

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2006年4月20日 (20.04.2006)

PCT

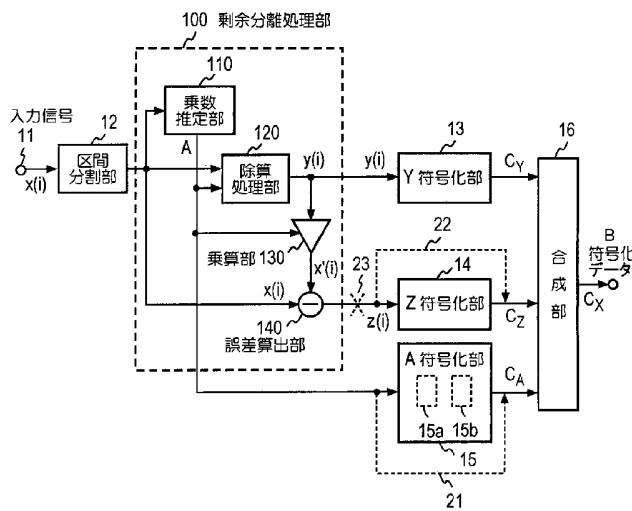
(10) 国際公開番号
WO 2006/041165 A1

- (51) 国際特許分類:
G06F 5/00 (2006.01) H03M 7/30 (2006.01)
G10L 19/00 (2006.01)
- (21) 国際出願番号: PCT/JP2005/018978
- (22) 国際出願日: 2005年10月14日 (14.10.2005)
- (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 2004-329735
2004年10月15日 (15.10.2004) JP
特願 2004-329736
2004年10月15日 (15.10.2004) JP
- (71) 出願人 (米国を除く全ての指定国について): 日本電信電話株式会社 (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) [JP/JP]; 〒1008116 東京都千代田区大手町二丁目3番1号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 原田 登 (HARADA, Noboru) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目9番11号 NTT 知的財産センター内 Tokyo (JP). 関川 浩 (SEKIGAWA, Hiroshi) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目9番11号 NTT 知的財産センター内 Tokyo (JP). 守谷 健弘 (MORIYA, Takehiro) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目9番11号 NTT 知的財産センター内 Tokyo (JP). 白柳 潔 (SHIRAYANAGI, Kiyoshi) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目9番11号 NTT 知的財産センター内 Tokyo (JP).
- (74) 代理人: 中尾 直樹, 外 (NAKAO, Naoki et al.); 〒1600022 東京都新宿区新宿三丁目1番22号 新宿 NSOビル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,

[続葉有]

(54) Title: INFORMATION ENCODING METHOD, DECODING METHOD, COMMON MULTIPLIER ESTIMATION METHOD, DEVICE USING THESE METHODS, PROGRAM, AND RECORDING MEDIUM

(54) 発明の名称: 情報符号化方法、復号化方法、共通乗数推定方法、これらの方法を利用した装置、プログラム及び記録媒体



- 11... INPUT SIGNAL
- 12... SECTION DIVIDING UNIT
- 100... REMAINDER ISOLATION UNIT
- 110... MULTIPLIER ESTIMATING UNIT
- 120... DIVISION UNIT
- 130... MULTIPLICATION UNIT
- 140... DIFFERENCE CALCULATION UNIT
- 13... Y ENCODING UNIT
- 14... Z ENCODING UNIT
- 15... A ENCODING UNIT
- 16... SYNTHESIS UNIT
- B... ENCODED DATA

(57) Abstract: There are provided a method for encoding signals by separating them by using a common multiplier so as to improve the compression efficiency of an audio signal and a method for obtaining a multiplier common to a numeric set including an error. A multiplier A common to each input signal sample $x(i)$ is estimated for each frame. Each sample $x(i)$ is divided by A and rounding is performed by the significant number of digits. A quotient obtained $y(i)$ is encoded. The $y(i)$ is multiplied by A and the multiplication result is subtracted from $x(i)$ to determine a difference signal $z(i)$, which is encoded. The A is encoded and the code of $y(i)$, the code of $z(i)$, and the code of A are combined and outputted. When estimating a common multiplier, each numeric value is divided by a decided representative value. Between the maximum value of difference of positive side and the maximum value of the difference of negative side, irreducible fractions are obtained by dividing each numeric value by a representative value of correct values and by rational approximation. Among the obtained irreducible fractions, the one having minimum denominator common to each numeric values is obtained and the common denominator is corrected by the representative value so as to be

a common multiplier.

[続葉有]



WO 2006/041165 A1



DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, 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, NA, 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,

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。

(57) 要約: 本発明は、オーディオ信号の圧縮効率を向上させるため、信号を共通の乗数を用いて分離して符号化する方法、および誤差を含む数値集団に共通の乗数を求める方法を提供することを目的とする。入力信号サンプル $x(i)$ をフレーム毎に各サンプルに共通の乗数 A を推定する。 A で各サンプル $x(i)$ を除算し、有効桁数で丸め処理し、商信号 $y(i)$ を符号化する。 $y(i)$ に A を乗算し、その乗算結果を $x(i)$ から減算処理して誤差信号 $z(i)$ を求め、 $z(i)$ を符号化する。 A を符号化し、 $y(i)$ の符号、 $z(i)$ の符号、 A の符号を合成して出力する。共通乗数の推定では、決定した代表数値で各数値を割算する。正側誤差の最大と負側誤差の最大との間に、正しい値の代表値で各数値を割った既約分数を有理近似で求める。この求めた既約分数値中の分母が各数値に対して共通となる最小のものを求め、その共通の分母を前記代表値で修正して共通乗数とする。

明 細 書

情報符号化方法、復号化方法、共通乗数推定方法、これらの方法を利用した装置、プログラム及び記録媒体

技術分野

[0001] この発明は、音声、音楽などの音響信号や画像信号(以下これらを含めて情報信号という)の複数のサンプル値を通信路により伝送したり、情報記録媒体に記録する場合に伝送効率を高めたり、記録容量を少なくするためなどに用いる情報圧縮符号化方法、その伸張復号化方法、前記情報圧縮符号化に用いる複数の情報集合に対する共通の乗数を推定する方法、これらの方法を利用した装置、プログラム及び記録媒体に関する。

背景技術

[0002] 音響信号データ(サンプル値)に対し歪を許す圧縮符号化方式としてMP3、AAC、TwinVQ等がある。また、画像情報データ(サンプル値)の圧縮符号化方式としてJPEG等がある。また歪を許さない可逆な符号化(ロスレス符号化)技術がある(例えば非特許文献1参照)。また編集加工が容易な浮動小数点形式のデータの可逆圧縮も重要である(例えば非特許文献2参照)。また、従来、複数の数値に対し、共通の最大公約数、つまり共通の乗数を求める方法としてユークリッド互除法が知られている。

音響信号の場合には、マイクロホン等で收音した音響信号をサンプリング周波数 f 、量子化ビット数 q でサンプリングし、各サンプル値をデジタル値に変換されたデジタルサンプルとして入力される。この場合、各アナログのサンプルに共通の定数を掛け合わせてゲイン調整を行う場合が多い。アナログ→デジタル変換が行われる前にアナログ領域で定数が掛け合わされ、その後アナログ→デジタル変換が行われているような場合には、アナログ信号領域で誤差を含む場合もある。

[0003] 同様に、画像情報信号を圧縮符号化する際に、2次元の画像信号をラスタ走査して1次元のサンプル列として得る場合にも、元の画像信号を取得する際に、各サンプルに共通の定数を掛け合わせてゲイン調整がなされている場合がある。

元のサンプル列の各サンプル $s_0(i)$ に、ゲインとして実数 G を共通に掛け合わせて

得られたサンプル $s_1(i)$ は、 $s_1(i) = s_0(i) \times G$ となる。

この実数 G を掛け合わせて得られたサンプル $s_1(i)$ を、2進表現やIEEE754形式の2進浮動小数点表現で表現し、これらデジタルサンプル列を符号化する場合が多い。IEEE-754として標準化されている浮動小数点形式は図1に示すように32ビットである。上位ビットから極性1ビット、指数部8ビット、仮数部23ビットで構成される。極性を S 、指数部8ビットで表す値を10進数で E 、仮数部の2進数を M とすると、この浮動小数点形式の数値は絶対値表現2進数で表わすと式(1)となる。

[0004] [数1]

$$(-1)^S \times 1.M \times 2^{E-E_0} \quad (1)$$

IEEE-754によれば、 $E_0 = 2^7 - 1 = 127$ と決められており、これにより、式(1)中の $E - E_0$ は

$$-127 \leq E - E_0 \leq 128$$

の範囲の任意の値を取ることができる。ただし、 $E - E_0 = 127$ の場合はall“0”、 $E - E_0 = 128$ の場合はall“1”と定義されている。 $E - E_0 = n$ は式(1)で表される値の整数部分の桁数(ビット数)から1を減算した値、即ち、最上位の“1”より下位ビット数を表わしている。

[0005] 従来の圧縮符号化方法、例えば文献1に示す方法は、入力サンプル列の原入力波形(例えば音楽信号波形)の冗長性をなるべく除去して、情報量を減少させるものが主であった。また符号化装置で符号化可能な入力波形の振幅(ビット長)は、例えば16ビットと決められている。したがって、例えばビット長が24ビットの場合は上位16ビットを符号化し、下位の8ビットはそのまま出力するか、別個に圧縮符号化して出力している。

非特許文献1:Hans M. and Schafer R.W.:Lossless Compression of Digital Audio, IE EE Signal Processing Magazine, Vol.18, No.4, pp.21-32(2001).

非特許文献2:Dai Yang and Takehiro Moriya:Lossless Compression for Audio Data in the IEEE Floating-Point Format, AES Convention Paper 5987,AES 115th Convention, New York, NY, USA, 2003 OCTOBER 10-13.

発明の開示

発明が解決しようとする課題

[0006] 従来の符号化方法(装置)では、予め決められた振幅(ビット長)より大きな入力サンプル列は、符号化できない。符号化可能な振幅(ビット長)に変換して符号化する場合は、符号化品質が劣化する。または符号化できない部分を更に別の情報(符号)とする場合は、情報量が増加するという課題があった。

例えば、このサンプル列を歪を許さないユニバーサル符号化方法で符号化するとし、サンプル列の各サンプルをビット列として見る。ここでサンプル列 $x(1) \sim x(3)$ が、 $x(1) = 10100000$ (10進数160)、 $x(2) = 100000$ (10進数32)、 $x(3) = 11100000$ (10進数224)のように0が連続して存在するような場合は、高い圧縮率で符号化することができる。これに対し、 $x(1) = 11111010$ (10進数250)、 $x(2) = 110010$ (10進数50)、 $x(3) = 101011110$ (10進数350)のように0と1が混在した複雑な場合は、高い圧縮率で符号化することは困難である。

[0007] さらに、従来のユークリッドの互除法は比較的処理が簡単であるが、数値が四捨五入される場合、あるいは誤差を含むような場合は、その誤差を考慮して最大公約数を求めることはできなかった。

この発明の目的は、入力波形の振幅(ビット長)が、圧縮符号化法が可能とする振幅(ビット長)より大きい場合でも、情報量を増加させることなく符号化することができる符号化技術を提供することである。また、入力サンプルを構成するビットの0と1の混在度合いに関わらず効率良くサンプル列を圧縮することができる符号化技術、復号化技術、前記符号化技術に利用する共通乗数を求める技術を提供することである。

課題を解決するための手段

[0008] この発明では、サンプル列から各サンプルに共通する数(共通の乗数A)を推定する。各サンプルをそれぞれ共通乗数Aで割算し、その割算結果を圧縮符号化し、その圧縮符号化符号 C_Y と、上記推定乗数Aとを少なくとも出力する。乗数Aは圧縮符号化してもしなくてもよい。

復号化は、符号 C_Y と乗数Aとを分離し、符号 C_Y をその符号化と対応する復号化方法により復号化し、その復号化結果に対し共通乗数Aを掛け算して、原サンプルを求

める。

また、前記圧縮符号化で共通乗数を求める処理では、まず数値集合中から代表数値を決定する。その代表数値で各数値割算した値で正側誤差の最大と負側誤差の最大との間に、正しい値の代表値(誤差を含まない)で各数値の正しい値を割った既約分数を有理近似で求める。そして、この求めた既約分数値中の分母が各数値に対して共通となる最小のものを求め、その共通の分母を前記代表値で修正して共通乗数とする。

発明の効果

[0009] 例えば、アナログレベルで共通のゲインが乗算された圧縮符号化処理対象のサンプル列の場合は、その共通のゲインが共通の乗数と推定され、この乗算により各サンプルが割算される。したがって、振幅が小さくなり、従来の符号化法(例えば非特許文献1)に示す方法をそのまま適用できる場合が多く、かつ品質劣化が少なく、情報量の増加もほとんどない。

また前記割算の結果、元のサンプル列の共通の冗長性を取り除くことができるため、商の集合のエントロピーを大きく減少させることができる。したがって、結果として全体の情報圧縮率を向上することができる。

[0010] 例えば、浮動小数点表現された音響信号にこの発明を適用した場合、従来方式と比較して、情報量を平均で $1/2 \sim 1/3$ に圧縮することが出来る。

さらに、共通乗数を求める処理では、誤差を含む数値の集合に対し、誤差を考慮した共通乗数を求めることができる。

図面の簡単な説明

[0011] [図1]IEEE-754規格の浮動小数点表現説明図。

[図2]符号化装置の実施例1の機能構成例を示す図。

[図3]その処理手順の例を示す流れ図。

[図4]復号化装置の実施例1の機能構成例を示す図。

[図5]その処理手順の例を示す流れ図。

[図6]図6Aは、この発明の効果を説明するための具体的数値例を示す図であって、サンプル列の例を示す図。図6Bは、この発明の効果を説明するための具体的数値

例を示す図であって、サンプル列を共通乗数で除算した例を示す図。図6Cは、この発明の効果を説明するための具体的数値例を示す図であって、除算されたサンプル列の例を示す図。

[図7]図7Aは、この発明の効果を説明するための他の具体的数値例を示す図であって、サンプル列の例を示す図。図7Bは、この発明の効果を説明するための他の具体的数値例を示す図であって、除算されたサンプル列の例を示す図。

[図8]符号化装置の実施例2の要部機能構成例を示す図。

[図9]その処理手順の例を示す流れ図。

[図10]復号化装置の実施例2の要部機能構成例を示す図。

[図11]符号化装置の実施例3の要部機能構成例を示す図。

[図12]その処理手順の例を示す流れ図。

[図13]符号化装置の実施例4の要部機能構成例を示す図。

[図14]その処理手順の例を示す流れ図。

[図15]符号化装置の実施例5の要部機能構成例を示す図。

[図16]符号化装置の実施例6の要部機能構成例を示す図。

[図17]符号化装置の実施例7の要部機能構成例を示す図。

[図18]復号化装置の実施例7の要部機能構成例を示す図。

[図19]符号化装置の実施例7の変形例要部機能構成例を示す図。

[図20]図19と対応した復号化装置の実施例の機能構成例を示す図。

[図21]符号化装置の実施例8の機能構成例を示す図。

[図22]符号化装置の実施例9の機能構成例を示す図。

[図23]符号化装置の実施例10の処理手順の例を示す流れ図。

[図24]復号化装置の実施例11の機能構成例を示す図。

[図25]復号化装置の実施例12の処理手順の例を示す流れ図。

[図26]図26Aは、符号化装置の実施例13の機能構成例を示す図。図26Bは、その復号化装置の実施例の機能構成例を示す図である。

[図27]符号化装置の実施例14の要部機能構成例を示す図。

[図28]符号化装置の実施例15の機能構成例を示す図。

- [図29]復号化装置の実施例16の機能構成例を示す図。
[図30]符号化装置の実施例17の機能構成例を示す図。
[図31]復号化装置の実施例18の機能構成例を示す図。
[図32]実施例19における共通乗数の探索手順の例を示す流れ図。
[図33]その共通乗数のチェックと商信号の変形例を示す流れ図。
[図34]商信号変形を浮動小数点形式に対し行う例を示す流れ図。
[図35]浮動小数点形式を整数形式に変換する処理例を示す流れ図。
[図36]商信号が正の場合における変形手順の処理例を示す流れ図。
[図37]商信号が負の場合における変形手順の処理例を示す流れ図。
[図38]有理近似により共通乗数を求める機能構成例を示す図。
[図39]その他の例を示す図。
[図40]更にその他の例を示す図。
[図41]有理近似により共通乗数を求める機能構成の他の例を示す図。
[図42]図42Aは、自己相関により共通乗数を求める数値例を示す図であって、1ビットシフトした自己相関結果を示す図。図42Bは、自己相関により共通乗数を求める数値例を示す図であって、自己相関のビットが1の数の例を示す図。

発明を実施するための最良の形態

- [0012] 以下この発明の実施形態を図面を参照して説明する。各図において、他の図と対応する部分には同一参照番号を付けて重複説明を省略する。またこの発明を、32ビットの浮動小数点数で表現された音響信号列に対して適用する場合を例とする。

[実施形態1]

実施例1

符号化処理

図2にこの発明の符号化装置の実施形態1の機能的構成例を、図3にその処理手順例をそれぞれ示す。

- [0013] 入力端子11よりの入力信号列 $X = (x(1), x(2), \dots)$ はデジタル化されたサンプル列であり、これが入力されると(ステップS1)、区間(フレーム)分割部12で所定数 N 個(例えば1024個)のサンプル列に分割され、図に示していないが一旦記憶部に記

憶される(ステップS2)。サンプル列は、音響信号の場合は、24ビットの量子化ビット数で量子化された整数サンプル列や32bit単精度浮動小数点形式サンプル列などが考えられる。カラー画像信号の場合は各色情報要素に分解してラスタ走査したピクセル情報のデジタル化されたサンプル列がある。

- [0014] 分割区間ごとの入力信号としてのサンプル $x(i)$ ($i=0, 1, \dots, N-1$)の集合は、剰余分離処理部100で乗数 A と、 $x(i)$ を乗数 A で割った商 $y(i)$ とに少なくとも分離される。この例では $x(i)$ を A で割った余り(誤差) $z(i)$ も分離される。

例えば入力信号列 X は、分割された1フレームごとに乗数推定部110に渡される。乗数推定部110では、全てのサンプル $x(i)$ ($i=0, 1, \dots, N-1$)に対して、共通の乗数 A を推定する(ステップS3)。例えば、全てのサンプル $x(i)$ が共通の値99.0で割り切れる場合には、共通の乗数 $A=99.0$ とする。この共通の乗数 A を推定する方法については、複数種類の方法が考えられるが、例えば近似共通ファクタ(ACF: Approximate Common Factor)の有理近似を用いる。ここでは、適切な乗数 A が与えられるものとする。

- [0015] 乗数推定部110で決定された乗数 A は、除算処理部120、乗算部130、 A 符号化部15に渡される。

除算処理部120では、乗数推定部110より渡された乗数 A と、 N 個のサンプル $x(i)$ を入力とし、 n 個の商信号 $y(i) = x(i) / A$ を算出する(ステップS4)。このとき、 $y(i)$ は整数形式、浮動小数点形式、固定小数点形式のいずれでもよい。その決められた表現形式に変換するときに、切り捨て、切り上げ、四捨五入、nearest tie to even(ニアレストタイツウイーブン)などの丸め処理を行ってもよい。

- [0016] 例えば、 $x(i)$ を倍精度の浮動小数点数(ここでは64ビットの浮動小数点数)に変換して乗数 A で除算する。得られた倍精度の商を、最も近い単精度(32ビット)の浮動小数点数に丸めて商信号 $y(i)$ とする。

除算処理部120で得られた N 個の商信号 $y(i)$ から成る商信号列 $Y = (y(0), y(1), \dots, y(N-1))$ は、 Y 符号化部13と、この例では乗算部130に渡される。

乗算部130では、乗数推定部110から渡された乗数 A を除算処理部120から渡された商信号列 Y の N 個の各信号 $y(i)$ にそれぞれ乗算して、復元された N 個のサン

ル $x'(i)$ を得る(ステップS5)。

- [0017] このとき、復元サンプル $x'(i)$ は32ビットの浮動小数点表現で表現可能な値の範囲に丸められる。例えば、 $y(i)$ とAを掛け合わせた結果を倍精度(64ビット)の浮動小数点数として保持し、得られた倍精度の乗算結果を、最も近い単精度(32ビット)の浮動小数点数に丸めて $x'(i)$ とする。得られたサンプル列 $X' = (x'(0), x'(1), \dots, x'(N-1))$ は、誤差算出部140に渡される。

誤差算出部140は、入力信号から取り出したN個のサンプル $x(i)$ それぞれから乗算部130から渡されたN個の復元サンプル $x'(i)$ を差し引いて、N個の誤差信号 $z(i)$ から成る誤差信号列 $Z = (z(0), z(1), \dots, z(N-1))$ を得る(ステップS6)。この誤差信号の算出は減算を用いる代わりに $x(i)$ と $x'(i)$ の32ビットをそのままビット単位の排他的論理和演算(xor)を行ってもよい。要するに $x(i)$ と $x'(i)$ との差分演算を行えばよい。

- [0018] 得られた誤差信号列Zはこの例ではZ符号化部14に渡される。

乗数推定部110よりの乗数AはA符号化部15にも入力される。

Y符号化部13、Z符号化部14、A符号化部15は、既存の情報圧縮手法を用いてそれぞれの入力信号列Y、Z、Aを圧縮符号化して得られた符号を符号 C_Y 、符号 C_Z 、符号 C_A として出力する(ステップS7)。これら符号 C_Y 、 C_Z 、 C_A は合成部16で1フレームごとのビット列として合成され、符号化データ C_X として出力される(ステップS8)。Y符号化部13には、波形値の相関を利用した高圧縮率の符号化方法、例えば非特許文献1に示す各種の可逆圧縮符号化、もしくはMPEG4、AAC、TwinVQなど非可逆の圧縮符号化法を用いてもよい。A符号化部15は可逆の圧縮符号化方式を用いる。

- [0019] 乗数Aは、商信号列Yや誤差信号列Zと比較して情報量が無視できる程度に小さい場合は破線21で示すようにA符号化部15を設けることなく乗数Aをそのまま合成部16へ入力してもよい。

誤差信号列Zは、入力信号 $x(i)$ の性質によっては十分な圧縮利得が得られず、演算量増加の影響が大きくなる場合がある。そのような場合は、Z符号化部14を省略して破線22で示すように誤差信号列Zをそのまま合成部16へ入力してもよい。

Z符号化部14は例えばエントロピ符号化法を用いるが、誤差信号 $z(i)$ 中の上位ビットのみを用いてエントロピ符号化し、非可逆圧縮符号化としてもよい。更に入力信号 $x(i)$ の性質によっては、誤差信号 $z(i)$ を必ず0にできると判っている場合や、 $z(i)$ が再生品質に実質的に影響を与えない程度に非常に小さくなることが判っている場合は、破線の掛け印×23で示すように誤差信号列Zを破棄してもよい。この場合はZ符号化部14は省略され、符号化データ C_x には符号 C_z は含まれない。

復号化処理

実施形態1の復号化装置機能構成例を図4に、復号化方法(処理手順)の例を図5にそれぞれ示す。

- [0020] 入力端子51からの符号化データ C_x は符号化装置と同じサンプル数を1フレームとし、フレームごとに処理される。符号化データ C_x が入力されると(ステップS11)、分離部52は符号化データ C_x を、フレームごとに符号 C_y と符号 C_z と符号 C_A とに分離し、対応するY復号化部53、Z復号化部54、A復号化部55に渡す(ステップS12)。Y復号化部53、Z復号化部54、A復号化部55には、それぞれY符号化部13、Z符号化部14、A符号化部15の各圧縮符号化方法と対応する伸張復号化方法が用いられる。
- [0021] 各復号化部53～55は、それぞれ入力された符号を復号化処理して、符号化部13～15の各入力に対応した商信号列Y、誤差信号列Z、共通の乗数Aを求める(ステップS13)。なおこの例では各符号化部13～15が可逆符号化方法を用いているものとしている。

剰余結合処理部200は、それらの復号化信号を結合処理する。

乗算部210は、A復号化部55から渡された乗数Aを、Y復号化部53から渡されたN個の商信号 $y(i)$ に対し、符号化装置の乗算部130で行われたのと同じやり方で乗算してN個の信号 $x'(i)$ を得る(ステップS14)。このとき、 $x'(i)$ は32ビットの浮動小数点表現で表現可能な値の範囲に丸められる。例えば、 $y(i)$ とAを掛け合わせた結果を倍精度(64ビット)の浮動小数点数として保持し、得られた倍精度の乗算結果を、最も近い単精度(32ビット)の浮動小数点数に丸めて $x'(i)$ とする。この丸めの処理は、符号化装置の乗算部130で行われた処理と同じやり方で行う。このようにして所定の有効桁までの $x'(i)$ を求める。

[0022] 誤差補正部220は、乗算部210からのN個のサンプル $x'(i)$ を、それぞれZ復号化部54からの誤差信号 $z(i)$ により補正処理をして、元の信号 $x(i)$ を復元する。つまり符号化装置で $x(i)$ から $x'(i)$ を減算した場合は、各 $x'(i)$ に対応する誤差信号 $z(i)$ を加算する。符号化装置で $x(i)$ と $x'(i)$ とのビット単位の排他的論理和演算(xor)をした場合には、対応する誤差信号 $z(i)$ の各ビットを参照して $z(i)$ が1となった位置にあたる $x'(i)$ のビットを反転して $x(i)$ を得る。

フレーム連結部56は、誤差補正部220からの復元されたN個のサンプル $x(i)$ ($i=0, 1, \dots, N-1$)を連結させて出力する(ステップS16)。

[0023] 符号化装置が乗数Aを符号化していない場合は、分離部52で分離された乗数Aは図4中に破線61で示すように乗算部210に直接入力される。同様に、符号化装置が、誤差信号 $z(i)$ を符号化していない場合は、分離部52で分離された $z(i)$ は破線62で示すように誤差補正部220に直接入力される。符号化装置で $z(i)$ が破棄された場合は、分離部52で符号 C_z と対応するものが得られず、誤差補正部220にはなにも入力されない。言い換えると0が入力されることになる。

なお、符号化装置は、乗数Aが前フレームで得られた乗数Aと同一であれば、さらに情報量を少なく出来る。具体的には、図2のA符号化部15に示すように、符号化装置にレジスタ15aと判定部15bを設ける。レジスタ15aには、前フレームの乗数Aを格納しておき、判定部15bで現フレームの乗数Aと前フレームの乗数Aとを比較する。そして、同一であればそのことを示す1ビットのみを符号 C_A として出力する。これにより符号化データ C_x の情報量を減少することができる。復号化装置では、図4中のA復号化部55内の前フレームの乗数Aを格納するレジスタ55aと、符号 C_A が前フレームと同一乗数であることを示すビットであるか否かを判定し、同一であれば、符号 C_A の復号化処理を行うことなく、レジスタ55a内の乗数Aを乗算部210に出力する判定部55bを設ければよい。この場合は、符号 C_A は乗数Aが前フレームと同一か否かを示す1ビットでよく、乗数Aを符号化しない場合よりも、情報量を減少させることができる。

[0024] この発明を適用すると圧縮効率を高くすることができることを、図6を参照して説明する。図6は、入力信号 $x(i)$ が2進数表現の場合の1例である。入力信号サンプル列

$x(1)$, $x(2)$, $x(3)$ が、図6Aに示すように10進数表現で $x(1) = 250$ 、 $x(2) = 50$ 、 $x(3) = 350$ の場合、その2進数表現はそれぞれ、「0」と「1」が比較的ランダムに配列されている。

これらサンプル $x(1)$, $x(2)$, $x(3)$ を共通の数 $A = 1.5625$ で割算すると、その商信号 $y(1)$, $y(2)$, $y(3)$ は図6Bに示すようにそれぞれ10進数表現で $y(1) = 160$ 、 $y(2) = 32$ 、 $y(3) = 224$ となる。

[0025] これら商信号 $y(1)$, $y(2)$, $y(3)$ の2進数表現はそれぞれ図6Cに示すように、「0」が連続して存在する部分が多い「1」と「0」の配列である。この $y(1)$, $y(2)$, $y(3)$ は高圧縮符号化が可能である。乗算 A も送出する必要があるが、サンプル数が多く、かつ、その大部分が高圧縮可能な商信号の場合は、全体としての圧縮符号化量を大きく減少させることができる。

