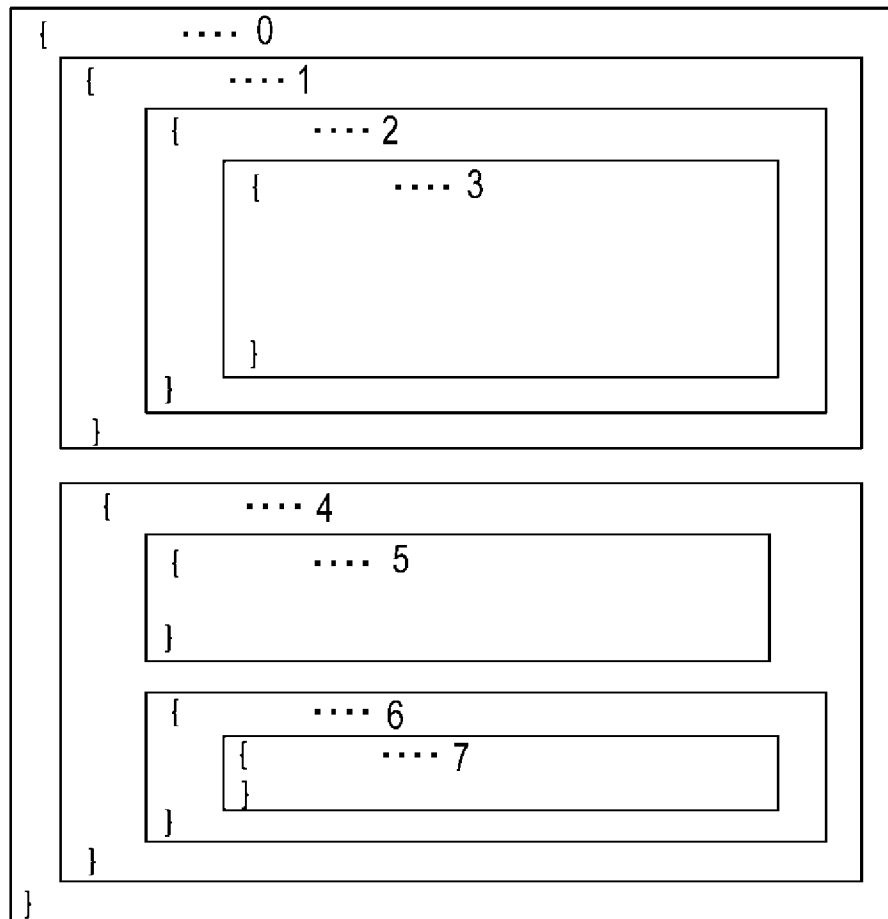
[図2B]



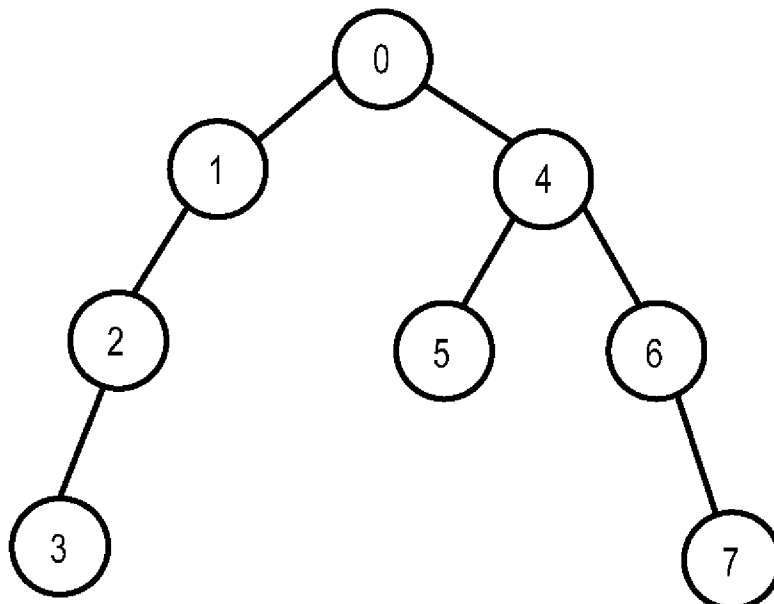
[図3]



[図4A]



