

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5649663号
(P5649663)

(45) 発行日 平成27年1月7日(2015.1.7)

(24) 登録日 平成26年11月21日(2014.11.21)

(51) Int.Cl. F 1
G 0 6 F 1 2 / 1 6 (2 0 0 6 . 0 1) G 0 6 F 1 2 / 1 6 3 2 0 G

請求項の数 16 (全 22 頁)

| | |
|--|--|
| <p>(21) 出願番号 特願2012-551186 (P2012-551186) (86) (22) 出願日 平成23年1月7日(2011.1.7) (65) 公表番号 特表2013-518345 (P2013-518345A) (43) 公表日 平成25年5月20日(2013.5.20) (86) 国際出願番号 PCT/US2011/020533 (87) 国際公開番号 W02011/094051 (87) 国際公開日 平成23年8月4日(2011.8.4) 審査請求日 平成25年12月19日(2013.12.19) (31) 優先権主張番号 12/695,038 (32) 優先日 平成22年1月27日(2010.1.27) (33) 優先権主張国 米国 (US)</p> | <p>(73) 特許権者 511226960 サンディスク テクノロジーズ インコーポレイテッド アメリカ合衆国、75024、テキサス州、プレーノー、ノース・ダラス・パークウェイ 6900、ツアー・レガシー・タウン・センター (74) 代理人 100075144 弁理士 井ノ口 壽 (72) 発明者 ドブリュー, マヌエル アントニオ アメリカ合衆国、95762、カリフォルニア州、エルドラドヒルズ、ハンティントン サークル 2166</p> |
|--|--|

最終頁に続く

(54) 【発明の名称】 ビット値の格納されているカウントを使用してデータの誤りを訂正するシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

方法であって、
 メモリに連結されたコントローラにて、
 前記メモリから読み込まれた、前記メモリのワード線のマルチビットに対応する第1のグループのデータビットが、誤り訂正符号(ECC)エンジンにより訂正不能な誤りを含むという通知を受け取るステップと、
 特定のビット値を有する前記第1のグループのデータビットのカウントと、特定のビット値を有する、先行動作中に前記ワード線のマルチビットに基づいて生成されるデータビットの先行カウントとを比較するステップと、
 前記カウントが前記先行カウントを上回るという判断にตอบสนองして、前記第1のグループのデータビットのうち、特定のビット値を有し、前記ワード線のマルチビットに対応する第2のグループのデータビットの訂正されたデータビットと同じメモリセルに対応するビットを識別するステップと、
 データビットの調整されたグループを生成するために、前記第1のグループの識別されたビットの値を変更するステップと、
 データビットの調整されたグループを、前記ECCエンジンに提供するステップと、
 を含む方法。

10

【請求項 2】

請求項 1 記載の方法において、

20

前記メモリは、複数のマルチレベルセルを含む方法。

【請求項 3】

請求項 1 記載の方法において、

識別されたビットを、ルックアップテーブルに基づき選択するステップをさらに含む方法。

【請求項 4】

請求項 3 記載の方法において、

前記ルックアップテーブルは、前記第 1 のグループのデータビットのうち、特定のビット値を有し、前記ワード線のマルチビットに対応する第 2 のグループのデータビットの訂正されたデータビットと同じメモリセルに対応するビットと、前記第 2 のグループのデータビットの訂正されたデータビットとを用いて決定されたハミング距離に少なくとも部分的に基づく方法。

10

【請求項 5】

請求項 3 記載の方法において、

前記ルックアップテーブルは、前記カウントに少なくとも部分的に基づく方法。

【請求項 6】

請求項 3 記載の方法において、

前記ルックアップテーブルは、状態値に少なくとも部分的に基づく方法。

【請求項 7】

請求項 1 記載の方法において、

前記第 1 のグループのデータビットのカウントと前記データビットの先行カウントとを比較するステップに基づいて前記第 1 のグループ内の反転するいくつかのビットを判断するステップをさらに含む方法。

20

【請求項 8】

請求項 1 記載の方法において、

前記第 1 のグループの識別されたビットの値を変更するステップは、前記第 1 のグループが訂正可能であると前記 ECC エンジンが判断するまで、前記第 1 のグループ内のビットを反復して反転することを含む方法。

【請求項 9】

メモリデバイスであって、

メモリアレイと、

前記メモリアレイのメモリセルのマルチビット値の特定のビットの誤りの可能性を示し、前記メモリセルの他のビットのいくつかの誤りに少なくとも部分的に基づくデータを格納する少なくとも 1 つの誤り確率テーブルと、

30

前記メモリアレイに連結されたコントローラであって、前記コントローラは前記メモリセルの他のビットの誤りを検出するように構成された誤り訂正符号 (ECC) エンジンを備え、前記コントローラは訂正する特定の値のいくつかのビットを判断するように構成され、前記ビットは前記メモリアレイにおける読み込み動作を介して取得され、いくつかのビットは特定の値を有する取得されたビットのカウントとビット値の格納されているカウントとを比較することに基づき判断され、前記コントローラは前記判断されたいくつかのビットのうち訂正する少なくとも 1 つのビットを識別するために、前記少なくとも 1 つの誤り確率テーブルにアクセスするようにさらに構成される、コントローラと、

40

を備えるメモリデバイス。

【請求項 10】

請求項 9 記載のメモリデバイスにおいて、

前記少なくとも 1 つの誤り確率テーブルは、読み込まれた値から部分的に訂正された値へのハミング距離によりインデックス付けされるメモリデバイス。

【請求項 11】

請求項 9 記載のメモリデバイスにおいて、

前記少なくとも 1 つの誤り確率テーブルは、部分的に訂正された値によりインデックス

50

付けされるメモリデバイス。

【請求項 1 2】

請求項 9 記載のメモリデバイスにおいて、
前記少なくとも 1 つの誤り確率テーブルは、前記コントローラにて格納されるメモリデバイス。

【請求項 1 3】

請求項 9 記載のメモリデバイスにおいて、
前記少なくとも 1 つの誤り確率テーブルは、前記メモリアレイにて格納されるメモリデバイス。

【請求項 1 4】

請求項 9 記載のメモリデバイスにおいて、
前記格納されているカウントは、前記コントローラにあるメモリに格納されるメモリデバイス。

【請求項 1 5】

請求項 9 記載のメモリデバイスにおいて、
前記格納されているカウントは、前記メモリアレイにて格納されるメモリデバイス。

【請求項 1 6】

請求項 9 記載のメモリデバイスにおいて、
前記メモリアレイは、フラッシュマルチレベルセルアレイを含むメモリデバイス。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般的に、データ処理およびストレージシステムに関し、特に、格納または送信されたデータ内の誤りを検出および訂正することに関する。

【背景技術】

【0002】

コンピュータデータストレージシステムは、典型的には、データの完全性を改善する誤り訂正符号 (ECC) プロセスを含む。例えば、ECC は、読み込み動作中、アクセスされるデータビット (例えば、プロセッサによって取得される情報を表すメモリセル) が誤った状態を有するかどうかを判断するために使用され得る。ECC 処理は、いくつかの誤りビットを検出し、訂正することによって、メモリデバイスの動作の信頼性を高めることもできる。訂正されるビットのしきい値数は、適用される特定の ECC 技法の訂正容量による。ECC によって進歩がもたらされてはいるが、適用される ECC 技法の訂正容量を超える誤りを訂正できることが望ましい。

【発明の概要】

【0003】

格納または送信されたデータ内の誤りを識別して訂正することができるシステムおよび方法が開示される。特定の実施形態は、メモリの、ワード線またはデータの 1 つ以上のページから読み込まれた、データ内にあるいくつかの誤りを追跡するとよい。読み込まれたデータ内のいくつかの誤りが、誤り訂正符号 (ECC) エンジンの能力を上回ると、ECC エンジンにより訂正可能なデータを与えるために調整する 1 つ以上のビットを判断するように、訂正モジュールが使用されるとよい。例えば、読み込まれるワード線は、ワード線の第 1 の論理ページ内の第 1 のビットおよび第 2 の論理ページ内の第 2 のビットなどの複数のビットを格納する、マルチレベルセル (MLC) を含むとよい。第 1 のページは、ECC エンジンにより訂正不能な誤りを有するとよく、第 2 のページは、ECC エンジンにより訂正される誤りを有するとよい。訂正モジュールは、第 2 のページ内の識別される誤りに少なくとも部分的に基づき、第 1 のページ内の 1 つ以上のビットを選択するとよい。例えば、特定の MLC が第 1 のページ内にビット誤りを有する可能性は、MLC が他のページ内で有するいくつかのビット誤りまたはビット誤りのタイプと相関関係にあることもある。第 1 のページは、第 1 のページ内の選択されたビットの値を変更することによ

10

20

30

40

50

て調整されるとよく（例えば、論理「1」から論理「0」または論理「0」から論理「1」に変更される）、調整された第1のページは、調整された第1のページが訂正可能かどうかを判断するために、ECCエンジンに再送されるとよい。第1のページ内のビットを選択して反転するプロセスは、調整された第1のページがECCエンジンにより訂正可能になるまで繰り返されるのが可能である。

