

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5523562号
(P5523562)

(45) 発行日 平成26年6月18日(2014.6.18)

(24) 登録日 平成26年4月18日(2014.4.18)

(51) Int.Cl.

F I

G 0 6 F 12/16 (2006.01)

G 0 6 F 12/16 3 1 0 B

請求項の数 6 (全 17 頁)

(21) 出願番号 特願2012-517703 (P2012-517703)
 (86) (22) 出願日 平成22年6月23日 (2010.6.23)
 (65) 公表番号 特表2012-532372 (P2012-532372A)
 (43) 公表日 平成24年12月13日 (2012.12.13)
 (86) 国際出願番号 PCT/US2010/039699
 (87) 国際公開番号 W02011/002656
 (87) 国際公開日 平成23年1月6日 (2011.1.6)
 審査請求日 平成25年4月8日 (2013.4.8)
 (31) 優先権主張番号 12/494,186
 (32) 優先日 平成21年6月29日 (2009.6.29)
 (33) 優先権主張国 米国 (US)

早期審査対象出願

前置審査

(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、タワー・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 ジ・アブレウ、マニエル 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 2166

最終頁に続く

(54) 【発明の名称】 ストレージデバイス内のエラーデータを追跡するシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

エラーデータを追跡する方法であって、
 メモリから読み出されたデータを受信するステップと、
 複数のエラーおよび複数の場所を特定するために、受信したデータにエラー検出演算を
 実行するステップであって、複数のエラーの各エラーが複数の場所の対応する場所に関連
 づけられるステップと、

複数のエラーの各々に関連づけられた複数の場所に基づいて、チェックサムを生成する
 ステップと、

前記チェックサムを前記メモリに格納するステップと、

第1の時間に実行された第1のエラー検出演算に基づいて第1のチェックサムを格納し
 、第2の時間に実行された第2のエラー検出演算に基づいて第2のチェックサムを格納す
 るステップと、

前記第1のチェックサムと前記第2のチェックサムとを比較するステップと、

前記第1のチェックサムと前記第2のチェックサムとの決定された差が第1の所定のし
 きい値を満たす場合、第1の動作を開始し、前記第1のチェックサムと前記第2のチェッ
 クサムとの決定された差が第2の所定のしきい値を満たす場合、第2の動作を開始するス
 テップと、

を含む方法。

【請求項 2】

10

20

エラーデータを追跡する方法であって、
メモリから読み出されたデータを受信するステップと、
複数のエラーおよび複数の場所を特定するために、受信したデータにエラー検出演算を
実行するステップであって、複数のエラーの各エラーが複数の場所の対応する場所に関連
づけられるステップと、

複数のエラーの各々に関連づけられた複数の場所に基づいて、チェックサムを生成する
ステップと、

前記チェックサムを前記メモリに格納するステップと、
第 1 の時間に実行された第 1 のエラー検出演算に基づいて第 1 のチェックサムを格納し
、第 2 の時間に実行された第 2 のエラー検出演算に基づいて第 2 のチェックサムを格納す
るステップと、

10

前記第 1 のチェックサムと前記第 2 のチェックサムとを比較するステップと、
前記メモリの不良の可能性のあるワード線を特定するために、前記第 1 のチェックサム
と前記第 2 のチェックサムとの決定された差が第 1 の所定のしきい値を満たす場合、前記
第 1 のチェックサムおよび前記第 2 のチェックサムのうちの少なくとも 1 つによって表さ
れる場所情報を評価するステップと、

を含む方法。

【請求項 3】

請求項 2 記載のエラーデータを追跡する方法であって、
前記メモリの不良の可能性のあるワード線が、前記不良の可能性のあるワード線ノズル
の一端に、複数のエラーのすべてが生じていることを示す場所情報に基づいて特定される
方法。

20

【請求項 4】

エラーデータを追跡する方法であって、
メモリから読み出されたデータを受信するステップと、
複数のエラーおよび複数の場所を特定するために、受信したデータにエラー検出演算を
実行するステップであって、複数のエラーの各エラーが複数の場所の対応する場所に関連
づけられるステップと、

複数のエラーの各々に関連づけられた複数の場所に基づいて、チェックサムを生成する
ステップと、

30

前記チェックサムを前記メモリに格納するステップと、
第 1 の時間に実行された第 1 のエラー検出演算に基づいて第 1 のチェックサムを格納し
、第 2 の時間に実行された第 2 のエラー検出演算に基づいて第 2 のチェックサムを格納す
るステップと、

前記第 1 のチェックサムと前記第 2 のチェックサムとを比較するステップと、
前記第 1 のチェックサムと前記第 2 のチェックサムとの差が第 2 の所定のしきい値を満
たすという決定にตอบสนองして、複数のエラーのうちのあるエラーがランダムであるかを決定
するステップと、

を含む方法。

【請求項 5】

請求項 4 記載の方法において、
前記エラーがランダムであるという決定にตอบสนองして、動作を開始しないステップをさら
に含む方法。

40

【請求項 6】

エラーデータを追跡する方法であって、
メモリ領域の第 1 のエラー場所セットに関連づけられた第 1 のチェックサムをメモリか
ら取得するステップと、

前記第 1 のチェックサムと、前記メモリ領域の第 2 のエラー場所セットに関連づけられ
た第 2 のチェックサムとを比較して、比較結果をもたらすステップと、

前記比較結果に基づいて、前記メモリ領域に対して動作を開始するステップと、

50

前記第 1 のチェックサムと前記第 2 のチェックサムとの決定された差が第 1 の所定のしきい値を満たす場合、第 1 の動作を開始し、前記第 1 のチェックサムと前記第 2 のチェックサムとの決定された差が第 2 の所定のしきい値を満たす場合、第 2 の動作を開始するステップと、

を含む方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータメモリシステムに関し、より詳しく言えば、データストレージデバイスのエラー検出およびエラー場所追跡に関する。

10

【背景技術】

【0002】

エラー検出訂正プロセスは、メモリストレージデバイスおよび通信チャネル内のデータの完全性の維持を促すプロセスである。例えば、エラー訂正コード（ECC）は、コンピュータメモリサブシステムの欠陥検出のために一般に使用される。ECCは、メモリの格納データに対して 1 つ以上のエラーを特定するための冗長データを含む。例えば、「ソフト」エラーは、典型的に、送電障害によって生じるのに対して、「ハード」エラーは、メモリデバイスが原因の場合もある。

ストレージデバイスの回路の縮小化および動作電圧の低下に伴い、エラーを特定し訂正するプロセスの精度および効率を高めることが必要とされている。

20

【発明の概要】

【0003】

エラーを特定し訂正するための改良されたシステムおよび方法が開示される。メモリの領域に対してエラー場所セットを各々特定する多数のエラー訂正コード演算が実行されてもよい。エラー場所セットは、先行エラー場所と比較されてもよく、適切な動作が開始されてもよい。エラー場所セットは、例えば、チェックサムを介して、エラー場所情報を保存するように符号化されてもよい。エラー場所セットに関連するチェックサムが、各エラー訂正コード演算に対して決定されてもよい。2 つ以上のチェックサムが、傾向を特定するために比較されてもよい。エラーをアドレス指定するための動作が、チェックサム比較の結果に基づいて、または検出された傾向を評価することによって開始されてもよい。

30

【0004】

例えば、経時的に同じチェックサムは、劣化していないビットのハード障害を示しうる。経時的に場所を移動するエラーが決定されれば、エラーは、ハードウェア欠陥ではなく、伝送障害に起因する可能性がある。このようなエラーは、ECCの総数が変わらない限り管理されてもよい。エラーがワード線の端部で特定されれば、破損したワード線などのワード線欠陥が特定されうる。

