

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5597666号
(P5597666)

(45) 発行日 平成26年10月1日(2014.10.1)

(24) 登録日 平成26年8月15日(2014.8.15)

(51) Int.Cl. F 1
G 0 6 F 12/00 (2006.01) G 0 6 F 12/00 5 2 0 A
G 0 6 F 17/30 (2006.01) G 0 6 F 17/30 4 1 2

請求項の数 18 (全 32 頁)

(21) 出願番号	特願2012-70322 (P2012-70322)	(73) 特許権者	000003078
(22) 出願日	平成24年3月26日(2012.3.26)		株式会社東芝
(65) 公開番号	特開2013-200839 (P2013-200839A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成25年10月3日(2013.10.3)	(74) 代理人	100089118
審査請求日	平成26年1月29日(2014.1.29)		弁理士 酒井 宏明
		(74) 代理人	100112656
			弁理士 宮田 英毅
		(72) 発明者	丸亀 孝生
			東京都港区芝浦一丁目1番1号 株式会社東芝内
		(72) 発明者	木下 敦寛
			東京都港区芝浦一丁目1番1号 株式会社東芝内

最終頁に続く

(54) 【発明の名称】 半導体記憶装置、情報処理システムおよび制御方法

(57) 【特許請求の範囲】

【請求項1】

バリュウ、および、前記バリュウに対応するキーに基づいて生成されるキーアドレスと前記バリュウの物理アドレスとを対応づけるアドレス情報を記憶する第1記憶部と、

前記キーを含む、当該キーに対応する前記バリュウの取得要求を受け付ける受付部と、前記アドレス情報に基づいて、前記取得要求に含まれる前記キーの前記キーアドレスに対応する前記物理アドレスを取得する取得部と、

取得された前記物理アドレスの前記バリュウを前記第1記憶部から取得し、取得した前記バリュウを、前記取得要求に対する応答として出力する出力制御部と、

を備える半導体記憶装置。

10

【請求項2】

キーとバリュウとを対応づけるキーバリュウ情報を記憶し、指定された前記キーを含む前記キーバリュウ情報が記憶されている場合に、指定された前記キーに対応する前記バリュウを出力する第2記憶部をさらに備える、

請求項1に記載の半導体記憶装置。

【請求項3】

前記第2記憶部は、指定された前記キーを含む前記キーバリュウ情報が記憶されていない場合に、記憶されていないことを示す情報を出力する、

請求項2に記載の半導体記憶装置。

【請求項4】

20

前記第 2 記憶部は、特定情報が記憶されている場合に、前記特定情報に応じて定められた処理を実行し、実行結果を出力する、

請求項 2 に記載の半導体記憶装置。

【請求項 5】

前記第 2 記憶部は、前記バリュウの記憶位置を表す前記特定情報が記憶されている場合に、前記記憶位置に記憶されている前記バリュウを出力する、

請求項 4 に記載の半導体記憶装置。

【請求項 6】

前記第 1 記憶部より高速にアクセス可能であり、前記第 1 記憶部から転送される前記アドレス情報を記憶する第 3 記憶部をさらに備え、

前記取得部は、前記第 3 記憶部に記憶された前記アドレス情報に基づいて、前記取得要求に含まれる前記キーの前記キーアドレスに対応する前記物理アドレスを取得する、

請求項 1 に記載の半導体記憶装置。

【請求項 7】

前記第 1 記憶部より高速にアクセス可能な第 4 記憶部へのアクセスを制御するアクセス制御部をさらに備え、

前記取得部は、前記アクセス制御部を介して、前記第 4 記憶部に記憶された前記アドレス情報を参照することにより、前記取得要求に含まれる前記キーの前記キーアドレスに対応する前記物理アドレスを取得する、

請求項 1 に記載の半導体記憶装置。

【請求項 8】

前記取得部は、起動時に前記アドレス情報を前記第 1 記憶部から読み出し、読み出した前記アドレス情報を前記第 4 記憶部に記憶する、

請求項 7 に記載の半導体記憶装置。

【請求項 9】

前記アクセス制御部は、前記取得要求が受け付けられるインタフェースと異なるインタフェースを介して、前記第 4 記憶部へのアクセスを制御する、

請求項 7 に記載の半導体記憶装置。

【請求項 10】

前記第 1 記憶部は、前記バリュウ、および、前記キーに基づいて生成される固定長のキーアドレスと前記物理アドレスとを対応づける前記アドレス情報を記憶する、

請求項 1 に記載の半導体記憶装置。

【請求項 11】

前記第 1 記憶部は、複数のページを含むブロックを複数含み、前記バリュウが記憶される前記ページの物理アドレスを表すページアドレスと、前記キーアドレスと、前記ページアドレスのページが有効であるか否かを表す判定情報と、を対応づけてさらに記憶し、

第 1 ブロックに含まれるページのうち前記判定情報が有効であることを表す有効ページを第 2 ブロックに書き込み、前記第 1 ブロックに記憶された前記有効ページを消去する転記処理部をさらに備える、

請求項 1 に記載の半導体記憶装置。

【請求項 12】

前記第 1 記憶部は、複数のページを含むブロックを複数含み、前記ページに、当該ページの次に読み出すページの物理アドレスを表す次ページアドレスを対応づけて記憶し、

取得部は、前記ページを読み出した後に、当該ページに対応づけられた前記次ページアドレスのページを読み出す、

請求項 1 に記載の半導体記憶装置。

【請求項 13】

前記第 1 記憶部は、さらに、前記バリュウと異なるデータを記憶し、

前記アドレス情報は、前記データの論理アドレスと前記データの物理アドレスとを対応づけた情報をさらに含む、

10

20

30

40

50

請求項 1 に記載の半導体記憶装置。

【請求項 1 4】

前記キーアドレスの長さと、前記論理アドレスの長さとが等しい、

請求項 1 3 に記載の半導体記憶装置。

【請求項 1 5】

前記第 1 記憶部は、前記バリューが記憶されるページの物理アドレスを表すページアドレスと、前記キーアドレスと、前記ページアドレスのページが有効であるか否かを表す判定情報と、前記ページアドレスのページの書換え頻度と、を対応づけて記憶し、

前記書換え頻度が低い前記ページに対して優先してデータを書き込む書込部をさらに備える、

10

請求項 1 に記載の半導体記憶装置。

【請求項 1 6】

前記第 1 記憶部は、前記バリュー、および、前記キーアドレスと前記物理アドレスと前記バリューのハッシュ値とを対応づける前記アドレス情報を記憶する、

請求項 1 に記載の半導体記憶装置。

【請求項 1 7】

ホスト装置と、半導体記憶装置と、を備える情報処理システムであって、

前記半導体記憶装置は、

バリュー、および、前記バリューに対応するキーに基づいて生成されるキーアドレスと前記バリューの物理アドレスとを対応づけるアドレス情報を記憶する第 1 記憶部を備え、

20

前記ホスト装置は、

前記キーを含む、当該キーに対応する前記バリューの取得要求を受け付ける受付部と、

前記アドレス情報に基づいて、前記取得要求に含まれる前記キーの前記キーアドレスに対応する前記物理アドレスを取得する取得部と、

取得された前記物理アドレスの前記バリューを前記第 1 記憶部から取得し、取得した前記バリューを、前記取得要求に対する応答として出力する出力制御部と、を備える、

情報処理システム。

【請求項 1 8】

バリュー、および、前記バリューに対応するキーに基づいて生成されるキーアドレスと前記バリューの物理アドレスとを対応づけるアドレス情報を記憶する第 1 記憶部を備える半導体記憶装置で実行される制御方法であって、

30

前記キーを含む、当該キーに対応する前記バリューの取得要求を受け付ける受付ステップと、

前記アドレス情報に基づいて、前記取得要求に含まれる前記キーの前記キーアドレスに対応する前記物理アドレスを取得する取得ステップと、

取得された前記物理アドレスの前記バリューを前記第 1 記憶部から取得し、取得した前記バリューを、前記取得要求に対する応答として出力する出力ステップと、

を含む制御方法。

【発明の詳細な説明】

【技術分野】

40

【0001】

本発明の実施形態は、半導体記憶装置、情報処理システムおよび制御方法に関する。

【背景技術】

【0002】

一般的なホストシステム、例えばコンピュータシステムが備えるストレージ装置として、磁気式のハードディスクドライブ (HDD)、不揮発性半導体メモリを搭載する固体ドライブ (SSD)、および、埋め込み NAND フラッシュメモリなどが存在する。SSD や埋め込み NAND フラッシュメモリはストレージに分類されるが、規模が拡張されたメモリシステムとも言える。

【0003】

50

このようなメモリシステムは、例えばインタフェースと、第1のメモリブロックと、第2のメモリブロックと、コントローラとを備える。第1のメモリブロックはデータを格納する。第2のメモリブロックはデータ書き込み/読み出し時のバッファメモリである。第1のメモリブロックは、不揮発性メモリであり、第2のメモリブロックよりも大容量であるが、アクセス速度が遅い。第2のメモリブロックは、第1のメモリブロックのアドレス変換テーブルを処理するための一時保存メモリである。また、第2のメモリブロックは、インタフェースの通信速度と、第1のメモリブロックの書き込み/読み出し速度との速度差を補うためにも使用される。

【0004】

例えば、第1のメモリブロックは不揮発性のフラッシュメモリであり、第2のメモリブロックは揮発性のDRAMまたはSRAMである。このような従来のストレージ型のメモリシステムでは、アドレスを指定したデータの書き込み/読み出し機能を実現するための構成をとっている。特に、SSDなどの大容量メモリシステムでは、フラッシュメモリ管理のために論理アドレスと物理アドレスとを分けて管理している。2種類のアドレスを使い分けることにより、管理が容易になる。

【0005】

一方、メモリシステムに保存されている、あるテキストに関連付けられた別のテキスト、バイナリファイルにおける特定のビットパターン、動画ファイルにおける特定パターン、および、音声ファイルにおける特徴的な音声パターンなどのデータを効率的に取り出すためには、データを指定したデータ読出し機能を持たせることが望ましい。このため、通常

【0006】

メタデータの管理方法の1つとして、1対1または1対多にデータが対応するKVS (key-value store (キーバリューストア)) が存在する。KVSでは、検索要求としてキー (key) が与えられると、それに対応付けられるバリュー (value) が出力される。

【先行技術文献】

【特許文献】

【0007】

【特許文献1】特許第4551958号公報

【特許文献2】特開2005-209214号公報

【発明の概要】

【発明が解決しようとする課題】

【0008】

しかしながら、従来システムでKVSを実現するには、メモリシステム内に格納されたデータや複数のメタデータを、ホストシステムの主記憶装置 (DRAMなど) にデータを展開した上で、中央演算装置 (CPU) を使用して演算を施した後に、再びストレージ (メモリシステム) からデータを読み出して照合するという、繰り返しのデータ入出力処理

【課題を解決するための手段】

【0009】

実施形態の半導体記憶装置は、記憶部と、受付部と、取得部と、出力制御部と、を備える。記憶部は、バリュー、および、バリューに対応するキーに基づいて生成されるキーアドレスとバリューの物理アドレスとを対応づけるアドレス情報を記憶する。受付部は、キーを含む、当該キーに対応するバリューの取得要求を受け付ける。取得部は、アドレス情報に基づいて、取得要求に含まれるキーのキーアドレスに対応する物理アドレスを取得する。出力制御部は、取得された物理アドレスのバリューを記憶部から取得し、取得したバリューを、取得要求に対する応答として出力する。

【図面の簡単な説明】

【 0 0 1 0 】

【図 1】第 1 の実施形態にかかる半導体記憶装置のハードウェア図。

【図 2】デバイス制御部のブロック図。

【図 3】L 2 P テーブルを使ったアクセスを説明する図。

【図 4 - 1】K 2 P テーブルのデータ形式の一例を示す図。

【図 4 - 2】K 2 P テーブルのデータ形式の一例を示す図。

【図 5】K 2 P テーブルを L 2 P テーブルと独立に管理する例を示す図。

【図 6】K 2 P テーブルと L 2 P テーブルとを 1 つのテーブルで管理する例を示す図。

【図 7】キードレスの衝突について説明する図。

【図 8】P 2 K テーブルのデータ形式の一例を示す図。

10

【図 9】P 2 L / P 2 K テーブルのデータ形式の一例を示す図。

【図 10】登録コマンドを受け付けた場合の処理のフローチャート。

【図 11】書込コマンドを受け付けた場合の処理のフローチャート。

【図 12】取得コマンドを受け付けた場合の処理のフローチャート。

【図 13】読出コマンドを受け付けた場合の処理のフローチャート。

【図 14】物理ブロックテーブルを用いる場合のデータアクセスの仕組みを説明する図。

【図 15】変形例 1 の物理ブロックテーブルのデータ形式の一例を示す図。

【図 16】多段階の検索テーブルを用いる変形例 2 を説明する図。

【図 17】変形例 3 の K 2 P テーブルのデータ形式の一例を示す図。

【図 18】2 種類のハッシュ関数を用いる例を説明する図。

20

【図 19】第 2 の実施形態にかかる半導体記憶装置のハードウェア図。

【図 20】C A M を用いた検索処理例を説明する図。

【図 21】第 3 の実施形態にかかる半導体記憶装置のハードウェア図。

【図 22 - 1】第 4 の実施形態にかかる半導体記憶装置のハードウェア図。

【図 22 - 2】第 4 の実施形態の変形例にかかる半導体記憶装置のハードウェア図。

【図 23】第 5 の実施形態にかかる半導体記憶装置のハードウェア図。

【発明を実施するための形態】

【 0 0 1 1 】

以下に添付図面を参照して、この発明にかかる半導体記憶装置の好適な実施形態を詳細に説明する。

30

【 0 0 1 2 】

以下では、従来システムとして、S S D を想定して説明する。以下の実施形態では、S S D は、広い意味での N A N D フラッシュベースの固体メモリによるストレージを意味し、埋め込み N A N D フラッシュメモリシステムも含む。また、実施形態の S S D は、それらよりも大型のサーバ向けストレージも含まれる。