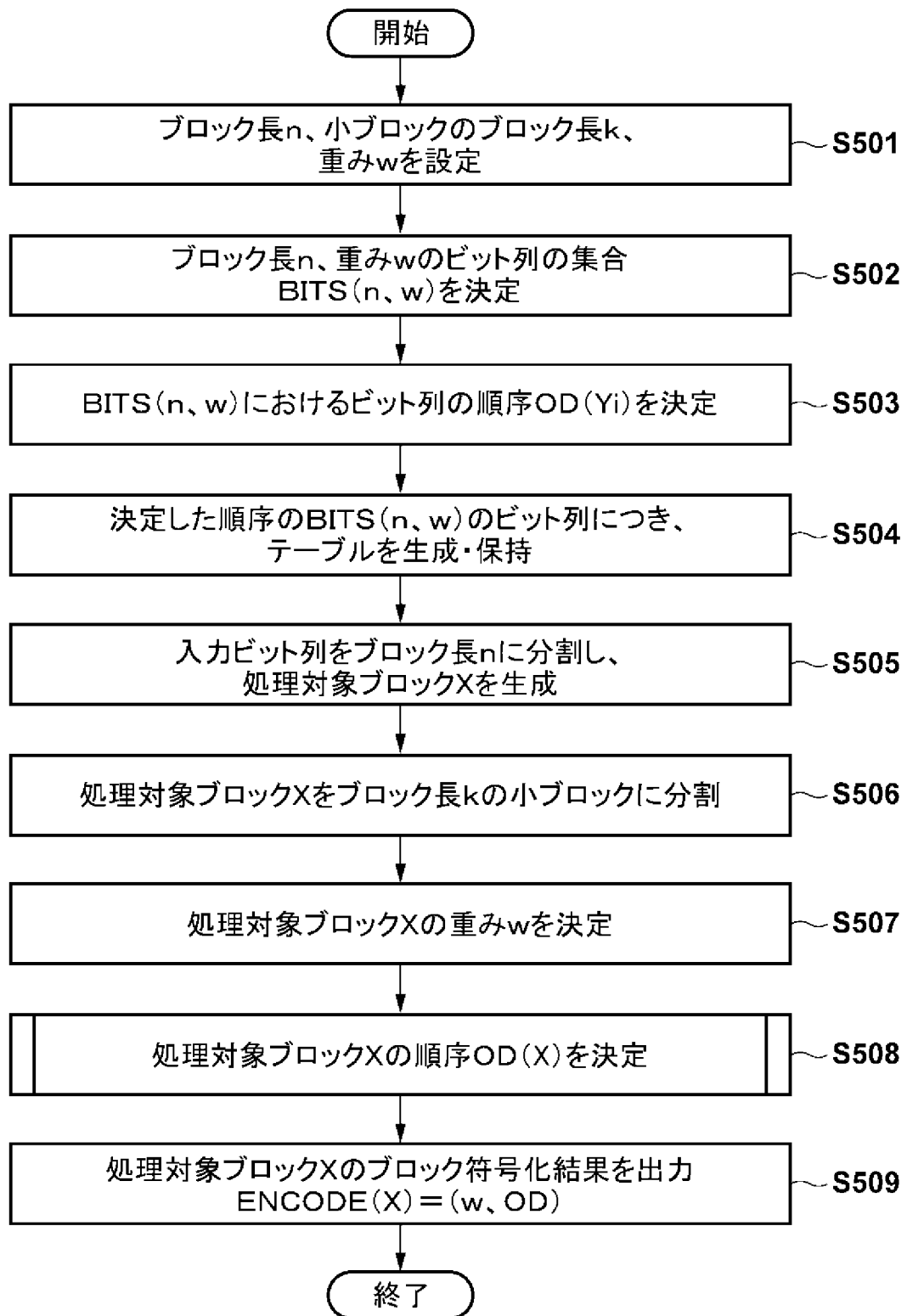
[図4B]



