(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5272265号
(P5272265)
(45)発行日　平成25年8月28日(2013.8.28)　(24)登録日　平成25年5月24日(2013.5.24)

(51)Int.Cl.　F I
G06F 13/10 (2006.01)　G06F 13/10 330C

請求項の数 13　(全 30 頁)

(21)出願番号　特願2008-250208(P2008-250208)
(22)出願日　平成20年9月29日(2008.9.29)
(65)公開番号　特開2010-79816(P2010-79816A)
(43)公開日　平成22年4月8日(2010.4.8)
審査請求日　平成23年6月16日(2011.6.16)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100114236
弁理士　藤井　正弘
(74)代理人　100075513
弁理士　後藤　政喜
(74)代理人　100120260
弁理士　飯田　雅昭
(72)発明者　馬場　貴成
東京都国分寺市東恋ヶ窪一丁目２８０番地
株式会社日立製作所　中央研究所内
(72)発明者　森木　俊臣
東京都国分寺市東恋ヶ窪一丁目２８０番地
株式会社日立製作所　中央研究所内
最終頁に続く

(54)【発明の名称】ＰＣＩデバイス共有方法

(57)【特許請求の範囲】
【請求項１】
仮想化部を備えて１以上の仮想サーバを提供する複数の物理サーバと、前記複数の物理サーバで共有されるＩ／Ｏデバイスと、前記複数の物理サーバと前記Ｉ／Ｏデバイスを接続するスイッチと、前記スイッチを初期化する管理部とを有するサーバシステムであって、
前記Ｉ／Ｏデバイスが、
仮想的な機能であるＶｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ（ＶＦ）を有し、
前記スイッチが、
前記物理サーバが認識する識別子である第１の識別子と、前記管理部が管理する識別子である第２の識別子と、の間の対応関係を示す識別子対応情報を保持し、
前記物理サーバからパケットを受信した場合に、
送信元の物理サーバを識別するサーバ識別子を含む前記パケットを前記スイッチ内で変換し、
前記パケットを前記Ｉ／Ｏデバイスに送信する際には、前記識別子対応情報に基づいて、前記パケットに含まれる前記第１の識別子を、当該第１の識別子に前記対応付けられた前記第２の識別子に変換し、かつ、前記サーバ識別子を削除した前記パケットを、前記Ｉ／Ｏデバイスに送信する、ことを特徴とするサーバシステム。
【請求項２】
請求項１記載のサーバシステムであって、

前記物理サーバは、前記第１の識別子に基づいて、前記仮想サーバから前記ＶＦへのアクセスを管理し、
前記Ｉ／Ｏデバイスは、前記第２の識別子に基づいて、前記仮想サーバから前記ＶＦへのアクセスを管理する、ことを特徴とするサーバシステム。
【請求項３】
請求項１記載のサーバシステムであって、
前記スイッチは、
更に前記物理サーバが認識する前記ＶＦのＭＭＩＯアドレス領域情報と、前記管理部が管理する前記ＶＦのＭＭＩＯアドレス領域情報との間の対応関係を示すＭＭＩＯアドレス対応情報を保持し、
前記物理サーバからパケットを受信した場合に、更に、前記ＭＭＩＯアドレス対応情報に従って前記パケットの宛先アドレスを変換する処理を行う、ことを特徴とするサーバシステム。
【請求項４】
請求項１記載のサーバシステムであって、
前記Ｉ／Ｏデバイスは、ＰＣＩ ｓｉｇ規定のＳｉｎｇｌｅ Ｒｏｏｔ Ｉ／Ｏ Ｖｉｒｔｕａｌｉｚａｔｉｏｎ ａｎｄ Ｓｈａｒｉｎｇ Ｓｐｅｃｉｆｉｃａｔｉｏｎ（ＳＲ－ＩＯＶ）に準拠、もしくはＳＲ－ＩＯＶの上位互換のＩ／Ｏデバイスであることを特徴とするサーバシステム。
【請求項５】
仮想化部を備えて１以上の仮想サーバを提供する複数の物理サーバと、仮想的な機能であるＶｉｒｔｕａｌ Ｆｕｎｃｔｉｏｎ（ＶＦ）を有して前記複数の物理サーバで共有されるＩ／Ｏデバイスと、を接続するスイッチング装置であって、
前記物理サーバと接続する第１のポートと、
前記Ｉ／Ｏデバイスと接続する第２のポートと、
前記物理サーバが認識する識別子である第１の識別子と、前記スイッチを初期化する管理部が管理する識別子である第２の識別子と、の間の対応関係を管理する識別子管理部と、
前記物理サーバにより送信された前記Ｉ／Ｏデバイス宛のパケットの前記Ｉ／Ｏデバイスへの転送を制御する転送制御部と、を備え、
前記第１のポートが前記物理サーバから前記Ｉ／Ｏデバイス宛のパケットを受信した場合に、
前記転送制御部は、送信元の物理サーバを識別するサーバ識別子を含む前記パケットを前記スイッチ内で変換し、
前記パケットを前記第２のポートから前記Ｉ／Ｏデバイスに送信する際には、前記識別子管理部が管理する前記対応関係に基づいて、前記パケットに含まれる前記第１の識別子を、当該第１の識別子に前記対応付けられた前記第２の識別子に変換し、かつ、前記サーバ識別子を削除した前記パケットを、前記Ｉ／Ｏデバイスに送信する、ことを特徴とするスイッチング装置。
【請求項６】
請求項５記載のスイッチング装置であって、
前記第１の識別子は、前記仮想サーバから前記ＶＦへのアクセスの管理のために前記物理サーバが用いる識別子であり、
前記第２の識別子は、前記仮想サーバから前記ＶＦへのアクセスの管理のために前記Ｉ／Ｏデバイスが用いる識別子である、ことを特徴とするスイッチング装置。
【請求項７】
請求項５記載のスイッチング装置であって、
前記物理サーバが認識する前記ＶＦのＭＭＩＯアドレス領域情報と、前記管理部が管理する前記ＶＦのＭＭＩＯアドレス領域情報と、の間の対応関係を管理するＭＭＩＯアドレス対応関係管理部を、更に有し、

前記第１のポートが前記物理サーバから前記Ｉ／Ｏデバイス宛のパケットを受信した場合に、
前記転送制御部は、更に、前記ＭＭＩＯアドレス対応管理部に従って前記パケットの宛先アドレスを変換する処理を行う、ことを特徴とするスイッチング装置。
【請求項８】
仮想化部を備えて１以上の仮想サーバを提供する複数の物理サーバと、Ｉ／Ｏデバイスと、前記複数の物理サーバと前記Ｉ／Ｏデバイスとを接続するスイッチと、前記スイッチを初期化する管理部と、を有するサーバシステムにおいて、前記Ｉ／Ｏデバイスを前記複数の物理サーバ間で共有するＩ／Ｏデバイス管理方法であって、
前記Ｉ／Ｏデバイスによって、仮想的な機能であるＶｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ（ＶＦ）を複数生成し、
前記スイッチによって、
前記物理サーバが認識する前記ＶＦの識別子である第１の識別子と、前記管理部が管理する前記ＶＦの識別子である第２の識別子と、の間の対応関係を示す識別子対応情報を管理し、
前記スイッチが前記物理サーバからパケットを受信した場合に、送信元の物理サーバを識別するサーバ識別子（＝ＶＨＮ）を含む前記パケットを前記スイッチ内で変換し、
前記パケットを前記Ｉ／Ｏデバイスに送信する際には、前記識別ｘ子対応情報に基づいて、前記パケットに含まれる前記第１の識別子を、当該第１の識別子に前記対応付けられた前記第２の識別子に変換し、かつ、前記サーバ識別子を削除した前記パケットを、前記Ｉ／Ｏデバイスに送信する、ことを特徴とするＩ／Ｏデバイス管理方法。
【請求項９】
請求項８記載のＩ／Ｏデバイス管理方法であって、
前記第１の識別子に基づいて、前記仮想サーバから前記ＶＦへのアクセスに関して前記物理サーバによる管理を行い、
前記第２の識別子に基づいて、前記仮想サーバから前記ＶＦへのアクセスに関して前記Ｉ／Ｏデバイスによる管理を行う、ことを特徴とするＩ／Ｏデバイス管理方法。
【請求項１０】
請求項８記載のＩ／Ｏデバイス管理方法であって、
前記スイッチによって、
更に前記物理サーバが認識する前記ＶＦのＭＭＩＯアドレス領域情報と、前記管理部が管理する前記ＶＦのＭＭＩＯアドレス領域情報と、の間の対応関係を示すＭＭＩＯアドレス対応情報を管理し、
前記物理サーバからパケットを受信した場合に、更に、前記ＭＭＩＯアドレス対応情報に従って前記パケットの宛先アドレスを変換する処理を行う、ことを特徴とした、Ｉ／Ｏデバイス管理方法。
【請求項１１】
請求項３記載のサーバシステムであって、
前記物理サーバが認識する前記ＶＦのＭＭＩＯアドレス領域情報は、前記仮想サーバがアクセスする前記ＶＦのＭＭＩＯ空間のアドレスのオフセットであって、
前記管理部が管理する前記ＶＦのＭＭＩＯアドレス領域情報は、前記管理部がアクセスする前記ＶＦのＭＭＩＯ空間のアドレスのオフセットであって、
前記サーバ識別子は、送信元の物理サーバを識別する仮想階層ナンバーであることを特徴とするサーバシステム。
【請求項１２】
請求項７記載のスイッチング装置であって、
前記物理サーバが認識する前記ＶＦのＭＭＩＯアドレス領域情報は、前記仮想サーバがアクセスする前記ＶＦのＭＭＩＯ空間のアドレスのオフセットであって、
前記管理部が管理する前記ＶＦのＭＭＩＯアドレス領域情報は、前記管理部がアクセスする前記ＶＦのＭＭＩＯ空間のアドレスのオフセットであって、

前記サーバ識別子は、送信元の物理サーバを識別する仮想階層ナンバーであることを特徴とするスイッチング装置。

【請求項１３】

請求項１０記載のＩ／Ｏデバイス管理方法であって、

前記物理サーバが認識する前記ＶＦのＭＭＩＯアドレス領域情報は、前記仮想サーバがアクセスする前記ＶＦのＭＭＩＯ空間のアドレスのオフセットであって、

前記管理部が管理する前記ＶＦのＭＭＩＯアドレス領域情報は、前記管理部がアクセスする前記ＶＦのＭＭＩＯ空間のアドレスのオフセットであって、

前記サーバ識別子は、送信元の物理サーバを識別する仮想階層ナンバーであることを特徴とするＩ／Ｏデバイス管理方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、複数の計算機を備えたブレードサーバに関し、特に、ひとつのＩ／Ｏデバイスを複数の計算機で共有する技術に関する。

【背景技術】

【０００２】

情報セキュリティやコンプライアンスへの意識の高まりから、サーバサイドでのウィルスチェックやメールフィルタなど、企業の情報システムに求められる処理要求が増大している。これらの処理要求の増大に対し、従来は処理内容ごとに個別にサーバを導入して対応してきた。しかし、サーバ台数の増大は運用コストの増大を招き、企業のIT予算を圧迫して問題化している。

【０００３】

この問題に対し、複数のサーバで実行されていた処理を１台の高性能サーバに集約して、情報システムを構成するサーバ台数を削減するサーバ統合が注目されている。サーバ統合ではサーバの台数に比例して継続して発生する消費電力やスペース(占有床面積)、ハードウェア障害時のメンテナンスコスト等を削減できる。

【０００４】

サーバ統合を実現する手段として、ＣＰＵを高密度に集積したブレードサーバが台頭している（例えば、特許文献１）。典型的なブレードサーバは、ＣＰＵ（プロセッサ）、メモリ及びＮＩＣ（Network Interface Card）を搭載した複数のブレードと、ネットワークスイッチ、および拡張用Ｉ／Ｏスロットを搭載したＩ／Ｏドロワーが１つの筐体に格納された構成をとる。ブレードサーバを用いると、ネットワーク経由で処理を行うサーバに対して効果的にサーバ統合を実現できる。

【０００５】

一方、ブレードサーバはＩ／Ｏドロワーの拡張用Ｉ／Ｏスロットは、ブレードとＩ／Ｏスロットとの対応が固定的であった。このため、ＮＩＣ以外のＩ／Ｏカードを稀にしか使用しない、もしくは全く使用しないブレードに対してもＩ／Ｏスロットが予約されており、スペースに無駄が生じるという問題があった。また、ブレードの用途によっては多数のＩ／Ｏカードを必要とする場合（例；複数ブレード間でのＨｏｔ－Ｓｔａｎｄｂｙ構成等）に、Ｉ／Ｏドロワー内のＩ／Ｏスロットの数を超えてＩ／Ｏスロットを割り当てることができない、という問題があった。

【０００６】

ブレードサーバのように多数のブレードで限られた数のＩ／Ｏスロット（またはＩ／Ｏデバイス）を利用する技術として、ひとつのＩ／Ｏデバイスを複数のホストで利用する技術が提案されている。

【０００７】

このような技術としては、ＰＣＩ－Ｓｉｇで標準化が行われているシングルルートＩ／Ｏ仮想化技術（ＳＲ－ＩＯＶ：Single Root IO Virtualization）やマルチルートＩ／Ｏ仮想化技術（ＭＲ－ＩＯＶ：Multi Root IO Virtualization）が知られている（例えば、

特許文献２、非特許文献１等）。
【０００８】
この他、任意のサーバ間でストレージ装置を共有する技術としては、ｉＳＣＳＩが広く知られている。ｉＳＣＳＩは、サーバとストレージ装置のアクセスに使用されるＳＣＳＩコマンドを、ネットワーク通信のプロトコルであるＴＣＰ／ＩＰパケットにカプセル化する技術である。これにより任意のサーバ間でｉＳＣＳＩ対応ターゲットデバイスを共有できる。
【特許文献１】特開特開２００２－３２１５３号
【特許文献２】米国特許第７０５８７３８号
【非特許文献１】「Single-Root I/O Virtualization and Sharing Specification, Revision 1.0, Section 1 - Architectural」、2008年発行、著者PCI-SIG、第15～34頁
【発明の開示】
【発明が解決しようとする課題】
【０００９】
上記従来例のシングルルートＩ／Ｏ仮想化技術（以下、ＳＲ－ＩＯＶ）では、１つのＩ／Ｏデバイスの物理機能（ＰＦ；Ｐｈｙｓｉｃａｌ　Ｆｕｎｃｔｉｏｎ）が複数の仮想機能（ＶＦ；Ｖｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ）を提供することができる。同一ブレード内の複数のＯＳインスタンス間で、複数の仮想機能をそれぞれ占有することでひとつのＳＲ－ＩＯＶ対応のＩ／Ｏデバイスを共有することができる。
【００１０】
一方、上記従来例のマルチルートＩ／Ｏ仮想化技術（以下、ＭＲ－ＩＯＶ）では、１つのＩ／Ｏデバイスに複数の物理機能（ＰＦ；Ｐｈｙｓｉｃａｌ　Ｆｕｎｃｔｉｏｎ）を搭載し、各物理機能上で複数の仮想機能（ＶＦ；Ｖｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ）を提供することができる。ＭＲ－ＩＯＶ対応のＩ／Ｏデバイスを利用し、仮想化ソフトウェアを導入することで、異なる物理サーバ上で動作する仮想計算機（ＶＭ；Ｖｉｔｕａｌ　Ｍａｃｈｉｎｅ）間で１つのＩ／Ｏデバイスを共有することができる。すなわち、ＭＲ－ＩＯＶでは、特定のブレード（物理計算機）に対して１つのＰＦを占有させ、各ＶＭはＰＦが提供する複数のＶＦを占有する。
【００１１】
ここで、ＳＲ－ＩＯＶに対応するＩ／Ｏデバイスでは、一つの物理機能を備えればよいので、ＭＲ－ＩＯＶに対応するＩ／Ｏデバイスに低コストで調達することができる。
【００１２】
上記従来例のブレードサーバにおいて複数のブレードでＩ／Ｏデバイスを共有する場合、上記ＳＲ－ＩＯＶに対応するＩ／Ｏデバイスを用いれば、Ｉ／Ｏデバイスの調達コストを抑制できるものの、ひとつのブレードに対してひとつのＩ／Ｏデバイスを割り当てることになり、複数のブレード間でひとつのＩ／Ｏデバイスを共有できない、という問題があった。
【００１３】
一方、従来例のブレードサーバにおいて複数のブレードでＩ／Ｏデバイスを共有する場合、上記ＭＲ－ＩＯＶに対応するＩ／Ｏデバイスを用いれば、複数のブレード間でひとつのＩ／Ｏデバイスを共有することができる。しかしながら、ＭＲ－ＩＯＶに対応するＩ／ＯデバイスはＳＲ－ＩＯＶに対応するＩ／Ｏデバイスに対して調達コストが高いため導入コストが増大する、という問題がある。
【００１４】
なお、上記ｉＳＣＳＩをブレードサーバで利用した場合、ストレージ装置の共有は可能であるが、Ｉ／ＯデバイスがＮＩＣ等の場合は利用することができない。
【００１５】
そこで本発明は、上記問題点に鑑みてなされたもので、複数のブレードを備えたブレードサーバにおいて、複数のブレード間で、Ｉ／Ｏドロワーに装着されるＩ／Ｏデバイスを安価に共有することを目的とする。

