

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2004 年 11 月 25 日 (25.11.2004)

PCT

(10) 国際公開番号
WO 2004/102810 A1

- (51) 国際特許分類: H03M 13/09, 13/19
- (21) 国際出願番号: PCT/JP2004/005551
- (22) 国際出願日: 2004 年 4 月 19 日 (19.04.2004)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願2003-133942 2003 年 5 月 13 日 (13.05.2003) JP
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 Tokyo (JP).

(YOKOKAWA, Takashi) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 菅 真紀子 (KAN, Makiko) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 飯田 康博 (IIDA, Yasuhiro) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 菊池 敦 (KIKUCHI, Atsushi) [JP/JP]; 〒2400053 神奈川県横浜市保土ヶ谷区新井町 2 2 0 の 2 0 Kanagawa (JP).

(74) 代理人: 稲本 義雄 (INAMOTO, Yoshio); 〒1600023 東京都新宿区西新宿 7 丁目 1 1 番 1 8 号 7 1 1 ビルディング 4 階 Tokyo (JP).

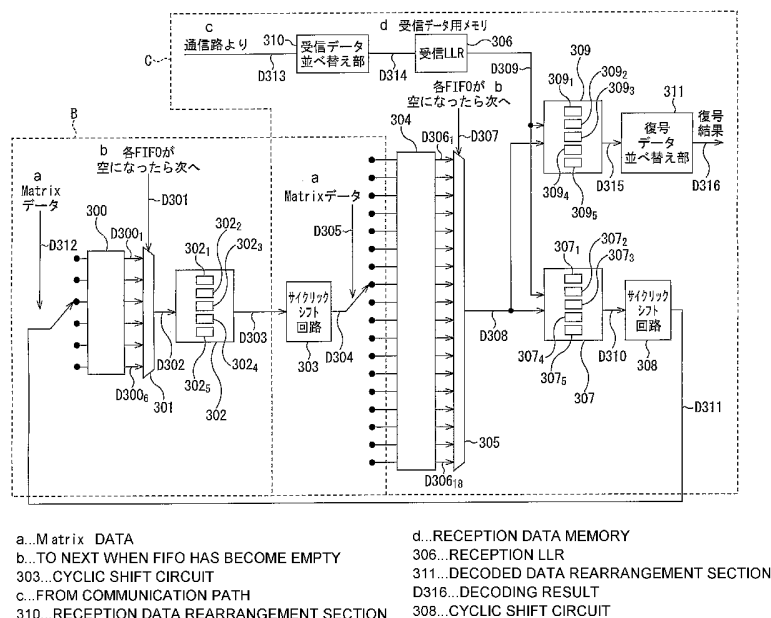
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT,

- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 横川 峰志

[続葉有]

(54) Title: DECODING METHOD, DECODING DEVICE, AND PROGRAM

(54) 発明の名称: 復号方法および復号装置、並びにプログラム



(57) Abstract: There are provided a decoding method, a decoding device, and a program capable of suppressing the operation frequency to a range capable of sufficient realization of it and easily performing control of memory access while suppressing the circuit size. LDPC (Low Density Parity Check) code is decoded by using a conversion inspection matrix obtained one or both of the row replacement and the column replacement for the inspection matrix of elements of the LDPC code. In this case, the conversion inspection matrix can be expressed by a combination of a plurality of constituting matrixes such as a unit matrix of $P \times P$, a quasi-unit matrix in which one or more components 1 of the unit matrix have become 0, a shift matrix in which a unit matrix or a quasi-unit matrix is cyclically shifted, a sum matrix which is a sum of at least two

[続葉有]



LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告書

- (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, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC,

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

of the unit matrix, the quasi-unit matrix, and the shift matrix, or 0 matrix of $P \times P$. A check node calculation section (302) simultaneously performs calculation of P check nodes while a variable node calculation section (304) simultaneously performs calculation of P variable nodes.

(57) 要約: 本発明は、回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるようにする復号方法および復号装置、並びにプログラムに関する。LDPC(Low Density Parity Check)符号の元の検査行列に対して、行置換と列置換のうち的一方または両方を行って得られる変換検査行列を用いて、LDPC符号が復号される。この場合に、 $P \times P$ の単位行列、その単位行列のコンポーネントである1のうちの1個以上が0になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、変換検査行列は、複数の構成行列の組合せで表される。チェックノード計算部302は、チェックノードの演算を、 P 個同時に行い、バリエブルノード計算部304は、バリエブルノードの演算を、 P 個同時に行う。

明細書

復号方法および復号装置、並びにプログラム

技術分野

- 5 本発明は、復号方法および復号装置、並びにプログラムに関し、特に、低密度パリティ検査符号による符号化が施された符号の復号を行う復号方法および復号装置、並びにプログラムに関する。

背景技術

- 10 近年、例えば、移動体通信や深宇宙通信といった通信分野、及び地上波又は衛星デジタル放送といった放送分野の研究が著しく進められているが、それに伴い、誤り訂正符号化及び復号の効率化を目的として符号理論に関する研究も盛んに行われている。

- 符号性能の理論的限界としては、いわゆるシャノン(C. E. Shannon)の通信路
- 15 符号化定理によって与えられるシャノン限界が知られている。符号理論に関する研究は、このシャノン限界に近い性能を示す符号を開発することを目的として行われている。近年では、シャノン限界に近い性能を示す符号化方法として、例えば、並列接続畳み込み符号(PCCC(Parallel Concatenated Convolutional Codes))や、縦列接続畳み込み符号(SCCC(Serially Concatenated
- 20 Convolutional Codes))といった、いわゆるターボ符号化(Turbo coding)と呼ばれる手法が開発されている。また、これらのターボ符号が開発される一方で、古くから知られる符号化方法である低密度パリティ検査符号(Low Density Parity Check codes) (以下、LDPC 符号という) が脚光を浴びつつある。

- LDPC 符号は、R. G. Gallager による「R. G. Gallager, "Low Density
- 25 Parity Check Codes", Cambridge, Massachusetts: M. I. T. Press, 1963」において最初に提案されたものであり、その後、「D. J. C. MacKay, "Good error correcting codes based on very sparse matrices",

Submitted to IEEE Trans. Inf. Theory, IT-45, pp. 399-431, 1999」や、
「M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman,
"Analysis of low density codes and improved designs using irregular
graphs", in Proceedings of ACM Symposium on Theory of Computing, pp.
5 249-258, 1998」等において再注目されるに至ったものである。

LDPC 符号は、近年の研究により、ターボ符号等と同様に、符号長を長くして
いくにしたがって、シャノン限界に近い性能が得られることがわかりつつある。
また、LDPC 符号は、最小距離が符号長に比例するという性質があることから、
その特徴として、ブロック誤り確率特性がよく、さらに、ターボ符号等の復号特
10 性において観測される、いわゆるエラーフロア現象が殆ど生じないことも利点と
して挙げられる。

以下、このような LDPC 符号について具体的に説明する。なお、LDPC 符号は、
線形符号であり、必ずしも 2 元である必要はないが、ここでは、2 元であるもの
として説明する。

15 LDPC 符号は、その LDPC 符号を定義する検査行列 (parity check matrix) が疎
なものであることを最大の特徴とするものである。ここで、疎な行列とは、行列
のコンポーネントの "1" の個数が非常に少なく構成されるものであり、疎な検査
行列を H で表すものとする、そのような検査行列としては、例えば、図 1 に
示すように、各列のハミング重み ("1" の数) (weight) が "3" であり、且つ、各
20 行のハミング重みが "6" であるもの等がある。

このように、各行及び各列のハミング重みが一定である検査行列 H によって
定義される LDPC 符号は、レギュラー LDPC 符号と称される。一方、各行及び各列
のハミング重みが一定でない検査行列 H によって定義される LDPC 符号は、イレ
ギュラー LDPC 符号と称される。

25 このような LDPC 符号による符号化は、検査行列 H に基づいて生成行列 G を生
成し、この生成行列 G を 2 元の情報メッセージに対して乗算することによって
符号語を生成することで実現される。具体的には、LDPC 符号による符号化を行

う符号化装置は、まず、検査行列 H の転置行列 H^T との間に、式 $GH^T=0$ が成立する生成行列 G を算出する。ここで、生成行列 G が、 $k \times n$ 行列である場合には、符号化装置は、生成行列 G に対して k ビットからなる情報メッセージ（ベクトル u ）を乗算し、 n ビットからなる符号語 c （ $= uG$ ）を生成する。この符号化

5 装置によって生成された符号語は、値が“0”の符号ビットが“+1”に、値が“1”の符号ビットが“1”にといったようにマッピングされて送信され、所定の通信路を介して受信側において受信されることになる。

一方、LDPC 符号の復号は、Gallager が確率復号(Probabilistic Decoding)と称して提案したアルゴリズムであって、バリアブルノード(variable node

10 (メッセージノード(message node)ともいう。))と、チェックノード(check node)とからなる、いわゆるタナーグラフ(Tanner graph)上での確率伝播(belief propagation)によるメッセージ・パッシング・アルゴリズムによって行うことが可能である。ここで、以下、適宜、バリアブルノードとチェックノードを、単に、ノードともいう。

15 しかしながら、確率復号においては、各ノード間で受け渡されるメッセージが実数値であることから、解析的に解くためには、連続した値をとるメッセージの確率分布そのものを追跡する必要がある、非常に困難を伴う解析を必要とすることになる。そこで、Gallager は、LDPC 符号の復号アルゴリズムとして、アルゴリズム A 又はアルゴリズム B を提案している。

20 LDPC 符号の復号は、一般的には、図 2 に示すような手順にしたがって行われる。なお、ここでは、受信値（受信した符号系列）を $U_0(u_{0i})$ とし、チェックノードから出力されるメッセージを u_j とし、バリアブルノードから出力されるメッセージを v_i とする。また、ここでは、メッセージとは、値の“0”らしさを、いわゆる対数尤度比(log likelihood ratio)で表現した実数値である。

25 まず、LDPC 符号の復号においては、図 2 に示すように、ステップ S 1 1 において、受信値 $U_0(u_{0i})$ が受信され、メッセージ u_j が“0”に初期化されるとともに、繰り返し処理のカウンタとしての整数をとる変数 k が“0”に初期化され、ステッ

プ S 1 2に進む。ステップ S 1 2において、受信値 $U_0(u_{0i})$ に基づいて、式

(1) に示す演算 (バリアブルノードの演算) を行うことによってメッセージ v_i が求められ、さらに、このメッセージ v_i に基づいて、式 (2) に示す演算 (チェックノードの演算) を行うことによってメッセージ u_j が求められる。

$$5 \quad v_i = u_{0i} + \sum_{j=1}^{d_v-1} u_j \quad \dots (1)$$

$$\tanh\left(\frac{u_j}{2}\right) = \prod_{i=1}^{d_c-1} \tanh\left(\frac{v_i}{2}\right) \quad \dots (2)$$

ここで、式 (1) と式 (2) における d_v と d_c は、それぞれ、検査行列 H の縦方向 (列) と横方向 (行) の "1" の個数を示す任意に選択可能とされるパラメータであり、例えば、(3,6) 符号の場合には、 $d_v=3$, $d_c=6$ となる。

- 10 なお、式 (1) または (2) の演算においては、それぞれ、メッセージを出力しようとする枝 (edge) (バリアブルノードとチェックノードとを結ぶ線) から入力されたメッセージを、和または積演算のパラメータとしては用いないことから、和または積演算の範囲が、1 乃至 d_v-1 または 1 乃至 d_c-1 となっている。また、式 (2) に示す演算は、実際には、2 入力 v_1 , v_2 に対する 1 出力で定義される式 (3) に示す関数 $R(v_1, v_2)$ のテーブルを予め作成しておき、これを式
- 15 (4) に示すように連続的 (再帰的) に用いることによって行われる。

$$x = 2 \tanh^{-1} \{ \tanh(v_1/2) \tanh(v_2/2) \} = R(v_1, v_2) \quad \dots (3)$$

$$20 \quad u_j = R(v_1, R(v_2, R(v_3, \dots R(v_{d_c-2}, v_{d_c-1})))) \quad \dots (4)$$

ステップ S 1 2では、さらに、変数 k が "1" だけインクリメントされ、ステップ S 1 3に進む。ステップ S 1 3では、変数 k が所定の繰り返し復号回数 N よりも大きいかな否かが判定される。ステップ S 1 3において、変数 k が N よりも

大きくないと判定された場合、ステップ S 1 2 に戻り、以下、同様の処理が繰り返される。

- また、ステップ S 1 3 において、変数 k が N よりも大きいと判定された場合、ステップ S 1 4 に進み、式 (5) に示す演算を行うことによって最終的に出力する復号結果としてのメッセージ v_i が求められて出力され、LDPC 符号の復号処理が終了する。

$$v_i = u_{0i} + \sum_{j=1}^{d_v} u_j \quad \dots (5)$$

ここで、式 (5) の演算は、式 (1) の演算とは異なり、バリアブルノードに接続している全ての枝からの入力メッセージを用いて行われる。

- このような LDPC 符号の復号は、例えば (3, 6) 符号の場合には、図 3 に示すように、各ノード間でメッセージの授受が行われる。なお、図 3 における "=" で示すノード (バリアブルノード) では、式 (1) に示した演算が行われ、"+" で示すノード (チェックノード) では、式 (2) に示した演算が行われる。特に、アルゴリズム A においては、メッセージを 2 元化し、"+" で示すノードにて、 $d_c - 1$ 個の入力メッセージの排他的論理和演算を行い、 "=" で示すノードにて、受信値 R に対して、 $d_v - 1$ 個の入力メッセージが全て異なるビット値であった場合には、符号を反転して出力する。

また、一方で、近年、LDPC 符号の復号の実装法に関する研究も行われている。実装方法について述べる前に、まず、LDPC 符号の復号を模式化して説明する。

- 図 4 は、(3, 6) LDPC 符号 (符号化率 1/2、符号長 12) の検査行列 (parity check matrix) の例である。LDPC 符号の検査行列は、図 5 のように、タナグラフを用いて書き表すことができる。ここで、図 5 において、"+" で表わされるのが、チェックノードであり、 "=" で表わされるのが、バリアブルノードである。チェックノードとバリアブルノードは、それぞれ、検査行列の行と列に対応する。チェックノードとバリアブルノードとの間の結線は、枝 (edge) であり、検査行列の "1" に相当する。即ち、検査行列の第 j 行第 i 列のコンポーネントが 1 であ

る場合には、図 5 において、上から i 番目のバリエブルノード（"=" のノード）と、上から j 番目のチェックノード（"+" のノード）とが、枝により接続される。枝は、バリエブルノードに対応する符号ビットが、チェックノードに対応する拘束条件を持つことを表わす。なお、図 5 は、図 4 の検査行列のタナグラフとなっている。

LDPC 符号の復号方法であるサムプロダクトアルゴリズム (Sum Product Algorithm) では、バリエブルノードの演算とチェックノードの演算とが繰り返して行われる。

バリエブルノードでは、図 6 のように、式 (1) の演算（バリエブルノード演算）を行う。すなわち、図 6 において、計算しようとしている枝に対応するメッセージ v_i は、バリエブルノードに繋がっている残りの枝からのメッセージ u_1 および u_2 と、受信情報 u_{0i} を用いて計算される。他の枝に対応するメッセージも同様に計算される。

次に、チェックノードの演算について説明する前に、式 (2) を、式 $a \times b = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$ の関係を用いて、式 (6) のように書き直す。但し、 $\text{sign}(x)$ は、 $x \geq 0$ のとき 1 であり、 $x < 0$ のとき -1 である。

$$\begin{aligned}
 u_j &= 2 \tanh^{-1} \left(\prod_{i=1}^{d_c-1} \tanh \left(\frac{v_i}{2} \right) \right) \\
 &= 2 \tanh^{-1} \left[\exp \left\{ \sum_{i=1}^{d_c-1} \ln \left(\left| \tanh \left(\frac{v_i}{2} \right) \right| \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign} \left(\tanh \left(\frac{v_i}{2} \right) \right) \right] \\
 &= 2 \tanh^{-1} \left[\exp \left\{ - \left(\sum_{i=1}^{d_c-1} - \ln \left(\tanh \left(\frac{|v_i|}{2} \right) \right) \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \right] \\
 &\quad \dots (6)
 \end{aligned}$$

更に、 $x \geq 0$ において、 $\phi(x) = \ln(\tanh(x/2))$ と定義すると、 $\phi^{-1}(x) = 2 \tanh^{-1}(e^{-x})$ であるから、式 (6) は、式 (7) のように書くことができる。

$$u_j = \phi^{-1} \left(\sum_{i=1}^{d_c-1} \phi(|v_i|) \right) \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \quad \dots (7)$$

チェックノードでは、図7のように、式(7)の演算(チェックノード演算)を行う。すなわち、図7において、計算しようとしている枝に対応するメッセー

5 ジ u_j は、チェックノードに繋がっている残りの枝からのメッセージ

v_1, v_2, v_3, v_4, v_5 を用いて計算される。他の枝に対応するメッセージも同様に計算される。

なお、関数 $\phi(x)$ は、 $\phi(x) = \ln((e^x+1)/(e^x-1))$ と表すことができ、 $x > 0$ に

10 する際には、LUT(Look Up Table)を用いて実装される場合があるが、両者共に同一のLUTとなる。

サンプロダクトアルゴリズムをハードウェアに実装する場合、式(1)で表わされるバリアブルノード演算および式(7)で表わされるチェックノード演算を、適度な回路規模と動作周波数で繰り返し行うことが必要である。

15 復号装置の実装の例として、まず、単純に各ノードの演算を一つずつ順次行うことによって復号を行う場合(full serial decoding)の実装法について説明する。

20 なお、ここでは、例えば、図8の、30(行)×90(列)の検査行列で表現される符号(符号化率2/3、符号長90)を復号することとする。図8の検査行列の1の数は269であり、従って、そのタナーグラフでは、枝の数は269個となる。ここで、図8の検査行列では(後述する図15乃至図17においても同様)、0を、“.”で表現している。