【図面の簡単な説明】

【0005】

【図 1】エラーデータを追跡するために使用されてもよいデータストレージデバイスのブロック図である。

40

【図 2】データストレージデバイスに関連づけられたエラーデータを追跡するために使用されてもよいシステムのブロック図である。

【図 3】エラー訂正コード演算の第 1 のセットの結果を示す。

【図 4】エラー訂正コード演算の第 2 のセットの結果を示す。

【図 5】エラー訂正コード演算の第 3 のセットの結果を示す。

【図 6】エラーデータを追跡する方法のフローチャートを示す。

【図 7】エラーデータを追跡し格納する方法のフローチャートを示す。

【図 8】エラーデータを追跡し評価する方法のフローチャートを示す。

【図 9】エラーデータを特定し訂正する方法のフローチャートを示す。

【図 10】複数のエラー訂正コード演算で発生したエラーによって示される傾向を特定す

50

る方法のフローチャートを示す。

【発明を実施するための形態】

【0006】

図1を参照すると、エラーデータを追跡するために使用されてもよいデータストレージデバイスの例示的な実施形態が描かれ、概して100で示されている。データストレージデバイス100は、メモリアレイ104に連結されたコントローラ102を含む。コントローラ102は、メモリアレイ104から読み出されたデータに対応するエラー場所データを追跡および比較するように構成され、その比較に基づいて動作を開始してもよい。

【0007】

メモリアレイ104は、フラッシュメモリなどの不揮発性メモリを含んでもよい。メモリアレイ104は、格納データ110を含む代表的なメモリ領域106を含む。メモリ領域106は、ワード線、メモリブロックまたはメモリページであってもよい。ワード線は、メモリアレイ104内の行方向の線を含んでもよい。メモリブロックが、消去可能メモリの最小単位を含んでもよい。メモリページが、書き込み可能なメモリの最小単位を含んでもよい。データ110は、格納データ110内のそれぞれの場所118に関連づけられた検出可能なエラー114を含む。

【0008】

コントローラ102は、メモリアレイ104から読み出されたデータ112を戻す論理通信リンクを介してデータ110を読み出すために、メモリアレイ104にアクセスするように構成されてもよい。コントローラ102のエラー検出モジュール108が、多数のエラー114を検出し、データ110の読み出しに応答してエラー場所118を検出するように構成されてもよい。例えば、エラー検出モジュール108は、エラー訂正コード(ECC)エンジンを含んでもよい。

【0009】

コントローラ102は、チェックサムジェネレータ120をさらに含んでもよい。チェックサムジェネレータ120は、エラー場所データ118に基づいてチェックサムを生成してもよい。例えば、チェックサムジェネレータ120は、第1および第2のエラー訂正コード(ECC)演算中に、エラー検出モジュール108によってそれぞれ生成されたエラー場所データ118の第1のチェックサム124および第2のチェックサム126を出力してもよい。第1のチェックサム124および第2のチェックサム126の各々は、エラー訂正コード演算のそれぞれに対する複数のエラーの各々に関連づけられた複数の場所に基づいたものであってもよい。あるいは、チェックサムジェネレータ120は、多入力シグネチャレジスタ(MISR: Multiple Input Signature Register)を含んでもよい。MISRを組み込むことで、ハードウェア要求を低下しうる。チェックサムジェネレータ120の別の例は、多項式ベースの回路を含んでもよい。多項式ベースの回路は、線形帰還シフトレジスタ(LFSR: Linear Feedback Shift Register)または組み合わせ論理を含んでもよい。チェックサムジェネレータの別の実施例は、累加算器として機能する加算器および累算器を備えた回路である。チェックサムジェネレータ120は、ECCエンジンの一部であってもよい。

【0010】

第1のチェックサム124および第2のチェックサム126は、メモリアレイ104のログファイル122内に格納されてもよい。例えば、コントローラ102は、メモリアレイ104の各領域に対して経時的に生成されたチェックサムを格納してもよい。コントローラ102は、第1のチェックサム124および第2のチェックサム126の各々を異なるタイムスタンプに論理的に関連づけてもよい。例えば、タイムスタンプは、チェックサムの生成中のエラー訂正コード演算に対応してもよい。

【0011】

演算中、コントローラ102は、メモリアレイ104のログファイル122から第1のチェックサム124および第2のチェックサム126を取得してもよい。コントローラ102は、第1のチェックサム124と第2のチェックサム126とをさらに比較してもよ

10

20

30

40

50

く、少なくとも部分的に、第1のチェックサム124と第2のチェックサム126との比較に基づいて、メモリ領域106に関連づけられた動作を開始してもよい。以下、図3～図10に関して説明するように、動作の例には、データの移動、読み出しスクラブの開始、データのリフレッシュ、データブロックを不良データとして指定、データのモニタリングおよび複数のエラーのエラー訂正がありうる。

【0012】

このように、図1は、それぞれのエラー訂正コード演算に対して複数のエラーの各々に関連づけられた複数の場所に各々基づくチェックサムを生成するためのデータストレージデバイス100を示す。第1のチェックサム124および第2のチェックサム126は、エラーデータを効率的に追跡および訂正するために比較される。エラー検出モジュール108およびチェックサムジェネレータ120が別々の要素として示されているが、エラー検出モジュール108およびチェックサムジェネレータ120は、同じ要素（例えば、ECC）に位置してもよい。さらに、第1のチェックサム124および第2のチェックサム126が、ログファイル122から取得されるように示されているが、図2に関して説明するように、他の実施形態において、第1のチェックサム124および第2のチェックサム126の1つ以上が、ログファイル122から取得されなくてもよい。

【0013】

図2を参照すると、ストレージデバイスに関連づけられたエラーデータを追跡するために使用されてもよいシステムの例示的な実施形態が描かれ、概して200で示されている。システム200は、メモリアレイ204に連結されたコントローラ202を含む。コントローラ202はまた、ホストコンピューティングデバイス250に連結される。コントローラ202およびメモリアレイ204は、取り外し可能なデータストレージデバイスのコンポーネントであってもよい。コントローラ202は、先行する読み出し演算から保存されたエラー場所情報と比較して、データ読み出し演算から得られたエラー場所情報を比較することによって、メモリアレイ204の状態を追跡するように構成される。例えば、メモリアレイ204のログファイル216が、チェックサム履歴218を含んでもよい。

メモリアレイ204は、第1のメモリ領域206と、第2のメモリ領域208とを含む。第1のメモリ領域206は、メモリアレイ204のページまたはブロックを表すものであってもよい。

【0014】

コントローラ202は、エラー検出モジュール220を含む。エラー検出モジュール220は、メモリアレイ204から読み出されたデータ212に応答して、多数のエラー214およびエラー場所222を検出するように構成されてもよい。エラー検出モジュール220は、ECCエンジンを含んでもよい。

コントローラ202はまた、エラー場所222を受信し、チェックサム226を生成するように構成されたチェックサムジェネレータ224を含む。チェックサムジェネレータ224は、MISRまたは別の多項式ベースの回路を含んでもよい。例えば、多項式ベースの回路は、LFSRまたは組み合わせ論理を用いて実装されてもよい。複数のエラー訂正コード演算中、チェックサムジェネレータ224は、メモリアレイ204の特定の領域に対応する複数のチェックサムを生成してもよい。チェックサム出力は、ログファイル216のチェックサム履歴218に格納され、その後、1つ以上の先行チェックサムとして取得されてもよい。各チェックサム（例えば、チェックサム226および先行チェックサム234）は、それぞれのエラー訂正コード演算に対してエラーの各々に関連づけられた複数の場所に基づいたものであってもよい。

