

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4290881号
(P4290881)

(45) 発行日 平成21年7月8日(2009.7.8)

(24) 登録日 平成21年4月10日(2009.4.10)

(51) Int.Cl.

F I

G 1 1 B 20/18 (2006.01)

G 0 6 F 3/06 (2006.01)

G 0 6 F 11/10 (2006.01)

G 1 1 B 20/18 5 1 2 B

G 1 1 B 20/18 5 3 6 C

G 1 1 B 20/18 5 7 2 B

G 1 1 B 20/18 5 7 2 G

G 0 6 F 3/06 3 0 3 G

請求項の数 9 (全 20 頁) 最終頁に続く

(21) 出願番号 特願2000-540540 (P2000-540540)
 (86) (22) 出願日 平成10年1月16日 (1998.1.16)
 (65) 公表番号 特表2002-509331 (P2002-509331A)
 (43) 公表日 平成14年3月26日 (2002.3.26)
 (86) 国際出願番号 PCT/GB1998/000055
 (87) 国際公開番号 WO1999/036913
 (87) 国際公開日 平成11年7月22日 (1999.7.22)
 審査請求日 平成17年1月6日 (2005.1.6)

前置審査

(73) 特許権者 398038580
 ヒューレット・パカード・カンパニー
 HEWLETT-PACKARD COMPANY
 アメリカ合衆国カリフォルニア州パロアル
 ト ハノーバー・ストリート 3000
 (73) 特許権者 000002185
 ソニー株式会社
 東京都港区港南1丁目7番1号
 (74) 代理人 100087642
 弁理士 古谷 聡
 (74) 代理人 100076680
 弁理士 溝部 孝彦
 (74) 代理人 100121061
 弁理士 西山 清春

最終頁に続く

(54) 【発明の名称】 エラー訂正コードを含む磁気媒体上へのデータの格納のための方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

複数のデータレコードからなるストリームを磁気媒体上に格納する装置であって、
 前記複数のデータレコードを複数のデータバイトからなる複数のグループにグループ
 化するグループフォーマット手段と、

前記複数のグループのそれぞれを、各サブグループが1以上のデータトラックに対応
 するデータバイトを含む複数のサブグループに分割するサブグループ処理手段と、

各データトラックからの前記データバイトをチェックサムアルゴリズムに適用し、各
 データトラックについて1以上のトラックチェックサムを計算するトラックチェックサム
 計算手段と、

前記サブグループのそれぞれをデータトラックにそれぞれ対応する少なくとも1つの
 配列に変換する変換手段と、

前記少なくとも1つの配列の各列をエンコードしてデータバイトおよびパリティバ
 イトを含む第1のECCコードワード(C1)を生成する第1レベルエラー訂正コード(E
 CC)エンコード手段であって、前記パリティバイトは前記第1のECCコードワード中
 のデータバイトのうちの少なくとも1つを第1レベル訂正することができる、第1レベル
 エラー訂正コード(ECC)エンコード手段と、

前記少なくとも1つの配列の各行をエンコードしてデータバイトおよびパリティバ
 イトを含む第2のECCコードワード(C2)を生成する第2レベルエラー訂正コード(E
 CC)エンコード手段であって、前記パリティバイトは前記第2のECCコードワード中

のデータバイトのうちの少なくとも1つを第2レベル訂正することができ、第2のECCコードワードを構成する前記データバイトおよびパリティバイトは、所与のコードワード中の全てのバイトのXOR演算の結果をゼロにしなければならないというECCルールに従う、第2レベルエラー訂正コード(ECC)エンコード手段と、

からなり、前記トラックチェックサム計算手段によって適用される前記アルゴリズムとして、前記ECCルールにより要求される前記XOR演算と相関性を持たない前記トラック中のデータバイトの算術加算を選択することにより、前記ECCコードワードのデコード並びに前記トラックチェックサムの再計算および検査の際に、トラックチェックサム障害の原因となるコードワードの誤訂正の確率を高める性質を有するチェックサムを生成する、装置。

10

【請求項2】

前記エラー訂正コードエンコード手段はリードソロモンコーディングを使用する、請求項1に記載の装置

【請求項3】

エンコードされた前記配列を変換し、各トラックについて前記磁気媒体に書き込むための複数のデータフラグメントを生成する手段と、

前記データフラグメントのそれぞれにフラグメントヘッダを提供するためのフラグメントヘッダ手段とを含み、

前記フラグメントヘッダのうちの少なくとも幾つかが、現在のトラックのトラックチェックサムを識別するデータバイトを含む、請求項1～2のうちのいずれか一項に記載の装置。

20

【請求項4】

1グループを構成する各トラックの対応するバイト位置からコードワードを計算するための第3レベルのエラー訂正コードエンコード手段を含む、請求項1～3のうちのいずれか一項に記載の装置。

【請求項5】

複数のデータレコードからなるストリームを磁気媒体上に格納する方法であって、

前記複数のデータレコードを複数のデータバイトからなる複数のグループへとグループ化するステップと、

前記複数のグループのそれぞれを、各サブグループが少なくとも1つのデータトラックに対応する複数のサブグループへと分割するステップと、

各データトラックからの前記データバイトをチェックサムアルゴリズムに適用し、各データトラックについて1以上のトラックチェックサムを計算するステップと、

前記1以上のトラックチェックサムを格納するステップと、

前記サブグループのそれぞれをデータトラックにそれぞれ対応する少なくとも1つの配列に変換するステップと、

30

前記少なくとも1つの配列の各列をエンコードしてデータバイトおよびパリティバイトを含む第1レベルのECCコードワードを形成するステップであって、前記パリティバイトは前記第1レベルのECCコード中のデータバイトのうちの少なくとも1つを第1レベル訂正することができる、第1レベルのECCコードワードを形成するステップと、

40

前記配列の各行をエンコードしてデータバイトおよびパリティバイトを含む第2レベルのECCコードワードを形成するステップであって、前記パリティバイトは前記第2レベルのECCコードワード中のデータバイトのうちの少なくとも1つを第2レベル訂正することができ、ECCコードワードを構成する前記データバイトおよびパリティバイトは、所与のコードワード中の全てのバイトのXOR演算の結果をゼロにしなければならないというECCルールに従う、第2レベルのECCコードワードを形成するステップと、

前記ECCコードワードのデコード並びに前記トラックチェックサムの再計算および検査の際に、前記ECCルールにより要求される前記XOR演算と相関性を持たない前記トラック中のデータバイトの算術加算を選択することにより、トラックチェックサム障害の原因

50

となるコードワードの誤訂正の確率を高めるチェックサムを生成するステップと、
からなる方法。

【請求項 6】

少なくとも前記第 2 レベルのエラー訂正エンコードはリードソロモンコーディングを使用する、請求項 5 に記載の方法。

【請求項 7】

1 グループを構成する複数のトラックの対応するバイト位置からコードワードを計算することにより、第 3 レベルのエラー訂正コードを適用するステップを含む、請求項 5 ~ 6 のうちのいずれか一項に記載の方法。

