

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5221593号
(P5221593)

(45) 発行日 平成25年6月26日(2013.6.26)

(24) 登録日 平成25年3月15日(2013.3.15)

(51) Int.Cl.

F I

G O 6 F 12/02 (2006.01)

G O 6 F 12/02 5 3 0 C

G O 6 F 12/00 (2006.01)

G O 6 F 12/00 5 9 7 U

請求項の数 20 (全 52 頁)

(21) 出願番号 特願2010-102697 (P2010-102697)
 (22) 出願日 平成22年4月27日(2010.4.27)
 (62) 分割の表示 特願2008-63404 (P2008-63404)
 の分割
 原出願日 平成20年3月12日(2008.3.12)
 (65) 公開番号 特開2010-176702 (P2010-176702A)
 (43) 公開日 平成22年8月12日(2010.8.12)
 審査請求日 平成23年3月14日(2011.3.14)

(73) 特許権者 000003078
 株式会社東芝
 東京都港区芝浦一丁目1番1号
 (74) 代理人 100089118
 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備え、各々の前記ブロックはデータ書き込み及び読み出し単位であるページを複数含む不揮発性半導体メモリと、

前記不揮発性半導体メモリが備える複数のブロックを、セクタ単位の自然数倍である第1の管理単位でデータが管理される第1の記憶領域と、前記第1の管理単位の2以上の自然数倍である第2の管理単位でデータが管理される第2の記憶領域とに割り当てるコントローラと、を具備し、

前記コントローラは、

前記第1の管理単位の有効データを前記第1の記憶領域から追い出し、当該有効データを前記第2の管理単位のデータに統合して、前記第2の記憶領域に記憶する第1の手段と、

前記第1の記憶領域内における前記第1の管理単位の有効データを複数選択して、前記第1の記憶領域内の新たなブロックに書き直す第2の手段と、を備え、

前記第2の手段を実行する前に、少なくとも1回、前記第1の手段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、前記第1の記憶領域に割り当てられたブロックの個数が上限値を超えた場合に、前記第1の手段を実行した後、前記第2の手段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。

10

20

【請求項 3】

前記コントローラは、前記第 1 の手段を実行した後、前記第 1 の記憶領域内における前記第 1 の管理単位の有効データの合計容量が所定値以上である場合、再度前記第 1 の手段を実行することを特徴とする請求項 2 に記載のメモリシステム。

【請求項 4】

前記コントローラは、前記第 1 の手段を実行した後、前記第 1 の記憶領域内における前記第 1 の管理単位の有効データの合計容量が所定値未満である場合、前記第 2 の手段を実行することを特徴とする請求項 2 に記載のメモリシステム。

【請求項 5】

前記所定値は、1 ブロック分の記憶容量に等しいことを特徴とする請求項 3 または請求項 4 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、前記第 1 の手段を実行する際に、複数ブロックの記憶容量に対応する前記第 2 の管理単位のアドレス範囲を選択し、選択された前記第 2 の管理単位のアドレス範囲内に含まれる前記第 1 の管理単位の有効データを追い出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】

前記コントローラは、前記第 2 の手段を実行する際に、前記第 1 の管理単位の有効データの個数が少ない方から順にブロック選択し、選択されたブロックに含まれる前記第 1 の管理単位の有効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。

20

【請求項 8】

前記コントローラは、前記第 1 の手段を実行する際に、前記第 2 の管理単位のアドレス範囲内に含まれる前記第 1 の管理単位の有効データの個数に基づいて、前記第 1 の記憶領域から前記第 2 の記憶領域に追い出す有効データを選択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 9】

前記コントローラは、前記第 1 の手段を実行する際に、前記有効データの個数が多い方から順に前記第 2 の管理単位のアドレス範囲を選択し、選択された前記第 2 の管理単位のアドレス範囲内に含まれる前記第 1 の管理単位の有効データを追い出すことを特徴とする請求項 8 に記載のメモリシステム。

30

【請求項 10】

前記コントローラは、前記第 1 の手段を実行する際に、前記第 1 の記憶領域内において前記第 1 の管理単位の有効データを含む前記第 2 の管理単位のアドレス範囲に対応する前記第 2 の管理単位の有効データが、前記第 2 の記憶領域内において前記第 2 の管理単位の無効データを含むブロック内に記憶されているか否かに基づいて、前記第 1 の記憶領域から前記第 2 の記憶領域に追い出すデータを選択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 11】

前記コントローラは、前記第 1 の手段を実行する際に、前記無効データの個数が多いブロックに記憶されている前記第 2 の管理単位の有効データに対応する前記第 2 の管理単位のアドレス範囲を優先的に選択し、選択された前記第 2 の管理単位のアドレス範囲内に含まれる前記第 1 の管理単位の有効データを追い出すことを特徴とする請求項 10 に記載のメモリシステム。

40

【請求項 12】

前記コントローラは、前記第 1 の手段を実行する際に、前記第 2 の記憶領域に追い出された前記第 1 の管理単位の有効データを前記第 1 の記憶領域内において無効化し、有効データが存在しないブロックを前記第 1 の記憶領域から解放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 13】

前記コントローラは、前記第 2 の手段を実行する際に、第 1 の記憶領域内の新たなプロ

50

ックに書き直された前記第 1 の管理単位の有効データを前記第 1 の記憶領域内において無効化し、有効データが存在しないブロックを前記第 1 の記憶領域から解放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 1 4】

前記コントローラは、前記第 1 の手段を実行する際に、前記第 1 の記憶領域から追い出される前記第 1 の管理単位の有効データと、前記第 1 の記憶領域及び前記第 2 の記憶領域の少なくとも一方から読み出されたデータとを、前記第 2 の管理単位のデータに統合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 1 5】

前記不揮発性半導体メモリに転送するデータを保持可能な半導体メモリを更に具備し、前記コントローラは、

前記半導体メモリに前記セクタ単位のデータを記憶する第 3 の手段と、

前記半導体メモリから追い出されたデータを、前記第 1 の管理単位のデータに統合して前記第 1 の記憶領域に記憶する第 4 の手段と、

前記半導体メモリから追い出されたデータを、前記第 2 の管理単位のデータに統合して前記第 2 の記憶領域に記憶する第 5 の手段と、

を備え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 1 6】

前記半導体メモリは、揮発性ランダムアクセスメモリであることを特徴とする請求項 1 5 に記載のメモリシステム。

【請求項 1 7】

前記半導体メモリは、不揮発性ランダムアクセスメモリであることを特徴とする請求項 1 5 に記載のメモリシステム。

【請求項 1 8】

前記ページ単位は、前記第 1 の管理単位の 2 以上の自然数倍であり、かつ、前記第 1 の管理単位は、前記ブロック単位よりも小さい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 1 9】

前記ブロック単位は、前記第 2 の管理単位の 2 以上の自然数倍であり、かつ、前記第 2 の管理単位は、前記ページ単位よりも大きい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 2 0】

前記不揮発性半導体メモリは、NAND型フラッシュメモリ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。

【発明の詳細な説明】

【技術分野】

【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

10

20

30

40

50

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。

【 0 0 0 5 】

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。

【 0 0 0 6 】

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。

10

【 0 0 0 7 】

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。

【 0 0 0 8 】

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。

【 0 0 0 9 】

20

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。

【 0 0 1 0 】

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。

30

【 0 0 1 1 】

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。

【 0 0 1 2 】

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ので、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を短くするための手法が求められている。

40

【 先行技術文献 】

【 特許文献 】

【 0 0 1 3 】

【 特許文献 1 】 特許第 3 6 8 8 8 3 5 号公報

50

【非特許文献】

【0014】

【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

【発明の概要】

【発明が解決しようとする課題】

【0015】

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。

【課題を解決するための手段】

10

【0016】

本願発明の一態様によれば、揮発性半導体メモリから構成されるキャッシュメモリとしての第1の記憶エリアと、ページ単位で読み出し及び書き込みが行われ、前記ページ単位の2以上の自然数倍の大きさであるブロック単位で消去が行われる不揮発性半導体メモリから構成される第2、第3及び第4の記憶エリアと、前記不揮発性半導体メモリの記憶領域を、1以上の前記ブロックに対応付けられた論理ブロック単位で前記第2、第3および第4の記憶エリアに割り当てるコントローラと、を具備し、前記コントローラは、前記第1の記憶エリアに書き込まれたセクタ単位の複数のデータを第1の管理単位のデータとして前記第2の記憶エリアに追い出す第1の処理と、前記第1の記憶エリアに書き込まれた複数のデータを前記第1の管理単位の2以上の自然数倍の大きさである第2の管理単位のデータとして第3の記憶エリアに追い出す第2の処理と、前記第2の記憶エリアに記憶された複数のデータを前記第2の管理単位のデータとして前記第3の記憶エリアに追い出す第3の処理と、前記第3の処理の実行後、前記第2の記憶エリアに記憶された複数のデータを論理ブロック単位で前記第4の記憶エリアに移動する第4の処理と、前記第4の記憶エリアに記憶された前記第1の管理単位の有効データを複数選択して新たな論理ブロック内に書き直す第5の処理と、第5の処理を実行する前に、前記第4の記憶エリアに書き込まれた複数のデータを前記論理ブロック単位に前記第3の記憶エリアに追い出す第6の処理とを実行することを特徴とする。

20

【発明の効果】

【0017】

30

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。

【図面の簡単な説明】

【0018】

【図1】SSDの構成例を示すブロック図。

【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。

40

【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。

【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。

50

【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。

【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ネルの関係を示す図。

10

【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。

【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。

【発明を実施するための形態】

【0019】

20

以下に添付図面を参照して、この発明にかかるメモリシステムの最良な実施の形態を詳細に説明する。

【0020】

(実施の形態)

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。

【0021】

まず、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 SSD 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 10 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。

30

・論理ページ：SSD 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、32 ビット倍速モードでは、32 k B。ただし、冗長ビットは含まないものとする。

・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 512 k B。ただし、主データに対して SSD 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、512 k B + 冗長ビット（例えば、数 10 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。

40

・論理ブロック：SSD 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、512 k B、32 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 512 B。

・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。

50

・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。

10

・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。

【第1の実施形態】

【0022】

図1は、SSD（Solid State Drive）100の構成例を示すブロック図である。SSD100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、SSD100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3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。SSD100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DRAM20と、電源回路5と、状態表示用のLED6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。

20

【0023】

30

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SSD100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SSD100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。

【0024】

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。

40

【0025】

DRAM20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よび作業領域用メモリとして機能する。また、DRAM20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DRAM20を介してデータ転送制御を行うとともに、SSD100内の各構成要素を制御す

50

る。また、ドライブ制御回路4は、状態表示用LED6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SSD100内の各部に供給する機能も有している。

