

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-146129

(P2012-146129A)

(43) 公開日 平成24年8月2日(2012.8.2)

(51) Int. Cl. F I テーマコード (参考)
G06F 12/16 (2006.01) G06F 12/16 310A 5B018
 G06F 12/16 320M

審査請求 未請求 請求項の数 14 O L (全 15 頁)

(21) 出願番号 特願2011-3988 (P2011-3988)
 (22) 出願日 平成23年1月12日 (2011.1.12)

(71) 出願人 591128453
 株式会社メガチップス
 大阪府大阪市淀川区宮原4丁目1番6号
 (74) 代理人 100104444
 弁理士 上羽 秀敏
 (74) 代理人 100112715
 弁理士 松山 隆夫
 (74) 代理人 100125704
 弁理士 坂根 剛
 (74) 代理人 100120662
 弁理士 川上 桂子
 (72) 発明者 玉川 裕子
 大阪市淀川区宮原4丁目1番6号 株式会社
 メガチップス内

最終頁に続く

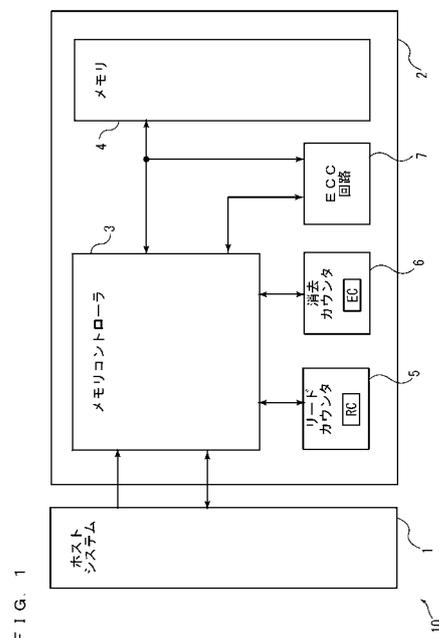
(54) 【発明の名称】 メモリアクセス制御装置

(57) 【要約】

【課題】再書き込み可能な不揮発性半導体メモリに対する負荷を抑えつつ、リードディスタ urb 現象により生じるエラーに対処する技術を提供することを課題とする。

【解決手段】メモリ4は、ページ単位での読み出しと、ブロック単位で消去が可能である。リードカウンタ5は、メモリ4に対するリード回数RCをブロック単位でカウントする。メモリコントローラ3は、リード回数RCがリード回数閾値RCthを上回る検査対象ブロックが存在するか否かを判定する。メモリコントローラ3は、検査対象ブロックが検出された場合、検査対象ブロックに含まれる各ページに対してエラー検出処理を行い、エラー閾値EBthを超えるエラーが検出されたエラーページが存在するか否かを検査する。エラーページが検出された場合、検査対象ブロックに対してデータの再書き込み処理を実行する。再書き込み処理終了後、検査対象ブロックのリード回数RCを初期化する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

再書き込み可能な不揮発性半導体メモリに対するアクセスを制御するメモリアクセス制御装置であって、

前記不揮発性半導体メモリは、ページ単位での読み出しが可能であり、かつ、複数のページを含むブロック単位で消去可能であり、

前記不揮発性半導体メモリに対するリード回数をブロック単位でカウントするリードカウンタと、

前記リードカウンタを参照し、リード回数がリード回数閾値を上回る検査対象ブロックが存在するか否かを判定する検査対象判定部と、

前記検査対象ブロックが検出された場合、前記検査対象ブロックに含まれる各ページに対してエラー検出処理を行い、エラー閾値を超えるエラーが検出されたエラーページが存在するか否かを検査する検査部と、

前記検査部によって前記エラーページが検出された場合、前記検査対象ブロックに対してデータの再書き込み処理を実行する再書き込み制御部と、

前記再書き込み処理終了後、前記検査対象ブロックのリード回数を初期化する初期化部と、

を備えるメモリアクセス制御装置。

10

【請求項 2】

請求項 1 に記載のメモリアクセス制御装置であって、

前記検査部は、前記検査対象ブロックの各ページをエラー検出回路の検出処理単位で区分し、各区分のエラー検出値の最大値をページ最大エラー値として設定し、前記ページ最大エラー値が前記エラー閾値を超える場合、当該ページを前記エラーページとして判定するメモリアクセス制御装置。

20

【請求項 3】

請求項 2 に記載のメモリアクセス制御装置であって、

前記検査部は、

前記検査対象ブロック内に前記エラーページが検出されない場合、前記検査対象ブロックに含まれる各ページの前記ページ最大エラー値を比較し、前記ページ最大エラー値の最大値をブロック最大エラー値として設定する最大値設定部、

を含み、

前記初期化部は、

前記ブロック最大エラー値の設定処理の終了後、前記ブロック最大エラー値に基づいて第 1 補正值を算出し、前記第 1 補正值に基づいて、前記検査対象ブロックのリード回数の初期化値を補正する第 1 補正部、

を含むメモリアクセス制御装置。

30

【請求項 4】

請求項 3 に記載のメモリアクセス制御装置であって、

前記第 1 補正部は、前記エラー閾値に対する前記ブロック最大エラー値の割合をエラー進行割合とすると、前記リード回数閾値に前記エラー進行割合を乗算した値を当該ブロックのリード回数の初期化値に加算するメモリアクセス制御装置。

40

【請求項 5】

請求項 3 または請求項 4 に記載のメモリアクセス制御装置であって、

前記初期化部は、

前記検査対象ブロックに含まれるページ数を当該ブロックのリード回数の初期化値に加算する第 2 補正部、

を含むメモリアクセス制御装置。

【請求項 6】

請求項 1 ないし請求項 5 のいずれかに記載のメモリアクセス制御装置であって、さらに

50

前記不揮発性半導体メモリに対する消去回数をブロック単位でカウントする消去カウンタと、

前記再書き込み処理終了後、前記消去カウンタに記録されている前記検査対象ブロックの消去回数を1だけインクリメントする消去回数加算部と、
を備え、

前記初期化部は、

前記再書き込み処理終了後、前記検査対象ブロックの消去回数に基づいて第3補正値を算出し、前記第3補正値に基づいて、前記検査対象ブロックのリード回数の前記初期化値を補正する第3補正部、
を含むメモリアクセス制御装置。

