

(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.C1.<sup>7</sup>**H03M 13/27****G06F 11/10****G11B 20/18****H03M 13/15**

F 1

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

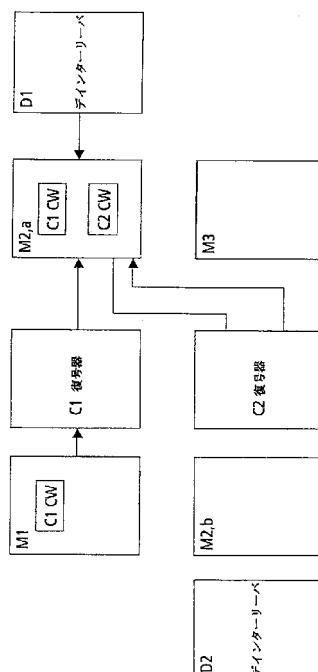
(54) 【発明の名称】クロスインターリーブリード-ソロモン符号訂正を行う方法及び装置

## (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符号語をディンターリープするステップと、

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C1符号語が訂正可能である場合には、第2のC1符号語は訂正された第1のC1符号語であり、そして、第1のC1符号語が訂正可能でない場合には、第2のC1符号語は第1のC1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C1符号語が訂正可能でない場合には、第2のC1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C2符号語を、第2のメモリ手段からC2復号器手段へ入力するステップと、
- 第2のC2符号語を、C2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C2符号語が訂正可能である場合には、第2のC2符号語は訂正された第1のC2符号語であり、そして、第1のC2符号語が訂正可能でない場合には、第2のC2符号語は第1のC2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C2符号語が訂正可能でない場合には、第2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C1符号語を、第2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3のC1符号語を、C1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C2符号語を発生するために、第3のC1符号語をディンターリープするステップと、

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C1符号語が訂正可能である場合には、第3のC1符号語は訂正された第2のC1符号語であり、そして、第2のC1符号語が訂正可能でない場合には、第3のC1符号語は第2のC1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C1符号語が訂正可能でない場合には、第3のC1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。 20

**【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。  
。

**【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。 30

**【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】**

本発明は、一般的には、リード - ソロモン符号を使用してデ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する。 30

**【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、インターリープ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、デインターリープが実行され、そして、その後に、C2復号が実行される。

**【0004】**

10

20

30

40

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。

#### 【0005】

C I R C 訂正符号の従来の復号方法に従って、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C1復号で実行され、そして、C1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C2復号で実行される。

#### 【0006】

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C1とC2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是、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。

#### 【0007】

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ず、そして、さらにポインタ情報の信頼性が比較的高くなければならない。

#### 【0008】

C I R C 訂正符号の従来の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C1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、C1ポインタは次の段階のC2復号器へ送られ、それにより、誤り訂正がC1ポインタを使用してC2復号器で実行される。

#### 【0009】

特に、従来のC I R C 訂正符号の場合には、C1符号の系列(C1系列)が、交互に2つの隣接するフレーム(1フレーム:32シンボル)に含まれる、32シンボルにより構成される。C2符号の系列(C2系列)は、108の連続するフレームの予め定められたものに含まれる28シンボルより構成される。C1系列のインターリーブ長は、C2系列のインターリーブ長よりも短いので、キューやリピューブ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、C1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、C1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、C2系列のインターリーブ長は108フレームを有し、そして、これらの108フレームは不連続の点を含みうる。従って、これも正しいC2系列ではない。前述のC1ポインタを使用して、不正確なC2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。

#### 【0010】

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C1復号とC2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C1復号とC2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、C1復号、それに続きC2復号、それに続きC1復号、そして、それに続きC2復号である。最初のC2復号では、前の段階でのC1復号により得られたC1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是、第2のC1復号とC2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の方法に似ている。

#### 【0011】

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。誤り訂正是、シンドロームの乗算の結果と誤り位置を使用して実行され、他のシンドロームは、それにより新たなシンドロ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。

#### 【0012】

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 (32, 28) を復号しそして、誤りのあるシンボルを訂正する ;
- 3 . デインターリープする ;
- 4 . C 2 符号 (28, 24) 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。