【0026】

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

【0027】

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

【0028】

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。

【0029】

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。

30

【0030】

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。

【0031】

40

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。

【0032】

また、ビット線BL0~BLpは、ブロック間で、選択トランジスタST1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NANDストリングは、同一のビット線BLに接続される。

50

【 0 0 3 3 】

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 4 】

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。

【 0 0 3 5 】

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。

【 0 0 3 6 】

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、NAND 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 ROM 1 0 5 が ROM 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。

【 0 0 3 7 】

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 I²C 回路 1 0 8、状態表示用 LED 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 8 】

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、NAND 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。NAND 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 ROM 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。

【 0 0 3 9 】

NAND コントローラ 1 1 3 は、NAND メモリ 1 0 とのインタフェース処理を行う NAND I / F 1 1 7、第 2 の E C C 回路 1 1 8、および NAND メモリ 1 0 - D R A M 2 0 間のアクセス制御を行う D M A 転送制御用 D M A コントローラ 1 1 9 を備えている。第 2 の E C C 回路 1 1 8 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 E C C 回路 1 1 2 は、第 2 の誤り訂正符号

10

20

30

40

50

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 (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。

【0040】

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。

10

(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 $\times 4$ (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 $\times 4$ (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。

20

(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 $\times 4 \times 2$ (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 $\times 4 \times 2$ (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。

【0041】

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 = 2^i トラック = 2^j ページ = 2^k クラスタ = 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)。

30

【0042】

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 ($2 \times 4ch$) 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パッドブロック BB が発生すると、そのパッドブロック 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、パッドブロック BB を含まないように変更される。

40

【0043】

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。

【0044】

データ管理部 120 は、NAND コントローラ 112、第 1 の ECC 回路 114 を介して、NAND メモリ 10 - DRAM 20 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 DRAM 20 - ホ

50

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 1 2 2 は、データ管理部 1 2 0 および A T A コマンド処理部 1 2 1 と協働して各種のセキュリティ情報を管理する。

【 0 0 4 5 】

ブートルoader 1 2 3 は、パワーオン時、各管理プログラム（ファームウェア）を N A N D メモリ 1 0 から S R A M 1 2 0 にロードする。初期化管理部 1 2 4 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 1 2 5 は、外部から R S 2 3 2 C 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 1 2 0、A T A コマンド処理部 1 2 1、およびセキュリティ管理部 1 2 2 が、S R A M 1 1 4 に記憶される各管理プログラムをプロセッサ 1 0 4 が実行することによって実現される機能部である。

10

【 0 0 4 6 】

本実施形態では、主としてデータ管理部 1 2 0 が実現する機能について説明する。データ管理部 1 2 0 は、A T A コマンド処理部 1 2 1 が記憶デバイスである N A N D メモリ 1 0 や D R A M 2 0 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 N A N D メモリ 1 0 との対応関係の管理および管理情報の保護と、D R A M 1 0 および N A N D メモリ 1 0 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、N A N D メモリ 1 0 の信頼性の確保などを行う。

【 0 0 4 7 】

図 5 は、N A N D メモリ 1 0 および D R A M 2 0 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 N A N D メモリ 1 0 との間には、D R A M 2 0 上に構成されたライトキャッシュ（W C）2 1 およびリードキャッシュ（R C）2 2 が介在している。W C 2 1 はホスト装置 1 からの Write データを一時保存し、R C 2 2 は N A N D メモリ 1 0 からの Read データを一時保存する。N A N D メモリ 1 0 内の論理ブロックは、書き込み時の N A N D メモリ 1 0 に対する消去の量を減らすために、データ管理部 1 2 0 により、前段ストレージ領域（F S : Front Storage）1 2、中段ストレージ領域（I S : Intermediate Storage）1 3 およびメインストレージ領域（M S : Main Storage）1 1 という各管理領域に割り当てられている。F S 1 2 は、W C 2 1 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。I S 1 3 は、F S 1 2 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

20

30

【 0 0 4 8 】

小さな管理単位を、N A N D メモリ 1 0 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、D R A M 2 0 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、N A N D メモリ 1 0 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 N A N D メモリ 1 0 の各ストレージを構成している。

【 0 0 4 9 】

図 6 は、W C 2 1 から N A N D メモリ 1 0 への書き込み処理（W R 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。F S 1 2 の前段には、W C 2 1 からのデータをバッファリングする F S インプットバッファ（F S I B）1 2 a が設けられている。また、M S 1 1 の前段には、W C 2 1、F S 1 2、または I S 1 3 からのデータをバッファリングする M S インプットバッファ（M S I B）1 1 a が設けられている。また、M S 1 1 には、トラック前段ストレージ領域（T F S）1 1 b が設けられている。T F S 1 1 b は、M S I B 1 1 a と M S 1 1 の間に介在する F I F O（First in First out）構造を有するバッファであり、T F S 1 1 b に記録されたデータは、M S I B 1 1 a から直接 M S 1 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。M S 1 1、M S 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。

40

50

【 0 0 5 0 】

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 = $2^{(1 - k)}$ セクタで、1 トラック = $2^{(k - i)}$ クラスタである。

10

【 0 0 5 1 】

・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 $2^{(k - i)}$ 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。

【 0 0 5 2 】

・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 $2^{(k - i)}$ 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。

20

30

【 0 0 5 3 】

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。

(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。

40

【 0 0 5 4 】

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 5 】

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、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、W C 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 5 0 % 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

50

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。

【0056】

また、(i) の条件でトラック追い出しが発生した場合で、FS12 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 WC21 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 FSIB12a 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、FSIB12a に論理ページ単位で追い出しする。なお、FS12 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。

【0057】

また、ATA コマンド処理部 121 からの Cache Flush 要求では、WC21 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 MS11 へ、超えていなければ FS12 へ）で、FS12 もしくは MS11 に対して追い出される。

【0058】

・前段ストレージ領域 (FS) 12

つぎに、FS12 について説明する。FS12 はクラスタ単位でデータを管理される FIFO である。FS12 は、ここを通過しているデータは、後段の IS13 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、FS12 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書き込みがあった場合無効化されるので、FS12 を通過中のクラスタは、FS12 から IS13 や MS11 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。

【0059】

FS12 を設けることで、後段の IS13 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、FS12 に割り当てる。

【0060】

WC21 から FS12 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 FSIB12a に割り当てられた論理ブロックに対して書き込まれる。FSIB12a 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 FSIB12a から FS12 に Move される。この FSIB12a から FS12 への Move の際に、FS12 のブロック数が FS12 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 FS12 から IS13 または MS11 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、MS11 (TFS11b) への書き込みを行い、有効クラスタが残ったブロックを、IS13 へ Move する。

【0061】

NAND メモリ 10 内の構成要素間のデータ移動には、Move と Copy の二通りがある。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

10

20

30

40

50

方法である。

【0062】

・中段ストレージ領域 (IS) 13

つぎに、IS 13 について説明する。IS 13 は、FS 13 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、IS 13 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。FS 12 から IS 13 に対して論理ブロックの移動 (Move)、すなわち FS 12 からの追い出しが発生すると、以前 FS 12 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 13 の管理対象ブロックとなる。この FS 12 から IS 13 への論理ブロックの移動により、IS 13 のブロック数が IS 13 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、IS 13 から MS 11 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、IS 13 のブロック数は規定値に戻される。

10

【0063】

IS 13 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 \times 有効クラスタ係数 (トラックが MS 11 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 MS IB 11a に追い出す。

20

・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、IS 13 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。

30

【0064】

・メインストレージ領域 (MS) 11

つぎに、MS 11 について説明する。MS 11 はトラック単位でデータの管理を行う。MS 11 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。WC 21、FS 12、IS 13 から MS 11 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 MS IB 11a 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 MS IB 11a に書き込む、後述する受動マージが行われる。MS 11 内に無効トラックが蓄積し、MS 11 に割り当てられている論理ブロックの個数が MS 11 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。

40

【0065】

MS 11 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 21、FS 12、IS 13 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、TF S 11 b に出力し (2^i トラック MS

50

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、MSIB11aに出力して(2^i トラック未満コンパクション)、より多くの無効なフリーブロックFBを作る。

【0066】

TFS11bは、トラック単位でデータを管理されるFIFOである。TFS11bは、ここを通過しているデータは、後段のMS11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、TFS11b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、TFS11bを通過中のトラックは、TFS11bからMS11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。

【0067】

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。

【0068】

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。

【0069】

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。

【0070】

・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 \times m-line \times 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(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、DRAM2

10

20

30

40

50

0 上のタグ記憶位置から L B A 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。

【 0 0 7 1 】

・ W C クラスタ管理テーブル 2 5 (逆引き)

つぎに、図 1 0 を用いて W C クラスタ管理テーブル 2 5 について説明する。W C 2 1 は、前述したように、論理クラスタアドレス L S B ($k - i$) 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。W C クラスタ管理テーブル 2 5 は、W C (クラスタサイズ \times m -line \times n -way) 2 1 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 2 5 a と、セクタ位置ビットマップ 2 5 b と、論理トラックアドレス 2 5 c によって構成されている。

10

【 0 0 7 2 】

状態フラグ 2 5 a 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 N A N D メモリ 1 0 への追い出し待ちか否かを示すビット、当該エントリが A T A 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 2 5 b は、1 クラスタに含まれる $2^{(1-k)}$ セクタのうちの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 2 5 b によって、W C 2 1 において、L B A と同じセクタ単位の管理を行うことができる。W C クラスタ管理テーブル 2 5 は、D R A M 2 0 上のタグ記憶位置から L B A 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

【 0 0 7 3 】

・ W C トラック管理テーブル 2 4 (正引き)

つぎに、図 1 1 を用いて W C トラック管理テーブル 2 4 について説明する。W C トラック管理テーブル 2 4 は、W C 2 1 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、F I F O 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 W C 2 1 に登録された順序 (L R U) を表現している。なお、W C 2 1 で最後に更新された順序によって L R U 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 2 4 a、当該論理トラックアドレスに含まれる W C 2 1 中の有効クラスタ数 2 4 b、way-line ビットマップ 2 4 c および次のエントリへのポインタを示す next ポインタ 2 4 d から構成されている。W C トラック管理テーブル 2 4 は、論理トラックアドレス 2 4 a 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

【 0 0 7 4 】

way-line ビットマップ 2 4 c は、W C 2 1 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 W C 2 1 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 2 4 c 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。W C トラック管理テーブル 2 4 はリンクドリスト構造を有しており、W C 2 1 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

【 0 0 7 5 】

・ トラック管理テーブル 3 0 (正引き)

つぎに、図 1 2 を用いてトラック管理テーブル 3 0 について説明する。トラック管理テーブル 3 0 は、論理トラックアドレス単位で M S 1 1 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 F S 1 2 や I S 1 3 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 3 0 a 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 3 0 a 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 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、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

