

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-9622

(P2010-9622A)

(43) 公開日 平成22年1月14日(2010.1.14)

(51) Int.Cl.	F I	テーマコード (参考)
G 0 6 F 13/38 (2006.01)	G 0 6 F 13/38 3 1 0 B	5 B 0 1 4
G 0 6 F 13/10 (2006.01)	G 0 6 F 13/10 3 3 0 A	5 B 0 7 7

審査請求 未請求 請求項の数 4 O L (全 18 頁)

(21) 出願番号	特願2009-233192 (P2009-233192)	(71) 出願人	307017291
(22) 出願日	平成21年10月7日 (2009. 10. 7)		株式会社メディアロジック
(62) 分割の表示	特願2007-112043 (P2007-112043)	(74) 代理人	110000970
の分割			特許業務法人 楓国際特許事務所
原出願日	平成19年4月20日 (2007. 4. 20)	(72) 発明者	紙井 秀之
			富山県高岡市オフィスパーク5番 株式会
			社メディアロジック内
		(72) 発明者	豊田 満
			富山県高岡市オフィスパーク5番 株式会
			社メディアロジック内
		Fターム(参考)	5B014 FB01
			5B077 AA04 AA15 AA23 BA02

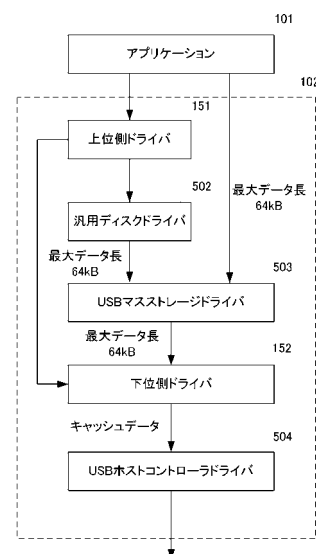
(54) 【発明の名称】 デバイスドライバ

(57) 【要約】

【課題】転送する総データ量が所定量以上となる場合に、デバイス側に特別な構成を必要とせず、転送速度を向上するデバイスドライバを提供する。

【解決手段】USBマストレージドライバ503とUSBホストコントローラドライバ504の間に下位側ドライバ152を設ける。下位側ドライバ152は、上位側からリード、ライト要求を受信し、転送されるデータをキャッシュする。ライト時には上位側から受信したデータをキャッシュし、ある程度の容量のデータとしてまとめて下位側に送信する。リード時には下位側からある程度の容量のデータをまとめて読み出し、上位側に送信する。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

周辺機器が接続されたパーソナルコンピュータの OS に搭載され、
前記周辺機器に対するライトの要求を上限のデータ転送サイズで複数のコマンドに分割して順次送信するソフトウェアの下位側に配置されるデバイスドライバであって、
前記パーソナルコンピュータを、
前記ライトの要求に対応するデータをキャッシュするキャッシュメモリと、
当該デバイスドライバが前記上限のデータ転送サイズのライトの要求を受信したとき、この後連続してライトの要求がされると予測し、各コマンドに対するコマンド終了を擬似的に上位側に返信するとともに、上位側から転送されるデータを順次前記キャッシュメモリにキャッシュし、前記ソフトウェアが分割する各コマンドあたりのデータ転送サイズの上限よりも大きいデータ転送サイズのコマンドを発行して前記キャッシュメモリのデータを前記周辺機器に転送する処理を行う制御部と、
して機能させるデバイスドライバ。

10

【請求項 2】

周辺機器が接続されたパーソナルコンピュータの OS に搭載され、
前記周辺機器に対するリードの要求を上限のデータ転送サイズで複数のコマンドに分割して順次送信するソフトウェアの下位側に配置されるデバイスドライバであって、
前記パーソナルコンピュータを、
前記リードの要求に対応するデータをキャッシュするキャッシュメモリと、
当該デバイスドライバが前記上限のデータ転送サイズのリードの要求を受信したとき、この後連続してリードの要求がされると予測し、前記ソフトウェアが分割する各コマンドあたりのデータ転送サイズの上限よりも大きいデータ転送サイズのコマンドを発行して前記周辺機器からデータを読み出して前記キャッシュメモリにキャッシュする処理を行い、その後上位側から順次送信されたリード要求のコマンドに対応するデータが前記キャッシュメモリにキャッシュされている場合に前記キャッシュメモリから当該データを上位側に転送する制御部と、
して機能させるデバイスドライバ。

20

【請求項 3】

前記制御部は、USB マスストレージデバイスである前記周辺機器とデータ転送を行うことを特徴とする請求項 1 または 2 に記載のデバイスドライバ。

30

【請求項 4】

前記制御部は、OS 標準の USB マスストレージドライバで動作する前記周辺機器とデータ転送を行うことを特徴とする請求項 3 に記載のデバイスドライバ。

【発明の詳細な説明】**【技術分野】****【0001】**

この発明は、パーソナルコンピュータの外付周辺機器を制御するデバイスドライバに関し、特に、データ転送を高速化するデバイスドライバに関する。

【背景技術】

40

【0002】

パーソナルコンピュータ（以下、PC と言う。）のオペレーティングシステム（以下、OS と言う。）が周辺機器を正常に認識、制御するためにはデバイスドライバというソフトウェアが必要になる。

【0003】

近年の PC に搭載される OS には、標準のデバイスドライバが多数組み込まれている。したがって、OS は、標準的な周辺機器が接続された場合には、標準のデバイスドライバを自動的に割り当て、接続された周辺機器を認識、制御する。

【0004】

しかし、標準のデバイスドライバは、各周辺機器に対して最適化されておらず、デバイ

50

ストライバが制御する転送速度よりも高速に動作する周辺機器（例えば高速にリード/ライトする外付HDD等）が存在した場合であっても、標準のデバイスドライバが速度を制限してしまう。また、アプリケーションが速度を制限する場合もある。

【0005】

特に、一般的なOSであるWindows（登録商標）の場合、USBマストレージデバイス（外付HDD等）を接続してデータ転送を行なうと、Windows（登録商標）標準搭載のアプリケーションまたはデバイスドライバが64kB毎にデータを区分し、USBマストレージドライバは区分したデータ毎にコマンド、ステータス等の情報を付加する。このコマンド、ステータス等の情報を処理する時間は長く（数百μsec）、データ転送速度が低下する要因となっていた。

10

【0006】

図1に、OS上に搭載されている各種ドライバの機能ブロック図を示す。同図において、アプリケーション51、汎用ディスクドライバ52、USBマストレージドライバ53、およびUSBホストコントローラドライバ54は、OS上に搭載されている機能部である。

【0007】

アプリケーション51は、各種ドライバを介してUSBマストレージデバイス55（例えば外付HDD）にリードまたはライトの要求を行う。同図においては、アプリケーション51がライトの要求を行い、データ転送する例を示す。アプリケーション51は、汎用ディスクドライバ52に、ライト要求を行い、データを転送する。汎用ディスクドライバ52は、このデータを下位側であるUSBマストレージドライバ53に転送する。このとき、USBマストレージドライバ53のデータ転送サイズの上限が64kBのため、汎用ディスクドライバ52は64kB毎にデータを分割する。USBマストレージドライバ53は、この64kB毎に分割されたデータにコマンド、ステータスを付与して、順次、USBホストコントローラドライバ54に転送する。USBホストコントローラドライバ54は、このコマンド、データ、ステータスをUSBバルク転送方式にて、順次、USBマストレージデバイス55へ転送する。

