

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-80762

(P2004-80762A)

(43) 公開日 平成16年3月11日(2004.3.11)

(51) Int.Cl.<sup>7</sup>

H03M 13/27

G06F 11/10

G11B 20/18

H03M 13/15

F I

H03M 13/27

G06F 11/10 330P

G11B 20/18 512C

G11B 20/18 520E

G11B 20/18 532E

テーマコード (参考)

5B001

5J065

審査請求 未請求 請求項の数 15 O L (全 17 頁) 最終頁に続く

(21) 出願番号 特願2003-198618 (P2003-198618)

(22) 出願日 平成15年7月17日 (2003.7.17)

(31) 優先権主張番号 02090289.6

(32) 優先日 平成14年8月10日 (2002.8.10)

(33) 優先権主張国 欧州特許庁 (EP)

(71) 出願人 501263810

トムソン ライセンシング ソシエテ ア  
ノニムThomson Licensing S  
. A.フランス国, エフ-92100 ブロー  
ニュ ビヤンクール, ケ アルフォンス  
ル ガロ, 46番地

(74) 代理人 100070150

弁理士 伊東 忠彦

(74) 代理人 100091214

弁理士 大貫 進介

(74) 代理人 100107766

弁理士 伊東 忠重

最終頁に続く

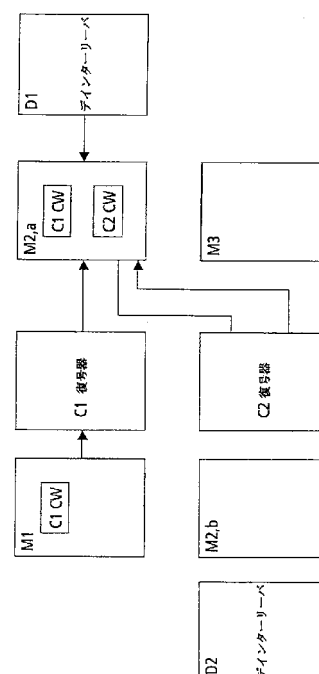
(54) 【発明の名称】 クロスインターリーブリード-ソロモン符号訂正を行う方法及び装置

(57) 【要約】

【課題】本発明は、クロスインターリーブリード-ソロモン符号訂正のための改善された方法及を提供することを目的とする。

【解決手段】本発明は、第1のC1符号語を、第1のメモリ手段からC1復号器手段へ入力するステップと、第2のC1符号語を、C1復号器手段から第2のメモリ手段へ出力するステップと、第2のメモリ手段内に第1のC2符号語を発生するために、第2のC1符号語をデインターリーブするステップと、を有する、クロスインターリーブリード-ソロモン符号訂正を行う方法に関連する。

【選択図】 図2



## 【特許請求の範囲】

## 【請求項 1】

- 第 1 の C 1 符号語を、第 1 のメモリ手段から C 1 復号器手段へ入力するステップと、  
- 第 2 の C 1 符号語を、C 1 復号器手段から第 2 のメモリ手段へ出力するステップと、  
- 第 2 のメモリ手段内に第 1 の C 2 符号語を発生するために、第 2 の C 1 符号語をデインターリーブするステップと、  
を有する、クロスインターリーブリード - ソロモン符号訂正を行う方法。

## 【請求項 2】

第 1 の C 1 符号語が訂正可能である場合には、第 2 の C 1 符号語は訂正された第 1 の C 1 符号語であり、そして、第 1 の C 1 符号語が訂正可能でない場合には、第 2 の C 1 符号語は第 1 の C 1 符号語のコピーである、請求項 1 に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。 10

## 【請求項 3】

第 1 の C 1 符号語が訂正可能でない場合には、第 2 の C 1 符号語の各シンボルに対して消失フラグが設定される、請求項 2 に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。

## 【請求項 4】

更に、

- 第 1 の C 2 符号語を、第 2 のメモリ手段から C 2 復号器手段へ入力するステップと、  
- 第 2 の C 2 符号語を、C 2 復号器手段から第 2 のメモリ手段へ出力するステップと、  
を有する、請求項 1 乃至 3 のうちいずれか一項に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。 20

## 【請求項 5】

第 1 の C 2 符号語が訂正可能である場合には、第 2 の C 2 符号語は訂正された第 1 の C 2 符号語であり、そして、第 1 の C 2 符号語が訂正可能でない場合には、第 2 の C 2 符号語は第 1 の C 2 符号語のコピーである、請求項 4 に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。

## 【請求項 6】

更に、

第 1 の C 2 符号語が訂正可能でない場合には、第 2 の C 2 符号語の各シンボルに対して消失フラグを設定することを含む、請求項 5 に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。 30

## 【請求項 7】

更に、

- 第 2 の C 1 符号語を、第 2 のメモリ手段から C 1 復号器手段へ入力するステップと、  
- 第 3 の C 1 符号語を、C 1 復号器手段から第 3 のメモリ手段へ出力するステップと、  
- 第 3 のメモリ手段内に第 3 の C 2 符号語を発生するために、第 3 の C 1 符号語をデインターリーブするステップと、  
を有する、請求項 1 乃至 6 のうちいずれか一項に記載のクロスインターリーブリード - ソロモン符号訂正を行う方法。 40

## 【請求項 8】

第 2 の C 1 符号語が訂正可能である場合には、第 3 の C 1 符号語は訂正された第 2 の C 1 符号語であり、そして、第 2 の C 1 符号語が訂正可能でない場合には、第 3 の C 1 符号語は第 2 の C 1 符号語のコピーである、請求項 7 に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。

## 【請求項 9】

更に、

第 2 の C 1 符号語が訂正可能でない場合には、第 3 の C 1 符号語の各シンボルに対して消失フラグを設定することを含む、請求項 8 に記載の、クロスインターリーブリード - ソロモン符号訂正を行う方法。 50

**【請求項 10】**

更に、

- 第3のC2符号語を、第3のメモリ手段からC2復号器へ入力するステップと、
  - 第4のC2符号語を、C2復号器手段から第4のメモリ手段へ出力するステップと、
- を有する、請求項7乃至9のうちいずれか一項に記載の、クロスインターリーブリード・ソロモン符号訂正を行う方法。

**【請求項 11】**

第3のC2符号語が訂正可能である場合には、第4のC2符号語は訂正された第3のC2符号語であり、そして、第3のC2符号語が訂正可能でない場合には、第4のC2符号語は第3のC2符号語のコピーである、請求項10に記載の、クロスインターリーブリード・ソロモン符号訂正を行う方法。

10

**【請求項 12】**

更に、

第3のC2符号語が訂正可能でない場合には、第4のC2符号語の各シンボルに対して消失フラグを設定することを含む、請求項11に記載の、クロスインターリーブリード・ソロモン符号訂正を行う方法。

**【請求項 13】**

請求項1乃至12のうちいずれか一項に記載のクロスインターリーブリード・ソロモン符号訂正を行う方法を実行するプログラム手段を有するコンピュータプログラムプロダクト。

20

**【請求項 14】**

請求項1乃至12のうちいずれか一項に記載のクロスインターリーブリード・ソロモン符号訂正を行う方法に従った、クロスインターリーブリード・ソロモン符号訂正を行う手段を有する、リード・ソロモン復号器。

**【請求項 15】**

請求項14に記載のリード・ソロモン復号器を有する、例えば、CD又はDVD再生装置又は記録装置の、オーディオ又はビデオ装置。

**【発明の詳細な説明】****【0001】****【発明の属する技術分野】**

30

本発明は、一般的には、リード・ソロモン符号を使用してディジタルデータに誤り訂正を実行する方法及び装置に関連し、特に、クロスインターリーブリード・ソロモン符号訂正に関連する。

