

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7010809号

(P7010809)

(45)発行日 令和4年1月26日(2022.1.26)

(24)登録日 令和4年1月17日(2022.1.17)

(51)国際特許分類

F I

G 0 6 F 12/0866(2016.01)

G 0 6 F 12/0866 1 0 0

請求項の数 21 (全28頁)

(21)出願番号	特願2018-234370(P2018-234370)	(73)特許権者	390019839
(22)出願日	平成30年12月14日(2018.12.14)		三星電子株式会社
(65)公開番号	特開2019-128949(P2019-128949 A)		S a m s u n g E l e c t r o n i c s C o . , L t d .
(43)公開日	令和1年8月1日(2019.8.1)		大韓民国京畿道水原市靈通区三星路129
審査請求日	令和3年12月6日(2021.12.6)		129, S a m s u n g - r o , Y e o n g t o n g - g u , S u w o n - s i , G y e o n g g i - d o , R e p u b l i c o f K o r e a
(31)優先権主張番号	62/619,723		
(32)優先日	平成30年1月19日(2018.1.19)	(74)代理人	110000051
(33)優先権主張国・地域又は機関	米国(US)		特許業務法人共生国際特許事務所
(31)優先権主張番号	15/934,940	(72)発明者	張 牧 天
(32)優先日	平成30年3月23日(2018.3.23)		アメリカ合衆国, 95051, カリフォルニア州, サンタクララ, ピアト
(33)優先権主張国・地域又は機関	米国(US)		最終頁に続く

(54)【発明の名称】 デデュープ可能なメモリキャッシュ及びそのための動作方法

(57)【特許請求の範囲】

【請求項1】

デデュープ可能な(d e d u p a b l e)リードキャッシュ及び非デデュープ可能な(n o n - d e d u p a b l e)ライトバッファを含むキャッシュメモリと、
前記デデュープ可能なリードキャッシュを利用してデータのリード及びライトを管理し、
前記デデュープ可能なリードキャッシュへのライト要請が成功したか否かを示すライト状態信号を伝送するように動作するデデュープエンジンと、
キャッシュコントローラと、を備え、
前記キャッシュコントローラは、
要請に含まれるアドレスが前記デデュープ可能なリードキャッシュで発見されるか否かをチェックするキャッシュヒット/ミスチェックロジックと、
前記アドレスが前記デデュープ可能なリードキャッシュで発見されることを前記キャッシュヒット/ミスチェックロジックが示す場合、前記キャッシュメモリからの第1データにアクセスするヒットブロックと、
前記アドレスが前記デデュープ可能なリードキャッシュで発見されないことを前記キャッシュヒット/ミスチェックロジックが示す場合、バックエンド大容量メモリからの第2データにアクセスするミスブロックと、
前記デデュープ可能なリードキャッシュ内の前記第1データへのアクセスに関する情報を格納するヒストリーストレージと、を含むことを特徴とするデデュープ可能なキャッシュ。

【請求項2】

前記デデュープ可能なリードキャッシュは、前記デデュープ可能なリードキャッシュに物理的に適合するものよりも多くのキャッシュラインを格納することをシミュレーションすることを特徴とする請求項 1 に記載のデデュープ可能なキャッシュ。

【請求項 3】

前記キャッシュメモリは、前記デデュープ可能なリードキャッシュ内のキャッシュラインに関する情報を格納するメタデータ領域を更に含むことを特徴とする請求項 1 に記載のデデュープ可能なキャッシュ。

【請求項 4】

前記キャッシュコントローラは、データをライトするための第 1 ライト要請をプロセッサから受信して前記非デデュープ可能なライトバッファに前記データを格納するように動作し、

10

前記ヒットブロックは、前記デデュープ可能なリードキャッシュ内のキャッシュラインが前記第 1 ライト要請によって修正される場合、前記キャッシュラインを無効化させるように前記デデュープエンジンに第 2 ライト要請を伝送するように動作することを特徴とする請求項 1 に記載のデデュープ可能なキャッシュ。

【請求項 5】

前記第 2 ライト要請は、前記デデュープ可能なリードキャッシュ内のキャッシュラインに 0 の値をライトする要請を含むことを特徴とする請求項 4 に記載のデデュープ可能なキャッシュ。

【請求項 6】

20

前記キャッシュコントローラは、前記キャッシュメモリのメタデータ領域に前記キャッシュラインが有効でないとしてマークするように更に動作することを特徴とする請求項 4 に記載のデデュープ可能なキャッシュ。

【請求項 7】

前記ヒットブロックは、前記第 2 ライト要請に応答して前記デデュープエンジンから非確認応答 (N A K) 信号を受信し、前記デデュープ可能なリードキャッシュから第 2 キャッシュラインを除去し、前記デデュープ可能なリードキャッシュから前記第 2 キャッシュラインを除去した後に前記デデュープエンジンに前記第 2 ライト要請を再び伝送するように更に動作することを特徴とする請求項 4 に記載のデデュープ可能なキャッシュ。

【請求項 8】

30

前記キャッシュコントローラは、データをリードするためのリード要請をプロセッサから受信して前記プロセッサに前記データを伝送するように動作し、
前記ミスブロックは、前記バックエンド大容量メモリから前記データをリトリブして、前記デデュープ可能なリードキャッシュ内に前記データをライトするために前記デデュープエンジンにライト要請を伝送するように動作することを特徴とする請求項 1 に記載のデデュープ可能なキャッシュ。

【請求項 9】

前記ミスブロックは、前記ライト要請に応答して前記デデュープエンジンから非確認応答 (N A K) 信号を受信し、前記デデュープ可能なリードキャッシュからキャッシュラインを除去し、前記デデュープ可能なリードキャッシュから前記キャッシュラインを除去した後に前記デデュープエンジンに前記ライト要請を再び伝送するように更に動作することを特徴とする請求項 8 に記載のデデュープ可能なキャッシュ。

40

【請求項 10】

前記デデュープ可能なリードキャッシュは、デデュープメモリを含み、
前記デデュープメモリ内のデータは、デデュープの対象になることを特徴とする請求項 1 に記載のデデュープ可能なキャッシュ。

【請求項 11】

キャッシュコントローラの動作方法であって、
データをライトするためのライト要請を受信するステップと、
前記データがデデュープ可能な (d e d u p a b l e) リードキャッシュ内のキャッシュ

50

ラインに存在すると判定するステップと、
前記デデュープ可能なリードキャッシュ内のキャッシュラインを無効化するステップと、
前記データを非デデュープ可能な (n o n - d e d u p a b l e) ライトバッファに格納するステップと、を有し、
前記デデュープ可能なリードキャッシュは、キャッシュメモリ内の第 1 領域であり、
前記キャッシュメモリは、第 2 領域として前記非デデュープ可能なライトバッファを含むことを特徴とする方法。

【請求項 1 2】

前記デデュープ可能なリードキャッシュ内のキャッシュラインを無効化するステップは、
前記キャッシュメモリ内のメタデータ領域に前記キャッシュラインが有効でないとしてマ
ークするステップと、
デデュープエンジンを介して前記キャッシュラインに 0 の値をライトするステップと、を
含むことを特徴とする請求項 1 1 に記載の方法。

10

【請求項 1 3】

前記デデュープ可能なリードキャッシュ内のキャッシュラインを無効化するステップは、
前記デデュープエンジンからライト状態信号の確認応答 (A C K) を受信するステップを
更に含むことを特徴とする請求項 1 2 に記載の方法。

【請求項 1 4】

前記デデュープ可能なリードキャッシュ内のキャッシュラインを無効化するステップは、
前記デデュープエンジンからライト状態信号の非確認応答 (N A K) を受信するステップ
と、
前記デデュープ可能なリードキャッシュから除去するために第 2 キャッシュラインを選択
するステップと、
前記第 2 キャッシュラインを無効化するステップと、
前記デデュープエンジンを介して前記第 2 キャッシュラインに 0 の値をライトするステッ
プと、を更に含むことを特徴とする請求項 1 2 に記載の方法。

20

【請求項 1 5】

前記データを非デデュープ可能なライトバッファに格納するステップは、前記データが前
記デデュープ可能なリードキャッシュ内のキャッシュラインに存在するか否かに拘らず、
前記データを前記非デデュープ可能なライトバッファに格納するステップを含むことを特
徴とする請求項 1 1 に記載の方法。

30

【請求項 1 6】

キャッシュコントローラの動作方法であって、
データをリードするためのリード要請を受信するステップと、
前記データがデデュープ可能な (d e d u p a b l e) リードキャッシュ内の複数のキャ
ッシュラインに存在しないと判定するステップと、
バックエンド大容量メモリから前記データをリードするステップと、
前記デデュープ可能なリードキャッシュ内の第 1 キャッシュラインを選択するステップと、
前記データを前記第 1 キャッシュラインにライトすることを試みるためにデデュープエン
ジンに前記データを提供するステップと、
前記リード要請に応答して前記データを伝送するステップと、を有し、
前記デデュープ可能なリードキャッシュは、キャッシュメモリ内の第 1 領域であり、
前記キャッシュメモリは、第 2 領域として非デデュープ可能な (n o n - d e d u p a b
l e) ライトバッファを含むことを特徴とする方法。

40

【請求項 1 7】

前記デデュープ可能なリードキャッシュ内の第 1 キャッシュラインを選択するステップは
、現在データを格納していない前記デデュープ可能なリードキャッシュ内の第 1 キャッシ
ュラインを選択するステップを含むことを特徴とする請求項 1 6 に記載の方法。

【請求項 1 8】

前記デデュープ可能なリードキャッシュ内の第 1 キャッシュラインを選択するステップは、

50

前記キャッシュメモリ内のメタデータ領域からのデデューブ可能なリードキャッシュメタデータ及びヒストリーデータにตอบสนองして、除去するために現在有効なデータを格納している前記デデューブ可能なリードキャッシュ内の第 1 キャッシュラインを選択するステップと、

前記第 1 キャッシュラインを無効化するステップと、を含むことを特徴とする請求項 1.6 に記載の方法。

【請求項 19】

前記第 1 キャッシュラインを無効化するステップは、

前記キャッシュメモリ内のメタデータ領域に前記第 1 キャッシュラインを有効でないとしてマークするステップと、

前記デデューブエンジンを介して前記第 1 キャッシュラインに 0 の値をライトするステップと、を含むことを特徴とする請求項 1.8 に記載の方法。

【請求項 20】

前記データを前記第 1 キャッシュラインにライトすることを試みるためにデデューブエンジンに前記データを提供するステップは、前記デデューブエンジンからライト状態信号の確認応答 (ACK) を受信するステップを含むことを特徴とする請求項 1.6 に記載の方法。

【請求項 21】

前記データを前記第 1 キャッシュラインにライトすることを試みるためにデデューブエンジンに前記データを提供するステップは、

前記デデューブエンジンからライト状態信号の非確認応答 (NAK) を受信するステップと、

前記キャッシュメモリ内のメタデータ領域からのデデューブ可能なリードキャッシュメタデータ及びヒストリーデータにตอบสนองして、除去するために現在有効なデータを格納している前記デデューブ可能なリードキャッシュから第 2 キャッシュラインを選択するステップと、

前記第 2 キャッシュラインを無効化するステップと、

前記データを前記選択された第 2 キャッシュラインにライトすることを試みるために前記デデューブエンジンに前記データを提供するステップと、を含むことを特徴とする請求項 1.6 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データキャッシングに関し、より詳細には、デデューブ可能なメモリキャッシュ及びそのための動作方法に関する。

【背景技術】

【0002】

デデューブ (重複排除: De-duplication) 可能な (dedupable) メモリは、システムにおいて「利用可能な」メモリを増加させる方式を提供する。重複データを検出して一つのデータのコピーを格納することにより、重複データのコピーを格納するために用いられるストレージは、他のデータを格納するために用いられる。特に、同じデータ値へのアクセスを要請する多様なアプリケーションは、たとえ要請が他のアドレスを利用していたとしても、全てが同じ物理アドレスに連結される。二個、三個、又はより多くの論理アドレスが同じ物理アドレスにマッピングされることから、追加的データのコピーに必要な可能性があるストレージは、他のデータを格納するために用いられ、メモリが物理的に格納可能なものよりもより多くの全体のデータを格納するように見える。

