

(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2019-54448  
(P2019-54448A)

(43) 公開日 平成31年4月4日(2019.4.4)

(51) Int.Cl.  
H03M 13/45 (2006.01)

F I  
H03M 13/45

テーマコード(参考)  
5J065

審査請求 未請求 請求項の数 17 O L (全 19 頁)

(21) 出願番号 特願2017-178095 (P2017-178095)  
(22) 出願日 平成29年9月15日 (2017.9.15)

(71) 出願人 318010018  
東芝メモリ株式会社  
東京都港区芝浦一丁目1番1号  
(74) 代理人 110002147  
特許業務法人酒井国際特許事務所  
(72) 発明者 木船 尚子  
東京都港区芝浦一丁目1番1号 東芝メモリ株式会社内  
(72) 発明者 内川 浩典  
東京都港区芝浦一丁目1番1号 東芝メモリ株式会社内  
(72) 発明者 渡邊 大毅  
東京都港区芝浦一丁目1番1号 東芝メモリ株式会社内  
Fターム(参考) 5J065 AB01 AC03 AD03 AE06 AF01

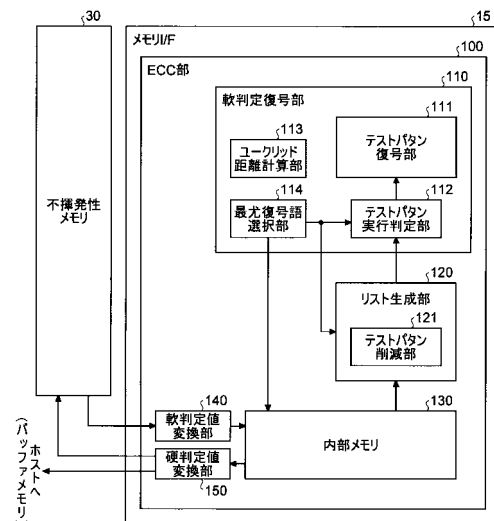
(54) 【発明の名称】 メモリシステム

(57) 【要約】

【課題】 誤り訂正能力を低下させずにレイテンシを高める。

【解決手段】 実施形態に係るメモリシステムは、複数のテストパターンから中間復号語を検出するテストパターン復号部と、前記テストパターン復号部で検出された中間復号語と前記受信語とのユークリッド距離を計算するユークリッド距離計算部と、最尤復号語候補を保持する最尤復号語選択部とを備え、前記最尤復号語選択部は、前記テストパターン復号部で検出された中間復号語である第1の中間復号語のユークリッド距離が、前記保持している最尤復号語候補のユークリッド距離よりも短い場合、前記第1の中間復号語で前記保持している最尤復号語候補を更新し、前記テストパターン復号部は、中間復号語のユークリッド距離が前記最尤復号語選択部に保持されている前記最尤復号語候補の前記ユークリッド距離よりも短くなる可能性のないテストパターンに対する復号を実行しない。

【選択図】 図2



## 【特許請求の範囲】

## 【請求項 1】

不揮発性メモリと、  
前記不揮発性メモリから読み出された受信語を軟判定値の符号語に変換する軟判定値変換部と、

前記軟判定値の符号語に対する複数のテストパタンのリストを生成するリスト生成部と

、  
前記リストに含まれるテストパターンから中間復号語を検出するテストパターン復号部と、  
前記テストパターン復号部で検出された中間復号語と前記受信語とのユークリッド距離を計算するユークリッド距離計算部と、

最尤復号語候補を保持する最尤復号語選択部と、

を備え、

前記最尤復号語選択部は、前記テストパターン復号部で検出された中間復号語である第 1 の中間復号語のユークリッド距離が、前記保持している最尤復号語候補のユークリッド距離よりも短い場合、前記第 1 の中間復号語で前記保持している最尤復号語候補を更新し、最終的に保持している前記最尤復号語候補を軟判定出力値として出力し、

前記テストパターン復号部は、中間復号語のユークリッド距離が前記最尤復号語選択部に保持されている前記最尤復号語候補の前記ユークリッド距離よりも短くなる可能性のないテストパターンに対する復号を実行しない

メモリシステム。

## 【請求項 2】

前記リスト生成部は、前記テストパターンそれぞれがフリップ対象とするビットの軟判定値の絶対値に基づいて、前記テストパターン復号部で検出される符号語のユークリッド距離が前記最尤復号語選択部に保持されている前記最尤復号語候補の前記ユークリッド距離よりも短くなる可能性のない前記テストパターンを含めずに前記リストを生成する請求項 1 に記載のメモリシステム。

## 【請求項 3】

前記テストパターンそれぞれがフリップ対象とするビットの軟判定値の絶対値に基づいて、前記テストパターン復号部で検出される符号語のユークリッド距離が前記最尤復号語選択部に保持されている前記最尤復号語候補の前記ユークリッド距離よりも短くなる可能性のないテストパターンを前記リストから削除するテストパターン削減部をさらに備える請求項 1 に記載のメモリシステム。

## 【請求項 4】

前記テストパターンそれぞれがフリップ対象とするビットの軟判定値の絶対値に基づいて、前記リスト生成部が生成した前記リストに含まれる前記テストパタンのうち、前記テストパターン復号部で検出される符号語のユークリッド距離が前記最尤復号語選択部に保持されている前記最尤復号語候補の前記ユークリッド距離よりも短くなる可能性のないテストパターンに対する前記テストパターン復号部による復号をスキップするテストパターン実行判定部をさらに備える請求項 1 に記載のメモリシステム。

## 【請求項 5】

前記リスト生成部は、フリップするビットの数ごとに異なるテストパタンのリストを生成し、

前記テストパターン復号部は、前記フリップするビットの数を 0 とした場合を含む前記フリップするビットの数の小さいリストに含まれるテストパターンから順に復号を実行する請求項 2 に記載のメモリシステム。

## 【請求項 6】

前記テストパターン復号部は、前記リストに含まれる前記テストパタンのうち、信頼度の低いビットをフリップ対象とするテストパターンを優先的に復号する請求項 1 に記載のメモリシステム。

## 【請求項 7】

前記テストパターン復号部は、前記リストに含まれる前記テストパタンのうち、信頼度の低いビットをフリップ対象とするテストパターンを優先的に復号する請求項 1 に記載のメモリシステム。

10

20

30

40

50

前記ユークリッド距離計算部は、前記リスト生成部でフリップされたビットと前記テストパターン復号部で検出される1つ以上のエラービットに対応する1つ以上の軟判定値の絶対値の総和を前記ユークリッド距離として算出する請求項1に記載のメモリシステム。

【請求項8】

前記所定のアルゴリズムは、Chase復号アルゴリズムであり、

前記リスト生成部は、前記符号語の訂正可能数を $t$ とし、フリップするビットの数を $f$ とし、 $a$ を任意の自然数とした場合であって、前記最尤復号語選択部が $(f+t)a+t+1$ のユークリッド距離の最尤復号語候補を保持している場合、フリップ対象とするビットの軟判定値の絶対値が前記 $a$ 以下であるテストパターンが含まれない前記リストを生成する

10

請求項2に記載のメモリシステム。

【請求項9】

前記所定のアルゴリズムは、Chase復号アルゴリズムであり、

前記テストパターン削減部は、前記符号語の訂正可能数を $t$ とし、フリップするビットの数を $f$ とし、 $a$ を任意の自然数とした場合であって、前記最尤復号語選択部が $(f+t)a+t+1$ のユークリッド距離の最尤復号語候補を保持している場合、フリップ対象とするビットの軟判定値の絶対値が前記 $a$ 以下であるテストパターンを前記リストから削除する請求項3に記載のメモリシステム。

【請求項10】

前記所定のアルゴリズムは、Chase復号アルゴリズムであり、

前記テストパターン実行判定部は、前記符号語の訂正可能数を $t$ とし、フリップするビットの数を $f$ とし、 $a$ を任意の自然数とした場合であって、前記最尤復号語選択部が $(f+t)a+t+1$ のユークリッド距離の最尤復号語候補を保持している場合、フリップ対象とするビットの軟判定値の絶対値が前記 $a$ 以下であるテストパターンに対する前記テストパターン復号部による復号をスキップする