**【0002】****【従来の技術】**

コンパクトオーディオディスクに典型的に使用されている1つの誤り訂正符号は、いわゆるCIRC訂正符号である。CIRCはクロスインターリーブリード・ソロモン符号の頭文字である。CIRC訂正符号の場合には、(28, 24)リード・ソロモン符号(C2符号)の符号化処理は、各シンボルが8ビットよりなる24データシンボルに対して実行される。そのような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ビットであり、これは各8ビットの2つのシンボルより構成される。従って、各8ビットのシンボルは、ステレオオーディオデータの2つのチャネルの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。

40

**【0003】**

これらのデータシンボルは、第1の状態の配置であり、次に、データは、第1の配置状態から第2の配置状態へ、インターリーブ動作により再配置される。(32, 28)リード・ソロモン符号(C1符号)の符号化処理は、第2の配置状態である28シンボルに対して実行される。CIRC訂正符号を復号するときには、最初にC1復号が行われ、そして、デインターリーブが実行され、そして、その後、C2復号が実行される。

**【0004】**

50

C I R C 訂正符号を復号する種々の方法が既に知られており、そのような方法の例は、例えば、特許文献 1、特許文献 2、及び、特許文献 3 に開示されている。他方では、リード - ソロモン符号を復号する種々の方法が既に知られており、そのような方法の例は、特許文献 4 に開示されている。

【 0 0 0 5 】

C I R C 訂正符号の従来 of 復号方法に従って、第 1 段階で 2 重誤り訂正まで且つそれを含む誤り処理が C 1 復号で実行され、そして、C 1 復号から得られたポインタ情報を参照して次の段階で、2 重誤り訂正が C 2 復号で実行される。

【 0 0 0 6 】

提案されている誤り訂正符号を復号する 1 つの方法は、いわゆる消失訂正法であり、そこ  
10  
では、誤りシンボルの位置がポインタ情報により示されそして、この誤りシンボルに必要な訂正が行われる。上述の C 1 と C 2 符号の場合には、2 重誤り ( 2 シンボル ) まで及びそれを含む検出と訂正は、それぞれ実行されうる。しかしながら、誤り位置が既に知られている場合には、4 重誤り ( 4 シンボル ) まで及びそれを含む誤り訂正が実行されることが可能である。

【 0 0 0 7 】

従って、誤り訂正能力を上げるために、誤りを復号する消失訂正法を実行することが好ま  
しかった。更に、消失訂正法は、バースト誤りを訂正するのに特に効果的であることがわ  
かった。他方では、消失訂正動作を適切に実行するために、誤り位置がポインタ情報から  
予備的に知られていなければならない、そして、さらにポインタ情報の信頼性が比較的高く  
20  
なければならない。

【 0 0 0 8 】

C I R C 訂正符号の従来 of 符号化方法に従って、2 重誤りまで及びそれを含む誤り訂正が  
C 1 復号器で実行される。そのような場合には、訂正できない、3 重誤りの発生のおそれ  
があるので、C 1 ポインタは次の段階の C 2 復号器へ送られ、それにより、誤り訂正が C  
1 ポインタを使用して C 2 復号器で実行される。

【 0 0 0 9 】

特に、従来 of C I R C 訂正符号の場合には、C 1 符号の系列 ( C 1 系列 ) が、交互に 2 つ  
の隣接するフレーム ( 1 フレーム : 3 2 シンボル ) に含まれる、3 2 シンボルにより構成  
される。C 2 符号の系列 ( C 2 系列 ) は、1 0 8 の連続するフレームの予め定められたも  
30  
のに含まれる 2 8 シンボルより構成される。C 1 系列のインターリーブ長は、C 2 系列の  
インターリーブ長よりも短いので、キューやリビュウのような早送り再生動作が行われる  
時には、幾つかのフレームが落とされそして、フレームの連続性が失われるという問題が  
発生する。即ち、C 1 ポインタが、不連続の点の前後の 1 フレームで誤りの存在を示すの  
に使用されるが、しかし、C 1 ポインタは、他のフレーム内の誤りの不存在を示すだけで  
ある。他方では、C 2 系列のインターリーブ長は 1 0 8 フレームを有し、そして、これら  
の 1 0 8 フレームは不連続の点を含みうる。従って、これも正しい C 2 系列ではない。前  
述の C 1 ポインタを使用して、不正確な C 2 系列に対して消失訂正を実行すると、誤り訂  
正が不正確となる。

【 0 0 1 0 】

消失訂正法により最大の誤り訂正能力が得られる、C I R C 訂正符号の誤り訂正を実行す  
るときに、不完全な又は不正確な誤り訂正のこの問題を解決するために、特定の順序で C  
1 復号と C 2 復号が 2 回実行される誤り訂正法が、提案されている ( そのような C 1 復号  
と C 2 復号の 2 回の実行は、特許文献 5 に示されている ) 。その順序は、例えば、C 1 復  
号、それに続き C 2 復号、それに続き C 1 復号、そして、それに続き C 2 復号である。最  
初の C 2 復号では、前の段階での C 1 復号により得られた C 1 ポインタを使用して消失訂  
正が実行され、そして、この消失訂正で発生し得る誤りのある訂正は、第 2 の C 1 復号と  
C 2 復号により防止され、これらは、C I R C 訂正符号を復号する従来 of 方法に似ている  
。  
50

【 0 0 1 1 】

10

20

30

40

50

特許文献 6 は、リード - ソロモン符号を使用する誤り訂正法を示す。誤り訂正は、シンδροームの乗算の結果と誤り位置を使用して実行され、他のシンδροームは、それにより新たなシンδροームを形成するために加えられる。この手順を繰返し実行することにより、幾つかの算術動作により誤りベクトルが得られ、それにより、いわゆる消失訂正技術により誤り訂正を実行する。そして、第 1 の計算された誤りベクトルを使用して他の誤りベクトルを計算することにより、算術動作の回数が減少されうる。

#### 【 0 0 1 2 】

図 1 は、リード - ソロモン ( R S ) 符号の、従来のクロスインターリーブリード - ソロモン符号 ( C I R C ) 復号を示すブロック図である。C I R C 復号器の主なタスクは、

- 1 . 1 シンボルだけ入力データを遅延させる ;
- 2 . C 1 符号 ( 3 2 , 2 8 ) を復号しそして、誤りのあるシンボルを訂正する ;
- 3 . デインターリーブする ;
- 4 . C 2 符号 ( 2 8 , 2 4 ) を復号しそして、誤りのあるシンボルを訂正する ;
- 5 . 2 シンボルの遅延を行う ;

ことを実行することである。

#### 【 0 0 1 3 】

1 フレームを構成する 3 2 シンボルの入力データは、時間遅延回路 ( 1 シンボルの遅延 ) に供給され、そこで、1 フレームだけ、偶数番号のシンボルのみが、遅延される。時間遅延回路から出力される 3 2 シンボルは、C 1 復号器に供給され、ここで、( 3 2 , 2 8 ) リード - ソロモン符号 ( C 1 符号語 ) の実際の復号が実行される。C 1 符号語内の 2 誤りシンボルまでの誤り訂正が、C 1 復号器内で実行される。C 1 復号器内で 3 又はそれ以上の誤りが検出されるときには、C 1 ポインタ ( 消失フラグ ) が、C 1 符号語の全シンボルに設定される。データと消失フラグは、更に、デインターリーブ回路 ( 等しくない長さの遅延線 ) で処理される。デインターリーブ回路の出力は、C 2 復号器に供給される。デインターリーブ回路では、C 1 復号器から発生された各シンボルの消失フラグが、データのデインターリーブと同様な方法でデインターリーブされる。4 誤りまでの消失訂正は、C 1 消失フラグを使用して、C 2 復号器内で実行される。C 2 復号器により、4 誤りシンボルまでの訂正が可能な場合には、C 1 消失フラグはクリアされる。可能でない場合には、C 2 符号語内の C 1 消失フラグはコピーされるか又は、C 2 符号語の全シンボルへ消失フラグが設定されるかのいずれかである。C I R C 復号の最後の段階は、2 シンボル遅延である。これは、2 シンボル遅延回路により実行される。C I R C 訂正符号を 2 回使用することは、誤り訂正システムの訂正能力の増加を導く。