図7にIEEE-754浮動小数点数のサンプル列に適用した例を示す。10進数表現の数値478.4, 95.68, 669.76の浮動小数点表現は、それぞれ図7Aに示すようにその仮数部の23ビットの「1」、「0」の配列はかなりランダムである。しかし、これらを共通乗数2.99で割算すると、その商信号は図7Bに示すように10進数表現が160, 32, 224になる。これらを浮動小数点表現にすると、仮数部の23ビットの「1」と「0」の配列は、1つ目が上位の2ビット目のみ「1」、2つ目はすべてゼロと、3つ目は上位の1ビット目と2ビット目のみ「1」となる。したがって、図7Aのまま圧縮するよりも、著しく高く圧縮することができる。

実施例2(有効桁・丸め選択)

入力信号がどのような有効桁、丸め手段によって生成されたかによって、 $x(i)$ の性質は異なる。一般に浮動小数点の演算を行う場合の丸め処理としては、単精度のまま演算を行ったり、倍精度で乗算を行った結果を単精度に丸めたりできる。

[0026] 丸めの処理としては、所定の有効桁で切り捨てたり、四捨五入したり、ニアレストタイツウイーブン(nearest tie to even)や切り上げがある。そこで商信号 $y(i)$ に乗数 A を乗じて $x'(i)$ を生成する際に、入力信号 $x(i)$ が生成された際に用いられたのと同じ方法を推定して用いることで、誤差 $z(i)$ を最小化する。具体的には、演算精度(有効桁)、丸めの方法として考えられるものを複数ためし、最も誤差 $z(i)$ が小さくなる方法を

選択する。以下にその具体例を説明する。

図2中の乗算部130において、商信号 $y(i)$ の有効桁(演算精度)及び丸め方法を選択する例を説明する。その要部の機能構成例を図8に、その処理手順例を図9に示す。剰余分離処理部100の乗算部130は、桁設定部130aと丸め設定部130bとが設けられる。有効桁・丸め制御部150は、有効桁数と丸め方法とを桁設定部130aと丸め設定部130bに設定する(ステップL1)。乗算部130では商信号 $y(i)$ と乗数Aとの乗算が設定された有効桁数の倍精度で行われ、その乗算結果に対し、設定された丸め方式により単精度の丸めが行われる(ステップL2)。その結果 $x'(i)$ と入力信号 $x(i)$ との差分が誤差算出部140でとられ、その誤差信号 $z(i)$ が有効桁・丸め制御部150内の記憶部150aに格納される(ステップL3)。

[0027] 次に有効桁・丸め制御部150は、予め想定した各有効桁数について、想定した各種の丸め方法の全てについて乗算部130における処理が行われたかを調べる(ステップL4)。Noならば、ステップL1に戻る。Yesならば、有効桁・丸め制御部150は、記憶部150a内の誤差信号 $z(i)$ 中の最小のものと対応する有効桁数と丸め方法とを、乗算部130内に設定する(ステップL5)。そして、フレームの全てのサンプルについての誤差信号 $z(i)$ を求める処理を行う。またその設定した有効桁数と丸め方法とを示す情報が補助符号 C_B を合成部16へ入力する(ステップL6)。

[0028] この場合の復号化装置の剰余結合処理部200は、図10に簡単に示すようにB復号化部58と桁設定部210aと丸め設定部210bを有する乗算部210を備える。B復号化部58は、分離部52で分離された符号 C_B を復号化する。その有効桁数と丸め方法とは、乗算部210内の桁設定部210a、丸め設定部210bにそれぞれ設定される。乗算部210は、これら設定情報に基づいて、復号化商信号 $y(i)$ と復号化乗数Aとの乗算を行う。

[実施形態2]

実施例3(変形商信号)

次に商信号 $y(i)$ を変形処理することにより、誤差信号 $z(i)$ の情報量を減少する他の例を説明する。その要部の機能構成例を図11に、処理手順例を図12に示す。

[0029] 本実施例は、符号化側の誤差算出部140から出力される誤差信号 $z(i)$ が最小とな

るように、除算処理部120で得た商信号 $y(i)$ を変形処理する部分が実施例1(図2)と異なる。

例えば、除算処理部120で、 $y(i)$ を所定の有効桁まで算出する場合に、有効桁より下位の桁で切り捨て、あるいは四捨五入などを行った場合には、 $y(i)$ は丸め誤差を含む。同様に復号化装置(図4)の復号化手順で、乗算部210で復号化した $y(i)$ と復号乗数 A を掛け合わせる際にも、丸め誤差が生じる場合がある。この2度の丸め処理によって、誤差が累積され、誤差信号 $z(i)$ が大きくなる場合がある。

[0030] 本実施例では、復号化時の乗算処理によって生じる丸め誤差を予め考慮して $y(i)$ を $y'(i)$ に変形しておくことによって、誤差信号 $z(i)$ の情報量が低減するようにする。

各入力サンプル $x(i)$ に対して独立に商信号 $y(i)$ を変形し、最も誤差信号 $z(i)$ が小さくなるような変形誤差信号を $y'(i)$ とする。このことにより、乗数推定部110において推定された乗数 A が誤差を含むような場合にも、乗数 A の推定誤差の影響を低減することが出来る。例えば、1フレームの1024サンプル $x(i)$ のほとんどが、有効桁の範囲内で乗数 A で割り切れ、一部のサンプル $x(i)$ だけが乗数 A で割り切れないような場合に、 $y(i)$ に対する前記変形操作を行うことで誤差信号 $z(i)$ の情報量を低減出来る。

[0031] 除算処理部120は、 $x(i)$ を A で割算し、丸め処理した商信号 $y(i)$ を、変形処理部160に渡す。

変形処理部160は、除算処理部120から渡された $y(i)$ を変形して、 $z(i)$ が最小となるように変形した $y'(i)$ を決定する。 $y'(i)$ を決定する手順を図12により簡略に説明する。(詳細は後で示す。)入力信号サンプル $x(i)$ の絶対値が乗算部130の乗算結果の絶対値 $|Ay'(i)|$ より大きいかを調べる(ステップL11)。大きければ、 $y'(i)$ の最下位ビットに1を加算する(ステップL12)。たとえば、 $y'(i)$ が2進数表現で

$$+1.000000000000000000111111 \times 2^0$$

の場合には、最下位ビットに1を加算することで $y'(i)$ の値は、

$$+1.0000000000000000001000000 \times 2^0$$

となる。ステップL11で $|x(i)|$ が $|Ay'(i)|$ より大きくなければ、 $y'(i)$ の最下位ビットから1を減算する(ステップL13)。ステップL12又はL13の後に、変更された $y'(i)$

)と乗数Aとを乗算部130で乗算処理する(ステップL14)。乗算処理結果の絶対値 $|Ay'(i)|$ が入力サンプルの絶対値 $|x(i)|$ と等しいかを調べる(ステップL15)。等しくなければ、ステップL11の判定に変化があったかを調べる(ステップL16)。例えば前回の判定が $|x(i)| > |Ay'(i)|$ であったが、今回は $|x(i)| < |Ay'(i)|$ に変化した場合は、前回の誤差信号 $z(i)$ と今回の誤差信号 $z(i)$ との小さい方をZ符号化部14へ出力し、また対応する $y'(i)$ をY符号化部13へ出力する(ステップL17)。従って、誤差算出部140からの誤差信号 $z(i)$ はステップL17で比較し、出力できるように、前回の分は常に保持しておく。なお $y'(i)$ の初期値は $y(i)$ を用いる。

[0032] ステップL16でステップL11の判定に変化がなければステップL11に戻る。またステップL15で両者が等しければ誤差信号 $z(i) = 0$ とし、その入力サンプル $x(i)$ に対する $y(i)$ の変形処理を終了する。変形処理部160にはステップL11, L15, L16, L17の判定を行う判定部160a、ステップL12, L13における前回の $y'(i)$ に対する変形とその保持を行う増減部160b、ステップL17で必要とする誤差信号の記憶部160cなどが設けられている。

なお誤差算出部140からの出力 $z(i)$ の極性(符号)に基づきステップL12又はL13と同様の変形処理を行うように $z(i)$ を変形処理部160に帰還して誤差信号 $z(i)$ が最小になるような $y'(i)$ を決定してもよい。要は変形処理部140の処理としては、誤差信号 $z(i)$ が最小になるように $y(i)$ の有効桁の最小桁を最小単位ずつ増減させて $y'(i)$ を決定する。

[0033] ここで、誤差信号 $z(i)$ が最小となる $y'(i)$ の候補が複数存在する場合には、 $y'(i)$ の最もLSB(最下位ビット)側の0でないビットが最もMSB(最上位)側にあるものを選択する($y'(i)$ の構成ビットのうち0でないビットの数を少なくすることによって符号化装置の圧縮効率が向上する場合がある。)

ここで、誤差が最小となる $y'(i)$ が複数存在する場合に、構成要素のビットで、1が最も少ない $y'(i)$ を選択するようにしてもよい。

以上のようにして $y(i)$ が変形された $y'(i)$ がY符号化部13へ入力される。なお、除算処理部120では丸めを行わず、 $y(i)$ の所望の有効桁よりも多い有効桁の精度で除算結果を算出しておき、乗算部130における乗算結果に対する丸めに基づく誤差

を含む全体の誤差信号 $z(i)$ が最小となるように $y(i)$ を変形した $y'(i)$ を求めてもよい。つまり、符号化装置における乗算部130の乗算処理が、復号化装置で行われる乗算部210の乗算処理と同じ結果となっていれば、丸め処理を除算処理部120で行っても、変形処理部160で変形と同時に行ってもどちらでもよい。

実施例4(変形商信号)

この実施例4も商信号 $y(i)$ に対し変形処理を行うが、 $y'(i)$ を決定する際の評価尺度が実施例3と異なる。変形処理部170の処理として、Y符号化部13、Z符号化部14で可逆符号化を用いる場合に、符号 C_Y と、符号 C_Z の合計の符号量が最も小さくなるように、 $y(i)$ を変形して $y'(i)$ とする。例えば、 $y'(i)$ の有効桁を所定の範囲で変化させ、出力符号サイズの総和が最も小さくなるような $y'(i)$ を探索する。

[0034] 具体的には、その要部の機能構成例を図13に、処理手順例を図14にそれぞれ示す。除算処理部120からの商信号 $y(i)$ は、変形処理部170に入力される。まず $y(i)$ と対応する誤差信号 $z(i)$ とをそれぞれ符号化する(ステップL21)。符号 C_Y の符号量 V_Y と符号 C_Z の符号量 V_Z との加算符号量(ビット数) $V_Y + V_Z$ を求める(ステップL22)。次に $y(i)$ の有効桁の最下位ビットに対し、最小単位だけ予め決めた変形、例えば1の加算を行う(ステップL23)。この変形商信号 $y'(i)$ と、この $y'(i)$ に基づき求めた誤差信号 $z(i)$ とをそれぞれY符号化部13、Z符号化部14で符号化する(ステップL24)。符号 C_Y 、 C_Z の符号量の和(ビット数)を求める(ステップL25)。

[0035] この符号量が前回より増加したかを調べる(ステップL26)。増加していれば、 $y(i)$ に対する変形を逆にする、つまりステップL23で最下位ビットに1を加算した場合は、最下位ビットから1の減算を行う。この変形 $y'(i)$ と対応 $z(i)$ を符号化し(ステップL28)、これらの加算符号量を求め(ステップL29)、符号量の変化が減少していたか、増加に変化したかを調べる(ステップL30)。

増加しなければ、 $y'(i)$ をそれまでと同一方向に更に変形してステップL28に戻る(ステップL31)。ステップL26で符号量が増加していなければ、ステップL31に移る。ステップL30で符号量の変化が増加に変化したと判定されると、その直前の符号量と今回の符号量との少ない方と対応する符号 C_Y と C_Z を合成部16へ出力する(ステップL32)。

[0036] 変形処理部170内には、ステップL22, L25, L29で符号 C_Y と C_Z の符号量を加算する符号量加算部170a、ステップL26, L30の判定、ステップL32内の大小判定などを行う判定部170b、ステップL24, L27, L31の $y(i)$ に対する変形を行う増減部170c、ステップL32の符号量の判定と、前回の C_Y, C_Z を出力するためなどに利用される記憶部170dなどが設けられる。

例えば $y(i)$ のビット列に1が連続しているような場合に、あえて1を加えると順次桁上りにより0の連続に変換することが出来る。この場合には、加えた1をA倍しただけ誤差信号 $z(i)$ は増加するが、 $y'(i)$ の符号化効率が良くなるので、全体として符号量は減少する。

実施例5(商信号変形)

この実施例では、例えば図15に要部を示すように $x(i)$ をAで割った商信号 $y(i)$ を変形処理部180内の圧縮判定部180aで観測し、例えばY符号化部13においてうまく、つまり効率的に圧縮できるか条件に合っているかを判定し、条件に合っていなかったサンプルについては $y'(i) = 0$ とする。つまり条件に合っていなければ圧縮判定部180aの出力によりスイッチ180bが0信号源180c側に切替えられ、 $y'(i) = 0$ に変形される。 $y(i)' = 0$ の場合は乗算部130の出力が0となり、誤差算出部140からは $z(i) = x(i)$ が出力される。 $y(i)$ が条件に合っていると圧縮判定部180aで判定されると、スイッチ180bは除算処理部120に切替えられ接続され、 $y(i)$ がそのまま $y'(i)$ として変形処理部180から出力される。

[0037] このようにして、あるサンプルについては $y(i)$ を圧縮するよりも $x(i)$ を圧縮した方が効率よく圧縮することができる場合に、全体の圧縮率を向上させることができる。

なお、実施例4も5も、 $x(i)$ から $x'(i)$ を差し引いた誤差が最小になるように $y(i)$ を変形しているともいえる。

[実施形態3]

実施例6(非可逆符号化)

実施例6は、商信号 $y(i)$ に対し非可逆符号化方法により符号化する例である。その機能構成例を図16に示す。除算処理部120よりの商信号 $y(i)$ は非可逆圧縮符号化方法の非可逆Y符号化部31で符号化される。その符号化された符号 C_Y を復号化部

32で、復号化し、その復号化された信号 $y(i)$ に対し、乗数 A を乗算処理し、その結果 $x'(i)$ と $x(i)$ との誤差信号 $z(i)$ を得る。

[0038] 復号化装置での処理は、図4に示した処理と同様に行えばよい。ただし、Y復号化部53は、非可逆Y符号化部31の非可逆圧縮符号化方法と対応する非可逆伸張復号化方法のものを用いる。

このように、非可逆Y符号化部31、Y復号化部として、既知の非可逆符号化方式を用いた場合にも、符号化部14、15、復号化部54、55の処理にそれぞれ可逆符号化方法とこれと対応する可逆伸張復号化を用いることによって、全体として可逆符号化を行うことが出来る。Z符号化部14、Z復号化部54にも非可逆符号化を用いた場合には、全体としても非可逆符号化となる。

[0039] 図16中に破線で示すように、除算処理部120と非可逆Y符号化部31との間に変形処理部35を挿入し、誤差算出部34からの誤差信号 $z(i)$ を変形処理部35に入力して、誤差信号 $z(i)$ が最小になるように $y(i)$ を変形して非可逆Y符号化部31へ入力するようにしてもよい。同様に全体の符号化サイズ、つまり符号化データ C_x の符号量が最小化するように変形処理部で $y(i)$ を、符号化データ C_x のサイズに応じて変形し、非可逆Y符号化部31へ供給するようにしてもよい。

[実施形態4]

実施例7

先に述べたように、商信号 $y(i)$ と誤差信号 $z(i)$ 、乗数 A に分離してそれぞれを符号化したほうが、 $x(i)$ をそのまま圧縮符号化する場合と比較して圧縮効率が良い場合には、分離して符号化を行い、 $x(i)$ をそのまま符号化したほうが圧縮効率が良い場合には、 $x(i)$ をそのまま圧縮符号化する例を実施形態4とし、その1例を図17に符号化装置を、図18に復号化装置をそれぞれ示す。

[0040] 図17の分離符号化部300は、剰余分離処理部310、可逆符号化方式のY符号化部320、可逆符号化方式のZ符号化部330、必要に応じて設けられるA符号化部340、合成部350を備え、これらの組は実施例1～5に示したいずれかと対応するものである。直接符号化部360は、入力信号 $x(i)$ を直接符号化するものである。分離符号化部300で符号化された符号化データ C_{sx} のデータ量(情報量、ビット数) V_s と、直接

符号化部360で符号化された符号化データ C_{DX} のデータ量 V_D とが圧縮率比較部370で比較される。そして選択部380は、データ量が少ない方の符号化データを選択する。合成部390は、選択された符号化データと、いずれを選択したかを示す選択符号(1ビット) C_C とを合成して、符号化データ C_X として出力する。

- [0041] 復号化装置においては図18に示すように、符号化データ C_X は分離部400で符号化データと選択符号 C_C とに分離される。符号 C_C が分離符号化データ C_{SX} であることを示していれば、切替部410が制御され、符号化データ C_{SX} が分離復号化部420に入力される。符号 C_C が直接符号化データ C_{DX} であることを示していれば、切替部410が制御され、直接符号化データ C_{DX} が直接復号化部430に入力される。

分離復号化部420内の分離部440、可逆復号化方式のY復号化部450、可逆復号化方式のZ復号化部460、必要に応じて設けられるA復号化部470、乗算部480、誤差補正部490の組は、実施例1～5の復号化装置のいずれかと対応するものである。切替部510は、選択符号 C_C に応じて、分離復号化部420からの復号信号または直接復号化部430からの復号信号を出力する。なお、切替部510は省略し、両復号化部420、430の両出力端を接続して、復号化処理がなされた信号が出力されるようにしてもよい。

- [0042] 直接符号化部360を独立に設けることなくまた選択符号 C_C を省略してもよい。例えば、図19に示すようにY符号化部321に剰余分離処理部310からの商信号 $y(i)$ だけでなく、乗数Aも入力する。この場合、Y符号化部321の判定部321aは、乗数Aが1.0か否かを判定する。乗数Aが1.0でなければ切替部321bが制御され、商信号 $y(i)$ が商符号化部322に入力され、商信号 $y(i)$ に適した符号化が行われる。判定部321aが乗数A=1.0と判定すると、切替部321bが制御されて商信号 $y(i)$ 、つまり入力信号 $x(i)$ が直接符号化部323に入力される。この場合の商信号 $y(i)$ は除算処理部120の処理出力はAが1.0であるから $y(i) = x(i)$ であり、 $x(i)$ の符号化に適する符号化が直接符号化部323で行われることになる。選択符号 C_C は出力されない。また、選択部380、合成部390も省略され、図17に示した構成より簡単になる。

- [0043] 対応する復号化装置を図20に示す。分離部440で分離された符号 C_Y は、Y復号化部451に入力されると共に、分離された乗数A(又は符号 C_A)もY復号化部451に

入力される。判定部451aで乗数Aが1.0でないと判定されると、切替部451bは、符号 C_Y を商復号化部452に入力する。そして、商復号化部452が符号 C_Y に適した復号を行う。判定部451aがA=1.0と判定されると、切替部451bは、符号 C_Y を直接復号化部453に入力する。そして、直接復号化部453が符号 C_Y に適する復号を行う。Y復号化部451からの復号化出力y(i)は、乗算部480に入力され、乗数Aとの乗算処理が行われる。図18に示した構成より簡単となっている。

実施例8

実施例8は分離符号化と、直接符号化の他の例であってその概略を図21にて示す。ここで用いられる分離符号化部301は、非可逆Y符号化部31として非可逆圧縮符号化を用いる場合である。この場合は、復号側から考えると、符号化サイズ(符号化量)よりも符号化波形歪が小さいことが重要である。

[0044] 分離符号化部301からのY符号 C_Y 、誤差符号 C_Z 及び乗数A(又はその符号 C_A)からなる符号化データ C_{SX} とは、局部復号化部302で復号化され、復号化信号列X'が得られる。直接符号化部360からの符号化データ C_{DX} は、局部復号化部303で復号化され、復号化信号列X"が得られる。歪計算部304は、局部復号化信号列X'と原入力信号列Xとの波形歪を計算する。歪計算部305は、局部復号化信号列X"と原入力信号列Xとの波形歪を計算する。これら計算された歪中の小さいものが判定部306で判定される。選択部380は、その判定結果に基づき、歪が小さい方の符号化データを選択する。合成部390は、選択部380からの符号化データと、判定部306からのいずれを選択したかを示す選択符号 C_C とを合成し、符号化データ C_X を出力する。

[実施形態5]

実施形態5は剰余分離処理により得られた、商信号列Y及び/又は誤差信号列Zを再帰的に剰余分離処理する。

実施例9(符号化1)

図22にその機能構成例を示す。入力信号列Xは、剰余分離処理部100に入力されて、商信号列 Y_1 、誤差信号列 Z_1 、乗数 A_1 に分離される。その商信号列 Y_1 は剰余分離処理部101Yで更に商信号列 Y_2 、誤差信号列 Z_2 、乗数 A_2 に分離される。その商信号列 Y_2 は剰余分離処理部102Yで商信号列 Y_3 、誤差信号列 Z_3 、乗数 A_3 に分

離される。

- [0045] この例では3回の再帰的分離を行った。最後の商信号列 Y_3 は、Y符号化部103Yで可逆圧縮符号化され、符号 C_{Y3} として出力される。誤差信号列 Z_1, Z_2, Z_3 は、それぞれZ符号化部103Z₁, 103Z₂, 103Z₃で可逆圧縮符号化され、符号 C_{Z1}, C_{Z2}, C_{Z3} としてそれぞれ出力される。乗数 A_1, A_2, A_3 は、そのまま、あるいは可逆符号化されて C_{A1}, C_{A2}, C_{A3} として出力される。合成部16は、符号 $C_{Y3}, C_{Z1}, C_{Z2}, C_{Z3}, A_1$ (又は C_{A1}), A_2 (又は C_{A2}), A_3 (又は C_{A3})を合成し、符号化データ C_X として出力する。
- [0046] 誤差信号列 Z_1 について同様に再帰的に符号化してもよい。その例を図22中に破線で示す。誤差信号列 Z_1 は、剰余分離処理部101Zで商信号列 $Y(Z_1)$ 、誤差信号列 $Z(Z_1)$ 、乗数 $A(Z_1)$ に分離される。この誤差信号列 $Z(Z_1)$ は、剰余分離処理部102ZZ₁で商信号列 $Y(ZZ_1)$ 、誤差信号列 $Z(ZZ_1)$ 、乗数 $A(ZZ_1)$ に分離される。この場合は、合成部16は、商信号 Y_1 の可逆符号化符号 C_{Y1} と、商信号列 $Y(ZZ_1)$ の可逆符号化部104Yにより符号化された符号 C_{YZZ1} と、誤差信号列 $Z(Z_1), Z(ZZ_1)$ をそれぞれ符号化部104Z₁, 104Z₂で可逆符号化された符号 C_{ZZ1}, C_{ZZZ1} と、乗数 $A(Z_1)$ (又はその符号 C_{AZ1}), $A(ZZ_1)$ (又はその符号 C_{AZZ1})とを合成し、符号化データ C_X として出力する。
- [0047] 更に、剰余分離処理部で分離された商信号と誤差信号との両者を、再帰的に分離符号化してもよい。その場合図22の例では破線で示すように、誤差信号 Z_2 は剰余分離処理部102Zで商信号列 $Y(Z_2)$ 、誤差信号列 $Z(Z_2)$ 、乗数 $A(Z_1)$ に分離される。 $Y(Z_2), Z(Z_2)$ は、それぞれ符号化部105Y, 105Zで可逆符号化され、符号 C_{YZ2}, C_{ZZ2} が出力される。また、剰余分離処理部101Zで分離された商信号列 $Y(Z_1)$ は、剰余分離処理部102YZ₁で商信号列 $Y(YZ_1)$ 、誤差信号列 $Z(YZ_1)$ 、乗数 $A(YZ_1)$ に分離される。商信号列 $Y(YZ_1)$ 、誤差信号列 $Z(YZ_1)$ は、それぞれ符号化部106Y, 106Zで符号化されて、可逆符号 C_{YYZ1}, C_{ZYZ1} が出力される。
- [0048] 結局、合成部16は、符号 $C_{Y3}, C_{Z3}, C_{YZ2}, C_{ZZ2}, C_{Z2}, C_{YYZ1}, C_{ZYZ1}, C_{YZZ1}, C_{ZZZ1}, C_{Z}$ の他に乗数 $A_3, A(Z_1), A_2, A(YZ_1), A(ZZ_1), A(Z_1), A_1$ 又はこれらの符号を合成して、符号化データ C_X とする。

実施例10(符号化2)

このように再帰的分離符号化は、対象とする商信号又は誤差信号が分割できなくなるまで繰り返す。あるいは分割による符号のサイズ(符号化データ C_X のビット数)が減少する効果が得られなくなるまで行う。

[0049] 実施例10では、商信号列 Y を再帰的に分割する場合を示す。図23にその処理手順を示す。まず、入力信号列 X に対し剰余分離処理をして分離符号化して C_Y 、 C_Z 、 A (又は C_A)よりなる符号化データ C_{SX} を求める。また、入力信号 X 列を直接符号化して、符号化データ C_{DX} を求める(ステップs1)。符号化データ C_{SX} と C_{DX} のサイズを比較して、分離符号化を選択するかを判定する(ステップs2)。 C_{SX} の方が、データ量が少なく、分離符号化が選択されると、その分離された商信号列 Y に対し剰余分離処理をして分離符号化し商符号 C_{Yi} 、誤差符号 C_{Zi} 、乗数 A_i (又はその符号 C_{Ai})からなる符号化データ C_{Xi} を求める(ステップs3)。

[0050] この状態での出力符号化データ C_X のデータ量として、現在の処理で求めた C_{Yi} と今までの処理で求めた全ての誤差符号 C_{Zi} と全ての乗数 A_i との合計のビット数 V_{Si} (符号化データのデータ量)を計算する(ステップs4)。データ量 V_{Si} と直接符号化の符号化データ C_{DX} のデータ量 V_D とを比較し、データ量 V_{Si} の方が小さい、かつ前回の分離符号化データ量 V_{Si-1} より小さいことを調べる(ステップs5)。この条件を満たせば、現在の処理で求めた C_{Zi} と A_i を記憶部に記憶してステップs3に戻る(ステップs6)。

ステップs5で V_D より小さいが V_{Si-1} より大であるか、 V_D より小でなければ、直前処理で得られた C_{Yi} と、直前処理までに得られた全ての C_{Zi} と A_i を符号化データ C_X として出力する(ステップs7)。ステップs2でデータ量 V_D の方が小さければ、直接符号化データ C_{DX} を符号化データ C_X として出力する(ステップs8)。

実施例11(復号化1)

図22中に示した商信号列 Y を再帰符号化した符号化データ C_X を復号する例を図24に示す。分離部52からの符号 C_{Y3} は、商復号化部71で復号化され、 Y_3 が出力される。乗算部210₁は、この Y_3 に分離部52からの乗数 A_3 を乗算処理する。誤差復号化部72₁は、誤差符号 C_{Z3} を復号化し、誤差信号列 Z_3 を得る。補正部220₁は、乗算部210₁の出力を復号化された誤差信号列 Z_3 で補正し、商信号列 Y_2 が得られる。

[0051] 乗算部210₂は、 Y_2 と分離部52からの乗数 A_2 との乗算処理を行う。誤差復号化部7

2₂ は、分離部52からの誤差符号C_{z2}を復号化し、誤差信号列Z₂を得る。補正部220₂は、乗算部210₂からの出力を、誤差信号Z₂で補正し、商信号列Y₁を得る。このY₁に対し、分離部52よりの乗数A₁が乗算部210₃で乗算処理される。誤差復号化部72₃は、分離部52よりの誤差符号C_{z1}を復号化し、誤差信号列Z₁を得る。補正部220₃は、乗算部210₃の出力を誤差信号列Z₁で補正し、原信号列Xを出力する。

実施例12(復号化2)

この再帰符号に対する一般的な復号化処理手順の例を、図25を参照して説明する。まず入力符号化データC_xを各符号に分離して記憶部に記憶する(ステップs11)。次にその記憶部中の未復号中の符号化における最も後段で得られた各符号を選択する(ステップs12)。これら各符号を復号化する(ステップs13)。最初は商信号と誤差信号と乗数が得られ、次回からは、例えば図24に示した例では誤差信号と乗数が得られる。

