

(19) 日本国特許庁 (JP)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2011-519095

(P2011-519095A)

(43) 公表日 平成23年6月30日 (2011.6.30)

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

審査請求 未請求 予備審査請求 未請求 (全 53 頁)

(21) 出願番号 特願2011-506353 (P2011-506353)
 (86) (22) 出願日 平成21年4月10日 (2009.4.10)
 (85) 翻訳文提出日 平成22年12月13日 (2010.12.13)
 (86) 国際出願番号 PCT/US2009/040153
 (87) 国際公開番号 W02009/131851
 (87) 国際公開日 平成21年10月29日 (2009.10.29)
 (31) 優先権主張番号 12/110,050
 (32) 優先日 平成20年4月25日 (2008.4.25)
 (33) 優先権主張国 米国 (US)

(71) 出願人 506197901
 サンディスク コーポレーション
 アメリカ合衆国、95035、カリフォル
 ニア州、ミルピタス、マッカシー ブルバ
 ード 601
 (74) 代理人 100075144
 弁理士 井ノ口 壽
 (72) 発明者 シンクレア, アラン ダブリュー.
 イギリス連邦共和国、FK2 OBU、フ
 ォールカーク、マディストン、キャンディ
 、ザ コテージース
 Fターム (参考) 5B060 AA08 AA09 AB26

最終頁に続く

(54) 【発明の名称】 マルチバンク記憶装置のためのストレージアドレス再マッピングのための方法およびシステム

(57) 【要約】

マルチバンクメモリにおけるストレージアドレス再マッピングのための方法およびシステムが開示される。この方法は、クラスタのブロックの中の論理アドレスを割り当てることと、論理アドレスをストレージアドレス空間内に再マッピングすることとを含み、論理アドレス空間内に分散しているホストデータの短いランはストレージアドレス空間においてメガブロックの中に連続的にマッピングされる。各バンクにおいて独立に、特定のバンク内で利用可能な新しいブロックの数が所望しきい値未満まで減ったとき、マルチバンクメモリの各バンクにおいてデータを受け取るために利用し得る新しいブロックを作るために、有効なデータおよび使用済データの両方を有するブロックからそれぞれのバンクの中で有効なデータがフラッシングされる。

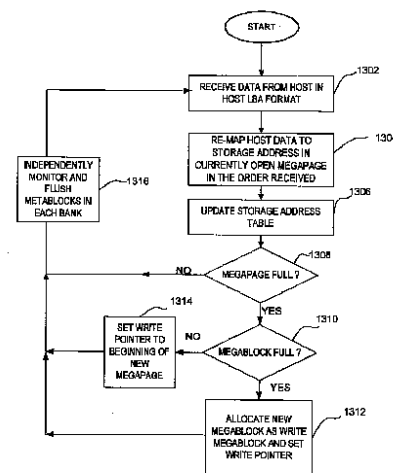


Fig. 13

【特許請求の範囲】**【請求項 1】**

ホストシステムと、一緒に消去可能なメモリセルのブロックとして各々構成されているメモリセルの複数のバンクを有する再プログラム可能な不揮発性大容量記憶システムとの間でデータを転送する方法であって、

前記ホストシステムにより割り当てられたホスト論理ブロックアドレス（LBA）アドレスと関連付けられたデータを受け取るステップと、

前記ホストLBAアドレスと関連付けられたデータをアドレス指定するために連続的ストレージLBAアドレスのメガブロックを割り当てるステップであって、連続的ストレージLBAアドレスの前記メガブロックは、メモリセルの複数のバンクの各々の中のメモリセルの少なくとも1つのブロックを含むと共に割り当て時に未書き込み容量だけをアドレス指定する、割り当てるステップと、

受け取られたデータのための前記ホストLBAアドレスの各々を連続的ストレージLBAアドレスのメガブロックに再マッピングするステップであって、各ストレージLBAアドレスは、前記ホストLBAアドレスに関わらず受け取られたデータが受け取られた順に受け取られたデータに連続的に順次割り当てられる、再マッピングするステップと、

複数のバンクのうちの第1のバンクの中のブロックを複数のバンクのうちの第2のバンクの中のブロックをフラッシングすることとは独立にフラッシングするステップであって、前記第1のバンクの中のブロックをフラッシングすることは、有効なデータのためのホストLBAアドレスを前記第1のバンクの中のブロックのストレージLBAアドレスから第1の再配置ブロックの中の連続的ストレージLBAアドレスへ再割り当てすることを含み、前記第2のバンクの中のブロックをフラッシングすることは、有効なデータのためのホストLBAアドレスを前記第2のバンクの中のブロックのストレージLBAアドレスから第2の再配置ブロックの中の連続的ストレージLBAアドレスへ再割り当てすることを含む、フラッシングするステップと、

を含む方法。

【請求項 2】

請求項1記載の方法において、

前記第1のバンクの中のブロックをフラッシングすることは、有効なデータのためのホストLBAアドレスを前記第1のバンクの中のブロックのストレージLBAアドレスから前記第1のバンクの中の再配置ブロックだけへ再割り当てすることをさらに含み、前記第2のブロックをフラッシングすることは、有効なデータのためのホストLBAアドレスを前記第2のバンクの中のブロックのストレージLBAアドレスから前記第2のバンクの中の再配置ブロックだけへ再割り当てすることを含む方法。

【請求項 3】

請求項2記載の方法において、

前記第1のバンクの中の連続的ストレージLBAアドレスのブロックを新しい再配置ブロックとして割り当てるステップをさらに含み、連続的ストレージLBAアドレスの前記新しい再配置ブロックは割り当て時に未書き込み容量だけと関連付けられ、前記新しい再配置ブロックの割り当ては、前記第1のバンクにおいて再配置ブロックの中のストレージLBAアドレスを完全に割り当てたときにだけ行われる方法。

【請求項 4】

請求項1記載の方法において、

前記受け取られたデータのためのホストLBAアドレスの各々を連続的ストレージLBAアドレスのメガブロックへ再マッピングするステップは、ストレージLBAアドレスを前記メガブロックのためのメガページ順にホストLBAアドレスと関連付けることを含み、前記メガページは前記メガブロックの各ブロックの中のメタページを含む方法。

【請求項 5】

請求項1記載の方法において、

複数のバンクの各々のためにホストLBAアドレスのストレージLBAアドレスに対す

10

20

30

40

50

る関係を特定する相関情報を単一のストレージアドレステーブルに記録するステップをさらに含む方法。

【請求項 6】

請求項 5 記載の方法において、

前記相関情報は、有効なデータと関連付けられているホスト L B A アドレスのランと、ホスト L B A アドレスのランにマッピングされているストレージ L B A アドレスとだけを含む方法。

【請求項 7】

請求項 5 記載の方法において、

前記相関情報は、ホスト L B A アドレスのメガブロックの中の全てのホスト L B A アドレスのためのマッピング情報を含む方法。

10

【請求項 8】

請求項 5 記載の方法において、

前記単一のストレージアドレステーブルは少なくとも 1 つのストレージアドレステーブルブロックを含み、前のストレージアドレステーブル書き込みブロックが相関情報に完全に割り当てられたときに、割り当て時に未書き込み容量だけと関連付けられている新しいストレージアドレステーブル書き込みブロックを割り当てるステップをさらに含む方法。

【請求項 9】

請求項 8 記載の方法において、

前記前のストレージアドレステーブル書き込みブロックを包含するバンク以外のバンクにおいて前記新しいストレージアドレステーブル書き込みブロックを割り当てるステップをさらに含む方法。

20

【請求項 10】

ホストシステムと、一緒に消去可能なメモリセルのブロックとして各々構成されているメモリセルの複数のバンクを有する再プログラム可能な不揮発性大容量記憶システムとの間でデータを転送する方法であって、

受け取られたホストデータのためのホスト論理ブロックアドレス (L B A) アドレスをストレージ L B A アドレスのメガブロックに再マッピングするステップであって、ストレージ L B A アドレスの前記メガブロックはメモリセルの複数のバンクの各々の中のメモリセルの少なくとも 1 つのブロックを含み、受け取られたデータのためのホスト L B A アドレスは前記メガブロックの中でメガページ順にストレージ L B A アドレスに連続的に割り当てられ、各メガページは、前記ホスト L B A アドレスに関わらず受け取られたデータが受け取られた順に、前記メガブロックのブロックの各々の中のメタページを含む、再マッピングするステップと、

30

複数のバンクの各々においてフラッシング操作を独立に実行するステップであって、前記フラッシング操作は、有効なデータのためのホスト L B A アドレスを特定のバンクの中のブロックのストレージ L B A アドレスから前記特定のバンクの中の再配置ブロックの中の連続的ストレージ L B A アドレスへ再割り当てする、フラッシング操作を独立に実行するステップと、

を含む方法。

40

【請求項 11】

請求項 10 記載の方法において、

複数のバンクの各々の中のピンク色ブロックを特定するステップであって、各ピンク色ブロックは、有効なデータと使用済データとの両方と関連付けられているストレージ L B A アドレスの完全に書き込まれたブロックを含む、特定するステップと、

各バンクについて、前記バンクの中の特定されたピンク色ブロックのうちの 1 つを次のフラッシング操作のために独立に選択するステップと、

をさらに含む方法。

【請求項 12】

請求項 11 記載の方法において、

50

複数のバンクの各々においてブロック情報テーブルを維持するステップをさらに含み、バンクのための前記ブロック情報テーブルは、そのバンクの中のピンク色ブロックのリストを含む方法。

【請求項 13】

請求項 10 記載の方法において、

前記フラッシング操作を独立に実行するステップは、複数のバンクのうちの 1 つのバンクにおいては第 1 のしきい値に基づいて、複数のバンクのうちの第 2 のバンクにおいては第 2 のしきい値に基づいて、フラッシング操作を開始することを含む方法。

【請求項 14】

請求項 10 記載の方法において、

複数のバンクの各々のためにホスト L B A アドレスのストレージ L B A アドレスに対する関係を特定する関連情報を単一のストレージアドレステーブルに記録するステップをさらに含む方法。

【請求項 15】

請求項 14 記載の方法において、

前記関連情報は、有効なデータと関連付けられているホスト L B A アドレスのランと、ホスト L B A アドレスのランにマッピングされているストレージ L B A アドレスとだけを含む方法。

【請求項 16】

請求項 14 記載の方法において、

前記関連情報は、ホスト L B A アドレスのメガブロックの中の全てのホスト L B A アドレスのためのマッピング情報を含む方法。

【請求項 17】

請求項 14 記載の方法において、

前記単一のストレージアドレステーブルは少なくとも 1 つのストレージアドレステーブルブロックを含み、前のストレージアドレステーブル書き込みブロックが関連情報に完全に割り当てられたときに、割り当て時に未書き込み容量だけと関連付けられている新しいストレージアドレステーブル書き込みブロックを割り当てるステップをさらに含む方法。

【請求項 18】

請求項 17 記載の方法において、

前記前のストレージアドレステーブル書き込みブロックを包含するバンク以外のバンクにおいて前記新しいストレージアドレステーブル書き込みブロックを割り当てるステップをさらに含む方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般的にはオペレーティングシステムと記憶装置との間のデータ通信に関する。より具体的には、本発明は、マルチバンク再プログラム可能な不揮発性半導体フラッシュメモリなどのメモリシステムと、メモリが接続されているかあるいは接続可能であるホスト装置との操作に関する。

【背景技術】

【0002】

従来のフラッシュデータメモリシステムにデータを書き込むとき、ホストは、通例、メモリシステムの連続的な仮想アドレス空間の中のセクタ、クラスタまたは他のデータユニットに一意の論理アドレスを割り当てる。ホストは、メモリシステムの論理アドレス空間の中のアドレスにデータを書き込み、アドレスからデータを読み出す。そのとき、メモリシステムは、一般に、論理アドレス空間と、データが論理アドレス空間内の範囲に対応する固定された論理グループを構成して格納されるメモリの物理的ブロックまたはメタブロックとの間でデータをマッピングする。一般に、各々の固定された論理グループはメモリシステムの別々の物理的ブロックに格納される。メモリシステムは、論理アドレス空間が

10

20

30

40

50

物理メモリにどのようにマッピングされるかを追跡するけれども、ホストはこれを知らない。ホストは、論理アドレス空間の中のそれ自身のデータファイルのアドレスを追跡するけれども、メモリシステムはこのマッピングについての情報なしで動作する。

【 0 0 0 3 】

このように動作するメモリシステムの欠点は断片化である。例えば、NTFSファイルシステムにしたがって動作するパーソナルコンピュータ（PC）においてソリッドステートディスク（SSD）ドライブに書き込まれたデータは、ドライブの論理アドレス空間の中の広く分布する位置に存在する連続するアドレスの短いランのパターンにより特徴付けられることが多い。ホストにより使用されるファイルシステムが、連続するファイルについて新しいデータのために順次アドレスを割り当てたとしても、削除されるファイルの気まぐれなパターンに起因して、利用可能な空きメモリ空間が新しいファイルデータのためにブロック化ユニットをなして割り当てられ得ないような利用可能な空きメモリ空間の断片化が生じる。

10

【 0 0 0 4 】

フラッシュメモリマネジメントシステムは、連続する論理アドレスのブロックを物理アドレスのブロックにマッピングすることによって動作しがちである。ホストからのアドレスの短いランが独立して更新されるとき、そのランを含むアドレスの論理ブロック全体が、単一のブロックへのそれ自身の長期マッピングを維持しなければならない。これは論理対物理メモリマネジメントシステムの中でのガーベッジコレクション操作を必要とし、その操作で、論理ブロック内のホストにより更新されない全てのデータは、更新されるデータとそれとを統合するために再配置される。データが、マルチバンクシステムを構成する別々のフラッシュメモリバンクにおける格納されたブロックであり得るマルチバンクフラッシュメモリシステムにおいて、統合プロセスは大規模になることがある。これはかなりのオーバーヘッドであり、書き込み速度およびメモリ寿命を厳しく制限するかもしれない。

20

【 先行技術文献 】

【 特許文献 】

【 0 0 0 5 】

【 特許文献 1 】 米国特許第 5 , 5 7 0 , 3 1 5 号

【 特許文献 2 】 米国特許第 5 , 7 7 4 , 3 9 7 号

30

【 特許文献 3 】 米国特許第 6 , 0 4 6 , 9 3 5 号

【 特許文献 4 】 米国特許第 6 , 3 7 3 , 7 4 6 号

【 特許文献 5 】 米国特許第 6 , 4 5 6 , 5 2 8 号

【 特許文献 6 】 米国特許第 6 , 5 2 2 , 5 8 0 号

【 特許文献 7 】 米国特許第 6 , 7 7 1 , 5 3 6 号

【 特許文献 8 】 米国特許第 6 , 7 8 1 , 8 7 7 号

【 特許文献 9 】 米国公開特許出願第 2 0 0 3 / 0 1 4 7 2 7 8 号

【 特許文献 1 0 】 米国公開特許出願第 2 0 0 3 / 0 1 0 9 0 9 3 号

【 特許文献 1 1 】 米国特許出願第 1 2 / 0 3 6 , 0 1 4 号

【 発明の概要 】

40

【 0 0 0 6 】

マルチバンクメモリシステムにおける改良されたメモリ管理に対する需要に対処するために、本願明細書において方法が開示される。第 1 の実施形態にしたがって、ホストシステムと再プログラム可能な不揮発性大容量記憶システムとの間でデータを転送する方法が開示される。この方法は、ホストシステムにより割り当てられたホスト論理ブロックアドレス（LBA）アドレスと関連付けられたデータを受け取ることと、ホストLBAアドレスと関連付けられているデータをアドレス指定するための連続的ストレージLBAアドレスのメガブロックを割り当てることとを含み、連続的ストレージLBAアドレスのメガブロックは、大容量記憶システム内のメモリセルの複数のバンクの各々の中のメモリセルの少なくとも 1 つのブロックを含み、割り当て時に未書き込み容量だけをアドレス指定する

50

。受け取られたデータのためのホスト L B A アドレスの各々について、連続的ストレージ L B A アドレスのメガブロックへの再マッピングが行われ、各ストレージ L B A アドレスは、受け取られたデータに対して、ホスト L B A アドレスに関わらず、受け取られたデータが受け取られた順に連続的に順次に割り当てられる。さらに、複数のバンクのうちの第 1 のもののの中のブロックは複数のバンクのうちの第 2 のもののの中のブロックとは独立にフラッシングされ、第 1 のバンク内のブロックをフラッシングすることは、有効なデータのためのホスト L B A アドレスを第 1 のバンク内のブロックのストレージ L B A アドレスから第 1 の再配置ブロック内の連続的ストレージ L B A アドレスに再割り当てすることを含み、第 2 のバンク内のブロックをフラッシングすることは、有効なデータのためのホスト L B A アドレスを第 2 のバンク内のブロックのストレージ L B A アドレスから第 2 の再配置ブロック内の連続的ストレージ L B A アドレスに再割り当てすることを含む。

10

【 0 0 0 7 】

他の実施形態にしたがって、ホストシステムと再プログラム可能な不揮発性大容量記憶システムとの間でデータを転送する方法が提供され、大容量記憶システムはメモリセルの複数のバンクを有し、複数のバンクの各々は、一緒に消去可能なメモリセルのブロックとして構成されている。この方法は、受け取られたホストデータのためのホスト論理ブロックアドレス (L B A) アドレスをストレージ L B A アドレスのメガブロックに再マッピングすることを含み、ストレージ L B A アドレスのメガブロックはメモリセルの複数のバンクの各々の中にメモリセルの少なくとも 1 つのブロックを有する。受け取られたデータのためのホスト L B A アドレスは、ホスト L B A アドレスに関わらずデータが受信された順にメガブロック内でメガページ順にストレージ L B A アドレスに連続的に割り当てられ、各メガページはメガブロックのブロックの各々についてメタページを含む。この方法は、バンクの各々において独立にフラッシング操作を実行することを含む。フラッシング操作は、有効なデータのためのホスト L B A アドレスを特定のバンク内のブロックのストレージ L B A アドレスから特定のバンクの中の再配置ブロック内の連続的ストレージ L B A アドレスに再割り当てすることを含む。

20

本発明の他の特徴および利点は、次の図面、詳細な説明および特許請求の範囲を検討すれば明らかとなる。

【図面の簡単な説明】

【 0 0 0 8 】

30

【図 1】マルチバンク不揮発性メモリを有するメモリシステムと接続されたホストを示す。

【図 2】図 1 のマルチバンク不揮発性メモリに用いられるフラッシュメモリシステムコントローラの例のブロック図の例である。

【図 3】図 1 に示されているフラッシュメモリバンクのうちの 1 つとして適切な 1 つのフラッシュメモリバンクの例である。

【図 4】図 3 のメモリバンクに使用され得るメモリセルアレイの代表的な回路図である。

【図 5】図 3 のメモリバンクの物理的メモリ構成の例を示す。

【図 6】図 5 の物理的メモリの一部の拡大図を示す。

【図 7】図 1 のマルチバンクメモリ内の複数のバンクの物理的メモリ構成を示す。

40

【図 8】ホスト L B A アドレス空間における割り当て済みクラスタおよび空きクラスタの典型的なパターンを示す。

【図 9】1 つの開示された実施例に従うブロックによるクラスタの割り当てのパターンを示す。

【図 1 0】メモリシステムのメモリマネージャがストレージアドレス指定再マッピング機能を組み入れている場合のホストとメモリシステムとの間でのストレージアドレス再マッピングの実施例を示す。

【図 1 1】図 1 0 に示されているストレージアドレス再マッピングの代替の実施例を示す。

【図 1 2】機能がホストに置かれている場合のストレージアドレス再マッピングの実施例

50

を示す。

【図 1 3】図 1 0 ~ 1 2 のシステムに用いられるマルチバンク書き込みアルゴリズムの流れ図である。

【図 1 4】メモリシステムの個々のバンクの中でのクラスタのブロックの割り当ての状態図である。

【図 1 5】マルチバンクメモリシステムの各バンクに独立に加えられ得るフラッシング操作の流れ図である。

【図 1 6】メガブロックにおける D L B A ラン分布を示す。

【図 1 7】図 1 6 の D L B A ラン分布のためのメガブロック書き込みプロシージャおよびストレージアドレステーブル生成を示す。

【図 1 8】図 1 6 のメガブロック内のブロックがフラッシングされた後の D L B A ランの再配列例を示す。

【図 1 9】マルチバンクメモリにおける 1 つのバンクの D L B A アドレス空間におけるフラッシング操作と、そのバンクのための物理アドレス空間における対応する更新ブロックとを示す。