#### 【0013】

1 フレームを構成する 32 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 32 シンボルは、C 1 復号器に供給され、ここで、(32, 28) 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、デインターリープ回路 (等しくない長さの遅延線) で処理される。デインターリープ回路の出力は、C 2 復号器に供給される。デインターリープ回路では、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プと同様な方法でデインターリープされる。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。

#### 【0014】

##### 【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。

#### 【0015】

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。

### 【0016】

#### 【課題を解決するための手段】

本発明は、請求項1に従った、CIRC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。

### 【0017】

本発明は、CIRC復号器の単一バス及び2回のバ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、CIRC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。

### 【0018】

本発明は、CIRC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。

### 【0019】

#### 【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。

### 【0020】

図2は、CIRC復号器のブロック図を示す。この復号器は、C1符号語(CW)を格納するメモリM1を有する。更に、C1復号器、メモリM2,aと関連するデインターリーバD1、メモリM2,bと関連するデインターリーバD2,C2復号器及び、メモリM3がある。

### 【0021】

第1のパスの最初では、C1CWはメモリM1に格納される。そこから、C1CWはC1復号器に入力される。C1復号器は、他のC1CWを、M2,aに出力する。C1復号器から出力されたC1CWは、メモリM1のC1CWが訂正可能でない場合には、メモリM1のC1CWのコピーである。この場合には、C1CW内の全シンボルについて、消失フラグ設定される。メモリM1のC1CWが訂正可能な場合には、C1復号器から出力されたC1CWは、メモリM1の訂正されたC1CWである。デインターリーブ動作は、デインターリーバD1により実行される。この結果、C2CWは、メモリM2,aに格納される。メモリM2,aのC2CWは、C2復号器へ入力される。C2復号器はC2CWをメモリM2,aへ書き戻す。このC2CWは、元のC2CWが訂正可能でない場合には元のC2CWのコピーであり、又は、訂正されたC2CWであるかのいずれかである。C2CW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。

### 【0022】

図3は、第2のパスを示す。M2,aのC1CWは、C1復号器は入力される。C1復号器はC1CWをメモリM2,bへ出力する。メモリM2,bのC1CWは、メモリM2,aのC1CWが訂正可能でない場合にはメモリM2,aのC1CWのコピーであり、又は、メモリM2,aの訂正されたC1CWであるかのいずれかである。メモリM2,aのC1CWが訂正可能でない場合には、C1復号器から出力されそしてメモリM2,bに格納される符号語C1CWに、消失フラグが、設定される。デインターリーバD2が開始する。この結果、符号語C2CWがメモリM2,bに供給される。これは、C2復号器に入力される。C2復号器は、符号語C2CWをメモリM3に出力する。C2復号器により出力されるC2CWは、メモリM2,bのC2CWが訂正可能でない場合にはメモリM2,bのC2CWのコピーであり又は、訂正されたメモリM2,bのC2CWであるかのいずれかである。メモリM2,bのC2CWが訂正可能でない場合には、消失フラグが設定される。

### 【0023】

以下に更に詳細な実施例を説明する。

10

20

30

40

50

## 【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、デインターリープしそして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、デインターリープしそして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) デインターリープ及び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。n × t\_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メモリのバンクヘロードされ、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、シンドロームが計算される。32シンボルを転送した後に、復号処理C1が開始される。デインターリープタスク(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、シンドロームが計算される。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復号器の单

10

20

30

40

50

ーの又は 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 MUX 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、3 2 シンボルの転送後に開始される（ステップ 11）。デインターリバータスク（ステップ 12）は、M 2 , b メモリ内で実行される。2 8 シンボルの C 2 符号語が、M 2 , b メモリから MUX 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、2 8 シンボルの転送後に開始される。

10

20

20

## 【 0 0 2 8 】

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、buf \_ reg 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、ECC 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。

## 【 0 0 2 9 】

## 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、デインターリービング及び、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、読み出し及び書き込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。

