

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第7部門第3区分

【発行日】平成20年1月31日(2008.1.31)

【公開番号】特開2001-244823(P2001-244823A)

【公開日】平成13年9月7日(2001.9.7)

【出願番号】特願2000-383787(P2000-383787)

【国際特許分類】

H 03M 13/27 (2006.01)

G 06 F 11/10 (2006.01)

H 03M 13/15 (2006.01)

【F I】

H 03M 13/27

G 06 F 11/10 330P

H 03M 13/15

【手続補正書】

【提出日】平成19年12月11日(2007.12.11)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】発明の詳細な説明

【補正方法】変更

【補正の内容】

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、クロスインターリーブドリードソロモンコード(CIRC)の復号方法を使用して、データのストリーム内の誤りを訂正する方法及び装置に関する。

【0002】

【従来の技術】

オーディオ記録及び再生システムは他の形式の通信システムと同様に品質の向上の要求がある。

【0003】

記録媒体の問題は、媒体の欠陥であり、その領域内では不適切なデジタルデータが記録され又は、信頼性をもって再生できない。この結果、再生データにはデジタルデータの伝送と同様に誤りが発生する。デジタルデータの記録中、再生中及び、伝送中には、有限の確率でデジタルデータ中に誤りが発生する。データは典型的には、バイナリユニットで構成され、(8のような)バイナリユニットのグループは、データバイトを構成し、そして、バイトのグループ(例えば、2)はデータワードを構成する。更にデータは、(32又は、28バイトブロックのような)データのブロックに構成される。

【0004】

2つの異なる形式の誤りがある。第1の形式はバイナリビットの2つの可能な値のうちの1つの反対の値への置換であり、単一ビット誤りである。そのような誤りは通常はランダムに発生する。第2の形式の誤りは、エラービットの連続シーケンスである。そのような誤りはバースト誤りと呼ばれるこれらのバースト誤りの長さとそれらの発生頻度もランダムである。

【0005】

コンパクトオーディオディスクで典型的に使用されている1つの誤り訂正コードはいわゆるCIRC訂正コードである。CIRCはクロスインターリーブドリードソロモンコードの略である。CIRCは2ステップの処理を使用する。第1の符号化器を通過すると、デー

タ多項式を表す 24 の連続するデータバイトが生成多項式により除算される。この処理では、4 つのパリティバイトが付加される。結果は、24 データバイトと 4 パリティのブロック (C<sub>2</sub> コード) となる。この連続するバイトのシーケンスは、他の符号化データバイト間でインターリーブされ又は分散される。

#### 【 0 0 0 6 】

インターリーブされたデータは、第 2 の符号化器を通る。第 2 の符号化器は、インターリーブによりデータバイトが異なるシーケンスで提示されることを除けば、第 1 の符号化器と同じであり、データバイトとパリティバイトが共に符号化され、28 バイトのブロック (28 データバイト足す 4 パリティバイト) が 24 バイトの代わりに符号化され、そして、更に 4 パリティバイトが付加される。第 2 の符号化処理の結果は 24 データバイトと 8 パリティバイトからなる 32 バイトブロック (C<sub>1</sub> コード) である。

#### 【 0 0 0 7 】

CIRC 訂正コードの場合には、(28, 4) リードソロモンコードの符号化処理は、24 データシンボル (バイト) に対して行われ、各シンボルは 8 ビットよりなる。典型的なオーディオアプリケーションでは、各オーディオサンプルは 16 ビットを有し、そして、各 8 ビットの 2 シンボルから構成される。このように、各 8 ビットシンボルは、ステレオオーディオデータの 2 チャンネルのうちの 1 つのオーディオサンプルの上位又は下位側のいずれかである。

#### 【 0 0 0 8 】

符号化されたデータは、光学的に記録されそして、読み出される。符号化されたデータが読み出された後には、記録、読み出し又は、伝送の問題により、単一ビット又は、バースト誤りがあり得る。

