

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5112468号
(P5112468)

(45) 発行日 平成25年1月9日(2013.1.9)

(24) 登録日 平成24年10月19日(2012.10.19)

(51) Int. Cl. F 1
HO3M 13/19 (2006.01) HO3M 13/19
GO6F 11/10 (2006.01) GO6F 11/10 330H

請求項の数 5 (全 21 頁)

(21) 出願番号	特願2010-73411 (P2010-73411)	(73) 特許権者	000003078 株式会社東芝
(22) 出願日	平成22年3月26日 (2010.3.26)		東京都港区芝浦一丁目1番1号
(65) 公開番号	特開2011-205578 (P2011-205578A)	(74) 代理人	100076233 弁理士 伊藤 進
(43) 公開日	平成23年10月13日 (2011.10.13)	(72) 発明者	坂上 健二 神奈川県横浜市栄区笠間2-5-1 ST Eビル 東芝メモリシステムズ株式会社内
審査請求日	平成24年2月8日 (2012.2.8)	(72) 発明者	石川 達之 神奈川県川崎市川崎区駅前本町25番地1 東芝マイクロエレクトロニクス株式会社 内
		(72) 発明者	内川 浩典 東京都港区芝浦一丁目1番1号 株式会社 東芝内

最終頁に続く

(54) 【発明の名称】 誤り検出訂正回路、メモリコントローラ、および半導体メモリ装置

(57) 【特許請求の範囲】

【請求項1】

低密度パリティ検査符号により符号化された受信データをサムプロダクト・アルゴリズムを用いて復号処理する誤り検出訂正回路であって、

サイズ p (p は 8 以上の整数) の単位行列からなるブロックと、前記単位行列の各行が「1」から「 $p - 1$ 」のシフト値に対応してシフトしている複数のブロックと、からなる検査行列をもとに、前記受信データを、 p 個のデータからなるグループに分割し出力する第1のセクタと、

それぞれの前記グループを、さらに Y 個 (Y は 2 以上の整数) のサブグループに分割し出力する第2のセクタと、

確率 を算出処理するための複数の第1の変数を、第1のアドレスと関連付けて格納する第1のメモリと、

外部値 を算出処理するための複数の第2の変数を、第2のアドレスと関連付けて格納する第2のメモリと、

前記第2のメモリに格納された前記複数の第2の変数に対して、前記シフト値と前記サブグループとに応じたローテイト値のローテイト処理を行い、演算ユニットに送信するローテイターと、

前記シフト値に連動して前記サブグループ単位で、前記第1の変数と前記第2の変数とを用いた並列演算処理を行う (p / Y) 個の演算回路からなる前記演算ユニットと、

前記シフト値およびローテイト値に応じて、前記第1のセクタ、前記第2のセクタ

、および前記ローテーターを制御する制御部と、を具備する。

【請求項 2】

前記制御部が、前記サブグループへの分割を、前記第 1 のアドレスおよび前記第 2 のアドレスの下位アドレスをもとに制御することを特徴とする請求項 1 に記載の誤り検出訂正回路。

【請求項 3】

前記制御部が、前記第 1 のアドレスおよび前記第 2 のアドレスの最下位アドレス 1 ビットをもとに、それぞれの前記グループを、2 個の前記サブグループに分割するように制御することを特徴とする請求項 1 に記載の誤り検出訂正回路。

【請求項 4】

請求項 1 に記載の誤り検出訂正回路を有することを特徴とするメモリコントローラ。

【請求項 5】

請求項 4 に記載のメモリコントローラと、前記メモリコントローラによってデータの読出し・書き込みが行われる NAND 型フラッシュメモリ部とを有し、前記 NAND 型フラッシュメモリ部からのデータの読出し時に前記誤り検出訂正回路により訂正されたデータを読み出すことを特徴とする半導体メモリ装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、LDPC 符号化されたデータを復号処理する誤り検出訂正回路、前記誤り検出訂正回路を有するメモリコントローラ、および前記誤り検出訂正回路を有する半導体メモリ装置に関し、特に部分並列処理を行う誤り検出訂正回路、前記誤り検出訂正回路を有するメモリコントローラ、および前記誤り検出訂正回路を有する半導体メモリ装置に関する。

【背景技術】

【0002】

半導体メモリ等のストレージ分野での高密度記録等のために、デジタルデータの誤り訂正符号に関する開発が行われている。誤り訂正符号は、代数系の誤り訂正方式と確率にもとづく反復計算による誤り訂正方式とに大別できる。そして、後者に属する低密度パリティ検査符号 (Low Density Parity Check codes、以下、「LDPC 符号」という) はシャノン限界に迫る、優れた性能が報告されている。

【0003】

LDPC 符号化データはサムプロダクト (Sum-Product) アルゴリズムを用いることにより、比較的少ない回路規模で並列処理を行うことができる。しかし、符号長の長い LDPC 符号化データでは、全ての処理を並列で行う完全並列処理は、多数の演算回路を全て実装する必要があるため現実的ではない。

【0004】

このため、例えば、特開 2006 - 279396 号公報には、正方行列を並べた構成の検査行列を用いて部分並列処理を行う復号装置が開示されている。

【0005】

しかし、特開 2006 - 279396 号公報に開示された部分並列処理であっても、正方行列のサイズが大きくなると、演算回路を実装することは容易ではなく、かつ消費電力も大きくなる。

【0006】

このため、処理速度、実装効率および消費電力のバランスのとれた処理効率のよい誤り検出訂正回路、前記誤り検出訂正回路を有するメモリコントローラ、および前記誤り検出訂正回路を有する半導体メモリ装置が求められていた。

【先行技術文献】

【特許文献】

【0007】

10

20

30

40

50

【特許文献1】特開2006-279396号公報

【発明の概要】

【発明が解決しようとする課題】

【0008】

本発明は処理効率のよい誤り検出訂正回路、前記誤り検出訂正回路を有するメモリコントローラ、および前記誤り検出訂正回路を有する半導体メモリ装置を提供することを目的とする。

【課題を解決するための手段】

【0009】

上記目的を達成すべく、本発明の一態様の、低密度パリティ検査符号により符号化された受信データをサムプロダクト・アルゴリズムを用いて復号処理する誤り検出訂正回路であって、サイズ p (p は8以上の整数)の単位行列からなるブロックと、単位行列の各行が「1」から「 $p-1$ 」のシフト値に対応してシフトしている複数のブロックと、からなる検査行列をもとに、受信データを、 p 個のデータからなるグループに分割し出力する第1のセレクタと、それぞれのグループを、さらに Y 個 (Y は2以上の整数)のサブグループに分割し出力する第2のセレクタと、確率を算出処理するための複数の第1の変数を、第1のアドレスと関連付けて格納する第1のメモリと、外部値を算出処理するための複数の第2の変数を、第2のアドレスと関連付けて格納する第2のメモリと、第2のメモリに格納された複数の第2の変数に対して、シフト値とサブグループとに応じたローテイト値のローテイト処理を行い、演算ユニットに送信するローテイターと、シフト値に連動してサブグループ単位で、第1の変数と第2の変数とを用いた並列演算処理を行う (p/Y) 個の演算回路からなる演算ユニットと、シフト値およびローテイト値に応じて、第1のセレクタ、第2のセレクタ、およびローテイターを制御する制御部と、を具備する。

【発明の効果】

【0010】

本発明によれば、処理効率のよい誤り検出訂正回路、前記誤り検出訂正回路を有するメモリコントローラ、および前記誤り検出訂正回路を有する半導体メモリ装置を提供することができる。

【図面の簡単な説明】

【0011】

【図1】検査行列 H_1 を説明するための説明図である。

【図2】タナグラフ G_1 を説明するための説明図である。

【図3】タナグラフ G_2 の場合の(A)行処理および(B)列処理を説明するための説明図である。

【図4】(A)は部分並列処理に用いる検査行列 H_2 であり、(B)はブロック毎のシフト値を示している。

【図5】検査行列 H_3 のブロックのシフトを説明するための図であり、(A)はシフト値=0、(B)はシフト値=1のブロックを示している。

【図6】公知の誤り検出訂正回路の処理を説明するための説明図である。

【図7】公知の誤り検出訂正回路のシフト処理を説明するための説明図である。

【図8】第1の実施の形態のメモリカードの構成を示す構成図である。

【図9】第1の実施の形態の誤り検出部の構成を示す構成図である。

【図10】第1の形態の誤り検出訂正回路の処理の流れを説明するためのフローチャートである。

【図11】第1の形態の誤り検出訂正回路の処理を説明するための説明図である。

【図12】第1の形態の誤り検出訂正回路の処理を説明するための説明図である。

【図13】第1の形態の誤り検出訂正回路の処理を説明するための説明図である。

【図14】第1の形態の誤り検出訂正回路のシフト処理を説明するための説明図である。

【図15】第2の形態の誤り検出訂正回路の処理を説明するための説明図である。

【図16】第2の形態の誤り検出訂正回路の処理を説明するための説明図である。

10

20

30

40

50

【図 17】第 2 の形態の誤り検出訂正回路の処理を説明するための説明図である。
 【図 18】第 2 の形態の誤り検出訂正回路のシフト処理を説明するための説明図である。
 【図 19】第 3 の形態の誤り検出訂正回路の処理を説明するための説明図である。
 【図 20】第 3 の形態の誤り検出訂正回路の処理を説明するための説明図である。
 【図 21】第 3 の形態の誤り検出訂正回路の処理を説明するための説明図である。
 【図 22】第 3 の形態の誤り検出訂正回路のシフト処理を説明するための説明図である。
 【発明を実施するための形態】