【 0 0 1 3 】

S S D での K V S の実現方法とその問題点を説明する。従来型 S S D で K V S を実現する場合、例えば、データ（実データ）はファイルとして保存され、データに付属するキーバリュー型のメタデータ（K V S データ）もまたファイルとして保存される。すなわち K V S を実現しているのはファイルシステム以上の上位システムである。例えば、O S （オペレーティングシステム）に実装されたファイルシステムまたはアプリケーションが K V S を実現する。この場合、汎用的なハードウェア構成で K V S を実現できるという利点がある。しかしこの場合、K V S の取り扱いは、通常データと同じである。このため、例えばホストシステムが K V S データファイルをメインメモリ（例えば D R A M ）に読み出してから、メタデータ（K V S データ）の読み出し / 書き込み操作および検索操作を行うことになる。従って、ソフトウェア（S W ）としての取り扱い以上の効果は期待できない。

40

【 0 0 1 4 】

一方、S S D の読み出し / 書き込みの過程では、N A N D フラッシュメモリのハードウェア（H W ）の仕組みに基づいてアドレス変換が行われる仕組みになっている。N A N D

50

フラッシュメモリは、読み出し／書き込み時に、例えば4KBや8KBなどのページ単位でアクセスされる。一方、NANDフラッシュメモリは、例えば512KBや1024KBなどのページを複数まとめたブロックと呼ばれる単位で消去される仕組みとなっている。

【0015】

通常、同一ページのデータ更新はできないので、更新されたデータは新しいページに書き込むことになる。このため、使われているページと使われていないページを管理するアドレス管理テーブルが必要となる。また同じページへの書き込みが集中しないように、書き込みアドレスをランダムに選択するようにしている。このため、使用している物理アドレス（物理ページアドレス）と、ホストシステム側やメモリ制御部（後述）が指定する論理アドレス（論理ページアドレス）の変換テーブルが必要になる。これが論理（Logical）アドレス物理（Physical）アドレス変換テーブルであり、通称L2Pテーブルと呼ばれる。L2Pテーブルでデータを管理することにより、SSDの寿命が向上することになるが、その反面、データ管理の仕組みは複雑化する。

10

【0016】

以下の実施形態の半導体記憶装置は、例えばNANDフラッシュメモリなどから構成される不揮発性メモリシステムであって、KVSデータ（キーバリュース情報）を、アドレス変換テーブルによって効率的かつ高速に処理する。また、通常のアドレス指定データ出力のためのアドレス変換テーブルとKVSのためのアドレス変換テーブルとを共存させて、効率よく動作させる。なお、以下では半導体記憶装置を、メモリシステムまたはデバイスともいう。

20

【0017】

次に、各実施形態に共通するKVSデータの詳細について説明する。メモリシステムのメモリアクセス可能なアドレス空間内には、アドレス指定で実データにアクセス可能なデータ格納領域（実アドレス空間）と、KVSデータ格納領域とが存在する。実アドレス空間は、例えば従来の論理アドレス空間に対応する。KVSデータ格納領域は、メモリシステム側が適宜用いるデータ領域である。従って、ユーザまたはクライアントは、メモリシステムのインタフェースに対するKVSコマンドによってアクセスする。

【0018】

ここでKVSコマンドの一例について説明する。ホストシステムからは、メモリシステムのホストインタフェースに対し、KVSに対する操作要求（KVS要求）のための、次のKVSコマンドが与えられる。

30

- ・登録コマンド（PUT）：キーに関連付けられた新たな集合（バリュー）を登録する。
- ・書込コマンド（APPEND）：あるキーに関連付けられた集合（バリュー）に新たな要素（バリュー）を追加する。
- ・取得コマンド（GET）：キーに関連付けられた集合（バリュー）の要素をワークメモリ（またはバッファメモリ）に格納しそのサイズを返す。
- ・読出コマンド（READ）：ワークメモリ（またはバッファメモリ）に格納されている要素（バリュー）を読み出す。

【0019】

40

コマンド名称は適宜変更されてもよい。KVS要求のために新たなコマンドが追加されてもよい。例えば、集合（キー）に属する要素（バリュー）を整理するためのコマンドを用いてもよい。また、K2Pテーブル（後述）内の集合（キー）の並び替え、および、要素（バリュー）同士の比較などを命令するコマンドを用いてもよい。

【0020】

メモリシステムは、L2PテーブルとK2Pテーブルとを有する。L2Pテーブルは、論理アドレスと物理アドレスの変換テーブルである。K2Pテーブルは、キーから得た固定長アドレス（キーアドレス）と物理アドレスの変換テーブルである。メモリシステム（デバイス）を制御するデバイス制御部（詳細は後述）は、ホストシステムからの要求に応じてこれら2種類のテーブルを使い分け、実アドレス空間へのアクセスと、KVSデータ

50

へのアクセスとを行う。

【 0 0 2 1 】

K 2 P テーブルは必要に応じて作られるので、ホストシステムから作成が要求されなければ、第 1 のメモリブロック内に存在しない場合も有り得る。このように、K V S データおよび K 2 P テーブルは固定されて設置されているわけではなく、任意に拡張および縮小されて存在できる。このため、ユーザは K V S データを任意に扱いつつ、アクセス可能な物理メモリ空間を最大効率で利用可能となる。

【 0 0 2 2 】

K V S データおよび K 2 P テーブルの管理は、デバイス側（ローカルシステム側）の機能である。このため、ホストシステム側はメタデータ（K V S データ）の管理から開放されている。

10

【 0 0 2 3 】

K V S データおよび K 2 P テーブルの実態は、第 1 のメモリブロックの物理ページに格納されている。通常の L 2 P テーブルでアクセスすることも可能であるし、L 2 P テーブルでアクセスできない特別領域として管理することも可能である。これらに関しては後述の実施形態で説明する。

【 0 0 2 4 】

次に、K V S データを取り出す処理の具体例を説明する。一般的に K V S とは、キー（key）と値（value）の組を書き込み、キーを指定することで値を読み出せるデータベース管理方式を意味する。一般的にはネットワーク越しに使われる例が多い。データの格納先は、いずれかのローカルのメモリまたはストレージシステムであることは間違いない。

20

【 0 0 2 5 】

データは通常、保存されているメモリの先頭アドレスとデータ長を指定することで読み出される。データアドレスは、ホストシステムの OS やファイルシステムによって、例えば 5 1 2 バイトのセクタ単位で管理される。または、ファイルシステムを限定する必要がなければ、例えば NAND フラッシュメモリの読み出し / 書き込みページサイズに合わせて、4 K B や 8 K B 単位で管理されてもよい。

【 0 0 2 6 】

最も単純な検索手順は、以下の（１）～（３）のようになる。

30

（１）キーを H a s h 関数などにより固定長データへ変換し、その固定長データを使用可能なメモリのアドレスに変換して固定長アドレスを得る。変換した固定長アドレスをキーアドレスとする。

（２）N A N D フラッシュメモリ内に保存されている K 2 P テーブルを参照し、物理アドレスを得る。

（３）物理アドレスのデータを読み、メモリシステム外へ出力する。

【 0 0 2 7 】

このような実データアドレスと K V S データの関係、および、キーとバリューの関係は、要素と集合の関係となっている。すなわち、通常のファイルでは、例えば「a - f i l e . t x t」というファイル名のファイルが集合であり、そのファイルの中身に「T h i s i s a b o o k」というテキストデータがあれば、そのそれぞれの単語が要素である。

40

【 0 0 2 8 】

一方、キーバリューでは集合と要素の関係が逆転して整理し直すことができる。すなわち、「転置」の関係に変換して保存することができる。例えば「b o o k」という集合の中には、「a - f i l e . t x t」、「b - f i l e . t x t」というファイル名が要素として保存されることになる。キーバリューでは、この整理しなおされた集合名（「b o o k」）を検索し、その要素（「a - f i l e . t x t」、「b - f i l e . t x t」）を求めていることになる。これは一般に全文検索で実行されている転置ファイル作成、および、検索手順そのものであって、キーバリューの実用上の一例といえる。

50

【 0 0 2 9 】

転置ファイルとは、全文検索機能の実現法の1つである転置インデックス法で用いられる検索のための索引ファイルである。転置インデックス法では、コンテンツごとに、当該コンテンツを含むファイルのリストを格納する転置ファイルと呼ばれる索引データファイルを予め作成する。そして、ファイルの追加/削除の都度、転置ファイルの内容を更新する。コンテンツ検索要求に対しては、検索対象のコンテンツに対応する転置ファイルの内容を検索結果として出力すればよい。そのため、全文検索の都度、すべてのファイルの内容を調べる必要がない。従って、検索を高速化できる。転置ファイルは、K V Sデータの1つの用例である。本実施形態のK V Sデータは転置ファイルに限定されるものではない。また、本実施形態は全文検索に特化した技術ではない。

10

【 0 0 3 0 】

以下、各実施形態の詳細について説明する。

【 0 0 3 1 】

(第1の実施形態)

図1は、第1の実施形態にかかる半導体記憶装置であるデバイス100およびホストシステム200のハードウェア構成の一例を示すブロック図である。図1に示すように、ホストシステム200は、CPU201と、メインメモリ202と、CPU201およびメインメモリ202を接続するバス211と、を備えている。

【 0 0 3 2 】

デバイス100は、ホストインタフェース101と、デバイス制御部110と、メモリ制御部120と、記憶部130と、を備えている。

20

【 0 0 3 3 】

ホストインタフェース101、デバイス制御部110、およびメモリ制御部120は、バス102を介して接続される。デバイス100内では、高速かつ効率的なバス線配置を取られることが望ましい。一方、例えばインタフェース規格と外部インタフェース規格の相違などで、2種類以上のバス線がデバイス100内で用いられていてもよい。

【 0 0 3 4 】

ホストシステム200は、例えばA M B A (Advanced Microcontroller Bus Architecture)、などのバス211を介してホストインタフェース101に接続される。ホストインタフェース101は、S A T A (Serial Advanced Technology Attachment)、P C I E x p r e s s、e M M C (embedded MMC)、U F S ((Universal Flash Storage))および、U S B (Universal Serial Bus)などを適宜選択する。

30

【 0 0 3 5 】

ホストインタフェース101は、ホストシステム200からアドレス指定による通常の日データ操作要求とK V S要求とを受け付けることができる。

【 0 0 3 6 】

第1のメモリブロックに相当する記憶部130は、実データブロック131と、テーブルブロック132と、K V Sデータブロック133と、を含む。実データブロック131は、実データを記憶するブロックを表す。テーブルブロック132は、各種テーブルを記憶するブロックを表す。K V Sデータブロック133は、K V Sデータを記憶するブロックを表す。

40

【 0 0 3 7 】

テーブルブロック132は、例えばL 2 Pテーブル132aと、K 2 Pテーブル132bと、P 2 L / P 2 Kテーブル132cと、を記憶する。K V Sデータブロック133は、例えば実データから抽出したK V Sデータを格納する。後述するように、K 2 Pテーブル132bを用いれば、キーに対応するバリュウの物理アドレスを特定できる。このため、K V Sデータは、少なくともバリュウを含んでいればよく、キーを含まなくてもよい。

【 0 0 3 8 】

K V S要求を処理するためには、少なくともK 2 Pテーブル132bが記憶されていればよい。P 2 L / P 2 Kテーブル132cは、実データおよびK V Sデータの追加や修正

50

を行う場合に用いる逆引きテーブルである（詳細は後述）。L 2 P テーブル 1 3 2 a を備えない場合は、K 2 P テーブル 1 3 2 b に対応する逆引きテーブル（P 2 K テーブル）のみを備えればよい。

【 0 0 3 9 】

記憶部 1 3 0 は、例えば不揮発性半導体メモリの N A N D フラッシュメモリである。記憶部 1 3 0 は、記憶容量を増大させるため複数のチップで構成されていてもよい。記憶部 1 3 0 は、これに限定されるものではなく、記憶不揮発性を有する半導体メモリであればあらゆる記憶媒体を適用できる。例えば、記憶部 1 3 0 として、M R A M (Magnetoresistive Random Access Memory) や R e R A M (Resistance Random Access Memory) や F e R A M (Ferroelectric Random Access Memory) や P C R A M (Phase-Change Random Access Memory) などの不揮発性メモリを用いてもよい。

10

【 0 0 4 0 】

K V S データは、データに関連付けられたメタデータとしてのキーと、関連付けられたデータの実データアドレスの先頭アドレスとを、例えばリストとして格納する。この K V S データを用いて、例えば、上述した転置ファイルなどを構成することができる。

【 0 0 4 1 】

メモリ制御部 1 2 0 は、記憶部 1 3 0 への書き込み / 読み出し要求を受付け、書き込み / 読み出し要求に応じて記憶部 1 3 0 に対するアクセスを制御する。メモリ制御部 1 2 0 は、書き込みまたは読み出しを行う際に一時的に使用する第 2 のメモリブロックとしてのバッファメモリ 1 2 1 を備える。バッファメモリ 1 2 1 は、例えば記憶部 1 3 0 の多値動作を制御するための演算機能を備えてもよい。メモリ制御部 1 2 0 と記憶部 1 3 0 は近い位置に接続されており、1 つのチップ内に収めることができる。また、チップを分けた場合でも 1 つのパッケージに収めることもできる。記憶部 1 3 0 の多値動作を制御するための演算機能は、記憶部 1 3 0 内に備わっていてもよい。