#### 【 0 0 0 9 】

従来の誤り訂正方法は、CIRC の復号での使用が既知である。そのような方法の例は、U.S.-A-4546474、U.S.-A-4476562 及び、U.S.-A-4497058 に開示されている。CIRC 誤りの復号の従来の方法に従って、いわゆる消失訂正方法に基づき行われる。消失訂正方法では、誤りシンボルのロケーションがピント情報により示される。誤り訂正是、誤りシンボルに行われる。上述の C<sub>1</sub> と C<sub>2</sub> コードでは、検出と 2 誤りまでの訂正が可能である。しかし誤りロケーションが既知の場合には、4 消失訂正まで可能である。従って、誤り訂正能力を上げるために、誤り訂正に対して消失訂正方法の実行が好ましい。さらに、消失訂正方法はバースト誤りを訂正するのに特に効果がある。

#### 【 0 0 1 0 】

CIRC 誤りの復号に使用される従来の方法に従って、C<sub>1</sub> 復号器で 2 誤りまでの訂正が行われる。もし 3 つ以上の誤りが発生したなら、訂正できず、C<sub>1</sub> ピンタ情報は次の段階の C<sub>2</sub> 復号器に送られ、C<sub>1</sub> ピンタ情報を使用して C<sub>2</sub> 復号器で誤り訂正が実行される。

#### 【 0 0 1 1 】

復号において CIRC 訂正の複合的な使用は、誤り訂正システムの訂正能力を向上する。しかし、従来の復号方法 (CIRC 復号器) は、CIRC コードのブロックの複合的な処理ができない。この問題を解決するために、複合的な処理を可能とする更なる手段又は装置を伴なう従来の CIRC 復号器が提供され又は、従来の CIRC 復号器が複合的な処理に使用できるようになされる。

#### 【 0 0 1 2 】

CIRC コードの複合的な処理のための 1 つの方法 (従来の CIRC 復号器) は、U.S.-A-4852099 に記載されている。既知の方法は、誤り訂正システムの訂正能力を向上するために消失訂正方法を使用する。C<sub>1</sub> 復号と C<sub>2</sub> 復号を特定の順序で 2 回行うことが提案されている。例えば、その順序は C<sub>1</sub> 復号、続いて C<sub>2</sub> 復号、続いて C<sub>1</sub> 復号、そして、続いて C<sub>2</sub> 復号である。

#### 【 0 0 1 3 】

この例に従って、C 1 コードワードは実際に(32, 28)リードソロモンコードの復号が行われるC 1 復号器に供給される。2誤りまでの誤り訂正が行われる。C 1 復号器により3又はそれ以上の誤りが検出されたときには、C 1 コードワード内の全シンボルにC 1 ポインタが設定される。そして、C 1 復号器により訂正されたデータと誤りポインタは更に、デインターリーブ処理で処理される。デインターリーバの出力は、C 2 復号器に供給される。4消失までの消失訂正は、C 1 ポインタ情報を使用してC 2 復号器内で行われる。C 2 復号器内での消失訂正の完了で、C 1 ポインタはクリアされ、第2のC 2 復号サイクルには、ポインタ情報は転送されない。

#### 【0014】

第2のサイクルでは、C 2 復号器からのデータは、インターリーバへ供給され、再生されたときであろうのと同じ配置にデータを戻す。その後、第2の復号サイクル内の処理は、第1の復号サイクル内の処理に対応する。この復号法を使用して、CIRCコードの複合的処理が行われるが、しかし、更なるハードウェア(インターリーブブロック)が必要である。

#### 【0015】

CIRCの複合的処理を使用するCIRC復号で使用される他の方法は、U.S.-A-4637021に開示されている。誤り検出と訂正はC 1 復号器とC 2 復号器でデジタルデータバイトの処理を行うことにより達成される。データが処理されるレートを最大にするために、復号器C 1 とC 2 は、システムメモリに蓄積されたデータに対して、C 1 復号器はC 2 復号器の前にデータに動作して、実際に同時に動作する。CIRC訂正コードの開示された復号方法に従って、第1段階でC 1 復号で2重誤り訂正までの誤り処理が行われ、そして、次の段階で、C 1 復号器から得られたC 1 ポインタ情報を参照することにより、C 2 復号で2重誤り訂正までの誤り処理が行われる。この場合、C 1 ポインタ情報は誤り訂正システムの誤り訂正能力を増加するのに使用されない。C 1 ポインタ情報は復号処理の品質をチェックするのに使用される。

