

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5768022号
(P5768022)

(45) 発行日 平成27年8月26日 (2015. 8. 26)

(24) 登録日 平成27年6月26日 (2015. 6. 26)

(51) Int. Cl. F I
HO 3M 13/29 (2006. 01) HO 3M 13/29
GO 6F 12/16 (2006. 01) GO 6F 12/16 3 2 0 G

請求項の数 19 (全 36 頁)

(21) 出願番号	特願2012-187140 (P2012-187140)	(73) 特許権者	000003078 株式会社東芝
(22) 出願日	平成24年8月28日 (2012. 8. 28)		東京都港区芝浦一丁目1番1号
(65) 公開番号	特開2013-225830 (P2013-225830A)	(74) 代理人	100089118 弁理士 酒井 宏明
(43) 公開日	平成25年10月31日 (2013. 10. 31)		
審査請求日	平成26年8月29日 (2014. 8. 29)	(72) 発明者	鳥井 修 東京都港区芝浦一丁目1番1号 株式会社東芝内
(31) 優先権主張番号	特願2012-61692 (P2012-61692)	(72) 発明者	菅野 伸一 東京都港区芝浦一丁目1番1号 株式会社東芝内
(32) 優先日	平成24年3月19日 (2012. 3. 19)	(72) 発明者	山城 遼 東京都港区芝浦一丁目1番1号 株式会社東芝内
(33) 優先権主張国	日本国 (JP)		

最終頁に続く

(54) 【発明の名称】 メモリコントローラ、記憶装置、誤り訂正装置および誤り訂正方法

(57) 【特許請求の範囲】

【請求項1】

不揮発性メモリを制御するメモリコントローラであって、
 ユーザデータに誤り訂正符号化処理を行うことにより第1乃至第nパリティを生成し、
 前記第1乃至第nパリティ各々に誤り訂正符号化処理を行うことにより第1乃至第n外部
 パリティを生成する符号化部と、

前記ユーザデータ、前記第1乃至第nパリティ、及び前記第1乃至第n外部パリティの
 前記不揮発性メモリへの書込み、及び前記不揮発性メモリからの読み出しを制御するメモ
 リインタフェース部と、

前記不揮発性メモリから読み出される前記ユーザデータ、前記第1乃至第nパリティ、
 及び前記第1乃至第n外部パリティを用いて、誤り訂正復号処理を行う復号部と、
 を備え、

前記第iパリティ(iは1以上n以下の整数)を生成するために用いられる生成多項式
 は、第1乃至第(i-1)パリティを生成するために用いられる生成多項式に基づいて選
 択されることを特徴とするメモリコントローラ。

【請求項2】

前記復号部は、前記第1乃至第k外部パリティ(kは1以上n以下の整数)を用いて前
 記第1乃至第kパリティの誤り訂正復号処理を行った後に、前記第1乃至第kパリティを
 用いて前記ユーザデータの誤り訂正復号処理を行うことを特徴とする請求項1に記載のメ
 モリコントローラ。

【請求項 3】

不揮発性メモリを制御するメモリコントローラであって、

ユーザデータに誤り訂正符号化処理を行うことにより第 1 パリティを生成し、前記ユーザデータと前記第 1 パリティに誤り訂正符号化処理を行うことにより第 2 パリティを生成し、前記ユーザデータと前記第 1 パリティに誤り訂正符号化処理を行うことにより第 n パリティを生成し、さらに、前記第 2 乃至第 n パリティ各々に誤り訂正符号化処理を行うことにより第 2 乃至第 n 外部パリティを生成する符号化部と、

前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 2 乃至第 n 外部パリティの前記不揮発性メモリへの書込み、及び前記不揮発性メモリからの読み出しを制御するメモリインタフェース部と、

10

前記不揮発性メモリから読み出される前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 2 乃至第 n 外部パリティを用いて、誤り訂正復号処理を行う復号部と、を備え、

前記第 i パリティ (i は 1 以上 n 以下の整数) を生成するために用いられる生成多項式は、第 1 乃至第 ($i - 1$) パリティを生成するために用いられる生成多項式に基づいて選択されることを特徴とするメモリコントローラ。

【請求項 4】

前記復号部は、前記第 1 パリティの誤り訂正能力では前記ユーザデータの誤り訂正復号できない場合に、前記第 2 乃至第 k 外部パリティ (k は 2 以上 n 以下の整数) を用いて前記第 2 乃至第 k パリティの誤り訂正復号処理を行った後に、前記第 2 乃至第 k パリティを用いて前記ユーザデータ及び前記第 1 パリティの誤り訂正を行うことを特徴とする請求項 3 に記載のメモリコントローラ。

20

【請求項 5】

前記第 i パリティ (i は 1 以上 n 以下の整数) を生成するために用いられる生成多項式 $G_i(x)$ は、 r_i 、 t_i を整数とした場合に、

r_1, r_2, \dots, r_n は、以下の関係を満たし、
 $r_2 = r_1 + 2t_1 - 1$ $r_2 + 2t_2 - 1$
 $r_3 = r_2 + 2t_2 - 1$ $r_3 + 2t_3 - 1$
 ...

$r_n = r_{n-1} + 2t_{n-1} - 1$ $r_n + 2t_n - 1$

30

$G_1(x) \times G_2(x) \times \dots \times G_i(x) = 0$ が、 $r_i, r_{i+1}, r_{i+2}, \dots, r_{2t_i+r_i-2}, r_{2t_i+r_i-1}$ の $2 \times t_i$ 個の根を有するよう選択されることを特徴とする請求項 1 乃至 4 いずれか 1 項に記載のメモリコントローラ。

【請求項 6】

前記第 1 乃至第 n パリティの生成する誤り訂正符号処理には、同一の誤り訂正符号が用いられることを特徴とする請求項 1 乃至 5 いずれか 1 項に記載のメモリコントローラ。

【請求項 7】

前記誤り訂正符号は、BCH符号であることを特徴とする請求項 6 に記載のメモリコントローラ。

【請求項 8】

40

前記パリティ及び前記外部パリティの少なくとも一部は、前記ユーザデータとは異なる、パリティを格納する専用のパリティ専用ページに格納されることを特徴とする請求項 1 乃至 7 いずれか 1 項に記載のメモリコントローラ。

【請求項 9】

前記パリティ及び前記外部パリティの少なくとも一部は、パリティの保護対象であるユーザデータまたはパリティとは異なるユーザデータページに格納されることを特徴とする請求項 1 乃至 7 いずれか 1 項に記載のメモリコントローラ。

【請求項 10】

不揮発性メモリを制御するメモリコントローラであって、

第 1 ユーザデータに誤り訂正符号化処理を行うことにより、第 1 パリティを生成し、前

50

記第 1 ユーザデータと前記第 1 パリティに誤り訂正符号処理を行うことにより第 2 パリティを生成し、第 2 ユーザデータと前記第 2 パリティを含む複数のパリティに誤り訂正符号処理を行うことにより第 3 パリティを生成する符号化部と、

前記第 1 ユーザデータと前記第 1 パリティの第 1 ページへの書込み、及び前記第 2 ユーザデータ及び前記第 2 のパリティを含む複数のパリティ、第 3 パリティの第 2 ページへの書込みを制御するメモインタフェース部と、

前記不揮発性メモリから読み出される前記第 1、第 2 ユーザデータ及び前記第 1 乃至第 3 パリティに基づき、誤り訂正復号処理を行う復号部と、

を備え、

前記第 2 パリティを生成するために用いられる生成多項式は、第 1 パリティを生成するために用いられる生成多項式に基づいて選択されることを特徴とするメモリコントローラ。

10

【請求項 1 1】

前記第 1 パリティのサイズは、前記第 2 パリティのサイズと前記第 3 のパリティのサイズの和以上であることを特徴とする請求項 1 0 に記載のメモリコントローラ。

【請求項 1 2】

前記第 i パリティ (i は 1 以上 n 以下の整数) を生成するために用いられる生成多項式 $G_i(x)$ は、 r_1 、 t_1 、 r_2 、 t_2 、 r_3 、 t_3 を整数とした場合に、

r_1 、 t_1 、 r_1 、 r_2 、 r_3 、 t_3 は、以下の関係を満たし、

$$r_2 = r_1 + 2t_1 - 1 \quad r_2 + 2t_2 - 1$$

$$r_3 = r_2 + 2t_2 - 1 \quad r_3 + 2t_3 - 1$$

$G_1(x) \times G_2(x) \times \dots \times G_i(x) = 0$ が、 r_i 、 r_{i+1} 、 r_{i+2} 、 \dots 、 $2t_i + r_{i-2}$ 、 $2t_i + r_{i-1}$ の $2 \times t_i$ 個の根を有するよう選択されることを特徴とする請求項 1 0 又は 1 1 に記載のメモリコントローラ。

20

【請求項 1 3】

不揮発性メモリと、

ユーザデータに誤り訂正符号化処理を行うことにより第 1 乃至第 n パリティを生成し、前記第 1 乃至第 n パリティ各々に誤り訂正符号化処理を行うことにより第 1 乃至第 n 外部パリティを生成する符号化部と、

前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 1 乃至第 n 外部パリティの前記不揮発メモリへの書込み、及び前記不揮発性メモリからの読み出しを制御するメモインタフェース部と、

30

前記不揮発性メモリから読み出される前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 1 乃至第 n 外部パリティを用いて、誤り訂正復号処理を行う復号部と、

を備え、

前記第 i パリティ (i は 1 以上 n 以下の整数) を生成するために用いられる生成多項式は、第 1 乃至第 ($i - 1$) パリティを生成するために用いられる生成多項式に基づいて選択されることを特徴とする記憶装置。

【請求項 1 4】

ユーザデータに誤り訂正符号化処理を行うことにより第 1 乃至第 n パリティを生成し、前記第 1 乃至第 n パリティ各々に誤り訂正符号化処理を行うことにより第 1 乃至第 n 外部パリティを生成する符号化部と、

40

受信した前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 1 乃至第 n 外部パリティを用いて、誤り訂正復号処理を行う復号部と、

を備え、

前記第 i パリティ (i は 1 以上 n 以下の整数) を生成するために用いられる生成多項式は、第 1 乃至第 ($i - 1$) パリティを生成するために用いられる生成多項式に基づいて選択されることを特徴とする誤り訂正装置。

【請求項 1 5】

不揮発性メモリを制御するメモリコントローラにおける誤り訂正方法であって、

50

ユーザデータに誤り訂正符号化処理を行うことにより第 1 乃至第 n パリティを生成するステップと、

前記第 1 乃至第 n パリティ各々に誤り訂正符号化処理を行うことにより第 1 乃至第 n 外部パリティを生成するステップと、

前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 1 乃至第 n 外部パリティの前記不揮発メモリへの書込むステップと、

前記不揮発性メモリから前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 1 乃至第 n 外部パリティの読み出しを行うステップと、

前記不揮発性メモリから読み出される前記ユーザデータ、前記第 1 乃至第 n パリティ、及び前記第 1 乃至第 n 外部パリティを用いて、誤り訂正復号処理を行うステップと、

を備え、

前記第 i パリティ (i は 1 以上 n 以下の整数) を生成するために用いられる生成多項式は、第 1 乃至第 ($i - 1$) パリティを生成するために用いられる生成多項式に基づいて選択されることを特徴とする誤り訂正方法。

【請求項 16】

前記復号部は、

第 1 乃至第 n パリティの全てを用いた場合の訂正能力を t_n ビットとすると、 t_n ビット以上の演算に対応可能なシンドローム計算部、を備え、

前記シンドローム計算部は、第 1 乃至第 n パリティを生成するために用いられる生成多項式の各根に対応するシンドロームを計算する t_n 個以上のシンドローム計算回路、を有し、

第 i パリティを生成するために用いられる生成多項式を G_i とするとき、 G_1 乃至 G_{n-1} に含まれる根に対応する前記シンドローム計算回路は、ユーザデータ、またはユーザデータおよび第 1 乃至第 ($n - 1$) パリティのうち少なくとも 1 つと、0 と、のうちいずれか一方を選択する選択回路と、

を備え、

前記第 1 乃至第 i パリティを用いた前記ユーザデータの誤り訂正処理と第 i 誤り訂正処理とすると、第 1 乃至第 n 誤り訂正処理を前記シンドローム計算部により実施することを特徴とする請求項 1 に記載のメモリコントローラ。

【請求項 17】

第 i 誤り訂正処理では、 G_i に含まれる根に対応する前記シンドローム計算回路以外の前記シンドローム計算回路の動作を停止させる、ことを特徴とする請求項 16 に記載のメモリコントローラ。

【請求項 18】

前記シンドローム計算回路が P (P は 2 以上の整数) ビット並列演算を行う場合に、前記符号化部は、前記第 1 乃至第 n パリティのうちビット数が P の倍数でないパリティがある場合、当該パリティのビット数が P の倍数となるようゼロ挿入処理を実施する、ことを特徴とする請求項 16 または 17 に記載のメモリコントローラ。

【請求項 19】

前記ゼロ挿入処理を、前記第 1 乃至第 n パリティのうちビット数が P の倍数でないパリティの生成処理において、当該パリティのビット数が P の倍数となるよう前記生成処理へゼロを入力する処理とすることを特徴とする請求項 18 に記載のメモリコントローラ。

【発明の詳細な説明】

【技術分野】

【0001】

本実施形態は、メモリコントローラ、記憶装置、誤り訂正装置および誤り訂正方法に関する。

【背景技術】

【0002】

メモリに格納されたユーザデータをメモリから読み出した場合に、本来の値と異なった

10

20

30

40

50

値に変化したり、送信装置が送信したデータが受信装置に受信されると受信されたデータが本来の値と異なった値に変化したり、等の現象が生じることがある。

【0003】

この問題に対応するために、ユーザデータを誤り訂正符号化してパリティデータを生成し、ユーザデータとパリティデータを組にして管理するという方法が一般的に取られる。誤り訂正符号としては、例えば、BCH符号やRS（リードソロモン）符号が存在する。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】米国特許第7418644号明細書

10

【特許文献2】特開2005-56397号公報

【特許文献3】特開2009-211209号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、従来技術では、ユーザデータに対して、誤りの規模に応じて処理量を変更する適応的な誤り訂正を行うことができない。

【課題を解決するための手段】

【0006】

本実施形態によれば、ユーザデータに誤り訂正符号化処理を行うことにより第1乃至第nパリティを生成し、第1乃至第nパリティ各々に誤り訂正符号化処理を行うことにより第1乃至第n外部パリティを生成する符号化部と、ユーザデータ、第1乃至第nパリティ、及び第1乃至第n外部パリティを用いて、誤り訂正復号処理を行う復号部と、を備え、第iパリティ（iは1以上n以下の整数）を生成するために用いられる生成多項式は、第1乃至第（i-1）パリティを生成するために用いられる生成多項式に基づいて選択される。