20

【0008】

図2は、汎用ディスクドライバ52、およびUSBマストレージドライバ53が転送するデータ（バルクオンリープロトコル）の構成を示した概念図である。同図に示すように、汎用ディスクドライバ52は、64kBにデータを分割し、USBマストレージドライバ53は、各データ部にコマンド、ステータスを付与し、順次、USBホストコントローラドライバ54に転送する。さらに、USBホストコントローラドライバ54は、このコマンド、64kBに分割されたデータ、ならびにステータスを順次、USBマストレージデバイス55にデータを転送する。これにより、アプリケーション51はUSBマストレージデバイス55にライトを行う。

30

【0009】

USBマストレージドライバ53、USBホストコントローラドライバ54、およびUSBマストレージデバイス55がコマンドおよびステータスを処理する時間は、USB2.0規格の環境下では500μsec程度である。すると、64kBのデータを転送する毎にコマンド、ステータスの処理時間あわせて500μsec程度の遅延が発生することになり、転送する総データ量が64kB以上であると、データ転送速度が低下する。

40

【0010】

また、図1に示すように、アプリケーション51が直接USBマストレージドライバ53にデータを転送する場合であっても、USBマストレージドライバ53のデータ転送サイズの上限が64kBのため、アプリケーション51は、64kB毎にデータを分割し、USBマストレージドライバ53は、各データ部にコマンド、ステータスを付与し、順次、USBホストコントローラドライバ54に転送する。なお、ドライバ側に64kBのデータ転送サイズの上限が存在しない場合で、かつアプリケーション51が64kBより大きいデータの転送を行なう場合であってもアプリケーション側で64kB毎にデー

50

タを分割する場合もある（例えばWindows（登録商標）標準搭載のエクスプローラなど）。

【0011】

そのため、いずれにしても、転送する総データ量が所定量以上となると、転送速度が低下するという問題が発生していた。

【0012】

一方で、従来、PCとデバイスとの間の転送速度を向上させる手法として、デバイスコントローラにバッファを備え、このバッファにデータをキャッシュし、まとめて転送するものがあった（例えば特許文献1参照）。特許文献1の装置は、転送するデータの総容量がバッファの容量を超えるデータの場合、都度書き込み処理が発生するため、バッファ容量を超えるデータについては、バッファを介さずに直接転送するものである。

10

【0013】

また、複数のバッファを備え、1つのバッファのデータを転送する処理を行いながら、他のバッファに上位側からのデータをキャッシュすることで転送速度を向上させるもの（例えば特許文献2参照）や、コマンド、データ、ステータスをそれぞれキャッシュし、まとめて処理するものが提案されている（例えば特許文献3参照）。

【0014】

また、PC側およびデバイス側で上記のようにデータを分割して処理する必要がある場合において、PC側のデバイスコントローラがある程度のデータをまとめて転送し、デバイス側の制御部でこれを再び分割して処理するものが提案されている（例えば特許文献4

20

【先行技術文献】

【特許文献】

【0015】

【特許文献1】特開平4-130523号公報

【特許文献2】特開2002-344537号公報

【特許文献3】特開2006-215891号公報

【特許文献4】特開2001-154811号公報

【発明の概要】

【発明が解決しようとする課題】

30

【0016】

しかし、特許文献1に記載の装置では、アプリケーションやドライバがデータを分割した場合に高速化できるものではなかった。また、特許文献2および特許文献3に記載の装置は、転送部の処理速度を向上することができるが、アプリケーションやドライバがデータを分割した場合に高速化できるものではなかった。

【0017】

また、特許文献4に記載の計算機システムは、デバイス側の制御部で再びデータを分割して処理する必要がある。また、リード時には、デバイス側の制御部において、分割された各要求を統合してデータ転送する構成が必要であった。

【0018】

40

そこで、この発明は、転送する総データ量が所定量である場合において、デバイス側に特別な構成を必要とせず、転送速度を向上するデバイスドライバを提供することを目的とする。

【課題を解決するための手段】

【0019】

本発明のデバイスドライバは、周辺機器が接続されたパーソナルコンピュータのOSに搭載され、前記周辺機器に対するライトの要求を上限のデータ転送サイズで複数のコマンドに分割して順次送信するソフトウェアの下位側に配置されるデバイスドライバであって、前記パーソナルコンピュータを、前記ライトの要求に対応するデータをキャッシュするキャッシュメモリと、前記ドライバが前記上限のデータ転送サイズのライトの要求を受信

50

したとき、この後連続してライトの要求がされると予測し、各コマンドに対するコマンド終了を擬似的に上位側に返信するとともに、上位側から転送されるデータを順次前記キャッシュメモリにキャッシュし、前記ソフトウェアが分割する各コマンドあたりのデータ転送サイズの上限よりも大きいデータ転送サイズのコマンドを発行して前記キャッシュメモリのデータを前記周辺機器に転送する処理を行う制御部と、して機能させることを特徴とする。

【 0 0 2 0 】

また、本発明のデバイスドライバは、周辺機器が接続されたパーソナルコンピュータのOSに搭載され、前記周辺機器に対するリードの要求を上限のデータ転送サイズで複数のコマンドに分割して順次送信するソフトウェアの下位側に配置されるデバイスドライバであって、前記パーソナルコンピュータを、前記リードの要求に対応するデータをキャッシュするキャッシュメモリと、前記ドライバが前記上限のデータ転送サイズのリードの要求を受信したとき、この後連続してリードの要求がされると予測し、前記ソフトウェアが分割する各コマンドあたりのデータ転送サイズの上限よりも大きいデータ転送サイズのコマンドを発行して前記周辺機器からデータを読み出して前記キャッシュメモリにキャッシュする処理を行い、その後上位側から順次送信されたリード要求のコマンドに対応するデータが前記キャッシュメモリにキャッシュされている場合に前記キャッシュメモリから当該データを上位側に転送する制御部と、して機能させることを特徴とする。

【 0 0 2 1 】

このように、本発明のデバイスドライバは、上位側のアプリケーションやデバイスドライバから受信したリード/ライトの要求を統合し、デバイスに転送する。これにより、アプリケーションやデバイスドライバから受信したデータをキャッシュし、ある程度の容量のデータとして、まとめて対象デバイスに転送する。また、その逆に、対象デバイスからある程度の容量をまとめて受信し、キャッシュしたデータとして順次アプリケーションやデバイスドライバに転送する。アプリケーションやデバイスドライバが転送データ容量を制限し、分割してデータを転送する場合であっても、ある程度の容量のデータをまとめて1つの（あるいは数を減らして）コマンド、ステータスを発行し、転送することで、高速化を実現する。また、リード要求を受信したとき、この要求に対応するデータがキャッシュメモリに記憶されていれば、これを転送することができる。一度転送したデータをキャッシュしておけば、次に同じデータのリード要求を受信したときに高速に転送することができる。また、一般に、ストレージデバイスのランダムアクセスは、ヘッドのシーク動作を伴うため、これが転送速度の低下を引き起こすが、上記のようにキャッシュメモリからデータを転送することで、対象デバイスへのリード要求の発行頻度を軽減することができるとともに、対象デバイスのランダムアクセスの発生頻度の軽減にもつながり、さらなる高速化を実現できる。