#### 【 0 0 1 4 】

##### 【 特許文献 1 】

米国特許番号 U S - A - 4 , 5 4 6 , 4 7 4 号

##### 【 特許文献 2 】

米国特許番号 U S - A - 4 , 4 7 6 , 5 6 2 号

##### 【 特許文献 3 】

米国特許番号 U S - A - 4 , 4 9 7 , 0 5 8 号

##### 【 特許文献 4 】

米国特許番号 U S - A - 4 , 4 7 6 , 5 6 2 号

##### 【 特許文献 5 】

米国特許番号 U S - A - 4 , 6 3 7 , 0 2 1 号

##### 【 特許文献 6 】

E P - A - 0 , 2 7 8 , 3 8 3

##### 【 発明が解決しようとする課題 】

しかしながら、図 1 に示されたような形式の従来の C I R C 復号器の共通の欠点は、C I R C 符号のブロックの 2 回の処理ができないことである。

#### 【 0 0 1 5 】

従って、本発明の目的は、C I R C 訂正のための改善された方法及び、C I R C 訂正のた

10

20

30

40

50

めの、改善されたコンピュータプログラム及びリード - ソロモン復号器を提供することである。

【 0 0 1 6 】

【課題を解決するための手段】

本発明は、請求項 1 に従った、C I R C 訂正の改善された方法を提供する。更に、本発明は、更なる独立請求項に従った、改善されたコンピュータプログラム及びリード - ソロモン復号器を提供する。本発明の好ましい実施例は、独立請求項に記載されている。

【 0 0 1 7 】

本発明は、C I R C 復号器の単一パス及び 2 回のパスの動作を可能とする。1 シンボル遅延動作、2 シンボル遅延動作及び、C I R C 訂正符号の 2 回の又は単一の復号は、4 つの別々のメモリを使用することにより、そして、各メモリの特別の制御及びアドレッシングにより、達成される。 10

【 0 0 1 8 】

本発明は、C I R C 符号のブロックを 2 回の処理することを可能とすることに、特に優位である。

【 0 0 1 9 】

【発明の実施の形態】

本発明の例示的な実施例を、添付の図面を参照して説明する。

【 0 0 2 0 】

図 2 は、C I R C 復号器のブロック図を示す。この復号器は、C 1 符号語 (C W) を格納するメモリ M 1 を有する。更に、C 1 復号器、メモリ M 2 , a と関連するデインターリーバ D 1、メモリ M 2 , b と関連するデインターリーバ D 2、C 2 復号器及び、メモリ M 3 がある。 20

【 0 0 2 1 】

第 1 のパスの最初では、C 1 C W はメモリ M 1 に格納される。そこから、C 1 C W は C 1 復号器に入力される。C 1 復号器は、他の C 1 C W を、M 2 , a に出力する。C 1 復号器から出力された C 1 C W は、メモリ M 1 の C 1 C W が訂正可能でない場合には、メモリ M 1 の C 1 C W のコピーである。この場合には、C 1 C W 内の全シンボルについて、消失フラグ設定される。メモリ M 1 の C 1 C W が訂正可能な場合には、C 1 復号器から出力された C 1 C W は、メモリ M 1 の訂正された C 1 C W である。デインターリーブ動作は、デインターリーバ D 1 により実行される。この結果、C 2 C W は、メモリ M 2 , a に格納される。メモリ M 2 , a の C 2 C W は、C 2 復号器へ入力される。C 2 復号器は C 2 C W をメモリ M 2 , a へ書き戻す。この C 2 C W は、元の C 2 C W が訂正可能でない場合には元の C 2 C W のコピーであり、又は、訂正された C 2 C W であるかのいずれかである。C 2 C W が訂正可能でない場合には、再び消失フラグが設定される。 30

【 0 0 2 2 】

図 3 は、第 2 のパスを示す。M 2 , a の C 1 C W は、C 1 復号器に入力される。C 1 復号器は C 1 C W をメモリ M 2 , b へ出力する。メモリ M 2 , b の C 1 C W は、メモリ M 2 , a の C 1 C W が訂正可能でない場合にはメモリ M 2 , a の C 1 C W のコピーであり、又は、メモリ M 2 , a の訂正された C 1 C W であるかのいずれかである。メモリ M 2 , a の C 1 C W が訂正可能でない場合には、C 1 復号器から出力されそしてメモリ M 2 , b に格納される符号語 C 1 C W に、消失フラグが、設定される。デインターリーバ D 2 が開始する。この結果、符号語 C 2 C W がメモリ M 2 , b に供給される。これは、C 2 復号器に入力される。C 2 復号器は、符号語 C 2 C W をメモリ M 3 に出力する。C 2 復号器により出力される C 2 C W は、メモリ M 2 , b の C 2 C W が訂正可能でない場合にはメモリ M 2 , b の C 2 C W のコピーであり又は、訂正されたメモリ M 2 , b の C 2 C W であるかのいずれかである。メモリ M 2 , b の C 2 C W が訂正可能でない場合には、消失フラグが設定される。 40

【 0 0 2 3 】

以下に更に詳細な実施例を説明する。

## 【0024】

## システム構成要素

図4は、CIRC復号器の主な構成要素を示すCIRC復号器のブロック図を示す(2回の及び単一のパスのCIRC復号器がこの構成で実行される)。取得部からのデータストリームは、入力論理1に入力される。入力論理は、32シンボルを格納できる。入力論理のバッファに十分なデータが利用できる場合には、フレームがM1メモリに格納される。M1メモリは、CIRC復号器の1シンボル遅延タスクのために使用される。MUX1マルチプレクサ3は、CIRC復号器のパスに依存して、M1メモリからC1復号器へ又はM2, aメモリからC1復号器へのいずれかからのデータを多重するのに使用される。

## 【0025】

ブロック4は、従来のRS復号器である。このブロックでは、いわゆるRS符号の消失訂正法が実行される。RS復号器4は、4消失又は2誤り又は、次の条件： $2 * t + E < d$ の下での誤りと消失の組合せを訂正でき、ここで、 $t$  - は符号語内の誤りの数、 $E$  - は消失数、 $d$  - はRS符号のハミングの距離(コンパクトオーディオディスク応用については $d = 5$ )である。フラグプロセッサブロックN0.1(ブロック5)は、C1復号器の誤り状態を発生しそして、出力消失フラグを定義する。デマルチプレクサ(DEMUX1)6は、C1復号器の出力を、M2, aメモリへ又は、レジスタ(buf\_reg)8への何れかへ、分離するために使用される。デマルチプレクサ6の出力は、CIRC復号器のパスに依存する。M2, aメモリ7は、CIRC復号器の第1のパス中に、デインターリーブしそしてC1符号語をC2符号語に集めるために使用される。レジスタ8は、訂正されたC1符号語を記憶するために使用される。M2, bメモリ9は、CIRC復号器の第2のパス中に、デインターリーブしそしてC1符号語をC2符号語に集めるために使用される。CIRC復号器のコントローラ10は、以下のタスク中にメモリ管理を実行する。

- a) 1シンボル遅延タスク；
- b) C1復号器タスク；
- c) デインターリーブ及びC2復号器タスク；
- d) 2シンボル遅延タスク。

## 【0026】

マルチプレクサ11(MUX2)は、CIRC復号器のパスに依存して、データを、M2, aメモリからC2復号器へ又は、M2, bメモリからC2復号器へのいずれかへ、多重するのに使用される。ブロック12は、従来のRS復号器を示す。この復号器は、RS復号器4と同じ性能を有する。フラグプロセッサブロックN0.2(ブロック13)は、C2復号器の誤り状態を発生しそして、出力消失フラグを定義する。M3メモリ14は、CIRC復号器の2シンボル遅延タスクのために使用される。出力論理15は、M3メモリから24シンボルを格納するのに使用される。