#### 【0016】

U.S.-A-4637021に従った方法では、以下のシーケンスで、データバイトがシステムメモリからC 1 とC 2 復号器により読み出される。第1のC 1 復号器は、C 1 ブロックのデータバイト(32バイト)を処理する。最初のC 2 処理は、既にC 1 復号器により処理されたC 2 ブロックのデータバイト(28バイト)を処理する。これで最初のパスは終了する。第2のC 1 復号器は最初のパスで処理されたデータバイトを処理し、第2のC 2 復号器は最初のパスで処理され且つ第2のC 1 復号器により処理されたデータバイトを処理する。これで第2のパスは終了する。復号器サイクルは、第1のパスと第2のパスの両方の実行を含む。各復号器サイクルの最後で、アドレスカウンタが増加され、復号器サイクルが繰り返される。入力バッファ内のデータ(C 1 ブロック)が準備完了するとすぐに、新たなデータがメモリロケーションに書き込まれる。このように、動作中は、有効なデータが処理されるまで、4つの復号器のメモリを通しての読み出しは進行する。

#### 【0017】

CIRCの復号に使用するためのU.S.-A-4637021に従った方法では、デインターリーバの実行とデータを制御する特別の方法を使用して実行される。デインターリーバブロックはシステムメモリ(256×32バイト)内に設けられており、C 1 メモリブロック(128×32バイト)とC 2 メモリブロック(128×32バイト)よりなる。C 1 メモリブロックはロ-112から239のアドレスを含む。C 2 メモリブロックはロ-240から255のアドレスを含み、そして、ロ-0から111を含む。システムメモリは循環的である。最も低い番号のシステムメモリアドレスは最も高い番号の次である。これを図7に示す。

#### 【0018】

既知の方法に従って、第1のC 1 復号器は、32バイトのC 1 メモリブロックを処理する。C 1 復号器は最初にアドレス112でC 1 メモリブロックを読みるように配置される。各連続する復号器サイクル(C 1 とC 2 復号器両方による2パスの実行)で、C 1 復号器

は次の連続する上のアドレスで C 1 メモリブロックを読むように配置される。このように、第 2 のサイクルに対しても、C 1 メモリブロックは位置 1 1 3 に配置される。

【 0 0 1 9 】

第 1 の C 2 デコーダは 2 8 データバイトの C 2 メモリブロックを処理する。C 2 復号器は最初にアドレス 0 で C 2 メモリブロックを読むように配置される。C 2 メモリブロックを構成するそれらのバイトは、システムメモリの対角線に沿って配置される(図 7)。C 2 メモリブロックを構成するそれらのバイトは、連続する復号サイクルに変換される。

【 0 0 2 0 】

第 2 の C 1 パスの最初では、C 1 復号器は最初にシステムメモリのアドレス 2 4 0 で C 1 メモリブロックを読むように配置される。第 2 のパスの最初で、C 2 デコーダは最初にシステムメモリのアドレス 1 1 3 で C 2 メモリブロックを読むように配置される。4 パスの実行(1 つの復号器サイクル)は、システムメモリを通して連続して進む。未処理のデータがまだメモリ内にある場合には、復号サイクルは連続する。システムメモリ内の全データが訂正されると、誤り訂正システムの動作は完了する。

【 0 0 2 1 】

【発明が解決しようとする課題】

上述の C I R C 訂正コードを復号する方法は以下の欠点がある。

【 0 0 2 2 】

a ) システムメモリ内のデータは 2 5 6 サイクル(各復号サイクルで 4 復号パス)中に処理される。システムメモリ内の全データが処理訂正されたときには誤り訂正システムの動作は完了する。入力バッファに十分なデータがある場合には、システムメモリ処理内に割り込みが発生する。割り込みは、システムメモリ処理(2 5 6 サイクル)中に何回も発生し得る。従って、システムメモリ内のデータは、2 回処理できない(最初の C 1 、 C 2 パスと第 2 の C 1 、 C 2 パス)。

【 0 0 2 3 】

b ) 2 つの復号器( C 1 復号器と C 2 復号器)は、システムメモリに蓄積されたデータに同時に使用されねばならない。従って追加のハードウェアが必要である。

