

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4675985号
(P4675985)

(45) 発行日 平成23年4月27日(2011.4.27)

(24) 登録日 平成23年2月4日(2011.2.4)

(51) Int.Cl.		F I			
G06F 12/02	12/02	(2006.01)	G06F 12/02	530C	
G06F 12/00	12/00	(2006.01)	G06F 12/00	597U	
G06F 3/08	3/08	(2006.01)	G06F 3/08	H	

請求項の数 11 (全 55 頁)

(21) 出願番号	特願2008-51464 (P2008-51464)	(73) 特許権者	000003078
(22) 出願日	平成20年3月1日(2008.3.1)		株式会社東芝
(65) 公開番号	特開2009-211219 (P2009-211219A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成21年9月17日(2009.9.17)	(74) 代理人	100089118
審査請求日	平成22年2月18日(2010.2.18)		弁理士 酒井 宏明
		(72) 発明者	矢野 純二
			東京都港区芝浦一丁目1番1号 株式会社東芝内
		(72) 発明者	松崎 秀則
			東京都港区芝浦一丁目1番1号 株式会社東芝内
		(72) 発明者	初田 幸輔
			東京都港区芝浦一丁目1番1号 株式会社東芝内

最終頁に続く

(54) 【発明の名称】 メモリシステム

(57) 【特許請求の範囲】

【請求項1】

データ消去単位であるブロックを複数有する不揮発性の半導体記憶素子と、
前記ブロック内の有効なデータを前記ブロックのサイズよりも小さい所定の管理単位で
選択して、異なるブロック内に書き直すコンパクション処理を実行するコントローラと、
を具備し、

前記コントローラは、

前記コンパクション処理を実行する際には、前記ブロックに含まれる有効な前記管理
単位のデータの個数毎に前記ブロックをリンクドリスト形式で管理する管理テーブル内か
ら、前記コンパクション処理の対象とするブロックを検索し、検索したブロックを用いて
コンパクション処理を実行することを特徴とするメモリシステム。

【請求項2】

前記コントローラは、前記コンパクション処理を実行する際には、前記リンクドリスト
に従って前記管理テーブル内から有効な前記管理単位のデータの個数が少ない順番でプロ
ックを検索し、検索したブロックを用いてコンパクション処理を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3】

前記コントローラは、前記リンクドリストの先頭ポインタとして、前記管理テーブル内
の所定のリスト位置に登録されているブロックを設定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】

前記コントローラは、前記コンパクション処理を開始する際に、検索対象とする有効な前記管理単位のデータの個数に応じたブロックを前記管理テーブル内から検索し、その後リンクリストに従って前記管理テーブル内からブロックを検索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 5】

前記コントローラは、前記コンパクション処理を実行する際に、検索したブロック内に含まれる有効な前記管理単位のデータの合計数が前記ブロックのサイズ以上となった場合に、未書き込みのブロックを新たに取得して有効な前記管理単位のデータを書き直す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 6】

前記コントローラは、前記コンパクション処理を実行する際に、検索した前記ブロック内に含まれる有効な前記管理単位のデータの合計数が前記ブロックのサイズに満たない場合に、追記書き込み用の入力バッファに割り当てられているブロックに有効な前記管理単位のデータを書き直す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 7】

前記管理テーブルは、前記ブロックに含まれる有効な前記管理単位のデータの個数毎に、有効な前記管理単位のデータの個数に対応する次のブロックが順番にリンクされた正順方向リンクで繋がっており、

前記コントローラは、前記正順方向リンクを用いて、前記管理テーブル内から前記コンパクション処理の対象とするブロックを検索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。

20

【請求項 8】

前記管理テーブルは、前記ブロックに含まれる有効な前記管理単位のデータの個数毎に、有効な前記管理単位のデータの個数に対応する 1 つ前のブロックが逆順にリンクされた逆順方向リンクで繋がっており、

前記コントローラは、前記ブロックに含まれる有効な前記管理単位のデータの個数が減少した場合に、前記正順方向リンク及び逆順方向リンクを用いて前記管理テーブル内のエントリを繋ぎかえることを特徴とする請求項 7 に記載のメモリシステム。

30

【請求項 9】

前記ブロックは、前記管理単位の自然数倍のサイズであ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 10】

前記ブロックは、書き込み単位である複数のページから構成され、前記管理単位は、前記ページの自然数倍のサイズであることを特徴とする請求項 9 に記載のメモリシステム。

【請求項 11】

前記ブロックは、書き込み単位である複数のページから構成され、前記ページは、前記管理単位の自然数倍のサイズであることを特徴とする請求項 9 に記載のメモリシステム。

【発明の詳細な説明】

【技術分野】

40

【0001】

本発明は、不揮発性半導体メモリを備えたメモリシステムに関する。

【背景技術】

【0002】

コンピュータシステムに用いられる外部記憶装置として、NAND型フラッシュメモリなどの不揮発性半導体メモリを搭載したSSD (Solid State Drive) が注目されている。フラッシュメモリは、磁気ディスク装置に比べ、高速、軽量などの利点を有している。

【0003】

SSD内には、複数のフラッシュメモリチップ、ホスト装置からの要求に応じて各フラッシュメモリチップのリード/ライト制御を行うコントローラ、各フラッシュメモリチッ

50

プとホスト装置との間でデータ転送を行うためのバッファメモリ、電源回路、ホスト装置に対する接続インタフェースなどを備えている（例えば、特許文献1）。

【0004】

不揮発性半導体メモリには、NAND型フラッシュメモリのように、データを記憶させる場合にブロック単位で一度データを消去してからその後に書き込みを行うもの、ページ単位で書き込み/読み出しを行うものなど、消去/書き込み/読み出しの単位が固定されているものがある。

【0005】

一方、パーソナルコンピュータなどのホスト機器がハードディスクをはじめとする2次記憶装置に対してデータの書き込み/読み出しを行う単位は、セクタと呼ばれる。セクタは、半導体記憶装置の消去/書き込み/読み出しの単位とは独立に定められる。

10

【0006】

例えば、不揮発性半導体メモリのブロックの大きさ（ブロックサイズ）は、512kB、ページの大きさ（ページサイズ）は、4kBであるのに対して、ホスト機器のセクタの大きさ（セクタサイズ）は、512Bのように定められている。

【0007】

このように、不揮発性半導体メモリの消去/書き込み/読み出しの単位は、ホスト機器の書き込み/読み出しの単位よりも大きい場合がある。

【0008】

そこで、不揮発性半導体メモリを用いてハードディスクのようなパーソナルコンピュータの2次記憶装置を構成する場合、ホスト機器としてのパーソナルコンピュータからの小さなサイズのデータは、不揮発性半導体メモリのブロックサイズ、ページサイズに適合させて書き込みを行う必要がある。

20

【0009】

また、パーソナルコンピュータなどのホスト機器が記録するデータは、時間的局所性、及び領域的局所性を兼ね備えている（例えば、非特許文献1参照。）。そのため、データを記録する際に外部から指定されたアドレスにそのまま記録していくと、特定の領域に短時間に書き換え、すなわち消去処理が集中し、消去回数の偏りが大きくなる。そのため、NAND型フラッシュメモリでは、データ更新箇所を均等に分散させるウェアレベリングと呼ばれる処理が行われる。

30

【0010】

ウェアレベリング処理では、例えば、ホスト機器から指定される論理アドレスを、データ更新箇所が均等に分散された不揮発性半導体メモリの物理アドレスにアドレス変換している。

【0011】

NAND型フラッシュメモリを用いて大容量の2次記憶装置を構成する場合においては、上記アドレス変換を行う場合に、データ管理の単位が小さいサイズ（例えば、ページサイズ）であると、管理テーブルのサイズが大きくなって、2次記憶装置のコントローラの主記憶メモリに納まらなくなり、アドレス変換が高速にできなくなるという問題がある。このように、2次記憶装置としてのNAND型フラッシュメモリの大容量化に伴い管理テーブルのサイズは必然的に大きくなるため、管理テーブルをできるだけ小容量にするための手法が求められている。

40

【0012】

また、上述したように、データの消去単位（ブロック）と、データの管理単位が異なる場合、フラッシュメモリの書き換えが進むと、無効な（最新ではない）データによって、ブロックは穴あき状態になる。このような穴あき状態のブロックが増えると、実質的に使用可能なブロックが少なくなり、フラッシュメモリの記憶領域を有効利用できない。これに対応して、有効な最新のデータを集めて、違うブロックに書き直すコンパクションと呼ばれる処理が行われることがある（例えば、特許文献2参照。）。このコンパクション処理においては、その手法によっては処理時間が長くなるので、コンパクションの処理時間

50

を短くするための手法が求められている。

【 0 0 1 3 】

【特許文献 1】特許第 3 6 8 8 8 3 5 号公報

【特許文献 2】特開 2 0 0 5 - 2 2 2 5 5 0 号公報

【非特許文献 1】David A. Patterson and John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Morgan Kaufmann Pub, 2004/8/31

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 4 】

本発明は、コンパクションの処理時間を短縮化することが可能なメモリシステムを提供する。 10

【課題を解決するための手段】

【 0 0 1 5 】

本願発明の一態様に係るメモリシステムは、データ消去単位であるブロックを複数有する不揮発性の半導体記憶素子と、前記ブロック内の有効なデータを前記ブロックのサイズよりも小さい所定の管理単位で選択して、異なるブロック内に書き直すコンパクション処理を実行するコントローラと、を具備し、前記コントローラは、前記コンパクション処理を実行する際には、前記ブロックに含まれる有効な前記管理単位のデータの個数毎に前記ブロックをリンクドリスト形式で管理する管理テーブル内から、前記コンパクション処理の対象とするブロックを検索し、検索したブロックを用いてコンパクション処理を実行することを特徴とする。 20

【発明の効果】

【 0 0 1 6 】

本発明によれば、コンパクションの処理時間を短縮化することが可能なメモリシステムを提供できる。

【発明を実施するための最良の形態】

【 0 0 1 7 】

以下、本発明の実施の形態について図面を参照して説明する。なお、以下の説明において、同一の機能および構成を有する要素については、同一符号を付し、重複説明は必要な場合にのみ行う。 30

【 0 0 1 8 】

先ず、本明細書で用いる用語について定義しておく。

・物理ページ：NANDメモリチップ内部において一括して書き込み/読み出しが可能な単位のこと。物理ページサイズは、例えば 4 k B。ただし、主データ（ユーザデータなど）に対して S S D 内で付加される誤り訂正符号などの冗長ビットは含まないものとする。通常、4 k B + 冗長ビット（例えば、数 1 0 B）が同時にメモリセルに書き込まれる単位となるが、説明の便宜上、上記のように定義する。

・論理ページ：S S D 内で設定される書き込み/読み出し単位であり、1 以上の物理ページに対応付けられている。論理ページサイズは、例えば 8 ビットノーマルモードでは、4 k B、3 2 ビット倍速モードでは、3 2 k B。ただし、冗長ビットは含まないものとする。 40

・物理ブロック：NANDメモリチップ内部において独立して消去可能な最小単位のことであり、複数の物理ページから構成される。物理ブロックサイズは、例えば 5 1 2 k B。ただし、主データに対して S S D 内で付加される誤り訂正符号などの冗長ビットは含まないものとする。通常、5 1 2 k B + 冗長ビット（例えば、数 1 0 k B）が同時に消去される単位となるが、説明の便宜上、上記のように定義する。

・論理ブロック：S S D 内で設定される消去単位であり、1 以上の物理ブロックに対応付けられている。論理ブロックサイズは、例えば 8 ビットノーマルモードでは、5 1 2 k B、3 2 ビット倍速モードでは、4 M B。ただし、冗長ビットは含まないものとする。

・セクタ：ホストからの最小アクセス単位のこと。セクタサイズは、例えば 5 1 2 B。 50

・クラスタ：SSD内で「小さなデータ」を管理する管理単位。クラスタサイズはセクタサイズ以上であり、クラスタサイズの2以上の自然数倍が論理ページサイズとなるように定められる。

・トラック：SSD内で「大きなデータ」を管理する管理単位。クラスタサイズの2以上の自然数倍がトラックサイズに、かつ、トラックサイズの2以上の自然数倍が論理ブロックサイズとなるように定められる。

・フリーブロック（FB）：用途未割り当てのNAND型フラッシュメモリ上の論理ブロックのこと。用途を割り当てる際に消去してから使用する。

・バッドブロック（BB）：NAND型フラッシュメモリ上の、誤りが多いなど記憶領域として使用できない物理ブロックのこと。例えば、消去動作が正常に終了しなかった物理ブロックがバッドブロックBBとして登録される。

・書き込み効率：所定期間内における、ホストから書き込んだデータ量に対する、論理ブロックの消去量の統計値のこと。小さいほどNAND型フラッシュメモリの消耗度が小さい。

・有効クラスタ：最新のデータを保持しているクラスタ。

・無効クラスタ：最新ではないデータを保持しているクラスタ。

・有効トラック：最新のデータを保持しているトラック。

・無効トラック：最新ではないデータを保持しているトラック。

・コンパクション：管理対象内の論理ブロックから、有効クラスタや有効トラックのみを取り出して、新しい論理ブロックに書き直すこと。

【0019】

[第1の実施形態]

図1は、SSD (Solid State Drive) 100の構成例を示すブロック図である。SSD 100は、ATAインタフェース (ATA I/F) 2などのメモリ接続インタフェースを介してパーソナルコンピュータあるいはCPUコアなどのホスト装置1と接続され、ホスト装置1の外部メモリとして機能する。また、SSD 100は、RS232Cインタフェース (RS232C I/F) 3などの通信インタフェースを介して、デバッグ用/製造検査用機器200との間でデータを送受信することができる。SSD 100は、不揮発性半導体メモリとしてのNAND型フラッシュメモリ (以下、NANDメモリと略す) 10と、コントローラとしてのドライブ制御回路4と、揮発性半導体メモリとしてのDRAM 20と、電源回路5と、状態表示用のLED 6と、ドライブ内部の温度を検出する温度センサ7と、フューズ8とを備えている。

【0020】

電源回路5は、ホスト装置1側の電源回路から供給される外部直流電源から複数の異なる内部直流電源電圧を生成し、これら内部直流電源電圧をSSD 100内の各回路に供給する。また、電源回路5は、外部電源の立ち上がりまたは立ち下がりを検知し、パワーオンリセット信号を生成して、ドライブ制御回路4に供給する。フューズ8は、ホスト装置1側の電源回路とSSD 100内部の電源回路5との間に設けられている。外部電源回路から過電流が供給された場合フューズ8が切断され、内部回路の誤動作を防止する。

【0021】

NANDメモリ10は、この場合、4並列動作を行う4つの並列動作要素10a~10dを有し、1つの並列動作要素は、2つのNANDメモリパッケージを有する。各NANDメモリパッケージは、積層された複数のNANDメモリチップ (例えば、1チップ=2GB) によって構成されている。図1の場合は、各NANDメモリパッケージは、積層された4枚のNANDメモリチップによって構成されており、NANDメモリ10は64GBの容量を有する。各NANDメモリパッケージが、積層された8枚のNANDメモリチップによって構成されている場合は、NANDメモリ10は128GBの容量を有することになる。

【0022】

DRAM 20は、ホスト装置1とNANDメモリ10間でのデータ転送用キャッシュお

10

20

30

40

50

よび作業領域用メモリとして機能する。また、DRAM 20の代わりに、FeRAMを使用しても良い。ドライブ制御回路4は、ホスト装置1とNANDメモリ10との間でDRAM 20を介してデータ転送制御を行うとともに、SSD 100内の各構成要素を制御する。また、ドライブ制御回路4は、状態表示用LED 6にステータス表示用信号を供給するとともに、電源回路5からのパワーオンリセット信号を受けて、リセット信号およびクロック信号を自回路内およびSSD 100内の各部に供給する機能も有している。

【0023】

各NANDメモリチップは、データ消去の単位である物理ブロックを複数配列して構成されている。図2(a)は、NANDメモリチップに含まれる1個の物理ブロックの構成例を示す回路図である。各物理ブロックは、X方向に沿って順に配列された(p+1)個のNANDストリングを備えている(pは、0以上の整数)。(p+1)個のNANDストリングにそれぞれ含まれる選択トランジスタST1は、ドレインがビット線BL0~BLpに接続され、ゲートが選択ゲート線SGDに共通接続されている。また、選択トランジスタST2は、ソースがソース線SLに共通接続され、ゲートが選択ゲート線SGSに共通接続されている。

10

【0024】

各メモリセルトランジスタMTは、半導体基板上に形成された積層ゲート構造を備えたMOSFET(Metal Oxide Semiconductor Field Effect Transistor)から構成される。積層ゲート構造は、半導体基板上にゲート絶縁膜を介在して形成された電荷蓄積層(浮遊ゲート電極)、および電荷蓄積層上にゲート間絶縁膜を介在して形成された制御ゲート電極を含んでいる。メモリセルトランジスタMTは、浮遊ゲート電極に蓄えられる電子の数に応じて閾値電圧が変化し、この閾値電圧の違いに応じてデータを記憶する。メモリセルトランジスタMTは、1ビットを記憶するように構成されていてもよいし、多値(2ビット以上のデータ)を記憶するように構成されていてもよい。

20

【0025】

また、メモリセルトランジスタMTは、浮遊ゲート電極を有する構造に限らず、MONOS(Metal-Oxide-Nitride-Oxide-Silicon)型など、電荷蓄積層としての窒化膜界面に電子をトラップさせることでしきい値調整可能な構造であってもよい。MONOS構造のメモリセルトランジスタMTについても同様に、1ビットを記憶するように構成されていてもよいし、多値(2ビット以上のデータ)を記憶するように構成されていてもよい。

30

【0026】

各NANDストリングにおいて、(q+1)個のメモリセルトランジスタMTは、選択トランジスタST1のソースと選択トランジスタST2のドレインとの間に、それぞれの電流経路が直列接続されるように配置されている。すなわち、複数のメモリセルトランジスタMTは、隣接するもの同士で拡散領域(ソース領域若しくはドレイン領域)を共有するような形でY方向に直列接続される。

【0027】

そして、最もドレイン側に位置するメモリセルトランジスタMTから順に、制御ゲート電極がワード線WL0~WLqにそれぞれ接続されている。従って、ワード線WL0に接続されたメモリセルトランジスタMTのドレインは選択トランジスタST1のソースに接続され、ワード線WLqに接続されたメモリセルトランジスタMTのソースは選択トランジスタST2のドレインに接続されている。

40

【0028】

ワード線WL0~WLqは、物理ブロック内のNANDストリング間で、メモリセルトランジスタMTの制御ゲート電極を共通に接続している。つまり、ブロック内において同一行にあるメモリセルトランジスタMTの制御ゲート電極は、同一のワード線WLに接続される。この同一のワード線WLに接続される(p+1)個のメモリセルトランジスタMTは1ページ(物理ページ)として取り扱われ、この物理ページごとにデータの書き込みおよびデータの読み出しが行われる。

【0029】

50

また、ビット線 $B L 0 \sim B L p$ は、ブロック間で、選択トランジスタ $S T 1$ のドレインを共通に接続している。つまり、複数のブロック内において同一列にある $N A N D$ ストリングは、同一のビット線 $B L$ に接続される。

【 0 0 3 0 】

図 2 (b) は、例えば、1 個のメモリセルトランジスタ $M T$ に 2 ビットの記憶を行う 4 値データ記憶方式でのしきい値分布を示す模式図である。4 値データ記憶方式では、上位ページデータ “ x ” と下位ページデータ “ y ” で定義される 4 値データ “ x y ” の何れか 1 つをメモリセルトランジスタ $M T$ に保持可能である。

【 0 0 3 1 】

この、4 値データ “ x y ” は、メモリセルトランジスタ $M T$ のしきい値電圧の順に、例えば、データ “ 1 1 ”、“ 0 1 ”、“ 0 0 ”、“ 1 0 ” が割り当てられる。データ “ 1 1 ” は、メモリセルトランジスタ $M T$ のしきい値電圧が負の消去状態である。

10

【 0 0 3 2 】

下位ページ書き込み動作においては、データ “ 1 1 ” (消去状態) のメモリセルトランジスタ $M T$ に対して選択的に、下位ビットデータ “ y ” の書き込みによって、データ “ 1 0 ” が書き込まれる。上位ページ書き込み前のデータ “ 1 0 ” のしきい値分布は、上位ページ書き込み後のデータ “ 0 1 ” とデータ “ 0 0 ” のしきい値分布の中間程度に位置しており、上位ページ書き込み後のしきい値分布よりブロードであってもよい。上位ページ書き込み動作においては、データ “ 1 1 ” のメモリセルと、データ “ 1 0 ” のメモリセルに対して、それぞれ選択的に上位ビットデータ “ x ” の書き込みが行われて、データ “ 0 1 ” およびデータ “ 0 0 ” が書き込まれる。

20

【 0 0 3 3 】

図 3 は、ドライブ制御回路 4 のハードウェア的な内部構成例を示すブロック図である。ドライブ制御回路 4 は、データアクセス用バス 1 0 1、第 1 の回路制御用バス 1 0 2、および第 2 の回路制御用バス 1 0 3 を備えている。第 1 の回路制御用バス 1 0 2 には、ドライブ制御回路 4 全体を制御するプロセッサ 1 0 4 が接続されている。第 1 の回路制御用バス 1 0 2 には、 $N A N D$ メモリ 1 0 に記憶された各管理プログラム ($F W$: ファームウェア) をブートするブート用プログラムが格納されたブート $R O M$ 1 0 5 が $R O M$ コントローラ 1 0 6 を介して接続されている。また、第 1 の回路制御用バス 1 0 2 には、図 1 に示した電源回路 5 からのパワーオンリセット信号を受けて、リセット信号およびクロック信号を各部に供給するクロックコントローラ 1 0 7 が接続されている。

30

【 0 0 3 4 】

第 2 の回路制御用バス 1 0 3 は、第 1 の回路制御用バス 1 0 2 に接続されている。第 2 の回路制御用バス 1 0 3 には、図 1 に示した温度センサ 7 からのデータを受け取るための $I^2 C$ 回路 1 0 8、状態表示用 $L E D$ 6 にステータス表示用信号を供給するパラレル $I O$ ($P I O$) 回路 1 0 9、 $R S 2 3 2 C$ I / F 3 を制御するシリアル $I O$ ($S I O$) 回路 1 1 0 が接続されている。

【 0 0 3 5 】

$A T A$ インタフェースコントローラ ($A T A$ コントローラ) 1 1 1、第 1 の $E C C$ ($E r r o r$ $C h e c k i n g$ $a n d$ $C o r r e c t i o n$) 回路 1 1 2、 $N A N D$ コントローラ 1 1 3、および $D R A M$ コントローラ 1 1 4 は、データアクセス用バス 1 0 1 と第 1 の回路制御用バス 1 0 2 との両方に接続されている。 $A T A$ コントローラ 1 1 1 は、 $A T A$ インタフェース 2 を介してホスト装置 1 との間でデータを送受信する。データアクセス用バス 1 0 1 には、データ作業領域作業領域およびファームウェア展開領域として使用される $S R A M$ 1 1 5 が $S R A M$ コントローラ 1 1 6 を介して接続されている。 $N A N D$ メモリ 1 0 に記憶されているファームウェアは起動時、ブート $R O M$ 1 0 5 に記憶されたブート用プログラムによって $S R A M$ 1 1 5 に転送される。

40

【 0 0 3 6 】

$N A N D$ コントローラ 1 1 3 は、 $N A N D$ メモリ 1 0 とのインタフェース処理を行う $N A N D$ I / F 1 1 7、第 2 の $E C C$ 回路 1 1 8、および $N A N D$ メモリ 1 0 - $D R A M$