20

請求項4に記載のメモリシステム。

【請求項11】

前記所定のアルゴリズムは、OSD (Ordered Statistics Decoding) アルゴリズムであり、

前記リスト生成部は、前記最尤復号語選択部に前記ユークリッド距離が $c$ の前記最尤復号語候補が保持されている場合、フリップ対象とするビットの軟判定値の絶対値の和が前記 $c$ より大きいテストパターンが含まれない前記リストを生成する

30

請求項2に記載のメモリシステム。

【請求項12】

前記所定のアルゴリズムは、OSDアルゴリズムであり、

前記テストパターン削減部は、前記最尤復号語選択部に前記ユークリッド距離が $c$ の前記最尤復号語候補が保持されている場合、フリップ対象とするビットの軟判定値の絶対値の和が前記 $c$ より大きいテストパターンを前記リストから削除する

請求項3に記載のメモリシステム。

【請求項13】

前記所定のアルゴリズムは、OSDアルゴリズムであり、

前記テストパターン実行判定部は、前記最尤復号語選択部に前記ユークリッド距離が $c$ の前記最尤復号語候補が保持されている場合、フリップ対象とするビットの軟判定値の絶対値の和が前記 $c$ より大きいテストパターンに対する前記テストパターン復号部による復号をスキップする

40

請求項4に記載のメモリシステム。

【請求項14】

前記所定のアルゴリズムは、OSDアルゴリズムであり、

前記リスト生成部は、前記最尤復号語選択部に前記ユークリッド距離が $c$ の前記最尤復号語候補が保持されている場合であって、前記符号語の訂正可能数を $t$ とし、フリップす

50

るビットの数を  $j$  とし、前記符号語において前記軟判定値の絶対値の最も小さいビットから順に  $(t + 1 - j)$  個のビットの軟判定値の絶対値の和を  $m$  とした場合、フリップ対象とするビットの軟判定値の絶対値の和が前記  $c - m$  より大きいテストパターンが含まれない前記リストを生成する

請求項 2 に記載のメモリシステム。

【請求項 15】

前記所定のアルゴリズムは、OSD アルゴリズムであり、

前記テストパターン削減部は、前記最尤復号語選択部に前記ユークリッド距離が  $c$  の前記最尤復号語候補が保持されている場合であって、前記符号語の訂正可能数を  $t$  とし、フリップするビットの数を  $j$  とし、前記符号語において前記軟判定値の絶対値の最も小さいビットから順に  $(t + 1 - j)$  個のビットの軟判定値の絶対値の和を  $m$  とした場合、フリップ対象とするビットの軟判定値の絶対値の和が前記  $c - m$  より大きいテストパターンを前記リストから削除する

10

請求項 3 に記載のメモリシステム。

【請求項 16】

前記所定のアルゴリズムは、OSD アルゴリズムであり、

前記テストパターン実行判定部は、前記最尤復号語選択部に前記ユークリッド距離が  $c$  の前記最尤復号語候補が保持されている場合であって、前記符号語の訂正可能数を  $t$  とし、フリップするビットの数を  $j$  とし、前記符号語において前記軟判定値の絶対値の最も小さいビットから順に  $(t + 1 - j)$  個のビットの軟判定値の絶対値の和を  $m$  とした場合、フリップ対象とするビットの軟判定値の絶対値の和が前記  $c - m$  より大きいテストパターンに対する前記テストパターン復号部による復号をスキップする

20

請求項 4 に記載のメモリシステム。

【請求項 17】

前記最尤復号語選択部から出力された前記軟判定出力値を硬判定値の符号語に変換する硬判定値変換部をさらに備える請求項 1 に記載のメモリシステム。

【発明の詳細な説明】

【技術分野】

【0001】

以下の実施形態は、一般的に、メモリシステムに関する。

30

【背景技術】

【0002】

メモリシステムでは、一般に、記憶するデータを保護するために、符号化されたデータが記憶される。このため、メモリシステムに記憶されたデータを読み出す際には、誤り訂正符号化されたデータに対する復号が行われる。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】米国特許第 8 3 4 1 5 0 2 号明細書

【特許文献 2】米国特許第 8 4 6 8 4 3 1 号明細書

40

【非特許文献】

【0004】

【非特許文献 1】Antoine Valembois and Marc Fossorier, Senior Member, IEEE, "Box and Match Techniques Applied to Soft-Decision Decoding" IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 50, NO. 5, MAY 2004.

【発明の概要】

【発明が解決しようとする課題】

【0005】

本発明の一つの実施形態は、誤り訂正能力を低下させずにレイテンシを高めることが可能なメモリシステムを提供することを目的とする。

50

## 【課題を解決するための手段】

## 【0006】

実施形態に係るメモリシステムは、不揮発性メモリと、前記不揮発性メモリから読み出された受信語を軟判定値の符号語に変換する軟判定値変換部と、前記軟判定値の符号語に対する複数のテストパタンのリストを生成するリスト生成部と、前記リストに含まれるテストパターンから中間復号語を検出するテストパターン復号部と、前記テストパターン復号部で検出された中間復号語と前記受信語とのユークリッド距離を計算するユークリッド距離計算部と、最尤復号語候補を保持する最尤復号語選択部とを備え、前記最尤復号語選択部は、前記テストパターン復号部で検出された中間復号語である第1の中間復号語のユークリッド距離が、前記保持している最尤復号語候補のユークリッド距離よりも短い場合、前記第1の中間復号語で前記保持している最尤復号語候補を更新し、最終的に保持している前記最尤復号語候補を軟判定出力値として出力し、前記テストパターン復号部は、中間復号語のユークリッド距離が前記最尤復号語選択部に保持されている前記最尤復号語候補の前記ユークリッド距離よりも短くなる可能性のないテストパターンに対する復号を実行しない。

10

## 【図面の簡単な説明】

## 【0007】

【図1】図1は、第1の実施形態に係るメモリシステムの概略構成例を示すブロック図である。

【図2】図2は、第1の実施形態に係るメモリI/Fのより詳細な構成例を示すブロック図である。

20

【図3】図3は、第1の実施形態における受信語のLLR列のソートを説明するための図である。

【図4】図4は、第1の実施形態に係るソート後のLLR列に対するフリップ対象範囲を示す図である。

【図5】図5は、第1の実施形態に係る最尤復号語選択部が選択する最尤復号語候補を説明するための図である。

【図6】図6は、第1の実施形態に係る復号動作の一例を示すフローチャートである。

【図7】図7は、第2の実施形態に係るソート後のLLR列に対するシンドロームサイズ及びフリップ対象範囲を示す図である。

## 【発明を実施するための形態】

30

## 【0008】

以下に添付図面を参照して、実施形態に係るメモリシステムを詳細に説明する。なお、以下の実施形態により本発明が限定されるものではない。

## 【0009】

最尤復号法を使った軟判定復号では、メモリセルに記憶されている各ビットの(0, 1)の値を“0である確率”で示したものを入力(受信語)とし、採用する復号アルゴリズムを使用して受信語から復号語の候補となる符号語のリストを生成する。なお、復号対象の符号語は、たとえば列方向の成分符号と行方向の成分符号との2次元の成分符号でユーザデータを二重に保護する積符号に代表されるような、2次元以上の成分符号でユーザデータの少なくとも一部を二重以上に保護する多次元の誤り訂正符号であってよい。その場合、説明文中の符号語は、多次元の誤り訂正符号における成分符号に相当する。このような多次元の誤り訂正符号には、上述した積符号の他に、積符号を一般化した概念であるグラフ符号(Graph Codes)や、グラフ符号を更に一般化した概念である一般化LDPC符号(Generalized Low-Density Parity Check Codes)などが存在する。