【 発明の効果 】

【 0 0 2 2 】

この発明によれば、アプリケーションやデバイスドライバが制限するデータ量以上のデータを転送することができるため、転送速度を向上することができる。

【 図面の簡単な説明 】

【 0 0 2 3 】

【 図 1 】 USB インタフェースで接続される PC と外付 HDD を表したブロック図である。

【 図 2 】 バルクオンリープロトコルの構成を示した概念図である。

【 図 3 】 本実施形態における PC と外付 HDD のブロック図である。

【 図 4 】 OS 1 1 上に搭載されている各種ドライバの機能ブロック図である。

【 図 5 】 上位側ドライバ 1 5 1 および下位側ドライバ 1 5 2 の詳細な構成を示す機能ブロック図である。

【 図 6 】 OS 1 1 上に搭載されている各種ドライバの機能ブロック図である。

【 図 7 】 上位側ドライバ 1 5 1 の動作を示すフローチャートである。

10

20

30

40

50

【図 8】下位側ドライバ 152 の動作を示すフローチャートである。

【図 9】コマンド終了割り込み時の動作を示すフローチャートである。

【図 10】タイマ割り込み、初期化時、ドライバ終了時の動作を示すフローチャートである。

【図 11】本発明のデバイスコントローラを実現する他の例を示すブロック図である。

【図 12】上位側ドライバ 151 および下位側ドライバ 152 が別の動作を行う場合の構成を示す機能ブロック図である。

【発明を実施するための形態】

【0024】

PC に搭載される OS で一般的に普及しているものとして Windows (登録商標) がある。本実施形態では、Windows (登録商標) XP を搭載している PC に、USB インタフェースの外付 HDD を接続する場合について、図を用いて説明する。なお、本発明において OS の種別や接続デバイスの種別、インタフェースを限定するものではない。

10

【0025】

図 3 は、本実施形態における PC と外付 HDD のブロック図である。PC 1 と外付 HDD 2 は USB インタフェース (I/F) で接続される。PC 1 は、全体を管理する OS 11 上に、種々の処理を行うアプリケーション 101、および周辺機器を動作させるための各種ドライバが含まれるドライバ 102 を搭載している。アプリケーション 101 は、ドライバ 102、USB コントローラ 12、および USB I/F 13 を介して外付 HDD 2 に記憶されているデータを読み出し (リード)、外付 HDD 2 にデータを書き込む (ライト)。

20

【0026】

外付 HDD 2 は、コントローラ 22 が USB I/F 21 を介してリードまたはライトの要求を受信し、これに応じて HDD 23 に記憶されているデータをリード、または HDD 23 にデータをライトする。

【0027】

図 4 は、OS 11 上に搭載されている各種ドライバの機能ブロック図である。同図において、アプリケーション 101、汎用ディスクドライバ 502、USB マスストレージドライバ 503、USB ホストコントローラドライバ 504、上位側ドライバ 151、および下位側ドライバ 152 は、OS 11 上に搭載されている機能部である。実際には、これらの機能部は、OS 11 にインストールされるソフトウェアとして実現する。

30

【0028】

アプリケーション 101 は、ドライバ 102 に含まれる各種ドライバを介して外付 HDD 2 にリードまたはライトの要求を行う。同図においては、アプリケーション 101 がライトの要求を行い、データ転送する例を示す。

【0029】

アプリケーション 101 は、上位側ドライバ 151 に、ライト要求を行い、外付 HDD 2 の HDD 23 に書き込むべきデータを転送する。上位側ドライバ 151 は、このライト要求およびデータを下位側である汎用ディスクドライバ 502 に転送するとともに、アプリケーション 101 からライト要求があった旨、およびそのデータ量を示す情報をアクセス情報として下位側ドライバ 152 に送信する。上位側ドライバ 151 が下位側ドライバ 152 に転送するアクセス情報については後に詳しく述べる。

40

【0030】

汎用ディスクドライバ 502 は、上位側ドライバ 151 から受信したデータを USB マスストレージドライバ 503 に転送する。このとき、USB マスストレージドライバ 503 のデータ転送サイズの上限が 64 kB であるため、汎用ディスクドライバ 502 は、64 kB 毎に分割して USB マスストレージドライバ 503 にデータを転送する。

【0031】

USB マスストレージドライバ 503 は、バルクオンリープロトコル (図 2 参照) によ

50

リデータ転送を行うため、この 64 k B に分割されたデータにコマンド、ステータスを付与し、順次、下位側ドライバ 152 に転送する。

【0032】

下位側ドライバ 152 は、USB マスストレージドライバ 503 から転送されたデータをキャッシュし、所定のタイミングで USB ホストコントローラドライバ 504 に転送する。USB ホストコントローラドライバ 504 は、このデータを外付 HDD 2 に転送する。下位側ドライバ 152 が、USB マスストレージドライバ 503 から転送されたデータをキャッシュし、ある程度の容量 (64 k B 以上) のデータとして、まとめて USB ホストコントローラドライバ 504 に転送するため、コマンド、ステータスの発行が 1 度で (あるいは 64 k B 毎に付与するよりも少なく) 済む。したがって、コマンド、ステータスの処理時間による遅延が最小限で済む。なお、汎用ディスクドライバ 502 から下位側ドライバ 152 まで転送されるデータは、64 k B 毎にコマンド、ステータスが付与されているが、この転送は、OS 上のメモリ (RAM) 転送であるため、外付 HDD 2 へのデータ転送速度に比較して非常に高速である (RAM 転送速度に依存する)。

10

【0033】

また、この実施形態において、アプリケーション 101 が直接 USB マスストレージドライバ 503 にデータ転送する場合があるが、上記と同様、USB マスストレージドライバ 503 のデータ転送サイズの上限が 64 k B であるため、アプリケーション 101 が 64 k B 毎に分割してデータを転送し、USB マスストレージドライバ 503 がコマンド、ステータスを付与したデータを転送する。この場合においても、下位側ドライバ 152 がデータをキャッシュしてから下位側にデータ転送するため、コマンド、ステータスの処理時間による遅延が最小限で済む。

20

次に、上位側ドライバ 151 および下位側ドライバ 152 の詳細な動作について説明する。図 5 は、上位側ドライバ 151 および下位側ドライバ 152 の詳細な構成を示す機能ブロック図である。なお、図 4 と共通する構成については同一の符号を付し、その説明を省略する。上位側ドライバ 151 は、リード、ライト要求受信部 155 を備えている。また、下位側ドライバ 152 は、コントローラ 156、データベース 157、キャッシュメモリ 158A ~ 158C、タイマ 159、およびコマンドキュー 160 を備えている。

【0034】

リード、ライト要求受信部 155 は、アプリケーション 101 からリードまたはライトの要求を受信する。リード、ライト要求受信部 155 は、この要求を下位側の汎用ディスクドライバ 502 に転送するとともに、要求の内容をアクセス情報として下位側ドライバ 152 に送信する。リードまたはライトの要求は、汎用ディスクドライバ 502、および USB マスストレージドライバ 503 を介して下位側ドライバ 152 に転送される。

