

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2014 年 7 月 3 日(03.07.2014)



(10) 国際公開番号
WO 2014/103489 A1

- (51) 国際特許分類:
G06F 12/08 (2006.01) *G06F 12/16* (2006.01)
G06F 3/06 (2006.01)
- (21) 国際出願番号: PCT/JP2013/078731
- (22) 国際出願日: 2013 年 10 月 23 日(23.10.2013)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
PCT/JP2012/008459 2012 年 12 月 28 日(28.12.2012) JP
- (71) 出願人: 株式会社日立製作所 (HITACHI, LTD.)
[JP/JP]; 〒1008280 東京都千代田区丸の内一丁目
6 番 6 号 Tokyo (JP).
- (72) 発明者: 杉本 定広 (SUGIMOTO, Sadahiro); 〒
1008280 東京都千代田区丸の内一丁目 6 番 6 号
株式会社日立製作所内 Tokyo (JP). 山本 彰
(YAMAMOTO, Akira); 〒1008280 東京都千代田区
丸の内一丁目 6 番 6 号 株式会社日立製作所内
Tokyo (JP). 本間 繁雄 (HOMMA, Shigeo); 〒
1008280 東京都千代田区丸の内一丁目 6 番 6 号
株式会社日立製作所内 Tokyo (JP).
- (74) 代理人: 特許業務法人第一国際特許事務所 (PAT-
ENT CORPORATE BODY DAI-ICHI KOKUSAI
TOKKYO JIMUSHO); 〒1080014 東京都港区芝 4
丁目 10 番 5 号 Tokyo (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保
護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA,
BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN,
CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES,
FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN,
IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR,
LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX,
MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH,
PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK,
SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA,
UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保
護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW,
MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシ

[続葉有]

(54) Title: INFORMATION PROCESSING DEVICE HAVING A PLURALITY OF TYPES OF MEMORY CACHES WITH DIFFERENT CHARACTERISTICS

(54) 発明の名称: 特性の異なる複数種類のキャッシュメモリを有する情報処理装置

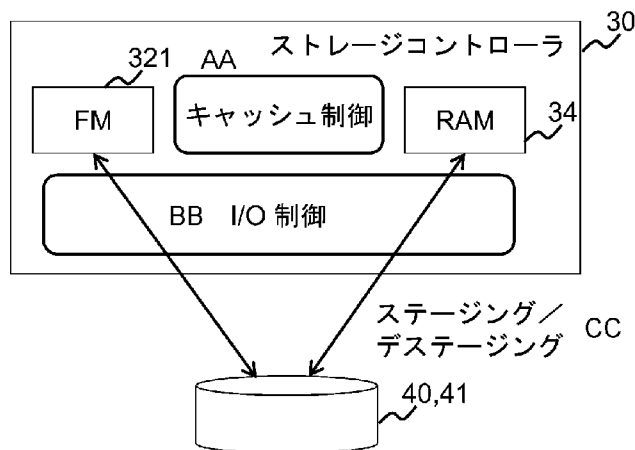


図 1

(57) Abstract: An information processing device comprises a plurality of types of memory caches with different characteristics, and, on the basis of the access characteristics of data to be cached, determines the type of memory cache whereto the data is to be cached, and caches the data in the memory cache of the determined type.

(57) 要約: 情報処理装置は、特性の異なる複数種類のキャッシュメモリを有し、キャッシュ対象のデータのアクセス特性に基づいて、データのキャッシュ先とするキャッシュメモリの種類を決定し、決定した種類のキャッシュメモリにデータをキャッシュする。

30 STORAGE CONTROLLER
AA CACHE CONTROL
BB I/O CONTROL
CC STAGING/DESTAGING



ア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ
(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR,
GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT,
NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI
(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML,
MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：特性の異なる複数種類のキャッシュメモリを有する情報処理装置

技術分野

[0001] 本発明は、データのキャッシュ制御の技術に関する。

背景技術

[0002] メモリの一例として、半導体不揮発性メモリであるフラッシュメモリが知られている。フラッシュメモリは、DRAMやSRAM等の揮発性メモリ（以下、RAMと表記する。）に比べて記憶密度を高めやすく、容量当たりのコスト（ビットコスト）を安くすることが可能である。また、フラッシュメモリは、磁気ディスク等に比べて高速なデータアクセスが可能である。そのため、フラッシュメモリをディスクキャッシュとして利用することで安価な大容量ディスクキャッシュを作ることが可能となる。

[0003] しかし、フラッシュメモリには次のような制限がある。まず、フラッシュメモリの各ビットの更新は1から0（または0から1）の一方向に限定される。そして、ビットに対して逆の変更が必要な場合は、ブロックからデータの消去を行って一旦ブロック全体を1（または0）にする必要がある。ここで、フラッシュメモリが有している複数のブロック（物理ブロック）をブロックと呼ぶ。各ブロックは、複数のページ（物理ページ）で構成されている。また、フラッシュメモリにおけるブロックの消去回数には上限があり、例えばSLC（Single Level Cell）のNAND型フラッシュメモリの場合には、消去回数の上限が、一万から十万回程度であり、MLC（Multiple Level Cell）のNAND型フラッシュメモリの場合には、消去回数の上限が数千回程度である。そのため、フラッシュメモリをディスクキャッシュとして使用する場合、高頻度な書き換えを行うと、比較的短期間に消去回数が上限に達してしまい、使えなくなるおそれがある。

[0004] また、フラッシュメモリのアクセス性能はRAMに比べると低いため、RAMの代わりにフラッシュメモリをディスクキャッシュに用いると、ディス

クキャッシュがシステム性能のボトルネックになるおそれがある。

[0005] また、フラッシュメモリ以外にも相変化メモリ、磁気抵抗メモリ、抵抗変化メモリなどの不揮発性半導体メモリも開発されており、これらの不揮発性半導体メモリもRAMに比べて記憶密度を高めやすく、RAMに比べて安価で大容量なメモリとなる可能性がある。しかしながら、これらの不揮発性半導体メモリも、一般的には、RAMに比べると低速であったり、あるいは短寿命であったりする。

[0006] キャッシュに関する技術としては、例えば特許文献1に開示された技術のように、RAMなどの揮発性メモリを一時格納先とし、不揮発性メモリを補助的に使用する技術が知られている。

先行技術文献

特許文献

[0007] 特許文献1：米国特許第8 1 3 1 9 3 0号明細書

発明の概要

発明が解決しようとする課題

[0008] フラッシュメモリなどの不揮発性半導体メモリは、一般的にRAMに比べてアクセス性能が低く、HDDに比べてアクセス性能が高い特性を有する。そのため、特許文献1のように、フラッシュメモリをRAMキャッシュと、HDD等のデータの最終格納先（最終ストレージデバイス）との間のキャッシュとして使用する方式が用いられている。

[0009] しかしながら、先に述べたとおりフラッシュメモリなどの不揮発性半導体メモリは消去回数の制約があり、高頻度な書き換えが行われる場合、比較的短期間のうちに消去回数が上限に達し、使用できなくなる（つまり、短寿命である）。一般にディスクキャッシュのアクセス頻度は、HDD等の最終ストレージデバイスのアクセス頻度よりも高いため、フラッシュメモリをディスクキャッシュとして用いる場合、フラッシュメモリを最終ストレージデバイスとして用いる場合に比べ、より早期に使用できなくなる（長期間の使用

に耐えられない) 可能性が高い。フラッシュメモリ部分が交換可能な装置構成であったとしても、フラッシュメモリ部分が短寿命であれば、当該部分の交換頻度が高くなるため、装置の保守管理コストを押し上げることになる。

[0010] また、特許文献1のようにRAMとフラッシュメモリとの階層構造を持つキャッシュ方式では、例えばリード対象データがフラッシュメモリに格納されていた場合、リード対象データを一旦RAMキャッシュにステージングした後に当該データをホスト計算機へ送信するため、ステージング処理のオーバーヘッドが発生する。昨今のストレージシステムに対する要求性能の伸びを鑑みるに、この処理オーバーヘッドは無視することのできないものであり、より処理オーバーヘッドの小さいI/O処理方式が求められる。

[0011] また、前述のとおり、フラッシュメモリなどの不揮発性半導体メモリのアクセス性能は一般的にRAMに比べて低いため、不揮発性半導体メモリを用いたディスクキャッシュがシステム性能のボトルネックになるおそれがある。

課題を解決するための手段

[0012] 本発明の情報処理装置は、特性の異なる複数種類のキャッシュメモリを有し、キャッシュ対象のデータのアクセス特性に基づいて、データのキャッシュ先とするキャッシュメモリの種類を決定し、決定した種類のキャッシュメモリにデータをキャッシュする。情報処理装置は、例えば、複数のストレージデバイスと、複数のストレージデバイスに接続されたコントローラとを有したストレージ装置であって良い。コントローラが、前述の複数種類のキャッシュメモリと、それら複数種類のキャッシュメモリに接続された制御デバイスとを有して良い。複数のストレージデバイスの各々は、例えば、後述の最終ストレージデバイスで良い。

[0013] 本発明の一実施形態では、特性の異なる複数種類のキャッシュメモリとして、フラッシュメモリとRAMが用いられる。フラッシュメモリとRAMの特性を比較すると、フラッシュメモリはRAMと比べて、アクセス性能が低く、また書き換え回数に制限がある等の特性の違いがあるので、情報処理装

置は、キャッシュメモリに格納されるデータが、RAMの特性に適合するデータは、RAMを用いたキャッシュメモリにキャッシュし、フラッシュメモリの特性に適合するデータは、フラッシュメモリを用いたキャッシュメモリにキャッシュする。具体的には、たとえば高スループットが求められる、あるいは更新頻度が高いと判断されるデータは、フラッシュメモリを用いたキャッシュメモリには書き込まず、RAMを用いたキャッシュメモリに直接キャッシュするよう制御する。

発明の効果

[0014] 本発明によれば、高価なRAMに加え、RAMよりも低価格でHDDよりもアクセス性能が高いフラッシュメモリなどの記憶媒体をキャッシュとして適切に利用することができるようになるので、低価格で大容量のキャッシュを備えた情報処理装置を提供することができる。

図面の簡単な説明

[0015] [図1]図1は、本発明に係るキャッシング先選択処理の概要を示す概念図である。

[図2]図2は、実施例1に係る情報システムの第1の構成例を示す図である。

[図3]図3は、実施例1に係る情報システムの第2の構成例を示す図である。

[図4]図4は、実施例1に係るFMボードの構成図である。

[図5]図5は、実施例1に係るストレージコントローラのRAMの構成図である。

[図6]図6は、実施例1に係るアクセスモニタテーブルの構成図である。

[図7]図7は、実施例1に係るキャッシュ管理データ構造の概念図である。

[図8]図8は、実施例1に係るキャッシュ管理データ構造の一部のデータ構造を示す図である。

[図9]図9は、実施例1に係るダーティキューおよびクリーンキューのデータ構造を示す図である。

[図10]図10は、実施例1に係るFMフリーキューおよびRAMフリーキューのデータ構造を示す図である。

[図11]図 1 1 は、実施例 1 に係るリードコマンド処理のフローチャートである。

[図12]図 1 2 は、実施例 1 に係るステージング処理のフローチャートである。

[図13]図 1 3 は、実施例 1 に係るデータ送信処理のフローチャートである。

[図14]図 1 4 は、実施例 1 に係るキャッシュ割り当て処理のフローチャートである。

[図15]図 1 5 は、実施例 1 に係る F M 優先セグメント割り当て処理のフローチャートである。

[図16]図 1 6 は、実施例 1 に係る R A M 優先セグメント割り当て処理のフローチャートである。

[図17]図 1 7 は、実施例 1 に係るアクセスモニタ集計処理のフローチャートである。

[図18]図 1 8 は、実施例 1 に係る判定用閾値算出方法を説明する図である。

[図19]図 1 9 は、実施例 1 に係るライトコマンド処理のフローチャートである。

[図20]図 2 0 は、実施例 1 に係るデータ受領処理（R A M）のフローチャートである。

[図21]図 2 1 は、実施例 1 に係るデータ受領処理（F M）のフローチャートである。

[図22]図 2 2 は、実施例 1 に係る F M データ読み出し処理のフローチャートである。

[図23]図 2 3 は、実施例 1 に係る F M データ書き込み処理のフローチャートである。

[図24]図 2 4 は、実施例 2 に係る情報システムの構成図である。

[図25]図 2 5 は、実施例 2 に係るデータ入出力処理の概要を示す図である。

[図26]図 2 6 は、実施例 3 に係る情報システムの構成を示す図である。

[図27]図 2 7 は、実施例 3 に係るデータ入出力処理の概要を示す図である。

[図28]図28は、実施例4に係るジョブ制御テーブルの構成図である。

[図29]図29は、実施例4に係るリードコマンド処理の第1のフローチャートである。

[図30]図30は、実施例4に係るリードコマンド処理の第2のフローチャートである。

[図31]図31は、実施例4に係るステージング処理のフローチャートである。

[図32]図32は、実施例4に係るデータ送信処理のフローチャートである。

[図33]図33は、実施例5に係るメモリ種類見直し処理のフローチャートである。

[図34]図34は、本発明の実施例に係る、論理ボリュームとキャッシュ管理データとの関係を表した概念図である。

発明を実施するための形態

[0016] 最初に、本発明の概要を説明する。図1は、本発明の情報処理装置の一例であるストレージシステムにおける、キャッシング先選択処理の概要を示す概念図である。

[0017] 企業や組織で扱われる電子情報は急増していることから、大量のデータをより高速に処理するために、ユーザのITシステムに対する要求性能は年々高くなってきている。一方でITシステムのコストを削減したいというニーズも高く、ユーザからは、高性能であるとともに、低価格な情報処理装置が求められている。たとえば大量のデータアクセスが発生する計算機システムでは、データを格納するためのストレージシステムの性能の良し悪しが、システムの性能を大きく左右する。

[0018] ストレージシステムには、そのアクセス性能を向上させるために、キャッシュ（ディスクキャッシュ）が設けられる。一般にキャッシュには、データの最終的な格納先（最終ストレージデバイス）であるハードディスク等よりもアクセス性能の高い記憶媒体である、DRAMやSRAMなどのランダムアクセスメモリ（以下、「RAM」と呼ぶ）が用いられることが多い。RA

Mの欠点は、高価格（ビットコストが高い）ことであり、RAMを用いたキャッシュをストレージシステムに多く搭載すれば、ストレージシステムの平均的な性能を引き上げることはできるが、高価格になるという問題がある。

[0019] 本発明におけるストレージシステムは、図1に示すように、ストレージコントローラ30上にキャッシュとして、FM321とRAM34を搭載し、最終ストレージデバイスであるHDD40やSSD41に格納されるデータを、FM321またはRAM34のいずれかにキャッシュする。FMはRAMと比べて、低ビットコストであるため、低価格で大容量のキャッシュ記憶領域を提供することができる。

[0020] 一方FMとRAMの特性を比較すると、FMはRAMと比べてアクセス性能が低く、また書き換え回数に制限がある等の特性の違いがあるので、ストレージコントローラ30は、キャッシュ対象のデータのアクセス特性に応じて、RAMの特性に適合するデータは、キャッシュ先としてRAM34を選択し、フラッシュメモリの特性に適合するデータは、キャッシュ先としてFM321を選択する制御を行う。データのアクセス特性としては、例えば、キャッシュ対象データのアクセス頻度や、アクセスパターン等が用いられ、ストレージコントローラ30はたとえば、高スループットが求められる、あるいは更新頻度が高いと判断されるデータは、フラッシュメモリを用いたキャッシュメモリには書き込まず、RAM34をキャッシュ先として選択する。

[0021] これにより、高価なRAMに加え、RAMよりも低価格でHDDよりもアクセス性能が高いフラッシュメモリなどの記憶媒体をキャッシュとして適切に利用することができるストレージシステムを実現できる。本発明におけるストレージシステムは特に、オンライントランザクション処理（OLTP）やERP（Enterprise Resource Planning）等の大量のデータに対するデータアクセスを行う業務アプリケーションが実行される、計算機システムにおける利用に好適である。

[0022] 以下では、いくつかの実施例について、図面を参照して説明する。なお、

以後の説明では「a a aテーブル」等の表現にて情報を説明する場合があるが、これら情報は、テーブル等のデータ構造以外で表現されていてもよい。そのため、データ構造に依存しないことを示すために「a a aテーブル」等について「a a a情報」と呼ぶことがある。

- [0023] また、以後の説明では「プログラム」を主語として説明を行う場合があるが、プログラムは、プロセッサ（典型的にはCPU（Central Processing Unit））を含む制御デバイスによって実行されることで、定められた処理をメモリ及びI/F（インタフェース）を用いながら行うため、プロセッサ又は制御デバイスを主語とした説明としてもよい。制御デバイスは、プロセッサであっても良いし、プロセッサとハードウェア回路を含んでも良い。また、プログラムを主語として開示された処理は、ホスト計算機、ストレージシステムが行う処理としてもよい。また、プログラムの一部または全ては専用ハードウェアによって実現されてもよい。また、各種プログラムはプログラム配布サーバや、計算機が読み取り可能な記憶メディアによって各計算機にインストールされてもよい。記憶メディアとしては、例えば、ICカード、SDカード、DVD等であってもよい。

実施例 1

- [0024] まず、実施例 1 に係る情報システムについて説明する。
- [0025] 図 2 は、実施例 1 に係る情報システムの第 1 の構成例を示す図である。
- [0026] 情報システムは、ホスト計算機 10 と、ホスト計算機に直接あるいはネットワークを介して接続されるストレージシステム 20（情報処理装置の一例）とを有する。ストレージシステム 20 は、ストレージコントローラ 30 と、ストレージコントローラ 30 に接続される HDD（Hard Disk Drive）40 および／または SSD（Solid State Drive）41 とを有する。HDD 40 および／または SSD 41 は、ストレージデバイスの一例である。HDD 40 および／または SSD 41 は、ストレージコントローラ 30 に内蔵されていてもよい。
- [0027] ストレージコントローラ 30 は、一つ以上のフロントエンドインタフェー

ス（F E I / F）31、1つ以上のバックエンドインタフェース（B E I / F）35、1つ以上のFM（フラッシュメモリ）ボード32、CPU33、およびRAM（Random Access Memory）34を含む。RAM34は、メモリ（メモリデバイス）であり、キャッシュメモリの一例である。

[0028] 本発明の実施例1に係るストレージコントローラ30は、複数のストレージデバイス（HDD40やSSD41）から1ないし複数の論理ボリューム（実体的な論理ボリューム）を形成して、ホスト計算機10に提供する（ホスト計算機10が当該形成された論理ボリュームを認識できるようにする）。あるいは、いわゆるシンプロビジョニング技術によって形成される論理ボリューム（仮想的な論理ボリュームであって、当該仮想的な論理ボリューム内の各領域には、動的に記憶領域が割り当てられる）をホスト計算機10に提供する。ホスト計算機10は、この提供された論理ボリューム（実体的な論理ボリュームまたは仮想的な論理ボリューム）及び論理ボリューム内の位置（論理ブロック番号。LBAと略記されることもある）を指定したI/Oコマンド（ライトコマンド又はリードコマンド）を発行して、論理ボリュームに対するデータのリード・ライト処理を行う。ただし、ストレージコントローラ30が論理ボリュームを提供するものでない態様、たとえばストレージシステム20が、各HDD40、各SSD41を、それぞれ単一の記憶デバイスとしてホスト計算機10に提供する構成であっても、本発明は有効である。なお、ホスト計算機が認識する論理ボリュームのことを、論理ユニット（Logical Unit。LUと略記されることもある）と呼ぶこともあり、本明細書においては特に断りのない限り、論理ボリュームと論理ユニット（LU）の語は、いずれも同一の概念を意味するものとして用いられている。

[0029] F E I / F 31は、ホスト計算機10と通信するためのインタフェースデバイスである。B E I / F 35は、HDD40又はSSD41と通信するためのインタフェースデバイスである。B E I / F 35は、例えばSASやFibre Channelのインタフェースデバイスである。FMボード32は、FMチップ321（図4参照）を搭載したボードである。CPU33は、各

種処理を実行する。RAM 34は、CPU 33で実行するプログラム及び各種テーブルを記憶する。RAM 34は、キャッシュメモリ領域を含んでおり、キャッシュメモリ領域は、複数のキャッシュセグメントで構成される。キャッシュセグメントは、CPU 33によって管理される単位領域である。例えば、RAM 34におけるキャッシュメモリ領域では、キャッシュセグメント単位で、領域確保、データの読出し及びデータの書込みが行われて良い。キャッシュメモリ領域には、最終ストレージデバイスから読み出されるデータ、及び、最終ストレージデバイスに書き込まれるデータ（典型的には、ホスト計算機10からのI/Oコマンド（ライトコマンド又はリードコマンド）に従うデータであるユーザデータ）がキャッシュされる（一時的に格納される）。最終ストレージデバイスとは、I/Oコマンドで指定されているI/O先に従いストレージコントローラ30によってI/Oが行われるデータが格納されるストレージデバイスである。具体的には、例えば、I/Oコマンド（ライトコマンド）に従うデータは、一旦はキャッシュメモリ領域に格納されるが、その後、I/Oコマンドで指定されている論理ユニット（論理ボリューム）を構成しているストレージデバイスの領域（論理ボリュームが仮想的な論理ボリュームである場合には、当該論理ボリュームの領域に対して割り当てられているストレージデバイスの領域）に格納される。最終ストレージデバイスとは、この論理ボリュームを形成するストレージデバイスのことを意味する。本実施例では、最終ストレージデバイスは、HDD又はSSDであるが、他種のストレージデバイス、例えば、複数のストレージデバイスを有する外部ストレージシステムであっても良い。

[0030] 図2では、情報システムの各構成要素を1つずつ図示しているが、冗長化、高性能化、あるいは大容量化などのために、各構成要素を複数備えるようにしてもよい。また、各構成要素間はネットワークを介して接続されてもよい。ネットワークは、スイッチやエキスパンダなどを含んでも良い。情報システムとしては、例えば図3のような構成が考えられる。

[0031] 図3は、実施例1に係る情報システムの第2の構成例を示す図である。

- [0032] 図3に示す情報システムは、2つのストレージコントローラ30（ストレージコントローラA、ストレージコントローラB）を含む、これらのストレージコントローラ30は、ノードインタフェース（ノードI/F）36を介して接続される。ノードインタフェース36は、例えばInfiniband、Fibre Channel（FC）、Ethernet（登録商標）等のネットワークインタフェースのデバイスでもよく、また、PCI Expressのようなバスのインタフェースのデバイスでもよい。
- [0033] これらストレージコントローラ30は、Fibre Channel、Ethernet、Infiniband等のネットワーク50を介してホスト計算機10に接続される。図3では、ネットワーク50をSAN(Storage Area Network)と総称する。
- [0034] 情報システムは、ドライブエンクロージャ60を有する。ドライブエンクロージャ60は、複数のHDD40やSSD41を格納する。複数のHDD40やSSD41は、ドライブエンクロージャ60内のエキスパンダ42に接続される。エキスパンダ42は、各ストレージコントローラ30のBEI/F35に接続される。BEI/F35がSASのインタフェースデバイスである場合には、エキスパンダ42は、例えばSAS Expanderであり、BEI/F35がFibre Channelのインタフェースデバイスである場合、エキスパンダ42は、例えばFCスイッチである。
- [0035] 図3では、ストレージシステム20は、ドライブエンクロージャ60を1つ備えているが、複数のドライブエンクロージャ60を備えてもよい。この場合、BEI/F35の複数のポートのそれぞれに対して各ドライブエンクロージャ60を直接接続してもよいし、BEI/F35のポートに、スイッチを経由して複数のドライブエンクロージャ60を接続してもよい。また、各ドライブエンクロージャ60のエキスパンダ42同士をカスケード接続することで複数のドライブエンクロージャ60を数珠繋ぎにして、BEI/F35のポートに接続するようにしてもよい。
- [0036] 図4は、実施例1に係るFMボードの構成図である。
- [0037] FMボード32は、一つ以上のフラッシュメモリ（FM）チップ321、

F Mアダプタ 3 2 0、バスコネクタ 3 2 2、バッファメモリ 3 2 3、及びバッテリー 3 2 4を含む。なお、本実施例及び以降の実施例では、フラッシュメモリチップ 3 2 1を備えたメモリボードであるF Mボード 3 2を代表例として説明をするが、F Mボード 3 2に代えて、フラッシュメモリ以外の不揮発半導体メモリ、例えばP R A M（相変化メモリ）やM R A M（磁気抵抗メモリ）、R e R A M（抵抗変化メモリ）のチップを備えたメモリボードを用いてもよい。F Mボード 3 2のようなメモリボードは、メモリ（メモリデバイス）であって、キャッシュメモリの一例である。

[0038] F Mチップ 3 2 1は、例えば、N A N D型のフラッシュメモリのチップである。本実施例では、複数のF Mチップ 3 2 1は、キャッシュメモリ領域として使用され、C P U 3 3によって、複数のキャッシュセグメントとして管理される。1つのキャッシュセグメントのサイズは、例えば、F Mチップ 3 2 1の消去単位であるブロックの複数個分のサイズである。F Mチップ 3 2 1は、R A M 3 4よりもアクセス性能が低く、また、データの消去回数に制限があるという特性を有している。1つのF Mチップ 3 2 1は、複数のブロック（物理ブロック）で構成されている。1つの物理ブロックは、複数のページ（物理ページ）で構成されている。

[0039] バスコネクタ 3 2 2は、ストレージコントローラ 3 0上のP C I ExpressなどのバスにF Mボード 3 2を接続するための接続端子である。例えば、F Mボード 3 2とストレージコントローラ 3 0の主基板とを一体のものとして実装する場合は、バスコネクタ 3 2 2を省略する構成としてもよい。

[0040] バッファメモリ 3 2 3は、例えば、D R A MやS R A MなどのR A Mであり、外部からF Mチップ 3 2 1へのデータ転送時や、F Mチップ 3 2 1から外部へのデータ転送時のバッファとして用いられる。バッファメモリ 3 2 3は、F Mプロセッサ 3 2 0 bが実行するプログラムや、F Mプロセッサ 3 2 0 bやD M A C 3 2 0 d等が使用するデータを格納してもよい。

