

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5887589号

(P5887589)

(45) 発行日 平成28年3月16日(2016. 3. 16)

(24) 登録日 平成28年2月26日(2016. 2. 26)

(51) Int.Cl. F I
G06F 12/16 (2006.01)
 G06F 12/16 320F
 G06F 12/16 320G

請求項の数 13 (全 28 頁)

(21) 出願番号	特願2011-514152 (P2011-514152)	(73) 特許権者	511242535
(86) (22) 出願日	平成21年5月6日(2009. 5. 6)		サンディスク テクノロジーズ インコーポレイテッド
(65) 公表番号	特表2011-525655 (P2011-525655A)		SanDisk Technologies, Inc.
(43) 公表日	平成23年9月22日(2011. 9. 22)		アメリカ合衆国 75024、テキサス州、プラーノ、ノース ダラス パークウェイ 6900 ツー レガシー タウンセンター
(86) 国際出願番号	PCT/IB2009/051856		Two Legacy Town Center, 6900 North Dallas Parkway, Plano, Texas 75024 United States of America
(87) 国際公開番号	W02009/156877		
(87) 国際公開日	平成21年12月30日(2009. 12. 30)		
審査請求日	平成24年4月27日(2012. 4. 27)		
(31) 優先権主張番号	61/075, 065		
(32) 優先日	平成20年6月24日(2008. 6. 24)		
(33) 優先権主張国	米国 (US)		
前置審査			

最終頁に続く

(54) 【発明の名称】 ソリッドステートメモリの消去カウントにしたがってエラー修正をする方法および装置

(57) 【特許請求の範囲】

【請求項 1】

エラー修正を扱うための方法であって、

a) ソリッドステートメモリの少なくとも 1 つのブロックのための消去カウントを維持するステップと、

b) 関連する消去カウントを有するメモリブロックのうちの 1 つからデータを読み出すステップと、

c) 消去カウントに基づいて、遷移条件および読み出しデータ中のエラーを修正するために使用されるエラー修正パラメータを識別するポリシーを確立して実行するステップと、

d) 関連する消去カウントにしたがって、

i) 遷移条件と、

i i) エラー修正試行リソースバジェットのうちの少なくとも 1 つを決定するステップと、

e) エラー修正パラメータの第 1 のセットを使用して読み出しデータ中のエラーを修正する第 1 の試行を行うステップと、

f) エラーを修正する第 1 の試行が失敗した場合、エラー修正パラメータに基づいて、エラー修正パラメータの第 2 のセットを使用して識別された異なるデコーダまたはデコーダの異なるデコーダモードに遷移することによって読み出しデータ中のエラーを修正する第 2 の試行を行うステップと、を含み、

前記第2の試行が、

i) 第1の試行による、エラーを修正せずに遷移条件が満たされる場合にトリガされるモード遷移条件のトリガと、

ii) 第1の試行によるリソースバジェットの消耗のうちの少なくとも1つに依存し、

エラー修正パラメータの第2のセットは、エラー修正パラメータの第1のセットとは異なり、

前記第1および第2のセット用のエラー修正パラメータは、

(i) 複数のデコーダからデコーダを選択する選択パラメータと、

(ii) 特定のデコーダのために複数のデコーダモードからデコーダモードを選択するデコーダモードパラメータのうちの少なくとも1つによって定義される方法。 10

【請求項2】

請求項1記載の方法において、

前記方法が、モード遷移条件を決定するステップをさらに含み、

第2の試行が、少なくとも第1の試行による決定されたモード遷移条件のトリガに依存する方法。

【請求項3】

請求項1記載の方法において、

前記方法が、リソースバジェットを決定するステップをさらに含み、

第2の試行が、少なくとも第1の試行によるリソースバジェットの消耗に依存する方法 20

【請求項4】

請求項1記載の方法において、

少なくとも1つのエラー修正パラメータが、

i) 複数のデコーダからデコーダを選択する選択パラメータと、

ii) 特定のデコーダのために複数のデコーダモードからデコーダモードを選択するデコーダモードパラメータと、

iii) 読み出しデータのためのビット確率値と、

iv) ソフトビット数と、

v) デコーダバス幅のうちの少なくとも1つによって定義される方法。 30

【請求項5】

請求項1記載の方法において、

遷移条件が、

i) タイムアウト条件と、

ii) 繰り返し条件数と、

iii) CPUサイクル条件数のうちの少なくとも1つによって定義される方法。

【請求項6】

請求項1記載の方法において、

エラー修正試行リソースバジェットが、

i) 時間バジェットと、 40

ii) CPUサイクルバジェットと、

iii) 繰り返しバジェットのうちの少なくとも1つによって定義される方法。

【請求項7】

請求項1記載の方法において、

前記維持するステップ、読み出すステップ、決定するステップ、および1つ以上の試行のうちの少なくとも1つが、ソリッドステートメモリが存するストレージデバイスのデバイスコントローラによって実行される方法。

【請求項8】

エラー修正を扱うための方法であって、

a) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持 50

するステップと、

b) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出すステップと、

c) 消去カウントに基づいて、読み出しデータ中のエラーを修正するために使用されるエラー修正パラメータを識別するポリシーを確立して実行するステップと、

d) 関連する消去カウントにしたがって、

i) データのために読み出されるソフトビット数を決定するステップと、

i i) デコーディングバス幅のサイズを選択するステップのうちの少なくとも1つを実行するステップと、

e) ソフトビット数が決定された場合、データのためにソフトビット数を読み出すステップと、

f) データ中のエラー修正を試行するステップであって、

i) 選択されたデコーディングバス幅のサイズと、

i i) 読み出しソフトビットのうちの少なくとも1つを使用してエラー修正を試行するステップと、

g) 読み出しソフトビットを使用してエラーを修正する試行が失敗した場合、

i) 関連する消去カウントにしたがって、追加で読み出されるソフトビット数を決定するステップと、

i i) 読み出された追加のソフトビット数を使用してデータ中のエラー修正を再度試行するステップと、

を含む方法。

【請求項 9】

請求項 8 記載の方法において、

前記方法が、

データのために読み出されるソフトビット数を決定し、データのためにソフトビット数を読み出すステップと、

少なくとも読み出しソフトビットを使用してデータ中のエラー修正を試行するステップと、

をさらに含む方法。

【請求項 10】

請求項 8 記載の方法において、

前記方法が、

デコーディングバス幅のサイズを選択するステップと、

選択されたデコーディングバス幅のサイズを使用してデータ中のエラー修正を試行するステップと、

をさらに含む方法。

【請求項 11】

請求項 8 記載の方法において、

前記維持するステップ、1つ以上の読み出すステップ、決定するステップ、選択するステップ、および試行するステップのうちの少なくとも1つが、ソリッドステートメモリが
存するストレージデバイスのデバイスコントローラによって実行される方法。

【請求項 12】

データストレージデバイスであって、

a) ソリッドステートメモリと、

b) デバイスコントローラであって、

i) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持し、

i i) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出し、

i i i) 消去カウントに基づいて、遷移条件および読み出しデータ中のエラーを修正

10

20

30

40

50

するために使用されるエラー修正パラメータを識別するポリシーを確立して実行し、

i v) 関連する消去カウントにしたがって、

A) 遷移条件と、

B) エラー修正試行リソースバジェットの中の少なくとも1つを決定し、

v) エラー修正パラメータの第1のセットを使用して読み出しデータ中のエラーを修正する第1の試行を行い、かつ

v i) エラーを修正する第1の試行が失敗した場合、エラー修正パラメータに基づいて、エラー修正パラメータの第2のセットを使用して識別された異なるデコーダまたはデコーダの異なるデコーダモードに遷移することによって読み出しデータ中のエラーを修正する第2の試行を行うように動作可能なデバイスコントローラと、を備え、

10

前記第2の試行が、

A) 第1の試行による、エラーを修正せずに遷移条件が満たされる場合にトリガされるモード遷移条件のトリガと、

B) 第1の試行によるリソースバジェットの消耗のうちの少なくとも1つに依存し、

前記第1および第2のセット用のエラー修正パラメータは、

(i) 複数のデコーダからデコーダを選択する選択パラメータと、

(i i) 特定のデコーダのために複数のデコーダモードからデコーダモードを選択するデコーダモードパラメータのうちの少なくとも1つによって定義されるデータストレージデバイス。

【請求項13】

20

データストレージデバイスであって、

a) ソリッドステートメモリと、

b) デバイスコントローラであって、

i) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持し、

i i) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出し、

i i i) 消去カウントに基づいて、読み出しデータ中のエラーを修正するために使用されるエラー修正パラメータを識別するポリシーを確立して実行し、

i v) 関連する消去カウントにしたがって、

30

A) データのために読み出されるソフトビット数を決定することと、

B) デコーディングバス幅のサイズを選択することのうちの少なくとも1つを実行し、

v) ソフトビット数が決定された場合、データのためにソフトビット数を読み出し、

v i) データ中のエラー修正を試行することであって、

A) 選択されたデコーディングバス幅のサイズと、

B) 読み出しソフトビットのうちの少なくとも1つを使用してエラー修正を試行し、かつ

v i i) 読み出しソフトビットを使用してエラーを修正する試行が失敗した場合、

i) 関連する消去カウントにしたがって、追加で読み出されるソフトビット数を決定し、

40

i i) 読み出された追加のソフトビット数を使用してデータ中のエラー修正を再度試行するように動作可能なデバイスコントローラと、

を備えるデータストレージデバイス。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、フラッシュメモリなどのメモリから読み出されたデータにおけるエラーを修正する装置、方法、およびコンピュータ媒体に関する。

【0002】

50

関連出願の相互参照

本願は、本願発明者により2008年6月24日に出版された米国仮特許出願第61/075,065号(特許文献1)の利益を主張する。

【背景技術】

【0003】

不揮発性メモリは、電源なしで記憶データを保持できるタイプのメモリである。種々の読み出し機能と書き込み機能と消去機能、アクセス時間、データ保持、データ持続サイクルなどを有するいくつかのタイプの不揮発性メモリがある。電氣的に消去可能でプログラム可能なリードオンリーメモリ(EEPROM)は、1バイトあたりのレベルで読み出し/書き込み操作を実行でき、個別にそれぞれの記憶域を読み書きできることを意味する。

フラッシュ形のフローティングゲートトランジスタまたはセルで構成されたフラッシュメモリは、機能性と性能においてEEPROMメモリと同様の不揮発性メモリである。フラッシュメモリは、ある制限を伴って動作するものの比較的安価であることが利点である。フラッシュメモリは、最初に全体のメモリ部を消去せずに、前に書き込まれた位置に書き直すことができない。すなわち、フラッシュセルを再びプログラムすることができるようになる前に、フラッシュセルを消去しなければならない(例えば、「1」にプログラムする)。フラッシュメモリは通常消去ブロックと呼ばれるセルの比較的大きいグループを消去することができるだけである(例えば、多くの現在の市販のデバイスのサイズは16KBから2MBであるが、より高い容量を有するデバイスが商業的に利用可能になる場合には、消去ブロックのサイズが将来増加すると予想されている)。

【0004】

フラッシュメモリデバイスのエラー修正

フラッシュメモリおよび他のメモリ(例えば、磁気メモリ、光記憶、および揮発性メモリ)も同様に、これらのメモリに関する1つの顕著な特徴は、これらのメモリが「壊乱媒体」であるということであって、データがメモリから読み戻されるときに、1つ以上の情報ビットが「反転された」状態でメモリから読み出されてしまうかもしれないので、1つ以上のエラーがメモリに書き込まれているオリジナルのデータに時々取り込まれてしまうことである。

この現象を克服し、実際のアプリケーションでNANDタイプのメモリを使用可能にするために、これらのメモリに関連してエラー修正コード(ECC)を使用することは慣用技術である。

フラッシュメモリデバイスおよびフラッシュメモリ以外のソリッドステートメモリを含むストレージデバイスにおけるエラー修正を扱うための改善された技術と装置に対する継続した必要性が求められている。

【0005】

デバイスアーキテクチャの議論

図1A(従来技術)は、フラッシュメモリストレージデバイス260(従来技術)のブロック図である。フラッシュメモリストレージデバイスは、フラッシュメモリ270と、フラッシュメモリ270に対してデータを読み出し、かつデータを書き込むように動作可能なフラッシュコントローラ280とを含む。「プログラム」、「プログラミング」、「プログラムされた」、および「プログラム可能な」という用語は、フラッシュメモリでのデータの保存を指示するためにそれぞれ「書き込む」、「書き込み」、「書き込まれた」、および「書き込み可能な」という用語と共に本願明細書において互換性を持って使用される。