【請求項 8】

請求項 5 ~ 6 のうちのいずれか一項に記載の方法によって格納されたデータを読み出す方法であって、

テープからデータを読み出すステップと、

該データからトラックチェックサムを抽出するステップと、

コードワードをデコードし、各トラックについてデータバイトを得るステップと、

デコードされた各データバイトについてトラックチェックサムを計算するステップと

、
前記チェックサムが一致しない場合に、失敗フラグを立てるステップと、
からなる方法。

【請求項 9】

前記データは 3 レベルのエラー訂正を有し、前記トラックチェックサムは、前記第 2 レベルのエラー訂正の後、第 2 レベルのコードワード障害を知らせるために使用され、それによって、最終レベルの訂正アルゴリズムが、第 2 レベルのコードワード障害に対応する第 3 レベルのコードワード中のデータバイトを識別し、該データバイトを消跡としてマスクする、請求項 8 に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、磁気媒体上へのデータ格納のための方法装置及び方法に関し、特に少なくとも 2 レベルのエラー訂正を実施するかかる方法及び装置に関する。

【0002】

【従来の技術】

DDS (Digital Data Storage) フォーマットによる磁気テープ上のコンピュータデータ等の大量のデジタルデータを格納することが可能な磁気データ記憶装置に関する要求が絶えず増大している。該フォーマットを用いる DDS リード/ライト機構では、データは、磁気媒体がコーティングされたテープからなる細長い記録媒体上に、1 つ又は 2 つ以上の電磁ヘッドを支持する回転ドラムにより、ヘリカルスキャン方式で記録される。該テープは、2 つのスプール又はリール間に延びる経路に沿ってモータ駆動式キャプスタンにより移動され、及び部分的に前記ドラムの周囲に巻き付けられる。該ドラムのヘッドの回転面は、テープの移動平面に対して所定角をなすよう配置され、これにより、テープの中心線に対して所定角をなして該テープの幅方向に横切って延びる連続する複数のトラックに沿って該テープを各ヘッドが横切るようになっている。該機構は、テープ上への記録に適した信号 (エラー検出及び訂正コードを含む) へとデータをエンコードし、及びかかる信号を調整して記録媒体の特性に最適に適合する形式にするための、適当な回路を備えている。データ読み出しのために、更に別の回路が配設される。該回路は、テープ上に格納されている磁界の変動を導出し、それに対応する信号を導出し、かかる信号を調整して後続の処理のための形式にし、エンコードされたデータをデコードし、及びエラーの検出及び訂正を行うものである。

【0003】

最近開発されたものに DDS3 フォーマットがある (1996 年 6 月に ECMA 規格 ECMA-236 で規定さ

10

20

30

40

50

れたもの、「3.81mm Wide Magnetic Tape Cartridge for Information Interchange - Helical Scan Recording - DDS-3 Format using 125mm Length Tapes (情報交換用3.81mmワイド磁気テープカートリッジ - ヘリカルスキャン記録 - 125m長のテープを用いたDDS-3フォーマット)」)。なお、該引用をもってその内容全体を本書中に包含させたものとする。

【0004】

DDS3では、少なくとも2レベル(C1,C2)及び随意選択的な第3レベル(C3)のエラー訂正コーディング(ECC)を提供するマルチレベルインタリーブ式リードソロモンコード(multilevel interleaved Reed-Solomon code)を用いることにより、エラーチェック及び検出が達成される。更に、読み出し時の最終的なデータチェックを提供するために、トラックチェックサムが、テープ上に書き込むべき各データトラックに対応して生成されてフラグメントヘッダ中に格納され、その幾つかが各トラックに組み込まれる。このため、読み出し時には、このトラックチェックサムアルゴリズムが、テープから読み出されるデータバイトに適用され、その際のトラックチェックサムが該テープへの書き込み時に計算されて格納されたトラックチェックサムと一致しない場合には当該トラックが拒絶される。

【0005】

【発明が解決しようとする課題】

トラックチェックサムは、可能な限り大きな割合で不良トラック(訂正不能なコードワード(codeword)又は誤って訂正されたコードワードを含むもの(以下で説明する))を拒絶するよう機能することが望ましい。既存のトラックチェックサムは十分良好に機能するが、このシステムには意外でありかつ重要な欠点が存在することを本出願人は発見した。即ち、かかるトラックチェックサムは、大部分の誤修正を識別しないものである。本出願人の分析により、データバイト中の「修正」が生じた誤修正されたコードワードは該コードワードのみからなり、トラックチェックサムはこの障害を明らかにするものとはならないことが判明した。この現象が生じる理由は、全バイト(データ及びパリティ)のXOR演算の結果がゼロになるという固有の特性を有するように該コードワードが構成されることにある。DDS3の場合にも、トラックチェックサムは、関連するトラックのデータバイトのXOR演算を行うことにより計算される。

【0006】

良好な(又は誤修正された)コードワードのバイトのXOR演算の結果がゼロであると仮定すると、特定の良好な又は誤修正されたコードワードのパリティバイトのXOR演算結果が例えば二進値Aとなる場合には、データバイトのXOR演算結果もまた二進値Aとならなければならない(これによりデータバイトとパリティバイトとのXOR演算結果がゼロとなる)。この特定のコードワードを誤修正した場合には、該「修正」はデータバイト中にしか含まれず、パリティバイトが変更されないため、該パリティバイトのXOR演算結果は依然と同様に二進値Aとなる。データバイトが変更された場合であっても、誤修正のXOR演算結果はゼロであるという要件を見たすために、該データバイトのXOR演算結果は二進値Aになる。しかし、該データバイトは同一のチェックサム(並びにXOR演算の結果)に全て寄与し、トラックチェックサムは無変更のままとなり、よって、この種の誤修正の場合には、コードワードの固有のXOR特性とチェックサムの計算に用いられる演算との間の相関関係に起因して、トラックチェックサムは、「修正」の全てがデータバイト中に現れる誤修正されたコードワードを明らかにするものとはならない。

【0007】

本出願人は、XOR演算と相互に関係しないようにトラックチェックサムアルゴリズムを修正することにより以前のコードワードにおける誤修正を検出するための信頼性の高い方法を提供することを決定した。この誤修正を高い信頼性で検出する能力はまた、第3レベルC3訂正が実施される場合に重要かつ有利な結果を提供するものとなる。

【0008】

これまで、本出願人は、トラックチェックサムを用いてC2コードワードの障害を識別し、及びそれに従ってかかるコードワードにマークを行おうとしてきた。エラー訂正コーディ

10

20

30

40

50

ングのインタリーブ及びマルチレベル構造のため、コードワードが(3レベルシステムにおいて)C1及びC2レベルで失敗した場合には、配列中のコードワードの位置を知ることにより、次のレベルのコードワードにおける疑わしい複数バイトの位置を識別する。エラー訂正コーディングの構造は、特定のコードワード中のバイトが既知のマッピングに従って後続のコードワード中の位置へとマップすることを意味している。このため、C2ステージで失敗したコードワードをフラグとして使用してC3コードワード中の特定のバイトをC3訂正アルゴリズムに知らせ、これによりそれらがエラーではなく「消跡(erasure)」として扱われるようにすることができる。