【0012】

< LDPC 符号および部分並列処理 >

最初に LDPC 符号および部分並列処理について説明する。LDPC 符号は、非常に疎な検査行列、すなわち、行列内の非零要素数が少ない検査行列により定義される線形符号であり、タナーグラフで示すことができる。そして、誤り訂正処理は、タナーグラフ上で接続された符号語の各ビットに対応したビットノード（ビットノード）と各パリティ検査式に対応したチェックノードとの間で局所的に推論した結果を、やりとりしながら更新していくことに相当する。

10

【0013】

図 1 に、(6, 2) LDPC 符号における、行重み $w_r = 3$ 、列重み $w_c = 2$ の検査行列 H_1 を示す。なお、(6, 2) LDPC 符号とは、符号長が 6 ビット、情報長が 2 ビットの LDPC 符号のことである。

【0014】

20

そして、図 2 に示すように、検査行列 H_1 をタナーグラフ G_1 で示すと、ビットノードは検査行列 H の列に対応し、チェックノードは検査行列 H の行に対応している。そして検査行列 H_1 の要素の中で、「1」であるノード間をエッジで結ぶことによりタナーグラフ G_1 が構成されている。例えば、検査行列 H_1 の第 2 行第 5 列の、丸で囲った「1」は、タナーグラフ G_1 の太線で示すエッジに対応している。そして、検査行列 H_1 の行重み $w_r = 3$ は、ひとつのチェックノードと接続されているビットノードの数、すなわちエッジ数 3 に対応し、検査行列 H の列重み $w_c = 2$ は、ひとつのビットノードと接続されているチェックノードの数、すなわちエッジ数 2 に対応している。

【0015】

LDPC 符号の復号処理はタナーグラフのエッジに割り当てられた信頼度（確率）情報をノードにおいて反復的に更新することで実行される。信頼度情報には、チェックノードからビットノードへの確率情報（以下、「外部値」または「外部情報」ともいい、記号「 λ 」で表示する）と、ビットノードからチェックノードへの確率情報（以下、「事前確率」、「事後確率」、単に「確率」、または「対数尤度比（LLR）」ともいい、記号「 μ 」または「 ν 」で表示する）の 2 種類がある。信頼度更新処理は、行処理と列処理とからなり、行処理と列処理とを 1 回ずつ実行した単位のことを 1 イタレーション（ラウンド）処理と呼び、復号処理はイタレーション処理を繰り返す反復処理により行われる。

30

【0016】

以上の説明のように、外部値 λ とは、LDPC 復号処理のときの、チェックノードからビットノードへの確率情報であり、確率 μ とはビットノードからチェックノードへの確率情報であり、当業者にとり周知の用語である。

40

【0017】

以下、図 3 (A) ~ 図 3 (D) に示すタナーグラフ G_2 の一部を用いて、行処理および列処理について説明する。最初に、図 3 (A) および図 3 (B) を用いて、ビットノード処理（列処理）について説明する。あるビットノード、例えばビットノード 3、に対応する入力された符号語ビットの初期信頼度である LLR、を μ_j (3) とし、このビットノード 3 へのチェックノードからの確率を、 $\lambda_{(j, 3)}$ とする。ここで、 j はビットノード 3 に接続しているチェックノード番号を示す。

【0018】

そして、ビットノード 3 は、 $(1, 3)$ に対応するエッジのチェックノードすなわち

50

、チェックノード1に対して、以下の(式1)で表される計算を行う。

【0019】

$$(3, 1) = (3) + (2, 3) \cdots (式1)$$

【0020】

同様に、ノード番号jのチェックノードに対して、以下の(式2)で表される計算を行う。

【0021】

$$(3, j) = (3) + (k, 3) \cdots (式2)$$

【0022】

ここで、はビットノード3に接続しているチェックノードの中で、 $k = j$ 以外の総和を意味する。

10

【0023】

以上の計算を全てのビットノードに対して行い、以下の(式3)で表される (i, j) を計算する。なお、ここで、符号長をN、ノード番号を、 i とすると、 $i = 1 \sim N$ である。また、はビットノードiに接続しているチェックノードの中で、 $k = j$ 以外の総和を意味する。

【0024】

$$(i, j) = (i) + (k, i) \cdots (式3)$$

【0025】

次に、図3(C)および図3(D)を用いて、チェックノード処理(行処理)について説明する。あるチェックノード、例えばチェックノード1、へのメッセージである外部情報を、 $(k, 1)$ としたとき、このチェックノードは $(1, 1)$ に対応するエッジのビットノードであるビットノード1に対して以下の(式4)で表される $(1, 1)$ を計算する。

20

【0026】

$$(1, 1) = \text{sign}((m, 1)) \times \min(|(m, 1)|) \cdots (式4)$$

【0027】

ただし、 k はこのチェックノード1に接続されているビットノード番号である。そして、 m は「2」から「3」の中から選択される。ここで、 $\text{sign}((m, 1))$ は、 $(m, 1)$ を、 $m = 2$ から3、まで乗算した結果の符号(「+1」あるいは「-1」)を意味する。また、 $| (m, 1) |$ は、 $(m, 1)$ の絶対値であり、 \min は複数の $| (m, 1) |$ の中から最小値を選択する関数である。

30

【0028】

同様にして、 $(1, i)$ についても、以下の(式5)を使って算出する。

【0029】

$$(1, i) = \text{sign}((m, 1)) \times \min\{|(m, 1)|\} \cdots (式5)$$

【0030】

ただし、 i はチェックノード1に接続しているビットノード番号であり、図1(B)のタナーグラフの例では、「1」、「2」、「3」である。また、 m はチェックノード1に接続したビットノードの中で、 $m = i$ 、以外とする。

40

【0031】

以上の計算を全てのチェックノードに対して行い、 (j, i) を以下の(式6)を使って計算する。

【0032】

$$(j, i) = \text{sign}((m, j)) \times \min(|(m, j)|) \cdots (式6)$$

ただし、 m はチェックノードjに接続したビットノードの中で、 $m = i$ 、以外とする。

【0033】

50

そして、反復復号では、以上に示したビットノード処理とチェックノード処理とを1回ずつ実行した1イタレーション処理(1ラウンド)毎に、以下の(式7)によって、事後確率 $P_b(i)$ を求める。

【0034】

$$P_b(i) = (i) + (k, i) \cdots (式7)$$

ここで、 $i = 1 \sim N$ であり、 N は符号長である。そして、 (i) はビットノード i に接続された全ての和である。

【0035】

この事後確率値(一時推定語) P_b を基にビット判定、すなわち、当該ビットが、「0」または「1」のいずれであるかが硬判定される。そして、この硬判定結果を使用して、LDPC符号のパリティチェックを行い、誤りないことが確認された時点で反復処理を終了する。以上が、LDPC符号の反復復号方法である。

10

【0036】

そして既に説明したように、メッセージ伝達アルゴリズムとして、サムプロダクト・アルゴリズムを用いた、ビットノードおよびチェックノードにおける信頼度更新アルゴリズムでは、並列処理による復号処理を行うことができる。

【0037】

そして、部分並列処理には、特開2006-279396号公報に開示された方法以外にも、いくつかの方法がある。例えば、複数の単位行列(以下「ブロック」ともいう。)を組み合わせて構成した検査行列を用い、ブロックサイズのビットノード分の演算回路により並列処理を行う方法がある。

20

【0038】

図4(A)に複数の単位行列を組み合わせて構成した検査行列 H_3 を示す。検査行列 H_3 は、1区画が 5×5 個の要素からなるブロックを横に6個、縦に3個並べて、縦15行、横30列となっている。

【0039】

そして、図4(B)に示すように、検査行列 H_3 の各ブロック B は、対角成分には1が並び、他は全て0となる単位行列を数値分だけシフト処理した正方行列(以下、シフト行列)である。なお、図4(A)に示した検査行列 H_3 は、ユーザーデータ用のブロックである符号化対象(メッセージ)ブロック部 H_3A と、ユーザーデータから生成したパリティ用のパリティ用ブロック部 H_3B とから構成されている。そして、図3(B)に示すように、シフト値「0」は単位行列を示し、シフト値「-1」は0行列を示している。なお、0行列は実際の演算処理が不要なため、以下では除いて説明する。

30

【0040】

シフト処理によってあふれたビットはブロック内の最左列に挿入されている。検査行列 H_3 を用いた復号処理ではシフト値を指定することにより必要なブロックの情報、すなわち、処理すべきノードの情報を得ることができる。なお、 5×5 のブロックからなる検査行列 H_3 においては、シフト値は、復号処理に直接は関係のない、0行列を除くと、0、1、2、3、4の、5種類のいずれかである。

【0041】

40

図4(A)に示したブロックサイズ 5×5 (以下、「ブロックサイズ5」という)の正方行列を組み合わせた検査行列 H_3 を用いた場合には、演算ユニットを5個設けることで、5個のチェックノードに対して部分並列処理を実行することができる。なお、部分並列処理を実行するためには、ビットノード単位での事前/事後確率 P_b を求めるための変数(以下、「LMEM変数」ともいう)を格納するビットノード記憶部(LMEM)と、チェックノード単位で外部値 P_b を求めるための変数(以下、「TMEM変数」ともいう)を格納するチェックノード記憶部(TMEM)が必要である。ビットノードは列方向のアドレス(列アドレス)で管理するのでLMEMは列アドレスで管理し、チェックノードは行方向のアドレス(行アドレス)で管理するのでTMEMは行アドレスで管理する。外部値 P_b 、確率 P_b を計算する場合は、LMEMからリードしたLMEM変数とTMEMからリード