30

【0035】

アクセス情報には、LBA (Logical Block Addressing: HDD のセクタ指定番号)、転送ブロック数、転送方向などが記載されている。転送ブロック数は転送データの容量を示す。転送方向は、下位側から上側への転送 (リード) であるか、上位側から下位側への転送 (ライト) であるかを示す情報である。

【0036】

このアクセス情報は、下位側ドライバ 152 のデータベース 157 に予想データベースとして記憶される。下位側ドライバ 152 のコントローラ 156 は、USB マスストレージドライバ 503 からリードまたはライトの要求を受信したとき、データベース 157 を参照する。予想データベースを参照することで、転送データ容量を予測することができる。

40

【0037】

また、コントローラ 156 は、USB マスストレージドライバ 503 から受信したリードまたはライトの要求を履歴データベースとしてデータベース 157 に記憶する。履歴データベースは、コントローラ 156 が予想データベースを作成、修正する場合に用いられる。すなわち、USB マスストレージドライバ 503 からリードまたはライトの要求を受

50

信したとき、過去の履歴データベースを参照し、L B A、転送ブロック数、転送方向が同一または一部一致する（類似する）ものを予想データベースとして転記する。また、一致する履歴が存在した場合、これと関連する履歴（例えばL B Aと転送ブロック数に基づいて、連続して要求があったと判断できるもの）をまとめ、1つのコマンドとして予想データベースに記憶することもできる。なお、履歴データベースは、同図においてはデータベース157内に記憶され、R A M上に一時的に記憶されているが、これをP C 1のO S 11上、または外付H D D 2のH D D 23に記憶しておいてもよい。これにより、外付H D D 2を再接続したときやP C 1を再起動したときにおいても過去の履歴を参照することができる。

【0038】

コントローラ156は、U S Bマストレージドライバ503から転送された要求がライトの要求であれば、予想データベースの転送ブロック数を参照することで、転送データ容量を予測することができる。このとき、上位側ドライバ151が下位側ドライバ152に、アクセス情報を送信して予想データベースに記憶されている場合は、転送データ容量（ライトすべき全データ）を予測することができる。コントローラ156は、64k B以上のデータ転送が行われると予測した場合、上位側から転送されたデータをキャッシュメモリ158A～158Cのいずれかに格納する。

【0039】

キャッシュメモリ158A～158Cは、R A M上に仮想的に形成されたエリアであり、その数および容量はコントローラ156が設定する。キャッシュメモリの数および容量は、適宜設定し得るが、同図の例では、3つのキャッシュメモリを用意し、それぞれ1M Bの容量を設定しているものとする。なお、動作中にキャッシュメモリの数および容量を変化させることも可能である。キャッシュメモリの数および容量はユーザが手動で設定するようにしてもよいし、後述のように動作状態に応じて適宜変化させるようにしてもよい。

【0040】

上記のように、コントローラ156は、キャッシュメモリ158A～158Cのうち、いずれか1つに上位側から転送されたデータを順次、格納する。このライト用のデータを格納したキャッシュメモリがライト用キャッシュメモリとなる。ただし、U S Bマストレージドライバ503から64k B毎にコマンド、ステータスが付与されてデータ転送されるため、コントローラ156は、64k Bのデータを受信する度に、U S Bマストレージドライバ503に対してコマンド終了を擬似的に返信する。

【0041】

コントローラ156は、所定のタイミングで下位側のU S Bホストコントローラドライバ504に対し、ライト要求を行い、ライト用キャッシュメモリのデータを転送する。この所定のタイミングは、ライト用キャッシュメモリの容量が一杯になったタイミング、またはタイマ159の割り込み処理により決定されるタイミングである。コントローラ156は、U S Bマストレージドライバ503からライト要求を受信したときにタイマ159を作動させる。また、ライト用キャッシュメモリの容量が一杯になった場合も、U S Bホストコントローラドライバ504に対し、ライト要求を行い、ライト用キャッシュメモリのデータを転送する。

【0042】

なお、U S Bホストコントローラドライバ504以下の下位側において他の処理を行っている場合、コントローラ156は、一旦ライト要求をコマンドキュー160に登録しておき、後に下位側の処理が空いているときにライト要求を行う。なお、タイマが無い場合、ライト用キャッシュメモリの内容がそのまま保持され、H D D 23に書き込まれなくなるため、この状態で異常発生した場合（U S Bケーブルが切断されたり、P C 1がハングアップした場合）、本来H D D 23に書き込まれるべきであったデータが破棄され、ファイル破損等の被害が想定される。そのため、タイマを使用して、例えば一定時間コマンド発行が無い状態が経過した場合に、ライト用キャッシュメモリの内容を書き込みすること

10

20

30

40

50

で、異常時のファイル破壊等を防止するようにしている。

【 0 0 4 3 】

以上のようにして、アプリケーション 1 0 1 からデータのライトが行われる。次に、図 6 において、アプリケーション 1 0 1 がリードの要求を行い、データ転送する例を示す。

【 0 0 4 4 】

図 6 は、OS 1 1 上に搭載されている各種ドライバの機能ブロック図である。なお、図 4 と共通する構成については同一の符号を付し、その説明を省略する。

【 0 0 4 5 】

まず、アプリケーション 1 0 1 は、上位側ドライバ 1 5 1 に、リード要求を行う。上位側ドライバ 1 5 1 は、この要求を下位側である汎用ディスクドライバ 5 0 2 に転送するとともに、アプリケーション 1 0 1 からリード要求があった旨（アクセス情報）を下位側ドライバ 1 5 2 に送信する。

【 0 0 4 6 】

汎用ディスクドライバ 5 0 2 は、上位側ドライバ 1 5 1 から受信したリード要求を USB マスストレージドライバ 5 0 3 に転送する。このとき、USB マスストレージドライバ 5 0 3 のデータ転送サイズの上限が 6 4 k B であるため、汎用ディスクドライバ 5 0 2 は、6 4 k B 毎にリード要求を分割する。USB マスストレージドライバ 5 0 3 は、下位側ドライバ 1 5 2 に 6 4 k B 毎に分割されたリード要求を転送する。

【 0 0 4 7 】

下位側ドライバ 1 5 2 は、USB マスストレージドライバ 5 0 3 から 6 4 k B 毎に分割されたリード要求を受信した場合、この後連続してリード要求がされると予測し、ある程度の容量のデータを転送するように、USB ホストコントローラドライバ 5 0 4 に予測リード要求を行う。USB ホストコントローラドライバ 5 0 4 は、この予測リード要求に基づいて、ある程度の容量のデータを HDD 2 の HDD 2 3 から読み出す。このリードデータが下位側ドライバ 1 5 2 にキャッシュされる。下位側ドライバ 1 5 2 は、USB マスストレージドライバ 5 0 3 から受信したリード要求にしたがって、キャッシュしたデータを順次、上位側に転送する。ある程度の容量とは、適宜設定すればよいが、転送容量が大きすぎる場合は、転送速度が遅くなる。すなわち、その転送の終了を待ち、その転送の正常終了のステータスを確認するまで、上位側にキャッシュしたデータの転送を開始できないため、転送速度が遅くなる原因となる。また、リード用キャッシュメモリの容量を大きく

