

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5739524号
(P5739524)

(45) 発行日 平成27年6月24日(2015. 6. 24)

(24) 登録日 平成27年5月1日(2015. 5. 1)

(51) Int.Cl. F I
G O 6 F 9/30 (2006.01) G O 6 F 9/30 3 5 O B
G O 6 F 9/30 3 5 O A

請求項の数 14 (全 44 頁)

(21) 出願番号	特願2013-515723 (P2013-515723)	(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2013-536486 (P2013-536486A)		I N T E R N A T I O N A L B U S I N E S S M A C H I N E S C O R P O R A T I O N
(43) 公表日	平成25年9月19日 (2013. 9. 19)		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/EP2010/067028		
(87) 国際公開番号	W02011/160710		
(87) 国際公開日	平成23年12月29日 (2011. 12. 29)	(74) 代理人	100108501
審査請求日	平成25年8月6日 (2013. 8. 6)		弁理士 上野 剛史
(31) 優先権主張番号	12/821, 194	(74) 代理人	100112690
(32) 優先日	平成22年6月23日 (2010. 6. 23)		弁理士 太佐 種一
(33) 優先権主張国	米国 (US)	(74) 代理人	100091568
早期審査対象出願			弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 アダプタと通信するための保存／保存ブロック命令

(57) 【特許請求の範囲】

【請求項 1】

アダプタにデータを保存するための保存命令を実行するための方法であって、前記方法は、

実行のための機械命令を得るステップを含み、前記機械命令はコンピュータ・アーキテクチャに従うコンピュータ実行のために規定されており、前記機械命令は、

前記機械命令がアダプタへの保存命令であることを識別するオペコード・フィールドと

、
アダプタに保存されるべきデータを含む第1の場所を識別する第1のフィールドと、
第2の場所を識別する第2のフィールドであって、前記第2の場所の内容は、前記アダプタを識別する関数ハンドル、データを保存すべき前記アダプタ内のアドレス空間の指示、および前記アドレス空間内のオフセットを含む、第2のフィールドとを含み、前記方法はさらに、

前記機械命令を実行するステップを含み、前記実行するステップは、
前記関数ハンドルを用いて前記アダプタに関連する関数表エントリを得るステップと、
前記関数表エントリ内の情報および前記オフセットの少なくとも1つを用いて前記アダプタのデータ・アドレスを得るステップと、

前記アドレス空間の前記指示によって識別された前記アドレス空間内の特定の場所に前記第1の場所からのデータを保存するステップとを含み、前記特定の場所は前記アダプタの前記データ・アドレスによって識別される、方法。

10

20

【請求項 2】

アクセスされる前記アドレス空間はメモリ空間または I/O 空間の一方であり、前記データ・アドレスを得るステップは、前記関数表エントリの 1 つまたはそれ以上のパラメータを用いて前記データ・アドレスを得るステップを含む、請求項 1 に記載の方法。

【請求項 3】

前記 1 つまたはそれ以上のパラメータを用いるステップは、前記関数表エントリの基底アドレス・レジスタの値を前記オフセットに加えて前記データ・アドレスを得るステップを含む、請求項 2 に記載の方法。

【請求項 4】

アクセスされる前記アドレス空間は構成空間であり、前記データ・アドレスは前記保存命令によって提供される前記オフセットであり、前記オフセットは前記構成空間内のレジスタ番号を識別する、請求項 1 に記載の方法。

10

【請求項 5】

前記第 2 の場所の前記内容は、保存されるべきデータの量を含む、請求項 1 に記載の方法。

【請求項 6】

前記保存命令は前記アダプタのアーキテクチャに基づいて実装される、請求項 1 に記載の方法。

【請求項 7】

アダプタにデータを保存するための保存命令を実行するためのコンピュータ・システムであって、前記コンピュータ・システムは、

20

メモリと、

前記メモリと通信するプロセッサとを含み、

前記プロセッサは実行のための機械命令を得、前記機械命令はコンピュータ・アーキテクチャに従うコンピュータ実行のために規定されており、前記機械命令は、

前記機械命令がアダプタへの保存命令であることを識別するオペコード・フィールドと

、
アダプタに保存されるべきデータを含む第 1 の場所を識別する第 1 のフィールドと、
第 2 の場所を識別する第 2 のフィールドであって、前記第 2 の場所の内容は前記アダプタを識別する関数ハンドル、データを保存すべき前記アダプタ内のアドレス空間の指示、および前記アドレス空間内のオフセットを含む、第 2 のフィールドとを含み、

30

前記プロセッサは前記機械命令を実行し、前記実行は、

前記関数ハンドルを用いて前記アダプタに関連する関数表エントリを得ることと、

前記関数表エントリ内の情報および前記オフセットの少なくとも 1 つを用いて前記アダプタのデータ・アドレスを得ることと、

前記アドレス空間の前記指示によって識別された前記アドレス空間内の特定の場所に前記第 1 の場所からのデータを保存することとを含み、前記特定の場所は前記アダプタの前記データ・アドレスによって識別される、コンピュータ・システム。

【請求項 8】

アダプタにデータを保存するための保存命令を実行するためのコンピュータ・システムであって、前記コンピュータ・システムは、

40

メモリと、

前記メモリと通信するプロセッサとを含み、前記コンピュータ・システムは方法を行うように構成され、前記方法は、

実行のための機械命令を得るステップを含み、前記機械命令はコンピュータ・アーキテクチャに従うコンピュータ実行のために規定されており、前記機械命令は、

前記機械命令がアダプタへの保存命令であることを識別するオペコード・フィールドと

、
アダプタに保存されるべきデータを含む第 1 の場所を識別する第 1 のフィールドと、
第 2 の場所を識別する第 2 のフィールドであって、前記第 2 の場所の内容は前記アダプ

50

タを識別する関数ハンドル、データを保存すべき前記アダプタ内のアドレス空間の指示、および前記アドレス空間内のオフセットを含む、第2のフィールドとを含み、前記方法はさらに、

前記機械命令を実行するステップを含み、前記実行するステップは、

前記関数ハンドルを用いて前記アダプタに関連する関数表エントリを得るステップと、

前記関数表エントリ内の情報および前記オフセットの少なくとも1つを用いて前記アダプタのデータ・アドレスを得るステップと、

前記アドレス空間の前記指示によって識別された前記アドレス空間内の特定の場所に前記第1の場所からのデータを保存するステップとを含み、前記特定の場所は前記アダプタの前記データ・アドレスによって識別される、コンピュータ・システム。

10

【請求項9】

アクセスされる前記アドレス空間はメモリ空間またはI/O空間の一方であり、前記データ・アドレスを得るステップは、前記関数表エントリの1つまたはそれ以上のパラメータを用いて前記データ・アドレスを得るステップを含む、請求項8に記載のコンピュータ・システム。

【請求項10】

前記1つまたはそれ以上のパラメータを用いるステップは、前記関数表エントリの基底アドレス・レジスタの値を前記オフセットに加えて前記データ・アドレスを得るステップを含む、請求項9に記載のコンピュータ・システム。

【請求項11】

20

アクセスされる前記アドレス空間は構成空間であり、前記データ・アドレスは前記保存命令によって提供される前記オフセットであり、前記オフセットは前記構成空間内のレジスタ番号を識別する、請求項8に記載のコンピュータ・システム。

【請求項12】

前記第2の場所の前記内容は、保存されるべきデータの量を含む、請求項8に記載のコンピュータ・システム。

【請求項13】

前記保存命令は前記アダプタのアーキテクチャに基づいて実装される、請求項8に記載のコンピュータ・システム。

【請求項14】

30

アダプタにデータを保存するための保存命令を実行するためのコンピュータ・プログラムを格納したコンピュータ読取可能な記録媒体であって、前記記録媒体は、

処理回路によって読取り可能であり、かつ方法を行うための、前記処理回路による実行のための命令を保存するコンピュータ読取可能な記録媒体であり、前記方法は、

実行のための機械命令を得るステップを含み、前記機械命令はコンピュータ・アーキテクチャに従うコンピュータ実行のために規定されており、前記機械命令は、

前記機械命令がアダプタへの保存命令であることを識別するオペコード・フィールドと

、

アダプタに保存されるべきデータを含む第1の場所を識別する第1のフィールドと、

40

第2の場所を識別する第2のフィールドであって、前記第2の場所の内容は前記アダプタを識別する関数ハンドル、データを保存すべき前記アダプタ内のアドレス空間の指示、および前記アドレス空間内のオフセットを含む、第2のフィールドとを含み、前記方法はさらに、

前記機械命令を実行するステップを含み、前記実行するステップは、

前記関数ハンドルを用いて前記アダプタに関連する関数表エントリを得るステップと、

前記関数表エントリ内の情報および前記オフセットの少なくとも1つを用いて前記アダプタのデータ・アドレスを得るステップと、

前記アドレス空間の前記指示によって識別された前記アドレス空間内の特定の場所に前記第1の場所からのデータを保存するステップとを含み、前記特定の場所は前記アダプタ

50

の前記データ・アドレスによって識別される、記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は一般的にコンピューティング環境の入出力処理に関し、特定的にはコンピューティング環境のアダプタとの通信を促進することに関する。

【背景技術】

【0002】

コンピューティング環境は、さまざまなタイプのアダプタを含む1つまたはそれ以上のタイプの入出力デバイスを含み得る。アダプタの1つのタイプは、周辺構成要素相互接続 (Peripheral Component Interconnect: PCI) または周辺構成要素相互接続エクスプレス (Peripheral Component Interconnect Express: PCIe) アダプタである。このアダプタは、アダプタとそのアダプタが取り付けられているシステムとの間のデータ通信に用いられる1つまたはそれ以上のアドレス空間を含む。PCIの仕様はワールド・ワイド・ウェブの www.pcisig.com/home より入手可能である。

【0003】

2004年3月9日に発行されたエイヴリー (Avery) の特許文献1「PCIバス・プロトコルとメッセージをパスするキュー指向のバス・プロトコルとの間でアドレス情報を変換するための方法および装置 (Method and Apparatus for Converting Address Information Between PCI Bus Protocol and a Message-Passing Queue-Oriented Bus Protocol)」は、PCIロード/保存動作およびDMA動作が、メッセージをパスするキュー指向のバス・アーキテクチャにおける作業キュー対を介して実行されることを記載している。PCIアドレス空間は複数のセグメントに分割され、続いて各セグメントは複数の領域に分割される。別個の作業キューが各セグメントに割り当てられる。PCIアドレスの第1の部分は、セグメントによって表されるアドレス範囲に適合され、メモリ・セグメントおよびその対応する作業キューを選択するために用いられる。作業キュー内のエントリは、特定のPCIデバイスに割り当てられた選択セグメント内の領域を指定するPCIアドレスの第2の部分を保持する。一実施形態において、PIOロード/保存動作は、PIO動作に割り当てられた作業キューを選択し、PCIデバイス上のレジスタのPCIアドレスとPIOデータに対するポインタとによって作業キュー・エントリを作成することによって実行される。作業キュー・エントリはPCIブリッジに送られ、そこでPCIアドレスが抽出され、データ・ポインタを用いてそのデータによって適切なデバイス・レジスタをプログラムするために用いられる。DMA転送も、PCIデバイスによって生成されるPCIアドレスの部分をアドレス範囲表と比較する手段によって作業キューを選択し、そのアドレス範囲を処理する作業キューを選択することによって実行される。PCIアドレスの残余とDMAデータに対するポインタとによって作業キュー・エントリが作成される。RDMA動作を用いてDMA転送が行われる。ページおよび領域データが、ホスト・チャンネル・アダプタ内の変換保護表と関連して用いられることにより、物理的メモリにアクセスしてDMA転送が行われる。

【0004】

2009年11月3日に発行されたキヨス (Kjos) らの特許文献2「仮想マシンによる使用のためにI/Oデバイスを部分的に仮想化する (Partially Virtualizing an I/O Device for Use by Virtual Machines)」に記載されるコンピュータ・システムは、物理的コンピュータと、物理的コンピュータを制御するように適合された少なくとも1つのゲスト・オペレーティング・システムのエミュレーションを作成するように構成された、物理的コンピュータ上で実行可能な仮想マシン・モニタとを含む。このコンピュータ・システムはさらに、

仮想マシン・モニタおよび少なくとも1つのゲスト・オペレーティング・システムの代わりに物理的コンピュータに結合された物理リソースを管理する、物理的コンピュータ上で実行可能なホストを含む。このホストは周辺構成要素相互接続（P C I）構成アドレス空間を仮想化するように適合されており、それによって少なくとも1つのゲスト・オペレーティング・システムは、I / Oエミュレーションの不在下で直接P C I入出力（input / output : I / O）デバイスを制御する。

【0005】

いくつかのシステムにおいては、アダプタに結合される中央処理ユニット（central processing unit : CPU）のアドレス空間の部分がアダプタのアドレス空間にマップされることによって、記憶装置にアクセスするCPU命令がアダプタのアドレス空間内のデータを直接操作することが可能になる。

10

【先行技術文献】

【特許文献】

【0006】

【特許文献1】米国特許第6,704,831号

【特許文献2】米国特許第7,613,847号

【発明の概要】

【発明が解決しようとする課題】

【0007】

本発明は、P C IまたはP C I e アダプタなどのアダプタとの通信を促進するための能力を提供する。さらに、アダプタとデータをやり取りするために特定の設計された制御命令を提供する。これらは、通信に用いられる。

20

【課題を解決するための手段】

【0008】

アダプタにデータを保存するための保存命令を実行するためのコンピュータ・プログラム製品の提供によって、先行技術の欠点が克服されて利点を提供される。このコンピュータ・プログラム製品は、処理回路によって読取り可能であり、かつある方法を行うために処理回路によって実行されるための命令を保存するコンピュータ読取り可能記憶媒体を含む。この方法は、たとえば実行のための機械命令を得るステップを含み、この機械命令はコンピュータ・アーキテクチャに従うコンピュータ実行のために規定されており、機械命令はたとえば、アダプタへの保存命令（a store to adapter instruction）を識別するオペコード・フィールドと、アダプタに保存されるべきデータを含む第1の場所を識別する第1のフィールドと、第2の場所を識別する第2のフィールドとを含み、第2のフィールドの内容はアダプタを識別する関数ハンドル、データを保存すべきアダプタ内のアドレス空間の指示、およびアドレス空間内のオフセットを含み、この方法はさらに機械命令を実行するステップを含み、この実行するステップは、関数ハンドルを用いてアダプタに関連する関数表エントリを得るステップと、関数表エントリ内の情報およびオフセットの少なくとも1つを用いてアダプタのデータ・アドレスを得るステップと、アドレス空間の指示によって識別されたアドレス空間内の特定の場所に第1の場所からのデータを保存するステップとを含み、この特定の場所はアダプタのデータ・アドレスによって識別される。

