(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6197585号
(P6197585)

(45)発行日　平成29年9月20日(2017.9.20)　(24)登録日　平成29年9月1日(2017.9.1)

(51)Int.Cl.		F I		
G06F 9/46	(2006.01)	G06F	9/46	350
G06F 9/52	(2006.01)	G06F	9/46	475Z
G06F 11/20	(2006.01)	G06F	11/20	
G06F 13/10	(2006.01)	G06F	13/10	330C

請求項の数 9　(全 44 頁)

(21)出願番号　特願2013-228182(P2013-228182)
(22)出願日　平成25年11月1日(2013.11.1)
(65)公開番号　特開2015-88103(P2015-88103A)
(43)公開日　平成27年5月7日(2015.5.7)
審査請求日　平成28年7月5日(2016.7.5)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100092978
弁理士　真田　有
(74)代理人　100112678
弁理士　山本　雅久
(72)発明者　山口　博史
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　原　忠

最終頁に続く

(54)【発明の名称】情報処理装置、及び、情報処理装置の制御方法

(57)【特許請求の範囲】

【請求項１】

入出力デバイスを用いてデータ入出力を行なうドライバを含む仮想計算機と、前記仮想計算機を管理するとともに本情報処理装置と対になる他の情報処理装置との同期処理を行なうハイパバイザと、を構築するプロセッサと、

メモリと前記入出力デバイスとの間で行なわれるアクセスを監視する監視部と、

前記監視部が前記メモリから前記入出力デバイスへのデータ出力に係る読出アクセスを検出した場合、前記読出アクセスを検出した旨を前記ハイパバイザに通知する通知部と、

前記ハイパバイザが前記通知部からの通知を受けるときに前記同期処理の実行中である場合、前記データ出力を抑止する抑止部と、

前記メモリにおいてデータの変更が発生した箇所の数を計数するカウンタと、

前回の同期処理時に参照された前記カウンタによる計数値を保存する保存領域と、を有し、

前記ハイパバイザが前記通知部からの通知を受けたときに前記同期処理の実行中でない場合、前記監視部は、前記カウンタの値と前記保存領域の計数値とを比較し、前記カウンタの値と前記保存領域の計数値とが不一致であれば、前記抑止部に前記データ出力を抑止させ、前記ハイパバイザに前記同期処理を実行させる、情報処理装置。

【請求項２】

前記抑止部は、前記ハイパバイザが前記同期処理を終了すると、前記データ出力の抑止を解除する、請求項１記載の情報処理装置。

【請求項３】
前記読出アクセスは、前記入出力デバイスから前記メモリへの読出リクエストの発行、または、前記メモリから前記入出力デバイスへの前記データ出力の対象データの読出である、請求項１または請求項２に記載の情報処理装置。
【請求項４】
前記抑止部は、前記読出リクエストまたは前記対象データをバッファに一時的に保持することにより前記データ出力を抑止し、前記ハイパバイザが前記同期処理を終了すると、前記読出リクエストを前記バッファから前記メモリ側へ送信、または、前記対象データを前記バッファから前記入出力デバイスへ送信する、請求項３記載の情報処理装置。
【請求項５】
前記監視部は、前記入出力デバイスから前記メモリへのデータ書込に係る書込アクセスを検出した場合、前記メモリを管理するテーブルに、前記データ書込に係るアドレスに対応するデータに変更が生じたことを記録し、
前記ハイパバイザは、前記テーブルに従って前記同期処理を行なう、請求項１～請求項４のいずれか一項に記載の情報処理装置。
【請求項６】
前記監視部は、前記テーブルにおいて変更ビットを設定することにより前記変更が生じたことを記録し、
前記ハイパバイザは、前記同期処理を終了すると、前記変更ビットをクリアする、請求項５記載の情報処理装置。
【請求項７】
前記監視部は、前記カウンタの値と前記保存領域の計数値とが一致していれば、前記抑止部に前記データ出力を抑止させることなく、前記データ出力を実行させる、請求項１～請求項６のいずれか一項に記載の情報処理装置。
【請求項８】
前記監視部が、前記ハイパバイザに前記同期処理を実行させるべく、前記読出アクセスを検出したことを要因とする割込みを発生させた場合に設定される一方、前記ハイパバイザが前記同期処理を終了するとクリアされる割込み要因フラグを有し、
前記監視部は、前記割込み要因フラグが設定されている場合に前記ハイパバイザが前記同期処理の実行中であると判断する一方、前記割込み要因フラグが未設定の場合に前記ハイパバイザが前記同期処理の実行中ではないと判断する、請求項１～請求項７のいずれか一項に記載の情報処理装置。
【請求項９】
入出力デバイスを用いてデータ入出力を行なうドライバを含む仮想計算機と、前記仮想計算機を管理するとともに本情報処理装置と対になる他の情報処理装置との同期処理を行なうハイパバイザと、を有する情報処理装置の制御方法であって、
メモリと前記入出力デバイスとの間で行なわれるアクセスを監視し、
前記監視によって前記メモリから前記入出力デバイスへのデータ出力に係る読出アクセスを検出した場合、前記読出アクセスを検出した旨を前記ハイパバイザに通知し、
前記ハイパバイザが前記通知を受けるときに前記同期処理の実行中である場合、前記データ出力を抑止するとともに、
前記メモリにおいてデータの変更が発生した箇所の数をカウンタによって計数し、
前回の同期処理時に参照された前記カウンタによる計数値を保存領域に保存し、
前記ハイパバイザが前記通知を受けたときに前記同期処理の実行中でない場合、前記カウンタの値と前記保存領域の計数値とを比較し、前記カウンタの値と前記保存領域の計数値とが不一致であれば、前記データ出力を抑止させ、前記ハイパバイザに前記同期処理を実行させる、情報処理装置の制御方法。
【発明の詳細な説明】
【技術分野】
【０００１】

本発明は、情報処理装置、及び、情報処理装置の制御方法に関する。

【背景技術】

【０００２】

フォールトトレラント（ＦＴ）サーバは、イーサネット（登録商標）等によって相互接続された、同じ構成の二組のハードウェアを有している。したがって、ＦＴサーバでは、一方のハードウェアが故障しても、ユーザに故障の発生を気づかせることなく、他方のハードウェアによって動作を継続することが可能になっている。上述のようなＦＴサーバを実現する手法としては、ハイパバイザを利用しない手法（特許文献１～３）と、ハイパバイザを利用する手法（特許文献４，５）とがある。

【０００３】

ハイパバイザを利用しないＦＴサーバとしては、二組のＣＰＵ（Central Processing Unit）に同一のクロックを与えて同一の動作をさせることにより二組のシステムの同期をとるものがある（特許文献２）。ＤＭＡ（Direct Memory Access）時におけるＣＰＵサブシステムから物理デバイス（例えば入出力コントローラ）へのデータは、各ＣＰＵサブシステムから入出力ルータに送られ、入出力ルータは、そのデータを、ＦＩＦＯ（First In First Out）形式のバッファをもつ比較器へ送る。比較器では、二つのＣＰＵサブシステムから受け取ったデータが一致していることが確認され、一つのデータが入出力コントローラへ送られる。上述のようなＦＴサーバを実現するためには、以下の理由(a1)～(a3)により、二つのＣＰＵサブシステムと各入出力コントローラとが物理的に分離されている必要がある。

【０００４】

(a1) ＦＴサーバとして単一故障点を避けるために、二つのＣＰＵサブシステムは、物理的に分離している必要がある。

(a2) ＣＰＵサブシステムと入出力コントローラとの間に入出力ルータや比較器を配置する必要がある。

(a3) 二つのＣＰＵサブシステムによって一つの入出力コントローラを共有する必要がある。

【０００５】

また、ハイパバイザを利用しないＦＴサーバとしては、二組のＣＰＵ（演算部）に別々のクロックを与えてクロック管理部により同一の動作をさせることで二組のシステムの同期をとるものもある（特許文献３）。ＤＭＡ時におけるＣＰＵサブシステム（演算部）から物理デバイス（ＩＯ(Input/Output)デバイス）へのデータは、各ＣＰＵサブシステムからＩＯ比較部へ送られる。ＩＯ比較部では、二つのＣＰＵサブシステムから受け取ったデータが一致していることが確認され、一つのデータが入出力コントローラへ送られる。ＩＯデバイスからのデータは、ＦＴ制御部によって受け取られ、クロックのずれを考慮したタイミングで各ＣＰＵサブシステム（演算部）へ送られる。上述のようなＦＴサーバを実現するためには、以下の理由(b1)～(b3)により、二つのＣＰＵサブシステム（演算部）とＩＯデバイスとが物理的に分離されている必要がある。

【０００６】

(b1) ＦＴサーバとして単一故障点を避けるために、二つのＣＰＵサブシステム（演算部）は、物理的に分離している必要がある。

(b2) ＣＰＵサブシステムとＩＯデバイスとの間にＦＴ制御部やＩＯ比較部を配置する必要がある。

(b3) 二つのＣＰＵサブシステムによって一つのＩＯデバイスを共有する必要がある。

【０００７】

近年、ＣＰＵサブシステム内に入出力コントローラ（物理デバイス，入出力デバイス）を内蔵するSystem on Chip（ＳｏＣ）化が進行している。しかし、上述のごとくハイパバイザを利用しないＦＴサーバでは、二つのＣＰＵサブシステム（演算部）と各入出力コントローラまたはＩＯデバイスとが物理的に分離されている必要がある。そのため、上述のごとくハイパバイザを利用しないＦＴサーバは、ＣＰＵサブシステム内に入出力デバイス

を備えた構成、つまりＳｏＣ化されたＣＰＵを採用した構成に適用することができない。
【０００８】
一方、ハイパバイザを利用するＦＴサーバは、ハイパバイザ上に構築された仮想計算機（仮想マシン，ＶＭ：Virtual Machine）をフォルトトレラントの対象としている。ハイパバイザを利用するＦＴサーバでは、故障発生時に一方のハードウェア（プライマリ）から他方のハードウェア（セカンダリ）へ処理を引き継ぐべく、外部へのデータの出力を把握する必要がある。このため、Ｉ／Ｏデバイスが仮想デバイスとしてエミュレートされ、外部へのデータ出力が全てハイパバイザ経由で出力されている。つまり、Ｉ／Ｏデバイスがハイパバイザによって仮想化され、ＶＭ上のＯＳ（Operating System）が必ずハイパバイザの仮想Ｉ／Ｏデバイスにアクセスするように制御する。これにより、ハイパバイザは、外部へのデータ出力を把握することができる。
【先行技術文献】
【特許文献】
【０００９】
【特許文献１】特開２００６－１７８６３６号公報
【特許文献２】特開２００８－２３４１４１号公報
【特許文献３】特開２００９－１９３５０４号公報
【特許文献４】特開２００９－８０６９５号公報
【特許文献５】特開２０１２－３３１３号公報
【発明の概要】
【発明が解決しようとする課題】
【００１０】
上述したように、ハイパバイザを利用しないＦＴサーバは、ＳｏＣ化されたＣＰＵを採用した構成に適用することができない。
これに対し、ハイパバイザを利用するＦＴサーバは、ＳｏＣ化されたＣＰＵを採用した構成に適用することができる。しかし、外部へのデータの出力を把握する必要がある。このため、Ｉ／Ｏデバイスが仮想デバイスとしてエミュレートされ、外部へのデータ出力が全てハイパバイザ経由で出力される。その結果、仮想化のオーバヘッドにより、ＶＭの入出力アクセス性能、特に、Ｉ／Ｏデバイスからの出力性能（外部へのデータ出力のパフォーマンス）が劣化する。
【００１１】
一つの側面で、本発明は、仮想計算機の入出力アクセス性能を改善することを目的とする。
なお、前記目的に限らず、後述する発明を実施するための最良の形態に示す各構成により導かれる作用効果であって、従来の技術によっては得られない作用効果を奏することも本件の他の目的の一つとして位置付けることができる。
【課題を解決するための手段】
【００１２】
一つの案では、情報処理装置は、入出力デバイスを用いてデータ入出力を行なうドライバを含む仮想計算機と前記仮想計算機を管理するとともに本情報処理装置と対になる他の情報処理装置との同期処理を行なうハイパバイザとを構築するプロセッサと、メモリと前記入出力デバイスとの間で行なわれるアクセスを監視する監視部と、前記監視部が前記メモリから前記入出力デバイスへのデータ出力に係る読出アクセスを検出した場合、前記読出アクセスを検出した旨を前記ハイパバイザに通知する通知部と、前記ハイパバイザが前記通知部からの通知を受けるときに前記同期処理の実行中である場合、前記データ出力を抑止する抑止部と、前記メモリにおいてデータの変更が発生した箇所の数を計数するカウンタと、前回の同期処理時に参照された前記カウンタによる計数値を保存する保存領域と、を有し、前記ハイパバイザが前記通知部からの通知を受けたときに前記同期処理の実行中でない場合、前記監視部は、前記カウンタの値と前記保存領域の計数値とを比較し、前記カウンタの値と前記保存領域の計数値とが不一致であれば、前記抑止部に前記データ出

力を抑止させ、前記ハイパバイザに前記同期処理を実行させる。
【発明の効果】
【００１３】
一実施形態によれば、仮想計算機の入出力アクセス性能を改善することができる。
【図面の簡単な説明】
【００１４】
【図１】ハイパバイザを利用するＦＴサーバの概念を示すブロック図である。
【図２】（Ａ）はハイパバイザを利用する場合のメモリマップを示す図、（Ｂ）はハイパバイザを利用しない場合のメモリマップを示す図である。
【図３】ＤＭＡリマップ機能の概要を説明する図である。
【図４】第１実施形態のＦＴサーバ（情報処理装置）のハードウェア構成および機能構成を示すブロック図である。
【図５】図４に示すＦＴサーバにおけるＤＭＡreadリクエスト監視部の処理を説明するフローチャートである。
【図６】図４に示すＦＴサーバにおけるＤＭＡreadデータ監視部の処理を説明するフローチャートである。
【図７】図４に示すＦＴサーバにおけるＤＭＡ割込み制御部の処理を説明するフローチャートである。
【図８】図４に示すＦＴサーバにおけるＤＭＡ割込み制御部の処理を説明するフローチャートである。
【図９】第２実施形態のＦＴサーバ（情報処理装置）のハードウェア構成および機能構成を示すブロック図である。
【図１０】図９に示すＦＴサーバにおけるＰＣＩｅ（Peripheral Components Interconnect express）トランザクションの例を説明する図である。
【図１１】図９に示すＦＴサーバにおけるシステム外へのデータ送出処理の概要を説明するフローチャートである。
【図１２】図９に示すＦＴサーバにおけるデータ転送時のメモリマップ例を示す図である。
【図１３】図９に示すＦＴサーバにおけるＶＭアドレスと物理アドレスとのメモリマッピング例を示す図である。
【図１４】図９に示すＦＴサーバにおけるＤＭＡreadリクエスト監視部の処理を説明するフローチャートである。
【図１５】（Ａ）はＰＣＩｅトランザクションリクエストヘッダのフォーマット例を示す図、（Ｂ）はＰＣＩｅトランザクション完了ヘッダのフォーマット例を示す図、（Ｃ）は図９に示すＦＴサーバにおけるＤＭＡ割込み制御部の監視対象ＰＣＩｅデバイス管理テーブルを示す図である。
【図１６】図９に示すＦＴサーバにおけるＤＭＡreadデータ監視部の処理を説明するフローチャートである。
【図１７】図９に示すＦＴサーバにおけるＤＭＡ割込み制御部の処理を説明するフローチャートである。
【図１８】図９に示すＦＴサーバにおけるＤＭＡ割込み制御部の処理を説明するフローチャートである。
【図１９】図９に示すＦＴサーバにおけるＤＭＡwrite監視部の処理を説明するフローチャートである。
【図２０】図９に示すＦＴサーバにおけるＶＭメモリ管理テーブルの例を示す図である。
【発明を実施するための形態】
【００１５】
以下、図面を参照して実施の形態を説明する。
〔１〕概要
〔１－１〕ハイパバイザを利用するＦＴサーバ

フォルトトレラントサーバ（ＦＴサーバ）とは、同じ構成の二組のハードウェアをそなえ、一方のハードウェアが故障しても、ユーザに故障の発生を気づかせることなく、他方のハードウェアによって動作を継続することが可能なコンピュータシステムである。このようなＦＴサーバを実現する手法の一つとして、ハイパバイザを利用する手法があり、当該手法は、ハイパバイザ上の仮想マシン（ＶＭ）をフォルトトレラント対象としている。図１は、ハイパバイザを利用するＦＴサーバの概念を示すブロック図である。

【００１６】

図１に示すＦＴサーバ１００では、同じ構成の二組のハードウェア１０１，１０１が、イーサネットなどのインタコネクト１０２を介して接続されている。ここでは、一方のハードウェア１０１をプライマリと呼び、他方のハードウェア１０１をセカンダリと呼ぶ。プライマリおよびセカンダリのそれぞれにおいてはハイパバイザ１０３が構築され、ハイパバイザ１０３によって管理ＶＭ１０４とＶＭ１０５とが構築され管理される。ここで、ＶＭ１０５が保護対象つまりフォールトトレラント対象であり、ＶＭ１０５を対象とするＦＴ機能１１０がハイパバイザ１０３に備えられている。なお、以下では、プライマリ側の機能，構成および動作について説明するが、セカンダリ側の機能，構成および動作はプライマリ側と同様であり、その説明や図示は省略する。

【００１７】

ＦＴ機能１１０は、プライマリハードウェア１０１で故障が発生した場合、プライマリ側のＶＭ１０５の処理をセカンダリ側のＶＭ１０５に引き継がせるように機能する。ＦＴ機能１１０は、例えば、プライマリ側のＶＭ１０５の状態（ＣＰＵコンテキストやメモリの内容など）をセカンダリ側に定期的にコピーする。これにより、ＦＴ機能１１０は、コピーした時点の状態をセカンダリ側に保持させ、プライマリが故障してもセカンダリが故障の発生直後から処理を再開できるようにしている。

【００１８】

〔１－２〕ハイパバイザによるＩ／Ｏデバイスの仮想化

上述のごとくハイパバイザを利用してＦＴサーバを実現する手法では、Ｉ／Ｏデバイス（入出力デバイス，物理デバイス）１２０が、ハイパバイザ１０３によって、仮想デバイス１３０として仮想化され、ＶＭ１０５内のＯＳからはＩ／Ｏデバイス１２０に直接アクセスすることはできない。Ｉ／Ｏデバイス１２０は、例えばイーサネットカード，ファイバチャネルカードである。Ｉ／Ｏデバイス１２０を仮想化することにより、ハイパバイザ１０３は、Ｉ／Ｏデバイス１２０による入出力アクセス（Ｉ／Ｏ）を監視することができる。

【００１９】

ここで、フォルトトレラント性を維持するためには、プライマリのＩ／Ｏデバイス１２０から外部へ何らかの出力を行なった直後にプライマリで故障が発生してもセカンダリが処理を引き継げなくてはならないという要件を満たす必要がある。当該要件は、一般に、“output commit”問題と呼ばれる。“output commit”問題に対応するためには、プライマリのＶＭ１０５が外部へ何らかの出力を行なう際、セカンダリのＶＭ１０５はプライマリと同じ状態になっている、つまり同期している必要がある。

【００２０】

そのため、ハイパバイザ１０３がＩ／Ｏを監視し、プライマリの状態とセカンダリの状態とが同期するまでＩ／Ｏデバイス１２０による出力を停止するという処理を行なう。当該処理を実現すべく、ハイパバイザ１０３は、Ｉ／Ｏデバイス１２０を仮想デバイス１３０として仮想化し、ＶＭ１０５上のＯＳが必ずハイパバイザ１０３の仮想デバイス１３０にアクセスするように制御する。これにより、ハイパバイザ１０３は、Ｉ／Ｏデバイス１２０による出力を把握することができる。

【００２１】

なお、図１に示す例では、ハードウェア１０１に、物理的なＩ／Ｏデバイス（物理デバイス）１２０が備えられ、ハイパバイザ１０３は、物理デバイス１２０を仮想デバイス１３０として仮想化する。ＶＭ１０５上のＯＳは、仮想デバイスドライバ１４０によって仮

想デバイス１３０にアクセスする（図１の矢印Ａ１参照）。仮想デバイス１３０へのアクセスは、管理ＶＭ１０４上の物理デバイスドライバ１５０によって物理デバイス１２０へのアクセスに変換される（図１の矢印Ａ２，Ａ３参照）。このような二段階処理により、Ｉ／Ｏデバイス１２０からの出力性能は、通常のサーバに比べて劣化する。

【００２２】

〔１－３〕Ｉ／Ｏデバイス（物理デバイス）の仮想化による性能劣化

性能劣化に特に影響するのは、上述した二段階処理に伴うデータのコピー処理である。以下、図１および図２（Ａ）を参照しながら、例えば、ネットワークパケットをＩ／Ｏデバイス１２０から外部へ出力する場合の処理手順(c1)～(c4)について説明する。なお、図２（Ａ）は、ハイパバイザ１０３を利用する場合のメモリマップを示す図である。