【 0 0 4 8 】

そのため、コントローラ 1 5 6 は、最初に 6 4 k B（あるいはそれ以下、3 2 k B 等であってもよい。）のリード要求を受信したとき、その所定倍（例えば 3 2 k B の 4 倍である 1 2 8 k B）の容量のデータを読み出してキャッシュする。このキャッシュしたデータが、結果的に全て上位側に転送される場合、さらに所定倍（例えば 1 2 8 k B の 4 倍である 5 1 2 k B）の容量のデータを読み出す。このように、順次、キャッシュするデータの容量を大きくすることで、転送時間を抑えながら、好適なリード用キャッシュメモリの容量に設定することができる。

【 0 0 4 9 】

このように、下位側ドライバ 1 5 2 が、外付 HDD 2 からある程度の容量を一度に読み出し、キャッシュしておくことで、外付 HDD 2 に対するコマンド、ステータスの発行が 1 度で（あるいは 6 4 k B 毎に付与するよりも少なくても）済む。したがって、コマンド、ステータスの処理時間による遅延が最小限で済む。なお、下位側ドライバ 1 5 2 から汎用ディスクドライバ 5 0 2 まで転送されるデータは、6 4 k B に分割されるが、この転送は、OS 上のメモリ（RAM）転送であるため、外付 HDD 2 からのデータ転送速度に比較して非常に高速である（RAM 転送速度に依存する）。

【 0 0 5 0 】

図 5 を参照して、上記リード要求発生時の上位側ドライバ 1 5 1 および下位側ドライバ 1 5 2 の詳細な動作について説明する。リード、ライト要求受信部 1 5 5 は、アプリケー

10

20

30

40

50

ション 101 からリードの要求を受信する。リード、ライト要求受信部 155 は、この要求を下位側の汎用ディスクドライバ 502 に転送するとともに、要求の内容をアクセス情報として下位側ドライバ 152 に送信する。リード要求は、汎用ディスクドライバ 502、および USB マスストレージドライバ 503 を介して下位側ドライバ 152 に転送される。なお、汎用ディスクドライバ 502、および USB マスストレージドライバ 503 から転送されるリード要求は 64 k B 毎のリード要求に分割されている。

【0051】

上位側ドライバ 151 が送信したアクセス情報は、下位側ドライバ 152 のデータベース 157 に予想データベースとして記憶される。下位側ドライバ 152 のコントローラ 156 は、USB マスストレージドライバ 503 からリード要求を受信したとき、データベース 157 を参照する。予想データベースを参照することで、転送データ容量（リードすべき全データ）を予測することができる。

10

【0052】

また、USB マスストレージドライバ 503 から受信したリード要求を履歴データベースとしてデータベース 157 に記憶する。履歴データベースは、コントローラ 156 が予想データベースを作成、修正する場合に用いられる。すなわち、USB マスストレージドライバ 503 からリード要求を受信したとき、過去の履歴データベースを参照し、LBA、転送ブロック数、転送方向が同一または類似するものを予想データベースとして転記する。一致する履歴が存在する場合、これと関連する履歴（例えば LBA と転送ブロック数に基づいて、連続して要求があったと判断できるもの）をまとめ、1つのコマンドとして予想データベースに記憶することもできる。

20

【0053】

コントローラ 156 は、USB マスストレージドライバ 503 からリード要求を受信したとき、予想データベースの転送ブロック数を参照することで、転送データ容量（リードすべき全データ）を予測することができる。コントローラ 156 は、64 k B 以上のデータ転送が行われると予測した場合、USB ホストコントローラドライバ 504 に予想データベースに記載された 64 k B 以上のデータを転送するようにリード要求を行い、転送されたデータをキャッシュメモリ 158 A ~ 158 C のいずれかに格納する。

【0054】

このデータを格納したキャッシュメモリがリード用キャッシュメモリとなる。コントローラ 156 は、USB マスストレージドライバ 503 から転送されるリード要求に応じて、順次リード用キャッシュメモリのデータを転送する。なお、上位側ドライバ 151 が、リードすべき全データのアクセス情報を送信した場合は、リード用キャッシュメモリに正確に全データをキャッシュすることができるが、アプリケーション 101 が直接 USB マスストレージドライバ 503 にリード要求を行う場合や上位側ドライバ 151 がリードすべき全データのアクセス情報を送信しない場合は、コントローラ 156 が上位側からリード要求を受信したときに予想データベースを作成する必要がある。この場合、上記のように履歴データベースを参照すればよい。例えば、最初に 64 k B（あるいはそれ以下）のリード要求を受信したとき、そのリード要求に連続した開始 LBA から、所定倍の容量のデータがリードされるように予測する。この予測した開始 LBA に対するリード要求を上位側のドライバから受信した場合、その予測容量分のリードを行い、予測したデータが全て上位側のドライバからリードされた場合、さらにその予測リードに連続する開始 LBA から、さらに所定倍の容量のデータがリードされるように予測する。

30

40

【0055】

以上のようにして、アプリケーション 101 からデータのリードが行われる。

【0056】

次に、上位側ドライバ 151 および下位側ドライバ 152 の動作をフローチャートを用いて説明する。図 7 (A) は、上位側ドライバ 151 の動作を示すフローチャートである。上位側ドライバ 151 は、アプリケーションからリード、ライト要求を受信したとき、この要求の内容をアクセス情報として下位側ドライバ 152 に送信する (s1)。また、

50

上位側ドライバ 1 5 1 は、下位側（汎用ディスクドライバ 5 0 2）にリード、ライト要求を転送する（s 2）。アクセス情報は、下位側ドライバ 1 5 2 のデータベース 1 5 7 に予想データベースとして記憶される。

【 0 0 5 7 】

なお、上位側ドライバ 1 5 1 は、図 7（B）および図 1 2 に示すような動作を行うこともできる。図 7（B）は、上位側ドライバ 1 5 1 の別の動作を示す図であり、図 1 2 は、この動作を行う場合の上位側ドライバ 1 5 1 および下位側ドライバ 1 5 2 の詳細な構成を示す機能ブロック図である。なお、図 5 と共通する構成については、同一の符号を付し、その説明を省略する。

【 0 0 5 8 】

図 7（B）において、上位側ドライバ 1 5 1 は、アプリケーションからリード、ライト要求を受信したとき、転送されるデータ容量が所定量以上（例えば 6 4 k B 以上）であるか否かを判断する（s 5）。所定量以上でなければ要求の内容をアクセス情報として下位側ドライバ 1 5 2 に転送し（s 6）、汎用ディスクドライバ 5 0 2 にリード、ライト要求を転送する（s 7）。所定量以上であれば、図 1 2 に示すように、リード、ライト要求を直接下位側ドライバ 1 5 2 のコントローラ 1 5 6 に転送する（s 8）。この場合、汎用ディスクドライバ 5 0 2 にはリード、ライト要求を転送しない。リード、ライト要求を直接コントローラ 1 5 6 に転送することで、6 4 k B にデータが分割されることなく転送することができる。また、汎用ディスクドライバ 5 0 2、および U S B マスストレージドライバ 5 0 3 を介さないため、処理時間を短縮することができ、より高速化を期待できる。転送されるデータ容量が所定量以上でなければ、U S B マスストレージドライバ 5 0 3 が一度にデータを処理することができるため、下位側（汎用ディスクドライバ 5 0 2）に転送するようにしている。

