

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5058060号  
(P5058060)

(45) 発行日 平成24年10月24日(2012.10.24)

(24) 登録日 平成24年8月10日(2012.8.10)

(51) Int.Cl.

F 1

G 11 B 5/09 (2006.01)  
G 11 B 20/18 (2006.01)G 11 B 5/09 361G  
G 11 B 5/09 361Z  
G 11 B 20/18 552E  
G 11 B 20/18 572B  
G 11 B 20/18 572F

請求項の数 6 (全 21 頁) 最終頁に続く

(21) 出願番号

特願2008-116955 (P2008-116955)

(22) 出願日

平成20年4月28日 (2008.4.28)

(65) 公開番号

特開2009-266333 (P2009-266333A)

(43) 公開日

平成21年11月12日 (2009.11.12)

審査請求日

平成22年9月6日 (2010.9.6)

(73) 特許権者 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(74) 代理人 100089118

弁理士 酒井 宏明

(72) 発明者 小川 兼司

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

審査官 井上 和俊

最終頁に続く

(54) 【発明の名称】データ記憶装置及び隣接トラック書き直し処理方法。

## (57) 【特許請求の範囲】

## 【請求項1】

回転する記憶媒体に、データのリード及びライトを行うヘッドと、  
前記記憶媒体の所望トラックに前記ヘッドを位置付けるアクチュエータと、  
上位からのライトコマンド及びライトデータを受領し、前記ライトデータを、前記ヘッド  
により、前記記憶媒体の指定されたトラックにライトするとともに、前記ヘッドによる  
前記トラックへのライト時に、前記ヘッドが、前記トラックから外れたオフトラックを検  
出し、前記ライトデータの前記トラックへのライトリトライを実行する制御回路と、  
少なくとも、対象トラック番号と、前記オフトラック時のヘッドのずれ量とを有する書  
き直し要求を格納する書き直し要求テーブルとを有し、

前記制御回路は、前記オフトラック検出時に、前記ずれ量が所定の閾値より大きいか否  
かを判定することにより、前記ヘッドのオフトラックによる隣接トラックのデータへの影  
響の可能性があるか否かを判定し、前記隣接トラックのデータへの影響の可能性があると  
判定した場合には、前記隣接トラックの書き直し要求を、前記書き直し要求テーブルに格  
納し、前記ずれ量に基づき、前記ライトコマンドとは異なるタイミングであるバックグラ  
ウンドで、振動が収まった後に、前記書き直し要求テーブルに格納された前記書き直し要  
求を実行する

ことを特徴とするデータ記憶装置。

## 【請求項2】

前記制御回路は、他のライトコマンドを受領した時に、前記他のライトコマンドの実行

前に、前記書き直し要求テーブルに格納された前記書き直し要求を実行することを特徴とする請求項1のデータ記憶装置。

【請求項3】

前記制御回路は、前記オフトラック要因である振動が収まったかを判定し、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記書き直し要求を実行する

ことを特徴とする請求項1のデータ記憶装置。

【請求項4】

回転する記憶媒体の所望のトラックに、データのリード及びライトを行うヘッドを位置付け、前記データのリード及びライトを行うデータ記憶装置の隣接トラック書き直し処理方法において、

前記ヘッドによる前記トラックへのライト時に、前記ヘッドが、前記トラックから外れたオフトラックを検出するステップと、

前記オフトラック検出時に、前記オフトラック時のヘッドのずれ量が所定の閾値より大きいか否かを判定することにより、前記ヘッドのオフトラックによる隣接トラックのデータへの影響の可能性があるか否かを判定するステップと、

前記隣接トラックのデータへの影響の可能性があると判定した場合には、少なくとも、対象トラック番号と、前記ヘッドのずれ量とを有する前記隣接トラックの書き直し要求を、書き直し要求テーブルに格納するステップと、

前記ライトデータの前記トラックへのライトリトライを実行するステップと、

前記ずれ量に基づき、前記ライトコマンドとは異なるタイミングであるバックグラウンドで、振動が収まった後に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップとを有する

ことを特徴とする隣接トラック書き直し処理方法。

【請求項5】

前記書き直し要求を実行するステップは、他のライトコマンドを受領した時に、前記他のライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップを有する

ことを特徴とする請求項4の隣接トラック書き直し処理方法。

【請求項6】

前記書き直し要求を実行するステップは、

前記オフトラック要因である振動が収まったかを判定するステップと、

前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップとを有する

ことを特徴とする請求項4の隣接トラック書き直し処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、上位からのライトデータを、ヘッドにより、媒体に記録するデータ記憶装置及び隣接トラック書き直し処理方法に関し、特に、媒体へのライトデータ記録時に、オフトラックが発生した場合に、隣接トラックのライトデータをリライトするデータ記憶装置及び隣接トラック書き直し処理方法に関する。

【背景技術】

【0002】

近年のデータ電子化処理の要求により、データを記憶する磁気ディスク装置や光ディスク装置の媒体記憶装置に、大容量化が要求されている。このため、ディスク媒体は、益々トラック密度や記録密度が高くなっている。

【0003】

このようなデータ記憶装置は、ディスクへのデータライト時に、何らかの理由で、ヘッドが、オフトラックしたライトエラーが発生することがある。即ち、記録媒体への書き込

10

20

30

40

50

み時、書き込みヘッドの位置が目標トラックのセンターからどれだけずれているかを検出し、このずれ量が設定した閾値を超えた際に、その書き込みをオフトラックエラーと判断し、リトライを行っている。

#### 【0004】

この時、ヘッド位置のずれ量が、大きすぎると、隣接するトラックに寄せ書きした可能性がある。このため、オフトラックエラーの判断基準に加えて、隣接トラックへ寄せ書きした可能性があるかの判断基準を設けて、隣接トラックへ寄せ書きした可能性がある場合には、当該ライトコマンドのリトライ（ライトコマンドのトラックのデータライト）と同時に、寄せ書きされたトラックのリード及び書き直しを実施していた（特許文献1, 2, 3参照）。即ち、従来は、隣接トラックへの書き直しを、当該ライトコマンドのリトライに続けて行っていた。

【特許文献1】特許第3, 942, 483号公報

【特許文献2】特許第3, 037, 250号公報

【特許文献2】特開2001-014606号公報

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0005】

近年のディスク記憶装置の記録密度の向上に伴い、トラックピッチも狭くなっている。このため、ヘッドが、オフトラックする程度が増加しつつある。特に、トラックピッチが狭くなると、振動による影響により、オフトラックする可能性が大きくなる。

#### 【0006】

このため、従来のように、隣接トラックへの書き直しを、当該ライトコマンドのリトライに続けて行う場合、当該ライトコマンドで発生したオフトラックエラーの要因となる振動の影響を受ける可能性がある。このため、書き直し処理実施時にも、オフトラックエラーに至る要素が存在することになり、最悪の場合さらなる寄せ書きが発生することになりうる。

#### 【0007】

また、即時の書き直し処理の実施は、当該ライトコマンドの実行時間の延伸に繋がり、ドライブの性能低下の要因となりうる。

#### 【0008】

従って、本発明の目的は、オフトラックにより、隣接トラックへライトを行った可能性があった場合、隣接トラックのデータを任意のタイミングで再書き込みするためのデータ記憶装置及び隣接トラック書き直し処理方法を提供することにある。

#### 【0009】

又、本発明の他の目的は、オフトラックにより、隣接トラックへライトを行った可能性があった場合に、ライトリトライとは、別のタイミングで、隣接トラックのデータを再書き込みするためのデータ記憶装置及び隣接トラック書き直し処理方法を提供することにある。

#### 【課題を解決するための手段】

#### 【0010】

この目的の達成のため、本発明のデータ記憶装置は、回転する記憶媒体に、データのリード及びライトを行うヘッドと、前記記憶媒体の所望トラックに前記ヘッドを位置付けるアクチュエータと、上位からのライトコマンド及びライトデータを受領し、前記ライトデータを、前記ヘッドにより、前記記憶媒体の指定されたトラックにライトするとともに、前記ヘッドによる前記トラックへのライト時に、前記ヘッドが、前記トラックから外れたオフトラックを検出し、前記ライトデータの前記トラックへのライトリトライを実行する制御回路と、少なくとも、対象トラック番号と、前記オフトラック時のヘッドのずれ量とを有する書き直し要求を格納する書き直し要求テーブルとを有し、前記制御回路は、前記オフトラック検出時に、前記ずれ量が所定の閾値より大きいか否かを判定することにより、前記ヘッドのオフトラックによる隣接トラックのデータへの影響の可能性があるか否か

10

20

30

40

50

を判定し、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックの書き直し要求を、前記書き直し要求テーブルに格納し、前記ずれ量に基づき、前記ライトコマンドとは異なるタイミングであるバックグラウンドで、振動が収まった後に、前記書き直し要求テーブルに格納された前記書き直し要求を実行する。