10

【請求項7】

請求項6に記載のメモリアクセス制御装置であって、

前記第3補正部は、前記検査対象ブロックの消去回数を所定ビットだけ左シフトした値を、当該ブロックのリード回数の初期化値に加算とするメモリアクセス制御装置。

【請求項8】

再書き込み可能な不揮発性半導体メモリに対するアクセスを制御するメモリアクセス制御装置であって、

前記不揮発性半導体メモリは、ページ単位での読み出しが可能であり、かつ、複数のページを含むブロック単位で消去可能であり、

前記不揮発性半導体メモリに対するリード回数を複数のブロックを含むバンク単位でカウントするリードカウンタと、

20

前記リードカウンタを参照し、リード回数がリード回数閾値を上回る検査対象バンクが存在するか否かを判定する検査対象判定部と、

前記検査対象バンクが検出された場合、前記検査対象バンクに含まれる各ページに対してエラー検出処理を行い、エラー閾値を超えるエラーが検出されたエラーページが存在するか否かを検査する検査部と、

前記検査部によって前記エラーページが検出された場合、前記検査対象バンクに対してデータの再書き込み処理を実行する再書き込み制御部と、

前記再書き込み処理終了後、前記検査対象バンクの読み出し回数を初期化する初期化部と、

30

を備えるメモリアクセス制御装置。

【請求項9】

請求項8に記載のメモリアクセス制御装置であって、

前記検査部は、前記検査対象バンクの各ページをエラー検出回路の検出処理単位で区分し、各区分のエラー検出値の最大値をページ最大エラー値として設定し、前記ページ最大エラー値が前記エラー閾値を超える場合、当該ページを前記エラーページとして判定するメモリアクセス制御装置。

【請求項10】

請求項9に記載のメモリアクセス制御装置であって、

前記検査部は、

40

前記検査対象バンク内に前記エラーページが検出されない場合、前記検査対象バンクに含まれる各ページの前記ページ最大エラー値を比較し、前記ページ最大エラー値の最大値をバンク最大エラー値として設定する最大値設定部、

を含み、

前記初期化部は、

前記バンク最大エラー値の設定処理の終了後、前記バンク最大エラー値に基づいて第1補正値を算出し、前記第1補正値に基づいて、前記検査対象バンクのリード回数の初期化値を補正する第1補正部、

を含むメモリアクセス制御装置。

【請求項11】

50

請求項 10 に記載のメモリアクセス制御装置であって、
前記第 1 補正部は、前記エラー閾値に対する前記バンク最大エラー値の割合をエラー進行割合とすると、前記リード回数閾値に前記エラー進行割合を乗算した値を当該バンクのリード回数の初期化値に加算するメモリアクセス制御装置。

【請求項 12】

請求項 10 または請求項 11 に記載のメモリアクセス制御装置であって、
前記初期化部は、
前記検査対象バンクに含まれるページ数を、当該バンクのリード回数の初期化値に加算する第 2 補正部、
を含むメモリアクセス制御装置。

10

【請求項 13】

請求項 8 ないし請求項 12 のいずれかに記載のメモリアクセス制御装置であって、さらに、
前記不揮発性半導体メモリに対する消去回数をバンク単位でカウントする消去カウンタと、

前記再書き込み処理終了後、前記消去カウンタに記録されている前記検査対象バンクの消去回数を 1 だけインクリメントする消去回数加算部と、
を備え、

前記初期化部は、

前記再書き込み処理終了後、前記検査対象バンクの消去回数に基づいて第 3 補正値を算出し、前記第 3 補正値に基づいて、前記検査対象バンクのリード回数の初期化値を補正する第 3 補正部、
を含むメモリアクセス制御装置。

20

【請求項 14】

請求項 13 に記載のメモリアクセス制御装置であって、
前記第 3 補正部は、前記検査対象バンクの消去回数を所定ビットだけ左シフトした値を、当該バンクのリード回数の初期化値に加算とするメモリアクセス制御装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、フラッシュメモリからデータが繰り返し読み出されることにより、データが意図せず書き換えられるリードディスタ urb 現象に対処する技術に関する。

30

【背景技術】

【0002】

NAND フラッシュメモリは、再書き込み可能な不揮発性半導体メモリである。NAND フラッシュメモリは、高集積化を実現しながら、低コスト化が図られているため、SD メモリカードなどの形態で広く普及している。

【0003】

NAND フラッシュメモリにおいては、特定のセルが繰り返し読み出されることにより、記憶データが意図せず書き換えられる現象が生じることが分かっている。この現象は、リードディスタ urb (Read Disturb) 現象と呼ばれている現象であり、NAND フラッシュメモリを利用する上での課題となっている。

40

【0004】

特定のセルに対する読み出し処理が繰り返し行われた場合であっても、一旦、そのセルを含む領域に対してデータの更新が行われれば、セルの状態が修復されるため、リードディスタ urb の発生が情報処理に影響を与えることを回避できる。しかし、特定のセルに対して書き込み処理が行われず、読み出し処理だけが繰り返し行われる場合、リードディスタ urb の発生する可能性が高くなり、情報処理に影響を与える可能性もある。

【0005】

最近では、NAND フラッシュメモリは、情報処理装置のプログラムメモリとして R O

50

Mのような役割で利用されるケースが多くなっている。NANDフラッシュメモリがそのように利用された場合、連続的な読み出し処理のみが行われるため、リードディスタープの発生可能性が高くなる。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2008-192266号公報