フラッシュメモリストレージデバイスの1つの実施形態が「周辺フラッシュストレージデバイス」である。周辺フラッシュストレージデバイスは、USBフラッシュドライブ(UFD)、PCカード、デジタルカメラ、ならびに音楽プレーヤ、ハンドヘルドおよびパームトップコンピュータ、および携帯電話と共に使用される小形状のストレージカードなどのフォームファクタにおけるコンピュータの技術分野において周知である。

【0006】

図1B（従来技術）は、デバイス側インターフェイス250を介してデータを交換するためにホストデバイス310（例えば、ラップトップもしくはデスクトップもしくはハンドヘルドコンピュータ、デジタルカメラ、携帯電話、音楽プレーヤ、およびテレビゲーム機）と「結合される」かまたは構成される周辺フラッシュメモリストレージデバイス260*（*は、フラッシュメモリストレージデバイスが周辺フラッシュストレージデバイスであることを示す）のブロック図である。周辺フラッシュメモリストレージデバイス260*とホストデバイス310が、ホスト側インターフェイス350とデバイス側インターフェイス250（例えば、それぞれのUSBまたはSDインターフェイス）を使用して通信リンク300を通して互いにやり取りする。

周辺フラッシュメモリストレージデバイス260*は、ホストデバイス310から分離されたデバイスとして図に示されているが、いくつかの実装形態では、フラッシュメモリストレージデバイス260*は、ホストデバイス310のハウジングの中に配置され得ることが理解できるはずである。

1つの実施形態では、フラッシュメモリストレージデバイス260*はホストデバイス310に対するデータの読み出しとデータの書き込みサービスを提供する。フラッシュメモリストレージデバイス260*によって受け取られたホストデバイス310からのデータは、フラッシュコントローラ280によってフラッシュメモリ270に書き込まれる。さらに、フラッシュメモリストレージで受け取られた「データ読み出し」要求に対応して、フラッシュコントローラ280はフラッシュメモリ270からデータを読み出す。

「読み出し時」または読み出し後のいつでも読み出しデータ中のエラーを修正できる。エラー修正は、少なくとも一部をフラッシュコントローラ280により、少なくとも一部をホストデバイス310（例えば、ホスト側プロセッサ320によるRAM330内の実行可能コード340の実行またはいかなる他の方法）により行うことができ、またその他の位置やその他の方法でも行うことができる。

【0007】

当業者であれば、「周辺フラッシュストレージデバイス」が唯一の類のフラッシュメモリストレージデバイスでないことが理解できるはずである。例えば、ある携帯電話、デスクトップ、ラップトップコンピュータ、PDAデバイスまたは他の電子装置もまた、フラッシュメモリとフラッシュコントローラを含むことができるが、必ずしもホストデバイスと結合するように構成され、かつ／またはホストデバイスのためのデータの読み出しサービスおよび／またはデータの書き込みサービスを提供することができるわけではない。

当業者であれば、図1A～1Bに示されているフラッシュメモリデバイスが、まさに1つの類の周辺不揮発性ストレージメモリデバイスであり、他のメモリデバイスが、光記憶や磁気抵抗のランダムアクセスメモリ（MRAM）のような他のタイプのソリッドステートメモリを含むことができることが理解できるはずである。

【先行技術文献】

【特許文献】

【0008】

【特許文献1】米国仮特許出願第61／075,065号

【特許文献2】米国特許第6,230,233号

【特許文献3】米国特許第5,341,339号

【特許文献4】米国特許第5,568,423号

【特許文献5】米国特許第5,388,083号

【特許文献6】米国特許第5,712,819号

【特許文献7】米国特許第6,570,790号

【特許文献8】米国特許第5,963,480号

【特許文献9】米国特許第6,831,865号

【特許文献10】米国特許第7,023,735号

【発明の概要】

【0009】

様々な実施形態は前述したような項目および関連する項目を扱い、方法とシステムを含む実施形態が本願明細書で提供される。

エラー修正を扱うための方法が今回初めて開示され、この方法は、a) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持することと、b) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出すことと、c) メモリブロックの関連する消去カウントにしたがって、i) 第1のデコーダと第2のデコーダのうちの1つを選択することと、i i) 第1のデコーダモードと第2のデコーダモードのうちの1つを選択することのうちの少なくとも1つを実行することと、d) 選択されたデコーダまたは選択されたモードだけを使用して読み出しデータ中のエラーを修正することと、を含む。

10

いくつかの実施形態では、ソリッドステートメモリは、フラッシュメモリである。

いくつかの実施形態では、維持すること、読み出すこと、実行すること、および修正することのうちの少なくとも1つが、ソリッドステートメモリが存するストレージデバイスのデバイスコントローラによって実行される。

いくつかの実施形態では、ストレージデバイスをデバイス間インターフェイスを介して動作可能にホストデバイスと結合し、前述した方法が、e) エラー修正済みデータをストレージデバイスからホストデバイスに送信することをさらに含む。

【0010】

エラー修正を扱うための方法が今回初めて開示され、この方法は、a) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持することと、b) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出すことと、c) 関連する消去カウントにしたがって、i) 決定することであって、A) より軽い重みデコーダを使用してエラー修正を試行するか、あるいはB) より軽い重みデコーダよりも重いより重い重みデコーダだけを使用してエラー修正を試行するかのいずれかを決定することと、i i) 決定することであって、A) より速いデコーダを使用してエラー修正を試行するか、あるいはB) より速いデコーダよりも遅いより遅いデコーダだけを使用してエラー修正を試行するかのいずれかを決定することと、i i i) 決定することであって、A) 特定のデコーダのより軽い重みモードを使用してエラー修正を試行するか、あるいはB) より軽い重みモードよりも重い、特定のデコーダのより重い重みモードだけを使用してエラー修正を試行するかのいずれかを決定することと、i v) 決定することであって、A) 特定のデコーダのより速いモードを使用してエラー修正を試行するか、あるいはB) より速いモードよりも遅い、特定のデコーダのより遅いモードだけを使用してエラー修正を試行するかのいずれかを決定することのうちの少なくとも1つを実行することと、d) 決定することのうちの少なくとも1つにしたがって、読み出しデータ中のエラーを修正することと、を含む。

20

30

いくつかの実施形態では、維持すること、読み出すこと、1つ以上の決定すること、および修正することのうちの少なくとも1つが、ソリッドステートメモリが存するストレージデバイスのデバイスコントローラによって実行される。

【0011】

エラー修正を扱うための方法が今回初めて開示され、この方法は、a) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持することと、b) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出すことと、c) 関連する消去カウントにしたがって、i) モード遷移条件と、i i) エラー修正試行リソースバジェットのうちの少なくとも1つを決定することと、d) エラー修正パラメータの第1のセットを使用して読み出しデータ中のエラーを修正する第1の試行を行うことと、e) エラーを修正する第1の試行が失敗した場合、エラー修正パラメータの第2のセットを使用して読み出しデータ中のエラーを修正する第2の試行を行うことであって、第2の試行が、i) 第1の試行によるモード遷移条件のトリガと、i i) 第1の試行によるリソースバジェットの消耗のうちの少なくとも1つに依存する、第2の試行を行うことと、を含む。

40

50

【0012】

いくつかの実施形態では、i) 前述した方法が、モード遷移条件を決定することを含み、ii) 第2の試行が、少なくとも第1の試行によるモード遷移条件のトリガに依存する。

いくつかの実施形態では、i) 前述した方法が、リソースバジェットを決定することを含み、ii) 第2の試行が、少なくとも第1の試行によるリソースバジェットの消耗に依存する。

いくつかの実施形態では、少なくとも1つのエラー修正パラメータが、i) 複数のデコーダからデコーダを選択する選択パラメータ（例えば、デコーダの「重み」または「デコーダの速度」を説明するパラメータ）と、ii) 特定のデコーダのために複数のデコーダモードからデコーダモードを選択するデコーダモードパラメータ（例えば、デコーダの「重み」またはデコーダモードの「速度」を説明するパラメータ）と、iii) 読み出しデータのためのビット確率値と、iv) ソフトビット数と、v) デコーダバス幅と、からなるグループから選択される。

いくつかの実施形態では、モード遷移条件が、i) タイムアウト条件と、ii) 繰り返し条件数と、iii) CPUサイクル条件数と、からなるグループから選択される。

いくつかの実施形態では、エラー修正試行リソースバジェットが、i) 時間バジェットと、ii) CPUサイクルバジェットと、iii) 繰り返しバジェットと、からなるグループから選択される。

いくつかの実施形態では、維持すること、読み出すこと、決定すること、および1つ以上の試行のうちの少なくとも1つが、ソリッドステートメモリが存するストレージデバイスのデバイスコントローラによって実行される。

【0013】

エラー修正を扱うための方法が今回初めて開示され、この方法は、a) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持することと、b) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出すことと、c) 関連する消去カウントにしたがって、i) データのために読み出されるソフトビット数を決定することと、ii) デコーダバス幅のサイズを選択することのうちの少なくとも1つを実行することと、d) ソフトビット数が決定された場合、データのためにソフトビット数を読み出すことと、e) データ中のエラー修正を試行することであって、i) 選択されたデコーダバス幅のサイズと、ii) 読み出しソフトビットのうちの少なくとも1つを使用してエラー修正を試行することと、を含む。

いくつかの実施形態では、i) 前述した方法が、データのために読み出されるソフトビット数を決定し、データのためにソフトビット数を読み出すことを含み、ii) 試行することは、少なくとも読み出しソフトビットを使用してデータ中のエラー修正を試行することを含む。

いくつかの実施形態では、i) 前述した方法は、デコーダバス幅のサイズを選択することを含み、ii) 試行することは、選択されたデコーダバス幅のサイズを使用してデータ中のエラー修正を試行することを含む。

いくつかの実施形態では、維持すること、1つ以上の読み出すこと、決定すること、選択すること、および試行することのうちの少なくとも1つが、ソリッドステートメモリが存するストレージデバイスのデバイスコントローラによって実行される。

【0014】

データストレージデバイスが今回初めて開示され、このデータストレージデバイスは、a) ソリッドステートメモリと、b) デバイスコントローラであって、i) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持し、ii) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出し、iii) メモリブロックの関連する消去カウントにしたがって、A) 第1のデコーダと第2のデコーダのうちの1つを選択することと、B) 第1のデコーダモードと第2のデコーダモードのうちの1つを選択することのうちの少なくとも1つを実行し、かつiv) 選択されたデコー

ダまたは選択されたモードだけを使用して読み出しデータ中のエラーを修正するように動作可能なデバイスコントローラと、を備える。

【0015】

データストレージデバイスが今回初めて開示され、このデータストレージデバイスは、
a) ソリッドステートメモリと、b) コントローラであって、i) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持し、i i) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出し、i i i) 関連する消去カウントにしたがって、A) 決定することであって、I) より軽い重みデコーダを使用してエラー修正を試行するか、あるいはI I) より軽い重みデコーダよりも重いより重い重みデコーダだけを使用してエラー修正を試行するかのいずれかを決定することと、B) 決定することであって、I) より速いデコーダを使用してエラー修正を試行するか、またはI I) より速いデコーダよりも遅いより遅いデコーダだけを使用してエラー修正を試行するかのいずれかを決定することと、C) 決定することであって、I) 特定のデコーダのより軽い重みモードを使用してエラー修正を試行するか、あるいはI I) より軽い重みモードよりも重い、特定のデコーダのより重い重みモードだけを使用してエラー修正を試行するかのいずれかを決定することと、D) 決定することであって、I) 特定のデコーダのより速いモードを使用してエラー修正を試行するか、あるいはI I) より速いモードよりも遅い、特定のデコーダのより遅いモードだけを使用してエラー修正を試行するかのいずれかを決定することのうちの少なくとも1つを実行し、かつi v) 決定することのうちの少なくとも1つにしたがって、読み出しデータ中のエラーを修正するように動作可能なコントローラと、を備える。