[0041] バッテリー 3 2 4は、バッファメモリ 3 2 3によるデータの記憶に必要な電力のバックアップを行うためのバッテリーである。従って、バッファメモリ 3

23は、外部からの電力供給が遮断された場合であっても、バッテリー324の電力により、継続してデータを記憶しておくことができる。

[0042] FMアダプタ320は、FMコントローラ320a、FMプロセッサ320b、バスコントローラ320c、DMA (Direct Memory Access) コントローラ (DMAC) 320d、およびRAMコントローラ320eを含む。FMアダプタ320は、例えば、ASICなどの集積回路である。なお、本実施例では、FMアダプタ320は、一つの集積回路に各構成の回路群を内蔵しているが、これらの回路を複数の集積回路に分割して実装してもよい。また、ある回路 (例えばDMAC 320d) の機能を別の回路 (例えばFMプロセッサ320b) で代替してもよい。

[0043] 図5は、実施例1に係るストレージコントローラのRAMの構成図である。

[0044] RAM34は、例えばDRAMやSRAM等のランダムアクセスメモリである。RAM34は、CPU33が実行するストレージ制御プログラム340、キャッシュ制御情報341、アクセスモニタテーブル342、およびジョブ制御テーブル344を格納する。また、RAM34には、データをキャッシュして管理するための複数のキャッシュセグメント343が格納される。このキャッシュセグメント343に、HDD40又はSSD41に格納されるデータ、又は、HDD40又はSSD41から読み出されたデータをキャッシュすることができる。

[0045] ストレージ制御プログラム340は、キャッシュ制御プログラムの一例であり、キャッシュに関わる各種制御処理を実行する。なお、処理の詳細については後述する。キャッシュ制御情報341は、キャッシュディレトリ100 (図7参照) と、クリーンキュー (図9参照) と、ダーティキュー (図9参照) と、FMフリーキュー200 (図7参照) と、RAMフリーキュー300 (図7参照) が含まれる。キャッシュ制御情報341に関するデータ構造については後述する。

[0046] RAM34の実装方法としては、例えば基板上に複数のRAMのメモリチ

ップを搭載したD I M Mのようなメモリモジュールを構成しておき、このメモリモジュールをストレージコントローラ30の主基板上のメモリスロットに接続するようにしてもよい。なお、ストレージコントローラ30の主基板と別の基板上にR A Mを搭載する構成とすることにより、ストレージコントローラ30の主基板とは独立に保守交換やR A M容量増設を行えるようにすることができる。また、停電など不慮の障害が発生した場合に、R A M 3 4上の記憶内容が消失することを避けるために、バッテリーを設けて、停電時等にもR A M 3 4上の記憶内容を維持できるような構成にしてもよい。

[0047] 図6は、実施例1に係るアクセスモニタテーブルの構成図である。

[0048] アクセスモニタテーブル342は、ストレージシステム20における論理ユニット（論理ボリューム）内の各部分領域に対するアクセスの頻度や、データ読み出し・書き込みの速度を集計するための情報を記憶するとともに、集計結果を記憶するためのテーブルである。アクセスモニタテーブル342は、論理ユニット内の各部分領域のそれぞれに対して、例えば、リード速度342 a、ライト速度342 b、リード頻度342 c、ライト頻度342 d、リード量(バイト数)カウンタ342 e、ライト量カウンタ342 f、リードコマンドカウンタ342 g、ライトコマンドカウンタ342 h、およびモニタ開始時刻342 iを記憶する。部分領域（つまり1つのアクセスモニタテーブル342でアクセス頻度やアクセス速度を集計する単位）のサイズとしては、様々なサイズ（ただし論理ボリュームのサイズよりは小さいサイズ）を選択可能であるが、後述する通り、本発明におけるストレージシステムでは、アクセスモニタテーブル342の情報に基づいて、キャッシュセグメントの割り当てを行うので、部分領域のサイズを、キャッシュセグメントと同じサイズとする、あるいはキャッシュセグメントの整数倍とすることが、好ましい。

[0049] リード速度342 aは、論理ユニット内の部分領域に対するリード速度（例えば、単位は、MB／S e c）である。ライト速度342 bは、論理ユニット内の部分領域に対するライト速度（例えば、単位は、MB／S e c）で

ある。リード頻度 3 4 2 c は、論理ユニット内の部分領域に対するリードの発生頻度である。ライト頻度 3 4 2 d は、論理ユニット内の部分領域に対するライトの発生頻度である。リード量カウンタ 3 4 2 e は、論理ユニット内の部分領域においてリードされたデータのデータ量のカウンタである。ライト量カウンタ 3 4 2 f は、論理ユニット内の部分領域においてライトされたデータのデータ量のカウンタである。リードコマンドカウンタ 3 4 2 g は、論理ユニット内の部分領域においてリードを行ったコマンドの数のカウンタである。ライトコマンドカウンタ 3 4 2 h は、論理ユニット内の部分領域においてライトを行ったコマンドの数のカウンタである。モニタ開始時刻 3 4 2 i は、論理ユニット内の部分領域に対するモニタを開始した時刻である。リード量カウンタ 3 4 2 e、ライト量カウンタ 3 4 2 f、リードコマンドカウンタ 3 4 2 g、およびライトコマンドカウンタ 3 4 2 h が集計用のカウンタであり、リード速度 3 4 2 a、ライト速度 3 4 2 b、リード頻度 3 4 2 c、およびライト頻度 3 4 2 d が、集計結果である。ストレージシステム 20 における論理ユニット内の各部分領域に対するアクセスの頻度（リード頻度、ライト頻度）や、データ読み出し・書き込みの速度（リード速度、ライト速度）を集計するアクセスモニタ集計処理（図 17 参照）については後述する。

[0050] 続いて、図 7 以降では、実施例 1 に係るキャッシュ管理データ構造について説明するが、その前に、ボリューム（論理ボリューム）とキャッシュ管理データとの関係について、概要を説明する。図 34 は、本発明の実施例における、論理ボリューム、スロット、キャッシュセグメントの関係を示す概念図である。ホスト計算機 10 が論理ボリューム 1000 にアクセスする際、最小アクセス単位はブロック（たとえば 512 バイト）であり、論理ボリュームの各ブロックには、論理ブロックアドレス（LBA。あるいは本明細書では、論理アドレスと呼ぶこともある）が付与されている（図 34 の要素 1010 は論理ブロックアドレスを表している）。また、ストレージシステム 20 では、論理ボリューム上の記憶領域へのアクセス時等に排他制御を行う

が、排他制御の単位として、スロット 1 1 0 0 という概念が定義されている。スロット 1 1 0 0 のサイズは、本発明の実施例では、2 5 6 K B（つまり 5 1 2 ブロック）としているが、その他のサイズを採用しても良い。各スロット 1 1 0 0 には一意な識別番号が付与されており、それをスロット I D と呼ぶ。なお、図 3 4 において、要素 1 1 1 0 がスロット I D であり、論理ブロックアドレス 1 0 1 0 とスロット I D 1 1 0 0 との関係は図 3 4 に示されたとおりである。例えばホスト計算機 1 0 から受信した I / O コマンドで指定されている論理ブロックアドレスをスロット I D に変換する時には、指定された論理ブロックアドレスを 5 1 2 で除算して得られた値がスロット I D になる。またこの除算を行った時に剰余が 0 の場合には、I / O コマンドで指定されている論理ブロックアドレスは、（スロット I D で特定される）スロットの先頭位置であることが分かり、剰余が 0 でない値（仮に、この値を R とする）の場合、この剰余 R は、論理ブロックアドレスで特定されるブロックが、（スロット I D で特定される）スロットの先頭ブロックから R 番目の位置に存在するブロックであることを表す情報となる（以下、この情報 R のことを、スロット内相対アドレスと呼ぶ）。

[0051] また論理ボリューム上のデータをキャッシュする場合、R A M 3 4 または F M チップ 3 2 1 上の記憶領域をキャッシュ領域として確保するが、キャッシュセグメント（またはセグメント）（図 3 4 中の要素 1 2 0 1、1 2 0 2、1 2 0 3、1 2 0 4 がキャッシュセグメントである。以下、キャッシュセグメント 1 2 0 1、1 2 0 2、1 2 0 3、1 2 0 4 を総称する場合、「キャッシュセグメント 1 2 0 0」と表記する）という領域単位でキャッシュ領域を確保する。本発明の実施例においては、キャッシュセグメント 1 2 0 0 のサイズは 6 4 K B で、各スロットに対して 4 つのキャッシュセグメント 1 2 0 1、1 2 0 2、1 2 0 3、1 2 0 4 が対応付けられる。図 3 4 は、R A M 3 4 あるいは F M チップ 3 2 1 上の領域（キャッシュセグメント 1 2 0 0）がスロット 1 1 0 0 に対応付けられる概念を表している。ストレージシステム 2 0 がこの対応付けを管理する方法の詳細は後述するが、ストレージシステム

20は、スロットを管理する情報としてスロット制御テーブル110（詳細は図7を説明する際に説明する。スロット制御テーブル110は、スロット1100ごとに1つ存在する）を有しており、スロット制御テーブル110内に、当該スロットに対応付けられているキャッシュセグメント1200の情報（正確には、キャッシュセグメント1200を管理するための情報へのポインタ）が格納されている。ストレージシステム20は、このスロット制御テーブル110を作成、管理することによって、スロット1100とキャッシュセグメント1200の対応付けを管理する。なお、キャッシュセグメント1200のサイズを、64KB以外のサイズにすることも可能であり、また1つのスロット1100に対応付けられるキャッシュセグメント1200の数を、4つ以外の値にすることも可能である。

[0052] ホスト計算機10が論理ボリューム1000上の領域にアクセス（リードまたはライト等）する際の、キャッシュ領域の管理に関連する処理の概要は以下の通りである。ホスト計算機10はストレージシステム20に、論理ユニット番号（論理ユニット／論理ボリュームを特定する番号で、一般的にはLUN[Logical Unit Number]と略記される）と論理ブロックアドレス1010を指定したI/Oコマンドを発行する。ストレージシステム20は、受信したI/Oコマンドに含まれる論理ブロックアドレスを、スロットID1110とスロット内相対アドレスの組に変換し、当該変換で得られたスロットID1110で特定されるスロット制御テーブル110を参照する。そしてスロット制御テーブル110の情報に基づき、I/Oコマンドで指定された論理ボリューム上の領域（論理ブロックアドレスで特定される領域）に対してキャッシュセグメント1200が確保されているか否かを判定し、キャッシュセグメント1200が確保されていない場合は、新規にキャッシュセグメント1200を確保する処理を行う。

[0053] 続いてキャッシュ管理データ構造についての説明を行う。図7は、実施例1に係るキャッシュ管理データ構造の概念図である。

[0054] キャッシュ管理データ構造は、キャッシュディレクトリ100と、FMフ

リーキュー２００と、ＲＡＭフリーキュー３００と、ダーティキューおよびクリーンキュー（図９参照）とを含む。本実施例では、ＲＡＭ３４と、ＦＭチップ３２１とにキャッシュセグメント（３４３、３２５）が管理される。各キャッシュセグメントは、セグメント制御テーブル１２０（ＳＧＣＴ：Segment Control Table）で管理される。ＳＧＣＴ１２０は、ＲＡＭ３４および全てのＦＭチップ３２１で管理されている全てのキャッシュセグメントのそれぞれと一対一に対応して管理される。

[0055] キャッシュディレクトリ１００は、キャッシュ対象データの論理アドレス（キャッシュセグメントに格納されたデータの格納先である、論理ボリュームの論理ブロックアドレス）とメモリ（ＲＡＭ３４およびＦＭチップ３２１）上の物理アドレスとの対応関係を管理するデータ構造である。キャッシュディレクトリ１００は、例えば、キャッシュ対象データの論理アドレス（又はスロットＩＤのような、論理アドレスから導かれる情報）をキーとするハッシュテーブルであり、ＳＧＣＴ１２０を示すためのポインタをエントリとして持っている。ＳＧＣＴ１２０は、そのＳＧＣＴ１２０に対応するキャッシュセグメント（３２５、３４３）へのポインタを管理している。したがって、キャッシュディレクトリ１００によると、キャッシュ対象データの論理アドレスに基づいて、当該論理アドレスに対応するデータがキャッシュされているキャッシュセグメントを特定することができる。なお、ＳＧＣＴ１２０の詳細な構成については、後述する。本実施例では、キャッシュディレクトリ１００は、ＲＡＭ３４のキャッシュセグメント３４３と、すべてのＦＭチップ３２１のキャッシュセグメント３２５とをまとめて管理している。このため、当該キャッシュディレクトリ１００を参照することにより、ＲＡＭ３４およびＦＭチップ３２１におけるキャッシュのヒット判定を容易に行うことができる。

[0056] ＦＭフリーキュー２００は、ＦＭチップ３２１のフリーセグメント、すなわちいずれのデータも格納していないキャッシュセグメント３２５を管理する制御情報である。ＦＭフリーキュー２００は、例えばＦＭチップ３２１の

フリーセグメントに対応するSGCT120をエントリに持つ双方向のリンクリストとして構成される。なお、フリーセグメントを管理する制御情報のデータ構造は、必ずしもキューである必要はなく、スタックなどを用いてもよい。

[0057] RAMフリーキュー300は、RAM34のフリーセグメントを管理する制御情報である。RAMフリーキュー300は、例えばRAM34のフリーセグメントに対応するSGCT120をエントリに持つ双方向のリンクリストとして構成される。なお、フリーセグメントを管理する制御情報のデータ構造は、必ずしもキューである必要はなく、スタックなどを用いてもよい。

[0058] SGCT120は、そのSGCT120に対応するキャッシュセグメントの状態および種類によって、キャッシュディレクトリ100、FMフリーキュー200、又はRAMフリーキュー300のいずれかに接続された状態をとる。具体的には、FMチップ321のキャッシュセグメント325に対応するSGCT120は、当該キャッシュセグメント325が未使用時には、FMフリーキュー200に接続され、当該キャッシュセグメント325がデータ格納用に割り当てられると、キャッシュディレクトリ100に接続される。一方、RAM34のキャッシュセグメント343に対応するSGCT120は、当該キャッシュセグメント343が未使用時には、RAMフリーキュー300に接続され、当該キャッシュセグメント343がデータ格納用に割り当てられると、キャッシュディレクトリ100に接続される。

[0059] 図8は、実施例1に係るキャッシュ管理データ構造の一部のデータ構造を示す図である。

[0060] キャッシュディレクトリ100は、例えばスロットIDをキーとするハッシュテーブルである。キャッシュディレクトリ100のエントリ100a（ディレクトリエントリ）は、スロットIDに対応するスロット制御テーブル110（SLCT: Slot Control Table）を示すディレクトリエントリポインタを格納する。ここで、スロットは、排他制御を行うデータの単位（ロック単位）である。例えば、1つのスロットは、複数のキャッシュセグメン

トを含むことができる。なお、スロットの内の一部しかデータが格納されていない場合には、キャッシュセグメントが1つしか含まれていないこともある。

[0061] S L C T 1 1 0 は、ディレクトリエントリポインタ 1 1 0 a と、前方ポインタ 1 1 0 b と、後方ポインタ 1 1 0 c と、スロット I D 1 1 0 d と、スロットステータス 1 1 0 e と、S G C T ポインタ 1 1 0 f とを含む。ディレクトリエントリポインタ 1 1 0 a は、ハッシュテーブルの次のエントリに対応する S L C T 1 1 0 を指すディレクトリエントリポインタである。前方ポインタ 1 1 0 b は、クリーンキュー又はダーティキューにおける前の順番の S L C T 1 1 0 を示すポインタである。後方ポインタ 1 1 0 c は、クリーンキュー又はダーティキューにおける後の順番の S L C T 1 1 0 を示すポインタである。スロット I D 1 1 0 d は、S L C T 1 1 0 に対応するスロットの識別情報である。スロットステータス 1 1 0 e は、スロットの状態を示す情報である。スロットの状態としては、例えば、当該スロットがロックされていることを示す「ロック中」等がある。S G C T ポインタ 1 1 0 f は、当該スロットに含まれるキャッシュセグメントに対応する S G C T 1 2 0 を指すポインタである。当該スロットにキャッシュセグメントが割り当てられていないときは、S G C T ポインタ 1 1 0 f は、ポインタ（アドレス）が無効であることを表す値（例えば N U L L ）となっている。またスロットに含まれるキャッシュセグメントが複数ある場合には、各 S G C T 1 2 0 はリンクリストとして管理され、S G C T ポインタ 1 1 0 f は、リンクリストにおける先頭のキャッシュセグメントに対応する S G C T 1 2 0 を指すポインタである。

[0062] S G C T 1 2 0 は、S G C T ポインタ 1 2 0 a と、セグメント I D 1 2 0 b と、メモリ種類 1 2 0 c と、セグメントアドレス 1 2 0 d と、ステージングビットマップ 1 2 0 e と、ダーティビットマップ 1 2 0 f とを含む。

[0063] S G C T ポインタ 1 2 0 a は、同じスロットに含まれる次のキャッシュセグメントに対応する S G C T 1 2 0 を指すポインタである。セグメント I D

120bは、キャッシュセグメントの識別情報で、スロット内の何番目に位置づけられるキャッシュセグメントであるかを表す情報である。本実施例では、1スロットに最大で4つのキャッシュセグメントが割り当てられるので、各キャッシュセグメントのセグメントID120bには、0、1、2、3のいずれかの値が格納される（スロットの先頭に位置するキャッシュセグメントのセグメントID120bは0となり、以下順に1、2、3のセグメントID120bが付与される。たとえば図34におけるキャッシュセグメント1201～1204を例にとると、スロット1100の先頭に対応付けられているキャッシュセグメント1201のセグメントID120bは0で、以下キャッシュセグメント1202、1203、1204のセグメントID120bはそれぞれ、1、2、3となる）。メモリ種類120cは、このSGCT120に対応するキャッシュセグメントが格納されているキャッシュメモリの種類である。キャッシュメモリの種類としては、FMと、RAMとのいずれかがある。セグメントアドレス120dは、キャッシュセグメントのアドレスである。ステージングビットマップ120eは、キャッシュセグメントの中でクリーンデータ、つまりドライブ（40、41）のデータと一致しているデータがキャッシュされている領域を示すビットマップである。ステージングビットマップ120eでは、各ビットがキャッシュセグメントにおける各領域に対応し、有効なデータ（ドライブと同じデータ）がキャッシュされている領域に対応するビットはON（1）に設定され、有効なデータがキャッシュされていない領域に対応するビットはOFF（0）に設定される。ダーティビットマップ120fは、キャッシュセグメントの中でダーティデータ、つまりドライブのデータと不一致なデータ（まだドライブに反映されていないデータ）がキャッシュされている領域を示すビットマップである。ダーティビットマップ120fでは、各ビットがキャッシュセグメントにおける各領域に対応し、ダーティデータがキャッシュされている領域に対応するビットはON（1）に設定され、ダーティデータがキャッシュされていない領域に対応するビットはOFF（0）に設定される。

[0064] 図9は、実施例1に係るダーティキューおよびクリーンキューのデータ構造を示す図である。

[0065] ダーティキューおよびクリーンキューは、キャッシュデータ管理構造の一部である。ダーティキューは、ダーティデータを含むスロットに対応するSELECT110を接続するキューである。クリーンキューは、クリーンデータのみを含むスロットに対応するSELECT110を接続するキューである。ダーティキューおよびクリーンキューは、キャッシュリプレイスやデステージのスケジューリング等に用いられ、それぞれのキャッシュリプレイスやデステージのスケジューリング方式等によってさまざまな構造を取りうる。本実施例では、キャッシュリプレイスおよびデステージのスケジューリングに用いるアルゴリズムをLRU (Least Recently Used) として説明する。なお、ダーティキューおよびクリーンキューは、接続するSELECT110が違うだけで、キューの基本的な構成は同様であるので、ここでは、ダーティキューを例に挙げて説明する。ダーティキューは、双方向のリンクリストとして構成されている。すなわち、ダーティキューは、MRU (Most Recently Used) ターミナル150の前方ポインタに、最近使用したダーティデータを含むスロット（最終使用時刻の新しいスロット）に対応するSELECT110を接続し、以降、SELECT110の前方ポインタ110bに、次の順番のスロット（次に最近使用したダーティデータを含むスロット）のSELECT110を順次接続し、最後の順番のSELECT110の前方ポインタ110bにLRUターミナル160を接続する一方、LRUターミナル160の後方ポインタに最後の順番のSELECT110を接続し、以降、後ろの順番のSELECT110の後方ポインタ110cに対して、その前の順番のスロットのSELECT110を順次接続し、最初の順番のSELECT110をMRUターミナル150に接続する。ダーティキューでは、MRUターミナル150側から、最終使用時刻の新しい順にSELECT110が並ぶこととなる。

[0066] 図10は、実施例1に係るFMフリーキューおよびRAMフリーキューのデータ構造を示す図である。

[0067] FMフリーキュー200は、FMチップ321に格納されている空きキャッシュセグメント325を管理するためのキューであり、RAMフリーキュー300は、RAM34の空きキャッシュセグメント343を管理するためのキューであり、それぞれ空きキャッシュセグメントのSGCT120をポインタで接続したリンクリストである。FMフリーキュー200と、RAMフリーキュー300とは、管理するSGCT120が異なるだけであり、構成は同一である。FMフリーキュー200（RAMフリーキュー300）のフリーキューポインタ201（301）は、キューの先頭のSGCT120を指す。SGCT120のSGCTポインタ120aは、次の空きキャッシュセグメントのSGCT120を指す。

[0068] 次に、実施例1に係る情報システムにおける処理動作を説明する。

[0069] 図11は、実施例1に係るリードコマンド処理のフローチャートである。

[0070] リードコマンド処理は、ホスト計算機10からストレージコントローラ30がリードコマンドを受信した場合に実行される。

[0071] まず、リードコマンドを受信したストレージコントローラ30のCPU33は、リードコマンドで指定されている、論理ボリューム上のリード対象ブロックの論理ブロックアドレス（以下、「リード対象アドレス」と呼ぶ）に対応するキャッシュセグメントが割り当て済みであるか否かを判定する（ステップS1）。具体的には、先に説明したとおり、論理ブロックアドレスをスロットIDとスロット内相対アドレスの組に変換し、当該変換により得られたスロットID110dを有するSLCT110内のSGCTポインタ110fを参照する。SGCTポインタ110fが無効（たとえばNULL）値である場合には、キャッシュセグメントが未割り当てであることが判明する。SGCTポインタ110fに有効な値が含まれている場合、少なくともキャッシュセグメントが1つ割り当てられていることが判明するので、SGCTポインタ110fをたどって、スロット内相対アドレスで特定されるスロット内の位置にキャッシュセグメントが割り当てられているかを確認する。具体的には、「スロット内相対アドレス÷128」で得られた結果（整数

値) と同一のセグメント I D 1 2 0 b を有する S G C T 1 2 0 があるかを確認することで、キャッシュセグメントが割り当てられているかを確認することができる (スロット内相対アドレス ÷ 1 2 8 の計算を行うことで、0 ~ 3 のいずれかの整数値が得られるので、スロット内相対アドレスが、0 ~ 3 のいずれのセグメント I D が付与されたキャッシュセグメントに対応するアドレスであるかを知ることができる)。この結果、キャッシュセグメントが割り当て済みであれば (ステップ S 1 : Y E S)、C P U 3 3 は、ステップ S 3 へ処理を進める一方、キャッシュセグメントが未割り当てであれば (ステップ S 1 : N O)、キャッシュ割り当て処理 (図 1 4 参照) を実行し (ステップ S 2)、ステップ S 3 へ処理を進める。ステップ S 2 のキャッシュ割り当て処理の詳細は後述するが、キャッシュ割り当て処理では、キャッシュされるデータのアクセス特性に応じて、F M チップ 3 2 1 のキャッシュセグメント、または R A M 3 4 のキャッシュセグメントを割り当てる処理が行われる。

[0072] ステップ S 3 では、C P U 3 3 は、リード対象アドレスに対応するキャッシュセグメントを含むスロットをロックする。具体的には、C P U 3 3 は、このキャッシュセグメントを含むスロットの S L C T 1 1 0 のスロットステータス 1 1 0 e の「ロック中」を表すビットを O N にすることで、当該スロットがロックされていることを表す。

[0073] 続いて C P U 3 3 は、リード対象のデータがキャッシュセグメントに格納されているか否か、すなわち、キャッシュヒットか否かを判定する (ステップ S 4)。具体的には、C P U 3 3 は、リード対象のキャッシュセグメントに対応する S G C T 1 2 0 のステージングビットマップ 1 2 0 e およびダーティビットマップ 1 2 0 f を調べ、リード対象の全ブロックに関して、当該ブロックに対応するステージングビットマップ 1 2 0 e のビットもしくはダーティビットマップ 1 2 0 f のビットのいずれかが O N であれば、キャッシュヒットであると判断する。一方、C P U 3 3 は、リード対象の範囲内に、ダーティビットマップ 1 2 0 f とステージングビットマップ 1 2 0 e との対

応するビットがいずれもOFFであるブロックが一つでもある場合は、キャッシュミスであると判断する。

[0074] この結果、キャッシュヒットの場合（ステップS4：YES）は、CPU33は、処理をステップS6へ進める一方、キャッシュミスの場合（ステップS4：NO）は、ステージング処理（図12参照）を実行し（ステップS5）、処理をステップS6に進める。ステージング処理では、ドライブ（HDD40、またはSSD41）からキャッシュセグメント（325又は343）へデータが読み込まれる。ステージング処理が完了すると、リード対象のデータがキャッシュセグメント（325又は343）に格納された状態になる。