## 【 0 0 3 0 】

## 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 3 2 シンボルより構成される。各シンボルは、9 バイトを含む。MS 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 next \_ frame 信号後に繰返される、即ち、図 6 a と図 6 d は同一である。

30

## 【 0 0 3 1 】

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：

40

$$\text{bank\_add\_m1\_w} = \text{count\_m1\_w},$$

ここで、count \_ m1 \_ w カウンタは、書き込み動作中の M 1 メモリ内のバンクアドレスを定義する。3 2 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。count \_ b \_ m1 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。

## 【 0 0 3 2 】

バンクアドレスは次の式に従って、計算される：

$$\text{bank\_add\_0\_m1\_r} = \text{count\_m1\_r}, \quad // \text{偶数}$$

$$\text{bank\_add\_1\_m1\_r} = (\text{count\_m1\_r} + 2) \% 3, \quad // \text{奇数}.$$

50

## 【0033】

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。`M1` メモリからのシンボルは、`TR` 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 = 12, 13, 14, 15, 28, 29, 30, 31` 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、`C1` 復号器のシンドローム発生器に転送される。

## 【0034】

`CIRC` 復号器の第1のパス中のデインターリーバタスク

`M2, a` メモリは、デインターリービングと、`C1` 符号語を`C2` 符号語に集めるために使用されそして、`110` バンクを有する。各バンクは`32` 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、`C1` 符号語をデインターリープし、そして、デインターリープされた`C1` 符号語を`C2` 符号語に集めることである。デインターリーバタスクは、`M2, a` メモリの対応するアドレスにより、`CIRC` 復号器の第1のパス中に`M2, a` メモリ内で実行される。デインターリーバタスクは、`C1` 復号、訂正及び、`C1` 訂正された符号語をメモリ`M2, a` へ格納の後に実行される。`M2, a` メモリは、循環の性質を有する。

## 【0035】

図8と9A及び9Bは、`M2, a` メモリを示す。良い理解のために、`M2, 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9A及び9B）。垂直Y軸の位置の各々は、`32` シンボルのデータ（`C1` 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。

## 【0036】

`C2` 符号語を含むシンボルは、`M2, 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, 54` 及び、`108` 復号器サイクル内の`C2` 符号語のシンボルが、図9A及び9Bに示されている。

## 【0037】

連続する復号サイクルでは、`C2` 符号語を含むシンボルは変更される。`C2` 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）。`C2` 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、

$$\text{add\_c2} = (\text{count\_rsd} + 4 * I) \% \text{mod } 110,$$

であり、ここで、`count_rsd` は、復号器サイクル（`0, . . . , 109`）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`27` の値を有する。

## 【0038】

`CIRC` 復号器の第2のパス中のデインターリーバタスク

`CIRC` 復号器の第2のパス中のデインターリーバタスクは、`M2, 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、`C1` 復号、訂正及び、`C1` 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。

## 【0039】

読み出し+書き込み動作中の`M2, a` メモリ内のデータのアドレッシング

`C1` 及び`C2` 復号器は、`CIRC` 復号器内のデータについて、同時に動作している。`C1` 復号器は、`C2` 復号器の前のデータに動作する。`C1` 復号器の出力（第1バス）は、最初に、Y位置`109` に`C1` 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、`C1` 復号器の出力は、次の連続するより高い値のY位置（`109, 0, 1, . . . , 0`）で訂正された`C1` 符号語に位置を定められそして訂正された`C1` 符号語を書き込む。`C1` 復号器の入力（第2バス）は最初に、Y位置`0` で`C1` 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、`C1` 復号器の入力は、次の

10

20

30

40

50

連続するより高い値のY位置(0, 1, 2, . . .)でC1符号語に位置を定められそして、C1符号語を読み出す。C2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)。

#### 【0040】

CIRC復号器の2回のパスについてのM2, 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) ディンターリービング、M2, aからのC2符号語をC2復号器へ読みこむ；
- 2) C1復号器からの訂正されたC2符号語をM2, aメモリへ書き込む；
- 3) C2復号器からの訂正されたC2符号語をM2, aメモリへ書き込む；
- 4) M2, aメモリからC1復号器へC1符号語を読み出す。