20

【図面の簡単な説明】

【0007】

【図1】図1は、第1の実施の形態にかかるメモリシステムの構成例を示すブロック図である。

30

【図2】図2は、第1の実施の形態のパリティの格納概念の一例を示す図である。

【図3】図3は、第1の実施の形態の符号演算の概念を示す図である。

【図4】図4は、第1の実施の形態の復号演算の概念を示す図である。

【図5】図5は、第1の実施の形態の符号演算手順の一例を示すフローチャートである。

【図6】図6は、第1の実施の形態の復号演算手順の一例を示すフローチャートである。

【図7】図7は、ユーザデータとパリティの格納形式の例を示す図である。

【図8】図8は、ユーザデータとパリティの格納形式の例を示す図である（パリティ数がユーザデータにより異なる例）。

【図9】図9は、ユーザデータとパリティの格納形式の例を示す図である（複数の記憶装置に格納する例）。

40

【図10】図10は、第2の実施の形態にかかるメモリシステムの構成例を示すブロック図である。

【図11】図11は、第2の実施の形態の符号演算の概念を示す図である。

【図12】図12は、第3の実施の形態にかかるメモリシステムの構成例を示すブロック図である。

【図13】図13は、パリティ保護表の一例を示す図である。

【図14】図14は、第3の実施の形態の符号演算の概念を示す図である。

【図15】図15は、第3の実施の形態の復号演算の概念とユーザデータおよびパリティの格納例を示す図である。

【図16】図16は、ユーザデータ1個とパリティ複数個とから外部パリティを計算する

50

場合のユーザデータおよびパリティの格納例を示す図である。

【図 17】図 17 は、ユーザデータとパリティの格納形式の例を示す図である（パリティ専用ページにパリティを格納する例）。

【図 18】図 18 は、ユーザデータとパリティの格納形式の例を示す図である（パリティ専用ページにパリティを格納し、ページによりパリティ量が異なる例）。

【図 19】図 19 は、ユーザデータとパリティの格納形式の例を示す図である（パリティ専用ページにパリティを格納し、他のページにもパリティを格納する例）。

【図 20】図 20 は、第 4 の実施の形態にかかる誤り訂正装置を備える通信装置の構成例を示すブロック図である。

【図 21】図 21 は、第 5 の実施の形態にかかる半導体記憶装置における復号部の構成例を示すブロック図である。

【図 22】図 22 は、第 5 の実施の復号処理手順の一例を示すフローチャートである。

【図 23】図 23 は、シンドローム計算部の構成例を示す図である。

【図 24】図 24 は、1 段階目の誤り訂正処理における回路使用状態を示す図である。

【図 25】図 25 は、第 5 の実施の形態のシンドローム計算の処理手順の一例を示すフローチャートである。

【図 26】図 26 は、外部パリティを用いた復号も行う場合の復号部の構成例を示す図である。

【図 27】図 27 は、第 6 の実施の形態にかかる半導体記憶装置におけるシンドローム計算部のシンドローム S_i 計算回路の構成例を示すブロック図である。

【図 28】図 28 は、第 6 の実施の形態のデータフォーマットの一例を示す図である。

【図 29】図 29 は、 d_i ビットのゼロパディングの位置をパリティ # 1 の最後の部分とした場合のデータフォーマットの一例を示す図である。

【図 30】図 30 は、3 段階の誤り訂正方式を採用する場合のデータフォーマットの一例を示す図である。

【図 31】図 31 は、復号時の P ビット並列処理に対応したパリティ # n を生成するための符号化部の構成例を示す図である。

【発明を実施するための形態】

【0008】

以下に添付図面を参照して、実施形態にかかるメモリコントローラ、記憶装置、誤り訂正装置および誤り訂正方法を詳細に説明する。なお、これらの実施形態により本発明が限定されるものではない。

【0009】

（第 1 の実施の形態）

図 1 は、第 1 の実施の形態にかかるメモリシステムの構成例を示すブロック図である。本実施の形態の半導体記憶装置（記憶装置）1 は、メモリコントローラ 2 と半導体メモリ部（不揮発性メモリ）3 を備える。記憶装置 1 は、ホスト 4 と接続可能であり、図 1 ではホスト 4 と接続された状態を示している。ホスト 4 は、例えば、パーソナルコンピュータ、携帯端末などの電子機器である。

【0010】

半導体メモリ部 3 は、データを不揮発に記憶する不揮発性メモリであり、例えば、NAND メモリである。なお、ここでは、半導体メモリ部 3 として NAND メモリを用いる例を説明するが、半導体メモリ部 3 として NAND メモリ以外の記憶手段を用いてもよい。NAND メモリでは、一般にページと言われる書込み単位（最小書込み）が存在し、ページ単位で書込みが行われる。

【0011】

メモリコントローラ 2 は、ホスト 4 からの書込みコマンド（要求）に従って半導体メモリ部 3 への書込みを制御し、またホスト 4 からの読み出しコマンド（要求）に従って NAND メモリ 3 からの読み出しを制御する。メモリコントローラ 2 は、Host I/F 21、メモリ I/F 22、制御部 23、符号化/復号部 24 を備える。符号化/復号部 24 は

10

20

30

40

50

、符号化部 25 と復号部 26 を備える。Host I / F 21、メモリ I / F 22、制御部 23、符号化部 25 および復号部 26 は、内部バス 20 で接続されている。

【0012】

Host I / F 21 は、ホスト 4 との間インタフェース規格に従った処理を実施し、ホスト 4 から受信した命令、ユーザデータなどを内部バス 20 に出力する。また、Host I / F 21 は、半導体メモリ部 3 から読み出されたユーザデータ、制御部 23 からの応答などをホスト 4 へ送信する。

【0013】

メモリ I / F 22 は、制御部 23 の指示に基づいて、書込みデータを半導体メモリ部 3 へ書込む処理および半導体メモリ部 3 からの読み出し処理を制御する。

10

【0014】

制御部 23 は、半導体記憶装置 1 の各構成要素を統括的に制御する制御部である。制御部 23 は、ホスト 4 から Host I / F 21 経由で命令を受けた場合に、その命令に従った制御を行う。例えば、制御部 23 は、ホスト 4 からの命令に従って、半導体メモリ部 3 へのユーザデータおよびパリティの書き込み、半導体メモリ部 3 からのユーザデータの読み出しなどを、メモリ I / F 22 へ指示する。

【0015】

符号化部 25 は、内部バス 20 に転送されたデータに基づいて、誤り訂正符号化処理を実施しパリティを生成する。誤り訂正符号としてはどのような符号を用いてもよいが、例えば、BCH 符号や RS (Reed - Solomon) 符号等を用いることができる。以下の説明では、BCH 符号を用いる例について説明するが、誤り訂正符号の種類はこれに限定されないが、符号化後に情報データとパリティとに分離できる符号が好ましい。

20

【0016】

復号部 26 は、半導体メモリ部 3 からの読み出されたユーザデータおよびパリティに基づいて復号処理を実施する。ユーザデータに誤りの無い場合、半導体メモリ部 3 からの読み出されたユーザデータをそのまま内部バス 20 へ出力し、ユーザデータに誤りのある場合、パリティを用いて誤り訂正を行った後にユーザデータを内部バス 20 へ出力する。

【0017】

NAND メモリでは、ページ単位で書込みが行われるが、信頼性の異なる（誤りの発生する確率が異なる）ページが存在することがある。例えば、NAND メモリのメモリセルとして多値セルを用いる場合、一般に、1 つのメモリセルに格納される各 bit には異なるページが割当てられる。例えば、1 つのメモリセルに格納されるビットを 3 ビットとすると、3 ビットにはそれぞれ異なるページが割当てられる。このような場合、ページにより信頼性が異なることがある。

30

【0018】

信頼性の低いページでは、誤り訂正能力の高いパリティを生成しておくことが望ましい。誤り訂正能力の高いパリティを実現するためにはパリティのサイズを大きくすることになるが、NAND メモリの各ページのサイズは同一である。このため、信頼性の低いページに対応した誤り訂正能力のパリティを生成するためには、例えば、信頼性の低いページのパリティを 2 つに分割して、信頼性の高いページに分割したうちの 1 つのパリティを格納する方法が考えられる。しかし、この方法では、信頼性の低いページのユーザデータを読み出す際に、分割したパリティを格納した別のページも読み出す必要があり、読み出し速度が低下する。また、信頼性の低いページでも、実際にはほとんどのページでは、誤りは無いまたは誤り数が最大の想定数より少ない場合が多く、信頼性の高いページに対応した誤り訂正能力で十分対応可能である。

40

【0019】

本実施の形態では、誤り訂正符号として、誤り訂正能力を高めた分のパリティを追加することにより段階的に誤り訂正能力の高い符号化を行うことが可能な符号を用いる。図 2 は、本実施の形態のパリティの格納概念の一例を示す図である。図 2 の上側には、比較例として、信頼性の低いページのユーザデータ # 1 に対して誤り訂正能力の高いパリティ #

50

1 を生成し、信頼性の高いページのユーザデータ # 2 に対して誤り訂正能力の低いパリティ # 2 を生成して格納した例を示している。このように信頼性の高さに応じて誤り訂正能力を変えてパリティを生成してユーザデータと同一のページに格納するとページごとにパリティのサイズが異なることになってしまう。

【 0 0 2 0 】

これに対して、図 2 の下側に示した本実施の形態では、1 段階で誤り訂正能力の高いパリティ # 1 を生成するのではなく、1 段階目としてユーザデータ # 1 に基づいて所定の誤り訂正能力を有するパリティ # 1 - 1 を生成し、2 段階目としてユーザデータ # 1 とパリティ # 1 - 1 を用いてパリティ # 1 - 2 を生成し、3 段階目としてユーザデータ # 1 とパリティ # 1 - 1 とパリティ # 1 - 2 を用いてパリティ # 1 - 3 を生成し、... というように段階的にパリティを生成する。そして、復号時には、まず、パリティ # 1 - 1 を用いて復号を行い誤り訂正不可であった場合に、パリティ # 1 - 2 を読み出し、パリティ # 1 - 1 およびパリティ # 1 - 2 を用いて復号を行い誤り訂正不可であった場合にパリティ # 1 - 3 を読み出し、... というように段階的に復号を行う。このような符号化および復号化を行うと、例えば、図 2 に示すように、信頼性の低い（平均誤りビット数の多い）ページ # 1 にユーザデータ # 1 を格納する場合、ユーザデータ # 1 に基づいて生成したパリティ # 1 - 1 を当該パリティに対応するユーザデータ # 1 が格納されるページに格納し、パリティ # 1 - 2 を信頼性の高いページ # 2 に格納することができる。復号時には通常は、まずユーザデータ # 1 およびパリティ # 1 - 1 を読み出す。このように構成しておくこと、パリティ # 1 - 1 を用いた復号で誤り訂正が可能であった場合には、ページ # 1 以外のページ（図 2 の例ではページ # 2）を読み出さなくてよい。なお、図 2 では、図の簡略化のため、2 段階のパリティを例に説明しているが、本実施の形態のパリティの生成は 2 段階に限らず、後述するように何段階で生成してもよい。

【 0 0 2 1 】

一方、パリティを多段階に分けて生成する方法として、独立した符号化処理を段階的に行う方法も考えられる。例えば、1 段階目としてユーザデータに対して所定の誤り能力を有するパリティ A を生成する。そして、2 段階目としてユーザデータに基づいて別の符号化処理により所定の誤り能力より高い誤り能力を有するパリティ B を生成し、図 2 と同様に、ユーザデータと同一ページにパリティ A を格納し、同一でないページにパリティ B を格納するという事も考えられる。しかし、この場合、パリティ B はパリティ A とはまったく独立に生成されているため、パリティ B を用いた復号の際にパリティ A を用いることができない。一方、上述した誤り訂正能力を高めた分のパリティを追加する符号を用いれば、パリティ # 1 - 2 を用いた復号の際にパリティ # 1 - 1 も用いることができる。したがって、合計のパリティサイズが同じであれば、誤り訂正能力を高めた分のパリティを追加する符号の方が独立した多段階の符号化に比べ高い誤り訂正能力を有し、また、同じ誤り訂正能力であれば、誤り訂正能力を高めた分のパリティを追加する符号の方がパリティのサイズが少なくてすむ。

【 0 0 2 2 】

このため、本実施の形態では、 i (i は 2 以上 n 以下の整数) 段階目のパリティのサイズは、1 乃至 i 段階目の全てのパリティを用いて誤り訂正を行う場合と同等の誤り訂正能力を有するよう単独で生成されるパリティのサイズより小さくなるように n 段階でパリティを生成する。例えば、上記の例では、本実施の形態の 2 段階目のパリティであるパリティ # 1 - 2 のサイズは、パリティ A と独立に生成した 2 段階目のパリティであるパリティ B のサイズより小さい。

【 0 0 2 3 】

以下、本実施の形態の誤り訂正符号化処理（以下、符号演算と略す）および復号演算について説明する。図 3 は、本実施の形態の符号演算の概念を示す図である。図 3 に示すように、本実施の形態の符号化部 25 は、符号演算として次のように n 段階でパリティを計算する。

(1) ユーザデータを用いて $G_1(x)$ によりパリティ # 1 を計算。

10

20

30

40

50

- (2) ユーザデータ + パリティ # 1 を用いて $G_2(x)$ によりパリティ # 2 を計算。
 (3) ユーザデータ + パリティ # 1 + パリティ # 2 を用いて $G_3(x)$ によりパリティ # 3 を計算。

...

- (n) ユーザデータ + パリティ # 1 + パリティ # 2 + ... + パリティ # (n - 1) を用いて $G_n(x)$ によりパリティ # n (n は 1 以上の整数) を計算。

なお、 $G_i(x)$ ($i = 1, 2, \dots, n$) は生成多項式であり、 $G_i(x)$ については後述する。

【0024】

図4は、本実施の形態の復号演算の概念を示す図である。図4に示すように、本実施の形態の復号部26は、復号演算として、次のようにn段階で誤り訂正を実施する。

- (1) パリティ # 1 を用いてユーザデータ + パリティ # 1 の誤り訂正を実施する。
 (2) (1) の誤り訂正に失敗した場合、パリティ # 1 ~ パリティ # 2 を用いてユーザデータ + パリティ # 1 + パリティ # 2 の誤り訂正を実施する。
 (3) (2) の誤り訂正に失敗した場合、パリティ # 1 ~ パリティ # 3 を用いてユーザデータ + パリティ # 1 + パリティ # 2 + パリティ # 3 の誤り訂正を実施する。

...

- (n) (n - 1) の誤り訂正に失敗した場合、パリティ # 1 ~ パリティ # n を用いてユーザデータ + パリティ # 1 + パリティ # 2 + パリティ # 3 + ... + パリティ # n の誤り訂正を実施する。

【0025】

なお、各段階で、誤り訂正に失敗した場合に次の段階へ進み、各段階で誤り訂正に成功した場合は次の段階へは進まず復号演算を終了する。n段階目の誤り訂正処理で誤り訂正に失敗した場合は、復号部26は、誤り訂正に失敗したことを制御部23へ通知する。制御部23は、誤り訂正に失敗した場合の所定の手順(例えば、ホスト4へ読み出したユーザデータに誤りがあることを通知する等)を実施する。

【0026】

また、パリティ # 1 を用いてユーザデータ + パリティ # 1 の復号処理を実施した際に、ユーザデータに誤りが無かった場合は、ユーザデータに対して誤り訂正は実施せず、読み出したユーザデータをそのまま出力する。なお、ユーザデータに誤り検出符号を付加しておいて、誤り検出符号を用いてユーザデータの誤りの有無を判定するようにしてもよい。

【0027】

以上の演算を実施する場合に、パリティ # 1 からパリティ # i までを用いて誤り訂正を行うよりも、パリティ # 1 からパリティ # (i + 1) までを用いて誤り訂正を行う方がより強力な誤り訂正能力を有するようにするために、本実施の形態では、例えば、以下のような符号演算を実施する。

【0028】

なお、以下の説明では、BCH符号を用いる場合の符号演算方法および復号演算方法について説明するが、BCH符号に限定されるものではない。

【0029】

まず、BCH符号に関する一般的な説明を行う。ユーザデータ、パリティ、符号語などは0または1の値の列であり、これらを表現するためには様々な方法が存在するが、ここでは、多項式表現を用いて説明する。例えば、ユーザデータ10101は、多項式表現によりユーザデータ多項式 $x^4 + x^2 + 1$ と表現することができる。

【0030】

符号演算に際しては、ユーザデータ多項式に生成多項式と呼ばれる多項式を用いた四則演算を施して、パリティの多項式表現であるパリティ多項式を計算する。そして、ユーザデータ多項式とパリティ多項式を組み合わせることで符号多項式を計算する。

【0031】

上記四則演算は、

10

20

30

40

50

- ・ガロア体 $GF(2)$
- ・ガロア拡大体 $GF(2^m)$
- ・原始多項式
- ・最小多項式

に基づいて行われるが、これらに関しては、詳細は省略する。

【0032】

ガロア拡大体 $GF(2^m)$ は 2^m 種類 の元があり、それらは、 $0, 1, \dots, 2, 3, \dots, M-1, M (M = 2^m - 2)$ と表現される。

【0033】

生成多項式を用いて計算される符号多項式に誤りが加わった結果（時間の経過とともに様々なストレスを受けた結果）得られる多項式を受信多項式とすると、受信多項式に基づいて誤り多項式を計算し、誤り多項式を用いて符号多項式が計算可能であるためには、生成多項式を適切に選択する必要がある。

10

【0034】

具体的には、 t ビットの誤りが計算可能であるためには、生成多項式 $= 0$ が、 r を任意の整数とし、 $r, r+1, r+2, \dots, 2t+r-2, 2t+r-1$ の $2t$ 個の根を持っていればよい（BCH限界）。

【0035】

次に、以上行った BCH 符号の説明に基づき、本実施の形態における生成多項式の選択方法の説明を行う。図3に示した n 段階の誤り訂正符号では、 $G_1(x), G_2(x), \dots, G_n(x)$ の n 個の生成多項式を用いて符号化を行う。なお、以下、添え字付の文字（ $r_1, b b_1$ 等）を、上付文字（指数の肩）や添え字（下付文字）のなかで用いる場合、上付または下付ではない表記（ $r_1, b b_1$ 等）で記載するが、両者は同じものを示している。

20

【0036】

このとき、

(1) r_1 を整数とし、 $G_1(x) = 0$ が $r_1, r_1+1, r_1+2, \dots, 2t_1+r_1-2, 2t_1+r_1-1$ の $2t_1 (= 2 \times t_1)$ 個の根を持っており、

(2) r_2 を整数とし、 $G_1(x) * G_2(x) = 0$ が、 $r_2, r_2+1, r_2+2, \dots, 2t_2+r_2-2, 2t_2+r_2-1$ の $2t_2$ 個の根を持っており、

30

...