【課題を解決するための手段】
【００１６】
本発明は、仮想化部を備えて１以上の仮想サーバを提供する複数の物理サーバと、前記複数の物理サーバで共有されるＩ／Ｏデバイスと、前記複数の物理サーバと前記Ｉ／Ｏデバイスを接続するスイッチと、前記スイッチを初期化する管理部とを有するサーバシステムであって、前記Ｉ／Ｏデバイスが、仮想的な機能であるＶｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ（ＶＦ）を有し、前記スイッチが、前記仮想サーバが認識する識別子である第１の識別子と、前記管理部が管理する識別子である第２の識別子と、の間の対応関係を示す識別子対応情報を保持し、前記仮想サーバからパケットを受信した場合に、送信元の物理サーバを識別するサーバ識別子を含む前記パケットを前記スイッチ内で変換し、前記パケットを前記Ｉ／Ｏデバイスに送信する際には、前記識別子対応情報に基づいて、前記パケットに含まれる前記第１の識別子を、当該第１の識別子に前記対応付けられた前記第２の識別子に変換し、かつ、前記サーバ識別子を削除した前記パケットを、前記Ｉ／Ｏデバイスに送信する、ことを特徴とするサーバシステム。
【発明の効果】
【００１７】
したがって、本発明は、ひとつのＩ／Ｏデバイス、特にSR-IOV対応のＩ／Ｏデバイスを複数のサーバで共有することが可能となる。
【発明を実施するための最良の形態】
【００１８】
以下、本発明の一実施形態を添付図面に基づいて説明する。
【００１９】
図１は、第１の実施形態を示し、本発明を適用したブレードサーバ（複合型計算機システム）のブロック図を示す。図１において、ブレードサーバ１は、物理計算機として機能するｎ台のブレード１０－１～１０－ｎと、シングルルートＩ／Ｏ仮想化技術（以下、ＳＲ－ＩＯＶ）に対応したＩ／Ｏデバイス５０と、マルチルートＩ／Ｏ仮想化技術（以下、ＭＲ－ＩＯＶ）でトランザクション（パケット）を処理するブレード１０－１～１０－ｎとＩ／Ｏデバイス５０を接続するＰＣＩ－ｅスイッチ４０と、Ｉ／Ｏデバイス５０のブレード１０－１～１０－ｎへの割当てを管理するＰＣＩ管理サーバ（ＰＣＩ管理計算機）２０と、ユーザや管理者がブレードサーバ１を制御するための管理用端末３０と、ブレードサーバ１の各部に電力を供給する電源供給装置６０から構成されている。なお、ＰＣＩ－ｅスイッチ４０、Ｉ／Ｏデバイス５０は、上記ＰＣＩ－ＳＩＧ（http://www.pcisig.com/）が策定したＰＣＩ－ｅｘｐｒｅｓｓの規格に準拠したインターフェースで接続される。なお、図１においては、ＰＣＩ－ｅスイッチ４０が１つ、Ｉ／Ｏデバイス５０が１つの例を示したが、複数のＰＣＩ－ｅスイッチ４０と複数のＩ／Ｏデバイス５０を備えることができる。
【００２０】
ブレード１０－１は１以上のＣＰＵ（プロセッサ）１１－１，１１－２と、１以上のメモリ１２－１と、１以上のチップセット１３－１と、を含むハードウェアで構成される。なお、ブレード１０－ｎとＰＣＩ管理サーバ２０もブレード１０－１と同一のハードウェアで構成される。
【００２１】
また、ブレード１０－１～１０－ｎとＰＣＩ管理サーバ２０の間はネットワーク７０を介して接続される。ネットワークとしては、ＬＡＮ（Local Area Network）やＩ^{2}Ｃ（Inter-Integrated Circuit）を用いることができる。各ブレード１０－１～１０－ｎにはＢＭＣ（Baseboard Management Controller）を搭載してもよく、ＢＭＣによって各ブレード１０－１～１０－ｎの構成情報（コンフィギュレーション）や電源状態を収集し、ＰＣＩ管理サーバ２０に通知することができる。
【００２２】
なお、各ブレード１０－１～１０－ｎ及びＰＣＩ管理サーバ２０のチップセット１３－

１～１３－ｎは、それぞれＰＣＩ－ｅｘｐｒｅｓｓに準拠したＩ／Ｏポート１３１－１～１３１－ｎと、イーサネット（登録商標）等のＬＡＮの規格に準拠したネットワークインターフェース１３２－１～１３２－ｎを備え、チップセット１３－１～１３－ｎのネットワークインターフェース１３２－１～１３２－ｎがネットワーク７０に接続され、チップセット１３－１～１３－ｎのＩ／Ｏポート１３１－１～１３１－ｎがＰＣＩ－ｅスイッチ４０に接続される。

【００２３】

ＰＣＩ－ｅスイッチ４０は、ブレード１０－１～１０－ｎ及びＰＣＩ管理サーバ２０のチップセット１３と接続されるアップストリームポート４１－０～４１－ｎと、Ｉ／Ｏデバイス５０が接続されるダウンストリームポート４２－１～４２－ｎを備える。

【００２４】

図１の例では、ブレード１０－１のチップセット１３－１のＩ／Ｏポート１３１－１がＰＣＩ－ｅスイッチ４０のアップストリームポート４１－１に接続され、ブレード１０－ｎのチップセット１３－ｎのＩ／Ｏポート１３１－ｎがＰＣＩ－ｅスイッチ４０のアップストリームポート４１－ｎに接続され、ＰＣＩ管理サーバ２０のチップセット１３－ＭのＩ／Ｏポート１３１－ＭがＰＣＩ－ｅスイッチ４０のアップストリームポート４１－０に接続され、ダウンストリームポート４２－１にＩ／Ｏデバイス５０が接続された例を示す。

【００２５】

なお、ブレード１０－１～１０－ｎとＰＣＩ－ｅスイッチ４０の接続及びＩ／Ｏデバイス５０とＰＣＩ－ｅスイッチ４０の接続は、ブレードサーバ１のバックプレーなどを利用することができる。

【００２６】

ＰＣＩ－ｅスイッチ４０は、アップストリームポート４１－１～４１－ｎがＭＲ－ＩＯＶに対応しており、ダウンストリームポート４２－１～４２－ｎがＳＲ－ＩＯＶに対応し、アップストリームポート４１－１～４１－ｎはブレード１０－１～１０－ｎとの間でマルチルートＩ／Ｏ仮想化技術のトランザクション（パケット）を送受信し、ダウンストリームポート４２－１～４２－ｎとＩ／Ｏデバイス５０の間ではシングルルートＩ／Ｏ仮想化技術のパケットを送受信する。このため、ＰＣＩ－ｅスイッチ４０は、マルチルートＩ／Ｏ仮想化技術のパケットとシングルルートＩ／Ｏ仮想化技術のパケットを相互に変換する処理を行う。

【００２７】

ＰＣＩ管理サーバ２０には、入出力装置（図示省略）を備えた管理端末３０が接続され、管理者などの操作により後述するＰＣＩマネージャ２０２に指令を行うことができる。

【００２８】

図２は、ブレードサーバ１の機能要素を示すブロック図である。ブレード１０－１～１０－ｎ上ではソフトウェア構成要素である仮想マシンモニタ（ＶＭＭ；Virtual Machine Monitor）１００－１～１００－ｎが稼動し、各仮想マシンモニタ１００－１～１００－ｎ上では複数の仮想計算機１０１－０～１０１－ｋ－１が提供される。ブレード１０－１では、仮想マシンモニタ１００－１上に２つの仮想計算機１０１－０、１０１－１が生成され、各各仮想計算機１０１－０、１０１－１ではＯＳ１０２－０とＯＳ－１０２－１がそれぞれ実行される。仮想計算機１０１－０、１０１－１のＯＳ１０２－０とＯＳ－１０２－１にはＩ／Ｏデバイス５０の仮想機能（ＶＦ；Ｖｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ）を利用するためのＶＦドライバ１０３がそれぞれロードされている。他のブレード１０－ｎもブレード１０－１と同様に構成され、それぞれ複数の仮想計算機１０１－ｋ－１でＯＳ１０２を実行し、ＶＦドライバ１０３によってＩ／Ｏデバイス５０の仮想機能を利用することができる。また、各仮想計算機１０１－１～１０１－ｋ－１のＯＳ１０２－０～１０２－Ｋ－１では、任意のアプリケーションまたはサービスが実行される。

【００２９】

一方、ＰＣＩ管理サーバ２０では、ＯＳ２０１上で各ブレード１０－１～１０－ｎとＰ

ＣＩ－ｅスイッチ４０及びＩ／Ｏデバイス５０を管理するＰＣＩマネージャ２０２が動作する。ＰＣＩ管理サーバ２０のＯＳ２０１にはＩ／Ｏデバイス５０の物理機能（ＰＦ；Ｐｈｙｓｉｃａｌ　Ｆｕｎｃｔｉｏｎ）を利用するためのＰＦドライバ２０３がロードされている。

【００３０】

ブレード１０－１～１０－ｎの各仮想計算機１０１－１～１０１－ｋ－１と、ＰＣＩ管理サーバ２０からアクセスされるＩ／Ｏデバイス５０は、ひとつの物理機能（ＰＦ）５０１と複数の仮想機能（ＶＦ）５０２を提供するＳＲ－ＩＯＶに準拠したＰＣＩ－ｅｘｐｒｅｓｓのＩ／Ｏデバイスである。物理機能５０１と仮想機能５０２は、Ｉ／Ｏデバイス５０の制御部５１０が提供する機能であり、仮想機能５０２の数等はＰＣＩ管理サーバ２０からの要求によって決定される。なお、図２では、Ｉ／Ｏデバイス５０が仮想機能５０２としてＶＦ１～ＶＦｋまでのｋ個の仮想機能５０２を提供する例を示す。

【００３１】

ここで、ブレード１０－１～１０－ｎチップセット１３－１～１３－ｎと各仮想計算機１０１－１～１０１－ｋ－１のＶＦドライバ１０３は、ＭＲ－ＩＯＶに対応したＰＣＩ－ｅｘｐｒｅｓｓのパケットを送受信する。一方、Ｉ／Ｏデバイス５０はＳＲ－ＩＯＶに対応したＰＣＩ－ｅｘｐｒｅｓｓのパケットを送受信する。

【００３２】

本発明の特徴部であるＰＣＩ－ｅスイッチ４０は、後述するように、ＭＲ－ＩＯＶのアップストリームポート４１－１～４１－ｎとＳＲ－ＩＯＶのダウンストリームポート４２－１～４２－ｎとの間で、パケットの変換を行って、複数のブレード１０－１～１０－ｎでＳＲ－ＩＯＶのＩ／Ｏデバイス５０を共有する。

【００３３】

各ブレード１０－１～１０－ｎ及びＰＣＩ管理サーバ２０を構成するチップセット１３－１～ｎ、１３－Ｍは、ＰＣＩ－Ｅｘｐｒｅｓｓのプロトコル階層としてルートコンプレックス（ＲＣ；Root Complex）を含み、ルートコンプレックスはエンドポイントとしてのＩ／Ｏデバイス５０までのローカルＰＣＩツリーを管理する。

【００３４】

ＰＣＩ管理サーバ２０のＰＣＩマネージャ２０２は、後述するように、ブレード１０－１～１０－ｎに割当てるＰＣＩツリーの初期化と、ＰＣＩ－ｅスイッチ４０からＩ／Ｏデバイス５０のトポロジーと、ブレード１０－１～１０－ｎへ割当てるＰＣＩツリーの対応関係を管理し、各ブレード１０－１～１０－ｎとＰＣＩ－ｅスイッチ４０間のＭＲ－ＩＯＶの設定を行う。

【００３５】

また、ＰＣＩマネージャ２０２は、ブレード１０－１～１０－ｎの管理部としても機能し、各ブレード１０－１～１０－ｎの仮想マシンモニタ１００－１～１００－ｎが生成する仮想計算機１０１－１～１０１－ｋ－１や起動するＯＳ１０２－０～１０２－ｋ－１を管理する。

【００３６】

図３は、ＰＣＩ－ｅスイッチ４０の構成を示すブロック図である。ＰＣＩ－ｅスイッチ４０は、各ブレード１０－１～１０－ｎに接続されてＭＲ－ＩＯＶのパケットを送受信するアップストリームポート４１－１～４１－ｎがマルチルートスイッチ（以下、ＭＲＳ）論理４１０に接続され、Ｉ／Ｏデバイス５０に接続されるダウンストリームポート４２－１～４２－ｎは、ＳＲ－ＩＯＶのパケットとＭＲ－ＩＯＶのパケットを相互に変換するマルチルート－シングルルート（以下、ＭＲ－ＳＲ）変換論理４３０に接続される。なお、ＭＲ－ＳＲ変換論理４３０は、ダウンストリームポート４２－１に対応するものだけを図示したが、実際には、各ダウンストリームポート４２－１～４２－ｎ毎にＭＲ－ＳＲ変換論理４３０が設けられる。

