

(51)Int.Cl. <sup>6</sup>	識別記号	F I
H 0 3 M 13/00		H 0 3 M 13/00
G 0 6 F 11/10	3 3 0	G 0 6 F 11/10 3 3 0 P

審査請求 未請求 請求項の数3 OL (全 11 頁)

(21)出願番号 特願平9-124411

(22)出願日 平成9年(1997)5月14日

(71)出願人 390020248

日本テキサス・インスツルメンツ株式会社  
東京都港区北青山3丁目6番12号 青山富士ビル

(72)発明者 沖田 茂

東京都港区北青山3丁目6番12号 青山富士ビル  
日本テキサス・インスツルメンツ株式会社内

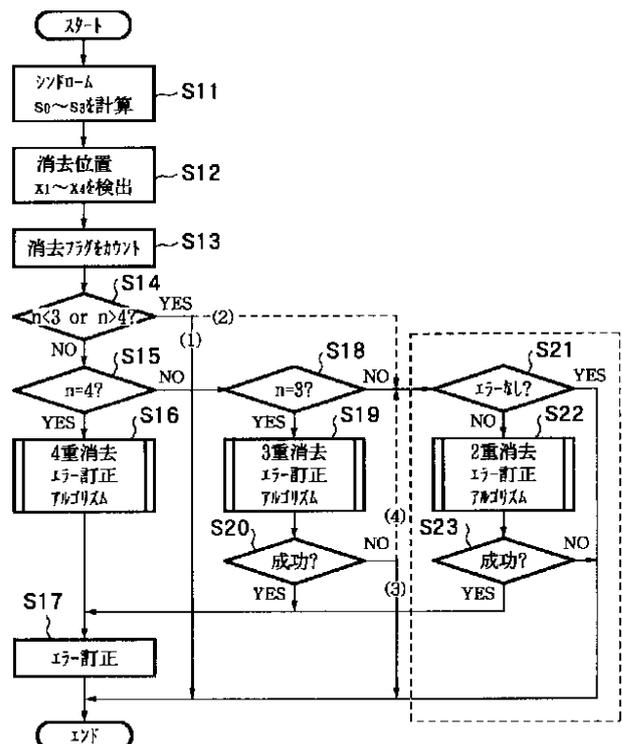
(74)代理人 弁理士 佐藤 隆久

(54)【発明の名称】 リードソロモン復号方法

(57)【要約】

【課題】 誤訂正を効果的に防止できるリードソロモン復号方法を提供する。

【解決手段】  $2t$  (正の整数) シンボルのパリティデータを持つ、リードソロモン符号化系列に対して、誤りのシンボルの位置を示す消失位置を用いて誤り訂正を行い、復号するリードソロモン復号方法であって、前記消失位置から誤りのシンボルの数  $n$  を求め (S13)、 $0 < m_1 < n < m_2 < 2t$  ( $m_1$ 、 $m_2$  および  $n$  は正の整数) である場合に (S14, S15, S18)、 $n$  シンボルの消失エラー訂正の復号演算を行って誤り値を求め (S19)、前記誤り値および前記消失位置を用いて、訂正操作を行う (S17)。



【特許請求の範囲】

【請求項 1】  $2t$  (正の整数) シンボルのパリティデータを持つ、リードソロモン符号化系列に対して、誤りのシンボルの位置を示す消失位置を用いて誤り訂正を行い、復号するリードソロモン復号方法において、前記消失位置から誤りのシンボルの数  $n$  を求め、  
 $0 < m_1 < n < m_2 < 2t$  ( $m_1$ 、 $m_2$  および  $n$  は正の整数) である場合に、  
 $n$  シンボルの消失エラー訂正の復号演算を行って誤り値を求め、  
 前記誤り値および前記消失位置を用いて、訂正操作を行うリードソロモン復号方法。

【請求項 2】  $n = 2t$  である場合に、 $2t$  シンボルの消失エラー訂正を行う請求項 1 に記載のリードソロモン復号方法。

【請求項 3】  $n < m_1$  あるいは  $n > m_2$  である場合に、前記消失位置を用いずに  $t$  重以下のエラー訂正を行う請求項 1 に記載のリードソロモン復号方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、記録媒体やデジタル伝送の誤り訂正符号化として用いられるリードソロモン符号化された信号を復号するリードソロモン復号方法に関する。

【0002】

【従来の技術および発明が解決しようとする課題】リード・ソロモン符号(以下RS符号)は、その符号化効率の良さとバーストエラーに対する適正から、主に記録媒体やデジタル伝送の外符号に用いられている。例えばコンパクトディスクで採用されているエラー訂正符号は、CIRC訂正符号(クロスインターリーブ・リード・ソロモン符号)と称され、インターリーブの技法と組み合わせた積符号である。その外符号としてRS(28、24)符号が、内符号としてRS(32、28)符号が採用されていて、それぞれC2符号及びC1符号と呼ばれる。いずれの符号とも、ひとつのRS符号化シンボルは1バイトで構成され、ひとつのRS符号化ブロックは4バイトのパリティ検査列を含んでいる。

