

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5276173号  
(P5276173)

(45) 発行日 平成25年8月28日(2013.8.28)

(24) 登録日 平成25年5月24日(2013.5.24)

(51) Int.Cl.

H03M 13/19 (2006.01)  
G 11 B 20/10 (2006.01)

F 1

H03M 13/19  
G 11 B 20/10 341Z  
G 11 B 20/10 321Z

請求項の数 10 (全 35 頁)

(21) 出願番号 特願2011-522963 (P2011-522963)  
 (86) (22) 出願日 平成20年12月12日 (2008.12.12)  
 (65) 公表番号 特表2012-500513 (P2012-500513A)  
 (43) 公表日 平成24年1月5日 (2012.1.5)  
 (86) 國際出願番号 PCT/US2008/086537  
 (87) 國際公開番号 WO2010/019169  
 (87) 國際公開日 平成22年2月18日 (2010.2.18)  
 審査請求日 平成23年12月12日 (2011.12.12)  
 (31) 優先権主張番号 61/089,297  
 (32) 優先日 平成20年8月15日 (2008.8.15)  
 (33) 優先権主張国 米国(US)

(73) 特許権者 508243639  
 エルエスアイ コーポレーション  
 アメリカ合衆国 95131 カリフォルニア, サンホセ, リッダー パーク ドライヴ 1320  
 (74) 代理人 100094112  
 弁理士 岡部 譲  
 (74) 代理人 100085176  
 弁理士 加藤 伸晃  
 (74) 代理人 100104352  
 弁理士 朝日 伸光  
 (74) 代理人 100128657  
 弁理士 三山 勝巳  
 (74) 代理人 100160967  
 弁理士 ▲濱▼口 岳久

最終頁に続く

(54) 【発明の名称】ニア・コードワードのROMリスト復号

## (57) 【特許請求の範囲】

## 【請求項1】

グラフベースの符号を使用して符号化された符号化されたデータを復号する方法であつて、

(a) 候補の復号された符号語を生成するために、前記符号化されたデータを復号することと、

(b) 前記候補の復号された符号語が復号された正しい符号語ではない場合に、前記復号された正しい符号語を生成することを試みるためにトラッピング・セット(TS)-ROMリスト復号方法を実行することであつて、

前記候補の復号された符号語は、少なくとも1つの満足されないチェックノードを有し、満足されないチェックノードは、パリティ検査に合格しないチェックノードであり、

前記TS-ROMリスト復号方法は、ROMメモリ内に格納された1つまたは複数のTSプロファイルにアクセスし、

第1の格納されたTSプロファイルは、少なくとも1つの満足されないチェック(USC)ノードの格納された情報および少なくとも1つの誤って満足されたチェック(MSC)ノードの格納された情報を含み、MSCノードは、(i)誤りのあるビットノード(EBN)に関連し、(ii)前記パリティ検査を満足するチェックノードであり、

第2の格納されたTSプロファイルは、1つまたは複数のUSCノードおよび1つまたは複数のMSCノードを有するトラッピング・セットに関連し、前記第2の格納されたTSプロファイルは、前記1つまたは複数のUSCノードの格納された情報を含むが、前

10

20

記 1 つまたは複数の M S C ノードに関する情報を含まない  
実行することと  
を含む方法。

【請求項 2】

前記第 1 の格納された T S プロファイルは、前記第 2 の格納された T S プロファイルの前記トラッピング・セットより支配的であるトラッピング・セットに関連する、請求項 1 に記載の発明。

【請求項 3】

前記 T S - R O M リスト復号方法は、

( b 1 ) 前記候補の復号された符号語内の前記少なくとも 1 つの U S C ノードに関連する 1 つまたは複数の E B N を識別することと。 10

( b 2 ) ( i ) 前記候補の復号された符号語が 1 つまたは複数の M S C ノードを有し、  
( i i ) 前記 R O M メモリが前記 1 つまたは複数の M S C ノードに関する情報を含む場合に、前記 1 つまたは複数の M S C に関連する 1 つまたは複数の E B N を識別することと、

( b 3 ) 前記識別された E B N を変更することと、

( b 4 ) さらなる処理を実行することであって、

ステップ ( b 2 ) が 1 つまたは複数の E B N を識別する場合に、前記さらなる処理は、前記変更された候補の復号された符号語に対して症候群チェックを実行することを含み、

ステップ ( b 2 ) が 1 つまたは複数の E B N を識別しない場合に、前記さらなる処理は、変更された符号化されたデータを復号することを含む 20

実行することと

を含む、請求項 1 に記載の発明。

【請求項 4】

T S プロファイル内の満足されないチェックノードごとに、前記 T S プロファイルは、

前記満足されないチェックノードが配置される復号レイヤ ( たとえば、 L A Y E R ) と、

前記復号レイヤ内の前記満足されないチェックノード ( U S C ) のインデックス ( たとえば、 U S C \_ I N D E X ) と、

前記満足されないチェックノードに関連する 1 つまたは複数の誤りのあるビットノード ( E B N ) の 1 つまたは複数のインデックス ( たとえば、 E B N \_ I N D E X ) と 30

を含み、 T S プロファイル内の誤って満足されたチェックノードごとに、前記 T S プロファイルは、

前記誤って満足されたチェックノードに関連する 1 つまたは複数の誤りのあるビットノードの位置情報 ( たとえば、 B L O C K \_ C O L U M N 、 B \_ I N D E X )

を含む、請求項 1 に記載の発明。

【請求項 5】

前記 R O M メモリは、複数の格納された T S プロファイルを含み、

前記複数の格納された T S プロファイルの前記復号レイヤ、 U S C インデックス、および E B N インデックスは、第 1 テーブル ( たとえば、 P テーブル 5 1 4 ) に格納され、 40

誤って満足されたチェックノードに関連する前記誤りのあるビットノードの前記位置情報は、第 2 テーブル ( たとえば、 E I テーブル 5 1 8 ) に格納される

請求項 4 に記載の発明。

【請求項 6】

前記格納された T S プロファイルは、前記第 1 テーブル内で、前記格納された T S プロファイル内の満足されないチェックノードの個数に基づいてグループ化され、

同一の個数の満足されないチェックノードを有する格納された T S プロファイルの各グループ内の前記格納された T S プロファイルは、 T S プロファイル優位性によって順序付けられ、 T S プロファイルの優位性は、前記関連するトラッピング・セットがステップ ( a ) の前記復号のエラーフロア特性に対して有する影響に依存する 50

請求項 5 に記載の発明。

【請求項 7】

複数の満足されないチェックノードを有する TS プロファイルごとに、

前記満足されないチェックノードは、前記 TS プロファイル内で復号レイヤによってグループ化され、

同一の復号レイヤを有する満足されないチェックノードのグループ内の前記満足されないチェックノードは、 USC インデックスによって配置される

請求項 6 に記載の発明。

【請求項 8】

前記 ROM メモリは、第 3 テーブル（たとえば、 E テーブル 516 ）および第 4 テーブル（たとえば、 B テーブル 512 ）をさらに含み、 10

前記第 3 テーブルは、一致する TS プロファイルの誤って満足されたチェックノードに関連する前記誤りのあるビットノードの前記位置情報の前記第 2 テーブル内のアドレス（たとえば、 EITABLE\_START\_ADDRESS 、 EITABLE\_END\_ADDRESS ）を識別し、

前記第 1 テーブル内と同一の個数の満足されないチェックノードを有する格納された TS プロファイルのグループごとに、前記第 4 テーブルは、

（ i ）格納された TS プロファイルの前記グループの前記第 1 テーブル内の開始アドレス（たとえば、 PTABLE\_START\_OFFSET ）と、 20

（ ii ）誤って満足されたチェックノードに関連する位置情報を有する前記グループ内の TS プロファイルの個数（たとえば、 NUM\_ETABLE\_ENTRIES ）と、

（ iii ）格納された TS プロファイルの前記グループの前記第 3 テーブル内の開始アドレス（たとえば、 ETABLE\_START\_OFFSET ）と

を識別する、請求項 5 に記載の発明。

【請求項 9】

前記グラフベースの符号は、 Low-Density Parity Check ( LDPC ) 符号である、請求項 1 に記載の発明。

【請求項 10】

グラフベースの符号を使用して符号化された符号化されたデータを復号する装置であって、 30

（ a ）候補の復号された符号語を生成するために、前記符号化されたデータを復号する手段と、

（ b ）前記候補の復号された符号語が復号された正しい符号語ではない場合に、前記復号された正しい符号語を生成することを試みるためにトラッピング・セット（ TS ） - ROM リスト復号方法を実行する手段であって、

前記候補の復号された符号語は、少なくとも 1 つの満足されないチェックノードを有し、満足されないチェックノードは、パリティ検査に合格しないチェックノードであり、

前記 TS - ROM リスト復号方法は、 ROM メモリ内に格納された 1 つまたは複数の TS プロファイルにアクセスし、

第 1 の格納された TS プロファイルは、少なくとも 1 つの満足されないチェック（ USC ）ノードの格納された情報および少なくとも 1 つの誤って満足されたチェック（ MSC ）ノードの格納された情報を含み、 MSC ノードは、（ i ）誤りのあるビットノード（ EBN ）に関連し、（ ii ）前記パリティ検査を満足するチェックノードであり、 40

第 2 の格納された TS プロファイルは、 1 つまたは複数の USC ノードおよび 1 つまたは複数の MSC ノードを有するトラッピング・セットに関連し、前記第 2 の格納された TS プロファイルは、前記 1 つまたは複数の USC ノードの格納された情報を含むが、前記 1 つまたは複数の MSC ノードに関する情報を含まないことを含む装置。

【発明の詳細な説明】

【技術分野】

【 0001 】

本発明は、デジタル信号処理に関し、具体的には、low-density parity check (LDPC) コーディングと称するデータ符号化方法に関する。

【背景技術】

【0002】

通信は、送信器による、通信チャネルを介する受信器への情報の伝送である。実世界では、通信チャネルは、送信器から受信される情報のひずんだ版を受信器に出力する、雑音のある通信路である。ハード・ディスク (HD) ドライブは、送信器から情報を受け入れ、その情報を格納し、その後、おそらくは、その情報の多少ひずんだコピーを受信器に送る、1つのそのような雑音のある通信路である。

【先行技術文献】

10

【特許文献】

【0003】

【特許文献1】米国仮出願第61/089,297号

【特許文献2】米国特許出願第XX/xxx,xxx号 (整理番号08-0241)

【発明の概要】

【発明が解決しようとする課題】

【0004】

HD ドライブなどの通信チャネルによって導入されるひずみは、チャネル・エラーすなわち、チャネル入力信号が 0 であった時に受信器がチャネル出力信号を 1 と解釈し、およびその逆を行う状況を引き起こすのに十分に大きい場合がある。チャネル・エラーは、スループットを下げ、したがって望ましくない。したがって、チャネル・エラーを検出し、かつ / または訂正するツールの継続する必要がある。low-density parity check (LDPC) コーディングは、チャネル・エラーを検出し、訂正する 1 つの方法である。LDPC 符号は、低信号対雑音比 (SNR) 応用のための非常に低いビット誤り率 (BER) を達成できる、既知のシャノン限界に近い符号の 1 つである。LDPC 復号は、その並列性の可能性、低い実施複雑さ、短い復号待ち時間、ならびに高 SNR でのより深刻でないエラーフロアによって区別される。LDPC 符号は、事実上すべての次世代通信標準規格について検討されている。

20

【課題を解決するための手段】

【0005】

30

ある種の実施形態で、本発明は、グラフベースの符号を使用して符号化された符号化されたデータを復号する方法を含む。この方法は、(a) 候補の復号された符号語を生成するために、符号化されたデータを復号することと、(b) 候補の復号された符号語が復号された正しい符号語ではない場合に、復号された正しい符号語を生成することを試みるためにトランピング・セット (TS) - ROM リスト復号方法を実行することとを含む。候補の復号された符号語は、少なくとも 1 つの満足されないチェックノードを有し、満足されないチェックノードは、パリティ検査に合格しないチェックノードである。TS - ROM リスト復号方法は、ROM メモリ内に格納された 1 つまたは複数の TS プロファイルにアクセスする。第 1 の格納された TS プロファイルは、少なくとも 1 つの満足されないチェック (USC) ノードの格納された情報および少なくとも 1 つの誤って満足されたチェック (MSC) ノードの格納された情報を含む。MSC ノードは、(1) 誤りのあるビットノード (EBN) に関連し、(2) パリティ検査を満足するチェックノードである。第 2 の格納された TS プロファイルは、1 つまたは複数の USC ノードおよび 1 つまたは複数の MSC ノードを有するトランピング・セットに関連し、第 2 の格納された TS プロファイルは、1 つまたは複数の USC ノードの格納された情報を含むが、1 つまたは複数の MSC ノードに関する情報を含まない。

40

【0006】

他の実施形態で、本発明は、グラフベースの符号を使用して符号化された符号化されたデータを復号する装置である。この装置は、(a) 候補の復号された符号語を生成するために、符号化されたデータを復号するように適合された復号器と、(b) 候補の復号され