図9は、LDPC符号の1回復号を行う復号装置の構成例を示している。

25 図9の復号装置では、その動作する1クロック(clock)ごとに、1つの枝に対応するメッセージが計算される。

即ち、図 9 の復号装置は、1 つの受信用メモリ 104、2 つの枝用メモリ 100 および 102、1 つのチェックノード計算器 101、1 つのバリエブルノード計算器 103 からなる。

図 9 の復号装置では、枝用メモリ 100 または 102 からメッセージデータが一つずつ読み出され、そのメッセージデータを用いて、所望の枝に対応するメッセージデータが計算される。そして、その計算によって求められたメッセージデータが一つずつ後段の枝用メモリ 102 または 100 に格納されていく。繰り返し復号を行う際には、この 1 回復号を行う図 9 の復号装置を複数個縦列に接続するか、もしくは図 9 の復号装置を繰り返し用いることによって、繰り返し復号を実現する。なお、ここでは、例えば、図 9 の復号装置が複数個接続されているものとする。

枝用メモリ 100 は、前段の復号装置（図示せず）のバリエブルノード計算器 103 から供給される出力メッセージ D100 を、後段のチェックノード計算器 101 が読み出す順番に格納していく。そして、枝用メモリ 100 は、チェックノード計算のフェーズでは、メッセージ D100 を、格納してある順番通りに、メッセージ出力 D101 として、チェックノード計算器 101 に供給する。チェックノード計算器 101 は、枝用メモリ 100 から供給されるメッセージ D101 を用いて、式 (7) に従って演算を行い、その演算によって求められたメッセージ D102 を、後段の枝用メモリ 102 に供給する。

ここで、図 10 は、チェックノード計算を一つずつ行う図 9 のチェックノード計算器 101 の構成例を示している。

図 10 のチェックノード計算器 101 では、枝用メモリ 100 から供給される、検査行列の各列に対応するバリエブルノードからのメッセージ v_i を一つずつ読み込み、式 (7) における $\phi(|v_i|)$ の演算を LUT によって行う。さらに、検査行列の 1 行に亘る各列に対応するバリエブルノードからのメッセージ v_i から求められた $\phi(|v_i|)$ が積算され、これにより、全ての枝からのメッセージ v_i から求められた $\phi(|v_i|)$ の積算値が求められる。その後、その積算値から、メッセ

- ージ u_j を求めたい枝から求められて FIFO (FIFO メモリ) で遅延された $\phi(|v_i|)$ が減算され、これにより、メッセージ u_j を求めたい枝について、式 (7) における $\sum \phi(|v_i|)$ が求められる。即ち、チェックノードへの枝すべてのメッセージの和から、メッセージ u_j を求めたい枝からのメッセージを減算することで、メッセージ u_j を求めたい枝へのメッセージが求められる。さらに、LUT によって、式 (7) における $\phi^{-1}(\sum \phi(|v_i|))$ の演算が行われる。同時に、メッセージ u_j の符号ビット、即ち、式 (7) における $\prod \text{sign}(v_i)$ も、EXOR 回路を用いて同様に計算される。以上のようにして、式 (7) の演算が行われ、メッセージ u_j が求められる。
- 10 なお、図 10 では、各メッセージが符号ビットを合わせて合計 6 ビット (bit) に量子化されているものとして、チェックノード計算器 101 を表している。また、ここで処理の対象としている図 8 の検査行列の行の重み (row weight) の最大は 9 であるため、即ち、チェックノードに供給されるメッセージの最大数は 9 であるため、チェックノード計算器 101 は、9 個のメッセージ ($\phi(|v_i|)$) を
- 15 遅延させる FIFO (First In First Out) を持っている。
- 図 9 に戻り、枝用メモリ 102 は、前段のチェックノード計算器 101 から供給される出力メッセージ D102 を、後段のバリアブルノード計算器 103 が読み出す順番に格納していく。そして、枝用メモリ 102 は、バリアブルノード計算のフェーズでは、メッセージ出力 D102 を、格納してある順番通りに、メッセージ出力 D103 として、バリアブルノード計算器 103 に供給する。
- 20 バリアブルノード計算器 103 は、枝用メモリ 102 から供給されるメッセージ D103 と受信用メモリ 104 から供給される受信データ (LDPC 符号の受信値) D104 を用いて式 (1) に従って演算を行い、その演算の結果得られるメッセージ D105 を、図示せぬ後段の復号装置の枝用メモリ 100 に供給する。
- 25 ここで、図 11 は、バリアブルノード計算を一つずつ行う図 9 のバリアブルノード計算器 103 の構成例を示している。

図 1 1 のバリエブルノード計算器 1 0 3 では、枝用メモリ 1 0 2 から供給される、検査行列の各行に対応するチェックノードからのメッセージ u_j を一つずつ読み込み、検査行列の 1 列に亘る各行に対応するチェックノードからのメッセージを積算して、その積算値を求める。その後、その積算値から、メッセージ v_i を求めたい枝から供給されて FIFO で遅延されたメッセージが減算される。さらに、その結果得られる減算値から、受信値 u_{0i} を加算することで、式 (1) の演算が行われ、これにより、メッセージ v_i が求められる。即ち、バリエブルノードへの枝すべてからのメッセージの和から、メッセージ v_i を求めたい枝からのメッセージを減算することで、メッセージ v_i を求めたい枝へのメッセージが求められる。

図 1 1 においても、図 1 0 における場合と同様に、各メッセージが符号ビットを合わせて合計 6 ビットに量子化されているものとして、バリエブルノード計算器 1 0 3 を表している。また、ここで処理の対象としている図 8 の検査行列においては、列の重み(column weight)の最大値が 5 であるため、バリエブル計算器 1 0 3 は、5 個のメッセージを遅延させる FIFO を持っており、列の重みが 5 未満の列のメッセージを計算するときには、FIFO における遅延量が、その列の重みの値に減らされる。

再び、図 9 に戻り、復号装置には、検査行列の重みにしたがって、図示しない制御信号が与えられる。そして、図 9 の復号装置によれば、枝用メモリ 1 0 0 および 1 0 2、並びにチェックノード計算器 1 0 1 およびバリエブルノード計算器 1 0 3 の FIFO の容量さえ足りれば、制御信号のみを変えることで様々な符号を復号することができる。

なお、図示しないが、図 9 の復号装置において、復号の最終段においては、式 (1) のバリエブルノード演算の代わりに、式 (5) の演算が行われ、その演算結果が、最終的な復号結果として出力される。

図 9 の復号装置を繰り返し用いて、LDPC 符号を復号する場合には、チェックノード演算とバリエブルノード演算とが交互に行われるため、2 6 9 の枝を有す

る図 8 の検査行列を用いた 1 回の復号に、 $269 \times 2 = 538$ クロック (clock) を必要とする。従って、例えば、50 回の繰り返し復号を行うためには、符号長である 90 個の符号情報 (受信値) を受信する間に、 $538 \times 50 = 26900$ クロック動作することが必要であり、受信周波数の約 300 ($\div 26900/90$) 倍の高速動作が必要になる。この場合、受信周波数が数十 MHz であるとする、GHz 以上の速度での動作を要求されることになり、実装は容易ではない。

また、図 9 の復号装置を、例えば、50 台連接して、LDPC 符号を復号する場合には、1 フレーム (frame) 目がバリアブルノード演算を行っている間に、2 フレーム目はチェックノード演算を行い、3 フレーム目は前段のバリアブルノード演算を行う、というように、複数のバリアブルノード演算とチェックノード演算とを同時に行うことができる。この場合、90 個の符号情報を受信する間に、269 個の枝を計算すればよいので、復号装置は、受信周波数の約 3 ($\div 269/90$) 倍の周波数で動作すればよいことになり、十分に実現可能である。しかしながら、この場合、回路規模が、単純には、図 9 の復号装置の 50 倍になる。

次に、全ノードの演算を同時に行うことによって復号を行う場合 (full parallel decoding) の復号装置の実装法について説明する。

この実装法については、例えば、C. Howland and A. Blanksby, "Parallel Decoding Architectures for Low Density Parity Check Codes", Symposium on Circuits and Systems, 2001 に記載されている。

図 12A 乃至図 12C は、図 8 の検査行列で表現される符号 (符号化率 $2/3$ 、符号長 90) を復号する復号装置の一例の構成を示している。なお、図 12A は、復号装置全体の構成を示している。また、図 12B は、図 12A の復号装置の点線 B で囲まれた図中上部の詳細構成を示し、図 12C は、図 12A の復号装置の点線 C で囲まれた図中下部の詳細構成を示している。

図 12A 乃至図 12C の復号装置は、1 つの受信用メモリ 205、2 つの枝入れ替え装置 200 および 203、2 つの枝用メモリ 202 および 206、30 個

のチェックノード計算器 201_1 乃至 201_{30} から構成されるチェックノード計算器 201 、 90 個のバリアブルノード計算器 204_1 乃至 204_{90} から構成されるバリアブルノード計算器 204 からなる。

- 図 $12A$ 乃至図 $12C$ の復号装置では、枝用メモリ 202 または 206 から、
 5 269 個ある枝に対応するメッセージデータを全て同時に読み出し、そのメッセージデータを用いて、 269 個の枝に対応する新たなメッセージデータを演算する。さらに、その演算の結果求められた新たなメッセージデータが全て同時に後段の枝用メモリ 206 または 202 に格納されていく。そして、図 $12A$ 乃至図 $12C$ の復号装置を繰り返し用いることで繰り返し復号が実現される。以下、各部について詳細に説明する。
 10

- 枝用メモリ 206 は、前段のバリアブルノード計算器 204_1 乃至 204_{90} からの出力メッセージ $D206_1$ 乃至 $D206_{90}$ を全て同時に格納し、次の時刻（次のクロックのタイミング）に、メッセージ $D206_1$ 乃至 $D206_{90}$ を、メッセージ $D207_1$ 乃至 $D207_{90}$ として読み出し、次段の枝入れ替え装置 200 に、メッセージ
 15 $D200$ ($D200_1$ 乃至 $D200_{90}$) として供給する。枝入れ替え装置 200 は、枝用メモリ 206 から供給されたメッセージ $D200_1$ 乃至 $D200_{90}$ の順番を、図 8 の検査行列に従って並び替え（入れ替え）、チェックノード計算器 201_1 乃至 201_{30} に、それぞれ必要なメッセージ $D201_1$ 乃至 $D201_{30}$ を供給する。

- チェックノード計算器 201_1 乃至 201_{30} は、枝入れ替え装置 200 から供給されるメッセージ $D201_1$ 乃至 $D201_{30}$ を用いて式 (7) に従って演算を行い、その演算の結果得られるメッセージ $D202_1$ 乃至 $D202_{30}$ を、枝用メモリ 202 に供給する。
 20

- ここで、図 13 は、チェックノード演算を同時に行う図 $12A$ 乃至図 $12C$ のチェックノード計算器 201_m ($m=1, 2, \dots, 30$) の構成例を示している。
 25

図 1 3 のチェックノード計算器 2 0 1_mでは、図 1 0 のチェックノード計算器 1 0 1 と同様にして、式 (7) のチェックノード演算が行われるが、そのチェックノード演算が、すべての枝について同時に行われる。

即ち、図 1 3 のチェックノード計算器 2 0 1_mでは、枝入れ替え装置 2 0 0 から供給される図 8 の検査行列の各列に対応するバリエブルノードからのメッセージが全て同時に読み込まれ、式 (7) における $\phi(|v_i|)$ の演算が LUT によって行われる。さらに、検査行列の 1 行に亘る各列に対応するバリエブルノードからのメッセージ v_i から求められた $\phi(|v_i|)$ が積算され、これにより、全ての枝からのメッセージ v_i から求められた $\phi(|v_i|)$ の積算値が求められる。その後、その積算値から、メッセージ u_j を求めたい枝から求められた $\phi(|v_i|)$ が減算され、これにより、メッセージ u_j を求めたい枝について、式 (7) における $\Sigma \phi(|v_i|)$ が求められる。即ち、チェックノードへの枝すべてからのメッセージの和から、メッセージ u_j を求めたい枝からのメッセージを減算することで、メッセージ u_j を求めたい枝へのメッセージが求められる。さらに、LUT によって、式 (7) における $\phi^{-1}(\Sigma \phi(|v_i|))$ の演算が行われる。同時に、メッセージ u_j の符号ビット、即ち、式 (7) における $\Pi \text{sign}(v_i)$ も、EXOR 回路を用いて同様に計算される。以上のようにして、式 (7) の演算が行われ、メッセージ u_j が求められる。

なお、図 1 3 では、各メッセージが符号ビットを合わせて合計 6 ビットに量子化されているものとして、チェックノード計算器 2 0 1_mを表している。また、図 1 3 の回路は一つのチェックノードに相当する。ここで処理の対象としている図 8 の検査行列については、その行数である 3 0 行のチェックノードが存在するから、図 1 2 A 乃至図 1 2 C の復号装置は、図 1 3 に示したようなチェックノード計算器 2 0 1_mを 3 0 個有している。

ここで、図 1 3 のチェックノード計算器 2 0 1_mでは、9 個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図 8 の検査行列の行の重みは、第 1 行が 8 で、第 2 乃至第 3 0 行が 9 であるため、即ち、チェ

ックノードに供給されるメッセージの数が、8のケースが1つと、9のケースが29あるため、チェックノード計算器201₁は、図13の回路と同様の8つのメッセージを同時に計算することができる回路構成となっており、チェックノード計算器201₂乃至201₃₀は、図13の回路と同一構成となっている。

- 5 図12A乃至図12Cに戻り、枝用メモリ202は、前段のチェックノード計算器201₁乃至201₃₀から供給される出力メッセージD202₁乃至D202₃₀を全て同時に格納し、次の時刻に、そのすべてのメッセージD202₁乃至D202₃₀を、出力メッセージD203₁乃至D203₃₀として、次段の枝入れ替え装置203に供給する。

- 枝入れ替え装置203は、枝用メモリ202から供給されたメッセージD203₁乃至D203₃₀の順番を図8の検査行列に従って並び替え、バリアブルノード計算器204₁乃至204₉₀に、それぞれ必要なメッセージD204₁乃至D204₉₀を供給する。

- バリアブルノード計算器204₁乃至204₉₀は、枝入れ替え装置203から供給されるメッセージD204₁乃至D204₉₀と、受信用メモリ205から供給される受信データ（受信値）D205₁乃至D205₉₀を用いて式（1）に従って演算を行い、その演算の結果得られるメッセージD206₁乃至D206₉₀を、次段の枝用メモリ206に供給する。

- ここで、図14は、バリアブルノード演算を同時に行う図12A乃至図12Cのバリアブルノード計算器204_p（ $p = 1, 2, \dots, 90$ ）の構成例を示している。

図14のバリアブルノード計算器204_pでは、図11のバリアブルノード計算器103と同様にして、式（7）のチェックノード演算が行われるが、そのチェックノード演算が、すべての枝について同時に行われる。

- 即ち、図14のバリアブルノード計算器204_pでは、枝入れ替え装置203から供給される、検査行列の各行に対応するチェックノードからのメッセージ u_j が全て同時に読み込まれ、検査行列の1列に亘る各行に対応するチェックノードからのメッセージが積算されて、その積算値が求められる。その後、その積算

値から、メッセージ v_i を求めたい枝から供給されたメッセージが減算され、その結果得られる減算値から、受信値 u_{0i} を加算することで、式 (1) の演算が行われ、これにより、メッセージ v_i が求められる。即ち、バリエブルノードへの枝すべてからのメッセージの和から、メッセージ v_i を求めたい枝からのメッセージを減算することで、メッセージ v_i を求めたい枝へのメッセージが求められる。

なお、図 1 4 では、各メッセージが符号ビットを合わせて合計 6 ビットに量子化されているものとして、バリエブルノード計算器 2 0 4_p を表している。また、図 1 4 の回路は一つのバリエブルノードに相当する。ここで処理の対象としている図 8 の検査行列については、その列数である 9 0 列のバリエブルノードが存在するから、図 1 2 A 乃至図 1 2 C の復号装置は、図 1 4 に示したようなバリエブルノード計算器 2 0 4_p を 9 0 個有している。

ここで、図 1 4 のバリエブルノード計算器 2 0 4_p では、5 個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図 8 の検査行列は、重みが 5, 3, 2, 1 の列が、それぞれ、1 5 列、4 5 列、2 9 列、1 列あるので、バリエブルノード計算器 2 0 4₁ 乃至 2 0 4₉₀ のうちの 1 5 個は、図 1 4 の回路と同一構成となっており、残りの 4 5, 2 9, 1 個は、図 1 4 の回路と同様の 3, 2, 1 つのメッセージを同時にそれぞれ計算することができる回路構成となっている。

なお、図示しないが、図 1 2 A 乃至図 1 2 C の復号装置においても、図 9 における場合と同様に、復号の最終段においては、式 (1) のバリエブルノード演算の代わりに、式 (5) の演算が行われ、その演算結果が復号結果として出力される。

図 1 2 A 乃至図 1 2 C の復号装置によれば、2 6 9 個ある枝に対応するメッセージすべてを 1 クロックで同時に計算することができる。

図 1 2 A 乃至図 1 2 C の復号装置を繰り返し用いて復号する場合には、チェックノード演算とバリエブルノード演算とを交互に行い、1 回の復号を 2 クロック

で行うことができる。従って、例えば、50回の復号を行うためには、90個の符号情報を受信する間に $2 \times 50 = 100$ クロック動作すれば良いことになり、ほぼ受信周波数と同一の動作周波数でよいことになる。一般的に、LDPC符号は、符号長が数千から数万と大きいことから、図12A乃至図12Cの復号装置を用い
5 れば、復号回数を極めて多くすることができ、誤り訂正性能の向上が期待できる。

しかしながら、図12A乃至図12Cの復号装置は、タナグラフのすべての枝に対応するメッセージの演算を、並列で行うため、回路規模が、符号長に比例して大きくなる。また、図12A乃至図12Cの復号装置を、ある符号長の、ある符号化率の、ある検査行列を持つLDPC符号の復号を行う装置として構成した
10 場合、その復号装置において、他の符号長や、他の符号化率、他の検査行列を持つLDPC符号の復号を行うことは困難となる。即ち、図12A乃至図12Cの復号装置は、図9の復号装置のように、制御信号を変えるだけでは、様々な符号を復号することはできず、符号依存性が高い。