【００２３】

(c1) 図２（Ａ）に示すように、ネットワークパケットは、ＶＭ１０５のメモリ空間に仮想デバイスドライバ１４０によって用意される。

【００２４】

(c2) 仮想デバイスドライバ１４０がネットワークパケットの送信を仮想デバイス１３０に指示すると（図１の矢印Ａ１参照）、ハイパバイザ１０３は、その指示を捉え、ネットワークパケットを、管理ＶＭ１０４がアクセス可能な管理ＶＭ１０４のメモリ空間（ＶＭ１０５からの受信領域）にコピーする（図２（Ａ）の矢印Ａ４参照）。その後、ハイパバイザ１０３は、管理ＶＭ１０４にネットワークパケットの送信の指示があったことを通知する（図１の矢印Ａ２参照）。

【００２５】

(c3) 管理ＶＭ１０４は、ネットワークパケットを、物理デバイス１２０が送信可能な形式に整形してから、物理デバイス１２０への送信領域にコピーする（図２（Ａ）の矢印Ａ５参照）。その後、管理ＶＭ１０４は、物理デバイス１２０にパケットの送信指示を行なう（図１の矢印Ａ３参照）。

【００２６】

(c4) 物理デバイス１２０は、管理ＶＭ１０４からの送信指示を受け、物理デバイス１２０への送信領域からパケットデータをＤＭＡ転送して物理デバイス１２０内に取り込む（図２（Ａ）の矢印Ａ６参照）。

【００２７】

なお、図１では、物理デバイスドライバ１５０が管理ＶＭ１０４に備えられている場合について説明しているが、物理デバイスドライバ１５０は、管理ＶＭ１０４ではなく、ハイパバイザ１０３に備えられてもよい。

【００２８】

一方、ハイパバイザのない、通常のサーバ環境では、図２（Ｂ）に示すように、物理デバイスは、ＯＳのメモリ空間から、直接、ＤＭＡ転送を行なう（矢印Ａ７参照）。これに対し、図１に示すようにハイパバイザ１０３による物理デバイス１２０の仮想化を行なうと、通常のサーバ環境よりも多くの処理つまり上述した二段階処理が必要となるため、物理デバイス１２０による出力性能が、ハイパバイザを利用しない通常のサーバよりも劣化する。なお、図２（Ｂ）は、ハイパバイザを利用しない場合のメモリマップを示す図である。

【００２９】

また、物理デバイスを仮想化することによるデメリットとしては、上述した性能劣化のほかに、使用可能な物理デバイスの種類が制限されるというデメリットもある。ＶＭ内のＯＳが使用可能なデバイスは、仮想デバイスとしてエミュレートしている機能がサポートしている物理デバイスのみである。したがって、広く採用されていないデバイスや最新技術のデバイスをＶＭ内のＯＳで使用するためには、これらのデバイスに対応するエミュレーション機能を作り込む必要があり、コストと時間がかかる。

【００３０】

〔１－４〕ＤＭＡリマップ機能

ところで、近年のプロセッサやチップセットには、ＶＭによる物理デバイス利用の高速化を支援する機能が存在する。当該機能を有する技術として、例えばIntel社のVT-ｄ（Virtualization Technology for Directed I/O）やＡＭＤ社のＩＯＭＭＵ（Input/Output Memory Management Unit）が挙げられる。当該機能のうち、ＤＭＡリマップ機能は、ＶＭ上のＯＳ内のドライバが物理デバイスを直接制御できるようにする機能である。ＤＭＡリマップ機能は、具体的には以下の処理手順(d1)～(d4)に従って動作する。

【００３１】

なお、図３は、ＤＭＡリマップ機能１７０の概要を説明する図である。図３に示す例では、保護対象のＶＭ１０５に物理デバイスドライバ１５０が備えられるほか、ハードウェア１０１には、物理デバイス１２０，ＣＰＵサブシステム１６０およびＤＭＡリマップ機能１７０が備えられている。ＣＰＵサブシステム１６０は、ＣＰＵとメモリからなるシステムである。

【００３２】

(d1) 物理デバイスドライバ１５０は、物理デバイス１２０から外部へデータを送出するために、ＶＭ１０５のメモリ空間にデータを用意し、用意したデータのあるＶＭ１０５のメモリ空間のアドレスを物理デバイス１２０に通知する（図３の矢印Ａ８参照）。しかし、物理デバイス１２０は物理メモリ空間のアドレスにしかアクセスできないため、ＶＭ１０５のメモリ空間のアドレスを何らかの方法で物理メモリ空間のアドレスに変換する必要がある。

【００３３】

(d2) 物理デバイス１２０は、データを送出するために、ＶＭ１０５のメモリ空間のアドレスからデータ転送するためのＤＭＡ転送リクエストを、ＤＭＡリマップ機能１７０経由でＣＰＵサブシステム１６０へ発行する（図３の矢印Ａ９，Ａ１０参照）。

【００３４】

(d3) このとき、ＤＭＡリマップ機能１７０は、物理デバイス１２０からのＤＭＡ転送リクエストを監視する（図３の矢印Ａ９参照）。そして、ＤＭＡリマップ機能１７０は、当該リクエストを受けると、当該リクエストに含まれるＶＭ１０５のメモリ空間のアドレスを、物理メモリ空間のアドレスに変換してＣＰＵサブシステム１６０へ送り出す（図３の矢印Ａ１０参照）。

【００３５】

(d4) これにより、ＣＰＵサブシステム１６０は、物理メモリ空間から送出に必要なデータを物理デバイス１２０に送出し（図３の矢印Ａ１１参照）、物理デバイス１２０は、ＶＭ１０５のメモリ空間から、直接、ＤＭＡ転送を行なうことができる（図３の矢印Ａ１２参照）。

【００３６】

上述したＤＭＡリマップ機能１７０により、ハイパバイザ１０３による二段階処理がなくなり、ＶＭ１０５上のＯＳは通常のサーバと同じように物理デバイス１２０にアクセスできるため、性能劣化がほとんど生じない。

しかしながら、ＤＭＡリマップ機能１７０をＦＴサーバ１００で利用しようとすると、ハイパバイザ１０３がＩ／Ｏを把握できなくなるため、上述した“output commit”問題に対応することができない。

【００３７】

〔１－５〕基本構成

以下に説明する実施形態では、“output commit”問題に対応しながら、つまりプライマリとセカンダリとの同期をとりながら、ハイパバイザを利用するＦＴサーバにおいて仮想化のオーバヘッドによる入出力アクセス性能の劣化を防ぐため、仮想マシン（ＶＭ）から物理デバイスを直接操作することを可能にする仕組みが提供される。当該仕組みは、下記手段(e1)～(e5)によって実現される。これらの手段(e1)～(e5)は、図４や図９を参照しながら後述するごとく、ハードウェア上にハイパバイザおよび仮想計算機（ＶＭ）を構築されたＦＴサーバ（情報処理装置）に適用される。ここで、ＶＭは、入出力デバイス（Ｉ

／Ｏデバイス，物理デバイス）を用いてデータ入出力を行なうドライバ（物理デバイスドライバ）を含む。また、ハイパバイザは、当該ＶＭを管理するとともに、プライマリ（本情報処理装置）とセカンダリ（他の情報処理装置）との同期処理を行なう機能を果たす。

【００３８】

なお、以下の説明において、ＣＰＵサブシステムは、ＣＰＵおよび当該ＣＰＵに接続されるメモリを示す。また、ＣＰＵサブシステムは、図４や図９を参照しながら後述するごとく、主として、ＣＰＵコアと、メモリコントローラと、メモリと、ＣＰＵに物理デバイス（Ｉ／Ｏデバイス）を接続するインタフェース（例えばPCI-Express Root Complex）とを含む。

【００３９】

(e1) ＦＴサーバのＣＰＵサブシステム内においてＤＭＡを監視する手段、つまり、ＦＴサーバ外部へのデータ出力を検出することを可能にする手段。この手段(e1)は、主に、後述するＤＭＡreadリクエスト監視部１２およびＤＭＡreadデータ監視部１４によって実現される。つまり、手段(e1)は、ＣＰＵサブシステム内のメモリと入出力デバイスとの間で行なわれるアクセスを監視する監視部として機能する。特に、前記監視部（手段(e1)）は、メモリから入出力デバイスへのデータ出力に係る読出アクセスを検出する。読出アクセスとしては、入出力デバイスからメモリへの読出リクエスト（ＤＭＡreadリクエスト）の発行、または、メモリから入出力デバイスへのデータ出力の対象データ（ＤＭＡreadデータ）の読出が検出される。

【００４０】

(e2) ＦＴサーバのＣＰＵサブシステム内において外部へのデータ出力を検出した場合に当該データ出力を検出した旨をハイパバイザに通知する手段、つまり、ハイパバイザが外部へのデータ出力を検出することを可能にする手段。この手段(e2)は、後述するＤＭＡ割込み制御部１６がハイパバイザに「ＤＭＡreadリクエスト検出割込み」または「ＤＭＡreadデータ検出割込み」を通知することによって実現される。つまり、手段(e2)は、前記監視部（手段(e1)）が読出アクセスを検出した場合、読出アクセスを検出した旨をハイパバイザに通知する通知部として機能する。

【００４１】

(e3) ハイパバイザが外部へのデータ出力を検出したことに伴いＦＴサーバ内の同期処理を行なっているときに、外部へのデータ出力を抑止する手段。この手段(e3)をＦＴサーバのＣＰＵサブシステム内に備えることで、同期処理中に外部へデータを出力してしまいプライマリとセカンダリとの同期状態が壊れてしまうことが防止される。手段(e3)は、主に、後述するＤＭＡreadリクエストバッファ部１３およびＤＭＡreadデータバッファ部１５によって実現される。つまり、手段(e3)は、ハイパバイザが前記通知部（手段(e2)）からの通知を受けるときに同期処理の実行中である場合、データ出力を抑止する抑止部として機能する。また、前記抑止部（手段(e3)）は、ハイパバイザが同期処理を終了すると、データ出力の抑止を解除する。特に、前記抑止部（手段(e3)）は、前記読出リクエストまたは前記対象データをバッファ部１３または１５に一時的に保持することによりデータ出力を抑止する。そして、前記抑止部（手段(e3)）は、ハイパバイザが同期処理を終了すると、前記読出リクエストをバッファ部１３からメモリ側へ送信、または、前記対象データをバッファ部１５から入出力デバイスへ送信する。

【００４２】

(e4) Ｉ／ＯデバイスからハイパバイザがＶＭのメモリへのＤＭＡを監視して記録する手段。この手段(e4)をＦＴサーバのＣＰＵサブシステム内に備えることで、同期処理を実行するために必要なメモリの情報をハイパバイザが把握することができる。手段(e4)は、主に、後述するＤＭＡwrite監視部１１ａによって実現される。手段(e4)としての機能は前記監視部に含まれる。前記監視部（手段(e4)）は、入出力デバイスからメモリへのデータ書込に係る書込アクセスを検出した場合、前記メモリを管理するテーブル（後述するＶＭメモリ管理テーブル２２）に、前記データ書込に係るアドレスに対応するデータに変更が生じたことを記録する。ハイパバイザは、前記テーブルに従って同期処理を行

なう。特に、前記監視部（手段(e4)）は、前記テーブルにおいて変更ビット（後述するDirtyビット）を設定することにより前記変更が生じたことを記録し、ハイパバイザは、同期処理を終了すると、変更ビットをクリアする。

【００４３】

(e5) 同期処理が必要ないときには、外部へのデータ出力をハイパバイザに通知しない手段。この手段(e5)をＦＴサーバのＣＰＵサブシステム内に備えることで、不要な同期処理が抑制され、ＦＴサーバの全体の性能が向上する。手段(e5)は、主に、後述するＤＭＡreadリクエスト監視部１２，ＤＭＡreadデータ監視部１４，Dirtyカウンタ１７ａおよび保存領域１７ｂによって実現される。ここで、カウンタ１７ａは、メモリにおいてデータの変更が発生した箇所の数を計数する。保存領域１７ｂは、前回の同期処理時に参照されたカウンタ１７ａの計数値を保存する。そして、ハイパバイザが前記通知部（手段(e2)）からの通知を受けたときに同期処理の実行中でない場合、前記監視部（手段(e1)）は、カウンタ１７ａの値と保存領域１７ｂの計数値とを比較する。カウンタ１７ａの値と保存領域１７ｂの計数値とが不一致であれば、前記監視部（手段(e1)）は、前記抑止部（手段(e3)）にデータ出力を抑止させ、ハイパバイザに同期処理を実行させる。カウンタ１７ａの値と保存領域１７ｂの計数値とが一致していれば、前記監視部（手段(e1)）は、前記抑止部（手段(e3)）にデータ出力を抑止させることなく、データ出力を実行させる。

【００４４】

なお、前記手段(e2)を実現する、後述のＤＭＡ割込み制御部１６には、割込み要因フラグ１６ａが備えられている。割込み要因フラグ１６ａは、前記監視部（手段(e1)）が、ハイパバイザに同期処理を実行させるべく、前記読出アクセスを検出したことを要因とする割込みを発生させた場合に設定される。一方、割込み要因フラグ１６ａは、ハイパバイザが同期処理を終了するとクリアされる。そして、前記監視部（手段(e1)）は、割込み要因フラグ１６ａが設定されている場合にハイパバイザが同期処理の実行中であると判断する一方、割込み要因フラグ１６ａが未設定の場合にハイパバイザが同期処理の実行中ではないと判断する。

【００４５】

上述した手段(e1)～(e5)を備えることにより、ハイパバイザがＩ／Ｏデバイスを仮想デバイスとしてエミュレートすることなく、外部へのデータの出力を検出することが可能になる。したがって、ハイパバイザはＩ／Ｏデバイスを仮想デバイスとしてエミュレートする必要がなくなり、ハイパバイザが提供するＶＭ内の物理デバイスドライバがＩ／Ｏデバイスを直接操作できるので、外部へのデータ出力のパフォーマンスが改善される。これにより、ＶＭの入出力アクセス性能が改善される。

【００４６】

また、上述した手段(e1)～(e5)を備えることにより、ＦＴサーバで使用する物理デバイスに対応するエミュレーション機能を作り込むことなく、広く採用されていないデバイスや最新技術のデバイスであっても、ＶＭ内のＯＳで使用することが可能になる。したがって、ハイパバイザを利用するＦＴサーバにおいて、通常のサーバで採用される物理デバイスの種類と同じ種類の物理デバイスを、容易に扱うことができる。

【００４７】

さらに、本実施形態（後述する第１実施形態および第２実施形態）のＦＴサーバは、ＣＰＵサブシステム内にＩ／Ｏデバイスが備えられた構成にも適用できる。つまり、本実施形態のＦＴサーバは、ＳｏＣ化されたＣＰＵを採用した構成にも適用され、高性能なＦＴサーバを実現することができる。

【００４８】

〔２〕第１実施形態のＦＴサーバ

以下、図４～図８を参照しながら第１実施形態のＦＴサーバ（情報処理装置）１について説明する。

【００４９】

〔２－１〕第１実施形態のＦＴサーバの構成

まず、図４を参照しながら第１実施形態のＦＴサーバ１の構成について説明する。図４は、第１実施形態のＦＴサーバ１のハードウェア構成および機能構成を示すブロック図である。

【００５０】

図４に示す第１実施形態のＦＴサーバ１では、同じ構成の二組のハードウェア１０，１０が、イーサネットなどのインタコネクト５０を介して接続されている。ここでは、一方のハードウェア１０をプライマリと呼び、他方のハードウェア１０をセカンダリと呼ぶ。プライマリおよびセカンダリのそれぞれにおいてはハイパバイザ２０が構築され、ハイパバイザ２０によって管理ＶＭ３０とＶＭ４０とが構築され管理される。なお、以下では、プライマリ側の機能，構成および動作について説明するが、セカンダリ側の機能，構成および動作はプライマリ側と同様であるので、その説明や図示は省略する。

【００５１】

ここで、ＶＭ４０が、保護対象つまりフォールトトレラント対象であり、ＶＭ４０上のＯＳ内には、後述する物理デバイス１０ｃを制御する物理デバイスドライバ４１が備えられ、ＶＭ４０から物理デバイス１０ｃを直接制御できるようになっている。また、ハイパバイザ２０には、ＶＭ４０を対象とするＦＴの制御を行なうＦＴ制御部２１と、ＶＭメモリ空間（メモリ１０ｂ）を管理するＶＭメモリ管理テーブル２２とが備えられている。

【００５２】

ＦＴ制御部２１は、プライマリハードウェア１０で故障が発生した場合、プライマリ側のＶＭ４０の処理をセカンダリ側のＶＭ４０に引き継がせるように機能する。ＦＴ制御部２１は、例えば、プライマリ側のＶＭ４０の状態（ＣＰＵコンテキストやメモリの内容など）をセカンダリ側に定期的にコピーすることで、プライマリとセカンダリとの同期処理を行なう。これにより、ＦＴ制御部２１は、コピーした時点の状態をセカンダリ側に保持させ、プライマリが故障してもセカンダリが故障の発生直後から処理を再開できるようにしている。

【００５３】

ハードウェア１０には、ＣＰＵとメモリからなるＣＰＵサブシステム１０Ａが備えられている。ＣＰＵサブシステム１０Ａは、ＣＰＵコア１０ａと、メモリおよびメモリコントローラ１０ｂと、物理デバイス１０ｃとを含んでいる。つまり、第１実施形態のＦＴサーバ１は、ＳｏＣ化されたＣＰＵを採用した構成に適用されている。さらに、ＣＰＵサブシステム１０Ａは、ＤＭＡ制御部１１，ＤＭＡreadリクエスト監視部１２，ＤＭＡreadリクエストバッファ部１３，ＤＭＡreadデータ監視部１４，ＤＭＡreadデータバッファ部１５，ＤＭＡ割込み制御部１６およびメモリアクセス監視部１７を含む。

【００５４】

物理デバイス１０ｃは、イーサネットなどの外部ネットワークへデータを出力するＩ／Ｏデバイスで、例えばイーサネットカード，ファイバチャネルカードである。なお、本実施形態において、物理デバイス１０ｃは、上述した通りＣＰＵサブシステム１０Ａ内に備えられているが、ＣＰＵサブシステム１０Ａ外に備えられていてもよい。

【００５５】

ＤＭＡ制御部１１は、物理デバイス１０ｃとＣＰＵサブシステム１０Ａ内のメモリ１０ｂとの間で行なわれるＤＭＡを制御するもので、ＤＭＡwrite監視部１１ａおよびＤＭＡリマップ機能１１ｂを有している。

【００５６】

ＤＭＡwrite監視部（監視部）１１ａは、前述した手段(e4)としての機能を果たすもので、ＤＭＡwrite（つまり物理デバイス１０ｃからメモリ１０ｂへの書込み）を監視し、ＶＭメモリ管理テーブル２２にメモリ１０ｂの内容が変更されたことを記録する。より具体的に、ＤＭＡwrite監視部１１ａは、物理デバイス１０ｃからメモリ１０ｂへのデータ書込に係る書込アクセスを検出した場合、ＶＭメモリ管理テーブル（例えばIntel Extended Page Table）２２に、前記データ書込に係るアドレスに対応するデータに変更が生じたことを記録する。このとき、ＤＭＡwrite監視部１１ａは、ＶＭメモリ管理テーブル２

２において、前記アドレスに対応するDirtyビット（変更ビット；図２０参照）に“１”を設定することにより前記変更が生じたことを記録する。ハイパバイザ２０のＦＴ制御部２１は、ＶＭメモリ管理テーブル２２に従って、変更の生じたデータをプライマリからセカンダリに転送しコピーすることで同期処理を行ない、同期処理を終了すると、Dirtyビットを“０”にクリアする。

【００５７】

ＤＭＡリマップ機能１１ｂは、図３を参照しながら前述したＤＭＡリマップ機能１７０と同様（上述した処理手順(d1)～(d4)参照）、ＶＭ４０上のＯＳ内の物理デバイスドライバ４１が物理デバイス１０ｃを直接制御できるようにする機能である。

【００５８】

ＤＭＡreadリクエスト監視部（監視部）１２は、前述した手段(e1)および(e5)としての機能を果たす。ＤＭＡreadリクエスト監視部１２は、物理デバイス１０ｃが外部へのデータ出力を行なう際に、物理デバイス１０ｃからＤＭＡ制御部１１へ発行されるＤＭＡreadリクエスト（物理デバイス１０ｃからメモリ１０ｂへの読出リクエスト）を監視する。また、ＤＭＡreadリクエスト監視部１２は、保存領域１７ｂ（後述）に保存される、前回の同期処理時に参照されたカウンタ１７ａの計数値と、Dirtyカウンタ１７ａ（後述）の現在値とを比較する。カウンタ１７ａの現在値と保存領域１７ｂの計数値とが不一致であれば、ＤＭＡreadリクエスト監視部１２は、メモリ１０ｂのデータ変更が行なわれプライマリとセカンダリとの同期がとれていない状態であると判断する。そして、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３（後述）にＤＭＡreadリクエストを保存させることで外部へのデータ出力を抑止させる。さらに、ＤＭＡreadリクエスト監視部１２は、ハイパバイザ２０のＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６を介してハイパバイザ２０にＤＭＡreadリクエスト（読出アクセス）を検出したこと（ＤＭＡreadリクエスト検出割込み）を通知する。一方、カウンタ１７ａの現在値と保存領域１７ｂの計数値とが一致していれば、ＤＭＡreadリクエスト監視部１２は、プライマリとセカンダリとの同期がとれている状態であると判断する。そして、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３（後述）にＤＭＡreadリクエストを保存させることなく、外部へのデータ出力を実行させる。なお、ＤＭＡreadリクエスト監視部１２は、項目〔２－２－２〕で後述するごとく、物理デバイス１０ｃからメモリ１０ｂへデータを転送するＤＭＡwriteリクエストを監視し検出する機能も有している。