[0075] ステップS6では、CPU33は、キャッシュセグメントに格納されたデータをホスト計算機10に送信するデータ送信処理（図13参照）を実行する。

[0076] 続いて、CPU33は、コマンドの完了ステータスをホスト計算機10に送信する（ステップS7）。すなわち、CPU33は、コマンドの処理中にエラーが発生してリード処理が正常に完了しなかった場合はエラーステータス（例えばCHECK CONDITION）を返す一方、リード処理が正常に完了した場合は正常ステータス（GOOD）を返す。

[0077] 続いて、CPU33は、ロックしているスロットを解放（アンロック）し（ステップS8）、アクセスモニタテーブル342を更新し（ステップS9）、リードコマンド処理を終了する。アクセスモニタテーブル342の更新とは、例えば、リード量カウンタ342eに今回のリードコマンドで読み出したデータ量を加算し、リードコマンドカウンタ342gをインクリメントする。

[0078] 図12は、実施例1に係るステージング処理のフローチャートである。

[0079] ステージング処理は、図11のリードコマンド処理のステップS5の処理に対応する。

[0080] まず、CPU33は、リード対象アドレスに対応するキャッシュセグメン

トのキャッシュメモリの種類を調べ、キャッシュセグメントがRAM 34上
のキャッシュセグメント（RAMセグメント）343であるか否かを判定す
る（ステップS11）。ここで、キャッシュセグメントの基になっているキ
ャッシュメモリの種類は、対応するSGCT120のメモリ種類120cを
参照することにより特定することができる。

[0081] この結果、キャッシュセグメントがRAMセグメント343であれば（ス
テップS11：YES）、CPU33は、処理をステップS12に進める一
方、キャッシュセグメントがRAMセグメント343でなければ（ステップ
S11：NO）、処理をステップS13に進める。

[0082] ステップS12では、CPU33は、リード対象（ステージング対象）の
データをドライブ（HDD40またはSSD41）から読み出して、RAM
セグメント343に格納し、ステージング処理を終了する。

[0083] ステップS13以降の処理では、キャッシュセグメントがRAMセグメン
ト343でない、すなわちFMチップ321上のキャッシュセグメント（F
Mセグメント）325であるので、ドライブから読み上げたデータを直接F
Mチップ321に書くのではなく、一旦、FMボード32のバッファメモリ
323に格納した後、バッファメモリ323からFMチップ321に書き込
むようにする。これはFMチップ321の書き込み速度が遅いので、ドライ
ブから読み上げたデータを直接FMチップ321に書いてしまうと、その速
度に引きずられてストレージコントローラ30のBE I/F35の動作が
遅くなり、ストレージシステム20のスループット性能を低くしてしまうこ
とを防ぐためである。なお、本実施例では、BE I/F35が、CPU3
3の指示を受けて、ドライブからFMボード32のバッファメモリ323に
対してデータを格納するようにしている。したがって、CPU33は、BE
I/F35に対して指示を行った後は、他の処理を実行することができる
。また、BE I/F35は、ドライブからFMボード32のバッファメモリ
323に対してデータを格納した後には、この処理から解放されて、他の
処理を実行することができるようになる。

- [0084] まず、ステップS 1 3では、CPU 3 3は、バッファメモリ 3 2 3にドライブから読み上げたデータを格納するための領域（バッファ）の確保を行う。すなわち、CPU 3 3は、バッファメモリ 3 2 3のうちステージング対象のデータを格納するのに十分な領域をバッファに割り当てる。
- [0085] 次に、CPU 3 3は、ステージング対象のデータをドライブから読み出してバッファに格納する（ステップS 1 4）。本実施例では、BE I / F 3 5が、CPU 3 3の指示を受けて、ドライブからFMボード 3 2のバッファメモリ 3 2 3のバッファに対してデータを格納している。
- [0086] そして、CPU 3 3は、FMプロセッサ 3 2 0 bに対し、バッファメモリ 3 2 3のバッファ上のデータをFMチップ 3 2 1に格納するよう要求する（ステップS 1 5）。この要求に対してFMプロセッサ 3 2 0 bは、FMデータ書き込み処理（図 2 3 参照）を実行する。FMプロセッサ 3 2 0 bは、FMデータ書き込み処理を終了すると、その要求に対する完了応答をCPU 3 3に返す。
- [0087] 続いて、CPU 3 3は、その要求に対する完了応答をFMプロセッサ 3 2 0 bから受け取り（ステップS 1 6）、バッファメモリ 3 2 3のバッファを解放し（ステップS 1 7）、ステージング処理を終了する。
- [0088] 図 1 3 は、実施例 1 に係るデータ送信処理のフローチャートである。
- [0089] データ送信処理は、図 1 1 に示すリードコマンド処理のステップS 6 の処理に対応する。
- [0090] データ送信処理では、FMセグメント 3 2 5からデータを送信する場合に、一旦バッファメモリ 3 2 3にデータを格納して、バッファメモリ 3 2 3からホスト計算機 1 0へデータを転送する。これは、FMチップ 3 2 1の読み出し速度が遅いので、FMチップ 3 2 1から直接転送すると、その速度に引きずられてストレージコントローラ 3 0のFE I / F 3 1の動作が遅くなり、ストレージシステム 2 0のスループット性能を低くしてしまうことを防ぐためである。
- [0091] まず、CPU 3 3は、リード対象アドレスに対応するキャッシュセグメン

トの基になっているキャッシュメモリの種類を調べ、キャッシュセグメントがRAMセグメント343であるか否かを判定する（ステップS21）。ここで、キャッシュセグメントの基になっているキャッシュメモリの種類は、対応するSGCT120のメモリ種類120cを参照することにより特定することができる。

[0092] この結果、キャッシュセグメントがRAMセグメント343であれば（ステップS21：YES）、CPU33は、処理をステップS22に進める一方、キャッシュセグメントがRAMセグメント343でなければ（ステップS21：NO）、処理をステップS23に進める。

[0093] ステップS22では、CPU33は、リード対象（送信対象）のデータをRAMセグメント343からホスト計算機10へ転送し、データ送信処理を終了する。

[0094] ステップS23では、CPU33は、バッファメモリ323に、FMチップ321から読み出した送信対象のデータを格納するための領域（バッファ）の確保を行う。すなわち、CPU33は、バッファメモリ323のうち送信対象のデータを格納するのに十分な領域をバッファに割り当てる。

[0095] 次に、CPU33は、FMプロセッサ320bに対し、FMチップ321上のデータをバッファメモリ323に読み出すよう要求する（ステップS24）。この要求に対してFMプロセッサ320bは、FMデータ読み出し処理（図22参照）を実行する。FMデータ読み出し処理により、バッファメモリ323には、送信対象のデータが格納される。FMプロセッサ320bは、FMデータ読み出し処理を終了すると、その要求に対する完了応答をCPU33に返す。

[0096] 続いて、CPU33は、その要求に対する完了応答をFMプロセッサ320bから受け取り（ステップS25）、送信対象のデータをバッファメモリ323からホスト計算機10に送信する（ステップS26）。本実施例では、FEI/F31が、CPU33の指示（例えば、読み出すデータのバッファメモリ323上のアドレス等）を受けて、バッファメモリ323のバッ

ファからホスト計算機 10 に対して送信対象のデータを送信している。その後、CPU 33 は、バッファメモリ 323 のバッファを解放し（ステップ S27）、データ送信処理を終了する。

[0097] 図 14 は、実施例 1 に係るキャッシュ割り当て処理のフローチャートである。

[0098] キャッシュ割り当て処理は、図 11 に示すリードコマンド処理のステップ S2 の処理、および後述する図 19 に示すライトコマンド処理のステップ S72 の処理に対応する。

[0099] キャッシュ割り当て処理では、CPU 33 がキャッシュするデータに対して、当該データに対するアクセス特性に応じて、FMチップ 321 のキャッシュセグメント、または RAM 34 のキャッシュセグメントを割り当てる。

[0100] ここで、まず、割り当てるキャッシュセグメントのメモリ種類、すなわち、FMチップ 321 または RAM 34 を選択する際の判断基準について説明する。ここで、FMチップ 321 は、（１）アクセス性能が RAM 34 に比べて低い、（２）書き換え回数に上限がある、という特性を有しているので、本実施例では、CPU 33 は、データの特性が、比較的 RAM の特性に適合しているデータ（高性能を要求される、キャッシュセグメントの更新頻度が高くなる）は、RAM 34 を用いたキャッシュセグメントを選択し、フラッシュメモリの特性に適合するデータ（それほど高性能を要求されない、キャッシュセグメントの更新頻度が低い）は、FMチップ 321 を用いたキャッシュセグメントを選択する制御を行う。具体的には、次のような基準で割り当てるキャッシュセグメントのメモリ種類を選択する。

（a）アクセス頻度（リード頻度／ライト頻度）が高いデータや、高スループットが求められるデータの場合は、CPU 33 は、RAM 34 を優先的に選択する。特に、アクセス頻度が高いデータを FMチップ 321 のキャッシュセグメントに格納するようにすると、FMチップ 321 の更新頻度が高くなる。更新頻度が高い場合、書き換えが多発し FMチップ 321 の寿命を短くしてしまうので、RAM 34 を優先的に選択した方がよい。これにより、

F Mチップ321の寿命の短縮を適切に抑制することができる。また、高スループットが求められるデータとしては、例えば、インメモリデータベースで利用するために読み込む大量データが該当する。この用途のデータは、一般的に、転送長が長いデータやシーケンシャルアクセスのデータであることが多いため、転送長が長いと判断されるデータについてはR A M 3 4を優先的に選択する。これにより、高スループットを実現することができる。

(b) F Mチップ321にキャッシュするとキャッシュヒットによる性能面の効果があり得られないデータの場合は、C P U 3 3は、R A M 3 4を優先的に選択する。キャッシュヒットによる性能面の効果があり得られないデータとしては、例えばS S D 4 1に格納されるデータがある。これによると、キャッシュヒットによる効果を適切に得られるようにすることができる。

(c) アクセス単位が小さいデータがキャッシュ対象であれば、C P U 3 3は、R A M 3 4を優先的に選択する。これはF Mチップ321における読み書き単位（ページ）の大きさがR A M 3 4の最小アクセス単位に比べて大きく（例えば8 K B）、小さい単位のデータの参照・更新の効率が悪いためである。例えば、制御情報などのメタデータは、通常16 B程度のサイズであり、F Mチップ321の読み書き単位のサイズと比較すると小さいため、R A M 3 4にキャッシュした方がよい。

(d) すぐにキャッシュ上から破棄されるデータがキャッシュ対象の場合は、C P U 3 3は、R A M 3 4を優先的に選択する。その理由は、破棄に伴うF Mチップ321の消去がすぐに発生することと、すぐ破棄するのであればR A M 3 4に置いても容量の消費が一時的であり影響が小さいことが挙げられる。どのようなデータを優先的に破棄するかについては、ストレージシステムのポリシーとして設定されている。例えばデータコピーのために割り当てられる一時的なキャッシュセグメントに格納されるデータは、コピー処理が完了するとキャッシュ上から破棄される。他の例としては、シーケンシャルリードが行われるデータや、シーケンシャルライトが行われるデータがあ

る。シーケンシャルリードが行われるデータについては、先頭から順番に読んでいって、読み終わったら同じ部分のデータがすぐに読まれることは基本的にはない。また、シーケンシャルライトが行われるデータは、例えば、当該データがRAIDで格納される場合には、必要なパリティが揃った時点で、データがデステージされ、その後キャッシュから破棄される。

(e) 上記(a) - (d) 以外の条件に適合するデータがキャッシュ対象の場合は、CPU 33は、FMチップ321を優先的に選択する。

[0101] 次に、以上の基準に基づいてキャッシュ割り当てを行うキャッシュ割り当て処理を、図14を参照して説明する。

[0102] まず、CPU 33は、アクセス対象（リード対象又はライト対象）のデータが高速にアクセスされるデータか否かを判定する（ステップS31）。具体的には、CPU 33は、例えば、予め求められたアクセス速度の閾値と、アクセスモニタテーブル342に記録されたアクセス対象のデータの格納される領域のリード速度342aやライト速度342bとを比較して、アクセス対象のデータが高速にアクセスされるデータか否かを判定する。リード速度342aやライト速度342bの集計方法は後述するが、データアクセス頻度が高い場合、あるいは転送長が長いデータのアクセスの場合、本実施例で集計されるリード速度342aやライト速度342bは高くなるため、データのリード速度342aやライト速度342bを閾値と比較することで、アクセス頻度が高いデータであるか、または転送長が長いデータであるかを判定することができる。あるいは別の実施形態として、アクセス頻度（アクセスモニタテーブル342に記録された、アクセス対象のデータの格納される領域のリード頻度342cやライト頻度342d）を閾値と比較して判定してもよい。ステップS31の判定の結果が、真であれば（ステップS31：YES）、CPU 33は処理をステップS37に進める一方、偽であれば（ステップS31：NO）、処理をステップS32に進める。

[0103] ステップS32では、CPU 33は、アクセス対象のデータに対するアクセスパターンがシーケンシャルアクセスであるか否かを判定する。この判定

は、CPU 33が、処理対象のリードコマンドが、連続したアドレスを順にリードする一連のコマンドの一部であるか否かを判断することにより実現できる。具体的には、CPU 33は、例えば、前回のリードコマンドの対象アドレスに当該コマンドの転送長を加えたアドレスが、今回のリードコマンドの対象アドレスであるか否かを判断するによって、シーケンシャルアクセスであるか否かを判定する。この結果、シーケンシャルアクセスであると判定した場合（ステップS 32：YES）は、CPU 33は、処理をステップS 37に進める一方、偽であると判定した場合（ステップS 32：NO）は、CPU 33は、処理をステップS 33に進める。

[0104] ステップS 33では、CPU 33は、アクセス対象のデータが最終的にSSD 41に格納されるデータであるか否か、すなわち、アクセス対象データの最終ストレージデバイスがSSD 41であるか否かを判定する。ここで、アクセス対象データの最終ストレージデバイスがSSD 41であるか否かの判定は、例えば、予め記憶している論理ボリュームとデバイスとの対応関係を表す情報に基づいて、リードコマンドで指定された論理ボリュームに対応するデバイス種類を特定することにより実現できる。なお、シンプロビジョニングに従う論理ボリュームである場合には、論理ボリュームに割り当てられている実ページを提供するデバイスのデバイス種類を特定することにより、アクセス対象データの最終ストレージデバイスがSSD 41であるか否かを判定することができる。この結果、真であれば（ステップS 33：YES）、CPU 33は、処理をステップS 37に進める一方、偽であれば、処理をステップS 34に進める。

[0105] ステップS 34では、CPU 33は、アクセス対象のデータがメタデータであるか否かを判定する。ここでいうメタデータには、ストレージコントローラ30のRAM 34からドライブ（40、41）に退避・格納した又は退避・格納する制御情報を含む。ここで、アクセス対象のデータがメタデータであるか否かは、例えば、アクセス先が、その論理ボリュームにおける制御情報が格納される所定の領域であるか否かにより判定することができる。な

お、論理ボリュームにおける制御情報が格納される領域のアドレスについては、論理ボリュームを使用するホスト計算機 10 等から取得することができる。この結果、真であれば（ステップ 34：YES）、CPU 33 は、処理をステップ S 37 に進める一方、偽であれば、処理をステップ S 35 に進める。

[0106] ステップ S 35 では、CPU 33 は、アクセス対象のデータに対応するキャッシュセグメントが一時的なキャッシュセグメント（テンポラリセグメント）であるか否かを判定する。ここで、テンポラリセグメントとは、次のいずれかである。

（１）パリティ生成において旧データや旧パリティがキャッシュミスした場合に、当該旧データあるいは旧パリティを格納するために割り当てられるセグメント。

（２）ドライブ（例えば最終ストレージデバイス）のデータをコピーする処理のため一時的に割り当てられるセグメント。

（３）他ストレージ装置との間でデータを送受信する処理のため（例えば、リモートコピー処理のため）一時的に割り当てられるセグメント。

[0107] なお、CPU 33 は、データに対してキャッシュセグメントを割り当てる際に、ホスト計算機 10 から高スループットか否かを示す情報、又は、I/O の優先度を示す情報を受け付け、当該高スループットか否かを示す情報、又は、I/O の優先度を示す情報をキャッシュセグメントに対応付けて記憶しておき、その情報に基づいて、キャッシュセグメントがテンポラリセグメントであるか否かを判断するようにしてもよい。

[0108] この結果、真であれば（ステップ 35：YES）、CPU 33 は、処理をステップ S 37 に進める一方、偽であれば、処理をステップ S 36 に進める。

[0109] ステップ S 36 では、CPU 33 は、FM チップ 321 のキャッシュセグメント 325 を優先して割り当てる FM 優先セグメント割り当て処理（図 15 参照）を実行し、キャッシュ割り当て処理を終了する。

- [0110] ステップS 3 7では、CPU 3 3は、RAM 3 4のキャッシュセグメント 3 4 3を優先して割り当てるRAM優先セグメント割り当て処理（図 1 6 参照）を実行し、キャッシュ割り当て処理を終了する。
- [0111] キャッシュ割り当て処理が完了すると、アクセス対象のデータに対して、FMチップ 3 2 1またはRAM 3 4のいずれか一方のキャッシュセグメントが割り当てられることとなる。
- [0112] 図 1 5は、実施例 1に係るFM優先セグメント割り当て処理のフローチャートである。
- [0113] FM優先セグメント割り当て処理は、図 1 4に示すキャッシュ割り当て処理のステップS 3 6に対応する処理である。
- [0114] まず、CPU 3 3は、使用可能なFMセグメント 3 2 5があるか否かを判定する（ステップS 4 1）。使用可能なFMセグメント 3 2 5とは、フリーもしくは、クリーンかつロックされていないキャッシュセグメント 3 2 5のことである。なお、使用可能なFMセグメント 3 2 5があるか否かは、キャッシュ管理データ構造を参照することにより判定することができる。この判定結果が真の場合（ステップS 4 1：YES）は、CPU 3 3は、処理をステップ 4 2に進める一方、偽の場合（ステップS 4 1：NO）は、処理をステップS 4 3に進める。
- [0115] ステップS 4 2では、CPU 3 3は、FMセグメント割り当て処理を行う。ここで、クリーンなキャッシュセグメントを割り当てる場合は、CPU 3 3は、当該キャッシュセグメントをクリーンキューおよびキャッシュディレクトリ 1 0 0から切り離してフリーセグメントにしたうえでFMセグメント割り当て処理を行う。
- [0116] FMセグメント割り当て処理では、まず、CPU 3 3は、SGCT 1 2 0に確保したキャッシュセグメントに対応するセグメントID 1 2 0 b、メモリ種類 1 2 0 c（FM）を設定する。そして、CPU 3 3は、このキャッシュセグメントを含むスロットに対応するSLCT 1 1 0のSGCTポインタ 1 1 0 fに当該キャッシュセグメントのSGCT 1 2 0へのポインタを設定

する。もし、対応する S L C T 1 1 0 がキャッシュディレクトリ 1 0 0 に接続されていない場合は、C P U 3 3 は、まず S L C T 1 1 0 の内容を設定したうえ、当該 S L C T 1 1 0 をキャッシュディレクトリ 1 0 0 に接続して、その後に S L C T 1 1 0 へ S G C T 1 2 0 を接続する。また、もし、確保したキャッシュセグメントに対応する S G C T 1 2 0 以外の S G C T 1 2 0 が既に S L C T 1 1 0 に接続されている場合は、C P U 3 3 は、その S L C T 1 1 0 に接続されている終端の S G C T 1 2 0 に、確保したキャッシュセグメントの S G C T 1 2 0 を接続する。なお、F M セグメント割り当て処理の終了後には、F M 優先セグメント割り当て処理を終了する。

[0117] ステップ S 4 3 では、C P U 3 3 は、使用可能な R A M セグメント 3 4 3 があるか否かを判定する。この判定結果が真の場合（ステップ S 4 3 : Y E S）は、C P U 3 3 は、処理をステップ S 4 5 へ進める一方、偽の場合（ステップ S 4 3 : N O）は、いずれかのキャッシュセグメントが使用可能になるまで待って（ステップ S 4 4）、処理をステップ S 4 1 に進める。

[0118] ステップ S 4 5 では、C P U 3 3 は、R A M セグメント割り当て処理を行う。R A M セグメント割り当て処理は、ステップ S 4 2 における F M セグメント割り当て処理において、F M セグメント 3 2 5 を割り当てていたものを、R A M セグメント 3 4 3 を割り当てるようにしたものである。R A M セグメント割り当て処理の終了後には、F M 優先セグメント割り当て処理を終了する。

[0119] この F M 優先セグメント割り当て処理では、F M セグメント 3 2 5 が優先して割り当てられる。

[0120] 図 1 6 は、実施例 1 に係る R A M 優先セグメント割り当て処理のフローチャートである。

[0121] R A M 優先セグメント割り当て処理は、図 1 4 に示すキャッシュ割り当て処理のステップ S 3 7 に対応する処理である。

[0122] R A M 優先セグメント割り当て処理は、図 1 5 に示す F M 優先セグメント割り当て処理における F M セグメントと、R A M セグメントとを入れ替えた

処理であるので、ここでは、簡単に説明を行う。

[0123] まず、CPU 33は、使用可能なRAMセグメント343があるか否かを判定する（ステップS51）。この判定結果が真の場合（ステップS51：YES）は、CPU 33は処理をステップS52に進める一方、偽の場合（ステップS51：NO）は、処理をステップS53に進める。

[0124] ステップS52では、CPU 33は、RAMセグメント割り当て処理を行う。このRAMセグメント割り当て処理は、図15のステップS45と同様な処理である。RAMセグメント割り当て処理の終了後には、RAM優先セグメント割り当て処理を終了する。

[0125] ステップS53では、CPU 33は、使用可能なFMセグメント325があるか否かを判定する。この判定結果が真の場合（ステップS53：YES）は、CPU 33は、処理をステップS55に進める一方、偽の場合（ステップS53：NO）は、いずれかのキャッシュセグメントが使用可能になるまで待って（ステップS54）、処理をステップS51に進める。

[0126] ステップS55では、CPU 33は、FMセグメント割り当て処理を行う。このFMセグメント割り当て処理は、図15のステップS42と同様な処理である。FMセグメント割り当て処理の終了後には、RAM優先セグメント割り当て処理を終了する。

[0127] このRAM優先セグメント割り当て処理では、RAMセグメント343が優先して割り当てられる。

[0128] 図17は、実施例1に係るアクセスモニタ集計処理のフローチャートである。

[0129] アクセスモニタ集計処理は、例えば一定時間周期に実行され、その期間におけるリード・ライト量やリード・ライト頻度を集計してアクセスモニタテーブル342を更新する処理である。

[0130] まずCPU 33は、アクセスモニタテーブル342のリード速度342aを更新する（ステップS61）。すなわち、CPU 33は、リード量カウンタ342eの値を、モニタ開始時刻342iから現在までの時間（以下、モ

ニタ時間と称する)で割った値を、リード速度として、アクセスモニタテーブル342のリード速度342aに設定する。

[0131] 次に、CPU33は、アクセスモニタテーブル342のライト速度342bを更新する(ステップS62)。すなわち、CPU33は、ライト量カウンタ342fの値を、モニタ時間で割った値を、ライト速度として、アクセスモニタテーブル342のライト速度342bに設定する。

[0132] 続いて、CPU33は、アクセスモニタテーブル342のリード頻度342cを更新する(ステップS63)。すなわち、CPU33は、リードコマンドカウンタ342gの値を、モニタ時間で割った値を、リード頻度として、アクセスモニタテーブル342のリード頻度342cに設定する。

[0133] その次に、CPU33は、アクセスモニタテーブル342のライト頻度342dを更新する(ステップS64)。すなわち、CPU33は、ライトコマンドカウンタ342hの値を、モニタ時間で割った値を、ライト頻度として、アクセスモニタテーブル342のライト頻度342dに設定する。

[0134] そして、CPU33は、アクセスモニタテーブル342のモニタ開始時刻342iに現在時刻を設定し(ステップS65)、リード量カウンタ342e、ライト量カウンタ342f、リードコマンドカウンタ342g、およびライトコマンドカウンタ342hの値を0にリセットし(ステップS66)、アクセスモニタ集計処理を終了する

[0135] このアクセスモニタ集計処理により、論理ユニット内の各部分領域についてのライト速度、リード速度、ライト頻度、およびリード頻度を適切に把握することができる。

[0136] 上で説明したアクセスモニタ集計処理では、リード速度、ライト速度を、リード量・ライト量をモニタ時間で除算して算出しているが、別の実施形態として、アクセスモニタテーブル342に、リードコマンド、ライトコマンドの処理時間(コマンドを受信してから、ホスト計算機10に応答を返すまでに要した時間)の累積を記憶するようにし、リード量・ライト量をそれぞれ、リードコマンド、ライトコマンドの処理時間の累積で除算するようにし

てもよい。論理ユニット内のある部分領域に対して、コマンド（リードあるいはライトコマンド）が複数回発行された場合を想定すると、リード量・ライト量をモニタ時間で除算してアクセス速度を算出すると、転送長が長いデータへのアクセスであっても、発行されるコマンドの時間間隔が長い場合には、算出されるアクセス速度が低くなる。リード量・ライト量をそれぞれ、リードコマンド、ライトコマンドの処理時間の累積で除算した値をリード速度、ライト速度とすると、論理ユニット内のある部分領域に対して発行されるコマンドの時間間隔が長い、短いにかかわらず、転送長が長いデータへのアクセスの場合には、算出されるリード速度、ライト速度が高くなるので、論理ユニット内のある部分領域に対して発行されるコマンドの時間間隔が長い、短いにかかわらず、転送長が長いデータへのアクセスの場合にRAMに優先的にキャッシュするようにしたい場合、リード量・ライト量をそれぞれ、リードコマンド、ライトコマンドの処理時間の累積で除算した値をリード速度、ライト速度とするとよい。