【0004】

メモリデバイスは、そのECCエンジンの訂正容量に、特定のワード線のいくつかの誤りが近づいていると判断すると、ワード線内の誤りのタイプおよび位置の識別を支援するために追加の情報を格納するとよい。例えば、いくつかの誤りがしきい値に達すると、ワード線内のビットのグループ内の、識別されたいくつかの1のカウントが格納されるとよい。ワード線またはページ（単数または複数）内のセルは、カウントに従ってグループ分けされてもよい。例えば、ページ内のいくつかの誤りが増加するにつれて、グループの数が増加してもよい。ページの各グループのいくつかの1のカウントが、以前の読み込みから保存された、そのグループのいくつかの1のカウントと比較されるとよい。この比較は、各グループ内で反転されるビットの値とビットの数とを判断するために使用されるとよい（例えば、読み込みのカウントが、格納されているカウントを2上回ると、グループ内の2つの1が0に反転されることになる）。グループ内のどのビットが誤りの可能性が最も高いかに基づいて反転するビット（単数または複数）を判断するために、確率テーブルが使用されるとよい。

【0005】

他のページ内の訂正されたビット誤りなど、他の1つ以上のビット誤りを有すると識別されたMLCに対応するビットを反転する決定を下すために、確率テーブルが、グループ毎のいくつかの1とともに使用されるとよい。選択されたビットが反転された後、選択されたビットを反転する決定をベリファイするように、ECC手順が実行されるとよい。このプロセスは、反復型であるとよく、ECCエンジンが元のデータを回復できる時、または所定数の反復が完了したとき、または他の終了基準に従い、終了するとよい。

【図面の簡単な説明】

【0006】

【図1】メモリアレイから取得されたデータにおける誤りを識別して訂正するように構成されたコントローラを含むシステムの特定の例示の実施形態のブロック図である。

【図2】誤りビットを反転するために、誤り訂正符号計算および確率ルックアップテーブルを使用して訂正される状態を有するメモリセルを示す一般図である。

【図3】読み込みのカウントおよび誤り確率テーブルを含むメモリアレイを含むシステムの特定の実施形態のブロック図である。

【図4】メモリセル内の誤りを分析して訂正する方法の特定の実施形態の流れ図である。

【図5】メモリセル内の誤りを分析して訂正する方法の別の特定の実施形態の流れ図である。

【図6】ルックアップテーブルに基づきメモリセル内の誤りを訂正する方法の特定の実施形態の流れ図である。

【発明を実施するための形態】

【0007】

図1は、メモリアレイ102（例えば、フラッシュマルチレベルセルアレイ）から読み込まれたデータ内の、ECCエンジン118の訂正容量を上回る誤りを訂正するように構成されたコントローラ104を含むシステム100の特定の例示の実施形態のブロック図である。コントローラ104は、ECCエンジン118により訂正不能なデータ内の調整すべき1つ以上のビットを判断するように構成された訂正モジュール130を含む。

【0008】

メモリアレイ102は、1つ以上のワード線（単数または複数）108を含むとよい。ワード線（単数または複数）108は、メモリアレイ102の行を表すこともでき、1つ以上のマルチレベルセル（MLC）144と、メモリアレイ102の行内のMLC144

10

20

30

40

50

のコントロールゲートに連結されている導電線とを含む。メモリアレイ 102 は、読み込み動作 146 などを介して、通信経路 156 上でコントローラ 104 とデータの送受信を行うこともできる。

【0009】

コントローラ 104 は、ホストデバイスなどの外部デバイス 106 に応答するとよく、メモリアレイ 102 に対する書き込みおよび読み込みアクセスを可能にする。コントローラ 104 は、ECC エンジン 118 により訂正不能なデータを調整し、調整されたデータを訂正のために ECC エンジン 118 に再び提供するように構成された訂正モジュール 130 を含む。コントローラ 104 はさらに、バッファ 162、訂正済みバッファ 124、ログファイル 134 および誤り確率テーブル 138 を含む。

10

【0010】

ECC エンジン 118 は、リードソロモン (RS)、ボーズチャドローリオッケンジェム (BCH)、低密度パリティチェック (LDPC)、その他 1 つ以上の ECC プロセスまたはその任意の組み合わせなどの誤り訂正方式に基づき、誤りを復号化して訂正するように構成されるとよい。ECC エンジン 118 は、ECC エンジン 118 の誤り訂正容量内のいくつかの誤りを有する受け取られたデータを復号化するように構成されるとよい。例えば、誤り訂正容量は、符号化されたデータの 512 バイト毎に 12 ビットの誤りであってもよい。ECC エンジン 118 は、訂正されたデータ (すなわち、誤りがなく訂正の必要がないデータまたは ECC エンジン 118 によって訂正されたデータ) を、経路 120 を介して訂正済みバッファ 124 に伝達するように構成されるとよい。ECC エンジン 118 は、ECC エンジン 118 により訂正不能なデータを、経路 116 を介してバッファ 162 に伝達するように構成されるとよい。

20

【0011】

バッファ 162 は、ECC エンジン 118 により訂正不能なデータを格納するように構成されるとよく、訂正済みバッファ 124 は、ECC エンジン 118 により訂正されたデータを格納するように構成されるとよい。例えば、ワード線 108 が読み込まれると、ワード線 108 から読み込まれた第 1 のグループ 110 のビットは、ECC エンジン 118 により訂正不能でバッファ 162 にて格納されてもよく、ワード線 108 から読み込まれた第 2 のグループ 126 のビットは、ECC エンジン 118 によって訂正され、訂正済みバッファ 124 にて格納されてもよい。

30

【0012】

第 1 のグループ 110 は、訂正されていない 1 つ以上のページ 112 を含むとよく、第 2 のグループ 126 は、訂正された 1 つ以上のページ 128 を含むとよい。例示として、各 MLC 144 は、3 ビット値に対応する状態を有し、各 MLC 144 の下位ビットが第 1 の論理ページに割り当てられ、中位ビットが第 2 の論理ページに割り当てられ、上位ビットが第 3 の論理ページに割り当てられてもよい。各ページは、個々に符号化され、独立して ECC エンジン 118 にて処理されてもよい。ワード線 108 の 1 つ以上のページは、ECC エンジン 118 により訂正不能で、訂正されていないページ 112 としてバッファ 162 に格納されることもある一方で、ワード線 108 の他の 1 つ以上のページは、ECC エンジン 118 によって訂正され、訂正済みバッファ 162 に訂正されたページ 128 として格納されることもある。

40

【0013】

訂正モジュール 130 は、経路 140 を介して、バッファ 162 にある第 1 のグループ 110 のビットにアクセスするように構成され、第 1 のグループ 110 の 1 つ以上のビットを反転して、ビットの調整されたグループ 170 を生成するように構成されている。ビットの調整されたグループ 170 は、処理のために ECC エンジン 118 に提供される。ビットの調整されたグループ 170 が ECC エンジン 118 によって訂正可能であれば、訂正および復号化されたデータが、ワード線 108 に対する読み込みアクセスの要求への応答として、外部デバイス 106 に提供されるとよい。ビットの調整されたグループ 170 が ECC エンジン 118 によって訂正可能でなければ、第 1 のグループ 110 の他の 1

50

つ以上のビットが、反復プロセスにおいて訂正モジュール130によって反転されて、ECCエンジン118により訂正可能な第1のグループ110のデータビットが与えられるとよい。

【0014】

訂正モジュール130は、第1のグループ110のビット値のカウント114と、ログファイル134から経路158を介して取得される、格納されている先行カウント(prior count)136とを比較することによって、第1のグループ110内の、訂正すべき特定の値のビットの数148を判断するように構成されるとよい。カウント114は、訂正モジュール130によって生成されても、コントローラ104内の他のハードウェアによって生成されてもよい。格納されている先行カウント136は、先行する書き込み動作中にワード線108に書き込まれるデータから生成されても、ワード線108から前に読み込まれてECCエンジン118によって訂正されたデータから生成されてもよい。

10

【0015】

例えば、カウント114は、訂正されていない第1のグループ110内のいくつかの1を示してもよく、格納されている先行カウント136は、すべての誤りが訂正された後に第1のグループ110内に存在するべきいくつかの1を示してもよい。カウント114が、格納されている先行カウント136を1上回る場合、第1のグループ110内の「0」値のビットが誤って「1」値に変化し、訂正モジュール130は、「0」値に反転する第1のグループ110内の「1」値のビットを選択するとよい。格納されている先行カウント136がカウント114を上回る場合、「1」値のビットが誤って「0」値に変化し、訂正モジュール130は、「1」値に反転する第1のグループ110内の「0」値のビットを選択するとよい。訂正モジュール130は、カウント114と、格納されている先行カウント136との差に等しくなるように、反転するビットの数148を選択するように構成されるとよい。カウント114および格納されている先行カウント136は、「1」値のカウントとして記載されているが、その代わりとして、カウント114および格納されている先行カウント136は、「0」値のカウントであってもよく、訂正モジュール130が、0のカウントの比較に基づいて、反転するいくつかのビットおよびビットの値を判断するように構成されてもよい。

20

【0016】

訂正モジュール130は、選択されたビットが誤っている可能性に従って、反転される第1のグループ110内の特定のビットを選択するとよい。例えば、訂正モジュール130は、誤り確率テーブル138のうちの少なくとも1つにアクセスして、訂正する第1のグループ110の少なくとも1つのビットを識別するように構成されてもよい。誤り確率テーブル138は、マルチビットセルの他のビットにおける誤り152が検出および/または訂正されたかどうかに基づき、マルチビットセル(例えば、MLC)のビットが誤っている可能性が高いかどうかに対応する、誤りデータ150を含む。