【0011】

又、本発明の隣接トラック書き直し処理方法は、回転する記憶媒体の所望のトラックに、データのリード及びライトを行うヘッドを位置付け、前記データのリード及びライトを行うデータ記憶装置の隣接トラック書き直し処理方法において、前記ヘッドによる前記トラックへのライト時に、前記ヘッドが、前記トラックから外れたオフトラックを検出するステップと、前記オフトラック検出時に、前記オフトラック時のヘッドのずれ量が所定の閾値より大きいか否かを判定することにより、前記ヘッドのオフトラックによる隣接トラックのデータへの影響の可能性があるか否かを判定するステップと、前記隣接トラックのデータへの影響の可能性があると判定した場合には、少なくとも、対象トラック番号と、前記ヘッドのずれ量とを有する前記隣接トラックの書き直し要求を、書き直し要求テーブルに格納するステップと、前記ライトデータの前記トラックへのライトリトライを実行するステップと、前記ずれ量に基づき、前記ライトコマンドとは異なるタイミングであるバックグラウンドで、振動が収まった後に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップとを有する。

10

20

【0012】

更に、本発明は、好ましくは、前記制御回路は、他のライトコマンドを受領した時に、前記他のライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記書き直し要求を実行する。

【0013】

更に、本発明は、好ましくは、前記制御回路は、前記オフトラック要因である振動が収まったかを判定し、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記書き直し要求を実行する。

【0014】

30

更に、本発明は、好ましくは、前記制御回路は、前記他のライトコマンドを受領した時に、前記ライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記ライトコマンドで指定されたトラックに隣接するトラックに対する前記書き直し要求を実行する。

【0015】

更に、本発明は、好ましくは、前記制御回路は、前記他のライトコマンドを受領した時に、前記書き直し要求テーブルに、前記ライトコマンドで指定されたトラックに隣接するトラックに対する前記書き直し要求が存在するかを判定し、前記隣接トラックの書き直し要求が存在する場合に、前記隣接トラックの書き直し要求を実行し、前記隣接トラックの書き直し要求が存在しない場合には、前記オフトラック要因である振動が収まったかを判定し、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記隣接トラック以外の書き直し要求を実行する。

40

【0016】

更に、本発明は、好ましくは、前記制御回路は、他のライトコマンドを受領した時に、前記書き直し要求テーブルの前記書き直し要求の書き直し領域が、前記他のライトコマンドのライト範囲内であるかを判定し、前記書き直し要求の書き直し領域が、前記他のライトコマンドのライト範囲内であるとの判定に応じて、前記書き直し要求テーブルに格納された前記書き直し要求をリセットする。

【0017】

更に、本発明は、好ましくは、前記書き直し要求テーブルは、少なくとも、対象トラック

50

ク番号と、前記オフトラック時のヘッドのずれ量とを有する複数の書き直し要求を格納し、前記制御回路は、前記隣接トラックのデータへの影響の可能性があると判定した場合に、前記隣接トラックの書き直し要求の前記ヘッドのずれ量と、前記書き直し要求テーブルに格納された前記書き直し要求のずれ量とを比較し、前記ずれ量が最小である前記書き直し要求を、前記書き直し要求テーブルから削除する。

【0018】

更に、本発明は、好ましくは、前記制御回路は、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックが、前記オフトラックを発生した前記ライトコマンドの対象範囲内であるかを判定し、前記隣接トラックが、前記ライトコマンドの対象範囲内であることの判定に応じて、前記隣接トラックの書き直し要求を、前記書き直し要求テーブルへの格納を禁止する。

10

【0019】

更に、本発明は、好ましくは、前記制御回路は、前記ライトコマンドを受領していないことを判定し、前記ライトコマンドを受領していない場合に、前記書き直し要求テーブルに格納された前記書き直し要求を実行する。

【0020】

更に、本発明は、好ましくは、前記ライトデータを一時格納するキャッシュメモリを更に有し、前記制御回路は、前記書き直し要求テーブルの前記書き直し要求の対象データが前記キャッシュメモリに存在するかを判定し、前記書き直し要求の対象データが、前記キャッシュメモリに存在する場合に、前記キャッシュメモリの対象データを前記隣接トラックにライトして、前記書き直し要求を実行する。

20

【0021】

更に、本発明は、好ましくは、前記制御回路は、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックのデータが、キャッシュメモリに存在するかを判定し、存在する場合に、前記キャッシュメモリに存在する識別子を付して、前記書き直し要求を、前記書き直し要求テーブルに格納し、前記ライトリトライとは、別のタイミングで、前記書き直し要求テーブルに格納された前記キャッシュメモリに存在する識別子を有する書き直し要求を優先的に実行する。

【発明の効果】

【0022】

30

ライトコマンドにて隣接トラックのデータに影響を与えるオフトラックエラーが発生した際、当該ライトコマンドに付随して、即時書き直し処理を実施せず、書き直しが必要なトラックの情報やオフトラック時のずれ量等を保持しておき、他のタイミングで保持した書き直し要求を実行するため、オフトラック要因となった振動が収まるのを待ってから書き直し処理を実施し、書き直し処理時のオフトラックエラーの発生、ないし寄せ書きの発生を抑止でき、また書き直し処理をバックグラウンドで行うことで当該ライトコマンドの実行時間の延伸を軽減し、書き直し処理によるドライブの性能低下を軽減させることができる。

【発明を実施するための最良の形態】

【0023】

40

以下、本発明の実施の形態を、データ記憶装置、書き直し処理の第1の実施の形態、書き直し処理の第2の実施の形態、他の実施の形態の順で説明するが、本発明は、この実施の形態に限られない。

【0024】

(データ記憶装置)

図1は、本発明の一実施の形態のデータ記憶装置の構成図であり、図2は、オフトラック検出の説明図である。図1は、データ記憶装置として、磁気ディスクにデータをリード/ライトする磁気ディスク装置(HDD)を例に示す。

【0025】

図1に示すように、磁気ディスク装置10は、パーソナルコンピュータに内蔵され、パ

50

ーソナルコンピュータのホスト(図示せず)とSCSI(Small Component Serial Interface), ATA(AT Attachment)規格等のインターフェースのケーブル9で接続される。

【0026】

磁気ディスク装置10は、磁気ディスク19と、磁気ディスク19を回転するスピンドルモータ20と、磁気ディスク19にデータをリード/ライトする磁気ヘッド25と、磁気ヘッド25を磁気ディスク19の半径方向(トラック横断方向)に移動するアクチュエータ(VCM)22とを備える。

【0027】

又、制御部として、HDC(Hard Disk Controller)26と、データバッファ14と、MPU11と、メモリ(RAM、ROM)13と、リードチャネル回路(RDC)24と、ヘッドIC18と、スピンドルモータドライバ21と、VCMドライバ23と、ショックセンサー(振動センサ)28と、これらを接続するバス17とを備える。

【0028】

HDC26は、ホストからタスクをセットするタスクファイル12Aを有するインターフェース制御回路12と、データバッファ14を制御するデータバッファ制御回路15と、記録データのフォーマット変換し、リードデータの逆フォーマット変換を行うフォーマット変換回路16とを有する。リードチャネル回路24は、リードデータの復調を行うとともに、リードゲート、ライトゲート、リードクロック、ライトクロックの生成を行う。

【0029】

データバッファ14は、キャッシュメモリの役目を果たし、ホストからのライトデータをキャッシュ領域14-1に保存し、且つ磁気ディスク19からのリードデータを保存する。そして、ライトバック時には、データバッファ14のライトデータを、磁気ディスクにライトし、リード時には、データバッファ14のリードデータを、ホストへ転送する。

【0030】

ヘッドIC18は、ライト時は、ライトデータに従い、磁気ヘッド25に記録電流を流し、リード時は、磁気ヘッド25からの読み取り信号を増幅して、リードチャネル回路16に出力する。スピンドルドライバ21は、スピンドルモータ20を回転駆動する。VCMドライバ23は、磁気ヘッド25を移動するVCM22を駆動する。

【0031】

MPU(Micro Processor)11は、磁気ヘッド25の位置検出及び位置制御、リード/ライト制御、リトライ制御を行う。メモリ(ROM/RAM)13は、MPU11の処理に必要なデータを格納する。ショックセンサー28は、振動を検出し、検出信号を、バス17を介し、MPU11に出力する。

【0032】

MPU11は、ヘッドIC18からの磁気ディスク19のサーボ信号から位置を検出し、VCM22をシーク、オントラック制御するとともに、検出位置からオフトラック検出する。ライト時に、オフトラック検出すると、ライトエラーと判定し、そのセクターのライトゲート発生を停止する。メモリ13は、図4以下で説明する書き直し処理要求テーブル13-1を格納する。

【0033】

図2は、本発明の実施の形態のオフトラック検出の説明図である。図2は、磁気ディスク19の対象トラックT0に、磁気ヘッド25のライト素子25-1に位置付け、データをライトしている様子を示す。ライト素子25-1が、対象トラックT0のセンターに位置している時は、正常なデータライトを行うことができる。