- [0137] なお、アクセスモニタテーブル342は論理ユニット内の部分領域ごとに設けられ、部分領域ごとにアクセス頻度やアクセス速度が集計される。部分領域（つまり1つのアクセスモニタテーブル342でアクセス頻度やアクセス速度を集計する単位）のサイズとしては、先に述べたとおり様々なサイズを選択可能であるが、本発明におけるストレージシステムでは、アクセスモニタテーブル342の情報に基づいて、キャッシュセグメントとして割り当てるべきメモリ種類を決定するため、キャッシュセグメントとして割り当てるべきメモリ種類の決定される際、図14等の処理はキャッシュセグメント単位で判定を行っているが、結果としては部分領域単位に割り当てられるメモリ種類が定まることになる。そのため、割り当てられるキャッシュメモリの種類を、できるだけ小さい領域ごとに選択したい場合、部分領域のサイズを小さくすることが好ましい。たとえば部分領域のサイズが、キャッシュセグメントと同じサイズであれば、キャッシュセグメントごとのアクセス特性に応じたメモリ割り当てが可能になる。ただし、部分領域のサイズが小さい

と、維持管理すべきアクセスモニタテーブル 342 の量が増加するため、キャッシュセグメントサイズの整数倍（たとえばスロットのサイズでもよい）、あるいはスロットサイズの整数倍としてもよい。

[0138] また、本実施例 1 におけるアクセスモニタテーブル 342 とアクセスモニタ集計処理は、キャッシュ割り当て処理を行うために専用に設けられているものだが、別の実施形態として、ストレージシステム 20 内に、キャッシュ割り当て処理以外の目的で、論理ボリュームの部分領域のアクセス頻度等の情報を集計する手段が設けられている場合、当該手段により集計された情報を用いるようにしてもよい。たとえば米国特許出願公開第 2013/0036250 号明細書や、米国特許出願公開第 2010/0205390 号明細書には、いわゆるシンプロビジョニングに従う論理ボリュームを提供するストレージ装置が、論理ボリュームの各部分領域（ページと呼ばれる）に割り当てるべき記憶領域を決定するために、ページ毎にアクセス頻度などを集計する機能を有することが開示されている。本発明の実施例に係るストレージシステム 20 がこのような機能を備えている場合には、上で説明したアクセスモニタテーブル 342 とアクセスモニタ集計処理に代えて、ページ毎に集計されたアクセス頻度情報を用いて、キャッシュ割り当ての判定を行うようにしてもよい。

[0139] 図 18 は、実施例 1 に係る判定用閾値算出方法を説明する図である。

[0140] 図 14 に示すキャッシュ割り当て処理のステップ S31 において、対象データが高速にアクセスされるか否かを判定する際に基準とする閾値は、FM ボード 32 で使用される FM チップ 321 の寿命が過度に短くならないよう、つまり FM チップ 321 に対する更新・消去頻度が高くなりすぎないような値を設定することが望ましい。そのため、使用する FM チップ 321 の書き換え可能回数や容量、ストレージシステム 20 の性能（たとえばホスト計算機 10 からのライト頻度）などを考慮して、閾値を決定する。この閾値はあらかじめ定められた固定値としてもよいが、下記のようにアクセスモニタテーブル 342 の値を用いて動的に決めてもよい。このようにすると、より

状況にあった閾値を設定することができる。

[0141] まず、CPU 33は、図18の左側のグラフに示すようなライト速度にある各領域(部分領域)について、ライト速度が遅い順にソートして図18の右側のグラフに示すようにする。ここで、各領域は、アクセスモニタテーブル342で管理されている部分領域である。そして、各領域のライト速度を順に足していき、合計が式(1)で求められる許容総ライト速度を超えたら、当該領域のライト速度を対象データが高速にアクセスされるか否かを判定する際に基準とする閾値とする。つまり当該領域のライト速度未満のライト速度となる領域がFMチップ321でキャッシュするのに適した領域(FM適切領域)であり、この領域についてFMチップ321にキャッシュするようにし、それ以外の領域については、FMチップ321にキャッシュしないようにする。

[0142] 許容総ライト速度＝残りライト可能量／(残り使用期間×マージン)－その他のFM更新速度・・・(1)

[0143] ここで、残りライト可能量は、FMチップ321の残り書き換え可能回数と容量、およびWA(Write Amplification:リクラメーションやウェアレベリングによってフラッシュメモリへのライト量は何倍に増幅されるかを表す指標)によって決まる。その他のFM更新速度は、ホスト計算機10からのライト以外の処理、例えばキャッシュリプレイスやデステージなどによってFMチップ321が更新される速度である。残り使用期間は、FMチップ321を交換するとして想定されている期日までの期間に相当する。

[0144] 情報システムでは、FMチップ321の残りライト可能量が少なくなった場合に、当該FMチップ321を搭載するFMボード32へのキャッシュ割り当てを止め、新しいFMボード32に交換させることもできる。例えば、FMチップ321の残りライト可能量が予め決められた閾値を下回った場合は、CPU 33は、許容総ライト速度をゼロに設定する。こうすることで、FMセグメントの割り当てが行われなくなる。さらに、CPU 33は、FMボード32の交換を促すメッセージを管理端末に表示したり、管理者宛にメ

ールを送信したりするなどの方法で、FMボード32の交換が必要であることを管理者に知らせる。

[0145] FMボード32の交換時には、CPU33は、まず古いFMボード32上に残っているダーティデータをドライブに書き出した後に、抜去可能であるとの通知を、例えば管理端末に表示する。管理者が古いFMボード32をストレージコントローラ30から抜去して、新しいFMボード32をストレージコントローラ30に挿した後に、CPU33は新しいFMボード32の初期化を行い、残りライト可能量を初期化し、上記同様に許容総ライト速度を計算する。これにより、以降においては、新しいFMボード32を用いてFMセグメントの割り当てが行われることとなる。

[0146] 図19は、実施例1に係るライトコマンド処理のフローチャートである。

[0147] ライトコマンド処理は、ホスト計算機10からストレージコントローラ30がライトコマンドを受信した場合に実行される。

[0148] まず、ライトコマンドを受信したストレージコントローラ30のCPU33は、ライトコマンドで指定されている、ライト対象の論理ボリュームの論理ブロックアドレス(ライト対象アドレス)に対応するキャッシュセグメントが割り当て済みであるか否かを判定する(ステップS71)。この処理はリード処理(図11のS1)と同様であるので詳細な説明は省略する。この結果、キャッシュセグメントが割り当て済みであれば(ステップS71: YES)、ステップS73へ処理を進める一方、キャッシュセグメントが未割り当てであれば(ステップS71: NO)、キャッシュ割り当て処理(図14参照)を実行し(ステップS72)、ステップS73へ処理を進める。キャッシュ割り当て処理では、ライト対象アドレスに対して、RAM34またはFMチップ321からキャッシュセグメントが割り当てられる。なお、ライトされたデータを二重化することで信頼性を確保するために、キャッシュセグメントを2つ割り当てるようにしてもよい。

[0149] ステップS73では、CPU33は、ライト対象アドレスに対応するキャッシュセグメントを含むスロットをロックする。具体的には、CPU33は

、このキャッシュセグメントを含むスロットの S L C T 1 1 0 のスロットステータス 1 1 0 e の「ロック中」を表すビットを O N にすることで、当該スロットがロックされていることを表す。

[0150] 続いて、C P U 3 3 は、ホスト計算機 1 0 に対し、例えば、XFER_RDYを送信することで、データ受領準備ができたことを通知する（ステップ S 7 4）。

[0151] そして、C P U 3 3 は、割り当てたキャッシュセグメントが R A M セグメント 3 4 3 であるか否かを判定する（ステップ S 7 5）。この結果、割り当てたキャッシュセグメントが R A M セグメント 3 4 3 であれば（ステップ S 7 5 : Y E S）、C P U 3 3 は、ホスト計算機 1 0 から受領したデータを R A M セグメント 3 4 3 に格納するデータ受領処理（R A M）（図 2 0 参照）を実行し（ステップ S 7 6）、処理をステップ S 7 8 に進める。一方、割り当てたキャッシュセグメントが F M セグメント 3 2 5 であれば（ステップ S 7 5 : N O）、C P U 3 3 は、ホスト計算機 1 0 から受領したデータを F M セグメント 3 2 5 に格納するデータ受領処理（F M）（図 2 1 参照）を実行し（ステップ S 7 7）、処理をステップ S 7 8 に進める。

[0152] ステップ S 7 8 では、C P U 3 3 は、アクセスモニタテーブル 3 4 2 を更新する。すなわち、C P U 3 3 は、アクセスモニタテーブル 3 4 2 のライト量カウンタ 3 4 2 f に、今回のライトコマンドで受領したデータ量を加算し、ライトコマンドカウンタ 3 4 2 h をインクリメントする。その後、C P U 3 3 は、ライトコマンド処理を終了する。

[0153] 図 2 0 は、実施例 1 に係るデータ受領処理（R A M）のフローチャートである。

[0154] データ受領処理（R A M）は、図 1 9 に示すライトコマンド処理のステップ S 7 6 の処理に対応する。

[0155] まず、C P U 3 3 は、ホスト計算機 1 0 から受領したデータを、R A M セグメント 3 4 3 に書き込む（ステップ S 8 1）。

[0156] 次に C P U 3 3 は、書き込んだデータがダーティデータであると設定する

(ステップS 8 2)。すなわち、C P U 3 3は、S G C T 1 2 0のダーティビットマップ1 2 0 fにおける受領したデータが書き込まれたブロックに対応するビットをO Nに設定する。

[0157] 続いて、C P U 3 3は、コマンドの完了ステータスをホスト計算機1 0に送信し、R A Mセグメント3 4 3を含むスロットを解放（アンロック）し（ステップS 8 4）、データ受領処理（R A M）を終了する。

[0158] 図2 1は、実施例1に係るデータ受領処理（F M）のフローチャートである。

[0159] データ受領処理（F M）は、図1 9に示すライトコマンド処理のステップS 7 7の処理に対応する。

[0160] まず、C P U 3 3は、ホスト計算機1 0から受領したデータを、F Mボード3 2のバッファメモリ3 2 3に書き込む（ステップS 9 1）。

[0161] 次に、C P U 3 3は、書き込んだデータをバッファメモリ3 2 3から読み出すことができるかテストする（ステップS 9 2）。このとき、C P U 3 3は、例えばデータに付加されたC R C（Cyclic Redundancy Check）などの保証コードをチェックすることで、データが正常であることを確認してもよい。

[0162] 次に、C P U 3 3は、書き込んだデータがダーティデータであると設定する（ステップS 9 3）。すなわち、C P U 3 3は、S G C T 1 2 0のダーティビットマップ1 2 0 fにおける、受領したデータが書き込まれたブロックに対応するビットをO Nにする。

[0163] 続いて、C P U 3 3は、コマンドの完了ステータスをホスト計算機1 0に送信し、F Mセグメント3 2 5を含むスロットを解放する（ステップS 9 5）。

[0164] 次に、C P U 3 3は、F Mプロセッサ3 0 2 bに対して、バッファメモリ3 2 3上のデータをF Mチップ3 2 1のキャッシュセグメント3 2 5に格納するよう要求し（ステップS 9 6）、データ受領処理（F M）を終了する。

[0165] 図2 2は、実施例1に係るF Mデータ読み出し処理のフローチャートであ

る。

[0166] FMデータ読み出し処理は、図13に示すデータ送信処理のステップS24における、FMチップ321上のデータをバッファに読み出す要求をFMプロセッサ320bが受信した場合に実行される。

[0167] まず、FMプロセッサ320bは、ストレージコントローラ30のCPU33から指定された論理アドレスを、FMチップ321上のデータ格納位置を表す物理アドレスに変換する（ステップS101）。なお、論理アドレスから物理アドレスへの変換は、バッファメモリ323に格納されている論理アドレスと物理アドレスとの対応関係を示すマッピングテーブルに基づいて行うことができる。

[0168] 次に、FMプロセッサ320bは、FMチップ321の物理アドレスに対応する領域から対象データを読み出し、バッファメモリ323に格納する（ステップS102）。

[0169] そして、FMプロセッサ320bは、完了応答をストレージコントローラ30のCPU33に対して送信し（ステップS103）、FMデータ読み出し処理を終了する。

[0170] 図23は、実施例1に係るFMデータ書き込み処理のフローチャートである。

[0171] FMデータ書き込み処理は、図21に示すデータ受領処理（FM）のステップS96における、バッファメモリ323上のデータをFMチップ321のキャッシュセグメント325に格納する要求を、FMプロセッサ320bが受信した場合に実行される。

[0172] まず、FMプロセッサ320bは、データ格納先のFMチップ321のページ（FMページともいう。）を確保する（ステップS111）。ここで、FMチップ321は、同一のページに対してデータを上書きできないので、FMプロセッサ320bは、既に消去済みのFMページをデータ格納先として選ぶ。もし消去済みのFMページが無い場合は、FMプロセッサ320bは、FMチップ321の空きブロック（空きFMブロックともいう）、すな

わち有効データが格納されていないFMチップ321のブロック（FMブロックともいう）を消去して、そのFMブロックの先頭から必要な量のFMページをデータ格納先として選ぶ。

[0173] 次に、FMプロセッサ320bは、バッファメモリ323上のデータを、確保したFMページに書き込む（ステップS112）。

[0174] そして、FMプロセッサ320bは、今回処理対象とした論理アドレスが、新たにデータを格納したFMページの物理アドレスに対応するよう、論理アドレスと物理アドレスとの対応関係を表すマッピングテーブルを更新し、また、古いデータが格納されているFMページが無効であることを記憶する（ステップS113）。このとき、FMプロセッサ320bは、無効になったFMページが含まれるFMブロックの全FMページが無効である場合には、当該FMブロックを空きFMブロックとして管理する。この時点で、空きFMブロックのデータを消去してもよいし、後でバックグラウンド処理として空きFMブロックのデータを消去してもよい。

[0175] そして、FMプロセッサ320bは、完了応答をストレージコントローラ30のCPU33に対して送信し（ステップS114）、FMデータ書き込み処理を終了する。

実施例 2

[0176] 次に、実施例2に係る情報システムについて説明する。その際、前述の実施例のうちの少なくとも1つの実施例との相違点を主に説明し、前述の実施例のうちの少なくとも1つの実施例との共通点については説明を省略あるいは簡略化する。これは、実施例2に限らず、実施例3以降についても同様である。

[0177] 図24は、実施例2に係る情報システムの構成図である。なお、実施例1に係る情報システムと同様な構成に対しては、同一の符号を付している。

[0178] 実施例2に係る情報システムと、実施例1に係る情報システムとの主な違いは、FMボード32がホスト計算機80に搭載されることにある。このホスト計算機80が、情報処理装置の一例である。

- [0179] 実施例2に係る情報システムは、ホスト計算機80と、ホスト計算機80に直接あるいはネットワークを経由して接続されるHDD40、SSD41、またはストレージシステム20とを含む。
- [0180] ホスト計算機80は、CPU81と、RAM84と、FMボード32と、ストレージインタフェース82と、ネットワークインタフェース83とを含む。
- [0181] ストレージインタフェース82は、HDD40又はSSD41を接続するためのインタフェースである。ネットワークインタフェース83は、ネットワークを介してストレージシステム20と接続するためのインタフェースである。FMボード32は、図4に示す実施例1に係るFMボードと同様な構成である。
- [0182] RAM84は、CPU81が実行するアプリケーションプログラム841と、オペレーティングシステム842（オペレーティングシステムA、オペレーティングシステムB）と、ハイパーバイザプログラム843と、ストレージ制御プログラム340と、キャッシュ制御情報341とを格納する。また、RAM84は、データをキャッシュするためのキャッシュセグメント343を格納する。
- [0183] ハイパーバイザプログラム843は、ホスト計算機80で構築する仮想マシン（VM）を管理するプログラムである。なお、ハイパーバイザプログラム843の機能を、ハードウェアとして実装してもよい。
- [0184] 図25は、実施例2に係るデータ入出力処理の概要を示す図である。
- [0185] 本実施例に係るホスト計算機80においては、一番下の階層には、CPU81がハイパーバイザプログラム843を実行することにより構築されるハイパーバイザHVが位置する。ハイパーバイザHVは、仮想化機構の一種である。仮想化機構は、プログラムを実行するプロセッサを有した計算機であっても良い。ハイパーバイザHVによって、1以上の仮想マシン（図25では、仮想マシンA（VMA）及び仮想マシンB（VMB）が実現される。仮想マシンAでは、オペレーティングシステムAが動作し、その上でアプリケ

ーションプログラム 841 が動く。また、仮想マシン B では、オペレーティングシステム B が動作し、その上でストレージ制御プログラム 340 が動く。ストレージ制御プログラム 340 は、RAM 34 のキャッシュセグメント 343 および FM チップ 321 のキャッシュセグメント 325 を用いて、実施例 1 と同様なキャッシュを制御する。なお、ストレージ制御プログラム 340 はホスト計算機 80 に接続された HDD 40、SSD 41、またはストレージシステム 20 に対するデータ入出力の制御も行う。

[0186] 仮想マシン A のアプリケーションプログラム 841 と、仮想マシン B のストレージ制御プログラム 340 とは、仮想マシン間通信によって互いに通信を行う。なお、この仮想マシン間通信は、ハイパーバイザ HV あるいは各オペレーティングシステム 842 によって仮想化され、アプリケーションプログラム 841 やストレージ制御プログラム 340 にとっては、例えば SCSI のようなストレージインタフェースを介した通信と同様の仮想的なインタフェースを用いて行ってもよい。

[0187] 実施例 2 に係る情報システムによると、ホスト計算機 80 において、RAM キャッシュセグメント 343 と、FM キャッシュセグメント 325 とを用いて、適切にデータのキャッシュを行うことができる。

実施例 3

[0188] 次に、実施例 3 に係る情報システムについて説明する。

[0189] 図 26 は、実施例 3 に係る情報システムの構成を示す図である。

[0190] 実施例 3 に係る情報システムは、実施例 2 に係る情報システムとは、ホスト計算機における RAM で管理する内容が異なっている。

[0191] 実施例 3 に係るホスト計算機 90 は、RAM 91 を含む。このホスト計算機 90 は、情報処理装置の一例である。RAM 91 は、オペレーティングシステム 911 を格納する。オペレーティングシステム 911 は、ストレージ制御プログラム 340 およびキャッシュ制御情報 341 をドライバとして含む。

[0192] 図 27 は、実施例 3 に係るデータ入出力処理の概要を示す図である。

[0193] 実施例 3 に係るホスト計算機 90 では、アプリケーションプログラム 841 がストレージ (HDD 40、SSD 41、またはストレージシステム 20) に対する入出力を行う際、オペレーティングシステム 911 に含まれるストレージ制御プログラム 341 がこの入出力要求を処理し、実施例 1 と同様に、RAM キャッシュセグメント 343 や FM キャッシュセグメント 325 へのキャッシングを行う。また、ストレージ制御プログラム 341 は、ストレージに対して入出力を行うために、各種デバイスドライバ (912、913) にストレージに対する入出力要求を渡す。デバイスドライバ A 912 は、入出力要求に基づいて、ストレージインタフェース 82 を制御する。デバイスドライバ B 913 は、入出力要求に基づいて、ネットワークインタフェース 83 を制御する。

[0194] 実施例 3 に係る情報システムによると、ホスト計算機 90 のオペレーティングシステム 911 が、RAM キャッシュセグメント 343 と、FM キャッシュセグメント 325 とを用いて、適切にデータのキャッシュを行うことができる。

実施例 4

[0195] 次に、実施例 4 に係る情報システムについて説明する。

[0196] 実施例 4 に係る情報システムと実施例 1 に係る情報システムとの違いは、リードコマンド処理の手順である。実施例 4 に係る情報システムでは、ドライブからステージングしたデータを、バッファメモリ 323 から FM チップ 321 に書き込む前に、まずバッファメモリ 323 からホスト計算機 10 へ送信し、その後で FM チップ 321 に書き込む。こうすることでリードコマンド完了までの所要時間 (レスポンスタイム) を短くすることができる。

[0197] 図 28 は、実施例 4 に係るジョブ制御テーブルの構成図である。

[0198] ジョブ制御テーブル 344 は、ジョブ種類 344a と、論理ユニット番号 344b と、論理ブロックアドレス 344c と、転送長 344d と、バッファアドレス 344e とを記憶する。ジョブ種類 344a は、ジョブが行う処理の種類を表す。ジョブ種類 344a は、例えば、リードコマンド処理であ

れば「1」、ライトコマンド処理であれば「2」などのジョブを示すIDである。論理ユニット番号344b、論理ブロックアドレス344c、及び転送長344dは、それぞれ、ホスト計算機10から受信したリード・ライトコマンドに指定された、アクセス対象についての論理ユニットの番号、論理ブロックアドレス（LBA）、転送長を表す。バッファアドレス344eは、このジョブが確保しているバッファのアドレスを表す。バッファアドレス344eは、バッファが確保されてないときは、アドレスが無効であることを表す値（例えばNULL）となっている。

[0199] 図29は、実施例4に係るリードコマンド処理の第1のフローチャートであり、図30は、実施例4に係るリードコマンド処理の第2のフローチャートである。図29のフローチャートの符号Aは、図30のフローチャートの符号Aに繋がっていることを示している。

[0200] リードコマンド処理は、ホスト計算機10からストレージコントローラ30がリードコマンドを受信した場合に実行される。

[0201] まず、リードコマンドを受信したストレージコントローラ30のCPU33は、リードコマンドで指定されているリード対象アドレスに対応するキャッシュセグメントが割り当て済みであるか否かを判定する（ステップS1）。この結果、キャッシュセグメントが割り当て済みであれば（ステップS1：YES）、CPU33は、ステップS3へ処理を進める一方、キャッシュセグメントが未割り当てであれば（ステップS1：NO）、キャッシュ割り当て処理（図14参照）を実行し（ステップS2）、ステップS3へ処理を進める。キャッシュ割り当て処理は、実施例1で説明した通りである。

[0202] ステップS3では、CPU33は、リード対象アドレスに対応するキャッシュセグメントを含むスロットをロックする。具体的には、CPU33は、このキャッシュセグメントを含むスロットのSLCT110のスロットステータス110eの「ロック中」を表すビットをONにすることで、当該スロットがロックされていることを表す。

[0203] 続いてCPU33は、リード対象のデータがキャッシュセグメントに格納

されているか否か、すなわち、キャッシュヒットか否かを判定する（ステップS4）。具体的には、CPU33は、リード対象のキャッシュセグメントに対応するSGCT120のステージングビットマップ120eおよびダーティビットマップ120fを調べ、リード対象の全ブロックに関して、当該ブロックに対応するステージングビットマップ120eのビットもしくはダーティビットマップ120fのビットのいずれかがONであれば、キャッシュヒットであると判断する。一方、CPU33は、リード対象の範囲内に、ダーティビットマップ120fとステージングビットマップ120eとの対応するビットがいずれもOFFであるブロックが一つでもある場合は、キャッシュミスであると判断する。

[0204] この結果、キャッシュヒットの場合（ステップS4：YES）は、CPU33は、処理をステップS122へ進める一方、キャッシュミスの場合（ステップS4：NO）は、ステージング処理（図31参照）を実行し（ステップS121）、処理をステップS122に進める。ステージング処理では、ドライブ（HDD40、またはSSD41）からキャッシュセグメント（325又は343）へデータが読み込まれる。ステージング処理が完了すると、リード対象のデータがキャッシュセグメント（325又は343）に格納された状態になる。

[0205] ステップS122では、CPU33は、キャッシュセグメントに格納されたデータをホスト計算機10に送信するデータ送信処理（図32参照）を実行する。

[0206] 続いて、CPU33は、コマンドの完了ステータスをホスト計算機10に送信する（ステップS7）。すなわち、CPU33は、コマンドの処理中にエラーが発生してリード処理が正常に完了しなかった場合はエラーステータス（例えばCHECK CONDITION）を返す一方、リード処理が正常に完了した場合は正常ステータス（GOOD）を返す。

[0207] 次に、CPU33は、FMチップ321へのライトを実施中か否かを判定する（ステップS123）。「FMチップ321へのライトを実施中」とは

、バッファメモリ 323 から FM チップ 321 へのデータ書き込み要求を FM プロセッサ 320b に送信した後、まだ FM プロセッサ 320b から完了通知を受信していない状態を意味する。この結果、真であれば（ステップ S123：YES）、CPU 33 は、FM プロセッサ 320b からの完了通知を待ち（ステップ S124）、処理をステップ S125 に進める。一方、偽であれば（ステップ S123：NO）、CPU 33 は、処理をステップ S125 に進める。

[0208] ステップ S125 では、CPU 33 は、バッファを解放する。次いで、CPU 33 は、ロックしているスロットを解放（アンロック）し（ステップ S8）、アクセスモニタテーブル 342 を更新し（ステップ S9）、リードコマンド処理を終了する。アクセスモニタテーブル 342 の更新とは、例えば、リード量カウンタ 342e に今回のリードコマンドで読み出したデータ量を加算し、リードコマンドカウンタ 342g をインクリメントする。

[0209] 図 31 は、実施例 4 に係るステージング処理のフローチャートである。

[0210] ステージング処理は、図 29 のリードコマンド処理のステップ S121 の処理に対応する。

[0211] まず、CPU 33 は、リード対象アドレスに割り当てられたキャッシュセグメントの基になっているキャッシュメモリの種類を調べ、キャッシュセグメントが RAM 34 上のキャッシュセグメント（RAM セグメント）343 であるか否かを判定する（ステップ S11）。ここで、キャッシュセグメントの基になっているキャッシュメモリの種類は、対応する SGCT 120 のメモリ種類 120c を参照することにより特定することができる。

[0212] この結果、キャッシュセグメントが RAM セグメント 343 であれば（ステップ S11：YES）、CPU 33 は、処理をステップ S12 に進める一方、キャッシュセグメントが RAM セグメント 343 でなければ（ステップ S11：NO）、処理をステップ S13 に進める。

[0213] ステップ S12 では、CPU 33 は、リード対象（ステージング対象）のデータをドライブ（HDD 40 または SSD 41）から読み出して、RAM