30

【0017】

例えば、MLCは、訂正されていないページ(単数または複数)112内の第1のビットの値および訂正されたページ(単数または複数)128内の第2のビットによって表される状態(例えば、しきい電圧)を有するとよい。MLCの状態は、ハードウェア障害(壊れたワード線または他の接続など)が原因で誤って読み込まれることも、他のイベント(隣接したワード線に対する書き込み動作によって生じるプログラム妨害など)によって間違いを含むこともある。訂正されたページ(単数または複数)128内でビット誤りをもたらすイベントは、訂正されていないページ(単数または複数)112内でもビット誤りをもたらす得る。図2~3に示されているように、特定のビットが誤っている可能性は、MLCの他のビットにおけるいくつかの誤り152、特定のビットの値、MLCの1つ以上の状態値160、1つ以上の他の要素またはその任意の組み合わせと相関関係にあることもある。

40

【0018】

訂正モジュール130は、第2のグループ126のどのビットが訂正されたかを示す訂

50

正データを、ECCエンジン118から受け取るように構成されるとよい。訂正モジュール130は、第2のグループ126の訂正されたビットと共通のMLCを共有する、第1のグループ110の各ビットを発見するように構成されるとよい。訂正モジュール130は、第1のグループ110内の発見された各ビットに関して、誤り確率テーブル138におけるルックアップ動作を開始して、ビットが誤っている可能性を示すデータを取得するとよい。

【0019】

例えば、訂正モジュール130は、経路142を介して訂正済みバッファ124にアクセスして、第1のグループ110の特定のビットとMLCを共有している第2のグループ126内の1つ以上のビットのビット値を取得するとよい。訂正モジュール130は、取得したビット値(単数または複数)を処理し、MLCのいくつかの訂正されたビット、MLCの1つ以上のビット値またはその他の情報を使用して、誤り確率テーブル138にてルックアップ動作を実行し、誤り確率テーブル138にてエントリを発見するとよい。誤り確率テーブル138にて発見されたエントリの誤り可能性値が訂正モジュール130に返されて、第1のグループ110の特定のビットに関連付けられるとよい。

10

【0020】

訂正モジュール130は、最も高い誤り可能性値を有すると示されたビットを選択することによって、第1のグループ110内で訂正する、判断された数148のビットを選択してもよい。例えば、訂正するビットの数148がNであると判断されると、N個の最も高い誤り可能性値に対応するNビットが選択されるとよい。訂正モジュール130は、第1のグループ110のビットを読み込み、選択されたビットを反転して、調整されたグループ170を生成し、ECCエンジン118における、調整されたグループ170の復号化を開始するとよい。

20

【0021】

動作中、ワード線108の読み込み動作が実行され得る。ECCエンジン118は、ワード線108の各論理ページを無事に復号化するとよく、各ページの訂正されたいくつかの誤りを示す訂正データを生成するとよい。訂正モジュール130は、訂正されたいくつかの誤りと、誤りのしきい値数とを比較して、ワード線108またはワード線108の1つ以上のページが、後の読み込み動作において、ECCエンジン118の訂正能力を上回る危険があるかどうかを判断するとよい。

30

【0022】

訂正されたいくつかの誤りがしきい値を満たすのに応答して、訂正モジュール130は、ワード線108またはワード線108のページを1つ以上のグループに論理的に分けて、各グループの1のカウントをログファイル134に格納するとよい。したがって、ログファイルに格納されるカウントは、訂正され誤りのないデータの表現(すなわち、ECCエンジン118による訂正の成功の後)に対応する。グループの数は、1が0に変化した誤りおよび0が1に変化した誤りの両方が同じグループで生じる可能性が低くなるように判断されるとよい。例えば、ECCエンジン118の訂正容量が、512バイトページ毎に12ビットであれば、各ページは、12以上のグループに論理的に分けられてもよい。後の読み込み動作中、ページ内の12を超える誤りにより、ページがECCエンジン118により訂正不能である場合、各グループは、統計的に、1つのみの誤りを有する可能性が高いとよい。ログファイル134は、コントローラ104内に格納されるものとして示されているが、代わりにメモリアレイ102にて格納されてもよい。

40

【0023】

ワード線108またはワード線108の1つ以上のページに関して、各グループ内で特定の値を有するビットの数に対応するカウントを格納した後、ワード線108におけるその後の読み込み動作が、ECCエンジン118の訂正容量よりも多い数の誤りを有するデータのページを返し得る。ECCエンジン118は、未収束誤り信号(non-convergence error signal)などの信号を生成するとよく、それに応答して、読み込まれたデータの訂正不能な1つ以上のページがバッファ162にて格納されるとよい。ワード線108の訂正

50

されたページは、訂正済みバッファ 1 2 4 にて格納されるとよい。訂正されたデータに対応する誤り位置データも、E C C エンジン 1 1 8 によって生成されるとよく、訂正済みバッファ 1 2 4 にて格納されるとよい。

【 0 0 2 4 】

訂正モジュール 1 3 0 は、格納されている先行カウント 1 3 6 など、訂正不能なページに対応するデータのグループ内にある 1 の、格納されているカウントを取得するとよい。訂正モジュール 1 3 0 は、各グループの最近のカウント(カウント 1 1 4 など)と、グループの格納されている先行カウントとを比較することによって、バッファ 1 6 2 にて格納されているデータの個々のグループに関して、反転するいくつかのビットおよびビット値を判断するとよい。訂正される 1 つ以上のビット誤りを有すると示される各グループに関して、誤り訂正モジュール 1 3 0 は、誤り確率テーブル 1 3 8 にアクセスして、確率ルックアップテーブル結果に基づき誤っている可能性が最も高いと示されるビットを選択することによって、反転する 1 つ以上のビットを選択するとよい。

10

【 0 0 2 5 】

訂正モジュール 1 3 0 は、調整されたグループ 1 7 0 と、調整されたグループ 1 7 0 を含めた復号化動作を開始するための E C C エンジン 1 1 8 に対する信号とを生成するとよい。例えば、訂正モジュール 1 3 0 は、各グループの選択されたビットが反転されていてバッファ 1 6 2 内のデータに対応する、データの調整されたページを生成してもよい(すなわち、調整されたページは、1 つ以上の調整されたグループをページ形式で含む)。データの調整されたページは、E C C エンジン 1 1 8 に送られるとよい。別の例として、訂正モジュール 1 3 0 は、選択されたビットをバッファ 1 6 2 にて反転して、復号化のためにバッファ 1 6 2 を読み込むように E C C エンジン 1 1 8 に指示してもよい(例えば、調整されたグループ 1 7 0 はバッファ 1 6 2 内に生成される)。

20

【 0 0 2 6 】

十分な数の誤りビットが反転されると、調整されたデータ内の誤りの残数は、E C C エンジン 1 1 8 の誤り訂正容量内であるとよく、復号化が完了されるとよい。E C C エンジン 1 1 8 は、調整されたデータ内の、訂正された誤りの位置を示すデータを生成するとよい。訂正モジュール 1 3 0 は、誤り(モジュール 1 3 0 によって反転されたビットおよび E C C エンジン 1 1 8 によって訂正されたビット)の総数、誤りの位置、誤り値またはその組み合わせに基づき、ページのグループへの論理分割を更新するとよい。訂正モジュール 1 3 0 は、新たなグループに関して 1 の値の新たなカウントを生成して、ワード線 1 0 8 の次の読み込みで使用されるように、グループ分け情報およびカウント情報をログファイル 1 3 4 に格納するとよい。

30

【 0 0 2 7 】

結果として、ワード線 1 0 8 から読み込まれるデータが E C C エンジン 1 1 8 によって訂正可能でなくなった後でも、ワード線 1 0 8 の使用が続けられるとよい。したがって、誤り訂正モジュール 1 3 0 を使用せず、E C C エンジン 1 1 8 の訂正容量によって限定されるシステムと比較して、メモリアレイ 1 0 2 が、より長い有効寿命を有するとよい。あるいは、またはさらに、コントローラ 1 0 4 は、性能がより低い E C C エンジン 1 1 8 とともに訂正モジュール 1 3 0 を使用して、より性能の高い E C C エンジンの有効誤り訂正容量を保つように設計されてもよく、より性能の高い E C C エンジンと比較して削減された電力消費、デバイスサイズおよび製造費で運用することもできる。

40

【 0 0 2 8 】

図 2 は、誤り識別および訂正のために分析されるメモリセルの複数の変形を示す。メモリセルは、図 1 のメモリアレイ 1 0 2 の M L C 1 4 4 などの M L C を含むとよい。図 2 は、メモリアレイから読み込まれ得る、メモリセルの種々の変形 2 0 0、2 5 0、および 2 9 0 を示す。説明を簡単にするために、メモリセルは、最上位ビット、中位ビット、および最下位ビットを含む列形式に配列された 3 つのビットを含むように示されている。最上位ビットは、第 1 のページの一部であってもよく、中位ビットは、第 2 のページの一部であってもよく、最下位ビットは、第 3 のページの一部であってもよい。第 2 および第 3 の

50

ページは、ECCエンジンを使用して訂正されてもよく、第1のページは、ECCエンジンによって訂正不能であってもよい。

【0029】

セルの変形200は、最上位、中位および最下位ビット値の異なる組み合わせを有する、異なる状態202、204、206、208、210、212、214および216を含む。各状態202~216は、メモリセルの異なるしきい電圧範囲に対応するとよい。メモリセルは、初めは、状態208に対応する「101」として読み込まれてもよい。状態208は、「前」値として格納されるとよい。