コントローラ202はまた、エラー検出モジュール220に連結されたプロセッサ240を含む。プロセッサ240は、メモリアレイ204内のログファイル216に対してチェックサム226の格納を開始してもよく、メモリアレイ204内のログファイル216から先行チェックサム234の取得を開始してもよい。

【0015】

プロセッサ240は、チェックサム226と先行チェックサム234とを比較するよう

10

20

30

40

50

に構成されたチェックサム解析モジュール 228 を含む。プロセッサ 240 はまた、動作選択モジュール 230 を含む。動作選択モジュール 230 は、チェックサム 226 と先行チェックサム 234 との比較に基づいて複数の利用可能な動作 232 を自動的に開始するように構成される。動作の例には、データの移動、読み出しスクラブの開始、データのリフレッシュ、データブロックを不良データとして指定、データのモニタリングおよび 1 つ以上のエラーの訂正がある。一例として、チェックサム解析モジュール 228 は、チェックサム 226 および先行チェックサム 234 を取得し比較するように、プロセッサ 240 によって実行されてもよい。チェックサム解析モジュール 228 はまた、比較に基づいた結果を特定するように構成されてもよい。例えば、チェックサム解析モジュール 228 は、エラーが無相関であり、無作為であり、障害に起因することを比較から特定してもよい。別の例において、チェックサム解析モジュール 228 は、エラーが同様の場所のものであり、ハードエラーの可能性のあることを特定してもよい。動作選択モジュール 230 は、チェックサム解析モジュール 228 からの結果を受信し、動作を選択的に開始するために、プロセッサ 240 によって実行されてもよい。

10

【0016】

このように、図 2 は、ストレージデバイスに関連づけられたエラーに関する動作を追跡し、選択的に行うためのシステム 200 を示す。システム 200 により、効率的かつ正確なエラー追跡訂正が可能になる。

【0017】

図 3 は、不良ストレージ要素を示す第 1 および第 2 のエラー訂正コード演算の結果の例を示す。エラー訂正コード演算は、ワード線または図 1 のデータ 110 など、メモリから読み出された別のデータに異なる時間で実行されてもよい。データ 302 は、第 1 のエラー訂正コード演算中に評価されてもよいエラー場所データの例である。例えば、データ 302 は、図 1 のエラー検出モジュール 108 によって生成されるエラー場所情報 118 であってもよく、図 2 のエラー検出モジュール 220 によって生成されたエラー場所情報 222 であってもよい。データ 302 は、メモリのストレージ要素に対応するメモリ要素の場所 306、308、310、312、314、316、318、320 および 322 を含む。一例として、エラー 350 は場所 308 に関連づけられ、エラー 352 は場所 316 に関連づけられる。

20

【0018】

データ 302 は、図 1 のチェックサムジェネレータ 120 など、チェックサムジェネレータ 342 によって受信されてもよい。チェックサムジェネレータ 342 は、エラー 350、352 に関連づけられた場所 308、316 に基づいて、図 1 の第 1 のチェックサム 124 などのチェックサム 344 を生成してもよい。図に示したように、チェックサム 344 は、2 進数「1011」に等しく、この値は 10 進数「11」に等しい。

30

データ 304 は、第 2 のエラー訂正コード演算中に評価されてもよい。データ 304 は、ビット場所 324、326、328、330、332、334、336、338 および 340 を含む。エラー 354 が場所 326 に関連づけられ、エラー 356 が場所 334 に関連づけられる。

データ 304 は、図 1 のチェックサムジェネレータ 120 などのチェックサムジェネレータ 346 によって受信されてもよい。チェックサムジェネレータ 346 は、エラー 354、356 に関連づけられた場所 326、334 に基づいて、図 1 の第 1 のチェックサム 124 などのチェックサム 348 を生成してもよい。図に示したように、チェックサム 348 は、2 進数「1011」に等しく、この値は 10 進数「11」に等しい。

40

【0019】

370 において、チェックサム 344、348 は、データ 302 およびデータ 304 のエラーが同じ場所にあるため、不良ストレージ要素を特定するために比較されてもよい。チェックサム 342 がチェックサム 346 と等しいことによって、不良ストレージ要素が示されうる。例えば、エラー 350、352、354、356 は、第 1 および第 2 のエラー訂正コード演算のデータ 302、304 内の同じ相対場所 308、316、326、3

50

3 4 を有するエラー 3 5 0、3 5 2、3 5 4、3 5 6 に基づいてハードエラーであるように決定されてもよい。

【 0 0 2 0 】

このように、図 3 は、エラー 3 5 0、3 5 2、3 5 4、3 5 6 および関連する場所 3 0 8、3 1 6、3 2 6、3 3 4 を決定するために用いられる第 1 および第 2 のエラー訂正コード演算を示す。エラー場所 3 0 8、3 1 6、3 2 6、3 3 4 に基づいて算出されたチェックサム 3 4 4、3 4 8 は、エラー 3 0 8、3 1 6、3 2 6、3 3 4 の場所が両方のエラー訂正コード演算において同じであることを決定するために比較されてもよい。エラー 3 5 0、3 5 2、3 5 4、3 5 6 の同じ相対場所 3 0 8、3 1 6、3 2 6、3 3 4 は、ハードエラーの存在を示すものであってもよい。

10

【 0 0 2 1 】

図 4 は、ランダムエラーを示す第 1 および第 2 のエラー訂正コード演算の結果の例を示す。エラー訂正コード演算は、図 1 のデータ 1 1 0 など、データ要素に異なる時間で実行されてもよい。データ 4 0 2 は、第 1 のエラー訂正コード演算中に評価されてもよい。例えば、データ 4 0 2 は、図 1 のエラー検出モジュール 1 0 8 によって生成されたエラー場所情報 1 1 8、または図 2 のエラー検出モジュール 2 2 0 によって生成されたエラー場所情報 2 2 2 であってもよい。データ 4 0 2 は、メモリのストレージ要素に対応するビット場所 4 0 6、4 0 8、4 1 0、4 1 2、4 1 4、4 1 6、4 1 8、4 2 0 および 4 2 2 を含む。エラー 4 5 0 が場所 4 0 8 に関連づけられ、エラー 4 5 2 が場所 4 1 0 に関連づけられ、エラー 4 5 4 が場所 4 1 6 に関連づけられる。

20

【 0 0 2 2 】

データ 4 0 2 は、図 1 のチェックサムジェネレータ 1 2 0 など、チェックサムジェネレータ 4 4 2 によって受信されてもよい。チェックサムジェネレータ 4 4 2 は、エラー 4 5 0、4 5 2 および 4 5 4 に関連づけられた場所 4 0 8、4 1 0 および 4 1 6 に基づいて、図 1 の第 1 のチェックサム 1 2 4 などのチェックサム 4 4 4 を生成してもよい。図に示したように、チェックサム 4 4 4 は 2 進値「1 1 0 0」に等しく、この値は 1 0 進値「1 2」に相当する。データ 4 0 4 は、第 2 のエラー訂正コード演算中に評価されてもよい。データ 4 0 4 は、ビット場所 4 2 4、4 2 6、4 2 8、4 3 0、4 3 2、4 3 4、4 3 6、4 3 8 および 4 4 0 を含む。エラー 4 5 6 が場所 4 2 4 に関連づけられ、エラー 4 5 8 が場所 4 3 0 に関連づけられる。