50

20間のアクセス制御を行うDMA転送制御用DMAコントローラ119を備えている。第2のECC回路118は第2の訂正符号のエンコードを行い、また、第1の誤り訂正符号のエンコードおよびデコードを行う。第1のECC回路112は、第2の誤り訂正符号のデコードを行う。第1の誤り訂正符号、第2の誤り訂正符号は、例えば、ハミング符号、BCH (Bose Chaudhuri Hocqenghem) 符号、RS (Reed Solomon) 符号、或いはLDP C (Low Density Parity Check) 符号等であり、第2の誤り訂正符号の訂正能力は、第1の誤り訂正符号の訂正能力よりも高いとする。

【0037】

図1および図3に示したように、NANDメモリ10においては、4つの並列動作要素10a~10dが各8ビットの4チャンネル(4ch)を介して、ドライブ制御回路4内部のNANDコントローラ112に並列接続されている。4つの並列動作要素10a~10dを単独動作させるか、並列動作させるか、NANDメモリチップの備える倍速モード(Multi Page Program / Multi Page Read / Multi Block Erase)を使用するか否か、という組み合わせにより、下記3種類のアクセスモードが提供される。

(1) 8ビットノーマルモード

1chだけ動作させ、8ビット単位でデータ転送を行うモードである。物理ページサイズ(4kB)で書き込み/読み出しが行われる。また、物理ブロックサイズ(512kB)で消去が行われる。1つの物理ブロックに対して1つの論理ブロックが対応付けられ、論理ブロックサイズは512kBとなる。

(2) 32ビットノーマルモード

4ch並列で動作させ、32ビット単位でデータ転送を行うモードである。物理ページサイズ×4(16kB)で書き込み/読み出しが行われる。また、物理ブロックサイズ×4(2MB)で消去が行われる。4つの物理ブロックに対して1つの論理ブロックが対応付けられ、論理ブロックサイズは2MBとなる。

(3) 32ビット倍速モード

4ch並列で動作させ、更に、NANDメモリチップの倍速モードを利用して書き込み/読み出しを行うモードである。物理ページサイズ×4×2(32kB)で書き込み/読み出しが行われる。また、物理ブロックサイズ×4×2(4MB)で消去が行われる。8つの物理ブロックに対して1つの論理ブロックが対応付けられ、論理ブロックサイズは4MBとなる。

【0038】

4ch並列動作する32ビットノーマルモードまたは32ビット倍速モードでは、並列動作する4または8物理ブロックが、NANDメモリ10としての消去単位となり、並列動作する4または8物理ページが、NANDメモリ10としての書き込み単位および読み出し単位となる。以下の動作では、基本的に32ビット倍速モードを使用し、例えば、1論理ブロック=4MB= $\frac{2^i}{2^j}$ トラック= $\frac{2^i}{2^k}$ ページ= $\frac{2^i}{2^l}$ クラスタ= $\frac{2^i}{2^l}$ セクタとして説明する(i、j、k、lは自然数、かつ、 $i < j < k < l$ の関係が成立する)。

【0039】

32ビット倍速モードでアクセスされる論理ブロックは4MB単位であり、8個(2×4ch)の物理ブロック(1物理ブロック=512KB)が対応付けられている。物理ブロック単位で管理されるバッドブロックBBが発生すると、そのバッドブロックBBは使用不可になるので、そのようなときには、論理ブロックに対応付けられた8個の物理ブロックの組み合わせが、バッドブロックBBを含まないように変更される。

【0040】

図4は、プロセッサ104により実現されるファームウェアの機能構成例を示すブロック図である。プロセッサ104により実現されるファームウェアの各機能は、大きく、データ管理部120、ATAコマンド処理部121、セキュリティ管理部122、ブートローダ123、初期化管理部124、デバッグサポート部125に分類される。

【0041】

データ管理部120は、NANDコントローラ112、第1のECC回路114を介し

10

20

30

40

50

て、NANDメモリ10 - DRAM20間のデータ転送、NANDメモリ10に関する各種機能を制御する。ATAコマンド処理部121は、ATAコントローラ110、およびDRAMコントローラ113を介して、データ管理部120と協働してDRAM20 - ホスト装置1間のデータ転送処理を行う。セキュリティ管理部122は、データ管理部120およびATAコマンド処理部121と協働して各種のセキュリティ情報を管理する。

【0042】

ブートローダ123は、パワーオン時、各管理プログラム（ファームウェア）をNANDメモリ10からSRAM120にロードする。初期化管理部124は、ドライブ制御回路4内の各コントローラ/回路の初期化を行う。デバッグサポート部125は、外部からRS232Cインタフェースを介して供給されたデバッグ用データを処理する。主に、データ管理部120、ATAコマンド処理部121、およびセキュリティ管理部122が、SRAM114に記憶される各管理プログラムをプロセッサ104が実行することによって実現される機能部である。

【0043】

本実施形態では、主としてデータ管理部120が実現する機能について説明する。データ管理部120は、ATAコマンド処理部121が記憶デバイスであるNANDメモリ10やDRAM20に対して要求する機能の提供（ホスト装置からのWrite要求、Cache Flush要求、Read要求等の各種コマンドへの応答）と、アドレス領域とNANDメモリ10との対応関係の管理および管理情報の保護と、DRAM10およびNANDメモリ10を利用した高速で効率の良いデータ読み出し/書き込み機能の提供、NANDメモリ10の信頼性の確保などを行う。

【0044】

図5は、NANDメモリ10およびDRAM20内に形成された機能ブロックを示すものである。ホスト1とNANDメモリ10の間には、DRAM20上に構成されたライトキャッシュ（WC）21およびリードキャッシュ（RC）22が介在している。WC21はホスト装置1からのWriteデータを一時保存し、RC22はNANDメモリ10からのReadデータを一時保存する。NANDメモリ10内の論理ブロックは、書き込み時のNANDメモリ10に対する消去の量を減らすために、データ管理部120により、前段ストレージ領域（FS：Front Storage）12、中段ストレージ領域（IS：Intermediate Storage）13およびメインストレージ領域（MS：Main Storage）11という各管理領域に割り当てられている。FS12は、WC21からのデータを「小さな単位」であるクラスタ単位に管理するものであり、小データを短期間保存する。IS13は、FS12から溢れたデータを「小さな単位」であるクラスタ単位に管理するものであり、小データを長期間保存する。MS11は、WC21、FS12、IS13からのデータを「大きな単位」であるトラック単位で長期間記憶する。例えば、記憶容量は、MS > IS、FS > WCの関係となる。

【0045】

小さな管理単位を、NANDメモリ10の記憶領域全てに適用すると、後述する管理テーブルのサイズが肥大化し、DRAM20に収まらないので、小さな管理単位で管理するのは、最近書き込まれたばかりのデータと、NANDメモリ10への書き込み効率が悪い小さなデータのみとするようにNANDメモリ10の各ストレージを構成している。

【0046】

図6は、WC21からNANDメモリ10への書き込み処理（WR処理）に係わるより詳細な機能ブロック図を示すものである。FS12の前段には、WC21からのデータをバッファリングするFSインプットバッファ（FSIB）12aが設けられている。また、MS11の前段には、WC21、FS12、またはIS13からのデータをバッファリングするMSインプットバッファ（MSIB）11aが設けられている。また、MS11には、トラック前段ストレージ領域（TF S）11bが設けられている。TF S 11bは、MSIB 11aとMS11の間に介在するFIFO（First in First out）構造を有するバッファであり、TF S 11bに記録されたデータは、MSIB 11aから直接MS1

10

20

30

40

50

1 に書き込まれるデータよりも更新頻度が高いデータである。MS 1 1、MS I B 1 1 a、T F S 1 1 b、F S 1 2、F S I B 1 2 a、および I S 1 3 には、N A N D メモリ 1 0 内の各論理ブロックの何れかが割り当てられている。

【 0 0 4 7 】

つぎに、図 5、図 6 の各構成要素の具体的な機能構成について詳述する。ホスト装置 1 は S S D 1 0 0 対し、Read または Write する際には、A T A インタフェースを介して論理アドレスとしての L B A (Logical Block Addressing) を入力する。L B A は、図 7 に示すように、セクタ (サイズ: 5 1 2 B) に対して 0 からの通し番号をつけた論理アドレスである。本実施の形態においては、図 5 の各構成要素である W C 2 1、R C 2 2、F S 1 2、I S 1 3、M S 1 1 の管理単位として、L B A の下位 (1 - k + 1) ビット目から上位のビット列で構成される論理クラスタアドレスと、L B A の下位 (1 - i + 1) ビットから上位のビット列で構成される論理トラックアドレスとを定義する。1 クラスタ = $\frac{2^{(1-k)}}{2^{(k-i)}}$ セクタで、1 トラック = $\frac{2^{(k-i)}}{2^{(k-i)}}$ クラスタである。

10

【 0 0 4 8 】

・リードキャッシュ (R C) 2 2

R C 2 2 について説明する。R C 2 2 は、A T A コマンド処理部 1 2 1 からの Read 要求に対して、N A N D メモリ 1 0 (F S 1 2、I S 1 3、M S 1 1) からの Read データを一時的に保存するための領域である。R C 2 2 は、本実施形態では例えば、m-line、n-way (m は $\frac{2^{(k-i)}}{2^{(k-i)}}$ 以上の自然数、n は 2 以上の自然数) セットアソシアティブ方式で管理されており、1 エントリに 1 クラスタ分のデータを保持できる。論理クラスタアドレスの L S B (k - i) ビットで line が決定される。なお、R C 2 2 は、フルアソシアティブ方式で管理されていても良いし、単純な F I F O 方式で管理されていてもよい。

20

【 0 0 4 9 】

・ライトキャッシュ (W C) 2 1

W C 2 1 について説明する。W C 2 1 は、A T A コマンド処理部 1 2 1 からの Write 要求に対して、ホスト装置 1 からの Write データを一時的に保存するための領域である。m-line、n-way (m は $\frac{2^{(k-i)}}{2^{(k-i)}}$ 以上の自然数、n は 2 以上の自然数) セットアソシアティブ方式で管理されており、1 エントリに 1 クラスタ分のデータを保持できる。論理クラスタアドレスの L S B (k - i) ビットで line が決定される。例えば、way 1 ~ way n の順で書き込み可能な way が検索される。また、W C 2 1 に登録されているトラックは最も古く更新された順が分かるように後述する W C トラック管理テーブル 2 4 の F I F O 構造によって L R U (Least Recently Used) で管理される。なお、W C 2 1 は、フルアソシアティブ方式で管理されていても良い。また、W C 2 1 は、R C 2 2 と line 数、way 数が互いに異なってもよい。

30

【 0 0 5 0 】

Write 要求により書き込まれたデータは、一旦 W C 2 1 上に格納される。W C 2 1 から N A N D メモリ 1 0 へ追い出すデータの決定方法は以下のルールに従う。

(i) タグによって決定された line の書き込み可能な way が最後の (本実施形態では、n 個目の) 空き way だった場合、則ち最後の空き way が使用される場合は、その line に登録されたトラックのうち、L R U に基づいて最も古く更新されたトラックを追い出し確定する。

40

(ii) W C 2 1 に登録されている異なるトラックの個数が所定数を越えた場合、L R U 順で、当該トラックに属する W C 中のクラスタ数が所定数未満のトラックの追い出しを確定する。

【 0 0 5 1 】

以上の方針で追い出すトラックを決定する。その際、追い出すのは同一トラックに含まれる全てのデータであり、追い出されるデータ量が、例えばトラックサイズの 5 0 % を超えていれば M S 1 1 へ、超えていなければ F S 1 2 へ追い出す。

【 0 0 5 2 】

さらに (i) の条件でトラック追い出しが発生した場合で、M S 1 1 へ追い出す場合は

50

、追い出されるトラック数が 2^i 個（もともと 2^i 個以上のときは 2^{i+1} 個）になるまで、WC 2 1 内のトラックのうち上記追い出されるデータ量がトラックサイズの 50% を超えるという条件を満たすトラックを上記 (i) のポリシーで選択して追い出し候補に追加する。別言すれば、追い出されるトラックが 2^i 個未満の場合、WC 中のトラックの古いものから 2^i 個になるまで、 $2^{(k-i-1)}$ 個以上有効クラスタをもつトラックを選択して追い出し候補に追加する。

【0053】

また、(i) の条件でトラック追い出しが発生した場合で、FS 1 2 に追い出す場合は、追い出されるクラスタ数が 2^k 個になるまで WC 2 1 内のトラックのうち LRU 順に上記追い出されるデータ量がトラックサイズの 50% 未満であるという条件を満たすトラックを探してそのクラスタを追い出し候補に追加する。別言すれば、WC 中のトラックを古い順に辿って $2^{(k-i-1)}$ 個未満の有効クラスタしかもたないトラックからクラスタを取り出していき、有効クラスタ数が $2^{(k-i-1)}$ 個になったら、それらクラスタを FSIB 1 2 a に論理ブロック単位で追い出しする。ただし、 $2^{(k-i-1)}$ 個見つからなかった場合は、FSIB 1 2 a に論理ページ単位で追い出しする。なお、FS 1 2 への追い出しを論理ブロック単位とするか、論理ページ単位とするかの有効クラスタ数の閾値は、 $2^{(k-i-1)}$ 個という 1 論理ブロック分の値にかぎるわけではなく、1 論理ブロック分より若干少ない値であってもよい。

【0054】

また、ATA コマンド処理部 1 2 1 からの Cache Flush 要求では、WC 2 1 の内容が全て、上記と同じ条件（追い出されるデータ量がトラックサイズの 50% を超えていれば MS 1 1 へ、超えていなければ FS 1 2 へ）で、FS 1 2 もしくは MS 1 1 に対して追い出される。

【0055】

・前段ストレージ領域 (FS) 1 2

つぎに、FS 1 2 について説明する。FS 1 2 はクラスタ単位でデータを管理される FIFO である。FS 1 2 は、ここを通過しているデータは、後段の IS 1 3 よりも更新頻度が高いとみなすためのバッファである。すなわち、FS 1 2 の FIFO 構造においては、FIFO 中を通過中の有効クラスタ（最新クラスタ）は、ホストからの同じアドレスに対する再書き込みがあった場合無効化されるので、FS 1 2 を通過中のクラスタは、FS 1 2 から IS 1 3 や MS 1 1 に追い出されたクラスタよりも、更新頻度が高いとみなすことができる。

【0056】

FS 1 2 を設けることで、後段の IS 1 3 におけるコンパクション処理に更新頻度の高いデータが紛れ込む可能性を低減している。無効化によって古いクラスタを保持していた論理ブロック自体の持つ有効クラスタ数が 0 となった場合、その論理ブロックは開放され、フリーブロック FB に割り当てられる。また、論理ブロックが無効化された場合、新たなフリーブロック FB を取得し、FS 1 2 に割り当てる。

【0057】

WC 2 1 から FS 1 2 に対してクラスタデータの移動が発生すると、そのクラスタは FSIB 1 2 a に割り当てられた論理ブロックに対して書き込まれる。FSIB 1 2 a 中に全てのページの書き込みが完了したブロックが存在する場合、後述する CIB 処理によってそれらのブロックは FSIB 1 2 a から FS 1 2 に Move される。この FSIB 1 2 a から FS 1 2 への Move の際に、FS 1 2 のブロック数が FS 1 2 として許容される所定の上限値を超えると、最も古いブロックが FS 1 2 から IS 1 3 または MS 1 1 に追い出されることになる。例えば、トラック内の有効クラスタの割合が 50% 以上のトラックは、MS 1 1 (TFS 1 1 b) への書き込みを行い、有効クラスタが残ったブロックを、IS 1 3 へ Move する。

【0058】

NAND メモリ 1 0 内の構成要素間のデータ移動には、Move と Copy の二通りが

10

20

30

40

50

ある。Moveは、後述する管理テーブルのポインタの付け替えを行うだけで、実際のデータの書き換えは行わない方法である。Copyは、一方の構成要素に格納されているデータを、ページ単位、トラック単位、ブロック単位で他方の構成要素に実際に書き換える方法である。

【0059】

・中段ストレージ領域 (IS) 13

つぎに、IS13について説明する。IS13は、FS13と同様にクラスタ単位でデータの管理が行われる。前述したように、IS13に格納されたデータは、更新頻度が低いデータとみなすことができる。FS12からIS13に対して論理ブロックの移動 (Move)、すなわちFS12からの追い出しが発生すると、以前FS12の管理対象であった追い出し対象の論理ブロックはポインタの付け替えによりIS13の管理対象ブロックとなる。このFS12からIS13への論理ブロックの移動により、IS13のブロック数がIS13として許容される所定の上限値を超えると、すなわちIS内の書き込み可能なフリーブロックFBの数が閾値を下回ると、IS13からMS11へのデータ追い出しおよびコンパクション処理が実行され、IS13のブロック数は規定値に戻される。

【0060】

IS13では、トラック内の有効クラスタ数を使って以下のような、追い出し処理およびコンパクション処理を、実行する。

・トラックをトラック内の有効クラスタ数×有効クラスタ係数(トラックがMS11内で無効トラックが存在する論理ブロックに存在するか否かによって重み付けされる数であり、存在したほうが存在しない場合より数が大きい)順にソートし、積の値が大きいトラック 2^{i+1} 個 (2論理ブロック分) を集めて論理ブロックサイズの自然数倍にしてMSIB11aに追い出す。

・有効クラスタ数が最も少ない2つの論理ブロックの合計有効クラスタ数が例えば、所定の設定値である 2^k 個 (1論理ブロック分) 以上ある場合は、上のステップを繰り返す (IS内の2つの論理ブロックから、フリーブロックFBを作れるようになるまで行うため)。

・有効クラスタ数の少ない論理ブロックから順にクラスタを 2^k 個集め、IS13内でコンパクションを行う。

なお、ここでは有効クラスタ数が最も少ない2つの論理ブロックを選択するとしたが、この数は2つに限定されず、2つ以上の数であればよい。また、所定の設定値は、選択する論理ブロック数よりも1つ少ない論理ブロック数に収容可能なクラスタ数以下であればよい。

【0061】

・メインストレージ領域 (MS) 11

つぎに、MS11について説明する。MS11はトラック単位でデータの管理を行う。MS11に格納されたデータは、更新頻度が低いとみなすことができる。WC21、FS12、IS13からMS11に対してトラックデータのCopyまたはMoveが発生すると、そのトラックはMSIB11aに割り当てられた論理ブロックに対して書き込まれる。一方で、トラック中の一部のデータ(クラスタ)のみがWC等から書き込まれるような場合には、既存のMS中のトラックデータと新しいデータをマージして新しいトラックデータを作った上でMSIB11aに書き込む、後述する受動マージが行われる。MS11内に無効トラックが蓄積し、MS11に割り当てられている論理ブロックの個数がMS11として許容されるブロック数の上限値を越えるような状況が発生すると、コンパクション処理を行って、無効なフリーブロックFBを作る。

【0062】

MS11のコンパクション処理は、例えば、論理ブロック内の有効トラック数のみに注目した以下の方法を実施する。

・有効トラックが少ない論理ブロックから順番に、無効トラックを合わせることで無効なフリーブロックFBが作れるようになるまで選択する。

・選択した論理ブロックに収容されたトラックを、WC 2 1、FS 1 2、IS 1 3内のデータと統合する受動マージを行いながらコンパクションを実行する。

・ 2^i トラック統合できた論理ブロックは、TFS 1 1 bに出力し(2^i トラックMSコンパクション)、 2^i トラックに満たない個数のトラックは、MSIB 1 1 aに出力して(2^i トラック未滿コンパクション)、より多くの無効なフリーブロックFBを作る。

【0063】

TFS 1 1 bは、トラック単位でデータを管理されるFIFOである。TFS 1 1 bは、ここを通過しているデータは、後段のMS 1 1よりも更新頻度が高いとみなすためのバッファである。すなわち、TFS 1 1 bのFIFO構造においては、FIFO中を通過中の有効トラック(最新トラック)は、ホストからの同じアドレスに対する再書き込みがあった場合無効化されるので、TFS 1 1 bを通過中のトラックは、TFS 1 1 bからMS 1 1に追い出されたトラックよりも、更新頻度が高いとみなすことができる。

【0064】

図8は、データ管理部120が図5および図6に示した各構成要素を制御管理するための管理テーブルを示すものである。データ管理部120は、前述したように、ATAコマンド処理部121とNANDメモリ10とをブリッジする機能を有し、DRAM20に記憶したデータの管理を行うDRAM層管理部120aと、NANDメモリ10に記憶したデータの管理を行う論理NAND層管理部120bと、NANDメモリ10を物理記憶デバイスとして管理する物理NAND層管理部120cとから構成される。RCクラスタ管理テーブル23、WCトラック管理テーブル24、WCクラスタ管理テーブル25は、DRAM層管理部120aにより制御される。トラック管理テーブル30、FS/IS管理テーブル40、MS論理ブロック管理テーブル35、FS/IS論理ブロック管理テーブル42、FS/IS内クラスタ管理テーブル44は、論理NAND層管理部120bにより管理される。論物変換テーブル50は、物理NAND層管理部120cにより管理される。

【0065】

RC22は、逆引きテーブルであるRCクラスタ管理テーブル23によって管理される。逆引きテーブルでは、記憶デバイスの位置からその位置に記憶されている論理アドレスを検索することができる。WC21は、逆引きテーブルであるWCクラスタ管理テーブル25および正引きテーブルであるWCトラック管理テーブル24によって管理される。正引きテーブルでは、論理アドレスからその論理アドレスに対応するデータが存在する記憶デバイス位置を検索することができる。

【0066】

NANDメモリ10内のFS12(FSIB12a)、IS13、MS11(TFS11b、MSIB11a)は、トラック管理テーブル30、FS/IS管理テーブル40、MS論理ブロック管理テーブル35、FS/IS論理ブロック管理テーブル42、FS/IS内クラスタ管理テーブル44によってその論理アドレスが管理される。また、NANDメモリ10内のFS12(FSIB12a)、IS13、MS11(TFS11b、MSIB11a)は、論物変換テーブル50によって論理アドレスと物理アドレスとの変換が行われる。これらの各管理テーブルは、NANDメモリ10上の領域に記憶されており、SSD100の初期化時にNANDメモリ10からDRAM20上に読み込まれて、使用される。

【0067】

・RCクラスタ管理テーブル23(逆引き)

まず、図9を用いてRCクラスタ管理テーブル23について説明する。RC22は、前述したように、論理クラスタアドレスLSB(k-i)ビットでインデックスされるn-wayセットアソシアティブ方式で管理されている。RCクラスタ管理テーブル23は、RC(クラスタサイズ×m-line×n-way)22の各エントリのタグを管理するためのテーブルであり、各タグは、複数ビットの状態フラグ23aと、論理トラックアドレス23bによって構成されている。状態フラグ23aには、当該エントリを使用しても良いか否か(

10

20

30

40

50

有効/無効)を示すValidビットの他に、当該エントリがNANDメモリ10からの読み出し待ちか否かを示すビット、当該エントリがATAコマンド処理部121への読み出し待ちか否かを示すビットなどが含まれる。RCクラスタ管理テーブル23は、DRAM20上のタグ記憶位置からLBAに一致する論理トラックアドレスを検索する逆引きテーブルとして機能する。

【0068】

・WCクラスタ管理テーブル25(逆引き)

つぎに、図10を用いてWCクラスタ管理テーブル25について説明する。WC21は、前述したように、論理クラスタアドレスLSB($k-i$)ビットでインデックスされるn-wayセットアソシアティブ方式で管理されている。WCクラスタ管理テーブル25は、WC(クラスタサイズ \times m-line \times n-way)21の各エントリのタグを管理するためのテーブルであり、各タグは、複数ビットの状態フラグ25aと、セクタ位置ビットマップ25bと、論理トラックアドレス25cによって構成されている。