【0030】

メモリセルの最上位ビットを含む第1のページは、訂正不能な誤りを含むとよい。セルの変形250および290は、第2および第3のページがECCエンジンによって訂正された後に生じ得る2つのシナリオを示す。例えば、セル変形250は、状態202~216を含む。セル変形250の訂正された状態は、「100」を含むとよく、これは状態210に対応する。したがって、第3のページの訂正の結果、第3のビットが「1」から「0」に変化し、第2のビットは変更されない。状態210は、「後」値として格納されるとよい。208から210への状態の変化は、1の状態ポジションの移動を表すことができる。矢印218は、「前」状態から「後」状態への移動を示す。移動は、1のハミング距離に対応する。

【0031】

対照的に、セル変形290の訂正された状態は、「110」を含むとよく、これは状態216に対応する。したがって、変形290では、第3のページの訂正の結果、第3のビットが「1」から「0」に変化し、第2のページの訂正の結果、第2のビットが「0」から「1」に変化する。状態216は、「後」値として格納されるとよい。208から216への状態の変化は、2の状態ポジションの移動を表すとよい。矢印220は、「前」状態から「後」状態への移動を示す。移動は、2のハミング距離に対応する。

【0032】

ハミング距離の値は、第1のビットが誤っている確率を判断するために、誤り確率テーブルをアドレス指定するための入力として使用され得る。例えば、セル変形250に対応する1のハミング距離が、図1の誤り確率テーブル138などの誤り確率テーブル内の値をルックアップするために使用され得る。図1のカウント114など、セルを含むグループの1のカウントも、ルックアップ要素として使用されてもよい。図1のコントローラ104などのコントローラは、ECC計算（例えば、1のハミング距離）およびカウントの誤り確率テーブル入力を使用して、セルの訂正された状態は「000」であるべきであると判断してもよい。矢印222は、正しい状態212への、セル状態の生じ得る反転を示す。

【0033】

別の例では、セル変形290に対応する2のハミング距離が、図1の誤り確率テーブル138などの誤り確率テーブル内の値をルックアップするために使用され得る。誤り確率テーブルは、図1のカウント114など、セルを含むグループの1のカウントも、ルックアップ要素として使用してもよい。図1のコントローラ104などのコントローラは、ECC計算（例えば、2のハミング距離）およびカウントを含む誤り確率テーブル入力を生成して、セルの訂正された状態が「010」であるべきであると判断してもよい。矢印224は、正しい状態214への、最上位ビットの生じ得る反転を示す。

【0034】

したがって、セル変形250および290は、ECC計算および確率ルックアップテーブルを使用して誤りビットを反転し、訂正されている状態を有する、メモリセルの例を示す。図1の誤り確率テーブル138などの誤り確率テーブルに提供されるデータは、読み込まれた値から部分的に訂正された値へのハミング距離によってインデックス付けされてもよい。比較的小さな処理要件を備えるECC方式が使用されてもよく、誤りを迅速に識別可能である。最終的に判断された訂正された状態が、データの完全性を改善するために

10

20

30

40

50

使用されるとよい。

【 0 0 3 5 】

図 3 は、データの誤りを識別して訂正するように構成されたシステム 3 0 0 の別の特定の
実施形態を示す。システム 3 0 0 は、図 1 のシステム 1 0 0 に類似しているが、格納され
ているカウント 3 1 0 および誤り確率テーブル 3 1 2 がメモリアレイ 3 0 2 内に含まれ
ている。図 3 は、メモリアレイ 3 0 2 内に格納されている、第 1 の格納されているカウ
ント（例えば、格納されているカウント 3 1 0）を示す一方で、図 1 は、コントローラ 1 0
4 内に格納されている、第 1 の格納されているカウント（例えば、格納されている先行カ
ウント 1 3 6）を示す。図 3 に示されているように、メモリアレイ 3 0 2 は、コントロー
ラ 3 0 4 に連結され、ワード線 3 0 8 を含む。コントローラ 3 0 4 は、図 1 のコントロー
ラ 1 0 4 に類似した機能性を含むとよい。コントローラ 3 0 4 は、外部デバイス 3 0 6 に
連結されている。外部デバイス 3 0 6 は、図 1 の外部デバイス 1 0 6 と同じであってもよ
い。

10

【 0 0 3 6 】

メモリアレイ 3 0 2 は、誤り確率テーブル 3 1 2 を含む。誤り確率テーブル 3 1 2 は、
図 1 の誤り確率テーブル 1 3 8 と同じであってもよい。図 3 に示されている誤り確率テ
ーブルは、ルックアップテーブル 3 5 0、3 5 2、3 5 4 および 3 5 6 を含む。

コントローラ 3 0 4 は、訂正モジュール 3 1 4、バッファ 3 1 6 および E C C エンジン
3 1 8 を含むとよい。訂正モジュール 3 1 4 は、図 1 の訂正モジュール 1 3 0 と同じであ
ってもよい。バッファ 3 1 6 は、図 1 のバッファ 1 6 2 および訂正済みバッファ 1 2 4 に
対応するとよい。E C C エンジン 3 1 8 は、図 1 の E C C エンジン 1 1 8 と同じであって
もよい。

20

【 0 0 3 7 】

ルックアップテーブル 3 5 0 は、M L C の、検出された他のいくつかの誤りに基づき、
M L C の特定のビットが誤っている確率を判断するために使用され得る。例えば、M L C
が、異なるページにそれぞれ対応する 3 ビット値を格納し、2 つのページが訂正可能で
あり、第 3 のページは訂正可能でない場合、第 3 のページ内の特定のビットにおける誤り
の確率は、M L C の他の 2 つのビットにおいて検出されたいくつかの誤りと相関関係にある
かもしれない。

【 0 0 3 8 】

ルックアップテーブル 3 5 0 は、経路 3 4 6 によって示されているように、第 1 の格納
されているカウント 3 1 0 と、第 2 の読み込みのカウントとの差が 1 と等しく、特定のビ
ット値が 1 と等しい場合に使用されるとよい。ルックアップテーブル 3 5 2 は、3 4 6 に
て示されているように、第 1 の格納されているカウント 3 1 0 と、読み込みのカウントと
の差が 1 と等しく、特定のビット値が 0 と等しい場合に使用されるとよい。ルックアップ
テーブル 3 5 2 は、M L C 内のいくつかの訂正された誤りを示す列を含む。この列は、ビ
ット誤りの確率を示す列と相互に関連付けられている。

30

【 0 0 3 9 】

例えば、M L C において 0 の誤りが検出された場合、特定のビットが誤っている確率は
、「 A 」と示されている値に対応するとよい。1 つの誤りが M L C において検出された場
合、特定のビットが誤っている確率は、「 B 」と示されている値に対応するとよい。もう
1 つの例では、2 つの誤りが M L C において検出された場合、特定のビットが誤っている
確率は、「 C 」と示されている値に対応するとよい。例えば、値「 A 」は、値「 B 」より
も小さいとよく、値「 B 」は、値「 C 」よりも小さいとよい。

40

【 0 0 4 0 】

ルックアップテーブル 3 5 4 は、3 4 8 にて示されているように、第 1 の格納されてい
るカウント 3 1 0 と、読み込みのカウントとの差が 1 より大きく、特定のビット値が 1 と
等しい場合に使用されるとよい。ルックアップテーブル 3 5 6 は、3 4 8 に示されている
ように、第 1 の格納されているカウント 3 1 0 と、読み込みのカウントとの差が 1 より大
きく、特定のビット値が 0 と等しい場合に使用されるとよい。ルックアップテーブル 3 5

50

6 は、部分的に訂正された値を示す列を含む。この列は、ビット誤りの確率を示す列と相互に関連付けられている。

【 0 0 4 1 】

例示として、3ビットMLCにおいて、特定のビットが誤っている確率は、他の2つのビットの訂正された値に対応し得る。例えば、3ビットMLCの第1のビットの訂正された値および第2のビットがどちらも「0」である場合、特定のビットが誤っている確率は、「D」と示されている値に対応するとよい。第1のビットの訂正された値が「0」であり、第2のビットの訂正された値が「1」である場合、特定のビットが誤っている確率は、「E」と示されている値に対応するとよい。別の例では、第1のビットの訂正された値が「1」であり、第2のビットの訂正された値が「0」である場合、特定のビットが誤っている確率は、「F」と示されている値に対応するとよい。3ビットMLCの第1のビットの訂正された値および第2のビットがどちらも「1」である場合、特定のビットが誤っている確率は、「G」と示されている値に対応するとよい。テーブル354および356は、単一のMLCの正しいビット値および読み込まれたビット値に応じてビット誤りの可能性を示すため、テーブル354および356は、MLCの状態に基づき（例えば、MLCのしきい電圧に基づき）誤りの可能性を反映するとよい。

10

【 0 0 4 2 】

動作中、メモリアレイ302から読み込まれたデータビットがECCエンジン318により訂正不可能な誤りを含むという通知を、コントローラ304が受け取ることもある。コントローラ304は、誤り確率テーブル312のルックアップ値に基づき、かつ格納されているカウント310に基づき、訂正モジュール314がデータビットの値を変更できるようにするとよい。コントローラ304は、ECCエンジン318がデータビットを訂正できるまで、反復してビットを反転するとよい。

20

【 0 0 4 3 】

図3は、ルックアップテーブル350および352をいくつかの誤りに基づき示し、ルックアップテーブル354および356を、部分的に訂正された値に基づき示すが、他の実施形態では、誤りの確率を判断または推定するための根拠として、他の基準が使用されてもよい。例えば、誤り確率テーブル312すべてが、部分的に訂正された値によってインデックス付けされてもよく、または誤り確率テーブル312すべてが、検出されたいくつかの誤りによってインデックス付けされてもよい。別の例として、誤り確率テーブル312は、第1の格納されているカウント310と読み込みのカウントとの差から独立していてもよい。さらに、誤り確率テーブル312は、3ビット値に対応するように示されているが（例えば、MLCの8つの状態）、別の実施形態の、メモリセルの状態によって表されるいくつかのビットは、3ビット未満でも3ビット超でもよい。