【００５９】

ＤＭＡreadリクエストバッファ部（抑止部，バッファ）１３は、前述した手段(e3)としての機能を果たすもので、物理デバイス１０ｃからＤＭＡ制御部１１へ発行されるＤＭＡreadリクエストを留めておくバッファである。つまり、ハイパバイザ２０がＤＭＡ割込み制御部１６から「ＤＭＡreadリクエスト検出割込み」の通知を受けるときに同期処理の実行中である場合、ＤＭＡreadリクエストバッファ部１３は、ＤＭＡreadリクエストを一時的に保持することにより外部へのデータ出力を抑止する。そして、ＤＭＡreadリクエストバッファ部１３は、ハイパバイザ２０（ＦＴ制御部２１）が同期処理を終了したことに伴うＤＭＡ割込み制御部１６（後述）からの指示により、保持していたＤＭＡreadリクエストをＤＭＡ制御部１１（メモリ１０ｂ側）へ送出する。

【００６０】

ＤＭＡreadデータ監視部（監視部）１４は、前述した手段(e1)および(e5)としての機能を果たす。ＤＭＡreadデータ監視部１４は、メモリコントローラ（メモリ）１０ｂから物理デバイス１０ｃへ読み出されるＤＭＡreadデータ（メモリ１０ｂから物理デバイス１０ｃへのデータ出力の対象データ）の読出を監視する。また、ＤＭＡreadデータ監視部１４は、ＤＭＡreadリクエスト監視部１２と同様、保存領域１７ｂ（後述）に保存される、前回の同期処理時に参照されたカウンタ１７ａの計数値と、Dirtyカウンタ１７ａ（後述）の現在値とを比較する。カウンタ１７ａの現在値と保存領域１７ｂの計数値とが不一致であれば、ＤＭＡreadデータ監視部１４は、メモリ１０ｂのデータ変更が行なわれプライマ

リとセカンダリとの同期がとれていない状態であると判断する。そして、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５（後述）にＤＭＡreadデータを保存させることで外部へのデータ出力を抑止させる。さらに、ＤＭＡreadデータ監視部１４は、ハイパバイザ２０のＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６を介してハイパバイザ２０にＤＭＡreadデータ（読出アクセス）を検出したこと（ＤＭＡreadデータ検出割込み）を通知する。一方、カウンタ１７ａの現在値と保存領域１７ｂの計数値とが一致していれば、ＤＭＡreadデータ監視部１４は、プライマリとセカンダリとの同期がとれている状態であると判断する。そして、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５（後述）にＤＭＡreadデータを保存させることなく、外部へのデータ出力を実行させる。

【００６１】

ＤＭＡreadデータバッファ部（抑止部，バッファ）１５は、前述した手段(e3)としての機能を果たすもので、メモリコントローラ（メモリ）１０ｂから物理デバイス１０ｃへ読み出されるＤＭＡreadデータを留めておくバッファである。つまり、ハイパバイザ２０がＤＭＡ割込み制御部１６から「ＤＭＡreadデータ検出割込み」の通知を受けるときに同期処理の実行中である場合、ＤＭＡreadデータバッファ部１５は、ＤＭＡreadデータを一時的に保持することにより外部へのデータ出力を抑止する。そして、ＤＭＡreadデータバッファ部１５は、ハイパバイザ２０（ＦＴ制御部２１）が同期処理を終了したことに伴うＤＭＡ割込み制御部１６（後述）からの指示により、保持していたＤＭＡreadデータを物理デバイス１０ｃへ送出する。

【００６２】

ＤＭＡ割込み制御部（通知部）１６は、前述した手段(e2)としての機能を果たす。ＤＭＡ割込み制御部１６は、ハイパバイザ２０へＤＭＡreadリクエストが検出されたことを通知するための割込みを発生させる。また、ＤＭＡ割込み制御部１６は、ハイパバイザ２０へＤＭＡreadデータが検出されたことを通知するための割込みを発生させる。つまり、ＤＭＡ割込み制御部１６は、監視部１２または１４がＤＭＡreadリクエストまたはＤＭＡreadデータを検出した場合、外部へのデータ出力を行なう読出アクセスを検出したものと判断し、読出アクセスを検出した旨を「ＤＭＡreadリクエスト検出割込み」または「ＤＭＡreadデータ検出割込み」としてハイパバイザ２０に通知する。

【００６３】

なお、ＤＭＡ割込み制御部１６は、割込み要因フラグ１６ａを有している。割込み要因フラグ１６ａは、監視部１２または１４がハイパバイザ２０（ＦＴ制御部２１）に同期処理を実行させるべくＤＭＡ割込み制御部１６に割込みを発生させた場合に設定される。一方、割込み要因フラグ１６ａは、ハイパバイザ２０（ＦＴ制御部２１）が同期処理を終了するとクリアされる。割込み要因フラグ１６ａがクリアされると、ＤＭＡ割込み制御部１６は、バッファ部１３，１５に対し、保持しているＤＭＡreadリクエストやＤＭＡreadデータを送出するように指示する。監視部１２および１４は、割込み要因フラグ１６ａが設定されている場合にハイパバイザ２０が同期処理の実行中であると判断する一方、割込み要因フラグ１６ａが未設定（クリア状態）の場合にハイパバイザ２０が同期処理の実行中ではないと判断する。

【００６４】

メモリアクセス監視部１７は、ＣＰＵ（ＣＰＵコア１０ａ）のメモリ１０ｂに対するアクセスを監視し、メモリ１０ｂの内容が変更されたときに、ＶＭメモリ管理テーブル２２にメモリ１０ｂの内容が変更されたことを記録する。このとき、メモリアクセス監視部１７は、上述したＤＭＡwrite監視部１１ａと同様、メモリ１０ｂの内容が変更されると、ＶＭメモリ管理テーブル２２において、前記内容変更に係るアドレスに対応するDirtyビット（変更ビット；図２０参照）に“１”を設定することにより変更が生じたことを記録する。ハイパバイザ２０のＦＴ制御部２１は、ＶＭメモリ管理テーブル２２に従って、変更の生じたデータをプライマリからセカンダリに転送しコピーすることで同期処理を行ない、同期処理を終了すると、Dirtyビットを“０”にクリアする。

【００６５】
なお、メモリアクセス監視部１７は、Dirtyカウンタ１７ａおよび保存領域１７ｂを有している。Dirtyカウンタ１７ａは、メモリ１０ｂにおいてデータの変更が発生した箇所の数を計数する。保存領域１７ｂは、前回の同期処理時に監視部１２または１４によって参照されたカウンタ１７ａの計数値を保存する。そして、ハイパバイザ２０がＤＭＡ割込み制御部１６からの通知を受けたときに同期処理の実行中でない場合、監視部１２または１４は、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの計数値とを比較する。Dirtyカウンタ１７ａの値と保存領域１７ｂの計数値とが不一致であれば、監視部１２または１４は、バッファ部１３または１５にデータ出力を抑止させ、ハイパバイザ２０に同期処理を実行させる。一方、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの計数値とが一致していれば、監視部１２または１４は、バッファ部１３または１５にＤＭＡreadリクエストまたはＤＭＡreadデータを保存させることなく、外部へのデータ出力を実行させる。
【００６６】
〔２－２〕第１実施形態のＦＴサーバの動作
次に、図４～図８を参照しながら、上述した第１実施形態のＦＴサーバ１の動作について説明する。
【００６７】
〔２－２－１〕外部へのデータ出力の把握動作
ここでは、まず、図４に示すＦＴサーバ１において、メモリ１０ｂから物理デバイス１０ｃへデータを転送する処理に際し、外部へのデータ出力を把握する動作（下記手順１～手順１５）について説明する。上記処理は、例えば、ネットワークパケットをネットワークカードである物理デバイス１０ｃから外部へ出力するために、メモリ１０ｂ上のネットワークパケットをネットワークカード１０ｃに転送する処理に該当する。
【００６８】
第１実施形態では、以下に説明する手順１～手順１５により、次の２点(f1)および(f2)が両立される。
(f1) ＶＭ４０内の物理デバイスドライバ４１が、物理デバイス１０ｃを直接操作すること。
(f2) ハイパバイザ２０のＦＴ制御部２１が、上述した“output commit”問題に対応するために外部へのデータ出力に係る読出アクセス（ＤＭＡreadリクエストの発行やＤＭＡreadデータの読出）を把握すること。
【００６９】
手順１： 物理デバイスドライバ４１は、物理デバイス１０ｃを通して外部へ送出するデータをネットワークパケットとしてメモリ１０ｂ上に用意し、物理デバイス１０ｃにデータの送出を指示する。
【００７０】
手順２： 物理デバイス１０ｃは、ＤＭＡ制御部１１に、メモリ１０ｂからデータを取得するためのＤＭＡreadリクエストを送信する。
【００７１】
手順３： 物理デバイス１０ｃとＤＭＡ制御部１１との間に備えられるＤＭＡreadリクエスト監視部１２は、物理デバイス１０ｃからのＤＭＡreadリクエストを検出する。ここで、ＦＴ制御部２１が同期処理を実行する必要がある場合、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１に同期制御開始を指示する必要がある。そのために、ＤＭＡreadリクエスト監視部１２は、まず、ＦＴ制御部２１が既に同期処理中か否かを確認する。具体的に、ＤＭＡreadリクエスト監視部１２は、ＤＭＡ割込み制御部１６において割込み要因フラグ１６ａがクリアされずに残っているか否かを確認する。割込み要因フラグ１６ａがクリアされずに残っているとき、つまり割込み要因フラグ１６ａが設定されているときには、ＦＴ制御部２１が同期処理中とみなすことができるので、“output commit”問題に対応すべくＤＭＡ処理を保留する必要がある。このため、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストをＤＭＡreadリクエストバッファ部１３に送出して保持さ

せる。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。一方、割込み要因フラグ１６ａがクリアされているときは、ＦＴ制御部２１が同期処理を実行していないとみなせる。その場合、ＤＭＡreadリクエスト監視部１２は、同期処理を実行する必要があるか否かを確認するために、Dirtyカウンタ１７ａを参照し、Dirtyカウンタ１７ａの現在値と、前回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する。Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが異なっている場合、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなすことができる。そこで、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６にＤＭＡreadリクエスト検出割込みをハイパバイザ２０に対して発生させるように要請する。このとき、ＤＭＡ割込み制御部１６の割込み要因フラグ１６ａが設定される。この後、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストをＤＭＡreadリクエストバッファ部１３に送出して保持させる。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。なお、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが同じ場合、プライマリとセカンダリとの同期がとれているとみなすことができる。そこで、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３にＤＭＡreadリクエストを保存させることなくＤＭＡ制御部１１へ送信し、外部へのデータ出力を実行させる。

【００７２】

手順４：　ＤＭＡ割込み制御部１６は、予めハイパバイザ２０のＦＴ制御部２１により設定された割込み送信先に、ＤＭＡreadリクエスト検出割込みを送信する。なお、ＤＭＡ割込み制御部１６は、例えば、図１５（Ａ）～図１５（Ｃ）を参照しながら後述する、ＦＴ制御部２１によって設定される監視対象ＰＣＩｅデバイス管理テーブル１６ｂを参照して、割込み送信先を判断する。

【００７３】

手順５：　ＤＭＡreadリクエスト検出割込みを受けたＣＰＵは、制御をＶＭ４０からハイパバイザ２０に移す。ハイパバイザ２０のＦＴ制御部２１は、割込み要因がＤＭＡreadリクエスト検出割込みであることが判明すると、同期処理を実行する。つまり、ＦＴ制御部２１は、ＶＭメモリ管理テーブル２２に記録されているメモリ変更記録を参照し、変更されたメモリ内容（前回同期処理を行なってから今回同期処理を行なうまでの間のメモリ差分）を全てプライマリからセカンダリへインタコネクト５０を通じて送信する。また、ＣＰＵのコンテキストもプライマリからセカンダリに送信する。

【００７４】

手順６：　セカンダリのハイパバイザ２０のＦＴ制御部２１は、プライマリからメモリ差分とＣＰＵコンテキスト情報とを受けとり、セカンダリのＶＭ４０のメモリおよびＣＰＵに反映する。その後、セカンダリのハイパバイザ２０のＦＴ制御部２１は、インタコネクト５０を通じてプライマリへ応答を返す。同期処理の高速化のため、プライマリへの応答は、メモリ差分とＣＰＵコンテキスト情報とを受け取った直後に行なってもよい。

【００７５】

手順７：　プライマリのハイパバイザ２０のＦＴ制御部２１は、セカンダリからの応答を受け取ると、同期処理が完了したとみなし、ＤＭＡ割込み制御部１６の割込み要因フラグ１６ａをクリアする。

【００７６】

手順８：　ＤＭＡread リクエストバッファ部１３は、外部へのデータ出力を抑止すべきときに、ＤＭＡreadリクエストを保存して留めておく。留めておいたＤＭＡreadリクエストをＤＭＡreadリクエストバッファ部１３からＤＭＡ制御部１１へ送出する契機は、ＦＴ制御部２１が同期処理を完了してＤＭＡ割込み制御部１６に割込み要因フラグ１６ａのクリアを指示したときである。このとき、ＤＭＡreadリクエストバッファ部１３は、ＦＴ制御部２１が同期処理を終了したことに伴うＤＭＡ割込み制御部１６からの指示により、ＤＭＡreadリクエストをＤＭＡ制御部１１へ送信する。

【００７７】

手順９：　ＤＭＡ制御部１１は、ＤＭＡリマップ機能１１ｂを用いて、ＤＭＡreadリクエスト内に含まれるアドレスをＶＭアドレス空間から物理アドレス空間に変換する。つまり、ＤＭＡ制御部１１は、予めハイパバイザ２０のＦＴ制御部２１が用意した、ＶＭメモリアドレスと物理メモリアドレスとの変換テーブル（例えばIntel VT-d DMA remapping table）を参照し、ＤＭＡreadリクエスト内に含まれるアドレスの変換を行なう。そして、ＤＭＡ制御部１１は、メモリコントローラ（メモリ）１０ｂへメモリアクセスリクエストを送信し、メモリコントローラ（メモリ）１０ｂからデータを受信する。

【００７８】

手順１０：　ＤＭＡ制御部１１は、手順９でメモリ１０ｂから受信したデータを、ＤＭＡreadデータとして物理デバイス１０ｃへ送信する。実際には、ＤＭＡreadデータは、ＤＭＡreadデータ監視部１４へ送出される。

【００７９】

手順１１：　ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータを１つ受ける毎にＤＭＡ割込み制御部１６に割込み要因フラグがクリアされずに残っているか否かを確認する。割込み要因フラグ１６ａがクリアされずに残っているとき、ＦＴ制御部２１が同期処理中であるので、“output commit”問題に対応すべくＤＭＡ処理を保留する必要がある。このため、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータをＤＭＡreadデータバッファ部１５に送出して保持させる。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。一方、割込み要因フラグ１６ａがクリアされているときは、ＦＴ制御部２１が同期処理を実行していないとみなせる。その場合、ＤＭＡreadデータ監視部１４は、同期処理を実行する必要があるか否かを確認するために、Dirtyカウンタ１７ａを参照し、Dirtyカウンタ１７ａの現在値と、前回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する。Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが異なっている場合、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなすことができる。そこで、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６にＤＭＡreadデータ検出割込みをハイパバイザ２０に対して発生させるように要請する。このとき、ＤＭＡ割込み制御部１６の割込み要因フラグ１６ａが設定される。この後、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータをＤＭＡreadデータバッファ部１５に送出して保持させる。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。なお、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが同じ場合、プライマリとセカンダリとの同期がとれているとみなすことができる。そこで、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５にＤＭＡreadデータを保存させることなく物理デバイス１０ｃへ送信し、外部へのデータ出力を実行させる。

【００８０】

手順１２：　ＤＭＡ割込み制御部１６は、予めハイパバイザ２０のＦＴ制御部２１により設定された割込み送信先に、ＤＭＡreadデータ検出割込みを送信する。このときも、手順４と同様、ＤＭＡ割込み制御部１６は、例えば、図１５（Ａ）～図１５（Ｃ）を参照しながら後述する、ＦＴ制御部２１によって設定される監視対象ＰＣＩｅデバイス管理テーブル１６ｂを参照して、割込み送信先を判断する。

【００８１】

手順１３：　ＤＭＡreadデータ検出割込みを受けたＣＰＵは、制御をＶＭ４０からハイパバイザ２０に移す。以後、ハイパバイザ２０のＦＴ制御部２１は、上述した手順５～手順７と同様の手順で同期処理を実行する。

【００８２】

手順１４：　ＤＭＡreadデータバッファ部１５は、外部へのデータ出力を抑止すべきときに、ＤＭＡreadデータを保存して留めておく。留めておいたＤＭＡreadデータをＤＭＡreadデータバッファ部１５から物理デバイス１０ｃへ送出する契機は、ＦＴ制御部２１が同期処理を完了してＤＭＡ割込み制御部１６に割込み要因フラグ１６ａのクリアを指示したときである。このとき、ＤＭＡreadデータバッファ部１５は、ＦＴ制御部２１が同期処

理を終了したことに伴うＤＭＡ割込み制御部１６からの指示により、ＤＭＡreadデータを物理デバイス１０ｃへ送信する。

【００８３】

手順１５：　物理デバイス１０ｃは、ＤＭＡreadデータを受け取って、ＤＭＡreadデータの転送を行なうことにより、ＤＭＡ転送を完了する。

以上の手順１～手順１５により、ＶＭ４０上の物理デバイスドライバ４１が物理デバイス１０ｃを直接操作しながら、外部へのデータ出力に係る読出アクセスが検出され、外部へのデータ出力が実行される前に必要な同期処理を行なうことができる。

【００８４】

〔２－２－２〕ＤＭＡwrite転送によるメモリ差分の把握動作

次に、図４に示すＦＴサーバ１において、物理デバイス１０ｃからメモリ１０ｂへデータを転送する処理に際し、ＤＭＡwrite転送によるメモリ差分を把握する動作（下記手順２１～手順２４）について説明する。上記処理は、例えば、外部からのネットワークパケットをネットワークカードである物理デバイス１０ｃが受信し、ネットワークカード１０ｃからメモリ１０ｂ上へネットワークパケットを転送する処理に該当する。

【００８５】

第１実施形態では、以下に説明する手順２１～手順２４により、ハイパバイザ２０のＦＴ制御部２１は、物理デバイス１０ｃからＶＭ４０のメモリにＤＭＡ転送されたデータを、メモリ差分として把握することができる。

【００８６】

既存の技術では、物理デバイス１０ｃからハイパバイザ２０がデータを受け取り、ハイパバイザ２０が、受け取ったデータをＶＭ４０のメモリに書き込んでいたので、ハイパバイザ２０は、前回の同期処理後に変更されたメモリ差分を把握することができていた。しかし、本実施形態では、ハイパバイザ２０を介さずに物理デバイス１０ｃからＶＭ４０のメモリにデータが直接書き込まれるため、以下の手順２１～手順２４により書き込まれたデータの位置（アドレス）および量を、後からハイパバイザ２０が把握できるようにする必要がある。

【００８７】

手順２１：　物理デバイス１０ｃは、ＶＭ４０のメモリアドレスに対してＤＭＡwriteリクエストを送信する。

【００８８】

手順２２：　ＤＭＡreadリクエスト監視部１２は、ＤＭＡwriteリクエストをＤＭＡ制御部１１に送信する。

【００８９】

手順２３：　ＤＭＡ制御部１１のＤＭＡリマップ機能１１ｂは、予めハイパバイザ２０のＦＴ制御部２１が用意した、ＶＭメモリアドレスと物理メモリアドレスとの変換テーブルを参照し、ＤＭＡwriteリクエスト内に含まれるアドレスの変換を行なう。つまり、ＤＭＡリマップ機能１１ｂは、ＤＭＡwriteリクエスト内の物理メモリアドレスをＶＭメモリアドレスに変換する。

【００９０】

手順２４：　ＤＭＡwrite監視部１１ａは、ハイパバイザ２０のＶＭメモリ管理テーブル２２を参照し、当該テーブル２２において、ＤＭＡwriteリクエスト内のアドレスに対応するDirtyビット（変更ビット；図２０参照）に“１”を設定する。これにより、ＤＭＡwriteリクエスト内のアドレスに対応するデータに変更が生じたことが、ＶＭメモリ管理テーブル２２に記録される。その後、ＤＭＡwrite監視部１１ａは、メモリコントローラ（メモリ）１０ｂへメモリアクセスリクエストを送信する。