50

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ID（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。

【0076】

クラスタビットマップ30b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MS11に存在するか、FS12もしくはIS13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MS11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FS12もしくはIS13に存在する可能性があることを示している。

10

【0077】

論理ブロックID30c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ID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30dは、論理ブロックID30c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(30a)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30dは、iビットで 2^i 個のトラック位置を識別する。

【0078】

クラスタテーブルポインタ30eは、リンクドリスト構造を有するFS/IS管理テーブル40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30bの検索で、当該クラスタがFS12/IS13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30eを用いてFS/IS管理テーブル40の検索を実行する。FSクラスタ数30fは、FS12内に存在する有効クラスタ数を示している。ISクラスタ数30gは、IS13内に存在する有効クラスタ数を示している。

20

【0079】

・FS/IS管理テーブル40（正引き）

つぎに、図13を用いてFS/IS管理テーブル40について説明する。FS/IS管理テーブル40は、論理クラスタ単位でFS12（FSIB12aも含む）もしくはIS13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13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30のクラスタテーブルポインタ30eのフィールドに保持されている。図13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40a、論理ブロックID40b、論理ブロック内クラスタ位置40c、FS/ISブロックID40d、nextポインタ40eから構成されている。FS/IS管理テーブル40は、論理クラスタアドレス40a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ID40b、論理ブロック内クラスタ位置40c（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

【0080】

論理ブロックID40bは、当該論理クラスタアドレス40a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ID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40cは、論理ブロックID40b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40a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40cは、kビットで 2^k 位置を識別する。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のインデックスであるFS/ISブロックIDが登録されている。FS/ISブロックIDは、FS12またはIS13に所属する論理ブロック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

40

50

ト内の次のエントリへのポインタを示している。

【 0 0 8 1 】

・ M S 論理ブロック管理テーブル 3 5 (逆引き)

つぎに、図 1 4 を用いて M S 論理ブロック管理テーブル 3 5 について説明する。M S 論理ブロック管理テーブル 3 5 は、M S 1 1 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、M S 論理ブロック管理テーブル 3 5 には、F S 1 2 (F S I B 1 2 も含む) 、I S 1 3 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。M S 論理ブロック管理テーブル 3 5 は、論理ブロック I D 3 5 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、3 2 K 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 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 から構成されている。この M S 論理ブロック管理テーブル 3 5 は、記憶デバイス位置に対応する論理ブロック I D 3 5 a 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。

【 0 0 8 2 】

トラック管理ポインタ 3 5 b は、論理ブロック I D 3 5 a 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 3 0 を検索することができる。有効トラック数 3 5 c は、論理ブロック I D 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 3 5 d は、論理ブロック I D 3 5 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 3 5 e は、当該論理ブロックエントリが M S 1 1 (M S I B 1 1 a も含む) として管理されている場合に “ 1 ” である。

【 0 0 8 3 】

・ F S / I S 論理ブロック管理テーブル 4 2 (逆引き)

つぎに、図 1 5 を用いて F S / I S 論理ブロック管理テーブル 4 2 について説明する。F S / I S 論理ブロック管理テーブル 4 2 は、F S / I S ブロック I D 4 2 a をインデックスとした配列形式で構成され、F S 1 2 または I S 1 3 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 I D との対応、F S / I S 内クラスタ管理テーブル 4 4 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。F S / I S 論理ブロック管理テーブル 4 2 は、主に F S / I S 管理テーブル 4 0 中の F S / I S ブロック I D 4 0 d を用いてアクセスされる。各エントリは、論理ブロック I D 4 2 b 、ブロック内クラスタテーブル 4 2 c 、有効クラスタ数 4 2 d 、書き込み可能先頭ページ 4 2 e 、Valid フラグ 4 2 f から構成されている。この M S 論理ブロック管理テーブル 3 5 は、記憶デバイス位置に対応する F S / I S ブロック I D 4 2 a 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。

【 0 0 8 4 】

論理ブロック I D 4 2 b には、M S 論理ブロック管理テーブル 3 5 に登録された論理ブロックの中で、F S 1 2 (F S I B 1 2 も含む) 、I S 1 3 に所属している論理ブロックに対応する論理ブロック I D が登録される。ブロック内クラスタテーブル 4 2 c 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 F S / I S 内クラスタ管理テーブル 4 4 へのインデックスが登録される。有効クラスタ数 4 2 d 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に記憶されているクラスタのうちの有効なものの個数 (最大 2^k 個) を示している。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 $FS12$ ($FSIB12$ も含む) または $IS13$ として管理されている場合に “ 1 ” である。

【 0085 】

・ FS/IS 内クラスタ管理テーブル 44 (逆引き)

つぎに、図 16 を用いて FS/IS 内クラスタ管理テーブル 44 について説明する。 FS/IS 内クラスタ管理テーブル 44 は、 $FS12$ もしくは $IS13$ 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 $FS12$ および $IS13$ に所属する論理ブロック数 (P 個) 分だけ保持されており、 FS/IS 論理ブロック管理テーブル 42 のブロック内クラスタテーブル 42c 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 44a 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 44a 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、 FS/IS 管理テーブル 40 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 44a 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。

10

【 0086 】

・ 論物変換テーブル 50 (正引き)

つぎに、図 17 を用いて論物変換テーブル 50 について説明する。論物変換テーブル 50 は、論理ブロック $ID50a$ 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、最大 32K エントリまで持つことができる。論物変換テーブル 50 は、論理ブロック ID と物理ブロック ID 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 50b、消去回数 50c、読み出し回数 50d から構成される。この論物変換テーブル 50 は、論理ブロック ID から物理ブロック ID (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。

20

【 0087 】

物理ブロックアドレス 50b は、1 つの論理ブロック $ID50a$ に所属する 8 個の物理ブロック ID (物理ブロックアドレス) を示している。消去回数 50c は、当該論理ブロック ID の消去回数を示している。バッドブロック (BB) 管理は、物理ブロック (512KB) 単位に行われるが、消去回数の管理は、32ビット倍速モードによる 1 論理ブロック (4MB) 単位に管理される。読み出し回数 50d は、当該論理ブロック ID の読み出し回数を示している。消去回数 50c は、例えば、NAND 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 50d 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。

30

【 0088 】

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。

RC 管理: RC クラスタ管理テーブル

WC 管理: WC クラスタ管理テーブル、WC トラック管理テーブル

MS 管理: トラック管理テーブル 30、MS 論理ブロック管理テーブル 35

FS/IS 管理: 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

40

【 0089 】

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、 $MS11$ 、 $MSIB11$

50

a、T F S 1 1 bに割り当てた論理ブロックなどを管理している。また、F S / I S 構造管理テーブル（図示せず）において、F S 1 2、F S I B 1 2 a、I S 1 3を含めたF S / I S 領域の構造を管理しており、具体的には、F S 1 2、F S I B 1 2 a、I S 1 3に割り当てた論理ブロックなどを管理している。

【 0 0 9 0 】

・ Read処理

つぎに、図 1 8 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。A T A コマンド処理部 1 2 1 から、Readコマンドおよび読み出しアドレスとしてのL B A が入力されると、データ管理部 1 2 0 は、図 9 に示したR C クラスタ管理テーブル 2 3 と図 1 0 に示したW C クラスタ管理テーブル 2 5 を検索する（ステップ S 1 0 0 ）。具体的には、L B A のクラスタアドレスのL S B (k - i) ビット（図 7 参照）に対応するlineをR C クラスタ管理テーブル 2 3 とW C クラスタ管理テーブル 2 5 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 2 3 b、2 5 c をL B A のトラックアドレスと比較し（ステップ S 1 1 0 ）、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R C クラスタ管理テーブル 2 3 またはW C クラスタ管理テーブル 2 5 の該当line、該当wayに対応するW C 2 1 またはR C 2 2 のデータを読み出して、A T A コマンド処理部 1 2 1 に送る（ステップ S 1 1 5 ）。

【 0 0 9 1 】

データ管理部 1 2 0 は、R C 2 2 またはW C 2 1 でヒットしなかった場合は（ステップ S 1 1 0 ）、検索対象のクラスタがN A N D メモリ 1 0 のどこに格納されているかを検索する。データ管理部 1 2 0 は、まず、図 1 2 に示したトラック管理テーブル 3 0 を検索する（ステップ S 1 2 0 ）。トラック管理テーブル 3 0 は、論理トラックアドレス 3 0 a でインデックスされているため、L B A 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 3 0 a のエントリだけをチェックする。

【 0 0 9 2 】

まず、チェックしたいL B A 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 3 0 b 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M S 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 S 1 3 0 ）。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 3 0 a のエントリ中の論理ブロックID 3 0 c と論理ブロック内トラック位置 3 0 d から得て、さらにL B A のクラスタアドレスのL S B (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、N A N D メモリ 1 0 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N A N D 層管理部 1 2 0 b では、上記のようにしてトラック管理テーブル 3 0 から取得した論理ブロックID 3 0 c と論理ブロック内トラック位置 3 0 d と、L B A の論理クラスタアドレスのL S B (k - i) ビットを物理N A N D 層管理部 1 2 0 c に与える。

【 0 0 9 3 】

物理N A N D 層管理部 1 2 0 c では、論理ブロックID 3 0 c 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 1 7 に示す論物変換テーブル 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 3 0 d から算出し、さらにL B A のクラスタアドレスのL S B (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。N A N D メモリ 1 0 のM S 1 1 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 ）。

【 0 0 9 4 】

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

10

20

30

40

50

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
10
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）。

【0 0 9 5】

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 6】

・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W r i t e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
30
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。

【0 0 9 7】

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。

【0 0 9 8】

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
40
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 9】

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 0は、図1 0に示したW Cクラスタ管理テーブル2 5を検索する（ステップS 3 0 0、S 3 0 5）。W C 2 1の状態は、図1 0に示したW Cクラスタ管理テーブル2 5の状態フラグ2 5 a（例えば3ビット）によって規定されている。状態フラグ2 5 aは、最も典型的
50

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 2 5 c が格納されている場合は (ステップ S 3 0 5)、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)。

【 0 1 0 0 】

そして、D R A M層管理部 1 2 0 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 1 2 1 に通知する。A T Aコマンド処理部 1 2 1 による書き込みが終了すると、W Cクラスタ管理テーブル 2 5 の該当エントリの状態フラグ 2 5 a をValid (使用不可) にし、さらにセクタ位置ビットマップ 2 5 b および論理トラックアドレス 2 5 c の欄に所要のデータを登録する。また、W Cトラック管理テーブル 2 4 を更新する。具体的には、W Cトラック管理テーブル 2 4 の各リスト中に既に登録済みの論理トラックアドレス 2 4 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W Cクラスタ数 2 4 b、way - lineビットマップ 2 4 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 2 4 d を変更する。また、W Cトラック管理テーブル 2 4 の各リスト中に登録済みの論理トラックアドレス 2 4 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 2 4 a、W Cクラスタ数 2 4 b、way - lineビットマップ 2 4 c、nextポインタ 2 4 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 3 2 0)。