30

40

【0009】

さらに、アダプタにデータを保存するための保存ブロック命令を実行するためのコンピュータ・プログラム製品が提供される。このコンピュータ・プログラム製品は、処理回路によって読取り可能であり、かつある方法を行うために処理回路によって実行されるための命令を保存するコンピュータ読取り可能記憶媒体を含む。この方法は、たとえば実行のための機械命令を得るステップを含み、この機械命令はコンピュータ・アーキテクチャに従うコンピュータ実行のために規定されており、機械命令はたとえば、アダプタへの保存ブロック命令（a store block to adapter instruction）を識別するオペコード・フィールドと、第1の場所を識別する第1のフィールド

50

であって、第1のフィールドの内容はアダプタを識別する関数ハンドル、およびデータを保存すべきアダプタ内のアドレス空間の指示を含む、第1のフィールドと、アドレス空間内のオフセットを含む第2の場所を識別する第2のフィールドと、アダプタに保存されるべきデータを含むメモリ中のアドレスを含む第3の場所を識別する第3のフィールドとを含み、この方法はさらに機械命令を実行するステップを含み、この実行するステップは、関数ハンドルを用いてアダプタに関連する関数表エントリを得るステップと、関数表エントリ内の情報およびオフセットを用いてアダプタのデータ・アドレスを得るステップと、アドレス空間の指示によって識別されたアドレス空間内の特定の場所にメモリの第3のフィールド内のアドレスから得られたデータを保存するステップとを含み、この特定の場所はアダプタのデータ・アドレスによって識別される。

10

【0010】

本発明の1つまたはそれ以上の局面に係する方法およびシステムも本明細書において説明および請求されている。

【0011】

本発明の技術によって、付加的な特徴および利点の実現される。本発明の他の実施形態および局面が本明細書に詳細に説明されており、それらも請求される発明の部分とみなされる。

【0012】

本明細書の終わりの請求項において、本発明の1つまたはそれ以上の局面が例として特定の指摘され明確に請求されている。本発明の前述およびその他の目的、特徴および利点は、添付の図面とともに提供される以下の詳細な説明から明らかとなる。

20

【図面の簡単な説明】

【0013】

【図1】本発明の1つまたはそれ以上の局面を組み込んで使用するためのコンピューティング環境の一実施形態を示す図である。

【図2】本発明の局面に従って使用される、図1のI/Oハブ内に位置するデバイス表エントリの一実施形態を示す図である。

【図3】本発明の1つまたはそれ以上の局面を組み込んで使用するためのコンピューティング環境の別の実施形態を示す図である。

【図4】本発明の局面に従う、アダプタ機能のアドレス空間の一例を示す図である。

30

【図5】図5Aは、本発明の局面に従って使用される関数表エントリの一例を示す図である。図5Bは、本発明の局面に従って使用される関数ハンドルの一実施形態を示す図である。

【図6】図6Aは、本発明の局面に従って使用されるPCIロード命令の一実施形態を示す図である。図6Bは、本発明の局面に従う、図6AのPCIロード命令によって使用されるフィールドによって示される場所の内容の一実施形態を示す図である。図6Cは、本発明の局面に従う、図6AのPCIロード命令によって使用される別のフィールドによって示される場所の内容の一実施形態を示す図である。

【図7】本発明の局面に従う、PCIロード動作を行うための論理の一実施形態を示す図である。

40

【図8】本発明の局面に従う、PCIロード動作を行うための論理の一実施形態を示す図である。

【図9】図9Aは、本発明の局面に従って使用されるPCI保存命令の一実施形態を示す図である。図9Bは、本発明の局面に従う、図9AのPCI保存命令によって使用されるフィールドによって示される場所の内容の一実施形態を示す図である。図9Cは、本発明の局面に従う、図9AのPCI保存命令によって使用される別のフィールドによって示される場所の内容の一実施形態を示す図である。

【図10】本発明の局面に従う、PCI保存動作を行うための論理の一実施形態を示す図である。

【図11】本発明の局面に従う、PCI保存動作を行うための論理の一実施形態を示す図

50

である。

【図 1 2】図 1 2 A は、本発明の局面に従って使用される P C I 保存ブロック命令の一実施形態を示す図である。図 1 2 B は、本発明の局面に従う、図 1 2 A の P C I 保存ブロック命令によって使用されるフィールドによって示される場所の内容の一実施形態を示す図である。図 1 2 C は、本発明の局面に従う、図 1 2 A の P C I 保存ブロック命令によって使用される別のフィールドによって示される場所の内容の一実施形態を示す図である。図 1 2 D は、本発明の局面に従う、図 1 2 A の P C I 保存ブロック命令によって使用されるさらに別のフィールドによって示される場所の内容の一実施形態を示す図である。

【図 1 3】本発明の局面に従う、P C I 保存ブロック動作を行うための論理の一実施形態を示す図である。

10

【図 1 4】本発明の局面に従う、P C I 保存ブロック動作を行うための論理の一実施形態を示す図である。

【図 1 5】本発明の 1 つまたはそれ以上の局面を組み込むコンピュータ・プログラム製品の一実施形態を示す図である。

【図 1 6】本発明の 1 つまたはそれ以上の局面を組み込んで使用するためのホスト・コンピュータ・システムの一実施形態を示す図である。

【図 1 7】本発明の 1 つまたはそれ以上の局面を組み込んで使用するためのコンピュータ・システムのさらなる例を示す図である。

【図 1 8】本発明の 1 つまたはそれ以上の局面を組み込んで使用す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である。

20

【図 1 9】本発明の 1 つまたはそれ以上の局面を組み込んで使用するためのコンピュータ・システムのさまざまな構成要素の一実施形態を示す図である。

【図 2 0】本発明の 1 つまたはそれ以上の局面を組み込んで使用するための、図 1 9 のコンピュータ・システムの実行ユニットの一実施形態を示す図である。

【図 2 1】本発明の 1 つまたはそれ以上の局面を組み込んで使用するための、図 1 9 のコンピュータ・システムの分岐ユニットの一実施形態を示す図である。

【図 2 2】本発明の 1 つまたはそれ以上の局面を組み込んで使用するための、図 1 9 のコンピュータ・システムのロード / 保存ユニットの一実施形態を示す図である。

【図 2 3】本発明の 1 つまたはそれ以上の局面を組み込んで使用するためのエミュレート後のホスト・コンピュータ・システムの一実施形態を示す図である。

30

【発明を実施するための形態】

【0014】

本発明の局面に従うと、コンピューティング環境のアダプタとの通信を促進するための 1 つまたはそれ以上の制御命令が提供される。この制御命令は、アダプタのアドレス空間へ、およびアダプタのアドレス空間からデータを通信するために特定の設計されている。

【0015】

本明細書において用いられるアダプタという用語は、あらゆるタイプのアダプタ（例、記憶装置アダプタ、ネットワーク・アダプタ、処理アダプタ、P C I アダプタ、暗号アダプタ、その他のタイプの入出力アダプタなど）を含む。一実施形態において、アダプタは 1 つのアダプタ機能を含む。しかし他の実施形態において、アダプタは複数のアダプタ機能を含んでもよい。アダプタが 1 つのアダプタ機能を含んでいても、複数のアダプタ機能を含んでいても、本発明の 1 つまたはそれ以上の局面を適用可能である。一実施形態において、もしアダプタが複数のアダプタ機能を含んでいれば、各機能が本発明の局面に従って通信されてもよい。さらに、本明細書に提供される実施例においては、別様に示されない限り、アダプタはアダプタ機能（例、P C I 機能）と互いに交換可能に用いられる。

40

【0016】

図 1 を参照して、本発明の 1 つまたはそれ以上の局面を組み込んで使用するためのコンピューティング環境の一実施形態を説明する。一例において、コンピューティング環境 100 は、インターナショナル・ビジネス・マシーンズ社（International

50

Business Machines Corporation)より提供されるSystem z (IBM社の登録商標)サーバである。System z (IBM社の登録商標)は、インターナショナル・ビジネス・マシーンズ社より提供されるz/Architecture (IBM社の登録商標)に基づいている。z/Architecture (IBM社の登録商標)に関する詳細は、「z/Architecture (IBM社の登録商標)動作の原理 (z/Architecture Principles of Operation)」と題する2009年2月、IBM (IBM社の登録商標)パブリケーションNo. SA22-7832-07のIBM (IBM社の登録商標)出版物に記載されている。IBM (IBM社の登録商標)、System z (IBM社の登録商標)およびz/Architecture (IBM社の登録商標)は、ニューヨーク州アーモニックのインターナショナル・ビジネス・マシーンズ社の登録商標である。本明細書に用いられる他の名前は、インターナショナル・ビジネス・マシーンズ社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。

10

【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 (主メモリとも呼ばれる) に結合される1つまたはそれ以上の中央処理ユニット (CPU) 102を含む。システム・メモリ104にアクセスするために、中央処理ユニット102は、システム・メモリにアクセスするために用いられるアドレスを含む読取りまたは書込み要求を発行する。その要求に含まれるアドレスは、典型的にシステム・メモリにアクセスするために直接使用できないため、システム・メモリのアクセスに直接使用可能なアドレスに変換される。そのアドレスは変換機構 (X L A T E) 108を介して変換される。たとえば、アドレスはたとえば動的アドレス変換 (dynamic address translation: D A T) など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。

20

【0018】

(必要であれば変換された) アドレスを含む要求は、メモリ・コントローラ106に受け取られる。一例において、メモリ・コントローラ106はハードウェアで構成されており、システム・メモリへのアクセスを仲裁してメモリの整合性を維持するために用いられる。この仲裁はCPU102から受け取った要求に対しても行われるし、1つまたはそれ以上のアダプタ110から受け取った要求に対しても行われる。中央処理ユニットと同様に、アダプタはシステム・メモリ104に対する要求を発行してシステム・メモリへのアクセスを得る。

30

【0019】

一例において、アダプタ110は、1つまたはそれ以上のP C I機能を含む周辺構成要素相互接続 (P C I) またはP C Iエクスプレス (P C I e) アダプタである。P C I機能は要求を発行し、その要求は1つまたはそれ以上のスイッチ (例、P C I eスイッチ) 114を経由して入出力ハブ112 (例、P C Iハブ) に送られる。一例において、入出力ハブは1つまたはそれ以上の状態機械を含むハードウェアで構成されており、I / O - メモリ・バス120を介してメモリ・コントローラ106に結合されている。

【0020】

40

入出力ハブは、たとえばスイッチからの要求を受け取るルート複合体 (root complex) 116を含む。この要求は入出力アドレスを含み、入出力アドレスはその要求に対して用いられる情報にアクセスするアドレス変換および保護ユニット118に提供される。例として、この要求は、直接メモリ・アクセス (direct memory access: D M A) 動作を行うため、またはメッセージ信号割り込み (message signaled interruption: M S I) を要求するために用いられる入出力アドレスを含んでもよい。アドレス変換および保護ユニット118は、D M AまたはM S I要求のために用いられる情報にアクセスする。特定の例として、D M A動作に対しては、アドレスを変換するために情報が得られてもよい。変換されたアドレスは、次いでシステム・メモリにアクセスするためにメモリ・コントローラに転送される。

50

【 0 0 2 1 】

一例においては、図 2 を参照して説明されるとおり、アダプタによって発行される D M A または M S I 要求のために用いられる情報は、I / O ハブ内（例、アドレス変換および保護ユニット内）に位置するデバイス表 1 3 2 のデバイス表エントリ 1 3 0 から得られる。デバイス表エントリはアダプタに対する情報を含み、各アダプタは自身に関連する少なくとも 1 つのデバイス表エントリを有する。たとえば、（システム・メモリ内の）アドレス空間当り 1 つのデバイス表エントリがアダプタに割り当てられる。アダプタ（例、P C I 機能 1 3 8）から発行された要求に対して、その要求内に与えられた要求元 I D を用いてデバイス表エントリが探し出される。

【 0 0 2 2 】

次に図 3 を参照すると、コンピューティング環境のさらなる実施形態においては、1 つまたはそれ以上の C P U 1 0 2 に加えて、またはその代わりに、中央処理複合体がメモリ・コントローラ 1 0 6 に結合される。この例において、中央処理複合体 1 5 0 はたとえば 1 つまたはそれ以上の区画またはゾーン 1 5 2（例、論理区画（l o g i c a l p a r t i t i o n s）L P 1 ~ L P n）と、1 つまたはそれ以上の中央プロセッサ（c e n t r a l p r o c e s s o r s）（例、C P 1 ~ C P m）1 5 4 と、ハイパーバイザ 1 5 6（例、論理区画マネージャ）とを含み、これら各々を以下に説明する。

【 0 0 2 3 】

各論理区画 1 5 2 は別個のシステムとして機能できる。すなわち、各論理区画を独立にリセットし、所望であればオペレーティング・システムまたはハイパーバイザ（たとえばニューヨーク州アーモックのインターナショナル・ビジネス・マシーンズ社より提供される z / V M（I B M 社の登録商標）など）で初期ロードし、異なるプログラムで動作させることができる。論理区画内で実行するオペレーティング・システム、ハイパーバイザまたはアプリケーション・プログラムは、全体の完全なシステムへのアクセスを有するよう見えるが、その一部しか利用できない。ハードウェアとライセンス内部コード（マイクロコードまたはミリコードとも呼ばれる）との組み合わせは、論理区画内のプログラムが異なる論理区画内のプログラムに干渉しないようにする。これによって、いくつかの異なる論理区画がタイム・スライスの態様で単一または複数の物理プロセッサ上で動作することが可能になる。この特定の例において、各論理区画は常駐オペレーティング・システム 1 5 8 を有し、この常駐オペレーティング・システム 1 5 8 は 1 つまたはそれ以上の論理区画に対して異なってもよい。一実施形態において、オペレーティング・システム 1 5 8 はニューヨーク州アーモックのインターナショナル・ビジネス・マシーンズ社より提供される z / O S（I B M 社の登録商標）または z L i n u x（登録商標）オペレーティング・システムである。z / O S（I B M 社の登録商標）および z / V M（I B M 社の登録商標）は、ニューヨーク州アーモックのインターナショナル・ビジネス・マシーンズ社の登録商標である。

【 0 0 2 4 】

中央プロセッサ 1 5 4 は、論理区画に割り当てられた物理プロセッサ・リソースである。たとえば論理区画 1 5 2 は 1 つまたはそれ以上の論理プロセッサを含み、論理プロセッサの各々はその区画に割り当てられた物理プロセッサ・リソース 1 5 4 のすべてまたは共用を表す。基礎をなすプロセッサ・リソースはその区画専用であっても、別の区画と共用されていてもよい。