【００９１】

以上の手順２１～手順２４により、ＶＭメモリ管理テーブル２２にメモリの変更があったことが記録されるので、ハイパバイザ２０のＦＴ制御部２１は、同期処理時に、プライマリからセカンダリに転送すべきメモリ変更差分を確実に把握することができる。

【００９２】
〔２－２－３〕ＤＭＡreadリクエスト監視部の動作
次に、図５に示すフローチャート（ステップＳ１０１～Ｓ１０９）に従って、図４に示すＦＴサーバ１におけるＤＭＡreadリクエスト監視部１２の処理（動作）について説明する。ここで説明するＤＭＡreadリクエスト監視部１２の処理（動作）は、上述した手順３で実行される。
【００９３】
ＤＭＡreadリクエスト監視部１２は、物理デバイス１０ｃからのＤＭＡパケットを待機し（ステップＳ１０１）、ＤＭＡパケットが来ない場合（ステップＳ１０２のＮＯルート）、ステップＳ１０１の処理に戻る。ＤＭＡパケットが来た場合（ステップＳ１０２のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、受信したＤＭＡパケットがＤＭＡreadリクエストであるか否かを判断する（ステップＳ１０３）。
【００９４】
ＤＭＡパケットがＤＭＡreadリクエストでない場合（ステップＳ１０３のＮＯルート）、ＤＭＡreadリクエスト監視部１２は、受信したＤＭＡパケットをＤＭＡ制御部１１に出力し（ステップＳ１０９）、処理を終了する。
【００９５】
一方、ＤＭＡパケットがＤＭＡreadリクエストである場合（ステップＳ１０３のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、ＤＭＡ割込み制御部１６に、ＤＭＡreadリクエストを検出したことによる割込み要因フラグ１６ａがセットされているか否かを判断する（ステップＳ１０４）。割込み要因フラグ１６ａがセットされている場合（ステップＳ１０４のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理中であるとみなし、受信したＤＭＡパケットつまりＤＭＡreadリクエストをＤＭＡreadリクエストバッファ部１３に送出して保持させ（ステップＳ１０５）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。
【００９６】
また、割込み要因フラグ１６ａがセットされていない場合（ステップＳ１０４のＮＯルート）、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理を実行していないとみなす。そして、ＤＭＡreadリクエスト監視部１２は、同期処理を実行する必要があるか否かを確認するために、Dirtyカウンタ１７ａを参照し、Dirtyカウンタ１７ａの現在値と、前回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する（ステップＳ１０６）。
【００９７】
Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致しない場合（ステップＳ１０７のＮＯルート）、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなす。そして、ＤＭＡreadリクエスト監視部１２は、どのデバイス（物理デバイス１０ｃ）からのＤＭＡreadリクエストを検出したのかをＤＭＡ割込み制御部１６に通知し、割込み要因フラグ１６ａをセットする（ステップＳ１０８）。これにより、ＤＭＡ割込み制御部１６からハイパバイザ２０に対してＤＭＡreadリクエスト検出割込みが発生され、ＦＴ制御部２１が同期処理を実行する。なお、ＤＭＡreadリクエスト監視部１２は、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値との比較の終了後、保存領域１７ｂの値をDirtyカウンタ１７ａの現在値に置き換えるようメモリアクセス監視部１７に対して指示する。
【００９８】
この後、ＤＭＡreadリクエスト監視部１２は、受信したＤＭＡパケットつまりＤＭＡreadリクエストをＤＭＡreadリクエストバッファ部１３に送出して保持させ（ステップＳ１０５）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。
【００９９】
Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致する場合（ステップＳ１

０７のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、プライマリとセカンダリとの同期がとれているとみなす。そして、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３にＤＭＡreadリクエストを保存させることなくＤＭＡ制御部１１へ送出し（ステップＳ１０９）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が実行される。

【０１００】

〔２－２－４〕ＤＭＡreadデータ監視部の動作

次に、図６に示すフローチャート（ステップＳ１１１～Ｓ１１９）に従って、図４に示すＦＴサーバ１におけるＤＭＡreadデータ監視部１４の処理（動作）について説明する。ここで説明するＤＭＡreadデータ監視部１４の処理（動作）は、上述した手順１１で実行される。

【０１０１】

ＤＭＡreadデータ監視部１４は、ＤＭＡ制御部１１からのＤＭＡパケットを待機し（ステップＳ１１１）、ＤＭＡパケットが来ない場合（ステップＳ１１２のＮＯルート）、ステップＳ１１１の処理に戻る。ＤＭＡパケットが来た場合（ステップＳ１１２のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、受信したＤＭＡパケットがＤＭＡreadデータであるか否かを判断する（ステップＳ１１３）。

【０１０２】

ＤＭＡパケットがＤＭＡreadデータでない場合（ステップＳ１１３のＮＯルート）、ＤＭＡreadデータ監視部１４は、受信したＤＭＡパケットを物理デバイス１０ｃに出力し（ステップＳ１１９）、処理を終了する。

【０１０３】

一方、ＤＭＡパケットがＤＭＡreadデータである場合（ステップＳ１１３のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、ＤＭＡ割込み制御部１６に、ＤＭＡreadデータを検出したことによる割込み要因フラグ１６ａがセットされているか否かを判断する（ステップＳ１１４）。割込み要因フラグ１６ａがセットされている場合（ステップＳ１１４のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１が同期処理中であるとみなし、受信したＤＭＡパケットつまりＤＭＡreadデータをＤＭＡreadデータバッファ部１５に送出して保持させ（ステップＳ１１５）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。

【０１０４】

また、割込み要因フラグ１６ａがセットされていない場合（ステップＳ１１４のＮＯルート）、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１が同期処理を実行していないとみなす。そして、ＤＭＡreadデータ監視部１４は、同期処理を実行する必要があるか否かを確認するために、Dirtyカウンタ１７ａを参照し、Dirtyカウンタ１７ａの現在値と、前回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する（ステップＳ１１６）。

【０１０５】

Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致しない場合（ステップＳ１１７のＮＯルート）、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなす。そして、ＤＭＡreadデータ監視部１４は、どのデバイス（物理デバイス１０ｃ）へのＤＭＡreadデータを検出したのかをＤＭＡ割込み制御部１６に通知し、割込み要因フラグ１６ａをセットする（ステップＳ１１８）。これにより、ＤＭＡ割込み制御部１６からハイパバイザ２０に対してＤＭＡreadデータ検出割込みが発生され、ＦＴ制御部２１が同期処理を実行する。なお、ＤＭＡreadデータ監視部１４は、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値との比較の終了後、保存領域１７ｂの値をDirtyカウンタ１７ａの現在値に置き換えるようメモリアクセス監視部１７に対して指示する。

【０１０６】

この後、ＤＭＡreadデータ監視部１４は、受信したＤＭＡパケットつまりＤＭＡreadデ

ータをＤＭＡreadデータバッファ部１５に送出して保持させ（ステップＳ１１５）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が抑止される。

【０１０７】

Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致する場合（ステップＳ１１７のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、プライマリとセカンダリとの同期がとれているとみなす。そして、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５にＤＭＡreadデータを保存させることなく物理デバイス１０ｃへ送出し（ステップＳ１１９）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が実行される。

【０１０８】

〔２－２－５〕ＤＭＡ割込み制御部の動作

次に、図７および図８に示すフローチャート（ステップＳ１２１～Ｓ１３２）に従って、図４に示すＦＴサーバ１におけるＤＭＡ割込み制御部１６の処理（動作）について説明する。ここで説明するＤＭＡ割込み制御部１６の処理（動作）は、上述した手順３，手順４，手順７，手順８，手順１１～手順１４に係る。

【０１０９】

ＤＭＡ割込み制御部１６は、ＤＭＡreadリクエスト監視部１２，ＤＭＡreadデータ監視部１４，ＦＴ制御部２１のいずれかからの指示を待機し（ステップＳ１２１）、指示が来ない場合（ステップＳ１２２のＮＯルート）、ステップＳ１２１の処理に戻る。指示が来た場合（ステップＳ１２２のＹＥＳルート）、ＤＭＡ割込み制御部１６は、受信した指示がＤＭＡreadリクエスト監視部１２からの指示であるか否かを判断する（ステップＳ１２３）。

【０１１０】

受信した指示がＤＭＡreadリクエスト監視部１２からの指示である場合（ステップＳ１２３のＹＥＳルート）、ＤＭＡ割込み制御部１６は、予めＦＴ制御部２１により設定された割込み送信先に、ＤＭＡreadリクエスト検出割込みを送信し（ステップＳ１２４）、ステップＳ１２１の処理に戻る。

【０１１１】

受信した指示がＤＭＡreadリクエスト監視部１２からの指示でない場合（ステップＳ１２３のＮＯルート）、ＤＭＡ割込み制御部１６は、受信した指示がＤＭＡreadデータ監視部１４からの指示であるか否かを判断する（ステップＳ１２５）。

【０１１２】

受信した指示がＤＭＡreadデータ監視部１４からの指示である場合（ステップＳ１２５のＹＥＳルート）、ＤＭＡ割込み制御部１６は、予めＦＴ制御部２１により設定された割込み送信先に、ＤＭＡreadデータ検出割込みを送信し（ステップＳ１２６）、ステップＳ１２１の処理に戻る。

【０１１３】

受信した指示がＤＭＡreadデータ監視部１４からの指示でない場合（ステップＳ１２５のＮＯルート）、ＤＭＡ割込み制御部１６は、受信した指示がＦＴ制御部２１からの指示であるか否かを判断する（ステップＳ１２７）。

【０１１４】

受信した指示がＦＴ制御部２１からの指示でない場合（ステップＳ１２７のＮＯルート）、ＤＭＡ割込み制御部１６は、ステップＳ１２１の処理に戻る。一方、受信した指示がＦＴ制御部２１からの指示である場合（ステップＳ１２７のＹＥＳルート）、ＤＭＡ割込み制御部１６は、受信した指示がＤＭＡreadリクエスト検出割込みフラグ１６ａのクリア指示（上記手順７，手順８参照）であるか否かを判断する（ステップＳ１２８）。

【０１１５】

受信した指示がＤＭＡreadリクエスト検出割込みフラグ１６ａのクリア指示である場合（ステップＳ１２８のＹＥＳルート）、ＤＭＡ割込み制御部１６は、ＤＭＡreadリクエストバッファ部１３に対し、パケット（ＤＭＡreadリクエスト）をＤＭＡ制御部１１へ送信

するよう指示する（ステップＳ１２９；上記手順８参照）。この後、ＤＭＡ割込み制御部１６は、ステップＳ１２１の処理に戻る。

【０１１６】

受信した指示がＤＭＡreadリクエスト検出割込みフラグ１６ａのクリア指示でない場合（ステップＳ１２８のＮＯルート）、ＤＭＡ割込み制御部１６は、受信した指示がＤＭＡreadデータ検出割込みフラグ１６ａのクリア指示（上記手順１４参照）であるか否かを判断する（ステップＳ１３０）。

【０１１７】

受信した指示がＤＭＡreadデータ検出割込みフラグ１６ａのクリア指示である場合（ステップＳ１３０のＹＥＳルート）、ＤＭＡ割込み制御部１６は、ＤＭＡreadデータバッファ部１５に対し、パケット（ＤＭＡreadデータ）を物理デバイス１０ｃへ送信するよう指示する（ステップＳ１３１；上記手順１４参照）。この後、ＤＭＡ割込み制御部１６は、ステップＳ１２１の処理に戻る。

【０１１８】

受信した指示がＤＭＡreadデータ検出割込みフラグ１６ａのクリア指示でない場合（ステップＳ１３０のＮＯルート）、受信した指示は、ＦＴ制御部２１からのその他の指示であるので、ＤＭＡ割込み制御部１６は、当該指示に応じた適切な処理を実行する（ステップＳ１３２）。この後、ＤＭＡ割込み制御部１６は、ステップＳ１２１の処理に戻る。

【０１１９】

〔２－３〕第１実施形態のＦＴサーバの効果

このように、第１実施形態のＦＴサーバ１によれば、以下のような作用効果(g1)～(g6)を得ることができる。

【０１２０】

(g1) ハイパバイザ２０が物理デバイス１０ｃを仮想デバイスとしてエミュレートすることなく外部へのデータの出力を検出することが可能になる。したがって、ハイパバイザ２０は物理デバイス１０ｃを仮想デバイスとしてエミュレートする必要がなくなり、ハイパバイザ２０が提供するＶＭ４０内の物理デバイスドライバ４１が物理デバイス１０ｃを直接操作できるので、外部へのデータ出力のパフォーマンスが改善される。これにより、ＶＭの入出力アクセス性能が改善される。

【０１２１】

(g2) その際、ハイパバイザ２０が外部へのデータ出力を検出したことに伴って同期処理実行中である場合には、外部へのデータ出力が抑止される。したがって、同期処理中に外部へデータを出力してしまいプライマリとセカンダリとの同期状態が壊れてしまうことが確実に防止される。

【０１２２】

(g3) 物理デバイス１０ｃからハイパバイザ２０が提供するＶＭ４０のメモリへのＤＭＡを監視してＶＭメモリ管理テーブル２２に記録する機能が備えられているので、同期処理を実行するために必要なメモリ１０ｂの情報をハイパバイザ２０が確実に把握することができる。

【０１２３】

(g4) ＦＴサーバ１で使用する物理デバイス１０ｃに対応するエミュレーション機能を作り込むことなく、広く採用されていないデバイスや最新技術のデバイスであっても、ＶＭ４０内のＯＳで使用することが可能になる。したがって、ハイパバイザ２０を利用するＦＴサーバ１において、通常のサーバで採用される物理デバイスの種類と同じ種類の物理デバイスを、容易に扱うことができる。

【０１２４】

(g5) ＦＴサーバ１は、ＣＰＵサブシステム１０Ａ内に物理デバイス１０ｃが備えられた構成にも適用できる。つまり、ＦＴサーバ１は、ＳｏＣ化されたＣＰＵを採用した構成にも適用され、高性能なＦＴサーバ１を実現することができる。

【０１２５】

(g6) 本実施形態では、同期処理が必要ないときには外部へのデータ出力をハイパバイザに通知しない手段が備えられている。つまり、本実施形態では、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致する場合にはプライマリとセカンダリとの同期がとれているとみなして、ＦＴ制御部２１による同期処理が回避されている。したがって、不要な同期処理が抑制され、ＦＴサーバ１の全体の性能が向上する。

【０１２６】

〔３〕第２実施形態のＦＴサーバ

以下、図９～図２０を参照しながら第２実施形態のＦＴサーバ（情報処理装置）１′について説明する。

【０１２７】

〔３－１〕第２実施形態のＦＴサーバの構成

まず、図９を参照しながら第２実施形態のＦＴサーバ１′の構成について説明する。図９は、第２実施形態のＦＴサーバ１′のハードウェア構成および機能構成を示すブロック図である。なお、既述の符号と同一の符号は、同一またはほぼ同一の部分を示している。

【０１２８】

図９に示す第２実施形態のＦＴサーバ１′においても、図４に示す第１実施形態のＦＴサーバ１と同様、同じ構成の二組のハードウェア１０′，１０′が、イーサネットなどのインタコネクト５０を介して接続されている。第２実施形態でも、一方のハードウェア１０′をプライマリと呼び、他方のハードウェア１０′をセカンダリと呼ぶ。プライマリおよびセカンダリのそれぞれにおいてはハイパバイザ２０が構築され、ハイパバイザ２０によって管理ＶＭ３０とＶＭ４０とが構築され管理される。なお、以下では、プライマリ側の機能，構成および動作について説明するが、セカンダリ側の機能，構成および動作はプライマリ側と同様であるので、その説明や図示は省略する。

【０１２９】

第２実施形態でも、ＶＭ４０が、保護対象つまりフォールトトレラント対象であり、ＶＭ４０上のＯＳ内には、物理デバイス（Ｉ／Ｏデバイス）としてのＰＣＩｅデバイス１０ｅを制御するドライバ４２が備えられ、ＶＭ４０からＰＣＩｅデバイス１０ｅを直接制御できるようになっている。また、ハイパバイザ２０には、第１実施形態と同様、ＶＭ４０を対象とするＦＴの制御を行なうＦＴ制御部２１と、ＶＭメモリ空間（メモリ１０ｂ）を管理するＶＭメモリ管理テーブル２２とが備えられている。

【０１３０】

ハードウェア１０′には、ＣＰＵとメモリからなるＣＰＵサブシステム１０Ｂが備えられている。ＣＰＵサブシステム１０Ｂは、ＣＰＵコア１０ａと、メモリおよびメモリコントローラ１０ｂと、ＰＣＩｅルートコンプレクス１０ｄと、物理デバイスとしての二つのＰＣＩｅデバイス１０ｅ，１０ｆと、メモリアクセス監視部１７とを、構成要素として含んでいる。また、第２実施形態のＦＴサーバ１′も、ＳｏＣ化されたＣＰＵを採用した構成に適用されている。さらに、ＰＣＩｅルートコンプレクス１０ｄは、ＤＭＡ制御部１１，ＤＭＡreadリクエスト監視部１２，ＤＭＡreadリクエストバッファ部１３，ＤＭＡreadデータ監視部１４，ＤＭＡreadデータバッファ部１５およびＤＭＡ割込み制御部１６を、構成要素として含む。

【０１３１】

以下、各構成要素について説明する。なお、各構成要素の数については特に規定していないが、それぞれ１以上備えられている。

ＰＣＩｅルートコンプレクス１０ｄは、ＰＣＩｅ規格に規定されているコンポーネントであり、ＣＰＵコア１０ａおよびメモリコントローラ１０ｂとＰＣＩｅデバイス１０ｅおよび１０ｆとを接続するコンポーネント（インタフェース）である。

【０１３２】

ＤＭＡ制御部１１は、ＰＣＩｅデバイス１０ｅとＣＰＵサブシステム１０Ｂ内のメモリ１０ｂとの間で行なわれるＤＭＡを制御するもので、ＤＭＡwrite監視部１１ａおよびＤＭＡリマップ機能１１ｂを有している。より具体的に、ＤＭＡ制御部１１は、ＰＣＩｅデ

バイス１０ｅ側から到着するＰＣＩｅトランザクションパケットを、メモリコントローラ１０ｂへのメモリアクセス命令に変換する処理と、メモリコントローラ側から到着するメモリアクセス命令をＰＣＩｅデバイス１０ｅへのＰＣＩｅトランザクションパケットに変換処理とを行なう。このとき、ＤＭＡリマップ機能１１ｂによって、ＰＣＩｅデバイス１０ｅ側の指定するアドレスがメモリコントローラ１０ｂ側のアドレスに変換される。このようなＤＭＡリマップ機能１１ｂは、例えば、Intel社のXeon E5シリーズのＣＰＵなどに搭載されている。なお、ＤＭＡリマップ機能１１ｂは、図３を参照しながら前述したＤＭＡリマップ機能１７０と同様（上述した処理手順(d1)～(d4)参照）、ＶＭ４０上のＯＳ内の物理デバイスドライバ４１が物理デバイス１０ｃを直接制御できるようにする機能である。

【０１３３】

ＤＭＡwrite監視部（監視部）１１ａは、前述した手段(e4)としての機能を果たすもので、Memory Write Requestトランザクション（ＭＷｒ；ＰＣＩｅデバイス１０ｅからメモリ１０ｂへの書込み）を監視し、ＶＭメモリ管理テーブル２２にメモリ１０ｂの内容が変更されたことを記録する。より具体的に、ＤＭＡwrite監視部１１ａは、ＰＣＩｅデバイス１０ｅからメモリ１０ｂへのデータ書込に係る書込アクセスであるMemory Write Requestトランザクション（ＭＷｒ）を検出した場合、ＶＭメモリ管理テーブル２２に、前記データ書込に係るアドレスに対応するデータに変更が生じたことを記録する。なお、第２実施形態のＤＭＡwrite監視部１１ａも、第１実施形態のＤＭＡwrite監視部１１ａと同様の機能を果たすので、その詳細な説明は省略する。

【０１３４】

ＤＭＡreadリクエスト監視部（監視部）１２は、前述した手段(e1)および(e5)としての機能を果たす。ＤＭＡreadリクエスト監視部１２は、ＰＣＩｅデバイス１０ｅからＰＣＩｅルートコンプレクス１０ｄへ送信されるＰＣＩｅトランザクションパケットを監視する。ＤＭＡreadリクエスト監視部１２は、Memory Read Requestトランザクション（ＭＲｄ；ＰＣＩｅデバイス１０ｅからＰＣＩｅルートコンプレクス１０ｄへの読出リクエスト）を検出すると、保存領域１７ｂに保存される、前回の同期処理時に参照されたカウンタ１７ａの計数値と、Dirtyカウンタ１７ａの現在値とを比較する。カウンタ１７ａの現在値と保存領域１７ｂの計数値とが不一致であれば、ＤＭＡreadリクエスト監視部１２は、メモリ１０ｂのデータ変更が行なわれプライマリとセカンダリとの同期がとれていない状態であると判断する。そして、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３にMemory Read Requestトランザクション（ＭＲｄ）を保存させることで外部へのデータ出力を抑止させる。さらに、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６を介してハイパバイザ２０にMemory Read Requestトランザクション（ＭＲｄ）を検出したことをMemory Read Request割込みとして通知する。一方、カウンタ１７ａの現在値と保存領域１７ｂの計数値とが一致していれば、ＤＭＡreadリクエスト監視部１２は、プライマリとセカンダリとの同期がとれている状態であると判断する。そして、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３にMemory Read Requestトランザクション（ＭＲｄ）を保存させることなく、外部へのデータ出力を実行させる。なお、ＤＭＡreadリクエスト監視部１２は、後述するごとく、ＰＣＩｅデバイス１０ｅからメモリ１０ｂへデータを転送するMemory Write Requestパケットを監視し検出する機能も有している。