[0052] これら復号化された信号を用いて剰余結合処理を行う(ステップs14)。つまり復号化された商信号(又は前回の処理で得られた商信号)は乗算処理される。また、その結果は、復号化された誤差信号により補正される。

次に記憶部中に未選択の符号があるかを調べ、あれば、ステップs12に戻り、なければ処理を終了する(ステップs15)。このようにして元入力信号列Xが復号化される。

[実施形態6]

実施例13

先に示した各実施例では、フレーム単位で剰余分離処理部100へ入力した。本実施例では、剰余分離処理部100に入力する前に、図26Aに示すように入力判定部41でx(i)を検査し、剰余分離処理部100へ入力するか否かの判定をする。例えば、入力サンプルが単精度の浮動小数点数で表される場合には、無限大の値や、数値として表すことが出来ないNaNという値であることがある。入力判定部41では、その判断部41aがx(i) = NaNであると判断すると、切替部41bを制御して、0源41cから剰余処理部100にx(i)の代わりに0を出力し、x(i)をX符号化部42に出力する。

[0053] 入力判定部41は、サンプルx(i)が0の場合には、剰余分離処理部100に0を出力

し、同時にX符号化部42にも0を出力する。

X符号化部42は、 $x(i)$ を符号化して得られた符号を合成部16に出力する。

復号化装置は、図26Bに示すようにY復号化部57で $y(i)$ を復元する。判定部57aは、 $y(i) = 0$ と判定すると、X復号化部58に通知する。X復号化部58は、その $y(i)$ に対応し、分離部52から分離された符号を復号して $x(i)$ を出力する。この $x(i)$ は誤差補正部220に入力される。誤差補正部220は、入力された $y(i) = 0$ と対応する $x'(i) = 0$ に対し、前記復号した $x(i)$ を加算する。

実施例14

その実施例3(図11)では、最も誤差 $z(i)$ が小さくなるように $y(i)$ の変形 $y'(i)$ を求める例を示した。

[0054] 入力されたサンプル $x(i)$ が、そもそも割り算が出来ないような値であったり、乗数Aで割ると誤差が大きくなりすぎるような場合がある。例えば、入力サンプルが単精度の浮動小数点数で表される場合には、無限大の値や、数値として表すことが出来ないNaNという値であることがある。

入力信号 $x(i)$ がNaNであった場合には、 $x(i)$ を乗数Aで除算した結果得られた $y(i)$ もNaNとなる。本実施例では、符号化装置は、図11中の変形処理部160に代り図27に変形処理部161を備える。変形処理部161は、判定部161aが、除算処理部120から渡された $y(i)$ が、NaNであったと判定すると、切替部161bを制御して、0源161cからの $y'(i) = 0$ を、Y符号化部13および乗算部130に出力する。

[0055] 乗算部130では、 $y'(i)$ に乗数Aを乗じて $x'(i)$ を算出する。よって、 $y'(i) = 0$ であることから乗算部130の出力は、 $x'(i) = 0$ となり、誤差算出部140の出力 $z(i)$ は $x(i) = NaN$ となる。この場合、誤差算出部140では、誤差を算出する。実施例3では、誤差算出に当たって $x(i)$ と $x'(i)$ の減算処理を行うこととしていたが、 $x(i) = NaN$ の場合には減算は未定義であるため、正しく減算を行うことが出来ない。

そこで、本実施例では、 $y'(i) = 0$ の場合には、誤差信号 $z(i)$ として $x(i)$ をそのまま設定する。なお、図27の変形処理部161中の処理部161dは図11中の変形処理部160と同様な機能をもつ。

[0056] 復号化処理装置の誤差補正部220では、上記と逆の処理を行う必要がある。

すなわち、復元した $y'(i) = 0$ であった場合には、 $y'(i)$ に乗数 A を乗じて得られた $x'(i)$ も 0 となる。誤差補正部220は、 $x'(i)$ が 0 であるから、Z復号化部54で復号化された $z(i) = x(i)$ をそのまま出力する。

この場合も、図26B中に示すように判定部57aが、 $y(i) = 0$ と判定すると、スイッチ261を制御してZ復号化部54からの復号化出力 $z(i) = x(i)$ が $x(i)$ として出力するように設定してもよい。この場合、乗算部130及び誤差算出部140における処理を行う必要がなく、処理が簡単になる。

[0057] なお、図27では、判定部161aが $y(i)$ はNaNであると判定すると、 $y'(i) = 0$ とする。また、スイッチ271はオンとなり、X符号化部272が入力信号 $x(i)$ を符号化する。その符号化出力 C_z とスイッチ273を通じて合成部16に入力することとしてもよい。この場合、復号化装置は、図26B中に示すように、分離部52で分離された符号 C_z をX復号化部58にも入力する。そして、判定部57aが復号した $y(i)$ が 0 であると判定すると、スイッチ262を制御してX復号化部58の復号結果を復号した $x(i)$ として出力してもよい。

[0058] Z符号化部14としては、例えばRice符号化法、代数符号化法などが適している。X符号化部272としては、ハフマン符号化法、LZ符号化法などが適している。Z復号化部54、X復号化部26B1は、対応する符号化部の符号化法に対するものである。

浮動小数点表現では、正規化して1. ……となり、この小数点以下の23ビットが仮数部とされている。また指数の最小値は -127 であり、指数が -127 の場合は 0 か非正規化数を表す。指数が -126 で、仮数部の最下位ビットのみが 1 の場合は浮動小数点で正規化表現できる絶対値の最小値である。絶対値がこれ以下の数値は、浮動小数点表現のための正規化を行うことができない。入力信号 $x(i)$ が浮動小数点表現されたサンプル列であった場合には、NaNの他に、 $x(i)$ が非正規化数である場合がある。つまり指数が -127 であり、かつ仮数部の最上位ビットの1つ上位ビットが 1 でない場合(IEEE-754形式の2進浮動小数点表現で正規化された形式により表わせる最小値より小さい場合)がある。 $x(i)$ が非正規化数であるような場合に、さらに $x(i)$ を乗数 A で割るとアンダーフローにより情報が失われる。したがって、 $y(i)$ として符号化する際に効率よく圧縮符号化が出来ない場合がある。

[0059] これを避けるために、変形処理部161中の判定部161aとして、 $x(i)$ が非正規化数も判定できるようにする方法がある。この方法では、 $x(i)$ が非正規化数であったと判定されると、変形処理部161は $y(i)$ を $y'(i) = 0$ に変形してY符号化部13および乗算部130に出力する。

乗算部130は、 $y'(i)$ に乗数Aを乗じて $x'(i)$ を算出する。したがって、 $x'(i) = 0$ となり、復号化した $z(i) = x(i)$ がZ符号化部14に入力される。先の場合と同様に、非正規化数であったと判定して $y'(i) = 0$ にした場合には、スイッチ271, 273を制御して復号化された $z(i) = x(i)$ をそのまま復号化 $x(i)$ として設定してもよい。この場合の復号化装置の処理は、図26Bを参照して述べたと同様である。

[0060] また、入力信号 $x(i)$ が整数で、乗数が $A < 1$ である場合に、 $x(i)$ を乗数Aで割って得られた商信号 $y(i)$ が、表現可能な範囲を超えてしまい、オーバーフロー状態となってしまう場合がある。従って、例えば図27中の変形処理部161の判定部161aは、 $y(i)$ のオーバーフローが検出された場合にも、 $y'(i) = 0$ とする。この場合、復号化装置で復号した商信号 $y'(i)$ が0であれば、前述と同様に、符号 C_z の復号化信号 $z(i)$ をそのまま復号化信号 $x(i)$ とする。

[実施形態7]

この発明と非特許文献2に示した技術とを組み合わせた実施形態7を説明する。

実施例15(符号化処理)

図28に実施例15の符号化装置の機能構成例を示す。この実施例15では、除算処理部120からの出力 $y(i)$ が整数化部281に入力される。整数化部281は、 $y(i)$ を整数形式の信号 $y_1(i)$ に変換する。なお、 $y(i)$ をそのまま整数形式に変換するのではなく、変形処理部で変形処理してから変換してもよい。整数形式信号 $y_1(i)$ は、出力する最大振幅の桁数が24ビット、あるいは16ビットと定められている場合もある。その整数形式信号 $y_1(i)$ は、その最下位桁に対し丸め処理がなされる。

[0061] この丸め処理された整数形式信号 $y_1(i)$ が、前述した各種実施例と同様にY符号化部13に出力される。

この実施例では、浮動小数点化部282は $y_1(i)$ を浮動小数点表現に変換し、乗算部130へ出力する。そして、この浮動小数点表現の整数に対し、乗数Aが乗算される

。誤差算出部140は、その乗算結果によって誤差信号生成処理を前述と同様に行う。

本実施例では、剰数判定部283は、乗数Aが1.0であるかの判定を行う。A=1.0であればスイッチ284が切替えられて、入力信号x(i)が誤差算出部140ではなく、差分生成部285に入力される。整数化部281内に設けられた桁数計算部281bは、整数形式信号の桁数nを計数する。この桁数nも差分生成部285に入力される。差分生成部285は、浮動小数点入力信号x(i)の極性Sと指数部E、仮数部Mの下位(23-n)ビットはそのままとして、仮数部中の上位nビット分をすべて0とした出力信号z'(i)を生成する。なお仮数部の整数部分の桁数 $n = E - E_0$ は整数形式信号y1(i)の最上位の“1”より下位の桁数と等しい。

[0062] この差分信号z'(i)は、Z'符号化部286で可逆符号化される。

従って、合成部16には、符号 C_Y と、符号 C_A (又はA)と、符号 C_Z (A=1.0であればZ'符号化部286の出力、A=1.0でなければZ符号化部14の出力)が入力されることになる。

実施例16(復号化処理)

図28に示した符号化装置と対応する復号化装置を図29に示す。分離部52で分離された符号 C_Y は、Y復号化部291で復号化され、整数形式信号y1(i)が得られる。整数形式信号y1(i)は、浮動小数点化部292で浮動小数点信号とされる。A復号化部55からの乗数Aは、剰数判定部293でA=1.0か否かが判定される。Aが1.0でなければ、スイッチ294がZ復号化部54側に切替えられ、このZ復号化部54で復号された誤差信号z(i)が補正部220に供給される。

[0063] Aが1.0の場合は、スイッチ294はZ'復号化部295側に切替えられ、符号 C_Z が復号化される。Z'復号化部295には、Y復号化部291内の桁数計算部291aで計算された整数形式信号の桁数nも入力される。Z'復号化部295は、浮動小数点表現における23ビットの仮数部の上位nビットに“0”を埋め、下位23-nビットに符号 C_Z の復号信号を設定する。組立部295aは、前記仮数部と、復号された符号ビットSと、復号された指数部 $E - E_0$ から、浮動小数点形式の差分信号(誤差信号)z(i)を組立てる。この差分信号z(i)が補正処理部220に入力される。

[0064] なお、スイッチ294、Z復号化部54、Z'復号化部を一体として、Z復号化部54'としても良い。この場合は、乗数判定部293は図29のように独立した構成部としても良いし、Y復号化部291の一部とし、整数形式信号の桁数nと一緒に乗数Aの値をZ復号化部54'に供給しても良い。

図28の変形処理部281aにより $y(i) = 0$ に変形処理される場合は、図28に破線で示すように、y変形判定部287で $y_1(i) = 0$ に変形したことが判定されると、スイッチ288を制御して、入力信号 $x(i)$ をX符号化部272で直接符号化し、その符号化結果を C_z として合成部16へ供給してもよい。

[0065] 図29中に破線で示すように、変形判定部296が、復号化された $y_1(i)$ が0かを判定する。 $y_1(i) = 0$ と判定された場合は、スイッチ297が、補正処理部220側からX'復号化部26B1側に切替えられる。そして、X'復号化部26B1で復号化された符号 $z(i)$ が復号化信号 $x(i)$ として出力される。

実施例17(符号化処理)

この実施例は、剰余分離処理部600の出力に対して非特許文献2に示した方法を適用した例である。図30にこの実施例の符号化装置の機能構成例を示す。剰余分離処理部600内の(入力判定)剰数推定部301で乗数Aを推定する。除算処理部120は、入力信号を除算処理する。変形処理部302は、除算結果 $y(i)$ を変形処理し、処理結果 $y'(i)$ を出力する。乗算部130は、 $y'(i)$ と乗数Aとを乗算する。更に、誤差算出部140は、その乗算結果と入力信号との誤差算出処理を行い、誤差信号 $z(i)$ を出力する。

[0066] 整数誤差分離部303内の整数化部303aは、変形処理された $y'(i)$ を、必要に応じて最大振幅が予め決められた整数形式信号 $y_1(i)$ に変換する。また、 $y'(i)$ 中の整数値未満(小数点以下)の部分が浮動小数点誤差信号 $y_2(i)$ として出力される。Y符号化部13は、整数形式信号 $y_1(i)$ を符号 C_Y に符号化する。誤差符号化部304は、浮動小数点誤差信号 $y_2(i)$ を可逆符号化し、符号 C_{2Z} として出力する。剰余分離処理部600からの誤差信号列Zは、Z符号化部14へ入力される。また、乗数Aは必要に応じて符号化されて合成部16に入力される。

実施例18(復号化処理)

実施例17の符号化装置に対応する復号化装置の機能構成例を、図31に示す。Y復号化部53は、整数形式信号である復号化信号 $y_1(i)$ を、浮動小数点化部312に出力する。誤差復号化部311は、分離部52からの符号 C_{2Z} を、浮動小数点形式の復号化信号 $y_2(i)$ に復号化し、浮動小数点化部312に出力する。浮動小数点化部312は、 $y_1(i)$ と $y_2(i)$ とを組み合わせ、最大振幅が前記予め決められた浮動小数点信号 $y'(i)$ を生成する。剰余結合処理部200は、 $y'(i)$ に乗数Aを乗算し、Z復号化部54で復号化された誤差信号を加算する。その他の処理は先の実施例と同じである。

[0067] 図30の誤算信号 $y_2(i)$ と誤差信号 $z(i)$ とは、いずれも小さい振幅の信号であり、比較的似た性質があるため、誤差符号化部304とZ符号化部14は破線で囲んで示すように1つのエントロピ符号器で符号化してもよい。例えば、フレーム中の $y_2(i)$ を全て符号化した後、 $z(i)$ を符号化する方法がある。このように1つの符号化器を用いる場合は、復号化装置でも、図31中に破線で囲んで示しているように、1つの対応する復号化器で復号してもよい。

図28の変形処理部281aや図30の変形処理部302としては、実施例2～7、9～14に説明した各種変形方法のいずれか1つ又はその中のいくつかを同時に用いるものであってもよい。更に、(入力判定)乗数推定部301では、実施例13(図26A)の入力判定部41と同様に、入力信号 $x(i)$ が無限大の値やNaNの場合、これを判定して、 $x(i)=0$ とし、乗数推定を行う。つまり、剰余分離処理部600からは、 $y'(i)=0$ を出力する。また、スイッチ305をオンにしてその時の $x(i)$ をX符号化部42で直接符号化し、合成部16へ出力する。復号化装置では、図26Bと同様に、判定部57aが復号化した $y(i)$ が0であると判定すると、スイッチ313をオンにする。X復号化部58は、分離部52からの分離符号を復号化して、 $x(i)$ を出力する。また、剰余分離処理部600内の変形処理部302でも実施例2～7、9～14で説明した処理と同様のことがなされ、これと対応する処理が復号化装置においても、図29に示したと同様に行われる。

[0068] 実施例15～18では、可逆符号化について説明した。実施例6(図16)、8(図21)でそれぞれ説明した非可逆符号化についても、実施例15～18に適用することもできる。また、実施例9～12(図22～図25)の再帰処理も、実施例15～18に適用することもできる。符号 Z' 、符号 Z はそれぞれ符号化することなく上記差分信号、誤差信号

をそのまま出力してもよい。つまり符号化しても圧縮効果がほとんど得られない場合もある。

[実施形態8]

乗数Aを有理近似で決定する手法の実施形態を以下に説明する。

実施例19

図32に、複数の数値の集合の共通乗数を求める処理手順を示す。まずフレームごとに乗数Aを新たに推定するか否かを決定する。現フレームの全入力信号 $x(i)$ を入力する(ステップ32-1)。前フレームの乗数Aが1.0でないかを調べる(ステップ32-2)。1.0でなければその現フレームの乗数Aの候補とする(ステップ32-3)。全ての入力信号 $x(i)$ についてその乗数Aが妥当であるかを調べ、妥当でなければその乗数を $A=1.0$ に変更する(ステップ32-4)。乗数Aが1.0でないかを調べる(ステップ32-5)。1.0でなければそのフレームの乗数候補を現フレームの乗数Aとする(ステップ32-6)。

[0069] ステップ32-2で前フレームの乗数Aが1.0、またはステップ32-5で乗数Aが1.0であれば、各入力信号を用いて有理近似で乗数Aを推定する(ステップ32-7)。その乗数Aが1.0でないかを調べ、1.0でなければステップ32-4に移る(ステップ32-8)。

次に図32中のステップ32-7で推定した乗数Aについて調べ、かつ除算処理結果 $y(i)$ を変形する手順の例を、図33を参照して説明する。全ての $x(i)$ のチェックを完了したかを調べる(ステップ33-1)。調べてないものがあればその入力信号 $x(i)$ がNaNであるか、無限大であるか、0であるか、非正規化数であるかを調べる(ステップ33-2)。Yesであれば $y(i)$ を0にし、ステップ33-1に戻る(ステップ33-3)。

[0070] ステップ33-2で前記条件外であれば(例外の値でなければ)、その信号 $x(i)$ を推定した乗数Aで割算して $y1$ とする(ステップ33-4)。 $y1$ の仮数部の最下位ビット(LSB)を0でマスクして $y2$ とする(ステップ33-5)。 $y2$ に乗数Aを掛算し、その結果を $x1$ とする(ステップ33-6)。信号 $x(i)$ から $x1$ を引算し、その結果を z とする(ステップ33-7)。

この引算の余り z が0でないかを調べる(ステップ33-8)。0でないならば、 $x1$ の絶

対値が信号 $x(i)$ の絶対値より小さいかを調べる(ステップ33-9)。 $x1$ の絶対値の方が小さければ $y1$ の仮数部最下位ビット(LSB)に1を足してその値を $y1$ とおく(ステップ33-10)。その $y1$ の仮数部最下位ビットを0でマスクして $y2$ とする(ステップ33-11)。その $y2$ に乗数 A を掛算してその結果を $x1$ とし、ステップ33-9に戻る(ステップ33-12)。

[0071] ステップ33-9で、 $x1$ の絶対値の方が小さくなければ、 $x1$ の絶対値が $x(i)$ の絶対値よりも大きいかを調べる(ステップ33-13)。 $x1$ の絶対値の方が大きければ、 $y1$ の仮数部最下位ビットから1を引算し、その結果を $y1$ とする(ステップ33-14)。その $y1$ の最下位ビットを0でマスクした値を $y2$ とする(ステップ33-15)。その $y2$ に乗数 A を掛算して、その結果を $x1$ とする(ステップ33-16)。次に信号 $x(i)$ から $x1$ を引算し、結果を z とする(ステップ33-17)。その引算した結果 z が0かを確認する(ステップ33-18)。 $z \neq 0$ ならば、その時の乗数 A を1.0として終了する(ステップ33-19)。ステップ33-13で、 $x1$ の絶対値が大きくなければステップ33-17に移る。

[0072] ステップ33-8及びステップ33-18のいずれかで、 $z=0$ ならば、ステップ33-20に移る。ステップ33-20では、信号 $x(i)$ の仮数部の最下位ビットの8ビットがすべて0であり、かつ $y1$ の最下位ビットが0でないかを調べる(ステップ33-20)。この両条件を満たせば、 $y1$ の最下位ビットが0でないサンプル(信号 $x(i)$)の数がしきい値(例えば10)を超えたかを調べる(ステップ33-21)。しきい値を超えていれば、ステップ33-19に移り、超えていなければステップ33-22に移る。ステップ33-22では信号 $x(i)$ の仮数部の最下位ビットがすべて0であり、かつ $y1$ の最下位ビットが0でないかを調べる(ステップ33-22)。

[0073] ステップ33-22の両条件を満たせば、 $y1$ の最下位ビットが0でないサンプル(信号 $x(i)$)の数がしきい値(例えば100)を超えたかを調べる(ステップ33-23)。しきい値を超えていれば、ステップ33-19に移る。

ステップ33-22でいずれかの条件を満たしていなければ、 $y(i)$ を $y1$ としてステップ33-1に移る(ステップ33-24)。ステップ33-8で、 z が0であればステップ33-20に移る。ステップ33-1ですべての信号 $x(i)$ についての乗数のチェックを完了すればその当該乗数 A を正しいと決定する(ステップ33-25)。この際 $y(i)$ も求まっている

ことになる。ステップ33-9ないしステップ33-16は、 y に対する変形処理である。ステップ33-9~33-12、ステップ33-13~33-16は、各々 $y(i)$ のダイナミックレンジがなるべく小さくなるような処理である。

[0074] y の変形の基本処理を図34に示す。図33と異なるところのみを説明する。ステップ33-8で、 z が0であれば、直ちにステップ33-24に移る。同様にステップ33-18で、 z が0であれば、直ちにステップ33-24に移る。ステップ33-18で、 z が0でなければ、 z が所定ビット数(例えば23ビット)以下で表現できるかを調べる(ステップ34-1)。表現できれば、ステップ33-24に直ちに移り、表現できなければステップ34-2に移る。ステップ34-2では、 $y(i)$ を0としステップ33-1に移る(ステップ34-2)。ステップ33-1ですべての信号 $x(i)$ について処理を完了すれば、この $y(i)$ を変形する処理は終了する。

[0075] 次に浮動小数点を整数に変換する処理手順の例を、図35を参照して説明する。実施例15(図28)の整数化部281や、実施例17(図30)の整数化部303aにこの処理は適用できる。

乗数 A 、信号 $x(i)$ 、除算処理結果 $y(i)$ に対し、図34で変形処理したもの(これも $y(i)$)と記載する)を入力する(ステップ35-1)。 $y(i)$ の絶対値の最大値からシフト係数 S_c を求める(ステップ35-2)。符号化装置が想定する量子化ビット数を BMA とし、スケール係数 S_F を次式により求める(ステップ35-3)。

$$[0076] \quad S_F = 2^{(BMA-1+S_c)}$$

すべての $x(i)$ について処理が完了したかを調べる(ステップ35-4)。完了していなければ、 $y(i)$ が0であるかを調べる(ステップ35-5)。0であれば、 $y'(i)=0$ としてステップ35-4に戻る(ステップ35-6)。ステップ35-5で、 $y(i)$ が0でなければ、 $y(i)$ が正であるかを調べる(ステップ35-7)。 $y(i)$ が正であれば、 $y(i)$ が正の場合の整数化処理を行って $y'(i)$ を求め、ステップ35-4に戻る(ステップ35-8)。ステップ35-7で $y(i)$ が正でなければ、 $y(i)$ が負の場合の整数化処理を行って $y'(i)$ を求め、ステップ35-4に戻る(ステップ35-9)。ステップ35-4ですべての $x(i)$ について処理が完了すれば、 $y'(i)$ を求める処理は終了する。

[0077] 図35のステップ35-8の $y(i)$ が正の場合の整数化処理の手順例を、図36を参照

して説明する。 $y(i)$ にスケール S_F をかけた値に対して小数点以下を切り捨てて $y'(i)$ を求める(ステップ36-1)。なお

[数2]

[A]

はAの小数点以下の切捨てを表わす。 $y'(i)$ をスケール S_F で割算し、その結果を y_1 とする(ステップ36-2)。 y_1 に乗数Aを掛算してその結果を1とする(ステップ36-3)。信号 $x(i)$ から x_1 を引算し、 z を求める(ステップ36-4)。 z が0でないかを調べる(ステップ36-5)。 z が0でなければ、 x_1 が $x(i)$ より小さいかを調べる(ステップ36-6)。 x_1 の方が小さければ、 $y'(i)$ に1を加算して $y'(i)$ とする(ステップ36-7)。その $y'(i)$ をスケール S_F で割算して y_1 とする(ステップ36-8)。その y_1 に乗数Aを掛算し x_1 としてステップ36-6に戻る(ステップ36-9)。

[0078] ステップ36-6で x_1 の方が小さくなければ、 x_1 が $x(i)$ より大きいかを調べる(ステップ36-10)。 x_1 の方が大きければ、 $y'(i)$ から1を減算しその結果を $y'(i)$ とする(ステップ36-11)。その $y'(i)$ をスケール S_F で割算し、その結果を y_1 とする(ステップ36-12)。その y_1 に乗数Aを掛算し、その結果を x_1 とする(ステップ36-13)。その x_1 を信号 $x(i)$ から引算し、その結果を z とする(ステップ36-14)。その z が0でないかを調べる(ステップ36-15)。 z が0でなければ z が所定ビット数(例えば23ビット)以下で表現可能かを調べる(ステップ36-16)。表現可能でなければ、 $y'(i)$ を0としてステップ36-18に移る(ステップ36-17)。ステップ36-5で z が0であれば、その時の $y'(i)$ が求める $y'(i)$ である。ステップ36-10で x_1 の方が大きくなければ、ステップ36-18に移る。ステップ36-15で z が0であれば、ステップ36-18に移る。ステップ36-16で z を所定ビットN以下で表現できれば、ステップ36-18に移る。

[0079] なおステップ36-16で、 z がNビット(この例では23ビット)以下で表現できない場合は $y'(i)$ を0とするかわりに乗数Aを1.0としてもよい。この場合は $y'(i)$ は $x(i)$ から実施例16又は実施例17で行っている手法によって再度求め直す。

次に $y(i)$ が負の場合の $y(i)$ の変形手順の例を、図37を参照して説明する。 $y(i)$ にスケール S_F をかけた値の小数点以上を切り上げて整数 $y'(i)$ とする(ステップ37-1)

。ここで

[数3]

[A]

はAの少数点以下を切り上げることを表わす。 $y'(i)$ をスケール S_F で割算し、 $y1$ とする(ステップ37-2)。その $y1$ に乗数Aを乗算して $x1$ とする(ステップ37-3)。信号 $x(i)$ から $x1$ を引算し、結果を z とする(ステップ37-4)。 z が0でないかを調べる(ステップ37-5)。 z が0でなければ $x1$ が $x(i)$ よりも大きいかを調べる(ステップ37-6)。 $x1$ の方が大きければ、 $y'(i)$ から1を減算し、その結果を $y'(i)$ とする(ステップ37-7)。その $y'(i)$ をスケール S_F で割算し、その結果を $y1$ とする(ステップ37-8)。その $y1$ に乗数Aを掛算し、その結果を $x1$ として、ステップ37-6に戻る(ステップ37-9)。

[0080] ステップ37-6で $x1$ の方が大きくなければ、 $x1$ が $x(i)$ より小さいかを調べる(ステップ37-10)。 $x1$ の方が小さければ、 $y'(i)$ に1を加算して、 $y'(i)$ とする(ステップ37-11)。その $y'(i)$ をスケール S_F で割算して、割算結果 $y1$ を求める(ステップ37-12)。 $y1$ に乗数Aを掛算し、結果を $x1$ とする(ステップ37-13)。 $x1$ を $x(i)$ から引算して、結果を z とする(ステップ37-14)。 z が0でないかを調べる(ステップ37-15)。 z が0でなければ、その z が所定値(この例では23)ビット以下で表現できるかを調べる(ステップ37-16)。表現できなければ、 $y'(i)$ を0としてステップ37-18に移る(ステップ37-17)。ステップ37-5で z が0であれば、ステップ37-18に移る。同様にステップ37-15で z が0であれば、ステップ37-18に移る。それぞれその時得られている $y'(i)$ を、求める $y'(i)$ とする。ステップ37-10で $x1$ の方が小さくなければステップ37-18に移る。ステップ37-16で z がNビット以下で表現できればステップ37-18に移る。このようにして $y'(i)$ を変形する処理が終了する。

[0081] なおステップ37-16で z がNビット以下で表現できない場合は、 $y'(i)$ を0とする代わりに $A=1.0$ としてもよい。この場合は、 $y'(i)$ は実施例15又は実施例17で説明した手法によって $x(i)$ から再度求め直す。