20

【 0 0 4 2 】

デバイス制御部 1 1 0 は、ホストインタフェース 1 0 1 と、メモリ制御部 1 2 0 を介して記憶部 1 3 0 の相互の信号送受信を制御する。デバイス制御部 1 1 0 は、R A M などのワークメモリ 1 1 1 を備える。

【 0 0 4 3 】

デバイス制御部 1 1 0 は、記憶部 1 3 0 から出力されたデータに対するエラー訂正符号復号 (E C C) 機能を備えてもよい。また、デバイス制御部 1 1 0 は、記憶部 1 3 0 の論理アドレス物理アドレス変換を行うことができる。E C C 機能は、メモリ制御部 1 2 0 に備わっている場合がある。同様に、E C C 機能は、記憶部 1 3 0 に備わっている場合がある。2 種類以上の E C C 機能が別々のブロックに備わっている場合がある。本実施形態ではメモリ制御部 1 2 0 に備わっていることを前提とし、データ読み出しの際には E C C 処理が施されてからデバイス制御部に伝えられるものとする。

30

【 0 0 4 4 】

メモリ制御部 1 2 0 のバッファメモリ 1 2 1 は、これら処理のために用いてもよい。バッファメモリ 1 2 1 に相当する第 2 のメモリブロックは、メモリ制御部 1 2 0 の内部にあることは必須ではなく、デバイス制御部 1 1 0 の外部にバス線を介して接続されていてもよい。第 2 のメモリブロックは必須ではなく、第 2 のメモリブロック (バッファメモリ 1 2 1) を備えないように構成してもよい。ただし、デバイス制御部 1 1 0 が第 2 のメモリブロックを利用できるのであれば、K V S データを記憶部 1 3 0 から第 2 のメモリブロック内へ読み出して参照することができる。

40

【 0 0 4 5 】

なお、第 2 のメモリブロックは、例えば揮発性であり、かつ、記憶部 1 3 0 に比べて小容量であるがアクセス速度が速い記憶媒体が用いられる。例えば、第 2 のメモリブロックは揮発性の D R A M または S R A M である。または、同等の速度と容量が得られれば不揮発性の M R A M などでもよい。

【 0 0 4 6 】

50

第2のメモリブロックは、ホストインタフェース101の通信速度と、記憶部130のアクセス速度との速度差を補うために使用される。記憶部130にフラッシュメモリを用いたメモリシステムであれば、デバイス制御部110、第2のメモリブロック、およびL2 Pテーブル132aを用いて、ウェアレベリング（メモリセル寿命の平坦化）機能を備えるのが一般的である。各実施形態でこのようなウェアレベリング機能を備えるように構成してもよい。

【0047】

図2は、デバイス制御部110の機能構成の一例を示すブロック図である。図2に示すように、デバイス制御部110は、受付部112と、取得部113と、出力制御部114と、書込部115と、転記処理部116と、生成部117と、を備えている。

10

【0048】

受付部112は、キーに対応するバリュウの取得要求を受け付ける。

【0049】

取得部113は、記憶部130から各種データを読み出す。例えば、取得部113は、記憶部130に記憶されたK2 Pテーブル132bを用いて、取得要求に含まれるキーのキーアドレスに対応するバリュウの物理アドレスを取得する。また、取得部113は、物理アドレスのバリュウをKVSデータから読み出す。

【0050】

書込部115は、記憶部130に対して各種データを書き込む。書込部115が、ウェアレベリング機能を備えてもよい。例えば、書込部115は、P2 Kテーブルに記憶された書換え回数（書換え頻度）を参照し、書換え回数が小さい物理ページを優先して使用するように構成してもよい。

20

【0051】

出力制御部114は、読み出されたバリュウを取得要求に対する応答として出力する。

【0052】

転記処理部116は、ガベージコレクションおよびコンパクションを実行する。ガベージコレクションとは、ブロック内の不使用ページを整理する処理である。コンパクションとは、離散化した使用中ページを1つの物理ブロックにまとめて空きブロックを確保する処理である。

【0053】

30

生成部117は、キーに対応する固定長のキーアドレスを生成する。生成部117は、例えばハッシュ関数を生成する機能を有する電子回路などにより実現できる。この電子回路は、専用回路、および、ハッシュ関数アルゴリズムを入力した汎用回路のいずれを用いてもよい。ハッシュ関数を用いたデータ格納方法および検索方法は後述する。

【0054】

なお、図2に示す各部の全部または一部は、ハードウェア回路により実現してもよいし、デバイス制御部110に備えられるCPUにより実行されるソフトウェア（プログラム）により実現してもよい。

【0055】

このプログラムは、例えばROM等に予め組み込んで提供される。または、第1のメモリブロックからシステムデータとして、デバイス起動時にプログラムが読み込まれる仕様にしてもよい。

40

【0056】

本プログラムを、インストール可能な形式又は実行可能な形式のファイルでCD-ROM（Compact Disk Read Only Memory）、フレキシブルディスク（FD）、CD-R（Compact Disk Recordable）、DVD（Digital Versatile Disk）等のコンピュータで読み取り可能な記録媒体に記録されてコンピュータプログラムプロダクトとして提供するように構成してもよい。

【0057】

また、本プログラムを、インターネット等のネットワークに接続されたコンピュータ上

50

に格納し、ネットワーク経由でダウンロードさせることにより提供するように構成してもよい。また、本プログラムをインターネット等のネットワーク経由で提供または配布するように構成してもよい。

【0058】

本プログラムは、上述した各部を含むモジュール構成となっており、実際のハードウェアとしてはCPU（プロセッサ）が上記記憶媒体からプログラムを読み出して実行することにより上記各部が主記憶装置上にロードされ、上述した各部が主記憶装置上に生成されるようになっている。

【0059】

次に、ハッシュ関数を用いたキーアドレス作成方法について説明する。デバイス制御部110がハッシュ関数を生成する機能またはハッシュ関数アルゴリズムを実行できるCPUを備えていると、任意長ビットデータをハッシュ関数によって固定長ビットデータに変換することができる。ここでは、この機能を用いて、生成部117が、任意長ビットデータのキーから、固定長ビットデータのキーアドレスを生成する例を示す。

【0060】

ハッシュ関数としては、なるべく均一かつ疎な暗号学的ハッシュ関数が好ましい。例えば、SHA-1（Secure Hash Algorithm-1）、SHA-2（Secure Hash Algorithm-2）、MD4（Message Digest 4）、および、MD5（Message Digest 5）などが使用される。

【0061】

例えば、MD5を用いて変換すると、「abcd」は、16バイトすなわち128ビット長の「e2fc714c4727ee9395f324cd2e7f331f（16進数）」となる。同様に、SHA-1アルゴリズムなどのアルゴリズムを用いて変換しても特定の固定長データパターンが得られる。

【0062】

生成部117は、ハッシュ関数に従って生成された、ある固定長ビットのビット列を、所望のビット長（BitLength）に短縮する機能を有する。例えば、生成部117は、次式の除算機能を有する。

$$\langle \text{KeyID} \rangle = \text{hash}(\langle \text{Key} \rangle) \bmod \text{BitLength}$$

【0063】

生成部117は、このようにビット除算や剰余算を行ってビット列を短縮する。生成部117が、単純に、生成された固定長ビットのビット列の先頭から所望の長さの分だけを切り取って使用してもよい。上記の例で、128ビットから32ビットを切り取ると「e2fc714c（16進数）」である。さらにKVSを格納するメモリのアドレス単位に沿ってアドレスを揃える。例えば下位8ビットを切り捨てて「e2fc7140（16進数）」となる。これがキーアドレスとなる。

【0064】

このように生成されたキーアドレスの長さを、L2Pテーブル132aの論理アドレス長と等しくしておけば、L2Pテーブル132aの管理方法をそのまま利用できる。すなわち、論理アドレスを物理アドレスに変換する方法と同様に、キーアドレスを物理アドレスに変換することができる。

【0065】

ハッシュ関数を用いたとしても、異なるデータからのハッシュ値が同一になる問題、いわゆるハッシュ値の衝突は、数学的には確率がゼロにはならない。

【0066】

ハッシュ関数以外の、任意長から固定長を生成する簡単な方法として、「book」を「bo」、「blue」を「bl」、「note」を「no」というように、先頭数バイト分を切り取って、例えばそれをASCIIコードで変換して、「bo（1バイト文字）」を「0x62, 0x6f」のようにして固定長を生成する方法を用いてもよい。ただしこの場合も衝突の可能性が生じるので注意が必要である。

10

20

30

40

50

【 0 0 6 7 】

ここで、一般的な L 2 P テーブル 1 3 2 a を使ったデータアクセスの仕組みを、図 3 を用いて説明する。

【 0 0 6 8 】

S S D 等のデバイス 1 0 0 へのデータアクセスは、ホストインタフェース 1 0 1 がコマンドを受け付けて、デバイス制御部 1 1 0 がそのコマンドを解釈して実行される（ステップ S 1 1 ）。

【 0 0 6 9 】

例えば書き込みコマンドのときには、書き込むデータは、当該コマンドと共にホストインタフェース 1 0 1 を介して送られる。このデータは、デバイス制御部 1 1 0 がアクセス可能な R A M （例えばワークメモリ 1 1 1 ）に格納される。

10

【 0 0 7 0 】

例えば、読み出しコマンドを実行する場合、デバイス制御部 1 1 0 は、ワークメモリ 1 1 1 に事前に読み出した L 2 P テーブル 1 3 2 a を用いて、当該コマンド内で指定された論理アドレスを物理アドレスに変換する（ステップ S 1 2 ）。

【 0 0 7 1 】

ワークメモリ 1 1 1 内に読み出されている L 2 P テーブル 1 3 2 a に、これから読む論理アドレスが存在しない場合、デバイス制御部 1 1 0 は、記憶部 1 3 0 に保存されている L 2 P テーブル 1 3 2 a を読み出して、ワークメモリ 1 1 1 に格納する（ステップ S 1 3 ）。その後、デバイス制御部 1 1 0 は、ワークメモリ 1 1 1 に格納した L 2 P テーブル 1 3 2 a を用いて、論理アドレスを物理アドレスに変換する。デバイス制御部 1 1 0 は、得られた物理アドレスを指定して、記憶部 1 3 0 内の物理ページにアクセスし、データを読み出す（ステップ S 1 4 ）。

20

【 0 0 7 2 】

上述のように、従来は K V S データも通常のデータとして管理されていた。このため、K V S データを読み出す際には、まず K V S のための管理ファイル（転置ファイル）を通常アクセスで L 2 P テーブル 1 3 2 a を使って読み出し、さらに管理ファイルを元に特定ファイル内に保存された K V S データを読み出していた。そして、K V S データを読み出す際にも、L 2 P テーブル 1 3 2 a を参照する必要があった。このため、結局 L 2 P テーブル 1 3 2 a へのアクセスを 2 回以上行う必要があった。

30

【 0 0 7 3 】

そこで、本実施形態では、L 2 P テーブル 1 3 2 a と同様のアドレス変換テーブルである K 2 P テーブル 1 3 2 b を用いて K V S データにアクセス可能とする。図 3 を用いて、K 2 P テーブル 1 3 2 b を使った K V S データへのアクセスの仕組みを説明する。

【 0 0 7 4 】

本実施形態のデバイス 1 0 0 へのデータアクセスは、ホストインタフェース 1 0 1 が K V S コマンドを受け付けて、デバイス制御部 1 1 0 がその K V S コマンドを解釈して実行される（ステップ S 1 1 ）。

【 0 0 7 5 】

例えば K V S コマンドが登録コマンド（P U T ）のときには、登録するデータは K V S コマンドと共にホストインタフェース 1 0 1 を介して送られて、デバイス制御部 1 1 0 がアクセス可能な R A M （例えばワークメモリ 1 1 1 ）に置かれる。登録するデータは、通常のデータと同じようにワークメモリ 1 1 1 に格納されてもよいし、記憶部 1 3 0 に格納する前のバッファとして他のメモリに格納されてもよい。説明を簡単化するためにワークメモリ 1 1 1 に格納されるものとする。

40

【 0 0 7 6 】

上述のように、本実施形態では、デバイス制御部 1 1 0 がデータの一部であるキーをアドレス（キーアドレス）に変換する仕組み（生成部 1 1 7 ）を有する。例えば、キーに対応するバリュウの取得要求に相当する取得コマンドを実行する場合、生成部 1 1 7 は、取得コマンド内で指定されたキーからキーアドレスを生成する（ステップ S 1 5 ）。このキ

50

ーアドレスを元に、デバイス制御部 110 は、ワークメモリ 111 内でキーアドレスと物理アドレスの変換を行う（ステップ S16）。

【0077】

ワークメモリ 111 内に読み出されている K2P テーブル 132b に、これから読むキーアドレスが存在しない場合、デバイス制御部 110 は、記憶部 130 に保存されている K2P テーブル 132b を読み出して、ワークメモリ 111 に格納する（ステップ S17）。その後、デバイス制御部 110 は、ワークメモリ 111 に格納した K2P テーブル 132b を用いて、キーアドレスを物理アドレスに変換する。デバイス制御部 110 は、得られた物理アドレスを指定して、記憶部 130 内の物理ページにアクセスし、KVS データを読み出す（ステップ S18）。

10

【0078】

本実施形態によれば、KVS データは K2P テーブル 132b で管理されているため、KVS データを読み出す際に、L2P テーブル 132a へのアクセスを省略して、直接記憶部 130 の物理アドレスを参照して読み出せるようになる。このため、KVS データに対するアクセス速度が向上する。

【0079】

図 4 - 1 は、K2P テーブル 132b のデータ形式の一例を示す図である。K2P テーブル 132b は、複数のエントリを含むテーブルデータの形式を取る。エントリは、キーアドレスと物理アドレスとを対応させたアドレス情報（K2P ペア）を少なくとも 1 つ含む。例えば、K2P テーブル 132b は、32 ビット（4 バイト）のキーアドレスを、32 ビット（4 バイト）の物理アドレスのペアである 8 バイトの K2P ペアを格納していく。アドレス長は一例であり、システムの規模に合わせて適宜変更してよい。