【0009】

したがって、完全なC2コードワードが失敗としてマークされる場合には、それに対応する(該コードワードを構成する)データバイト位置を決定して、次のC3レベルに対してエラーの位置が既知となるようにすることが可能である。N個のパリティバイトを有するリードソロモンコードは、「e」エラー及び「v」消跡を訂正することが可能なものである(ここで、 $2e+v \leq N$ であり、消跡は既知の位置における不良バイトである)。典型的なC3訂正コードワードは2つのパリティバイト(即ち $N=2$)を有しており、このため、C3訂正アルゴリズムは単一のエラー($e=1$)又は2つの消跡($v=2$)を訂正することができる。2つの訂正が行われる場合、これはダブルエラー訂正と呼ばれる。したがって、誤修正されたC2コードワードを消跡として高い信頼性でマークする能力は、C3アルゴリズムがダブルエラー訂正を実行できることを意味するものとなる。これまで、高信頼性の誤訂正検出方法の欠如は、C3ステージでのダブルエラー訂正の試行が現実的でないことを意味してきた。

【0010】

したがって、本出願人は、トラックチェックサムにより大幅に増大した信頼性を有する誤修正チェックが提供される方法及び装置を提供する。

【0011】

一態様では、本発明は、磁気媒体上に複数のデータレコードからなるストリームを格納するための装置を提供する。該装置は、

前記複数のデータレコードを複数のデータバイトからなる複数のグループへとグループ化するグループフォーマット手段と、

前記複数のグループの各々を複数のサブグループへと分割するサブグループ処理手段であって、その各サブグループが、1つ又は複数のデータトラックに対応するデータバイトからなる、サブグループ処理手段と、

前記1つのデータトラックについて又は前記複数のデータトラックの各々毎に1つ又は2つ以上のチェックサムを計算するトラックチェックサム計算手段と、

前記各サブグループを少なくとも1つの別個の配列へと変換する変換手段であって、その1つの配列又は複数の配列の各々がデータトラックに1つずつ対応するものである、変換手段と、

前記1つの配列又は複数の配列の各々の列をエンコードしてデータバイト及びパリティバイトからなる第1の(C1)ECCコードワードを提供する、第1のエラー訂正コードエンコード手段と、

前記1つの配列又は複数の配列の各々の行をエンコードしてデータバイト及びパリティバイトからなる第2の(C2)ECCコードワードを提供する、第2のエラー訂正コードエンコード手段とを備えており、

前記トラックチェックサム計算手段が、前記1つのトラック又は複数のトラックの各々毎に別個のトラックチェックサムを計算し、これにより、前記ECCコードワードのデコード時に、データバイト中に誤訂正が存在する誤訂正コードワードが、対応するオリジナルのコードワードと同じ寄与を対応するトラックチェックサムに与えるという実質的にランダムな(無作為の)蓋然性しか有さないようにする、というものである。

【0012】

なお、用語「第1」及び「第2」は、第1のエラー訂正コードエンコード手段が第2のエラー訂正コードエンコード手段の前に動作する必要があることを意図したものではないこ

10

20

30

40

50

とを強調しておく。

【0013】

このように、コードワード生成ルールと相関しないトラックチェックサムアルゴリズムを選択することは、トラックチェックサムが誤訂正の検出に関して従来の訂正アルゴリズムよりも一層信頼性の高いものであることを意味している。これはまた、トラックチェックサムを使用して大幅に向上した訂正率で誤訂正コードワードをマークすることを可能にする。

【0014】

好適には、前記第1及び第2のエラー訂正コードエンコード手段は、それぞれのリードソロモンコードアルゴリズムを適用する。

10

【0015】

トラックチェックサムの生成に使用することが可能なアルゴリズムは多数存在する。⁰にルートを有するリードソロモンコードワードが使用される特定の一例では、トラックチェックサムは、関連するトラックにおけるデータバイトの算術加算をベースとして計算することが可能である。

【0016】

コードワード及びチェックサムは、テープへの書き込みに先立って様々な方法で処理することが可能であるが、前記磁気媒体上に書き込むための複数のデータフラグメントを各トラック毎に提供するように前記エンコードされた配列を変換する手段と、前記データフラグメントの各々毎にフラグメントヘッダを提供するフラグメントヘッダ手段とを備えており、前記フラグメントヘッダの内の少なくとも幾つかが、現在のトラックに関する対応するトラックチェックサムを識別するデータバイトを含んでいることが装置にとって好ましい。

20

【0017】

誤訂正の検出のための一層信頼性の高いトラックチェックサム方式を提供することにより、第3レベルでのダブルエラー訂正が容易化される。したがって、本装置は、好適には、1グループを形成するトラックの各々にまたがるそれぞれの対応するバイト位置からコードワードを計算する第3のエラー訂正コードエンコード手段を備えるものとなる。

【0018】

本発明はまた、データレコードストリームを磁気媒体上に格納するための方法へと拡張される。該方法は、

30

前記データレコードを複数のデータバイトからなる複数のグループへとグループ化し、前記複数のグループの各々を複数のデータバイトからなる複数のサブグループへと分割し、該サブグループの各々が1つ又は複数のデータトラックに対応するものであり、

前記1つのデータトラックについて又は前記複数のデータトラックの各々毎に1つ又は複数のトラックチェックサムを計算して格納し、

前記サブグループの各々を少なくとも1つの個々の配列へと変換し、該配列の各々がデータトラックに1つずつ対応するものであり、

前記1つの配列又は複数の配列の各々の列をエンコードしてデータバイト及びパリティバイトからなる第1のエラー訂正コードワードを形成し、

40

前記1つの配列又は複数の配列の各々の行をエンコードしてデータバイト及びパリティバイトからなる第2のエラー訂正コードワードを形成し、

前記トラックチェックサムを計算するステップが、前記1つのトラックについて又は複数のトラックの各々について個々のトラックチェックサムを計算して、前記エラー訂正コードワードのデコード時に、データバイト中に誤訂正が存在する誤訂正コードワードが、対応するオリジナルのコードワードと同じ寄与に対応するトラックチェックサムに与えるという実質的にランダムな（無作為の）蓋然性しか有さないようにする、というものである。

【0019】

本発明はまた、上記方法に従って格納されたデータを読み出すための方法へと拡張される

50

ものである。該方法は、テープからデータを取り出し、該データから前記トラックチェックサムを抽出し、コードワードをデコードして各トラック毎のデータバイトを獲得し、該デコードされたデータバイトについてトラックチェックサムを計算し、それらチェックサムが一致しない場合に障害の発生を知らせる、という各ステップを有するものである。

【 0 0 2 0 】

好適には、前記データは、3レベルのエラー訂正を含み、前記トラックチェックサムが第2レベルのエラー訂正の後に使用されて第2のコードワード障害が知らされ、これにより、第2レベルのコードワード障害に対応する第3レベルのコードワード中のあらゆるバイトが識別され及び第3レベルのエラー訂正アルゴリズムに対して消跡としてマークされる。