入力信号 $x(i)$ が浮動小数点形式ではなく整数形式表現の場合に共通の乗数を求めるには、その入力信号 $x(i)$ の極性をまず判断する。正であれば $y(i)$ に対する変形

処理を図36に示した手法で行い、負であれば $x(i)$ が $y(i)$ に対する変形処理は図37に示した手法で行えばよい。

[実施形態9]

以下に前述の乗数推定部110における乗数Aの決定について説明する。

実施例20(有理近似)

有理近似による乗数Aを推定する方法を、図38を参照して説明する。仮数部変換部381は、入力された浮動小数点数 $x(i)$ の各仮数部の最上位ビット(MSB)の1ビット上位に、1を配して符号なし24ビット整数 $mx(i)$ ($i=0, 1, \dots, N$)とする。ただし $x(i)$ が、NaNや非正規化数の場合は $mx(i)=0$ に変換する。この場合、全入力信号を入力することなく、例えば $x(i)$ を絶対値表現で小さい順に配列し、1フレーム(例えば2048サンプル)中の0以外の複数個(例えば512個)をソート部380で選出して仮数部変換部381に入力しても良い。

[0082] GCD計算部382では、必要に応じて、通常用いられているユークリッドの互除法を用いて最大公約数gcdを計算する。gcdが判定部382aの基準値よりも大きければ、その値を乗数として出力しても良い。しかし、この例ではそのgcdをA0とし、分母選定部383が、例外値($mx(i)=0$)以外の $mx(i)$ をA0で割算し、この結果 $my(i)$ を出力する。そして、以下のアルゴリズムを使い、乗数Aの推定値を計算する。その際にA0分の調整を加える場合もある。

このような乗数Aの推定方法はいくつか考えられる。

第一の方法(最初の方法. 基本)

前述したように例外値を $mx(i)$ とした $mx(i)$ を求めることは同様である。

(1)分母選定部383は、 $mx(i)$ 中の最大値を選択し、その値をXとする。

(2)例外値($mx(i)=0$)以外の $mx(i)$ を一つずつ取り、下記式(1)を有理近似計算部384で計算する。

$$\begin{aligned} (mx(i) - 1/2) / (X + 1/2) < my(i) / Y(i) \\ < (mx(i) + 1/2) / (X - 1/2) \quad (2) \end{aligned}$$

この式(2)を満たす分母最小の既約分数 $my(i) / Y(i)$ を求める。

[0083] ここで連分数展開を利用する。

有理近似計算部384における処理手順を以下に示す。

ステップ1. $(mx(i)-1/2)/(X+1/2)=nlow/dlow, (mx(i)+1/2)/(X-1/2)=nhigh/dhigh$ と既約分数で書く。

ステップ2. $nlow/dlow < 1 < nhigh/dhigh$ ならば1/1を出力する。

ステップ3. $dn=\max\{dlow, dhigh\}$ とし、 dn に対応する分子を nm とする。

ステップ4. $n0=d1=0, n1=d0=1, i=0$ とおく。

[0084] ステップ5. $k=[nm/dn]$ (nm/dn を超えない最大の整数), $r=(nm$ を dn で割ったあまり), $nm=dn, dn=r$ とおく。

ステップ6. $s=n0+n1 \times k, t=d0+d1 \times k, n0=n1, d0=d1$ とおく。

ステップ7. $nm(i)=n1=s, dn(i)=d1=t$ とおく。 $dn > 0$ ならば i を1増やしステップ5へ戻り、 $dn=0$ ならばステップ8へ移る。

ステップ8. $nlow/dlow < nm(i)/dn(i) < nhigh/dhigh$ となる最小の i を求める ($dn=dlow$ の場合は奇数のみ、 $dn=dhigh$ の場合は2以上の偶数のみ調べれば十分)。

[0085] ステップ9. $i=1$ のときは、 $(1+k \times nm(0))/k < nhigh/dhigh$ を満たす最小の整数 k を求め、 $(1+k \times nm(0))/k$ を出力する。

i が1より大きい奇数のときは、 $(nm(i-2)+k \times nm(i-1))/(dn(i-2)+k \times dn(i-1)) < nhigh/dhigh$ を満たす最小の整数 k を求め、 $(nm(i-2)+k \times nm(i-1))/(dn(i-2)+k \times dn(i-1))$ を返す。

i が偶数のときは、 $(nm(i-2)+k \times nm(i-1))/(dn(i-2)+k \times dn(i-1)) > nlow/dlow$ を満たす最小の整数 k を求め、 $(nm(i-2)+k \times nm(i-1))/(dn(i-2)+k \times dn(i-1))$ を返す。

[0086] 以上の処理が可能なのは高木貞治著「初等整数論理講義」第2版、2003年9月15日共立出版株式会社発行、124～145頁参照(以下非特許文献3という)から次のように理解される。この非特許文献3の140頁の定理2. 7から以下が成立つ。

実数 ω に対して誤差範囲 $\varepsilon > 0$ を指定したとき、

$$\omega - \varepsilon < a/b < \omega$$

$$(\text{または } \omega < a/b < \omega + \varepsilon)$$

となる分母最小の分数 a/b は、 ω の奇数番目(または偶数番目)の主あるいは中間

近似分数である。これは、定理2.7のすぐ下にある証明(全体で4行)の、「また」から始まる後半3行を言い換えたものである。

[0087] 上記主張の ω が前記有理近似計算部384の nm/dn ,
 ε が $nm/dn - nlow/dlow$ (または $nhigh/dhigh - nm/dn$)である。

定理2.7のある第2章では ω を無理数としているが、140頁中程の[注意]にある通り、場合分けなどに注意すれば、結論は有理数の場合にも成立する。今の場合、有理近似計算部384のステップ3の、

「 $dn = \max\{dlow, dhigh\}$ とし、 dn に対応する分子を nm とする。」

がその注意にあたる(分母が大きい方を ω として連分数展開)。

(3) 共通分母確認部385で例外値以外の $mx(i)$ の数だけ得られた Yi の最大値 Y を求め、

(4) この Y の約数となる Yi の数 N が、あらかじめ決めてある条件(下記に示す)を満たせば、共通因数計算部386で乗数 A の推定値として、 Y をシフトして $1 < X/Y < 2$ となるようにしたときの X/Y を、浮動小数点表現で計算したものを出力する。

[0088] 条件を満たさなければ乗数 A として1を出力する。

前記あらかじめ決めてある条件の具体例は、例外値($mx(i) = 0$)を除いたデータ数を、 m とすると下記の通りである。

$N \geq C$ (あらかじめ決めた数)

$N \geq cm$ ($0 < c < 1$ はあらかじめ決めておく)

入力信号によっては桁上りが生じる確率が知られており、1フレーム中で例えば20%程度が桁上りとなる場合は、 N は70%の値とする。また乗数 A をなるべく利用したほうが符号量を小さくできる点から N は20%程度でも良い。また A の精度をよくしたい場合は80%程度とする。なお A の正当性は後で調べる。

変形

前記(4)で、条件を満たすとき、 Y の約数となる Yi について、 mxi/Yi (Yi は条件を満たすようにシフトした上で)の平均を A として出力する。この平均により A の誤差が少なくなることが期待できる。条件を満たさないときは上記と同じ様に $A = 1.0$ を出力する。

[0089] 前記(4)で、 Y_i の最小公倍数 Y を求め、それが X 以下なら、 X/Y (Y をシフトした上で)を浮動小数点表現で計算した値を出力する。つまり Y_i は、必ず最大公約数となるか不明であるが、 Y_i の最小公倍数を求めれば、必ず全体に対する最大公約数になる。あるいは、 Y の約数となる Y_i から計算したものの平均を用いる。条件を満たさないときは上記と同様に $A=1.0$ を出力する。

このようにしてこれまでよりも広い範囲から Y_i を選択できる。

前記(1)で、最大値 X の代わりに、大きい方から M 個(あらかじめ決めた数、2以上、好ましくは、3~50内の数)を取り、それぞれについて A を推定し、多数決で勝ったものつまり m_{xi} 中に最も多く共通に使用できるものを出力する。

[0090] この場合は、一個だけで求めると、たまたま素性の悪いデータ(例えば桁上りしたものとしないものが混在した場合に対する X)を回避する。

前記(1)でランダムに M 個(あらかじめ決めた数)を取り、以下同様に各 A を推定し、多数決で決める。

この理由は大きい方から M 個とる理由と同様である。

前記(1)で最大値の代わりに、大きい方から順に取り、それぞれについて A を推定する。

[0091] 1以外の推定値 A が M 個(あらかじめ定められた数)集まったところで、それを出力する。 M 個集まる前にデータがなくなれば、足りない分は $A=1$ として出力する。これら複数候補でいずれがよいかは検算ルーチンにまかせた方がよい。例えば、前記実施例3(図11、図12)で説明した変形処理、あるいは実施例4(図13、図14)で説明した変形処理を各乗数 A ごとに処理して、最も好ましい A を選択する。

図39に示すように、図38に示した処理を次のように変更してもよい。分母選定部391は、GCD計算部382で得られた $\text{gcd}A_0$ で各 $m_x(i)$ を割算する。これは、分母選定部391で、GCD計算部382で得られた gcd を後の処理で再び求めないため、つまり無駄な処理を少なくするためである。有理近似計算部392は、 $m_x(i)/A_0$ を、有理近似計算部384での $m_x(i)$ と同様に扱った処理を行う。また $m_x(i)/A_0$ として処理をしているため、共通因数計算部386は、得られた乗数候補を A_0 倍して乗数 A とする。

[0092] 図40に示すように、図38に示した処理を次のように変更してもよい。共通分母確認部402は、 Y_i が見つからないと判定部402aで判定すると、分母選定部401で他の1個の分母 X を選定させる。この選定は、 $m_x(i)$ の大きい順、あるいはランダムに行う。分母 X の選定をあらかじめ決めた回数、例えば5回行っても Y_i が見つからない場合は $A=1.0$ を出力する。

図39中に破線394で示すように、共通分母確認部385で Y_i が求まらない場合は、図40で説明したと同様に、分母選定部391で他の分母 X を選択するようにしてもよい。分母選定部391では X を M 個ではなく、1個ずつ選定する。

[0093] 図41の変形例では、有理近似計算部411は、各 $m_x(i)$ と対応した $m_y(i)/Y_i$ を求める。ここで、乗数 A は、除算処理結果 $y(i)$ がなるべく小さくなるものがよい。従って共通分母確認部412では、分母 Y の最も小さい1個を選択し、共通分母となり得るかを確認する。共通分母となり得ない場合は、有理近似計算部411が、次に大きい Y と $y(i)$ の組を選択する。共通分母確認部412は、この組に対して共通分母となり得るかを確認する。このようなことを繰り返し、最も好ましい $y(i)$ 、 $Y(i)$ の組を決定する。図41中に破線で示すように、最も大きい共通分母が得られなければ、最初に入力された Y_i 、 y_i をそれぞれ1単位ずつ増加させて、共通分母を求めるようにしてもよい。

第二の方法(いくつか足してグループ作成)

この方法は、 x_0, \dots, x_n をあらかじめ定めた数ずつ足して、相対的に誤差を減らす方法である。

[0094] 最初に仮数部を符号なしの24ビット整数形式 $m_x(i)$ に変換し、例外値は $m_x(i) = 0$ とする。そして、

(1) $m_x(i)$ を大きい順に配列し、大きい方から M 個ずつ(M はあらかじめ決めておく)加えたものを X_i とする(仮数部変換部381で)。

(2) X_i の最大値を取り X とする(分母選定部383で)。

(3) X_i を一つずつ取り、

$$\begin{aligned} (X_i - 1 / (2\sqrt{M})) / (X + 1 / (2\sqrt{M})) &< y_i / Y_i \\ &< (X_i + 1 / (2\sqrt{M})) / (X - 1 / (2\sqrt{M})) \end{aligned}$$

を満たす分母最小の既約分数 y_i/Y_i を求める(有理近似計算部384で)。

[0095] 以下は第一の方法の(4)以下の処理と、その変形処理を行う。

このようにすると、第一の方法よりも相対的にデータの誤差が減るので、正しいYを求める確率が上がる。

第三の方法(全探索その1. yベース)

この方法も、 $mx(i)$ を求める処理は同じである。

(1) $mx(i)$ の最大値を取り、Xとする。

(2) $mx(i) < X$ となるi最小の例外値($mx(i) = 0$)ではない x_i を用い、

$(x_i - 1/2) / (X + 1/2) < y_i / Y_i < (x_i + 1/2) / (X - 1/2)$ を満たす分母最小の既約分数 y_i / Y を求める。

(3) 例外値($mx(i) = 0$)以外のすべての x_j に対し、

$$(x_i - 1/2) / (X + 1/2) < y_j / Y < (x_i + 1/2) / (X - 1/2)$$

を満たす分母がYとなる分数 y_j / Y が、存在するかどうか調べる。

(4) すべての場合に存在すれば、Aの推定値として X / Y を出力する。あるいは、 x_j / y_j の平均を出力する。

[0096] 存在しないjがあれば、

$$(x_i - 1/2) / (X + 1/2) < y_i / Y_i < (x_i + 1/2) / (X - 1/2)$$

を満たす分母であって、Yのつぎに小さい分数を求め、その分母を改めてYとする。

もし、 $Y < X$ なら(3)へ戻り、もし、 $Y = X$ なら1を出力する。

第四の方法(全探索その2. Aベース)

この方法では、Aについての全探索を行う。

まず、前述の有理近似を用いた共通乗数を求める考え方を、図38を参照して説明する。前述したように、GCD計算部382では、入力信号 $x(i)$ に誤差が含まれていると正しい値を求めることができない。分母選定部383で、1フレーム中の $mx(i) = 0$ 以外の適当な代表値をX(分母)として選ぶ。この場合、最大公約数という点からなるべく情報量が多い信号(前記では最大値)が好ましい。

[0097] 有理近似計算部384では、代表値Xとそのフレーム中の全ての $mx(i)$ (前述したように、 $mx(i) = 0$ を処理から除くことは、以後の説明では省略する。)を分子とする分数 $mx(i) / X$ において、 $mx(i)$ 中に含まれている誤差を考慮した時に、 $mx(i) / X$ が

、正しい値に対して負の誤差か正の誤差を含むが、その最大誤差範囲内にある y_i/Y を求める。前述では四捨五入による誤差としたため、 $1/2$ を±したが、誤差が± α であるとすれば、次式を満たす分母最小の既約分数 y_i/Y_i を求めることになる。

$$[0098] \quad (mx(i) - \alpha) / (X + \alpha) < y_i / Y_i < (mx(i) + \alpha) / (X - \alpha)$$

例えば $mx(i)$ に±2の誤差があるとすれば $\alpha = 2$ とする。正負の誤差が非対称の場合は+ α 、- β を用いればよい。

この既約分数 y_i/Y_i は、先に示した非特許文献3中の連分数展開を利用した定理から必ず求まる。

このようにして求めた y_i/Y_i は、フレーム中の $mx(i)$ の数だけある。従って、共通分母確認部385で Y_i 中の各 $mx(i)$ に共通な値を求める。この場合、同一の Y_i の数があるから決められた数以上であれば、その Y_i を共通の Y とし、共通因数計算部386で、その Y_i で X を割算して、共通乗数 A として出力する。

実施例21(自己相関利用)

乗数 A を求める他の例を、図42を参照して説明する。

[0099] 各信号 $x(i)$ の仮数部の最上位ビット(MSB)の1ビット上位に、“1”を配して24ビットしたものを $mx(i)$ とする。これと $mx(i)$ を j ビット($j=1, 2, \dots, 23$)下位にシフトしたものとの自己相関を求める。図42Aは、 $mx(i)$ と、これを1ビットシフトしたものの $mx(i)$ (1)とこれらの自己相関結果の例を示す。図42Aは、 $z(i)$ が12ビット構成を例としている。

この自己相関値中の1のビット数を計数する。このことを $j=1, \dots, 23$ について行う。 j ごとに全ての $z(i)$ についての相関値中の1のビット数を合計する。この合計値がしきい値を超えた j についての合計値を1とし、しきい値以下の場合を0とする。この判定結果のビット列が求める乗数 A である。

[0100] 例として、 $z(i)$ のビット数が8の場合の信号 $x(0) \sim x(4)$ について、 $j=1, 2, \dots, 9$ での各自己相関のビットが“1”の数を、図42Bに示す。図42Bで、 $j=1$ とした場合の自己相関値におけるビットが“1”の数は、 $x(0)$ は5、 $x(1)$ は3、 $x(2)$ は2…である。 $j=2$ の場合は、 $x(0) \sim x(4)$ の自己相関値のビットが“1”の数は、全て0である。 $j=3$ の場合は、 $x(0), x(1), x(2) \dots$ に対する自己相関値のビットが“1”の数は、2, 4, 1

…である。これら自己相関値のビットが“1”の数を各 $j=1, 2, 3, \dots$ ごとに合計した値が、23, 0, 15, …である。しきい値を10とした場合、しきい値を超えたか否かの判定結果が、 $j=1, 2, 3, \dots$ に対し1, 0, 1, …となる。この判定結果が、 $x(0) \sim x(4)$ に対する共通乗数 A とされる。なお j が大きくなるに従って、相関値中で1となるビット数が減少するから、しきい値を j が大きくなる程、共通乗数 A を小さくしてもよい。

実施例22

音響信号としては、例えば24ビット整数形式で信号化された信号素材を変形、振幅調整、効果付加、混合などの編集処理したものを浮動小数点形式に変換されたものが入力信号 $x(i)$ として入力されることがある。このような場合、元の24ビット整数形式信号に対して行った加減乗除処理が記録されて、その加減乗除処理を知ることができる場合がある。この場合、その入力信号 $x(i)$ と共に加減乗除の演算順序と係数を符号化装置に与えられる。乗数推定部では、その入力された加減乗除情報をもとに乗数 A を決定する。例えば定数 A を元信号に乗算した場合は乗数 A としてその入力された定数 A を出力すればよい。

- [0101] 上述した符号化装置、復号化装置、有理近似共通乗数計算装置は、それぞれコンピュータにより機能させることもできる。この場合はコンピュータに、目的とする装置(各種実施例で図に示した機能構成をもつ装置)として機能させるためのプログラム、又はその処理手順(各実施例で示したもの)の各過程をコンピュータに実行させるためのプログラムを、CD-ROM、磁気ディスク、半導体記録装置などの記録媒体から、あるいは通信回線を介してそのコンピュータ内にダウンロードし、そのプログラムを実行させればよい。

請求の範囲

- [1] 所定区間ごとに複数の入力信号をまとめた列を、共通の乗数とその乗数で各入力信号を除算処理した商信号の列とその残りの誤差信号の列とに分解する分解ステップと、
- 前記商信号の列を圧縮符号化して商符号を求める商符号化ステップと、
- 前記商符号と前記乗数と前記誤差信号の列とを出力する出力ステップと
- を有する情報符号化方法。
- [2] 請求項1記載の情報符号化方法であって、
- 前記分解ステップは、整数形式の前記商信号の列と浮動小数点形式の前記誤差信号の列とに分解するステップであり、
- 前記誤差信号の列を圧縮符号化して誤差符号を求める誤差符号化ステップも有し、
- 、
- 前記出力ステップが、前記商符号と前記乗数と前記誤差符号とを出力するステップである
- ことを特徴とする情報符号化方法。
- [3] 請求項1記載の情報符号化方法であって、
- 前記分解ステップは、浮動小数点形式の前記商信号の列と浮動小数点形式の前記誤差信号の列とに分解するステップであり、
- 前記浮動小数点形式の各商信号を、整数形式の商信号と浮動小数点形式の差分信号とに分離し、整数形式の商信号の列と浮動小数点形式の差分信号の列とを求める商信号分離ステップも有し、
- 前記商符号化ステップは、前記整数形式の商信号の列を圧縮符号化するステップであり、
- 前記差分信号の列を圧縮符号化して差分符号を求める差分符号化ステップも有し、
- 、
- 前記誤差信号の列を圧縮符号化して誤差符号を求める誤差符号化ステップも有し、
- 、
- 前記出力ステップが、前記商符号と前記差分符号と前記乗数と前記誤差符号とを

出力するステップである

ことを特徴とする情報符号化方法。

- [4] 所定の区間ごとに複数の浮動小数点形式の入力信号をまとめた列を符号化する情報符号化方法であって、

共通乗数推定部で、前記入力信号列を構成する入力信号に共通する乗数(共通乗数)を推定する共通乗数推定ステップと、

除算処理部で、当該乗数で前記入力信号列を構成する各入力信号を除算処理した商信号の列を求める除算処理ステップと、

変形処理部で、前記商信号の列を構成する各商信号に丸め処理を行った整数形式の列(整数形式商信号列)を求める変形処理ステップと、

商信号符号化部で、前記整数形式商信号列を符号化する商符号化ステップと、

乗算部で、前記整数形式商信号列の各信号に前記共通乗数を乗算した浮動小数点形式の信号の列(浮動小数点形式信号列)を求める乗算ステップと、

誤差符号化部で、前記浮動小数点形式信号列の各信号と入力信号列の各信号との差である誤差信号の列を符号化する誤差符号化ステップと

を有する情報符号化方法。

- [5] 所定の区間ごとに複数の浮動小数点形式の入力信号をまとめた列(入力信号列)を符号化する情報符号化方法であって、

共通乗数推定部で、前記入力信号列を構成する入力信号に共通する乗数(共通乗数)を推定する共通乗数推定ステップと、

除算処理部で、当該乗数で前記入力信号列を構成する各入力信号を除算処理した商信号の列を求める除算処理ステップと、

整数化部で、前記商信号の列を構成する各商信号を整数形式に変換した商信号の列(整数形式商信号列)に変換する整数化ステップと、

商信号符号化部で、前記整数形式商信号列を符号化する商符号化ステップと、

浮動小数点化部で、前記整数形式商信号列を構成する各商信号に基づく浮動小数点形式の商信号の列(浮動小数点形式商信号列)を求める浮動小数点化ステップと、

乗算部で、前記浮動小数点形式商信号列の各信号に前記共通乗数を乗算した浮動小数点形式の信号の列(浮動小数点形式信号列)を求める乗算ステップと、
誤差符号化部で、前記浮動小数点形式信号列の各信号と入力信号列の各信号との差である誤差信号の列を符号化する誤差符号化ステップと
を有する情報符号化方法。

- [6] 請求項4記載の情報符号化方法であって、
前記共通乗数推定ステップが、
前記入力信号列を構成する入力信号の値から代表値を選定し、
その代表値を分母とし、前記入力信号列を構成する各入力信号の値を分子とし、これら分数の中の、前記各数値の誤差に基づく正側誤差と負側誤差間に存在する分母最小の既約分数を有理近似計算により求め、
これら求めた既約分数中の共通分母を求め、
その共通分母と前記代表値とから前記入力信号列を構成する各入力信号の値に対する共通の乗数を求める
ことを特徴とする情報符号化方法。

- [7] 請求項4または6記載の情報符号化方法であって、
前記変形処理ステップが、
前記各商信号に丸め処理を行った後の各商信号に、
あらかじめ定めた範囲で
最下位ビットに1ずつの加算または減算を行い、
入力信号から、前記の加算または減算された商信号と前記共通乗数の積を減算した結果が0となる商信号を探索する探索サブステップと
前記探索サブステップの結果、全ての商信号に対して前記探索が成功した場合には、当該商信号の集合を前記整数形式商信号列とし、
前記探索の一部が失敗した場合には、共通乗数を1として前記入力信号そのものを商信号とする商信号変更サブステップと
を有する
こと特徴とする情報符号化方法。

- [8] 入力符号から商符号と乗数と誤差信号列とを分離する分離ステップと、
前記商符号を復号して商信号列を生成する商信号列復号ステップと、
前記商信号列を構成する各商信号に前記乗数を乗算する乗算ステップと、
前記乗算ステップで生成した信号列の各信号を、前記誤差信号列を構成する各誤差信号で補正して、復号化信号の列を得る補正ステップと
を有する情報復号化方法。
- [9] 請求項8記載の方法において、
前記分離ステップは、入力符号から商符号と乗数と誤差符号を分離するステップであり、
前記商信号列復号ステップは、整数形式の商信号列を生成するステップであり、
前記誤差符号を復号して浮動小数点形式の誤差信号列を生成する誤差信号列復号ステップも有し、
前記乗算ステップは、前記整数形式の商信号列の各商信号に前記乗数を乗算して浮動小数点形式の信号列を生成するステップである、
ことを特徴とする情報復号化方法。
- [10] 前記請求項8の方法において、
前記分離ステップは、入力符号から商符号と差分符号と乗数と誤差符号を分離するステップであり、
前記商信号列復号ステップは、整数形式の商信号列を生成するステップであり、
前記差分符号を復号して浮動小数点形式の差分信号列を生成する差分信号列復号ステップも有し、
前記整数形式の商信号列の各商信号と前記差分信号列の対応する差分信号とを加算して浮動小数点形式の商信号の列を求める加算ステップも有し、
前記誤差符号を復号して浮動小数点形式の誤差信号列を生成する誤差信号列復号ステップも有し、
前記乗算ステップは、前記浮動小数点形式の商信号の列の各商信号に前記乗数を乗算して浮動小数点形式の信号列を生成するステップである、
ことを特徴とする情報復号化方法。

- [11] 商復号化部で、商符号を整数形式の商信号列(整数形式商信号列)に復号する商復号化ステップと、
誤差復号化部で、誤差符号に基づいて誤差信号列を得る誤差復号化ステップと、
乗算部で、前記整数形式商信号列を構成する各信号に共通乗数を乗算して、浮動小数点形式の信号列を得る乗算ステップと、
前記乗算ステップで生成した信号列の各信号に、前記誤差信号列を構成する各信号を加算する加算ステップと
を有する情報復号化方法。
- [12] 商復号化部で、商符号を整数形式の商信号列(整数形式商信号列)に復号する商復号化ステップと、
誤差復号化部で、誤差符号に基づいて誤差信号列を得る誤差復号化ステップと、
浮動小数点化部で、前記整数形式商信号列を構成する各信号に基づく浮動小数点形式の商信号の列(浮動小数点形式商信号列)を生成する浮動小数点化ステップと、
乗算部で、前記浮動小数点形式商信号列を構成する各信号に共通乗数を乗算して、浮動小数点形式の信号列を得る乗算ステップと、
前記乗算ステップで生成した信号列の各信号に、前記誤差信号列を構成する各信号を加算する加算ステップと
を有する情報復号化方法。
- [13] 請求項11または12記載の情報復号化方法であって、
前記誤差復号化ステップが、
共通乗数が1の場合には、各誤差信号の仮数部として、前記整数形式商信号列の対応する商信号の桁数を仮数部の総桁数から減算した数分の下位側のビットに、誤差符号を復号して得られた信号を設定し、残りの上位側のビットに“0”を設定して、浮動小数点表現の誤差信号列を求め、
共通乗数が1以外の場合には、誤差符号から求めた各信号を、浮動小数点表現の各誤差信号の仮数部として、浮動小数点表現の誤差信号列を求める
処理を含む

ことを特徴とする情報復号化方法。

- [14] 所定区間ごとに複数の入力信号をまとめた入力信号列を生成する分割部と、
前記入力信号列が入力され、その入力信号列の各入力信号に共通の乗数と、その乗数で各入力信号を除算処理した商信号の列とその残りの誤差信号の列とに分解する剰余分離処理部と、
前記商信号の列を圧縮符号化して商符号を求める商符号化部と、
前記商符号と前記乗数と前記誤差信号の列とを出力する出力部と
を具備する情報符号化装置。
- [15] 請求項14に記載の装置において、
前記剰余分離処理部は、整数形式の前記商信号の列と浮動小数点形式の前記誤差信号の列とに分解する構成部であり、
前記誤差信号の列を圧縮符号化して誤差符号を求める誤差符号化部も有し、
前記出力部は、前記商符号と前記乗数と前記誤差符号とを出力する構成部であることを特徴とする情報符号化装置。
- [16] 請求項14に記載の装置において、
前記剰余分離処理部は、浮動小数点形式の前記商信号の列と浮動小数点形式の前記誤差信号の列とに分解する構成部であり、
前記浮動小数点形式の各商信号を、整数形式の商信号と浮動小数点形式の差分信号とに分離し、整数形式の商信号の列と浮動小数点形式の差分信号の列とを求める整数化部も有し、
前記商符号化部は、前記整数形式の商信号の列を圧縮符号化する構成部であり、
、
前記差分信号の列を圧縮符号化して差分符号を求める差分符号化部も有し、
前記誤差信号の列を圧縮符号化して誤差符号を求める誤差符号化部も有し、
前記出力部が、前記商符号と前記差分符号と前記乗数と前記誤差符号とを出力する構成部である
ことを特徴とする情報符号化装置。
- [17] 所定の区間ごとに複数の浮動小数点形式の入力信号をまとめた列を符号化する情