30

データ 4 0 4 は、図 1 のチェックサムジェネレータ 1 2 0 など、チェックサムジェネレータ 4 4 6 によって受信されてもよい。チェックサムジェネレータ 4 4 6 は、エラー 4 5 6、4 5 8 に関連づけられた場所 4 2 4、4 3 0 に基づいて、図 1 の第 1 のチェックサム 1 2 4 などのチェックサム 4 4 8 を生成してもよい。図に示したように、チェックサム 4 4 8 は 2 進値「0 1 0 0」に等しく、この値は 1 0 進値「4」に相当する。

【 0 0 2 3 】

4 7 0 において、チェックサム 4 4 4、4 4 8 は、エラーをランダムとして特定するために比較されてもよい。チェックサム 4 4 4 の 1 0 進値「1 2」とチェックサム 4 4 8 の 1 0 進値「4」との間の不一致は、無相関のランダムエラーを示しうる。エラー 4 5 0、4 5 2、4 5 4、4 5 6 は、第 1 および第 2 のエラー訂正コード演算の両方のデータ 4 0 2、4 0 4 内に異なる相対場所 4 0 8、4 1 6、4 2 6、4 3 4 を有するエラー 4 5 0、4 5 2、4 5 4、4 5 6 に基づいてランダムエラーであるように決定されてもよい。また、チェックサム 4 4 4 とチェックサム 4 4 8 との差が、エラー場所が異なることを特定するために、しきい値と比較されてもよい。

40

【 0 0 2 4 】

このように、図 4 は、エラー 4 5 0、4 5 2、4 5 4、4 5 6 および関連する場所 4 0 8、4 1 6、4 2 6、4 3 4 を決定するために用いられる第 1 および第 2 のエラー訂正コード演算を示す。エラー場所 4 0 8、4 1 6、4 2 6、4 3 4 に基づいて算出されたチェックサム 4 4 4、4 4 8 は、エラー 4 0 8、4 1 6、4 2 6、4 3 4 の場所が両方のエラー訂正コード演算において異なることを決定するために比較されてもよい。

50

【 0 0 2 5 】

図5は、第1および第2のエラー訂正コード演算の結果の例を示す。第1および第2のエラー訂正演算は、データ502の端部分付近のエラー550、552、554の場所が、データ504の端部分付近に位置するエラー558、560、562に同様に分散されることを示す。エラー訂正コード演算は、ワード線または図1のデータ110など、メモリから読み出された他のデータに異なる時間で実行されてもよい。データ502は、第1のエラー訂正コード演算中に評価されてもよい。例えば、データ502は、図1のエラー検出モジュール108によって生成されるエラー場所情報118であってもよく、図2のエラー検出モジュール220によって生成されたエラー場所情報222であってもよい。データ502は、メモリのストレージ要素に対応するメモリ要素の場所506、508、510、512、514、516、518、520および522を含む。エラー550が場所506に関連づけられ、エラー552が場所508に関連づけられる。第3のエラー554は場所512にあり、第4のエラー556は場所516にある。

10

【 0 0 2 6 】

データ502は、図1のチェックサムジェネレータ120などのチェックサムジェネレータ542によって受信されてもよい。チェックサムジェネレータ542は、エラー550、552、554および556に関連づけられた場所506、508、512および516に基づいて、図1の第1のチェックサム124などのチェックサム544を生成してもよい。図に示したように、チェックサム544は2進値「1110」を含み、この値は10進値「14」に相当する。

20

データ504は、第2のエラー訂正コード演算中に評価されてもよい。データ504は、ビット場所524、526、528、530、532、534、536、538および540を含む。エラー558が場所524に関連づけられ、エラー560が場所526に関連づけられる。エラー562が場所530に関連づけられ、エラー564が場所540に関連づけられる。

データ504は、図1のチェックサムジェネレータ120など、チェックサムジェネレータ546によって受信されてもよい。チェックサムジェネレータ546は、エラー558、560、562および564に関連づけられた場所524、526、530および540に基づいて、図1の第1のチェックサム124などのチェックサム548を生成してもよい。図に示したように、チェックサム548は2進値「1101」を含み、この値は10進値「13」に相当する。

30

【 0 0 2 7 】

570において、チェックサム544、548は、不良ワード線を特定するために比較されてもよい。両方のチェックサム544、548は、比較的大きな値を含む。ほとんどのエラーは、ワード線の同じ端部に存在する。データ502のエラー550、552、554は、両方の第1および第2のエラー訂正コード演算中に評価されるとき、データ504の同様に位置するエラー558、560、562に基づいて不良ワード線に関連づけられるように決定されてもよい。

【 0 0 2 8 】

このように、図5は、エラー550、552、554、556、558、560、562および564および関連する場所506、508、512、516、524、526、530および540を決定するために用いられた第1および第2のエラー訂正コード演算を示す。エラー場所506、508、512、516、524、526、530および540に基づいて算出されたチェックサム544、548は、エラー550、552、554、558、560および562の場所が、両方のエラー訂正コード演算においてデータ502、504の同様の部分に集中していることを決定するために比較されてもよい。エラー550、552、554、558、560および562の同じ相対場所506、508、512、524、526および530は、不良ワード線から得られるハードエラーの存在を示すものであってもよい。

40

【 0 0 2 9 】

50

当業者であれば、図３～図５が説明をしやすくするために簡略化されていることを認識するはずである。図３～図５は、いくつかあるパラメータの中でも特に、正確な値またはデータワードサイズを必ずしも示しているわけではない。結果的に、実施形態は、これらの例示的な実施例に限定されない。

【００３０】

図６は、エラーデータを追跡する方法６００のフローチャートを示す。図６の方法６００は、図１のコントローラ１０２または図２のコントローラ２０２など、メモリにアクセスするコンピューティングシステムによって実行されてもよい。６０２において、複数のエラーの各々にそれぞれ関連づけられた複数のエラーおよび複数の場所を特定するために、エラー訂正コード演算が実行されてもよい。例えば、図１のコントローラ１０２は、メモリアレイ１０４からデータ１１０を取得してもよい。データ１１０は、エラー検出モジュール１０８に出力されてもよい。場所にそれぞれ関連づけられたエラー１１４を特定するために、第１の時間に第１のエラー検出演算をエラー検出モジュール１０８に実行させるように制御信号が生成されてもよい。

10

【００３１】

６０４において、複数のエラーの各々に関連づけられた複数の場所に基づいて、チェックサムが生成されてもよい。例示するために、図１のチェックサムジェネレータ１２０は、エラー場所データ１１８を受信し、検出されたエラー１１４に関連づけられた場所に基づいて第１のチェックサム１２４を生成してもよい。

６０６において、メモリアレイから第２のチェックサムが取得されてもよい。例示するために、第２のチェックサムは、図２の先行チェックサム２３４であってもよい。第２のチェックサムは、第１のチェックサムとして同じメモリ領域からの先行データ読み出し演算に相当するものであってもよい。

20

第１のチェックサムおよび第２のチェックサムは、６０８で比較されてもよい。例えば、図１のコントローラ１０２は、第１のチェックサム１２４と第２のチェックサム１２６とを比較してもよい。別の実施例において、図２のチェックサム解析モジュール２２８は、チェックサム２２６と先行チェックサム２３４とを比較してもよい。

【００３２】

６１０において、第１のチェックサムと第２のチェックサムとの差が小さいかに関する決定がなされてもよい。例えば、図１のコントローラ１０２は、第１のチェックサム１２４と第２のチェックサム１２６との差が小さいかを決定してもよい。