【０１３５】

ＤＭＡreadリクエストバッファ部（抑止部，バッファ）１３は、前述した手段(e3)としての機能を果たすもので、ＰＣＩｅデバイス１０ｅからＰＣＩｅルートコンプレクス１０ｄへ送信されるMemory Read Requestトランザクション（ＭＲｄ）を留めておくバッファである。つまり、ハイパバイザ２０がＤＭＡ割込み制御部１６からMemory Read Request割込みの通知を受けるときに同期処理の実行中である場合、ＤＭＡreadリクエストバッファ部１３は、Memory Read Requestトランザクション（ＭＲｄ）を一時的に保持することにより外部へのデータ出力を抑止する。そして、ＤＭＡreadリクエストバッファ部１３は、Ｆ

Ｔ制御部２１が同期処理を終了したことに伴うＤＭＡ割込み制御部１６からの指示により、保持していたMemory Read Requestトランザクション（ＭＲd）をＤＭＡ制御部１１（メモリ１０ｂ側）へ送出する。

【０１３６】

ＤＭＡreadデータ監視部（監視部）１４は、前述した手段(e1)および(e5)としての機能を果たす。ＤＭＡreadデータ監視部１４は、メモリコントローラ（メモリ）１０ｂからＰＣＩｅデバイス１０ｅへ送信されるＰＣＩｅトランザクションパケットを監視する。ＤＭＡreadデータ監視部１４は、Completion with Dataトランザクション（ＣｐｌＤ；メモリ１０ｂからＰＣＩｅデバイス１０ｅへのデータ出力の対象データの読出）を検出すると、前回の同期処理時に参照されたカウンタ１７ａの計数値と、Dirtyカウンタ１７ａの現在値とを比較する。カウンタ１７ａの現在値と保存領域１７ｂの計数値とが不一致であれば、ＤＭＡreadデータ監視部１４は、メモリ１０ｂのデータ変更が行なわれプライマリとセカンダリとの同期がとれていない状態であると判断する。そして、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５にCompletion with Dataトランザクション（ＣｐｌＤ）を保存させることで外部へのデータ出力を抑止させる。さらに、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６を介してハイパバイザ２０にCompletion with Dataトランザクション（ＣｐｌＤ）を検出したことをCompletion with Data割込みとして通知する。一方、カウンタ１７ａの現在値と保存領域１７ｂの計数値とが一致していれば、ＤＭＡreadデータ監視部１４は、プライマリとセカンダリとの同期がとれている状態であると判断する。そして、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５にCompletion with Dataトランザクション（ＣｐｌＤ）を保存させることなく、外部へのデータ出力を実行させる。

【０１３７】

ＤＭＡreadデータバッファ部（抑止部，バッファ）１５は、前述した手段(e3)としての機能を果たすもので、メモリコントローラ（メモリ）１０ｂからＰＣＩｅデバイス１０ｅへ送信されるCompletion with Dataトランザクション（ＣｐｌＤ）を留めておくバッファである。つまり、ハイパバイザ２０がＤＭＡ割込み制御部１６からCompletion with Data割込みの通知を受けるときに同期処理の実行中である場合、ＤＭＡreadデータバッファ部１５は、Completion with Dataトランザクション（ＣｐｌＤ）を一時的に保持することにより外部へのデータ出力を抑止する。そして、ＤＭＡreadデータバッファ部１５は、ＦＴ制御部２１が同期処理を終了したことに伴うＤＭＡ割込み制御部１６からの指示により、保持していたCompletion with Dataトランザクション（ＣｐｌＤ）をＰＣＩｅデバイス１０ｅへ送出する。

【０１３８】

ＤＭＡ割込み制御部（通知部）１６は、前述した手段(e2)としての機能を果たす。ＤＭＡ割込み制御部１６は、ハイパバイザ２０へMemory Read Requestトランザクション（ＭＲd）が検出されたことを通知するための割込み（Intel ＣＰＵであればＶMexit）を発生させる。また、ＤＭＡ割込み制御部１６は、ハイパバイザ２０へCompletion with Dataトランザクション（ＣｐｌＤ）が検出されたことを通知するための割込み（Intel ＣＰＵであればＶMexit）を発生させる。つまり、ＤＭＡ割込み制御部１６は、監視部１２または１４がMemory Read Requestトランザクション（ＭＲd）またはCompletion with Dataトランザクション（ＣｐｌＤ）を検出した場合、外部へのデータ出力を行なう読出アクセスを検出したものと判断し、読出アクセスを検出した旨を「Memory Read Request割込み」または「Completion with Data割込み」としてハイパバイザ２０に通知する。

【０１３９】

なお、ＤＭＡ割込み制御部１６は、割込み要因フラグ１６ａおよび監視対象ＰＣＩｅデバイス管理テーブル１６ｂを有している。割込み要因フラグ１６ａは、監視部１２または１４がＦＴ制御部２１に同期処理を実行させるべくＤＭＡ割込み制御部１６に割込みを発生させた場合に設定される。一方、割込み要因フラグ１６ａは、ＦＴ制御部２１が同期処理を終了するとクリアされる。割込み要因フラグ１６ａがクリアされると、ＤＭＡ割込み

制御部１６は、バッファ部１３，１５に対し、保持しているMemory Read Requestトランザクション（ＭＲｄ）やCompletion with Dataトランザクション（ＣｐｌＤ）を送出するように指示する。監視部１２および１４は、割込み要因フラグ１６ａが設定されている場合にＦＴ制御部２１が同期処理の実行中であると判断する一方、割込み要因フラグ１６ａが未設定（クリア状態）の場合にＦＴ制御部２１が同期処理の実行中ではないと判断する。なお、監視対象ＰＣＩｅデバイス管理テーブル１６ｂについては、図１５（Ａ）～図１５（Ｃ）を参照しながら後述する。

【０１４０】

メモリアクセス監視部１７は、第１実施形態のメモリアクセス監視部１７と同様、ＣＰＵ（ＣＰＵコア１０ａ）のメモリ１０ｂに対するアクセスを監視し、メモリ１０ｂの内容が変更されたときに、ＶＭメモリ管理テーブル２２にメモリ１０ｂの内容が変更されたことを記録する。また、第２実施形態のメモリアクセス監視部１７もDirtyカウンタ１７ａおよび保存領域１７ｂを有している。なお、第２実施形態のメモリアクセス監視部１７，Dirtyカウンタ１７ａおよび保存領域１７ｂも、第１実施形態のメモリアクセス監視部１７，Dirtyカウンタ１７ａおよび保存領域１７ｂと同様の機能を果たすので、その詳細な説明は省略する。

【０１４１】

ＰＣＩｅデバイス１０ｅは、イーサネットやファイバチャネルなど、システムの外部との間でデータをやり取りする機能を有する物理デバイスである。ＰＣＩｅデバイス１０ｅは、ＣＰＵサブシステムに内蔵されている場合もあるし、ＣＰＵサブシステム外に存在しＣＰＵサブシステムとはＰＣＩｅリンクによって接続されている場合もある。第２実施形態では、図９に示すように、ＰＣＩｅデバイス１０ｅは、ＣＰＵサブシステム１０Ｂに内蔵されている場合について説明している。ＰＣＩｅデバイス１０ｅは、ＤＭＡリマップ機能１１ｂによりＶＭ４０上のＰＣＩｅデバイス１０ｅのドライバ４２からアクセス可能に構成されている。

【０１４２】

ＰＣＩｅデバイス１０ｆは、プライマリとセカンダリとが通信するための機能を有する物理デバイスで、イーサネットやインフィニバンドなどのインタコネクト５０に接続される。ＰＣＩｅデバイス１０ｆは、ＣＰＵサブシステムに内蔵されている場合もあるし、ＣＰＵサブシステム外に存在しＣＰＵサブシステムとはＰＣＩｅリンクによって接続されている場合もある。第２実施形態では、図９に示すように、ＰＣＩｅデバイス１０ｆは、ＣＰＵサブシステム１０Ｂに内蔵されている場合について説明している。

【０１４３】

ハイパバイザ２０は、管理ＶＭ３０およびＶＭ４０を構築し管理するもので、前述の通り、ＦＴ制御部２１およびＶＭメモリ管理テーブル２２を有している。

ＶＭメモリ管理テーブル２２は、ＶＭ４０に割り当てるメモリを管理するためのテーブルであり、ＶＭメモリ管理テーブル２２としては、例えばIntel社のExtended Page Tableが用いられる。また、ＶＭメモリ管理テーブル２２は、ＶＭ４０の物理メモリアドレスとシステムの物理アドレスとの対応情報を保持する。さらに、ＶＭメモリ管理テーブル２２は、ＣＰＵ（ＣＰＵコア１０ａ）から書込アクセスがあったメモリの情報を保持する。なお、ＶＭメモリ管理テーブル２２の詳細な構成については、図２０を参照しながら後述する。

【０１４４】

ＦＴ制御部２１は、第１実施形態のＦＴ制御部２１と同様、フォールトトレラント機能を実現するために、以下の「同期処理」を実行する。ＦＴ制御部２１は、ＤＭＡ割込み制御部１６からの割込み（例えばＶＭexit）を契機として、ＶＭ４０のＣＰＵコンテキストとＶＭのメモリ内容とをセカンダリにＰＣＩｅデバイス１０ｆを利用して送信する。当該契機となる割込み要因は、ＰＣＩｅデバイス１０ｅからＰＣＩｅルートコンプレクス１０ｄへのMemory Read Requestトランザクション（ＭＲｄ）の検出、または、ＰＣＩｅルートコンプレクス１０ｄからＰＣＩｅデバイス１０ｅへのCompletion with Dataトランザク

ション（ＣｐｌＤ）の検出である。ＶＭ４０のＣＰＵコンテキストは、例えばＶＭexitにおいてはメモリ１０ｂ上にＣＰＵによって退避される。また、プライマリからセカンダリへ送信されるＶＭ４０のメモリ内容は、前回ＶＭ４０のメモリ変更部分を送付してから今回送付するまでの間に、ＶＭ４０のメモリ内容について変更された部分だけとする。これにより、同期処理に伴う送信時間を短縮することができる。上述のごとくＶＭ４０のメモリ内容について変更された部分だけを送信可能にすべく、ＦＴ制御部２１は、ＶＭメモリ管理テーブル２２を検索し、ＣＰＵから書込アクセスがあったメモリ変更部分（Dirtyビットに“１”が設定されているアドレスに対応するデータ）をセカンダリに送信する。ＦＴ制御部２１は、メモリ変更部分をセカンダリに送信すると、ＶＭメモリ管理テーブル２２においてメモリ変更部分に対応するアドレスに設定されているDirtyビットを“０”にクリアする。このようなDirtyビットを用いることで、次回のセカンダリへのメモリ変更部分の送信時に、ＦＴ制御部２１は、今回との差分を確実に検出することができる。

【０１４５】

ＶＭ４０は、前述した通り、ハイパバイザ２０によって構築される仮想マシンである。ＶＭ４０内で動作するＯＳは、ＰＣＩｅデバイス１０ｅのドライバ４２により、ＰＣＩｅデバイス１０ｅを使用して外部とデータのやり取りを実施する。

【０１４６】

〔３－２〕第２実施形態のＦＴサーバの動作

次に、図１０～図２０を参照しながら、上述した第２実施形態のＦＴサーバ１′の動作について説明する。

【０１４７】

〔３－２－１〕ＰＣＩｅトランザクション

まず、図１０における符号(1)～(9)と、図１１に示すフローチャート（ステップＳ２０１～Ｓ２０９）とに従って、図１２および図１３を参照しながら、図９に示すＦＴサーバ１′の動作について説明する。以下の説明における項目(1)～(9)の手順は、それぞれ図１０における符号(1)～(9)に対応している。また、図１１のステップＳ２０１～Ｓ２０９は、それぞれ項目(1)～(9)の手順に対応している。

【０１４８】

なお、図１０は、図９に示すＦＴサーバ１′におけるＰＣＩｅトランザクションの例を説明する図であり、特に、本システムがＰＣＩｅデバイス１０ｅを経由して外部にデータを送出するときのＰＣＩｅトランザクションに注目した図である。また、図１１は、図９に示すＦＴサーバ１′におけるシステム外へのデータ送出処理の概要を説明するフローチャートである。図１２は、図９に示すＦＴサーバ１′におけるデータ転送時のメモリマップ例を示す図である。図１３は、図９に示すＦＴサーバ１′におけるＶＭアドレスと物理アドレスとのメモリマッピング例を示す図である。

【０１４９】

(1) ＶＭ４０上のＰＣＩｅデバイス１０ｅのドライバ４２は、図１２に示すように、外部にデータを送出するために、ＶＭ４０に割り当てられたメモリ上に、送出したいデータ（data）を書き込む。また、同ドライバ４２は、当該データのＶＭ４０のメモリ上のアドレス（address）と長さ（length）をＶＭ４０のメモリ上の領域に書き込む（図１０の(1)および図１１のステップＳ２０１参照）。このアドレスと長さを書き込むメモリ領域のアドレスは、事前にＰＣＩｅデバイス１０ｅのドライバ４２により、ＰＣＩｅデバイス１０ｅに通知されている。

【０１５０】

(2) ＶＭ４０上のＰＣＩｅデバイス１０ｅのドライバ４２は、ＰＣＩｅデバイス１０ｅにデータの送出を指示するために、ＰＣＩｅデバイス１０ｅのレジスタにデータ送出指示（データ転送指示）を書き込む。ＰＣＩｅデバイス１０ｅのレジスタは、図１３に示すように、ＶＭ４０のメモリにメモリマップされているので、メモリマップされたレジスタへの書込みは、メモリコントローラ１０ｂによってＰＣＩｅルートコンプレクス１０ｄに送られる。そして、ＰＣＩｅルートコンプレクス１０ｄは、上記レジスタへの書込みに応じ

て、Memory Write Requestトランザクションパケット（ＭＷｒ）を生成してＰＣＩｅデバイス１０ｅへ送信する（図１０の(2) Request DMA transfer, 図１０の矢印Ａ２１および図１１のステップＳ２０２参照）。

【０１５１】

(3) ＰＣＩｅデバイス１０ｅは、Memory Write Requestトランザクションパケットの内容に基づきデータ送出指示が与えられたことを知る。このデータ送出指示に応じて、ＰＣＩｅデバイス１０ｅは、ＰＣＩｅルートコンプレクス１０ｄにMemory Read Requestトランザクションパケット（ＭＲｄ）を送信する（図１０の(3) Request address/length, 図１０の矢印Ａ２２および図１１のステップＳ２０３参照）。ここでＰＣＩｅルートコンプレクス１０ｄに送信されるパケットの内容は、address/length情報の取得指示である。

【０１５２】

(4) Memory Read RequestトランザクションパケットがＰＣＩｅルートコンプレクス１０ｄに到着すると、ＤＭＡreadリクエスト監視部１２がMemory Read Requestトランザクションパケットを検出し、以下の項目(4-1)～(4-5)の処理が実行される（図１０の(4) Syncおよび図１１のステップＳ２０４参照）。

【０１５３】

(4-1) ＤＭＡreadリクエスト監視部１２は、ＰＣＩｅルートコンプレクス１０ｄに到着するＰＣＩｅトランザクションパケットを監視する。到着したパケットがＶＭ４０に割り合てられていないＰＣＩｅデバイス（例えば図９に示すＰＣＩｅデバイス１０ｆ）からのパケットである場合、ＤＭＡreadリクエスト監視部１２は、単にＤＭＡ制御部１１にそのパケットを送付する。この送付処理は、後述する図１４のステップＳ２１３のＮＯルートからステップＳ２１９の処理に相当する。どのＰＣＩｅデバイスからのＰＣＩｅトランザクションパケットであるかは、パケットのヘッダを参照することで判別することができる。その判別方法は、ＰＣＩ-Express仕様の範囲内である。

【０１５４】

(4-2) さて、ＰＣＩｅデバイス１０ｅからのMemory Read Requestトランザクションパケット（ＭＲｄ）を検出したＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理中か否かを判断するために、ＤＭＡ割込み制御部１６においてMemory Read Request割込またはCompletion with Data割込みの要因フラグ１６ａがセットされているか否かを確認する。もしセットされていれば、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理中とみなせるので、検出したMemory Read RequestトランザクションパケットをＤＭＡreadリクエストバッファ部１３へ送信する。ＤＭＡreadリクエストバッファ部１３は、送信されたパケットをＦＩＦＯ（First-in First-out）バッファに入れる。これにより、ＰＣＩｅデバイス１０ｅから外部へのデータ出力が抑止される。

【０１５５】

(4-3) ＤＭＡ割込み制御部１６において割込み要因フラグ１６ａがセットされていなければ、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理中ではないとみなせる。ここで、ＤＭＡreadリクエスト監視部１２は、Dirtyカウンタ１７ａの現在値（現在の計数値）を参照する。また、保存領域１７ｂには、前回Dirtyカウンタ１７ａを参照したときの計数値が保存されている。ＤＭＡreadリクエスト監視部１２は、Dirtyカウンタ１７ａの現在値と、前回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する。カウンタ１７ａの現在値と保存領域１７ｂの値とが異なる場合、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなせる。そこで、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１に同期処理を実行させるべく、ＤＭＡ割込み制御部１６に、Memory Read Requestトランザクションパケットを検出したことを要因とする割込み（VMexit）を発生させる。このとき、ＤＭＡ割込み制御部１６の割込み要因フラグ１６ａが設定される。

【０１５６】

(4-4) 上記項目(4-3)で発生された割込みを検知したＦＴ制御部２１は、同期処理を行なう。同期処理を完了したＦＴ制御部２１は、割込み要因フラグ１６ａをクリアし、Ｖ

Ｍ４０に処理を移す。割込み要因フラグ１６ａをクリアしたとき、ＤＭＡ割込み制御部１６は、ＤＭＡreadリクエストバッファ部１３に対し、留めているMemory Read Requestトランザクションパケットを ＤＭＡ制御部１１へ送信する指示を出す。この指示を受けたＤＭＡreadリクエストバッファ部１３は、留めている全てのMemory Read Requestトランザクションパケットを ＤＭＡ制御部１１へ送信する。なお、ＦＴ制御部２１の同期処理は、第１実施形態の手順５および手順６と同様にして実行されるので、ここでは、その説明を省略する。

【０１５７】

(4-5) ＤＭＡ割込み制御部１６において割込み要因フラグ１６ａがセットされておらず、カウンタ１７ａの現在値と保存領域１７ｂの値とが一致していた場合、プライマリとセカンダリとの同期がとれており同期処理すべきメモリ差分が存在しない。したがって、ＤＭＡreadリクエスト監視部１２は、Memory Read Requestトランザクションパケットを ＤＭＡ制御部１１へ送信する。この送信処理は、後述する図１４のステップＳ２１７のＹＥＳルートからステップＳ２１９の処理に相当する。以上の処理により、不要な割込みを削減することができるので、システム全体の処理が高速化する。

【０１５８】

(5) ＤＭＡ制御部１１は、Memory Read Requestトランザクションパケットを受信すると、ＤＭＡリマップ機能１１ｂを用いて、当該パケットの内容をメモリコントローラ１０ｂへの指示に変換し、メモリの内容（ここではaddress/length）を読み出す（図１０の矢印Ａ２３参照）。そして、ＤＭＡ制御部１１は、メモリ１０ｂから読み出した内容をCompletion with Dataトランザクションパケット（ＣｐｌＤ）に格納してＰＣＩｅデバイス１０ｅに送信する（図１０の矢印Ａ２４／(5) Get address/lengthおよび図１１のステップＳ２０５参照）。

【０１５９】

(6) Completion with Dataトランザクションパケットを取得したＰＣＩｅデバイス１０ｅは、当該パケットに格納されているaddress/lengthを参照する。そして、ＰＣＩｅデバイス１０ｅは、参照したアドレスからデータ（data）を読み出すべく、Memory Read Requestトランザクションパケット（ＭＲｄ）をＰＣＩｅルートコンプレクス１０ｄへ送信する（図１０の(6) Request data／矢印Ａ２５および図１１のステップＳ２０６参照）。なお、データの長さが長ければ、複数のMemory Read Requestトランザクションパケットが送信される場合もある。

【０１６０】

(7) Memory Read Requestトランザクションパケットを受信したＰＣＩｅルートコンプレクス１０ｄでは、上記項目(4)および(4-1)～(4-5)と同様の処理が実行される（図１０の(7) Syncおよび図１１のステップＳ２０７参照）。なお、本項目(7)の処理を実行するのは、上記項目(4)および(4-1)～(4-5)の処理で同期処理を実行した後であるため、同期処理すべきメモリ差分が存在しない場合が多い。