報符号化装置であって、

前記入力信号列を構成する入力信号に共通する乗数(共通乗数)を推定する共通乗数推定部と、

当該乗数で前記入力信号列を構成する各入力信号を除算処理した商信号の列を求める除算処理部と、

前記商信号の列を構成する各商信号に丸め処理を行った整数形式の列(整数形式商信号列)を求める変形処理部と、

前記整数形式商信号列を符号化する商符号化部と、

前記整数形式商信号列の各信号に前記共通乗数を乗算した浮動小数点形式の信号の列(浮動小数点形式信号列)を求める乗算部と、

前記浮動小数点形式信号列の各信号と入力信号列の各信号との差である誤差信号の列を符号化する誤差符号化部と

を備える情報符号化装置。

[18] 所定の区間ごとに複数の浮動小数点形式の入力信号をまとめた列(入力信号列)を符号化する情報符号化装置であって、

前記入力信号列を構成する入力信号に共通する乗数(共通乗数)を推定する共通乗数推定部と、

当該乗数で前記入力信号列を構成する各入力信号を除算処理した商信号の列を求める除算処理部と、

前記商信号の列を構成する各商信号を整数形式に変換した商信号の列(整数形式商信号列)に変換する整数化部と、

前記整数形式商信号列を符号化する商符号化部と、

前記整数形式商信号列を構成する各商信号に基づく浮動小数点形式の商信号の列(浮動小数点形式商信号列)を求める浮動小数点化部と、

前記浮動小数点形式商信号列の各信号に前記共通乗数を乗算した浮動小数点形式の信号の列(浮動小数点形式信号列)を求める乗算部と、

前記浮動小数点形式信号列の各信号と入力信号列の各信号との差である誤差信号の列を符号化する誤差符号化部と

を備える情報符号化装置。

- [19] 請求項17記載の情報符号化装置であって、
前記共通乗数推定部が、
前記入力信号列を構成する入力信号の値から代表値を選定し、
その代表値を分母とし、前記入力信号列を構成する各入力信号の値を分子とし、これら分数の中の、前記各数値の誤差に基づく正側誤差と負側誤差間に存在する分母最小の既約分数を有理近似計算により求め、
これら求めた既約分数中の共通分母を求め、
その共通分母と前記代表値とから前記入力信号列を構成する各入力信号の値に対する共通の乗数を求める
ことを特徴とする情報符号化装置。

- [20] 請求項17または19記載の情報符号化装置であって、
前記変形処理部が、
前記各商信号に丸め処理を行った後の各商信号に、
あらかじめ定めた範囲で
最下位ビットに1ずつの加算または減算を行い、
入力信号から、前記の加算または減算された商信号と前記共通乗数の積を減算した結果が0となる商信号を探索する探索手段と
前記探索サブステップの結果、全ての商信号に対して前記探索が成功した場合には、当該商信号の集合を前記整数形式商信号列とし、
前記探索の一部が失敗した場合には、共通乗数を1として前記入力信号そのものを商信号とする商信号変更手段と
を有する
ことを特徴とする情報符号化装置。

- [21] 入力符号から商符号と乗数と誤差信号列とを分離する分離部と、
前記商符号を復号して商信号列を生成する商復号化部と、
前記商信号列を構成する各商信号に前記乗数を乗算する乗算部と、
前記乗算部で生成した信号列の各信号を、前記誤差信号列を構成する各誤差信

号で補正して、復号化信号の列を得る補正処理部と
を備えることを特徴とする情報復号化装置。

- [22] 請求項21記載の装置において、
前記分離部は、入力符号から商符号と乗数と誤差符号を分離する構成部であり、
前記商復号化部は、整数形式の商信号列を生成する構成部であり、
前記誤差符号を復号して浮動小数点形式の誤差信号列を生成する誤差復号化部
も有し、
前記乗算部は、前記整数形式の商信号列の各商信号に前記乗数を乗算して浮動
小数点形式の信号列を生成する構成部である
ことを特徴とする情報復号化装置。

- [23] 前記請求項21の装置において、
前記分離部は、入力符号から商符号と差分符号と乗数と誤差符号を分離する構成
部であり、
前記商復号化部は、整数形式の商信号列を生成する構成部であり、
前記差分符号を復号して浮動小数点形式の差分信号列を生成する差分復号化部
も有し、
前記整数形式の商信号列の各商信号と前記差分信号列の対応する差分信号とを
加算して浮動小数点形式の商信号の列を求める加算部も有し、
前記誤差符号を復号して浮動小数点形式の誤差信号列を生成する誤差復号化部
も有し、
前記乗算部は、前記浮動小数点形式の商信号の列の各商信号に前記乗数を乗算
して浮動小数点形式の信号列を生成する構成部である
ことを特徴とする情報復号化装置。

- [24] 商符号を整数形式の商信号列(整数形式商信号列)に復号する商復号化部と、
誤差符号に基づいて誤差信号列を得る誤差復号化部と、
前記整数形式商信号列を構成する各信号に共通乗数を乗算して、浮動小数点形
式の信号列を得る乗算部と、
前記乗算部で生成した信号列の各信号に、前記誤差信号列を構成する各信号を

- 加算する加算部と
を備える情報復号化装置。
- [25] 商符号を整数形式の商信号列(整数形式商信号列)に復号する商復号化部と、
誤差符号に基づいて誤差信号列を得る誤差復号化部と、
前記整数形式商信号列を構成する各信号に基づく浮動小数点形式の商信号の列
(浮動小数点形式商信号列)を生成する浮動小数点化部と、
前記浮動小数点形式商信号列を構成する各信号に共通乗数を乗算して、浮動小
数点形式の信号列を得る乗算部と、
前記乗算部で生成した信号列の各信号に、前記誤差信号列を構成する各信号を
加算する加算部と
を備える情報復号化装置。
- [26] 請求項24または25記載の情報復号化装置であって、
前記誤差復号化部が、
共通乗数が1の場合には、各誤差信号の仮数部として、前記整数形式商信号列の
対応する商信号の桁数を仮数部の総桁数から減算した数分の下位側のビットに、誤
差符号を復号して得られた信号を設定し、残りの上位側のビットに“0”を設定して、
浮動小数点表現の誤差信号列を求める手段、
共通乗数が1以外の場合には、誤差符号から求めた各信号を、浮動小数点表現の
各誤差信号の仮数部として、浮動小数点表現の誤差信号列を求める手段
を有することを特徴とする情報復号化装置。
- [27] 複数の数値の集団から代表値を選定し、
その代表値を分母とし、前記集団の各数値を分子とし、これら分数の中の、前記各
数値の誤差に基づく正側誤差と負側誤差間に存在する分母最小の既約分数を有理
近似計算により求め、
これら求めた既約分数中の共通分母を求め、
その共通分母と前記代表値とから前記複数の数値に対し共通の乗数を求めること
を特徴とする共通乗数推定方法。
- [28] 複数の数値の集団から代表値を選定する分母選定部と、

前記代表値、前記数値の集団、前記数値の誤差が入力され、前記代表値を分母とし、前記各数値を分子とし、これら分数中の、前記誤差に基づく正側誤差と負側誤差との間に存在する分母最小の既約分数を有理近似計算により求める有理近似計算部と、

前記各既約分数が入力され、共通の分母を求める共通分母確認部と、

前記共通の分母と前記代表値が入力され、共通の分母に対し補正を行って前記数値の集団に対する共通乗数を出力する共通因数計算部と

を具備する共通乗数推定装置。

- [29] 請求項1～13、27のいずれかに記載した方法の各過程をコンピュータに実行させるプログラム。
- [30] 請求項29記載の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。

[図1]

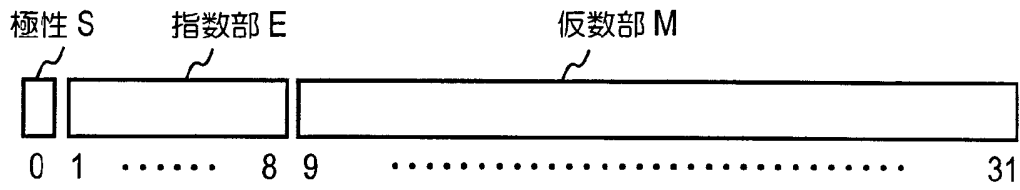


図1

[図2]

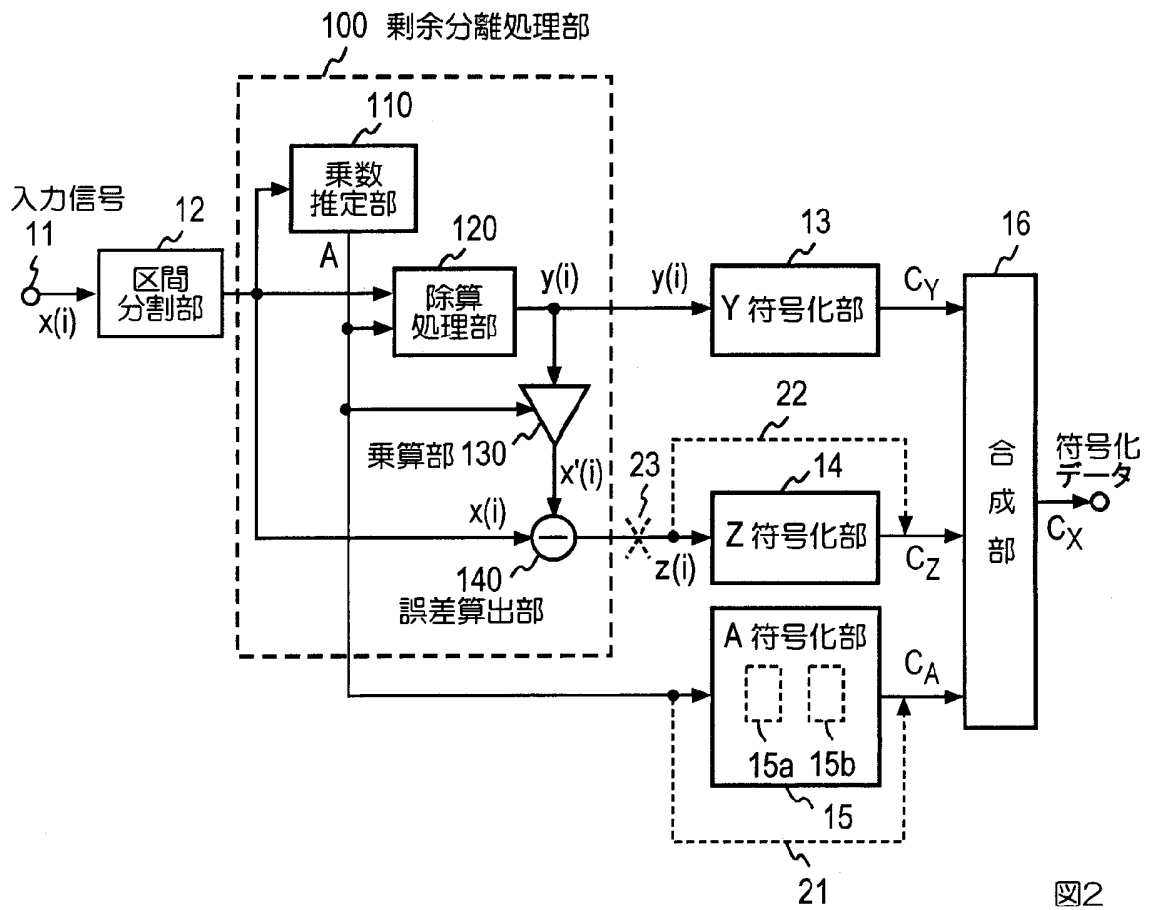


図2

[図3]

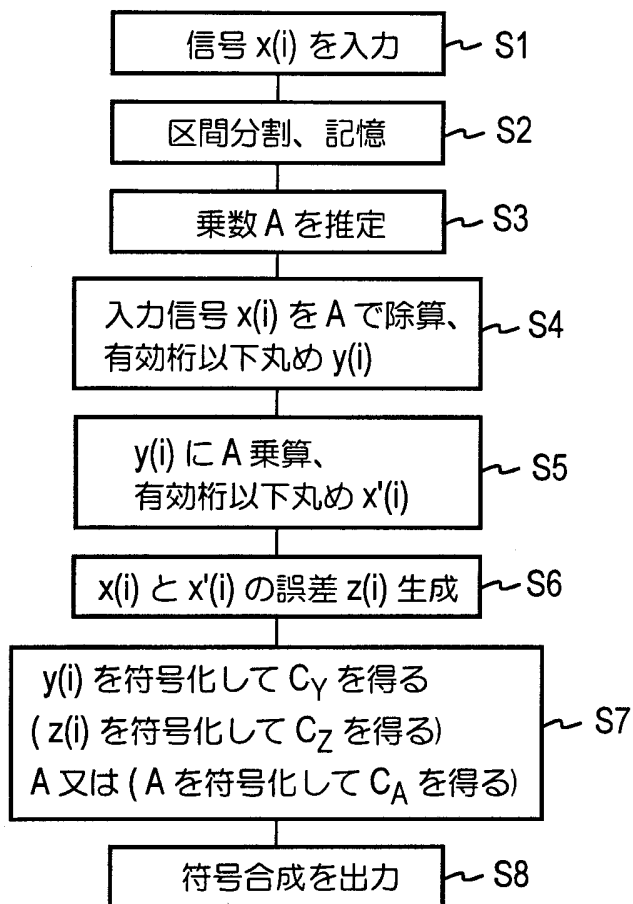


図3

[図4]

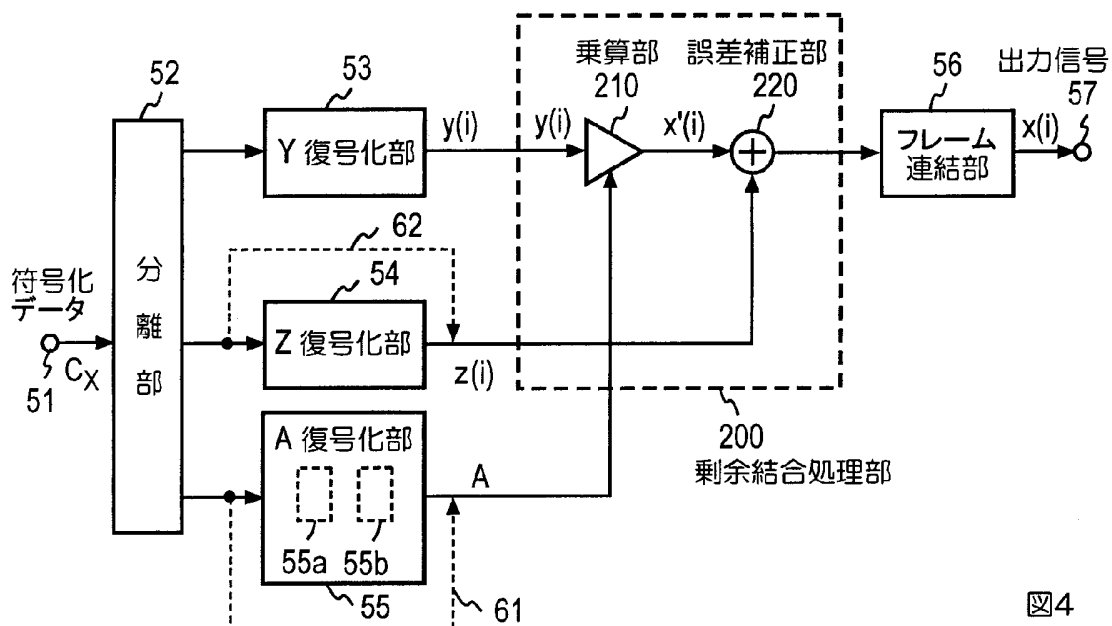


図4

[図5]

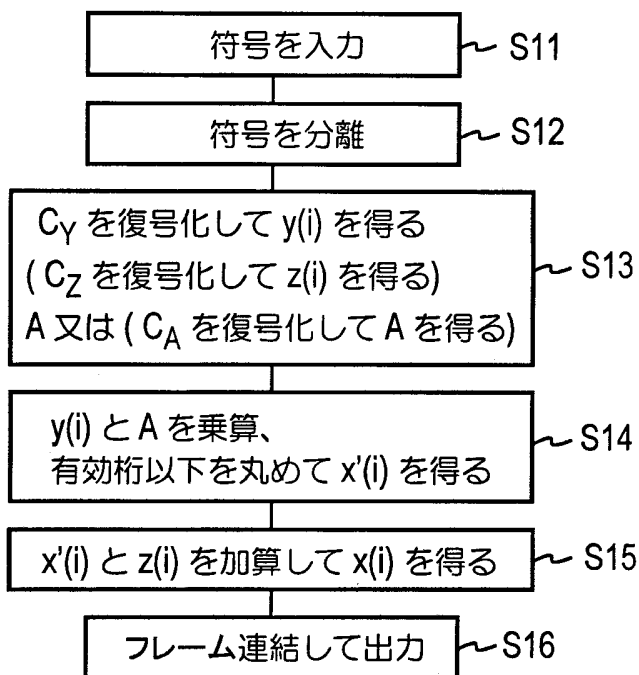


図5

[図6]

	10進数表現	2進数表現
A	$\begin{cases} x(1) = 250 \\ x(2) = 50 \\ x(3) = 350 \end{cases}$	$\begin{cases} 11111010 \\ 110010 \\ 101011110 \end{cases}$
B	$\begin{cases} y(1) = x(1) \div A = 250 \div 1.5625 = 160 \\ y(2) = x(2) \div A = 50 \div 1.5625 = 32 \\ y(3) = x(3) \div A = 350 \div 1.5625 = 224 \end{cases}$	
C	$\begin{cases} y(1) = 160 \\ y(2) = 32 \\ y(3) = 224 \end{cases}$	$\begin{cases} 10100000 \\ 100000 \\ 11100000 \end{cases}$

図6

[図7]

	数值	符号	指数部	仮数部 (23bit)
A	478.4	: 0	: 8	: 110111110011001100110011:
	95.68	: 0	: 6	: 011111110101110000101001:
	669.76	: 0	: 9	: 01001110111000010100100:
B	160 (= 478.4/2.99)	: 0	: 7	: 01000000000000000000000:
	32 (= 95.68/2.99)	: 0	: 5	: 00000000000000000000000:
	224 (= 669.76/2.99)	: 0	: 7	: 11000000000000000000000:

図7

[図8]

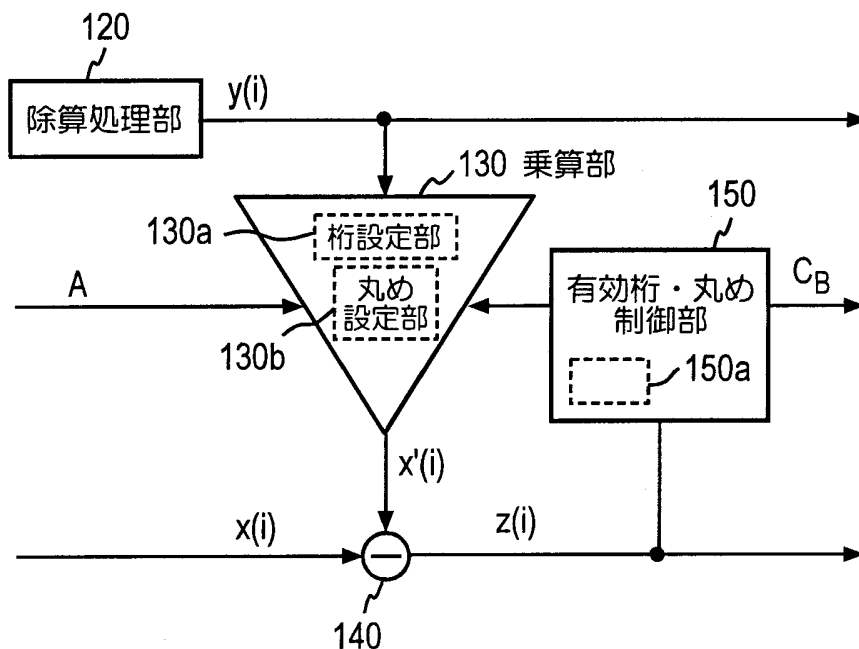


図8

[図9]

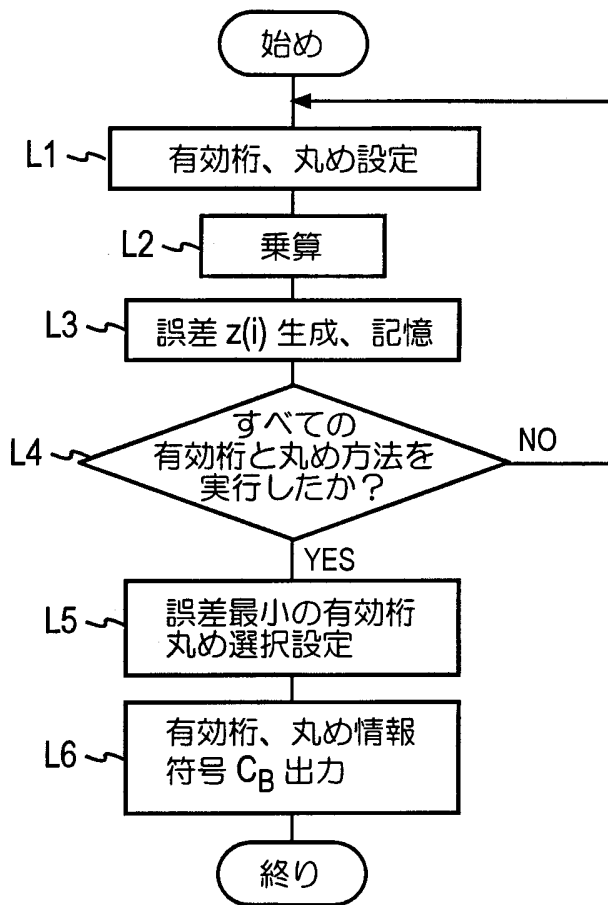


図9

[図10]

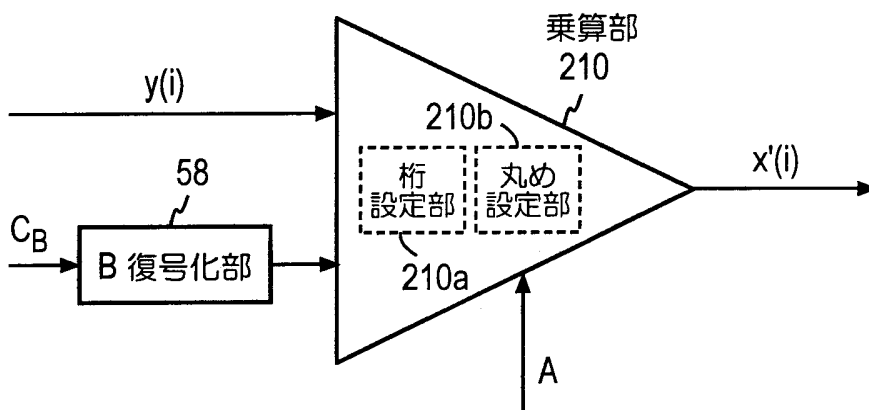


図10

[図11]

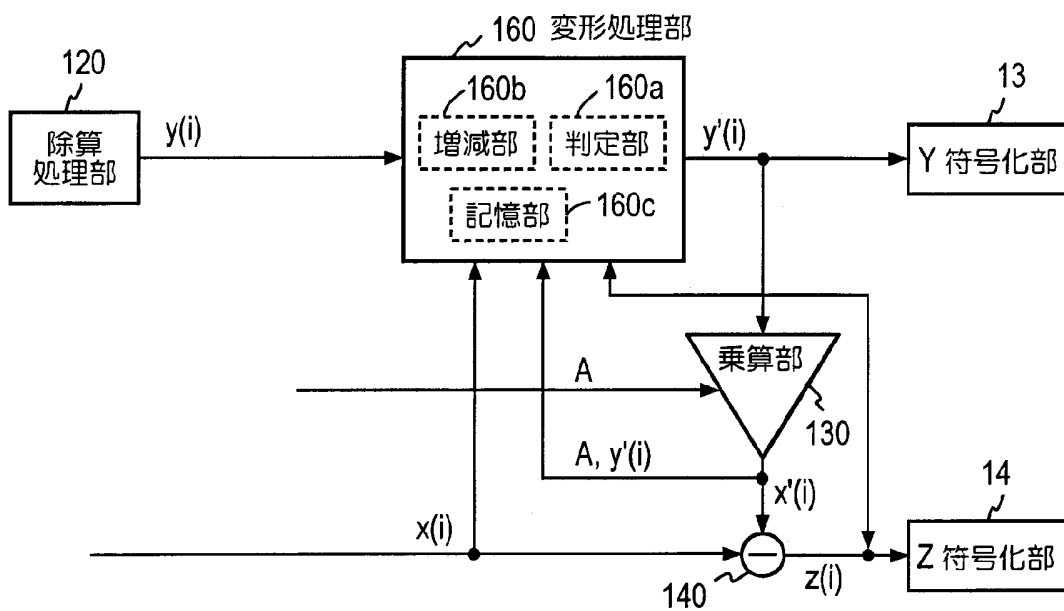


図11

[図12]

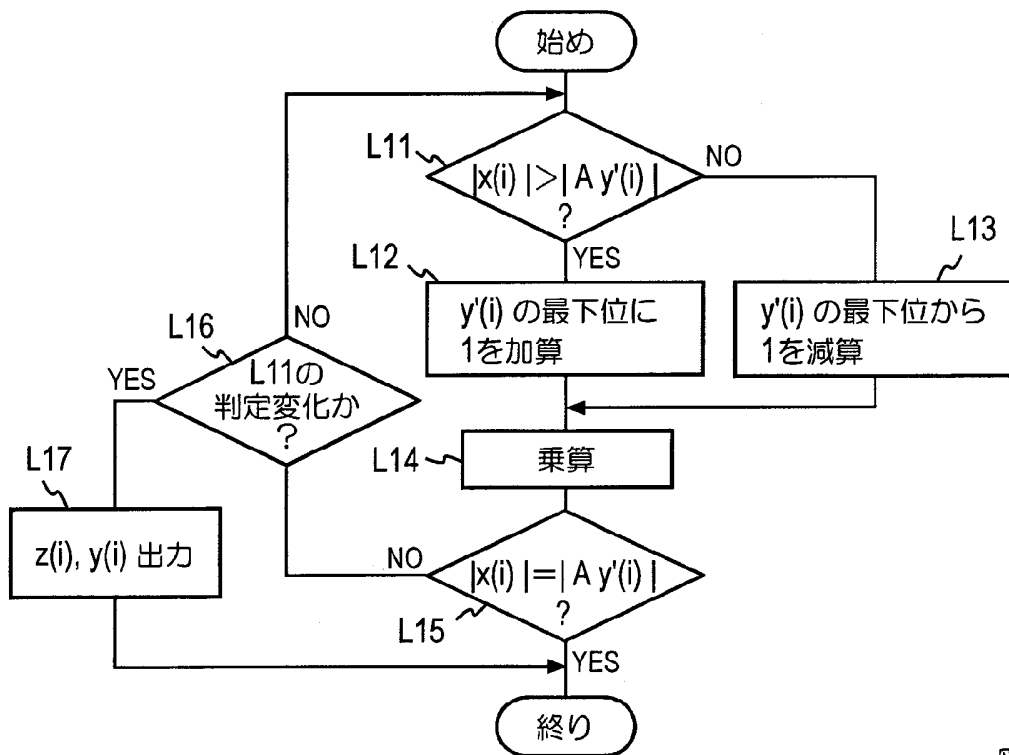


図12

[図13]