セグメント 3 4 3 に格納し、ステージング処理を終了する。

[0214] ステップ S 1 3 以降の処理では、キャッシュセグメントが R A M セグメント 3 4 3 でない、すなわち F M チップ 3 2 1 上のキャッシュセグメント (F M セグメント) 3 2 5 であるので、ドライブから読み上げたデータを直接 F M チップ 3 2 1 に書くのではなく、一旦、 F M ボード 3 2 のバッファメモリ 3 2 3 に格納した後、バッファメモリ 3 2 3 から F M チップ 3 2 1 に書き込むようにする。

[0215] まず、ステップ S 1 3 では、 C P U 3 3 は、バッファメモリ 3 2 3 にドライブから読み上げたデータを格納するための領域 (バッファ) の確保を行う。すなわち、 C P U 3 3 は、バッファメモリ 3 2 3 のうちステージング対象のデータを格納するのに十分な領域をバッファに割り当てる。

[0216] 次に、 C P U 3 3 は、ステージング対象のデータをドライブから読み出してバッファに格納する (ステップ S 1 4) 。本実施例では、 B E I / F 3 5 が、 C P U 3 3 の指示を受けて、ドライブから F M ボード 3 2 のバッファメモリ 3 2 3 のバッファに対してデータを格納している。

[0217] そして、 C P U 3 3 は、 F M プロセッサ 3 2 0 b に対し、バッファメモリ 3 2 3 のバッファ上のデータを F M チップ 3 2 1 に格納するよう要求する (ステップ S 1 5) 。この要求に対して F M プロセッサ 3 2 0 b は、データ送信処理 (図 3 2 参照) を実行する。

[0218] その後、 C P U 3 3 は、ステージング処理を終了する。

[0219] 図 3 2 は、実施例 4 に係るデータ送信処理のフローチャートである。

[0220] データ送信処理は、図 2 9 に示すリードコマンド処理のステップ S 1 2 2 の処理に対応する。

[0221] まず、 C P U 3 3 は、リード対象アドレスに割り当てられたキャッシュセグメントの基になっているキャッシュメモリの種類を調べ、キャッシュセグメントが R A M セグメント 3 4 3 であるか否かを判定する (ステップ S 2 1) 。ここで、キャッシュセグメントの基になっているキャッシュメモリの種類は、対応する S G C T 1 2 0 のメモリ種類 1 2 0 c を参照することにより

特定することができる。

- [0222] この結果、キャッシュセグメントがRAMセグメント343であれば（ステップS21：YES）、CPU33は、処理をステップS22に進める一方、キャッシュセグメントがRAMセグメント343でなければ（ステップS21：NO）、処理をステップS131に進める。
- [0223] ステップS22では、CPU33は、リード対象（送信対象）のデータをRAMセグメント343からホスト計算機10へ転送し、データ送信処理を終了する。
- [0224] ステップS131では、CPU33は、リード・ライトコマンドに対応するジョブ制御テーブル344のバッファアドレス344eが有効であるか否かをチェックする。この結果、バッファアドレス344eが有効であれば（ステップS131：VALID）、CPU33は、処理をステップS132に進める一方、バッファアドレス344eが無効であれば、処理をステップS23に進める。
- [0225] ステップS23では、CPU33は、バッファメモリ323にバッファの確保を行う。すなわち、CPU33は、バッファメモリ323から送信対象のデータを格納するのに十分な領域を割り当てる。
- [0226] 次に、CPU33は、FMプロセッサ320bに対し、FMチップ321のデータをバッファメモリ323のバッファに読み出すよう要求する（ステップS24）。この要求に対してFMプロセッサ320bは、FMデータ読み出し処理（図22参照）を実行する。FMデータ読み出し処理により、バッファメモリ323には、送信対象のデータが格納される。FMプロセッサ320bは、FMデータ読み出し処理を終了すると、その要求に対する完了応答をCPU33に返す。
- [0227] 続いて、CPU33は、その要求に対する完了応答をFMプロセッサ320bから受け取り（ステップS25）、処理をステップS132に進める。
- [0228] ステップS132では、CPU33は、送信対象のデータをバッファメモリ323からホスト計算機10に送信する。

実施例 5

- [0229] 次に、実施例 5 に係る情報システムについて説明する。
- [0230] 実施例 5 に係る情報システムと実施例 1 に係る情報システムとの違いは、実施例 5 に係る情報システムが、キャッシュセグメントが割り当てられているメモリ種類の見直しを行い、データのアクセス特性等に応じて、割り当て済のキャッシュセグメントから、異なるメモリ種類のキャッシュセグメントへデータを移動させるようにしたことである。これにより、データをそのデータのアクセス特性に応じた適切なキャッシュセグメントに格納することができる。
- [0231] 図 3 3 は、実施例 5 に係るメモリ種類見直し処理のフローチャートである。
- [0232] メモリ種類見直し処理は、例えば割り当て済の各セグメントに対して定期的に行ってもよいし、データのあるドライブから別のドライブへ移す際に、当該データが格納されているセグメントに対して行ってもよい。
- [0233] まず、CPU 3 3 は、処理対象のキャッシュセグメント（図 3 3 の説明において処理対象セグメントという）を含むスロットをロックする（ステップ S 1 5 1）。
- [0234] 次に、CPU 3 3 は、処理対象セグメントのメモリ種類が適切であるか否かを判定する（ステップ S 1 5 2）。この判定の基準には、図 1 4 に示したキャッシュ割り当て処理の判定基準の一部あるいは全部を用いることができる。例えばアクセス頻度が高い場合や、SSD のデータを格納している場合などは、RAM セグメント 3 4 3 が適切であり、それ以外の場合は、FM セグメント 3 2 5 が適切であると判定する。
- [0235] この判定の結果が真、すなわちメモリ種類が適切な場合（ステップ S 1 5 2：YES）は、CPU 3 3 は、スロットを解放し（ステップ S 1 5 3）、メモリ種類見直し処理を終了する。一方、判定の結果が偽、すなわちメモリ種類が不適切な場合（ステップ S 1 5 2：NO）は、以下の処理に進む。
- [0236] すなわち、CPU 3 3 は、適切なメモリ種類の空きセグメントがあるか調

べる（ステップS 1 5 4）。この結果、適切なメモリ種類の空きセグメントが無ければ（ステップS 1 5 4 : N O）、C P U 3 3は、スロットを解放し（ステップS 1 5 3）、メモリ種類見直し処理を終了する。

[0237] 一方、適切なメモリ種類の空きセグメントが有れば（ステップS 1 5 4 : Y E S）、C P U 3 3は、処理対象セグメントのデータに対して、適切なメモリ種類のセグメントを新しく割り当てる（ステップS 1 5 5）。次に、C P U 3 3は、データを古いキャッシュセグメントから新しいキャッシュセグメントにコピーし（ステップS 1 5 6）、古いキャッシュセグメントを解放し（ステップS 1 5 7）、スロットを解放し（ステップS 1 5 3）、メモリ種類見直し処理を終了する。

[0238] 以上、いくつかの実施例を説明したが、これらは、本発明の説明のための例示であって、本発明の範囲をこれらの実施例にのみ限定する趣旨ではない。すなわち、本発明は、他の種々の形態でも実施する事が可能である。

符号の説明

[0239] 2 0 : ストレージシステム
3 0 : ストレージコントローラ
3 2 : F M ボード
3 4 : R A M
3 2 1 : F M チップ

請求項 4 に記載の情報処理装置。

[請求項6] 前記第 2 のキャッシュメモリよりもアクセス性能の高いバッファメモリを更に備え、

前記制御デバイスは、前記第 2 のキャッシュメモリに対して前記データを格納する際には、前記バッファメモリに前記データを格納し、その後、前記バッファメモリから前記第 2 のキャッシュメモリに格納する

請求項 3 に記載の情報処理装置。

[請求項7] 前記第 1 のメモリは、R A M (Random Access Memory) であり、
前記第 2 のメモリは、フラッシュメモリである

請求項 3 に記載の情報処理装置。

[請求項8] 前記制御デバイスは、前記データがシーケンシャルアクセスの対象となるデータである場合に、前記第 2 のキャッシュメモリよりも前記第 1 種のキャッシュメモリに優先的にキャッシュする

請求項 3 に記載の情報処理装置。

[請求項9] 前記制御デバイスは、
前記データが最終的に管理されるストレージデバイスが前記第 2 のメモリで構成されている場合には、前記第 1 のキャッシュメモリに優先的にキャッシュする

請求項 3 に記載の情報処理装置。

[請求項10] 前記制御デバイスは、前記データが一時的に使用されるデータである場合に、前記第 1 のキャッシュメモリに優先的にキャッシュする

請求項 3 に記載の情報処理装置。

[請求項11] 前記複数種類のキャッシュメモリは、第 3 のメモリで構成された第 3 のキャッシュメモリと、前記第 3 のメモリよりもデータの更新回数が制限されている第 4 のキャッシュメモリであり、

前記制御デバイスは、更新頻度が高いデータについては、前記第 4 のキャッシュメモリよりも前記第 3 のキャッシュメモリに優先的にキ

ヤツシュする

請求項 1 に記載の情報処理装置。

[請求項12] 前記第3のメモリは、RAMであり、
前記第4のメモリは、フラッシュメモリである
請求項11に記載の情報処理装置。

[請求項13] ホストコンピュータからのI/Oリクエストを受信し、

前記I/Oリクエストに伴うデータのアクセス特性に応じて、特性の異なる複数のキャッシュメモリのいずれか1つに前記データをキャッシュする

キャッシュ制御方法。

[請求項14] 前記複数のキャッシュメモリは、ＲＡＭにより構成される第１のキャッシュメモリ及びフラッシュメモリにより構成される第２のキャッシュメモリであり、

所定の閾値よりもアクセス頻度が高いデータについては、前記第２のキャッシュメモリよりも前記第１のキャッシュメモリに優先的にキャッシュする

請求項１３に記載のキャッシュ制御方法。

[請求項15] 前記複数のキャッシュメモリは、RAMにより構成される第1のキャッシュメモリ及びフラッシュメモリにより構成される第2のキャッシュメモリであり、

前記データがシーケンシャルアクセスの対象となるデータである場合は、前記第2のキャッシュメモリよりも前記第1のキャッシュメモリに優先的にキャッシュする

請求項13に記載のキャッシュ制御方法。

[図1]

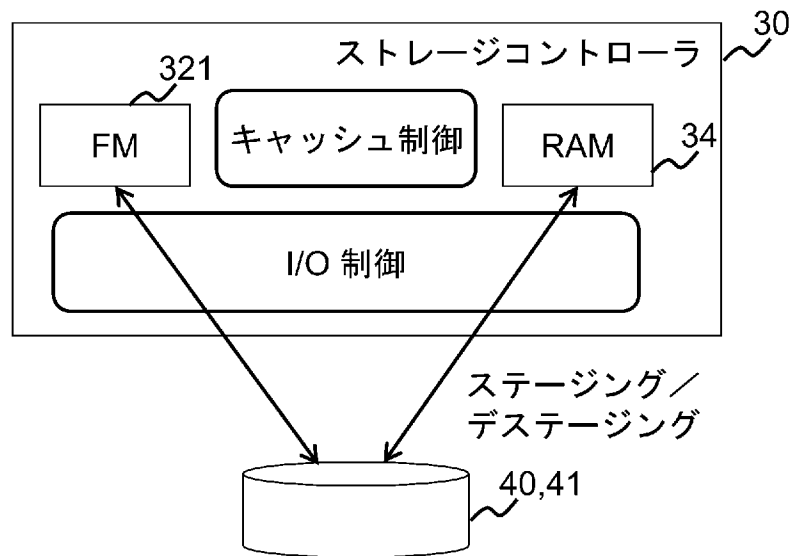


図 1

[図2]

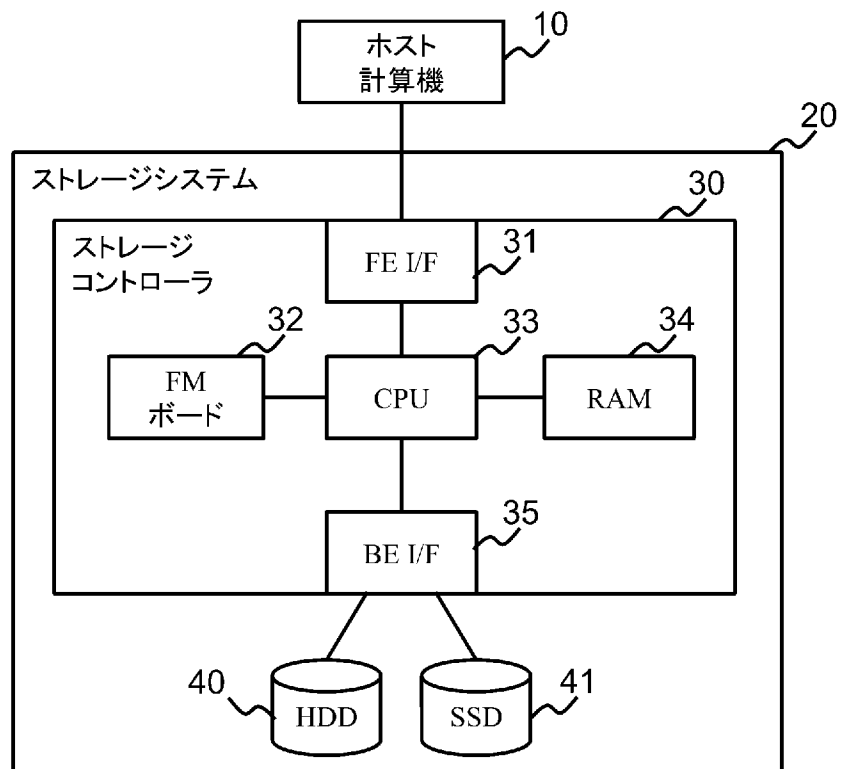


図 2

[図3]

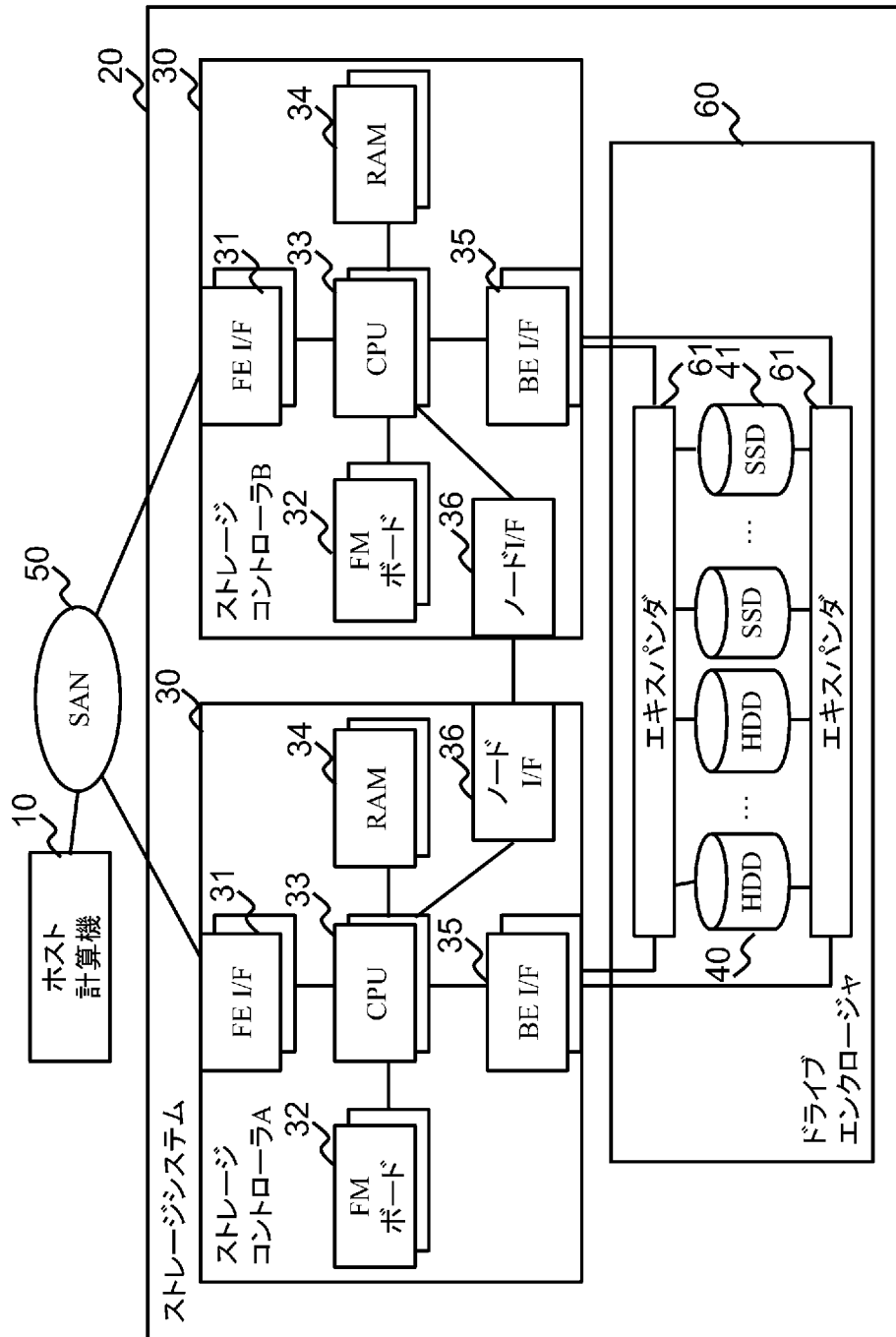


図 3

[図4]

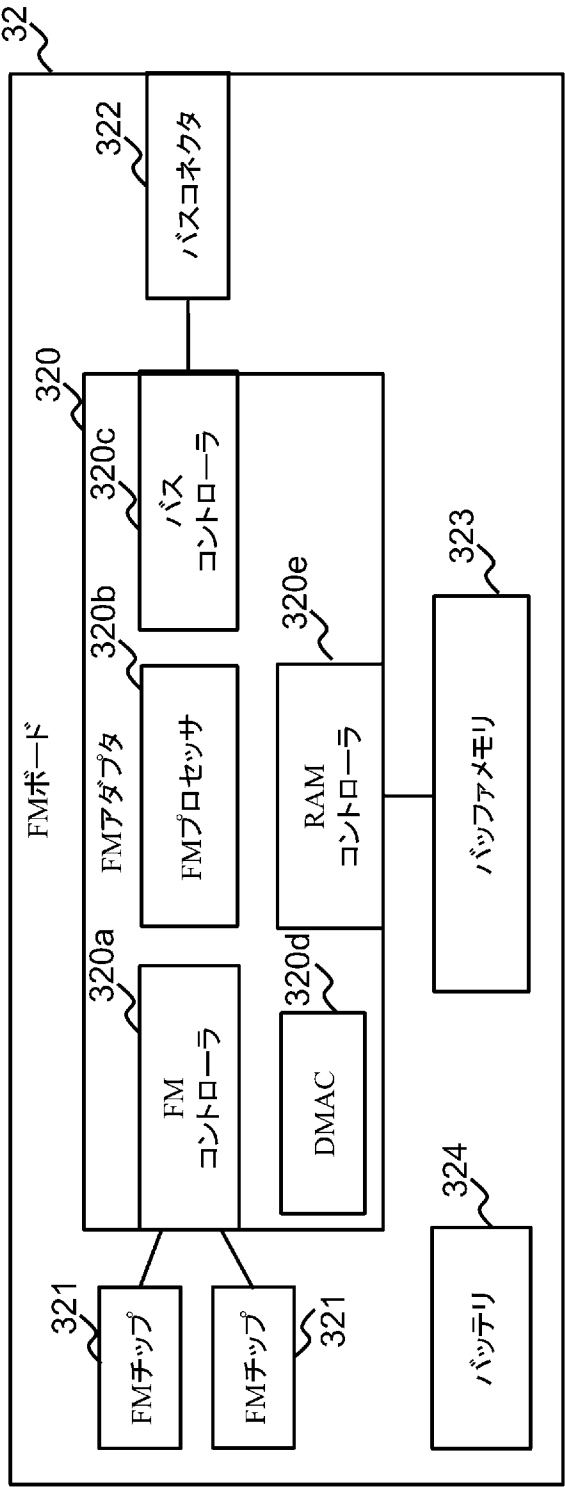


図 4

[図5]

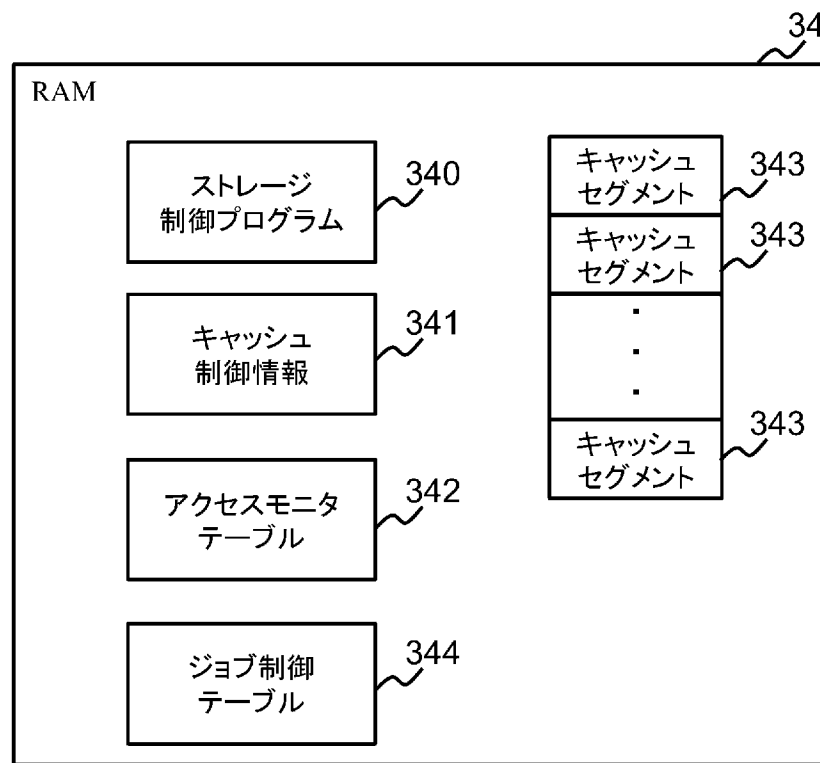


図 5

[図6]

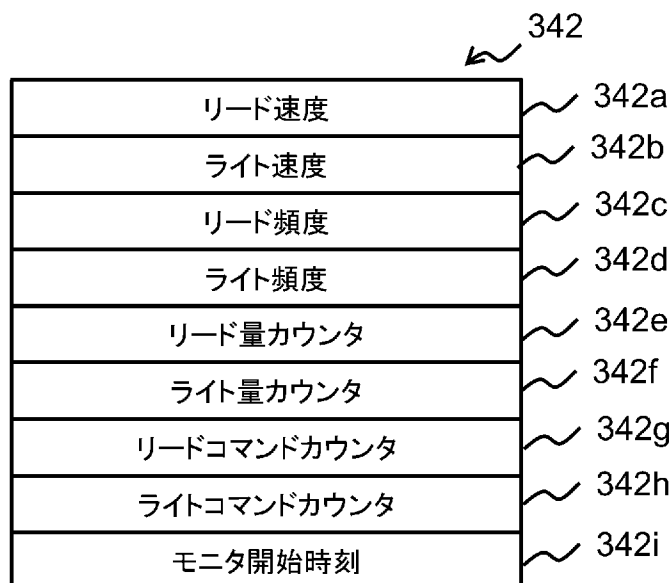


図 6

[図7]

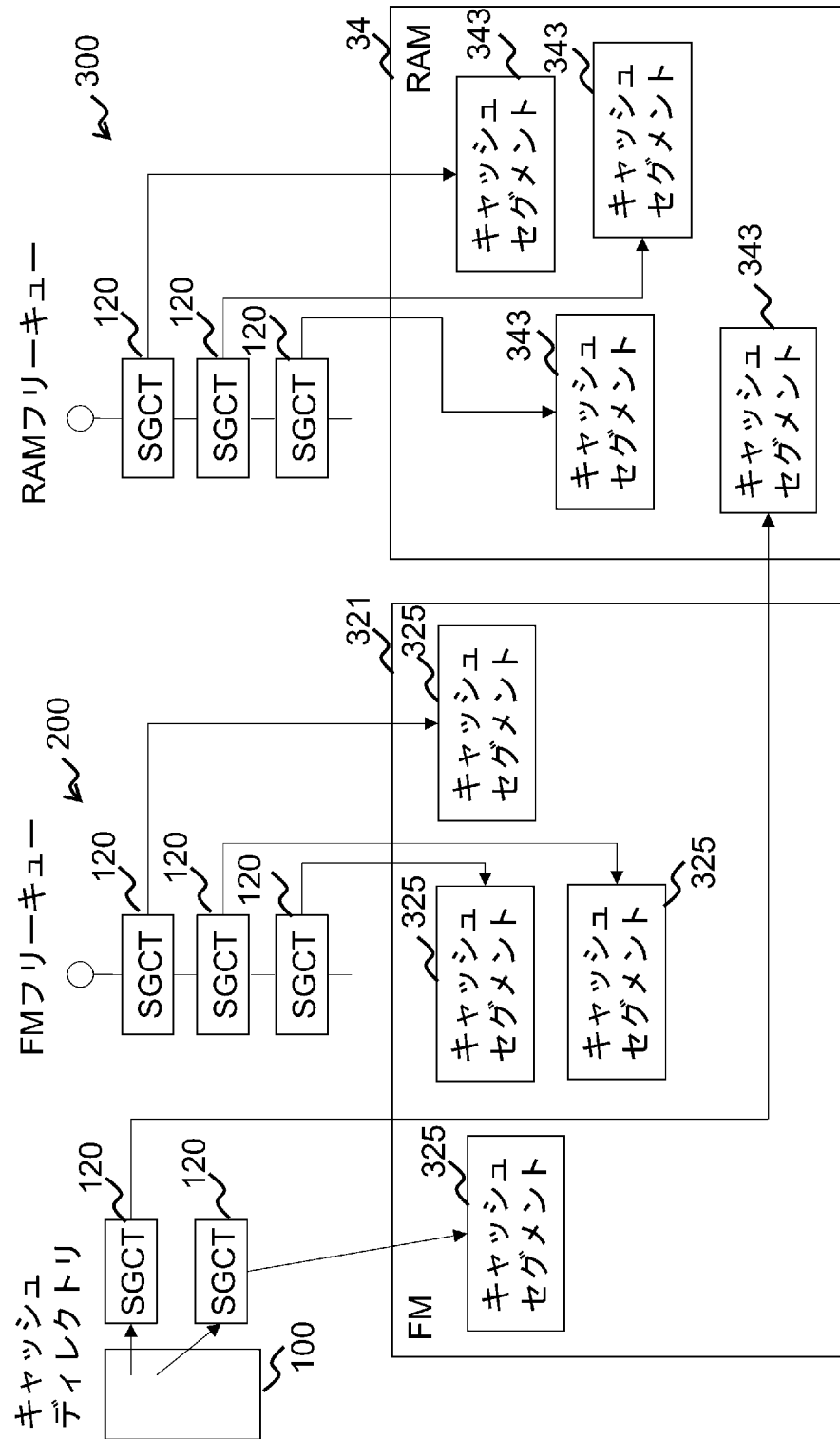


図 7

[図8]