【 0 1 0 1 】

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 2 5 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 3 0 5)。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 2 5 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 2 5 a を持つwayである。状態フラグ 2 5 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 2 5 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 1 2 1 に通知し、A T Aコマンド処理部 1 2 1 によって書き込みを実行させる。そして、前記同様、W Cクラスタ管理テーブル 2 5 およびW Cトラック管理テーブル 2 4 を更新する (ステップ S 3 2 0)。

【 0 1 0 2 】

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 2 5 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 2 5 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 1 2 1 に通知し、A T Aコマンド処理部 1 2 1 によって書き込みを実行させる。そして、前記同様、W Cクラスタ管理テーブル 2 5 およびW Cトラック管理テーブル 2 4 を更新する (ステップ S 3 2 0)。

【 0 1 0 3 】

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

10

20

30

40

50

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 WC 2 1 から NAND 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基づいて追い出すトラックすなわち WC 2 1 内エントリを選択する。

【 0 1 0 4 】

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 WC 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 WC 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 WC 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) 。 WC 2 1 から M S I B 1 1 a への追い出し、WC 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エントリの状態フラグ 2 5 a は、Valid (使用不可) から NAND メモリ 1 0 への追い出し待ちに移行される。

【 0 1 0 5 】

この追い出し先の判定は、WC 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、WC 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 WC クラスタ数 2 4 b のエントリが登録されており、この WC クラスタ数 2 4 b のエントリを参照することで WC 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 WC 2 1 での記憶位置を知ることができる。

【 0 1 0 6 】

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) WC 2 1 に登録されているトラックの数が所定数を越えた場合、
が成立した場合も、上記同様にして NAND メモリ 1 0 への追い出し処理を実行する。

【 0 1 0 7 】

WC 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 WC 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、次のような手順を実行する (ステップ S 3 1 0) 。

1 . WC 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、NAND 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 WC 2 1 内に存在しないクラスタについては NAND 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、WC 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 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. Copy 終了後に既に FS、IS、MS 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。

【0108】

このような WC 2 1 から MS IB 1 1 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。

【0109】

一方、WC 2 1 から MS IB 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 MS 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 (MS 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 ID 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 ID 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、MS 論理ブロック管理テーブル 3 5 における、論理ブロック ID 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。

【0110】

なお、他の領域 (FS 1 2 や IS 1 3) 等から MS 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 MS 1 1 内部のコンパクション処理による MS 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 WC 2 1 内有効クラスタも同時に MS に書き込まれる。WC 2 1 から MS 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 WC 2 1 上から削除 (無効化) されることになる。

【0111】

WC FS IB (Copy)

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 WC 2 1 から FS IB 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. WC 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、NAND 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。

2. WC 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 FS IB 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 FS IB 1 2 a に書き込む。

4. Copy 終了後に既に FS、IS 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。

【0112】

このような WC 2 1 から FS IB 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

10

20

30

40

50

、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、W C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、例えば直前のリストのnextポインタ24dが変更または削除されて、無効化される。一方、W C21からF S I B 12a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30のクラスタテーブルポインタ30e、F Sクラスタ数30fなどを更新するとともに、F S / I S管理テーブル40の論理ブロックI D 40b、論理ブロック内クラスタ位置40cなどを更新する。なお、もともとF S 12に存在していなかったクラスタについては、F S / I S管理テーブル40のリンクドリストへのリストが追加される。この更新に伴い、M S論理ブロック管理テーブル35、F S / I S論理ブロック管理テーブル42、およびF S / I S内クラスタ管理テーブル44の該当個所を更新する。

10

【0113】

C I B 処理

上記のようなW C F 処理が終了すると、つぎに、論理N A N D層管理部120bは、W C F 処理によって書き込まれたF S I B 12aのデータをF S 12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 11aのデータをM S 11にM o v eする処理などを含むC I B 処理を実行する。C I B 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C I B 処理においては、基本的には、先ずM S 11でのC I B 処理が行われ(ステップS330)、つぎに、F S 12でのC I B 処理が行われ(ステップS340)、つぎに再びM S 11でのC I B 処理が行われ(ステップS350)、つぎにI S 13でのC I B 処理が行われ(ステップS360)、最後に再びM S 11でのC I B 処理が行われる(ステップS370)。なお、F S 12からM S I B 11aへの追い出し処理、あるいはF S 12からI S 13への追い出し処理、あるいはI S 13からM S I B 11a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。M S 11、F S 12およびI S 13でのC I B 処理を別々に説明する。

20

【0114】

M S 11のC I B 処理

まず、M S 11でのC I B 処理について説明する(ステップS330)。W C 21、F S 12、I S 13からM S 11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M S I B 11aに書き込まれる。M S I B 11a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30を更新してトラックが配置される論理ブロックI D 30cとブロック内トラック位置30dなどを変更する(M o v e)。M S I B 11aに新たなトラックデータが書き込まれた場合、もともとM S 11もしくはT F S 11b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、M S 論理ブロック管理テーブル35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、M S 論理ブロック管理テーブル35の該当エントリ中のトラック管理ポインタ35b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35eが無効化される。このような無効化などにより、M S 11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

【0115】

データ管理部120は、このような事態が発生して、M S 11に割り当てられている論理ブロックの個数がM S 11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F Bを作る。無効なフリーブロックF Bは、物理N A N D層管理部120cに返却される。そして、論理N A N D層管理部120bは、M S 11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F Bを物理N A N D層管理部120cから取得す

50

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C o p yしたりすることで、物理N A N D層管理部1 2 0 cに返却する無効なフリーブロックF B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、W C、F S、I S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、T F S 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。

【 0 1 1 6 】

以下に、M S I B 1 1 a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、M S I B 1 1 aからM S 1 1またはT F S 1 1 b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 . M S 論理ブロック管理テーブル3 5のValidフラグ3 5 eを参照することにより、M S 1 1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F Bとする。

2 . M S I B 1 1 aでフルに成った論理ブロックをM S 1 1に追い出す。具体的には、前述したM S 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M S I B 管理下からM S 管理下に移し変える。

3 . M S 1 1に割り当てられている論理ブロックの個数がM S 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か否かを判断し、発生している場合に、以下のM S コンパクションを実行する。

4 . 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、T F S 1 1 b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C o p y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W C 2 1、F S 1 2、I S 1 3に有効クラスタを持つ場合にはそれらもマージする。

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F B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れば、T F S 1 1 bの先頭にM o v e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C o p yして、無効なフリーブロックF B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M S I B 1 1 a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F Bとする。

1 0 . M S 1 1に割り当てられている論理ブロックの個数がM S 1 1として許容されるブロック数の上限値を下回ると、M S コンパクション処理を終了する。

【 0 1 1 7 】

F S 1 2のC I B 処理

つぎに、F S 1 2でのC I B 処理について説明する(ステップS 3 4 0)。W C 2 1からF S I B 1 2 a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F S I B 1 2 a中に作られた場合、F S I B 1 2 a中のそれらのブロックは、F S I B 1 2 aからF S 1 2に対してM o v eされる。このM o v eにともなって複数の論理ブロックで構成されるF I F O構造のF S 1 2から古い論理ブロックが追い出される状況が発生する。

【 0 1 1 8 】

F S I B 1 2 aからF S 1 2に対する追い出しおよびF S 1 2からのブロック追い出しは、具体的には、次のように実現される。

1 . F S / I S 論理ブロック管理テーブル4 2のValidフラグ3 5 eなどを参照することにより、F S 1 2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

10

20

30

40

50

なフリーブロック F B とする。

2. F S I B 1 2 a でフルに成ったブロックを F S 1 2 に追い出す。具体的には、前述した F S / I S 構造管理テーブル (図示せず) を更新して、該当ブロックを F S I B 管理下から F S 管理下に移し変える。

3. F S 1 2 に割り当てられている論理ブロックの個数が F S 1 2 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 I S 1 3 に移動せずに、直接 M S 1 1 に移動すべきものを決定する (実際には、M S 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) 。

10

(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 F S 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 3 0 の F S クラスタ数 3 0 f 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 5 0 %) 以上だった場合、そのトラックを M S への追い出し候補とする。

5. M S 1 1 に追い出すべきトラックを M S I B 1 1 a 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 M S I B 1 1 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 I S 1 3 に M o v e する。

20

なお、F S 1 2 から M S I B 1 1 a への追い出しが発生したときには、その直後、M S 1 1 での前述した C I B 処理が実行される (ステップ S 3 5 0) 。

【 0 1 1 9 】

I S 1 3 の C I B 処理

つぎに、I S 1 3 での C I B 処理について説明する (ステップ S 3 6 0) 。上記した F S 1 2 から I S 1 3 へのブロック移動によって論理ブロックが I S 1 3 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 I S 1 3 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、I S 1 3 では、まず M S 1 1 への 1 ~ 複数個の論理ブロックの追い出しを行った後、I S 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。

30

1. I S 1 3 に含まれるトラックをトラック内の有効クラスタ数 \times 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 M S I B 1 1 a 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、I S 1 3 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 F B として返還する。

40

なお、I S 1 3 から M S I B 1 1 a への追い出しが発生したときには、その直後、M S 1 1 での前述した C I B 処理が実行される (ステップ S 3 7 0) 。

【 0 1 2 0 】

図 2 0 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。F S 1 2 は、基本的には、W C 2 1 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、W C 2 1 から F S 1 2 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、F S 1 2、I S 1 3、M S 1 1 からのデータがコピーされる。W C 2 1 では、W C クラスタ管理テーブル 2 5 のタグ中のセクタ位置ビットマップ 2 5 b によって当該クラスタア

50

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。

【0121】

10

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。

【0122】

20

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

【0123】

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。

【0124】

・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われる。

40

【0125】

・IS構成

50

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。

【0126】

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タの付け替えにより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。

【0127】

・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。

【0128】

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。

【0129】

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。

【0130】

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ックに含まれるWC21、FS部12Q、IS部13Q内の有効クラスタがDRAM20の作業領域を介してMSコンパクションバッファ11c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、MSIB11aおよびTFS11bに登録されている

10

20

30

40

50

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。

【0131】

図22は、データ管理部120のより詳細な機能構成を示すものである。データ管理部120は、前述したように、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。