10

【 0 0 2 1 】

本発明について上述してきたが、それらは、上記の本発明の特徴の組み合わせ又は以下で説明するように拡張することが可能である。

【 0 0 2 2 】

本発明は、様々な態様で実施することが可能なものであり、ここで図面を参照してかかる実施例を詳細に説明することとする。

【 0 0 2 3 】

【課題を解決するための手段】

【 0 0 2 4 】

【発明の実施の形態】

20

【 0 0 2 5 】

【実施例】

以下に示すテープ記憶システムの実施例は、DDS3規格（ECMA236に大要が説明されている）をベースとするものであり、第3レベル(C3)訂正アルゴリズムと、修正されたトラックチェックサムアルゴリズムとを含むものである。該修正されたトラックチェックサムアルゴリズムは、より信頼性の高いC2誤訂正標識であるチェックサムを提供するものである。該修正されたアルゴリズムに従って決定されるトラックチェックサムは、C2誤訂正（C3誤訂正が使用されない場合）の後、又はC3誤訂正の後に基本データの最終チェックとして使用することが可能なものである。以下でより詳細に説明するように、C3誤訂正が使用されない場合には、該修正されたトラックチェックサムアルゴリズムを使用することにより、最終チェックとしてC2誤訂正を検出するための遙かに信頼性の高い方法が提供される。またC3誤訂正が使用される場合には、C2誤訂正のチェック及びマークを高い信頼性で実行する能力は、C2修正不能コードワード及びC2誤訂正の両方をC3に対して消跡としてマークしてC3訂正コードの全能力を使用することが可能になることを意味するものとなる。

30

【 0 0 2 6 】

ここで図1を参照する。テープ上に格納されるべき基本データストリームは、基本グループモジュール10により384,296バイトの基本グループへとグループ化される。該基本グループモジュール10に続き、C3訂正プロセッサ14及びトラックチェックサム生成器20が配設されている。便宜上、C3訂正プロセッサ14及びトラックチェックサム生成器20は、サブグループにより導入されるバイト命名法（byte nomenclature）を用いて簡単に説明するが、かかる構成要素については以下で一層詳細に説明することとする。

40

【 0 0 2 7 】

基本グループが完成すると、該基本グループはG1モジュール12により、22個のG1サブグループへと分割される。各G1サブグループは、0~17,467まで番号付けされた17,468バイトのサイズを有するものである。各G1サブグループはまた、1~22（図2参照）の範囲の実行番号を1つずつ有している。エラー訂正コード(ECC3)プロセッサ14は、22個のG1サブグループの各々からデータを導出して23番目のG1サブグループを形成する。エラー訂正コードC3はGF(2⁸)リードソロモンコード(46,44,3)である。GF(2⁸)での計算は次のように規定されるべきである。

$$G(x)=x^8+x^4+x^2+1$$

50

GF(2⁸)における基本要素(primitive element) は00000010である。

【 0 0 2 8 】

ECC3のインタリーブ深さ(interleave depth)は1トラックである。ECCバイトは次式を満たす。

$$H_R \times V_R = 0$$

前記トラックチェックサム生成器の多項式は次の通りである。

【 0 0 2 9 】

【 数 1 】

$$G_R(X) = \prod_{i=0}^{i=1} (X - \alpha^i)$$

10

$$H_R \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ \alpha^{45} & \alpha^{44} & \alpha^{43} & \dots & \alpha^2 & \alpha & 1 \end{bmatrix}$$

$$V_R = \begin{bmatrix} D_{n,1} \\ D_{n+8734,1} \\ D_{n,2} \\ D_{n+8734,2} \\ \dots \\ D_{n+8734,22} \\ R_{n,23} \\ R_{n+8734,23} \end{bmatrix}$$

20

【 0 0 3 0 】

ここで、

$$n = 0, 1, 2, \dots, 8733$$

$D_{x,y}$ = G1サブグループにおけるユーザデータバイト番号

y = G1サブグループ番号

$R_{x,23}$ x = ECC2 G2 サブグループにおけるパリティバイト番号である。

30

【 0 0 3 1 】

エラー訂正コードC3は、記録されているデータグループにおける任意の2つの不良トラックを訂正することができるものである。

【 0 0 3 2 】

次いで、各G1サブグループのバイトがG2モジュール16によりランダム化(randomise)されてG2サブグループが形成される。該G2サブグループにおいては、全てのバイトは依然として $D_0 \sim D_{17467}$ に番号付けされている。それらの順番はG1サブグループの場合と同じである。次いで、G3モジュール18がG2サブグループを操作して、17468バイトの各G2サブグループが、(図3に示すように)G3サブグループのトラックAにおいてG2サブグループのグループバイト $D_0 \sim D_{8733}$ 、G3サブグループのトラックBにおいてG2サブグループのグループバイト $D_{8734} \sim D_{17467}$ 、と配列されるようにする。各トラックにおいてバイトがワードへと割り当てられる。偶数番号が付されたバイトが下位バイトに割り当てられる一方、奇数番号が付されたバイトが上位バイトに割り当てられる。各トラックの最初のワード(ワード番号0)は、論理フレーム識別子(LFID)及びデータフォーマット識別子(DFID)として知られる2バイトを含む。

40

50

【 0 0 3 3 】

DFIDは、DDSフォーマットを示すものである。LFIDは、フレーム番号と23番目のフレーム（C3フレーム）の有無とを示すものである。

【 0 0 3 4 】

次いで、G3サブグループの各バイトが、
そのトラック（A又はB）、
そのバイト名（上位又は下位）
そのワード番号（0～4367）
により識別される。

【 0 0 3 5 】

以下の説明では次の表記法を使用することとする。

【 0 0 3 6 】

A_{i_l} は、 i 番目のワードにおけるトラックAの下位バイトにより識別されるバイトを示す。

【 0 0 3 7 】

A_{i_u} は、 i 番目のワードにおけるトラックAの上位バイトにより識別されるバイトを示す。

【 0 0 3 8 】

B_{i_l} は、 i 番目のワードにおけるトラックBの下位バイトにより識別されるバイトを示す。

【 0 0 3 9 】

B_{i_u} は、 i 番目のワードにおけるトラックBの上位バイトにより識別されるバイトを示す。

【 0 0 4 0 】

トラックチェックサム生成器20は、リードソロモンコードワード生成ルールとの相関性を有さないアルゴリズムに従ってトラックチェックサムを生成する。このため、該トラックチェックサムアルゴリズムは、各トラック毎にチェックサムとして算術加算を使用する。1トラック中のバイトが加算されて16ビットの結果が形成され、この場合、16ビットを越える桁上げは却下される（即ち、該加算はモジュロ 2^{16} で計算される）。トラックAについては、トラックチェックサム（TCS(A)）は次式により与えられる。

【 0 0 4 1 】

【 数 2 】

8733

$$TCS(A) = [LFID] + [DFID] + \sum_{i=0}^{8733} D(i)$$

【 0 0 4 2 】

またトラックBについては次式の通りである。

【 0 0 4 3 】

【 数 3 】

17467

$$TCS(B) = [LFID] + [DFID] + \sum_{i=8734}^{17467} D(i)$$

【 0 0 4 4 】

LFID及びDFIDは、論理フレームID及びデータフォーマットIDであり、G3サブグループに関連する上述のような監視用バイトである。1トラック内のあらゆるC2コードワードにおける基本データバイトは全て、同一のトラックチェックサムに寄与するものである、ということに留意されたい。トラックチェックサムの計算は、算術加算であり、それ故、C2パリティバイトの計算に使用されるXOR演算とは全く相関性がない。