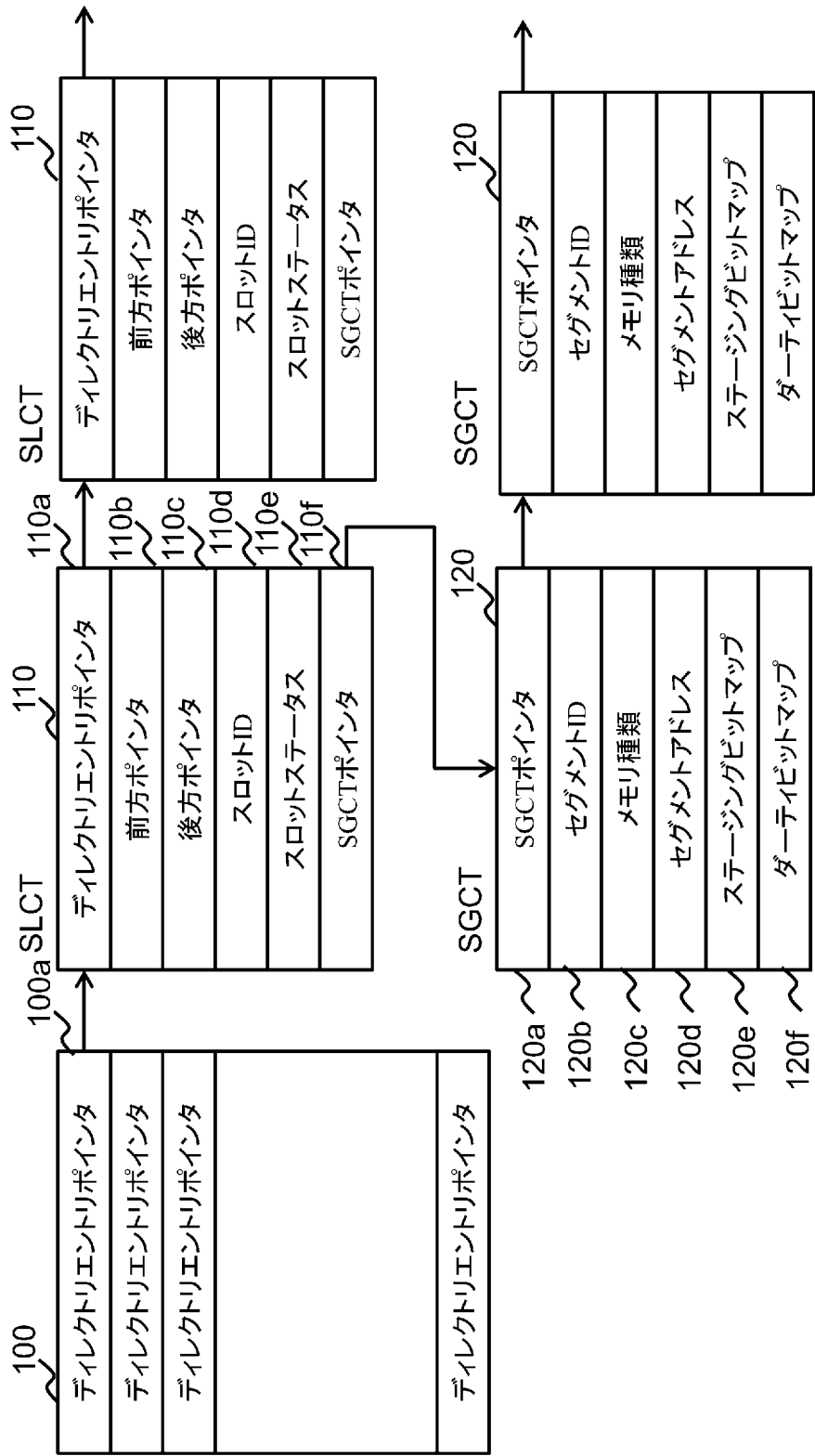


図 8

[図9]

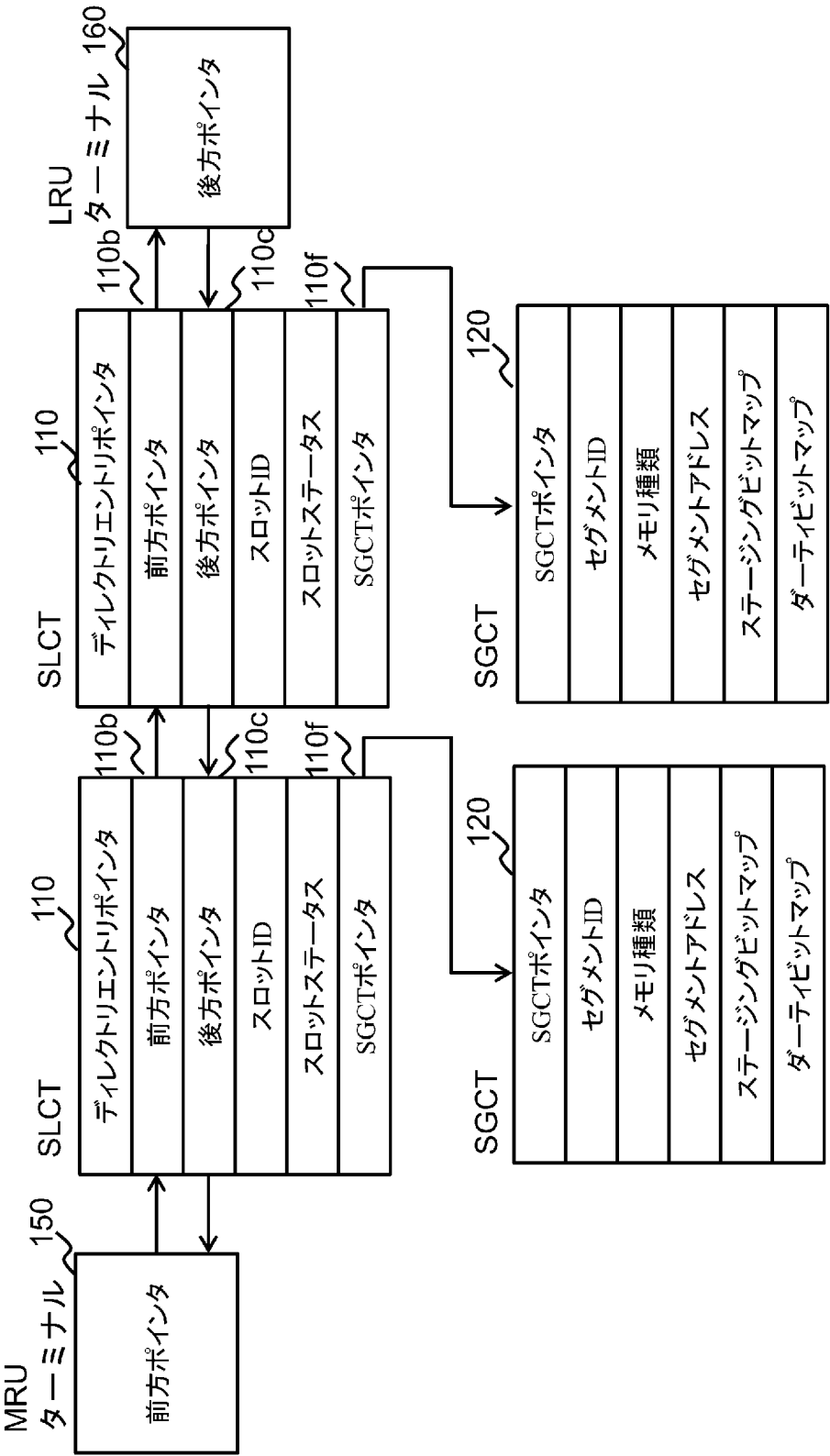


図9

[図10]

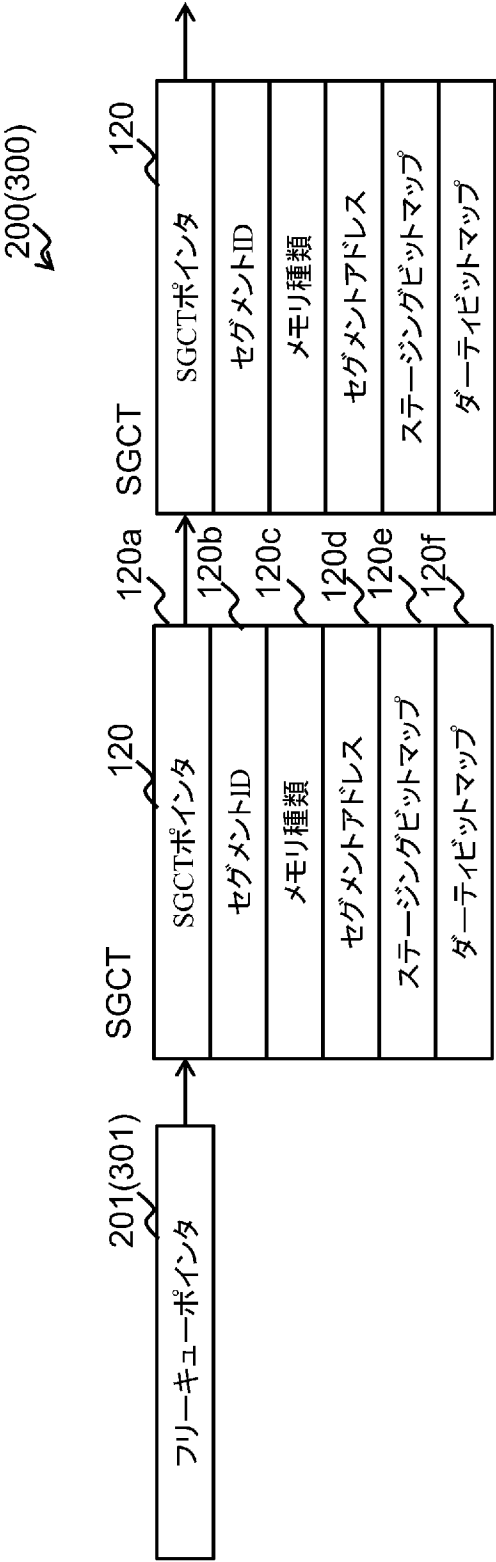


図 10

[図11]

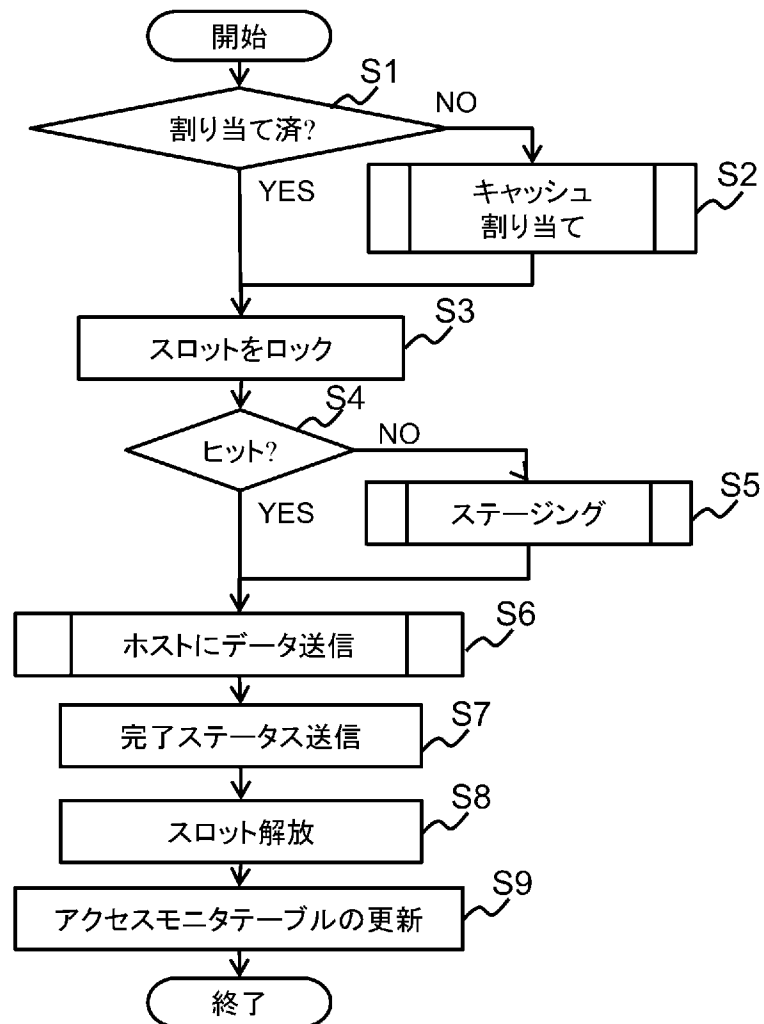


図 1 1

[図12]

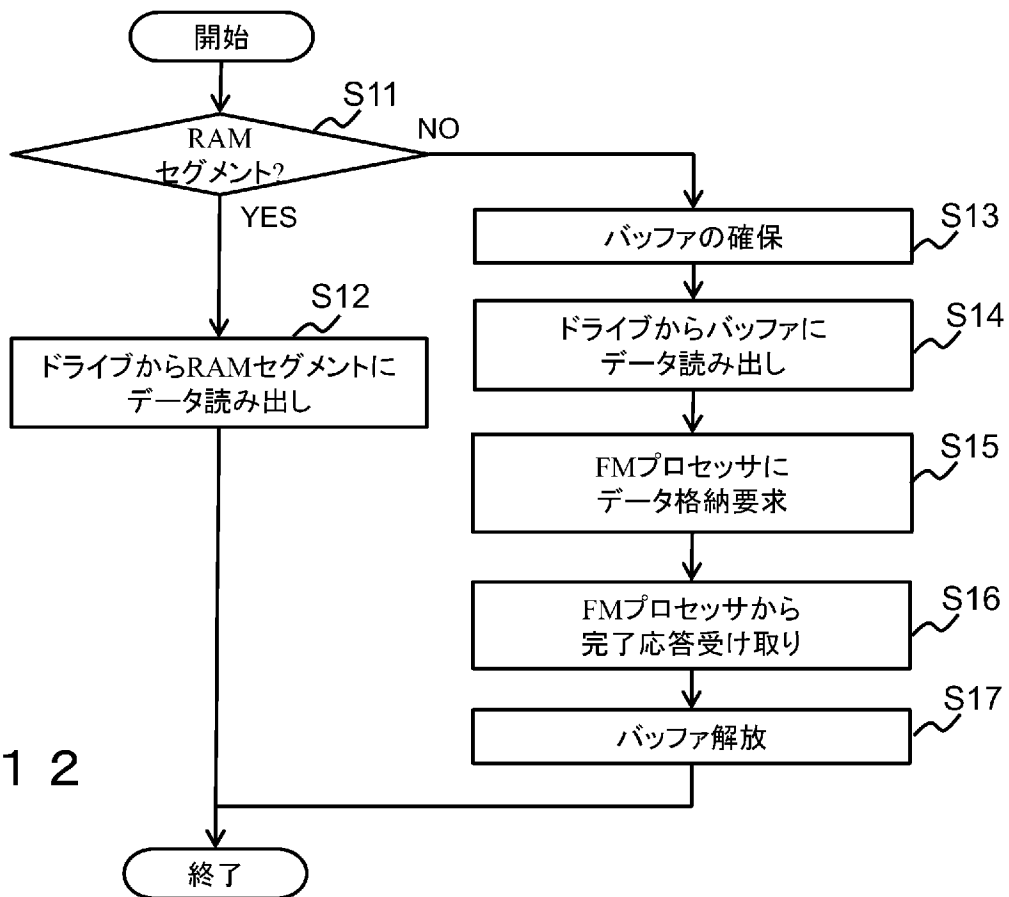


図 1 2

[図13]

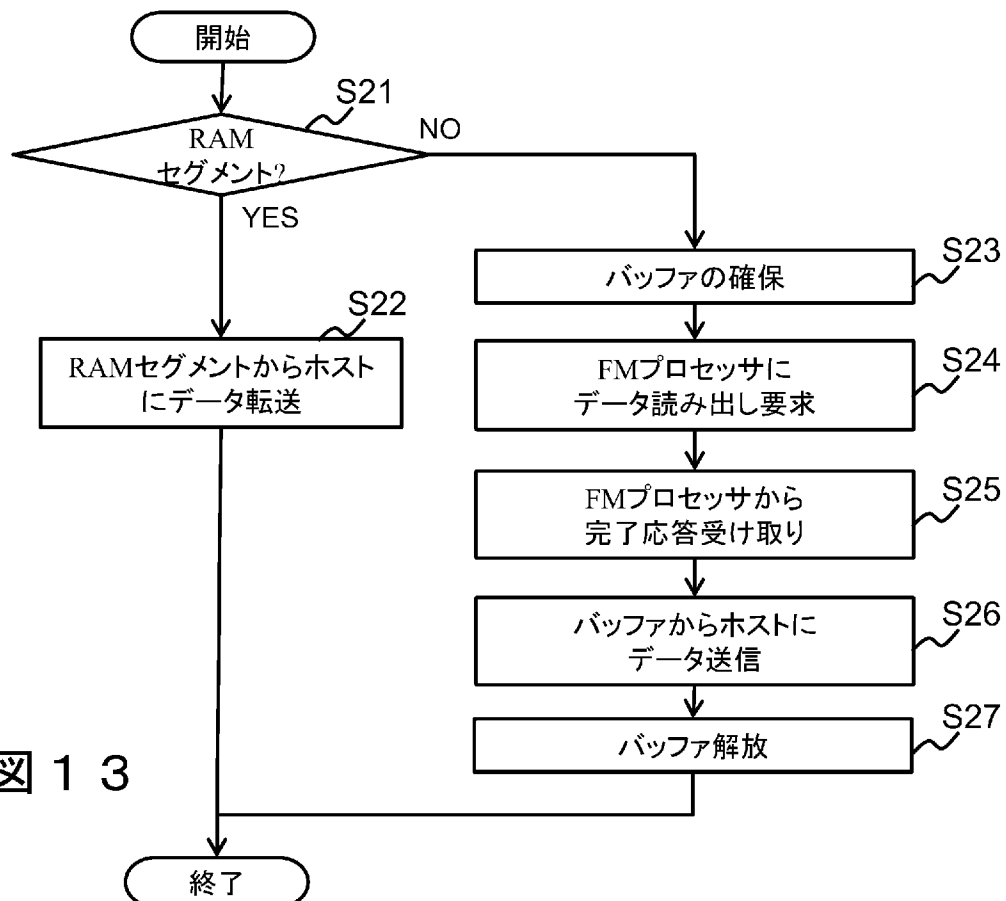


図 1 3

[図14]

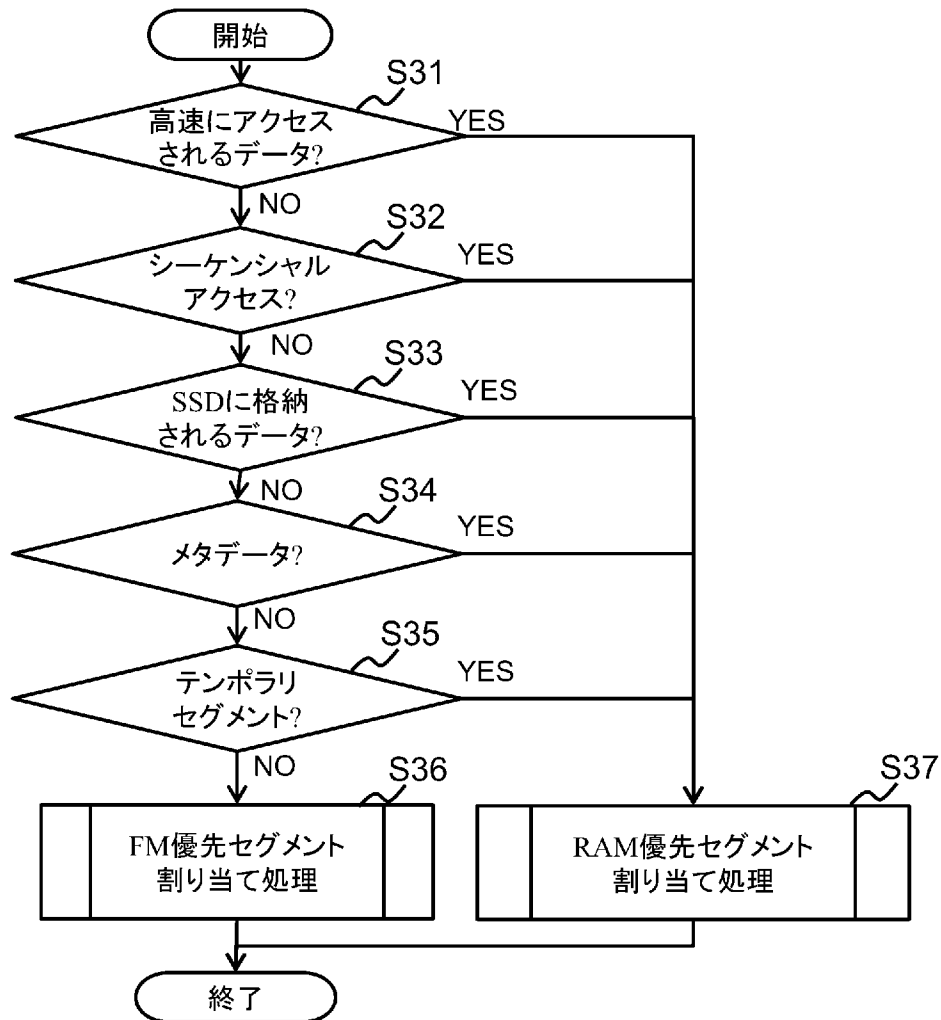


図 1 4

[図15]

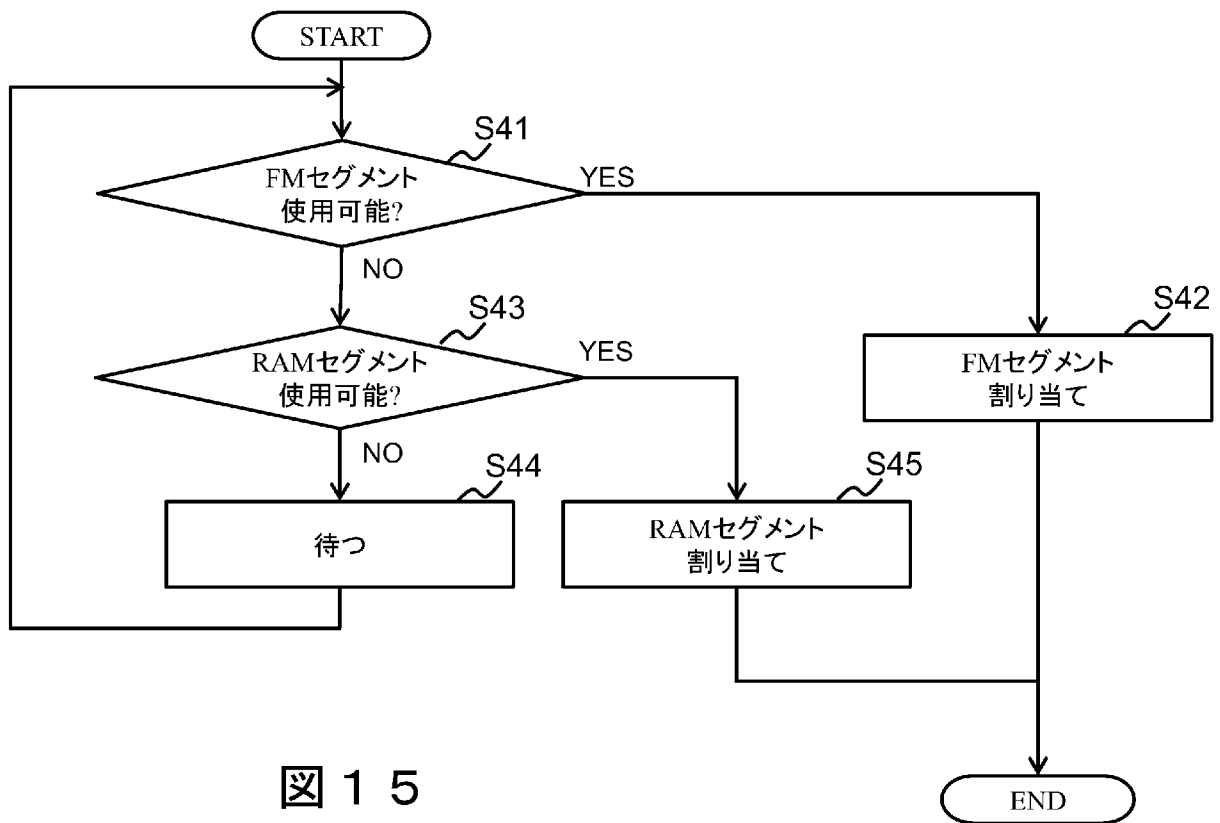


図 1 5

[図16]