【0034】

一方、ライト素子25-1が、対象トラックT0のセンターに位置制御しているにも係らず、磁気ヘッド25が受ける風圧や振動により、ライト素子25-1が、トラックセンターから外れることがある。これを検出するため、MPU11は、オフトラックスライス値を設け、磁気ヘッド25からのサーボ信号の復調結果であるヘッド位置が、オフトラッ

10

20

30

40

50

クスライス値を越えると、ライトオフトラックを検出する。このオフトラックスライス値は、ライト素子 25 - 1 が、対象トラック T0 に、正常にデータを書き込めないと判断できる値に設定される。MPU11 は、オフトラックを検出すると、当該トラック T0 のライトデータのライトリトライを実行する。

【0035】

更に、MPU11 は、オフトラックスライス値より大きい値の重大なオフトラックスライス値を設ける。この重大なオフトラックスライス値は、ライト素子によるデータ書き込みが、隣接トラック T1 のデータに影響する可能性があると判断するためのものである。このため、MPU11 は、磁気ヘッド 25 からのサーボ信号の復調結果であるヘッド位置が、重大なオフトラックスライス値を越えると、隣接トラック T1 のデータに影響する可能性があると判断する。この場合には、隣接トラック T1 のデータの書き直しも実行する。

【0036】

(書き直し処理の第 1 の実施の形態)

図 3 は、本発明のライトリトライ処理の第 1 の実施の形態の処理フロー図、図 4 は、図 1 及び図 3 の書き直し処理要求テーブル 13 - 1 の説明図、図 5 及び図 6 は、図 3 の処理による図 4 のテーブルへの登録動作の説明図である。

【0037】

図 4 により、書き直し処理要求テーブル 13 - 1 を説明する。図 4 に示すように、書き直し処理要求テーブル 13 - 1 は、書き直し処理要求を格納するものであり、エントリ番号と、有効フラグと、トラック番号と、ずれ量とで構成される。有効フラグは、書き直し処理未了のものを、有効、書き直し処理完了済みを、無効とセットされる。トラック番号は、書き直すべきトラック（即ち、前述のヘッド位置が、重大なオフトラックスライス値

を越え、隣接トラック T1 のデータに影響する可能性があると判断された場合のその隣接トラック番号）である。ずれ量は、隣接トラックに影響すると判断した時のトラックセンターからヘッド位置までの距離（ずれ）である。

【0038】

本願は、このような書き直し要求発生時に、この処理テーブル 13 - 1 に、書き直し要求を登録しておき、ライトリトライとは、別の任意のタイミングで、処理テーブル 13 - 1 の書き直し要求を実行するものである。

【0039】

以下、図 5 及び図 6 により、図 3 のライトリトライ処理を説明する。

【0040】

(S10) MPU11 は、図 2 のように、ヘッド位置が、オフトラックスライス値を越えると、ライトオフトラックを検出し、その時のヘッド 25 のトラックセンター位置からのずれ量 new を取得し、ずれ量 new が、図 2 の重大なオフトラックスライス値以上であるかを判定する。MPU11 は、ずれ量 new が、図 2 の重大なオフトラックスライス値以上でない場合には、隣接トラックへの影響の可能性なしと判断し、ステップ S22 の通常のライトリトライを実行する。

【0041】

(S12) MPU11 は、逆に、ずれ量 new が、図 2 の重大なオフトラックスライス値以上である場合には、隣接トラックへの影響の可能性ありと判断し、処理テーブル 13 - 1 への登録処理に移る。先ず、MPU11 は、処理テーブル 13 - 1 の既存エントリが、全て有効であるかを、有効フラグから判定する。

【0042】

(S14) MPU11 は、処理テーブル 13 - 1 の既存エントリが、全てが、有効でない（一部又は全部、無効）と判定すると、図 5 に示すように、無効なエントリの 1 つを、登録用エントリに選択する。そして、ステップ S20 に進む。

【0043】

(S16) MPU11 は、処理テーブル 13 - 1 の既存エントリが、全てが、有効であ

10

20

30

40

50

ると判定すると、処理テーブル 13-1 は、図 6 の状態である。このままでは、新規エントリを登録できない。このため、MPU11 は、処理テーブル 13-1 の全エントリのずれ量 1 ~ M と、今回のずれ量 new の大小関係を比較する。そして、この比較で、今回のずれ量 new が、最小であるかを判断する。MPU11 は、処理テーブル 13-1 の全エントリのずれ量 1 ~ M と、今回のずれ量 new の大小関係の比較により、今回のずれ量 new が、最小であると判断すると、処理テーブル 13-1 への登録を行うことなく、ステップ S22 の通常のライトリトライを実行する。この場合、今回、隣接トラックへの影響の可能性ありと判断された書き直し要求は、実行されない。

#### 【0044】

(S18) 一方、MPU11 は、処理テーブル 13-1 の全エントリのずれ量 1 ~ M と、今回のずれ量 new の大小関係の比較により、今回のずれ量 new が、最小でないと判断すると、処理テーブル 13-1 のずれ量が最小であったエントリを、登録エントリに選択する。 10

#### 【0045】

(S20) MPU11 は、図 5 及び図 6 に示すように、ステップ S14 又は S18 で選択された登録用エントリに、書き直し対象トラック番号 Nnew を、ずれ量に、new を上書きする。そして、処理テーブル 13-1 の登録用エントリの有効フラグを有効にセットする。 20

#### 【0046】

(S22) MPU11 は、オフトラック検出したトラックのライトリトライを実行する。即ち、キャッシュメモリ 14-1 から再度、ライトデータを読み出し、そのトラックにライトする。 20

#### 【0047】

このように、オフトラック検出した際にライトリトライ時には、隣接トラックへの影響の可能性があっても、このライトリトライと同時に、隣接トラックのリライトを実行せずに、処理テーブル 13-1 に、書き直し要求を格納しておく。そして、後述するように、任意のタイミングで、処理テーブル 13-1 を参照して、隣接トラックの書き戻し処理を実行する。

#### 【0048】

又、処理テーブル 13-1 から最小のずれ量のエントリを排除する理由は、ずれ量が小さい場合には、隣接トラックへの影響の可能性があっても、当該隣接トラックのオフセットリード等でデータをリードできる場合があり、それより重大な影響のあるエントリを格納するためである。 30

#### 【0049】

次に、この処理テーブル 13-1 を使用したライトコマンド処理を、図 7 で説明する。

#### 【0050】

(S30) MPU11 は、ライトコマンドを受領すると、そのコマンドを処理する前に、書き戻し処理する。即ち、MPU11 は、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、処理テーブル 13-1 に存在するかを判定する。即ち、ライトコマンドで指定された書き込みトラック番号と、処理テーブル 13-1 の各エントリのトラック番号とを比較して、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、存在するかを判定する。 40

#### 【0051】

(S32) MPU11 は、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、処理テーブル 13-1 に存在すると判定した場合には、当該エントリを実行エントリに選択し、ステップ S38 に進む。

#### 【0052】

(S34) 一方、MPU11 は、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、処理テーブル 13-1 に存在しないと判 50

定すると、処理テーブル 13 - 1 に有効な書き直し要求エントリが存在するかを判定する。存在しない場合には、ステップ S 4 0 に進む。一方、存在する場合には、MPU11 は、当該エントリを実行エントリに選択する。

#### 【0053】

(S36) MPU11 は、ショックセンサー 28 の出力から、直前の重度のライトオフトラック発生時の振動が収まっているかを判定する。MPU11 は、直前の重度のライトオフトラック発生時の振動が収まっていないと判定すると、ステップ S 4 0 に進む。

#### 【0054】

(S38) 一方、MPU11 は、直前の重度のライトオフトラック発生時の振動が収まっていると判定すると、選択した実行エントリの書き戻し処理を実施する。例えば、隣接トラックのデータをリードし、リードしたデータをその隣接トラックに書き戻す。

10

#### 【0055】

(S40) そして、MPU11 は、ライトコマンドで指定するライト範囲内に、対象範囲が全て含まれる有効な書き直し要求エントリが、処理テーブル 13 - 1 に存在するかを判定する。即ち、ライトコマンドで指定された書き込みトラック番号と、処理テーブル 13 - 1 の各エントリのトラック番号とを比較して、ライトコマンドで指定するライト範囲内に、対象範囲が全て含まれる有効な書き直し要求エントリが、存在するかを判定する。存在する場合は、当該ライトコマンドのデータで上書きされるため、そのエントリの有効フラグを無効に変更する。即ち、書き直しは、必要なため、無効とする。

#### 【0056】

(S42) MPU11 は、ライトコマンドを実行する、即ち、ライトコマンドで指定するトラックにライトデータを書き込む。

20

#### 【0057】

このように、リトライしたライトコマンドと別のライトコマンドに同期して、ライトコマンドで指定するトラック領域の隣接トラックを、書き直しし、ライトコマンドを実行する。隣接トラックのため、ライトコマンドの実行時間への影響は、小さい。