50

した T M E M 変数が、演算回路にそれぞれ入力されて演算処理が実行される。

【 0 0 4 2 】

以下、T M E M からリードした複数の T M E M 変数をシフト値に応じてローテイトする方法について説明する。すなわち、図 5 および図 6 に示すように、ブロックサイズ 8 の検査行列 H 4 を用いた、T M E M 1 3 2 からリードした 8 個の T M E M 変数の処理の場合には、誤り検出訂正回路 1 0 1 では、L M E M 1 2 6 と、T M E M 1 2 6 と、演算ユニット 1 3 0 と、ローテイター 1 3 2 と、が使用される。演算ユニット 1 3 0 は 8 個の演算回路 A L U 0 ~ A L U 8 からなり、8 個の処理を並列処理することができる。なお、ブロックサイズ 8 の検査行列 H 3 を用いた場合のシフト値は、0 ~ 7 の 8 種類である。

【 0 0 4 3 】

図 5 (A) および図 6 (A) に示すように、シフト値が「 0 」のブロック B (0) 場合は、ローテイター 1 3 2 によりローテイト値 0 のローテイト処理が行われ、同一アドレスの変数同士で演算が実行される。ただし、ローテイト値 0 のローテイト処理とはローテイトされないことである。

【 0 0 4 4 】

列アドレス 0 の L M E M 変数、行アドレス 0 の T M E M 変数 (図 5 (A) 破線表示)

列アドレス 1 の L M E M 変数、行アドレス 1 の T M E M 変数

列アドレス 2 の L M E M 変数、行アドレス 2 の T M E M 変数

・

・

列アドレス 7 の L M E M 変数、行アドレス 7 の T M E M 変数 (図 5 (A) 破線表示)

【 0 0 4 5 】

これに対して、図 5 (B) および図 6 (B) に示すように、シフト値が「 1 」のブロック B (1) の場合は、ローテイター 1 3 2 によりローテイト値 1 のローテイト処理が行われ、以下のような変数同士で演算が実行される。すなわち、ローテイト値 1 のローテイト処理とは、それぞれの変数をひとつ下段に移動し、あふれた最下段の変数を最上段に挿入するシフト処理である。

【 0 0 4 6 】

列アドレス 0 の L M E M 変数、行アドレス 7 の T M E M 変数 (図 5 (B) 破線表示)

列アドレス 1 の L M E M 変数、行アドレス 0 の T M E M 変数 (図 5 (B) 破線表示)

列アドレス 2 の L M E M 変数、行アドレス 1 の T M E M 変数

・

・

列アドレス 7 の L M E M 変数、行アドレス 6 の T M E M 変数

【 0 0 4 7 】

そして、図 6 (C) に示すように、シフト値が「 7 」のブロック B (7) の場合は、ローテイター 1 3 2 によりローテイト値 7 のローテイト処理が行われ、以下のような変数同士で演算が実行される。すなわち、ローテイト値 7 のローテイト処理とは、ローテイト値 1 のローテイト処理を 7 回行う処理である。

【 0 0 4 8 】

列アドレス 0 の L M E M 変数、行アドレス 1 の T M E M 変数

列アドレス 1 の L M E M 変数、行アドレス 2 の T M E M 変数

列アドレス 2 の L M E M 変数、行アドレス 3 の T M E M 変数

・

・

列アドレス 7 の L M E M 変数、行アドレス 0 の T M E M 変数

【 0 0 4 9 】

図 7 に、ブロックシフト値 (0 ~ 7) 毎の演算回路入力変数の組み合わせを示す。

【 0 0 5 0 】

以上のように、ローテイター 1 3 2 は、L M E M 1 2 6 あるいは T M E M 1 3 2 からリ

10

20

30

40

50

ードした変数を演算ユニット130に入力する前にブロックのシフト値に応じたローテイト値にローテイトする。ブロックサイズ8の検査行列H3を用いた誤り検出訂正回路101の場合には、ローテイター132の最大ローテイト値は「ブロックサイズ-1」の「7」になる。信頼度の量子化ビット数を「u」とすると、それぞれの変数のビット数は「u」になるので、ローテイター132の入出力データ幅は「8×u」ビットになる。

【0051】

<第1の実施の形態>

以下、本発明の第1の実施の形態の誤り検出訂正回路、メモリコントローラ、および半導体メモリ装置（以下、「誤り検出訂正回路等」ともいう）について説明する。

【0052】

以上の説明では、説明を簡単にするために、ブロックサイズ等が極端に小さい例（ $p = 8$ 等）について説明したが、例えば、半導体メモリ装置のメモリコントローラの誤り検出訂正回路では、ブロックサイズ p は、128～256規模の値になり、非常に大きい。このため、第1の実施の形態の誤り検出訂正回路では、ブロックサイズの p 個、設置する必要のある演算回路ALUを、 $1/Y$ （ Y は2以上の整数）に削減して並列度を意図的に下げている。なお、以下の説明では、やはり説明を簡単にするために、ブロックサイズ等が小さく（ $p = 8$ ）、 $Y = 2$ の場合を例に説明する。

【0053】

最初に、図9を用いて本発明の第1の実施の形態の誤り検出訂正回路1、誤り検出訂正回路1を有するメモリコントローラ2および誤り検出訂正回路1を有する半導体メモリ装置であるメモリカード3の構成を説明する。

【0054】

図9に示すように、メモリカード3はパソコンまたはデジタルカメラ等のホスト4から受信したデータを記憶し、記憶したデータをホスト4に送信する記憶媒体である。メモリカード3は、半導体メモリ部（以下、単に「メモリ部」ともいう。）13と、誤り検出訂正回路1を具備したメモリコントローラ2とを有する。メモリ部13は、NAND型フラッシュメモリから構成されており、単位セルである多数のメモリセル13Dが、ビット線13E等で接続された構造を有する。

【0055】

そして、メモリコントローラ2は、バス17を介して接続された、ROM10と、CPUコア11と、RAM18と、ホストI/F（インターフェイス）14と、誤り訂正（ECC：Error Correcting Code）部15と、NAND I/F（インターフェイス）16とを有する。

【0056】

メモリコントローラ2は、CPUコア11を用いて、ホストI/F14を介してホスト4とのデータ送受信を、NAND I/F16を介してメモリ部13とのデータ送受信を行う。またメモリコントローラ2は、書き換え回数の制限を含めたメモリ部13のアドレス管理をCPUコア11で実行されるFW（Firm Ware）で実現している。また、ホスト4からのコマンド入力に応じたメモリカード3全体の制御もFWで実行される。ROM10には、メモリカード3の制御プログラム等が記憶されており、RAM18には、アドレス管理で必要となるアドレス変換テーブル等が記憶される。

【0057】

ECC部15は、データ記憶時に誤り訂正符号を生成し付与する符号化器12と、データ読み出し時に、読み出された符号化データを復号する誤り検出訂正回路1とを有する。本実施の形態のECC部15は、確率にもとづく反復計算により復号処理される誤り訂正符号であるLDPC符号を用いる。

【0058】

そして、図10に示すように、誤り検出訂正回路1は、制御部20と、受信部21と、初期LLR算出部22と、硬判定部24と、パリティ検査部25と、第1のメモリであるビットノード記憶部（LMEM）26と、ローテイター32と、第2のメモリであるチェ

10

20

30

40

50

ックノード記憶部 (T M E M) 3 1 と、演算ユニット 3 0 と、第 1 のセクタであるセクタ 2 3 と、第 2 のセクタであるセクタ 3 6 と、を有する。制御部 2 0 は誤り検出訂正回路 1 全体の制御を行うが、CPU コア 1 1 の一部であってもよい。受信部 2 1 は、バス 1 7 を介してメモリ部 1 3 に記憶されていた符号化データを受信する。ここで、符号化データは、例えば、N A N D 型フラッシュメモリの読み出し電圧として誤り検出訂正回路 1 に入力される。初期 L L R 算出部 2 2 は入力された符号化データをもとに図示しない対数尤度比テーブル等を用いて初期 L L R を算出する。硬判定部 2 4 は、事後確率 (L L R) をもとに、ビットデータが「 0 」または「 1 」のいずれかが硬判定する。

【 0 0 5 9 】

パリティ検査部 2 5 は、硬判定されたデータのパリティ検査を行う。セクタ 2 3 は、制御部 2 0 の制御によりデータを、検査行列のブロックサイズ p にあわせた $p (8)$ 個のグループ単位にデータを分割する。セクタ 3 5、3 6 は、制御部 2 0 の制御によりそれぞれのグループから、 $1 / Y (1 / 2)$ のサイズの $p / Y (4)$ 個のサブグループを選択し分割する。すなわち、それぞれの分割処理はセクタ 2 3、3 5、3 6 が所定のデータのみを選択することにより行われる。ビットノード記憶部 (L M E M) 2 6 はビットノード単位での事前/事後確率 を求めるための第 1 の変数である L M E M 変数を格納する。チェックノード記憶部 (T M E M) はチェックノード単位で外部値 を求めるための第 2 の変数である T M E M 変数を格納する。ローテーター 3 2 は制御部 2 0 の制御により、T M E M 変数を所定のローテイト値にローテイト処理する。演算ユニット 3 0 は、チェックノード演算部 2 9 と、ビットノード演算部 2 8 とを有する。

【 0 0 6 0 】

次に、図 1 0 ~ 図 1 4 を用いて、本実施の形態の誤り検出訂正回路 1 の動作の流れについて説明する。

【 0 0 6 1 】

<ステップ S 1 0 > データ受信

受信部 2 1 は、メモリ部 1 3 に記憶されていた符号化データを受信する。

【 0 0 6 2 】

<ステップ S 1 1 > 初期 L L R 値受信

初期 L L R 算出部 2 2 が、尤度比テーブル (不図示) 等をもとに符号化データから初期 L L R (事前確率) を算出する。

【 0 0 6 3 】

<ステップ S 1 2 > 硬判定

L L R をもとに、硬判定が行われ、一時推定語が生成される。

【 0 0 6 4 】

<ステップ S 1 3 > パリティチェック

パリティ検査部 2 5 は、一時推定語を用いてパリティチェックを行う。

【 0 0 6 5 】

<ステップ S 1 4 > パリティチェック O K ?