【 0 0 2 5 】

論理区画 1 5 2 は、プロセッサ 1 5 4 上で実行するファームウェアによって実現されるハイパーバイザ 1 5 6 によって管理される。論理区画 1 5 2 およびハイパーバイザ 1 5 6 の各々は、中央プロセッサに関連する中央記憶装置のそれぞれの部分に存在する 1 つまたはそれ以上のプログラムを含む。ハイパーバイザ 1 5 6 の一例は、ニューヨーク州アーモックのインターナショナル・ビジネス・マシーンズ社より提供されるプロセッサ・リソース / システムズ・マネージャ（P r o c e s s o r R e s o u r c e / S y s t e m s M a n a g e r : P R / S M）（I B M 社の商標）である。

10

20

30

40

50

【 0 0 2 6 】

本明細書において用いられるファームウェアは、たとえばプロセッサのマイクロコード、ミリコードもしくはマクロコードまたはその組み合わせなどを含む。それはたとえばハードウェアレベルの命令、もしくはもっと高レベルの機械コードの実現に用いられるデータ構造、またはその組み合わせなどを含む。一実施形態において、ファームウェアはたとえば、基礎をなすハードウェアに対して特定の信頼できるソフトウェアまたはマイクロコードを含み、かつシステム・ハードウェアへのオペレーティング・システム・アクセスを制御するマイクロコードとして典型的に配信される商標コード (p r o p r i e t a r y c o d e) などを含む。

【 0 0 2 7 】

この例においては論理区画を有する中央処理複合体が記載されているが、本発明の1つまたはそれ以上の局面は、分割されていない単一プロセッサまたはマルチプロセッサ処理ユニットなどを含む他の処理ユニットに組み込まれて使用されてもよい。本明細書に記載される中央処理複合体は単なる一例である。

【 0 0 2 8 】

上述のとおり、アダプタはプロセッサに要求を発行して、たとえば直接メモリ・アクセス、メッセージ信号割り込みなどのさまざまな動作を要求できる。さらに、プロセッサはアダプタに要求を発行できる。たとえば図2に戻って、プロセッサ102はアダプタ機能138にアクセスするための要求を発行してもよい。この要求は、プロセッサからI/Oハブ112および1つまたはそれ以上のスイッチ114を経由してアダプタ機能に送られる。この実施形態において、メモリ・コントローラは示されていない。しかし、I/Oハブはプロセッサに直接結合されてもよいし、メモリ・コントローラを介して結合されてもよい。

【 0 0 2 9 】

例として、プロセッサ内で実行するオペレーティング・システム140が、特定の動作を要求してアダプタ機能に命令を発行する。この例において、オペレーティング・システムが発行した命令はI/Oインフラストラクチャに特定のなものである。つまり、I/OインフラストラクチャはPCIまたはPCIe (別様に示されない限り、本明細書においてこれらはどちらもPCIと呼ばれる) に基づいているので、この命令はPCI命令である。PCI命令の例をいくつか挙げると、PCIロード (L o a d)、PCI保存 (S t o r e) およびPCI保存ブロック (S t o r e B l o c k) などがある。この例においてはI/Oインフラストラクチャおよび命令がPCIに基づいているが、他の実施形態においては他のインフラストラクチャおよび対応する命令が用いられてもよい。

【 0 0 3 0 】

一特定例において、この命令はアダプタ機能のアドレス空間内の特定の場所に向けられる。たとえば図4に示されるとおり、アダプタ機能138は複数のアドレス空間として定義される記憶装置200を含み、記憶装置200はたとえば構成空間202 (例、PCI機能のためのPCI構成空間) と、I/O空間204 (例、PCI I/O空間) と、1つまたはそれ以上のメモリ空間206 (例、PCIメモリ空間) とを含む。他の実施形態においては、これより多いか、少ないか、またはこれとは異なるアドレス空間が提供されてもよい。命令は、特定のアドレス空間およびアドレス空間内の特定の場所を目標とする。これによって、命令を発行する構成 (例、オペレーティング・システム、L P A R、プロセッサ、ゲストなど) がアダプタ機能にアクセスすることが認可されることが確実になる。

【 0 0 3 1 】

命令の処理を容易にするために、1つまたはそれ以上のデータ構造に保存された情報が用いられる。アダプタに関する情報を含むこうしたデータ構造の1つは、たとえば安全なメモリなどに保存された関数表300である。図5Aに示されるとおり、一例において、関数表300は1つまたはそれ以上の関数表エントリ (f u n c t i o n t a b l e e n t r i e s : F T E s) 302を含む。一例においては、アダプタ機能当り1つの関

10

20

30

40

50

数表エントリがある。各関数表エントリ 3 0 2 は、そのアダプタ機能に関連する処理に用いられるべき情報を含む。一例において、関数表エントリ 3 0 2 はたとえば以下のものなどを含む。

【 0 0 3 2 】

インスタンス番号 3 0 8。このフィールドは、関数表エントリに関連する関数ハンドルの特定のインスタンスを示す。

【 0 0 3 3 】

デバイス表エントリ (Device Table Entry : D T E) インデックス 1 ... n 3 1 0。1 つまたはそれ以上のデバイス表インデックスがあってもよく、各インデックスはデバイス表エントリ (D T E) を探し出すためのデバイス表へのインデックスである。アダプタ機能当り 1 つまたはそれ以上のデバイス表エントリがあり、各エントリはそのアダプタ機能に関連する情報を含み、その情報はアダプタ機能の要求 (例、D M A 要求、M S I 要求) を処理するために用いられる情報、およびアダプタ機能に対する要求 (例、P C I 命令) に関する情報を含む。各デバイス表エントリは、アダプタ機能に割り当てられたシステム・メモリ内の 1 つのアドレス空間に関連付けられている。アダプタ機能は、そのアダプタ機能に割り当てられたシステム・メモリ内の 1 つまたはそれ以上のアドレス空間を有してもよい。

【 0 0 3 4 】

ビジー・インジケータ 3 1 2。このフィールドは、アダプタ機能がビジー (使用中) かどうかを示す。

【 0 0 3 5 】

永久エラー状態インジケータ 3 1 4。このフィールドは、アダプタ機能が永久エラー状態であるかどうかを示す。

【 0 0 3 6 】

回復開始インジケータ 3 1 6。このフィールドは、アダプタ機能に対する回復が開始されたかどうかを示す。

【 0 0 3 7 】

許可インジケータ 3 1 8。このフィールドは、アダプタ機能を有効にしようとしているオペレーティング・システムがそれを行うための認可を有するかどうかを示す。

【 0 0 3 8 】

イネーブル・インジケータ 3 2 0 : このフィールドは、アダプタ機能が有効にされているかどうかを示す (例、1 = 有効、0 = 無効) 。

【 0 0 3 9 】

要求元識別子 (Requestor Identifier : R I D) 3 2 2。これはアダプタ機能の識別子であり、たとえばバス番号、デバイス番号および機能番号などを含んでもよい。このフィールドは、たとえばアダプタ機能の構成空間のアクセスなどのために用いられる。

【 0 0 4 0 】

たとえば、オペレーティング・システム (またはその他の構成) からアダプタ機能に発行される命令において構成空間を指定することによって、構成空間がアクセスされてもよい。命令において指定されるのは、構成空間へのオフセット、および R I D を含む適切な関数表エントリを捜し出すために用いられる関数ハンドルである。ファームウェアは命令を受け取り、それが構成空間に対するものであると判断する。したが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。アダプタ機能の場所は 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。たとえば、オフセットは構成空間におけるレジスタ番号を指定する。

【 0 0 4 1 】

基底アドレス・レジスタ (Base Address Register : B A R) (1 から n) 3 2 4。このフィールドは B A R 0 ~ B A R n と示される複数の無符号の整数

10

20

30

40

50

を含み、これらの整数は最初に指定されたアダプタ機能に関連付けられており、それらの値はアダプタ機能に関連する基底アドレス・レジスタにも保存される。各BAR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、さらにアドレス空間のタイプ、つまりそれがたとえば64ビットもしくは32ビットのメモリ空間なのか、または32ビットのI/O空間なのかも示す。

【0042】

一例において、BARはアダプタ機能のメモリ空間もしくはI/O空間またはその両方へのアクセスのために用いられる。たとえば、アダプタ機能にアクセスするための命令において提供されるオフセットを、命令において指示されるアドレス空間に関連する基底アドレス・レジスタの値に加えることによって、アダプタ機能にアクセスするために用いられるべきアドレスを得る。命令において提供されるアドレス空間識別子は、アクセスすべきアダプタ機能内のアドレス空間および使用されるべき対応のBARを識別する。

10

【0043】

サイズ1...n326。このフィールドはSIZE0~SIZEnと示される複数の無符号の整数を含む。サイズ・フィールドの値は、0以外のときは、前述のBARに対応する各エントリを有する各アドレス空間のサイズを表す。

【0044】

BARおよびサイズに関するさらなる詳細を以下に説明する。

1. BARがアダプタ機能に対して実施されないとき、BARフィールドおよび対応するサイズ・フィールドはどちらも0として保存される。

20

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とき、対応するサイズ・フィールドは0ではなく、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とき、

a. BARnフィールドは最下位アドレス・ビットを表す。

b. 次に連続するBARn+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SIZEnフィールドは0ではなく、アドレス空間のサイズを表す。

d. 対応するSIZEn+1フィールドは意味を有さず、0として保存される。

【0045】

内部ルーティング情報328。この情報は、アダプタへの特定のルーティングを行うために用いられる。この情報は、たとえばノード、プロセッサ・チップ、およびI/Oハブ・アドレス指定情報などを含む。

30

【0046】

状態インジケータ330。これは、ロード/保存動作がブロックされているかどうかの表示、およびその他の表示を提供する。

【0047】

一例において、ビジー・インジケータと、永久エラー状態インジケータと、回復開始インジケータとは、ファームウェアが行うモニタリングに基づいて設定される。さらに、たとえばポリシーに基づいて許可インジケータが設定される。BAR情報は、プロセッサ(例、プロセッサのファームウェア)によるバス・ウォーク(bus walk)の際に発見された構成情報に基づいている。その他のフィールドは構成、初期化もしくはイベントまたはその組み合わせに基づいて設定されてもよい。他の実施形態において、関数表エントリはもっと多くの情報、もっと少ない情報または異なる情報を含んでもよい。含まれる情報は、アダプタ機能に支持される動作またはアダプタ機能のために有効にされる動作に依存し得る。

40

【0048】

1つまたはそれ以上のエントリを含む関数表の中の関数表エントリを捜し出すために、一実施形態においては関数ハンドルが用いられる。たとえば、関数ハンドルの1ビットまたはそれ以上のビットを関数表へのインデックスとして用いることで、特定の関数表エントリを捜し出す。

50

【 0 0 4 9 】

図 5 B を参照して、関数ハンドルに関するさらなる詳細を説明する。一例において、関数ハンドル 3 5 0 は、P C I 関数ハンドルが有効にされたかどうかを示すイネーブル・インジケータ 3 5 2 と、機能を識別する P C I 機能番号 3 5 4（これは静的識別子であり、一実施形態においては関数表へのインデックスである）と、この関数ハンドルの特定のインスタンスを示すインスタンス番号 3 5 6 とを含む。たとえば、機能が有効にされるたびにインスタンス番号が増分されて新しいインスタンス番号を提供する。

【 0 0 5 0 】

本発明の局面に従うと、アダプタ機能にアクセスするために、ある構成がアダプタ機能に対する要求を発行し、これはプロセッサによって実行される。本明細書の例において、この構成はオペレーティング・システムであるが、他の例においてはそれがシステム、プロセッサ、論理区画、ゲストなどであってもよい。これらの要求はアダプタにアクセスする特定の命令を紹介する。命令の例は P C I ロード、P C I 保存、および P C I 保存ブロック命令を含む。これらの命令はアダプタ・アーキテクチャ（例、P C I）に特有のものである。これらの命令に関するさらなる詳細を下に説明する。たとえば、図 6 A ~ 図 8 を参照して P C I ロード命令の一実施形態を説明し、図 9 A ~ 図 1 1 を参照して P C I 保存命令の一実施形態を説明し、図 1 2 A ~ 図 1 4 を参照して P C I 保存ブロック命令の一実施形態を説明する。

【 0 0 5 1 】

最初に図 6 A を参照すると、P C I ロード命令の一実施形態が示される。図示されたとおり、P C I ロード命令 4 0 0 はたとえば、P C I ロード命令を示すオペコード 4 0 2 と、アダプタ機能からフェッチされたデータがロードされる場所を指定する第 1 のフィールド 4 0 4 と、そこからデータをロードすべきアダプタ機能に関するさまざまな情報が含まれる場所を指定する第 2 のフィールド 4 0 6 とを含む。フィールド 1 およびフィールド 2 と示される場所の内容を下にさらに説明する。

【 0 0 5 2 】

一例において、フィールド 1 は一般的なレジスタを示し、図 6 B に示されたとおり、そのレジスタの内容 4 0 4 は、命令において指定されたアダプタ機能の場所からロードされた 1 バイトまたはそれ以上のバイトの連続する範囲を含む。一例において、データはレジスタの右端のバイト位置にロードされる。

【 0 0 5 3 】

一実施形態において、フィールド 2 はさまざまな情報を含む一対の一般的なレジスタを示す。図 6 C に示されたとおり、そのレジスタの内容はたとえば以下のものなどを含む。

【 0 0 5 4 】

有効化ハンドル 4 1 0。このフィールドは、そこからデータをロードすべきアダプタ機能の有効化された関数ハンドルである。

【 0 0 5 5 】

アドレス空間 4 1 2。このフィールドは、そこからデータをロードすべきアダプタ機能内のアドレス空間を識別する。

【 0 0 5 6 】

アドレス空間内のオフセット 4 1 4。このフィールドは、そこからデータをロードすべき指定アドレス空間内のオフセットを指定する。

【 0 0 5 7 】

長さフィールド 4 1 6。このフィールドはロード動作の長さ（例、ロードされるべきバイト数）を指定する。

【 0 0 5 8 】

状態フィールド 4 1 8。このフィールドは、予め規定された条件コードによって命令が完了するときに適用可能な状態コードを提供する。

【 0 0 5 9 】

一実施形態において、アダプタ機能からロードされるバイトは、そのアダプタ機能の指

10

20

30

40

50

示 P C I アドレス空間の整数境界内に含まれるべきである。アドレス空間フィールドがメモリ・アドレス空間を指示するとき、整数境界サイズはたとえばダブルワードである。アドレス空間フィールドが I / O アドレス空間または構成アドレス空間を指示するとき、整数境界サイズはたとえばワードである。