30

第１のチェックサムと第２のチェックサムとの決定された差が小さい場合、６１０において、場所情報が評価されてもよい。例えば、図１のコントローラ１０２は、エラーの場所に基づいて、第１のチェックサム１２４と第２のチェックサム１２６との差を決定してもよい。図３に示すように、第１のエラーコード演算におけるエラー３５０の場所３０８は、第２のエラーコード演算におけるエラー３５４の場所３２６と同じであってもよい。あるいは、図４に示すように、第１のエラーコード演算におけるエラー４５０の場所４０８は、第２のエラーコード演算のエラー４５６の場所４２４とは異なるものであってもよい。第１のチェックサムと第２のチェックサムとの差が小さいかを決定することは、差をしきい値と比較することを含んでもよい。しきい値は、特定のチェックサム生成タイプの場所符号化特性に基づいて設定されてもよい。

40

【００３３】

場所情報は、６１２において、不良の可能性のあるワード線を特定するために評価されてもよい。例えば、エラーが、ワード線の端部付近に一貫して集中していることが決定されてもよい。例示するために図５を用いながら、第１のエラーコード演算におけるエラー５５０、５５２、５５４の場所５０６、５０８、５１２は、第２のエラーコード演算におけるエラー５５８、５６０、５６２に関連づけられた場所５２４、５２６、５３０に類似してもよい。ワード線の一端で繰り返し起こるエラーは、不良ワード線を示しうる。

６１４において、欠陥ストレージ要素を有するものとしてメモリ領域が特定されてもよい。例えば、図１のコントローラ１０２は、格納されたデータストリング内に欠陥セルを

50

有するものとしてメモリ領域 106 を特定してもよい。図 3 に示すように、特定のメモリ場所で繰り返し起こるエラーは、不良メモリセルを示しうる。

616 において、データ内の場所はモニタリングされてもよい。例えば、図 1 のコントローラ 102 は、後のエラー発生に備えて検出されたエラー 114 に対応する場所をモニタリングしてもよい。

【0034】

618 において、第 1 のチェックサムと第 2 のチェックサムとの差が大きいかに関する決定がなされてもよい。例えば、図 1 のコントローラ 102 は、第 1 のチェックサム 124 と第 2 のチェックサム 126 との差が大きいかを決定してもよい。差が大きいかを決定することは、差をしきい値と比較することを含んでもよい。しきい値は、特定のチェックサム生成タイプの場所符号化特性に基づいて設定されてもよい。

10

【0035】

618 において、第 1 のチェックサムと第 2 のチェックサムとの決定された差が大きい場合、620 において、複数のエラーのうちのあるエラーがランダムであるかを決定するために、場所情報が評価されてもよい。例えば、図 1 のコントローラ 102 は、検出されたエラー 114 が、第 1 のチェックサム 124 と第 2 のチェックサム 126 との差に基づいてランダムであることを決定してもよい。

622 において、エラーがランダムであるという決定に応答して、動作が開始されなくてもよい。624 において、第 1 のチェックサムは格納されてもよい。例えば、図 2 のコントローラ 102 は、メモリアレイ 104 のログファイル 122 内に第 1 のチェックサム 124 を格納してもよい。

20

【0036】

このように、図 6 は、エラーデータを追跡するために用いられる方法 600 を示す。第 1 のチェックサムおよび第 2 のチェックサムが生成され比較される。チェックサムは、ワード線または他のデータストリング内の検出されたエラーの場所に基づいて生成される。チェックサムの比較によって、チェックサム間の差が大きいかまたは小さいかを決定してもよい。小さな差は欠陥ストレージ要素を示しうるのに対して、大きな差はランダムエラーを示しうる。検出されたエラーがランダムであると決定されれば、動作を起こさないように決定がなされてもよい。

【0037】

30

図 7 は、エラーデータを追跡し格納する方法 700 のフローチャートを示す。図 7 の方法 700 は、図 1 のコントローラ 102 または図 2 のコントローラ 202 など、メモリにアクセスするコンピューティングシステムによって実行されてもよい。702 において、第 1 の複数の場所にそれぞれ関連づけられた第 1 の複数のエラーを特定するために、第 1 のエラー訂正コード演算が第 1 の時間で実行されてもよい。例えば、第 1 の複数の場所のうちのある場所が、不揮発性データストレージデバイス内のページまたはブロックのビット位置を含む。第 1 のエラー訂正コード演算は、不揮発性データストレージデバイス内においてコントローラで実行されてもよい。例えば、図 2 のコントローラ 202 は、メモリアレイ 204 からデータ 212 を取得してもよい。データ 212 は、エラー検出モジュール 220 に出力されてもよい。第 1 の複数の場所にそれぞれ関連づけられたエラー 214 を特定するために、第 1 の時間で第 1 のエラー検出演算をエラー検出モジュール 220 に実行させるように制御信号が生成されてもよい。

40

【0038】

704 において、第 2 の複数の場所にそれぞれ関連づけられた第 2 の複数のエラーを特定するために、第 2 のエラー訂正コード演算が第 2 の時間で実行されてもよい。例示するために、図 2 のコントローラ 202 は、メモリアレイ 204 からデータを再度取得してもよい。データは、エラー検出モジュール 220 に出力されてもよい。第 2 の複数の場所にそれぞれ関連づけられたエラーを特定するために、第 2 の時間で第 2 のエラー検出演算をエラー検出モジュール 220 に実行させるように制御信号が生成されてもよい。

【0039】

50

706において、不揮発性データストレージデバイス内のメモリアレイのログファイル内に、第1の複数の場所に相関された第1のデータおよび第2の複数の場所に相関された第2のデータが格納されてもよい。第1のデータは、第1の複数の場所の第1のチェックサムを含んでもよく、第2のデータは、第2の複数の場所の第2のチェックサムを含んでもよい。例えば、図2のコントローラ202は、メモリアレイ204のログファイル216内にチェックサム履歴データ218を格納してもよい。

【0040】

このように、図7は、第1のエラー訂正コード演算および第2のエラー訂正コード演算を実行する方法700を示す。各演算は、場所に関連づけられたエラーを特定してもよい。場所に相関されたデータは、訂正するためのエラーを効率的に特定する際に使用するためにメモリ内に格納されてもよい。

10

【0041】

図8は、エラーデータを追跡し評価する方法800のフローチャートを示す。図8の方法800は、図1のコントローラ102または図2のコントローラ202など、コンピューティングシステムによって実行されてもよい。802において、第1のエラー訂正コード演算の第1の複数のエラー場所に関連づけられた第1のチェックサムが受信されてもよい。第1のエラー訂正コード演算は、メモリアレイの領域内のエラーを訂正することに関連づけられてもよい。また、第2のエラー訂正コード演算の第2の複数のエラー場所に関連づけられた第2のチェックサムが受信されてもよい。例えば、図2のコントローラ202は、メモリアレイ204のログファイル216から第1のチェックサム226および先行チェックサム234を受信してもよい。

20

【0042】

804において、第1のチェックサムは、第2のチェックサムに比較されてもよい。例えば、図2のチェックサム解析モジュール228は、チェックサム226と先行チェックサム234とを比較してもよい。

806において、メモリアレイの領域内のエラーがランダムであるかの決定がなされてもよい。例示するために、図2のチェックサム解析モジュール228は、エラーがランダムであるかを決定するためにチェックサムデータ226を評価してもよい。