【 0 0 2 4 】

c ) リードソロモンコードの消失訂正方法が上述の方法と共に使用できない。

【 0 0 2 5 】

【課題を解決するための手段】

本発明の目的は、ハードウェアを減少する一方データストリーム内の誤りを効率的に訂正するクロスインターリーブドリードソロモンコード( C I R C )を復号するのに使用する方法及び装置を開示することである。

【 0 0 2 6 】

本発明の他の目的は、いわゆる消失訂正方法を使用して、データストリーム内の誤りを訂正するクロスインターリーブドリードソロモンコード( C I R C )を復号するのに使用する方法及び装置を提供することである。

【 0 0 2 7 】

本発明の特徴に従って、幾つかのロケーション  $N_{i,j}$  ( $i = 0, \dots, 217, j = 0, \dots, 31$ )と共に、メモリ内に C 1 コードワード  $C1\_CDW_k$  ( $k = 0, \dots, 108$ ) 及び C 2 コードワード  $C2\_CDW_m$  ( $m = 0, \dots, 108$ ) として蓄積されたデータ内の誤りを訂正するクロスインターリーブドリードソロモンコード( C I R C )を復号する処理内で使用する誤り訂正方法であって、前記ロケーション  $N_{i,j}$  の各々は前記データのデータバイトを含み、

a 1 ) ここで、 $k = 0, \dots, 108, j = 0, \dots, 31, i = k + 109$  及び  $C1\_CDW_k^j$  は前記 C 1 コードワード  $C1\_CDW_k$  の位置  $j$  のデータバイトであり、 $C1\_CDW_k^j = N_{i,j}$  に基づき復号手段を使用して、前記ロケーション  $N_{i,j}$  からの前記 C 1 コードワード  $C1\_CDW_k$  を読み出し且つ復号するステップと、

a 2 ) 少なくとも前記 C 1 コードワード  $C1\_CDW_k$  を部分的に訂正するステップと、

b 1 ) ここで、 $m = 0, \dots, 108$ 、 $j = 0, \dots, 28$ 、 $i = m + 4 * j$ 、及び、 $C2\_CDW_m^j$  は前記  $C2$  コードワード  $C2\_CDW_m$  の位置  $j$  のデータバイトであり、 $C2\_CDW_m^j = N_{i,j}$  に基づき復号手段を使用して、前記ロケーション  $N_{i,j}$  からの前記  $C2$  コードワード  $C2\_CDW_m$  を読み出し且つ復号するステップと、

b 2 ) 少なくとも前記  $C2$  コードワード  $C2\_CDW_m$  を部分的に訂正するステップとを有する方法が提供される。

#### 【 0028 】

この方法は、新しい方法での、 $C1$  と  $C2$  のコードワードの読み出しと復号を提供する。これは、新たな方法は、どのデータワードが  $C1$  と  $C2$  復号に使用されるかを定義し、且つ処理順序のための特定の規則を含むことを意味する。デインターリーバ実行のためのシステムメモリは、従来のメモリと比較して小さいサイズであることができる。更に、本発明に従った方法を実行するために、単一の  $CIRC$  復号器のみを使用すれば良い。

#### 【 0029 】

$C1$  コード内の大きなバースト誤りの訂正を、低入力誤り率に対して非常に効率的に実行できる。

#### 【 0030 】

本発明の他の特徴に従って、上述の誤り訂正方法を実行する新たな装置が請求項 13 に記載されている。

#### 【 0031 】

独立請求項に従った本発明の優位な特徴は、関連する独立請求項に記載されている。

#### 【 0032 】

##### 【発明の実施の形態】

###### 1. システム要素

図 2 は本発明に従った  $CIRC$  復号器のブロック図である。

#### 【 0033 】

取得部分からのデータストリームは入力論理 1 内で蓄積される。入力論理 1 は、109 の  $C1$  コードワード (32 バイト) をバッファに蓄積できる。十分なデータが入力バッファにある場合には、データブロック ( $109 \times 32$ ) はシステムメモリ 2、好ましくは RAM へ書き込まれる。システムメモリ 2 では、データを蓄積するのに 2 つのメモリブロック  $B1, B2$  が形成される。

#### 【 0034 】

”1シンボルの遅延”要素 3 は従来の  $CIRC$  復号器から知られるデータ遅延のために使用される。

#### 【 0035 】

読み出し、書き込み又は、誤り訂正動作が行われるシステムメモリ 2 の中のアドレッシングとメモリロケーションの決定のための手段は、タイミング及び制御要素 4 に含まれる。

