

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5663060号  
(P5663060)

(45) 発行日 平成27年2月4日(2015.2.4)

(24) 登録日 平成26年12月12日(2014.12.12)

(51) Int.Cl.	F 1
GO6F 12/02 (2006.01)	GO6F 12/02 570A
GO6F 12/00 (2006.01)	GO6F 12/00 597U
	GO6F 12/00 560A

請求項の数 28 (全 25 頁)

(21) 出願番号	特願2013-128554 (P2013-128554)
(22) 出願日	平成25年6月19日 (2013.6.19)
(62) 分割の表示	特願2009-546065 (P2009-546065) の分割
原出願日	平成20年1月16日 (2008.1.16)
(65) 公開番号	特開2013-211056 (P2013-211056A)
(43) 公開日	平成25年10月10日 (2013.10.10)
審査請求日	平成25年6月19日 (2013.6.19)
(31) 優先権主張番号	60/885,412
(32) 優先日	平成19年1月18日 (2007.1.18)
(33) 優先権主張国	米国(US)
(31) 優先権主張番号	11/808,451
(32) 優先日	平成19年6月11日 (2007.6.11)
(33) 優先権主張国	米国(US)

(73) 特許権者	508159260 サンディスク イエル リミテッド イスラエル国、44425、クファー サ バ、アティール イエダ ストリート 7
(74) 代理人	100075144 弁理士 井ノ口 寿
(72) 発明者	ラッサー、メナヘム イスラエル国、44531、コハブ ヤイ リ、ドレブ ストリート 4
(72) 発明者	メイア、エイブラハム イスラエル国、75438、リション レ ジオン、ハシーラ ストリート 16
審査官	野田 佳邦

最終頁に続く

(54) 【発明の名称】フラッシュメモリシステムの高速起動を容易にする方法およびシステム

## (57) 【特許請求の範囲】

## 【請求項 1】

起動動作を容易にするためにデータを維持するメモリシステムであって、

複数のブロックを有する不揮発性メモリと、

揮発性メモリと、

前記不揮発性メモリのコントローラであって、

メモリシステムの特定の状態変化のためには更新されるが、メモリシステムのすべての状態変化のためには更新されない管理テーブルの第1のコピーを、第1の部分および第2の部分を含む複数のブロックの第1の部分の中にのみ記憶し、

管理テーブルの第2のコピーを前記揮発性メモリに記憶し、

前記管理テーブルがメモリシステムの第1の状態と関連付けられたデータを含み、前記第1の状態が第1の時間と関連付けられ、

将来情報データ構造を記憶した後で発生することが予測されるメモリシステムのイベントに関連する記録を含み、かつどの空きブロックが割り当てられるべきかに基づいた順序を示す前記不揮発性メモリの空いているブロックの順序付きリストも含む前記将来情報データ構造を、複数のブロックの第2の部分の中にのみ記憶し、

前記順序付きリストにより示された少なくとも第1のブロックが割り当てられるのかを判断し、

前記第1のブロックが割り当てられたと判断したことに応じて、前記第1のブロックに対応する仮想ブロック番号を判断し、その仮想ブロック番号に基づいて前記管理テーブ

10

20

ルの第 2 のコピーを更新し、

前記順序付きリストにより示された第 2 のブロックが割り当てられないのかを判断し、

第 2 のブロックが割り当てられないと判断したことに応じて、前記管理テーブルの更新された第 2 のコピーを前記不揮発性メモリに保存するように機能する前記不揮発性メモリのコントローラと、

を備えるメモリシステム。

【請求項 2】

請求項 1 記載のメモリシステムにおいて、

前記コントローラは、前記将来情報データ構造を記憶した後の第 2 の時間でのイベントの発生に応じて、前記将来情報データ構造に従って前記揮発性メモリの管理テーブルの更新された第 2 のコピーを修正し、この修正を前記不揮発性メモリをスキャンすることなく行って、複数のブロックのうちのどのブロックが未使用のブロックであるのかを判断するようにさらに機能するメモリシステム。 10

【請求項 3】

請求項 2 記載のメモリシステムにおいて、

前記コントローラは、イベントの発生に応じて、前記管理テーブルの第 1 のコピーを更新するのかを決定するようにさらに機能するメモリシステム。 15

【請求項 4】

請求項 3 記載のメモリシステムにおいて、

前記コントローラは、前記管理テーブルの第 1 のコピーを更新する決定に応じて、前記不揮発性メモリをスキャンすることなく前記将来情報データ構造に従って前記管理テーブルの第 1 のコピーを選択的に更新するようにさらに機能するメモリシステム。 20

【請求項 5】

起動動作を容易にするために、複数のブロックを有する不揮発性メモリを備えるメモリシステムのデータ構造を維持する方法であって、

メモリシステムの特定の状態変化のためには更新されるが、メモリシステムのすべての状態変化のためには更新されない管理テーブルの第 1 のコピーを、第 1 の部分および第 2 の部分を含む複数のブロックの第 1 の部分の中にのみ記憶し、

管理テーブルの第 2 のコピーを揮発性メモリに記憶し、

前記管理テーブルが第 1 の時間にメモリシステムの第 1 の状態と関連付けられたデータを含み、

将来情報データ構造を記憶した後で発生することが予測されるメモリシステムのイベントに関連する記録を含み、かつどの空きブロックが割り当てられるべきかに基づいた順序を示す前記不揮発性メモリの空いているブロックの順序付きリストも含む前記将来情報データ構造を、複数のブロックの第 2 の部分の中にのみ記憶し、

前記順序付きリストにより示された少なくとも第 1 のブロックが割り当てられるのかを判断し、

前記第 1 のブロックが割り当てられたと判断したことに応じて、前記第 1 のブロックに対応する仮想ブロック番号を判断し、その仮想ブロック番号に基づいて前記管理テーブルの第 2 のコピーを更新し、 40

前記順序付きリストにより示された第 2 のブロックが割り当てられないのかを判断し、

第 2 のブロックが割り当てられないと判断したことに応じて、前記管理テーブルの更新された第 2 のコピーを前記不揮発性メモリに保存する方法。

【請求項 6】

請求項 5 記載の方法において、

前記将来情報データ構造を記憶した後の第 2 の時間でのイベントの発生に応じて、前記将来情報データ構造に従って前記揮発性メモリの管理テーブルの更新された第 2 のコピーを修正し、この修正を前記不揮発性メモリをスキャンすることなく行って、複数のブロックのうちのどのブロックが未使用のブロックであるのかを判断する方法。 50

**【請求項 7】**

請求項 6 記載の方法において、  
イベントの発生に応じて、前記管理テーブルの第 1 のコピーを更新するのかを決定する方法。

**【請求項 8】**

請求項 7 記載の方法において、  
前記管理テーブルの第 1 のコピーを更新する決定に応じて、前記不揮発性メモリをスキヤンすることなく前記将来情報データ構造に従って前記管理テーブルの第 1 のコピーを選択的に更新する方法。

**【請求項 9】**

方法であって、  
不揮発性メモリを有するデータ記憶デバイスにおいて、  
前記不揮発性メモリでメモリ管理テーブルの第 1 のコピーを記憶するステップと、  
前記不揮発性メモリでデータエントリのリストを記憶するステップであって、前記エントリが前記不揮発性メモリの未使用のブロックを特定し、前記リストが未使用のブロックを割り当てる順序を規定する、データエントリのリストを記憶するステップと、  
電力投入イベントの検出に応じて、  
揮発性メモリで前記メモリ管理テーブルの第 2 のコピーを記憶するステップと、  
前記リストのエントリにアクセスして、ブロックを特定するステップと、

特定されたブロックが使用中の状態にあることに応じて、当該ブロックに対応する仮想ブロック番号を判断し、その仮想ブロック番号に基づいてメモリ管理テーブルの第 2 のコピーを更新して、特定されたブロックの使用を反映させるステップと、

前記メモリ管理テーブルの第 2 のコピーの状態の変化を分類するステップと、  
分類された状態の変化に基づいて前記メモリ管理テーブルの第 1 のコピーを選択的に更新するステップと、  
を実行する方法。

**【請求項 10】**

請求項 9 記載の方法であって、  
前記メモリ管理テーブルの第 1 のコピーを記憶するステップを前記不揮発性メモリ全体をスキヤンすることなく行って、どのブロックが未使用のブロックであるのかを判断する方法。

**【請求項 11】**

請求項 10 記載の方法であって、  
特定されたブロックが使用中の状態にあるかどうかを判断するステップをさらに含み、  
その判断は特定されたブロックの制御フィールドに記憶されたデータからなされる方法。

**【請求項 12】**

請求項 9 記載の方法であって、  
前記データ記憶デバイスの揮発性メモリで前記メモリ管理テーブルの第 2 のコピーを記憶するステップをさらに含む方法。

**【請求項 13】**

請求項 9 記載の方法であって、  
特定されたブロックが使用中の状態にある場合に、第 2 の特定されたブロックが使用中の状態にあるかどうかを判断するステップをさらに含み、その第 2 の特定されたブロックは前記リスト内の特定されたブロックの次に順序付けられる方法。

**【請求項 14】**

請求項 13 記載の方法であって、  
第 2 の特定されたブロックが未使用の状態にある場合に、前記メモリ管理テーブルの第 1 のコピーを前記メモリ管理テーブルの更新された第 2 のコピーに置き換えることによって、前記メモリ管理テーブルの第 1 のコピーを更新するステップをさらに含む方法。

**【請求項 15】**

10

20

30

40

50

請求項 9 記載の方法において、

前記メモリ管理テーブルの第 1 のコピーを更新するステップは、前記メモリ管理テーブルの第 2 のコピーを N 回更新した後に行われ、N は正の整数である方法。

【請求項 16】

請求項 9 記載の方法において、

前記メモリ管理テーブルの第 1 のコピーを更新するステップは、定期的に行われる方法

。

【請求項 17】

請求項 9 記載の方法において、

前記不揮発性メモリは、フラッシュメモリである方法。

10

【請求項 18】

装置であって、

不揮発性メモリを有するデータ記憶デバイスと、

コントローラと、を備え、

前記コントローラは、

前記不揮発性メモリでメモリ管理テーブルの第 1 のコピーを記憶し、

前記不揮発性メモリでデータエントリのリストを記憶し、前記エントリが前記不揮発性メモリの未使用のブロックを特定し、前記リストが未使用のブロックを割り当てる順序を規定し、

検出された電力投入イベントに応じて、

20

揮発性メモリで前記メモリ管理テーブルの第 2 のコピーを記憶し、

前記リストのエントリにアクセスして、ブロックを特定し、

特定されたブロックが使用中の状態にあることに応じて、当該ブロックに対応する仮想ブロック番号を判断し、その仮想ブロック番号に基づいてメモリ管理テーブルの第 2 のコピーを更新して、特定されたブロックの使用を反映させ、