【 0 0 6 0 】

図 7 ~ 8 を参照して、P C I ロード命令に関連する論理の一実施形態を説明する。一例において、この命令はオペレーティング・システム（またはその他の構成）によって発行され、そのオペレーティング・システムを実行するプロセッサ（例、ファームウェア）によって実行される。本明細書の例においては、命令およびアダプタ機能が P C I に基づいている。しかし、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が用いられてもよい。

10

【 0 0 6 1 】

命令を発行するために、オペレーティング・システムは命令（例、命令の指示する 1 つまたはそれ以上のレジスタの中）に次のオペランドを与える。すなわち、P C I 関数ハンドル、P C I アドレス空間（P C I a d d r e s s s p a c e : P C I A S）、P C I アドレス空間へのオフセット、およびロードされるべきデータの長さである。P C I ロード命令がうまく完了するとき、データはその命令が指示した場所（例、レジスタ）にロードされる。

【 0 0 6 2 】

図 7 を参照すると、最初に P C I ロード命令を可能にする機構がインストールされているかどうかの判断が行われる（問い合わせ（I N Q U I R Y）5 0 0）。この判断は、たとえば制御ブロックなどに保存されたインジケータをチェックすることなどによって行われる。もしこの機構がインストールされていなければ、例外条件が提供される（ステップ（S T E P）5 0 2）。そうでなければ、オペランドが整列されているかどうかの判断が行われる（問い合わせ 5 0 4）。たとえば、もし特定のオペランドが偶数 / 奇数レジスタ対にある必要があれば、それらの要求が満たされているかどうかの判断が行われる。もしオペランドが整列されていなければ、例外条件が提供される（ステップ 5 0 6）。そうではなくて、機構がインストールされておりかつオペランドが整列されていれば、P C I ロード命令のオペランドに与えられているハンドルが有効にされているかどうかの判断が行われる（問い合わせ 5 0 8）。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行われる。もしハンドルが有効にされていなければ、例外条件が提供される（ステップ 5 1 0）。

20

30

【 0 0 6 3 】

もしハンドルが有効にされていれば、そのハンドルを用いて関数表エントリを捜し出す（ステップ 5 1 2）。すなわち、ハンドルの少なくとも一部を関数表へのインデックスとして用いて、そこからデータをロードすべきアダプタ機能に対応する関数表エントリを捜し出す。

【 0 0 6 4 】

その後、もしこの命令を発行している構成がゲストであれば、その機能がゲストによる使用のために構成されているかどうかの判断が行われる（問い合わせ 5 1 4）。もしそれが認可されていなければ、例外条件が提供される（ステップ 5 1 6）。もしこの構成がゲストでなければこの問い合わせは無視されてもよいし、指示されれば他の認可がチェックされてもよい。（一例において、z / A r c h i t e c t u r e（I B M 社の登録商標）においては、ページ可能ゲストは解釈のレベル 2 にて解釈実行開始（S t a r t I n t e r p r e t i v e E x e c u t i o n : S I E）命令を介して解釈的に実行される。たとえば、論理区画（l o g i c a l p a r t i t i o n : L P A R）ハイパーバイザは S I E 命令を実行して、物理的な固定メモリにおいて論理区画を開始する。もし z / V M（I B M 社の登録商標）がその論理区画におけるオペレーティング・システムであれば、それが S I E 命令を発行してそのゲスト（仮想）マシンを自身の V = V（仮想）記憶装置内で実行する。したがって L P A R ハイパーバイザはレベル 1 S I E を使用し、z / V

40

50

M (I B M 社の登録商標) ハイパーバイザはレベル 2 S I E を使用する。)

【 0 0 6 5 】

次いで、その機能が有効にされているかどうかの判断が行われる (問い合わせ 5 1 8) 。一例において、この判断は関数表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って行われる。もし機能が有効にされていなければ、例外条件が提供される (ステップ 5 2 0) 。

【 0 0 6 6 】

もし機能が有効にされていれば、次いでアドレス空間が有効であるかどうかの判断が行われる (問い合わせ 5 2 2) 。たとえば、指定されたアドレス空間がアダプタ機能の指示アドレス空間であり、かつこの命令に対して適切なアドレス空間であるかどうかである。もしアドレス空間が無効であれば、例外条件が提供される (ステップ 5 2 4) 。そうでなければ、ロード / 保存がブロックされているかどうかの判断が行われる (問い合わせ 5 2 6) 。一例において、この判断は関数表エントリ内の状態インジケータをチェックすることによって行われる。もしロード / 保存がブロックされていれば、例外条件が提供される (ステップ 5 2 8) 。

【 0 0 6 7 】

しかし、もしロード / 保存がブロックされていなければ、回復が活動状態であるかどうかの判断が行われる (問い合わせ 5 3 0) 。一例において、この判断は関数表エントリ内の回復開始インジケータをチェックすることによって行われる。もし回復が活動状態であれば、例外条件が提供される (ステップ 5 3 2) 。そうでなければ、機能がビジーかどうかの判断が行われる (問い合わせ 5 3 4) 。この判断は、関数表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って行われる。もし機能がビジーであれば、ビジー条件が提供される (ステップ 5 3 6) 。ビジー条件によって、命令が中止される代わりに再試行されてもよい。

【 0 0 6 8 】

もし機能がビジーでなければ、命令において指定されたオフセットが有効かどうかのさらなる判断が行われる (問い合わせ 5 3 8) 。すなわち、オフセットが関数表エントリにおいて指定されているとおり基底内の動作の長さおよびアドレス空間の長さとは組み合わせられているかどうかである。もし有効でなければ、例外条件が提供される (ステップ 5 4 0) 。しかしもしオフセットが有効であれば、長さが有効かどうかの判断が行われる (問い合わせ 5 4 2) 。すなわち、アドレス空間タイプによって、アドレス空間内のオフセット、および整数境界サイズが有効な長さである。もし有効でなければ、例外条件が提供される (ステップ 5 4 4) 。そうでなければ、ロード命令による処理が続けられる。 (一実施形態においては、ファームウェアが上記のチェックを行う。)

【 0 0 6 9 】

図 8 とともに続けると、ロードがアダプタ機能の構成アドレス空間に対するものかどうかの判断がファームウェアによって行われる (問い合わせ 5 5 0) 。すなわち、アダプタ機能のメモリの構成に基づいて、命令に与えられた指定アドレス空間が構成空間であるかどうかである。もしそうであれば、ファームウェアはさまざまな処理を行って、その要求をアダプタ機能に結合されたハブに提供する。次いでハブはその要求を機能に送る (ステップ 5 5 2) 。

【 0 0 7 0 】

たとえばファームウェアは、命令オペランドに与えられた関数ハンドルによって指された関数表エントリから要求元 I D を得る。さらに、ファームウェアは関数表エントリ内の情報 (例、内部ルーティング情報) に基づいて、この要求を受け取るべきハブを定める。すなわち、ある環境は 1 つまたはそれ以上のハブを有してもよく、ファームウェアはアダプタ機能に結合されたハブを定める。次いでファームウェアは要求をそのハブに転送する。ハブは構成読取り要求パケットを生成し、それは関数表エントリ内の R I D によって識別されるアダプタ機能に向けて P C I バス上に流出する。以下に説明されるとおり、構成読取り要求はデータをフェッチするために用いられる R I D およびオフセット (すなわち

10

20

30

40

50

データ・アドレス)を含む。

【0071】

問い合わせ550に戻って、もし指示されたアドレス空間が構成空間でなければ、ファームウェアは再びさまざまな処理を行ってその要求をハブに提供する(ステップ554)。ファームウェアはハンドルを用いて関数表エントリを選択し、そのエントリから適切なハブを捜し出すための情報を得る。加えてファームウェアはロード動作に用いられるデータ・アドレスを算出する。このアドレスは、関数表エントリから得られたBAR開始アドレス(このBARは命令において提供されるアドレス空間識別子に関連するものである)を命令において提供されるオフセットに加えることによって算出される。この算出データ・アドレスがハブに提供される。次いでハブはそのアドレスを取って、たとえばDMA読取り要求パケットなどの要求パケットに含ませ、それはアダプタ機能に向けてPCIバス上に流出する。

10

【0072】

ステップ552またはステップ554のいずれかを介して要求を受け取ったことに応答して、アダプタ機能は要求されるデータを指定場所(すなわちデータ・アドレスの場所)からフェッチし、要求への応答においてそのデータを戻す(ステップ556)。この応答はアダプタ機能からI/Oハブに転送される。応答を受け取ったことに応答して、ハブはその応答を開始プロセッサに転送する。次いで開始プロセッサは応答パケットからそのデータを取り、命令の指定する指示場所(例、フィールド1 404)にロードする。PCIロード動作は成功を示して終了する(例、条件コード0を設定する)。

20

【0073】

アダプタ機能からデータを検索してそれを指示された場所に保存するロード命令に加えて実行され得る別の命令は、保存命令である。保存命令は、アダプタ機能内の指定された場所にデータを保存する。図9Aを参照して、PCI保存命令の一実施形態を説明する。図示されるとおり、PCI保存命令600は、たとえばPCI保存命令を示すオペコード602と、アダプタ機能内の保存されるべきデータを含む場所を指定する第1のフィールド604と、データが保存されるべきアダプタ機能に関するさまざまな情報が含まれる場所を指定する第2のフィールド606とを含む。フィールド1および2によって示される場所の内容を、以下にさらに説明する。

【0074】

30

一例において、フィールド1は一般的なレジスタを示し、図9Bに示されるとおり、そのレジスタの内容604は、アダプタ機能の指定場所に保存されるべきデータの1バイトまたはそれ以上のバイトの連続する範囲を含む。一例においては、レジスタの右端のバイト位置のデータが保存される。

【0075】

一実施形態において、フィールド2はさまざまな情報を含む一对の一般的なレジスタを示す。図9Cに示されるとおり、そのレジスタの内容はたとえば以下のものなどを含む。

【0076】

有効化ハンドル610。このフィールドは、データを保存すべきアダプタ機能の有効化された関数ハンドルである。

40

【0077】

アドレス空間612。このフィールドは、データを保存すべきアダプタ機能内のアドレス空間を識別する。

【0078】

アドレス空間内のオフセット614。このフィールドは、データを保存すべき指定アドレス空間内のオフセットを指定する。

【0079】

長さフィールド616。このフィールドは保存動作の長さ(例、保存されるべきバイト数)を指定する。

【0080】

50

状態フィールド 6 1 8。このフィールドは、予め規定された条件コードによって命令が完了するときに適用可能な状態コードを提供する。

【 0 0 8 1 】

図 1 0 ~ 1 1 を参照して、P C I 保存命令に関連する論理の一実施形態を説明する。一例において、この命令はオペレーティング・システムによって発行され、そのオペレーティング・システムを実行するプロセッサ（例、ファームウェア）によって実行される。

【 0 0 8 2 】

命令を発行するために、オペレーティング・システムは命令（例、命令の指示する 1 つまたはそれ以上のレジスタの中）に次のオペランドを与える。すなわち、P C I 関数ハンドル、P C I アドレス空間（P C I A S）、P C I アドレス空間へのオフセット、保存されるべきデータの長さ、および保存されるべきデータに対するポインタである。P C I 保存命令がうまく完了するとき、データはその命令が指示した場所に保存される。

【 0 0 8 3 】

図 1 0 を参照すると、最初に P C I 保存命令を可能にする機構がインストールされているかどうかの判断が行われる（問い合わせ 7 0 0）。この判断は、たとえば制御ブロックなどに保存されたインジケータをチェックすることなどによって行われる。もしこの機構がインストールされていなければ、例外条件が提供される（ステップ 7 0 2）。そうでなければ、オペランドが整列されているかどうかの判断が行われる（問い合わせ 7 0 4）。たとえば、もし特定のオペランドが偶数 / 奇数レジスタ対にある必要があれば、それらの要求が満たされているかどうかの判断が行われる。もしオペランドが整列されていなければ、例外条件が提供される（ステップ 7 0 6）。そうではなくて、機構がインストールされておりかつオペランドが整列されていれば、P C I 保存命令のオペランドに与えられているハンドルが有効にされているかどうかの判断が行われる（問い合わせ 7 0 8）。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行われる。もしハンドルが有効にされていなければ、例外条件が提供される（ステップ 7 1 0）。

【 0 0 8 4 】

もしハンドルが有効にされていれば、そのハンドルを用いて関数表エントリを捜し出す（ステップ 7 1 2）。すなわち、ハンドルの少なくとも一部を関数表へのインデックスとして用いて、データを保存すべきアダプタ機能に対応する関数表エントリを捜し出す。

【 0 0 8 5 】

その後、もしこの命令を発行している構成がゲストであれば、その機能がゲストによる使用のために構成されているかどうかの判断が行われる（問い合わせ 7 1 4）。もしそれが認可されていなければ、例外条件が提供される（ステップ 7 1 6）。もしこの構成がゲストでなければこの問い合わせは無視されてもよいし、指示されれば他の認可がチェックされてもよい。

【 0 0 8 6 】

次いで、その機能が有効にされているかどうかの判断が行われる（問い合わせ 7 1 8）。一例において、この判断は関数表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って行われる。もし機能が有効にされていなければ、例外条件が提供される（ステップ 7 2 0）。

【 0 0 8 7 】

もし機能が有効にされていれば、次いでアドレス空間が有効であるかどうかの判断が行われる（問い合わせ 7 2 2）。たとえば、指定されたアドレス空間がアダプタ機能の指示アドレス空間であり、かつこの命令に対して適切なアドレス空間であるかどうかである。もしアドレス空間が無効であれば、例外条件が提供される（ステップ 7 2 4）。そうでなければ、ロード / 保存がブロックされているかどうかの判断が行われる（問い合わせ 7 2 6）。一例において、この判断は関数表エントリ内の状態インジケータをチェックすることによって行われる。もしロード / 保存がブロックされていれば、例外条件が提供される（ステップ 7 2 8）。

【 0 0 8 8 】

しかし、もしロード / 保存がブロックされていなければ、回復が活動状態であるかどうかの判断が行われる（問い合わせ 7 3 0）。一例において、この判断は関数表エントリ内の回復開始インジケータをチェックすることによって行われる。もし回復が活動状態であれば、例外条件が提供される（ステップ 7 3 2）。そうでなければ、機能がビジーかどうかの判断が行われる（問い合わせ 7 3 4）。この判断は、関数表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って行われる。もし機能がビジーであれば、ビジー条件が提供される（ステップ 7 3 6）。ビジー条件によって、命令が中止される代わりに再試行されてもよい。