【0003】

しかし、デデューブ可能なメモリも自体的な問題をもたらす。どのアドレスが同じデータを示すのかを管理することは難しく、リード (read) 及びライト (write) のようなデータアクセスが他の形態のストレージに比して遅い可能性がある。相対的にアクセスの遅いデータを利用する場合 (ハードディスクドライブのような)、必ずしも大きな困

10

20

30

40

50

難性があるわけではないが、より速いストレージデバイスに適用される場合、このような遅延は相当な制約になる。ストレージデバイスが一般的に速く動作する程、データ重複の全体的な影響は更に大きくなる。例えば、ダイナミックランダムアクセスメモリ (D R A M : D y n a m i c R a n d o m A c c e s s M e m o r y) のアクセスは、ソリッドステートディスク (S S D : S o l i d S t a t e D i s k) 又は他の形態のフラッシュメモリよりもデータ重複に関連する遅延に対してはるかに敏感になり、これはハードディスクドライブよりも遅延に対してより敏感である。

【 0 0 0 4 】

キャッシュにおいて、デデュープ可能なメモリを利用する方式は、バックエンド (b a c k e n d) メモリのデータに対するアクセス速度を高くするために必要な可能性がある。

10

【先行技術文献】

【特許文献】

【 0 0 0 5 】

【文献】米国特許第 9 1 1 6 8 1 2 号明細書

米国特許第 9 3 0 4 9 1 4 号明細書

米国特許第 9 3 9 0 1 1 6 号明細書

米国特許第 9 4 0 1 9 6 7 号明細書

米国特許出願公開第 2 0 1 5 / 0 0 7 4 3 3 9 号明細書

米国特許出願公開第 2 0 1 6 / 0 2 2 4 5 8 8 号明細書

【発明の概要】

20

【発明が解決しようとする課題】

【 0 0 0 6 】

本発明は、上記従来の問題点に鑑みてなされたものであって、本発明の目的は、デデュープ可能なメモリを利用するキャッシュ及びそのための動作方法を提供することにある。

【課題を解決するための手段】

【 0 0 0 7 】

上記目的を達成するためになされた本発明の一態様によるデデュープ可能なキャッシュは、デデュープ可能な (d e d u p a b l e) リードキャッシュ (c a c h e) 及び非デデュープ可能な (n o n - d e d u p a b l e) ライトバッファ (b u f f e r) を含むキャッシュメモリと、前記デデュープ可能なリードキャッシュを利用してデータのリード及びライトを管理し、前記デデュープ可能なリードキャッシュへのライト要請が成功したか否かを示すライト状態信号を伝送するデデュープエンジンと、キャッシュコントローラと、を備え、前記キャッシュコントローラは、要請に含まれるアドレスが前記デデュープ可能なリードキャッシュで発見されるか否かをチェックするキャッシュヒット (h i t) / ミス (m i s s) チェックロジックと、前記アドレスが前記デデュープ可能なリードキャッシュで発見されることを前記キャッシュヒット / ミスチェックロジックが示す場合、前記キャッシュメモリの第 1 データにアクセスするヒットブロックと、前記アドレスが前記デデュープ可能なリードキャッシュで発見されることを前記キャッシュヒット / ミスチェックロジックが示す場合、バックエンド (b a c k e n d) 大容量メモリの第 2 データにアクセスするミスブロックと、前記デデュープ可能なリードキャッシュの前記第 1 データへのアクセスに関する情報を格納するヒストリー (h i s t o r y) ストレージと、を含む。

30

40

【 0 0 0 8 】

上記目的を達成するためになされた本発明の一態様によるキャッシュコントローラの動作方法は、データをライト (w r i t e) するためのライト要請を受信するステップと、前記データがデデュープ可能な (d e d u p a b l e) リードキャッシュのキャッシュラインに存在すると判断するステップと、前記デデュープ可能なリードキャッシュの前記キャッシュラインを無効化するステップと、前記データを非デデュープ可能な (n o n - d e d u p a b l e) ライトバッファに格納するステップと、を有し、前記デデュープ可能なリードキャッシュは、キャッシュメモリの第 1 領域であり、前記キャッシュメモリは、第

50

２領域として前記非デデューブ可能なライトバッファを含む。

【０００９】

上記目的を達成するためになされた本発明の他の態様によるキャッシュコントローラの動作方法は、データをリード(read)するためのリード要請を受信するステップと、前記データがデデューブ可能な(dedupable)リードキャッシュの複数のキャッシュラインに存在しないと判断するステップと、バックエンド大容量メモリから前記データをリードするステップと、前記デデューブ可能なリードキャッシュの第１キャッシュラインを選択するステップと、前記データを前記第１キャッシュラインにライトすることを試みるためにデデューブエンジンに前記データを提供するステップと、前記リード要請に回答して前記データを伝送するステップと、を有し、前記デデューブ可能なリードキャッシュは、キャッシュメモリの第１領域であり、前記キャッシュメモリは、第２領域として非デデューブ可能なライトバッファを含む。

10

【発明の効果】

【００１０】

本発明のキャッシュによれば、より多くのデータを同じ物理的な大きさのメモリに格納することができる。

【００１１】

また、本発明のキャッシュによれば、デデューブ可能なメモリへのライト固有の遅延を防止することができる。

【図面の簡単な説明】

20

【００１２】

【図１】本発明の一実施形態によるデデューブ可能なキャッシュを使用するマシンを示す図である。

【図２】図１のマシンの追加的な詳細事項を示す図である。

【図３】図１のデデューブキャッシュの例示的なレイアウトを示す図である。

【図４】デデューブエンジンを有する図１のデデューブキャッシュを示す図である。

【図５】図３及び図４のキャッシュコントローラの詳細事項を示す図である。

【図６Ａ】本発明の一実施形態による図１のデデューブキャッシュがライト要請を処理する例示的な手順のフローチャートである。

【図６Ｂ】本発明の一実施形態による図１のデデューブキャッシュがライト要請を処理する例示的な手順のフローチャートである。

30

【図７】本発明の一実施形態による図１のデデューブキャッシュが図３のデデューブ可能なリードキャッシュのキャッシュラインを無効化するための例示的な手順のフローチャートである。

【図８Ａ】本発明の一実施形態による図１のデデューブキャッシュがリード要請を処理する例示的な手順のフローチャートである。

【図８Ｂ】本発明の一実施形態による図１のデデューブキャッシュがリード要請を処理する例示的な手順のフローチャートである。

【図８Ｃ】本発明の一実施形態による図１のデデューブキャッシュがリード要請を処理する例示的な手順のフローチャートである。

40

【発明を実施するための形態】

【００１３】

以下、本発明を実施するための形態の具体例を、図面を参照しながら詳細に説明する。後述する詳細な説明で、発明の完璧な理解を可能にするために、多様な特定の詳細事項を記載する。しかし、本技術分野における通常の技術者がこのような特定の詳細事項なしに発明を実施することは自明である。他の場合において、よく知られている方法、手順、構成要素、回路、そしてネットワークは実施形態を不必要に曖昧にしないために詳細に説明しない。

【００１４】

第１、第２などの用語は、本明細書で多様な要素を説明するために使用されとしても、

50

このような要素は、これらの用語によって限定されないことが理解される。これらの用語は、一つの要素を他の要素と区別するために使用される。例えば、本発明の権利範囲を逸脱せずに、第 1 モジュールは、第 2 モジュールと称され、第 2 モジュールは、第 1 モジュールと称される。

【 0 0 1 5 】

本明細書において、発明の詳細な説明で使用する用語は、単に特定の実施形態を説明するための目的であり、本発明を制限しようとする意図ではない。発明の説明及び請求項で使用するように、文脈が明らかに他のことを示していない限り、単数形の使用は複数形を含むように意図する。本明細書で「及び／又は」という用語は、列挙された項目に関連する一つ以上の可能な組み合わせを全て示すものと理解される。「含む (c o m p r i s e s) 」及び／又は「含む (c o m p r i s i n g) 」という用語は、本明細書で使用する場合、記述する特徴、整数、ステップ、動作、要素、及び／又は構成要素の存在を明示するものと理解される。しかし、一つ以上の他の特徴、整数、ステップ、動作、要素、構成要素、及び／又はそれらのグループの追加又は存在を排除するものではない。図面の構成要素及び特徴は、必ずしも縮尺に合わせて描かれていない。

10

【 0 0 1 6 】

デデューブダイナミックランダムアクセスメモリ (D R A M) は、所定の D R A M ユニット内で向上した論理的容量を提供するが、一般の D R A M の動作よりも一般的に遅い。この遅いパフォーマンスにより、デデューブ D R A M を一般の D R A M キャッシュとしてすることは使用できない。

20

【 0 0 1 7 】

このような懸念を解決するために、デデューブ D R A M キャッシュのメモリを、デデューブ可能な D R A M リード (r e a d) キャッシュ、及び非デデューブ可能な (言い換えると、デデューブ不可能な) ライト (w r i t e) バッファに分ける。デデューブ D R A M へのライトは遅いため、既存の D R A M がライトバッファ (W B) に用いられる。リード要請は、デデューブ可能なリードキャッシュ (R C) から (殆ど) 提供されるが、ライトは通常の D R A M ライトバッファ (W B) によって提供される。

【 0 0 1 8 】

例えば、物理的な D R A M が 8 ギガバイト (G B) である場合、物理的な D R A M は 4 G B の物理的なデデューブ D R A M R C (リードキャッシュ) (総 8 G B の仮想容量を提供する) 及び 4 G B の物理的な D R A M の W B (ライトバッファ) に分割される。デデューブエンジンは、単にデデューブ D R A M の R C によって利用される特定の 4 G B の範囲についてデデューブメカニズム (m e c h a n i s m) を動作させる。

30

【 0 0 1 9 】

しかし、デデューブ D R A M がリードキャッシュとして使用されるときも、ライトはデデューブ D R A M に相変わらず影響を与える。例えば、ホスト (h o s t) からの 6 4 バイト (B) のアップデートは、デデューブ D R A M の R C に格納されたデータに影響を与えるか (ライトヒット (h i t)) 、或いはデータが現在のデデューブ D R A M R C にない場合 (リードミス (m i s s)) 、バックエンド大容量のメモリからの 2 キロバイト (K B) が必要になる。デデューブ D R A M の R C にライトする状況を解決するために、新たな W R _ S t a t u s 信号が用いられる。即ち、ライトが成功した場合、デデューブエンジンは A C K (確認応答、又は肯定応答) を返し、そうでない場合、デデューブエンジンは N A K (非確認応答、又は否定応答) を返す。キャッシュコントローラが N A K を受信すると、キャッシュコントローラは、キャッシュフィル (c a c h e f i l l) をキャンセル (c a n c e l) する (即ち、2 K B フィル (f i l l) はキャッシュされない) 。

40

【 0 0 2 0 】

従って、デデューブ D R A M の R C ライトヒット処理のために、デデューブ D R A M キャッシュは、キャッシュライン (c a c h e l i n e) が有効でないように R C のメタデータ (m e t a d a t a) をアップデートし、6 4 B の 0 (キャッシュラインを有効しな

50

いようにする)をデデューブエンジンにライトする(これはデデューブDRAMのRCをガーベジコレクション(garbage collection)することに関連付ける)。このような処理が完了した後、デデューブDRAMキャッシュは、デデューブDRAMのRCライトヒットがなかったかのように進行する(即ち、デデューブDRAMキャッシュは、デデューブDRAMのRCライトミスがあったかのように進行する)。

【0021】

デデューブDRAMのRCフィル(fill)を処理するために、デデューブDRAMキャッシュは、デデューブDRAMのRCから削除されるキャッシュラインを選択するためにヒストリー及びRCメタデータをリードする。選択されたキャッシュラインが有効なデータを含む場合、デデューブDRAMキャッシュは、キャッシュラインを無効としてマークするためにRCのメタデータをアップデートし、キャッシュラインが有効でないように2KBの0をデデューブエンジンにライトする。そして(選択されたキャッシュラインが有効でない場合)、デデューブDRAMキャッシュはデデューブエンジンに新たなキャッシュラインのデータをライトし、デデューブエンジンはWR_Status信号を返す。デデューブエンジンがACKを返すと、デデューブDRAMキャッシュはキャッシュラインを有効としてマークするためにRCのメタデータをアップデートする。そうでない場合、デデューブエンジンがNAKを返すと、デデューブDRAMキャッシュは、デデューブDRAMのRCにデータをライトするための要請をドロップ(drop)する。