30

【 0 0 4 4 】

示されている値「A」、「B」などの1つ以上は、経験的データに基づき、理論的計算に基づき、計算の単純さおよびテーブルのストレージサイズなどの他の1つ以上の要素に基づき、またはその任意の組み合わせに基づき判断されてもよい。よって、示されている値「A」、「B」などは、ビット誤りの確率に対応するように記載されているが、他の実施形態では、示されている値「A」、「B」などの1つ以上は、実際の確率には対応しなくてもよく、代わりに、誤りの相対的可能性を示しても、生じ得る誤りに関してMLCを検査する優先順位の体系を示してもよい。

40

【 0 0 4 5 】

図4は、メモリセル内の誤りを分析して訂正する方法の特定の実施形態の流れ図である。方法は、概して400と示され、402にてワード線を読み込むことを含む。例えば、図1のコントローラ104が、メモリアレイ102からワード線（単数または複数）108を読み込んでよい。

方法400は、404にて、訂正不能な誤りを含むページを読み込むことを含むとよい。例えば、図1のコントローラ104が、読み込み動作146中にメモリアレイ102からページ112を読み込むとよい。

50

406にて、読み込まれたセルの現在の状態が保存されるとよい。この現在の状態は、「前」状態とラベル付けされるとよい。例えば、図1の第1のグループ110およびカウント114が、バッファ162内に格納されるとよい。

408にて、ワード線のコピーが作成されるとよく、訂正可能なページ上の誤りが訂正されるとよい。訂正されたデータのコピーは、「後」とラベル付けされるとよい。例えば、図1のECCエンジン118は、ワード線(単数または複数)108の他のページ上の誤りを訂正して、訂正されたデータをページ(単数または複数)128として格納するとよい。

方法400は、410にて、訂正されて別の状態に移行されたセルを識別することを含むとよい。例えば、図1の訂正モジュール130は、訂正されて別の状態に移行された、ページ(単数または複数)128内のMLC144を識別するとよい。訂正されたセルに関する情報は、ECCエンジン118によって訂正モジュール130に提供されるとよい。

10

【0046】

412にて、反転する特定のセルが分析されるとよい。例えば、図1の訂正モジュール130は、ECCエンジン118が訂正できないビットのグループの、訂正するべきいくつかのビットを判断することによって、MLC144を分析するとよい。訂正モジュール130は、確率テーブルにおいてルックアップ動作を行って、誤っている可能性が最も高いMLCビットを判断するとよい。訂正モジュール130は、判断されたビットを反転して、データをECCエンジン118に送るとよい。ビットのグループがまだ訂正不能であれば、訂正モジュール130は、次に反転されるビットを、次のビットに関連する、判断された確率に基づき選択するとよい。

20

【0047】

図5は、メモリセル内の誤りを訂正する方法の実施形態の流れ図である。方法は、概して500と示され、図1のコントローラ104など、メモリアレイに連結されたコントローラによって実行されてもよい。方法500は、502にて、メモリアレイから読み込まれた第1のグループのデータビットが、ECCエンジンにより訂正不能な誤りを含むという通知を受け取ることを含む。例えば、図1のコントローラ104は、読み込み動作146から、第1のグループ110が、ECCエンジン118により訂正不能な誤りを含むと判断してもよい。例示として、図1の訂正モジュール130は、ECCエンジン118が第1のグループ110の1つ以上のページ(単数または複数)112を訂正できなかったことを示す信号を、ECCエンジン118から受け取ってもよい。訂正モジュール130はさらに、ECCエンジン118によって無事に訂正された、第2のグループ126の1つ以上のページ(単数または複数)128で生じている誤りを示す信号を、ECCエンジン118から受け取ってもよい。

30

【0048】

504にて、特定のビット値を有する第1のグループのデータビットのカウントが、特定のビット値を有するデータビットの先行カウントと比較されるとよい。例えば、図1の訂正モジュール130は、通信経路140を介してカウント114を受け取ってもよい。訂正モジュール130は、ログファイル134にアクセスし、第1のグループ110に対応する格納されている先行カウント136を発見し(例えば、第1のグループ110のメモリアドレスと、格納されている先行カウント136に対応するメモリアドレスとを比較することによって)、格納されている先行カウント136をログファイル134から取得することなどによって、格納されている先行カウント136を受け取ってもよい。訂正モジュール130は、減算演算、ビット単位比較演算、またはカウント114と格納されている先行カウント136との差を判断する別の演算を実行することなどによって、カウント114と、格納されている先行カウント136とを比較するとよい。特定のビット値は、「0」であってもよく、カウント114は、第1のグループ110のいくつかの「0」値に対応してもよい。あるいは、特定のビット値は、「1」であってもよく、カウント114は、第1のグループ110のいくつかの「1」値に対応してもよい。

40

50

【 0 0 4 9 】

第1のグループにおいて反転するべきいくつかのビットが、506にて判断されるとよい。例えば、図1の訂正モジュール130は、格納されている先行カウント136からカウント114を減算して、訂正するビットの数148として減算の結果を代入する結果として、訂正するビットの数148を判断してもよい。負値は、1つ以上のビットが、特定の値を有するように反転されるべきであることを示してもよく、正值は、特定の値を有する1つ以上のビットが反転されるべきであることを示してもよい。あるいは、他の1つ以上のプロセスが使用されて、カウント114と格納されている先行カウント136との差の指標および大きさが判断されてもよい。

【 0 0 5 0 】

カウントが先行カウントを上回るという判断にตอบสนองして、第1のグループのデータビットのうち、特定のビット値を有し第2のグループのデータビットの訂正されたデータビットと同じメモリセルに対応するビットが、508にて識別されるとよい。例えば、図1のコントローラ104は、カウント114が格納されている先行カウント136を上回るのにตอบสนองして、第1のグループ110の或るビットが、第2のグループ126の訂正されたビットと同じメモリセルに対応すると識別してもよい。

【 0 0 5 1 】

例示として、ECCエンジン118は、ECCエンジン118によって訂正された第2のグループ126のビットを示すデータを生成するとよい。訂正モジュール130は、ECCエンジン118からこのデータを受け取るとよく、第2のグループ126の訂正されたビットに対応する、第1のグループ110の1つ以上のビットを発見するとよい。例えば、MLC(単数または複数)144は、別々のページ上の同じビット位置に各ビット値が関連する、3ビット値に対応する状態を有してもよい(例えば、図2に関して記載したように)。ページ(単数または複数)128のうち1つ以上にある訂正された誤りのビット位置が識別されて、訂正された誤りのビット位置にあるページ(単数または複数)112内のビットが、訂正された誤りと同じMLCに対応するものとして識別されてもよい。他の実施形態では、訂正モジュール130は、同じMLC144に関連する第1および第2のグループ110、126の対応するビット間の、他の1つ以上のマッピングに従って、第1のグループ110のビットを識別してもよい。

【 0 0 5 2 】

第2のグループ126内の訂正された誤りのMLCに対応する、第1のグループ110の複数のビットが発見されるとよい。図1の訂正モジュール130は、誤り確率テーブル138にて複数のルックアップ動作を実行するとよい。各ルックアップ動作は、第1のグループ110の対応するビットが誤っている可能性値または確率を返すとよい。訂正モジュール130は、第2のグループ126内の訂正された誤りと同じMLCに対応する第1のグループ110の発見されたビットそれぞれのルックアップ動作を実行して、ルックアップ動作によって返された最大値1つ以上を識別するとよい。返された最大値に関連する、第1のグループ110のビット(例えば、確率テーブル138によって誤っている可能性が最も高いと示されたビット)が識別されるとよい。

【 0 0 5 3 】

例えば、訂正モジュール130は、第2のグループ126内の訂正された誤りに対応する第1のグループ110内のビット位置を含み、ビット位置のそれぞれに関してルックアップテーブルの結果を含む、エントリのテーブルまたはバッファを生成するとよい。各ルックアップテーブルは、図3で示されたものなど、インデックス付けデータを判断することを含んでもよい。例えば、訂正モジュール130は、同じMLCに対応する、第1のグループ110内のビット位置と、第2のグループ126のビットとを比較するとよい。訂正モジュール130は、ECCエンジン118によって生成されたデータに基づき、第2のグループ126内の対応するビット(同じMLCの)のいくつかが発見されたかを判断するとよく、ルックアップ動作中に、図3のテーブル350または352などの確率テーブルに、結果を供給するとよい。もう1つの例として、訂正モジュール130は、訂正済み

10

20

30

40

50

バッファ 1 2 4 にアクセスして、第 2 のグループ 1 2 6 の対応するビット (同じ M L C の) の訂正されたビット値を読み込むとよく、ルックアップ動作中に、図 3 のテーブル 3 5 4 または 3 5 6 などの確率テーブルに、訂正されたビット値を提供してもよい。