10

20

【0016】

データストレージデバイスが今回初めて開示され、このデータストレージデバイスは、
a) ソリッドステートメモリと、b) デバイスコントローラであって、i) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持し、i i) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出し、i i i) 関連する消去カウントにしたがって、A) モード遷移条件と、B) エラー修正試行リソースバジェットの中の少なくとも1つを決定し、i v) エラー修正パラメータの第1のセットを使用して読み出しデータ中のエラーを修正する第1の試行を行い、かつv) エラーを修正する第1の試行が失敗した場合、エラー修正パラメータの第2のセットを使用して読み出しデータ中のエラーを修正する第2の試行を行い、第2の試行が、A) 第1の試行によるモード遷移条件のトリガと、B) 第1の試行によるリソースバジェットの消耗のうちの少なくとも1つに依存するように動作可能なデバイスコントローラと、を備える。

30

【0017】

データストレージデバイスが今回初めて開示され、このデータストレージデバイスは、
a) ソリッドステートメモリと、b) デバイスコントローラであって、i) ソリッドステートメモリの少なくとも1つのブロックのための消去カウントを維持し、i i) 関連する消去カウントを有するメモリブロックのうちの1つからデータを読み出し、i i i) 関連する消去カウントにしたがって、A) データのために読み出されるソフトビット数を決定することと、B) デコーダバス幅のサイズを選択することのうちの少なくとも1つを実行し、i v) ソフトビット数が決定された場合、データのためにソフトビット数を読み出し、かつv) データ中のエラー修正を試行することであって、A) 選択されたデコーダバス幅のサイズと、B) 読み出しソフトビットのうちの少なくとも1つを使用してエラー修正を試行するように動作可能なデバイスコントローラと、を備える。

40

【0018】

ある実施形態に関して前に説明した特徴は、以下の本願明細書においてそれら特徴と反対のことが示されないかぎり他の実施形態にも含まれ得ることに留意するべきである。

【図面の簡単な説明】

【0019】

【図1A】フラッシュメモリストレージデバイスのブロック図である。

50

【図 1 B】ホストデバイスに結合される周辺フラッシュメモリストレージデバイスのブロック図である。

【図 2 A】1つ以上のメモリブロックの消去カウントにしたがって選択されたデコーダを使用してエラーを修正するルーチンのフローチャートである。

【図 2 B】1つ以上のメモリブロックの消去カウントにしたがって選択されたデコーダモードを使用してエラーを修正するルーチンのフローチャートである。

【図 3 A】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントに依存して、より軽い重みデコーダを使用してエラー修正の試行が行われる。

【図 3 B】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントに依存して、より速いデコーダを使用してエラー修正の試行が行われる。

10

【図 3 C】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントに依存して、より軽い重みデコーダモードを使用してエラー修正の試行が行われる。

【図 3 D】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントに依存して、より速いデコーダモードを使用してエラー修正の試行が行われる。

【図 4 A】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントにしたがってモード遷移トリガ条件が決定される。

20

【図 4 B】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントにしたがってエラー修正試行のためのリソースバジェットが決定される。

【図 5 A】ルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントにしたがってエラー修正試行のために読み出されるソフトビット数が決定される。

【図 5 B】ルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントにしたがってエラー修正試行のために読み出されるソフトビット数が決定される。

【図 6 A】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントにしたがってデコーダバス幅が決定される。

30

【図 6 B】エラーを修正するルーチンのフローチャートであって、ここで1つ以上のメモリブロックのための消去カウントにしたがってデコーダバス幅が決定される。

【発明を実施するための形態】

【0020】

図面を参照する例示の実施形態の本願における詳細な説明について言及することによって、添付の特許請求の範囲をより良く理解することができる。説明、実施形態、および図面が特許請求の範囲を限定するものとみなされてはならない。エラー修正を扱うための本願で開示された方法と装置のあらゆる特徴が、すべてのインプリメンテーションでも必要であるというわけではないことが理解されるべきである。また、本願での開示の全体にわたり、1つのステップが先に実行された別のものに依存することが文脈から明確でない場合であっても、明示もしくは説明したところのプロセスまたは方法にて、方法のステップが順不同または同時に実行され得ることが理解されるべきである。本願全体にわたり使用されているように、「・ ・ かもしれない」という用語は、義務的な感覚（すなわち、「・ ・ すべきである」という意味）というよりはむしろ許容的な意味（すなわち、「・ ・ 可能性を持っている」という意味）で使用されている。

40

【0021】

本願発明者は、データが読み出されるメモリブロックの消去カウントにしたがって、ソリッドステートメモリから読み出されたデータのための ECC デコーディングを行ういくつかの技法をここで開示する。

50

どんな特定の理論によっても拘束されることを願わず、ソリッドステートメモリ（例えば、フラッシュメモリ、MRAMまたはいかなる他のタイプのソリッドステートメモリ）が複数回消去されるときには、このことが記憶媒体としてメモリの品質を下げ得ることに留意すべきである。そういうものとして、比較的「大きい」回数データが消去されたソリッドステートメモリ（またはその区分）は、比較的「小さい」回数データが消去されたソリッドステートメモリ（またはその区分）よりも、「より質が悪い」または「より信頼性が低い」かまたは「より壊乱的」な記憶媒体であるかもしれない。

データが読み出される1つのブロックの消去カウントにしたがって、メモリから読み出されたデータのためのECCデコーディングを行ういくつかの技法をここで説明する。データが「高い消去カウント」を有するブロックから読み出される場合、メモリブロックが記憶媒体として比較的「より信頼性が低い」ことを示すことができる。そういうものとして、「より信頼性が低い」メモリブロックから読み戻されたデータは、比較的「大きい」エラー回数を含むかもしれない。この場合、比較的「大きな数」のエラーを有するデータ中のエラー修正に適切な「悲観的」な技法が使用できる。逆に、データが「低い消去カウント」を有するブロックから読み出される場合、ブロックが記憶媒体として「より信頼性が高い」ことを示すことができ、比較的「小さい数」のエラーを有するデータ中のエラー修正に適切な「楽観的」な技法が使用できる。

【0022】

第1の技法（図2Aに関する議論を参照する）は、1つ以上のメモリブロックから読み出されたデータにおけるエラーを修正する場合に使用できる「より軽い重み」デコーダまたは「より重い重み」デコーダのいずれかの状況に関する。データが読み出されるメモリブロックの消去カウントが比較的「高い」（メモリブロックが比較的多くの「消耗」にさらされてきていることを示す）場合、データが読み出されたメモリブロックの「品質」が比較的「低い」傾向があることを示すかもしれない。この場合、このように「高い」消去カウントは、たとえより重い重みデコーダがより遅く、かつ／またはより多くのコンピュータリソース（例えば、CPUサイクル、電力、および／またはメモリリソース）を必要としても、エラーを修正する場合により「悲観的」な技法を使うことや、「より重い重み」デコーダを使うことが有利であり得ることを示すことができる。逆に、消去カウントが比較的「低い」場合、データが読み出されたメモリブロックの「品質」が比較的「高い」傾向にあり、「より軽い重み」デコーダを使うことが有利であり得ることを示すことができる。

【0023】

第2の技法（図3A～3Dに関する議論を参照する）は、「試行錯誤」デコーディングルーチンに関する。「試行錯誤」デコーディングルーチンは、以下のように動作する。まず最初に、比較的「軽い」または「速い」デコーダ（または、デコーダモード）を使用してエラーを修正する試行が行われる。この試行が成功した場合には、比較的高速なデコーディングおよび／または比較的わずかなコンピュータリソースを消費するデコーディングを「享受」することが可能である。そうではなくて、この試行が失敗した場合には、より遅く、かつ／またはより多くのコンピュータリソースを消費するが、エラー修正をより失敗しない傾向にある、「より重い重み」（またはより遅い）デコーダ（またはデコーダモード）が、データ中のエラーを修正するのに使用される。

この第2の技法にしたがって、比較的「高い」消去カウントを有するブロックからデータが読み出される場合には、悲観的なアプローチが取られるべきであって、より軽い重みまたはより速い（が、信頼性が低い）デコーダ（またはモード）を使用してエラーをデコードするのを試行するステップを「悲観的に」スキップすることは有利であることがここで開示されている。この場合、この「悲観的な」アプローチは、どんな試行でもとにかく失敗しそうな軽い重みおよび／または速いデコーダを使用させられないところで、時間の節約および／またはコンピュータリソースを節約するのに役立つかもしれない。逆に、比較的「低い」消去カウントを有するブロックからデータが読み出されるならば、軽い重みおよび／または速いデコーダがより失敗しそうにないと想定されるかもしれない。このよ

10

20

30

40

50

うに、データがこの「より低い」消去カウントに対応しているブロックから読み出される場合、より軽い重みおよび／またはより速いデコーダが失敗するかもしれないという「リスクを取ることに」に対して有利であるかもしれない。したがって、データがこの「より低い」消去カウントに対応しているブロックから読み出される場合、「より信頼性の高い」より重い重みおよび／またはより遅いデコーダ（またはモード）を試行する前に、より軽い重みおよび／またはより速いデコーダ（またはモード）を使用してデータをデコードするのを試行することは有利であるかもしれない。

【0024】

また、第3の技法（例えば、図4A～4Bに関する議論を参照する）は、「試行錯誤」デコーディングルーチンに関する。この第3の技法に関連するいくつかの実施形態によると、最初に、「より小さい」エラー回数を扱うのに適当なより軽い重みデコーダおよび／またはより速いデコーダおよび／またはデコーダモード（またはデコーダ）を使用してエラーを修正するように試行がなされる。この第1の試行が失敗した場合（例えば、ある種の「タイムアウト」条件によりトリガされ、その結果として、この第1の試行が特定の時間内に成功しなかった場合）には、「より大きい」エラー回数を扱うのに適当なより重い重みおよび／またはより遅いデコーダおよび／またはモード（またはデコーダ）を使用してエラーを修正するように第2の試行がなされる。

【0025】

「高い」消去カウントを有する1つのブロックのソリッドステートメモリからデータ（すなわち、そこでのエラーを修正することが望ましい）が読み出される場合、次により軽い重みまたはより速い（が、それほど信頼性が高くない）デコーダ（またはモード）を使用して単に「悲観的」な方法で、すなわち比較的「わずかな」リソースを投入することによって、エラーをデコードするように試行するステップを行うことが有利であり得ることがここで開示されている。このように1つの実施形態において、「より軽い重み」または「より速いデコーダ」を使用してエラーを修正する試行が比較的「高速に」（「小さいタイムアウト値」の中で）成功しなかった場合には、次に、（i）この試行は比較的「高速に」断念され（すなわち、「失敗」であると見なされ）、（ii）より遅くかつ／またはより重い重み、および／または「より大きい」エラー回数の取り扱いにより適している別のデコーダおよび／またはデコーダモードを使用してエラーを修正する「新しい」試行がなされる。

【0026】

逆に、修正すべきデータが「高い」消去カウントを有する1つのブロックのソリッドステートメモリから読み出される場合、より「楽観的」な方法で、より軽い重みまたはより速い（が、信頼性の低い）デコーダ（またはモード）を使用してエラーをデコードし、すなわち「より多く」のリソースを軽い重みおよび／またはより速いデコーダ（またはデコーダモード）に投入することによって、試行するステップを行うことが有利であるかもしれない。このように1つの実施形態では、「より軽い重み」または「より速いデコーダ」を使用することでエラーを修正する試行が比較的「高速に」成功しなかった場合でも、それにもかかわらず、まだこの試行が結局成功するという何らかの見込みがあると考えることができる。このように、この実施形態では、「より重い重み」および／または「より遅い」デコーダまたはモードでエラーを修正する試行を「進める」前に、「より重い重み」および／または「より遅い」デコーダまたはモードでエラーを修正する初期の試行に「機会が与えられ」、例えば、失敗として明示される前に、より長い期間（またはより多くのCPUサイクルで）動作するのが容認される。

【0027】

第4の技法（例えば、図5A～5Bに関する議論を参照する）は、エラー修正のために読み出されるソフトビット数を決定することに関する。その全体が本願明細書において参照により援用されている、米国特許出願第11／642,708号および第11／651,483号では、メモリにおけるエラー修正能力を改善するためのソフトビットの使用について議論されている。ソフトビットの読み出しには時間と電力というコストを招くので