図9および図12A乃至図12Cの復号装置の他に、一つでも全てでもなく、
15 4つずつのメッセージの計算を同時に行う実装法について、例えば、E. Yeo, P. Pakzad, B. Nikolic and V. Anantharam, "VLSI Architectures for iterative Decoders in Magnetic Recording Channels", IEEE Transactions on Magnetics, Vol. 37, No. 2, March 2001に述べられているが、この場合、メモリの異なるアドレスからの同時読み出し、もしくは同時書き込みを避けることが一般的には容易でなく、メモリアクセス制御が困難である
20 という問題がある。

また、サンプロダクトアルゴリズムを近似して実装する方法なども提案されているが、この方法では、性能の劣化を招いてしまう。

図15は、符号長90、符号化率2/3のLDPC符号の検査行列の一例である。この検査行列で表されるLDPC符号の復号装置を実装することを考える場合、枝に対応するメッセージを一つずつ計算する復号装置、もしくは枝に対応するメッセージを全て同時に計算する復号装置の設計自体は難しいことではない。
25

しかしながら、その復号装置の実現は、回路規模や動作速度の面から見て、容易ではない。

- また、図 15 の検査行列で表される符号を、ある数 P 個の枝を同時に計算する復号装置を用いて復号する場合、枝データ（枝に対応するメッセージ）を格納するメモリにおいては、各行または各列毎に異なる位置（アドレス）からの読み出しまたは書き込みのアクセスが必要になるため、各行または各列毎に別々の FIFO を用いることが必要になる。更に、メッセージについては、チェックノード演算で計算された順序と、次のバリアブルノード演算で使われる順番が入れ替わることもあり、メッセージを格納するメモリを、単純に FIFO で実現することも容易ではない。

発明の開示

- 本発明は、このような状況に鑑みてなされたものであり、ロジック、メモリ共に回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるようにするものである。

本発明の復号方法は、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC 符号を復号する復号ステップを備えることを特徴とする。

- 本発明の復号装置は、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC 符号を復号する復号手段を備えることを特徴とする。

本発明のプログラムは、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC 符号を復号する復号ステップを備えることを特徴とする。

- 本発明においては、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC 符号が復号される。

図面の簡単な説明

図 1 は、LDPC 符号の検査行列 H を説明する図である。

図 2 は、LDPC 符号の復号手順を説明するフローチャートである。

図 3 は、メッセージの流れを説明する図である。

5 図 4 は、LDPC 符号の検査行列の例を示す図である。

図 5 は、検査行列のタナーグラフを示す図である。

図 6 は、バリアブルノードを示す図である。

図 7 は、チェックノードを示す図である。

図 8 は、LDPC 符号の検査行列の例を示す図である。

10 図 9 は、ノード演算を一つずつ行う LDPC 符号の復号装置の構成例を示すブロック図である。

図 10 は、メッセージを一つずつ計算するチェックノード計算器の構成例を示すブロック図である。

図 11 は、メッセージを一つずつ計算するバリアブルノード計算器の構成例を示すブロック図である。

図 12 A は、ノード演算を全て同時に行う LDPC 符号の復号装置の構成例を示すブロック図である。

図 12 B は、ノード演算を全て同時に行う LDPC 符号の復号装置の構成例を示すブロック図である。

20 図 12 C は、ノード演算を全て同時に行う LDPC 符号の復号装置の構成例を示すブロック図である。

図 13 は、メッセージを同時に計算するチェックノード計算器の構成例を示すブロック図である。

図 14 は、メッセージを同時に計算するバリアブルノード計算器の構成例を示すブロック図である。

図 15 は、LDPC 符号の検査行列の例を示す図である。

図 16 は、検査行列に行置換と列置換を施した行列を示す図である。

図 1 7 は、 5×5 単位に分割した検査行列を示す図である。

図 1 8 A は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

図 1 8 B は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

図 1 8 C は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

図 1 9 は、本発明を適用したコンピュータの一実施の形態の構成例を示すブロック図である。

10

発明を実施するための最良の形態

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。

図 1 6 は、図 1 5 の検査行列に、式 (8) の行置換と、式 (9) の列置換を施して得られる検査行列を示している。

行置換: $6x+y+1$ 行目 $\rightarrow 5y+x+1$ 行

目 $\dots (8)$

列置換: $6s+t+61$ 列目 $\rightarrow 5t+s+61$ 列

目 $\dots (9)$

20 但し、式 (8) および (9) において、 x, y, s, t は、それぞれ、 $0 \leq x < 5$, $0 \leq y < 6$, $0 \leq s < 5$, $0 \leq t < 6$ の範囲の整数である。

式 (8) の行置換によれば、6 で割って余りが 1 になる 1, 7, 13, 19, 25 行目を、それぞれ、1, 2, 3, 4, 5 行目に、6 で割って余りが 2 になる 2, 8, 14, 20, 26 行目を、それぞれ、6, 7, 8, 9, 10 行目に、という具合に置換が行われる。

また、式 (9) の列置換によれば、61 列目以降に対して、6 で割って余りが 1 になる 61, 67, 73, 79, 85 列目を、それぞれ、61, 62, 63,

6 4, 6 5 列目に、6 で割って余りが 2 になる 6 2, 6 8, 7 4, 8 0, 8 6 列目を、それぞれ、6 6, 6 7, 6 8, 6 9, 7 0 列目に、という具合に置換が行われる。

このようにして、図 1 5 の検査行列に対して、行と列の置換を行って得られた
5 行列(matrix)が、図 1 6 の検査行列である。

図 1 6 の検査行列（以下、適宜、置換検査行列という）に対して、図 1 5 の検査行列（以下、適宜、元の検査行列という）で表わされる符号（誤りのない元の符号）の符号語の系列に、式（9）と同一の置換を行ったものを乗じると、0 ベクトルが出力されることは自明である。即ち、元の検査行列を行列 H で、置換検査行列を行列 H' で、元の符号の符号語の系列を行ベクトル c で、行ベクトル c に式（9）の列置換を施して得られる行ベクトルを c' で、それぞれ表すこととすると、検査行列の性質から、 $H c^T$ （上付の T は転置を表す）は、0 ベクトルとなるから、 $H' c'^T$ も、当然、0 ベクトルとなる。

そして、以上のことから、図 1 6 の変換検査行列は、元の符号の符号語の系列
15 c に、式（9）の列置換を行ったものを符号語とする符号 c' の検査行列になっている。

従って、元の符号によって符号化されたデータを受信して復号する際に、受信した符号系列に、式（9）の列置換を行い、その列置換後の符号系列を、図 1 6 の変換検査行列に基づく復号装置を用いて復号し、復号結果の系列に、式（2）
20 の列置換の逆置換を行っても、元の符号の復号装置を用いた場合と復号結果に違いはないため、性能の劣化を招くことはないことになる。即ち、図 1 5 の元の検査行列に基づく復号装置は、図 1 6 の変換検査行列に基づく復号装置を用いて実現することができる。

次に、図 1 7 は、 5×5 の行列の単位に間隔を空けた、図 1 6 の変換検査行列
25 を示している。

図 1 7 においては、検査行列（変換検査行列）は、 5×5 の単位行列、その単位行列の 1 のうち 1 個以上が 0 になった行列（以下、適宜、準単位行列という）、

単位行列または準単位行列をサイクリックシフト(cyclic shift)した行列（以下、適宜、シフト行列という）、単位行列、準単位行列、またはシフト行列のうちの2以上の和（以下、適宜、和行列という）、 5×5 の0行列の組合わせで表わされている。

- 5 図17の検査行列は、 5×5 の単位行列、準単位行列、シフト行列、和行列、0行列で構成されているといふことができる。そこで、検査行列を構成する、これらの 5×5 の行列を、以下、適宜、構成行列という。

以上のような $P \times P$ の構成行列で表される検査行列で表される符号の復号には、チェックノードとバリアブルノードの計算を、 P 個同時に行うアーキテクチャ

- 10 (architecture)を用いることができる。

図18A乃至図18Cは、そのような復号装置の一実施の形態の構成例を示すブロック図である。即ち、図18A乃至図18Cは、図15の元の検査行列に対して、行または列置換を行って得られる図17の変換検査行列を用いて、LDPC符号の復号を行う復号装置の構成例を示している。

- 15 なお、図18Aは、復号装置全体の構成を示している。また、図18Bは、図18Aの復号装置の点線Bで囲まれた図中左部の詳細構成を示し、図18Cは、図18Aの復号装置の点線Cで囲まれた図中右部の詳細構成を示している。

- この復号装置は、6つのFIFO300₁乃至300₆からなる枝データ格納用メモリ300、FIFO300₁乃至300₆を選択するセレクタ301、チェックノード計算部302、2つのサイクリックシフト回路303および308、18個のFIFO304₁乃至304₁₈からなる枝データ格納用メモリ304、FIFO304₁乃至304₁₈を選択するセレクタ305、受信情報を格納する受信データ用メモリ306、バリアブルノード計算部307、復号語計算部309、受信データ並べ替え部310、復号データ並べ替え部311からなる。

- 25 この復号装置の各部について詳細に説明する前に、まず、枝データ格納用メモリ300と304へのデータの格納方法について説明する。

枝データ格納用メモリ 300 は、図 17 の変換検査行列の行数 30 を構成行列の行数 5 で除算した数である 6 つの $FIF0300_1$ 乃至 300_6 から構成されている。 $FIF0300_y$ ($y = 1, 2, \dots, 6$) は、構成行列の行数および列数である 5 つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようになっており、その長さ（段数）は、図 17 の変換検査行列の行方向の 1 の数（ハミング重み）の最大数である 9 になっている。

$FIF0300_1$ には、図 17 の検査行列（変換検査行列）の第 1 行目から第 5 行目までの 1 の位置に対応するデータが、各行共に横方向に詰めた形に（0 を無視した形で）格納される。すなわち、第 j 行第 i 列を、 (j, i) と表すこととすると、

10 $FIF0300_1$ の第 1 の要素（第 1 段）には、検査行列の $(1, 1)$ から $(5, 5)$ の 5×5 の単位行列の 1 の位置に対応するデータが格納される。第 2 の要素には、検査行列の $(1, 21)$ から $(5, 25)$ のシフト行列（ 5×5 の単位行列を右方向に 3 つだけサイクリックシフトしたシフト行列）の 1 の位置に対応するデータが格納される。第 3 から第 8 の要素も同様に検査行列と対応づけてデータが格納される。そして、

15 第 9 の要素には、検査行列の $(1, 86)$ から $(5, 90)$ のシフト行列（ 5×5 の単位行列のうちの 1 行目の 1 を 0 に置き換えて 1 つだけ左にサイクリックシフトしたシフト行列）の 1 の位置に対応するデータが格納される。ここで、検査行列の $(1, 86)$ から $(5, 90)$ のシフト行列においては、1 行目に 1 がいないため、 $FIF0300_1$ の 1 行目のみ要素数は 8、残りの行は要素数が 9 となる。

20 $FIF0300_2$ には、図 17 の検査行列の第 6 行目から第 10 行目までの 1 の位置に対応するデータが格納される。すなわち、 $FIF0300_2$ の第 1 の要素には、検査行列の $(6, 1)$ から $(10, 5)$ の和行列（ 5×5 の単位行列を右に 1 つだけサイクリックシフトした第 1 のシフト行列と、右に 2 つだけサイクリックシフトした第 2 のシフト行列の和である和行列）を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納される。また、第 2 の要素には、検査行列の $(6, 1)$ から

25 $(10, 5)$ の和行列を構成する第 2 のシフト行列の 1 の位置に対応するデータが格納される。

- 即ち、重みが2以上の構成行列については、その構成行列を、重みが1である $P \times P$ の単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位
- 5 行列、またはシフト行列の1の位置に対応するデータ（単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ）は、同一アドレス（FIFO 300_1 乃至 300_6 のうちの同一の FIFO）に格納される。

以下、第3から第9の要素についても、検査行列に対応づけてデータが格納される。FIFO 300_2 は全行共に要素数は9となる。

- 10 FIFO 300_3 乃至 300_6 も同様に検査行列に対応づけてデータを格納し、各 FIFO 300_3 乃至 300_6 それぞれの長さは9である。

- 枝データ格納用メモリ 304 は、検査行列の列数90を、構成行列の列数である5で割った18個の FIFO 304_1 乃至 304_{18} から構成されている。FIFO 304_x ($x = 1, 2, \dots, 18$) は、構成行列の行数および列数である5つの
- 15 枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようになっている。

- FIFO 304_1 には、図17の検査行列の第1列目から第5列目までの1の位置に対応するデータが、各列共に縦方向に詰めた形に（0を無視した形で）格納される。すなわち、FIFO 304_1 の第1の要素（第1段）には、検査行列の(1, 1)
- 20 から(5, 5)の 5×5 の単位行列の1の位置に対応するデータが格納される。第2の要素には、検査行列の(6, 1)から(10, 5)の和行列（ 5×5 の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列との和である和行列）を構成する第1のシフト行列の1の位置に対応するデータが格納される。また、第3の要素には、検査行列の
- 25 (6, 1)から(10, 5)の和行列を構成する第2のシフト行列の1の位置に対応するデータが格納される。

即ち、重みが 2 以上の構成行列については、その構成行列を、重みが 1 である $P \times P$ の単位行列、そのコンポーネントである 1 のうち 1 個以上が 0 になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが 1 の単位行列、準単位
 5 行列、またはシフト行列の 1 の位置に対応するデータ（単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ）は、同一アドレス（FIFO 304_1 乃至 304_{18} のうちの同一の FIFO）に格納される。

以下、第 4 および第 5 の要素についても、検査行列に対応づけて、データが格納される。この FIFO 304_1 の要素数（段数）は、検査行列の第 1 列から第 5 列
 10 における行方向の 1 の数（ハミング重み）の最大数である 5 になっている。

FIFO 304_2 と 304_3 も同様に検査行列に対応づけてデータを格納し、それぞれの長さ（段数）は、5 である。FIFO 304_4 乃至 304_{12} も同様に検査行列に対応づけてデータを格納し、それぞれの長さは 3 である。FIFO 304_{13} 乃至 304_{18} も同様に検査行列に対応づけてデータを格納し、それぞれの長さは 2 であ
 15 る。但し、FIFO 304_{18} の第 1 の要素は、検査行列の (1, 86) から (5, 90) に相当し、第 5 列目（検査行列の (1, 90) から (5, 90)）に 1 がいないため、データは格納されない。

以下、図 18 A 乃至図 18 C の復号装置の各部の動作について詳細に説明する。

枝データ格納用メモリ 300 は、6 つの FIFO 300_1 乃至 300_6 からなり、
 20 前段のサイクリックシフト回路 308 から供給される 5 つのメッセージデータ D311 が、検査行列どの行に属するかの情報（Matrix データ）D312 に従って、データを格納する FIFO を、FIFO 300_1 乃至 300_6 の中から選び、選んだ FIFO に 5 つのメッセージデータ D311 をまとめて順番に格納していく。また、枝データ格納用メモリ 300 は、データを読み出す際には、FIFO 300_1 から 5 つのメ
 25 ッセージデータ D300₁ を順番に読み出し、次段のセクタ 301 に供給する。枝データ格納用メモリ 300 は、FIFO 300_1 からのメッセージデータの読み出し

の終了後、FIFO 300₂乃至300₆からも、順番に、メッセージデータを読み出し、セクタ301に供給する。

セクタ301は、セレクト信号D301に従って、FIFO 300₁乃至300₆のうちの、現在データが読み出されているFIFOからの5つのメッセージデータを
5 選択し、メッセージデータD302として、チェックノード計算部302に供給する。

チェックノード計算部302は、5つのチェックノード計算器302₁乃至302₅からなり、セクタ301を通して供給されるメッセージD302(D302₁乃至D302₅)を用いて、式(7)に従って演算を行い、その演算の結果得られる5つの
10 メッセージD303(D303₁乃至D303₅)をサイクリックシフト回路303に供給する。

ここで、チェックノード計算器302₁乃至302₅それぞれは、図10に示したチェックノード計算器101と同様に構成される。

サイクリックシフト回路303は、チェックノード計算部302で計算された5つのメッセージD303₁乃至D303₅を、対応する枝が検査行列において元となる
15 単位行列を幾つサイクリックシフトしたものであるかの情報(Matrixデータ)D305を元にサイクリックシフトし、その結果をメッセージD304として、枝データ格納用メモリ304に供給する。

枝データ格納用メモリ304は、18個のFIFO 304₁乃至304₁₈からなり、前段のサイクリックシフト回路303から供給される5つのメッセージデータ
20 D304が検査行列のどの行に属するかの情報D305に従って、データを格納するFIFOを、FIFO 304₁乃至304₁₈の中から選び、選んだFIFOに5つのメッセージデータD304をまとめて順番に格納していく。また、枝データ格納用メモリ304は、データを読み出す際には、FIFO 304₁から5つのメッセージD306₁を順番に読み出し、次段のセクタ305に供給する。枝データ格納用メモリ3
25 04は、FIFO 304₁からのデータの読み出しの終了後、FIFO 304₂乃至304₁₈からも、順番に、メッセージデータを読み出し、セクタ305に供給する。

セクタ 305 は、セレクト信号 D307 に従って、FIFO 304₁ 乃至 304₁₈ のうちの、現在データが読み出されている FIFO からの 5 つのメッセージデータを選択し、メッセージデータ D308 として、バリアブルノード計算部 307 と復号語計算部 309 に供給する。

- 5 一方、受信データ並べ替え部 310 は、通信路を通して受信した LDPC 符号の符号系列（受信データ）D313 を、式（9）の列置換を行うことにより並べ替え、符号系列 D314 として、受信用データメモリ 306 に供給する。受信データ用メモリ 306 は、受信データ並べ替え部 310 から供給される符号系列 D314 から、受信 LLR（対数尤度比）を計算しており、その計算した受信 LLR を 5 つまとめて
- 10 データ D309 として、バリアブルノード計算部 307 と復号語計算部 309 に供給する。