20

【0080】

図 4 - 1 では、K2P テーブル内にキーアドレスを格納している例を示している。図 4 - 2 のように、エントリのアドレス保存順をキーアドレス値に対応させてもよい。これにより、キーアドレスを保存する容量をなくすることができる。図 4 - 2 の例では、物理アドレス（4 バイト）だけでよいので、K2P テーブルの容量は、図 4 - 1 の場合に比べて半分になる。

【0081】

また、1 エントリあたりの K2P ペアの数は、K2P テーブル 132b にアクセスするデバイス制御部 110 の仕様、ワークメモリ 111 の仕様、および、記憶部 130 のページサイズなどから、速度や設計容易性などを考慮して決めればよい。例えば、K2P ペアのサイズが 8 バイトのとき、1 エントリあたり 8 KB で管理すると K2P ペアは 1000 個格納される。また例えば、1 エントリあたり 256 B で管理すると K2P ペアは 32 個格納される。

30

【0082】

K2P テーブル 132b は、キーアドレスの生成に伴い、任意の拡張性を持たせることができる。例えば、ホストシステム 200 からの KVS コマンドの要求によって、キーアドレスが生成された場合、生成順に K2P ペアを作成していくこともできる。

【0083】

この場合、K2P テーブル 132b 自体は小さくなるが、ランダムなキーアドレスが生成順に格納される。従って、このまま K2P ペアを検索すると検索時間が増加する可能性がある。このため、検索の高速化のために、K2P ペアの検索のためのテーブルをさらに用意してもよい。

40

【0084】

また反対に、K2P テーブル 132b は最初から固定の大きさに設定してもよい。そもそも、デバイス 100 の中で記憶部 130 の容量追加の可能性がない場合、すなわち拡張性がない場合、物理ページの総数は決まっている。このため、最初から K2P テーブル 132b の大きさを固定しておいてもよい。例えば K2P ペアが 8 バイトで 1 エントリあたり 8 KB のとき、1 エントリあたり 1000 ページ分の物理アドレスデータを格納してい

50

ることになる。記憶部 130 の記憶容量が 8 GB でページサイズが 8 KB のときには、1000000 ページが存在する。従って、K2P テーブル 132b は 1000 エントリを格納できればよい。すなわち K2P テーブル 132b の大きさは 8 MB となる。

【0085】

どの方式を採用するかは、デバイス 100 の規模および拡張性を考慮して決めることができる。

【0086】

次に、K2P テーブル 132b と L2P テーブル 132a の管理方法について図 5 および図 6 を用いて説明する。図 5 は、K2P テーブル 132b を、L2P テーブル 132a と独立に管理する例を示している。

10

【0087】

本実施形態では、デバイス 100 が、通常 of データを取り扱うための L2P テーブル 132a を搭載している場合、L2P テーブル 132a および K2P テーブル 132b に対して同様のデータ形式を用いる。これにより、デバイス制御部 110 のアルゴリズムや命令を共通化させて、必要なハードウェアの追加分を少なくすることができる。

【0088】

例えば、L2P テーブル 132a は、論理アドレス 32 ビット (4 バイト) と物理アドレス 32 ビット (4 バイト) の合計 8 バイトのペアを格納する。同様に、K2P テーブル 132b は、キーアドレス 32 ビット (4 バイト) と物理アドレス 32 ビット (4 バイト) の合計 8 バイトのペア (K2P ペア) を格納する。

20

【0089】

デバイス制御部 110 は、分類機能 401 を用いて、扱うべきアドレスが L2P テーブル 132a のアドレスであるか、または、K2P テーブル 132b のアドレスであるかを判断する。デバイス制御部 110 は、判断の後は、両者を同様に処理していくことができる。

【0090】

図 5 のように、例えば、アドレス値として「0x0000__1000」が参照される場合であって、分類機能 401 によって、通常 of データの論理アドレスであると判断された場合は、デバイス制御部 110 は L2P テーブル 132a を参照する。

【0091】

また、アドレス値として「0xF356__af14」が参照される場合であって、分類機能 401 によって、キーアドレスであると判断された場合は、デバイス制御部 110 は、K2P テーブル 132b を参照する。いずれかのテーブルを参照して物理アドレスが得られた後の処理は、基本的に L2P テーブル 132a も K2P テーブル 132b も同様である。

30

【0092】

分類機能 401 は、いくつかの方法で実現できる。1 つ目は、デバイス制御部 110 がホストシステム 200 からの要求 (コマンド) を保持しているときに、保持するコマンドが K2P コマンドであれば K2P テーブル 132b を読み、通常 of データアドレス指定のコマンドであれば L2P テーブル 132a を読む方法である。

【0093】

分類機能 401 の方法の 2 つ目は、分類のためのテーブル (分類テーブル) を設けておき、論理アドレスとキーアドレスのいずれを参照すべきか判断する方法である。分類テーブルを用いる場合は、論理アドレスとキーアドレスとが衝突しないようにキーアドレスの生成の段階で管理しておく必要がある。例えば、論理アドレスとして「0x0000__1000」が既に使用されていて、生成したキーアドレスが偶然「0x0000__1000」となったとする。この場合は、アドレスが衝突しているので、分類テーブル内でキーアドレスであることを判断して K2P テーブル 132b を読むようにする。分類機能 401 は、デバイス制御部 110 の設計要件に従い、製造者が選択することができる。

40

【0094】

分類テーブルは、例えばアドレス値と使用状況を示す値の対をエントリとして構成して

50

おく。アドレスが与えられると、分類テーブル内をサーチして、該当するアドレスが論理アドレスで使用されているか、物理アドレスで使用されているかを調べることができる。ただし、デバイス制御部 110 は予め通常のアドレス指定による読み書きか K V S コマンドなのかを把握しているので、分類テーブルは必ずしも必要ではない。

【0095】

図 6 は、K 2 P テーブル 132 b と L 2 P テーブル 132 a とを 1 つの全体テーブルとして管理する例を示している。

【0096】

例えば、32 ビットの論理アドレスのうち、L 2 P テーブル 132 a で使用する数を例えば「0x1000__0000」までのように所定サイズに制限しておく。そして、「0x1000__0001」以上のアドレスはキーアドレスであると判断する。境となるアドレス値は、デバイス制御部 110 が管理できるようにしてもよい。図 6 は、キーアドレスがそのまま全体テーブルの後半のアドレスに対応している例を示している。

10

【0097】

図 6 に例示するような方法では、分類機能 401 が、全体テーブルのアドレスが所定の境目よりも前半か後半かを判断することで、アクセスすべきテーブルが L 2 P テーブル 132 a および K 2 P テーブル 132 b のいずれであるかを知ることができる。

【0098】

図 6 の方法と、分類テーブルを用いる上述の方法とを組み合わせることもできる。全体テーブルとしては、メモリシステム内の総物理ページ分を格納できる大きさのテーブルを用意する。

20

【0099】

次に、生成部 117 が生成するキーアドレスの衝突について図 7 を用いて説明する。

【0100】

一般に K V S 方式では、キーの数に制限は設けない。このため、メモリシステム内に格納される K V S の数、言い換えるとキーの種類が多くなると、キーに対して生成されるキーアドレスが衝突するようになる。例えば、キーが「Blue」という単語のときのキーアドレスが「0x0000__41a9b」とであるとする。別のキーである「Car」という単語から作られるキーアドレスが偶然に「0x0000__41a9b」と一致してしまう確率はゼロではない。キーアドレスの生成に用いる Hash 関数に高度なものを用いて、数学的に疎な数を生成したとしても、キーアドレスの値を固定長サイズで小さく変換した際に、衝突の可能性が生じる。

30

【0101】

キーアドレスの衝突をできるだけ回避するための方法はいくつか考えられる。1 つ目は、キーアドレスをできるだけ長いままで用いる方法である。例えば、Hash 関数で変換したままの値をキーアドレスとして用いればよい。ただし、アドレス長が長いため、K 2 P テーブル 132 b が大きくなるので、その分だけ容量を消費するという問題、および、L 2 P テーブル 132 a とのデータ形式の整合性が失われるという問題が生じる。記憶部 130 の容量を十分に大きくし、キーの種類に対して十分大きな数の物理ページ数を用意できるならば、キーの衝突確率を下げることもできる。しかし、この場合であっても衝突確率をゼロにすることはできない。

40

【0102】

2 つ目は、任意長データを固定長変換方式として 2 つ以上の方式を組み合わせる方法である。例えば、キーアドレスを 32 ビット長にすると、16 ビット分を Hash 関数より生成し、ASCII コードなどによってキー自体を変換したバイナリデータで残り 16 ビット分を表し結合させる方法を用いることができる。前半 16 ビットはランダムな値だが後半 16 ビットがデータそのものに由来する値となるため、キーの衝突は極力低下させることができる。しかし、この方法でも数学的に衝突の可能性はゼロではなく、キーの数が増えていくと衝突の可能性は増える。

【0103】

50

結局、論理アドレスとは異なり、与えられるキーは任意長かつ無限のバリエーションを有していることから、キーから生成されるアドレス値は衝突を避けられない。

【 0 1 0 4 】

そこで、本実施形態では、衝突が発生した場合にも正しくバリューを読み出せる機能を備える。図 7 は、K 2 P テーブル 1 3 2 b と、物理ページに格納されたバリュー（バリューデータ）とを模式的に示す図である。

【 0 1 0 5 】

図 7 の上部には、「Key 1」が「Blue」であり、「Value 1」がそれに対応するコンテンツ（バリュー）「< contents 1 >」である K V S データの例が示されている。同様に、「Key 2」が「Car」であり、「Value 2」がそれに対応するコンテンツ「< contents 2 >」である K V S データの例が示されている。

【 0 1 0 6 】

生成部 1 1 7 が H a s h 関数により「Blue」と「Car」をそれぞれ変換してキーアドレスを生成した結果、これらのキーアドレス「0x0000_41a9b」が衝突したものとする。取得部 1 1 3 は、このキーアドレスに関連付けられた物理アドレスを参照し、バリューデータを読み出す。物理ページ内には「Key 1」=「Blue」と「Key 2」=「Car」にそれぞれ対応するバリューが保存されている。取得部 1 1 3 は、物理ページ全体を例えばワークメモリ 1 1 1 に読み出す。その後、デバイス制御部 1 1 0 またはホストシステム 2 0 0 が、読み出された物理ページを参照して、目的としていたキーに対応するバリューが保存されているかを判断する。例えば、目的とするキーが「Blue」であれば、デバイス制御部 1 1 0 またはホストシステム 2 0 0 は、読み出した物理ページ全体から、「Blue」に対応するバリュー「< contents 1 >」を取得できる。また、例えば、目的とするキーが「Car」であれば、デバイス制御部 1 1 0 またはホストシステム 2 0 0 は、読み出した物理ページ全体から、「Car」に対応するバリュー「< contents 2 >」を取得できる。なお、図 7 の例では、「< contents 2 >」は、「< contents 2 - 1 >」および「< contents 2 - 2 >」として、2 つのページに分割されて記憶されている。以下に述べるように、次のページを読むためのポインタを用いることで、分割されたバリューを連続して読み出すことができる。

【 0 1 0 7 】

すなわち、K V S では、キーおよびバリューは任意長であるため、1 つの物理ページ内にデータが収まるとは限らない。このため、図 7 に示したように、物理ページ内の特定箇所に次のページを読むためのポインタとなる記憶位置を示すアドレス（次ページアドレス：next page pointer（以下、次ページポインタという））を保存し、連続して読めるようにしておく。このように K V S データを格納する物理ページが複数に渡る場合は、その分だけ物理ページを消費する。次ページポインタは、N A N D フラッシュメモリを用いた記憶部 1 3 0 の場合は、1 つのページ内の冗長データ部または管理データ部と呼ばれる領域に格納しておくことができる。

【 0 1 0 8 】

次に、逆引きテーブルについて説明する。実データおよび K V S データの追加や修正を行う場合、それぞれ L 2 P テーブル 1 3 2 a および K 2 P テーブル 1 3 2 b の逆引きテーブルが必要となる。

【 0 1 0 9 】

記憶部 1 3 0 として N A N D フラッシュメモリを用いている場合、記憶セルは主に書き込みによって寿命が低下していく。このため、物理ページを均等に利用して記憶セルの寿命を最大限に利用する手続きが用いられる。物理ページを均等に利用して寿命を延ばすための技術は、ウェアレベリングと呼ばれる。N A N D フラッシュメモリは、一般的にページ単位で読み書きされる。一方、N A N D フラッシュメモリは、ブロック単位で消去が行われる。このため、特定のブロックにデータが集中すると、そのブロックの寿命が低下し、同時に信頼性も低下する。また、N A N D フラッシュメモリは通常、同一ページに対す

10

20

30

40

50

る追記書き込みができない仕様になっていることが多い。このため、ある物理ページに書き込まれたデータを変更する場合は、別の物理ページに変更したデータを書き込み、その物理ページのアドレス（物理アドレス）に対して論理アドレスを対応させる。

【0110】

このように、NANDフラッシュメモリを用いたメモリシステムでは、物理アドレスを論理アドレスに対応づけたP2Lテーブルを持つことが一般的である。この場合、新しく物理ページを割り当てる時に、いずれの物理ページが使用されていないか、または、いずれのページの書換え頻度が少ないかという履歴を元にして、いずれの物理ページを用いるかが決定される。

【0111】

本実施形態でも、P2Lテーブルを用いて物理ページを管理するのと同様の手法により、K2Pテーブル132bの逆引きテーブルであるP2Kテーブルを用いてKVSデータの物理ページを管理する。これにより、デバイス100の寿命と信頼性を向上させることができる。

