

(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2015-172965

(P2015-172965A)

(43) 公開日 平成27年10月1日(2015.10.1)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 12/00 (2006.01)	G06F 12/00 597U	5B018
G06F 12/16 (2006.01)	G06F 12/16 310A	5B060
G06F 12/02 (2006.01)	G06F 12/02 530C	

審査請求 有 請求項の数 33 O L (全 49 頁)

(21) 出願番号	特願2015-108226 (P2015-108226)	(71) 出願人	595168543
(22) 出願日	平成27年5月28日 (2015. 5. 28)		マイクロン テクノロジー, インク.
(62) 分割の表示	特願2013-552536 (P2013-552536) の分割		アメリカ合衆国, アイダホ州 83716 -9632, ボイズ, サウス フェデ ラル ウェイ 8000
原出願日	平成24年1月12日 (2012. 1. 12)	(74) 代理人	100106851
(31) 優先権主張番号	13/019, 977		弁理士 野村 泰久
(32) 優先日	平成23年2月2日 (2011. 2. 2)	(74) 代理人	100074099
(33) 優先権主張国	米国 (US)		弁理士 大菅 義之
		(72) 発明者	ヴァン エーケン, スティーブン ピー. アメリカ合衆国, コロラド州 80301 , ボルダー, ブランディワイン コート 5958

最終頁に続く

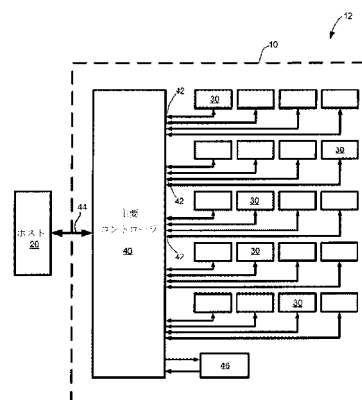
(54) 【発明の名称】 メモリシステム内の少なくとも半自律性のモジュールおよび方法

(57) 【要約】 (修正有)

【課題】 データ記憶容量を提供するようにホストデバイスとのデジタルデータ通信を行うためのメモリシステムを提供する。

【解決手段】 システム12は、コントローラ40及び複数のモジュール30を含む。各モジュールは、不揮発性メモリデバイスを含む。モジュールは、モジュールについて少なくとも部分的にパラメータに基づいて管理機能を行うように、構成される。パラメータは、コントローラ及び／又はモジュールによって提供される。システム及びモジュールは、1つの機能において、モジュールを用いたデータアクセスについて、複数の形態の並行処理を支援する。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

コントローラと、

前記コントローラに接続されたメモリモジュールであって、前記モジュールは、前記コントローラから供給される制御パラメータと、前記モジュール自身が発生するモジュールパラメータとに基づいて、前記モジュールについての管理機能を行うように構成されている、モジュールと、

を含むことを特徴とするメモリシステム。

【請求項 2】

前記モジュールが、前記管理機能を前記コントローラとの相互作用なしに実行すること
を特徴とする請求項 1 に記載のメモリシステム。 10

【請求項 3】

前記モジュールが前記管理機能を前記コントローラの制御の下実行することを特徴とする請求項 1 に記載のメモリシステム。

【請求項 4】

前記管理機能が、ウェアレベリング機能を含むことを特徴とする請求項 1 乃至 3 のいずれか一項に記載のメモリシステム。

【請求項 5】

前記管理機能が、ガーベジコレクション機能を含むことを特徴とする請求項 1 乃至 4 のいずれか一項に記載のメモリシステム。 20

【請求項 6】

前記管理機能が、ビット密度構成機能を含むことを特徴とする請求項 1 乃至 5 のいずれか一項に記載のメモリシステム。

【請求項 7】

前記メモリモジュールが複数のメモリ部を含み、前記ビット密度構成機能は、前記複数のメモリ部の各々に独立にビット密度構成を付与することを特徴とする請求項 6 に記載のメモリシステム。

【請求項 8】

前記管理機能が、フラッシュ変換層管理機能を含むことを特徴とする請求項 1 乃至 7 のいずれか一項に記載のメモリシステム。 30

【請求項 9】

前記メモリモジュールが複数のメモリ部を含み、前記フラッシュ変換層管理機能は、前記複数のメモリ部に対するフラッシュ変換層を互いに独立に管理することを特徴とする請求項 8 に記載のメモリシステム。

【請求項 10】

外部のコントローラから制御パラメータを受け取り、

メモリモジュールの内部でモジュールパラメータを発生し、

前記制御パラメータと前記モジュールパラメータとに基づいて、前記メモリモジュールについての管理機能を実行する、メモリシステムの制御方法。

【請求項 11】

前記管理機能の実行が、前記コントローラとの相互作用なしに実行されることを特徴とする請求項 10 に記載の方法。 40

【請求項 12】

前記管理機能の実行が、前記コントローラの制御の下実行されることを特徴とする請求項 10 に記載の方法。

【請求項 13】

前記管理機能が、ウェアレベリング機能を含むことを特徴とする請求項 10 乃至 12 のいずれか一項に記載の方法。

【請求項 14】

前記管理機能が、ガーベジコレクション機能を含むことを特徴とする請求項 10 乃至 1 50

3のいずれか一項に記載の方法。

【請求項15】

前記管理機能が、ビット密度構成機能を含むことを特徴とする請求項10乃至14のいずれか一項に記載の方法。

【請求項16】

前記管理機能が、フラッシュ変換層管理機能を含むことを特徴とする請求項10乃至15いずれか一項に記載の方法。

【請求項17】

各々が不揮発性メモリ部を含む複数のモジュールであって、少なくとも1つのモジュールが、当該少なくとも1つのモジュールのみの前記不揮発性メモリ部についてのローカルフラッシュ変換層を、モジュール入力パラメータに応じて、独立に管理するように構成された、モジュールを含むことを特徴とするメモリシステム。

10

【請求項18】

前記モジュール入力パラメータを前記少なくとも1つのモジュールに供給して、前記ローカルフラッシュ変換層の独立動作を規定する、コントローラを更に含むことを特徴とする請求項17に記載のメモリシステム。

【請求項19】

前記モジュール入力パラメータを前記少なくとも1つのモジュールに供給するコントローラを更に含み、前記少なくとも1つのモジュールは、さらに、第1の動作モードにおいて前記ローカルフラッシュ変換層の独立動作を制御するために、前記コントローラから第1の組のフラッシュ変換層パラメータを受け取るフラッシュ変換層パラメータ部を含むことを特徴とする請求項17又は18に記載のメモリシステム。

20

【請求項20】

前記フラッシュ変換層パラメータ部が、さらに、第2の動作モードにおいて前記ローカルフラッシュ変換層の独立動作を制御するために、前記モジュール自身が発生する第2の組のフラッシュ変換層パラメータを受け取ることを特徴とする請求項19に記載のメモリシステム。

【請求項21】

前記モジュール入力パラメータを前記少なくとも1つのモジュールに供給するコントローラを更に含み、前記少なくとも1つのモジュールは、さらに、前記少なくとも1つのモジュールが前記ローカルフラッシュ変換層への管理を独立に行う第1の動作モードと、前記コントローラの部分的制御の下、前記ローカルフラッシュ変換層への管理を行う第2の動作モードとを、選択的に実行することを特徴とする請求項17乃至20のいずれかに記載のメモリシステム。

30

【請求項22】

前記少なくとも1つのモジュールが、更に、ウェアレベリング機能、ガーベジコレクション機能、及び、ビット密度構成機能のうちの少なくとも1つを独立に実行する機能エンジンを含むことを特徴とする請求項17乃至21のいずれかに記載のメモリシステム。

【請求項23】

前記機能エンジンは、前記少なくとも1つのモジュールの前記不揮発性メモリ部に関連する利用統計を、少なくとも前記機能エンジンで利用可能とするために、当該利用統計を報告する報告部を備えることを特徴とする請求項22に記載のメモリシステム。

40

【請求項24】

前記報告部は、読み出し動作に応じて複数のリードバック値を発生し、前記利用統計は、前記読み出し動作の間の1組のリードバック値に対する平均リードバック値と、前記読み出し動作の間の当該1組のリードバック値に対する標準偏差とのうちの少なくとも1つと、1または複数の誤差修正統計を含むことを特徴とする、請求項23に記載のメモリシステム。

【請求項25】

前記1又は複数の誤差修正統計は、ブロック誤差修正コードをデコードする前記読み出

50

し動作に基づく、累積誤差カウント及びブロック毎の誤差カウントの少なくとも1つを含むことを特徴とする請求項24に記載のメモリシステム。

【請求項26】

前記報告部は、前記利用統計に基づく、前記不揮発性メモリ部に関連する健康指標を提供することを特徴とする請求項23乃至25のいずれかに記載のメモリシステム。

【請求項27】

各々が不揮発性メモリ部を含む複数のモジュールであって、少なくとも1つのモジュールが、モジュール入力パラメータに応じて、当該少なくとも1つのモジュールのみの前記不揮発性メモリ部についてのウェアレベリング機能、ガーベジコレクション機能、及び、ビット密度構成機能のうちの少なくとも1つを独立に実行すると共に、エラー訂正動作を独立に実行するように構成された、モジュールと、

前記モジュール入力パラメータを前記少なくとも1つのモジュールに供給して、前記ウェアレベリング機能、前記ガーベジコレクション機能、及び、前記ビット密度構成機能のうちの少なくとも1つの独立動作を規定する、コントローラと、

を含むことを特徴とするメモリシステム。

【請求項28】

前記エラー訂正動作と、前記少なくとも1つのモジュールのみの前記不揮発性メモリ部についての前記ウェアレベリング機能、前記ガーベジコレクション機能、及び、前記ビット密度構成機能のうちの少なくとも1つとを独立に実行する機能エンジンをさらに含むことを特徴とする請求項27に記載のメモリシステム。

【請求項29】

前記機能エンジンは、前記少なくとも1つのモジュールの前記不揮発性メモリ部に関連する利用統計を、少なくとも前記機能エンジンで利用可能とするために、当該利用統計を報告する報告部を備えることを特徴とする請求項28に記載のメモリシステム。

【請求項30】

前記報告部は、前記利用統計に基づく、前記不揮発性メモリ部に関連する健康指標を提供することを特徴とする請求項29に記載のメモリシステム。

【請求項31】

前記少なくとも1つのモジュールは、第1の動作モードにおいて、前記モジュール入力パラメータのうちの第1の組に基づいて、前記ウェアレベリング機能、前記ガーベジコレクション機能、及び、前記ビット密度構成機能のうちの選択された少なくとも1つを実行し、第2の動作モードにおいて、前記モジュール入力パラメータのうちの、前記第1の組と異なる第2の組と前記第1の組とに基づいて、前記ウェアレベリング機能、前記ガーベジコレクション機能、及び、前記ビット密度構成機能のうちの前記選択された少なくとも1つを実行することを特徴とする請求項27乃至30のいずれかに記載のメモリシステム。

【請求項32】

前記少なくとも1つのモジュールが、さらに、当該少なくとも1つのモジュールのみの前記不揮発性メモリ部についてのローカルフラッシュ変換層を、前記モジュール入力パラメータに応じて、独立に管理することを特徴とする請求項27乃至31のいずれかに記載のメモリシステム。

【請求項33】

前記少なくとも1つのモジュールが、さらに、前記ローカルフラッシュ変換層への管理を独立に行う第1の動作モードと、前記コントローラの部分的制御の下、前記ローカルフラッシュ変換層への管理を行う第2の動作モードとを、選択的に実行することを特徴とする請求項32に記載のメモリシステム。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

本発明は、不揮発性メモリ（例えば、ソリッドステートドライブ（「SSD」）および他の記憶装置）を用いたメモリシステムの分野に主に関し、より詳細には、メモリシステム内において用いられる少なくとも半自律性のモジュールおよび関連方法と、関連機能とに関する。

【背景技術】

【0002】

従来技術のソリッドステートドライブは典型的には、コントローラを含む。コントローラは、標準的ホストデバイスおよび複数のフラッシュメモリデバイスとインターフェースをとり得る。これらのフラッシュメモリデバイスはそれぞれ、概してメモリダイの形態をとる。コントローラとフラッシュメモリデバイスとの間の接続は典型的には、フラッシュ
10
オリエンテッドインターフェースの形態をとり、各メモリデバイスは、フラッシュオリエンテッドインターフェースへと個別に接続される。コントローラは、複数のフラッシュオリエンテッドインターフェースを持ち得る。これらのインターフェースはそれぞれ、複数のメモリデバイスへと接続される。出願人は、この従来技術構成の場合、特に大容量のソリッドステートドライブの場合において複数の不利点があることを認識する。例えば、コントローラは、多数の個々のメモリデバイスを制御する必要がある。別の例として、コントローラ上に必要な信号接続点数が、ドライブの容量に比例して増加する。さらに別の例において、コントローラは、管理機能（例えば、オーバーヘッド機能（例えば、ウェアレベリング）を担当し、そのため、コントローラとメモリデバイスとの間のインターフェースが、これらの機能に関連するトラフィックの負担を負う。さらに別の例として、多数の
20
メモリデバイスの試験および監視をコントローラによって行うことは、複雑なタスクである。

【0003】

出願人はまた、従来技術における不揮発性メモリをブロックオリエンテッド不揮発性メモリについて用いた場合、関連して懸念が発生することを認識している。ブロックオリエンテッド不揮発性メモリへのアクセスについての従来技術の一般的アプローチの場合、ユーザデータおよび付随する追加情報を高精度に物理ページ上へと割り当てる方法が用いられることが理解されるべきである。本議論の目的のため、「ブロックオリエンテッド」という用語は、ページベースのメモリの利用に関連する。ページベースのメモリは、ユーザデータブロックを用いることにより、用いられる。これらのユーザデータブロックの長さ
30
は、概してページ長さに等しい。さらに説明するように、ブロックオリエンテッドメモリについて、いくつかのページは、1ページ長さよりも短いユーザデータブロックに基づいて部分的に非使用である場合があるが、ユーザデータブロックは、ページ長さよりも長くない。非限定的な例により、NANDフラッシュは典型的には、この種のブロックオリエンテッドメモリとなるように構成される。そのため、このようにブロックオリエンテッドメモリにおける割当を非柔軟に行うことが、記憶情報と各物理ページを含む記憶素子数との間に正確な対応を発生させ、シングルビットパーセルメモリ（SBC）およびマルチビットパーセルメモリ（MBC）双方において業界において慣習となっている。

【0004】

なお、SSDを用いた従来メモリシステムの一例が、特許文献1に記載されている。

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特表2010-538372号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

出願人は、ユーザデータを特定のメモリデバイス中のページ単位に整合させる場合、上述したようなアプローチは柔軟性に欠けることをさらに認識している。物理ページサイズは、メモリの設計パラメータであり、変更することができない。物理ページ内の記憶素子
50

数と、アプリケーションにおいてブロック毎に必要な素子数との間に不整合が発生した場合、ペナルティが発生してしまう。このようなペナルティは、ECC要求を満たさない不十分な素子の形態をとり得、その結果、アプリケーションの信頼性低下に繋がる場合もあるし、あるいは、このようなペナルティは、各物理ページ内に必要な素子数を超える素子数となり得、その結果記憶が不十分になる場合もある。そのため、不揮発性メモリデバイス（例えば、NANDフラッシュ）の設計者にとって、メモリ製造の前に行われる必要なページサイズの予測する作業が困難になる。そのため、当該メモリが用いられるアプリケーションのうち少なくとも一部に対して、所与のページサイズが不適切になることがほとんど回避できなくなる。

【0007】

10

上記の関連技術およびその制約の例は例示的なものであり、排他的なものではない。当業者であれば、本明細書および図面を読めば、関連分野の他の制約を理解する。

【課題を解決するための手段】

【0008】

上記の課題に鑑み、本発明のメモリシステムの一態様は、コントローラと、前記コントローラに接続されたメモリモジュールであって、前記モジュールは、前記コントローラから供給される制御パラメータと、前記モジュール自身が発生するモジュールパラメータとに基づいて、前記モジュールについての管理機能を行うように構成されている、モジュールと、を含むことを特徴とするものである。

【0009】

20

また、本発明のメモリシステムの制御方法の一態様は、外部のコントローラから制御パラメータを受け取り、メモリモジュールの内部でモジュールパラメータを発生し、前記制御パラメータと前記モジュールパラメータとに基づいて、前記メモリモジュールについての管理機能を実行する、ことを特徴とするものである。

【0010】

更に、本発明のメモリシステムの他の態様は、各々が不揮発性メモリ部を含む複数のモジュールであって、少なくとも1つのモジュールが、当該少なくとも1つのモジュールのみの前記不揮発性メモリ部についてのローカルフラッシュ変換層を、モジュール入力パラメータに応じて、独立に管理するように構成された、モジュールを含むことを特徴とするものである。

30

【0011】

加えて、本発明のメモリシステムの更に他の態様は、各々が不揮発性メモリ部を含む複数のモジュールであって、少なくとも1つのモジュールが、モジュール入力パラメータに応じて、当該少なくとも1つのモジュールのみの前記不揮発性メモリ部についてのウェアレベリング機能、ガーベジコレクション機能、及び、ビット密度構成機能のうちの少なくとも1つを独立に実行すると共に、エラー訂正動作を独立に実行するように構成された、モジュールと、前記モジュール入力パラメータを前記少なくとも1つのモジュールに供給して、前記ウェアレベリング機能、前記ガーベジコレクション機能、及び、前記ビット密度構成機能のうちの少なくとも1つの独立動作を規定する、コントローラと、を含むことを特徴とするものである。

40

【0012】

図面中、例示的实施形態を示す。本明細書中に開示される実施形態および図は例示的なものであり、限定的なものではないことが意図される。

【図面の簡単な説明】

【0013】

【図1】本開示の実施形態によるSSDを示すブロック図である。SSDは、システム全体の一部として図示されている。

【図2】本開示の実施形態によるSSDのさらなる詳細を示すブロック図である。

【図3】SSD（例えば、図1および／または図2のSSD）内において用いられるモジュールの実施形態のブロック図である。

50

【図４】モジュール（例えば、図３のモジュール）と共に用いられるモジュールインターフェースの実施形態のブロック図である。

【図５】モジュールのメモリデバイスを模式的に示すブロック図であり、モジュールは、これらのメモリデバイス上のメモリ部の配置構成と共に用いられる。

【図６】モジュール（例えば、図３のモジュール）において用いることが可能な機能エンジンの実施形態のブロック図である。

【図７】図６の機能エンジンを用いて行うことが可能なウェアレベリング機能の実施形態のフロー図である。

【図８】図６の機能エンジンを用いて行うことが可能なガーベジコレクション機能の実施形態のフロー図である。

10

【図９】図６の機能エンジンを用いて行うことが可能なビット密度構成機能の実施形態のフロー図である。

【図１０】図６の機能エンジンを用いて行うことが可能なＦＴＬ（フラッシュ変換層）機能の実施形態のフロー図である。

【図１１】図６の機能エンジンを用いて行うことが可能な読出動作の実施形態のフロー図である。

【図１２】図６の機能エンジンを用いて行うことが可能な書込動作の実施形態のフロー図である。

【図１３】図６の機能エンジンを用いて行うことが可能な消去機能の実施形態のフロー図である。

20

【図１４】物理ページの長さに相対して異なる長さを有するユーザデータブロックの模式図である。

【図１５】１組のユーザデータブロックを保存する１組の連続する物理ページのブロック図である。これらのユーザデータブロックそれぞれの長さは、１つの物理ページの長さよりも短い。

【図１６】１組のユーザデータブロックを保存する１組の連続する物理ページのブロック図である。これらのユーザデータブロックそれぞれの長さは、１つの物理ページの長さよりも長い。

【図１７】ユーザデータブロックの保存／書込方法の実施形態のフロー図である。この方法は、図１５および図１６の文脈に記載されるデータ構造を達成する。図１５および図１６の文脈において、ブロック長さおよびページ長さは異なり得る。

30

【図１８】ブロック長さおよびページ長さが異なり得る図１５および図１６の文脈において上述したデータ構造からユーザデータブロックらを読み出すための方法の実施形態を示す別のフロー図である。

