

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6764362号
(P6764362)

(45) 発行日 令和2年9月30日(2020.9.30)

(24) 登録日 令和2年9月15日(2020.9.15)

(51) Int.Cl.	F I	
G06F 16/174 (2019.01)	G06F 16/174	
G06F 16/00 (2019.01)	G06F 16/00	
G06F 12/02 (2006.01)	G06F 12/02	530C
G06F 16/172 (2019.01)	G06F 16/172	

請求項の数 20 (全 25 頁)

(21) 出願番号	特願2017-58521 (P2017-58521)	(73) 特許権者	390019839
(22) 出願日	平成29年3月24日 (2017.3.24)		三星電子株式会社
(65) 公開番号	特開2017-182803 (P2017-182803A)		Samsung Electronics Co., Ltd.
(43) 公開日	平成29年10月5日 (2017.10.5)		大韓民国京畿道水原市靈通区三星路129
審査請求日	令和2年2月13日 (2020.2.13)		129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, Republic of Korea
(31) 優先権主張番号	62/314, 918	(74) 代理人	110000051
(32) 優先日	平成28年3月29日 (2016.3.29)		特許業務法人共生国際特許事務所
(33) 優先権主張国・地域又は機関	米国 (US)	(72) 発明者	サラ, フレデリック
(31) 優先権主張番号	15/161, 136		アメリカ合衆国 90046 カリフォルニア州 ロサンゼルス ノース カーソン
(32) 優先日	平成28年5月20日 (2016.5.20)		アベニュー 1300 アパート4
(33) 優先権主張国・地域又は機関	米国 (US)		最終頁に続く
早期審査対象出願			

(54) 【発明の名称】 メモリの重複除去方法及び重複除去DRAMメモリモジュール

(57) 【特許請求の範囲】

【請求項1】

複数のハッシュテーブル (Hash table) の各々が、ハッシュ関数に対応し、前記複数のハッシュテーブルの各々が、複数の物理的なハッシュバケット (Hash bucket) を含み、各々の物理的なハッシュバケットは、複数のハッシュウェイ (Ways) を含み、データブロックを格納する、前記複数のハッシュテーブルを識別する段階と、

複数の仮想バケット (Virtual bucket) の各々が、前記複数の物理的なハッシュバケットの中の一部を含み、前記複数の仮想バケットの中で他の仮想バケットと少なくとも1つの前記物理的なハッシュバケットを共有する、前記複数の仮想バケットを識別する段階と、

前記複数の仮想バケットの中で対応する1つの仮想バケットに割当てられて格納されたデータブロックを有する前記複数の物理的なハッシュバケットの各々を識別する段階と、

ハッシュ値 (Hash value) を生成するために複数のハッシュ関数の中で対応するハッシュ関数にしたがってデータライン (Data line) をハッシング (Hashing) する段階と、

対応するハッシュテーブルの前記複数の仮想バケットの中で対応する仮想バケットが前記ハッシュ値にしたがってデータブロックのための使用可能な空間を有するか否かを決定する段階と、

前記複数の仮想バケットの中で対応する仮想バケットが使用可能な空間を有しない場合

、前記仮想バケットの中で対応する仮想バケットが前記データブロックのための空間を有するようになるまで前記複数の仮想バケットの中で対応する仮想バケットから前記仮想バケットの中で隣接する仮想バケットに順次データブロックを移動させる段階と、

前記複数の仮想バケットの中で対応する仮想バケットに前記データブロックを格納する段階と、を含むメモリの重複除去方法。

【請求項 2】

前記データブロックに対応する 1 つ以上の検索アドレス (Lookup address) を変更するためにアドレス検索テーブルメモリをアップデートする段階をさらに含むことを特徴とする請求項 1 に記載のメモリの重複除去方法。

【請求項 3】

前記複数のハッシュテーブルの各々は、参照カウントライン (Reference count line)、署名ライン (Signature line)、及びホップワードライン (Hopword line) をさらに含むことを特徴とする請求項 1 に記載のメモリの重複除去方法。

【請求項 4】

前記仮想バケットに対応するデータブロックを含む前記物理的なハッシュバケットを示すためのホップワードベクトル (Hopword vector) を生成する段階をさらに含むことを特徴とする請求項 3 に記載のメモリの重複除去方法。

【請求項 5】

前記ホップワードベクトルを生成する段階は、

前記複数の仮想バケットの各々に対し、前記複数の仮想バケットの中の各々の仮想バケットの前記複数の物理的なハッシュバケットの各々が前記複数の仮想バケットの中の各々の仮想バケットと連動されるデータブロックを含むか否かを示すために二進表示子 (Binary indicator) を使用する段階を含むことを特徴とする請求項 4 に記載のメモリの重複除去方法。

【請求項 6】

前記複数の物理的なハッシュバケットの中でデータブロックを含むいずれの物理的なハッシュバケットが前記複数の仮想バケットの中でいずれの仮想バケットに対応するかを示すために物理的なハッシュバケット毎に $\log_2(H)$ ビットで構成されるホップワード値を生成する段階をさらに含むことを特徴とする請求項 3 に記載のメモリの重複除去方法。

【請求項 7】

前記ホップワード値を生成する段階は、前記複数の物理的なハッシュバケット及び仮想バケットの対と関連して表現される位置にデータブロックを含む前記複数の物理的なハッシュバケットの各々のための準アドレスで構成される 2 次元アレイを生成する段階を含むことを特徴とする請求項 6 に記載のメモリの重複除去方法。

【請求項 8】

前記対応するハッシュテーブルは、揮発性メモリに格納されることを特徴とする請求項 1 に記載のメモリの重複除去方法。

【請求項 9】

前記揮発性メモリは、DRAM (Dynamic random access memory) を含むことを特徴とする請求項 8 に記載のメモリの重複除去方法。

【請求項 10】

前記対応するハッシュテーブルを $\log_2(H)$ ビットの仮想バケット利用値フィールド及び前記複数の仮想バケットの中の対応する仮想バケットのデータブロックの数と同一である値を含む物理的なライン ID (PLID) で索引 (Indexing) する段階をさらに含むことを特徴とする請求項 1 に記載のメモリの重複除去方法。

【請求項 11】

前記複数の仮想バケットの中で対応する仮想バケットに項目を書き込む場合、前記仮想バケット利用値フィールドを 1 増加させる段階をさらに含むことを特徴とする請求項 10

10

20

30

40

50

に記載のメモリの重複除去方法。

【請求項 1 2】

対応する前記ハッシュテーブルが一杯に満たされた場合、バッファメモリに前記データブロックを格納する段階をさらに含むことを特徴とする請求項 1 に記載のメモリの重複除去方法。

【請求項 1 3】

メモリでデータブロックの重複を減少させる重複除去メモリのための重複除去 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) メモリモジュールにおいて、

重複除去されたデータブロックのみを格納し、複数のハッシュテーブルの各々が、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットが、複数のハッシュウェイ (W a y s) を含み、データブロックを格納するように構成され、前記重複除去 D R A M メモリモジュールに格納されたハッシュテーブルの 3 次元アレイを含む、ハッシュテーブルメモリと、

前記重複除去されたデータブロックに対応するアドレスを格納するためのアドレス検索テーブルメモリと、

前記重複除去 D R A M メモリモジュールが前記ハッシュテーブルメモリから前記データブロックを検索し、前記データブロックを伝送するようにする読出し要求、または前記重複除去 D R A M メモリモジュールが前記ハッシュテーブルメモリに前記データブロックを格納するようにする書込み要求を受信するプロセッサと、を含むことを特徴とする重複除去 D R A M メモリモジュール。

【請求項 1 4】

チップ上で重複除去を実行し、重複除去 D R A M メモリモジュールの論理的な容量を増加させ、ホストプロセッサのリソースの使用を低減し、重複除去 D R A M メモリモジュールへのメモリアクセス数を低減することによりメモリを重複除去する重複除去 D R A M メモリモジュールであって、

前記重複除去 D R A M メモリモジュールは、

重複除去されたデータブロックのみを格納するためのハッシュテーブルメモリと、

前記重複除去されたデータブロックに対応するアドレスを格納するためのアドレス検索テーブルメモリと、

前記重複除去 D R A M メモリモジュールが前記ハッシュテーブルメモリから前記データブロックを検索し、前記データブロックを伝送するようにする読出し要求、または前記重複除去 D R A M メモリモジュールが前記ハッシュテーブルメモリの前記データブロックを格納するようにする書込み要求を受信するプロセッサと、を含み、

前記ハッシュテーブルメモリは、揮発性メモリに格納され、前記ハッシュテーブルメモリに格納される複数のハッシュテーブルの 3 次元アレイを含み、前記ハッシュテーブルの各々は、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットは、複数のハッシュウェイ (W a y s) を含み、前記重複除去されたデータブロックを格納するように構成されることを特徴とする重複除去 D R A M メモリモジュール。

【請求項 1 5】

前記複数のハッシュテーブルの各々は、複数の仮想バケットをさらに含み、各々の仮想バケットは、2 つ以上の前記物理的なハッシュバケットを含むことを特徴とする請求項 1 3 又は 1 4 に記載の重複除去 D R A M メモリモジュール。

【請求項 1 6】

前記複数のハッシュテーブルの中の対応するハッシュテーブル内で前記複数の仮想バケットの中の隣接する仮想バケットの間で前記重複除去されたデータブロックを移動させることを特徴とする請求項 1 5 に記載の重複除去 D R A M メモリモジュール。

【請求項 1 7】

外部から提供される命令無しでデータ重複除去を実行することを特徴とする請求項 1 3 又は 1 4 に記載の重複除去 D R A M メモリモジュール。

10

20

30

40

50

【請求項 18】

前記ハッシュテーブルメモリが一杯に満たされた場合、データブロックを格納するためのバッファメモリをさらに含むことを特徴とする請求項 13 又は 14 に記載の重複除去 D R A Mメモリモジュール。

【請求項 19】

重複除去 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)メモリモジュールにおいて、

複数のハッシュテーブルの各々が、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットが、複数のハッシュウェイ (W a y s) を含み、データブロックを格納するように構成され、前記重複除去 D R A Mメモリモジュールに格納されたハッシュテーブルの 3 次元アレイと、

プロセッサと、

メモリと、を含み、

前記メモリは、命令を格納し、前記命令が前記プロセッサによって実行される時、前記メモリは、前記重複除去 D R A Mメモリモジュールが前記複数のハッシュテーブルの中で対応するハッシュテーブル内の隣接する仮想バケットの間で事前に格納された重複除去されたデータブロックを移動させるようにし、前記仮想バケットの各々は、2 つ以上の物理的なハッシュバケットを含むことを特徴とする重複除去 D R A Mメモリモジュール。