【００３７】

ＭＲＳ（マルチルートスイッチ）論理４１０は、アップストリームポート４１－１～４

１－ｎに接続されるポート４１１－０～４１１－ｎと、ポート４１１－０～４１１－ｎにそれぞれ接続されたアップストリームポートブリッジ４１２と、アップストリームポートブリッジ４１２に接続されたダウンストリームポートブリッジ４１３と、ダウンストリームポートブリッジ４１２に接続されたマルチルートポート４１４と、ポート４１１－０～４１１－ｎとマルチルートポート４１４と、ポート４１１－０～４１１－ｎとマルチルートポート４１４の接続関係等のＭＲＳ（マルチルートスイッチ）構成情報（コンフィギュレーション）４１５とを備える。ＭＲＳ構成情報４１５は、ポート４１１－０～４１１－ｎとマルチルートポート４１４のルーティングを管理する。このＭＲＳ構成情報４１５は、コンフィギュレーション空間アクセスパス４２１によりアップストリームポートブリッジ４１２に接続されてアップストリームポート４１－１～４１－ｎからアクセス可能となっており、例えば、アップストリームポート４１－０に接続されたＰＣＩ管理サーバ２０からアクセスすることができる。さらに、コンフィギュレーション空間アクセスパス４２１は、ＭＲ－ＳＲ変換論理４３０にも接続され、ＰＣＩ管理サーバ２０はＭＲＳ論理４１０からコンフィギュレーション空間アクセスパス４２１を介してＭＲ－ＳＲ変換論理４３０にもアクセス可能となっている。

【００３８】

ＭＲＳ論理４１０のマルチルートポート４１４は、内部マルチルートリンク４２０を介してＭＲ－ＳＲ変換論理４３０に接続される。

【００３９】

ＭＲ－ＳＲ変換論理４３０は、内部マルチルートリンク４２０から受信したＭＲ－ＩＯＶのパケットをＳＲ－ＩＯＶのパケットに変換し、ダウンストリームポート４２－１からＩ／Ｏデバイス５０へ送信するユニット４３１～４３４と、ダウンストリームポート４２－１から受信したＩ／Ｏデバイス５０からのＳＲ－ＩＯＶのパケットをＭＲ－ＩＯＶのパケットに変換して、内部マルチルートリンク４２０に送信するユニット４３５～４３８と、ＭＲ－ＩＯＶのパケットとＳＲ－ＩＯＶのパケットの変換を行うための情報を格納するＴＬＰ（Transaction Layer Packet）変換情報４４０を主体に構成される。

【００４０】

ここで、本発明のパケットの構成について説明する。図１４は、マルチルートＴＬＰ及びＰＣＩｅ（ＰＣＩ－ｅｘｐｒｅｓｓ）ベースＴＬＰのフォーマットを示す説明図である。

【００４１】

マルチルートＴＬＰ（図中ＭＲ　ＴＬＰ）１３００は、ＰＣＩ－ｅｘｐｒｅｓｓのパケットであるＰＣＩｅベースＴＬＰ１２００のヘッダーの前に、マルチルートＴＬＰプリフィックスヘッダー１３１０を付加したものである。

【００４２】

ＰＣＩｅベースＴＬＰ１２００は、スタートフレーム（図中ＳＴＰ）と、シーケンス番号と、ＴＬＰヘッダーと、ＥＣＲＣ(End to End Cyclic Redundancy Check)と、ＬＣＲＣ（Link Cyclic Redundancy Check）と、エンドフレームから構成される。

【００４３】

マルチルートＴＬＰ１３００は、ＰＣＩｅベースＴＬＰ１２０のシーケンス番号とＴＬＰヘッダーの間に、マルチルートＴＬＰプリフィックスヘッダ１３１０を挿入したものである。

【００４４】

マルチルートＴＬＰプリフィックスヘッダ１３１０は、ＭＲ－ＩＯＶにおいてＰＣＩｅパケットの発行元のブレード１０－１～１０－ｎを特定するための仮想階層ナンバー（ＶＨｎ；Virtual Hierarchy Number）１３１１を含む。なお、本実施例における仮想階層ナンバー１３１１はＰＣＩｅスイッチ４０の内部で付与される識別番号であり、ＭＲＳ構成情報４１５に格納されている。

【００４５】

本発明のブレードサーバ１では、仮想計算機１０１－１～１０１－ｋ－１、仮想マシン

モニタ１００－１～１００－ｎ、チップセット１３－１～１３－ｎ及びアップストリームポート４１－１～４１－ｎ間では、ＭＲ－ＩＯＶのパケットであるマルチルートＴＬＰ１３００で送受信を行い、ダウンストリームポート４２－１とＩ／Ｏデバイス５０の間では、マルチルートＴＬＰ１３００からマルチルートＴＬＰプリフィックスヘッダ１３１０を削除して、ＰＣＩｅベースＴＬＰ１２００で送受信を行う。

【００４６】

ここで、本発明のブレードサーバ１では、ブレード１０－１～１０－ｎからＩ／Ｏデバイス５０への下り方向（Outbound）のパケットではマルチルートＴＬＰ１３００は、ＰＣＩ－ｅスイッチ４０のＭＲ－ＳＲ変換論理４３０が、マルチルートＴＬＰプリフィックスヘッダ１３１０を削除して、送信元をＰＣＩ管理サーバ２０としたＰＣＩｅベースＴＬＰ１２００に変換してＩ／Ｏデバイス５０に送信する。

【００４７】

逆に、Ｉ／Ｏデバイス５０からブレード１０－１～１０－ｎへの上り方向（Inbound）のパケットでは、ＰＣＩ－ｅスイッチ４０のＭＲ－ＳＲ変換論理４３０が、Ｉ／Ｏデバイス５０から受信したＰＣＩｅベースＴＬＰ１２００に、下り方向のパケットに付与されていた仮想階層ナンバー１３１１を含むマルチルートＴＬＰプリフィックスヘッダ１３１０を付加してブレード１０－１～１０－ｎに送信する。

【００４８】

以上のような構成により、ＰＣＩ－ｅスイッチ４０のブレード１０－１～１０－ｎ側ではＭＲ－ＩＯＶのパケットで通信を行い、ＰＣＩ－ｅスイッチ４０のＩ／Ｏデバイス５０側ではＳＲ－ＩＯＶのパケットで通信を行うことで、複数のブレード１０－１～１０－ｎでひとつのＳＲ－ＩＯＶに準拠したＩ／Ｏデバイス５０を共有できる。

【００４９】

また、ＰＣＩ－ｅスイッチ４０は、発行元のブレード１０－１～１０－ｎをＩ／Ｏデバイス５０に対して隠蔽するのに加え、各ブレード１０－１～１０－ｎ間で異なるＭＭＩＯ（Memory-mapped I/O））空間を吸収する。

【００５０】

このため、ＰＣＩ管理サーバ２０は、仮想マシンモニタ１００－１～１００－ｎの起動時に、各仮想計算機１０１－１～１０１－ｋ－１のＭＭＩＯ空間のアドレスと、Ｉ／ＯデバイスＩ／Ｏデバイス５０へのＰＣＩｅベースＴＬＰ１２００の発行元となるＰＣＩ管理サーバ２０のＭＭＩＯ空間の差分（オフセット）を、ＰＣＩ－ｅスイッチ４０のＴＬＰ変換情報４４０に格納しておき、上り方向のマルチルートＴＬＰ１３００の宛先アドレスを、実際の宛先となる仮想マシンモニタ１００－１～１００－ｎのＭＭＩＯ空間に変換する。

【００５１】

以下に、ＰＣＩ－ｅスイッチ４０が行う仮想階層ナンバー１３１１の削除及び付加と宛先アドレスの変換について詳述する。

【００５２】

図４は、ブレード１０－１（Ｂｌａｄｅ＃１）～１０－ｎ（Ｂｌａｄｅ＃ｎ）及びＰＣＩ管理サーバ２０（ＰＣＩ－Ｍ）が管理するメモリ１２－１～ｎ、１２－Ｍ上に設定したＭＭＩＯ空間を示す。例えば、図示の例では、ブレード１０－１に２つの仮想機能５０２＝ＶＦ１、ＶＦ２を割り当て、ブレード１０－ｎにも２つの仮想機能５０２＝ＶＦｋ－１、ＶＦ２ｋ割り当てた例を示す。

【００５３】

ブレード１０－１～１０－ｎの仮想マシンモニタ１００－１～ｎはローカルＰＣＩツリーの後にＩ／Ｏデバイス５０の仮想機能５０２（ＶＦ）へアクセスするためのＭＭＩＯ空間を設定する。各ブレードのＭＭＩＯ空間は、搭載しているメモリ１２－１～ｎの容量等の違いにより、アドレスは相違する。各ブレードのＭＭＩＯ空間は、各ブレード１０－１～１０－ｎが使用する仮想機能５０２（ＶＦ）毎にＰＣＩ管理サーバ２０が設定するメモリ１２－Ｍ上のＭＭＩＯ空間に割り当てられる。

【００５４】

各ブレード１０－１～１０－ｎが使用するＭＭＩＯ空間のアドレスの違いは、ＰＣＩマネージャ２０２が設定する宛先アドレス修飾情報４４０１の「Ｏｆｆｓｅｔ」にその差分が設定されることで、ＰＣＩ－ｅスイッチ４０は、Ｉ／Ｏデバイス５０とブレード１０－１～１０－ｎ間のパケットの宛先アドレスを変更することができる。

【００５５】

つまり、ブレード１０－１～１０－ｎからＩ／Ｏデバイス５０に対する読み込み要求等では、各ブレードが送信するパケットの発行元アドレスは、ＰＣＩ－ｅスイッチ４０が、宛先アドレス修飾情報４４０１の「Ｏｆｆｓｅｔ」によって、ＰＣＩ管理サーバ２０のＭＭＩＯ空間が発行元アドレスのパケットに書き換えてＩ／Ｏデバイス５０に送信する。

【００５６】

逆に、Ｉ／Ｏデバイス５０からブレード１０－１～１０－ｎに対する読み込み要求等への応答では、Ｉ／Ｏデバイス５０が送信するパケットの宛先アドレスは、ＰＣＩ管理サーバ２０のＭＭＩＯ空間となっているので、ＰＣＩ－ｅスイッチ４０は宛先アドレス修飾情報４４０１の「Ｏｆｆｓｅｔ」によって、ＰＣＩ管理サーバ２０のＭＭＩＯ空間をブレード１０－１～１０－ｎ毎のＭＭＩＯ空間に書き換えて返信することができる。

【００５７】

以上の処理により、ＰＣＩ－ｅスイッチ４０は、ＭＭＩＯ空間のアドレスを差し替えることで、複数のブレード１０－１～１０－ｎでＳＲ－ＩＯＶに対応するＩ／Ｏデバイス５０の共有を実現する。

【００５８】

図５は、ブレード１０－１（Ｂｌａｄｅ＃１）～１０－ｎ（Ｂｌａｄｅ＃ｎ）及びＰＣＩ管理サーバ２０（ＰＣＩ－Ｍ）のチップセット１３－１～ｎ、Ｍが管理するルーティングＩＤの関係を示す。各チップセット１３－１～ｎ、Ｍは、それぞれ、各計算機内のデバイスについて認識したローカルＰＣＩツリー用のルーティングＩＤと、ＰＣＩ－ｅスイッチ４０のＭＲＳ論理４１０から取得したＭＲＳ論理用ルーティングＩＤと、各チップセット１３－１～ｎ、ＭからのＩ／Ｏデバイス５０のアクセス用ルーティングＩＤとを備える。

【００５９】

ブレード１０－１～１０－ｎのチップセット１３－１～ｎでは、接続しているＰＣＩ－ｅスイッチ４０のアップストリームポート４１－１～ｎが異なるため、チップセット１３－１～ｎが認識するＩ／Ｏデバイス５０のバス番号は異なる。例えば、図示の例では、ブレード１０－１に２つの仮想機能５０２＝ＶＦ１、ＶＦ２を割り当て、ブレード１０－ｎにも２つの仮想機能５０２＝ＶＦｋ－１、ＶＦ２ｋ割り当てた例を示す。

【００６０】

チップセット１３－１からＩ／Ｏデバイス５０の仮想機能５０２＝ＶＦ１へのルーティングＩＤは、１１：０：１であり、バス番号は１１であるのに対し、チップセット１３－ｎからＩ／Ｏデバイス５０の仮想機能５０２＝ＶＦｋ－１へのルーティングＩＤは、１３：ｘ：ｙ－１でありバス番号は１３である。ＭＲＳ論理用ルーティングＩＤも同様であり、ブレード１０－１～１０－ｎ毎に異なる。なお、ルーティングＩＤは、バス番号（Ｂｕｓ＃）：デバイス番号（Ｄｅｖ＃）：ファンクション番号（Ｆｕｎ＃）の順に設定される。

【００６１】

一方、Ｉ／Ｏデバイス５０の物理機能５０１にアクセス可能なＰＣＩマネージャ２０２は、起動時に物理機能５０１と仮想機能５０２のルーティングＩＤを初期化してチップセット１３－Ｍで管理しており、図示の例ではルーティングＩＤ＝１０：０：０～１０：ｘ：ｙ－１をＩ／Ｏデバイス５０のアクセス用ルーティングＩＤに設定する。

【００６２】

そして、ＰＣＩマネージャ２０２は、各ブレード１０－１～１０－ｎが起動する度に、図７に示すＴＬＰ変換情報４４０の仮想機能ＩＤ修飾情報４４０２に、各ブレード１０－１～１０－ｎのチップセット１３－１～ｎからＩ／Ｏデバイス５０の仮想機能５０２まで

のルーティングＩＤと、ＰＣＩ管理サーバ２０のチップセット１３－Mが管理するＩ／Ｏデバイス５０のルーティングＩＤを、各ブレード１０－１～１０－ｎ毎の仮想階層ナンバー１３１１と共に設定する。

【００６３】

ルーティングＩＤで各ブレード１０－１～１０－ｎとＩ／Ｏデバイス５０がアクセスを行う場合には、ＰＣＩ－ｅスイッチ４０が仮想機能ＩＤ修飾情報４４０２を参照し、パケット中のルーティングＩＤを付け替えて通信を行う。

【００６４】

すなわち、ＰＣＩ－ｅスイッチ４０は、ブレード１０－１～１０－ｎからＩ／Ｏデバイス５０へのパケットがルーティングＩＤでアクセスする場合、各ブレード１０－１～１０－ｎのルーティングＩＤをＰＣＩ管理サーバ２０のルーティングＩＤに差し替えてＩ／Ｏデバイス５０に送信する。

【００６５】

逆に、Ｉ／Ｏデバイス５０からブレード１０－１～１０－ｎへのパケットがルーティングＩＤでアクセスする場合、ＰＣＩ－ｅスイッチ４０は、ＰＣＩ管理サーバ２０のルーティングＩＤを各ブレード１０－１～１０－ｎのルーティングＩＤ差し替えてＩ／Ｏデバイス５０に送信する。

【００６６】

以上の処理により、ＰＣＩ－ｅスイッチ４０は、ルーティングＩＤを差し替えることで、複数のブレード１０－１～１０－ｎでＳＲ－ＩＯＶに対応するＩ／Ｏデバイス５０の共有を実現する。

【００６７】

ここで、ブレードサーバ１で使用されるＰＣＩ－ｅｘｐｒｅｓｓのパケットのフォーマットを図１５に示す。図１５はＰＣＩ－ｅスイッチ４０で転送されるＰＣＩｅベースＴＬＰ１２００のうちＴＬＰヘッダをアクセス形態毎に示す説明図である。