前記メモリ管理テーブルの第 2 のコピーの状態の変化を分類し、

分類された状態の変化に基づいて前記メモリ管理テーブルの第 1 のコピーを選択的に更新するように構成される装置。

【請求項 19】

請求項 18 記載の装置であって、

30

前記メモリ管理テーブルの第 1 のコピーを前記不揮発性メモリ全体をスキャンすることなく更新して、どのブロックが未使用のブロックであるのかを判断する装置。

【請求項 20】

請求項 19 記載の装置であって、

前記コントローラは、特定されたブロックが使用中の状態にあるかどうかを判断するようさらに構成され、その判断は特定されたブロックの制御フィールドに記憶されたデータに基づいてなされる装置。

【請求項 21】

請求項 18 記載の装置であって、

前記コントローラは、前記データ記憶デバイスの揮発性メモリで前記メモリ管理テーブルの第 2 のコピーを記憶するようにさらに構成される装置。

40

【請求項 22】

請求項 19 記載の装置であって、

前記コントローラは、特定されたブロックが使用中の状態にある場合に、第 2 の特定されたブロックが使用中の状態にあるかどうかを判断するようさらに構成され、その第 2 の特定されたブロックは前記リスト内の特定されたブロックの次に順序付けられる装置。

【請求項 23】

請求項 22 記載の装置であって、

前記コントローラは、第 2 の特定されたブロックが未使用の状態にある場合に、前記メモリ管理テーブルの第 1 のコピーを前記メモリ管理テーブルの更新された第 2 のコピーに

50

置き換えることによって、前記メモリ管理テーブルの第1のコピーを更新するようにさらに構成される装置。

【請求項24】

プロセッサ実行可能命令を含むコンピュータ可読コードを記憶するコンピュータ可読記憶媒体であって、

前記プロセッサ実行可能命令により、プロセッサによって実行された場合に、前記プロセッサが、

不揮発性メモリでメモリ管理テーブルの第1のコピーを記憶し、

前記不揮発性メモリでデータエントリのリストを記憶し、各エントリが前記不揮発性メモリの未使用のブロックを特定し、前記リストが特定された未使用のブロックを割り当てる順序を規定し、10

検出された電力投入イベントに応じて、

揮発性メモリで前記メモリ管理テーブルの第2のコピーを記憶し、

前記リストの第1のエントリにアクセスして、第1のブロックを特定し、

第1のブロックが使用中の状態にあることに応じて、当該第1のブロックに対応する仮想ブロック番号を判断し、その仮想ブロック番号に基づいてメモリ管理テーブルの第2のコピーを更新して、第1のブロックの使用を反映させ、

所定の時間が満了するか、前記メモリ管理テーブルの第2のコピーを所定の回数更新するかの少なくともいずれか1つに基づいて前記メモリ管理テーブルの第1のコピーを選択的に更新し、20

前記メモリ管理テーブルの第1のコピーを更新することは、前記不揮発性メモリ全体をスキャンすることなく行って、どのブロックが未使用のブロックであるのかを判断するコンピュータ可読記憶媒体。

【請求項25】

請求項24記載のコンピュータ可読記憶媒体において、

前記プロセッサ実行可能命令により、前記プロセッサによって実行された場合に、前記プロセッサが、第1のブロックが使用中の状態にある場合に、第2のブロックが使用中の状態にあるかどうかを判断し、その第2のブロックは前記リスト内の第1のブロックの次に順序付けられるコンピュータ可読記憶媒体。30

【請求項26】

請求項24記載のコンピュータ可読記憶媒体において、

前記プロセッサ実行可能命令により、前記プロセッサによって実行された場合に、前記プロセッサが、第2のブロックが未使用の状態にある場合に、前記メモリ管理テーブルの第1のコピーを前記メモリ管理テーブルの更新された第2のコピーに置き換えることによって、前記メモリ管理テーブルの第1のコピーを更新するコンピュータ可読記憶媒体。30

【請求項27】

請求項24記載のコンピュータ可読記憶媒体において、

前記メモリ管理テーブルの第2のコピーを定期的に更新するコンピュータ可読記憶媒体。40

【請求項28】

請求項24記載のコンピュータ可読記憶媒体において、

前記不揮発性メモリは、フラッシュメモリであるコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、フラッシュメモリシステムの起動を容易にするのに役立つデータ構造を維持する方法およびシステムに関する。

【背景技術】

【0002】

Lasserの「Method For Fast Wake-Up of a Flash Memory System」という米国特許第6

50

, 510, 488号(特許文献1)(以下「特許文献1」という)には、フラッシュシステムソフトウェアが最初から生成するのに時間を要する管理テーブルに依存している場合でも、フラッシュメモリシステムがその電源投入後に高速な起動時間を実現することができるようになる方法およびシステムが開示されている。このような高速な起動時間は、データ完全性を犠牲にすることなく達成される。特許文献1は、どこから見ても本願明細書に完全に記載されているかのように参考することによって援用されている。

#### 【0003】

特許文献1で説明されるように、コンピュータデータ記憶にフラッシュメモリデバイスを使用するには、ホストコンピュータのオペレーティングシステムとデバイスローレベルアクセスルーチンとの間に位置するソフトウェア変換層が従来から必要である。というのも、フラッシュ技術が、フラッシュメモリを簡単なランダムアクセス線形法でアクセスできないようにするいくつかの使用制限を有するためである。このような制限の1つとして、どんな所望のフラッシュメモリの記憶場所にもランダムに上書きすることができないことがある。具体的には、フラッシュメモリの記憶場所に新しいコンテンツを書き込むには、最初にその記憶場所を含むブロック全体を消去し(まだ必要とされる他のどの記憶場所のコンテンツも保存したまま)、その後初めて新しいコンテンツを書き込むことが必要となる。

#### 【0004】

変換層は、ホスティングオペレーティングシステムに、独立データセクタのランダムアクセスアドレス可能アレイの仮想的な見方を提示し、同時にフラッシュ媒体の実際の記憶場所にこれらの仮想アドレスをマッピングする詳細を隠しながら処理する。このような変換機構は、非常に重要であり、このようなフラッシュメモリ変換層の例は、本願明細書に完全に記載されているかのように援用されているAmir Banの米国特許第5,937,425号(特許文献2)に開示されている。Amir Banは、仮想フラッシュアドレスと物理フラッシュアドレスとの間でマッピング機構を実施する方法を開示している。このようなシステムの別の例は、本願明細書に完全に記載されているかのように参考することによってさらに援用されている米国特許第6,678,785号(特許文献3)に詳述されている。

#### 【0005】

変換プロセスは、ホストコンピュータデータアクセス要求を実際のフラッシュアクセス要求に変換するのに必要な情報をフラッシュシステムソフトウェアに提供する内部変換テーブルに依存している。このような変換テーブルは、フラッシュデバイス内に記憶されている制御情報に基づいて、システム起動中に(またはしばらく経ってホスティングソフトウェアに作製するよう要求された場合)、フラッシュメモリシステムソフトウェアによって作製される。このようなテーブルを作製しないで代わりにフラッシュメモリからの生の制御データのみを使用することが理論的に可能であっても、アクセス要求に対する応答時間が遅過ぎるので、この方法は実際には使用できない。というのも、フラッシュ上のデータへのアクセスがRAMのメモリ内のデータへのアクセスより大幅に遅いためと、RAMのメモリテーブルが、通常、実行時に必要な種類の動作の効率に対して最適化されているのに対して、フラッシュに記憶されている制御データはそうではないためである。

#### 【0006】

例えば、フラッシュ物理ユニットは、これにマッピングされる仮想ユニットの番号を含んでもよい。プログラム実行中に、仮想ユニット番号をその物理ユニット番号に何度も変換する必要があることもある。フラッシュに記憶されている制御データのみに頼らなければならぬ場合、規定の仮想ユニット番号を有するユニットを見つけるまで、簡単な媒体アクセス標準によってユニットを長時間スキャンする必要があることもある。しかし、システム起動時に一度フラッシュデバイスをスキャンし、各仮想ユニット番号を対応する物理ユニット番号にマッピングするテーブルを作製することによって、その後このようなマッピングを非常に効率的に行うことができる。

#### 【0007】

問題は、システム起動時にフラッシュデータ記憶デバイスをスキャンするのは、特に大

10

20

30

40

50

容量デバイスの場合に長い時間がかかることがあることがある。この問題は、ユーザが迅速な電源投入を期待するシステムや装置（すなわち、携帯電話、PDAなど）にとって特に悩ましい。フラッシュメモリにテーブルを記憶するだけで、ユーザが変更できないコンピュータ実行可能コードのみを記憶するフラッシュデバイスなどの読み取り専用デバイスに役立てるこができる。しかし、単にフラッシュメモリにテーブルを記憶するだけでは、頻繁に変わることもあるデータ（PDAのテキストファイルまたはスプレッドシートなど）を記憶するために使用されるデバイスを使用する場合にはうまくいかない。というのも、デバイスへの書き込みおよびデバイス内容の変更を継続的に行うと、変換テーブルの内容まで変化してしまうためである。RAMでテーブルが変化する毎にフラッシュメモリのテーブルのコピーを更新するのは、発生するオーバーヘッドがシステムの速度を大幅に低下させるので実用的ではない。その結果、フラッシュメモリに記憶されているテーブルとRAMの「正確な」テーブルとの間で差が蓄積されて行く。ここでユーザがテーブルを更新しないで電源を切って再び入れ直すと、ソフトウェアはフラッシュメモリから誤った変換テーブルを読み出して、新しいデータを書き込むときにデータが失われてしまうこともある。

#### 【0008】

特許文献1で開示されるいくつかの実施形態によれば、この問題は、フラッシュメモリに変換テーブルを記憶してそれを読み出すときには必ず検出できるように、変換テーブルを無効にするソフトウェアに対していくつかの手段を追加することによって解決される。可能な実施例（唯一の実施例ではないが）は、すべてのエントリの合計をある固定の周知の値と一致させるチェックサム値を追加すること、あるいは、記憶されているテーブルに有効性フラグを追加することを含む。さらに、変換層の特定の機能を呼び出してからシステムの電源を落とすようにアプリケーションソフトウェアに要求するべきである。

#### 【0009】

このようにして、フラッシュメモリは、システムの電源が規定どおりに切られる場合には高速起動を開始することができ、システムの電源が規定どおりに切られない場合には通常の起動に戻る。

この解決策は、多くの場合に有用であるが、この解決策が適切でないかもしれない状況もある。この解決策が適切でないこともある第1の例は、突然の停電が頻発し、電源投入イベントの多く（またはそのほとんど）が、記憶されている無効なテーブルに遭遇し、速度の遅い通常の起動となることが予測される場合である。