#### 【 0036 】

より理解するために、システムメモリ 2 内のデータバイトは図 3 に示される 2 次元デカルト座標系要素であると考える。この仮定は、システムメモリ 2 の実際のハードウェア実行と独立している。

#### 【 0037 】

各垂直又は、 $i$  軸位置は 32 バイトの  $C1$  コードワード (図 3) を有する。各水又は、 $j$  軸は各  $j$  軸位置で 8 ビットバイトを表す。バイトは左から右へ移動する各  $j$  位置内で順次番号が振られる。それは、システム的なリードソロモン (RS) コードに従っている。パリティバイトは位置 28 から 31 である。

#### 【 0038 】

図 2 に従って、従来の RS 復号器 5 は、システムメモリ 2 とタイミング及び制御要素 4 に接続されている。従来の RS 復号器 5 では RS コードのいわゆる消失訂正方法が実行される。RS 復号器 5 は、4 消失又は、2 誤り又は、 $2 * t + E < d$  に従った組合せを訂正できる。ここで  $t$  はコード内の誤りの数、 $E$  は消失数、及び、 $d$  は RS コードのハミング

距離である。コンパクトオーディオディスクアプリケーションでは  $d = 4$  である。

【 0 0 3 9 】

システムメモリ 2 内のアドレッシングと位置の決定と、読出し、書きこみ、及び、訂正動作中のデータバイトの転送の制御はタイミング及び制御要素 4 を使用して行われる。

【 0 0 4 0 】

2 シンボルの遅延は ” 2 シンボル遅延 ” 要素 6 内で行われる。1 つの C 2 コードワードは、C I R C 復号器の第 2 の復号器サイクル後に訂正され、出力バッファ 7 に書きこまれる。

【 0 0 4 1 】

2. 本発明に従った誤り訂正方法の詳細な動作

図 4 を参照すると、C I R C 訂正コードの複合的な処理のための新たな復号方法の動作を詳細に説明する。十分なデータが入力バッファにある場合には、データブロック (109 × 32) は、ステップ 410 と 420 で、システムメモリ 2 の i - 位置 109, 110, . . . , 217 へ書きこまれる。

【 0 0 4 2 】

B 1 メモリブロックのカウンタバスは最初に 1 に初期化される (図 4 の 430)。そして、ステップ 440 で、条件 Count\_ps = 2 - ? が分析される。結果が NO であるならステップ 450 へ進み、。ステップ 450 では、この時点で、C 1 レベルである R S 復号器 5 は、i - 位置 109 で、C 1 コードワード (32 バイト、j - 位置に沿って j = 0, 1, . . . , 31) を読み出すために配置される。各連続する C 1 復号器サイクルで、R S 復号器 5 は前に配置されそして、連続する次の値の i - 位置で C 1 コードワードを読む。このように、第 2 の C 1 復号器サイクル中に、R S 復号器 5 は i - 位置 110 に配置される。同様に、全ての j - 位置 (j = 0, 1, . . . , 31) に対して i - 位置 217 が読み出されるまでアドレッシングは続く。

【 0 0 4 3 】

各 C 1 復号器サイクルで R S 復号器 5 は、2 誤りまでを訂正しようと試みる。3 つ又はそれ以上の誤りがある場合には、訂正されず、C 1 コードワードの未訂正のデータバイトの全てに消失フラグが割り当てられる。消失フラグはシステムメモリ 2 内の更なるメモリブロック (図示していない) に蓄積される。

【 0 0 4 4 】

一般的に、R S 復号器 5 (C 1 レベル) は連続して 109 の復号と訂正サイクルを行う (図 4 の 450)。そして、B 1 メモリブロック (図 3) は完全に処理される、即ち、C 1 復号の最初のバスは終了する。その後、この時点で C 2 レベルの R S 復号器 5 は、i - 位置 0 (図 4 の 460) で、C 2 コードワード (28 バイト) を読み出す位置にある。C 2 コードワードは B 1 と B 2 メモリブロックの対角線に沿って読み出される。読出し動作の C 2 コードワード (C 2 - CW) のアドレッシングは、以下のアルゴリズムにより行われる。