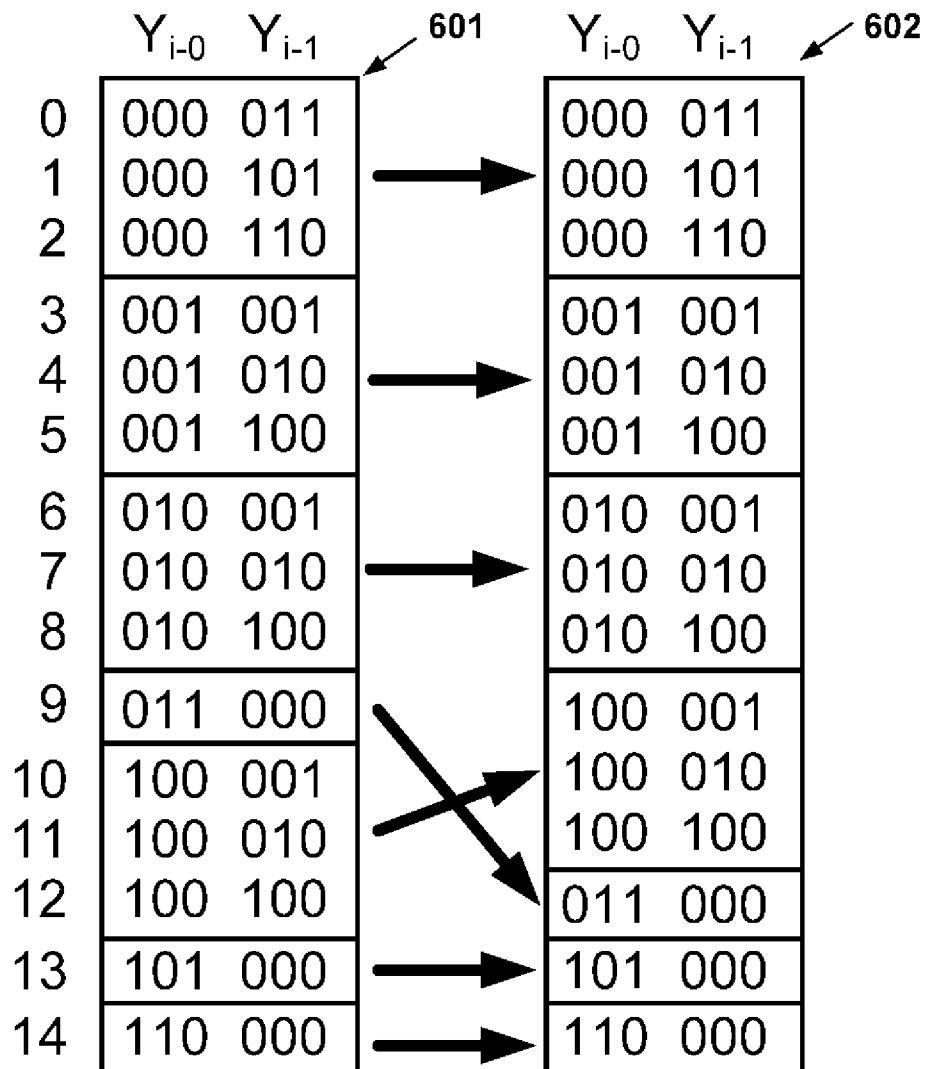
[図4C]

ノード位置	0	1	2	3			4	5	6	7						
ビット位置	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ブロック番号	0								1							
BP表現	(	(	(	(	)	)	)	)	(	(	(	(	)	)	)	)
ビット列	1	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0