#### 【0010】

この解決策が適切でないこともある第2の例は、フラッシュメモリシステムをホストとする機器のオペレーティングシステムが、規定どおりのデバイスの取り外しまたは電源停止に対するサービスをソフトウェアアプリケーションに提供しない場合である。リナックス（登録商標）（Linux（登録商標））のような複雑なオペレーティングシステムはこのようなサービスを提供するが、電源を入れると記憶システムを起動するように設計され、いかなる場合もオペレーティングシステムを終了しないような、もっと単純で小規模なオペレーティングシステムが数多く存在する。このような場合、特許文献1の方法は、システムのどの電源投入でもフラッシュ管理システムの通常起動を行うことになり、この方法から得るものは何もない。

#### 【0011】

この解決策が適切でないこともある第3の例は、システムの電源を入れてからシステムが動作可能な状態になるまでの間の時間間隔の長さに厳しい制約がある場合である。そのため、停電がまれであり、ほとんどすべての電源投入事例がフラッシュ管理システムの高速起動となっても、停電がどれほど起こらいかにかかわらず、停電がもっと遅い通常電源投入シーケンスを招くことはやはり許容できない。

#### 【0012】

特許文献1の前述した欠点のため、Lasserの米国特許出願第11/382,056号（特許文献4）（以下「特許文献4」という）には、フラッシュ管理システムの高速起動の

10

20

30

40

50

問題に対する別の解決策が開示されている。特許文献4は、どこから見ても本願明細書に完全に記載されているかのように参照することによって援用されている。

特許文献4は、フラッシュメモリシステムのすべてのイベント後ではないが一部のイベント後に、1つ以上のフラッシュ管理テーブルが更新されてフラッシュメモリに保存される技術を開示している。起動時に、フラッシュメモリに記憶されている所与のフラッシュ管理テーブルが古い情報を含むことが分かれば、記憶されているテーブル（単数または複数）を使用してシステムの起動を容易にすることがまだ可能であり、その古いテーブルを無効にする必要はない。このテーブルを無効にする代わりに、電源停止および／または電力損失の前にフラッシュメモリに保存された古いフラッシュ管理テーブルを起動時に使用して、「適切な」（すなわち、システムの現状を反映する）テーブルを復元することもできる。

10

#### 【0013】

フラッシュメモリでイベントログを同時に維持することによって、これは実施される。起動時に、イベントログに記憶されているデータを使用してフラッシュメモリテーブルを更新し、これによって電力損失または電源停止の前に所定のイグジット(exit)がなかった場合でもデータ完全性を維持する。ほとんどの場合、イベントログを使用してフラッシュメモリに記憶されている「古い」テーブルから「最新の」テーブルを得るほうが、フラッシュ記憶デバイスをスキャンして最新のテーブルを作製するよりも速い。

#### 【0014】

特許文献4の欠点は、フラッシュメモリでイベントログを維持する必要があることである。一部のフラッシュ管理システムではイベントログが既に他の理由で維持されるために、これは真の欠点とはならないが、イベントログを他に必要とせず、特許文献4の方法を書き込み性能を犠牲にして利用しているフラッシュ管理システムも数多くある。

20

#### 【0015】

したがって、フラッシュデータ構造の完全性を損なわず性能も犠牲にしないで、フラッシュメモリシステムを高速起動する方法を提供することができる方法およびシステムの必要性が広く認識され、これを有することが非常に有利となる。

#### 【0016】

##### 定義

ここで開示のため、用語「ブロック」は、単一の動作で消去することができるフラッシュメモリの最小単位と定義される。用語「ページ」は、単一の動作で書き込む（歴史的背景から「プログラムする」とも呼ばれる）ことができるフラッシュメモリの最小単位と定義される。一般に、1つのブロックは多くのページを含む。

30

ここで開示のため、用語「フラッシュ管理システム」と「フラッシュファイルシステム」とは、同義語であり交換可能に使用される。これらの各用語は、モジュールによってエクスポートされるインターフェイスがファイル指向（「ファイルを開く」または「ファイルに書き込む」のようなコマンドを有する）であるか、ブロック指向（「ブロックを読み出す」または「ブロックに書き込む」のようなコマンドを有する）であるかにかかわらず、およびソフトウェアモジュールがフラッシュ管理専用のコントローラ上で動作するか、記憶システムを使用するアプリケーションが動作している同じホストコンピュータ上で動作するかにかかわらず、フラッシュメモリデバイスでデータの記憶を管理するソフトウェアモジュールを指す。

40

#### 【0017】

ここで開示のため、「フラッシュ管理テーブル」とは、フラッシュシステムアルゴリズムの動作をサポートするフラッシュ管理システムによって使用されるデータを含む任意のテーブルであり、任意の特定時点のテーブル内のデータは、その特定時点のフラッシュメモリシステムの状態のある態様を表す。例えば、フラッシュ管理テーブルがフラッシュメモリの各ブロック当たりのビットを含むテーブルであり、そのビットが対応するブロックが使用できるように空いているか否かを示す場合、第1の時点のテーブルの内容は、その時点でどのブロックが空いていてどのブロックが空いていないかに関してシステムの状

50

態の態様を表す第1のパターンのビットである。その後の時点で、そのテーブルのビットのパターンは、第1の時点のパターンと同じであることもあれば異なることもあり、一部の空きブロックが占有ブロックとなり一部の占有ブロックが空きブロックとなることによって生じる空きブロックと占有ブロックの様々な構成を示す。

#### 【0018】

ここでの開示のため、「イベント」とは、フラッシュメモリを制御する実体からフラッシュメモリへの任意の書き込み要求、消去要求、またはハウスキーピング要求である。実体とは、図1で以下に説明されるフラッシュメモリコントローラ、図5で以下に説明される汎用CPU、または図6で以下に説明される協働するフラッシュメモリコントローラと汎用CPUであってもよい。

10

ここでの開示のため、「選択されるイベント」とは、1つ以上の更新を1つ以上のメモリ管理テーブルに起こすためにフラッシュメモリシステム設計者によって選択されるイベントである。

#### 【先行技術文献】

##### 【特許文献】

##### 【0019】

【特許文献1】米国特許第6,510,488号

【特許文献2】米国特許第5,937,425号

【特許文献3】米国特許第6,678,785号

【特許文献4】米国特許出願第11/382,056号

20

#### 【発明の概要】

##### 【0020】

前述した必要性の一部または全部、および他の必要性は、本発明のいくつかの形態によって満たされる。

本発明は、フラッシュメモリシステムの今後のイベントに関連する情報をフラッシュメモリに記憶する、前述した必要性を達成する方法を含む。例えば、仮想対物理アドレス変換テーブルの現在の値をフラッシュメモリに記憶するとき、システムはそのテーブルとともに、システムのイベントに新しい空き物理ブロックの割り当てが必要となる場合に今後使用されることになる次のいくつかの物理ブロックのリストも記憶する。

##### 【0021】

30

その後、システムのイベント（書き込み要求など）によって、新しい物理ブロックが割り当てられてもよく、この新しい物理ブロックはある仮想ブロックの対応するブロックとして別の物理ブロックを置き換えることになる。その結果、アドレスマッピングフラッシュ管理テーブルの状態が変化する。このようなイベントが発生すると、フラッシュ管理システムは、記憶されているリストの先頭にあるブロックを新しく割り当てられるブロックとして割り当てる。書き込み要求の処理が完了したら、書き込み動作によって達成される仮想ブロックのマッピングはシステムの真の状態と同期していないので、記憶されているテーブルのコピーはもう最新ではない。

##### 【0022】

40

システムが起動すると、記憶されている管理テーブルのコピーがまずロードされる。この段階では、システムの電源が切れる前のいくつかのイベントが、記憶されているテーブルと正しいテーブルとの間にいくつかの不一致を招いた可能性があるので、このテーブルがシステムの状態を正確に反映していると確信することはできない。しかし、フラッシュメモリに記憶され、記憶されているテーブルに関連する「将来情報」は、記憶されているテーブルを訂正し、テーブルの正確な最新版を復元するのに充分な情報を提供する。

##### 【0023】

これが可能な理由は、マッピング状態を変更した物理ブロックのどの割り当ても（1つであった場合）リストに現れる物理ブロックを使用したはずであると分かるためである。さらに、ブロックは、リスト内の順序に従って厳密に消費されたはずである。したがって、記憶されているテーブルが最新か否かを知るためには、リストの先頭にある物理ブロッ

50

クに関してシステムの状態をチェックすれば充分である。この最初のブロックがまだ空いていると分かれば、新しい割り当ては発生しておらずテーブルは最新である。しかし、リストの最初のブロックがもう空いていない場合には、ある状態の変化がシステムに起きたはずであり、このテーブルは最新ではない。このような場合、テーブルに適用される必要のある更新は、リストの最初のブロックの割り当てを反映するものであると分かり、RAM内のテーブルのコピーを更新する。最初のブロックが使用中であると分かった場合は、リスト内の次のブロックについて同じテストを引き続き行い、同じ論理を何度も繰り返す。まだ使用されていなかったリスト内のブロックに遭遇するまで、あるいはリストの最後に達するまで、これが繰り返される。

【0024】

10

ブロックが現在使用中かどうかを見つけるおよびテーブルに適用される必要のある変更を見つけるには、フラッシュメモリのおそらく関連するブロック自体の中に記憶されている制御データにアクセスすることが必要である。フラッシュメモリのアクセスは、RAMのアクセスに比べて遅いこともあるので、この発明の方法で必要なのは、「将来リスト」内のブロックのみを調べたり、一般に前述した論理に従ってブロックの一部のみを実際に調べたりするような、いくつかのフラッシュメモリへのアクセスのみである。したがって、古いバージョンのフラッシュ管理テーブルから最新バージョンのテーブルを生成するこの方法は、全フラッシュメモリブロックをすべてスキャンすることによってテーブルを完全に復元するよりも大幅に速い。

【0025】

20

このように、本発明は、特許文献4の方法に類似するものとして見ることができる。両方の方法は、フラッシュに記憶されている追加情報を活用して記憶されているテーブルのコピーから最新バージョンの管理テーブルを復元する。しかし、特許文献4では、追加データはイベントログであるのに対して、本発明では、追加データは「将来イベント」に関連するリストである。特許文献4では、追加情報は、フラッシュメモリにテーブルを記憶した時点よりも遅い時点で記憶されるのに対して、本発明では、追加情報は、一般に、テーブルがフラッシュメモリ内に記憶されるのと同時に記憶される。