10

#### 【0041】

CIRC復号器の単一のパスについてのM2, 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) ディンターリービング、M2, aからのC2符号語をC2復号器へ読みこむ；
- 2) C1復号器からの訂正されたC1符号語をM2, aメモリへ書き込む。

#### 【0042】

図10(アドレス発生器)は、対応するステップ中の、M2, aメモリのアドレス(ad d\_m2)の計算を示す。count\_rsd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M2, 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。count\_b\_m2\_c1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31までカウントアップする)。count\_b\_m2\_c2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27までカウントアップする)。+1sym\_addrで、count\_b\_m2\_c1カウンタとcount\_b\_m2\_c2カウンタはカウントアップする。サブブロック1は、CIRC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。M2, aメモリのバンクアドレスを計算する式は、

```
bank_addr = (count_rsd + 4 * count_b_m2_c2) % mod 110,
```

であり、ここで、count\_b\_m2\_c2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、M2, aメモリからC2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。M2, aメモリのバンクアドレスの計算のための式は、

```
bank_addr = (count_rsd + 109) % mod 110,
```

であり、ここで、count\_b\_m2\_c1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C1符号語のシンボルは、これらのアドレスに従って、C1復号器からM2, 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。M2, aメモリのバンクアドレスの計算のための式は、

```
bank_addr = (count_rsd + 4 * count_b_m2_c2) % mod 110,
```

であり、ここで、count\_b\_m2\_c2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C2符号語のシンボルは、これらのアドレスに従って、C2復号器からM2, 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。M2, aメモリのバンクアドレスの計算のための式は、

```
bank_addr = (count_rsd + 109) % mod 110,
```

であり、ここで、count\_b\_m2\_c1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、M2, aメモリからC1復号器へ読み出される。

#### 【0043】

CIRC復号器の第2のパス中のディ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、従来のディンターリーバは、図1を参照して、等し

30

40

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 C 1 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 C 1 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 C 1 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 C 1 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 C 1 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 C 1 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 C 1 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 C 1 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 C 1 符号語のバースト誤り；

10 10 ) 各符号語内に 4 つの誤りを有する 40 の C 1 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 C 1 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 C 1 符号語のバースト誤り。

#### 【 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 C 1 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 C 1 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 C 1 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 C 1 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、符号語当たりのより高い誤り数に対しても、より低い。

#### 【 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 M 1 、 M 2 , a メモリ及び、 M 2 , a 、 M 2 , 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。

#### 【 0053 】

デインターリーバタスクは、第 1 のバス中に M 2 , a メモリ内で実行される。

#### 【 0054 】

M 2 , a メモリのサイズは、 $110 \times 32 \times 9$  である。デインターリーバタスクは、第 2 のバス中に、 M 2 , b メモリ内で実行される。 M 2 , b メモリのサイズは、 $1404 \times 9$  (従来のデインターリーバ) である。サイズ $256 \times 32 \times 9$  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、 M 3 メモリ内で実行される。 M 3 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。

#### 【 0056 】

本発明の CIRC 復号器では、 CIRC 復号器の単一のバ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。 C 1 の壊れた符号語の大きなバースト誤りの訂正是、低入力誤り率により非常に効率的である（各 C 1 符号語は多くても 6 までのランダム誤りを含む）。

#### 【 0058 】

10

20

30

40

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つた、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 10 チャート示す図である。

【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 10】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 11】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

