

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5607825号
(P5607825)

(45) 発行日 平成26年10月15日(2014.10.15)

(24) 登録日 平成26年9月5日(2014.9.5)

(51) Int.Cl. F I
G06F 12/10 (2006.01)
 G06F 12/10 555
 G06F 12/10 505B
 G06F 12/10 551Z

請求項の数 14 (全 48 頁)

(21) 出願番号	特願2013-515721 (P2013-515721)	(73) 特許権者	390009531
(86) (22) 出願日	平成22年11月8日 (2010.11.8)		インターナショナル・ビジネス・マシーンズ・コーポレーション
(65) 公表番号	特表2013-539085 (P2013-539085A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公表日	平成25年10月17日 (2013.10.17)		アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
(86) 国際出願番号	PCT/EP2010/067024		
(87) 国際公開番号	W02011/160708	(74) 代理人	100108501
(87) 国際公開日	平成23年12月29日 (2011.12.29)		弁理士 上野 剛史
審査請求日	平成25年8月5日 (2013.8.5)	(74) 代理人	100112690
(31) 優先権主張番号	12/821, 174		弁理士 太佐 種一
(32) 優先日	平成22年6月23日 (2010.6.23)	(74) 代理人	100091568
(33) 優先権主張国	米国 (US)		弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 コンピューティング環境のシステム・メモリの管理を容易にする方法、システム、コンピュータ・プログラム

(57) 【特許請求の範囲】

【請求項1】

コンピューティング環境のシステム・メモリの管理を容易にする方法であって、
1つのアダプタをイネーブルにするために論理プロセッサ呼び出し(CLP)命令を実行すること及び前記アダプタに割り当てられる多数のダイレクト・メモリ・アクセス(DMA)アドレス空間を要求することに応答して、前記1つのアダプタのための、前記CLP命令によって要求される複数のDMAアドレス空間をイネーブルにするステップであって、前記CLP命令は前記1つのアダプタを識別する機能ハンドルを含み、前記機能ハンドルはアダプタがイネーブルにされていないインジケータを有し、前記1つのアダプタのためにイネーブルされた一つのDMAアドレス空間が、当該アドレス空間に関連付けられた変換テーブルの一組を有し、且つ、前記1つのアダプタのためのイネーブルにされた他の一つのDMAアドレス空間が、当該他のアドレス空間に関連付けられた変換テーブルの他の一組を有し、前記変換テーブルの他の一組が前記変換テーブルの前記一組と異なる、前記イネーブルにするステップと、

アドレス変換パラメータ登録操作を指定するPCI機能制御修正(MPFC)命令を実行することに応答して、前記1つのアダプタについてイネーブルにされた前記複数のDMAアドレス空間のうちの第1のDMAアドレス空間を定めるステップと、

システム・メモリにアクセスするための要求を前記1つのアダプタから受信するステップと、

前記要求内において与えられるリクエスト識別子及びアドレス空間識別子を使用して、

前記アクセスにおいて用いられる前記DMAアドレス空間を選択するステップであって、前記DMAアドレス空間は、前記1つのアダプタについてイネーブルにされた前記複数のDMAアドレス空間から選択される、前記選択するステップとを含む、前記方法。

【請求項2】

前記アドレス空間識別子は1以上のビットを含み、
前記方法は、

グループに照会するためのCLP命令を実行することに対応して、前記1つのアダプタにより提供されるアドレスのどの1以上のビットが、前記アドレス空間識別子の前記1以上のビットであるかを判断するステップ

をさらに含む、請求項1に記載の方法。

10

【請求項3】

前記使用することは、

前記リクエスト識別子及びアドレス空間識別子を使用して、前記1つのアダプタに関連付けられたデータ構造体内のエントリを見つけ出すこと

を含み、

前記エントリは、前記アドレス空間に関する1以上の特性を提供する、
請求項1に記載の方法。

【請求項4】

前記エントリは、前記1つのアダプタ及び前記システム・メモリに接続された入力/出力ハブのデバイス・テーブル内に配置される、請求項3に記載の方法。

20

【請求項5】

前記方法は、

1以上のアドレス変換テーブルを前記アドレス空間に関連付けるステップをさらに含む、

前記1以上のアドレス変換テーブルは第1の形式のものである、
請求項1に記載の方法。

【請求項6】

前記1つのアダプタのための別のアドレス空間を選択するステップと、

1以上の他のアドレス変換テーブルを前記別のアドレス空間に関連付けるステップとをさらに含む、

前記1以上の他のアドレス変換テーブルは第2の形式のものであり、前記第2の形式は前記第1の形式とは異なる、

請求項5に記載の方法。

30

【請求項7】

前記第1の形式はアドレス変換形式の第1の変形を含み、

前記第2の形式は前記アドレス変換形式の第2の変形を含む、

請求項6に記載の方法。

【請求項8】

前記第2の形式は、前記第1の形式とは異なるタイプのアドレス変換形式のものである、請求項6に記載の方法。

40

【請求項9】

前記1つのアダプタは1又は複数のアダプタ機能を含み、

前記要求は前記アダプタ機能から受信され、前記アダプタ機能は当該アダプタ機能に割り当てられた複数のアドレス空間を有する、

請求項1に記載の方法。

【請求項10】

前記アドレス空間識別子は前記要求内で提供されるアドレスのビットを含み、

前記ビットの第1の値は、前記リクエスト識別子と組み合わせられて第1のアドレス空間を示し、

50

前記ビットの第2の値は、前記リクエスト識別子と組み合わせられて第2のアドレス空間を示す、

請求項1に記載の方法。

【請求項11】

前記アドレス空間識別子は、前記要求において提供される前記アドレスの1以上のビットを含む、請求項1に記載の方法。

【請求項12】

前記1つのアダプタから別の要求を受信するステップと、
前記別の要求において提供される別のリクエスト識別子及び別のアドレス空間識別子を使用して、別のアドレス空間を選択するステップと

をさらに含み、

前記アドレス空間は、当該アドレス空間に関連付けられた第1のアドレス変換形式を有し、

前記別のアドレス空間は、当該別のアドレス空間に関連付けられた第2のアドレス変換形式を有し、

前記第1のアドレス変換形式は前記第2のアドレス変換形式とは異なる、

請求項1に記載の方法。

【請求項13】

請求項1～12のいずれか一項に記載の方法の各ステップを実行するように適合された手段を備えているシステム。

【請求項14】

コンピュータに、請求項1～12のいずれか一項に記載の方法の各ステップを実行させるコンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、コンピューティング環境においてシステム・メモリを管理することに関し、具体的には、システム・メモリ内のアドレス空間の提供を容易にすること、および、必要に応じて、システム・メモリにアクセスする際に使用可能なアドレス変換テーブルを提供することに関する。

【背景技術】

【0002】

システム・メモリには、読み出しおよび書き込み要求によりアクセスすることができる。これらの要求は、中央演算処理装置、およびアダプタを含むコンピューティング環境の種々のコンポーネントによってもたらされ得る。各々の要求は、システム・メモリにアクセスするのに使用するためのアドレスを含む。しかしながら、このアドレスは、典型的には、システム・メモリ内の物理的位置と1対1対応を有していない。従って、アドレス変換が行われる。

【0003】

アドレス変換は、システム・メモリにアクセスするのに直接使用できない1つの形式で提供されるアドレスを、システム・メモリ内の物理的位置にアクセスするのに直接使用できる別の形式に変換するために用いられる。例えば、中央演算処理装置により提供される要求内に含まれる仮想アドレスは、システム・メモリにおける実アドレスまたは絶対アドレスに変換される。更に別の例として、アダプタからの要求内に提供されるPeripheral Component Interconnect (PCI) アドレスを、システム・メモリにおける絶対アドレスに変換することができる。

【0004】

アドレス変換を行うために、1以上のアドレス変換テーブルが用いられる。このテーブルは階層構造で構成され、要求内に提供されるアドレス・ビットを用いて、最上位レベルのテーブル内のエントリがを見つけ出される。次に、そのエントリが、別の変換テーブルま

10

20

30

40

50

たはアクセスされるページそれ自体を指し示す。

【0005】

Hummel他による「Efficiently Controlling Special Memory Mapped System Accesses」という名称の2008年5月15日発行の特許文献1は、1つの実施形態において、変換テーブルの組の基底アドレスを格納するように構成された制御レジスタと、制御レジスタに結合された制御論理とを含む、入力/出力メモリ管理ユニット(IOMMU)を記載する。制御論理は、周辺機器相互接続に対応するアドレス空間のアドレス範囲内のアドレスを有する、入力/出力(I/O)デバイスが開始した要求に応答するように構成される。メモリ操作以外の1以上の操作が、アドレス範囲と関連付けられ、制御論理は、変換テーブルがアドレスから第2のアドレスへの変換を指定する場合に、アドレスをアドレス範囲外の第2のアドレスに変換するように構成され、それにより、アドレス範囲と関連した1以上の操作の代わりに、要求に応答してメモリ操作が行われる。

10

【0006】

Hummel他による「Chained Hybrid IOMMU」という名称の2007年7月19日発行の特許文献2は、1つの実施形態において、メモリ要求を変換するように構成されたI/Oメモリ管理ユニット(IOMMU)を含む、入力/出力(I/O)ノードを記載する。I/Oノードは、相互接続部に結合され、相互接続部上のトンネルとして動作するように構成され、ここで、IOMMUは、トンネルを上流方向に通過するメモリ要求を変換するように構成される。別の実施形態において、システムは、相互接続部に別の相互接続部を架橋するように構成された別のI/Oノードを含み、ここで、I/Oノードは他のI/Oノード用のトンネルとなる。

20

【0007】

Madukkarumukumana他による「Address Window Support for Direct Memory Access Translation」という名称の2006年12月21日発行の特許文献3において、装置が開示されている。この装置は、ダイレクト・メモリ・アクセス(DMA)トランザクションのために、1以上のI/Oデバイスがメモリ・デバイスにアクセスするのを容易にするための再マッピング回路を含む。この再マッピング回路は、アドレス・ウィンドウ・ベースの変換を介するI/O DMAトランザクションのために、メモリ・アドレス変換を行うための変換機構を含む。

30

【先行技術文献】

【特許文献】

【0008】

【特許文献1】米国特許出願公開第2008/0114906 A1号明細書

【特許文献2】米国特許出願公開番号第2007/0168636 A1号明細書

【特許文献3】米国特許出願公開番号第2006/0288130 A1号明細書

【特許文献4】米国特許第5551013号明細書

【特許文献5】米国特許第6009261号明細書

【特許文献6】米国特許第5574873号明細書

40

【特許文献7】米国特許第6308255号明細書

【特許文献8】米国特許第6463582号明細書

【特許文献9】米国特許第5790825号明細書

【非特許文献】

【0009】

【非特許文献1】「z/Architecture Principles of Operation」、IBM出版番号SA22-7832-07、2009年2月

【発明の概要】

【発明が解決しようとする課題】

【0010】

50

コンピューティング環境のシステム・メモリの管理を容易にする方法、システム、およびコンピュータ・プログラム製品を提供する。

【課題を解決するための手段】

【0011】

コンピューティング環境のシステム・メモリの管理を容易にするための、請求項1による方法、および対応するシステム並びにコンピュータ・プログラム製品を提供することにより、従来技術の欠点が克服され、利点をもたらされる。

【0012】

本発明の1以上の態様が、本明細書の最後にある特許請求の範囲において、例として具体的に示され、明確に特許請求されている。本発明の前記および他の目的、特徴、並びに利点は、添付図面と関連して用いられる以下の詳細な説明から明らかである。

10

【図面の簡単な説明】

【0013】

【図1】本発明の1以上の態様を組み入れ、使用するためのコンピューティング環境の1つの実施形態を示す。

【図2】本発明の一態様による、図1のシステム・メモリおよび入力/出力(I/O)ハブの更なる詳細の1つの実施形態を示す。

【図3】本発明の一態様による、1アダプタ機能に割り当てられた複数のアドレス空間の一例を示す。

【図4】本発明の一態様による、アダプタのためにDMA(ダイレクト・メモリ・アクセス)アドレス空間を登録するための論理の概要の1つの実施形態を示す。

20

【図5】本発明の一態様による、アダプタのためにDMAアドレス空間を登録することの種々の詳細の1つの実施形態を示す。

【図6】本発明の一態様による、DMA操作を処理するための論理の1つの実施形態を示す。

【図7】アドレスを変換し、ページにアクセスするために、アドレス全体を用いてアドレス変換テーブルに索引付けする際に用いられる変換のレベルの一例を示す。

【図8】本発明の一態様による、アドレス変換テーブルに索引付けする際にアドレスの一部が無視されるときに用いられる変換のレベルの一例を示す。

【図9】本発明の1以上の態様に従って使用可能な種々のCPU DAT互換形式の例を示す。

30

【図10】本発明の1以上の態様に従って使用可能な種々のI/O拡張アドレス変換形式の例を示す。

【図11】本発明の一態様に従って用いられるPCI機能制御修正(Modify PCI Function Controls)命令の1つの実施形態を示す。

【図12】本発明の一態様による、図11のPCI機能制御修正命令により用いられるフィールドの1つの実施形態を示す。

【図13】本発明の一態様による、図11のPCI機能制御修正命令により用いられる別のフィールドの1つの実施形態を示す。

【図14】本発明の一態様に従って用いられる機能情報ブロック(FIB)のコンテンツの1つの実施形態を示す。

40

【図15】本発明の一態様による、PCI機能制御修正命令の論理の概要の1つの実施形態を示す。

【図16】本発明の一態様による、PCI機能制御修正命令により指定することができるI/Oアドレス変換パラメータ登録操作(register I/O address translation parameters operation)と関連付けられた論理の一実施形態を示す。

【図17】本発明の一態様による、PCI機能制御修正命令により指定することができるI/Oアドレス変換パラメータ登録解除操作(unregister I/O address translation parameters operation)と関連付けられた論理の一実施形態を示す。

【図18】本発明の一態様に従って用いられる論理プロセッサ呼び出し(Call Logical P

50

rocessor) 命令の 1 つの実施形態を示す。

【図 19】本発明の一態様による、図 18 の論理プロセッサ呼び出し命令により用いられる要求ブロックの 1 つの実施形態を示す。

【図 20】本発明の一態様による、図 18 の論理プロセッサ呼び出し命令により提供される応答ブロックの 1 つの実施形態を示す。

【図 21】本発明の一態様による、PCI 機能をイネーブルにするための論理の 1 つの実施形態を示す。

【図 22】本発明の一態様による、グループ照会操作に関する、図 18 の論理プロセッサ呼び出し命令により用いられる要求ブロックの 1 つの実施形態を示す。

【図 23】本発明の一態様による、図 22 のグループ照会操作に関する応答ブロックの 1 つの実施形態を示す。

【図 24】本発明の 1 以上の態様を組み込むコンピュータ・プログラム製品の 1 つの実施形態を示す。

【図 25】本発明の 1 以上の態様を組み込み、用いるためのホスト・コンピュータ・システムの 1 つの実施形態を示す。

【図 26】本発明の 1 以上の態様を組み込み、用いるためのコンピュータ・システムの異なる例を示す。

【図 27】本発明の 1 以上の態様を組み込み、用いるためのコンピュータ・ネットワークを含むコンピュータ・システムの別の例を示す。

【図 28】本発明の 1 以上の態様を組み込み、用いるためのコンピュータ・システムの種々の要素の 1 つの実施形態を示す。

【図 29】本発明の 1 以上の態様を組み込み、用いるための、図 28 のコンピュータ・システムの実行ユニットの 1 つの実施形態を示す。

【図 30】本発明の 1 以上の態様を組み込み、用いるための、図 28 のコンピュータ・システムの分岐ユニットの 1 つの実施形態を示す。

【図 31】本発明の 1 以上の態様を組み込み、用いるための、図 28 のコンピュータ・システムのロード/ストア・ユニットの 1 つの実施形態を示す。

【図 32】本発明の 1 以上の態様を組み込み、用いるためのエミュレートされたホスト・コンピュータ・システムの 1 つの実施形態を示す。

【発明を実施するための形態】

【0014】

本発明の一態様によると、一つのアダプタが、複数のアドレス空間と関連付けられる。これにより、システム・メモリにアクセスする際に、一つのアダプタにより複数アドレス変換形式を使用することが可能になり、必要または所望に応じて、システム・メモリにアクセスするのに使用可能なアドレスを変換する際に、複数の組のアドレス変換テーブルを使用することが可能になる。具体的には、一例において、一つのアダプタは 1 以上のアダプタ機能を含み、複数のアドレス空間がアダプタ機能の少なくとも 1 つに割り当てられる。

【0015】

本明細書で用いられるアダプタという用語は、任意のタイプのアダプタ（例えば、ストレージ・アダプタ、ネットワーク・アダプタ、処理アダプタ、PCI アダプタ、暗号化アダプタ、他のタイプの入力/出力アダプタなど）。1 つの実施形態においては、アダプタは、1 つのアダプタ機能を含む。しかしながら、他の実施形態においては、アダプタは複数のアダプタ機能を含み得る。本発明の 1 以上の態様は、アダプタが 1 つのアダプタ機能を含む場合であっても、複数のアダプタ機能を含む場合であっても適用可能である。さらに、特に断りのない限り、本明細書で提示される例においては、アダプタは、アダプタ機能（例えば、PCI 機能）と交換可能に用いられる。

【0016】

本発明の 1 以上の態様を組み込み、用いるためのコンピューティング環境の 1 つの実施形態が、図 1 を参照して説明される。一例において、コンピューティング環境 100 は、

10

20

30

40

50

インターナショナル・ビジネス・マシーニズ・コーポレーションが提供する System z (登録商標)サーバである。System z (登録商標)は、インターナショナル・ビジネス・マシーニズ・コーポレーションが提供する z/Architecture (登録商標)に基づいている。z/Architecture (登録商標)に関する詳細は、非特許文献 1 において説明される。IBM (登録商標)、System z (登録商標)、および z/Architecture (登録商標)は、ニューヨーク州アーモンク所在のインターナショナル・ビジネス・マシーニズ・コーポレーションの登録商標である。本明細書で用いられる他の名称は、インターナショナル・ビジネス・マシーニズ・コーポレーションまたは他の会社の登録商標、商標、または製品名であり得る。

【0017】

一例において、コンピューティング環境 100 は、メモリ・コントローラ 106 を介して、システム・メモリ 104 (主メモリとしても知られる) に結合された 1 以上の中央演算処理装置 (CPU) 102 を含む。システム・メモリ 104 にアクセスするために、中央演算処理装置 102 は、システム・メモリにアクセスするのに用いられるアドレスを含む読み出し要求または書き込み要求を発行する。要求内に含まれるアドレスは、典型的には、システム・メモリにアクセスするのに直接使用できず、従って、そのアドレスは、システム・メモリにアクセスするのに直接使用できるアドレスに変換される。アドレスは、変換機構 (X L A T E) 108 を介して変換される。例えば、アドレスは、例えば、動的アドレス変換 (D A T) を用いて、仮想アドレスから実アドレスまたは絶対アドレスに変換される。

【0018】

変換されたアドレスを含む要求は、メモリ・コントローラ 106 により受信される。一例において、メモリ・コントローラ 106 は、ハードウェアからなり、システム・メモリにアクセスするために仲裁を行い、メモリの一貫性を保持するために用いられる。この仲裁は、CPU 102 から受信した要求、および、1 以上のアダプタ 110 から受信した要求に対して行われる。中央演算処理装置と同様に、アダプタは、要求をシステム・メモリ 104 に発行して、システム・メモリへのアクセスを獲得する。

【0019】

一例において、アダプタ 110 は、1 以上の P C I 機能を含む、Peripheral Component Interconnect (P C I) または P C I E x p r e s s (P C I e) アダプタである。P C I 機能は、システム・メモリへのアクセスを要求する要求を発行する。この要求は、1 以上のスイッチ (例えば、P C I e スイッチ) 114 を介して、入力/出力ハブ 112 (例えば、P C I ハブ) に送られる。一例において、入力/出力ハブは、1 以上の状態マシンを含むハードウェアからなる。

【0020】

入力/出力ハブは、例えば、スイッチから要求を受信するルート・コンプレックス 116 を含む。要求は、典型的には、変換される入力/出力アドレスを含み、従って、ルート・コンプレックスは、そのアドレスをアドレス変換および保護ユニット 118 に与える。以下でさらに詳細に説明されるように、このユニットは、例えば、I/O アドレスを、必要に応じて、システム・メモリ 104 にアクセスするのに直接使用できるアドレスに変換するハードウェア・ユニットである。

【0021】

アドレス (変換されたアドレス、または変換が必要でない場合は変換されていないアドレス) を含む、アダプタから開始された要求は、例えば I/O ・メモリ間バス 120 を介して、メモリ・コントローラ 106 に与えられる。メモリ・コントローラはその仲裁を行い、適切な時期に、変換されたアドレス (または、変換されていない場合は、初期アドレス) を有する要求をシステム・メモリに転送する。

【0022】

システム・メモリおよび入力/出力ハブに関する更なる詳細が、図 2 を参照して説明される。この実施形態においては、メモリ・コントローラは図示されない。しかしながら、

10

20

30

40

50

I/Oハブは、直接的にまたはメモリ・コントローラを介して、システム・メモリに結合され得る。一例においては、システム・メモリ104は、1以上のアドレス空間200を含む。アドレス空間は、特定のアダプタまたはアダプタ機能などのコンピューティング環境の特定のコンポーネントに割り当てられた、システム・メモリの特定の部分である。一例において、アダプタ（またはアダプタ機能）により開始されるダイレクト・メモリ・アクセス(DMA)により、アクセス空間にアクセス可能であり、従って、アドレス空間は、本明細書での例においては、DMAアドレス空間と呼ばれる。しかしながら、他の例においては、アドレス空間にアクセスするのに、ダイレクト・メモリ・アクセスは使用されない。