50

た符号語が復号された正しい符号語ではない場合に、復号された正しい符号語を生成することを試みるためにトラッピング・セット (T S) - R O Mリスト復号方法を実行するよう 10 に適合されたポストプロセッサとを含む。候補の復号された符号語は、少なくとも 1 つの満足されないチェックノードを有し、満足されないチェックノードは、パリティ検査に合格しないチェックノードである。T S - R O Mリスト復号方法は、R O Mメモリ内に格納された 1 つまたは複数の T S プロファイルにアクセスする。第 1 の格納された T S プロファイルは、少なくとも 1 つの満足されないチェック (U S C) ノードの格納された情報を含む。M S C ノードは、(1) 誤りのあるビットノード (E B N) に関連し、(2) パリティ検査を満足するチェックノードである。第 2 の格納された T S プロファイルは、1 つまたは複数の U S C ノードおよび 1 つまたは複数の M S C ノードを有するトラッピング・セットに関連し、第 2 の格納された T S プロファイルは、1 つまたは複数の U S C ノードの格納された情報を含むが、1 つまたは複数の M S C ノードに関する情報を含まない。

#### 【0 0 0 7】

本発明の他の態様、特徴、および利益は、次の詳細な説明、添付の特許請求の範囲、および添付図面からより十分に明らかになり、添付図面では、類似する符号が、類似するまたは同一の要素を識別する。

#### 【図面の簡単な説明】

#### 【0 0 0 8】

【図 1】L D P C コーディングを利用する通常のハード・ディスク (H D) ドライブ 10 0 の一部を示すブロック図である。

【図 2】図 2 (A) は、L D P C H 行列 2 0 0 を示す図であり、図 2 (B) は、H 行列 2 0 0 のタナー・グラフを示す図である。

【図 3】復号器 1 1 2 によって使用される通常の L D P C 復号方法 3 0 0 を示す流れ図である。

【図 4】トラッピング・セットを識別し、これらのトラッピング・セットに関するさまざまな情報を記録するオフライン・トラッピング・セット (T S) シミュレーション・ツール 4 0 0 を示すブロック図である。

【図 5】本発明の一実施形態による L D P C 復号システム 5 0 0 を示すブロック図である。

【図 6】図 5 の R O M P テーブル 5 1 4 の例示的レイアウトを示す図である。

【図 7】図 5 の B テーブル 5 1 2 の例示的レイアウトを示す図である。

【図 8】図 5 の E テーブル 5 1 6 の例示的レイアウトを示す図である。

【図 9】図 5 の E I テーブル 5 1 8 の例示的レイアウトを示す図である。

【図 10】図 5 の R A M P テーブル 5 2 2 の例示的レイアウトを示す図である。

【図 11】図 5 の R A M インデックス・テーブル 5 2 4 の例示的レイアウトを示す図である。

【図 12】図 5 の L D P C 復号システム 5 0 0 によって使用される例示的プロセス 1 2 0 0 を示す流れ図である。

【図 13】図 5 のポストプロセッサ 5 0 4 によって実施される図 1 2 の例示的な T S - R O Mリスト復号プロセス 1 2 0 6 を示す流れ図である。

【図 14】図 1 3 の例示的な T S - R O M検索プロセス 1 3 1 4 を示す流れ図である。

【図 15】図 1 2 の例示的な T S - R A Mリスト復号プロセス 1 2 0 8 を示す流れ図である。

【図 16】図 1 2 の例示的な T S - R A M更新プロセス 1 2 1 6 を示す流れ図である。

#### 【発明を実施するための形態】

#### 【0 0 0 9】

図 1 は、L D P C コーディングを利用する通常のハード・ディスク (H D) ドライブ 10 0 の一部のブロック図である。H D ドライブ 1 0 0 は、プラッタ 1 0 2 および読み取りチャネル 1 0 4 を含む。読み取りチャネル 1 0 4 は、L D P C 符号器 1 0 6 、書き込みプロセッ 50

サ 1 0 8 、 読取りプロセッサ 1 1 0 、 および L D P C 復号器 1 1 2 を含む。 経路 1 1 4 は 、 L D P C 符号器 1 0 6 と L D P C 復号器 1 1 2 との間の雑音のある通信路である。

#### 【 0 0 1 0 】

プラッタ 1 0 2 に書き込まれる情報語は、 L D P C 符号語を作るために L D P C 符号器 1 0 6 によって処理される。 L D P C 符号語は、 書込みプロセッサ 1 0 8 に送られ、 書込みプロセッサ 1 0 8 は、 複数のモジュール、 たとえば、 B P S K ( 二進移相変調 ) 符号器 、 ディジタル - アナログ変換器などを含む。 書込みプロセッサ 1 0 8 の出力 1 1 6 が、 プラッタ 1 0 2 に書き込まれる。

#### 【 0 0 1 1 】

プラッタ 1 0 2 から読み取られた信号 1 1 8 は、 読取りプロセッサ 1 1 0 に送られ、 読取りプロセッサ 1 1 0 は、 複数のモジュール、 たとえば、 前置増幅器、 連続時間フィルタ 、 固定インパルス応答フィルタ、 検出器、 アナログ - ディジタル変換器などを含む。 読取りプロセッサ 1 1 0 は、 対数尤度比 ( L L R ) 値  $L_{c,h}$  を L D P C 復号器 1 1 2 に出力し 、 L D P C 復号器 1 1 2 は、 復号された情報語を出力する。 さらに、 L D P C 復号器 1 1 2 は、  $E_{L D P C}$  値を読み取るプロセッサ 1 1 0 に送り返す。  $E_{L D P C}$  は、 下の式 6 によって定義され、 中間の計算された L L R 値を表す。 読取りプロセッサ 1 1 0 は、  $E_{L D P C}$  値を使用して、 その性能を調整し、 これは、 ターボ復号として知られるプロセスである。

#### 【 0 0 1 2 】

##### L D P C 符号化

L D P C 符号器 1 0 6 は、 情報語のビットに、 L D P C 符号によって指定される複数のパリティ・ビットを付加して、 符号語を作る。 情報語のビットは、 可変ビットとして知られ、 これらの可変ビットの個数は、 K と表される。 ある L D P C 符号語内のビットの総数は、 N と表される。 したがって、 パリティ・ビットの個数は、 N - K によって与えられる。 特定の L D P C 符号のレートは、 K / N すなわち、 符号語長さに対する情報語長さの比である。 したがって、 9 ビット符号語を作るために 3 ビット情報語のそれぞれに 6 つのパリティ・ビットを付加する L D P C 符号は、 1 / 3 のレートを有する。 通常の H D ドライブの場合に、 情報語長さ K は、 4 0 9 6 ビット ( 通常の H D ドライブ・セクタの長さ ) であり、 パリティ・ビットの個数は、 4 5 0 6 ビットの符号語長さおよび 0 . 9 のレートについて、 約 4 1 0 ビットである。

#### 【 0 0 1 3 】

L D P C 符号語内の各パリティ・ビットは、 その特定の L D P C 符号によって指定される特定の形でその符号語内の 1 つまたは複数の他の ( 可変またはパリティ ) ビットに関連し、 パリティ・ビットに割り当てる値は、 L D P C 符号を満足するようにセットされる。 通常の L D P C 符号は、 関連するビットがパリティ検査制約を満足する、 たとえば、 関連するビットの合計が偶数である、 すなわち、 合計  $m o d u l o \ 2 = 0$  であることを指定する。

#### 【 0 0 1 4 】

##### L D P C 符号

特定の L D P C 符号は、 パリティ検査行列または H 行列あるいは単純に H として知られる 1 および 0 の 2 次元行列によって定義される。 H は、 L D P C 符号器と L D P C 復号器との両方によって、 先驗的に既知である。 H は、 N 列 N - K 行を含む、 すなわち、 符号語のすべてのビットの列およびすべてのパリティ・ビットの行を含む。 H の各 1 は、 列の符号語ビットと行のパリティ・ビットとの間の関連を表す。 たとえば、 H の第 3 行第 7 列の 1 は、 第 3 パリティ検査ビットが符号語の第 7 ビットに関連することを意味する。 検査ビットおよびその検査ビットに関連するすべての可変ビットの  $m o d u l o \ 2$  合計は、 0 でなければならない。

#### 【 0 0 1 5 】

H のある列内の 1 の個数は、 その列の重み  $w_c$  として知られる。 同様に、 H のある行内の 1 の個数は、 その行の重み  $w_r$  として知られる。 すべての列が同一の  $w_c$  を有し、 すべ

10

20

30

40

50

ての行が同一の  $w_r$  を有する  $H$  によって定義される LDPC 符号は、正則 LDPC 符号 (regular LDPC code) として知られる。 $w_c$  および / または  $w_r$  がそれすべての列および / または行にまたがって同一ではない  $H$  によって定義される LDPC 符号は、非正則 LDPC 符号として知られる。

【0016】

通常の LDPC を定義する特性は、 $H$  が「疎」であるすなわち、 $H$  の要素が、少数の 1 を伴ってほとんどが 0 であることである。研究は、 $H$  行列が、通常は良好に動作するため  $w_c = 3$  を必要とし、非正則 LDPC 符号が正則 LDPC 符号より性能が優れていることを示した。

【0017】

図 2 (A) に、LDPC  $H$  行列 200 を示す。 $H$  行列 200 は、 $N = 9$  列および  $N - K = 6$  行を含む。したがって、 $H$  行列 200 は、3 ビットの情報語を受け入れ、6 つのパリティ・ビットを付加し、9 ビットの符号語を出力する LDPC 符号を定義する。したがって、この特定の LDPC 符号のレートは、3 / 9 すなわち 1 / 3 である。 $H$  行列 200 によって定義される LDPC 符号は、正則であり、2 の  $w_c$  および 3 の  $w_r$  を伴う。

【0018】

チャネル出力：対数尤度比

図 1 に戻って、LDPC 符号器 106 と LDPC 復号器 112 との間の経路 114 は、雑音のある通信路であり、したがって、復号器 112 は、LDPC 符号器 106 によって出力された符号語の完全なコピーは受け取らない。その代わりに、読み取りプロセッサ 110 は、1 つまたは複数の  $L_{ch}$  値を出力し、ここで、各  $L_{ch}$  値は、チャネル入力符号語内の 1 ビットに対応する。

【0019】

各  $L_{ch}$  値は、対数尤度比 (LLR) である。LLR は、複数のビットを含むデータ構造であり、ここで、単一の符号ビットは、硬判定 (すなわち、オリジナル・ビットが 1 または 0 のどちらであったかに関する読み取りプロセッサ 110 の最良の推測) を示し、残りの大きさビットは、その硬判定での読み取りプロセッサ 110 の信頼度の度合を示す。より正確には、LLR は、

【数 1】

$$\log \frac{p_0}{p_1}$$

を表し、ここで、 $p_0$  は、サンプルが 0 を表す確率であり、 $p_1$  は、サンプルが 1 を表す確率である。

【0020】

たとえば、読み取りプロセッサ 110 は、各  $L_{ch}$  値を 5 ビット・データ構造として出力することができ、ここで、最上位ビットは、硬判定値を示す符号ビットであり、4 つの大きさビットの 16 個の値は、硬判定の信頼度を示す。したがって、たとえば、1 つの通常の方式で、2 進数 00000 の LLR 値は、最小信頼度を有する 0 の硬判定値を示し、2 進数 01111 の値は、最大信頼度を有する 0 の硬判定値を示し、2 進数 10000 は、未使用であり、2 進数 10001 は、最小信頼度を有する 1 の硬判定値を示し、2 進数 11111 の値は、最大信頼度を有する 1 の硬判定値を示すはずである。

【0021】

LDPC 復号：確率伝搬

図 3 は、復号器 112 によって使用される通常の LDPC 復号方法 300 の流れ図である。LDPC 復号器 112 は、 $N$  個の  $L_{ch}$  値を受け取り、復号された情報語を出力する。復号方法 300 の心臓部は、確率伝搬と呼ばれる反復的 2 相メッセージパッシング・アルゴリズムである。確率伝搬は、タナー・グラフと呼ばれる視覚化の使用を用いて最もよ

10

20

30

40

50

く説明される。

【0022】

図2(B)は、H行列200のタナー・グラフである。一般に、タナー・グラフは、1) H内の列の数と等しい(したがって、可変ビットの個数と等しい)ビットノードの個数n、2) H内の行の個数と等しい(したがって、パリティ・ビットの個数と等しい)チェックノードの個数m、3) それぞれが单一のビットノードを单一のチェックノードに接続する、辺としても知られる線、4) ビットノードnのそれぞれについて、受信器から受け取られたオリジナルの $L_{c,h}$ 値、および5) ビットノードnのそれぞれについて、計算された硬判定出力値

【数2】

$\hat{x}_n$

10

を含む。タナー・グラフ2(B)は、9つのビットノード $n_0 \sim n_8$ 、6つのチェックノード $m_0 \sim m_5$ 、ビットノードをチェックノードに接続する18個の辺202、9つの $L_{c,h}$ 値、および9つの

【数3】

$\hat{x}_n$

20

値を含む。

【0023】

タナー・グラフの辺は、(すなわち、可変)ビットノードnとチェックノードmの間の関係を表す、すなわち、辺は、H内の1を表す。たとえば、図2(B)では、辺202は、第1のビットノード $n_0$ を第4のチェックノード $m_3$ に接続し、これは、図2(A)のH行列200の第1列第4行に1があることを意味する。

【0024】

タナー・グラフは、二部グラフである、すなわち、辺は、ビットノードをチェックノードに接続することだけができ、ビットノードを別のビットノードにまたはチェックノードを別のチェックノードに接続することはできない。辺によって特定のチェックノードmに接続されたすべてのビットノードnの集合を、N(m)と表す。辺によって特定のビットノードnに接続されたすべてのチェックノードmの集合を、M(n)と表す。

30

【0025】

特定の(ビットまたはチェック)ノードのインデックスは、グラフ内のそのノードのオリジナル・シーケンスである。(ビットまたはチェック)ノードの次数は、そのノードに接続された辺の個数である。したがって、タナー・グラフ内のビットノードnの次数は、対応するH行列内の列nの重み $w_c$ と等しく、タナー・グラフ内のチェックノードmの次数は、対応するH行列内の行mの重み $w_r$ と等しい。