【特許文献2】特開2008-192267号公報

【発明の概要】

【発明が解決しようとする課題】

10

【0007】

リードディスタープの発生を回避あるいはリードディスタープにより発生したエラーを修復するために、いくつかの方策がとられている。たとえば、所定のアルゴリズムでNANDフラッシュメモリの各ページのデータを読み込み、エラー検出訂正回路を利用してエラー検出を行う方法がある。エラー検出訂正回路によって、エラーが発生しているデータが検出された場合、データを訂正した上で、メモリにデータを再書き込みする。これにより、リードディスタープの発生していたセルが修復される。

【0008】

エラーを検出するアルゴリズムとしては、様々なアルゴリズムが考えられる。しかし、いずれのアルゴリズムを用いた場合にも、エラー検出のための読み出し処理自体が、NANDフラッシュメモリに負荷を掛けることになる。NANDフラッシュメモリの全ページを順番に検査する方法を採用した場合、エラー検出のための読み出し処理自体が、リードディスタープの発生要因となることも考えられる。

20

【0009】

上記特許文献1においては、ページごとに読み出し回数をカウントし、読み出し回数の多い領域に対してリフレッシュ処理を行うようにしている。この方法により、全ての記憶領域に対してエラー検出用の読み出し処理を行う必要がなくなる。しかし、読み出し回数が多い領域に対しては、実際に、リードディスタープが発生していない場合にも、再書き込み処理が行われるため、必要以上にNANDフラッシュメモリに負荷を掛ける可能性がある。

30

【0010】

上記特許文献2においても、読み出し回数が閾値を超えた場合には、データの書き直し処理を行うため、NANDフラッシュメモリに負荷を掛ける可能性がある。

【0011】

そこで、本発明は前記問題点に鑑み、再書き込み可能な不揮発性半導体メモリに対する負荷を抑えつつ、リードディスタープ現象により生じるエラーに対処する技術を提供することを目的とする。

【課題を解決するための手段】

【0012】

上記課題を解決するため、請求項1記載の発明は、再書き込み可能な不揮発性半導体メモリに対するアクセスを制御するメモリアクセス制御装置であって、前記不揮発性半導体メモリは、ページ単位での読み出しが可能であり、かつ、複数のページを含むブロック単位で消去可能であり、前記不揮発性半導体メモリに対するリード回数をブロック単位でカウントするリードカウンタと、前記リードカウンタを参照し、リード回数がリード回数閾値を上回る検査対象ブロックが存在するか否かを判定する検査対象判定部と、前記検査対象ブロックが検出された場合、前記検査対象ブロックに含まれる各ページに対してエラー検出処理を行い、エラー閾値を超えるエラーが検出されたエラーページが存在するか否かを検査する検査部と、前記検査部によって前記エラーページが検出された場合、前記検査対象ブロックに対してデータの再書き込み処理を実行する再書き込み制御部と、前記再書き込み処理終了後、前記検査対象ブロックのリード回数を初期化する初期化部と、を備え

40

50

る。

【0013】

請求項2記載の発明は、請求項1に記載のメモリアクセス制御装置であって、前記検査部は、前記検査対象ブロックの各ページをエラー検出回路の検出処理単位で区分し、各区分のエラー検出値の最大値をページ最大エラー値として設定し、前記ページ最大エラー値が前記エラー閾値を超える場合、当該ページを前記エラーページとして判定する。

【0014】

請求項3記載の発明は、請求項2に記載のメモリアクセス制御装置であって、前記検査部は、前記検査対象ブロック内に前記エラーページが検出されない場合、前記検査対象ブロックに含まれる各ページの前記ページ最大エラー値を比較し、前記ページ最大エラー値の最大値をブロック最大エラー値として設定する最大値設定部、を含み、前記初期化部は、前記ブロック最大エラー値の設定処理の終了後、前記ブロック最大エラー値に基づいて第1補正値を算出し、前記第1補正値に基づいて、前記検査対象ブロックのリード回数の初期化値を補正する第1補正部、を含む。

10

【0015】

請求項4記載の発明は、請求項3に記載のメモリアクセス制御装置であって、前記第1補正部は、前記エラー閾値に対する前記ブロック最大エラー値の割合をエラー進行割合とすると、前記リード回数閾値に前記エラー進行割合を乗算した値を当該ブロックのリード回数の初期化値に加算する。

【0016】

請求項5記載の発明は、請求項3または請求項4に記載のメモリアクセス制御装置であって、前記初期化部は、前記検査対象ブロックに含まれるページ数を当該ブロックのリード回数の初期化値に加算する第2補正部、を含む。

20

【0017】

請求項6記載の発明は、請求項1ないし請求項5のいずれかに記載のメモリアクセス制御装置であって、さらに、前記不揮発性半導体メモリに対する消去回数をブロック単位でカウントする消去カウンタと、前記再書き込み処理終了後、前記消去カウンタに記録されている前記検査対象ブロックの消去回数を1だけインクリメントする消去回数加算部と、を備え、前記初期化部は、前記再書き込み処理終了後、前記検査対象ブロックの消去回数に基づいて第3補正値を算出し、前記第3補正値に基づいて、前記検査対象ブロックのリード回数の前記初期化値を補正する第3補正部、を含む。

30

【0018】

請求項7記載の発明は、請求項6に記載のメモリアクセス制御装置であって、前記第3補正部は、前記検査対象ブロックの消去回数を所定ビットだけ左シフトした値を、当該ブロックのリード回数の初期化値に加算とする。