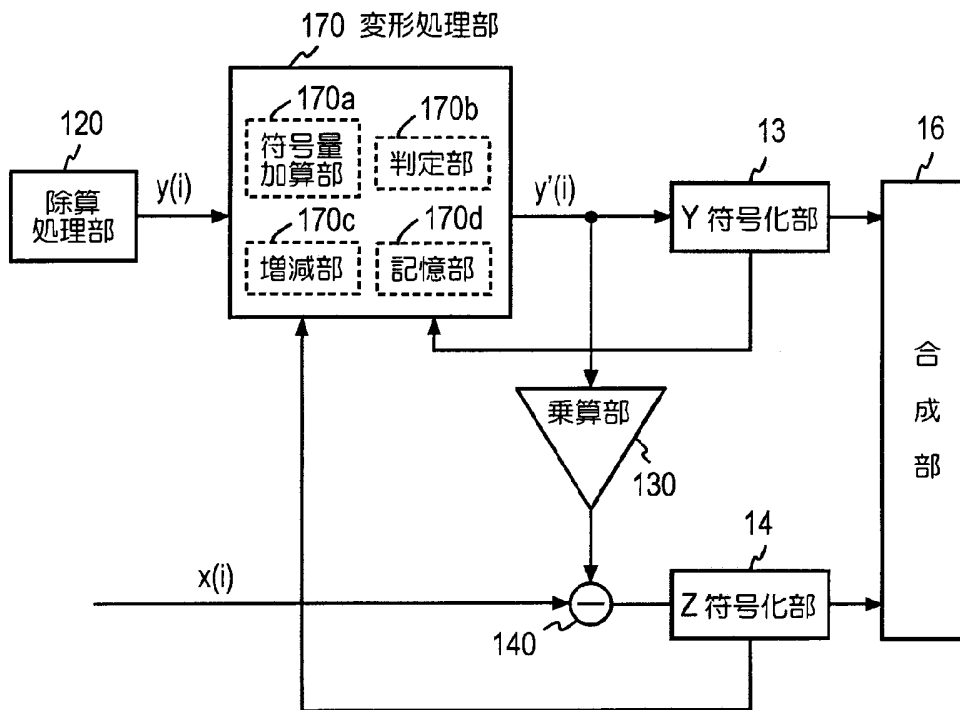


図13

[図14]

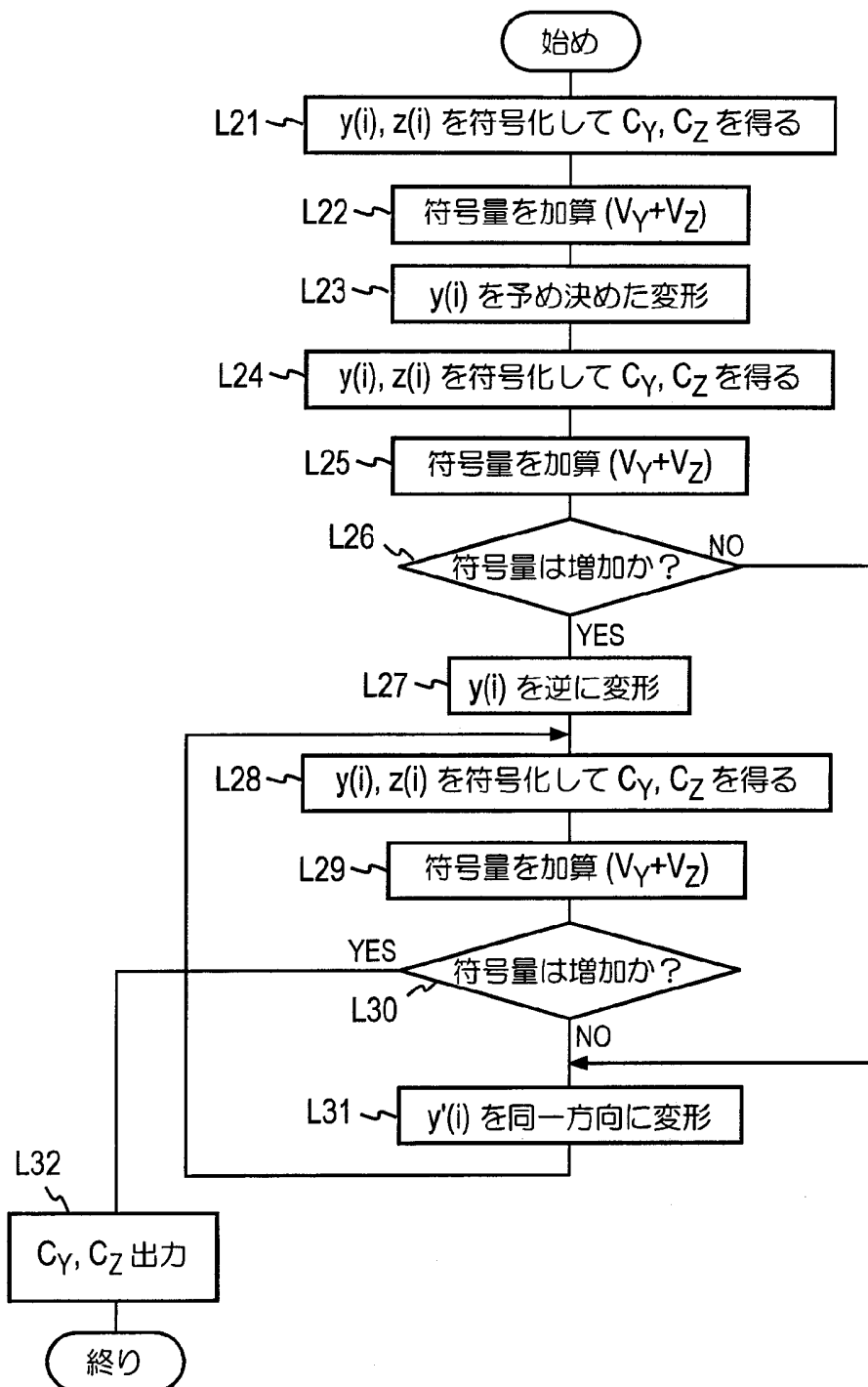


図14

[図15]

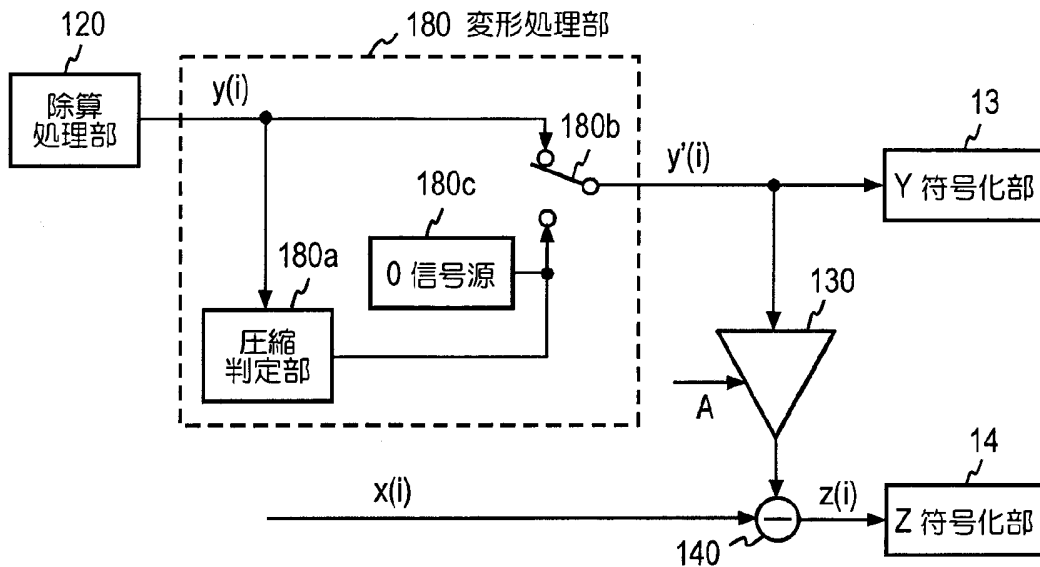


図15

[図16]

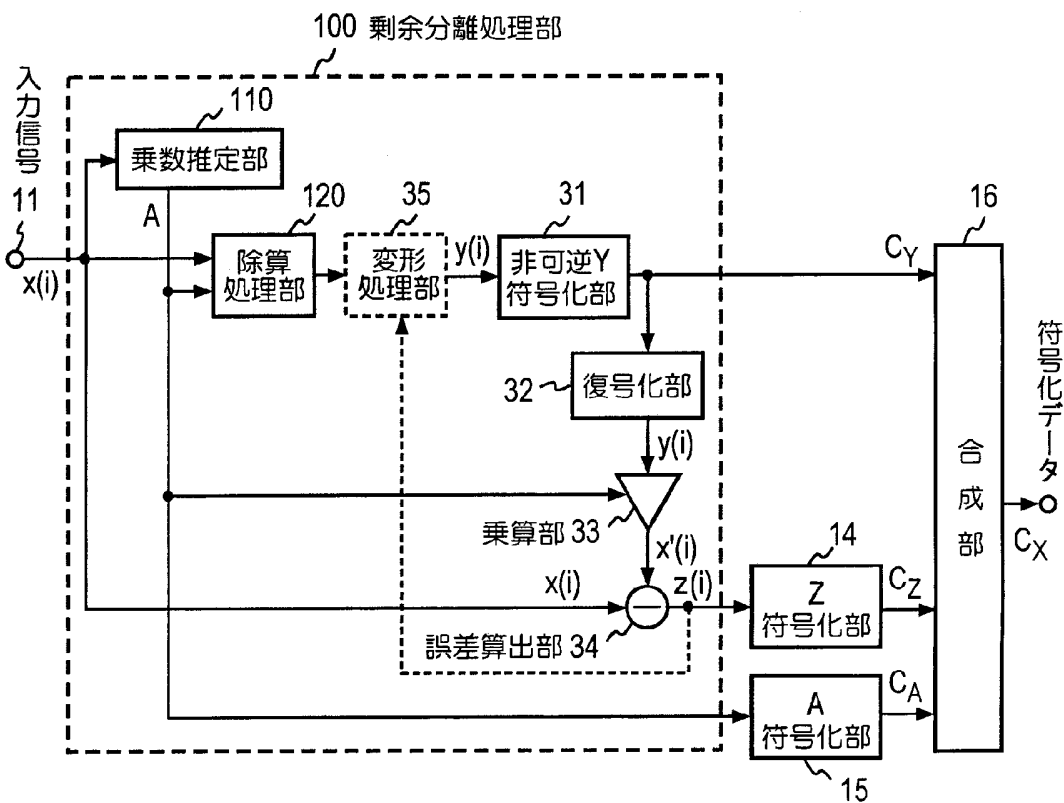


図16

[図17]

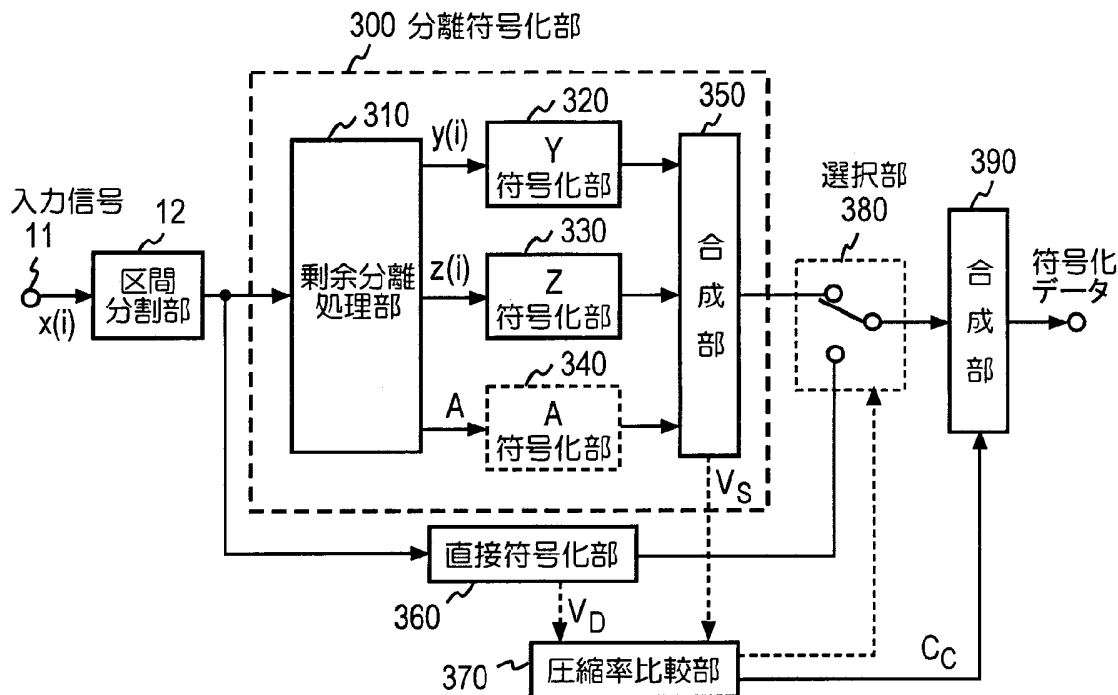


図17

[図18]

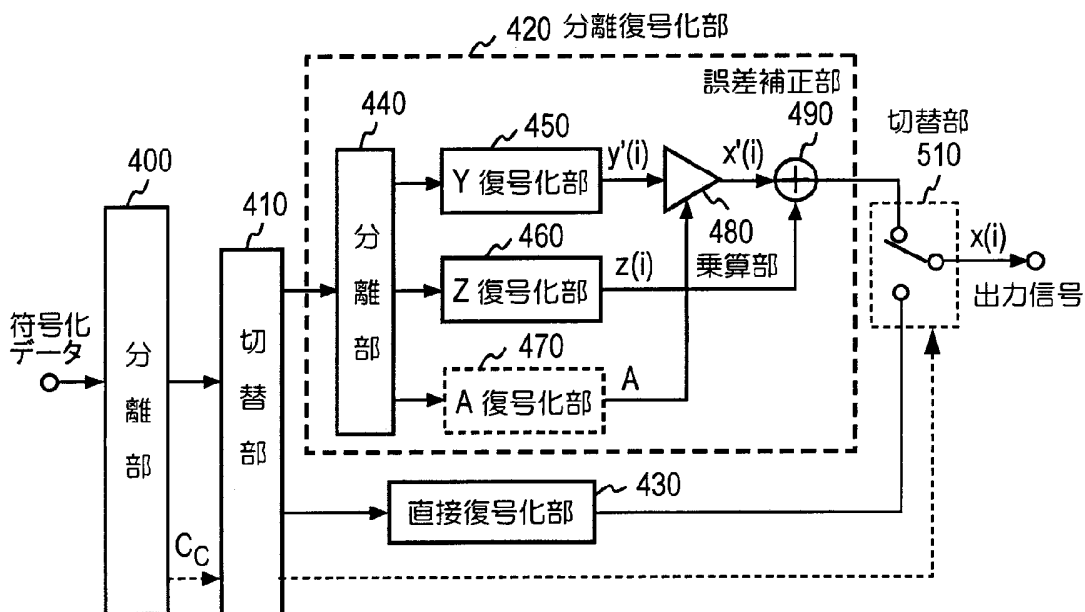


図18

[図19]

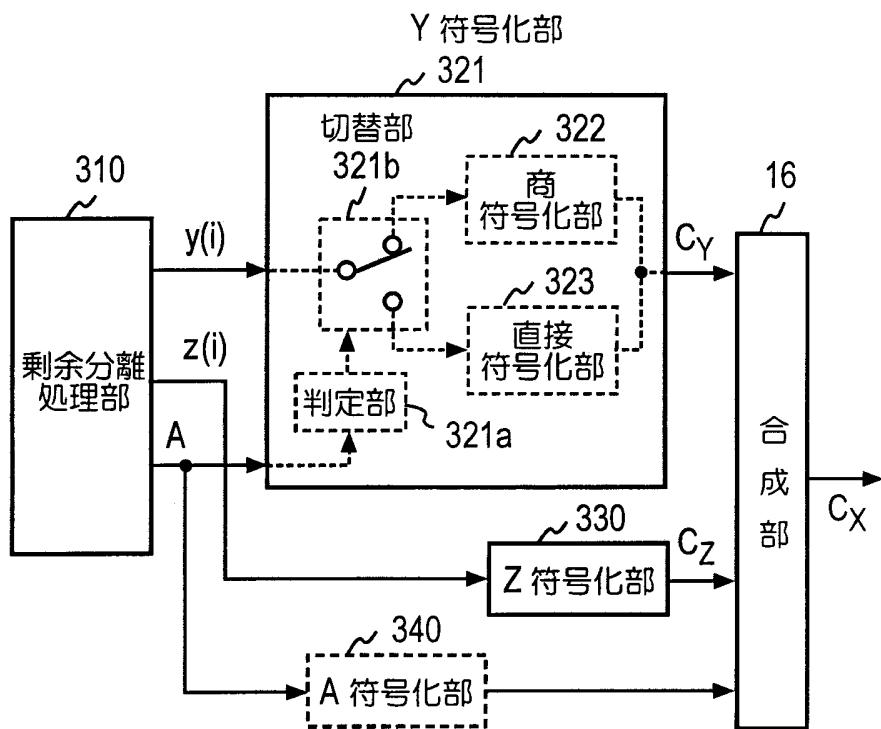


図19

[図20]

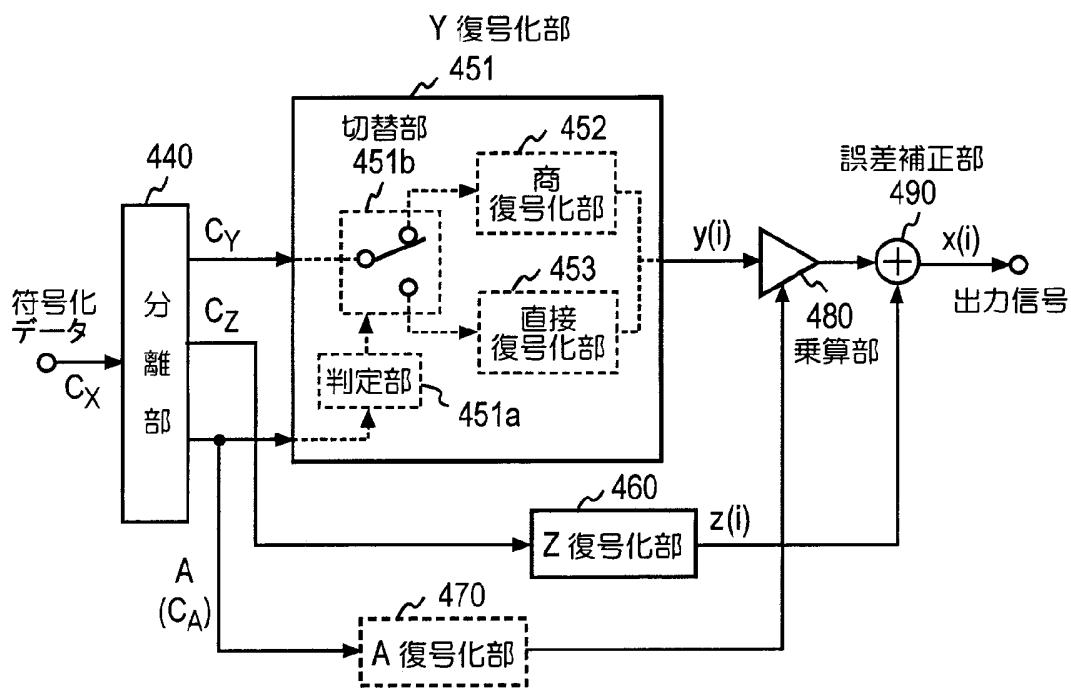


図20

[図21]

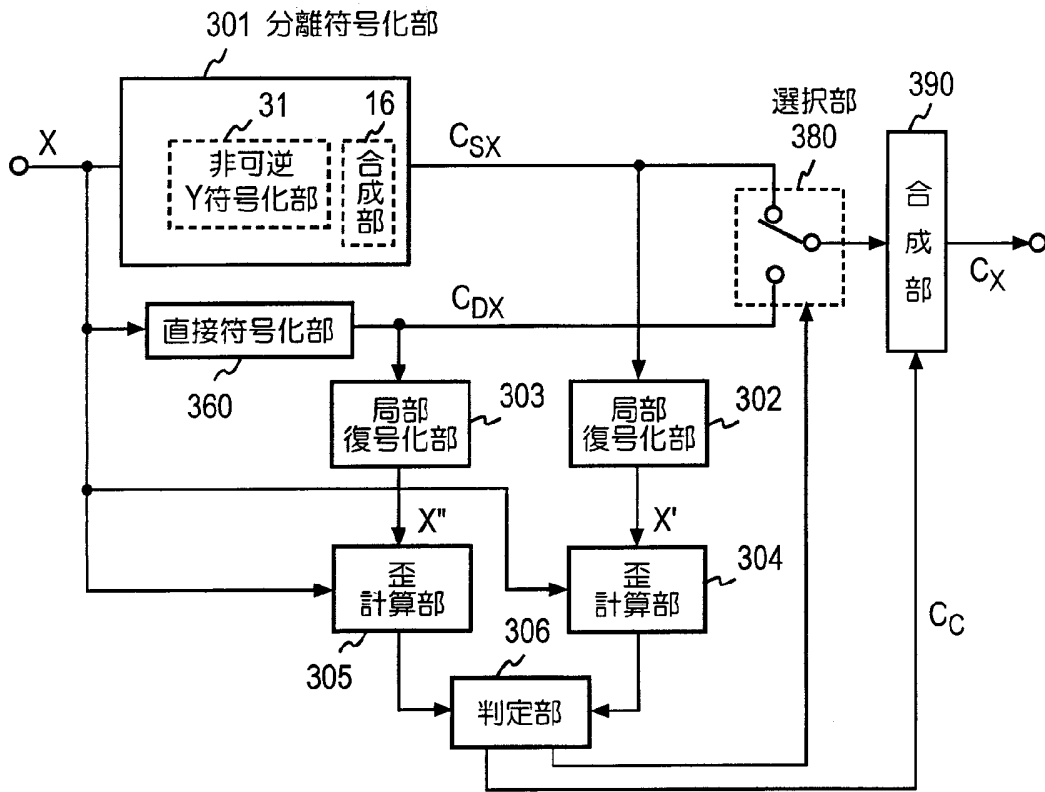


図21

[图22]

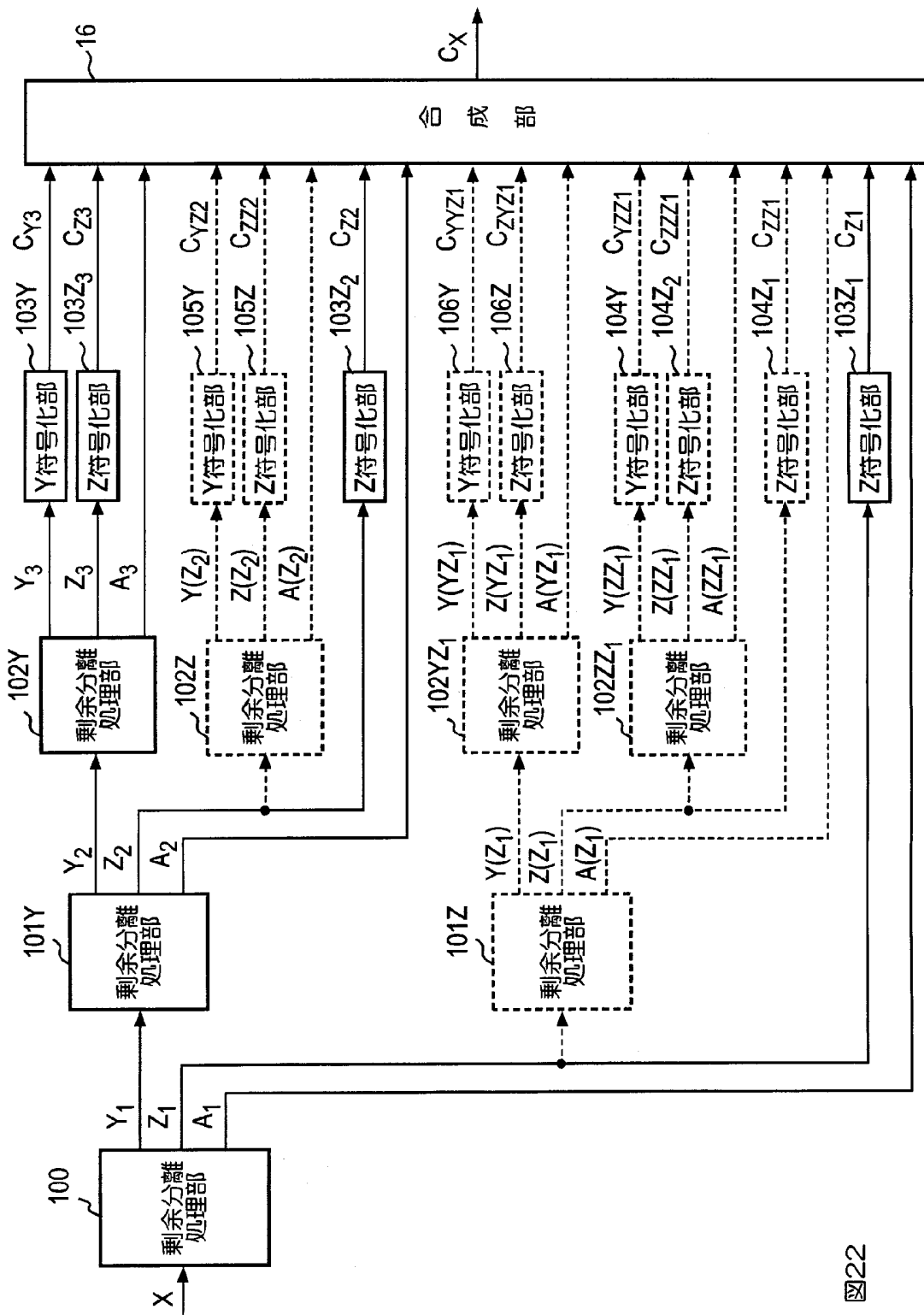


图22

[図23]

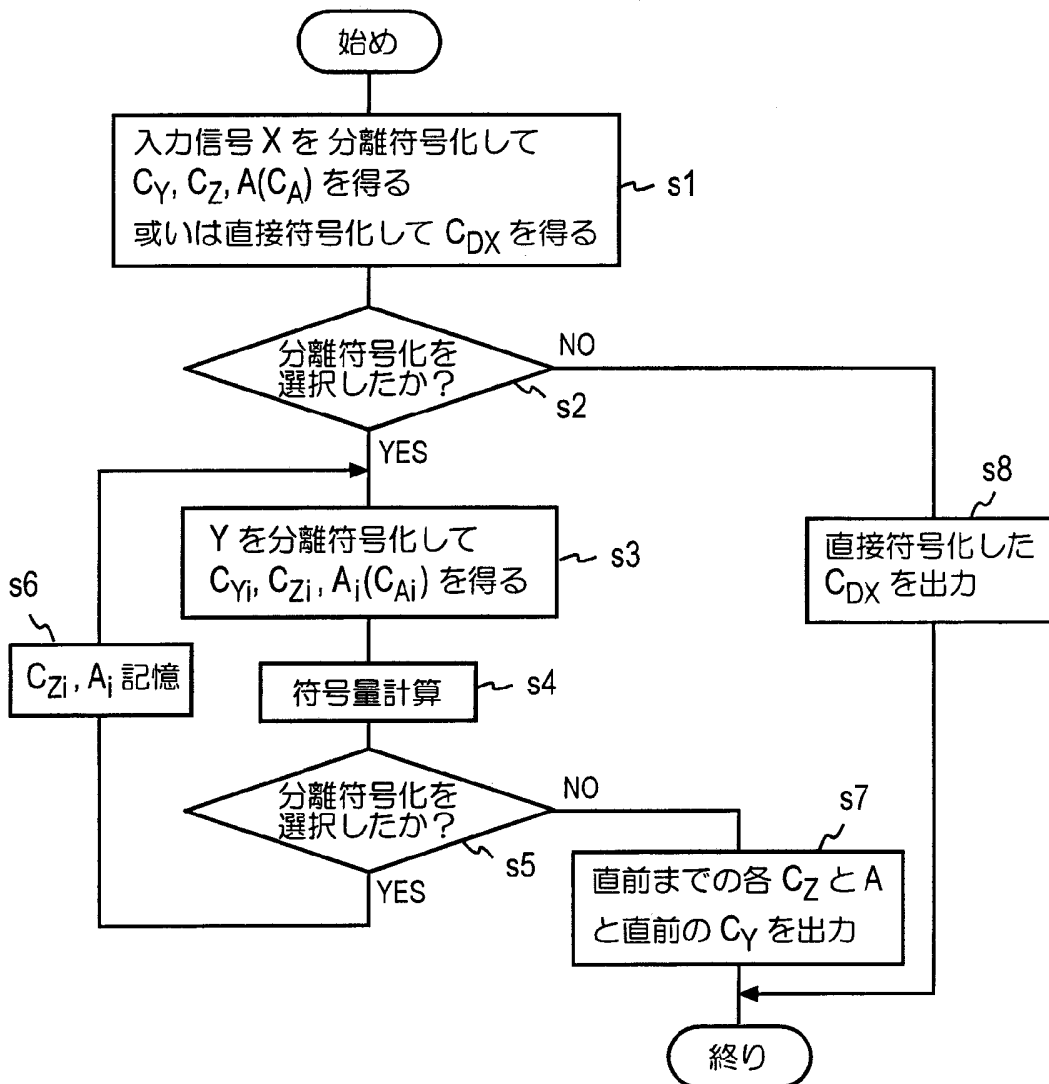


図23

[図24]