【0022】

上述した本発明は、デデューブ可能なメモリを使用して潜在的に低いライトパフォーマンスの問題を解決するために有効的である。しかし、デデューブ可能なメモリの使用は他の問題を引き起こす。即ち、それはライト動作が保証されないということである。例えば、デデューブDRAMのRCフィル動作が遂行されないが、デデューブDRAMのRCの全てのキャッシュラインは有効なデータを格納するものと仮定する。無効化のために選択されたキャッシュラインがデデューブされたデータを示す場合、即ちキャッシュラインによって示されたデータが他のキャッシュラインによって示される場合、キャッシュラインを無効化することは任意の物理的メモリを確保しないはずである。或いは、データが他のキャッシュラインによって使用されている場合、デデューブRCのライトヒット時、デデューブエンジンがキャッシュラインを0で上書きすることができない。このようなライト完了の保証不足の問題に対する二つの解決策がある。即ち、ライトが保証されないことを受け入れるか、又はライト動作の完了を保証するためにライト動作を再び試みることである。

【0023】

ライト動作を再び試みることを達成するために、ライト再試みが遂行される。ライトが失敗した場合、即ちデデューブエンジンが試みられたライトからNAK信号を返す場合、他のキャッシュラインが無効化のために選択され(デデューブDRAMキャッシュの具現に応じてライトバック(write back)が含まれる)、ライトが再び試みられる。

【0024】

他のキャッシュラインを無効化すると、デデューブエンジンはデデューブDRAMのRCへのライト動作を完了するため、ライト再試みが成功する。しかし、ライト再試みの部分として、無効のために選択されたキャッシュラインもまたデデューブの対象データを指し示す場合、ライト再試みが再び失敗する。従って、キャッシュラインを無効化してライトを再び試みるプロセスは、所望の何回かが繰り返し遂行される。一般的に、再試みのプロセスに対する終了条件は、成功的なライト(デデューブエンジンからのACK信号により示されるように)、一セットの全てのキャッシュラインの排除、又は予め決定された数のライト再試みである。全てのキャッシュラインが排除されるか、又は予め決められた数のライト再試みの中の一つが発生した場合、デデューブDRAMキャッシュは、所望のライト動作を成功的に完了せずに、単にプロセッサに結果を返すだけでよい。

【0025】

図1は、本発明の一実施形態によるデデューブキャッシュを使用するマシン(machine)を示す図である。図1には、マシン105を示す。マシン105は、プロセッサ1

10

20

30

40

50

110を含む。プロセッサ110は、任意の多様なプロセッサである。例えば、プロセッサ110は、インテルジーオン(Intel Xeon(登録商標))、セレロン(Celeron(登録商標))、アイテニウム(Itanium(登録商標))、又はアトム(Atom(登録商標))プロセッサ、エーエムディーオプテロン(AMD Opteron(登録商標))プロセッサ、アーム(ARM(登録商標))プロセッサなどである。図1は、マシン105内に、一つのプロセッサ110を示しているが、マシン105は、任意の数のプロセッサを含み、それぞれのプロセッサは、シングルコア又はマルチコアプロセッサであり、任意の所望の組み合わせで混合される。プロセッサ110は、デバイスドライバ(driver)115を実行し、デバイスドライバ115は、ストレージデバイス120へのアクセスをサポートする。他のデバイスドライバは、マシン105の他の構成要素へのアクセスをサポートする。

10

【0026】

マシン105は、メモリコントローラ125を含み、メモリコントローラ125は、メインメモリ130へのアクセスを管理するために使用される。メモリ130は、フラッシュメモリ、Dynamic Random Access Memory(DRAM)、Static Random Access Memory(SRAM)、Persistent Random Access Memory(PRAM)、Ferroelectric Random Access Memory(FeRAM)、Non-Volatile Random Access Memory(NVRAM)、Magnetoresistive Random Access Memory(MRAM)などのように、任意の多様なメモリである。メモリ130は、他のメモリタイプの任意の所望の組み合わせである。マシン105は、以下に説明するようにデデュープキャッシュ135を更に含む。

20

【0027】

図1は、サーバ(サーバは、スタンドアロン(standalone)又はラック(rack)サーバの可能性あり)としてマシン105を図示するが、本発明の実施例は、制限なしに任意の所望のタイプのマシン105を含む。例えば、マシン105は、デスクトップ又はラップトップコンピュータ、或いは本発明の実施例から得られる任意の他のデバイスに代替される。マシン105は、特化された携帯コンピューティングデバイス、タブレットコンピュータ、スマートフォン、及び他のコンピューティングデバイスを含む。

【0028】

30

図2は、図1のマシン105の追加の詳細事項を示す図である。図2において、一般的に、マシン105は一つ以上のプロセッサ110を含み、プロセッサ110はメモリコントローラ125及びクロック205を含み、クロック205はマシン105の構成要素の動作を調整するために利用される。プロセッサ110はメモリ130に連結され、メモリ130は、例としてRAM(random access memory)、ROM(read-only memory)、又は他の保存メディアを含む。プロセッサ110はストレージデバイス120及びネットワークコネクタ(connector)210に連結され、ネットワークコネクタ210は、例えば、イーサネット(Ethernet(登録商標))コネクタ又は無線コネクタである。プロセッサ110はバス215に連結され、バス215には、他の構成要素の中のユーザーインターフェース220及び入出力(Input/Output)エンジン225を利用して管理される入出力ポートが付着される。

40

【0029】

図3は、図1のデデュープキャッシュ135の例示的なレイアウトを示す図である。図3において、デデュープキャッシュ135は、4つの一般的な構成要素に分割される。即ち、DRAM305、キャッシュコントローラ310、ホスト層(layer)315、メディア層320である。DRAM305は、デデュープキャッシュ135に対して、実際のメモリとして動作し、三つの領域に分割される。即ち、デデュープ可能なリードキャッシュ325、非デデュープ可能なライトバッファ330、メタデータ領域335である。デデュープ可能なリードキャッシュ325は、プロセッサ110によって、デデュープキ

50

キャッシュ 135 からリードされるデータを格納するためのデデューブメモリとして使用される。非デデューブ可能なライトバッファ 330 は、プロセッサ 110 からライトされるデータを格納するための従来の（即ち、非デデューブ）メモリとして使用される。メタデータ領域 335 は、デデューブ可能なリードキャッシュ 325 及び非デデューブ可能なライトバッファ 330 のキャッシュライン（*cache lines*）に関する情報を格納する。例えば、メタデータ領域 335 は、どのようなキャッシュラインが有効であるか否か、及びどのようなキャッシュラインがバックエンド大容量メモリ 130 にライトするためにバッファリング（*buffering*）されるかを格納する。図 3 には、デデューブエンジンを示していないが、デデューブエンジンは、デデューブ可能なリードキャッシュ 325 に対する実際のデータのライトを管理する。即ち、デデューブエンジンは、図 4 を参照して以下で説明される。デデューブ可能なリードキャッシュ 325、非デデューブ可能なライトバッファ 330、及びメタデータ領域 335 は、DRAM 305 の任意の所望の割合を含む。例えば、DRAM 305 が合計 8 ギガバイト（GB）を含む場合、デデューブ可能なリードキャッシュ 325 お及び非デデューブ可能なライトバッファ 330 は、約 4 GB のストレージをそれぞれ含む（メタデータ領域 335 は、相対的に DRAM 305 の少ない量を要求する）。予想されるデデューブ率を考慮すると、デデューブ可能なリードキャッシュ 325 は、物理容量よりも大きな容量をシミュレーションする。例えば、デデューブ可能なリードキャッシュ 325 が、4 GB の物理メモリを含み、2.0 のターゲットデデューブ率を有する場合、デデューブ可能なリードキャッシュ 325 は、8 GB の仮想メモリをシミュレーションする。デデューブキャッシュ 135 は、デデューブ可能なリードキャッシュ 325 の仮想容量だけキャッシュラインをサポートし、これはデデューブ可能なリードキャッシュ 325 の物理的な容量でサポート可能なキャッシュラインの数を超える。

【0030】

図 3 は、デデューブ可能なリードキャッシュ 325 及び非デデューブ可能なライトバッファ 330 の両方を含む DRAM 305 を図示してデデューブ可能なリードキャッシュ 325 及び非デデューブ可能なライトバッファ 330 の異なる機能を提供するが、一部の機能は重複する。例えば、本発明のいくつかの実施例において、非デデューブ可能なライトバッファ 330 は、プロセッサ 110 からライトされたデータを格納するが（そして最終的には、バックエンド大容量メモリ 130 にライトされる可能性あり）、非デデューブ可能なライトバッファ 330 は、「リードキャッシュ」として動作し（但し、デデューブ対象ではないとしても）、非デデューブ可能なライトバッファ 330 からデータがリードされる。従って、非デデューブ可能なライトバッファ 330 は、バックエンド大容量メモリ 130 にライトされるデータのための「一時的な（*transient*）」ストレージ以上と見なされる。このような本発明の実施例において、特定のキャッシュラインに対してデデューブ可能なリードキャッシュ 325 をチェックすることへの任意の言及は、非デデューブ可能なライトバッファ 330 からのキャッシュラインのチェック及びデデューブ可能なリードキャッシュ 325 又は非デデューブ可能なライトバッファ 330 から適切にデータをリードすることで理解され、非デデューブ可能なライトバッファ 330 が一杯になるか非デデューブ可能なライトバッファ 330 のキャッシュラインが新たなキャッシュラインのためのスペースを作るために無効化される場合にのみ、データはバックエンド大容量メモリ 130 にライトされる。このような本発明の実施例において、データがデデューブ可能なリードキャッシュ 325 又は非デデューブ可能なライトバッファ 330 に格納されない場合にのみ、バックエンド大容量メモリ 130 は、要請されたデータに対してアクセスされる。

【0031】

キャッシュコントローラ 310 は、デデューブキャッシュ 135 の背後から「頭脳」としての役割を担い、キャッシュコントローラ 310 は、どのようなデータがライトされるか又はリードされるかを管理して、無効化又はライトに対するキャッシュラインを選択するなどの役割を担う。キャッシュコントローラ 310 は、プロセッサ 110 と相互作用（即

10

20

30

40

50

ち、プロセッサ 110 からの要請 / データを受信してプロセッサ 110 にデータを送信) するためにホストレイヤー 315 を利用し、メモリ 130 (図 3 の「バックエンド大容量メモリ」と表示する) と相互作用するためにメディアレイヤー 320 を利用する。メモリ 130 は、デデュープキャッシュ 135 を含むモジュールの一部であるか、又はデデュープキャッシュ 135 から分離される (そして通信経路を介して通信される)。

【0032】

図 3 に示したように、キャッシュコントローラ 310、ホストレイヤー 315、及びメディアレイヤー 320 は、FPGA (field programmable gate array) を利用して具現される。しかし、本発明の実施例は、他のハードウェア設計 (PROM (Programmable Read Only Memory)、EPROM (Erasable PROM)、又はEEPROM (Electrically Erasable PROM) のような変形メモリ、或いは特別に設計された回路) 及びソフトウェア設計を含む任意の所望の具現をサポートする。なお、図 3 は、単一の FPGA を用いて具現されたキャッシュコントローラ 310、ホストレイヤー 315、及びメディアレイヤー 320 を図示するが、本発明の実施例は、ネットワークの相互連結のような通信経路を使用して連結された、複数分割された構成要素を利用する具現をサポートする。

【0033】

図 4 は、デデュープエンジンを有する図 1 のデデュープキャッシュ 135 を示す図である。図 4 において、DRAM 305、キャッシュコントローラ 310、ホストレイヤー 315、及びメディアレイヤー 320 は、図 3 に示した構成要素と同じであり、メモリコントローラ 125 は、図 1 のメモリコントローラ 125 と同じである。しかし、図 4 は、その他に、デデュープエンジン 405 及びネットワークインターコネクト (interconnect) 410 を示す。ネットワークインターコネクト 410 は、DRAM 305、キャッシュコントローラ 310、及びデデュープエンジン 405 との間に通信を提供する。デデュープエンジン 405 は、デデュープメモリ (図 3 のデデュープ可能なリードキャッシュ 325 のような) を含む DRAM 305 の部分からのデータのリードと DRAM 305 の部分へのデータのライトを管理する (デデュープメモリを含まない DRAM 305 の部分は、デデュープエンジン 405 を用いずに、直接にアクセスされる)。