【0019】

請求項8記載の発明は、再書き込み可能な不揮発性半導体メモリに対するアクセスを制御するメモリアクセス制御装置であって、前記不揮発性半導体メモリは、ページ単位での読み出しが可能であり、かつ、複数のページを含むブロック単位で消去可能であり、前記不揮発性半導体メモリに対するリード回数を複数のブロックを含むバンク単位でカウントするリードカウンタと、前記リードカウンタを参照し、リード回数がリード回数閾値を上回る検査対象バンクが存在するか否かを判定する検査対象判定部と、前記検査対象バンクが検出された場合、前記検査対象バンクに含まれる各ページに対してエラー検出処理を行い、エラー閾値を超えるエラーが検出されたエラーページが存在するか否かを検査する検査部と、前記検査部によって前記エラーページが検出された場合、前記検査対象バンクに対してデータの再書き込み処理を実行する再書き込み制御部と、前記再書き込み処理終了後、前記検査対象バンクの読み出し回数を初期化する初期化部と、を備える。

40

【0020】

請求項9記載の発明は、請求項8に記載のメモリアクセス制御装置であって、前記検査部は、前記検査対象バンクの各ページをエラー検出回路の検出処理単位で区分し、各区分

50

のエラー検出値の最大値をページ最大エラー値として設定し、前記ページ最大エラー値が前記エラー閾値を超える場合、当該ページを前記エラーページとして判定する。

【0021】

請求項10記載の発明は、請求項9に記載のメモリアクセス制御装置であって、前記検査部は、前記検査対象バンク内に前記エラーページが検出されない場合、前記検査対象バンクに含まれる各ページの前記ページ最大エラー値を比較し、前記ページ最大エラー値の最大値をバンク最大エラー値として設定する最大値設定部、を含み、前記初期化部は、前記バンク最大エラー値の設定処理の終了後、前記バンク最大エラー値に基づいて第1補正値を算出し、前記第1補正値に基づいて、前記検査対象バンクのリード回数の初期化値を補正する第1補正部、を含む。

10

【0022】

請求項11記載の発明は、請求項10に記載のメモリアクセス制御装置であって、前記第1補正部は、前記エラー閾値に対する前記バンク最大エラー値の割合をエラー進行割合とすると、前記リード回数閾値に前記エラー進行割合を乗算した値を当該バンクのリード回数の初期化値に加算する。

【0023】

請求項12記載の発明は、請求項10または請求項11に記載のメモリアクセス制御装置であって、前記初期化部は、前記検査対象バンクに含まれるページ数を、当該バンクのリード回数の初期化値に加算する第2補正部、を含む。

【0024】

20

請求項13記載の発明は、請求項8ないし請求項12のいずれかに記載のメモリアクセス制御装置であって、さらに、前記不揮発性半導体メモリに対する消去回数をバンク単位でカウントする消去カウンタと、前記再書き込み処理終了後、前記消去カウンタに記録されている前記検査対象バンクの消去回数を1だけインクリメントする消去回数加算部と、を備え、前記初期化部は、前記再書き込み処理終了後、前記検査対象バンクの消去回数に基づいて第3補正値を算出し、前記第3補正値に基づいて、前記検査対象バンクのリード回数の初期化値を補正する第3補正部、を含む。

【0025】

請求項14記載の発明は、請求項13に記載のメモリアクセス制御装置であって、前記第3補正部は、前記検査対象バンクの消去回数を所定ビットだけ左シフトした値を、当該バンクのリード回数の初期化値に加算とする。

30

【発明の効果】

【0026】

本発明のメモリアクセス制御装置によれば、再書き込み可能な半導体メモリに対する負荷を低減させながら、リードディスタープの発生を抑制し、あるいは、リードディスタープの発生が情報処理に影響を与えることを未然に防ぐことができる。

【図面の簡単な説明】

【0027】

【図1】本実施の形態に係る情報処理システムのブロック図である。

【図2】NANDフラッシュメモリのメモリ構成図である。

40

【図3】NANDフラッシュメモリとECCのエラー検出単位との関係を示す図である。

【図4】本実施の形態に係る制御処理のフローチャートである。

【図5】本実施の形態に係る制御処理のフローチャートである。

【発明を実施するための形態】

【0028】

{第1の実施の形態}

以下、図面を参照しつつ本発明の実施の形態について説明する。図1は、本実施の形態に係る情報処理システム10の機能ブロック図である。この情報処理システム10は、ホストシステム1およびメモリモジュール2を備えて構成される。メモリモジュール2は、メモリコントローラ3、メモリ4、リードカウンタ5、消去カウンタ6およびエラー検出

50

訂正回路（ECC回路：Error Checking and Correction 回路）7を備えている。

【0029】

メモリ4は、再書き込み可能な不揮発性半導体メモリである。本実施の形態において、メモリ4は、NANDフラッシュメモリである。メモリコントローラ3は、メモリ4に対するデータの読み出しや書き込みを制御する。例えば、この情報処理システムは、携帯電話装置として機能する。この場合、ホストシステム1は、携帯電話装置本体を構成し、メモリモジュール2がメモリカードを構成する。メモリモジュール2は、ホストシステム1に対して着脱自在なタイプであってもよいし、ホストシステム1を構成する装置に内蔵されるタイプであってもよい。

10

【0030】

なお、本実施の形態においては、メモリコントローラ3が、メモリモジュール2に搭載される構成となっているが、ホストシステム1がメモリコントローラ3を備える構成であってもよい。また、エラー検出回路7は、メモリ4内に設けられてもよいし、メモリコントローラ3内に設けられてもよい。

【0031】

図2は、メモリ4の構成を示す図である。メモリ4は、1枚のダイ（Die）で構成されている。ダイは、複数のバンク（Bank）を含んでいる。図2の例では、ダイは、J個のバンクを含んでいる。バンクは、複数のブロック（Block）を含んでいる。図2の例では、バンクは、M個のブロックを含んでいる。さらに、ブロックは、複数のページ（Page）を含んでいる。図2の例では、ブロックは、Z個のページを含んでいる。

20

【0032】

ページは、メモリ4の読み出し単位である。メモリコントローラ3は、メモリ4に格納されているデータをページ単位で読み出すことが可能である。ブロックは、メモリ4の消去単位である。メモリコントローラ3は、メモリ4に格納されているデータをブロック単位で消去することが可能である。