【0026】

図3に戻って、処理は、ステップ302で開始され、ステップ304すなわち復号器初期化に進む。復号器初期化304は、ビットノードnに接続されたすべての辺(たとえば、図2(B)の202)にビットノードnに関連する対応する $L_{c,h}$ 値をセットし、ビットノードnの

40

【数4】

$\hat{x}_n$

値にビットノードnの $L_{c,h}$ の硬判定値をセットすることを含む。したがって、たとえば、図2(B)では、ビットノード $n_0$ に関連する $L_{c,h}$ 値が+5である場合に、ステップ304で、ビットノード $n_0$ をチェックノード $m_0$ および $m_3$ に接続する2つの辺202

50

に + 5 がセットされ、ビットノード n の

【数 5】

$\hat{x}$

値に 1 がセットされる。このステップの第 1 部分を表す代替の形は、ビットノード  $n_0$  が、集合  $M(n_0)$  内の各チェックノード  $m$  に + 5 のメッセージを送ることである。ビットノード  $n$  からチェックノード  $m$  に送られるメッセージは、 $Q_{n,m}$  と表され、ここで、 $Q_{n,m}$  は、LLR の形である。初期化されたばかりの復号器の状態を、状態 0 と称する。

【0027】

次に、ステップ 304 は、症候群チェック・ステップ 306 に、N 個の

10

【数 6】

$\hat{x}$

値を含むベクトル

【数 7】

$\hat{x}_n$

を送る。ベクトル

【数 8】

20

$\hat{x}$

は、符号語候補である。症候群チェック・ステップ 306 は、次の式 1 を使用して症候群ベクトル  $z$  を計算する。

【数 9】

$$z = \hat{x}H^T \quad (1)$$

ここで、 $H^T$  は、 $H$  行列の転置行列である。 $z$  が 0 ベクトルである場合には、ベクトル

【数 10】

$\hat{x}$

は、 $H$  によって定義されるすべてのパリティ検査制約を満足している、すなわち、

【数 11】

$\hat{x}$

は有効な符号語である。その場合には、処理は、巡回冗長検査 (CRC) チェック 318 に進む。

【0028】

その代わりに、 $z$  が 0 ベクトルではない場合には、ベクトル

【数 12】

40

$\hat{x}$

は、パリティ検査制約のうちの 1 つまたは複数に不合格であり、これを、通常は、満足されないチェックノードまたは USC と称する。0 スカラ値ではない症候群ベクトル  $z$  内の要素の個数は、ベクトル

【数 13】

$\hat{x}$

内の USC の個数  $b$  である。さらに、症候群ベクトル  $z$  の非ゼロ・スカラ要素のインデックスは、ベクトル

50

【数14】

 $\hat{x}$ 

内のU.S.Cのインデックスである。

【0029】

ベクトル

【数15】

 $\hat{x}$ 

が症候群チェック306に不合格である場合には、処理は、1つまたは複数の復号反復308の第1の反復に継続する。復号反復308は、3つのステップすなわち、1)確率伝搬チェックノード更新ステップ310、2)確率伝搬ビットノード更新ステップ312、および3)症候群チェック・ステップ314(ステップ306と同一である)を含む。

【0030】

確率伝搬チェックノード更新ステップ310では、各チェックノードmは、集合N(m)内のすべてのビットノードnから受け取った $Q_{n,m}$ 個のメッセージを使用して、次の式2、3、および4に従って、 $R_{m,n}$ と表されるメッセージを計算する。

【数16】

$$R_{mn}^{(i)} = \delta_{mn}^{(i)} \max(\kappa_{mn}^{(i)} - \beta, 0) \quad (2)$$

$$\kappa_{mn}^{(i)} = |R_{mn}^{(i)}| = \min_{n' \in N(m) \setminus n} |Q_{n',m}^{(i-1)}| \quad (3)$$

$$\delta_{mn}^{(i)} = \left( \prod_{n' \in N(m) \setminus n} \text{sgn}(Q_{n',m}^{(i-1)}) \right) \quad (4)$$

ここで、iは、復号反復であり、 $N(m) \setminus n$ は、ビットノードnを除外した集合N(m)であり、 $\beta$ は、正の定数であり、その値は、符号パラメータに依存する。計算された $R_{m,n}$ 個のメッセージは、その後、同一の辺に沿って、集合N(m)内のすべてのビットノードnに送り返される。 $Q_{n,m}$ 個のメッセージに似て、 $R_{m,n}$ 個のメッセージは、LLRである。

【0031】

次に、確率伝搬ビットノード更新ステップ312では、各ビットノードnは、次の式5に従って $Q_{n,m}$ 個のメッセージを計算する。

【数17】

$$Q_{nm}^{(i)} = L_n^{(0)} + \sum_{m' \in M(n) \setminus m} R_{m',n}^{(i)} \quad (5)$$

ここで、

【数18】

 $L_n^{(0)}$ 

は、ビットノードnの $L_{c,h}$ 値であり、 $M(n) \setminus m$ は、チェックノードmを除外した集合 $M(n)$ である。次に、ビットノードnは、計算された $Q_{n,m}$ 個のメッセージを集合 $M(n)$ 内のすべてのチェックノードmに送る。

【0032】

また、ビットノード更新ステップ312中に、各ビットノードnは、次の式6および7に従って、その

【数19】

 $\hat{x}_n$ 

値を更新する。

【数20】

$$E_n^{(i)} = \sum_{m \in M(n)} R_{mn}^{(i)} \quad (6)$$

$$P_n = L_n^{(0)} + E_n^{(i)} \quad (7)$$

10

 $P_n = 0$  の場合には、

【数21】

 $\hat{x}_n = 0$ であり、 $P_n < 0$  の場合には、

【数22】

 $\hat{x}_n = 1$ 

20

である。式6によって生成される値を、E値またはE<sub>LDP<sub>C</sub></sub>値とも称する。通常、E<sub>LDP<sub>C</sub></sub>値は、ターボ復号として知られるチューニング・プロセスの一部として読み取りプロセッサ（たとえば、図1の読み取りプロセッサ110）に送り返される。式7によって生成される値を、P値と称する。式2～7によって表される特定の確率伝搬アルゴリズムは、min-sumアルゴリズムとして知られる。

【0033】

【数23】

 $\hat{x}_n$ 

30

が、各復号反復308中に更新され、最終的に復号プロセス300によって出力されることに留意されたい。オリジナルのLLR値L<sub>ch</sub>は、復号プロセス300中に変更されないままになる。言い換えると、各復号反復308中に、各ビットノードnは、その投票を、チェックノードmを介して関連するすべての他のビットノードnの正しい値について投じる。たとえば、図2（B）では、ビットノードn<sub>0</sub>は、チェックノードm<sub>0</sub>およびm<sub>3</sub>に関連する。したがって、n<sub>0</sub>は、その投票を、チェックノードm<sub>0</sub>およびm<sub>3</sub>に関連するビットノードすなわちn<sub>3</sub>、n<sub>5</sub>、n<sub>6</sub>、およびn<sub>7</sub>の正しい値について投じる。ビットノードnのL<sub>ch</sub>値の大きさ値が大きいほど（すなわち、信頼度が高いほど）、ビットノードnの投票はより多くカウントされる。この投票の正味の効果は、低いL<sub>ch</sub>大きさ値（すなわち、信頼度）を有するビットノードの

【数24】

 $\hat{x}_n$ 

40

値が、そのビットノードが関連する高信頼度ビットノードの確率を変更し、これに従うことである。言い換えると、あるビットノードのL<sub>ch</sub>値が、誤りのある硬判定値および小さい大きさを含む場合に、他のビットノードの組み合わされた投票は、1回または複数回の反復の後に、その誤りのある硬判定値を訂正する傾向がある。

【0034】

50

ビットノード更新ステップ 312 は、復号器の現在の  
【数 25】

$\hat{x}$

値から構成されたベクトル

【数 26】

$\hat{x}_n$

を症候群チェック・ステップ 314 に送る。ステップ 314 の症候群チェックは、上で述べたステップ 306 の症候群チェックと同一である。ベクトル 10

【数 27】

$\hat{x}$

が症候群チェック 314 に合格する場合には、ベクトル

【数 28】

$\hat{x}$

が CRC ステップ 318 に送られる。

【0035】

L D P C 復号：巡回冗長検査および誤って満足されたチェックノード  
症候群チェック 306 または 314 に合格することは、ベクトル 20

【数 29】

$\hat{x}$

が有効な符号語であることだけを意味し、必ずしも復号された正しい符号語 ( DCCW ) であることは意味しない。L D P C 復号器が DCCW ではない有効な符号語を生成することが有り得る。その場合に、ベクトル

【数 30】

$\hat{x}$

内には、U S C ではなく、誤って満足されたチェックノード ( MSC ) がある。したがって、有効なベクトル

【数 31】

$\hat{x}$

が DCCW であることを保証するために、プロセス 300 は、ベクトル

【数 32】

$\hat{x}$

を巡回冗長検査 ( CRC ) 318 に渡す。CRC チェックは、伝送または格納中のデータの変更を検出できるチェックサム演算である。 40

【0036】

ベクトル

【数 33】

$\hat{x}$

が CRC チェックに合格する場合には、ベクトル

【数34】

は、DCCWであり、プロセス300は、グローバル変数DCCWに真をセットし、ベクトル

【数35】

を出力し、ステップ320で終了する。そうでない場合には、ベクトル

【数36】

10

はDCCWではなく、プロセス300は、グローバル変数DCCWに偽をセットし、ベクトル

【数37】

を出力し、ステップ320で終了する。グローバル変数DCCWは、DCCWが生成されたか否かを他の復号プロセス（たとえば、下で述べる図12のTS-ROMリスト復号プロセス1206）に知らせる。

20

【0037】

ステップ314に戻って、ベクトル

【数38】

が症候群チェックに不合格である場合には、ベクトル

【数39】

は、それでも1つまたは複数のUSCを含む。USCを解決する通常の方法は、もう1つの復号反復308を実行することである。しかし、特定のベクトル

30

【数40】

内に、穩当な長さの時間では絶対に満足できない1つまたは複数のUSCが存在する場合がある。したがって、LDPD復号器は、通常、特定のベクトル

【数41】

に対して実行できる復号反復の回数において制限される。反復の最大回数の通常の値は、50から200までの範囲にわたる。

40

【0038】

図3では、ステップ316が、反復の最大回数に達したかどうかを判定する。そうでない場合には、もう1つの復号反復308が実行される。その代わりに、反復の最大回数に達した場合には、復号器プロセス300が失敗した、すなわち、復号器は、「失敗した復号器」である。その場合には、プロセス300は、グローバル変数DCCWに偽をセットし、ベクトル

【数42】

50

を出力し、ステップ 320 で終了する。

【0039】

失敗した復号器のベクトル

【数43】

$\hat{x}$

が、少数（たとえば、16個未満）のUSCを含む場合には、ベクトル

【数44】

$\hat{x}$

10

は、ニア・コードワード (near codeword) (NCW) と呼ばれる。失敗した復号器のベクトル

【数45】

$\hat{x}$

が、多数の（たとえば、15個を超える）USCを含む場合には、ベクトル

【数46】

$\hat{x}$

20

は、無効符号語 (invalid codeword) (ICW) と呼ばれる。

【0040】

失敗した復号プロセスを処理する2つの通常の方法は、1) 対応するデータの再送を要求することまたは2) 1つもしくは複数の後処理 (PP) 方法にベクトル

【数47】

$\hat{x}$

を渡すことである。通常、ベクトル

【数48】

$\hat{x}$

30

内のUSCの個数bは、この2つの方法のどちらが使用されるかを規定する。大きいb（たとえば、16を超える）は、通常、再送または他の後処理方法によって処理され、小さいb値は、エラーフロア軽減後処理方法によって処理される。

【0041】

BER、SNR、およびエラーフロア

L D P C 復号器のビット誤り率 (BER) は、処理されるx個のビットについて何個の誤って復号されたビットが生成されるかを表す比率である。したがって、たとえば、10<sup>-9</sup>のBERを有する復号器は、平均して、処理される10億ビットごとに1つの誤ったビットを生成する。BERが小さいほど、復号器はよりよい。L D P C 復号器のBERは、復号器が失敗するすなわち、復号された正しい符号語DCCWに収束せずに終了する時に、高まる（悪化する）。

40

【0042】

L D P C 復号器のBERは、復号器の入力信号の信号対雑音比 (SNR) によって強く影響される。SNRの関数としてのBERのグラフは、通常、2つの別個の領域すなわち、BERがSNRの単位上昇を前提としてすばやく改善される（減少する）最初の「滝」領域と、SNRの単位上昇がBERのささやかな改善だけをもたらす後続の「エラーフロア」領域とを含む。したがって、エラーフロア領域での大きいBER改善を達成するためには、SNR向上以外の方法が必要である。

【0043】

L D P C 復号のエラーフロア特性を改善する1つの方法は、符号語長さを増やすこと

50

ある。しかし、符号語長さの増加は、L D P C 復号に必要なメモリおよび他のコンピューティング・リソースをも増やす。したがって、H D ドライブ上の読み取りチャネル・デバイスについて通常そうであるように、そのようなリソースが厳密に制限される場合には、必要なエラーフロア改善をもたらすために、他の方法を見つけなければならない。