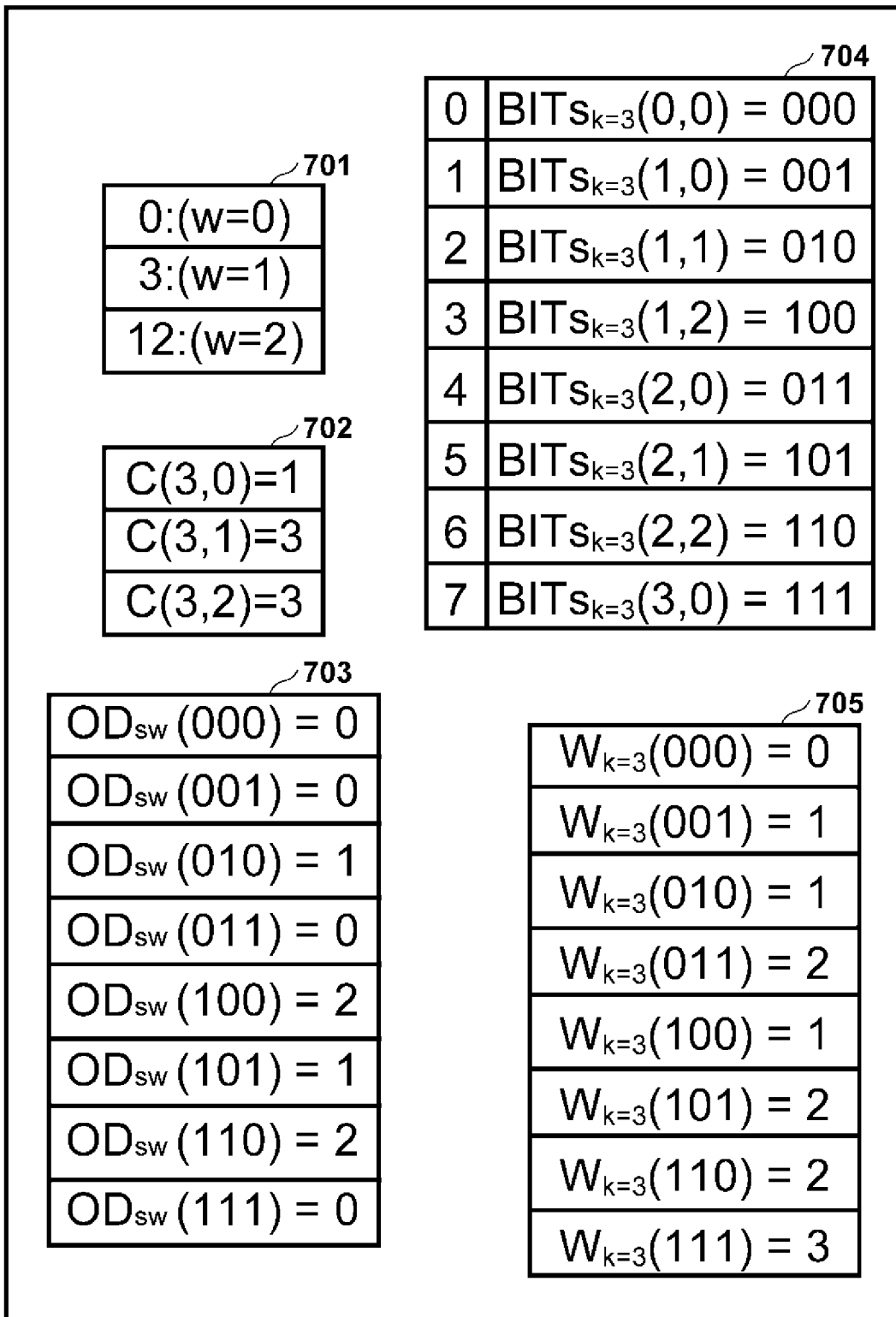
[図5]