【0132】

DRAM層管理部120aは、RCクラスタ管理テーブル23、WCクラスタ管理テーブル25、WCトラック管理テーブル24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120bは、トラック管理テーブル30、MSブロック管理テーブル35、FS/IS管理テーブル40、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の他に、MS構造管理テーブル60、FS/IS構造管理テーブル65を備え、これらの管理テーブルに基づいてNANDメモリ10の論理NAND層の管理を行う。物理NAND層管理部120cは、論物変換テーブル50の他に、パッドブロック管理テーブル(BB管理テーブル)200、予約ブロック管理テーブル(RBブロック管理テーブル)210、フリーブロック管理テーブル(FB管理テーブル)220、アクティブブロック管理テーブル(AB管理テーブル)230を備え、これらの管理テーブルを用いてNANDメモリ10の物理NAND層の管理を行う。

【0133】

・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(ch0、ch1、ch2、ch3)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10a~10d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112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。

【0134】

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。

【0135】

図24は、論物変換テーブル50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50は、図17に示した論物変換テーブル50に対し、論理ブロックID50a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50eのフィールドが更に追加されている。消去時刻50e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112の通電時間等を利用すればよい。消去時刻50eは、後述するFB管理テーブル220でのフリーブロックFB管理に利用される。

【0136】

BB管理テーブル200は、物理ブロック(512kB)単位でパッドブロックBBを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ID 200aが保持されている。

【0137】

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。

【0138】

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。

【0139】

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。

【0140】

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。

【0141】

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。

【0142】

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。

【0143】

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される。

【0144】

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

10

20

30

40

50

であるリードディスタースの影響で、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。

【0145】

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。先ず、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。

10

【0146】

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。

20

【0147】

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。

【0148】

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。

30

【0149】

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。

40

【0150】

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

50

には以下のような処理を行う。例えば、論理ブロック L 1 の 4 行 3 列の物理ブロックがバッドブロック B B として B B 管理テーブル 2 0 0 に登録されたとし、さらに、このバッドブロック B B 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 R B が存在しない場合には、まず、論理ブロック L 1 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 R B 管理テーブル 2 1 0 に登録する。その後は、前記同様、F B 管理テーブル 2 2 0 から新たなフリーブロック F B を確保し、前記同様の論理ブロック I D の入れ替えを実行した後、F B 管理テーブル 2 2 0 から取得した論理ブロック I D を使用不可とする。

【 0 1 5 1 】

このように、物理 N A N D 層管理部 1 2 0 c は、バッドブロック B B が発生した場合でも、論理ブロック I D の入れ替えを実行しているので、論理 N A N D 層管理部 1 2 0 b で使用する論理ブロック I D は、バッドブロック B B 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、L B A 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 N A N D 層管理部 1 2 0 b 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。

【 0 1 5 2 】

つぎに、3 2 ビット倍速モードでの消去処理について説明する。物理 N A N D 層管理部 1 2 0 c は、N A N D メモリ 1 0 のデータが論理ブロック単位で消去されるごとに、図 2 4 に示す論物変換テーブル 5 0 の消去された論理ブロックに対応する論理ブロック I D 中の消去回数 5 0 c 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 5 0 e を最新のデータに更新する。

【 0 1 5 3 】

・ 論理 N A N D 層

つぎに、図 2 8、図 2 9 を用いて論理 N A N D 層での管理に使用される M S 構造管理テーブル 6 0 および F S / I S 構造管理テーブル 6 5 について説明する。図 2 8 に示す M S 構造管理テーブル 6 0 は、M S 部 1 1 Q 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。M S 構造管理テーブル 6 0 は、M S F B 1 1 a a、M S 追記 I B 1 1 a b、T F S 1 1 b として割り当てられた論理ブロック I D を管理する M S バッファ管理テーブル 6 1 と、M S 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 I D を保持する有効トラック数別論理ブロック I D リスト 6 2 と、状態情報としての最大論理ブロック数 M B L および有効論理ブロック数 V B L を管理する領域 6 3、6 4 とを備える。

【 0 1 5 4 】

M S 構造管理テーブル 6 0 では、M S F B 1 1 a a 用、M S 追記 I B 1 1 a b 用、T F S 1 1 b 用に、それぞれ所要エントリ数の固定フィールド 6 1 a ~ 6 1 c が用意され、各固定フィールド 6 1 a ~ 6 1 c に論理ブロック I D が記録される。T F S 1 1 b 用のフィールド 6 1 c は、リンクドリスト構造を有し、F I F O 構造を有する T F S 1 1 b に対する F I F O 的な管理が行われている。

【 0 1 5 5 】

有効トラック数別論理ブロック I D リスト 6 2 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 I D が記録される。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 I D リスト 6 2 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 I D リスト 6 2 には、M S I B 1 1 a、T F S 1 1 b として M S バッファ管理テーブル 6 1 に登録されている論理ブロックについては、エントリされない。

【 0 1 5 6 】

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

10

20

30

40

50

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 7】

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 8】

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 9】

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 × 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 6 0】

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 6 1】

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N A N D 層管理部1 2 0 cに対し、フリーブロックF Bの獲得要求を出して、フリーブロックF Bを該F Bとして割り当てられた論理ブロックI Dと共に取得する。そして、取得したフリーブロックF Bの書き込み可能トラック数2ⁱをD R A M 層管理部1 2 0

10

20

30

40

50

aに通知する。

【0162】

D R A M層管理部120aでは、論理N A N D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、W C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、トラック内セクタ埋めを行う必要はないとする。D R A M層管理部120aは、追い出すトラックがN A N D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N A N D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。

【0163】

この通知を受けると、論理N A N D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F S / I S管理テーブル40を検索し、N A N D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N A N D層管理部120cに通知する。これによって、物理N A N D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W C 21からM S追記I B 11abへの追い出しが行われる。

【0164】

物理N A N D層管理部120cから、W C 21 M S追記I B 11abの追い出し終了が通知されると、論理N A N D層管理部120bは、M S追記I B 11abとしてあらたなフリーブロックF Bを物理N A N D層管理部120cから取得した場合は、物理N A N D層管理部120cから与えられたフリーブロックF Bの論理ブロックI Dに対応するM S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、M Sバッファ管理テーブル61のM S追記I B用のフィールド61bに論理ブロックI Dを登録し、M S構造管理テーブル60の有効論理ブロック数V B Lをインクリメントする。

【0165】

また、論理N A N D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、W C 21からM S追記I B 11ab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I D 30c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。

【0166】

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、W C 21から追い出された新しいトラックに関するデータをM S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I Dに対応するエントリに対し登録する。M S論理ブロック管理テーブル35への登録としては、M S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ト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。

【0167】

W C 21とM S 11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I D 30c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I D 30cがM S 11に存在していたときの論理ブロックI DからM S追記I B 11abに対応する論理ブロックI D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。

【0168】

さらに、M S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M S構造管理テーブル60の有効トラック数別論

10

20

30

40

50

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、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。

【0169】

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、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。

【0170】

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。

【0171】

・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。

【0172】

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。

【0173】

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。

【0174】

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。

10

【0175】

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。

【0176】

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。

20

【0177】

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。

30

【0178】

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。

【0179】

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FIFO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。

40

【0180】

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

50

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。

【0181】

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。

10

【0182】

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。

20

【0183】

・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。

30

【0184】

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）。

【0185】

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。

40

【0186】

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

50

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 B_{Lfsmax} を越えているかを判断する。

【0187】

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 B_{Lfsmax} 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 F_B 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

【0188】

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を判断する（ステップ S455）。具体的には、FS12 の FIF O 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。

【0189】

20

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 FS FIF O 管理テーブル 65 から FIF O 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。

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。

30

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。

【0190】

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 a 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

【0191】

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 a 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 IB11a b 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 追記 IB11a b に対してトラック単位に追記書き込みする（ステップ S460）。その後、FIF O 構造の FS12 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 NAND 層管理部 120c に返却す

50

る。

【0192】

・MS11のCIB処理(図19:ステップS350)

このようにして、FSからMSIB11aへの追い出しが発生すると、つぎに、MS部11QのCIB処理が再び実行される(ステップS480)。このステップS480のMS部11QのCIB処理は、先の第1回目のMS部11Q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。このMS部11QでのCIB処理の後、論理NAND層管理部120bは、FS12からMSIB11a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S455)。この追い出し条件が成立すると、前述した、FS12からMSIB11aへの2論理ブロックの追い出し、およびMS11でのCIB処理が、再度実行される。このような処理が、ステップS455の判断がNOになるまで繰り返される。

10

【0193】

・FS12のCIB処理

ステップS455の判断がNOになると、論理NAND層管理部120bは、つぎに、FS12からISIB13a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S490)。具体的には、上記FS12からMSIB11aへの追い出し処理の際に、チェックされたFIFO構造を有するフル状態のFS12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S490でのFS12からIS13への追い出し条件が成立したとして、FS12からISIB13aへの追い出しを実行する。

20

【0194】

すなわち、ステップS490で条件が成立した場合は、MSIB11a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ISIB13aへMoveする(ステップS500)。このステップS500においては、例えば1論理ブロックの追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S520~S585の手順を経た後、ステップS590の判断により、ステップS500でのFS12からISIB13aへの追い出しが再度発生することもある。

【0195】

ステップS500での再追い出しがある状態とは、例えば、FSIB12aに複数のフルになった論理ブロックを有するバッファ(FSFB12aまたはFS追記IB12ab)が存在する状態において、FIFO構造を有するFS12がフルであった場合には、FSIB12aからFS12へのフルになったブロックのMoveに伴い、FS12からMSIB11aまたはISIB13a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、FS12からISIB13a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。

30

【0196】

・ISのCIB処理(図19ステップS360)

つぎに、ステップS490での条件が成立したときに、IS13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30の他に、図31のフローチャートを参照して説明する。まず、論理NAND層管理部120bは、前記と同様にして、IS部13Q内に無効論理ブロックが存在するか否かを調べ、IS部13Q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S520)。返却された論理ブロックのエントリに対応する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のエントリはValidフラグ35eおよび42fがそれぞれInvalidにされる。

40

【0197】

つぎに、論理NAND層管理部120bは、FIFO構造を有するIS13の論理ブロック数が、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

50

否かを判断する。

【0198】

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 $B_{L\text{ismax}}$ 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 $MSFB11a$ 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 $S585$ ）。

【0199】

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 $S710$ ）。

【0200】

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。

【0201】

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。

【0202】

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 $S740$ ）。

【0203】

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。

【0204】

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 2^{i+1} 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。

10

20

30

40

50

スコア値 S = トラック内の有効クラスタ数 \times 有効クラスタ係数

有効クラスタ係数は、トラックが MS 部 1 1 Q 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多いとする。

【0205】

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 30 g 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 35 b 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 35 c のフィールドを見ることで取得することができる。

【0206】

論理 $NAND$ 層管理部 120 b 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 2^{i+1} 個のトラックを、 MS 1 1 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 2^{i+1} 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。

【0207】

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 2^{i+1} 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 2^{i+1} 個のトラックを、 MS 1 1 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 2^{i+1} 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。