10

【0069】

状態フラグ25aには、当該エントリを使用しても良いか否か(有効/無効)を示すValidビットの他に、当該エントリがNANDメモリ10への追い出し待ちか否かを示すビット、当該エントリがATAコマンド処理部からの書き込み待ちか否かを示すビットなどが含まれる。セクタ位置ビットマップ25bは、1クラスタに含まれる $2^{(1-k)}$ セクタのうちどのセクタに有効なデータを保持しているかを $2^{(1-k)}$ ビットに展開して示すものである。このセクタ位置ビットマップ25bによって、WC21において、LBAと同じセクタ単位の管理を行うことができる。WCクラスタ管理テーブル25は、DRAM20上のタグ記憶位置からLBAに一致する論理トラックアドレスを検索する逆引きテーブルとして機能する。

20

【0070】

・WCトラック管理テーブル24(正引き)

つぎに、図11を用いてWCトラック管理テーブル24について説明する。WCトラック管理テーブル24は、WC21上に格納されているクラスタをトラック単位でまとめた情報を管理するものであり、FIFO的な機能を有するリンクドリスト構造によってトラック間のWC21に登録された順序(LRU)を表現している。なお、WC21で最後に更新された順序によってLRUを表現するようにしてもよい。各リストのエントリは、論理トラックアドレス24a、当該論理トラックアドレスに含まれるWC21中の有効クラスタ数24b、way-lineビットマップ24cおよび次のエントリへのポインタを示すnextポインタ24dから構成されている。WCトラック管理テーブル24は、論理トラックアドレス24aから所要情報を得るので、正引きテーブルとして機能する。

30

【0071】

way-lineビットマップ24cは、WC21中で当該論理トラックアドレスに含まれる有効クラスタがWC21中の $m \times n$ のエントリ中のどのエントリに格納されているかを示すマップ情報であり、有効クラスタが格納されているエントリではValidビットが"1"になっている。このway-lineビットマップ24cは、例えば、(1ビット(Valid) + $\log_2 n$ ビット(n-way)) \times mビット(m-line)で構成されている。WCトラック管理テーブル24はリンクドリスト構造を有しており、WC21中に存在する論理トラックアドレスに関する情報のみがエントリされている。

40

【0072】

・トラック管理テーブル30(正引き)

つぎに、図12を用いてトラック管理テーブル30について説明する。トラック管理テーブル30は、論理トラックアドレス単位でMS11上の論理的なデータ位置を管理するためのテーブルであり、クラスタ単位でFS12やIS13にデータが保持されている場合には、それらに関する基本情報と、詳細情報へのポインタも保持している。論理トラックアドレス30aをインデックスとした配列形式で構成される。論理トラックアドレス30aをインデックスとした各エントリは、クラスタビットマップ30b、論理ブロックI

50

D 3 0 c + 論理ブロック内トラック位置 3 0 d、クラスタテーブルポインタ 3 0 e、F S クラスタ数 3 0 f、I S クラスタ数 3 0 g などの情報で構成されている。トラック管理テーブル 3 0 は、論理トラックアドレスをインデックスとして、その論理トラックアドレスに対応する論理トラックが記憶されている論理ブロック I D（記憶デバイス位置に対応）などの所要情報を得るので、正引きテーブルとして機能する。

【 0 0 7 3 】

クラスタビットマップ 3 0 b は、1 つの論理トラックアドレス範囲に属する $2^{(k-i)}$ 個のクラスタをクラスタアドレス昇順に例えば 8 分割したビットマップであり、8 個の各ビットは、 $2^{(k-i-3)}$ 個のクラスタアドレスに対応するクラスタが M S 1 1 に存在するか、F S 1 2 もしくは I S 1 3 に存在するかを示している。ビットが “ 0 ” の場合は、その検索対象のクラスタは確実に M S 1 1 内に存在することを示し、ビットが “ 1 ” の場合は、そのクラスタは F S 1 2 もしくは I S 1 3 に存在する可能性があることを示している。

10

【 0 0 7 4 】

論理ブロック I D 3 0 c は、当該論理トラックアドレスに対応する論理トラックが記憶されている論理ブロック I D を識別するための情報である。論理ブロック内トラック位置 3 0 d は、論理ブロック I D 3 0 c で指定された論理ブロック中における当該論理トラックアドレス (3 0 a) に対応するトラックの記憶位置を示すものである。1 論理ブロックは最大 2^i 個の有効トラックで構成されるので、論理ブロック内トラック位置 3 0 d は、 i ビットで 2^i 個のトラック位置を識別する。

20

【 0 0 7 5 】

クラスタテーブルポインタ 3 0 e は、リンクドリスト構造を有する F S / I S 管理テーブル 4 0 の各リストの先頭エントリへのポインタである。クラスタビットマップ 3 0 b の検索で、当該クラスタが F S 1 2 / I S 1 3 に存在する可能性があることを示していた場合、クラスタテーブルポインタ 3 0 e を用いて F S / I S 管理テーブル 4 0 の検索を実行する。F S クラスタ数 3 0 f は、F S 1 2 内に存在する有効クラスタ数を示している。I S クラスタ数 3 0 g は、I S 1 3 内に存在する有効クラスタ数を示している。

【 0 0 7 6 】

・ F S / I S 管理テーブル 4 0（正引き）

つぎに、図 1 3 を用いて F S / I S 管理テーブル 4 0 について説明する。F S / I S 管理テーブル 4 0 は、論理クラスタ単位で F S 1 2（F S I B 1 2 a も含む）もしくは I S 1 3 に保持されているデータの位置を管理するためのテーブルである。図 1 3 に示すように、論理トラックアドレス毎に独立したリンクドリスト形式で構成され、各リストの先頭エントリへのポインタは、前述したように、トラック管理テーブル 3 0 のクラスタテーブルポインタ 3 0 e のフィールドに保持されている。図 1 3 では、2 つの論理トラックアドレス分のリンクドリストが示されている。各エントリは、論理クラスタアドレス 4 0 a、論理ブロック I D 4 0 b、論理ブロック内クラスタ位置 4 0 c、F S / I S ブロック I D 4 0 d、next ポインタ 4 0 e から構成されている。F S / I S 管理テーブル 4 0 は、論理クラスタアドレス 4 0 a から、その論理クラスタアドレスに対応する論理クラスタが記憶されている論理ブロック I D 4 0 b、論理ブロック内クラスタ位置 4 0 c（記憶デバイス位置に対応）などの所要情報を得るので、正引きテーブルとして機能する。

30

40

【 0 0 7 7 】

論理ブロック I D 4 0 b は、当該論理クラスタアドレス 4 0 a に対応する論理クラスタが記憶されている論理ブロック I D を識別するための情報である。論理ブロック内クラスタ位置 4 0 c は、論理ブロック I D 4 0 b で指定された論理ブロック中における当該論理クラスタアドレス 4 0 a に対応するクラスタの記憶位置を示すものである。1 論理ブロックは最大 2^k 個の有効クラスタで構成されるので、論理ブロック内クラスタ位置 4 0 c は、 k ビットで 2^k 位置を識別する。F S / I S ブロック I D 4 0 d は、後述する F S / I S 論理ブロック管理テーブル 4 2 のインデックスである F S / I S ブロック I D が登録されている。F S / I S ブロック I D は、F S 1 2 または I S 1 3 に所属する論理ブロック

50

を識別するための情報であり、このFS/IS管理テーブル40でのFS/ISブロックID40dは、後述するFS/IS論理ブロック管理テーブル42とのリンクのために登録されている。nextポインタ40eは、論理トラックアドレス毎にリンクされる同じリスト内の次のエントリへのポインタを示している。

【0078】

・MS論理ブロック管理テーブル35（逆引き）

つぎに、図14を用いてMS論理ブロック管理テーブル35について説明する。MS論理ブロック管理テーブル35は、MS11に用いられている論理ブロックに関する情報（どの論理トラックが記憶されているか、追記可能か等）を一元管理するためのテーブルである。なお、MS論理ブロック管理テーブル35には、FS12（FSIB12も含む）、IS13に所属する論理ブロックに関する情報も登録されている。MS論理ブロック管理テーブル35は、論理ブロックID35aをインデックスとした配列形式で構成され、エントリ数は128GBのNANDメモリ10の場合は、32Kエントリまで持つことができる。各エントリは、 2^i トラック分のトラック管理ポインタ35b、有効トラック数35c、書き込み可能先頭トラック35d、Validフラグ35eから構成されている。このMS論理ブロック管理テーブル35は、記憶デバイス位置に対応する論理ブロックID35aから、この論理ブロックに記憶されている論理トラックアドレスなどの所要情報を得るので、逆引きテーブルとして機能する。

【0079】

トラック管理ポインタ35bは、論理ブロックID35aで指定される当該論理ブロック内の 2^i 個のトラック位置毎に対応する論理トラックアドレスを保持する。この論理トラックアドレスを用いて、論理トラックアドレスをインデックスとするトラック管理テーブル30を検索することができる。有効トラック数35cは、論理ブロックID35aで指定される当該論理ブロックに記憶されているトラックのうちの有効なもの個数（最大 2^i 個）を示している。書き込み可能先頭トラック位置35dは、論理ブロックID35aで指定される当該論理ブロックが追記中のブロックであった場合における追記可能なトラック先頭位置（ $0 \sim 2^i - 1$ 、追記終了時は 2^i ）を示している。Validフラグ35eは、当該論理ブロックエントリがMS11（MSIB11aも含む）として管理されている場合に“1”である。

【0080】

・FS/IS論理ブロック管理テーブル42（逆引き）

つぎに、図15を用いてFS/IS論理ブロック管理テーブル42について説明する。FS/IS論理ブロック管理テーブル42は、FS/ISブロックID42aをインデックスとした配列形式で構成され、FS12またはIS13として利用されている論理ブロックに関する情報（論理ブロックIDとの対応、FS/IS内クラスタ管理テーブル44へのインデックス、追記可能か等）を管理するためのテーブルである。FS/IS論理ブロック管理テーブル42は、主にFS/IS管理テーブル40中のFS/ISブロックID40dを用いてアクセスされる。各エントリは、論理ブロックID42b、ブロック内クラスタテーブル42c、有効クラスタ数42d、書き込み可能先頭ページ42e、Validフラグ42fから構成されている。このMS論理ブロック管理テーブル35は、記憶デバイス位置に対応するFS/ISブロックID42aから、この論理ブロックに記憶されている論理クラスタなどの所要情報を得るので、逆引きテーブルとして機能する。

【0081】

論理ブロックID42bには、MS論理ブロック管理テーブル35に登録された論理ブロックの中で、FS12（FSIB12も含む）、IS13に所属している論理ブロックに対応する論理ブロックIDが登録される。ブロック内クラスタテーブル42cには、論理ブロック中の各クラスタ位置にどの論理クラスタアドレスで指定される論理クラスタが記録されているかを示す後述するFS/IS内クラスタ管理テーブル44へのインデックスが登録される。有効クラスタ数42dは、FS/ISブロックID42aで指定される当該論理ブロックに記憶されているクラスタのうちの有効なもの個数（最大 2^k 個）を

10

20

30

40

50

示している。書き込み可能先頭ページ位置 4 2 e は、F S / I S ブロック I D 4 2 a で指定される当該論理ブロックが追記中のブロックであった場合における追記可能な先頭ページ位置 ($0 \sim 2^j - 1$ 、追記終了時は 2^j) を示している。Valid フラグ 4 2 f は、この論理ブロックエントリが F S 1 2 (F S I B 1 2 も含む) または I S 1 3 として管理されている場合に “ 1 ” である。

【 0 0 8 2 】

・ F S / I S 内クラスタ管理テーブル 4 4 (逆引き)

つぎに、図 1 6 を用いて F S / I S 内クラスタ管理テーブル 4 4 について説明する。F S / I S 内クラスタ管理テーブル 4 4 は、F S 1 2 もしくは I S 1 3 として利用されている論理ブロック中の各クラスタ位置にどの論理クラスタが記録されているのかを示すテーブルである。1 論理ブロックあたり、 2^j ページ $\times 2^{(k-j)}$ クラスタ = 2^k 個のエントリを持ち、当該論理ブロック内のクラスタ位置の 0 番目 $\sim 2^k - 1$ 番目に対応する情報が連続領域に配置される。さらにこの 2^k 個の情報を含むテーブルが F S 1 2 および I S 1 3 に所属する論理ブロック数 (P 個) 分だけ保持されており、F S / I S 論理ブロック管理テーブル 4 2 のブロック内クラスタテーブル 4 2 c は、この P 個のテーブルに対する位置情報 (ポインタ) となっている。連続領域に配される各エントリ 4 4 a の位置は、1 論理ブロック中のクラスタ位置を示し、また各エントリ 4 4 a の内容は、当該クラスタ位置にどの論理クラスタが記憶されているかが識別できるように、F S / I S 管理テーブル 4 0 で管理される該当論理クラスタアドレスを含むリストへのポインタが登録されている。すなわち、エントリ 4 4 a は、リンクドリストの先頭を指し示すのではなく、リンクドリスト中の該当論理クラスタアドレスを含む 1 つのリストへのポインタが登録されている。

【 0 0 8 3 】

・ 論物変換テーブル 5 0 (正引き)

つぎに、図 1 7 を用いて論物変換テーブル 5 0 について説明する。論物変換テーブル 5 0 は、論理ブロック I D 5 0 a をインデックスとした配列形式で構成され、エントリ数は、1 2 8 G B の N A N D メモリ 1 0 の場合は、最大 3 2 K エントリまで持つことができる。論物変換テーブル 5 0 は、論理ブロック I D と物理ブロック I D との変換、寿命に関する情報を管理するためのテーブルである。各エントリは、物理ブロックアドレス 5 0 b、消去回数 5 0 c、読み出し回数 5 0 d から構成される。この論物変換テーブル 5 0 は、論理ブロック I D から物理ブロック I D (物理ブロックアドレス) などの所要情報を得るので、正引きテーブルとして機能する。

【 0 0 8 4 】

物理ブロックアドレス 5 0 b は、1 つの論理ブロック I D 5 0 a に所属する 8 個の物理ブロック I D (物理ブロックアドレス) を示している。消去回数 5 0 c は、当該論理ブロック I D の消去回数を示している。バッドブロック (B B) 管理は、物理ブロック (5 1 2 K B) 単位に行われるが、消去回数の管理は、3 2 ビット倍速モードによる 1 論理ブロック (4 M B) 単位に管理される。読み出し回数 5 0 d は、当該論理ブロック I D の読み出し回数を示している。消去回数 5 0 c は、例えば、N A N D 型フラッシュメモリの書き換え回数を平準化するウェアレベリング処理で利用することが可能である。読み出し回数 5 0 d は、リテンション特性の劣化した物理ブロックに保持されるデータの再書き込みを行うリフレッシュ処理で利用することが可能である。

【 0 0 8 5 】

図 8 に示した管理テーブルを管理対象毎にまとめると次のようになる。

R C 管理 : R C クラスタ管理テーブル

W C 管理 : W C クラスタ管理テーブル、W C トラック管理テーブル

M S 管理 : トラック管理テーブル 3 0、M S 論理ブロック管理テーブル 3 5

F S / I S 管理 : トラック管理テーブル 3 0、F S / I S 管理テーブル 4 0、M S 論理ブロック管理テーブル 3 5、F S / I S 論理ブロック管理テーブル 4 2、F S / I S 内クラスタ管理テーブル 4 4

【 0 0 8 6 】

なお、MS構造管理テーブル（図示せず）において、MS11, MSIB11a、TFS11bを含めたMS領域の構造を管理しており、具体的には、MS11、MSIB11a、TFS11bに割り当てた論理ブロックなどを管理している。また、FS/IS構造管理テーブル（図示せず）において、FS12、FSIB12a、IS13を含めたFS/IS領域の構造を管理しており、具体的には、FS12、FSIB12a、IS13に割り当てた論理ブロックなどを管理している。

【 0 0 8 7 】

・Read処理

つぎに、図18に示すフローチャートを参照して、読み出し処理について説明する。ATAコマンド処理部121から、Readコマンドおよび読み出しアドレスとしてのLBAが入力されると、データ管理部120は、図9に示したRCクラスタ管理テーブル23と図10に示したWCクラスタ管理テーブル25を検索する（ステップS100）。具体的には、LBAのクラスタアドレスのLSB(k-i)ビット（図7参照）に対応するlineをRCクラスタ管理テーブル23とWCクラスタ管理テーブル25から選択し、選択したlineの各wayにエントリされている論理トラックアドレス23b、25cをLBAのトラックアドレスと比較し（ステップS110）、一致したwayが存在している場合は、キャッシュヒットとし、ヒットしたRCクラスタ管理テーブル23またはWCクラスタ管理テーブル25の該当line、該当wayに対応するWC21またはRC22のデータを読み出して、ATAコマンド処理部121に送る（ステップS115）。

【 0 0 8 8 】

データ管理部120は、RC22またはWC21でヒットしなかった場合は（ステップS110）、検索対象のクラスタがNANDメモリ10のどこに格納されているかを検索する。データ管理部120は、まず、図12に示したトラック管理テーブル30を検索する（ステップS120）。トラック管理テーブル30は、論理トラックアドレス30aでインデックスされているため、LBAで指定された論理トラックアドレスに一致する論理トラックアドレス30aのエントリだけをチェックする。

【 0 0 8 9 】

まず、チェックしたいLBAの論理クラスタアドレスに基づいてクラスタビットマップ30bから対応するビットを選択する。対応するビットが“0”を示していれば、そのクラスタは確実にMS内に最新のデータが存在していることを意味する（ステップS130）。この場合は、このトラックが存在する論理ブロックIDおよびトラック位置を、同じ論理トラックアドレス30aのエントリ中の論理ブロックID30cと論理ブロック内トラック位置30dから得て、さらにLBAのクラスタアドレスのLSB(k-i)ビットを利用して、トラック位置からのオフセットを算出することで、NANDメモリ10内の当該クラスタアドレスに対応するクラスタデータが格納されている位置を算出することができる。具体的には、論理NAND層管理部120bでは、上記のようにしてトラック管理テーブル30から取得した論理ブロックID30cと論理ブロック内トラック位置30dと、LBAの論理クラスタアドレスのLSB(k-i)ビットを物理NAND層管理部120cに与える。

【 0 0 9 0 】

物理NAND層管理部120cでは、論理ブロックID30cに対応する物理ブロックアドレス（物理ブロックID）を、論理ブロックIDをインデックスとしている図17に示す論物変換テーブル50から取得し（ステップS160）、さらに取得した物理ブロックID中のトラック位置（トラック先頭位置）を論理ブロック内トラック位置30dから算出し、さらにLBAのクラスタアドレスのLSB(k-i)ビットから、前記算出した物理ブロックID中のトラック先頭位置からのオフセットを算出することで、物理ブロック中のクラスタデータを取得することができる。NANDメモリ10のMS11から取得されたクラスタデータは、RC22を介してATAコマンド処理部121に送られる（ステップS180）。

10

20

30

40

50

【 0 0 9 1 】

一方、L B Aのクラスタアドレスに基づくクラスタビットマップ3 0 bの検索で、対応するビットが“ 1 ”を示していた場合は、そのクラスタがF S 1 2またはI S 1 3に格納されている可能性がある(ステップS 1 3 0)。この場合は、トラック管理テーブル3 0の該当する論理トラックアドレス3 0 aのエントリ中のクラスタテーブルポインタ3 0 eのエントリを取り出し、このポインタを用いてF S / I S管理テーブル4 0の該当する論理トラックアドレスに対応するリンクドリストを順次検索する(ステップS 1 4 0)。具体的には、該当する論理トラックアドレスのリンクドリスト中のL B Aの論理クラスタアドレスに一致する論理クラスタアドレス4 0 aのエントリを検索し、一致する論理クラスタアドレス4 0 aのエントリが存在した場合は(ステップS 1 5 0)、一致したリスト中の論理ブロックI D 4 0 bおよび論理ブロック内クラスタ位置4 0 cを取得し、前述と同様にして、論物変換テーブル5 0を用いて物理ブロック中のクラスタデータを取得する(ステップS 1 6 0、S 1 8 0)。具体的には、取得した論理ブロックI Dに対応する物理ブロックアドレス(物理ブロックI D)を、論物変換テーブル5 0から取得し(ステップS 1 6 0)、さらに取得した物理ブロックI D中のクラスタ位置を、論理ブロック内クラスタ位置4 0 cのエントリから取得した論理ブロック内クラスタ位置から算出することで、物理ブロック中のクラスタデータを取得することができる。N A N Dメモリ1 0のF S 1 2またはI S 1 3から取得されたクラスタデータは、R C 2 2を介してA T Aコマンド処理部1 2 1に送られる(ステップS 1 8 0)。

10

【 0 0 9 2 】

このF S / I S管理テーブル4 0の検索によって、検索対象のクラスタが存在しなかった場合は(ステップS 1 5 0)、再度トラック管理テーブル3 0のエントリを検索してM S 1 1上の位置を確定する(ステップS 1 7 0)。

20

【 0 0 9 3 】

・Write処理

つぎに、図1 9に示すフローチャートを参照して、書き込み処理について説明する。F U A(D R A MキャッシュをバイパスしてN A N Dへの直接書き込みを行う)でないWriteコマンドにより書き込まれたデータは必ず一旦W C 2 1上に格納され、その後条件に応じてN A N Dメモリ1 0に対して書き込まれることになる。書き込み処理では、追い出し処理、コンパクション処理が発生する可能性がある。この実施の形態では、書き込み処理を、ライトキャッシュフラッシュ処理(以下W C F処理)と、クリーンインプットバッファ処理(以下C I B処理)との2ステージに大きく分割している。ステップS 3 0 0からステップS 3 2 0までは、A T Aコマンド処理部1 2 1からのWrite要求からW C F処理までを示しており、ステップS 3 3 0~最終ステップまでがC I B処理を示している。

30

【 0 0 9 4 】

W C F処理は、W C 2 1にあるデータをN A N Dメモリ1 0(F S 1 2のF S I B 1 2 aまたはM S 1 1のM S I B 1 1 a)にc o p yする処理であり、A T Aコマンド処理部1 2 1からのWrite要求もしくはCache Flush要求単体は、この処理のみで完結することができる。これにより処理が開始されたA T Aコマンド処理部1 2 1のWrite要求の処理遅延を最大でもW C 2 1の容量分のN A N Dメモリ1 0への書き込み時間に限定することができるようになる。

40

【 0 0 9 5 】

C I B処理は、W C F処理によって書き込まれたF S I B 1 2 aのデータをF S 1 2にM o v eする処理と、W C F処理によって書き込まれたM S I B 1 1 aのデータをM S 1 1にM o v eする処理とを含む。C I B処理を開始すると、連鎖的にN A N Dメモリ1 0内の各構成要素(F S 1 2、I S 1 3、M S 1 1など)間のデータ移動やコンパクション処理が発生する可能性があり、処理全体に要する時間は状態によって大きく変化する。

【 0 0 9 6 】

まず、W C F処理の詳細について説明する。A T Aコマンド処理部1 2 1から、Writeコマンドおよび書き込みアドレスとしてのL B Aが入力されると、D R A M層管理部1 2

50

0 は、図 10 に示した WC クラスタ管理テーブル 25 を検索する (ステップ S 300, S 305)。WC 21 の状態は、図 10 に示した WC クラスタ管理テーブル 25 の状態フラグ 25 a (例えば 3 ビット) によって規定されている。状態フラグ 25 a は、最も典型的には、Invalid (使用可能) A T A からの書き込み待ち Valid (使用不可) N A N D への追い出し待ち Invalid (使用可能) という順に状態が遷移していく。まず、L B A のクラスタアドレス L S B (k - i) ビットから書き込み先の line を決定し、決定した line の n 個の way を検索する。決定した line の n 個の way 中に、入力された L B A と同じ論理トラックアドレス 25 c が格納されている場合は (ステップ S 305)、このエントリに上書きするのでこのエントリをクラスタ書き込み用に確保する (Valid (使用不可) A T A からの書き込み待ち)。