【0034】

上述したように、デデュープエンジン 405 は、デデュープメモリを管理する。このため、デデュープエンジン 405 は、デデュープエンジン 405 がデデュープを遂行するように指示する信号 (図 4 の「Dedup」と表示された信号) 及びアクセスのためのアドレスと、使用のためのデータを明示するリード / ライトコマンド (図 4 の「RD / WR」と表示された信号) をサポートする。しかし、デデュープエンジン 405 は、またライト状態信号 (図 4 の「WR_Status」と表示された信号) を提供する。ライト状態信号は、特定のライト動作が成功であるか否かを示すことに利用される。ライト状態信号の利用については、図 5 を参照して、以下で更に説明する。

【0035】

図 5 は、図 3 及び図 4 のキャッシュコントローラ 310 の詳細事項を示す図である。図 5 において、キャッシュコントローラ 310 は、リスクマネージャ 505、キャッシュヒット / ミスチェック 510、ヒットブロック 515、ミスブロック 520、及びヒストリーストレージ 525 を含む。図 1 のデデュープキャッシュ 135 によって、図 1 のプロセッサ 110 からのデータ要請を受信する場合、リスクマネージャ 505 は、データの従属性が正しく処理されるようにするために、リード後ライト及びライト後リードのようなコマンドの多様な順序を追跡する。例えば、キャッシュコントローラ 310 が、図 3 の非デデュープ可能なライトバッファ 330 に格納されたデータをライトする要請を受信し、図 1 のバックエンド大容量メモリ 130 にライトされる前に、そのデータをリードする要請を受信した場合、リスクマネージャ 505 は、図 3 の非デデュープ可能なライトバッファ 330 からのデータにアクセスするリード要請を伝達する。キャッシュヒット / ミスチェック 510 は、アクセスされる特定のアドレスが、図 1 のデデュープキャッシュ 135 から

10

20

30

40

50

見つけ出されるか否かを判定する。これらの判定は、図3のメタデータ領域335にアクセスすることによって行われることから、キャッシュヒット/ミスチェック510は、図3のメタデータ領域335にアクセスするために、リード/ライト信号に従ってリード要請を伝送する。

【0036】

キャッシュヒット/ミスチェック510が、特定の要請が既に図1のデデュープキャッシュ135に有るアドレスにアクセスするか否かを判定する場合、制御は適切にヒットブロック515又はミスブロック520に伝達される。ヒットブロック515は、図1のデデュープキャッシュ135に格納された現在のキャッシュラインにアクセスするために利用される。ミスブロック520は、図3のバックエンド大容量メモリ130からのデータにアクセスするために利用される。ヒットブロック515及びミスブロック520は、図3のDRAM305に存在するデータにアクセスするためのリード/ライト信号に応じてリード及び/又はライト要請を伝送する。なお、ミスブロック520は、図4のデデュープエンジン405からライト状態の信号を受信する。最後に、ヒストリストレージ525は、図1のデデュープキャッシュ135に有るキャッシュラインへのアクセスヒストリーに関する情報を判定するために利用される（例えば、適切な場合、キャッシュラインを除外することを選択するために）。また、図5に示したネットワークインターコネクトのように、多様なネットワークインターコネクトは、キャッシュコントローラ310の多様なエレメント間の通信をサポートする。

【0037】

図3のデデュープキャッシュ135及びキャッシュコントローラ310のハードウェアを説明することで、図3のデデュープキャッシュ135の動作が説明される。図1のデデュープキャッシュ135が、図1のプロセッサ110からの要請を受信する場合、二つのタイプの要請及び二つの可能なキャッシュの結果がある。即ち、要請はリード又はライト要請であり、問題のデータは、キャッシュヒット又はキャッシュミスを引き起こす。従って、合計4つの発生可能な場合が存在し、それぞれの場合を以下で説明する。

【0038】

リード要請、キャッシュヒット

【0039】

図1のプロセッサ110がリード要請を発行し、データが現在の図3のデデュープ可能なリードキャッシュ325に格納されている場合、図5のキャッシュヒット/ミスチェック510は、図3のメタデータ領域335をリードした後に、この場合を判定する。問題のデータが現在の図3のデデュープ可能なリードキャッシュ325に存在していることから、制御は図5のヒットブロック515に伝達され、ヒットブロック515は、データが図3のデデュープ可能なリードキャッシュ325からリードされることを要請するリード要請を、図4のデデュープエンジン405に伝送する。図4のデデュープエンジン405はデータを図5のヒットブロック515に返し、ヒットブロック515はホストレイヤー315を介して図1のプロセッサ110にデータを返す。

【0040】

リード要請、キャッシュミス

【0041】

図1のプロセッサ110がリード要請を発行し、データが現在の図3のデデュープ可能なリードキャッシュ325に格納されていない場合、図5のキャッシュヒット/ミスチェック510は、図3のメタデータ領域335をリードした後に、この場合を判定する。問題のデータが現在の図3のデデュープ可能なリードキャッシュ325に存在していないことから、制御はミスブロック520に伝達され、ミスブロック520は、図3のメディアレイヤー320を介して図3のバックエンド大容量メモリ130からのデータを要請する。

【0042】

データがリードされると、図5のミスブロック520は、図3のデデュープ可能なリードキャッシュ325に現在有効なデータを格納していないキャッシュラインが存在するか否

かを判定する。図3のデデュープ可能なリードキャッシュ325に現在有効なデータを格納していない利用可能なキャッシュラインが存在する場合、図5のミスブロック520は、データを格納するためにそのようなキャッシュラインを選択する。そうでない場合、図5のミスブロック520は、図3のデデュープ可能なリードキャッシュ325から除去するための有効なデータを含むキャッシュラインを選択する。この選択プロセスは、図5のヒストリーストレージ525からのデータを利用し、除去するためのキャッシュラインを選択することに任意の所望のアルゴリズムを使用する。即ち、LRU (Least Recently Used) 又はLFU (Least Frequently Used) は、除去するためのキャッシュラインを選択するために使用されるよく知られているアルゴリズムの例である。

10

【0043】

図3のデデュープ可能なリードキャッシュ325からキャッシュラインを除去するために、図5のミスブロック520は、キャッシュラインが有効でないとしてマークすることを図3のメタデータ領域335にライトするために、図3のDRAM305にライト動作を伝送する。ミスブロック520は、図3のデデュープ可能なリードキャッシュ325に実際のデータを上書きするために、図4のデデュープエンジン405にライト動作を伝送する。例えば、キャッシュラインを満たす (fill) ための十分な0 (zero) の値をライトする。

【0044】

キャッシュラインが除去されてデータが無効化されると、キャッシュラインが最初から利用可能であったように、キャッシュラインは自由にデータを受信する。従って、図5のミスブロック520は、データ (図1のバックエンド大容量メモリ130から先にリードされた) を、図3のデデュープ可能なリードキャッシュ325にライトするために、図4のデデュープエンジン405にライト動作を伝送する。図3のキャッシュコントローラ310は、図3のホストレイヤー315を介して図1のプロセッサ110にデータを返す。

20

【0045】

ライト要請、キャッシュヒット

【0046】

図1のプロセッサ110がライト要請を発行し、データが現在図3のデデュープ可能なリードキャッシュ325に格納されている場合、図5のキャッシュヒット/ミスチェック510は、図3のメタデータ領域335をリードした後に、この場合を判定する。問題のデータが現在図3のデデュープ可能なリードキャッシュ325に存在することから、制御はヒットブロック515に伝達される。

30

【0047】

図3のデデュープ可能なリードキャッシュ325は、現在の交替されるデータを格納するため、図3のデデュープ可能なリードキャッシュ325のキャッシュラインのデータも処理しなければならない。新たなデータが、図3のデデュープ可能なリードキャッシュ325のキャッシュラインにライトされるが、デデュープメモリにデータをライトすることは、相対的に遅い動作である (そして、図1のバックエンド大容量メモリ130に格納されたデータは、いずれの場合でもアップデートされる必要がある)。従って、図3のデデュープ可能なリードキャッシュ325のキャッシュラインは無効化される。この方式で、データがプロセッサ110によって後でリードされると、新たな値は、図3の非デデュープ可能なライトバッファ330 (依然としてそこに存在する場合) 又は図1のバックエンド大容量メモリ130 (上記「リード要請、キャッシュミス」と表示した場合で説明したように) からリトリブ (retrieve) される。

40

【0048】

図3のデデュープ可能なリードキャッシュ325からキャッシュラインを無効化するために、図5のヒットブロック515は、キャッシュラインを有効でないとしてマークすることを図3のメタデータ領域335にライトするために、図3のDRAM305にライト動

50

作を伝送する。ヒットブロック 515 は、図 3 のデデューブ可能なリードキャッシュ 325 に実際のデータを上書きするために、図 4 のデデューブエンジン 405 にライト動作を伝送する。例えば、キャッシュラインを満たす (fill) ための十分な 0 の値をライトする。

【0049】

データが図 3 のデデューブ可能なリードキャッシュ 325 から無効化されると、図 5 のヒットブロック 515 は、図 3 の非デデューブ可能なライトバッファ 330 にデータをライトする。図 3 の非デデューブ可能なライトバッファ 330 が、デデューブメモリを使用しないため、図 3 の非デデューブ可能なライトバッファ 330 にデータをライトすることは、図 3 のデデューブ可能なリードキャッシュ 325 にデータをライトするよりも速い。その後、例えば図 3 の非デデューブ可能なライトバッファ 330 の両方が満た (fill) されると、適切な時間に、データは図 3 のメディアレイヤー 320 を介して図 3 の非デデューブ可能なライトバッファ 330 から図 3 のバックエンド大容量メモリ 130 にフラッシュ (flush) される。

10

【0050】

ライト要請、キャッシュミス

【0051】

図 1 のプロセッサ 110 がライト要請を発行し、データが現在図 3 のデデューブ可能なリードキャッシュ 325 に格納されていない場合、図 5 のキャッシュヒット/ミスチェック 510 は、図 3 のメタデータ領域 335 をリードした後、この場合を判定する。問題のデータが図 3 のデデューブ可能なリードキャッシュ 325 に存在していないことから、制御はミスブロック 520 に伝達される。

20

【0052】

図 5 のミスブロック 520 は、図 3 の非デデューブ可能なライトバッファ 330 にデータをライトする。その後、例えば図 3 の非デデューブ可能なライトバッファ 330 の両方が満た (fill) されると、適切な時間で、データは図 3 のメディアレイヤー 320 を介して図 3 の非デデューブ可能なライトバッファ 330 から図 3 のバックエンド大容量メモリ 130 にフラッシュ (flush) される。

【0053】

上記の説明は、全体的なパフォーマンスを向上させるために、キャッシュからどのようにしてデデューブ可能なメモリが利用されるかを示している。しかし、デデューブ可能なメモリの使用は、特にリード要請、キャッシュミス、ライト要請、及びキャッシュヒットの場合を引き起こす可能性のある他の問題を発生させる。このような二つの場合において、データは図 3 のデデューブ可能なリードキャッシュ 325 にライトされる。問題は、図 3 のデデューブ可能なリードキャッシュ 325 が物理的容量よりも大きな容量をシミュレーションするという事実から生じる。キャッシュラインが利用可能だとしても、図 3 のデデューブ可能なリードキャッシュ 325 の物理的メモリが全て満た (fill) され、このような場合、データは成功的に図 3 のデデューブ可能なリードキャッシュ 325 にライトされない可能性がある。なお、図 3 のデデューブ可能なリードキャッシュ 325 のキャッシュラインを無効化することは、任意の物理的メモリを確保していない可能性がある。例えば、無効化のために選択されたキャッシュラインが、他のキャッシュラインによって参照されているデータを参照する場合、選択されたキャッシュラインの無効化は、実際にはデータを図 3 のデデューブ可能なリードキャッシュ 325 の物理的メモリから解放しない。これらの問題は、ライト保証の欠如として説明される。