【0003】一般に、RS符号は $2t$ シンボルの検査列で $t$ シンボルの訂正が可能である。 $t$ シンボルの訂正には $t$ 個の誤り位置とそのそれぞれの誤りに対応した $t$ 個の誤りの値を知る必要がある。RS符号は $t$ 個の誤りの\*

$$\begin{aligned}
 s_0 &= e_1 + e_2 + e_3 + e_4 \\
 s_1 &= x_1 \cdot e_1 + x_2 \cdot e_2 + x_3 \cdot e_3 + x_4 \cdot e_4 \\
 s_2 &= x_1^2 \cdot e_1 + x_2^2 \cdot e_2 + x_3^2 \cdot e_3 + x_4^2 \cdot e_4 \\
 s_3 &= x_1^3 \cdot e_1 + x_2^3 \cdot e_2 + x_3^3 \cdot e_3 + x_4^3 \cdot e_4
 \end{aligned}$$

.... (2)

【0009】ここで記号‘ $\cdot$ ’はガロア体上の乗算を示し、記号‘ $+$ ’はガロア体上の加算を示す。以下、ある

ガロア体の元同士の四則演算は、そのガロア体上での演算を示すこととする。前記式(2)を連立方程式を解い

\*発生に対し、復号側でシンドローム演算をすることで $2t$ 個の線形独立な方程式を得る。この方程式を解くことで $2t$ 個の未知数である、前記 $t$ 個の誤り位置とそのそれぞれの誤り位置に対応した前記 $t$ 個の誤りの値を求めることができる。

【0004】一方、CIRC符号のように積符号の構成を取っているものは、内符号に対する内側RS復号において訂正できなかったRS符号化ブロックや誤訂正の可能性の比較的高いRS符号化ブロックに消失フラグを付加することで、外符号に対応した外側RS復号において消失エラー訂正が可能になる。消失フラグが付加された内符号の消失シンボルはデ・インターリーブによって複数の外側のRS符号化ブロックに分散される。消失エラー訂正では、前記消失シンボルに誤りが存在すると仮定してシンドローム演算から得られる連立方程式を解く。誤り位置を既知として解くので最大 $2t$ 個の誤りを値を求めることができる。即ち、 $2t$ シンボルの検査列を持つRS符号に対しては、消失エラー訂正を実行することで最大 $2t$ シンボルの誤り訂正が可能である。

【0005】CIRC符号を例に挙げて、消失エラー訂正の手法を説明する。CIRC符号の場合、内符号であるC1符号のRS復号(C1復号)において消失フラグを付加することで、外符号であるC2符号のRS復号(C2復号)で消失エラー訂正が可能である。C1符号、C2符号ともに $t = 2$ であるから、C1復号は最大2バイトの訂正が、C2復号の消失エラー訂正では最大4バイトの訂正がそれぞれ可能である。そのC2復号におけるシンドローム $s_0 \sim s_3$ と、誤り値 $e_1 \sim e_4$ が次のようにして求められる。CIRC符号の符号生成多項式 $G_e(x)$ は、下記式(1)で示される。

【0006】

【数1】

$$G_e(x) = \prod_{j=0}^3 (x + \alpha^j) \dots (1)$$

【0007】ここで $\alpha$ はガロア体の原始元である。このとき、受信系列からシンドローム演算により得た $s_0 \sim s_3$ は、前記 $x_1 \sim x_4$ および $e_1 \sim e_4$ との間に下記式(2)で示される関係がある。

【0008】

【数2】

$$\begin{aligned}
 s_0 &= e_1 + e_2 + e_3 + e_4 \\
 s_1 &= x_1 \cdot e_1 + x_2 \cdot e_2 + x_3 \cdot e_3 + x_4 \cdot e_4 \\
 s_2 &= x_1^2 \cdot e_1 + x_2^2 \cdot e_2 + x_3^2 \cdot e_3 + x_4^2 \cdot e_4 \\
 s_3 &= x_1^3 \cdot e_1 + x_2^3 \cdot e_2 + x_3^3 \cdot e_3 + x_4^3 \cdot e_4
 \end{aligned}$$

.... (2)

3

4

て、未知数である誤り値  $e_1 \sim e_4$  を求めると次のよう \* 【0010】  
 になるまず、 $e_4$  は、下記式 (3) として得られる。 \* 【数3】

$$e_4 \leftarrow \frac{s_3 + (x_1 + x_2 + x_3) \cdot s_2 + (x_1 \cdot x_2 + x_2 \cdot x_3 + x_3 \cdot x_1) \cdot s_1 + x_1 \cdot x_2 \cdot x_3 \cdot s_0}{(x_4 + x_1) \cdot (x_4 + x_2) \cdot (x_4 + x_3)}$$

.... (3)

【0011】この得られた  $e_4$  を前記式 (2) に代入し ることで、前記式 (2) の連立方程式は下記式 (5) に  
 て3つの等式から成る連立方程式を再構成する。即ち、 変形される。

CIRC符号で用いられているガロア体は加算と減算が 【0012】  
 同じであることに注意して下記式 (4) のように修正す 10 【数4】

$$\begin{aligned} s_0 &\leftarrow s_0 + e_4 \\ s_1 &\leftarrow s_1 + x_4 \cdot e_4 \\ s_2 &\leftarrow s_2 + x_4^2 \cdot e_4 \end{aligned} \quad \dots (4)$$

【0013】 【数5】

$$\begin{aligned} s_0 &= e_1 + e_2 + e_3 \\ s_1 &= x_1 \cdot e_1 + x_2 \cdot e_2 + x_3 \cdot e_3 \\ s_2 &= x_1^2 \cdot e_1 + x_2^2 \cdot e_2 + x_3^2 \cdot e_3 \end{aligned} \quad \dots (5)$$