パリティ検査部 2 5 は、パリティチェックの結果、誤りがないことが確認された場合 (Y e s) には、ステップ S 1 5 において、復号データをホスト I / F 1 4 を介してホスト 4 へ出力し、処理は終了する。

【 0 0 6 6 】

<ステップ S 1 6 > 初期化 / 分割

パリティ検査部 2 5 は、パリティチェックの結果、誤りがあった場合 (S 1 4 : N o) には、イタレーション処理を開始するために、ブロックのシフト値を初期化する。また、制御部 2 0 は、セクタ 2 3 により、事前確率 となる初期 L L R のデータを $p (8)$ 個単位のグループに分割し、グループ単位で以下の処理を行う。

【 0 0 6 7 】

<ステップ S 1 7 > 事前確率 格納

所定のグループの事前確率 (L M E M 変数) が、L M E M 2 6 に送信され、第 1 のア

10

20

30

40

50

ドレスである L M E M アドレスと関連付けて格納される。

【 0 0 6 8 】

<ステップ S 1 8 > サブグループ分割

L M E M 2 6 に格納された L M E M 変数は、セクタ 3 6 により Y (2) 個のサブグループに分割され、順に S 1 9 からの処理が行われる。ここで、サブグループは、L M E M アドレスにもとづいて、例えば、アドレス中央値を基準に、上位アドレスサブグループと下位アドレスサブグループとに分割される。後に説明するように T M E M 変数も、セクタ 3 5 により Y (2) 個のサブグループに分割され処理される。例えば、L M E M 変数、T M E M 変数は、アドレスが (0 、 1 、 2 、 3) のサブグループ L A 、 T A と、アドレスが (4 、 5 、 6 、 7) のサブグループ L B 、 T B と、に分割される。

10

【 0 0 6 9 】

<ステップ S 1 9 > チェックノード処理 (行処理) / ビットノード処理 (列処理)

ビットノードから接続されているチェックノードに事前確率 が送信され、チェックノード演算部 2 9 は、受信した確率の積を算出し接続されているビットノードに外部情報として送信する。外部情報は T M E M 3 1 に第 2 のアドレスである T M E M アドレスと関連付けて格納される。

【 0 0 7 0 】

チェックノードでは、接続されているビットノードに外部情報 が送信され、ビットノード演算部 2 8 は受信した確率の積を算出し接続されているチェックノードに、事後確率として送信する。事後確率は L M E M 2 6 に、第 1 のアドレスである L M E M アドレスと関連付けて格納される。なお T M E M 3 1 、 L M E M 2 6 に、既に格納されている変数がある場合には、新しい変数が上書きされる。

20

【 0 0 7 1 】

このとき、図 1 1 ~ 図 1 4 に示すように、シフト値が偶数の場合、ステップ 1 にてサブグループ L A (変数アドレス 0 ~ 3) 、サブグループ T A (変数アドレス 0 ~ 3) をアクセスし、ステップ 2 にてサブグループ L B (変数アドレス 4 ~ 7) 、サブグループ T B (変数アドレス 4 ~ 7) をアクセスし、シフト値が奇数の場合、ステップ 1 にてサブグループ L A (変数アドレス 0 ~ 3) 、サブグループ T B (変数アドレス 4 ~ 7) をアクセスし、ステップ 2 にてサブグループ L B (変数アドレス 4 ~ 7) 、サブグループ T A (変数アドレス 0 ~ 3) をアクセスする。ここで、サブグループにアクセスするとは、演算ユニット 3 0 がアクセスされた 2 つの変数を用いて演算処理を行うことを意味する。

30

すなわち、並列処理毎に、T M E M 3 1 に格納されていた T M E M 変数は、シフト値とローテイト値とに応じたローテイト処理がローテイター 3 2 により行われ、セクタ 3 5 、 3 6 により演算に必要なサブグループの変数が選択される。

【 0 0 7 2 】

なお、図 1 4 等において、ローテイト値 4 (0) とは、ローテイト処理により、並び順が 1 周して再び「 0 」になったという意味である。

【 0 0 7 3 】

<ステップ S 2 0 ~ S 2 3 >

全てのサブグループ (ステップ : ローテイト値) / グループ (シフト値) の処理が完了するまで、行処理 / 列処理が行われる。すなわち、 $p = 8$ 、 $Y = 2$ の場合には、ステップ S 1 8 ~ S 2 3 の処理が 8 回行われ、さらに、それぞれの繰り返し処理の中でステップ S 1 9 ~ S 2 1 の処理が 2 回行われる。

40

【 0 0 7 4 】

1 ラウンドの処理が完了すると、ステップ S 1 2 において硬判定処理が行われ、さらにステップ S 1 3 においてパリティチェックが行われ、OK の場合には処理は完了する。NG の場合には、ステップ S 1 7 からの処理が繰り返し行われる。

【 0 0 7 5 】

以上で説明した誤り検出訂正回路 1 は、演算ユニット 3 0 の演算回路 A L U の数を p (8) 個から $1 / Y$ ($1 / 2$) の、 p / Y (4) 個に低減できる。すなわち、回路規模が小

50

さく、消費電力が低減した処理効率のよい誤り検出訂正回路である。また、誤り検出訂正回路 1 を有するメモリコントローラ 2、および誤り検出訂正回路 1 を有する半導体メモリ装置であるメモリカード 3 は処理効率が良い。

【 0 0 7 6 】

なお、ブロックサイズ p が 64 以上のときに、上記実施の形態の誤り検出訂正回路 1 の効果が明瞭となり、256 以上のとき、さらに顕著となる。 p の上限は実装の制限から決定されるが、例えば 1k 程度である。

【 0 0 7 7 】

以上の説明のように、本実施の形態の誤り検出訂正回路 1 は、低密度パリティ検査符号の検査行列のタナーグラフに対応して設定された複数の検査ノードおよび複数のビットノードの間で外部値、確率を反復伝搬させ、その過程において確率から算出される事後値（事後確率： LLR ）の硬判定値を検査行列でパリティ検査することにより収束判定し受信データの推定値を決定するサムプロダクト・アルゴリズムを用いて、受信データからの符号検出を行い、サイズ p （ p は 8 以上の整数）の単位行列からなるブロックと、前記単位行列の各行が「1」から「 $p - 1$ 」のシフト値に対応してシフトしている複数のブロックと、からなる前記検査行列をもとに、前記受信データを、 p 個のグループに分割する第 1 のセクタと、それぞれの前記グループを、さらに Y 個（ Y は 2 以上の整数）のサブグループに分割する第 2 のセクタと、確率を算出処理するための複数の第 1 の変数を、第 1 のアドレスと関連付けて格納する第 1 のメモリと、外部値を算出処理するための複数の第 2 の変数を、第 2 のアドレスと関連付けて格納する第 2 のメモリと、前記第 2 のメモリに格納された複数の前記第 2 の変数に対して、前記シフト値と前記サブグループとに応じたローテイト値のローテイト処理を行い、演算ユニットに送信するローテイターと、前記シフト値に連動して前記サブグループ単位で、前記第 1 の変数と前記第 2 の変数とを用いて並列演算処理を行う（ p / Y ）個の演算回路からなる前記演算ユニットと、前記シフト値およびローテイト値に応じて、前記第 1 のセクタ、前記第 2 のセクタ、前記演算ユニット、および前記ローテイターを制御する制御部と、を具備する。

【 0 0 7 8 】

< 第 2 の実施の形態 >

以下、図面を参照して本発明の第 2 の実施の形態の誤り検出訂正回路 1 A 等について説明する。第 2 の実施の形態の誤り検出訂正回路 1 A 等は、第 1 の実施の形態の誤り検出訂正回路 1 と類似しているため、同じ説明は省略する。

【 0 0 7 9 】

第 1 の実施の形態の誤り検出訂正回路 1 は、演算ユニット 30 の演算回路 ALU の数を $1 / Y$ に低減した。しかし、ブロックのシフト値制御の都合で、ローテイターの最大ローテイト値は「ブロックサイズ - 1」を維持する必要がある。

【 0 0 8 0 】

すなわち、ここで、1 回目のラウンドで T MEM 変数 $T_1 \sim T_4$ をローテイターに入力し、2 回目のラウンドで $T_5 \sim T_8$ を入力するように制御する場合を考えてみる。ブロックのシフト値が「7」だった場合、例えば T MEM 変数 T_1 を、最大シフト値の 7 ローテイトする必要があるが（図 6（B）参照）、ローテイターは最大で 4 しかローテイトできないため、要求機能を満足することができない。すなわち、演算回路数は 4 に半減できるが、これにあわせてローテイターのローテイト機能（回路規模）を半減することはできなかった。また、ローテイター 32 に入力する T MEM 変数の数も、ブロックサイズ分の 8 個必要になるので、T MEM 31 のリードデータ幅は「 $8 \times u$ 」ビットのままである。さらに、演算ユニット 30 に必要な 4 個の T MEM 変数を入力するためには、「 $8u$ ビット to $4u$ ビット」のセクタ 36 が必要になる。