10 コントローラ

11 マルチプレクサ

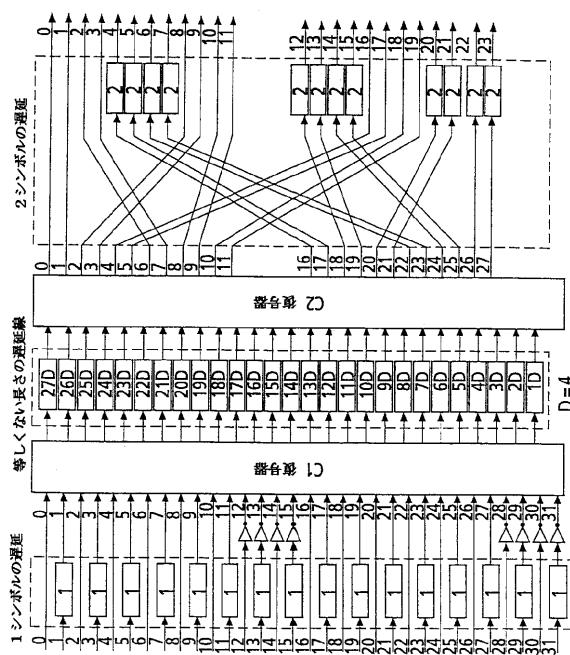
14 M 3 メモリ

10

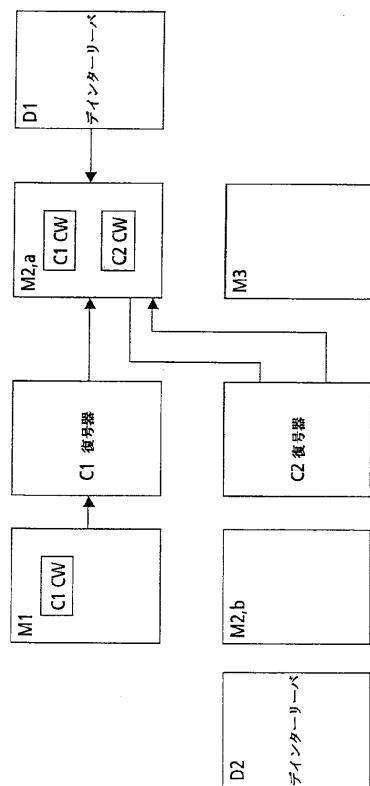
20

30

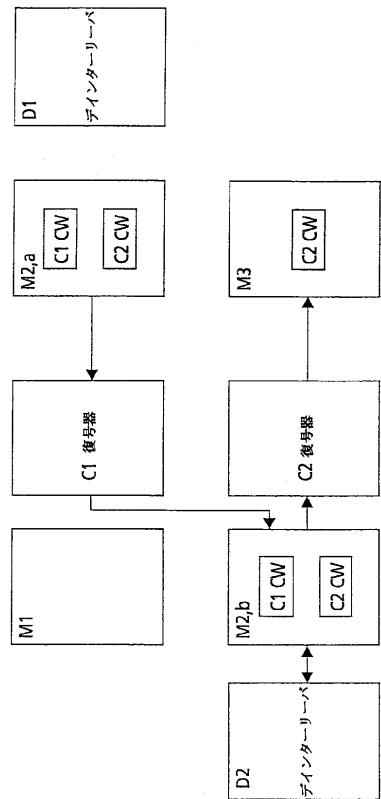
【図1】



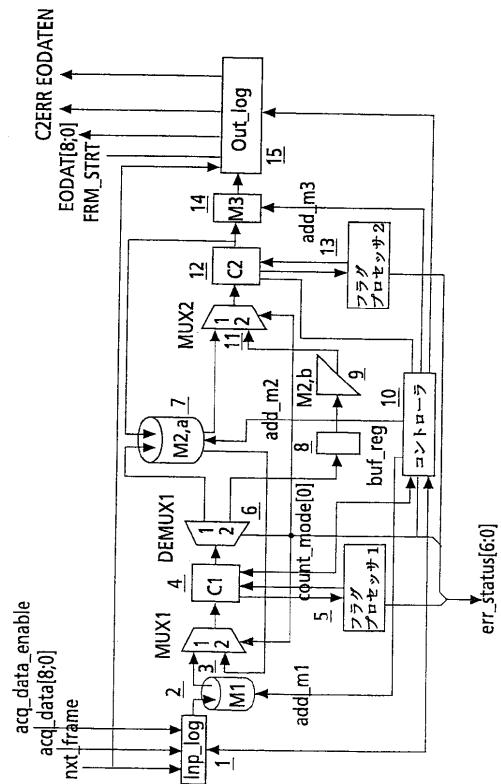
【図2】



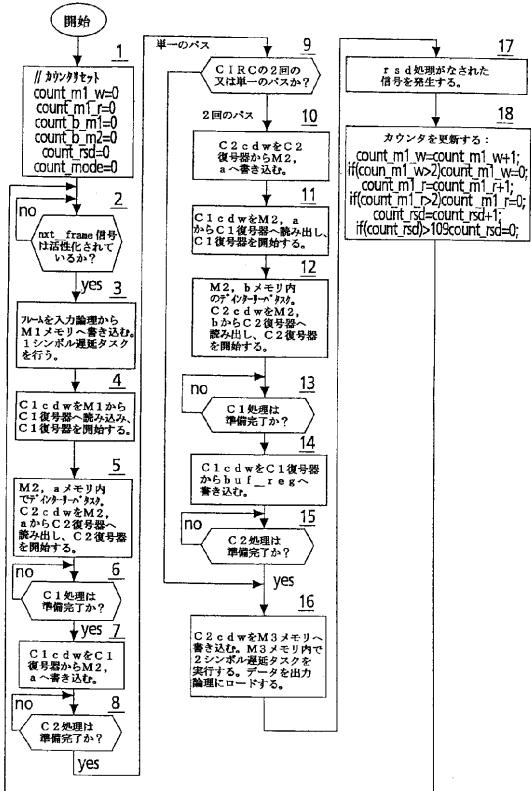
【図3】