【 0 0 4 5 】

16ビットの算術チェックサムの場合、C2誤訂正が本質的にランダムであるとき、トラック

10

20

30

40

50

チェックサムが変化しない蓋然性（即ち $1 / 2_{16}$ ）は、「 1.5×10^{-6} 」となる。

【 0 0 4 6 】

次いで、各G3サブグループがG4モジュール22によってG4サブグループ（C1,C2コードワード生成を含む）へと変換される。各G4サブグループは後述するように2つの配列対から構成される（図4参照）。符号、フラグメント番号、及びシリアル番号が、次式を用いて各バイトに割り当てられる。

符号 : $(-1)^a$

フラグメント番号 : $i(\bmod 78)+9$

シリアル番号 : $(2(u+\text{Integer}(i/78))-(\text{Integer}(i/78))(\bmod 2))$

Integer=商の整数部分

$i = 0 \sim 4367$

$a = 0$ (A_{1u}, A_{1l} バイトの場合)

$a = 1$ (B_{1u}, B_{1l} バイトの場合)

$u = 0$ (A_{1u}, B_{1u} バイトの場合)

$u = 1$ (A_{1l}, B_{1l} バイトの場合)

ここで図4を参照する。同図には、トラックA及びBに対応してプラス及びマイナスという名称が付された2つの配列が示されている。各配列において、各バイトは、そのフラグメント番号（0～95）とそのシリアル番号（0～123）とで識別される。

【 0 0 4 7 】

上記の式を適用することにより、図4に示す配列の非ハッチング部分にデータバイトが分配される。次いで、該テーブルの非ハッチング部分におけるデータバイトが、該配列内に既に割り当てられているバイトにわたって計算された2つのリードソロモンエラー検出及び訂正コードC1,C2を用いてエンコードされる。C1コードワードは列方向に延び、C2コードワードは行方向に延びる。

【 0 0 4 8 】

C2バイトは、0～8又は87～95の範囲のフラグメント番号を有する各フラグメントにおける0～111の範囲のシリアル番号を有するバイト位置について、それ以外のフラグメントの全てにおける同一のシリアル番号を有するバイトから計算される。

【 0 0 4 9 】

次いで、C1バイトは、全てのフラグメントにおける112～123の範囲のシリアル番号を有するバイト位置について、それと同一のフラグメントにおけるそれ以外の全てのバイトから計算される。0～8又は87～95の範囲のフラグメント番号を有するフラグメントでは、C1バイトは、以前に計算されたC2バイトから計算される。C1は、 $GF(2^8)$ リードソロモンコード（62,56,7）である。この表現は、コードワードが62バイト長であり、その内の56バイトがデータバイトであり、及び7バイトの区切り(separation)を有していることを示している。C2は、 $GF(2^8)$ リードソロモンコード（32,26,7）である。

【 0 0 5 0 】

$GF(2^8)$ における計算は次式により規定される。

【 0 0 5 1 】

$G(x)=x^8+x^4+x^3+x^2+1$

$GF(2^8)$ における基本要素 は、00000010である。

【 0 0 5 2 】

C1のインタリーブ深さは2バイトであり、このため、一配列のあらゆる列においては、偶数のシリアル番号を有するデータバイトは、該列のC1コードワードの内の1つに属し、奇数のシリアル番号を有するデータバイトは、同列における他のC1コードワードに属する。C2のインタリーブ深さは3フラグメントであり、このため、あらゆる行において第3のフラグメントは同じコードワードに属し、1行中に3つのC2コードワードが存在する。エラー訂正コード（ECC）バイトは、次の関係を満たす。

【 0 0 5 3 】

$H_p \times V_p = 0$

10

20

30

40

50

$$H_Q \times V_Q = 0$$

生成器の多項式は次の通りである。

【 0 0 5 4 】

【 数 4 】

$$G_P(X) = \prod_{i=0}^{i=5} (X - \alpha^i)$$

$$G_Q(X) = \prod_{i=0}^{i=5} (X - \alpha^i)$$

10

$$H_P = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ \alpha^{61} & \alpha^{60} & \alpha^{59} & \dots & \alpha^2 & \alpha & 1 \\ \alpha^{122} & \alpha^{120} & \alpha^{118} & \dots & \alpha^4 & \alpha^2 & 1 \\ \alpha^{183} & \alpha^{180} & \alpha^{177} & \dots & \alpha^6 & \alpha^3 & 1 \\ \alpha^{244} & \alpha^{240} & \alpha^{236} & \dots & \alpha^8 & \alpha^4 & 1 \\ \alpha^{50} & \alpha^{45} & \alpha^{40} & \dots & \alpha^{10} & \alpha^5 & 1 \end{bmatrix}$$

20

$$H_Q = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ \alpha^{31} & \alpha^{29} & \alpha^{28} & \dots & \alpha^2 & \alpha & 1 \\ \alpha^{62} & \alpha^{60} & \alpha^{58} & \dots & \alpha^4 & \alpha^2 & 1 \\ \alpha^{93} & \alpha^{90} & \alpha^{87} & \dots & \alpha^6 & \alpha^3 & 1 \\ \alpha^{124} & \alpha^{120} & \alpha^{116} & \dots & \alpha^8 & \alpha^4 & 1 \\ \alpha^{155} & \alpha^{150} & \alpha^{145} & \dots & \alpha^{10} & \alpha^5 & 1 \end{bmatrix}$$

30

【 0 0 5 5 】

(V_P 及び V_Q は以降で示す)

ここで、

P_{ij} = C1 バイト

Q_{ij} = C2 バイト

i = フラグメント番号

j = シリアル番号

であり、C1 の場合、

$k = 0, 1, \dots, 95$

$l = 0, 1$

40

であり、 $k = 0, 1, \dots, 8$ 又は $k = 87, 88, \dots, 95$ の場合には、 V_P 中の D_{ij} は Q_{ij} と読まれ、