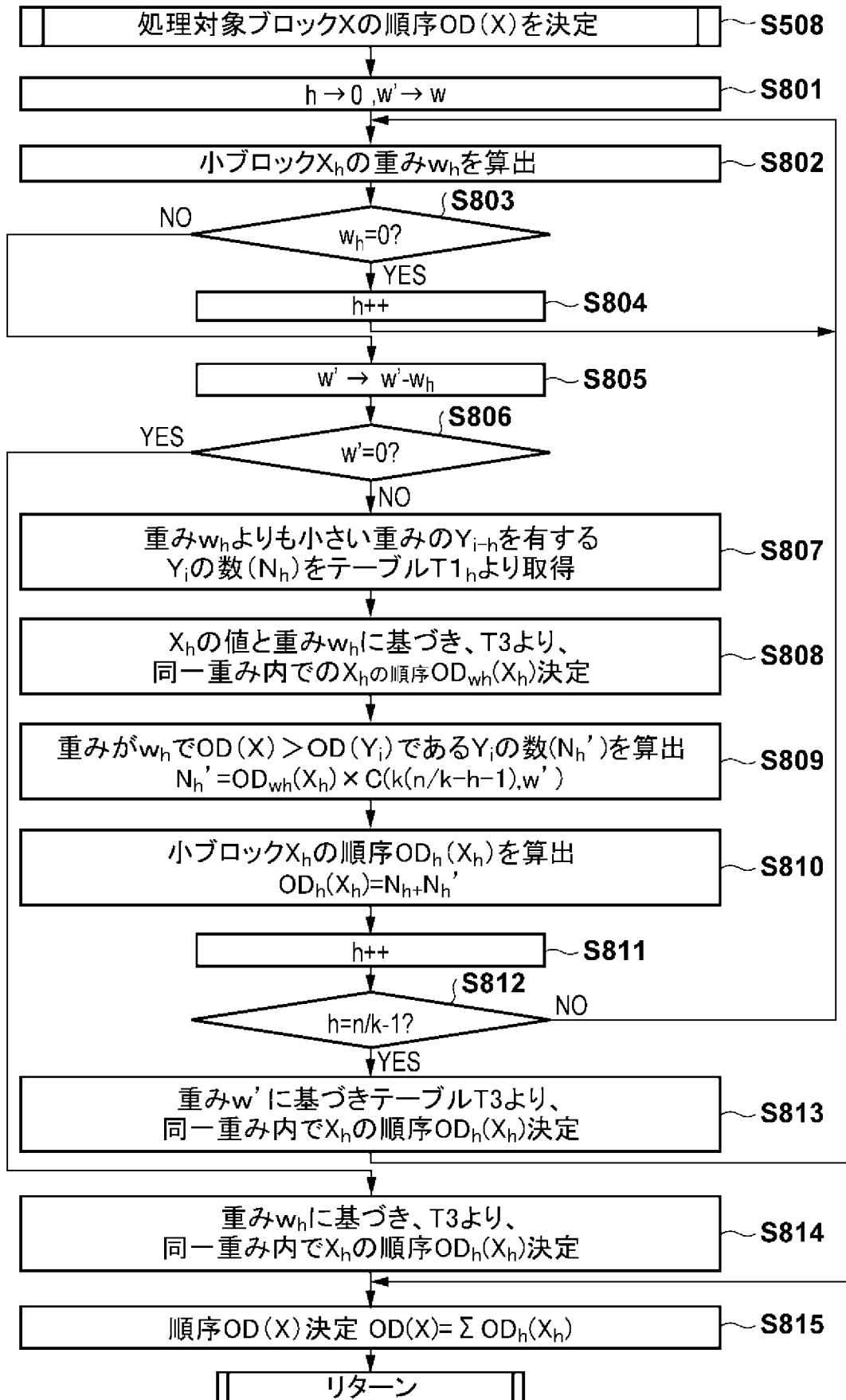
[図6]



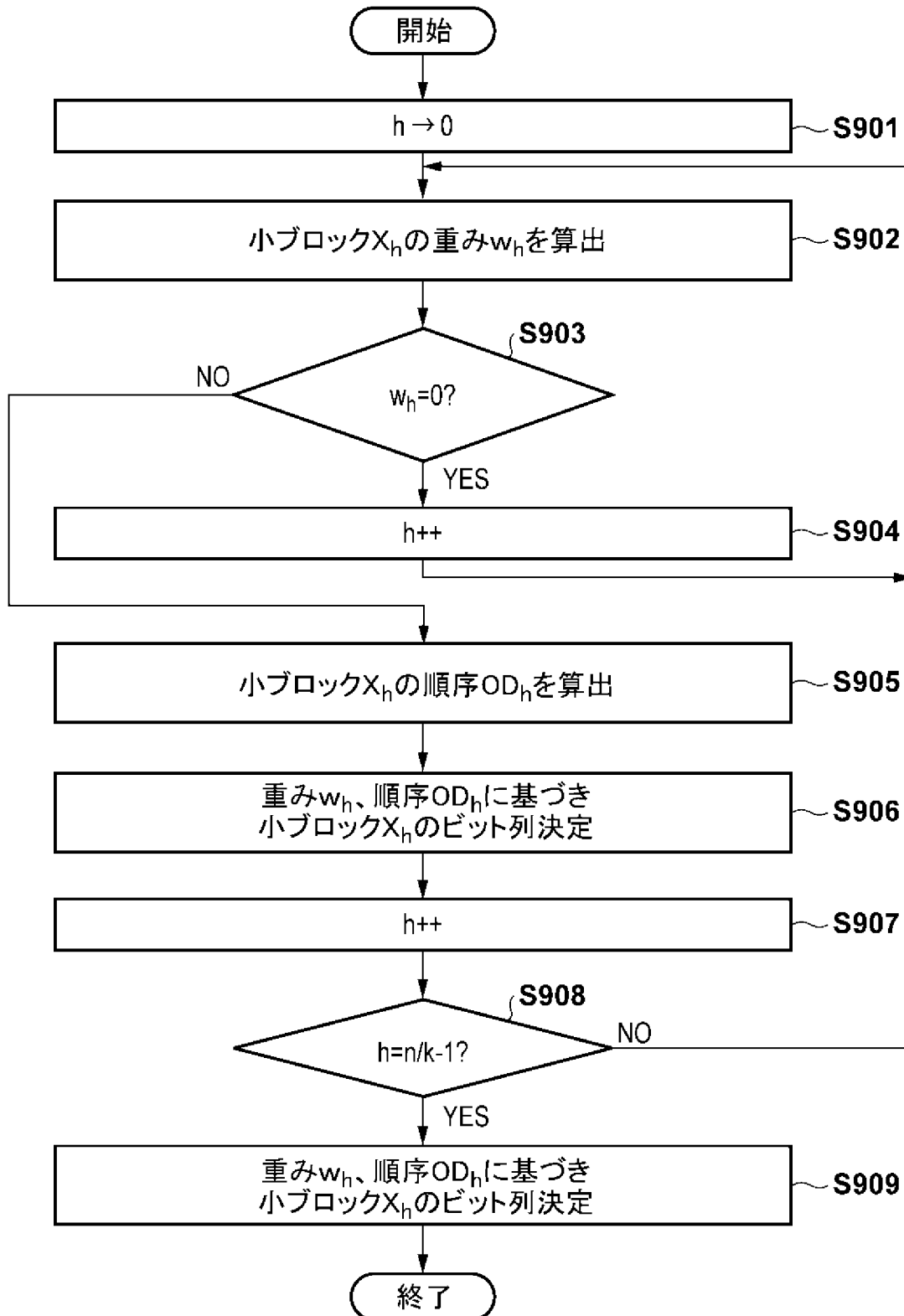
[図7]