【0023】

さらに、一例において、システム・メモリ104は、アドレスを、システム・メモリにアクセスするのに直接使用できないものから、直接使用できるものに変換するために用いられるアドレス変換テーブル202を含む。1つの実施形態において、DMAアドレス空間に割り当てられた1以上のアドレス変換テーブルが存在し、これらの1以上のアドレス変換テーブルは、例えば、それが割り当てられるアドレス空間のサイズ、アドレス変換テーブル自体のサイズ、および/またはアクセスされるページ（または他のメモリ単位）のサイズに基づいて構成される。

【0024】

一例において、アドレス変換テーブルの階層が存在する。例えば、図2に示されるように、IOATポインタ218（以下で説明する）により指し示される第1レベルのテーブル202a（例えば、セグメント・テーブル）と、第1レベル・テーブルのエントリ206aにより指し示される第2の下位レベルのテーブル202b（例えば、ページ・テーブル）とが存在する。受信したアドレス204の1以上のビットが、テーブル202a内に索引付けして特定のエントリ206aを見つけ出すのに使用され、この特定のエントリ206aが特定の下位レベル・テーブル202bを示す。次いで、アドレス204の1以上の他のビットが、そのテーブル内の特定のエントリ206bを見つけ出すのに用いられる。この例においては、そのエントリが、正しいページを見つけ出すのに用いられるアドレスを提供し、アドレス204内の付加的なビットが、データ転送を行うためにページ内の特定の記憶位置(location)208を見つけ出すのに用いられる。つまり、エントリ206b内のアドレス、および、受信したPCIアドレス204の選択されたビットを用いて、システム・メモリにアクセスするのに直接使用できるアドレスを提供する。例えば、直接使用できるアドレスは、エントリ206b内のアドレスの上位ビット（例えば、4kページの例においては、ビット63:12）、および、受信したPCIアドレスからの選択された下位ビット（例えば、4kページにおいては、ビット11:0）の連結から形成される。

【0025】

本発明の一態様によると、複数のアドレス空間を、特定のアダプタ（または、アダプタ機能）などの特定のコンポーネントに割り当てることができる。例えば、図3に示されるように、システム・メモリ104の2つ以上のアドレス空間200a...200nが、アダプタ機能220aに割り当てられる。この例では、2つのアドレス空間が示されるが、他の例においては、2つより多くのアドレス空間が割り当てられる。特定のアダプタ機能に複数アドレス空間を割り当てることにより、オペレーティング・システムがDMAアドレス空間を分離するのが可能になる。例えば、1つのアドレス空間を制御情報およびキュー（例えば、SCSI制御データ・ブロック）のために使用し、1つのアドレス空間をデータ転送のために使用することができる（例えば、SCSIブロック）。他の例も存在する。さらに、各々のアドレス空間は、1つの大きいアドレス空間より小さいものとしてことができ、従って、改善された変換効率が改善され、より細かい粒度が保護される。

【0026】

1つの実施形態においては、アダプタ機能に割り当てられるDMAアドレス空間の各々は、異なる変換形式と関連付けられる（例えば、バイパス、フェッチなし、CPU DA

10

20

30

40

50

T 互換性、I/O 拡張アドレス変換（以下で説明される）など）。さらに、変換形式が変換テーブルを使用する場合には、1 以上のアドレス変換テーブルの組 250a - 250n が割り当てられる。1 以上のアドレス変換テーブルの各組は、特定の形式のものである（例えば、CPU DAT 互換形式、または I/O 拡張アドレス変換形式）。変換テーブルの 1 つの組についての形式は、変換テーブルの別の組と同じであってもよく、またはこれとは異なるものであってもよい。

【0027】

一例においては、オペレーティング・システムが、1 以上の DMA アドレス空間を特定のアダプタに割り当てる。この割り当ては登録プロセスを介して行われ、この登録プロセスは、そのアダプタに関する 1 以上のデバイス・テーブル・エントリ 210 の初期化（図 2）を引き起こす（例えば、信頼できるソフトウェアを介して）。以下でより詳細に説明されるように、登録プロセスはまた、アドレス空間識別子（例えば、PCI アドレスの 1 以上のビット）を各アドレス空間に関連させる。

10

【0028】

各々のデバイス・テーブル・エントリは、I/O ハブ 112 内に配置されたデバイス・テーブル 211 内に配置される。例えば、デバイス・テーブル 211 は、I/O ハブのアドレス変換および保護ユニット内に配置される。

【0029】

一例において、デバイス・テーブル・エントリ (DTE) 210 は、以下のような多数のフィールドを含む。：即ち、

20

形式 212：このフィールドは、例えば、アドレス変換テーブルの上位レベル・テーブルのアドレス変換形式を含む、種々の情報を表示するための複数のビットを含む。アドレス変換形式は、テーブルのレベル（例えば、上記の例では、第 1 レベル・テーブル）、並びに、システム・メモリにアクセスする際に直接使用可能なアドレスの提供において用いられる選択されたアドレス変換形式（変換形式としても知られる）を表示する（例えば、CPU DAT 互換形式、I/O 拡張アドレス変換形式、バイパス形式、フェッチなし形式、など）。

ページ・サイズ 213：このフィールドは、アクセスされるページ（または他のメモリ単位）のサイズを表示する。

PCI 基底アドレス 214 および PCI 制限 216：これらの値は、DMA アドレス空間を定めるのに用いられる範囲を提供し、受信したアドレス（例えば、PCI アドレス）が有効であることを検証する。

30

IOAT（入力/出力アドレス変換）ポインタ 218：このフィールドは、DMA アドレス空間に用いられる最上位レベルのアドレス変換テーブルへのポインタを含む。

イネーブル 219：このフィールドは、DTE がイネーブルにされているかどうかを示す。

キー 221：システム・メモリにおいて DMA 操作を行うときに、ストレージ保護のために用いられるストレージ・キー。

【0030】

他の実施形態においては、DTE はより多くの、より少ない、または異なる情報を含むことができる。

40

【0031】

本発明の一態様によると、アドレス空間毎に 1 つのデバイス・テーブル・エントリが存在し、従って、アダプタ（またはアダプタ機能）毎に複数デバイス・テーブル・エントリが存在し得る。1 つの実施形態においては、特定の変換に用いられるデバイス・テーブル・エントリが、リクエスト識別子 (RID)（および/またはアドレスの一部）、およびアドレス空間識別子を用いて見つけ出される。リクエスト ID（例えば、例を挙げればバス番号、デバイス番号、および機能番号を指定する 16 ビットの値）が、アダプタと関連した PCI 機能 220 により発行された要求内に含まれる。アドレス空間識別子は、要求内に含まれる I/O アドレスの 1 以上のビットである。アドレス空間識別子として用い

50

られる特定の1以上のビットは、前もってアドレス空間識別子として定められる。R I D およびI / Oアドレス(アドレス空間識別子を含む)を含む要求は、例えばスイッチ114を介して、例えばコンテンツ・アドレス可能メモリ(C A M)230に与えられる。C A Mを用いて索引値を与え、この索引値を用いてデバイス・テーブル211に索引付けして、特定のデバイス・テーブル・エントリ210を見つけ出す。例えば、C A Mは複数のエントリを含み、各々のエントリはデバイス・テーブルへの索引に対応する。各々のC A Mエントリは、R I Dの値と、アドレス空間識別子とを含む。受信したR I Dおよびアドレス空間識別子がC A M内のエントリに収容された値と合致する場合、対応するデバイス・テーブルの索引を用いて、デバイス・テーブル・エントリを見つけ出す。合致が存在しない場合、受信したパケットは破棄され、システム・メモリへのアクセスは行われない。(他の実施形態においては、C A Mまたは他のルックアップは必要とされず、R I Dおよびアドレス空間識別子が索引として用いられる。)

10

【0032】

その後、デバイス・テーブル・エントリ内のフィールドを用いて、もしあれば、アドレスの有効性およびアドレス変換テーブルの構成を保証する。例えば、I / Oハブのハードウェア(例えば、アドレス変換および保護ユニット)により要求内の受信アドレスをチェックして、この受信アドレスが、アドレスを提供した要求のR I Dおよびアドレス空間識別子を用いて見つけ出された、デバイス・テーブル・エントリ内に格納されたP C I基底アドレス214およびP C I制限216が定める境界内にあることを保証する。これにより、アドレスが既に登録された範囲にあり、かつ、アドレス変換テーブルが有効に構成されていることが保証される。

20

【0033】

登録プロセスの1つの実施形態が、図4 - 図5を参照して説明される。この例では、登録プロセスは、アダプタ(または、具体的にはアダプタ機能)に割り当てられるアドレス空間の各々について実施される。一例として、この論理は、オペレーティング・システムの要求に回答して、システム・メモリに結合された中央演算処理装置の1つにより実施される。

【0034】

最初に、図4を参照すると、アダプタ機能がアクセスするアドレス空間のサイズおよび位置を判断する(ステップ300)。一例において、アドレス空間のサイズは、オペレーティング・システムにより設定されたP C I基底アドレスおよびP C I制限によって決定される。オペレーティング・システムは、1以上の基準を用いて、基底値および制限値を決定する。例えば、P C IアドレスをC P U仮想アドレスに直接マッピングさせることをオペレーティング・システムが望む場合には、基底値および制限値はそのように設定される。更に別の例においては、アダプタおよび/またはオペレーティング・システム・イメージ間の更なる分離が望まれる場合、用いられるアドレスは、オーバーラップせず、互いに素なアドレス空間を与えるように選択される。記憶位置もまた、オペレーティング・システムにより指定され、これは、例えばアダプタの特性に基づいている。

30

【0035】

さらに、登録プロセスの一部として、どのアドレス変換形式が、アダプタ機能に関して登録されるかについて判断がなされる(ステップ301)。つまり、システム・メモリにアクセスする際に直接使用可能な、アダプタ機能のアドレスを提供するために、どの形式が用いられるかについての判断がなされる。

40

【0036】

1つの実施形態において、複数のアドレス変換形式が利用可能であり、その複数の形式から、オペレーティング・システムは、アダプタ機能に関する1つの形式を選択する。この選択は、例えば、アドレス空間の構成、アダプタ・タイプ等に基づいている。種々の可能な形式は、以下のものを含む。: 即ち、

(a) アドレス変換をバイパスするバイパス形式。登録を行うアダプタが、信頼できるアダプタであるときは、この形式を用いることができる。例えば、アダプタのハードウェア

50

ア設計が十分にロバストであり保護され、アドレスが破損され得ない場合、アダプタは信頼できるアダプタと考えられる。例えば、それぞれの変換および保護機構を提供する内部で開発されたアダプタ、または、信頼できるファームウェアにより管理されるアダプタは、信頼できるアダプタであると考えられることができる。

【 0 0 3 7 】

本明細書において用いられるファームウェアは、例えば、処理ユニットのマイクロコード、ミリコード、および/またはマクロコードを含む。ファームウェアは、例えば、より上位レベルのマシン・コードの実装に用いられる、ハードウェア・レベルの命令および/またはデータ構造体を含む。1つの実施形態において、ファームウェアは、例えば、典型的には、信頼できるソフトウェアを含むマイクロコード、または基礎をなすハードウェアに特有のマイクロコードとして配信される独自のコードを含み、システム・ハードウェアへのオペレーティング・システムのアクセスを制御する。

10

【 0 0 3 8 】

例えば、System z (登録商標) 上でのI/Oアダプタのネイティブ取り付けの場合は、I/Oアドレス変換(I/OAT)を用いて、アダプタによるシステム・メモリのDMAアクセスの保護および分離をもたらす。しかしながら、上述のものを含めて、この追加レベルの保護を必要としないアダプタのクラスが存在する。従って、これらのアダプタについては、バイパス形式を選択することができる。

(b) いずれの変換テーブルもフェッチすることなく、アダプタからの初期要求内に含まれるアドレスが使用可能な、フェッチなし(no fetch)形式。メモリが連続しているとき、ページ・サイズが既知であるとき、および、アドレスが、システム・メモリからのいずれの変換テーブルのフェッチも必要とされない制約された領域(例えば、4kページまたは1Mページ)のためのものであるときに、この形式を選択することができる。システム・メモリにアクセスするために使用可能なアドレス(即ち、フェッチなし形式が選択されたときに、結果として得られるアドレス)は、I/OATポインタのアドレスから得られる。例えば、4kページ・サイズの場合、PCIアドレスの下位ビット(例えば、ビット11:0)は、I/OATポインタの上位52ビットと連結されて、システム・メモリにアクセスする際に使用可能な結果物としてのアドレスを得る。

20

(c) I/Oアドレスを変換するために用いられる変換テーブルが、CPU DAT変換のために用いられる変換テーブルと互換性がある、CPU DAT互換形式。つまり、既にCPUの動的アドレス変換のために用いられているものと同様であり、かつ、これと互換性のあるアドレス変換テーブルが用いられる。これにより、これらのタイプのテーブルの使用に精通しているオペレーティング・システムにとって使用が容易になり、CPUとI/Oアダプタとの間でテーブルを共有することが可能になり、そのページング可能ゲストのDMA空間を管理する際に特定のオペレーティング・システム(例えば、z/VM(登録商標))の効率がもたらされる。図9を参照して以下でさらに詳細に説明されるように、使用可能な種々のCPU DAT互換形式が存在する。

30

(d) 拡張されたアドレス変換テーブルがI/Oアドレス変換のために用いられる、I/O拡張アドレス変換形式。この形式では、アドレス変換テーブルは、I/O操作専用であり、そのサイズは、典型的にCPUアドレス変換に用いられるものより大きくすることができる。例えば、1Mまたはそれより大きいページ・テーブルおよび/または他の変換テーブルが存在し得る。さらに、ページ・テーブルを含む異なるレベルの変換テーブルのサイズは、互いに異なるものであってもよく、それらがページ自体とは異なるものであってもよい。従来のサイズを増大させることは、バスのトランザクションを低減させ、また、I/O変換のキャッシングを改善する助けとなる。ページ・テーブルおよび他の変換テーブルのサイズ、並びにページのサイズは、幾つのレベルの変換が必要とされるかを決定する。異なるI/O拡張アドレス変換形式の例は、図10を参照して、以下でさらに詳細に説明される。

40

【 0 0 3 9 】

その後、必要に応じて、そのDMAアドレス空間をカバーするように、1以上のアドレ

50

ス変換テーブルが作成される(ステップ302)。一例において、作成することは、テーブルを構築することおよびテーブル・エントリ内に適切なアドレスを置くことを含む。一例として、変換テーブルの1つは、512個の64ビット・エントリを有する4kページ・テーブルであり、各々のエントリは、割り当てられたアドレス空間と互換性のある4kページ・アドレスを含む。

【0040】

その後、図5を参照してさらに詳細に説明されるように、そのアダプタ(またはアダプタ機能)に関するDMAアドレス空間を登録する(ステップ304)。この例では、アダプタ毎に1つのPCI機能があり、従って、アダプタ毎に1つのリクエストIDがあると仮定される。この論理は、例えば、オペレーティング・システムの要求にตอบสนองして、システム・メモリに結合された中央演算処理装置により実施される。

10

【0041】

最初に、1つの実施形態において、アダプタのリクエストIDおよびアドレス空間識別子に対応する利用可能なデバイス・テーブル・エントリを選択する(ステップ310)。つまり、リクエストIDおよびアドレス空間識別子を用いて、デバイス・テーブル・エントリを見つけ出す。1つの実施形態においては、中央演算処理装置の1つのファームウェアが、アドレスのどのビットがアドレス空間識別子を表すかを判断し、この情報が、登録(そのCPUまたは別のCPU上で実行することができる)を要求するオペレーティング・システムに提供され、オペレーティング・システムはこの情報を用いてデバイス・テーブル・エントリを選択する。

20

【0042】

付加的に、デバイス・テーブル・エントリ内に、PCI基底アドレスおよびPCI制限を格納する(ステップ312)。さらに、もしあれば、最上位レベルのアドレス変換テーブルの形式が、デバイス・テーブル・エントリの形式フィールド内に格納される(ステップ314)。例えば、形式フィールドは複数のビットを含み、これらのビットの1つまたは複数、最上位レベルのテーブルの形式および選択されたアドレス変換形式(例えば、セグメント・レベル、CPU DAT互換形式)を表示する。更に別の実施形態においては、1以上のビットは最上位レベルを表示し、他の1以上のビットは判断された変換形式(例えば、バイパス形式、フェッチなし形式、特定のCPU DAT互換形式、特定のI/O拡張アドレス変換形式等)を示す。

30

【0043】

付加的に、デバイス・テーブル・エントリ内に、最上位レベルのアドレス変換テーブル(または、フェッチなしの場合はページ)を指し示すのに用いられる入力/出力アドレス変換(I/OAT)ポインタが格納される(ステップ316)。これで登録プロセスが完了する。

【0044】

登録を実施することに対応して、もしあれば、DMAアドレス空間および対応するアドレス変換テーブル、並びにデバイス・テーブル・エントリを使用するための準備が整う。システム・メモリにアクセスするための、アダプタなどのリクエストが発行する要求の処理に関する詳細が、図6を参照して説明される。以下で説明される処理は、I/Oハブにより実施される。一例において、論理を実施するのは、I/Oハブのアドレス変換および保護ユニットである。

40

【0045】

1つの実施形態において、最初に、入力/出力ハブにおいて、DMA要求を受信する(ステップ400)。例えば、PCI機能が要求を発行し、この要求は、例えばPCIスイッチを介してPCIハブに転送される。要求内のリクエストIDおよびアドレス空間識別子(要求内のI/Oアドレスの1以上のビットである)を用いて、適切なデバイス・テーブル・エントリを見つけ出す(ステップ402)。例えば、CAMは、どのビットがアドレス空間識別子として指定されるかを知っており、これらのビットおよびRIDを用いて、適切なデバイス・テーブル・エントリを選択するためのデバイス・テーブルへの索引を

50

作成する。

【 0 0 4 6 】

その後、デバイス・テーブル・エントリが有効であるかどうかについて判断がなされる（問い合わせ 4 0 4）一例において、エントリ自体の中の有効性ビット（validity bit）をチェックすることにより、有効性を判断する。このビットは、例えば、オペレーティング・システムによる機能イネーブル要求の実行にตอบสนองして設定される。イネーブルにされた場合、ビットは、例えば 1（即ち、有効）に設定される。他の場合には、ビットはゼロのままである（即ち、無効）。更に別の例において、登録プロセスが完了したときに、ビットを設定することができる。

【 0 0 4 7 】

デバイス・テーブル・エントリが無効である場合は、エラーが提示される（ステップ 4 0 5）。他の場合には、要求内に与えられた P C I アドレスが、デバイス・テーブル・エントリ内に格納された P C I 基底アドレスより小さいかどうかについて、更なる判断がなされる（問い合わせ 4 0 6）。小さい場合には、アドレスは有効範囲外にあり、エラーが与えられる（ステップ 4 0 7）。しかしながら、P C I アドレスが基底アドレスより大きいかまたはこれと等しい場合には、P C I アドレスが、デバイス・テーブル・エントリ内の P C I 制限値より大きいかどうかについての別の判断がなされる（問い合わせ 4 0 8）。P C I アドレスが制限より大きい場合には、アドレスが有効範囲外にあるので、今度の場合もエラーが提示される（ステップ 4 0 9）。しかしながら、アドレスが有効範囲内にある場合には、処理が続行する。

【 0 0 4 8 】

一例において、デバイス・テーブル・エントリ内で指定されたアドレス変換形式がバイパス変換を示すかどうかについての判断がなされる（ステップ 4 1 0）。バイパス変換を示す場合、いずれの変換エントリもフェッチすることなく、アドレスが、I / O バスを通じて直接メモリ・コントローラに渡され、メモリにアクセスする。I / O ハブは処理を続行し、そのアドレスにおけるデータのフェッチ / ストアをイネーブルにする（ステップ 4 2 6）。

【 0 0 4 9 】

問い合わせ 4 1 0 に戻ると、形式がバイパスを示していない場合、いずれのアドレス変換テーブルのフェッチも必要とすることなく、形式が、I O A T ポインタに基づくメモリに直接アクセスする能力を示すかどうかについての更なる問い合わせがなされる（問い合わせ 4 1 2）。フェッチなしが示される場合、結果として得られるアドレスは、I O A T ポインタから得られ、システム・メモリからのアドレス変換テーブルのフェッチは必要とされない（ステップ 4 1 4）。結果として得られるアドレスは、メモリ・コントローラに送られ、ページおよびページ内の特定のエントリを見つけ出すために用いられる。例えば、ページ・サイズが 4 k である場合、ビット 1 1 : 0 が、I O A T ポインタからのオフセットとして用いられる。I / O ハブは処理を続行し、そのページ・エントリにおけるデータのフェッチ / ストアをイネーブルにする（ステップ 4 2 6）。

【 0 0 5 0 】

問い合わせ 4 1 2 に戻ると、一方で、変換テーブルの使用が必要である場合、デバイス・テーブル・エントリ内で提供される形式を用いて、変換テーブルのタイプ（例えば、C P U D A T 互換形式、または I / O 拡張アドレス変換形式）を決定し、かつ、アドレス変換に用いられるようにアドレス内の P C I アドレス・ビットを決定する（ステップ 4 1 6）。例えば、形式が、以下で説明される 4 k ページおよび 4 k アドレス変換テーブルを用いる I / O 拡張アドレス変換形式を示し、上位レベルのテーブルが 4 k ページの第 1 レベル・テーブルである場合、アドレスのビット 2 9 : 2 1 が第 1 レベル・テーブルへの索引付けに用いられ、ビット 2 0 : 1 2 がページ・テーブルへの索引付けに用いられ、ビット 1 1 : 0 がページへの索引付けに用いられる。用いられるビットは、所与のサイズのページまたはテーブルへの索引付けに何個のビットが必要であるかによって決まる。例えば、バイト・レベルのアドレッシングを用いる 4 k ページの場合、4 0 9 6 バイトをアドレ