【発明を実施するための形態】

【００１４】

以下の記載は、当業者が本発明を作製および利用できるようにするためのものであり、特許出願およびその要求の文脈において記載される。当業者であれば、記載の実施形態において、多様な改変例を想起し、本明細書中に教示される一般的原理は、他の実施形態に適用され得る。よって、本発明は、記載の実施形態に限定されるものではなく、本明細書中に記載の原理および特徴ならびに添付の特許請求の範囲中に記載のその改変例および均等例に基づいた最大の範囲と一致する。図面は縮尺通りではなく、対象となる特徴を最良に例示するために模式的に示したものである点に留意されたい。記述用語は、図面中に記載される多様な図面について読者の理解の向上のために用いられるものであり、限定的なものではない。

40

【００１５】

以下、図面を参照する。図面中、多様な図面中、類似の要素は、類似の構成要素を指し得る。図１は、本開示の実施形態によるソリッドステートドライブ（ＳＳＤ）１０のブロック図である。ソリッドステートドライブ（ＳＳＤ）１０は、システム１２全体の一部を形成する。システム１２において、ソリッドステートドライブは、ホスト２０とインター

50

フェースをとる。図2の例示的实施形態において、SSD10の記憶容量は、合計20個のモジュールによって得られる。これらのモジュールのうちいくつかを参照番号30によって示す。本明細書中の教示範囲内において、任意の適切な数のモジュールを用いることが可能である。各モジュールは、不揮発性メモリ(「NVM」)部(例えば、複数のNVMデバイス(例えば、複数のNVMダイ))を含む。不揮発性メモリはフラッシュメモリであり得るが、任意の適切なメモリ技術を用いることが可能である(例えば、相変化メモリ(「PCM」)および他の形態の可変抵抗メモリ)。図1のモジュール配置構成は、「スペア」モジュール(例えば、合計モジュール数のうち一部の利用の損失を確保するためのもの)の利用をサポートすることが理解されるべきである。SSDの目標合計容量が1テラバイト(1TB)である場合、16個のモジュールそれぞれの容量を64ギガバイト(64GB)にすることで、この目標を満足させることができる。よって、合計20個のモジュールを用意した場合、5個のモジュールが余分になり、スペアとして用いることができる。単一のモジュール中に複数のNVMデバイスを含めることにより、読出動作および書込動作時におけるコントローラに直接接続される個々のメモリデバイス数を、メモリモジュール内のNVMダイ数の倍数だけ低減させることが可能になる。

10

【0016】

理解されるように、モジュールとの通信は、モジュールインターフェースを介して提供される。モジュールインターフェースは、ポイントツーポイントインターフェースであり得、モジュールインターフェースの独立インスタンスをシステム内の各モジュールのために設けることができる。これは、低モジュールピンカウントに起因して实际的であり得る。このような低ピンカウントにより、ASIC形態のSSDコントローラの従来のパッドリング制約内において多数のモジュールインターフェースポートを得ることが可能になり、各ポートにより、1つのモジュールに対するインターフェースが得られる。モジュールおよびSSDコントローラポートのポイントツーポイント相互接続により、特定のモジュールインターフェースのリソースをSSDコントローラと当該特定のモジュールとの間のトランザクションへ専用に割り当てることが可能になる。主要コントローラと各モジュールとの間のモジュールインターフェースによって確立された相互接続が専用のものである場合、主要コントローラとモジュールとの間のトランザクションは、インターフェースのリソースを他のモジュールと共有する必要が無い。そのため、これらのトランザクションを各モジュールインターフェースの全帯域幅において実行することが可能になる。また、接続が1つのモジュールの専用である場合、インターフェースが1つよりも多くのモジュールへ接続された場合に必要となるアービトレーションおよびリソース共有を回避することが可能になる。その結果、トランザクション数およびインターフェースを介して転送されるデータ量双方において性能が向上するため、システム性能の向上が可能になる。

20

30

【0017】

再度図1を参照して、コントローラ40は、各モジュールのためのモジュールインターフェース42を介して各モジュール30とインターフェースをとる(モジュール30のうちいくつかを参照番号を用いて示す)。コントローラは、適切な高レベルホストインターフェース44を介してホスト20とインターフェースをとる。高レベルホストインターフェース44は、例えばメモリシステムをコンピュータへ取り付けるための多様なインターフェースのうち任意のものでよい。例示目的のため、ホストインターフェース44は、何らかのシステム12のためのSATA(シリアルATA)インターフェースであり得る。他のシステム12において、SAS(直列に取り付けられたSCSI)インターフェースを用いてもよいし、あるいはインターフェースをバックプレーンオリエンテッドとしてもよい(例えば、PCIエクスプレスを用いたもの)。この点について、特定の最終用途に合わせて、適切な形態のホストインターフェースを選択することができる。

40

【0018】

例えばSSDシステム10全体の性能を向上させる目的のために、キャッシュ46を設けることができる。キャッシュは、任意の適切なメモリデバイス(例えば、SDRAM、PCM、高速フラッシュ(例えば、SBC))を取り付けることにより、実現することが

50

できる。SSDコントローラ40は、1つ以上のキャッシュデバイスとの通信のための専用インターフェースを持ち得る。特定のSSD設計に合わせて選択されたキャッシュ技術の特性に応じて、このインターフェースを当該キャッシュ専用としてもよい。キャッシュデバイス46がモジュールである場合、キャッシュインターフェースは、NVMモジュールインターフェースと同一または類似のものでよい。

【0019】

コントローラ40と従来技術SSDのためのコントローラとを比較した場合、コントローラ40はより簡略化されたものとみなすことができる。システム10内のモジュールは、以下に記載する詳細な管理機能によって給電可能であるため、SSDコントローラは、従来技術コントローラによって行われることが多かった多数の機能を負担しなくてよくなる。オフロードされる機能の非限定的な例は、少なくとも以下を含み得る：(1)ウェアレベリング、(2)ガージコレクション、(3)論理／物理アドレス変換、(4)物理ページ／論理ブロック抽象化、(5)ECC機能および(6)メモリ調整および信号処理。機能のうち多くは、高速CPUおよび従来技術SSDコントローラにおける他の支援動作(例えば、上述したもの)を必要とするものを低減または除去することが可能であるため、SSDコントローラの帯域幅のうちより多くを、ホストと接続先モジュールとの間のユーザデータ転送に専用に用いることが可能になる。

【0020】

ここで図2を図1と共に参照して、図2は、コントローラ40の実施形態を示すブロック図である。コントローラは、複数のコントローラモジュールポート100を含む。これらのコントローラモジュールポート100はそれぞれ、複数の記憶モジュール30(図1)との通信のためのモジュールインターフェース42を提供する。各コントローラモジュールポート100は、関連付けられたインターフェース信号接続部を1組提供する。これらのインターフェース信号接続部は、コントローラモジュールポート間において同じである。非限定的な例として、コントローラ40は、図1に示すように、20個の個々のモジュールインターフェース42を介して20個のモジュール30と通信することができる。SSDコントローラ中のコントローラモジュールポートの数は、変動し得る。所与のSSDコントローラ中において実装される特定の数のコントローラモジュールポートは、SSDコントローラそのものを含む設計事項ならびにSSDシステムに依存し得る。これらの設計事項は、SSDの所望の記憶容量、所望の性能および所望のコストを含み得る。コントローラモジュールインターフェースの恩恵は、特定の実施形態のSSDコントローラにおいて実装されるコントローラモジュールポート数に関係無く、提供される。

【0021】

主要コントローラ40の説明を続ける。データ経路および制御論理部120は、各活性コントローラモジュールポート100およびSSDコントローラの残りの部分との間で、双方向データ経路(例えば、バス接続部122(そのうちいくつかを参照番号によって示す))を用いてデータ転送を行うように構成される。制御バス124は、マイクロプロセッサ130とレジスタ134との間のデータ経路を提供することができる。マイクロプロセッサ130は、メモリ132中に保存されたコードを用いて動作する。レジスタ134は、SSDコントローラを構成する論理ブロック中に含まれる。各コントローラモジュールポートがバス124へと接続されることで、マイクロプロセッサ130による制御が促進される。よって、マイクロプロセッサ130は、各モジュールポート中のレジスタ136からの読み出しおよび各モジュールポート中のレジスタ136への書き込みを行うことができ、このような読み出しおよび書き込みを行うことで、各コントローラモジュールポートの動作(例えば、設定、起動、停止および監視)を管理することができる。データバッファ部140は、SSDコントローラ中を流れるデータのブロックをホストインターフェースポート144を通じてホストへ転送されるべき読み出しデータとしてまたはコントローラモジュールポート100のうちの1つを通じてモジュールへと転送されるべき書き込みデータとして中継するためのメモリを提供することができる。双方向データ経路146(そのうちいくつかを参照番号を用いて示す)は、データ経路および制御部120と、

データバッファ部 140 との間に設けられ得る。双方向データ経路 146 の数とコントローラモジュールポート 100 の数との間の一対一の対応は不要であることが理解されるべきである。この点について、十分なデータ帯域幅全体を確保するために、十分な数の別個の双方向データ経路 146 を設けることができる。すなわち、双方向データ経路 146 の合計数は、コントローラモジュールポートの合計数未満であり、モジュールポートの合計数に等しくあるか、またはコントローラモジュールポートの合計数を超え得る。

【0022】

メモリ 132 は、マイクロプロセッサ 130 によってアクセス可能であり、これにより、SSD コントローラを管理する機能（例えば、SSD コントローラの通常動作時において発生し得るようなエラーおよび例外（例えば、モジュールのうち 1 つ以上から受信されたデータ中のエラーに起因するもの、SSD コントローラ中の例外条件に起因するもの（例えば、リソース矛盾、バッファメモリエラー）、および他の通常動作時において発生する予期しない問題に起因するもの）が行われる。

【0023】

高速データ経路（例えば、バス 150）を用いて、バッファ部 140 とホストインターフェースポート 144 との間のデータ転送を行うことができる。このバスは、例えば、以下の項目について SSD システム要求をサポートする任意の適切な設計であればよい：ホストインターフェースを通じた転送速度、ホストインターフェースを通じて発行されたコマンドとこれらのコマンドに対する応答との間の待ち時間、および複数のコマンドの同時実行をサポートする並行度の集計。ホストインターフェースポート 144 は、制御バス 124 を介してマイクロプロセッサ 130 と通信して、マイクロプロセッサがホストインターフェース 44 を管理（例えば、制御および／または監視）することが可能になる。図 2 中においては単一のホストインターフェース 44 のみを図示しているが、1 つよりも多くのホストインターフェースが所与の SSD コントローラ実施形態中において存在することができる。複数のホストインターフェースを用いた実施形態において、ホストインターフェースの各インスタンスは、同一種類のインターフェース（複数の SAS ポートの場合と同様の）であってもよいし、あるいは異なる種類のインスタンス（例えば、SAS、SATA および PCI エクスプレスの一定の組み合わせ）であってもよい。

【0024】

SSD コントローラインターフェース 44 へのホストは、ホストインターフェースポート 144 によって提供され得る。このようなポート 144 は、ホストインターフェース定義に必要な全動作を行うように構成された信号接続部および／または回路を含む。SAS または SATA インターフェースがホストインターフェースポート 144 によって提供された場合、物理的接続は、最少で 4 つの信号接続部によって構成される。これら 4 つの信号接続部は、2 つの差動対として構成される。他の実施形態において、例えば SAS が用いられる場合および第 2 のインターフェース信号接続部が用いられる場合、さらなる差動信号対が存在し得る。SAS の場合、1 つよりも多くのホストポートが用いられ得る。このような実施形態において、1 組ではなく 2 組の物理的コネクタが存在し得る。

【0025】

ここで図 3 を参照して、図 3 は、本開示の実施形態に従って生成されたモジュール 30（図 1 も参照）を示すブロック図である。モジュールは、複数のメモリデバイス 170 を含み得る。メモリデバイスの合計数は、1～8 以上の範囲で変動し得、メモリデバイスは、任意の適切な技術に基づき得る。いくつかの実施形態において、メモリ技術は NAND フラッシュである。モジュールのメモリ全体を構成するデバイスは、標準的な市販の部品であってもよいし、あるいはモジュール単位でカスタマイズされたデバイスであってもよい。例えば、取り付けられたメモリデバイスの同時動作の増加および／または SSD 中のデータスループットの増加のために複数のメモリデバイス 170 をメディアポート 176 に取り付けることができる。図 3 に示すように、いくつかの実施形態において、各メディアポート 176 は、メモリデバイス群に対してインターフェースを提供することができる。

。

10

20

30

40

50

【0026】

さらに図3を参照して、各メディアポート176は、メモリデバイスインターフェース180を提供する。メモリデバイスインターフェース180を用いて、複数のメモリデバイスと通信することができる。さらに、各モジュール30は、複数のメディアポート176を含み得る。いくつかの実施形態において、メディアポートの数は、2個または4個である。各メディアポート176は、標準的NANDフラッシュデバイスと通信するための業界標準のインターフェースを提供し得る（例えば、ONFI（オープンNANDフラッシュインターフェース））。あるいは、各メディアポートは、カスタマイズされたメモリデバイスとの通信のためのカスタムインターフェースを提供し得る。上述したように、1つ以上のメモリデバイス170は、各メモリデバイスインターフェース180を介して通信し得る。いくつかの実施形態において、数は、1～4の範囲で変化し得る。さらに、各メディアポートに同一数のメモリデバイスを接続すべき要求は無い。さらに、いくつかの場合において、特定のメディアポートに対し、メモリデバイスがまったく接続されていない場合もある。このような状況は、例えば、特定のモジュール配置構成において利用可能なメディアポート全てが用いられるわけではない場合において発生し得る。

【0027】

データ経路および制御部190は、モジュール30の一部を形成し、多様なシステムリソースをメディアポートそのものと共に用いて複数のメディアポートへのデータトラフィックおよび複数のメディアポートからのデータトラフィックを管理（例えば、調整）する状態で、メディアポート176へのデータトラフィックおよびメディアポート176からのデータトラフィックを方向付けるように構成され得る。複数のデータ経路は、各メディアポート176と残りのモジュールとの間のデータフロー中においてサポートされ得る。デコーダ読み出しデータ経路200は、デコーダ202へのデータ経路と、データバッファ部210への直接データ経路204と、読み出し論理212へのデータ経路と、少なくともデータ経路および制御部190中を流れている（読み出しモードにおける）1つのメディアポートから（書き込みモードにおける）別のメディアポートへのデータ経路とを含む。書き込みデータ経路は、エンコーダ220からのデータ経路と、データバッファ部210からの直接データ経路230と、書き込み論理222からのデータ経路とを含み得る。全てではないにしろ、これらの多様なデータ経路上に発生し得るデータフローを他のデータフロー全てと同時に保持することが可能である点に留意されたい。このような並行処理を保持することが可能な状態のうちいくつかを以下に記載する。

【0028】

a. 1つのメディアポート176を用いて、取り付けられたメモリデバイス170のうち1つに対して第1の書き込み機能を行う。第1の書き込み機能によって用いられるデータ経路は、モジュール内の利用可能な書き込みデータ経路のうちの1つである。同時に、別のメディアポート176を用いて、取り付けられたメモリデバイス170のうちの1つに対して第2の書き込み機能を行う。第2の書き込み機能において用いられるデータ経路は、第1の書き込み機能によって未だ用いられていない残りの書き込みデータ経路のうちいずれかでよい。2つのメディアポートが同時書き込み機能を行うため、書き込みモードデータを受信する2つのメモリデバイスは別個のデバイスである。単一のメモリデバイスが2つのメディアポートからの同時コマンドの標的である場合、このシナリオは不可能である点に留意されたい。なぜならば、各メモリデバイスは1つのメディアポートのみに取り付けられるからである。

【0029】

b. 第1のメディアポート176を用いて、取り付けられたメモリデバイス170のうち1つに対して書き込み機能を行う。書き込み機能によって用いられるデータ経路は、モジュール内の利用可能な書き込みデータ経路のうちの1つである。同時に、第2のメディアポート176を用いて、取り付けられたメモリデバイスのうち1つから読み出し機能を行う。読み出しデータ経路は、第2のメディアポートにとって利用可能な任意の読み出しデータ経路でよい。

10

20

30

40

50

【0030】

c. 単一のメディアポート176を用いて、1つよりも多くのコマンドを同時に行うことができる。これは、同一メディアポート176に取り付けられた2つ以上のメモリデバイスそれぞれの上の活性コマンドとして発生させることもできるし、あるいは、同一メディアポート176に取り付けられた単一のメモリデバイス上の複数のコマンドとして発生させることもできる。いずれの状況においても、コマンドの並行処理を行うためには、モジュールの特定のメディアポート176およびデータ経路および制御部190によって提供された単一のメディアインターフェース192上における同時データフローが必要となる。このデータフローの並行処理は、任意の単一の瞬間において特定のメディアポートインターフェース192上においてデータ値が1つだけ存在し得るという事実に起因する時間的制約があるため、上記した(a)および(b)の場合と異なる。複数のデータフローを維持するために、特定のメディアポートインターフェース192は経時的に共有されるように動作することができ、これにより、各活性コマンドがデータ転送を限られた時間にわたって行った後、当該活性コマンドからその他の活性コマンドのうち1つへと変化する。データ通信分野においてこのコンセプトを指す際に一般的に用いられる用語として、当業者によく知られている「時間領域多重化」がある。

10

【0031】

d. 単一のデータ単位(例えば、ブロック)を分割して1つよりも多くのメモリデバイス上に書き込む場合、複数のメディアポート176を用いて、単一の読み出しコマンドまたは書き込みコマンドを行うことができる。データの分割は、本開示の文脈において多様な方法で行うことができる。1つの分割として、1つのブロックをインターリーピングによって2つ以上に分割する方法がある。インターリーピングは、ビットまたはバイト境界に沿って行ってもよいし、あるいはより大きな単位を用いて行ってもよい(例えば、ワード、ダブルワード、セクター)。インターリーピングは、書込動作に対する符号化の前に行ってもよいし、後に行ってもよい。インターリーピングステップを符号化の前に行うかまたは後に行うかにより、デコーダの対応する順序が決定される。インターリーピングを符号化前に行う場合、別個の符号化ステップが各分割に対して行われる(つまり、この機能をサポートするために、分割数と同数のエンコーダが必要となる)。複数のメモリデバイス上において単一のブロックを書き込む/読み出すインターリーピング機能の1つの利点として、インターリーピングを用いないアプローチと比較してデータスループットを大幅に高くすることが可能である点がある。

20

30

【0032】

モジュール30は、1つ以上のデコーダ202を含む。データ経路リソースとして、本実施形態において、各デコーダ202は、一度に1つの転送(例えば、データブロックまたはデータ分割の転送)しかサポートできないため、複数の異なる転送を同時復号化する場合、図3に示すようなモジュール実施形態に記載の複数のデコーダが必要となる。複数のデコーダを設けることにより、2つの異なる転送を真に同時復号化する能力が得られる。

【0033】

デコーダの構成は、メモリデバイスの使用方法によって異なり得るため、モジュール中において複数の改変例がある。一例として、特定のモジュールのメモリデバイスがフラッシュ技術に基づく場合、異なる種類のフラッシュメモリを用いることができる。さらに、複数の種類の利用可能なフラッシュ内において、各フラッシュ種類について1つ以上の使用モードがあり得る。例えば、市販のフラッシュデバイスが選択された場合、メモリデバイス(またはメモリデバイスの一部)をシングルビットパーセル素子(SBC)モードまたはマルチプルビットパーセル(MBC)モードにおいて用いることができる。実施形態において、代数エラー訂正符号(「代数ECC」)で符号化されたデータを(使用モードに関係無く)書き込むことができる。よって、この場合における適切なデコーダは、適用された特定のコードを復号化し、特定のコードによって許容される範囲までエラーを訂正するデコーダである。別の場合として、SBCまたはMBCメモリのためのモジュールに

40

50

よって用いられる特定の代数コードは、SBCまたはMBCメモリがメモリデバイスのページ構造に直接対応する単位でアクセスされるかまたはホストのビュー（および管理責任）からのページレベル詳細を除去するホストの代わりにモジュールが抽象化サービスを行うかにさらに依存し得る。いずれの場合も、異なるデコーダ実施形態が用いられ得る。