【0014】これは、連立方程式の解を、手計算で順に (6) のようになる。  
 求めるときによく用いられる手法である。次に、式 20 【0015】

(5) の連立方程式を解いて  $e_3$  を求めると、下記式 【数6】

$$e_3 \leftarrow \frac{s_2 + (x_1 + x_2) \cdot s_1 + x_1 \cdot x_2 \cdot s_0}{(x_3 + x_1) \cdot (x_3 + x_2)} \quad \dots (6)$$

【0016】同様にして修正を施すことで、前記式 【0017】  
 (5) の連立方程式は下記式 (7) , (8) のように変 形される。 【数7】

$$\begin{aligned} s_0 &\leftarrow s_0 + e_3 \\ s_1 &\leftarrow s_1 + x_3 \cdot e_3 \end{aligned} \quad \dots (7)$$

【0018】 【数8】

$$\begin{aligned} s_0 &= e_1 + e_2 \\ s_1 &= x_1 \cdot e_1 + x_2 \cdot e_2 \end{aligned} \quad \dots (8)$$

【0019】さらに、式 (8) の連立方程式を解いて  $e_2$  を求めると、下記式 (9) となる。 【0020】  
 【数9】

$$e_2 \leftarrow \frac{s_1 + x_1 \cdot s_0}{x_2 + x_1} \quad \dots (9)$$

【0021】次に、この求めた  $e_2$  を前記式 (8) に代 \* 【0022】  
 入して下記式 (10) を得る。 \* 【数10】

$$e_1 \leftarrow s_0 + e_2 \quad \dots (10)$$

【0023】このようにして、誤り値  $e_1 \sim e_4$  を順次 40 位置  $x'_1, x'_2$  を求める。以上は、4重消去エラー  
 求めることができる。上述した手法において、情報とし 訂正、すなわち、消去位置の数が4の場合の復号演算処  
 て元来もっているものと、実際の復号の際に行われる演 理過程である。

算操作とを区別するために、記号 '=' と ' ' とを使 40 【0024】従来手法による消失エラー訂正処理のフロ  
 い分けている。つまり、実際の復号演算に対応するの ーチャートを図2に示す。図2に示すように、受信系列  
 は、式 (3) 、 (4) 、 (6) 、 (7) 、 (9) および からシンドローム  $s_0 \sim s_3$  を計算し (ステップS  
 (10) であり、少なくとも、ガロア体上の加算が23 1) 、その消失フラグから消失位置  $x_1 \sim x_4$  を得る  
 回、乗算が17回、除算が3回必要である。一方、消去 (ステップS2) 。そして、消失フラグの数をカウント  
 エラー訂正を行わない場合には、C2復号で最大2パイ し、このカウント値を  $n$  とする (ステップS3) 。

ットまで訂正 (2重エラー訂正) ができる。このときは、 【0025】次に、 $n = 4$  (2t) の場合には (ステッ  
 シンドローム  $s_0 \sim s_3$  から、誤り値  $e_1, e_2$  と誤り 50 プS4) 、前記式 (3) , (4) , (6) , (7) ,

(9)および(10)のアルゴリズムにより、4重消去エラー訂正の復号演算を実行して、誤り値 $e_1 \sim e_4$ を求める(ステップS5)。そして、誤り値 $e_1 \sim e_4$ と消去位置 $x_1 \sim x_4$ とを用いて訂正操作を行う(ステップS6)。

【0026】また、 $n = 3$  ( $t < n < 2t$ )の場合には(ステップS7)、前記式(6)、(7)、(9)および(10)のアルゴリズムにより、3重消去エラー訂正の復号演算を実行して、誤り値 $e_1 \sim e_3$ を求める(ステップS8)。そして、誤り値 $e_1 \sim e_3$ と消去位置 $x_1 \sim x_3$ とを用いて訂正操作を行う(ステップS6)。

【0027】一方、 $n = 2$  ( $t$ )の場合には、これに対応した所定の誤り訂正を実行する(ステップS9、S6)。

【0028】上述したように、4重消去エラー訂正ではシンドローム $s_0 \sim s_3$ の全てを使用しているが、従来の3重消去エラー訂正では $s_0 \sim s_2$ の3つしか使用していない。また、 $n = 2$  ( $= t$ )の場合の訂正方法が明確化されていない。さらには、従来の手法では、誤った訂正が生じた場合の対策が十分に施されていないという問題がある。

【0029】以下、上述した消去エラー訂正処理を行うリードソロン復号装置の一般的な構成について説明する。図3は、従来のリードソロン復号装置1の構成図である。図3に示すように、リードソロン復号装置1は、メモリブロック2、バスI/Fブロック3および復号演算処理部4を備えている。メモリブロック2は、スクラッチパッドメモリ5、6およびスイッチ7、8を備えている。スイッチ7は、入力データを選択的にスクラッチパッドメモリ5、6に出力する。スイッチ8は、スクラッチパッドメモリ5の記憶内容を選択的に訂正操作実行器12に出力する。

【0030】バスI/Fブロック3は、入力パラメータ演算器9、レジスタB<sub>OUT</sub> 10、バイナリカウンタ11、訂正操作実行器12およびレジスタB<sub>IN</sub> 13を備えている。復号演算処理部4は、スイッチ14、レジスタG<sub>IN</sub> 15、レジスタG<sub>OUT</sub> 16および復号演算器17を備えている。図4は、リードソロン復号装置1の動作時におけるデータおよび構成要素の時系列的な状態を示し、(A)は入力データ、(B)は出力データ、(C)はレジスタB<sub>OUT</sub> 10の記憶状態、(D)はレジスタB<sub>IN</sub> 13の記憶状態、(E)はレジスタG<sub>OUT</sub> 16の記憶状態、(F)はレジスタG<sub>IN</sub> 15の記憶状態、(G)は復号演算器17の処理状態をそれぞれ示している。