【 0 0 5 9 】

なお、s 5 の処理に換えて、リード要求であるか否かの判断としてもよい。リード要求であれば直接下位側ドライバ 1 5 2 のコントローラ 1 5 6 に転送する。コントローラ 1 5 6 では、上位側（U S B マスストレージドライバ）からリード、ライト要求を受信したときに予測処理を行うため、特に、リード時には予測結果に基づいてある程度のデータ容量をデバイス側から読み出し、キャッシュしてから上位側に返信するため、その分の遅延が生じる可能性がある。そのため、リード要求時には上位側ドライバ 1 5 1 からコントローラ 1 5 6 に直接この要求を転送し、予測処理やキャッシュ処理を行わず、転送時間（転送開始までの時間）を短縮することができる。

【 0 0 6 0 】

なお、リード要求であるか否かの判断と s 5 の判断とをいずれか 1 つ行うようにしてもよいし、両方行ってもよい。なお、その他の判断手法を用いてもよい。例えば、デバイス側から予めデータを読み出し、キャッシュしておくことを重視する場合（アプリケーションが同じデータ内容にアクセスすることを繰り返すような場合）、リード要求時においても直接コントローラ 1 5 6 に要求を転送せずに、下位側に転送する。上位側ドライバ 1 5 1 が下位側ドライバ 1 5 2 のコントローラ 1 5 6 に直接リード、ライト要求を転送する場合、キャッシュメモリにデータが蓄積されない。そのため、アプリケーションが同じファイルにアクセスするような場合であればキャッシュメモリにデータが蓄積されるように下位側に転送すればよい。ただし、下位側ドライバ 1 5 2 がデータを転送するとともに、このデータをキャッシュメモリにコピーすることで、直接リード、ライト要求を受信した場合であっても、データを蓄積することが可能である。この場合、R A M 上のコピーであるため、コピー速度は高速であり、処理速度が著しく低下することはない。

【 0 0 6 1 】

図 8 は、下位側ドライバ 1 5 2 の動作を示すフローチャートである。上位側ドライバ（U S B マスストレージドライバ 5 0 3 またはアプリケーション 1 0 1）からリード要求、ライト要求、またはその他の要求を受信するとこの動作を開始する。

【 0 0 6 2 】

まず、下位側ドライバ 152 は、アクセス予測を行う (s 11)。アクセス予測は、上位側ドライバ 151 から送信されたアクセス情報や履歴データベースを参照することで行う。また、以下のようにして、精密にアクセス予測を行ってもよい。

【0063】

(1) ファイルシステムに基づく予測

この予測手法は、外付 HDD 2 のファイルシステムを解析することでアクセス予測を行う手法である。HDD のファイルシステムが、例えば FAT ファイルシステムフォーマットである場合、ファイル管理領域を参照することで、個々のファイルの記録位置 (セクタ) やサイズを判断することができる。よって、上位側のアプリケーションやドライバからリード要求があったとき、その要求が示す記録位置に対応するファイルのデータにアクセスが有ると予測する。またファイルサイズ以上のリードを行わないように予測することで、必要以上のリードをせずに、無駄な処理時間を削減する。

【0064】

なお、このファイル管理領域は、頻繁にアクセスを行う領域であるため、リード用キャッシュメモリにリードし、保持しておいてもよい。都度、HDD 2 にアクセスしなくとも、リード用キャッシュメモリに保持しているファイル管理領域にアクセスすることで、アクセス予測の処理時間を削減することができる。

【0065】

(2) デバイス種別、コンテンツ種別に基づく予測

この予測手法は、接続される対象デバイスのタイプ (上記実施形態では HDD) や転送すべきデータのファイル構成 (コンテンツ種別) に基づいてアクセス予測を行う手法である。例えば、接続されるデバイスが DVD-ROM であり、このデバイスに DVD-Video メディアが挿入されている場合、順次リードがされ、ライト要求がされることがない。そのため、リード用キャッシュメモリの容量一杯までデータがキャッシュされるように予測データベースを作成する。なお、リード用キャッシュメモリにキャッシュしたデータが全てリードされた場合、即時このキャッシュを破棄し、新たなデータをキャッシュすれば、効率的にリード用キャッシュメモリを用いることができる。

【0066】

また、キャッシュメモリの数および容量を変化させる場合、デバイス種別に基づいて変化させることができる。例えば、上記のように DVD-Video メディアが挿入されている場合、ライト要求されることがないため、リード用キャッシュメモリの数を増やし、その容量も大きく設定する。逆に未書き込みの DVD-R メディアが挿入されている場合、主にライト要求がされるため、ライト用キャッシュメモリの数を増やし、その容量を大きく設定する。

【0067】

(3) 転送速度の解析に基づく予測

この予測手法は、上記デバイス種別、コンテンツ種別に基づく予測の応用例であり、デバイス種別に応じてデータ転送速度の傾向を判断し、これに基づいてアクセス予測を行う手法である。例えば、多層型 DVD メディアが接続される場合、各層の境界線を越えるランダムアクセスは、転送速度が非常に遅くなる。そのため、各層の境界線をまたぐアクセスの頻度が低くなるようにキャッシュするようにして、予測データベースを作成する。

【0068】

また、キャッシュメモリの数および容量を変化させる場合には、このデバイス種別やデータ転送速度の傾向に基づいて変化させることもできる。例えば、対象デバイスとのデータ転送速度が遅い場合 (例えば USB MO (光磁気ディスク) のように 4 MB/s 程度である場合)、キャッシュメモリの容量を大きくして大量にデータをキャッシュしても、そもそもデータ転送速度が遅いために転送速度の向上が望めない。そのため、対象デバイスとのデータ転送速度が遅い場合、キャッシュメモリの容量を小さく設定し、RAM の消費を抑える。

【0069】

10

20

30

40

50

(4) リード要求時に1度に転送するデータを徐々に増加させる手法

この予測手法は、リード要求時に、上記(1)～(3)の手法で予測したリード要求に基づいて、対象デバイスから一度にデータ転送しないように、予測データベースを修正する(分割する)手法である。例えばリード要求の所定倍(例えば4倍)の容量のデータをまずキャッシュするように予測データベースを修正する。このキャッシュしたデータが、結果的に全て上位側に転送される場合、さらに所定倍の容量のデータをキャッシュするように予想データベースを修正する。順次、キャッシュするデータの容量を大きくすることで、転送時間を抑えることができる。また、実際の要求が予想データベースと異なり、その後リードが無かった場合に無駄となるデータ量を削減することができる。また、都度キャッシュメモリの容量を変化させることで、RAMの消費を抑えることもできる。なお、予想データベースの修正処理は、実際にリードを行うとき(後述のs29の処理時)に行ってもよい。

10

【0070】

以上のようにして、アクセス予測を行う。なお、キャッシュメモリの数および容量はユーザが手動で設定することができる。また、ライト用キャッシュメモリ、リード用キャッシュメモリの確保数もユーザが設定することができる。ユーザがリードを優先したい場合はリード用キャッシュメモリの数を増やし、ライトを優先したい場合はライト用キャッシュメモリの数を増やす。