【 0 0 8 1 】

このため、ブロックサイズが 256 規模あり、 $u = 6$ ビットの誤り検出訂正回路では、ローテイター単体で例えば 50K ゲート規模に達し、データバス幅も 256×6 ビット = 1536 ビットと膨大になる。このため、T MEM リードデータ幅、ローテイター規模、

10

20

30

40

50

セクタ規模、および前記回路を接続するデータバス幅等を、演算ユニット削減数に比例させて削減することはできなかった。

【 0 0 8 2 】

これに対して、本実施の形態の誤り検出訂正回路 1 A では、L M E M 変数および T M E M 変数をアドレスの大小による分割ではなく、アドレスの下位ビットにもとづき分割する。例えば、例えば、アドレスの最下位 1 ビットにもとづいて、アドレスが偶数のサブグループ (0 、 2 、 4 、 6) T e と、アドレスが奇数のサブグループ (1 、 3 、 5 、 7) T o と、の、2 グループに分割する。

【 0 0 8 3 】

すなわち、アドレスが偶数とは、アドレスの最下位 1 ビットが「 0 」であることを意味し、アドレスが奇数とは、アドレスの最下位 1 ビットが「 1 」であることを意味している。

10

【 0 0 8 4 】

そして、誤り検出訂正回路 1 A のイタレーション処理においては、シフト値が偶数の場合、ステップ 1 にてサブグループ L e、サブグループ T e をアクセスし、ステップ 2 にてサブグループ L o、サブグループ T o をアクセスし、シフト値が奇数の場合、ステップ 1 にてサブグループ L e、サブグループ T o をアクセスし、ステップ 2 にて L o、T e をアクセスする。

【 0 0 8 5 】

すなわち、ブロック演算処理のステップ 1 でサブグループ L e の処理を実行し、ステップ 2 でサブグループ L o の処理を実行すると、ブロックシフト値が偶数の場合、ステップ 1 にてサブグループ L e はサブグループ T e と演算を実行し、ステップ 2 にてサブグループ L o はサブグループ T o と演算を実行する。そしてブロックシフト値が奇数の場合、ステップ 1 にてサブグループ L e はサブグループ T o と演算を実行し、ステップ 2 にてサブグループ L o はサブグループ T e と演算を実行する。よって、ブロックシフト値が偶数の場合、ステップ 1 にてサブグループ L e、サブグループ T e をアクセスし、ステップ 2 にてサブグループ L o、サブグループ T o をアクセスする。そしてブロックシフト値が奇数の場合、ステップ 1 にてサブグループ L e、サブグループ T o をアクセスし、ステップ 2 にてサブグループ L o、サブグループ T e をアクセスする。

20

【 0 0 8 6 】

すなわち、誤り検出訂正回路 1 A では、T M E M 変数は下位ビットにもとづいてサブグループ分けされているために、ローテイター 3 2 A には演算処理に必要なサブグループだけが入力する。

30

【 0 0 8 7 】

図 1 5 ~ 図 1 7 に、誤り検出訂正回路 1 A の各シフト値 / 各ステップにおける T M E M 変数ローテイト値の関係を示す。また図 1 7 にシフト値、ステップ、ローテイト値のまとめを示す。

【 0 0 8 8 】

誤り検出訂正回路 1 A では、演算ユニット 3 0 A の演算回路数を、 $1 / Y = 1 / 2$ 、に削減できるだけでなく、T M E M 3 1 A へのアクセス単位 (1 ワード) のデータ幅を 8 u ビットから 4 u ビットと $1 / 2$ 、に削減できる。さらに、ローテイター 3 2 A の最大ローテイト値および入出力データ幅、T M E M 3 1 A とローテイター 3 2 A と演算ユニット 3 0 A とを接続するデータバス幅、も $1 / 2$ に削減できる。また、「 8 u ビット to 4 u ビット」のセクタ 3 6 は不要となる。

40

【 0 0 8 9 】

すなわち、誤り検出訂正回路 1 A は、第 1 の実施の形態の誤り訂正部 1 が有する効果に加えてローテイターを含む周辺メモリ、回路、およびデータバスの幅を、演算回路数に合わせて削減できる。すなわち、誤り検出訂正回路 1 A は、柔軟に処理速度と回路規模 / 消費電力の調整が可能なスケーラビリティを備える。

【 0 0 9 0 】

50

以上の説明のように、本実施の形態の誤り検出訂正回路 1 A では、制御部 2 0 が、第 1 のアドレスおよび第 2 のアドレスの最下位アドレス 1 ビットをもとに、各グループを 2 個のサブグループに分割するように制御する。

【 0 0 9 1 】

また、誤り検出訂正回路 1 A は、LDPC 符号による誤り検出 / 訂正処理機能を備え、1 ラウンドの処理 (ラウンド処理) で、選択したビットノードに関して、LDPC 符号のパリティ検査行列のタナグラフに対応して設定された複数のチェックノードから当該チェックノードに接続されている複数のビットノードに向け外部情報 () を伝搬する行処理、および、当該複数のビットノードから該複数のチェックノードに向け事後確率 () を伝搬する列処理を連続実行する。そして、ラウンド処理繰り返しの過程において、から算出される事後値を用いて収束判定し符号データの推定し、入力値からの符号検出を行う。そして、サイズ $p \times p$ (p は 8 以上の整数) の正方行列を複数個組み合わせて検査行列を構成するもので、ビットノード単位での事前 / 事後確率 を求めるための変数を格納 / 管理するメモリ (LMEM) と、チェックノード単位で外部値 を求めるための変数を格納 / 管理するメモリ (TMEM) を備え、ブロックのシフト値に連動して p 個のビットノードと、 p 個のチェックノード単位で演算処理する機能を備えるもので、 p 個のビットノードとチェックノードを p / Y 個 (Y は 2 以上の整数) のサブグループに分け、サブグループに対応する変数グループ単位で、LMEM / TMEM アクセスとアクセスした変数のローテーション処理を実行するもので、LMEM / TMEM の下位アドレスでサブグループ分けをする。

【 0 0 9 2 】

さらに、誤り検出訂正回路 1 A は、演算処理を実行するとき、LMEM、TMEM いずれかのワードアクセス順とローテーターのローテイト値を、ブロックシフト値に応じて、決定する機能を備えることが好ましく、特に好ましくは演算処理を実行するとき、TMEM のワードアクセス順とローテーターのローテイト値を、ブロックシフト値に応じて、決定する機能を備える。

【 0 0 9 3 】

< 第 3 の実施の形態 >

以下、図面を参照して本発明の第 3 の実施の形態の誤り検出訂正回路 1 B 等について説明する。第 3 の実施の形態の誤り検出訂正回路 1 B 等は、第 2 の実施の形態の誤り検出訂正回路 1 A と類似しているため、同じ説明は省略する。

【 0 0 9 4 】

本実施の形態の誤り検出訂正回路 1 B は、サブグループ分割数 $Y = 4$ の場合であり、演算ユニット 3 0 B は、 $2 (8 / 4)$ 個の演算回路 ALU を有している。すなわち、LMEM 2 6 B、TMEM 3 1 B に格納された変数のアドレス最下位 2 ビットが「00」、「01」、「10」、「11」のいずれであるかに、もとづいて、LMEM 変数および TMEM 変数を、それぞれ 4 つのサブグループに分割して、処理する。

【 0 0 9 5 】

誤り検出訂正回路 1 B では、第 2 の実施の形態の誤り検出訂正回路 1 A ($Y = 2$) と同様に、制御部 2 0 がシフト値毎に、LMEM 2 6 B、TMEM 3 1 B からアクセスする変数グループを、ローテイト値に応じてローテーター 3 2 B がローテイト処理する。すなわち、制御部 2 0 が、ステップに応じて TMEM 変数のローテイト値を制御する。図 2 2 に誤り検出訂正回路 1 B におけるシフト値、ステップ毎、ローテイト値のまとめを示す。

【 0 0 9 6 】

誤り検出訂正回路 1 B は、誤り検出訂正回路 1 A が有する効果に加えて、より回路規模 / 消費電力を低減できる。なお、サブグループ分割数 Y の上限は、処理速度と回路規模 / 消費電力とのトレードオフの関係から適宜、決定されるが、ブロックサイズ p 以下であり例えば 8 である。

【 0 0 9 7 】

以上の説明のように、本実施の形態の誤り検出訂正回路 1 B では、制御部 2 0 が、第 1

10

20

30

40

50

のアドレスおよび第2のアドレスの下位アドレス2ビットをもとに、各グループを4個のサブグループに分割するように制御する。

【0098】

なお以上は、ホスト4と接続されるメモリカード3からなるメモリシステムを例に説明したが、メモリシステムとしては、ホスト4の内部に収納され、ホスト4の起動データ等を記憶する、いわゆるエンベデッドタイプのNAND型フラッシュメモリ装置等でもメモリカード3等と同じ効果を得ることができる。さらには、符号化データの復号を確率にもとづく反復計算により行う誤り検出訂正回路であれば、デジタルテレビ放送受信機等の誤り検出訂正回路でも半導体メモリシステムと同じ効果を得ることができる。また、誤り検出訂正回路としては専用の回路ではなく汎用CPUに読み込まれ動作するものであってもよい。