【0031】図4に示すように、メモリブロック2のスクラッチパッドメモリ5においてC1符号に関する入力データの入出力を行っているときには、バスI/Fブロック3はC1符号に関する入力データに対して、入力パラメータ演算器9において復号演算入力パラメータの計算を行い、訂正操作実行器12において訂正操作を行って

いる。また、このとき、復号演算処理部4ではC2符号に関する入力データについてC2復号処理が行われている。

【0032】また、スクラッチパッドメモリ6においてC2符号に関する入力データの入出力を行っているときには、バスI/Fブロック3はC2符号に関する入力データに対して、入力パラメータ演算器9において復号演算入力パラメータの計算を行い、訂正操作実行器12において訂正操作を行っている。また、このとき、復号演算処理部4ではC1符号に関する入力データについてC1復号処理が行われている。

【0033】ここで、復号演算入力パラメータとは、具体的には、シンドローム(S)および消失位置(I)である。シンドローム(S)は、図3に示す入力パラメータ演算器9およびレジスタB<sub>OUT</sub> 10の組み合わせによって演算される。図5は、入力パラメータ演算器9およびレジスタB<sub>OUT</sub> 10の構成図である。図5に示すように、入力パラメータ演算器9は、乗算器24~27、加算器20~23、消去フラグ検出器28および分配器29を備えている。また、レジスタB<sub>OUT</sub> 10は、レジスタ30~33およびレジスタ34~37を備えている。

【0034】乗算器24~27は、乗算係数が固定値のガロア体の乗算器であり、それぞれ $x^0$ 、 $x^1$ 、 $x^2$ 、 $x^3$ の乗算を行う。消去フラグ検出器28は、入力データに含まれる消去フラグが「1」であるか否かを検出する。分配器29は、入力データに含まれる各RSシンボル位置と対応して動作するバイナリカウンタ11の出力をレジスタB<sub>OUT</sub> 10のレジスタ34~37のいずれかに出力して記憶させる。このレジスタ34~37の記憶結果が消失位置(I)を示す。

【0035】消失位置(I)は、図3に示す復号演算器17において、後述するコンバータにより、ガロア体の表現に、すなわち「i」から「i」に変換される。具体的には、 $I = \{i_1, i_2, i_3, i_4\}$ が、 $X = \{x_1, x_2, x_3, x_4\}$ に変換される。

【0036】4重消去エラー訂正を実行する場合の前記式(3)、(4)、(6)、(7)および(10)に対応する復号演算は、復号演算処理部4において実行され、レジスタB<sub>OUT</sub> 10からの復号演算入力パラメータ $S = \{s_0, s_1, s_2, s_3\}$ と、 $I = \{i_1, i_2, i_3, i_4\}$ を変換して得られた $X = \{x_1, x_2, x_3, x_4\}$ とを用いて、復号演算出力パラメータ $E = \{e_1, e_2, e_3, e_4\}$ および $X' = X = \{x_1, x_2, x_3, x_4\}$ を得る。消失エラー訂正を行わない場合には、前述した2重エラー訂正においては、復号演算入力パラメータ $S = \{s_0, s_1, s_2, s_3\}$ を用いて、復号演算出力パラメータ $E = \{e_1, e_2\}$ および $X' = \{x'_1, x'_2\}$ を得る。

【0037】誤り位置XあるいはX'は、復号演算処理部4において、後述するコンバータで、指数値に、すな

わち  $i$  から  $i$  に変換される。具体的には、 $X' = X = \{x_1, x_2, x_3, x_4\}$  が  $I = \{i_1, i_2, i_3, i_4\}$  に変換され、 $X' = \{x'_1, x'_2\}$  が  $I' = \{i'_1, i'_2\}$  に変換される。

【0038】図6は、訂正操作実行器12およびレジスタBIN13の構成図である。図6に示すように、訂正操作実行器12は、比較器40、加算器45およびゲートロジック46を備えている。また、レジスタBIN13は、レジスタ41~44およびレジスタ47~50を備えている。

【0039】バスI/Fブロック3は、レジスタGOUT16から入力した誤り値(E)と誤り位置(I')とを用いて、訂正操作を実行する。バイナリカウンタ11は、スイッチ7, 8によるスクラッチパッドメモリ5, 6からの出力の切り換わりに対応して動作し、バイナリカウンタ11のバイナリカウンタ値が誤り位置(I')の構成要素のいずれか( $i'_n$ )と一致したときに、ゲートロジック46から対応する誤り値 $e_n$ が加算器45に出力される。そして、加算器45において、誤り値 $e_n$ と、スイッチ8からのメモリブロックのデータ出力とについて、ガロア体の加算が行われ、加算結果が出力データとなる。

【0040】次に、復号演算処理部4について説明する。図7は、復号演算処理部4の構成図である。図7に示すように、復号演算処理部4は、マイクロコードROM50、シーケンサ51、デスティネーションコントローラ52、ワーキングレジスタ53、GLU(Global Logic Unit)54、ポート選択器55を備えている。

【0041】CIRC符号のように $t$ が4より小さく、連立方程式から解が直接的に求められる場合で、かつ、処理スピードが比較的遅くても良いときには、復号演算処理部4としては、RISC(Reduced Instruction Set Computer)型のものが用いられる。復号演算処理部4では、各演算は逐次的に行われ、演算セットは、GLU54として時分割共有化される。また、一連の演算処理は、マイクロコード化されて、インストラクションコー