【請求項 20】

前記メモリは、命令をさらに格納し、前記命令が前記プロセッサによって実行される時、前記メモリは、重複除去 D R A Mメモリモジュールが仮想バケットの中で前記事前に格納された重複除去されたデータブロックが移動された 1 つの仮想バケットにインカミングデータを格納するようにすることを特徴とする請求項 19 に記載の重複除去 D R A Mメモリモジュール。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はメモリの重複除去方法及び重複除去 D R A Mメモリモジュールに係り、さらに詳細には重複除去アプリケーションを直ちに処理する効率的なメモリのための最適化されたホップスコッチ動作の複数のハッシュテーブルを用いたメモリの重複除去方法及び重複除去 D R A Mメモリモジュールに関する。

【背景技術】

【0002】

データ重複除去はメモリ装置の容量費用を減らすために、メモリ装置で不必要なデータを減らすことを意味する。データ重複除去で、データ対象 / 項目 (例えば、データファイル) は 1 つ又はそれ以上のデータライン / チャンク / ブロックに分けられる。同一のデータで構成された複数のデータブロックを 1 つの格納されたデータブロックに関連させることによって、データブロックの重複コピーはコンピュータメモリによって減少、或いは除去され得る。そのようにすることによってメモリ装置のデータ重複ブロックの全体量は減少される。データの不必要なコピーの減少は読出し速度及びメモリ帯域幅を増加させ、潜在的に電力節減をもたらすことができる。

【0003】

したがって、仮に重複されたデータコピーを 1 つのデータコピーに減少させることができれば、物理的な資源の量は同様に使用しながら、メモリ装置の全体使用可能な容量は増加される。その結果としてメモリ装置の経済的な使用はデータ再書込み (R e w r i t e) 回数の減少を可能であるようにし、メモリに既に格納された重複されたデータブロックに対する書込み要求は破棄されるので、データ重複除去が適用されたメモリ装置の寿命は効果的に書込み耐久性を増加させることによって長期化することができる。

【0004】

一般的なデータ重複除去方法は、CPU中心のアプローチとして重複除去エンジンがCPU又はメモリコントローラ(Memory Controller; MC)に集積されるメモリ内重複除去(in-memory deduplication)技術を使用することができる。このような方法は重複をCPUプロセッサが認識できるように、そしてメモリコントローラの制御にしたがって重複除去メモリ動作(例えば、コンテンツ検索(Content lookups)、参照カウンタアップデート(Reference Count Updates)、等)の提供を試みるようにメモリコントローラと動作する重複除去キャッシュ(Deduplicated Cache; DDC)を具現する。重複除去の方法はまた、重要な経路から変換フェッチ(Fetch)を除去してデータ読出しを増加させるために変換ラインをキャッシング(Caching)するキャッシュであり、索引バッファ(Lookaside Buffer)と類似である、直接変換バッファ(Direct Translation Buffer; DTB)を具現することができる。

10

【0005】

重複除去は普通ハードドライブのために使用される。しかし、DRAM(Dynamic Random Access Memory)のような揮発性メモリの領域で微細な重複除去を提供するのに関心がある。

このような背景技術で前述した情報は単なる発明の理解を助けるためのものであり、したがって従来技術を構成しない情報を含むことができる。

【発明の概要】

20

【発明が解決しようとする課題】**【0006】**

本発明は上記従来メモリの重複除去における技術的課題を解決するためのものであって、本発明の目的は重複除去アプリケーションを直ちに処理する効率的なメモリのための仮想バケットを含む複数のハッシュテーブルを用いたメモリの重複除去方法及び重複除去DRAMメモリモジュールを提供することにある。

【課題を解決するための手段】**【0007】**

上記目的を達成するためになされた本発明の実施形態の側面はDRAM(Dynamic random access memory)システムでメモリ重複除去に係る。

30

本発明の実施形態によるメモリの重複除去方法が提供され、前記メモリの重複除去方法は、複数のハッシュテーブル(Hash table)の各々が、ハッシュ関数に対応し、前記複数のハッシュテーブルの各々が、複数の物理的なハッシュバケット(Hash bucket)を含み、各々の物理的なバケットは、複数のハッシュウェイ(Ways)を含み、データブロックを格納する、前記複数のハッシュテーブルを識別する段階と、複数の仮想バケット(Virtual bucket)の各々が、前記複数の物理的なハッシュバケットの中の一部を含み、前記複数の仮想バケットの中で他の仮想バケットと少なくとも1つの前記物理的なハッシュバケットを共有する、前記複数の仮想バケットを識別する段階と、前記複数の仮想バケットの中で対応する1つの仮想バケットに割当てられて格納されたデータブロックを有する前記複数の物理的なハッシュバケットの各々を識別する段階と、ハッシュ値(Hash value)を生成するためにハッシュ関数の中で対応するハッシュ関数にしたがってデータライン(Data line)をハッシング(Hashing)する段階と、対応するハッシュテーブルの前記仮想バケットの中で対応する仮想バケットが前記ハッシュ値にしたがってデータブロックのための使用可能な空間を有するか否かを決定する段階と、前記仮想バケットの中で対応する仮想バケットが使用可能な空間を有しない場合、前記仮想バケットの中で対応する仮想バケットが前記データブロックのための空間を有するようになるまで前記複数の仮想バケットの中で対応する仮想バケットから前記仮想バケットの中で隣接する仮想バケットに順次データブロックを移動させる段階と、前記複数の仮想バケットの中で対応する仮想バケットに前記データブロックを格納する段階と、を含むことを特徴とする。

40

50

【0008】

メモリの重複除去方法は、前記移動されたデータブロックに対応する1つ以上の検索アドレス(Lookup addresses)を変更するためにアドレス検索テーブルメモリをアップデートする段階をさらに含むことができる。

前記複数のハッシュテーブルの各々は、参照カウンライン(Reference count line)、署名ライン(Signature line)、及びホップワードライン(Hopword line)をさらに含むことができる。

【0009】

メモリの重複除去方法は前記仮想バケットに対応するデータブロックを含む前記物理的なハッシュバケットを示すためのホップワードベクトル(Hopword vector)を生成する段階をさらに含むことができる。

前記ホップワードベクトルを生成する段階は、前記複数の仮想バケットの各々に対し、前記複数の仮想バケットの中の各々の仮想バケットの前記複数の物理的なハッシュバケットの各々が前記複数の仮想バケットの中で前記各々の仮想バケットと連動されるデータブロックを含むか否かを示すために二進表示子(Binary indicator)を使用する段階を含むことができる。

【0010】

メモリの重複除去方法は前記複数の物理的なハッシュバケットの中でデータブロックを含むいずれの物理的なハッシュバケットが前記複数の仮想バケットの中でいずれの仮想バケットに対応するかを示すために物理的なハッシュバケット毎に $\log_2(H)$ ビットで構成されるホップワード値を生成する段階をさらに含むことができる。

前記ホップワード値を生成する段階は、前記複数の物理的なハッシュバケット及び仮想バケットの対と関連して表現される位置にデータブロックを含む前記複数の物理的なハッシュバケットの各々のための準アドレスで構成される2次元アレイを生成する段階を含むことができる。

【0011】

前記複数のハッシュテーブルは、揮発性メモリに格納されることができる。

前記揮発性メモリは、DRAM(Dynamic random access memory)を含むことができる。

【0012】

メモリの重複除去方法は、前記ハッシュテーブルを $\log_2(H)$ ビットの仮想バケット利用値フィールド及び前記複数の仮想バケットの中の対応する仮想バケットのデータブロックの数と同一である値を含む物理的なラインID(PLID)で索引(Indexing)する段階をさらに含むことが好ましい。

メモリの重複除去方法は、前記複数の仮想バケットの中で対応する仮想バケットに項目を書き込む場合、前記仮想バケット利用値フィールドを1増加させる段階をさらに含むことができる。

メモリの重複除去方法は、対応するハッシュテーブルが一杯に満たされた場合、バッファメモリに前記データブロックを格納する段階をさらに含むことができる。

【0013】

本発明の実施形態による重複除去メモリのためにメモリでデータブロックの重複を減少させる重複除去DRAM(Dynamic random access memory)メモリモジュールが提供され、重複除去DRAMメモリモジュールは、重複除去されたデータブロックのみを格納し、複数のハッシュテーブルの各々が、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットが、複数のハッシュウェイ(Ways)を含み、データブロックを格納するように構成され、前記重複除去DRAMメモリモジュールに格納されたハッシュテーブルの3次元アレイを含む、ハッシュテーブルメモリと、前記重複除去されたデータブロックに対応するアドレスを格納するためのアドレス検索テーブルメモリと、前記重複除去DRAMメモリモジュールが前記ハッシュテーブルメモリから前記データブロックを検索し、前記データブロックを伝送するようにする読出し

10

20

30

40

50

要求、または前記重複除去DRAMメモリモジュールが前記ハッシュテーブルメモリに前記データブロックを格納するようにする書込み要求を受信するプロセッサと、を含むことを特徴とする。

また、本発明の実施形態による重複除去DRAMメモリモジュールは、チップ上で重複除去を実行し、重複除去DRAMメモリモジュールの論理的な容量を増加させ、ホストプロセッサのリソースの使用を低減し、重複除去DRAMメモリモジュールへのメモリアクセス数を低減することによりメモリを重複除去する重複除去DRAMメモリモジュールであって、前記重複除去DRAMメモリモジュールは、重複除去されたデータブロックのみを格納するためのハッシュテーブルメモリと、前記重複除去されたデータブロックに対応するアドレスを格納するためのアドレス検索テーブルメモリと、前記重複除去DRAMメモリモジュールが前記ハッシュテーブルメモリから前記データブロックを検索し、前記データブロックを伝送するようにする読出し要求、または前記重複除去DRAMメモリモジュールが前記ハッシュテーブルメモリの前記データブロックを格納するようにする書込み要求を受信するプロセッサと、を含み、前記ハッシュテーブルメモリは、揮発性メモリに格納され、前記ハッシュテーブルメモリに格納される複数のハッシュテーブルの3次元アレイを含み、前記ハッシュテーブルの各々は、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットは、複数のハッシュウェイ(Ways)を含み、前記重複除去されたデータブロックを格納するように構成されることを特徴とする。

10

【0014】

前記ハッシュテーブルメモリは、前記ハッシュテーブルメモリに格納される複数のハッシュテーブルの3次元アレイを含み、前記ハッシュテーブルの各々は、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットは、複数のハッシュウェイ(Ways)を含み、前記重複除去されたデータブロックを格納することができる。

20

前記ハッシュテーブルの各々は、複数の仮想バケットをさらに含み、各々の仮想バケットは、2つ以上の前記物理的なハッシュバケットを含むことができる。

【0015】

重複除去DRAMメモリモジュールは、前記ハッシュテーブルの中の対応するハッシュテーブル内で前記複数の仮想バケットの中の隣接する仮想バケットの間で前記重複除去されたデータブロックを移動させることができる。