10

【0099】

本発明の、さらに別の実施の形態によるLDPC符号による誤り検出訂正方法は、受信信号からLLRを算出するLLR算出工程と、前記LLRまたは確率をもとに硬判定を行う硬判定工程と、硬判定結果をもとにパリティ検査を行うパリティ検査工程と、前記受信信号をp個のグループに分割するグループ分割工程と、それぞれの前記グループをY個のサブグループに分割するサブグループ分割工程と、前記確率を算出処理するための前記サブグループの複数の第1の変数を、第1のアドレスと関連付けて第1のメモリに格納するとともに、外部値を算出処理するための前記サブグループの複数の第2の変数を、第2のアドレスと関連付けて第2のメモリに格納する変数格納工程と、前記第1のメモリに格納された前記複数の第1の変数に対して選択処理する選択工程と、前記第2のメモリに格納された前記複数の第2の変数に対してローテイト処理するローテイト工程と、選択処理されたp/Y個の前記第1の変数とローテイト処理されたp/Y個の前記第2の変数とをp/Y個の演算回路により並列処理する並列処理工程と、を具備する。ただし前記LDPC符号はサイズp(pは8以上の整数)の単位行列からなるブロックと、前記単位行列の各行が「1」から「p-1」のシフト値に対応してシフトしている複数のブロックと、からなる検査行列をもとにし、前記グループは前記シフト値に対応しており、前記ローテイト処理および前記選択処理は、前記シフト値および前記サブグループに応じたローテイト値にもとづき実行される。

20

【0100】

さらに、上記誤り検出訂正方法において、サブグループ分割工程は、前記第1のアドレスおよび前記第2のアドレスの下位アドレスをもとに行われることが好ましく、より好ましくは前記第1のアドレスおよび前記第2のアドレスの最下位アドレス1ビットをもとに行われる。

30

【0101】

上記のように、本発明は、上述した実施の形態に限定されるものではなく、本発明の要旨を変えない範囲において、種々の変更、改変等が可能である。

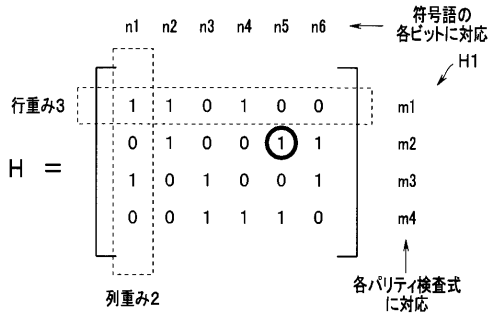
【符号の説明】

【0102】

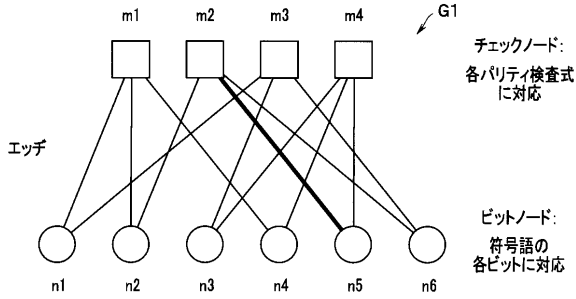
1、1A、1B...誤り検出訂正回路、2...メモリコントローラ、3...メモリカード、4...ホスト、10...ROM、11...CPUコア、12...符号化器、13...メモリ部、13D...メモリセル、13E...ビット線、14...Host I/F、16...NAND I/F、17...バス、18...RAM、20...制御部、21...受信部、22...算出部、23...セレクタ、24...硬判定部、25...パリティ検査部、26、26A、26B...ビットノード記憶部(LMEM)、28...ビットノード演算部、29...チェックノード演算部、30、30A、30B...演算ユニット、31、31A、31B...チェックノード記憶部(TMEM)、32、32A、32B...ローテイター、35、36...セレクタ、101...誤り検出訂正回路、130...演算ユニット、132...ローテイター、ALU...演算回路