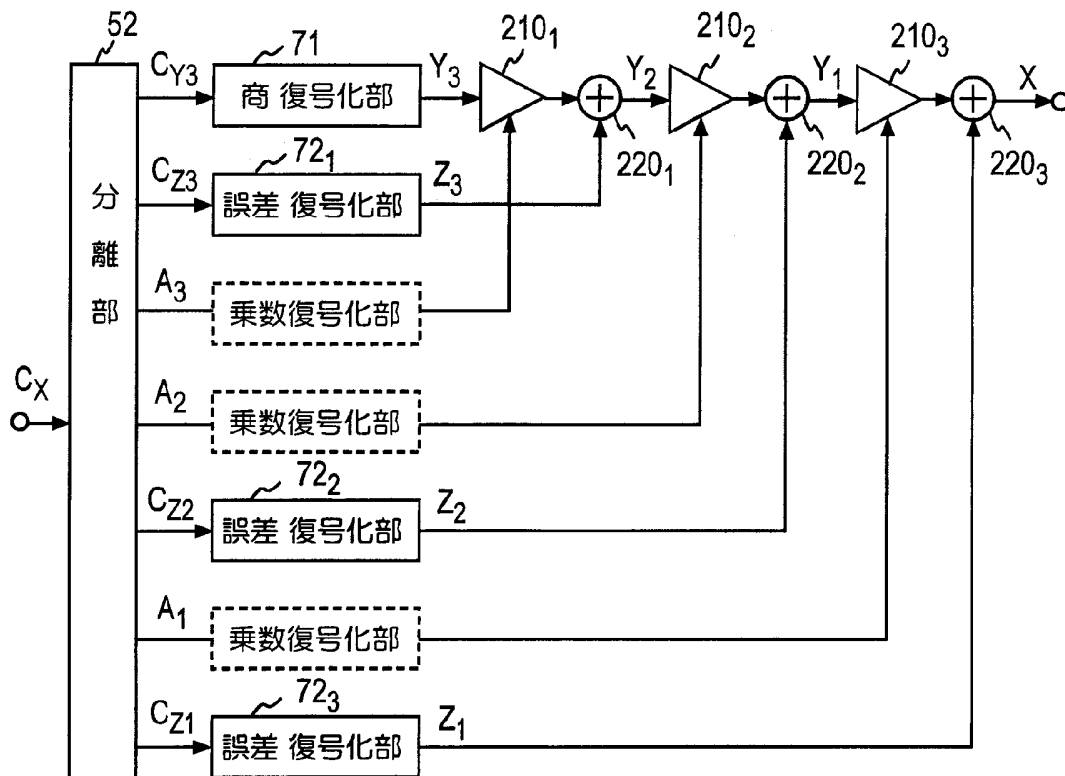


図24

[図25]

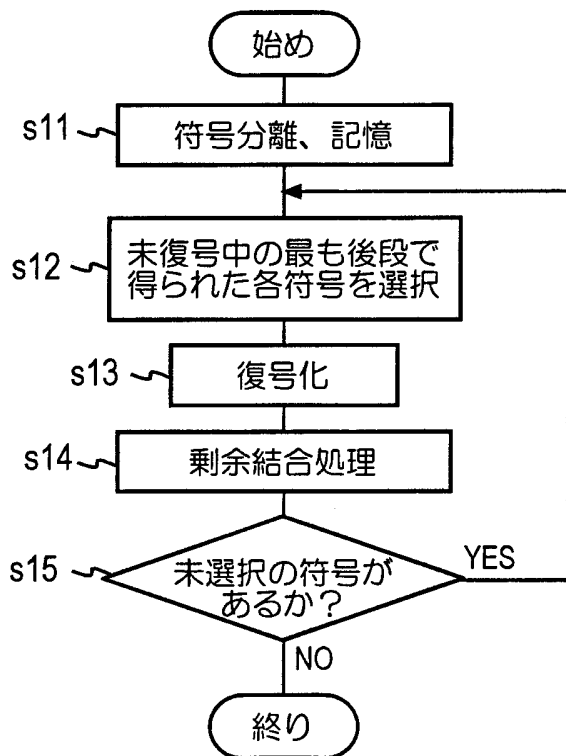


図25

[図26]

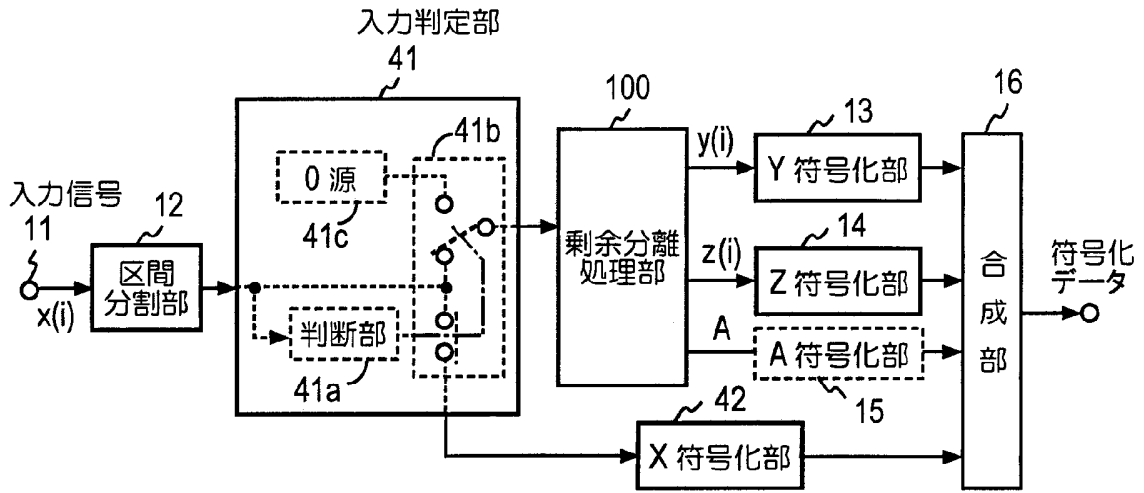


図26A

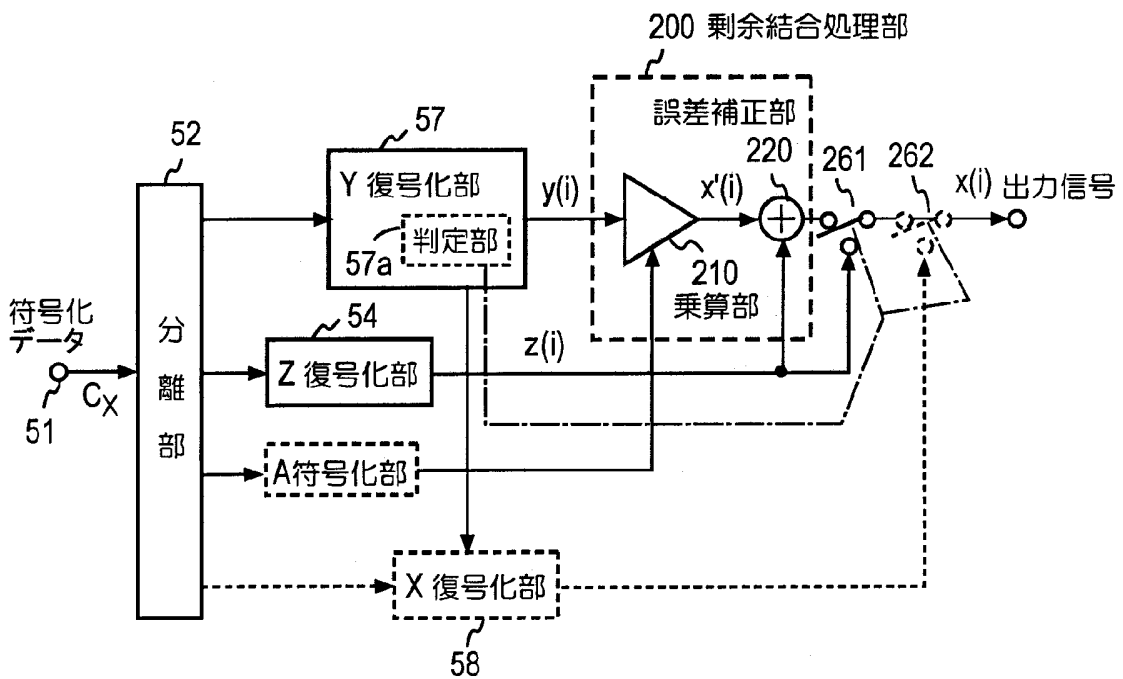


図26B

[図27]

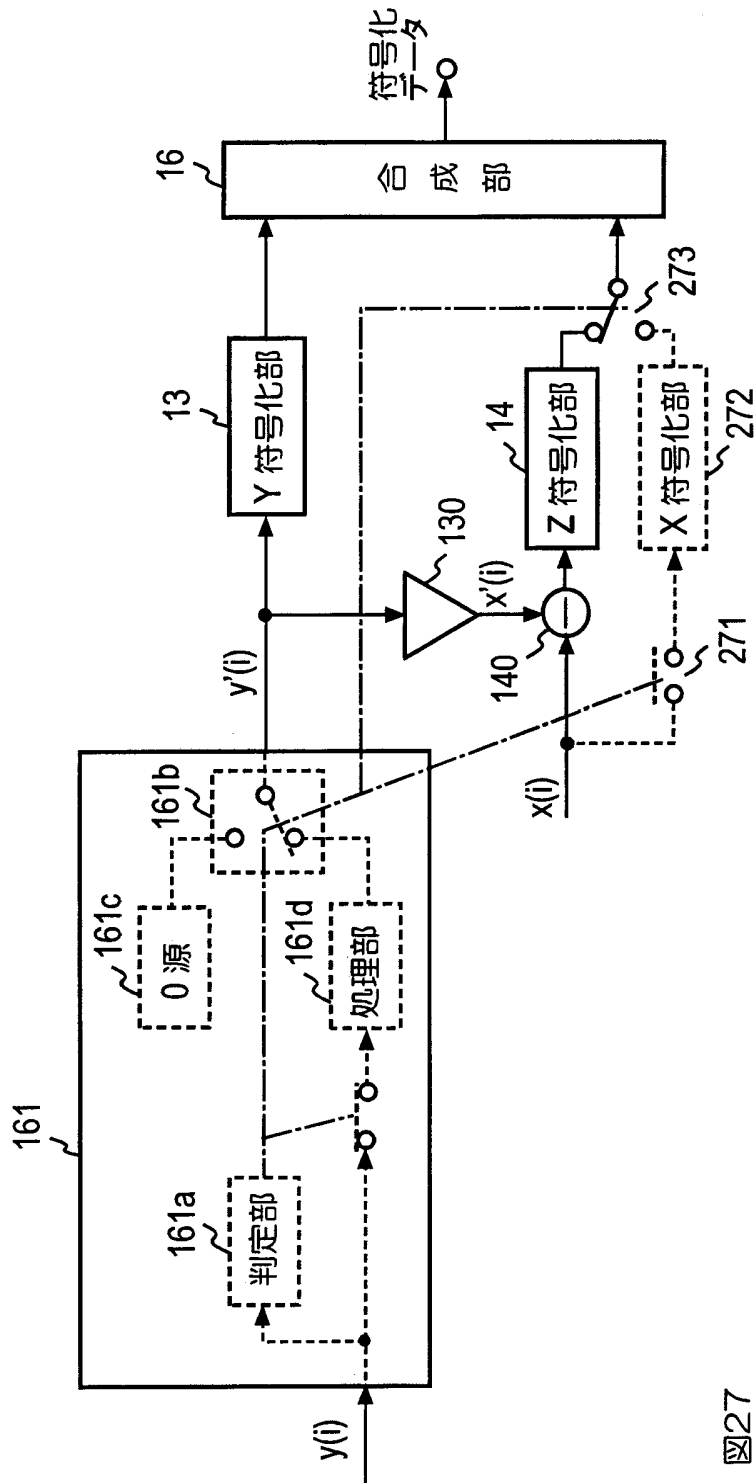


図27

[図28]

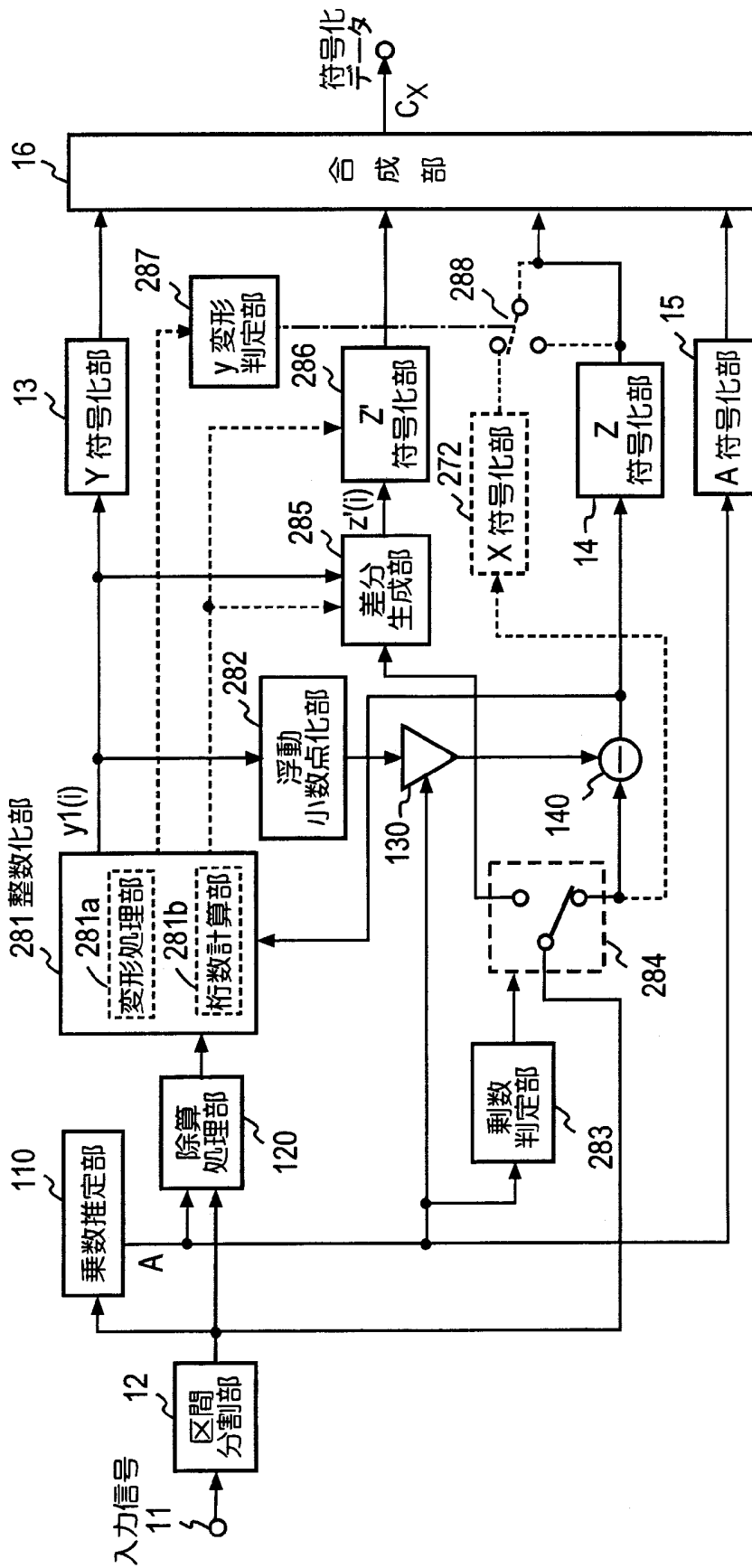


図28

[図30]

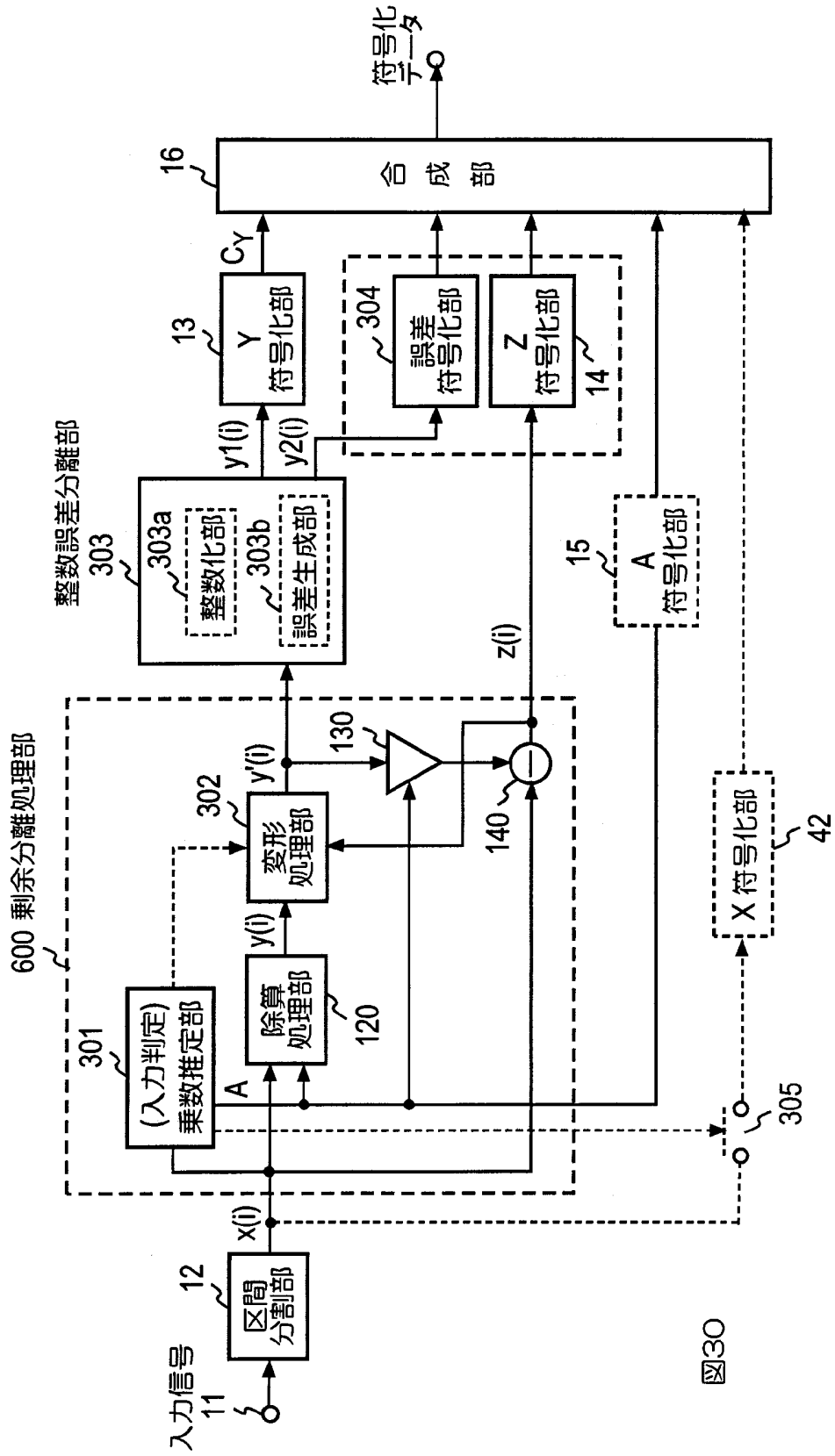


図30

[図31]

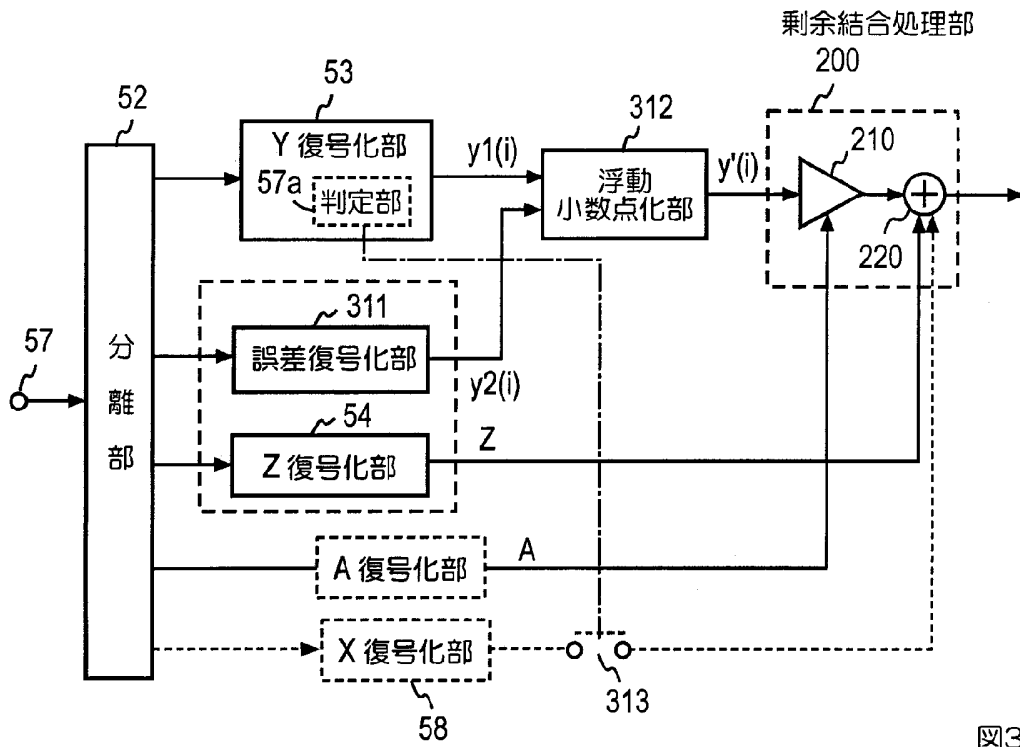


図31

[図32]

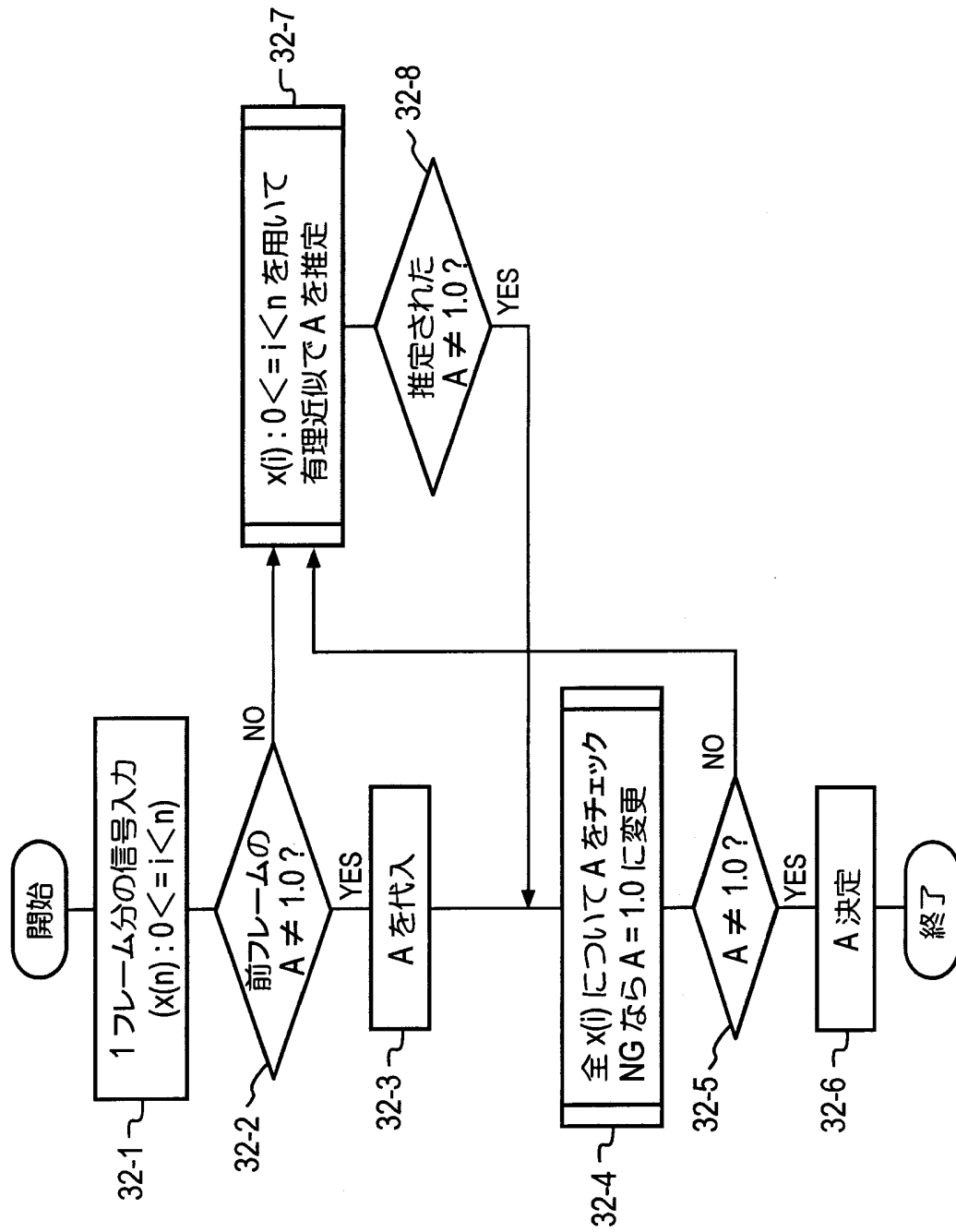
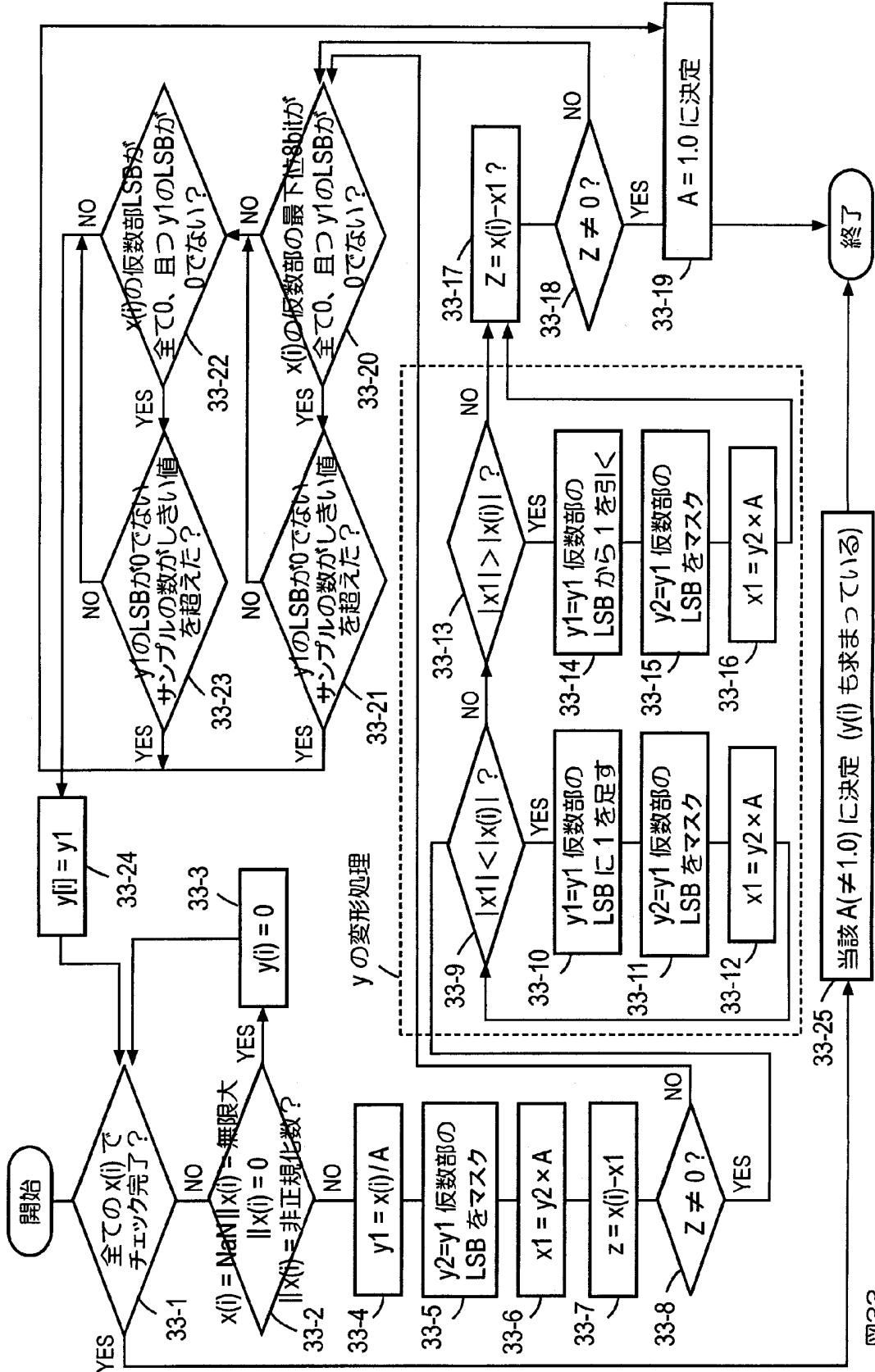