## 【0027】

## CIRC復号器内のデータの制御

CIRC訂正符号の2回の及び単一の処理のCIRC復号器の動作が、図5のフロー図に示されている。next\_frame信号が活性化されている場合には(ステップ2)、フレームが入力論理から対応するM1メモリのバンクへロードされ、そして、1シンボル遅延タスクが実行される(ステップ3)。ステップ4中に、データは、M1メモリ(図4参照)からMUX1マルチプレクサを通してC1復号器へ転送される。同時に、シンδροームが計算される。32シンボルを転送した後に、復号処理C1が開始される。デインターリーブタスク(ステップ5)が、M2, aメモリ内で実行される。28シンボルのC2符号語は、M2, aメモリからMUX2マルチプレクサを通してC2復号器へ転送される。同時に、シンδροームが計算される。28シンボルを転送した後に、復号処理C2が開始される。C1、C2処理の開始後に、コントローラは、イーブンの - C1処理の準備完了を待つ(ステップ6)。C1処理が準備完了の場合には、C1復号器からのC1符号語はM2, aメモリへ書きこまれる(ステップ7)。コントローラは、イーブンの - C2処理の準備完了を待つ(ステップ7)。データフローの次のステップは、CIRC復号器の単

10

20

30

40

50

一の又は2回のパスの動作に依存する(ステップ9)。C I R C復号器が、単一のパスを実行し、そして、C 2処理が準備完了している場合には、C 2復号器からの訂正されたC 2符号語は、M 3メモリへ書きこまれる。そして、2シンボルの遅延タスクが、M 3メモリへ実行される。その後に、対応するデータ項目が、出力論理へロードされる(ステップ16)。C I R C復号器が、2回のパスを実行しそしてC 2処理の準備が完了している場合には、C 2復号器からの訂正されたC 2符号語がM 2, aメモリに書き込まれ(ステップ10)、そして、C I R C復号器の第2のパスが開始される。C 1符号語はM 2, aメモリからM U X 1マルチプレクサを通してC 1復号器へ転送される。同時にシンドロームが計算される。復号処理C 1は、32シンボルの転送後に開始される(ステップ11)。デインターリーバタスク(ステップ12)は、M 2, bメモリ内で実行される。28シンボルのC 2符号語が、M 2, bメモリからM U X 2マルチプレクサを通してC 2復号器へ転送される。同時に、シンドロームが計算される。復号処理C 2は、28シンボルの転送後に開始される。

10

20

30

40

50

#### 【0028】

C 1、C 2処理の開始後に、コントローラは、イーブンの - C 1処理の準備完了を待つ(ステップ13)。C 1処理が準備完了の場合には、C 1復号器からの訂正されたC 1符号語は、b u f \_ r e gに書きこまれる(ステップ14)。C 2処理が準備完了である場合には、C 2復号器からの訂正されたC 2符号語が、M 3メモリに書きこまれる(ステップ16)。そして、2シンボル遅延タスクが、M 3メモリに実行される。その後に、対応するデータ項目が、出力論理へロードされる(ステップ16)。ステップ18中に、カウンタが更新される。ステップ18に続き、E C Cブロックが、入力論理からの新たなフレームを処理する準備が完了している。

#### 【0029】

C I R C復号器内のデータのアドレッシング

1シンボル遅延、デインターリーピング及び、2シンボル遅延動作中は、異なるメモリ内のアドレッシングと位置の決定及び、読み出し及び書きこみ動作での転送されたデータバイトの制御は、コントローラ10を使用して実行される。

#### 【0030】

1シンボル遅延タスク

M 1メモリは、C I R C復号器の1シンボル遅延タスクを実行するのに使用され、そして、アドレス0, 1及び、2を有する3バンクを有する。各バンクは32シンボルより構成される。各シンボルは、9バイトを含む。M Sビットは"消失フラグ"であり、(7:0)ビットは、データである。消失ビットは、シンボルが壊れていることを示す。十分なデータが入力論理内のバッファ内で有効である場合には、1フレームがM 1メモリの対応するバンクに格納され、そして、1シンボル遅延動作が実行される。図6は、M 1メモリ内のデータフローを示す。M 1メモリは、循環の性質を有する。M 1メモリ内のデータフローは、3 n e x t \_ f r a m e信号後に繰返される、即ち、図6aと図6dは同一である。

#### 【0031】

図7は、M 1メモリ内のアドレス計算を示す。入力論理からのフレームは、M 1メモリの対応するバンクへロードされる。バンクアドレスは、次の式に従って、計算される：

b a n k \_ a d d \_ m 1 \_ w = c o u n t \_ m 1 \_ w、

ここで、c o u n t \_ m 1 \_ wカウンタは、書きこみ動作中のM 1メモリ内のバンクアドレスを定義する。32シンボルが、バンクの対応するシンボルアドレスにロードされる。c o u n t \_ b \_ m 1カウンタは、バンクへのシンボルアドレスを定義する。メモリM 1を読み出すためのバンクアドレスは偶数又は奇数シンボルに依存する。

#### 【0032】

バンクアドレスは次の式に従って、計算される：

b a n k \_ a d d \_ 0 \_ m 1 \_ r = c o u n t \_ m 1 \_ r、 / / 偶数

b a n k \_ a d d \_ 1 \_ m 1 \_ r = ( c o u n t \_ m 1 \_ r + 2 ) % 3 ) / / 奇数。



## 【 0 0 3 3 】

バンクアドレスは、`count__m1__r` カウンタと `mod 3` 回路を定義する。M 1 メモリからのシンボルは、T R 一時レジスタへロードされる（図 7 参照）。バイトが反転されるかどうかは、`count__b__m1` カウンタのカウンタ状態に依存する（`count__b__m1 = 1 2, 1 3, 1 4, 1 5, 2 8, 2 9, 3 0, 3 1` の場合には、反転される）。消失ビットは反転されない。遅延されたシンボルは、第 1 のパス中に、C 1 復号器のシンドローム発生器に転送される。

## 【 0 0 3 4 】

C I R C 復号器の第 1 のパス中のデインターリーバタスク

M 2, a メモリは、デインターリーピングと、C 1 符号語を C 2 符号語に集めるために使用されそして、1 1 0 バンクを有する。各バンクは 3 2 シンボルを有する。各シンボルは、9 ビットを含む。デインターリーバのタスクは、C 1 符号語をデインターリーブし、そして、デインターリーブされた C 1 符号語を C 2 符号語に集めることである。デインターリーバタスクは、M 2, a メモリの対応するアドレスにより、C I R C 復号器の第 1 のパス中に M 2, a メモリ内で実行される。デインターリーバタスクは、C 1 復号、訂正及び、C 1 訂正された符号語をメモリ M 2, a へ格納の後に実行される。M 2, a メモリは、循環の性質を有する。 10

## 【 0 0 3 5 】

図 8 と 9 A 及び 9 B は、M 2, a メモリを示す。良い理解のために、M 2, a メモリを 2 次元のデカルト座標系の要素として記載できる（図 9 A 及び 9 B）。垂直 Y 軸の位置の各々は、3 2 シンボルのデータ（C 1 符号語）を含む。各水平 X 軸は、各 Y 軸位置でシンボル（9 ビット）を表す。シンボルは、各 Y 軸位置内で、左から右に移動して、順次に番号が付されている。 20

## 【 0 0 3 6 】

C 2 符号語を含むシンボルは、M 2, a メモリ空間内の対角線上に沿って配置される。1, 2, 3, 4, 5 4 及び、1 0 8 復号器サイクル内の C 2 符号語のシンボルが、図 9 A 及び 9 B に示されている。