【0034】

別の例として、読み出し、書き込みおよび消去に関連するメモリデバイスの内部機能について可視性、アクセスおよび制御を提供するメモリデバイスが用いられる。このようなメモリデバイスによって構成されるモジュールは、先ほど述べた例において典型的に用いられる標準的メモリよりもずっと高い精度で書き込みおよび読み出しが可能である。より高精度が可能となることにより、情報記憶密度が向上するだけでなく、復号化機能において有利に用いることが可能なさらなる情報も得ることができる。このさらなる情報は、フラッシュメモリに書き込まれたデータおよびフラッシュメモリから読み出されたデータ中においてさらなるビット精度の様態で存在し得、「ソフトビット」と呼ばれ得る。「ソフトビット」とは、データ通信分野の当業者にとって公知の用語である。また、このようなメモリにより、データ中における読み出しまたは書き込みに関連するゆがみを補償する機能も得られ、これにより、読み出し機能および書き込み機能の構成（例えば、適合および／または訂正）の能力が得られ、これにより、（セルあたりのより多ビットの）より高いデータ記憶密度と、（フラッシュメモリ中のプログラム／消去サイクル増加として現れることが多い）より高い信頼性とのうちいずれかまたは双方を提供することができる。このような能力には、異なる構成のデコーダ（単数または複数）が必要になり得る。

【0035】

これらの例を鑑みて、モジュールの実施形態の1つの基準が異なる種類のメモリデバイスの利用についての柔軟性を提供することである場合、モジュール中の対応する復号化配置構成により、マルチモード機能が得られる。予測され得る特定の変動を以下に挙げる。

- a) 1つ以上のビットエラー容量設定を用いた BCH（ボースチャウドゥーリーオッカングム）復号化のみ、
- b) もう1つの符号エラー容量設定を用いた RS（リードソロモン）復号化のみ、
- c) 適切な形態の逐次復号化（例えば、畳み込み）と組み合わせられた BCH 復号化、および
- d) 適切な形態の逐次復号化と組み合わせられた RS 復号化

1つよりも多くのデコーダ機能を提供するように構成可能なモジュールは、デコーダの特定のインスタンスを特定のモードで動作させることができる。例えば、2デコーダモジュールは、2つのデータ経路上の真の復号化並行処理をサポートすることができる。モジュール中の複数のデコーダによって提供される柔軟性の一部は、所与の時間において非限定的な例として1つのデコーダが BCH オンリーデコーダとして構成することができる一方、その他のデコーダが畳み込み（ビタビ検出器）デコーダおよびリードソロモンデコーダの組み合わせとして構成することができる点に起因する。

【0036】

図3のエンコーダ220について、直前の議論は、少なくともエンコーダおよび符号化機能に一般的に適用される。なぜならば、これら2つの機能は直接的に類似するからである。そのため、メモリ利用、デコーダモード、柔軟性および並行処理についての議論は、符号化の場合にも等しく適用される。

【0037】

読み出し論理212は、記憶サブシステムとしてのモジュールの少なくとも半自律能力の動作サポートとして提供され得る。読み出し論理212によって行われる機能を例示目的のために挙げると、較正および／または補償機能により内部動作に対する特定の制御を行うメモリデバイスについての、較正または補償（例えば、平均値およびアレイプロセッサ機能（共有されている同時係属中の米国特許出願シリアル番号第12／888,585号（出願日：2010年9月23日、名称：「MEMORY QUALITY MONITOR BASED COMPENSATION METHOD AND APPARA

T U S」(本明細書中以下「585出願と呼ぶ」)中に記載)と共に用いられるデータ処理がある。詳細には、補償においては、メモリデバイスの物理的配置構成と関連付けられたパラメータを用いた信号処理が行われ、較正においては、計量収集分析機能が行われ得る。計量収集分析機能は、直接ソフトウェア制御下のハードウェア機能として行われ得る。

【0038】

書き込み論理222について、読み出し論理212および読み出し機能についての議論は、書き込み論理および書き込み機能にも概して適用され得る。いくつかの場合において、読み出しモードの代わりに書き込みモードにおける補償または適応と関連付けられた動作を行う特定の理由または利点があり得る。このような適応について、情報(例えば、信号対ノイズ比(「SNR」)関連データ)を後続パラメータ値のために収集することができる。ビット密度構成の例および非限定的な例において、モジュールは、ガーベジコレクション機能によってリクレームされたメモリの特定の物理的領域が再割当される際に当該領域のビット密度を構成する際、当該物理的領域に対し、SNR統計値を用いることができる。これは、書き込み機能と読み出し機能との間の主な差であり得る。モジュールは、読み出し機能および書き込み機能の任意の適切な組み合わせを含み得る。よって、所与のモジュールにおいて、構成可能な書き込み機能はあり得るが構成可能な読み出し機能は無く、あるいは、構成可能な読み出し機能はあり得るが構成可能な書き込み機能は無い。いくつかの実施形態において、構成可能な読み出し機能および書き込み機能をどちらも提供することができる。

【0039】

データバッファ部210からデータ経路および制御部190への直接書き込みデータ経路230により、必要なときに低遅延データ通信を提供することができ、復号化機能または読み出し機能が不要な転送のための別のデータ転送経路も提供することができる。この直接書き込み経路は、メディアポート176および関連付けられたメモリデバイス170のうち1つへとデータ転送が行われる点において、書き込み経路である。

【0040】

直接読み出しデータ経路204は、直接書き込み経路230に類似し、例えば必要な時に低遅延データ通信と、符号化機能または書き込み機能が不要な転送のための別のデータ転送経路とを提供するために、データ経路および制御部190からデータバッファ部210へと延びる。直接読み出し経路は、メディアポート176および関連付けられたメモリデバイス170のうち1つからデータ転送が行われる点において、読み出し経路である。

【0041】

データバッファ部210は、コントローラポート240へと接続される。コントローラポート240は、モジュールインターフェース42の1つのインスタンスを提供する(図1および図2を参照)。データバッファ部210は、データバッファ機能を実行する。このデータバッファ機能により、データバッファとメディアポートとの間およびデータバッファとモジュールインターフェースとの間のデータ中継が可能になる。このようなデータバッファ機能により、例えば、バッファ内への転送とバッファからの転送との間の速度マッチングが可能になる。例えば、任意のフラッシュ読み出しコマンドにおいては、モジュールインターフェースへのメモリデバイスデータの転送が必要になり得る。この転送は、以下の組み合わせとなる：(1)メモリデバイス170から発生しかつバッファ210において終了する別個の転送、および(2)バッファ210から発生しかつモジュールインターフェース42中を流れる後続転送。逆に、メモリデバイスへの書き込みを含む任意の機能は、以下の組み合わせを含む：(1)モジュールインターフェース42を通じて送られかつデータバッファ210において終了する別個の転送、および(2)データバッファ210から発生しかつメモリデバイス170において終了する後続転送。

【0042】

データバッファ部210は、DMA部242を含み得る点が理解されるべきである。DMA部242は、データバッファ内外におけるデータフローを制御するように構成される

。一般的に、DMAリソースは、開始オフセット、カウントによって構成され、優先順位付け機能が各可能なデータ経路へと割り当てられる。

【0043】

続けて図3を参照して、各モジュール30は、埋設マイクロプロセッサ300を含む。埋設マイクロプロセッサ300は、プログラムメモリ302から動作する。マイクロプロセッサは、マイクロプロセッサバスを介して接続される。マイクロプロセッサバスは存在していることが理解されるが、モジュールを構成する多様なコンポーネントの例示を明確にする目的のために図示していない。詳細には、マイクロプロセッサは、多様なコンポーネントレジスタにアクセスすることができる。プログラムメモリ302中に保存されているソフトウェアは、特定の実施形態に基づき得る。当業者であれば、所与の機能に対するハードウェアサポートの範囲は異なり得ることを認識する。よって、管理機能を実行する際のソフトウェアとハードウェアとの間の協働は、以下の変動に応答するように構成することができる。

- a. メモリデバイス種類。
- b. 符号化機能および復号化機能（例えば、アルゴリズム）。
- c. 補償機能（読み出しデータ上の補償およびデータ書き込み前の補償に適用される）。
- d. フラッシュメモリのためのブロック消去機能。
- e. モジュールインターフェースサポート（コマンド実行）機能。
- f. 自律性ウェアレベリング機能（モジュールが、主要コントローラから独立した自立型機能としてウェアレベリングを実行する）。
- g. 半自律ウェアレベリング機能（主要コントローラが、関連パラメータおよびコマンドをモジュールインターフェースを介してモジュールへと提供することにより、ウェアレベリング機能を制御する）。
- h. ローカル自律性フラッシュ変換層（FTL）の実装。
- i. 指向性フラッシュ変換層の（ホストによって制御された）実装（主要コントローラが、関連パラメータおよびコマンドをモジュールインターフェースを介してモジュールへと提供することにより、フラッシュ変換動作を制御する）。
- j. ホストによって指示される物理的メモリアクセス。ホストは、主要コントローラに適切なコマンドを発行することにより、複数のモジュール中の物理的メモリへアクセスすることができ、主要コントローラは、選択されたモジュールへ取り付けられたメモリデバイスのうち1つに適切な直接アクセスコマンドを発行することにより、このようなコマンドに応答することができる。
- k. ホストに対する論理ブロック抽象化により、モジュールによって指示された物理的メモリアクセス。
- l. 自律性ガージコレクション（通常はローカルFTLに適用される）。
- m. 半自律ガージコレクション（主要コントローラは、この制御の適用先であるモジュールに適切なパラメータを書き込むことにより、主要コントローラに取り付けられたモジュールのうち1つ以上におけるガージコレクション機能を制御する）。
- n. 指向性ガージコレクションのサポート（例えば、内部ブロックコピー機能）（主要コントローラは、各モジュールのモジュールインターフェースを介して複数のモジュールのうち少なくとも一部に適切なパラメータを書き込むことにより、ガージコレクション機能を制御する）。
- o. 取り付けられたメモリデバイスのローカルセルフテスト
- p. 自律性電力管理機能

リストは例示目的のためのものであり、包括的または限定的なものを意図していないことが理解されるべきである。よって、モジュール構成は、少なくともモジュール動作の前述の態様の選択および再構成を任意の適切な組み合わせで支援する。

【0044】

再度図3を参照して、モジュール30は、機能エンジン320を含む。機能エンジン320を用いて、CPU300およびプログラムメモリ302を通じてリソースとして提供

される機能の任意の適切な組み合わせを実行（例えば、表現）することができる。機能エンジン320の実施形態は、例えば、前述の多様な管理機能のうち1つ以上を実行することができる。例えば、ウェアレベリング、FTL、およびガベージコレクションおよび所望および／または必要とされ得る他の機能を実施形態において提供することができる。さらに、機能エンジンは、コントローラ40からかつ／または機能を構成するモジュールそのものから提供された1組のパラメータを受信し得る。すなわち、モジュールの管理機能は、コントローラまたはモジュールそのものから提供されたパラメータに基づいて少なくとも一定範囲において構成することが可能である。例えば、コントローラによって管理機能が指示される範囲は、コントローラによって設定することが可能なパラメータに基づいて構成され得る。しかし、パラメータが確立された後、管理機能は、実施形態において自律的に動作することができる。本明細書中以下、パラメータ優先順位について詳述する。しかし、いずれの場合においても、機能エンジンは、以下にさらに説明するような対象機能（単数または複数）を行うように構成される。主要コントローラ40は、コントローラから機能エンジンの共有パラメータ部322へと提供される1組の構成パラメータを通じて、所与の機能についての動作特性を決定することができる。例えばCPU300がコントローラ40（図2）およびモジュール30（図3）のモジュールインターフェース42に応答することにより、共有パラメータ部を設けることができる。これらの構成パラメータにより、機能エンジンがどのように各機能（例えば、ウェアレベリング、ガベージコレクションおよびビット密度構成）を実行（例えば、初期化および／または作動）するかが決定される。各機能についてパラメータによって規定された構成により、モジュールおよび主要コントローラ双方によって実行されるべき活動についての要求が規定され得る。例えば、少なくとも半自律ウェアレベリングがモジュールによって実行され得、主要コントローラからの最少量の関連通信が元々の構成に追従する。一方、指向性ウェアレベリングの場合、コントローラからの連続的構成または定期的構成の後、モジュールからの何らかの形態の報告を介してウェアレベリング活動の監視をコントローラによって行う必要があり得る。このような報告は、モジュールによって達成することができる（例えば、（後述する）報告部を介して提供される出力パラメータ値を更新することにより、達成することができる）。そのため、各モジュールの機能エンジンを、1つ以上のモジュール入力パラメータに少なくとも部分的に基づいて、当該モジュールのみの不揮発性メモリデバイスのための管理機能を行うように構成することができる。主要コントローラは、ホストデバイスとのデジタルデータ通信を提供し、複数のモジュールの各モジュールとのモジュールデジタルデータ通信を提供し、これにより、モジュールのうち1つの内外において流れるデータ全てがコントローラを通じて流れる。さらに、コントローラは、入力パラメータを各モジュールへと提供するように構成され得、これにより、各対象機能の少なくとも半自律実行を構成することができる。

【0045】

モジュールインターフェース42からデータバッファ210への入力データ経路330は、モジュールインターフェースからのデータおよび制御情報をバッファ部中へと転送する。このデータ経路は、比較的高速のものであり得る（例えば、バッファ部210とデータ経路および制御部190との間に延びる少なくともいくつかの他のデータ経路よりも比較的高速のものであり得る）。なぜならば、入来経路330は、データバッファ部210からの複数のデータ経路をデータ経路および制御部190へと提供できるからである。

【0046】

出力データ経路340は、データバッファ部210からコントローラポート240へと延びて、データおよび制御情報をバッファ部からモジュールインターフェースへと転送し、モジュール30からSSDコントローラ40への出力データ経路の一部をコントローラポート240によって提供されるモジュールインターフェース42を介して形成する。データ経路340は、バッファ部210の内外の他のデータ経路よりも高い帯域幅を持ち得る。なぜならば、データ経路330と同様に、複数の他のデータ経路が出力データ経路340によって提供され得るからである。

【0047】

モジュールインターフェース42の実施形態の詳細を図4のブロック図に示す。インターフェース接続が、図2のコントローラ40のモジュールポート100と、図3に示すモジュール30のコントローラポート240との間に延びる。コントローラポートおよびモジュールポートにより、各モジュールインターフェース42を構成する各モジュール30およびコントローラ40間にコマンドおよびデータ接続が提供される。データ伝送経路400により、コントローラからのコマンド、データ、状態および他の情報をモジュールへと伝送するための通信経路が得られる。もちろん、これらのアイテムは、ホストデバイス20（図1）から発生し得る。実施形態において、データ伝送経路は差動信号生成によって確立され得、これにより、2つの物理的信号によってデータ伝送経路が構成され得る。データ伝送経路は、双方向信号対の形態で任意選択的に具現化することができ、これにより、逆方向の動作が、（後述する）データ受信経路のデータ転送能力の性能向上として機能することができる。

10

【0048】

データ受信経路402は、モジュール30からのデータ、状態および他の情報をコントローラ40へと伝送するための通信経路を提供する。データ伝送経路400と同様に、経路402は、実施形態において差動信号生成によって確立することができ、これにより、2つの物理的信号がデータ受信経路を構成する。実施形態において、データ受信経路は、双方向信号対として実装することができ、これにより、逆方向の動作で動作する場合、データ受信経路のデータ転送容量の性能向上として機能することができる。

20

【0049】

信号差動対により、モジュールインターフェース42上にクロック410が提供され得る。クロック対上を指向するクロック信号は、ホスト中において生成され得、データ送信および受信の同期する際にもホストによって用いられ得る。モジュールインターフェース上に提供された差動クロック信号を用いて、モジュール中におけるデータ送信および受信を同期させることができる。モジュールによって受信されたクロックは、ホストによって生成および利用されるクロックに直接関連する。送信遅延および他の回路効果に起因して、モジュールによって受信されたクロック信号は、ホストによって用いられる信号から位相がずれる場合があるため、ホストおよびモジュール双方内のサンプリング回路をクロックアライメント回路（例えば、DLLまたはPLL）によって支援することができる。クロックアライメント回路は、クロック信号とデータ信号との間の最適アライメントを達成できるように、動的に構成（例えば、適合）され得る。

30

【0050】

開示のインターフェースを用いれば、SSDコントローラに必要なピン数を大幅に減少させることができる。各メモリモジュールは8個のフラッシュチップを支持することが可能であるため、ピンカウントが少ないモジュールインターフェースを1個介してモジュールからSSDインターフェースへの接続を行うことができる。このようにすることにより、各メモリデバイスが直接コントローラへと接続された従来技術の実装と比較して、システム中の全フラッシュデバイスを制御するために必要なピン数を低減することが可能になる。図1および図4に示す実施形態を用いることにより、コントローラからモジュールへの接続に必要なピンを限定することにより、ピン数を低減することができる。すなわち、図4のピンカウントが低減したモジュールインターフェースを用いることにより、従来技術インターフェース（例えば、ONFI）と比較して、コントローラへの接続をより少数のピンにより行うことが可能になる。このようにして、SSDコントローラは、SSDアプリケーションに適したトランザクションレートおよびデータスループットレートを達成しつつ、（代替例（例えば、ONFI）と比較して）コントローラ上の少数のピンで多数のモジュールへの接続および多数のモジュールの制御を行うことができる。本明細書中に記載のモジュールインターフェースにより、高レベルSSD機能から低レベルフラッシュ機能を抽象化することが可能になる。よって、コントローラを例えば高レベルのコマンド実行局面のみで構成することにより、コントローラの負荷を低減することができる。低レ

40

50

ベル機能（例えば、物理的フラッシュ特性に関連する機能）は、モジュールレベルにおいて実行することができる。より多数の取り付けられたメモリデバイスにより、少なくともメモリデバイス管理のための半自律機能の利用を通じて、SSD性能を向上させることができることが理解されるべきである。単一のハードウェア構成（事前構成されたモジュールおよび特定のSSDコントローラの特定の組み合わせ）内において、モジュールを少なくとも半自律性のエンティティとするようにSSDを構成することができ、コントローラの役割は、モジュール中の論理アドレス指定されたデータ単位にアクセスすることである。このようなシステムにおいて、管理についてのコントローラの主な役割は、各モジュールによって提供される（モジュール単位の集約としての）記憶を単一のより大きな容量で管理することである。

10

【0051】

モジュールインターフェースは、意図するアプリケーションをSSD利用に合わせてカスタマイズするための、コマンドおよびプロトコルレベルにおける機能および能力を含み得る。これらの機能および能力を上げると、低遅延読み出し機能および書き込み機能の支援、高データスループットの支援、コマンド待ち行列および同時動作コマンドの支援、多ブロック転送動作を含むデータブロックの、非順次転送の支援、異なるレベルのブロック抽象化におけるコマンドの支援、および簡単かつ高度な高性能プロトコルがあり、これらそれぞれについて、本明細書中以下に説明していく。

【0052】

低遅延読み出し機能および書き込み機能の支援

20

コントローラ40から受信されたコマンドに対する最も高速の可能な応答を支援することにより、低遅延がモジュール30によって支援される。この支援は、以下の少なくとも2つのアプローチを用いた実施形態を用いて、モジュールインターフェース42によって達成される：第1のアプローチは、同時実行コマンドを支援することであり、第2のアプローチは、短データパケットを任意選択することである。第1のアプローチを通じて、コマンドをコントローラ40からモジュール30へと発行することができ、1つ以上の他のコマンドが既に実行している場合であっても、すぐに作用することができる。その結果、（コマンドが同一メモリデバイスに既にアクセスしていない場合でも）コマンドが標的メモリデバイスへとアクセスすることができ、（現在活性状態のコマンドを完了させた後に別のコマンド実行を開始する場合よりも）コントローラ40とモジュール中のメモリデバイス170との間のデータ転送をより迅速に開始することが可能になる。コマンド待ち行列および同時コマンド実行のためのインターフェース支持により、このような同時実行能力が可能になる。第2のアプローチを通じて、読み出しコマンドまたは書き込みコマンドのデータ転送位相をより迅速に開始することができるため、（より長いパケットを用いた場合よりも）より迅速に完了することが可能になる。第2のアプローチと、短いパケットではなく長いパケットを用いたアプローチとの間の違いとして、より短いパケットを用いることにより、待ち時間を短くすることができる点がある。なぜならば、長パケットと比較して、短パケットは転送時間を短縮できるため、同時に実行されている転送全てに対し、低遅延ベースでバッファ帯域幅が割り当てられるからである。所与のコマンドに対し、短データパケットにより、（より長いデータパケットが用いられる場合に長期化し得る待ち時間と比較して）インターフェースへのアクセス待ち時間がより短縮することができる。データパケット長さを構成することができる。しかし、特定の実施形態のためのデータパケット長さは、例えば、部分割当時間において用いられるパラメータによって決定され得る。このパラメータは、アクセス待ち時間をパケット長さに従って優先順位付けする。そのため、所与のモジュール構成について最大パケット長さをコントローラ40によって決定することができる。また、第2のアプローチを用いて、図4のデータ伝送経路400およびデータ受信経路402について説明した方向切り換え機能を用いることができる。選択された場合、方向切り換え機能により、全パケットサイズのより高速な送信が可能になり、これにより、スループットおよび待ち時間性能双方が増加する。