#### 【0044】

もう1つの不足するリソースが、処理サイクルである。通常、指定されたスループットを達成するために、H D ドライブは、符号語を復号するために固定された個数の読み取りチャネル処理サイクルを計上する。その予算を超える方法（すなわち、オフザフライ（o f f - t h e - f l y）法）は、スループットを下げる。より望ましいのは、クロックサイクル割当内でD C C Wを回復し、したがってスループットを下げるオンザフライ法である。

10

#### 【0045】

##### トラッピング・セットおよび支配的トラッピング・セット

(a, b) トラッピング・セットは、復号器が反復の最大回数以内に満足できないb個のU S CおよびこれらのU S Cに関連するa個の誤りのあるビットノード（E B N）の集合である。トラッピング・セットの大多数は、5つ未満のU S Cおよび10個未満のE B Nを含む。トラッピング・セットは、L D P C 復号器のエラーフロア特性に大きい影響を有する、すなわち、L D P C 復号器がD C C Wへの収束に失敗する時に、これは、しばしば、トラッピング・セットのゆえである。

#### 【0046】

L D P C 復号器のエラーフロア特性を改善する1つの形は、(i) 失敗した復号器の【数49】

20



ベクトル内のU S Cを調べ、トラッピング・セット（ある場合に）を識別し、(i i)これらのU S Cに関連するE B Nを識別し、(i i i)これらのトラッピング・セットに関連する1つまたは複数のE B Nを反転し、(i v)復号器を再始動することである。1つの可能な実施態様では、L D P C 復号器が初期化されたばかりである、すなわち、復号器が状態0である場合に、E B Nの反転は、(i) E B NのL<sub>c h</sub>値の硬判定値を反転することすなわち、1を0にし、逆も同様にすることと、(i i)その同一のL<sub>c h</sub>値の大きさビットすなわち信頼度に最大値、たとえば全ビット1をセットすることとを含む。復号器が、状態0以外のある状態である場合には、E B Nの反転は、(i) E B NのP値（上の式7によって定義される）の硬判定値を判定することと、(i i)そのE B NのL<sub>c h</sub>値、P値、およびすべての関連するQ<sub>n m</sub>個のL L Rの硬判定値に、ステップ(i)の硬判定値の反対をセットすることと、(i i i)そのE B NのL<sub>c h</sub>値、P値、およびすべての関連するQ<sub>n m</sub>個のL L Rの大きさビットに最大値をセットすることとを含む。しばしば、1つまたは2つのE B Nを反転することによって、トラッピング・セットが「破壊され」、再始動された復号器は、D C C Wに収束する。

30

#### 【0047】

異なるトラッピング・セットは、破壊された時に、エラーフロア特性の異なる改善をもたらす。支配的トラッピング・セット（D T S）は、その破壊がB E R / エラーフロア特性の指定された改善をもたらす、トラッピング・セットの最小限のセットを指す。たとえば、D T S - 1は、B E Rの単一桁の、たとえば10<sup>-9</sup>から10<sup>-10</sup>への大きさ改善をもたらすトラッピング・セットの最小限のセットを指し、D T S - 3は、B E Rの3桁の、たとえば10<sup>-10</sup>から10<sup>-13</sup>への大きさ改善をもたらす。

40

#### 【0048】

##### ニア・コードワードのリスト復号

リスト復号は、トラッピング・セットを検出し、破壊する、1つの後処理法である。リスト復号では、ベクトル

## 【数 5 0】

$\lambda$

内の観察されたトラッピング・セットが、既知のトラッピング・セットの1つまたは複数のリストと照合される。トラッピング・セット・リストは、通常、リスト内の各トラッピング・セット内のすべてのU S CのインデックスおよびこれらのU S Cに関連する1つまたは複数のE B Nのインデックスを含む。観察されたトラッピング・セットと一致するトラッピング・セットが、リスト内で見つかる場合には、E B Nインデックス値（1つまたは複数）が、リストから取り出される。次に、これらのビットノードが反転され、図3の復号プロセス300が再始動される。

10

## 【0049】

## トラッピング・セット・シミュレーション

リスト復号に必要なトラッピング・セット・リストは、通常、ソフトウェア・シミュレーション・ツールおよびハードウェア・シミュレーション・ツールを使用してオフラインで生成される。図4は、トラッピング・セットを識別し、これらのトラッピング・セットに関するさまざまな情報を記録するオフライン・トラッピング・セット（T S）シミュレーション・ツール400のブロック図である。ツール400は、たとえば、フィールドプログラマブル・ゲート・アレイ（F P G A）内で実施することができる。L D P Cの正しい符号語（C C W）402が、チャネルおよび信号モデル404に送られ、このチャネルおよび信号モデル404は、図1の雑音のある通信路114の挙動をエミュレートする。チャネルおよび信号モデル404は、 $L_{c,h}$ 値406をL D P C復号器408に出力する。L D P C復号器408が、ニア・コードワード（N C W）410を生成する場合には、N C W 410が、症候群チェック・モジュール412および不一致位置レコーダ414に送られる。症候群チェック412は、N C W 410内のすべてのU S Cのインデックス416を出力する。不一致位置レコーダ414は、N C W 410をC C W 402と比較し、N C W 410内のすべてのE B Nのインデックス418を出力する。U S Cインデックス416およびE B Nインデックス418が、トラッピング・セット（T S）情報420を構成する。

20

## 【0050】

所与のL D P C実施態様について、すべての可能なトラッピング・セットが、数百万個に達する場合がある。しかし、その実施態様のエラーフロアにおける大きい（すなわち、1桁以上の）改善の達成は、通常、すべての可能なトラッピング・セットのサブセットすなわち支配的トラッピング・セット（D T S）のみを必要とする。したがって、オフラインT Sシミュレーション・ツール400は、D T S - Nコンパイラ422を含み、D T S - Nコンパイラ422は、その入力としてT S情報420をとり、D T S - N情報424を生成する。

30

## 【0051】

D T S - Nコンパイラ422は、3ステップ・プロセスすなわち、収集、ランキング、および評価を使用する。トラッピング・セット収集法は、トラッピング・セットを検出するのに、符号グラフの構造に基づく決定論的雑音インパルスを利用する。次に、収集されたトラッピング・セットは、ディスタンス・トゥー・エラー・バウンダリ（d i s t a n c e - t o - e r r o r b o u n d a r y、D E B）値によってランキングされ、ここで、小さいD E B値を有するトラッピング・セットが、エラーフロアにより多く寄与する。次に、重点的サンプリングが、トラッピング・セットを評価し、予測されたランキングを確認するのに使用される。

40

## 【0052】

実際には、そのようなF P G Aベースのオフライン・シミュレーションは、実行に1年を要する可能性がある。たとえば、4 G b / s H D ドライブについて $10^{-15}$ のB E Rをもたらすトラッピング・セットを識別するためには、オフライン・シミュレーション・ツール（たとえば、図4のツール400）を約289日間動作させる必要がある。通常

50

、この時間制約は問題ではない。というのは、しばしば、H D ドライブ読取りチャネルの最終設計とチップの量産との間に1年から2年の遅延があるからである。

【0053】

トラッピング・セット読取り専用メモリ ( T S - R O M )

したがって、図4のツール400のようなオフラインT Sシミュレーション・ツールを使用することによって、特定のL D P C実施態様についてエラーフロア特性の改善をもたらす1つまたは複数のトラッピング・セットまたは支配的トラッピング・セットを先驗的に識別することが可能である。ランタイム環境でリスト復号を実施する1つの形は、オフライン生成された図4のトラッピング・セット情報420をトラッピング・セット読取り専用メモリ ( T S - R O M ) に格納し、そのT S - R O Mをリスト・デコーダ・プログラムと結合することである。T S - R O Mリスト・デコーダ・プログラムは、

【数51】

↑

で観察されるU S CをT S - R O Mに格納されたトラッピング・セットと比較する。一致が見つかる場合には、T S - R O Mリスト・デコーダ・プログラムは、L D P C復号器内の適当なビットノード値を反転し、復号器を再始動する。

【0054】

通常、T S - R O M情報は、單一リンク・リストまたは二重リンク・リストにランダムに格納され、このリストは、ブルート・フォース順次検索を使用して検索される。通常、各 ( a , b ) トラッピング・セットは、T S - R O Mリスト内で ( 2 + a + b ) 個のレコードを占める。したがって、( 4 , 4 ) トラッピング・セット・プロファイル ( すなわち、4つのU S Cおよび4つのE B N ) について、10個のレコードすなわち、4つのU S Cがあることを示す1つのレコード、それに続く4つの個々のU S Cレコード、次に4つのE B Nがあることを示すレコード、それに続く4つの個々のE B Nレコードがある。通常のT S - R O Mリスト実施態様は、約100個のトラッピング・セットを格納し、誤って満足されたチェックノードに関する情報は全く格納しない。

【0055】

そのようなT S - R O M実施態様が経済的に実用的であるためには、單一のT S - R O Mが、多数の実施態様で、要求されるエラーフロア改善を達成できなければならない。しかし、トラッピング・セットは、同一のL D P C符号が実施される時であっても、実施態様によって変化する。たとえば、2つのH D ドライブで使用されるL D P C符号が同一の場合であっても、これらのH D ドライブに関連するトラッピング・セットが異なる場合がある。具体的に言うと、研究から、トラッピング・セットが、H D ドライブのジッタ・プロファイル、記号間干渉特性、およびパルス整形方式によって影響を受けることが示された。これらの要因は、異なる製造業者のH D ドライブの間だけではなく、同一製造業者の異なるH D ドライブ・モデルの間でも変化する可能性があり、同一モデルの異なるプロダクション・ランの間の変動すらある。したがって、トラッピング・セットは、2つの同一モデルのハード・ドライブの間でさえ変化し得る。そのように多数の異なるH D ドライブのL D P Cトラッピング・セットをシミュレートすることは、非実用的である。それでも、H D ドライブの大きいクラスに共通するトラッピング・セットだけをロードされたT S - R O Mは、特定のH D ドライブと対にされた時に、要求されるレベルのエラーフロア改善をもたらさない可能性がある。

【0056】

T S - R O Mの性能を改善する1つの方法は、製造されたデバイスのテスト・モデルから得られた結果を用いて、F P G Aベースのオフライン・シミュレーション・ツール(たとえば、図4のツール400)によって生成された情報を補足することである。通常、回路設計が最終化された後に、その設計の限られた個数のテスト・モデルが、量産が始まる前に製造され、テストのために配布される。特定のH D ドライブ実施態様について10<sup>-15</sup>のB E Rをもたらすトラッピング・セットを判定するのに1年を要する場合があるが

10

20

30

40

50

、 $10^{-12}$  の B E R をもたらすトラッピング・セットを判定するのには、1日のみを要する。したがって、テスト・モデルは、限られた時間期間の間に L D P C テスト・モードで動作させられ、発見されたトラッピング・セットのすべてが格納される。まだ T S - R O M 内にはない、発見されたトラッピング・セットのすべてが、T S - R O M に追加される。消費者に配布される実際のデバイスを使用することによって、この方法は、F P G A ベースのオフライン・シミュレーション・ツール（たとえば、図 4 のツール 4 0 0 ）に発見されなかつた可能性があるトラッピング・セットを取り込む。

#### 【 0 0 5 7 】

##### トラッピング・セット・ランダムアクセス・メモリ（T S - R A M ）

T S - R O M の静的トラッピング・セット・リストに対する 1 つのランタイム代替案は 10 、トラッピング・セット情報をトラッピング・セット・ランダムアクセス・メモリ（T S - R A M ）に格納し、図 4 のオフライン・トラッピング・セット・シミュレーション・ツール 4 0 0 を、実際の個々のデバイス（たとえば、H D ドライブ）に対して動作するランタイム・トラッピング・セット収集分析ツールにすることである。チャネルおよび信号モデル（たとえば、図 4 のモデル 4 0 4 ）から初期値を受け取るのではなく、ランタイム・ツールは、その特定のデバイスの実際の信号を処理する。ランタイム・ツールは、リスト復号器機能性を含む、すなわち、ランタイム・ツールは、観察された U S C を T S - R A M 内の格納されたトラッピング・セット情報を照合し、一致が見つかる場合には、格納された情報を使用して、復号器ビットノード値を変更し、復号器を再始動することを試みる。一致が見つからない場合には、ランタイム・ツールは、観察されたトラッピング・セットを分析する、すなわち、U S C に関する E B N を識別し、観察されたトラッピング・セットが、T S - R A M への格納に関するしきい要件（たとえば、D T S - N へのメンバシップ）を満足したかどうかを判定する。 20

#### 【 0 0 5 8 】

理論的に、上で説明した T S - R A M ツールは、すべての実施態様のトラッピング・セット・プロファイルに適合することができる。現実には、図 4 のオフライン・シミュレーション・ツール 4 0 0 によって実行されるトラッピング・セット / 支配的トラッピング・セット・シミュレーションは、計算的に複雑である。具体的に言うと、おそらくは数百万個のトラッピング・セットから支配的トラッピング・セットを構成することが、特に複雑である。この複雑さが、上で説明した T S - R A M ツールをほとんどの H D ドライブに不適切にする。通常、H D ドライブは、高いレート（たとえば、4 ギガビット毎秒）でデータを出力し、非常に低い B E R / エラーフロア・レート（たとえば、 $10^{-13}$  から  $10^{-15}$  ）を要求するが、そのファームウェアでささやかなコンピューティング・リソースのみを提供する。 30

#### 【 0 0 5 9 】

さらに、T S - R A M ツールは、図 4 のオフライン・シミュレーション・ツール 4 0 0 に似て、E B N インデックスを生成するために正しい符号語（C C W ）を必要とする。C C W は、オフライン・シミュレーション環境では簡単に入手可能であるが、ランタイム環境ではそうではない。