【 0 0 5 4 】

この例の続きとして、ビット位置に対応するエントリと、ルックアップテーブルの関連する結果とによりテーブルまたはバッファをポピュレートした後、最大のルックアップテーブル結果が発見されるとよい。例えば、ルックアップテーブル結果値に基づき結果がソートされてもよく、ソート後の第 1 のエントリが識別されてもよい。別の例として、テーブルまたはバッファがトラバースされてもよく、最大のルックアップテーブル結果値が発見されてもよい。この例では、最大のテーブル結果値に基づき単一ビットが識別されるが、他の実施形態では、複数のビット (すなわち、N ビット) が、発見される N 個の最大のルックアップテーブル結果値に基づき識別されてもよい。例えば、N の値は、反転されるビットの数と等しくてもよい。

10

【 0 0 5 5 】

5 1 0 にて、第 1 のグループの識別されたビットの値が変更されて、データビットの調整されたグループが生成されるとよい。例えば、訂正モジュール 1 3 0 が、識別されたビットを変更して、調整されたグループを生成してもよい。例示として、訂正モジュール 1 3 0 は、バッファ 1 6 2 にアクセスして、第 1 のグループ 1 1 0 からビット値のコピーを生成するとよい。訂正モジュール 1 3 0 は、ビット値のコピー内で、第 1 のグループ 1 1 0 の識別されたビットに対応するビット位置を発見するとよい。識別されたビットが「0」値を有する場合、訂正モジュール 1 3 0 は、「0」値を「1」値に置き換えて、データビットの調整されたグループ 1 7 0 を生成するとよい。識別されたビットが「1」値を有する場合、訂正モジュール 1 3 0 は、「1」値を「0」値に置き換えて、データビットの調整されたグループ 1 7 0 を生成するとよい。複数のビットが識別された場合、訂正モジュール 1 3 0 は、識別された各ビットに対して置き換え動作を実行して、データビットの調整されたグループを生成するとよい。

20

【 0 0 5 6 】

識別されたビットは、5 1 2 にて、ルックアップテーブルに基づき選択されるとよい (例えば、ハミング距離、カウントまたは状態値の 1 つ以上に少なくとも部分的に基づいて)。例示として、図 3 のテーブル 3 5 0 および 3 5 2 に関して記載したように、誤りの可能性を判断するために M L C 内のいくつかの誤りが使用される場合、ルックアップテーブルは、ハミング距離に少なくとも部分的に基づくとよい。別の例として、図 3 の経路 3 4 6 および経路 3 4 8 に関して記載したように、ルックアップテーブルは、アクセスする確率テーブルを選択するためのパラメータとしてカウントを使用することによって、カウントに少なくとも部分的に基づいてもよい。さらに別の例として、図 3 のテーブル 3 5 4 および 3 5 6 に関して記載したように、ルックアップテーブルは、特定のビット値に関連して、M L C セルの部分的に訂正されたビット値に基づくなど、状態値に少なくとも部分的に基づいてもよい。識別されたビットは、発見されたビットのうちの、最大のルックアップテーブル結果を有するものとして選択されてもよい。

30

【 0 0 5 7 】

データビットの調整されたグループは、5 1 4 にて、E C C エンジンに提供されるとよい。例えば、図 1 の訂正モジュール 1 3 0 は、経路 1 2 2 を介して、調整されたグループ 1 7 0 を E C C エンジン 1 1 8 の入力に伝達してもよい。訂正モジュール 1 3 0 は、データビットの調整されたグループ 1 7 0 を入力データとして使用する復号化動作を開始するために、1 つ以上の制御信号を、生成するか、またはその他の形で E C C エンジン 1 1 8 に提供させるとよい。E C C エンジン 1 1 8 は、復号化動作を開始するとよく、訂正された出力または失敗インジケータを生成するとよい。E C C エンジン 1 1 8 が、訂正された出力を生成すれば、E C C エンジン 1 1 8 の訂正能力に適合するのに十分な数の第 1 のグループ 1 1 0 の誤りが、調整されたグループ 1 7 0 内で訂正されたことになる。

40

【 0 0 5 8 】

50

失敗インジケータを受け取ると、516にて、第1のグループが訂正可能であるとECCエンジンが判断するまで、第1のグループ内のビットが反復して反転されるとよい。例えば、データビットの調整されたグループに対するECC動作の結果、失敗インジケータが図1の訂正モジュール130に提供されることもある。それに応答して、訂正モジュール130は、反転する追加または代替りのビットの1つ以上を選択して、ECCエンジン118を使用して評価される、データビットの別の調整されたグループを生成するとよい。

【0059】

例えば、訂正モジュール130が、反転する単一のビットを反復毎に識別する場合(すなわち、単一の「1」値を「0」値に置き換えるか、または単一の「0」値を「1」値に置き換える)、訂正モジュール130は、直前に反転されたビットをその元の値に戻して、第1のグループ110の、誤りビットである可能性が次に高いものを反転して、ECCエンジン118にてテストされる、データビットの次の調整されたグループを生成するとよい。誤りビットである可能性が次に高いものは、前述したものなどのルックアップテーブル結果に従って識別されるとよい。結果として生じる、データビットの次の調整されたグループが、反復プロセスの次の反復としてECCエンジン118に提供されるとよく、任意の特定の反復において、ECCエンジンがデータビットの調整されたグループを訂正できれば、反復プロセスを終了してもよい。反復プロセスの他の終了基準には、発見されたすべてのビットがテストされ、結果として生じる調整されたグループはいずれも復号化できなかった、第1のグループ110のすべてのビット(すなわち、第2のグループ126の訂正された誤りと同じセルに対応しないビットを含む)がテストされ、結果として生じる調整されたグループはいずれも復号化できなかった、所定数の反復が完了した、その他1つ以上の終了基準、またはその任意の組み合わせが含まれ得る。

【0060】

図6は、ルックアップテーブルに基づきメモリセル内の誤りを訂正する方法の実施形態の流れ図である。方法は、概して600と示され、図1のコントローラ104など、メモリアレイに連結されたコントローラにて実行されてもよい。方法600は、602にて、ビットのグループのうち、特定のビット値を有すると識別されたビットの第1のカウントを格納することを含み、ビットのグループは、メモリアレイのワード線の少なくとも一部に対応する。例えば、図1のカウント114は、ワード線108に対応してもよく、ワード線108のデータ読み込みがECCエンジン118により訂正不能な場合に後で取得するために、格納されている先行カウント136としてログファイル134内に格納されるとよい。特定のビット値は、「0」であってもよく、バッファ162内に格納されるカウント114は、第1のグループ110のいくつかの「0」値に対応してもよい。あるいは、特定のビット値は、「1」であってもよく、バッファ162内に格納されるカウント114は、第1のグループ110のいくつかの「1」値に対応してもよい。図1では、バッファ162がカウント114を含むように示されているが、代わりにカウントは、訂正モジュール130または同等の論理デバイスと通信している任意の適切なメモリ内に格納されてもよい。

【0061】

ビットの第1のカウントは、コントローラにて格納されても、メモリアレイにて格納されてもよい。図1では、格納されている先行カウント136は、ログファイル134内にあるように示されているが、他の実施形態では、格納されているカウントは、ログファイル内に格納されていなくてもよい。図1は、コントローラ104にて格納された、格納されている先行カウント136を示すが、他の実施形態では、格納されているカウントは、コントローラにて格納されず、代わりにメモリアレイ102にて格納されてもよい。例えば、第1のカウントは、メモリアレイ内のログファイルに格納されてもよい。

【0062】

604にて、ワード線が、誤りのしきい値数を上回るいくつかの誤りを有するという通知を受け取られてもよく、通知を受け取るのに応答して、カウントが格納されてもよい。

10

20

30

40

50

例えば、図1のコントローラ104は、ワード線108が過剰な数の誤りを有するという通知をECCエンジン118から受け取るのに応答して、先行カウント136を格納してもよい。ECCエンジン118は、いくつかの誤りを示してもよく、訂正モジュール130は、ワード線108が、ECCエンジン118の訂正容量を近いうちに上回る可能性が高いと判断してもよい。訂正容量に近くなるとき、コントローラ104は、ワード線108内のデータビットの1つ以上のグループに関する、先行カウント136の格納を開始するとよい。メモリアレイ102またはコントローラ104内で、フラグまたは他の指示子がセットされ、ワード線108が、多数の誤りを有することまたは格納されているカウントに関連することを示してもよい。例として、図1の訂正モジュール130は、ワード線108の、しきい値を上回るいくつかの誤りがECCエンジン118によって検出されたことを示す信号を、ECCエンジン118から受け取ってもよい。ワード線108に対応する訂正されたデータは、データビットの1つ以上のグループに論理的に分割されてもよく、各グループに対してカウントが生成される。ワード線108に関連するカウントは、後の読み込み動作中にワード線108からデータが読み込まれるときに取得されるように、ログファイル134にて格納されるとよい。

10

【0063】

606にて、ワード線に関連するいくつかのカウントは、ワード線内のいくつかの誤りが増加するにつれて増加させられるとよい。例えば、図1のワード線108内の多数の誤りは、いくつかのカウント114を増加させるとよい。例示として、訂正モジュール130は、ECCエンジン118によって検出された誤りの数を受け取ってもよく、ワード線108のカウント114と格納されている先行カウント136との差に基づいて数を判断してもよい。訂正モジュール130は、誤りの数を使用して、バッファ162内に格納されるべきいくつかのカウントを判断してもよい。ワード線108内の誤りの数が増加するにつれて、いくつかのカウントは増加するとよい。いくつかのカウントを増加させることは、単一のグループ内の複数のセルの状態によって表されるビット値が誤りにより反転する確率を下げるるとよい。