重複除去DRAMメモリモジュールは外部から提供される命令無しでデータ重複除去を実行することができる。

30

重複除去DRAMメモリモジュールは、前記ハッシュテーブルメモリが一杯に満たされた場合、データブロックを格納するためのバッファメモリをさらに含むことができる。

【0016】

本発明の実施形態による重複除去メモリのためにメモリでデータブロックの重複を減少させるための重複除去DRAMメモリモジュールが提供され、重複除去DRAMメモリモジュールにおいて、複数のハッシュテーブルの各々は、複数の物理的なハッシュバケットを含み、各々の物理的なハッシュバケットは、複数のハッシュウェイ(Ways)を含み、前記データブロックを格納し、前記重複除去DRAMメモリモジュールに格納されるハッシュテーブルの3次元アレイと、プロセッサと、メモリと、を含み、前記メモリは、命令を格納し、前記命令が前記プロセッサによって実行される時、前記メモリは、前記重複除去DRAMメモリモジュールが前記複数のハッシュテーブルの中で対応するハッシュテーブル内の隣接する仮想バケットの間で事前に格納された重複除去されたデータブロックを移動させるようにし、前記仮想バケットの各々は、2つ以上の物理的なハッシュバケットを含むことを特徴とする。

40

【0017】

前記メモリは、命令をさらに格納し、前記命令が前記プロセッサによって実行される時、前記メモリは、重複除去DRAMメモリモジュールが仮想バケットの中で前記事前に格納された重複除去されたデータブロックが移動された1つの仮想バケットにインカミングデータを格納するようにすることができる。

50

【発明の効果】

【0018】

本発明の実施形態による重複除去DRAMメモリモジュールによれば、メモリアクセスは減少することができ、DRAMシステムの寿命を延長することができる。

【図面の簡単な説明】

【0019】

【図1】本発明の実施形態による重複除去DRAMシステムアーキテクチャのブロック図である。

【図2】図1の実施形態の重複除去メモリモジュール内メモリ形態のブロック図である。

【図3】図2の実施形態のハッシュテーブルメモリのハッシュテーブルのブロック図である。

10

【図4】本発明の実施形態による複数のハッシュテーブルアレイのブロック図である。

【図5】(A)～(C)はそれぞれ本発明の実施形態による仮想ケットと特定の物理的なケットを関連させるホップワード(Hop words)を生成するための2次元のアレイを示す図である。

【図6】本発明の実施形態によるハッシュテーブルメモリのデータブロックのアドレッシングのための物理的なラインID(PLID)のブロック図である。

【図7】本発明の実施形態による、ホップスコッチ方法を使用するメモリモジュールの複数のハッシュテーブルアレイにデータを書き込む過程を説明するためのフローチャートである。

20

【図8】本発明の実施形態による、メモリモジュールの複数のハッシュテーブルアレイからデータを読み出す過程を説明するためのフローチャートである。

【発明を実施するための形態】

【0020】

本発明の特徴、及びそれを達成する方法は実施形態の詳細な説明及び添付された図面を参照すれば、明確になる。以下、例示的な実施形態を、類似な参照番号は類似な構成要素を指称する添付図面を参照して詳細に説明する。しかし、本発明は様々な多様な形態に具現することができ、本明細書で単に例示した実施形態に限定されるものではない。むしろ、このような実施形態はこの開示が徹底のためであり、完全にするための例として提供され、当業者に本発明の特徴及び機能を完全に伝達するものである。したがって、本発明の技術分野で通常の知識を有する者が本発明の特徴及び機能を完璧に理解するために必要としないプロセス、要素、及び技術は説明しない。特別に言及しない限り、類似な参照番号は添付した図面及び記載した説明で類似な構成要素を示し、したがって、それに対する説明は反復しない。図面で、構成要素、層、及び領域の相対的な大きさは明確性のために誇張することがある。

30

【0021】

たとえば、ここで第1、第2、第3等の用語は多様な要素、成分、領域、層、及び/又はセクションを説明するために使用するが、このような要素、成分、領域、層、及び/又はセクションはこのような用語によって制限されないことと理解されるべきである。このような用語は他の要素、成分、領域、層、又はセクションから1つの要素、構成、領域、層又はセクションを区別するために使用する。したがって、後述する第1構成要素、成分、領域、層、又はセクションは本発明の思想及び範囲を逸脱することなく、第2構成要素、成分、領域、層、又はセクションを指称することができる。

40

【0022】

1つの要素又は図面で示した他の構成要素又は特徴との特徴的な関係を説明するための説明を容易にするのに“下の”、“下”、“低い”、“特定部分の下”、“上に”、“上部”と同一の空間的であり、相対的な用語をここで使用することがある。空間的であり、相対的な用語は図面で示した方向に加えて使用又は動作で装置の他の方向を含むように意図することが理解されるべきである。例えば、仮に図面の装置を裏返したら、他の構成要素又は特徴の“下”又は“下の”又は“特定部分の下”として説明した構成要素は他の構

50

成要素又は特徴の“上に”合わせられるようになる。したがって、“下の”又は“特定部分の下”の例示的な用語は上又は下方向の全てを含むことができる。装置は異なるように合わせられ（例えば、90°は他の方向に回転されること）、そして空間的に相対的な技術語はそれにしたがって解釈されなければならない。

【0023】

要素、層、領域、又は成分が他の要素、層、領域又は成分“に”、“に結合された”、“に連結された”と言及する時、それは他の要素、層、領域、又は成分“に直接的に”、“に直接的に結合された”、“に直接的に連結された”ものであるか、或いは1つ又はそれ以上の間の要素、層、領域、又は成分が存在することができることを意味する。また、要素又は層が2つの要素又は層の間と言及する時、それは単なる要素又は層が2つの要素又は層間にあるか、又は1つ又はそれ以上の間の要素又は層がまた存在し得ることを意味する。

10

【0024】

次の例で、x軸、y軸、及びz軸は直角座標システムの3つの軸に限定されることなく、広い意味に解釈することができる。例えば、x軸、y軸、及びz軸は互いに直交することができる、又は互いに直交しない他の方向を示すことができる。

本明細書で使用する用語は単なる特定のな実施形態を説明するためのものであり、本発明を制限しようとする意図したものではない。本明細書で使ったように、文脈の上に明確に異なるように意味しない限り、単数形態の“1つ”は複数の形態も含むことを意図する。“構成される”、“構成されている”、“含む”、及び“含んでいる”の用語を本明細書で使用するとき、このような用語は定められた特徴、整数、段階、動作、要素、及び/又は成分が存在することを明示するが、1つ又はそれ以上の他の特徴、整数、段階、動作、要素、成分、及び/又はそれらのグループの追加又は存在を不可能にすることではない。本明細書で使用するように、“及び/又は”の用語は1つ又はそれ以上の列挙した項目と関連する任意の、そしてすべての組合せを含む。“少なくとも1つ”のような表現は要素の全体リストを修正し、そしてリストの個別要素を修正しない。

20

【0025】

本明細書で使用するように、“大体に”、“約”の用語及びこれと類似な用語は近似値の用語として使用し、程度の用語として使用するのではなく、本発明の当業者によって識別される測定された又は計算された値の固有な変動を考慮するためである。また、本発明の実施形態を記述する時“することができる”の使用は“本発明の1つ以上の実施形態”を意味する。本明細書で使用するように、“使用”、“使用される”、そして“使用された”の用語は“利用”、“利用される”、そして“利用された”の用語の同義語として各々看做することができる。また、“例示”の用語は例又は図面を意味する。

30

【0026】

特定の実施形態は異なるように具現される場合、特定プロセス順序は説明した順序と異なるように遂行することができる。例えば、説明した連続的な2つのプロセッサは同時に概ね実行されるか、或いは説明した順序と反対の順序に実行されてもよい。

【0027】

本明細書で記述する本発明の実施形態による電子又は電気装置及び/又は他の任意の関連した装置又は要素は任意の適合なハードウェア、ファームウェア（例えば、Application Specific Integrated Circuit; ASIC）、ソフトウェア、又はソフトウェア、ファームウェア、及びハードウェアの組合を利用して具現することができる。例えば、このような装置の多様な要素は1つの集積回路（Integrated Circuit; IC）チップ又は分離されたICチップで形成することができる。また、このような装置の多様な要素は可撓性の印刷回路フィルム（Flexible Printed Circuit Film）、TCP（Tape Carrier Package）、印刷回路基板（Printed Circuit Board; PCB）上に具現するか、或いは1つの基板上で形成することができる。また、このような装置の多様な要素はコンピュータプログラム命令を実行し、本明細書で説明する

40

50

多様な機能を遂行するための他のシステム要素と相互作用する1つ以上のコンピューティング装置で、又は1つ以上のプロセッサで遂行されるプロセス又はスレッド(Thread)である。

【0028】

コンピュータプログラム命令は、例えばRAM(Random Access Memory)のような標準メモリ装置を利用するコンピューティング装置で具現されるメモリ内に格納される。コンピュータプログラム命令はまた、例えばCD-ROM、フラッシュドライブ(Flash Drive)、又はそのような他の一時的ではないコンピュータ読み出し可能なメディア(Non-transitory Computer Readable Media)に格納されることもあり得る。また、本発明の当業者は本発明の例示的な実施形態の思想及び範囲を逸脱することなく、多様なコンピューティング装置の機能は単一コンピューティング装置に統合されるか、或いは集積され、特定コンピューティング装置の機能が1つ又はそれ以上の他のコンピューティング装置に分散されることを認識しなければならない。

10

【0029】

異なるように定義されない限り、本明細書で使用するすべての用語(技術的そして科学的用語を含む)は本発明が属する技術分野で当業者によって一般的に理解される同一の意味を有する。一般的に使用される事前に定義されたこのような用語は本明細書及び/又は関連技術の文脈でそれらの意味と一致する意味を有することと解釈されるべきであり、本明細書で明確に定義されない限り、理想化されるか、或いは過度に形式的なこととして解釈されてはならない。

20

【0030】

図1は本発明の実施形態による重複除去DRAMシステムアーキテクチャのブロック図である。

図1を参照すると、コンピュータメモリとして機能するために、重複除去メモリは原本データのコンテンツ及び重複除去された固有のメモリブロックのセット(Set)の間の関係を記録するための“変換(Translation)”として公知された機能を遂行し、記録された関係は圧縮された形態に記録される。例えば、原本データのアドレスは検索テーブル(Lookup Table)に格納される。

【0031】

一般的に、CPUのプロセッサ(Processor)110は物理的なメモリ(例えば、重複除去DRAMメモリモジュール130)に直接的なアクセスが難しく、上述したアクセスはメモリライン(Memory Lines)のアレイ(Array)にメモリコントローラ120によって代わりに管理される。CPU中心の重複除去システムはデータがメモリシステムに到達する前にCPU内部のキャッシュデータを求める。