- バリアブルノード計算部 307 は、5 つのバリアブルノード計算器 307₁ 乃至 307₅ からなり、セクタ 305 を通して供給されるメッセージ D308 (D308₁ 乃至 D308₅) と、受信データ用メモリ 306 から供給される 5 つの受信 LLR D309
- 15 を用いて、式（1）に従って演算を行い、その演算の結果得られるメッセージ D310 (D310₁ 乃至 D310₅) を、サイクリックシフト回路 308 に供給する。

ここで、バリアブルノード計算器 307₁ 乃至 307₅ それぞれは、図 11 のバリアブルノード計算器 103 と同様に構成される。

- サイクリックシフト回路 308 は、バリアブルノード計算部 307 で計算されたメッセージ D310₁ 乃至 D310₅ を、対応する枝が検査行列において元となる単位
- 20 行列を幾つサイクリックシフトしたものであるかの情報を元にサイクリックシフトし、その結果をメッセージ D311 として、枝データ格納用メモリ 300 に供給する。

以上の動作を 1 巡することで、LDPC 符号の 1 回の復号を行うことができる。

- 25 図 18A 乃至図 18C の復号装置は、所定の回数だけ LDPC 符号を復号した後、復号語計算部 309 および復号データ並べ替え部 311 において、最終的な復号結果を求めて出力する。

即ち、復号語計算部 309 は、5つの復号語計算器 309₁乃至309₅からなり、セクタ 305 が出力する 5つのメッセージ D308 (D308₁乃至 D308₅)と、受信データ用メモリ 306 から供給される 5つの受信 LLR D309 を用い、複数の復号の最終段において、式 (5) に基づいて、復号結果 (復号語) を計算して、

- 5 その結果得られる復号データ D315 を、復号データ並べ替え部 311 に供給する。

復号データ並べ替え部 311 は、復号語計算部 309 から供給される復号データ D315 を対象に、式 (9) の列置換の逆置換を行うことにより、その順序を並べ替え、最終的な復号結果 D316 として出力する。

- 10 なお、枝データ (枝に対応するメッセージ) が欠けている箇所に関しては、メモリ格納時 (枝データ格納用メモリ 300 と 304 へのデータ格納時) には、何のメッセージも格納せず、また、ノード演算時 (チェックノード計算部 302 でのチェックノード演算時とバリアブルノード計算部 307 でのバリアブルノード演算時) にも何の演算も行わない。

- 15 また、サイクリックシフト回路 303 および 308 には、バレルシフタを用いると回路規模を小さくしながら所望の操作を実現できる。

- 20 上記説明には、枝データ格納に FIFO を用いたが (枝データ格納メモリ 300 と 304 を FIFO で構成するようにしたが)、FIFO の代わりに RAM を用いても構わない。その場合、RAM には、P 個の枝情報 (枝に対応するメッセージ) を同時に読み出すことの出来るビット幅と、枝総数/P のワード (word) 数が必要となる。さらに、RAM への書き込みは、検査行列の情報から、書き込もうとしているデータが次に読み出される際に何番目に読み出されるかを求め、その位置に書き込む。また、RAM からの読み出しの際には、アドレスの先頭から順次データを読み出す。FIFO の代わりに RAM を用いると、セクタ 301 および 305 は不要になる。

- 25 なお、FIFO や RAM の物理的なビット幅が足りない場合には、複数の RAM を用いて同じ制御信号を与えることで、論理的に 1つの RAM とみなすことができる。

また、上述の場合には、説明を簡単にするために、 P が 5 の場合、即ち、検査行列を構成する構成行列の行数および列数が 5 の場合を例に挙げたが、構成行列の行数および列数 P は必ずしも 5 である必要はなく、検査行列によって異なる値を取ることもあり得る。例えば、 P は 360 や 392 であってもよい。

5 また、本実施の形態では、符号長 90、符号化率 $2/3$ の LDPC 符号を用いたが、LDPC 符号の符号長や符号化率は、幾つであっても構わない。例えば、構成行列の行数および列数 P が 5 の場合、枝総数が 5 以下であれば、どんな符号長、符号化率の LDPC 符号でも、制御信号を代えるだけで、図 18A 乃至図 18C の復号装置を用いて復号可能である。

10 さらに、構成行列の行数および列数 P が所定の値で、枝の総数がある値以下、という条件を満たすある LDPC 符号の復号装置は、その条件を満たす、任意の符号長で、任意の符号化率の LDPC 符号を復号することができる。

以上のように、検査行列（元の検査行列）に対して、行置換と列置換うちの一方または両方を施し、 $P \times P$ の単位行列、そのコンポーネントの 1 のうち 1 個以上が 0 になった準単位行列、単位行列もしくは準単位行列をサイクリックシフトしたシフト行列、単位行列、準単位行列、もしくはシフト行列の複数の和である和行列、 $P \times P$ の 0 行列の組合せ、つまり、構成行列の組み合わせで表わすこと

15 ができる検査行列（変換検査行列）に変換することで、LDPC 符号の復号を、チェックノードとバリアブルノードの演算を P 個同時に行うアーキテクチャ

20 (architecture)を採用することが可能となり、これにより、ノード演算を、 P 個同時に行うことで動作周波数を実現可能な範囲に抑えることができ、多数の繰り返し復号を行うことを可能にしつつ、メモリ（FIFO や RAM）への書き込みと読み出し時に、異なるアドレスへの同時アクセスが起きることを防止することができる。

25 即ち、上述した検査行列の行置換と列置換の双方もしくは片方を行うことで、チェックノードとバリアブルノードの演算を P 個同時に行うことが可能となり、さらに、このようにノード演算を P 個同時に行うことで、動作周波数を実現可

能な範囲に抑えることができ、多数の繰り返し復号を行うことを可能にしつつ、メモリ（FIFO や RAM）への書き込みと読み出し時に、異なるアドレスへの同時アクセスが起きることを防止することができる。

- さらに、図 1 7 の検査行列（変換検査行列）で表わされる LDPC 符号を復号する場合には、269 個の枝をチェックノード、バリエブルノード毎に 5 個ずつ演算することが可能であることから、1 回の復号に、 $269/5 \times 2 \div 108$ クロック動作すればよいことになる。50 回の復号には、90 個の符号情報を受信する間に、 $108 \times 50 = 5400$ クロック動作すればよいことになり、受信周波数の約 60 倍の動作周波数でよいことになる。従って、図 1 8 A 乃至図 1 8 C の復号装置によれば、各ノード演算を一つずつ行う図 9 の復号装置に比べて、1/5 の動作周波数で済むことになる。また、回路規模の面から見ても、メモリの大きさは同じであるため、論理回路が多少大きくなっても全体への影響は小さいと言える。

- 一般的に、LDPC 符号は符号長が数千から数万と大きいので、P の値も数百の大きさを持つものが使われる。その場合には、更に本発明に係る復号装置を用いる効果は大きくなる。

以上のように、ロジック、メモリ共に回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑えることができ、メモリアクセスの制御も容易に行うことができる。

- また、LDPC 符号の性質から、検査行列に行置換や列置換を施しても符号の性能は変わらない。従って、行置換または列置換によって、構成行列の組み合わせで表すことができる変換検査行列を得られる検査行列に対応する LDPC 符号については、どのような符号長、符号化率の LDPC 符号であっても、性能の劣化を招くことなく、実装が容易で、かつ効率の良い復号を行うことができる。

- さらに、本発明に係る復号装置は、サンプロダクトアルゴリズムを忠実に実装するものであるため、メッセージの量子化以外の復号損失が起きることはない。

以上の観点から、本発明に係る復号装置を用いることで、高性能な復号が可能になる。

なお、検査行列が、構成行列の行数および列数 P の倍数でない場合は、検査行列の端数の外側にすべて 0 (all 0) の成分を付けて P の倍数とみなして適用できることがある。

- 次に、上述した一連の処理は、ハードウェアにより行うこともできるし、ソフトウェアにより行うこともできる。一連の処理をソフトウェアによって行う場合には、そのソフトウェアを構成するプログラムが、汎用のコンピュータ等にインストールされる。

そこで、図 19 は、上述した一連の処理を実行するプログラムがインストールされるコンピュータの一実施の形態の構成例を示している。

- 10 プログラムは、コンピュータに内蔵されている記録媒体としてのハードディスク 405 や ROM 403 に予め記録しておくことができる。

- あるいはまた、プログラムは、フレキシブルディスク、CD-ROM (Compact Disc Read Only Memory)、MO (Magneto Optical) ディスク、DVD (Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体 411 に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバブル記録媒体 411 は、いわゆるパッケージソフトウェアとして提供することができる。

- なお、プログラムは、上述したようなリムーバブル記録媒体 411 からコンピュータにインストールする他、ダウンロードサイトから、デジタル衛星放送用の人工衛星を介して、コンピュータに無線で転送したり、LAN (Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを、通信部 408 で受信し、内蔵するハードディスク 405 にインストールすることができる。

- 25 コンピュータは、CPU (Central Processing Unit) 402 を内蔵している。CPU 402 には、バス 401 を介して、入出力インタフェース 410 が接続されており、CPU 402 は、入出力インタフェース 410 を介して、ユーザによって、

キーボードや、マウス、マイク等で構成される入力部 4 0 7 が操作等されることにより指令が入力されると、それにしたがって、ROM(Read Only Memory) 4 0 3 に格納されているプログラムを実行する。あるいは、また、CPU 4 0 2 は、ハードディスク 4 0 5 に格納されているプログラム、衛星若しくはネットワークから
5 転送され、通信部 4 0 8 で受信されてハードディスク 4 0 5 にインストールされたプログラム、またはドライブ 4 0 9 に装着されたリムーバブル記録媒体 4 1 1 から読み出されてハードディスク 4 0 5 にインストールされたプログラムを、RAM(Random Access Memory) 4 0 4 にロードして実行する。これにより、CPU 4 0 2 は、上述したフローチャートにしたがった処理、あるいは上述したブロック
10 図の構成により行われる処理を行う。そして、CPU 4 0 2 は、その処理結果を、必要に応じて、例えば、入出力インタフェース 4 1 0 を介して、LCD(Liquid Crystal Display)やスピーカ等で構成される出力部 4 0 6 から出力、あるいは、通信部 4 0 8 から送信、さらには、ハードディスク 4 0 5 に記録等させる。

ここで、本明細書において、コンピュータに各種の処理を行わせるためのプログラムを記述する処理ステップは、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいは個別に実行される処理
15 (例えば、並列処理あるいはオブジェクトによる処理)も含むものである。

また、プログラムは、1のコンピュータにより処理されるものであっても良いし、複数のコンピュータによって分散処理されるものであっても良い。さらに、
20 プログラムは、遠方のコンピュータに転送されて実行されるものであっても良い。

産業上の利用可能性

以上の如く、本発明によれば、回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことが可能となる。

請求の範囲

1. LDPC (Low Density Parity Check) 符号の復号方法であって、
元の検査行列に対して、行置換と列置換のうち的一方または両方を行って得られる変換検査行列を用いて、前記 LDPC 符号を復号する復号ステップを備える
5 ことを特徴とする復号方法。
2. 請求の範囲第 1 項に記載の復号方法であって、
 $P \times P$ の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上
が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイク
リックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくは
10 シフト行列のうちの複数の和である和行列、または $P \times P$ の 0 行列を構成行列と
して、前記変換検査行列は、複数の前記構成行列の組合せで表される
ことを特徴とする復号方法。
3. 請求の範囲第 1 項に記載の復号方法であって、
受信した前記 LDPC 符号の符号系列に対して、前記元の検査行列に対して行わ
15 れた列置換と同一の列置換を行い、置換符号系列を出力する符号系列置換ステッ
プをさらに備え、
前記復号ステップにおいて、前記変換検査行列と、前記置換符号系列とを用い
て、前記符号系列を復号する
ことを特徴とする復号方法。
- 20 4. LDPC (Low Density Parity Check) 符号の復号装置であって、
元の検査行列に対して、行置換と列置換のうち的一方または両方を行って得ら
れる変換検査行列を用いて、前記 LDPC 符号を復号する復号手段を備える
ことを特徴とする復号装置。
5. 請求の範囲第 4 項に記載の復号装置であって、
25 $P \times P$ の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上
が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイク
リックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくは

シフト行列のうちの複数の和である和行列、または $P \times P$ の 0 行列を構成行列として、前記変換検査行列は、複数の前記構成行列の組合せで表される

ことを特徴とする復号装置。

6. 請求の範囲第 5 項に記載の復号装置であって、

5 前記復号手段は、

前記 LDPC 符号の復号のための P 個のチェックノードの演算を同時に行うチェックノード計算手段と、

前記 LDPC 符号の復号のための P 個のバリエブルノードの演算を同時に行うバリエブルノード計算手段と

10 を有する

ことを特徴とする復号装置。

7. 請求の範囲第 6 項に記載の復号装置であって、

前記チェックノード計算手段は、チェックノードの演算を行う P 個のチェックノード計算器を有し、

15 前記バリエブルノード計算手段は、バリエブルノードの演算を行う P 個のバリエブルノード計算器を有する

ことを特徴とする復号装置。

8. 請求の範囲第 6 項に記載の復号装置であって、

前記復号手段は、前記 P 個のチェックノードの演算、または前記 P 個のバリエブルノードの演算の結果得られる P 個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段をさらに有する

ことを特徴とする復号装置。

9. 請求の範囲第 8 項に記載の復号装置であって、

前記メッセージ記憶手段は、チェックノード演算時に読み出される枝に対応するメッセージデータを、前記変換検査行列の 1 を行方向に詰めるように格納する

ことを特徴とする復号装置。

10. 請求の範囲第 8 項に記載の復号装置であって、

前記メッセージ記憶手段は、バリアブルノード演算時に読み出される枝に対応するメッセージデータを、前記変換検査行列の 1 を列方向に詰めるように格納する

ことを特徴とする復号装置。

- 5 1 1. 請求の範囲第 8 項に記載の復号装置であって、

前記メッセージ記憶手段は、前記変換検査行列を表す構成行列のうちの、重みが 2 以上の構成行列について、その構成行列を、重みが 1 の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが 1 の単位行列、準単位行列、またはシフト行列に属する P 個の枝に対応するメッセージを、同一

- 10 のアドレスに格納する

ことを特徴とする復号装置。

- 1 2. 請求の範囲第 8 項に記載の復号装置であって、

前記メッセージ記憶手段は、行数/P 個の FIFO と、列数/P 個の FIFO とで構成され、

- 15 前記行数/P 個の FIFO と列数/P 個の FIFO は、それぞれ、前記検査行列の行と列の重みに対応するワード数を有する

ことを特徴とする復号装置。

- 1 3. 請求の範囲第 8 項に記載の復号装置であって、

前記メッセージ記憶手段は、RAM(Random Access Memory)で構成され、

- 20 前記 RAM は、前記メッセージデータを、読み出される順番に詰めて格納し、格納位置順に読み出す

ことを特徴とする復号装置。

- 1 4. 請求の範囲第 6 項に記載の復号装置であって、

前記復号手段は、受信情報を格納するとともに、P 個の前記受信情報を同時に

- 25 読み出す受信情報記憶手段をさらに有する

ことを特徴とする復号装置。

- 1 5. 請求の範囲第 1 4 項に記載の復号装置であって、

前記受信情報記憶手段は、前記受信情報を、前記バリエブルノードの演算に必要となる順番に読み出すことができるように格納する

ことを特徴とする復号装置。

16. 請求の範囲第6項に記載の復号装置であって、

- 5 前記復号手段は、前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得られるメッセージをサイクリックシフトするサイクリックシフト手段をさらに有する

ことを特徴とする復号装置。

17. 請求の範囲第16項に記載の復号装置であって、

- 10 前記サイクリックシフト手段は、バレルシフタで構成される
ことを特徴とする復号装置。

18. 請求の範囲第4項に記載の復号装置であって、

受信した前記LDPC符号の符号系列に対して、前記元の検査行列に対して行われた列置換と同一の列置換を行い、置換符号系列を出力する符号系列置換手段を
15 さらに備え、

前記復号手段は、前記変換検査行列と、前記置換符号系列とを用いて、前記符号系列を復号する

ことを特徴とする復号装置。

19. 請求の範囲第18項に記載の復号装置であって、

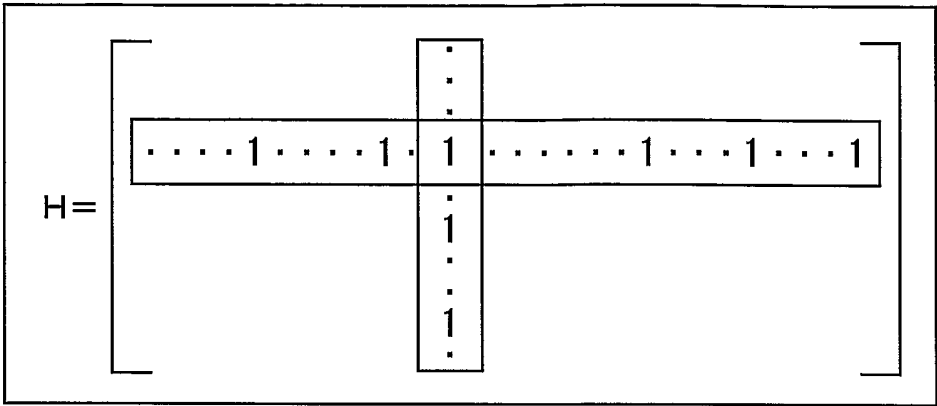
- 20 前記復号手段の出力に対して、前記元の検査行列に対して行われた列置換の逆置換を行い、最終的な復号結果を出力する逆置換手段をさらに備える