【図 2 0】図 1 9 のバンクの D L B A 空間における第 2 のフラッシング操作を示す。

【図 2 1】フラッシング操作のためのピンク色ブロック選択プロセスの流れ図である。

【図 2 2】ホスト論理アドレスが第 2 の論理アドレス空間に再マッピングされる構成におけるストレージアドレステーブル (S A T) 階層を示す。

【図 2 3】論理対論理マッピングを追跡するのに使用されるストレージアドレステーブル (S A T) 書き込みブロックを示す。

【図 2 4】図 2 3 の S A T テーブルの S A T ページに用いられる L B A エントリである。

【図 2 5】図 2 3 の S A T テーブルの S A T ページに用いられる D L B A エントリである。

。

【図 2 6】図 2 3 の S A T テーブルの S A T ページに用いられる S A T 索引エントリである。

【図 2 7】図 1 1 および 1 2 のストレージアドレス再マッピングの実施例に用いられるストレージアドレステーブル変換プロシージャを示す。

【図 2 8】S A T ブロック遷移の状態図を示す。

【図 2 9】S A T ブロックのフラッシング順序を決定するためのプロセスの流れ図である。

。

【図 3 0】ブロック情報テーブル (B I T) 書き込みブロックを示す。

【図 3 1】メガブロックにおける D L B A ラン分布を示す。

【図 3 2】論理アドレスの完全なメガブロックが D L B A ランにマッピングされる S A T の実施形態を示す。

【図 3 3】L B A アドレスのためのアドレスフォーマットの例を示す。

【発明を実施するための形態】

【 0 0 0 9 】

本発明の諸態様を実施するために用いるのに適するフラッシュメモリシステムが図 1 ~ 7 に示されている。図 1 のホストシステム 1 0 0 は、メモリシステム 1 0 2 にデータを格納し、メモリシステムからデータを取り出す。メモリシステムは、パーソナルコンピュータにインストールされたソリッドステートディスク (S S D) ドライブの形などの、ホストに組み込まれたフラッシュメモリであり得る。代わりに、メモリシステム 1 0 2 は、図 1 に示されている機械的電氣的コネクタの接合部品 1 0 3 および 1 0 4 を通してホストに取り外し可能に接続されるカードの形であり得る。内部のあるいは組み込みの S S D ドライブとして用いられるべく構成されたフラッシュメモリは図 1 の略図に類似して見えるかもしれないが、主な違いは、ホスト内のメモリシステム 1 0 2 の位置である。S S D ドライブは、回転式磁気ディスクドライブのドロップイン代替品 (完全互換品) である別々のモジュールの形であり得る。

【 0 0 1 0 】

10

20

30

40

50

商業的に入手可能なSSDドライブの一例は、サンディスク コーポレーションにより製造されている32ギガバイトSSDである。商業的に入手可能な取り外し可能のフラッシュメモリカードの例は、コンパクトフラッシュ(CF)、マルチメディアカード(MMC)、セキュアデジタル(SD)、ミニSD、メモリスティック、スマートメディアおよびトランスフラッシュカードを含む。これらのカードの各々は、その標準化された仕様に従う独特の機械的および/または電氣的インターフェイスを有するけれども、各々に含まれるフラッシュメモリシステムは類似している。これらのカードは、全て、本願の譲受人であるサンディスク コーポレーションから入手可能である。サンディスク コーポレーションは、そのCruzer(クルーザー)という登録商標のもとで一系列のフラッシュドライブも提供し、それらは、ホストのUSBレセプタクルに差し込むことによってホストと接続されるユニバーサルシリアルバス(USB)プラグを有する小型パッケージ内のハンドヘルドのメモリシステムである。これらのメモリカードおよびフラッシュドライブの各々は、ホストとインターフェイスしてそれらの中のフラッシュメモリの動作を制御するコントローラを含む。

【0011】

SSD、メモリカードおよびフラッシュドライブを使用することのできるホストシステムは多数あって多様である。それらは、デスクトップまたはラップトップおよび他の携帯用コンピュータなどのパーソナルコンピュータ(PC)、セルラー電話、個人用携帯情報端末(PDA)、デジタルスチルカメラ、デジタルムービーカメラおよび携帯用オーディオプレーヤを含む。携帯用メモリカードアプリケーションのために、ホストは1つ以上のタイプのメモリカードまたはフラッシュドライブのための内蔵型レセプタクルを含むこともあるし、あるいはホストはメモリカードが差し込まれるアダプタを必要とすることもある。メモリシステムは普通それ自身のメモリコントローラおよびドライバを含むが、代わりに、メモリが接続されたホストにより実行されるソフトウェアによって制御されるメモリオンリーのシステムもある。コントローラを含むあるメモリシステム、特にホストに組み込まれるものでは、メモリ、コントローラおよびドライバはしばしば単一の集積回路チップ上に形成される。

【0012】

図1のホストシステム100は、メモリ102に関する限り、回路とソフトウェアとの組み合わせから成る2つの主要部分を有すると見られ得る。それらは、アプリケーション部分105と、メモリ102とインターフェイスするドライバ部分106とである。例えばPCでは、アプリケーション部分105は、ワードプロセッシング、グラフィクス、制御または他のポピュラーなアプリケーションソフトウェアを実行するプロセッサ109と、ホスト100上のデータを管理するためのファイルシステム110とを含むことができる。カメラ、セルラー電話あるいは他の、単一セットの機能を実行するために主として専用されるホストシステムでは、アプリケーション部分105は、写真を撮って記憶するようにカメラを操作したり、電話をかけた受けたりするようにセルラー電話を操作したりなどするソフトウェアを含む。

【0013】

図1のメモリシステム102は、マルチバンクフラッシュメモリ107などの不揮発性メモリと、メモリシステム102が接続されているホスト100とデータをやり取りするためにインターフェイスすると共にメモリ107を制御するコントローラ回路108とを含むことができる。コントローラ108は、データプログラミングおよび読み出し中に、ホスト100により使用されるデータの論理アドレスとマルチバンクフラッシュメモリ107の物理アドレスとの間の変換を行うことができる。マルチバンクフラッシュメモリ107は任意の数のメモリバンクを含むことができ、ここでは4つのメモリバンク107A~107Dが単に実例として示されている。

【0014】

図2を参照すると、システムコントローラ108は、特定用途向け集積回路(ASIC)などの単一の集積回路チップ上に実装され得る。コントローラ108のプロセッサ20

10

20

30

40

50

6 は、マルチバンクフラッシュメモリ 107 内のそれぞれのバンク 107A ~ 107D の各々のための I/O ポートを有するメモリアンターフェイス 204 を介してそれぞれのメモリバンク 107A ~ 107D の各々と別々に通信することのできるマルチスレッドプロセッサとして構成され得る。コントローラ 108 は内部クロック 218 を含むことができる。プロセッサ 206 は、内部データバス 202 を介して誤り訂正符号 (ECC) モジュール 214、RAM バッファ 212、ホストインターフェイス 216、およびブートコード ROM 210 と通信する。

【0015】

図 3 の単一バンク 107A の図を参照すると、マルチバンクフラッシュメモリ 107 内の各バンクは 1 つ以上の集積回路チップから成ることができ、各チップは複数のサブアレイまたはプレーンに構成されたメモリセルのアレイを含むことができる。2 つのそのようなプレーン 310 および 312 が簡略化して図に示されているけれども、その代わりに 4 個または 8 個などのより多くのそのようなプレーンを使用してもよい。あるいは、メモリバンクのメモリセルアレイはプレーンに分割されないかもしれない。しかし、そのように分割されるならば、各プレーンは、互いに独立に動作し得るそれ自身の列制御回路 314 および 316 を有する。回路 314 および 316 は、それらのそれぞれのメモリセルアレイのアドレスをシステムバス 302 のアドレス部分 306 から受け取って、それぞれのビット線 318 および 320 のうちの特定の 1 つ以上をアドレス指定するためにそれらを復号する。ワード線 322 は、アドレスバス 19 で受け取られるアドレスに応答して行制御回路 324 を通してアドレス指定される。ソース電圧制御回路 326 および 328 も、p ウェル電圧制御回路 330 および 332 と同じく、それぞれのプレーンと接続されている。バンク 107A がメモリセルの単一のアレイを有するメモリチップの形であるならば、そして 2 つ以上のそのようなチップがシステム内に存在するならば、各チップのアレイは前述したマルチプレーンチップの中のプレーンまたはサブアレイと同様に操作され得る。各バンク 107A ~ 107D は、諸機能がコントローラ 108 によって独立に同時にあるいは非同期に制御され得るように構成される。例えば、第 1 のバンクは、第 2 のバンクがデータを読み出している間、データを書き込むように命令され得る。

【0016】

データは、システムバス 302 のデータ部分 304 と接続されているそれぞれのデータ入出力回路 334 および 336 を通してプレーン 310 および 312 内へ、およびプレーン外へ転送される。回路 334 および 336 は、それぞれの列制御回路 314 および 316 を通してプレーンに接続されているライン 338 および 340 を通して、データをメモリセル内にプログラムすると共にそれらのそれぞれのプレーンのメモリセルからデータを読み出すことに配慮している。

【0017】

コントローラ 108 内のプロセッサ 206 は、データをプログラムし、データを読み出し、消去し、さらに種々のハウスキーピング事項を処理するために各バンク 107A ~ 107D 内のメモリチップの動作を制御するけれども、各メモリチップも、そのような機能を実行するためのコントローラ 108 からのコマンドを実行するある制御回路を含む。インターフェイス回路 342 は、システムバス 302 の制御およびステータス部分 308 に接続されている。コントローラ 108 からのコマンドは状態マシン 344 に提供され、これはその後、これらのコマンドを実行するために他の回路の具体的な制御を提供する。制御線 346 ~ 354 は、状態マシン 344 を、図 3 に示されているようにこれらの他の回路に接続する。状態マシン 344 からのステータス情報は、バス部分 308 を介してコントローラ 108 へ送信されるべく線 356 を介してインターフェイス 342 に伝えられる。

【0018】

以下でメモリセルアレイ 310 および 312 の NAND アーキテクチャが論じられるけれども、NOR などの他のアーキテクチャが代わりに使用されてもよい。米国特許第 5,570,315 号 (特許文献 1)、第 5,774,397 号 (特許文献 2)、第 6,04

10

20

30

40

50

6, 935号(特許文献3)、第6, 373, 746号(特許文献4)、第6, 456, 528号(特許文献5)、第6, 522, 580号(特許文献6)、第6, 771, 536号(特許文献7)および第6, 781, 877号(特許文献8)ならびに米国公開特許出願第2003/0147278号(特許文献9)を参照することによってNANDフラッシュメモリおよびメモリシステムの一部としてのそれらの動作の例が得られる。一例であるNANDアレイが図4の回路図により図に示され、これは、図3のメモリシステムのメモリセルアレイ310の一部である。多数のグローバルビット線が設けられ、説明を簡単にするためにそのような線402~408が4つだけ図4に示されている。これらのビット線のうちの1つと基準電位との間に、直列接続されているメモリセルストリング410~424がいくつか接続されている。メモリセルストリング414を代表として用いると、複数の電荷蓄積メモリセル426~432が、ストリングの両端の選択トランジスタ434および436と直列に接続されている。ストリングの選択トランジスタが導通させられると、ストリングはそのビット線と基準電位との間に接続される。その後、そのストリングの中のメモリセルが一度に1つずつプログラムまたは読み出しされる。

【0019】

図4のワード線438~444はメモリセルのいくつかのストリングのうちの各々のうちの1つのメモリセルの電荷蓄積素子を横断してそれぞれ延び、ゲート446および450はストリングの各端の選択トランジスタの状態を制御する。共通のワード線およびコントロールゲート線438~450を共有するメモリセルストリングは、一緒に消去されるメモリセルのブロック452を形成させられる。セルのこのブロックは、一度に物理的に消去可能な最小数のセルを含む。ワード線438~444のうちの1つに沿う1行のメモリセルが一度にプログラムされる。通例、NANDアレイの行は、この場合には接地または他の共通電位に接続されているストリングの端に最も近いワード線444に沿う行から始まる、所定の順序でプログラムされる。ワード線442に沿うメモリセルの行が次にプログラムされるなどして、ブロック452全体にわたってプログラムが行われる。ワード線438に沿う行は最後にプログラムされる。

【0020】

第2のブロック454は同様であって、そのメモリセルのストリングは、第1のブロック452のストリングと同じグローバルビット線に接続されているけれどもワード線およびコントロールゲート線の異なるセットを有する。ワード線およびコントロールゲート線は、行制御回路324によってそれらの適切な動作電圧まで駆動される。図3のプレーン1および2など、システムに2つ以上のプレーンまたはサブアレイが存在すれば、1つのメモリアーキテクチャは、それらの間で延びる共通のワード線を使用する。代わりに、共通のワード線を共有する3つ以上のプレーンまたはサブアレイがあり得る。他のメモリアーキテクチャでは、個々のプレーンまたはサブアレイのワード線は別々に駆動される。

【0021】

前に援用されているNAND特許および公開された特許出願のうちのいくつかに記載されているように、メモリシステムは、各々の電荷蓄積素子または領域に2つより多い検出可能な電荷のレベルを記憶させ、それによって各々に1ビットより多いデータを記憶させるように操作され得る。メモリセルの電荷蓄積素子は、非常に一般的には伝導性フローティングゲートであるけれども、その代わりに、米国公開特許出願第2003/0109093号(特許文献10)に記載されているように、非伝導性誘電体電荷トラッピング材料であってもよい。

【0022】

図5は、以下のさらなる記述において例として用いられるマルチバンクフラッシュメモリ107(図1)の1つのバンク107Aの構成を概念的に示す。メモリセルの4つのプレーンまたはサブアレイ502~508は、単一の統合メモリセルチップ上に、2つのチップ上に(各チップ上にプレーンのうちの2つ)あるいは4つの別々のチップ上に存在し得る。具体的な構成は、以下の論議には重要でない。もちろん、1、2、8、16あるいはそれ以上などの他の数のプレーンがシステム内に存在し得る。プレーンは、それぞれの

10

20

30

40

50

プレーン 5 0 2 ~ 5 0 8 内に位置するブロック 5 1 0、5 1 2、5 1 4 および 5 1 6 などの長方形により図 5 に示されているメモリセルのブロックにそれぞれ分割されている。各プレーン内に数十または数百のブロックが存在し得る。

【 0 0 2 3 】

前に言及されたように、メモリセルのブロックは、消去の単位、物理的に一緒に消去可能な最小数のメモリセルである。しかし、並列性を高めるために、ブロックはより大きなメタブロック単位で操作される。各プレーンからの 1 ブロックが 1 つのメタブロックを形成するように互いに論理的にリンクされる。4 つのブロック 5 1 0 ~ 5 1 6 は 1 つのメタブロック 5 1 8 を形成して示されている。メタブロック内のセルの全てが通例一緒に消去される。メタブロックを形成するために使用されるブロックは、ブロック 5 2 2 ~ 5 2 8 から成る第 2 のメタブロック 5 2 0 において示されているように、それぞれのプレーンの中の同じ相対位置に限定されなくてもよい。全プレーンにわたってメタブロックを広げるのが普通は好ましいけれども、システム性能を高めるために、メモリシステムは異なるプレーン内の 1、2 または 3 ブロックのいずれかまたは全部からメタブロックを動的に形成する能力を用いて操作され得る。これにより、メタブロックのサイズを、1 プログラミング操作で記憶させるのに使用できるデータの量により厳密に調和させることが可能となる。

【 0 0 2 4 】

一方、個々のブロックは、操作の目的のために、図 6 に示されているようにメモリセルのページに分割される。例えば、ブロック 5 1 0 ~ 5 1 6 の各々のメモリセルは、各々 8 ページ P 0 ~ P 7 に分割されている。代わりに、各ブロック内にメモリセルの 1 6、3 2 あるいはそれ以上のページがあってもよい。ページは、ブロックの中でのデータのプログラミングおよび読み出しの単位であって、一度にプログラムされるかまたは読み出される最少量のデータを含む。図 3 の N A N D アーキテクチャでは、1 ページは 1 ブロック内の 1 ワード線に沿うメモリセルから形成される。しかし、メモリシステムの動作並列性を高めるために、2 以上のブロックの中のそのようなページが論理的にリンクされてメタページにされ得る。メタページ 6 0 2 が図 6 に示され、4 つのブロック 5 1 0 ~ 5 1 6 の各々からの 1 物理ページから形成されている。例えば、メタページ 6 0 2 は 4 ブロックの各々のページ P 2 を含むけれども、メタページのページは必ずしも各ブロック内の同じ相対位置を持っていなくてもよい。バンク内で、メタページはプログラミングの最大単位である。

【 0 0 2 5 】

前述したように、図 5 ~ 6 は、マルチバンクメモリ 1 0 7 の 1 つのメモリバンク 1 0 7 A に存在し得るメモリセル構成の一実施形態を示す。一実施形態では、各バンク 1 0 7 A ~ 1 0 7 D のための個々のメモリセル構成に関わらず、メモリシステム 1 0 2 は好ましくはメガブロックというプログラミングの最大単位を有するように構成され、メガブロックは、メモリバンクが単一プレーン構成に整えられているならばマルチバンクメモリ内の各バンクの少なくとも 1 ブロックにまたがるか、あるいは、メモリバンクが複数プレーン構成に整えられているならばマルチバンクフラッシュメモリ 1 0 7 内の各バンクの 1 メタブロックにまたがる。以下の論議では、記述を明瞭にするために、各バンクはメタブロックの列として構成されていると仮定される。図 7 を参照すると、示されている各列は、前に論じられたメタブロック 5 1 8、5 2 0 などのメタブロック 7 0 2 のバンク 1 0 7 A ~ 1 0 7 D を表す。メガブロック 7 0 4 は各バンク 1 0 7 A ~ 1 0 7 D 内の少なくとも 1 つのメタブロック 7 0 2 を含み、各メタブロック 7 0 2 は複数のメタページ 7 0 6 に分割されている。図 7 において特定されているメガブロック 7 0 4 は各バンク 1 0 7 A ~ 1 0 7 D 内の同じ相対的物理的位置にメタブロック 7 0 2 を示しているけれども、メガブロック 7 0 4 を形成するために使用されるメタブロック 7 0 2 は同じ相対的物理的位置に限定されなくてもよい。さらに、本願明細書で言及されるとき、メガページ 7 0 8 は、メガブロック 7 0 4 内のメタブロック 7 0 2 の各々からのメタページ 7 0 6 を指す。メモリバンク 1 0 7 A ~ 1 0 7 D は、それぞれ同様に整えられてもよいし、あるいは互いに異なるメモリ

セル配列を持ってもよい。例えば、バンクは、バイナリ（単層セル、すなわち S L C ）フラッシュの第 1 のバンクと多層セル（ M L C ）フラッシュの他のバンクを持つなどして、種々のメモリ技術を使用することができる。さらに別の実施形態では、通常バンクブロックで必要となるデータの移動を伴わずにメガページの属性を更新できるように第 1 のバンクを再書き込み可能な不揮発性フラッシュとして製造することができて、残りのバンクは標準的フラッシュ（例えば、バイナリフラッシュまたは多層セルフラッシュ）を使用することができる。

【 0 0 2 6 】

次に、図 8 を参照すると、ホスト 1 0 0 とメモリシステム 1 0 2 との間の共通論理インターフェイスは、メモリシステム 1 0 2 に格納され得るデータ全てのためにアドレスを提供するために十分に大きい連続的論理アドレス空間 8 0 0 を利用する。前述したホスト 1 0 0 およびメモリシステム 1 0 2 を参照すると、マルチバンクフラッシュメモリ 1 0 7 に格納されるべきデータは、通例、ホスト論理ブロックアドレス（ L B A ）フォーマットで受け取られる。このホストアドレス空間 8 0 0 は、通例、データのクラスタのインクリメントに分割されている。各クラスタはデータの数個のセクタを包含するように所与のホストシステムにおいて設計することができ、およそ 4 ～ 6 4 セクタが一般的である。標準的セクタは 5 1 2 バイトのデータを包含する。図 8 を参照すると、 N T F S ファイルシステムのための論理アドレス空間 8 0 0 における割り当て済みクラスタ（陰影付き） 8 0 2 と空きクラスタ（陰影なし） 8 0 4 との代表的なパターンが示されている。

10

【 0 0 2 7 】

図 8 に見られる論理アドレス空間 8 0 0 の断片化を処理するための組織的構造が図 9 に示されている。本願明細書に記載されているストレージアドレス再マッピングのためのシステムおよび方法は、以下の論議において一般的に「ブロック」と称されるクラスタ 9 0 0 のメタブロックという項において L B A アドレスを割り当てる。以下の記述において、有効なデータで完全に満たされているブロック 9 0 0 は赤色ブロック 9 0 2 と称され、有効なデータを持っていないで、したがって未書き込み容量だけを包含するブロックは白色ブロック 9 0 4 と称される。白色ブロック 9 0 4 内の未書き込み容量は、メモリシステム 1 0 2 が「使用後消去」型のプロシージャを採用しているならば、消去済み状態であり得る。あるいは、メモリシステム 1 0 2 が「使用前消去」型のプロシージャを採用しているならば、白色ブロック 9 0 4 内の未書き込み容量は、割り当て時に消去されなければならない使用済みデータから成り得る。完全にプログラムされていてデータの有効なクラスタ 8 0 2 とデータの無効な（使用済みとも言われる）クラスタ 8 0 4 との両方を有するブロックはピンク色ブロック 9 0 6 と称される。本願明細書においてより詳細に論じられるように、各バンク 1 0 7 A ～ 1 0 7 D 内の少なくとも 1 つの白色ブロック 9 0 4 から構成されるメガブロック 7 0 4 は、ホストからデータを受け取るために割り当てられ、書き込みメガブロックと称される。

20

30

【 0 0 2 8 】

以下で記述されるマルチバンク書き込みアルゴリズムおよびフラッシング技術の実施例は、ホスト 1 0 0 およびメモリシステム 1 0 2 の構成に応じて変化し得る。図 1 0 ～ 1 2 は、ホストおよびメモリシステムの間での再マッピング機能のいくつかの構成を示す。図 1 0 ～ 1 1 の構成は、ストレージアドレス再マッピング（ S T A R : storage address re-mapping ）機能が完全にメモリシステム 1 0 0 4 、 1 1 0 2 に包含される実施形態を表す。これら始めの 2 つの構成では、メモリシステム 1 0 0 4 、 1 1 0 2 は、ホスト 1 0 0 2 での改変を必要とせずにレガシーホスト 1 0 0 2 と協働することができる。逆に、図 1 2 に示されている構成は、ストレージアドレス再マッピング機能が完全にホスト 1 2 0 2 に包含される実施形態のものである。この後者の実施形態では、ホスト 1 2 0 2 は、改変を必要としないレガシーストレージ装置 1 2 0 4 と協働することができる。 S T A R 書き込み機能の図 1 0 ～ 1 2 の各構成において実施例がさまざまであるのに加えて、以下でより詳しく記述されるフラッシング操作も様々である。単一バンクメモリにおける書き込みおよびフラッシングのためのフラッシュブロック管理方式の例が、その全体が本願明細書に