(n) r_n を整数とし、 $G_1(x) * G_2(x) \dots * G_n(x) = 0$ が、 $r_n, r_n+1, r_n+2, \dots, 2t_n+r_n-2, 2t_n+r_n-1$ の $2t_n$ 個の根を持っている、

という n 種の条件を満たすように n 個の生成多項式 $G_1(x), G_2(x), \dots, G_n(x)$ を選択する。なお、 r_1, r_2, \dots, r_n は、以下の関係を満たす。

- ・ $r_2 = r_1, r_1 + 2t_1 - 1 = r_2 + 2t_2 - 1$
- ・ $r_3 = r_2, r_2 + 2t_2 - 1 = r_3 + 2t_3 - 1$
- ...
- ・ $r_n = r_{n-1}, r_{n-1} + 2t_{n-1} - 1 = r_n + 2t_n - 1$

40

これにより、 $t_n > t_{n-1} > \dots > t_2 > t_1$ となる。

【0037】

上記の生成多項式選択を行うと、

$t_0 = 0, b_i = t_i - t_{i-1} (1 \leq i \leq n)$ とするとき、

$G_i(x)$ によって計算されるパリティ # i は、 b_i ビットの訂正能力を持つ。

【0038】

また、上記の生成多項式選択を行うと、パリティ # 1 からパリティ # i を用いると、ユーザデータ + パリティ # 1 + パリティ # 2 + ... + パリティ # i 内に t_i ビットまでの誤りがある場合に、ユーザデータ内の誤りを訂正し、ユーザデータが本来取るべき値を得ることが可能である。

50

【0039】

したがって、上記の生成多項式選択を行うと、パリティ#1からパリティ#nを用いると、ユーザデータ+パリティ#1+パリティ#2+...+パリティ#n内にt_nビットまでの誤りがある場合にユーザデータ内の誤りを訂正し、ユーザデータが本来取るべき値を得ることが可能である。

【0040】

上記のn-1種の条件を満たすよう選択した生成多項式の一例として、r₁=r₂=...=r_n=1とした例を説明する。iを根とするGF(2)上の最小多項式をM_i(x)と置く。0<b₁<b₂<b₃<...<b_nを満たす任意の整数b₁、b₂、b₃、...、b_nに関して以下の式(1)のようにGG_i(x)を置く。

$$GG_i(x) = M_1(x) * M_3(x) * M_5(x) * \dots * M_{2bb_{i-1}}(x) \quad (1 \quad i \quad n) \quad \dots (1)$$

10

【0041】

このとき、GG_i(x)を生成多項式とするBCH符号はb_iビットまでの誤り訂正が可能である。

【0042】

以下の式(2)のように、G_i(x)を置くと、

$$G_0(x) = 1, \\ G_i(x) = GG_i(x) / GG_{i-1}(x) \\ = M_{2bb_{(i-1)+1}}(x) * M_{2bb_{(i-1)+3}}(x) * \dots * M_{2bb_{i-1}}(x) \quad (1 \quad i \quad n) \quad \dots (2)$$

20

なお、b_{b(i-1)}=b_{b_{i-1}}とする。

【0043】

式(2)で示した生成多項式は、上記のn-1種の条件を満たすよう選択した生成多項式の一例となり、上記のn-1種の条件において、b_i=b_{b_i}-b_{b_{i-1}}とし、t_i=b_{b_i}とした例に相当する。

【0044】

図5は、本実施の形態の符号演算手順の一例を示すフローチャートである。図5の手順は、上記で説明した生成多項式を選択してn段階の符号化を行う場合の符号演算手順を示している。まず、i=1とし、符号語#0をユーザデータ(符号語#0=ユーザデータ)とする(ステップS1)。次に、符号語#iを符号語#(i-1)x^K(Kは生成多項式G_i(x)の次数)とする(符号語#i=符号語#(i-1)x^K) (ステップS2)。パリティ#iを、符号語#iの生成多項式G_i(x)による剰余として求める(ステップS3)。

30

【0045】

そして、符号語#iにパリティ#iを加えて符号語#iを更新(符号語#i=符号語#i+パリティ#i)する(ステップS4)。その後、i=i+1とし(ステップS5)、n<iであるか否かを判断する(ステップS6)。n<iである場合(ステップS6 Yes)、処理を終了し、n<iでない場合(ステップS6 No)、ステップS2に戻る。

40

【0046】

図6は、本実施の形態の復号演算手順の一例を示すフローチャートである。図6の手順は、図5で説明した符号演算を行った符号語が、半導体メモリ部3に格納された後に、読み出された場合の復号部26における復号手順を示している。まず、i=1とし、ユーザデータとパリティ#1を受信する(ステップS12)。具体的には、制御部23の指示に基づいてメモリI/F22が半導体メモリ部3から読み出したユーザデータと当該ユーザデータに対応するパリティ#1を内部バス20経由で受信する。

【0047】

ユーザデータとパリティ#1からパリティ#iまでのパリティに基づいて誤り訂正を試みる(ステップS13)。そして、誤り訂正に成功したか否かを判断し(ステップS14

50

)、誤り訂正に成功した場合(ステップS14 Yes)、処理を終了する。

【0048】

誤り訂正に成功しなかった場合(ステップS14 No)、 $i = i + 1$ とし(ステップS15)、 $n < i$ であるか否かを判断する(ステップS16)。 $n < i$ である場合(ステップS16 Yes)、処理を終了する。 $n < i$ でない場合(ステップS16 No)、パリティ# i を受信し(ステップS17)、ステップS13へ戻る。パリティ# i の受信は、具体的には、復号部26が誤り訂正の失敗を制御部23へ通知し、制御部23が当該通知に基づいてメモリI/F22に対して、パリティ# i の読出しを指示する。そして、メモリI/F22により半導体メモリ部3から読み出されたパリティ# i を内部バス20経由で受信する。

10

【0049】

次に、ユーザデータとパリティの半導体メモリ部3への格納方法の例について説明する。図7~図8は、ユーザデータとパリティの半導体メモリ部3への格納形式の例を示す図である。

【0050】

図7の例では、すべてのユーザデータのサイズが同一(ユーザデータ#1, ユーザデータ#2, ..., ユーザデータ# U のサイズが同一)とした例を示しており、全てのユーザデータに対して同一サイズのパリティを生成して半導体メモリ部3へ格納している。この例では、ユーザデータ# i ($i = 1, 2, \dots, U$)に対して、上述の1段階目のパリティ#1に相当するパリティ# $i - 1$, 上述の2段階目のパリティ#2に相当するパリティ# $i - 2$, 上述の3段階目のパリティ#3に相当するパリティ# $i - 3$, ..., 上述の n 段階目のパリティ#2に相当するパリティ# $i - n$ を生成して格納している。例えば、半導体メモリ部3がNANDメモリである場合、図7の一行を1ページとしてもよい。

20

【0051】

なお、図7では、すべてのユーザデータのサイズを同一としているが、ユーザデータのうち一部またはすべてのサイズが異なっても同様にユーザデータ# i に対してパリティ# $i - 1 \sim i - n$ を生成して格納することができる。

【0052】

また、図7の例では、 $1 \leq i \leq U$ の全ての i について、パリティ# $i - 1$ のサイズを同一とし、パリティ# $i - 2$ のサイズを同一とし、パリティ# $i - 3$ のサイズを同一とし、...、パリティ# $i - n$ のサイズを同一としているが、これらパリティのうち一部またはすべてのサイズが異なってもよい。

30

【0053】

また、図7の例では、すべてのユーザデータのパリティの数(n)が同一であるが、ユーザデータのうちの一部または全てについてパリティの数が異なってもよい。

【0054】

また、図7の例では、パリティ# $i - 1$, パリティ# $i - 2$, パリティ# $i - 3$, ..., パリティ# $i - n$ が連続する領域に記憶されているが、これらパリティのうち一部またはすべてが非連続な領域に記憶されてもよい。

40

【0055】

図8の例では、ユーザデータ#2とユーザデータ#1およびユーザデータ#3とで、パリティの数(n)が異なり、また、パリティ# $i - 1$, パリティ# $i - 2$, パリティ# $i - 3$, ..., パリティ# $i - n$ を非連続な領域に格納した例を示している。ユーザデータ#2では $n = 3$ とし、ユーザデータ#1およびユーザデータ#3では、 $n = 1$ としている。そして、ユーザデータ#2に連続する領域にパリティ# $2 - 1$ を格納し、パリティ# $2 - 2$ はユーザデータ#1に連続する領域に格納し、パリティ# $2 - 3$ はユーザデータ#3に連続する領域に格納している。各ユーザデータに対応するパリティの数、各パリティの配置位置は図8の例に限定されない。例えば、半導体メモリ部3がNANDメモリである場合、図8の一行を1ページとしてもよい。

【0056】

50

図 6、図 7、図 8 の例では、半導体メモリ部 3 が 1 つの記憶装置である例を示したが、図 9 は、半導体メモリ 3 がメモリ部 3 - 1 ~ 3 - (n + 1) の複数の記憶装置で構成される例を示している。メモリ部 3 - 1 ~ 3 - (n + 1) は、例えばそれぞれ異なるメモリチップであってもよいし、それぞれ異なる形式（例えば、NANDメモリと磁気記憶装置等）の記憶装置であってもよい。図 9 の例では、メモリ部 3 - 1 にユーザデータ # 1 ~ # U を格納し、パリティ # i - 1 をメモリ部 3 - 2 に格納し、パリティ # i - 2 をメモリ部 3 - 3 に格納し、パリティ # i - 3 をメモリ部 3 - 4 に格納し、...、パリティ # i - n をメモリ部 3 - (n + 1) に格納している。図 9 は、各メモリ部へユーザデータおよび各パリティを分散して格納する場合の一例を示しており、具体的な分散方法は、図 9 の例に限定されない。また、図 9 の例では、半導体メモリ部 3 内の異なる記憶装置にユーザデータおよび各パリティを分散して格納する例を示したが、ユーザデータおよび各パリティの一部を半導体メモリ部 3 外の別の記憶装置に格納してもよい。

10

【 0 0 5 7 】

以上のように、本実施の形態では、パリティを多段階で生成する。そして、復号時に、1 段階目と 2 段階目のパリティを 2 段階目の誤り訂正で用い、1 段階目、2 段階目および 3 段階目のパリティを 3 段階目の誤り訂正で用い、... というように、他の段階のパリティを復号において用いることにより誤り訂正能力を向上させることができる符号を、符号演算において生成するようにした。このため、平均的な読み出し速度を低下させずに、誤りの規模に応じて誤り訂正に用いるパリティ数を適応的に変化させることができ、また、独立して多段階のパリティを生成する場合に比べ、パリティ量が同一であれば誤り訂正能力を

20

【 0 0 5 8 】

(第 2 の実施の形態)

図 10 は、第 2 の実施の形態にかかるメモリシステムの構成例を示すブロック図である。本実施の形態の半導体記憶装置 1 a は、メモリコントローラ 2 a と半導体メモリ部 3 を備える。半導体メモリ部 3 は第 1 の実施の形態の半導体メモリ部 3 と同様である。記憶装置 1 a は、第 1 の実施の形態の記憶装置 1 と同様にホスト 4 と接続可能である。第 1 の実施の形態と同様の機能を有する構成要素は、第 1 の実施の形態と同一の符号を付して重複する説明を省略する。

【 0 0 5 9 】

メモリコントローラ 2 a は、符号化 / 復号部 2 4 を符号化 / 復号部 2 4 a に替える以外は、第 1 の実施の形態のメモリコントローラ 2 と同様である。符号 / 復号部 2 4 a は、符号化部 2 5 a と復号部 2 6 a で構成される。本実施の形態の動作は、半導体メモリ部 3 への書込み時の符号演算および半導体メモリ部 3 からの読み出し時の復号演算を除き、第 1 の実施の形態と同様である。

30

【 0 0 6 0 】

本実施の形態の符号化部 2 5 a における符号演算について説明する。図 11 は、本実施の形態の符号演算の概念を示す図である。図 11 に示した $G_1(x) \sim G_n(x)$ は、第 1 の実施の形態で述べた生成多項式 $G_1(x) \sim G_n(x)$ と同様である。

【 0 0 6 1 】

図 11 に示すように、本実施の形態の符号化部 2 5 a は、符号演算として次のように n 段階でパリティを計算する。

(1) ユーザデータを用いて $G_1(x)$ によりパリティ # 1 を計算し、パリティ # 1 のパリティである外部パリティ # 1 を計算する。

(2) ユーザデータを用いて $G_2(x)$ によりパリティ # 2 を計算し、パリティ # 2 のパリティである外部パリティ # 2 を計算する。

(3) ユーザデータを用いて $G_3(x)$ によりパリティ # 3 を計算し、パリティ # 3 のパリティである外部パリティ # 3 を計算する。

...

(n) ユーザデータを用いて $G_n(x)$ によりパリティ # n (n は 1 以上の整数) を計算

40

50

し、パリティ # n のパリティである外部パリティ # n を計算する。

【 0 0 6 2 】

次に、本実施の形態の復号部 2 6 a における復号演算について説明する。本実施の形態の復号部 2 6 a は、復号演算として次のように n 段階で誤り訂正を実施する。

(1) 外部パリティ # 1 を用いてパリティ # 1 の誤り訂正を行う。パリティ # 1 の誤り訂正に成功した場合、誤り訂正後のパリティ # 1 (誤りの無かった場合には元のパリティ # 1) とユーザデータを用いて、ユーザデータの誤り訂正を実施する。

(2) (1) でユーザデータの誤り訂正に失敗した場合、外部パリティ # 2 を用いてパリティ # 2 の誤り訂正を行う。パリティ # 2 の誤り訂正に成功した場合、誤り訂正後のパリティ # 1 と誤り訂正後のパリティ # 2 とユーザデータとを用いて、ユーザデータの誤り訂正を実施する。

(3) (2) でユーザデータの誤り訂正に失敗した場合、外部パリティ # 3 を用いてパリティ # 3 の誤り訂正を行う。パリティ # 3 の誤り訂正に成功した場合、誤り訂正後のパリティ # 1 と誤り訂正後のパリティ # 2 と誤り訂正後のパリティ # 3 とユーザデータとを用いて、ユーザデータの誤り訂正を実施する。

...