#### 【 0 0 6 0 】

本発明のある種の実施形態によれば、R O M 内の格納されたトラッピング・セット・プロファイルの編成のために、方法が実行される。トラッピング・セット・プロファイルは、優位性によってすなわち、L D P C 復号器のエラーフロア特性に対する影響によって、ランキングされる。より支配的なトラッピング・セット・プロファイルは、満足されないチェックノード（U S C ）と誤って満足されたチェックノード（M S C ）との両方に関する情報を含むが、より支配的でないトラッピング・セット・プロファイルは、U S C に関する情報をだけを含む。次に、トラッピング・セット・プロファイル情報は、複数のリンクされた階層データ・テーブルに編成され、この階層データ・テーブルは、ポインタチェース検索を使用する最も支配的な一致するトラッピング・セット・プロファイルのすばやい突止めおよび取出しを可能にする。 40 50

## 【0061】

本発明のある種の実施形態によれば、RAM内の支配的トラッピング・セットの収集および識別のために、効率的なランタイム方法が実行される。可能な場合には、新たに発見されたトラッピング・セットが、RAMに格納され、次に、トラッピング・セットが最後に一致した時以降にRAMが検索された回数、トラッピング・セットがRAMに追加された時以降にそのトラッピング・セットが一致した総回数、満足されないチェックノードの個数、および誤りのあるビットノードの個数という要因のうちのいずれか1つまたは複数に基づいてソートされ、またはランキングされる。低いランキングのトラッピング・セット・プロファイルは、新たに発見されたトラッピング・セット・プロファイルのための余地を作るためにRAMから削除される。したがって、図4のDTS-Nコンパイラ422で使用されるものなどの支配的トラッピング・セットの先駆的識別のための非常に計算的に複雑なオフライン方法を使用することに加えてまたはその代わりに、本発明のこれらの実施形態は、できる限り多数の新たに発見されたトラッピング・セットが格納され、非支配的トラッピング・セット・プロファイルが周期的なランキングおよび削除によって選別される、低い計算的複雑さの後天的方法を実行する。

## 【0062】

本発明の実施形態は、通常は、オンザフライ方法である、すなわち、LDPCL復号のために計上されたクロックサイクル内にDCCWを回復でき、したがって、システム・スループットに悪影響を与えない。

## 【0063】

図5は、本発明の一実施形態によるLDPCL復号システム500のブロック図である。図1の従来技術のHDドライブ100に類似する本発明のHDドライブでは、図5のLDPCL復号システム500は、図1のLDPCL復号器112に類似するLDPCL復号器の一部として実施される。その範囲で、図5の入力L<sub>ch</sub>値は、図1の復号器入力L<sub>ch</sub>値に類似し、図5の出力

## 【数52】

$$\hat{x}_{pp}$$

ベクトルは、図1の復号された情報語に類似する。

## 【0064】

LDPCL復号器502は、L<sub>ch</sub>値を受け取り、図3のLDPCL復号プロセス300を実行し、ベクトル

## 【数53】

$$\hat{x}$$

をポストプロセッサ504およびTS-RAMアップデータ506に出力する。ポストプロセッサ504は、後処理(PP)方法リスト508に接続され、PP方法リスト508は、後処理方法、たとえばTS-ROMリスト復号、TS-RAMリスト復号などを表す1つまたは複数の実行可能プログラムを含むメモリである。ポストプロセッサ504が、特定のPP方法を実行することを必要とする場合には、ポストプロセッサ504は、PP方法リスト508から実行可能プログラムを読み取り、そのプログラムを実行する。ポストプロセッサ504は、任意の個数のこれらのPP方法を並列にまたは順次実行することができる。ポストプロセッサ504は、ベクトル

## 【数54】

$$\hat{x}_{pp}$$

を出力し、これは、LDPCL復号システム500から出力されることに加えて、TS-RAMアップデータ506にも送られる。

10

20

40

50

## 【0065】

## データ・テーブル

実行中に、特定のPP方法が、PP方法実行可能プログラム・コードとは別々のデータ構造にアクセスすることを必要とする場合がある。具体的に言うと、TS-ROMリスト復号方法およびTS-RAMリスト復号方法は、それぞれTS-ROM 510およびTS-RAM 520に格納されたトラッピング・セット情報の1つまたは複数のリストにアクセスする。

## 【0066】

図5の例示的実施形態では、TS-ROM 510は、4つのテーブルすなわち、Bテーブル512、Pテーブル514、Eテーブル516、およびEIテーブル518を含む。TS-RAM 520は、2つのテーブルすなわち、RAM Pテーブル522およびRAMインデックス524を含む。テーブルは、1つまたは複数の等しいサイズの行(レコード)および1つまたは複数の等しいサイズの列(フィールド)に編成された、デジタル・データの2次元行列である。テーブルのレコードは、0から始めて上から下へ順番に番号を付けられる。この番号が、レコード番号である。

10

## 【0067】

Pテーブル514および522は、USCおよびそれに関連するEBNに関する情報を含む。Bテーブル512は、ROM Pテーブル514のポインタ情報を含む。EIテーブル518は、MSCに関する情報を含み、Eテーブル516は、EIテーブル518のポインタ情報を含む。RAMインデックス・テーブル524は、RAM Pテーブル522のポインタ情報を含む。

20

## 【0068】

図6は、図5のROM Pテーブル514の例示的レイアウトである。ROM Pテーブル514は、トラッピング・セット・プロファイル情報すなわち、USCインデックスおよびEBNインデックスを含む。ROM Pテーブル514は、各格納されたトラッピング・セットのUSCごとに1つの、複数のレコード(行)を含む。レコード番号602は、0から始まる、Pテーブル514内のレコードの順序位置である。

## 【0069】

ROM Pテーブル514内の各レコードは、3つのフィールドすなわち、AYER 604、USC\_INDEX 606、およびEBN\_INDEX 608を含む。一部のLDPD復号器は、更新動作のセットを並列に実行するように構成され、これを、レイヤと称する。AYER 604は、USCを含んだ復号レイヤの番号を示す。USC\_INDEX 606は、USCのインデックスを含む。EBN\_INDEX 608は、USCに関連する1つまたは2つのEBNのインデックスを含む。

30

## 【0070】

ROM Pテーブル514は、まずb(すなわち、

## 【数55】



内のUSCの個数)に基づいてソートされ、たとえば、b=2を有するすべてのトラッピング・セットが最初に現れ、すべてのb=3トラッピング・セットがそれに続くなどである。したがって、b=2範囲にトラッピング・セットごとに2つのレコード(たとえば、610、612)があり、最終的に、b=3を有するトラッピング・セットの3レコード・セット(たとえば、614、616)、b=4を有するトラッピング・セットの4レコード・セット(618、620)などが続く。

40

## 【0071】

各b範囲内では、トラッピング・セットは、優位性すなわち、そのトラッピング・セットがエラーフロア特性に対して有する影響によってソートされる。エラーフロア特性に対してより大きい影響を有するトラッピング・セットは、b範囲の始めに現れ、より少ない影響を有するトラッピング・セットは、終り近くに現れる。その後、特定のトラッピング

50

・セットのレコードは、U S C \_ I N D E X 6 0 6 によってソートされる。

【0072】

図7は、図5のBテーブル512の例示的レイアウトである。Bテーブル512は、ROM Pテーブル514内のb値ごとに、ROM Pテーブル514内およびEテーブル516内のそのb値の最初の出現へのポインタと、そのb値のEテーブル516内のレコードの個数とを含む。したがって、Bテーブル512内には、b値ごとに单一のレコードがあり、このb値は、レコード番号702によって示される。しかし、レコード番号702は0から始まるが、b値は、通常は、2以上から始まる。したがって、本発明のこの例示的実施形態では、レコード番号702にオフセットを加算して、対応するb値を作る。たとえば、b 2を有するトラッピング・セットだけが格納される場合には、2のオフセットが、対応するb値に達するために各レコード番号に加算される。 10

【0073】

フィールドP T A B L E \_ S T A R T \_ O F F S E T 704は、ROM Pテーブル514内の特定のb値の最初の出現の位置を含む。フィールドE T A B L E \_ S T A R T \_ O F F S E T 706は、Eテーブル516内の特定のb値の最初の出現の位置を含む。フィールドN U M \_ E T A B L E \_ E N T R I E S 708は、この特定のb値の、Eテーブル516内のレコードの個数を含む。

【0074】

図8は、図5のEテーブル516の例示的レイアウトである。Eテーブル516は、E Iテーブル518内のM S Cレコードへのポインタを含む。Eテーブル516内の各レコードは、レコード番号802、E I T A B L E \_ S T A R T \_ A D D R E S Sフィールド804、およびE I T A B L E \_ E N D \_ A D D R E S Sフィールド806を有する。E I T A B L E \_ S T A R T \_ A D D R E S Sフィールド804は、E Iテーブル518内の対応するデータの最初の出現へのポインタを含み、E I T A B L E \_ E N D \_ A D D R E S Sフィールド806は、E Iテーブル518内の対応するデータの最後の出現へのポインタを含む。 20

【0075】

図9は、図5のE Iテーブル518の例示的レイアウトである。E Iテーブル518は、M S Cに関連するE B Nのインデックスを格納する。E Iテーブル518内の各レコードは、レコード番号902を有し、2つのフィールドすなわちB L O C K \_ C O L U M Nフィールド904およびB \_ I N D E Xフィールド906を含む。B L O C K \_ C O L U M Nフィールド904は、E B Nが配置されたブロック列を示し、B \_ I N D E Xフィールド906は、E B Nのインデックスである。 30

【0076】

図10は、図5のR A M Pテーブル522の例示的レイアウトである。R A M Pテーブル522は、ROM Pテーブル514内では見つからない新たに識別されたトラッピング・セットのプロファイルを格納する。R A M Pテーブル522内の各行(すなわち、レコード)は、レコード番号1002を有し、2つのフィールドすなわち2ビットのT A Gフィールド1004およびR \_ W O R Dフィールド1006を含む。T A Gフィールド1004の4つの可能な値は、R \_ W O R Dフィールド1006内のレコード・タイプおよびデータの構造を示す。T A Gフィールド1004が11の値を有する場合には、レコードは、トラッピング・セット全体に関する情報を含むプライマリ・レコードである。T A Gフィールド1004が10の値を有する場合には、レコードは、セカンダリ・レコードであり、トラッピング・セット・プロファイル内の特定のU S Cに関する情報を含む。T A Gフィールド1004が00または01の値を有する場合には、R \_ W O R Dフィールド1006は空である、すなわち、このレコードは、新たに識別されたトラッピング・セット・プロファイル情報を格納するのに使用可能である。トラッピング・セット・プロファイルは、通常、单一のプライマリ・レコードとそれに続くb個のセカンダリ・レコードとを含む。 40

【0077】

レコードがプライマリ・レコードである場合には、R\_WORDフィールド1006は、4つのサブフィールドすなわち、(i)b値サブフィールド1008、(ii)トラッピング・セットEBNの個数を記録するa値サブフィールド1010、(iii)このトラッピング・セットに最後に一致したTS-RAM検索の個数を示すLAST\_HIT\_NUMサブフィールド1012、および(iv)このトラッピング・セットがTS-RAMに格納された時以降にこの特定のトラッピング・セット・プロファイルが観察されたトラッピング・セットと一致した回数を記録するHIT\_COUNTERサブフィールド1014を含む。

#### 【0078】

レコードがセカンダリ・レコードである場合には、R\_WORDフィールド1006は、トラッピング・セット内の単一のUSCのレイヤ(LAYERフィールド1016)およびインデックス(USC\_INDEXフィールド1018)と、そのUSCに関連する1つまたは複数のEBNのインデックス(EBN\_INDEXフィールド1020)とを含む。

#### 【0079】

図11は、図5のRAMインデックス・テーブル524の例示的レイアウトである。RAMインデックス・テーブル524内には、RAM\_Pテーブル522内のトラッピング・セット・プロファイルごとに1つのレコードがある。RAMインデックス・テーブル524内の各レコードは、単一のフィールドRAM\_PTABLE\_OFFSET\_1102を含む。RAM\_PTABLE\_OFFSET\_1102は、RAM\_Pテーブル522内の特定のトラッピング・セット・プロファイルの始めへのポインタである、すなわち、RAM\_PTABLE\_OFFSET\_1102は、トラッピング・セット・プロファイル・プライマリ・レコードの図10のレコード番号1002を含む。RAMインデックス・テーブル524内のレコードは、優位性によってソートされ、その結果、RAMインデックス・テーブル524内の最初のレコードは、RAM\_Pテーブル522内の最も支配的なトラッピング・セットをポイントし、RAMインデックス・テーブル524内の最後のレコードは、RAM\_Pテーブル522内の最も支配的でないレコードをポイントするようになる。

#### 【0080】

図12は、図5のLDPD復号システム500によって使用される例示的プロセス1200の流れ図である。処理は、ステップ1202で開始され、ステップ1204すなわち、図5のLDPD復号器502による $L_{ch}$ 値のLDPD復号に進む。ステップ1204でのLDPD復号がDCCWを作る場合には、プロセス1200は、ステップ1218で終了する。そうでない場合には、処理は、ステップ1206すなわち、TS-RAMリスト復号(図5のポストプロセッサ504によって実行される)に進む。