10

20

30

40

50

ス指定するために12ビットが用いられ、各々が8バイトの512個のエントリを有する4kページ・テーブルの場合は、512個のエントリをアドレス指定するために9ビットが用いられる等である。

【0051】

次に、PCIハブが適切なアドレス変換テーブル・エントリをフェッチする(ステップ418)。例えば、最初に、デバイス・テーブル・エントリのIOATポインタを用いて、最上位レベルの変換テーブルを見つけ出す。次いで、アドレスのビット(有効性のために用いられ、変換のためには用いられない上位ビットの後のビット、例えば、上記の例におけるビット29:21)を用いて、そのテーブル内の特定のエントリを見つけ出す。

【0052】

次いで、例えば、デバイス・テーブル・エントリ内に与えられた形式に基づいて、見つかったアドレス変換エントリが正しい形式を有するかどうかについての判断がなされる(問い合わせ420)。例えば、デバイス・テーブル・エントリ内の形式を、アドレス変換エントリ内に示された形式と比較する。等しい場合は、デバイス・テーブル・エントリ内の形式は有効である。等しくない場合は、エラーが示され(ステップ422)、他の場合には、処理は、これが処理される最後のテーブルであるかどうかについての判断を続行する(問い合わせ424)。つまり、実アドレスまたは絶対アドレスを取得するために必要とされる他のアドレス変換テーブルが存在するかどうか、または、最下位レベル・テーブル・エントリが見つかったかどうかについての判断がなされる。この判断は、既に処理されたテーブルの与えられた形式およびサイズに基づいてなされる。これが最後のテーブルでない場合、処理はステップ418を続行する。他の場合には、I/Oハブが処理を続行し、変換されたアドレスにおけるデータのフェッチまたは格納をイネーブルにする(ステップ426)。一例において、I/Oハブは、変換されたアドレスをメモリ・コントローラに転送し、メモリ・コントローラはそのアドレスを用いて、変換されたアドレスが指定するDMA位置においてデータをフェッチまたは格納する。

【0053】

1つの実施形態において、変換のレベル数、従って、変換を実施するのに必要とされるフェッチ数が低減される。このことは、例えば、変換中にアドレスの上位のビットを無視し、下位のビットのみを用いて、例えばアダプタに割り当てられたDMAアドレス空間のサイズに基づく変換テーブルをトラバースすることによって達成することができる。完全アドレスに対する部分アドレスの使用が、以下の例においてさらに示される。

【0054】

最初に図7を参照すると、アドレス全体がアドレス変換/メモリ・アクセスに用いられる一例が示される。この従来技術では、ページ・テーブルを含む6つのレベルの変換テーブルが必要とされる。最上位レベルのテーブル(例えば、この例では、第5レベル・テーブル)の開始部が、IOATポインタにより指し示され、次いで、PCIアドレスのビットを用いて、テーブル内のエントリを見つけ出す。各々の変換テーブル・エントリは、下位のレベルの変換テーブルの開始部、またはページを指し示す(例えば、第5レベル・テーブル内のエントリが第4レベル・テーブルの開始部を指し示す等)。

【0055】

この例では、DMAアドレス空間(DMAAS)のサイズは6Mであり、各々のテーブルは4kバイトであり、最大で512個の8バイト・エントリを有する(アドレスのサイズに基づいた128個のエントリのみをサポートする、第5テーブルを除いて)。アドレスは、例えば、64ビット:FFFF C000 0009 C600である。第5レベル・テーブルの開始部が、IOATポインタにより指し示され、PCIアドレスのビット63:57を第5レベル・テーブルへの索引付けに用いて、第4レベル・テーブルの開始部を見つけ出し、PCIアドレスのビット56:48を第4レベル・テーブルへの索引付けに用いて、第3レベル・テーブルの開始部を見つけ出し、ビット47:39を第3レベル・テーブルへの索引付けに用いて、第2レベル・テーブルの開始部を見つけ出し、ビット38:30を第2レベル・テーブルへの索引付けに用いて、第1レベル・テーブルの開

10

20

30

40

50

始部を見つけ出し、ビット 29 : 21 を第 1 レベル・テーブルへの索引付けに用いて、ページ・テーブルの開始部を見つけ出し、ビット 20 : 12 をページ・テーブルの索引付けに用いて、ページの開始部を見つけ出し、そして、ビット 11 : 0 を用いて、4 k ページ内のエントリを見つけ出す。このように、この例では、アドレス・ビットの全てが、変換 / アクセスに用いられる。

【 0 0 5 6 】

これは、アドレス空間が同じサイズであり（例えば 6 M）、アドレスも同じであるが、変換中にアドレス・ビットの一部が変換技術により無視される図 8 の例とは対照的である。この例では、変換においてアドレスのビット 63 : 30 が無視される。I O A T ポインタが第 1 レベル・テーブルの開始部を指し示し、P C I アドレスのビット 29 : 21 を第 1 レベル・テーブルへの索引付けに用いて、ページ・テーブルの開始部を見つけ出し、ビット 20 : 12 を適切なページ・テーブルへの索引付けに用いて、ページの開始部を見つけ出し、ビット 11 : 0 を 4 k ページへの索引付けに用いる。

10

【 0 0 5 7 】

示されるように、第 1 レベル・テーブル 500 は、各々がアドレスを 3 つのページ・テーブル 504 の 1 つを提供する、3 つのエントリ 502 を含む。必要とされるページ・テーブルの数、従って、他のレベルのテーブルの数は、例えば、D M A アドレス空間のサイズ、変換テーブルの数、および / またはページのサイズによって決まる。この例では、D M A アドレス空間は 6 M であり、各々のページ・テーブルは 4 k であり、最大で 512 個のエントリを有する。従って、各々のページ・テーブルは、最大 2 M のメモリまでマッピングすることができる（4 k % 512 エントリ）。従って、6 M のアドレス空間には、3 つのページ・テーブルが必要とされる。第 1 レベル・テーブルは、各ページ・テーブルにつき 1 つの、3 つのエントリを保持することができ、従って、この例では、それ以上のレベルのアドレス変換テーブルは必要とされない。

20

【 0 0 5 8 】

付加的に、上述されたように、異なる形式のアドレス変換テーブルをアドレス変換に用いることができ、形式内には変形が存在し得る。例えば、種々の C P U D A T 互換形式が存在してもよく、その例が図 9 を参照して説明される。図示されるように、例として、1 つの C P U D A T 互換形式は 4 k ページの C P U D A T 互換形式 550 であり、別の C P U D A T 互換形式は 1 M ページの C P U D A T 互換形式 552 である。示されるビットの数は、そのページまたはテーブルに索引付けする（または他の方法でそのページまたはテーブル内のエントリを見つけ出す）ために使用されるアドレス・ビットの数である。例えば、P C I アドレスの 12 ビット 554 は、4 k ページ 556 へのバイト・オフセットとして使用され、8 ビット 558 は、ページ・テーブル 560 への索引として使用され、11 ビット 562 は、セグメント・テーブル 564 への索引として使用される等である。示されるアドレス変換テーブルの下に配置されているのは、そのアドレス変換テーブルがサポートするアドレス空間の最大サイズである。例えば、ページ・テーブル 560 は、1 M の D M A アドレス空間をサポートし、セグメント・テーブル 564 は、2 G の D M A アドレス空間をサポートする等。この図および図 10 において、K = キロバイト、M = メガバイト、G = ギガバイト、T = テラバイト、P = ペタバイト、および E = エクサバイトである。

30

40

【 0 0 5 9 】

示されるように、ページのサイズが増大するにつれて、変換テーブルのレベル数は減少する。例えば、4 k ページ 556 の場合はページ・テーブルが必要であるが、1 M ページの場合、ページ・テーブルは必要ない。他の例および変形が可能である。

【 0 0 6 0 】

I / O 拡張アドレス変換形式の種々の例が、図 10 に示される。例えば、以下の形式が図示される。: 即ち、4 k ページの 4 k アドレス変換テーブル 570、4 k ページの 1 M アドレス変換テーブル 572、および、1 M ページの 1 M アドレス変換テーブル 574 である。C P U D A T 互換形式と同様に、列挙されたビット数は、特定のテーブル内のエ

50

ントリを見つけ出すために使用されるビットのものである。例えば、参照番号576において、12ビットは、4kページへのオフセットである。同様に、参照番号578において、9ビットは、I/Oページ・テーブルへの索引として使用される。このI/Oページ・テーブルは、2Mのサイズを有するDMAアドレス空間を可能にする。多数の他の例が存在する。

【0061】

本明細書で説明されるように、1つのアドレス空間の1つのアドレス変換形式を、別のアドレス空間のアドレス変換形式とは異なるものとして行うことができる。例えば、形式は、異なるタイプのものとしてもよく（例えば、1つのアドレス空間についてはバイパス形式とし、別のアドレス空間についてはCPU DAT互換形式とする、または1つのアドレス空間についてはCPU DAT互換形式とし、別のアドレス空間についてはI/O拡張アドレス変換形式とする、または他のいずれかの組み合わせとする）、または、特定のタイプの形式の変形（例えば、1つのアドレス空間については4kページのCPU DAT互換形式、別のアドレス空間については1MページのCPU DAT互換形式とする；1つのアドレス空間については4kページの4kテーブルのI/O拡張アドレス変換とし、別のアドレス空間については4kページの1MテーブルのI/O拡張アドレス変換形式とする等）としてもよい。さらに、アドレス空間は同じ形式のものであってもよく、異なる（或いは同じ長さ）であってよく、依然として、一意のアドレス識別子によって識別することができる。サポートされるアドレス空間の数は2つより大きくてもよく、サポートされる数は実装によって決まる。

【0062】

1つの特定の実装において、アダプタへのDMAアドレス空間の登録を実施するために、PCI機能制御修正(MPFC)命令と呼ばれる命令が用いられる。例えば、オペレーティング・システムは、どのアドレス変換形式を使用したいと望むかを決定し、その形式についてのアドレス変換テーブルを構築し、次いで、命令のオペランドとして含まれるその形式を有するMPFC命令を発行する。一例において、命令の形式および他のオペランドは、命令のオペランドである、機能情報ブロック(以下で説明する)内に含まれる。次いで、機能情報ブロックを用いて、DTEを更新し、1つの実施形態においては随意的に、アダプタの操作パラメータを含む機能テーブル・エントリ(FTE)を更新する。

【0063】

この命令、および特に登録プロセスに関する詳細の1つの実施形態が、図11 - 図17を参照して説明される。図11を参照すると、PCI機能制御修正命令600が、例えばそのPCI機能制御修正命令を示すオペコード602と、操作パラメータを確立するアダプタ機能に関する種々の情報が含まれている記憶位置を指定する第1のフィールド604と、PCI機能情報ブロック(FIB)がフェッチされる記憶位置を指定する第2のフィールド606とを含む。フィールド1および2により示される記憶位置のコンテンツが、以下でさらに説明される。

【0064】

1つの実施形態において、フィールド1は、種々の情報を含む汎用レジスタを示す。図12に示されるように、このレジスタのコンテンツは、例えば、これに代わって修正命令を実施するアダプタ機能のハンドルを識別する機能ハンドル610と、機能ハンドルにより示されるアダプタ機能と関連したシステム・メモリ内のアドレス空間を示すアドレス空間612と、そのアダプタ機能に関して実施される操作を指定する操作制御614と、命令が所定のコードにより完了したときに、その命令に関する状態を提供する状態616とを含む。

【0065】

1つの実施形態において、機能ハンドルは、例えば、ハンドルがイネーブルにされるかどうかを示すイネーブル・インジケータと、アダプタ機能を識別する機能番号(これは静的識別子であり、機能テーブルへの索引付けのために用いることができる)と、この機能ハンドルの特定のインスタンスを指定するインスタンス番号とを含む。各々のアダプタ機

10

20

30

40

50

能について1つの機能ハンドルが存在し、これは、機能テーブル内の機能テーブル・エントリ (F T E) を見つけ出すのに用いられる。各々の機能テーブル・エントリは、そのアダプタ機能と関連した操作パラメータおよび/または他の情報を含む。一例として、機能テーブル・エントリは、以下のものを含む。：即ち、

インスタンス番号：このフィールドは、機能テーブル・エントリと関連したアダプタ機能ハンドルの特定のインスタンスを示す。

デバイス・テーブル・エントリ (D T E) 索引 1 . . . n : 1以上のデバイス・テーブル・エントリ索引が存在することができ、各々の索引は、デバイス・テーブル・エントリ (D T E) を見つけ出すためのデバイス・テーブルへの索引である。アダプタ機能毎に1以上のテーブル・エントリが存在し、各々のエントリは、アダプタ機能の要求 (例えばDMA要求、MSI要求) を処理するために使用される情報、およびアダプタ機能と関連した要求 (例えばPCI命令) に関する情報を含む、そのアダプタ機能と関連した情報を含む。各々のデバイス・テーブル・エントリは、アダプタ機能に割り当てられたシステム・メモリ内の1アドレス空間と関連付けられる。アダプタ機能は、該アダプタ機能に割り当てられた1以上のアドレス空間をシステム・メモリ内に有することができる。

10

ビジー状態インジケータ：このフィールドは、アダプタ機能がビジーであるかどうかを示す。

永続エラー状態インジケータ：このフィールドは、アダプタ機能が永続エラー状態にあるかどうかを示す。

回復開始インジケータ：このフィールドは、アダプタ機能に関して回復が開始されたかどうかを示す。

20

許可インジケータ：このフィールドは、アダプタ機能を制御しようとしているオペレーティング・システムが、そのようなことを行う権限をもっているかどうかを示す。

イネーブル・インジケータ：このフィールドは、アダプタ機能がイネーブルにされるかどうかを示す (例えば、1 = イネーブルにされる、0 = ディスエーブルにされる)。

リクエスト識別子 (R I D) : これは、アダプタ機能の識別子であり、例えば、バス番号、デバイス番号、および機能番号を含む。

【 0 0 6 6 】

一例において、このフィールドは、アダプタ機能の構成空間 (configuration space) のアクセスのために用いられる。(アダプタのメモリを、例えば構成空間、I/O空間、および/または1以上のメモリ空間を含むアドレス空間として定めることができる。) 一例において、オペレーティング・システム (または他の構成) によりアダプタ機能に対して発行された命令内で構成空間を指定することにより、構成空間にアクセスすることができる。構成空間内へのオフセット、および、R I D を含む適切な機能テーブル・エントリを見つけ出すのに用いられる機能ハンドルが、命令内で指定される。ファームウェアが命令を受信し、それが構成空間に関するものであると判断する。従って、ファームウェアは、R I D を用いてI/Oハブへの要求を生成し、I/Oハブは、アダプタにアクセスするための要求を作成する。アダプタ機能の記憶位置はR I D に基づいており、オフセットは、アダプタ機能の構成空間へのオフセットを指定する。

30

【 0 0 6 7 】

基底アドレス・レジスタ (B A R) (1 乃至 n) : このフィールドは、B A R₀ - B A R_n として示される複数の符号なし整数を含み、これらの符号なし整数は、もともと指定されたアダプタ機能と関連付けられており、その値もまた、アダプタ機能と関連付けられた基底アドレス・レジスタ内に格納される。各々のB A R は、アダプタ機能内のメモリ空間またはI/O空間の開始アドレスを指定し、同じくアドレス空間のタイプ、即ち、例としてそれが64ビット若しくは32ビットのメモリ空間であるか、または32ビットのI/O空間であるかも示す。

40

【 0 0 6 8 】

一例において、B A R は、アダプタ機能のメモリ空間および/またはI/O空間へのアクセスに用いられる。例えば、アダプタ機能にアクセスするための命令内に与えられるオ

50

フセットが、命令内で指定されるアドレス空間と関連した基底アドレス・レジスタ内の値に加算され、アダプタ機能にアクセスするのに用いられるアドレスを得る。命令内に与えられるアドレス空間識別子は、アクセスされるアダプタ機能内のアドレス空間、および用いられる対応するBARを識別する。

サイズ1 . . . n : このフィールドは、 $SIZE_0 - SIZE_n$ として示される複数の符号なし整数を含む。

サイズ・フィールドの値は、非ゼロであるとき、各々のアドレス空間のサイズを表し、各々のエントリは前述のBARに対応する。

【0069】

BARおよびサイズに関する更なる詳細が以下で説明される。

1 . アダプタ機能に関してBARが実装されない場合、BARフィールドおよび対応するサイズ・フィールドの両方ともゼロとして格納される。

2 . BARフィールドがI/Oアドレス空間または32ビット・メモリ・アドレス空間のいずれかを表す場合、対応するサイズ・フィールドは非ゼロであり、アドレス空間のサイズを表す。

3 . BARフィールドが64ビット・メモリ・アドレス空間を表す場合、

a . BAR_n フィールドは、最下位アドレス・ビットを表す。

b . 次の連続した BAR_{n+1} フィールドは、最上位アドレス・ビットを表す。

c . 対応する $SIZE_n$ フィールドは非ゼロであり、アドレス空間のサイズを表す。

d . 対応する $SIZE_{n+1}$ フィールドは有意味ではなく、ゼロとして格納される。

内部経路指定情報 : この情報は、アダプタへの特定の経路指定を行うために用いられる。この情報には、例として、例えばノード、プロセッサ・チップ、およびハブ・アドレッシング情報が含まれる。

状態表示 : これは、例えば、ロード/ストア操作がブロックされるかどうか、またはアダプタがエラー状態にあるかどうかの表示、並びに他の表示を提供する。

【0070】

一例において、ビジー・インジケータ、永続エラー状態インジケータ、および回復開始インジケータは、ファームウェアが行う監視に基づいて設定される。さらに、許可インジケータが、例えばポリシーに基づいて設定され、BAR情報は、プロセッサ(例えば、プロセッサのファームウェア)によるバス・ウォーク中に発見された構成情報に基づいている。他のフィールドは、構成、初期化、および/またはイベントに基づいて設定することができる。他の実施形態においては、機能テーブル・エントリは、より多くの、より少ない、または異なる情報を含むことができる。含まれる情報は、そのアダプタ機能によりサポートされる操作、またはアダプタ機能に関してイネーブルにされる操作によって決まり得る。

【0071】

図13を参照すると、一例において、フィールド2は、関連したアダプタ機能に関する情報を含むPCI機能情報ブロック(FIB)の論理アドレス620を指定する。機能情報ブロックは、アダプタ機能と関連したデバイス・テーブル・エントリおよび/または機能テーブル・エントリ(または他の記憶位置)を更新するために用いられる。情報は、アダプタの初期化および/または構成中に、および/または特定のイベントにตอบสนองして、FIB内に格納される。

【0072】

機能情報ブロック(FIB)に関する更なる詳細が、図14を参照して説明される。1つの実施形態においては、機能情報ブロック650は、以下のフィールドを含む。 : 即ち、

形式651 : このフィールドは、FIBの形式を指定する。

インターセプト制御652 : このフィールドは、ページング可能モード・ゲストによる、特定の命令のゲスト実行が、命令のインターセプトをもたらすかどうかについて示すために用いられる。

10

20

30

40

50

エラー表示 6 5 4 : このフィールドは、ダイレクト・メモリ・アクセスおよびアダプタ割り込みに関するエラー状態表示を含む。ビットが設定されると(例えば 1)、アダプタ機能に関するダイレクト・メモリ・アクセスおよびアダプタ割り込みを実施する間、1以上のエラーが検出される。

ロード/ストアのブロック 6 5 6 : このフィールドは、ロード/ストア操作がブロックされているかどうかを表示する。

PCI機能の有効性 6 5 8 : このフィールドは、アダプタ機能に関するイネーブルメント制御を含む。ビットが設定されると(例えば 1)、アダプタ機能はI/O操作に関してイネーブルにされると考えられる。

アドレス空間の登録 6 6 0 : このフィールドは、そのアダプタ機能に関するダイレクト・メモリ・アクセスのイネーブルメント制御を含む。ビットが設定されると(例えば 1)、ダイレクト・メモリ・アクセスはイネーブルにされる。

ページ・サイズ 6 6 1 : このフィールドは、DMAによりアクセスされるページまたは他のメモリ単位のサイズを示す。

PCI基底アドレス(PBA) 6 6 2 : このフィールドは、アダプタ機能に割り当てられたシステム・メモリ内のアドレス空間についての基底アドレスである。これは、アダプタ機能が、指定されたDMAアドレス空間へのダイレクト・メモリ・アクセスのために用いることができる最下位の仮想アドレスを表す。

PCIアドレス制限(PAL) 6 6 4 : このフィールドは、アダプタ機能が、指定されたDMAアドレス空間内でアクセスすることができる最上位の仮想アドレスを表す。

入力/出力アドレス変換ポインタ(IOAT) 6 6 6 : 入力/出力アドレス変換ポインタは、PCI仮想アドレス変換により用いられるいずれかの変換テーブルの第1のものを示し、または、これは、変換の結果であるストレージ・フレームの絶対アドレスを直接指定することができる。

割り込みサブクラス(ISC) 6 6 8 : このフィールドは、アダプタ機能に関するアダプタ割り込みを提示するのに用いられる割り込みサブクラスを含む。

割り込みの数 6 7 0 : このフィールドは、アダプタ機能について受け入れた別個の割り込みコードの数を示す。このフィールドはまた、アダプタ割り込みビット・ベクトル・アドレスおよびアダプタ割り込みビット・ベクトル・オフセット・フィールドにより示されるアダプタ割り込みビット・ベクトルのサイズもビット単位で定める。