【 0 0 4 5 】

【 数 1 】

```

for(cycle=0; cycle<=108; ++cycle)
{for(j=0; j<28; ++j)
 {i=cycle+4*j;
  C2_CDW[j]=memory[i][j];
 }
}

```

ここで、[i] [j]はシステムメモリ2内のロケーションである。

【0046】

各C2復号器サイクルでは、RS復号器5は4消失又は、2誤り又は、以下の条件  $2*t + E < d$ に基づき誤り又は、消失の組合せを訂正しようと試みる。ここでtはコード内の誤りの数、Eは消失数、及び、dはRSコードのハミング距離である。

【0047】

C1復号サイクルで割り当てられた消失情報(C1ポインタ情報)は、C2復号と訂正動作で使用される。ステップ460で、未訂正C2コードワードはマークされない、即ち、消失フラグは割り当てられない。

【0048】

全体的に、RS復号器5は(C2レベルで)、連続して109のC2復号及び、訂正サイクルを行う(図4のステップ460)。109のC2コードワードが処理されそして、第1のC2復号は完了する。109のC2復号と訂正サイクルの後に、カウンターの内容Count\_ps(カウンターパス;図4の470)は増加され2となる。

【0049】

ステップ440中に再び、条件Count\_ps = 2 - ?が分析される。結果がYESであるなら、ステップ480が行われそして、B1メモリブロック内の全データバイトの消失フラグがクリアされる。そして、RS復号器5は(C1レベルで)、再び、ステップ490中に、i - 位置109で、C1コードワードを読み出すために配置される。各連続するC1復号器サイクルでは、RS復号器5は前に配置され、そして、次の連続する高いi - 位置で、C1コードワードは読みだされる。このように、第2のC1復号器サイクル中に、RS復号器5は、i - 位置110に配置される。同様に、アドレッシングはi - 位置217まで続く。各C1復号器サイクルで、RS復号器5は2つまでの誤りを訂正しようと試みる。3つ又はそれ以上の誤りがある場合には、訂正されず、C1コードワードの未訂正のデータバイトの全てに消失フラグが割り当てられる。

【0050】

一般的に、RS復号器5(C1レベル)は連続して109の復号と訂正サイクルを行う(図4の490)。そして、B1メモリブロック(図3)は完全に処理される、即ち、第2のC1復号のパスは終了する。その後、(C2レベルで)RS復号器5は、i - 位置0で、C2コードワード(28バイト)を読み出すために再び配置される。ステップ500で、C2コードワードはB1とB2メモリブロックの対角線に沿って読み出される。読み出し動作のC2コードワード(C2-CW)のアドレッシングは、前述のアルゴリズム(1)により行われる。

【0051】

RS復号器5は再び、各C2復号器サイクルにより、4消失又は、2誤り又は誤り又は消失の組合せを訂正しようと試みる。RS復号器5が現在のC2コードワードを訂正できない場合には、データバイトにフラグが付される。その後、処理されたC2コードワー

ドは出力バッファ7へ書きこまれる(図2)。

【0052】

RS復号器(C2レベル)は、連続して、109のC2復号と訂正サイクルを行う。109のC2コードワードが処理された後、C2復号の第2のパスは完了する。そして、消失情報を伴なうB1メモリブロックは、システムメモリ2のB2メモリブロックへ移動される(図4のステップ510)。タイミング及び制御要素4で、信号DONEが発生される。この信号は、システムメモリ2内の全てのデータバイトが丁度2回処理されたという情報を与える。そしてステップ410は再び実行される。

【0053】

3. バースト誤り訂正

以下に本発明に従った方法を使用するバースト誤り訂正を説明する。

【0054】

A) 高入力誤り率でのC1コードワードの訂正(C1コードワードは完全に又は、ほぼ完全に壊れている)

本発明に従ったCIRC訂正コードの訂正方法の詳細な動作は、バースト誤りと関連して説明される。(前の段階で) B2メモリブロックは既に訂正されかつB2メモリブロックには、誤りがないと仮定する。B1メモリブロックが入力論理1から(により)入力される。i-位置130, . . . , 140に、バースト誤りがある(11の破壊されたC1コードワード、図5参照)。