10

20

30

40

50

、ECCデコーダが全くソフトビットを読み出さずに（または「低い」数のソフトビットだけを読み出して）最初にデータをデコードするのを試行する多くの状況がある。それが失敗したのであれば、1つ以上のソフトビットを読み出すことができ、別のデコーディング試行を行うことができる。また、この第2の試行も失敗したのであれば、より多くのソフトビットが読み出され、別のデコーディング試行が行われる。デコーディングが成功するかあるいは有効なソフトビットの最大限度の数に達するまで、このプロセスを繰り返し続けることができる。

【0028】

初期の試行用いくつかのソフトビットを読み出したたり、使用したりするべきかの決定（例えば、図5AのステップS419を参照する）、および／または初期の試行が失敗した後に、いくつかの追加のソフトビットを読み出すべきかの決定（例えば、図5BのステップS473を参照する）が、データが読み出されるメモリブロックの消去カウントにしたがって行われ得ることがここで開示されている。

10

このように、この消去カウントが比較的「高い」場合、「悲観的」なアプローチが取られるかもしれない、ゼロまたは「小さい」数のソフトビットを使用した繰り返しが、これらの繰り返しが成功しそうにない（すなわち、データが多くのエラーを含んでいそうであり、このようにより多くのソフトビットが「ノイジー」データ中でエラーを修正する必要があるという前提でスキップされるかもしれない（し、もしくは失敗した後に、相対的に「大きい」数のソフトビットが読み出されるかもしれない）。

【0029】

20

逆に、この消去カウント（すなわち、データが読み出されるメモリのブロックの消去カウント）が比較的「低い」場合、「楽観的」なアプローチが取られるかもしれない、データが比較的「高い」品質であるか、ソフトビットなし、あるいは比較的「小さい」数のソフトビットを使用することにおいて妥当な成功の機会が得られそうであるという前提の、ゼロまたは「小さい」数のソフトビットを使用することでエラー修正の試行を行うことは、この状況では有利であるかもしれない。

【0030】

第5の技法（例えば、図6Aを参照する）は、デコーダバスの幅に関する。データが比較的「高い」消去カウントを有するメモリブロックから読み出される場合、データが低品質であるかもしれない、エラーを修正することへの比較的「悲観的」なアプローチが好まれ得ることが開示されている。このように、エラーを修正するには、より大きいデコーダバスが、より遅くかつ／またはより多くのコンピュータリソースを必要とするかもしれないが、この状況ではより大きいデコーダバスを使用することが有利であるかもしれない。逆に、データが比較的「低い」消去カウントを有するメモリブロックから読み出される場合、データが高品質であって、より小さいデコーダバスでエラー修正を試行する、エラーを修正することへの比較的「楽観的」なアプローチが好ましいことを示している。

30

【0031】

本願での開示全体を通じて、ある教示について「フラッシュメモリ」の表現で説明した場合には、この教示はフラッシュメモリ以外の他のソリッドステートメモリ（例えば、光記憶またはMRAM）にも適切なものであり得ることを理解するべきである。

40

当業者であれば、ここで開示されたエラー修正の技法がメモリから読み出されたデータにおけるエラーを修正するためのいかなる他の技法とも組み合わせることができることが理解できるはずである。

図面についての説明の前に、フラッシュメモリにおけるECC使用の概要を提示し、（次の2つのセクションで）様々な用語の定義を提供する。

【0032】

フラッシュメモリデバイスのエラー修正

フラッシュメモリにおけるECC使用の概要は、以下に提示され、以下のステップを含む。

（1）メモリにデータを書き込む前に、後で誤り検出・修正に使用される、追加の（す

50

なわち、冗長な) ビットを計算するために E C C アルゴリズムがデータに適用される。これらの冗長ビットはしばしば「パリティビット」または「パリティ」と呼ばれる。E C C モジュールとパリティ出力へのそのモジュールによるデータ入力の組み合わせは符号語と呼ばれる。E C C モジュールへの入力データの各々異なる値は符号語の異なる処理結果をもたらす。

(2) 符号語全体(すなわち、オリジナルのデータとパリティ)はフラッシュメモリに記録される。N A N D タイプのフラッシュメモリの実際のサイズはオリジナルデータのサイズよりも大きく、またメモリはパリティにも対応するように設計されていることに留意するべきである。

(3) データがメモリから取り出されるときに、符号語全体が再度読み出され、また検出、修正することが可能な「ビットフリップ」(すなわち、エラー)データおよびパリティに E C C アルゴリズムが適用される。

10

【0033】

E C C インプリメンテーションがハードウェア、ソフトウェア、またはハードウェアとソフトウェアとの組み合わせで行われ得ることに留意するべきである。さらに、E C C をメモリデバイス中、メモリデバイスコントローラ中、またはホストコンピュータ中にインプリメントすることもできるし、あるいはシステムのこれらのコンポーネント中に「分散」することもできる。

共用のアルゴリズムは、リード-ソロモン、B C H、ハミング、および多くの他のものを含む。それぞれの E C C アルゴリズムは、2つの部分、すなわちデータビットを受信しパリティビットを生成する部分(もしくは符号語を生成するのと同様)と、符号語を受信し修正済みデータビットを生成する部分とで構成される。第1の部分「エンコーダ」と呼ばれて、書き込み中に使用され、第2の部分「デコーダ」と呼ばれて、読み出し中に使用される。ハードウェアまたはソフトウェアのいずれかでそれぞれの2つの部分をインプリメントすることもできるし、また一方の部分をハードウェアでインプリメントし、もう一方の部分をソフトウェアでインプリメントすることも可能である。また、それぞれの部分がハードウェアとソフトウェアとの組み合わせでインプリメントすることも可能である。

20

【0034】

データビットを受信し対応する符号語を生成することを「エンコーディング」と呼ぶ。符号語を受信し修正済みデータビットを生成することを「デコーディング」と呼ぶ。

30

2種類の E C C が実際にあることに留意するべきである。前に説明した E C C の種類は「システムの」な E C C と呼ばれ、データビットのアイデンティティは符号語で保持される。「非システムの」な E C C では、データビットは元のデータビットのアイデンティティが保持されない符号語に変換される。

フラッシュメモリシステムで使用されるべき E C C アルゴリズムとして、B C H のようなアルゴリズムを選択することが唯一選択されたソリューションを定義するものではない。どのような E C C アルゴリズムであっても実際にはただ一つのアルゴリズムではなく、アルゴリズムのファミリー(一群)である。同じファミリーの中のアルゴリズムがその中でそれらが保護できるデータビットの量は異なる。2つのアルゴリズムが通常非常に類似し、かつ同じ原理で動作するものであったとしても、100のデータビットを保護する必要があるアルゴリズムは、1万のデータビットを保護する必要があるアルゴリズムと同じではない。

40

【0035】

ともに同じ数のデータビットを保護する同じファミリーの2つのアルゴリズムであっても必ず同じであるというわけではない。アルゴリズムはそのアルゴリズムが修正することができるデータのビットエラーの数において提供される信頼度のレベルまたは等価なもので異なるかもしれない。例えば、あるシステムでは、3ビットエラーまでのいかなる組み合わせに対しても1,000のデータビットのチャンクの保護を必要とするかもしれない(が、4ビットエラー以上の出現に対しては不可能である)一方で、別のシステムでは、

50

はるかに高い信頼性が望まれ、したがって、10ビットエラーまでのいかなる組み合わせに対しても1,000のデータビットのチャンクの保護を必要とする。通常、より多くのエラーに対する保護は、より多くのパリティビット（またはより長い符号語）の使用を必要とし、ECCスキームをより非「効率的」にし、ここで効率とは符号語のデータビットの数対符号語のビットの総数の比率によって測られる（システム的なECCにデータビットとパリティビットの両方を含む）。この計測は通常ECCコード化の「レート」と呼ばれる。

【0036】

また、種々のECCアルゴリズムとインプリメンテーションとは他のアスペクトでも異なる。例えば、エンコーディングプロセスの速度、デコーディングプロセスの速度、エンコーディングプロセスの複雑さ、デコーディングプロセスの複雑さ、デコーダへの入力における許容エラーレート（記憶セルの品質にしたがって定義される）等々である。単に動作の速度に影響するだけではなく、ECCスキームのハードウェアインプリメンテーションの消費電力とシリコン面積にも影響するので、エンコーディングとデコーディングの複雑さは重要である。

10

【0037】

このように、メモリシステムに対するECCソリューションの選択が複数の考慮事項間の複雑なトレードオフに関わることは明らかである。ECCの設計における当該技術分野での典型的いくつかの非限定的な経験則は以下のとおりである。

a. 特定のメモリ信頼性において、出力信頼性が良ければ良いほど（修正可能なエラーの数がより大きいと同等）コードのレートがより低くなる（システム的なECCにおいて、より多くのパリティビットが必要であることと同等）。

20

b. 特定のメモリ信頼性において、出力信頼性が良ければ良いほどデコーダはより複雑になる。

c. 特定のレベルの出力信頼性において、コードのレートが高ければ高いほどデコーダはより複雑になる。

d. 特定のレベルの出力信頼度において、コードのレートが高ければ高いほどデコーディングはより遅くなる。

【0038】

ECCソリューションを設計する場合には、デコーダの入力におけるエラーレート（記憶セルの品質によって決定される）と所望の出力信頼性（アプリケーションの必要条件によって決定される）から通常始める。これらの数に基づいて、特定のECCファミリーを通常選択し、パリティビットの必要数を算出して、エンコーダとデコーダの速度と複雑さを見積もっている。

30

【0039】

いくつかの場合、システムの設計者にとって最も重要な考慮すべき事柄はデコーディングの速度であり、このことがメモリからデータを読み出す速度に制限を設けるかもしれない。そのような場合には、設計者はジレンマに遭遇するかもしれない。出力信頼性の要求を満たすのに必要なECC方式が、システムのスピード目標を満たさない、低速動作のかなり複雑なデコーダとなってしまうかもしれないからである。しかし、その一方で、高速なデコーディングをもたらす比較的簡単なECC方式を選択することは、必要な出力信頼性レベルを提供しない。

40

【0040】

定義

利便上、ここで記述される文脈の中で様々な用語が提示されている。本願における明細書の色々なところで、定義が明確または暗黙に提供されている範囲にわたり、そのような定義が関連する技術のその技術により定義された用語の語法とで一貫性を持つべきであることが理解されるはずである。さらに、そのような定義はそのような用語の語法と一致した可能な限り広い意味で解釈されるべきである。

ソリッドステートメモリのあるタイプは「壊乱媒体」である（例えば、フラッシュメモ

50

リ、光記憶、またはMRAM) ので、時々1つ以上のエラーがメモリに書き込まれる「オリジナル」のデータに取り込まれてしまい、データ（すなわち、オリジナルデータの表示）がメモリから読み戻されるときに1情報ビット以上が「フリップ」されてしまう。

【0041】

本発明の実施形態は、「メモリブロック」のための「消去カウント」に関連する。メモリ「ブロック」とは、消去のための最も小さい「チャンク」またはメモリ容量である。フラッシュメモリに関する1つの例では、ブロックは、いくつかのフラッシュセル、例えば、少なくとも数十、数百または数千、あるいはそれ以上のフラッシュセルを含む。他のタイプのメモリ、例えばRAMでは、「ブロック」は単一のメモリセルまたはいくつかの複数のメモリセルであるかもしれない。

10

メモリの特定のブロックのための「消去カウント」は、ソリッドステートメモリのブロックが今までに何回消去されたかを示すカウンタである。メモリのブロックが消去されるときはいつでもこの消去カウントが増加されることになるかもしれない。前に言及したように、ブロックが比較的「大きい」消去カウントを有する場合には、ブロックがエラーを起こしがちな比較的「プア」な記憶媒体であることを示すかもしれない。

【0042】

本発明の実施形態は、「フラッシュメモリの少なくとも1つのブロックのための消去カウントを維持する」ことに関する。消去カウントを、ホスト側および／またはデバイス側のいずれかの上で維持でき、揮発性（例えば、RAMまたはレジスタ）および／または不揮発性メモリ（例えば、フラッシュメモリ）のどのような組み合わせにも保存できる。1つの例では、消去カウントはコントローラのダイ上に保存される。別の例では、消去カウントはフラッシュダイの1つの中に保存される。消去カウントを、単一な場所および／または複数の場所に分散して保存することができる。1つの非限定的な例では、消去カウントはフラッシュブロックの中でウェアレベリングをインプリメントするのに使用されるデータ構造に保存することができる。