【0033】

メモリ4のブロックは、物理的に隣り合って配置された複数のページで構成されている。たとえば、ブロックは、メモリアドレスの順番で隣り合って配列された複数のページで構成される。したがって、あるブロック内の特定のページに対して繰り返し読み出し処理が行われた場合、当該ブロック内の周辺のページに対して負荷が掛かり、周辺のページについてリードディスタurbが発生する可能性がある。メモリ4のバンクは、物理的に隣り合って配置された複数のブロックで構成されている。たとえば、バンクは、ブロック番号順で隣り合って配列された複数のブロックで構成される。

30

【0034】

再び、図1を参照する。リードカウンタ5は、メモリ4に対する読み出しアクセスの回数をカウントする。上述したように、メモリコントローラ3は、メモリ4に対してページ単位での読み出し処理を実行する。メモリコントローラ3は、メモリ4に対して読み出し処理を実行したとき、リードカウンタ5に対して、読み出し処理を実行したページを特定する情報を通知する。具体的には、メモリコントローラ3は、リードカウンタ5に対して、読み出し処理を実行したページのバンク番号、ブロック番号およびページアドレスを通知する。

40

【0035】

リードカウンタ5は、メモリコントローラ3から、読み出し対象のページのバンク番号、ブロック番号およびページアドレスを取得するが、ページ単位で読み出しアクセスの回数をカウントするのではなく、ブロック単位で読み出しアクセスの回数をカウントする。つまり、リードカウンタ5は、ブロック内に含まれるいずれのページに読み出しアクセスが行われたかを区別することなく、各ブロックに対する読み出しアクセスをカウントする。リードカウンタ5は、ブロック数に対応した複数のリード回数RCを保持している。

【0036】

50

消去カウンタ6は、メモリ4に対する消去回数をカウントする。上述したように、メモリコントローラ3は、メモリ4に対してブロック単位での消去処理を実行する。メモリコントローラ3は、メモリ4に対して消去処理を実行したとき、消去カウンタ6に対して、消去処理を実行したブロックを特定する情報を通知する。具体的には、メモリコントローラ3は、消去カウンタ6に対して、消去処理を実行したブロックのバンク番号およびブロック番号を通知する。消去カウンタ6は、ブロック単位で消去処理の回数をカウントする。消去カウンタ6は、ブロック数に対応した複数の消去回数ECを保持している。

【0037】

メモリコントローラ3は、情報処理システム10の電源がOFFされるとき、リードカウンタ5あるいは消去カウンタ6に格納されているカウント値を、メモリ4に書き込む。メモリコントローラ3は、情報処理システム10の電源がONされたとき、メモリ4から、カウント値を読み込んでリードカウンタ5あるいは消去カウンタ6にセットする。これにより、情報処理システム10の電源がOFFされる場合にも、カウント値を保持することができる。また、メモリコントローラ3は、電源がONされている状態においても、所定の周期で、リードカウンタ5あるいは消去カウンタ6に記録されているカウンタ値をメモリ4に書き戻すようにしてもよい。

10

【0038】

エラー検出訂正回路7(以下、ECC回路7と呼ぶ。)は、メモリ4に誤ったデータが記録されていることを検出するとともに、誤ったデータを訂正する回路である。本実施の形態のECC回路7は、図3に示すように、1024Byte(1KByte)単位で、エラー検出および訂正を行う。たとえば、1024Byteに対して、ECC回路7は、10bitのエラーを検出および訂正可能である。

20

【0039】

図3の例では、4KByteのNANDフラッシュメモリの記憶領域は、ECC回路7によって、4分割されてエラー検出されている。つまり、ECC回路7のエラー検出単位が1024Byteである。図では、それぞれ1024Byteのエラー検出単位に対して、5bit、7bit、3bit、0bitのエラーが検出されたことを示している。ECC回路7が10bitのエラー訂正能力を持っている場合には、図3の例では、全ての記憶領域についてエラーを訂正することが可能である。

【0040】

30

次に、本実施の形態に係る制御内容について説明する。図4および図5は、メモリコントローラ3、リードカウンタ5、消去カウンタ6あるいはECC回路7において実行される制御内容を示すフローチャートである。メモリコントローラ3は、所定のトリガによって図4のフローを開始する。トリガとしては、たとえば、メモリ4に対する読み出しアクセスが行われた直後のタイミングが例として挙げられる。あるいは、メモリ4に対するアクセスが行われていない場合であって、タイマーで計測した所定のタイミングが到来した場合が例として挙げられる。あるいは、情報処理システム10が起動しているときには、常時、図4および図5の制御を実行するようにしてもよい。

【0041】

メモリコントローラ4は、まず、リードカウンタ5にアクセスし、リード回数RCを取得する。メモリコントローラ4は、リード回数RCにオフセット値RCoffsetを加算した値がリード回数閾値RCthを超えるブロックが存在する否かをチェックする(ステップS1)。上述したように、リードカウンタ5には、ブロック数に対応したリード回数RCが保持されている。メモリコントローラ4は、全てのブロックについてステップS1の判定を行う。

40

【0042】

リード回数閾値RCthは、たとえば、10,000などの数字が設定される。オフセット値RCoffsetは、過去のエラー発生状況に応じて設定されるオフセット値である。ステップS1において、純粹なリード回数RCだけで評価するのではなく、オフセット値RCoffsetを加味した値を評価することで、過去にエラーが発生したブロック

50

に対しては、厳しくチェックするようにしている。オフセット値 `R C o f f s e t` の算出方法については後述する。

【0043】

リード回数 `R C` にオフセット値 `R C o f f s e t` を加算した値がリード回数閾値 `R C t h` を超えるブロックを検査対象ブロックとする。ステップ `S 1` において、検査対象ブロックが検出された場合、検査対象ブロックに対してステップ `S 2` ~ ステップ `S 1 2` の検査処理が行われる。