アダプタ割り込みビット・ベクトル(AIBV) 6 7 2 : このフィールドは、アダプタ機能に関するアダプタ割り込みビット・ベクトルのアドレスを指定する。このベクトルは、割り込み処理において用いられる。

アダプタ割り込みビット・ベクトル・オフセット 6 7 4 : このフィールドは、アダプタ機能に関する第1のアダプタ割り込みビット・ベクトルのオフセットを指定する。

アダプタ割り込みサマリ・ビット・アドレス(AISB) 6 7 6 : このフィールドは、割り込み処理において随意的に用いられるアダプタ割り込みサマリ・ビットを指定するアドレスを提供する。

アダプタ割り込みサマリ・ビット・オフセット 6 7 8 : このフィールドは、アダプタ割り込みサマリ・ビット・ベクトルへのオフセットを提供する。

機能測定ブロック・アドレス(FMB) 6 8 0 : このフィールドは、アダプタ機能に関する測定値を収集するのに用いられる機能測定ブロックのアドレスを提供する。

機能測定ブロック・キー 6 8 2 : このフィールドは、機能測定ブロックにアクセスするためのアクセス・キーを含む。

サマリ・ビット通知制御 6 8 4 : このフィールドは、使用中のサマリ・ビット・ベクトルが存在するかどうかを示す。

命令許可トークン 6 8 6 : このフィールドは、ページング可能ストレージ・モード・ゲストに、ホストの介入なしにPCI命令を実行する権限が付与されているかどうかを判断するのに用いられる。

一例において、z/Architecture(登録商標)においては、ページング可

10

20

30

40

50

能ゲストが、レベル2の解釈において、Start Interpretive Execution (SIE) 命令を介して解釈実行される。例えば、論理パーティション (LPAR) ハイパーバイザがSIE命令を実行して、物理的固定メモリにおける論理パーティションを開始する。z/VM (登録商標) がその論理パーティションにおけるオペレーティング・システムである場合、これがSIE命令を発行して、そのV=V (仮想) ストレージ内でゲスト (仮想) マシンを実行する。従って、LPARハイパーバイザはレベル1のSIEを使用し、z/VM (登録商標) ハイパーバイザはレベル2のSIEを用いる。および、

アドレス変換形式687:変換に用いられる最上位レベルの変換テーブルのアドレス変換に関する選択された形式を示す (例えば、最上位レベルのテーブルの表示 (例えば、セグメント・テーブル、領域第3テーブル等)、および選択された形式の表示 (例えば、CPU DAT互換形式、I/O拡張アドレス変換形式、バイパス形式、フェッチなし形式))。

【0073】

PCI機能制御修正命令内で指定された機能情報ブロックは、選択されたデバイス・テーブル・エントリ、機能テーブル・エントリ、および/または命令内で指定されたアダプタ機能と関連した他のファームウェア制御を修正するために用いられる。デバイス・テーブル・エントリ、機能テーブル・エントリ、および/または他のファームウェア制御を修正することにより、アダプタに対して特定のサービスが提供される。これらのサービスには、例えば、アダプタ割り込み、アドレス変換、エラー状態のリセット、ロード/ストアのブロックのリセット、機能測定パラメータの設定、およびインターセプト制御の設定が含まれる。

【0074】

PCI機能制御修正命令と関連した論理の1つの実施形態が、図15を参照して説明される。一例において、命令が、オペレーティング・システム (または他の構成) により発行され、オペレーティング・システムを実行するプロセッサ (例えば、ファームウェア) により実行される。本明細書での例においては、命令およびアダプタ機能は、PCIベースのものである。しかしながら、他の例においては、異なるアダプタ・アーキテクチャおよび対応する命令を用いることができる。

【0075】

一例において、オペレーティング・システムは、命令に対して、以下のオペランド: 即ち、PCI機能ハンドル、DMAアドレス空間識別子、操作制御、および、機能情報ブロックのアドレスを提供する (例えば、命令により指定される1以上のレジスタにおいて)。

【0076】

図15を参照すると、最初に、PCI機能制御修正命令を可能にするファシリティがインストールされているかどうかについての判断がなされる (問い合わせ700)。例えば、この判断は、例えば制御ブロック内に格納されたインジケータをチェックすることによりなされる。ファシリティがインストールされていない場合、例外条件が与えられる (ステップ702)。他の場合には、命令がページング可能ストレージ・モード・ゲスト (または他のゲスト) により発行されたかどうかについての判断がなされる (問い合わせ704)。肯定であれば、ホスト・オペレーティング・システムは、そのゲストに関する動作をエミュレートする (ステップ706)。

【0077】

他の場合には、オペランドの1つまたは複数位置合わせされているかどうかについての判断がなされる (問い合わせ708)。例えば、機能情報ブロックのアドレスがダブルワード境界上にあるかどうかについて判断がなされる。一例においては、これは随意的である。オペランドが位置合わせされていない場合、例外条件が与えられる (ステップ710)。

【0078】

10

20

30

40

50

他の場合には、機能情報ブロックがアクセス可能であるかどうかについて判断がなされる（問い合わせ 7 1 2）。アクセス可能でない場合、例外条件が与えられる（ステップ 7 1 4）。他の場合には、P C I 機能制御修正命令のオペランド内に提供されたハンドルがイネーブルにされているかどうかについての判断がなされる（問い合わせ 7 1 6）。一例において、この判断は、ハンドル内のイネーブル・インジケータをチェックすることによってなされる。ハンドルがイネーブルにされていない場合、例外条件が与えられる（ステップ 7 1 8）。

【 0 0 7 9 】

ハンドルがイネーブルにされている場合、ハンドルを用いて、機能テーブル・エントリを見つけ出す（ステップ 7 2 0）。つまり、ハンドルの少なくとも一部を機能テーブルへの索引として使用して、操作パラメータが確立されるアダプタ機能に対応する機能テーブル・エントリを見つけ出す。

10

【 0 0 8 0 】

機能テーブル・エントリが見つかったかどうかについての判断がなされる（問い合わせ 7 2 2）。見つかっていない場合、例外条件が与えられる（ステップ 7 2 4）。他の場合には、命令を発行する構成がゲストである場合（問い合わせ 7 2 6）、例外条件（例えば、ホストへのインターセプト）が与えられる（ステップ 7 2 8）。構成がゲストでない場合は問い合わせを無視することができ、または指定された場合、他の権限付与をチェックすることができる。

【 0 0 8 1 】

20

次いで、機能がイネーブルにされているかどうかについての判断がなされる（問い合わせ 7 3 0）。一例において、この判断は、機能テーブル・エントリ内のイネーブル・インジケータをチェックすることによりなされる。機能がイネーブルにされていない場合、例外条件が与えられる（ステップ 7 3 2）。

【 0 0 8 2 】

機能がイネーブルにされている場合、回復がアクティブであるかどうかについての判断がなされる（問い合わせ 7 3 4）。機能テーブル・エントリ内のインジケータにより判断されるように回復がアクティブである場合、例外条件が与えられる（ステップ 7 3 6）。しかしながら、回復がアクティブでない場合、機能がビジーであるかどうかについてのさらなる判断がなされる（問い合わせ 7 3 8）。この判断は、機能テーブル・エントリ内のビジー・インジケータをチェックすることによりなされる。機能がビジーである場合、ビジー条件が与えられる（ステップ 7 4 0）。ビジー条件において、命令を、ドロップせずに再試行することができる。

30

【 0 0 8 3 】

機能がビジーでない場合、機能情報ブロック形式が有効であるかどうかについてのさらなる判断がなされる（問い合わせ 7 4 2）。例えば、F I B の形式フィールドをチェックして、この形式がシステムによりサポートされているかどうかについての判断がなされる。形式が無効である場合、例外条件が与えられる（ステップ 7 4 4）。機能情報ブロックの形式が有効である場合、命令のオペランド内で指定される操作制御が有効であるかどうかについての更なる判断がなされる（問い合わせ 7 4 6）。つまり、操作制御は、この命令に関して指定された操作制御の 1 つであるかどうかである。操作制御が無効である場合、例外条件が与えられる（ステップ 7 4 8）。しかしながら、操作制御が有効である場合、処理は、指定された特定の操作制御を続行する。

40

【 0 0 8 4 】

指定することができる 1 つの操作制御は、アダプタに関するアドレス変換の制御に用いられる登録 I / O アドレス変換パラメータ操作である。この操作により、適切な F I B のパラメータから、D T E、F T E、および / または他の記憶位置内に、I / O アドレス変換に関連する P C I 機能パラメータが設定され、これは命令に対するオペランドである。これらのパラメータは、例えば、P C I 基底アドレス、P C I アドレス制限（P C I 制限または制限としても知られる）、アドレス変換形式、ページ・サイズ、および I / O アド

50

レス変換ポインタを含み、これらはこの操作に対するオペランドである。これらに加えて、開始DMAアドレス(SDMA)および終了DMAアドレス(EDMA)を含む、暗黙のオペランドが存在し、これらは、命令を実行するプロセッサがアクセス可能な記憶位置内に格納される。

【0085】

I/Oアドレス変換に関する操作パラメータを確立するための論理の1つの実施形態が、図16を参照して説明される。最初に、FIB内のPCI基底アドレスが、FIB内のPCI制限より大きいかどうかについての判断がなされる(問い合わせ800)。基底アドレスと制限との比較が、基底アドレスが制限より大きいことを示す場合、例外条件が認識される(ステップ802)。しかしながら、基底アドレスが制限より小さいかまたはこれと等しい場合、アドレス変換形式およびページ・サイズが有効であるかどうかについての更なる判断がなされる(問い合わせ804)。これらが無効である場合は、例外条件が与えられる(ステップ806)。しかしながら、これらが有効である場合、アドレス空間のサイズ(基底アドレスおよび制限に基づいた)が変換容量を上回るかどうかについての更なる判断がなされる(問い合わせ808)。一例において、アドレス空間のサイズを、より上位レベルのテーブルの形式に基づいた可能な最大アドレス変換容量と比較する。例えば、より上位レベルのテーブルがDAT互換性セグメント・テーブルである場合、最大変換容量は2Gバイトである。

10

【0086】

アドレス空間のサイズが変換容量を上回る場合、例外条件が与えられる(ステップ810)。他の場合には、基底アドレスが開始DMAアドレスより小さいかどうかについての更なる判断がなされる(問い合わせ812)。開始DMAアドレスより小さい場合、例外条件が与えられる(ステップ814)。他の場合には、アドレス制限が終了DMAアドレスより大きいかどうかについての別の判断がなされる(問い合わせ816)。終了DMAアドレスより大きい場合、例外条件が与えられる(ステップ818)。一例において、開始DMAアドレスおよび終了DMAアドレスは、システム全体のポリシーに基づいている。

20

【0087】

その後、I/Oアドレス変換を実施するために、必要であれば、十分なりソースが利用可能であるかどうかについての判断がなされる(問い合わせ820)。利用可能でない場合、例外条件が与えられる(ステップ822)。他の場合には、I/Oアドレス変換パラメータが、FTEおよびDTE内に既に登録されているかどうかについての更なる判断がなされる(問い合わせ824)。このことは、FTE/DTE内のパラメータの値をチェックすることにより判断される。例えば、FTE/DTEにおける値がゼロまたは他の規定値である場合、登録はまだ行われていない。FTEを見つけ出すためには、命令内で与えられるハンドルが用いられ、DTEを見つけ出すためには、FTE内のデバイス索引が用いられる。

30

【0088】

アドレス変換に関してアダプタ機能が既に登録されている場合、例外条件が与えられる(ステップ826)。まだ登録されていない場合、指定されたDMAアドレス空間が有効であるか(即ち、DTEがイネーブルにされたアドレス空間であるか)どうかについての判断がなされる(問い合わせ828)。有効でない場合、例外条件が与えられる(ステップ830)。全てのチェックが成功した場合、デバイス・テーブル・エントリ内に、随意的には、対応する機能テーブル・エントリ(または、他の指定された記憶位置)内に、変換パラメータが入れられる(ステップ832)。例えば、I/Oアドレス変換に関連するPCI機能パラメータが、機能情報ブロックからコピーされ、DTE/FTE内に入れられる。これらのパラメータは、例えば、PCI基底アドレス、PCIアドレス制限、変換形式、ページ・サイズ、およびI/Oアドレス変換ポインタを含む。この操作により、指定されたDMAアドレス空間へのDMAアクセスが可能になる。この操作は、アダプタ機能に関するI/Oアドレス変換をイネーブルにする。

40

50

【 0 0 8 9 】

PCI機能制御修正命令により指定することができる別の操作制御は、I/Oアドレス変換パラメータ登録解除操作であり、その一例が、図17を参照して説明される。この操作により、I/Oアドレス変換に関連する機能パラメータがゼロにリセットされる。この操作は、指定されたアドレス空間へのDMAアクセスをディスエーブルにし、そのDMAアドレス空間に関するI/O変換ルックアサイド・バッファ・エントリのページを発生させる。それにより、アドレス変換がディスエーブルにされる。

【 0 0 9 0 】

図17を参照すると、1つの実施形態において、I/Oアドレス変換パラメータが登録されていないかどうかについての判断がなされる(問い合わせ900)。一例において、この判断は、FTEまたはDTE内の適切なパラメータの値をチェックすることによりなされる。それらのフィールドがゼロであるかまたは何らかの規定値である場合、それらは登録されない。従って、例外条件が与えられる(ステップ902)。I/Oアドレス変換パラメータが登録される場合、DMAアドレス空間が有効かどうかについての更なる判断がなされる(問い合わせ904)。DMAアドレス空間が無効である場合、例外条件が与えられる(ステップ906)。DMAアドレス空間が有効である場合、デバイス・テーブル・エントリ内の変換パラメータ、随意的には対応する機能テーブル・エントリ内の変換パラメータがクリアされる(ステップ908)。

【 0 0 9 1 】

1つの実施形態において、アダプタに割り当てられるDMAアドレス空間の各々について、登録プロセスが実施される。本明細書で説明されるように、多数のアドレス空間を割り当てることができ、1つの特定の実装においては、割り当てられるアドレス空間の数は、論理プロセッサ呼び出し命令イネーブル機能により示される。

【 0 0 9 2 】

この命令の1つの実施形態を図18に示す。図示されるように、一例において、論理プロセッサ呼び出し命令1000は、それが論理プロセッサ呼び出し命令であることを示す操作コード1002と、コマンドについての表示1004とを含む。一例において、この表示は、実施されるコマンドを説明する要求ブロックのアドレスである。このような要求ブロックの1つの実施形態を図19に示す。

【 0 0 9 3 】

図19に示されるように、一例において、要求ブロック1020は、例えば、要求ブロックの長さを示す長さフィールド1022; PCI機能設定(set PCI function)コマンドを示すコマンド・フィールド1024; イネーブル機能またはディスエーブル機能に与えられるハンドルであるPCI機能ハンドル1026; イネーブル操作またはディスエーブル操作を指示するのに用いられる操作コード1028; および特定のPCI機能と関連したアドレス空間の要求数を示すDMAアドレス空間(DMAAS)の数1030といった、多数のパラメータを含む。他の実施形態においては、より多くの、より少ない、または異なる情報を含ませることができる。例えば、命令がページング可能ストレージ・モード・ゲストのホストにより発行される仮想環境において、ゲストの識別が与えられる。

【 0 0 9 4 】

論理プロセッサ呼び出し命令の発行および処理にตอบสนองして、応答ブロックが戻され、応答ブロック内に含まれる情報は、実施される操作によって決まる。応答ブロックの1つの実施形態を図20に示す。一例において、応答ブロック1050は、応答ブロックの長さを表示する長さフィールド1052と、コマンドの状態を示す応答コード1054と、PCI機能を識別するPCI機能ハンドル1056とを含む。イネーブル・コマンドにตอบสนองして、PCI機能ハンドルは、PCI機能のイネーブルにされたハンドルとなる。さらに、ディスエーブル操作の完了時に、PCI機能ハンドルは、将来のイネーブル機能によりイネーブルにすることができる汎用ハンドルとなる。

【 0 0 9 5 】

PCI機能をイネーブルにするための論理の1つの実施形態が、図21を参照して説明

10

20

30

40

50

される。一例において、この論理は、コマンドが P C I 機能設定コマンドに設定され、操作コードがイネーブル機能に設定される、論理プロセッサ呼び出し命令の発行に回答して開始される。この論理は、この論理を実施する権限が付与されたオペレーティング・システムまたはオペレーティング・システムのデバイス・ドライバが命令を発行するのに回答して実施される。他の実施形態においては、論理プロセッサ呼び出し命令を使用することなく、この論理を実施することができる。

【 0 0 9 6 】

図 2 1 を参照すると、最初に、論理プロセッサ呼び出し命令の要求ブロック内で与えられるハンドルが有効なハンドルであるかどうかについての判断がなされる（問い合わせ 1 1 0 0）。つまり、ハンドルが機能テーブル内の有効なエントリを指し示すか、またはハンドルは有効なエントリの範囲外にある（例えば、ハンドルの機能番号部分はインストール済みの機能を指し示す）。ハンドルが既知のものでない場合、ハンドルが認識されないことを示す、対応する応答コードが与えられる（ステップ 1 1 0 2）。しかしながら、ハンドルが既知のものである場合、ハンドルがイネーブルにされているかどうかについての更なる問い合わせがなされる（問い合わせ 1 1 0 4）。この判断は、P C I 機能ハンドル内のイネーブル・インジケータをチェックすることにより行われる。表示がセットされており、ハンドルがイネーブルにされていることを示す表示が設定されている場合、それを示す応答コードが戻される（ステップ 1 1 0 6）。

【 0 0 9 7 】

しかしながら、ハンドルが既知のものであり、かつ、イネーブルにされていない場合（即ち、イネーブルメントの場合に有効である）、P C I 機能に割り当てられるアドレス空間の要求数が最大値より大きいかどうかについての更なる判断がなされる（問い合わせ 1 1 0 8）。この判断を行うために、要求ブロック内に指定されている D M A アドレス空間の数を最大値（一例においては、ポリシーに基づいて与えられる）と比較する。アドレス空間の数が最大値より大きい場合、D M A アドレス空間についての無効値を示す応答コードが与えられる（ステップ 1 1 1 0）。他の場合、要求数のアドレス空間が利用可能なものであるかどうかについての判断がなされる（問い合わせ 1 1 1 2）。この判断は、要求数のアドレス空間について利用可能なデバイス・テーブル・エントリが存在するかどうかをチェックすることにより行われる。要求数のアドレス空間が利用可能ではない場合、リソースが不十分であることを示す応答コードが戻される（ステップ 1 1 1 4）。他の場合、処理は続行し、P C I 機能をイネーブルにする。

【 0 0 9 8 】

与えられたハンドルを用いて、機能テーブル・エントリを見つけ出す（ステップ 1 1 1 6）。例えば、ハンドルの 1 以上の指定ビットを機能テーブルへの索引として用いて、特定の機能テーブル・エントリを見つけ出す。適切な機能テーブル・エントリを見つけ出すことに回答して、機能がイネーブルにされているかどうかについての判断がなされる（問い合わせ 1 1 1 8）。この判断は、機能テーブル・エントリ内のイネーブル・インジケータをチェックすることにより行われる。機能が既にイネーブルにされている（即ち、インジケータが 1 にセットされている）場合、P C I 機能が既に要求された状態にあることを示す応答コードが戻される（ステップ 1 1 2 0）。

【 0 0 9 9 】

機能がまだイネーブルにされていない場合、処理は、機能が永続エラー状態にあるかどうかの判断を続行する（問い合わせ 1 1 2 2）。機能テーブル・エントリ内の永続エラー状態インジケータが、それが永続エラー状態にあることを示している場合、そのことを示す応答コードが戻される（ステップ 1 1 2 4）。しかしながら、機能が永続エラー状態ではない場合、機能に関してエラー回復が開始されているかどうかについての更なる判断がなされる（問い合わせ 1 1 2 6）。機能テーブル・エントリ内の回復開始インジケータが設定されている場合は、回復が開始されていることを示す応答コードが与えられる（ステップ 1 1 2 8）。他の場合には、P C I 機能がビジーであるかどうかについての更なる問い合わせがなされる（問い合わせ 1 1 3 0）。同様に、機能テーブル・エントリ内のビジ

10

20

30

40

50

ー・インジケータのチェックが、P C I機能がビジーであることを示している場合には、そのような表示が与えられる（ステップ1132）。しかしながら、P C I機能が永続エラー状態にはなく、回復が開始されておらず、かつ、機能がビジーではない場合は、オペレーティング・システムがこのP C I機能をイネーブルにすることが許可されているかどうかについての更なる問い合わせがなされる（問い合わせ1134）。P C I機能が機能テーブル・エントリの許可インジケータに基づいて許可されない場合は、不正な動作を示す応答コードが与えられる（ステップ1136）。しかしながら、全てのテストに成功裏に合格した場合には、このP C I機能に関し、利用可能ないずれかのD T Eが存在するかどうかについての更なる判断がなされる（問い合わせ1138）。例として、D T Eの利用可能性の判断は、I / Oハブにおける現在イネーブルにされていないD T Eに基づくことができる。付加的に、ポリシーを適用して、所与のオペレーティング・システムまたは論理パーティションが利用可能なD T Eの数をさらに限定することができる。アダプタにアクセスできる、あらゆる利用可能なD T Eを割り当てることができる。利用可能なD T Eが存在しない場合は、要求されたD T Eの1つまたは複数が利用可能でないことを示す応答コードが戻される（ステップ1140）。

10

【0100】

D T Eが利用可能である場合は、要求されたアドレス空間の数に対応する数のD T Eが割り当てられ、イネーブルにされる（ステップ1142）。一例において、イネーブルにすることは、イネーブルにされる各D T E内のイネーブル・インジケータを設定することを含む。さらに、イネーブルにすることは、この例において、C A Mを設定して各D T Eへの索引を提供することを含む。例えば、各々のD T Eについて、C A M内のエントリが、索引と共にロードされる。