【0208】

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 120 b は、該選出された 2 論理ブロック分のトラック（すなわち 2^{i+1} 個のトラック）を $MSIB$ 1 1 a の $MSFB$ 1 1 a 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 1 3 Q 内に存在しないクラスタについては、 WC 2 1 および FS 部 1 2、および MS 1 1 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。

【0209】

・ MS の CIB 処理（図 19 ステップ $S370$ ）

このようにして、 IS 1 3 から $MSIB$ 1 1 a への追い出しが発生すると、つぎに、 MS 1 1 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 MS 1 1 の CIB 処理は、先の第 1 回目の MS 1 1 での CIB 処理（ステップ $S400 \sim S420$ ）と同様なので、重複する説明は省略する。

【0210】

IS の CIB 処理

つぎに、論理 $NAND$ 層管理部 120 b は、 IS 1 3 から $MSIB$ 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 $S570$ ）。すなわち、論理 $NAND$ 層

10

20

30

40

50

管理部 120b 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 の有効クラスタ数 42d のフィールドなどを用いて、ステップ S540 での追い出しを行った後の IS 13 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、IS 13 から MSIB 11a への追い出し条件が成立したと判断する (ステップ S570)。

【0211】

IS 13 から MSIB 11a への追い出し条件が成立した場合は、手順をステップ S540 に移行し、図 31 のステップ S700 ~ S750 および S770 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 S570 の判断が YES である限り、IS 13 から MSIB 11a への 2 論理ブロック分の追い出し処理と、MS 11 での CIB 処理が繰り返し実行される。そして、ステップ S570 の判断が NO となった場合は、IS 13 でのコンパクション処理が実行される (ステップ S580)。

【0212】

IS コンパクション処理では、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 の有効クラスタ数 42d のフィールドなどを用いて、IS 部 13Q 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 IS コンパクションバッファ 13c に Copy する。この Copy 処理が終了すると、論理 NAND 層管理部 120b は、コンパクション元 (Copy 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 NAND 層管理部 120c 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 IS コンパクションバッファ 13c を IS 13 に Move する。

【0213】

このコンパクションの後、MSIB 13a 中のフルになった論理ブロックは IS 13 に Move される (ステップ S585)。この Move 処理は、具体的には、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 の MSIB 用フィールド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。

【0214】

この後、論理 NAND 層管理部 120b は、前述した FS 12 から MSIB 13a 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 S590)、この FS 12 から MSIB 13a 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 S500 に移行し、上述した手順を再度繰り返す。また、論理 NAND 層管理部 120b は、IS コンパクション処理の終了後、FS 12 から MSIB 13a 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。

【0215】

以上説明したように本実施の形態においては、IS 13 でのコンパクション処理に先立ち、図 30 のステップ S540 で、有効クラスタ数が多くてかつ MS に穴あきブロックが存在していることでソートされる上位の複数のトラックによって構成される複数のブロックを MS 11 へ追い出すようにしており、これによりコンパクション前に IS 13 の状態をコンパクションを実行し易い状態にすることができる。すなわち、IS 13 をコンパクションする時点では、有効クラスタ数の少ない論理ブロックが 2 つ以上存在する第 1 の状態と、有効クラスタ数の少ないブロックが 2 つ以上存在しない第 2 の状態が存在する。第 1 の状態は、そのままコンパクション処理が可能な状態であるが、第 2 の状態は、このままでは効率のよいコンパクション処理ができない状態である。そこで、本実施の形態においては、有効クラスタ数が多く、MS に穴あきブロックが存在していることでソートされる上位の複数のトラックによって構成される複数のブロックをコンパクション前に MS 11 へ追い出すようにしており、これにより IS 13 の状態をコンパクションを実行する前

10

20

30

40

50

に、上記の第 1 の状態に移行させることが可能となる。これにより、一度のコンパクションにかかる時間を平準化かつ高速化することが可能となる。

【 0 2 1 6 】

また、本実施の形態においては、トラック管理テーブル 3 0 の全てのエントリを検索するのではなく、落選トラックリスト中のスコア値 S が上位の N 個の論理トラックと、新規追加ブロック内トラックリスト中のスコア値 S が上位の M 個の論理トラックと、有効クラスタ数を持つことで選んだ新規検索トラックリスト中の L 個の論理トラックとの中から、スコア値 S が高い 2^{i+1} 個の論理トラックを $M S 1 1$ への追い出しを行う論理トラックとして選出するようにしており、スコア値に基づくサーチ精度を維持しつつサーチ速度を向上させることができる。また、トラック管理テーブル 3 0 の検索による新規検索トラックリストの作成時、検索済み最終トラックを記憶し、次回検索時は、この記憶した最終トラックの次のトラックから検索を行うサイクリックサーチを行うようにしているので、サーチ速度の向上を目的としてトラック管理テーブル 3 0 の全てのエントリを検索してないにもかかわらず、結果的に検索箇所が片寄ることなく、検索処理の精度を平準化することができる。

10

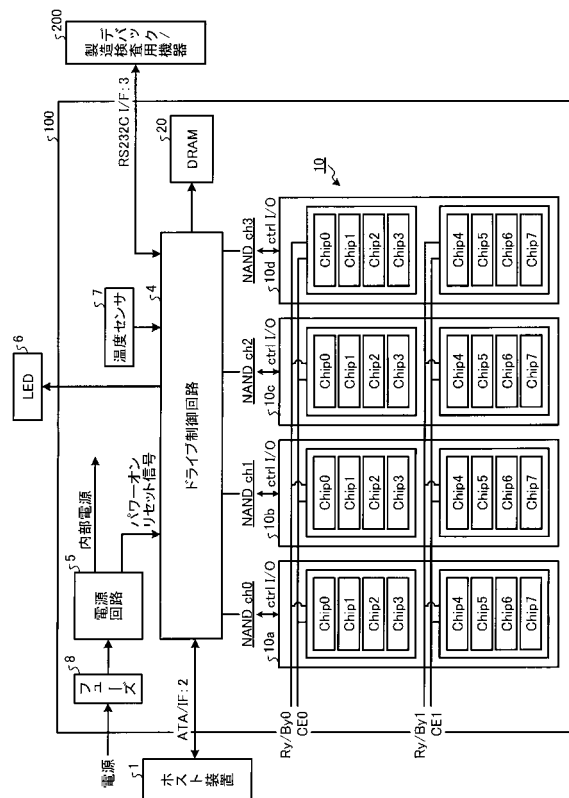
【 符号の説明 】

【 0 2 1 8 】

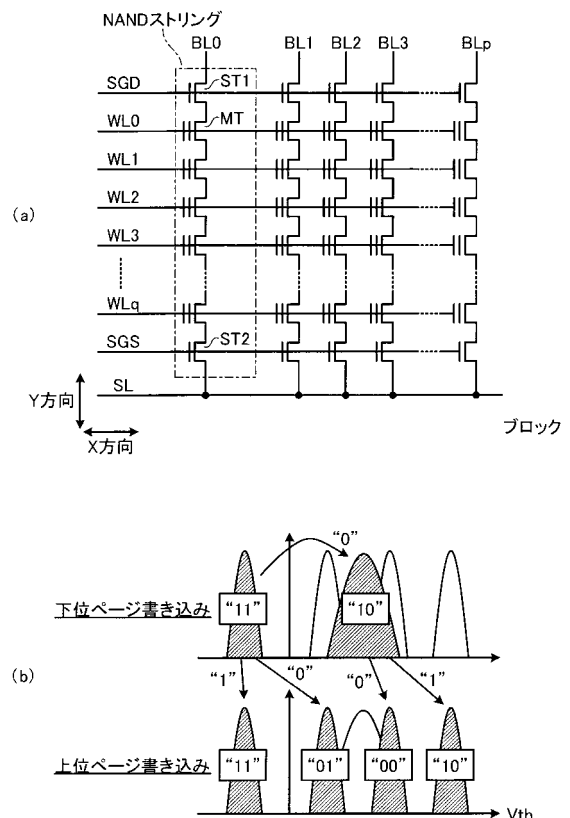
1 0 NANDメモリ、1 1、メインストレージ領域 (MS)、1 1 a MS入力バッファ (MSIB)、1 1 a a フルブロック入力バッファ (MSFB)、1 1 a b MS追加入力バッファ (MS追加IB)、1 1 b トラック中段ストレージ領域 (TFS)、1 2 前段ストレージ領域 (FS)、1 2 a FS入力バッファ (FSIB)、1 3 中段ストレージ領域 (IS)、1 3 a IS入力バッファ (ISIB)、2 0 DRAM、2 1 ライトキャッシュ (WC)、2 2、リードキャッシュ (RC)。

20

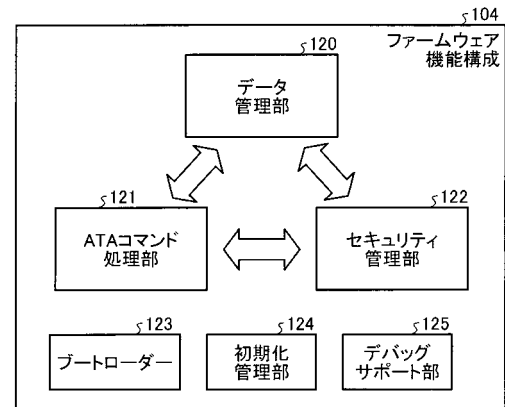
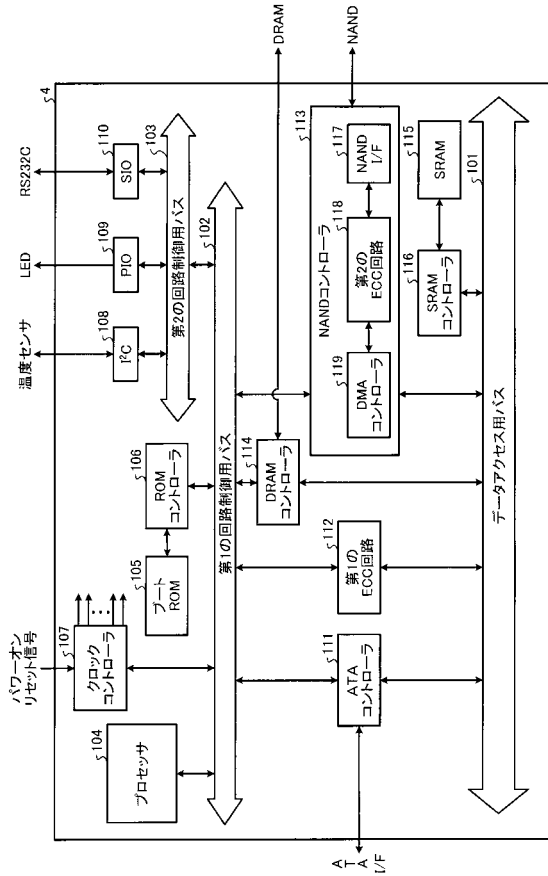
【 図 1 】