(n) ($n - 1$) でユーザデータの誤り訂正に失敗した場合、外部パリティ # n を用いてパリティ # n の誤り訂正を行う。パリティ # n の誤り訂正に成功した場合、誤り訂正後のパリティ # 1 と誤り訂正後のパリティ # 2 と誤り訂正後のパリティ # 3 と...誤り訂正後のパリティ # n とユーザデータとを用いて、ユーザデータの誤り訂正を実施する。

【 0 0 6 3 】

なお、外部パリティ # i ($i = 1, 2, \dots, n$) を用いたパリティ # i の誤り訂正に失敗した場合、ユーザデータの誤り訂正ができないため、復号部 2 6 a は制御部 2 3 へ当該ユーザデータの誤り訂正の失敗を通知する。

【 0 0 6 4 】

なお、ユーザデータ、パリティ # i 、外部パリティ # i の半導体メモリ部 3 への格納方法に制約はなく、ユーザデータ、パリティ # i 、外部パリティ # i を連続した領域に格納してもよいし、分散させて格納してもよい。また、第 1 の実施の形態で述べたように、複数の記憶装置に分散させて格納してもよい。例えば、半導体メモリ部 3 が NAND メモリである場合、パリティ # 1 ~ # n をユーザデータと同一ページ内に格納し、外部パリティ # 1 ~ # n を半導体メモリ部 3 外の別の記憶装置に格納してもよい。また、パリティ # 1 をユーザデータと同一ページ内に格納し、パリティ # 2 ~ # n と外部パリティ # 1 ~ # n を別のページに格納してもよい。

【 0 0 6 5 】

以上のように、本実施の形態では、ユーザデータに基づいて第 1 の実施の形態と同様の生成多項式を用いて各段階のパリティを生成し、各段階のパリティのパリティである外部パリティを生成するようにした。このため、ユーザデータの誤り訂正を行う演算において、パリティに誤りは含まれないとして演算することができ、第 1 の実施の形態に比べ、演算を高速化することができる。

【 0 0 6 6 】

(第 3 の実施の形態)

図 1 2 は、第 3 の実施の形態にかかるメモリシステムの構成例を示すブロック図である。本実施の形態の半導体記憶装置 1 b は、メモリコントローラ 2 b と半導体メモリ部 3 を備える。半導体メモリ部 3 は第 1 の実施の形態の半導体メモリ部 3 と同様である。記憶装置 1 b は、第 1 の実施の形態の記憶装置 1 と同様にホスト 4 と接続可能である。第 1 の実施の形態と同様の機能を有する構成要素は、第 1 の実施の形態と同一の符号を付して重複する説明を省略する。

【 0 0 6 7 】

メモリコントローラ 2 b は、符号 / 復号部 2 4 を符号 / 復号部 2 4 b に替える以外は、第 1 の実施の形態のメモリコントローラ 2 と同様である。符号 / 復号部 2 4 b は、符号化

10

20

30

40

50

部 2 5 b と復号部 2 6 b とパリティ保護表記憶部 2 7 と構成される。本実施の形態の動作は、半導体メモリ部 3 への書込み時の符号演算および半導体メモリ部 3 からの読み出し時の復号演算を除き、第 1 の実施の形態と同様である。

【 0 0 6 8 】

第 1 の実施の形態では、 n 段階目の誤り訂正手続きの主要部において、パリティ # 1、パリティ # 2、...、パリティ # n の誤り訂正対象が、ユーザデータ + パリティ # 1 + パリティ # 2 + ... + パリティ # n であり、すべてのパリティが誤り訂正の対象となっている。すなわち、パリティ # 1、パリティ # 2、...、パリティ # n に誤りが含まれているかもしれないという前提で誤り訂正を行う。

【 0 0 6 9 】

これに対し、第 2 の実施の形態では、パリティ # 1、パリティ # 2、...、パリティ # n の誤り訂正対象が、ユーザデータのみであり、パリティ # 1、パリティ # 2、...、パリティ # n 自身はパリティ # 1、パリティ # 2、...、パリティ # n の誤り訂正の対象外である。パリティ # 1、パリティ # 2、...、パリティ # n は、パリティ # 1、パリティ # 2、...、パリティ # n とは異なる別のパリティである外部パリティ # 1、外部パリティ # 2、...、外部パリティ # n の誤り訂正対象である。

【 0 0 7 0 】

すなわち、パリティ # i ($1 \leq i \leq n$) に誤りが存在する場合には、第 1 の実施の形態では、パリティ # i 自身とパリティ # j ($i < j$) を用いてパリティ # i を訂正し、第 2 の実施の形態では、外部パリティ # i を用いてパリティ # i を訂正する。

【 0 0 7 1 】

本実施の形態では、パリティ # i の誤りを、第 1 の実施の形態の方法で訂正するか、第 2 の実施の形態の方法で訂正するか、をパリティ # i ごとに設定可能とし、パリティ # i の誤りを、どのパリティで誤り訂正するかを示すパリティ保護情報をパリティ保護表として保持する。そして、符号演算、復号演算を行う際に、パリティ保護表に従って演算を実施する。パリティ保護表を任意に設定可能とすることで、パリティ # i の誤りをどのパリティを用いて誤り訂正するかを任意に設定することができる。パリティ保護表は、パリティ保護表記憶部 2 7 に記憶される。

【 0 0 7 2 】

図 1 3 は、パリティ保護表の一例を示す図である。図 1 3 の表の縦方向は、誤り訂正対象となるパリティを示し、横方向は誤り訂正対象をする際に用いることができるパリティを示している。図 1 3 の表では、丸印は誤り訂正に用いることが可能なことを示し、バツ印はパリティを用いることができないことを示す。また、図 1 3 の表では、パリティ # i を誤り訂正対象とする場合のパリティ # 1 ~ # i は、設定により変更されることはないため「-」として示している。例えば、1 行目のパリティ # 1 は、用いることが自明なパリティ # 1 と、丸印となっているパリティ # 2、パリティ # 3、...、パリティ # n を用いて誤り訂正することができる。パリティ # 2 は、用いることが自明なパリティ # 2 と、外部パリティを用いて誤り訂正を行うことができる。パリティ # 3 は、用いることが自明なパリティ # 3 と、外部パリティを用いて誤り訂正を行うことができる。なお、図 1 3 は一例であり、パリティ保護表の具体的内容はこの例に限定されない。

【 0 0 7 3 】

図 1 4 は、本実施の形態の符号演算の概念を示す図である。図 1 4 は、図 1 3 に示したパリティ保護表に対応する符号演算の例を示している。

【 0 0 7 4 】

図 1 4 の例では、符号化部 2 5 b は、符号演算として次のように n 段階でパリティを計算する。

(1) ユーザデータを用いて $G_1(x)$ によりパリティ # 1 を計算する。

(2) ユーザデータとパリティ # 1 を用いて $G_2(x)$ によりパリティ # 2 を計算し、パリティ # 2 のパリティである外部パリティ # 2 を計算する。

(3) ユーザデータとパリティ # 1 を用いて $G_3(x)$ によりパリティ # 3 を計算し、パ

10

20

30

40

50

リティ # 3 のパリティである外部パリティ # 3 を計算する。

...

(n) ユーザデータとパリティ # 1 を用いて $G_n(x)$ によりパリティ # n (n は 1 以上の整数) を計算し、パリティ # n のパリティである外部パリティ # n を計算する。

【0075】

符号化部 25b は、パリティ保護表に基づいてパリティの生成方法を選択して、対応するパリティを生成する。例えば、図 13 の例の場合、パリティ # 1 については、外部パリティが用いられないため、第 1 の実施の形態と同様にユーザデータに基づいてパリティ # 1 を生成し、パリティ # 1 に対する外部パリティは生成しない。一方、パリティ # 2 については、外部パリティが用いるため、ユーザデータに基づいてパリティ # 2 を生成し、さらにパリティ # 2 に対する外部パリティを生成する。

10

【0076】

なお、パリティ # 1 からパリティ # ($i - 1$) までの全てが外部パリティにより保護されている場合は、ユーザデータのみに基づいてパリティ # i を生成すればよいが、パリティ # 1 からパリティ # ($i - 1$) の中に外部パリティによって保護されないパリティが存在する場合には、ユーザデータ + 外部パリティによって保護されないパリティすべてを用いてパリティ # i を生成する。

【0077】

復号部 26b は、パリティ保護表に基づいてパリティの復号方法を選択して、対応する誤り訂正処理を実施する。例えば、図 13 の例の場合、パリティ # 1 を用いた 1 段階目の誤り訂正処理では、外部パリティが生成されていないため、第 1 の実施の形態と同様にパリティ # 1 とユーザデータとに基づいてユーザデータの誤り訂正を行う。一方、パリティ # 2 については外部パリティが生成されているため、パリティ # 2 の外部パリティを用いてパリティ # 2 の誤り訂正を行った後に、パリティ # 1、パリティ # 2 およびユーザデータに基づいてユーザデータの誤り訂正を実施する。

20

【0078】

図 15 は、本実施の形態の復号演算の概念とユーザデータおよびパリティの格納例を示す図である。図 15 は、半導体メモリ部 3 への格納例として、ユーザデータ # 1、ユーザデータ # 2、ユーザデータ # 3 について、ユーザデータ # 2 にはユーザデータ # 1、# 3 より多くのパリティを割り当てる例を示している。図 15 のユーザデータ # 2 に関しては、3 段階の符号演算が実施され、ユーザデータ # 1、ユーザデータ # 3 については、1 段階の符号化が実施されている。ここでは、半導体メモリ部 3 を NAND メモリとし、ユーザデータ # 1、ユーザデータ # 2、ユーザデータ # 3 をそれぞれ異なるページに格納されるとする。

30

【0079】

具体的には、ユーザデータ # 2 に関しては、1 段階目のパリティとしてパリティ # 2 - 1 が生成され (パリティ # 2 - 1 に対する外部パリティは無し)、2 段階目のパリティとしてパリティ # 2 - 2 が生成されるとともにパリティ # 2 - 2 の外部パリティ (図 15 のパリティ # 1 - 1 が外部パリティに相当する) が生成され、3 段階目のパリティとしてパリティ # 2 - 3 が生成されるとともにパリティ # 2 - 3 の外部パリティ (図 15 のパリティ # 3 - 1 が外部パリティに相当する) が生成される。また、ユーザデータ # 1 に関しては、ユーザデータ # 1 とパリティ # 2 - 2 を通常のユーザデータと同様とみなしてパリティ # 1 - 1 を生成する。同様に、ユーザデータ # 3 に関しては、ユーザデータ # 3 とパリティ # 2 - 3 を通常のユーザデータと同様とみなしてパリティ # 3 - 1 を生成する。このように、パリティ # 1 - 1 はユーザデータ # 1 のパリティであるが、パリティ # 2 - 2 の外部パリティでもある。また、パリティ # 3 - 1 はユーザデータ # 3 のパリティであるが、パリティ # 2 - 3 の外部パリティでもある。

40

【0080】

図 15 の例では、多段階の符号化を行うユーザデータ # 2 の外部パリティを他のページに格納し、他のページのパリティ (パリティ # 1 - 1, # 3 - 1) の誤り訂正対象として

50

ユーザデータだけでなく当該ページに格納されたユーザデータ# 2のパリティを含めるように構成しているが、外部パリティの生成方法はこの方法に限定されない。他のページ(ユーザデータ# 1、ユーザデータ# 3)のユーザデータのパリティとは別にユーザページ# 2のパリティの外部パリティを生成して、半導体メモリ3上または他の記憶装置に格納するようにしてもよい。

【0081】

図15の例のユーザデータ# 2の復号演算の手順を説明する。

(1) ユーザデータ# 2およびパリティ# 2-1を用いてユーザデータの誤り訂正を実施する。

(2) (1)でユーザデータの誤り訂正に失敗した場合、外部パリティであるパリティ# 1-1とユーザデータ# 1とパリティ# 2-2を用いて、パリティ# 2-2の誤り訂正を行う。パリティ# 2-2の誤り訂正に成功した場合、パリティ# 2-1と誤り訂正後のパリティ# 2-2とユーザデータ# 2とを用いて、ユーザデータ# 2の誤り訂正を実施する。

10

(3) (2)でユーザデータの誤り訂正に失敗した場合、外部パリティであるパリティ# 3-1とユーザデータ# 3とパリティ# 2-3を用いて、パリティ# 2-3の誤り訂正を行う。パリティ# 2-3の誤り訂正に成功した場合、パリティ# 2-1と誤り訂正後のパリティ# 2-2と誤り訂正後のパリティ# 2-3とユーザデータ# 2とを用いて、ユーザデータ# 2の誤り訂正を実施する。

【0082】

図16は、ユーザデータ1個とパリティ複数個とから外部パリティを計算する場合のユーザデータおよびパリティの格納例を示す図である。図16の例では、ユーザデータ# 2、# 3に関しては、3段階のパリティ生成が行われ、ユーザデータ# 1については1段階のパリティ生成が行われる。ユーザデータ# 2、# 3に関しては、それぞれ1段階目のパリティとして生成多項式 $G_1(x)$ に基づいてパリティ# 2-1、# 3-1が生成され、2段階目のパリティとして生成多項式 $G_2(x)$ に基づいてパリティ# 2-2、# 3-2が生成される。そして、ユーザデータ# 1、パリティ# 2-2および# 3-2に生成多項式 $G_3(x)$ を用いてパリティ# 1-1が生成される。パリティ# 1-1は、パリティ# 2-2および# 3-2の外部パリティとなっている。

20

【0083】

図17~図19は、半導体メモリ部3をNANDメモリとした場合の本実施の形態のユーザデータおよび各パリティの格納方法の例を示している。ここでは、1つのメモリセルに3bitのデータを格納するとし、この3bitにそれぞれ異なるページが割当てられるとし、1ワードラインに格納されるページをlower(L)ページ、middle(M)ページ、upper(U)ページとそれぞれ呼ぶこととするが、半導体メモリ部3を構成するNANDメモリは、このようなタイプのメモリに限定されるものではない。図17~図18では、パリティ専用ページ以外については、lowerページ、middleページ、upperページをそれぞれ異なるハッチングをして示している。例えば、図17~図19において、1行目がlowerページ、2行目がmiddleページ、3行目がupperページ、4行目がlowerページ、...とする。パリティ専用ページについては、lowerページ用のパリティ、middle用ページのパリティ、upperページ用のパリティをそれぞれlowerページ、middleページ、upperページと同一のハッチングで示している。パリティ専用ページは、lowerページ、middleページ、upperページのいずれであってもよいが、図17~図19では、パリティ専用ページは、middleページ(下から2段目)とupperページ(最下段)の例を示している。

30

40

【0084】

図17~図18の例では、パリティの一部を、(ユーザデータが格納されない)パリティ専用ページに格納している。図17の例では、lowerページ、middleページ、upperページの全てに同一サイズのパリティを割当てている。そして、1段階目で生成したパリティ(122B)をそれぞれ対応するユーザデータ(512B×2)と同一ページに格納し、2段階目のパリティ(7B)を、下から2行に示した2ページのパリティ専用ページに格

50

納している。また、最下段のパリティ専用ページでは、2段階目のパリティをユーザデータと同等に扱って外部パリティ(129B)を生成し、各ページ(パリティ専用ページ)内に格納している。

【0085】

図17の例では、lowerページ、middleページ、upperページの全てについて、それぞれページ内に格納された1段階目のパリティ(122B)を用いて69ビットまで訂正可能であり、さらにパリティ専用ページに格納された2段階目のパリティ(7B)を用いると73ビットまで訂正可能である。