40

50

において参照により援用されている 2008 年 2 月 22 日に出願された同時係属中の米国特許出願第 12 / 036 , 014 号 (特許文献 11) に記載されている。

【 0029 】

図 10 の例では、ストレージアドレスマッピングアルゴリズムはストレージ装置 1004 の各バンクのメモリ管理 1006 に統合されることができ、物理メモリの第 1 のメガブロックが次のメガブロックに進む前にデータで完全に満たされるようにホスト 1002 からの LBA アドレスはマルチバンクフラッシュメモリ内の物理的ブロックに直接マッピングされる。その代わりに、図 11 では、ストレージアドレス再マッピングメカニズムは、ストレージ装置 1102 上の、装置 1102 の各バンクのためのメモリマネージャ 1104 とは別のアプリケーションにおいて、実現され得る。図 11 の実施例では、ホスト 1002 からの各論理アドレスは、ホストからのデータを完全なメガブロックという項に書き込む技術を利用して、本願明細書においてストレージ論理ブロックアドレス (ストレージ LBA) と称され、さらに本願明細書において装置論理ブロックアドレス (DLBA : device logical block address) とも称される第 2 の論理アドレスに再マッピングされ、その後、メモリマネージャ 1104 は、DLBA 配列のもとで組織されたデータをそれぞれのバンクのための物理メモリのブロックに移す。DLBA アドレス空間は、物理的メガブロックのものに等しい均一なサイズの DLBA ブロックとして構造化される。

10

【 0030 】

図 12 の実施例は、ストレージアドレス再マッピングの機能をストレージ装置 1204 からホスト 1202 上のアプリケーションに移す。この実施例では、LBA アドレスを DLBA アドレスにマッピングする機能は図 11 のものと類似し、主な違いは、変換が記憶装置 1204 ではなくてホスト 1202 上で行われることである。その後、ホスト 1202 は、ホストにおいて生成された DLBA アドレス情報と、DLBA アドレスに関連付けられたデータとの両方を記憶装置 1204 に送る。図 12 の実施例のために論理アドレス空間 800 を論理アドレスのブロックという項に分割して管理するために、ホストとメモリシステムとはフラッシュメモリ内の物理的ブロックのブロックサイズに関する情報を交換しなければならないかもしれない。論理ブロックのサイズは好ましくは物理的ブロックと同じサイズであり、この情報は、メモリシステムがホストと接続されるときに伝えられ得る。この通信は、パワーアップ時あるいはメモリシステムのホストへの接続時にハンドシェイク動作として行われるようにセットアップされてもよい。一実施形態では、ホストは、ブロックサイズおよびアライメント情報を要求して「識別ドライブ」クエリをメモリシステムに送ることができ、ブロックサイズは、その特定のメモリシステムのための個々の物理的ブロックのサイズであり、アライメント情報は、各物理的ブロックの一部を既に占めているかもしれないシステムデータのために物理的ブロックの先頭から、あるならば、どれだけのオフセットが考慮されなければならないかである。

20

30

【 0031 】

識別ドライブコマンドは、レガシー LBA インターフェイスコマンドセットにおいて予備のコードとして実現され得る。コマンドは、標準的通信インターフェイスにおいて予備のあるいは未割り当てのコマンドコードを介してホストからメモリシステムへ送られ得る。適切なインターフェイスの例は、ソリッドステートディスクのための ATA インターフェイス、あるいは、例えば CF または SD メモリカードに用いられるものなどの ATA 関連インターフェイスを含む。メモリシステムがブロックサイズおよびオフセット情報の両方を提供できなければ、ホストはデフォルトのブロックサイズおよびオフセットを仮定することができる。メモリシステムが識別ドライブコマンドに対してオフセット情報なしでブロックサイズ情報だけで答えたならば、ホストはデフォルトのオフセットを仮定することができる。デフォルトのブロックサイズは、いくつかの標準的ブロックサイズのうちのどれであってもよく、好ましくは、現実の物理的ブロックサイズらしいものよりは大きくセットされる。各物理的ブロックが物理的ブロック内の第 1 のアドレスからホストからのデータを受け取り始めることができると仮定されるように、デフォルトのオフセットはゼロオフセットにセットされ得る。ホストが SSD などの所定の内部ドライブに結合されて

40

50

いるならば、記憶装置の能力は既に知られていて前もってプログラムされているかもしれないので、ブロックサイズおよびオフセットを判定するこのステップを行う必要はないかもしれない。しかし、内部ドライブも交換されることがあるので、ホストは、記憶装置能力を常に確かめるように構成され得る。取り外し可能なメモリシステムのためには、ホストは、識別ドライブコマンドまたは類似のメカニズムを通してブロックサイズおよびオフセットについて常にたずねることができる。

【0032】

マルチバンクメガブロック書き込みアルゴリズム

図13に示されている一実施形態にしたがって、マルチバンクメモリにおいてホストデータ書き込み動作を管理する方法は、ホストファイルシステム110から、図8に関して前述したホストLBAフォーマットでホストデータを受け取ることを含む(1302において)。ホストデータを受け取られると、データは、現在開いている書き込みメガブロック内の現在開いているメガページにホストデータをホストLBA順序に関わらずそれが受け取られた順に書き込むことによって、ストレージアドレスに再マッピングされる(1304において)。以下でより詳しく論じられるように、オリジナルのホストLBAアドレスの、マルチバンクメモリ107内の現在のアドレスへのマッピングを追跡するために、ホストデータがマルチバンクメモリ107内のメガブロックに書き込まれるとストレージアドレステーブル(SAT)が更新される(1306において)。各メガページ708は次のメガページへの書き込みの前に完全に書き込まれ、新しいメガブロック704は、好ましくは、現在の書き込みメガブロックが完全に書き込まれた後に初めてさらなるホストデータを受け取るために割り当てられる(1308、1310および1312において)。現在のメガブロック704内の次のメガページ708が利用可能であれば、その次のメガページ708の先頭に書き込みポインタがセットされ(1314において)、ホストデータはメガページの各メタページ内の連続するストレージアドレスに、1バンクずつ、受け取られた順に、再マッピングされ続ける。ホストデータ書き込みアルゴリズムがマルチバンクメモリシステム107全体に対してメガブロックレベルでメガページ順に実行されている間、メモリシステム102内のバンク107A~107Dの各々に対してフラッシングアルゴリズムが独立に適用される(1316において)。フラッシングアルゴリズムは、以下で詳しく説明されるように、各バンクの中に、ホストデータ書き込みのためまたは他の記憶ニーズのために、新しいメガブロックで使用する新しい白色ブロックを作る。前に単一の書き込みメガブロックについて論じられたけれども、バンク107A~107Dが適宜に分割されるならば複数の書き込みメガブロックが実現され得る。

【0033】

ストレージアドレス再マッピングアルゴリズムの一実施例に従う各バンク107A~107Dの中でのデータの流れとブロック状態変化のパターンとが図14に示されている。現在の書き込みブロック内の最後のページが有効なデータで満たされると、現在の書き込みブロックは赤色ブロックになり(ステップ1404において)、新しい書き込みブロックが次のメガブロック704の一部となるべく白色ブロックリストから割り当てられる(ステップ1404において)。現在の書き込みブロックが完全にプログラムされる前に現在の書き込みブロックの中のいくつかのページが使用済みとなったならば、現在の書き込みブロックは完全にプログラムされたときにピンク色ブロックへの直接遷移も行い得るということに留意するべきである。この遷移を、明瞭性を目的として、示されていないけれども、書き込みブロックからピンク色ブロックへの矢によって表すことができる。

【0034】

図14のデータフローの特定の例を再び参照すると、赤色ブロックの中の1つ以上のページが後にLBAランの削除によって使用済みとなると、その赤色ブロックはピンク色ブロックになる(ステップ1406において)。バンクにおいてより多くの白色ブロックが必要であることをストレージアドレス再マッピングアルゴリズムが検出すると、アルゴリズムは、そのピンク色ブロックが白色ブロックになるようにピンク色ブロックから有効なデータを移すために、他のバンクにおいてアクティブであるかもしれない他のフラッシン

グアルゴリズムから独立して、バンクの中でフラッシング操作を開始する（ステップ 1 4 0 8 において）。ピンク色ブロックをフラッシングするために、ピンク色ブロックの有効なデータは、再配置ブロックとして指定されている白色ブロックへ出現の順に順次再配置される（ステップ 1 4 1 0 において）。その再配置ブロックは、いったん満たされれば、赤色ブロックになる（ステップ 1 4 1 2 において）。前に書き込みブロックに関して言及したように、再配置ブロックは、完全にプログラムされるまでに既にその中の何ページかが使用済みとなっていたならば、ピンク色ブロックへの直接遷移を行うこともできる。この遷移を、明瞭性を目的として、示されていないけれども、図 1 4 において再配置ブロックからピンク色ブロックへの矢によって表すことができる。

【 0 0 3 5 】

前に言及したように、ホストデータをメモリシステム 1 0 2 に書き込むとき、図 1 3 のマルチバンク書き込みアルゴリズムは、メガブロックという項でアドレス空間を割り当ててメガブロック全体をメガページ順に満たす。したがって、図 1 4 は単一のバンクを説明しているので、ホストからのデータは任意の所与のバンクにおいて書き込みブロックで、そのバンクの書き込みブロック内の 1 つのメタページが満たされるまで受け取られ、その後、そのバンク内の書き込みブロックにおいて他のメタページが利用できるかもしれないけれども、次のメタページ量のホストデータはメガページ内、すなわちマルチバンクフラッシュメモリ 1 0 7 内の次のバンクの書き込みブロック内の、次のメタページに書き込まれるということが理解されるべきである。このように、メモリの 1 つのバンク内にある所与の書き込みブロックは、ホストが提供するホストデータの N メタページごとにホストデータの 1 メタページというパターンを受け取り、ここで N はマルチバンクフラッシュメモリ 1 0 7 内のバンクの数である。この系統的ホストデータ書き込みシーケンスとは対照的に、前に言及した S A T などの、メモリシステム 1 0 2 の中で生成された情報、あるいはバンク内に新しい白色ブロックを作るためにフラッシング操作の一部分として再配置されるピンク色ブロックからの有効なデータは、バンク内のそれぞれの個々の書き込みブロックに完全に書き込まれる。

【 0 0 3 6 】

マルチバンクフラッシング操作

ストレージアドレス再マッピングアルゴリズムのある実施形態は、有効なデータの、ピンク色ブロック 9 0 6 から再配置ポイントとして知られている特別の書き込みポイントへの、本願明細書においてフラッシングとも称される再配置を行うことによって、白色ブロック 9 0 4 の作成を管理する。ストレージアドレス空間が前述したように範囲あるいはファイルサイズによって細分化されるならば、ストレージアドレスの各範囲はそれ自身の再配置ブロックおよび関連する再配置ポイントを持つことができる。図 1 5 を参照すると、マルチバンクフラッシュメモリのためのフラッシング操作の実施形態は、各バンク 1 0 7 A ~ 1 0 7 D のために別々にかつ独立して、十分な数の白色ブロックがあるかどうかを追跡することを含む（1 5 0 2 において）。この判定は、バンク内に現在存在する白色ブロックの総数に基づいて行うこともできるし、あるいはバンク内で白色ブロックが消費されつつある速度に基づいて行うこともできる。十分な数の白色ブロックがあるならば、フラッシング操作は不要であってバンクは次の書き込み操作を待つことができる（1 5 0 4 において）。白色ブロックの数が不十分であると判定されたならば、以下のようにバンクのために維持されているピンク色ブロックリストからバンク内のピンク色ブロックが選択される（1 5 0 6 において）。バンク内の現在の再配置ブロックが満杯でなければ、有効なデータが、選択されたピンク色ブロックからピンク色ブロック内での出現の順に再配置ブロック内の連続する位置へコピーされる（1 5 0 8、1 5 1 0 において）。一実施形態では、再配置ブロックが完全にプログラムされているときにだけ、同じバンクから他の白色ブロックが次の再配置ブロックとして割り当てられる（1 5 1 2 において）。さらに、一実施形態では、選択されたピンク色ブロックからの有効なデータだけが、そのピンク色ブロックがコピーされていない有効なデータを依然として包含している間、再配置ブロックにコピーされる（1 5 1 4 において）。マルチバンクフラッシュメモリ 1 0 7 において、

10

20

30

40

50

1つの特定のバンク内のピンク色ブロック906の中の有効なデータは同じバンク内の再配置ブロックにフラッシングされるだけであるように、フラッシング操作はそれぞれのバンク107A~107Dの中で独立に実行され、かつそれぞれのバンク107A~107Dの中に完全に包含されるということを、図15に示されているフラッシング操作は表している。フラッシング操作は、通常、ピンク色ブロックを白色ブロックに変換するために、バックグラウンド操作として実行される。

【0037】

ピンク色ブロック906は、その特性に応じてフラッシング操作のために選択される。一実施形態では、ピンク色ブロックのリストはマルチバンクフラッシュメモリ107内の各バンク107A~107Dのために独立に維持される。図9を再び参照すると、一実施例では、有効なデータを有するアドレスが少なければ、その特定のピンク色ブロックがフラッシングされるときに再配置を必要とするデータは少ないので、最少量の有効なデータ（すなわち、図9において最小数の陰影付きクラスタ）を有するピンク色ブロックが選択される。したがって、図9の例では、ピンク色ブロックBのほうがより少数の有効なデータを伴うアドレスを有するので、ピンク色ブロックAではなくてピンク色ブロックBのほうが選択される。他の実施例では、フラッシング操作のために選択されるピンク色ブロックは、あるしきい値量より少ない有効なデータと関連付けられているピンク色ブロックのグループのうちのいずれであってもよい。そのしきい値は、ピンク色ブロックのセット全体に包含される有効なデータの平均量未満であってもよい。そのしきい値量あるいはそれ未満の有効なデータのピンク色ブロックのサブセットを、ホストまたはメモリシステムがピンク色ブロックを選択できるリストにおいて維持することができる。例えば、しきい値要件を現在満たしているピンク色ブロックのうちの所定の数（例えば、16）あるいはパーセンテージ（例えば、30%）のピンク色ブロックの動的リストを維持することができ、そのリスト中の選択されるピンク色ブロックが絶対最少量の有効なデータを有するかどうかを考慮せずにそのリストからフラッシングのためにどのピンク色ブロックでも選択され得る。各バンクにおけるリストを形成する、メモリシステムまたはホストが選択するピンク色ブロックの数またはパーセンテージは、固定された値あるいはユーザにより選択可能な値であり得る。リストは、使用可能なピンク色ブロックからの絶対最少量の有効なデータを有するピンク色ブロックをランク付けされた順に表すピンク色ブロックのグループを含むこともできるし、あるいは単にしきい値要件の範囲内にあるピンク色ブロックを含むこともできる。

【0038】

その代わりに、あるいは組み合わせられて、ピンク色ブロックの選択は、特定のピンク色ブロック906にさらなる使用済データを蓄積する計算された確率に基づいて行われてもよい。ピンク色ブロック906にさらなる使用済データが蓄積される確率は、メモリにおいて最も長く存在し続けているデータは最も削除されそうもないという仮定に基づいて行うことができる。再配置ブロックであったピンク色ブロック906は、新しいホストデータを有する書き込みブロックであったピンク色ブロック906より古くから存在し続けているデータを含み得る。フラッシングのためのピンク色ブロック906の選択プロセスは、始めに、近ごろ再配置ブロックであったピンク色ブロック906を標的にし得る。それは、それらがさらにデータを削除される見込みがより低く、したがってより少量の使用済データの追加が期待され得るからである。以前は書き込みブロックであったピンク色ブロック906は、より新しいデータが削除される見込みがより高く、したがってより多くの使用済データを作る見込みがより高いという仮定に基づいて後にフラッシングのために選択される。

【0039】

メガブロック書き込みプロセスのより具体的な例が図16~17に示されている。この例では、図11のシステム構成が使用されると仮定され、ホストLBAアドレスは、メモリシステム102内のコントローラ108により実行されるアプリケーションにおいてDLLBAアドレスとも称される中間ストレージLBAアドレスに変換される。図16に示さ

れているように、6個のメタページ(P1~P6)を各々有するメタブロック1602を有する4バンクメモリ内の開いている書き込みメガブロック1600は、図17に示されているLBAラン1702のためのLBAアドレスと関連付けられる。マルチバンクメモリ107への書き込みの順序は、第1の開いているメタページ(バンク2内のP2)から始まってメガページの残りに沿って左から右へ順に続く(バンク3内のP2、その次にバンク4内のP2)。コントローラは、LBAラン1702の入ってくるLBAアドレスが、それらが受け取られた順に、各メタページに関連付けられている連続するDLBAアドレスに再マッピングされて、次のメタページへ移る前にメタページ全体がプログラムされるように、LBAアドレスをメガページ内のそれぞれのメタページヘルディングする。LBAラン1702は、次のメガページと関連付けられているDLBAアドレスに再マッピングされ続ける(連続して、バンク1~4の各々のメタページP3)。その後、LBAラン1702の最後の部分は、バンク1およびバンク2内のメタページP4と関連付けられているDLBAアドレスに連続的に再マッピングされる。

10

20

30

40

50

【0040】

コントローラ108により管理される書き込みアルゴリズムは、メガブロック1600内の次のメガページへ進む前に1メガページ分のLBAアドレス指定されたホストデータをバンクの各々に順次分配することによってメガブロック1600に順に書き込むけれども、単一のラン1702のための各バンク内の不連続的LBAアドレスの集合は、各バンクによって、この例では図16~17においてDLBAランA1~A4として特定されるDLBAランとして管理される。LBAアドレスから各バンク内のDLBAアドレスへのマッピングは、メモリにおいて維持されるマルチバンクフラッシュメモリ107のためのストレージアドレステーブル(SAT)1704において追跡される。図17に示されているSAT1704のバージョンは、有効なデータを包含する各LBAランに関連するDLBAランにマッピングする。SAT1704内のLBAエントリ1706は、ラン内の第1のLBAアドレスと、ランの長さと、LBAラン1702にマッピングされた第1のDLBAラン(DLBAランA1)のDLBAアドレスおよびバンク識別子を含む。対応するDLBAエントリ1708は、DLBAランの第1のDLBAアドレスおよびバンク番号と、第1のDLBAアドレスがマッピングされるLBAラン1702内のLBAアドレスオフセットとを有する第1のDLBAエントリ1710を含み、それは、第1のDLBAエントリ1710の場合にはゼロであり、所与のLBAラン1702のためのその後の全てのDLBAエントリでは非ゼロ値である。

【0041】

LBAラン1702と関連付けられたデータがDLBAアドレスに再マッピングされて、DLBAアドレスと関連付けられているメガブロック1600内の物理アドレス位置に書き込まれた後、1つ以上のその後のLBAランが再マッピングされてメガブロック1600内の残りの未書き込み容量(バンク3および4内のP4と整列しているメガページの残りと、P5およびP6とそれぞれ整列しているメガページ)に書き込まれる。メガブロック1600などのメガブロックが完全にプログラムされた後、コントローラはメガブロックを最早追跡しなくて、メガブロック1600内の各ブロック1602~1608はその後それらのそれぞれのバンクにおいて実行される独立のフラッシング操作によって管理される。オリジナルのメガブロック1600のブロック1602~1608は、それぞれ使用済データの蓄積に起因してピンク色ブロックになると、関連していない再配置ブロックへ独立にフラッシングされ得る。図18は、DLBAランA1~A4がどのようにそれぞれのバンク内での独立のフラッシング操作によって新しいブロック1802~1808へ移され得るかを示す。DLBAランA1~A4と関連付けられているデータの残存は、当然、そのデータが有効なデータでブロック1600内の他のデータが使用済みでそれぞれのフラッシング操作を誘発したということを仮定している。さらに、ブロック1802~1808は、参照を容易にすると共に図16のメガブロック内でのDLBAランA1~A4のそれぞれの元の相対ページアライメントに関してのあり得る移動を説明するために、図18において、互いに隣接して示されているけれども、それぞれのフラッシング操作

後ブロック 1802 ~ 1808 は各バンクにおいて異なる物理的または相対的な位置に配置されそうである。

【0042】

メモリシステム上のコントローラ 108 またはホスト 100 上のプロセッサ 109 によって実行されるアプリケーションによって論理 - 論理、LBA - DLBA 変換が実行される図 11 および 12 に示されているストレージアドレス再マッピングの実施例を参照し、図 14 の状態図に従うアドレス操作の例が図 8 ~ 9 および 19 ~ 20 を参照して以下で論じられる。システムは図 15 によって表されるストレージアドレス再マッピングアルゴリズムにしたがって動作してきたと仮定すると、LBA アドレス空間 (図 8) では、空きクラスタ 804 は本質的にランダムな位置に分散されている。所与のバンクのための DLBA アドレス空間では (図 9)、2 つの白色ブロック 904 が使用可能であり、異なる数の使用済 (空き) クラスタ 804 を有する 3 つのピンク色ブロック 906 がある。