30

【0032】

本発明の実施形態による重複除去DRAMシステムアーキテクチャ(Deduplication DRAM System Architecture)100は従来のCPU中心の重複除去ではないメモリ中心の重複除去を使用し、これは重複除去DRAMメモリモジュール130がプロセッサ110からの命令無しでもメモリ重複除去を遂行することを意味する。重複除去DRAMシステムアーキテクチャ100はまたメモリの容量利得を増加させ、このようにすることによって高容量メモリソリューションを提供するために、重複除去DRAMメモリモジュール130に格納された設定可能な重複除去アルゴリズムを使用する。即ち、CPU中心の重複除去と異なり、本発明の実施形態による重複除去DRAMシステムアーキテクチャ100は、RAMモジュール(例えば、重複除去DRAMメモリモジュール130)内に含まれたすべての重複除去情報(Intelligence)を有する。したがって、重複除去はCPUモジュール140が認識せずに重複除去DRAMメモリモジュール130内で遂行されることが可能であり、これにより重複除去DRAMメモリモジュール130の容量が増加されるようになる。即ち、重複除去は微細粒(fine grain)であり、揮発性メモリ(例えば、重複除去DRAMメモリモ

40

50

ジュール130)内で動作するので、本発明の実施形態によるすべての重複除去情報は重複除去DRAMメモリモジュール130の自体内で発生し、一方CPU内のカーネルモジュール(Kernel Module)140は重複除去DRAMメモリモジュール130内で遂行される重複除去動作の細部事項を認識しないことがありうる。

【0033】

本発明の実施形態は重複除去DRAMメモリモジュール130としてDRAMを使用して説明したが、他の種類のメモリが本発明の他の実施形態に使用され得ると理解されなければならない。また、本発明の実施形態による重複除去DRAMシステムアーキテクチャ100は多様な種類のメモリとインターフェイシング(Interfacing)を支援することが可能である。即ち、本発明の実施形態による重複除去DRAMメモリモジュール130は、メモリコントローラ120を通じて多様な他の種類のメモリインターフェイスと関連され得る(例えば、DDR4(Double Data Rate Fourth-Generation Synchronous Dynamic Random-Access Memory)、コンピュータと1つ以上の周辺装置とを接続するための直列拡張バス標準であるPCIe(Peripheral Component Interconnect Express)、DDR-T、及びKTI)。したがって、重複除去DRAMシステムアーキテクチャ100に重複除去DRAMメモリモジュール130を集積するために他のアーキテクチャが使用され得ることに注目しなければならない。

10

【0034】

また、本発明の実施形態を具現するために、既存DRAMメモリモジュールに若干の変化(例えば、ドライバーアップグレード(Driver Upgrade))があるが、ソフトウェアの具現はオペレーティングシステム(Operating System; OS)/CPUモジュール140又はプロセッサ110の物理的な変化無しで本発明の実施形態による重複除去DRAMシステムアーキテクチャ100を使用するようにすることができる。

20

【0035】

本発明の実施形態による重複除去DRAMシステムアーキテクチャ100は重複除去、コンテンツアドレス指定能力(Content Addressability)、保安(Security)、メモリ内処理(Processor-in-memory; PIM)、及び関連したアドレスが行アドレスであり、それによってDRAM内データビットが列アドレス及び行アドレス等の交差点に位置されるセルに格納されることをDRAMに通知するためにDRAMに伝送された信号であるRAS(Row Address Strobe)のようなDRAMの知能的なプロトコルのために重複除去DRAMメモリモジュール130にSoC(System on Chip)を実装することができる。

30

【0036】

重複除去DRAMシステムアーキテクチャ100はまたプロセッサ110がメモリコントローラ120と関連した仮想密度管理(Virtual Density Management)、スマートデータ配置(Smart Data Placement)、及びDRAM情報APIs(Application Programming Interfaces)、等を可能にすることを引き起こすスマートシステムソフトウェア(Smart System Software)を有することができる。

40

【0037】

重複除去DRAMメモリモジュール130は多様なフォームファクタ(form factors、例えばDIMM(Dual In-line Memory Module)、2.5In、FHHL(Full Height Half Length)、HHHL(Half Height Half Length)、FHFL(Full Height Full Length)、等)の高容量DRAMメモリモジュールのような3DS DRAM構成要素をさらに有することができる。

【0038】

したがって、本発明の実施形態による重複除去DRAMシステムアーキテクチャ100

50

を使用するメモリ中心の重複除去システムを提供することによって、重複除去書込みプロセス (Duplicate Write Process) はメモリインターフェイスで直接的に遂行され、これにより重複除去DRAMメモリモジュール130の容量が増加する。

【0039】

図2は図1の実施形態の重複除去DRAMメモリモジュール内メモリ形態のブロック図であり、図3は図2の実施形態のハッシュテーブルメモリのハッシュテーブルのブロック図である。

【0040】

図2を参照すると、本発明の実施形態による重複除去DRAMメモリモジュール130は重複除去アルゴリズムアーキテクチャを有し、重複除去メモリDRAMモジュール130の内部のメモリ領域は3つの異なる領域に分類される。3つの異なる領域は重複除去されたデータブロック (Blocks of data) が格納される位置を示すためのアドレス検索テーブルメモリ (Address Lookup Table Memory; 以下、ALUTM) 210、重複除去されたデータブロックを格納するためのハッシュテーブルメモリ (Hash Table Memory) 220、及びハッシュテーブルメモリのハッシュテーブルのハッシュウェイ (Hash ways) がいっぱい満たされた時にデータを格納するための超過/バッファメモリ (Overflow/Buffer Memory) 230を含む。

【0041】

データブロックが重複除去DRAMメモリモジュール130に入力される時、重複除去アルゴリズムはデータブロックが新しいものであるか、以前に格納されないALUTM 210内の任意の対応するアドレスが無いデータブロックであるかを決定するために動作する。このような動作を遂行するために、重複除去アルゴリズムはALUTM 210にアクセスする。同一のデータブロックは単一の項目で格納されることを保障するために、ALUTM 210内ポインター (例えば、下の図5で説明する物理的なラインID (Physical Line ID; PLID)) はハッシュテーブルメモリ220に同一のデータブロック格納位置を示す。即ち、ALUTM 210はハッシュテーブル内で検索アドレスマッピングポインター (Lookup Address Mapping Pointer、例えばPLID) と関連する位置 (例えば、アドレス) のための格納装置である。したがって、データブロックがハッシュテーブルメモリ220に以前に格納されたら、ALUTM 210内ポインターは同一のデータブロックが格納されたハッシュテーブルメモリ220のアドレスを示すことが可能であり、このようにすることによってデータブロックの重複コピーを格納する必要を除去し、重複除去DRAMメモリモジュール130の容量は増加する。

【0042】

図3を参照すると、メモリ重複除去は高い水準の重複除去、そしてそれに対応する、重複除去DRAMメモリモジュール130の大きいメモリ容量を保障するために、相対的に効率的であるが、簡単な多重方式のハッシュテーブル/ハッシュアレイ (Multiple-way Hash Table/Hash Array、380) を使用することができる。本発明の実施形態による重複除去DRAMメモリモジュール130のハッシュテーブルメモリ220は1つ以上のハッシュテーブル380があり、データブロックが唯一であるか否かを決定するのに、その有用性のために使用される。ハッシュテーブル380はハッシュバケット (Hash Buckets、行) 310及びハッシュウェイ (Hash Ways、列) 320で構成される2次元アレイとして考慮することができる。即ち、本発明の実施形態によるハッシュテーブル380はm個のハッシュバケット310を行に含み、各ハッシュバケット310はハッシュバケット310の容量を示すデータライン/スロット/項目/ハッシュウェイ320のn個の列を含む (m及びnは整数)。

【0043】

ハッシュテーブルメモリ220のハッシュウェイ320にデータブロックが格納され、

A L U T M 2 1 0 のアドレスポインターは特定のデータブロックと関連した特定のハッシュバケット 3 1 0 及び特定のハッシュウェイ 3 2 0 を示す値を格納することができる。したがって、アドレス（例えば、64 - ビットアドレス）は A L U T M 2 1 0 に索引され、それから、アドレスに対応するデータブロックを格納するハッシュテーブル 3 8 0 のハッシュバケット 3 1 0 の関連したハッシュウェイ 3 2 0 が決定され得る。

【 0 0 4 4 】

したがって、書込みプロセス（例えば、64 - バイト（64 - byte）のデータ書込み）の間に、書込み要求（即ち、1つ以上のデータブロックで構成されるインカミング（Inc o m i n g）データを記録するための要求）を受信した後、ハッシュ値が対応するハッシュバケット 3 1 0 及びハッシュウェイ 3 2 0 を決定するためにハッシュ関数 / ハッシュアルゴリズム（即ち、インカミングデータが“ハッシュされる”）を利用してインカミングデータに対して計算される。したがって、ハッシュ値はデータブロックが何処に配置されたかを示すか、又は、データブロック（例えば、64 バイトのデータブロック）が重複である場合、ハッシュ値はデータブロックがハッシュテーブルメモリ 2 2 0 に既に格納された位置を示す。メモリにデータコンテンツが追加されることによって、m ハッシュバケット 3 1 0 の中で一部は先ず飽和状態に到達することがある。したがって、重複除去 D R A M メモリモジュール 1 3 0 はハッシュテーブルメモリ 2 2 0 に入らないデータブロックを格納するためのバッファメモリ 2 3 0 を使用する超過提供（O v e r f l o w P r o v i s i o n）を含む。その後、原本検索アドレス（O r i g i n a l L o o k u p A d d r e s s）は検索され、A L U T M 2 1 0 はインカミングデータのハッシング（H a s h i n g）から計算された検索アドレスにしたがってアップデートされる。

【 0 0 4 5 】

試みられた書込みプロセスの間に、ハッシュウェイ 3 2 0 の全てが一杯に満たされたことと判断される時、バッファメモリ 2 3 0 が使用される。即ち、ハッシュテーブル 3 8 0 が一杯に満たされれば、データはバッファメモリ 2 3 0 の重複除去されない超過領域（n o n - d e d u p l i c a t e d o v e r f l o w r e g i o n）に配置され、これにより重複除去水準が減少される。したがって、バッファメモリ 2 3 0 は根本的に予約された、標準の、簡単な超過メモリ領域であり、仮想密度過剰提供管理超過（v i r t u a l d e n s i t y o v e r - p r o v i s i o n m a n a g e m e n t o v e r f l o w）を具現するための S O C メモリバッファ / キャッシュに提供される。データがバッファメモリ 2 3 0 に一旦配置されれば、それ以上ハッシュされなく、そしてそれ以上重複除去されることができない。