30

40

【0053】

50

高データスループットの支援

高スループットのためにより大型のパケットが必要な場合、高スループットおよび低遅延は時に相反する目的となる。短パケットサイズを用いれば待ち時間性能が向上するが、スループット性能の劣化も招く。なぜならば、インターフェースの固有オーバーヘッドは、大型パケットの場合よりも小型データパケットの場合によって比例的に増大するからである。すなわち、長パケットの場合よりも短パケットの場合の方が転送効率が低下する。そのため、インターフェースに対する要求がインターフェースの帯域幅容量に近づいた場合、短パケットの場合よりも長パケットの場合の方がデータスループットが向上する。構成選択肢として、パケットサイズを選択できる能力も、高スループットおよび低遅延の競合する利益に対処するための、高データスループット支持のための局面である。

10

【0054】

コマンド待ち行列および同時動作コマンドの支援

実施形態において、コマンド待ち行列機能は、主にハードウェア機能として、モジュールインターフェース42中に存在し得る。各実行コマンドは、コマンド実行に関係する全トランザクション（例えば、コマンド転送、データ転送、およびハンドシェーキングトランザクションの状態）を支援する必要がある。これらのトランザクションは、施設（例えば、手続き中のDMAリソース）を用いることができる。このようなハードウェアリソースは、コマンド待ち行列機能をイネーブルするために必要な範囲において、モジュールインターフェース定義中に設けることができる。コマンド待ち行列のさらなる機能として、キュータグの生成および確認がある。キュータグとは短い2値であり、コマンドに必要な前述のハードウェアリソースをアドレス指定するために用いられる整数である。キュータグは、ホストからデバイスへと送られる各新規コマンドコードと共に送信され、コマンドが実行し続けている限り、活性状態のままである。コマンドが終了した後、キュータグは活性状態ではなくなり、後続コマンドが発行されたときにキュータグを再度割り当てることが可能になる。

20

【0055】

多ブロック転送動作を含むデータブロックの非順次転送支援

モジュールインターフェースプロトコルは、長い多ブロック転送内におけるブロックの非順次送信を可能にするコントローラおよびモジュール間のトランザクションモデルを支援することができる。トランザクションモデルにより、任意の読み出しまたは書き込みコマンド中に1つ以上の「接続」を含めることが可能になる。各接続は仮想構成概念であり、より長い転送の一部についてアドレス指定制御およびカウント制御を割り当てる。そのため、接続は時間エンティティとして存在し、コマンドの一部（通常はデータ転送）を維持するために存在し、存在において時間的制約があるため終了する必要がある。時間的制約は、活性状態になり得る他のコマンドに対してインターフェースが必ず利用できるようにするためのものである。この制約は、固定してもよいし、あるいは変化してもよい。読み出しまたは書き込み転送においては、単一の接続のみが必要である場合もあるし、あるいは多数の接続が必要な場合もある。そして、コマンドに必要な接続数は、読み出しまたは書き込みコマンドの合計長さ（単位：バイト）の関数であり、また、時間的制約のサイズの関数である。この接続数の決定は、モジュール中において実行しているコマンド取り扱いソフトウェアによって行われ得る。複数の接続を一度に確立することができるため、所与の時間において所与のコマンド内に1つの接続のみが存在するか、または、所与のコマンドが複数の活性接続を持っていてもよく、各接続からコマンドの合計転送長さの異なる部分が転送される。（ブロックの）非順次転送は、例えば以下のようにして支援することができる。すなわち、コマンドによって転送されているブロックの全範囲内のオフセットと比較して非順次ブロックまたはいくつかのブロックを転送する接続を確立する。

30

40

【0056】

コマンド全体の実行の一部として、各接続を確立することができる。接続は、割り当てられたデータ長さ（これは、（コマンドに相対して規定された）ブロックであり得る）を転送できるだけの十分な長さで存在するか、または、コマンドコード中のカウントフィー

50

ルドによって規定されたバイト数として存在する。規定長さのデータが転送された後、接続が終了する。

【0057】

異なるレベルのブロック抽象化におけるコマンドの支援

コントローラ40は、モジュール30へ取り付けられたメモリデバイスへアクセスするために異なるアプローチを用いることができる。変動がある場合、本明細書中以下に記載するような2つの実施形態が一般的である。

【0058】

第1の実施形態において、コントローラは、標準的メディアとしてモジュールに取り付けられたメモリデバイスへモジュールインターフェース42を介してアクセスする。標準的メディアでは、メモリデバイスが階層状に配置されたメモリを含み、その階層の最低レベルにページが位置し、その階層のより高レベルに消去ブロックが位置する。読み出しまたは書き込みアクセスは、一定量のデータを有するページを有する特定のページを標的とし、ECCパリティに通常用いられるさらなるビットを含み得る。別の種類のアクセスとして消去があり、消去ブロックを標的とする。フラッシュメモリ中の消去ブロックは、ページよりもずっと大きな構造である。実際、消去ブロックは、整数のページをいくつか含み、整数は変動し得る。典型的な値は、消去ブロックあたり128ページである。ページは一般的には単位で書き込む必要があり、消去は一般的には消去ブロック全体に対して行う必要がある。メモリデバイスからのページ読み出しは全体的に行ってもあるいは部分的に行ってもよいが、ページを部分的に読み出した場合、ECC符号化機能および復号化機能に影響が出る場合がある。いくつかのECC符号化機能においては、ページ全体を符号化することができるため、ECC復号化を行うために、ページ全体を読み出す必要が出てくる。他のECC符号化機能も実装可能であり、符号化および復号化をページ中のデータ区画上に行う。この場合、部分的ページ読み出しにより、これらの区画のうち1つ以上を読み出すことができ、ECC復号化を行うことができる。ページを全体的または部分的に読み出すかに関係無くまたページ全体に書き込むかに関係無く、物理アドレスを用いて、物理ページのアドレス指定を行う。物理アドレスを用いるためには、標的モジュールへ取り付けられた各メモリデバイス中のページの物理的配置構成について、コマンドの発行者（この場合はコントローラ40）において知識が必要になる。

【0059】

第1の実施形態は、モジュール中の符号化、復号化およびデータ経路ハードウェアによって支援される。アドレス指定、読み出し、書き込みならびにECC符号化および復号化は全て、選択可能な構成としてそしてこれまでみられなかった状態で、モジュール中において支援される。

【0060】

第2の実施形態において、メモリデバイスが論理単位として、アドレス指定およびアクセスされる。これらの論理単位は、（物理アドレスではなく）論理アドレスを用いた論理セクターまたは論理ブロックを含む。このアプローチにおいては、データ長さはページの物理的長さによる制約を受けないため、モジュールは、論理構造を物理的構造上にマッピングするように物理ページおよび消去ブロックを管理することができ、これにより、コントローラは、所望のデータを転送する際に、コマンドに論理アドレスを付加するだけですむ。この論理形態のアクセス（ブロック抽象化とも呼ばれる）をサポートするモジュール中の機能を挙げると、図3を参照して説明するようなエンコーダおよびデコーダ構成の多様な局面がある。このサポートの他の局面についても図3を参照して説明しており、例を挙げると、データ経路配置構成や、補償および較正に関連するその他の読み出し機能および書き込み機能がある。ソフトウェア機能および機能は、このサポートの重要部分である。なぜならば、ソフトウェアは、ブロック抽象化能力に関連する機能を支援するからである。これらの機能をいくつか挙げると、ローカルフラッシュ変換層（FTL）、消去動作（例えば、ガーベジコレクション）に対する制御、ウェアレベリング、ビット密度構成（例えば、エンコーダ／デコーダ構成）などがある。ブロック抽象化についてのさらなる詳

細について、図 5 を参照してさらに説明する。

【0061】

簡単かつ高性能なプロトコル

最も重要な機能のみを実行することにより、モジュールインターフェースは、可能なコマンドオーバーヘッドよりも低いコマンドオーバーヘッドで動作することができる。インターフェースはポイントツーポイント接続のみの形態をとるため、プロトコルは、多デバイスサポートの負担から逃れられる。ホストがデバイスとのトランザクションを開始、実行および監視する際、デバイスアドレスの生成、記録または利用は不要である。同様に、モジュールインターフェースによって実装されるポイントツーポイント物理的接続は、（コマンド実行の一部として既に述べた有限時間論理／仮想接続とは対照的に）動的に確立するのではなく、専用として用いられる。固有のモジュールインターフェースを介した各ホスト／デバイス接続の構成は製造時において確立されるため、ホストがデバイスをパワーオン時間に構成する必要がなくなる。

10

【0062】

加えて、インターフェースは、真の並行処理（同時独立転送）および時間領域多重並行処理の両方をサポートする。真の並行処理は、1つ以上の読み出しコマンドおよび1つ以上の書き込みコマンドが同時に実行される際に発生する。インターフェース並行処理は、読み出しコマンドおよび書き込みコマンド双方のデータ転送段階が発生する際に発生し、これにより、データ伝送経路400がコントローラ40からのデータをモジュール30へと活性転送するのと同時に、モジュールインターフェース（図7）のデータ受信経路402がモジュール30からのデータをコントローラ40へと活性転送する。この種類の並行処理は、前述の方向切り換え機能が関わっていない構成またはコマンドのみに対して発生する点に留意されたい。

20

【0063】

時分割多重化形態の並行処理は、モジュールインターフェースによって頻繁に達成することができる。これは、例えば、1つ以上の活性コマンドがモジュールインターフェース上におけるデータパケットを用いてデータ位相トランザクションを実行する接続を確立した場合に発生する。複数の活性接続を確立することができるが、単一のデータ送信またはデータ受信信号対に沿った任意の瞬間において、最大でも1つのパケットを送信することができる。時間領域並行処理が発生するのは、活性接続によってデータパケットが交互に転送されるからであり、1つの接続がデータパケットを1つの瞬間において転送し、別の接続が別の瞬間においてデータパケットを転送する。このようにして、複数のコマンドがデータパケットを同時に転送することができるが、このようなことは、物理的転送接続へのアクセスを共有することによって行われる。例えば、図4に示すように、P1、P2およびP3のパケットをデータ受信経路402を介して順次転送し、各パケットは、データ転送全体の異なる部分を形成する。同様の転送は、データ伝送経路400を介してサポートすることもできる。実施形態において、このようなパケット転送は、データ伝送経路およびデータ受信経路上において同時に発生し得る。

30

【0064】

コントローラによって異なるモジュールに対する異なる数の管理が可能なように、SSDを構成することができる。このような実施形態は、モジュールに対する混合モードアプローチと呼ばれる場合があり、機能（例えば、ウェアレベリング、ガーベジコレクションおよびビット密度構成）の実行についてのコントローラとの少なくとも何らかの相互作用によってモジュールの一部を管理することができ、モジュールの別の部分は、これらの機能の詳細管理について自律的に動作することができる。

40

【0065】

ビット密度構成について、各モジュールは、高密度動作（例えば、素子あたり2ビットを超える密度）が可能なメモリデバイスを含み得、実施形態において、各モジュールは、モジュール中に含まれる全メモリデバイス上において高密度モードで動作するかまたは低密度モード（例えば、素子あたり2ビット以下）で動作するように構成することができる。

50

ように、コントローラによって構成することができる。いくつかの実施形態において、いくつかのメモリデバイスを高密度モードで（例えば、素子あたり2ビットを超える密度で密度で）動作させかつ他のメモリデバイスを低密度モードで（例えば、素子あたり2ビット以下で）動作させるように、モジュールをコントローラによって構成することができる。モジュールが低密度動作（例えば、素子あたり1または2ビット）が可能なメモリデバイスを含む場合、低密度動作に従った符号化および復号化構成で動作するように、このモジュールをコントローラによって構成することができる。この選択のための構成パラメータは、各ECC符号化および復号化単位に適用されるデータ長さと、各ECC符号化および復号化単位と共に含まれるパリティビット数とを含み得る。さらに、低密度メモリが素子毎に1ビットまたは2ビットとして構成される場合、モジュールを選択された低密度モードを用いるようにコントローラによって構成することができる。

10

【0066】

上述したように、少なくとも半自律性のメモリモジュールは、多様な機能を独立的に行うことができる。ウェアレベリング機能をさらに考慮すると、モジュールに提供されたパラメータに基づいて、各メモリモジュールがその固有の1組のメモリチップ内においてウェアレベリングを行う機能を担うことができることが理解されるべきである。分散型ウェアレベリング制御により、コントローラに各個々のメモリチップ上の摩耗を、監視および調節させる負担がほとんどなくなる。

【0067】

非限定的な例として、本開示によるモジュールにより、他の機能を少なくとも半自律的に取り扱うことができる。例えば、各メモリモジュールは、その固有の電力消費を管理することができる。メモリモジュールが不活性であるかまたは軽負荷を受けている場合、メモリモジュール上の電力管理回路を活性化することで、電力消費をモジュール毎に低減することができる。その結果、コントローラオーバーヘッドが各メモリモジュールの電力状態を監視および制御する必要無く、分散型電力管理が可能となる。

20

【0068】

別の例として、各メモリモジュールは、その固有の1組のメモリチップのデータの監視および更新を行うことができる。その結果、コントローラが各メモリチップ中のデータ品質を監視する負担が無くなる。

【0069】

さらに別の例として、各メモリモジュールは、その固有のNVMチップのための誤差修正を全て取り扱うことができる。このような分散型誤差修正処理を用いることにより、従来のSSDの場合のように誤差修正全てをコントローラが担当する場合に達成可能なスループットよりも、スループットを大幅に高くすることが可能になる。誤差修正処理はSSD内において分散させることが可能であるため、適用可能な誤差修正量を従来のSSDよりもずっと高くすることが可能になり、その結果、いくつかの実施形態の個々のメモリチップ内のデータ密度を高くすることが可能になる。

30

【0070】

本明細書中に記載されるメモリモジュールの分散型処理電力に起因して、このようなモジュールを含むSSDのスループットを、従来のSSDアーキテクチャによって達成されるスループットの数倍だけ増加させることができる。機能（例えば、誤差修正、データ更新および低レベルウェアレベリング）により、SSDコントローラには負担がかからなくなる（かまたは少なくとも低減された負担を有する）。各メモリモジュールは、（例えば、メモリモジュール毎に少なくとも8NVMダイまで）制御することができるため、高データレートを取り扱うことができる。コントローラからの各バス上に複数のメモリモジュールを設けることが可能であるため、高データレートを倍加させることが可能になる。コントローラとメモリモジュールとの間に並列に延びるいくつかのバスを設けることが可能であるため、これを倍加させることができる。全体的な結果として、SSDにより、既存のSSDまたはハードドライブのデータレート（スループット）を複数倍にすることができる。

40

50

【0071】

モジュールインターフェースにより、少なくともコントローラが担う代わりに、少なくとも半自律的に各モジュールによって分散的に行われる低レベル機能に関連する大量のデータを搬送する必要性からコントローラインターフェースに対するモジュールが逃れることができるという理由のため、従来技術を用いた場合よりも高い性能が可能になる。そのため、これらの低レベル機能の負担無く、コントローラが自由に動作することができる。その結果、SSDコントローラが高レベルSSD機能の実行をより簡潔かつより効率的に行うことが可能になる。このように処理デューティを分離することにより、より効率的かつ高性能SSDシステムが得られる。

【0072】

図5を図3と共に参照して、図5は、図3の1つのメモリモジュール30の複数のメモリデバイスを模式的に示すブロック図であり、例示目的のため、このモジュールNVMメモリをまとめて参照番号500によって示す。本例において、モジュールNVMメモリは、8個のメモリデバイスを有する。これら8個のメモリデバイスをNVM1～NVM8として示す。8個のメモリデバイスの利用は例示目的のためであり、限定的なものではなく、本明細書中の教示の範囲内において他のモジュールが異なる状態で配置された異なる数のメモリデバイスを含み得ることが理解されるべきである。よって、モジュールNVMメモリ500は、数個の部分から構成されるようにコントローラ40（図2）によって配置され得る。コントローラは、以下に示すような多様な基準のうちのいずれかに基づいて、複数のメモリデバイスを部分（セクション）に分割することができる。部分（セクション）を極めて柔軟に8個のメモリデバイス間に割り当てることができる。まず、いずれのモジュールも同一特性のNVMを含む場合であっても、1つのモジュールのメモリの部分を次の部分に割り当てることが完全に異なり得、物理的領域およびビット密度双方に依存しないことが理解されるべきである。第1の例として、第1のメモリ部（第1のメモリセクション）502は、メモリデバイスNVM1を1つだけ含む。第2の例として、第2のメモリ部（第2のメモリセクション）504は、メモリデバイスNVM2およびNVM4を含む。第3の例として、第3のメモリ部（第3のメモリセクション）506は、メモリデバイスNVM3の一部と、メモリデバイスNVM5の全体とを含む。第4の例として、第4のメモリ部（第4のメモリセクション）508は、メモリデバイスNVM8の一部を含む。これらの例の目的のため、部分（セクション）の外側に図示されているメモリデバイスの場所は割り当てられていないことが推定され得る。よって、任意の1つの部分は、個々のメモリデバイスの一部および／または1つ以上のメモリデバイス全体を任意の所望の組み合わせで含み得る。NANDフラッシュメモリデバイスを用いた実施形態において、部分のサイズ上の下界は、単一の消去ブロック中に保存可能なデータ範囲であり得る。よって、所与の1組のメモリデバイス中において利用可能な記憶空間をパラメータ（例えば、記憶容量、記憶保持、プログラム／消去サイクル、データスループット、またはこれらの何らかの組み合わせ）に基づいて最適化する目的のために、モジュールを構成しているメモリデバイスを分割することができる。メモリデバイスの分割は動的な能力であり得、必要に応じてメモリの特定の物理的部分の割当、割当解除および再割当を可能にして、メモリデバイスを寿命にわたって管理することが可能になる。部分（セクション）の利用を通じて、例えば、物理的特性またはメモリ種類に基づきかつ物理的範囲の固有の特性およびその摩耗履歴に従ったメモリデバイスの最適利用が可能になる。

【0073】

図5を参照して、ブロック抽出においては一般的には、ページ境界を跨ぐ（これをページラッピングと呼び得る）状態で、1つよりも多くのページの書き込みおよび読み出しが行われることが理解されるべきである。この点について非限定的な例として、NVM1は、物理的に隣接したページ510a～510dを含むものとして図示されている。読出動作または書込動作が、ページ510aの斜線表示された第2の半分と、ページ510bの斜線表示された第1の半分とを含む場合、読出動作または書込動作においては、ページ510aおよび510b全体が読み出されるかまたは書き込まれる。これらのページの斜線

10

20

30

40

50

表示領域は、例えば、図3のエンコーダ220およびデコーダ202によって用いられるコードワードを示し得る。そのため、ページラッピングにより、動作目的のためにページ間に物理的切れ目が無いかのような、ページ境界を跨ぐ符号化機能および復号化機能が可能となる。