20

【0064】

608にて、1が0に変化した誤りおよび0が1に変化した誤りのうちの1つのみがビットのグループ内に含まれるように、いくつかのカウントが判断されるとよい。例えば、図1のいくつかのカウント114は、1が0に変化した誤りおよび0が1に変化した誤りのうちの1つのみがグループ110内に含まれるように、ECCエンジン118によって検出された誤りの数に基づき判断されるとよい。例示として、ワード線108の読み込み動作146の結果、1つ以上のグループ(例えば、グループ110)が、グループの格納されているカウント(例えば、格納されているカウント136)とは2以上異なるカウントを有することもある。対応する格納されているカウントを2以上上回るカウントを有するグループの数、および格納されているカウントより2以上少ないカウントを有するグループの数が判断されて、グループの現在の数に追加され、ワード線108を論理的に分割するための多数のグループが判断されてもよい。各グループが「1」または「0」の関連するカウントを有し、実質的に等しいサイズのグループのデータビットにワード線108を分割することによって、ワード線108に対応するグループの新たなセットが判断されてもよい。別の例として、グループの格納されているカウントと2以上異なるカウントを有する各グループが、2つ以上のより小さいグループに分割されてもよく、その結果、ワード線108の不均一なグループサイズがもたらされるが、ワード線108のうち、他の部分よりも多くの誤りを有する部分において、より細かい粒度が提供される。

30

40

【0065】

610にて、ECCエンジンにより訂正不能なデータをもたらすワード線の読み込みに応答して、第1のカウントが取得されるとよい。例えば、ECCエンジン118には訂正不能な数の誤りを有するデータをもたらすワード線108の読み込み動作146に応答して、図1の格納されている先行カウント136が取得されてもよい。例えば、1つ以上のページ(単数または複数)112をECCエンジン118が訂正できなかったことを示す

50

信号を、図1の訂正モジュール130が、ECCエンジン118から受け取ってもよい。それに応答して、訂正モジュール130は、格納されている先行カウント136を取得するとよい。例示として、訂正モジュール130は、ログファイル134にアクセスし、第1のグループ110に対応する格納されている先行カウント136を発見してもよい。例えば、訂正モジュール130は、第1のグループ110のメモリアドレスと、格納されている先行カウント136に対応するメモリアドレスとを比較して、格納されている先行カウント136をログファイル134から取得してもよい。612にて、第1のカウントが、特定のビット値を有するビットの第2のカウントと比較されるとよく、ビットの第2のカウントは、ワード線から読み込まれたデータに基づき判断される。例えば、図1の格納されている先行カウント136が、カウント114と比較されてもよい。例示として、訂正モジュール130は、減算演算、ビット単位比較演算、またはカウント114と格納されている先行カウント136との差を判断する別の演算を実行することなどによって、カウント114と、格納されている先行カウント136とを比較してもよい。

【0066】

614にて、訂正するいくつかのビットが、比較から判断されるとよい。例示として、「1」値が「0」値になるように訂正されても、「0」値が「1」値に置き換えられてもよい。一例では、図1の訂正するビットの数148は、特定の値を有する取得されたビットのカウント114と、格納されている先行カウント136とを比較することによって判断されてもよい。例示として、図1の訂正モジュール130は、格納されている先行カウント136からカウント114を減算して、訂正するビットの数148として減算の結果を代入する結果として、訂正するビットの数148を判断してもよい。負値は、1つ以上のビットが、特定の値を有するように反転されるべきであることを示してもよく、正値は、特定の値を有する1つ以上のビットが反転されるべきであることを示してもよい。あるいは、他の1つ以上のプロセスが使用されて、カウント114と格納されている先行カウント136との差の指標および大きさが判断されてもよい。

【0067】

メモリアレイのセルの状態が、616にて訂正されるとよい。状態は、ルックアップテーブルに基づき訂正されてもよい。例えば、メモリアレイは、複数のマルチレベルセルを含んでもよく、メモリアレイ102のMLC144の状態を変更するためのルックアップテーブルを行うために、図1の誤り確率テーブル138が使用されてもよい。例示として、訂正モジュール130は、バッファ162にアクセスし、ルックアップテーブルを使用して、訂正する第1のグループ110のビットを識別してもよい。図3のテーブル350および352に関して記載したように、誤りの可能性を判断するためにMLC内のいくつかの誤りが使用される場合、ルックアップテーブルは、ハミング距離に少なくとも部分的に基づくとよい。別の例として、図3の346および348に関して記載したように、ルックアップテーブルは、アクセスする確率テーブルを選択するためのパラメータとしてカウントを使用することによって、カウントに少なくとも部分的に基づいてもよい。さらに別の例として、図3のテーブル354および356に関して記載したように、ルックアップテーブルは、特定のビット値に関連して、MLCセルの部分的に訂正されたビット値に基づくなど、状態値に少なくとも部分的に基づいてもよい。

【0068】

本願明細書に示された様々なコンポーネントは、ブロックコンポーネントとして示され、一般用語で記載されているが、そうしたコンポーネントは、1つ以上のマイクロプロセッサ、状態マシン、または図1のコントローラ104もしくは図3のコントローラ304がそうしたコンポーネントによる特定の機能を実行できるように構成された他の回路、またはその任意の組み合わせを含んでもよい。例えば、図1の訂正モジュール130、または図3のECCエンジン318および訂正モジュール314、またはその任意の組み合わせは、ハードウェアコントローラ、状態マシン、論理回路、または図1のコントローラ104もしくは図3のコントローラ304が誤りを検出および訂正できるようにする他の構造などの物理的コンポーネントを表し得る。

10

20

30

40

50

【 0 0 6 9 】

例えば、訂正モジュール 1 3 0 は、訂正するいくつかのビットを判断することによって M L C 1 4 4 を分析するようにプログラムされたマイクロプロセッサまたはマイクロコントローラを使用して実装されてもよい。例示として、コントローラ 1 0 4 などのマイクロプロセッサまたはマイクロコントローラが、反転する可能性が最も高い M L C ビットを判断するために、ルックアップ動作を行うようにプログラムされてもよい。コントローラ 1 0 4 は、判断されたビットを反転して、E C C 動作を試行してもよい。ビットを含むデータがまだ訂正不能な場合、コントローラ 1 0 4 は、反転される次のビットを、次のビットに関連する判断された確率に基づき選択するとよい。訂正モジュール 1 3 0 は、コントローラ 1 0 4 によって実行される実行可能な命令を含むとよい。訂正モジュール 1 3 0 に含まれるとよく、コントローラ 1 0 4 によって実行される実行可能な命令は、読み出し専用メモリ (R O M) (図示せず) などの個別のメモリ位置にて格納されてもよい。

10

【 0 0 7 0 】

本願明細書における開示では、様々な特定の値が使用されているが、そうした値は、限定ではなく説明の目的で提供されている。例えば、3 ビットを格納する M L C が記載されているが、2 ビットまたは 4 ビット以上を格納する M L C が使用されてもよい。別の例として、E C C 訂正能力は、5 1 2 バイトページ毎に 1 2 ビットとして記載されているが、他の E C C 訂正能力が使用されてもよい。例えば、E C C 訂正能力は、5 1 2 バイトワード毎に 1 2 ビットより大きくても、5 1 2 バイトワード毎に 1 2 ビットより少なくてもよく、5 1 2 バイト以外のワードサイズに基づいてもよい (例えば、2 5 6 バイト、1 , 0 2 4 バイトまたはその他の任意のワードサイズ) 。さらに、E C C 訂正能力は、説明に役立つ例として、E C C 方式 (例えば、リードソロモン対 B C H) および強度 (例えば、1 2 パリティビット / ワード対 1 6 パリティビット / ワード) によって異なってもよい。

20

【 0 0 7 1 】

図 3 のルックアップテーブル 3 1 2 などを通してビット値が誤っている確率または可能性を判断するものとして、様々な実施形態について記載してきたが、他の実施形態では、ビット値が正しい可能性または確率が使用されてもよい。例えば、特定のビットが正しい値を有する可能性またはその他のインジケータを示す 1 つ以上の確率テーブルが使用されてもよい。訂正されていないビットのグループ内の反転するべきビットの判断は、対応するビットが正しいことの最低の信頼度を示す、ルックアップテーブルから返される最小値を発見することを含んでもよい。

30

【 0 0 7 2 】

E C C エンジンにて独立して復号化が可能な複数のページを含むワード線、ビットのグループに論理的に分けられるページ、および識別されたページ内の各グループに関して格納される 1 (または 0) のカウントに関して、様々な実施形態が記載されている。しかし、他の実施形態では、ワード線は、単一のページを含んでもよく、E C C にて独立して復号化可能なデータの最小単位であってもよい (例えば、単一のコードワード) 。あるいは、またはさらに、論理グループへのワード線の分割が誤り密度に基づき、特定のページが比較的誤りがなく、結果としてページ全体が単一グループ内にある場合などは、ビットのグループ (格納されているカウントに対応する) が 1 つ以上のページを含んでもよい。格納されて後の読み込み動作中に使用される各グループの 1 (または 0) のカウントを生成するように、ページまたはワード線内のビットのグループ分けについて記載されているが、一部の実施形態では、ページまたはワード線の全部ではなく一部が、論理的にグループに分けられてもよい。例えば、ワード線またはページ (またはストレージの他の単位) の中央領域または終端領域内のビットは、グループおよび格納されているカウントに割り当てられてもよく、ワード線またはページの残りの領域 (単数または複数) 内のビットは、グループに割り当てられなくてもよい。

40

【 0 0 7 3 】

図 1 のメモリアレイ 1 0 2 に連結されているコントローラ 1 0 4 など、図 1 および 3 に示されているシステムは、外部デバイス 1 0 6 などの 1 つ以上の外部デバイスに選択的に