【 0 0 4 6 】

コンピュータアプリケーションが数回に亘ってメモリに同一のシーケンスの値を格納するように試みれば、A L U T M 2 1 0 に格納された変換アレイの多重項目はハッシュテーブルメモリ 2 2 0 に格納されたデータブロックの同一のアドレスを参照し、ここで、A L U T M 2 1 0 の項目は原本の固有なデータブロックより小さく、これにより効率的な圧縮が達成され得る。

【 0 0 4 7 】

m 個のハッシュバケット 3 1 0 の各々はハッシュバケット 3 1 0 の対応するハッシュウェイ 3 2 0 を示すための固有の識別子を含む参照 / 頻度カウンタライン 3 4 0、及び署名ライン（S i g n a t u r e L i n e）3 3 0 をさらに含む。各ハッシュバケット 3 1 0 のための、対応する署名ライン 3 3 0 は空いているライン（F r e e L i n e）を示すための 0、又はコンテンツ検索最適化のための 0 ではない 2 次的なハッシュ値の中でいずれか 1 つを含む。したがって、コンテンツ検索のために、一般的に署名ラインの 0 項目に基づいて空いているラインが割当されるように要請する署名一致がないか、或いは、後続のデータライン読出し及びコンテンツ比較が重複の存在を確認するようにする 1 つの署名一致が存在する。m 個のハッシュバケット 3 1 0 の各々は下の図 5（A）～（C）で説明するホップワードライン（H o p w o r d L i n e）をさらに含むことができる。

【 0 0 4 8 】

10

20

30

40

50

物理的なラインID (PLID) 350はデータをハッシュテーブル380に索引するために使用される。PLID 350はALUTM 210、ハッシュテーブルメモリ220、又はバッファメモリ230の中でいずれか1つに区別されるメモリラインを識別するために使用される。各メモリラインはハッシュテーブル380に固有のコンテンツを格納するためのデータライン、又は各々のPLID 350を格納し、プロセッサバスアドレス (Processor Bus Address) からハッシュテーブル380の重複除去されたデータブロックへのマッピングを提供するための変換ライン (Translation Line) の中でいずれか1つに関連される。即ち、バスアドレスは変換ラインを識別し、順に特定データラインを指定する関連あるPLID 350が含まれた変換ライン内項目をさらに識別する。したがって、PLID 350は超過タグ (Overflow Tag) を含むように具現され、特定の対応するハッシュテーブル380、対応するハッシュバケットビット、及びPLID 350に対応するデータブロックの位置を示す対応するウェイビット (Way Bits) を示すためのデータを含む。

10

【0049】

各ハッシュバケット310には、ハッシュバケット310のデータを索引するために使用される $\log_2(m)$ -ビットハッシュを生成するアルゴリズムである関連したハッシュ関数/ハッシュアルゴリズム“ $h(x)$ ”が存在する(例えば、ハッシュテーブル380が8個の物理的なハッシュバケット310を有すれば、ハッシュテーブル380のハッシュ関数は3ビットハッシュを生成する)。即ち、ハッシュ関数 $h(x)$ は相対的に大きい入力データ量(例えば、メモリに格納される入力データファイル)をハッシュ関数 $h(x)$ に入力することを可能とし、出力データ(例えば、ハッシュ値)の相当に異なる小さい量がハッシュテーブル380に格納されるようにするためにハッシュ関数 $h(x)$ によって生成され、出力される。したがって、ハッシュ関数 $h(x)$ は圧縮を可能にし、互いに異なるデータセットは時々同じハッシュ値にハッシュされることもある。

20

【0050】

重複除去されたメモリへの書込みにおいて、データファイルに対応する書込み要求を受信した後、重複除去されたメモリは最初に同一/重複データブロックが既にハッシュテーブル380に格納されているか否かを決定するために重複検索を実行する。それから、重複除去されたメモリはALUTM 210及びハッシュテーブルメモリ220の項目をアップデートする。例えば、参照/頻度カウントライン340はハッシュテーブルメモリ220内の原本検索アドレス(即ち、1ずつ減少される)の頻度カウントをアップデートし、参照カウントが0に到達すると、対応するデータブロックは削除される。それだけでなく、新しいPLID 350がALUTM 210で生成される。

30

【0051】

コンテンツ検索と称される重複検索の間に、重複除去DRAMメモリモジュール130は書き込もうとするデータファイル又はその一部の既存のインスタンス (Pre-existing instances) を探す。ハッシュテーブルメモリ220に格納されたデータの既存のインスタンスがある場合、重複検索は対応するデータラインを示すPLID 350を返す。データの既存のインスタンスが見つからない場合は、対応するデータブロックのための新しいデータラインが、ハッシュテーブル380の空間を割当てし、そこにコンテンツを書き込み、そして新しいPLID 350を返すことによって生成される。コンテンツはバスアドレスによって決定されたオフセットでALUTM 210にPLID 350を格納することによって記録される。

40

【0052】

ハッシュテーブル380にデータライン“C”を挿入するために、Cに対応するハッシュ関数“ $h(C)$ ”は数学的演算で計算される。データラインCのためにハッシュ関数が計算されると、ハッシュテーブル $T(h(C))$ の行はデータラインCの挿入を許容するために使用可能な充分な空間があるか否かを見るための(又はハッシュテーブル380に重複されるデータラインCが既にあるか否かを見るための)コンテンツ検索動作によってチェックされる。

50

【 0 0 5 3 】

前述したように、ハッシュテーブル 3 8 0 の各ハッシュバケット 3 1 0 は、追加的に署名ライン 3 3 0 及び参照 / 頻度カウントライン 3 4 0 をさらに含み、署名ライン 3 3 0 の署名 3 3 2 及び参照 / 頻度カウントライン 3 4 0 の参照カウント 3 4 2 が様々な数量を各ハッシュバケット 3 1 0 に満たすのに十分に小さく設計されるという事実によって、署名ライン 3 3 0 及び参照 / 頻度カウントライン 3 4 0 の各々は、単一のハッシュウェイ 3 2 0 を占有することができる。即ち、ハッシュテーブル 3 8 0 で、ハッシュテーブル 3 8 0 の 1 つの全体列は各々ハッシュバケット 3 1 0 に属する署名ライン 3 3 0 に割当てられ、1 つの全体列は各々ハッシュバケット 3 1 0 に属する参照 / 頻度カウントライン 3 4 0 に割当てられる。

10

【 0 0 5 4 】

データライン “ C ” のような実際データブロックがハッシュテーブル 3 8 0 に加えられるとき、ハッシュテーブル 3 8 0 には A L U T M 2 1 0 に格納された対応する P L I D 3 5 0 を各個別的な重複除去されたデータラインのハッシュテーブル 3 8 0 内アドレスにマッチングすることによって後にアクセスされるデータが満たされて始まる。ハッシュテーブル 3 8 0 内アドレスは、データが位置する特定のハッシュバケット 3 1 0 及び特定のハッシュウェイ 3 2 0 を識別すること（例えば、ハッシュテーブル 3 8 0 の行及び列を識別すること）によって識別される。したがって、ハッシュテーブル 3 8 0 に格納される各データブロックのために、データブロックの位置を示す A L U T M 2 1 0 に格納される対応する P L I D 3 5 0 によって識別される 1 つ以上の対応するアドレスがある。ハッシュテーブル 3 8 0 がデータで一杯に満たされれば、新しく導入されるデータは重複除去されない超過領域 / バッファメモリ 2 3 0 に配置され、これにより重複除去水準が減少される。

20

【 0 0 5 5 】

重複除去されたメモリからの読み出しで、重複除去されたメモリはハッシュテーブルメモリ 2 2 0 からのデータライン又はバッファメモリ 2 3 0 からの超過ラインの中でいずれか 1 つのコピーを返す。例えば、格納されたデータが読み出される時、読み出し要求を受信した後、ハッシュテーブル 3 8 0 の対応するアドレスが A L U T M 2 1 0 に格納された P L I D 3 5 0 を利用して検索される。そして、各アドレスに対応するブロックが検索され、再構成される。

【 0 0 5 6 】

図 4 は本発明の実施形態による複数のハッシュテーブルアレイのブロック図である。
図 4 を参照すると、本発明の実施形態による重複除去 D R A M システムアーキテクチャは複数のハッシュテーブル (M u l t i p l e H a s h T a b l e s ; M H T) 4 8 0 で構成されるハッシュテーブルアレイ 4 0 0 を使用し、各々の複数のハッシュテーブル 4 8 0 は m 個の物理的なバケット 4 1 0 を含み、各ハッシュバケット 4 1 0 は n 個のハッシュウェイ 4 2 0 を含む。ここで、本発明の実施形態はハッシュテーブル 4 8 0 及びハッシュバケット 4 1 0 をそれらの大きさが一定であると説明したが（例えば、m 及び n の整数として説明した）、他の実施形態において、同一の複数のハッシュテーブルアレイ内で他のハッシュテーブルは異なる数のハッシュバケットを有し、そして同様に、複数のハッシュテーブルアレイ内で他のハッシュバケット、又は同一のハッシュテーブル内であっても、異なる数のハッシュウェイを有する。それだけでなく、複数のハッシュテーブル 4 8 0 が集合的に利用されても、ある側面では、互いに異なるハッシュテーブル 4 8 0 は互いに独立である（例えば、互いに異なるハッシュテーブル 4 8 0 は互いに異なるハッシュ関数を有するか、或いは共通のハッシュ関数を有することができる。）。

30

40

【 0 0 5 7 】

ハッシュテーブルアレイ 4 0 0 が k 個の並列ハッシュテーブル (T 1 , T 2 , … , T k , k は整数) を含み、各ハッシュテーブル 4 8 0 は分離され、独立的なハッシュ関数 (h 1 (x) , h 2 (x) , … , h k (x)) を各々使用する場合、各々のハッシュテーブル (T 1 , T 2 , … , T k) は m 個のハッシュバケット 4 1 0 を含むので、ハッシュ関数 (h 1 (x) , h 2 (x) , … , h k (x)) は前述のように、 $\log_2 (m)$ - ビットの

50

ハッシュを生成し、そして各ハッシュバケット 410 は n 個のハッシュウェイ 420 を含むことから、3次元 (3D) のハッシュテーブルアレイ (例えば、複数のハッシュテーブルアレイ) の容量は $m \times n \times k$ である。

【0058】

各ハッシュテーブル 480 は、如何にしてデータが索引されるかを決定する1つのハッシュ関数に対応する。書き込むためのインカミングデータをハッシュングすることによって、結果計算 (例えば、検索アドレス及びキーを含むハッシュ値) はキー及び値と比較することができ、値が一致する場合、対応するハッシュバケット 410 の参照 / 頻度カウンタライン 340 は増加され、それにより ALUTM 210 の追加的な PLID 350 は特定ラインを示す。