20

【0043】

いくつかの非限定的な例（しかし、すべての実施形態ではない）では、ソリッドステートメモリの書き込み時はいつでも、最初に書き込みプロセスの一部としてソリッドステートメモリを消去しなければならない。そのようなシステムでは、「消去カウント」は特定のブロックに書き込まれた回数を少なくとも部分的に表現できる。

30

その全体が本願明細書において参照により援用されている、Lofgren らによる米国特許第6, 230, 233号（特許文献2）、Wells による米国特許第5, 341, 339号（特許文献3）、Jou らによる米国特許第5, 568, 423号（特許文献4）、Assar らによる米国特許第5, 388, 083号（特許文献5）、Harariによる米国特許第5, 712, 819号（特許文献6）、Harariによる米国特許第6, 570, 790号（特許文献7）、Harariによる米国特許第5, 963, 480号（特許文献8）、Chang らによる米国特許第6, 831, 865号（特許文献9）に開示された技法を含むが、これらに限定されないどのような技法も消去カウントを維持するために使用できる。

【0044】

メモリから読み戻されたデータからオリジナルのエラーのないデータを復元するプロセスが「エラー修正」である。「エラー修正」という用語を複数形で使用しているが、「エラー修正」がただ一つのエラーをも修正することを示すことを理解するべきである。また、いくつかの実施形態では、エラー「修正」がエラー修正の1つ以上の試行の失敗を含み得ることも理解されるべきである。

40

前に言及したように、エラーを修正するために「デコーダ」を使用でき、オリジナルのデータを符号語の表示から復元するためにも使用できる。いくつかの実施形態では、「より軽い重み」デコーダと「より重い重み」デコーダを含む、複数のデコーダを伴っている。

用語「より重い重み」と「より軽い重み」は、2つのデコーダおよび／または単一のデコーダの2つのモードを比較するのに使用される。「より重い重み」デコーダ（単一のデ

50

コーダのより重い重みモード)は、(i)「より軽い重み」対照物(デコーダ)より多くの電流を消費し、かつ／または(ii)「より軽い重み」対照物より多くのメモリ(デコーダ自体のコンピュータ実行可能コードのためのメモリおよび／またはエラー修正時の計算の「中間結果」を保存するためのメモリのいずれか)を必要とし、かつ／または(iii)「より軽い重み」対照物より多くのコンピュータによる動作を必要とする。

【0045】

いくつかの実施形態が、データが読み出されるメモリブロックの消去カウントにしたがって、エラー修正時に読み出されかつ／または使用される「ソフトビット」数を決定することに関する。「ソフトビット」は見積りの信頼性尺度に結合されたビット値の見積りである。多くの場合、同じ数がビット値の見積りと信頼性尺度の両方に役立つ。例えば、対数尤度比(LLR: log-likelihood-ratio)のサインはLLRによって表されたビット値の見積りであり、またLLRの大きさはその見積りの信頼性の基準である。

10

【0046】

いくつかの実施形態では、「ソフトビット」はデータがセルの中に書き込まれた分解能よりも「より高い分解能」を使用することでメモリから読み出すことによって、計算したりかつ／または決定したりすることができる。「より高い分解能」を使用して読み出された追加のビットはデータのビット値のあらゆる見積りの「信頼性」のある種の指標を与えることができる。1つの例では、電圧バンドの境界を画定する異なった「基準電力値」を使用してメモリにデータを書き込み、また電圧バンドまたは「断片的な基準電力値」に属する基準電力値を使用してデータを読み戻すことによりソフトビットを生成し得る。当業者は、例えばその全体が本願明細書において参照により援用されている、Banの米国特許第7,023,735号(特許文献10)に言及している。1つ以上のソフトビットにしたがってデータが修正される場合には、エラー修正ルーチンを改善するのに役立つかもしれない。しかし、ソフトビットを決定することに関連した時間および／またはコンピュータリソースというコストが存在する。

20

【0047】

事前の議論：消去カウントの維持とソリッドステートメモリからのデータ読み出し(図2A~2B、3A~3D、4A~4B、5A~5B、6A~6DのステップS411とS423の議論)

ステップS411では、消去カウントは少なくとも1つのメモリブロックのために維持される。前に言及したように、どのような場所および／またはどのような方法でも消去カウントを維持できる。1つの非限定的な例では、消去カウントはフラッシュコントローラ280によって維持される。1つの非限定的な例では、消去カウントはデータ構造内に維持される(例えば、不揮発性メモリに保存される。)例えば、データ構造はウェアレベリングの目的にも使用される。ステップS423では、データはメモリブロックの1つ以上から読み戻される。データが読み出される1つ以上のメモリブロックの消去カウントを、例えば、適切なデータ構造における「ルックアップ」に作用することによって決定できる。

30

【0048】

図2A~2B、3A~3Dの議論

40

以下の特性を有する2つのデコーダ(または2つの異なったモードで動作可能な1つのデコーダ)を検討する。

a. 第1のデコーダ(またはデコーダモード)は、より速いおよび／または「より軽い重み」のものであるが、一方では、デコーディングプロセスの結果を生成するのを成功するように保証されてはいない。

b. 第2のデコーダ(またはデコーダモード)は、より遅いおよび／または「より重い重み」のものであるが、いつもデコーディングプロセスの正しい結果を生成するように保証される(または少なくともデコーディングプロセスの正しい結果をより生成しそうである)。

通常、データのエラー数が高ければ高いほど、最初のデコーダはより頻繁に失敗する。

50

残念ながら、メモリからデータを読み戻すときに、読み戻したデータ中にいくつかのエラーがあるのかはいつも先験的に知られているというわけではない。このように、「より重い重み」デコーダ（または単一のデコーダのより重い重みモード）が必要であるかどうか、あるいは「より軽い重み」デコーダで十分であるかどうかはいつも知られているというわけではない。

【0049】

本願発明者は、データが読み出されるメモリブロックの消去カウント数が比較的「低い」場合には、読み戻しているデータが良い「品質」でありそうなことを示し、またステップS615またはS625におけるデータの読み出し中のエラーを修正するための比較的「楽観的」な技法が選択され得ることをここで開示する。このように、この場合には、「より軽い重み」デコーダ（または単一のデコーダのより軽い重みモード）を使用してエラーを修正する「危険を冒す」ことは有利であるかもしれない。そうでなければ、エラーを修正するのは単一のデコーダの「より重い重み」デコーダまたはより重い重みモードを使用することが望ましいかもしれない。

10

【0050】

図2Aをここで参照する。ステップS615では、デコーダはデータが読み出されるブロックの消去カウントにしたがって複数の候補デコーダから選択される。

ステップS619では、選択されたデコーダのみを使用し、選択されていない「排除された」デコーダを使用せずに、エラーは修正される。

20

【0051】

図2Bをここで参照する。ステップS625では、特定のデコーダのモードはデータが読み出されるブロックの消去カウントにしたがって複数の候補デコーダモードから選択される。

ステップS629では、選択されたデコーダモードのみを使用し、選択されていない「排除された」デコーダモードを使用せずに、エラーは修正される。

【0052】

図3Aをここで参照する。

ステップS651では、データが読み出されるブロックの消去カウントにしたがってより軽い重みデコーダ（すなわち、複数の候補デコーダ内のデコーダ）を使用してエラー修正の試行を行うかどうかの決定が行われる。

30

データが読み出されるソリッドステートメモリのブロックの消去カウントが比較的「低い」場合、読み戻しているデータが良い「品質」でありそうなことを示し、比較的「楽観的」なエラー修正ストラテジーを採用できる。このように、このシナリオでは、ステップS655でより軽い重みデコーダを使用してエラー修正の試行が行われる。より軽い重みデコーダが成功した場合（ステップS659を参照する）には、より重い重みデコーダを使用してエラー修正の試行を行う必要はない（ステップS667を参照する）。より軽い重みデコーダがエラー修正に失敗したのであれば、次により重い重みデコーダを使用してエラー修正の試行がS663で行われる。

さらに、ステップS651を参照すると、データが読み出されるブロックの消去カウントが比較的「高い」場合、読み戻しているデータが「プア」な品質でありそうであり、比較的「悲観的」なエラー修正ストラテジーを採用できることを示していることを言及している。このシナリオでは、ステップS665をスキップして、より軽い重みデコーダよりもむしろより重い重みデコーダを使用してエラー修正の試行を行うこと（ステップS663で）が可能である。

40

【0053】

図3Bをここで参照する。

ステップS851では、消去カウントにしたがって、より速いデコーダ（すなわち、複数の候補デコーダ内のデコーダ）を使用してエラー修正の試行を行うかどうかを決定する。

消去カウントが「低く」、比較的「楽観的」なエラー修正ストラテジーが採用される場

50

合には、ステップS 8 5 5でより速いデコーダを使用してエラー修正の試行を行うことができる。より速いデコーダが成功した場合（ステップS 8 5 9を参照する）には、より遅いデコーダを使用してエラー修正の試行を行う必要はない（ステップS 8 6 7を参照する）。より速いデコーダがエラー修正に失敗したのであれば、次により遅いデコーダを使用してエラー修正の試行がS 8 6 3で行われる。

さらに、ステップS 8 5 1を参照すると、消去カウントが「高く」、比較的「悲観的」なエラー修正ストラテジーが採用される場合、ステップS 8 6 5をスキップして、より速いデコーダよりもむしろより遅いデコーダを使用してエラー修正の試行を（ステップS 8 6 3で）行うことが可能であることを言及している。

【0054】

図3Cをここで参照する。

ステップS 6 7 1では、ブロックの消去カウントにしたがって、より軽い重みモード（すなわち、特定のデコーダの複数の候補デコーダモード内のデコーダ）を使用してエラー修正の試行を行うかどうかを決定する。

消去カウントが「低く」、比較的「楽観的」なエラー修正ストラテジーが採用される場合には、ステップS 6 7 5で、より軽い重みモードを使用してエラー修正の試行を行うことができる。より軽い重みモードが成功した場合（ステップS 6 7 9を参照する）には、より重い重みモードを使用してエラー修正の試行を行う必要はない（ステップS 6 8 7を参照する）。より軽い重みモードがエラー修正に失敗したのであれば、次により重い重みモードを使用してエラー修正の試行がS 6 8 3で行われる。

さらに、ステップS 6 7 1を参照すると、消去カウントが「高く」、比較的「悲観的」なエラー修正ストラテジーが採用される場合には、ステップS 6 7 5をスキップして、より軽い重みモードよりもむしろより重い重みモードを使用してエラー修正の試行を（ステップS 6 8 3で）行うことが可能であることを言及している。

【0055】

図3Dをここで参照する。

ステップS 8 7 1では、メモリブロックの消去カウントの結果にしたがって、より速いデコーダモード（すなわち、複数の候補デコーダモード内のデコーダ）を使用してエラー修正の試行を行うかどうかを決定する。消去カウントが「低く」、比較的「楽観的」なエラー修正ストラテジーが採用される場合には、ステップS 8 7 5で、より速いデコーダモードを使用してエラー修正の試行を行うことができる。より速いデコーダモードが成功した場合（ステップS 8 7 9を参照する）には、より遅いデコーダモードを使用してエラー修正の試行を行う必要はない（ステップS 8 8 7を参照する）。より速いデコーダモードがエラー修正に失敗したのであれば、より遅いデコーダモードを使用してエラー修正の試行がS 8 8 3で行われる。

さらに、ステップS 8 7 1を参照すると、消去カウントが「高く」、比較的「悲観的」なエラー修正ストラテジーが採用される場合には、ステップS 8 6 5をスキップして、より速いデコーダモードよりもむしろより遅いデコーダモードを使用してエラー修正の試行を（ステップS 8 6 3で）行うことが望ましいかもしれないことを言及している。

【0056】

図4A～4Bの議論

図4A～4Bは、以下の技法のブロック図である。すなわち、（i）（ステップS 5 1 9で）エラー修正の第1の試行が行われ、（i i）何らかの時点で、第1の試行がエラー修正に成功しなかった場合、（ステップS 5 3 9で）第1の試行を中止して、エラー修正の第2の試行が行われる。図4A～4Bは、第1の試行が失敗であったと見なされて中止された場合を決定するためのポリシーを確立し実行することに関する。この決定は些細なことではない。（ステップS 5 1 9で）エラー修正の第1の試行が「時期尚早」に失敗であると見なされた場合、不必要に遅いエラー修正プロセスおよび／またはリソース消費をもたらすかもしれない。なぜならば、第1の試行が行われたときにその過程を実行することが容認されていれば、成功した／満足したであろう場合に、エラー修正の1つ以上の「