C2 の場合には、

$m = 0, 1, 2$

$n = 0, 1, \dots, 111$

である。

【 0 0 5 6 】

【 数 5 】

$$\begin{array}{c}
 V_P = \begin{bmatrix} D_{k,1} \\ D_{k,1+2} \\ D_{k,1+4} \\ D_{k,1+6} \\ D_{k,1+8} \\ D_{k,1+10} \\ D_{k,1+12} \\ D_{k,1+14} \\ D_{k,1+16} \\ D_{k,1+18} \\ D_{k,1+20} \\ D_{k,1+22} \\ D_{k,1+24} \\ D_{k,1+26} \\ D_{k,1+28} \\ D_{k,1+30} \\ \dots \\ D_{k,1+94} \\ D_{k,1+96} \\ D_{k,1+98} \\ D_{k,1+100} \\ D_{k,1+102} \\ D_{k,1+104} \\ D_{k,1+106} \\ D_{k,1+108} \\ D_{k,1+110} \\ P_{k,1+112} \\ P_{k,1+114} \\ P_{k,1+116} \\ P_{k,1+118} \\ P_{k,1+120} \\ P_{k,1+122} \end{bmatrix}
 \end{array}
 \begin{array}{c}
 V_Q = \begin{bmatrix} Q_{m,n} \\ Q_{m+3,n} \\ Q_{m+6,n} \\ D_{m+9,n} \\ D_{m+12,n} \\ D_{m+15,n} \\ D_{m+18,n} \\ D_{m+21,n} \\ D_{m+24,n} \\ D_{m+27,n} \\ D_{m+30,n} \\ D_{m+33,n} \\ D_{m+36,n} \\ D_{m+39,n} \\ D_{m+42,n} \\ D_{m+45,n} \\ D_{m+48,n} \\ D_{m+51,n} \\ D_{m+54,n} \\ D_{m+57,n} \\ D_{m+60,n} \\ D_{m+63,n} \\ D_{m+66,n} \\ D_{m+69,n} \\ D_{m+72,n} \\ D_{m+75,n} \\ D_{m+78,n} \\ D_{m+81,n} \\ D_{m+84,n} \\ D_{m+87,n} \\ Q_{m+90,n} \\ Q_{m+93,n} \end{bmatrix}
 \end{array}
 \begin{array}{c}
 10 \\
 20 \\
 30 \\
 40
 \end{array}$$

【 0 0 5 7 】

G4サブグループの各フラグメントは、ヘッダプロセッサ24により変換されて、8バイトのヘッダを前置することにより132バイトのメインデータフラグメントとなる。該ヘッダは、図5に示すような制御及び監視データを含むものであるが、サブコードバイト番号0～3について言及する必要がある。該サブコードバイトは、16個の4バイトパックアイテムとして配列されたサブコード情報を含むものである。各トラックは96個のフラグメントを含み、該フラグメントの各々は、1つのパックアイテムを組み込んだそれ自体のヘッダを有しており、各パックアイテムが1トラックについて6回繰り返されるようになっている。該実施態様では、図6に示すように、トラックA,Bについてパックアイテム番号5がト

ラックチェックサム専用割り当てられる。

【0058】

したがって、本装置は、最初に基本データを基本グループへとグループ化し、次いで該基本グループが22個のサブグループへと分割され、それらの各サブグループが1フレームでテープ上に書き込まれる。各フレームは2トラック(A,B)からなり、該トラックの各々は、基本データと、それ以外の当該フォーマットにより付与された監視及び誤差訂正用データとを含んでいる。テープ上の1グループは、22個のフレームと追加のエラー訂正コードフレームとから構成される(C3訂正が実施される場合)。

【0059】

C3パリティバイトは、1グループ内で44トラックにわたり計算される。この場合、各トラックから1バイトが取得されて、2つのパリティバイトを有するC3コードワードが生成される。該C3パリティバイトは、テープ上の1グループに付与される上述の23番目のフレームに格納される。C3の使用は随意選択的なものである。

10

【0060】

C1,C2,C3コードについての生成器の多項式は、 0 での累乗根を有しており、このため、良好なコードワードでは、該コードワード中の全バイトのXORの結果として常にゼロが得られることになる、ということに留意されたい。

【0061】

したがって、各基本グループは、22個(又はC3訂正が存在する場合には23個)のG4サブグループへと変換され、該G4サブグループの各々は、96個のフラグメントから各々構成される2つのトラックを備えたものとなる。

20

【0062】

次いで、(ヘッダを有する)該フラグメントが、8:10エンコーダ26において8ビット 10ビットエンコーディングを受けた後、従来の態様でテープに書き込まれる。

【0063】

データの読み出し時には、マルチレベルインタリーブ式リードソロモンエラー訂正コーディングを利用して、読み出されるデータにおけるエラーの検出及び訂正が行われる。

【0064】

リードソロモンコードは、テープから読み出されるコードワード内のエラーの訂正を可能にする。訂正の際に、N個のパリティバイトを有するリードソロモンコードワードは、 $2e+v$ N(eはランダムエラーの数、vはコードワード内の既知の位置におけるエラー(消跡)の数)である場合には常に訂正される。また、 $2e+v > N$ である場合には、コードワードは訂正不能であり誤訂正される可能性がある。誤訂正は、訂正中には検出することができない。

30

【0065】

リードソロモンコードワードを訂正するためのアルゴリズムは、多くの標準的な書物、例えば、Richard E Blahut著の「Theory and Practice of Error Control Codes」(ISBN 0-201-10102-5, Addison-Wesley Publishing Company Inc.)に記載されている。本引用をもってその開示内容を本書に取り込んだものとする。

【0066】

適当なアルゴリズムの例として、Euclidean Division アルゴリズムがある。

40

【0067】

かかるアルゴリズムがコードワードに適用された場合、その結果は、コードワードが良好であり訂正は行わないことを示す「適正(correct)」、コードワードがエラーを有しており当該アルゴリズムにより訂正されることを示す「訂正可能」、訂正アルゴリズムがコードワードを訂正して有効ではあるが誤ったコードワードが生成されたことを示す「誤訂正」、又はコードワードが不正でありアルゴリズムにより不正なものとして識別されたことを示す「訂正不能」となる。誤訂正及び訂正不能は「失敗」と呼ばれる。

【0068】

本出願人によるDDS3の実施においては、C1訂正不能コードワードはC2訂正については消跡としてマークされる。

50

【 0 0 6 9 】

ここで図 4 を参照する。C1コードワードはインタリーブ深さ「2」を有し各列内に延びるものであることが想起されよう。したがって、C2訂正アルゴリズムが例えばフラグメント 9 において最初のコードワードを訂正不能なものとして識別した場合、これは、フラグメント 9 中の偶数シリアル番号におけるバイトをそれに対応する（訂正不能が生じる）C2コードワードにおいて消跡としてマークすることができることを意味する。

【 0 0 7 0 】

また、誤訂正は次のレベルで検出される（及び訂正される）ことになる。このため、該マルチレベルインタリーブ式エラー訂正方式は、テープからのデータの読み出し時におけるエラーに対して高い許容性を提供するものとなる。

10

【 0 0 7 1 】

トラックチェックサムは、C2（C3の不使用时）又はC3訂正の後に基本データの最終チェックとして用いられる。テープからデータを読み出す装置により不正な基本データが検出されることが重要である。

【 0 0 7 2 】

C3が使用されない場合には、訂正不能なC2コードワードは、C2訂正を行う際に容易に検出することができる。しかし、C2誤訂正は、C2訂正中に検出することができず、トラックチェックサムにより検出されるのが望ましい。トラックチェックサムの計算に関する上述のアルゴリズムは、リードソロモンコードワード生成ルールとは全く相関性がなく、このためC2誤訂正を検出するための信頼性の高い方法を提供するものとなる。

20

【 0 0 7 3 】

また、C3が使用される場合には、C2訂正不能コードワード及びC2誤訂正コードワードをC3に対して消跡としてマークしてC3コードの全能力を訂正に利用できるようにすることが望ましい。この場合も、トラックチェックサムアルゴリズムがリードソロモンコードワード生成ルールとの相関性を全く有さないため、これを達成することが可能となる。