【００６８】

図１５は、ＰＣＩｅベースＴＬＰ１２００のＴＬＰヘッダの詳細を示し、ＴＬＰヘッダ１２００Ａは、ＭＭＩＯ空間のアドレスでブレード１０－１～１０－ｎとＩ／Ｏデバイス５０のアクセスを行うパケットを示し、ＴＬＰヘッダ１２００Ｂ、１２００Ｃはルーティング ＩＤでアクセスを行うパケットを示しており、ＴＬＰヘッダ１２００Ｂは要求（書き込み要求など）に対する完了通知を通知するパケットの例を示し、ＴＬＰヘッダ１２００Ｃは対象のデバイスの構成情報（コンフィギュレーション）の設定するパケットを示す。

【００６９】

各ＴＬＰヘッダ１２００Ａ～Ｃは、バイト０～１５の１６バイトで構成される。

【００７０】

ＭＭＩＯ空間のアドレスでアクセスするＴＬＰヘッダ１２００Ａは、メモリリード要求やメモリライト要求の際に使用される。

【００７１】

ＴＬＰヘッダ１２００Ａは、バイト４、５にパケットの発行元（リクエスタ）ＩＤに要求元のルーティングＩＤを格納し、バイト８～１５に読み込む対象のＭＭＩＯ空間のアドレスを格納し、バイト０の０～４ビットに要求の種別を格納する。

【００７２】

ＰＣＩ－ｅスイッチ４０は、ブレード１０－１～１０－ｎからＩ／Ｏデバイス５０へ向かう下りパケットの時には、リクエスタＩＤに格納されたブレードのルーティングＩＤをＰＣＩ管理サーバ２０のルーティングＩＤに書き換えて、要求元がＰＣＩ管理サーバ２０であることを設定する。つまり、ＳＲ－ＩＯＶのＩ／Ｏデバイス５０は、ひとつの計算機のみと接続されるため、Ｉ／Ｏデバイス５０へのパケットはＰＣＩ管理サーバ２０発に差し替える。

【００７３】

そして、ＰＣＩ－ｅスイッチ４０は、ブレードのＭＭＩＯ空間のアドレスを宛先アドレス修飾情報４４０１の「ｏｆｆｓｅｔ」でＰＣＩ管理サーバ２０（ＰＣＩマネージャ２０２）のＭＭＩＯ空間のアドレスに変更し、Ｉ／Ｏデバイス５０が認識可能なアドレス空間に変更する。

【００７４】

ルーティングＩＤでアクセスする場合の、ＴＬＰヘッダ１２００Ｂは、要求（書き込み要求等）に対する完了通知の際に使用される。ＴＬＰヘッダ１２００Ｂは、バイト４，５のコンプリータＩＤに完了通知（コンプリーション）を発行するデバイスのルーティングＩＤを格納し、バイト８、９のリクエスタＩＤに要求元（リクエスタ）のルーティングＩＤを格納し、バイト０の０～４ビットにコンプリーションを示す値を格納する。

【００７５】

ＰＣＩ－ｅスイッチ４０は、仮想機能ＩＤ修飾情報４４０２を参照して、Ｉ／Ｏデバイス５０へ向かう上りパケットの時には、リクエスタＩＤに格納されたＰＣＩ管理サーバ２０のルーティングＩＤをＰＣＩ管理サーバ２０のルーティングＩＤから送信先のブレードのルーティングＩＤに差し替えて送信する。

【００７６】

ルーティングＩＤでアクセスするＴＬＰヘッダ１２００Ｃは、デバイスに対する構成情報の設定を要求するパケットで使用される。ＴＬＰヘッダ１２００Ｃは、バイト４，５のリクエスタＩＤに構成情報の設定を要求するデバイスのルーティングＩＤを格納し、バイト８、９には構成情報の設定対象のバス番号とデバイス番号とファンクション番号からなるルーティングＩＤを格納し、バイト０の０～４ビットに構成情報の設定要求（コンフィギュレーション）を示す値を格納する。

【００７７】

ＰＣＩ－ｅスイッチ４０は、ブレード１０－１～１０－ｎからＩ／Ｏデバイス５０へ向かう下りパケットの時には、リクエスタＩＤに格納されたブレードのルーティングＩＤをＰＣＩ管理サーバ２０のルーティングＩＤに書き換えて、要求元がＰＣＩ管理サーバ２０であることを設定する。また、ＰＣＩ－ｅスイッチ４０は、仮想機能ＩＤ修飾情報４４０２を参照して、ブレードが設定したＩ／Ｏデバイス５０のルーティングＩＤ（バス番号、デバイス番号、ファンクション番号を、）ＰＣＩ管理サーバ２０が認識するルーティングＩＤに差し替えて、Ｉ／Ｏデバイス５０にはＰＣＩ管理サーバ２０からの要求であることに変更する。

【００７８】

図１６は、ブレードサーバ１の起動手順を示すフローチャートである。この処理は、ブレードサーバ１の管理者またはユーザのスイッチ操作に基づいて開始される。

【００７９】

ステップＳ１では、管理者（またはユーザ）がＰＣＩ－ｅスイッチ４０の電源を投入する。ＰＣＩ－ｅスイッチ４０は、起動するとＰＣＩ－ｅリンクの初期化処理と各種レジスタの初期化処理を実行する（Ｓ２）。すなわち、図３のＭＲＳ構成情報４１５及びＴＬＰ変換情報４４０の初期化が行われる。また、管理者などによりＩ／Ｏデバイス５０の電源が投入される。なお、Ｉ／Ｏデバイス５０の電源は、ＰＣＩ－ｅスイッチ４０と連動していても良い。

【００８０】

ステップＳ３では、管理者（またはユーザ）が、ＰＣＩ管理サーバ２０の電源を投入する。ＰＣＩ管理サーバ２０では、後述する図１７のように各種初期化が行われる。ＰＣＩ管理サーバ２０の起動が完了すると、ステップＳ４ではＰＣＩマネージャ２０２が管理端末３０からのブレードに対する起動の指示を待ち受ける。

【００８１】

ＰＣＩマネージャ２０２は管理端末３０から起動するブレード１０－１～１０－ｎに対する指令を受け付けると、ステップＳ５に進んで指令を受け付けたブレード１０－１～１０－ｎの電源を投入し、その後、ステップＳ６では、ＰＣＩマネージャ２０２が、起動が

完了したブレードに対して仮想マシンモニタ１００－１～１００－ｎを起動する。仮想マシンモニタ１００－１～１００－ｎの起動については後述の図１８にて詳述する。ステップＳ７では、仮想マシンモニタの起動が完了した後に、受け付けた指令に基づいて仮想計算機１０１－１～１０１－ｋ－１を生成し、各仮想計算機１０１－１～１０１－ｋ－１でＯＳ１０２－０～１０２－Ｋ－１を起動させる。仮想計算機の起動が完了すると再びステップＳ４に戻って次のブレード１０－１～１０－ｎの起動の指令を待つ。

【００８２】

上記処理により、ＰＣＩ－ｅスイッチ４０、ＰＣＩ管理サーバ２０、Ｉ／Ｏデバイス５０の順で電源が投入されて初期化が行われた後、管理端末３０から指令されたブレード１０－１～１０－ｎが起動される。

【００８３】

図１７は、図１６に示したステップＳ３のＰＣＩ管理サーバ２０の起動の際に行われる処理の一例を示すフローチャートである。

【００８４】

ＰＣＩ管理サーバ２０は、電源を投入されるとステップＳ１１でＢＩＯＳ（Basic I/o System）またはＥＦＩ(Extensible Firmware Interface)が起動する。ＢＩＯＳ（またはＥＦＩ）の起動時には、ステップＳ１２でチップセット１３－MがＰＣＩ管理サーバ２０内のデバイスについて、ルートコンプレックスの下にローカルＰＣＩツリーを構成し、ＰＣＩローカルツリーの初期化を行う。すなわち、チップセット１３－Mは、図５で示したような、ローカルＰＣＩツリーのルーティングＩＤをチップセット１３－Mのレジスタ等の所定の領域に設定する。この例では、ＰＣＩ管理サーバ２０内のローカルルーティングＩＤとして、０：０：０～７：１Ｆ：７が各エントリに設定される。

【００８５】

ステップＳ１３では、ＢＩＯＳ（またはＥＦＩ）の起動が完了するとＯＳ２０１が起動し、その後に、ＰＣＩマネージャ２０２（ＰＣＩ－M）が起動する。このとき、ＯＳ２０１はＰＦドライバ２０３を読み込んで、ＰＣＩマネージャ２０２がＩ／Ｏデバイス５０の物理機能ＰＦ５０１を利用する準備を行う。なお、Ｉ／Ｏデバイス５０の物理機能を使用するのはＰＣＩ－ｅスイッチ４０及びＩ／Ｏデバイス５０を管理するＰＣＩマネージャ２０２のみであり、他のブレード１０－１～１０－ｎは仮想機能を利用する。

【００８６】

ステップＳ１４では、ＯＳ２０１がネットワークインターフェース１３２－Mの初期化を実施する。ネットワークインターフェース１３２－Mの初期化は、ネットワーク７０への接続とＩＰアドレスの付与及びブレード１０－１～１０－ｎの仮想計算機からの接続要求に対する待ち受けの開始が含まれる。

【００８７】

次に、ステップＳ１５では、ＰＣＩマネージャ２０２がＩ／Ｏポート１３１－MからＰＣＩ－ｅスイッチ４０にアクセスして、マルチルートスイッチ論理４１０のアップストリームポートブリッジ４１２からアップストリームポート４１－０に接続されたポート４１１－０に設定された構成情報（例えば、マルチルート化可能ポインタ：Capability pointer）を取得し、当該ポート４１１－０のバス番号（例えば、９）を取得する。ここで、構成情報としてのマルチルート化可能ポインタは、図１９で示すように、４バイト目の０～１５ビットのＶＳＥＣ＿ＩＤがマルチルート化可能ポインタを示す。ＰＣＩマネージャ２０２は、ＰＣＩ－ｅスイッチ４０から仮想階層ナンバー１３１１を取得する。図４の例では、ＰＣＩ管理サーバ２０の仮想階層ナンバー１３１１（ＶＨ）は「０」となる。

【００８８】

ステップＳ１６では、ＰＣＩマネージャ２０２が、Ｉ／ＯポートＰＣＩ－ｅスイッチ４０にアクセスして、ＭＲＳ論理４１０のダウンストリームポートブリッジ４１３の構成情報（例えば、コンフィギュレーションヘッダ）を取得する。

【００８９】

次に、ステップＳ１７では、ＰＣＩマネージャ２０２は、Ｉ／Ｏポート１３１－M及び

アップストリームポート４１－０からＩ／Ｏデバイス５０にアクセスし、Ｉ／Ｏデバイス５０のＳＲ－ＩＯＶに関する構成情報（例えば、コンフィギュレーションヘッダ）を取得する。ＰＣＩマネージャ２０２は、Ｉ／Ｏデバイス５０の構成情報から仮想機能（ＶＦ）の個数やＭＭＩＯ空間などを取得する。また、ＰＣＩマネージャ２０２はＩ／Ｏデバイス５０からバス番号及びデバイス番号を取得する。例えば、ＰＣＩ管理サーバ２０のチップセット１３－Ｍのルートコンプレックスから見たＩ／Ｏデバイス５０のアクセス用ルーティングＩＤは０：２：０となる。

【００９０】

ステップＳ１８では、上記ステップＳ１５～Ｓ１７で取得したマルチルートスイッチ論理４１０のアップストリームポートブリッジ４１２とダウンストリームポートブリッジ４１３の構成情報と、Ｉ／Ｏデバイス５０の構成情報から、図５で示すように、ＭＲＳ論理用ルーティングＩＤと、Ｉ／Ｏデバイス５０をアクセスするためのルーティングＩＤ（ＤＥＶアクセス用Routing ID）をチップセット１３－Ｍのレジスタ等の所定の領域に設定する。また、ＰＣＩマネージャ２０２は、めもり１２－Ｍの所定の領域に物理機能５０１にアクセするためのＭＭＩＯと、仮想機能５０１にアクセスするためのＭＭＩＯをｋ個設定する。

【００９１】

図５の例では、ＰＣＩマネージャ２０２がＰＣＩ管理サーバ２０のＩ／Ｏポート１３１－ＭからＰＣＩ－ｅスイッチ４０のアップストリームポート４１－０からダウンストリームポートブリッジ４１３までのＭＲ－ＩＯＶのルーティングＩＤとして、ポート４１１－０がバス番号＝９、ＭＲＳ構成情報４１５がバス番号＝８、Ｉ／Ｏデバイス５０のバス番号＝１０となり、ＭＲＳ論理４１０用のルーティングＩＤは、８：０：０、９：０：０となる。また、Ｉ／Ｏデバイス５０のデバイスアクセス用ルーティングＩＤは、１０：０：０がＩ／Ｏデバイス５０の物理機能にアクセスするためのルーティングＩＤに設定され、１０：０：１～１０：０：ＶＦｋまでのｋ個のルーティングＩＤがＩ／Ｏデバイス５０のアクセス用に設定される。そして、ＰＣＩマネージャ２０２がＰＦドライバ２０３でＩ／Ｏデバイス５０の物理機能ＰＦ５０１と仮想機能５０２を初期化する。

【００９２】

上記処理によって、Ｉ／Ｏデバイス５０の物理機能ＰＦ５０１と仮想機能５０２を初期化がＰＣＩマネージャ２０２によって行われる。

【００９３】

図１８は、図１６のステップＳ６、Ｓ７で行われるブレード１０－１～１０－ｎの起動と仮想計算機（ゲストＶＭ）の起動処理の一例を示すフローチャートである。この例では、管理端末３０から図１のブレード１０－１（ＢＬＡＤＥ＃１）を起動し、ブレード１０－１の仮想マシンモニタ１００－１上で仮想計算機１０１－１（ＶＭ＃０）、１０１－２（ＶＭ＃１）を生成する指令があった例を示す。

【００９４】

ＰＣＩマネージャ２０２は、既に起動してステップＳ３１にて仮想マシンモニタ１００－１～１００－ｎからの通信を待ち受ける状態となっている（図１７のＳ１４）。

【００９５】

ＰＣＩ管理サーバ２０は管理端末３０から指令を受けたブレード１０－１（ＢＲＡＤＥ＃１）に起動の指令を出力し、ブレード１０－１を起動する。この処理は、例えば、ブレード１０－１～１０－ｎがＢＭＣを備えている場合には、該当するブレードのＢＭＣに対して起動を指令し、ネットワークインターフェース１３２－１～ｎへ起動を指令する場合にはマジックパケットを送信することで行われる。

【００９６】

ＰＣＩ管理サーバ２０からの指令で起動したブレード１０－１は、電源を投入されるとステップＳ２１でＢＩＯＳまたはＥＦＩが起動する。ＢＩＯＳ（またはＥＦＩ）の起動時には、ステップＳ２２でチップセット１３－１がブレード１０－１内のデバイスについてルートコンプレックスの下にローカルＰＣＩツリーを構成し、ＰＣＩローカルツリーの初

期化を行う。すなわち、チップセット１３－１は、図５で示したＢｌａｄｅ＃１のように、ローカルＰＣＩツリーのルーティングＩＤをチップセット１３－１のレジスタ等の所定の領域に設定する。この例では、ブレード１０－１内のローカルルーティングＩＤとして、０：０：０～８：１Ｆ：７が各エントリに設定される。

【００９７】

ステップＳ２３では、ＢＩＯＳ（またはＥＦＩ）の起動が完了すると仮想マシンモニタ１００－１が起動する。仮想マシンモニタ１００－１がネットワークインターフェース１３２－１の初期化を実施する。ネットワークインターフェース１３２－１の初期化は、ネットワーク７０への接続とＩＰアドレスの付与が含まれる。