ことを特徴とする復号装置。

20. LDPC(Low Density Parity Check)符号の復号をコンピュータに行わせるプログラムであって、

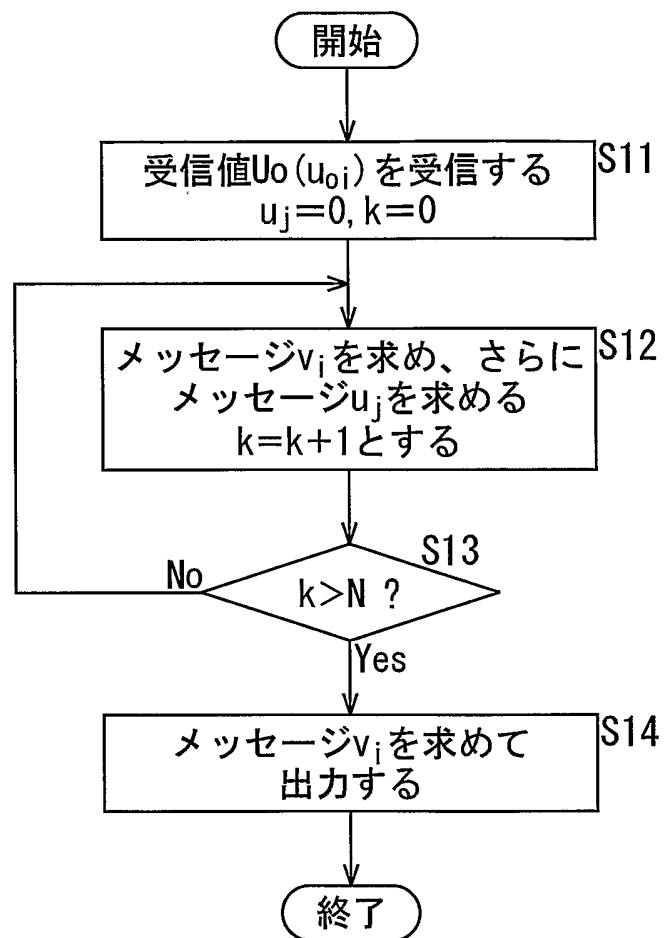
- 25 元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、前記LDPC符号を復号する復号ステップを備える
ことを特徴とするプログラム。

図 1



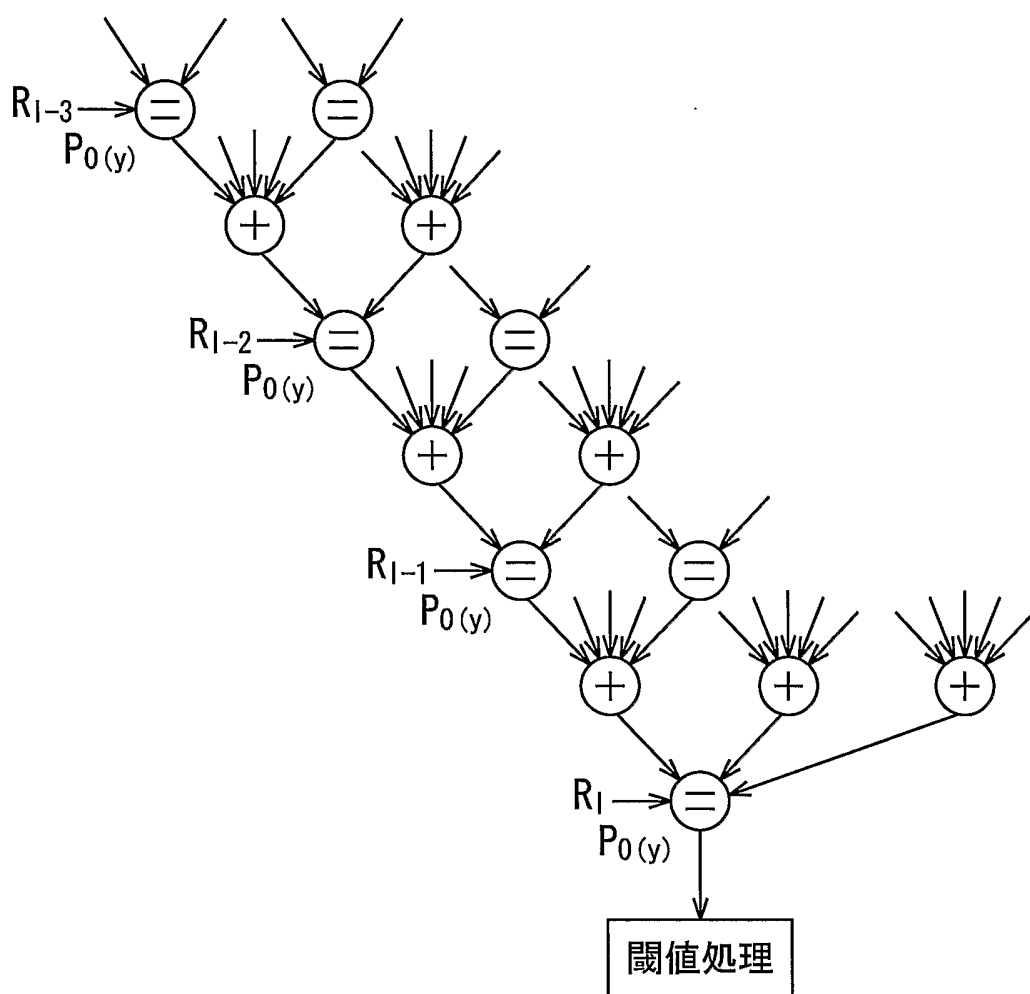
2/22

図 2



3/22

図 3



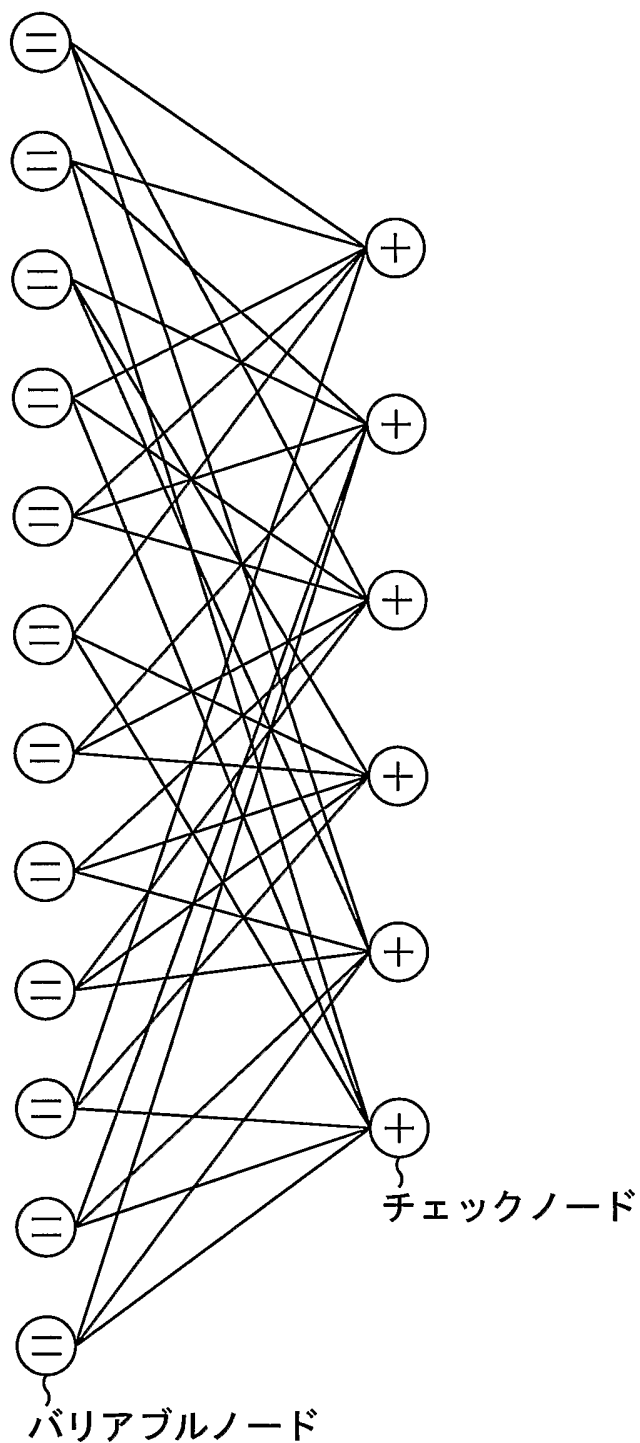
4/22

図 4

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

5/22

図 5



6/22

図 6

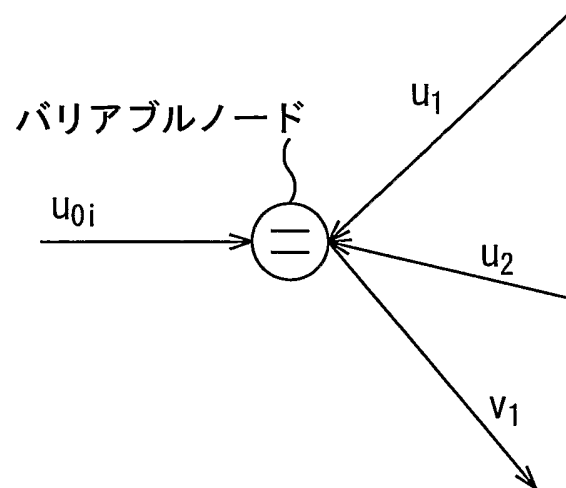
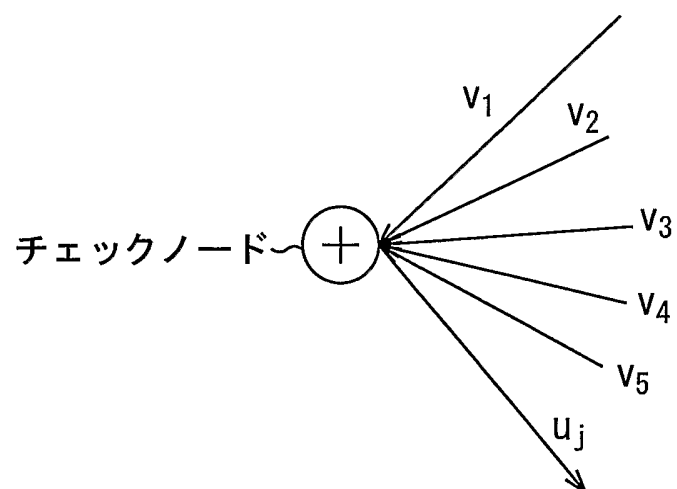


図 7



7/22

図 8

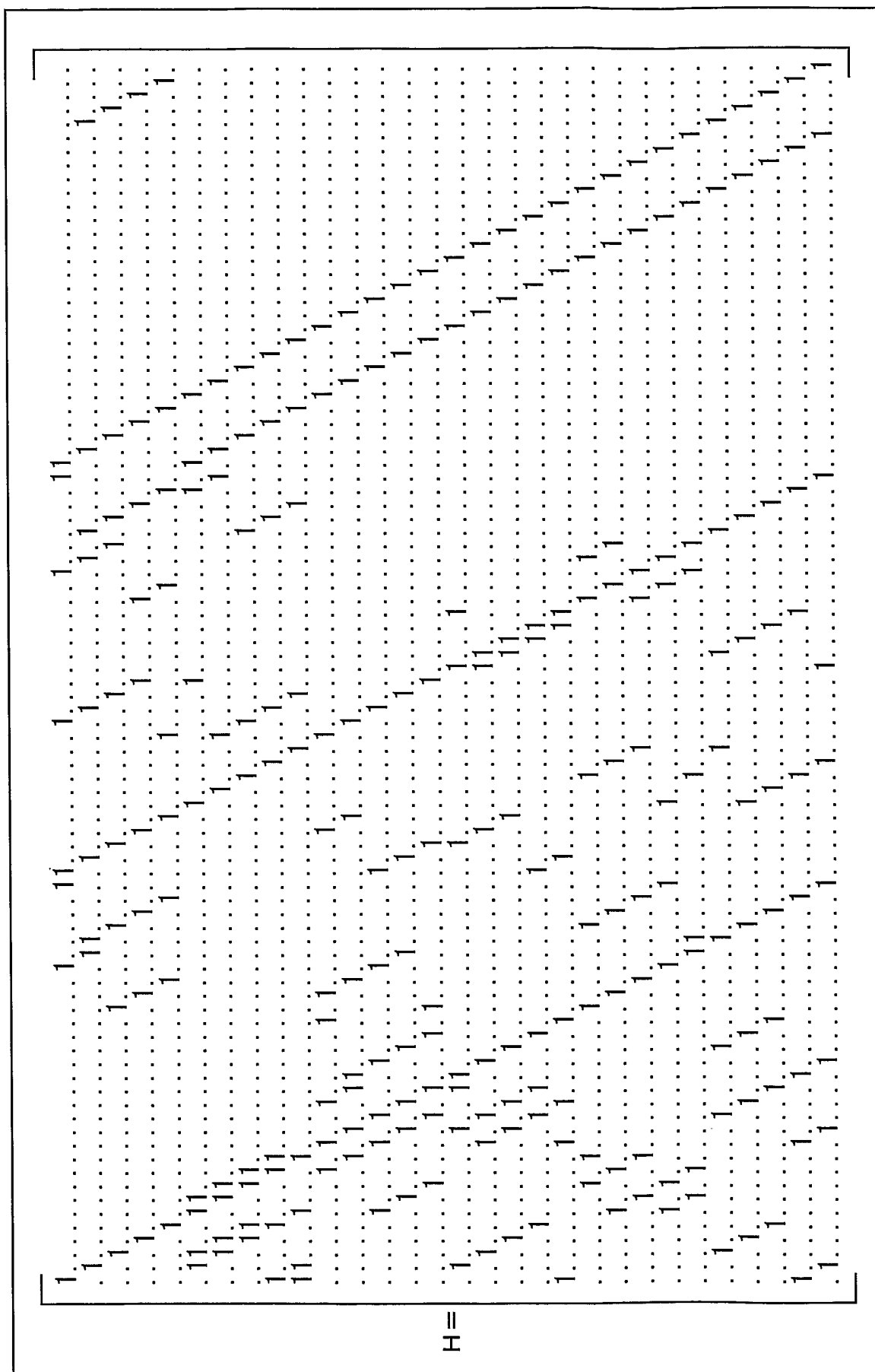
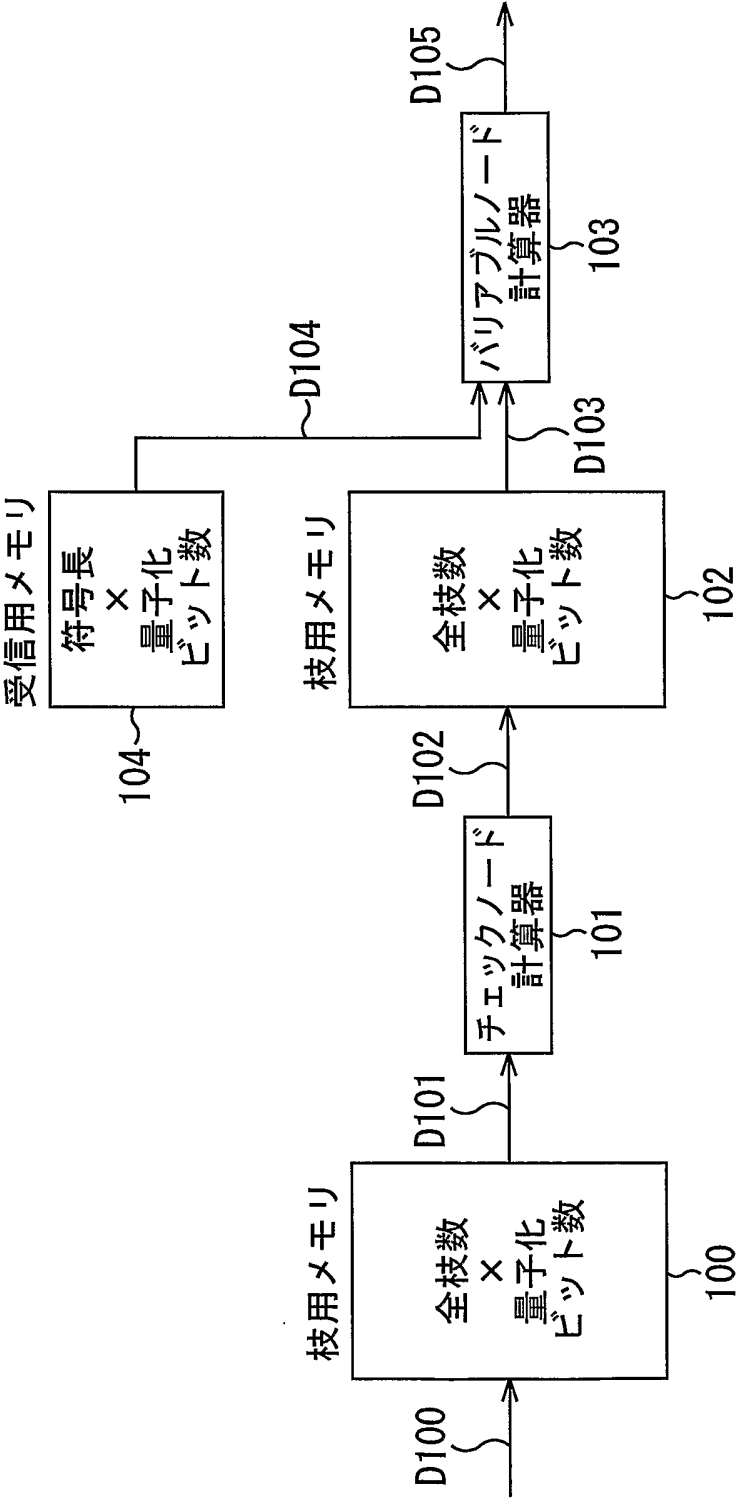
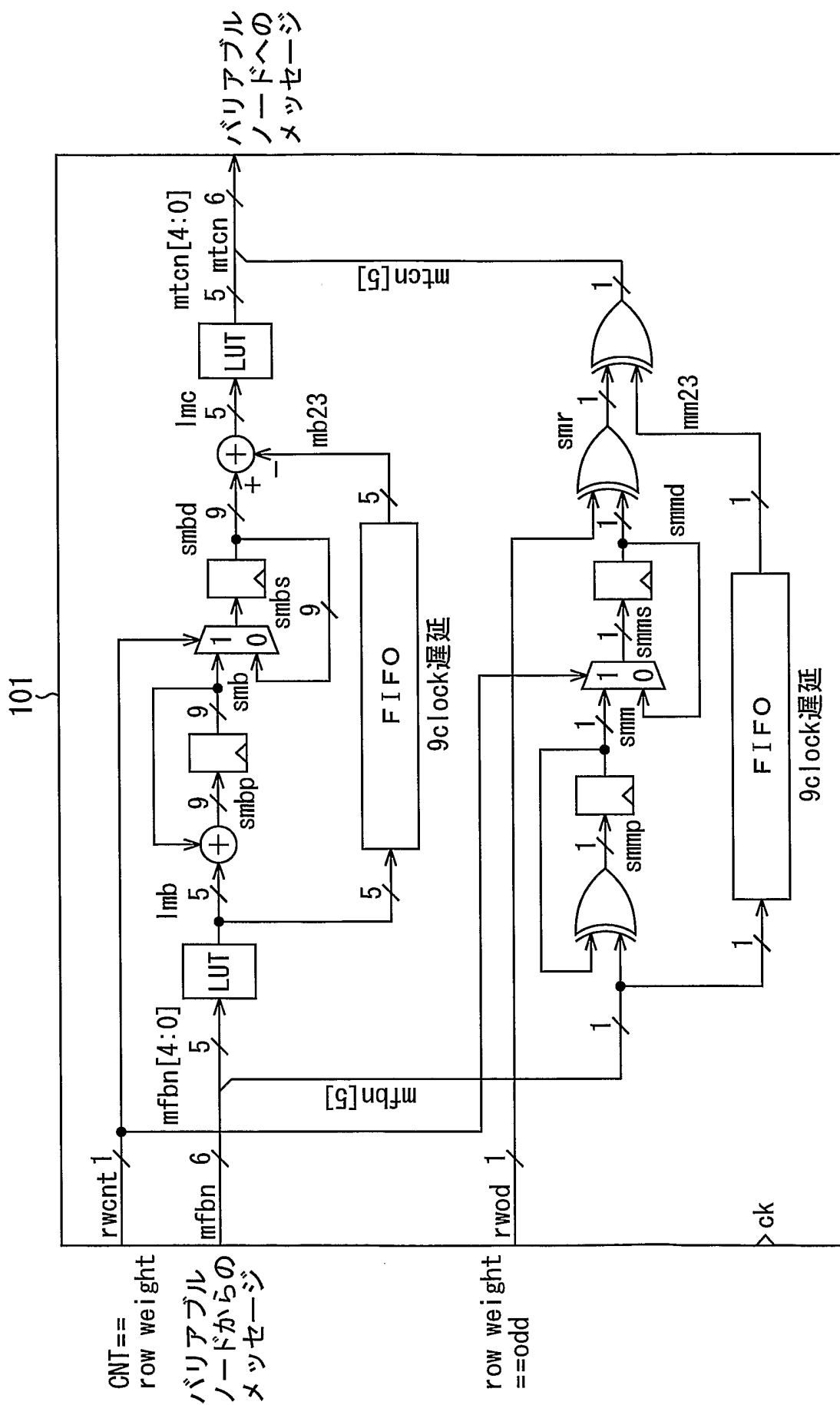