## 【 0 0 3 7 】

連続する復号サイクルでは、C 2 符号語を含むシンボルは変更される。C 2 符号語を書きこむために最初に配置されたシンボルは、Y 位置 0 及び X 位置 0 に配置される（図 8）。C 2 符号語内のシンボルのアドレスを計算する一般的な式は、  

$$add\_c2 = (count\_rsd + 4 * I) \% mod110,$$
 であり、ここで、`count__rsd` は、復号器サイクル（0, . . . , 1 0 9）のカウンタであり、これは、Y 位置（バンクメモリ）を定義し、そして、I は 0 から 2 7 の値を有する。 30

## 【 0 0 3 8 】

C I R C 復号器の第 2 のパス中のデインターリーバタスク

C I R C 復号器の第 2 のパス中のデインターリーバタスクは、M 2, b メモリ内で実行される。それは、等しくない長さの遅延線を有する典型的なデインターリーバである（図 1 参照）。デインターリーバタスクは、C 1 復号、訂正及び、C 1 訂正された符号語を `buf__reg` レジスタへ格納の後に実行される。 40

## 【 0 0 3 9 】

読み出し + 書きこみ動作中の M 2, a メモリ内のデータのアドレッシング

C 1 及び C 2 復号器は、C I R C 復号器内のデータについて、同時に動作している。C 1 復号器は、C 2 復号器の前のデータに動作する。C 1 復号器の出力（第 1 パス）は、最初に、Y 位置 1 0 9 に C 1 符号語を書き込むように位置を定められる（図 8）。各連続する復号器サイクルで、C 1 復号器の出力は、次の連続するより高い値の Y 位置（1 0 9, 0, 1, . . . , 0）で訂正された C 1 符号語に位置を定められそして訂正された C 1 符号語を書き込む。C 1 復号器の入力（第 2 パス）は最初に、Y 位置 0 で C 1 符号語を書き込みように配置される（図 8）。各連続する復号器サイクルで、C 1 復号器の入力は、次の 50

連続するより高い値の Y 位置 ( 0 , 1 , 2 , . . . ) で C 1 符号語に位置を定められそして、C 1 符号語を読み出す。C 2 符号語を書き込むために最初に配置されたシンボルは、Y 位置 0 で且つ X 位置 0 である ( 図 8 )。

#### 【 0 0 4 0 】

C I R C 復号器の 2 回のパスについての M 2 , a メモリのデータ処理 ( 図 5 と図 8 参照 ) は、以下のステップを含む：

- 1 ) デインターリーピング、M 2 , a からの C 2 符号語を C 2 復号器へ読みこむ；
- 2 ) C 1 復号器からの訂正された C 2 符号語を M 2 , a メモリへ書き込む；
- 3 ) C 2 復号器からの訂正された C 2 符号語を M 2 , a メモリへ書き込む；
- 4 ) M 2 , a メモリから C 1 復号器へ C 1 符号語を読み出す。

10

#### 【 0 0 4 1 】

C I R C 復号器の単一のパスについての M 2 , a メモリのデータ処理 ( 図 5 と図 8 参照 ) は、以下のステップを含む：

- 1 ) デインターリーピング、M 2 , a からの C 2 符号語を C 2 復号器へ読みこむ；
- 2 ) C 1 復号器からの訂正された C 1 符号語を M 2 , a メモリへ書き込む。

#### 【 0 0 4 2 】

図 1 0 ( アドレス発生器 ) は、対応するステップ中の、M 2 , a メモリのアドレス ( a d d \_ m 2 ) の計算を示す。c o u n t \_ r s d カウンタは、復号器サイクルを定義し、そして、書き込み及び読み出し動作中の M 2 , a メモリの異なるバンクアドレスを計算する基本カウンタである。c o u n t \_ b \_ m 2 \_ c 1 カウンタは、バンク内のシンボルアドレスを定義する ( カウンタは 3 1 までカウントアップする )。c o u n t \_ b \_ m 2 \_ c 2 カウンタは、バンク内のシンボルアドレスを定義する ( カウンタは 2 7 までカウントアップする )。+ 1 s y m \_ a d d で、c o u n t \_ b \_ m 2 \_ c 1 カウンタと c o u n t \_ b \_ m 2 \_ c 2 カウンタはカウントアップする。サブブロック 1 は、C I R C 復号器の 2 回のパスの第 1 ステップ中に、アドレスを計算する。M 2 , a メモリのバンクアドレスを計算する式は、

20

$$b a n k \_ a d d = ( c o u n t \_ r s d + 4 * c o u n t \_ b \_ m 2 \_ c 2 ) \% m o d 1 1 0、$$

であり、ここで、c o u n t \_ b \_ m 2 \_ c 2 カウンタはバンク内のシンボルのアドレスを定義する。シンボルは、これらのアドレスに従って、M 2 , a メモリから C 2 復号器へ、読みこまれる。サブブロック 2 は、2 回のパスの第 2 ステップ中にアドレスを計算する。M 2 , a メモリのバンクアドレスの計算のための式は、

30

$$b a n k \_ a d d = ( c o u n t \_ r s d + 1 0 9 ) \% m o d 1 1 0、$$

であり、ここで、c o u n t \_ b \_ m 2 \_ c 1 カウンタはバンク内のシンボルのアドレスを定義する。訂正された C 1 符号語のシンボルは、これらのアドレスに従って、C 1 復号器から M 2 , a メモリへ書き込まれる。サブブロック 3 は、2 回のパスの第 3 ステップ中にアドレスを計算する。M 2 , a メモリのバンクアドレスの計算のための式は、

$$b a n k \_ a d d = ( c o u n t \_ r s d + 4 * c o u n t \_ b \_ m 2 \_ c 2 ) \% m o d 1 1 0、$$

であり、ここで、c o u n t \_ b \_ m 2 \_ c 2 カウンタはバンク内のシンボルのアドレスを定義する。訂正された C 2 符号語のシンボルは、これらのアドレスに従って、C 2 復号器から M 2 , a メモリへ書き込まれる。サブブロック 4 は 2 回のパスの第 4 ステップ中にアドレスを計算する。M 2 , a メモリのバンクアドレスの計算のための式は、

40

$$b a n k \_ a d d = ( c o u n t \_ r s d + 1 0 9 ) \% m o d 1 1 0、$$

であり、ここで、c o u n t \_ b \_ m 2 \_ c 1 カウンタはバンク内のシンボルのアドレスを定義する。シンボルは、これらのアドレスに従って、M 2 , a メモリから C 1 復号器へ読み出される。

#### 【 0 0 4 3 】

C I R C 復号器の第 2 のパス中のデインターリーバタスク

使用されるべきこのタスクのために、従来のデインターリーバは、図 1 を参照して、等し

50

くない長さの遅延線を有する。このデインターリーバは、M2, bメモリ内で実行される。

【0044】

2シンボル遅延タスク

M3メモリは、CIRC復号器内の2シンボル遅延タスクのために使用され、そして、アドレス0, 1, 2の3つのバンクを有する。bank0は、28シンボルよりなり、bank1とbank2は、各々が24シンボルよりなる。C2復号と訂正後のデータは、bank0へロードされる。2シンボル遅延タスクは、2つのステップ：

1) 遅延されたデータを出力論理に読みこむ；

2) M3メモリを上書きする、

を有する。

【0045】

以下の表1は、第1ステップ中にアドレスを発生するために使用される。

【0046】

【表1】

count_b	0	1	2	3	4	5	6	7	8	9	10	11
bank	0	0	0	0	2	2	2	2	0	0	0	0
sym_add	0	1	6	7	4	5	6	7	2	3	8	9

count_b	12	13	14	15	16	17	18	19	20	21	22	23
bank	2	2	2	2	0	0	0	0	2	2	2	2
sym_add	12	13	14	15	4	5	10	11	20	21	22	23