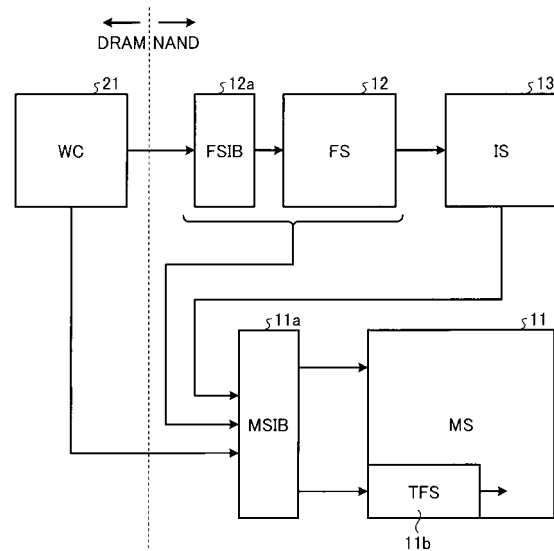
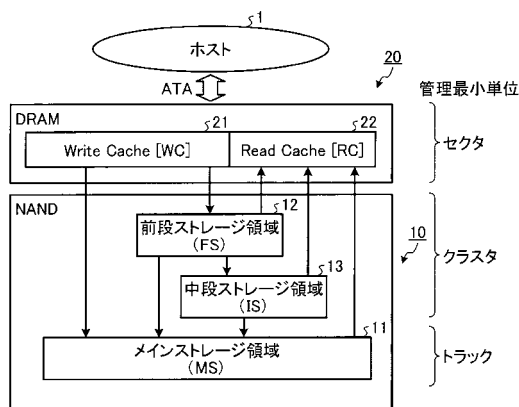
【 図 2 】