【0086】

また、図18の例では、lowerページ、middleページ、upperページの全てに、1段階目のパリティについては同一サイズのパリティを割当てている。そして、1段階目で生成したパリティ(122B)をそれぞれ対応するユーザデータ(512B×2)と同一ページに格納している。2段階目のパリティ(23B)については、middleページに対して生成して下から2行に示した2ページのパリティ専用ページに格納している。また、最下段のパリティ専用ページでは、2段階目のパリティをユーザデータと同等に扱って外部パリティ(145B)を生成し、各ページ(パリティ専用ページ)内に格納している。

【0087】

図18の例では、lowerページ、middleページ、upperページの全てについて、それぞれページ内に格納された1段階目のパリティ(122B)を用いて69ビットまで訂正可能であり、さらにパリティ専用ページに格納された2段階目のパリティ(23B)を用いるとmiddleページについて82ビットまで訂正可能である。

【0088】

また、図19の例では、middleページの1段階目のパリティのサイズをlowerページおよびupperページの1段階目のパリティのサイズより大きくしている。middleページの1段階目のパリティは69ビットまで訂正可能な122Bであり、lowerページおよびupperページの1段階目のパリティは、67ビットまで訂正可能な118Bである。また、2段階目のパリティ(23B)については、middleページに対して生成して下から2行に示した2ページのパリティ専用ページに格納する。さらに3段階目および4段階目のパリティ(各4B)をmiddleページに対して生成して、lowerページおよびupperページに格納している。また、最下段のパリティ専用ページでは、2段階目のパリティをユーザデータと同等に扱って外部パリティ(153B)を生成し、各ページ(パリティ専用ページ)内に格納している。

【0089】

図19の例では、lowerページおよびupperページについては、67ビットまで訂正可能であり、middleページについては、ユーザデータと同一ページ内の1段階目のパリティを用いると69ビットまで訂正可能であり、パリティ専用ページの2段階目のパリティを用いると82ビットまで訂正可能であり、4段階目までの全てのパリティを用いると87ビットまで訂正可能である。

【0090】

図18、図19の例のように、ページごとにパリティのサイズを変えるようにしておくと、例えば、信頼性の低いページに大きなサイズのパリティを割当てることが可能となる。また、実際の誤り数が少ない場合は、ページ内のパリティの読出しだけですむ。さらに、上述したように各段階のパリティを追加して誤り訂正を行うような符号化を行っているため、独立して多段階のパリティを生成する場合に比べ、パリティ量が同一であれば誤り訂正能力を高めることができる。

【0091】

なお、図18、図19の例では、3ページのうち1ページにサイズの大きなパリティを割り当て、残りの2ページには同一サイズのパリティを割り当てたが、3ページすべてに異なるサイズのパリティを割り当ててもよい。

【0092】

10

20

30

40

50

以上のように、本実施の形態では、パリティ保護表を用いて、パリティ# i を保護するパリティを設定可能とすることにより、第1の実施の形態の符号演算および復号演算と、第2の実施の形態の符号演算および復号演算と、を組み合わせ実施することができる。このため、想定される誤り数や演算速度への要求等に応じて適切な符号演算および復号演算を行うことができる。

【0093】

(第4の実施の形態)

図20は、第4の実施の形態にかかる誤り訂正装置を備える通信装置の構成例を示すブロック図である。図20では、誤り訂正処理にかかる構成要素を示しており、例えば通信装置が無線通信装置である場合のアンテナ等、通信装置が通常備える機能は図示を省略している。

10

【0094】

通信装置30-1, 30-2は、それぞれ送信機能および受信機能を有し、通信装置30-1から送信されたユーザデータは通信装置30-2で受信され、通信装置30-2から送信されたユーザデータは通信装置30-1で受信される。通信装置30-1と通信装置30-2の間の通信路において、誤りが発生することがある。

【0095】

このような通信路における誤りに対する誤り訂正にも第1の実施の形態で述べた多段階の符号化を適用することができる。

【0096】

20

通信装置30-1, 30-2は同様の構成を有する。図20に示すように、通信装置30-1, 30-2は、符号化部40と復号部50を備える。符号化部40は、ユーザデータ生成部41、符号演算部42、符号記憶部43、送信部44および符号送信要求受信部45を備える。復号部50は、受信部51、復号演算部52および符号送信要求送信部53を備える。符号化部40および復号部50は、本実施の形態の誤り訂正装置を構成する。

【0097】

まず、ユーザデータの送信動作について説明する。例えば、通信装置30-1から通信装置30-2へユーザデータを送信する場合、通信装置30-1のユーザデータ生成部41は、ユーザデータを生成して符号演算部42へ入力する。符号演算部42は、第1の実施の形態(または第2の実施の形態、第3の実施の形態のいずれかでもよい)と同様の多段階の符号演算によりパリティを生成し、符号記憶部43へパリティを格納する。例えば、パリティ#1~パリティ# n が格納されるとする。送信部44は、ユーザデータと所定の段階までのパリティ(例えば、パリティ#1)とを通信装置30-2へ送信する。なお、第3の実施の形態の符号化を行う場合には、通信装置30-1および通信装置30-2はパリティ保護表を保持する。

30

【0098】

通信装置30-2では、受信部51が、ユーザデータおよびパリティ(例えば、パリティ#1)を受信し、復号演算部52へ入力する。復号演算部52は、第1の実施の形態(または第2の実施の形態、第3の実施の形態のいずれかでもよい)と同様に受信したパリティに応じた処理を実施する。例えば、第1の実施の形態の符号演算が実施されて、パリティ#1を受信した場合、パリティ#1とユーザデータに基づいてユーザデータの誤り訂正を行う。復号演算部52は、この誤り訂正が成功した場合は、当該ユーザデータの受信処理を終了する。一方、この誤り訂正に失敗した場合は、復号演算部52は、誤り訂正の失敗を符号送信要求送信部53へ通知する。符号送信要求送信部53は、この通知を受けると、当該ユーザデータに対する次の段階のパリティ(例えば、パリティ#2)の送信要求を通信装置30-1へ送信する。なお、復号演算部52は、誤り訂正に失敗した場合、受信済みのパリティとユーザデータとを保持しておく。

40

【0099】

通信装置30-1の符号送信要求受信部45は、通信装置30-2からの送信要求を受

50

信すると、対応するパリティを符号記憶部 4 3 から送信部 4 4 へ出力するよう制御し、送信部 4 4 は入力されたパリティを通信装置 3 0 - 1 へ送信する。通信装置 3 0 - 1 の受信部 5 1 は、このパリティを復号演算部 5 2 へ渡し、復号演算部 5 2 は、保持しているパリティとユーザデータと新たに受信したパリティとを用いて復号演算を実施する。以降、復号演算において誤り訂正が失敗した場合には、同様に、次の段階のパリティの送信を要求し、当該パリティを受信した場合に、受信したパリティと保持しているパリティおよびユーザデータとを用いて復号演算を行うことを繰り返す。

【 0 1 0 0 】

例えば、まず、通信装置 3 0 - 1 は、ユーザデータとともに第 1 の実施の形態のパリティ # 1 を送信しておき、パリティの送信要求を受信した場合（すなわち、パリティ # 1 による誤り訂正が失敗であった場合）に、パリティ # 2 を送信し、さらにパリティの送信要求を受信した場合（すなわち、パリティ # 2 までによる誤り訂正が失敗であった場合）に、パリティ # 2 を送信する、... というように段階的にパリティを送信することができる。これにより、誤りの規模に応じて誤り訂正に用いるパリティ数を適応的に変化させることができ、パリティによる通信帯域の浪費を防ぎつつ誤り数が多い場合にも誤り訂正を実施することができる。

10

【 0 1 0 1 】

なお、ここでは、通信装置 3 0 - 1 , 3 0 - 2 がユーザデータを生成するようにしたが、通信装置 3 0 - 1 , 3 0 - 2 がユーザ端末等と接続し、ユーザデータをユーザ端末等から受信し、受信したユーザデータに対して符号演算を行うようにしてもよい。

20

【 0 1 0 2 】

（第 5 の実施の形態）

図 2 1 は、第 5 の実施の形態にかかる半導体記憶装置における復号部の構成例を示すブロック図である。本実施の形態では、第 3 の実施の形態で述べた多段階誤り訂正方式を実現するための復号部 2 6 b の構成と動作の一例を説明する。

【 0 1 0 3 】

図 2 1 に示すように、本実施の形態の復号部 2 6 b は、シンドローム計算部 2 6 1、誤り位置多項式演算部 2 6 2、チエンサーチ部 2 6 3 および復号制御部 2 6 4 を備える。

【 0 1 0 4 】

図 2 2 は、本実施の形態の復号処理手順の一例を示すフローチャートである。図 2 1、2 2 を用いて、本実施の形態の復号処理の全体動作を説明する。制御部 2 3 は、Host I/F 2 1 経由で Host 4 からデータの読み出し要求を受け取ると、読み出し対象のデータを読み出すようメモリ I/F 2 2 へ指示し、メモリ I/F 2 2 は読み出し対象のデータに対応するパリティとともに読み出し、復号部 2 6 へ渡す。復号部 2 6 では、メモリ I/F 2 2 から読み出し対象のデータおよびパリティを受け取ると復号処理を開始する。

30

【 0 1 0 5 】

本実施の形態では、第 3 の実施の形態と同様に多段階誤り訂正方式により符号化演算が実施される。以下の説明では、符号化演算において 2 段階のパリティが生成され、また 2 段階目のパリティに外部パリティが生成される例について説明するが、多段階誤り訂正方式の段階数や外部パリティの付与方法はこれに限定されず、外部パリティを用いなくてもよい。外部パリティを用いない多段階誤り訂正方式を採用する場合も、本実施の形態の復号処理は適用可能であり、この場合は、例えば、各段階のパリティに誤りが無いと仮定して復号処理を実施する。

40

【 0 1 0 6 】

図 2 2 の手順は、多段階の復号演算のうちの 1 段階分の処理を示している。第 1 の実施の形態～第 3 の実施の形態で述べたように、1 段階目のパリティを用いた図 2 2 に示す手順により復号処理を実施し、誤り訂正に失敗した場合、2 段階目のパリティを用いた復号処理が実施される。この際、2 段階目のパリティに外部パリティが付与されている場合は、外部パリティを用いた復号処理を図 2 2 と同様の手順で実施し、復号後（誤り訂正後）の 2 段階目のパリティを用いて 2 段階目の復号処理を実施する。

50

【 0 1 0 7 】

まず、復号制御部 2 6 4 は、復号処理を開始すると、シンドローム計算部 2 6 1 に対してシンドローム計算の実施を指示し、シンドローム計算部 2 6 1 がシンドローム計算を実施する（ステップ S 2 1）。その後、復号制御部 2 6 4 は、誤り位置多項式演算部 2 6 2 に対して、計算済みのシンドローム計算結果を用いて誤り多項式演算を行うよう指示し、誤り位置多項式演算部 2 6 2 が誤り位置多項式演算を実施する（ステップ S 2 2）。

【 0 1 0 8 】

次に、復号制御部 2 6 4 は、チエンサーチ部 2 6 3 に対して、誤り位置多項式演算の結果を用いてチエンサーチを行うよう指示し、チエンサーチ部 2 6 3 は誤り位置多項式演算の結果である誤り位置多項式の係数を用いてチエンサーチを実施し（ステップ S 2 3）、誤り位置を特定し、処理を終了する。復号制御部 2 6 4 は、チエンサーチにより、誤りがあると判定されたビットを反転させることにより誤り訂正を実施する。

【 0 1 0 9 】

1 段階目のパリティを用いた復号処理では、シンドローム計算部 2 6 1 に入力されるのは、ユーザデータ（読み出しデータ）と 1 段階目のパリティである。1 段階目のパリティを用いた復号処理において誤り訂正不可であると判定された場合、本実施の形態では、2 段階目のパリティに外部パリティが付与されるため、外部パリティと 2 段階目のパリティとを用いた復号処理が行われる。そして、ユーザデータと 1 段階目のパリティと誤り訂正後の 2 段階目のパリティとをシンドローム計算部 2 6 1 に入力し、2 段階目の復号処理が実施される。なお、外部パリティと 2 段階目のパリティとを用いた復号処理については、図 2 1 に示したシンドローム計算部 2 6 1、誤り位置多項式演算部 2 6 2、チエンサーチ部 2 6 3 および復号制御部 2 6 4 を用いて実施してもよいし、これらとは別に復号処理演算の回路を設けて、この回路により実施してもよい。

【 0 1 1 0 】

以下、本実施の形態では、第 1 の実施の形態で述べた生成多項式 $G_1(x)$ に基づいて 1 段階目のパリティとしてパリティ # 1 が生成され、生成多項式 $G_2(x)$ に基づいて 2 段階目のパリティとしてパリティ # 2 が生成され、パリティ # 2 の外部パリティが生成されているとする。

【 0 1 1 1 】

パリティ # 1 が t_1 の誤り訂正能力を有し（ t_1 ビットの誤り訂正が可能）、パリティ # 1 とパリティ # 2 の両方を用いた場合の誤り訂正能力を t_2 とする（ t_2 ビットの誤り訂正が可能）。このとき、 t_1 の誤り訂正能力に対応した復号処理回路と t_2 の誤り訂正能力に対応した復号処理回路を個別に備え、段階ごとに使い分けるように構成することも可能であるが、個別に備えるとハードウェア規模が大きくなるため、復号処理回路を共用することが望ましい。本実施の形態では、各段階の復号処理を単一の復号処理回路で実施する場合の構成および動作を説明する。

【 0 1 1 2 】

まず、本実施の形態のシンドローム計算部 2 6 1 について説明する。一般に、シンドローム計算部 2 6 1 は、 i 倍の乗算を行うための乗算器を含む最小単位の計算回路（ i 倍の乗算器、XOR 回路、レジスタ等を含む）を誤り訂正可能なビット数個備える。 i の添字 i は、 $G_1(x) \times G_2(x)$ の根（ α^i をガロア拡大体 $GF(2^m)$ の原始元としたときの根 $\alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \dots, \alpha^{2^{i2}-1}$ ）の指数部に対応する。ここでは、この最小単位の計算回路（シンドローム S_i を算出する回路）をシンドローム S_i 計算回路とよぶ。1 段階目の誤り訂正処理（復号処理）と、2 段階目の誤り訂正処理とで回路を共用するためには、少なくとも、 $i = 1, 3, 5, \dots, 2t_2 - 1$ にそれぞれ対応する合計 t_2 個のシンドローム S_i 計算回路を備えることになる。

【 0 1 1 3 】

図 2 3 は、シンドローム計算部 2 6 1 の構成例を示す図である。図 2 3 では、一例としてパリティ # 1、パリティ # 2 がそれぞれ 2 ビットの訂正能力を有する例を示しているが、パリティ # 1、パリティ # 2 の訂正能力はそれぞれこれらに限定されない。ここでは、

2段階目のパリティ#1およびパリティ#2を用いた誤り訂正処理を実施可能な復号化回路を共有するため、訂正能力の高い2段階目の誤り訂正能力 t_2 (図23の例では $t_2 = 2$)個の回路を備える構成としている。

【0114】

シンドローム S_j 計算回路は、 i を乗算する m ビットのガロア拡大体乗算回路 $62-j$ ($j = (i + 1) / 2$)と、シンドローム S_j の計算途中結果を保持する m ビットのレジスタ(F F (FlipFlop)) $64-j$ と、 m ビットのX O R回路 $62-j$ とを備える。以上は、一般的なシンドローム計算回路と同様である。本実施の形態では、 i が $G_1(x)$ の根であるシンドローム S_j 計算回路がグループ#1を構成し、 i が $G_2(x)$ の根であるシンドローム S_j 計算回路がグループ#2を構成とする。本実施の形態では、グループ#1に属するシンドローム S_j 計算回路については、シンドロームのレジスタ更新に用いる値を、入力データ(I D A T A)とするか0とするかを切り替えるセクタ回路(選択回路) $61-j$ を備え、セクタ回路 $61-j$ を制御する制御回路65をグループ#1内に少なくとも1つ備える。入力データとは、ユーザデータおよびパリティである。図23の例では、制御回路65をグループ#1内に1つとしているが、例えば、セクタ回路ごとに制御回路を備えてもよい。