40

## 【0010】

つづいて、受信語と復号語候補の符号語とがどの程度かけ離れているかを表すユークリッド距離といわれるメトリック(正しい符号語であることの尤もらしさを表す指標)を計算し、ユークリッド距離が最も小さい復号語候補の符号語を最尤復号語の候補(以下、最尤復号語候補という)として選択する。その後、最尤復号語候補として選択した符号語の軟判定出力値を計算して出力する。最尤復号語候補となる符号語のリストを生成する復号

50

アルゴリズムとしては、C h a s e 復号、O S D (Ordered Statistics Decoding) など  
が存在する。

【 0 0 1 1 】

このような最尤復号法を使った軟判定復号において最も高い訂正能力を得るには、復号語の候補となるリストに全符号語を含め、それぞれに対してユークリッド距離を計算して最尤符号語を決定する必要がある。しかしながら、全ての符号語に対してのユークリッド距離を計算するには膨大な計算量が発生するため、データ読出し時のレイテンシが非常に大きくなってしまふという課題がある。

【 0 0 1 2 】

そこで、一般的にメモリシステムに実装される軟判定復号では、「テストパターン」と呼ばれるエラービットの位置及び個数の組合せを仮定するパターンのリストを生成し、これらのテストパターンを用いつつ復号アルゴリズムに従って候補となる符号語を検出することで、ユークリッド距離の算出対象とする符号語を絞り込むことが行なわれる。

10

【 0 0 1 3 】

テストパターン生成方法の例としては、軟判定入力値の対数尤度比の絶対値の和の小さいものから順に候補とするテストパターンを選択する方法が存在する。また、O S D では、Box and Match Algorithmなどの方法が存在する。しかしながら、これらの方法では、リストアップするテストパターンの数を削減するほど誤り訂正能力が低下してしまうという課題が存在する。そこで以下の実施形態では、誤り訂正能力を低下させずにレイテンシを小さくすることが可能なメモリシステムについて、例を挙げて詳細に説明する。

20

【 0 0 1 4 】

( 第 1 の実施形態 )

図 1 は、第 1 の実施形態に係るメモリシステムの概略構成例を示すブロック図である。図 1 に示すように、メモリシステム 1 は、メモリコントローラ 1 0 と不揮発性メモリ 2 0 とを備える。メモリコントローラ 1 0 及び不揮発性メモリ 2 0 は、例えばそれらの組み合わせにより一つのメモリシステムを構成している。このようなメモリシステム 1 の例としては、S D (登録商標) カードのようなメモリカード、S S D (Solid State Drive) 等が挙げられる。

【 0 0 1 5 】

不揮発性メモリ 2 0 は、例えば N A N D 型フラッシュメモリなどの不揮発性メモリであってよい。また、一つのメモリコントローラ 1 0 に接続される不揮発性メモリ 2 0 は一つに限られず、複数の不揮発性メモリ 2 0 を接続することが可能である。

30

【 0 0 1 6 】

メモリコントローラ 1 0 は、ホストバスを介して接続されたホスト 3 0 から受信した命令に回答して、不揮発性メモリ 2 0 にアクセスする。ホスト 3 0 は、コンピュータの構成を備えている。コンピュータとは、例えば、パーソナルコンピュータ、サーバ装置、ポータブルな情報機器、デジタルスチルカメラ等であってよい。ホストバスの準拠する規格としては、任意の規格が採用可能である。

【 0 0 1 7 】

メモリコントローラ 1 0 は、C P U (Central Processing Unit) 1 1、R A M (Random Access Memory) 1 2、バッファメモリ 1 4、メモリインタフェース ( I / F ) 1 5 及びホストインタフェース ( I / F ) 1 7 を備え、これらが内部バス 1 8 を介して相互に接続されている。

40

【 0 0 1 8 】

ホスト I / F 1 7 は、ホスト 3 0 から受信した命令及びデータを、それぞれ C P U 1 1 及びバッファメモリ 1 4 に転送する。また、ホスト I / F 1 7 は、C P U 1 1 からの命令に回答して、バッファメモリ 1 4 内のデータをホスト 3 0 へ転送する。

【 0 0 1 9 】

C P U 1 1 は、メモリコントローラ 1 0 全体の動作を制御する。例えば、C P U 1 1 は、ホスト 3 0 から書込み命令を受信した際には、それに回答して、メモリ I / F 1 5 に対

50

して書込み命令を発行する。読出し及び消去の際も同様に、ホスト30からの命令に応答して、メモリI/F15に対して読出し命令又は消去命令を発行する。またCPU11は、ウェアレベリングやガベージコレクション等、不揮発性メモリ20を管理するための様々な処理を実行する。

#### 【0020】

RAM12は、例えばDRAM (Dynamic RAM) 等の半導体メモリであり、CPU11等の作業領域として使用される。RAM12には、不揮発性メモリ20を管理するためのファームウェア、及び、各種の管理テーブル等がロードされ得る。

#### 【0021】

バッファメモリ14は、書込みデータ又は読出しデータを一時的に保持するメモリ領域として機能する。バッファメモリ14は、DRAM、SRAM (Static RAM) 等によって構成され得る。

#### 【0022】

メモリI/F15は、チャンネルを介して1つ以上の不揮発性メモリ20と接続され、不揮発性メモリ20との通信を制御する。メモリI/F15は、CPU11から受信した命令に基づき、信号ALE、信号CLE、信号WEn、及び信号REnを不揮発性メモリ20へ出力する。例えば書込み時には、メモリI/F15は、CPU11が発行した書込み命令及びバッファメモリ14内の書込みデータを入出力信号I/Oとして不揮発性メモリ20へ転送する。また、読出し時には、メモリI/F15は、CPU11が発行した読出し命令を入出力信号I/Oとして不揮発性メモリ20へ転送する。そして、メモリI/F15は、不揮発性メモリ20から読み出されたデータを入出力信号I/Oとして受信し、これをバッファメモリ14へ転送する。

#### 【0023】

ここで、信号CEnは、不揮発性メモリ20をイネーブルにするための信号である。信号LEは、入力信号I/Oが命令であることを不揮発性メモリ20に通知する信号である。信号ALEは、入力信号I/Oがアドレスであることを不揮発性メモリ20に通知する信号である。信号WEnは、入力信号I/Oを不揮発性メモリ20に取り込ませるための信号である。信号REnは、不揮発性メモリ20から出力信号I/Oを読み出すための信号である。レディ・ビジー信号Rbnは、不揮発性メモリ20がレディ状態 (メモリコントローラ10からの命令を受信できる状態) であるか、それともビジー状態 (メモリコントローラ10からの命令を受信できない状態) であるかを示す信号である。入出力信号I/Oは、例えば8ビットの信号である。入出力信号I/Oは、不揮発性メモリ20とメモリコントローラ10との間で送受信されるデータの実体であり、命令、アドレス、書込みデータ、読出しデータ等である。

#### 【0024】

図2は、図1に示すメモリI/F15のより詳細な構成例を示すブロック図である。図2に示すように、メモリI/F15は、不揮発性メモリ20に書き込むデータを符号化するとともに、不揮発性メモリ20から読み出されたデータを復号するECC (Error Correction Code) 部100を備える。ECC部100は、軟判定値変換部140と、内部メモリ130と、リスト生成部120と、軟判定復号部110と、硬判定値変換部150とを備える。

#### 【0025】

軟判定値変換部140は、不揮発性メモリ20から読み出されたデータを軟判定値 (後述するLLR) に変換する。

#### 【0026】

内部メモリ130は、不揮発性メモリ20から読み出された受信語、軟判定値変換部140により軟判定値 (LLR) に変換された符号語、この符号語に対するテストパターンを軟判定復号部110が復号する過程で生成された軟判定値の復号語 (以下、中間復号語という) 等を格納する。