[図8]



[図9]



**INTERNATIONAL SEARCH REPORT**

International application No.  
PCT/JP2016/055583

**A. CLASSIFICATION OF SUBJECT MATTER**  
H03M7/30(2006.01)i, G06F17/30(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)  
H03M7/30, G06F17/30

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	1996-2016
Kokai Jitsuyo Shinan Koho	1971-2016	Toroku Jitsuyo Shinan Koho	1994-2016

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

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	Yusaku KANETA, "Jitsuyoteki na Asshuku Rank/ Select Jisho", IPSJ SIG Notes, 28 September 2015 (28.09.2015), vol.2015-AL-154, no.1, pages 1 to 6	1, 13, 15 2-12, 14, 16
A	Kunihiko SADAKANE, "Succinct Data Structures for Large-Scale Data Processing", IPSJ Magazine, 15 August 2007 (15.08.2007), vol.48, no.8, pages 899 to 902	1-16
A	WO 2015/151444 A1 (NEC Solution Innovators, Ltd.), 08 October 2015 (08.10.2015), (Family: none)	1-16

Further documents are listed in the continuation of Box C.       See patent family annex.

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 09 May 2016 (09.05.16)	Date of mailing of the international search report 17 May 2016 (17.05.16)
---	--

Name and mailing address of the ISA/ Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Authorized officer  Telephone No.
--	---

**INTERNATIONAL SEARCH REPORT**

International application No.

PCT/JP2016/055583

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2015/173870 A1 (Rakuten, Inc.), 19 November 2015 (19.11.2015), & JP 5752337 B1	1-16

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int.Cl. H03M7/30(2006.01)i, G06F17/30(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. H03M7/30, G06F17/30

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	1971-2016年
日本国実用新案登録公報	1996-2016年
日本国登録実用新案公報	1994-2016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)

IEEE Xplore, CiNii

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
X A	金田悠作, 実用的な圧縮Rank/Select辞書, 情報処理学会研究報告, 2015.09.28, vol.2015-AL-154, no.1, pp.1-6	1, 13, 15 2-12, 14, 16
A	定兼邦彦, 大規模データ処理のための簡潔データ構造, 情報処理, 2007.08.15, vol.48, no.8, pp.899-902	1-16
A	W0 2015/151444 A1 (NECソリューションイノベータ株式会社) 2015.10.08 (ファミリーなし)	1-16

☑ 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

の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

09.05.2016

国際調査報告の発送日

17.05.2016

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)

北村 智彦

電話番号 03-3581-1101 内線 3556

5K

9297

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	WO 2015/173870 A1 (楽天株式会社) 2015. 11. 19 & JP 5752337 B1	1-16