10

【 0097】

そして、D R A M 層管理部 120 a は、該当エントリに対応する D R A M アドレスを A T A コマンド処理部 121 に通知する。A T A コマンド処理部 121 による書き込みが終了すると、WC クラスタ管理テーブル 25 の該当エントリの状態フラグ 25 a を Valid (使用不可) にし、さらにセクタ位置ビットマップ 25 b および論理トラックアドレス 25 c の欄に所要のデータを登録する。また、WC トラック管理テーブル 24 を更新する。具体的には、WC トラック管理テーブル 24 の各リスト中に既に登録済みの論理トラックアドレス 24 a と同じ L B A アドレスが入力された場合は、該当するリストの WC クラスタ数 24 b、way - line ビットマップ 24 c を更新するとともに、当該リストが最新のリストとなるように next ポインタ 24 d を変更する。また、WC トラック管理テーブル 24 の各リスト中に登録済みの論理トラックアドレス 24 a と異なる L B A アドレスが入力された場合は、新たに新しい論理トラックアドレス 24 a、WC クラスタ数 24 b、way - line ビットマップ 24 c、next ポインタ 24 d の各エントリを有するリストを作成し、最新のリストとして登録する。以上のようなテーブル更新を行って、書き込み処理が完了する (ステップ S 320)。

20

【 0098】

一方、決定した line の n 個の way 中に、入力された L B A と同じ論理トラックアドレス 25 c が格納されていない場合は、N A N D メモリへの追い出しが必要であるか否かを判断する (ステップ S 305)。すなわち、まず、決定した line 中の書き込み可能な way が最後の n 個目の way であるか否かを判断する。書き込み可能な way とは、Invalid (使用可能) の状態フラグ 25 a をもつ way かあるいは Valid (使用不可) でかつ N A N D への追い出し待ちの状態フラグ 25 a を持つ way である。状態フラグ 25 a が、N A N D への追い出し待ちであるとは、追い出しが開始されて追い出しの終了待ちであることを意味する。そして、書き込み可能な way が最後の n 個目の way でない場合であって、かつ書き込み可能な way が、Invalid (使用可能) の状態フラグ 25 a をもつ way である場合は、このエントリをクラスタ書き込み用に確保する (Invalid (使用可能) A T A からの書き込み待ち)。そして、該当エントリに対応する D R A M アドレスを A T A コマンド処理部 121 に通知し、A T A コマンド処理部 121 によって書き込みを実行させる。そして、前記同様、WC クラスタ管理テーブル 25 および WC トラック管理テーブル 24 を更新する (ステップ S 320)。

30

40

【 0099】

また、書き込み可能な way が最後の n 個目の way でない場合であって、かつ書き込み可能な way が、Valid (使用不可) でかつ N A N D への追い出し待ちの状態フラグ 25 a を持つ way である場合は、このエントリをクラスタ書き込み用に確保する (Valid (使用不可) でかつ N A N D への追い出し待ち Valid (使用不可) でかつ N A N D への追い出し待ちかつ A T A からの書き込み待ち)。そして、追い出しが終了すると、状態フラグ 25 a を A T A からの書き込み待ちにし、さらに、該当エントリに対応する D R A M アドレスを A T A コマンド処理部 121 に通知し、A T A コマンド処理部 121 によって書き込みを実行させる。そして、前記同様、WC クラスタ管理テーブル 25 および WC トラック管理テーブル 24 を更新する (ステップ S 320)。

50

【 0 1 0 0 】

以上の処理は、A T Aコマンド処理部 1 2 1 からからの書き込み要求が入力された際に、追い出し処理をトリガしなくてもよい場合である。一方、この後の説明は、書き込み要求が入力された時点後に、追い出し処理をトリガする場合である。ステップ S 3 0 5 において、決定したline中の書き込み可能なwayが最後の n 個目のwayである場合は、前述した W C 2 1 から N A N Dメモリ 1 0 へ追い出すデータの決定方法の (i) の箇所で説明した条件、すなわち、

(i) タグによって決定されたlineの書き込み可能なwayが最後の(本実施形態では、 n 個目の)空きwayだった場合、則ち最後の空きwayが使用される場合は、そのlineに登録されたトラックのうち L R U に基づいて最も古く更新されたトラックを追い出し確定するに
10 基づいて追い出すトラックすなわち W C 2 1 内エントリを選択する。

【 0 1 0 1 】

D R A M 層管理部 1 2 0 a は、以上の方針で追い出すトラックを決定すると、前述したように、その際追い出すのは同一トラックに含まれる W C 2 1 中の全クラスタであり、追い出されるクラスタ量がトラックサイズの 5 0 % を超えていれば、すなわち追い出し確定トラックのうち W C 中に有効クラスタ数が $2^{(k-i-1)}$ 個以上ある場合は、 M S I B 1 1 a へ追い出しを行い(ステップ S 3 1 0)、超えていなければ、すなわち追い出し確定トラックのうち W C 中に有効クラスタ数が $2^{(k-i-1)}$ 個未満である場合は、 F S I B 1 2 a へと追い出す(ステップ S 3 1 5)。 W C 2 1 から M S I B 1 1 a への追い出し、 W C 2 1 から F S I B 1 2 a への追い出しの詳細は、後述する。選択された追い出し
20 エントリの状態フラグ 2 5 a は、Valid (使用不可) から N A N Dメモリ 1 0 への追い出し待ちに移行される。

【 0 1 0 2 】

この追い出し先の判定は、 W C トラック管理テーブル 2 4 を用いて実行される。すなわち、 W C トラック管理テーブル 2 4 には、論理トラックアドレス毎に、有効なクラスタ数を示す W C クラスタ数 2 4 b のエントリが登録されており、この W C クラスタ数 2 4 b のエントリを参照することで W C 2 1 からの追い出し先を、 F S I B 1 2 a、 M S I B 1 1 a の何れにするかを決定する。また、 way - line ビットマップ 2 4 c に、当該論理トラックアドレスに所属する全てのクラスタがビットマップ形式で登録されているので、追い出し
30 を行う際には、この way - line ビットマップ 2 4 c を参照することで、容易に、追い出すべき各クラスタの W C 2 1 での記憶位置を知ることができる。

【 0 1 0 3 】

また、上記書き込み処理中または書き込み処理後、前述の下記条件、
(ii) W C 2 1 に登録されているトラックの数が所定数を超えた場合、
が成立した場合も、上記同様にして N A N Dメモリ 1 0 への追い出し処理を実行する。

【 0 1 0 4 】

W C M S I B (C o p y)

つぎに、上記有効クラスタ数(有効クラスタ数が $2^{(k-i-1)}$ 個以上)に基づく判定により W C 2 1 から M S I B 1 1 a への追い出しが発生したときは、前述したように、
40 次のような手順を実行する(ステップ S 3 1 0)。

1 . W C クラスタ管理テーブル 2 5 を参照し、追い出しを行うクラスタに対応するタグ中のセクタ位置ビットマップ 2 5 b を参照し、セクタ位置ビットマップ 2 5 b が全て " 1 " でない場合は、 N A N Dメモリ 1 0 に含まれる同一クラスタ中のセクタとマージする、後述のトラック内セクタ穴埋めを行う。また、トラック中の W C 2 1 内に存在しないクラスタについては N A N Dメモリ 1 0 から読み出してマージする、受動マージ処理を実行する。

2 . 追い出し確定トラックが 2^i 個未満の場合、 W C 2 1 中のトラックの古いものから 2^i 個になるまで $2^{(k-i-1)}$ 個以上有効クラスタを持つ追い出し確定トラックを追加する。

3 . C o p y されるトラックが 2^i 個以上あれば、 2^i 個ずつを組として、 M S I B 1
50

1 a に対して論理ブロック単位に書き込みを行う。

4. 2^i 個組みに出来なかったトラックを M S I B 1 1 a に対してトラック単位に書き込みを行う。

5. C o p y 終了後に既に F S、I S、M S 上に存在していたクラスタ、トラックのうちコピーされたトラックに属するものを無効化する。

【0105】

このような W C 2 1 から M S I B 1 1 a への C o p y 処理に伴う各管理テーブルの更新処理について説明する。W C クラスタ管理テーブル 2 5 中の追い出されたトラックに所属する W C 2 1 中の全クラスタに対応するエントリ中の状態フラグ 2 5 a は Invalid とされ、その後これらエントリに対する書き込みが可能となる。また、W C トラック管理テーブル 2 4 中の追い出されたトラックに対応するリストについては、例えば直前のリストの n e x t ポインタ 2 4 d が変更または削除されて、無効化される。

10

【0106】

一方、W C 2 1 から M S I B 1 1 a に対するトラック移動が発生すると、これに伴いトラック管理テーブル 3 0 および M S 論理ブロック管理テーブル 3 5 が更新される。まず、トラック管理テーブル 3 0 のインデックスである論理トラックアドレス 3 0 a を検索することで、移動されたトラックに対応する論理トラックアドレス 3 0 a が既に登録されているか否かを判定する。既に登録されている場合は、該当インデックスのクラスタビットマップ 3 0 b (M S 1 1 側への移動であるので、該当ビットを全て " 0 " にする)、論理ブロック I D 3 0 c + 論理ブロック内トラック位置 3 0 d のフィールドを更新する。移動されたトラックに対応する論理トラックアドレス 3 0 a が未登録の場合は、該当する論理トラックアドレス 3 0 a のエントリに対し、クラスタビットマップ 3 0 b、論理ブロック I D 3 0 c + 論理ブロック内トラック位置 3 0 d を登録する。また、トラック管理テーブル 3 0 の変更に応じて、M S 論理ブロック管理テーブル 3 5 における、論理ブロック I D 3 5 a、該当トラック管理ポインタ 3 5 b、有効トラック数 3 5 c、書き込み可能先頭トラック 3 5 d などのエントリを必要に応じて更新する。

20

【0107】

なお、他の領域 (F S 1 2 や I S 1 3) 等から M S 1 1 に対してトラック書き込みが発生した場合、もしくは M S 1 1 内部のコンパクション処理による M S 内トラック書き込みが発生した場合、書き込み対象のトラックに含まれる W C 2 1 内有効クラスタも同時に M S に書き込まれる。W C 2 1 から M S 1 1 への書き込みとしてこのような受動的マージも存在する。そのような受動的マージが行われた場合は、それらのクラスタは W C 2 1 上から削除 (無効化) されることになる。

30

【0108】

W C F S I B (C o p y)

つぎに、上記有効クラスタ数 (有効クラスタ数が $2^{(k-i-1)}$ 個未満) に基づく判定により W C 2 1 から F S I B 1 2 a への追い出しが発生したときは、前述したように、次のような手順を実行する。

1. W C クラスタ管理テーブル 2 5 における追い出しを行うクラスタに対応するタグ中のセクタ位置ビットマップ 2 5 b を参照し、セクタ位置ビットマップ 2 5 b が全て " 1 " でない場合は、N A N D メモリ 1 0 に含まれる同一クラスタ中のセクタとマージする、クラスタ内セクタ穴埋めを行う。

40

2. W C 内のトラックを古い順に辿って $2^{(k-i-1)}$ 個未満の有効クラスタしか持たないトラックからクラスタを取り出して行き、有効クラスタ数が 2^k 個になったらそれら全クラスタを F S I B 1 2 a に論理ブロック単位に書き込む。

3. 2^k 個見つからなかった場合には、有効クラスタ数が $2^{(k-i-1)}$ 個未満の全てのトラックを必要な論理ページ数分だけ F S I B 1 2 a に書き込む。

4. C o p y 終了後に既に F S、I S 上に存在していたクラスタのうちコピーされたのと同じものを無効化する。

【0109】

50

このようなWC 2 1からFSIB 1 2 aへのCopy処理に伴う各管理テーブルの更新処理について説明する。WCクラスタ管理テーブル2 5中の追い出されたトラックに所属するWC 2 1中の全クラスタに対応するエントリ中の状態フラグ2 5 aはInvalidとされ、この後これらエントリに対する書き込みが可能となる。また、WCトラック管理テーブル2 4中の追い出されたトラックに対応するリストについては、例えば直前のリストのnextポインタ2 4 dが変更または削除されて、無効化される。一方、WC 2 1からFSIB 1 2 aに対するクラスタ移動が発生すると、これに伴いトラック管理テーブル3 0のクラスタテーブルポインタ3 0 e、FSクラスタ数3 0 fなどを更新するとともに、FS/IS管理テーブル4 0の論理ブロックID 4 0 b、論理ブロック内クラスタ位置4 0 cなどを更新する。なお、もともとFS 1 2に存在していなかったクラスタについては、FS/IS管理テーブル4 0のリンクリストへのリストが追加される。この更新に伴い、MS論理ブロック管理テーブル3 5、FS/IS論理ブロック管理テーブル4 2、およびFS/IS内クラスタ管理テーブル4 4の該当箇所を更新する。

10

【0 1 1 0】

CIB処理

上記のようなWCF処理が終了すると、つぎに、論理NAND層管理部1 2 0 bは、WCF処理によって書き込まれたFSIB 1 2 aのデータをFS 1 2にMoveする処理と、WCF処理によって書き込まれたMSIB 1 1 aのデータをMS 1 1にMoveする処理などを含むCIB処理を実行する。CIB処理を開始すると、前述したように、連鎖的に各ブロック間のデータ移動やコンパクション処理が発生する可能性があり、処理全体に要する時間は状態によって大きく変化する。このCIB処理においては、基本的には、先ずMS 1 1でのCIB処理が行われ(ステップS 3 3 0)、つぎに、FS 1 2でのCIB処理が行われ(ステップS 3 4 0)、つぎに再びMS 1 1でのCIB処理が行われ(ステップS 3 5 0)、つぎにIS 1 3でのCIB処理が行われ(ステップS 3 6 0)、最後に再びMS 1 1でのCIB処理が行われる(ステップS 3 7 0)。なお、FS 1 2からMSIB 1 1 aへの追い出し処理、あるいはFS 1 2からIS 1 3への追い出し処理、あるいはIS 1 3からMSIB 1 1 aへの追い出し処理の際に、手順にループが発生した場合は、上記順番通りにならない場合もある。MS 1 1、FS 1 2およびIS 1 3でのCIB処理を別々に説明する。

20

【0 1 1 1】

MS 1 1のCIB処理

まず、MS 1 1でのCIB処理について説明する(ステップS 3 3 0)。WC 2 1、FS 1 2、IS 1 3からMS 1 1に対してトラックデータの移動が発生すると、そのトラックデータはMSIB 1 1 aに書き込まれる。MSIB 1 1 aへの書き込み完了後は、前述したように、トラック管理テーブル3 0を更新してトラックが配置される論理ブロックID 3 0 cとブロック内トラック位置3 0 dなどを変更する(Move)。MSIB 1 1 aに新たなトラックデータが書き込まれた場合、もともとMS 1 1もしくはTF S 1 1 bに存在していたトラックデータは無効化される。この無効化処理は、MS論理ブロック管理テーブル3 5における古いトラック情報が保存されていた論理ブロックのエントリからトラックを無効化することで実現する。具体的には、MS論理ブロック管理テーブル3 5の該当エントリ中のトラック管理ポインタ3 5 bのフィールド中の該当トラックのポインタが削除され、有効トラック数が- 1される。このトラック無効化によって1論理ブロック中の全てのトラックが無効になった場合は、Validフラグ3 5 eが無効化される。このような無効化などにより、MS 1 1のブロックは無効なトラックを含んだものが発生し、これが繰り返されるとブロックの利用効率が低下して、使用可能な論理ブロックに不足が生じることがある。

30

40

【0 1 1 2】

データ管理部1 2 0は、このような事態が発生して、MS 1 1に割り当てられている論理ブロックの個数がMS 1 1として許容されるブロック数の上限値を越えるような状況が発生すると、コンパクション処理を行って、無効なフリーブロックFBを作る。無効なフ

50

リーブロックFBは、物理NAND層管理部120cに返却される。そして、論理NAND層管理部120bは、MS11に割り当てられている論理ブロックの個数を減らした後、新たに書き込み可能なフリーブロックFBを物理NAND層管理部120cから取得する。コンパクション処理とは、コンパクション対象の論理ブロックが持つ有効クラスタを新しい論理ブロックに集めたり、あるいはコンパクション対象の論理ブロック中の有効トラックを他の論理ブロックにCopyしたりすることで、物理NAND層管理部120cに返却する無効なフリーブロックFBを作り、論理ブロックの利用効率を向上させるための処理である。なお、コンパクションを行う際には、コンパクション対象となったトラック領域に対して、WC、FS、IS上の有効なクラスタが存在する場合、それらを全てマージする受動マージを実行する。また、TFS11bに登録されている論理ブロックについて、コンパクション対象に含めない。

10

【0113】

以下に、MSIB11aにフルに成ったブロックが存在する場合を発生条件とした、MSIB11aからMS11またはTFS11bへの追い出しとコンパクション処理の一例について具体的に説明する。

1. MS論理ブロック管理テーブル35のValidフラグ35eを参照することにより、MS11内に無効となった論理ブロックが存在する場合、そのブロックを無効なフリーブロックFBとする。

2. MSIB11aでフルに成った論理ブロックをMS11に追い出す。具体的には、前述したMS構造管理テーブル(図示せず)を更新して、該当論理ブロックをMSIB管理下からMS管理下に移し変える。

20

3. MS11に割り当てられている論理ブロックの個数がMS11として許容されるブロック数の上限値を越えるような状況が発生するか否かを判断し、発生している場合に、以下のMSコンパクションを実行する。

4. MS論理ブロック管理テーブル35の有効トラック数35cのフィールドなどを参照することにより、TFS11bに含まれない論理ブロックのうち無効にされたトラックを持つものを、有効トラック数でソートする。

5. 有効トラック数の少ない論理ブロックから、トラックを集めてコンパクションを実施する。この際にまず、1論理ブロック分(2ⁱトラック)ずつCopyしてコンパクションを実施する。なお、コンパクション対象のトラックがWC21、FS12、IS13に有効クラスタを持つ場合にはそれらもマージする。

30

6. コンパクション元の論理ブロックを無効なフリーブロックFBとする。

7. コンパクションして有効な2ⁱトラックで構成された1論理ブロックができたなら、TFS11bの先頭にMoveする。

8. 論理ブロック内の有効トラックを他の論理ブロックにCopyして、無効なフリーブロックFBが作れる場合は、2ⁱトラック未満の個数の有効トラックをMSIB11aに対し、トラック単位で追記書き込みする。

9. コンパクション元の論理ブロックを無効なフリーブロックFBとする。

10. MS11に割り当てられている論理ブロックの個数がMS11として許容されるブロック数の上限値を下回ると、MSコンパクション処理を終了する。

40

【0114】

FS12のCIB処理

つぎに、FS12でのCIB処理について説明する(ステップS340)。WC21からFSIB12aへのクラスタ書き込み処理によって全ページ書き込み済みの論理ブロックがFSIB12a中に作られた場合、FSIB12a中のそれらのブロックは、FSIB12aからFS12に対してMoveされる。このMoveにともなって複数の論理ブロックで構成されるFIFO構造のFS12から古い論理ブロックが追い出される状況が発生する。

【0115】

FSIB12aからFS12に対する追い出しおよびFS12からのブロック追い出し

50

は、具体的には、次のように実現される。

1. FS/IS論理ブロック管理テーブル42のValidフラグ35eなどを参照することにより、FS12内に無効となった論理ブロックが存在する場合、そのブロックを無効なフリーブロックFBとする。

2. FSIB12aでフルに成ったブロックをFS12に追い出す。具体的には、前述したFS/IS構造管理テーブル(図示せず)を更新して、該当ブロックをFSIB管理下からFS管理下に移し変える。

3. FS12に割り当てられている論理ブロックの個数がFS12として許容されるブロック数の上限値を越えるような状況が発生するかどうかを判断し、発生している場合に、以下の追い出しを実行する。

4. まず追い出し対象の最古の論理ブロック中のクラスタデータのうちIS13に移動せずに、直接MS11に移動すべきものを決定する(実際には、MSの管理単位がトラックであるので、トラック単位での決定)。

(ア) 追い出し対象の論理ブロック中の有効クラスタをページの先頭から順にスキャンする。

(イ) クラスタが属するトラックがFS中に何個の有効クラスタを保有しているか、トラック管理テーブル30のFSクラスタ数30fのフィールドを参照して検索する。

(ウ) トラック内有効クラスタ数が所定の閾値(例えば 2^{k-i} 個の50%)以上だった場合、そのトラックをMSへの追い出し候補とする。

5. MS11に追い出すべきトラックをMSIB11aに対して書き込む。

6. 追い出しトラックが残っている場合、さらにMSIB11への追い出しを実行する。

7. 上記2~4の処理の後も追い出し対象の論理ブロックに有効なクラスタが存在している場合、この論理ブロックをIS13にMoveする。

なお、FS12からMSIB11aへの追い出しが発生したときには、その直後、MS11での前述したCIB処理が実行される(ステップS350)。

【0116】

IS13のCIB処理

つぎに、IS13でのCIB処理について説明する(ステップS360)。上記したFS12からIS13へのブロック移動によって論理ブロックがIS13に追加されるが、これにともなって複数個の論理ブロックで構成されるIS13に対して管理可能なブロック数の上限を超えてしまう状況が発生する。このような状況が発生した場合、IS13では、まずMS11への1~複数個の論理ブロックの追い出しを行った後、ISコンパクションを実行する。具体的には、次のような手順を実行する。

1. IS13に含まれるトラックをトラック内の有効クラスタ数×有効クラスタ係数でソートし、積の値が大きいトラック 2^{i+1} 個(2論理ブロック分)を集めてMSIB11aに追い出す。

2. 有効クラスタ数が最も少ない 2^{i+1} 個の論理ブロックの合計有効クラスタ数が例えば、所定の設定値である 2^k 個(1論理ブロック分)以上ある場合は、上のステップを繰り返す。

3. 上記の追い出しを行った後、有効クラスタ数の少ない論理ブロックから順にクラスタを 2^k 個集め、IS13内でコンパクションを行う。

4. コンパクション元の論理ブロックのうち有効クラスタがなくなったものを無効なフリーブロックFBとして返還する。

なお、IS13からMSIB11aへの追い出しが発生したときには、その直後、MS11での前述したCIB処理が実行される(ステップS370)。

【0117】

図20は、各構成要素間のデータの流れにおける入力と出力の組み合わせ、およびそのデータの流れが何をトリガとして発生するかを示すものである。FS12は、基本的には、WC21からのクラスタ追い出しによってデータが書き込まれるが、WC21からFS

10

20

30

40

50

12への追い出しに付随してクラスタ内セクタ穴埋め(クラスタ穴埋め)が必要な場合は、FS12、IS13、MS11からのデータがコピーされる。WC21では、WCクラスタ管理テーブル25のタグ中のセクタ位置ビットマップ25bによって当該クラスタアドレス中の $2^{(1-k)}$ 個のセクタの有無を識別させることによってセクタ(512B)単位の管理をすることが可能である。これに対し、NANDメモリ10での機能要素であるFS12、IS13の管理単位はクラスタであり、MS11の管理単位は、トラックである。このように、NANDメモリ10での管理単位は、セクタより大きいため、WC21からNANDメモリ10に対して、データを書き込む際に、書き込まれるデータと同一クラスタアドレスのデータがNANDメモリ10中に存在する場合、WC21からNANDメモリ10に書き込まれるクラスタ中のセクタと、NANDメモリ10中に存在する同一クラスタアドレス内のセクタとをマージしてから、NANDメモリ10に書き込む必要がある。