【0026】

30

システムのイベントに従って、メモリシステムのデータ構造を維持する方法であって、(a)その内容が第1の時点のメモリシステムの状態を示す少なくとも1つの管理テーブルの少なくとも一部をメモリシステムの不揮発性メモリに記憶するステップと、(b)将来情報データ構造を記憶した後で発生することが予測されるメモリシステムのイベントに関連する複数の記録を含む将来情報データ構造を不揮発性メモリに記憶するステップと、(c)第1の時点の後の第2の時点で、将来情報データ構造に従ってイベントを処理するステップと、を含む方法が初めてここで開示される。

【0027】

不揮発性メモリは、フラッシュメモリであるのが好ましい。いくつかの実施形態によれば、少なくとも1つの管理テーブルの少なくとも一部を記憶するステップは、不揮発性メモリの少なくとも1つの管理テーブルの少なくとも一部を更新するステップを含む。

いくつかの実施形態によれば、その内容が、第1の時点のメモリシステムの状態を示す少なくとも1つの管理テーブルの少なくとも一部を記憶するステップは、メモリシステムの揮発性メモリに少なくとも1つの管理テーブルの少なくとも一部を記憶するステップも含む。更新するステップは、少なくとも1つの管理テーブルの少なくとも一部が揮発性メモリでN>1回変更される毎に達成される。更新するステップを行う他の方法は、選択されるイベントに応答して定期的に更新するステップと、メモリシステムのリソースの利用可能性に従って更新するステップとを含む。

いくつかの実施形態によれば、将来情報データ構造は、第1の時点で空いている、不揮発性メモリのブロックのリストを含む。

【0028】

メモリシステムを起動する方法であって、(a)起動前の時点のメモリシステムの状態

40

50

を示す少なくとも 1 つの管理テーブルの少なくとも一部をメモリシステムの不揮発性メモリから読み出すステップと、( b ) 将来情報データ構造を記憶した後で発生することが予測されるイベントに関連する複数の記録を含む将来情報データ構造を不揮発性メモリから読み出すステップと、( c ) 将来情報データ構造の少なくとも 1 つの記録に従って少なくとも 1 つの管理テーブルの少なくとも一部を更新するステップと、を含む方法が初めてここで開示される。

#### 【 0 0 2 9 】

いくつかの実施形態によれば、不揮発性メモリはフラッシュメモリである。

いくつかの実施形態によれば、更新するステップは、少なくとも 1 つの管理テーブルの少なくとも一部を変更してメモリシステムの現在の状態を表す。更新するステップは、条件を伴う。例えば、イグジットが規定どおりであった場合、更新が実施される必要はない。例えば、いくつかの実施形態によれば、この方法は、システムの状態がその時点以降変化したかどうかを判断するために複数の記録を不揮発性メモリと比較するステップをさらに含み、更新するステップが、メモリシステムの状態がその時点以降に変化したことを条件とする。

#### 【 0 0 3 0 】

メモリモジュールであって、( a ) 第 1 の不揮発性メモリと、( b ) 不揮発性メモリのコントローラであって、( i ) その内容が第 1 の時点のメモリシステムの状態を示す少なくとも 1 つの管理テーブルの少なくとも一部を第 1 の不揮発性メモリに記憶するステップと、( i i ) 将来情報データ構造を記憶した後で発生することが予測されるメモリシステムのイベントに関連する複数の記録を含む将来情報データ構造を第 1 の不揮発性メモリに記憶するステップと、( i i i ) 将来情報データ構造を記憶するステップの後の第 2 の時点で、将来情報データ構造に従ってイベントを処理するステップと、によって、第 1 の不揮発性メモリを管理するように機能する不揮発性メモリのコントローラと、を備えるメモリモジュールが初めてここで開示される。

いくつかの実施形態によれば、このモジュールは、第 2 の不揮発性メモリをさらに備え、コントローラは、第 2 の不揮発性メモリに記憶されるコードを実行するステップによってこれらのステップを達成するように機能する。

#### 【 0 0 3 1 】

メモリシステムであって、( a ) 不揮発性メモリを備えるメモリモジュールと、( b ) メモリモジュールのホストであって、( i ) その内容が第 1 の時点のメモリシステムの状態を示す少なくとも 1 つの管理テーブルの少なくとも一部を第 1 の不揮発性メモリに記憶するステップと、( i i ) 将来情報データ構造を記憶した後で発生することが予測されるメモリシステムのイベントに関連する複数の記録を含む将来情報データ構造を第 1 の不揮発性メモリに記憶するステップと、( i i i ) 将来情報データ構造を記憶するステップの後の第 2 の時点で、将来情報データ構造に従ってイベントを処理するステップと、によって、不揮発性メモリを管理することに関与するメモリモジュールのホストと、を備えるメモリシステムが初めてここで開示される。

いくつかの実施形態によれば、これらのステップは、ホストによってのみ達成される。他の実施形態によれば、このメモリモジュールは、ホストと協働してステップを達成するコントローラも備える。

#### 【 0 0 3 2 】

メモリモジュールであって、( a ) 第 1 の不揮発性メモリと、( b ) 不揮発性メモリのコントローラであって、( i ) 起動前の時点のメモリモジュールの状態を示す少なくとも 1 つの管理テーブルの少なくとも一部を第 1 の不揮発性メモリから読み出すステップと、( i i ) その時点の後で発生することが予測されるイベントに関連する複数の記録を含む将来情報データ構造を不揮発性メモリから読み出すステップと、( i i i ) 将来情報データ構造の少なくとも 1 つの記録に従って少なくとも 1 つのフラッシュ管理テーブルの少なくとも一部を更新するステップと、によって、メモリモジュールを起動するように機能する不揮発性メモリのコントローラと、を備えるメモリモジュールが初めてここで開示され

10

20

30

40

50

る。

いくつかの実施形態によれば、このメモリモジュールは、第2の不揮発性メモリをさらに備え、コントローラは、第2の不揮発性メモリに記憶されるコードを実行するステップによってこれらのステップを達成するように機能する。

【0033】

メモリシステムであって、(a)不揮発性メモリを備えるメモリモジュールと、(b)メモリモジュールのホストであって、(i)起動前の時点のメモリシステムの状態を示す少なくとも1つの管理テーブルの少なくとも一部を不揮発性メモリから読み出すステップと、(ii)将来情報データ構造を記憶した後で発生することが予測されるイベントに関連する複数の記録を含む将来情報データ構造を不揮発性メモリから読み出すステップと、(iii)将来情報データ構造の少なくとも1つの記録に従って少なくとも1つのフラッシュ管理テーブルの少なくとも一部を更新するステップと、によって、不揮発性メモリを管理することに関与するメモリモジュールのホストと、を備えるメモリシステムが初めてここで開示される。

いくつかの実施形態によれば、これらのステップは、ホストによってのみ達成される。他の実施形態によれば、メモリモジュールは、ホストと協働してこれらのステップを達成するコントローラも備える。

【0034】

コンピュータ可読記憶媒体であって、システムのイベントに従って、メモリシステムのデータ構造を維持するためにその内部で具体化されるコンピュータ可読コードであって、(a)その内容が第1の時点のメモリシステムの状態を示す少なくとも1つの管理テーブルの少なくとも一部をメモリシステムの不揮発性メモリに記憶するプログラムコードと、(b)将来情報データ構造を記憶した後で発生することが予測されるメモリシステムのイベントに関連する複数の記録を含む将来情報データ構造を不揮発性メモリに記憶するプログラムコードと、(c)将来情報データ構造を記憶した後の第2の時点で、将来情報データ構造に従ってイベントを処理するプログラムコードと、を含むコンピュータ可読コードを有するコンピュータ可読記憶媒体が初めてここで開示される。

【0035】

コンピュータ可読記憶媒体であって、メモリシステムを起動するためにその内部で具体化されるコンピュータ可読コードであって、(a)起動前の時点のメモリシステムの状態を示す少なくとも1つの管理テーブルの少なくとも一部をメモリシステムの不揮発性メモリから読み出すプログラムコードと、(b)将来情報データ構造を記憶した後で発生することが予測されるイベントに関連する複数の記録を含む将来情報データ構造を不揮発性メモリから読み出すプログラムコードと、(c)将来情報データ構造の少なくとも1つの記録に従って少なくとも1つの管理テーブルの少なくとも一部を更新するプログラムコードと、を含むコンピュータ可読コードを有するコンピュータ可読記憶媒体が初めてここで開示される。

【図面の簡単な説明】

【0036】

【図1】本発明のいくつかの実施形態による例示的なフラッシュメモリシステムを示すブロック図である。

【図2A】本発明のいくつかの実施形態による例示的な変換テーブルを示す説明図である。

【図2B】本発明のいくつかの実施形態による例示的な変換テーブルを示す説明図である。

【図3】本発明の一実施形態によるフラッシュ管理テーブルおよび将来情報データ構造をフラッシュメモリで維持するステップを示す流れ図である。

【図4】起動のための例示的なルーチンを示す流れ図である。

【図5】本発明のいくつかの実施形態による追加の例示的なフラッシュメモリシステムを示すブロック図である。

10

20

30

40

50

【図6】本発明のいくつかの実施形態による追加の例示的なフラッシュメモリシステムを示すブロック図である。

【発明を実施するための形態】

【0037】

ここで、本発明は、特定の例示的な実施形態について説明される。本発明は、開示される例示的な実施形態に限定されないことが理解されるべきである。さらに、ここで開示されている、フラッシュメモリシステムのイベントに従ってデータ構造を維持する方法、装置およびコンピュータ可読コードのすべての特徴が、添付の特許請求の範囲のうちのどれか特定の請求項で請求される発明を実施するのに必要なわけではないことも理解されるべきである。装置の様々な要素および特徴は、本発明を完全に可能にするために記載される。さらに、プロセスまたは方法が示されているかまたは説明されているこの開示全体を通して、方法のステップは、一方のステップが最初に実施される他方のステップに依存していることが文脈から明白でない限り、どんな順序で実施されてもあるいは同時に実施されてもよいことも理解されるべきである。

10

【0038】

ここで開示されている、データ構造を維持する方法、システムおよびコンピュータ可読コードは、例えば、停電が再三発生するような環境で、フラッシュメモリの「高速起動」を容易にするのに役立つ。ただし、このことは本発明を限定するものと見なされるべきではなく、ここで開示されている、フラッシュメモリシステムのデータ構造を維持する技術のうちの1つの限定されない適用として単に開示されるだけである。

20

具体的には、ここで開示されている技術を用いて、予期しない停電が度々発生する状況においてもデータ完全性を犠牲にしないで、フラッシュ管理システムの高速起動を提供する。