- 1:  $v \leftarrow a^v$
- 2:  $w \leftarrow a^w$
- 3:  $v + w \leftarrow v, w$
- 4:  $a^{w+w} \leftarrow v + w$

【0046】除算も同様で、乗算における加算の代わりに減算を実行する。従って、上述した手法では、誤り値 $e_1 \sim e_4$ を求めるには、上記式(3), (4), (6), (7), (9)および(10)における乗算・除算が20回あるので、これだけでも80ステップ以上必要となる。これに、23回の加算を含めると、合計103ステップ以上必要となる。

【0047】本発明は、上述した従来技術に鑑みてなされ、誤訂正を効果的に防止できるリードソロモン復号方

\*ドとしてマイクロコードROM50に格納され、シーケンサ51からのROMアドレスによって、処理順序(メモリからの読み出し順序)が制御される。

【0042】また、途中の演算結果は、予め用意された複数のワーキングレジスタ53に一時的に記憶されるが、どのワーキングレジスタ53に記憶するかもインストラクションコード内のデスティネーションコントロールコードに記述されている。この手法によれば、処理スピードの制限はあるものの、GLU54の時分割共有化による装置の縮小化ができると共に、演算処理のマイクロコード化により設計の自由度を高めることができる。例えば、2つのガロア体の元同士の加算は、各ビットの排他的論理和に相当し、復号演算処理部4では1ステップで実現できる。すなわち、GLU54は、ビット毎の排他的論理和の機能を含んでいる。但し、ガロア体における乗算は、加算に比べてはるかに複雑であり、これをROMを用いて実現しようとする、2バイトのアドレス入力に対して1バイトの出力を得ることになり、非常に規模が大きくなってしまふ。

10 【0043】GLU54の構成について説明する。図8は、GLU54の構成図である。図8に示すように、GLU54は、オペレーションロジック60, 61、コンバータ62, 63およびオペレーションセクタ64を備えている。GLU54では、2つの入力データa, bのガロア体の元のそれぞれを、コンバータ62において、対応する原始元の指数の値に変換、すなわち  $i$  を  $i$  に変換し、指数同士の加算を実行する。そして、その得られた加算結果を、コンバータ63において、対応するガロア体の元に変換、すなわち、 $i$  を  $i$  に変換する。

20 【0044】例えば、 $v$  と  $w$  の乗算を実行して、 $v+w$  を得るには、GLU54において、下記式(11)に示す4つの演算処理が必要で、少なくとも4ステップを要する。

【0045】

【数11】

$$\dots (11)$$

法を提供することを目的とする。また、本発明は、 $n$  シンボルのデータ語について、 $n-2 (=t)$  の場合の訂正方法を明確化したリードソロモン復号方法を提供することを目的とする。

【0048】

【課題を解決するための手段】上述した従来技術の問題点を解決し、上述した目的を達成するために、本発明のリードソロモン復号方法は、 $2t$  (正の整数)シンボルのパリティデータを持つ、リードソロモン符号化系列に

対して、誤りのシンボルの位置を示す消失位置を用いて誤り訂正を行い、復号するリードソロモン復号方法であって、前記消失位置から誤りのシンボルの数  $n$  を求め、 $0 < m_1 - n < m_2 - 2t$  ( $m_1$ 、 $m_2$  および  $n$  は正の整数) である場合に、 $n$  シンボルの消失エラー訂正の復号演算を行って誤り値を求め、前記誤り値および前記消失位置を用いて、訂正操作を行う。

【0049】また、本発明のリードソロモン復号方法は、好ましくは、 $n = 2t$  である場合に、 $2t$  シンボルの消失エラー訂正を行う。

【0050】さらに、本発明のリードソロモン復号方法は、好ましくは、 $n < m_1$  あるいは  $n > m_2$  である場合に、前記消失位置を用いずに  $t$  重以下のエラー訂正を行う。

【0051】

【発明の実施の形態】以下、本発明の実施形態に係わるリードソロモン復号方法について説明する。

**第1実施形態**

図1は、本実施形態に係わるリードソロモン復号方法のフローチャートである。なお、以下、 $m_1 = 3$ 、 $m_2 = 4$  および  $t = 2$  の場合を例示して説明する。このリードソロモン復号方法では、図1に示すように、入力系列からシンドローム  $s_0 \sim s_3$  を計算し(ステップS11)、その消去フラグから消去位置  $x_1 \sim x_4$  を得る(ステップS12)。そして、消去フラグの数をカウントし、このカウント値を  $n$  とする(ステップS13)。

【0052】次に、 $n < m_1$  あるいは  $n > m_2$  であるか、すなわち、 $n < 3$  あるいは  $n > 4$  であるかを判断し(ステップS14)、 $n < 3$  あるいは  $n > 4$  である場合\*

$$\begin{aligned}
 s_0 &= e_1 + e_2 + e_3 & \dots & (12a) \\
 s_1 &= x_1 \cdot e_1 + x_2 \cdot e_2 + x_3 \cdot e_3 & \dots & (12b) \\
 s_2 &= x_1^2 \cdot e_1 + x_2^2 \cdot e_2 + x_3^2 \cdot e_3 & \dots & (12c) \\
 s_3 &= x_1^3 \cdot e_1 + x_2^3 \cdot e_2 + x_3^3 \cdot e_3 & \dots & (12d)
 \end{aligned}$$