10

【0118】

この処理が、図20に示したクラスタ内セクタ穴埋め処理(クラスタ穴埋め)と、トラック内セクタ穴埋め(トラック穴埋め)であり、これらの処理を行わないと、正しいデータが読み出せなくなる。そこで、WC21からFSIB12aまたはMSIB11aにデータを追い出す際には、WCクラスタ管理テーブル25を参照し、追い出しを行うクラスタに対応するタグ中のセクタ位置ビットマップ25bを参照し、セクタ位置ビットマップ25bが全て“1”でない場合は、NANDメモリ10に含まれる同一クラスタまたは同一トラック中のセクタとマージする、クラスタ内セクタ穴埋めまたはトラック内セクタ穴埋めを行う。この処理には、DRAM20の作業領域が使用され、DRAM20の作業領域からMSIB11aに書き込まれたり、FSIB12aに書き込まれたりする。

20

【0119】

IS13は、基本的には、FS12からのブロック追い出しによってデータが書き込まれる(Move)か、IS内部のコンパクションによってデータが書き込まれる。MS11は、全ての箇所からデータが書き込まれ得る。その際、MS11は、トラック単位にしかデータを書き込めないために、MS自身のデータによる穴埋めが発生しうる。また、トラック単位の書き込みを行う際には、他のブロックにある断片化されたデータも受動マージによって書き込まれることになる。さらにMS11は、MSコンパクションによる書き込みもある。なお、受動マージにおいては、WC21、FS12またはIS13の3つの構成要素のうちの1つの構成要素からMS11へのトラック追い出しまたは論理ブロック追い出し(2^i トラック分の追い出し)が発生した際、1つの構成要素での追い出し対象のトラック(または論理ブロック)に含まれる他の2つの構成要素内の有効クラスタおよびMS11内の有効クラスタが、DRAM20の作業領域に集められて、DRAM20の作業領域から1トラック分のデータとしてMSIB11aに書き込まれる。

30

【0120】

つぎに、本実施形態をより詳細に説明する。図21は、図6に示したNANDメモリ10の書き込み処理に係わるより詳細な機能構成を示すものである。重複する説明は、省略する。

【0121】

・FS構成

FS部12Qは、FSIB12aとFS12によって構成されている。FS12は、多数の論理ブロック分の容量を有し、論理ブロック単位でFIFO構造が管理されている。FS12の前段には、WC21から追い出されたデータが入力されるFSインプットバッファ(FSIB)12aが設けられている。FSIB12aは、FSフルブロックバッファ(FSFB)12aaと、FS追記バッファ(FS追記IB)12abとを備えている。FSFB12aaは、1~複数個の論理ブロック分の容量を有し、またFS追記IB12abも1~複数個の論理ブロック分の容量を有する。WC21からの追い出しデータが1論理ブロック分あればFSFB12aaに対する論理ブロック単位のデータコピーが行われ、そうでなければFS追記IB12abに対して論理ページ単位の追記書き込みが行

40

50

われる。

【0122】

・IS構成

IS部13Qは、IS13と、ISインプットバッファ（ISIB）13aと、ISコンパクションバッファ13cによって構成されている。ISIB13aは、1～複数個の論理ブロック分の容量を有し、ISコンパクションバッファ13cは例えば1論理ブロック分の容量を有し、IS13は、多数の論理ブロック分の容量を有する。IS13は、例えば、論理ブロック単位で、FS12と同様、FIFO構造が管理されている。ISコンパクションバッファ13cは、IS部13Qでのコンパクションを行うためのバッファである。

10

【0123】

前述したように、IS部13Qは、FS部12Qと同様クラスタ単位でデータの管理を行う。FS部12QからIS部13Qに対して論理ブロックの移動、すなわちFS12からの追い出しが発生すると、以前FS部12Qの管理対象であった追い出し対象の論理ブロックはポイントの付け替えによりIS部13（詳細には、ISIB13a）の管理対象ブロックとなる。このFS部12QからIS部13Qへの論理ブロックの移動により、IS13のブロック数が所定の上限値を超えると、IS13からMS部11Qへのデータ追い出しおよびISコンパクション処理が実行され、IS部13Qのブロック数は規定値に戻される。

20

【0124】

・MS構成

MS部11Qは、MSIB11aと、トラック前段バッファ（TFS）11bと、MS11によって構成されている。MSIB11aは、1～複数個（この実施の形態では4個）のMSフルブロックインプットバッファ（以下MSFB）11aaと、1～複数個（この実施の形態では2個）の追記インプットバッファ（以下MS追記IB）11abとを備えている。1つのMSFB11aaは、1論理ブロック分の容量を有する。MSFB11aaは、論理ブロック単位の書き込み時に用いられる。1つのMS追記IB11abは、論理ブロック分の容量を有する。MS追記IB11abは、トラック単位の追記書き込み時に用いられる。

30

【0125】

MSFB11aaには、WC21から追い出された論理ブロック、またはFS12から追い出された論理ブロック、またはIS13から追い出された論理ブロックがCopyされる。1つのMSFB11aaにCopyされた論理ブロックは、TFS11bを経ることなくMS11に直接Moveされる。このMS11へのMove後は、フリーブロックFBがMSFB11aaとして割り当てられる。

【0126】

MS追記IB11abには、WC21から追い出されたトラック、またはFS12から追い出されたトラックが追記的にCopyされる。このようなトラック単位で追記書き込みされるMS追記IB11abのうちでフルになった論理ブロックは、TFS11bにMoveされる。このTFS11bへのMove後は、フリーブロックFBがMS追記IB11abとして割り当てられる。

40

【0127】

TFS11bは、前述したように、多数の論理ブロック分の容量を有し、MS追記IB11abとMS11との間に介在するFIFO構造を有するバッファである。FIFO構造を持つTFS11bの入力側には、トラック単位で追記書き込みされるMS追記IB11abのうちでフルになった論理ブロックがMoveされ、またMS11内のコンパクション処理によって形成された論理ブロックが、MSコンパクションバッファ11cからMoveされる。MSコンパクションバッファ11cは、MS11でのコンパクションを行うためのバッファである。なお、MS11内部のコンパクション処理によって、MS内トラックがMSコンパクションバッファ11cに書き込まれる際には、書き込み対象のトラ

50

ックに含まれるWC 2 1、FS部 1 2 Q、IS部 1 3 Q内の有効クラスタがDRAM 2 0の作業領域を介してMSコンパクションバッファ 1 1 cに書き込まれる、受動マージが行われる。本実施の形態においては、MSIB 1 1 aおよびTFS 1 1 bに登録されている論理ブロックについては、コンパクション対象に含めない。

【0128】

図22は、データ管理部 1 2 0のより詳細な機能構成を示すものである。データ管理部 1 2 0は、前述したように、DRAM 2 0に記憶したデータの管理を行うDRAM層管理部 1 2 0 aと、NANDメモリ 1 0に記憶したデータの管理を行う論理NAND層管理部 1 2 0 bと、NANDメモリ 1 0を物理記憶デバイスとして管理する物理NAND層管理部 1 2 0 cとから構成される。

10

【0129】

DRAM層管理部 1 2 0 aは、RCクラスタ管理テーブル 2 3、WCクラスタ管理テーブル 2 5、WCトラック管理テーブル 2 4を備え、これら管理テーブルに基づいてDRAM層の管理を行う。論理NAND層管理部 1 2 0 bは、トラック管理テーブル 3 0、MSブロック管理テーブル 3 5、FS/IS管理テーブル 4 0、FS/IS論理ブロック管理テーブル 4 2、FS/IS内クラスタ管理テーブル 4 4の他に、MS構造管理テーブル 6 0、FS/IS構造管理テーブル 6 5を備え、これらの管理テーブルに基づいてNANDメモリ 1 0の論理NAND層の管理を行う。物理NAND層管理部 1 2 0 cは、論物変換テーブル 5 0の他に、パッドブロック管理テーブル(BB管理テーブル) 2 0 0、予約ブロック管理テーブル(RBブロック管理テーブル) 2 1 0、フリーブロック管理テーブル(FB管理テーブル) 2 2 0、アクティブブロック管理テーブル(AB管理テーブル) 2 3 0を備え、これらの管理テーブルを用いてNANDメモリ 1 0の物理NAND層の管理を行う。

20

【0130】

・物理NAND層

まず、物理NAND層について説明する。前述したように、32ビット倍速モードでは、4ch(ch 0、ch 1、ch 2、ch 3)を並列で動作させ、更に、NANDメモリチップの備える倍速モードを利用して消去/書き込み/読み出しを行う。図23に示すように、4つの並列動作要素 1 0 a ~ 1 0 d内の各NANDメモリチップは、例えば、プレーン 0、プレーン 1の2つの領域(District)に分割されている。なお、分割数は2に限定されるものではない。プレーン 0およびプレーン 1は、互いに独立した周辺回路(例えば、ロウデコーダ、カラムデコーダ、ページバッファ、データキャッシュ等)を備えており、NANDコントローラ 1 1 2から入力されるコマンドに基づき、同時に消去/書き込み/読み出しを行うことが可能である。NANDメモリチップの倍速モードでは、上記プレーン 0およびプレーン 1を並列制御することにより、高速書き込みを実現している。

30

【0131】

物理ブロックサイズは512kBであるので、32ビット倍速モードでは、4chの並列動作および2つのプレーンに対する同時アクセスにより、その消去単位は512kB×4×2=4MBとなる。すなわち、この32ビット倍速モードでは、結果的に、8プレーンが並列動作することになる。

40

【0132】

図24は、論物変換テーブル 5 0の他の例を示すものである。図24に示す論物変換テーブル 5 0は、図17に示した論物変換テーブル 5 0に対し、論理ブロックID 5 0 aに対応する論理ブロックが消去された時刻を示す消去時刻 5 0 eのフィールドが更に追加されている。消去時刻 5 0 eとしては、例えば、NANDメモリチップ内の論理ブロックに対して消去動作が行われた回数を計測することで得られる値、或いは、NANDコントローラ 1 1 2の通電時間等を利用すればよい。消去時刻 5 0 eは、後述するFB管理テーブル 2 2 0でのフリーブロックFB管理に利用される。

【0133】

BB管理テーブル 2 0 0は、物理ブロック(512kB)単位でパッドブロックBBを

50

管理するためのテーブルである。BB管理テーブル200は、図25に示すように、例えば、4(チャンネル)×2(プレーン/チャンネル)個のチャンネル内プレーン毎に、(物理ブロック数/プレーン)×(NANDメモリチップ数/1並列動作要素)個数分の物理ブロックに関する情報をもつ二次元配列形式で構成されており、各エントリには、各物理ブロックに対する物理ブロックID200aが保持されている。

【0134】

本実施形態の場合、1NANDメモリチップは2GBサイズであり、第1番目のチップのプレーン0には、「0」～「2047」の物理ブロックIDが割り当てられ、第1番目のチップのプレーン1には、「2048」～「4095」の物理ブロックIDが割り当てられる。物理NAND層管理部120cは、使用中に発生したバッドブロックBBをBB管理テーブル200に登録する際には、ソートはせずに対応するチャンネル内プレーンID(ID#0～ID#7)の最後尾の有効エントリの直後に順に追加する。

10

【0135】

RB管理テーブル210は、8個の物理ブロック単位(512kB)で4MB論理ブロックを形成する際に余ったブロック(予約ブロックRB)を管理するためのテーブルであり、BB管理テーブル200と同様のフォーマットで管理される。対応するチャンネル内プレーンIDごとにFIFO管理とすることで、最も古く登録された予約ブロックから優先して利用する。

【0136】

FB管理テーブル220は、4MB論理ブロック単位で現在用途未割り当てのフリーブロックFBを管理するためのテーブルであり、フリーブロックFBとなった順にソートされたFIFO形式のリストである。各エントリは、論理ブロックIDを保持する。コンパクション処理などにより、FB管理テーブル220に返却されたフリーブロックFBは、リストの最後尾に追加され、フリーブロックFB割り当ては、リストの先頭ブロックを返すことで行う。

20

【0137】

FB管理テーブルは、図26に示すように、返却FIFOリスト220aと、割り当てリスト220bとの2段構成になっている。返却FIFOリスト220aは、消去時刻50e順に整列しており、割り当てリスト220bは、消去回数50cの少ない論理ブロックほどリストの先頭に位置する。これは、短い時間間隔で消去動作を繰り返すことを防ぐための構成である。不要になり、FB管理テーブル220に返却された論理ブロックは、返却FIFOリスト220aの最後尾に追加され、一定の期間にここに保持される。

30

【0138】

返却FIFOリスト220aから押し出された論理ブロックは、その消去回数50cに応じて割り当てリスト220bの途中に挿入される。論理NAND層管理部120bからフリーブロックFBの割り当てを要求されると、論理NAND層管理部120cは、割り当てリスト220bの先頭から取り出し、割り当てを行う。

【0139】

上記FB管理テーブルにより、全ての論理ブロックの消去回数、消去間隔が概ね等しくなるように、消去する論理ブロックを均等に分散させる(ウェアレベリング処理)ことが可能となる。NAND型フラッシュメモリの寿命は、消去回数の他に、消去処理の間隔にも依存しており、その間隔が長いほどリテンション特性が良く、寿命が延びることが知られている。これは、消去間隔が短いとリテンション特性が悪く、寿命が損なわれることも示している。また、短い間隔で書き込みを行ったとしても、相応の長期間消去を行わなければリテンション特性が回復することも知られている。

40

【0140】

AB管理テーブル230は、フリーブロックFBから割り当てられた、用途が割り当てられた論理ブロック(アクティブブロックAB)のリストであり、FB管理テーブル220と同様に、各エントリは、論理ブロックIDを保持する。登録順序が古い論理ブロックほど先頭に位置することになる。AB管理テーブルは、例えば、リフレッシュ処理に使用

50

される。

【 0 1 4 1 】

リフレッシュ処理とは、書き込んだデータの経年変化や読み出し処理に伴うデータ破損であるリードディスタブの影響で、SSD100の誤り訂正能力を超える誤りが発生することを防止するための技術である。具体的には、例えば、誤り訂正能力を超える誤りが発生する前に、記憶してあるデータを読み出して誤り訂正を行い、その後再びNAND型フラッシュメモリにデータを書き直すという処理を行う。例えば、読み出し回数50dの多いブロック、AB管理テーブル230の先頭ブロックなどをリフレッシュ処理の監視対象とすることができる。

【 0 1 4 2 】

物理NAND層管理部120cは、以下のような論理ブロック - 物理ブロック管理を行う。まず、図27を用いて論物変換テーブル50を用いた、論理ブロックIDと8個の物理ブロックIDとの対応関係について説明する。

【 0 1 4 3 】

前述したように、論物変換テーブル50の物理ブロックID50bのフィールドには、論物変換テーブル50のインデックスである論理ブロックID50aに対応付けられた8個の物理ブロックIDが登録されている。図27は、NANDメモリ10の論理ブロックIDと物理ブロックIDとの対応関係を示すものであり、1つの区画が1物理ブロックを表している。各物理ブロックに物理ブロックIDが割り当てられている。論理ブロックL0は、例えば、1行3列、2行2列、3行2列、4行2列、5行2列、6行2列、7行2列、8行3列の8個の物理ブロックで構成されている。また、破線BL1で囲まれた論理ブロックL1は、例えば、1行4列、2行3列、3行3列、4行3列、5行3列、6行3列、7行3列、8行4列の8個の物理ブロックで構成されているとする。

【 0 1 4 4 】

この後、例えば、論理ブロックL1の4行3列の物理ブロックが、記憶領域として使用できないバッドブロックBBとして、BB管理テーブル200に登録されたとする。これを検知した物理NAND層管理部120cは、RB管理テーブル210から、このバッドブロックBBとして登録された物理ブロックと同一チャンネルで同一プレーンの予約ブロックRBを当該バッドブロックBBに対する入れ替え候補として選択する。図27の場合は、隣接する4行4列の物理ブロック(予約ブロックRB)が4行3列のBBに対する入れ替え候補として選択されている。

【 0 1 4 5 】

物理NAND層管理部120cは、論物変換テーブル50の論理ブロックL1に対応する論理ブロックID50aのエントリを検索し、このエントリ中の物理ブロックID50bのフィールドに含まれる8個の物理ブロックIDのうちの4行3列に対応するバッドブロックBBの物理ブロックIDを、RB管理テーブル210から選択した4行4列のRBに対応する物理アドレスIDに変更する。これにより、この後、論理ブロックL1は、一点鎖線で囲まれた1行4列、2行3列、3行3列、4行4列、5行3列、6行3列、7行3列、8行4列の8個の新たな物理ブロックの組み合わせで構成されることになる。この論理ブロックL1の論理ブロックIDが「L1」であるとする。

【 0 1 4 6 】

その後、物理NAND層管理部120cは、FB管理テーブル220から新たなフリーブロックFBを確保する。この確保したフリーブロックFBの論理ブロックIDが「L3」であるとする。物理NAND層管理部120cは、論物変換テーブル50を用いて論理ブロックIDの入れ替えを実行する。具体的には、論理ブロックID「L1」に対し、論理ブロックIDが「L3」である新たなフリーブロックFBに対応付けられた8個の物理ブロックを対応付け、さらに論理ブロックID「L3」に対し、前述の物理ブロックの組み合わせが変更された一点鎖線で囲まれた1行4列、2行3列、3行3列、4行4列、5行3列、6行3列、7行3列、8行4列の8個の物理ブロックを対応付ける。消去回数50c、読み出し回数50d、消去時刻50eも論理ブロックID入れ替えに伴い入れ替え

10

20

30

40

50

る。その後、論理ブロックID「L3」を、FB管理テーブル220に登録する。

【0147】

一方、バッドブロックBBに対して入れ替え可能な予約ブロックRBが存在しない場合には以下のような処理を行う。例えば、論理ブロックL1の4行3列の物理ブロックがバッドブロックBBとしてBB管理テーブル200に登録されたとし、さらに、このバッドブロックBBに対しては同一チャネルかつ同一プレーン内に予約ブロックRBが存在しない場合には、まず、論理ブロックL1内の4行3列を除く、1行4列、2行3列、3行3列、5行3列、6行3列、7行3列、8行4列の7個の物理ブロックをRB管理テーブル210に登録する。その後は、前記同様、FB管理テーブル220から新たなフリーブロックFBを確保し、前記同様の論理ブロックIDの入れ替えを実行した後、FB管理テ

10

【0148】

このように、物理NAND層管理部120cは、バッドブロックBBが発生した場合でも、論理ブロックIDの入れ替えを実行しているので、論理NAND層管理部120bで使用する論理ブロックIDは、バッドブロックBBの発生の前後で変化しない。したがって、複数の物理ブロックの少なくとも1つをバッドブロックとして登録する場合であっても、LBA論理アドレスと論理ブロックとの対応関係は変更されず、論理NAND層管理部120bでの管理テーブルの書き換えのオーバーヘッドを回避することができる。

【0149】

つぎに、32ビット倍速モードでの消去処理について説明する。物理NAND層管理部120cは、NANDメモリ10のデータが論理ブロック単位で消去されるごとに、図24に示す論物変換テーブル50の消去された論理ブロックに対応する論理ブロックID中の消去回数50cのフィールドを1カウントアップするとともに、消去時刻50eを最新のデータに更新する。

20

【0150】

・論理NAND層

つぎに、図28、図29を用いて論理NAND層での管理に使用されるMS構造管理テーブル60およびFS/IS構造管理テーブル65について説明する。図28に示すMS構造管理テーブル60は、MS部11Qの構造を管理するための領域と、状態情報を保存するための領域とを有する。MS構造管理テーブル60は、MSFB11aa、MS追記IB11ab、TFS11bとして割り当てられた論理ブロックIDを管理するMSバッファ管理テーブル61と、MSコンパクション時のソート処理を高速化するために、有効トラック数が少なくなった論理ブロックIDを保持する有効トラック数別論理ブロックIDリスト62と、状態情報としての最大論理ブロック数MBLおよび有効論理ブロック数VBLを管理する領域63、64とを備える。

30

【0151】

MS構造管理テーブル60では、MSFB11aa用、MS追記IB11ab用、TFS11b用に、それぞれ所要エントリ数の固定フィールド61a~61cが用意され、各固定フィールド61a~61cに論理ブロックIDが記録される。TFS11b用のフィールド61cは、リンクドリスト構造を有し、FIFO構造を有するTFS11bに対するFIFO的な管理が行われている。

40

【0152】

有効トラック数別論理ブロックIDリスト62では、有効トラック数が1個の論理ブロック用として所要数のエントリが用意され、有効トラック数が2個の論理ブロック用として所要数のエントリが用意され、...、有効トラック数が $2^i - 1$ の論理ブロック用として所要数のエントリが用意されており、各エントリには、論理ブロックIDが記録される。MS論理ブロック管理テーブル35の有効トラック数35cのフィールドがサーチされることによって有効トラック数別論理ブロックIDリスト62は、常に最新の状態に更新される。なお、有効トラック数別論理ブロックIDリスト62には、MSIB11a、TFS11bとしてMSバッファ管理テーブル61に登録されている論理ブロックについては

50

、エントリされない。

【 0 1 5 3 】

状態情報としての最大論理ブロック数 M B L 用の固定フィールド 6 3 には、 M S 部 1 1 Q が取得するのを許容されている論理ブロック数である最大論理ブロック数 M B L が記録される。状態情報としての有効論理ブロック数 V B L 用の固定フィールド 6 4 には、 M S 部 1 1 Q として現在管理されている論理ブロック数としての有効論理ブロック数 V B L が記録される。

【 0 1 5 4 】

図 2 9 に示す F S / I S 構造管理テーブル 6 5 は、 F S 部 1 2 Q および I S 部 1 3 Q の構造を管理するための領域を有する。 F S / I S 構造管理テーブル 6 5 は、 F S I B 1 2 a および F S 追記 I B 1 2 a b として割り当てられた論理ブロック I D を管理する F S 入力バッファ管理テーブル 6 6 と、 F S 1 2 の F I F O 構造を管理する F S F I F O 管理テーブル 6 7 と、 I S I B 1 3 a として割り当てられた論理ブロック I D を管理する I S 入力バッファ管理テーブル 6 8 と、 I S 1 3 の F I F O 構造を管理する I S F I F O 管理テーブル 6 9 とを備える。

【 0 1 5 5 】

F S 入力バッファ管理テーブル 6 6 は、 F S F B 1 2 a a 用、 F S 追記 I B 1 2 a b 用に、それぞれ所要エントリ数の固定フィールド 6 6 a、 6 6 b が用意され、各固定フィールド 6 6 a、 6 6 b に F S / I S 論理ブロック管理テーブル 4 2 のインデックスである F S / I S ブロック I D 4 2 a が登録される。 I S 入力バッファ管理テーブル 6 8 は、 I S I B 1 3 a 用に所要エントリ数の固定フィールドが用意され、各固定フィールドに、 F S / I S ブロック I D 4 2 a が登録される。 F S F I F O 管理テーブル 6 7 には、 F S 1 2 の F I F O 構造を形成する論理ブロック数分のエントリが固定フィールドに用意され、 F S F I F O 管理テーブル 6 7 の各固定フィールドには、 F S / I S ブロック I D 4 2 a が登録される。 I S F I F O 管理テーブル 6 9 には、 I S 1 3 の F I F O 構造を形成する論理ブロック数分のエントリが固定フィールドに用意され、各固定フィールドには、 F S / I S ブロック I D 4 2 a が登録される。