#### 【0027】

軟判定復号部110は、内部メモリ130内の軟判定値の符号語又は中間復号語に対し

10

20

30

40

50

て軟判定復号（ソフトビット復号：S B復号）を実行する。

【0028】

硬判定値変換部150は、内部メモリ130内に格納された軟判定値の符号語（例えば復号完了後の復号語）を（0，1）の2値で表された硬判定値の符号語に変換する。なお、硬判定値変換部150により硬判定値に変換された符号語は、不揮発性メモリ20内に再度書き込まれてもよいし（例えばガベージコレクション時）、バッファメモリ14等を介してホスト30へ出力されてもよい。

【0029】

リスト生成部120は、S B復号の入力となるテストパタンのリストを生成する。例えばリスト生成部120は、既出の復号語候補がある場合（例えば後述する最尤復号語選択部114において最尤復号語候補が保持されている場合）、その復号語候補のユークリッド距離を元に、当該ユークリッド距離よりも短いユークリッド距離となる可能性のあるテストパタンのリストを生成する。

10

【0030】

また、リスト生成部120は、テストパターン削減部121を備える。このテストパターン削減部121は、リスト生成部120により生成されたリストが既に存在する場合であって、既出の復号語候補がある場合には、その復号語候補のユークリッド距離を元に、当該ユークリッド距離よりも短いユークリッド距離となる可能性のないテストパターンを上記リストから削除する。

【0031】

軟判定復号部110は、テストパターン復号部111と、ユークリッド距離計算部113と、最尤復号語選択部114と、テストパターン実行判定部112とを備える。

20

【0032】

テストパターン復号部111は、規定のアルゴリズムに従ってリスト生成部120から入力されたリスト内のテストパターンを順次復号する。

【0033】

ユークリッド距離計算部113は、テストパターン復号部111で検出された符号語（中間復号語）と受信語とのユークリッド距離を計算する。

【0034】

最尤復号語選択部114は、ユークリッド距離計算部113で新たに計算されたユークリッド距離と既出の最尤復号語候補のユークリッド距離との大きさを比較し、ユークリッド距離が小さい方の中間復号語で最尤復号語候補を更新して保持する。

30

【0035】

テストパターン実行判定部112は、最尤復号語選択部114に保持されている最尤復号語候補のユークリッド距離に基づいて、リスト生成部120から入力されたリストにおけるテストパターンに対してS B復号を実行するか否かを判定し、S B復号を実行すると判定したテストパターンをテストパターン復号部111に入力する。

【0036】

このように本実施形態では、最尤復号語となり得ないテストパターンをリストに含めない機能（リスト生成部120及びテストパターン削減部121）を備えることで、復号時の計算量を削減し、レイテンシを小さくしている。また、本実施形態では、最尤復号語となり得ないテストパターンに対するS B復号をスキップする機能（テストパターン実行判定部112）を備えることでも、復号時の計算量を削減して、レイテンシを小さくしている。

40

【0037】

なお、ECC部100は、メモリI/F15とは独立してメモリコントローラ10内に設けられてもよい。その場合、例えばCPU11は、不揮発性メモリ20から読み出されたデータをメモリI/F15を経由してECC部100へ入力するような制御を実行する。

【0038】

つづいて、本実施形態における復号処理の流れについて、図面を用いて詳細に説明する

50

。

## 【 0 0 3 9 】

復号処理では、まず、不揮発性メモリ 2 0 からの対象データ（符号語）の読出しが実行される。不揮発性メモリ 2 0 に対するデータの読出しでは、硬判定読出し（ハードビット（H B）リードともいう）と、軟判定読出し（ソフトビット（S B）リードともいう）とが実行される。

## 【 0 0 4 0 】

硬判定読出し（ハードビット（H B）リード）では、読み出したビット値が（0，1）で切り替わる読出し電圧（以下、H Bリードレベルという）を各メモリセルに印加することで、2 値のデータである硬判定値のデータ（ハードビット（H B）データともいう）が読み出される。軟判定読出し（ソフトビット（S B）リード）では、H Bリードレベルから電圧値をそれぞれ  $\pm 1$ ， $\pm 2$ ，... シフトした複数のリードレベル（以下、S Bリードレベルという）を各メモリセルに印加することで、LOWER ページ、MIDDLE ページ、UPPER ページ等よりなる複数ページのデータ（ソフトビット（S B）データともいう）が読み出される。

10

## 【 0 0 4 1 】

S B リードにより各メモリセルから読み出された S B データは、各メモリセルの閾値電圧が想定される（0，1）の状態（閾値電圧分布）からどの程度ずれているかを示している。この S B データは、各メモリセルから読み出したビット値の信頼性（値の正しさ）を表現する対数尤度比（Log Likelihood Ratio：LLR）に変換することができる。

20

## 【 0 0 4 2 】

ここで、不揮発性メモリ 2 0 から受信語  $y$  が読み出されたとすると、その受信語  $y$  における  $i$  番目のビット  $b_i$  の LLR は、以下に式（1）で表すことができる。なお、式（1）において、 $P(b_i = 0 | y)$  は、 $i$  番目のビット  $b_i$  の値が“0”である確率を示し、 $P(b_i = 1 | y)$  は、 $i$  番目のビット  $b_i$  の値が“1”である確率を示している。

## 【 数 1 】

$$LLR(b_i) = \log\left(\frac{P(b_i = 0 | y)}{P(b_i = 1 | y)}\right) \quad (1)$$

## 【 0 0 4 3 】

式（1）から求まる LLR がプラスであれば、この LLR は  $i$  番目のビット  $b_i$  の値が“0”であることを示唆しており、マイナスであれば、 $i$  番目のビット  $b_i$  の値が“1”であることを示唆している。また、LLR の絶対値は、式（1）から求まる LLR の正負によって示唆される値の信頼度を示しており、絶対値が大きいほど信頼度が高く、小さいほど信頼度が低い。

30

## 【 0 0 4 4 】

軟判定値変換部 1 4 0 は、各メモリセルから読み出された複数ページの S B データを LLR に変換し、内部メモリ 1 3 0 に格納する。図 3 の上段に、不揮発性メモリ 2 0 から読み出した S B データの受信語を変換することで生成された LLR 列の一例を示す。理論的には、LLR は実数で表されるものであるが、本説明では、図 3 に示すように、LLR が整数に丸め込まれて、離散化された値を持つものとする。したがって、内部メモリ 1 3 0 内の符号語は、離散化された値を持つ LLR で表現された符号語となる。

40

## 【 0 0 4 5 】

リスト生成部 1 2 0 は、図 3 に示すように、S B 復号対象の符号語の各ビットを軟判定入力値（LLR）の絶対値の小さい順にソートする（S 1 1）。その際、リスト生成部 1 2 0 は、S B 復号対象の符号語における全ビットをソートしてもよいし、符号語の一部のビット（例えば LLR の絶対値の小さいものから順に所定数のビット）をソートしてもよい。

## 【 0 0 4 6 】

また、リスト生成部 1 2 0 は、テストパターン復号部 1 1 1 が C h a s e 復号アルゴリズム

50

ムに従って復号を実行するように設計されていた場合には、Chase復号用のテストパタンのリストを生成し、OSDアルゴリズムに従って復号を実行するように設計されていた場合には、OSD用のテストパタンのリストを生成する。本実施形態では、Chase復号アルゴリズムが採用されている場合を例に挙げ、OSDアルゴリズムを採用した場合については、後述する第2の実施形態において説明する。

【0047】

Chase復号用テストパタンのリストを生成する場合、リスト生成部120は、フリップ数を1としたテストパタンのリスト、フリップ数を2としたテストパタンのリスト、フリップ数を3としたテストパタンのリスト、...、フリップ数を $f$  ( $f$ は1以上の整数)としたテストパタンのリストを順次生成する。

10

【0048】

生成されるリストの大きさ(テストパタンの数)は、フリップするビットの数(以下、フリップ数という)の最大値から定められてもよいし、予めテストパタンの数の上限値として定められてもよい。ただし、リストの大きさ(テストパタンの数)は、許容されるレイテンシを満足し得る程度の大きさに設定される(LRB: LeastReliableBasis)。