10

【0043】

ホストは、ストレージ装置に書き込むべきデータを次に得ると、使用可能な任意の場所の LBA アドレス空間を割り当てる。図 19 は、ストレージアドレス再マッピングアルゴリズムが、どのように図 9 の白色ブロック 904 などの使用可能な白色ブロックのうちの 1 つをより大きなメガブロックの一部である書き込みブロック 1904 とするべく割り当てるかを示すと共に、各 LBA アドレスが書き込みブロック 1904 内の使用可能な DLBA 空間内の連続的クラスタにどのようにマッピングされるかを示す。DLBA 空間内の書き込みブロック 1904 は、LBA アドレス位置に関わらずに、LBA アドレスが書き込まれた順に、前に論じられたメガブロック書き込みパターンにしたがって書き込まれる。バンクに適用されるストレージアドレス再マッピングアルゴリズムは、LBA アドレス番号順に関わらず、LBA アドレスが受け取られた時間順に書き込みブロック 1904 内の DLBA アドレスを割り当てる。データは、書き込みブロックにおいて 1 つ以上の DLBA ランをなして書き込まれる。DLBA ランは、同じ LBA ラン内の連続する LBA アドレスにマッピングされた連続する DLBA アドレスのセットである。DLBA ランは、DLBA アドレス空間 1902 においてブロック境界 (これはバンク境界である) で終わらなければならない。書き込みブロック 1904 が満杯になると、白色ブロック 904 が次の書き込みブロック 1904 として割り当てられる。

20

【0044】

各バンクにおいて、DLBA ブロックはフラッシュメモリ 107 の物理アドレス空間内のブロック 1906 で整列し、したがって DLBA ブロックサイズと物理アドレスブロックサイズとは同じである。さらに、DLBA 書き込みブロック 1904 におけるアドレスの配列は、物理アドレス空間における対応する更新ブロック 1906 の配列と同じである。この対応により、一般にガーベッジコレクションと称される別のデータ統合は物理的更新ブロックにおいて決して必要とされない。一般的なガーベッジコレクション操作では、論理アドレスのブロックは、論理ブロックにおいて LBA アドレスの特定の範囲を維持するために一般的に常に再組み立てされ、それは物理的ブロックにおいても反映される。より具体的には、一般的ガーベッジコレクション操作を利用するメモリシステムが特定の物理的ブロック内のセクタに対応する情報の更新されたセクタを受け取ると、メモリシステムは、その更新された 1 つまたは複数のセクタを受け取るために物理メモリ内の更新ブロックを割り当て、その後残っている有効なデータの全部を元の物理的ブロックから更新ブロックの残りの中へ統合する。このように、標準的ガーベッジコレクションは、特定の LBA アドレス範囲のためのデータのブロックを、その特定のアドレス範囲に対応するデータが常に共通の物理的ブロックの中に統合されるように、永続させる。本願明細書において論じられるフラッシング操作は、同じアドレス範囲内のデータの統合を必要としない。代わりに、フラッシング操作は、種々の物理的ブロックからのデータの集合であり得るデータの新しいブロックを作るためにアドレスマッピングを実行し、データの特定の LBA アドレス範囲が意図的に統合されることはない。

30

40

【0045】

50

前述したように、十分な供給量の白色ブロックが確実に利用可能であるように、ストレージアドレス再マッピングアルゴリズムは各バンク 107A ~ 107D において独立に動作する。ストレージアドレス再マッピングアルゴリズムは、データをピンク色ブロックから再配置ブロック 1908 として知られる特別の書き込みブロックへフラッシングすることによって白色ブロックの作成を管理する(図 19)。フラッシングされるべく現在選択されているピンク色ブロックはフラッシングブロックと称される。

【0046】

次に、図 19 ~ 20 を順に参照すると、所与のバンクのためのブロックフラッシングプロセスの図が示されている。図 11 の実施例において各バンク 107A ~ 107D のために独立にコントローラ 108 によって実行されるストレージアドレス再マッピングアルゴリズムは、白色ブロックを再配置ブロック 1908 として指定し、これに、さらなる白色ブロックを作成するために同じバンク内の選択されたピンク色ブロックからデータがフラッシングされる。図 19 に示されているように、フラッシングブロック(図 9 のピンク色ブロック A)内の、赤色データとも称される有効なデータは、フラッシングブロックを白色ブロック 904 に変換するために、再配置ブロック 1908 内の連続するアドレスへ再配置される。物理アドレス空間 1910 内の対応する更新ブロック 1906 も、フラッシングされるデータを受け取るために割り当てられる。ホストから受け取られる新しいデータのために使用される更新ブロック 1906 の場合と同じく、フラッシングされるデータを受け取るための更新ブロック 1906 は、フラッシング操作が DLBA アドレス空間 1902 における統合を既に成し遂げているので、有効なデータを統合するためのガーベジコレクション操作を決して必要としない。

【0047】

図 20 に示されているように、残りのピンク色ブロックから次のフラッシングブロック(図 19 のピンク色ブロック B)が特定される。最も少ない赤色データを有するピンク色ブロックが再びフラッシングブロックとして指定され、ピンク色ブロックの赤色データ(有効なデータ)は、開いている再配置ブロック内の連続する位置へ転送される。更新ブロック 1906 における物理アドレスの並行する割り当ても行われる。再び、再配置ブロック 1908 にマッピングされた物理的更新ブロック 1906 において、データ統合は必要とされない。ピンク色ブロックに対するフラッシング操作は、書き込みブロックとして指定されている白色ブロックの消費を補うのに十分な速度で白色ブロックを作るためにバックグラウンド操作として実行される。図 8 ~ 9 および 19 ~ 20 の例は、ホストからの新しいデータのためおよびピンク色ブロックから再配置されるデータのために、物理アドレス空間内のそれぞれの別々の更新ブロックと共に、書き込みブロックおよび再配置ブロックがどのように別々に維持され得るかを示している。現在のメガブロックが完全にプログラムされたときに初めてメガブロックの一部として動作する新しい書き込みブロックを割り当ててホストから受け取られた新しいデータを関連付けるプロセスと同様に、新しい再配置ブロックは、好ましくは、前の再配置ブロックが完全にプログラムされた後に初めて割り当てられる。新しい再配置ブロックは、好ましくは、割り当て時に、未書き込み容量を包含するだけである、すなわち、いつでも消去できる使用済データと関連付けられているだけであるか、あるいは既に消去されていて有効なデータを全く包含していないかである。

【0048】

前述した実施形態では、ホストからの新しいデータはホストからの他の新しいデータを受け取るだけの書き込みブロックと関連付けられ、フラッシング操作でピンク色ブロックからフラッシングされる有効なデータは、特定のバンク内の、そのバンクのための 1 つ以上のピンク色ブロックからの有効なデータを包含するだけの再配置ブロックの中へ移される。前述したように、他の実施形態でフラッシングされるピンク色ブロックの選択を行うことができ、その場合には、現在のピンク色ブロックについての平均量などのしきい値を下回る量の赤色データと関連付けられているピンク色ブロックのリストから任意のピンク色ブロックを選択することができ、あるいはそのピンク色ブロックは、使用可能なピンク

色ブロックのうちの（ピンク色ブロックと関連付けられている有効データの量に基づく）特定の順位を有するピンク色ブロックのうちのどれであってもよい。

【 0 0 4 9 】

フラッシング操作は「熱い」データが使用済みにされたブロックから割合に「冷たい」データを、同様に割合に「冷たい」データを包含する再配置ブロックへ再配置する。これは、割合に「熱い」ブロックおよび割合に「冷たい」ブロックの別々の集団を作るという効果を有する。フラッシングされるべきブロックは、常に、最少量のデータを包含する「熱い」ブロックとして選択される。「熱い」ブロックの集団を作れば、再配置されなければならないデータの量が減ることにより、メモリのストレス要因が減少する。

【 0 0 5 0 】

一実施形態では、フラッシングブロックとして選択されるピンク色ブロックは、内容が最もまばらなピンク色ブロック、すなわち、最少量の有効なデータを包含するピンク色ブロックであってよくて、ホストにより実行される特定の書き込みおよび消去操作に応答して選択されるのではない。このようにして行われるピンク色ブロックのフラッシングブロックとしての選択は、最小限の有効データ再配置でブロックフラッシング操作を行うことを可能にする。それは、そのように選択されるどのピンク色ブロックも、ホストによるファイルの削除に起因して最大限の数の未割り当てデータアドレスを蓄積しているからである。

【 0 0 5 1 】

ピンク色ブロック選択プロセスの一例は、最低数の有効ページまたはクラスタを有する 5 % のピンク色ブロックのうちの任意のピンク色ブロックを選択することである。バックグラウンド処理で、最低のページまたはクラスタ総数値を有する 16 個のピンク色ブロックのリストが作られる。ピンク色ブロック特定プロセスは、「P」個のスケジュールされたブロックフラッシング操作により費やされる時間のうちに 1 サイクルを完了することができる。フラッシングブロック特定プロセスにおける 1 サイクルが図 21 に示されている。白色、ピンクおよび他のタイプの D L B A アドレスブロックについてのブロックアドレスのリストを包含するブロック情報テーブル（B I T）が、以下でより詳しく記述されるように、各バンク 107 A ~ 107 B のためにストレージアドレス再マッピング機能によって別々に維持され、前のプロセスサイクルの間に特定されたブロックのセットの後に Q 個のピンク色ブロックの次のセットを特定するために読み出される（ステップ 2102 において）。各バンクのために独立に、装置初期化後の第 1 のプロセスサイクルでピンク色ブロックの第 1 のセットが特定されるべきである。フラッシングブロックの利用可能性を保証するために、Q の値は P のものより大であるべきである。一実施例では、Q の値は 8 で P は 4 であり得る。有効ページ総数値はセット中のピンク色ブロックの各々のためにゼロにセットされる（ステップ 2104 において）。セット中のどのピンクブロックの中に位置する有効データページをも特定するために、L B A および D L B A の関係を追跡するために維持されるストレージアドレステーブル（S A T）ページエントリが一度に 1 つずつスキャンされる（ステップ 2106 において）。ストレージアドレステーブルは以下でより詳しく記述される。有効ページ総数値は、それに応じてインクリメントされる。全ての S A T ページがスキャンされた後、セット中のピンク色ブロックの各々についての有効ページ総数値は、低有効ページ総数値を求めてリスト中のピンク色ブロックについてのものと対照して評価され、必要ならばリスト中のブロックはセットからのブロックにより取って代わられる（ステップ 2108 において）。ブロックフラッシング操作の完了後、次のブロックフラッシング操作のために次のブロックが選択されるべきである。これは、リスト中の最低有効ページ総数値を有するブロックであるべきである。

【 0 0 5 2 】

図 19 ~ 20 に関して記述したような、特定のバンク 107 A ~ 107 D におけるブロックフラッシング操作を開始する前に、再配置されなければならない有効な D L B A ランの位置を判定するために、選択されたブロックはマッピングされなければならない。これは、ブロックから読み出されるデータの選択されたページのヘッダ内の L B A アドレスと

10

20

30

40

50

、これらのDLBAアドレスについてのSATエン트리とを使用するサーチアルゴリズムによって成し遂げられる。サーチアルゴリズムは、それ自身が徐々に構築する既知の有効なおよび使用済みとなったDLBAランのマップを使用する。有効なDLBAランは、SATエントリがブロックにおけるその存在を明示していると、ブロックマップに加えられる。使用済みとなったDLBAランは、マッピングされるブロック内のデータページヘッダ内のDLBAの範囲についてのSATエントリが他のブロックにおける有効なDLBAの存在を明示しているとき、ブロックマップに加えられる。サーチプロセスは、ブロック内の全てのDLBAアドレスが有効または使用済として明白にマッピングされるまで続く。

【0053】

ブロックフラッシング操作では、前述したブロックマッピングプロセスで特定された有効なDLBAランの中の全てのページが、選択されたピンク色ブロックから同じバンク内の再配置ブロック内の再配置ポインタへ再配置される。再配置されたDLBAについてのエントリがSATリストにおいて記録される。有効なおよび使用済みとなったDLBAランのためのサーチは、図11に示されている構成の場合にはメモリシステム102のコントローラ108によって実行され、ブロックDLBAマップはコントローラと関連付けられたRAMに格納され得る。図12の構成では、ホストシステム100に存するCPU109がサーチを実行し、その結果として得られたブロックDLBA情報をホストシステムCPUと関連付けられているRAMに格納することができる。

【0054】

マルチバンクメモリ構成のためのストレージアドレス再マッピングアルゴリズムは、特定のバンク内の白色ブロックの数が所定しきい値未満にまで減ると、書き込みブロックにおけるホストデータの書き込みにより白色ブロック容量が消費されるのと同じ速度でデータ書き込みのために割り当てられ得る使用可能な白色ブロック容量が作られることを保証するのに十分な速度でそのバンクにおいてピンク色ブロックに対するフラッシング操作が行われなければならないという原理にしたがって動作する。ホストからのデータを書き込むことによって消費される書き込みブロック内のページの数、ブロックフラッシング操作によって回復される使用済みページの数と釣り合わされなければならない。ブロックフラッシング操作の完了後、前述したように、BITおよびSATから特定のエントリを読み出すことによって、次のブロックフラッシング操作のために選択されたピンク色ブロック内の使用済みデータのページの数、次のブロックフラッシング操作は、データのこの数の有効ページを書き込みブロックに書き込んだ直後に始まるようにスケジュールされ得る。さらに、フラッシング操作を開始するためのしきい値は各バンクのために異なっているてもよい。例えば、ある1つのバンク内のピンク色ブロックの有効データの平均量に基づいてしきい値がトリガされれば全てのバンクにおいてほぼ同じ速度で白色ブロックが作成され得るように、フラッシングのためのしきい値はバンク内で再配置されるべきデータの量に基づいて適応的であり得る。

【0055】

ストレージアドレステーブル

前述したストレージアドレス再マッピングを実行するために、図17に関して一般的に記述したものなどのストレージアドレステーブル(SAT)1704が、ストレージアドレス空間内のデータの位置を追跡するために使用される。SAT内の情報は、順次更新の一部として完全なフラッシュメタブロックにも書き込まれる。したがって、一実施例では、SAT情報は、ホストから受け取られるデータのために使用される書き込みブロックとは別であり、かつフラッシング操作に使用される再配置ブロックとも異なる書き込みブロックに書き込まれる。他の実施例では、SAT情報は、ブロック、例えば、非SAT情報により占められるMLCフラッシュパーティションではないバイナリフラッシュパーティション内のブロックの異なるグループに格納され得る。あるいは、SATデータと非SATデータとは同じタイプのフラッシュブロックに、ブロックを単位として分離されて、格納され得る。さらに他の実施形態では、SATデータと非SATデータとは同じブロック内で混ぜ合わされ得る。SAT1704はマルチバンクメモリ107内の全てのバンク1

10

20

30

40

50

07A～107Dのための単一のテーブルであることができ、他の実施形態では各バンクは、その特定のバンク内の情報をマッピングするだけの独立のSATを維持することができる。

【0056】

SATは、図10～12の実施形態の各々に関連する。さらに、以下の論議はホストLBAから図11～12のホストおよびメモリシステム構成に関連するDLBAと呼ばれる第2のLBA空間（ストレージLBAとも称される）への再マッピングに焦点を当てるけれども、その同じSAT手法が、ホストLBAアドレスに関連付けられたデータが中間の論理対論理変換を伴わずに物理的ブロックに直接マッピングされる図10の実施形態に適用可能である。SAT情報は、論じられる実施形態に関わらず、好ましくは記憶装置内のフラッシュメモリに格納される。ホストLBAからDLBAへの再マッピングがホスト1202において行われる図12の実施形態では、SAT情報はメモリシステム1204のフラッシュメモリに格納されるべく送られる。ストレージアドレス再マッピングアルゴリズムがメモリシステムの中のメモリマネージャにおいて実行される図10の実施形態では、DLBAという用語は、図11～12の実施形態で使用される第2の論理アドレス空間ではなくてフラッシュメモリ107内の物理アドレスを指し、DLBAアドレスのブロックは物理メモリ内のメタブロックを表す。

【0057】

ストレージアドレステーブル（SAT）は、ホストファイルシステムによって割り当てられたLBAアドレスをDLBAアドレスに関連付ける関連情報を含む。より具体的には、SATは、ホストファイルシステムによって有効なデータに割り当てられたLBAアドレス空間の中のアドレスの全ランの各々と、ストレージアドレス再マッピングアルゴリズムによって作成されたDLBAアドレス空間の中のアドレスの1つ以上のランとの間のマッピングを記録するために使用される。前述したように、システムアドレス空間の単位はLBAであり、LBAランは、ホストファイルシステムによって有効なデータに現在割り当てられているLBAアドレスの連続的セットである。LBAランはしばしば未割り当てLBAアドレスと接するけれども、SATデータ構造により必要とされるならばLBAランは複数のより小さなLBAランとして管理され得る。装置アドレス空間の単位はDLBAであり、DLBAランは、同じLBAランの中の連続するLBAアドレスにマッピングされているDLBAアドレスの連続的セットである。DLBAランは、DLBAアドレス空間の中のブロック境界で終わらせられる。各LBAランは、SATによって1つ以上のDLBAランにマッピングされる。LBAランの長さは、それがマッピングされているDLBAランの累積長さに等しい。

【0058】

LBAランのためのSATエントリは、それがマッピングされている第1のDLBAランと、DLBAランが置かれているバンクとのためのエントリへのリンクを包含する。それがさらにマッピングされ得るその後のDLBAランは、このランに直接続く連続するエントリである。DLBAランは、LBAランの絶対LBAアドレスへのではなくて、それがマッピングされているLBAランの中のもののオフセットアドレスへの逆方向リンクを包含する。個々のLBAアドレスは、LBAランの中でのLBAオフセットとして明示され得る。SATは、LBAランにマッピングされている各DLBAランの先頭に対応するLBAオフセットを記録する。したがって、個々のLBAアドレスに対応する個々のDLBAアドレスは、DLBAランの中のDLBAオフセットとして特定され得る。SAT内のLBAランは有効なデータのランだけのためのものであり得るけれども、他の実施例では、SATは有効なデータおよび使用済みとなったデータの両方のためのLBAランを記憶するようにも構成されてもよい。

【0059】

SATは、SATブロックとして知られるLBAアドレスのブロックの中に実現され得る。SATは、所定の最大数のSATブロックを含み、所定の最大数の有効なSATページを包含する。したがって、SATは、指定された最大数のSATブロックについて、そ

れ自身が索引付けできる最大数の D L B A ランを有する。一実施形態では、最大数の S A T ブロックが定義されるけれども、S A T のエントリの数はホストにより割り当てられた L B A の断片化に順応するので、S A T は最大数まで自動的にスケーラブルな可変サイズのテーブルである。したがって、ホストが非常に断片化した L B A を割り当てれば、S A T は、断片化の程度がより低い L B A のグループをデータに割り当てる場合より多いエントリを含む。したがって、ホスト L B A の断片化の程度がより低くなれば、S A T のサイズは小さくなる。断片化の程度が低ければマッピングすべき別々のランが少なくなり、別々のランが少なければ S A T のエントリは少なくなる。それは、S A T は、固定された数の論理アドレスを厳格に追跡し更新するのではなくてエントリにおいてホスト L B A アドレスのランを 1 つ以上の D L B A ランにマッピングするからである。

10

【0060】

図 17 の S A T の L B A ラン対 D L B A ランのマッピング構成に起因して、ホスト L B A アドレスの 1 つのランを 2 つ以上の D L B A ランにマッピングすることができ、ここでホスト L B A ランは有効なデータに割り当てられている連続する論理アドレスのセットであり、D L B A (あるいはストレージ L B A) ランは同じメタブロックの中にあって同じホスト L B A ランにマッピングされている D L B A アドレスの連続的セットである。S A T 索引付けおよびマッピング構造の階層が図 22 に示されている。L B A ラン 2204 および対応する D L B A ラン 2202 が示されている。L B A 対 D L B A マッピング情報は S A T ページ 2206 に包含されている。L B A 対 S A T ページ索引付け情報は S A T 索引ページ 2208 に包含され、マスタページ索引 2210 は、図 12 の実施例ではホストプロセッサと関連する R A M に、図 10 ~ 11 の実施例ではコントローラ 108 と関連する R A M 212 にキャッシュされる。

20

【0061】

S A T は通常複数の S A T ブロックを含むけれども、S A T 情報は S A T 書き込みブロックと現在指し示されている単一のブロックに書き込まれ得るだけである。他の全ての S A T ブロックは完全に書き込まれていて、有効なページおよび使用済ページの組み合わせを包含し得る。S A T ページは、ホスト L B A アドレス空間の 1 つの可変範囲内の全ての L B A ランのためのエントリを、それらがマッピングされている装置アドレス空間内のランのためのエントリと共に包含する。多数の S A T ページが存在し得る。S A T 索引ページは、ホスト L B A アドレス空間のより大きな範囲の中の全ての有効な S A T ページの位置への索引を包含する。少数の S A T 索引ページが存在し、それは通例 1 である。S A T 内の情報は、単一の S A T 書き込みブロック内の次の使用可能な位置に更新済みページを再書き込みしてページの前のバージョンを使用済みとして扱うことによって改変される。したがって、S A T 内に多数の無効なページが存在し得る。S A T ブロックは、S A T ページがメガブロックにではなくて 1 つのバンク内の個々のブロックに書き込まれるということと、ピンク色 S A T ブロックからの有効なデータが別の再配置ブロックではなくて現在の S A T 書き込みブロックにコピーされるということとを除いてホストデータについて前述したものと類似する、ページを書き込みブロックをフラッシングするためのアルゴリズムによって、管理される。

30

【0062】

各 S A T ブロックは、S A T 情報の格納に専用される D L B A アドレスのブロックである。S A T ブロックはテーブルページに分割され、それらに S A T ページ 2206 または S A T 索引ページ 2208 を書き込むことができる。S A T ブロックは、有効な S A T ページ 2206、有効な S A T 索引ページ 2208 および使用済ページの任意の組み合わせを包含することができる。図 23 を参照すると、サンプルである S A T 書き込みブロック 2300 が示されている。データは、S A T 書き込みブロック 2300 内の、インクリメントな S A T 書き込みポインタ 2302 によって定められる連続する位置に書き込まれる。データは、S A T 書き込みブロック 2300 として指定されている単一の S A T ブロックに書き込まれ得るだけである。前述したホストデータ書き込みブロックの場合と同様に、S A T 書き込みブロック 2300 が完全に書き込まれて初めて 1 つの白色ブロックが新