【 0 1 5 6 】

つぎに、先の図 1 9 を用いて説明した、 2 ステージ (W C F 処理および C I B 処理) に分割された書き込み処理が実行されるとき、 W C 2 1 から M S I B 1 1 a への C o p y 処理に伴う各管理テーブルの更新処理を説明する。なお、ここでは、 W C 2 1 から M S 追記 I B 1 1 a b へのトラック単位の C o p y が発生した場合について説明する。 D R A M 層管理部 1 2 0 a では、 W C トラック管理テーブル 2 4 を先頭から順にチェックし、追い出し確定トラックに対応する論理トラックアドレス 2 4 a が登録されたトラックエントリ中の way - line ビットマップ 2 4 c を参照し、 way - line ビットマップ 2 4 c の m x n のエントリ中の Valid ビットが “ 1 ” のエントリに対応する W C クラスタ管理テーブル 2 5 中のエントリにおける状態フラグ 2 5 a を Valid から N A N D への追い出し待ちにし、論理 N A N D 層管理部 1 2 0 b に対し追い出し要求を通知する。

【 0 1 5 7 】

一方、論理 N A N D 層管理部 1 2 0 b では、図 2 8 に示す M S 構造管理テーブル 6 0 の M S バッファ管理テーブル 6 1 および図 1 4 に示す M S 論理ブロック管理テーブル 3 5 を参照し、 M S 追記 I B 1 1 a b の状態をチェックする。すなわち、 M S バッファ管理テーブル 6 1 の M S 追記 I B 用のフィールド 6 1 b から M S 追記 I B 1 1 a b が既に存在していると判断されたときは、 M S 追記 I B 用のフィールド 6 1 b に登録されている論理ブロック I D についての書き込み可能トラック数に関する情報を M S 論理ブロック管理テーブル 3 5 の有効トラック数 3 5 c のフィールドから取得し、該取得した書き込み可能トラック数を D R A M 層管理部 1 2 0 a に通知する。

【 0 1 5 8 】

また、論理 N A N D 層管理部 1 2 0 b は、 M S バッファ管理テーブル 6 1 の M S 追記 I B 用のフィールド 6 1 b から M S 追記 I B 1 1 a b が存在していないと判断されたときは

、物理NAND層管理部120cに対し、フリーブロックFBの獲得要求を出して、フリーブロックFBを該FBとして割り当てられた論理ブロックIDと共に取得する。そして、取得したフリーブロックFBの書き込み可能トラック数 2^i をDRAM層管理部120aに通知する。

【0159】

DRAM層管理部120aでは、論理NAND層管理部120bから通知された書き込み可能トラック数だけ、WCトラック管理テーブル24からトラックを選択し、さらに前述のトラック内セクタ穴埋めと受動マージを行うか否かを判断する。ここでは、トラック内セクタ穴埋めを行う必要はないとする。DRAM層管理部120aは、追い出すトラックがNANDメモリ10に存在しているか否かをチェックして受動マージを行うか否かを調べるために、論理NAND層管理部120bに対し追い出しを行う論理トラックアドレスなどの所要の情報を通知する。

10

【0160】

この通知を受けると、論理NAND層管理部120bは、トラック管理テーブル30のインデックスである論理トラックアドレス30a、必要な場合はさらにFS/IS管理テーブル40を検索し、NANDメモリ10側に追い出しを行う論理トラックアドレスと同一の論理トラックが存在しているかをサーチし、このサーチ結果を物理NAND層管理部120cに通知する。これによって、物理NAND層管理部120cによって、受動マージを伴うあるいは受動マージを伴わないWC21からMS追記IB11abへの追い出しが行われる。

20

【0161】

物理NAND層管理部120cから、WC21 MS追記IB11abの追い出し終了が通知されると、論理NAND層管理部120bは、MS追記IB11abとしてあらたなフリーブロックFBを物理NAND層管理部120cから取得した場合は、物理NAND層管理部120cから与えられたフリーブロックFBの論理ブロックIDに対応するMS論理ブロック管理テーブル35のエントリのValidフラグ35eをValidにし、MSバッファ管理テーブル61のMS追記IB用のフィールド61bに論理ブロックIDを登録し、MS構造管理テーブル60の有効論理ブロック数VBLをインクリメントする。

【0162】

また、論理NAND層管理部120bは、トラック管理テーブル30を更新する。すなわち、WC21からMS追記IB11abへ追い出されたトラックに対応する論理トラックアドレス30aのエントリに対し、クラスタビットマップ30b、論理ブロックID30cおよび論理ブロック内トラック位置30dなどの所要情報を登録する。

30

【0163】

受動マージが行われていない場合は、前述のように、トラック管理テーブル30の対応する論理トラックエントリに所要情報を登録する。また、WC21から追い出された新しいトラックに関するデータをMS論理ブロック管理テーブル35の書き込みを行った論理ブロックIDに対応するエントリに対し登録する。MS論理ブロック管理テーブル35への登録としては、MS論理ブロックに格納されたトラックに対応するトラック管理テーブル30のインデックスとしての論理トラックアドレス(トラック管理ポインタ35b)の更新と、有効トラック数35cの更新と、書き込み可能先頭トラック35dの更新などがある。

40

【0164】

WC21とMS11との受動マージが行われたとすると、トラック管理テーブル30中のマージ元トラックに対応する論理トラックアドレス30aのエントリ中の論理ブロックID30cおよび論理ブロック内トラック位置30dなどの所要情報が更新される。具体的には、論理ブロックID30cがMS11に存在していたときの論理ブロックIDからMS追記IB11abに対応する論理ブロックIDに変更され、論理ブロック内トラック位置30dが追記書き込み状態に応じて変更される。

【0165】

50

さらに、MS論理ブロック管理テーブル35中のマージ元の論理ブロックに対応するエントリにおけるトラック管理ポインタ35bのフィールドの該当箇所を削除し、有効トラック数35cをデクリメントし、さらにMS構造管理テーブル60の有効トラック数別論理ブロックIDリスト62を更新する。マージ元の論理ブロックに対応するエントリにおける有効トラック数35cがデクリメントにより0になった場合は、MS構造管理テーブル60の有効論理ブロック数VBLをデクリメントし、物理NAND層管理部120cに対しこの書き込み不可能な無効なフリーブロックFBを返却する。そして、この返却した論理ブロックに対応するエントリのValidフラグ35eをInvalidにする。さらに、前述と同様にして、WC21から追い出された新しいトラックに関するデータをMS論理ブロック管理テーブル35に登録する。

10

【0166】

また、物理NAND層管理部120cから、WC21 MS追記IB11abの追い出し終了が通知されたとき、論理NAND層管理部120bは、DRAM層管理部120aに対し追い出し終了を通知する。この通知を受けたDRAM層管理部120aでは、WCクラスタ管理テーブル25中の追い出されたトラックに所属する全クラスタに対応するエントリ中の状態フラグ25aはInvalid(使用可能)とされ、その後これらエントリに対する書き込みが可能となる。また、WCトラック管理テーブル24中の追い出されたトラックに対応するリストについては、例えば直前リストのnextポインタ24dが変更または削除されて、無効化される。

【0167】

20

つぎに、CIB処理について説明する。WCF処理が終了すると、WCF処理によって書き込まれたFSIB12aのデータをFS12にMoveする処理と、WCF処理によって書き込まれたMSIB11aのデータをMS11またはTFS11bにMoveする処理などを含むCIB処理が実行される。以下、図30のフローチャートを用いてCIB処理の詳細手順について説明する。

【0168】

・MS部11QのCIB処理

まず、図19のステップS330で説明したMS部11Qでの第1回目のCIB処理について詳述する。論理NAND層管理部120bは、MS構造管理テーブル60のMSバッファ管理テーブル61のMSFB用フィールド61a、MS追記IB用のフィールド61bに登録されている論理ブロックIDについての有効トラック数の情報をMS論理ブロック管理テーブル35の有効トラック数35cのフィールドから取得し、MSIB11aのMSFB11aまたはMS追記IB11abに全トラック書き込み済みのフルになったブロックが一つ以上存在するか否かを調べ(ステップS400)、MSIB11aにフルになったブロックが一つ以上存在する場合に、以下の処理を行う。ステップS400の判定がNOである場合は、手順はステップS440に移行する。

30

【0169】

ステップS400の判断がYESの場合、論理NAND層管理部120bは、MS論理ブロック管理テーブル35の有効トラック数35cを参照することにより、MS内に有効トラック数35cが0となった無効論理ブロックが存在するか否かを調べ、MS内に無効論理ブロックが存在する場合、該無効論理ブロックを物理NAND層管理部120cに返却する(ステップS405)。返却された無効論理ブロックに対応するMS論理ブロック管理テーブル35のエントリはValidフラグ35eがInvalidにされ、MS構造管理テーブル60の有効論理ブロック数VBLがデクリメントされる。つぎに、MSFB11a中のフルになった論理ブロックは、直接MS11にMoveされ、MS追記IB11ab中のフルになった論理ブロックはTFS11bにMoveされる(ステップS407)。このMove処理は、具体的には、MS構造管理テーブル60のMSバッファ管理テーブル61のMSFB用フィールド61a、MS追記IB用のフィールド61bに登録されている該当する論理ブロックIDを削除するのみである。

40

【0170】

50

つぎに、論理NAND層管理部120bは、MS構造管理テーブル60の状態情報としての有効論理ブロック数VBLを最大論理ブロック数MBLと比較する(ステップS410)。この比較の結果、有効論理ブロック数VBLが最大論理ブロック数MBLを越えている場合、論理NAND層管理部120bは、書き込み可能なフリーブロックFBが不足していると判断し、以下のMSコンパクション処理を1ブロックずつ実行して、全て無効なトラックで構成された物理NAND層管理部120cに返却すべき無効論理ブロックを増やし、有効論理ブロック数VBLを最大ブロック数未満MBLとする(ステップS420)。なお、ステップS410の判定において、書き込み可能なFBが不足していない場合は、手順はステップS440に移行される。

【0171】

MSコンパクション処理には、前述したように、 2^i トラックMSコンパクションと、 2^i トラック未満MSコンパクションとの2種類がある。 2^i トラックMSコンパクションでは、MSコンパクションバッファ11cが使用されて、コンパクション後の論理ブロックは、TFS11bの先頭にMoveされる。 2^i トラック未満MSコンパクションでは、トラック単位に、MS追記IB11abにCopyされる。

【0172】

まず、論理NAND層管理部120bは、MS構造管理テーブル60の有効トラック数別論理ブロックIDリスト62を参照することで、有効トラック数の少ない論理ブロックから、トラックを 2^i 個集めて、集めたトラックを 2^i 個ずつ、書き込み可能なフリーブロックFBとして物理NAND層管理部120cから取得したMSコンパクションバッファ11cに対しCopyする、 2^i トラックMSコンパクションを実行する。

【0173】

具体的には、論理NAND層管理部120bは、物理NAND層管理部120cに対し、フリーブロックFBの獲得要求を出して、フリーブロックFBを該フリーブロックFBとして割り当てられた論理ブロックIDと共に取得する。そして、コンパクション対象として選択した複数のトラックをこのフリーブロックFBにCopyするよう、物理NAND層管理部120cに対し、要求する。なお、このコンパクション対象のトラックがWC21、FS部12Q、IS部13Qに有効クラスタを持つ場合にはそれらもマージしてMSコンパクションバッファ11cに集められる、前述の受動マージが実行される。

【0174】

物理NAND層管理部120cからコンパクションの終了が通知されると、論理NAND層管理部120bは、トラック管理テーブル30のコンパクションが行われたトラックに対応する論理トラックアドレス30aを持つエントリにおける、論理ブロックID30cを物理NAND層管理部120cから取得したフリーブロックFBの論理ブロックIDに更新し、さらに論理ブロック内トラック位置30dを更新する。

【0175】

また、MSコンパクションバッファ11cとして使用した、物理NAND層管理部120cから取得したフリーブロックFBの論理ブロックIDをMS論理ブロック管理テーブル35に新たなエントリとして登録し、エントリ中の各フィールドに所要の情報を登録する。この登録としては、トラック管理ポインタ35bの更新と、有効トラック数35cの更新と、書き込み可能先頭トラック35dの更新などがある。

【0176】

また、MS構造管理バッファ60のMSバッファ管理テーブル61のTFS用フィールド61cのFILO構造(リンクリスト)の先頭に、MSコンパクションバッファ11cとして使用した論理ブロックIDを登録することで、MSコンパクションの結果、有効な 2^i 個のトラックを含む1論理ブロックで構成されたMSコンパクションバッファ11cを、TFS11bの先頭(最も古い位置)にMoveする。なお、TFS11bがフルになったら、最も古い先頭のブロックがMS11にMoveされる。

【0177】

つぎに、MS11中のコンパクション元の古いトラックデータを無効化する。具体的に

10

20

30

40

50

は、MS論理ブロック管理テーブル35中のコンパクション元の論理ブロックに対応するエントリにおけるトラック管理ポイント35bのフィールドの該当箇所を削除し、有効トラック数35cをデクリメントし、さらにMS構造管理テーブル60の有効トラック数別論理ブロックIDリスト62を更新する。有効トラック数35cがデクリメントにより0になった場合は、MS構造管理テーブル60の有効論理ブロック数VBLをデクリメントし、物理NAND層管理部120cに対しこの無効論理ブロックを返却する。そして、この返却した論理ブロックに対応するMS論理ブロック管理テーブル35のエントリはValidフラグ35eがInvalidにされる。

【0178】

このようなコンパクション処理および無効論理ブロックFBの返却処理が終了すると、有効論理ブロック数VBLと最大論理ブロック数MBLを比較し、有効論理ブロック数VBLが最大論理ブロック数MBLを越えている場合は、再度、有効トラックを 2^i 個集める 2^i トラックMSコンパクションを実行する。また、有効論理ブロック数VBLが最大論理ブロック数MBLを越えている状態で、有効トラックを 2^i 個集める 2^i トラックMSコンパクションが不可能になった場合は、 2^i トラック未満MSコンパクションが実行される。

【0179】

この 2^i トラック未満MSコンパクションでは、コンパクション対象の 2^i トラックに満たない個数のトラックをMS追記IB11abにコピーすることで、無効な 2^i トラックで構成された無効論理ブロックを生成し、生成した無効論理ブロックを物理NAND層管理部120cに返却することで、有効論理ブロック数VBLを減少させる。 2^i トラック未満MSコンパクションについての、管理テーブルの更新については、その説明を省略する。

【0180】

・FS12のCIB処理

つぎに、図19のステップS340で説明したFS12でのCIB処理について詳述する。論理NAND層管理部120bは、FS/IS構造管理テーブル65のFS入力バッファ管理テーブル66のFSFB用フィールド66a、FS追記IB用のフィールド66bに登録されている論理ブロックIDについての有効クラスタ数の情報を、FS/IS論理ブロック管理テーブル32の有効クラスタ数42dのフィールドから取得し、FSIB12aのFSFB12aaまたはFS追記IB12abに全ページ(全クラスタ)書き込み済みのフルになった論理ブロックが一つ以上存在するか否かを調べ(ステップS440)、FSIB12aにフルになった論理ブロックが一つ以上存在する場合に、以下の処理を行う。ステップS440の判定がNOである場合は、手順はここで終了する。

【0181】

ステップS440の判断がYESの場合、論理NAND層管理部120bは、FS/IS構造管理テーブル65およびFS/IS論理ブロック管理テーブル42の有効クラスタ数42dを参照することにより、FS部12Q内に有効クラスタ数42dが0である無効論理ブロックが存在するか否かを調べ、FS部12Q内に無効論理ブロックが存在する場合、該無効論理ブロックを物理NAND層管理部120cに返却する(ステップS445)。

【0182】

返却された論理ブロックのエントリはMS論理ブロック管理テーブル35およびFS/IS論理ブロック管理テーブル42から削除される。つぎに、FSFB12aa中のフルになった論理ブロックは、FS12にMoveされ、FS追記IB12ab中のフルになった論理ブロックもFS12にMoveされる(ステップS447)。このMove処理は、具体的には、FS/IS構造管理テーブル65のFS入力バッファ管理テーブル66のFSFB用フィールド66a、FS追記IB用のフィールド66bに登録されている該当する論理ブロックIDを削除するのみである。

【0183】

10

20

30

40

50

つぎに、論理NAND層管理部120bは、FIFO構造を有するFS12の論理ブロック数が、FS12として許容される所定の最大論理ブロック数BLfsmxを越えているか否かを判断する(ステップS450)。具体的には、FSFIFO管理テーブル67から算出した論理ブロック数が予め設定した最大論理ブロック数BLfsmxを越えているか否かを判断する。

【0184】

この比較の結果、算出した論理ブロック数が最大論理ブロック数BLfsmxを越えている場合、論理NAND層管理部120bは、書き込み可能なフリーブロックFBが不足していると判断し、そのときの状態に応じて、以下の、MS11への例えば2論理ブロック分ずつの追い出し処理(ステップS460)と、IS13への1論理ブロック分の追い出し処理を実行する(ステップS500)。なお、ステップS450の判定において、FS12内がフルでない場合は、FS12からMSIB11aへの追い出し処理と、FS12からISIB13aへの追い出し処理を行うことなく、手順をここで終了する。

10

【0185】

FS12からMSIB11aへの追い出し処理では、まず、FS12からIS部13Qを経ることなくMSIBS11aに直接移動する論理ブロックがあるか否かを判断する(ステップS455)。具体的には、FS12のFIFOの先頭にある最古の論理ブロック中のクラスタを順に1つずつチェックし、当該クラスタが属するトラックがFS部12Q中に何個の有効クラスタを保有しているか、トラック管理テーブル30のFSクラスタ数30fのフィールドを参照して検索する。そして、トラック内有効クラスタ数が所定の閾値(例えば 2^{k-i-1} 個)以上だった場合、その論理トラックをMSIB11aへの追い出し確定トラックとする。

20

【0186】

上記の検索は、以下のルートを経る。

1. FS/IS構造管理テーブル65のFSFIFO管理テーブル65からFIFOの先頭にある最古のFS/ISブロックIDを得る。

2. このFS/ISブロックIDに対応するFS/IS論理ブロック管理テーブル42のエントリ中のブロック内クラスタテーブル42cのフィールドからFS/IS内クラスタ管理テーブル44へのインデックスを得る。

3. FS/IS内クラスタ管理テーブル44において、取得したインデックスで指定された1論理ブロック内の各エントリからFS/IS管理テーブル40へのポインタを1つ取得して、FS/IS管理テーブル40の該当リンクに飛ぶ。

30

4. 飛び先の該当リンクが所属する該当論理トラックアドレスを得る。

5. 取得した論理トラックアドレスを使ってトラック管理テーブル30の該当エントリ中のFSクラスタ数30fのフィールドをチェックする。

6. 3~5が繰り返される。

【0187】

FS12からMS11への追い出しは、例えば2論理ブロックずつ行われる。すなわち、論理NAND層管理部120bは、上記トラック内有効クラスタ数が所定の閾値(例えば 2^{k-i-1} 個)以上のトラックを、2論理ブロック分集め、該集めた2論理ブロック分のトラックをMSIB11aのMSFB11aに追い出す(ステップS460)。この追い出しの際、追い出しトラック中のFS12内に存在しないクラスタについては、WC21、IS部13Q、およびMS11から読み出してマージする前述の受動マージが実行される。

40

【0188】

ただし、MSIB11aへの追い出し確定トラックが、2論理ブロック分存在しない場合は、1論理ブロックをMSIB11aのMSFB11aに追い出し、1論理ブロック分に満たない個数のトラックをMS追記IB11abに対してトラック単位に追記書き込みする(ステップS460)。同様に、MSIB11aへの追い出し確定トラックが、1論理ブロック分存在しない場合は、1論理ブロック分に満たない個数のトラックをMS追

50

記 I B 1 1 a b に対してトラック単位に追記書き込みする (ステップ S 4 6 0)。その後、F I F O 構造の F S 1 2 の先頭論理ブロックに有効なクラスタが残っていない場合には、先頭論理ブロックを無効論理ブロックとして、物理 N A N D 層管理部 1 2 0 c に返却する。

【 0 1 8 9 】

・ M S 1 1 の C I B 処理 (図 1 9 : ステップ S 3 5 0)

このようにして、F S から M S I B 1 1 a への追い出しが発生すると、つぎに、M S 部 1 1 Q の C I B 処理が再び実行される (ステップ S 4 8 0)。このステップ S 4 8 0 の M S 部 1 1 Q の C I B 処理は、先の第 1 回目の M S 部 1 1 Q での C I B 処理 (ステップ S 4 0 0 ~ S 4 2 0) と同様なので、重複する説明は省略する。この M S 部 1 1 Q での C I B 10
処理の後、論理 N A N D 層管理部 1 2 0 b は、F S 1 2 から M S I B 1 1 a への追い出し条件が成立するか否かを、前記同様にして調べる (ステップ S 4 5 5)。この追い出し条件が成立すると、前述した、F S 1 2 から M S I B 1 1 a への 2 論理ブロックの追い出し、および M S 1 1 での C I B 処理が、再度実行される。このような処理が、ステップ S 4 5 5 の判断が N O になるまで繰り返される。

【 0 1 9 0 】

・ F S 1 2 の C I B 処理

ステップ S 4 5 5 の判断が N O になると、論理 N A N D 層管理部 1 2 0 b は、つぎに、F S 1 2 から I S I B 1 3 a への追い出し条件が、成立するか否かを判断する (ステップ S 4 9 0)。具体的には、上記 F S 1 2 から M S I B 1 1 a への追い出し処理の際に、チ 20
ェックされた F I F O 構造を有するフル状態の F S 1 2 の先頭論理ブロックに有効なクラスタが残っている場合に、ステップ S 4 9 0 での F S 1 2 から I S 1 3 への追い出し条件が成立したとして、F S 1 2 から I S I B 1 3 a への追い出しを実行する。

【 0 1 9 1 】

すなわち、ステップ S 4 9 0 で条件が成立した場合は、M S I B 1 1 a への追い出しトラックに含まれないクラスタだけを含む先頭論理ブロックを I S I B 1 3 a へ M o v e する (ステップ S 5 0 0)。このステップ S 5 0 0 においては、例えば 1 論理ブロックの追 30
い出しが実行される。そして、状態によっては、その後、ステップ S 5 2 0 ~ S 5 8 5 の手順を経た後、ステップ S 5 9 0 の判断により、ステップ S 5 0 0 での F S 1 2 から I S I B 1 3 a への追い出しが再度発生することもある。

【 0 1 9 2 】

ステップ S 5 0 0 での再追い出しがある状態とは、例えば、F S I B 1 2 a に複数のフルになった論理ブロックを有するバッファ (F S F B 1 2 a a または F S 追記 I B 1 2 a b) が存在する状態において、F I F O 構造を有する F S 1 2 がフルであった場合には、F S I B 1 2 a から F S 1 2 へのフルになったブロックの M o v e に伴い、F S 1 2 から M S I B 1 1 a または I S I B 1 3 a への複数のブロックの追い出しが発生する状態である。このような条件下で、F S 1 2 から I S I B 1 3 a への複数の論理ブロックの追い出しが発生する可能性がある。

【 0 1 9 3 】

・ I S の C I B 処理 (図 1 9 ステップ S 3 6 0)

つぎに、ステップ S 4 9 0 での条件が成立したときに、I S 1 3 で行われる追い出し処理およびコンパクション処理の詳細について、図 3 0 の他に、図 3 1 のフローチャートを参照して説明する。まず、論理 N A N D 層管理部 1 2 0 b は、前記と同様にして、I S 部 1 3 Q 内に無効論理ブロックが存在するか否かを調べ、I S 部 1 3 Q 内に無効論理ブロックが存在する場合、該無効論理ブロックを物理 N A N D 層管理部 1 2 0 c に返却する (ステップ S 5 2 0)。返却された論理ブロックのエントリに対応する M S 論理ブロック管理 40
テーブル 3 5 および F S / I S 論理ブロック管理テーブル 4 2 のエントリは Valid フラグ 3 5 e および 4 2 f がそれぞれ I n V a l i d にされる。