20

【0101】

さらに、D T Eは、機能テーブル・エントリと関連付けられる（ステップ1144）。これには、例えば、機能テーブル・エントリ内に各々のD T E索引を含ませることが含まれる。次いで、機能は、機能テーブル・エントリ内のイネーブル・インジケータを設定することによりイネーブルされるものとしてマーク付けされる（ステップ1146）。さらに、ハンドル内のイネーブル・ビットが設定され、インスタンス番号が更新される（ステップ1148）。次いで、このイネーブルにされたハンドルが戻され（ステップ1150）、P C Iアダプタの使用が可能になる。例えば、機能をイネーブルにすることに応答して、アドレス変換のための登録および割り込みを実施することができ、P C I機能によりD M A操作を実施することができ、および/または、ロード、ストア、およびストア・ブロック命令を機能に対して発行することができる。

30

【0102】

各々のアドレス空間は、アドレス空間識別子により識別され、アドレス空間識別子は、アダプタが受信するアドレスの1以上のビットである。特定のビットが、D M Aアドレス空間マスク内で示され、このD M Aアドレス空間マスクは、C L Pのグループ照会コマンドにより取り出される。C L P命令の一例は、図18を参照して上述される。

【0103】

P C I機能グループ照会コマンドに関する要求ブロックの1つの実施形態が、図22を参照して説明される。一例において、要求ブロック1200は、以下を含む。：即ち、

40

長さフィールド1202：このフィールドは要求ブロックの長さを示す。

コマンド・コード1204：このフィールドは、クエリP C I機能グループ・コマンドを示す。

機能グループID1206：このフィールドは、属性が取得されるP C I機能グループ識別子を指定する。一例において、属性は、選択された機能に関する詳細を提供するグループ照会コマンドから取得される。

【0104】

P C I機能グループ照会コマンドによる論理プロセッサ呼び出し命令の発行および処理に応答して、応答ブロックが戻される。応答ブロックの1つの実施形態が、図23に示さ

50

れる。一例において、応答ブロック 1 2 5 0 は、次のものを含む。：即ち、
長さフィールド 1 2 5 2：このフィールドは、応答ブロックの長さを示す。

応答コード 1 2 5 4：このフィールドは、コマンドの状態を示す。

割り込み数 1 2 5 6：このフィールドは、指定された P C I 機能グループ内の各々の P C I 機能について、P C I ファシリティによりサポートされる連続した M S I ベクトル数（即ち、割り込みイベント・インジケータ）の最大数を示す。一例において、割り込み数の可能な有効値は、ゼロから 2, 0 4 8 までの範囲にある。

バージョン 1 2 5 8：このフィールドは、指定された P C I グループ識別子によって指定される P C I 機能のグループが付属する、P C I ファシリティによりサポートされる P C I 仕様のバージョンを示す。

10

フレーム 1 2 6 2：このフィールドは、I / O アドレス変換のためにサポートされるフレーム（またはページ）のサイズを示す。

ブロック更新間隔測定値 1 2 6 4：これは、P C I 機能測定ブロックが更新される、およびその時間間隔を示す値（例えばミリ秒単位での）である。

D M A アドレス空間マスク 1 2 6 6：これは D M A アドレス空間を識別するために、P C I アドレス内のどのビットが使用されるかを示すために用いられる値である。これが、サポートされている D M A アドレス空間の最大数を暗黙に定めることがある。つまり、これは、マスクにおいて 1 であるビット数に対する 2 のべき乗である。

M S I アドレス 1 2 6 8：これは、メッセージ信号割り込み要求のために用いられる値である。

20

【 0 1 0 5 】

グループ情報は、所与のシステム I / O インフラストラクチャ、並びに、ファームウェアおよび I / O ハブの能力に基づく。これは、後でクエリの処理中に取り出すために、F T E または他のいずれかの使いやすい位置に格納することができる。具体的には、グループ照会コマンドがこの情報を取り出し、それを、オペレーティング・システムにアクセス可能なコマンドの応答ブロックに格納する。

【 0 1 0 6 】

上記で詳細に説明されたのは、複数の D M A アドレス空間を各々のアダプタに、特に各々のアダプタ機能（他のアダプタ機能と P C I バスを共有する）割り当てるための能力である。アダプタまたはアダプタ機能毎に複数アドレス空間を使用することにより、必要に応じて、異なるサイズのアドレス空間の使用、異なる変換形式の使用、および/または異なるアドレス変換テーブルの使用が可能になる。複数のアドレス空間を使用することは、D T E を各々のアドレス空間と関連付けることによって達成される。D T E は、その関連したアドレス空間の特性を定める。R I D とアドレス空間識別子の組み合わせにより、適切な D T E が選択される。

30

【 0 1 0 7 】

本明細書で説明される実施形態において、アダプタは、P C I アダプタである。本明細書で用いられる P C I という用語は、これらに限定されるものではないが、P C I または P C I e を含む、P e r i p h e r a l C o m p o n e n t I n t e r c o n n e c t S p e c i a l I n t e r e s t G r o u p (P C I - S I G) により定められるような P C I ベースの仕様 (www.pcisig.com/home) に従って実装されるいずれかのアダプタを指す。1 つの特定の例において、P e r i p h e r a l C o m p o n e n t I n t e r c o n n e c t E x p r e s s (P C I e) は、I / O アダプタとホスト・システムの間でトランザクションのための双方向通信プロトコルを定める、コンポーネント・レベルの相互接続規格である。P C I e 通信は、P C I e バス上での伝送のための P C I e 規格に従って、パケットの形でカプセル化される。I / O アダプタから発し、ホスト・システムで終了するトランザクションは、上り方向トランザクションと呼ばれる。ホスト・システムから発し、I / O アダプタで終了するトランザクションは、下り方向トランザクションと呼ばれる。P C I e トポロジーは、対にされて（例えば、1 つの上り方向リンク、1 つの下り方向リンク）P C I e を形成する、ポイント・ツー・ポイントの単

40

50

方向リンクに基づいている。P C I e 標準は、P C I - S I G により保持され、公開されている。

【 0 1 0 8 】

当業者により認識されるように、本発明の態様は、システム、方法、またはコンピュータ・プログラム製品として具体化することができる。従って、本発明の態様は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコード等を含む）、または、ソフトウェアの態様とハードウェアの態様とを組み合わせた実施形態の形を取ることができ、これらは全て、本明細書において、一般的に「回路」、「モジュール」または「システム」と呼ぶことができる。さらに、本発明の態様は、コンピュータ可読プログラム・コードが組み込まれた、1以上のコンピュータ可読媒体内に具体化されたコンピュータ・プログラム製品の形を取ることができ

10

【 0 1 0 9 】

1以上のコンピュータ可読媒体のいずれの組み合わせを用いることもできる。コンピュータ可読媒体は、コンピュータ可読ストレージ媒体とすることができる。コンピュータ可読ストレージ媒体は、これらに限定されるものではないが、例えば、電子、磁気、光学、電磁気、赤外線または半導体のシステム、装置若しくはデバイス、または上記のいずれかの適切な組み合わせとすることができる。コンピュータ可読ストレージ媒体のより具体的な例（非網羅的なリスト）として、以下のもの、即ち、1以上の配線を有する電氣的接続、ポータブル・コンピュータ・ディスク、ハード・ディスク、ランダム・アクセス・メモリ（R A M）、読み出し専用メモリ（R O M）、消去可能なプログラム可能読み出し専用メモリ（E P R O M またはフラッシュメモリ）、光ファイバ、ポータブル・コンパクト・ディスク読み出し専用メモリ（C D - R O M）、光記憶装置、磁気記憶装置、または上記のいずれかの適切な組み合わせが挙げられる。本明細書の文脈において、コンピュータ可読ストレージ媒体は、命令実行システム、装置若しくはデバイスによって用いるため、またはそれらと接続して用いるためにプログラムを収容または格納することができるい

20

【 0 1 1 0 】

ここで図 2 4 を参照すると、一例において、コンピュータ・プログラム製品 1 3 0 0 が、例えば、本発明の 1 以上の態様を提供し、容易にするように、コンピュータ可読プログラム・コード手段または論理 1 3 0 4 をその上に格納するための 1 以上のコンピュータ可読ストレージ媒体 1 3 0 2 を含む。

30

【 0 1 1 1 】

コンピュータ可読媒体上に具体化されたプログラム・コードは、これらに限られるものではないが、無線、有線、光ファイバ・ケーブル、R F 等、または上記のいずれかの適切な組み合わせを含む、適切な媒体を用いて伝送することができる。

【 0 1 1 2 】

本発明の態様に関する操作を実行するためのコンピュータ・プログラム・コードは、J a v a、S m a l l T a l k、C + + 等のようなオブジェクト指向型プログラミング言語、および、「C」プログラミング言語、アセンブラ、または同様のプログラミング言語のような従来の手続き型プログラミング言語を含む、1以上のプログラミング言語のいずれかの組み合わせで書くことができる。プログラム・コードは、完全にユーザのコンピュータ上で実行される場合もあり、スタンドアロンのソフトウェア・パッケージとして、一部がユーザのコンピュータ上で実行される場合もあり、一部がユーザのコンピュータ上で実行され、一部が遠隔コンピュータ上で実行される場合もあり、または完全に遠隔コンピュータ若しくはサーバ上で実行される場合もある。最後のシナリオにおいては、遠隔コンピュータは、ローカル・エリア・ネットワーク（L A N）若しくは広域ネットワーク（W A N）を含むいずれかのタイプのネットワークを通じてユーザのコンピュータに接続される場合もあり、または外部コンピュータへの接続がなされる場合もある（例えば、インターネット・サービス・プロバイダを用いたインターネットを通じて）。

40

【 0 1 1 3 】

50

本発明の態様は、本発明の実施形態による方法、装置（システム）およびコンピュータ・プログラム製品のフローチャート図および/またはブロック図を参照して、本明細書で説明される。フローチャート図および/またはブロック図の各ブロック、並びにフローチャート図および/またはブロック図内のブロックの組み合わせは、コンピュータ・プログラム命令によって実装できることが理解されるであろう。これらのコンピュータ・プログラム命令を、汎用コンピュータ、専用コンピュータ、または他のプログラム可能データ処理装置のプロセッサに与えてマシンを製造し、それにより、コンピュータまたは他のプログラム可能データ処理装置のプロセッサによって実行される命令が、フローチャートおよび/またはブロック図の1以上のブロックにおいて指定された機能/動作を実装するための手段を作り出すようにすることができる。

10

【0114】

これらのコンピュータ・プログラム命令はまた、コンピュータ、他のプログラム可能データ処理装置、または他のデバイスを特定の方式で機能させるように指示することができるコンピュータ可読媒体内に格納し、それにより、そのコンピュータ可読媒体内に格納された命令が、フローチャートおよび/またはブロック図の1以上のブロックにおいて指定された機能/動作を実装する命令を含む製品を製造するようにすることもできる。

【0115】

コンピュータ・プログラム命令はまた、コンピュータ、他のプログラム可能データ処理装置、または他のデバイス上にロードして、一連の動作ステップをコンピュータ、他のプログラム可能データ処理装置、または他のデバイス上で行わせてコンピュータ実施のプロセスを生成し、それにより、コンピュータまたは他のプログラム可能装置上で実行される命令が、フローチャートおよび/またはブロック図の1以上のブロックにおいて指定された機能/動作を実行するためのプロセスを提供するようにもすることもできる。

20

【0116】

図面内のフローチャートおよびブロック図は、本発明の種々の実施形態によるシステム、方法およびコンピュータ・プログラム製品の可能な実装のアーキテクチャ、機能および動作を示す。この点に関して、フローチャートまたはブロック図内の各ブロックは、指定された論理機能を実装するための1以上の実行可能命令を含むモジュール、セグメント、またはコードの部分を表すことができる。幾つかの代替的な実装において、ブロック内に記載された機能は、図面内に記載された順序とは異なる順序で行われ得ることに留意すべきである。例えば、連続して示された2つのブロックが、関与する機能に応じて、実際には、実質的に同時に実行されることもあり、ときにはブロックが逆順に実行されることもある。また、ブロック図および/またはフローチャート図の各ブロック、並びにブロック図および/またはフローチャート図内のブロックの組み合わせは、指定された機能または動作を行う専用ハードウェア・ベースのシステムによって、または専用ハードウェアとコンピュータ命令との組み合わせによって実装できることに留意されたい。

30

【0117】

上記に加えて、本発明の1以上の態様は、顧客環境の管理を提供するサービス・プロバイダにより、供与し、提供し、配置し、管理し、サービスを行うことなどができる。例えば、サービス・プロバイダは、1または複数の顧客に対して本発明の1以上の態様を実施するコンピュータ・コードおよび/またはコンピュータ・インフラストラクチャを作成し、保持し、サポートすることなどができる。見返りとして、サービス・プロバイダは、例として、予約申し込みおよび/または報酬契約の下で顧客から支払いを受けることができる。付加的にまたは代替的に、サービス・プロバイダは、1または複数の第三者に対する広告コンテンツの販売から支払いを受けることができる。

40

【0118】

本発明の1つの態様において、本発明の1以上の態様を実施するために、アプリケーションを配置することができる。一例として、アプリケーションの配置は、本発明の1以上の態様を実施するように動作可能なコンピュータ・インフラストラクチャを提供することを含む。

50

【0119】

本発明の更に別の態様として、コンピュータ可読コードをコンピュータ・システムに統合することを含む、コンピュータ・インフラストラクチャを配置することが可能であり、そこでは、コードは、コンピューティング・システムと協働して、本発明の1以上の態様を実施することができる。

【0120】

本発明の更に別の態様として、コンピュータ可読コードをコンピュータ・システムに統合することを含む、コンピュータ・インフラストラクチャを統合するためのプロセスを提供することができる。コンピュータ・システムは、コンピュータ可読媒体を含み、ここで、コンピュータ媒体は本発明の1以上の態様を含む。コードは、コンピュータ・システムと協働して、本発明の1以上の態様を実施することができる。

10

【0121】

種々の実施形態が上述されたが、これらは一例にすぎない。例えば、他のアーキテクチャのコンピューティング環境が、本発明の1以上の態様を組み込み、用いることが可能である。例として、インターナショナル・ビジネス・マシーンス・コーポレーションにより提供される Power Systems サーバ若しくは他のサーバ、または他の会社のサーバのような、System z (登録商標) サーバ以外のサーバが、本発明の1以上の態様を含み、使用し、および/またはそこから利益を得ることができる。さらに、本明細書での例では、アダプタおよびPCIハブはサーバの一部と見なされるが、他の実施形態においては、これらを必ずしもサーバの一部と見なす必要はなく、単にコンピューティング環境のシステム・メモリおよび/または他のコンポーネントに結合されていると見なすことができる。コンピューティング環境は、サーバである必要はない。さらに、変換テーブルについて説明されるが、いずれのデータ構造体を用いることもでき、テーブルという用語は、そのようなデータ構造体の全てを含むものである。さらにまた、アダプタはPCIベースのものであるが、本発明の1以上の態様は、他のアダプタまたは他のI/Oコンポーネントと共に使用可能である。アダプタおよびPCIアダプタは、単なる例である。さらに、本発明の思想から逸脱することなく、他のサイズのアドレス空間およびアドレス・テーブル、および/またはテーブルを用いることもできる。さらに、DTEは、より多くの、より少ない、および異なる情報を含んでもよい。さらにまた、本発明の1以上の態様を用いて、他のタイプのアドレスを変換することもできる。さらに、アドレス空間識別子および/またはリクエスト識別子に関して、他の値を用いることができる。多くの他の変形が可能である。

20

30

【0122】

さらに、他のタイプのコンピューティング環境が、本発明の1以上の態様から利益を得ることができる。一例として、システム・バスを通してメモリ要素に直接的にまたは間接的に結合された少なくとも2つのプロセッサを含む、プログラム・コードを格納および/または実行するのに適したデータ処理システムが使用可能である。メモリ要素は、例えば、プログラム・コードの実際の実行時に用いられるローカル・メモリと、大容量記憶装置と、実行時に大容量記憶装置からコードを取得しなければならない回数を減少させるために少なくとも幾つかのプログラム・コードの一時的なストレージを提供するキャッシュ・メモリとを含む。

40

【0123】

入力/出力即ちI/O装置(キーボード、ディスプレイ、ポインティング装置、DASD、テープ、CD、DVD、親指ドライブ、および他のメモリ媒体等を含むが、これらに限定されるものではない)は、直接的に、または介在するI/Oコントローラを通して、システムに結合することができる。データ処理システムが、介在するプライベート・ネットワークまたは公衆ネットワークを通して、他のデータ処理システムまたはリモート・プリンタ若しくはストレージ装置に結合できるように、ネットワーク・アダプタをシステムに結合することもできる。モデム、ケーブル・モデムおよびイーサネット・カードは、利用可能なタイプのネットワーク・アダプタのほんの数例にすぎない。

50

【 0 1 2 4 】

図 2 5 を参照すると、本発明の 1 以上の態様を実装するためのホスト・コンピュータ・システム 5 0 0 0 の代表的なコンポーネントが描かれている。代表的なホスト・コンピュータ 5 0 0 0 は、コンピュータ・メモリ（即ち、中央ストレージ）5 0 0 2 と通信状態にある 1 以上の CPU 5 0 0 1 に加えて、ストレージ媒体デバイス 5 0 1 1 および他のコンピュータまたは SAN 等と通信するためのネットワーク 5 0 1 0 への I/O インターフェースを含む。CPU 5 0 0 1 は、アーキテクチャ化された（architected）命令セットおよびアーキテクチャ化された機能を有するアーキテクチャに準拠している。CPU 5 0 0 1 は、プログラム・アドレス（仮想アドレス）をメモリの実アドレスに変換するための動的アドレス変換（Dynamic Address Translation、DAT）5 0 0 3 を有することができる。DAT は、一般的に、変換をキャッシュに入れるための変換ルックアサイド・バッファ（TLB）5 0 0 7 を含むので、コンピュータ・メモリ 5 0 0 2 のブロックへの後のアクセスは、アドレス変換の遅延を必要としない。一般的に、コンピュータ・メモリ 5 0 0 2 とプロセッサ 5 0 0 1 との間に、キャッシュ 5 0 0 9 が用いられる。キャッシュ 5 0 0 9 は、1 つより多い CPU が利用可能な大容量のキャッシュと、大型のキャッシュと各 CPU との間のより小型でより高速な（下位レベルの）キャッシュとを有する階層とすることができる。幾つかの実装において、下位レベルのキャッシュは、命令のフェッチおよびデータ・アクセスのために別個の下位レベル・キャッシュを与えるように分割される。一実施形態においては、キャッシュ 5 0 0 9 を介して、命令フェッチ・ユニット 5 0 0 4 により、命令がメモリ 5 0 0 2 からフェッチされる。命令は、命令デコード・ユニット 5 0 0 6 でデコードされ、命令実行ユニット 5 0 0 8 にディスパッチされる（幾つかの実施形態においては他の命令と共に）。一般的には、例えば、算術演算実行ユニット、浮動小数点実行ユニット、および分岐命令実行ユニットなどの幾つかの実行ユニット 5 0 0 8 が用いられる。命令は、実行ユニットにより実行され、必要に応じて命令が指定したレジスタまたはメモリからオペランドにアクセスする。メモリ 5 0 0 2 からオペランドにアクセスする（ロードまたはストアする）場合、典型的には、ロード/ストア・ユニット 5 0 0 5 が、実行される命令の制御下でアクセスを処理する。命令は、ハードウェア回路または内部のマイクロコード（ファームウェア）において、またはその両方の組み合わせによって実行することができる。

10

20

【 0 1 2 5 】

既述のように、コンピュータ・システムは、ローカル（または主）ストレージ内の情報、並びに、アドレッシング、保護、参照、および変更の記録を含む。アドレッシングの幾つかの態様は、アドレスの形式、アドレス空間の概念、種々のタイプのアドレス、および 1 つのタイプのアドレスを別のタイプのアドレスに変換する方法を含む。主ストレージの一部は、永続的に割り当てられた記憶位置を含む。主ストレージは、システムに、データの直接アドレス指定可能な高速アクセス・ストレージを与える。データおよびプログラムを処理できるようになる前に、（入力装置から）データおよびプログラムの両方は、主ストレージにロードされる。

30

【 0 1 2 6 】

主ストレージは、キャッシュと呼ばれることもある、1 以上のより小さくより高速アクセスのバッファ・ストレージを含むことができる。キャッシュは、典型的には、CPU または I/O プロセッサと物理的に関連付けられる。物理的構成および別個のストレージ媒体を使用することの影響は、性能に対するものを除き、通常、プログラムにより観察することはできない。

40

【 0 1 2 7 】

命令およびデータ・オペランドについて、別個のキャッシュを保持することができる。キャッシュ内の情報は、キャッシュ・ブロックまたはキャッシュ・ライン（または短縮してライン）と呼ばれる、整数境界（integral boundary）上にある連続したバイト内に保持される。モデルは、キャッシュ・ラインのサイズをバイトで返す、EXTRACT CACHE ATTRIBUTE 命令を提供することができる。モデルはまた、データ若し

50

くは命令キャッシュへのストレージのプリフェッチ、または、キャッシュからのデータの解放に影響を与える、`PREFETCH DATA`および`PREFETCH DATA RELATIVE LONG`命令を提供することができる。

【0128】

ストレージは、長い水平方向のビットの文字列と考えられる。大部分の操作において、ストレージへのアクセスは、左から右への順序で進む。ビットの文字列は、8ビット単位で分割される。8ビットの単位は1バイトと呼ばれ、全ての情報の形式の基本的な構成要素 (building block) である。ストレージ内の各々のバイト位置は、負でない一意の整数により識別され、この整数がそのバイト位置のアドレスであり、即ち、簡単にバイト・アドレスである。隣接するバイト位置は、連続するアドレスを有し、左の0で始まり、左から右への順序で進む。アドレスは、符号なしの2進整数であり、24ビット、31ビット、または64ビットである。