10

【0115】

図24は、1段階目の誤り訂正処理における回路使用状態を示す図である。図24では、図23に示したシンドローム計算部261により、1段階目のパリティ#1を用いて(1段階目のパリティ#1とユーザデータとを入力として)誤り訂正を行い場合の回路使用状態を示している。1段階目の誤り訂正では、図24に示すように、グループ#1に属するシンドローム S_j 計算回路を用い、グループ#2に属するシンドローム S_j 計算回路は停止されておいてよい。1段階目の誤り訂正では、制御回路65は、セクタ回路 $61-j$ が入力データ(ユーザデータおよびパリティ#1)を選択するよう制御する。なお、図23, 24では、ユーザデータ、パリティ#1、パリティ#2を、それぞれD a t a、P 1、P 2と略して記載している。

20

【0116】

図24に示すように、1段階目の誤り訂正では、パリティ#2は入力されず、パリティ#1の最終ビット位置を0番目と定義して通常のシンドローム計算が実施される。一般に、誤り訂正処理では、多項式表示で高次の項から順にデータがシンドローム計算回路に入力され、最終入力ビットを0番目のビットとしてチェンサーチ処理において誤り位置の特定が実施される。ここでは、ユーザデータ、パリティの順に入力されるとし、パリティの最終ビット(最後に入力されるビット)を0番目とする。1段階目の誤り訂正処理におけるシンドローム計算は、通常の誤り訂正処理と同様である。1段階目の誤り訂正処理におけるシンドローム計算以外の処理も、通常の誤り訂正処理と同様である。

30

【0117】

次に、本実施の形態のシンドローム計算処理について説明する。図25は、本実施の形態のシンドローム計算の処理手順の一例を示すフローチャートである。前述のように、2段階目の誤り訂正処理は、1段階目の誤り訂正処理に失敗した場合に実施されるが、2段階目の誤り訂正処理を実施する際、パリティ#2に外部パリティが付与されている場合は、パリティ#2は外部パリティを用いた誤り訂正が実施済であるとする。

40

【0118】

まず、グループ#1およびグループ#2のシンドローム S_j 計算回路にユーザデータとパリティ#1を高次の項から1ビット入力する(ステップS31)。具体的には、図23の例では、制御回路65が、入力データ(ユーザデータとパリティ)を選択するようセクタ回路 $61-1$ 、 $61-2$ を制御する。これにより、セクタ回路 $61-1 \sim 61-4$ の全てに入力データの1ビットが入力される。

【0119】

次に、制御回路65は、パリティ#1の最終ビットを入力したか否かを判断し(ステップS32)、入力していない場合(ステップS32 No)、ステップS31へ戻り、次

50

の次数の項の入力データを1ビット入力する。パリティ#1の最終ビットを入力したか否かは、例えば、入力済みのビット数とユーザデータ+パリティ#1のビット数との比較により判定することができる。

【0120】

パリティ#1の最終ビットを入力した場合(ステップS32 Yes)、パリティ#2を入力する(すなわち、2段階目の誤り訂正処理である)か否かを判断する(ステップS33)。パリティ#2を入力しない場合(ステップS33 No)、処理を終了する。

【0121】

パリティ#2を入力する場合(ステップS33 Yes)、グループ#1のシンドローム S_i 計算回路には0を1ビット入力し、グループ#2のシンドローム S_i 計算回路にはパリティ#2を高次の項から1ビット入力する(ステップS34)。具体的には、図23の例では、制御回路65が、0を選択するようセクタ回路61-1, 61-2を制御する。

10

【0122】

次に、制御回路65は、パリティ#2の最終ビットを入力したか否かを判断し(ステップS35)、入力していない場合(ステップS35 No)、ステップS34へ戻り次の次数のパリティ#2を1ビット入力する。パリティ#2の最終ビットを入力した場合(ステップS35 Yes)、処理を終了する。

【0123】

以上のようにシンドローム計算を実施することで、2段階目の誤り訂正処理では、パリティ#2を入力し終えたときには、グループ#1のシンドロームを保持するレジスタ64-1, 64-2については、「データとパリティ#1」をパリティ#2の次数分ケタ上げた(ビット位置を高次側にシフトさせた)多項式に i を代入した結果が格納され、グループ#1のシンドロームを保持するレジスタ64-3, 64-4については、「データとパリティ#1とパリティ#2」を並べて表示した多項式に i を代入した結果が格納される。

20

【0124】

このようにすると、グループ#1に入力した多項式と、グループ#2に入力した多項式とは次数が揃うため、「データとパリティ#1とパリティ#2」のパリティ#2の最低次数の位置を、「データとパリティ#1とパリティ#2」の0番目のビットとして扱うことができる。このため、この後は、通常の t_2 重(t_2 ビットまでの誤り訂正処理が可能な)の誤り位置多項式の計算回路と、誤り位置の計算回路(チエンサーチ回路)とがあれば、2段階目の誤り訂正処理が可能である。

30

【0125】

また、本実施の形態では、2段階の誤り訂正方式を採用しているが、 n 段階(n は3以上の整数)の誤り訂正方式を採用する場合は、グループ#1~グループ# n を設け、それぞれが、生成多項式 $G_1(x)$ の根~生成多項式 $G_n(x)$ の根に対応するシンドローム S_i 計算回路を備えるようにすればよい。また、グループ#2~グループ# n は、それぞれセクタ回路および制御回路を備え、ユーザデータと自グループに対応する根に対応するパリティが入力される間はセクタ回路が入力データを選択し、自グループに対応する根に対応しないパリティが入力される間は、セクタ回路が0を選択するようにすればよい。

40

【0126】

次に、誤り位置多項式演算部262およびチエンサーチ部263について述べる。誤り位置多項式演算部262は、シンドローム計算部261により計算されたシンドロームに基づいて誤り位置多項式を計算する。多段階の誤り訂正方式に対応するためには、誤り位置多項式演算部262は、 t_2 重の誤りにまで対応できる構成を備えていなければならない。また誤り位置多項式計算のアルゴリズムの1つである、バーレカンプ・マッシュィのアルゴリズムは、符号の最大訂正ビット数に対応したループ回数の処理を行う。このアルゴリズムを用い、「データとパリティ#1」だけを入力として計算したシンドロームを入力とする1

50

段階目の誤り訂正処理では、 t_1 重の誤りに対応するループ回数で処理を止めれば、処理を早く終わらせることができる。

【0127】

チエンサーチ部263は、誤り位置多項式に基づいて、誤り位置を計算する回路である。チエンサーチ部263については、 t_2 重の誤りにまで対応できる構成を備えた上で、下記に示す2通りの制御方式が考えられる。

【0128】

チエンサーチ部263では、誤り位置多項式演算部262により求められた誤り位置多項式に、誤りロケータを代入した値を順次計算することで、誤りロケータが誤り位置多項式の根となる位置を求める。ここで、データとパリティを合わせた符号語を多項式で表現したものを符号多項式と呼ぶ。符号多項式の最高次数の項がデータの先頭ビットに対応し、最低次数の項がパリティの最終ビットに対応する。符号多項式の最高次数の側の誤りロケータから代入する場合と、最低次数の側の誤りロケータから代入する場合とで、制御の仕方が異なる。

【0129】

符号多項式の最高次数の側の誤りロケータから代入する場合は、パリティ#2を含めるとき(2段階目の誤り訂正処理)と含めないとき(1段階目の誤り訂正処理)とで符号多項式の最高次数が異なるため、チエンサーチ回路に最初に代入するロケータの値が異なる。すなわち、復号対象にパリティ#2を含めるかどうか(2段階目の誤り訂正処理か1段階目の誤り訂正処理か)で、チエンサーチ部263に最初に代入する値を切り替える処理が必要になる。

【0130】

符号多項式の最低次数の側の誤りロケータから代入する場合は、パリティ#2を含めるとき(2段階目の誤り訂正処理)と含めないとき(1段階目の誤り訂正処理)とで符号多項式の最低次数の位置を表す誤りロケータは同じであるため、上記のようなチエンサーチ部263に最初に代入する値の切り替えは必要ない。

【0131】

符号多項式の最高次数の側の誤りロケータから代入しても、最低次数の側の誤りロケータから代入しても、代入した誤りロケータの位置を順にずらしていく方向が、前者であれば次数が下がる方向、後者であれば次数が上がる方向というように異なるだけで、代入した誤りロケータの位置をずらして、誤り位置多項式の根になっているかチェックする、という処理自体は同じである。

【0132】

仮に、グループ#1に属するシンδροーム S_i 計算回路に、入力データ切り替えの機構(セレクタ回路61-1, 61-2, 制御回路65)が無かったとする。そうすると、グループ#1に属するシンδροーム S_i 計算回路にもパリティ#2が入力される。パリティ#1の生成多項式 $G_1(x)$ はグループ#1に対応する根(図23の例では、 x^1, x^3)を持つが、生成多項式 $G_2(x)$ の根は持たない。したがって、グループ#1のシンδροーム S_i 計算回路に、パリティ#2が入力されると、シンδροームにおいて符号語の寄与を排除できず、誤り位置多項式を求めるのが非常に困難になる(シンδροームが誤りパターンにのみ依存して決まるのが誤り位置多項式を効率的に求めるために必要である)。そのため、通常の誤り位置多項式計算回路と、チエンサーチ回路とでは誤り訂正ができない。

【0133】

また、グループ#1に属するシンδροーム S_i 計算回路への入力データを「データとパリティ#1」を入力した時点で打ち切ってしまうと、グループ#1とグループ#2とで入力した多項式の次数が異なるため、誤り位置を示す値が定められず、誤りロケータを定義できない。そのため、誤り位置多項式そのものが定義できず、通常の誤り位置多項式計算回路と、チエンサーチ回路とでは誤り訂正ができないことになる。

【0134】

10

20

30

40

50

また、復号の各段階において、復号に使用しないパリティの生成に用いた生成多項式の根に対応するシンδροーム計算を行う部分は、動作を停止するような制御を行うことで、各段階での誤り訂正処理でシンδροーム計算部 261 を共用しても、消費電力を、各段階での誤り訂正に必要な最低限の回路構成とした場合と同程度に抑えることができる。

【0135】

また、パリティ # 2 を保護する外部パリティを用いたパリティの誤り訂正処理では、パリティ # 1、パリティ # 2 を用いた誤り訂正処理とは別の復号回路を備えてもよいが、復号回路を共用することもできる。

【0136】

具体的には、パリティ # 2 を保護する外部パリティの訂正能力を t_2 ビット以下に設定し、 t_2 重の誤り訂正を行うための回路を備えていれば、外部パリティを用いた誤り訂正処理についても回路を共用できる。特に外部パリティの生成多項式に、生成多項式 $G_1(x)$ を用い、かつ符号長も 1 段階目の符号語 (ユーザデータ + パリティ # 1) と同じになるように揃えた場合、1 段階目のパリティ # 1 を用いたユーザデータの誤り訂正処理と全く同じ処理で復号を行うことができ、制御を簡単化できる。

【0137】

図 26 は、外部パリティを用いた復号も行う場合の復号部 26b の構成例を示す図である。図 26 の decoder 100 は、図 21 のシンδροーム計算部 261、誤り位置多項式演算部 262、チエンサーチ部 263 および復号制御部 264 により構成される。図 26 に示すように、外部パリティを用いた復号と共用する場合、訂正済み拡張パリティ一時保存領域 101 およびセクタ回路 102 をさらに備える。

【0138】

セクタ回路 102 は、制御部 23 または復号制御部 264 の制御により、パリティ # 1 を用いた 1 段階目の誤り訂正処理では、入力データを選択して decoder 100 へ入力する。1 段階目の誤り訂正処理に失敗した場合、セクタ回路 102 は、制御部 23 または復号制御部 264 の制御により、入力データを選択して decoder 100 へ入力する。この際、入力データとしては、パリティ # 2 と外部パリティが入力される。外部パリティを用いたパリティ # 2 の誤り訂正処理を行った後、誤り訂正処理後のパリティ # 2 は、一時的に訂正済み拡張パリティ一時保存領域 101 に格納される。

【0139】

そして、ユーザデータに対する 2 段階目の誤り訂正処理を行う際に、セクタ回路 102 は、制御部 23 または復号制御部 264 の制御により、はじめは入力データ (ユーザデータとパリティ # 1) を選択し、ユーザデータとパリティ # 1 の入力が終わると、訂正済み拡張パリティ一時保存領域 101 に格納されたデータ (誤り訂正済みのパリティ # 2) を選択して decoder 100 へ入力する。

【0140】

以上のように、本実施の形態では、2 段階で誤り訂正符号化を実施する場合に、パリティ # 1 とパリティの両方を用いた場合の訂正能力に対応する復号部 26b を有し、シンδροーム S_i 計算回路を、同一段階の生成多項式の根に対応するシンδροームを計算するシンδροーム S_i 計算回路を同一グループとして段階ごとにグループ分けする。そして、2 段階目の誤り訂正処理では、1 段階目に対応するグループに属するシンδροーム S_i 計算回路には、パリティ # 2 の代わりに 0 を入力するようにした。このため、1 段階目と 2 段階目の誤り訂正処理で回路を共有でき、かつ誤り位置を正しく定義することができる。

【0141】

(第 6 の実施の形態)

図 27 は、第 6 の実施の形態にかかる半導体記憶装置におけるシンδροーム計算部 261a のシンδροーム S_i 計算回路の構成例を示すブロック図である。本実施の形態の半導体記憶装置の構成は、第 5 の実施の形態のシンδροーム計算部 261 を本実施の形態のシンδροーム計算部 261a に替える以外は、第 5 の実施の形態の半導体記憶装置と同様である。以下、第 5 の実施の形態と異なる部分を説明する。

【 0 1 4 2 】

第 1 の実施の形態では、シンドローム計算部 2 6 1 に、1 ビットずつデータを入力する例を示したが、本実施の形態では、シンドローム計算部 2 6 1 a において P (P は 2 以上の整数) ビットの並列処理を行う例について説明する。本実施の形態の符号化演算は、第 5 の実施の形態と同様に 2 段階の誤り訂正方式で実施されているとする。図 2 7 では、P = 2 の場合のグループ # 1 のシンドローム S_i 計算回路の構成例を示している。グループ # 1 のシンドローム S_i 計算回路は、セクタ回路 7 1 - 1 , 7 1 - 2 と、X O R 回路 7 2 - 1 , 7 2 - 2 と、 x^i の乗算を行う乗算器 7 3 - 1 , 7 3 - 2 と、レジスタ 7 4 と、制御回路 7 5 と、を備える。なお、P が 3 以上の場合も、同様に、各シンドローム S_i 計算回路が乗算器および X O R 回路を P 個備え、グループ # 1 のシンドローム S_i 計算回路が、P 個のセクタ回路と制御回路 7 5 とを備えるようにすればよい。

10

【 0 1 4 3 】

本実施の形態のシンドローム計算部 2 6 1 a は、第 5 の実施の形態と同様に、グループ # 2 のシンドローム S_i 計算回路も備えるが、グループ # 2 のシンドローム S_i 計算回路については、従来の P ビット並列処理を行うシンドローム S_i 計算回路と同様である。すなわち、図 2 7 のシンドローム S_i 計算回路からセクタ回路 7 1 - 1 , 7 1 - 2 および制御回路 7 5 を削除したものであるため図示を省略する。