【 0 1 9 4 】

つぎに、論理 N A N D 層管理部 1 2 0 b は、F I F O 構造を有する I S 1 3 の論理プロ 50

ック数が、IS13として許容される所定の最大論理ブロック数BLISmaxを越えているか否かを判断する(ステップS530)。具体的には、ISFIFO管理テーブル69から算出した論理ブロック数が予め設定した最大論理ブロック数BLISmaxを越えているか否かを判断する。

【0195】

この比較の結果、算出した論理ブロック数が最大論理ブロック数BLISmaxを越えている場合、論理NAND層管理部120bは、書き込み可能なフリーブロックFBが不足していると判断し、IS13から例えば2論理ブロック分のトラックをMSIB11aのMSFB11aaに追い出す(ステップS540)。なお、ステップS530の判定において、IS13がフルでない場合は、MSIB11aへの追い出し処理などを行うことなく、MSIB13a中のフルになった論理ブロックを、IS13bへMoveする(ステップS585)。

10

【0196】

ステップS540の追い出しの際には、図12に示したトラック管理テーブル30などを用いて、図31に示す追い出しトラックの選別処理を実行する。図31において、論理NAND層管理部120bは、選別処理(サイクリックサーチ処理、以下単にサーチ処理という)を開始すると(ステップS700)、前回のサーチの際に検索済み最終トラックとしてステップS740において記憶したトラック管理テーブル30のインデックスである論理トラックアドレス30aの次の論理トラックアドレスからサーチを開始する(ステップS710)。

20

【0197】

サーチが初回(第1サイクル)である場合は、トラック管理テーブル30の最初のエントリからサーチを開始する(ステップS710)。また、ステップS740で記憶した検索済みトラックがトラック管理テーブル30の最終エントリ(図12のトラックn)である場合は、ステップS710での次のトラック検索では、先頭エントリ(図12のトラック0)に戻る。

【0198】

このサーチにおいては、トラック管理テーブル30の当該エントリ中のISクラスタ数30gのフィールド(当該論理トラック中の有効クラスタ数)を参照し、当該エントリIS13内に有効クラスタを保持している場合は、図示しない新規検索トラックリストに当該エントリの論理トラックアドレスを登録する(ステップS720)。つぎに、新規検索トラックリストに登録されたトラックの個数を所定の閾値Lと比較し、登録数が閾値Lより少ない場合は、手順をステップS710に移行し、トラック管理テーブル30の次のエントリを前述と同様にして調べる。

30

【0199】

このような処理を繰り返すことにより、新規検索トラックリストに閾値L個分の論理トラックアドレスを登録する(ステップS730 Yes)。そして、最後に新規検索トラックリストに登録した論理トラックアドレスに対応するトラック管理テーブル30のエントリ(インデックス)を検索済み最終トラックとして記憶して、今回サイクルの検索を終了する(ステップS740)。

40

【0200】

つぎに、論理NAND層管理部120bは、前回落選の論理トラック(図示せず)がリストされている落選トラックリストがあるか否かを判断する(ステップS750)。初回サイクルの際は、落選トラックリストが存在しないので、新規検索トラックリストと新規追加ブロック内トラックリスト(図示せず)との2つのリストに基づいて 2^{i+1} 個の論理トラックを選出する(ステップS760)。新規追加ブロック内トラックリストとは、図30のステップS500で、FS12からIS部13Qに追い出されたブロック(FS/IS構造管理テーブル65のIS入力バッファ管理テーブル68にエントリされている)に含まれるトラックに関するリストである。

【0201】

50

初回サイクルの際は、このような2つのリストを用いて、追い出し候補の $\frac{2^{i+1}}{2}$ 個のトラックを選出する。この選出には、前述したように、トラック内の有効クラスタ数と、有効クラスタ係数とを用いた選出基準（スコア値）Sが用いられる。

スコア値S = トラック内の有効クラスタ数 × 有効クラスタ係数

有効クラスタ係数は、トラックがMS部11Q内で無効トラックが存在する論理ブロックに存在するか否かによって重み付けされる数であり、存在したほうが存在しない場合より数が大きいとする。

【0202】

有効クラスタ数は、トラック管理テーブル30のISクラスタ数30gのフィールドを見ることで取得することができる。また、有効クラスタ係数は、トラック管理テーブル30とトラック管理ポインタ35bのフィールドでリンクしているMS論理ブロック管理テーブル35の有効トラック数35cのフィールドを見ることで取得することができる。

10

【0203】

論理NAND層管理部120bは、新規追加ブロック内トラックリストに含まれる複数のトラックから、上記スコア値Sが上位のM個（所定の設定値）のトラックを選択する。選択されたM個のトラックに、先のサーチによって新規検索トラックリストに登録されたL個のトラックを加え、これらL+M個のトラック中から、スコア値Sが高い $\frac{2^{i+1}}{2}$ 個のトラックを、MS11への追い出しを行うトラックとして選出する。そして、L+M個のトラックのなかで、選出された $\frac{2^{i+1}}{2}$ 個のトラック以外のトラックを、前述の落選トラックリストに登録する。

20

【0204】

2回目以降のサイクルの際は、落選トラックリストと、新規検索トラックリストと、新規追加ブロック内トラックリストとの3つのリストに基づいて $\frac{2^{i+1}}{2}$ 個のトラックを選出する（ステップS770）。なお、2回目以降の、追い出しを行うか否かは、後述する図30のステップS570の判断に応じて決められる。3つのリストを用いた選出処理では、落選トラックリストに含まれる複数のトラックから、スコア値Sが上位のN個（所定の設定値）のトラックを選択し、新規追加ブロック内トラックリストに含まれる複数のトラックから、スコア値Sが上位のM個（所定の設定値）のトラックを選択し、これらN+M個のトラックに、さらに今回2回目以降のサイクルの際に得られた新規検索トラックリストに登録されたL個のトラックを加え、これらL+M+N個のトラック中から、スコア値Sが高い $\frac{2^{i+1}}{2}$ 個のトラックを、MS11への追い出しを行うトラックとして選出する。そして、L+M+N個の論理トラックのなかで、選出された $\frac{2^{i+1}}{2}$ 個のトラック以外のトラックを、次回サイクルに使用する落選トラックリストに登録する。

30

【0205】

つぎに、図30のステップS540に戻って説明する。前述のようにして、2論理ブロック分のトラックの追い出し候補が選出されると、論理NAND層管理部120bは、該選出された2論理ブロック分のトラック（すなわち $\frac{2^{i+1}}{2}$ 個のトラック）をMSIB11aのMSFB11aaに追い出す（ステップS540）。この追い出しの際、追い出しトラック中のIS部13Q内に存在しないクラスタについては、WC21およびFS部12、およびMS11から読み出してマージする前述の受動マージが実行される。なお、上記では、有効クラスタ数とMSに穴あきブロックが存在しているか否かの係数とに基づくスコア値Sによって追い出しトラックを選出するようにしているが、有効クラスタ数のみによって追い出しトラックを選出するようにしてもよい。

40

【0206】

・MSのCIB処理（図19ステップS370）

このようにして、IS13からMSIB11aへの追い出しが発生すると、つぎに、MS11のCIB処理が再び実行される（ステップS560）。このステップS560のMS11のCIB処理は、先の第1回目のMS11でのCIB処理（ステップS400～S420）と同様なので、重複する説明は省略する。

【0207】

50

・ I S の C I B 処理

つぎに、論理 N A N D 層管理部 1 2 0 b は、I S 1 3 から M S I B 1 1 a に対する追い出しを再度実行するか否かを判断する（ステップ S 5 7 0）。すなわち、論理 N A N D 層管理部 1 2 0 b は、M S 論理ブロック管理テーブル 3 5 および F S / I S 論理ブロック管理テーブル 4 2 の有効クラスタ数 4 2 d のフィールドなどを用いて、ステップ S 5 4 0 の追い出しを行った後の I S 1 3 内の論理ブロックを有効クラスタ数の少ない順にソートし、最も有効クラスタ数の少ない 2 つの論理ブロックの合計有効クラスタ数が、所定の設定値である 2^k 個（1 論理ブロック分）以上ある場合は、I S 1 3 から M S I B 1 1 a への追い出し条件が成立したと判断する（ステップ S 5 7 0）。

【 0 2 0 8 】

I S 1 3 から M S I B 1 1 a への追い出し条件が成立した場合は、手順をステップ S 5 4 0 に移行し、図 3 1 のステップ S 7 0 0 ~ S 7 5 0 および S 7 7 0 の処理を実行することで、上記の 2 論理ブロック分の追い出し処理を再度実行する。ステップ S 5 7 0 の判断が Y E S である限り、I S 1 3 から M S I B 1 1 a への 2 論理ブロック分の追い出し処理と、M S 1 1 での C I B 処理が繰り返し実行される。そして、ステップ S 5 7 0 の判断が N O となった場合は、I S 1 3 でのコンパクション処理が実行される（ステップ S 5 8 0）。

【 0 2 0 9 】

I S コンパクション処理では、M S 論理ブロック管理テーブル 3 5 および F S / I S 論理ブロック管理テーブル 4 2 の有効クラスタ数 4 2 d のフィールドなどを用いて、I S 部 1 3 Q 内の有効クラスタの少ない論理ブロックから順にクラスタを 1 論理ブロック分、すなわち 2^k 個集め、これら 2^k 個のクラスタを I S コンパクションバッファ 1 3 c に C o p y する。この C o p y 処理が終了すると、論理 N A N D 層管理部 1 2 0 b は、コンパクション元（C o p y 元）の論理ブロックのうち有効クラスタがなくなったものを無効論理ブロックとして物理 N A N D 層管理部 1 2 0 c に返却し、コンパクション処理によって有効クラスタが詰まった論理ブロックで構成される I S コンパクションバッファ 1 3 c を I S 1 3 に M o v e する。

【 0 2 1 0 】

このコンパクションの後、I S I B 1 3 a 中のフルになった論理ブロックは I S 1 3 に M o v e される（ステップ S 5 8 5）。この M o v e 処理は、具体的には、F S / I S 構造管理テーブル 6 5 の I S 入力バッファ管理テーブル 6 8 の I S I B 用フィールドに登録されている該当する論理ブロック I D を削除するのみである。

【 0 2 1 1 】

この後、論理 N A N D 層管理部 1 2 0 b は、前述した F S 1 2 から I S I B 1 3 a への追い出し条件が成立しているか否かを判断し（ステップ S 5 9 0）、この F S 1 2 から I S I B 1 3 a への追い出し条件が成立している場合は、手順をステップ S 5 0 0 に移行し、上述した手順を再度繰り返す。また、論理 N A N D 層管理部 1 2 0 b は、I S コンパクション処理の終了後、F S 1 2 から I S I B 1 3 a への追い出し条件が成立していないと判断した場合は、これで今回の書き込み処理を終了する。以上が、書き込み処理の詳細である。

【 0 2 1 2 】

つぎに、本実施の形態の要部について詳細に説明する。前述したように、 2^i トラック M S コンパクションまたは 2^i トラック未満 M S コンパクションを行なう際には、予め M S 論理ブロック管理テーブル 3 5 の有効トラック数 3 5 c のフィールドなどを参照することにより、M S I B 1 1 a および T F S 1 1 b に含まれない M S 1 1 中の論理ブロックのうち無効にされたトラックを持つものを、論理ブロック内の有効トラック数でソートしている。そこで、以下では M S 1 1 中の論理ブロックを有効トラック数によって効率良く管理し、コンパクションを高速化する方法について説明する。

【 0 2 1 3 】

前述したように、M S 1 1 のコンパクションには、図 1 4 に示した M S 論理ブロック管

10

20

30

40

50

理テーブル 3 5 が使用される。本実施の形態では、MS 1 1 のコンパクションの処理時間を短縮化するために、論理ブロック内の有効トラック数を、有効トラック数ごとのリンクドリスト（リンクリスト）で管理する。具体的には、MS 論理ブロック管理テーブル 3 5 内で、有効トラック数ごとのリンクドリストが形成される。

【 0 2 1 4 】

図 3 2 は、有効トラック数ごとのリンクドリストを有した MS 論理ブロック管理テーブルの一例を示す図である。図 3 2 に示すように、ここでの MS 論理ブロック管理テーブル 3 5 は、論理ブロック ID 3 5 a をインデックスとした配列形式で構成され、各エントリは、トラック管理ポインタ 3 5 b、有効トラック数 3 5 c、書き込み可能先頭トラック 3 5 d、Valid フラグ 3 5 e の他に、prev ポインタ 3 5 g と next ポインタ 3 5 h とが追加されて構成されている。

10

【 0 2 1 5 】

前述したように、有効トラック数 3 5 c は、論理ブロック ID 3 5 a で指定される当該論理ブロックに記憶されているトラックのうちの有効なもの個数（最大 2^i 個）を示している。

【 0 2 1 6 】

また、MS 論理ブロック管理テーブル 3 5 では、prev ポインタ 3 5 g と next ポインタ 3 5 h とによって、有効トラック数毎にリンクドリスト構造が形成されている。next ポインタ 3 5 h は、有効トラック数毎にリンクされる次のエントリへのポインタを示し、prev ポインタ 3 5 g は、有効トラック数毎にリンクされる 1 つ前のエントリへのポインタを示している。

20

【 0 2 1 7 】

図 3 3 は、リンクドリストの構造を説明するための図である。有効トラック数 1、有効トラック数 2、...、有効トラック数 2^i の各リンクドリストは、同様の構造を有しているので、ここではリンクドリストの一例として有効トラック数 1 のリンクドリストについて説明する。

【 0 2 1 8 】

MS 論理ブロック管理テーブル 3 5 は、有効トラック数 1 のリスト先頭ポインタを有している。リスト先頭ポインタは、MS 論理ブロック管理テーブル 3 5 の各有効トラック数の先頭エントリへのポインタである。MS 論理ブロック管理テーブル 3 5 の末尾には、リンクドリスト管理用の特別なエントリ（図示せず）を設けておき、このエントリに各有効トラック数に対応するリスト先頭ポインタを格納しておく。

30

【 0 2 1 9 】

有効トラック数 1 の論理ブロック ID を検索する場合には、まず有効トラック数 1 のリスト先頭ポインタに基づいて、有効トラック数 1 の先頭のエントリを検索する。図 3 3 では、ブロック 0 が有効トラック数 1 の先頭のエントリである場合を示している。

【 0 2 2 0 】

MS 論理ブロック管理テーブル 3 5 では、各論理ブロック ID 3 5 a が prev ポインタ 3 5 g や next ポインタ 3 5 h と対応付けられており、prev ポインタ 3 5 g や next ポインタ 3 5 h によって、前のエントリや次のエントリが示されている。

40

【 0 2 2 1 】

図 3 3 では、ブロック 0 の next ポインタ 3 5 h が、有効トラック数 1 の次のエントリ（ブロック 0 の次のエントリ）としてブロック 8 のエントリを示している。また、ブロック 8 の next ポインタ 3 5 h は、有効トラック数 1 の次のエントリ（ブロック 8 の次のエントリ）としてブロック X のエントリを示している。また、ブロック X の prev ポインタ 3 5 g は、有効トラック数 1 の 1 つ前のエントリとしてブロック 8 のエントリを示している。また、ブロック 8 の prev ポインタ 3 5 g は、有効トラック数 1 の 1 つ前のエントリとしてブロック 0 のエントリを示している。

【 0 2 2 2 】

論理 NAND 層管理部 1 2 0 b は、有効トラック数 1 の先頭のエントリとしてブロック

50

0を検索すると、ブロック0のnextポインタ35hに基づいて、有効トラック数1の次のエントリとしてブロック8を検索する。さらに、論理NAND層管理部120bは、有効トラック数1のエントリとしてブロック8を検索すると、ブロック8のnextポインタ35hに基づいて、有効トラック数1の次のエントリとしてブロックXを検索する。このように、論理NAND層管理部120bは、有効トラック数1のエントリを順番に検索し、検索したエントリの論理ブロックを用いてコンパクション処理を行なう。

【0223】

図33では、有効トラック数1の論理ブロックを検索する場合について説明したが、MS論理ブロック管理テーブル35では、有効トラック数が1個の論理ブロック、有効トラック数が2個の論理ブロック、...、有効トラック数が 2^i 個(1論理ブロック分)の論理ブロックがそれぞれリンクドリスト構造を有している。したがって、論理NAND層管理部120bは、有効トラック数ごとに論理ブロックを検索することが可能である。

10

【0224】

このように、MS論理ブロック管理テーブル35内から有効トラック数ごとに論理ブロックを検索できるので、コンパクションに用いる適切な論理ブロック(有効トラック数の少ない論理ブロック)をMS論理ブロック管理テーブル35から高速に見つけ出すことが可能となる。リンクドリストがない場合に、有効トラック数の少ない論理ブロックを抽出するには、例えばMS論理ブロック管理テーブル35を論理ブロック0から順番にサーチする作業が必要になる。しかしながら、この方法では、MS論理ブロック管理テーブル35内を最後までサーチしないと、最適解が得られない可能性がある。

20

【0225】

また、有効トラック数毎に論理ブロックを抽出する方法として、図28に示した有効トラック数別論理ブロックIDリスト62を用いる方法がある。この方法では、論理NAND層管理部120bが、MS11内でのコンパクションを行なう前に、MS11中の論理ブロック(論理ブロックID)を有効トラック数によってソートしておかなければならない。

【0226】

一方、本実施の形態では、リンクドリストを用いて有効トラック数の少ない論理ブロックから順番にMS論理ブロック管理テーブル35内をサーチできるので、コンパクションを効率良く行なうのに必要な、有効トラック数の少ない論理ブロックの集合を容易に見つけることが可能となる。また、コンパクションを行う際に、有効トラック数別論理ブロックIDリスト62が不要となる。このため、狭い領域で効率良くコンパクションに関する管理を行うことが可能となる。

30

【0227】

図34は、有効トラック数ごとのリンクドリストを説明するための図である。同図では、有効トラック数1のリスト先頭ポインタP1がブロック0を示し、有効トラック数2のリスト先頭ポインタP2がブロック3を示し、有効トラック数 2^i のリスト先頭ポインタPiがブロック1を示している。

【0228】

有効トラック数1の論理ブロックは、ブロック0、ブロック7の順番でリンクドリストを形成(連結)している。また、有効トラック数2の論理ブロックは、ブロック3、ブロック9、ブロック4、ブロック8の順番でリンクドリストを形成している。また、有効トラック数 2^i の論理ブロックは、ブロック1、ブロック2、ブロック5、ブロック6の順番でリンクドリストを形成している。

40

【0229】

したがって、論理NAND層管理部120bが有効トラック数1の論理ブロックを検索すると、ブロック0、ブロック7の順番で論理ブロックを検索でき、有効トラック数2の論理ブロックを検索すると、ブロック3、ブロック9、ブロック4、ブロック8の順番で論理ブロックを検索できる。また、論理NAND層管理部120bが有効トラック数 2^i の論理ブロックを検索すると、ブロック1、ブロック2、ブロック5、ブロック6の順番

50

で論理ブロックを検索できる。

【0230】

ところで、有効トラック数が少ない論理ブロックを集めてコンパクションを行なうと、無駄なデータの移動が少なくなるので、効率良くコンパクションを行なえる。したがって、本実施の形態では、有効トラック数が少ないものが優先される順番で論理ブロックを抽出し、抽出した順番で論理ブロックのコンパクションを行なう。

【0231】

具体的には、論理NAND層管理部120bは、MS論理ブロック管理テーブル35内から、リンクドリストを用いて有効トラック数1の論理ブロックを順番に抽出し、有効トラック数1の論理ブロックが 2^i (1論理ブロック分) 集まる場合には、MSコンパクションバッファ11cを介した 2^i トラックMSコンパクションを実行する。そして、論理NAND層管理部120bは、MS論理ブロック管理テーブル35内から有効トラック数1の論理ブロックが 2^i 個集まる度に、 2^i トラックMSコンパクションを実行する処理を繰り返す。有効トラック数が1個の論理ブロックを 2^i 個集めることが不可能な状態になると、論理NAND層管理部120bは、論理ブロックから有効トラックを($2^i - 1$)よりも少ない個数だけ集める、MS追記IB11abを介した 2^i トラック未満コンパクションを実行する。

【0232】

なお、ここでは有効トラック数 2^i (最大数) の論理ブロックについてもリンクドリストを形成する場合について説明したが、有効トラック数 2^i の論理ブロックはコンパクション対象から除外してもよい。この場合、有効トラック数 $1 \sim (2^i - 1)$ 個の各論理ブロックについてリンクドリストを形成すればよい。

【0233】

なお、prevポインタ35g、nextポインタ35h、リスト先頭ポインタは、以下の場合に更新される。

- ・新規にTFS11bに論理ブロックが登録されたとき
- ・新規にMS11に論理ブロックが登録されたとき
- ・論理ブロックが持っているトラックと同じアドレスのトラックが他の論理ブロックに再書込みされて、元ブロック側がInvalidateされた場合 (有効トラック数が減った場合)

別言すれば、論理ブロックがMS11に利用され始めてから開放されるまで、有効トラック数が変化するたびに、prevポインタ35g、nextポインタ35h、リスト先頭ポインタは、該当するリスト (エントリ) に繋ぎ替えられる。

【0234】

また、コンパクションが完了した論理ブロックは、MS論理ブロック管理テーブル35内のエントリから削除され、コンパクションによって発生した新たな論理ブロックは、MS論理ブロック管理テーブル35に登録される。

【0235】

なお、本実施の形態では、MS論理ブロック管理テーブル35内における各有効トラック数の先頭のエントリ (論理ブロックID) をリスト先頭ポインタによって指定する場合について説明したが、リスト先頭ポインタは無くてもよい。この場合は、検索対象となっている有効トラック数を有したエントリの何れか1つをMS論理ブロック管理テーブル35から検索し、その後、検索したエントリのprevポインタ35gやnextポインタ35hを用いて検索対象の有効トラック数を有した論理ブロックIDを順番に検索する。

【0236】

また、本実施の形態では、prevポインタ35gとnextポインタ35hとによって、有効トラック数毎にリンクドリスト構造を形成したが、nextポインタ35hのみによって、有効トラック数毎にリンクドリスト構造を形成してもよい。

【0237】

また、本実施の形態では、MS論理ブロック管理テーブル35にリンクドリストを適用して有効トラック数毎に論理ブロックを管理したが、FS/IS論理ブロック管理テーブ

10

20

30

40

50

ル42にリンクドリストを適用して有効クラスタ数毎に論理ブロックやFS/ISブロックを管理してもよい。

【0238】

このように本実施の形態では、MS論理ブロック管理テーブル35内で論理ブロックを有効トラック数毎にリンクドリストで管理しているため、有効トラック数毎の論理ブロックをMS11内から容易かつ迅速に検索できる。また、有効トラック数の少ない論理ブロックから順番にコンパクションを行なうことができるため、無駄なデータ移動を増やすことなく、効率良くコンパクションを行なうことが可能となる。

【0239】

また、MS論理ブロック管理テーブル35内では、各有効トラック数の先頭のエントリをリスト先頭ポインタによって指定しているため、有効トラック数毎の論理ブロックの検索が容易になる。

10

【0240】

また、prevポインタ35gとnextポインタ35hとによる双方向リンクドリストによって、有効トラック数毎に論理ブロックを管理しているため、ある論理ブロックの有効トラック数が減少した際には、その論理ブロックを別のリンクドリストに容易に繋ぎ代えることが可能となる。