【 0 0 8 9 】

もし機能がビジーでなければ、命令において指定されたオフセットが有効かどうかのさらなる判断が行われる（問い合わせ 7 3 8）。すなわち、オフセットが関数表エントリにおいて指定されているとおり基底内の動作の長さおよびアドレス空間の長さとは組み合わされているかどうかである。もし有効でなければ、例外条件が提供される（ステップ 7 4 0）。しかしもしオフセットが有効であれば、長さが有効かどうかの判断が行われる（問い合わせ 7 4 2）。すなわち、アドレス空間タイプによって、アドレス空間内のオフセット、および整数境界サイズが有効な長さである。もし有効でなければ、例外条件が提供される（ステップ 7 4 4）。そうでなければ、保存命令による処理が続けられる。（一実施形態においては、ファームウェアが上記のチェックを行う。）

【 0 0 9 0 】

図 1 1 とともに続けると、保存がアダプタ機能の構成アドレス空間に対するものかどうかの判断がファームウェアによって行われる（問い合わせ 7 5 0）。すなわち、アダプタ機能のメモリの構成に基づいて、命令に与えられた指定アドレス空間が構成空間であるかどうかである。もしそうであれば、ファームウェアはさまざまな処理を行って、その要求をアダプタ機能に結合されたハブに提供する。次いでハブはその要求を機能に送る（ステップ 7 5 2）。

【 0 0 9 1 】

たとえばファームウェアは、命令オペランドに与えられた関数ハンドルによって指された関数表エントリから要求元 ID を得る。さらに、ファームウェアは関数表エントリ内の情報（例、内部ルーティング情報）に基づいて、この要求を受け取るべきハブを定める。すなわち、ある環境は 1 つまたはそれ以上のハブを有してもよく、ファームウェアはアダプタ機能に結合されたハブを定める。次いでファームウェアは要求をそのハブに転送する。ハブは構成書込み要求パケットを生成し、それは関数表エントリ内の R I D によって識別されるアダプタ機能に向けて P C I バス上に流出する。以下に説明されるとおり、構成書込み要求はデータを保存するために用いられる R I D およびオフセット（すなわちデータ・アドレス）を含む。

【 0 0 9 2 】

問い合わせ 7 5 0 に戻って、もし指示されたアドレス空間が構成空間でなければ、ファームウェアは再びさまざまな処理を行ってその要求をハブに提供する（ステップ 7 5 4）。ファームウェアはハンドルを用いて関数表エントリを選択し、そのエントリから適切なハブを捜し出すための情報を得る。加えてファームウェアは保存動作に用いられるデータ・アドレスを算出する。このアドレスは、関数表エントリから得られた B A R 開始アドレスを命令において提供されるオフセットに加えることによって算出される。この算出データ・アドレスがハブに提供される。次いでハブはそのアドレスを取って、たとえば D M A 書込み要求パケットなどの要求パケットに含ませ、それはアダプタ機能に向けて P C I バス上に流出する。

【 0 0 9 3 】

ステップ 7 5 2 またはステップ 7 5 4 のいずれかを介して要求を受け取ったことに応答して、アダプタ機能は要求されるデータを指定場所（すなわちデータ・アドレスの場所）に保存する（ステップ 7 5 6）。P C I 保存動作は成功を示して終了する（例、条件コー

10

20

30

40

50

ド 0 を設定する)。

【 0 0 9 4 】

典型的にたとえば最大 8 バイトをロードまたは保存するロードおよび保存命令に加えて、実行され得る別の命令は保存ブロック命令である。保存ブロック命令は、アダプタ機能内の指定場所にデータのより大きなブロック (例、16、32、64、128 または 256 バイト) を保存する。ブロック・サイズは必ずしも 2 の累乗のサイズに制限されなくてもよい。一例において、指定場所はアダプタ機能のメモリ空間内にある (I/O または構成空間ではない)。

【 0 0 9 5 】

図 12A を参照して、PCI 保存ブロック命令の一実施形態を説明する。図示されたとおり、PCI 保存ブロック命令 800 は、たとえば PCI 保存ブロック命令を示すオペコード 802 と、データが保存されるべきアダプタ機能に関するさまざまな情報が含まれる場所を指定する第 1 のフィールド 804 と、データが保存されるべき指定アドレス空間内のオフセットを含む場所を指定する第 2 のフィールド 806 と、アダプタ機能に保存されるべきデータのシステム・メモリ内のアドレスを含む場所を指定する第 3 のフィールド 808 とを含む。フィールド 1、2 および 3 によって示される場所の内容を、以下にさらに説明する。

10

【 0 0 9 6 】

一実施形態において、フィールド 1 はさまざまな情報を含む一般的なレジスタを示す。図 12B に示されたとおり、そのレジスタの内容はたとえば以下のものなどを含む。

20

【 0 0 9 7 】

有効化ハンドル 810。このフィールドは、データを保存すべきアダプタ機能の有効化された関数ハンドルである。

【 0 0 9 8 】

アドレス空間 812。このフィールドは、データを保存すべきアダプタ機能内のアドレス空間を識別する。

【 0 0 9 9 】

長さフィールド 814。このフィールドは保存動作の長さ (例、保存されるべきバイト数) を指定する。

【 0 1 0 0 】

30

状態フィールド 816。このフィールドは、予め規定された条件コードによって命令が完了するときに適用可能な状態コードを提供する。

【 0 1 0 1 】

一例において、フィールド 2 は一般的なレジスタを示し、図 12C に示されたとおり、そのレジスタの内容はデータを保存すべき指定アドレス空間内のオフセットを指定する値 (例、64 ビットの無符号整数) を含む。

【 0 1 0 2 】

一例において、フィールド 3 は図 12D に示されたとおり、アダプタ機能内に保存されるべきデータの第 1 バイトのシステム・メモリ内の論理アドレス 822 を含む。

【 0 1 0 3 】

40

図 13 ~ 14 を参照して、PCI 保存ブロック命令に関連する論理の一実施形態を説明する。一例において、この命令はオペレーティング・システムによって発行され、そのオペレーティング・システムを実行するプロセッサ (例、ファームウェア) によって実行される。

【 0 1 0 4 】

命令を発行するために、オペレーティング・システムは命令 (例、命令の指示する 1 つまたはそれ以上のレジスタの中) に次のオペランドを与える。すなわち、PCI 関数ハンドル、PCI アドレス空間 (PCI AS)、PCI アドレス空間へのオフセット、保存されるべきデータの長さ、および保存されるべきデータに対するポインタである。ポインタ・オペランドはレジスタと、符号付きまたは無符号の変位との両方を含んでもよい。PC

50

I 保存ブロック命令がうまく完了するとき、データはその命令が指示したアダプタ内の場所に保存される。

【 0 1 0 5 】

図 1 3 を参照すると、最初に P C I 保存ブロック命令を可能にする機構がインストールされているかどうかの判断が行われる（問い合わせ 9 0 0）。この判断は、たとえば制御ブロックなどに保存されたインジケータをチェックすることなどによって行われる。もしこの機構がインストールされていなければ、例外条件が提供される（ステップ 9 0 2）。そうではなくて機構がインストールされていれば、P C I 保存ブロック命令のオペランドに与えられているハンドルが有効にされているかどうかの判断が行われる（問い合わせ 9 0 4）。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行われる。もしハンドルが有効にされていなければ、例外条件が提供される（ステップ 9 0 6）。

10

【 0 1 0 6 】

もしハンドルが有効にされていれば、そのハンドルを用いて関数表エントリを捜し出す（ステップ 9 1 2）。すなわち、ハンドルの少なくとも一部を関数表へのインデックスとして用いて、データを保存すべきアダプタ機能に対応する関数表エントリを捜し出す。

【 0 1 0 7 】

その後、もしこの命令を発行している構成がゲストであれば、その機能がゲストによる使用のために構成されているかどうかの判断が行われる（問い合わせ 9 1 4）。もしそれが認可されていなければ、例外条件が提供される（ステップ 9 1 6）。もしこの構成がゲストでなければこの問い合わせは無視されてもよいし、指示されれば他の認可がチェックされてもよい。

20

【 0 1 0 8 】

次いで、その機能が有効にされているかどうかの判断が行われる（問い合わせ 9 1 8）。一例において、この判断は関数表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って行われる。もし機能が有効にされていなければ、例外条件が提供される（ステップ 9 2 0）。

【 0 1 0 9 】

もし機能が有効にされていれば、次いでアドレス空間が有効であるかどうかの判断が行われる（問い合わせ 9 2 2）。たとえば、指定されたアドレス空間がアダプタ機能の指示アドレス空間であり、かつこの命令に対して適切なアドレス空間（すなわちメモリ空間）であるかどうかである。もしアドレス空間が無効であれば、例外条件が提供される（ステップ 9 2 4）。そうでなければ、ロード / 保存がブロックされているかどうかの判断が行われる（問い合わせ 9 2 6）。一例において、この判断は関数表エントリ内の状態インジケータをチェックすることによって行われる。もしロード / 保存がブロックされていれば、例外条件が提供される（ステップ 9 2 8）。

30

【 0 1 1 0 】

しかし、もしロード / 保存がブロックされていなければ、回復が活動状態であるかどうかの判断が行われる（問い合わせ 9 3 0）。一例において、この判断は関数表エントリ内の回復開始インジケータをチェックすることによって行われる。もし回復が活動状態であれば、例外条件が提供される（ステップ 9 3 2）。そうでなければ、機能がビジーかどうかの判断が行われる（問い合わせ 9 3 4）。この判断は、関数表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って行われる。もし機能がビジーであれば、ビジー条件が提供される（ステップ 9 3 6）。ビジー条件によって、命令が中止される代わりに再試行されてもよい。

40

【 0 1 1 1 】

もし機能がビジーでなければ、命令において指定されたオフセットが有効かどうかのさらなる判断が行われる（問い合わせ 9 3 8）。すなわち、オフセットが関数表エントリにおいて指定されているとおりに基底内の動作の長さおよびアドレス空間の長さとは組み合わせられているかどうかである。もし有効でなければ、例外条件が提供される（ステップ 9 4

50

0)。しかしもしオフセットが有効であれば、長さが有効かどうかの判断が行われる（問い合わせ942）。すなわち、アドレス空間タイプによって、アドレス空間内のオフセット、および整数境界サイズが有効な長さである。もし有効でなければ、例外条件が提供される（ステップ944）。そうでなければ、保存ブロック命令による処理が続けられる。（一実施形態においては、ファームウェアが上記のチェックを行う。）

【0112】

図14とともに続けると、保存されるべきデータを含む記憶装置がアクセス可能かどうかの判断がファームウェアによって行われる（問い合わせ950）。もしアクセス可能でなければ、例外条件が提供される（ステップ952）。もしそうであれば、ファームウェアはさまざまな処理を行って、その要求をアダプタ機能に結合されたハブに提供する。次いでハブはその要求を機能に送る（ステップ954）。 10

【0113】

たとえば、ファームウェアはハンドルを用いて関数表エントリを選択し、そのエントリから適切なハブを捜し出すための情報を得る。加えてファームウェアは保存ブロック動作に用いられるデータ・アドレスを算出する。このアドレスは、関数表エントリから得られたBAR開始アドレス（BARはアドレス空間識別子によって識別される）を命令において提供されるオフセットに加えることによって算出される。この算出データ・アドレスがハブに提供される。加えて、命令に与えられたアドレスによって参照されるデータが、システム・メモリからフェッチされてI/Oハブに提供される。次いでハブはそのアドレスおよびデータを取って、たとえばDMA書込み要求パケットなどの要求パケットに含ませ、それはアダプタ機能に向けてPCIバス上に流出する。 20

【0114】

要求を受け取ったことに応答して、アダプタ機能は要求されるデータを指定場所（すなわちデータ・アドレスの場所）に保存する（ステップ956）。PCI保存ブロック動作は成功を示して終了する（例、条件コード0を設定する）。

【0115】

上に詳細に説明したのは、アダプタとの通信のために特定の設計された制御命令を用いてコンピューティング環境のアダプタと通信する能力である。この通信はメモリ・マップされたI/Oを用いずに行われ、アダプタ機能内のレジスタを制御することに限定されない。この命令は、命令を発行する構成がアダプタ機能にアクセスすることを認可されることを確実にする。さらに保存ブロック命令については、この命令は指定された主記憶装置の場所が構成のメモリ内にあることを確実にする。 30

【0116】

本明細書に記載される実施形態において、アダプタはPCIアダプタである。本明細書において用いられるPCIとは、周辺構成要素相互接続分科会（Peripheral Component Interconnect Special Interest Group：PCI-SIG）によって定義されたPCIに基づく仕様に従って実施されるあらゆるアダプタを示し、それはPCIまたはPCIeを含むがそれに限定されない。1つの特定例において、周辺構成要素相互接続エクスプレス（PCIe）は、I/Oアダプタとホスト・システムとの間のトランザクションのための双方向性通信プロトコルを定める構成要素レベルの相互接続規格である。PCIe通信は、PCIeバス上の送信のためのPCIe規格に従ってパケットにカプセル化されている。I/Oアダプタで生じてホスト・システムで終わるトランザクションは、上りのトランザクションと呼ばれる。ホスト・システムで生じてI/Oアダプタで終了するトランザクションは、下りのトランザクションと呼ばれる。PCIeトポロジは、2地点間の一方向性リンクが対にされて（例、1本の上りリンク、1本の下りリンク）PCIeバスを形成したものに基いている。背景技術のセクションで上述したとおり、PCIe規格はPCI-SIGによって維持され公表されている。 40

【0117】

当業者に認識されるとおり、本発明の局面はシステム、方法またはコンピュータ・プロ 50

グラム製品として実施されてもよい。したがって本発明の局面は、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、マイクロコードなどを含む）、またはソフトウェアおよびハードウェアの局面を組み合わせた実施形態の形を取ってもよく、本明細書においてこれらはすべて一般的に「回路」、「モジュール」または「システム」と呼ばれることがある。さらに、本発明の局面は、媒体において実施されるコンピュータ読取り可能プログラム・コードを有する１つまたはそれ以上のコンピュータ読取り可能媒体（単数または複数）において実施されるコンピュータ・プログラム製品の形を取ってもよい。