40

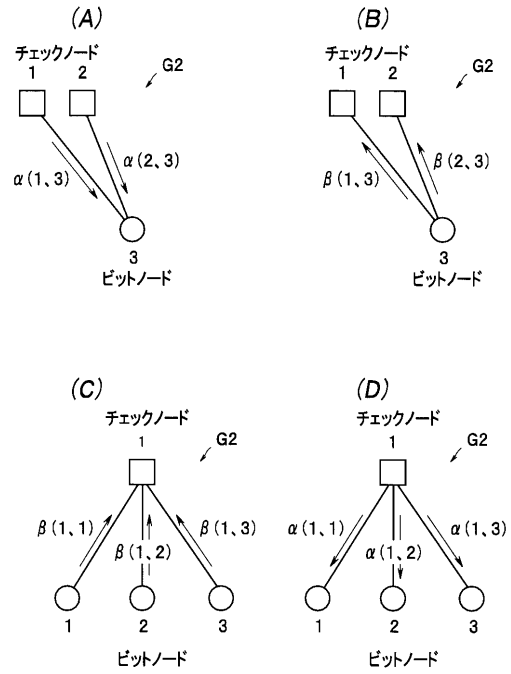
【図1】



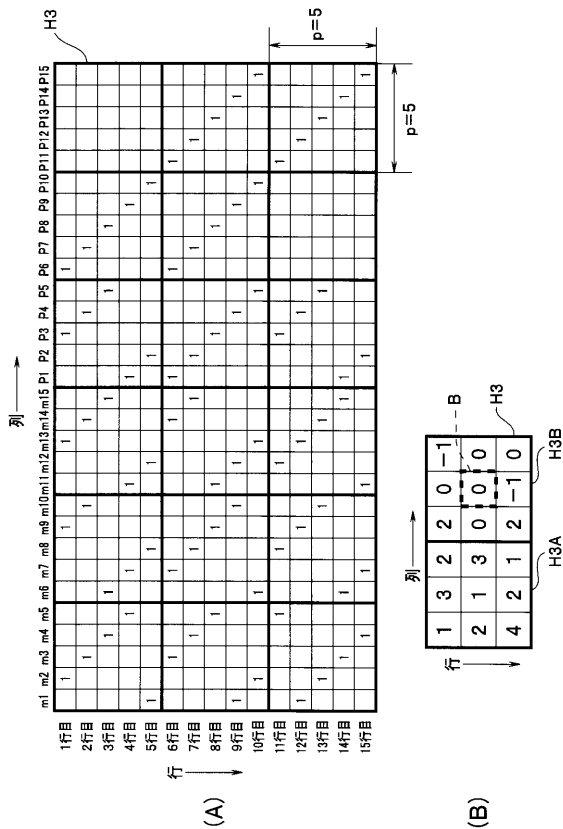
【図2】



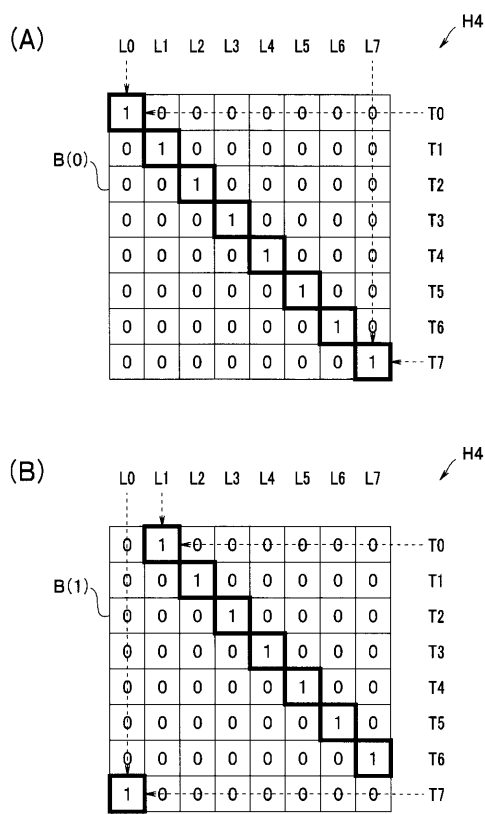
【図3】



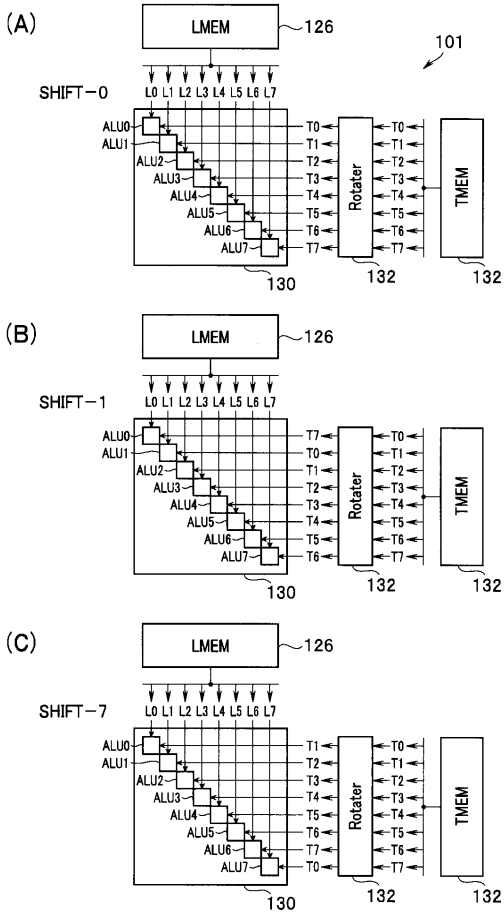
【図4】



【図5】



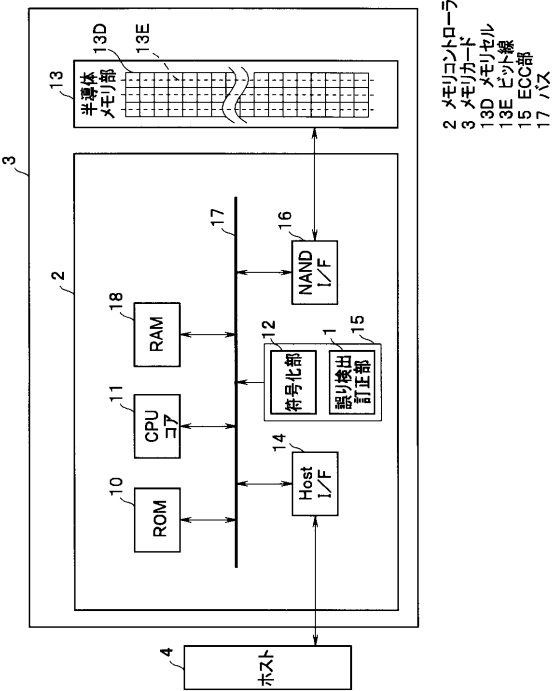
【図 6】



【図 7】

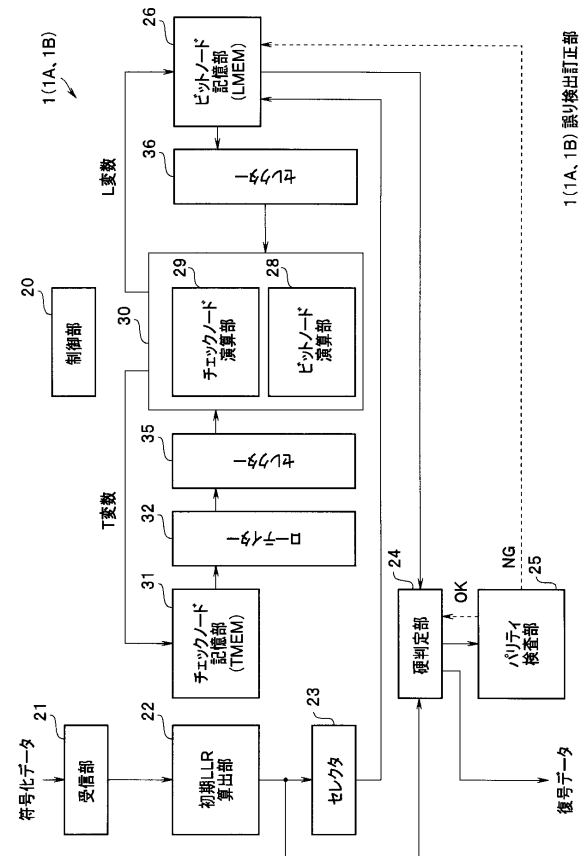
シフト値	LMEM	TMEM	ローテイト値
	変数アドレス	変数アドレス	
0	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	0
1	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	1
2	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	2
3	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	3
4	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	4
5	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	5
6	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	6
7	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7	7

【図 8】



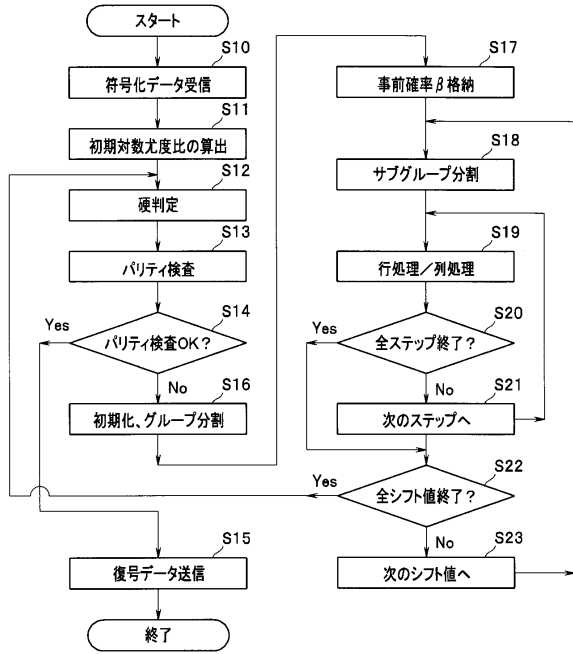
- 2 メモリコントローラ
- 3 メモリカード
- 13D メモリセル
- 13E ビット線
- 15 ECC部
- 17 パス

【図 9】

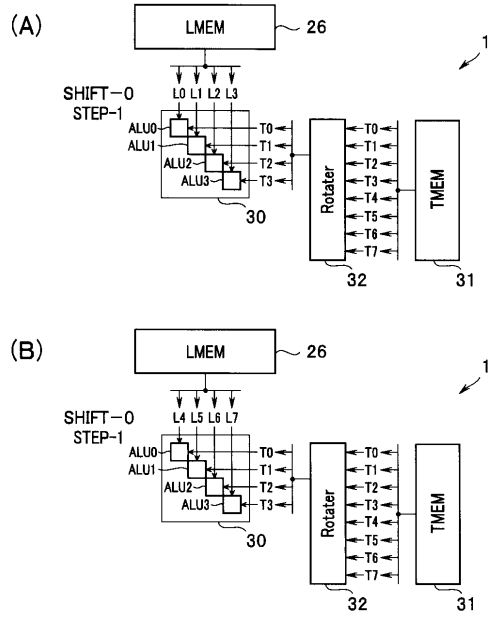


- 1 (1A, 1B) 誤り検出訂正部

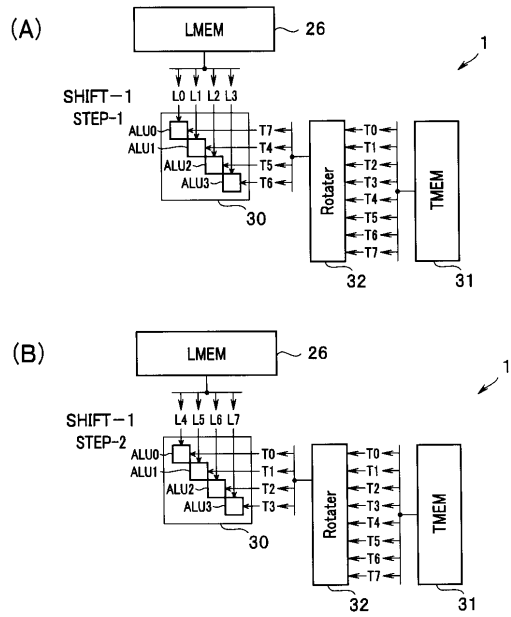
【図10】



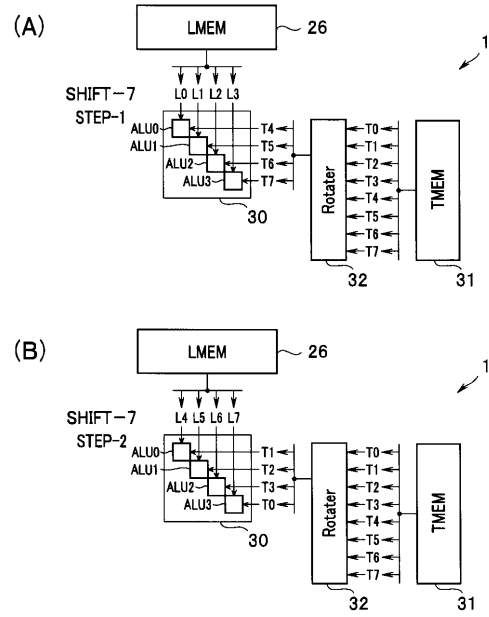
【図11】



【図12】



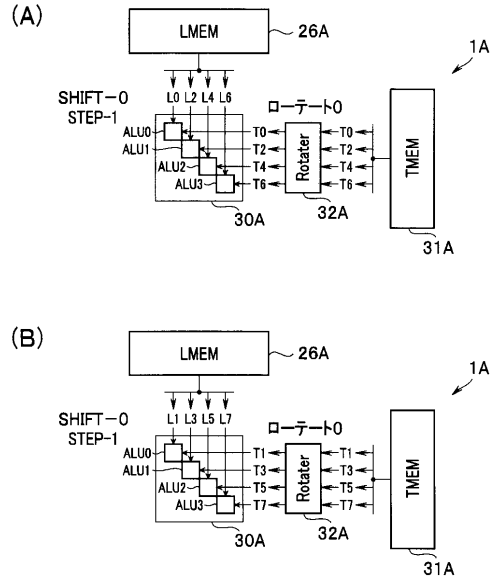
【図13】



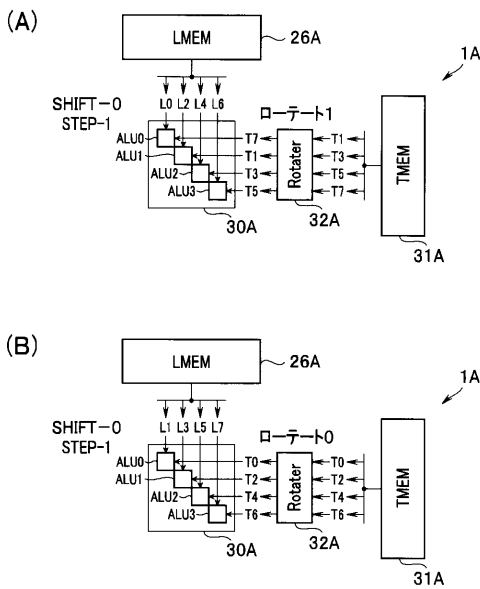
【図14】

シフト値	ステップ	LMEM	TMEM	ローテート値
		変数アドレス	変数アドレス	
0	1	0, 1, 2, 3	0, 1, 2, 3	0
	2	4, 5, 6, 7	4, 5, 6, 7	0
1	1	0, 1, 2, 3	4, 5, 6, 7	1
	2	4, 5, 6, 7	0, 1, 2, 3	0
2	1	0, 1, 2, 3	0, 1, 2, 3	1
	2	4, 5, 6, 7	4, 5, 6, 7	1
3	1	0, 1, 2, 3	4, 5, 6, 7	2
	2	4, 5, 6, 7	0, 1, 2, 3	1
4	1	0, 1, 2, 3	0, 1, 2, 3	2
	2	4, 5, 6, 7	4, 5, 6, 7	2
5	1	0, 1, 2, 3	4, 5, 6, 7	3
	2	4, 5, 6, 7	0, 1, 2, 3	2
6	1	0, 1, 2, 3	0, 1, 2, 3	3
	2	4, 5, 6, 7	4, 5, 6, 7	3
7	1	0, 1, 2, 3	4, 5, 6, 7	4(0)
	2	4, 5, 6, 7	0, 1, 2, 3	3