10

【0129】

情報は、ストレージとCPUまたはチャンネル・サブシステムとの間で、一度に1バイトずつ、または1バイト・グループずつ伝送される。特に断りのない限り、例えば`z/Architecture` (登録商標) においては、ストレージ内のバイト・グループは、グループの左端のバイトによりアドレス指定される。グループ内のバイト数は、実行される操作により暗黙にまたは明示的に指定される。CPU操作に用いられる場合、バイト・グループはフィールドと呼ばれる。例えば`z/Architecture` (登録商標) においては、バイト・グループの中の各々において、ビットは、左から右の順序で番号が付けられる。`z/Architecture` (登録商標) においては、左端ビットは「上位 (high-order)」ビットと呼ばれることがあり、右端ビットは「下位 (low-order)」ビットと呼ばれることがある。しかしながら、ビット数は、ストレージ・アドレスではない。バイトだけを、アドレス指定することができる。ストレージ内の1つのバイトの個々のビットに対して操作を行うためには、そのバイト全体にアクセスされる。1バイトの中のビットには、左から右に0から7までの番号が付けられる (例えば`z/Architecture` (登録商標) において)。1つのアドレスの中のビットには、24ビット・アドレスの場合は8 - 31若しくは40 - 63の番号を付けることができ、または、31ビット・アドレスの場合は1 - 31若しくは33 - 63の番号を付けることができ、64ビット・アドレスの場合は0 - 63の番号が付けられる。複数バイトから成る他のいずれかの固定長形式の中では、その形式を構成するビットには、0から始まる連続番号が付けられる。エラー検出のためおよび好ましくは訂正のために、各バイトまたはバイト・グループと共に、1以上の検査ビットが伝送されることがある。このような検査ビットは、マシンにより自動的に生成されるものであり、プログラムが直接制御することはできない。記憶容量は、バイト数で表わされる。ストレージ・オペランド・フィールドの長さが命令のオペレーション・コードで暗黙的に指定される場合、そのフィールドは固定長を有すると言われる。固定長は、1バイト、2バイト、4バイト、8バイト、または16バイトとすることができる。一部の命令では、より長いフィールドが暗黙的に指定されることもある。ストレージ・オペランド・フィールドの長さが暗黙的に指定されず明示的に記述される場合は、そのフィールドは可変長を有すると言われる。可変長オペランドは、1バイトのインクリメントにより変化し得る (または、一部の命令では、2バイトの倍数若しくは他の倍数)。情報がストレージ内に置かれるとき、ストレージへの物理パスの幅が格納されるフィールドの長さを上回り得るとしても、指定されたフィールド内に含まれるバイトの記憶位置のコンテンツのみが置き換えられる。

20

30

40

【0130】

特定の情報単位は、ストレージ内の整数境界上にあることになる。そのストレージ・アドレスがバイトでの単位での長さの倍数であるとき、境界は、情報単位に関して整数のものであると言われる。整数境界上にある2バイト、4バイト、8バイト、および16バイトのフィールドには、特別な名称が与えられる。ハーフワード (halfword) は、2バイト境界上にある2個の連続したバイトのグループであり、これは、命令の基本的な構成要素

50

である。ワード (word) は、4 バイト境界上にある 4 個の連続したバイトのグループである。ダブルワード (doubleword) は、8 バイト境界上にある 8 個の連続したバイトのグループである。クワッドワード (quadword) は、16 バイト境界上にある 16 個の連続したバイトのグループである。ストレージ・アドレスが、ハーフワード、ワード、ダブルワード、およびクワッドワードを示す場合、そのアドレスの 2 進表現は、それぞれ、右端の 1 個、2 個、3 個、または 4 個のビットが 0 になる。命令は、2 バイトの整数境界上にあることになる。大部分の命令のストレージ・オペランドは、境界合わせ (boundary alignment) 要件をもたない。

【0131】

命令およびデータ・オペランドに対して別個のキャッシュを実装するデバイスにおいては、ストアが、後にフェッチされる命令を変更するかどうかに関係なく、後に命令をフェッチするキャッシュ・ライン内にプログラムが格納される場合には、著しい遅延が生じることがある。

10

【0132】

1 つの実施形態において、本発明は、ソフトウェア (ライセンス内部コード、ファームウェア、マイクロコード、ミリコード、ピココードなどと呼ばれる場合もあるが、そのいずれも本発明と整合性がある) により実施することができる。図 25 を参照すると、本発明を具体化するソフトウェア・プログラム・コードは、典型的には、ホスト・システム 5000 のプロセッサ 5001 により、CD-ROM ドライブ、テープドライブ、またはハードドライブといった長期ストレージ媒体デバイス 5011 からアクセスされる。ソフトウェア・プログラム・コードは、ディスク、ハードドライブ、または CD-ROM といった、データ処理システムと共に用いるための種々の周知の媒体のいずれかの上で具体化することができる。コードは、こうした媒体上に分散させても、またはコンピュータ・メモリ 5002 からユーザに分散させても、または、こうした他のシステムのユーザが使用するために、ネットワーク 5010 上の 1 つのコンピュータ・システムのストレージから他のコンピュータ・システムに分散させてもよい。

20

【0133】

ソフトウェア・プログラム・コードは、種々のコンピュータ・コンポーネントおよび 1 以上のアプリケーション・プログラムの機能および相互作用を制御するオペレーティング・システムを含む。プログラム・コードは、通常、ストレージ媒体デバイス 5011 から相対的により高速のコンピュータ・ストレージ 5002 にページングされ、そこでプロセッサ 5001 による処理のために利用可能になる。ソフトウェア・プログラム・コードをメモリ内、物理的媒体上で具体化し、および/または、ネットワークを介してソフトウェア・コードを分散させる技術および方法は周知であり、ここではこれ以上論じない。プログラム・コードは、有形の媒体 (これらに限定されるものではないが、電子メモリ・モジュール (RAM)、フラッシュメモリ、コンパクトディスク (CD)、DVD、磁気テープなどを含む) 上に作成され格納されたとき、「コンピュータ・プログラム製品」と呼ばれることが多い。コンピュータ・プログラム製品媒体は、典型的には、処理回路による実行のために、好ましくはコンピュータ・システム内の処理回路によって読み取り可能である。

30

40

【0134】

図 26 は、本発明を実施することができる代表的なワークステーションまたはサーバ・ハードウェア・システムを示す。図 12 のシステム 5020 は、随意的な周辺機器を含む、パーソナル・コンピュータ、ワークステーション、またはサーバなどの代表的なベース・コンピュータ・システム 5021 を含む。ベース・コンピュータ・システム 5021 は、1 以上のプロセッサ 5026 と、周知の技術に従ってプロセッサ 5026 とシステム 5021 の他のコンポーネントを接続し、これらの間の通信を可能にするために用いられるバスを含む。バスは、プロセッサ 5026 を、ハードドライブ (例えば、磁気媒体、CD、DVD、およびフラッシュメモリのいずれかを含む) またはテープドライブを含むことができる、メモリ 5025 および長期ストレージ 5027 に接続する。システム 502

50

1はまた、バスを介して、マイクロプロセッサ5026を、キーボード5024、マウス5023、プリンタ/スキャナ5030、および/またはタッチ・センシティブ・スクリーン、デジタル化された入力パッド等のいずれかのユーザ・インターフェース機器とすることができる他のインターフェース機器といった、1以上のインターフェース機器に接続する、ユーザ・インターフェース・アダプタを含むこともできる。バスはまた、ディスプレイ・アダプタを介して、LCDスクリーンまたはモニタなどのディスプレイ装置5022をマイクロプロセッサ5026にも接続する。

【0135】

システム5021は、ネットワーク5029と通信する5028ことができるネットワーク・アダプタを介して、他のコンピュータまたはコンピュータ・ネットワークと通信することができる。例示的なネットワーク・アダプタは、通信チャネル、トークン・リング、イーサネットまたはモデムである。或いは、システム5021は、CDPD(セルラー・デジタル・パケット・データ)カードのような無線インターフェースを用いて通信することもできる。システム5021は、ローカル・エリア・ネットワーク(LAN)若しくは広域ネットワーク(WAN)、またはシステム5021内のこうした他のコンピュータと関連付けることができ、または、別のコンピュータ等とのクライアント/サーバ構成におけるクライアントとすることができる。これらの構成の全て、並びに、適切な通信ハードウェアおよびソフトウェアは、当技術分野において周知である。

【0136】

図27は、本発明を実施することができるデータ処理ネットワーク5040を示す。データ処理ネットワーク5040は、各々が複数の個々のワークステーション5041、5042、5043、5044を含むことができる、無線ネットワークおよび有線ネットワークのような複数の個々のネットワークを含むことができる。さらに、当業者であれば理解するように、1以上のLANを含ませることができ、そこで、LANは、ホスト・プロセッサに結合された複数のインテリジェント・ワークステーションを含むことができる。

【0137】

さらに図27を参照すると、ネットワークはまた、ゲートウェイ・コンピュータ(クライアント・サーバ5046)、またはアプリケーション・サーバ(データ・リポジトリにアクセスすることができ、かつ、ワークステーション5045から直接アクセスすることもできる遠隔サーバ5048)のような、メインフレーム・コンピュータまたはサーバを含むこともできる。ゲートウェイ・コンピュータ5046は、各々の個々のネットワークへの入口点として働く。ゲートウェイは、1つのネットワーク・プロトコルを別のものに接続するときに必要なとされる。ゲートウェイ5046は、通信リンクによって別のネットワーク(例えば、インターネット5047)に結合することが好ましい。ゲートウェイ5046はまた、通信リンクを用いて、1以上のワークステーション5041、5042、5043、5044に直接結合することもできる。ゲートウェイ・コンピュータは、International Business Machines Corporation社から入手可能なIBM eServer(商標)System z(登録商標)サーバを用いて実装することができる。

【0138】

図26および図27を同時に参照すると、本発明を具体化することができるソフトウェア・プログラム・コードには、一般的に、CD-ROMドライブまたはハードドライブといった長期ストレージ媒体5027から、システム5020のプロセッサ5026によってアクセスすることができる。ソフトウェア・プログラム・コードは、ディスク、ハードドライブ、またはCD-ROMといった、データ処理システムと共に用いるための種々の周知の媒体のいずれかの上で具体化することができる。コードは、そのような媒体上で分散させても、またはメモリからユーザ5050、5051に分散させても、或いは、こうした他のシステムのユーザが用いるために、ネットワーク上の1つのコンピュータ・システムのストレージから他のコンピュータ・システムに分散させてもよい。

【0139】

或いは、プログラム・コードをメモリ5025内で具体化し、プロセッサ・バスを用いてプロセッサ5026によってプログラム・コードにアクセスすることができる。このようなプログラム・コードは、種々のコンピュータ・コンポーネントおよび1以上のアプリケーション・プログラム5032の機能および相互作用を制御するオペレーティング・システムを含む。プログラム・コードは、通常、ストレージ媒体5027から高速メモリ5025にページングされ、そこでプロセッサ5026による処理のために利用可能になる。ソフトウェア・プログラム・コードをメモリ内、物理的媒体上で具体化し、および/または、ネットワークを介してソフトウェア・コードを配布する技術および方法は周知であり、ここではこれ以上論じない。プログラム・コードは、作成され、有形の媒体（これらに限定されるものではないが、電子メモリ・モジュール（RAM）、フラッシュメモリ、コンパクトディスク（CD）、DVD、磁気テープなどを含む）に格納されたとき、「コンピュータ・プログラム製品」と呼ばれることが多い。コンピュータ・プログラム製品媒体は、典型的には、処理回路による実行のために、好ましくはコンピュータ・システム内の処理回路によって読み取り可能である。

【0140】

プロセッサが最も容易に利用できるキャッシュ（通常、プロセッサの他のキャッシュよりも高速で小さい）は、最下位（L1またはレベル1）のキャッシュであり、メインストア（主メモリ）は、最上位レベルのキャッシュ（3つのレベルがある場合にはL3）である。最下位レベルのキャッシュは、実行されるマシン命令を保持する命令キャッシュ（I-キャッシュ）と、データ・オペランドを保持するデータ・キャッシュ（D-キャッシュ）とに分割されることが多い。

【0141】

図28を参照すると、プロセッサ5026についての例示的なプロセッサの実施形態が示される。典型的には、メモリ・ブロックをバッファに入れてプロセッサ性能を向上させるために、1以上のレベルのキャッシュ5053が用いられる。キャッシュ5053は、用いられる可能性が高いメモリ・データのキャッシュ・ラインを保持する高速バッファである。典型的なキャッシュ・ラインは、64バイト、128バイト、または256バイトのメモリ・データである。データをキャッシュに入れるのではなく、命令をキャッシュに入れるために、別個のキャッシュが用いられることが多い。キャッシュ・コヒーレンス（メモリおよびキャッシュ内のラインのコピーの同期）は、多くの場合、当技術分野において周知の種々の「スヌープ（snoop）」アルゴリズムによって与えられる。プロセッサ・システムの主メモリ・ストレージ5025は、キャッシュと呼ばれることが多い。4つのレベルのキャッシュ5053を有するプロセッサ・システムにおいて、主ストレージ5025は、典型的にはより高速であり、かつ、コンピュータ・システムが利用できる不揮発性ストレージ（DASD、テープ等）の一部だけを保持するので、レベル5（L5）のキャッシュと呼ばれることがある。主ストレージ5025は、オペレーティング・システムによって主ストレージ5025との間でページングされるデータのページを「キャッシュに入れる」。

【0142】

プログラム・カウンタ（命令カウンタ）5061は、実行される現行の命令のアドレスを常時監視している。z/Architecture（登録商標）プロセッサのプログラム・カウンタは64ビットであり、従来のアドレッシング制限をサポートするために、31ビットまたは24ビットに切り捨てることができる。プログラム・カウンタは、典型的には、コンテキスト・スイッチの際に持続するように、コンピュータのPSW（プログラム状況ワード）内で具体化される。従って、例えば、オペレーティング・システムにより、プログラム・カウンタ値を有する進行中のプログラムに割り込みをかけることが可能である（プログラム環境からオペレーティング・システム環境へのコンテキスト・スイッチ）。プログラムのPSWは、プログラムがアクティブでない間、プログラム・カウンタ値を保持し、オペレーティング・システムが実行されている間、オペレーティング・システムの（PSW内の）プログラム・カウンタが用いられる。典型的には、プログラム・カウ

10

20

30

40

50

ンタは、現行の命令のバイト数に等しい量だけインクリメントされる。RISC (Reduce Instruction Set Computing、縮小命令セット・コンピューティング) 命令は、典型的には固定長であり、CISC (Complex Instruction Set Computing、複合命令セット・コンピューティング) 命令は、典型的には可変長である。IBM z/Architecture (登録商標) の命令は、2バイト、4バイト、または6バイトの長さを有するCISC命令である。例えば、コンテキスト・スイッチ操作または分岐命令の分岐成立 (Branch taken) 操作により、プログラム・カウンタ5061が変更される。コンテキスト・スイッチ操作において、現行のプログラム・カウンタ値は、実行されるプログラムについての他の状態情報 (条件コードのような) と共にプログラム状況ワード内に保存され、実行される新しいプログラム・モジュールの命令を指し示す新しいプログラム・カウンタ値がロードされる。分岐成立操作を行い、分岐命令の結果をプログラム・カウンタ5061にロードすることにより、プログラムが判断を下すことまたはプログラム内でループすることを可能にする。

10

【0143】

典型的には、プロセッサ5026の代わりに命令をフェッチするために、命令フェッチ・ユニット5055が用いられる。フェッチ・ユニットは、「次の順次命令 (next sequential instruction)」、分岐成立命令のターゲット命令、またはコンテキスト・スイッチの後のプログラムの最初の命令のいずれかをフェッチする。今日の命令フェッチ・ユニットは、プリフェッチされた命令を用いることができる可能性に基づいて、命令を投機的にプリフェッチするプリフェッチ技術を用いることが多い。例えば、フェッチ・ユニットは、次の順次命令を含む16バイトの命令と、付加的なバイトの更なる順次命令とをフェッチすることができる。

20

【0144】

次いで、フェッチされた命令が、プロセッサ5026によって実行される。一実施形態において、フェッチされた命令は、フェッチ・ユニットのディスパッチ・ユニット5056に渡される。ディスパッチ・ユニットは命令をデコードし、デコードされた命令についての情報を適切なユニット5057、5058、5060に転送する。実行ユニット5057は、典型的には、命令フェッチ・ユニット5055からデコードされた算術命令についての情報を受け取り、命令のオペコードに従ってオペランドに関する算術演算を行う。オペランドは、好ましくは、メモリ5025、アーキテクチャ化レジスタ5059、または実行される命令の即値フィールドのいずれかから、実行ユニット5057に与えられる。実行の結果は、格納された場合には、メモリ5025、レジスタ5059、または他のマシン・ハードウェア (制御レジスタ、PSWレジスタなどのような) 内に格納される。

30

【0145】

プロセッサ5026は、典型的には、命令の機能を実行するための1以上の実行ユニット5057、5058、5060を有する。図29を参照すると、実行ユニット5057は、インターフェース論理5071を介して、アーキテクチャ化された汎用レジスタ5059、デコード/ディスパッチ・ユニット5056、ロード・ストア・ユニット5060、および他のプロセッサ・ユニット5065と通信することができる。実行ユニット5057は、幾つかのレジスタ回路5067、5068、5069を用いて、算術論理演算ユニット (ALU) 5066が動作する情報を保持することができる。ALUは、加算 (add)、減算 (subtract)、乗算 (multiply)、および除算 (divide) などの算術演算、並びに、論理積 (and)、論理和 (or)、および排他的論理和 (XOR)、ローテート (rotate) およびシフト (shift) のような論理関数を実行する。ALUは、設計に依存する専用の演算をサポートすることが好ましい。他の回路は、例えば条件コードおよび復旧サポート論理を含む、他のアーキテクチャ化ファシリティ5072を提供することができる。典型的には、ALU演算の結果は、出力レジスタ回路5070に保持され、この出力レジスタ回路5070が、結果を種々の他の処理機能に転送することができる。多数のプロセッサ・ユニットの構成が存在し、本説明は、一実施形態の代表的な理解を与えることのみを意図している。

40

50

【0146】

例えばADD命令は、算術および論理機能を有する実行ユニット5057で実行され、一方、例えば浮動小数点命令は、特化された浮動小数点能力を有する浮動小数点実行部で実行される。実行ユニットは、オペランドに対してオペコードが定めた関数を行うことにより、命令が特定したオペランドに対して動作することが好ましい。例えば、ADD命令は、命令のレジスタ・フィールドによって特定された2つのレジスタ5059内に見出されるオペランドに対して、実行ユニット5057により実行することができる。

【0147】

実行ユニット5057は、2つのオペランドに対して算術加算を実行し、結果を第3オペランドに格納し、ここで第3オペランドは、第3のレジスタであってもまたは2つのソース・レジスタのいずれかであってもよい。実行ユニットは、シフト、ローテート、論理積、論理和、および排他的論理和のような種々の論理関数、並びに、加算、減算、乗算、除法のいずれかを含む、種々の代数関数を実行することができる算術論理演算ユニット(ALU)5066を用いることが好ましい。スカラー演算のために設計されたALU5066もあり、浮動小数点のために設計されたものALU5066もある。データは、アーキテクチャに応じて、ビッグエンディアン(Big Endian)(最下位のバイトが最も高いバイト・アドレスである)、またはリトルエンディアン(Little Endian)(最下位のバイトが最も低いバイト・アドレスである)とすることができる。IBM z/Architecture(登録商標)は、ビッグエンディアンである。符号付きフィールドは、アーキテクチャに応じて、符号および大きさ、1の補数、または2の補数とすることができる。2の補数における負の値または正の値は、ALU内で加法しか必要としないため、ALUが減算能力を設計する必要がないという点で、2の補数は有利である。数値は、通常、省略表現で記述され、12ビット・フィールドは、4,096バイトブロックのアドレスを定め、通常、例えば4Kバイト(キロバイト)ブロックのように記述される。

【0148】

図30を参照すると、分岐命令を実行するための分岐命令情報が、典型的には、分岐ユニット5058に送られ、この分岐ユニット5058は、多くの場合、分岐履歴テーブル5082のような分岐予測アルゴリズムを用いて、他の条件付き演算が完了する前に分岐の結果を予測する。条件付き演算が完了する前に、現行の分岐命令のターゲットがフェッチされ、投機的に実行される。条件付き演算が完了すると、投機的に実行された分岐命令は、条件付き演算の条件および投機された結果に基づいて、完了されるかまたは破棄される。典型的な分岐命令は、条件コードを試験し、条件コードが分岐命令の分岐要件を満たす場合、ターゲット・アドレスに分岐することができ、ターゲット・アドレスは、例えば、命令のレジスタ・フィールドまたは即値フィールド内に見出されるものを含む幾つかの数に基づいて計算することができる。分岐ユニット5058は、複数の入力レジスタ回路5075、5076、5077と、出力レジスタ回路5080とを有するALU5074を用いることができる。分岐ユニット5058は、例えば、汎用レジスタ5059、デコード・ディスパッチ・ユニット5056、または他の回路5073と通信することができる。