【0049】

また、フリップ対象とするビットの範囲(フリップ対象範囲) $F$ は、図4に示すように、例えばソート後のLLR列に対し、LLRの絶対値の小さいものから順に所定の個数(以下、閾値数という)( $n$ )までの範囲となるように制限されてもよい。なお、図4では、閾値数( $n$ )を4とした場合が示されている。

20

【0050】

以上のような方法において、フリップ数の最大値を $k$ とし、閾値数( $n$ )を4とした場合、生成されるリストの大きさは、フリップ数を1としたテストパタンの組合せ数 $nC_1$ と、フリップ数を2としたテストパタンの組合せ数 $nC_2$ と、フリップ数を3としたテストパタンの組合せ数 $nC_3$ と、...、フリップ数を $k$ としたテストパタンの組合せ数 $nC_k$ との合計値( $nC_1 + nC_2 + nC_3 + \dots + nC_k$ )となる。

【0051】

なお、リスト内におけるテストパタンの優先順位は、例えばLLRの絶対値の小さいビットをフリップするテストパターンほど高い優先度となるような優先順位であってよい。言い換えれば、テストパターン復号部111における復号順序は、例えばLLRの絶対値の小さいビットをフリップするテストパターンほど高い優先度となるような優先順位であってよい。

30

【0052】

また、本実施形態では、後述するように、Chase復号アルゴリズムを使って訂正可能数 $t$ の符号語に対してフリップ数 $f$ のテストパタンを生成する際、既に1つの候補となる復号語が見つかっており(言い換えれば、既に最尤復号語候補が最尤復号語選択部114に保持されており)、且つ、そのユークリッド距離が $(f+t)a+t+1$ (なお、 $a$ は任意の自然数)より大きい場合には、リスト生成時のフリップ対象とするビットの条件に、LLRの絶対値が $a$ 以下のものという条件を追加する。このような条件を追加することで、フリップ対象範囲 $F$ (図4参照)内のビットからさらにフリップ対象とするビットが絞り込まれ、それにより、フリップするビットの組合せ数が減少し、生成されるテストパタンの数が減少する。その結果、復号時の計算量が削減し、レイテンシを小さくすることが可能となる。

40

【0053】

軟判定復号部110は、入力された軟判定入力値(LLR)を用いてSB復号を実行する。Chase復号アルゴリズムが採用されている本実施形態では、軟判定復号部110は、フリップ数を0とした復号(復号実行回数が1回)、フリップ数を1とした復号、フリップ数を2とした復号、フリップ数を3とした復号、...、フリップ数を $k$ とした復号を順次実行する。

【0054】

50

ユークリッド距離計算部 1 1 3 は、テストパターン復号部 1 1 1 で検出された符号語（中間復号語）と受信語とのユークリッド距離を計算する。このユークリッド距離は、例えばエラービットに対応する軟判定入力値（LLR）の絶対値の総和であってよい。

【0055】

例えば図 5 に示すように、テストパターン復号部 1 1 1 が中間復号語 # 1 の 1 番目のビットと 3 番目のビットがエラービットであると検出した場合、ユークリッド距離計算部 1 1 3 は、中間復号語 # 1 のユークリッド距離を 1 番目のビットの LLR の絶対値 ' 1 ' と、3 番目のビットの LLR の絶対値 ' 5 ' との総和である ' 6 ' を、中間復号語 # 1 のユークリッド距離として算出する。また、中間復号語 # 2 についても同様に、ユークリッド距離計算部 1 1 3 は、テストパターン復号部 1 1 1 がエラービットとして検出した 1 番目と 9 番目とのビットの LLR の絶対値の総和をユークリッド距離（' 4 '）として算出する。同様に、中間復号語 # 3 についても、ユークリッド距離計算部 1 1 3 は、テストパターン復号部 1 1 1 がエラービットとして検出した 1 番目と 5 番目と 7 番目と 9 番目とのビットの LLR の絶対値の総和をユークリッド距離（' 6 '）として算出する。

【0056】

最尤復号語選択部 1 1 4 は、最初に検出された中間復号語とユークリッド距離とを最尤復号語候補として保存する。また、最尤復号語選択部 1 1 4 は、2 回目以降に見つかった中間復号語については、その中間復号語のユークリッド距離を計算し、計算されたユークリッド距離と保存されている最尤復号語候補のユークリッド距離とを比較し、ユークリッド距離が小さい方の復号語を最尤復号語候補として保存し直す。したがって、新たに検出された中間復号語のユークリッド距離の方が小さければ、新たに検出された中間復号語で保存されている最尤復号語候補が更新される。

【0057】

つづいて、Chase 復号アルゴリズムを使った SB 復号の流れについて、具体例を挙げて説明する。

【0058】

Chase 復号アルゴリズムにおいて、訂正可能数  $t$  を 1 とし、フリップ数が 1 のテストパターンを用いて復号した場合、得られるエラー数はフリップ数 1 に訂正可能数  $t = 1$  を加えた合計 2 ビットとなる。このとき、得られるユークリッド距離が検出されたエラービットの LLR の絶対値の総和であることから、得られるユークリッド距離とエラービットの LLR の絶対値との組み合わせは、以下の表 1 のように列挙することができる。

【表 1】

ユークリッド距離	LLR絶対値の組合せ
1	(0,1)
2	(0,2),(1,1)
3	(0,3),(1,2)
4	(0,4),(1,3),(2,2)
5	(0,5),(1,4),(2,3)
6	(0,6),(1,5),(2,4),(3,3)

【0059】

表 1 から分かるように、訂正可能数  $t$  を 1、フリップ数を 1 とした場合に、ユークリッド距離が 3 以下の最尤復号語候補を得るには、フリップ対象の 1 ビットは LLR の絶対値が 1 であるビットに限定される。また、ユークリッド距離が 5 以下の最尤復号語候補を得るには、フリップ対象の 1 ビットは LLR の絶対値が 1 又は 2 であるビットに限定される。これを一般化すると、ユークリッド距離が  $2a + 1$  以下（ $a$  は 0 以上の整数）の最尤復号語候補を得るには、フリップ対象の 1 ビットは LLR の絶対値が  $a$  以下のビットに限定

されることが得られる。

【0060】

また、訂正可能数  $t$  を 1 とし、フリップ数を 2 とした場合には、Chase 復号の結果として得られるエラービットの数は 3 ビットとなる。したがって、得られるユークリッド距離とエラービットの LLR の絶対値との組み合わせは、以下の表 2 のように列挙することができる。

【表 2】

ユークリッド距離	LLR絶対値の組合せ
1	(0,0,1)
2	(0,0,2),(0,1,1)
3	(0,0,3),(0,1,2),(1,1,1)
4	(0,0,4),(0,1,3),(0,2,2),(1,1,2)
5	(0,0,5),(0,1,4),(0,2,3),(1,1,3),(1,2,2)
6	(0,0,6),(0,1,5),(0,2,4),(0,3,3),(1,1,4),(1,2,3),(2,2,2)

10

【0061】

表 2 から分かるように、ユークリッド距離が  $3a + 1$  以下の最尤復号語候補を得るには、フリップ対象とする 2 ビットは、それぞれ LLR の絶対値が  $a$  以下のビットに限定される。

20

【0062】

以上のことから、訂正可能数  $t$  を 1 とした符号において、フリップ数  $f$  でユークリッド距離が  $(f + 1)a + 1$  以下の最尤復号語候補を検出するためには、フリップ対象の  $f$  ビットそれぞれを、LLR の絶対値が  $a$  以下のビットに限定すればよいことが分かる。

【0063】

次に、訂正可能数  $t$  を 2 とした場合について説明する。訂正可能数  $t$  を 2 とし、フリップ数を 1 とした場合には、Chase 復号の結果として得られるエラービットの数は 3 ビットとなる。したがって、得られるユークリッド距離とエラービットの LLR の絶対値との組み合わせは、以下の表 3 のように列挙することができる。