【0044】

ステップ `S 2` において、ページカウント値 `i` に “ 1 ” がセットされる。これにより、検査対象ブロックの先頭ページが設定される。また、ブロック最大エラー値 `E B m a x B l o c k` を 0 に初期化する。ブロック最大エラー値 `E B m a x B l o c k` は、検査対象ブロックに含まれる全ページのエラー検出値の中で、最大のエラー検出値を示す。

10

【0045】

ステップ `S 3` において、メモリコントローラ 3 は、メモリ 4 にアクセスし、検査対象ブロックの検査対象ページの読み出し処理を行う。検査対象ページとは、ページカウント値 `i` で特定されているページである。

【0046】

ステップ `S 4` において、ページ最大エラー値 `E B m a x P a g e` とエラー閾値 `E B t h` との比較が行われる。具体的には、`E C C` 回路 7 は、検査対象ページに対してエラー検出処理を行う。`E C C` 回路 7 のエラー検出単位の `B y t e` 長は、必ずしも、ページサイズと一致しない。`E C C` 回路 7 のエラー検出単位の `B y t e` 長が、ページサイズよりも小さい場合、検査対象ページに対して複数個のエラー検出値が得られることになる。メモリコントローラ 3 は、複数個のエラー検出値の中で最大値をページ最大エラー値 `E B m a x P a g e` として設定する。つまり、メモリコントローラ 3 は、ページ内の総エラー `B i t` 数を評価するのではなく、`E C C` 検出単位における最大エラー `B i t` 数を評価する。ブロック全体でエラー `B i t` 数が大きくなる前に、特定のページにおいてエラー `B i t` 数が増大した時点で、リードディスタープの発生を修復することができる。

20

【0047】

エラー閾値 `E B t h` は、リフレッシュを行うべきか否かを判定するための閾値である。ページ最大エラー値 `E B m a x P a g e` がエラー閾値 `E B t h` 以上となっているとき（ステップ `S 4` で `Y E S`）、リフレッシュが必要と判断され、ステップ `S 5` に移行する。ページ最大エラー値 `E B m a x P a g e` がエラー閾値 `E B t h` よりも小さいとき、まだリフレッシュを行う必要はないと判断され、ステップ `S 8` に移行する。

30

【0048】

たとえば、図 3 に示した例であれば、`N A N D` フラッシュメモリの 1 ページは、4 つの `E C C` 検出単位に区分される。それぞれの `E C C` 検出単位について、`5 B i t`、`7 B i t`、`3 B i t`、`0 B i t` のエラーが検出されている。この例であれば、ページ最大エラー値 `E B m a x P a g e` は `7 B i t` である。エラー閾値 `E B t h` が、たとえば `8 B i t` であれば、この例では、リフレッシュを行う必要はないと判定される。エラー閾値 `E B t h` が、たとえば、`7 B i t` であれば、検査対象ページを含む検査対象ブロックのリフレッシュが行われることになる。

40

【0049】

ステップ `S 5` において、メモリコントローラ 3 は、検査対象ブロックに対してリフレッシュを行う。具体的には、メモリコントローラ 3 は検査対象ブロックの全ページのデータを読み出し、バッファに格納する。次に、メモリコントローラ 3 は、検査対象ブロックの全ページのデータを消去する。そして、メモリコントローラ 3 は、`E C C` 回路 7 を利用してバッファに格納しているデータに対してエラー訂正を行った後、訂正後のデータを検査対象ブロックに書き戻す。これによりリフレッシュが完了する。

【0050】

ステップ `S 5` のリフレッシュ処理が完了すると、メモリコントローラ 3 は、消去カウン

50

タ 6 に格納されている検査対象ブロックの消去回数 EC を 1 インクリメントする (ステップ S 6)。

【 0 0 5 1 】

続いて、メモリコントローラ 3 は、リードカウンタ 5 に格納されている検査対象ブロックのリード回数 RC を 0 に初期化する (ステップ S 7)。つまり、検査対象ブロックに対してリフレッシュ処理を行ったので、検査対象ブロックに対して発生しているリードディスタープに基づくエラーは修復されている。そこで、検査対象ブロックについては、しばらくの期間、リフレッシュが不要となるので、リード回数 RC を 0 で初期化する。

【 0 0 5 2 】

ステップ S 7 の処理が完了すると、ステップ S 1 に戻り、再び、検査対象ブロックの存在の有無をチェックする。つまり、リード回数 RC にオフセット値 $RCoffset$ を加算した値がリード回数閾値 $RCth$ を超えるブロックが存在する否かをチェックする。ここで、ステップ S 7 の処理の後、即座にステップ S 1 に戻ってもよいが、所定のタイミングが到来した時点で、ステップ S 1 を再開するようにしてもよい。

【 0 0 5 3 】

再び、ステップ S 4 を参照する。ページ最大エラー値 $EBmaxPage$ がエラー閾値 $EBth$ より小さいとき (ステップ S 4 で NO)、ステップ S 8 に移行する。ステップ S 8 において、メモリコントローラ 3 は、ページ最大エラー値 $EBmaxPage$ とブロック最大エラー値 $EBmaxBlock$ とを比較する。

【 0 0 5 4 】

ページ最大エラー値 $EBmaxPage$ がブロック最大エラー値 $EBmaxBlock$ を超える場合 (ステップ S 8 で YES)、メモリコントローラ 3 は、最大エラー値 $EBmaxBlock$ にページ最大エラー値 $EBmaxPage$ の値をセットする (ステップ S 9)。つまり、検査対象ページで検出されたエラー検出値が、現時点で、エラー検出値の最大値となったので、ブロック最大エラー値 $EBmaxBlock$ を更新する。