【0112】

図8は、P2Kテーブルのデータ形式の一例を示す図である。P2Kテーブルは、物理アドレスとキーアドレスとのペアを各エントリに含む。

【0113】

各エントリには、物理アドレスが使用されていることを示す例えば1ビットの判定情報（フラグ）を持たせることができる。このフラグを参照することにより物理アドレスの使用状況が判別できる。図8に示すように、例えば、「0x0」および「0x1」の場合に、それぞれ物理アドレスが使用されていないこと、および、使用されていること、を表すフラグを用いることができる。

【0114】

なお、図8の判定情報は一例であり、これに限られるものではない。物理アドレスが使用されているか否か（物理アドレスのページが有効であるか否か）を表す情報であればどのような情報であってもよい。転記処理部116は、P2Kテーブルのフラグ（判定情報）を参照して、ガベージコレクションおよびコンパクションを実行する。例えば、転記処理部116は、P2Kテーブルのフラグが「0x1」である（使用されている（有効な））物理アドレスのページのデータを対象としてコンパクションを実行する。

【0115】

K2Pペアが作られたあとで、P2Kテーブルも作成しておくことで、物理ページの管理を容易に行うことが可能となる。各エントリが、物理アドレスに対応する物理ページの書換え回数（書換え頻度）を含んでもよい。書換え回数を記録しておけば、書換え回数の少ない物理ページを選んで使用する制御が可能となる。

【0116】

図9は、P2LテーブルとP2Kテーブルとを融合させたテーブル（P2L/P2Kテーブル132c）のデータ形式の一例を示す図である。このテーブルを参照することにより、物理アドレスが論理アドレスに対応付けられているのか、キーアドレスに対応付けられているのかを知ることができる。

【0117】

なお、図7に示すようにKVSデータを格納する物理ページが複数に渡る場合も、各物理ページの物理アドレスと、KVSデータの論理アドレスとをP2Kテーブルに記録して管理しておく。

【0118】

次に、このように構成された第1の実施形態にかかるデバイス100による各種処理について図10～図13を用いて説明する。図10は、登録コマンドを受け付けた場合の処理の一例を示すフローチャートである。

【0119】

受付部112が、登録コマンドを受け付けると、図10の処理が開始される。登録コマ

10

20

30

40

50

ンドは、例えば登録するKVSデータを含む。生成部117は、登録するKVSデータに含まれるキーをキーアドレスに変換する(ステップS101)。

【0120】

取得部113は、K2Pテーブル132bを参照し、キーアドレスが既にK2Pテーブル132bに存在していないか検索する(ステップS102)。取得部113は、キーアドレスがK2Pテーブル132b内に見つかったか否かを判断する(ステップS103)。見つかった場合(ステップS103: Yes)、取得部113は、キーアドレスに対応するバリューの物理アドレスを参照し(ステップS104)、バリューの物理ページに空きがあるかを判断する(ステップS105)。バリューの物理ページに空きがない場合(ステップS105: No)、取得部113は、次の物理アドレスへジャンプするためのポインタ(次ページポインタ)を格納して、その物理アドレスを参照する(ステップS106)。取得部113は、P2KテーブルおよびP2Lテーブルの少なくとも1つを参照して空いている物理アドレスを探し、ジャンプする物理アドレスを決定する。書込部115は、使用した物理アドレスをP2Kテーブルに登録する(ステップS107)。

10

【0121】

バリューの物理ページに空きがある場合(ステップS105: Yes)、書込部115は、登録するKVSデータに含まれるバリューを当該物理ページに追記する(ステップS108)。ただし、フラッシュメモリなどのように、同一ページ内への追記が禁止される場合がある。その場合は、一般的なSSDなどと同様に、別の物理アドレスの物理ページにまとめてデータ(バリュー)を書き込む。出力制御部114は、追記後のバリューのデータサイズを出力し(ステップS109)、処理を終了する。

20

【0122】

ステップS103で、キーアドレスがK2Pテーブル132b内に見つからなかった場合(ステップS103: No)、書込部115は、空いている物理アドレスの物理ページにバリューを追加する(ステップS110)。書込部115は、キーとバリューの物理アドレスとを関連付けてK2Pテーブル132bに登録する(ステップS111)。書込部115は、使用した物理アドレスをP2Kテーブルに登録する(ステップS112)。出力制御部114は、追記後のバリューのデータサイズを出力し(ステップS113)、処理を終了する。

【0123】

30

図11は、書込コマンドを受け付けた場合の処理の一例を示すフローチャートである。受付部112が、書込コマンドを受け付けると、図11の処理が開始される。書込コマンドは、例えばKVSデータを含む。書込コマンドは、既に存在するキーに対するバリューの追記コマンドである。

【0124】

ステップS201～ステップS209は、図10のステップS101～ステップS109までと同様であるため説明を省略する。

【0125】

ステップS103で、キーアドレスがK2Pテーブル132b内に見つからなかった場合(ステップS203: No)、取得部113は、キーが存在しないことを返して処理を終了する(ステップS210)。取得部113は、例えば、SIZE=0を返すことにより、キーが存在しないことを通知する。

40

【0126】

図12は、取得コマンドを受け付けた場合の処理の一例を示すフローチャートである。受付部112が、取得コマンドを受け付けると、図12の処理が開始される。取得コマンドは、例えばキーを含む。

【0127】

ステップS301～ステップS303は、図10のステップS101～ステップS103までと同様であるため説明を省略する。

【0128】

50

キーアドレスが見つかった場合（ステップS303：Yes）、取得部113は、キーアドレスに対応するバリュウの物理アドレスを参照し（ステップS304）、キーアドレスに対応するバリュウを読み出してワークメモリ111（またはバッファメモリ121）に格納する（ステップS305）。出力制御部114は、読み出したバリュウのデータサイズを出力し（ステップS306）、処理を終了する。

【0129】

ステップS303でキーアドレスがK2Pテーブル132b内に見つからなかった場合（ステップS303：No）、取得部113は、キーが存在しないことを返して処理を終了する（ステップS307）。取得部113は、例えば、SIZE=0を返すことにより、キーが存在しないことを通知する。

10

【0130】

図13は、読出コマンドを受け付けた場合の処理の一例を示すフローチャートである。受付部112が、読出コマンドを受け付けると、図13の処理が開始される。読出コマンドは、例えばサイズの指定を含む。

【0131】

取得部113は、集合（バリュウ）の要素のワークメモリ111の保管場所を参照し（ステップS401）、バリュウがこの保管場所内に見つかったか否かを判断する（ステップS402）。集合（バリュウ）の要素が見つからなければ（ステップS402：No）、取得部113は、出力として例えばS=NULLを返すことにより、集合（バリュウ）の要素が存在しないことを通知する（ステップS405）。見つかった場合（ステップS402：Yes）、指定されたサイズ分に対応する集合（バリュウ）の要素を読み出す（ステップS403）。取得部113は、読み出したバリュウの要素を出力し（ステップS404）、処理を終了する。

20

【0132】

図13の例ではサイズ指定のみだが、ワークメモリ111内の場所（アドレス）を指定して読み出してもよい。

【0133】

なお、実際の手順やコマンドは図10～図13に示す例に限定されない。例えば、複数のキーが見つかった場合には、キーが見つかったことを示すフラグをたてておき、後でまとめてバリュウを読むなどの手順を実行可能としてもよい。

30

【0134】

以上のように、本実施形態では、ホストインタフェース101、または、ホストインタフェース101を介してデバイス制御部110やメモリ制御部120などがコマンドを受け付けてKVSの一連の処理を実行することができる。

【0135】

（変形例1）

変形例1では、物理ブロックテーブルを用いる例を説明する。図14は、物理ブロックテーブルを用いる場合のデータアクセスの仕組みを説明する図である。

【0136】

図14は、論理アドレスから変換された物理アドレスを、さらに物理ブロックとページオフセットとに変換する物理ブロックテーブル1401を用いる例を示している。図15は、物理ブロックテーブルのデータ形式の一例を示す図である。図15の物理ブロックテーブルは、物理アドレスから、当該物理アドレスのページが対応する物理ブロックを特定する際に利用される。このような物理ブロックテーブルを有しておくことで、例えばNANDフラッシュメモリを記憶部130として用いたデバイス100であれば、ガベージコレクションおよびコンパクションを効率的に行うことができる。

40

【0137】

本実施形態のK2Pテーブル132bから参照されるのは物理アドレスである。このため、物理ブロックテーブルを用いることにより、KVSデータのガベージコレクションとコンパクションを、通常のL2P形式のデータ（実データ）と同様に取り扱うことができ

50

る。従って、K 2 P と L 2 P が同居したデバイス 1 0 0 であっても、比較的容易に信頼性の高いシステムを作ることができる。

【 0 1 3 8 】

(変形例 2)

変形例 2 では、多段階の検索テーブルを用いて L 2 P テーブルへアクセスする例を説明する。図 1 6 に示すように、分類機能 4 0 1 により論理アドレス (実データ) とキーアドレス (K V S データ) のいずれを参照するか分類した後、さらに 1 以上の検索テーブルを用いて、論理アドレスに対応する物理アドレスを参照するように構成してもよい。

【 0 1 3 9 】

L 2 P テーブルは、記憶部 1 3 0 の全ページ情報を格納しているため、ワークメモリ 1 1 1 の容量と比較して大きなサイズとなる。デバイス制御部 1 1 0 は、アドレス指定の読み出し・書き込み命令を受けた時に、目的の論理アドレスを格納する L 2 P テーブルのエントリを第 1 のメモリブロックから探す必要がある。

【 0 1 4 0 】

例えば、記憶部 1 3 0 の容量が 6 4 G B のとき、1 ページ容量が 4 K B だとすると、1 6 0 0 0 0 0 0 ページ存在することになる。アドレス単位が 3 2 ビット (4 バイト) であるとするれば、L 2 P テーブルの容量は 6 4 M B になる。ワークメモリ 1 1 1 は一般的に S R A M で構成されるので、全 L 2 P テーブルを格納することができない。従って、目的とする L 2 P エントリを記憶部 1 3 0 から効率よく探すためには、L 2 P エントリを探すための検索テーブルを用いることができる。検索テーブルをツリー方式でたどって目的まで到達する。L 2 P エントリのエントリ数とワークメモリ 1 1 1 の容量に応じて、検索テーブルを多段にする。検索テーブルを多段にすると、検索テーブルの読み出し回数が増えるため、その分 L 2 P 処理が遅くなる場合もあるため、適当な段数を用いる。

【 0 1 4 1 】

上記実施形態では、K 2 P テーブル 1 3 2 b を単独で用いることができるため、または、L 2 P テーブル 1 3 2 a と共存させることが容易なため、ユーザは K V S データに対する処理によるシステムの負荷増大を感じることがない。

【 0 1 4 2 】

共存させる場合であっても、図 1 6 のように、K 2 P テーブル 1 3 2 b の参照が L 2 P テーブル 1 3 2 a の参照よりも少ない回数で行うことができるならば、K V S データの方がアクセス速度を速くすることができる。

【 0 1 4 3 】

なお、K V S のデータ形式を採用している時点で、要求しているキーに対応するバリュエーのデータが直ちに得られるので検索が速くなる。本変形例では、K V S データへのアクセス自体が余計な L 2 P 参照の手順を省略できるため、さらに高速になる。L 2 P テーブルの管理と同様の仕組みを用いているため、K V S データを格納することのハードウェアコストの増大はほとんど無くすることができる。

【 0 1 4 4 】

(変形例 3)

図 4 では、キーアドレスと物理アドレスの対応を複数個まとめたものを 1 つのエントリとした K 2 P テーブル 1 3 2 b の例を示した。変形例 3 では、これを拡張し、物理アドレスのあとにバリュエーのハッシュ値を格納する例を示す。図 1 7 は、変形例 3 の K 2 P テーブル 1 3 2 b のデータ形式の一例を示す図である。

【 0 1 4 5 】

ホストシステム 2 0 0 から K V S 要求の、例えば、G E T コマンドと集合演算命令の A N D がデバイス 1 0 0 に届いたとき、デバイス 1 0 0 側はハッシュ値を含む K 2 P テーブル 1 3 2 b を参照することで、バリュエーの中身とその集合演算を予め記憶部 1 3 0 から読み出す前に判断することができる。無駄な読み出しが減るため、検索および集合演算に係る時間が短縮できる。

【 0 1 4 6 】

10

20

30

40

50

KVSではこの例のように、1つのキーに対し、複数のバリューが付与される場合がある。バリューは、物理アドレスが指し示す、記憶部130のページ内にそのまま保存される。図17の例では、各バリューのハッシュ値も、K2Pテーブル132bにまとめて保存しておく。こうしておくことで、バリューを読み出すことなく、各バリューが一致しているか否かを、K2Pテーブル132bを用いて判別することができる。

【0147】

前述したようにハッシュ値は衝突している可能性があるので、ハッシュ値が同一であってもバリューを読み出したあとに、実際に同一なデータかどうかは比較照合しなければならない。しかし、同一データであって異なるハッシュ値であることはないので、ハッシュ値比較の段階で、まったく条件に一致していないデータは除外しておくことができる。この仕組みにより、余計なバリューの読み出しが少なくなり、NANDフラッシュメモリのように比較的読み出し速度が遅いRAMの場合は、検索速度が速くなる。

10

【0148】

図17の例のように、K2Pテーブル132b内にバリューのハッシュ値が格納されている場合、そのハッシュ値と、ハッシュ値をアドレスとして格納できるRAMと、を使って、ハッシュ検索を行うことができる。例えば、ハッシュ値をアドレスとして、該当するアドレスにデータが書き込まれる。