図32

[図33]



[図34]

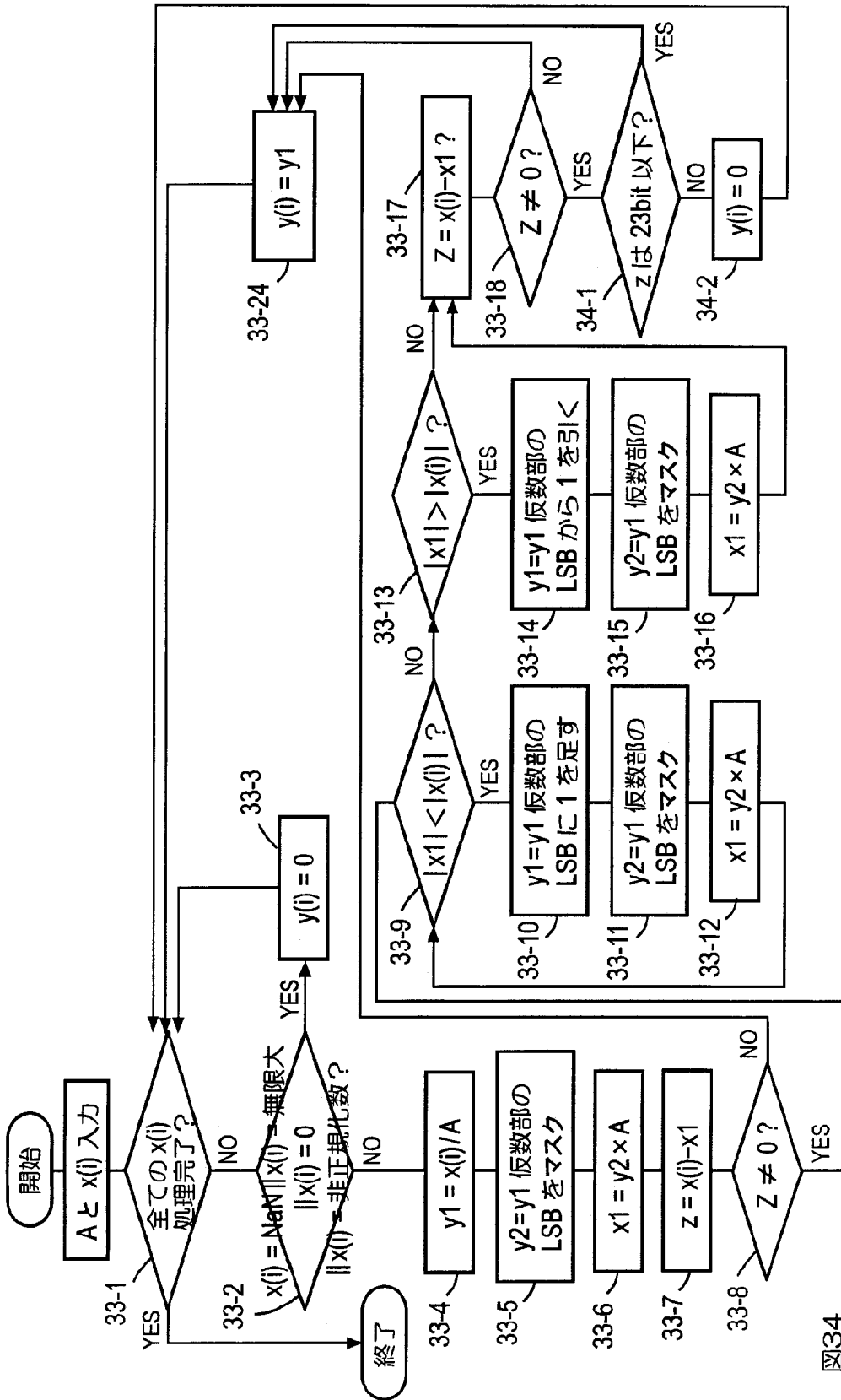


図34

[図35]

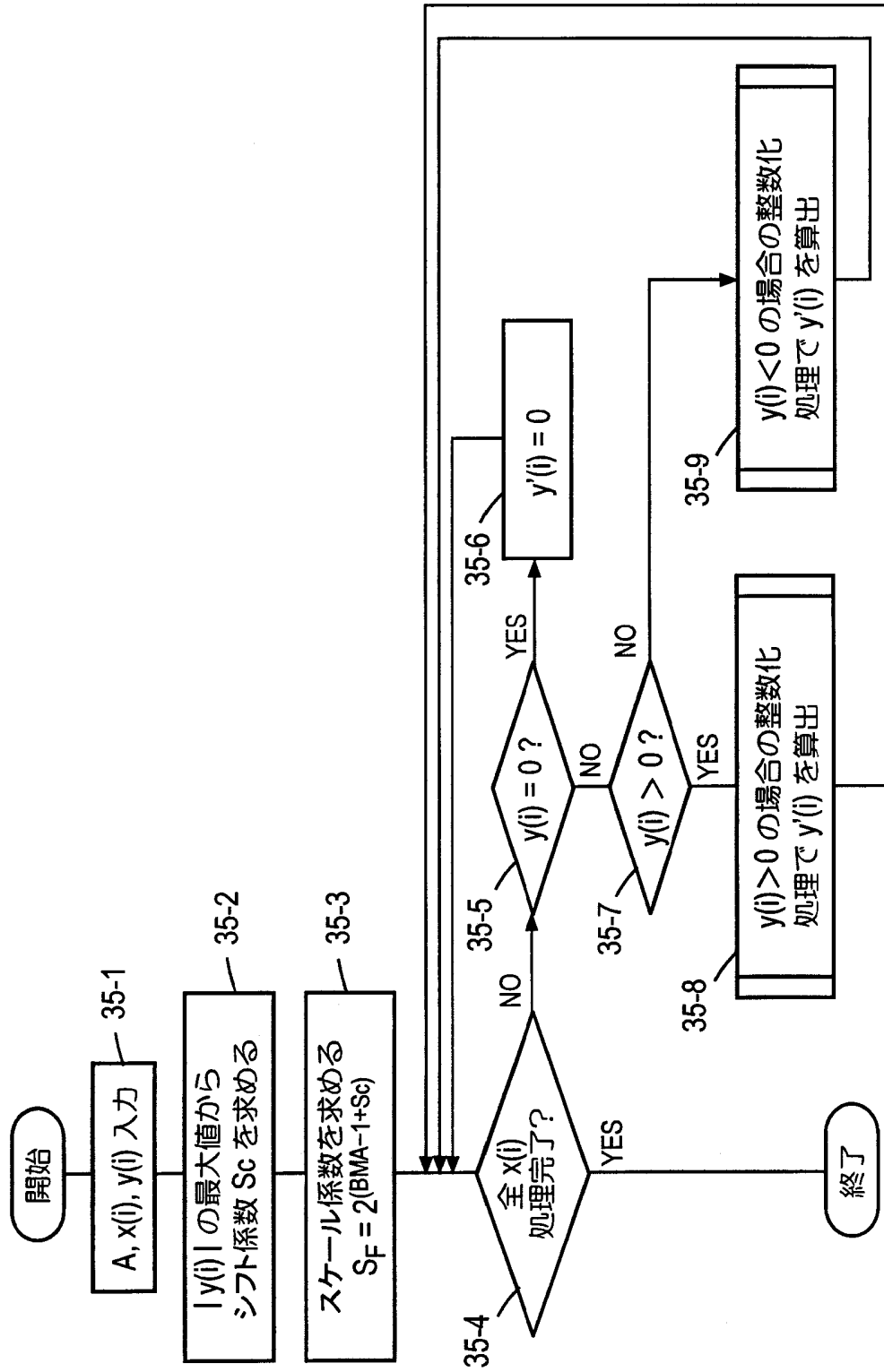


図35

[図36]

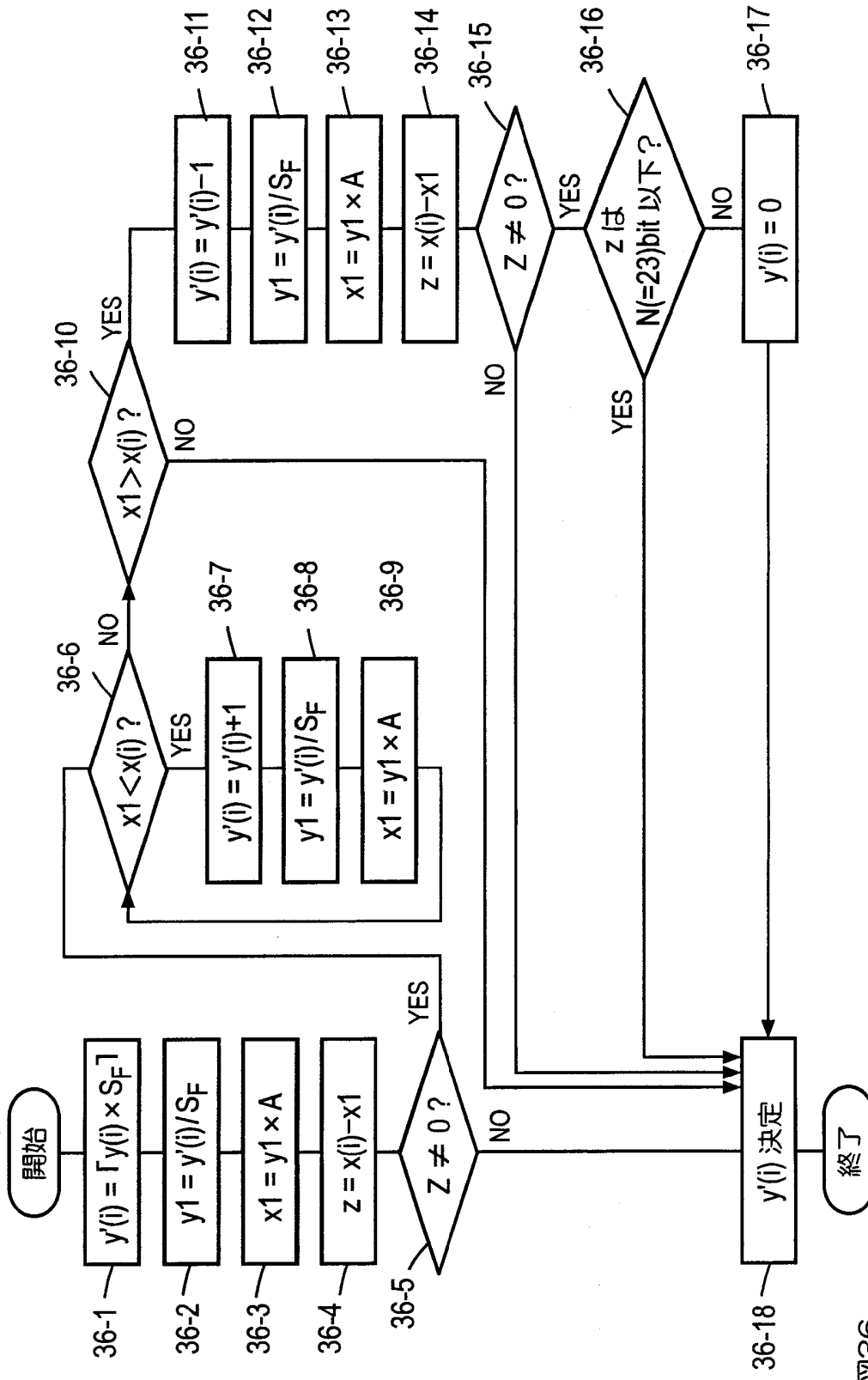


図36

[図37]

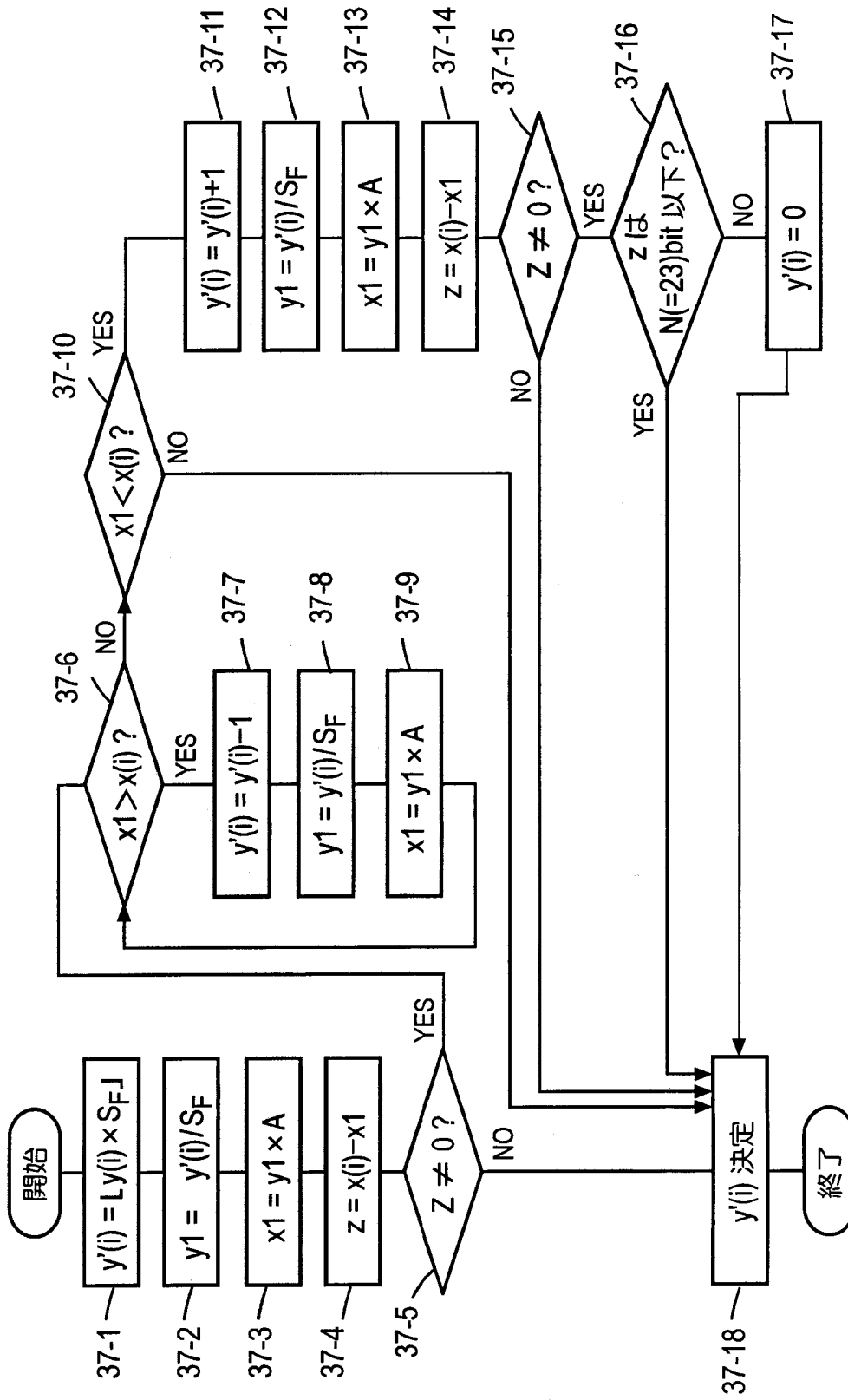


図37

[図38]

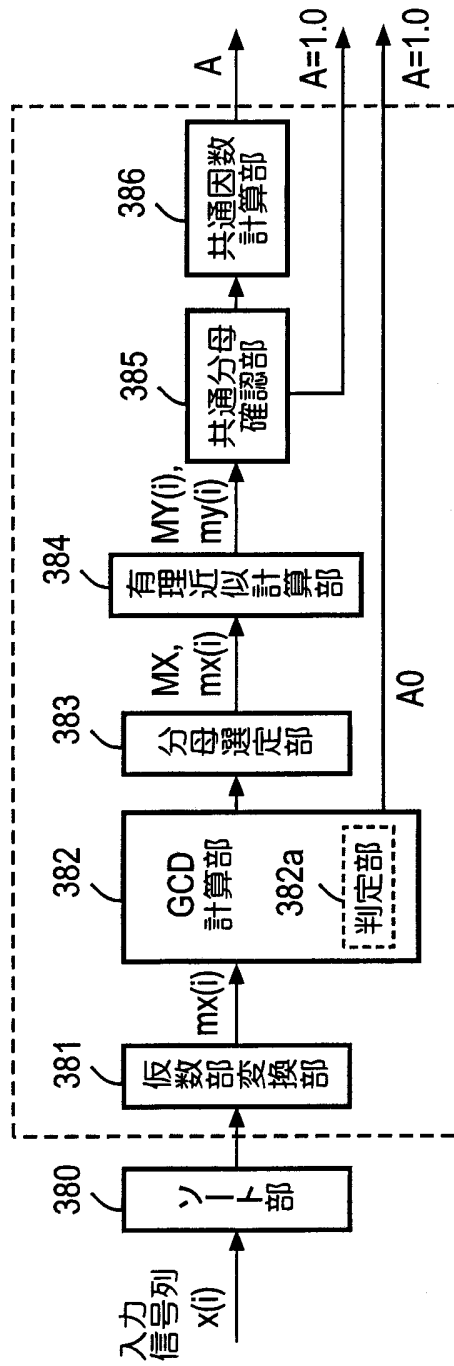


図38

[図39]

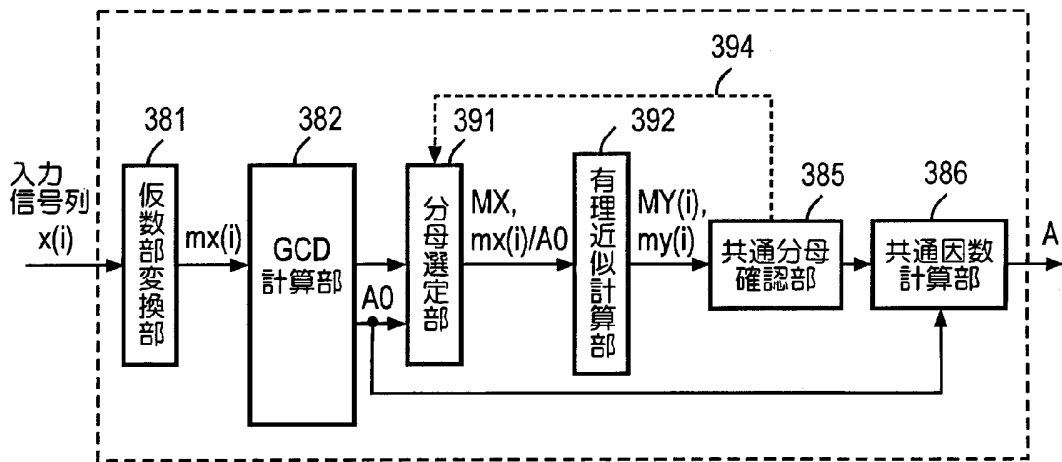


図39

[図40]

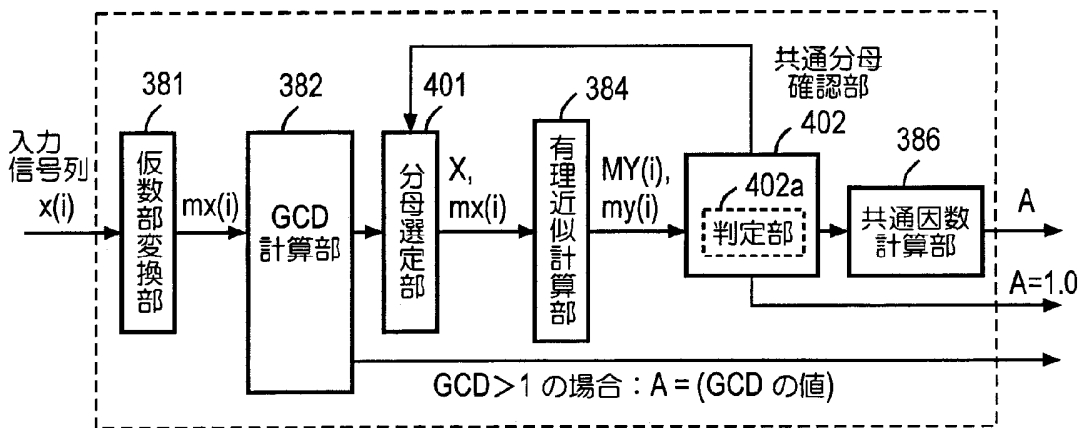


図40

[図41]

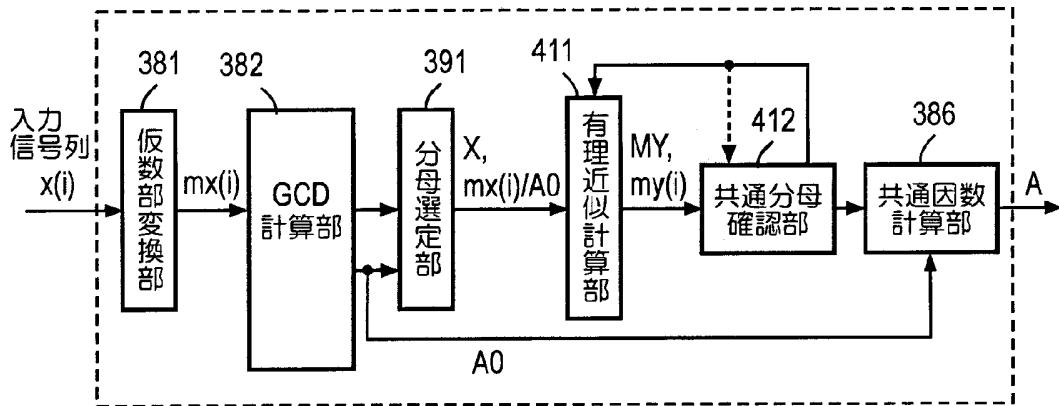


図41

[図42]

A	$z(i)$	100100100100
	$z(i)(1)$	100100100100
	相関値	000000000000

	j: 1 2 3 4 5 6 7 8 9
	$x(0)(j): 5 0 2 0 0 0 3 2 0$
	$x(1)(j): 3 0 4 1 0 0 0 8 0$
B	$x(2)(j): 2 0 1 0 0 3 1 1 0$
	$x(3)(j): 4 0 2 1 0 8 0 4 0$
	$x(4)(j): 9 0 6 0 0 8 0 6 6$
	合計 : 23 0 15 2 0 19 4 21 6
	判定 : 1 0 1 0 0 1 0 1 0

図42

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2005/018978

<p>A. CLASSIFICATION OF SUBJECT MATTER G06F5/00(2006.01) , G10L19/00(2006.01) , H03M7/30(2006.01)</p> <p>According to International Patent Classification (IPC) or to both national classification and IPC</p>											
<p>B. FIELDS SEARCHED</p> <p>Minimum documentation searched (classification system followed by classification symbols) G06F5/00 , G10L19/00 , H03M7/30</p> <p>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 1996-2006 Kokai Jitsuyo Shinan Koho 1971-2006 Toroku Jitsuyo Shinan Koho 1994-2006</p> <p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)</p>											
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1"> <thead> <tr> <th>Category*</th> <th>Citation of document, with indication, where appropriate, of the relevant passages</th> <th>Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>JP 11-145843 A (Sharp Corp.) , 28 May, 1999 (28.05.99) , Par. Nos. [0033] to [0045] (Family: none)</td> <td>1-30</td> </tr> <tr> <td>A</td> <td>JP 10-078863 A (Motorola, Inc.) , 24 March, 1998 (24.03.98) , All pages; all drawings & US 5768170 A1 & EP 821303 A2</td> <td>1-30</td> </tr> </tbody> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	A	JP 11-145843 A (Sharp Corp.) , 28 May, 1999 (28.05.99) , Par. Nos. [0033] to [0045] (Family: none)	1-30	A	JP 10-078863 A (Motorola, Inc.) , 24 March, 1998 (24.03.98) , All pages; all drawings & US 5768170 A1 & EP 821303 A2	1-30
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.									
A	JP 11-145843 A (Sharp Corp.) , 28 May, 1999 (28.05.99) , Par. Nos. [0033] to [0045] (Family: none)	1-30									
A	JP 10-078863 A (Motorola, Inc.) , 24 March, 1998 (24.03.98) , All pages; all drawings & US 5768170 A1 & EP 821303 A2	1-30									
<p><input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.</p>											
<p>* Special categories of cited documents:</p> <table border="0"> <tr> <td style="vertical-align: top;"> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </td> <td style="vertical-align: top;"> <p>"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</p> <p>"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</p> <p>"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</p> <p>"&" document member of the same patent family</p> </td> </tr> </table>			<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"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</p> <p>"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</p> <p>"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</p> <p>"&" document member of the same patent family</p>							
<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"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</p> <p>"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</p> <p>"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</p> <p>"&" document member of the same patent family</p>										
<p>Date of the actual completion of the international search 10 January, 2006 (10.01.06)</p>		<p>Date of mailing of the international search report 17 January, 2006 (17.01.06)</p>									
<p>Name and mailing address of the ISA/ Japanese Patent Office</p>		<p>Authorized officer</p>									
<p>Facsimile No.</p>		<p>Telephone No.</p>									

A. 発明の属する分野の分類 (国際特許分類 (IPC)) Int.Cl. G06F5/00 (2006.01), G10L19/00 (2006.01), H03M7/30 (2006.01)		
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. G06F 5/00, G10L 19/00, H03M 7/30		
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 1971-2006年 日本国実用新案登録公報 1996-2006年 日本国登録実用新案公報 1994-2006年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP 11-145843 A (シャープ株式会社) 1999.05.28, 段落0033-0045 (ファミリーなし)	1-30
A	JP 10-078863 A (モトローラ・インコーポレイテッド) 1998.03.24, 全頁、全図 & US 5768170 A1 & EP 821303 A2	1-30
<input type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日	10.01.2006	国際調査報告の発送日
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 友章 電話番号 03-3581-1101 内線 3521	5E 9376