【0043】

808において、エラーがランダムであるという決定に応答して、動作が開始されなくてもよい。810において、メモリアレイの領域への動作が、比較の結果に基づいて開始されてもよい。第1のチェックサムと第2のチェックサムとの比較に基づいた複数の利用可能な動作の1つが実行されてもよい。例えば、図2の動作選択モジュール230は、データ212の移動、読み出しスクラブの開始、データのリフレッシュ、データ212のブロックを不良として指定、データ212のモニタリングまたはエラー214の訂正動作の開始を行うものであってもよい。

30

【0044】

このように、図8は、第1および第2のエラー訂正コード演算のエラー場所に関連づけられたチェックサムを受信することを含む方法800のフローチャートを示す。チェックサムは、エラーがランダムであるかを決定するために比較されてもよい。エラーがランダムでない場合、訂正動作が開始されてもよい。

40

【0045】

図9は、エラーデータを追跡しアドレス指定する方法900のフローチャートを示す。図9の方法900は、図1のコントローラ102または図2のコントローラ202など、コンピューティングシステムによって実行されてもよい。902において、メモリから第1のチェックサムが取得されてもよい。第1のチェックサムは、メモリ領域の第1のエラー場所セットに関連づけられてもよい。第1のエラー場所セットのうちのあるエラー場所が、不揮発性データストレージデバイス内のページまたはブロックのビット位置を含んでもよい。例えば、図1のコントローラ102は、メモリアレイ114のログファイル122から第1のチェックサム124を取得してもよい。第1のチェックサム124は、メモ

50

リ領域 106 のエラー場所に関連づけられてもよい。

【0046】

904 において、第1のチェックサムは、メモリ領域の第2のエラー場所セットに関連づけられた第2のチェックサムに比較されてもよい。例えば、図2のチェックサム解析モジュール228は、チェックサム226と先行チェックサム234とを比較してもよい。

【0047】

906 において、比較の結果に基づいてメモリ領域に対して動作が開始されてもよい。例えば、図2の動作選択モジュール230は、データの結果に基づいて動作を開始してもよい。第1のチェックサムと第2のチェックサムとの決定された差が小さいとき、第1の動作が開始されてもよい。例示するために、決定された差が小さいということは、エラー場所が同様のものであることを示しうる。第1のチェックサムと第2のチェックサムとの決定された差が大きいとき、第2の動作が開始されてもよい。一例において、決定された差が大きいということは、エラー場所の差が大きいことを示しうる。

10

【0048】

908 において、複数の利用可能な動作の1つが実行されてもよい。例えば、図2の動作選択モジュール230は、コントローラ202にデータ212を移動させ、データ212のブロックを不良であると指定し、データ212をモニタリングし、読み出しスクラブを実行し、データをリフレッシュし、またはエラー214の訂正を行うためのプロセスを実行する動作を開始してもよい。

【0049】

20

図9は、エラーを特定し訂正する方法900のフローチャートを示す。エラー場所に関連づけられたチェックサムを取得し比較する。エラーを訂正するための動作が、比較の結果に基づいて開始されてもよい。

【0050】

図10は、エラーデータを追跡しアドレス指定する方法1000のフローチャートを示す。図10の方法1000は、図1のコントローラ102または図2のコントローラ202など、コンピューティングシステムによって実行されてもよい。1002において、第1のエラー訂正コード演算の第1の複数のエラー場所に関連づけられた第1のチェックサムが受信されてもよい。例えば、図2のチェックサム解析モジュール228は、チェックサムジェネレータ224からチェックサム226を受信してもよい。

30

【0051】

また、1004において、第2のエラー訂正コード演算の第2の複数のエラー場所に関連づけられた第2のチェックサムが受信されてもよい。例えば、図2のコントローラ202は、メモリアレイ204のログファイル216から先行チェックサム234を取得してもよい。

1006 において、第1のチェックサムは、第2のチェックサムに比較されてもよい。例えば、図2のチェックサム解析モジュール228は、チェックサム226と先行チェックサム234とを比較してもよい。

1008 において、比較の結果に基づいてメモリ領域に対して動作が開始されてもよい。例えば、図2の動作選択モジュール230は、データの結果に基づいて動作を開始してもよい。

40

【0052】

1010 において、第3のエラー訂正コード演算の第3の複数のエラー場所に関連づけられた第3のチェックサムが受信されてもよい。また、第4のエラー訂正コード演算の第4の複数のエラー場所に関連づけられた第4のチェックサムが受信されてもよい。例えば、図2のチェックサム解析モジュール228は、メモリアレイ204のログファイル216のチェックサム履歴218から第3および第4のチェックサムを受信してもよい。

【0053】

1012 において、傾向を特定するために、第1のチェックサム、第2のチェックサム、第3のチェックサムおよび第4のチェックサムが使用されてもよい。例示するために、

50

図 2 のチェックサム解析モジュール 2 2 8 は、傾向を特定するためにチェックサム履歴 2 1 8 から取得されてもよい。例えば、チェックサム解析モジュール 2 4 0 は、多数のエラーが一貫して起こるか、または同じ相対場所に集中していることを決定してもよい。

1 0 1 4 において、傾向に基づいて動作が開始されてもよい。例えば、図 2 の動作選択モジュール 2 3 0 は、データを移動し、読み出しスクラブを開始し、データブロックを不良であると指定し、データをモニタリングし、またはエラーを訂正するための動作を開始してもよい。

【 0 0 5 4 】

図 1 0 は、一連のエラー訂正コード演算で生じたエラーが示す傾向を特定する方法 1 0 0 0 のフローチャートを示す。特定された傾向は、適切な訂正動作を開始するために使用されてもよい。

10

【 0 0 5 5 】

開示された実施形態の態様は、エラーの総数だけではなく、エラーの場所を評価することによってエラーの特定精度を上げうる。このようにして、プロセスは、エラー総数が一定のままの場合でも、場所を移動するエラーを特定しうる。

開示された実施形態の他の態様は、各読み出し演算で起こる E C C 訂正に基づいたデータを使用してエラーを追跡しうる。追跡されたエラーデータは、エラー場所および複数のエラーの両方を含みうる。追跡された複数のエラーは、消耗および欠陥をモニタリングするために使用されてもよく、エラー場所データは、障害検出に使用されてもよい。

開示された実施形態の他の態様は、1 つ以上のエラー場所に基づいて、複数のチェックサムを生成してもよい。例えば、メモリの各ページに対して、4 つのチェックサムが集められてもよい。チェックサムが決定され、追跡用ログファイル内に含められてもよい。

20

【 0 0 5 6 】

本願明細書に示されたさまざまなコンポーネントは、ブロックコンポーネントとして例示され、一般論として記載されているが、このようなコンポーネントは、1 つ以上のマイクロプロセッサ、状態マシン、または図 1 のデータエラー検出モジュール 1 0 8 およびチェックサムジェネレータ 1 2 0、および図 2 のエラー検出モジュール 2 2 0、チェックサムジェネレータ 2 2 4、チェックサム解析モジュール 2 2 8 および動作選択モジュール 2 3 0 が、このようなコンポーネントまたはそれらの任意の組み合わせによって生じる特定の機能を実行できるように構成された他の回路を含んでもよい。例えば、図 1 のエラー検出モジュール 1 0 8、チェックサムジェネレータ 1 2 0 またはそれらの両方は、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 のデータストレージデバイス 1 0 0 にエラーデータを追跡させる他の構造などの物理コンポーネントであってもよい。