30

40

【0054】

一つの解決策は、図 3 のデデューブ可能なリードキャッシュ 325 のライトが失敗することを単に許容することである。このような解決策は、少なくともリード要請、キャッシュミスの場合に合理的である。即ち、最悪の場合に、図 1 のバックエンド大容量メモリ 130 からのアクセスを要請し (少なくともデータが、図 3 のデデューブ可能なリードキャッシュ 325 に、実際にキャッシュされるまで)、少なくともデータが図 3 のデデューブ可

50

能なリードキャッシュ 325 に実際にキャッシュされるまで、データは実際にキャッシュされない。そしてライト要請、キャッシュヒットの場合でも、同じデータにアクセスする全てのキャッシュラインを無効化する方法（図 3 のデデュープ可能なリードキャッシュ 325 から無効なデータ (stable data) にアクセスするキャッシュラインへのアクセスを防ぐために）がある場合に解決策が受け入れられるが、他の解決策が存在する。

【0055】

他の解決策は、データが成功的に図 3 のデデュープリードキャッシュ 325 にライトされるまで（又は再試みの臨界回数に到達するまで）、複数のキャッシュラインを無効化することである。図 3 のデデュープ可能なリードキャッシュ 325 にデータをライトすることを試みた後、図 4 のデデュープエンジン 405 がライト状態信号の NAK を返した場合、図 3 のキャッシュコントローラ 310（図 5 のヒットブロック 515 又は図 5 のミスブロック 520 を介して）は、無効化のための図 3 のデデュープ可能なリードキャッシュ 325 からのキャッシュラインを選択する。選択されたキャッシュラインが無効化された後、図 3 のキャッシュコントローラ 310（再び、図 5 のヒットブロック 515 又は図 5 のミスブロック 520 を介して）は、図 3 のデデュープ可能なリードキャッシュ 325 に、元のライトを再び試みる。このようなプロセスは、図 3 のデデュープ可能なリードキャッシュ 325 にライトが成功するか（ライト状態信号の ACK によって示されるように）又は再試みの臨界回数が発生するまでに必要なだけ反復される。

【0056】

図 6 A 及び図 6 B は、本発明の一実施形態による図 1 のデデュープキャッシュ 135 がライト要請を処理する例示的な手順のフローチャートである。図 6 A において、605 ステップで、図 3 のキャッシュコントローラ 310 は、図 1 のプロセッサ 110 からライト要請を受信する。610 ステップで、図 5 のキャッシュヒット/ミスチェック 510 は、図 3 のデデュープ可能なリードキャッシュ 325 が上書きされるデータを含むキャッシュラインを含むか否かを判定する。そうでない場合、615 ステップで、図 5 のミスブロック 520 は、図 3 の非デデュープ可能なライトバッファ 330 にライト要請からのデータを格納する。620 ステップで、図 3 のキャッシュコントローラ 310 は、図 3 の非デデュープ可能なライトバッファ 330 から図 1 のバックエンド大容量メモリ 130 にデータをフラッシュする。そして 625 ステップで、図 3 のキャッシュコントローラ 310 は、図 3 の非デデュープ可能なライトバッファ 330 からのデータを削除する。

【0057】

一方、610 ステップで、図 3 のデデュープ可能なリードキャッシュ 325 がライト要請によって上書きされたデータを含むキャッシュラインを含む場合、630 ステップで（図 6 B）、図 5 のヒットブロック 515 は、キャッシュラインを無効化する（図 3 のメタデータ領域 335 にキャッシュラインを有効でないとマークし、図 4 のデデュープエンジン 405 を介してキャッシュラインに 0 の値をライトすることにより）。635 ステップで、図 5 のヒットブロック 515 は、図 4 のデデュープエンジン 405 が確認応答 (ACK) 又は非確認応答 (NAK) 信号に応答するか否かを判定する。図 4 のデデュープエンジン 405 が ACK 信号を返した場合、処理は、ライト要請を完了するために図 6 A の 615 ステップに続く。図 4 のデデュープエンジン 405 が NAK 信号を返した場合、640 ステップで、図 5 のヒットブロック 515 は、再試みの最大回数に達したか否かを判定する。再試みの最大回数に達していない場合、645 ステップで、図 5 のヒットブロック 515 は、無効化のために他のキャッシュラインを選択し、処理は、新たに選択されたキャッシュラインを無効化するために、630 ステップに戻る。そうでない場合、650 ステップで、ヒットブロック 515（そして、図 3 のキャッシュコントローラ 310）は、図 3 のデデュープ可能なリードキャッシュ 325 からのデータを削除する問題があったとして報告し、その後、処理は終了する。

【0058】

図 7 は、本発明の一実施形態による図 1 のデデュープキャッシュ 135 が図 3 のデデュープ可能なリードキャッシュ 325 のキャッシュラインを無効化するための例示的な手順の

フローチャートである。図 7 において、705 ステップで、図 3 のキャッシュコントローラ 310 は、図 3 のメタデータ領域 335 にキャッシュラインを有効でないとしてマークする（適切な値を図 3 のメタデータ領域 335 にライトすることにより）。710 ステップで、図 3 のキャッシュコントローラ 310 は、無効なデータが DRAM のスペースを占有しないように、問題のキャッシュラインのデータに 0 の値をライトするために図 4 のデデュープエンジン 405 にライト要請を伝送する。

【0059】

図 8 A ~ 図 8 C は、本発明の一実施形態による図 1 のデデュープキャッシュ 135 がリード要請を処理する例示的な手順のフローチャートである。図 8 A において、805 ステップで、図 3 のキャッシュコントローラ 310 は、図 1 のバックエンド大容量メモリ 130（又は図 3 の DRAM 305 にデータが格納される場合、図 3 の DRAM 305）からデータをリードするためのリード要請を受信する。810 ステップで、図 5 のキャッシュヒット/ミスチェック 510 は、要請されたデータが、図 3 のデデュープ可能なリードキャッシュ 325 で検出されるか否かをチェックする。要請されたデータが、図 3 のデデュープ可能なリードキャッシュ 325 のキャッシュラインで発見された場合、815 ステップで、図 5 のヒットブロック 515 は、要請されたデータを、図 3 のデデュープ可能なリードキャッシュ 325 のキャッシュラインからリードするために図 4 のデデュープエンジン 405 にリード要請を伝送し、820 ステップで、図 3 のキャッシュコントローラ 310 は、要請されたデータを、図 1 のプロセッサ 110 に再び伝送し、その後、処理は終了する。

【0060】

要請されたデータは、図 3 のデデュープ可能なリードキャッシュ 325 又は非デデュープ可能なライトバッファ 330 のキャッシュラインで発見されない場合、825 ステップで、図 5 のミスブロック 520 は、図 1 のバックエンド大容量メモリ 130 からデータをリードする。830 ステップで、図 5 のミスブロック 520 は、データを格納するために図 3 のデデュープ可能なリードキャッシュ 325 からのキャッシュラインを選択する。選択されたキャッシュラインは、現在の有効なデータを格納していないキャッシュラインであるか、又は有効なデータを格納しているキャッシュライン（従って、先に無効化が必要である）である。835 ステップで（図 8 B）、図 5 のミスブロック 520 は、選択されたキャッシュラインが有効なデータを含むか否かを判定する。選択されたキャッシュラインが有効なデータを含む場合、840 ステップで、図 5 のミスブロック 520 は、上記の図 7 で記述した例示的な手順を用いて、キャッシュラインを無効化する。キャッシュラインが有効なデータを含んでいない場合、845 ステップで、図 5 のミスブロック 520 は、図 3 のデデュープ可能なリードキャッシュ 325 の選択されたキャッシュラインに要請されたデータをライトするために、図 4 のデデュープエンジン 405 にライト要請を伝送する。即ち、845 ステップで、図 5 のミスブロック 520 は、図 3 のデデュープ可能なリードキャッシュ 325 の選択されたキャッシュラインに要請されたデータをライトすることを試みるために、図 4 のデデュープエンジン 405 に要請されたデータを提供する。

【0061】

850 のステップで、図 5 のミスブロック 520 は、図 4 のデデュープエンジン 405 がライト要請に回答して ACK 又は NAK のいずれを返すかを判定する。図 3 のデデュープ可能なリードキャッシュ 325 が自由なキャッシュラインを有していても最大物理的容量に到達した可能性がある場合、図 4 のデデュープエンジン 405 は、835 ステップで選択されたキャッシュラインが有効なデータを含んでいるか否かに拘らず、NAK を返す。図 4 のデデュープエンジン 405 が ACK 信号を返した場合、ライトは成功したことを示す。即ち、855 ステップで、図 5 のミスブロック 520 は、図 3 のメタデータ領域 335 にキャッシュラインを有効なデータを含んでいるとしてマークし、処理は、図 1 のプロセッサ 110 に要請されたデータを送り返すために図 8 A の 820 ステップに続く。

【0062】

一方、図 4 のデデュープエンジン 405 が NAK 信号を返した場合、図 4 のデデュープエ

ンジン 405 は、図 3 のデデューブ可能なリードキャッシュ 325 のキャッシュラインにデータをライトすることができなかったことを示す。このような場合、860 ステップで（図 8C）、図 5 のミスブロック 520 は、再試みの最大回数に達したか否かを判定する。再試みの最大回数がまだ達していない場合、865 ステップで、図 5 のミスブロック 520 は、無効化のための他のキャッシュラインを選択し、処理は、新たに選択されたキャッシュラインを無効化するために 840 ステップに戻る。そうでない場合、870 ステップで、ミスブロック 520（そして、図 3 のキャッシュコントローラ 310）は、図 3 のデデューブ可能なリードキャッシュ 325 にデータをライトすることに問題が生じたと報告し、その後の処理は終了する。

【0063】

図 3 を参照して上述したように、本発明のいくつかの実施例において、図 3 の非デデューブ可能なライトバッファ 330 は、図 3 のデデューブ可能なリードキャッシュ 325 と共に、既存のキャッシュとして動作する。このような本発明の実施例において、要請されたキャッシュラインのために図 3 のデデューブ可能なリードキャッシュ 325 をチェックするか又はアクセスすることは、同様に図 3 の非デデューブ可能なライトバッファ 330 をチェックするか又はアクセスすることを含むものと理解される。例えば、図 8A において、810 ステップはデータを含むキャッシュラインのために図 3 のデデューブ可能なリードキャッシュ 325 及び図 3 の非デデューブ可能なライトバッファ 330 の両方をチェックするために修正され、815 ステップは、実際にデータがどこで発見されるかに応じて、図 3 のデデューブ可能なリードキャッシュ 325 又は図 3 の非デデューブ可能なライトバッファ 330 のキャッシュラインからデータをリードすることで修正される。同様に、図 8C の 870 ステップにおいて、データが成功的に図 3 のデデューブ可能なリードキャッシュ 325 にライトされていない場合、データは代わりに、図 3 の非デデューブ可能なライトバッファ 330 にライトされる。或いは、例示のフローチャートは、図 3 のデデューブ可能なリードキャッシュ 325 のキャッシュラインに、先にデータをライトすることを試みるよりも、図 3 のデデューブ可能なリードキャッシュ 325 又は図 3 の非デデューブ可能なライトバッファ 330 からのデータがライトされるキャッシュラインを選択することで修正される。

【0064】

図 6A ~ 図 8C において、本発明のいくつかの実施例を示した。しかし、本技術分野における熟練した者は、ステップの順序を変えたり、ステップを省略したり、図面に示されていない関係を含むことにより、本発明の他の実施例が可能であることを認めるはずである。明示的に説明されるか又は説明されていないとしても、このようなフローチャートの全ての変形は、本発明の実施例として考慮される。