【0071】

次に、図8において、下位側ドライバ152は、上位側から送信された要求の内容を判断する(s12)。リード、ライトの要求でなければ、対象デバイスが他のリード、ライト等の処理を行っているか否かを判断し(s13)、処理が空いていればこの要求を下位側に転送する(s14)。他の処理を行っていればこの要求をコマンドキューに登録する(s15)。コマンドキューに登録しておけば、後に処理が空いたときに実行することができる。また、FLUSH CACHEコマンドのような、ライト用キャッシュメモリのデータを書き込み指示するようなコマンドを受領した場合、このときにライト用キャッシュメモリにキャッシュされているデータを下位側へ転送する。

20

【0072】

s12において、下位側ドライバ152は、リード、ライトの要求であると判断した場合、キャッシュメモリを参照する(s16)。ここで、上位側からの要求がリード要求であり、リード用キャッシュメモリまたはライト用キャッシュメモリに該当するデータがキャッシュされていれば、これを上位側に転送し、コマンド終了を返信する(s17)。

30

【0073】

s12において、下位側ドライバ152は、上位側からの要求が連続したライト要求であると判断した場合、ライト用キャッシュメモリに連続してデータを書き込み、コマンド終了を返信する(s18)。その後、ライト用キャッシュメモリの容量が一杯であるかを判断し(s19)、容量が一杯でなければタイマを起動させ(s20)、動作を終える。

【0074】

s19において、ライト用キャッシュメモリの容量が一杯であれば、対象デバイスが他のリード、ライト等の処理を行っているか否かを判断し(s21)、処理が空いていればライト要求を下位側に送信し、キャッシュしたデータを転送する(s22)。他の処理を行っていればこのライト要求をコマンドキューに登録する(s23)。コマンドキューに登録しておけば、後に処理が空いたときに実行することができる。

40

【0075】

s16において、下位側ドライバ152は、上位側からの要求に該当するキャッシュデータがないと判断した場合、予想データベースを参照し、上位側からの要求と予想データベースの内容を比較する(s24)。上位側からの要求と予想データベースの内容が不一致(LBAや転送ブロック数が不一致)であれば、この要求をそのまま下位側に転送する(s25)。なお、このとき、下位側が他の処理を行っていればコマンドキューに登録し、後に実行するようにしてもよい。

50

【 0 0 7 6 】

上位側からの要求がライト要求であり、予想データベースの内容と一致（L B A や転送ブロック数が一致）すれば、ライト用キャッシュメモリを確保する（ s 2 6 ）。ここで、空きキャッシュメモリがあれば、そのキャッシュメモリをライト用キャッシュメモリとして確保し、キャッシュメモリが全てライト用またはリード用に用いられていた場合、最も過去に更新したキャッシュメモリの内容を消去してライト用キャッシュメモリを確保する。なお、ライト用キャッシュメモリのデータは、リード用キャッシュメモリのデータとしても使用可能である。

【 0 0 7 7 】

その後、確保したライト用キャッシュメモリに上位側から転送されるデータを保存し、コマンド終了を返信する（ s 2 7 ）。

10

【 0 0 7 8 】

s 2 4 において、下位側ドライバ 1 5 2 は、上位側からの要求がリード要求であり、予想データベースの内容と一致（L B A や転送ブロック数が一致）すると判断した場合、リード用キャッシュメモリを確保する（ s 2 8 ）。上記と同様、空きキャッシュメモリがあればこれをリード用キャッシュメモリとし、空きが無ければ最も過去に更新したキャッシュメモリの内容を消去してリード用キャッシュメモリを確保する。

【 0 0 7 9 】

その後、予想データベースに基づいて、予想分のリード要求を下位側に送信し、リード用キャッシュメモリにデータをキャッシュする（ s 2 9 ）。このとき、一度に大量のデータをキャッシュするのではなく、複数回に分けてデータキャッシュしてもよい。この場合、残りのリード要求はコマンドキューに登録する。リード要求を複数に分割することで、キャッシュ済みデータを上位側のドライバへ転送しながら、それと並行して、次に連続するリードデータをキャッシュすることができ、U S B バスの使用率向上が期待できる。

20

【 0 0 8 0 】

次に、図 9 は、下位側からのコマンド終了割り込みがあった場合の動作を示すフローチャートである。

【 0 0 8 1 】

まず、下位側ドライバ 1 5 2 は、下位側から受信したコマンド終了が、ライト用キャッシュメモリにキャッシュしておいたデータを書き終えた旨を示すコマンドであるか否かを判断する（ s 3 1 ）。ライト用キャッシュメモリのデータが書き終わった場合は、このライト用キャッシュメモリをリード用キャッシュメモリに変更する（ s 3 2 ）。なお、ライト用キャッシュメモリをリード用キャッシュメモリに変更するタイミングは、このタイミングに限るものではない。また、上述のように、ライト用キャッシュメモリのデータをリード用キャッシュメモリのデータとして用いることも無論可能である。

30

【 0 0 8 2 】

その後、下位側ドライバ 1 5 2 は、上位側に対して終了報告（転送）が必要なコマンドであるか否かを判断する（ s 3 3 ）。必要であれば上位側にコマンド終了を返送する（ s 3 4 ）。

【 0 0 8 3 】

さらに、下位側ドライバ 1 5 2 は、コマンドキュー 1 6 0 を参照し、キューイングされたコマンドが存在するか否かを判断する（ s 3 5 ）。キューイングされたコマンドが存在すれば、これを下位側に転送する（ s 3 6 ）。

40

【 0 0 8 4 】

次に、図 1 0 は、タイマ割り込み、ドライバ初期化時、ドライバ終了時の動作を示すフローチャートである。

【 0 0 8 5 】

まず、同図（ A ）は、タイマ割り込み動作を示すフローチャートである。下位側ドライバ 1 5 2 は、タイマがタイムアップするとこの動作を開始する。下位側ドライバ 1 5 2 は、ライト用キャッシュメモリにデータがキャッシュされているか否かを確認する（ s 4 1

50

）。キャッシュされていれば、下位側にライト要求を送信し、ライト用キャッシュメモリにキャッシュされているデータを下位側に転送し（s 4 2）、動作を終える。ライト用キャッシュメモリにデータがキャッシュされていなければ動作を終える。

【 0 0 8 6 】

同図（B）は、ドライバ終了時の動作を示すフローチャートである。P C のシャットダウン時、外付H D D 2の接続を解除する時にこの動作を開始する。下位側ドライバ1 5 2は、履歴データベースを外付H D D 2またはO S 1 1に転記して保存する（s 5 1）。

【 0 0 8 7 】

同図（C）は、ドライバ初期化時の動作を示すフローチャートである。P C 1を再起動したり、外付H D D 2を再接続した場合にこの動作を開始する。下位側ドライバ1 5 2は、履歴データベースを外付H D D 2またはO S 1 1から読み出し、R A Mに展開してデータベース1 5 7を構築する（s 6 1）。

【 0 0 8 8 】

なお、上記実施形態では、上位側ドライバ1 5 1、下位側ドライバ1 5 2を実現し、汎用ディスクドライバ5 0 2、またはU S B マスストレージドライバ5 0 3に接続される例を示したが、U S B マスストレージドライバを専用ドライバに置き換える形でも、本発明のデバイスコントローラを実現することができる。図 1 1 は、U S B マスストレージドライバを専用ドライバに置き換える場合の例を示す機能ブロック図である。なお、図 4 と共通する構成については同一の符号を付し、その説明を省略する。