#### 【0058】

又、有効エントリも、振動が収まった後、書き直しするため、振動が継続した状態での、書き直しを防止できる。更に、ライトコマンドに範囲に含まれる書き直し領域は、特別の書き直しをせず、ライトコマンドの実行により、隣接トラックも上書きされるため、ライトコマンドの実行時間への影響を防止できる。

30

#### 【0059】

(書き直し処理の第 2 の実施の形態)

図 8 は、本発明のライトリトライ処理の第 2 の実施の形態の処理フロー図、図 9 は、図 1 及び図 8 の書き直し処理要求テーブル 13 - 1 A の説明図、図 10 及び図 11 は、図 8 の処理による図 9 のテーブルへの登録動作の説明図である。この第 2 の実施の形態は、第 1 の実施の形態に、更に、ライトコマンドが到来しない場合に、バックグラウンドで、書き直し処理を実行することと、キャッシュメモリに残っている書き戻しデータは、優先処理し、キャッシュメモリの有効利用を図る機能を追加したものである。

#### 【0060】

40

図 9 により、書き直し処理要求テーブル 13 - 1 A を説明する。図 9 に示すように、書き直し処理要求テーブル 13 - 1 A は、書き直し処理要求を格納するものであり、エントリ番号と、有効フラグと、トラック番号と、ずれ量、キャッシュデータに有無と、キャッシュデータの位置 (アドレス) とで構成される。有効フラグは、トラック番号、ずれ量は、第 1 の実施の形態で説明したものと同一である。キャッシュデータの有無は、書き直すべきトラックのデータが、バッファ 14 のキャッシュメモリ 14 - 1 に存在しているか否かを示す。キャッシュデータの位置は、存在している場合に、そのデータのキャッシュメモリ 14 - 1 の位置 (アドレス) を示す。

#### 【0061】

本願は、このような書き直し要求発生時に、この処理テーブル 13 - 1 に、書き直し要

50

求を登録しておき、ライトリトライとは、別の任意のタイミングで、処理テーブル13-1の書き直し要求を実行するものである。

【0062】

以下、図10及び図11により、図8のライトリトライ処理を説明する。

【0063】

(S50) MPU11は、図2のように、ヘッド位置が、オフトラックスライス値を越えると、ライトオフトラックを検出し、その時のヘッド25のトラックセンター位置からのずれ量 newを取得し、ずれ量 newが、図2の重大なオフトラックスライス値以上であるかを判定する。MPU11は、ずれ量 newが、図2の重大なオフトラックスライス値以上でない場合には、隣接トラックへの影響の可能性なしと判断し、ステップS68の通常のライトリトライを実行する。10

【0064】

(S52) MPU11は、逆に、ずれ量 newが、図2の重大なオフトラックスライス値以上である場合には、隣接トラックへの影響の可能性ありと判断し、処理テーブル13-1への登録処理に移る。MPU11は、この書き直し範囲が、実行中(ライトオフトラックしたライトコマンドの実行中)のコマンドの対象範囲内であるか、又は、未だライトされていない領域であるかを判定する。MPU11は、書き直し範囲が、実行中(ライトオフトラックしたライトコマンドの実行中)のコマンドの対象範囲内である、又は、未だライトされていない領域であると判定すると、書き直しは、必要ないため、ステップS68の通常のライトリトライを実行する。20

【0065】

(S54) MPU11は、書き直し範囲が、実行中(ライトオフトラックしたライトコマンドの実行中)のコマンドの対象範囲内でない、又は、未だライトされていない領域でないと判定すると、処理テーブル13-1Aの既存エントリが、全て有効であるかを、有効フラグから判定する。

【0066】

(S56) MPU11は、処理テーブル13-1Aの既存エントリが、全てが、有効でない(一部又は全部、無効)と判定すると、図10に示すように、無効なエントリの1つを、登録用エントリに選択する。そして、ステップS62に進む。

【0067】

(S58) MPU11は、処理テーブル13-1Aの既存エントリが、全てが、有効であると判定すると、処理テーブル13-1Aは、図11の状態である。このままでは、新規エントリを登録できない。このため、MPU11は、処理テーブル13-1Aの全エントリのずれ量 1~Mと、今回のずれ量 newの大小関係を比較する。そして、この比較で、今回のずれ量 newが、最小であるかを判断する。MPU11は、処理テーブル13-1Aの全エントリのずれ量 1~Mと、今回のずれ量 newの大小関係の比較により、今回のずれ量 newが、最小であると判断すると、処理テーブル13-1Aへの登録を行うことなく、ステップS68の通常のライトリトライを実行する。この場合、今回、隣接トラックへの影響の可能性ありと判断された書き直し要求は、実行されない。30

【0068】

(S60) 一方、MPU11は、処理テーブル13-1Aの全エントリのずれ量 1~Mと、今回のずれ量 newの大小関係の比較により、今回のずれ量 newが、最小でないと判断すると、処理テーブル13-1Aのずれ量が最小であったエントリを、登録エントリに選択する。

【0069】

(S62) MPU11は、図10及び図11に示すように、ステップS56又はS60で選択された登録用エントリに、書き直し対象トラック番号Nnewを、ずれ量 newを上書きする。

【0070】

10

20

30

40

50

(S64) MPU11は、バッファメモリ14の領域を管理するバッファ制御回路15に、バッファメモリ14のキャッシュメモリ14-1に、この書き直し要求の対象範囲のデータが、存在するかを問い合わせる。存在する場合には、MPU11は、処理テーブル13-1Aの登録用エントリのキャッシュメモリ14-1の位置を、キャッシュメモリの位置に登録する。更に、キャッシュメモリ14-1の当該データの上書き禁止（プロテクト）を設定する。更に、MPU11は、処理テーブル13-1Aの登録エントリのキャッシュデータの有無を、有りとセットする。一方、キャッシュメモリ14-1に、この書き直し要求の対象範囲のデータが、存在しない場合には、MPU11は、処理テーブル13-1Aの登録エントリのキャッシュデータの有無を、無しにセットする。

【0071】

10

(S66) そして、MPU11は、処理テーブル13-1Aの登録用エントリの有効フラグを有効にセットする。

【0072】

(S68) MPU11は、オフトラック検出したトラックのライトリトライを実行する。即ち、キャッシュメモリ14-1から再度、ライトデータを読み出し、そのトラックにライトする。

【0073】

このように、オフトラック検出した際にライトリトライ時には、隣接トラックへの影響の可能性があっても、このライトリトライと同時に、隣接トラックのリライトを実行せずに、処理テーブル13-1Aに、書き直し要求を格納しておく。そして、後述するように、任意のタイミングで、処理テーブル13-1Aを参照して、隣接トラックの書き戻し処理を実行する。

20

【0074】

又、処理テーブル13-1から最小のずれ量のエントリを排除する理由は、ずれ量が小さい場合には、隣接トラックへの影響の可能性があっても、当該隣接トラックのオフセットリード等でデータをリードできる場合があり、それより重大な影響のあるエントリを格納するためである。

【0075】

更に、リトライ範囲の書き直し要求は、リトライで実行できるため、テーブルに登録しない。又、キャッシュメモリにデータが存在するかを予め調べておく。

30

【0076】

次に、この処理テーブル13-1Aを使用したライトコマンド処理を、図12、図13で説明する。

【0077】

(S70) MPU11は、処理テーブル13-1Aに、有効な書き直し処理要求のエントリがある場合に、この処理を開始する。

【0078】

(S72) 先ず、MPU11は、ライトコマンドを受領したかを判定する。

【0079】

(S74) MPU11は、ライトコマンドを受領していないと判定すると、その他のコマンド要求（リード、診断等）を受領しているかを判定する、MPU11は、その他のコマンドを受領している場合は、その要求コマンドを実行し、ステップS72に戻る。

40

【0080】

(S76) MPU11は、その他のコマンドも受領していないと判定すると、バックグラウンドで、書き直し処理を実行する。即ち、MPU11は、ショックセンサー28の出力から、直前の重度のライトオフトラック発生時の振動が収まっているかを判定する。MPU11は、直前の重度のライトオフトラック発生時の振動が収まっていないと判定すると、ステップS72に戻る。

【0081】

(S78) 一方、MPU11は、直前の重度のライトオフトラック発生時の振動が収ま

50

つていると判定すると、処理テーブル 13 - 1 A を参照して、キャッシュデータ有りの有効エントリがあるかを判定する。キャッシュデータ有りの有効エントリが存在しない場合には、有効エントリから任意のエントリを選択し、書き戻し処理を実施する。例えば、隣接トラックのデータをリードし、リードしたデータをその隣接トラックに書き戻す。逆に、キャッシュデータ有りの有効エントリが存在する場合には、その有効エントリの書き直し処理を行う。即ち、処理テーブル 13 - 1 A からキャッシュデータに位置を得て、キャッシュメモリ 14 - 1 のその位置のキャッシュデータを、読み出し、処理テーブル 13 - 1 A で指定されたトラックに書き直す。