【0149】

例えばアドレスごとに4バイト格納できるRAMであれば、「value1」の値として「0x1010_1010」を書く。次に「value2」の値として、「0x0101_0101」をXORで上書きして書き込む。同じアドレスに対して書き込んだ場合、そのアドレスのデータは「0x1111_1111」になる。これを読めば一致していることが判断できる。

20

【0150】

ただし、上述のようにハッシュ値は衝突している可能性があるので、衝突の有無を検証する必要がある。バリューを参照して確かめれば確実であるが、ここでは図18を用いて別の方法を例示する。各バリューは2種類以上のハッシュ関数で変換した値を持つとする。図18の例は、2種類のハッシュ関数を用いる例である。1種類だけでは衝突していたバリューも、異なるハッシュ関数で変換した場合は異なるハッシュ値になる確率が高くなる。このため、衝突の可能性を可及的に減らすことができる。ただし、2種類以上のハッシュ値を格納することはK2Pテーブル132bの肥大化を招くため、バリューの一致判定の目的に合わせて設計する必要がある。なお、ハッシュ値が一致している条件は集合演算のAND条件に相当する。

30

【0151】

このように、第1の実施形態にかかる半導体記憶装置では、KVSデータを取り出す処理を、不揮発性メモリのアドレス管理方式と共存させることができる。これにより、KVS併用による書き込みなどデータ管理の負荷を増大させずに、読み出しではK2Pテーブルを用いてキーから直接的に物理アドレスを参照できる。このため、従来の方法では必要であった途中のL2P処理（例えばL2Pテーブルへのアクセス）を省略し、簡素かつ高速に検索を行うことができる。

【0152】

40

（第2の実施形態）

第2の実施形態にかかる半導体記憶装置は、CAM（Content Addressable Memory）を用いてデータを検索可能とする。図19は、第2の実施形態にかかるデバイス100-2のハードウェア構成の一例を示すブロック図である。図19に示すように、デバイス100-2は、ホストインタフェース101と、デバイス制御部110と、メモリ制御部120-2と、記憶部130と、を備えている。

【0153】

第2の実施形態では、メモリ制御部120-2がCAM122-2をさらに備える点が第1の実施形態と異なっている。その他の構成は第1の実施形態の図1と同様であるため説明を省略する。

50

【 0 1 5 4 】

記憶部 1 3 0 が N A N D フラッシュメモリである場合は、記憶部 1 3 0 内に存在する読み出し / 書き込みのためのバッファメモリが C A M であってもよい。すなわち、記憶部 1 3 0 から読み出されたデータが、デバイス 1 0 0 - 2 内のバス 1 0 2 を通って、デバイス制御部 1 1 0 が管理するワークメモリ 1 1 1 に到達する前に、C A M 動作によってデータの照合が行われる構成であればよい。

【 0 1 5 5 】

上述したように、キーの衝突が問題となる場合、ページ内からデータを読み出して、そのページ内に特定のキーが保存されているか判断する機能が必要となる。また同様に、ページが複数に渡る場合も、例えばページの冗長データ部に保存されている次ページポインタを判別して次ページを読むための制御信号を出力するための機能が必要となる。

10

【 0 1 5 6 】

本実施形態で用いられる C A M 1 2 2 - 2 はこのために利用される。C A M 1 2 2 - 2 は、予め読み出された K V S データを記憶する。C A M 1 2 2 - 2 は、読み出された K V S データにキーが含まれていれば、正常値として当該キーに対応するバリュースデータをワークメモリ 1 1 1 へ転送する。C A M 1 2 2 - 2 は、キーが含まれていなければ、エラー信号をデバイス制御部 1 1 0 に返す。

【 0 1 5 7 】

この仕組みにより、デバイス制御部 1 1 0 が、例えばワークメモリ 1 1 1 内のバリュースデータから K V S データを検索することが不要になり、K V S 動作がより円滑に行われるようになる。次ページポインタの検索もまた同様である。

20

【 0 1 5 8 】

例えば、次ページポインタはページの最後に付与されており、実データ部、または、管理データ部の中に格納されている。いずれの場合も次ページポインタの存在を示す特定のデータ（特定情報）を、同様に C A M 1 2 2 - 2 を用いて検索することができる。次ページポインタの発見をメモリ制御部 1 2 0 へ通知することで、例えばメモリ制御部 1 2 0 は、次ページポインタの示すアドレスを続けて読みに行くことができる。次ページポインタの発見をデバイス制御部 1 1 0 へ通知する場合も同様であって、いずれの場合も次ページポインタの存在をページ内データから読み取る方式よりは速度が向上する。

【 0 1 5 9 】

30

図 2 0 は、C A M 1 2 2 - 2 を用いた検索処理例を説明する図である。例えば、キー「C a r」が入力された場合、C A M 1 2 2 - 2 は、一致するキー「C a r」を記憶しているため、このキー「C a r」に対応するバリューである「< c o n t e n t s 2 - 1 >」を出力する。また、例えば、次ページポインタが記憶されていることを発見した場合、C A M 1 2 2 - 2 は、次ページポインタが示す位置に記憶されているバリューを出力する。

【 0 1 6 0 】

なお、特定のデータ（特定情報）は、次ページポインタの存在を示す情報に限られるものではない。当該特定情報に応じて実行する処理が予め定められた情報であればどのような情報であってもよい。

40

【 0 1 6 1 】

（第 3 の実施形態）

第 3 の実施形態にかかる半導体記憶装置は、ワークメモリ以外に、ワークメモリよりも大きなサイズのバッファメモリをさらに備える。図 2 1 は、第 3 の実施形態にかかるデバイス 1 0 0 - 3 のハードウェア構成の一例を示すブロック図である。図 2 1 に示すように、デバイス 1 0 0 - 3 は、ホストインタフェース 1 0 1 と、デバイス制御部 1 1 0 と、メモリ制御部 1 2 0 と、記憶部 1 3 0 と、に加えて、バッファメモリ 1 4 0 - 3 を備えている。

【 0 1 6 2 】

バッファメモリ 1 4 0 - 3 は、ワークメモリ 1 1 1 よりサイズが大きいメモリである。

50

バッファメモリ 140 - 3 は、デバイス制御部 110 からバス 102 を介してアクセスすることができる。バッファメモリ 140 - 3 は、例えば DRAM や MRAM や PCRAM など、NAND フラッシュメモリよりも容量は小さいが高速な RAM により構成できる。

【0163】

デバイス制御部 110 は、記憶部 130 に格納されている K2P テーブル 132 b および P2K テーブル (P2L / P2K テーブル 132 c) などの管理テーブルをバッファメモリ 140 - 3 にすべて転送しておく。デバイス制御部 110 は、バッファメモリ 140 - 3 上で、データのアクセスや変更を行う。これにより、記憶部 130 から都度読み書きするよりも高速に K2P 処理を行うことが可能となる。

【0164】

なお、バッファメモリ 140 - 3 に対しても、第 2 の実施形態の CAM 122 - 2 と同様の CAM を備えるように構成してもよい。

【0165】

(第 4 の実施形態)

第 4 の実施形態にかかる半導体記憶装置は、ダイレクトメモリアクセスコントローラ (DMAC) をさらに備える。図 22 - 1 は、第 4 の実施形態にかかるデバイス 100 - 4 のハードウェア構成の一例を示すブロック図である。図 22 - 1 に示すように、デバイス 100 - 4 は、ホストインタフェース 101 - 4 と、デバイス制御部 110 と、メモリ制御部 120 と、記憶部 130 と、に加えて、DMAC 150 - 4 を備えている。

【0166】

DMAC 150 - 4 は、デバイス 100 - 4 内でホストインタフェース 101 - 4 へデータを転送することを可能とする。DMAC 150 - 4 は、例えば、記憶部 130 内の L2P テーブル 132 a、K2P テーブル 132 b、および、P2L / P2K テーブル 132 c をホストインタフェース 101 - 4 に転送する。ホストインタフェース 101 - 4 は、デバイス 100 - 4 内からの L2P テーブル、K2P テーブルの転送要求を受け付け、メインメモリ 202 - 4 に転送する。ホストインタフェース 101 - 4 は、ホストシステム 200 - 4 に DMAC が備わっていればこれを利用することができる。ホストシステム 200 - 4 は、転送したテーブルにアクセスすることにより、記憶部 130 から都度読み書きするよりも高速に K2P 処理を行うことが可能となる。

【0167】

なお、上述のように、L2P テーブル 132 a および K2P テーブル 132 b に対して同様のデータ形式を用いれば、DMAC 150 - 4 およびホストインタフェース 101 - 4 のアルゴリズムや命令も共通化させることができる。これにより、必要なハードウェアの追加分を少なくすることができる。

【0168】

(第 4 の実施形態の変形例)

第 4 の実施形態の変形例は、ダイレクトメモリアクセスコントローラ (DMAC) にホストとデバイスを繋ぐ別の通信線 300 をさらに備える。図 22 - 2 は、第 4 の実施形態の変形例のデバイス 100 - 5 のハードウェア構成の一例を示すブロック図である。図 22 - 2 に示すように、デバイス 100 - 5 は、ホストインタフェース 101 - 4 と、デバイス制御部 110 と、メモリ制御部 120 と、記憶部 130 と、に加えて、DMAC 150 - 5 を備えている。また、DMAC 150 - 5 は、デバイス 100 - 5 からホストシステム 200 - 4 へのホストインタフェース 101 - 4 とは別の通信線 300 を介してホストシステム 200 - 4 と接続される。

【0169】

DMAC 150 - 5 は、通信線 300 を介して、デバイス 100 - 4 内からホストシステム 200 - 4 側のメインメモリ 202 - 4 にアクセスすることを可能とする。DMAC 150 - 5 は、例えば、記憶部 130 内の L2P テーブル 132 a、K2P テーブル 132 b、および、P2L / P2K テーブル 132 c をメインメモリ 202 - 4 に転送する。ホストシステム 200 - 4 は、転送したテーブルにアクセスすることにより、記憶部 13

10

20

30

40

50

0 から都度読み書きするよりも高速に K 2 P 処理を行うことが可能となる。

【 0 1 7 0 】

なお、上述のように、L 2 P テーブル 1 3 2 a および K 2 P テーブル 1 3 2 b に対して同様のデータ形式を用いれば、D M A C 1 5 0 - 4 のアルゴリズムや命令も共通化させることができる。これにより、必要なハードウェアの追加分を少なくすることができる。

【 0 1 7 1 】

(第 5 の実施形態)

第 5 の実施形態では、ホストシステムが、デバイスと同様の K 2 P 処理の実行機能 (サブ制御部) を備える。図 2 3 は、第 5 の実施形態にかかるデバイス 1 0 0 - 4 と、ホストシステム 2 0 0 - 5 のハードウェア構成の一例を示すブロック図である。デバイス 1 0 0 - 4 の構成は第 4 の実施形態 (図 2 2 - 1) と同様である。

10

【 0 1 7 2 】

図 2 3 に示すように、ホストシステム 2 0 0 - 5 は、さらにサブ制御部 2 2 0 - 5 を備える点が、第 4 の実施形態と異なっている。

【 0 1 7 3 】

サブ制御部 2 2 0 - 5 は、例えばデバイス制御部 1 1 0 の機能のうち、少なくとも K 2 P 処理に必要な機能と同様の機能を備えればよい。例えば、サブ制御部 2 2 0 - 5 は、キーに対応するバリュウの取得要求を受け付ける機能 (受付部 1 1 2 と同様の機能) を備える。また、例えばサブ制御部 2 2 0 - 5 は、メインメモリ 2 0 2 - 4 から各種データを読み出す機能 (取得部 1 1 3 と同様の機能) を備える。また、例えばサブ制御部 2 2 0 - 5 は、メインメモリ 2 0 2 - 4 に対して各種データを書き込む機能 (書込部 1 1 5 と同様の機能) を備える。また、例えばサブ制御部 2 2 0 - 5 は、読み出されたバリュウを取得要求に対する応答として出力する機能 (出力制御部 1 1 4 と同様の機能) を備える。

20

【 0 1 7 4 】

このような構成により、ホストシステム 2 0 0 - 5 の C P U 2 0 1 が、メインメモリ 2 0 2 - 4 内の K 2 P テーブルを直接参照できるようになる。C P U 2 0 1 は、キーの有無を知った上でのデバイス 1 0 0 - 4 への K V S 要求を送信することができる。

【 0 1 7 5 】

デバイス 1 0 0 - 4 内のデバイス制御部 1 1 0 と連携して、ホストシステム 2 0 0 - 4 が K V S 要求を送信する前に、予め決められたルールでデバイス 1 0 0 - 4 側のデータをメインメモリ 2 0 2 - 4 に転送するように構成してもよい。予め決められたルールとしては、例えばホストシステム 2 0 0 - 4 側で特定のキーに対するアクセスが頻発しているときに、デバイス 1 0 0 - 4 内の K V S データをメインメモリ 2 0 2 - 4 に転送してデータキャッシュするといったルールである。

30

【 0 1 7 6 】

本発明のいくつかの実施形態を説明したが、これらの実施形態は、例として提示したものであり、発明の範囲を限定することは意図していない。これら新規な実施形態は、その他の様々な形態で実施されることが可能であり、発明の要旨を逸脱しない範囲で、種々の省略、置き換え、変更を行うことができる。これら実施形態やその変形は、発明の範囲や要旨に含まれるとともに、特許請求の範囲に記載された発明とその均等の範囲に含まれる。

40

【 0 1 7 7 】

例えば、デバイス制御部 1 1 0 が、メモリ管理ユニット (M M U (Memory Management Unit)) を備えるように構成してもよい。一般にメモリ管理ユニットは、仮想アドレス (論理アドレス) と物理アドレスとを変換する機能などを備える。例えば、L 2 P テーブルおよび K 2 P テーブルなどを M M U 内に記憶し、M M U 内のテーブルを参照して上記実施形態の手法を適用するように構成できる。