【０１６１】

(8) ＤＭＡ制御部１１は、Memory Read Requestトランザクションパケットを受信すると、ＤＭＡリマップ機能１１ｂを用いて、当該パケットの内容をメモリコントローラ１０ｂへの指示に変換し、メモリの内容（ここではdata）を読み出す（図１０の矢印Ａ２６参照）。そして、ＤＭＡ制御部１１は、メモリ１０ｂから読み出した内容をCompletion with Dataトランザクションパケット（ＣｐｌＤ）に格納してＰＣＩｅデバイス１０ｅに送信する（図１０の矢印Ａ２７／(8) Get dataおよび図１１のステップＳ２０８参照）。

【０１６２】

(9) Completion with Dataトランザクションパケットによってデータを受信したＰＣＩｅデバイス１０ｅは、転送すべきデータ（data）を受け取って、当該データの転送を行なうことにより、ＤＭＡ転送を完了する。そして、ＰＣＩｅデバイスは、ＤＭＡ転送が完了したことをＶＭ４０上のＯＳのドライバ４２に知らせるために、ＰＣＩ-Express仕様で規定されているＭＳＩ割込みを発行する（図１０の矢印Ａ２８／(9) Completion with MSI

および図１１のステップＳ２０９参照）。ＭＳＩ割込みは、具体的には、Memory Write Requestトランザクションパケット（ＭＷｒ）である。

【０１６３】

〔３－２－２〕

次に、図９に示すＦＴサーバ１′において、ＰＣＩｅデバイス１０ｅからＶＭ４０のメモリへデータを転送する処理に際し、ＦＴ制御部２１が、ＤＭＡ転送されたデータをメモリ差分として把握する動作（下記手順(11)～手順(14)）について説明する。なお、下記手順(11)～手順(14)は、それぞれ、第１実施形態で説明した手順２１～手順２４に対応している。

【０１６４】

(11) ＰＣＩｅデバイス１０ｅは、ＶＭ４０のメモリアドレスにデータを書き込むためにMemory Write Requestトランザクションパケット（ＭＷｒ）を送信する。

【０１６５】

(12) ＤＭＡreadリクエスト監視部１２は、Memory Write Requestトランザクションパケット（ＤＭＡwriteリクエスト）をＤＭＡ制御部１１に送信する。

【０１６６】

(13) ＤＭＡ制御部１１のＤＭＡリマップ機能１１ｂは、予めハイパバイザ２０のＦＴ制御部２１が用意した、ＶＭメモリアドレスと物理メモリアドレスとの変換テーブルを参照し、Memory Write Requestトランザクションパケット内に含まれるアドレスの変換を行なう。つまり、ＤＭＡリマップ機能１１ｂは、Memory Write Requestトランザクションパケット内の物理メモリアドレスをＶＭメモリアドレスに変換する。

【０１６７】

(14) ＤＭＡwrite監視部１１ａは、ハイパバイザ２０が用意したＶＭメモリ管理テーブル２２を参照し、当該テーブル２２において、Memory Write Requestトランザクションパケット内のアドレスに対応するDirtyビット（変更ビット；図２０参照）に“１”を設定する。これにより、Memory Write Requestトランザクションパケット内のアドレスに対応するデータに変更が生じたことが、ＶＭメモリ管理テーブル２２に記録される。その後、ＤＭＡwrite監視部１１ａは、メモリコントローラ（メモリ）１０ｂへメモリアクセスリクエストを送信する。

【０１６８】

以上の手順(11)～(14)により、ＶＭメモリ管理テーブル２２にメモリの変更があったことが記録されるので、ハイパバイザ２０のＦＴ制御部２１は、同期処理時に、プライマリからセカンダリに転送すべきメモリ変更差分を確実に把握することができる。

【０１６９】

〔３－２－３〕ＤＭＡreadリクエスト監視部の動作

次に、図１５（Ａ）～図１５（Ｃ）を参照しながら、図１４に示すフローチャート（ステップＳ２１１～Ｓ２１９）に従って、図９に示すＦＴサーバ１′におけるＤＭＡreadリクエスト監視部１２の処理（動作）について説明する。ここで説明するＤＭＡreadリクエスト監視部１２の処理（動作）は、上述した項目(4-1)～(4-3)の手順で実行される。

【０１７０】

なお、図１５（Ａ）は、ＰＣＩｅトランザクションリクエストヘッダのフォーマット例を示す図、図１５（Ｂ）は、ＰＣＩｅトランザクション完了ヘッダのフォーマット例を示す図である。また、図１５（Ｃ）は、図９に示すＦＴサーバ１′におけるＤＭＡ割込み制御部１６の監視対象ＰＣＩｅデバイス管理テーブル１６ｂを示す図である。

【０１７１】

ＤＭＡreadリクエスト監視部１２は、ＰＣＩｅデバイス１０ｅからのＰＣＩｅトランザクションパケットを待機し（ステップＳ２１１）、パケットが来ない場合（ステップＳ２１２のＮＯルート）、ステップＳ２１１の処理に戻る。パケットが来た場合（ステップＳ２１２のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、受信したパケットが、監視対象ＰＣＩｅデバイス１０ｅからのMemory Read Requestトランザクションパケット（

ＤＭＡreadリクエスト）であるか否かを判断する（ステップＳ２１３）。

【０１７２】

このとき、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１によって作成された、ＤＭＡ割込み制御部１６における監視対象ＰＣＩｅデバイス管理テーブル１６ｂ（図９および図１５（Ｃ）参照）を参照することにより、監視対象ＰＣＩｅデバイスを把握し、ステップＳ２１３の判断処理を行なう。さらに、ＤＭＡreadリクエスト監視部１２は、図１５（Ａ）や図１５（Ｂ）に示すようなＰＣＩｅトランザクションパケットのヘッダのRequesterＩＤを参照することで、当該パケットが監視対象ＰＣＩｅデバイスに該当するか否かを判断することができる。

【０１７３】

ハイパバイザ２０がＤＭＡリマップ機能１１ｂを用いてＶＭ４０にＰＣＩｅデバイスを割り当てる際、ＦＴ制御部２１は、図１５（Ｃ）に示すように、監視対象ＰＣＩｅデバイス管理テーブル１６ｂにおいて、当該ＰＣＩｅデバイスに相当するRequester IDを、当該ＰＣＩｅデバイスを使用するＶＭ４０の識別子（ＶＭＩＤ）に対応付けて設定する。ＤＭＡ割込み制御部１６は、本テーブル１６ｂを参照することで、割込みを発生させるべきＶＭ４０を判断することができる（図７のステップＳ１２４，Ｓ１２６および図１７のステップＳ２３４，Ｓ２３６参照）。

【０１７４】

受信したパケットがMemory Read Requestトランザクションパケットでない場合（ステップＳ２１３のＮＯルート）、ＤＭＡreadリクエスト監視部１２は、受信したパケットをＤＭＡ制御部１１に出力し（ステップＳ２１９）、処理を終了する。

【０１７５】

一方、受信したパケットがMemory Read Requestトランザクションパケットである場合（ステップＳ２１３のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、ＤＭＡ割込み制御部１６に、Memory Read Requestトランザクションパケットを検出したことによる割込み要因フラグ１６ａがセットされているか否かを判断する（ステップＳ２１４）。割込み要因フラグ１６ａがセットされている場合（ステップＳ２１４のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理中であるとみなし、受信したパケットをＤＭＡreadリクエストバッファ部１３に送出して保持させ（ステップＳ２１５）、処理を終了する。これにより、ＰＣＩｅデバイス１０ｅから外部へのデータ出力が抑止される。

【０１７６】

また、割込み要因フラグ１６ａがセットされていない場合（ステップＳ２１４のＮＯルート）、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１が同期処理を実行していないとみなす。そして、ＤＭＡreadリクエスト監視部１２は、同期処理を実行する必要があるか否かを確認するために、Dirtyカウンタ１７ａを参照し、Dirtyカウンタ１７ａの現在値と、前回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する（ステップＳ２１６）。

【０１７７】

Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致しない場合（ステップＳ２１７のＮＯルート）、ＤＭＡreadリクエスト監視部１２は、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなす。そして、ＤＭＡreadリクエスト監視部１２は、どのデバイス（ＰＣＩｅデバイス１０ｅ）からのMemory Read Requestトランザクションパケットを検出したのかをＤＭＡ割込み制御部１６に通知し、割込み要因フラグ１６ａをセットする（ステップＳ２１８）。これにより、ＤＭＡ割込み制御部１６からハイパバイザ２０に対して割込みが発生され、ＦＴ制御部２１が同期処理を実行する。なお、ＤＭＡreadリクエスト監視部１２は、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値との比較の終了後、保存領域１７ｂの値をDirtyカウンタ１７ａの現在値に置き換えるようメモリアクセス監視部１７に対して指示する。

【０１７８】

この後、ＤＭＡreadリクエスト監視部１２は、受信したパケットつまりMemory Read Requestトランザクションパケットを ＤＭＡreadリクエストバッファ部１３に送出して保持させ（ステップＳ２１５）、処理を終了する。これにより、ＰＣＩｅデバイス１０ｅから外部へのデータ出力が抑止される。

【０１７９】

Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致する場合（ステップＳ２１７のＹＥＳルート）、ＤＭＡreadリクエスト監視部１２は、プライマリとセカンダリとの同期がとれているとみなす。そして、ＤＭＡreadリクエスト監視部１２は、ＤＭＡreadリクエストバッファ部１３にMemory Read Requestトランザクションパケットを保存させることなくＤＭＡ制御部１１へ送出し（ステップＳ２１９）、処理を終了する。これにより、物理デバイス１０ｃから外部へのデータ出力が実行される。

【０１８０】

〔３－２－４〕ＤＭＡreadデータ監視部の動作

次に、図１６に示すフローチャート（ステップＳ２２１～Ｓ２２９）に従って、図９に示すＦＴサーバ１'におけるＤＭＡreadデータ監視部１４の処理（動作）について説明する。ここで説明するＤＭＡreadデータ監視部１４の処理（動作）は、上述した項目(7)の手順で実行される。

【０１８１】

ＤＭＡreadデータ監視部１４は、ＤＭＡ制御部１１からのＰＣＩｅトランザクションパケットを待機し（ステップＳ２２１）、パケットが来ない場合（ステップＳ２２２のＮＯルート）、ステップＳ２２１の処理に戻る。パケットが来た場合（ステップＳ２２２のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、受信したパケットがCompletion with Dataトランザクションパケット（ＤＭＡreadデータ）であるか否かを判断する（ステップＳ２２３）。

【０１８２】

このとき、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１によって作成された、ＤＭＡ割込み制御部１６における監視対象ＰＣＩｅデバイス管理テーブル１６ｂ（図９および図１５（Ｃ）参照）を参照することにより、監視対象ＰＣＩｅデバイスを把握し、ステップＳ２２３の判断処理を行なう。さらに、ＤＭＡreadデータ監視部１４は、図１５（Ａ）や図１５（Ｂ）に示すようなＰＣＩｅトランザクションパケットのヘッダのRequester ＩＤを参照することで、当該パケットが監視対象ＰＣＩｅデバイスに該当するか否かを判断することができる。

【０１８３】

受信したパケットがCompletion with Dataトランザクションパケットでない場合（ステップＳ２２３のＮＯルート）、ＤＭＡreadデータ監視部１４は、受信したパケットをＰＣＩｅデバイス１０ｅに出力し（ステップＳ２２９）、処理を終了する。

【０１８４】

一方、受信したパケットがCompletion with Dataトランザクションパケットである場合（ステップＳ２２３のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、ＤＭＡ割込み制御部１６に、Completion with Dataトランザクションパケットを検出したことによる割込み要因フラグ１６ａがセットされているか否かを判断する（ステップＳ２２４）。割込み要因フラグ１６ａがセットされている場合（ステップＳ２２４のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１が同期処理中であるとみなし、受信したパケットをＤＭＡreadデータバッファ部１５に送出して保持させ（ステップＳ２２５）、処理を終了する。これにより、ＰＣＩｅデバイス１０ｅから外部へのデータ出力が抑止される。

【０１８５】

また、割込み要因フラグ１６ａがセットされていない場合（ステップＳ２２４のＮＯルート）、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１が同期処理を実行していないとみなす。そして、ＤＭＡreadデータ監視部１４は、同期処理を実行する必要があるか否かを確認するために、Dirtyカウンタ１７ａを参照し、Dirtyカウンタ１７ａの現在値と、前

回参照したときのDirtyカウンタ１７ａの値（保存領域１７ｂの値）とを比較する（ステップＳ２２６）。

【０１８６】

Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致しない場合（ステップＳ２２７のＮＯルート）、ＤＭＡreadデータ監視部１４は、ＦＴ制御部２１による同期処理が必要なメモリ差分がＶＭメモリ１０ｂ上に存在しているとみなす。そして、ＤＭＡreadデータ監視部１４は、どのデバイス（ＰＣＩｅデバイス１０ｅ）へのCompletion with Dataトランザクションパケットを検出したのかをＤＭＡ割込み制御部１６に通知し、割込み要因フラグ１６ａをセットする（ステップＳ２２８）。これにより、ＤＭＡ割込み制御部１６からハイパバイザ２０に対して割込みが発生され、ＦＴ制御部２１が同期処理を実行する。なお、ＤＭＡreadデータ監視部１４は、Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値との比較の終了後、保存領域１７ｂの値をDirtyカウンタ１７ａの現在値に置き換えるようメモリアクセス監視部１７に対して指示する。

【０１８７】

この後、ＤＭＡreadデータ監視部１４は、受信したＤＭＡパケットつまりCompletion with Dataトランザクションパケットを ＤＭＡreadデータバッファ部１５に送出して保持させ（ステップＳ２２５）、処理を終了する。これにより、ＰＣＩｅデバイス１０ｅから外部へのデータ出力が抑止される。

【０１８８】

Dirtyカウンタ１７ａの現在値と保存領域１７ｂの値とが一致する場合（ステップＳ２２７のＹＥＳルート）、ＤＭＡreadデータ監視部１４は、プライマリとセカンダリとの同期がとれているとみなす。そして、ＤＭＡreadデータ監視部１４は、ＤＭＡreadデータバッファ部１５にCompletion with Dataトランザクションパケットを保存させることなくＰＣＩｅデバイス１０ｅへ送出し（ステップＳ２２９）、処理を終了する。これにより、ＰＣＩｅデバイス１０ｅから外部へのデータ出力が実行される。

【０１８９】

〔３－２－５〕ＤＭＡ割込み制御部の動作

次に、図１７および図１８に示すフローチャート（ステップＳ２３１～Ｓ２４２）に従って、図９に示すＦＴサーバ１′におけるＤＭＡ割込み制御部１６の処理（動作）について説明する。ここで説明するＤＭＡ割込み制御部１６の処理（動作）は、上述した項目(4-4), (4-5), (7)の手順で実行される。

【０１９０】

ＤＭＡ割込み制御部１６は、ＤＭＡreadリクエスト監視部１２，ＤＭＡreadデータ監視部１４，ＦＴ制御部２１のいずれかからの指示を待機し（ステップＳ２３１）、指示が来ない場合（ステップＳ２３２のＮＯルート）、ステップＳ２３１の処理に戻る。指示が来た場合（ステップＳ２３２のＹＥＳルート）、ＤＭＡ割込み制御部１６は、受信した指示がＤＭＡreadリクエスト監視部１２からの指示であるか否かを判断する（ステップＳ２３３）。

【０１９１】

受信した指示がＤＭＡreadリクエスト監視部１２からの指示である場合（ステップＳ２３３のＹＥＳルート）、ＤＭＡ割込み制御部１６は、予めＦＴ制御部２１により設定された割込み送信先に、Memory Read Requestトランザクションパケット検出割込みを送信し（ステップＳ２３４）、ステップＳ２３１の処理に戻る。このとき、ＤＭＡ割込み制御部１６は、監視対象ＰＣＩｅデバイス管理テーブル１６ｂや、Memory Read Requestトランザクションパケットのヘッダの RequesterＩＤを参照することで、割込み送信先を判断する。

【０１９２】

受信した指示がＤＭＡreadリクエスト監視部１２からの指示でない場合（ステップＳ２３３のＮＯルート）、ＤＭＡ割込み制御部１６は、受信した指示がＤＭＡreadデータ監視部１４からの指示であるか否かを判断する（ステップＳ２３５）。

【０１９３】
受信した指示がＤＭＡreadデータ監視部１４からの指示である場合（ステップＳ２３５のＹＥＳルート）、ＤＭＡ割込み制御部１６は、予めＦＴ制御部２１により設定された割込み送信先に、ＤＭＡreadデータ検出割込みを送信し（ステップＳ２３６）、ステップＳ２３１の処理に戻る。このときも、ＤＭＡ割込み制御部１６は、監視対象ＰＣＩｅデバイス管理テーブル１６ｂや、Completion with Dataトランザクションパケットのヘッダの Requester ＩＤを参照することで、割込み送信先を判断する。
【０１９４】
受信した指示がＤＭＡreadデータ監視部１４からの指示でない場合（ステップＳ２３５のＮＯルート）、ＤＭＡ割込み制御部１６は、受信した指示がＦＴ制御部２１からの指示であるか否かを判断する（ステップＳ２３７）。
【０１９５】
受信した指示がＦＴ制御部２１からの指示でない場合（ステップＳ２３７のＮＯルート）、ＤＭＡ割込み制御部１６は、ステップＳ２３１の処理に戻る。一方、受信した指示がＦＴ制御部２１からの指示である場合（ステップＳ２３７のＹＥＳルート）、ＤＭＡ割込み制御部１６は、受信した指示がMemory Read Request検出割込みフラグ１６ａのクリア指示であるか否かを判断する（ステップＳ２３８）。
【０１９６】
受信した指示がMemory Read Request検出割込みフラグ１６ａのクリア指示である場合（ステップＳ２３８のＹＥＳルート）、ＤＭＡ割込み制御部１６は、ＤＭＡreadリクエストバッファ部１３に対し、Memory Read Requestトランザクションパケットを ＤＭＡ制御部１１へ送信するよう指示する（ステップＳ２３９）。この後、ＤＭＡ割込み制御部１６は、ステップＳ２３１の処理に戻る。
【０１９７】
受信した指示がMemory Read Request検出割込みフラグ１６ａのクリア指示でない場合（ステップＳ２３８のＮＯルート）、ＤＭＡ割込み制御部１６は、受信した指示がCompletion with Data検出割込みフラグ１６ａのクリア指示であるか否かを判断する（ステップＳ２４０）。
【０１９８】
受信した指示がCompletion with Data検出割込みフラグ１６ａのクリア指示である場合（ステップＳ２４０のＹＥＳルート）、ＤＭＡ割込み制御部１６は、ＤＭＡreadデータバッファ部１５に対し、Completion with Dataトランザクションパケットを ＰＣＩｅデバイス１０ｅへ送信するよう指示する（ステップＳ２４１）。この後、ＤＭＡ割込み制御部１６は、ステップＳ２３１の処理に戻る。
【０１９９】
受信した指示がCompletion with Data検出割込みフラグ１６ａのクリア指示でない場合（ステップＳ２４０のＮＯルート）、受信した指示は、ＦＴ制御部２１からのその他の指示であるので、ＤＭＡ割込み制御部１６は、当該指示に応じた適切な処理を実行し（ステップＳ２４２）、ステップＳ２３１の処理に戻る。
【０２００】
〔３－２－６〕ＤＭＡwrite監視部の動作
次に、図２０を参照しながら、図１９に示すフローチャート（ステップＳ２５１～Ｓ２５５）にしたがって、図９に示すＦＴサーバ１′におけるＤＭＡwrite監視部１１ａの処理（動作）について説明する。ここで説明するＤＭＡwrite監視部１１ａの処理（動作）は、上述した手順(11)～(14)の手順で実行される。なお、図２０は、図９に示すＦＴサーバ１′におけるＶＭメモリ管理テーブル２２の例を示す図である。
【０２０１】
ＤＭＡwrite監視部１１ａは、ＰＣＩｅデバイス１０ｅからのMemory Write Requestトランザクションパケット（ＭＷｒ）を待機し（ステップＳ２５１）、パケット（ＭＷｒ）が来ない場合（ステップＳ２５２のＮＯルート）、ステップＳ２５１の処理に戻る。Memory

Write Requestトランザクションパケット（ＭＷｒ）が来た場合（ステップＳ２５２のＹＥＳルート）、ＤＭＡwrite監視部１１ａは、受信したパケット（ＭＷｒ）内のaddress（ＶＭアドレス）をキーにして、図２０に示すようなＶＭメモリ管理テーブル２２を検索する。これにより、ＤＭＡwrite監視部１１ａは、パケット（ＭＷｒ）内のＶＭアドレスにおけるデータが変更されたことを示すDirtyビット（変更ビット）に“１”がセットされているか否かを確認する（ステップＳ２５３）。

【０２０２】

Dirtyビットに“１”がセットされている場合（ステップＳ２５４のＹＥＳルート）、つまり受信したパケットによる書込み対象のデータが既に変更されている場合、ＤＭＡwrite監視部１１ａは、ステップＳ２５１の処理に戻る。