40

50

しいSAT書き込みブロック2300として割り当てられる。SATページ位置は、そのSATブロックの中でのもののシーケンス番号によってアドレス指定される。全てのバンクのために単一のSATが維持される1つの実施形態では、コントローラは、新しいSAT白色ブロックを割り当てるためにどのバンク107A~107Dを使用するかを交互に選択することができる。このようにして、SATを格納するために1つのバンクを不均衡に使用することを避けることができる。

【0063】

SATページ

SATページ2206は、SATにおけるマッピング情報の最小更新可能単位である。更新済みSATページ2206はSAT書き込みポインタ2302により定められる位置に書き込まれる。SATページ2206は、インクリメントするLBAアドレスを有するLBAランのセットについてのマッピング情報を包含するけれども、引き続くLBAランのアドレスは連続していなくてもよい。SATページ2206内のLBAアドレスの範囲は他のどのSATページ2206内のLBAアドレスの範囲とも重なり合わない。SATページ2206は、SATブロックの完全なセットの全体にわたって無制限に分散され得る。LBAアドレスの任意の範囲についてのSATページ2206が任意のSATブロック内にあってよい。SATページ2206は、索引バッファフィールド2304と、LBAフィールド2306と、DLBAフィールド2308と制御ポインタ2310とを含むことができる。パラメータバックアップエントリも、揮発性RAMに格納されているいくつかのパラメータの値を包含する。

【0064】

SATページ2206内のLBAフィールド2306は、LBAアドレスのある範囲の中で、データ格納のために割り当てられている連続するLBAアドレスのランのためのエントリを包含する。SATページ2206がまたがるLBAアドレスの範囲は、他のどのSATページ2206がまたがるLBAエントリの範囲とも重なり合わない。LBAフィールドは、可変長のフィールドであって、可変数のLBAエントリを包含する。LBAフィールド2306の中に、SATページ2206により索引付けされるLBAアドレスの範囲の中のどのLBAランのためにもLBAエントリ2312が存在する。LBAランは、1つ以上のDLBAランにマッピングされる。図24に示されているように、LBAエントリ2312は、次の情報、すなわち、ランの中の第1のLBA2402と、セクタ数で表したLBAランの長さ2404と、LBAランがマッピングされている第1のDLBAランの、同じSATページ2206内のDLBAフィールドの中での、DLBAエントリ番号およびバンク番号2406とを含む。

【0065】

SATページ2206の中のDLBAフィールド2308は、同じSATページ2206内のLBAフィールドの中のLBAランにマッピングされたDLBAアドレスの全てのランのためのエントリを包含する。DLBAフィールド2308は、可変長であって、可変数のDLBAエントリ2314を包含する。DLBAフィールド2308の中で、同じSATページ2206のLBAフィールド2306の中のLBAランにマッピングされているどのDLBAランについてもDLBAエントリ2314が存在する。各DLBAエントリ2314は、図25に示されているように、次の情報、すなわち、ラン内の第1のDLBAアドレス2502と、第1のDLBAアドレスがマッピングされているLBAランにおけるLBAオフセット2504とを含む。全てのSATページ2206の各々の一部分として書き込まれるけれども最も最近に書き込まれたSATページ2206においてのみ有効なままであるSATページ/索引バッファフィールドは、SAT索引エントリ2316を包含する。マルチバンクメモリ107のために単一のSATが維持される実施形態において、ラン内の第1のDLBAのエントリ2502にバンク番号も含まれる。各バンクにおいて別々のSATが維持される代替の実施形態では、出発DLBAアドレスが既にバンク特有であるから、DLBAエントリ2314においてバンク情報は不要である。

【0066】

図 2 6 に示されている S A T 索引エントリ 2 3 1 6 は、関連する S A T 索引ページ 2 2 0 8 内に有効なエントリを現在持っていない S A T 内のどの S A T ページ 2 2 0 6 のためにも存在する。S A T 索引エントリは、S A T ページ 2 2 0 6 が書き込まれるたびに作成または更新され、関連する S A T 索引ページ 2 2 0 8 が更新されると削除される。それは、S A T ページ 2 2 0 6 により索引付けされる第 1 の L B A 2 6 0 2 と、S A T ページ 2 2 0 6 により索引付けされる最後の L B A 2 6 0 4 と、S A T ページ 2 2 0 6 を包含する S A T ブロック番号およびバンク番号 2 6 0 8 と、S A T ブロックの中の S A T ページ 2 2 0 6 のページ番号 2 6 1 0 とを包含する。S A T 索引フィールド 2 3 1 8 は、固定された数の S A T 索引エントリ 2 3 2 0 のための容量を有する。この数は、S A T ページ 2 2 0 6 および S A T 索引ページ 2 2 0 8 が書き込まれ得る相対頻度を決定する。一実施例では、この固定された数は 3 2 であり得る。

10

【 0 0 6 7 】

S A T ページのフィールドポインタ 2 3 1 0 は、L B A フィールドの先頭から D L B A フィールドの先頭までのオフセットを定める。それは、L B A エントリの数としてのオフセット値を包含する。S A T ページ 2 2 0 6 内のパラメータバックアップエントリは、揮発性 R A M に格納されているパラメータの値を包含する。これらのパラメータ値は、パワーサイクル後に（図 8 ~ 9 の実施例ではコントローラ 1 0 8 と関連付けられ、図 1 0 の実施例ではホスト C P U と関連付けられている）R A M 内の情報の初期化中に使用される。それらは、最も最近に書き込まれた S A T ページ 2 2 0 6 においてのみ有効である。

20

【 0 0 6 8 】

S A T 索引ページ

S A T 索引ページ 2 2 0 8 のセットは、S A T 内のどの有効な S A T ページ 2 2 0 6 の位置への索引をも提供する。個々の S A T 索引ページ 2 2 0 8 は、1 つの範囲の L B A アドレスに関連する有効な S A T ページの位置を定めるエントリ 2 3 2 0 を包含する。S A T 索引ページ 2 2 0 8 がまたがる L B A アドレスの範囲は、他のどの S A T 索引ページ 2 2 0 8 がまたがる L B A アドレスの範囲とも重なり合わない。エントリは、それらが関連する S A T ページの L B A アドレス範囲値にしたがって整理される。S A T 索引ページ 2 2 0 8 は、固定された数のエントリを包含する。S A T 索引ページ 2 2 0 8 は、S A T ブロックの完全なセットの全体にわたって無制限に分散され得る。任意の範囲の L B A アドレスのための S A T 索引ページ 2 2 0 8 が任意の S A T ブロック内にあってよい。S A T 索引ページ 2 2 0 8 は、S A T 索引フィールドとページ索引フィールドとを含む。

30

【 0 0 6 9 】

S A T 索引フィールド 2 3 1 8 は、S A T 索引ページ 2 2 0 8 がまたがる L B A アドレス範囲の中の全ての有効な S A T ページのための S A T 索引エントリを包含する。S A T 索引エントリ 2 3 2 0 は、単一の S A T ページ 2 2 0 6 に関連し、次の情報、すなわち、S A T ページ 2 2 0 6 により索引付けされる第 1 の L B A と、S A T ページ 2 2 0 6 を包含する S A T ブロック番号と、S A T ブロックの中の S A T ページ 2 2 0 6 のページ番号とを包含する。ページ索引フィールドは、S A T 内の全ての有効な S A T 索引ページ 2 2 0 8 のためのページ索引エントリを包含する。ページ索引エントリは、S A T 内のどの有効な S A T 索引ページ 2 2 0 8 のためにも存在し、次の情報、すなわち、S A T 索引ページによって索引付けされる第 1 の L B A と、S A T 索引ページを包含する S A T ブロック番号と、S A T ブロックの中の S A T 索引ページのページ番号とを包含する。ページ索引エントリは、最も最近に書き込まれた S A T 索引ページ 2 2 0 8 においてのみ有効である。

40

【 0 0 7 0 】

一時的な S A T データ構造

図 2 2 に示されているアドレスマッピングの長期記憶のための S A T 階層の一部ではないけれども、S A T を更新するための階層的プロシージャの中で追加のデータ構造が使用され得る。1 つのそのような構造は、L B A エントリと、S A T ページ 2 2 0 6 にまだ書き込まれていない L B A ランに対する更新操作またはブロックフラッシング操作から生じ

50

た新しいアドレスマッピングのための新しいエントリのための対応する D L B A マッピングとを含む S A T リストである。S A T リストは R A M 内の揮発性構造であり得る。S A T リスト内のエントリは、S A T ページ更新中にそれらが S A T ページ 2 2 0 6 に書き込まれると、クリアされる。

【 0 0 7 1 】

テーブルページ

テーブルページは、S A T ブロック内の D L B A アドレス空間の固定されたサイズの単位であり、これは 1 つの S A T ページ 2 2 0 6 または 1 つの S A T 索引ページ 2 2 0 8 を記憶するために使用される。テーブルページの最小サイズは 1 ページであり、最大サイズは 1 メタページであり、ここでページおよびメタページは、各バンク 1 0 7 A ~ 1 0 7 D のための物理メモリにおけるページおよびメタページに対応する D L B A アドレス空間の単位である。

【 0 0 7 2 】

S A T におけるエントリのサイズ

S A T ページ 2 2 0 6 および S A T 索引ページ 2 2 0 8 の中のエントリのサイズが表 1 に示されている。

【表 1】

S A T エントリのサイズ

エントリ	アドレス指定の範囲	バイト数で表された エントリのサイズ
S A T ページ / L B A フィールド / L B A エントリ / 第 1 の L B A	2, 0 4 8 G B	4
S A T ページ / L B A フィールド / L B A エントリ / ランレングス	3 2 M B	2
S A T ページ / L B A フィールド / L B A エントリ / D L B A エントリ番号	6 4 K エントリ	2
S A T ページ / D L B A フィールド / D L B A エントリ / 第 1 の D L B A	2, 0 4 8 G B	4
S A T ページ / D L B A フィールド / D L B A エントリ / L B A オフセット	3 2 M B	2
S A T ページ / 索引バッファフィールド / S A T 索引エントリ / 第 1 の L B A	2, 0 4 8 G B	4
S A T ページ / 索引バッファフィールド / S A T 索引エントリ / 最後の L B A	2, 0 4 8 G B	4
S A T ページ / 索引バッファフィールド / S A T 索引エントリ / S A T ブロック位置	6 4 K ブロック	2
S A T ページ / 索引バッファフィールド / S A T 索引エントリ / S A T ページ位置	6 4 K ページ	2
S A T ページ / フィールドポインタ	6 4 K エントリ	2
S A T 索引ページ / S A T 索引フィールド / S A T 索引エントリ / 第 1 の L B A	2, 0 4 8 G B	4
S A T 索引ページ / S A T 索引フィールド / S A T 索引エントリ / S A T ブロック位置	6 4 K ブロック	2
S A T 索引ページ / S A T 索引フィールド / S A T 索引エントリ / S A T ページ位置	6 4 K ページ	2
S A T 索引ページ / ページ索引フィールド / ページ索引エントリ / 第 1 の L B A	2, 0 4 8 G B	4
S A T 索引ページ / ページ索引フィールド / ページ索引エントリ / S A T ブロック位置	6 4 K ブロック	2
S A T 索引ページ / ページ索引フィールド / ページ索引エントリ / S A T ページ位置	6 4 K ページ	2

【 0 0 7 3 】

アドレス変換

S A T は、ホストファイルシステムの L B A アドレスに対応する D L B A アドレスを迅速に突き止めるために有益である。一実施形態では、有効なデータにマッピングされている L B A アドレスだけが S A T に含まれる。S A T ページ 2 2 0 6 は、1 S A T ページ 2 2 0 6 から別の S A T ページへと L B A 範囲同士が重なり合わないよう L B A 順に配列されるので、所望のデータに向かって迅速に進むために単純なサーチアルゴリズムが使用

され得る。このアドレス変換プロシージャの一例が図 2 7 に示されている。ターゲット L B A 2 7 0 2 がまず (ストレージアドレス再マッピングの実施例がそれぞれ図 1 1 のように構成されているかまたは図 1 2 のように構成されているかにより) コントローラまたはプロセッサによって受け取られる。他の実施形態では、S A T が有効なデータおよび使用済みとなったデータにマッピングされた L B A アドレスを含んで、データが有効であるのかそれとも使用済みとなっているのかを追跡するということが考慮され得る。

【 0 0 7 4 】

図 2 7 は、アドレス変換プロシージャを示すほかに、最後に書き込まれた S A T 索引ページからのページ索引フィールドと最後に書き込まれた S A T ページからの索引バッファフィールドとがどのように構成され得るかも示す。図 2 7 の実施例では、これら 2 つのフィールドは、ストレージ装置またはホストの R A M などの揮発性メモリで一時的に維持される。最後に書き込まれた S A T 索引ページのページ索引フィールドは、どの S A T 索引ページへのポインタをも含む。索引バッファフィールドは、索引ページにまだ書き込まれていない最近書き込まれた S A T ページのための索引エントリのセットを包含し得る。

【 0 0 7 5 】

対応する D L B A アドレスとのターゲット L B A アドレスについてのマッピング情報は、そのターゲットアドレスを含む 1 つの範囲の L B A アドレスについての全マッピング情報を包含する特定の S A T ページ 2 2 0 6 において保持される。アドレス変換プロシージャの第 1 の段階は、このターゲット S A T ページを識別して読むことである。図 2 7 を参照すると、ターゲット L B A のための S A T 索引エントリが存在するか否かを判定するために、最後に書き込まれた S A T ページの索引バッファフィールドのキャッシュされているバージョンに対して二分探索が実行される (ステップ 2 7 0 4 において)。ターゲット S A T ページが最近再書き込みされているけれどもターゲット S A T ページの新しい位置を記録した S A T 索引エントリを組み込んだ S A T 索引ページがまだ書き込まれていなければ、1 つのエントリが存在する。ターゲット L B A のための S A T 索引エントリが発見されたならば、それはターゲット S A T ページの位置を定義し、このページが読み出される (ステップ 2 7 0 6 において)。

【 0 0 7 6 】

ステップ 2 7 0 4 でターゲット L B A のための S A T 索引エントリが発見されなければ、ターゲット L B A のための S A T 索引エントリの位置を突き止めるために、最後に書き込まれた S A T 索引ページのページ索引フィールドのキャッシュされたバージョンに対して二分探索が実行される (ステップ 2 7 0 8 において)。ステップ 2 7 0 8 で発見されたターゲット L B A のための S A T 索引エントリは、ターゲット L B A を包含する L B A アドレス範囲のための S A T 索引ページの位置を定める。このページが読み出される (ステップ 2 7 1 0 において)。ターゲット L B A のための S A T 索引エントリの位置を突き止めるために二分探索が実行される (ステップ 2 7 1 2 において)。ターゲット L B A のための S A T 索引エントリは、ターゲット S A T ページの位置を定める。このページが読み出される (ステップ 2 7 1 4 において)。

【 0 0 7 7 】

ターゲット S A T ページがステップ 2 7 0 6 またはステップ 2 7 1 4 で読まれると、L B A から D L B A への変換が次のように実行され得る。ターゲット L B A を組み込んだターゲット L B A ランのための L B A エントリの位置を突き止めるために、L B A フィールドに対して二分探索が実行される。ターゲット L B A ランの中でのターゲット L B A のオフセットが記録される (ステップ 2 7 1 6 において)。フィールドポインタ内の情報は、二分探索のための L B A フィールドの長さ、さらに L B A フィールドの先頭に対しての D L B A フィールドの先頭をも定める (ステップ 2 7 1 8 において)。ステップ 2 7 1 6 で発見された L B A エントリは、L B A ランにマッピングされている第 1 の D L B A エントリの D L B A フィールドの中での位置を定める (ステップ 2 7 2 0 において)。ステップ 2 7 1 6 で判定されたオフセットは、ステップ 2 7 2 0 で位置が突き止められたより多くの D L B A エントリのうちの 1 つと共に、ターゲット D L B A アドレスを判定するために

使用される（ステップ 2722 において）。

【0078】

ストレージアドレス再マッピングアルゴリズムは、白色ブロックの数が所定のしきい値未満にまで減ると、書き込みブロックにおけるホストデータの書き込みにより白色容量が消費されるのと同じ速度でデータ書き込みのために割り当てられ得る使用可能な白色容量が作られることを保証するのに十分な速度でピンク色ブロックに対するフラッシング（再配置とも称される）操作が行われなければならないという原理にしたがって動作する。データ書き込みのために割り当てられ得る使用可能な白色クラスタ容量は、白色ブロック内の容量に加えて、フラッシング操作中にデータが書き込まれ得る再配置ブロックの中の白色クラスタ容量である。

10

【0079】

フラッシング操作のために選択されたピンク色ブロック内の白色クラスタ容量が各ピンク色ブロックの $x\%$ を占めるならば、1つのピンク色ブロックに対するフラッシング操作によって作られる新しい使用可能な容量は、そのピンク色ブロックから作られる1つの完全な白色ブロックからそのフラッシングされるブロックからのデータの再配置によって再配置ブロックにおいて消費される1ブロックを差し引いた $(100 - x)$ パーセンテージである。したがって、1つのピンク色ブロックに対するフラッシング操作は、1白色ブロックの $x\%$ の新しい使用可能な容量を作る。したがって、書き込まれるホストデータにより満たされる各書き込みブロックのために、 $100 / x$ 個のピンク色ブロックに対してフラッシング操作が実行されなければならない、再配置されなければならないデータは $(100 - x) / x$ ブロックである。したがって、プログラムされたセクタの、ホストにより書き込まれたセクタに対する比は、およそ $1 + (100 - x) / x$ として定義される。

20

【0080】

1つの平均的ピンク色ブロックにおける白色クラスタ容量のパーセンテージは、使用されている総装置容量のパーセンテージと、赤色ブロックであるデータを包含しているブロックのパーセンテージとにより決定される。例えば、装置が80%満たされ、データを包含するブロックの30%が赤色ブロックであるならば、ピンク色ブロックは26.2%白色クラスタ容量を含む。装置内のLBAアドレスでのデータ削除の分布が不均一であれば、あるピンク色ブロックが白色容量の平均%の2倍を有するという結果になりそうである。したがって、この例では、フラッシング操作のために選択されるピンク色ブロックは52.4%白色容量を有し、すなわち $x = 52.4$ であり、ホストにより書き込まれたデータのセクタあたりのプログラムされたセクタの比は1.90になる。

30

【0081】

どのピンク色ブロックをフラッシングするかを決定するときには、それがホストデータピンク色ブロックであろうとSATピンク色ブロックであろうと、ストレージアドレス再マッピングアルゴリズムは、NTFSにより書き込まれる\$bitmapファイルを監視することによって未割り当てアドレスの指示を検出することができる。フラッシング操作は2つの仕方ですケジュールされ得る。好ましくは、フラッシング操作は、バックグラウンド操作として動作し、したがって、SSDまたは他の携帯可能なフラッシュ記憶装置がアイドルでホストデータ書き込み速度が影響を受けない間に限って機能する。あるいは、フラッシング操作は、ホストがデータを書き込んでいるときにアクティブであるフォアグラウンド操作で利用されてもよい。フラッシング操作がフォアグラウンド操作として構成されるならば、これらの操作は、ホストアクティビティが発生したとき、あるいは「フラッシュキャッシュ」コマンドがSSDまたは携帯可能なフラッシュ記憶装置の潜在的なパワーダウンを示すときに、自動的にサスペンドされ得る。フォアグラウンドおよびバックグラウンドフラッシング操作選択は動的決定であってよく、記憶装置がアイドル状態である間に達成され得るよりも高いフラッシング速度が必要とされるときにはフォアグラウンド操作が行われる。例えば、記憶装置が満杯になるまで一定のホストデータ書き込み速度を維持するようにフラッシング速度が制御されるようにホストまたは記憶装置はフォアグラウンドフラッシング操作およびバックグラウンドフラッシング操作の間でトグルするこ

40

50

とができる。フォアグラウンドフラッシング操作は、ホストデータ書き込み操作と交互配置され得る。例えば、ホストインターフェイスでアクティビティが続けられているために十分なアイドル時間が利用できなければ、ブロックフラッシング操作を実行するためのデータページの再配置は、ホストコマンドに応答して短いバーストをなして装置アクティビティと交互配置され得る。

【 0 0 8 2 】

S A T 更新手続き

S A T データ構造の中の要素は、表 2 に示されている階層的プロシージャを用いて更新される。

【 表 2 】

S A T のための更新構造の階層

構造	位置	内容	更新トリガ
D L B A ラン	書き込みブロック または 再配置ブロック	ホストデータ	ホストにより決定される
S A T リスト	R A M	S A T ページにまだ 書き込まれていない、 L B A 対 D L B A マッピングエントリ	D L B A ランが書き込みブロック または再配置ブロックに書き込まれるとき
S A T ページ	S A T 書き込みブロック	L B A 対 D L B A マッピングエントリ	S A T リストが満杯であるか、 あるいは所定量のホストデータが D L B A ランとして書き込まれたとき
S A T 索引バッファ	最後に書き込まれた S A T ページ	S A T 索引ページにまだ 書き込まれていない、 S A T 索引エントリ	いずれかの S A T ページが書き込まれるとき
S A T 索引ページ	S A T 書き込みブロック	S A T 索引エントリ	S A T 索引バッファが満杯になるとき、 あるいは所定数の S A T 索引ページが 更新されなければならないとき