#### 【0082】

(S80) 一方、ステップ S72 で、MPU11 は、ライトコマンドを受領していると判定すると、そのコマンドを処理する前に、書き戻し処理する。即ち、MPU11 は、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、処理テーブル 13 - 1 A に存在するかを判定する。即ち、ライトコマンドで指定された書き込みトラック番号と、処理テーブル 13 - 1 A の各エントリのトラック番号とを比較して、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、存在するかを判定する。 10

#### 【0083】

MPU11 は、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、処理テーブル 13 - 1 A に存在すると判定した場合には、当該エントリの書き戻し処理(図 13 参照)を実行し、ステップ S82 に進む。逆に、MPU11 は、ライトコマンドで指定するトラック領域の隣接トラックを対象とした有効な書き直し要求エントリが、処理テーブル 13 - 1 に存在すると判定した場合には、書き直し処理せずに、ステップ S82 に進む。 20

#### 【0084】

(S82) 次に、MPU11 は、処理テーブル 13 - 1 A に、ライトコマンドで指定するライト範囲内に、対象範囲が全て含まれる有効な書き直し要求エントリが、存在するかを判定する。即ち、ライトコマンドで指定された書き込みトラック番号と、処理テーブル 13 - 1 の各エントリのトラック番号とを比較して、ライトコマンドで指定するライト範囲内に、対象範囲が全て含まれる有効な書き直し要求エントリが、存在するかを判定する。存在する場合は、当該ライトコマンドのデータで上書きされるため、そのエントリの有効フラグを無効に変更する。即ち、書き直しは、必要ないため、無効とする。 30

#### 【0085】

(S84) MPU11 は、ライトコマンドを実行する、即ち、ライトコマンドで指定するトラックにライトデータを書き込む。

#### 【0086】

(S86) MPU11 は、処理テーブル 13 - 1 A を参照して、有効な書き直し処理要求が残っているかを判定する。MPU11 は、有効な書き直し処理要求が残っている場合には、ステップ S72 に戻り、残っていなければ、この処理を終了する。

#### 【0087】

図 13 により、図 12 の書き直し処理を説明する。 40

#### 【0088】

(S90) MPU11 は、処理テーブル 13 - 1 A を参照して、そのエントリが、キャッシュデータ有りの有効エントリであるかを判定する。

#### 【0089】

(S92) 逆に、キャッシュデータ有りの有効エントリである場合には、MPU11 は、処理テーブル 13 - 1 A からキャッシュデータに位置を得て、キャッシュメモリ 14 - 1 のその位置のキャッシュデータを、読み出し、処理テーブル 13 - 1 A で指定されたトラックに書き直す。そして、MPU11 は、キャッシュメモリ 14 - 1 のそのキャッシュ位置を上書き可能に設定する(プロテクトを外す)。

#### 【0090】

10

20

30

40

50

(S94) キャッシュデータ有りの有効エントリでない場合には、書き直し対象の隣接トラックのデータをリードし、リードしたデータをその隣接トラックに書き戻す。

#### 【0091】

(S96) そして、MPU11は、処理テーブル13-1Aの当該エントリを無効に設定する。そして、書き直し処理を終了する。

#### 【0092】

図13の書き直し処理は、図12のステップS78の書き直し処理と、ステップS80の書き直し処理とを兼用するため、ステップS90の判断が設けられている。しかしながら、ステップS78の書き直し処理は、実際に実行される処理は、図13のいずれか一方である。

10

#### 【0093】

このように、リトライしたライトコマンドと別のライトコマンドに同期して、ライトコマンドで指定するトラック領域の隣接トラックを、書き直しし、ライトコマンドを実行する。隣接トラックのため、ライトコマンドの実行時間への影響は、小さい。又、ライトコマンドに範囲に含まれる書き直し領域は、特別の書き直しをせず、ライトコマンドの実行により、隣接トラックも上書きされるため、ライトコマンドの実行時間への影響を防止できる。

#### 【0094】

更に、有効エントリも、コマンドの到来しないバックグラウンドで、振動が収まった後、書き直しするため、振動が継続した状態での、書き直しを防止できる。しかも、書き直し処理にキャッシュデータを使用する際、該当するキャッシュデータは当該書き直し処理を実施するまで、保持され続けることになり、使用できるキャッシュ領域の減少及び断片化を招く。このため、キャッシュデータを使用した書き直し処理は、キャッシュデータを使用しない書き直し処理に優先して実施し、キャッシュメモリの有効活用を図る。

20

#### 【0095】

さらに、書き直し処理要求が登録されているシリンドの隣接シリンドに対してライトを実行する際、書き直し対象シリンドがこれ以上、隣接シリンドのライトに伴う寄せ書きの影響を受けないよう、当該書き直し処理を実施してから当該ライト処理を実施している。

#### 【0096】

##### (他の実施の形態)

30

前述の実施の形態では、データ記憶装置を磁気ディスク装置で説明したが、光ディスク、光磁気ディスク、セクタ単位に記録する他の記憶媒体を使用した記憶装置にも適用できる。更に、インターフェースは、ATAに限らず、他のインターフェースにも適用できる。

#### 【0097】

以上、本発明を実施の形態により説明したが、本発明の趣旨の範囲内において、本発明は、種々の変形が可能であり、本発明の範囲からこれらを排除するものではない。

#### 【0098】

尚、本発明は、以下の発明を包含する。

#### 【0099】

(付記1) 回転する記憶媒体に、データのリード及びライトを行うヘッドと、前記記憶媒体の所望トラックに前記ヘッドを位置付けるアクチュエータと、上位からのライトコマンド及びライトデータを受領し、前記ライトデータを、前記ヘッドにより、前記記憶媒体の指定されたトラックにライトするとともに、前記ヘッドによる前記トラックへのライト時に、前記ヘッドが、前記トラックから外れたオフトラックを検出し、前記ライトデータの前記トラックへのライトリトライを実行する制御回路と、少なくとも、対象トラック番号と、前記オフトラック時のヘッドのずれ量とを有する書き直し要求を格納する書き直し要求テーブルとを有し、前記制御回路は、前記オフトラック検出時に、前記ずれ量が所定の閾値より大きいか否かを判定することにより、前記ヘッドのオフトラックによる隣接トラックのデータへの影響の可能性があるか否かを判定し、前記隣接トラックのデータへの

40

50

影響の可能性があると判定した場合には、前記隣接トラックの書き直し要求を、前記書き直し要求テーブルに格納し、前記ずれ量に基づき、前記ライトリトライとは、別の任意のタイミングで、前記書き直し要求テーブルに格納された前記書き直し要求を実行することを特徴とするデータ記憶装置。

【0100】

(付記2) 前記制御回路は、他のライトコマンドを受領した時に、前記他のライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記書き直し要求を実行することを特徴とする付記1のデータ記憶装置。

【0101】

(付記3) 前記制御回路は、前記オフトラック要因である振動が収まったかを判定し、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記書き直し要求を実行することを特徴とする付記1のデータ記憶装置。

10

【0102】

(付記4) 前記制御回路は、前記他のライトコマンドを受領した時に、前記ライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記ライトコマンドで指定されたトラックに隣接するトラックに対する前記書き直し要求を実行することを特徴とする付記2のデータ記憶装置。

【0103】

(付記5) 前記制御回路は、前記他のライトコマンドを受領した時に、前記書き直し要求テーブルに、前記ライトコマンドで指定されたトラックに隣接するトラックに対する前記書き直し要求が存在するかを判定し、前記隣接トラックの書き直し要求が存在する場合に、前記隣接トラックの書き直し要求を実行し、前記隣接トラックの書き直し要求が存在しない場合には、前記オフトラック要因である振動が収まったかを判定し、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記隣接トラック以外の書き直し要求を実行することを特徴とする付記2のデータ記憶装置。

20

【0104】

(付記6) 前記制御回路は、他のライトコマンドを受領した時に、前記書き直し要求テーブルの前記書き直し要求の書き直し領域が、前記他のライトコマンドのライト範囲内であるかを判定し、前記書き直し要求の書き直し領域が、前記他のライトコマンドのライト範囲内であるとの判定に応じて、前記書き直し要求テーブルに格納された前記書き直し要求をリセットすることを特徴とする付記1のデータ記憶装置。

30

【0105】

(付記7) 前記書き直し要求テーブルは、少なくとも、対象トラック番号と、前記オフトラック時のヘッドのずれ量とを有する複数の書き直し要求を格納し、前記制御回路は、前記隣接トラックのデータへの影響の可能性があると判定した場合に、前記隣接トラックの書き直し要求の前記ヘッドのずれ量と、前記書き直し要求テーブルに格納された前記書き直し要求のずれ量とを比較し、前記ずれ量が最小である前記書き直し要求を、前記書き直し要求テーブルから削除することを特徴とする付記1のデータ記憶装置。