【0149】

例えば、オペレーティング・システムによって開始されるコンテキスト・スイッチ、コンテキスト・スイッチを発生させるプログラム例外またはエラー、コンテキスト・スイッチを発生させるI/O割り込み信号、或いは、(マルチスレッド環境における)複数のプログラムのマルチスレッド活動を含む様々な理由により、命令のグループの実行に割り込みがかけられることがある。コンテキスト・スイッチ動作は、現在実行中のプログラムについての状態情報を保存し、次いで、起動される別のプログラムについての状態情報をロードすることが好ましい。状態情報は、例えば、ハードウェア・レジスタまたはメモリ内に保存することができる。状態情報は、実行される次の命令を指し示すプログラム・カウンタ値と、条件コードと、メモリ変換情報と、アーキテクチャ化されたレジスタの内容とを含むことが好ましい。コンテキスト・スイッチの活動は、ハードウェア回路、アプリケ

10

20

30

40

50

ーション・プログラム、オペレーティング・システム・プログラム、またはファームウェア・コード（マイクロコード、ピココード、またはライセンス内部コード（LIC））単独でまたはその組み合わせで実施することができる。

【0150】

プロセッサは、命令により定義された方法に従ってオペランドにアクセスする。命令は、命令の一部の値を用いて即値オペランドを与えることができ、汎用レジスタまたは専用レジスタ（例えば、浮動小数点レジスタ）のいずれかを明示的に指し示す1以上のレジスタ・フィールドを与えることができる。命令は、オペコード・フィールドによって、オペランドとして識別されるインプライド・レジスタ（implied register）を用いることができる。命令は、オペランドのためのメモリ位置を用いることができる。z / A r c h i t e c t u r e（登録商標）の長変位ファシリティ（long displacement facility）により例示されるように、オペランドのメモリ位置を、レジスタ、即値フィールド、またはレジスタと即値フィールドの組み合わせによって与えることができ、命令は、基底レジスタ、索引レジスタ、および即値フィールド（変位フィールド）を定め、これらが、例えば互いに加算されてメモリ内のオペランドのアドレスをもたらす。ここでの位置（location）は、典型的には、特に断りのない限り、主メモリ（主ストレージ）内の記憶位置を意味する。

10

【0151】

図31を参照すると、プロセッサは、ロード/ストア・ユニット5060を用いて、ストレージにアクセスする。ロード/ストア・ユニット5060は、メモリ5053内のターゲット・オペランドのアドレスを取得し、オペランドをレジスタ5059または別のメモリ5053の記憶位置にロードすることによってロード操作を行うことができ、或いは、メモリ5053内のターゲット・オペランドのアドレスを取得し、レジスタ5059または別のメモリ5053の記憶位置から取得したデータをメモリ5053内のターゲット・オペランドの記憶位置に格納することによって、ストア操作を行うことができる。ロード/ストア・ユニット5060は、投機的なものであってもよく、命令シーケンスに対してアウト・オブ・オーダー式の順序でメモリにアクセスすることができるが、プログラムに対して、命令がイン・オーダー式に実行されたという外観を維持することになる。ロード/ストア・ユニット5060は、汎用レジスタ5059、デコード/ディスパッチ・ユニット5056、キャッシュ/メモリ・インターフェース5053、または他の要素5083と通信することができ、ストレージ・アドレスを計算し、かつ、パイプライン処理を順に行って操作をイン・オーダー式に保持するための、種々のレジスタ回路、ALU5085、および制御論理5090を含む。一部の動作は、アウト・オブ・オーダー式とすることができるが、ロード/ストア・ユニットは、アウト・オブ・オーダー式動作が、プログラムに対して、当技術分野において周知のようなイン・オーダー式に実行されたように見えるようにする機能を提供する。

20

30

【0152】

好ましくは、アプリケーション・プログラムが「見ている」アドレスは、仮想アドレスと呼ばれることが多い。仮想アドレスは、「論理アドレス」および「実効アドレス（effective address）」と呼ばれることもある。これらの仮想アドレスは、これらに限定されるものではないが、単に仮想アドレスをオフセット値にプリフィックス付加すること、1以上の変換テーブルを介して仮想アドレスを変換することを含む、種々の動的アドレス変換（DAT）技術の1つによって、物理的メモリ位置にリダイレクトされるという点で仮想のものであり、変換テーブルは、少なくともセグメント・テーブルおよびページ・テーブルを単独でまたは組み合わせて含むことが好ましく、セグメント・テーブルは、ページ・テーブルを指し示すエントリを有することが好ましい。z / A r c h i t e c t u r e（登録商標）では、領域第1テーブル、領域第2テーブル、領域第3テーブル、セグメント・テーブル、および随意的なページ・テーブルを含む、変換の階層が提供される。アドレス変換の性能は、仮想アドレスを関連した物理的メモリ位置にマッピングするエントリを含む変換ルックアサイド・バッファ（TLB）を用いることにより改善されることが多

40

50

い。D A T が変換テーブルを用いて仮想アドレスを変換したときに、エントリが作成される。次いで、後に仮想アドレスを用いることで、低速の順次変換テーブル・アクセスではなく、高速の T L B のエントリを用いることが可能になる。T L B の内容は、L R U (Least Recently Used) を含む種々の置換アルゴリズムによって管理することができる。

【 0 1 5 3 】

プロセッサがマルチプロセッサ・システムのプロセッサである場合には、各プロセッサは、コヒーレンシのために、I / O、キャッシュ、T L B、およびメモリといった共有リソースをインターロック状態に保持する責任を負う。キャッシュ・コヒーレンシを保持する際に、一般的には「スヌープ」技術が用いられる。スヌープ環境においては、共有を容易にするために、各キャッシュ・ラインを、共有状態、排他的状態、変更状態、無効状態等のいずれか 1 つの状態にあるものとしてマーク付けすることができる。

10

【 0 1 5 4 】

I / O ユニット 5 0 5 4 (図 2 8) は、プロセッサに、例えば、テープ、ディスク、プリンタ、ディスプレイ、およびネットワークを含む周辺機器に取り付けるための手段を与える。I / O ユニットは、ソフトウェア・ドライバによってコンピュータ・プログラムに提示されることが多い。I B M (登録商標) による S y s t e m z (登録商標) のようなメインフレームにおいては、チャンネル・アダプタおよびオープン・システム・アダプタが、オペレーティング・システムと周辺機器との間に通信をもたらすメインフレームの I / O ユニットである。

【 0 1 5 5 】

さらに、他のタイプのコンピューティング環境が、本発明の 1 以上の態様から利益を得ることができる。一例として、環境は、特定のアーキテクチャ(例えば、命令実行、アドレス変換などのアーキテクチャ化された機能、およびアーキテクチャ化されたレジスタを含む)またはそのサブセットをエミュレートする(例えば、プロセッサおよびメモリを有するネイティブ・コンピュータ・システム上で)エミュレータ(例えば、ソフトウェアまたは他のエミュレーション機構)を含むことができる。このような環境においては、エミュレータを実行しているコンピュータが、エミュレートされる機能とは異なるアーキテクチャを有することができたとしても、エミュレータの 1 以上のエミュレーション機能 n により、本発明の 1 以上の態様が実施され得る。一例として、エミュレーション・モードにおいては、エミュレートされる特定の命令または操作がデコードされ、適切なエミュレーション機能が構築され、個々の命令または操作を実施する。

20

30

【 0 1 5 6 】

エミュレーション環境においては、ホスト・コンピュータは、例えば、命令およびデータを格納するためのメモリと、メモリから命令をフェッチし、随機的に、フェッチされた命令のためのローカル・バッファリングを提供するための命令フェッチ・ユニットと、フェッチされた命令を受信し、フェッチされた命令のタイプを判断するための命令デコード・ユニットと、命令を実行するための命令実行ユニットとを含む。実行は、データをメモリからレジスタ内にロードすること、データをレジスタから再びメモリに格納すること、またはデコード・ユニットにより判断されるように、何らかのタイプの算術演算または論理演算を実行することを含むことができる。一例においては、各ユニットは、ソフトウェアで実装される。例えば、ユニットが実行する演算は、エミュレータ・ソフトウェア内の 1 以上のサブルーチンとして実装される。

40

【 0 1 5 7 】

より具体的には、メインフレームにおいて、アーキテクチャ化されたマシン命令は、通常、プログラマによって、多くの場合コンパイラ・アプリケーションを介して、今日では「C」プログラマによって用いられる。ストレージ媒体内に格納されたこれらの命令は、z / A r c h i t e c t u r e (登録商標) の I B M (登録商標) サーバにおいて、または代替的に他のアーキテクチャを実行するマシンにおいて、ネイティブに実行することができる。これらの命令は、既存のおよび将来の I B M (登録商標) メインフレーム・サーバにおいて、および、I B M (登録商標) の他のマシン(例えば、P o w e r S y s t

50

emsサーバおよびSystem x（登録商標）サーバ）上で、エミュレートすることができる。これらの命令は、IBM（登録商標）、Intel（登録商標）、AMD（商標）などによって製造されたハードウェアを用いて種々のマシン上でLinuxを実行しているマシンにおいて実行することができる。Z/Architecture（登録商標）下でそのハードウェア上で実行することに加えて、Linuxを用いること、並びに、一般に実行がエミュレーション・モードにある、Hercules（www.hercules-390.orgを参照されたい）、またはFSI（Fundamental Software, Inc）（www.funsoft.comを参照されたい）によるエミュレーションを用いるマシンを用いることもできる。エミュレーション・モードにおいては、ネイティブ・プロセッサによって、エミュレーション・ソフトウェアが実行され、エミュレートされたプロセッサのアーキテクチャをエミュレートする。

10

【0158】

ネイティブ・プロセッサは、一般的に、エミュレートされたプロセッサのエミュレーションを実行するためにファームウェアまたはネイティブ・オペレーティング・システムのいずれかを含むエミュレーション・ソフトウェアを実行する。エミュレーション・ソフトウェアは、エミュレートされたプロセッサ・アーキテクチャの命令のフェッチと実行を担当する。エミュレーション・ソフトウェアは、エミュレートされたプログラム・カウンタを維持し、命令境界を常時監視している。エミュレーション・ソフトウェアは、一度に1以上のエミュレートされたマシン命令をフェッチし、ネイティブ・プロセッサにより実行するために、その1以上のエミュレートされたマシン命令を、対応するネイティブマシン命令のグループに変換することができる。これらの変換された命令は、より速い変換を達成できるようにキャッシュに入れることができる。それにも関わらず、エミュレーション・ソフトウェアは、エミュレートされたプロセッサ・アーキテクチャのアーキテクチャ規則を維持して、オペレーティング・システムおよびエミュレートされたプロセッサのために書かれたアプリケーションが正確に動作することを保証しなければならない。さらに、エミュレーション・ソフトウェアは、これらに限られるものではないが、制御レジスタ、汎用レジスタ、浮動小数点レジスタ、例えばセグメント・テーブルおよびページ・テーブルを含む動的アドレス変換機能、割り込み機構、コンテキスト・スイッチ機構、時刻（Time of Day、TOD）クロック、およびI/Oサブシステムへのアーキテクチャ化インターフェースを含む、エミュレートされたプロセッサのアーキテクチャによって識別されるリソースを提供し、オペレーティング・システムまたはエミュレートされたプロセッサ上で実行するように設計されたアプリケーション・プログラムが、エミュレーション・ソフトウェアを有するネイティブ・プロセッサ上で実行できるようにしなければならない。

20

30

【0159】

エミュレートされた特定の命令がデコードされ、個々の命令の機能を実行するためのサブルーチンが呼び出される。エミュレートされたプロセッサ1の機能をエミュレートするエミュレーション・ソフトウェア機能は、例えば、「C」サブルーチンまたはドライバにおいて、或いは好ましい実施形態の説明を理解した後で当業者の技術の範囲内にあるような特定のハードウェアのためにドライバを提供する他の何らかの方法で実装される。種々のソフトウェアおよびハードウェア・エミュレーションの特許には、これらに限られるものではないが、Beausoleil他による「Multiprocessor for Hardware Emulation」という名称の特許文献4、Scalzi他による「Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor」という名称の特許文献5、Davidian他による「Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions」という名称の特許文献6、Gorishchek他による「Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor

40

50

Support Allowing Non-Native Code to Run in a System」という名称の特許文献7、Lethin他による「Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method」という名称の特許文献8、Eric Trautによる「Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions」という名称の特許文献9、および他の多くが挙げられ、これらの参考文献は、当業者が利用可能なターゲット・マシンのための異なるマシン用に設計された命令形式のエミュレーションを達成する様々な既知の方法を示す。

10

【0160】

図32において、ホスト・アーキテクチャのホスト・コンピュータ・システム5000'をエミュレートする、エミュレートされたホスト・コンピュータ・システム5092の一例が提供される。エミュレートされたホスト・コンピュータ・システム5092では、ホスト・プロセッサ(CPU)5091は、エミュレートされたホスト・プロセッサ(または仮想ホスト・プロセッサ)であり、かつ、ホスト・コンピュータ5000'のプロセッサ5091のものとは異なるネイティブな命令セット・アーキテクチャを有するエミュレーション・プロセッサ5093を含む。エミュレートされたホスト・コンピュータ・システム5092は、エミュレーション・プロセッサ5093がアクセス可能なメモリ5094を有する。例示的な実施形態において、メモリ5094は、ホスト・コンピュータ・メモリ5096の部分と、エミュレーション・ルーチン5097の部分とに区分化される。ホスト・コンピュータ・メモリ5096は、ホスト・コンピュータ・アーキテクチャに従い、エミュレートされたホスト・コンピュータ・システム5092のプログラムに利用可能である。エミュレーション・プロセッサ5093は、エミュレートされたプロセッサ5091のもの以外のアーキテクチャのアーキテクチャ化された命令セットのネイティブ命令を実行し、このネイティブ命令はエミュレーション・ルーチン・メモリ5097から取得されたものであり、かつ、エミュレーション・プロセッサ5093は、シーケンスおよびアクセス/デコード・ルーチンにおいて取得される1以上の命令を用いることにより、ホスト・コンピュータ・メモリ5096の中のプログラム由来の実行のためのホスト命令にアクセスすることができ、このシーケンスおよびアクセス/デコード・ルーチンは、アクセスされたホスト命令をデコードして、アクセスされたホスト命令の機能をエミュレートするためのネイティブ命令実行ルーチンを判断することができる。ホスト・コンピュータ・システム5000'のアーキテクチャのために定められた、例えば、汎用レジスタ、制御レジスタ、動的アドレス変換、およびI/Oサブシステムのサポート、並びにプロセッサ・キャッシュといったファシリティを含む他のファシリティを、アーキテクチャ化ファシリティ・ルーチンによってエミュレートすることができる。エミュレーション・ルーチンは、エミュレーション・ルーチンの性能を高めるために、エミュレーション・プロセッサ5093において利用可能な(汎用レジスタ、および仮想アドレスの動的変換といった)機能を利用することもできる。ホスト・コンピュータ5000'の機能をエミュレートする際にプロセッサ5093を補助するために、専用のハードウェアおよびオフロード・エンジンを設けることもできる。

20

30

40

【0161】

本明細書で用いられる用語は、特定の実施形態を説明する目的のためのものにすぎず、本発明を限定することを意図したものではない。本明細書で用いられる場合、単数形「1つの(a)」、「1つの(an)」および「その(the)」は、文脈が特に明示しない限り、複数形も同様に含むことを意図したものである。「含む(comprise)」および/または「含んでいる(comprising)」という用語は、本明細書で用いられる場合、記述された特徴、整数、ステップ、動作、要素、および/またはコンポーネントの存在を指示するが、1以上の他の特徴、整数、ステップ、動作、要素、コンポーネント、および/またはそれら

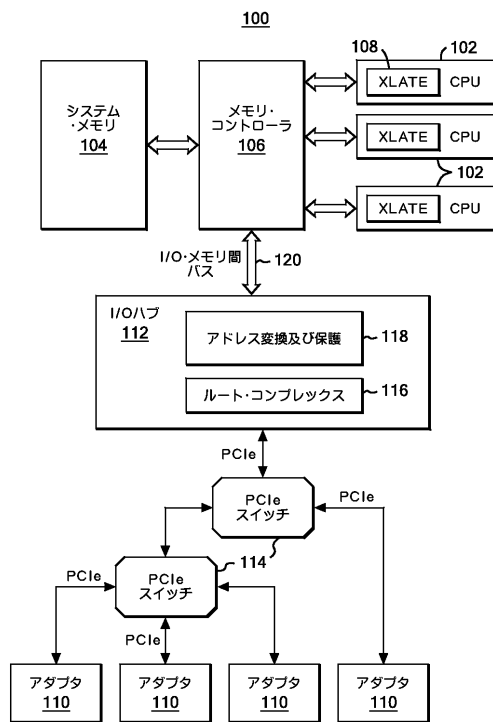
50

の群の存在または追加を排除するものではないこともさらに理解されるであろう。

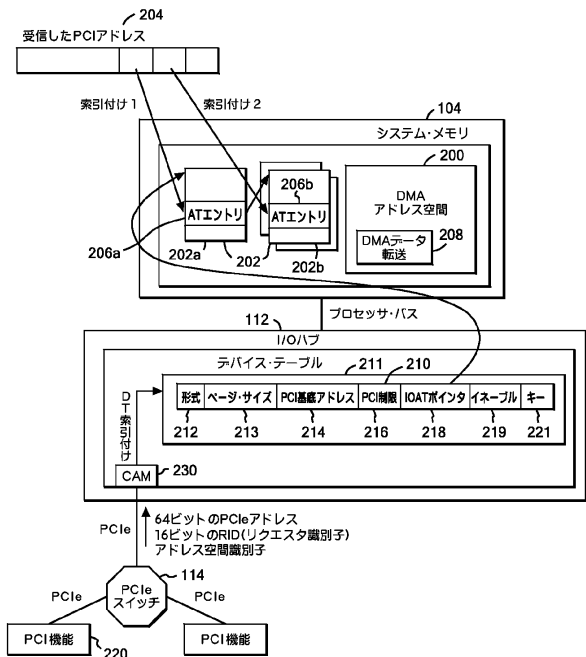
【 0 1 6 2 】

以下の特許請求の範囲に存在する場合、「手段またはステップと機能との組合せ（ミーンズまたはステップ・プラス・ファンクション）」要素の対応する構造、材料、動作および均等物は、明確に特許請求された他の請求要素と共に機能を実行するための任意の構造体、材料、または行為を含むことを意図したものである。本発明の説明は、例証および説明のためだけに提示されたものであり、網羅的であることまたは本発明を開示した形態に限定することを意図したのではない。当業者には、本発明の範囲および思想から逸脱しない多くの修正物および変形物が明らかとなるであろう。実施形態は、本発明の原理および実際の用途を最もよく説明するため、および、当業者が、企図した特定の用途に適するように種々の修正を有する種々の実施形態に関して本発明を理解することができるように、選択され記述された。

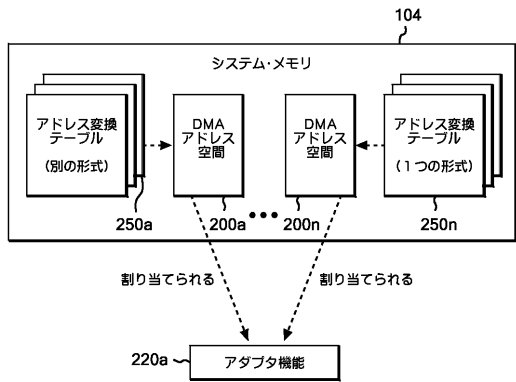
【 図 1 】



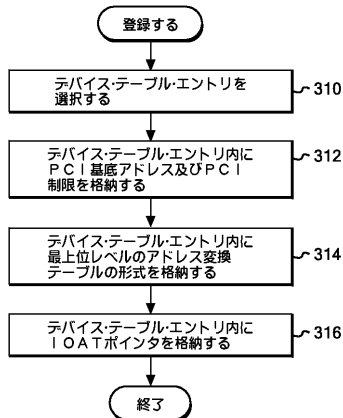
【 図 2 】



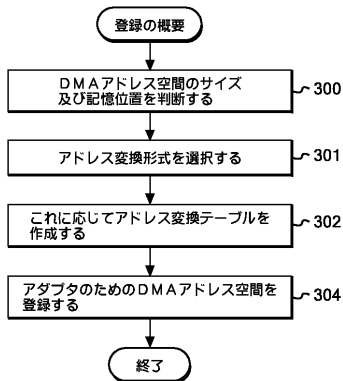
【図3】



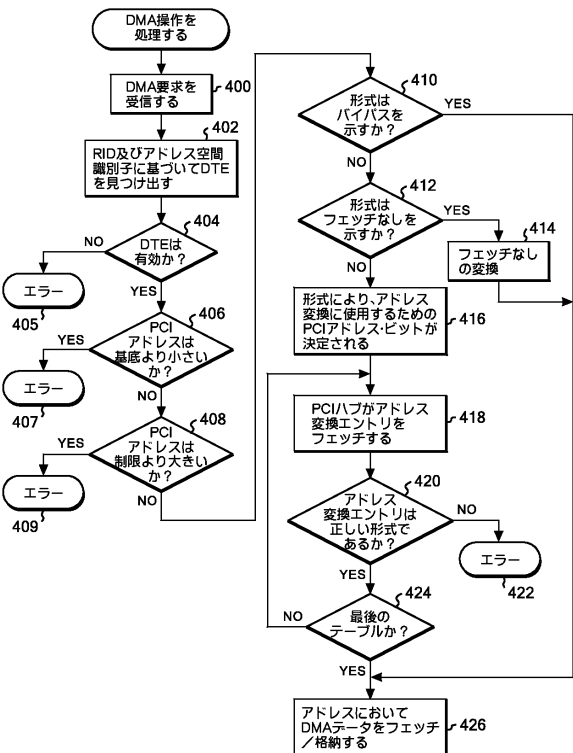
【図5】



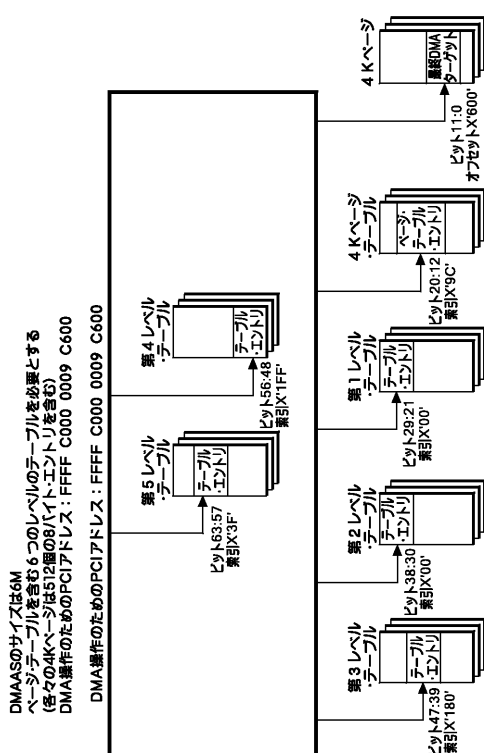
【図4】



【図6】

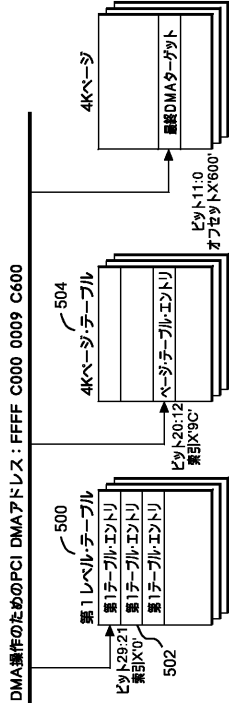


【図7】



【 図 8 】

DMAASのサイズは6M
 1つの4Kの第1レベルテーブル及び3つの4Kのページテーブルを必要とする
 (各々の4Kページは512個の8バイト・エントリを含む)
 PCI基礎アドレス: FFFF C000 0000 0000
 PCI制限アドレス: FFFF C000 005F FFFF
 DMA操作のためのPCIアドレス: FFFF C000 0009 C600
 PCI基礎/制限の子エックレにより、PCIアドレスビット68:30を無効にすることが可能になる