30

【表 3】

ユークリッド距離	LLR絶対値の組合せ
1	(0,0,1)
2	(0,0,2),(0,1,1)
3	(0,0,3),(0,1,2),(1,1,1)
4	(0,0,4),(0,1,3),(0,2,2),(1,1,2)
5	(0,0,5),(0,1,4),(0,2,3),(1,1,3),(1,2,2)
6	(0,0,6),(0,1,5),(0,2,4),(0,3,3),(1,1,4),(1,2,3),(2,2,2)

40

【0064】

表 3 から分かるように、訂正可能数  $t$  を 2 とした符号において、フリップ数  $f$  でユークリッド距離が  $3a + 2$  以下の最尤復号語候補を検出するためには、フリップ対象の  $f$  ビットそれぞれを、LLR の絶対値が  $a$  以下のビットに限定すればよいことが分かる。

【0065】

また、訂正可能数  $t$  を 2 とし、フリップ数を 2 とした場合には、検出されるエラービットの数が 4 ビットとなるため、得られるユークリッド距離とエラービットの LLR の絶対

50

値との組み合わせは、以下の表4のように列挙することができる。

【表4】

ユークリッド距離	LLR絶対値の組合せ
1	(0,0,0,1)
2	(0,0,0,2), (0,0,1,1)
3	(0,0,0,3), (0,0,1,2), (0,1,1,1)
4	(0,0,0,4), (0,0,1,3), (0,0,2,2), (0,1,1,2), (1,1,1,1)
5	(0,0,0,5), (0,0,1,4), (0,0,2,3), (0,1,1,3), (0,1,2,2), (1,1,1,2)
6	(0,0,0,6), (0,0,1,5), ..., (0,1,1,4), (0,1,2,3), (0,2,2,2), (1,1,2,2)
7	(0,0,0,7), (0,0,1,6), ..., (1,1,1,4), (1,1,2,3), (1,2,2,2), (1,2,2,2)
8	(0,0,0,8), (0,0,1,7), ..., (1,1,2,4), (1,1,3,3), (1,2,2,3), (2,2,2,2)

10

【0066】

表4から分かるように、訂正可能数  $t$  を2とし、フリップ数を2とした場合において、ユークリッド距離が  $4a + 2$  以下の最尤復号語候補を検出するためには、フリップ対象の2ビットそれぞれを、LLRの絶対値が  $a$  以下のビットに限定すればよい。

20

【0067】

以上のように、Chase復号アルゴリズムを使って訂正可能数  $t$  の符号語をフリップ数  $f$  のビットフリップしたテストパターンを用いて復号する場合において、ユークリッド距離が  $(f + t)a + t$  以下となる最尤復号語候補を検出するためには、フリップ対象の  $f$  ビットそれぞれを、LLRの絶対値が  $a$  以下のビットに限定すればよいことが分かる。

【0068】

たとえば訂正可能数が2の符号語に関し、フリップ数0で1つの最尤復号語候補が検出され、そのユークリッド距離が9であった場合、最尤復号語候補を更新するためには、新たに検出された中間復号語のユークリッド距離は8以下である必要がある。そこで、リスト生成部120は、フリップ数を1としたテストパターンのリストを生成する際に、フリップ対象のビットをLLRの絶対値が1又は2のビットに制限する。または、既にフリップ数を1としたテストパターンのリストが生成されている場合には、リスト生成部120内のテストパターン削減部121が、フリップ対象のビットをLLRの絶対値が1又は2のビットでないテストパターンをリストから削除する。これにより、最尤復号語となり得ないテストパターンをリストから除外することが可能となるため、復号時の計算量が削減されて、レイテンシが小さくなるという効果が得られる。

30

【0069】

また、フリップ数を1としたテストパターンのリストを用いてChase復号を実行した結果又は実行中にユークリッド距離が例えば6の最尤復号語候補が検出された場合、リストの残りのテストパターンのうちLLRの絶対値が2のビットがフリップされたテストパターンを用いてChase復号を実行したとしても、最尤復号語選択部114に保持された最尤復号語候補が更新されることはない。そこで、このような場合には、軟判定復号部110内のテストパターン実行判定部112は、LLRの絶対値が2のビットがフリップされたテストパターンに対し、テストパターン復号部111がChase復号が実行を実行しないように制限する。例えばテストパターン実行判定部112は、LLRの絶対値が2のビットがフリップされたテストパターンのテストパターン復号部111への入力をスキップする。これにより、最尤復号語となり得ないテストパターンに対するSB復号がスキップされるため、復号時の計算量が削減されて、レイテンシを小さくするという効果が得られる。

40

【0070】

次に、本実施形態に係る復号動作について、図面を参照して詳細に説明する。図6は、

50

本実施形態に係る復号動作の一例を示すフローチャートである。図6に示すように、本実施形態では、まず、リスト生成部120がフリップ数を $f$ としたテストパターン（以下、 $f$ ビットフリップのテストパターンという）のリストを生成する（ステップS101）。なお、 $f$ の初期値は1であるとする。また、既に $f$ ビットフリップのテストパターンのリストが存在する場合には、ステップS101がスキップされてもよい。

#### 【0071】

次に、リスト生成部120のテストパターン削減部121が、リストアップされた $f$ ビットフリップのテストパターンのうち、最尤復号語候補となる可能性のないテストパターンを削除することで、復号対象のテストパターンを絞り込む（ステップS102）。具体的には、テストパターン削減部121は、まず、リストアップされたテストパターンそれぞれについて、エラービットと仮定されているビットのLLRの総和からユークリッド距離を算出する。つづいて、テストパターン削減部121は、最尤復号語選択部114に格納されている最尤復号語候補のユークリッド距離を取得する。その後、テストパターン削減部121は、リストアップされたテストパターンのうち、ユークリッド距離が最尤復号語候補のユークリッド距離以上のものをリストから削除する。これにより、復号対象とするテストパターンが絞り込まれるため、復号時の計算量が削減されて、レイテンシを小さくすることができる。なお、ステップS101においてリスト生成部120が最尤復号語候補となる可能性のないテストパターンを生成しないように動作する場合には、ステップS102を省略することも可能である。また、絞り込み後のリストは、軟判定復号部110のテストパターン実行判定部112に入力される。

10

20

#### 【0072】

次に、テストパターン実行判定部112が、入力されたリストの中から未選択のテストパターンを1つ選択する（ステップS103）。なお、選択時の優先順位としては、上述したように、例えばLLRの絶対値の小さいビットをフリップするテストパターンほど高い優先度となるような優先順位であってよい。このような選択順序とすることで、エラーベクタが検出される可能性の高いテストパターンに対する復号を優先的に実行することが可能になるとともに、復号処理の流れにおける比較的早期に、最尤復号語選択部114に格納される最尤復号語候補のユークリッド距離を短くすることが可能となる。その結果、リスト生成部120がリストに含めないこととするテストパターンの数、テストパターン削減部121がリストから削除するテストパターンの数、及び、テストパターン実行判定部112がスキップ対象とするテストパターンの数が増加するため、復号時の計算量がより削減されて、レイテンシをより小さくすることが可能となる。

30

#### 【0073】

次に、テストパターン実行判定部112は、選択したテストパターンを用いたSB復号を実行するか否かを判定する（ステップS104）。具体的には、テストパターン実行判定部112は、選択したテストパターンのユークリッド距離を特定する。このユークリッド距離は、リスト生成部120からリストとともに入力されてもよいし、テストパターン実行判定部112が選択したテストパターンにおいてエラービットと仮定されているビットのLLRの総和から算出してもよい。つづいて、テストパターン実行判定部112は、最尤復号語選択部114に格納されている最尤復号語候補のユークリッド距離を取得する。その後、テストパターン実行判定部112は、選択したテストパターンのユークリッド距離が最尤復号語候補のユークリッド距離未満であれば、当該テストパターンに対するSB復号を実行すると判定し（ステップS104のYES）、ステップS105へ進む。一方、選択したテストパターンのユークリッド距離が最尤復号語候補のユークリッド距離以上であれば、テストパターン実行判定部112は、当該テストパターンに対するSB復号を実行しないと判定し（ステップS104のNO）、ステップS112へ進む。これにより、最尤復号語候補が検出され得ないテストパターンに対するSB復号がスキップされるため、復号時の計算量が削減されて、レイテンシを小さくすることができる。