【0039】

ここで、図面を参照すると、図1は、本発明のいくつかの実施形態による限定されない例示的なフラッシュメモリシステム100のブロック図である。例示的なシステム100は、データを記憶するメモリモジュール120と、ホストインターフェイス180を介してメモリモジュール120と通信するホスト装置110（ホスト装置110の例：マイクロコンピュータ、スマートカード端末、デジタルカメラ、携帯電話、PDAまたは他の何らかの装置）とを含む。

30

【0040】

メモリモジュール120は、どんな種類のものでもよいフラッシュメモリ130と、ホストインターフェイス180から受信される読み出しおよび／または書き込み要求に従ってフラッシュメモリ130にアクセスするコントローラ140とを含む。図1に示される例の場合、コントローラ140は、CPU150と、ROM160（CPU150によって実行されるコードがここに記憶されている）と、コントローラ140によるコードの実行をサポートするCPU150によって使用されるRAM170とを含む。

限定されない例示的な図1のこのブロック図は、セキュアデジタルフラッシュメモリカードまたは携帯用USBフラッシュドライブなどの一般的な不揮発性記憶モジュールを代表している。

40

【0041】

図5は、本発明のいくつかの実施形態による別の限定されない例示的なフラッシュメモリシステム220のブロック図である。例示的なフラッシュメモリシステム220は、汎用CPU250と、RAM260と、フラッシュメモリ280と、フラッシュメモリ280へのバスインターフェイス290と、ブート用ROM270と、記憶媒体装置300と、様々な他の構成要素を互いに相互接続するバス240とを含む。システム220が電源投入されると、システムはROM270からブートされ、次いでコンピュータコードおよびデータが記憶媒体300からRAM260内にロードされる。さらに、フラッシュメモリ280を制御するエミュレーションコードも記憶媒体300からロードされる。バスインターフェイス290は、CPU250から受信された読み出しおよび／または書き込み

50

要求に従って、フラッシュメモリ 280 にアクセスする。記憶媒体 300 は、本発明の方法を実施するコンピュータコードを担持するコンピュータ可読記憶媒体の一例である。一般に、記憶媒体 300 は、ハードディスクまたはフラッシュメモリデバイスである。このようなコンピュータ可読記憶媒体の他の例として、CD、DVD、ディスクケットなどが挙げられる。図 1 の例示的なフラッシュメモリシステムと異なり、この例示的なフラッシュメモリシステム 220 には、(フラッシュメモリシステムを制御する) フラッシュメモリコントローラがない。その代わりに、CPU 250 が大容量記憶 300 から RAM 260 にコントローラエミュレーションコードをロードし、次いで CPU 250 が RAM 260 からのコードを実行して、図 1 のコントローラ 140 をエミュレートする。フラッシュ管理テーブルの保存、その修復、電源投入時の復元、および他のフラッシュ管理機能は、CPU 250 によって実行されるエミュレーションコードによってすべて実施される。  
10

#### 【0042】

図 6 は、本発明のいくつかの実施形態による別の限定されない例示的なフラッシュメモリシステム 320 のブロック図である。例示的なシステム 320 は、汎用 CPU 350 と、RAM 360 と、フラッシュメモリモジュール 330 と、フラッシュメモリコントローラ 310 と、フラッシュメモリ 380 と、フラッシュメモリ 330 へのバスインターフェイス 390 と、ブート用 ROM 370 と、記憶媒体 400 と、様々な他の構成要素を互いに相互接続するバス 340 とを含む。システム 320 が電源投入されると、システムは ROM 370 からブートされ、次いでコンピュータコードおよびデータが記憶媒体 400 から RAM 360 内にロードされる。さらに、フラッシュメモリモジュール 330 を制御するエミュレーションコードも記憶媒体 400 からロードされる。バスインターフェイス 390 は、CPU 350 から受信された読み出しおよび/または書き込み要求に従って、フラッシュメモリ 380 にアクセスする。記憶媒体 300 と同様に、記憶媒体 400 は、本発明の方法を実施するコンピュータコードを担持するコンピュータ可読記憶媒体の一例である。図 5 の例示的なフラッシュメモリシステムと異なり、この例示的なフラッシュメモリシステム 320 は、フラッシュメモリモジュール 330 内に、CPU 350 と協働してフラッシュメモリシステム 380 を制御するフラッシュメモリコントローラ 310 も有する。CPU 350 は、記憶媒体 400 から RAM 360 にコントローラエミュレーションコードをロードし、次いで CPU 350 は、RAM 360 からのコードを実行して、図 1 のコントローラ 140 の機能の一部をエミュレートする。フラッシュ管理テーブルの保存、その修復、電源投入時の復元、および他のフラッシュ管理機能は、コントローラ 310 および CPU 350 によって共同で実施される。  
20  
30

#### 【0043】

##### フラッシュ管理テーブル

一般に、フラッシュメモリシステム 100 は、揮発性メモリ、例えば、メモリモジュール 120 の RAM 170、ホスト装置 110 の RAM、または他のどこか適切な場所に 1 つ以上のフラッシュ管理テーブルを記憶することに留意すべきである。このようなフラッシュ管理テーブルの 1 つの限定されない例は、仮想ブロック番号から物理ブロック番号へのアドレス変換を提供する変換テーブルである。このテーブルは、例えば、米国特許第 5,937,425 号(特許文献 2)のシステム内のような、多くのフラッシュ管理システム内に存在するマッピングである。同様な概念および方法は、他の多くの種類のフラッシュ管理テーブル、例えば、どのブロックが現在割り当てられていて占有されて使用できないか、どのブロックが割り当てられていないかを表す割り当てブロックテーブルや、1 つの仮想ブロック番号から 1 つ以上の物理ブロック番号のグループへのマッピングを表すテーブルなどにも適用可能であることに留意すべきである。  
40

時間とともに、フラッシュメモリに記憶されているデータは変化することもあり、フラッシュメモリに関連する様々な補助データも変化することがある。フラッシュ管理システムの様々なイベント(例えば、書き込み要求、ハウスキーピング動作など)が発生するのに伴い、フラッシュメモリシステムの「状態」が時間とともに変化する。

#### 【0044】

各フラッシュ管理テーブルは、フラッシュ管理システムの全体の「状態」の1つ以上の態様を表すことに留意すべきである。したがって、どの所与のフラッシュ管理テーブルまたは一連のテーブルもシステムの完全な状態を必ずしも表すとは限らず、単にシステムの1つ以上の態様だけである。前述した空きロックテーブルの例では、どのロックが空いていてどのロックが空いていないかを知ることは、システムの完全な状態を定義するのに確かに充分ではない。占有ロックは、単一の使用ページだけを含むこともある、占有ロックのページの一部または全部に有効データが書き込まれていることもある。これは、空きロックテーブルが反映されているのではなくて、他のフラッシュ管理テーブルか他の何らかの手段が反映されているにしても、そのテーブルはシステム状態の態様を表し、したがって、フラッシュ管理テーブルの定義の中に含まれる。

10

#### 【0045】

したがって、フラッシュメモリシステムの「現在の状態」または「以前の状態」または「以後の状態」または「一番最近の状態」を表す特定のフラッシュメモリテーブル（単数または複数）が、揮発性または不揮発性メモリに記憶される場合、このテーブルは、特定のフラッシュメモリテーブル（単数または複数）によって表されるフラッシュメモリシステムの態様の「以前の」または「現在の」または「以後の」または「一番最近の」状態を表す。

時間とともに、フラッシュ管理テーブルは、一連の状態に沿って移り変わり、各状態はその時点のフラッシュメモリシステムの1つ以上の態様を所与の時点で表している。テーブルによってモデル化されるシステムの状態の態様は、フラッシュメモリシステムのイベントに対応する明確な変換点を有する個別状態の間を切り換わる。

20

#### 【0046】

##### 例示的なフラッシュ管理テーブル

図2Aは、第1の時点 ( $t = t_1$ ) の仮想ロック番号を物理ロック番号にマッピングする例示的なフラッシュ管理テーブルの内容を示す。テーブルは、仮想ロック番号毎にインデックスが付けられ、対応する仮想ロックのデータを現在含んでいる物理ロック番号をリストしている。実際の実施例では、テーブルはこれらの仮想ロック番号順に並べられ、そのため、正しいエントリに直接インデックスを付けることができるので、仮想ロック番号を記憶する領域を割り当てる必要はないことが多い。しかし、提示する便宜上、テーブルは両方の列が物理的に表されるように示されている。

30

#### 【0047】

図2Aで示される仮想対物理マッピングの状態は、仮想ロック番号2をアクセスする必要があれば、対応する物理ロックはロック番号172であることをテーブルが教えてくれるというものである。ある時点で、フラッシュ管理システムにコマンドが送信されることで（あるいは、ガーベッジコレクション動作など、何も外部コマンドのないシステムによる内部ハウスキーピング動作によって）、マッピングが変化することもある。例えば、新しいデータを仮想ロック番号2に書き込む、したがって、仮想ロック番号2の以前のデータに上書きするコマンドによって、フラッシュ管理システムは、別の物理ロックを仮想ロック番号2に対応するように割り当てるることもでき、新しいデータはそのロックに記憶される。

40

#### 【0048】

図2Bは、その変化が発生した後の時点 ( $t = t_2$ ) の仮想対物理フラッシュ管理テーブルの内容を示す。このとき、仮想ロック番号2は、テーブル内で物理ロック番号777に対応し、物理ロック172にはもう対応していない。したがって、テーブルは、第1の状態から第2の状態に切り換わっている。フラッシュ管理テーブルの内容のどんな変化もそのテーブルの状態の変化として定義される。

#### 【0049】

フラッシュ管理システムで発生するすべての書き込み動作が、すべてのフラッシュ管理システムのフラッシュ管理テーブルの状態を変化させるとは限らないことが強調されるべきである。例えば、物理ロック172のページの一部が未使用であり、書き込み動作が

50

仮想ブロック番号 2 を対象に行われて、フラッシュ管理システムのアルゴリズムに従って、新しいデータが物理ブロック 172 のこれらの未使用ページに記憶されるようにする場合でも、仮想対物理マッピングは変更されず、テーブルはその書き込み動作の後でもそのまま変化しない。したがって、フラッシュ管理テーブルの状態の変化の割合は、システム内で実行される動作の割合よりも一般に低い。さらに、複数のフラッシュ管理テーブルが同じフラッシュ管理システム内で維持される場合（各テーブルはシステムの状態の異なる様子を表す）、各テーブルは、様々な時点での状態を変化させてよい。