図9



10



二
[X]

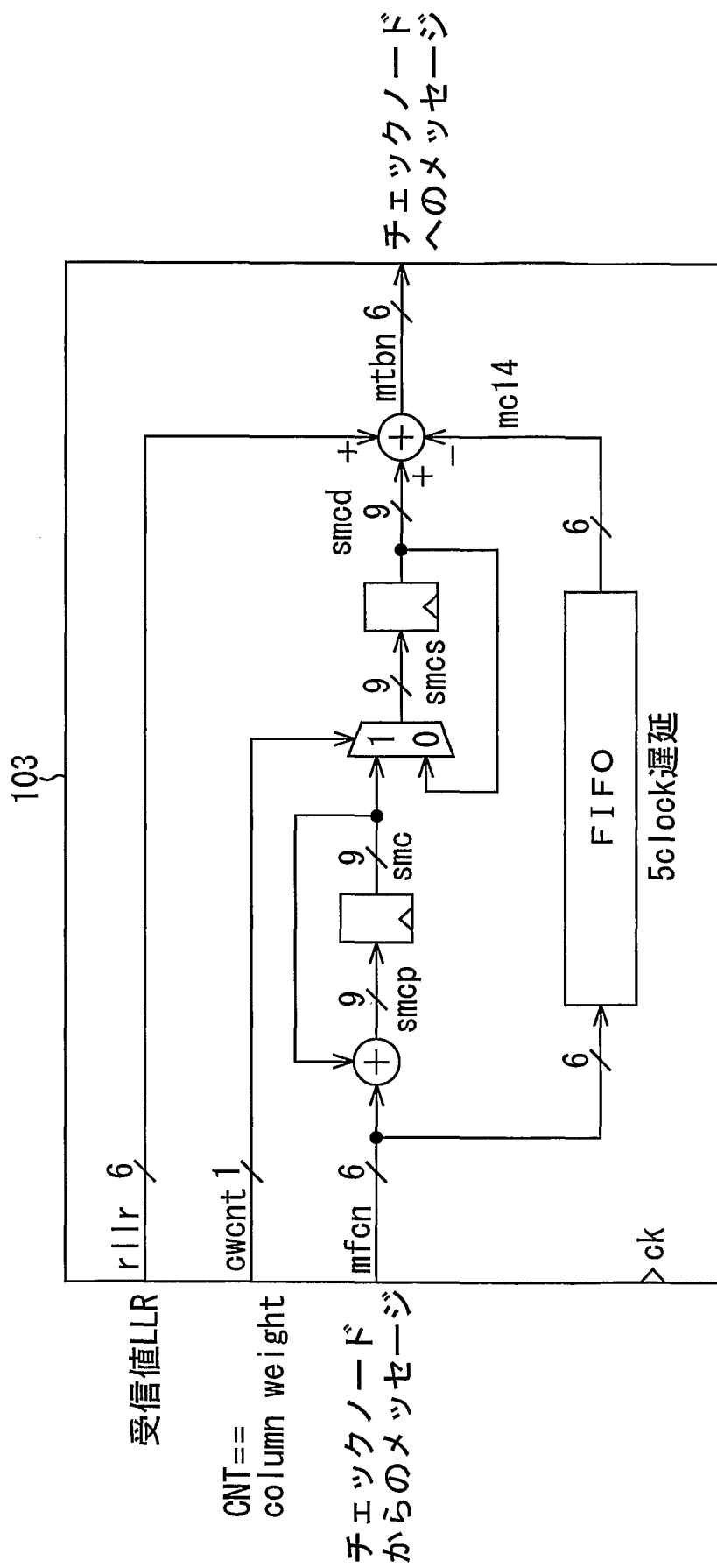


図12A

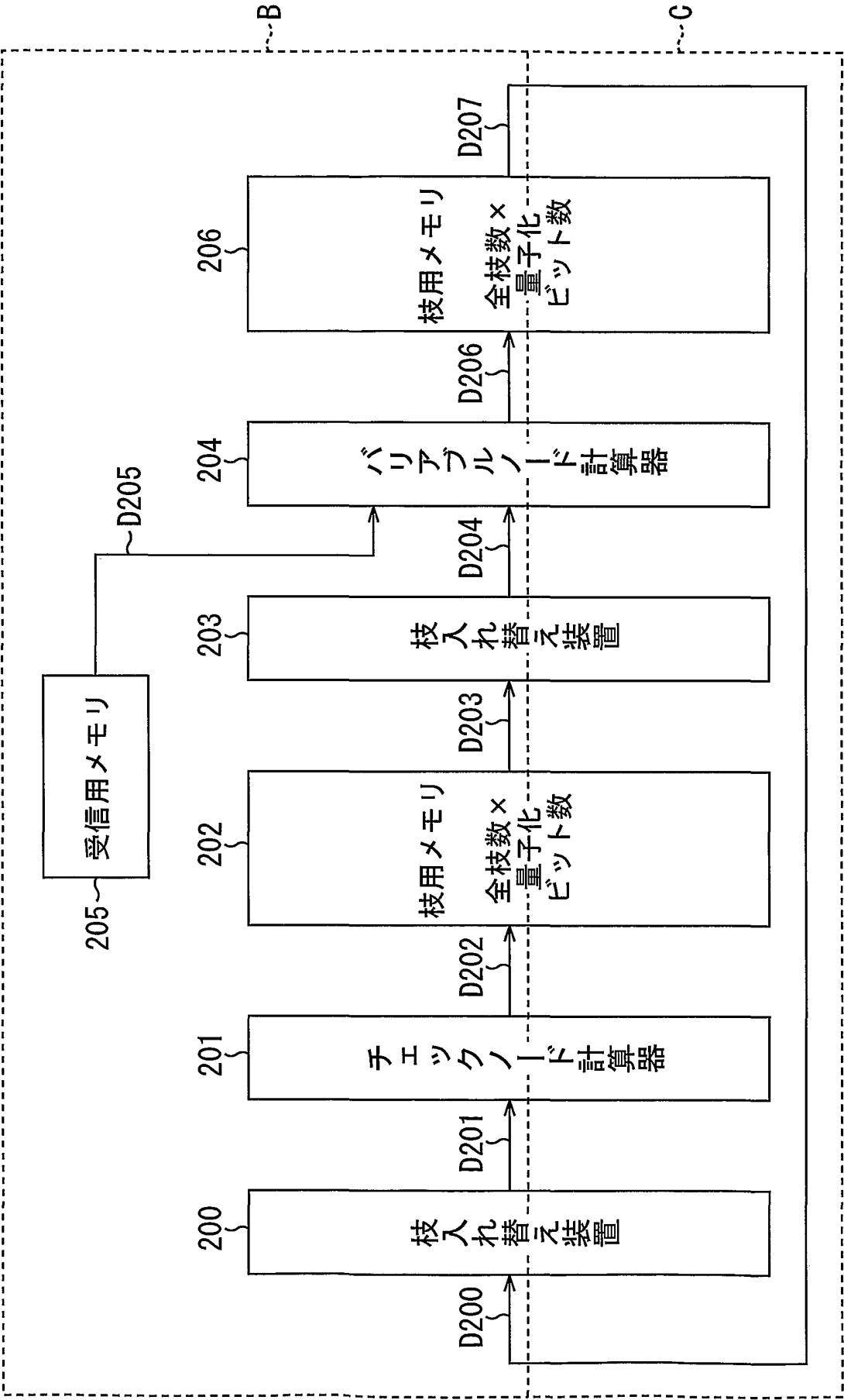


図12B

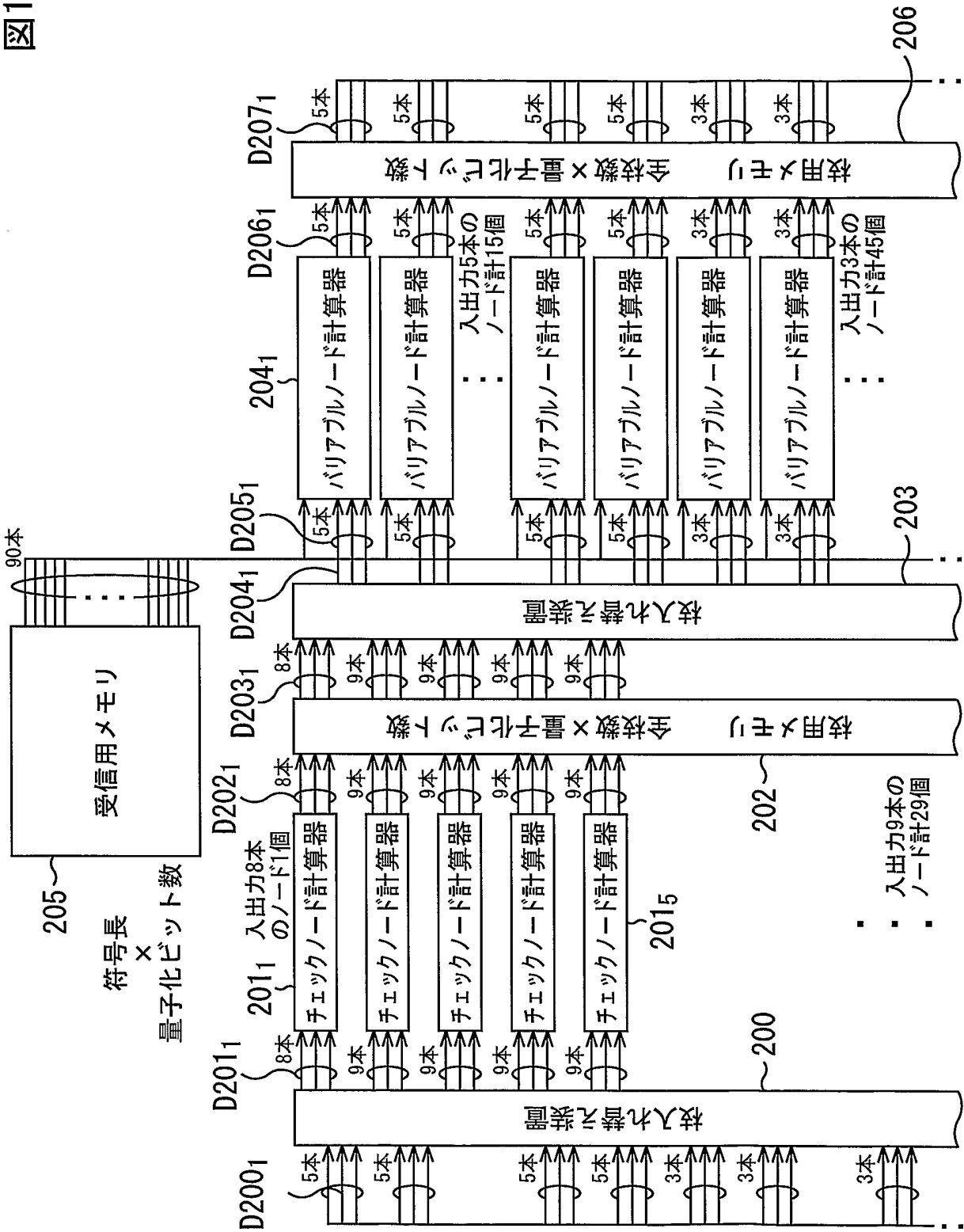


図12C

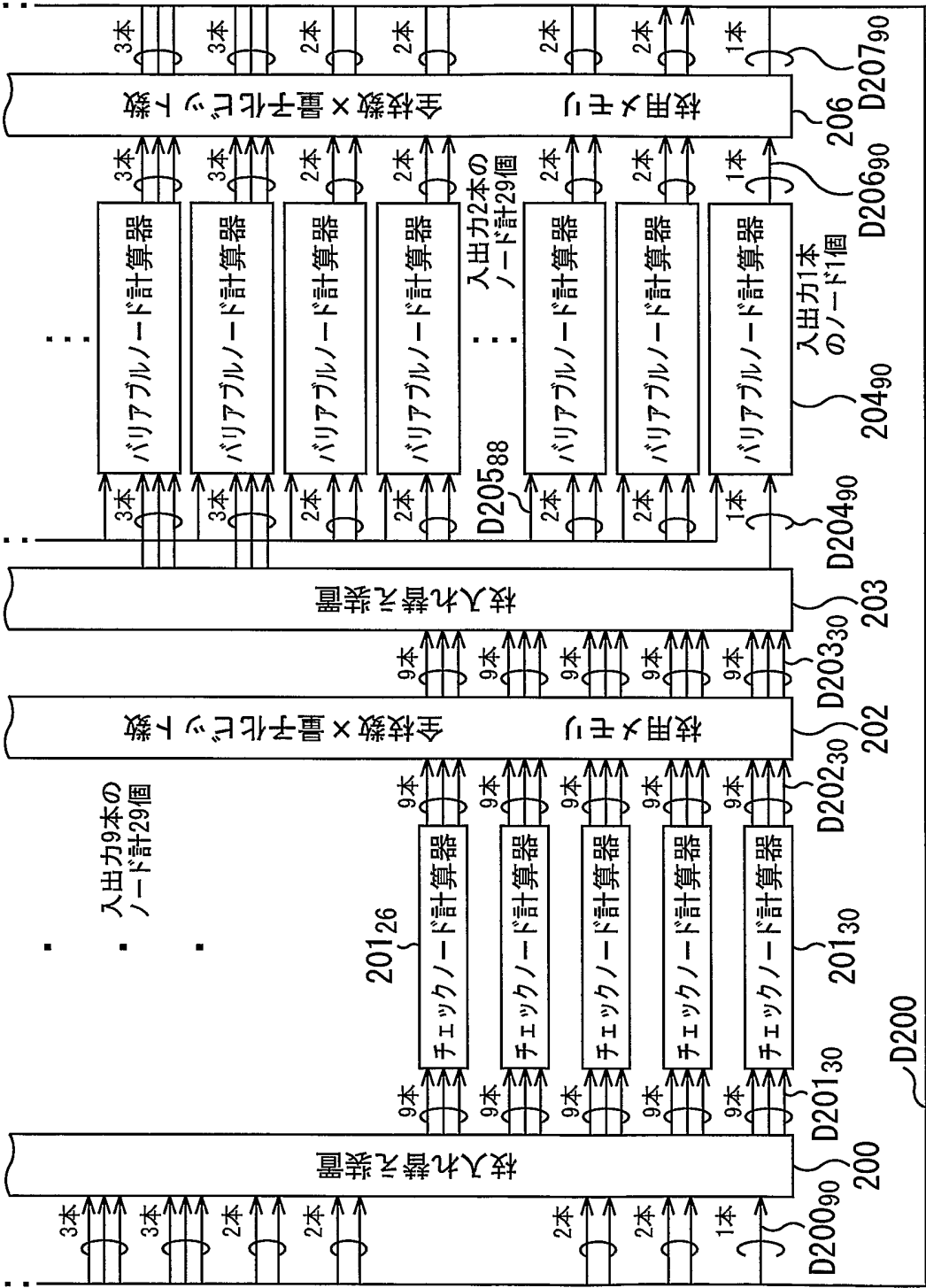


図13

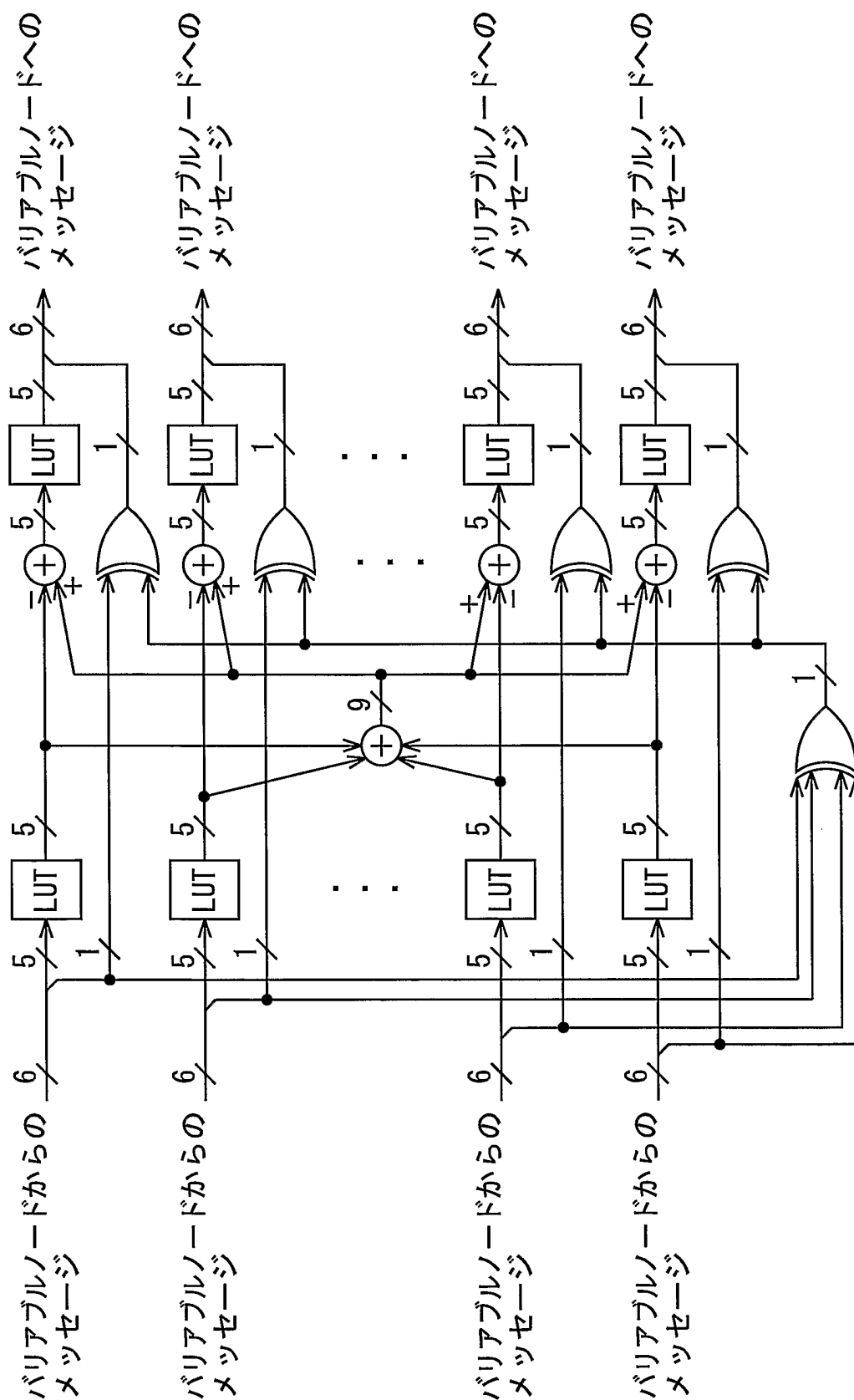


図14

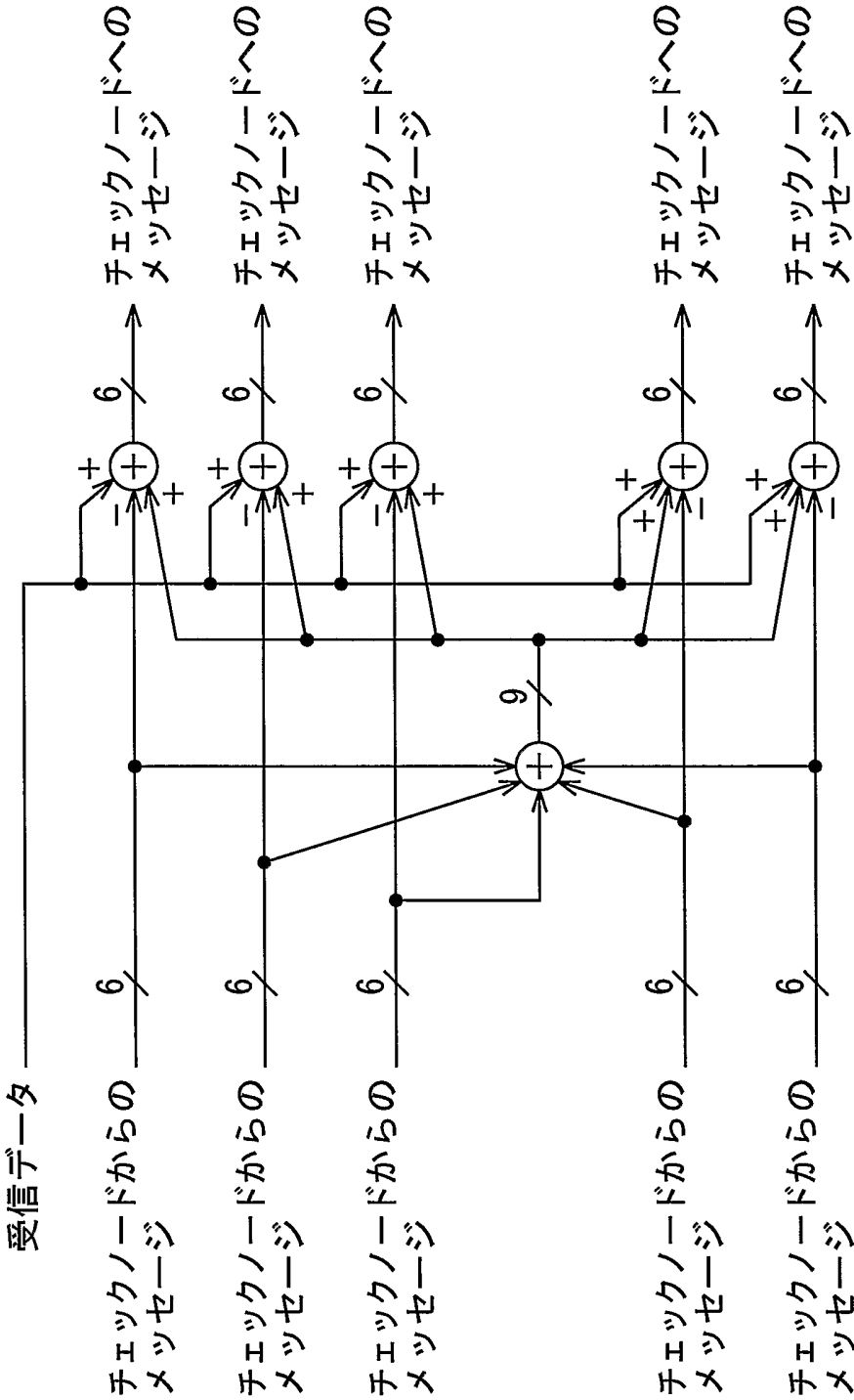


図15

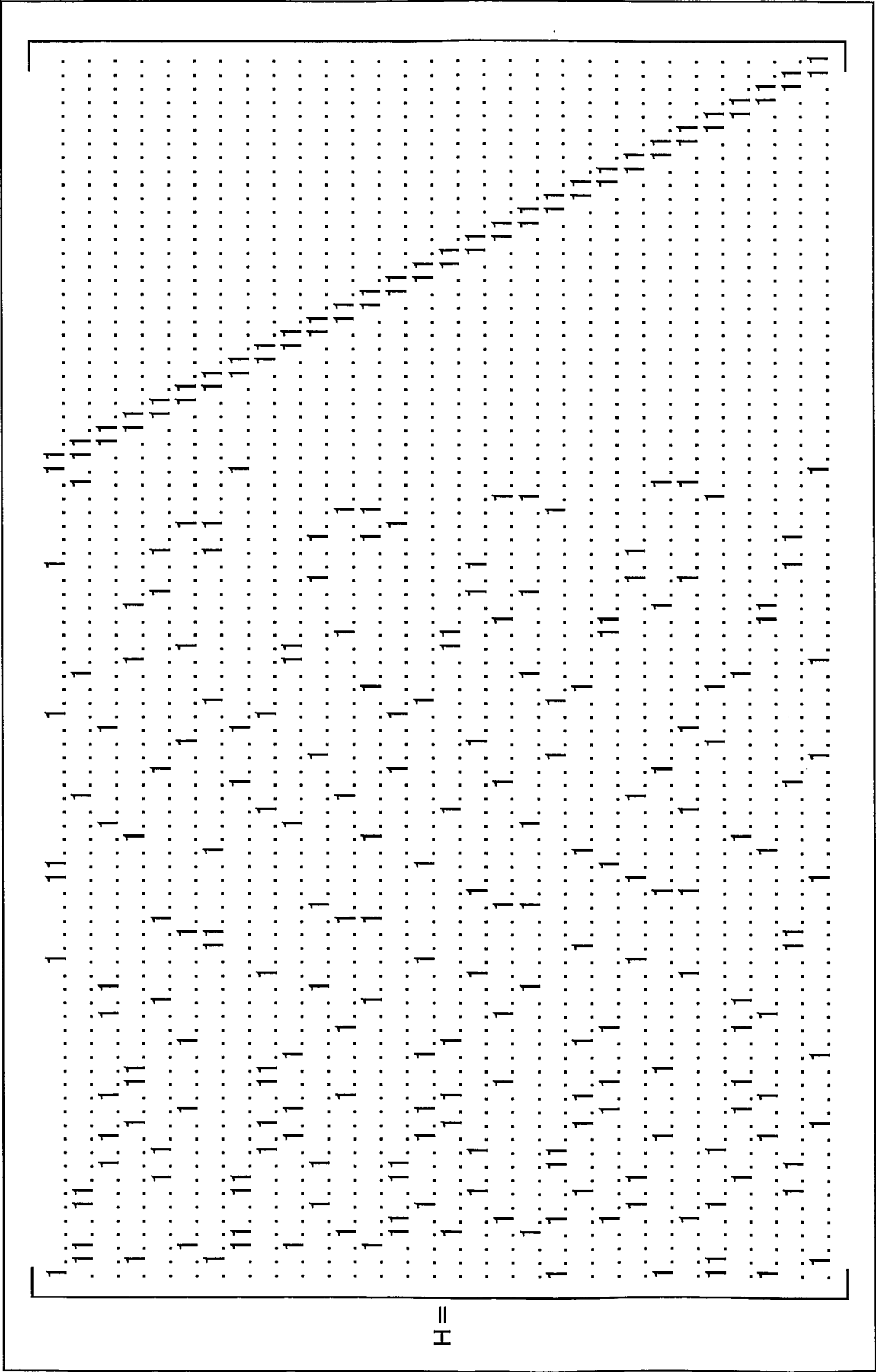


図16

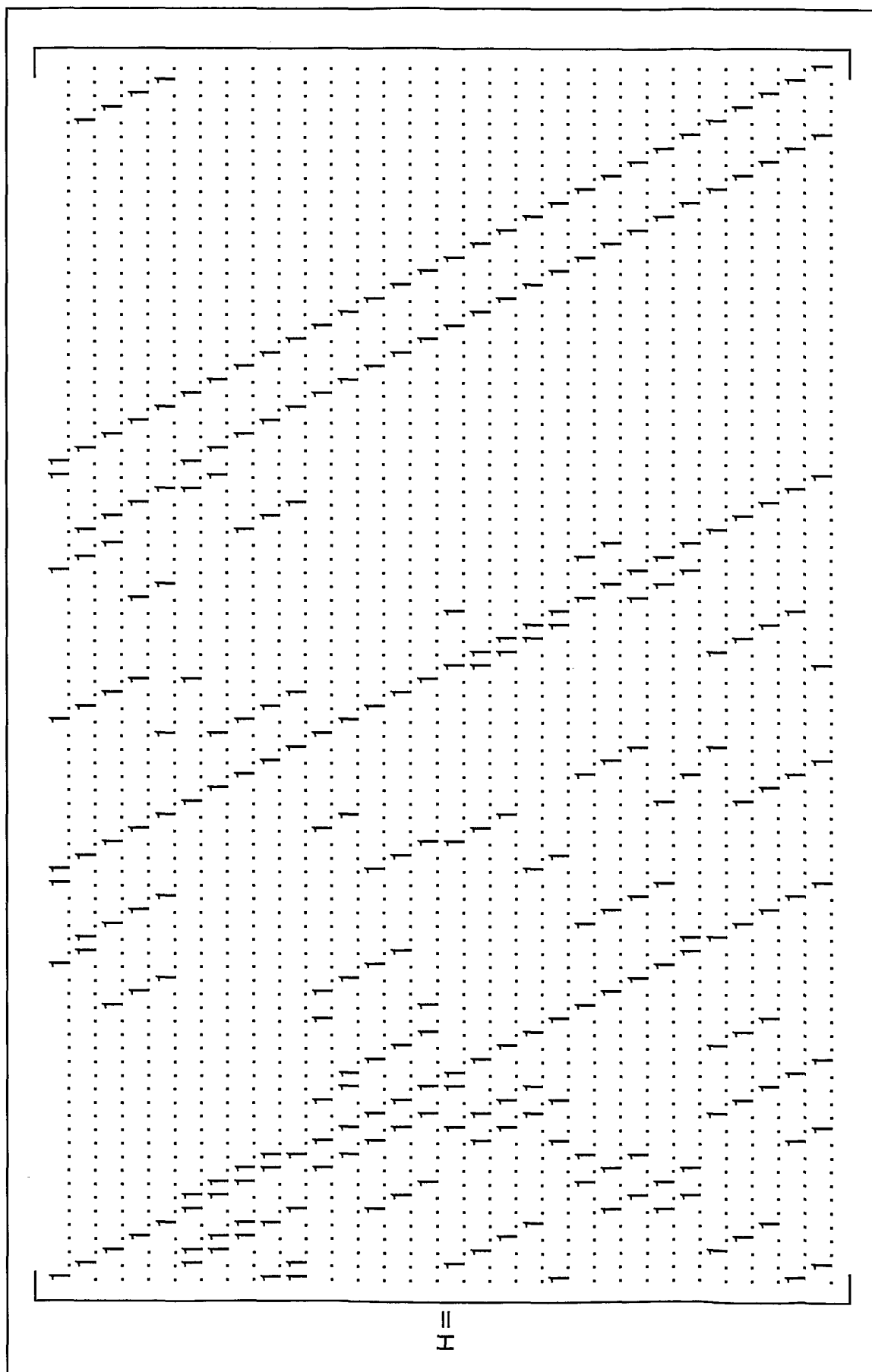


図17

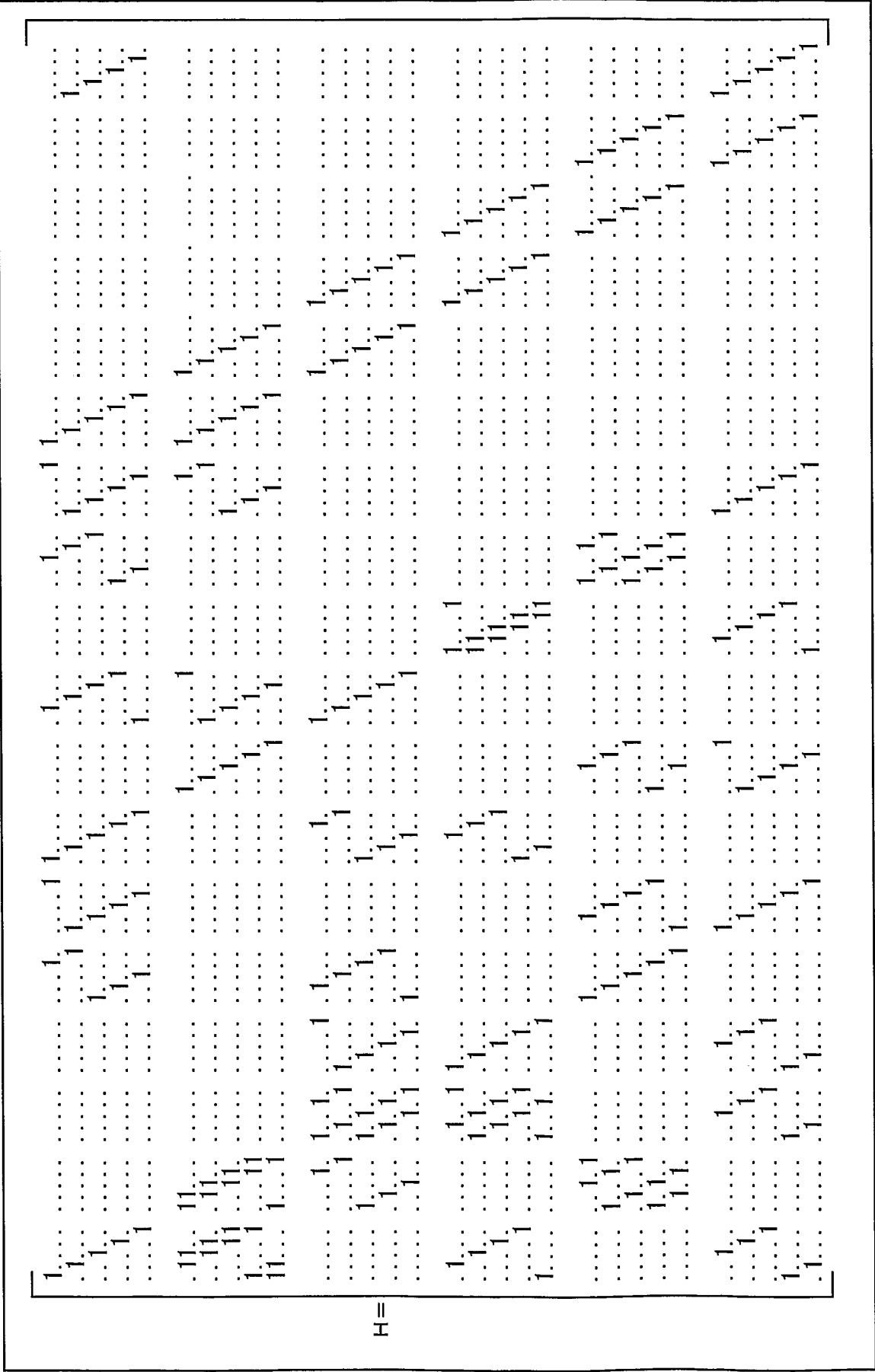


図18A

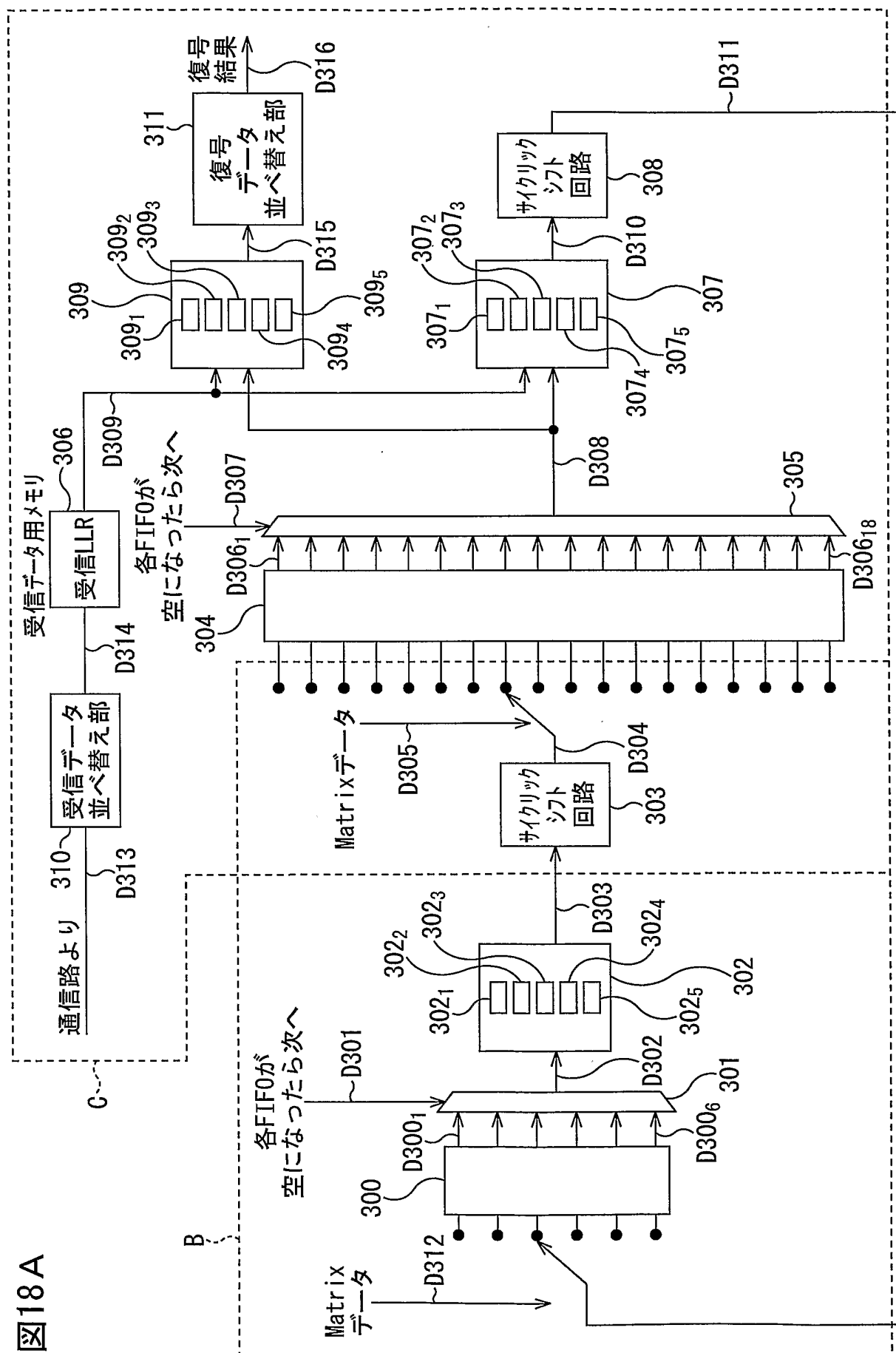
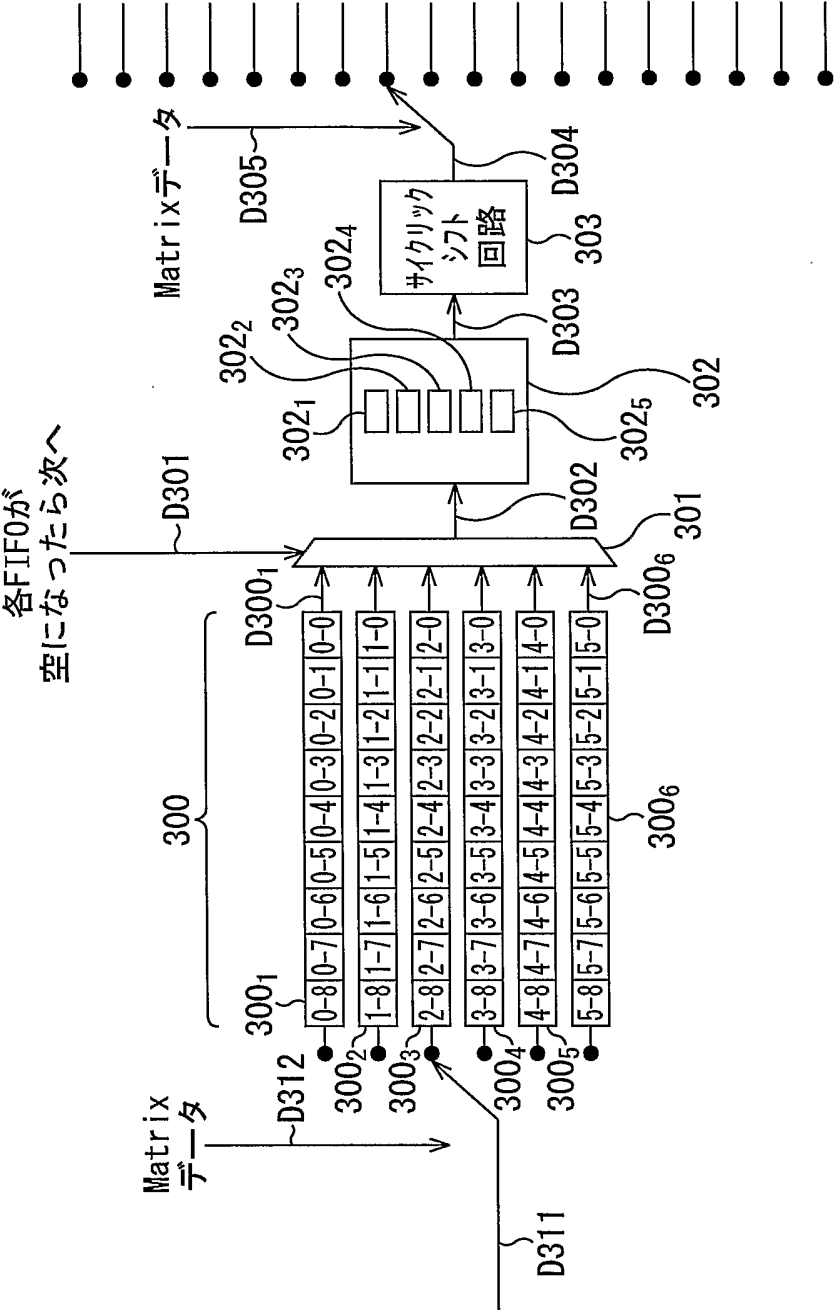


図18B



21/22

図18C

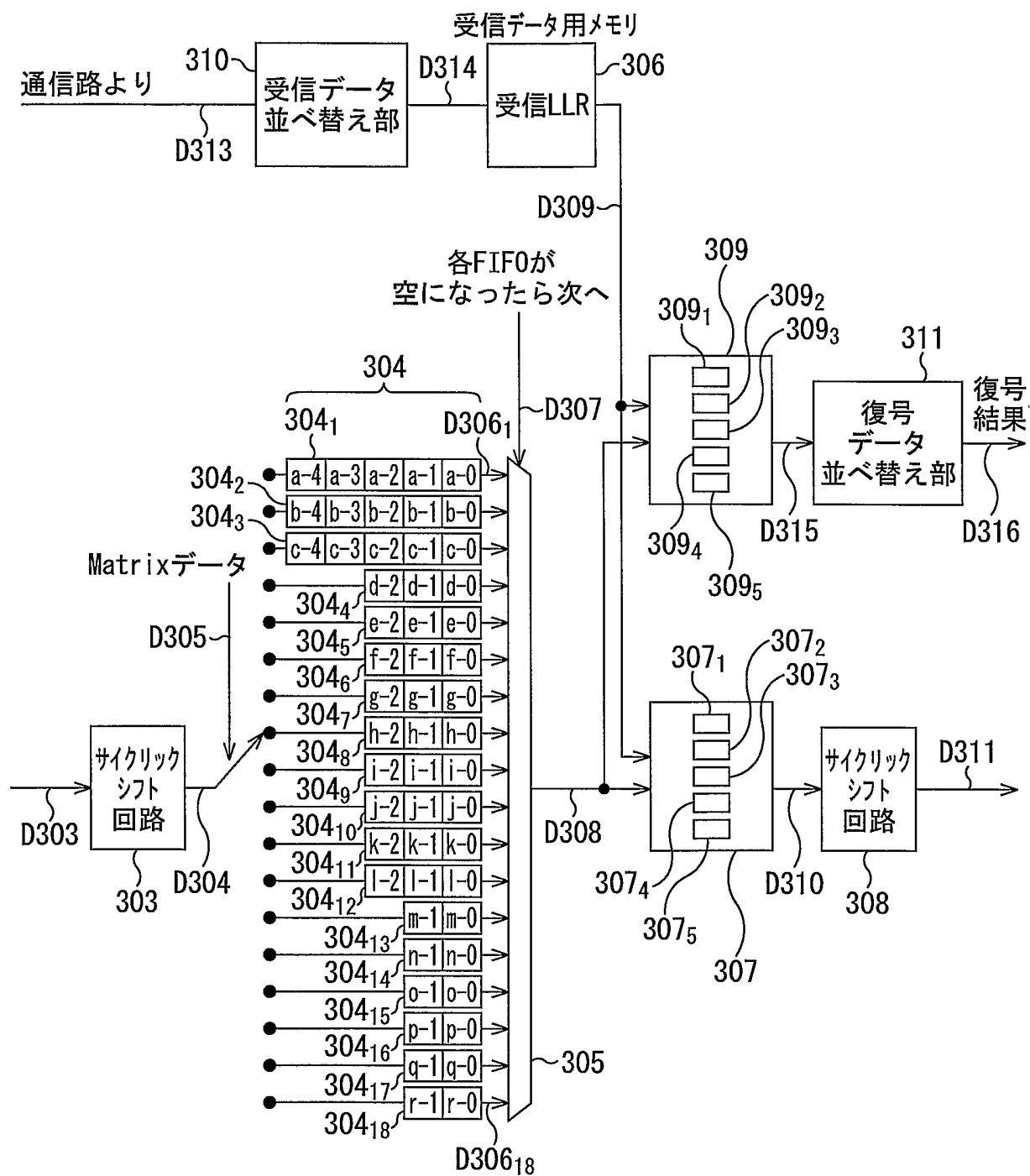
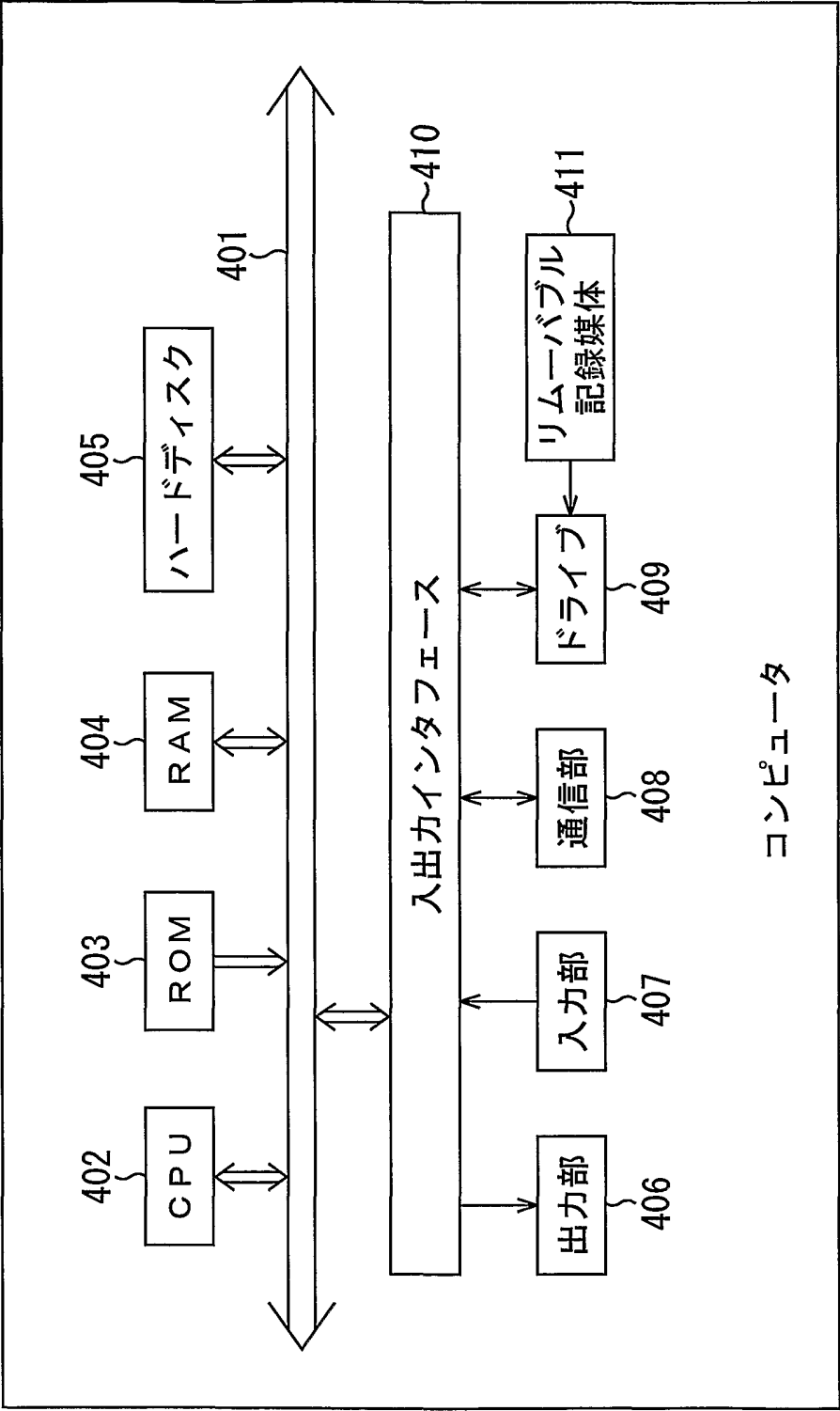


図19



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2004/005551

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ H03M13/09, 13/19

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ H03M13/00-13/53

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	1996-2004