【００９８】

次に、ステップＳ２４では、仮想マシンモニタ１００－１がＩ／Ｏポート１３１－１からＰＣＩ－ｅスイッチ４０にアクセスして、マルチルートスイッチ論理４１０のアップストリームポートブリッジ４１２からアップストリームポート４１－０に接続されたポート４１１－１に設定された構成情報（例えば、マルチルート化可能ポインタ：Capability pointer）を取得し、当該ポート４１１－１のバス番号（例えば、９）を取得する。構成情報としてのマルチルート化可能ポインタは、図１９で示したように、４バイト目の０～１５ビットのＶＳＥＣ＿ＩＤがマルチルート化可能ポインタを示す。仮想マシンモニタ１００－１は、ＰＣＩ－ｅスイッチ４０から仮想階層ナンバー１３１１を取得する。図５の例では、ブレード１０－１の仮想階層ナンバー１３１１（ＶＨ）は「１」となる。

【００９９】

ステップＳ２５では、仮想マシンモニタ１００－１が、Ｉ／Ｏポート１３１－１からＰＣＩ－ｅスイッチ４０にアクセスして、ＭＲＳ論理４１０のダウンストリームポートブリッジ４１３の構成情報（例えば、コンフィギュレーションヘッダ）を取得する。仮想マシンモニタ１００－１は、ルートコンプレックスから見たＩ／Ｏデバイス５０のルーティングＩＤのデバイス番号として、ＭＲＳ論理４１０のダウンストリームポートブリッジ４１３のデバイスを割り当てる。この例では、ルートコンプレックスから見たＩ／Ｏデバイス５０のルーティングＩＤは、０：６：０となる。

【０１００】

次に、ステップＳ２６では、仮想マシンモニタ１００－１が必要とする仮想機能５０２の数を含む確保要求をＰＣＩマネージャ２０２にＶＦ確保要求データとしてネットワーク７０から送信する。この確保要求は、図２０で示すように、仮想マシンモニタ１００－１が必要とする仮想機能（ＶＦ）５０２の数（＝２）と、仮想マシンモニタ１００－１がひとつの仮想機能５０２に対してメモリ１２－１に設定可能なＭＭＩＯのサイズの最大値と、仮想機能５０２を設定するＭＭＩＯのベースアドレス及び仮想階層ナンバー１３１１（ＶＨ＝１）と、Ｉ／Ｏデバイス５０からアクセスするためのルーティングＩＤ（１１：０：１及び１１：０：２）とを含む。なお、ブレード１０－１から見たＩ／Ｏデバイス５０のバス番号は１１となり、ＰＣＩ管理サーバ２０から見たバス番号とは異なる。また、ブレード１０－１は、確保要求にルートコンプレックスが取得したＩ／Ｏデバイス５０のアクセス用ルーティングＩＤ（０：６：０）も通知する。なお、Ｉ／Ｏデバイス５０からアクセスするためのルーティングＩＤは、各ブレード１０－１～１０－ｎの起動時にＢＩＯＳが決定した値を用いることができる。

【０１０１】

仮想機能５０２を確保する要求を受け付けたＰＣＩマネージャ２０２は、ブレード１０－１から見たＰＣＩ－ｅスイッチ４０のＭＲＳ論理４１０及びＩ／Ｏデバイス５０のルーティングＩＤと、ブレード１０－１のＭＭＩＯ空間を認識する。

【０１０２】

ＰＣＩマネージャ２０２は、ブレード１０－１から要求された仮想機能５０２に対するＩ／Ｏデバイス５０のアクセス用ルーティングＩＤから図５に示したＶＦ１及びＶＦ２をブレード１０－１に割り当て、図４に示したＭＭＩＯ空間に確保したＶＦ１，ＶＦ２のＭＭＩＯをブレード１０－１に割り当てる。

【０１０３】
そして、ＰＣＩマネージャ２０２は、ブレード１０－１から受信した確保要求に含まれるブレード１０－１のＩ／Ｏデバイス５０のアクセス用ルーティングＩＤと、仮想階層ナンバー１３１１と、ＭＭＩＯのベースアドレスと、仮想階層ナンバー１３１１を、ＭＲ－ＳＲ変換論理５３０のＴＬＰ変換情報４４０に書き込む。このとき、ＰＣＩマネージャ２０２は、確保要求で要求された仮想機能５０２の数に応じてＰＣＩマネージャ２０２が管理する仮想機能５０２のＭＭＩＯとブレード１０－１が管理するＭＭＩＯのオフセットを求める。
【０１０４】
ＰＣＩマネージャ２０２は、ＭＲ－ＳＲ変換論理５３０のＴＬＰ変換情報４４０（図７）を構成する宛先アドレス変換論理４４０１へ、仮想機能５０２毎に仮想階層ナンバー１３１１、ブレード１０－１のＭＭＩＯのベースアドレスと、ＭＭＩＯのサイズと、ＰＣＩマネージャ２０２が管理する仮想機能５０２のＭＭＩＯとブレード１０－１が管理するＭＭＩＯのオフセットを書き込む。ＰＣＩマネージャ２０２は、書き込みを行ったエントリのＶａｌｉｄに有効であることを示す「１」をセットする。
【０１０５】
次に、ＰＣＩマネージャ２０２は、図５のルーティングＩＤからブレード１０－１に割り当てた仮想機能５０２（ＶＦ１とＶＦ２）のルーティングＩＤ（１０：０：１と１０：０：２）と、ブレード１０－１から通知を受けたルーティングＩＤ（１１：０：１と１１：０：２）を対応づける。
【０１０６】
そして、ＰＣＩマネージャ２０２は、ＭＲ－ＳＲ変換論理５３０のＴＬＰ変換情報を構成する仮想機能ＩＤ修飾情報４４０２（図７）へ、各ブレードからＩ／Ｏデバイス５０のアクセス用ルーティングＩＤ毎に、ブレード１０－１～１０－ｎの仮想階層ナンバー１３１１（ＶＨ）をＶＨに書き込み、／Ｏデバイス５０のアクセス用ルーティングＩＤをＶＨｘ＿ＲｏｕｔｉｎｇＩＤに書き込み、ブレードのルーティングＩＤに対応するＰＣＩマネージャ２０２が対応付けたルーティングＩＤをＶＨ０　ＲｏｕｔｉｎｇＩＤに書き込む。ＰＣＩマネージャ２０２は、書き込みを行ったエントリのＶａｌｉｄに有効であることを示す「１」をセットする。
【０１０７】
次に、ＰＣＩマネージャ２０２は、図７のルートポートＩＤ情報４４０３に、ブレード１０－１のルートコンプレックスが認識したＩ／Ｏデバイス５０のルーティングＩＤ（０：６：０）を仮想階層ナンバー１３１１毎に書き込む。ＰＣＩマネージャ２０２は、書き込みを行ったエントリのＶａｌｉｄに有効であることを示す「１」をセットする。
【０１０８】
以上により、ＭＲ－ＳＲ変換論理４３０のＴＬＰ変換情報４４０にブレード１０－１から受信したＶＦ確保要求データの値を設定し、ブレード１０－１がＩ／Ｏデバイス５０を共有することを設定する。
【０１０９】
ＰＣＩマネージャ２０２は、ＭＲ－ＳＲ変換論理４３０のＴＬＰ変換情報４４０にブレード１０－１から受信したＶＦ確保要求データの設定が完了すると、ＴＬＰ変換情報４４０に設定した情報をＶＦ確保完了データとしてネットワーク７０からブレード１０－１に送信する。ＶＦ確保完了データの一例としては図２１のようになる。図２１において、ＶＦ確保完了データは、ＰＣＩマネージャ２０２がＴＬＰ変換情報４４０に確保した仮想機能５０２の数（＝２）と、仮想機能５０２当たりのＭＭＩＯのサイズと、Ｉ／Ｏデバイス５０の物理機能ＰＦ５０１の構成情報の空間サイズと、Ｉ／Ｏデバイス５０の物理機能５０１の構成空間データと、Ｉ／Ｏデバイス５０の仮想機能５０２の構成空間サイズと、Ｉ／Ｏデバイス５０の仮想機能５０２の構成空間データが含まれる。
【０１１０】
ＶＦ確保完了データを受信した仮想マシンモニタ１００－１は、ステップＳ２７で管理

端末３０から指令された仮想計算機１０１－１、１０１－２（ゲストＶＭ）を起動し、ＰＣＩマネージャ２０２から受信したＩ／Ｏデバイス５０の仮想機能５０２（ＶＦ１，ＶＦ２）をそれぞれ割り当てる。

【０１１１】

上記の処理により、仮想マシンモニタ１００－１が必要とするＩ／Ｏデバイス５０の仮想機能５０２をＰＣＩマネージャ２０２が確保し、仮想マシンモニタ１００－１はＰＣＩマネージャ２０２が確保した仮想機能５０２を起動する仮想計算機１０１－１、１０１－２に割り当てることができる。

【０１１２】

なお、上記の処理は他のブレード１０－２～１０－ｎを起動するときにも同様に実行され、ＴＬＰ変換情報４４０にはＩ／Ｏデバイス５０を共有するブレード１０－１～１０－ｎの情報が加えられることになる。

【０１１３】

図２１は、上記図１８のステップＳ２７で行われる仮想計算機１０１－０（１０１－１）の起動処理の一例を示すフローチャートである。

【０１１４】

仮想マシンモニタ１００－１は、ステップＳ４１で仮想計算機１０１－０を起動し、ステップＳ４２で仮想計算機１０１－０に割り当てた仮想ＢＩＯＳを起動する。ステップＳ４２では、仮想ＢＩＯＳによって仮想ＰＣＩツリーの初期化を実施する。

【０１１５】

ステップＳ４４では、仮想ＢＩＯＳが仮想マシンモニタ１００－１によって提供される仮想コンフィギュレーションにアクセスを開始する。ステップＳ４５では、仮想マシンモニタ１００－１は仮想ＢＩＯＳのアクセス対象が、Ｉ／Ｏデバイス５０の仮想機能５０２に対するものであるか否かを判定する。仮想機能５０２以外に対して仮想ＢＩＯＳがアクセスした場合には、ステップＳ４６で仮想マシンモニタ（ＶＭＭ）１００－１がアクセス対象のエミュレーションを実施する。一方、仮想機能５０２に対して仮想ＢＩＯＳがアクセスした場合には、ステップＳ４７で仮想マシンモニタ１００－１が、ＰＣＩマネージャ２０２から仮想機能確保応答情報により取得したＩ／Ｏデバイス５０の物理機能ＰＦ５０１の構成情報の空間サイズと、Ｉ／Ｏデバイス５０の物理機能５０１の構成空間データと、Ｉ／Ｏデバイス５０の仮想機能５０２の構成空間サイズと、Ｉ／Ｏデバイス５０の仮想機能５０２の構成情報の空間データを応答する。

【０１１６】

ステップＳ４８では、仮想マシンモニタ１００－１が仮想ＢＩＯＳによる仮想ＰＣＩツリーの初期化が完了したか否かを判定する。仮想ＰＣＩツリーの初期化が完了していなければステップＳ４４に戻り上記初期化を繰り返す。一方、仮想ＰＣＩツリーの初期化が完了すると、ステップＳ４９で仮想マシンモニタ１００－１がＯＳ１００－０（ゲストＯＳ）の起動を開始する。ステップＳ５０でＯＳ１００－０の起動が完了し、仮想計算機１０１－０でアプリケーションやサービスの提供が可能となる。

【０１１７】

上記により、仮想ＢＩＯＳやＯＳ１００－０（ゲストＯＳ）がＩ／Ｏデバイス５０の仮想機能５０２の構成情報を取得する場合には、仮想マシンモニタ１００－１がＰＣＩマネージャ２０２から取得した物理機能５０１及び仮想機能５０２の情報を提供することで、ＯＳ１００－０等がＶＦドライバ１０３を組み込んで仮想機能５０２を利用することが可能となる。

【０１１８】

図２３は、上記実施形態の変形例を示し、図３に示したＭＲＳ論理４１０のうち、アップストリームポートブリッジ４１２に接続されるポート４１１－１～４１１－ｎをＭＲＳ論理４１０から分離して、アップストリームポート４１－０～４１－ｎの増設を可能にしたものである。

【０１１９】

ＰＣＩ－ｅスイッチ４０のアップストリームポート４１－０～４１－ｎは、所定数（例えば、２つ）毎にＭＲＳ論理４１１０－１～４１１０－ｎに接続される。各ＭＲＳ論理４１１０－１～ｎには、２つのポート４１１－０、４１１－１等がそれぞれ設けられて、アップストリームポート４１－０～４１－ｎに接続される。各ＭＲＳ論理４１１０－１～４１１０－ｎのＭＲＳ論理４１０側にはひとつのポート４１１１－１～４１１１－ｎが設けられており、それぞれのポート４１１１－１～４１１１－ｎはＭＲＳ論理４１０に設けたポート４１１２－１～４１１２－ｎに接続される。

【０１２０】

この変形例では、ＭＲＳ論理４１０のアップストリームポート４１－０～４１－ｎ側を多段化することによって、ブレードサーバ１のアップストリームポート４１－０～４１－ｎの構成を柔軟に行うことが可能となる。

【０１２１】

なお、上記実施形態では、チップセット１３－１～ｎとプロセッサ１１－１，２が独立した構成を示したが、プロセッサ１１－１、２にチップセット１３－１～ｎが組み込まれていても良い。

【０１２２】

また、上記実施形態では、ＰＣＩ管理サーバ２０でＰＣＩマネージャ２０２を稼動させる例を示したが、ブレード１０－１～１０－ｎの何れかでＰＣＩマネージャ２０２を稼動させてもよい。

【産業上の利用可能性】

【０１２３】

以上のように、本発明は、複数の物理計算機をＰＣＩ－ｅｘｐｒｅｓｓのスイッチでＩ／Ｏデバイスに接続する計算機システムに適用することができ、特に、物理計算機がＭＲ－ＩＯＶでアクセスを行い、Ｉ／ＯデバイスがＳＲ－ＩＯＶでアクセスする計算機システム及びＰＣＩスイッチに好適である。