【0074】

ここで図6を参照する。図6は、図3に示すようなモジュール30の機能エンジン320の実施形態を示すブロック図である。機能エンジンのコンポーネントおよびその相互接続が図示されている。図示の多様なコンポーネントの機能を達成する目的のために、ハードウェアおよびソフトウェアの任意の適切な組み合わせが利用可能であることが理解されるべきである。実施形態において、CPU300およびプログラムメモリ302は、機能エンジン320の動作を包含し得る。機能エンジン320の共通施設（例えば、領域）602は、1組のデータ構造を構成し得る。1組のデータ構造は、パラメータ（例えば、パラメータを含むデータテーブル）を含む。これらのパラメータは、例えばウェアレベリング機能610、ガーベジコレクション機能612、ビット密度構成機能614、およびFTL機能616によって共通的または共有的に利用または変更される。これらの機能はそれぞれ、割当タスクを少なくとも部分的に特定の入力パラメータに基づいて実行するように、構成される。共有パラメータ部322（図3）が共通施設602の一部を形成している様子が図示されており、よって、共有パラメータ部322（図3）を図3または図6を参照して言及する場合がある点に留意されたい。これらの入力パラメータを必要とする他に、各機能は、他のモジュールまたはSDDシステムおよびコントローラに対して全体的に自律的に動作することができる。共通施設602は、これらの機能の間において情報導管として機能する。なぜならば、場合によっては、多様な機能は、情報を共有し、協働して動作するからである。一般的に、各機能は、タスク完了のために、共通施設中のデータ構造からの最新情報にアクセスする。例えば、FTL機能616は、最新論理／物理アドレス変換を用いる。最新論理／物理アドレス変換は、データ単位の論理アドレスが付与されたデータ単位の物理的位置を返送する機能を提供するために、例えばテーブル領域620中に保存され得る。さらに、共通施設中に保持される情報の例について、本明細書中以下において適切なときに説明する。本実施形態において、共通施設は、報告部624をさらに含む。報告部624は、1つ以上の出力パラメータ（例えば、利用統計を示すもの）を報告するように構成される。これらの出力パラメータは、モジュールの不揮発性メモリに関連する。利用統計は、機能エンジンそのものおよび主要コントローラにとって利用可能であり、読出動作時に得られた1組の読出値に基づき得る。利用統計は、例示目的のため、1組のリードバック値に対するリードバック動作に基づいて生成され得る1組の読取値についての平均リードバック値と、その1組のリードバック値の標準偏差と、のうち少なくとも1つを含み得る。利用統計はまた、誤差修正統計値も含み得る（例えば、読出動作によってブロック誤差修正コードが復号化される場合における、累積誤差カウントおよびブロック毎の誤差カウント）。誤差修正統計値は、1つ以上のデコーダ202（図3）から生成され得ることが理解されるべきである。これら1つ以上のデコーダ202（図3）から、報告部が統計情報を入手する。利用統計に基づいて、機能エンジンは、利用統計によって示されるメモリの状態を鑑みて、パラメータを設定および／または選択することができる。例えば、利用統計が多数のプログラム／消去サイクルに基づいてメモリ劣化を示す場合、メモリのビット密度を低下させることができる。実施形態において、プログラム／消去サイクルの数は、閾値プログラム消去パラメータを超え得る。閾値プログラム消去パラメータは、利用されている特定の不揮発性メモリについて指定される。

【0075】

いくつかの実施形態において、利用統計に基づいたモジュールの不揮発性メモリに関連して健康指標を提供するように、モジュールの一部のモジュールの報告部を構成することができる。例えば、パラメータは、閾値を指定し得る。閾値において、機能エンジンは、以下のように応答する。すなわち、メモリの一部から離隔方向に既存のデータをコピーすることによりより高いビット密度で前回割り当てられたメモリの一部を回収し、その部分

10

20

30

40

50

を消去し、その後、その部分を相対的により低いビット密度へと再割当てして、当該部分へのさらなる書き込みがより低いビット密度となるようにする。この活動は、少なくともいくつかのコントローラパラメータ（例えば、メモリの進行中の摩耗に基づいて初期構成中に設けられるもの）に優先し得る。実施形態において、コントローラは、健康指標にアクセスおよび監視し得、パラメーター制御を提供して、上記のようにメモリの一部を回収するようモジュールに命令し得る。もちろん、これらの多様な活動を行うためには、以下において適切なときに詳述される、機能エンジンによって提供される機能の利用が必要である。

【0076】

上記を鑑みて、コントローラパラメータ（制御パラメータ）630は主要コントローラ40によって提供され、モジュールパラメータ632はモジュールそのものの内部から発生することが理解されるべきである。後者の非限定的な例を挙げると、デコーダ202（図3）によって生成される利用統計がある。これらのコントローラパラメータおよびモジュールパラメータは、共通施設602へと提供される。これらのパラメータは、主に4つの例示の機能そのものによって用いられ得、各機能の構成に影響を及ぼし得る。

【0077】

さらに図6を参照すると、ウェアレベリング機能610は、モジュール中の全メモリデバイス上におけるサイクル利用数を均一に最大化することに関する。当業者であれば、ウェアレベリング機能の構成についての基本的原理を理解し、このような機能が必要となる理由として、特定の形態の不揮発性メモリが限られた数のプログラム／消去サイクルに耐える際に限られた寿命を示す点があることを理解する。ウェアレベリング機能は、特定のモジュールのメモリデバイスの有効寿命を延ばすことに関する。

【0078】

ガーベジコレクション機能612は、不要になったデータが書き込まれたメモリ単位（部分（セクション）、ブロックまたはページ）を特定し、新規データの後続書き込みにおいて利用可能なメモリ単位のプールにこのメモリを返送する。ウェアレベリング機能と同様に、当業者であれば、ガーベジコレクション機能の構成の基礎となる原理を理解する。

【0079】

ビット密度構成機能614は、メモリ単位にビット密度構成を割り当てる。ビット密度構成は、メモリ中の各適用可能な物理的部分中の好適なビット密度構成について、基準（後述する）に基づいて決定された選択の結果決定される。実施形態において、メモリ単位は、ブロックによって構成された部分であり得る。

【0080】

F T L機能616は、特定のデータ単位（部分、ブロックまたはページ）の物理的位置を当該単位の論理アドレスのみに基づいて提供する。ウェアレベリング、ガーベジコレクション、およびビット密度構成は全て動的な機能であり得、進行する更新を発生させて、アドレス変換テーブルに影響を与えるため、F T Lおよび他の機能は、リアルタイムで相互に協働して、共通施設602中に最新状態を保持することができる。

【0081】

類似のパラメータがコントローラパラメータ630およびモジュールパラメータ632の一部を形成する場合、実施形態において、一般的にはコントローラパラメータが優先される。この点について、パラメータを任意の適切かつ柔軟な状態で編成することができる。例えば、いくつかの実施形態において、パラメータをサブセットとして編成することで、類似するパラメータからなる1つのサブセットについてはコントローラ40からの値が優先され、類似するパラメータからなる別のサブセットについてはモジュール30からの値が優先されるようにする。

【0082】

ここで図7を参照して、図6のウェアレベリング機能610の実施形態は一般的には、参照番号700によって示される。このウェアレベリング機能は、702において主要コントローラ40から利用可能なパラメータと、704においてモジュールから利用可能な

10

20

30

40

50

内部パラメータ（例えば、機能エンジンの共有パラメータ部 3 2 2（図 3 および図 6）中に含まれるもの）とに基づいて動作し得る。そのため、特定のパラメータ源の範囲内において、ウェアレベリング機能の性能において高い柔軟性が可能となる。パラメータ源に関係無く、ウェアレベリング機能は典型的には、これらのメモリデバイスが一定数のプログラムおよび消去動作の後に劣化する記憶素子を含む場合において、モジュールのメモリデバイスの動作寿命を最大化することに関連する。

【0083】

共有パラメータ部 3 2 2 に提供されるパラメータを選択することにより、主要コントローラ 4 0（図 2）は、ウェアレベリング機能に関して、ウェアレベリング機能についての制御の増減を行使することができる。上述したように、主要コントローラによって確立されたパラメータは、モジュールによって提供されたパラメータよりも優先される場合がある。コントローラおよびモジュールから提供されたパラメータは、初期化パラメータおよび動作パラメータによって特徴付けられ得る。これらの初期化パラメータは、7 0 8 において選択され、これらの動作パラメータは、7 1 0 において選択され、初期動作後の進行中のウェアレベリングに適用され得る。初期化パラメータの例を挙げると、ブロック合計と、ブロック合計のサブセットのうちデータ記憶のために初期に利用可能であるものによってモジュールのメモリ容量の初期範囲を示すアドレスがある。初期化パラメータはまた、パラメータ（例えば、部分（セクション）サイズ、部分（セクション）割当順序、ブロック割当順序）を含み得る。これらのパラメータは、シーケンスに対するディレクティブとして機能し、これらのパラメータに基づいて、ステップ 7 2 0、7 2 4 および 7 3 0 が部分（セクション）および当該部分内のブロックを初期メモリ配置構成に割り当てる。初期化パラメータ選択 7 0 8 は、プロセス全体の初期化分岐 7 1 2 の開始を含む点に留意されたい。これらの初期化パラメータは一般的には主要コントローラによって提供され、動作開始時においてウェアレベリング機能の構成を確立する。一般的には、モジュールの元々のウェアレベリング構成を確立するため、初期化を一度行うことができる。主要コントローラからの動作パラメータは、ウェアレベリング機能の非初期化局面に適用することができる。いくつかの実施形態において、モジュールによって提供される動作パラメータにより、主要コントローラからの初期パラメータに基づいた初期動作の後、進行中のウェアレベリング動作を定義することができる。ウェアレベリング機能の非初期化局面について設けられ得る種類のパラメータの例を挙げると、部分（セクション）およびブロック割当および再割当のための順序付けディレクティブがある。他の例を挙げると、再割当が必要となるまでにブロックまたは部分（セクション）中に静的データを保存しておく期間についての数値基準またはウェアレベリング再割当決定の判断基準として用いられる他の数値基準（例えば、累積書き込み／消去サイクル数、誤差修正統計値）がある。

【0084】

7 2 0 において、部分割当初期化を行ってモジュールのメモリデバイス間の元々の 1 組の部分の割り当てることにより、初期化分岐 7 1 2 を継続する。その後、ブロック割当機能 7 2 4 を行って、1 組のブロックを各部分中に割り当てる。部分割当の例を図 5 に示し、図 5 に関連して上述した。

【0085】

7 2 4 の後、初期構成が確立され、7 3 0 において例えば機能エンジン（図 3 および図 6）の共有パラメータ部 3 2 2 中に保存される。そのため、初期構成が完成し、ウェアレベリング機能の動作分岐 7 4 0 時において参照および更新のために利用可能となる。

【0086】

7 4 4 において、構成回収プロセスが行われる。初期化分岐 7 1 2 の後、7 3 0 において保存された構成によって確立された初期構成を回収することができる。その後、進行中の動作と共に、構成を動作分岐 7 4 0 によって管理することができる。進行中の動作時において、現在の F T L テーブル情報を F T L 更新経路 7 4 8 を介して（例えば、機能エンジンのテーブル領域 6 2 0（図 6）から）回収することができる。F T L 更新は、後述する書込および消去動作の結果発生する。書込および消去動作において、F T L 機能によっ

て物理アドレスが付加または削除されると、ウェアレベリング機能への影響が必ず発生する。F T L機能を更新する目的のために、経路748は双方向性であることが理解されるべきである。同様に、ウェアレベリング機能も、例えばテーブル領域620のF T L領域を編集することにより、F T L機能にとって可視状態の変更を行うことができる、一例として、書込動作の前に、F T Lは、書き込みに好適な位置をウェアレベリング機能から入手する。なぜならば、ウェアレベリング機能の役割は、さらに詳述するように、全ての利用可能な位置内からの各新規書込動作について物理的位置を定義することであるからである。ガーベジコレクション機能612からの更新は、ガーベジコレクション更新経路750上において利用可能となる。これらの更新は、通常のF T L更新およびガーベジコレクション更新に加えて、ウェアレベリング機能への専用としてもよい。ウェアレベリング機能はガーベジコレクション機能にとって可視性であるため、例えばテーブル領域620を介して、ウェアレベリング機能は、ガーベジコレクション機能によって利用可能な状態に戻されたメモリから、新規書き込みのための物理アドレスを割り当てることができる。ガーベジコレクション機能を更新する目的のために、経路750は双方向性になっていることが理解されるべきである。

10

【0087】

760における各書込動作は、メモリ構成内の書き込みのための物理的位置を確立するために、ウェアレベリング機能の回収構成ステップ744とインターフェースをとり得る。さらに記載するように、ウェアレベリング機能は通常は、（主要コントローラ40（図2）からの書き込みコマンドによって開始される）書き込み機能と、F T L機能との双方と通信するため、経路748および760は双方向性である。新規書き込みデータ位置が書き込み機能によって用いられる場合、この新規位置を（データ書き込み対象となる論理アドレスに対応する物理的位置として）F T Lに付加する必要がある。既存の論理データ単位が書き込みコマンドによって上書きされている場合、ウェアレベリング機能は、論理上書きが構成更新の目的のために（NANDフラッシュの場合のように）新規物理アドレスへのデータ書き込みだけでなく、データの現在の物理的位置の割当解除にも対応していることを認識する必要がある。

20

【0088】

消去動作においては、書込動作と同様に、ウェアレベリング機能およびF T L機能双方が対応すべき変更を必要とする物理的動作が行われる。そのため、770において、各書込動作は、構成回収ステップ744とインターフェースをとって、メモリ構成内における消去のための物理的位置を確立する。消去機能更新の目的のため、経路770は双方向性であることが理解されるべきである。論理消去においては、既存の物理的データ単位の割当解除が行われ得る。物理的消去は、メモリの物理的範囲の実際の消去に対応し、その結果、当該メモリに対する新規書き込みが可能となる。

30

【0089】

774において、部分（セクション）割当ステップにおいて、回収された構成に基づいて、モジュール中の累積メモリデバイス上における現在の部分（セクション）構成を調査し、新規部分の割当が必要かを決定する。現在の部分（セクション）構成への部分（セクション）付加は全て、ウェアレベリング構成に対して更新される。後者は、例えば機能エンジンのテーブル領域620（図6）中へと保存され得る。

40

【0090】

776において、各部分（セクション）内のブロック構成を分析して、現在未割当のブロックを割当する必要があるかを決定する。付加が行われると、ウェアレベリング構成の適切な部分へ付加が行われる。また、変更はF T Lおよびガーベジコレクション機能とも呼ばれる。

【0091】

778において、陳腐化したブロックでいっぱいになった部分（セクション）を再割当することができる。NANDフラッシュにおいて、この動作は、消去ステップの後、新規消去されたメモリを1組の利用可能な部分（セクション）に付加するかまたは新規消去さ

50

れたメモリを新規部分への再割当を待機している空の記憶部へ付加することに対応する。

【0092】

780において、部分内の各ブロックを評価して、当該ブロックを再割当する必要があるかを決定する。再割当は通常は、陳腐化したデータ（書き込まれているが無効フラグが付けられているデータ）がブロックに書き込まれる結果発生し、この場合、ブロックを消去することができ、新規書き込み用に利用可能である旨のフラグを空のブロックに付ける。次に、その結果得られた構成を730において保存し、動作分岐740が終了して、再実行の必要性が保留となる。

【0093】

図8を参照して、図6のガーベジコレクション機能612の実施形態は一般的には、参照番号800によって示される。ガーベジコレクション機能は、804における主要コントローラ40から利用可能なパラメータ801の一部と、機能エンジンの共有パラメータ部322（図3）に含まれる806におけるモジュールから利用可能な内部パラメータとに基づいて動作することができる。パラメータ源に関係無く、ガーベジコレクション機能は、前回書き込まれたメモリ位置をリクレイムし、ウェアレベリング機能と協働してメモリ位置を後続書き込み対象としてアクセス可能とすることに関する。本開示において用いられるように、ページが最下層レベルに位置するような単位の階層として使用できるようにメモリが割り当てられる。限られた数のページが累積してブロックとなる。1つのページは通常は、一定数の物理的記憶素子からなる。しかし、1つのページ内のデータ記憶量は、ビット密度構成によって異なり得る。部分（セクション）は、複数のブロックから構成され、1つの部分（セクション）中のブロック数は、部分（セクション）毎に異なり得る。各モジュールは、或る数の部分（セクション）（図5を参照）を含む。これらの部分は、モジュールのメモリデバイス上において多様な状態で割り当てることができる。

【0094】

不揮発性メモリの動作は一般的には、メモリデバイスの物理的構成と無関係の規則に縛られることが理解されるべきである。本議論の目的のため、非限定的な例として、一般的にはNANDフラッシュメモリの挙動に準拠した規則が用いられるが、当業者であれば、本議論を他の不揮発性メモリ技術に容易に適合することができる。メモリデバイス（例えば、NANDデバイス）の読み出しおよび書き込みについての一般的規則を以下に示す。

【0095】

a. 1つのページは、一度に書き込むことが可能な最小物理的メモリ範囲である。ページへの部分的書き込みは不可能である。

b. 1つのブロック中の全ページを満たすだけの十分なデータを収集した後、ブロックへの書き込みを開始し、その後、データ全てを1単位としてブロック中へと書き込むことが（必須ではないにしろ）望ましい。このようにすることで、ページ間干渉機構に対する補償が得られる。

【0096】

c. 1つのブロックは、一度に消去することが可能なメモリの最小単位である。

前述の規則は、読み出しおよび書き込みアクセスのためのブロック中心の機構を暗示する。そのため、ガーベジコレクションステップにおいて、全ての活性データページまたは全ての陳腐化データページを含むブロックとしてページを整理することによって活性データページから陳腐化ページを分離することにより、記憶容量が取り出され得る。陳腐化データページしか含んでいないブロックは消去することができ、利用可能な状態にすることができる。ここで、本文脈における「陳腐化」という用語はメモリ単位（ページ、ブロックまたは部分（セクション））を指し得、この単位の要素に書き込まれている値はもはやモジュールによって有効と認識されていない点に留意されたい。これは、モジュールによって有効と認識されている書き込みデータの別の場合とは対照的である。この後者の場合を「活性（アクティブ）」データと呼ぶ。要素中の消去状態意外に何も値が現在書き込まれていないデータ単位を「利用可能」なものとして定義する。

【0097】

再度図 8 を参照して、主要コントローラ 40 およびモジュール 30 から得られたパラメータ 801 のサブセットと、他の構成パラメータとがウェアレベリング機能に適用される。他の構成パラメータを挙げると、メモリ構成およびウェアレベリング構成に関する情報がある。メモリ構成は、モジュールのメモリデバイスのうちの割り当てられた部分上の、セクション、ブロックおよびページの配置構成を含み、ウェアレベリング構成は、ガーベジコレクション機能とウェアレベリング機能との間の関係を含むパラメータを含み得る。

【0098】

ガーベジコレクションは、1組の動作を含む。この1組の動作に起因して、陳腐化したページの特定と、陳腐化ページの陳腐化ブロックへの結合と、ブロック消去とが行われる。機能はまた、部分（セクション）のレベルにおいて動作することができ、陳腐化したデータ単位を部分（セクション）へ結合した後、部分（セクション）の構成ブロックを消去することにより、これらの部分を利用可能な部分中へと移動させる。ガーベジコレクション機能および消去機能は、メモリ中のブロックへ影響を与える更新についての情報を通信して交換することが理解されるべきである。消去機能 802 は、ガーベジコレクション機能に対してブロック消去完了を通知し、その結果、ガーベジコレクション機能は、消去保留となっているブロックのリストからこのような消去ブロックを消去することができる。803 において、双方向通信をパラメータ 801 を介してガーベジコレクション機能 612 とウェアレベリング機能 800 との間に行う。例えば、ウェアレベリング機能は、書き込み完了および消去完了双方について、ガーベジコレクション機能へと信号送信する。これらの通知に起因して、ガーベジコレクション機能は、消去保留となっているブロックと書き込まれているブロックとについて自身が保持している構成情報を更新する。その後、ガーベジコレクション機能は、これらの変更についてウェアレベリング機能に通知し、これにより、ウェアレベリング機能は、次の書き込み位置としての優先順位に従って、利用可能な物理アドレスのリストを命令することができる。