10

20

30

40

50

余分／不要」な試行が行われてしまうからである。その一方で、第２の試行への遷移が不必要に遅れて「あまりにも余分」な時間／リソースが第１の試行を行うのに費やされた場合、また不必要に遅いエラー修正プロセスおよび／またはリソース消費をもたらすことになるかもしれない。

【００５７】

図４Ａ～４Ｂでは、エラー修正されるデータが読み出されるメモリブロックの消去カウントにしたがって、いつ第１の試行が失敗であると見なされる（その結果、中止される）かを決定するためのポリシーが確立され実行され得るかを開示している。

図４Ａ～４Ｂでは、第１の試行は、ステップＳ５１５で確立されたエラー修正パラメータの「第１のセット」を使用して、ステップＳ５１９で行われ、また第２の試行は、ステップＳ５３５で確立されたエラー修正パラメータの「第２のセット」を使用して、ステップＳ５３９で行われる。図４Ａと図４Ｂの両方とも、第１の試行がステップＳ５２３で成功であると決定された場合には、ステップＳ５３９で追加の試行を行う必要はない（ステップＳ５２７を参照する）。

【００５８】

図４Ａは、ステップＳ５１９でのエラー修正の第１の試行（その結果、第１の試行を中止する）から、ステップＳ５３９でのエラー修正の第２の試行にいつ「遷移」するかを決定する基準に関する。

メモリブロックの消去カウントが「低い」場合、次に第１の試行が成功するという比較的「高い」見込みがあるという仮定にしたがって、楽観的アプローチを採用できる。このように、この場合には、ステップＳ５１１で決定する「モード遷移条件」は、比較的「速い」かまたは「軽い」エラーを修正する試行を開始するために比較的「寛容な」ものであるかもしれない（試行がどの程度「速い」か「遅い」か「重い重み」か「軽い重み」かはステップＳ５１５および／またはステップＳ５３５で少なくとも一部は確定できる）。逆に、消去カウントが「高い」場合、次に悲観的なアプローチが採用されるかもしれない、これらの初期の試行がとにかく失敗するという高い可能性があるものと仮定されるので、ステップＳ５１１で決定する「モード遷移条件」は比較的「速い」かまたは「軽い」エラー修正試行を開始するために「よりけちな」ものであるかもしれない。

【００５９】

「モード遷移条件」は、第１の試行が失敗して、エラー修正のための「第２の」試行または「異なる」試行を行うために、異なる「モード」への遷移が保証されることを決定するための条件である。図４Ａに示されているように、モード遷移条件がステップＳ５３１でトリガされる場合には、エラー修正の新しい試行がステップＳ５３５とＳ５３９とで行われる。条件が第１の試行の間のいかなる特定の時間にもトリガされない場合には、第１の試行がステップＳ５２５に続く。

【００６０】

１つの例では、「遷移条件」は「タイムアウト条件」に関連する。すなわち、ステップＳ５３１では、ステップＳ５１９とＳ５３４の第１の試行が、その試行を「中止」せず、異なるエラー修正パラメータで「新しい試行」も開始せずに、「実行」することがどのくらいの時間許容されるかを決定する。別の例では、第１の試行自体は繰り返しルーチンを使用し、「遷移条件」は「繰り返し条件数」に関する。すなわち、「使用している繰り返しルーチン」のどのくらいのステップが、この「使用している繰り返しルーチン」が失敗であると見なされて、「新しい」または第２の試行が選択され中止される前に、エラーを成功裏に修正せずに実行するのを許容されているかということである。さらに別の例では、「遷移条件」は、この第１の試行を中止する前に、第１の試行で「消費」するかまたは使用できるＣＰＵサイクル数に関する。

【００６１】

図４Ａ（および図４Ｂ）の教示は、「第１の試行」がステップＳ５１９で行われ、第１の試行が「成功しなかった」場合には、「第２の試行」が後で行われるというどんな「繰り返しルーチン」にも適用される。

このように、異なった実施形態では、「モード遷移」が以下のように言及される。すなわち、(i) 特定のデコーダのより軽いモードからより重いモードに移行するか、あるいは(ii) 複数のデコーダから選択された第1のデコーダが使用される第1の「モード」から、複数のデコーダから選択された第2のデコーダが使用される第2の「モード」に移行するか、あるいは(iii) ソフトデコーダに関して、ビット確率値の第1のセットに関連している第1のモードから、ビット確率値の第2のセットに関連している第2のモードに移行するか、あるいは(iv) ソフトビットの第1の数（すなわち、非負整数）が使用される第1のモードから、ソフトビットの第2の数が使用される第2のモードに移行し、また(v) 第1のデコーダバス幅を有する第1のモードから、第2のデコーダバス幅を有する第2のモードに移行する（図6Bを参照する）。

10

この提供されたリストは、例示であって、完全なリストではないことが理解されるべきである。

【0062】

図4Bでは、データが読み出される（すなわち、エラー修正が試行される）メモリブロックの消去カウントにしたがった「バジェット」を伴った第1の試行が提供されている。このバジェットは、最初の試行で「消費できる」時間バジェット、CPUサイクル数バジェット、繰り返し数バジェットまたはいかなる他のタイプの「リソース」バジェットであってもよい。別の実施形態では、このバジェットは、第1の試行に使用される「中間結果」を保存するために割り当てることができる最大の量の揮発性メモリであるかもしれない。

20

【0063】

消去カウントが比較的「低い」場合、ステップS561で「より大きい」バジェットを割り当てることができ、消去カウントが比較的「高い」場合には、「より小さい」バジェットを割り当てることができる。ステップS561で、この繰り返しステップのエラー修正リソースバジェットを決定し、またステップS565で、このリソースバジェットが消耗した場合にはその決定がなされる。バジェットが消耗した場合には、エラー修正の新しい試行がステップS535とS539とで行われ、バジェットが第1の試行の間まだ消耗していない場合には、エラー修正の第1の試行がステップS525で持続する。

【0064】

図5A～5Bの議論

30

図5A～5Bは、エラー修正のために読み出されるソフトビット数を決定するための技法に関する（図5A～5Bに言及した前述した議論も参照する）。図5AのステップS419で、エラー修正のために初期の試行用に読み出されるソフトビット数が決定される。このソフトビット数は、エラー修正されるべきデータが（ステップS423で）読み出される1つ以上の対象メモリブロック（ステップS415で決定されている）に関連している消去カウントにしたがって決定される。

ステップS427で、決定されたソフトビット数が読み出され、またステップS431で、ソフトビットを使用してデータ（すなわち、ステップS423で読み出されたデータ）中のエラー修正の試行が行われる。試行が成功している場合（ステップS439を参照する）には、エラー修正プロセスを持続させる必要はない（ステップS443を参照する）。試行が成功しなかった場合には、ステップS435で1つ以上の追加のソフトビットを読み出すことができる。

40

図5Aにおけるステップの順番（および本願の開示によって示されているすべての図）は限定的ではないことに留意するべきである。例えば、ステップS423のソリッドステートメモリの対象ブロックからのデータ読み出しを、ステップS419の決定の前に行うこともできる。

【0065】

図5Bをここで参照する。オプションのステップS461で、ソフトビット数Nが読み出され、Nは非負整数である（Nがゼロであれば、ステップS461をスキップし、またステップS431でエラー修正の第1の試行はソフトビットを使用しない）。ステップS

50

4 3 1で、（ソフトビットが使用可能であれば、ソフトビットを使用して）エラー修正の試行は行われる。この試行が成功した場合（ステップS 6 6 5を参照する）には、エラー修正プロセスを持続させる必要はない（ステップS 4 6 9を参照する）。この試行が成功しなかった場合には、（すなわち、エラー修正の次の試行における使用のために）ソフトビットがいくつ読み出されるかの決定をステップS 4 7 3で行う。この数は、ステップS 4 1 5とS 4 2 3の1つ以上の対象メモリブロックに関連している消去カウントにしたがって、決定される。

ステップS 4 7 7で、ソフトビット数（すなわち、ステップS 4 7 3で決定されている）が読み出される。これらのソフトビットはステップS 4 3 1で使用される。

【0 0 6 6】

10

図 6 A の 議 論

図 6 A は、データが読み出されるメモリのブロックの消去カウントにしたがってエラーを修正するための技法のフローチャートである。

ソフトデコーダは、符号語の各ビットに確率を割り当てて（例えば、ビットの値が1であり0ではないという確率）、また他のビットの現在の確率にしたがってそれぞれのビットが変更される確率のそれぞれの複数の繰り返しを実行することによって動作する。

ソフトデコーダではないどんなデコーダもここで「代数的なデコーダ」と呼ばれる。

【0 0 6 7】

このタイプの計算は、各ビットがピアに「メッセージをパスする」ので、「メッセージパッシング」技法を使用すると言われる。そのようなデコーダをインプリメントする主要な設計の決定は、メッセージパッシングのバス幅である。広いバス（例えば、10ビット）を使用するのは、アルゴリズムがエラーに対処できるのであれば、真にアルゴリズムが正しいデータに収束することを保証する。しかし、その一方で、そのような設計ではデコーダの消費電力が高い。狭いバス（例えば、3ビット）を使用すれば、より低い消費電力を提供することができるが、比較的大きいエラー数の場合、（データとエラーの同じパターンが、対応する広いバスデコーダでは正しくデコードされるものであったとしても）デコーダは正しいデータに収束するのを失敗するかもしれない。メッセージパッシングに使用されるビット数はデコーディングプロセスの消費電力に主要な影響を与える、より高いビット数であれば、より高い消費電力となる。

20

【0 0 6 8】

30

この現象の理由の1つに、ソフトデコーダのメッセージとチャネル入力（ソフト値）がデコーディング操作の間、大きい消費電力のRAMに保存されるということが挙げられる。例えば、1メッセージあたり10ビットの代わりに3ビットを使用すると、RAMの70%をシャットダウンすることができる。消費電力の節約の別の本元は、これらのパッシングメッセージを処理する処理装置である。10ビットの代わりに3ビットで計算を実行することが、より小さく、そしてより消費電力の低いユニットを要するのは明らかである。

【0 0 6 9】

デコーディングの始めに「メッセージパッシング」におけるビット数を設定できるソフトデコーダをインプリメントすることは可能である。このように、いくつかの実施形態では、データが読み出されるブロックの消去カウントにしたがって、デコーダバス幅のサイズが選択される。「楽観的な状況」を示している、消去カウントが「低い」場合には、それにより、より小さいデコーダバス幅のサイズを選択してエラーを修正することができる。そうでなければ、より大きいデコーダバス幅のサイズを選択するのが望ましいかもしれない。

40

このことは、図 6 A に表を用いて示されている。ステップS 7 5 1で、データが読み出されるブロックの消去カウントにしたがって、デコーダバス幅が選択される。ステップS 7 5 5で、選択されたデコーダバス幅の値にしたがって、エラーが修正される。

【0 0 7 0】

図 6 B の 議 論

50

図 6 B のステップの S 4 1 1、S 4 2 3、および S 7 5 1 は、図 6 A と同じである。

ステップ S 7 7 5 で、ステップ S 7 5 1 で選択された、選択されたデコーダバス幅を使用してエラー修正の試行が行われる。この試行が成功した場合（ステップ S 7 7 9 を参照する）には、エラー修正プロセスを持続させる必要はない（ステップ S 7 8 7 を参照する）。そうでなければ、この試行が成功しなかった場合（ステップ S 7 8 3 を参照する）に、新しいデコーダバス幅が選択され、エラー修正の新しい試行が行われる。

【0071】

追加の議論

ここで開示されたいかなる技法も「デバイス側」上にまたは「ホスト側」上でインプリメントすることができることに留意するべきである。1つの特定の実施形態で、ここで開示されたいかなるステップまたはそのようなステップの組み合わせも（消去カウントを維持するステップ、データまたはソフトビットを読み出すステップ、エラーを修正するステップ、エラー修正の試行を行うステップ、デコーダまたはモードまたはソフトビット数を選択するステップ、特定のデコーダまたはデコーダモードを使用するかどうかを決定するステップ、デコーダバス幅を選択するステップ、遷移条件またはリソースバジェットを決定するかまたは実施するステップなどのステップを含むが、それらに限定されるものではない）、フラッシュデバイス 260（またはソリッドステートメモリを含むその他のデータストレージデバイスでも）のデバイスコントローラ 280 によって実行することができる。例えば、周辺ストレージデバイス 260* のデバイスコントローラによって実行してもよい。