【0056】ここで、消失位置以外に誤りが生じていて復号演算の結果が正しくないときにも、求めた  $e_1 \sim e_3$  は前述した式(5)の解であるから上記式(12a)~(12c)は必ず成立する。ところが、式(12d)は、復号演算には用いられていないので必ず成立するとは限らず、消失位置以外に誤りが生じていて復号演算の結果が正しくないときは、むしろ成立しない確率が高い。従って、復号演算の後に、上記式(12d)の右辺を計算し、それが入力系列からシンドローム演算により得た  $s_3$  と一致するかどうかで、正常終了判定を行う(ステップS20)。もし、正常に終了していたならば、誤り値  $e_1 \sim e_3$  と消失位置  $x_1 \sim x_3$  とを用いて訂正操作を行う(ステップS17)。

【0057】一方、正常に終了していない場合には、訂正操作は行わない(図1中実線(3))。あるいは、訂正の速度に余裕がある場合には、消失位置を用い

10

20

40

50

\*には、消去エラー訂正ができないので、訂正操作を行わない(図1中実線(1))。あるいは、消去位置を用いず、従来から行われているエラー訂正を実行し(図1中点線(2))、2重以下の訂正を行う(ステップS21, 22, 23)。

【0053】一方、 $0 < m_1 - n < m_2 - 2t$  である場合に、すなわち、 $3 - n < 4$  の場合には、 $n = 4$  ( $2t$ ) であるか否かを判断し(ステップS15)、 $n = 4$  であれば、従来技術において述べた、式(3)、

(4)、(6)、(7)、(9)および(10)のアルゴリズムにより、4重消失エラー訂正の復号演算を実行して、誤り値  $e_1 \sim e_4$  を求める(ステップS16)。次に、この求めた誤り値  $e_1 \sim e_4$  と、前述した消失位置(すなわち、誤りの位置)  $x_1 \sim x_4$  を用いて訂正操作を行う(ステップS17)。

【0054】また、ステップS15において  $n = 4$  でない場合には、 $n = 3$  であるか否かが判断され(ステップS18)、 $n = 3$  である場合には、前述した式(6)、(7)、(9)および(10)のアルゴリズムにより、3重消失エラー訂正の復号演算を実行して、誤り値  $e_1 \sim e_3$  を求める(ステップS19)。ところで、消失シンボル数が3の場合で、もし復号演算の結果が正しいとき、すなわち、消失位置以外に誤りが生じていない場合には、入力系列からシンドローム演算により得た  $s_0 \sim s_3$  と、消失位置  $x_1 \sim x_3$  および復号および復号演算から求めた誤り値  $e_1 \sim e_3$  との間には、下記式(12a)~(12d)に示す関係がある。

【0055】

$$\begin{aligned}
 \text{【数12】} & \quad e_3 & \dots & (12a) \\
 & \quad x_3 \cdot e_3 & \dots & (12b) \\
 & \quad x_3^2 \cdot e_3 & \dots & (12c) \\
 & \quad x_3^3 \cdot e_3 & \dots & (12d)
 \end{aligned}$$

で、従来から行われているエラー訂正を実行することで、2重以下の訂正を行う(図1中点線(4))。この場合には、誤り値  $e_1$ 、 $e_2$  と共に、新たに誤り位置  $x'_1$ 、 $x'_2$  を得て、訂正操作を行う(ステップS21, 22, 23)。

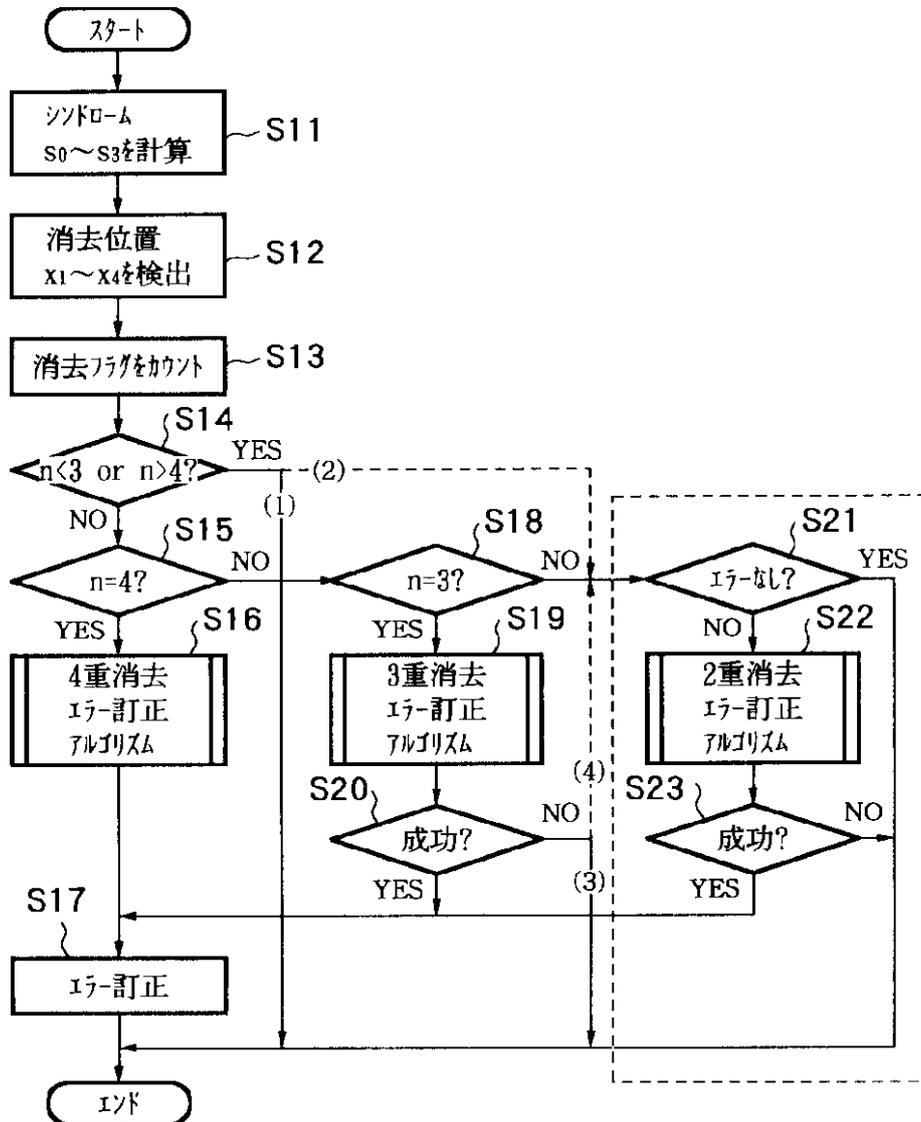
【0058】 $n = 2$  のときには、2重消去エラー訂正および1重エラー訂正を実行するものが考えられる。これは、2重消去エラー訂正のアルゴリズムで消去位置  $x_1$ 、 $x_2$  を用いて誤り値  $e_1$ 、 $e_2$  を求め、1重エラー訂正のアルゴリズムで誤り位置  $x'_3$  と誤り値  $e_3$  を得る。これらの値と入力系列からシンドローム演算により得た  $s_0 \sim s_3$  との間には、下記式(13)に示すような関係がある。