30

【 0 0 5 7 】

データストレージデバイス 1 0 0 は、1 つ以上の外部デバイスに選択的に連結されるように構成されたポータブルデバイスであってもよい。しかし、他の実施形態において、データストレージデバイス 1 0 0 は、ポータブル通信デバイスのハウジング内など、1 つ以上のホストデバイス内に取り付けられるかまたは埋め込まれてもよい。例えば、データストレージデバイス 1 0 0 は、無線電話、携帯用情報端末 (P D A)、ゲーム機またはゲームコンソール、ポータブル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装置内であってもよい。特定の実施形態において、データストレージデバイス 1 0 0 は、フラッシュメモリ (例えば、N A N D、N O R、マルチレベルセル (M L C)、分割ビット線 N O R (D I N O R : Divided bit-line NOR)、A N D、高容量結合率 (H i C R)、非対称コンタクトレストランジスタ (A C T : asymmetrical contactless transistor) または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 (E P R O M)、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 (E E P R O M)、読み出し専用メモリ (R O M)、ワンタイムプログラマブルメモリ (O T P) または任意の他のタイプのメモリなどの不揮発性メモリを含む。

40

【 0 0 5 8 】

本願明細書に記載した実施形態の説明は、種々の実施形態を一般的に理解するために与

50

えることを意図したものである。本開示から他の実施形態が利用され導き出されてもよく、本開示の範囲から逸脱することなく、構造的および論理的な代替および変更がなされてもよい。本開示は、種々の実施形態の任意およびすべての後続の適応または変更に及ぶように意図される。以上のことから、本開示および図面は、制限的なものではなく例示的なものとして見なされるべきものである。

【0059】

前に開示された主題は、制限的ではなく例示的なものとして見なされるべきであり、添付の特許請求の範囲は、本開示の範囲内にあるこのようなすべての修正例、改良例および他の実施形態に及ぶように意図される。このようにして、法によって認められる最大限の範囲において、本発明の範囲は、以下の特許請求の範囲およびその均等物を最大限に幅広く解釈することによって決定されるべきであり、前述した詳細な説明によって制約または制限されるものではない。