【0099】

ガーベジコレクション機能は開始 810 から開始し、部分毎に行われる。第1のステップにおいて、第1の部分の状態が 812 において取り出される。814 において、部分状態によって反映される部分状態を測定するための部分基準が適用される。部分は：（a）有効ブロックを完全に含み得るか、（b）陳腐化したブロックおよび有効ブロックの混合物を含み得るか、または、（c）陳腐化したブロックを完全に含み得る。場合（a）が適用される場合、現在の部分においてウェアレベリング活動を行う必要は無いが、部分内において、部分のコンポーネントブロックにガーベジコレクションを適用する必要がある得、その結果、制御がブロックレベルプロセスへ転送され、このブロックレベルプロセスは、ブロック状態入手ステップ 816 を開始する。場合（b）が 814 において適用された場合、部分基準が肯定され、陳腐化したブロックの数と、現在の部分の特定の構成について適用されたブロック閾パラメータ 818 とを比較する。陳腐化したブロックの数が閾値よりも高い場合、820 において部分を再割当するプロセスが行われる。実施形態において、再割当において、現在の部分と同一の構成の新規部分が割り当てられ得、未だに有効なブロック全てを現在の部分から新規部分へとコピーされる。その後、現在の部分中の全ブロックを陳腐化したものとして指定する。その後、ブロックそれぞれを消去することにより、現在の部分を消去することができる。これらのステップの後、現在の部分は、未割当のメモリのプールへと戻ることができる。814 において場合（c）が適用された場合、部分基準が肯定され、陳腐化したブロック数が閾値を超える。なぜならば、ブロック全てが陳腐化しているからである。820 における再割当により、ブロック全てが消去され得、現在の部分が未割当のメモリのプールへと返送され得る。その後、動作は 822 へと進み、別の部分が利用可能かが決定される。別の部分が利用可能ではない場合、プロセスは 824 において終了する。別の部分が利用可能である場合、動作は 812 へと戻って、部分は現在の部分となる。割り当てられた部分全てが取り扱われるまで、プロセスが繰り返される。

【0100】

10

20

30

40

50

8 1 6においてブロックレベルプロセスに入った部分については、上述したように、ブロックレベルプロセスは部分内において動作する。いくつかの場合において、各部分においてブロックレベル処理が必要になり得、これにより、モジュール全体上においてブロックレベルプロセスが完了した場合、モジュールの部分それぞれの内部においてブロックレベル処理を連続して行うことが可能になる。

【0101】

ブロック基準決定830において、ブロック再割当を必要とする1つのブロック内の陳腐化ページ数のページ閾パラメータ832が呼び出される。このブロックレベルプロセスは、陳腐化単位が部分を構成するブロックではなくブロックを構成するページである点を除いて、814の部分基準について上述したような場合(a)、(b)および(c)における部分レベルプロセスに類似する。830において、ページ閾値832を超えるページが現在のブロック内において陳腐化した場合、場合(b)および(c)に対応して、新規ブロックが836において現在の部分内に割り当てられる。新規ブロックは、未だに活性であるページを現在のブロックから受信する。新規に割り当てられたブロックに活性ページを充填するために、このステップを複数の現在のブロック上において同時に実行することにより、新規に割り当てられたブロックに複数の現在のブロックからの活性ページを含ませることが可能になる点に留意されたい。その後、活性ページが除去された現在のブロック(または現在のブロックの組み合わせ)を消去してから、部分内の利用可能な記憶部へと消去されたブロックを返送することができる。従って、新規ブロック(単数または複数)が現在の部分内へと割り当てられて、消去対象となっているブロックから有効ページが受信される。このような割り当てられたブロックは、活性データページを含む。

【0102】

ブロック再割当が完了した後、動作は840へと進み、最終ブロックが確認される。別のブロックが利用可能である場合、動作は816へと戻り、ブロックレベル取り扱いの目的のために次のブロックが現在のブロックとして割り当てられる。他に利用可能なブロックが無い場合、動作は842へと進み、別の部分が試される。別の部分が利用可能である場合、動作は812へと戻る。一方、別の部分が利用不可能である場合、動作は824において終了する。よって、ブロックレベルプロセスは、ブロックレベル処理が行われている1つの部分中の全ブロックについて実行されるが、しかし、一定比率の活性データまたは陳腐化したページを含むブロックのみがプロセスによって調査される。活性データページも陳腐化したデータページも含まないブロックの場合、このプロセスによる調査は不要である。しかし、このようなブロックは、プロセスによる割当のために利用することが可能である。

【0103】

824において最終部分が完了することは、ウェアレベリング機能において1つのアプリケーションが完了することに対応する。いくつかの実施形態において、モジュールによって明確に呼び出された場合にのみ、ウェアレベリング機能が実行され得る。他の実施形態において、連続的に実行するバックグラウンドタスクとしてウェアレベリング機能を用いることができる。ウェアレベリングが完了した場合、ブロックレベルまたは部分レベルにおいて、ガーベジコレクション機能によって更新750(図7)が適用され得る。更新750は、ウェアレベリング機能700(図7)および後述するFTL機能にとって可視性であり、ウェアレベリング機能700(図7)および後述するFTL機能へ影響を与え

【0104】

図9を参照して、図6のビット密度構成機能614の実施形態は一般的には、参照番号900によって示され、モジュールの不揮発性メモリのビット密度を構成する機能を示すフロー図として示される。ビット密度構成機能は、904において主要コントローラ40から利用可能なパラメータ902の一部と、機能エンジンの共有パラメータ部322(図3)に含有され得る、906においてモジュールから利用可能な内部パラメータと、に基づいて動作することができる。この実施形態によれば、特定のビット密度における動作の

ためにモジュール内に部分が再割当される機能は、ガーベジコレクション機能による部分の初期割当の後に行われることが図示されている。多数の異なるビット密度が利用可能であり、これらの利用可能なビット密度は、モジュールのメモリデバイスのメモリ技術と、読み出し動作および書込動作時に用いられる符号化／復号化機能との双方に少なくとも部分的に基づいて決定することができることが理解されるべきである。ビット密度構成は、シングルビットパーセル構成からマルチビットパーセル構成へと変化し得る。セル毎に保存されたビットについての各ビット密度値も、符号化および復号化アプローチと、符号化および復号化機能によって用いられるさらなるオーバーヘッドの量および種類とに関係して変動し得る。符号化および復号化アプローチは、上述したようなコードの組み合わせを含み得る。この組み合わせは、適用されたオーバーヘッドの量の関数として結果を達成する。畳み込みコードおよびブロックコードはどちらも、多様なレベルのオーバーヘッドにおいて動作し得、特定の量のオーバーヘッドが、このようなコードをコントローラ設定において実行するために適用されたパラメータによって決定される。オーバーヘッド量は通常は、S N Rと、手元にあるメモリデバイス部の他の任意の劣化機構特性とを前にして最適性能が得られるように指定される。ビット密度は、N A N Dフラッシュ中のブロック単位レベルまで適用可能な属性として、適用され得る。最終的に、ビット密度構成の粒度は、選択された符号化および復号化機能と、モジュールのメモリデバイスにおけるメモリ技術との関数であり得る。本開示に記載される例において、所与の部分内の全ブロックが当該部分に対して選択されたビット密度において動作するように、ビット密度構成が当該部分レベルに設定される。この部分は、単一のブロックを含み得、これはN A N Dフラッシュの利用に対する関係を示し、個々のビット密度構成は、単一のブロック位に小さいデータ保持単位に適用される点に留意されたい。

【0105】

方法900は、初期化分岐910と、更新分岐912とを含む。初期化分岐910は、新規部分が割り当てられる（または例えばガーベジコレクションおよびメモリ再構成に起因して再割当される）たびに、実行される。更新分岐912は、1つの部分の下側の物理的メモリの1領域に適用されるビット密度構成について、モジュール寿命の間に定期的に行われ得る。メモリデバイスの寿命にわたって、特定の範囲の物理的メモリは、異なる時期において異なる部分に属し得るが、任意の所与の時期において、特定の範囲は、1つよりも多くの部分に属さない場合がある。活性利用されている部分の場合、ビット密度構成機能は、部分を構成している物理的メモリの状態を監視することができる。例えば、特定のビット密度構成において動作している物理的メモリの一定部分を監視しているビット密度構成機能は、物理的メモリの劣化を検出し得、これにより、同一の物理的メモリの後続再割当において、劣化に起因するより低いS N R（信号対ノイズ比）により良く耐久するビット密度構成が必要になり得る。

【0106】

主要コントローラ40（図2）からのパラメータは、モジュールから供給されたものよりも優先され得る。いくつかの場合において、主要コントローラは、機能エンジンによってコントローラパラメータに基づいて行われるビット密度構成機能のうち大部分または全体をカスタマイズすることができる。他の場合において、主要コントローラは管理をモジュールに委託して、ビット密度構成機能がモジュールパラメータによって定義されるようにする。さらに他の場合において、1組の管理用パラメータは、コントローラおよびモジュールパラメータの協働的な組み合わせであり得る。しかし、いずれの場合においても、実際のビット密度構成機能は、例えば自律的にパラメータが確立された後、特定パラメータ源に無関係にモジュールによって行われる。パラメータ値は、各部分についてのビット密度構成の選択と、1つのビット密度構成から別のビット密度構成への変化と、さらには（最大数のプログラム／消去サイクルを超えることに起因して「摩耗した」コンポーネントメモリの場合のような）活性メモリプールからの一部の割当解除とについて、ビット密度の局面を決定することができることが理解されるべきである。

【0107】

パラメータ 902 のサブセットは、包括的なものを意図していない点に留意されたい。主要パラメータ分類を挙げると、標的使用モード（例えば、容量、密度）ならびに寿命基準（例えば、プログラム／消去サイクル数）がある。パラメータによって決定される他の基準としてメモリ物理特性があり、割当、再割当および割当解除を行う際にビット密度構成機能によって用いられる。別のパラメータ分類により、可能な構成が定義される（例えば、部分レベルにおけるビット密度構成）。複数の制御ディレクティブが、エラーレートおよびビット密度構成の変化を決定するための他の測定基準に適用される閾値を規定するパラメータと共に用いられ得る。これらの制御ディレクティブは、閾基準に相対して何らかのアクションをとる（かまたはとらない）ように機能エンジンを命令し得る。一例として、ページ中の修正されたエラーレートが一定の規定された閾値を超え始めた場合、関連付けられた制御ディレクティブに起因して、ガーベジコレクション機能が影響を受ける部分に対してプロセスを開始する。

10

【0108】

続けて図9を参照して、初期化分岐910は、モジュール動作の初期開始時に実行されるプロセスに適用される。初期化は一般的には、ビット密度構成の継続がパラメータの進行中の維持（例えば、モジュールメモリの動作寿命における多様な劣化および摩耗機構から発生するもの）に依存するため、1回だけ発生する。920において、メモリ能力データの形態で情報が収集されて、モジュール中の特定のメモリの能力の決定が可能となる。メモリ能力データは、モジュール中のメモリデバイスの局所化された部分について分類することができ、これにより、パラメータ（例えば、測定基準）がこれらの局所化された部分についてそれぞれ存在する。この局所化の粒度は、実装状態に依存し得、単一のモジュール内においても変動し得る。初期メモリ能力情報は、モジュールまたは主要コントローラによってビット密度構成についてメモリ能力を決定する目的のために行われる自己試験または他の機能の1つの結果として、得られ得る。1つの有用なパラメータとして、メモリのSNR（信号対ノイズ比）の測定値があり、この測定値そのものが、メモリの素子が保存することが可能なビット数の指標である。モジュールのメモリデバイスの使用履歴が累積するにつれて、さらなるパラメータ（例えば、ビット密度構成機能に対して更新時において収集された利用統計）により、各メモリデバイスの各物理的範囲のためにビット密度構成および再構成について後続決定が可能になる。他の利用統計が例えばデコーダ202（図3）から利用可能となる。初期化および動作（例えば、前述の援用された'585出願中に記載のもの）の目的のために、品質測定基準および関連利用統計を生成することができる。上述したようなメモリ性能に関連するパラメータ（例えば、1組の測定情報）についてこの開示において用いられる一般的用語としてメモリ能力データがあることが理解されるべきである。

20

30

【0109】

1組の初期メモリ能力データが初期化分岐910によって取得された後、1組のビット密度構成パラメータ（例えば、テンプレート）を924において組み立てることができる。このパラメータは、モジュールのためのビット密度構成メニューを示す。メニュー選択は、所定の1組のビット密度構成から直接得ることもできるし、あるいは導出することもできる。所定の1組のビット密度構成は、符号化および復号化機能によって用いられるパラメータの組み合わせ全体を示す。本実施形態において、各メニュー選択はメモリの一部に適用可能であるが、他の実施形態において、メニュー選択は、ブロックレベルにおいてまたはメモリ階層中におけるより低レベルにおいて適用することができる。広範囲のパラメータがメニュー選択間において可能であるが、各パラメータは、少なくともメモリ素子中のビット数、符号化／復号化パラメータおよび用いられるべきパリティ量を定義することができる。

40

【0110】

928において、924からのビット密度構成パラメータをサブセット902からの他のパラメータと共に適用することにより、部分マップを生成する。部分マップは、図5に模式的に示すようなデータ構造であり、各割り当てられた部の位置、範囲および構成をリ

50

スト化することができる。その結果、1組の部分がモジュールのメモリデバイス（図5）上にマッピングされる。部分マップ中において、各部分は、少なくとも固有のビット密度構成と共に構成され得る。部分に、ビット密度構成を適用することにより、同一ビット密度構成（例えば、メモリ素子あたり4ビット）を用いた各部分から異なるビット密度構成を用いた各部分にわたる広範囲の変動に対応することが可能になる。

【0111】

930において、部分マップ生成の後、ビット密度構成が部分割当の一部として適用される。これは、論理アドレスに対応する物理的メモリの位置および範囲を定義する共有パラメータ部322（図3）中のFTLテーブルの更新を含み得る。加えて、（ビット密度構成テンプレートからの）ビット密度構成と、その対応する部分との間の関連付けを得ることができる。この関連付けにより、主要コントローラからのコマンドにアクセスするための論理上から物理上への正しいマッピングを後続の読み出し機能、書き込み機能および消去機能によって特定することができるだけでなく、モジュール中の読み出し／書き込み設定を正しく構成することも可能になる。

10

【0112】

部分割当930によって定義された各部分は最終的には、1組のブロックを含む。ブロックの数を定義する必要がある。934において、例えば部分を含むことになるブロックの正確な数に対応できるようにテーブル領域620中のテーブル構造を更新することにより、各部分に対してブロック割当を行う。

【0113】

20

936において、モジュールのための部分構成を保存する。部分構成は、適切なパラメータ（例えば、テーブルおよび）テンプレートを含み得る。これらのパラメータおよびテンプレートを用いて、（少なくとも初期化フローの終了時における）モジュールの部分構成を完全に定義し、用いられ得る多様なビット密度構成を認識したメモリ動作をイネーブルし、継続使用に起因する劣化および摩耗効果によって必要になり得る将来の更新をイネーブルする。

【0114】

ここで、ビット密度構成機能の更新分岐912を参照して、更新分岐は、いくつかの実施形態において、モジュールの寿命にわたって間隔を空けて実行され得る。初期化プロセスの場合と同様に、適用されるパラメータは、主要コントローラおよびモジュールのいずれかによって提供される。ここでも、主要コントローラ40（図2）からのパラメータが、モジュールから供給されるパラメータよりも優先され得る。ビット密度構成機能は、コントローラから提供されたパラメータ、モジュールから提供されたパラメータまたはこれらの任意の適切な組み合わせによって規定され得る。しかし、いずれの場合も、実際のビット密度構成機能は、特定のパラメータ源に関係無く、モジュールによって行われる。更新分岐912は通常は、既存の部分についてビット密度構成の変更が必要になる度に行われる。例を挙げると、ウェアレベリングおよびガーベジコレクションアクションに起因して部分を割当解除しかつ／または割り当てる必要またはメモリ劣化に起因してビット密度構成を訂正する必要がある。

30

【0115】

40

940において、初期化分岐910によって元々展開されたメモリ能力データを訂正することにより、更新分岐が開始する。ガーベジコレクションおよびウェアレベリング機能は、942および944それぞれにおいて入力をステップ940へ提供する。なぜならば、これらの機能はメモリ能力データに影響を与えるからである。上述したように、メモリ能力データは、メモリが多様なビット密度においてデータを保存および保持する能力を示し指標であるパラメータを含む。ガーベジコレクションおよびウェアレベリング機能からの入力により、メモリ能力データが訂正される。946において訂正されたメモリ能力データを鑑みて、再割当マップを決定する。再割当マップは、現在の部分の現在の割当と比較したときの現在の部分のビット密度構成の変化を反映する。このマップは、（ガーベジコレクション結果からの）新規割当において必要な部分を確立し、これらの部分を（ウェ

50

アレベリングから) 配置することが可能な位置を確立する。

【0116】

ステップ946からの再割当マップを用いて、ステップ948は、現在の割当から新規割当への変更を指定することにより、更新された部分マップを生成し(ただし、現在の割当が初期分岐910または更新分岐912の前の通過によって決定されている)、これらの変化が現在のバージョンの部分マップへと付加される。950において、新規/更新部分マップによって指定された通りに、新規部分がメモリへと再割当される。更新は、例えばテーブル領域620(図6)中のデータ構造へ更新を書き込むことにより、達成することができる。その後、ブロック割当は上述したように934に進む。

【0117】

10

図10を参照して、図6のフラッシュ変換層(FTL)機能616の実施形態を主に参照番号1000によって示す。図6は、本開示のモジュールのうちの1つの不揮発性メモリに関連するフラッシュ変換層を管理するためのフロー図を示す。FTL機能は、コマンド中に設けられた論理アドレスに基づいてデータの物理的位置を見つけることにより、主要コントローラコマンドにサービスする。この点について、図6を参照して、ウェアレベリング機能610、ガベージコレクション機能612およびビット密度構成機能614の活動に起因して、モジュール中への記憶データの物理的配置が動的に行われる点に留意されたい。よって、FTL機能は、静的物理的配置のための論理/物理アドレス変換よりも複雑なものである。すなわち、FTL機能は、その他の機能から生成されたパラメータを用いて、部分、ブロックおよびページ中へのデータ配置に対応する動的FTLデータ構造を決定して、対応する論理アドレスのみに基づいて活性データ単位を常に配置することができるようにする。これらの動的データ構造は、図6のテーブル領域620中に保存することができる点に留意されたい。

20

【0118】

FTLパラメータ部1002は、機能エンジンの共有パラメータ部322(図3)の一部を形成することができ、主要コントローラからのパラメータを1004において受信し、1006においてモジュールからモジュールパラメータを受信する。これらのパラメータの値は、モジュールによって実行されるカスタマイゼーション量とは対照的に、主要コントローラによるFTL機能の相対的カスタマイゼーションレベルに影響を及ぼし得る。他の機能の場合と同様に、上述したように、いくつかのパラメータ値に起因して、主要コントローラは、モジュールに相対してFTLへより高く影響し得、他のパラメータは、モジュールによって影響を受ける傾向になる。主要コントローラまたはモジュールによる影響レベルに関係無く、機能エンジンは、パラメータ入力のみに基づいてFTL機能を例えば自律的に行うことができる。

30

【0119】

FTLパラメータ部1002は、以下の情報も受信する: 1010においてウェアレベリング機能700(図7)によって生成された情報、1012においてガベージコレクション機能800(図8)によって生成された情報、1014においてビット密度構成機能900(図9)によって生成された情報。FTL機能の初期化分岐1020は、モジュールの初回利用の前かつユーザデータ記憶動作の任意の実行における主要コントローラによる任意のアクセスの前に、実行される。1024において、ウェアレベリング機能からの入力1010およびビット密度構成機能からの入力1014(例えば、FTLパラメータ部1002によって反映されるようなもの)により、初期データ構造の構成を、論理テーブル入力の出力生成と共に開始することが可能になる。これらのデータ構造は、本質的に多レベルであるため、部分、ブロックおよびページレベルにおけるマッピングを提供する。マッピングも、本質的に動的である。なぜならば、マッピングは、新規データがモジュールに書き込まれた際、および、前回書き込まれたデータが上書きまたは消去に起因して無効化される際に変更が必要である。