【 0 1 4 4 】

グループ # 1 のシンドローム S_i 計算回路のセクタ回路 7 1 - 1 , 7 1 - 2 は、制御回路 7 5 からの制御により、入力データ (ユーザデータ、パリティ) と 0 のうちいずれか一方を選択してそれぞれ X O R 回路 7 2 - 1 , 7 2 - 2 へ入力する。具体的には、制御回路 7 5 は、ユーザデータとパリティ # 1 が入力される間は入力データを選択し、パリティ # 2 が入力される間は 0 を選択するようセクタ回路 7 1 - 1 , 7 1 - 2 を制御する。制御回路 7 5 は、第 5 の実施の形態の制御回路 6 5 と同様に複数のシンドローム S_i 計算回路で共用してもよい。

20

【 0 1 4 5 】

並列処理を行う場合、シンドローム S_i 計算回路には、1 サイクルで P ビットずつ入力データが入力される。図 2 7 では、P = 2 であり、一例として、入力データの 0 番目のビットと 1 番目のビット (図 2 7 の I D A T A [0] , I D A T A [1]) が同時に入力される例を示している。セクタ回路 7 1 - 1 , 7 1 - 2 による入力データと 0 との切り替え動作以外は、通常の並列処理の動作と同様である。すなわち、P ビット並列処理を行う場合の 1 サイクルでのレジスタ 7 4 の更新値 R E G _ u p d a t e は、C U R を更新前のレジスタ 7 4 の値とするとき、以下の式 (3) で表すことができる。

30

$$\begin{aligned}
 R E G _ u p d a t e = & \\
 & C U R \times (x^i)^P \\
 & + I D A T A [0] \times (x^i)^{P-1} \\
 & + I D A T A [1] \times (x^i)^{P-2} \\
 & \dots \\
 & + I D A T A [P - 2] \times (x^i)^1 \\
 & + I D A T A [P - 1] \times (x^i)^0 \qquad \dots (3)
 \end{aligned}$$

40

【 0 1 4 6 】

本実施の形態のシンドローム計算手順は、図 2 5 に示した処理手順のステップ S 3 1 およびステップ S 3 4 において、1 ビット入力の代わりに P ビット入力とする以外は第 5 の実施の形態の手順と同様である。

【 0 1 4 7 】

以上のようにシンドローム計算において並列処理を行う場合、パリティ # 1、パリティ # 2 のビット数が並列ビット数 P の倍数でない場合、入力データが P ビット未満となるサイクルが生じる。なお、ユーザデータは P ビットの倍数であるとする (P ビットの倍数でない場合も 0 等のデータが付加することによりことにより、最終的に P ビットの倍数であるとする)。一方、パリティ # 1、パリティ # 2 については、それぞれ生成多項式 $G_1 ($

50

x)、 $G_2(x)$ の最大次数により決定される数値であり、 P とは無関係に生成される。

【0148】

入力データが P ビット未満となるサイクルを含む場合、当該サイクルの処理は、他の P ビット並列処理と異なることになり、別のハードウェアを用意する等の対応が必要となる。そこで、本実施の形態では、符号化演算の際に、別のハードウェアを用意する等の対応を実施しなくてよいように、パリティ#1、パリティ#2のビット数をそれぞれ P の倍数となるようゼロパディング(ゼロ挿入処理)を実施する。

【0149】

図28は、本実施の形態のデータフォーマットの一例を示す図である。ここでは、パリティ#1のビット数(生成多項式 $G_1(x)$ の最大次数になる)が P ビットの整数倍になっていないとする。本実施の形態では、 $\text{roundup}(\text{パリティ\#1のビット数}/P) \times P - \text{パリティ\#1のビット数} = d_1$ としたとき、図28に示す位置81に、 d_1 ビットのゼロパディングを行う。この d_1 ビットのゼロは、符号化部25におけるパリティ#1を計算する演算で、情報多項式の下位 d_1 ビットとして入力する。

10

【0150】

パリティ#2についても、同様に、 $\text{roundup}(\text{パリティ\#2のビット数}/P) \times P - \text{パリティ\#2のビット数} = d_2$ としたとき、図28に示す位置82に、 d_2 ビットのゼロパディングを行う。また、 d_2 ビットのゼロについては、符号化部25におけるパリティ生成演算に入力せず、パリティ#2の生成後にゼロを付加する。

【0151】

20

このようにすることで、図28に示すように、パリティ#1、パリティ#2のビット数が P の倍数となり、シンドローム計算において入力データが P ビット未満となるサイクルの発生を防ぐことができる。このため、入力データが P ビット未満となるサイクルに対応する場合に比べ、余分な処理を行う必要がないため、回路規模を抑えることができる。

【0152】

また、パリティ#1のビット数が P の倍数でない場合、2段階目の誤り訂正処理では、パリティ#1とパリティ#2の間に無効なデータが含まれることになり、毎サイクル毎に P 箇所の位置が誤りかどうかチェックする P 並列チェンサーチ回路を使うときに、無効なデータ部分を飛ばした誤りロケータの割り当てを行うことになり、 P ビットにアラインしたフォーマットに合わせて誤り位置を出力するためにはその部分をずらして誤り位置を出す処理が必要になる。図28に示すように、 d_1 ビットのゼロパディングを行うことにより、このような処理は不要になり、回路規模を抑えることができる。

30

【0153】

また、 d_1 ビットのゼロを符号化部25におけるパリティ#1の生成処理に入力せずに、パリティ#1を生成した後にゼロを付加してもよい。この場合、 d_1 ビットのゼロの位置81をパリティ#1の最後とする。図29は、 d_1 ビットのゼロパディングの位置81をパリティ#1の最後の部分とした場合のデータフォーマットの一例を示す図である。位置81は、パリティ#1の始めではなく最後としている。同様に、 d_2 ビットのゼロを符号化部25におけるパリティ#2の生成処理に入力せずに、パリティ#2を生成した後に d_2 ビットのゼロを付加するようにしてもよい。 d_2 ビットのパディングの位置82もパリティ#2内の最後とする。

40

【0154】

なお、本実施の形態では、符号化部25bにおけるパリティ演算にゼロを挿入してパリティのビット数を P の倍数とする処理と、符号化部25bにおいてパリティ(ビット数は P の倍数でない)を生成した後、パリティのビット数を P の倍数となるようゼロを挿入する処理と、の両方をゼロパディングと呼ぶこととする。

【0155】

図30は、3段階の誤り訂正方式を採用する場合のデータフォーマットの一例を示す図である。パリティ#1の始めの部分の位置81に d_1 ビットのゼロをパディングし、パリティ#2の始めの部分の位置82に d_2 ビットのゼロをパディングする。具体的には、図

50

30の例では、図28の例と同様に、 d_1 ビットのゼロをパリティ#1の生成時に情報多項式の下位ビットとして入力する。また、 d_2 ビットのゼロをパリティ#2の生成時に情報多項式の下位ビット(パリティ#1の下位ビット)として入力する。なお、ここではパリティ#3のビット数はPの倍数であるとしてゼロパディングを行っていないが、パリティ#3のビット数がPの倍数でない場合は、パリティ#3の生成後にroundup(パリティ#3のビット数/P)×P-パリティ#3のビット数= d_3 ビットをパリティ#3の始めまたは最後に付加する。また、 d_1 ビットのゼロ、 d_2 ビットのゼロについても、パリティ#1、パリティ#2のそれぞれの生成時に入力とせず、パリティ#1、パリティ#2の生成後にゼロを付加するようにしてもよい。この場合、位置81、位置82はパリティ#1、パリティ#2の最後の部分とする。

10

【0156】

図31は、n段階の誤り訂正方式において復号時のPビット並列処理に対応したパリティ#nを生成するための符号化部25bの構成例を示す図である。ここでは、パリティ#nの生成時に、 d_n ビット($d_n = \text{roundup}(\text{パリティ\#nのビット数}/P) \times P - \text{パリティ\#nのビット数}$)を情報多項式の下位ビットとして入力する場合に構成例を示している。図31に示す符号化部25は、入力データに対して生成多項式 $G_n(x)$ を用いた剰余計算を行うP並列剰余計算回路91と、パリティ#nの計算途中結果を保持するレジスタ(FF)93と、レジスタ93への入力を切り替えるセクタ回路95と、セクタ回路95の切り替えを制御するセクタ回路制御回路92と、 d_n ビットのゼロに対して生成多項式 $G_n(x)$ を用いた剰余計算を行う d_n ビットパディング処理部94と、を備える。

20

【0157】

P並列剰余計算回路91に対する(すなわち、符号化部25に対する)入力データ(情報ビット)は、ユーザデータ+パリティ#1+...+パリティ#(n-1)となる(ただし、パリティ#1を計算する際は、入力データはユーザデータのみである)。符号化部25へ入力データが入力されている間は、セクタ回路制御回路92は、 $G_n(x)$ によるP並列剰余計算回路91の出力をレジスタ93へ入力するようセクタ回路95を制御する。入力が完了すると、セクタ回路制御回路92は、 d_n ビットパディング処理部94回路の出力をレジスタ93へ入力するようセクタ回路95を制御する。

【0158】

d_n ビットパディング処理部94回路は、例えば、 d_n' (d_n' は d_n の約数)並列の $G_n(x)$ による剰余回路構成でされ、入力をゼロに固定し、 $d_n \div d_n'$ サイクル分、この回路の出力でレジスタ93を更新すると、符号化処理は完了し、パリティ#nを出力することができる。

30

【0159】

なお、以上では、外部パリティを用いる場合について説明を省略したが、外部パリティを用いたパリティ#i($i = 1, 2, \dots, n$)の復号処理に、別の回路を用いる場合は外部パリティを用いたパリティ#iの復号処理は通常の場合と同様である。第1の実施の形態で述べたように、外部パリティを用いたパリティ#iと、パリティ#1等を用いたユーザデータの復号処理とで回路を共有する場合は、外部パリティを用いたパリティ#iの復号処理においてもP並列演算が実施される。このため、外部パリティについてもビット数がPの倍数でない場合は、Pの倍数となるようゼロパディングを行うことが望ましい。

40

【0160】

以上のように、本実施の形態では、復号演算においてPビット並列処理を実施する場合に、符号化演算において各段階のパリティのビット数がPの倍数でない場合、各段階のパリティがPの倍数となるようゼロパディングを行うようにした。このため、復号時に、パリティがPの倍数でないことにより生じる「余分な処理を行う必要がないため、回路規模を抑えることができる。

【0161】

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

50

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。

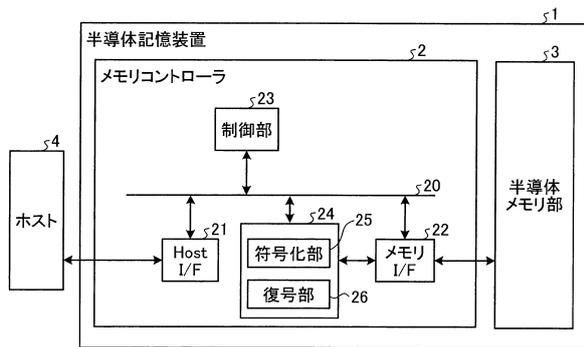
【符号の説明】

【0162】

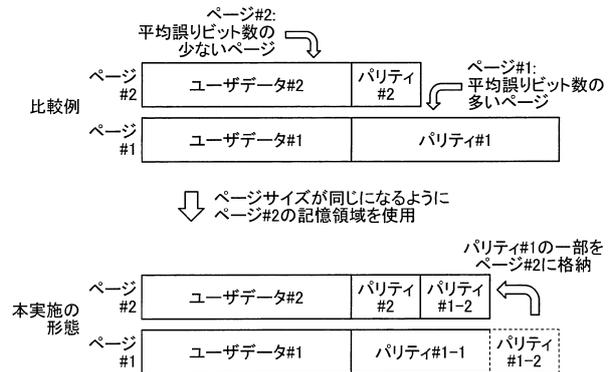
1 半導体記憶装置、2 メモリコントローラ、3 半導体メモリ部、22 メモリI/F、23 制御部、24, 24a, 24b 符号/復号部、25, 25a, 25b, 40 符号化部、26, 26a, 26b, 50 復号部、61-1, 61-2, 71-1, 71-2 セレクタ回路、65, 75 制御回路、261 シンドローム計算部、262 誤り位置多項式演算部、263 チェンサーチ部、264 復号制御部。