Kokai Jitsuyo Shinan Koho	1971-2004	Toroku Jitsuyo Shinan Koho	1994-2004

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

IEEE Xplore LDPC, Transposition, IEICE Web LDPC, TEIMITSUDO PARITY, CHIKAN (in English and in Japanese)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
E, A	JP 2004-147318 A (Samsung Electronics Co., Ltd.), 20 May, 2004 (20.05.04), Full text; all drawings (Family: none)	1-20
A	JP 2002-118474 A (Sony Corp.), 19 April, 2002 (19.04.02), Full text; all drawings & US 2002-116677 A1	1-20
A	JP 2002-33670 A (Agere Systems Guardian Corp.), 31 January, 2002 (31.01.02), Full text; all drawings & EP 1158682 A2 & US 6539367 B	1-20

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

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

05 July, 2004 (05.07.04)

Date of mailing of the international search report

20 July, 2004 (20.07.04)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl⁷ H03M13/09, 13/19

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. Cl⁷ H03M13/00-13/53

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	1971-2004年
日本国実用新案登録公報	1996-2004年
日本国登録実用新案公報	1994-2004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 LDPC, Transposition
 IEICE Web LDPC, 低密度パリティ, 置換

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 請求の範囲の番号
EA	JP 2004-147318 A (三星電子株式会社) 2004.05.20, 全文, 全図 (ファミリーなし)	1-20
A	JP 2002-118474 A (ソニー株式会社) 2002.04.19, 全文, 全図 & US 2002-116677 A1	1-20
A	JP 2002-33670 A (アギア システムズ ガーディアン コーポレーション) 2002.01.31, 全文, 全図 & EP 1158682 A2 & US 6539367 B	1-20

☐ 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

05.07.2004

国際調査報告の発送日

20.7.2004

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)
 郵便番号100-8915
 東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

田中 庸介

5K

8529

電話番号 03-3581-1101 内線 3555