【0118】

１つまたはそれ以上のコンピュータ読取り可能媒体（単数または複数）のあらゆる組み合わせが使用されてもよい。コンピュータ読取り可能媒体は、コンピュータ読取り可能記憶媒体であってもよい。コンピュータ読取り可能記憶媒体は、たとえば電子、磁気、光学、電磁気、または半導体のシステム、装置、もしくはデバイス、または前述のもののあらゆる好適な組み合わせなどであってもよいがそれに限定されない。コンピュータ読取り可能記憶媒体のより特定の例（非網羅的なリスト）は以下を含む。すなわち、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（random access memory：RAM）、リード・オンリ・メモリ（read-only memory：ROM）、消去可能プログラマブル・リード・オンリ・メモリ（erasable programmable read-only memory）（EPROMまたはフラッシュ・メモリ）、ポータブル・コンパクト・ディスク・リード・オンリ・メモリ（compact disc read-only memory：CD-ROM）、光学記憶装置、磁気記憶装置、または前述のもののあらゆる好適な組み合わせである。この文書の状況において、コンピュータ読取り可能記憶媒体は、命令実行システム、装置またはデバイスによって、またはそれに関連して使用されるためのプログラムを含有または保存できるあらゆる有形の媒体であってもよい。

【0119】

次に図１５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１０００はたとえば、本発明の１つまたはそれ以上の局面を提供および促進するためにコンピュータ読取り可能プログラム・コード手段または論理１００４を保存するための１つまたはそれ以上のコンピュータ読取り可能記憶媒体１００２を含む。

【0120】

コンピュータ読取り可能媒体において実施されるプログラム・コードは、ワイヤレス、ワイヤライン、光ファイバ・ケーブル、RFなど、または前述のもののあらゆる好適な組み合わせを含むがそれに限定されない適切な媒体を用いて伝送されてもよい。

【0121】

本発明の局面に対する動作を行うためのコンピュータ・プログラム・コードは、オブジェクト指向プログラミング言語、たとえばJava（商標）、Smalltalk（商標）、C++など、および従来の手続き型プログラミング言語、たとえば「C」プログラミング言語、アセンブラまたは類似のプログラミング言語などを含む、１つまたはそれ以上のプログラミング言語のあらゆる組み合わせで書かれていてもよい。プログラム・コードは、すべてがユーザのコンピュータで実行されてもよいし、スタンド・アロン・ソフトウェア・パッケージとして部分的にユーザのコンピュータで実行されてもよいし、一部がユーザのコンピュータで、一部がリモート・コンピュータで実行されてもよいし、すべてがリモート・コンピュータまたはサーバで実行されてもよい。後者のシナリオにおいて、リモート・コンピュータは、ローカル・エリア・ネットワーク（local area network：LAN）または広域ネットワーク（wide area network：WAN）を含むあらゆる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てもよいし、（たとえば、インターネット・サービス・プロバイダを用いてインターネットを通じて）外部コンピュータへの接続が行われてもよい。

【0122】

本明細書においては、本発明の実施形態に従う方法、装置（システム）およびコンピュータ・プログラム製品の流れ図もしくはブロック図またはその両方を参照して、本発明の局面を説明した。流れ図もしくはブロック図またはその両方の各ブロック、および流れ図もしくはブロック図またはその両方における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現され得ることが理解されるだろう。これらのコンピュータ・プログラム命令が、汎用コンピュータもしくは特定目的のコンピュータのプロセッサ、またはマシンを生成するためのその他のプログラマブル・データ処理装置に与えられることによって、そのコンピュータのプロセッサまたはその他のプログラマブル・データ処理装置を介して実行された命令が、流れ図もしくはブロック図またはその両方の単数または複数のブロックにおいて指定された機能／動作を実現するための手段を生成するようにしてもよい。

10

【0123】

これらのコンピュータ・プログラム命令は、コンピュータ、その他のプログラマブル・データ処理装置またはその他のデバイスに特定の態様で機能するよう指示できるコンピュータ読取り可能媒体の中にも保存されることによって、コンピュータ読取り可能媒体中に保存された命令が、流れ図もしくはブロック図またはその両方の単数または複数のブロックにおいて指定された機能／動作を実現する命令を含む製造品を生成するようにしてもよい。

【0124】

さらに、コンピュータ・プログラム命令は、コンピュータ、その他のプログラマブル・データ処理装置またはその他のデバイスにロードされることによって、そのコンピュータ、その他のプログラマブル装置またはその他のデバイスにおいて一連の動作ステップを行わせることにより、コンピュータまたはその他のプログラマブル装置において実行される命令が、流れ図もしくはブロック図またはその両方の単数または複数のブロックにおいて指定された機能／動作を実現するためのプロセスを提供するような、コンピュータに実現されるプロセスを生成してもよい。

20

【0125】

図面中の流れ図およびブロック図は、本発明のさまざまな実施形態に従うシステム、方法およびコンピュータ・プログラム製品の実現可能なアーキテクチャ、機能および動作を例示するものである。これに関して、流れ図またはブロック図における各ブロックは、指定された論理関数（単数または複数）を実現するための1つまたはそれ以上の実行可能な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していてもよい。さらに、いくつかの代替的な実現においては、ブロック中に示される機能が図面に示される以外の順序で起こることがあることに留意すべきである。たとえば、連続して示される2つのブロックは、伴われる機能に依存して、実際には実質的に同時に実行されてもよいし、それらのブロックがときには逆の順序で実行されてもよい。さらに、ブロック図もしくは流れ図またはその両方の各ブロック、およびブロック図もしくは流れ図またはその両方のブロックの組み合わせは、指定された機能もしくは動作を行う特定目的のハードウェアに基づくシステム、または特定目的の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現され得ることが注目される。

30

40

【0126】

上記に加えて、顧客環境の管理を提供するサービス・プロバイダによって、本発明の1つまたはそれ以上の局面の提供、提案、配置、管理、サービスなどが行われてもよい。たとえばサービス・プロバイダは、1人またはそれ以上の顧客に対して本発明の1つまたはそれ以上の局面を行うコンピュータ・コードもしくはコンピュータ・インフラストラクチャまたはその両方の作成、維持、支持などを行い得る。それに対して、サービス・プロバイダはたとえば契約もしくは料金の合意またはその両方の下で、顧客から報酬を受け取ってもよい。付加的または代替的に、サービス・プロバイダは1人またはそれ以上の第三者に対する広告内容の販売から報酬を受け取ってもよい。

【0127】

50

本発明の一局面においては、本発明の1つまたはそれ以上の局面を行うためにアプリケーションが配置されてもよい。一例として、アプリケーションを配置することは、本発明の1つまたはそれ以上の局面を行うため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。

【0128】

本発明のさらなる局面として、コンピューティング・インフラストラクチャが配置されてもよく、それはコンピューティング・システムにコンピュータ読取り可能コードを統合することを含み、ここでコンピューティング・システムと組み合わされたコードは本発明の1つまたはそれ以上の局面を行うことができる。

【0129】

本発明のさらなる局面として、コンピュータ・システムにコンピュータ読取り可能コードを統合することを含む、コンピューティング・インフラストラクチャを統合するためのプロセスが提供されてもよい。コンピュータ・システムはコンピュータ読取り可能媒体を含み、コンピュータ媒体は本発明の1つまたはそれ以上の局面を含む。コンピュータ・システムと組み合わされたコードは本発明の1つまたはそれ以上の局面を行うことができる。

【0130】

上にさまざまな実施形態を説明したが、それらは単なる例である。たとえば、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つまたはそれ以上の局面を組み込んで使用してもよい。例として、System z (IBM社の登録商標) サーバ以外のサーバ、たとえばPower Systems (IBM社の商標) サーバもしくはインターナショナル・ビジネス・マシーンズ社より提供されるその他のサーバ、または他社のサーバなどが、本発明の1つまたはそれ以上の局面を含んだり、使用したり、そこから利益を得たりしてもよい。さらに、本明細書の例においてはアダプタおよびPCIハブはサーバの部分とみなされているが、他の実施形態においてそれらは必ずしもサーバの部分とみなされなくてもよく、簡単にシステム・メモリもしくはコンピューティング環境のその他の構成要素またはその両方に結合されているとみなされてもよい。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、表が記載されているが、あらゆるデータ構造が使用されてもよく、表という用語はこうしたデータ構造すべてを含むものである。さらに、アダプタはPCIに基づくものであるが、他のアダプタまたは他のI/O構成要素とともに本発明の1つまたはそれ以上の局面を使用できる。アダプタおよびPCIアダプタは単なる例である。さらに、FTEまたはFTEのパラメータは、たとえばハードウェア(例、PCI機能ハードウェア)などを含む、安全なメモリ以外の場所に位置して維持されてもよい。DTE、FTEもしくはハンドルまたはその組み合わせは、もっと多くの情報、もっと少ない情報または異なる情報、および命令または命令フィールドのいずれかを含んでもよい。その他多くの変形が可能である。

【0131】

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つまたはそれ以上の局面から利益を得てもよい。例として、システム・バスを通じてメモリ構成要素に直接的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・コードの保存もしくは実行またはその両方を行うために好適なデータ処理システムを使用できる。メモリ構成要素はたとえば、プログラム・コードの実際の実行の際に用いられるローカル・メモリ、バルク記憶装置、および実行中にバルク記憶装置からコードを検索しなければならない回数を減らすために少なくともいくつかのプログラム・コードの一時的記憶を提供するキャッシュ・メモリなどを含む。

【0132】

入出力すなわちI/Oデバイス(キーボード、ディスプレイ、ポインティング・デバイス、DASD、テープ、CD、DVD、サム・ドライブおよびその他のメモリ媒体などを含むがそれに限定されない)は、システムに直接結合されてもよいし、介在するI/Oコントローラを通じて結合されてもよい。システムにネットワーク・アダプタも結合される

ことによって、介在する私的または公的ネットワークを通じてデータ処理システムが他のデータ処理システムまたは遠隔のプリンタもしくは記憶装置に結合されることを可能にしてもよい。モデム、ケーブル・モデムおよびイーサネット・カードは、使用可能なタイプのネットワーク・アダプタのうちのほんのいくつかである。

【0133】

図16を参照すると、本発明の1つまたはそれ以上の局面を実施するためのホスト・コンピュータ・システム5000の代表的な構成要素が描かれている。代表的なホスト・コンピュータ5000は、コンピュータ・メモリ（すなわち中央記憶装置）5002と通信する1つまたはそれ以上のCPU5001と、記憶媒体デバイス5011および他のコンピュータまたはSANなどと通信するためのネットワーク5010に対するI/Oインタフェースとを含む。CPU5001は、設計された（architected）命令セットおよび設計された機能を有するアーキテクチャに準拠する。CPU5001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形させるための動的アドレス変換（DAT）5003を有してもよい。DATは典型的に、変換をキャッシュすることによってコンピュータ・メモリ5002のブロックへのその後のアクセスがアドレス変換の遅延を必要としないようにするための変換ルックアサイド・バッファ（translation lookaside buffer：TLB）5007を含む。典型的に、キャッシュ5009はコンピュータ・メモリ5002とプロセッサ5001との間で用いられる。キャッシュ5009は階層的であってもよく、2つ以上のCPUに対して利用可能な大型キャッシュと、大型キャッシュおよび各CPUの間のより小型で高速（低レベル）のキャッシュとを有してもよい。いくつかの実施においては、低レベルのキャッシュを分割することによって、命令フェッチおよびデータ・アクセスに対して別個の低レベル・キャッシュを提供する。一実施形態において、命令はメモリ5002から命令フェッチ・ユニット5004によってキャッシュ5009を介してフェッチされる。命令は命令復号ユニット5006において復号され、（いくつかの実施形態においては他の命令とともに）単数または複数の命令実行ユニット5008に送られる。典型的にはいくつかの実行ユニット5008が用いられ、それらはたとえば算術実行ユニット、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である。命令は実行ユニットによって、必要に応じて命令指定レジスタまたはメモリからのオペランドにアクセスしながら実行される。メモリ5002からのオペランドがアクセス（ロードまたは保存）されるとき、ロード/保存ユニット5005は典型的に、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内または内部マイクロコード（ファームウェア）内で実行されてもよいし、その両方の組み合わせによって実行されてもよい。

【0134】

示したとおり、コンピュータ・システムはローカル（または主）記憶装置に情報を含み、さらにアドレス指定、保護、ならびに参照および変更記録を含む。アドレス指定のいくつかの局面は、アドレスのフォーマット、アドレス空間の概念、さまざまなタイプのアドレス、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態様を含む。主記憶装置のあるものは永続的に割り当てられた記憶場所を含む。主記憶装置は、直接アドレス指定可能なデータの高速アクセス記憶装置をシステムに提供する。データおよびプログラムは両方とも、（入力デバイスから）主記憶装置にロードされてから処理可能になる。

【0135】

主記憶装置は1つまたはそれ以上のより小型の高速アクセスバッファ記憶装置を含んでもよく、それはキャッシュと呼ばれることもある。キャッシュは典型的にCPUまたはI/Oプロセッサに物理的に関連付けられる。物理構成の性能以外に対する影響、および別個の記憶媒体の使用は、一般的にプログラムによって観察できない。

【0136】

命令およびデータ・オペランドに対して別々のキャッシュが維持されてもよい。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（もしくは略してライ

10

20

30

40

50

ン)と呼ばれる整数境界上の連続的なバイトにおいて維持される。あるモデルは、バイトでのキャッシュ・ラインのサイズを戻す「キャッシュ属性抽出 (EXTRACT CACHE ATTRIBUTE)」命令を提供してもよい。加えてあるモデルは、記憶装置のデータもしくは命令キャッシュへのプリフェッチ、またはキャッシュからのデータの解放をもたらす「データのプリフェッチ (PREFETCH DATA)」および「相対的に長いデータのプリフェッチ (PREFETCH DATA RELATIVE LONG)」命令を提供してもよい。

【0137】

記憶装置は長い水平のビットの列として示される。ほとんどの動作について、記憶装置へのアクセスは左から右の順序で進行する。ビットの列は8ビットの単位に細分される。8ビット単位はバイトと呼ばれ、これはすべての情報フォーマットの基本的な構成単位である。記憶装置内の各バイトの場所は一意的でない整数によって識別され、その整数はそのバイト場所のアドレスまたは単にバイト・アドレスである。隣接するバイト場所は連続するアドレスを有し、アドレスは左側の0から始まって左から右の順序で進行する。アドレスは無符号2進整数であり、24、31または64ビットである。