#### 【0081】

ステップ1206がDCCWを作る場合には、プロセス1200は、ステップ1218で終了する。そうでない場合には、処理は、ステップ1208すなわち、TS-RAMリスト復号(図5のポストプロセッサ504によって実行される)に進む。ステップ1208がDCCWを作る場合には、プロセス1200は、ステップ1218で終了する。そうでない場合には、処理は、1つまたは複数の追加の後処理方法1210、1212、...、1214(図5のポストプロセッサ504によって実行される)に進み、これらは、類似する形で動作する。

#### 【0082】

TS-RAMリスト復号1208または追加の後処理方法1210、1212、...、1214のいずれかがDCCWを作る場合には、処理はステップ1216に進み、ここで、図5のTS-RAM 520が、おそらくはTS-RAMアップデータ506によって更新される。その後、処理はステップ1218で終了する。

#### 【0083】

図12は、後処理方法1207～1214の1つの可能なシーケンシングを示す。後処

10

20

30

40

50

理方法のほとんどすべてのシーケンスを使用することができるが、いくつかのシーケンスは、他のシーケンスより実用的である。たとえば、ROMに既に格納されているトラッピング・セットがRAM内で重複しないことを保証するために、TS-ROMリスト復号をTS-RAMリスト復号の前にシーケンシングすることが望ましい。

【0084】

TS-ROMリスト復号

図13は、図5のポストプロセッサ504によって実施される図12の例示的なTS-ROMリスト復号プロセス1206の流れ図である。処理は、ステップ1302で開始され、ステップ1304に進み、ここで、プロセス1206は、図12のLDPC復号器1204から受け取られたベクトル

10

【数56】

$\hat{x}$

内で観察されたUSCの個数  $b_{observed}$  が、0を超え、プロセス1206によって効率的に処理できるUSCの最大個数  $b_{max}$  未満であるかどうかを判定する。 $b_{observed} = 0$  の場合には、ベクトル

【数57】

$\hat{x}$

内にUSCはなく(すなわち、

20

【数58】

$\hat{x}$

は、ニア・コードワード・ミスコレクションである)、したがって、照合すべきトラッピング・セットはない。ステップ1304が偽に評価される場合には、プロセス1206は終了する。そうでない場合には、処理は、ステップ1306に継続される。

【0085】

ステップ1306では、復号器の現在の状態を格納し、状態1というラベルを付ける。次に、処理はステップ1308に継続され、ここで、観察されたUSCを、まず復号レイヤによって、次にインデックスによってソートする。次に、ステップ1310で、次の4つの値を図5のBテーブル512からフェッチし、格納する。

30

(1)  $b = b_{observed}$  について、図7のPTABLE\_START\_OFFSETフィールド704、

(2)  $b = b_{observed} + 1$  について、図7のPTABLE\_START\_OFFSETフィールド704、

(3)  $b = b_{observed}$  について、図7のETABLE\_START\_OFFSETフィールド706、および

(4)  $b = b_{observed}$  について、図7のNUM\_ENTRYESフィールド708。

第1の値は、プロセス1206に、図5のPテーブル514内のどこで一致するトラッピング・セット情報の検索を開始すべきか(すなわち、USCインデックスおよびEBNインデックス)を指示し、第2の値は、プロセス1206に、いつその検索を終了すべきかを指示する。同様に、第3および第4の値は、プロセス1206に、拡張情報(すなわち、MSCインデックス)の検索を開始すべき場所および終了すべき場所を指示する。

40

【0086】

したがって、たとえば、 $b_{observed} = 5$  の場合には、プロセス1206は、 $b = 5$  および  $b = 6$  について図7のPTABLE\_START\_OFFSETフィールド704の値をフェッチし、 $b = 5$  について図7のETABLE\_START\_OFFSETフィールド706の値および図7のNUM\_ENTRYESフィールド708の値をフェッチする。

50

## 【0087】

次に、ステップ1312で、プロセス1206は、図5のPテーブル514を選択し、  
 $b = b_{\text{observed}}$ のP TABLE\_START\_OFFSETの格納された値によって示されるアドレスに行く。次に、ステップ1314で、TS-ROMから、観察されたUSCと一致するトラッピング・セットを検索する。

## 【0088】

図14は、図13の例示的なTS-ROM検索プロセス1314の流れ図である。プロセス1314は、ステップ1402で開始され、ステップ1404で、観察されたUSCの同形一致である図5のPテーブル514内の次のレコードを検索する。観察されたUSCの特定のセットの同形一致は、USCの個数およびこれらのUSCの間の距離が観察されたUSCと同一であるトラッピング・セットである。したがって、観察されたUSCが[1, 3, 10]である場合には、[1, 3, 10]は一致であり、[2, 4, 11]は同形一致であり、[3, 5, 12]、[4, 6, 13]なども同形一致である。一致が見つからない場合には、プロセス1314は、一致なしの状況1406で終了する。

10

## 【0089】

そうではなく、ステップ1404で一致が見つかる場合には、ステップ1408で、一致するPテーブル・レコードの図6のEBN\_INDEXフィールド608の値を格納する。EBN\_INDEXフィールドは、この一致するトラッピング・セットに関連する1つおよびおそらくは2つの誤りのあるビットノードのインデックスを格納する。

20

## 【0090】

次に、プロセス1314は、すべての拡張情報すなわち、この一致するトラッピング・セット内の誤って満足されたチェックノード(MSC)に関連するEBNのインデックスを突き止めることを試みる。拡張情報は、EIテーブル518内で保持される。しかし、拡張情報は、Pテーブル514内に格納されたすべてのトラッピング・セットについて保持されるのではなく、各b範囲内のトラッピング・セットのサブセットについてのみ保持される。そのサブセットは、特定のb範囲内のより支配的なトラッピング・セットすなわち、エラーフロア特性に対するより大きい影響を有するトラッピング・セットに対応する。上で述べたように、Pテーブル514内では、特定のb範囲内のトラッピング・セットが、優位性によってソートされ、したがって、拡張情報は、そのb範囲内の最初のx個のレコードのみについて保持される。EIテーブル518内の各b範囲の始めおよび終りは、Bテーブル512内のフィールド、ETABLE\_START\_OFFSET 706およびNUM\_ETABLE\_ENTRIES 708によって示される。

30

## 【0091】

プロセス1314は、ROM Pテーブル514のレコードから検索する時に、トラッピング・セットの内部カウントを維持する。したがって、たとえば、プロセス1314が、図6のPテーブル514内のb=2トラッピング・セットから検索している場合に、プロセス1314は、レコード0および1をトラッピング・セット0(たとえば、図6のトラッピング・セット610)として、レコード2および3をトラッピング・セット1(たとえば、図6のトラッピング・セット612)としてなど、識別する。このトラッピング・セット番号を、TSNUMと称する。

40

## 【0092】

ステップ1410で、TSNUMを、図13のステップ1310で格納されたNUM\_ETABLE\_ENTRIESフィールドの値と比較する。TSNUMがNUM\_ETABLE\_ENTRIESの値より大きい場合には、拡張情報は使用可能ではなく、プロセス1314は、一致、拡張情報なしの状況1412で終了する。

## 【0093】

そうではなく、ステップ1410で、TSNUMがNUM\_ETABLE\_ENTRIESの格納された値以下であることがわかる場合には、拡張情報が、この一致するトラッピング・セットについて存在する。ステップ1414で、TSNUMをETABLE\_START\_OFFSETの格納された値に加算して、変数ETABLE\_ENTRY\_A

50

D D R E S S を作る。

【 0 0 9 4 】

次に、ステップ 1 4 1 6 で、プロセス 1 3 1 4 は、図 5 の E テーブル 5 1 6 を選択し、変数 E T A B L E \_ E N T R Y \_ A D D R E S S の値と等しいアドレスを有するレコードに行き、図 8 の E I T A B L E \_ S T A R T \_ A D D R E S S フィールド 8 0 4 および図 8 の E I T A B L E \_ E N D \_ A D D R E S S フィールド 8 0 6 の値を格納する。

【 0 0 9 5 】

次に、ステップ 1 4 1 8 で、プロセス 1 3 1 4 は、図 5 の E I テーブル 5 1 8 を選択し、格納された E I T A B L E \_ S T A R T \_ A D D R E S S 値と E I T A B L E \_ E N D \_ A D D R E S S 値との間のすべてのレコードの、図 9 の B L O C K \_ C O L U M N フィールド 9 0 4 および図 9 の B \_ I N D E X フィールド 9 0 6 の値を格納する。最後に、プロセス 1 3 1 4 は、一致、拡張情報ありの状況 1 4 2 0 で終了する。

【 0 0 9 6 】

図 1 3 に戻って、ステップ 1 3 1 4 が、一致なしの状況で終了する場合には、プロセス 1 2 0 6 は、ステップ 1 3 1 6 で終了する。

【 0 0 9 7 】

ステップ 1 3 1 4 が、一致、拡張情報なしの状況で終了する場合には、プロセス 1 2 0 6 は、このトラッピング・セットに関連する U S C インデックスおよびいくつかの E B N インデックスを、拡張情報なしで（すなわち、M S C に関連する E B N のインデックスなしで）処理する。この場合に、ステップ 1 3 1 8 で、これらの E B N インデックスのビットノードを反転し、ステップ 1 3 2 0 で、反復 L D P C 復号を実行する。ステップ 1 3 2 0 の処理は、復号器初期化ステップ 3 0 4 および初期症候群チェック・ステップ 3 0 6 がスキップされることを除いて、図 3 のプロセス 3 0 0 と同一である。ステップ 1 3 2 0 が D C C W に収束する場合には、プロセス 1 2 0 6 は、ステップ 1 3 1 6 で終了する。そうでない場合には、ステップ 1 3 2 2 で、復号器を状態 1 に復元し、次の一致するトラッピング・セットを、ステップ 1 3 1 4 で検索する。

【 0 0 9 8 】

ステップ 1 3 1 4 が、一致、拡張情報ありの状況で終了する場合には、プロセス 1 2 0 6 は、一致するトラッピング・セットに関連するすべての E B N のインデックスを有する。この場合には、確率伝搬（たとえば、図 3 のステップ 3 1 0 および 3 1 2 ）を実行する必要はない。そうではなく、ステップ 1 3 2 4 で、E B N を反転し、結果のベクトル

【 数 5 9 】

$\hat{x}$

を、症候群チェック 1 3 2 6 にサブミットする。ベクトル

【 数 6 0 】

$\hat{x}$

が、ステップ 1 3 2 6 で症候群チェックに不合格である場合には、プロセス 1 2 0 6 は、ステップ 1 3 2 2 に進む。そうではなく、ベクトル

【 数 6 1 】

$\hat{x}$

が症候群チェックに合格する（すなわち、ベクトル

【 数 6 2 】

$\hat{x}$

が、有効な符号語である）場合には、ステップ 1 3 2 8 で、C R C チェックをベクトル

【数63】

 $\hat{x}$ 

に対して実行して、これが実際に正しい符号語であるかどうかを判定する。ベクトル

【数64】

 $\hat{x}$ 

がCRCチェック1328に合格する（すなわち、ベクトル

【数65】

 $\hat{x}$ 

10

がDCCWである）場合には、プロセス1206は、ステップ1316で終了する。ベクトル

【数66】

 $\hat{x}$ 

がCRCチェック1328に不合格である場合には、プロセス1206は、ステップ1322に進む。

【0099】

TS-RAMリスト復号

20

図5のポストプロセッサ504によって利用されるもう1つのPP方法が、図12のTS-RAMリスト復号1208すなわち、ランダムアクセス・メモリなどの揮発性メモリに格納されたトラッピング・セット情報を使用するトラッピング・セットのリスト復号である。TS-RAMリスト復号は、図5のRAM-Pテーブル522が選択されたトラッピング・セットのUSCおよびEBNの情報を格納するという点で、TS-ROMリスト復号に似ている。しかし、ROM-Pテーブル514とは異なって、RAM-Pテーブル522は、図5のTS-RAMアップデータ506によってランタイム中に変更される。したがって、最も重要な情報だけ、たとえばUSCインデックスおよびEBNインデックスだけが、格納される。拡張情報（たとえば、図5のEIテーブル518）は、TS-RAM内では維持されない。

30

【0100】

プロファイルも、RAM-Pテーブル522内にはどのような形であれソートされない。そうではなく、図5の別々のRAMインデックス・テーブル524は、優位性によってソートされた、RAM-Pテーブル522内に格納されたトラッピング・セット・プロファイルのアドレスのリストを維持する。

【0101】

図15は、図12の例示的なTS-RAMリスト復号プロセス1208の流れ図である。処理は、ステップ1502で開始され、ステップ1504に進み、このステップ1504は、目的および動作において図13のステップ1304と同一である。ステップ1504が偽に評価される場合には、プロセス1208は、ステップ1506で終了し、そうでない場合には、処理は、ステップ1508に継続され、ここで、現在の復号器状態を記録し、状態1というラベルを付ける。その後、処理はステップ1510に継続される。

40

【0102】

ステップ1510では、プロセス1208は、図5のRAM-Pテーブル522内の最も支配的なトラッピング・セット・プロファイルに行く。具体的に言うと、RAMインデックス・テーブル524は、優位性によってRAM-Pテーブル522内のプロファイルをランキングするので、プロセス1208は、RAMインデックス・テーブル524内の最初のレコードに行き、図11のRAM\_PTABLE\_OFFSETフィールド1102の値を取り出す。次に、プロセス1208は、RAM-Pテーブル522内のポインタをその格納されたオフセット値に移動する。

50

## 【0103】

ステップ1512で、プロセス1208は、T S - R A M検索が実行された総数すなわち、プロセス1208が実行された総回数を記憶するグローバル変数R A M \_ S E A R C H \_ C O U N Tを増分する。また、ステップ1512で、プロセス1208は、観察されたU S Cの同形一致について、R A Mインデックス・テーブル524によって示される順序すなわち減少する優位性の順で、R A M Pテーブル522内のプロファイルを調べる。一致が見つからない場合には、処理はステップ1514に継続される。