【０２０３】

一方、Dirtyビットに“１”がセットされていない場合、つまりDirtyビットが“０”である場合（ステップＳ２５４のＮＯルート）、ＤＭＡwrite監視部１１ａは、ＶＭメモリ管理テーブル２２において、パケット（ＭＷｒ）内のＶＭアドレスに対応するDirtyビットに“１”をセットする。そして、ＤＭＡwrite監視部１１ａは、メモリアクセス監視部１７に対し、Dirtyカウンタ１７ａを１だけインクリメントするよう指示してから（ステップＳ２５５）、ステップＳ２５１の処理に戻る。

【０２０４】

なお、ＶＭメモリ管理テーブル２２におけるDirtyビットは、通常、ＣＰＵ（ＣＰＵコア１０ａ）上のプログラムによってメモリ１０ｂ上のデータを書き換えた場合にセットされている。本実施形態のDirtyビットは、これに加え、ネットワーク側（物理デバイス１０ｃまたはＰＣＩｅデバイス１０ｅ側）からのアクセス（ＤＭＡwriteリクエストまたはMemory Write Request）によってメモリ１０ｂ上のデータを書き換えた場合にもセットされるようになっている。

【０２０５】

〔３－３〕第２実施形態のＦＴサーバの効果

このように、物理デバイスとしてＰＣＩｅデバイス１０ｅを有する第２実施形態のＦＴサーバ１′において、ＰＣＩｅルートコンプレクス１０ｄを、ＤＭＡ制御部１１，ＤＭＡreadリクエスト監視部１２，ＤＭＡreadリクエストバッファ部１３，ＤＭＡreadデータ監視部１４，ＤＭＡreadデータバッファ部１５およびＤＭＡ割込み制御部１６として機能させることにより、第１実施形態と同様の作用効果(g1)～(g6)を得ることができる。

【０２０６】

〔４〕その他

以上、本発明の好ましい実施形態について詳述したが、本発明は、係る特定の実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲内において、種々の変形、変更して実施することができる。

【０２０７】

上述したＤＭＡ制御部１１，ＤＭＡwrite監視部１１ａ，ＤＭＡリマップ機能１１ｂ，ＤＭＡreadリクエスト監視部１２，ＤＭＡreadリクエストバッファ部１３，ＤＭＡreadデータ監視部１４，ＤＭＡreadデータバッファ部１５，ＤＭＡ割込み制御部１６，メモリアクセス監視部１７，ハイパバイザ２０，ＦＴ制御部２１，管理ＶＭ３０およびＶＭ４０としての機能を含む、本実施形態のＦＴサーバ１，１′の機能の全部もしくは一部は、コンピュータ（ＣＰＵ，各種端末を含む）が所定のプログラムを実行することによって実現される。

【０２０８】

そのプログラムは、例えばフレキシブルディスク，ＣＤ（ＣＤ－ＲＯＭ，ＣＤ－Ｒ，ＣＤ－ＲＷなど），ＤＶＤ（ＤＶＤ－ＲＯＭ，ＤＶＤ－ＲＡＭ，ＤＶＤ－Ｒ，ＤＶＤ－ＲＷ，ＤＶＤ＋Ｒ，ＤＶＤ＋ＲＷなど），ブルーレイディスク等のコンピュータ読取可能な記録媒体に記録された形態で提供される。この場合、コンピュータはその記録媒体からプログラムを読み取って内部記憶装置または外部記憶装置に転送し格納して用いる。

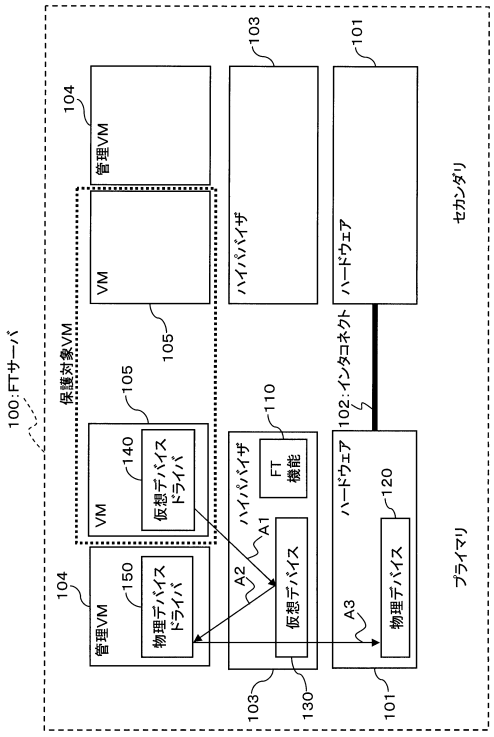
【０２０９】
〔５〕付記
以上の実施形態に関し、さらに以下の付記を開示する。
（付記１）
入出力デバイスを用いてデータ入出力を行なうドライバを含む仮想計算機と、
前記仮想計算機を管理するとともに、本情報処理装置と対になる他の情報処理装置との同期処理を行なうハイパバイザと、
メモリと前記入出力デバイスとの間で行なわれるアクセスを監視する監視部と、
前記監視部が前記メモリから前記入出力デバイスへのデータ出力に係る読出アクセスを検出した場合、前記読出アクセスを検出した旨を前記ハイパバイザに通知する通知部と、
前記ハイパバイザが前記通知部からの通知を受けるときに前記同期処理の実行中である場合、前記データ出力を抑止する抑止部と、を有する、情報処理装置。
【０２１０】
（付記２）
前記抑止部は、前記ハイパバイザが前記同期処理を終了すると、前記データ出力の抑止を解除する、付記１記載の情報処理装置。
【０２１１】
（付記３）
前記読出アクセスは、前記入出力デバイスから前記メモリへの読出リクエストの発行、または、前記メモリから前記入出力デバイスへの前記データ出力の対象データの読出である、付記１または付記２に記載の情報処理装置。
【０２１２】
（付記４）
前記抑止部は、前記読出リクエストまたは前記対象データをバッファに一時的に保持することにより前記データ出力を抑止し、前記ハイパバイザが前記同期処理を終了すると、前記読出リクエストを前記バッファから前記メモリ側へ送信、または、前記対象データを前記バッファから前記入出力デバイスへ送信する、付記３記載の情報処理装置。
【０２１３】
（付記５）
前記監視部は、前記入出力デバイスから前記メモリへのデータ書込に係る書込アクセスを検出した場合、前記メモリを管理するテーブルに、前記データ書込に係るアドレスに対応するデータに変更が生じたことを記録し、
前記ハイパバイザは、前記テーブルに従って前記同期処理を行なう、付記１～付記４のいずれか一項に記載の情報処理装置。
【０２１４】
（付記６）
前記監視部は、前記テーブルにおいて変更ビットを設定することにより前記変更が生じたことを記録し、
前記ハイパバイザは、前記同期処理を終了すると、前記変更ビットをクリアする、付記５記載の情報処理装置。
【０２１５】
（付記７）
前記メモリにおいてデータの変更が発生した箇所の数を計数するカウンタと、
前回の同期処理時に参照された前記カウンタによる計数値を保存する保存領域と、を有し、
前記ハイパバイザが前記通知部からの通知を受けたときに前記同期処理の実行中でない場合、前記監視部は、前記カウンタの値と前記保存領域の計数値とを比較し、前記カウンタの値と前記保存領域の計数値とが不一致であれば、前記抑止部に前記データ出力を抑止させ、前記ハイパバイザに前記同期処理を実行させる、付記１～付記６のいずれか一項に記載の情報処理装置。

【０２１６】
（付記８）
前記監視部は、前記カウンタの値と前記保存領域の計数値とが一致していれば、前記抑止部に前記データ出力を抑止させることなく、前記データ出力を実行させる、付記７記載の情報処理装置。
【０２１７】
（付記９）
前記監視部が、前記ハイパバイザに前記同期処理を実行させるべく、前記読出アクセスを検出したことを要因とする割込みを発生させた場合に設定される一方、前記ハイパバイザが前記同期処理を終了するとクリアされる割込み要因フラグを有し、
前記監視部は、前記割込み要因フラグが設定されている場合に前記ハイパバイザが前記同期処理の実行中であると判断する一方、前記割込み要因フラグが未設定の場合に前記ハイパバイザが前記同期処理の実行中ではないと判断する、付記１～付記８のいずれか一項に記載の情報処理装置。
【０２１８】
（付記１０）
入出力デバイスを用いてデータ入出力を行なうドライバを含む仮想計算機と、前記仮想計算機を管理するとともに本情報処理装置と対になる他の情報処理装置との同期処理を行なうハイパバイザと、を有する情報処理装置の制御方法であって、
メモリと前記入出力デバイスとの間で行なわれるアクセスを監視し、
前記監視によって前記メモリから前記入出力デバイスへのデータ出力に係る読出アクセスを検出した場合、前記読出アクセスを検出した旨を前記ハイパバイザに通知し、
前記ハイパバイザが前記通知を受けるときに前記同期処理の実行中である場合、前記データ出力を抑止する、情報処理装置の制御方法。
【０２１９】
（付記１１）
前記ハイパバイザが前記同期処理を終了すると、前記データ出力の抑止を解除する、付記１０記載の情報処理装置の制御方法。
【０２２０】
（付記１２）
前記読出アクセスは、前記入出力デバイスから前記メモリへの読出リクエストの発行、または、前記メモリから前記入出力デバイスへの前記データ出力の対象データの読出である、付記１０または付記１１に記載の情報処理装置の制御方法。
【０２２１】
（付記１３）
前記読出リクエストまたは前記対象データをバッファに一時的に保持することにより前記データ出力を抑止し、
前記ハイパバイザが前記同期処理を終了すると、前記読出リクエストを前記バッファから前記メモリ側へ送信、または、前記対象データを前記バッファから前記入出力デバイスへ送信する、付記１２記載の情報処理装置の制御方法。
【０２２２】
（付記１４）
前記監視によって前記入出力デバイスから前記メモリへのデータ書込に係る書込アクセスを検出した場合、前記メモリを管理するテーブルに、前記データ書込に係るアドレスに対応するデータに変更が生じたことを記録し、
前記ハイパバイザは、前記テーブルに従って前記同期処理を行なう、付記１０～付記１３のいずれか一項に記載の情報処理装置の制御方法。
【０２２３】
（付記１５）
前記テーブルにおいて変更ビットを設定することにより前記変更が生じたことを記録し

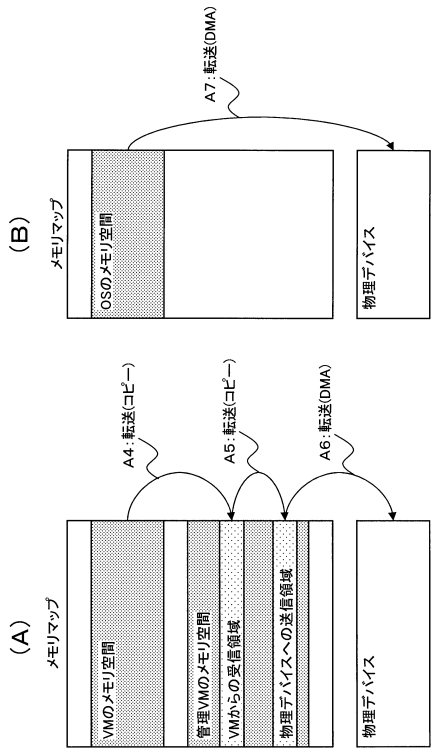
、
前記ハイパバイザは、前記同期処理を終了すると、前記変更ビットをクリアする、付記１４記載の情報処理装置の制御方法。
【０２２４】
（付記１６）
前記メモリにおいてデータの変更が発生した箇所の数をカウンタによって計数し、
前回の同期処理時に参照された前記カウンタによる計数値を保存領域に保存し、
前記ハイパバイザが前記通知部からの通知を受けたときに前記同期処理の実行中でない場合、前記カウンタの値と前記保存領域の計数値とを比較し、前記カウンタの値と前記保存領域の計数値とが不一致であれば、前記抑止部に前記データ出力を抑止させ、前記ハイパバイザに前記同期処理を実行させる、付記１０～付記１５のいずれか一項に記載の情報処理装置の制御方法。
【０２２５】
（付記１７）
前記カウンタの値と前記保存領域の計数値とが一致していれば、前記抑止部に前記データ出力を抑止させることなく、前記データ出力を実行させる、付記１６記載の情報処理装置の制御方法。
【０２２６】
（付記１８）
前記情報処理装置は、前記監視部が、前記ハイパバイザに前記同期処理を実行させるべく、前記読出アクセスを検出したことを要因とする割込みを発生させた場合に設定される一方、前記ハイパバイザが前記同期処理を終了するとクリアされる割込み要因フラグを有し、
前記割込み要因フラグが設定されている場合に前記ハイパバイザが前記同期処理の実行中であると判断する一方、前記割込み要因フラグが未設定の場合に前記ハイパバイザが前記同期処理の実行中ではないと判断する、付記１０～付記１７のいずれか一項に記載の情報処理装置の制御方法。
【符号の説明】
【０２２７】
１，１′　フォルトトレラントサーバ（ＦＴサーバ，情報処理装置）
１０，１０′　ハードウェア
１０Ａ，１０Ｂ　ＣＰＵサブシステム
１０ａ　ＣＰＵコア
１０ｂ　メモリおよびメモリコントローラ
１０ｃ　物理デバイス（入出力デバイス，Ｉ／Ｏデバイス，ネットワークカード）
１０ｄ　ＰＣＩｅルートコンプレクス（インタフェース）
１０ｅ，１０ｆ　ＰＣＩｅデバイス（物理デバイス，入出力デバイス，Ｉ／Ｏデバイス）
１１　ＤＭＡ制御部
１１ａ　ＤＭＡwrite監視部（監視部）
１１ｂ　ＤＭＡリマップ機能
１２　ＤＭＡreadリクエスト監視部（監視部，抑止部）
１３　ＤＭＡreadリクエストバッファ部（バッファ，抑止部）
１４　ＤＭＡreadデータ監視部（監視部，抑止部）
１５　ＤＭＡreadデータバッファ部（バッファ，抑止部）
１６　ＤＭＡ割込み制御部（通知部）
１６ａ　割込み要因フラグ
１６ｂ　監視対象ＰＣＩｅデバイス管理テーブル
１７　メモリアクセス監視部
１７ａ　Dirtyカウンタ（カウンタ）

１７ｂ　　保存領域
２０　　ハイパバイザ
２１　　ＦＴ制御部
２２　　ＶＭメモリ管理テーブル（テーブル）
３０　　管理ＶＭ
４０　　ＶＭ（仮想マシン，仮想計算機）
４１　　物理デバイスドライバ（ドライバ）
４２　　ＰＣＩｅデバイス１０ｅのドライバ（ドライバ）
５０　　インタコネクト

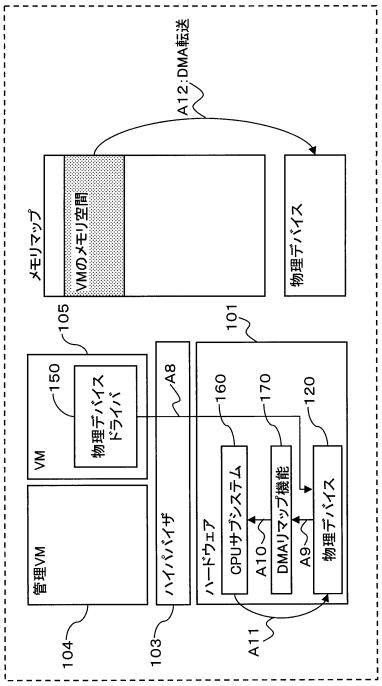
【図１】



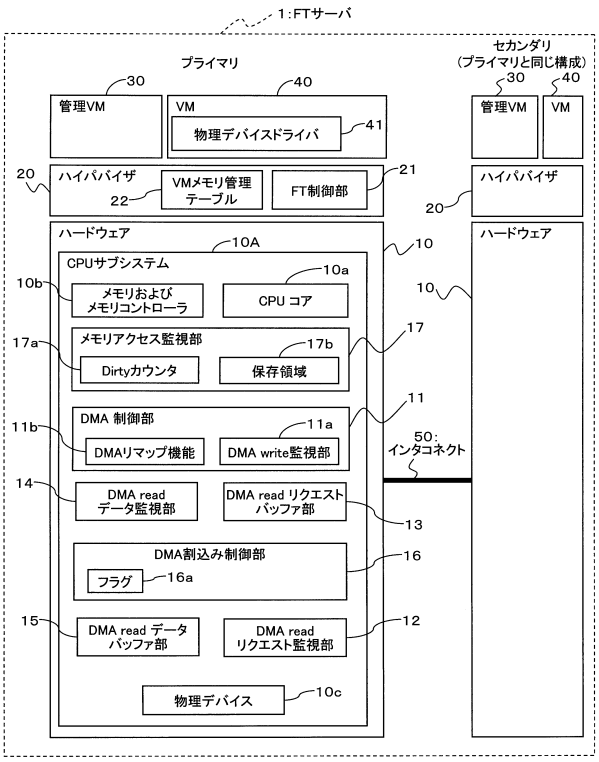
【図２】



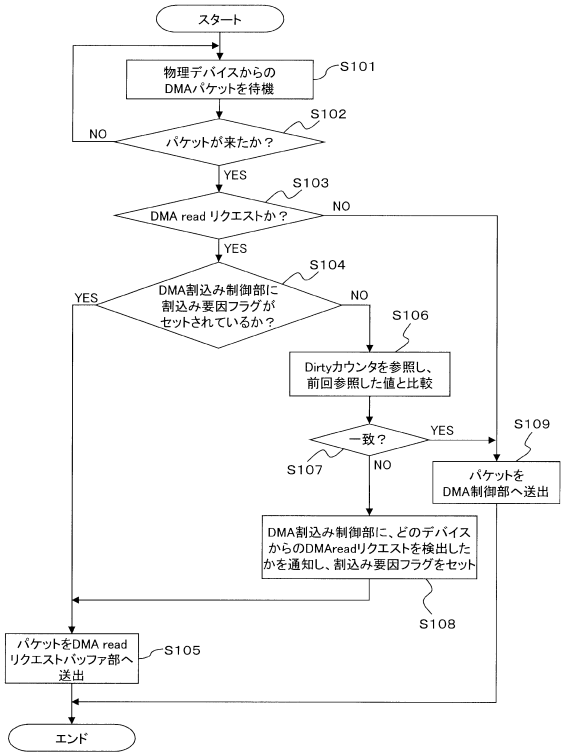
【図３】


A12:DMA転送
メモリマップ
VMのメモリ空間
物理デバイス
105
101
150
物理デバイスドライバ
A8
160
170
120
VM
管理VM
ハイパバイザ
ハードウェア
CPUサブシステム
A10
DMAリマップ機能
A9
物理デバイス
104
103
A11

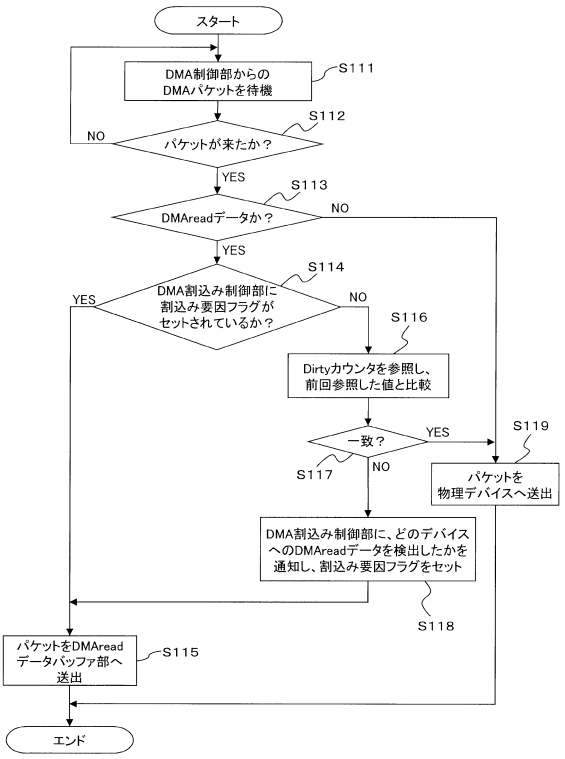
【図４】


1:FTサーバ
プライマリ
セカンダリ
(プライマリと同じ構成)
30
40
管理VM
VM
物理デバイスドライバ
41
20
ハイパバイザ
VMメモリ管理テーブル
22
FT制御部
21
ハードウェア
10A
10
CPUサブシステム
10a
10b
メモリおよびメモリコントローラ
CPU コア
17
メモリアクセス監視部
17a
Dirtyカウンタ
17b
保存領域
11
DMA 制御部
11b
DMAリマップ機能
11a
DMA write監視部
50:
インタコネクト
14
DMA read データ監視部
DMA read リクエストバッファ部
13
DMA割込み制御部
16
フラグ
16a
15
DMA read データバッファ部
DMA read リクエスト監視部
12
物理デバイス
10c