50

連結されるように構成された携帯デバイスとして実装されてもよい。一方、他の実施形態では、図1のコントローラ104およびメモリアレイ102は、携帯通信デバイスの筐体内など、1つ以上のホストデバイス内に取り付けられていても、組み込まれていてもよい。例えば、図1のコントローラ104およびメモリアレイ102は、無線電話、携帯情報端末(PDA)、ゲーム用デバイスまたはコンソール、携帯ナビゲーションデバイス、または内部不揮発性メモリを使用する他のデバイスなどのパッケージ化装置内であってもよい。メモリアレイ102は、フラッシュメモリ(例えば、NAND、NOR、マルチレベルセル(MLC)、分割ビット線NOR(DINOR: Divided bit-line NOR)、AND、高容量結合比(HiCR: high capacitive coupling ratio)、非対称無接点トランジスタ(ACCT: asymmetrical contactless transistor)、またはその他のフラッシュメモリ)、消去可能でプログラム可能な読み出し専用メモリ(EPROM)、電氣的に消去可能でプログラム可能な読み出し専用メモリ(EEPROM)、読み出し専用メモリ(ROM)、ワンタイムプログラマブルメモリ(OTP)、またはその他任意のタイプのメモリなどの不揮発性メモリを含むとよい。

【0074】

本願明細書に記載した実施形態の説明は、様々な実施形態の全般的な理解を提供することを目的としている。本願明細書における開示から、他の実施形態が利用され、得られることが可能であり、本願明細書における開示の範囲から逸脱することなく、構造上および論理上の置き換えおよび変更が行われ得る。本願明細書における開示は、様々な実施形態の、後のあらゆる適応および変形を対象とするものとする。

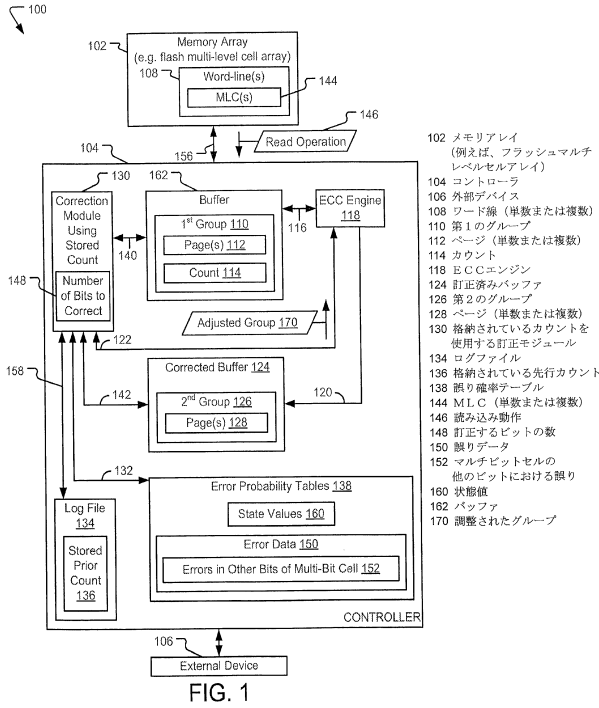
【0075】

前に開示した主題は、制限ではなく例示と見なされるものとし、添付の特許請求の範囲は、この開示の範囲内に入るそうした変更、改良および他の実施形態すべてを対象とするものとする。したがって、許可される最大の範囲まで、本発明の範囲は、添付の特許請求の範囲およびその等価物の許容される最も広い解釈によって決定され、前述した詳細な説明によって制限も限定もされないものとする。

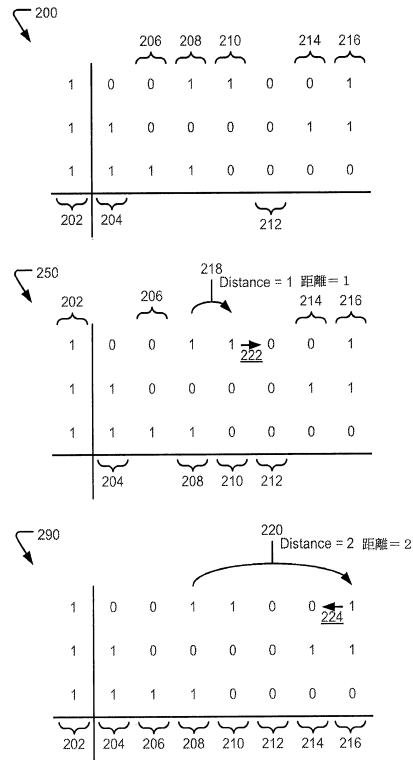
10

20

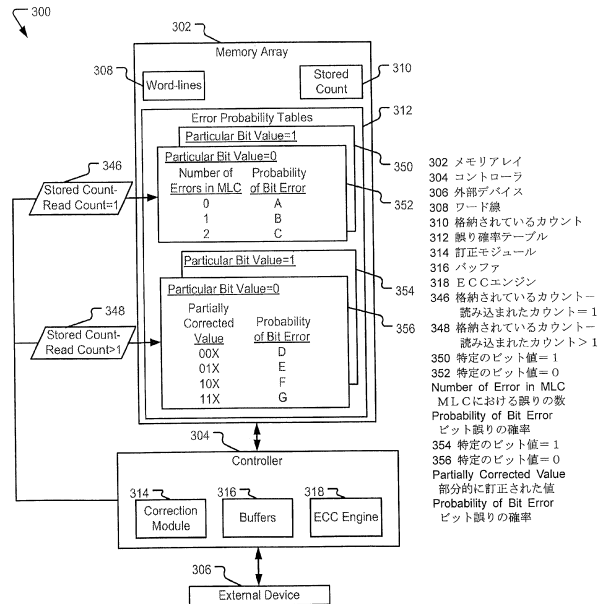
【 図 1 】



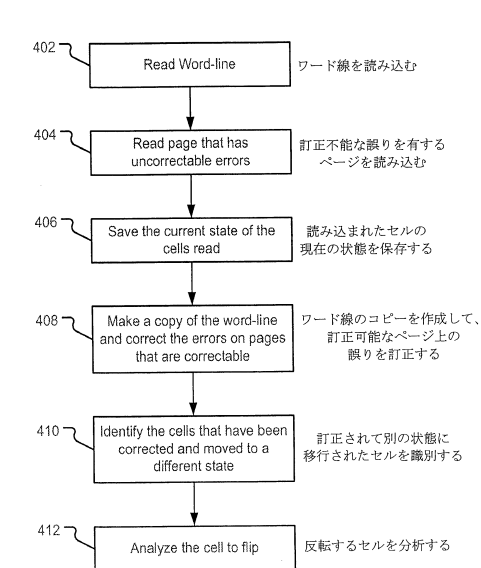
【 図 2 】



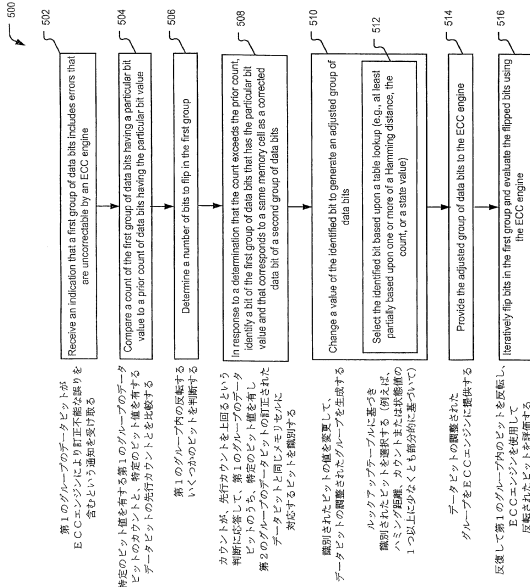
【 図 3 】



【 図 4 】



【 5 】



【 6 】

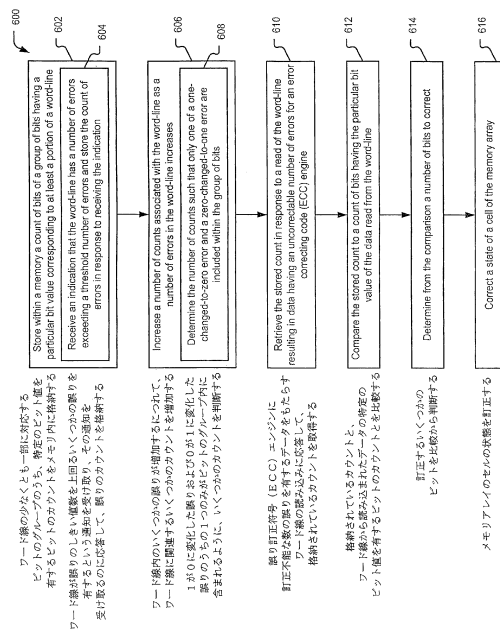


FIG. 5

FIG. 6

フロントページの続き

(72)発明者 スカラ, スティーブン

アメリカ合衆国、94539、カリフォルニア州、フリーモント、パゴサ ウェイ 293

(72)発明者 ゴンザレス, カルロス ジョセフ

アメリカ合衆国、95030、カリフォルニア州、モンテセリーノ、ブランシャール ドライブ
17675

審査官 酒井 恭信

(56)参考文献 米国特許第07634706 (US, B1)

米国特許出願公開第2010/0050046 (US, A1)

特開2005-004288 (JP, A)

特開2005-011386 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/16

G11C 29/42