【 0 0 8 9 】

この例では、U S B ホストコントローラドライバ5 0 4、汎用ディスクドライバ5 0 2、およびアプリケーション1 0 1に接続される専用U S B マスストレージクラスドライバ1 7 1を備えている。専用U S B マスストレージクラスドライバ1 7 1は、従来の標準のU S B マスストレージドライバを置き換えるものである（図 1 を参照）。

【 0 0 9 0 】

この専用U S B マスストレージクラスドライバ1 7 1は、上記実施形態と同様に、キャッシュメモリを内蔵しており、上位側から受信したリード、ライト要求に応じてデータをキャッシュする。専用マスストレージクラスドライバ1 7 1の動作は図 8 から図 1 0 に示した下位側ドライバの動作と同様である。

【 0 0 9 1 】

この例においても、アプリケーション1 0 1が6 4 k B 毎にデータを分割して、リード/ライトの要求を発行する場合は、専用U S B マスストレージクラスドライバ1 7 1がキャッシュメモリにデータをキャッシュし、データ転送する。

【 0 0 9 2 】

なお、この専用U S B マスストレージクラスドライバ1 7 1は、6 4 k B のデータ転送サイズの上限を持たず、6 4 k B より大きいサイズのリード/ライトの要求を処理可能なものである。したがって、アプリケーション1 0 1、または汎用ディスクドライバ5 0 1は、リード/ライトの要求を6 4 k B 単位に分割することなく発行することが可能であるため、コマンド、ステータスの処理の遅延が少なく済み、データ転送を高速にすることができる。

【 符号の説明 】

【 0 0 9 3 】

1 - P C

2 - 外付H D D

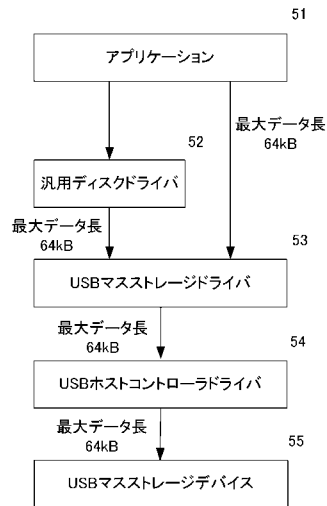
10

20

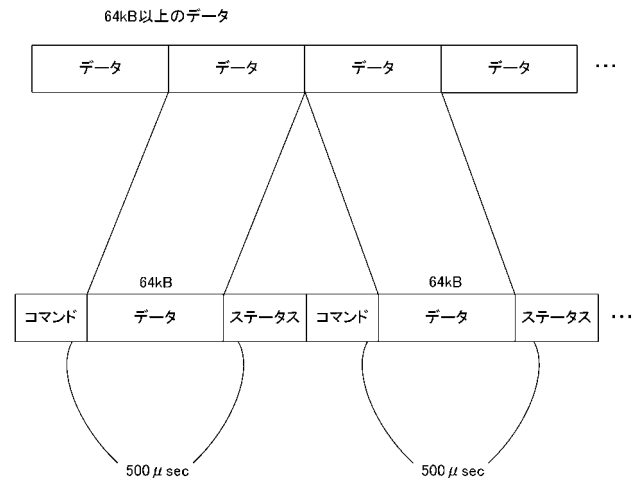
30

40

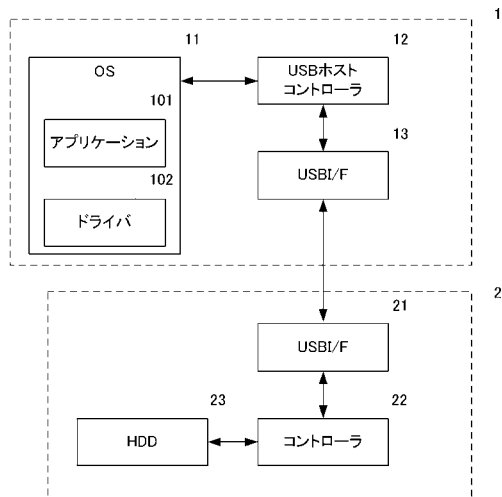
【図 1】



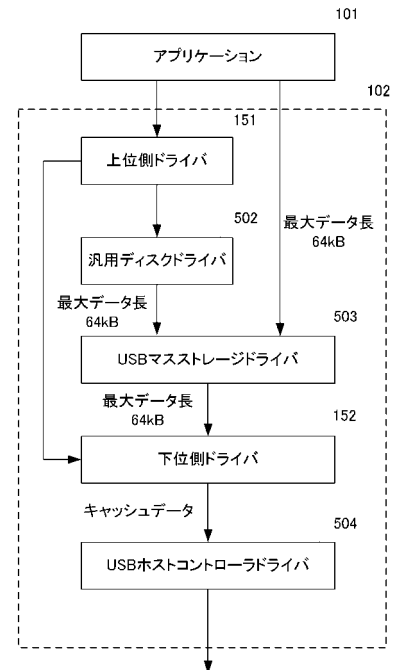
【図 2】



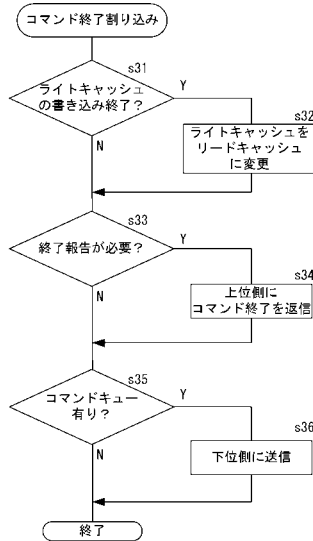
【図 3】



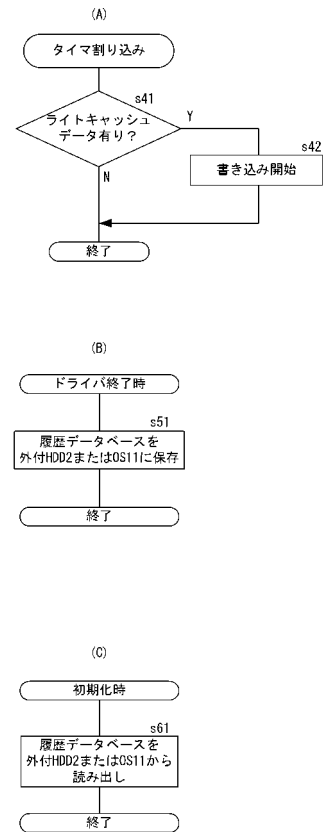
【図 4】



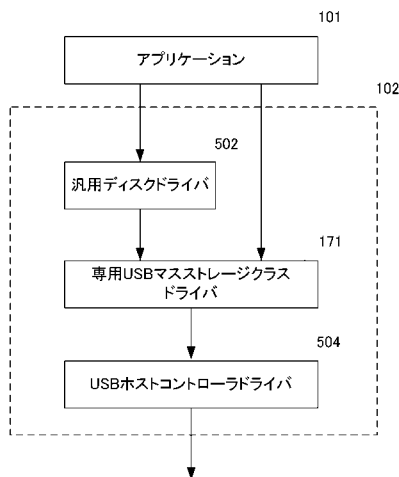
【図 9】



【図 10】



【図 11】



【図 12】