【 0 0 7 4 】

したがって、テープから読み出されたデータが、デコーダ28による10:8デコーディングを受けて、ヘッダを含むデータフラグメントが得られる。該ヘッダが再フォーマットモジュール30により抽出され、トラックチェックサム値がトラックチェックサムチェッカー32により読み出される。次いでデータフラグメントがG4再フォーマッタ34によりG4サブグループフォーマットへと戻され、次いで該G4サブグループがC1及びC2訂正アルゴリズムを受けて、C1及びC2コードワードにおける訂正可能なエラー及びその場所が検出される。これには、上述のようにC1訂正不能をC2アルゴリズムに対して消跡としてマークして6つのC2パリティバイトから得ることができる能力を良好に利用することが含まれる。

30

【 0 0 7 5 】

C2訂正の後、C3訂正ステージが存在しない場合には、コードワードが再フォーマットモジュール36～40によりG1サブグループフォーマットへと戻される。トラックチェックサムが再計算され、これが、トラックチェックサムチェッカー32により、ヘッダから取り出されたトラックチェックサムと比較されて、C2誤訂正（C2訂正アルゴリズムによりマークされたC2訂正不能）が識別される。

40

【 0 0 7 6 】

C1及びC2訂正に続いてC3訂正が存在する場合には、トラックチェックサムチェッカーがC2誤訂正をC3訂正アルゴリズムモジュール42に対して消跡としてマークし、これにより、2つの消跡がマークされている場合に、C3訂正アルゴリズムがダブルエラー訂正を行うのが容易となり、C3コードの全能力を訂正に用いることが可能となる。次いで、トラックチェックサムを、最終データチェックとしてC3訂正済データに用いることが可能となる。

【 0 0 7 7 】

エラー訂正の失敗

以下の（ランダムエラーについてののみ有効な）分析は、C2誤訂正を高い信頼性で消跡としてマークすることでC2ダブルエラー訂正を可能にすることにより達成される訂正率の改善

50

を示すものである。

【 0 0 7 8 】

Pは、1チャンネルにおけるランダムビットエラーの確率として規定され、Pc1は、良好なC1訂正の確率として規定されるものである。

【 0 0 7 9 】

【数 6】

$\overline{P_{c1}} = 1 - P_{c1}$, C1失敗(訂正不能又は誤訂正となり得る失敗)の確率

Pc2 及び $\overline{P_{c2}}$ も同様に定義される

10

【 0 0 8 0 】

C2失敗率の推定値は次式により求めることができる。

【 0 0 8 1 】

【数 7】

$$\frac{\overline{P_{c2}}}{P_{c2}} = \left[\frac{32}{7} \right] \frac{\overline{P_{c1}}}{P_{c1}}^7 = \left[\frac{32}{7} \right] \left[\frac{62}{4} \right]^7 P^{28}$$

【 0 0 8 2 】

ここで、消跡はC1訂正では使用されない(このため62バイトのC1コードワード中の3バイトしか補正できない)ものと仮定する。表1は様々な値のPについてPc1を示したものである。C1訂正率、即ち、C1コードワードが1つ又は2つ以上のエラーを含む確率は、同表に示すとおりである。

【 0 0 8 3 】

【表 1】

C1エラー率に対するC2失敗の確率

P	C1エラー率	$\overline{P_{c2}}$
10^{-2}	0.46	10×10^{-9}
5×10^{-3}	0.27	2.1×10^{-18}
2×10^{-3}	0.12	1.5×10^{-29}
10^{-3}	0.06	5.7×10^{-38}

30

40

【 0 0 8 4 】

C3シングルエラー訂正を用いる場合にはC3失敗率は次式により与えられる。

【 0 0 8 5 】

【数 8】

$$\frac{\overline{P_{c3}}}{P_{c3}} (1) = \left[\frac{46}{2} \right] \frac{\overline{P_{c2}}}{P_{c2}}^2$$

【 0 0 8 6 】

50

またC3ダブルエラー訂正を用いる場合にはC3失敗率は次式により与えられる。

【 0 0 8 7 】

【 数 9 】

$$\overline{P_{c3}}(2) = \left[\frac{46}{3} \right] \overline{P_{c2}}^3$$

【 0 0 8 8 】

前記表 1 の最上行の値を適用することにより、上記式からC2失敗に関する値が次表の通り与えられる。

【 0 0 8 9 】

【 表 2 】

C3失敗の確率

p	C1エラー率	Pc2	Pc3 (1)	Pc3 (2)
10 ⁻²	0.46	1*10 ⁻⁹	3.3*10 ⁻¹⁶	2.8*10 ⁻²⁴

【 0 0 9 0 】

次いで、C3ダブルエラー訂正は、失敗率における少なくとも 8 桁の大きさの改善を与え（表 1 の他の行はこれと同等の良好な改善を示すものとなり）、これは明らかに重要な利益である。

【 0 0 9 1 】

上述のアルゴリズムによる全体的な利益を計算するために、C2誤訂正の確率を決定する必要があるが、これはチェックサムからは取り出されない。これは、C3へ渡されるC2消跡フラグにおける「エラー率」である。上記表 1 は、C2失敗に関する幾つかの望ましくない場合を与えるものであるが、そのうちの何パーセントが誤訂正であるかを推定する必要がある。これは、計算が困難なものであるが、誤訂正において生成されたエラー位置が0～255の範囲にわたってランダムである場合には、4 が0～31（32はC2コードワードの長さ）の範囲内にある可能性は、 $(32/256)^4 = 2.4 \times 10^{-4}$ となる。これに基づき、現行（従来）のトラックチェックサムアルゴリズムと本書で説明するアルゴリズムとの比較を表 3 に示す。

【 0 0 9 2 】

【 表 3 】

C2フラグエラー率

P	C1エラー率	現行(従来の) チェックサム	新規のチェックサム
10 ⁻²	0.46	6*10 ⁻¹⁴	2*10 ⁻¹⁸

【 0 0 9 3 】

したがって、上記構成は、既存の方式よりも少なくとも 4 桁の大きさの改善を提供するものである。該フォーマットでは特別なバイトは必要とされない。これはハードウェアで容易に実施できる。16ビット加算器は多数のゲートを使用することがなく、インタリーブも存在しないからである。