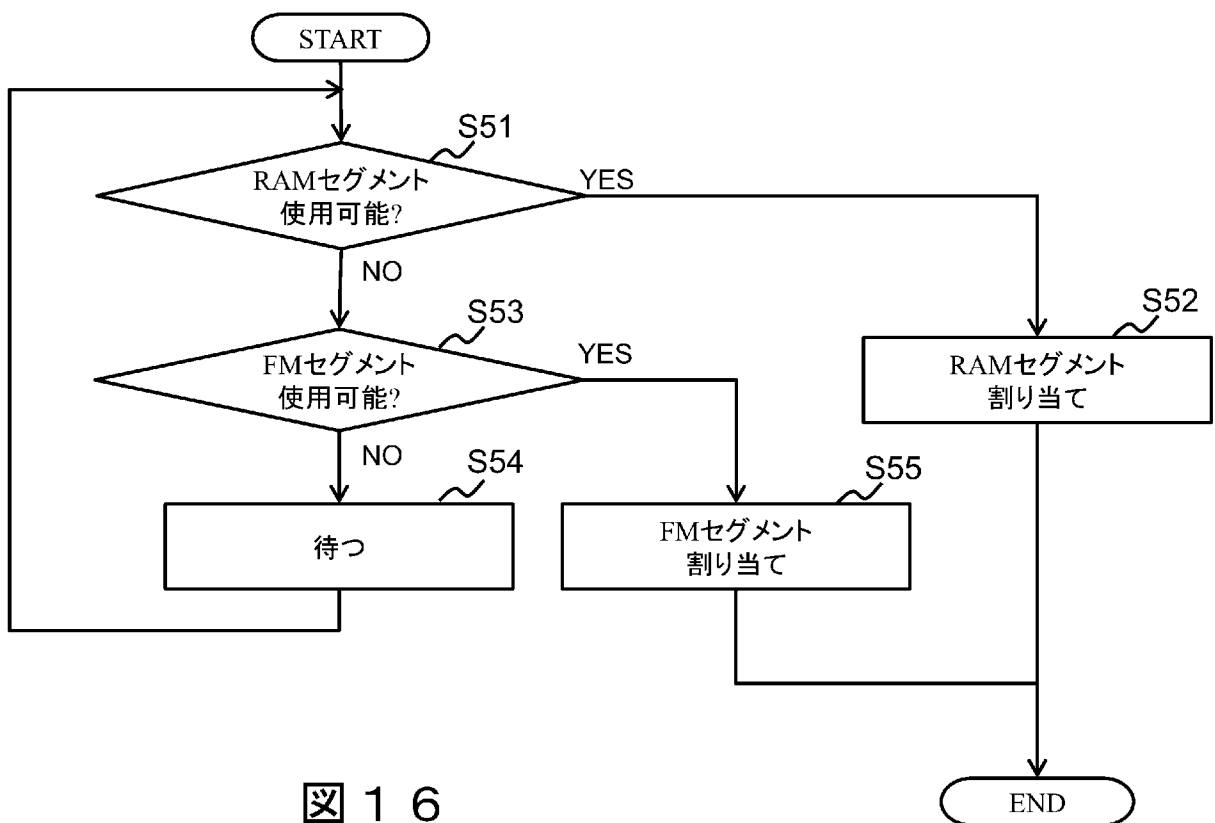


図 1 6

[図17]

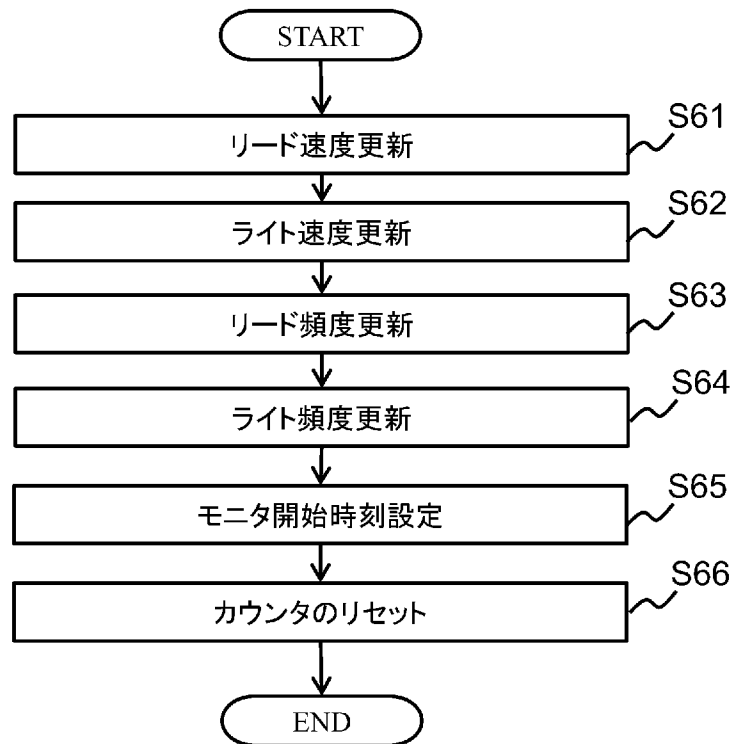


図 1 7

[図18]

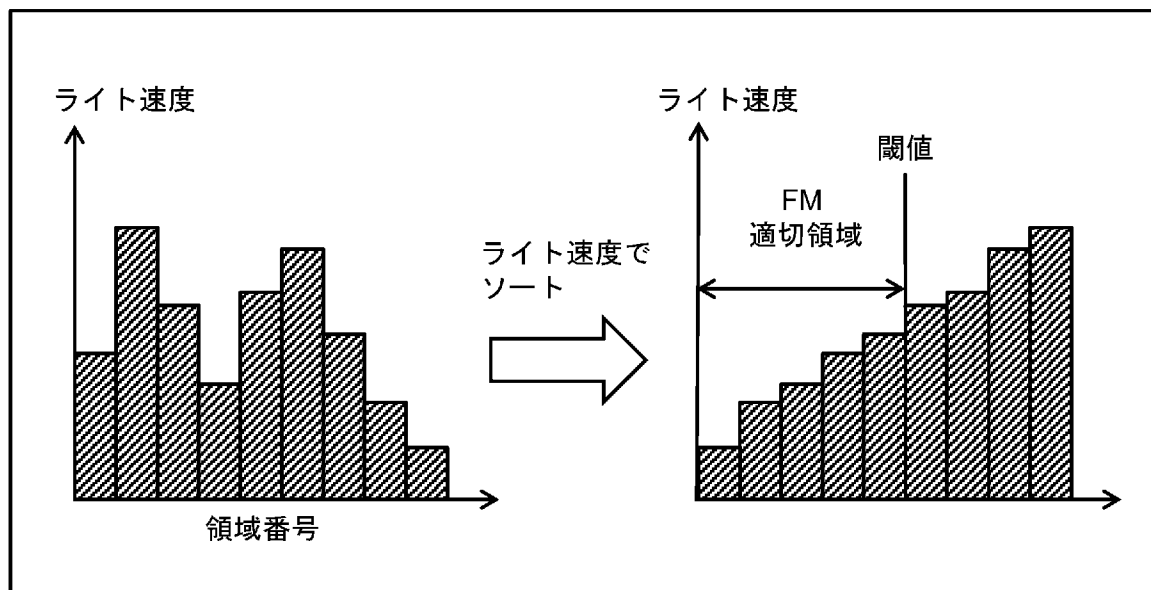


図 1 8

[図19]

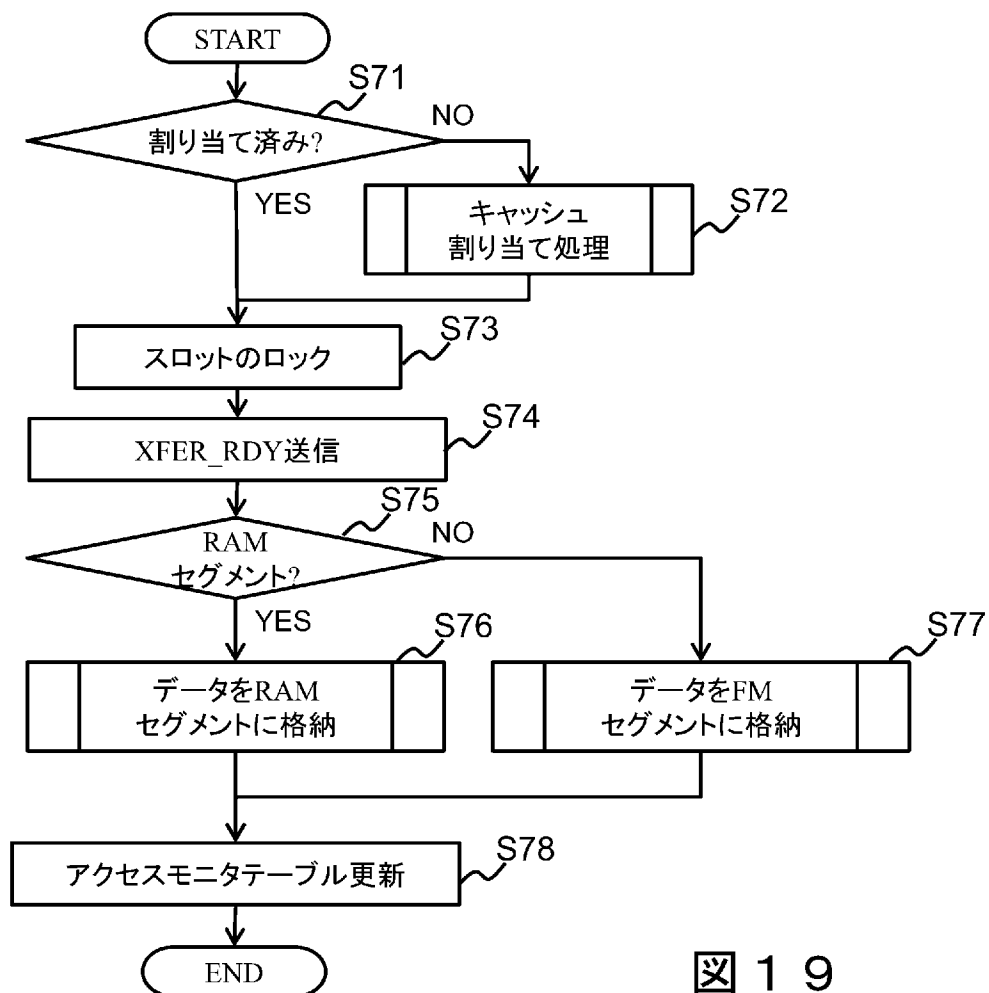


図 19

[図20]

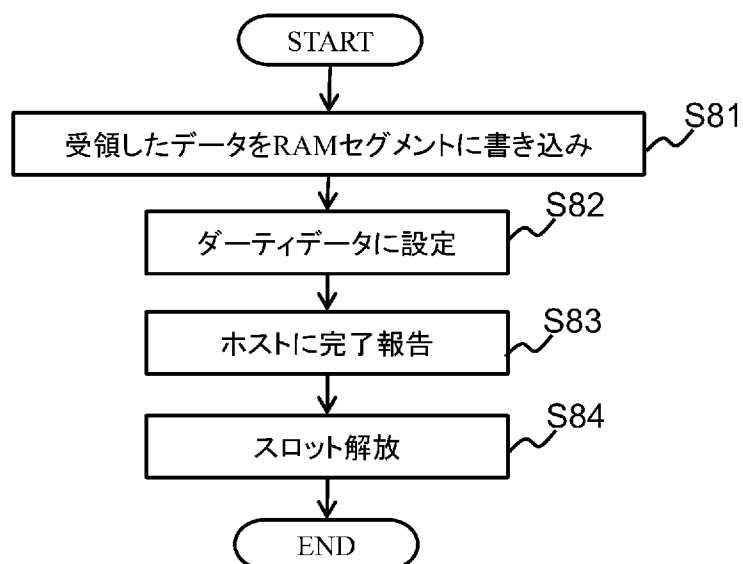


図 20

[図21]

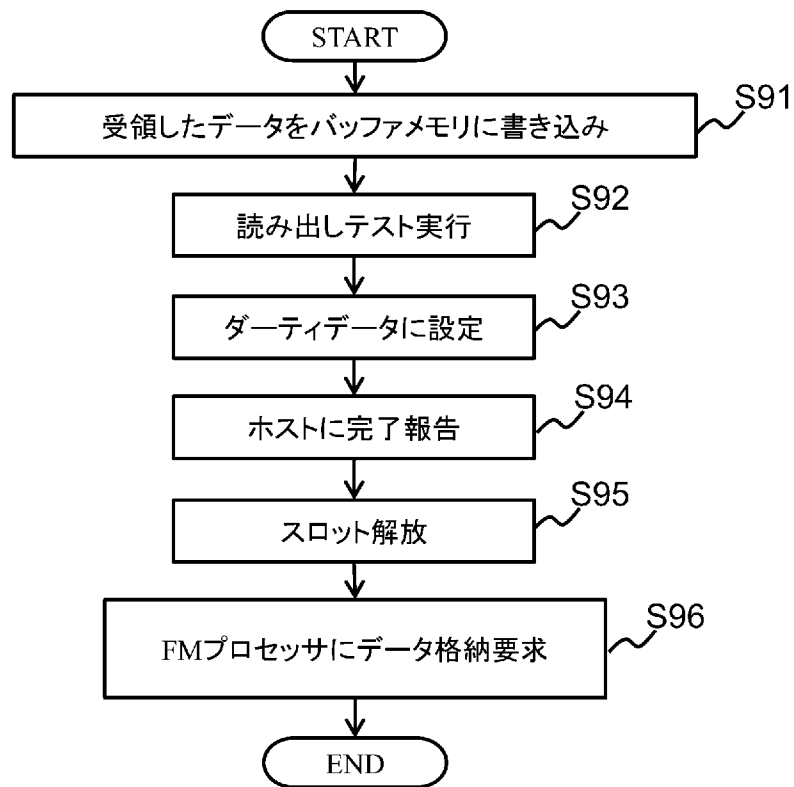


図 2 1

[図22]

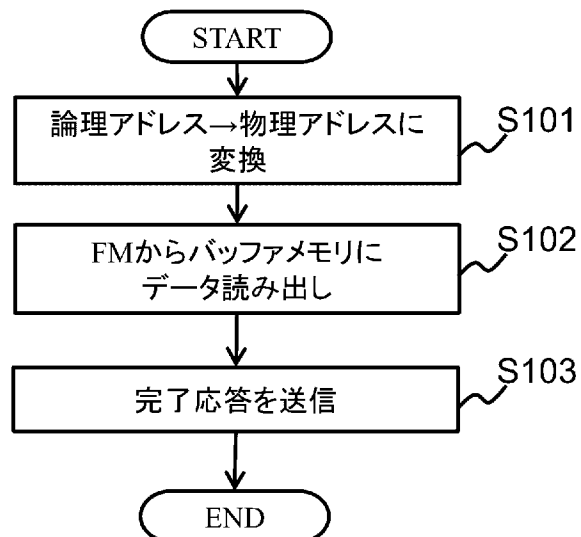


図 2 2

[図23]

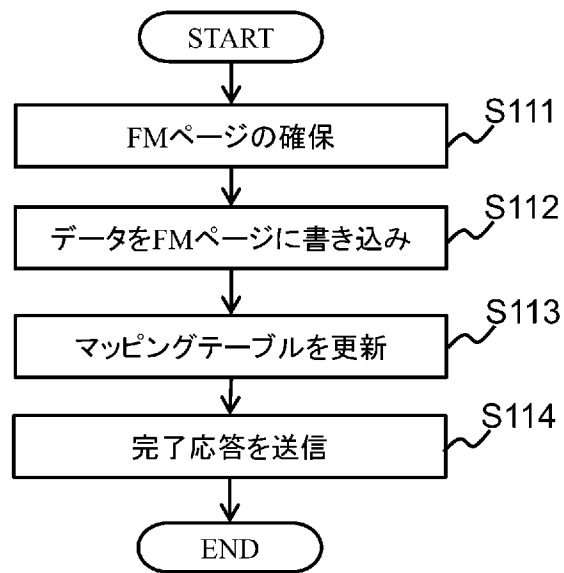


図 2 3

[図24]

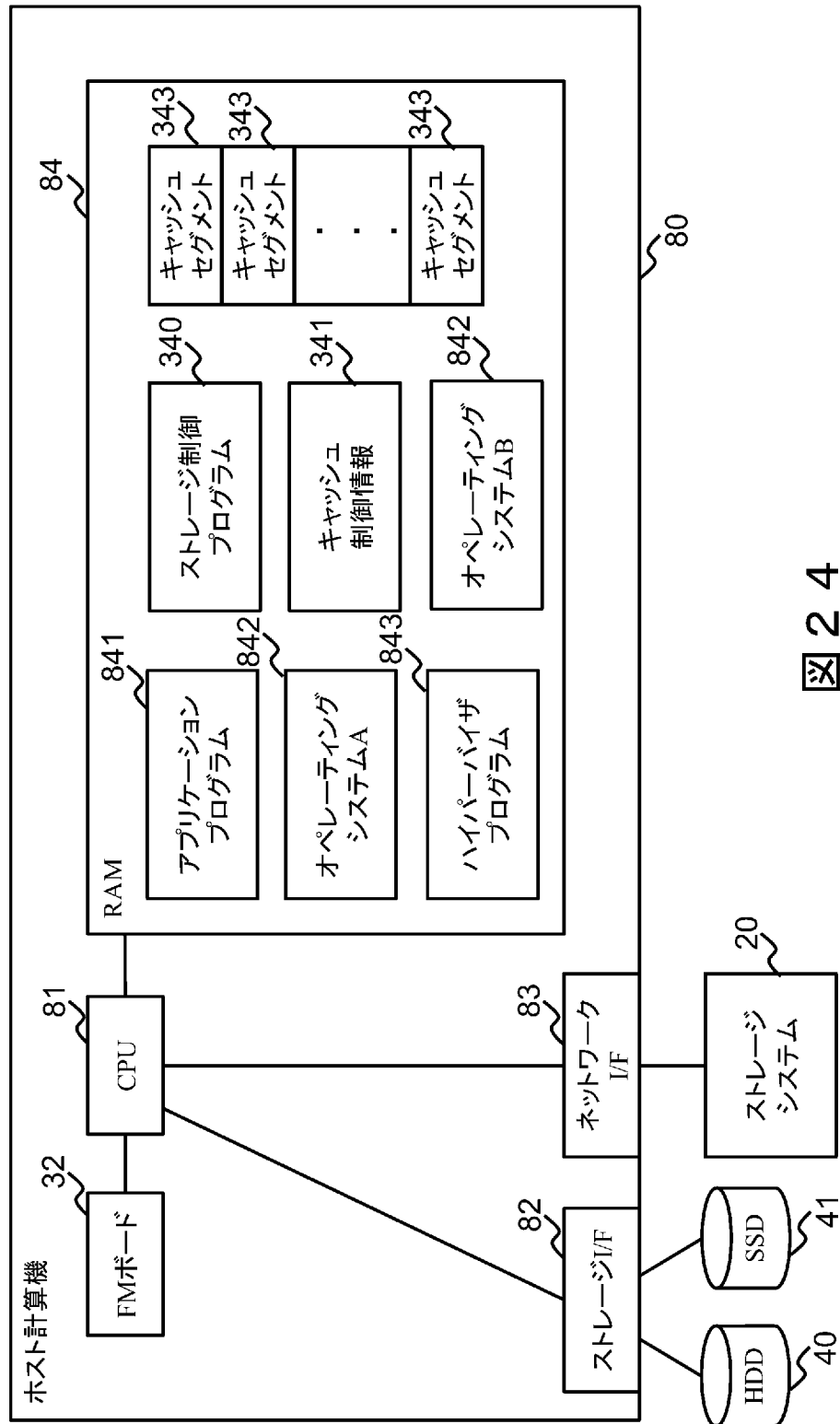


図24

[図25]

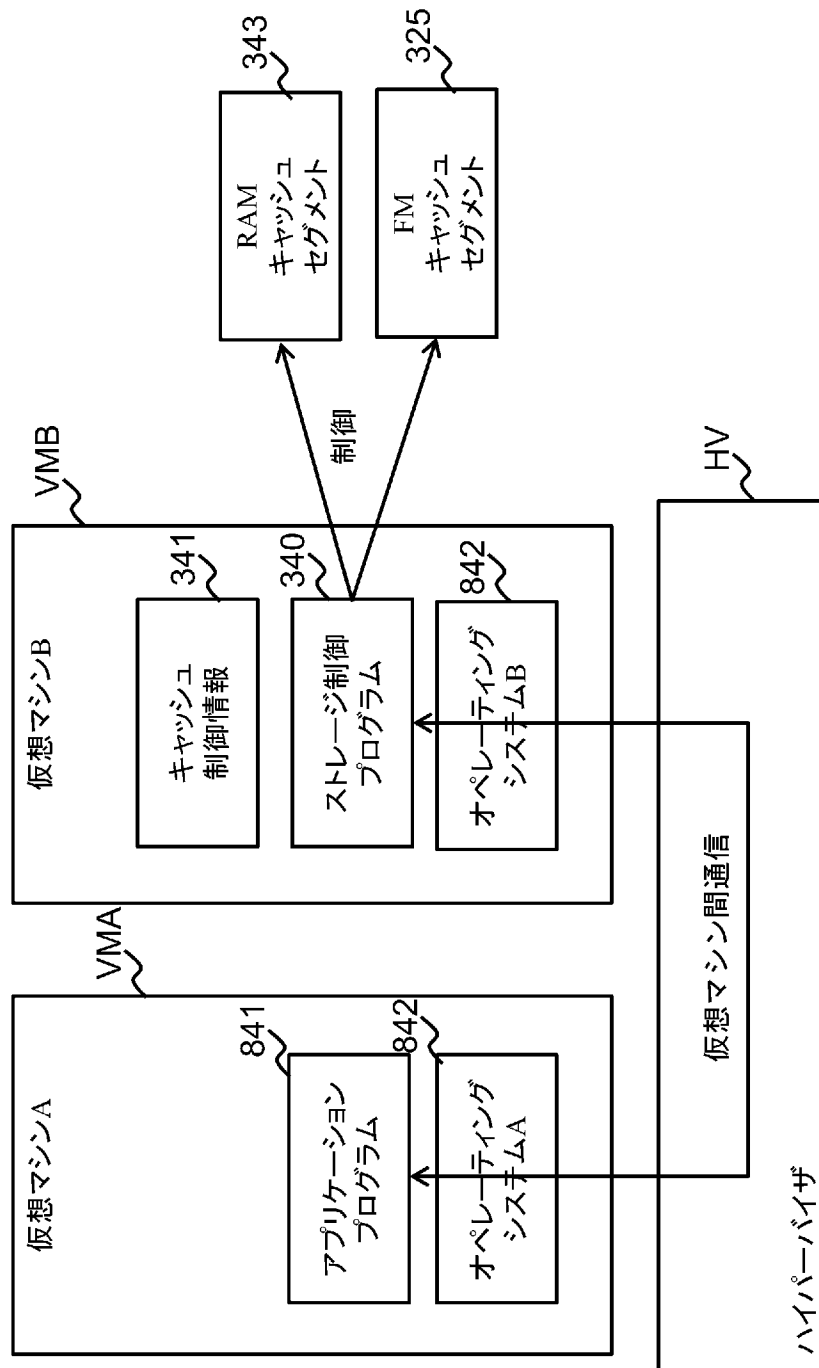


図 2 5

[図26]

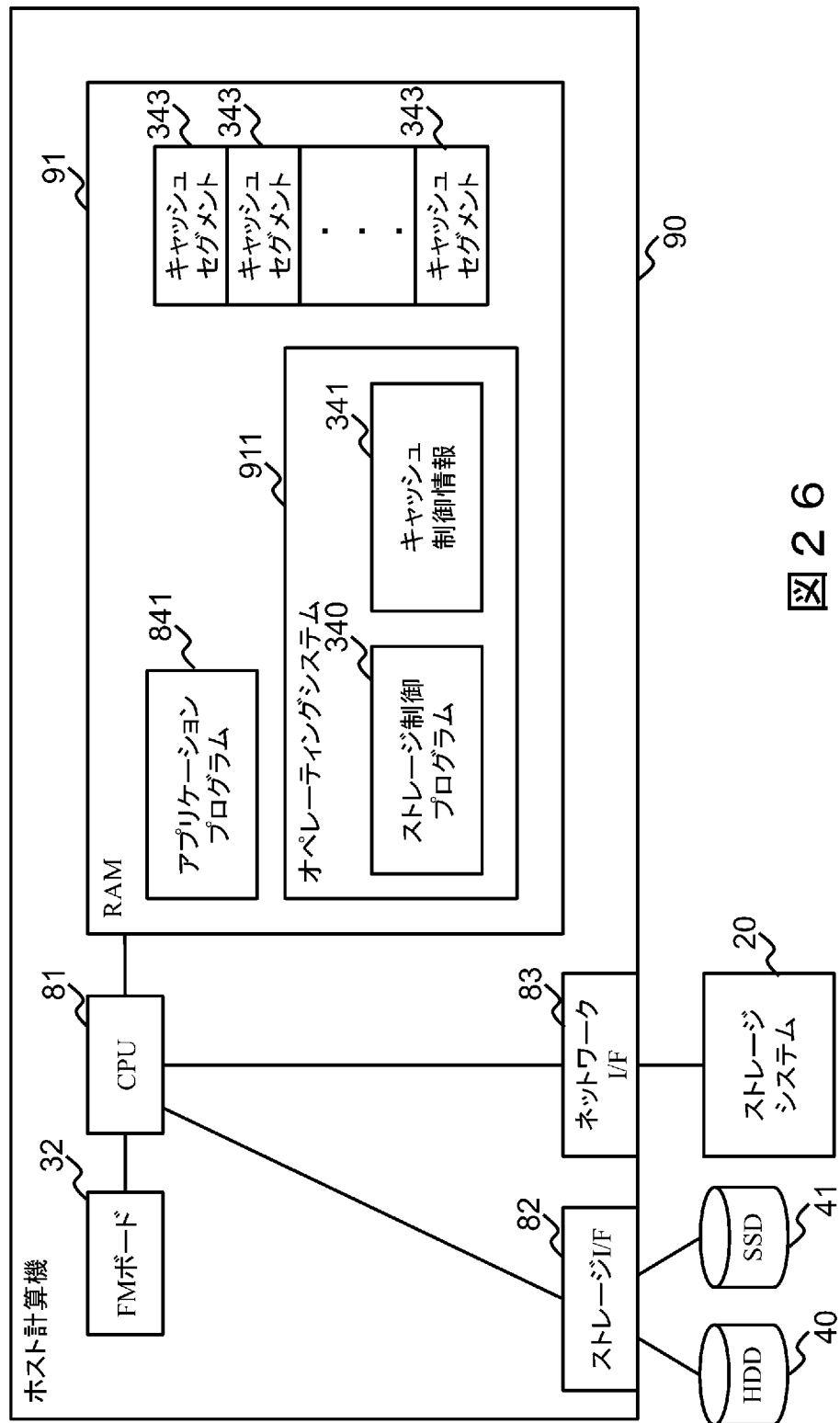


図 26

[図27]