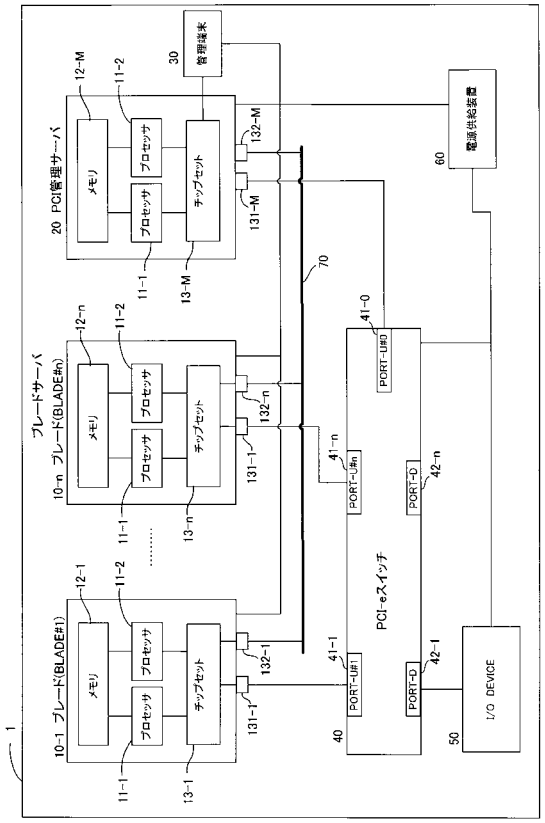
【図面の簡単な説明】

【０１２４】

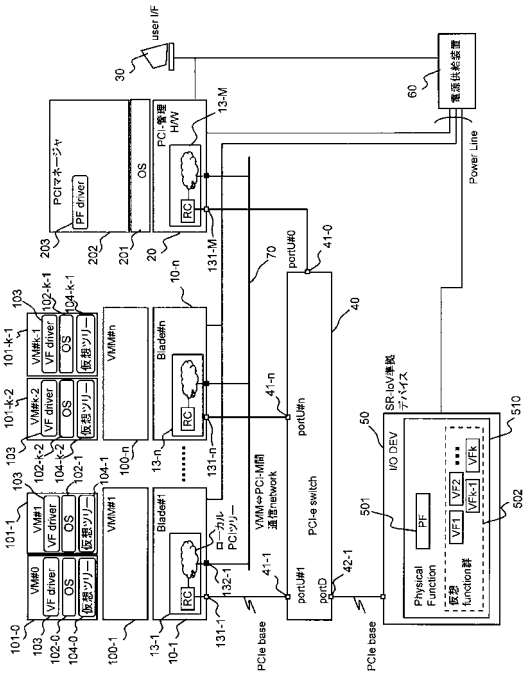
【図１】本発明を適用したブレードサーバのハードウェア構成を示すブロック図。
【図２】同じくブレードサーバの機能要素を示すブロック図。
【図３】ＰＣＩ－ｅスイッチの内部を示すブロック図。
【図４】各ブレードとＰＣＩ管理サーバのＭＭＩＯ空間の関係を示す説明図。
【図５】各ブレードとＰＣＩ管理サーバのルーティングＩＤの関係を示す説明図。
【図６】ＭＲ－ＳＲ変換論理の宛先情報変換回路の構成を示すブロック図。
【図７】ＴＬＰ情報の構成を示す説明図。
【図８】宛先情報変換回路を構成する宛先アドレス修飾部の構成を示すブロック図。
【図９】宛先情報変換回路を構成する宛先ＩＤ修飾部の構成を示すブロック図。
【図１０】発行元ＩＤ変換回路１の構成を示すブロック図。
【図１１】仮想階層ナンバー付加回路の構成を示すブロック図。
【図１２】発行元ＩＤ変換回路２の構成を示すブロック図。
【図１３】宛先ＩＤ変換回路２の構成を示すブロック図。
【図１４】マルチルートＴＬＰ及びＰＣＩｅベースＴＬＰのフォーマットを示す説明図。
【図１５】マルチルートＴＬＰ及びＰＣＩｅベースＴＬＰのＴＬＰヘッダのフォーマットを示す説明図。
【図１６】ブレードサーバの電源投入の手順を示すフローチャート。
【図１７】ＰＣＩ管理サーバ２０で行われる起動処理を示すフローチャート。
【図１８】ブレードで行われる仮想マシンモニタの起動処理を示すフローチャート。
【図１９】マルチルート化可能フォーマットを示す説明図。
【図２０】仮想機能確保要求情報のフォーマットを示す説明図。
【図２１】仮想機能確保完了情報のフォーマットを示す説明図。
【図２２】仮想計算機の起動処理を示すフローチャート。

【図２３】ＰＣＩ－ｅスイッチのＭＲＳ論理の変形例を示すブロック図。
【符号の説明】
【０１２５】
１０－１～１０－ｎ　ブレード
１３－１～１３－ｎ、１３－Ｍ　チップセット
２０　ＰＣＩ管理サーバ２０
４０　ＰＣＩ－ｅスイッチ
５０　Ｉ／Ｏデバイス５０
４１－０～４１－ｎ　アップストリームポート
４２－１　ダウンストリームポート
４１０　ＭＲＳ論理
４３０　ＭＲ－ＳＲ変換論理
４３１　宛先情報変換回路
４３２　発行元ＩＤ変換回路（１）
４３３　Ｖｈｎ消去部
４３４　ＬＣＲＣ再生成部
４３５　Ｖｈｎ付加部
４３６　発行元ＩＤ変換回路（１）
４３７　宛先ＩＤ変換回路（２）
４３８　ＬＣＲＣ再生成部

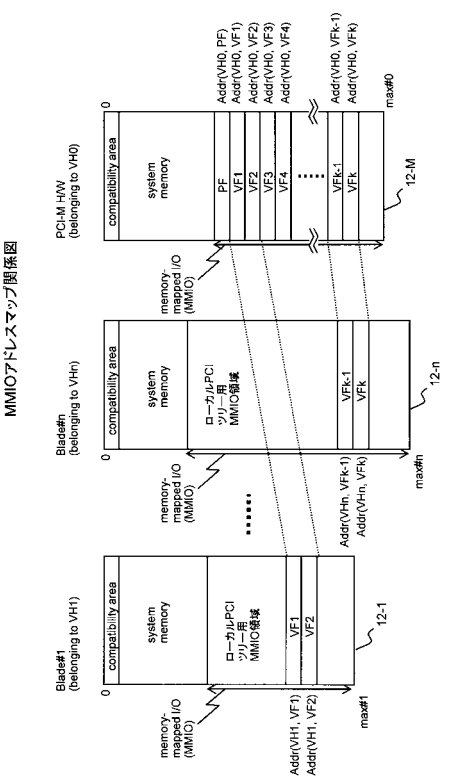
【図１】



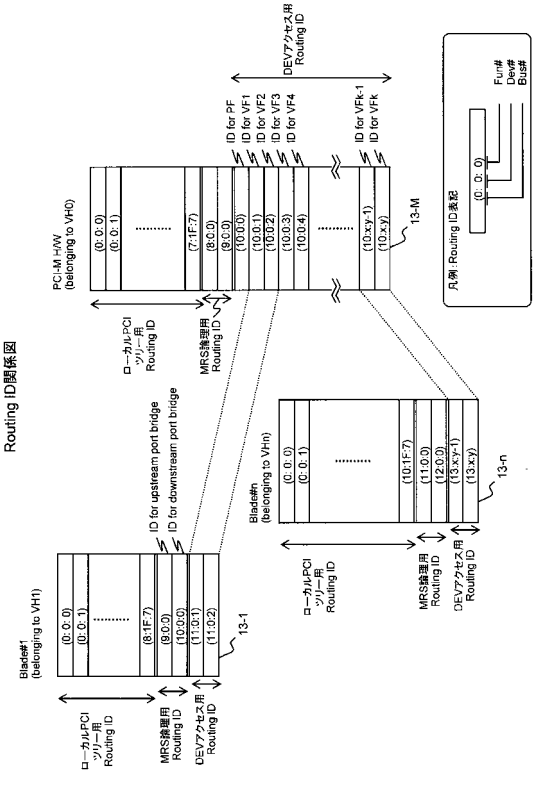
【図２】



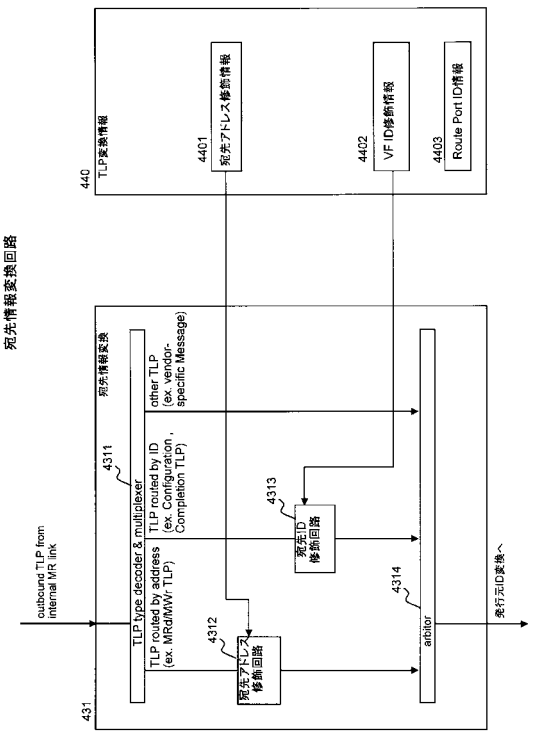
【図４】


MMIOアドレスマップ関係図
Blade#1 (belonging to VH1)
Blade#n (belonging to VHn)
PCI-M H/W (belonging to VH0)
compatibility area
system memory
ローカルPCIツリー用 MMIO領域
memory-mapped I/O (MMIO)
PF
VF1
VF2
VF3
VF4
VFk-1
VFk
Addr(VH0, PF)
Addr(VH0, VF1)
Addr(VH0, VF2)
Addr(VH0, VF3)
Addr(VH0, VF4)
Addr(VH0, VFk-1)
Addr(VH0, VFk)
Addr(VH1, VF1)
Addr(VH1, VF2)
Addr(VHn, VFk-1)
Addr(VHn, VFk)
12-1
12-n
12-M
max#1
max#n
max#0

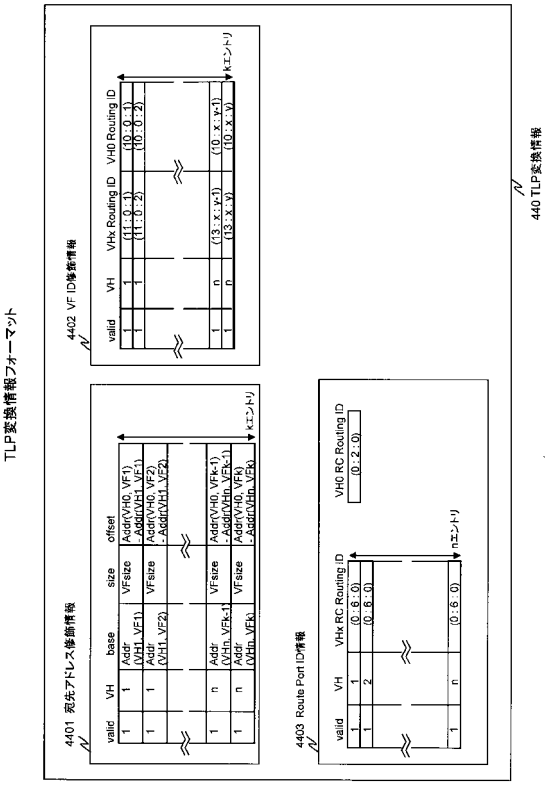
【図５】


Routing ID関係図
Blade#1 (belonging to VH1)
Blade#n (belonging to VHn)
PCI-M H/W (belonging to VH0)
ID for upstream port bridge
ID for downstream port bridge
ローカルPCIツリー用 Routing ID
MRS兼用 Routing ID
DEVアクセス用 Routing ID
ID for PF
ID for VF1
ID for VF2
ID for VF3
ID for VF4
ID for VFk-1
ID for VFk
13-1
13-n
13-M
凡例 Routing ID表記
Fun#
Dev#
Bus#

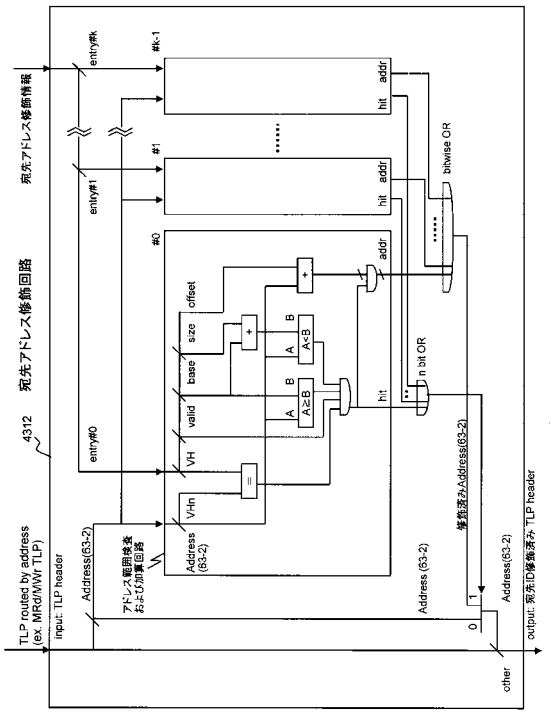
【図６】


宛先情報変換回路
outbound TLP from internal MR link
431
宛先情報変換
4311
TLP type decoder & multiplexer
TLP routed by address (ex. MRd/MWr TLP)
TLP routed by ID (ex. Configuration, Completion TLP)
other TLP (ex. vendor-specific Message)
4312
宛先アドレス修飾回路
4313
宛先ID修飾回路
4314
arbitor
発行元ID変換へ
440
TLP変換情報
4401
宛先アドレス修飾情報
4402
VF ID修飾情報
4403
Route Port ID情報

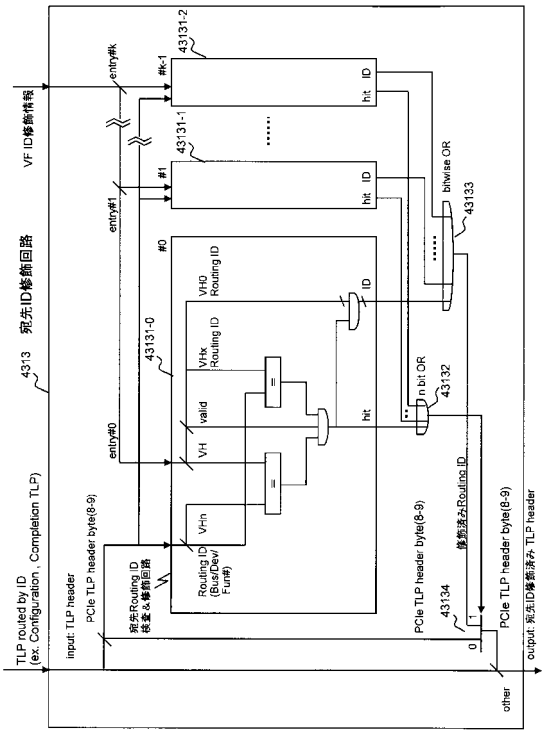
【図７】


TLP変換情報フォーマット
4401 宛先アドレス修飾情報
valid
VH
base
size
offset
VFsize
Addr(VH0, VF1)
Addr(VH0, VF2)
Addr(VH0, VFk-1)
Addr(VH0, VFk)
Addr(VH1, VF1)
Addr(VH1, VF2)
Addr(VHn, VFk-1)
Addr(VHn, VFk)
kエントリ
4402 VF ID修飾情報
VHx Routing ID
VH0 Routing ID
(11:0:1)
(11:0:2)
(13:x:y-1)
(13:x:y)
(10:0:1)
(10:0:2)
(10:x:y-1)
(10:x:y)
4403 Route Port ID情報
VHx RC Routing ID
VH0 RC Routing ID
(0:6:0)
(0:2:0)
nエントリ
440 TLP変換情報