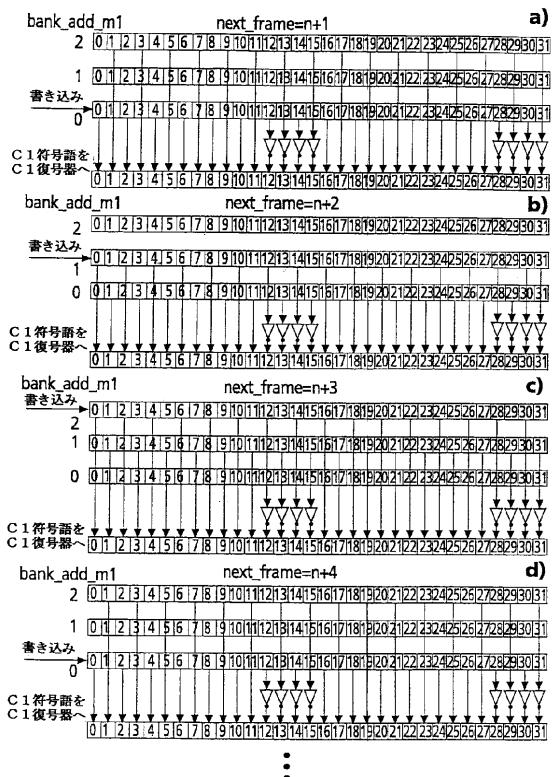
【図4】



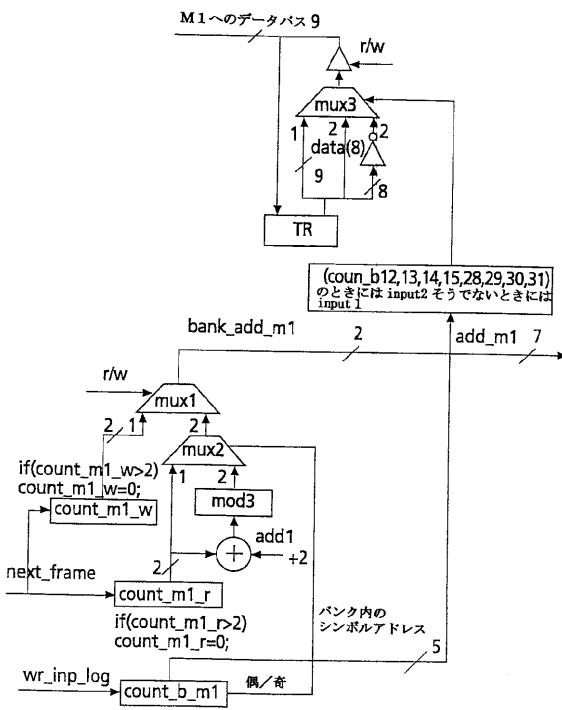
【図5】



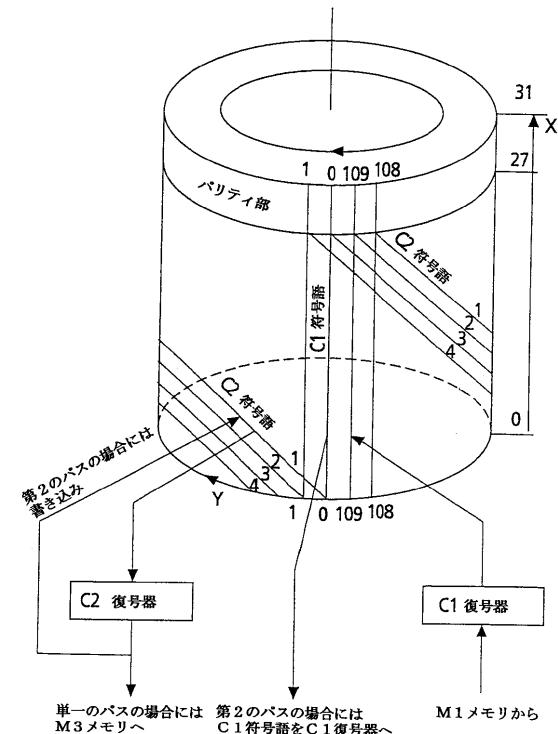
【図6】



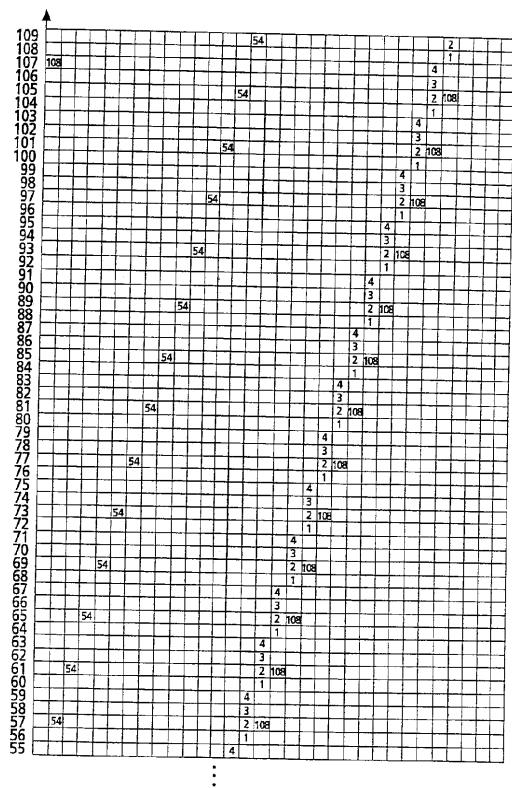
【図7】



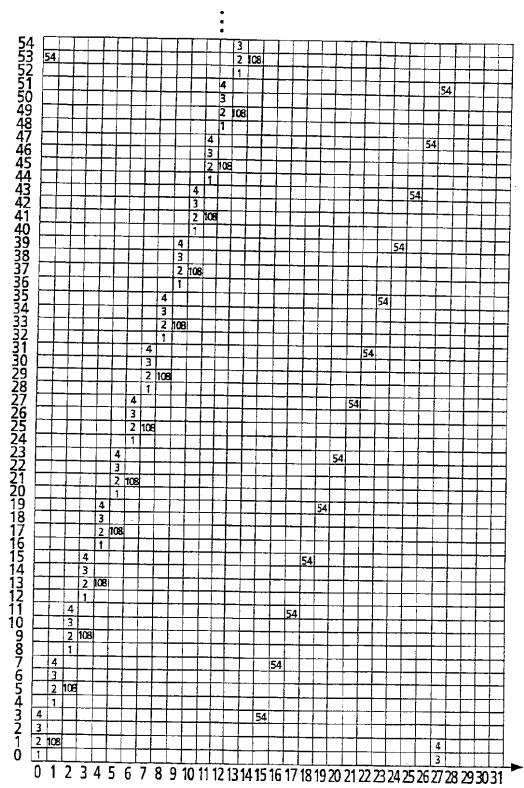
【図8】



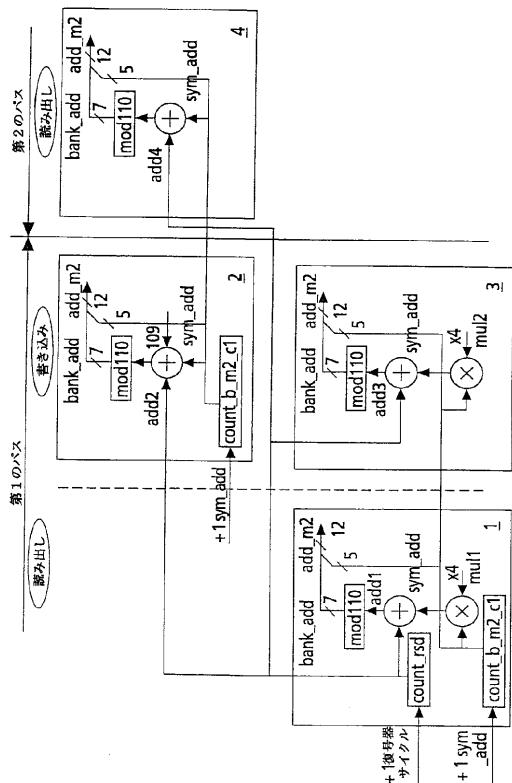
【図 9 A】