10

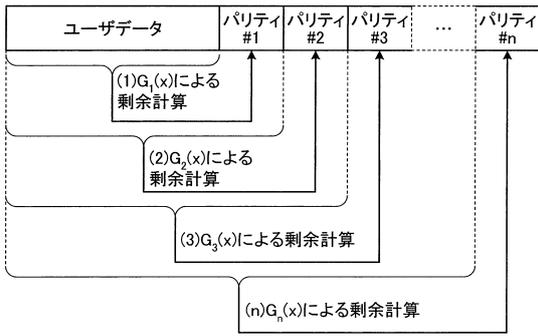
【図1】



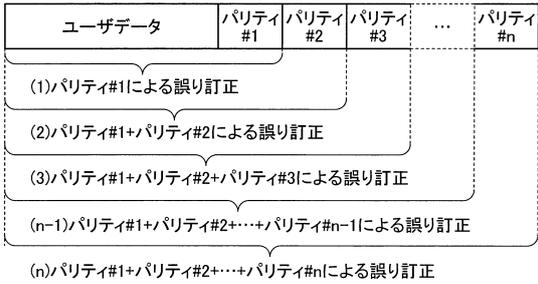
【図2】



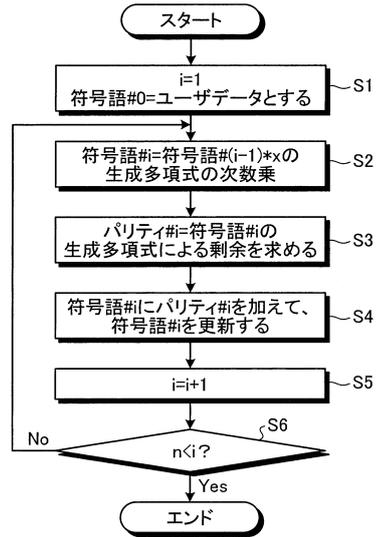
【図3】



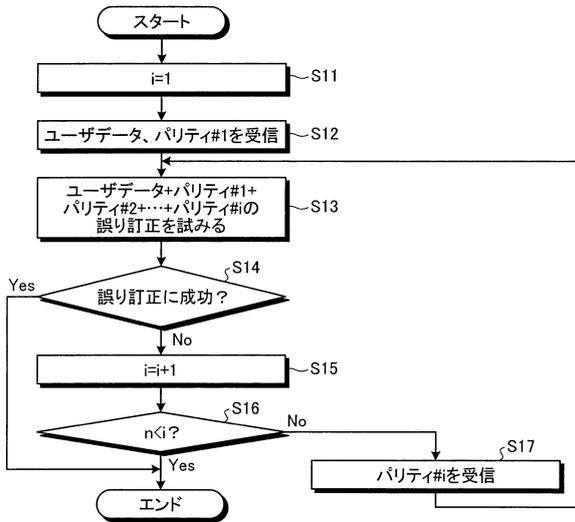
【図4】



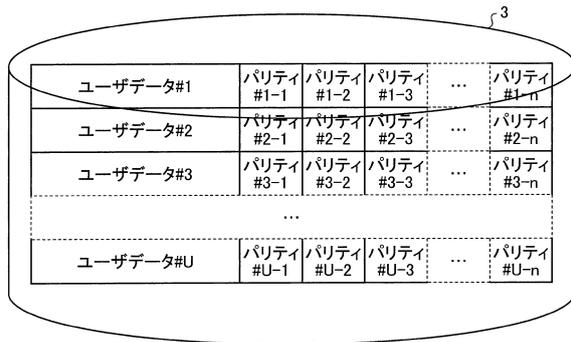
【図5】



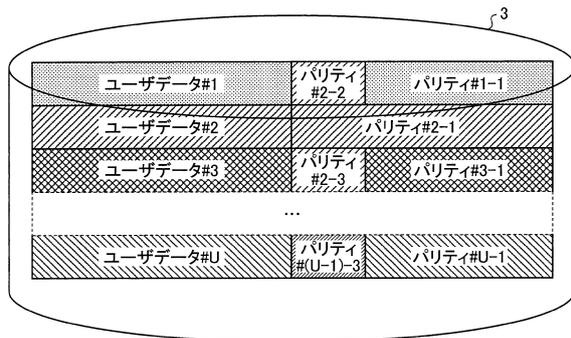
【図6】



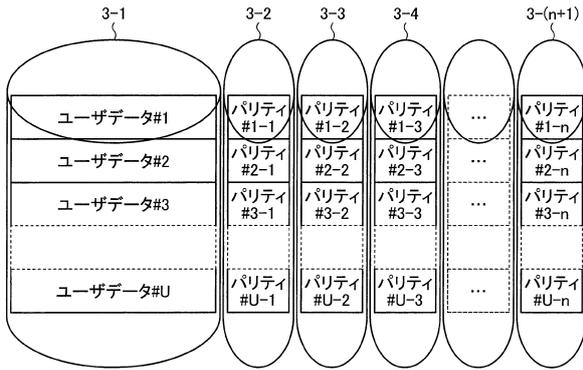
【図7】



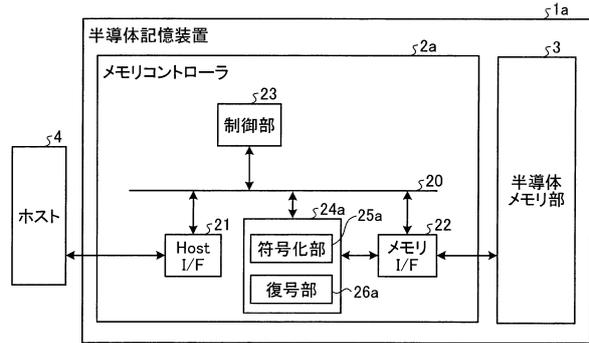
【図8】



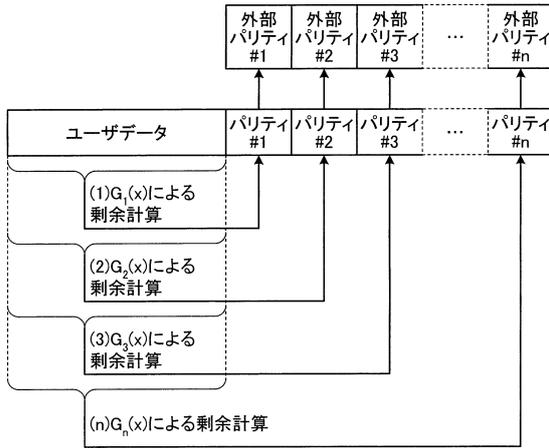
【図9】



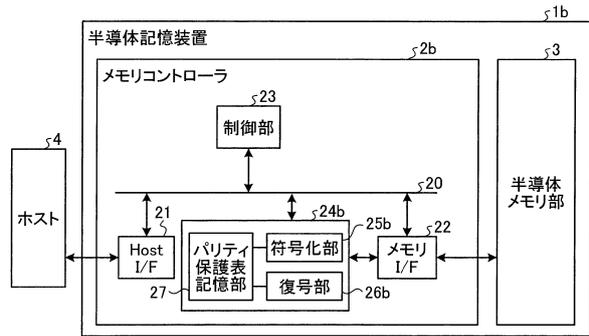
【図10】



【図11】



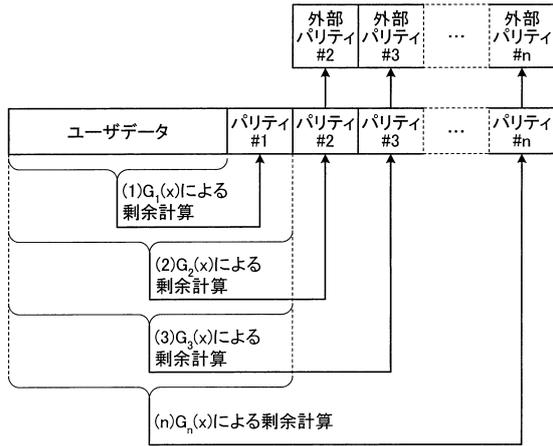
【図12】



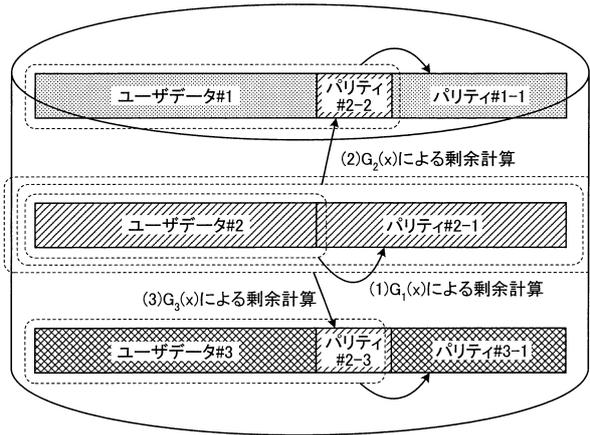
【図13】

	外部 パリティ	パリティ #1	パリティ #2	パリティ #3	...	パリティ #n
パリティ#1	×	-	○	○	...	○
パリティ#2	○	-	-	×	...	×
パリティ#3	○	-	-	-	...	×
...
パリティ#n	○	-	-	-	...	-

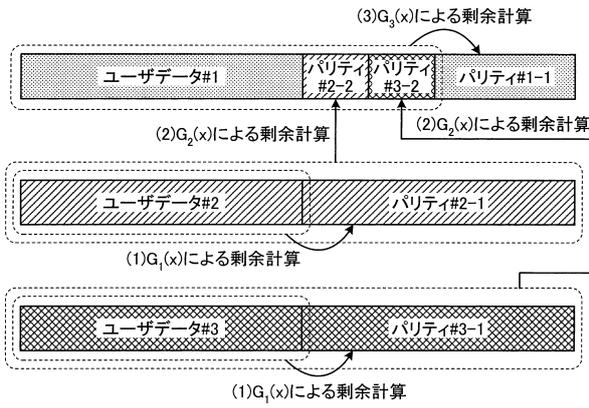
【図14】



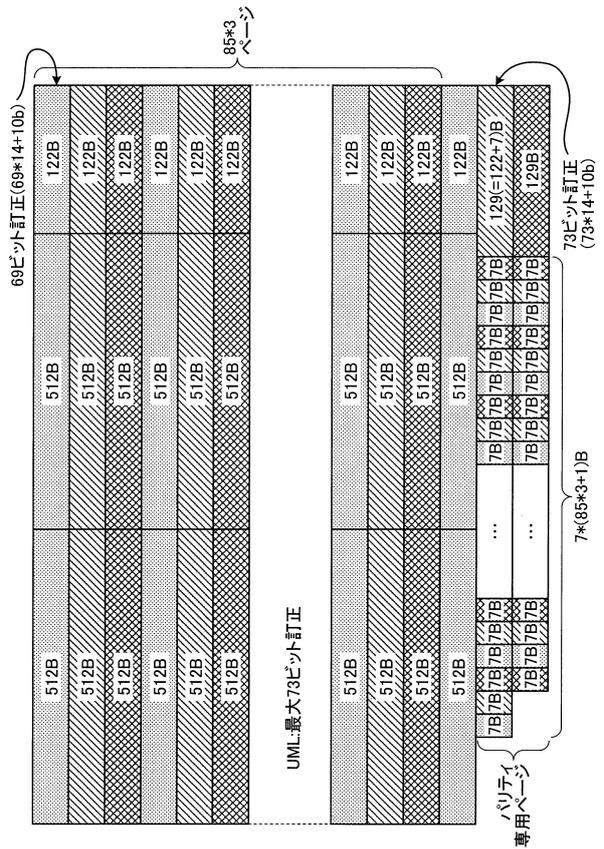
【図15】



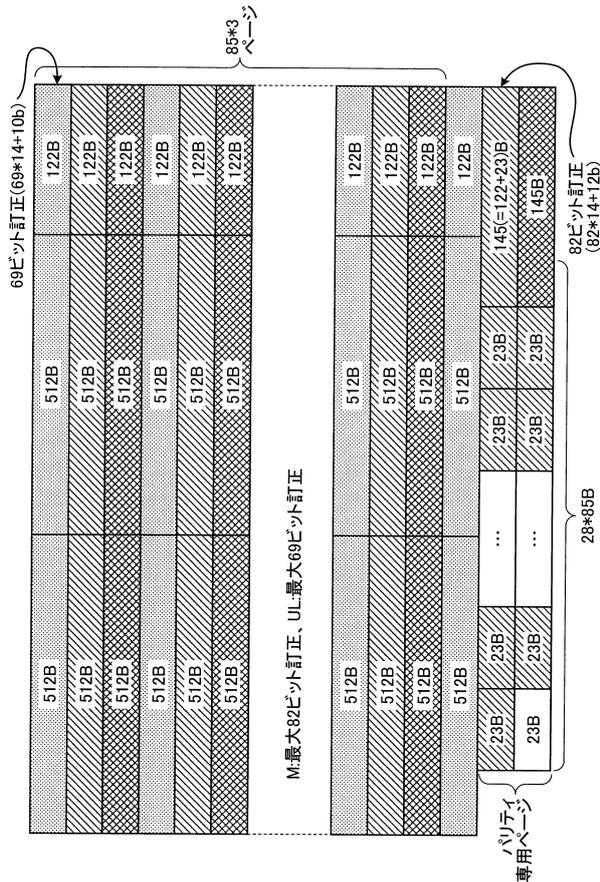
【図16】



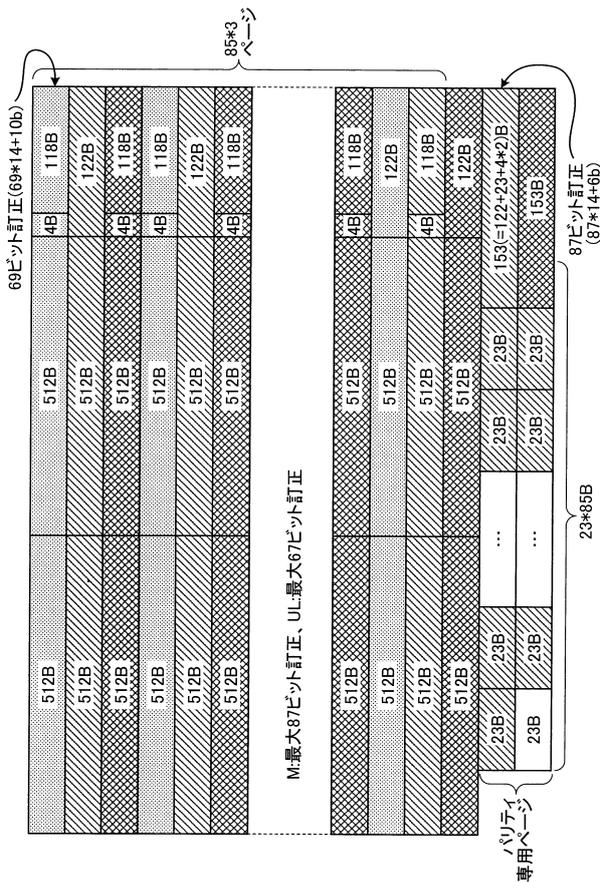
【図17】



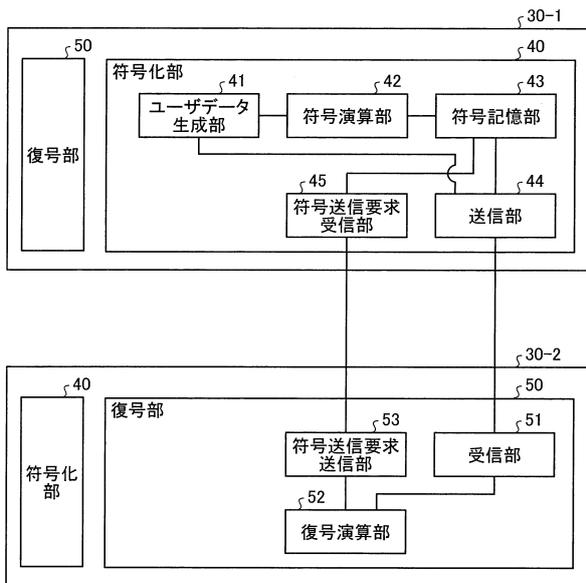
【図18】



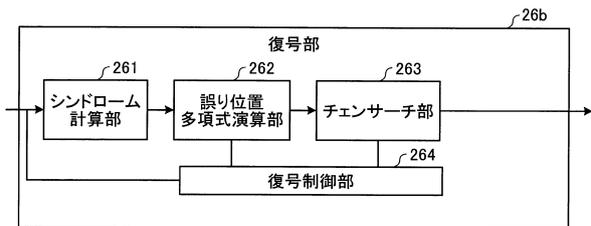
【図19】



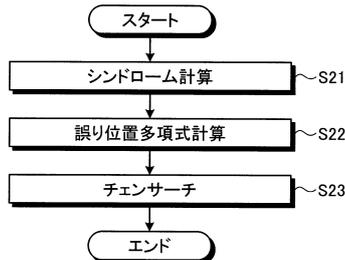
【図20】



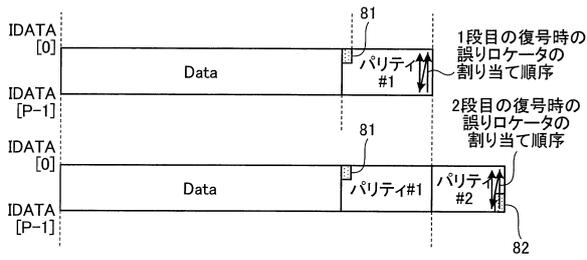
【図21】



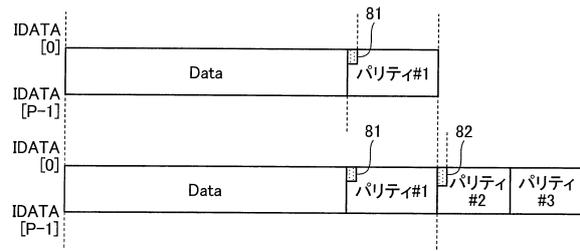
【図22】



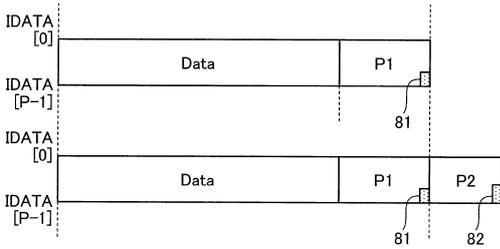
【図28】



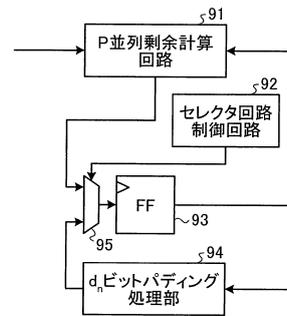
【図30】



【図29】



【図31】



フロントページの続き

審査官 岡 裕之

- (56)参考文献 特開2005-184780(JP,A)
特開平9-245442(JP,A)
特開2008-299922(JP,A)
特開2006-295510(JP,A)
辻岡 哲夫 他, Rate-Compatible LDPC符号のレート推定に基づくHybrid-ARQ方式と適応制御,
電子情報通信学会技術研究報告, 2008年 1月21日, Vol.107, No.463, pp.79-83, CS200
7-68

- (58)調査した分野(Int.Cl., DB名)
H03M 13/29
G06F 12/16
Cinii
IEEE Xplore