40

【0120】

1028において、物理的位置がデータ構造中の各論理入力に対応するように、論理/

50

物理マップ入力各論理アドレスについて生成される。初期化分岐は、実際のデータがモジュール中に書き込まれる前に発生するため、論理入力は、モジュールメモリ中の既に割り当てられかつ未だ空の領域を指定する。よって、この時点におけるデータ構造は、主要コントローラにとってアクセス可能でありかつデータを含んでいないメモリボリュームを形成する1組の部分の指し得る。論理データブロックおよび物理的ブロックとボリューム内のページとの間の接続によりこれらのデータ構造を増強することは、後続の読み出し機能、書き込み機能および消去機能のタスクである。

【0121】

初期化の後、1030において、通常動作に入り、メモリがデータ保存のためにアクセス可能となる。FTL機能との相互作用に関する限り、読み出し動作、書き込み動作および消去動作について記載する点に留意されたい。当業者であれば、これらの動作の他の局面を理解する。機能の読み出し分岐1034は、読み出しコマンドに応答して実行することができ、1036において、機能の消去／書き込み分岐は、さらに説明するように、書き込みコマンドまたは消去コマンドに応答して実行することができる。

【0122】

読出動作は記憶データを変更しないため、FTLデータ構造の更新の際、読み出し分岐1034は不要であることが理解されるべきである。そのため、1038において、読み出し分岐は、自身に提出された各論理アドレスの物理アドレスを返送する。よって、モジュールが読み出しデータコマンドを主要コントローラから受信した場合、その結果得られたFTL機能への呼び出しは、物理アドレスルックアップ動作のみを必要とする。単一の読み出しコマンドに回答して、読み出し分岐を通る1つよりも多くの通過が発生し得る。なぜならば、読み出しコマンド中で要求されたデータは、複数の論理アドレスに存在するデータを含むからである。

【0123】

消去／書き込み分岐1036は、前回書き込まれていないデータブロックが書き込まれているときまたは現在書き込まれている論理アドレスの存在するデータが上書きされているときに、実行され得る。前者の場合において、論理アドレス（単数または複数）をメモリ中の物理的位置へと割り当てる必要がある。そのためには、物理的メモリを既存の部分内から特定のブロックおよびページへと指定する必要がある。コマンドによって書き込まれるべき全データに対応する物理的位置へと割り当てるために、必要なだけ消去／書き込み分岐を通過することができる。1040において、書込動作の対象となる物理的メモリを割り当てることができる。1042において、書込動作を反映するように、FTLテーブルを更新する。消去動作が必要な場合、後者の場合、ステップ1040は、論理アドレス（単数または複数）に対応する物理アドレス（単数または複数）に前回書き込まれたデータを無効化するために、対象メモリ位置の消去を考慮することができる。その後、ステップ1042は、FTLテーブルを更新して、書き込み機能に備えた無効データを含むものとして対象メモリ位置を指定する。FTL消去機能は、主要コントローラからモジュールへの書き込みコマンドまたは消去コマンドと共に行われ得る。消去／書き込み分岐1036のFTL消去機能は、対応する論理アドレスの物理的消去を達成しないが、その代わりに、テーブル構造を更新して、論理アドレスと物理的位置との間の対応が無効となったことを示す。よって、消去／書き込み分岐1036を通じた第1の通過は、FTL構造についての消去を考慮することができ、第2の通過は、対応する書込動作を考慮することができる。テーブル入力は、ガーベジコレクション機能612が陳腐化した物理的データ単位を特定した際にガーベジコレクション機能612によって後で更新することができ、物理的ブロック消去を行って、影響を受ける物理的データ単位を後続書き込みにおいて利用可能とする。通常動作は1044へ進む。

【0124】

ここで図11を参照して、読出動作の実施形態がブロック図の形態で図示され、機能エンジン320（図6）による読出動作の補助的な相互作用を例示する目的のために、主に参照番号1100によって示されている。1102において、読出動作が開始される。読

出動作の機構は、個々のメモリ素子のアクセスおよび読み出しに関する限り、当業者にとって周知であり、広範な適切なプロセスをこの目的のために利用することが可能である点に留意されたい。各読出動作は、論理アドレスに向けられる。論理アドレスは、読み出し実行コマンドの一部としてホストから外部的に得られる場合もあれば、あるいは、内部的にモジュールへ得られる場合もある。いずれの場合においても、1104において、実際のデータが読み出し可能となる前に、読出動作により、論理アドレスがF T L機能616（図6および図10）へと送られる。1106において、F T L機能616は、読出動作からの論理アドレスの受信に応答して、物理アドレスを前記読み出し機能へと返送する。これは読出動作であるため、テーブル部620（図6）中のF T Lテーブルおよびウェアレベリングおよびガーベジコレクション機能に対する更新は不要である。なぜならば、メモリは何も変更されないからである。物理アドレスを入手した後、1108において、読出動作は、読み出しデータを得るように実行される。図11は、記載を簡略化するために、単一の論理アドレスまたは物理アドレスの交換に基づく、読み出し動作について言及しているように見えるが、読出動作は複数のアドレスを用い得ることが理解されるべきである。よって、図示された読出動作の多様なステップにおいて、物理アドレスそれぞれに対する対応する読み出しステップを用いた複数の論理／物理アドレス変換も行われ得る。1110において、読出動作が完了し、通常動作が再開する。

【0125】

図12を参照して、書込動作の実施形態がブロック図の形態で図示されており、機能エンジン320（図6）との書込動作の補助的相互作用を例示する目的のために、主に参照番号1200によって示されている。1202において、書込動作が開始される。書込動作の機構は、個々のメモリ素子に対するアクセスおよび書き込みに関する限り、当業者にとって周知であり、この目的のために広範な適切なプロセスを用いることが可能である点に留意されたい。書込動作に起因して、一定範囲のメモリが書き込まれる。このような動作は、前回書き込まれていない論理アドレスに対して行うこともできるし、あるいは、現在書き込まれている論理アドレスに対して行うこともできる。いずれの場合においても、書込動作は、1204において、標的論理アドレスをF T L機能616（図6および図10）へと送る。1206において、F T L機能は、書き込みデータがメモリ中へと物理的に書き込まれるべき物理アドレスを返送することにより、応答する。当該書き込みが新規である（前回書き込まれていない論理アドレスに対するもの）であるにしろまたは上書きである（現在書き込まれている論理アドレスを標的とするもの）であるにしろ、F T L機能は、書き込まれるべき物理アドレスを返送する。いずれかの場合も、F T L機能によってF T L構成に対して更新が行われ、これは、ウェアレベリング機能610（図6および図7）ならびにガーベジコレクション機能612（図6および図8）にとって可視化状態となる。1208において、書込動作によって用いられる物理アドレスが返送されるのに応答して、書込動作が実行される。図12は記載を簡潔にするために単一の論理アドレスまたは物理アドレスの交換に基づいて書込動作に言及しているかのように見えるが、書込動作において、複数のアドレスを行うことが可能であることが理解されるべきである。よって、例示されてきた書込動作の多様なステップにおいて、物理アドレスそれぞれについての対応する書き込みステップを用いた複数の論理／物理アドレス変換が含まれ得る。同様に、F T L、ウェアレベリングおよびガーベジコレクションパラメータを更新することで、書込動作によって書き込まれた複数の物理アドレスを反映する。1210において、通常動作が再開する。

【0126】

図13を参照して、消去動作の実施形態がブロック図の形態で図示されており、機能エンジン320（図6）との消去動作の補助的相互作用を例示する目的のために、主に参照番号1300によって示されている。1302において、消去動作が開始される。消去動作の構造は、個々のメモリ素子のアクセスおよび消去に関する限り、当業者にとって周知であり、この目的のために広範な適切なプロセスを用いることが可能である点に留意されたい。

10

20

30

40

50

【0127】

ここに示される消去動作の結果、一定範囲のメモリの物理的消去が行われる。そのため、この動作は、ホストから送られた消去コマンドから発生し得るような論理消去とは異なる。しかし、ホスト消去コマンドは最終的にはこの動作を呼び出す。このような物理的消去は、ホストからではなくモジュールから行われる。

【0128】

図6および図13を参照して、1304において、現在のガーベジコレクション構成パラメータに例えば機能エンジン（図6）のテーブル部620からアクセスして、消去すべき物理的ブロック位置を得る（ただし、利用可能である場合）。このような消去対象となる物理的位置の同定情報は、ガーベジコレクション機能612（図6）によって生成され得る。1306において、消去のためにブロックが利用可能であるかが決定される。このようなブロックアドレスが現在利用できない場合、消去プロシージャは1310において終了して、通常動作へと戻る。ブロックアドレスが利用可能である場合、消去プロシージャは、ガーベジコレクション機能によって指定された消去対象として利用可能な次のブロックアドレスを得る目的のために1312へと進む。1314において、消去機能は、1312において得られた物理アドレスにおけるブロックをアクセスおよび消去するための消去動作を実行する。消去が完了した後、動作は1316へと進み、例えばテーブル部620を介してガーベジコレクション機能612を完了状態とする。実施形態において、完了状態はガーベジ接続機能へと直接提供され得るその結果、ガーベジコレクション機能の構成の更新が可能となる。1310において、消去動作が終了して、通常動作状態へと戻る。

【0129】

図3および図6を参照して、各モジュール30は、コントローラパラメータ630に応じて、多様なモードにおける動作に合わせて構成することができる。実施形態において、モジュールは、第1のモードにおいて機能のうち選択された1つに合わせてコントローラからの第1の1組のモジュール入力パラメータに基づいて動作することができ、第2のモードにおいて選択された機能に合わせてコントローラからの第2の1組のモジュール入力パラメータに基づいて動作することができる。第1の1組のモジュール入力パラメータは、第2の1組のモジュール入力パラメータと異なる。1組のパラメータは、例えば共有パラメータ部322によって保存され得る。例示目的のため、モジュールは、モジュール入力パラメータに基づいて、第1のモードをモジュールの所与の不揮発性メモリ部の第1の部分（例えば、部分）へ適用し、第2のモードを所与の不揮発性メモリ部の第2の部分（例えば、部分）へと適用するように構成され得る。第1の部分は、第2の部分と異なる。よって、図5において、入力パラメータに基づいて、NVM1（部502）は第1のビット密度でデータを保存するように構成され得、NVM2およびNVM4（部504）は、第1のビット密度と異なる第2のビット密度でデータを保存するように構成され得る。実施形態において、不揮発性メモリ全体の一部分をコントローラパラメータ630に基づいて決定することができ、モジュールの不揮発性メモリの1つ以上のモジュールパラメータ（例えば、監視特性）に基づいてビット密度を決定することができる。いくつかの実施形態において、モジュールパラメータは誤差修正統計値であり得、例えば以下にさらに説明するようにデコーダによって生成され得る。

【0130】

いくつかの実施形態において、1つ以上のモジュールが、共有パラメータ部322（図3および図6）を介してコントローラから送られた第1の組のモジュール入力パラメータに基づいて、機能のうち選択された1つに対して第1のモードにおいて動作することができ、これにより、このように構成された各モジュールが（コントローラからの管理またはコントローラとの相互作用無く）機能を例えば自律的に行い、共有パラメータ部322からの第2の組のモジュール入力パラメータに基づいて選択された機能に対して第2のモードで動作することができる。第1の組のモジュール入力パラメータは、第2の1組のモジュール入力パラメータと異なるため、このように構成された各モジュールは、少なくとも

コントローラからの部分的制御の下、機能を第2のモードにおいて行う。例えば、第1のモードにおいて、モジュールは、テーブル領域620（図6）内のフラッシュ変換層をコントローラから独立して自律的に管理することができ、これにより、ローカルフラッシュ変換層はコントローラによってアクセス可能となるが変更はされず、一方、第2のモードにおいて、モジュールは、コントローラからの一定の貢献（例えば、モジュールフラッシュ変換層のモジュール部を変化させること）により、フラッシュ変換層全体のモジュール部を管理することができる、なぜならば、コントローラは、多様なモジュール全てのフラッシュ変換層に対して可視性であるからである。

【0131】

図5を参照して上述したように、本開示は、ページ境界を跨ぐことが可能なブロック抽出を提供する。ブロック抽出についてのさらなる詳細について、先ず図14を参照して説明する。図14は、物理ページ1402の模式図である。物理ページ1402は、典型的な不揮発性メモリにより、第1のユーザデータブロック1404および第2のユーザデータブロック1406に関連して提供される。第1のユーザデータブロック1404は、物理ページ1402よりも短い。第2のユーザデータブロック1406は、物理ページ1402よりも長い。長さは、関連付けられた量のデータを示す。上述したように、従来技術は一般的には、物理ページサイズとブロックサイズとの高精度のマッチングを強要する。ブロックがページよりも短い場合、フィラービットを用いて未充填ページ中の空き空間を充填することができるが、ブロックがページよりも長い場合、従来技術の従来の制約に起因して、ブロックを保存することができない。従来技術の場合、これらの状況下において保存を行う場合は、各ページのうち少なくとも一部が利用されず、このような利用されない部分のサイズは、ブロックサイズとより大型のページサイズとの間の不整合に依存する。そのため、従来技術の場合、ユーザデータのページサイズ単位は、極めて特定の量のデータとなる。しかし、図14のユーザデータブロックはそれぞれ、図示の物理ページと長さが異なる。物理ページ中の合計素子数未満の素子数を用いたユーザデータブロックの場合、従来技術は、このような状況に対応することができる。なぜならば、ブロックは（未使用の素子（セル）がある場合でも）1つの物理ページ中へ適合することができるからである。そのため記憶効率において損失が発生する。ユーザデータブロックが物理ページの範囲を超えた場合、従来技術の場合、このような状況に対応することができない。なぜならば、ユーザブロックを単一の物理ページ中に適合させることが不可能だからである。

【0132】

しかし、本開示は、各ブロックのサイズを単一の物理ページに適合させる必要無く情報ブロックを不揮発性メモリデバイス中に保存することにより、従来技術の制約を一蹴する。例えば、各物理ページ中に含まれるスペアビット数に関係無く、ユーザデータブロックを物理的メモリに割り当てることが可能である。この点について、各ページには一般的にはスペアビットが割り当てられる。スペアビットは、例えば、パリティデータ、多様な形態のメタデータ（例を非限定的に挙げると、論理ブロックアドレス、およびFTL、ウェアレベリング、およびガーベジコレクション機能によって用いられ得るその他のフラグ、ポインタおよびカウンタカウント）に方向付けられることが理解されるべきである。本明細書中の教示内容を通じて、ユーザデータ、ECCパリティまたはメタデータとしての使用を考慮することなく、各物理ページ中のユーザデータビットおよびスペアビットの組み合わせ全てが、実質的に各ページ中の記憶空間となる。そのため、図14のユーザデータブロックにおいて、実際のユーザデータとオーバーヘッドデータとは区別されない。しかし、関連付けられたオーバーヘッドを含むユーザ情報のブロックを、第1のユーザデータブロック1404の場合のように物理ページ1402よりも小さくすることもできるし、あるいは第2のユーザデータブロック1406の場合のように物理ページ1402よりも大きくすることもできる。いずれの場合においても、理解されるように、ユーザブロック数と物理ページ数との対応関係を用いること無く、複数のユーザブロックシーケンスを物理ページの群に適合させることができる。非限定的な例として、ユーザデータブロックは、関連付けられたオーバーヘッドを含む1つ以上の512バイト単位（セクター）とし

て定義することができる。すなわち、1つのブロックが1つのセクターまたは複数のセクターで構成されているかは重要ではない。なぜならば、ブロックは所望のサイズでよいからである。特定のページ内の各ブロックの起点（例えば、開始点）は、指定されたオフセットによって特徴付けられる。オフセットの追跡は、ファームウェア、ハードウェアまたはこれらの任意の適切な組み合わせで実装することができる。1つのユーザブロックが1つの物理ページを超える場合、ブロックの残り部分が後続の物理ページへと割り当てられる。このような状況が発生した場合、（以下にさらに詳述するような）ユーザブロック起点を追跡する場合と同様の方法により、ユーザブロック内の切れ目の位置をアプリケーションによって追跡する。

【0133】

ここで図15を参照して、図15は、主に参照番号1500によって示されるブロック図であり、不揮発性メモリの一部を形成する4つの連続する物理ページPP1～PP4を模式的に示す。連続する物理ページにおいて、例えば物理アドレスを所要量だけ1ページから次のページへとインクリメントすることにより、この1組の物理ページを連続的に物理的にアドレス指定することができる。この例において、ユーザデータブロックB1～B4が用いられる。これらのユーザデータブロックB1～B4それぞれの長さは、図14の第1のユーザデータブロック1404によって例示され、ユーザデータブロックは、物理ページよりも短い。例示を明確にするため、ブロック長さBLはページ長さPLの0.75に等しいと仮定する。理解されるように、ブロックB1は、物理ページPP1によって完全に包含される。ブロックB2は、第1の部分1510と、第2の部分1512とを含む。第1の部分1510は、物理ページPP1の最終の四分の一中に保存され、ブロックB1の長さ（すなわち、 $3/4 PL$ ）に対応する、ページ開始部からのオフセット位置にある。第2の部分1512は、物理ページPP2中に保存され、PP2の開始部からPP2の第1の半分を充填し、ブロックB2の最終2/3を示す。ブロックB3の初期部1520は、PP2の開始部から $1/2 PL$ だけオフセットして物理ページPP2の第2の半分中に保存される。ブロックB3の最終1/3部分1522は、物理ページPP3の第1の1/4中に保存される。ブロックB4は、物理ページPP3の残り部分1524を充填する。その後、ブロックB5は、PP4の開始から $1/4 PL$ だけオフセットして物理ページPP4の初期3/4部1526を充填する。PP4の最終1/4部分1528が空の状態で図示されているが、PP4の最終1/4部分1528を後続ブロックの一部によって充填してもよいし、あるいは空のままにしておいてもよい。本例において、5個のユーザブロックが4つの物理ページ上に保存される。理解されるように、ブロック長さがページ長さよりも短い場合でも、物理ページを完全に利用することが可能である。さらに、ページ境界を跨ぐ状態でブロックを保存することができる。図15に例示するような保存は、任意の数のユーザデータブロックに対して継続してもよいし、あるいは、シーケンス中の最終ユーザデータブロックが保存されるまで継続してもよい。いくつかの場合において、シーケンス中の最終の物理ページは未利用のメモリ素子を含み得るが、従来技術の場合と対照的に、効率を実質的に向上させて、先行する物理ページ全てを完全に利用することが可能である。

【0134】

図16は、参照番号1600によって示されるブロック図であり、4つの連続する物理ページPP1～PP4を模式的に示す。これら4つの連続する物理ページPP1～PP4は、不揮発性メモリの一部を形成する。この例において、ユーザデータブロックB1～B3が用いられる。ユーザデータブロックB1～B3それぞれの長さは、図14の第2のユーザデータブロック1406によって例示される。ユーザデータブロックは、物理ページよりも長い。もちろん、従来技術の場合、このように比較的長いブロックを保存することはできない。本例において例示を明確にするために、ブロック長さBLはページ長さPLの1.25に等しいと仮定する。理解されるように、初期ブロックB1は物理ページPP1によって部分的に包含され、B1の最終部分1610（20パーセント）は物理ページPP2によって包含される。ブロックB2の初期部1612は、PP2の開始部から1

/4 P L のオフセットで物理ページ P P 2 の残りを充填する。ブロック B 2 の最終部分 1 6 1 4 は、物理ページ P P 3 の第 1 の 5 0 パーセントを充填する。ブロック B 3 の初期部 1 6 1 6 は、P P 3 の開始部から 1 / 2 P L のオフセットで物理ページ P P 3 の最終 5 0 パーセントを充填する。ブロック B 3 の最終部分 1 6 1 8 は、物理ページ P P 4 の初期 7 5 パーセントを充填する。P P 4 の最終の 2 5 パーセントは、利用可能なブロックがもう無いいため利用されないか、または、後続ブロックによって利用可能である。本例において、3つのユーザブロックが4つの物理ページ上に保存される。ブロック長さがページ長さよりも長い場合でも、物理ページを完全に利用することができる。ここでも、ページ境界を跨ぐ状態でブロックを保存することができる。この点について、本明細書中に記載された教示内容に基づいて、ユーザデータブロックサイズと物理ページサイズとの間の任意の 10 関係に対応することができる。このような柔軟性により、従来技術で可能であった記憶効率に比して、記憶効率の向上または最大化が可能になる。このような柔軟性により、ユーザブロックサイズを変更できる能力が得られる（例えば、所与のメモリ中の物理ページサイズを対応して増加させる必要無く、ユーザブロックフォーマットに E C C パリティを付加できる能力）。所与のメモリデバイス中の物理ページリソースをフル活用できる傾向により、記憶効率の最適化が可能となる。