## 【0104】

そうではなく、ステップ1512で一致が見つかる場合には、ステップ1516で、一致したプロファイルの図10のL A S T \_ H I T \_ N U Mフィールド1012に、グローバル変数R A M \_ S E A R C H \_ C O U N Tの値をセットし、図10のH I T \_ C O U N T E Rフィールド1014を1つ増分する。次に、ステップ1518で、図10のE B N \_ I N D E Xフィールド1020の値を格納する。ステップ1520で、E B N \_ I N D E X値に配置されたビットノードを反転し、ステップ1522で、L D P C復号を実行する。ステップ1522は、図13のステップ1320と同一である。復号プロセス1522がD C C Wに収束する場合には、処理はステップ1514に継続され、そうでない場合には、ステップ1524で、復号器を状態1にリセットし、その後、処理は、ステップ1512に継続され、ここで、別の同形一致をPテーブル522内で検索する。

## 【0105】

ステップ1514では、プロセス1208は、図11のR A Mインデックス・テーブル524を更新する。具体的に言うと、ステップ1514で、R A M Pテーブル522内のフィールド、たとえばL A S T \_ H I T \_ N U M 1012、H I T \_ C O U N T E Rフィールド1014、U S Cノードの個数フィールド1008、およびE B Nの個数フィールド1010の任意の組合せに基づいて、R A M Pテーブル522内のすべてのT S - R A Mプロファイルをソートする。次に、すべてのソートされたプロファイルのアドレスすなわちすべてのプライマリ・レコードのレコード番号1002を、図11のR A Mインデックス・テーブル524内にレコードとして格納する。プロファイル・アドレスは、それらがステップ1514でソートされたのと同一の順序（たとえば、最も支配的から最も少なく支配的へ）でR A Mインデックス・テーブル524に格納される。

ステップ1514が完了した後に、処理は、ステップ1506で終了する。

## 【0106】

## T S - R A Mアップデータ

図12のプロセス1200の議論で説明したように、T S - R O Mリスト復号1206以外の任意の後処理方法がD C C Wに達する場合に、それが、新しいトラッピング・セットが発見されたことを意味する場合がある。そうである場合には、ステップ1216が、その新しいトラッピング・セットを図5のR A M Pテーブル522に追加することを試みる場合がある。

## 【0107】

一実施形態では、ステップ1216は、支配的トラッピング・セットをT S - R A M 520内で保持するための、低い複雑さのプロセスである。具体的に言うと、この実施形態では、ステップ1216は、図4のD T S - Nコンパイラ422によって実行される計算などの支配的トラッピング・セットを先驗的に判定するための網羅的計算を実行するのではなく、その代わりに、（i）1つまたは複数の要因、たとえば、トラッピング・セットが最後に一致した時以降にT S - R A Mが何回検索されたか、トラッピング・セットが一致した総回数、U S Cの個数、E B Nの個数などの任意の組合せに基づいてT S - R A Mトラッピング・セットをランキングし、次に、（ii）新たに発見されたトラッピング・セットのための余地を作るために、最も低いランキングのトラッピング・セットを追い出す。これらの要因を使用するT S - R A M 520内のトラッピング・セットのランキングは、通常、オフライン・シミュレーション・ツール（たとえば、図4のコンパイラ422）によって実行される分析よりかなりより少なく複雑である。

10

20

30

40

50

## 【0108】

図16は、図12の例示的なTS-RAM更新プロセス1216の流れ図である。処理は、ステップ1602で開始され、ステップ1604に進み、ここで、DCCWが図12のTS-RAMリスト復号プロセス1208によって生成されたかどうかを判定する。そうである場合には、トラッピング・セット・プロファイルをTS-RAMに追加する必要はなく、プロセス1216は、ステップ1608で終了する。

## 【0109】

そうではなく、ステップ1604がno/偽に評価される場合には、これは、TS-ROMリスト復号またはTS-RAMリスト復号以外のある後処理方法がDCCWに達し、したがって、新しいトラッピング・セットが発見され、RAMに付加されなければならぬことを意味する。ステップ1610で、トラッピング・セット・プロファイルを生成する。トラッピング・セット・プロファイルは、トラッピング・セットUSBのインデックスと、これらのUSBに関連するEBNのインデックスとを含む。USBインデックスは、図5のLDPD復号器502によって既に生成済みである。EBNインデックスを生成するために、ステップ1610で、ポストプロセッサ504によって生成されたDCCW

## 【数67】

$$\hat{x}_{pp}$$

を、LDPD復号器502によって生成されたベクトル

20

## 【数68】

$$\hat{x}$$

と比較する。

## 【0110】

ステップ1612で、新しいトラッピング・セット・プロファイルを付加するのに十分な空き空間があるかどうかを判定する。そうである場合には、ステップ1614で、新しいトラッピング・セット・プロファイルを、RAM\_Pテーブル522に付加し、プロセス1216は、ステップ1616に進む。

## 【0111】

しかし、ステップ1612で、新しいトラッピング・セット・プロファイルを付加するのに十分な空き空間がRAM\_Pテーブル522内にない場合には、ステップ1618で、最も低いランキングの追出しに適するトラッピング・セット・プロファイルを追い出す。この例では、プロファイルの追出し適格性は、そのプロファイルが最後に一致した時以降にRAMが検索された回数すなわち、グローバル変数RAM\_SEARCH\_COUNTの値からプロファイルのLAST\_HIT\_NUMフィールドの値を引いたものによって決定される。介在する検索の回数が、指定されたしきい値より多い場合には、そのプロファイルは、追出しに適する。すべてのプロファイルが、まず追出し適格性によってランキングされると仮定すると、すべての追出しに適するレコードは、図11のRAMインデックス・テーブル524の終りにある。

30

## 【0112】

したがって、ステップ1618では、RAMインデックス・テーブル524内の最後のレコードを選択し、RAM\_PTABLE\_OFFSETフィールド1102の値を取り出す。RAM\_Pテーブル522内で格納されたオフセット値(取り出されたRAM\_PTABLE\_OFFSET値によって示される)に配置されたプロファイルが、追出しに適する場合には、RAM\_Pテーブル522内の格納されたオフセット値に配置されたプライマリ・レコードおよび関連するセカンダリ・レコードが削除される。ステップ1620で、RAMインデックス・テーブルMSBを更新し、その後、制御はステップ1612にループ・バックし、ここで、新しいトラッピング・セット・プロファイルを付加するのに十分な空き空間がRAM\_Pテーブル522内にあるかどうかを判定する。ステップ1

40

50

620の処理は、図15のステップ1514の処理と同一である。

【0113】

そうではなく、ステップ1618で、最低ランキングのプロファイルが追出しに適さない場合には、処理は、ステップ1616に継続される。ステップ1616では、RAMインデックス・テーブル524を更新し、処理は、ステップ1608で終了する。ステップ1616の処理は、図15のステップ1514の処理と同一である。

【0114】

本発明を、LDPCコーディングおよび復号を実施するハード・ディスク・ドライブの文脈で説明したが、本発明は、それに限定はされない。一般に、本発明を、LDPCコーディングおよび復号を用いる任意の適切な通信経路内で実施することができる。

10

【0115】

さらに、上で使用された例示的な確率伝搬アルゴリズムは、オフセットmin-sumアルゴリズム(OMS)であるが、本発明は、それに限定はされず、本発明を、任意の確率伝搬変形、たとえば、sum-productアルゴリズム(SPA)またはBahl-Cocke-Jelinek-Raviv(BCJR)アルゴリズムと共に使用することができる。

【0116】

さらに、上で使用された確率伝搬の例は、すべてのチェックノードが单一のチェックノード更新ステップ中に更新され、その後、すべてのビットノードが单一のビットノード更新ステップで更新される、特定の復号スケジュール(フラッディング・スケジュール)を使用したが、本発明は、それには限定されず、本発明を、任意の復号スケジュール、たとえば、ローシリアルスケジュール、カラムシリアルスケジュール、およびローカラム・シリアルスケジュールと共に使用することができる。

20

【0117】

さらに、上で使用された例示的LDPC復号器は、非階層復号器であったが、本発明は、それに限定はされず、本発明を、階層復号器と非階層復号器との両方と共に使用することができる。

【0118】

さらに、上で与えた例示的TS-RAM実施態様は、トラッピング・セット・プロファイルをHDドライブの読み取りチャネル内のRAMに格納すると仮定したが、本発明は、それに限定はされない。RAM-Pテーブル(たとえば、図5の522)を、HDドライブのプラッタに格納するか、フラッシュ・メモリなどの別々のメモリ内に格納することもできる。

30

【0119】

さらに、上で与えた例示的TS-ROM実施態様は、読み取り専用メモリの文脈で説明されたが、本発明は、それに限定はされない。一般に、本明細書と特許請求の範囲との両方で使用される時に、用語「ROM」は、そのデバイス内のデータを変更できるか否かにかかわりなく、静的TSプロファイル・データを格納するすべてのデータストレージ・デバイスを指すと解釈されなければならない。

【0120】

40

さらに、本発明の実施形態を、LDPC符号の文脈で説明したが、本発明は、それに限定はされない。本発明の実施形態を、グラフによって定義できる任意の符号、たとえば、トルネード符号、構造化IRA符号のために実施することができる。というのは、これが、トラッピング・セットに苦しむグラフ定義される符号であるからである。

【0121】

本発明を、対数尤度比の受取りに関して説明したが、本発明は、それに限定はされない。尤度比などの他のソフト値または硬判定値が処理される、本発明の実施形態を想定することができる。

【0122】

本発明を、方法およびこれら的方法を実践する装置の形で実施することができる。本発

50

明を、磁気記録媒体、光記録媒体、ソリッド・ステート・メモリ、フロッピ・ディスクエット、CD-ROM、ハード・ドライブ、または任意の他の機械可読記憶媒体などの有形の媒体内で実施されるプログラム・コードの形で実施することもでき、プログラム・コードがコンピュータなどの機械にロードされ、その機械によって実行される時に、その機械は、本発明を実践する装置になる。本発明を、記憶媒体に格納される、機械にロードされ、かつ／またはこれによって実行される、あるいは電気配線またはケーブリングを介する、光ファイバを介する、もしくは電磁放射を介するなどのある伝送媒体または搬送波を介して伝送されるいずれであれ、プログラム・コードの形で実施することもでき、プログラム・コードがコンピュータなどの機械にロードされ、その機械によって実行される時に、その機械は、本発明を実践する装置になる。汎用プロセッサ上で実施される時に、プログラム・コード・セグメントは、プロセッサと組み合わさせて、特定の論理回路に類似して動作する独自のデバイスをもたらす。

#### 【0123】

そうではないと明示的に述べられない限り、各数値および範囲は、単語「約」(“about” or “approximately”)が値または範囲の値に先行するかのように近似として解釈されなければならない。

#### 【0124】

さらに、添付の特許請求の範囲に表された本発明の範囲から逸脱せずに、本発明の性質を説明するために説明され、図示された部分の詳細、材料、および配置におけるさまざまな変更を当業者が行えることを理解されたい。

#### 【0125】

特許請求の範囲での図面番号および／または図面符号の使用は、特許請求の範囲の解釈を容易にするために請求される主題の1つまたは複数の可能な実施形態を識別することを意図されたものである。そのような使用を、これらの特許請求の範囲の範囲を対応する図面に示された実施形態に必ず限定するものとして解釈してはならない。

#### 【0126】

本明細書に示された例示的方法のステップが、必ずしも説明された順序で実行されることを必要とせず、そのような方法のステップの順序が、単に例示と理解されなければならないことを理解されたい。同様に、本発明のさまざまな実施形態と一貫する方法において、追加ステップをそのような方法に含めることができ、ある種のステップを削除したまは組み合わせることができる。

#### 【0127】

次の方法請求項の要素は、存在する場合に、対応するラベル付けを伴って特定のシーケンスで詳説されるが、請求項詳説が、これらの要素の一部またはすべてを実施するための特定のシーケンスを他の形で暗示しない限り、これらの要素は、必ずしもその特定のシーケンスで実施されることに限定はされないことが意図されている。

#### 【0128】

本明細書での「一実施形態」(“one embodiment” or “an embodiment”)への言及は、その実施形態に関連して説明される特定の特徴、構造、または特性を、本発明の少なくとも1つの実施形態に含めることができることを意味する。本明細書のさまざまな場所での句「一実施形態で」の出現は、必ずしもすべてが同一の実施形態に言及するものではなく、必ずしも他の実施形態と相互に排他的な別々のまたは代替の実施形態でもない。同一のことが、用語「実施態様」にもあてはまる。