#### 【 0050 】

##### 起動時に役立つ将来情報データ構造の記憶

前述したように、電源投入後にシステムを初期化する場合、管理ソフトウェアは、管理ソフトウェアが使用する任意のフラッシュ管理テーブルの最新のコピーを RAM（例えば、RAM170）内に作製することが必要になる場合が多い。システムの高速起動を達成するためには、記憶システムのブロック全体に分散されているデータからのみでテーブルの内容を再生するのは時間がかかり過ぎるので通常は許容することができない。テーブルの状態が変化する度にテーブルのコピーをフラッシュメモリ内に保存するという対照的な選択肢も、テーブルの状態が変化する度にテーブルのコピーをフラッシュメモリ内に保存することによって大きなオーバーヘッドが加わり、システム性能を低下させるので許容することができない。

#### 【 0051 】

本発明の様々な実施形態によって提供される技術は、これら 2 つの従来技術の方法の間の妥協点を提供する。したがって、一般に、状態の変化がある度ではなく一部の状態の変化に対してだけフラッシュ管理テーブルはフラッシュメモリ内に保存される。

これが図 3 に示され、図 3 は、本発明のいくつかの実施形態によるフラッシュ管理システムのデータ構造をフラッシュメモリ内で維持する例示的なルーチンの流れ図である。本願明細書で使用されるように、「フラッシュメモリで維持すること」は、フラッシュメモリに記憶することを含む。一般に、フラッシュメモリで維持することは、テーブルおよび / または揮発性メモリ / RAM にテーブルを投入するための関連データを維持することと、複数の時点で、テーブルの更新バージョン（すなわち、フラッシュメモリシステムの現在の状況に同期するテーブル）をフラッシュメモリに保存することとを含む。

#### 【 0052 】

図 3 の限定されない実施形態によれば、限定されない実施形態では、いくつかの時点で、システムは待機状態であり次のイベントを待っている（206）ことに留意すべきである。フラッシュ管理システムのイベントを処理する（210）と、一般に、1 つ以上のテーブルが、発生したイベントに従って揮発性メモリで更新される（214）。これらのテーブル（単数または複数）は、所与のイベントの後にフラッシュメモリで更新されてもされなくてよい。

#### 【 0053 】

##### 例示的な将来情報データ構造の説明

フラッシュ管理テーブルが、すべてとは限らないが一部のイベントの発生の後で保存されるような状況では、テーブルに関連する将来のイベントの処理に影響する情報を含むデータ構造をフラッシュメモリで維持することも有用である。例示的な実施形態では、この将来情報データ構造は、複数の記録を含み、各記録は、フラッシュ管理システムの 1 つの将来イベントに影響するデータを含んでいる。本発明を限定するものではないが、一般に、この将来情報データ構造は、各イベント後にフラッシュメモリで更新されるわけではないことに留意すべきである。将来情報データ構造は、更新後の（すなわち、フラッシュメモリシステムの最新イベントに従って更新された）テーブル（単数または複数）が必ずしもフラッシュメモリに保存されるとは限らない状況に対してデータ完全性を維持するのに役立つ。

#### 【 0054 】

一般に、任意の所与の瞬間にデータ完全性が提供される（例えば、電力の損失がある場

10

20

30

40

50

合でも)実施形態の場合、以下の状況のうちの1つが当てはまる。すなわち、a)一番最近のイベントに従って更新されるテーブルがフラッシュメモリに記憶されること、b)以前の状態(すなわち、「以前の」イベント以降に他のイベントがフラッシュメモリシステム内で発生したことを意味する、一番最近のイベントより以前の)に従って更新されるテーブルがフラッシュメモリに記憶され、フラッシュメモリに記憶される将来情報データ構造は、「以前のイベント」の時点以降に発生したすべてのイベントに関連する情報を含むことのうちの1つである。

#### 【0055】

将来情報データ構造の一つの例示的な実施例は、システムの状態(すなわち、1つ以上のフラッシュ管理テーブルに記憶されている「システムの少なくとも1つの態様」の状態)に影響を与える新しいいくつかのイベントの処理を進める情報が提供される順序付きリストであり、そのイベントとは、外部書き込みコマンド、内部ハウスキーピング動作、またはシステムの状態に影響を与える他の何らかのイベントである。

当然ながら、将来情報データ構造は、フラッシュメモリ内のどの場所に記憶されてもよく、必ずしも将来情報データ構造が関連しているフラッシュ管理テーブルに物理的に隣接する必要はない。

#### 【0056】

図3に示されるように、フラッシュ管理テーブル(单数または複数)は、フラッシュメモリシステムのすべてとは限らないが一部について(すなわち、「テーブル保存」条件を満足した(226)イベントのみについて(この条件は以下に説明される))フラッシュメモリに保存される(230)。したがって、いくつかの実施形態では、フラッシュ管理テーブル(单数または複数)がフラッシュメモリに保存される場合は必ず、更新後の将来情報データ構造も保存される(234)。このとき、フラッシュメモリに保存されるフラッシュ管理テーブル(单数または複数)は、フラッシュメモリシステムの「最新の状態に同期している」または「最新のイベントに同期している」と言われる。図3にはっきりとは示されていないが、不揮発性メモリにアクセスして将来情報データ構造を検索しなくても今後のイベントの処理を行うことができるよう、一番最近保存された将来情報データ構造のコピーをRAMメモリ内に保持しておくのが(本発明の使用にとって重要ではないにしても)望ましいことが理解されるべきである。

ステップ230を参照して、フラッシュメモリシステムの現在の状態を表すテーブルがフラッシュメモリに保存される場合、これは本願明細書では「フラッシュメモリ内のテーブルを現在の状態に同期させること」として定義されることに留意すべきである。

#### 【0057】

全体として、フラッシュ管理テーブル(单数または複数)が様々なイベントの後で(すべてのイベントの後とは限らないが)フラッシュメモリに記憶されるプロセスは、本願明細書では、「フラッシュメモリでテーブルを維持すること」として定義される。したがって、様々な時点で、フラッシュ管理テーブル(单数または複数)の別バージョン(一般に、フラッシュメモリシステムの以後の状態を示す各後続バージョン)が、フラッシュメモリに記憶される。したがって、「フラッシュメモリでテーブルを維持すること」には、フラッシュメモリに一番最近記憶されたテーブルが、どんな瞬間にもフラッシュメモリシステムの現在の状態と同期していることが必要なわけではない。図3の検査は、一番最近記憶されたフラッシュメモリテーブル(单数または複数)が、フラッシュメモリシステムの以前の状態(すなわち、もっと最近のイベントがフラッシュメモリシステムで発生する以前のシステムの状態)を示している期間が通常存在することを示す。

#### 【0058】

将来情報データ構造は、後から、例えば、フラッシュシステムの電源投入後に、フラッシュメモリから検索することもできることに留意すべきである。(以下でさらに説明されるように)データ構造を検査することによって、保存されたフラッシュ管理テーブル(单数または複数)が、システムの一番最近の状態(すなわち、テーブルに表される状態の態様)を反映しているか否かがどんな時でも判断される。

10

20

30

40

50

## 【0059】

ここで、将来情報データ構造の限定されない例を提示するために、図2Aおよび図2Bのフラッシュ管理テーブルの例を用いる。図2Aおよび図2Bのフラッシュ管理テーブルは、仮想ブロック番号から対応する物理ブロック番号へのマッピングを提供する。システムの動作中、フラッシュ管理ソフトウェアが空いている物理ブロックを所与の仮想ブロックに割り当てる場合には必ずテーブルの状態が変化し、同時に、その仮想ブロックに対応する以前の物理ブロックを解放する。

## 【0060】

本発明の例示的な実施形態によるテーブルに関連する将来情報データ構造は、現在空いているブロックの順序付きリストである。このリストは、どの空きブロックが将来割り当てられるかに基づいて正確な順序を定義する。言い換えれば、次にフラッシュ管理ソフトウェアが空きブロックを割り当てるとき、フラッシュ管理ソフトウェアはリストの先頭のブロックを選択することが保証される。2番目にフラッシュ管理ソフトウェアが空きブロックを割り付けるときには、フラッシュ管理ソフトウェアはリストの2番目のエントリのブロックを選択することが保証され、以下同様に続く。

10

## 【0061】

一般に、各物理ブロックは、その物理ブロックが現在割り付けられている仮想ブロックの番号をそれ自体の中に含んでいることに留意すべきである。システムの起動プロセスの説明の中で以下に説明されるように、この事実によって、フラッシュ管理ソフトウェアは、保存されている（最新ではない）バージョンのテーブルからと、そのテーブルの関連している将来情報リストから、テーブルの現在の状態を復元することができる。

20

## 【0062】

更新後のフラッシュ管理テーブル（単数または複数）をフラッシュメモリに保存するタイミングを決定するポリシー

更新後のテーブルがフラッシュメモリ内に保存されるためのイベントを構成するもの、すなわち、ステップ226の「保存条件」を決定するどんなポリシーも、本発明の範囲に含まれる。

本発明の範囲は、本発明の特定の実施形態に従って、更新後のフラッシュ管理テーブル（単数または複数）をフラッシュメモリに保存するタイミングを決定する多くのポリシーを含む。

30

## 【0063】

第1の実施形態では、テーブルは、テーブルがN回変化する毎に保存される。ただし、Nは所定の値である。保存のオーバーヘッドが小さい場合、Nは2と小さいこともあれば、保存のオーバーヘッドが大きい場合には、例えば、Nは100あるいはそれ以上まで大きくなることもある。したがって、カウンタ変数は初期化されてゼロになる。各イベント後、このカウンタ変数は増加される。カウンタ変数が所定のN値を超えると、1つ以上のテーブルがフラッシュメモリに保存され、これらのテーブルに関連する将来情報データ構造が再計算されて、テーブル（単数または複数）とともにフラッシュメモリに記憶される。

40

## 【0064】

第2の実施形態では、「テーブル保存」条件は、一番最近のイベントによってトリガされる状態の変化に従って決定される。そのために、状態の変化は、「小さい」変化か「大きい」変化のいずれかに分類される。小さいテーブル状態の変化で、テーブルがすぐに（すなわち、次のイベントを処理する（210）前に）保存されるわけではないが（すなわち、ステップ226後の「NO」分岐）、大きい変化では、速やかに保存される。空きブロックのフラッシュ管理テーブルにおいて、変化を小さい変化および大きい変化に分類することについての限定されない例では、空きブロックを占有ブロックに変える変化は小さいと見なされるが、占有ブロックを空きブロックに変える変化は大きいと見なされる。

## 【0065】

第3の実施形態では、テーブル（単数または複数）は、定期的に、所定の時間間隔を超

50

えると必ず保存される。一般に、テーブルが更新される頻度とフラッシュ管理テーブル（単数または複数）をフラッシュメモリに保存する際に消費されるシステムリソースの量との間にはトレードオフがある。一方、管理テーブル（単数または複数）を保存する頻度が多いほど、概して、起動中にフラッシュメモリから検索されるテーブルがより更新されている可能性が高いことを意味し、これによって、より高速な起動を提供することができる。どんな時間間隔も、本発明の範囲に含まれることに留意すべきである。例示的な限定されない実施形態では、この時間間隔は、1 / 10秒から5分以上の間である。