10

【0059】

従来のハッシュテーブルとは異なり、本発明の実施形態による複数のハッシュテーブル 480 は複数の仮想ハッシュバケット / 仮想バケット (Virtual Bucket; VB) 460 を含み、仮想バケット 460 は複数の物理的なハッシュバケット / 物理的なバケット 410 で構成される。以下で“物理的なバケット”は前述したハッシュバケット 310 を示し、仮想バケット 460 から前述したハッシュバケット 310 を区別するために使用される。

【0060】

各仮想バケット 460 は対応するハッシュテーブル 480 の m 個の物理的なバケット 410 の H 個を含み、 H は m より小さい整数である。しかし、同一のハッシュテーブル 480 中で仮想バケット 460 の中で異なる仮想バケットは1つ以上の物理的なバケット 410 を共有できることに注目しなければならない。後述するように、本発明の実施形態による仮想バケット 460 を使用することによって、第4次元が3次元のハッシュテーブルアレイに加えらる。したがって、データを配置し、整列するのに大きな柔軟性が提供され、これにより重複除去DRAMシステムアーキテクチャの効率及び圧縮比率が高くなる。

20

【0061】

本発明の実施形態は、その他の水準のデータ配置柔軟性を高めるために仮想バケット 460 を使用するのに、他の仮想バケット 460 によって共有される他の物理的なバケット 410 を自由にするために、ハッシュテーブル 480 の中でいずれか1つに格納されたデータブロックは対応する仮想バケット 460 内で、又は他の物理的なバケット 410 に移動するようにする。ハッシュテーブル 480 内で空間を自由にして、重複除去は使い道がない / 重複されたデータを除去することによって達成される。即ち、本発明の実施形態による仮想バケット 460 の使用によって、制限された対応位置へのハッシュ関数を利用してデータラインをハッシュングすることによって生じる厳格な制限はなく、そしてデータは近接 / “近い位置” の物理的なバケット 410 に配置されることが可能であり、これは初期の意図された (しかし、占有された) 物理的なハッシュバケット 410 を含む同一の仮想バケット 460 内に存在する物理的なバケット 410 を示す。

30

【0062】

1つの例で、コンテンツ (例えば、データライン C) は k 個のハッシュテーブル ($T_1 (h_1 (C))$ 、 $T_2 (h_2 (C))$ 、...、 $T_k (h_k (C))$) の中のいずれか1つの物理的なバケット 410 の中でいずれか1つに配置される。仮にデータライン C が $T_1 (h_1 (C))$ に配置されようとするれば、データライン C が $T_1 (h_1 (C))$ で表示される物理的なバケット 410 に配置されるための要求をする代わりに、本発明の実施形態は1つの物理的なバケット 410 よりさらに大きく、そして $T_1 (h_1 (C))$ で表示される物理的なバケット 410 のみならず、 H 個の総物理的なバケット 460 を含む仮想バケット 460 を許容する。即ち、仮想バケット 460 は $T_1 (h_1 (C))$ 、 $T_1 (h_1 (C) + 1)$ 、 $T_1 (h_1 (C) + 2)$ 、...、 $T_1 (h_1 (C) + H - 1)$ を含むハッシュテーブル 480 内で整列された近接的であるか、或いは隣接する H 個の物理的なバケット 410 を含む。

40

【0063】

50

したがって、仮想バケット460はデータブロックがハッシュテーブル480内で移動されるか、或いは未来の書込み動作のために空間を自由にするを可能にする。以前にハッシュテーブル480（ハッシュテーブル480の物理的なバケット410を含む仮想バケット460内）に入ったデータブロックの移動を可能とする本発明の実施形態の動作はホップスコッチ（Hopscotch）と呼ぶことができる。メモリ重複除去のための複数のハッシュテーブル480を使用するホップスコッチ動作は後述するように改善することができる。

【0064】

最初に、重複除去DRAMモジュール130はハッシュテーブル480のハッシュ関数の結果としてハッシュテーブル480にデータラインCを挿入しようとする。しかし、しばしば他のデータラインが同一のハッシュ関数の結果として以前にハッシュテーブル480に挿入されていることもある。即ち、他のデータラインは、それが異なることがあるにも拘らず、ハッシュ関数の結果としてハッシュテーブル480内同一の位置に送られる。データラインCが何処に挿入されるか決定するために、動作は先ず $T(h(C))$ として表現される物理的なバケット410で（又は後続の）最初に使用可能な物理的なバケット410を探す。

【0065】

したがって、データラインCを何処に書き込むかを決定するのにおいて、 $T(h(C))$ として表現される初期に意図された物理的なバケット410は占有される可能性があるため、最初に使用可能な物理的なバケット410（即ち、データラインが挿入され得る第1番目の空いた空間）は $T(h(C) + f)$ として表現することができ、ここで、 f は0以上である。 $T(h(C))$ として表現される物理的なバケット410が、対応する仮想バケット460のH個の物理的なバケット410の第1番目の物理的なバケット410であると仮定し、仮に f がHより小さければ（即ち、仮に同一仮想バケット460内で占有されない物理的なバケット410が存在すれば）、データラインCは対応する仮想バケット460に配置される。同様に、 $T(h(C))$ として表現される物理的なバケット410が対応する仮想バケット460の第2番目の物理的なバケットであり、 f が $H - 1$ より小さければ、データラインCは対応する仮想バケット460に配置される。

【0066】

しかし、対応する仮想バケット460の第1番目の物理的なバケット410が意図された物理的なバケット410であると仮定し、仮に f がHより大きいか、或いは同一であれば（即ち、仮想バケット460の物理的なバケット410にデータラインCが入る可能性がない場合）、たとえデータラインCがその仮想バケット460に当てはまらなくても、動作は次のような方式で仮想バケット460内に空いた空間を作るための試みをする。例えば、本発明の実施形態による重複除去DRAMメモリモジュール130は、 $T(h(C) + f - 1)$ で表現される物理的なバケット410がその内に含まれたデータを有するかどうかを決定する時まで $T(h(C) + f - H)$ で表現される物理的なバケット410を開始して物理的なバケット410を探し、その次に $T(h(C) + f - H + 1)$ で表現される物理的なバケット410、及び同様に続けてその他の物理的なバケット410を探す（例えば、仮想バケット460の最初から最後までスキャンする）。その次に、重複除去DRAMメモリモジュール130は $T(h(C) + f - H)$ から $T(h(C) + f - 1)$ までの物理的なバケット410内に入っている任意のデータ対象が空いた空間 $T(h(C) + f)$ に配置される可能性あるか否かを決定する。即ち、重複除去DRAMメモリモジュール130は、 $T(h(C) + f - H)$ から $T(h(C) + f - 1)$ までの物理的なバケットの中でどれかが物理的なバケット $T(h(C) + f)$ を有する共通の仮想バケット460内にいるか否かを決定し、これによりその内に入っているデータを移動させるようにする。その次に、重複除去DRAMメモリモジュール130は、発見された最も初期のそのようなデータ対象を空いた空間内に配置し、これにより $T(h(C) + e)$ （ e は f より小さい整数）で表現される物理的なバケット410内の新しい空いた空間を作る。このような過程は e がHより小さくなるまで反復され（例えば、データは連続的な方式（Ca

10

20

30

40

50

s c a d i n g F a s h i o n) にハッシュテーブル内で移動される))、それにより対応する仮想バケット460内でデータラインCを配置するために必要な空間を確保する。

【0067】

例えば、図5(B)を参照すれば、本発明の実施形態において、物理的なバケットPB2を意図する物理的なバケット410に割当てることができる。意図する物理的なバケットPB2は仮想バケットVB1と関連して占有されているので、仮想バケットVB2は最初から最後までスキャンされる(例えば、物理的なバケットPB2から物理的なバケットPB5まで)。物理的なバケットPB3、PB4、及びPB5もまた占有されているので、最初に使用可能な物理的なバケット410は物理的なバケットPB6である((即ち、fが4と同一であり、そしてfがHと同一であるか、或いはHより大きく、そして最初に使用可能な物理的なバケット410は対応する仮想バケットVB2に存在しない)。したがって、物理的なバケットPB5内データは物理的なバケットPB6に移動することができ、これにより仮想バケットVB2内に空間を確保し、データラインCは対応する仮想バケットVB2内(物理的なバケットPB5内)に配置される。しかし、意図する物理的なバケットがPB1(即ち、対応する仮想バケット460はVB1)であれば、処理過程は物理的なバケットPB4内のデータが仮想バケットVB1から隣接仮想バケットVB2に、即ち、物理的なバケットPB5の新しく確保された空間に移動されるように反復される。その後、データラインCは意図する物理的なバケットPB1に対応する仮想バケットVB1の物理的なバケットPB4に書き込まれる。

10

20

【0068】

したがって、他の仮想バケット460の重複に看做される他の仮想バケット460による特定の物理的なバケット410の共通所有によって、データは1つの仮想バケット460から他の仮想バケット460に移動され、これにより初期ハッシュバケット410のための空間を作ることができる。

【0069】

他の実施形態において、書込み過程の間に、データブロックをハッシュテーブルのレイ400に書き込む要求を受信した後、重複除去DRAMメモリモジュール130は既存の項目が既にハッシュテーブル480の中の1つにあるか否かをチェックするために、データに対して意味がある各ハッシュテーブルの全体仮想バケット460を検索することができる。仮に最初に意図するハッシュテーブル480が一杯に満たされている場合、そしてデータブロックが最初に意図するハッシュテーブル480で探すことができない場合(即ち、各物理的なバケット410の各ハッシュウェイ420が他のデータブロックによって占有されている場合)、重複除去DRAMメモリモジュール130はハッシュテーブルレイ400の他のハッシュテーブル480にデータを入力することを追求する。しかし、複数のハッシュテーブルレイ400のハッシュテーブル480の全部が一杯に満たされている場合、データブロックはバッファメモリ230にこぼれ出る(Spill over)。このような実施形態において、ハッシュテーブルレイ400内でデータ移動は重複除去DRAMメモリモジュール130によって許容されないことが有り得る。

30

【0070】

図5(A)~(C)はそれぞれ本発明の実施形態による仮想バケットと特定の物理的なバケットを連関させるホップワード(Hop words)を生成するための2次元のレイを示す図である。

40

【0071】

図5(A)~(C)を参照すると、本発明の実施形態による多様な仮想バケット460は、ホップワード値591又はホップワードベクトル592の中でいずれか1つを利用すること、そして、データの移動を効果的に追跡するための仮想バケット利用値を利用することによってそれらに対応する物理的なバケット410と関連させることができる。各占有された物理的なバケット410は単一の仮想バケット460に対応するので、ホップワード値591又はホップワードベクトル592はどの仮想バケット460が各占有された