40

#### 【0074】

ステップS105では、テストパターン実行判定部112からテストパターン復号部111

50

に S B 復号を実行すると判定されたテストパターンが入力され、テストパターン復号部 1 1 1 が入力されたテストパターンを用いた S B 復号を実行する。つづいて、テストパターン復号部 1 1 1 は、エラーベクタが検出されたか否かを判定する (ステップ S 1 0 6)。エラーベクタが検出されなかった場合 (ステップ S 1 0 6 の N O)、ステップ S 1 1 0 へ進む。一方、エラーベクタが検出された場合 (ステップ S 1 0 6 の Y E S)、ユークリッド距離計算部 1 1 3 が、テストパターンによりフリップされたビットの L L R と、検出されたエラーベクタが示すエラービットの L L R とからユークリッド距離を計算する (ステップ S 1 0 7)。つづいて、最尤復号語選択部 1 1 4 が、ユークリッド距離計算部 1 1 3 により新たに計算されたユークリッド距離と、保存している最尤復号語候補のユークリッド距離とを比較し (ステップ S 1 0 8)、新たに計算されたユークリッド距離が最尤復号語候補のユークリッド距離未満であれば (ステップ S 1 0 8 の Y E S)、保存している最尤復号語候補及びそのユークリッド距離を、テストパターン復号部 1 1 1 による S B 復号により新たに検出された中間復号語及びその中間復号語についてユークリッド距離計算部 1 1 3 が新たに計算したユークリッド距離で更新し (ステップ S 1 0 9)、ステップ S 1 1 0 へ進む。一方、新たに計算されたユークリッド距離の方が最尤復号語候補のユークリッド距離以上である場合 (ステップ S 1 0 8 の N O)、最尤復号語選択部 1 1 4 が保持する最尤復号語候補及びそのユークリッド距離を更新せずに、ステップ S 1 1 0 へ進む。

10

20

30

40

50

#### 【 0 0 7 5 】

ステップ S 1 1 0 では、ステップ S 1 0 2 による絞り込み後のリストにおけるテストパターンの全てに対して S B 復号が完了したか否かが判定される。未だ S B 復号が実行されていないテストパターンが存在する場合 (ステップ S 1 1 0 の N O) は、ステップ S 1 0 3 へリターンし、新たに選択されたテストパターンに対して以降の動作が実行される。一方、リストにある全てのテストパターンに対して S B 復号が完了している場合 (ステップ S 1 1 0 の Y E S)、フリップ数  $f$  を 1 インクリメント (ステップ S 1 1 1) した後、ステップ S 1 1 2 へ進む。

#### 【 0 0 7 6 】

ステップ S 1 1 2 では、フリップ数  $f$  がフリップ数の最大値  $k$  よりも大きいかが否か、すなわち、フリップ数の最大値  $k$  までの全てのテストパターンについての処理が完了したか否かが判定される。フリップ数  $f$  が最大値  $k$  以下である場合 (ステップ S 1 1 2 の N O)、ステップ S 1 0 1 へリターンして、新たなフリップ数  $f$  について  $f$  ビットフリップのテストパターンのリストが生成された後、以降の動作が実行される。一方、フリップ数  $f$  が最大値  $k$  より大きい場合 (ステップ S 1 1 2 の Y E S)、本動作が終了される。

#### 【 0 0 7 7 】

以上のように、本実施形態によれば、既出の最尤復号語候補よりもユークリッド距離が小さくなる可能性がないテストパターンをリストから排除して復号対象に含めないことで、最尤復号法での計算量を削減してレイテンシを小さくすることが可能となる。また、復号の途中で新たな最尤復号語候補が検出された場合には、この新たな最尤復号語候補よりもユークリッド距離が小さくなる可能性がないテストパターンに対する S B 復号をスキップすることで、最尤復号法での計算量を削減してレイテンシを小さくすることが可能となる。

#### 【 0 0 7 8 】

( 第 2 の実施形態 )

上述した第 1 の実施形態では、C h a s e 復号アルゴリズムが採用されている場合を例に挙げた。これに対し、第 2 の実施形態では、O S D アルゴリズムが採用された場合について、例を挙げて詳細に説明する。

#### 【 0 0 7 9 】

本実施形態において、メモリ I / F を含むメモリシステムの概略構成は、上述した第 1 の実施形態に係るメモリシステム 1 ( 図 1 及び図 2 参照 ) と同様であってよいため、ここでは詳細な説明を省略する。また、復号動作についても同様に、上述した第 1 の実施形態において図 6 等を用いて説明した概略動作例と同様であってよいため、ここでは詳細な説明を省略する。

## 【0080】

つづいて、OSDアルゴリズムを使ったSB復号の流れについて、具体例を挙げて説明する。本説明では、保護対象の情報語が訂正可能数が $t$ の符号語で保護されているものとする。また、入力された符号語に対して、OSD復号の前に実行された限界距離復号の結果、ユークリッド距離が $b$ の中間復号語が最尤復号語候補として検出されているものとする。この場合、OSD復号の結果として得られるエラーベクタには、少なくとも $(t+1)$ 個のエラービットが含まれることになる。

## 【0081】

OSD用テストパターンを生成する場合においては、符号語に対してフリップ対象とするビットの範囲(フリップ対象範囲)Fを設定する場合、リスト生成部120は、図7に示すように、例えば図3を用いて説明したソート(S11)後のLLR列に対し、LLRの絶対値の小さいものから順に、シンドロームサイズ(パリティサイズ) $p$ に所定の閾値数( $n$ )を加えた数( $p+n$ )をエラーが存在する可能性のある範囲(以下、エラー仮定範囲という)とし、所定の閾値数( $n$ )に相当する範囲をフリップ対象範囲Fに設定する。その際、閾値数( $n$ )は、フリップ数ごとに異なる値としてもよい。なお、図7では、シンドロームサイズを4とし、閾値数( $n$ )を4とした場合が示されている。

## 【0082】

OSDアルゴリズムにおけるフェーズ(以下、OSDフェーズという)0では、軟判定復号部110は、ソートされたビットのうち、LLRの絶対値の最も小さいものから順にシンドロームサイズ分のビットを使用してOSDフェーズ0の復号を実行する。ただし、LLRの絶対値の最も小さいビットから数えて $(t+1)$ 番目のビットのLLRが最尤復号語候補のユークリッド距離 $b$ 以上である場合、OSDフェーズ0の復号を実行したとしても最尤復号語選択部114内の最尤復号語候補が更新されることはない。そのため、この場合には、リスト生成部120はリストを生成せずに、「ボタン無し」であることを軟判定復号部110に通知する。この場合、ECC部100は、以降のOSDを実行せず、既に得られている最尤復号語候補を最終的な復号語である軟判定出力値として出力する。

## 【0083】

OSDフェーズ0以降、OSDフェーズ1からOSDフェーズ $k$ ( $k$ はフリップ数の最大値に相当)までが順次実行される。具体的には、OSDフェーズ $j$ ( $j$ は1以上 $k$ 以下)では、リスト生成部120が、ソートされたビットのうちOSDフェーズ0で使用したシンドロームサイズ $p$ のビットを除いたフリップ対象範囲F内のビットを $j$ 個フリップするテストパタンのリストを生成し、軟判定復号部110が生成されたテストパタンのリストに従いOSDフェーズ $j$ の復号を実行する。

## 【0084】

その際、OSDフェーズ( $j-1$ )( $j$ は1以上 $k$ 以下)までの過程でユークリッド距離が $c$ の符号語が検出されている場合、OSDフェーズ $j$ 以降では、エラーと仮定する $j$ 個のビットのLLRの絶対値の和が $c$ より大きくなるテストパターンを復号したとしても、当該復号により検出された中間復号語が最尤復号語候補となることはない。そこで、このような場合、OSDフェーズ $j$ 以降では、リスト生成部120は、フリップ対象とした $j$ 個のビットのLLRの絶対値の和が $c$ より大きくなるテストパターンを含まないリストを生成する。若しくは、既にリストが存在する場合には、テストパターン削減部121がリストから該当のテストパターンを削除するか、テストパターン実行判定部112が当該テストパターンをテストパターン復号部111による復号対象から除外する。