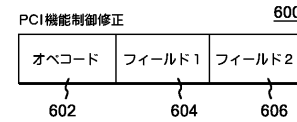
【 図 9 】

形式	11ビット	11ビット	11ビット	11ビット	11ビット	11ビット	8ビット	12ビット
4Kページ DAT互換	領域1 DMAAS 16E	領域2 DMAAS 8P	領域3 DMAAS 4T	セグメントテーブル DMAAS 2G	セグメントテーブル DMAAS 1M	ページテーブル DMAAS 4K	ページテーブル DMAAS 1M	ページテーブル DMAAS 4K
1Mページ DAT互換	領域1 DMAAS 16E	領域2 DMAAS 8P	領域3 DMAAS 4T	セグメントテーブル DMAAS 2G	セグメントテーブル DMAAS 1M	ページテーブル DMAAS 4K	ページテーブル DMAAS 1M	ページテーブル DMAAS 4K

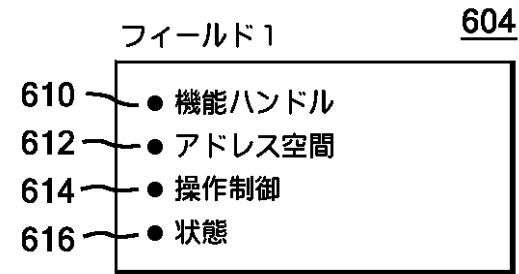
【 図 10 】

形式	7ビット	9ビット	9ビット	9ビット	9ビット	9ビット	9ビット	9ビット	12ビット
4Kページ 4K ATテーブル	第5レベル DMAAS 16E	第4レベル DMAAS 128P	第3レベル DMAAS 256T	第2レベル DMAAS 512G	第1レベル DMAAS 1G	IOPT DMAAS 2M	IOPT DMAAS 2M	IOPT DMAAS 2M	バイト・オフセット 4K
4Kページ 1M ATテーブル	第2レベル DMAAS 8E	第1レベル DMAAS 64T	第1レベル DMAAS 64T	IOPT DMAAS 512M	IOPT DMAAS 512M	IOPT DMAAS 512M	IOPT DMAAS 512M	IOPT DMAAS 512M	バイト・オフセット 4K
1Mページ 1M ATテーブル	第2レベル DMAAS 16E	第1レベル DMAAS 16P	第1レベル DMAAS 16P	IOPT DMAAS 128G	IOPT DMAAS 128G	IOPT DMAAS 128G	IOPT DMAAS 128G	IOPT DMAAS 128G	バイト・オフセット 1M

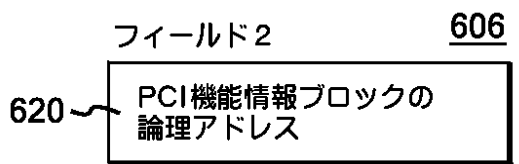
【 図 11 】



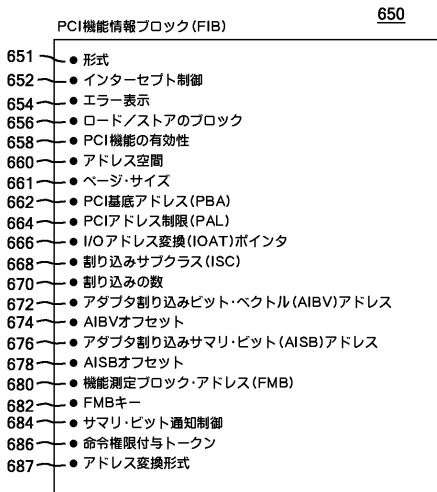
【 図 12 】



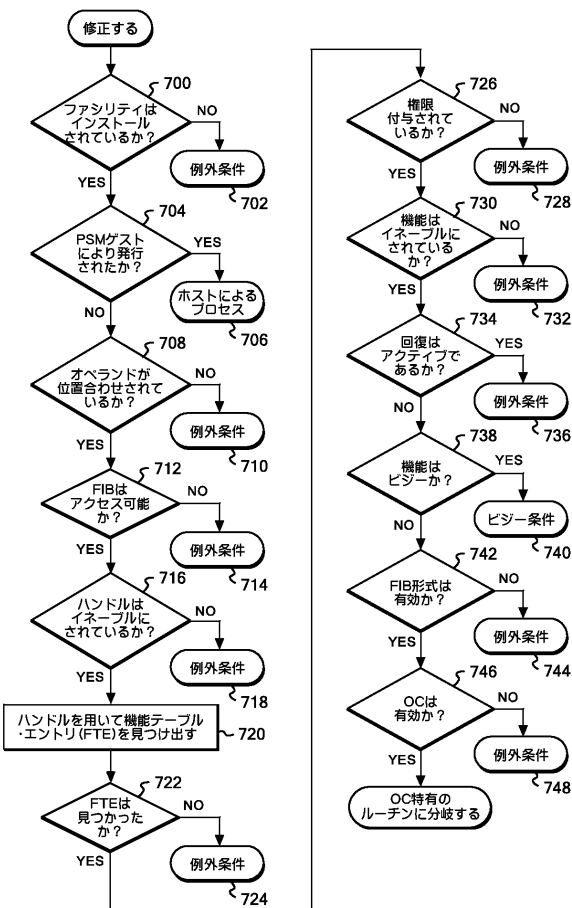
【 図 13 】



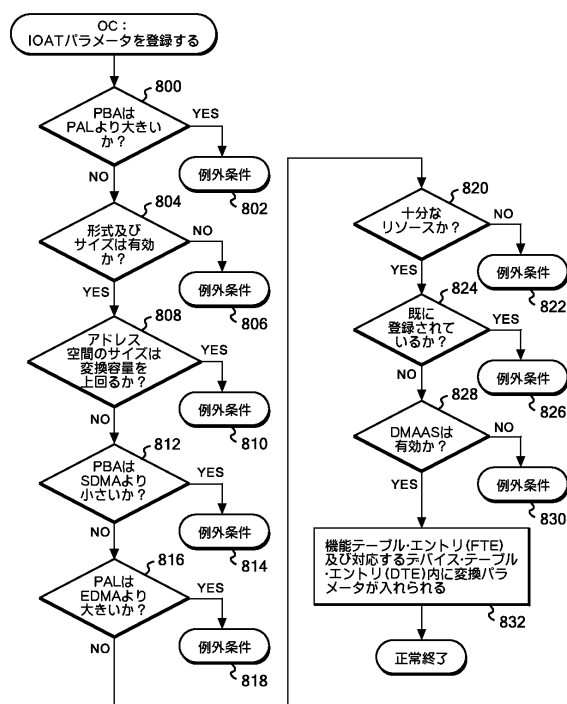
【図14】



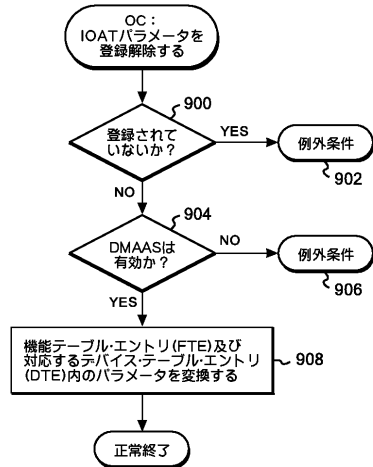
【図15】



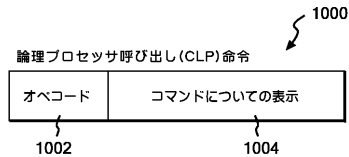
【図16】



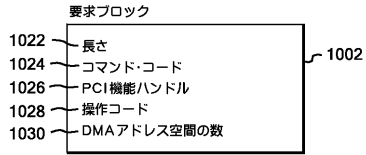
【図17】



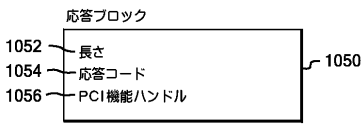
【図18】



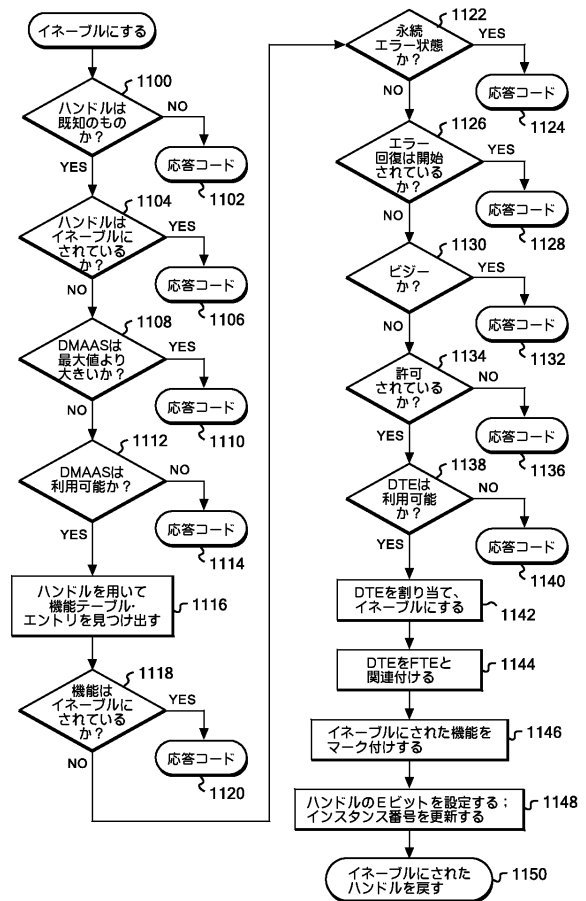
【図19】



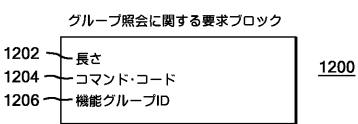
【図20】



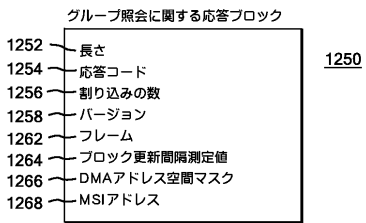
【図21】



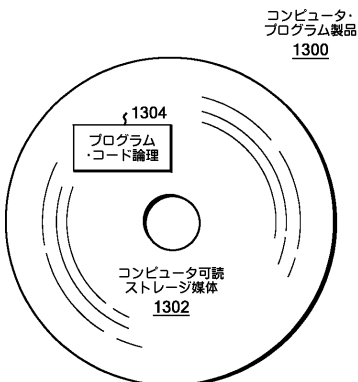
【図22】



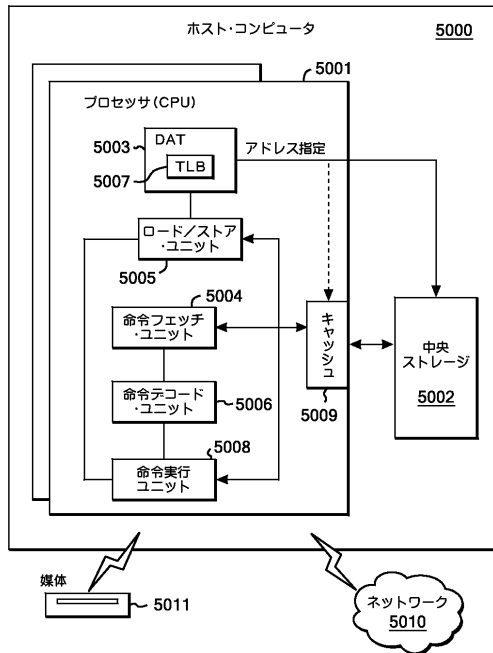
【図23】



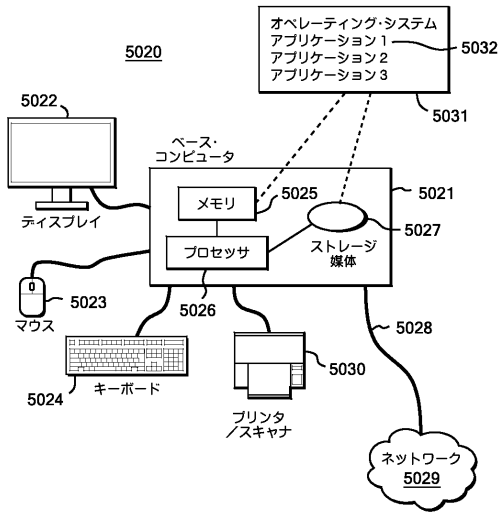
【図24】



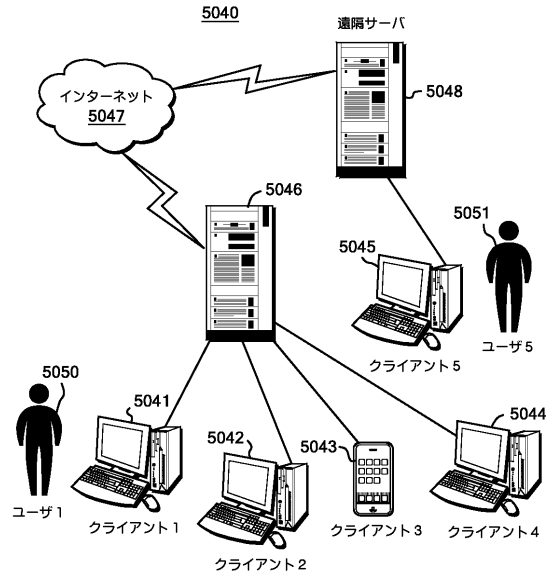
【図25】



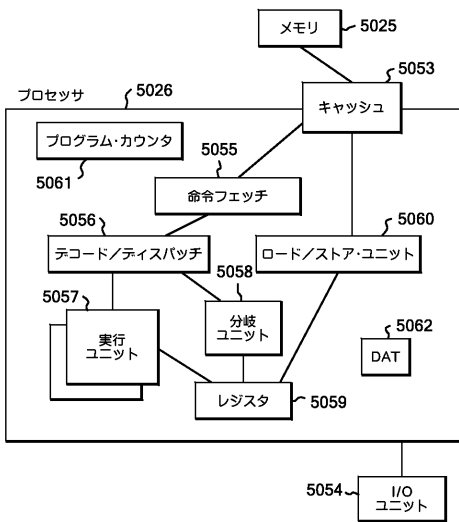
【図26】



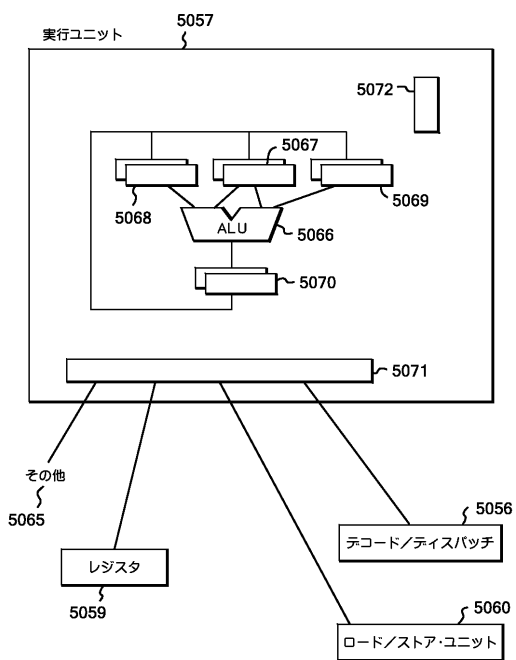
【図27】



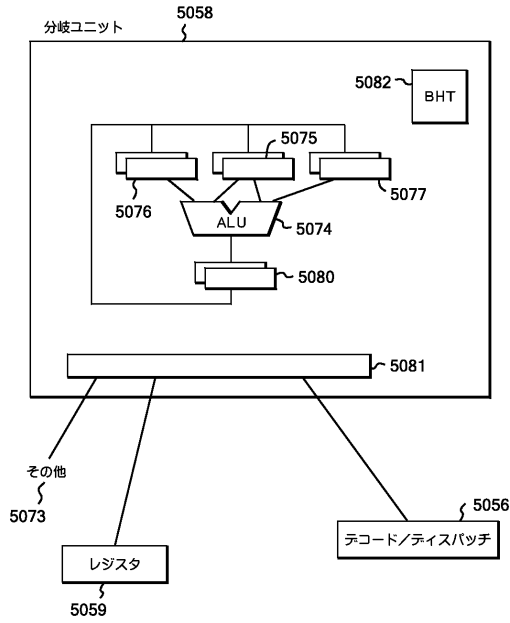
【図28】



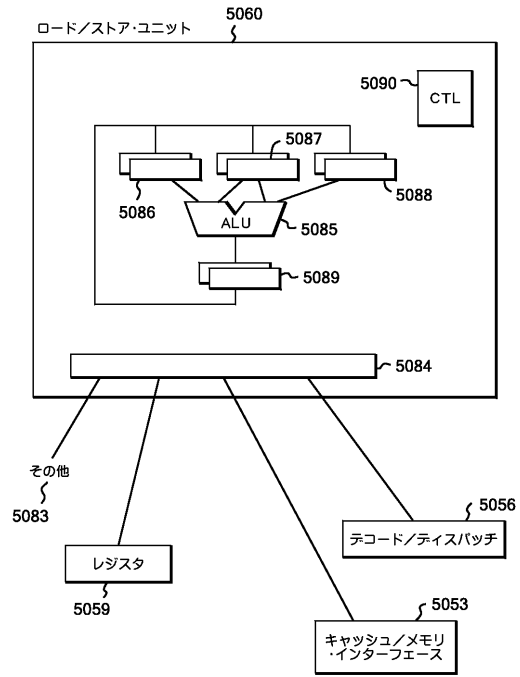
【図29】



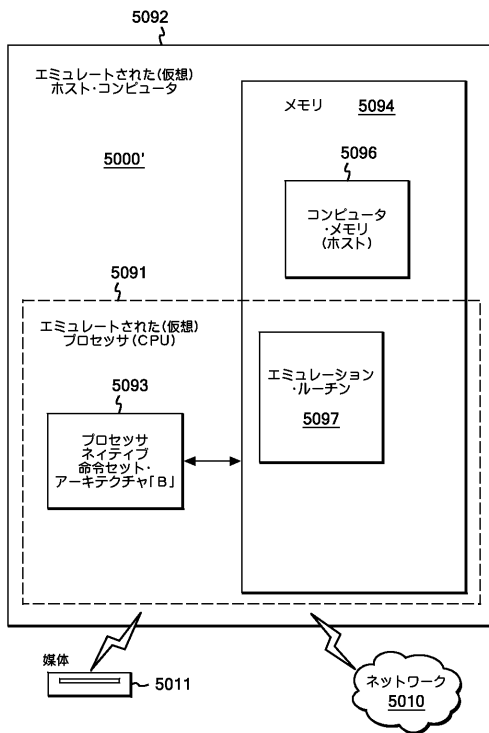
【図30】



【図31】



【図32】



フロントページの続き

- (72)発明者 クラドック、デービッド
アメリカ合衆国 12601 ニューヨーク州 ポキプシー サウス・ロード 2455 デパー
トメント イー57エー
- (72)発明者 グレッグ、トーマス
アメリカ合衆国 12601 ニューヨーク州 ポキプシー サウス・ロード 2455 ピーエ
スノピー314
- (72)発明者 ライシュ、クリストフ
ドイツ連邦共和国 71032 ベープリングエン シェーナヒャー・シュトラッセ 220 エ
ム/ディー 09-03

審査官 野田 佳邦

- (56)参考文献 特開2007-287140(JP, A)
米国特許出願公開第2008/0114906(US, A1)
米国特許出願公開第2008/0209130(US, A1)

- (58)調査した分野(Int.Cl., DB名)
G06F 12/08 - 12/12