【 0 0 8 3 】

表 2 に書き留められているように、D L B A ラン更新を除いて、特定の構造のための S A T 更新は、S A T 階層の中の下位の構造におけるアクティビティによってトリガされる。S A T リストは、完全な D L B A ランに関連するデータが書き込みブロックに書き込まれるたびに更新される。最大許容数のエントリが S A T リストに存在するときに 1 つ以上の S A T ページが更新される。S A T ページが更新されるとき、S A T リストからの 1 つ以上のエントリが S A T ページに加えられ、S A T リストから除去される。S A T リストが満杯であるときに更新される S A T ページを、ページのいくつかの異なるグループに分割することができ、単一のグループだけが単一の操作で更新されなければならない。これは、S A T 更新操作がホストからのデータ書き込み操作を遅らせるかもしれない時間を最小にするのに役立つ。この場合、S A T リストから、更新された S A T ページのグループにコピーされるエントリだけが S A T リストから除去される。更新される S A T ページのグループのサイズは、ホストシステム 1 0 0 のメモリシステム 1 0 2 にアクセスする能力を妨げないポイントにセットされ得る。1 つの実施例では、グループサイズは 4 S A T ページであり得る。

【 0 0 8 4 】

S A T 索引バッファフィールドは、最も最近に書き込まれた S A T ページにおいて有効である。それは、S A T ページが書き込まれるたびに、さらなるプログラミングなしで更新される。最後に、最大許容数のエントリが S A T 索引バッファに存在するとき、S A T 索引ページが更新される。S A T 索引ページ更新中、S A T 索引バッファからの 1 つ以上のエントリが S A T 索引ページに加えられ、S A T 索引バッファから除去される。S A T

ページ更新に関して前に書き留められたように、更新されなければならない S A T 索引ページをページのいくつかの異なるグループに分割することができ、単一のグループだけが単一の操作で更新されなければならない。これは、S A T 更新操作がホストからのデータ書き込み操作を遅らせるかもしれない時間を最小にする。S A T 索引バッファから更新された S A T 索引ページのグループにコピーされるエントリだけが、S A T 索引バッファから除去される。更新された S A T 索引ページのグループのサイズは、一実施例では 4 ページであり得る。

【 0 0 8 5 】

S A T ページまたは S A T 索引ページがまたがる L B A 範囲の中で必要とされるエントリ数は可変であって、時間と共に変化し得る。したがって、S A T 内のページがオーバーフローしたり、あるいはページの内容が非常にまばらになったりするのは稀なことではない。これらの事態は、S A T 内のページを分割し結合するための方式によって管理され得る。

10

【 0 0 8 6 】

S A T ページまたは S A T 索引ページの更新中にエントリが加えられなければならないけれどもページにおいてその変化のために利用し得る未使用スペースが不十分であるときには、そのページは 2 つに分割される。1 つの新しい S A T ページまたは S A T 索引ページが導入され、以前満杯であったページと新しい空のページとのために、それらを半分満杯にするいくつかのエントリを各々に与える L B A 範囲が判定される。その後、可能ならば単一のプログラミング操作で、両方のページが書き込まれる。ページが S A T ページである場合には、両方のページのための S A T 索引エントリは、最後に書き込まれる S A T ページ内の索引バッファフィールドに含まれる。ページが S A T 索引ページである場合には、ページ索引エントリは、最後に書き込まれる S A T 索引ページ内のページ索引フィールドに含まれる。

20

【 0 0 8 7 】

隣接する L B A 範囲を伴う 2 つ以上の S A T ページまたは 2 つの S A T 索引ページの内容がまばらであるときには、ページは結合されて単一のページにされ得る。結合は、その結果としての単一のページが 8 0 % 未満で満杯となるとときに開始される。新しい単一のページのための L B A 範囲は、別々の結合されるページがまたがる範囲により定められる。結合されるページが S A T ページである場合には、新しいページと結合されるページのための S A T 索引エントリは、最後に書き込まれる S A T ページ内の索引バッファフィールドにおいて更新される。ページが S A T 索引ページである場合には、ページ索引エントリは、最後に書き込まれる S A T 索引ページ内のページ索引フィールドにおいて更新される。

30

【 0 0 8 8 】

パワーサイクル後、すなわち電力が除去され回復された後、S A T リストを、それがパワーサイクル前にあったのと正確に同じ状態に、R A M において再構築する必要がある。これは、データヘッダ内の L B A アドレス情報から、最後の S A T ページ更新以来書き込まれた追加データを特定するために全ての書き込みブロックおよび再配置ブロックをスキャンすることによって成し遂げられ得る。最後の S A T ページ更新時におけるこれらのブロックの位置と、それらの中の書き込みポインタおよび再配置ポインタの位置とも、最後に書き込まれた S A T ページ内のフィールドに記録される。したがって、スキャンは単にこれらのポインタの位置から開始されればよい。

40

【 0 0 8 9 】

S A T ブロックをフラッシングすること

S A T ブロックをフラッシングするプロセスは、ホストから受け取られるデータについて前述したプロセスと類似するけれども、S A T ブロックだけに作用する。ストレージアドレス再マッピング書き込みおよびフラッシングアルゴリズムによって引き起こされる S A T に対する更新は、S A T ブロックを図 2 8 に示されているブロック状態間で遷移させる。始めに、次の S A T ブロックを受け取るように現在指定されているバンクのための白

50

色ブロックリストから1つの白色ブロックがS A T書き込みブロックとして割り当てられる(2802において)。S A T書き込みブロック内の最後のページが割り当てられると、ブロックは赤色S A Tブロックになる(2804において)。S A T書き込みブロックは、その中のいくつかのページが既に使用済みとなっていたならば、ピンク色S A Tブロックへの遷移を行ない得る。しかし、明瞭性を目的として、その遷移は図28には示されていない。S A TページまたはS A T索引ページが更新されると赤色S A Tブロックの中の1つ以上のページが使用済みとなり、赤色S A Tブロックはピンク色S A Tブロックになる(2806において)。再配置されるデータのためにだけ指定されている特別の書き込みブロックへ有効なデータが移される、ホストデータを包含するピンク色ブロックのフラッシング操作とは違って、ピンク色S A Tブロックのためのフラッシング操作は、有効なS A Tデータを現在のS A T書き込みブロックへ単に再配置する。選択されたピンク色S A Tブロックに対するフラッシング操作が完了すると、そのピンク色S A Tブロックは白色ブロックになる(2808において)。S A Tピンク色ブロックは、好ましくは、同じバンク107A~107D内のS A T書き込みブロックへフラッシングされる。

【0090】

次に、どのS A Tブロックがフラッシングプロシージャを受けるかを選択するプロセスを記述する。少数の有効なページまたはクラスタを包含するS A Tブロックが、フラッシングされるべき次のS A Tブロックとして選択される。ブロックは、その特定バンク内のS A Tブロックのうちの最小数の有効なページを有するS A Tブロックのうちの5%の中の1つであるべきである。ブロックの選択は、各バンクにおいて最小の有効ページ総数値を有する16個のS A Tブロックのリストを構築するバックグラウンドプロセスによって成し遂げられ得る。このプロセスは、好ましくは、M個のスケジュールされたS A Tブロックフラッシング操作に費やされる時間のうちに1サイクルを終えるべきである。

【0091】

次に、どのS A Tブロックをフラッシングするかを決定するためのバックグラウンドプロセスの1サイクルで行われるアクティビティの例が図29に示されている。まず、前のプロセスサイクルの間に特定されたブロックのセットに次ぐ、それぞれのバンク内のN個のS A Tブロックの次のセットを特定するために、各バンクのためのブロック情報テーブル(BIT)がスキャンされる(ステップ2902において)。装置初期化後の第1のプロセスサイクルにおいてS A Tブロックの第1のセットが特定されるべきである。Nの値は、その特定のアプリケーションのために適宜選択されてよく、好ましくは、S A Tフラッシングブロックの利用可能性を保証するためにMのために選択される値より大きい。一例として、Mは4であり、Nは8であり得る。セット中の各S A Tブロックについて有効ページ総数値がゼロにセットされる(ステップ2904において)。その後、セット中のいずれかのS A Tブロック内に位置する有効なS A T索引ページを特定するために、キャッシュされているページ索引フィールドにおいてページ索引エントリがスキャンされる(ステップ2906において)。それに応じて有効ページ総数値がインクリメントされる。セット中のいずれかのS A Tブロック内に位置する有効なS A Tページを特定するために、S A T索引エントリが各S A T索引ページにおいて順にスキャンされる(ステップ2908において)。それに応じて有効ページ総数値がインクリメントされる(ステップ2910において)。有効ページ総数値を判定するためにページ索引およびS A T索引ページがスキャンされた後、セット中のS A Tブロックの各々についての有効ページ総数値は、低有効ページ総数値を求めてリスト中のS A Tブロックについてのものと対照して評価され、必要ならばリスト中のブロックはセットからのブロックにより取って代わられる(ステップ2912において)。S A Tブロックフラッシング操作がスケジュールされるべきであるときには、リスト中の最低有効ページ総数値を有するブロックが選択される。

【0092】

S A Tブロックフラッシング操作において、全ての有効なS A T索引ページおよびS A Tページが、選択されたブロックから、それぞれのバンク内のS A T書き込みブロック2300のS A T書き込みポインタ2302へ再配置される。ページ索引フィールドは、最

10

20

30

40

50

後に書き込まれる S A T 索引ページにおいてだけ更新される。S A T ブロックの数がほぼ一定に保たれるためには、S A T ページおよび S A T 索引ページに対する更新操作によって消費される S A T 内のページの数、S A T ブロックフラッシング操作によって回復される使用済みとなった S A T ページおよび S A T 索引ページの数と釣り合わされなければならない。次の S A T フラッシング操作のために選択された S A T ブロック内の使用済みとなった情報のページ数は、前に図 29 に関して論じられたように判定される。次の S A T ブロックフラッシング操作は、前の S A T フラッシング操作以後に S A T に同数の有効情報ページが書き込まれたときに行われるようにスケジュールされ得る。さらに、コントローラ 108 は、各ブロックについて独立に、S A T データのピンク色ブロックをフラッシングするかあるいはホストデータのピンク色ブロックをフラッシングするかを、ピンク色ブロック内の有効なデータの量に基づいてあるいは 1 つ以上の他のパラメータに基づいて選択することができる。

【0093】

ブロック情報テーブル (B I T)

ブロック情報テーブル (B I T) は、白色ブロック、ピンク色ブロック、および S A T ブロックのためにブロックアドレスの別々のリストを記録するために使用される。マルチブロックメモリでは、各バンク 107A ~ 107D において別々の B I T が維持される。B I T 書き込みブロックは、同じバンク内の他の全ての B I T ブロックがどこに位置するかに関する情報を包含する。一実施例では、ストレージアドレス再マッピングアルゴリズムおよび関連するシステムは、書き込みブロック、再配置ブロックまたは S A T ブロックとして割り当てられるべきブロックの選択を可能にするために白色ブロックのリストを維持するのが望ましい。各バンクにおいてブロックフラッシング操作の対象となるべきピンク色ブロックおよび S A T ブロックの選択を可能にするために、ピンク色ブロックのリストを維持することも望ましい。これらのリストは、その構造が S A T のものを厳密に反映する B I T において維持される。一実施形態では、各バンク 107A ~ 107D において別々の B I T が維持され記憶される。他の 1 つの実施形態では、B I T は、バンクにより索引付けされる情報を有する単一のテーブルであり得る。

【0094】

B I T データ構造

各バンク内の B I T は、B I T ブロックとして知られる D L B A アドレスのブロックの中で実現される。ブロックリスト情報が B I T ページの中に格納され、「D L B A ブロック対 B I T ページ」索引付け情報が B I T 索引ページの中に格納される。B I T ページおよび B I T 索引ページは、同じ B I T ブロックの中で任意の順序で混合されてよい。B I T は複数の B I T ブロックから成ることができるが、B I T 情報は B I T 書き込みブロックとして現在指定されている単一のブロックに書き込まれ得るだけである。他の全ての B I T ブロックは、以前に完全に書き込まれていて、有効なページおよび使用済みページの組み合わせを包含し得る。使用済み B I T 情報のページを削除して再使用される白色ブロックを作るために前述した S A T ブロックのためのものと同じの B I T ブロックフラッシング方式が実行される。

【0095】

B I T ブロック

図 30 に示されている B I T ブロックは、B I T 情報の格納に専用される D L B A アドレスのブロックである。B I T ページ 3002 と B I T 索引ページ 3004 とを含むことができる。B I T ブロックは、有効な B I T ページ、有効な B I T 索引ページ、および使用済みページの任意の組み合わせを包含することができる。B I T 情報は、B I T 書き込みブロック 3000 として指定されている単一の B I T ブロックに書き込まれ得るだけである。B I T 情報は、B I T 書き込みブロック 3000 において、インクリメントな B I T 書き込みポインタ 3006 により定められる順次の位置に書き込まれる。B I T 書き込みブロック 3000 が完全に書き込まれると、1 つの白色ブロックが新しい B I T 書き込みブロックとして割り当てられる。B I T を構成するブロックは、それらの B I T ブロック

位置によってそれぞれ特定され、装置内のブロックの集団の中でのそれらのブロックアドレスである。B I Tブロックはテーブルページに分割され、それらにB I Tページ3 0 0 2またはB I T索引ページ3 0 0 4が書き込まれ得る。B I Tページ位置は、そのB I Tブロックの中でのもののシーケンス番号によってアドレス指定される。B I T情報を、フラッシュメモリの異なるブロック内で非B I T情報と分離することができ、非B I T情報とは異なるタイプのブロック（例えば、バイナリとM L C）へ分離することができたり、あるいは1つのブロック内で非B I T情報と混合することができたりする。

【0096】

B I Tページ3 0 0 2は、B I Tにおけるブロックリスト情報の最小更新可能単位である。更新されたB I Tページは、B I T書き込みポインタ3 0 0 6によって定められる位置に書き込まれる。B I Tページ3 0 0 2は、定められた範囲の中のD L B Aブロックアドレスを有する白色ブロック、ピンク色ブロックおよびS A Tブロックのリストを包含するけれども、どのリスト中の引き続くブロックのブロックアドレスも連続的でなくてもよい。B I Tページ内のD L B Aブロックアドレスの範囲は、他のどのB I Tページ内のD L B Aブロックアドレスの範囲とも重なり合わない。B I Tページは、B I Tブロックの完全なセットの全体にわたって無制限に分散され得る。D L B Aアドレスのどの範囲のためのB I Tページも、どのB I Tブロック内にあってもよい。B I Tページは、白色ブロックリスト（W B L）フィールド3 0 0 8、ピンク色ブロックリスト（P B L）フィールド3 0 1 0、S A Tブロックリスト（S B L）フィールド3 0 1 2および索引バッファフィールド3 0 1 4を含むとともに、さらに2つの制御ポインタ3 0 1 6をも含む。パラメータバックアップエントリも、揮発性R A Mに格納されているいくつかのパラメータの値を含む。

10

20

【0097】

B I Tページ3 0 0 2内のW B Lフィールド3 0 0 8は、B I Tページ3 0 0 2に関連するD L B Aブロックアドレスの範囲の中の、白色ブロックリスト内のブロックのためのエントリを包含する。B I Tページ3 0 0 2がまたがるD L B Aブロックアドレスの範囲は、他のどのB I Tページ3 0 0 2がまたがるD L B Aブロックアドレスの範囲とも重なり合わない。W B Lフィールド3 0 0 8は、可変長のフィールドであって、可変数のW B Lエントリを包含する。W B Lフィールドの中に、B I Tページ3 0 0 2により索引付けされるD L B Aブロックアドレスの範囲の中のどの白色ブロックについてもW B Lエントリが存在する。W B Lエントリは、ブロックのD L B Aアドレスを包含する。

30

【0098】

B I Tページ3 0 0 2内のP B Lフィールド3 0 1 0は、B I Tページ3 0 0 2に関連するD L B Aブロックアドレスの範囲の中の、ピンク色ブロックリスト内のブロックのためのエントリを包含する。B I Tページ3 0 0 2がまたがるD L B Aブロックアドレスの範囲は、他のどのB I Tページ3 0 0 2がまたがるD L B Aブロックアドレスの範囲とも重なり合わない。P B Lフィールド3 0 1 0は、可変長のフィールドであって、可変数のP B Lエントリを包含する。P B Lフィールド3 0 1 0の中に、B I Tページ3 0 0 2により索引付けされるD L B Aブロックアドレスの範囲の中のどのピンク色ブロックについてもP B Lエントリが存在する。P B Lエントリは、ブロックのD L B Aアドレスを包含する。

40

【0099】

B I Tページ内のS B Lフィールド3 0 1 2は、B I Tページ3 0 0 2に関連するD L B Aブロックアドレスの範囲の中の、S A Tブロックリスト内のブロックのためのエントリを包含する。B I Tページ3 0 0 2がまたがるD L B Aブロックアドレスの範囲は、他のどのB I Tページ3 0 0 2がまたがるD L B Aブロックアドレスの範囲とも重なり合わない。S B Lフィールド3 0 1 2は、可変長のフィールドであって、可変数のS B Lエントリを包含する。S B Lフィールド3 0 1 2の中に、B I Tページ3 0 0 2により索引付けされるD L B Aブロックアドレスの範囲の中のどのS A TブロックについてもS B Lエントリが存在する。S B Lエントリは、ブロックのD L B Aアドレスを包含する。

50

【 0 1 0 0 】

索引バッファフィールド 3 0 1 4 は、全ての B I T ページ 3 0 0 2 の一部として書き込まれるけれども、最も最近に書き込まれた B I T ページにおいてのみ有効なままである。B I T ページ 3 0 0 2 の索引バッファフィールド 3 0 1 4 は、B I T 索引エントリを包含する。関連する B I T 索引ページ 3 0 0 4 内に有効なエントリを現在有しない B I T 内のどの B I T ページ 3 0 0 2 についても B I T 索引エントリが存在する。B I T 索引エントリは、B I T ページ 3 0 0 2 が書き込まれるたびに作成または更新され、関連する B I T 索引ページ 3 0 0 4 が更新されると削除される。B I T 索引エントリは、B I T ページ 3 0 0 2 によって索引付けされる範囲の第 1 の D L B A ブロックアドレスと、B I T ページ 3 0 0 2 によって索引付けされる範囲の最後の D L B A ブロックアドレスと、B I T ページ 3 0 0 2 を包含する B I T ブロック位置と、B I T ブロックの中の B I T ページの B I T ページ位置とを含むことができる。索引バッファフィールド 3 0 1 4 は、仮に 3 2 と定められる固定された数の B I T 索引エントリのための容量を有する。この数は、B I T ページ 3 0 0 2 および B I T 索引ページ 3 0 0 4 が書き込まれ得る相対頻度を決定する。

10

B I T ページ 3 0 0 2 の制御ポインタ 3 0 1 6 は、P B L フィールド 3 0 1 0 の先頭および S B L フィールド 3 0 1 2 の先頭の、W B L フィールド 3 0 0 8 の先頭からのオフセットを定める。B I T ページ 3 0 0 2 は、リストエントリの数としてのオフセット値を包含する。

【 0 1 0 1 】

B I T 索引ページ

20

B I T 索引ページ 3 0 0 4 のセットは、B I T 内の全ての有効な B I T ページ 3 0 0 2 の位置への索引を提供する。個々の B I T 索引ページ 3 0 0 4 は、D L B A ブロックアドレスの 1 つの範囲に関連する有効な B I T ページの位置を定めるエントリを包含する。B I T 索引ページがまたがる D L B A ブロックアドレスの範囲は、他のどの B I T 索引ページ 3 0 0 4 がまたがる D L B A ブロックアドレスの範囲とも重なり合わない。エントリは、それらが関連する B I T ページ 3 0 0 2 の D L B A ブロックアドレス範囲値にしたがって整理される。B I T 索引ページ 3 0 0 4 は、固定された数のエントリを包含する。

【 0 1 0 2 】

B I T 索引ページは、B I T ブロックの完全なセットの全体にわたって無制限に分散され得る。D L B A ブロックアドレスのどの範囲のための B I T 索引ページ 3 0 0 4 も、どの B I T ブロック内にあってもよい。B I T 索引ページ 3 0 0 4 は、B I T 索引フィールド 3 0 1 8 およびページ索引フィールド 3 0 2 0 を含む。B I T 索引フィールド 3 0 1 8 は、B I T 索引ページ 3 0 0 4 がまたがる D L B A ブロックアドレス範囲の中の全ての有効な B I T ページのための B I T 索引エントリを含む。B I T 索引エントリは、単一の B I T ページ 3 0 0 2 に関連し、B I T ページにより索引付けされる第 1 の D L B A ブロックと、B I T ページを包含する B I T ブロック位置と、B I T ブロックの中の B I T ページの B I T ページ位置とを含むことができる。

30

【 0 1 0 3 】

B I T 索引ページ 3 0 0 4 のページ索引フィールド 3 0 2 0 は、B I T 内の全ての有効な B I T 索引ページのためのページ索引エントリを包含する。B I T ページ索引エントリは、B I T 内のどの有効な B I T 索引ページ 3 0 0 4 のためにも存在し、B I T 索引ページにより索引付けされる第 1 の D L B A ブロックと、B I T 索引ページを包含する B I T ブロック位置と、B I T ブロックの中の B I T 索引ページの B I T ページ位置とを含むことができる。

40

【 0 1 0 4 】

B I T を維持すること

B I T ページ 3 0 0 2 は、W B L 3 0 0 8、P B L 3 0 1 0 および S B L 3 0 1 2 にエントリを加え、あるいはこれらからエントリを削除するために更新される。数個のエントリに対する更新が R A M 内のリストに、そのリストがパワーサイクル後に R A M に回復され得るということを条件として、蓄積されて B I T において単一の操作で実行され得る。

50

B I T 索引バッファフィールドは、最も最近に書き込まれた B I T ページにおいて有効である。B I T ページが書き込まれるたびに付加的なプログラミングなしで更新される。B I T 索引ページが更新されるとき、B I T 索引バッファからの 1 つ以上のエントリが B I T 索引ページに加えられ、B I T 索引バッファから除去される。最大許容数のエントリが B I T 索引バッファ内に存在するときには 1 つ以上の B I T 索引ページ 3 0 0 4 が更新される。