#### 【0066】

第4の実施形態では、テーブルを更新する頻度は、システムリソースの使用可能性に従って決定される。一例では、コントローラ140が、読み出し／書き込み／消去要求を数多く処理する場合、あるいは、ハウスキーピング動作が多い期間の間には、システムリソースを保護するために、フラッシュ管理テーブル（単数または複数）がフラッシュメモリに保存される頻度は少ない。逆にシステムが待機状態または使用レベルが低い「低使用」の期間では、通常、フラッシュ管理テーブル（単数または複数）をフラッシュシステムの性能に大きな影響を及ぼさずにフラッシュメモリにより頻繁に保存することが可能である。フラッシュ管理テーブル（単数または複数）をフラッシュにより頻繁に保存することによって、特に、所定のイグジット（例えば、予期しない電力損失）がなかった状況において、より高速な起動が可能となることが多い。

10

#### 【0067】

フラッシュ管理テーブルをフラッシュメモリに保存するタイミングのポリシーに関する他の実施形態も可能である。

20

フラッシュ管理テーブル（単数または複数）がフラッシュメモリに保存される場合、付随する将来情報データ構造は、次の保存が発生するまでのすべての将来イベントの処理をサポートするのに充分な情報をおそらく含んだほうがよい。余裕度を提供するとともに次の保存までに発生することが予測されるよりも多くのイベントに備えるのが望ましい。この事例に関連して、次の保存までに必要とされることが予測されるロックよりも多くの空きロックを含む、次に使用される空きロックのリストを準備してもよい。しかし、本発明は、そういう場合のリストに依存しているわけではなく、すべての記憶されている将来情報が既に使用された後で追加のイベントが受信されることに最終的になった場合でもすべてのイベントを正確に処理する。このような場合は、「保存条件」が満足されたかのように、テーブルをフラッシュに直ちに同期させるだけである。簡単にするために、このような将来情報を使い切る事例は、図3に示されていない。

30

#### 【0068】

すべての状態変化がテーブルの保存のトリガとなるわけではないので、最後に保存されたテーブルのコピーとRAMの最も更新されたテーブルのコピーとの間に、差異または不整合が最終的に生じる。テーブルが保存される度に、この差異は解消され、2つのコピー（すなわち、揮発性RAMメモリ内のコピーとフラッシュメモリ内に保存されるコピー）は一致するが、その後テーブルの保存を発生させない新しい状態の変化によって、差異が再び生じる。

#### 【0069】

40

#### 起動

いくつかの実施形態では、アプリケーションが所定のイグジットを実行する場合、更新後のフラッシュ管理テーブル（単数または複数）（すなわち、フラッシュメモリシステムの一番最近のイベントに従って更新されたフラッシュテーブル（単数または複数））がフラッシュメモリに保存される。フラッシュメモリシステムが起動すると、このような管理テーブル（単数または複数）は、そのときフラッシュメモリから検索される。

所定のイグジットがない場合は、起動時に、フラッシュメモリから揮発性メモリ内へと「古い」フラッシュ管理テーブル（単数または複数）が引き出され、その後、将来情報データ構造に記憶されている情報に従って、揮発性メモリ内の古いフラッシュ管理テーブルを更新することが可能である。

50

したがって、本発明によって提供される実施形態によって、特許文献1で実施されるような古いテーブルを無効にする必要性がなくなる。

#### 【0070】

図4は、本発明のいくつかの実施形態による例示的な起動ルーチンの流れ図である。図4は、将来情報データ構造が割り当てのために並んでいる空きブロックの順序付きリストである図2Aおよび図2Bのテーブルの以前の例に関連している例示的な起動ルーチンを提示する。電源投入(410)後、フラッシュ管理システムは、保存されているフラッシュ管理テーブルのコピーを検索する(414)。その後、フラッシュ管理システムは、フラッシュ管理テーブルがフラッシュメモリに保存されたときに保存された関連する将来情報リストを検索する(418)。

10

次に、フラッシュ管理システムは、リスト内の最初の物理ブロック番号を受け取り(422)、さらにポインタをリストの最初のエントリを指し示すように設定する(426)。

#### 【0071】

フラッシュ管理テーブルを最後に保存した直後からテーブルの状態がさらに変化する(例えば、「所定のイグジット」があったか、あるいは他の何らかの状況で)前までにシステムの電源が切れた場合、リストの最初のブロックはまだ空いているはずである。しかし、テーブルの最後の保存後からシステム停止前までにフラッシュ管理テーブル(単数または複数)の1つ以上の状態の変化が発生した場合には、リストの最初のブロックは現在使用中のはずである。

20

#### 【0072】

電源投入フラッシュ管理ソフトウェアは、物理ブロックが現在使用中か否かを判断することができる。この判断は、ブロック内に含まれる1つ以上の制御フィールドの内容をレビューすることによって、どのフラッシュ管理テーブルとも無関係に行うことができる。一部のフラッシュ管理システムでは、「対応する仮想ブロック番号」の制御フィールドを探せば充分である。すなわち、有効な仮想ブロック番号がそのフィールドに規定されていれば、ブロックは使用中であり、そのフィールドが有効な仮想ブロック番号を含まなければ、ブロックは使用中ではない。他のフラッシュ管理システムでは、物理ブロックが現在空いているか否かの判断は、単一の制御フィールドをレビューするだけでは済まないが、すべての場合において、このような判断は相対的に容易であり処理は速い。

30

したがって、将来情報リストの最初の物理ブロックの「空き」状態を判断する前述した技術を適用することによって、フラッシュ管理ソフトウェアは、検索されたテーブルとシステムの現在の状態を反映するであろう「真の」テーブルとの間に不一致があるかどうかを判断することができる。

#### 【0073】

リストの最初のブロックが割り当てられると分かれば、検索されたテーブルを更新すべきであることが分かる。次いで、最初のブロックからそのブロックの現在対応する仮想ブロック番号を検索する(434)。これによって、仮想ブロック番号と物理ブロック番号との間の現在の対応を反映するように、RAMのテーブルを更新することができる(438)。(ここでリストの最初のブロックに置き換えられたことが分かっている)その仮想ブロックに対応するように検索されたテーブルが示す物理ブロックは、現時点ではその物理ブロックは使用されていないので、テーブルによるポインタ指示はもうされていない。

40

#### 【0074】

次に、リスト内へポインタが進められ(442)、リストの次の物理ブロックに対して同様なプロセスが繰り返される。検査されるブロックが使用中であると分かる限り、RAM内のテーブルは、そのブロックを割り当てるイベントを反映するように更新される。

最終的に、検査される物理ブロックが空いていると分かるポイントに達する。これは、RAMのテーブルをこれ以上更新する必要がなく、テーブルは現在システムの状態を正確に示していることを意味する。このポイントで、更新後のテーブルはフラッシュメモリ内に記憶される(446)。同時に、新しく計算された将来情報リスト(空きブロックが割

50

り当てられるべき次のブロックを示している)もフラッシュメモリ内に記憶される。したがって、システムの電源が再び切れた場合でも、同様な電源投入ルーチンがテーブルの現在の状態をもう一度復元することができるようになっている。テーブルをフラッシュメモリに将来リストから何もブロックが割り当てられなかつたと分かる場合には記憶されているテーブルは既に最新であるため、保存するステップ(446)はそれほど必要というわけではないことに留意すべきである。

フラッシュメモリテーブル(単数または複数)が揮発性メモリおよび不揮発性メモリで更新されることが決定されると、その後はフラッシュメモリシステムの通常動作を続行すること(450)が可能である。

#### 【0075】

本発明の方法は、ある意味で、特許文献4の方法の正反対であるのは注目されることである。両方の方法には共通点が多くある。両方ともフラッシュメモリのフラッシュ管理テーブルを更新するのはごく稀なことであり、記憶されている状態と現在の状態との間で不一致が生じてしまう。両方とも、その不一致を生じさせる原因となるイベントを識別するフラッシュ内に記憶されている情報に従って、電源投入時に相互に作用する更新プロセスを「再生すること」によって、この不一致に対処している。しかし、この情報に関してこれらのことの間には基本的な違いがある。特許文献4は、これから発生するイベントがどんなイベントであっても、フラッシュ管理システムの特定のアルゴリズムに従ってイベントを処理してから、これらのイベントがたまたまどう処理されたかを示すようにフラッシュ内にイベントを記録する。本発明は、これから発生するイベントが今後どう処理されるべきかをフラッシュ内にまず記録してから、イベントが実際に発生したときに記憶されている処理決定に従ってそのイベントを処理する。

#### 【0076】

##### 性能に関する一般的な説明

本発明の特定の実施形態によって提供される1つの利点は、保存されるフラッシュ管理テーブルのコピーの読み出し、および保存されるテーブルのコピーにまだ反映されていないイベントに対するテーブルの更新が、記憶システムの多くのブロックをスキャンすることによって最初からテーブルを復元するよりも一般にかなり短い時間で済むことである。テーブルを保存した後からさらに状態が変化する(例えば、所定のイグジットがある場合、あるいは、「幸運にも」何か新しいイベントが発生する前に電源が切れる場合)前までに、システムの電源が切れた場合は、いくつかの実施形態において起動時間は高速であり、所定の電源停止が発生する場合の特許文献1の方法の起動時間と同じである。しかし、たとえテーブルの最後の保存の後でいくつかの状態の変化が発生したとしても、システムの起動時間は、所定の電源停止がなかった場合の特許文献1の方法で発生する起動時間ほど長くはないこともある。多くの場合、検出される必要があるのはほんの数個のイベントであり、それらのテーブルの状態に対する影響は再現される。これにかかる正確な時間は、一般に、将来情報データ構造から処理される必要のあるエントリの数に依存している。これはつまりテーブルが不揮発性メモリに保存される頻度に依存してもよい。頻度が高いほど、概して処理されるエントリの数は少なく、概して起動は速い。一方、保存する頻度が高いほど、将来情報データ構造を計算して準備するのにかかる時間は長い。

#### 【0077】

前述した態様において、本発明は、特許文献4の方法と性能の点で類似している。しかし、これが言えるのは、特許文献4のイベントログの書き込みに使われるオーバーヘッドが、例えば、ログが他の理由でどのみち保持される場合などのように、さらに余分なオーバーヘッドを増やさない場合だけである。そうではない場合には、犠牲を払ってログを維持する必要がもうないため、本発明の方法の使用は、特許文献4よりも優れた全体性能を提供する。