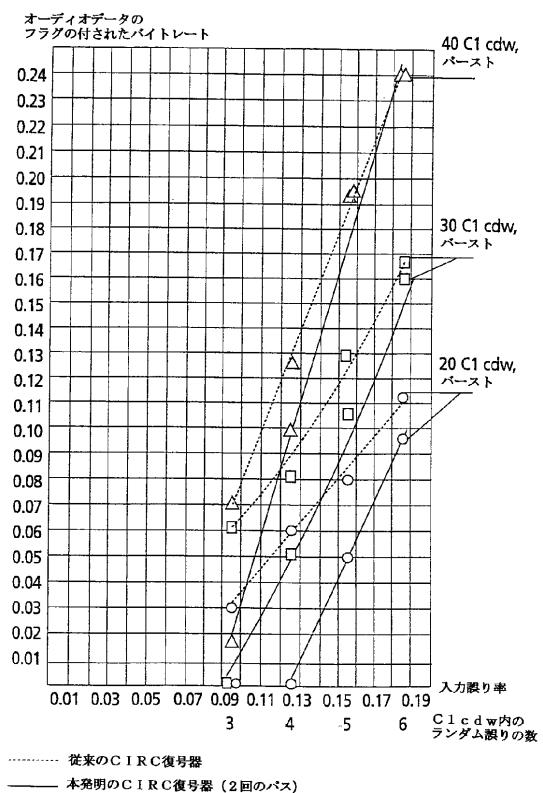
【図 9 B】



【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 11 B 20/18	5 3 6 B
	G 11 B 20/18	5 3 6 E
	G 11 B 20/18	5 7 2 C
	G 11 B 20/18	5 7 2 F
	H 03 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 78056 フィリングン - シュヴェニンゲン ドイテンベルクリング 14  
6

(72)発明者 マルテン カブツ

ドイツ連邦共和国 78052 フィリングン - シュヴェニンゲン ティローラーシュトラーセ  
23

(72)発明者 ヴィジヤヤ ラマドス

アメリカ合衆国 インディアナ州 46074 ウエストフィールド ウエスト・コロンバイン・  
レーン 301

(72)発明者 アミット シン

アメリカ合衆国 インディアナ州 46240 インディアナポリス ノートル・ダム・ドライヴ  
9254 アパートメント 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