【 0 1 7 8 】

また、デバイス制御部 1 1 0 が、T L B (Translation Lookaside Buffer) を備えるように構成してもよい。T L B は、仮想アドレスから物理アドレスへの変換の高速化を図

50

るための専用キャッシュである。例えば、L 2 P テーブルおよび K 2 P テーブルなどを T L B 内に記憶し、T L B 内のテーブルを参照して上記実施形態の手法を適用するように構成できる。

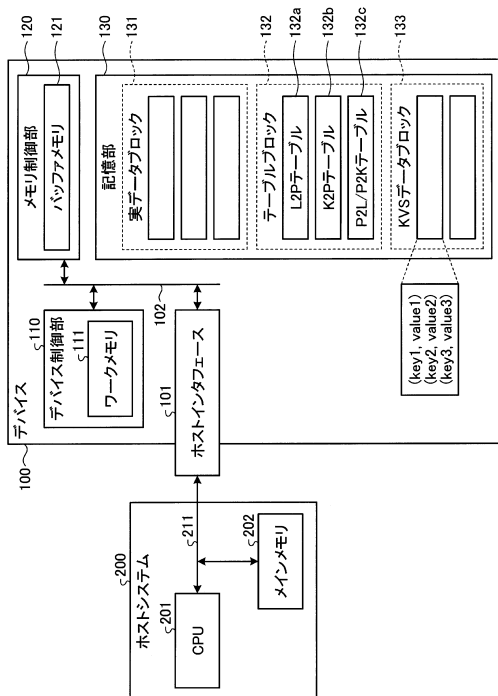
【符号の説明】

【 0 1 7 9 】

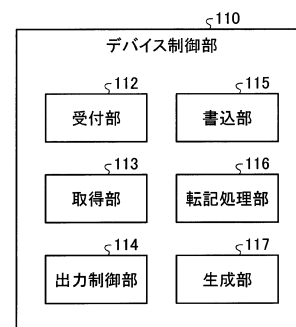
- | | |
|-------|------------|
| 1 0 0 | デバイス |
| 1 0 1 | ホストインタフェース |
| 1 0 2 | バス |
| 1 1 0 | デバイス制御部 |
| 1 1 1 | ワークメモリ |
| 1 2 0 | メモリ制御部 |
| 1 2 1 | バッファメモリ |
| 1 3 0 | 記憶部 |
| 2 0 0 | ホストシステム |
| 2 0 1 | C P U |
| 2 0 2 | メインメモリ |
| 2 1 1 | バス |