【 0 1 0 5 】

B I T ページ 3 0 0 2 または B I T 索引ページ 3 0 0 4 がまたがる D L B A ブロック範囲の中に必要とされるエントリ数は可変であって、時間と共に変化し得る。したがって、B I T 内のページがオーバーフローしたり、あるいはページの内容が非常にまばらになったりするものは稀なことではない。これらの事態は、B I T 内のページを分割し結合するための方式によって管理され得る。

10

【 0 1 0 6 】

B I T ページ 3 0 0 2 または B I T 索引ページ 3 0 0 4 の更新中にエントリが加えられなければならないけれどもページにおいてその変化のために利用し得る未使用スペースが不十分であるときには、そのページは 2 つに分割される。1 つの新しい B I T ページ 3 0 0 2 または B I T 索引ページ 3 0 0 4 が導入され、以前満杯であったページと新しい空のページとのために、それらを半分満杯にするいくつかのエントリを各々に与える D L B A ブロック範囲が判定される。その後、可能ならば単一のプログラミング操作で、両方のページが書き込まれる。ページが B I T ページ 3 0 0 2 である場合には、両方のページのための B I T 索引エントリは、最後に書き込まれる B I T ページ内の索引バッファフィールドに含まれる。ページが B I T 索引ページ 3 0 0 4 である場合には、ページ索引エントリは、最後に書き込まれる B I T 索引ページ内のページ索引フィールドに含まれる。

20

【 0 1 0 7 】

逆に、隣接する D L B A ブロック範囲を有する 2 つ以上の B I T ページ 3 0 0 2、あるいは 2 つの B I T 索引ページ 3 0 0 4 の内容がまばらであるときには、ページは結合されて単一のページとされ得る。結合は、結果として生じる単一のページが 8 0 % 満たされるに過ぎないときに開始される。新しい単一のページのための D L B A ブロック範囲は、別々の結合されるページがまたがる範囲により定められる。結合されるページが B I T ページである場合には、新しいページおよび結合されるページのための B I T 索引エントリは、最後に書き込まれる B I T ページ内の索引バッファフィールドにおいて更新される。ページが B I T 索引ページである場合には、ページ索引エントリは、最後に書き込まれる B I T 索引ページ内のページ索引フィールドにおいて更新される。

30

【 0 1 0 8 】

B I T ブロックをフラッシングすること

B I T ブロックをフラッシングするプロセスは、S A T ブロックについて前に記載したプロセスに厳密に従い、ここでは繰り返されない。

【 0 1 0 9 】

制御ブロック

他の実施形態では、B I T および S A T 情報は同じブロックの異なるページに格納され得る。制御ブロックと称されるこのブロックは、1 ページの S A T または B I T 情報が制御ブロック内の 1 ページを占めるように構造化され得る。制御ブロックは、整数個のページを有するページユニットから成ることができ、各ページユニットは制御ブロックの中のそれ自身のシーケンス番号によりアドレス指定される。ページユニットは、物理メモリにおいて 1 ページの最小サイズおよび 1 メタページの最大サイズを有することができる。制御ブロックは、有効な S A T ページ、S A T 索引ページ、B I T ページ、B I T 索引ページ、および使用済ページの任意の組み合わせを包含することができる。したがって、別々の S A T および B I T ブロックを持つのではなくて、S A T および B I T 情報の両方を同じ 1 つのまたは複数のブロックに格納することができる。前述した別々の S A T および B I T 書き込みブロックの場合と同じく、制御情報 (S A T または B I T 情報) を単一の

40

50

制御書き込みブロックに書き込むことができるに過ぎず、制御書き込みポインタは制御データを受け取るための次の順次位置を特定し、制御書き込みブロックが完全に書き込まれると1つの書き込みブロックが新しい制御書き込みブロックとして割り当てられる。さらに、制御ブロックは、それぞれ、メモリシステム102内のバイナリブロックの集団の中でそれら自身のブロックアドレスによって特定され得る。制御ブロックは、前述した分離されているSATおよびBITブロックについて記載されたのと同じ仕方で新しい未書き込み容量を生成するためにフラッシングされ得るけれども、制御ブロックのための再配置ブロックが有効なSATまたはBIT情報に関連するページを受け入れ得るという違いがある。フラッシングのための適切なピンク色制御ブロックの選択およびタイミングはSATフラッシングプロセスについて前述したのと同じ仕方で実行され得る。

10

【0110】

LBA割り当てステータスを監視すること

ストレージアドレス再マッピングアルゴリズムは、ホストによって有効なデータに現在割り当てられているホストLBAアドレスだけのためにアドレスマッピング情報を記録する。したがって、このマッピング情報を正確に維持するために、クラスタがホストによってデータ記憶から割り当て解除されたときを判定する必要がある。

【0111】

一実施形態では、ホストファイルシステムからのコマンドが、割り当て解除されたクラスタに関する情報をストレージアドレス再マッピングアルゴリズムに提供することができる。例えば、「データセット」コマンドが、マイクロソフト コーポレイションのビスタ (Vista) オペレーティングシステムに用いられるように提案されている。「ATA8-ACS2についての削除済みデータ通知提案 (Notification of Deleted Data Proposal for ATA8-ACS2)」についての提案が、マイクロソフトによりT13に提出されている。この新しいコマンドは、削除済みデータの通知を提供するように意図されている。単一のコマンドが、2GBに及ぶ使用済みデータを表す、連続するLBAアドレスにおけるデータの削除を装置に知らせることができる。

20

【0112】

NTFSメタデータを解釈すること

トリムコマンドなどのホストファイルシステムコマンドが利用できなければ、LBA割り当てステータスを、NTFSにより書き込まれる\$bitmapシステムファイルにおける情報変化を追跡することによって監視することができ、ボリューム上の全てのクラスタの割り当てステータスのビットマップを包含する。パーソナルコンピュータ(PC)において\$bitmap変化を追跡する一例が以下で論じられる。

30

【0113】

パーティションブートセクタ

パーティションブートセクタは、パーティション上のセクタ0である。バイトオフセット0x30におけるフィールドは、表3への例の場合のように、マスタファイルテーブル(MFT)の先頭のための論理クラスタ番号を包含する。

【表3】

パーティションブートセクタにおけるバイトオフセット								MFTクラスタ
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	
D2	4F	0C	00	00	00	00	00	0xC4FD2

40

【0114】

MFTにおける\$bitmap記録

\$bitmapと称されるシステムファイルは、ボリューム上の全てのクラスタの割り

50

当てステータスのビットマップを包含する。\$ bitmap ファイルのための記録は、MFT 内の記録番号 6 である。MFT 記録は 1,024 バイトの長さを有する。したがって、\$ bitmap 記録は MFT の先頭に関して十進 12 セクタのオフセットを有する。前の例では、MFT はクラスタ 0xC4FD2、すなわち十進 806866 から始まり、セクタ十進 6454928 である。したがって、\$ bitmap ファイル記録はセクタ十進 6454940 から始まる。

【0115】

(記述されている例において) 次の情報は \$ bitmap 記録の中に存在する。表 4 の例のように、バイトオフセット 0x141 から 0x142 までのところのフィールドは、\$ bitmap ファイルについての、第 1 のデータ属性のクラスタ単位の長さを包含する。

10

【表 4】

\$ bitmap 記録における バイトオフセット		データ属性長さ
0x141	0x142	
FB	00	0xFB

20

【0116】

表 5 の例のように、バイトオフセット 0x143 から 0x145 までのところのフィールドは、\$ bitmap ファイルについての第 1 のデータ属性の先頭のクラスタ番号を包含する。

【表 5】

\$ bitmap 記録における バイトオフセット			データ属性 クラスタ
0x143	0x144	0x145	
49	82	3E	0x3E8249

30

【0117】

表 6 の例のように、バイトオフセット 0x147 から 0x148 までのところのフィールドは、\$ bitmap ファイルについての第 2 のデータ属性のクラスタ単位の長さを包含する。

【表 6】

\$ bitmap 記録における バイトオフセット		データ属性長さ
0x147	0x148	
C4	00	0xC4

40

【0118】

表 7 の例のように、バイトオフセット 0x149 から 0x14B までのところのフィールドは、\$ bitmap ファイルについての第 1 のデータ属性の先頭と第 2 のデータ属性の先頭との間のクラスタの数を包含する。

50

【表 7】

\$ bitmap 記録における バイトオフセット			データ属性 クラスタ ジャンプ
0 x 1 4 9	0 x 1 4 A	0 x 1 4 B	
3 5	8 2	3 E	0 x 3 E 8 2 3 5

【0119】

10

\$ bitmap ファイルについてのデータ属性

\$ bitmap ファイルについてのデータ属性の中のセクタは、論理クラスタ番号の順に、ボリューム内の全てのクラスタの割り当てステータスのビットマップを包含する。「1」はフィルシステムによってクラスタがデータ格納のために割り当てられていることを表し、「0」はクラスタが空いていることを表した。ビットマップ内の各バイトは、8 クラスタまたは 64 十進セクタの論理範囲に関連する。ビットマップ内の各セクタは、0 x 1, 0 0 0 (十進 4 0 9 6) クラスタまたは 0 x 8, 0 0 0 (十進 3 2 7 6 8) セクタの論理範囲に関連する。ビットマップ内の各クラスタは、0 x 8, 0 0 0 (十進 3 2 7 6 8) クラスタまたは 0 x 4 0, 0 0 0 (十進 2 6 2 1 4 4) セクタの論理範囲に関連する。

【0120】

20

クラスタ割り当てステータスを維持すること

ホストからの書き込み操作が \$ bitmap ファイルについてのデータ属性の中のセクタに向けられるたびに、セクタの前のバージョンがストレージ装置から読み出されて、そのデータがホストによりちょうど書き込まれたばかりのデータと比較されなければならない。「1」状態から「0」状態へトグルした全てのビットが特定されなければならない。ホストによって割り当て解除されたクラスタの対応する論理アドレスが判定されなければならない。提案されたトリムコマンドなどのコマンドまたは NTFS メタデータ追跡が、ホストによるクラスタ割り当て解除があったことを示すたびに、ストレージアドレステーブル (SAT) は、指定されたクラスタについてのアドレスの割り当て解除を記録するために更新されなければならない。

30

【0121】

LBA アドレスのブロック全体の DLBA ランへの SAT マッピング

図 17 に示されている DLBA アドレスのランへの有効なホスト LBA ランだけのマッピングとは対照的に、SAT を作る代替りの方法が図 3 1 ~ 3 2 に示され、ここでは LBA アドレスのメガブロックの中の全ての LBA アドレスが、LBA アドレスが有効なデータと関連付けられているかいないかに関わらず、マッピングされる。有効なデータと関連付けられている LBA アドレスの各ランのために別々の LBA エントリを SAT において作る代わりに、各 LBA アドレスメガブロックが SAT 上で単一のエントリであるように LBA アドレスのメガブロックが SAT においてマッピングされ得る。

【0122】

40

図 3 1 を参照すると、DLBA 空間内のメガブロック 3 1 0 2 が示され、単一の連続的 LBA ランがメガブロックにおいて DLBA 空間にマッピングされている。説明を簡単にするために、メガブロック 3 1 0 2 は第 1 のメガページ 3 1 0 4 の冒頭 (バンク 1 および 2 の P 1) に使用済データを含むと仮定される。LBA アドレスの 1 つの連続的ラン (図 3 2 を参照) が、前述したように 1 バンクあたりに 1 メタページずつ全バンクに渡って LBA ランを「ストライピング」するメガページ順に、メタページ P 1、バンク 3 から始まってメタページ P 3、バンク 3 まで、DLBA アドレスにマッピングされている。図 3 1 においてメガブロックの残りは使用済データを包含している。図に示されているように、各バンクは垂直に示されているそれ自身の DLBA ラン (DLBA ラン B 1 ~ B 4) を包含し、それは、(この図では水平の) メガページ書き込みアルゴリズムが連続的 LBA ア

50

ドレスの各々の引き続くメガページに沿っているために、それぞれのバンクでDLBAランのメガページ間でLBAアドレスにおいては不連続である。図32を参照すると、LBAアドレス空間3202のメガブロックは1つの連続的LBAラン3204を示し、これは、メガページにより分割されていて、図31に示されているDLBAランおよびDLBAランの中のページで標識付けされている。したがって、LBAラン3204内の第1のメガページはDLBAランB1、第1のメガページ(バンク3)にマッピングされ、次にLBAラン3204の次のメガページはDLBAランB2、ページ1(バンク4)にマッピングされるなどである。

【0123】

図32に示されているように、LBAアドレス空間内の完全なLBAアドレスメガブロックは、SAT内の単一のLBAエン트리3206として記録され得る。この実施例におけるLBAエン트리3206は、LBAアドレスメガブロックがマッピングされるDLBAランの数と、同じSATページ内の第1のDLBAエン트리へのポインタ3208とを記録する。LBAアドレスメガブロックは、記憶装置に格納されているデータの断片化の程度に応じて、最大でLBAアドレスメガブロック内のクラスの数までマッピングされ得る。

【0124】

図32の例では、LBAアドレスメガブロックは6個のLBAランを含み、4個のランは有効なデータに割り当てられ(LBAオフセットL1~L9から始まる陰影部分)、2個のランは未割り当てアドレスランである(LBAオフセット0およびL10から始まる白色部分)。LBAアドレスメガブロックのための対応するDLBAエン트리3210は、DLBAブロックによって示されるDLBAランのDLBAアドレス、アドレスオフセット(P1~P3)および長さに対応するLBAオフセットに関連付ける。各LBAランのための別々のLBAエントリを記録し、有効なデータと関連付けられているLBAランだけを記録する、図17に関して前に論じられたSATのバージョンとは違って、LBAアドレスメガブロック内の全てのLBAランが記録される。したがって、有効なデータに現在割り当てられていないLBAアドレスブロック480内のLBAランも、有効なデータに割り当てられているLBAランも、記録される。図32に示されているSATページのDLBAエン트리部分3210において、LBAアドレスの未割り当てセットの先頭を示すLBAオフセットは、DLBAアドレス空間内の「FFFFFFFFF」値と対にされる。これは、未割り当てアドレスのためのリザーブ値を示すデフォルト十六進数を表す。前述した同じ総SAT構造および機能も、図22に関して論じられた基本的SAT階層も、LBAアドレスメガブロックマッピングの実施例に当てはまるけれども、SATページは、個別LBAラン対DLBAラン情報ではなくてLBAアドレスメガブロック対DLBAランマッピング情報を表す。さらに、SAT索引ページは、この実施例では、LBAアドレスブロック対SATページマッピング情報を記憶する。

【0125】

図33を参照すると、サンプルであるLBAアドレスフォーマット3300が示されている。アドレスフォーマット3300は32ビットの長さとして示されているが、いくつかのアドレス長のうちのいずれも使用され得る。最下位ビットをメモリシステム102内のコントローラ108によってメガページにおけるLBAアドレス3302に関連するものとして扱うことができ、アドレスの中の次のビットをバンク識別子3304を表すものとして扱うことができる。4個のバンク107A~107Dがある前述した例では、これはアドレスのうちの2ビットであり得る。次のビットをデータが関連付けられるべきメガブロック内のページ3306として扱うことができ、最後のビットをメガブロック識別子3308として解釈することもできる。一実施形態では、本願明細書で論じられたメガブロック書き込みアルゴリズムにおいて各バンク内でLBAアドレスをインターリーブすることになるけれども、DLBAアドレスがバンクの中で連続的であり得るように、コントローラはバンク識別子3304のビットを取り去ることができる。図31とメガブロック書き込みアルゴリズムとを再び参照すれば、このことをより良く理解できるはずである。

10

20

30

40

50

ホストデータがメモリシステム 102 に書き込まれるときに、現在の書き込みメガブロックの第 1 の利用可能な部分がバンク 3 のメタページ P 1 であれば、アドレスが P 1、バンク 3 へ、次に、P 1、バンク 3 が完全に書き込まれた後に P 1、バンク 4 へ、再マッピングされてゆくときにコントローラ 108 はバンク識別子ビットを除去する。書き込みアルゴリズムがホストデータをメガブロックの次のメタページ（バンク順に、バンク 1 ~ 4 の各々の中の P 2）の全体にわたって連続的にストライピングし続けてゆくとき、同じアドレス手続きが適用され得る。これは、各々の連続的ページを見たときにバンク内で左から右へ垂直下方へ各バンク内で連続的 D L B A アドレス指定が行われるという結果をもたらす。図 17 および 32 の S A T パージョンは、データが記憶装置から正確に読み出され得るけれども、各バンク内のホストデータに対するフラッシング操作が各ブロックおよびバンク内の連続的 D L B A アドレスで管理され得るように、バンク情報を追跡する。

【0126】

前の論議は、主として、ホスト L B A アドレス空間から D L B A アドレス空間（ストレージ L B A アドレス空間とも称される）への論理対論理マッピングが望ましいストレージアドレス再マッピングの実施例に焦点を合わせている。この論理対論理マッピングは、図 11 および 12 の構成に利用され得る。D L B A アドレスに再マッピングされているホストデータおよびストレージ装置生成データ（例えば、S A T および B I T）は、D L B A アドレス空間内のメタブロックに現在対応するそれぞれのバンクの中のメタブロックの物理アドレスに書き込まれる。本願明細書においてグループアドレステーブルあるいは G A T と称されるこのテーブルは、D L B A アドレス空間内の全ての論理ブロックについて 1 つずつのエントリを有すると共に 1 メタブロックの物理的ブロック粒状度を有する固定サイズのテーブルであり得る。一実施形態では、各バンク内の物理的ブロックへの論理ブロックマッピングが追跡され得るように各バンク 107 A ~ 107 D はそれ自身の G A T を有する。

【0127】

論理対物理マッピング

前述したように、図 10 の実施形態では、ストレージアドレス再マッピング（S T A R）アルゴリズムは、図 11 ~ 12 の場合のようにそれぞれ記憶装置またはホスト上の別のアプリケーションにではなくて記憶装置のメモリマネージャに組み込まれている。コントローラ 108 は、ホストデータをホスト L B A からメモリシステム 102 内の各バンク 107 A ~ 107 D 内の物理アドレスへ直接マッピングする。図 10 の実施形態では、前に論じられた D L B A アドレスは、中間の D L B A（ストレージ L B A）アドレスではなくて物理メモリアドレスにより取って代われ、S A T において、D L B A ランはデータランにより取って代われる。各バンクを横切るメガページに沿って「ストライプ」を成して物理アドレスのメガブロックにホストデータを書き込む動作は、物理的ブロックの各バンクのための独立のピンク色ブロック選択およびフラッシングと同じく、同じままである。図 10 の論理対物理の実施形態は、前に論じられた D L B A アドレスおよび D L B A ランの代わりに物理アドレスおよび物理的データランに関して同じ S A T および B I T（または制御）メタブロック構造も含む。図 10 の構成におけるストレージ再マッピングアルゴリズムは、メモリシステム 102 またはホスト 100（それぞれ図 11 および 12）上の別のアプリケーションではなくてメモリシステム 102 内のメモリコントローラ 108 の一部である。

【0128】

従来の論理対物理ブロックマッピングでは、ホストデータの断片が論理アドレスのブロックに孤立して書き込まれるたびにガーベッジコレクション操作の間にデータの塊が再配置されなければならない。ストレージアドレス再マッピングアルゴリズムでは、データは常に（論理または物理的）ブロックが満たされるまで順次アドレスに書き込まれるので、ガーベッジコレクションは不要である。本願明細書で開示されたストレージアドレス再マッピングにおいて、フラッシング操作は書き込みプロセスによってはトリガされなくて、データが使用済みとなったことに応答してトリガされるだけである。したがって、本願明

10

20

30

40

50

細書に記述されたストレージアドレス再マッピング機能を有するシステムにおいて、データ再配置オーバーヘッドは少ないはずである。フラッシング操作が最少量あるいは少なくともあるしきい値未満の、有効なデータを有するピンク色ブロックのほうへ偏らされることと、別々のバンクが独立にフラッシング可能であることとの組み合わせを、再配置されなければならない有効なデータの量および関連するオーバーヘッドを減らすためにさらに役立てることができる。

【 0 1 2 9 】

パーソナルコンピュータにおいてソリッドステートディスクアプリケーションで経験されることのある、装置の L B A アドレス空間内の関連性のないエリアにデータの短いバーストを書き込む必要があることを特徴とするランダム書き込みアプリケーションにおいてメモリシステムの性能を高めることのできる、マルチバンクメモリにおけるストレージアドレス再マッピングのためのシステムおよび方法が記述された。開示されたストレージアドレス再マッピングのある実施形態では、ホストデータは、ホストによって割り当てられた第 1 の論理アドレスから第 2 の論理アドレス空間内の連続する論理アドレスのメタブロックを有するメガブロックにマッピングされる。アドレスが完全にプログラムされたブロックと関連するデータが使用済みとなると、フラッシング手続きが開示され、各バンクについて独立に、最少量の有効なデータを有するかあるいはしきい値未満の有効なデータを有するピンク色ブロックのグループからピンク色ブロックを選択して、それらのブロックの中の有効なデータを、それらのブロックをさらなるデータ書き込みのために使用するべく解放するために、再配置する。バンク内のピンク色ブロックの中の有効なデータは、同じバンク内の再配置ブロックへ、ホストによって割り当てられた論理アドレスに関わらず、選択されたピンク色ブロックの中でそれが出現する順に、連続的に書き込まれる。このように、ホストによって割り当てられた論理アドレスランを故意に統合しないことによってオーバーヘッドが低減され得る。ホストによって割り当てられた論理アドレスと第 2 の論理アドレスとの間のマッピングおよび関連するバンクも、フラッシングに起因するマッピングの爾後の変化も追跡するために、ストレージアドレステーブルが使用される。ホストによって割り当てられた論理アドレスが物理アドレスに直接マッピングされる実施形態では、ストレージアドレステーブルがその関係を追跡し、例えば特定のブロックが有効データおよび使用済データの両方を有するピンク色ブロックであるのか未書き込み容量だけを有する白色ブロックであるのかを追跡するために、ブロック情報テーブルが維持される。