【0072】

いくつかの実施形態では、「デバイス側」でデータ中のエラーが修正された後に、エラー修正データはストレージデバイス 260* からホストデバイス 310 へそれぞれのデバイス間インターフェイスを介して送信され得る。（ステップ S 4 2 3 のデータ読み出しとあらゆるエラー修正とともに）、ソリッドステート不揮発性メモリ（例えば、フラッシュメモリ 270）に保存されたデータを読み出すために、ホスト 310 から周辺ストレージデバイス 260* に送信された要求に応答して行われ得る。

【0073】

デバイスコントローラ 280 は、いかなるソフトウェアおよび／またはファームウェアおよび／またはフィールドプログラマブル論理アレイ（FPGA）素子、配線論理素子、フィールドプログラマブルゲートアレイ（FPGA）素子、および特定用途向け集積回路（ASIC）素子を含むが、これらに限定されないハードウェア素子も含むことができる。縮小命令セットコンピュータ（RISC）のアーキテクチャおよび／または複合命令セットコンピュータ（CISC）のアーキテクチャを含むがこれらに限定されないいかなる命令セットのアーキテクチャも、コントローラ 280 で使用できる。

【0074】

ハードウェア、ハードウェアが実行するコンピュータ可読コードモジュール、およびファームウェアのどのような組み合わせにおいてもここで開示されたいかなる技法をインプリメントすることができる。

本発明の実施形態は、開示されたいかなる技法をもインプリメントするための装置またはコンピュータ媒体を提供する。

本願明細書で引用されたすべての参考文献は、その全体が参照により援用されている。参考文献からの引用がこの参考文献が従来技術であるという認容を構成するものではない。

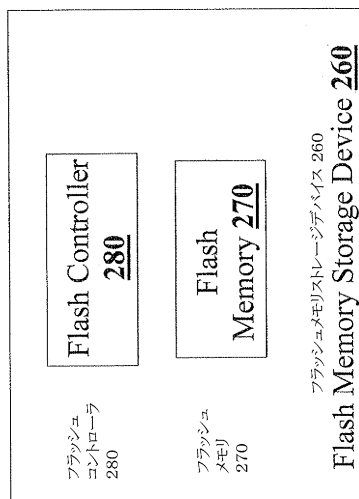
前に説明した実施形態のいずれもが、図に関連して前に説明した操作をインプリメントするための、コンピュータ可読媒体上の、命令および／またはデータを、受信したり、送信したり、あるいは保存したりすることをさらに含むことができることにさらに留意するべきである。一般的には、コンピュータ可読媒体は、ストレージメディアまたは磁気、フラッシュまたは光学メディアなどのメモリメディア、例えばディスクまたは CD-ROM、RAM/ROM などの揮発性または不揮発性メディアを含むことができ、またネットワ

ークおよび／またはワイヤレスリンクなどの通信媒体を通して転送される、電気信号、電磁気信号またはデジタル信号などの伝送メディアまたは信号を含むことができる。

【0075】

このように、前に説明した実施形態に関して、添付の特許請求の範囲で列挙されている範囲と趣旨から逸脱せずに、様々な同等物、それらの改変、修正および改良が可能であることが当業者にとって明らかである。特に、種々の実施形態が本願明細書に記載されたものの以外の特徴の組み合わせを含むことができる。したがって、特許請求の範囲は前述した議論に限定されるものではない。