## 【0085】

さらに、 $1 < j < t$ ( $t$ は符号語の訂正可能数)の場合、LRBに基づいて $(t+1-j)$ 個のLLRの絶対値の和を $m$ とおくとき、エラーと仮定する $j$ 個のビットのLLRの和が $c-m$ より大きくなるテストパターンを復号したとしても、当該復号により検出された中間復号語が最尤復号語候補となることはない。そこで、このような場合、OSDフェーズ $j$ 以降では、リスト生成部120は、フリップ対象とする $j$ 個のビットのLLRの絶対値の和が $c-m$ 以上となるテストパターンを含まないリストを生成する。若しくは、既にリ

10

20

30

40

50

ストが存在する場合には、テストパターン削減部 1 2 1 がリストから該当のテストパターンを削除するか、テストパターン実行判定部 1 1 2 が当該テストパターンをテストパターン復号部 1 1 1 による復号対象から除外する。

【 0 0 8 6 】

以上のような構成及び動作とすることで、本実施形態によれば、OSDアルゴリズムを採用した場合も、第 1 の実施形態と同様に、既出の最尤復号語候補よりもユークリッド距離が小さくなる可能性がないテストパターンをリストから排除して復号対象に含めないことで、最尤復号法での計算量を削減してレイテンシを小さくすることが可能となる。また、復号の途中で新たな最尤復号語候補が検出された場合には、この新たな最尤復号語候補よりもユークリッド距離が小さくなる可能性がないテストパターンに対するSB復号をスキップすることで、最尤復号法での計算量を削減してレイテンシを小さくすることが可能となる。

10

【 0 0 8 7 】

その他の構成、動作及び効果は、上述した実施形態と同様であるため、ここでは詳細な説明を省略する。

【 0 0 8 8 】

本発明のいくつかの実施形態を説明したが、これらの実施形態は、例として提示したものであり、発明の範囲を限定することは意図していない。これら新規な実施形態は、その他の様々な形態で実施されることが可能であり、発明の要旨を逸脱しない範囲で、種々の省略、置き換え、変更を行うことができる。これら実施形態やその変形は、発明の範囲や要旨に含まれるとともに、特許請求の範囲に記載された発明とその均等の範囲に含まれる。

20

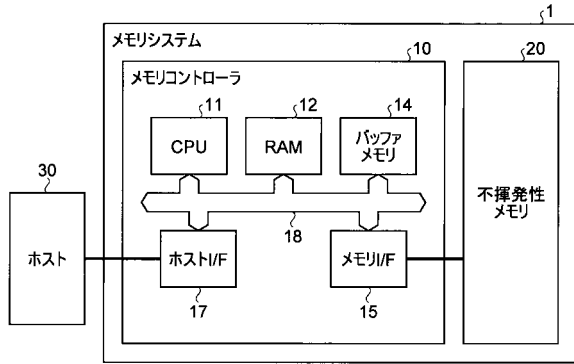
【 符号の説明 】

【 0 0 8 9 】

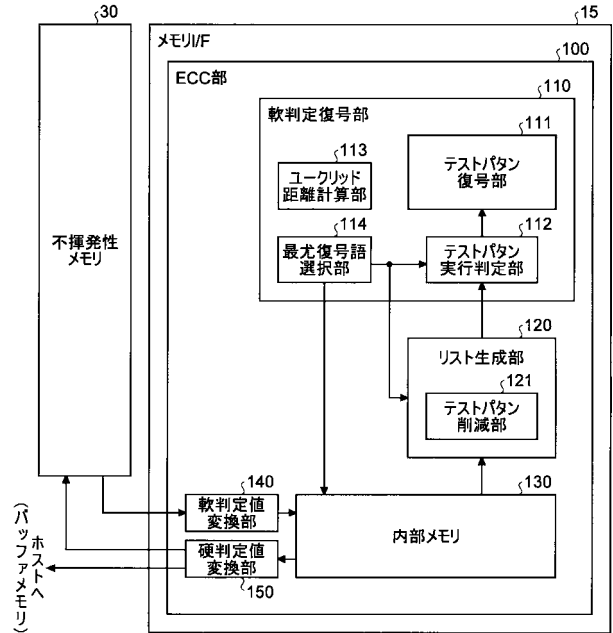
1 ... メモリシステム、1 0 ... メモリコントローラ、1 1 ... CPU、1 2 ... RAM、1 4 ... バッファメモリ、1 5 ... メモリ I / F、1 7 ... ホスト I / F、1 8 ... 内部バス、2 0 ... 不揮発性メモリ、3 0 ... ホスト、1 0 0 ... ECC 部、1 1 0 ... 軟判定復号部、1 1 1 ... テストパターン復号部、1 1 2 ... テストパターン実行判定部、1 1 3 ... ユークリッド距離計算部、1 1 4 ... 最尤復号語選択部、1 2 0 ... リスト生成部、1 2 1 ... テストパターン削減部、1 3 0 ... 内部メモリ、1 4 0 ... 軟判定値変換部、1 5 0 ... 硬判定値変換部。

30

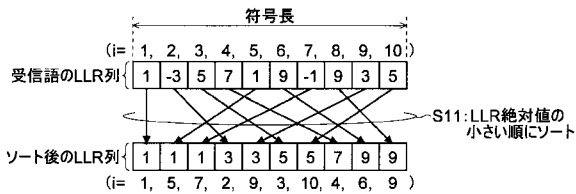
【 図 1 】



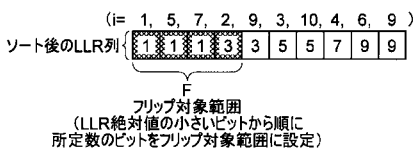
【 図 2 】



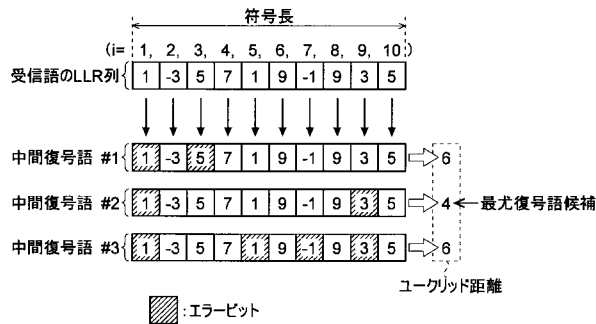
【 図 3 】



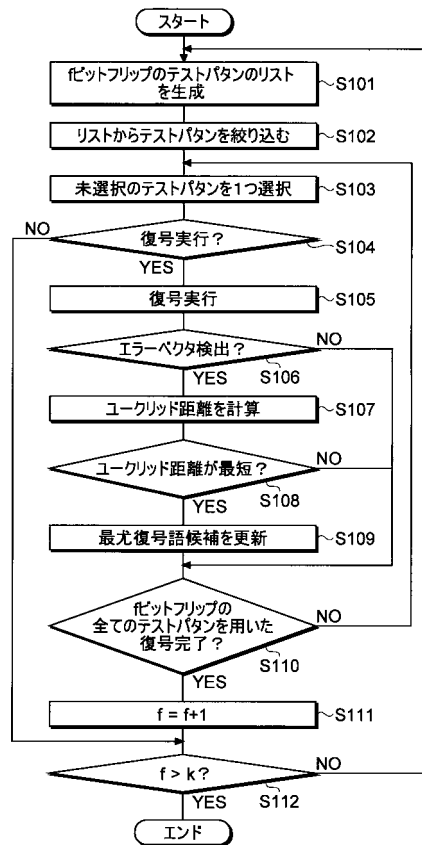
【 図 4 】



【 図 5 】



【 図 6 】



【 図 7 】