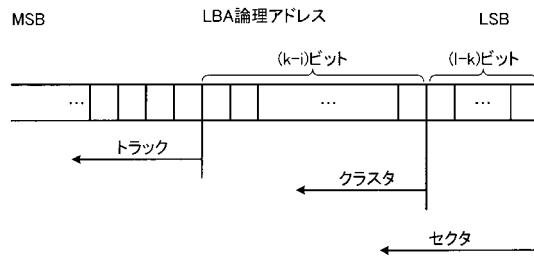
【 図 4 】



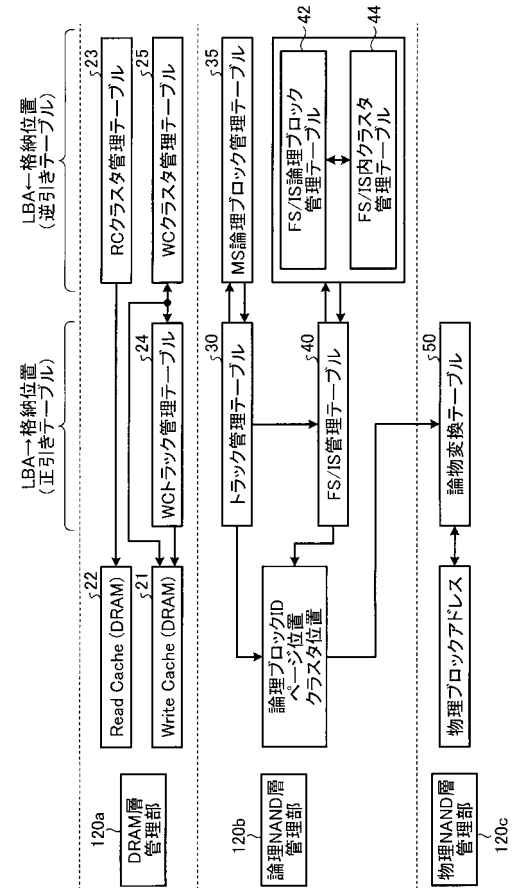
【 図 6 】



【図 7】



【図 8】



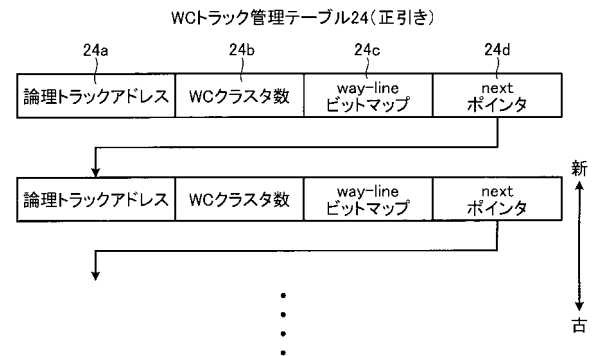
【図 9】

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】

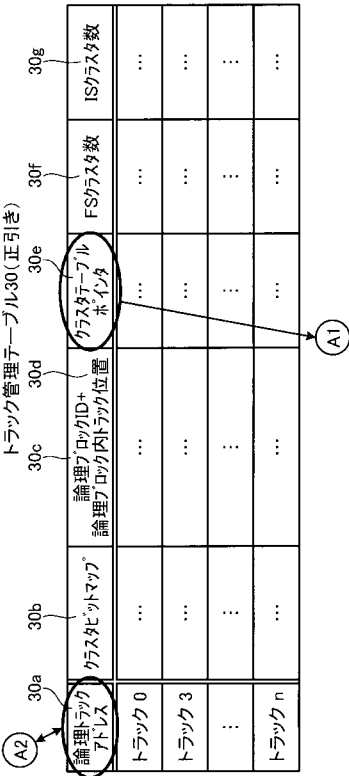


【図 10】

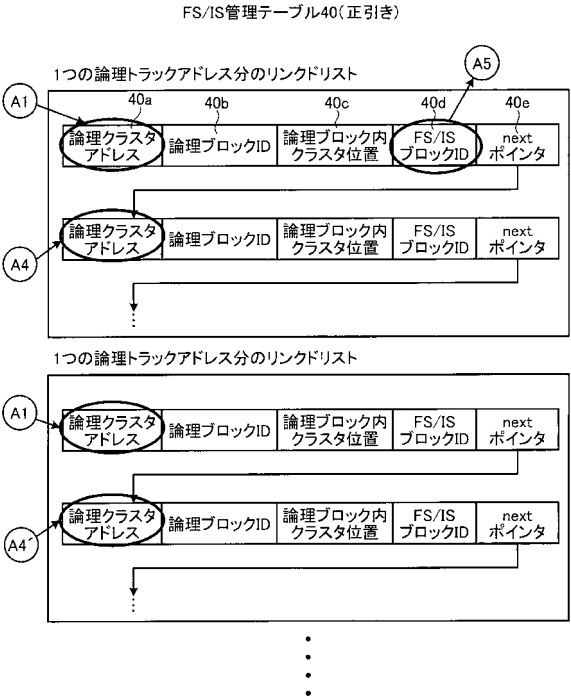
WCクラスタ管理テーブル25(逆引き): n-way セットアソシアティブ

	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

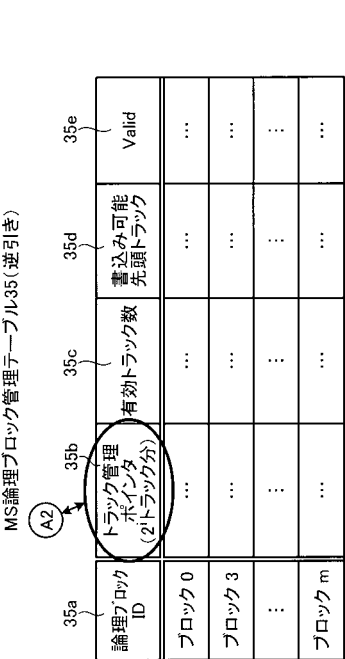
【図 1 2】



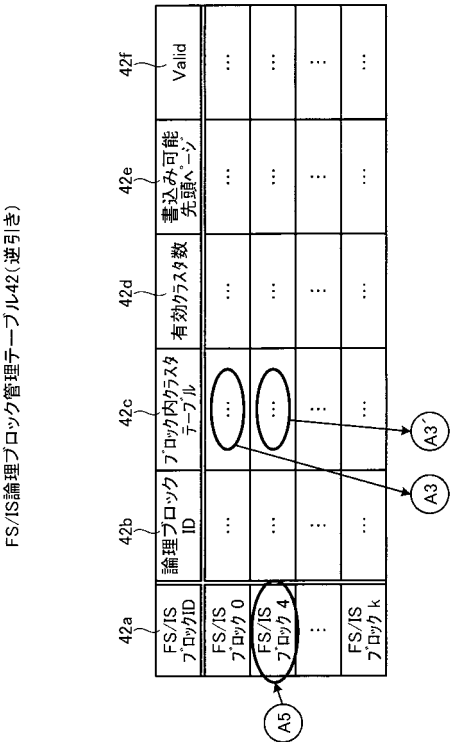
【図 1 3】



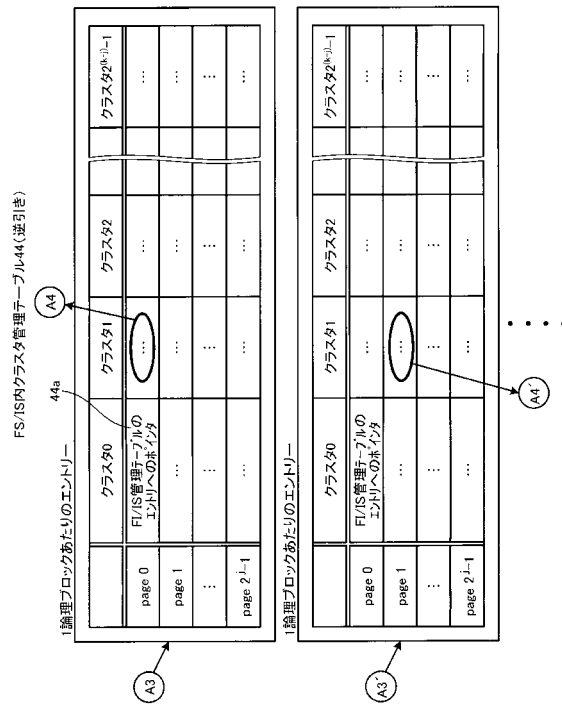
【図 1 4】



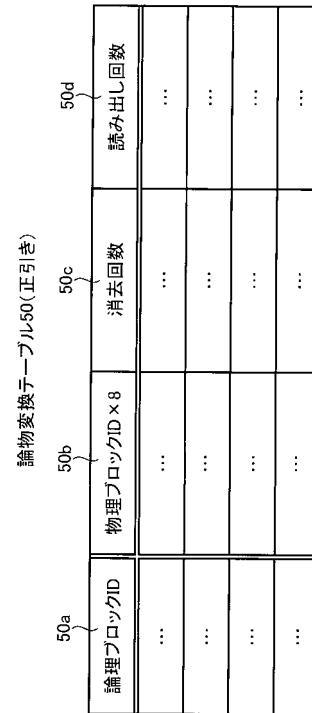
【図 1 5】



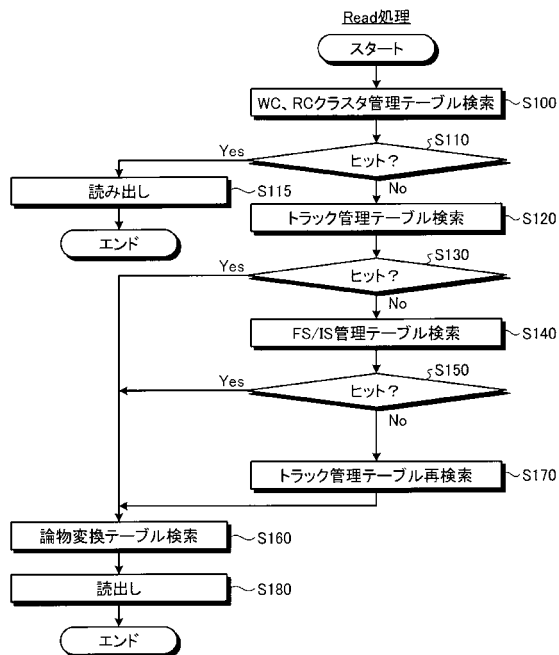
【 図 1 6 】



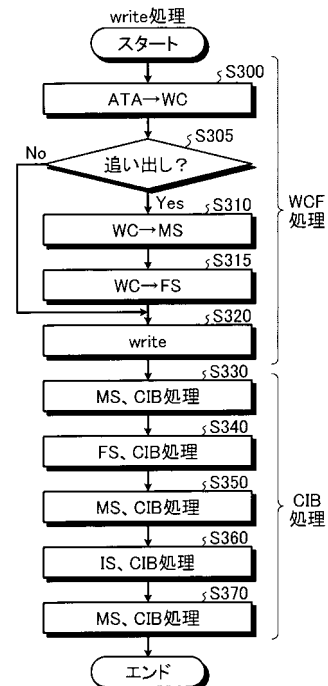
【 図 1 7 】



【 圖 1 8 】



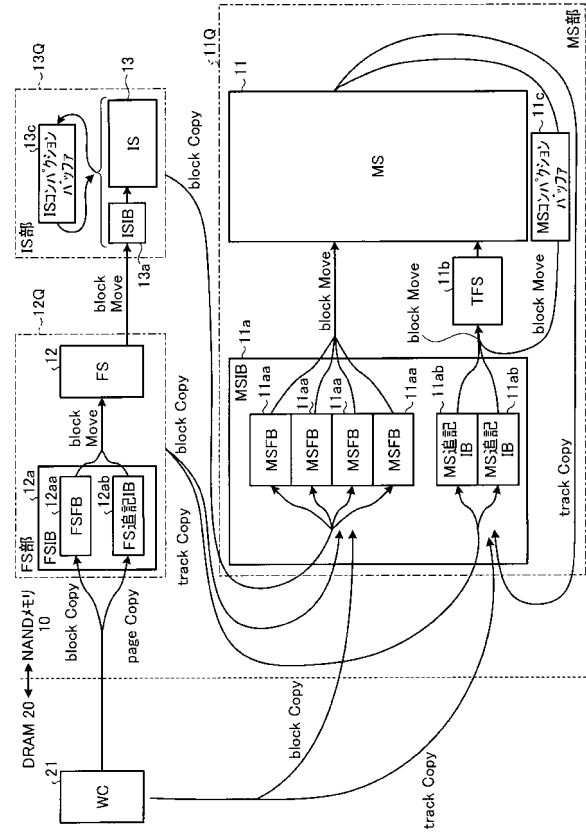
【 図 1 9 】



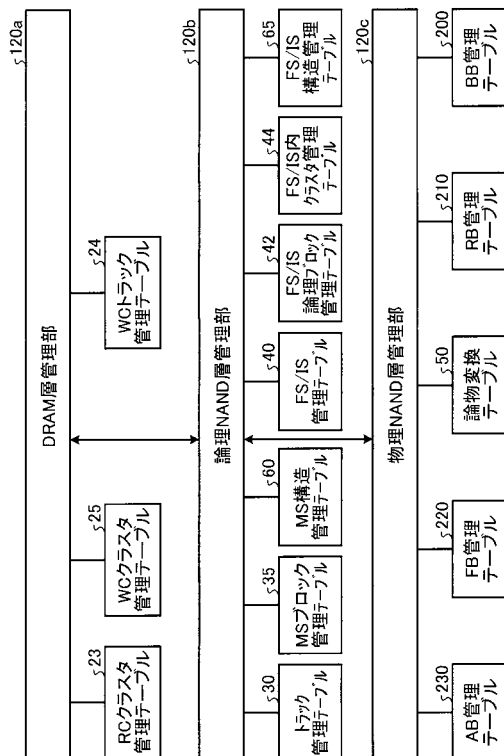
【 図 2 0 】

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

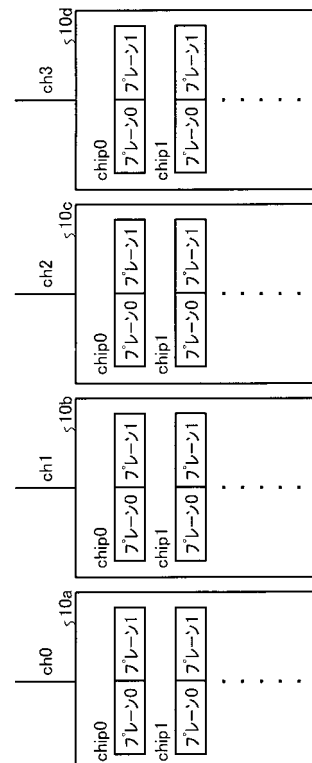
【 図 2 1 】



【 ㄨ 2 2 】



【 図 2 3 】



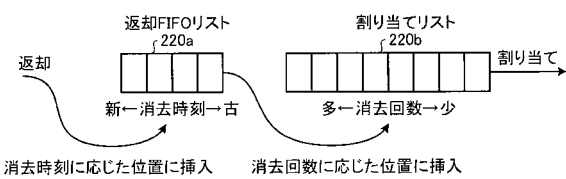
【図 2 4】

論物変換テーブル50(正引き)				
50a 論理ブロックID	50b 物理ブロックID × 8	50c 消去回数	50d 読み出し回数	50e 消去時刻
...
...
...
...

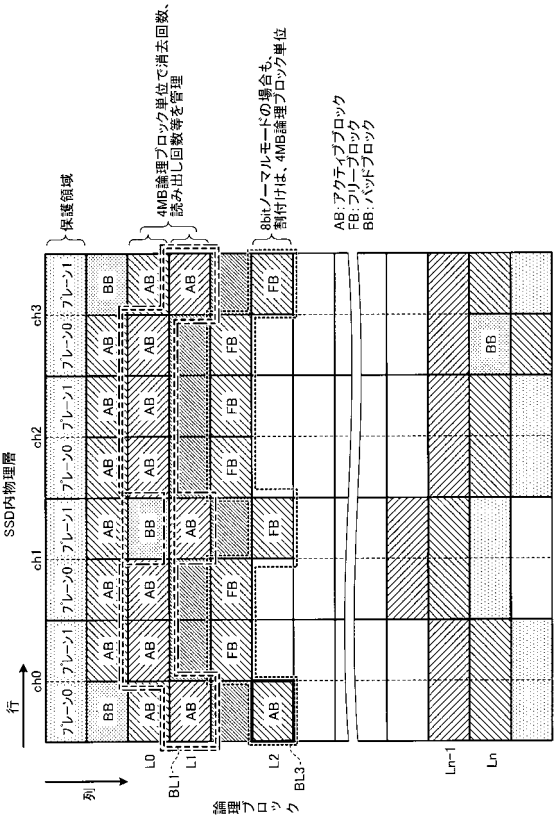
【図 2 5】

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 1
ID#0	ID#1	ID#2	ID#3	ID#4	ID#5	ID#7
物理 ブロックID
...
...
...
...

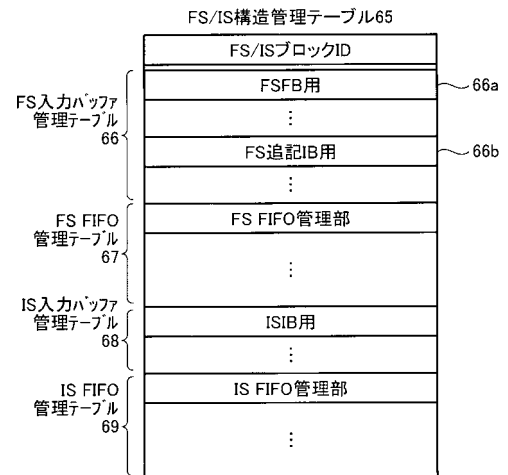
【図 2 6】



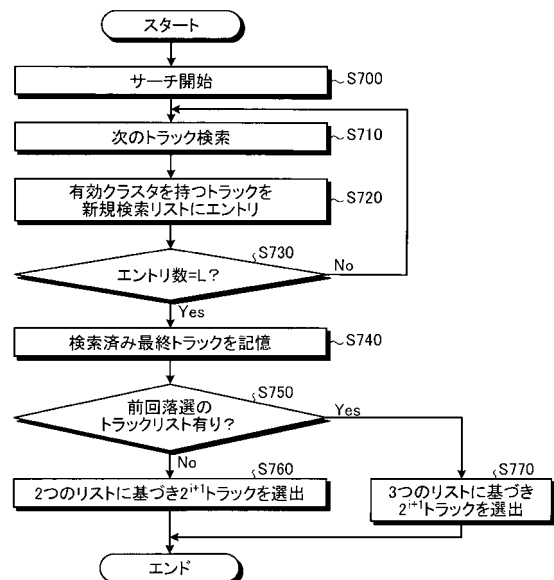
【図 2 7】



【 図 2 9 】



【 図 3 1 】



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
- (72)発明者 加藤 亮一
東京都港区芝浦一丁目1番1号 株式会社東芝内

審査官 野田 佳邦

- (56)参考文献 特開2006-11818(JP,A)
特開2008-33788(JP,A)
特開2009-211227(JP,A)
特開2009-211229(JP,A)
特開2009-211220(JP,A)
特開2009-211231(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|---------|-----------------------|
| G 0 6 F | 1 2 / 0 0 - 1 2 / 0 6 |
| G 0 6 F | 1 2 / 0 8 |