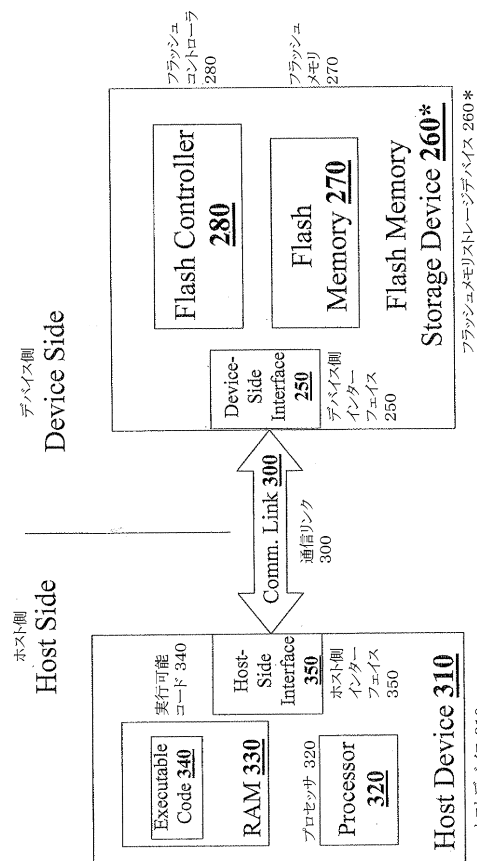
【図 1 A】



従来技術
PRIOR ART

FIG. 1A

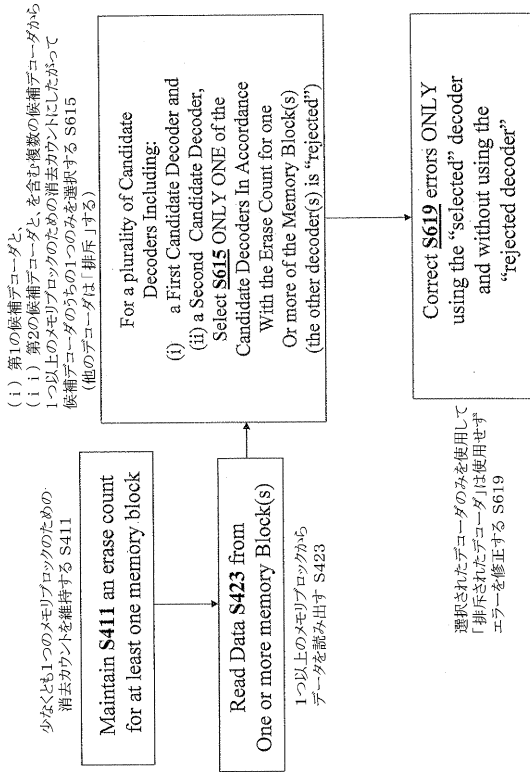
【図 1 B】



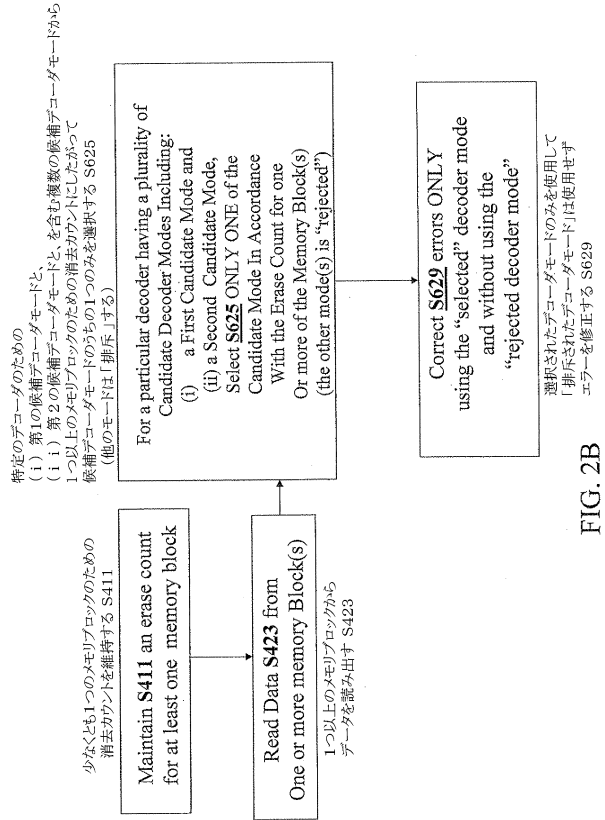
従来技術
PRIOR ART

FIG. 1B

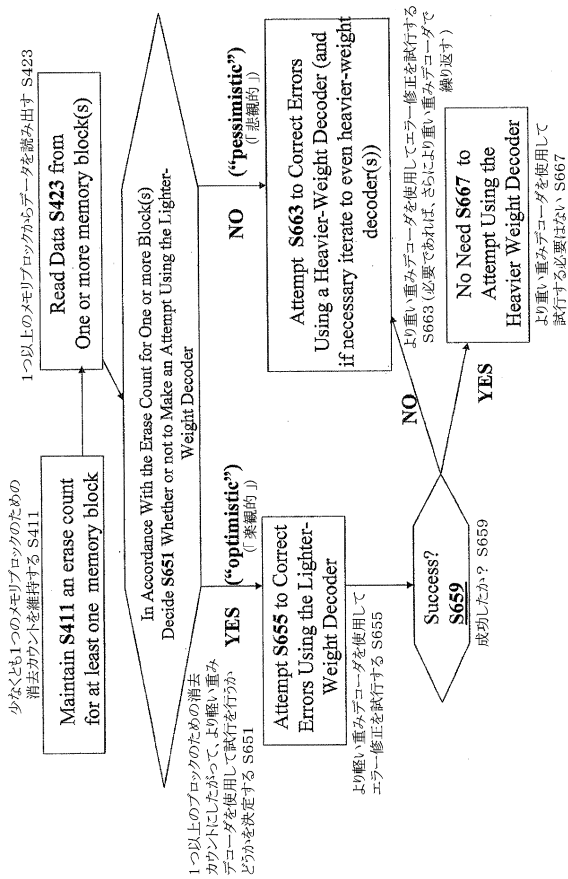
【図 2 A】



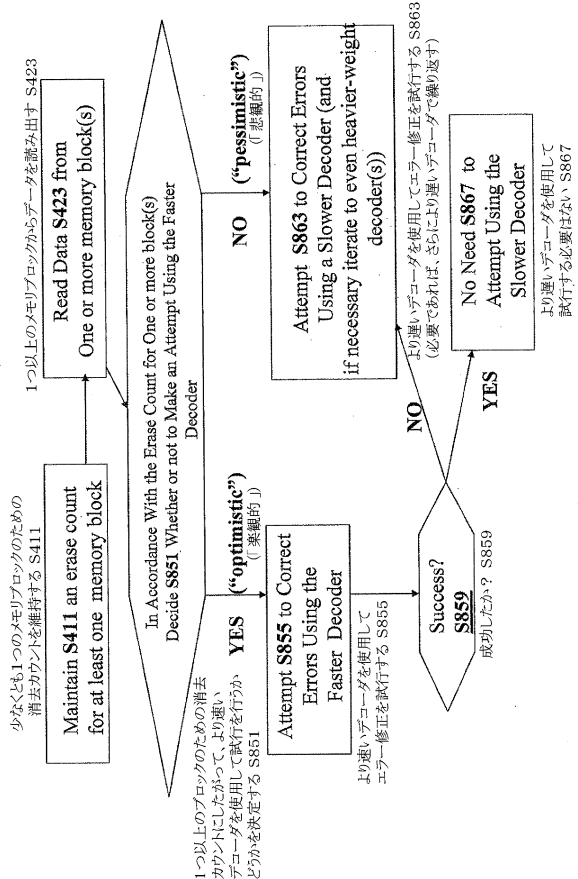
【図 2 B】



【図 3 A】



【図 3 B】



【図 3 C】

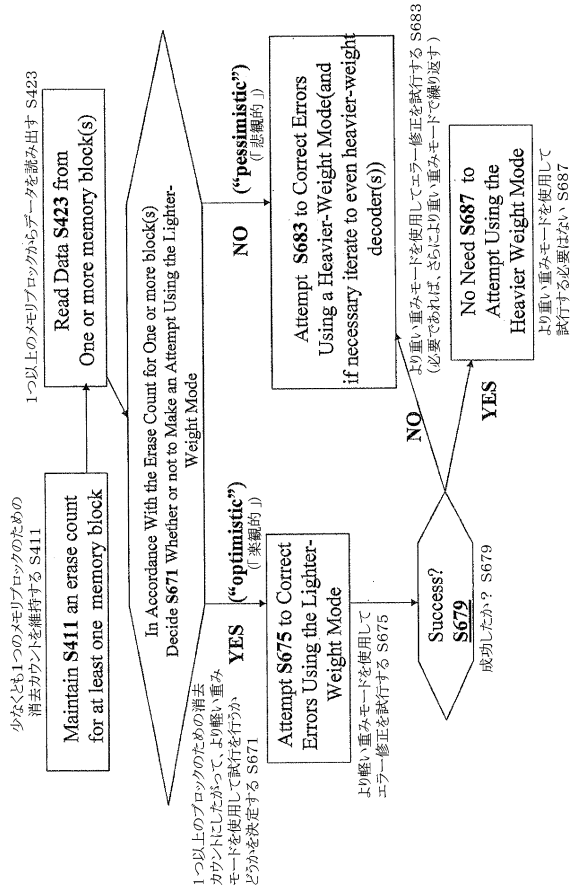


FIG. 3C

【図 3 D】

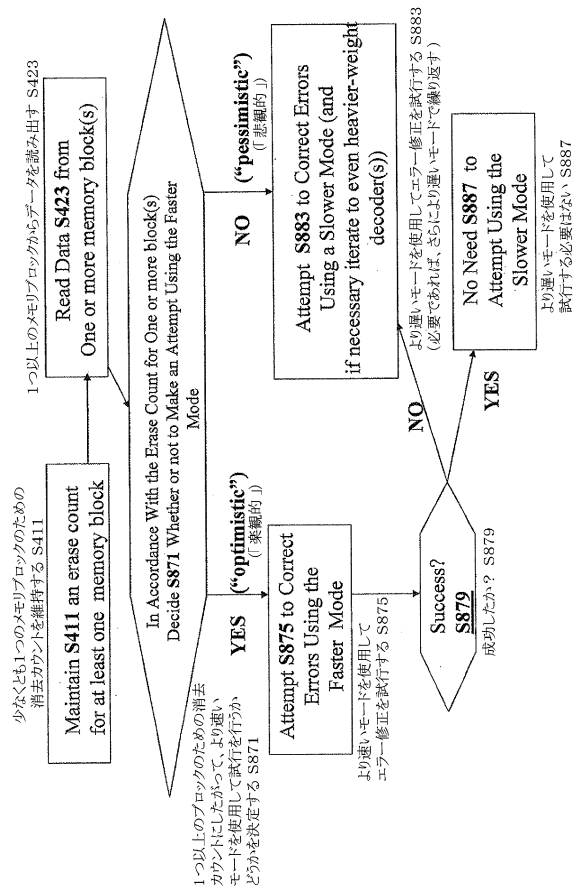


FIG. 3D

【図 4 A】

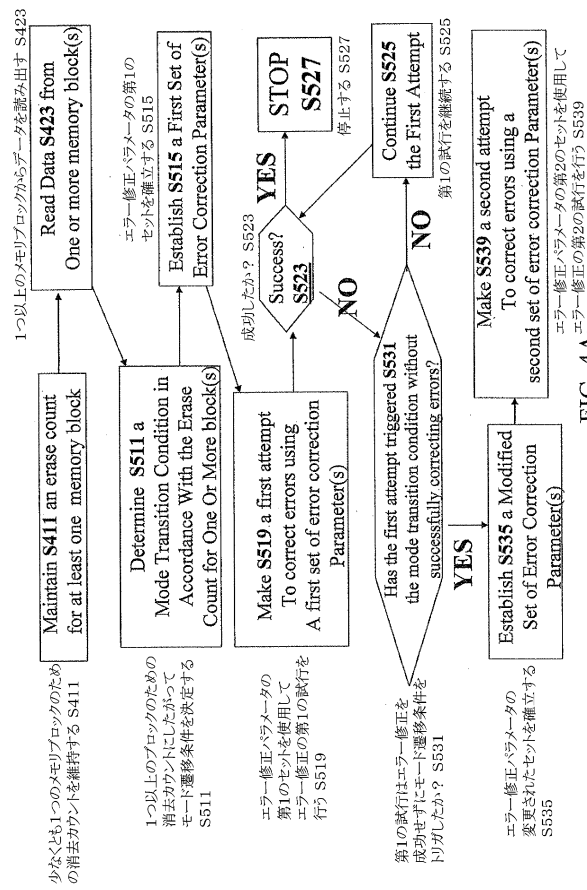


FIG. 4A

【図 4 B】

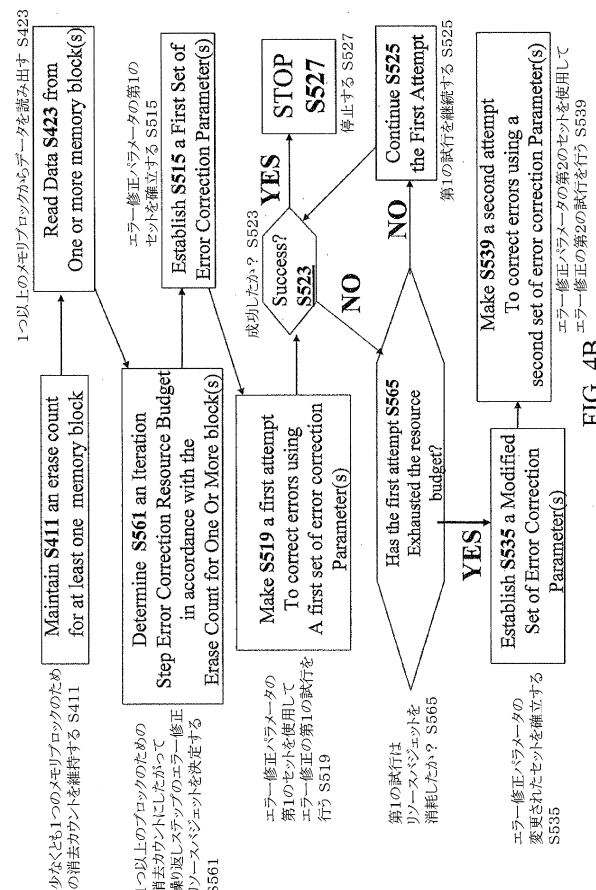


FIG. 4B

【図 5 A】

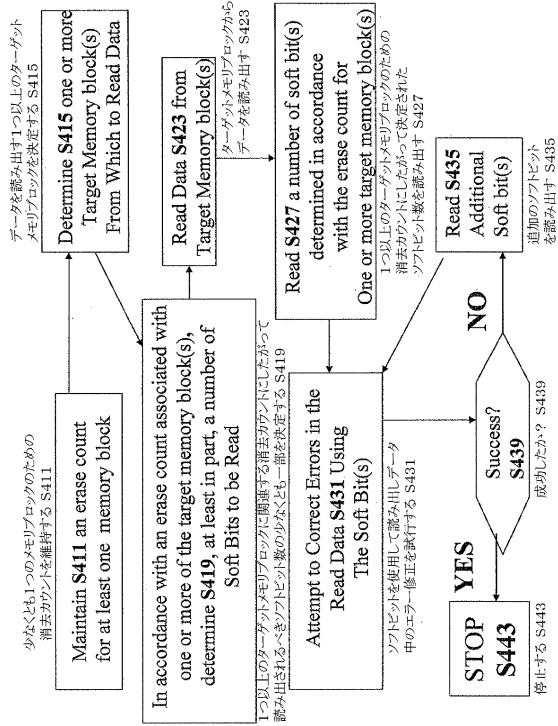


FIG. 5A

【図 5 B】

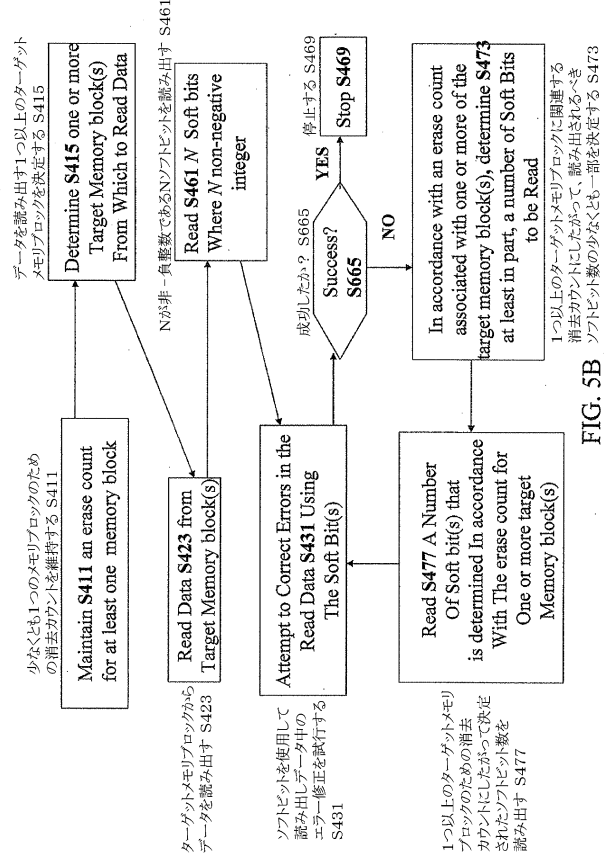


FIG. 5B

【図 6 A】

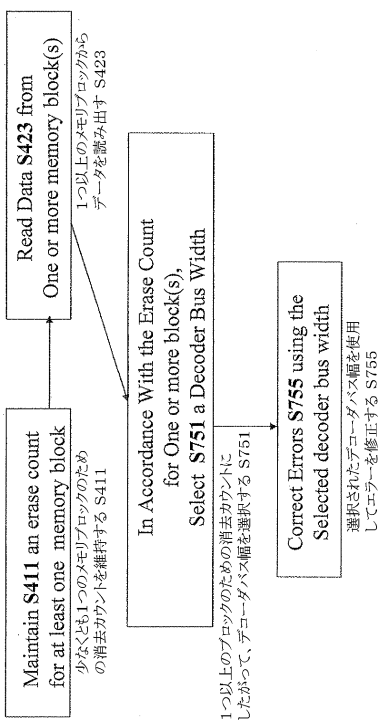


FIG. 6A

【図 6 B】

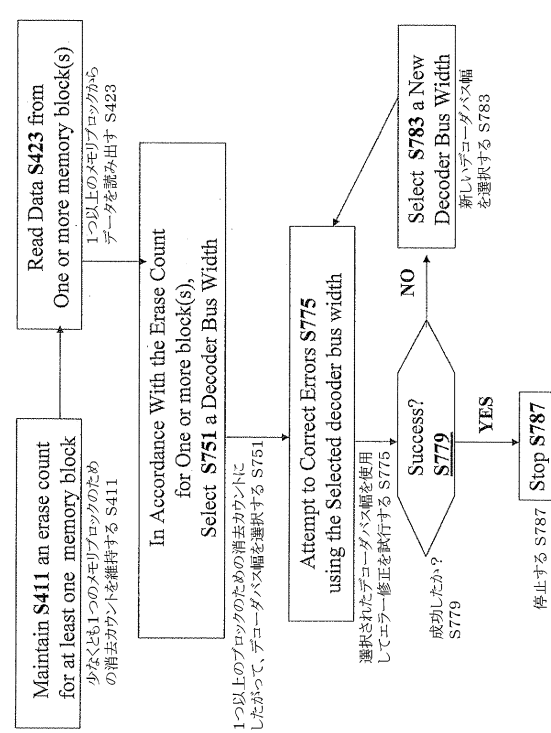


FIG. 6B

フロントページの続き

(74)代理人 110000110

特許業務法人快友国際特許事務所

(72)発明者 アルロッド, イダン

イスラエル国、4 6 6 4 3、ヘルツェリヤ、ハエッシェル ストリート 7 0

(72)発明者 シャロン, エラン

イスラエル国、7 5 4 9 3、リション レジオン、ハダガン ストリート 1 6 / 1

(72)発明者 ラッセル, メナハム

イスラエル国、4 4 5 3 1、コハブ イエール、ドルブ ストリート 4

審査官 酒井 恭信

(56)参考文献 特開2004-164634 (JP, A)

欧州特許出願公開第01424631 (EP, A1)

米国特許出願公開第2004/0083333 (US, A1)

特表2008-508632 (JP, A)

国際公開第2006/013529 (WO, A1)

国際公開第2007/057885 (WO, A1)

特開平07-264077 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 1 2 / 1 6