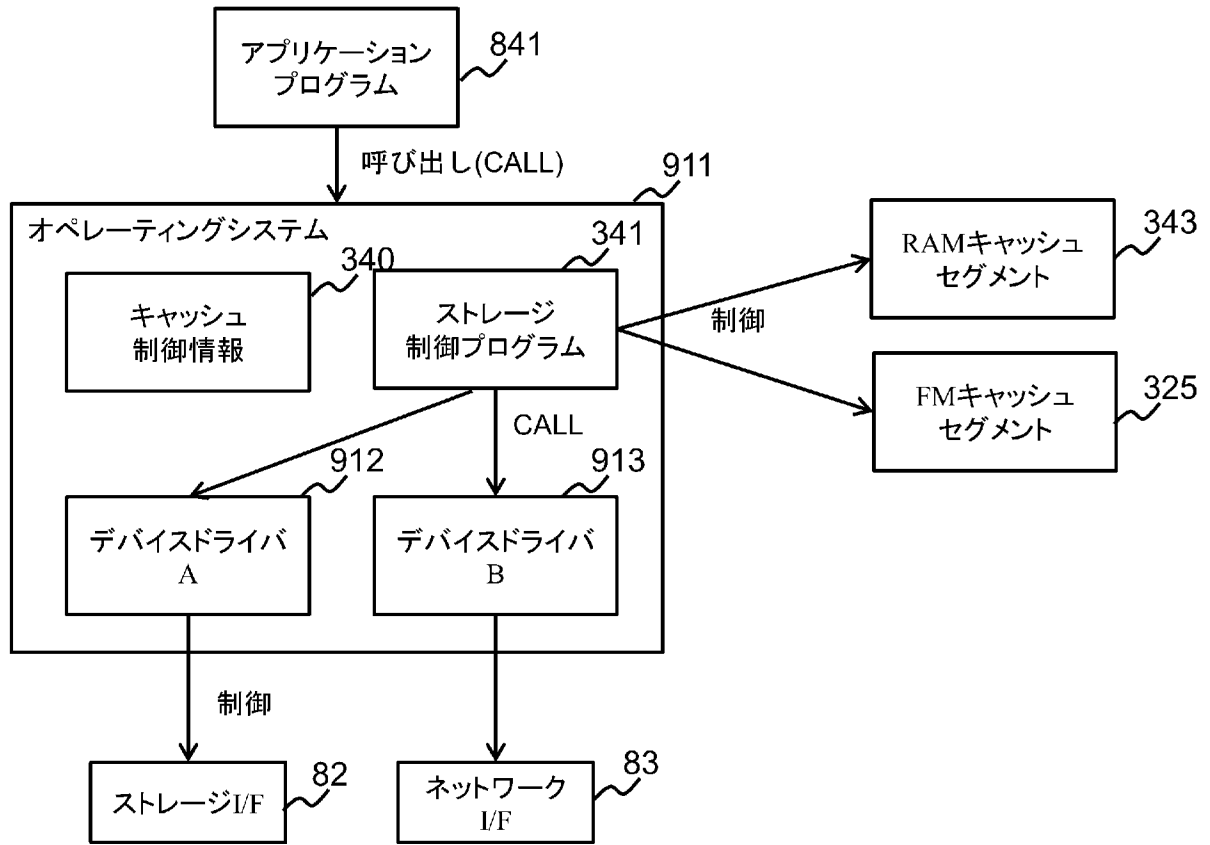


図 2 7

[図28]

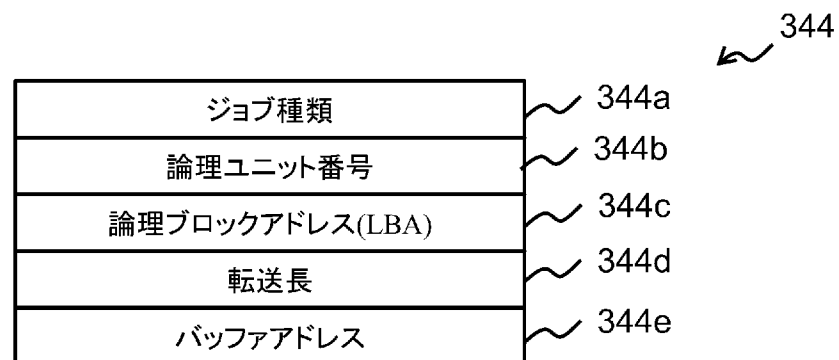


図 2 8

[図29]

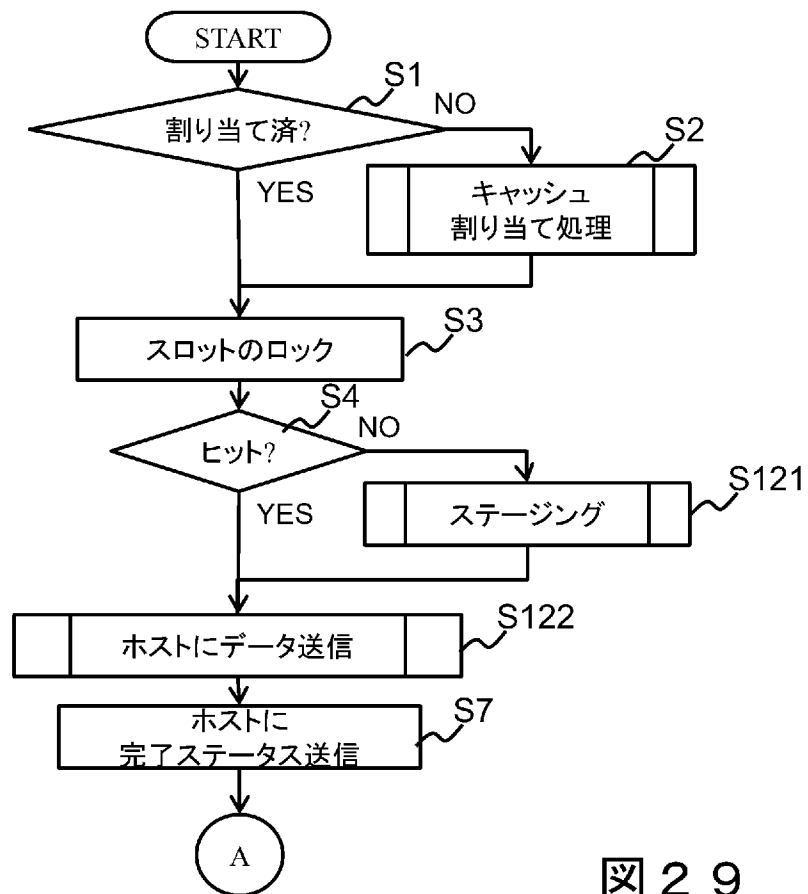


図 2 9

[図30]

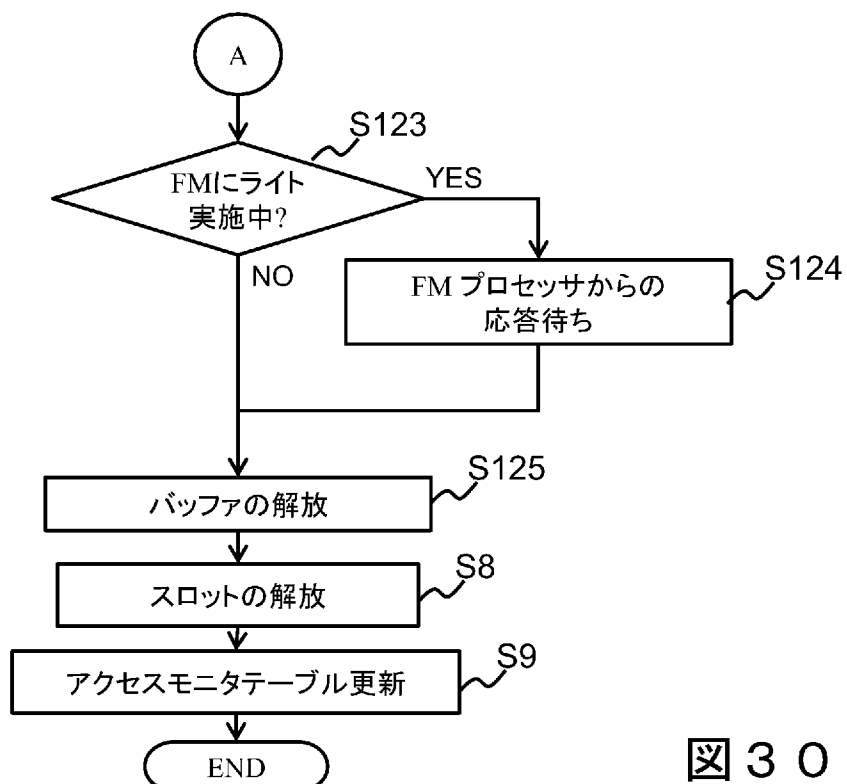


図 3 0

[図31]

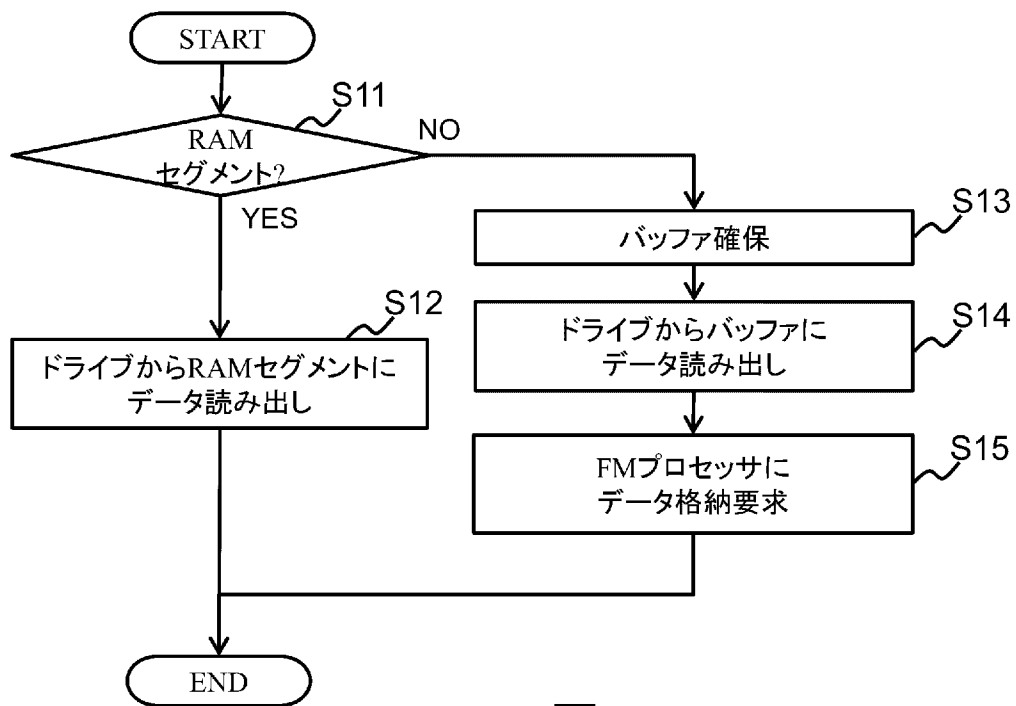


図 3 1

[図32]

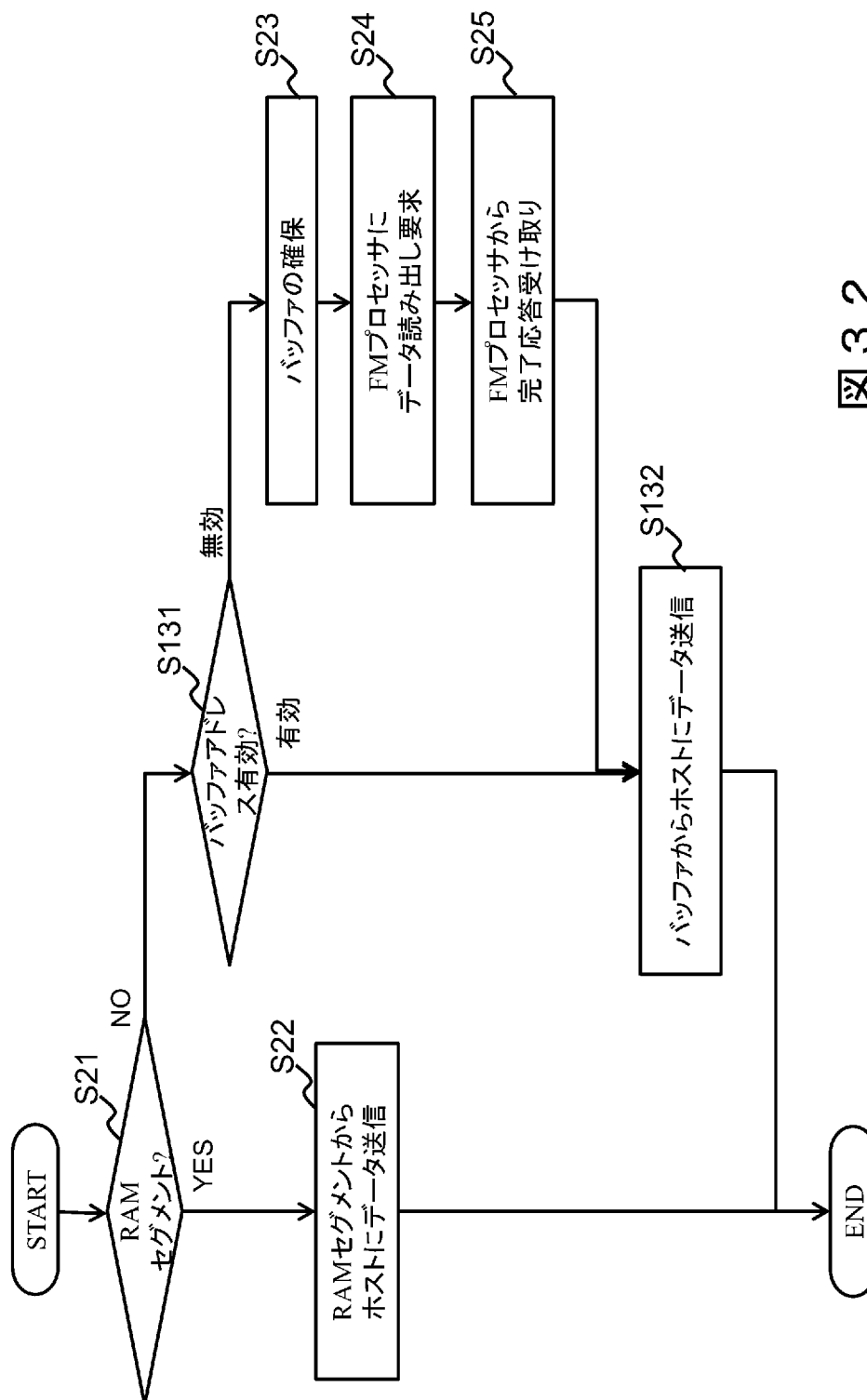


図 3 2

[図33]

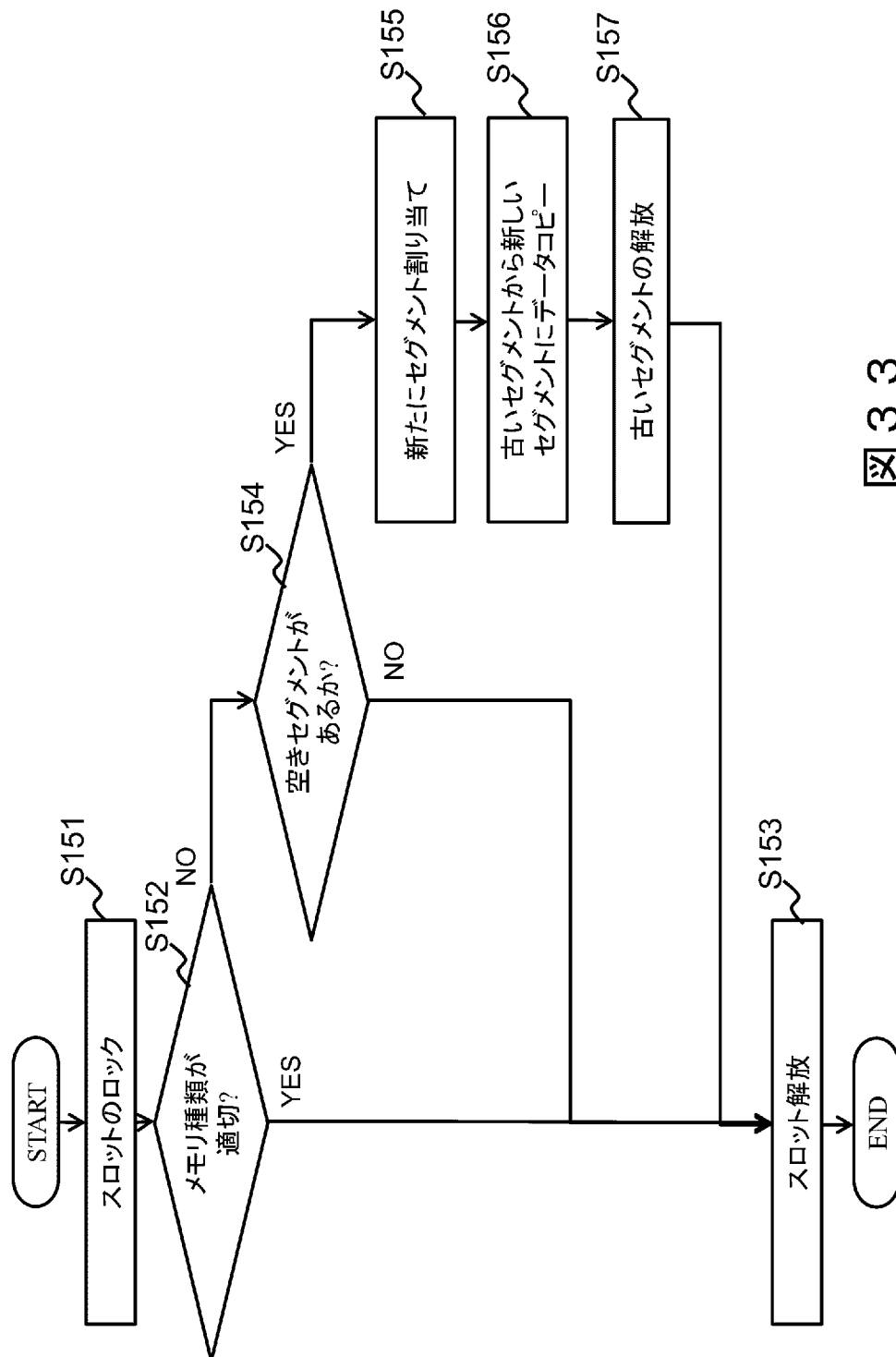


図 3 3

[図34]

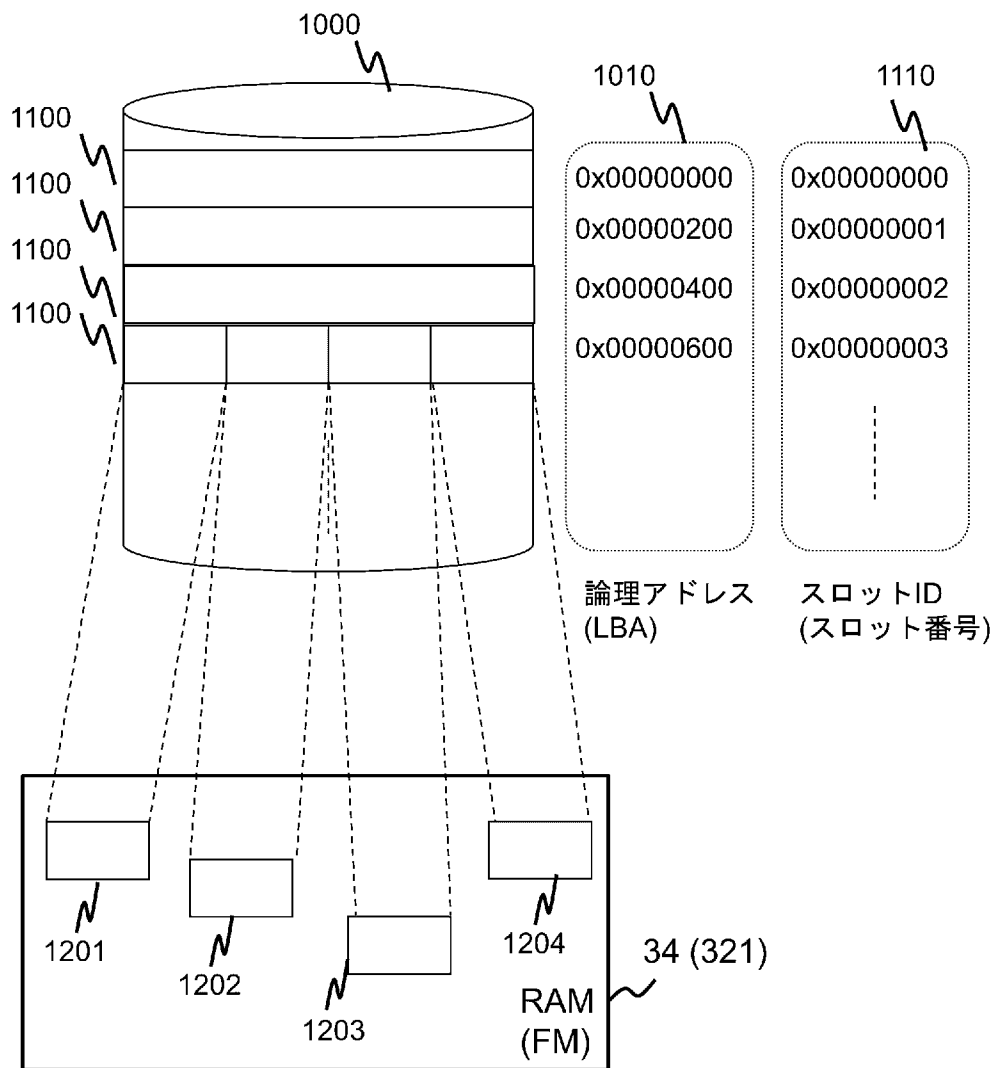


図 3 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2013/078731

A. CLASSIFICATION OF SUBJECT MATTER

G06F12/08(2006.01)i, G06F3/06(2006.01)i, G06F12/16(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F12/08, G06F3/06, G06F12/16

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2014
Kokai Jitsuyo Shinan Koho	1971-2014	Toroku Jitsuyo Shinan Koho	1994-2014

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2009-205335 A (Hitachi, Ltd.), 10 September 2009 (10.09.2009), paragraphs [0009] to [0082], [0171]	1-3, 6, 7, 9, 11-14
Y	& US 2009/0216945 A1	4, 8, 10, 15
A		5
X	JP 2003-076498 A (Hitachi, Ltd.), 14 March 2003 (14.03.2003), paragraphs [0005] to [0014]	1, 2, 13 4, 8, 15
Y	& US 2003/0046489 A1	3, 5-7, 9-12, 14
A		
Y	JP 2011-022933 A (Toshiba Corp.), 03 February 2011 (03.02.2011), paragraphs [0058] to [0061], [0265]; fig. 2, 20, 23 & US 2010/0064111 A1 & EP 2455865 A1 & WO 2011/007599 A1 & CN 101673245 A & KR 10-2010-0030602 A	10

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
08 January, 2014 (08.01.14)

Date of mailing of the international search report
21 January, 2014 (21.01.14)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F12/08(2006.01)i, G06F3/06(2006.01)i, G06F12/16(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06F12/08, G06F3/06, G06F12/16

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 4 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 4 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 4 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X Y A	JP 2009-205335 A（株式会社日立製作所）2009.09.10, 【0009】－【0082】、【0171】段落 & US 2009/0216945 A1	1-3, 6, 7, 9, 11-14 4, 8, 10, 15 5
X Y A	JP 2003-076498 A（株式会社日立製作所）2003.03.14, 【0005】－【0014】段落 & US 2003/0046489 A1	1, 2, 13 4, 8, 15 3, 5-7, 9-12, 14

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

0 8 . 0 1 . 2 0 1 4

国際調査報告の発送日

2 1 . 0 1 . 2 0 1 4

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）
郵便番号100-8915
東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

滝谷 亮一

5 U

3 1 3 5

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 6 5

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y	JP 2011-022933 A (株式会社東芝) 2011.02.03, 【0058】－【0061】、【0265】、第2図、第20図、 23図 & US 2010/0064111 A1 & EP 2455865 A1 & WO 2011/007599 A1 & CN 101673245 A & KR 10-2010-0030602 A	10