【0138】

情報は、記憶装置とCPUまたはチャネル・サブシステムとの間で、一度に1バイトまたはバイトの群として送信される。別様に指定されない限り、たとえばz/Architecture (IBM社の登録商標) などにおいて、記憶装置内のバイトの群はその群の左端のバイトによってアドレス指定される。群のバイト数は、行われる動作によって暗示または明示される。CPU動作において用いられるとき、バイトの群はフィールドと呼ばれる。たとえばz/Architecture (IBM社の登録商標) などにおけるバイトの各群のビットは、左から右の順序で番号が付けられる。z/Architecture (IBM社の登録商標) において、左端のビットは「上位 (high-order)」ビットと呼ばれ、右端のビットは「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番号は記憶装置のアドレスではない。バイトのみがアドレス指定され得る。記憶装置内のバイトの個々のビットに対して動作するために、バイト全体がアクセスされる。(たとえばz/Architecture (IBM社の登録商標) などにおいて) バイト内のビットは左から右に0から7の番号が付けられている。アドレス内のビットは、24ビット・アドレスに対して8~31または40~63の番号が付けられてもよく、31ビット・アドレスに対して1~31または33~63の番号が付けられてもよい。それらは64ビット・アドレスに対して0~63の番号が付けられる。複数バイトのあらゆるその他の固定長フォーマットにおいて、そのフォーマットを構成するビットは0から始まる連続的な番号を付けられる。エラー検出の目的のため、および好ましくは訂正のために、各バイトまたはバイトの群とともに1つまたはそれ以上のチェック・ビットが送信されてもよい。こうしたチェック・ビットは機械によって自動的に生成され、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される。記憶オペランド・フィールドの長さが命令の演算コードによって暗示されているとき、そのフィールドは固定長を有するといい、その長さは1、2、4、8または16バイトであってもよい。命令によってはもっと大きなフィールドが実施されてもよい。記憶オペランド・フィールドの長さが暗示されずに明確に記述されるとき、そのフィールドは可変長を有するという。可変長オペランドの長さは、1バイト単位 (または命令によっては2バイトの倍数もしくはその他の倍数の単位) で変動し得る。情報が記憶装置に位置するとき、たとえ記憶装置への物理経路の幅が保存されるフィールドの長さよりも大きかったとしても、指示されたフィールドに含まれるバイト場所の内容のみが置換される。

【0139】

情報の特定のユニットは、記憶装置の整数境界にくる。ある情報のユニットの記憶アドレスがユニットの長さの倍数バイトであるとき、そのユニットに対する境界は整数と呼ばれる。整数境界における2、4、8および16バイトのフィールドには特別な名前が与えられている。ハーフワードは2バイト境界における2つの連続するバイトの群であり、命

10

20

30

40

50

令の基本的な構成単位である。ワードは4バイト境界における4つの連続するバイトの群である。ダブルワードは8バイト境界における8つの連続するバイトの群である。クワドワード (quadword) は16バイト境界における16の連続するバイトの群である。記憶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ドワードを指示するとき、アドレスの2進表現はそれぞれ1、2、3または4個の右端0ビットを含む。命令は2バイト整数境界上にくる。ほとんどの命令の記憶オペランドは境界整列要求を有さない。

【0140】

命令およびデータ・オペランドに対して別々のキャッシュを実施するデバイスにおいては、後でフェッチされる命令が保存によって変更されるかどうかに関わらず、後で命令がフェッチされるキャッシュ・ラインにプログラムが保存を行うときに、かなりの遅延が生じることがある。

【0141】

一実施形態において、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こともあり、これらはいずれも本発明に整合する) によって実行されてもよい。図16を参照して、本発明を実施するソフトウェア・プログラム・コードは典型的に、たとえばCD-ROMドライブ、テープ・ドライブまたはハード・ドライブなどの長期記憶媒体デバイス5011から、ホスト・システム5000のプロセッサ5001によってアクセスされる。ソフトウェア・プログラム・コードは、たとえばディスケット、ハード・ドライブまたはCD-ROMなど、データ処理システムとともに使用するためのさまざまな公知の媒体のいずれかにおいて実現されてもよい。コードはこうした媒体に分散されてもよいし、1つのコンピュータ・システムのコンピュータ・メモリ5002または記憶装置からネットワーク5010を通じて他のコンピュータ・システムへ、こうした他のシステムのユーザによる使用のためにユーザに分散されてもよい。

【0142】

ソフトウェア・プログラム・コードは、さまざまなコンピュータ構成要素および1つまたはそれ以上のアプリケーション・プログラムの機能および対話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、記憶媒体デバイス5011から相対的により高速のコンピュータ記憶装置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となる。メモリ内、物理媒体上のソフトウェア・プログラム・コードの実現、もしくはネットワークを介したソフトウェア・コードの分散、またはその両方を行うための技術および方法は周知であり、本明細書においてさらに考察されない。プログラム・コードが有形の媒体 (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がそれに限定されない) において作成および保存されるとき、それはしばしば「コンピュータ・プログラム製品」と呼ばれる。コンピュータ・プログラム製品媒体は典型的に処理回路によって読取り可能であり、好ましくはその処理回路による実行のためのコンピュータ・システムにおいて読取り可能である。

【0143】

図17は、本発明を実施し得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17のシステム5020は、任意の周辺デバイスを含むたとえばパーソナル・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つまたはそれ以上のプロセッサ5026と、公知の技術に従ってプロセッサ (単数または複数) 5026およびシステム5021の他の構成要素を接続して通信を可能にするために用いられるバスとを含む。バスはプロセッサ5026をメモリ5025および長期記憶装置5027に接続し、長期記憶装置5027はたとえばハード・ドライブ (たとえば磁気媒体、CD、DVDおよびフラッシュ・メモリなどのいずれかを含む) またはテープ・ドライブなどを含んでもよい。加えてシステム5021はユーザ・インタフェース・アダ

プタを含んでもよく、これはバスを介してマイクロプロセッサ5026を1つまたはそれ以上のインタフェース・デバイス、たとえばキーボード5024、マウス5023、プリンタ/スキャナ5030、もしくはたとえばタッチ・センシティブ・スクリーン、デジタル化エントリ・パッドなどのあらゆるユーザ・インタフェース・デバイスであってもよいその他のインタフェース・デバイスまたはその組み合わせなどに接続する。さらにバスは、ディスプレイ・アダプタを介してたとえばLCDスクリーンまたはモニタなどのディスプレイ・デバイス5022をマイクロプロセッサ5026に接続する。

【0144】

システム5021は、ネットワーク5029と通信できるネットワーク・アダプタ5028によって、他のコンピュータまたはコンピュータのネットワークと通信してもよい。ネットワーク・アダプタの例は、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。代替的に、システム5021はたとえばCDPD(セルラ・デジタル・パケット・データ(cellular digital packet data))カードなどのワイヤレス・インタフェースを用いて通信してもよい。システム5021はローカル・エリア・ネットワーク(LAN)または広域ネットワーク(WAN)においてこうした他のコンピュータと関連付けられてもよいし、システム5021は別のコンピュータとのクライアント/サーバ配置におけるクライアントなどであってもよい。これらの構成ならびに適切な通信ハードウェアおよびソフトウェアは、すべて当該技術分野において公知である。

【0145】

図18は、本発明を実施し得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、たとえばワイヤレス・ネットワークおよびワイヤード・ネットワークなどの複数の個別のネットワークを含んでもよく、各ネットワークは複数の個別のワークステーション5041、5042、5043、5044を含んでもよい。加えて当業者が認識するとおり、1つまたはそれ以上のLANが含まれてもよく、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んでもよい。

【0146】

なおも図18を参照して、ネットワークはメインフレーム・コンピュータまたはサーバ、たとえばゲートウェイ・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(リモート・サーバ5048、これはデータ・リポジトリにアクセスしてもよく、さらにワークステーション5045から直接アクセスされてもよい)などをさらに含んでもよい。ゲートウェイ・コンピュータ5046は個別ネットワークの各々へのエントリのポイントの働きをする。ゲートウェイは、1つのネットワーキング・プロトコルを別のネットワーキング・プロトコルと接続するときに必要なとされる。ゲートウェイ5046は、好ましくは通信リンクによって別のネットワーク(たとえばインターネット5047など)に結合されてもよい。加えてゲートウェイ5046は、通信リンクを用いて1つまたはそれ以上のワークステーション5041、5042、5043、5044に直接結合されてもよい。ゲートウェイ・コンピュータは、インターナショナル・ビジネス・マシーンズ社より入手可能なIBM(IBM社の登録商標) e S e r v e r (IBM社の商標) S y s t e m z (IBM社の登録商標)サーバを用いて実現されてもよい。

【0147】

図17および図18を同時に参照して、本発明を実施し得るソフトウェア・プログラミング・コードは、たとえばCD-ROMドライブまたはハード・ドライブなどの長期記憶媒体5027からシステム5020のプロセッサ5026によってアクセスされてもよい。ソフトウェア・プログラミング・コードは、たとえばディスク、ハード・ドライブまたはCD-ROMなど、データ処理システムとともに使用するためのさまざまな公知の媒体のいずれかにおいて実現されてもよい。コードはこうした媒体に分散されてもよいし、1つのコンピュータ・システムのメモリまたは記憶装置からネットワークを通じて他のコンピュータ・システムへ、こうした他のシステムのユーザによる使用のためにユーザ5050、5051に分散されてもよい。

【0148】

代替的に、プログラミング・コードはメモリ5025内で実現されて、プロセッサ5026によってプロセッサ・バスを用いてアクセスされてもよい。こうしたプログラミング・コードは、さまざまなコンピュータ構成要素および1つまたはそれ以上のアプリケーション・プログラム5032の機能および対話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、記憶媒体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となる。メモリ内、物理媒体上のソフトウェア・プログラミング・コードの実現、もしくはネットワークを介したソフトウェア・コードの分散、またはその両方を行うための技術および方法は周知であり、本明細書においてさらに考察されない。プログラム・コードが有形の媒体（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がそれに限定されない）において作成および保存されるとき、それはしばしば「コンピュータ・プログラム製品」と呼ばれる。コンピュータ・プログラム製品媒体は典型的に処理回路によって読取り可能であり、好ましくはその処理回路による実行のためのコンピュータ・システムにおいて読取り可能である。

10

【0149】

プロセッサにとって最も容易に利用可能なキャッシュ（通常はプロセッサの他のキャッシュよりも速くて小型）は最低（L1またはレベル1）キャッシュであり、主記憶装置（主メモリ）は最高レベルのキャッシュ（3レベルあるとすればL3）である。最低レベル・キャッシュはしばしば、実行されるべき機械命令を保持する命令（instruction）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（data）キャッシュ（D-キャッシュ）とに分割される。

20

【0150】

図19を参照して、プロセッサ5026に対する例示的なプロセッサ実施形態が示される。典型的に、プロセッサ性能を改善するためにメモリ・ブロックをバッファリングするために1つまたはそれ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、使用される可能性のあるメモリ・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、128または256バイトのメモリ・データである。命令のキャッシュと、データのキャッシュとに対してしばしば別々のキャッシュが用いられる。当該技術分野において周知のさまざまな「スヌープ」アルゴリズムによって、キャッシュ・コヒーレンス（メモリおよびキャッシュにおけるラインのコピーの同期）がしばしば提供される。プロセッサ・システムの主メモリ記憶装置5025はしばしばキャッシュと呼ばれる。4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、主記憶装置5025はレベル5（L5）キャッシュと呼ばれることがある。なぜなら主記憶装置5025は典型的により高速であり、コンピュータ・システムが利用可能な不揮発性記憶装置（DASD、テープなど）の部分のみを保持するからである。主記憶装置5025は、オペレーティング・システムによって主記憶装置5025の中および外にページングされるデータのページを「キャッシュ」する。

30

【0151】

プログラム・カウンタ（命令カウンタ）5061は、実行されるべき現行命令のアドレスを追跡する。z/Architecture（IBM社の登録商標）プロセッサにおけるプログラム・カウンタは64ビットであり、前のアドレス指定の制限を支持するために31ビットまたは24ビットに短縮されてもよい。プログラム・カウンタは典型的にコンピュータのPSW（プログラム状態ワード（program status word））において実現されることによって、文脈切り換えの際に持続するようにされる。つまり、プログラム・カウンタ値を有する進行中のプログラムは、たとえばオペレーティング・システム（プログラム環境からオペレーティング・システム環境への文脈切り換え）などによって中断されることがある。プログラムのPSWはプログラムが活動していない間もプログラム・カウンタ値を維持し、オペレーティング・システムが実行している間はオペレーティング・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

40

50

、プログラム・カウンタは現行命令のバイト数に等しい量だけ増分される。RISC (縮小命令セット・コンピューティング (Reduced Instruction Set Computing)) 命令は典型的に固定長であるのに対し、CISC (複雑命令セット・コンピューティング (Complex Instruction Set Computing)) 命令は典型的に可変長である。IBM z/Architecture (IBM社の登録商標) の命令は、2、4または6バイトの長さのCISC命令である。プログラム・カウンタ5061は、たとえば文脈切り換え動作または分岐命令の分岐取得動作 (branch taken operation) のいずれかなどによって修正される。文脈切り換え動作において、現行プログラム・カウンタ値は、実行中のプログラムに関する他の状態情報 (条件コードなど) とともにプログラム状態ワードにセーブされ、実行されるべき新しいプログラム・モジュールの命令を指す新しいプログラム・カウンタ値がロードされる。分岐取得動作は、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って、プログラムが判定を行うかまたはプログラム内でループすることを可能にするために行われる。

10

【0152】

典型的に、命令フェッチ・ユニット5055はプロセッサ5026の代わりに命令をフェッチするために用いられる。フェッチ・ユニットは「次の順次命令」か、分岐取得命令の目標命令か、または文脈切り換え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。最近の命令フェッチ・ユニットはしばしばプリフェッチ技術を用いて、プリフェッチされた命令が用いられ得る可能性に基づいて命令を投機的にプリフェッチする。たとえば、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、付加的なバイトのさらなる順次命令とをフェッチしてもよい。

20

【0153】

フェッチされた命令は、次いでプロセッサ5026によって実行される。実施形態において、フェッチされた命令 (単数または複数) はフェッチ・ユニットのディスパッチ・ユニット5056に送られる。ディスパッチ・ユニットは命令 (単数または複数) を復号化し、復号化した命令 (単数または複数) に関する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、命令フェッチ・ユニット5055から復号化算術命令に関する情報を受け取り、その命令のオペコードに従うオペランドに対して算術演算を行う。オペランドは、好ましくはメモリ5025、設計された (architected) レジスタ5059、または実行中の命令の即値フィールドのいずれかから実行ユニット5057に提供される。実行の結果を保存するときは、メモリ5025、レジスタ5059、またはその他の機械ハードウェア (たとえば制御レジスタ、PSWレジスタなど) のいずれかに保存する。