ここで、count\_bは、バイトカウンタであり、bankは、M3内のバンクアドレスであり、そして、sym\_addは、bank内のシンボルアドレスである。

【0047】

以下の表2は、第2ステップ中にアドレスを発生するために使用される（読み出し／書き込み）。

【0048】

【表2】

count_b	0	1	2	3	4	5	6	7	8	9	10	11
bank_r	1	1	1	1	0	0	0	0	1	1	1	1
sym_add	4	5	6	7	16	17	22	23	12	13	14	15
bank_w	2	2	2	2	1	1	1	1	2	2	2	2
sym_add	4	5	6	7	4	5	6	7	12	13	14	15

count_b	12	13	14	15	16	17	18	19	20	21	22	23
bank_r	0	0	0	0	1	1	1	1	0	0	0	0
sym_add	18	19	24	25	20	21	22	23	20	21	26	27
bank_w	1	1	1	1	2	2	2	2	1	1	1	1
sym_add	12	13	14	15	20	21	22	23	20	21	22	23

CIRC復号器のシミュレーション結果（2回のパス）

図11は、異なるバースト誤りの訂正の結果を示す。従来のCIRC復号器の訂正結果が、破線で示されており、本発明のCIRC復号器の結果が実線で示されている。以下の誤りの組合せが、訂正の入力であった：

10

20

30

40

50

- 1) 各符号語内に3つの誤り(ランダム誤り)を有する20のC1符号語のバースト誤り;
- 2) 各符号語内に4つの誤りを有する20のC1符号語のバースト誤り;
- 3) 各符号語内に5つの誤りを有する20のC1符号語のバースト誤り;
- 4) 各符号語内に6つの誤りを有する20のC1符号語のバースト誤り;
- 5) 各符号語内に3つの誤りを有する30のC1符号語のバースト誤り;
- 6) 各符号語内に4つの誤りを有する30のC1符号語のバースト誤り;
- 7) 各符号語内に5つの誤りを有する30のC1符号語のバースト誤り;
- 8) 各符号語内に6つの誤りを有する30のC1符号語のバースト誤り;
- 9) 各符号語内に3つの誤りを有する40のC1符号語のバースト誤り;
- 10) 各符号語内に4つの誤りを有する40のC1符号語のバースト誤り;
- 11) 各符号語内に5つの誤りを有する40のC1符号語のバースト誤り;
- 12) 各符号語内に5つの誤りを有する40のC1符号語のバースト誤り。

10

## 【0049】

本発明のCIRC復号器と従来のCIRC復号器の両方は、14の完全に壊れたC1符号語のバースト誤りを訂正できる。従来のCIRC復号器と比較して、優位に本発明のCIRC復号器は、各符号語内に3又は4の誤りを有する20のC1符号語のバースト誤りを訂正できることは、明らかである。各符号語が5又は6の誤りを有する場合には、復号器は、バースト誤りを訂正できないが、しかし、オーディオデータのフラグの付されたバイトレートは、従来のCIRC復号器よりも、まだ著しく低い。'30のC1符号語'のバースト誤りでも、より良い訂正能力が、得られる。本発明のCIRC復号器は、各符号語内に3つの誤りを有する30のC1符号語のバースト誤りを訂正でき、そして、フラグの付されるバイトレートは、符号語当りのより高い誤り数に対しても、より低い。

20

## 【0050】

4つの別のメモリの別々の制御とアドレッシングは、本発明のCIRC復号器の速度を大きく増加させることに、注意する。

## 【0051】

読み出し及び書き込み動作中のM1、M2、aメモリ及び、M2、a、M2、bメモリの並列の制御とアドレッシングも、本発明のCIRC復号器の速度を大きく増加させる。

## 【0052】

上述のCIRC復号器と比較して、より小さなサイズのメモリが、デインターリーバの実行に使用される。

30

## 【0053】

デインターリーバタスクは、第1のパス中にM2、aメモリ内で実行される。

## 【0054】

M2、aメモリのサイズは、110×32×9である。デインターリーバタスクは、第2のパス中に、M2、bメモリ内で実行される。M2、bメモリのサイズは、1404×9(従来のデインターリーバ)である。サイズ256×32×9を有するメモリは、上述のCIRC復号器内の両方のデインターリーバについて使用される。

## 【0055】

2シンボル遅延タスクは、M3メモリ内で実行される。M3メモリは、調査テーブルとして動作し、これも、CIRC復号器の動作速度を増加させる。

40

## 【0056】

本発明のCIRC復号器では、CIRC復号器の単一のパスのモードを実行することが可能である。

## 【0057】

本発明の復号方法は、いわゆるRS符号の消失訂正法を使用することに適する。C1の壊れた符号語の大きなバースト誤りの訂正は、低入力誤り率により非常に効率的である(各C1符号語は多くても6までのランダム誤りを含む)。