【0135】

図 1 7 はフロー図であり、主に参照番号 1 7 0 0 によって示される。図 1 7 は、図 1 5 および図 1 6 の文脈において上述したデータ構造を達成するような状態でユーザデータブロックの保存／書込方法の実施形態を示す。まず、図 1 5 および図 1 6 に示すように、書 20 込動作を含む連続するかまたは一連のユーザデータブロックがプロセスによって受信され得る点に留意されたい。ブロック長さおよびページ長さは異なり得る。この方法は 1 7 0 2 から開始し、1 7 0 4 へと進み、プロセスの開始時において初期ユーザデータブロックを回収する。1 7 0 6 において、物理ページの開始部から一定のオフセットを以て、現在のユーザデータブロックを物理ページ上へとマッピングする。一定のオフセットは、ゼロまたは非ゼロのオフセット値であり得る。オフセットは、例えば、先行ブロックメタデータ中の値として保存してもよいし、あるいは、F T L 機能によって生成および利用されるテーブル中に含まれるアドレスとして保存してもよい。1 7 1 0 において、現在のユーザデータブロックが書き込み転送における最終ブロックであるかが決定される。現在のユーザデータブロックが書き込み転送における最終ブロックではない場合、動作は 1 7 1 2 へと 30 進み、現在のページが満杯であるかについて試験が行われる。現在のページが満杯である場合、動作は 1 7 1 4 へと進み、ページが書き込まれる。その後、動作は 1 7 0 4 へと戻る。1 7 1 2 において、ページが満杯ではない場合、動作は 1 7 0 4 へと戻る。ここで 1 7 1 0 へと戻って、最終ユーザブロックが検出された場合、動作は 1 7 1 6 へと進み、現在のページ中に空間が残っている場合、当該空間を充填する。1 7 1 8 において、最終ページが書き込まれ、動作は 1 7 2 0 において終了し、次の書込動作を待機する。

【0136】

図 1 8 は別のフロー図であり、主に参照番号 1 8 0 0 によって示される。図 1 8 は、ユーザデータブロックを図 1 5 および図 1 6 の文脈において上記したデータ構造から読み出す方法の実施形態を示す。ブロック長さおよびページ長さは異なり得る。方法は 1 8 0 2 40 から開始し、1 8 0 4 へと進んで、プロセス開始時において初期物理ページを回収する。1 8 0 6 において、回収されたページをユーザブロックへとマッピングして、ユーザブロックを回復させる。マッピングは、ユーザブロックを取り出すためにものであり、例えば書込動作（図 1 7）時に生成されたオフセットに基づき得、先行ブロックと共にメタデータとして保存されたかまたは F T L 機能によって生成および利用されるテーブル中に保存されたアドレスとして保存される。1 8 0 8 において試験を行って、ユーザデータブロックの回復が完了したかを決定する。回復が完了した場合、回復されたブロックを 1 8 1 0 において転送する。しかし、一方、1 8 0 8 において、ブロックが1つの物理ページから次の物理ページに跨がっていること（例えば、図 1 5 中のブロック B 2 および図 1 6 中のブロック B 1 を参照）に起因してユーザデータブロックが不完全であることが特定される 50

場合もあることが理解されるべきである。この場合、動作は1804へと戻り、次のページを読み出す。この次のページは、不完全なブロックの残り部分を含む。マッピング1806により、不完全なブロックが再度組み立てられ、完成したブロックがブロック送り1810へと送られる。1812において、現在の読出動作の最終ユーザブロックが回復されたかが試験により決定される。現在の読出動作の最終ユーザブロックが回復されていない場合、動作は1804へ戻って次のページを読み出す。最終のユーザブロックが回復された場合、現在の読出動作は1816において終了し、次の読出動作を待機する。

【0137】

上記を鑑みて、極めて柔軟な読み出し／書き込み配置構成および関連付けられた方法が提供される。この構成および方法は、読出／書込動作において少なくとも1つのソリッドステートメモリデバイスへアクセスするために用いられる。メモリデバイスは、複数のメモリ素子から構成される。これらのメモリ素子は、1組のページとして編成される。これらのページは、物理的かつ逐次的にアドレス指定可能であり、1組のページ中の連続するページ間にページ境界が規定されるように、各ページがページ長さを有する。システムによって提供される制御配置構成は、連続する一連のページにおける所与の書込動作と関連付けられたデータブロックの群を保存およびアクセスするように構成され、これにより、一連のページのうち少なくとも最初のページが充填され、群の各ブロックのブロック長さは、ページ長さと異なり得る。ブロックのうち少なくとも1つは、任意のページ境界からオフセットした状態で保存することができることが理解されるべきである。一連の物理ページが最初の2ページ以上および最終ページを有する場合、システムは、データブロックの群に基づいて、最初のページのうち少なくとも各ページを充填することができる。最終ページは、充填してもよいし、あるいは部分的に充填してもよい。最終ページが部分的に充填された場合、利用可能な空間が別のブロックのうち少なくとも一部によって充填され得る。物理ページは、1つ以上のブロック境界を含み得る。ブロックの開始は、例えば、物理ページの開始からオフセットした状態で特定され得る。これを少し言い換えると、特定のブロックの第1の部分が第1のページ内に保存されかつ特定のブロックの第2の部分が第2のページ内に保存されるよう、1つのデータブロック群中の特定のブロックを保存することができ、これにより、特定のブロックの保存は、第1のページと第2のページとの間のページ境界を横断する。本議論の目的のため、第1のページおよび第2のページは、図15および図16のデータ構造内の任意の2つの隣接する物理ページであり得る。

【0138】

本明細書中に詳述される本開示のSSDおよび関連付けられたモジュールにより、分散型機能を備えたシステムが得られ、このシステムは、従来技術のアプローチの場合は集中して行われることが多かった活動を分散させることが理解されるべきである。スケーラビリティの属性は、集中型機能よりも非集中型（分散型）機能によってより容易に達成される。すなわち、非集中型機能を備えた分散型コンポーネント（例えば、上記したモジュール）を用いれば、所与のシステムがスケールアップされる際の、システムオーバーヘッドのインクリメント付加を限定することが可能になる。これとは対照的に、集中型システムの場合、記憶装置の追加と共にシステム複雑性もさらに増すため、集中型コントローラの場合、能力増強が必要になる。集中型埋設マイクロプロセッサおよび埋設ソフトウェアの場合、リソース管理能力が、中央コントローラの演算能力および他の能力の制約によって限定される。そのため、より低い機能階層レベルにおいて機能を追加することが可能なシステムを用いれば、集中型能力として同一機能を追加しようとするシステムの場合よりも、属性としてのスケーラビリティをより容易に達成することができる。

【0139】

本発明の上記の記載を、例示および説明目的のために提示してきた。この記載は網羅的なものではなく、本発明を開示の様態のみに限定するものでもない。上記教示内容を鑑みれば、他の改変および変更が可能である。当業者であれば、特定の改変、置換、追加およびその副結合が可能であることを認識する。

【符号の説明】

10

20

30

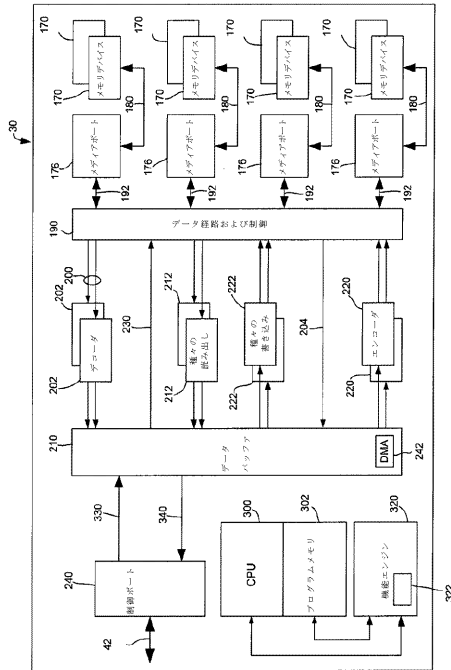
40

50

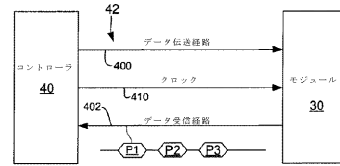
【 0 1 4 0 】

1 0	ソリッドステートドライブ (S S D)	
1 2	システム	
2 0	ホスト	
3 0	モジュール	
4 0	コントローラ	
4 2	モジュールインターフェース	
4 4	ホストインターフェース	
4 6	キャッシュ	
1 0 0	コントローラモジュールポート	10
1 2 0	データ経路および制御部	
1 2 2	バス接続部	
1 2 4	制御バス	
1 3 0	マイクロプロセッサ	
1 3 2	メモリ	
1 3 4、1 3 6	レジスタ	
1 4 0	データバッファ部	
1 4 4	ホストインターフェースポート	
1 4 6	双方向データ経路	
1 5 0	バス	20
1 7 0	メモリデバイス	
1 7 6	メディアポート	
1 8 0	メモリデバイスインターフェース	
1 9 0	データ経路および制御部 1 9 0	
1 9 2	メディアポートインターフェース	
2 0 0	デコーダ読み出しデータ経路	
2 0 2	デコーダ	
2 0 4	直接読み出しデータ経路	
2 1 0	データバッファ部	
2 1 2	読み出し論理	30
2 2 0	エンコーダ	
2 2 2	書き込み論理	
2 3 0	直接書き込みデータ経路	
2 4 0	コントローラポート	
2 4 2	D M A 部	
3 0 0	埋設マイクロプロセッサ (C P U)	
3 0 2	プログラムメモリ	
3 2 0	機能エンジン	
3 3 0	入力データ経路	
3 4 0	出力データ経路	40
4 0 0	データ伝送経路	
4 0 2	データ受信経路	
4 1 0	クロック	
5 0 0	モジュール N V M メモリ	
5 0 2、5 0 4、5 0 6、5 0 8	メモリ部 (メモリセクション)	
5 1 0 a ~ 5 1 0 d	ページ	
6 0 2	共通施設	
6 1 0	ウェアレベリング機能	
6 1 2	ガーベジコレクション機能	
6 1 4	ビット密度構成機能	50

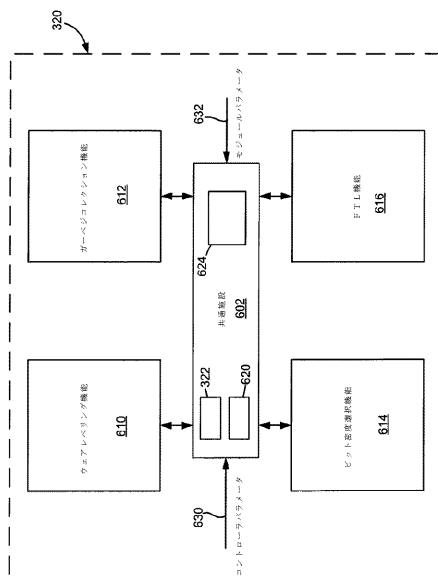
【 図 3 】



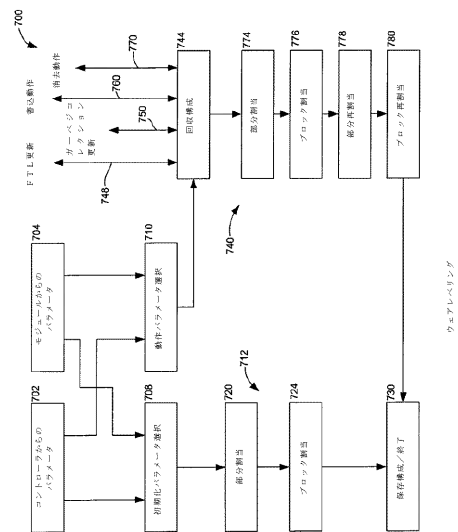
【図 4】



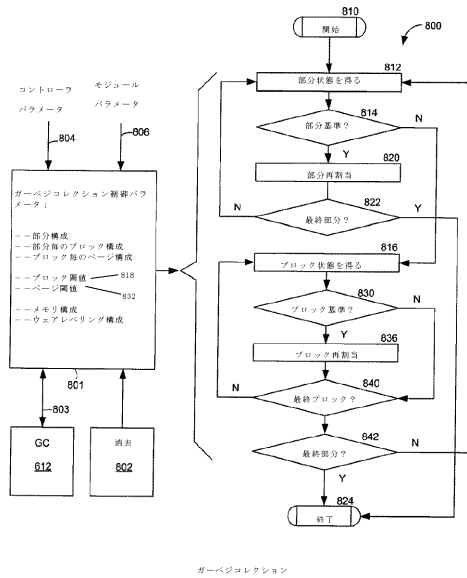
【 図 6 】



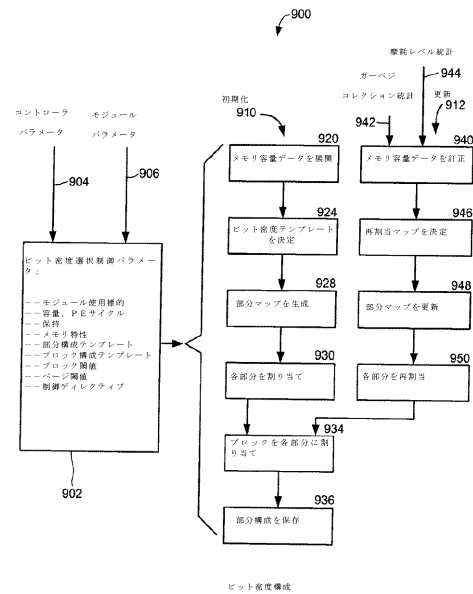
【图 7】



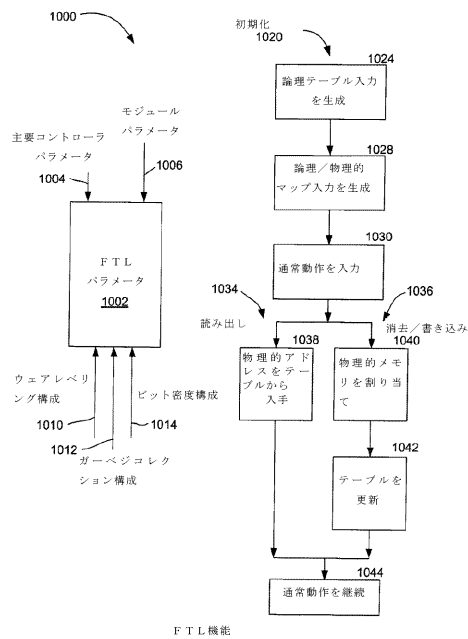
【図 8】



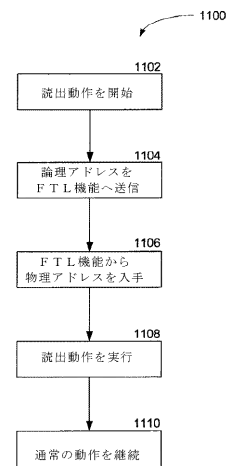
【図 9】



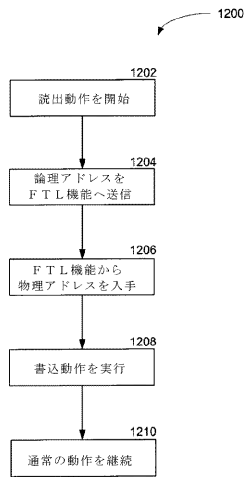
【图 10】



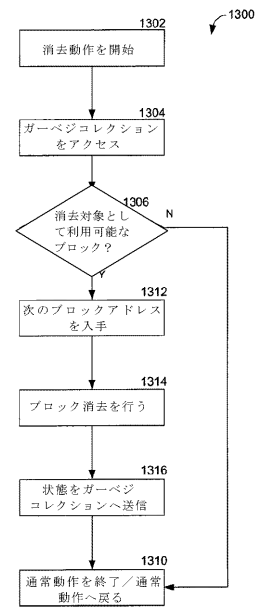
【图 1 1】



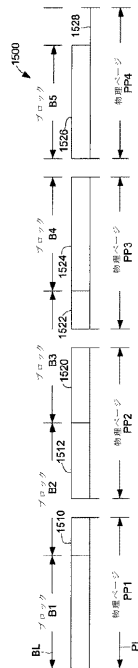
【図 12】



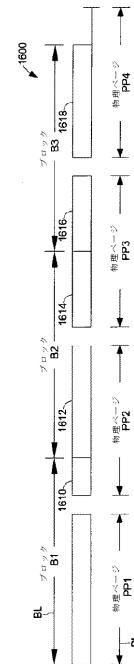
【図 13】



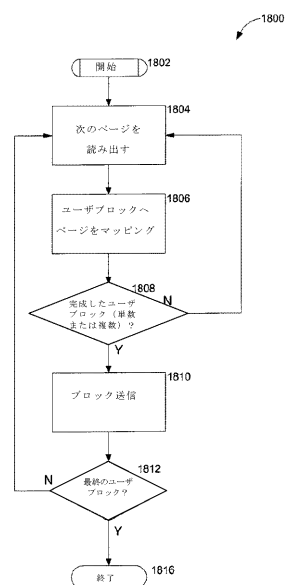
【図 15】



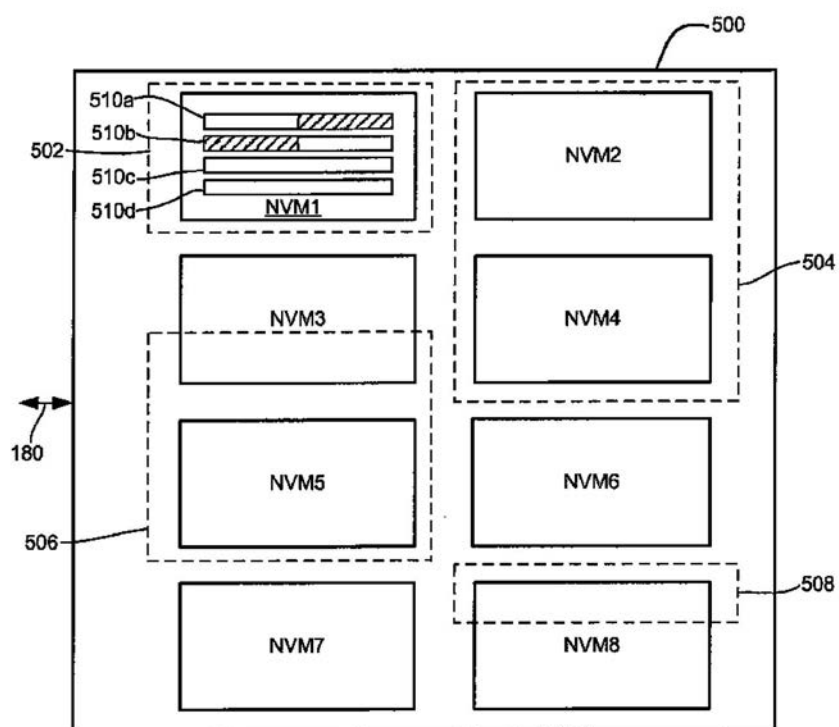
【図 16】



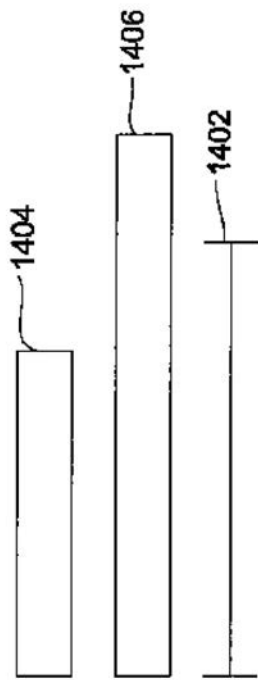
【図 18】



【図 5】



【図 14】



フロントページの続き

(72)発明者 シーベリー, ジョン エル.

アメリカ合衆国, コロラド州 80516, エリー, アン プレイス 4748

Fターム(参考) 5B018 GA04 HA23 MA24 NA06

5B060 AA10 CA12 MM18