【0059】

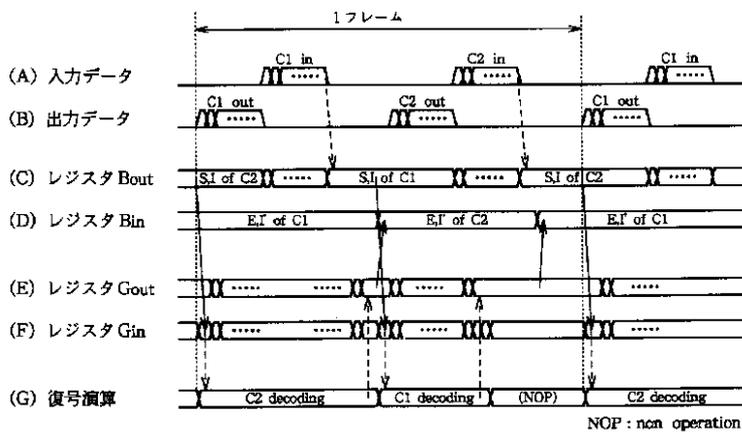
【数13】



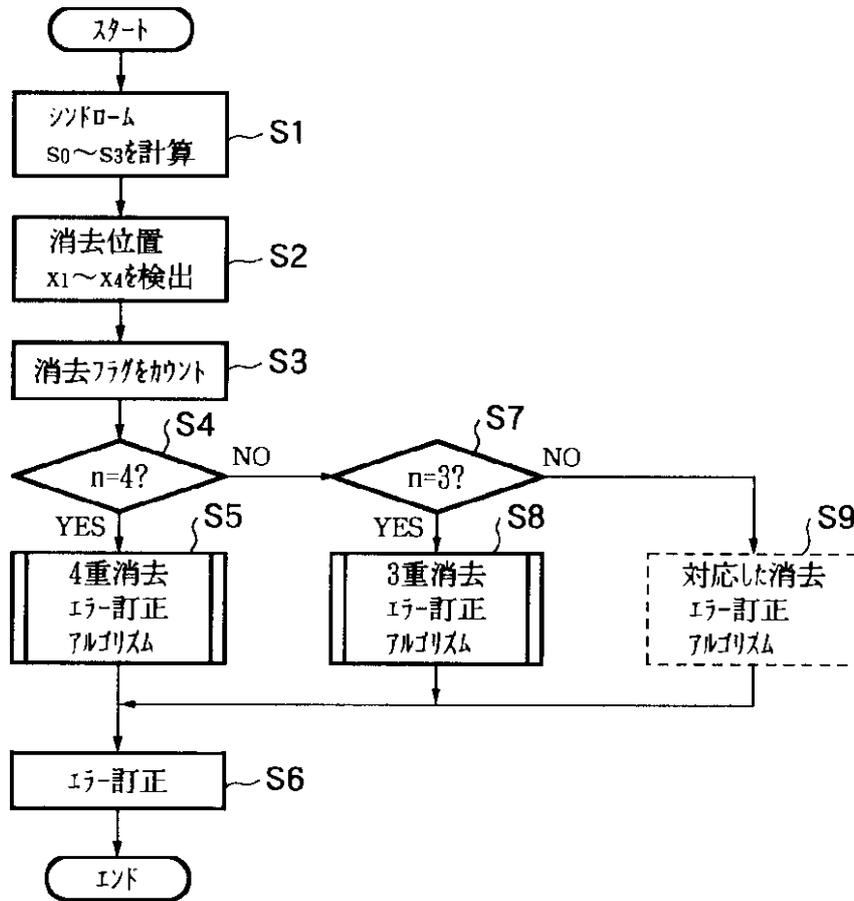
【図1】



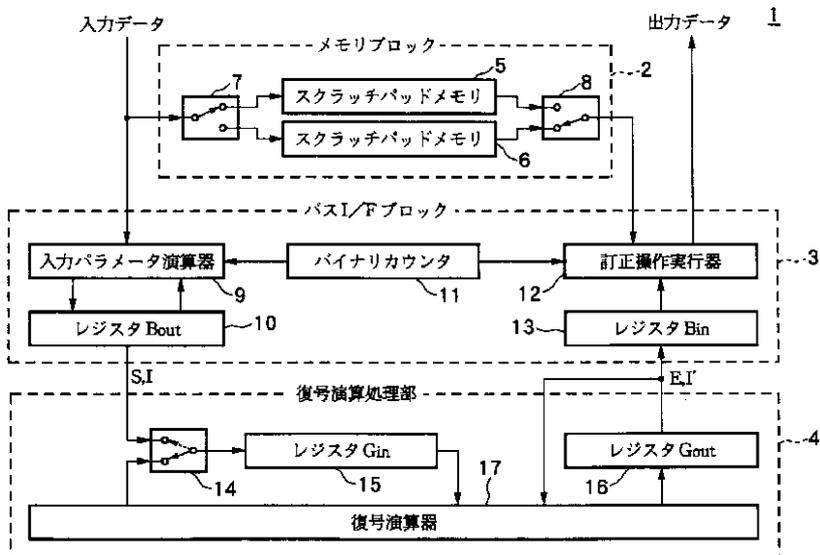
【図4】