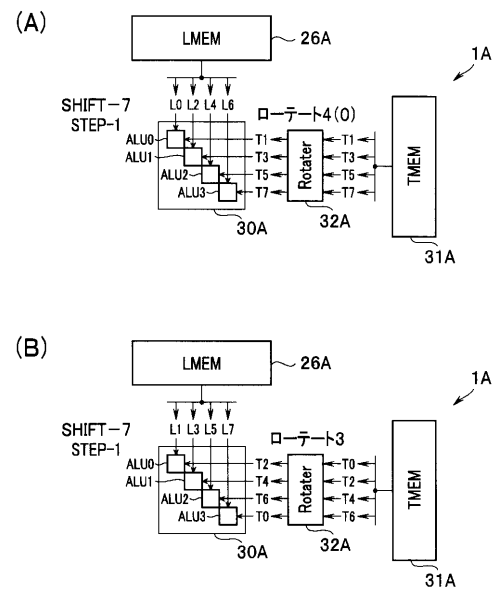
【図15】



【図16】



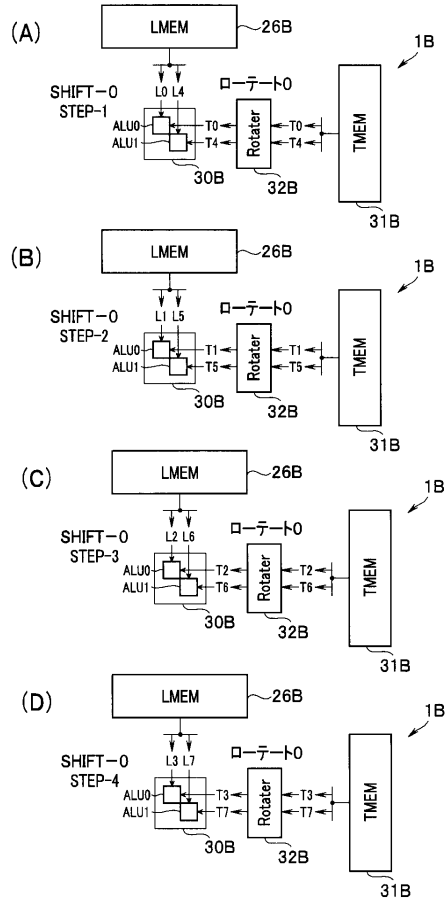
【図17】



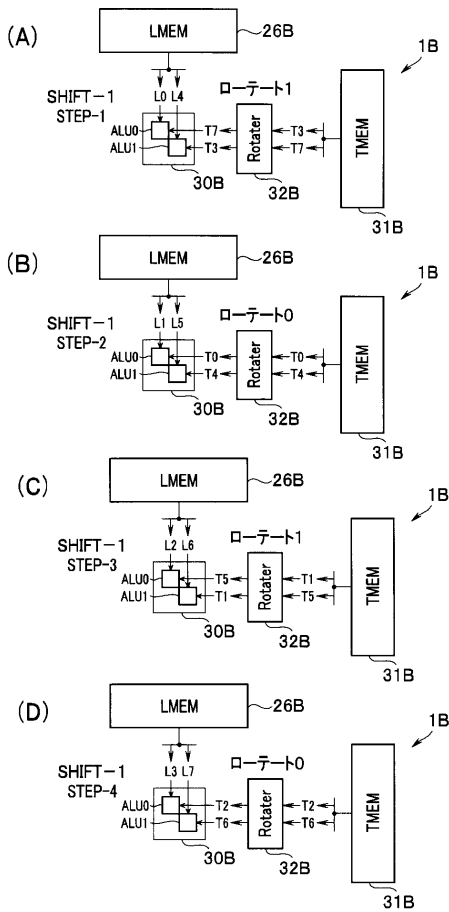
【図18】

シフト値	ステップ	LMEM	TMEM	ローテート値
		変数アドレス	変数アドレス	
0	1	0, 2, 4, 6	0, 2, 4, 6	0
	2	1, 3, 5, 7	1, 3, 5, 7	0
1	1	0, 2, 4, 6	1, 3, 5, 7	1
	2	1, 3, 5, 7	0, 2, 4, 6	0
2	1	0, 2, 4, 6	0, 2, 4, 6	1
	2	1, 3, 5, 7	1, 3, 5, 7	1
3	1	0, 2, 4, 6	1, 3, 5, 7	2
	2	1, 3, 5, 7	0, 2, 4, 6	1
4	1	0, 2, 4, 6	0, 2, 4, 6	2
	2	1, 3, 5, 7	1, 3, 5, 7	2
5	1	0, 2, 4, 6	1, 3, 5, 7	3
	2	1, 3, 5, 7	0, 2, 4, 6	2
6	1	0, 2, 4, 6	0, 2, 4, 6	3
	2	1, 3, 5, 7	1, 3, 5, 7	3
7	1	0, 2, 4, 6	1, 3, 5, 7	4(0)
	2	1, 3, 5, 7	0, 2, 4, 6	3

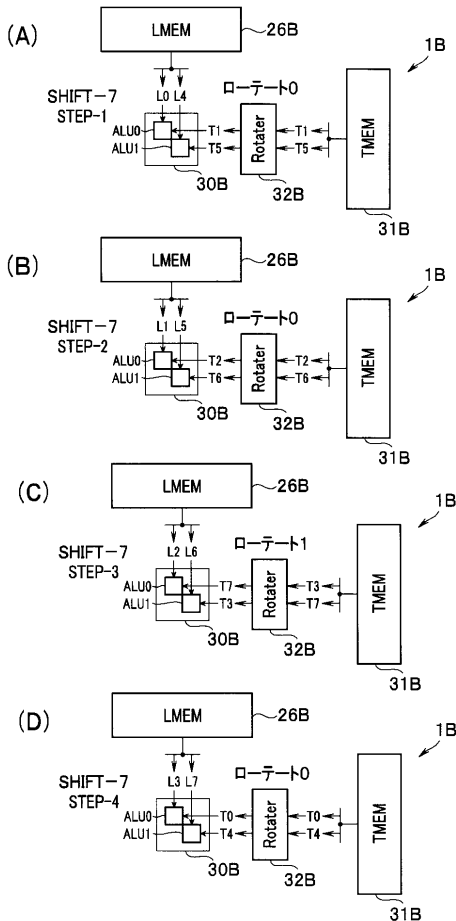
【図19】



【図20】



【図21】



【 図 2 2 】

シフト値	ステップ	LMEM	TMEM	ローテート値
		変数アドレス	変数アドレス	
0	1	0,4	0,4	0
	2	1,5	1,5	0
	3	2,6	2,6	0
	4	3,7	3,7	0
1	1	0,4	3,7	1
	2	1,5	0,4	0
	3	2,6	1,5	1
	4	3,7	2,6	0
2	1	0,4	2,6	1
	2	1,5	3,7	1
	3	2,6	0,4	0
	4	3,7	1,5	0
3	1	0,4	1,5	1
	2	1,5	2,6	1
	3	2,6	3,7	1
	4	3,7	0,4	0
4	1	0,4	0,4	1
	2	1,5	1,5	1
	3	2,6	2,6	1
	4	3,7	3,7	1
5	1	0,4	3,7	1
	2	1,5	0,4	1
	3	2,6	1,5	1
	4	3,7	2,6	1
6	1	0,4	2,6	0
	2	1,5	3,7	1
	3	2,6	0,4	1
	4	3,7	1,5	1
7	1	0,4	1,5	0
	2	1,5	2,6	0
	3	2,6	3,7	1
	4	3,7	0,4	0

フロントページの続き

- (72)発明者 市川 和博
神奈川県川崎市川崎区駅前本町2-5番地1 東芝マイクロエレクトロニクス株式会社内
- (72)発明者 石川 幸夫
神奈川県川崎市川崎区駅前本町2-5番地1 東芝マイクロエレクトロニクス株式会社内

審査官 上田 翔太

- (56)参考文献 特開2009-260692(JP,A)
特表2008-515342(JP,A)
特開2009-271852(JP,A)
石川 達之、他3名,メモリ容量削減手法を用いた高スループットLDPC復号器,電子情報通信学会技術研究報告 ICD,2006年 1月20日,集積回路 105(570),pp.29-34
石川 達之、他3名,Min-Sumアルゴリズムを用いたLDPC復号器のメモリ削減手法,電子情報通信学会技術研究報告. CAS, 回路とシステム,2005年 6月21日,105(146),pp.43-48

- (58)調査した分野(Int.Cl.,DB名)
- | | |
|------|-------|
| H03M | 13/19 |
| G06F | 11/10 |