【0065】

本発明の実施例は、先行技術よりもいくつかの技術的利点を提供する。まず、図 1 のデデューバッシュ 135 のデデューブ可能なメモリの使用は、より多くの（ユニークな）データが同一の物理的量のメモリに格納されるようにしながら（又は、同じ量のデータを格納するために、より少ないメモリスペースを使用しながら）、複数のコピーの同じデータが図 3 のキャッシュ DRAM 305 に格納されることを防ぐ。例えば、図 3 のデデューブ可能なリードキャッシュ 325 が二つの予想されたデデューブ率を有する 4GB のメモリを含む場合、図 3 のデデューブ可能なリードキャッシュ 325 は、理論的に 8GB の非デデューブ可能なキャッシュメモリに格納されるようなユニークなデータが格納される。二番目に、図 3 の非デデューブ可能なライトバッファ 330 を使用することにより、アプリケーションがデータをライトしている状況について（単にデータをリードする状況よりも）デデューブ可能なメモリへのライト固有の遅延が防止される。三番目に、保証されていない図 3 のデデューブ可能なリードキャッシュ 325 へのライトに対する懸念は、図 3 のデデューブ可能なリードキャッシュ 325 のキャッシュラインを無効化させた後のライトの再試みをサポートすることにより、減少する（図 3 のデデューブ可能なリードキャッシュ 325 へのライトがいくつかのキャッシュライン無効化の後にも成功しない可能性が依

10

20

30

40

50

然として存在するが、このような発生の可能性は非常に低い)。

【0066】

以下の内容は、本発明の特定の実施例が具現される適切なマシン又はマシンの簡略で一般的な説明を提供するためのものである。マシン又は複数のマシンは、他のマシンから受信された指示、バーチャルリアリティ(VR: virtual reality)環境との相互作用、生体フィードバック、又は他の入力信号によるものだけではなく、キーボード、マウスなどのような従来の入力デバイスからの入力により、少なくともある程度は制御される。本明細書で使用するように、用語「マシン」は、単一のマシン、バーチャルマシン、マシンとの通信に連結されたシステム、複数のバーチャルマシン、又は共に動作するデバイスを包括的に含むことを意味する。例示的なマシンは、例えば自動車、電車、タクシーのような私的又は公的運送手段のような運送装置だけでなく、パーソナルコンピュータ、ワークステーション、サーバ、ポータブルコンピュータ、携帯デバイス、電話器、タブレットのようなコンピューティングデバイスを含む。

10

【0067】

マシン又は複数のマシンは、プログラム可能な、又はプログラム可能でないロジックデバイス若しくはロジックアレイ、アプリケーション特定用途向け集積回路(ASICs: Application Specific Integrated Circuits)、エンベデッドコンピュータ、スマートカードなどのようなエンベデッドコントローラを含む。マシン又は複数のマシンは、ネットワークインターフェース、モデム、又は他の通信連結を介して一つ以上のリモートマシンに一つ以上の連結を活用する。複数のマシンは、イントラネット、インターネット、ローカルエリアネットワーク、広域ネットワークなどのような物理的及び/又は論理的ネットワークを利用して相互連結される。通常の技術者は、ネットワーク通信が高周波(RF)、衛星、マイクロ波(microwave)、IEEE(Institute of Electrical and Electronics Engineers)802.11、ブルートゥース(登録商標)(Bluetooth(登録商標))、光、赤外線、ケーブル、レーザーなどを含む多様な有線及び/又は無線の近距離又は遠距離キャリア及びプロトコル(protocol)を活用することができることを理解するだろう。

20

【0068】

本発明の実施例は、機能、手順、データ構造、アプリケーションプログラムなどを含む関連するデータのように、又は関連するデータを参照することにより説明される。関連するデータは、マシンによってアクセスされる場合、マシンがタスクを遂行するか、或いは抽象的データ型又は下位レベルのハードウェアコンテキスト(contexts)を定義する。関連するデータは、例えばRAM、ROMなどの揮発性及び/又は不揮発性メモリ、又はハードドライブ、フロッピー(登録商標)ディスク、光学ストレージ、テープストレージ、メモリカード、デジタルビデオディスク、生物学的ストレージなどを含む他のストレージデバイス及び関連するストレージメディアに格納される。関連するデータは、パケット(packets)、シリアルデータ、並列データ、伝播される信号などの形態で物理的及び/又は論理的ネットワークを含む伝送環境を通じて伝達され、圧縮又は暗号化された形態で用いられる。関連するデータは、分散環境で用いられ、マシンのアクセスのためにローカル(local)及び/又はリモートで格納される。

30

40

【0069】

本発明の実施例は、一つ以上のプロセッサによって実行されるコマンドを含むタイプのマシンリード可能な非一時的な記録媒体を含む。コマンドは、本明細書で説明したように、本発明のエレメントを遂行するコマンドを含む。

【0070】

上述した多様な方法の動作は、多様なハードウェア及び/又はソフトウェア構成(複数可)、回路(複数可)、及び/又はモジュール(複数可)のように、動作を遂行する適切な手段によって遂行される。ソフトウェアは、論理的な機能を遂行するための実行コマンドの順序リストを含み、シングル又はマルチコアプロセッサ、又はプロセッサを含むシステ

50

ムのように、コマンドの実行システム、装置、又はデバイスによって、或いは連結されて使用されるための任意の「プロセッサ可読媒体」に含まれる。

【 0 0 7 1 】

本明細書で開示した実施例に関連して説明したステップ、方法のステップ又はアルゴリズム及び機能は、ハードウェアで、プロセッサによって実行されるソフトウェアモジュールで、又はその両方の組み合わせで直接に具現される。ソフトウェアで具現される場合、機能は、コンピュータ読み取り可能な記録媒体に一つ以上のコマンド又はコードとして格納又は伝送される。ソフトウェアモジュールは、RAM (Random Access Memory)、フラッシュメモリ、ROM (Read Only Memory)、EPROM (Electrically Programmable ROM)、EEPROM (Electrically Erasable Programmable ROM)、レジスタ、ハードディスク、リムーバブル (removable) ディスク、CD - ROM、又は知られている任意の他の形態の記憶媒体で有る。

10

【 0 0 7 2 】

上述した実施例を参照して、本発明の原理を説明したように、説明した実施例は、このような原理から逸脱せずに配置及び詳細事項において変形され、任意の方法で結合される。そして、上述した内容は、特定の実施例に焦点を合わせたとしても、他の構成が考慮される。特に、「本発明の一つの実施例によれば」のような表現又は本明細書で使用したこのような表現でも、このようなフレーズは、一般的に可能な実施例を参照するための意味であり、本発明を特定の実施例の構成に限定しようとするものではない。本明細書で使

20

【 0 0 7 3 】

上述した実施例は、本発明を制限するものとして解釈されない。いくつかの実施例を説明したが、通常の技術者は、本発明の新たなティーチング (teaching) 及び利点から逸脱しない実施例について多様な変形を容易にする。これにより、このような変形は、本発明の権利範囲に含まれるものとして意図される。

【 0 0 7 4 】

本発明の実施例は、制限なしで、次のようなステートメントに拡張される。

【 0 0 7 5 】

ステートメント 1 . 本発明の一実施例によるデデュープ可能なキャッシュは、デデュープ可能なリード (read) キャッシュ及び非デデュープ可能なライト (write) バッファを含むキャッシュメモリと、デデュープ可能なリードキャッシュを利用してデータのリード (read) 及びライト (write) を管理し、デデュープ可能なリードキャッシュへのライト要請が成功したか否かを示すライト状態信号を伝送するデデュープエンジンと、キャッシュコントローラと、を備え、キャッシュコントローラは、要請に含まれるアドレスがデデュープ可能なリードキャッシュで発見されるか否かをチェックするキャッシュヒット / ミスチェックロジックと、アドレスがデデュープ可能なリードキャッシュで発見されることをキャッシュヒット / ミスチェックロジックが示す場合、キャッシュメモリの第 1 データにアクセスするヒットブロックと、アドレスがデデュープ可能なリードキャッシュで発見されないことをキャッシュヒット / ミスチェックロジックが示す場合、バックエンド大容量メモリの第 2 データにアクセスするミスブロックと、デデュープ可能なリードキャッシュの第 1 データへのアクセスに関する情報を格納するヒストリーストレージと、を含む。

30

40

【 0 0 7 6 】

ステートメント 2 . ステートメント 1 によるデデュープ可能なキャッシュにおいて、キャッシュコントローラは、デデュープ可能なリードキャッシュ及び非デデュープ可能なライトバッファにおけるデータの従属性を管理するリスクマネージャと、を更に含む。

【 0 0 7 7 】

ステートメント 3 . ステートメント 1 によるデデュープ可能なキャッシュにおいて、デデュープ可能なリードキャッシュは、デデュープ可能なリードキャッシュに物理的に適合す

50

るものよりもより多くのキャッシュラインを格納することをシミュレーションする。

【 0 0 7 8 】

ステートメント 4 . ステートメント 1 によるデデュープ可能なキャッシュにおいて、キャッシュメモリは、デデュープ可能なリードキャッシュのキャッシュラインに関する情報を格納するメタデータ領域を更に含む。

【 0 0 7 9 】

ステートメント 5 . ステートメント 1 によるデデュープ可能なキャッシュにおいて、キャッシュコントローラは、データをライトするための第 1 ライト要請をプロセッサから受信し、非デデュープ可能なライトバッファにデータを格納し、ヒットブロックは、デデュープ可能なリードキャッシュのキャッシュラインが第 1 ライト要請によって修正される場合、キャッシュラインを無効化させるようにデデュープエンジンに第 2 ライト要請を伝送する。

10

【 0 0 8 0 】

ステートメント 6 . ステートメント 5 によるデデュープ可能なキャッシュにおいて、第 2 ライト要請は、デデュープ可能なリードキャッシュのキャッシュラインに 0 の値をライトする要請を含む。

【 0 0 8 1 】

ステートメント 7 . ステートメント 5 によるデデュープ可能なキャッシュにおいて、キャッシュコントローラは、キャッシュメモリのメタデータ領域にキャッシュラインが有効でないとしてマークする。

20

【 0 0 8 2 】

ステートメント 8 . ステートメント 5 によるデデュープ可能なキャッシュにおいて、ヒットブロックは、第 2 ライト要請に応答するデデュープエンジンから非確認応答 (N A K) 信号を受信してデデュープ可能なリードキャッシュから第 2 キャッシュラインを除去し、デデュープ可能なリードキャッシュから第 2 キャッシュラインを除去した後にデデュープエンジンに第 2 ライト要請を再び伝送する。

【 0 0 8 3 】

ステートメント 9 . ステートメント 1 によるデデュープ可能なキャッシュにおいて、キャッシュコントローラは、データをリードするためのリード要請をプロセッサから受信してプロセッサにデータを伝送し、ミスブロックは、バックエンド大容量メモリからデータをリトリブ (r e t r i e v e) し、デデュープ可能なリードキャッシュにデータをライトするためにデデュープエンジンにライト要請を伝送する。

30

【 0 0 8 4 】

ステートメント 1 0 . ステートメント 9 によるデデュープ可能なキャッシュにおいて、ミスブロックは、ライト要請に応答するデデュープエンジンから非確認応答 (N A K) 信号を受信してデデュープ可能なリードキャッシュからキャッシュラインを除去し、デデュープ可能なリードキャッシュからキャッシュラインを除去した後にデデュープエンジンにライト要請を再び伝送する。

【 0 0 8 5 】

ステートメント 1 1 . 本発明の一実施形態によるキャッシュコントローラの動作方法は、データをライトするためのライト要請を受信するステップと、データがデデュープ可能なリードキャッシュのキャッシュラインに存在すると判定するステップと、デデュープ可能なリードキャッシュのキャッシュラインを無効化するステップと、データを非デデュープ可能なライトバッファに格納するステップと、を有し、デデュープ可能なリードキャッシュは、キャッシュメモリの第 1 領域であり、キャッシュメモリは、第 2 領域として非デデュープ可能なライトバッファを含む。

40

【 0 0 8 6 】

ステートメント 1 2 . ステートメント 1 1 による方法において、方法は、非デデュープ可能なライトバッファからバックエンド大容量メモリにデータをフラッシュ (f l u s h) するステップと、非デデュープ可能なライトバッファからデータを削除するステップと、

50

を更に含む。

【 0 0 8 7 】

ステートメント 1 3 . ステートメント 1 1 による方法において、デデューブ可能なリードキャッシュのキャッシュラインを無効化するステップは、キャッシュメモリのメタデータ領域にキャッシュラインが有効でないとしてマークするステップと、デデューブエンジンを介してキャッシュラインに 0 の値をライトするステップと、を含む。