【 0 1 3 0 】

したがって、以上の詳しい記述は限定をするものではなくて説明をするものと見なされるべきであることと、本発明の趣旨および範囲を定めるべく意図されているものは全ての同等物を含む添付の特許請求の範囲であることが理解されるべきであることとが意図されている。

【 図 5 】

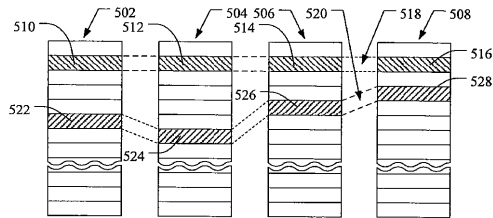


Fig. 5

【 図 6 】

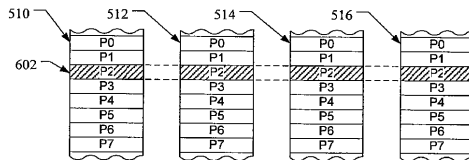


Fig. 6

【 図 8 】

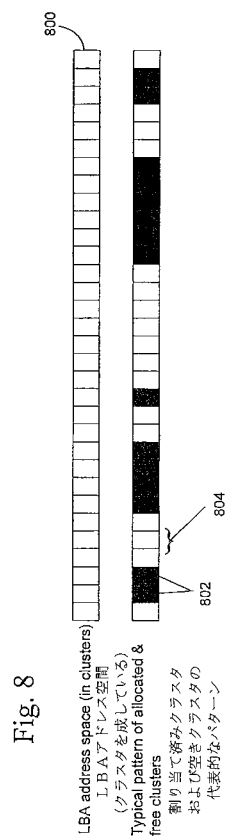


Fig. 8

【 図 9 】

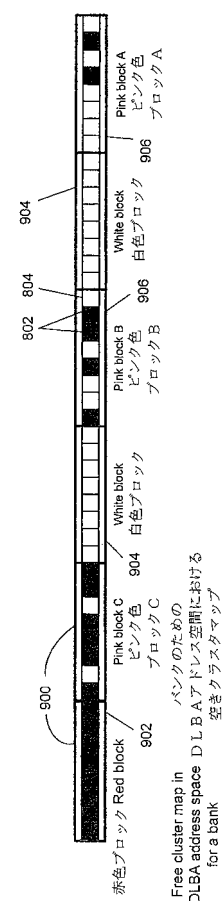
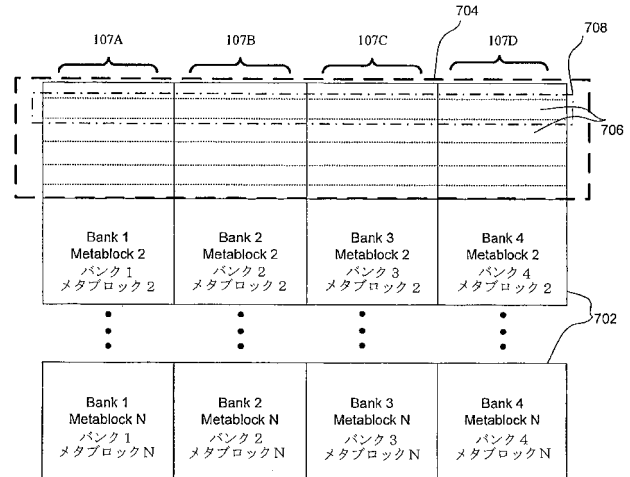


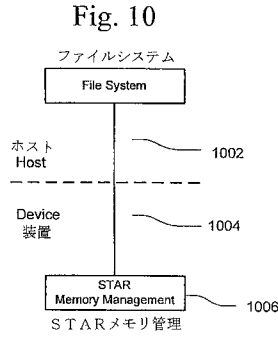
Fig. 9

【 図 7 】

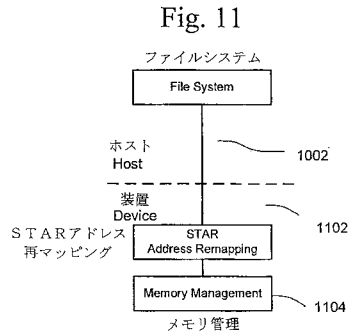
Fig. 7



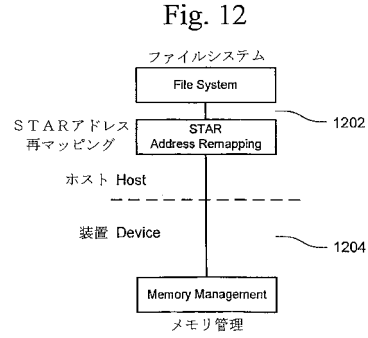
【図 10】



【図 11】



【図 12】



【図 13】

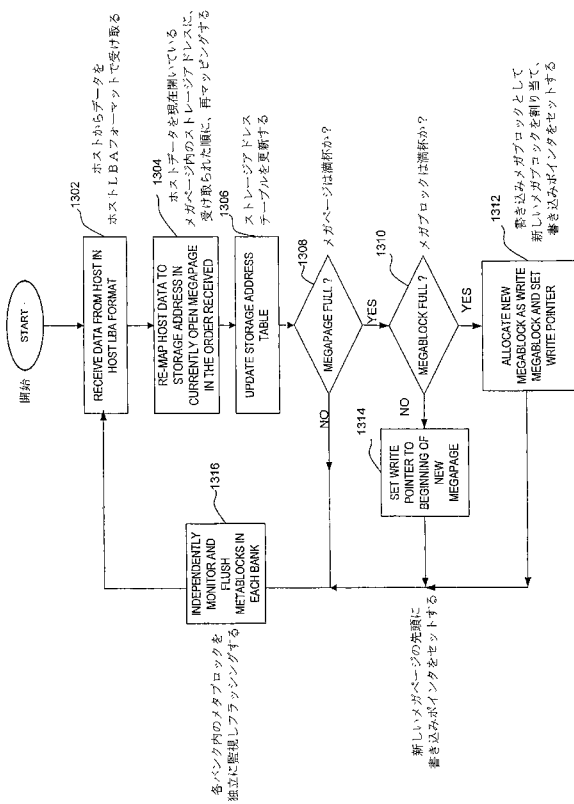


Fig. 13

【図 14】

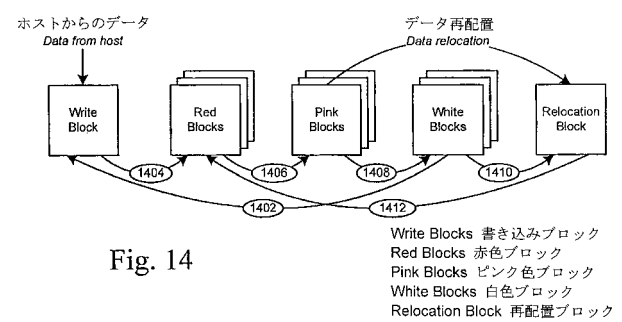
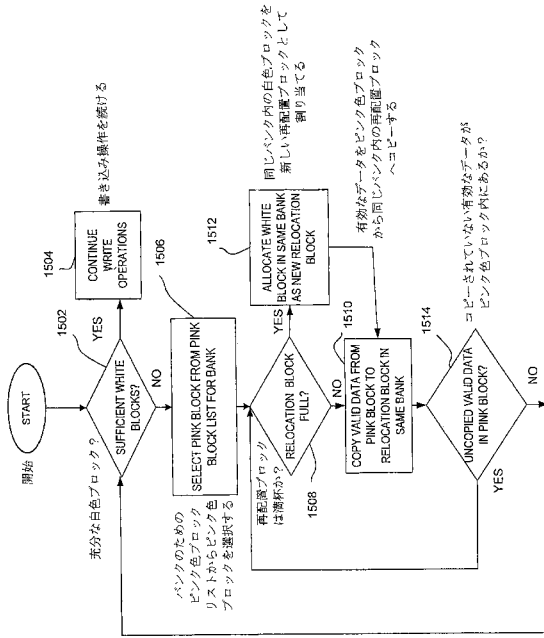


Fig. 14

【図 15】



【図 16】

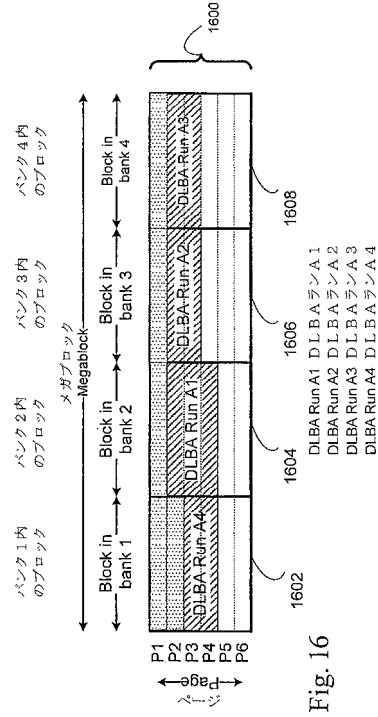


Fig. 16

【図 17】

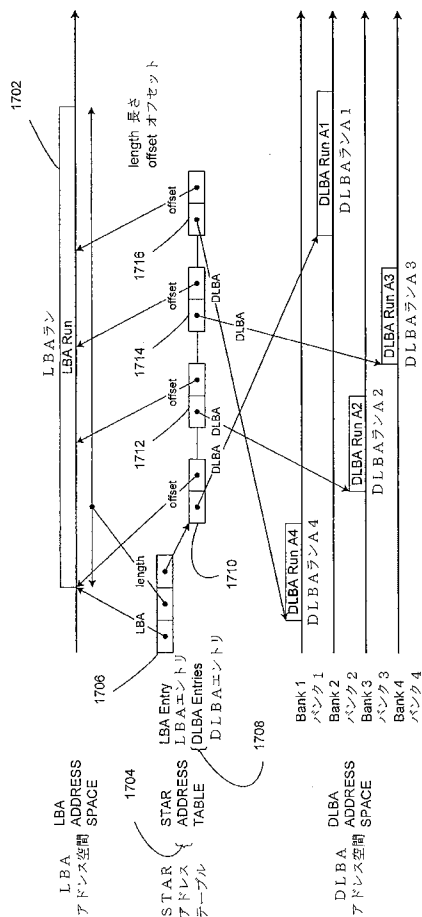


Fig. 17

【図 18】

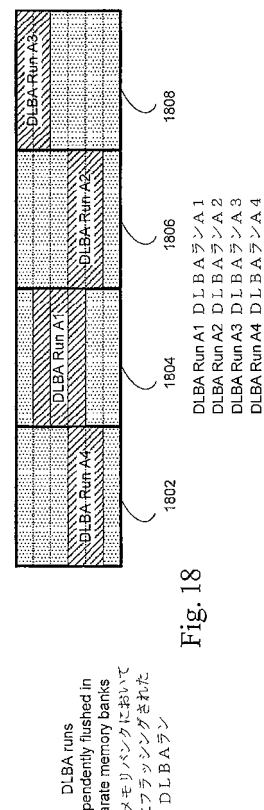
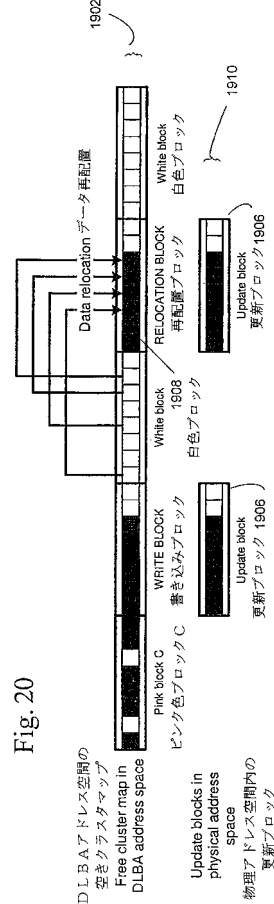


Fig. 18

【 図 2 0 】



【 図 2 3 】

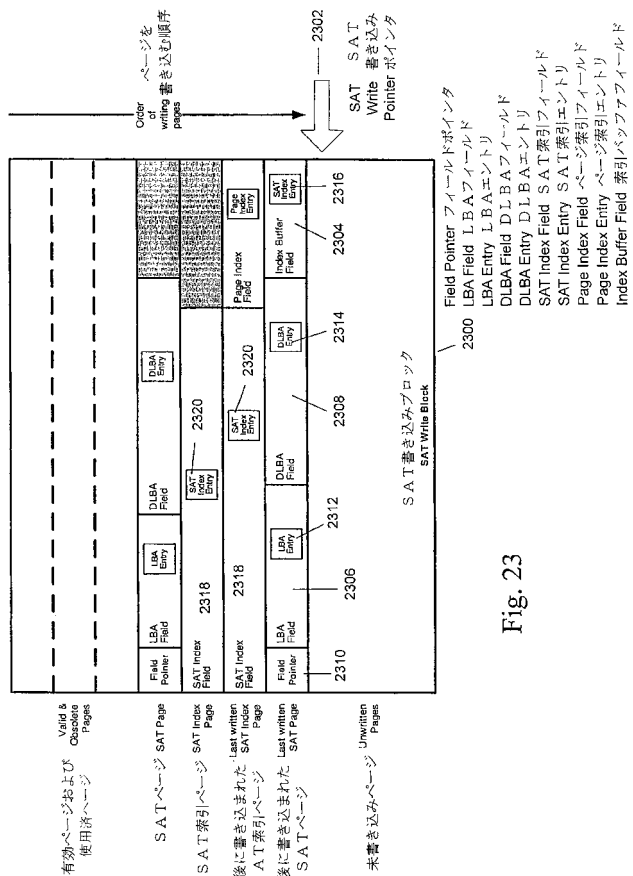
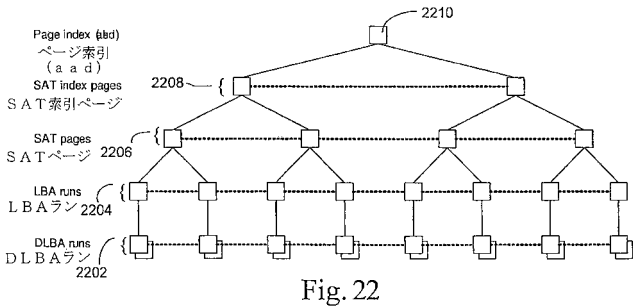


Fig. 23



【図 28】

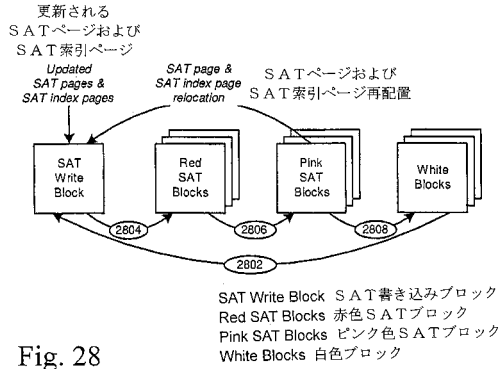
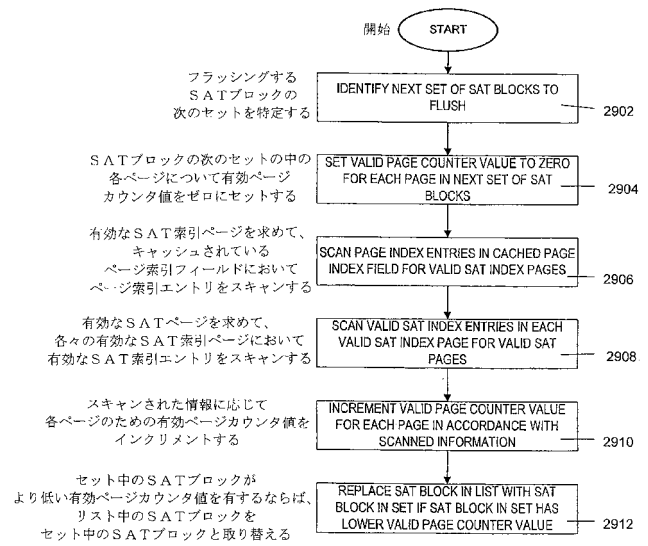


Fig. 28

【図 29】

Fig. 29



【図 30】

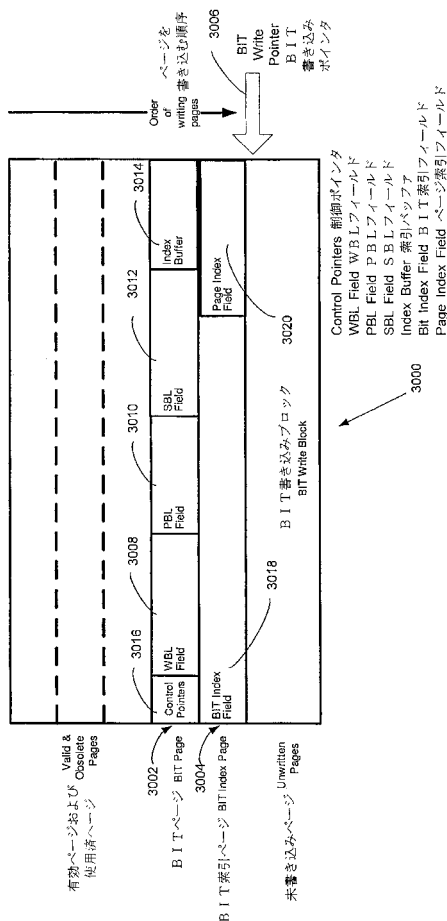


Fig. 30

【図 31】

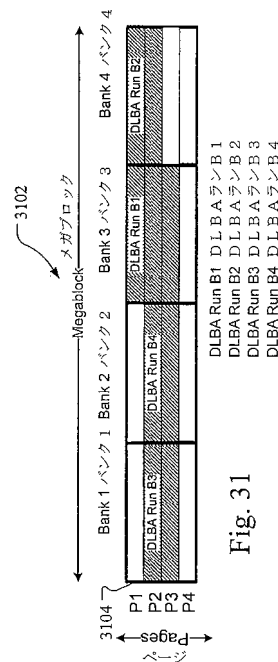
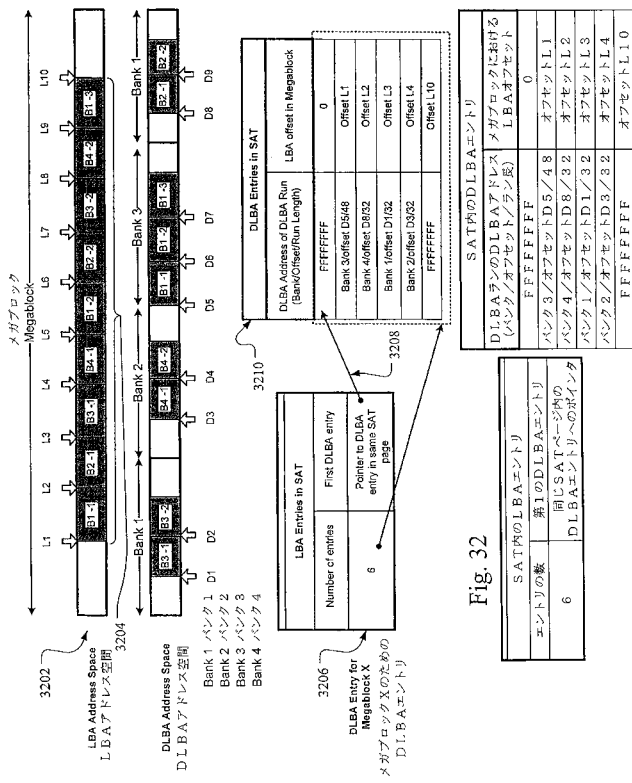


Fig. 31

【図 3 2】



【図 3 3】

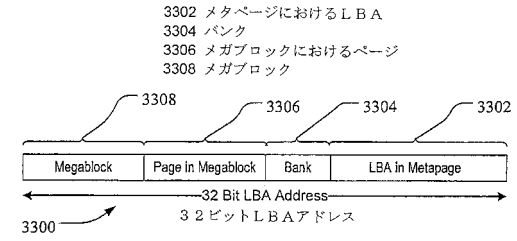


Fig. 33

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2009/040153

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F12/02		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/144361 A1 (GONZALEZ CARLOS J [US] ET AL) 30 June 2005 (2005-06-30) abstract paragraph [0012] - paragraph [0013] paragraph [0017] paragraph [0039] - paragraph [0057] figures 1-4,6	1-18
A	US 2008/034154 A1 (LEE CHARLES C [US] ET AL) 7 February 2008 (2008-02-07) paragraph [0032] - paragraph [0036] paragraph [0042] - paragraph [0053] paragraph [0076] - paragraph [0080] figures 5-7,12	1-18
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "A" document member of the same patent family		
Date of the actual completion of the international search 2 July 2009		Date of mailing of the international search report 10/07/2009
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Knutsson, Frédéric

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2009/040153

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
US 2005144361	A1	30-06-2005	CN 1918552 A	21-02-2007
			EP 1700221 A2	13-09-2006
			JP 2007517320 T	28-06-2007
			KR 20060130084 A	18-12-2006
			WO 2005066794 A2	21-07-2005
US 2008034154	A1	07-02-2008	NONE	

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM), EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM,ZW

(特許庁注：以下のものは登録商標)

１．コンパクトフラッシュ