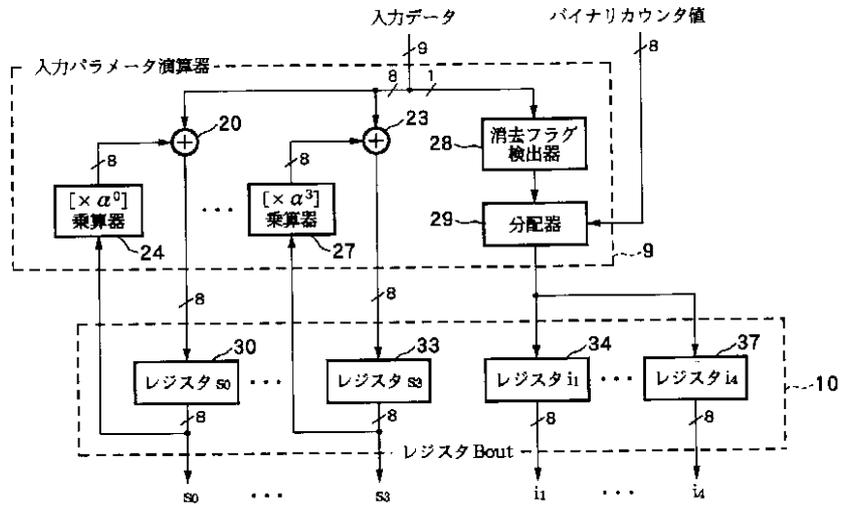
【図2】



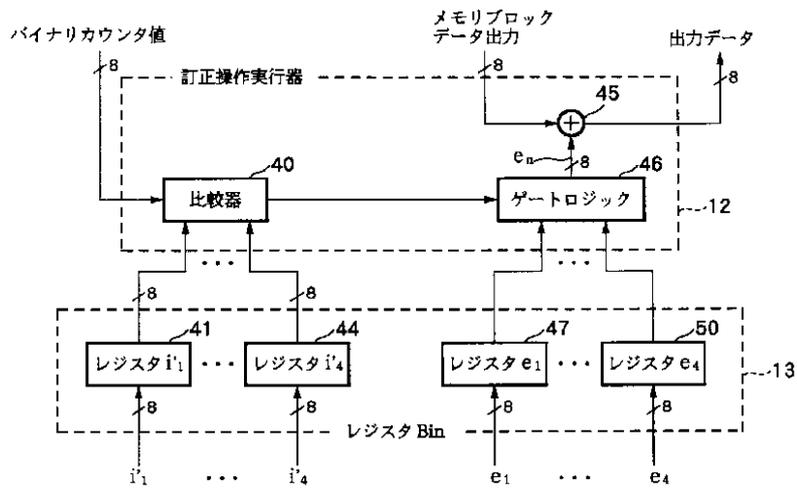
【図3】



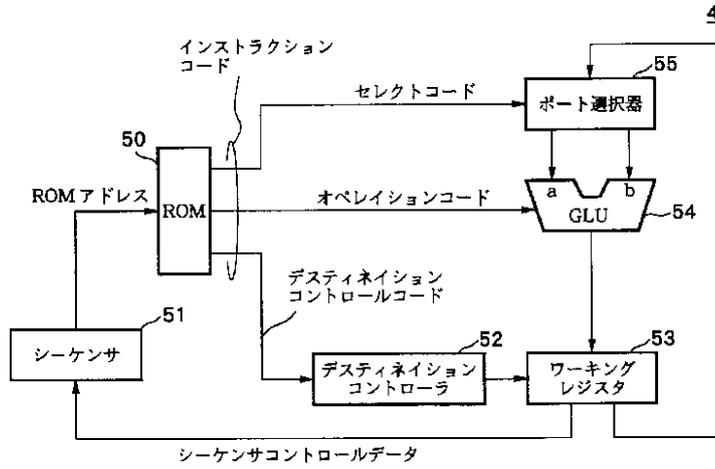
【図 5】



【図 6】



【図 7】



【図 8】