【 0 0 5 5 】

ステップ S 10 において、ページカウント値 i が、検査対象ブロック内の最終ページを示す値かどうかを判定する。つまり、検査対象ページが検査対象ブロック内の最終ページであるか否かを判定する。検査対象ページが検査対象ブロック内の最終ページでない場合 (ステップ S 10 で NO)、ステップ S 11 において、ページカウント値 i を 1 だけインクリメントする。これにより、ステップ S 3 に戻り、次の検査対象ページについて、ステップ S 3 ~ S 12 に処理が行われる。

【 0 0 5 6 】

検査対象ページが検査対象ブロック内の最終ページである場合 (ステップ S 10 で YES)、リード回数 RC を初期化する (ステップ S 12)。ステップ S 7 においては、リード回数 RC を 0 に初期化した。しかし、ステップ S 12 においては、リード回数 RC は 0 に初期化されない。ステップ S 12 において、リード回数 RC の初期化値は、2 つの補正值に基づいて補正される。

【 0 0 5 7 】

メモリコントローラ 3 は、エラー閾値 $EBth$ に対するブロック最大エラー値 $EBmaxBlock$ の割合を、エラー進行割合として設定する。メモリコントローラ 3 は、エラー進行割合 ($EBmaxBlock / EBth$) に、リード回数閾値 $RCth$ を乗算した値を第 1 補正值として設定する。そして、メモリコントローラ 3 は、初期化値に、第 1 補正值を加算する。

【 0 0 5 8 】

また、メモリコントローラ 3 は、検査対象ブロックに含まれるページの数 $PageNum$ を、第 2 補正值として設定する。そして、メモリコントローラ 3 は、初期化値に、第 2 補正值を加算する。これにより、検査処理において読み出した回数も考慮して、初期化値が補正される。

【 0 0 5 9 】

10

20

30

40

50

このように、メモリコントローラ3は、第1補正值($RCth * (EBmaxBlock / EBth)$)および第2補正值($PageNum$)をリード回数RCの初期化値に加算する。したがって、この検査対象ブロックについては、再び、ステップS1においてリード回数閾値RCthとの比較が行われるとき、既に、補正值分だけリード回数が増加された状態となっている。ある程度、リードディスタurbによるエラー発生が進行していると考えられるので、リード回数RCを0に初期化せず、早い時期に再び検査対象となるように厳しくチェックするのである。これにより、リードディスタurbによるエラー発生が情報処理に影響を与えることを、未然に回避することができる。

【0060】

また、ステップS1においては、初期化されたリード回数RCには、オフセット値RCoffsetが加算される。オフセット値RCoffsetは、消去回数RCに応じて設定される第3補正值である。この実施の形態においては、消去カウンタ6に格納されている各ブロックの消去回数ECを1Bit左シフトすることにより、オフセット値RCoffsetを求めるようにしている。たとえば、消去回数ECが1回であれば、オフセット値RCoffsetは2である。消去回数ECが1000回であれば、オフセット値RCoffsetは、2000である。このように、検査処理の中で行われた消去回数も加味して、検査対象ブロックを選択するようにし、リードディスタurbによるエラー発生が情報処理に影響を与える前に、未然に修復することができる。もちろん、左シフトするときのシフト量は自由に設定することができる。また、左シフトは一例であり、消去回数ECが反映されるような補正值を適宜選択することができる。

【0061】

なお、消去カウンタ6に格納されている消去回数ECは、原則的には、初期化されない。したがって、累積的にリフレッシュ回数が増えているブロックについては、より厳しくチェックされることになる。

【0062】

{ 第2の実施の形態 }

次に、本発明の第2の実施の形態について説明する。第2の実施の形態については、リード回数RCおよび消去回数ECの集計単位が異なる。第1の実施の形態においては、リード回数RCおよび消去回数ECは、ブロック単位で集計した。第2の実施の形態においては、リード回数RCおよび消去回数ECは、バンク単位で集計する。

【0063】

図4のステップS1において、第1の実施の形態においては、検査対象ブロックの有無を検出した。第2の実施の形態においては、ステップS1において、検査対象バンクの存在を検出する。

【0064】

ステップS5において、第1の実施の形態においては、検査対象ブロックをリフレッシュした。第2の実施の形態においては、ステップS5において、検査対象バンクをリフレッシュする。

【0065】

ステップS8において、第1の実施の形態においては、ページ最大エラー値EBmaxPageとブロック最大エラー値EBmaxBlockとを比較した。第2の実施の形態においては、ステップS8において、ページ最大エラー値EBmaxPageとバンク最大エラー値EBmaxBankを比較する。つまり、第1の実施の形態においては、ページ最大エラー値EBmaxPageのブロック内での最大値を求めたが、第2の実施の形態においては、ページ最大エラー値EBmaxPageのバンク内での最大値を求める。

【0066】

ステップS10においては、第1の実施の形態においては、検査対象ブロック内の最終ページであるかどうかを判定した。第2の実施の形態においては、ステップS10において、検査対象バンク内の最終ページであるかどうかを判定する。

【0067】

10

20

30

40

50

ステップ S 1 2 において、第 1 の実施の形態においては、エラー閾値 E B t h に対するブロック最大エラー値 E B m a x B l o c k の割合をエラー進行割合とした。第 2 の実施の形態においては、ステップ S 1 2 において、エラー閾値 E B t h に対するバンク最大エラー値 E B m a x B a n k の割合をエラー進行割合とする。

【 0 0 6 8 】

第 2 の実施の形態においても、第 1 の実施の形態と同様、リードディスタープの発生しているページを検出し、情報処理にエラーが発生することを未然に防止できる。

【 0 0 6 9 】

第 1 の実施の形態と第 2 の実施の形態は、それぞれ異なる長所を備えているので、それらを使い分けるようにすればよい。第 1 の実施の形態においては、リフレッシュされる単位がブロック単位であるので、メモリ 4 に対する負荷を小さくすることができる。第 2 の実施の形態においては、バンク単位でカウント値を保持するので、カウント値を保持するために必要な記憶領域を小さくすることができる。