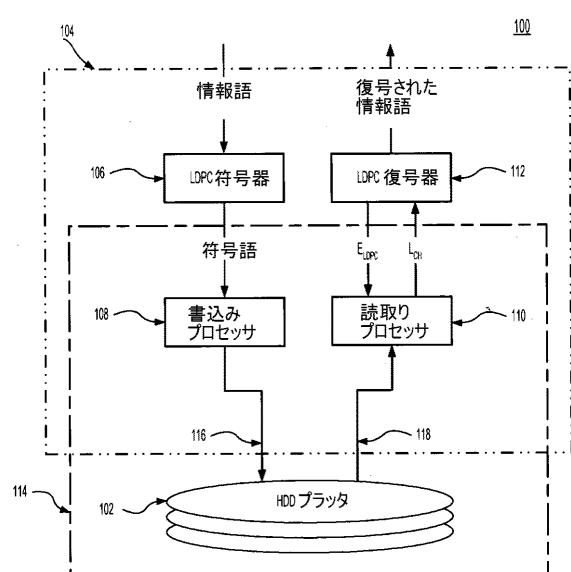
10

20

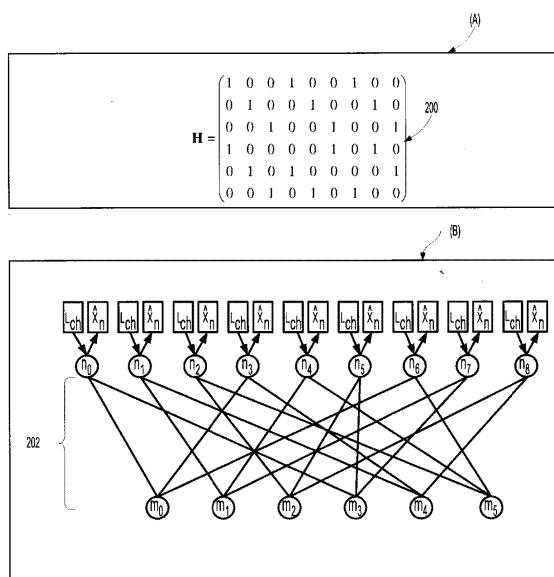
30

40

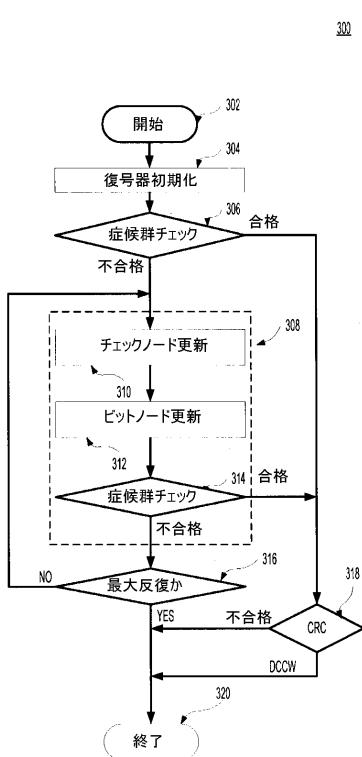
【図1】



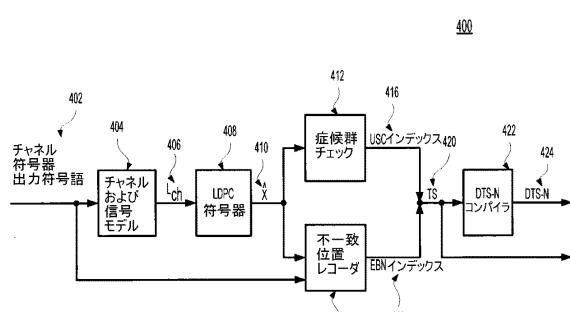
【図2】



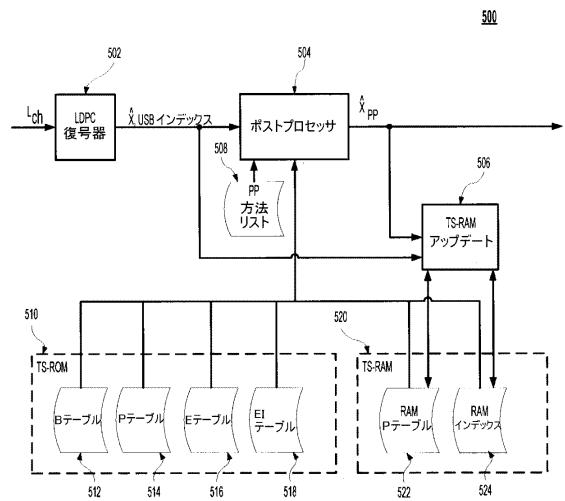
【図3】



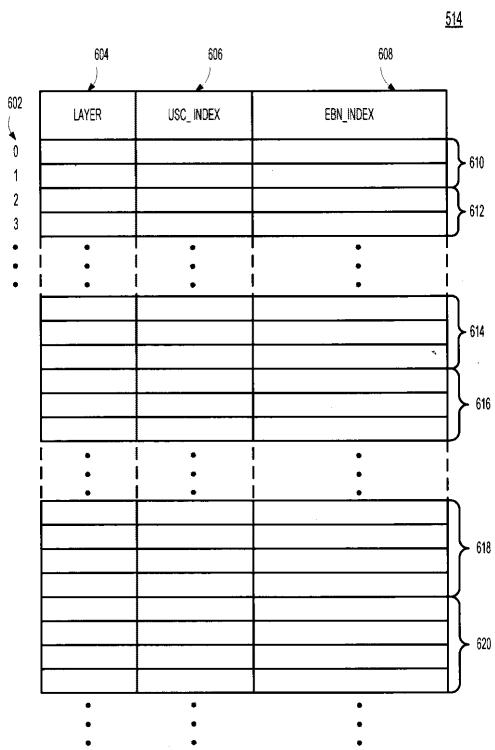
【図4】



【図5】



【図6】



【図7】

702	704	706	708	512
0	PTABLE_START_OFFSET	ETABLE_START_OFFSET	NUM_ETABLE_ENTRIES	
1				
2				
3				
...	...	...	...	
bMax-2				

【図8】

802	804	806	516
0	EITABLE_START_ADDRESS	EITABLE_END_ADDRESS	
1			
2			
3			
...	...	...	

【図 9】

518

902	904	906
0	BLOCK_COLUMN	B_INDEX
1		
2		
3		
⋮	⋮	⋮

【図 10】

522

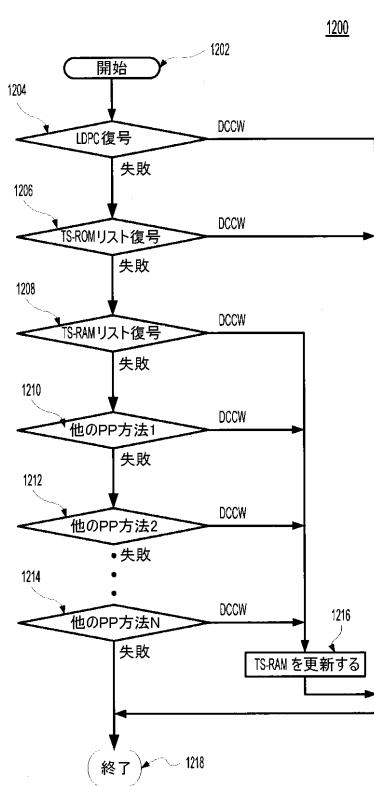
1002	R_WORD				
0	11	b	a	LAST_HIT_NUM	HIT_COUNTER
1	10	LAYER	1018	USC_INDEX	EBN_INDEX
2	10	LAYER	1018	USC_INDEX	EBN_INDEX
3	10	LAYER	1018	USC_INDEX	EBN_INDEX
⋮	11	b	a	LAST_HIT_NUM	HIT_COUNTER
10	LAYER	1018	USC_INDEX	EBN_INDEX	
10	LAYER	1018	USC_INDEX	EBN_INDEX	
10	LAYER	1018	USC_INDEX	EBN_INDEX	
01	[空の位置]				1014
00	[空の位置]				1020
⋮	⋮	⋮	⋮	⋮	1022

【図 11】

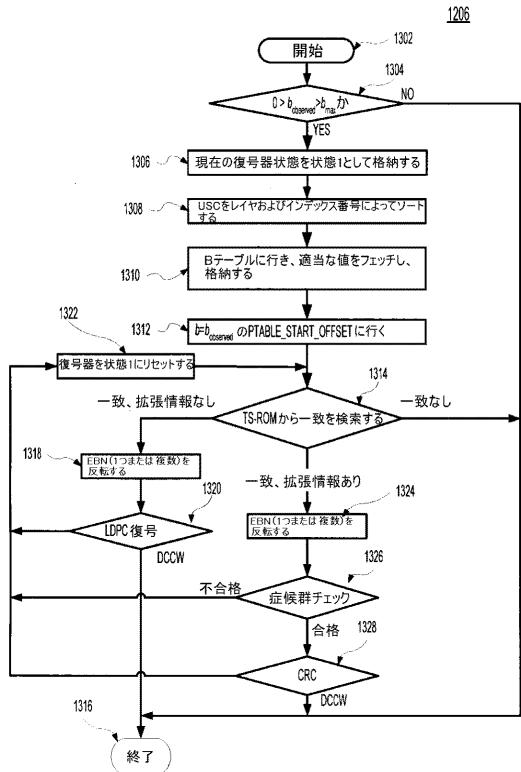
524

1102
RAM_PTABLE_OFFSET
⋮
⋮
⋮

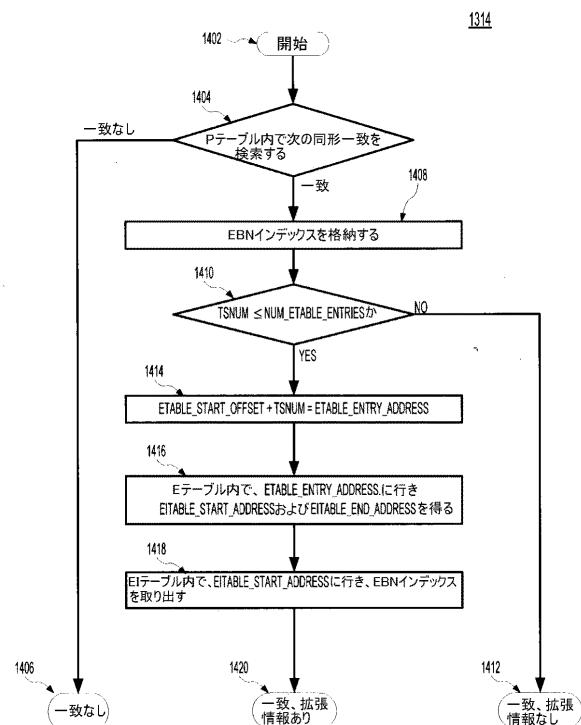
【図 12】



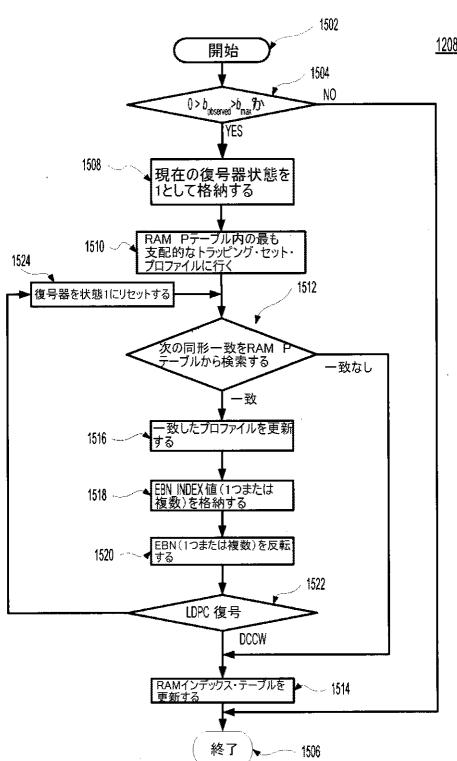
【図13】



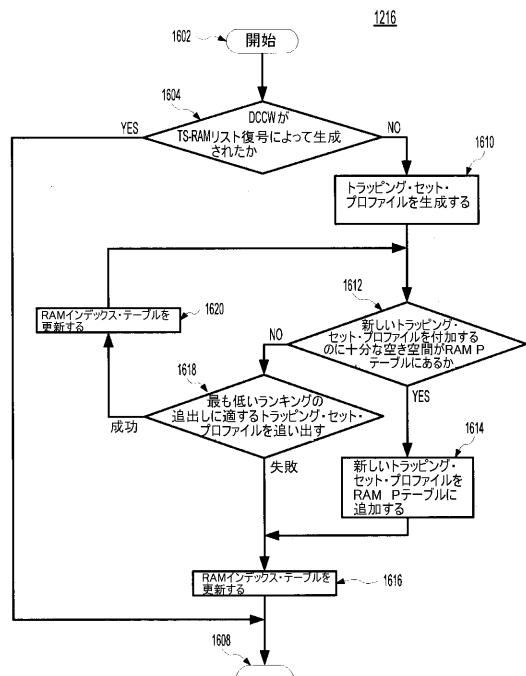
【図14】



【図15】



【図16】



---

フロントページの続き

(72)発明者 ガンナム , キラン

アメリカ合衆国 95134 カリフォルニア , サン ホセ , エラン ヴィレッジ レーン 37  
1 , アパートメント 226

審査官 岡 裕之

(56)参考文献 特表2012-500512 (JP, A)

特表2011-525771 (JP, A)

特開2008-141439 (JP, A)

Enver Cavus et al. , A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC codes , Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on , 2005年 9月14日 , Vol.4 , pp.2386-2390

Enver Cavus et al. , Low BER performance estimation of LDPC codes via application of importance sampling to trapping sets , Communications, IEEE Transactions on , 2009年 7月 , Vol.57, No.7 , pp.1886-1888

Enver Cavus et al. , An IS Simulation Technique for Very Low BER Performance Evaluation of LDPC Codes , Communications, 2006. ICC '06. IEEE International Conference on , 2006年 6月 , Vol.3 , pp.1095-1100

(58)調査した分野(Int.Cl. , DB名)

H 03M 13 / 19

G 11 B 20 / 10

I E E E X p l o r e

C i N i