【図面の簡単な説明】

【図 1】 本発明による修正されたDDS3フォーマットを用いた磁気テープに対する情報の格納及び読み出しの概要を表すブロック図である。

【図 2】 DDS3フォーマットで実施されたデータフォーマットにおける基本グループから分離されたG2サブグループを示す説明図である。

【図 3】 G1サブグループのランダム化及び再配置の後に得られるG3サブグループを示す説明図である。

【図 4】 G3サブグループから得られるG4サブグループの構造を示す説明図である。

【図 5】 メインデータフラグメントの構造を示す説明図である。

【図 6】 トラックチェックサムを含むバックアイテムの構造を示す説明図である。

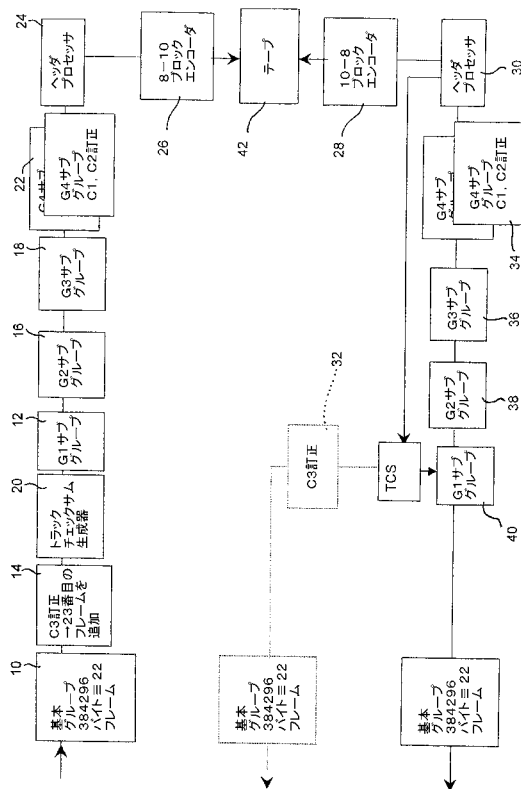
10

【符号の説明】

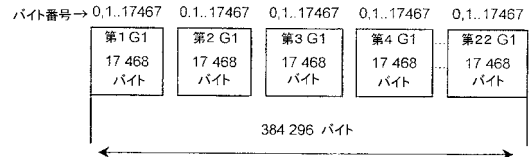
- 10 基本グループモジュール
- 12 G1モジュール
- 14 C3訂正プロセッサ
- 16 G2モジュール
- 18 G3モジュール
- 20 トラックチェックサム生成器
- 22 G4モジュール
- 24 ヘッダプロセッサ
- 26 8:10エンコーダ
- 28 10:8デコーダ
- 30 再フォーマットモジュール
- 32 トラックチェックサムチェッカー
- 34 G4再フォーマッタ
- 36 ~ 40 再フォーマットモジュール
- 42 C3訂正アルゴリズムモジュール

20

【 図 1 】



【 図 2 】



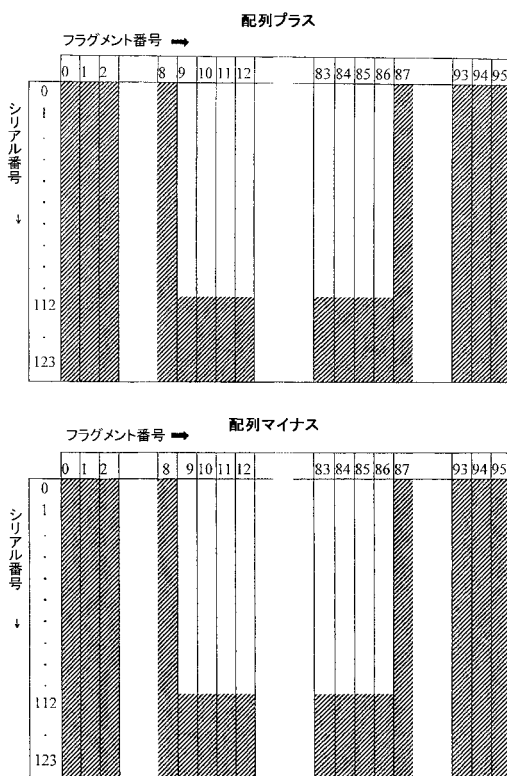
【 図 3 】

トラック名	A								B									
	下位				上位				下位				上位					
	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1		
バイト名	87654321				87654321				87654321				87654321					
ビット番号	87654321				87654321				87654321				87654321					
ヘッダ	0	0	0	0	DF-ID	LF-ID				0	0	0	0	DF-ID	LF-ID			
	1	D ₀				D ₁				D _{8 734}				D _{8 735}				
	2	D ₂				D ₃				D _{8 736}				D _{8 737}				
	3	D ₄				D ₅				D _{8 738}				D _{8 739}				
					
					
					
	4 366	D _{8 730}				D _{8 731}				D _{17 464}				D _{17 465}				
	4 367	D _{8 732}				D _{8 733}				D _{17 466}				D _{17 467}				

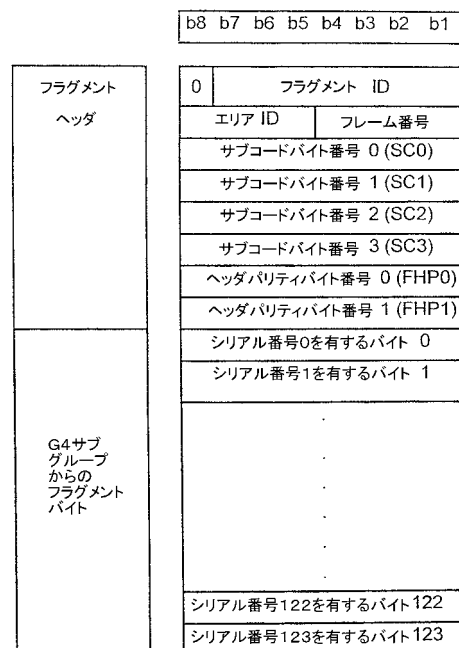
↑

ワード番号

【 図 4 】



【 図 5 】



【 図 6 】

	b8	b7	b6	b5	b4	b3	b2	b1
SC 0	トラックAチェックサムへ (MSB)							
SC 1	トラックAチェックサムへ (LSB)							
SC 2	トラックBチェックサムへ (MSB)							
SC 3	トラックBチェックサムへ (LSB)							

フロントページの続き

(51)Int.Cl. F I
G 0 6 F 11/10 3 1 0 B

(72)発明者 モーレイ, ステファン
イギリス国ブリストル・ピーエス 3 5 ・ 2 ピーエックス, ソーンバリー, ホブキンス・クローズ・
1 7

(72)発明者 ウィリアムズ, ロバート
イギリス国ブリストル・ピーエス 4 ・ 4 ジェイユー, プリスリングトン, ジーン・ロード・4 4

(72)発明者 尾崎 真也
東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内

(72)発明者 ワトキンス, マーク, ロバート
イギリス国ブリストル・ピーエス 6 ・ 7 エスアール, ウェストバーリー・パーク, ハルスバーリー
・ロード・1 2

(72)発明者 廣瀬 寿幸
東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内

審査官 堀 洋介

(56)参考文献 特開平 0 7 - 2 9 5 8 4 6 (J P , A)
特表平 0 6 - 5 1 0 1 4 7 (J P , A)
特開平 0 9 - 0 0 7 3 1 2 (J P , A)
特開平 0 9 - 0 4 5 0 1 2 (J P , A)
特開平 0 9 - 0 7 3 7 3 8 (J P , A)
Standard ECMA - 236, 1996年 6月, p. 37 - 68

(58)調査した分野(Int.Cl., DB名)

G11B 20/18

G06F 3/06

G06F 11/10