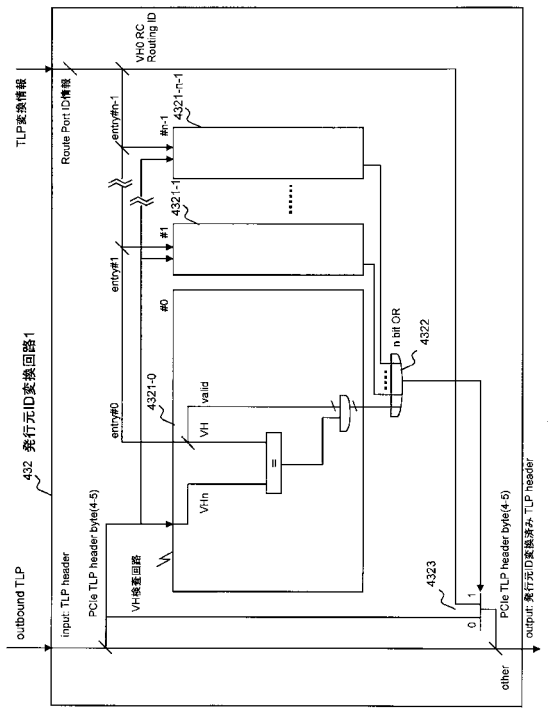
【図８】


宛先アドレス修飾回路
4312
TLP routed by address
(ex. MRd/MWr TLP)
input: TLP header
Address(63-2)
entry#0
entry#1
entry#k
#0
#1
#k-1
VH
V-Hn
valid
base
size
offset
addr
hit
n bit OR
bitwise OR
Address (63-2)
修飾済みAddress(63-2)
other
output: 宛先ID修飾済み TLP header

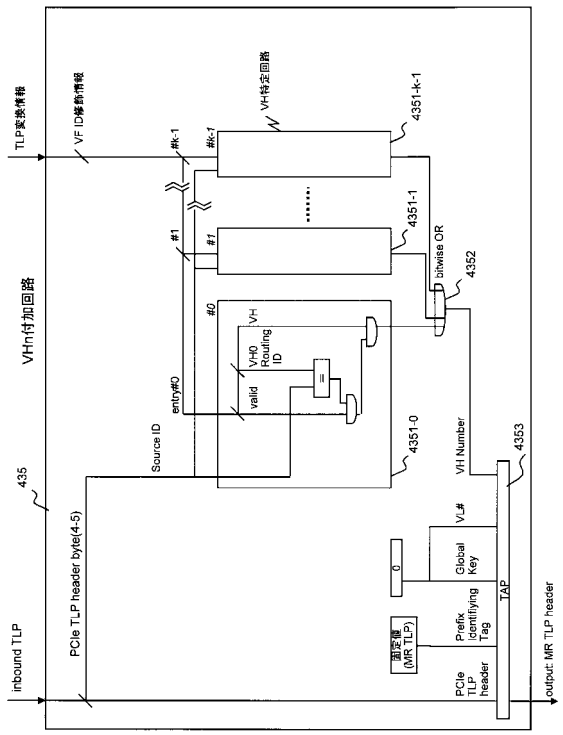
【図９】


宛先ID修飾回路
4313
TLP routed by ID
(ex. Configuration , Completion TLP)
input: TLP header
PCIe TLP header byte(8-9)
entry#0
entry#1
entry#k
#0
#1
#k-1
VH
V-Hn
valid
VH0 Routing ID
VHk Routing ID
Routing ID
(Bus/Dev/Fun#)
ID
hit
n bit OR
bitwise OR
修飾済みRouting ID
other
output: 宛先ID修飾済み TLP header

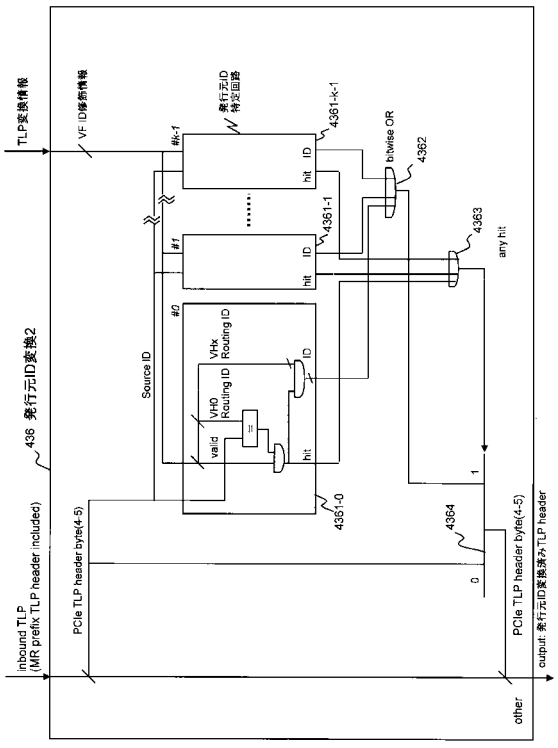
【図１０】


発行元ID変換回路1
432
outbound TLP
input: TLP header
PCIe TLP header byte(4-5)
TLP変換情報
Route Port ID情報
VH0 RC Routing ID
entry#0
entry#1
entry#n-1
#0
#1
#n-1
VH
V-Hn
valid
n bit OR
VH修飾回路
other
output: 発行元ID変換済み TLP header

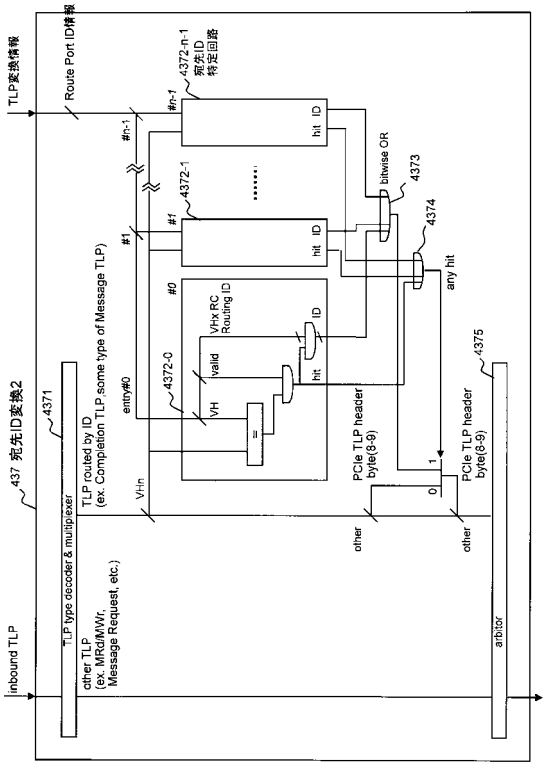
【図１１】


VHn付加回路
435
inbound TLP
PCIe TLP header byte(4-5)
TLP変換情報
VF ID修飾情報
VH特定回路
entry#0
entry#1
entry#k
#0
#1
#k-1
VH
VH0 Routing ID
valid
Source ID
bitwise OR
VH Number
V_#
Global Key
TAP
固定値 (MR TLP)
Prefix identifying Tag
PCIe TLP header
output: MR TLP header