【図面の簡単な説明】

【0241】

【図1】SSDの構成例を示すブロック図。

20

【図2】NANDメモリチップに含まれる1個のブロックの構成例と、4値データ記憶方式でのしきい値分布を示す図。

【図3】ドライブ制御回路のハードウェア的な内部構成例を示すブロック図。

【図4】プロセッサの機能構成例を示すブロック図。

【図5】NANDメモリおよびDRAM内に形成された機能構成を示すブロック図。

【図6】WCからNANDメモリへの書き込み処理に係わるより詳細な機能ブロック図。

【図7】LBA論理アドレスを示す図。

【図8】データ管理部内の管理テーブルの構成例を示す図。

【図9】RCクラスタ管理テーブルの一例を示す図。

【図10】WCクラスタ管理テーブルの一例を示す図。

30

【図11】WCトラック管理テーブルの一例を示す図。

【図12】トラック管理テーブルの一例を示す図。

【図13】FS/IS管理テーブルの一例を示す図。

【図14】MS論理ブロック管理テーブルの一例を示す図。

【図15】FS/IS論理ブロック管理テーブルの一例を示す図。

【図16】FS/IS内クラスタ管理テーブルの一例を示す図。

【図17】論物変換テーブルの一例を示す図。

【図18】読み出し処理の動作例を示すフローチャート。

【図19】書き込み処理の動作例を示すフローチャート。

【図20】各構成要素間のデータの流れにおける入力と出力の組み合わせおよびその発生要因を示す図。

40

【図21】WCからNANDメモリへの書き込み処理に係わるさらにより詳細な機能ブロック図。

【図22】データ管理部内の管理テーブルの他の構成例を示す図。

【図23】並列動作要素とプレーンとチャンネルの関係を示す図。

【図24】論物変換テーブルの他の例を示す図。

【図25】BB管理テーブルの一例を示す図。

【図26】FB管理テーブルの内部構成例を示す図。

【図27】NANDメモリの論理ブロックと物理ブロックとの対応関係を示す図。

【図28】MS構造管理テーブルの一例を示す図。

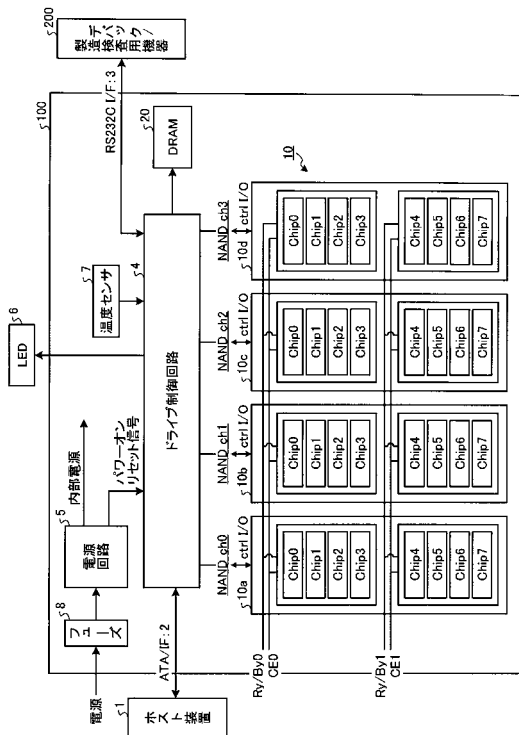
50

- 【図29】FS / IS 構造管理テーブルの一例を示す図。
- 【図30】書き込み処理の動作例を示す詳細なフローチャート。
- 【図31】IS の追い出し動作例を示すフローチャート。
- 【図32】有効トラック数ごとのリンクリストを有したMS 論理ブロック管理テーブルの一例を示す図。
- 【図33】リンクリストの構造を説明するための図。
- 【図34】有効トラック数ごとのリンクリストを説明するための図。
- 【符号の説明】
- 【0242】

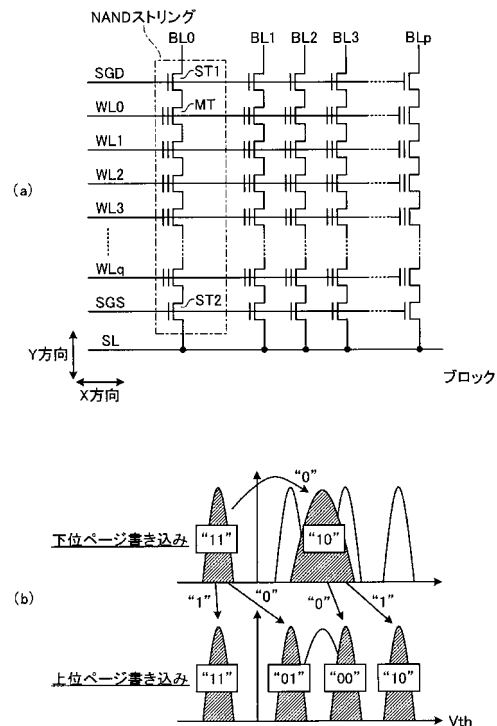
10 NANDメモリ、11、メインストレージ(MS)、11a メインストレージ入力バッファ(MSIB)、11aa フルブロック入力バッファ(MSFB)、11ab MS追記入力バッファ(MS追記IB)、11b トラック前段バッファ(TFS)、12 前段ストレージ(FS)、12a 前段ストレージ入力バッファ(FSIB)、13 中段ストレージ(IS)、13a 中段ストレージ入力バッファ(ISIB)、13aa 中段ストレージ本体(IS本体)、20 DRAM、21 ライトキャッシュ(WC)、22、リードキャッシュ(RC)、120 データ管理部、120a DRAM層管理部、120b 論理NAND層管理部、120c 物理NAND層管理部。

10

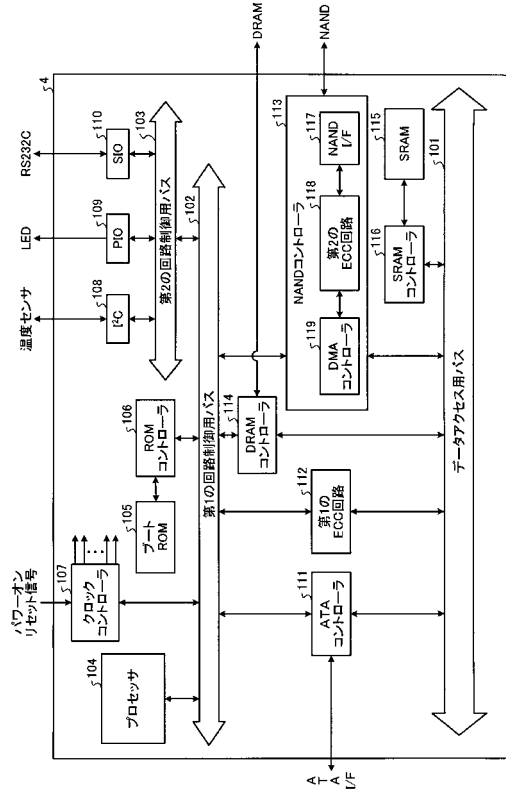
【図1】



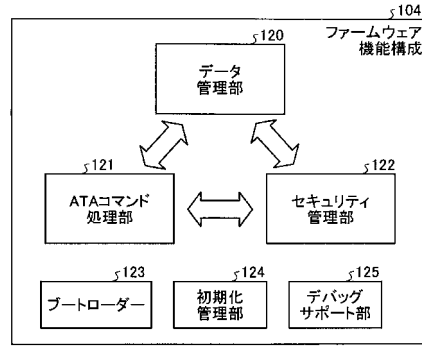
【図2】



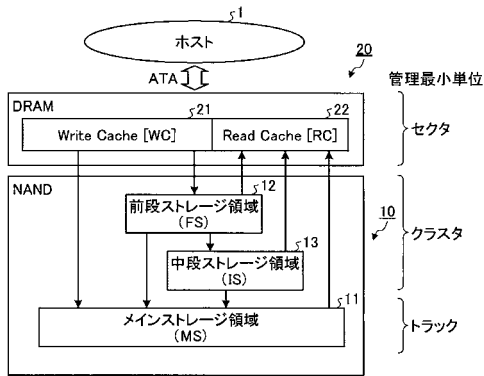
【図3】



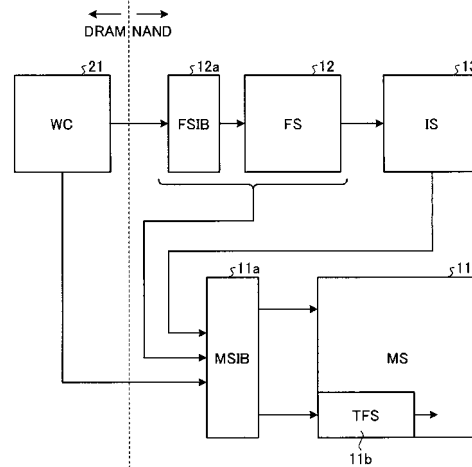
【図4】



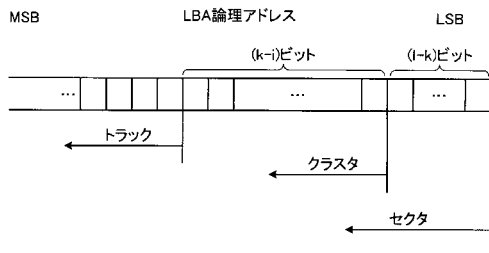
【図5】



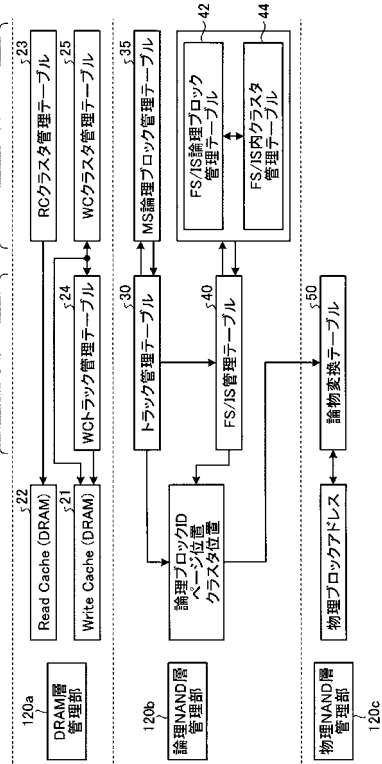
【図6】



【図7】



【図8】



【図9】

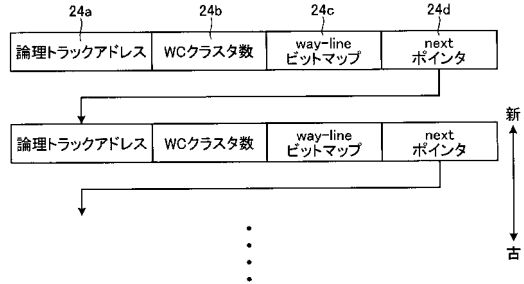
RCクラスタ管理テーブル23(逆引き): n-way セットアソシアティブ

	way 1	way n
Line 1	状態フラグ+論理トラックアドレス	...
⋮	⋮	⋮
Line m	状態フラグ+論理トラックアドレス	...

23a 23b

【図11】

WCトラック管理テーブル24(正引き)



【図10】

WCクラスタ管理テーブル25(逆引き): n-way セットアソシアティブ

	way 1	way n
Line 1	状態フラグ+セクタ位置ビットマップ+論理トラックアドレス	...
Line 2	25a 25c ... 25b	...
⋮	⋮	⋮
Line m

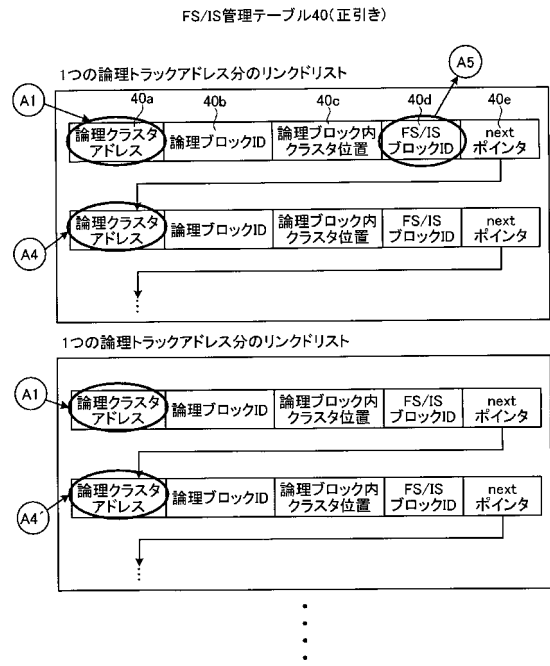
【図12】

トラック管理テーブル30(正引き)

30a	30b	30c	30d	30e	30f	30g
論理トラックアドレス	クラスヒットマップ	論理ブロックID+論理ブロック内トラック位置	クラステーブルポインタ	FSクラス数	ISクラス数	
トラック0
トラック3
...
トラックn

(A2) points to 30a, (A1) points to 30d.

【図13】



【図14】

MS論理ブロック管理テーブル35(逆引き)

35a	35b	35c	35d	35e
論理ブロックID	トラック管理ポインタ(2トラック分)	有効トラック数	書き込み可能先頭トラック	Valid
ブロック0
ブロック3
...
ブロックm

(A2) points to 35b.

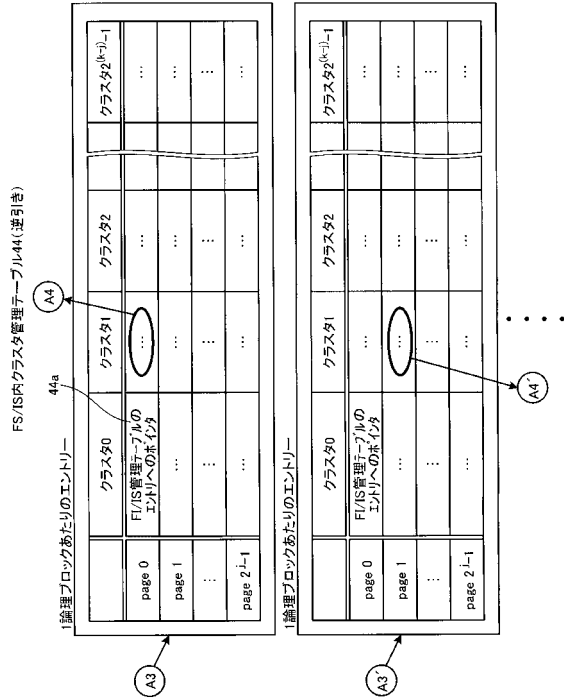
【図15】

FS/IS論理ブロック管理テーブル42(逆引き)

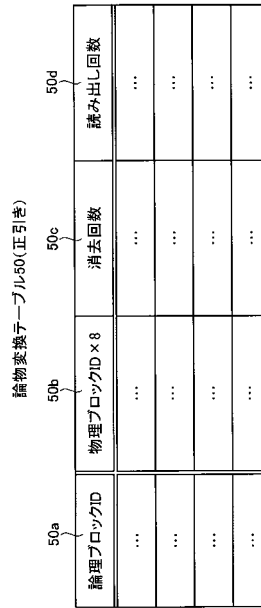
42a	42b	42c	42d	42e	42f
FS/ISブロックID	論理ブロックID	ブロック内クラステーブル	有効クラス数	書き込み可能先頭ページ	Valid
FS/ISブロック0
FS/ISブロック4
...
FS/ISブロックk

(A5) points to 42a, (A3) points to 42c, (A3') points to 42c.

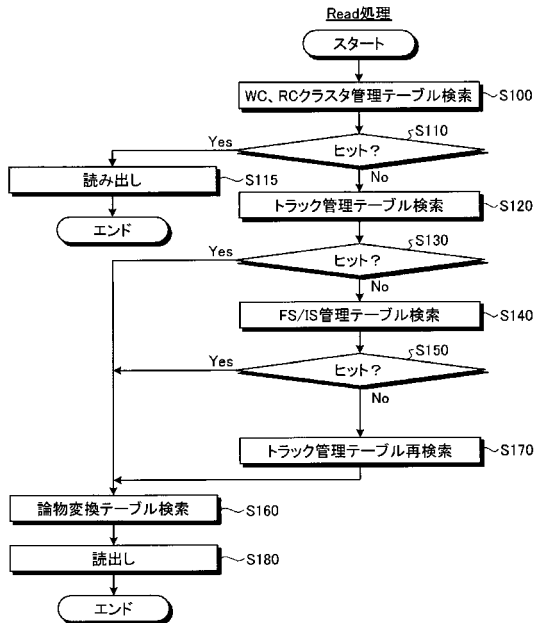
【図16】



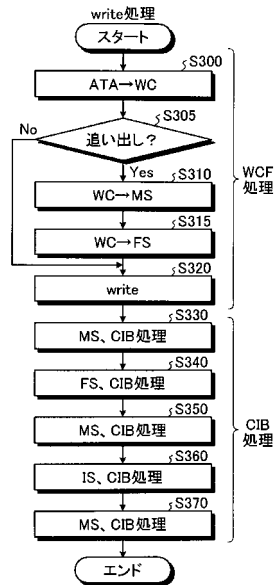
【図17】



【図18】



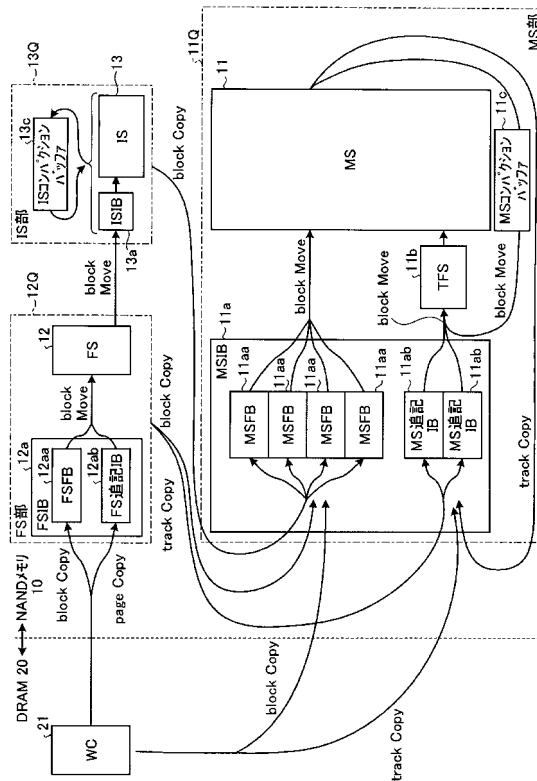
【図19】



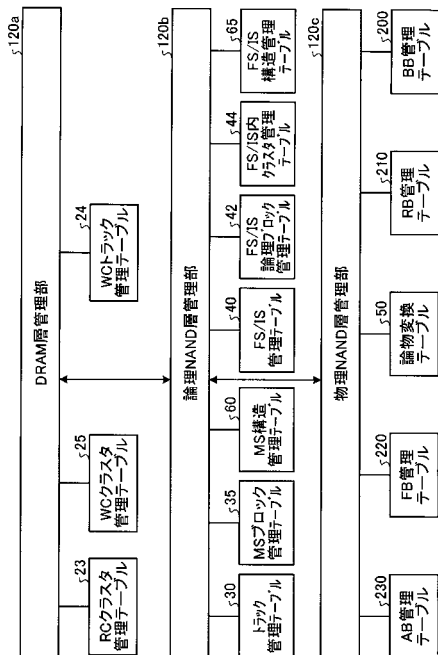
【図20】

From \ To	WC	FS	IS	MS
WC	N/A	クラスタ追い出し	N/A	トラック追い出し 受動マージ
FS	N/A	クラスタ穴埋め (ブロック追い出し)		トラック追い出し 受動マージ
IS	N/A	クラスタ穴埋め	IS compaction	トラック追い出し 受動マージ
MS	N/A	クラスタ穴埋め	N/A	トラック穴埋め MS compaction

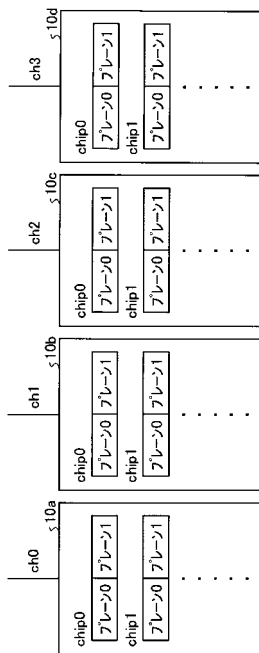
【図21】



【図22】



【図23】



【図24】

論物変換テーブル50(正引き)

50a	50b	50c	50d	50e
論理ブロックID	物理ブロックID×8	消去回数	読み出し回数	消去時刻
...
...
...
...

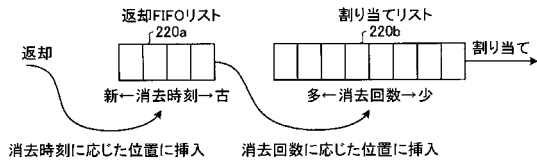
【図25】

BB管理テーブル200

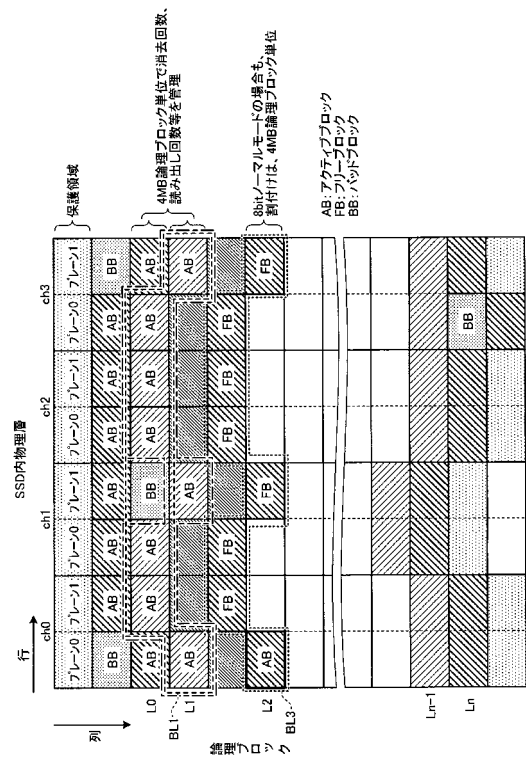
channel 0		channel 1		channel 2		channel 3	
plane 0	plane 1	plane 0	plane 1	plane 0	plane 1	plane 0	plane 1
ID#0	ID#1	ID#2	ID#3	ID#4	ID#5	ID#6	ID#7
物理
アドレスID
...
...
...

200a

【図26】



【図27】

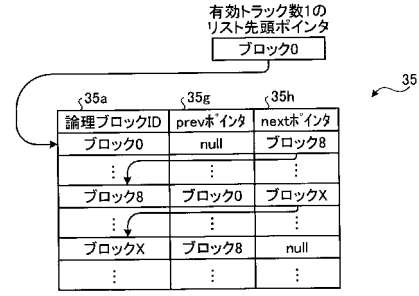


【図 3 2】

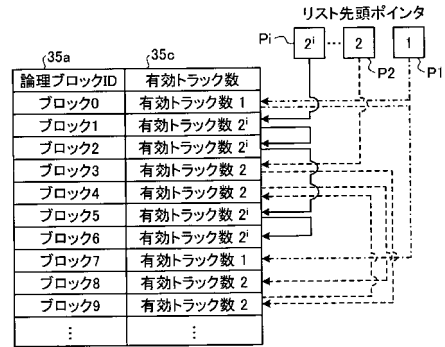
MS論理ブロック管理テーブル35(逆引き)

35a	35b	35c	35d	35e	35g	35h
論理ブロックID	トラック管理 ポイント (2トラック分)	有効トラック数	書き込み可能 先頭トラック	Valid	prev ポインタ	next ポインタ
ブロック0
ブロック3
...
ブロックm

【図 3 3】



【図 3 4】



フロントページの続き

(72)発明者 岡本 渉
東京都港区芝浦一丁目1番1号 株式会社東芝内

審査官 堀江 義隆

(56)参考文献 特開2000-132396(JP,A)
特開2006-48237(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 12/02
G06F 12/00
G06F 3/08