50

物理的バケット410に対応するかを追跡するために使用される。

【0072】

本発明の例で4個の仮想バケットVB0、VB1、VB2、及びVB3の各々は物理的なバケットPB0、PB1、PB2、PB3、PB4、PB5、及びPB6のグループから4個の隣接する物理的なバケットの互いに異なるセットを有する(即ち、Hは4)。

【0073】

例えば、図5(A)及び図5(B)を参照すると、ホップワードベクトル592は物理的なバケット位置及び仮想バケット位置(例えば、準アドレス(quasi-address))で構成される2次元アレイを作ること、そして対応する物理的なバケット410に対応する任意の列には単一の1があることに留意して、各仮想バケット460のためのデータを含む各物理的なバケット410に1(例えば、2進表示子)を配置することによって決定することができる。したがって、ホップワードベクトル592は各仮想バケット460のための物理的なバケット使用を追跡するために使用できる1又は0のアレイを含むことができる。本発明の例において、物理的なバケットPB0、PB1、及びPB3は第1番目の仮想バケットVB0のために占有され、物理的なバケットPB2及びPB4は第2番目の仮想バケットVB1のために占有され、物理的なバケットPB5は第3番目の仮想バケットVB2のために占有され、そして第4番目の仮想バケットVB3は占有されない。

10

【0074】

同様に、図5(C)を参照すると、ホップワード値591は、いずれの仮想バケット460が占有される物理的なバケットに対応するかを認識することによって、占有される物理的なバケット410に基づいて作ることができる。ホップワード値591は $\log_2(H)$ 長さのビットになる(Hは仮想バケット460毎の物理的なハッシュバケット410の数)。

20

【0075】

ホップワードベクトル592又はホップワード値591の情報は各ハッシュバケット410のホップワードラインに格納され、物理的なバケット410及び仮想バケット460の間の関係はメモリに索引される。

【0076】

図6は本発明の実施形態によるハッシュテーブルメモリのデータブロックのアドレッシングのための物理的なラインID(PLID)のブロック図である。

30

【0077】

図6を参照すると、本発明の実施形態による修正されたPLID650が提供される。本発明の実施形態によるPLID650はアドレス、オフセット、テーブルの索引、ハッシュ、そしてスロット/ウェイ(Way)、及び仮想バケット460の間で移動する項目を追跡するための特定の仮想バケット460と対をなすキー651の各々を示す複数のビットを含む。したがって、仮にキー651が特定の仮想バケット460と一致すれば、その特定の仮想バケット460はそこに書き込まれたデータ項目を含む。

【0078】

しかし、他の実施形態において、PLID650はキー651を $\log_2(H)$ ビットを含む仮想バケット利用値フィールド652(例えば、仮想バケット索引)に代替する(例えば、16個の物理的なバケットの高さを有する仮想バケットはPLID650の4ビット仮想バケット利用値フィールドに対応される)。仮想バケット利用値フィールド652は各占有された物理的なバケット410にいずれの仮想バケット460が対応するかを示す。したがって、データ項目を仮想バケット460に書き込む時、仮想バケット460に既に存在する項目の数は計算され、そして仮想バケットに既に存在するアイテムの数に1を加えた数と同一のP値は仮想バケット利用値フィールド652に仮想バケット利用値として書き込まれる。PLID650の仮想バケット利用値を利用することによって、PLID650のストレージ(Storage)過負荷は減少することができる。

40

【0079】

50

図7は本発明の実施形態による、ホップスコッチ方法を使用するメモリモジュールの複数のハッシュテーブルアレイにデータを書き込む過程を説明するためのフローチャートである。

【0080】

図7を参照すると、S701段階の動作で、複数のハッシュテーブルは識別され、ハッシュテーブルの各々はハッシュ関数に対応し、そして各々は物理的なハッシュバケットを含み、各物理的なハッシュバケットはハッシュウェイを含み、データを格納するように構成される(例えば、重複除去DRAMメモリモジュール130はk個のハッシュテーブル480を識別し、各々はハッシュ関数 $h(x)$ に対応し、各々はm個の物理的なハッシュバケット410を含み、各物理的なハッシュバケットはn個のハッシュウェイ420を含む)。

10

【0081】

S702段階で、複数の仮想バケットが識別され、仮想バケットの各々は物理的なハッシュバケットの一部を含み、そして各々は他の仮想バケットと少なくとも1つの物理的なハッシュバケットを共有する(例えば、重複除去DRAMメモリモジュール130は複数の仮想バケット460を識別し、仮想バケット460の各々はm個の物理的なハッシュバケット410のH個を含み、そして図4に示したように、各仮想バケット460は他の仮想バケット460と少なくとも1つの物理的なハッシュバケット410を共有する)。S702a段階で、複数の仮想バケットは、 $\log_2(h)$ ビットの仮想バケット利用値フィールド、及び仮想バケットの中で対応する1つの仮想バケット中のデータブロックの数と同一の値を含む物理的なラインID(PLID)でハッシュテーブルを索引すること、そして仮想バケットの中で対応する1つの仮想バケットに項目が書き込まれる時に仮想バケット利用値フィールドに1を増加させることによって識別される(例えば、図6に示したように、仮想バケット460は仮想バケット利用値フィールド652、及び仮想バケット460の中で対応する1つの仮想バケット中のデータブロックの数と同一の値を含む物理的なラインID(PLID)650でハッシュテーブル480を索引することによって識別され、仮想バケット利用値フィールド652は仮想バケット460の中で対応する1つの仮想バケットに項目又はデータブロックが書き込まれる時に1が増加される)。

20

【0082】

S703段階で、格納されたデータを有する物理的なハッシュバケットの各々は、仮想バケットの中で対応する1つの仮想バケットに割当てられていると識別される(例えば、重複除去DRAMメモリモジュール130は図5(B)及び図5(C)で示したようにPB0、PB1、PB2、PB3、PB4、及びPB5に格納されたデータを有する物理的なハッシュバケット410を仮想バケット460(VB0、VB1、及びVB2)の中で対応する1つの仮想バケットに割当てて識別する)。S703a段階で、データを含む物理的なハッシュバケットの中で何れが仮想バケットの中のいずれに対応するかを示すためのホップワードベクトル又はホップワード値を生成することによって識別される(例えば、図5(B)及び図5(C)で示したように、重複除去DRAMメモリモジュール130はデータを含む物理的なハッシュバケット410の中でいずれが仮想バケット460の中のいずれに対応するかを示すためのホップワードベクトル592又はホップワード値591を生成する)。

30

40

【0083】

S704段階で、データラインはハッシュ値を生成するためにハッシュ関数の中で対応する1つのハッシュ関数にしたがってハッシュされる(例えば、重複除去DRAMメモリモジュール130はメモリコントローラ120からデータラインCに対応する書込み要求を受信し、そしてハッシュ値を生成するためにハッシュ関数 $h(x)$ の中で対応する1つのハッシュ関数にしたがってインカミング(*incoming*)データをハッシュする)。

【0084】

S705段階で、対応するハッシュテーブルの仮想バケットの中で対応する1つの仮想

50

バケットがハッシュ値にしたがうデータブロックのための使用可能な空間を有するか否かを決定する（例えば、図5（B）及び図5（C）で示したように、重複除去DRAMメモリモジュール130は仮想バケット460（VB3）がデータブロックのための物理的なバケットPB6に空間を有するか決定する）。

【0085】

S706段階で、仮想バケットの中で対応する1つの仮想バケットが使用可能な空間を有しない場合、データは、仮想バケットの中で対応する1つの仮想バケットがデータブロックのための空間を有するようになるまで仮想バケットの中で対応する1つの仮想バケットから仮想バケットの中で隣接する1つの仮想バケットに順次移動される。（例えば、図5（B）及び図5（C）で示したように、重複除去DRAMメモリモジュール130は、仮想バケットVB2が任意の他の使用可能な物理的なバケットを有しない場合、仮想バケットVB2がデータブロックのための空間を有するようになるまで仮想バケットVB2の物理的なバケットPB5から仮想バケットVB3にデータを順次移動させ、上述した過程は仮に仮想バケットVB1が仮想バケット460の中で対応する1つの仮想バケットであれば、仮想バケットVB1の物理的なバケットPB4から仮想バケットVB2の物理的なバケットPB5にデータを移動させるために反復される）。S706a段階で、アドレス検索テーブルメモリは移動されたデータブロックに対応する1つ又はそれ以上の検索アドレスに変更するためにアップデートされる（例えば、重複除去DRAMメモリモジュール130はハッシュテーブルメモリ220の移動されたデータブロックの新しいアドレスが検索できるように移動されたデータブロックに対応する1つ又はそれ以上のアドレスポインターを変更するためにALUTM210をアップデートする）。

10

20

【0086】

S707段階で、データブロックは仮想バケットの中で対応する1つの仮想バケットに格納される（例えば、図5（B）及び図5（C）に示したように、重複除去DRAMメモリモジュール130は、仮に仮想バケットVB1が意図する仮想バケット460であれば、データブロックを仮想バケットVB1の物理的なバケットPB4に格納する）。仮に、仮想バケットVB1を含むハッシュテーブル480が一杯に満たされたら決定されれば、データブロックはバッファメモリ230に格納される。

【0087】

図8は本発明の実施形態による、メモリモジュールの複数のハッシュテーブルアレイからデータを読み出す過程を説明するためのフローチャートである。

30

【0088】

S801段階で、ハッシュテーブルアレイに格納された複数のデータブロックに対応する読み出し要求が受信される（例えば、重複除去DRAMメモリモジュール130は、メモリコントローラ120からデータラインCで構成される複数のデータブロックに対応する読み出し要求を受信し、複数のデータブロックはハッシュテーブルメモリ220のハッシュテーブルアレイ400に格納されている）。

【0089】

S802段階で、複数のデータブロックに対応するポインターの中で対応するポインターはALUTM210から検索される（例えば、重複除去DRAMメモリモジュール130はALUTM210からデータラインCで構成される複数のデータブロックに対応するアドレスポインターを検索する）。

40

【0090】

S803段階で、ポインターの中で対応するポインターに基づく複数のデータブロックはハッシュテーブルメモリでアクセスされる（例えば、重複除去DRAMメモリモジュール130はハッシュテーブルメモリ220のハッシュテーブルアレイ400内の検索されたアドレスポインターに対応する異なるアドレスからデータブロックにアクセスし、検索する）。

【0091】

S804段階で、複数のデータブロックは再構成されたデータを生成するために再構成

50

される（例えば、重複除去DRAMメモリモジュール130は受信した読出し要求に対応するデータラインCと同等である再構成されたデータを生成するためにハッシュテーブルメモリ220から検索されたデータブロックを再構成する）。

【0092】

S805段階で、再構成されたデータはメモリモジュールからメモリコントローラへ伝送される（例えば、重複除去DRAMメモリモジュール130はデータラインCをメモリコントローラ120に伝送する）。