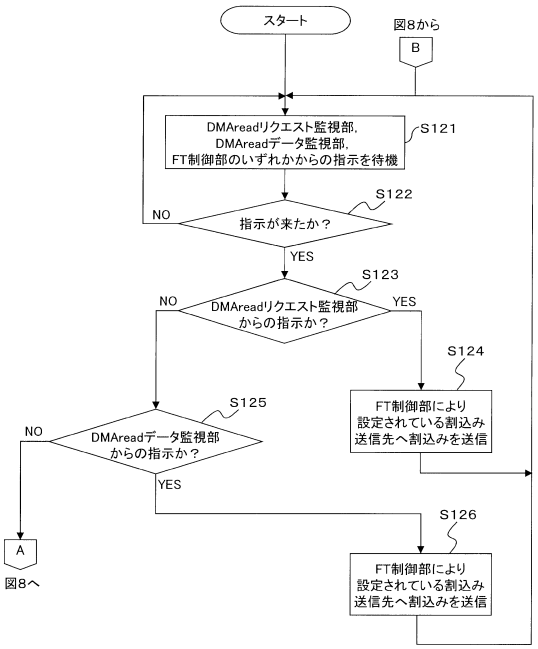
【図５】


スタート
物理デバイスからのDMAパケットを待機
S101
S102
NO
パケットが来たか？
YES
S103
DMA read リクエストか？
NO
YES
S104
YES
DMA割込み制御部に割込み要因フラグがセットされているか？
NO
S106
Dirtyカウンタを参照し、前回参照した値と比較
一致？
YES
S109
S107
NO
パケットをDMA制御部へ送出
DMA割込み制御部に、どのデバイスからのDMAreadリクエストを検出したかを通知し、割込み要因フラグをセット
S108
パケットをDMA read リクエストバッファ部へ送出
S105
エンド

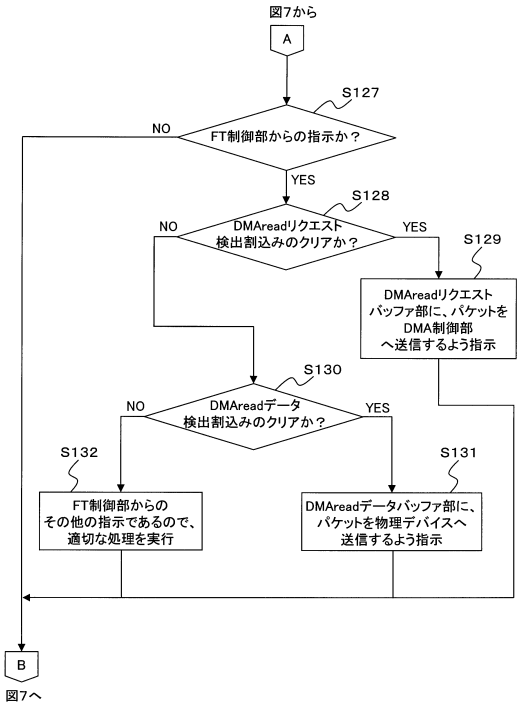
【図６】


スタート
DMA制御部からのDMAパケットを待機
S111
S112
NO
パケットが来たか？
YES
S113
DMAreadデータか？
NO
YES
S114
YES
DMA割込み制御部に割込み要因フラグがセットされているか？
NO
S116
Dirtyカウンタを参照し、前回参照した値と比較
一致？
YES
S119
S117
NO
パケットを物理デバイスへ送出
DMA割込み制御部に、どのデバイスへのDMAreadデータを検出したかを通知し、割込み要因フラグをセット
S118
パケットをDMAread データバッファ部へ送出
S115
エンド

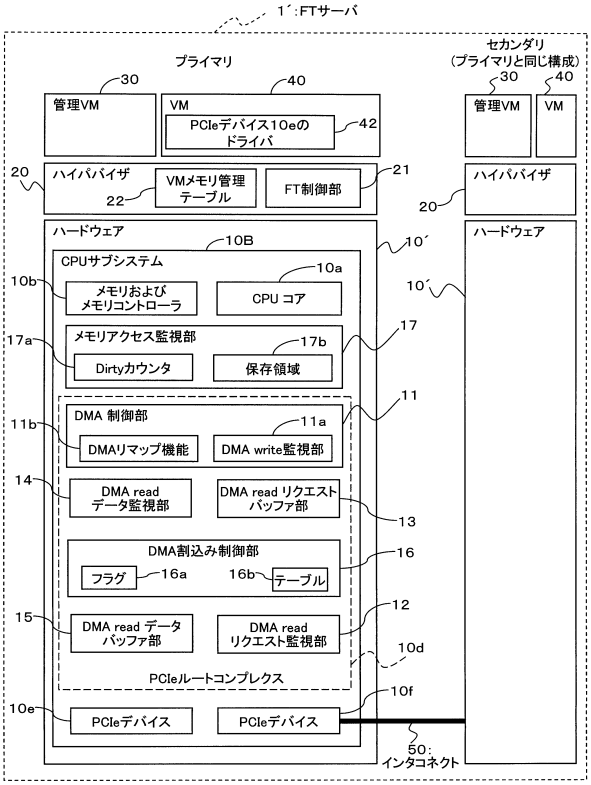
【図７】


スタート
図8から
B
DMAreadリクエスト監視部，
DMAreadデータ監視部，
FT制御部のいずれかからの指示を待機
S121
S122
NO
指示が来たか？
YES
S123
NO
DMAreadリクエスト監視部
からの指示か？
YES
S124
FT制御部により
設定されている割込み
送信先へ割込みを送信
S125
NO
DMAreadデータ監視部
からの指示か？
YES
S126
FT制御部により
設定されている割込み
送信先へ割込みを送信
A
図8へ

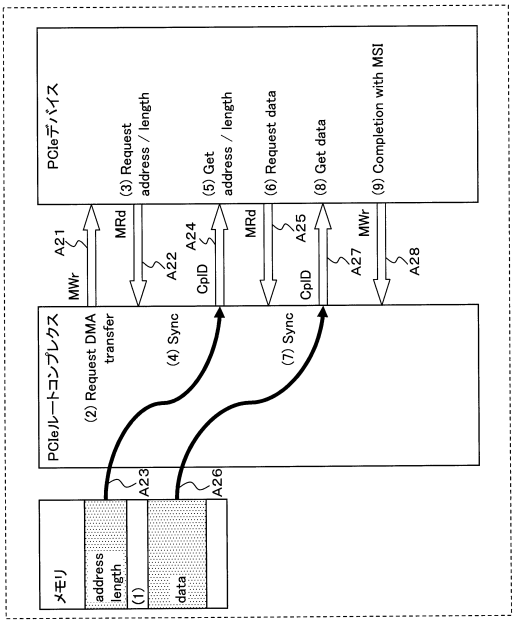
【図８】


図7から
A
S127
NO
FT制御部からの指示か？
YES
S128
NO
DMAreadリクエスト
検出割込みのクリアか？
YES
S129
DMAreadリクエスト
バッファ部に、パケットを
DMA制御部
へ送信するよう指示
S130
NO
DMAreadデータ
検出割込みのクリアか？
YES
S132
FT制御部からの
その他の指示であるので、
適切な処理を実行
S131
DMAreadデータバッファ部に、
パケットを物理デバイスへ
送信するよう指示
B
図7へ

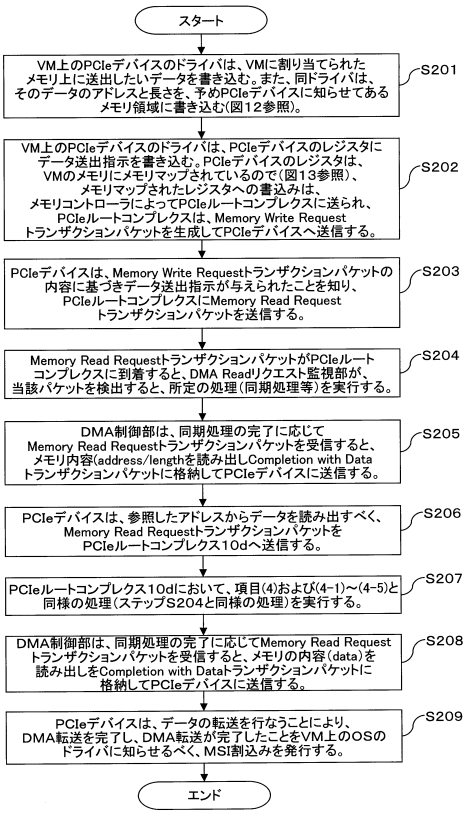
【図９】


1´:FTサーバ
プライマリ
セカンダリ
（プライマリと同じ構成）
30
40
管理VM
VM
PCIeデバイス10eの
ドライバ
42
20
ハイパバイザ
VMメモリ管理
テーブル
22
FT制御部
21
ハードウェア
10B
10´
CPUサブシステム
10a
メモリおよび
メモリコントローラ
10b
CPU コア
メモリアクセス監視部
17a
17b
17
Dirtyカウンタ
保存領域
DMA 制御部
11
11a
11b
DMAリマップ機能
DMA write監視部
DMA read
データ監視部
14
DMA read リクエスト
バッファ部
13
DMA割込み制御部
16
フラグ
16a
16b
テーブル
DMA read データ
バッファ部
15
DMA read
リクエスト監視部
12
10d
PCIeルートコンプレクス
10f
10e
PCIeデバイス
50:
インタコネクト

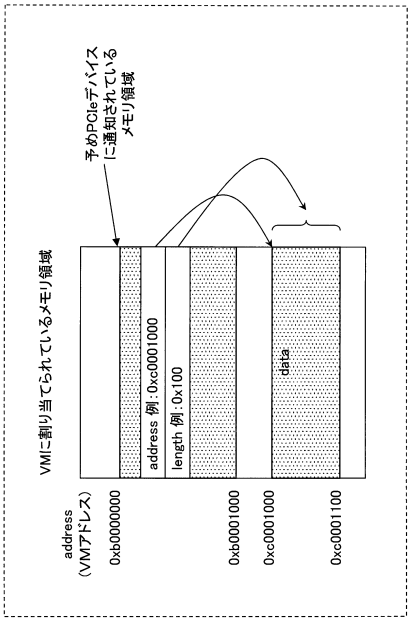
【図１０】


PCIeデバイス
(3) Request address / length
(5) Get address / length
(6) Request data
(8) Get data
(9) Completion with MSI
A21
MWr
MRd
A22
A24
CplD
MRd
A25
CplD
A27
MWr
A28
PCIeルートコンプレクス
(2) Request DMA transfer
(4) Sync
(7) Sync
A23
A26
メモリ
address length
(1)
data

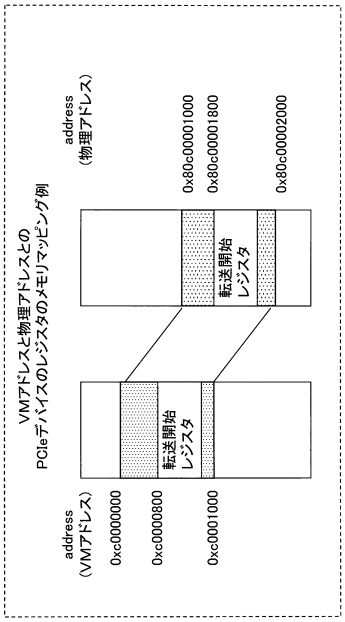
【図１１】



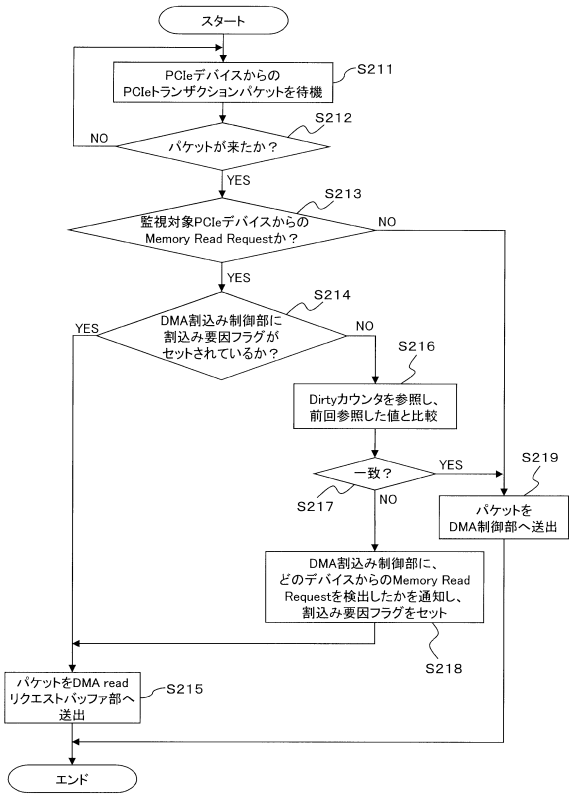
【図１２】



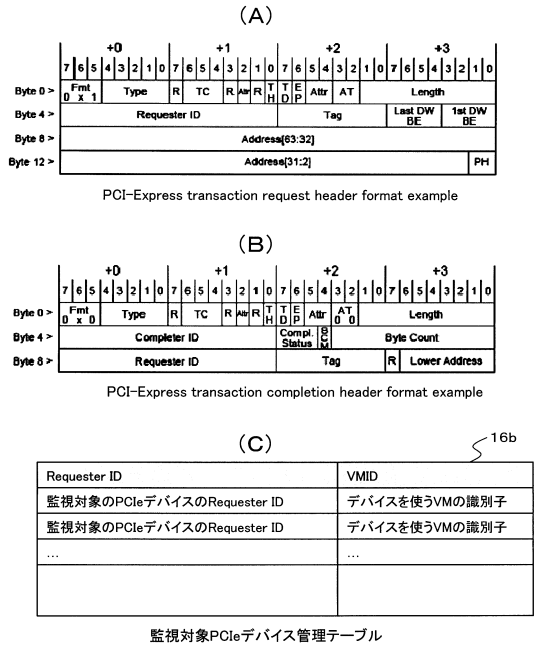
【図１３】



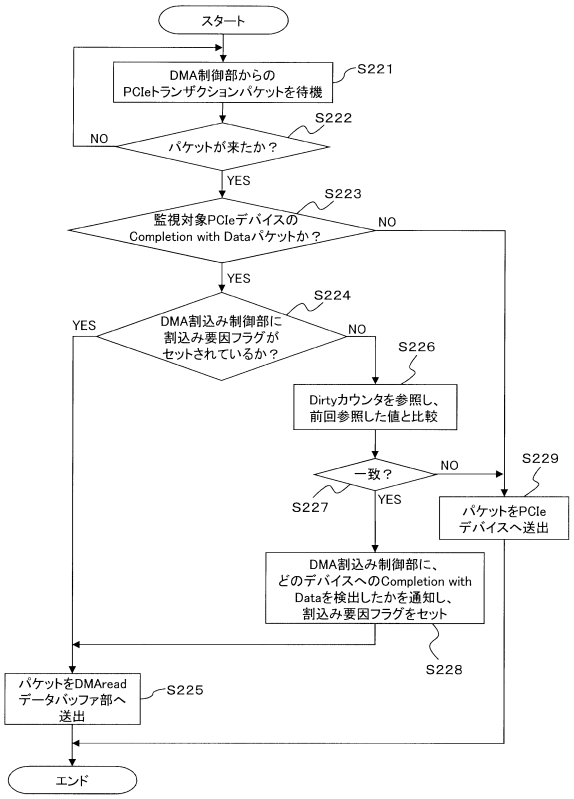
【図１４】



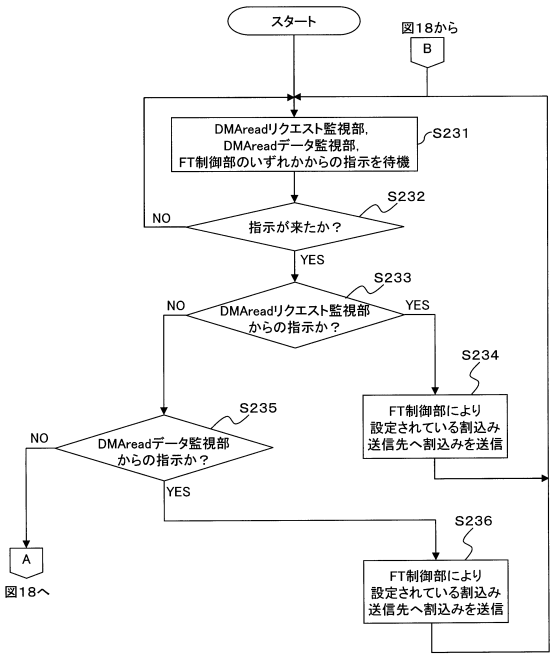
【図１５】



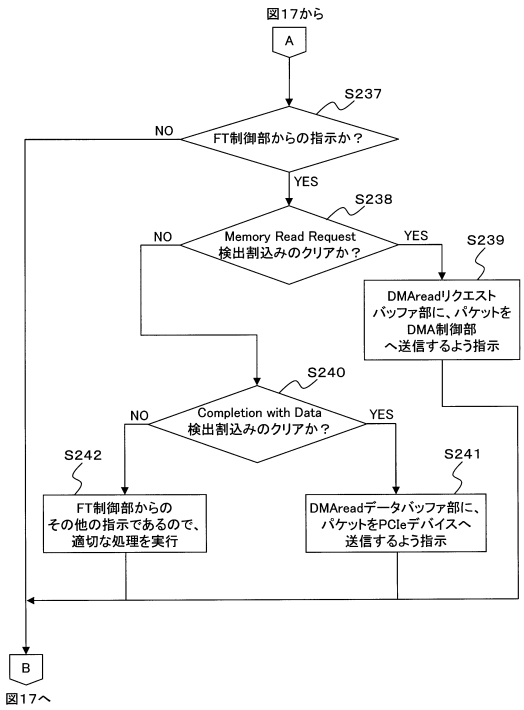
【図１６】



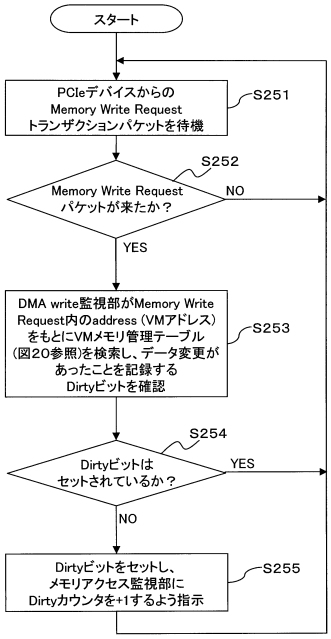
【図１７】



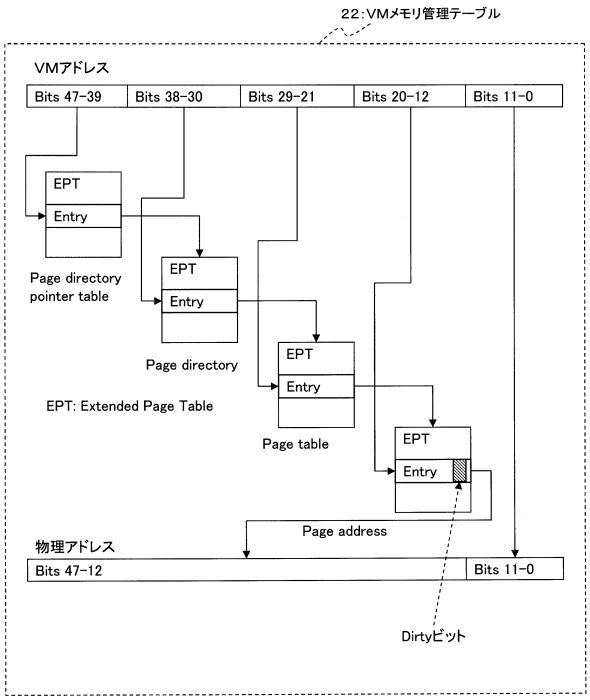
【図１８】



【図１９】


スタート
PCIeデバイスからの
Memory Write Request
トランザクションパケットを待機
S251
S252
Memory Write Request
パケットが来たか？
NO
YES
DMA write監視部がMemory Write
Request内のaddress（VMアドレス）
をもとにVMメモリ管理テーブル
（図20参照）を検索し、データ変更が
あったことを記録する
Dirtyビットを確認
S253
S254
Dirtyビットは
セットされているか？
YES
NO
Dirtyビットをセットし、
メモリアクセス監視部に
Dirtyカウンタを+1するよう指示
S255

【図２０】


22：VMメモリ管理テーブル
VMアドレス
Bits 47-39
Bits 38-30
Bits 29-21
Bits 20-12
Bits 11-0
EPT
Entry
Page directory
pointer table
Page directory
Page table
EPT: Extended Page Table
Page address
物理アドレス
Bits 47-12
Bits 11-0
Dirtyビット

フロントページの続き

(56)参考文献 特開２０１２－０１４２３９（ＪＰ，Ａ）
特表平０７－５０９０８８（ＪＰ，Ａ）
特開平０９－１２８３５４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　９／４６－９／５４
Ｇ０６Ｆ　１１／２０
Ｇ０６Ｆ　１３／１０