【 0 0 8 8 】

ステートメント 1 4 . ステートメント 1 3 による方法において、デデューブ可能なリードキャッシュのキャッシュラインを無効化するステップは、デデューブエンジンからライト状態信号の確認応答 (A C K) を受信するステップと、を更に含む。

10

【 0 0 8 9 】

ステートメント 1 5 . ステートメント 1 3 による方法において、デデューブ可能なリードキャッシュのキャッシュラインを無効化するステップは、デデューブエンジンからライト状態信号の非確認応答 (N A K) を受信するステップと、デデューブ可能なリードキャッシュから除去する第 2 キャッシュラインを選択するステップと、第 2 キャッシュラインを無効化するステップと、デデューブエンジンを介して第 2 キャッシュラインに 0 の値をライトするステップと、を更に含む。

【 0 0 9 0 】

ステートメント 1 6 . ステートメント 1 1 による方法において、データを非デデューブ可能なライトバッファに格納するステップは、データがデデューブ可能なリードキャッシュのキャッシュラインに存在するか否かに拘らず、データを非デデューブ可能なライトバッファに格納するステップを含む。

20

【 0 0 9 1 】

ステートメント 1 7 . 本発明の他の実施形態によるキャッシュコントローラの動作方法は、データをリードするためのリード要請を受信するステップと、データがデデューブ可能なリードキャッシュの複数のキャッシュラインに存在しないと判定するステップと、バックエンド大容量メモリからデータをリードするステップと、デデューブ可能なリードキャッシュの第 1 キャッシュラインを選択するステップと、データを第 1 キャッシュラインにライトすることを試みるためにデデューブエンジンにデータを提供するステップと、リード要請に回答してデータを伝送するステップと、を有し、デデューブ可能なリードキャッシュは、キャッシュメモリの第 1 領域であり、キャッシュメモリは、第 2 領域として非デデューブ可能なライトバッファを含む。

30

【 0 0 9 2 】

ステートメント 1 8 . ステートメント 1 7 による方法において、デデューブ可能なリードキャッシュの第 1 キャッシュラインを選択するステップは、デデューブ可能なリードキャッシュから現在データを格納していない第 1 キャッシュラインを選択するステップを含む。

【 0 0 9 3 】

ステートメント 1 9 . ステートメント 1 7 による方法において、デデューブ可能なリードキャッシュの第 1 キャッシュラインを選択するステップは、キャッシュメモリのメタデータ領域からのデデューブ可能なリードキャッシュメタデータ及びヒストリーデータに回答して、デデューブ可能なリードキャッシュから除去するために現在有効なデータを格納している第 1 キャッシュラインを選択するステップと、第 1 キャッシュラインを無効化するステップと、を含む。

40

【 0 0 9 4 】

ステートメント 2 0 . ステートメント 1 9 による方法において、第 1 キャッシュラインを無効化するステップは、キャッシュメモリのメタデータ領域に第 1 キャッシュラインを有効でないとしてマークするステップと、デデューブエンジンを介して第 1 キャッシュラインに 0 の値をライトするステップと、を含む。

【 0 0 9 5 】

ステートメント 2 1 . ステートメント 1 7 による方法において、データを第 1 キャッシュ

50

ラインにライトすることを試みるためにデデュープエンジンにデータを提供するステップは、デデュープエンジンからライト状態信号の確認応答（ACK）を受信するステップを含む。

【0096】

ステートメント22．ステートメント17による方法において、データを第1キャッシュラインにライトすることを試みるためにデデュープエンジンにデータを提供するステップは、デデュープエンジンからライト状態信号の非確認応答（NAK）を受信するステップと、キャッシュメモリのメタデータ領域からのデデュープ可能なリードキャッシュメタデータ及びヒストリーデータにตอบสนองして、デデュープ可能なリードキャッシュから除去するために現在有効なデータを格納している第2キャッシュラインを選択するステップと、第2キャッシュラインを無効化するステップと、データを選択された第2キャッシュラインにライトすることを試みるためにデデュープエンジンにデータを提供するステップと、を含む。

10

【0097】

ステートメント23．本発明の一実施形態による非一時的記憶媒体は、格納されたコマンドを含み、コマンドがマシンによって実行される場合、コマンドは、データをライトするためのライト要請を受信し、データがデデュープ可能なリードキャッシュのキャッシュラインに存在すると判定し、デデュープ可能なリードキャッシュのキャッシュラインを無効化し、データを非デデュープ可能なライトバッファに格納することを惹起し、デデュープ可能なリードキャッシュは、キャッシュメモリの第1領域であり、キャッシュメモリは、第2領域として非デデュープ可能なライトバッファを含む。

20

【0098】

ステートメント24．ステートメント23による非一時的記憶媒体において、コマンドは、非デデュープ可能なライトバッファからバックエンド大容量メモリにデータをフラッシュ（flush）し、非デデュープ可能なライトバッファからデータを削除することを惹起する。

【0099】

ステートメント25．ステートメント23による非一時的記憶媒体において、デデュープ可能なリードキャッシュのキャッシュラインを無効化することは、キャッシュメモリのメタデータ領域にキャッシュラインが有効でないとしてマークし、デデュープエンジンを介してキャッシュラインに0の値をライトすることを含む。

30

【0100】

ステートメント26．ステートメント25による非一時的記憶媒体において、デデュープ可能なリードキャッシュのキャッシュラインを無効化することは、デデュープエンジンからライト状態信号の確認応答（ACK）を受信することを更に含む。

【0101】

ステートメント27．ステートメント25による非一時的記憶媒体において、デデュープ可能なリードキャッシュのキャッシュラインを無効化することは、デデュープエンジンからライト状態信号の非確認応答（NAK）を受信し、デデュープ可能なリードキャッシュから除去する第2キャッシュラインを選択し、第2キャッシュラインを無効化し、デデュープエンジンを介して第2キャッシュラインに0の値をライトすることを更に含む。

40

【0102】

ステートメント28．ステートメント23による非一時的記憶媒体において、データを非デデュープ可能なライトバッファに格納することは、データがデデュープ可能なリードキャッシュのキャッシュラインに存在するか否かに拘らず、データを非デデュープ可能なライトバッファに格納することを含む。

【0103】

ステートメント29．本発明の他の実施形態による非一時的記憶媒体は、格納されたコマンドを含み、コマンドがマシンによって実行される場合、コマンドは、データをリードするためのリード要請を受信し、データがデデュープ可能なリードキャッシュの複数のキャ

50

ッシュラインに存在しないと判定し、バックエンド大容量メモリからデータをリードし、デデュープ可能なリードキャッシュの第1キャッシュラインを選択し、データを第1キャッシュラインにライトすることを試みるためにデデュープエンジンにデータを提供し、リード要請に 응답してデータを伝送することを惹起し、デデュープ可能なリードキャッシュは、キャッシュメモリの第1領域であり、キャッシュメモリは、第2領域として非デデュープ可能なライトバッファと、を含む。

【0104】

ステートメント30．ステートメント29による非一時的記憶媒体において、デデュープ可能なリードキャッシュの第1キャッシュラインを選択することは、デデュープ可能なリードキャッシュから現在データを格納していない第1キャッシュラインを選択することを

10

【0105】

ステートメント31．ステートメント29による非一時的記憶媒体において、デデュープ可能なリードキャッシュの第1キャッシュラインを選択することは、キャッシュメモリのメタデータ領域からのデデュープ可能なリードキャッシュメタデータ及びヒストリーデータに 응답して、デデュープ可能なリードキャッシュから除去するために現在有効なデータを格納している第1キャッシュラインを選択し、第1キャッシュラインを無効化することを

【0106】

ステートメント32．ステートメント31による非一時的記憶媒体において、第1キャッシュラインを無効化することは、キャッシュメモリのメタデータ領域に第1キャッシュラインを有効でないとしてマークし、デデュープエンジンを介して第1キャッシュラインに0の値をライトすることを

20

【0107】

ステートメント33．ステートメント29による非一時的記憶媒体において、データを第1キャッシュラインにライトすることを試みるためにデデュープエンジンにデータを提供することは、デデュープエンジンからのライト状態信号の確認応答(ACK)を受信することを

【0108】

ステートメント34．ステートメント29による非一時的記憶媒体において、データを第1キャッシュラインにライトすることを試みるためにデデュープエンジンにデータを提供することは、デデュープエンジンからライト状態信号の非確認応答(NAK)を受信し、キャッシュメモリのメタデータ領域からのデデュープ可能なリードキャッシュメタデータ及びヒストリーデータに 응답して、デデュープ可能なリードキャッシュから除去するために現在有効なデータを格納している第2キャッシュラインを選択し、第2キャッシュラインを無効化し、データを選択された第2キャッシュラインにライトすることを試みるためにデデュープエンジンにデータを提供することを

30

【0109】

以上、本発明の実施形態について図面を参照しながら詳細に説明したが、本発明は、上述の実施形態に限定されるものではなく、本発明の技術的範囲から逸脱しない範囲内で多様に

40

【産業上の利用可能性】

【0110】

本発明は、ライト遅延を防止できるデデュープ可能なメモリを用いるキャッシングシステムに有用である。

【符号の説明】

【0111】

105 マシン

110 プロセッサ

115 デバイスドライバ

50

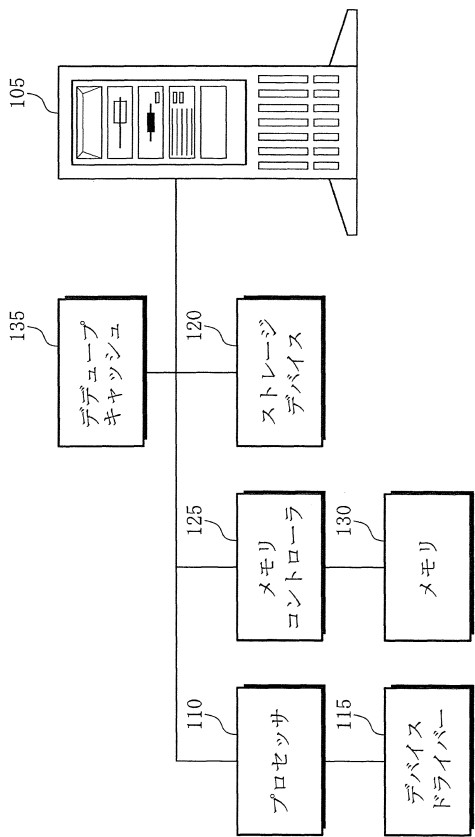
- 1 2 0 ストレージデバイス
- 1 2 5 メモリコントローラ
- 1 3 0 主メモリ (バックエンド大容量メモリ)
- 1 3 5 デデュープキャッシュ
- 2 0 5 クロック
- 2 1 0 ネットワークコネクタ
- 2 1 5 バス
- 2 2 0 ユーザーインターフェース
- 2 2 5 入出力エンジン
- 3 0 5 D R A M
- 3 1 0 キャッシュコントローラ
- 3 1 5 ホストレイヤー
- 3 2 0 メディアレイヤー
- 3 2 5 デデュープ可能なリードキャッシュ
- 3 3 0 非デデュープ可能なライトバッファ
- 3 3 5 メタデータ領域
- 4 0 5 デデュープエンジン
- 4 1 0 ネットワークインターコネクタ
- 5 0 5 リスクマネージャ
- 5 1 0 キャッシュヒット / ミスチェック
- 5 1 5 ヒットブロック
- 5 2 0 ミスブロック
- 5 2 5 ヒストリーストレージ