【0055】

RS復号器5(C1レベル)は開始し、B1メモリブロック内の109のC1コードワードを連続して復号しつつ訂正することを試みる。i-位置130, . . . , 140でコードワードは、訂正不能でありそしてそれゆえ、フラグが付される(各C1コードワード内の各バイトに消失フラグが割り当てられる)。

【0056】

そして、RS復号器5(C2レベル)は、開始しそして、B1とB2メモリブロック内の109のC1コードワードを連続して復号しつつ訂正することを試みる。その動作中に、RS復号器5(C2レベル)は、メモリセクタ2B1と1B2(図5参照)からバイトを読まなければならない。RS復号器5の第1パスの完了後(109のC1パスと109のC2パス)、メモリセクタ2B1のi-位置(130-140)でフラグが付されたバイトは訂正され、一方、メモリセクタ1B1のi-位置(130-140)でフラグが付されたバイトは訂正されずに残る。

【0057】

RS復号器5の第2のパスは、B1メモリブロック内の109のC1コードワードを復号し連続して訂正しようと試みる。位置130, . . . , 140でコードワードは、再びフラグを付されるがしかし、実際には、この時点で誤りはメモリセクタ1B1内にのみ存在する。そして、RS復号器5(C2レベル)は開始し、B1とB2メモリブロック内の109のC2コードワードを連続して復号しつつ訂正することを試みる。

【0058】

次の段階で、消失情報を伴なう訂正されたB1メモリブロックは、システムメモリのB2メモリブロックにコピーされる(図6参照)。その後、新たなデータブロックが入力論理1からB1メモリブロックへロードされ、そして、CIRC復号器が再度開始する。残りの誤りは、(消失情報を伴なうバイト)メモリセクタ1B2内に存在する。RS復号器5(C1レベル)は開始し、そして、B1メモリブロック内の109のC1コードワードを連続して復号しつつ訂正することを試みる。

【0059】

そして、RS復号器5(C2レベル)は、開始し、B1とB2メモリブロック内の109のC2コードワードを連続して復号しつつ訂正することを試みる。この段階では、誤りはメモリセクタ1B2内にのみある誤り(i-位置21, . . . , 31に残る誤り)は、訂正される。

**【 0 0 6 0 】**

本発明に従った方法を実行する C I R C 復号器は、 14 の壊れた C 1 コードワード（限界）を高入力誤り率で訂正可能である。

**【 0 0 6 1 】**

B ) 低入力誤り率を伴なう C 1 コードワードバーストの訂正（各 C 1 コードワードは 3 から 6 の誤りを有する）

本発明に従った方法を実行する C I R C 復号器は、破壊された C 1 コードワードが低入力誤り率を有する場合には、非常に効率的に動作する。以下にこれを説明する。

**【 0 0 6 2 】**

C 2 コードワード誤り訂正（C I R C 復号器の第 1 のパス）では、R S 復号器 5（C 2 レベル）は消失のどのようなランダムの組み合わせ（4 消失まで）も訂正できる、即ち、C I R C 復号器の第 2 のパス中は、R S 復号器 5（C 1 レベル）は、C I R C 復号器（C 1 レベル）の第 1 のパス中で訂正されなかつどのような C 1 コードワードも訂正することができる。従って、C I R C 復号器の第 2 のパス中に、C I R C 復号器（C 2 レベル）の第 1 のパス中で訂正されなかつどのような C 2 コードワードも訂正できる。

**【 0 0 6 3 】**

これは、図 7 を参照して示すことができる。図 7 では、それぞれ、従来の C I R C 復号器と本発明に従った方法を実行する C I R C 復号器の、2 つの C I R C 復号器の誤り訂正結果が表示されている。従来の C I R C 復号器の結果は、破線で示されている。本発明に従った方法を実行する C I R C 復号器の結果は、実線で示されている。