10

【図1】

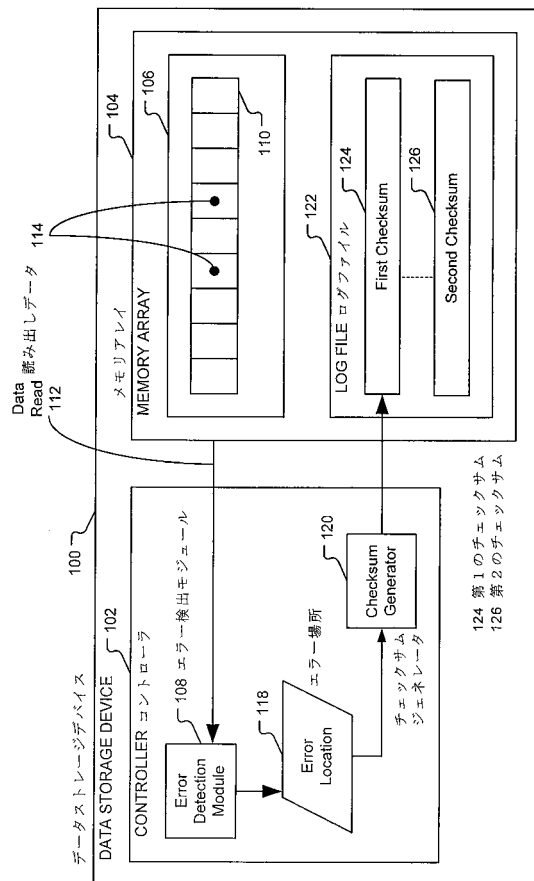


FIG. 1

【図2】

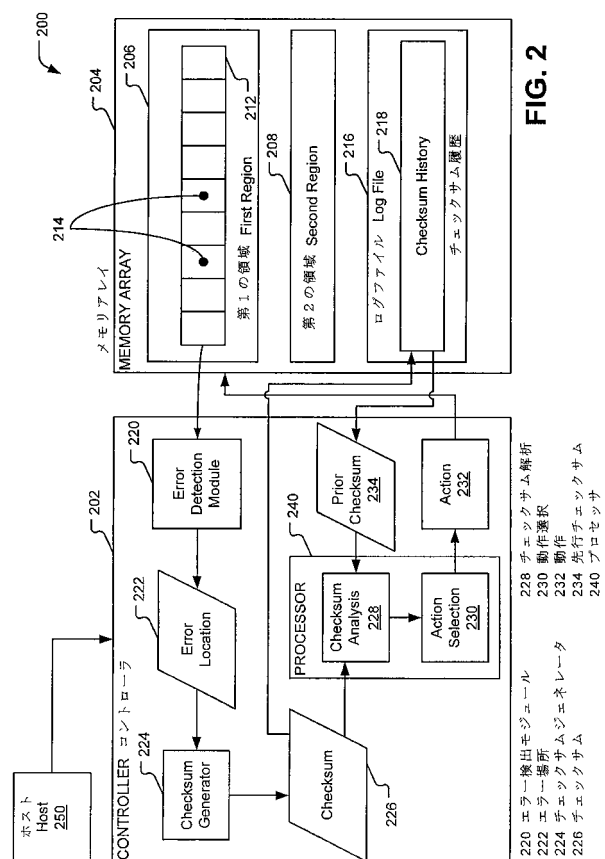


FIG. 2

【図 3】

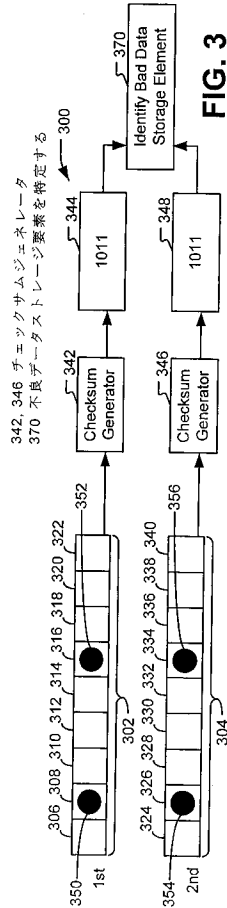


FIG. 3

【図 5】

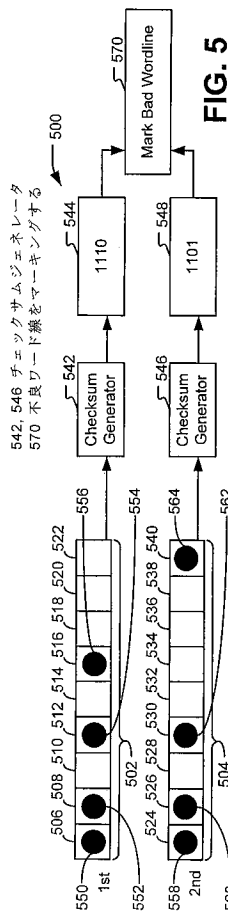


FIG. 5

【図 4】

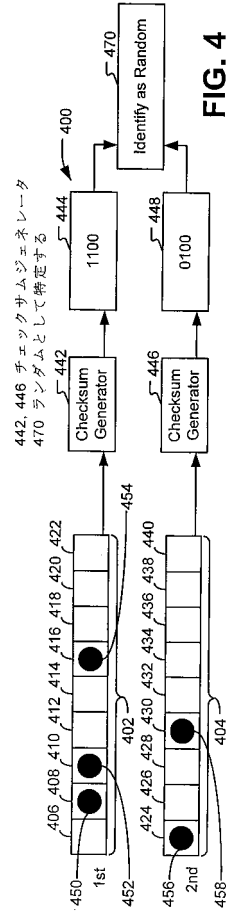


FIG. 4

【図 6】

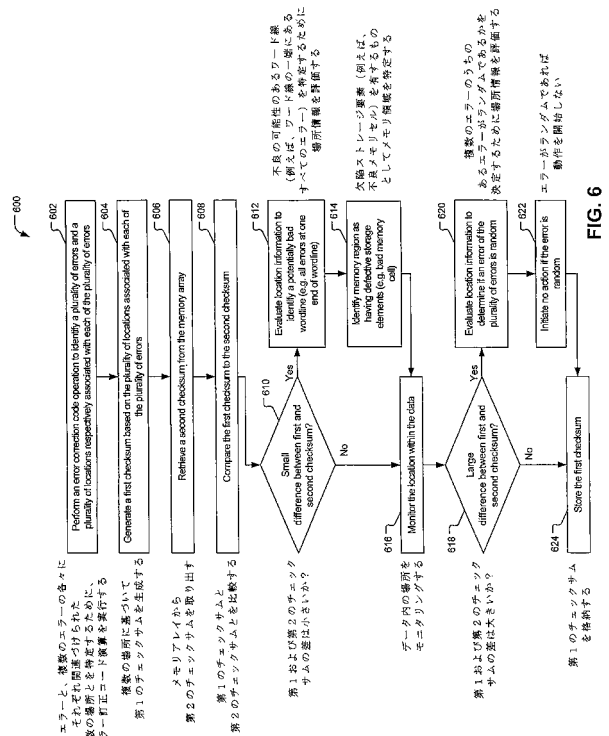


FIG. 6

【図 7】

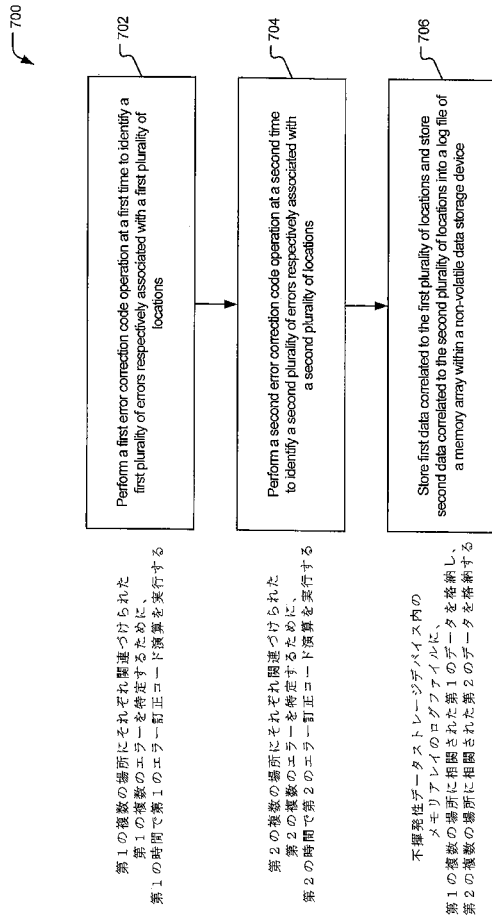


FIG. 7

【図 8】

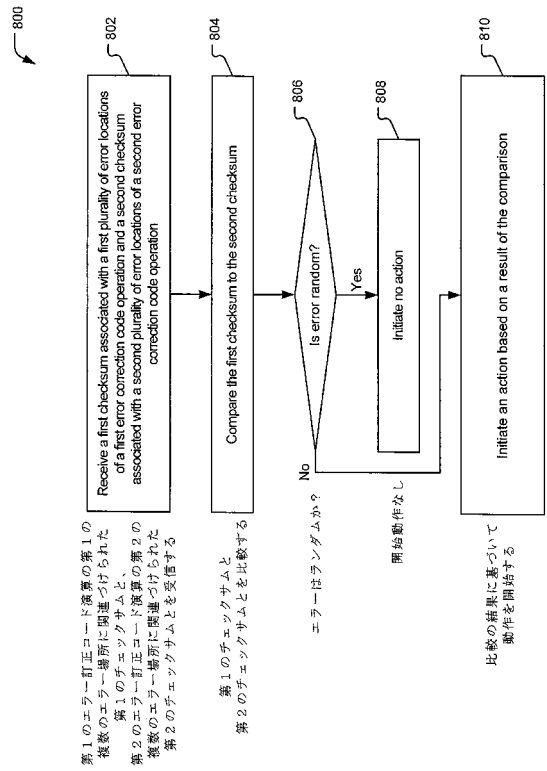


FIG. 8

【図 9】

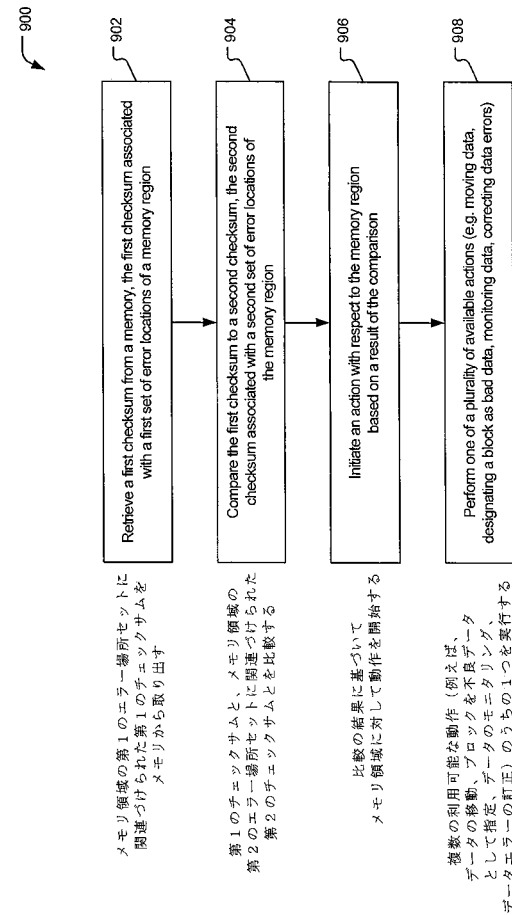


FIG. 9

【図 10】

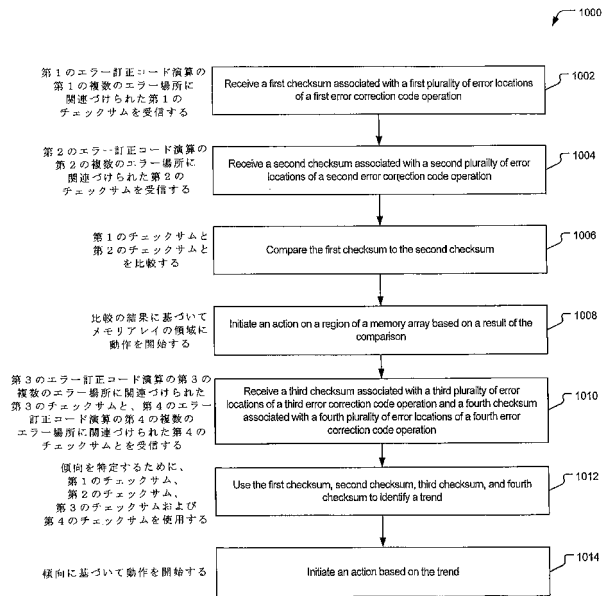


FIG. 10

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

審査官 野田 佳邦

(56)参考文献 特開2005-196680(JP, A)

特開2008-027296(JP, A)

特表2005-527062(JP, A)

特表2002-509331(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/16