## 【0058】

50

## 【発明の効果】

上述のように、本発明により、C I R C 訂正のための改善された方法を提供できる。

## 【図面の簡単な説明】

【図 1】従来技術の C I R C 復号器のブロック図である。

【図 2】本発明の実施例に従った、C I R C 復号器の第 1 のパスを示すブロック図である。

【図 3】図 2 の復号器の第 2 のパス示すブロック図である。

【図 4】C I R C 復号器の代わりの実施例を示すブロック図である。

【図 5】C I R C 訂正符号の 2 回の及び単一の処理のための C I R C 復号器の動作のフローチャート示す図である。

10

【図 6】M 1 メモリ内のデータフローを示す機構を示す図である。

【図 7】M 1 メモリ内のアドレス計算を示すブロック図である。

【図 8】メモリ M 2 の動作を示す図である。

【図 9 A】更に、メモリ M 2 の動作を示す図である。

【図 9 B】更に、メモリ M 2 の動作を示す図である。

【図 1 0】M 2 メモリのアドレスの計算を示すブロック図である。

【図 1 1】異なるバースト誤りの訂正結果を示す特徴線を示す図である。

## 【符号の説明】

D 1    デインターリーバ

D 2    デインターリーバ

M 1    メモリ

M 2 , a    メモリ

M 2 , b    メモリ

M 3    メモリ

1    入力論理

3    マルチプレクサ

4    R S 復号器

6    デマルチプレクサ

7    M 2 , a メモリ

9    M 2 , b メモリ

1 0    コントローラ

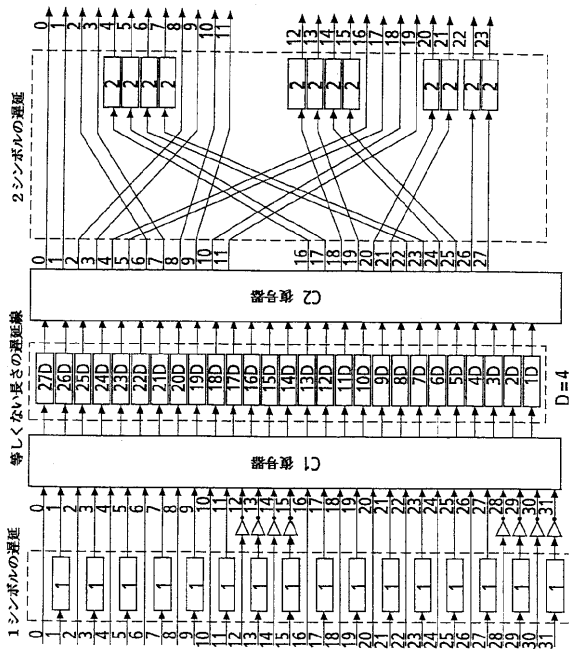
1 1    マルチプレクサ

1 4    M 3 メモリ

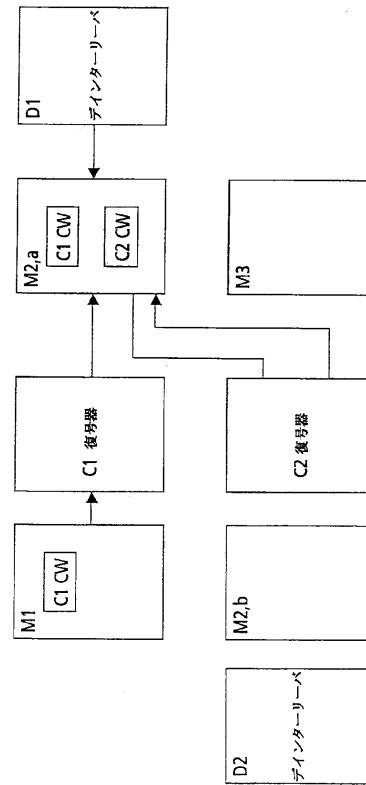
20

30

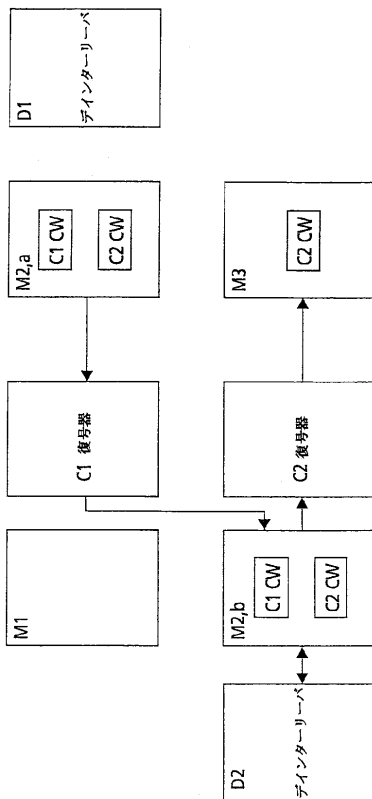
【図 1】



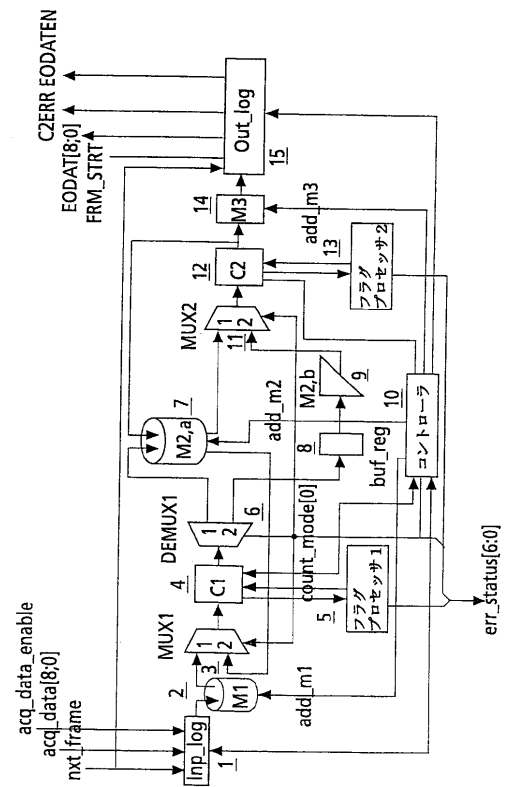
【図 2】



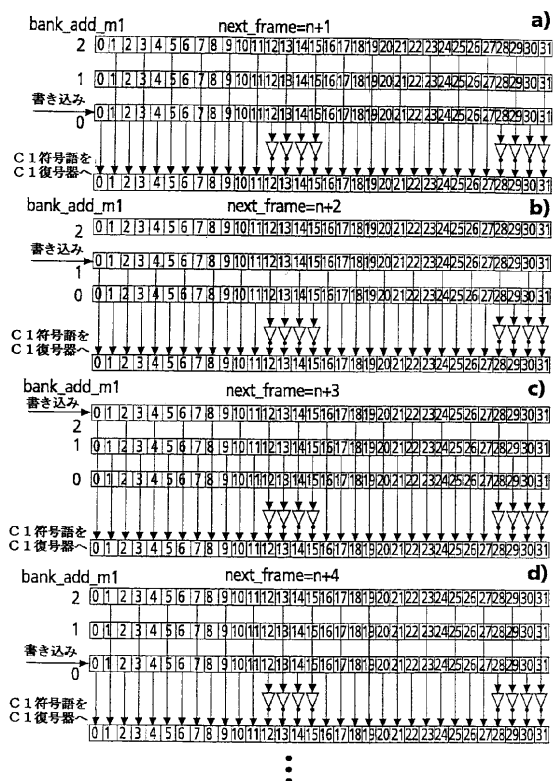
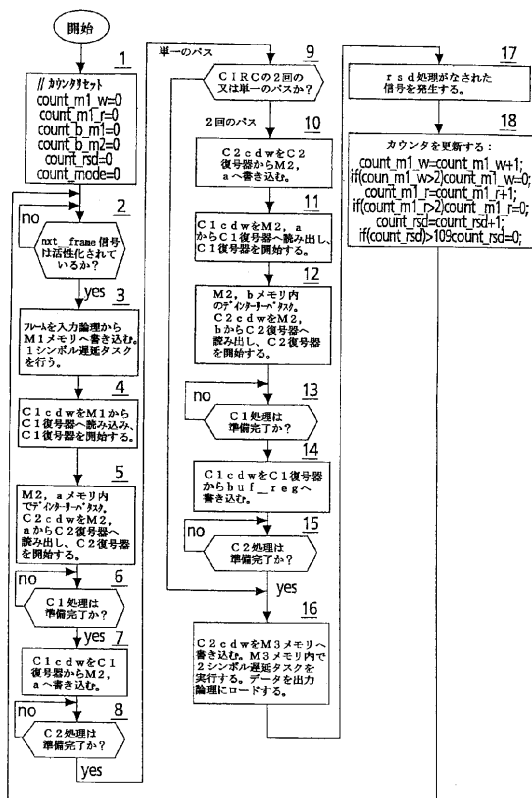
【図 3】



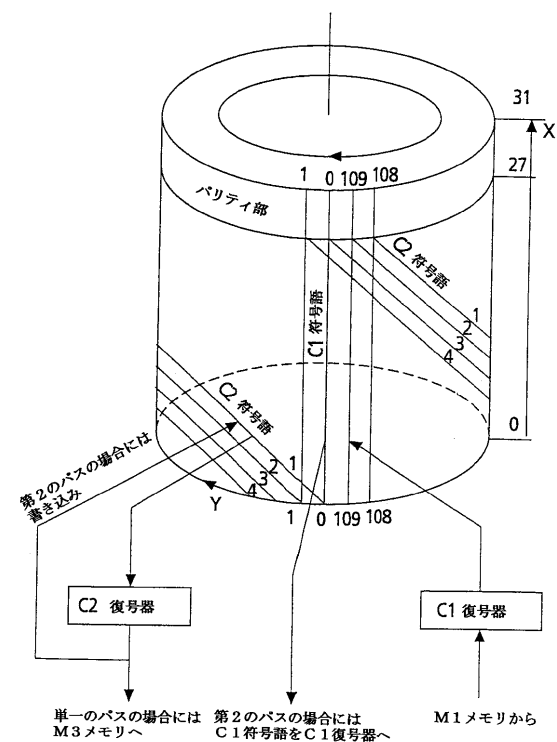
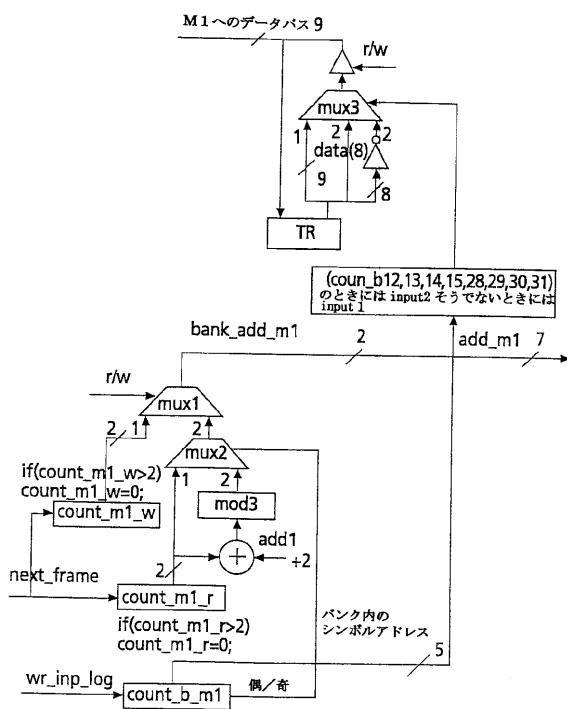
【図 4】