【0106】

(付記8) 前記制御回路は、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックが、前記オフトラックを発生した前記ライトコマンドの対象範囲内であるかを判定し、前記隣接トラックが、前記ライトコマンドの対象範囲内であるとの判定に応じて、前記隣接トラックの書き直し要求を、前記書き直し要求テーブルへの格納を禁止することを特徴とする付記1のデータ記憶装置。

40

【0107】

(付記9) 前記制御回路は、前記ライトコマンドを受領していないことを判定し、前記ライトコマンドを受領していない場合に、前記書き直し要求テーブルに格納された前記書き直し要求を実行することを特徴とする付記1のデータ記憶装置。

【0108】

(付記10) 前記ライトデータを一時格納するキャッシュメモリを更に有し、前記制御

50

回路は、前記書き直し要求テーブルの前記書き直し要求の対象データが前記キャッシュメモリに存在するかを判定し、前記書き直し要求の対象データが、前記キャッシュメモリに存在する場合に、前記キャッシュメモリの対象データを前記隣接トラックにライトして、前記書き直し要求を実行することを特徴とする付記1のデータ記憶装置。

#### 【0109】

(付記11) 前記制御回路は、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックのデータが、キャッシュメモリに存在するかを判定し、存在する場合に、前記キャッシュメモリに存在する識別子を付して、前記書き直し要求を、前記書き直し要求テーブルに格納し、前記ライトリトライとは、別のタイミングで、前記書き直し要求テーブルに格納された前記キャッシュメモリに存在する識別子を有する書き直し要求を優先的に実行することを特徴とする付記10のデータ記憶装置。10

#### 【0110】

(付記12) 回転する記憶媒体の所望のトラックに、データのリード及びライトを行うヘッドを位置付け、前記データのリード及びライトを行うデータ記憶装置の隣接トラック書き直し処理方法において、前記ヘッドによる前記トラックへのライト時に、前記ヘッドが、前記トラックから外れたオフトラックを検出するステップと、前記オフトラック検出時に、前記オフトラック時のヘッドのずれ量が所定の閾値より大きいか否かを判定することにより、前記ヘッドのオフトラックによる隣接トラックのデータへの影響の可能性があるか否かを判定するステップと、前記隣接トラックのデータへの影響の可能性があると判定した場合には、少なくとも、対象トラック番号と、前記ヘッドのずれ量とを有する前記隣接トラックの書き直し要求を、書き直し要求テーブルに格納するステップと、前記ライトリトライの前記トラックへのライトリトライを実行するステップと、前記ずれ量に基づき、前記ライトリトライとは、別の任意のタイミングで、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップとを有することを特徴とする隣接トラック書き直し処理方法。20

#### 【0111】

(付記13) 前記書き直し要求を実行するステップは、他のライトコマンドを受領した時に、前記他のライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップを有することを特徴とする付記12の隣接トラック書き直し処理方法。30

#### 【0112】

(付記14) 前記書き直し要求を実行するステップは、前記オフトラック要因である振動が収まったかを判定するステップと、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップとを有することを特徴とする付記12の隣接トラック書き直し処理方法。

#### 【0113】

(付記15) 前記書き直し要求を実行するステップは、前記他のライトコマンドを受領した時に、前記ライトコマンドの実行前に、前記書き直し要求テーブルに格納された前記ライトコマンドで指定されたトラックに隣接するトラックに対する前記書き直し要求を実行するステップを有することを特徴とする付記13の隣接トラック書き直し処理方法。40

#### 【0114】

(付記16) 前記書き直し要求を実行するステップは、前記他のライトコマンドを受領した時に、前記書き直し要求テーブルに、前記ライトコマンドで指定されたトラックに隣接するトラックに対する前記書き直し要求が存在するかを判定するステップと、前記隣接トラックの書き直し要求が存在する場合に、前記隣接トラックの書き直し要求を実行するステップと、前記隣接トラックの書き直し要求が存在しない場合には、前記オフトラック要因である振動が収まったかを判定し、前記振動が収まったと判定した時に、前記書き直し要求テーブルに格納された前記隣接トラック以外の書き直し要求を実行するステップとを有することを特徴とする付記13の隣接トラック書き直し処理方法。

#### 【0115】

(付記 17) 前記書き直し要求を実行するステップは、他のライトコマンドを受領した時に、前記書き直し要求テーブルの前記書き直し要求の書き直し領域が、前記他のライトコマンドのライト範囲内であるかを判定するステップと、前記書き直し要求の書き直し領域が、前記他のライトコマンドのライト範囲内であるとの判定に応じて、前記書き直し要求テーブルに格納された前記書き直し要求をリセットするステップとを有することを特徴とする付記 12 の隣接トラック書き直し処理方法。

【0116】

(付記 18) 前記格納ステップは、前記隣接トラックのデータへの影響の可能性があると判定した場合に、前記隣接トラックの書き直し要求の前記ヘッドのずれ量と、前記書き直し要求テーブルに格納された前記書き直し要求のずれ量とを比較し、前記ずれ量が最小である前記書き直し要求を、前記書き直し要求テーブルから削除するステップを有することを特徴とする付記 12 の隣接トラック書き直し処理方法。

10

【0117】

(付記 19) 前記格納ステップは、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックが、前記オフトラックを発生した前記ライトコマンドの対象範囲内であるかを判定するステップと、前記隣接トラックが、前記ライトコマンドの対象範囲内であることの判定に応じて、前記隣接トラックの書き直し要求を、前記書き直し要求テーブルへの格納を禁止するステップとを有することを特徴とする付記 12 の隣接トラック書き直し処理方法。

【0118】

20

(付記 20) 前記書き直し要求を実行するステップは、前記ライトコマンドを受領していないことを判定するステップと、前記ライトコマンドを受領していない場合に、前記書き直し要求テーブルに格納された前記書き直し要求を実行するステップとを有することを特徴とする付記 12 の隣接トラック書き直し処理方法。

【0119】

30

(付記 21) 前記書き直し要求を実行するステップは、前記書き直し要求テーブルの前記書き直し要求の対象データが、ライトデータを一時格納するキャッシュメモリに存在するかを判定するステップと、前記書き直し要求の対象データが、前記キャッシュメモリに存在する場合に、前記キャッシュメモリの対象データを前記隣接トラックにライトして、前記書き直し要求を実行するステップとを有することを特徴とする付記 12 の隣接トラック書き直し処理方法。

【0120】

(付記 22) 前記格納するステップは、前記隣接トラックのデータへの影響の可能性があると判定した場合には、前記隣接トラックのデータが、キャッシュメモリに存在するかを判定するステップと、存在する場合に、前記キャッシュメモリに存在する識別子を付して、前記書き直し要求を、前記書き直し要求テーブルに格納するステップとを有し、前記書き直し要求を実行するステップは、前記ライトリトライとは、別のタイミングで、前記書き直し要求テーブルに格納された前記キャッシュメモリに存在する識別子を有する書き直し要求を優先的に実行するステップを有することを特徴とする付記 21 の隣接トラック書き直し処理方法。

40

【産業上の利用可能性】

【0121】

ライトコマンドにて隣接トラックのデータに影響を与えるオフトラックエラーが発生した際、当該ライトコマンドに付随して、即時書き直し処理を実施せず、書き直しが必要なトラックの情報やオフトラック時のずれ量等の書き直し要求を保持しておき、ライトリトライと別のタイミングで、保持した書き直し要求を実行するため、オフトラック要因となった振動が収まるのを待ってから書き直し処理を実施でき、書き直し処理時のオフトラックエラーの発生、ないし寄せ書きの発生を抑止でき、また書き直し処理をバックグラウンドで行うことで当該ライトコマンドの実行時間の延伸を軽減し、書き直し処理によるドライブの性能低下を軽減させることができる。

50

## 【図面の簡単な説明】

## 【0122】

【図1】本発明の一実施の形態のデータ記憶装置の構成図である。

【図2】本発明の一実施の形態のオフトラック及び隣接トラックの寄せ書きの説明図である。

【図3】本発明の第1の実施の形態のライトリトライ処理のフロー図である。

【図4】図1及び図3の書き直し要求テーブルの説明図である。

【図5】図4の書き直し要求テーブルへの図3の書き直し要求格納処理の説明図である。

【図6】図4の書き直し要求テーブルのフル時の図3の書き直し要求格納処理の説明図である。

10

【図7】本発明の第1の実施の形態の書き直し処理のフロー図である。

【図8】本発明の第2の実施の形態のライトリトライ処理のフロー図である。

【図9】図1及び図8の書き直し要求テーブルの説明図である。

【図10】図9の書き直し要求テーブルへの図8の書き直し要求格納処理の説明図である。

【図11】図9の書き直し要求テーブルのフル時の図8の書き直し要求格納処理の説明図である。

【図12】本発明の第2の実施の形態の書き直し処理のフロー図である。

【図13】図12の書き直し処理フロー図である。

## 【符号の説明】

20

## 【0123】

10 ディスク装置

11 MPU(処理ユニット)

26 HDC

13 メモリ(ROM/RAM)