#### 【0078】

前の説明は、フラッシュ管理システム内の単一のフラッシュ管理テーブルに主に焦点を合わせているが、本発明は、それぞれシステムの状態の様々な態様を表す複数のフラッ

10

20

30

40

50

ユ管理テーブルに対しても同様に適用することができる。複数のテーブルが存在する場合、各テーブルは、テーブル独自の保存ポリシーを使用して保存され、必ずしも同じ時点で保存される必要はない。電源投入時、各フラッシュ管理テーブルは、各フラッシュ管理テーブルの個別の将来情報データを使用し、前に提示される方法を用いて復元される。2つ以上のテーブルが、これらの複数のテーブルに関連するこれから発生するイベントの処理を進めるために使用される共通の将来情報データ構造を共有することも可能である。

#### 【0079】

さらに、本発明は、前に提示され、本発明の方法を説明するために用いられる実施例に限定されないことも理解されるべきである。本発明は、他の数多くの種類のフラッシュ管理テーブルおよび将来情報データ構造にも同様に適用することでき、それらはすべて、請求される本発明の範囲に含まれるものとする。 10

#### 【0080】

##### ここで開示されているシステムに関する追加の説明

ここで開示されている技術は、ハードウェア、ファームウェアおよびソフトウェアのどんな組み合わせを使用して実施されてもよい。

1つの限定されない例では、フラッシュ管理テーブルの保存、および電源投入時のその修復および復元は、ROM160からのコードを実行するコントローラ140、さらに正確に言えば、CPU150によってすべて実施される。しかし、本発明を使用するのが可能なシステムアーキテクチャはこれだけではない。例えば、記憶モジュールがオンボードのNANDフラッシュデバイスでスタンダードアロンのコントローラが存在しない場合のように、ホストコンピュータ110で実行されるコードによって本発明の方法を実施することも可能である。別の可能性として、本発明の方法は、ホストコンピュータ110によって少なくとも部分的に実施され、さらにコントローラ140によって部分的に実施される場合がある。これらすべてのアーキテクチャおよび他の多くのアーキテクチャは、本発明の範囲に含まれている。 20

#### 【0081】

本願明細書の説明および特許請求の範囲では、「含む」、「有する」、およびその派生語の各動詞は、その動詞の目的語（単数または複数）が、動詞の主語（単数または複数）である部材、構成要素、要素または部分の必ずしも完全なリストではないことを示すために使用される。 30

本願明細書で引用されるすべての文献は、参照することによって完全に援用されている。文献の引用は、その文献が先行技術であることを承認するものではない。

本願明細書で使用される冠詞の「1つの」は、その冠詞の1つまたは2つ以上の（すなわち、少なくとも1つの）文法上の目的語を指す。例として、「1つの要素」は、1つの要素または2つ以上の要素を意味する。

本願明細書で使用される用語の「含む」は、「含むがこれに限定されるものではない」という表現を意味し、これと交換可能に使用されている。

本願明細書で使用される用語の「または」は、文脈が他の意味を明確に示していない限り、用語の「および/または」を意味し、これと交換可能に使用されている。

本願明細書で使用される用語の「などの」は、「などであるがこれに限定されるものではない」という表現を意味し、これと交換可能に使用されている。 40

#### 【0082】

本発明は、例として提供され本発明の範囲を限定することを意図しない本発明の実施形態の詳細な説明を用いて説明された。記載されている実施形態は様々な特徴を含むが、そのすべてが本発明のすべての実施形態で必要なわけではない。本発明の一部の実施形態は、その特徴の一部、またはその特徴の可能な組み合わせのみを利用している。当業者であれば、記載されている本発明の実施形態の変形、および記載されている実施形態に述べられる特徴の様々な組み合わせを含む本発明の実施形態が思い当たるはずである。

本発明は、限定された数の実施形態に関して説明されたが、当然のことながら、本発明の多くの変形、改変および他の適用がなされてもよい。 50

【図1】

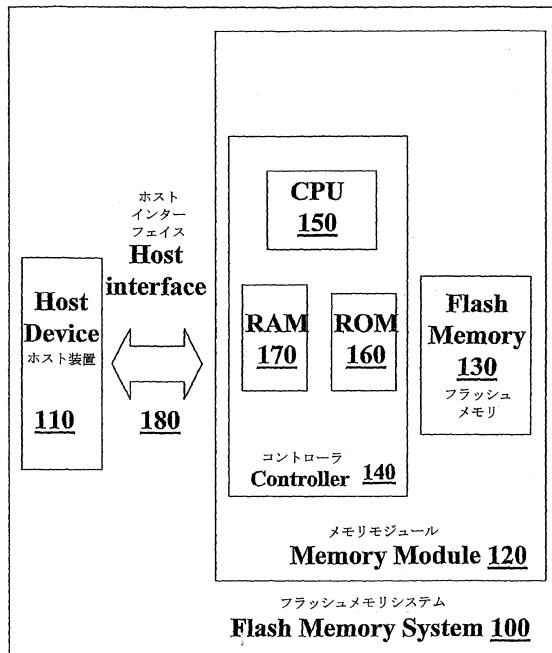


FIG.1

【図2A】

$t = t_1$  の場合  
For  $t=t_1$

変換テーブル  
Translation Table 200

Virtual Block	Physical Block
0	893
1	485
2	172
...	...
1022	1
1023	985

FIG.2A

【図2B】

$t = t_2$  の場合、ただし  $t_2 > t_1$   
For  $t=t_2$  where  $t_2 > t_1$

変換テーブル  
Translation Table 200

Virtual Block	Physical Block
0	893
1	485
2	777
...	...
1022	1
1023	985

FIG.2B

【図3】

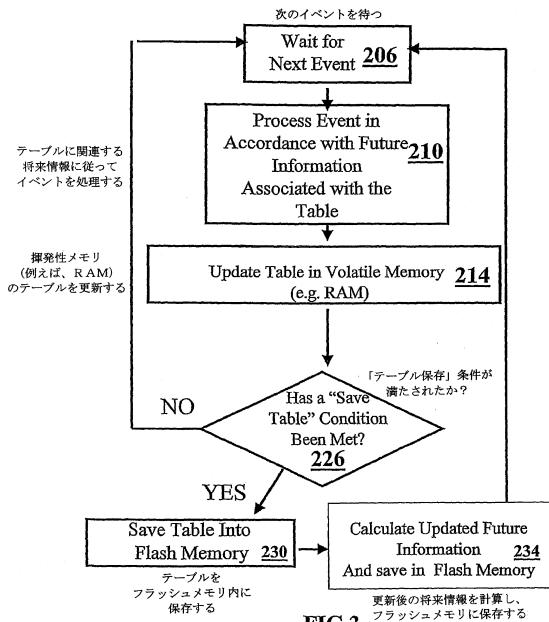
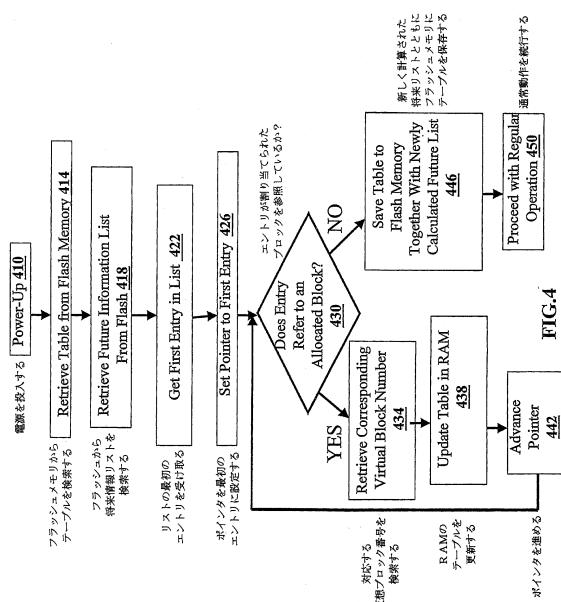


FIG.3

【図4】



【 四 5 】

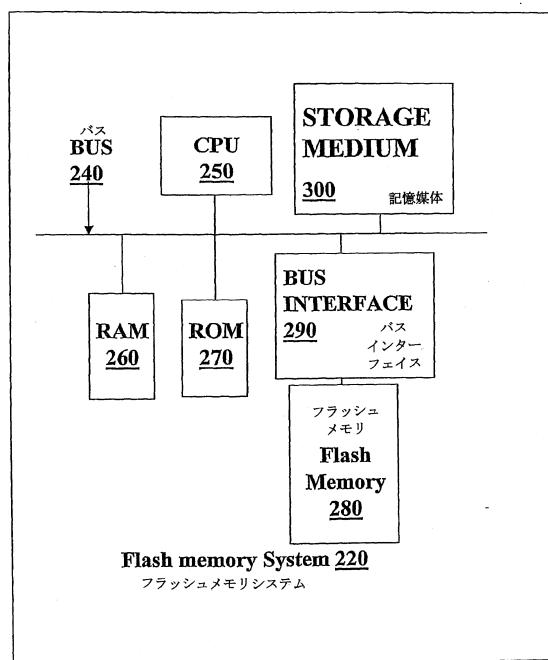
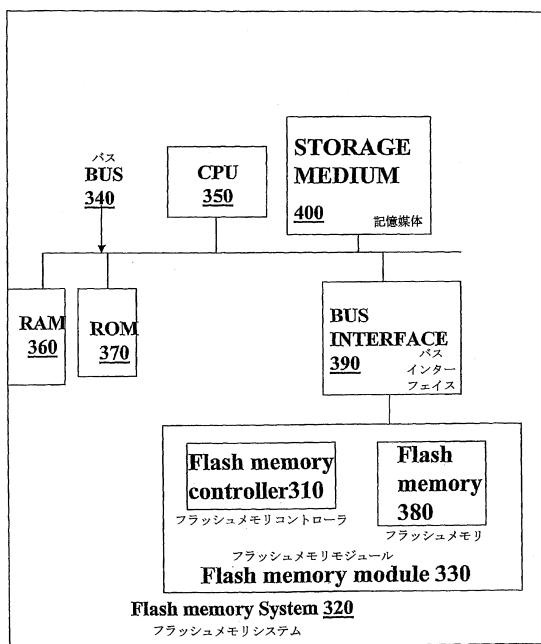


FIG.5

【 四 6 】



---

フロントページの続き

(31)優先権主張番号 11/808,452

(32)優先日 平成19年6月11日(2007.6.11)

(33)優先権主張国 米国(US)

(56)参考文献 国際公開第2006/120679(WO, A1)

国際公開第99/030239(WO, A1)

特開平08-194640(JP, A)

特開平07-134674(JP, A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 12/00-12/06

G 06 F 12/16