【 0 0 7 0 】

{ 変形例 }

この実施の形態では、第 1、第 2、あるいは第 3 補正值を利用して、リード回数 R C の初期化値を補正したが、リード回数閾値 R C t h を補正するようにしてもよい。

【 0 0 7 1 】

リード回数閾値 R C t h は、メモリ 4 が S L C (S i n g l e L e v e l C e l l) であるか、M L C (M u l t i L e v e l C e l l) であるかに応じて、最適な値に調整すればよい。また、リード回数閾値 R C t h は、B a n k サイズ、ベンダーに応じて最適な値に調整すればよい。同様に、エラー閾値 E B t h についても、S L C、M L C の種別、B a n k サイズ、ベンダーに応じて最適な値に調整すればよい。

【 0 0 7 2 】

この実施の形態において、ブロック単位で、あるいは、バンク単位で、ページに対する読み出しアクセス回数をカウントした。ブロック内の先頭アドレスのページ、あるいは、ブロック内の最終アドレスのページについては、リードディスタープ現象が発生しやすいことが判明している。そこで、ブロック内の先頭アドレスのページ、あるいは、ブロック内の最終アドレスのページについては、読み出し回数のカウントに重みをつけてもよい。たとえば、ブロック内の先頭アドレスのページ、あるいは、ブロック内の最終アドレスのページについては、1 回の読み出しに対して、リード回数 R C を 2 だけインクリメントするようにしてもよい。

【 符号の説明 】

【 0 0 7 3 】

- 1 ホストシステム
- 2 メモリモジュール
- 3 メモリコントローラ
- 4 メモリ
- 5 リードカウンタ
- 6 消去カウンタ
- 7 E C C 回路

10

20

30

40

【 図 1 】

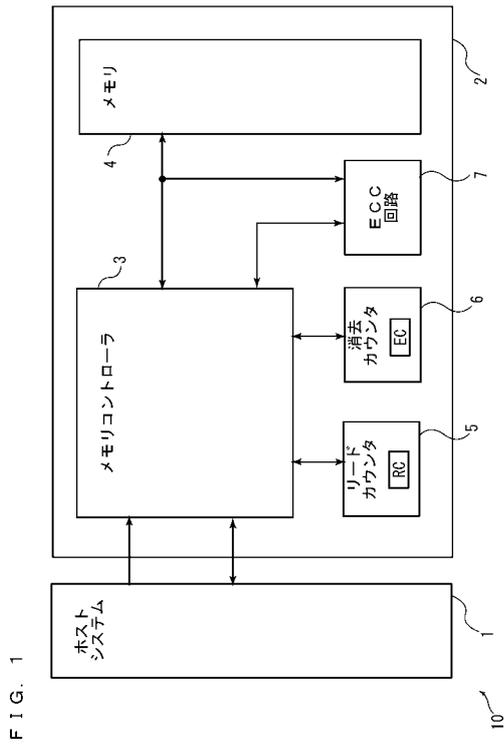
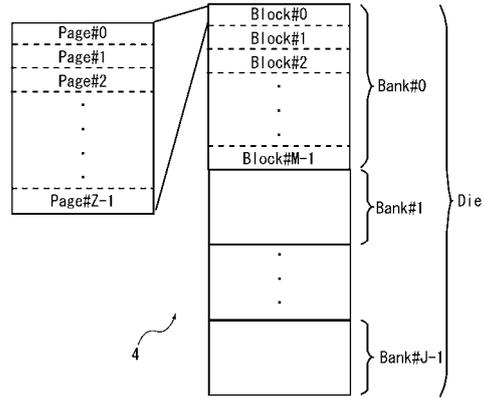


FIG. 1

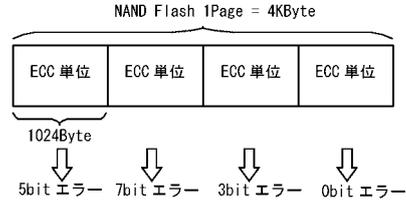
【 図 2 】

FIG. 2



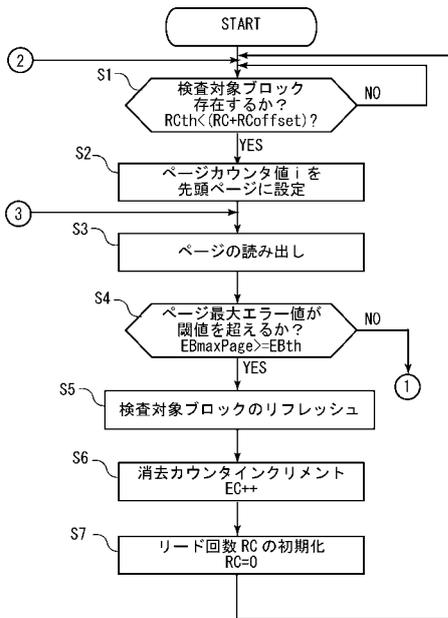
【 図 3 】

FIG. 3



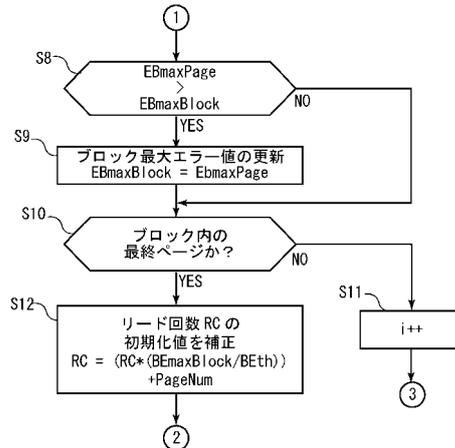
【 図 4 】

FIG. 4



【 図 5 】

FIG. 5



フロントページの続き

Fターム(参考) 5B018 GA04 HA14 HA22 HA23 KA22 MA24 NA06 QA05 QA11 QA14
QA15 QA16