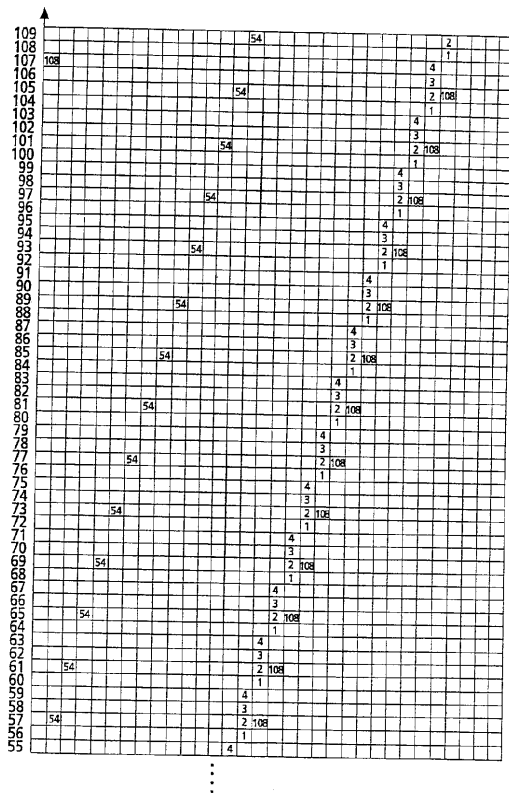
【 図 6 】



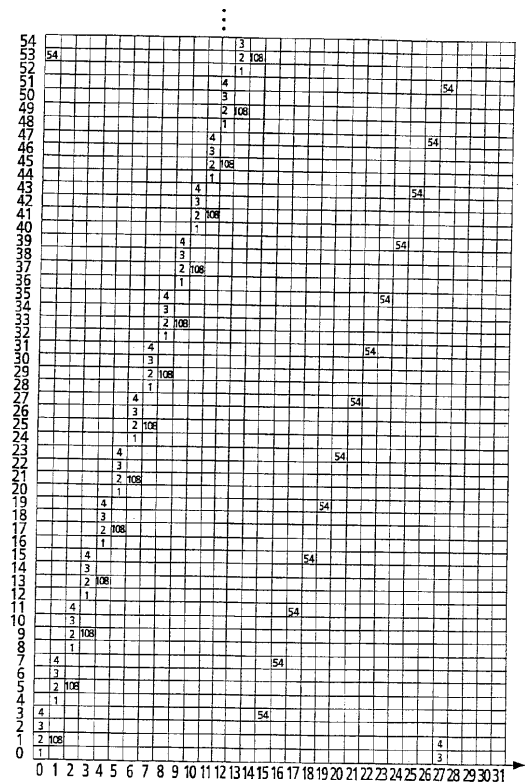
【 ㄨ 8 】



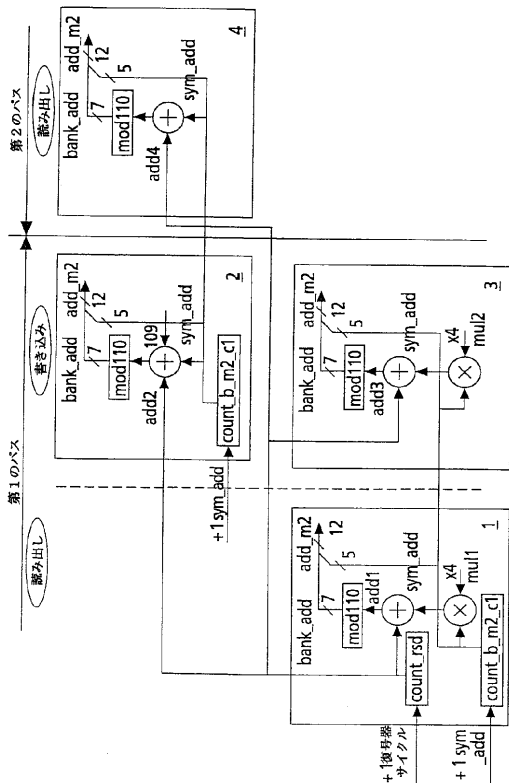
【図 9 A】



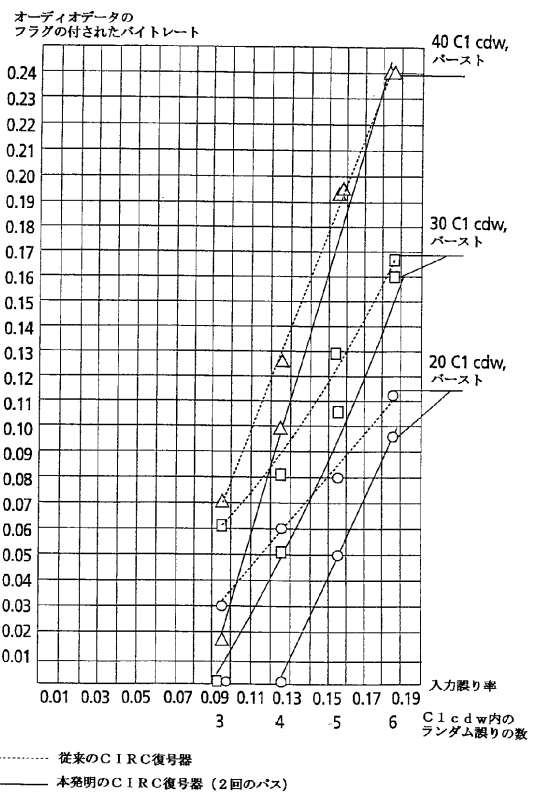
【図 9 B】



【図 10】



【図 11】





## フロントページの続き

(51)Int.Cl.<sup>7</sup> F I テーマコード(参考)

	G 1 1 B	20/18	5 3 6 B
	G 1 1 B	20/18	5 3 6 E
	G 1 1 B	20/18	5 7 2 C
	G 1 1 B	20/18	5 7 2 F
	H 0 3 M	13/15	

(74)代理人 100120307  
弁理士 中村 雅文

(72)発明者 アレクサンダー クラフチェンコ  
ドイツ連邦共和国 7 8 0 5 6 フィリンゲン - シュヴェニンゲン ドイツンベルクリング 1 4  
6

(72)発明者 マルテン カブツ  
ドイツ連邦共和国 7 8 0 5 2 フィリンゲン - シュヴェニンゲン ティローラーシュトラッセ  
2 3

(72)発明者 ヴィジャヤ ラマドス  
アメリカ合衆国 インディアナ州 4 6 0 7 4 ウェストフィールド ウェスト・コロンバイン・  
レーン 3 0 1

(72)発明者 アミット シン  
アメリカ合衆国 インディアナ州 4 6 2 4 0 インディアナポリス ノートル・ダム・ドライブ  
9 2 5 4 アpartment シー

F ターム(参考) 5B001 AA11 AB03 AC05 AD04  
5J065 AA01 AA03 AB01 AC03 AD01 AD11 AE06 AF02 AG06 AH06  
AH09