10

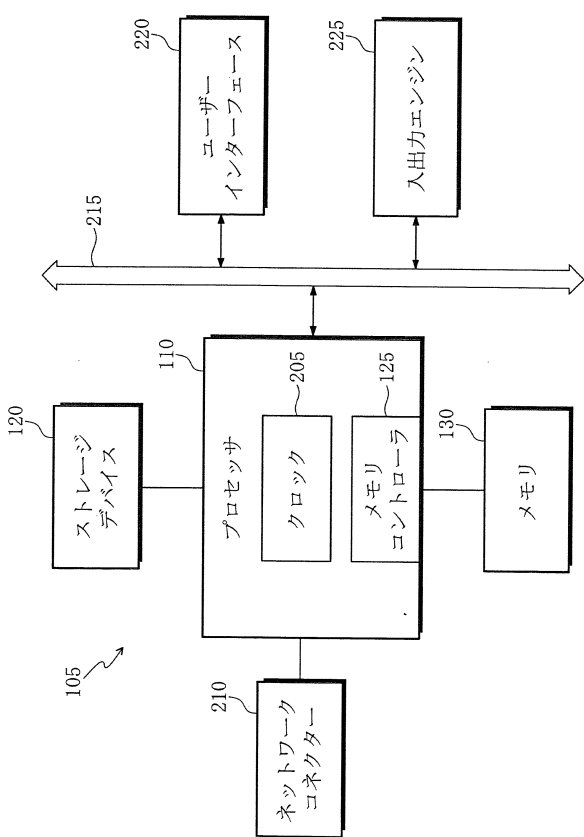
20

【図面】

【図 1】



【図 2】

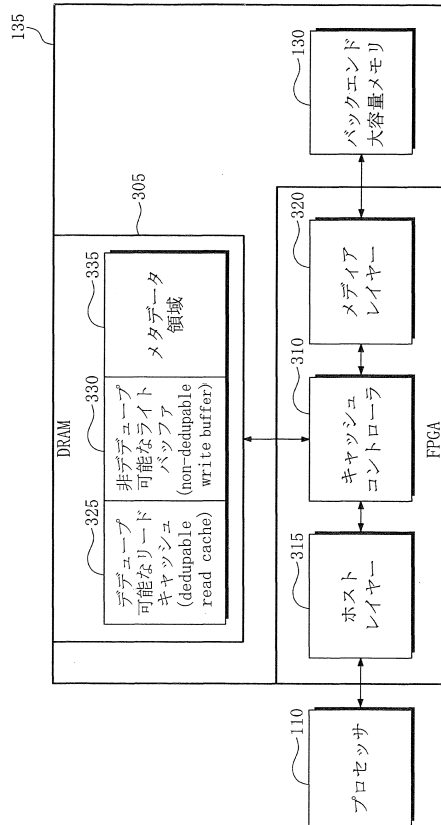


30

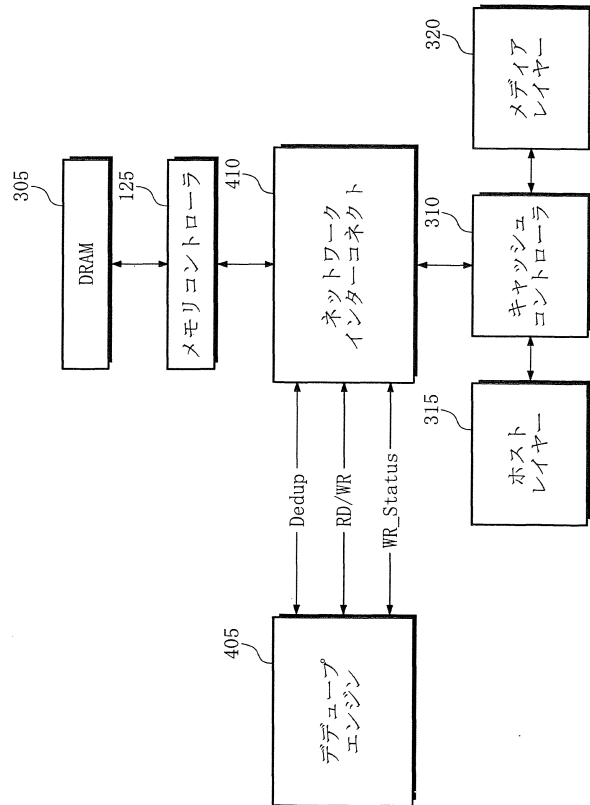
40

50

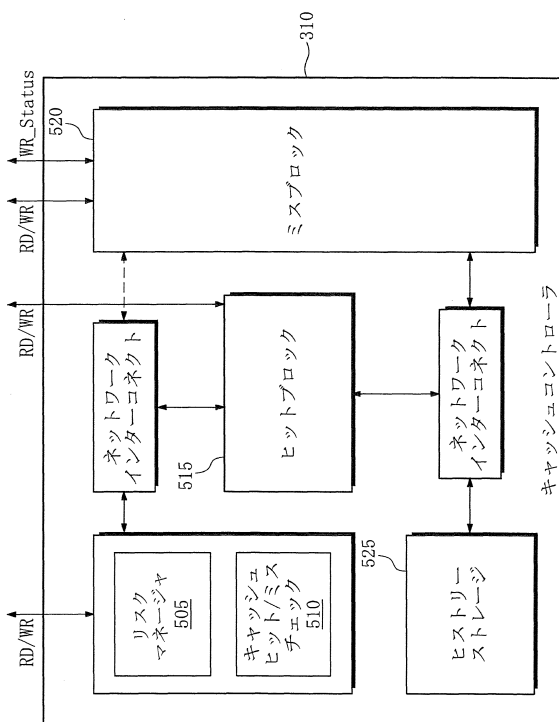
【図 3】



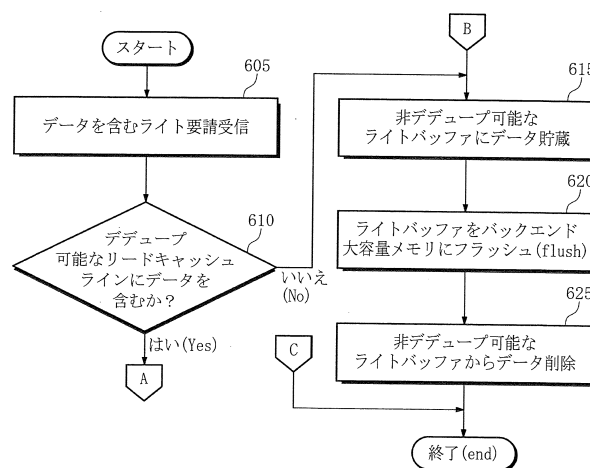
【図 4】



【図 5】



【図 6 A】



10

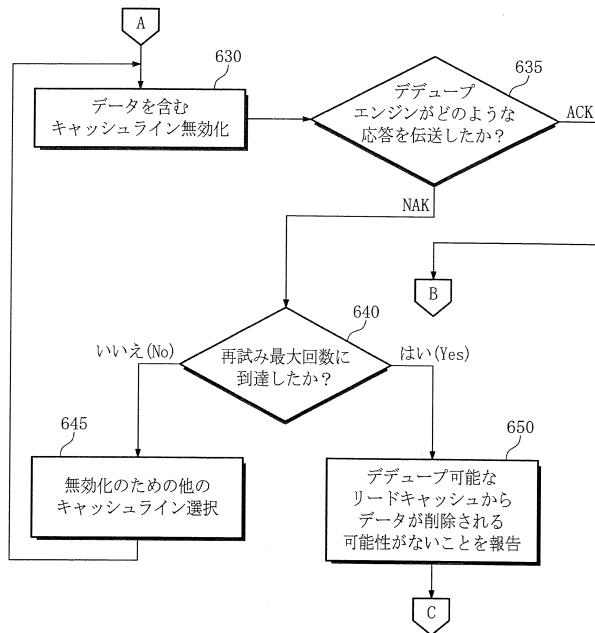
20

30

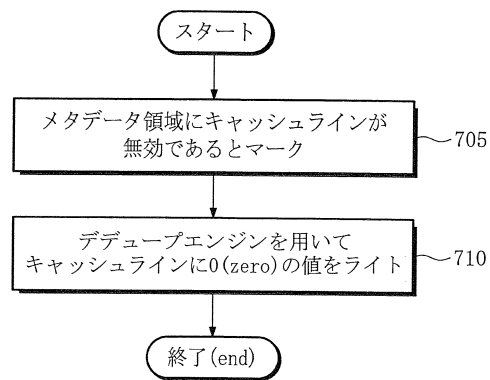
40

50

【図 6 B】



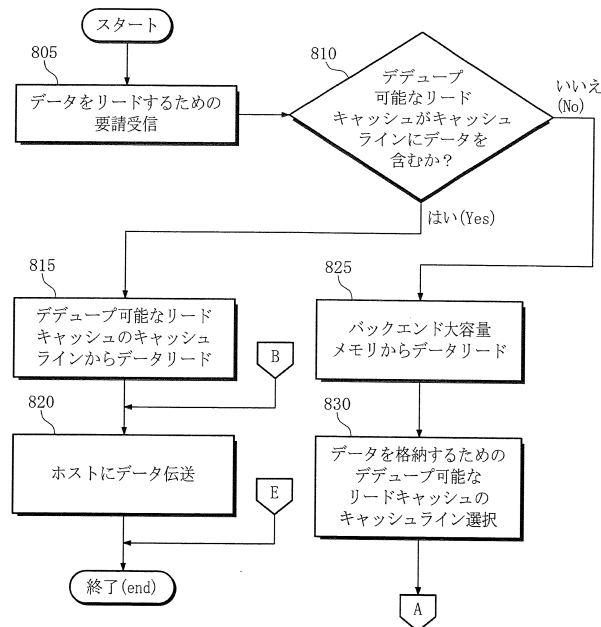
【図 7】



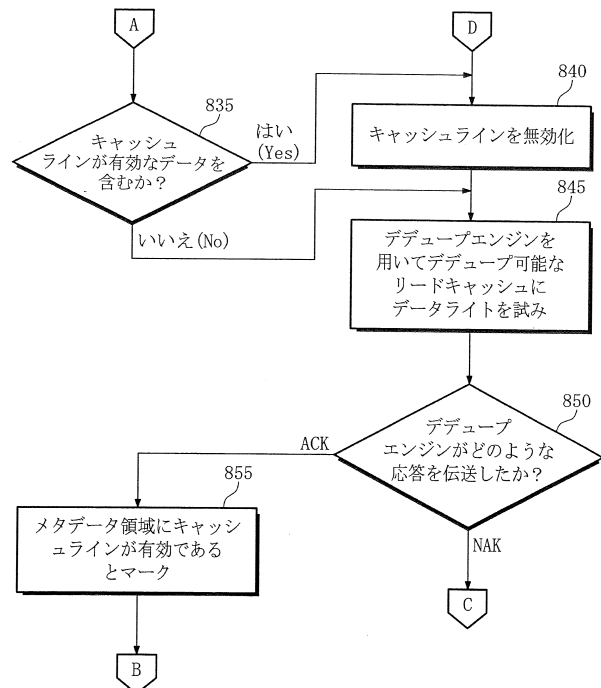
10

20

【図 8 A】



【図 8 B】

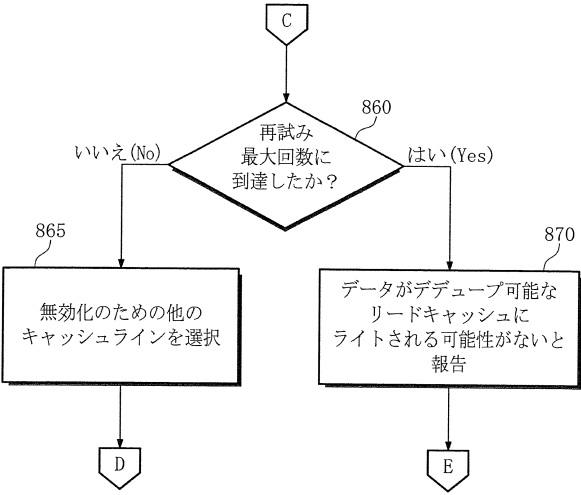


30

40

50

【図 8 C】



10

20

30

40

50

フロントページの続き

リノ プレイス 2920

(72)発明者 チャン, アンドリュー

アメリカ合衆国, 94022, カリフォルニア州, ロス アルトス, オレンジ アベニュー 679

(72)発明者 姜 冬 岩

アメリカ合衆国, 95138, カリフォルニア州, サン ノゼ, グラビーナ コート, 833

(72)発明者 鄭 宏 忠

アメリカ合衆国, 95032, カリフォルニア州, ロス ガトス, カールトン アベニュー #
6, 120

審査官 後藤 彰

(56)参考文献 特開2017-208096(JP, A)

米国特許第8732403(US, B1)

(58)調査した分野 (Int.Cl., DB名)

G06F 12/0866