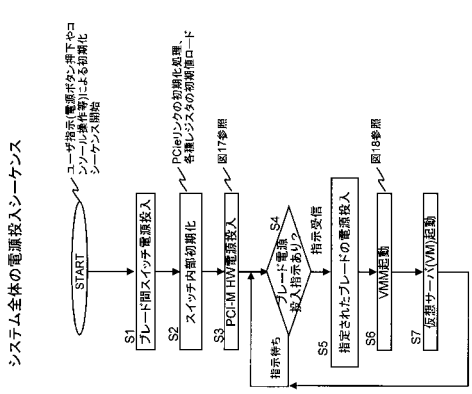
【図１２】



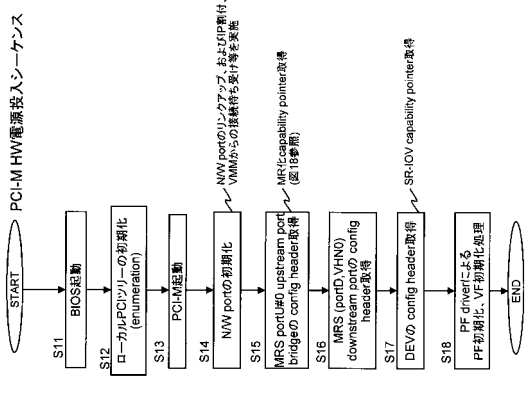
【図１３】



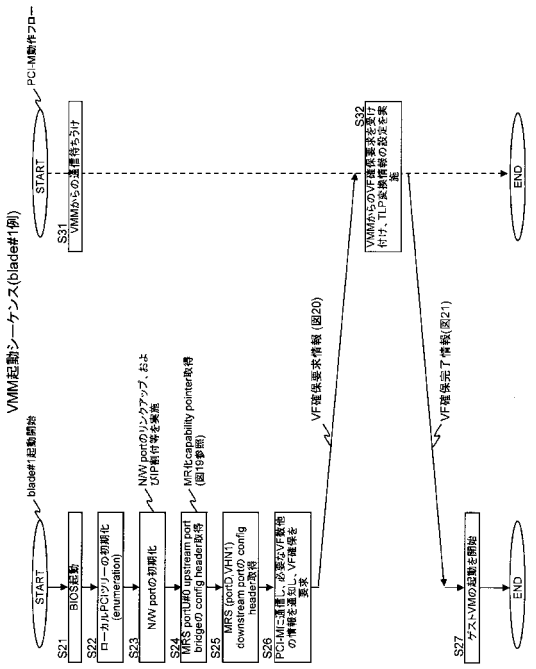
【図１６】



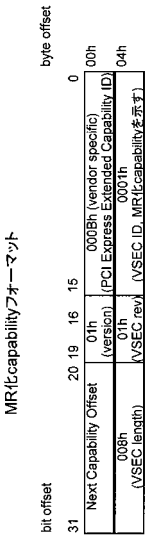
【図１７】



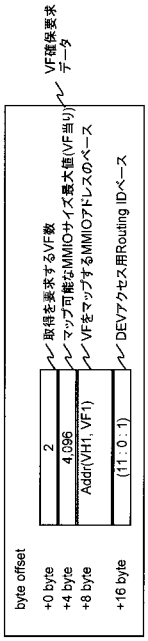
【図１８】



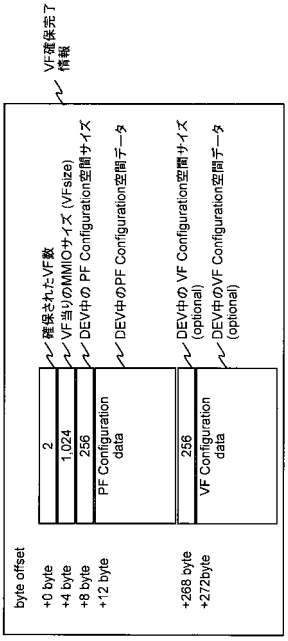
【図１９】



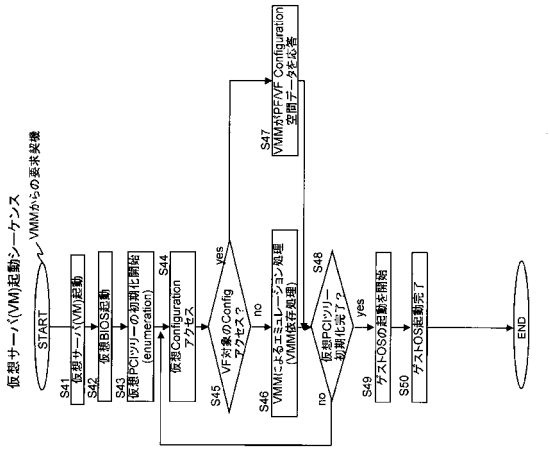
【図２０】



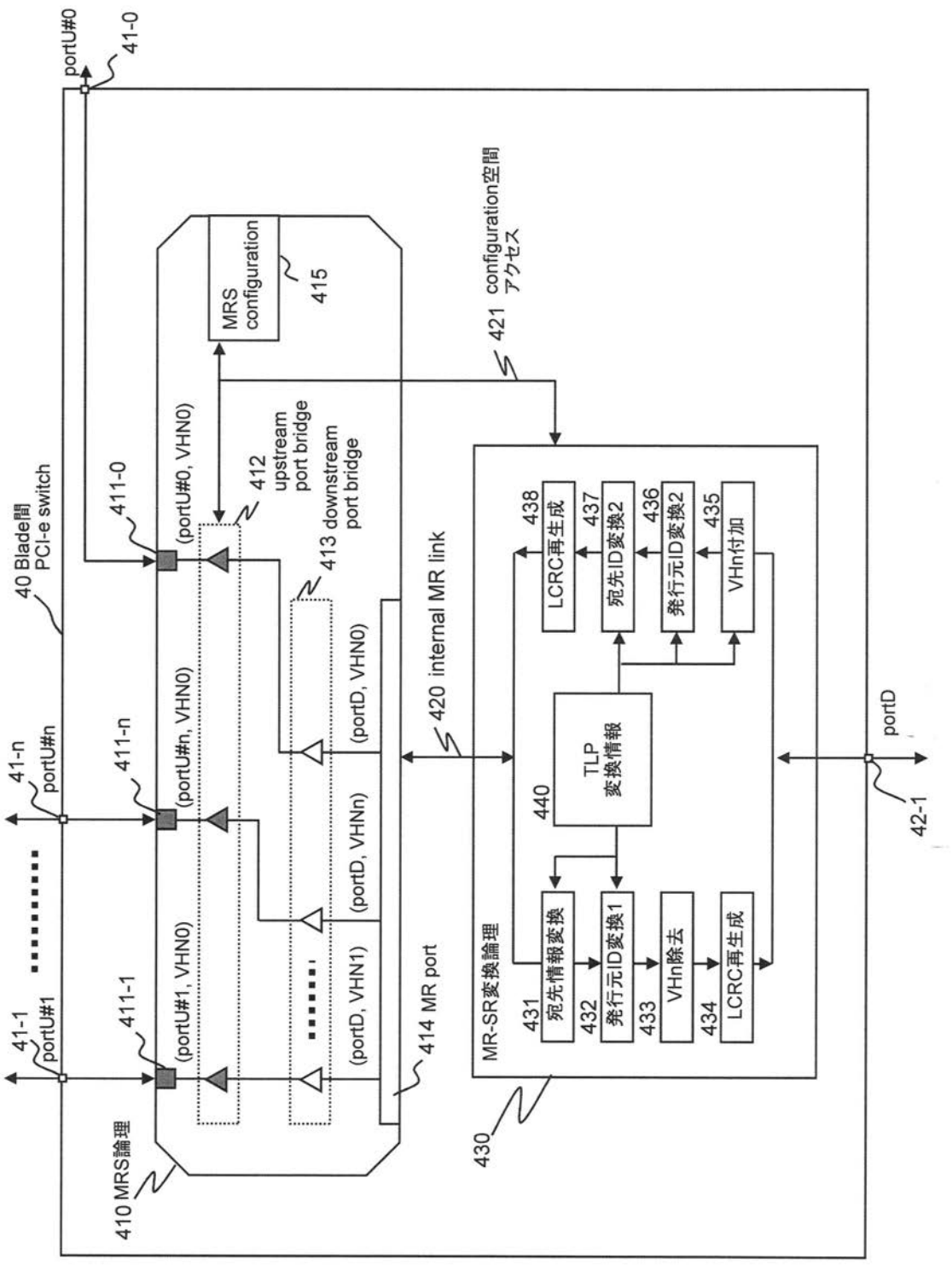
【図２１】



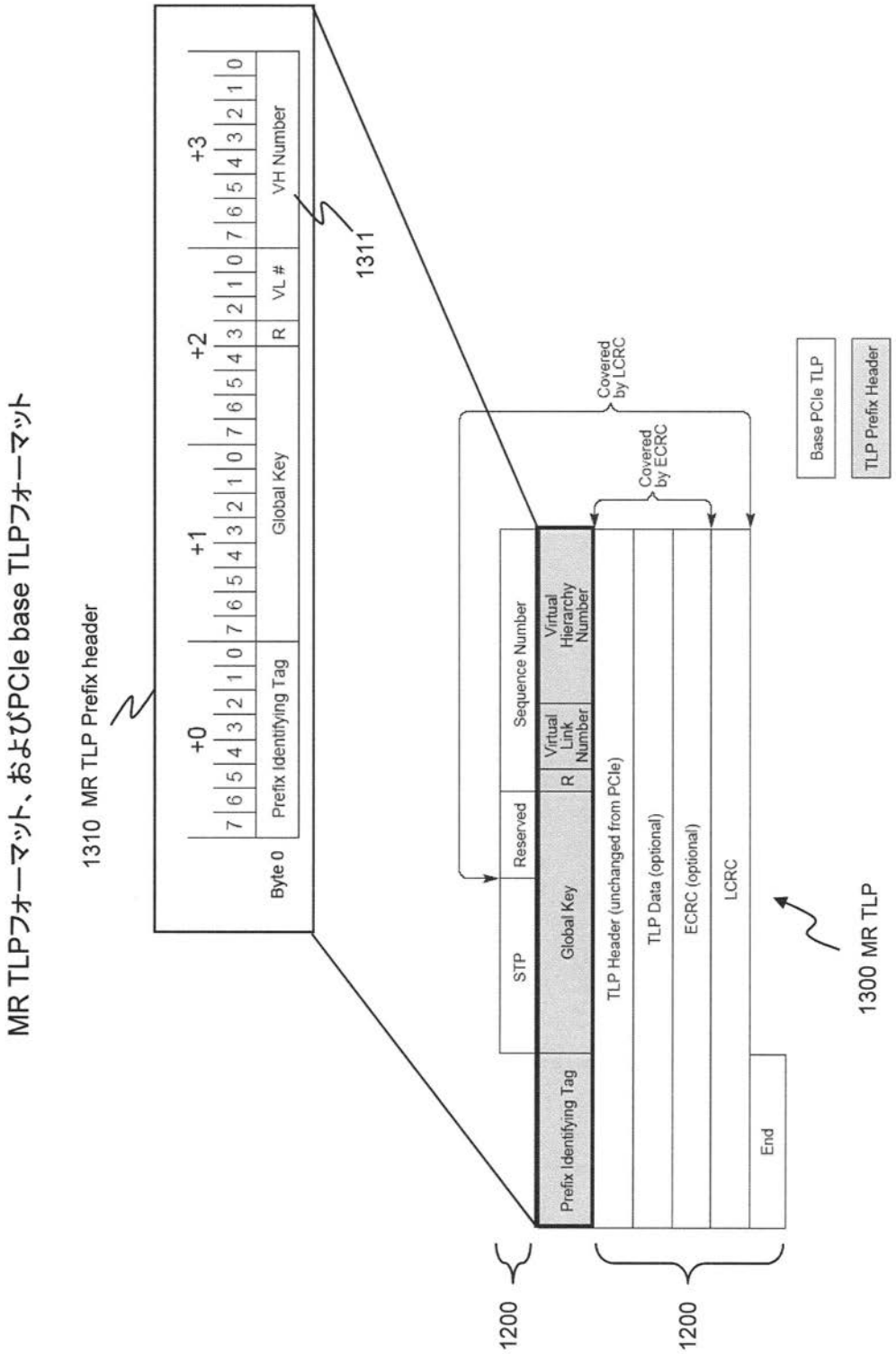
【図２２】


仮想サーバ(VM)起動シーケンス
VMMからの要求実機
START
S41
仮想サーバ(VM)起動
S42
仮想BIOS起動
S43
仮想PCIツリーの初期化開始(enumeration)
S44
仮想Configuration アクセス
S45
VF対象のConfig アクセス?
yes
no
S46
VMMによるエミュレーション処理(VMM仮想処理)
S47
VMMがPF/VF Configuration空間データを応答
S48
仮想PCIツリー初期化完了?
yes
no
S49
ゲストOSの起動を開始
S50
ゲストOS起動完了
END

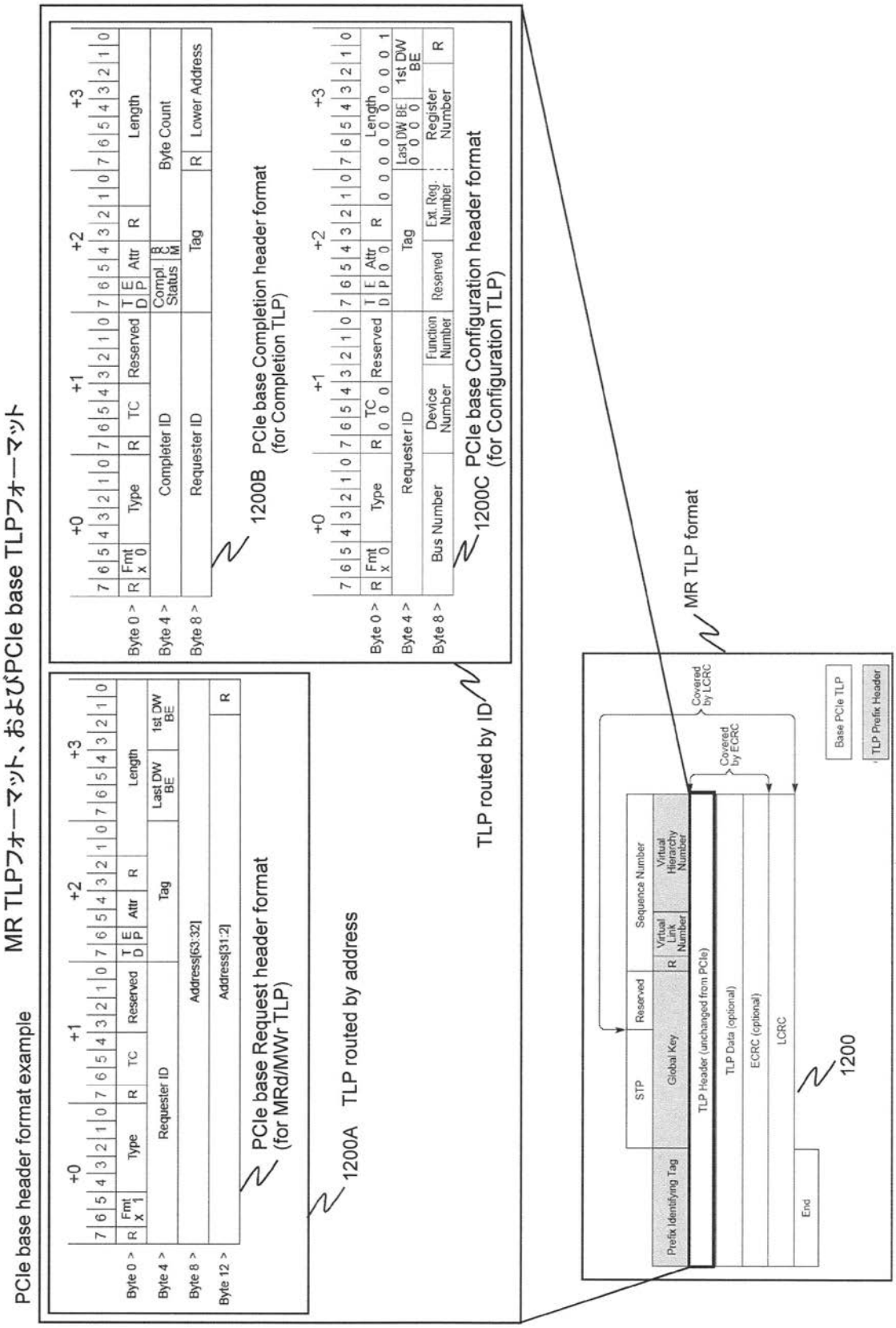
【図３】


portU#0
41-0
40 Blade間 PCI-e switch
MRS configuration
415
421 configuration空間 アクセス
411-0
(portU#0, VHN0)
412 upstream port bridge
413 downstream port bridge
438 LCRC再生成
437 宛先ID変換2
436 発行元ID変換2
435 VHn付加
420 internal MR link
41-n
portU#n
411-n
(portU#n, VHN0)
(portD, VHN0)
(portD, VHNn)
440 TLP 変換情報
portD
41-1
portU#1
411-1
(portU#1, VHN0)
(portD, VHN1)
414 MR port
MR-SR変換論理
431 宛先情報変換
432 発行元ID変換1
433 VHn除去
434 LCRC再生成
42-1
410 MRS論理
430

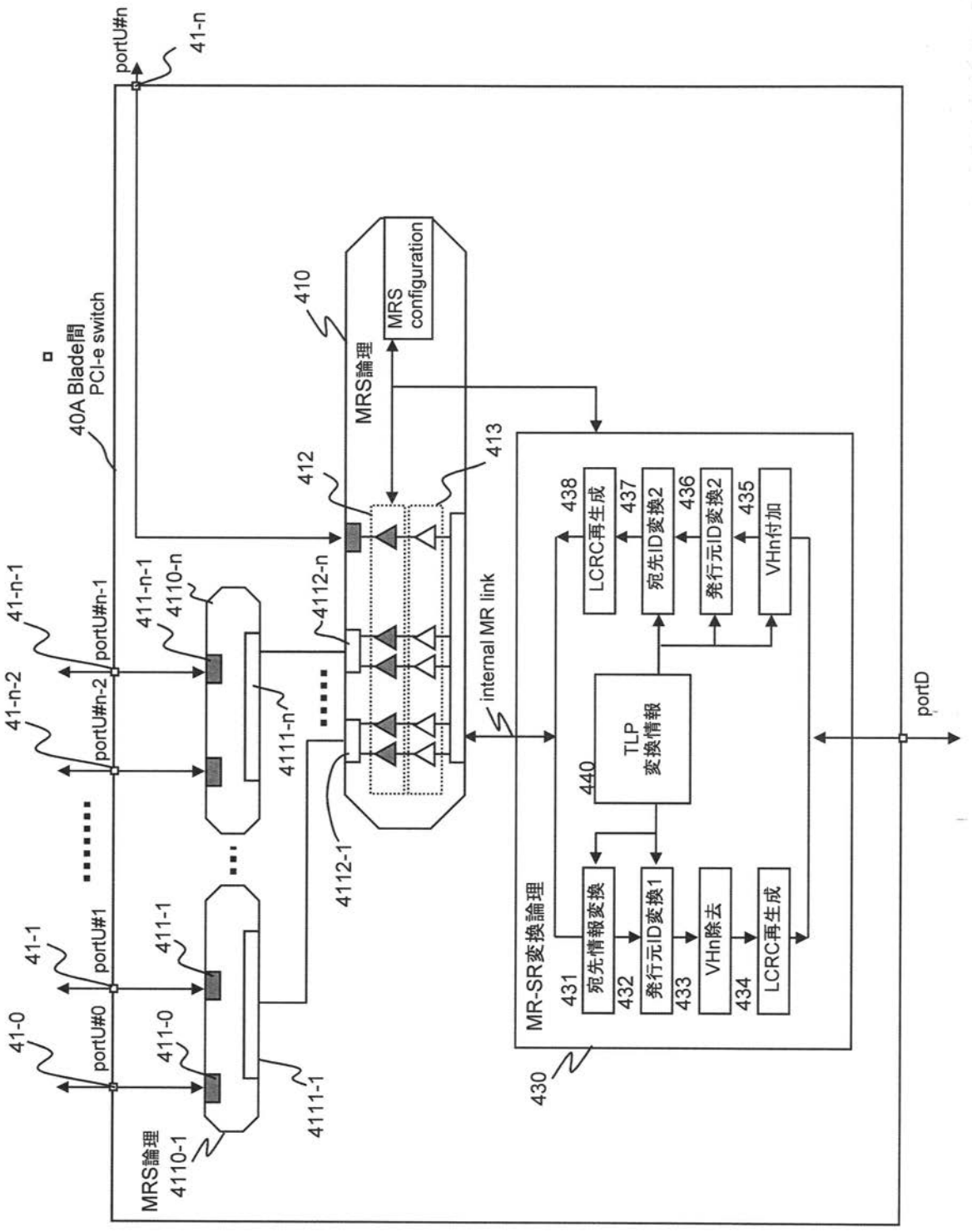
【図１４】


MR TLPフォーマット、およびPCIe base TLPフォーマット
1310 MR TLP Prefix header
Byte 0
+0
+1
+2
+3
7 6 5 4 3 2 1 0
Prefix Identifying Tag
Global Key
R
VL #
VH Number
1311
STP
Global Key
Reserved
R
Virtual Link Number
Sequence Number
Virtual Hierarchy Number
TLP Header (unchanged from PCIe)
TLP Data (optional)
ECRC (optional)
LCRC
End
Covered by LCRC
Covered by ECRC
Base PCIe TLP
TLP Prefix Header
1300 MR TLP
1200
1200

【図１５】


MR TLPフォーマット、およびPCIe base TLPフォーマット
PCIe base header format example
PCIe base Request header format (for MRd/MWr TLP)
1200A TLP routed by address
PCIe base Completion header format (for Completion TLP)
1200B
PCIe base Configuration header format (for Configuration TLP)
1200C
TLP routed by ID
MR TLP format
STP
Reserved
Sequence Number
Prefix Identifying Tag
Global Key
R
Virtual Link Number
Virtual Hierarchy Number
TLP Header (unchanged from PCIe)
TLP Data (optional)
ECRC (optional)
LCRC
End
Covered by LCRC
Covered by ECRC
Base PCIe TLP
TLP Prefix Header
1200

【図２３】


portU#n
41-n
40A Blade間
PCI-e switch
410
MRS論理
MRS configuration
412
413
internal MR link
41-n-1
portU#n-1
411-n-1
4110-n
4112-n
41-n-2
portU#n-2
4111-n
4112-1
41-1
portU#1
411-1
41-0
portU#0
411-0
4111-1
MRS論理
4110-1
MR-SR変換論理
430
431
宛先情報変換
432
発行元ID変換1
433
VHn除去
434
LCRC再生成
440
TLP
変換情報
438
LCRC再生成
437
宛先ID変換2
436
発行元ID変換2
435
VHn付加
portD

フロントページの続き

(72)発明者　上原　敬太郎
東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内

審査官　横山　佳弘

(56)参考文献　特開２００８－０７８８７（ＪＰ，Ａ）
特開２００９－３０１１６２（ＪＰ，Ａ）
国際公開第２００８／０１８４８５（WO，Ａ１）
特開２００８－０２１２５２（ＪＰ，Ａ）
特開２００７－２１９８７３（ＪＰ，Ａ）
鈴木　順　Jun　Suzuki，ＥｘｐＥｔｈｅｒ（エクスプレスイーサ）による単一ホスト仮想化対応Ｉ／Ｏのマルチホスト同時共有，第７０回（平成２０年）全国大会講演論文集（１）　アーキテクチャ　ソフトウェア科学・工学　データベースとメディア，日本，社団法人情報処理学会，２００８年　3月１３日，ｐ．１－２３～ｐ．１－２４

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１３／１０
Ｇ０６Ｆ　１３／１４
Ｇ０６Ｆ　　９／４６