【0093】

前述したように、データ重複除去は本発明の実施形態による重複除去DRAMメモリモジュールを使用して遂行することができる。したがって、メモリアクセスは減少することができ、DRAMシステムの寿命を延長することができる。

10

【0094】

前述した内容は例示的な実施形態を示し、それによって限定されない。幾つかの例示的な実施形態を説明したが、当業者は例示的な実施形態で新規な教示及び例示的な実施形態の長所から逸脱することなく、多様な変形が可能なのは容易に理解できる。したがって、そのようなすべての変形は請求範囲に記載された例示的な実施形態の範囲内に含まれることと意図される。請求範囲で、機能的な節は引用された機能を遂行する構造及び構造的な均等物のみならず、均等な構造物も含むことと意図される。したがって、前述した内容は開示された特定な実施形態に制限されなく、そして開示された例示的な実施形態の変形のみならず、他の例示的な実施形態は添付された請求項の範囲内に含まれると意図されることと理解されなければならない。本発明は請求項に含まれる請求項の均等物内の、以下の請求範囲によって定義される。

20

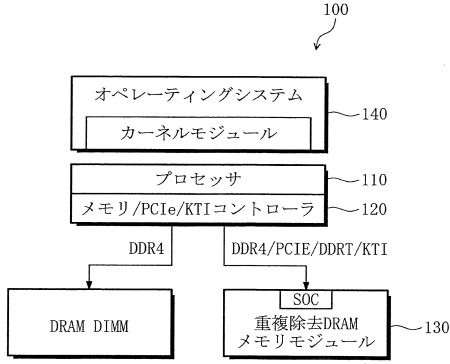
【符号の説明】

【0095】

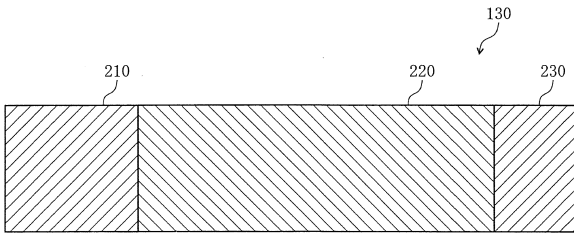
- 100 重複除去DRAMシステムアーキテクチャ
- 110 プロセッサ
- 120 メモリ/PCIe/KTIコントローラ
- 130 重複除去DRAMメモリモジュール
- 140 オペレーティングシステム/CPUモジュール
- 210 アドレス検索テーブルメモリ(ALUTM)
- 220 ハッシュテーブルメモリ
- 230 バッファメモリ
- 310、410 ハッシュバケット
- 320 ハッシュウェイ
- 330 署名ライン
- 340 参照/頻度カウントライン
- 380、480 ハッシュテーブル
- 400 ハッシュテーブルアレイ
- 650 PLID

30

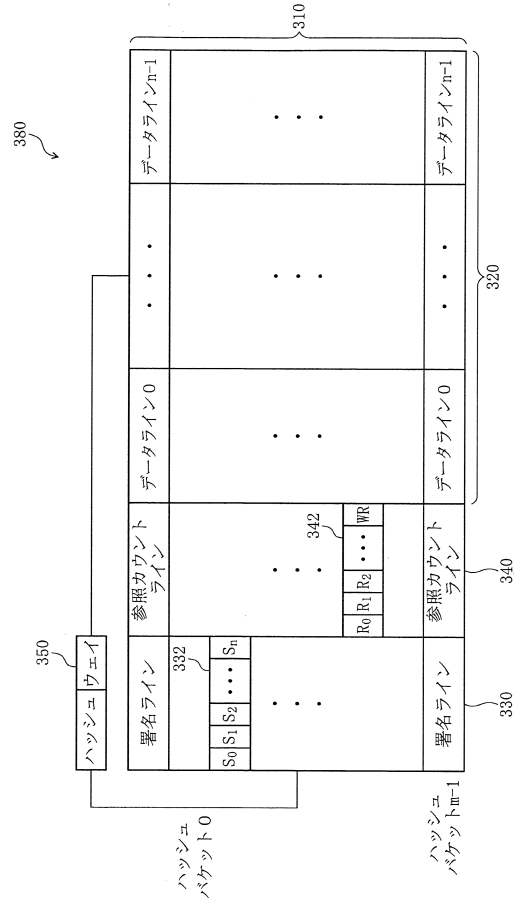
【図1】



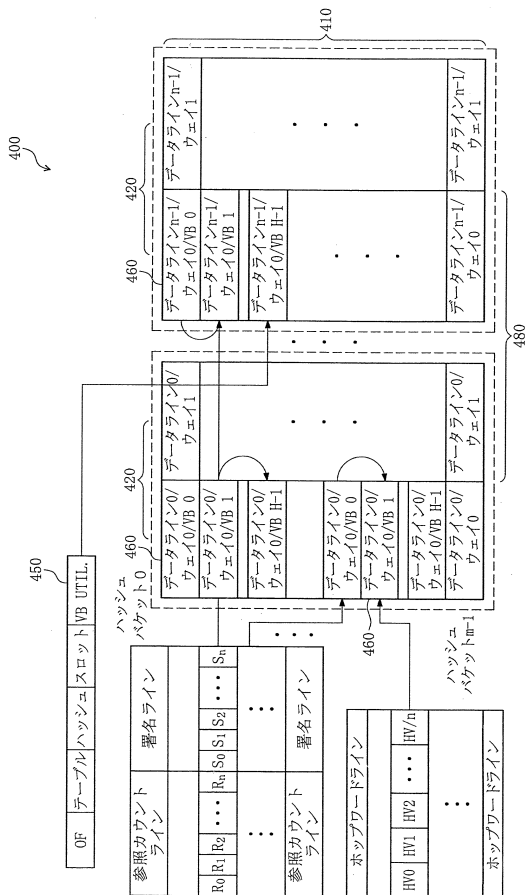
【図2】



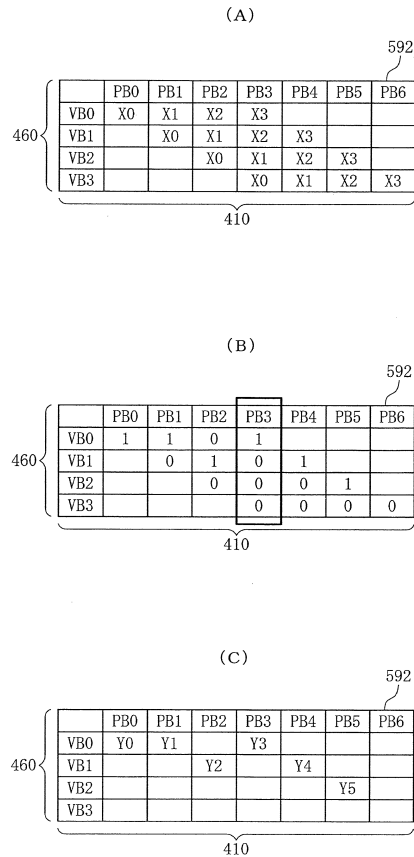
【図3】



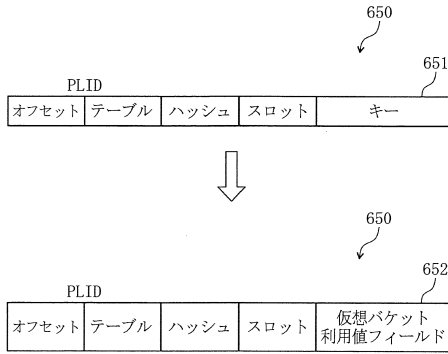
【図4】



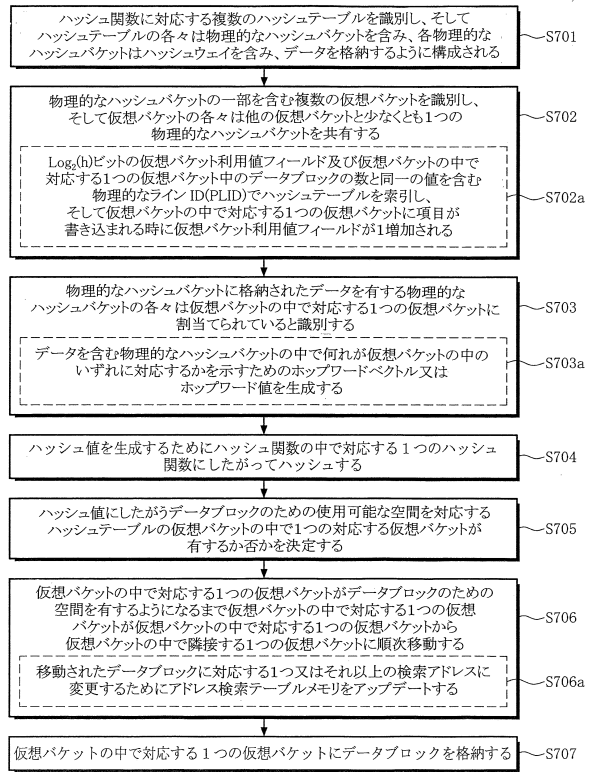
【図5】



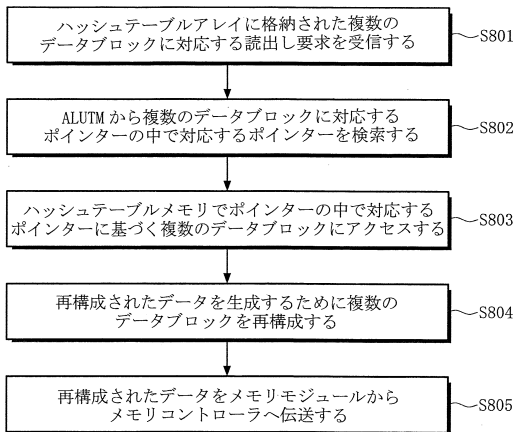
【図6】



【図7】



【図8】



フロントページの続き

- (72)発明者 胡 潮 ホン
アメリカ合衆国 9 5 1 4 8 カリフォルニア州 サンノゼ シェミン デ リビエラ 3 4 6 1
- (72)発明者 チェン 宏 忠
アメリカ合衆国 9 4 0 8 7 カリフォルニア州 サニーベール パーキンソン アベニュー 1
2 9 5
- (72)発明者 張 牧 天
アメリカ合衆国 9 5 0 5 1 カリフォルニア州 サンタクララ ピア トリノ プレイス 2 9
2 0

審査官 野村 和史

- (56)参考文献 特開2009-087021(JP,A)
国際公開第2016/006050(WO,A1)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 1 6 / 0 0 - 1 6 / 9 5 8
1 2 / 0 0 - 1 2 / 0 6
1 3 / 1 6 - 1 3 / 1 8