13-1, 13-1A 書き直し要求テーブル

14 データバッファ

14-1 キャッシュメモリ

15 データバッファ制御回路

24 リードチャネル回路

30

17 バス

18 ヘッドIC

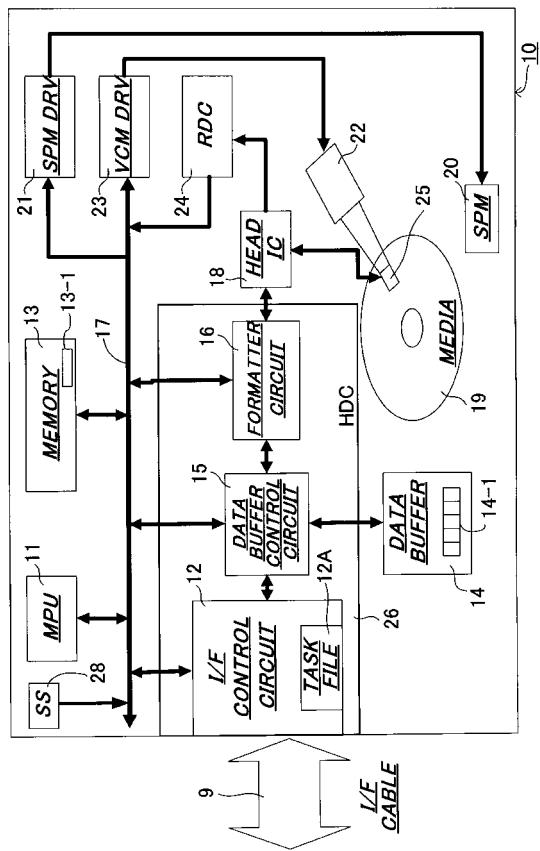
19 記憶媒体(磁気ディスク)

20 スピンドルモータ

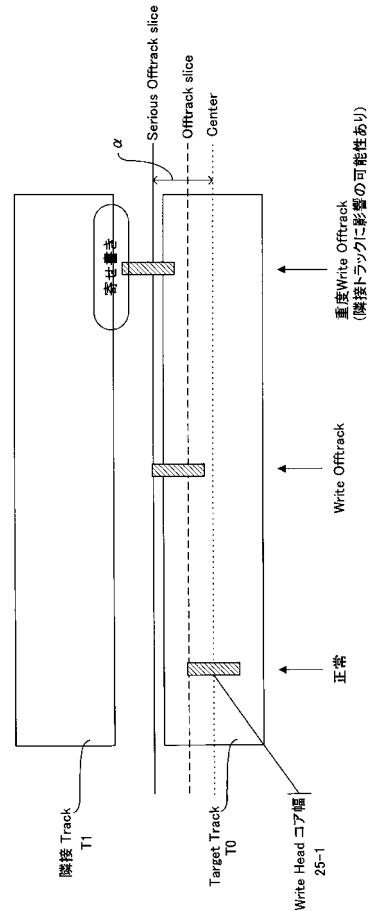
22 アクチュエータ(VCM)

25 ヘッド(磁気ヘッド)

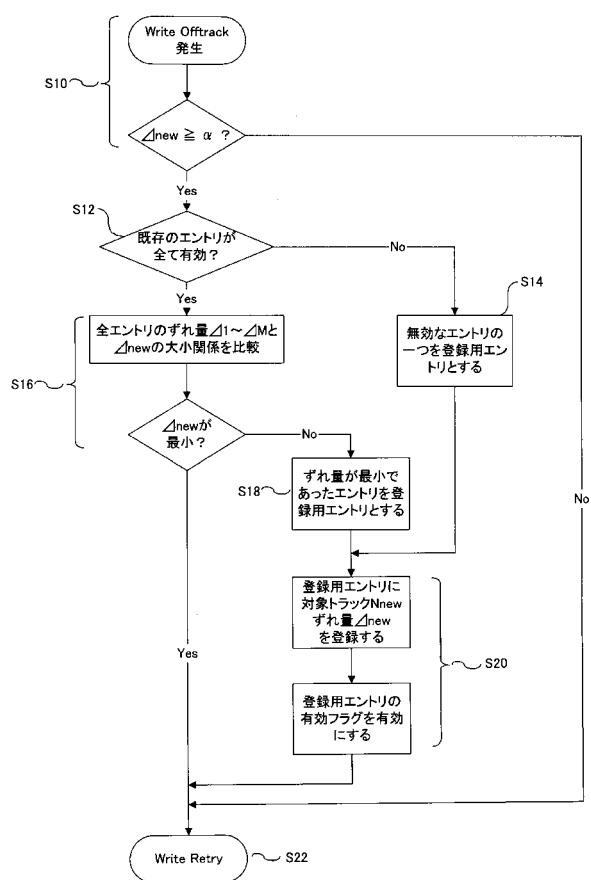
【図1】



【図2】



【図3】



【図4】

エントリ番号	有効フラグ	トラック番号	ずれ量
1	有効	N <sub>1</sub>	△ <sub>1</sub>
2	有効	N <sub>2</sub>	△ <sub>2</sub>
3	無効	N <sub>3</sub>	△ <sub>3</sub>
:	:	:	:
M	無効	N <sub>M</sub>	△ <sub>M</sub>

【図5】

エントリ番号	有効フラグ	トラック番号	ずれ量
1	有効	N <sub>1</sub>	△ <sub>1</sub>
2	有効	N <sub>2</sub>	△ <sub>2</sub>
3	無効	N <sub>3</sub>	△ <sub>3</sub>
:	:	:	:
M	無効	N <sub>M</sub>	△ <sub>M</sub>

(無効なエントリに上書き)

エントリ番号	有効フラグ	トラック番号	ずれ量
1	有効	N <sub>1</sub>	△ <sub>1</sub>
2	有効	N <sub>2</sub>	△ <sub>2</sub>
3	有効	N <sub>new</sub>	△ <sub>new</sub>
:	:	:	:
M	無効	N <sub>M</sub>	△ <sub>M</sub>

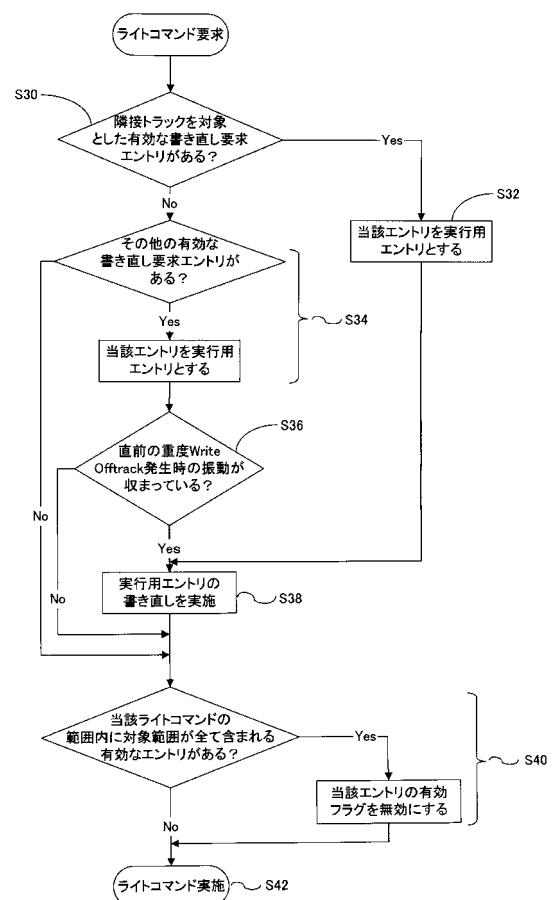
【図6】

エントリ番号	有効フラグ	トラック番号	ずれ量
1	有効	N <sub>1</sub>	Δ <sub>1</sub>
2	有効	N <sub>2</sub>	Δ <sub>2</sub>
3	有効	N <sub>3</sub>	Δ <sub>3</sub>
:	:	:	:
M	有効	N <sub>M</sub>	Δ <sub>M</sub>