10

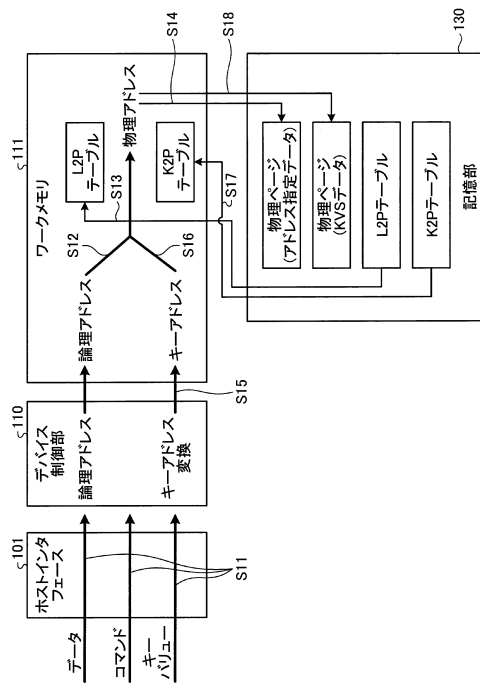
【 図 1 】



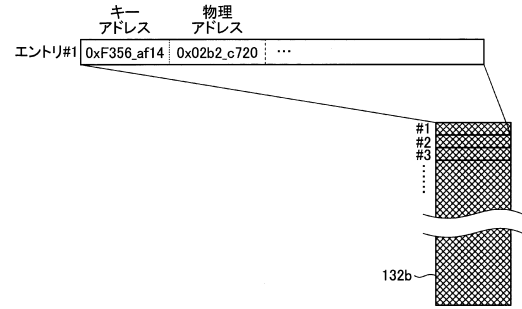
【圖 2】



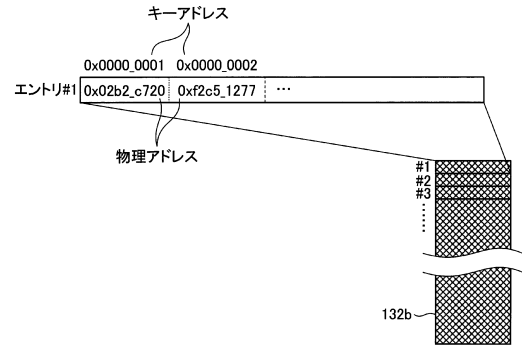
【図 3】



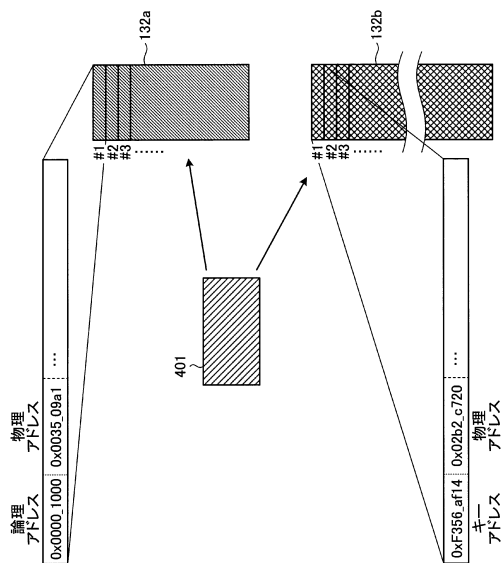
【図 4 - 1】



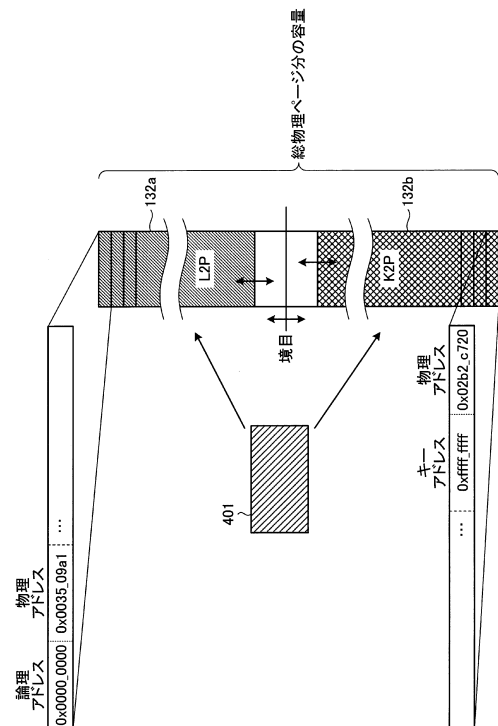
【図 4 - 2】



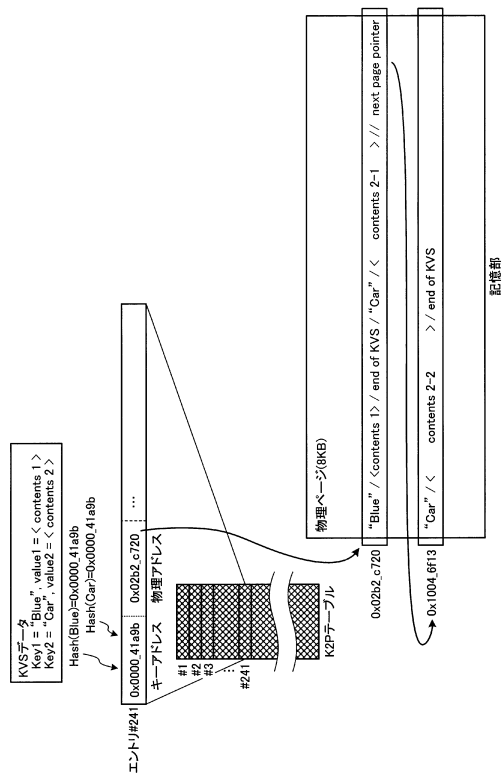
【図 5】



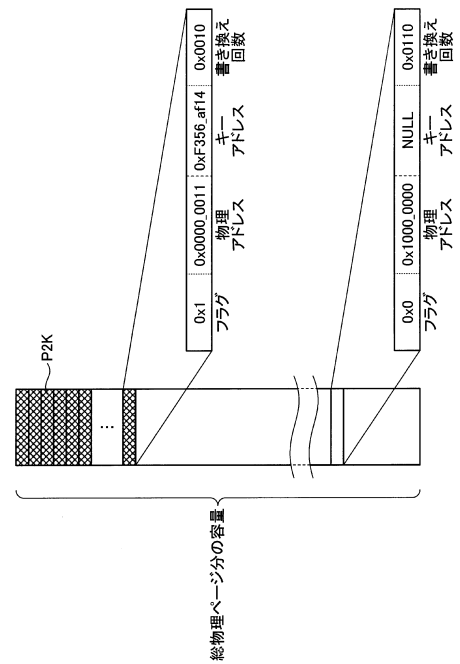
【図 6】



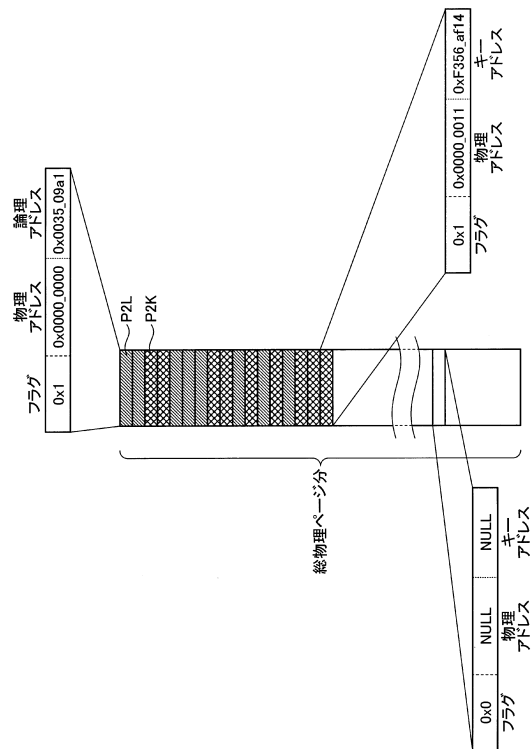
【圖 7】



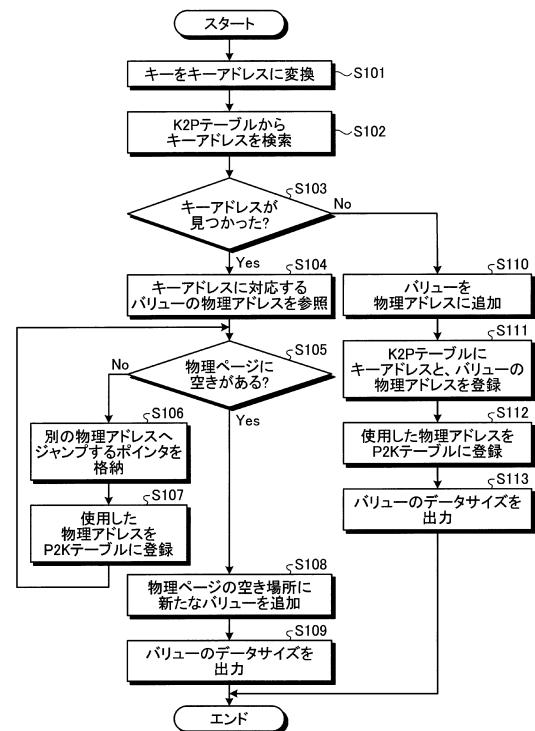
【 図 8 】



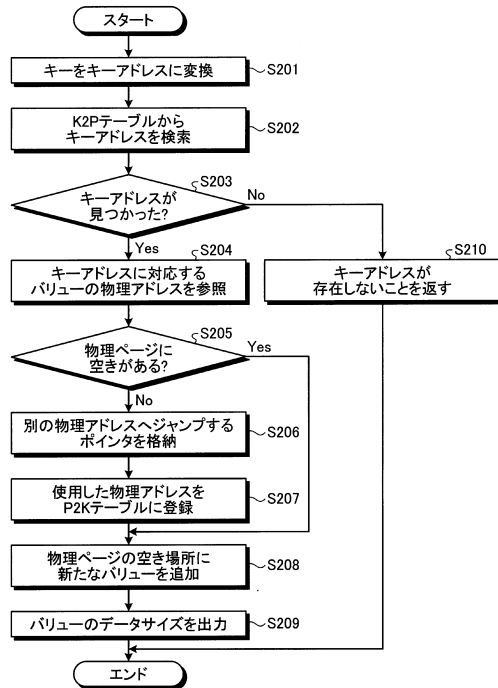
【 図 9 】



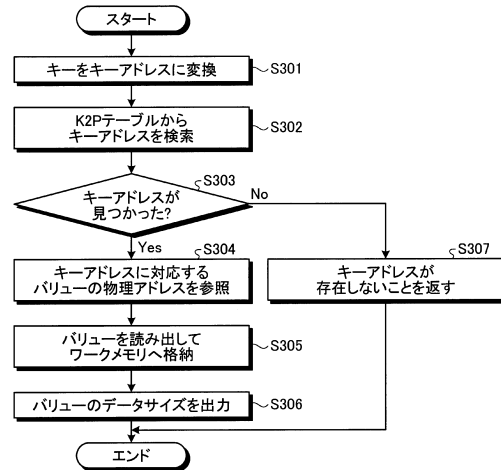
【 図 1 0 】



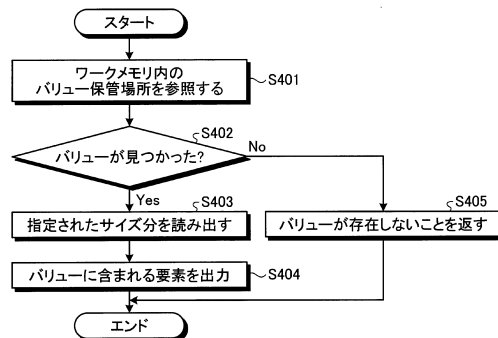
【図 1 1】



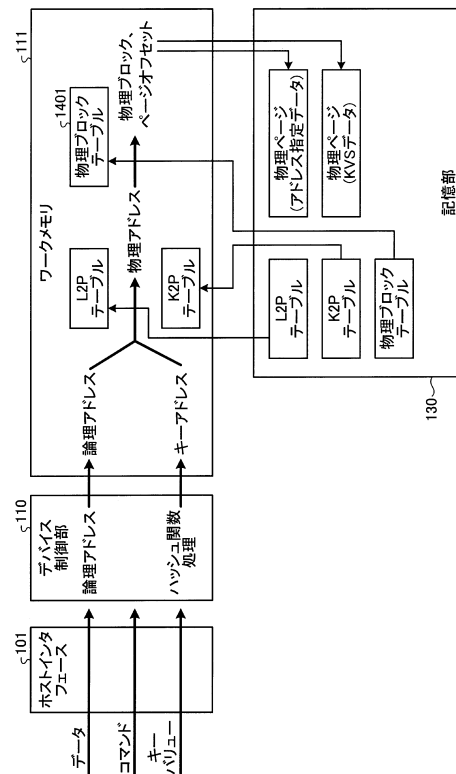
【図 1 2】



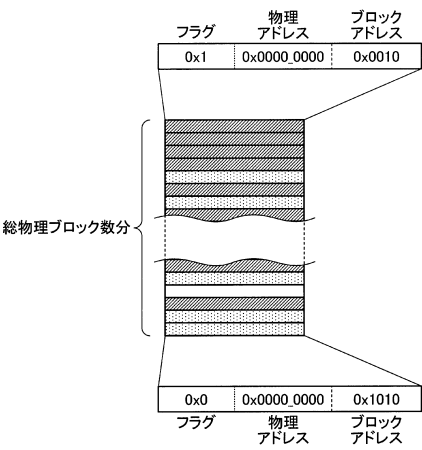
【図 1 3】



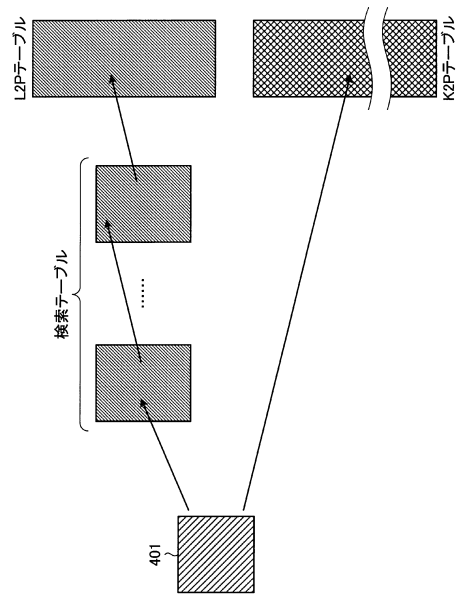
【図 1 4】



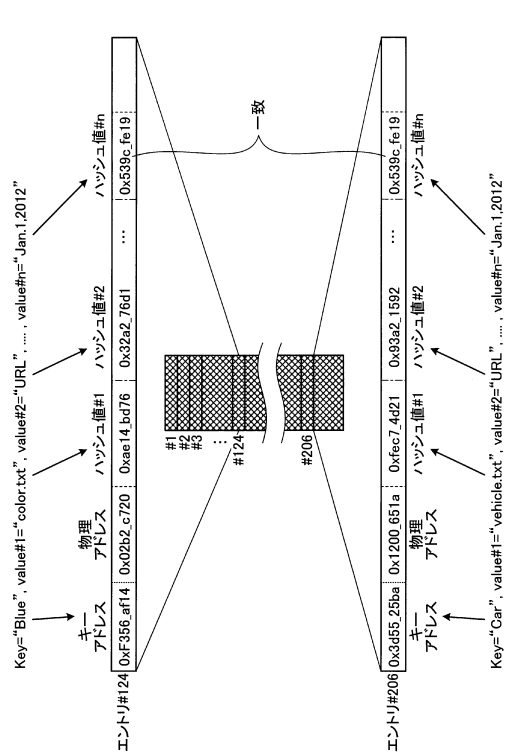
【図 15】



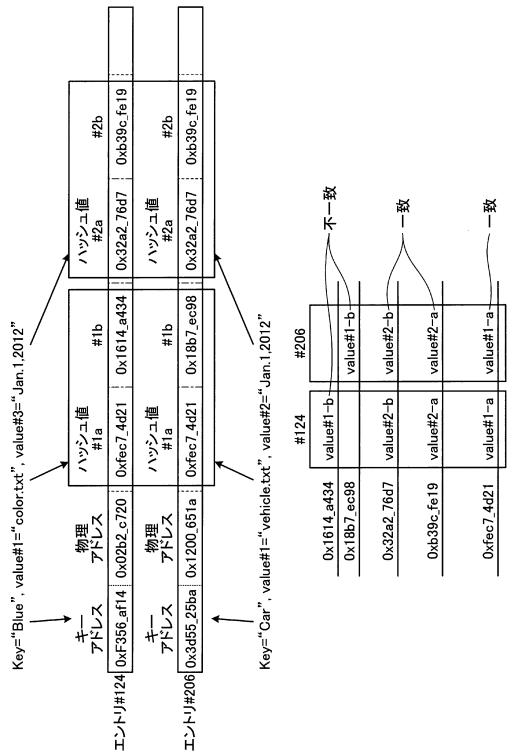
【図 16】



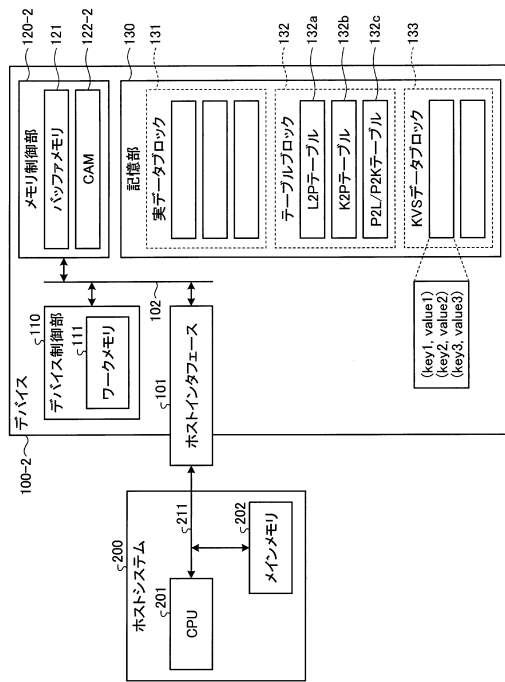
【図 17】



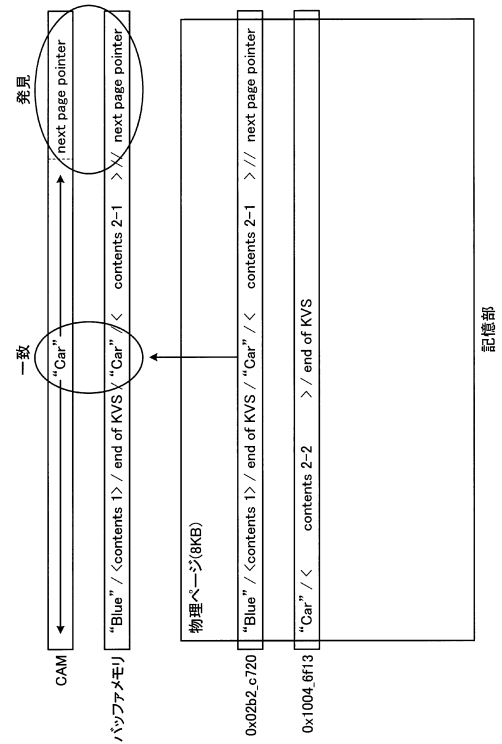
【図 18】



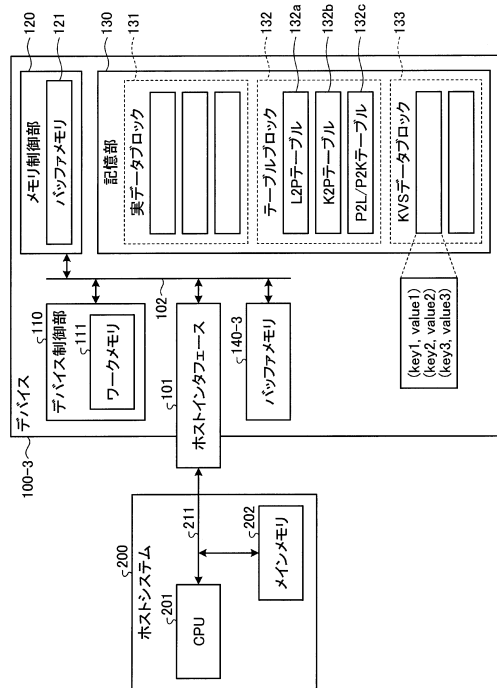
【図 19】



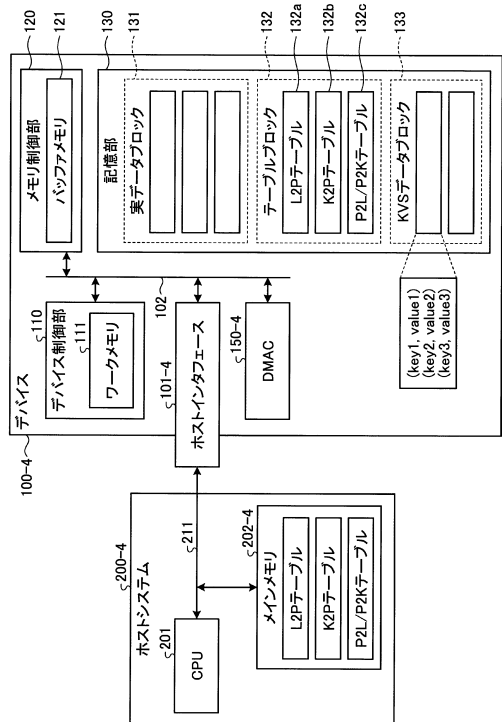
【図 20】



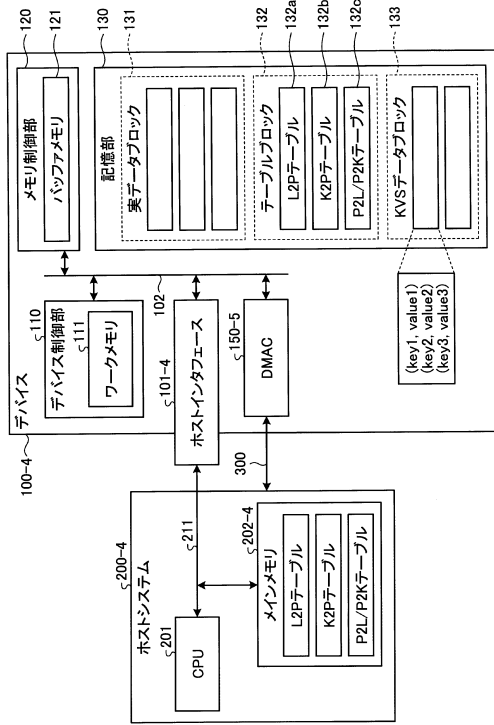
【図 21】



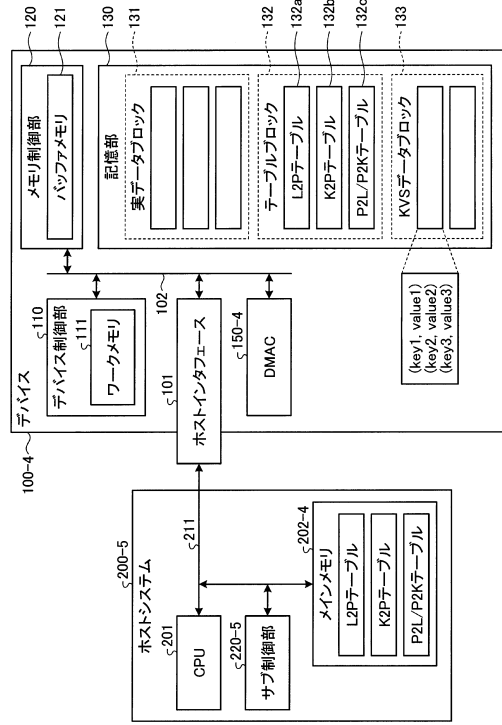
【図 22 - 1】



【図 22 - 2】



【図 23】



フロントページの続き

(72)発明者 栗田 貴宏
東京都港区芝浦一丁目1番1号 株式会社東芝内

審査官 池田 聡史

(56)参考文献 特開2004-334273(JP,A)
特開2011-123601(JP,A)
国際公開第2010/114006(WO,A1)
特開2001-067258(JP,A)
特開平09-282111(JP,A)
特開平07-028690(JP,A)
特開2000-047932(JP,A)
国際公開第2011/013054(WO,A1)
国際公開第2006/067923(WO,A1)
特開2011-227802(JP,A)
特開2001-188686(JP,A)
特開2007-310823(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 12/00
G06F 17/30