**【 0 0 6 4 】**

結果は、以下の組合せの誤りに対して示されている。

- 1 . 各 C 1 コードワード内に 3 つの誤り（ランダム誤り）を有する、20 の C 1 コードワードのバースト誤り。
- 2 . 各 C 1 コードワード内に 4 つの誤りを有する、20 の C 1 コードワードのバースト誤り。
- 3 . 各 C 1 コードワード内に 5 つの誤りを有する、20 の C 1 コードワードのバースト誤り。
- 4 . 各 C 1 コードワード内に 6 つの誤りを有する、20 の C 1 コードワードのバースト誤り。
- 5 . 各 C 1 コードワード内に 3 つの誤りを有する、30 の C 1 コードワードのバースト誤り。
- 6 . 各 C 1 コードワード内に 4 つの誤りを有する、30 の C 1 コードワードのバースト誤り。
- 7 . 各 C 1 コードワード内に 5 つの誤りを有する、30 の C 1 コードワードのバースト誤り。
- 8 . 各 C 1 コードワード内に 6 つの誤りを有する、30 の C 1 コードワードのバースト誤り。
- 9 . 各 C 1 コードワード内に 3 つの誤りを有する、40 の C 1 コードワードのバースト誤り。
- 10 . 各 C 1 コードワード内に 4 つの誤りを有する、40 の C 1 コードワードのバースト誤り。
- 11 . 各 C 1 コードワード内に 5 つの誤りを有する、40 の C 1 コードワードのバースト誤り。
- 12 . 各 C 1 コードワード内に 6 つの誤りを有する、40 の C 1 コードワードのバースト誤り。

**【 0 0 6 5 】**

各 C 1 コードワード内に異なる数の誤りを有する、20 の C 1 コードワードのバースト誤りの訂正結果を、図 7 に示す（円）。本発明に従った方法を実行する C I R C 復号器は、各コードワード内に 3 又は 4 つの誤りを有する 20 の C 1 コードワードのバースト誤り

訂正できる。C 1 コードワード内に 5 又は 6 つの誤りが発生した場合には、本発明に従った方法を実行する C I R C 復号器は、20 の C 1 コードワードのバースト誤りを訂正できない。しかし、本発明に従った方法を実行するオーディオデータ C I R C 復号器の出力消失率は非常に低い。

#### 【 0 0 6 6 】

本発明に従った方法を実行する C I R C 復号器に関しては、従来の C I R C 復号器と比較して、30 の C 1 コードワードのバースト誤りの訂正に関して、誤り訂正のより良好な性能が得られる。本発明に従った方法を実行する C I R C 復号器は、各コードワード内に 3 つの誤りを有する、30 の C 1 コードワードのバースト誤りを訂正できる。本発明に従った方法を実行する C I R C 復号器内のオーディオデータの出力消失率はも低い（結果は四角により示されている）。本発明に従った方法を実行する C I R C 復号器の優位点は、更なる増大する入力誤り率で減少することである。

#### 【 0 0 6 7 】

従来の C I R C 復号器と本発明に従った方法を実行する C I R C 復号器の両方に対する制限は、バースト誤り 14 C 1（完全に壊れた）コードワードである。この場合、従来の C I R C 復号器と本発明に従った方法を実行する C I R C 復号器に対する誤り訂正能力は異なる。

#### 【 0 0 6 8 】

##### 【発明の効果】

本発明により、ハードウェアを減少する一方データストリーム内の誤りを効率的に訂正するクロスインターリーブドリードソロモンコード（C I R C）を復号するのに使用する方法及び装置を提供できる。

##### 【図面の簡単な説明】

###### 【図 1】

従来の誤り訂正方法と共に使用されるシステムメモの例を示す図である。

###### 【図 2】

本発明に従った方法を実行するのに好適な C I R C 復号器のブロック図である。

###### 【図 3】

本発明に従ったメモリブロック B 1 と B 2 を伴なうメモリブロックの概略を示す図である。

###### 【図 4】

本発明に従った方法のフローチャートを示す図である。

###### 【図 5】

メモリセクタを伴なう図 3 のメモリブロックを示す図である。

###### 【図 6】

メモリブロック B 2 へ移動されたメモリブロックを伴なう図 5 のメモリブロックを示す図である。

###### 【図 7】

それぞれ、従来の誤り訂正方法と本発明に従った誤り訂正方法の比較結果を示す図である。

##### 【符号の説明】

- 1 入力論理
- 2 システムメモリ
- 3 1 シンボル遅延
- 4 タイミング及び制御要素
- 5 R S 復号器
- 6 2 シンボル遅延
- 7 出力バッファ