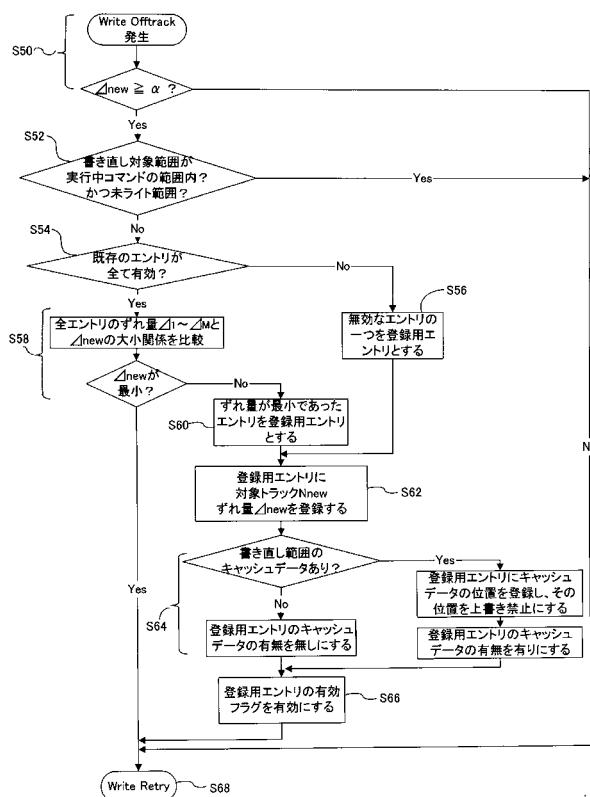
新規登録分  $\Delta_{\text{new}}$  を含めて  $\Delta_2$  が最小の場合

エントリ番号	有効フラグ	トラック番号	ずれ量
1	有効	N <sub>1</sub>	Δ <sub>1</sub>
2	有効	N <sub>new</sub>	Δ <sub>new</sub>
3	有効	N <sub>3</sub>	Δ <sub>3</sub>
:	:	:	:
M	有効	N <sub>M</sub>	Δ <sub>M</sub>

【図7】



【図8】



【図9】

エントリ番号	有効フラグ	トラック番号	ずれ量	キャッシュデータの有無			
				P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
1	有効	N <sub>1</sub>	Δ <sub>1</sub>	有り	有り	無し	無し
2	有効	N <sub>2</sub>	Δ <sub>2</sub>	無し	有り	有り	無し
3	無効	N <sub>3</sub>	Δ <sub>3</sub>	無し	無し	有り	有り
M	無効	N <sub>M</sub>	Δ <sub>M</sub>	無し	有り	有り	有り

【図10】

図10は、2つの状態遷移図を示す。左側の図は「状態S70」から始まる「有効な書き直し処理」である。右側の図は「状態S70」から始まる「無効な書き直し処理」である。

左側の「有効な書き直し処理」の流れは以下の通り。

- 状態S70: 「有効な書き直し処理 要求エントリあり」
- 状態S72: 「ライトコマンド 要求あり?」
  - Yes: 「隣接トラックを 対象とした有効な エントリがある?」
  - No: 「他のコマンド 要求あり?」
- 状態S74: 「他のコマンド 要求あり?」
  - Yes: 「要求コマンドを 実施」
  - No: 「直前の重度Write Offtrack発生時の振動が 収まっている?」
- 状態S76: 「直前の重度Write Offtrack発生時の振動が 収まっている?」
  - Yes: 「当該書き直し処理を実施 (図13)」
  - No: 「隣接トラックを 対象とした有効な エントリがある?」
- 状態S78: 「隣接トラックを 対象とした有効な エントリがある?」
  - Yes: 「当該書き直し処理を実施 (図13)」
  - No: 「当該ライトコマンドの 範囲内に対象範囲が 全て含まれる有効な エントリがある?」
- 状態S80: 「当該ライトコマンドの 範囲内に対象範囲が 全て含まれる有効な エントリがある?」
  - Yes: 「当該エントリの有効 フラグを無効にする」
  - No: 「ライトコマンドを実施」
- 状態S82: 「ライトコマンドを実施」
- 状態S84: 「ライトコマンドを実施」
- 状態S86: 「有効な書き直し 処理要求エントリが 残っている?」
- 状態S88: 「書き直し処理終了」

右側の「無効な書き直し処理」の流れは以下の通り。

- 状態S70: 「有効な書き直し処理 要求エントリあり」
- 状態S72: 「ライトコマンド 要求あり?」
- 状態S74: 「他のコマンド 要求あり?」
- 状態S76: 「直前の重度Write Offtrack発生時の振動が 収まっている?」
- 状態S78: 「隣接トラックを 対象とした有効な エントリがある?」
- 状態S80: 「隣接トラックを 対象とした有効な エントリがある?」
- 状態S82: 「当該書き直し処理を実施 (図13)」
- 状態S84: 「当該書き直し処理を実施 (図13)」
- 状態S86: 「有効な書き直し 処理要求エントリが 残っている?」
- 状態S88: 「書き直し処理終了」

【図11】

図11は、2つの状態遷移図を示す。左側の図は「状態S70」から始まる「有効な書き直し処理」である。右側の図は「状態S70」から始まる「無効な書き直し処理」である。

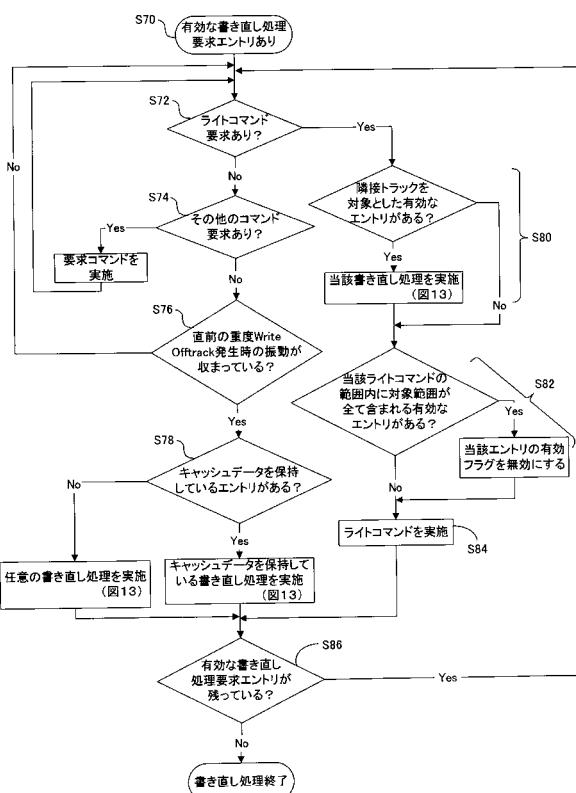
左側の「有効な書き直し処理」の流れは以下の通り。

- 状態S70: 「有効な書き直し処理 要求エントリあり」
- 状態S90: 「キャッシュデータ あり?」
  - Yes: 「キャッシュデータを 要求範囲にライト」
  - No: 「要求範囲をリード」
- 状態S92: 「当該キャッシュ位置を 上書き可能とする」
- 状態S94: 「リードしたデータを 要求範囲にライト」
- 状態S96: 「エントリを無効にする」
- 状態S98: 「書き直し処理終了」

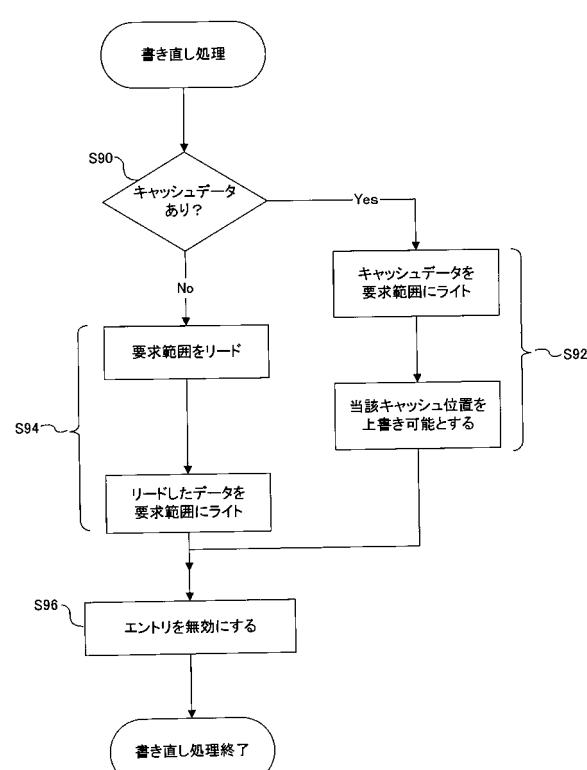
右側の「無効な書き直し処理」の流れは以下の通り。

- 状態S70: 「有効な書き直し処理 要求エントリあり」
- 状態S90: 「キャッシュデータ あり?」
- 状態S92: 「当該キャッシュ位置を 上書き可能とする」
- 状態S94: 「リードしたデータを 要求範囲にライト」
- 状態S96: 「エントリを無効にする」
- 状態S98: 「書き直し処理終了」

【図12】



【図13】



---

フロントページの続き

(51)Int.Cl.

F I

G 1 1 B 20/18 5 7 6 B

(56)参考文献 特開平06-338134 (JP, A)

特開平06-267192 (JP, A)

特開平11-203614 (JP, A)

特開2001-014606 (JP, A)

特開2003-085883 (JP, A)

特開2007-122835 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 1 1 B 5 / 0 9

G 1 1 B 2 0 / 1 8