30

【0154】

プロセッサ5026は典型的に、命令の機能を実行するための1つまたはそれ以上のユニット5057、5058、5060を有する。図20を参照して、実行ユニット5057はインタフェース論理5071によって、設計された一般的レジスタ5059、復号/ディスパッチ・ユニット5056、ロード保存ユニット5060、およびその他のプロセッサ・ユニット5065と通信してもよい。実行ユニット5057は、算術論理ユニット (arithmetic logic unit: ALU) 5066が演算を行うための情報を保持するためにいくつかのレジスタ回路5067、5068、5069を用いてもよい。ALUはたとえば加算、減算、乗算および除算などの算術演算、ならびにたとえば論理積 (and)、論理和 (or) および排他的論理和 (exclusive-or: XOR)、回転およびシフトなどの論理関数を実行する。好ましくは、ALUは設計に依存する専門演算を支持する。他の回路は、たとえば条件コードおよび回復支持論理などを含むその他の設計された機能5072を提供してもよい。典型的に、ALU演算の結果は出力レジスタ回路5070に保持され、出力レジスタ回路5070はその結果をさまざまな他の処理機能に転送できる。プロセッサ・ユニットの配置は多数あり、本記載は一実施形態の代表的な理解を提供することのみが意図されている。

40

50

【 0 1 5 5 】

たとえば加算 (ADD) 命令は算術および論理機能を有する実行ユニット 5057 において実行されるのに対し、たとえば浮動小数点命令は専門の浮動小数点能力を有する浮動小数点実行において実行される。好ましくは、実行ユニットはオペランドに対してオペコードの定めた関数を実行することによって、命令が識別するオペランドに対する演算を行う。たとえば ADD 命令は、命令のレジスタ・フィールドによって識別される 2 つのレジスタ 5059 において見出されるオペランドに対して実行ユニット 5057 が実行してもよい。

【 0 1 5 6 】

実行ユニット 5057 は 2 つのオペランドに対する算術加算を行い、その結果を第 3 のオペランドに保存し、この第 3 のオペランドは第 3 のレジスタであってもよいし、2 つのソース・レジスタの一方であってもよい。好ましくは実行ユニットは、たとえばシフト、回転、論理積、論理和および XOR などのさまざまな論理関数、ならびに加算、減算、乗算、除算のいずれかを含むさまざまな代数関数を実行し得る算術論理ユニット (ALU) 5066 を用いる。ALU 5066 のあるものはスカラ演算のために設計されており、あるものは浮動小数点のために設計されている。データは、アーキテクチャに依存してビッグ・エンディアン (最下位バイトが最高バイト・アドレスにある) であってもよいし、リトル・エンディアン (最下位バイトが最低バイト・アドレスにある) であってもよい。IBM z/Architecture (IBM 社の登録商標) はビッグ・エンディアンである。符号付きフィールドは符号と大きさとであってもよく、アーキテクチャに依存して 1 の補数または 2 の補数であってもよい。2 の補数における負の値または正の値はいずれも ALU 内で加算のみを必要とするため、ALU が減算能力を設計する必要がないという点において、2 の補数が有利である。数は通常略記され、たとえば 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るが、通常は 4 K バイト (キロバイト (Kilo-byte)) ブロックと記載される。

【 0 1 5 7 】

図 21 を参照して、分岐命令を実行するための分岐命令情報は典型的に分岐ユニット 5058 に送られ、分岐ユニット 5058 はしばしばたとえば分岐履歴表 (branch history table) 5082 などの分岐予測アルゴリズムを用いて、他の条件付き動作が完了する前に分岐の結果を予測する。条件付き動作が完了する前に、現行分岐命令の目標がフェッチされて投機的に実行される。条件付き動作が完了したとき、投機的に実行された分岐命令は、条件付き動作および推測される結果の条件に基づいて完了または廃棄される。典型的な分岐命令は条件コードをテストして、もし条件コードが分岐命令の分岐要求を満たしていれば目標アドレスに分岐してもよく、目標アドレスはたとえば命令のレジスタ・フィールドまたは即値フィールドなどに見出される数を含むいくつかの数に基づいて算出されてもよい。分岐ユニット 5058 は、複数の入力レジスタ回路 5075、5076、5077 および出力レジスタ回路 5080 を有する ALU 5074 を用いてもよい。分岐ユニット 5058 は、たとえば一般的レジスタ 5059、復号ディスパッチ・ユニット 5056 またはその他の回路 5073 などと通信してもよい。

【 0 1 5 8 】

命令群の実行は、たとえばオペレーティング・システムが開始する文脈切り換え、文脈切り換えをもたらすプログラム例外もしくはエラー、文脈切り換えをもたらす I/O 割り込み信号、または (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ディング活動などを含むさまざまな理由で中断され得る。好ましくは、文脈切り換え動作は現在実行中のプログラムに関する状態情報をセーブしてから、呼び出された別のプログラムに関する状態情報をロードする。状態情報は、たとえばハードウェア・レジスタまたはメモリなどにセーブされてもよい。状態情報は、好ましくは実行されるべき次の命令を指すプログラム・カウンタ値と、条件コードと、メモリ変換情報と、設計されたレジスタ内容とを含む。文脈切り換え活動は、ハードウェア回路、アプリケーション・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

10

20

30

40

50

ド、ピココードまたはライセンス内部コード (l i c e n s e d i n t e r n a l c o d e : L I C)) の単体または組み合わせによって実行されてもよい。

【 0 1 5 9 】

プロセッサは、命令が定める方法に従ってオペランドにアクセスする。命令は、その命令の一部の値を用いた即値オペランドを提供してもよいし、汎用レジスタまたは特定目的のレジスタ（たとえば浮動小数点レジスタ）のいずれかを明示する1つまたはそれ以上のレジスタ・フィールドを提供してもよい。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暗黙レジスタを用いてもよい。命令は、オペランドに対するメモリ場所を用いてもよい。オペランドのメモリ場所は、レジスタ、即値フィールド、または z / A r c h i t e c t u r e (I B M 社の登録商標) の長変位機構によって例示されるとおりレジスタと即値フィールドとの組み合わせによって提供されてもよく、ここでは命令が基底レジスタと、インデックス・レジスタと、即値フィールド（変位フィールド）とを定め、それらがともに加算されてたとえばメモリ内のオペランドのアドレスを提供する。別様に示されない限り、本明細書における場所とは典型的に主メモリ（主記憶装置）内の場所を意味する。

【 0 1 6 0 】

図22を参照して、プロセッサはロード/保存ユニット5060を用いて記憶装置にアクセスする。ロード/保存ユニット5060は、メモリ5053内の目標オペランドのアドレスを得てそのオペランドをレジスタ5059または別のメモリ5053の場所にロードすることによってロード動作を行ってもよいし、メモリ5053内の目標オペランドのアドレスを得てレジスタ5059または別のメモリ5053の場所から得られたデータをメモリ5053内の目標オペランド場所に保存することによって保存動作を行ってもよい。ロード/保存ユニット5060は投機的であってもよく、命令順序に関して順不同の順序でメモリにアクセスしてもよいが、ロード/保存ユニット5060はプログラムに対して命令が順番に実行されたという外観を維持するべきである。ロード/保存ユニット5060は一般的レジスタ5059、復号/ディスパッチ・ユニット5056、キャッシュ/メモリ・インタフェース5053またはその他の構成要素5083と通信してもよく、さらに記憶装置アドレスを算出し、かつ動作を順番に保つためのパイプライン順序付けを提供するためのさまざまなレジスタ回路、A L U 5 0 8 5 および制御論理5090を含む。当該技術分野において周知であるとおり、いくつかの動作は順不同であってもよいが、ロード/保存ユニットはその順不同の動作が順番に行われたようにプログラムに見せるための機能を提供する。

【 0 1 6 1 】

好ましくは、アプリケーション・プログラムが「見る」アドレスはしばしば仮想アドレスと呼ばれる。仮想アドレスは「論理アドレス」および「有効アドレス」と呼ばれることもある。これらの仮想アドレスは、さまざまな動的アドレス変換 (D A T) 技術の1つによって物理的メモリ場所にリダイレクトされるという点において仮想的であり、このD A T 技術は、単に仮想アドレスの前にオフセット値を付けること、1つまたはそれ以上の変換表を介して仮想アドレスを変換することを含むがそれに限定されず、好ましくは変換表は少なくともセグメント表およびページ表の単体または組み合わせを含み、好ましくはセグメント表はページ表を指すエントリを有する。z / A r c h i t e c t u r e (I B M 社の登録商標) においては、領域第1表、領域第2表、領域第3表、セグメント表および任意のページ表を含む変換の階層が提供される。アドレス変換の性能はしばしば、仮想アドレスに関連の物理的メモリ場所にマップするエントリを含む変換ルックアサイド・バッファ (T L B) を使用することによって改善される。D A T が変換表を用いて仮想アドレスを変換するときに、エントリが作成される。その後仮想アドレスを使用するときには、遅い順次変換表アクセスではなく、速いT L B のエントリを使用できる。T L B 内容は、L R U (最長時間未使用 (L e a s t R e c e n t l y u s e d)) を含むさまざまな置換アルゴリズムによって管理されてもよい。

【 0 1 6 2 】

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、各プロセッサは、たとえばI/O、キャッシュ、TLBおよびメモリなどの共用リソースをコヒーレンスのためにインタロックさせて保つ責任を負う。典型的に、キャッシュのコヒーレンスの維持には「スヌープ」技術が用いられる。スヌープ環境において、共用を容易にするために各キャッシュ・ラインには共用状態、排他的状態、変更状態、無効状態などのうちいずれか1つであるというマークが付けられてもよい。

【0163】

I/Oユニット5054(図19)は、たとえばテープ、ディスク、プリンタ、ディスプレイ、およびネットワークなどを含む周辺デバイスに接続するための手段をプロセッサに提供する。I/Oユニットはしばしばソフトウェア・ドライバによってコンピュータ・プログラムに提供される。たとえばIBM(IBM社の登録商標)のSystem z(IBM社の登録商標)などのメインフレームにおいて、チャンネル・アダプタおよびオープン・システム・アダプタは、オペレーティング・システムと周辺デバイスとの間の通信を提供するメインフレームのI/Oユニットである。

【0164】

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つまたはそれ以上の局面から利益を得ることもできる。例として、ある環境はエミュレータ(例、ソフトウェアまたはその他のエミュレーション機構)を含んでもよく、ここでは特定のアーキテクチャ(たとえば命令実行、設計された機能、たとえばアドレス変換など、および設計されたレジスタなどを含む)またはその部分集合が(たと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・システム上などで)エミュレートされる。こうした環境において、エミュレータを実行するコンピュータはエミュレートされる能力とは異なるアーキテクチャを有するかもしれないが、エミュレータの1つまたはそれ以上のエミュレーション機能が本発明の1つまたはそれ以上の局面を実行してもよい。一例として、エミュレーション・モードにおいて、エミュレートされる特定の命令または動作が復号化され、適切なエミュレーション機能が構築されて、個々の命令または動作を実行する。

【0165】

エミュレーション環境において、ホスト・コンピュータはたとえば、命令およびデータを保存するためのメモリと、メモリから命令をフェッチし、かつ任意にはフェッチされた命令に対するローカル・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け取り、フェッチされた命令のタイプを定めるための命令復号ユニットと、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、メモリからレジスタヘデータをロードすること、レジスタからメモリにデータを戻して保存すること、または復号ユニットが定めるとおりにある種の算術的または論理的演算を行うことを含んでもよい。一例において、各ユニットはソフトウェアにおいて実現される。たとえば、それらのユニットによって行われる動作は、エミュレータ・ソフトウェア内の1つまたはそれ以上のサブルーチンとして実現される。

【0166】

より特定的には、メインフレームにおいて、設計された機械命令はしばしばコンパイラ・アプリケーションによって、現在通常は「C」プログラマであるプログラマによって使用される。記憶媒体に保存されるこれらの命令は、z/Architecture(IBM社の登録商標)IBM(IBM社の登録商標)サーバにおいてネイティブに実行されてもよいし、代替的には他のアーキテクチャを実行している機械において実行されてもよい。それらは既存および将来のIBM(IBM社の登録商標)メインフレーム・サーバ、ならびにIBM(IBM社の登録商標)の他の機械(例、Power Systems(IBM社の登録商標)サーバおよびSystem x(IBM社の登録商標)サーバ)においてエミュレートされてもよい。それらは、IBM(IBM社の登録商標)、インテル(登録商標)、AMD(登録商標)およびその他によって製造されるハードウェアを用いた多様な機械においてリナックス(登録商標)を実行する機械において実行されてもよい。z/Architecture(IBM社の登録商標)の下でのそのハードウェアにおける実行に

加えて、リナックス（登録商標）、およびハーキュリーズ（Hercules）またはFSI（ファンダメンタル・ソフトウェア・インク（Fundamental Software, Inc.））によるエミュレーションを用いる機械が用いられてもよく、ここでは一般的に実行がエミュレーション・モードである。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て、エミュレートされるプロセッサのアーキテクチャをエミュレートする。上に参照したエミュレータ製品に関する情報は、ワールド・ワイド・ウェブのwww.hercules-390.orgおよびwww.funsoft.comにおいてそれぞれ入手可能である。

【0167】

ネイティブ・プロセッサは典型的に、エミュレートされるプロセッサのエミュレーションを行うための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るプロセッサ・アーキテクチャの命令をフェッチして実行する責任を負う。エミュレーション・ソフトウェアはエミュレートされるプログラム・カウンタを維持して命令境界を追跡する。エミュレーション・ソフトウェアは一度に1つまたはそれ以上のエミュレートされる機械命令をフェッチして、その1つまたはそれ以上のエミュレートされる機械命令をネイティブ・プロセッサによる実行のためにネイティブ機械命令の対応する群に変換してもよい。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されてもよい。それにもかかわらず、エミュレーション・ソフトウェアはエミュレートされるプロセッサ・アーキテクチャのアーキテクチャ規則を維持することによって、エミュレートされるプロセッサに対して書き込まれたアプリケーションおよびオペレーティング・システムが正しく動作することを確実にする。さらに、エミュレーション・ソフトウェアは、制御レジスタ、汎用レジスタ、浮動小数点レジスタ、たとえばセグメント表およびページ表などを含む動的アドレス変換関数、割り込み機構、文脈切り換え機構、時刻（Time of Day：TOD）クロック、ならびにI/Oサブシステムに対する設計されたインタフェースを含むがそれに限定されない、エミュレートされるプロセッサ・アーキテクチャによって識別されるリソースを提供することによって、エミュレートされるプロセッサにおいて実行するよう設計されたオペレーティング・システムまたはアプリケーション・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・プロセッサにおいて実行され得るようにする。

【0168】

エミュレートされる特定の命令が復号化され、個々の命令の機能を行うためにサブルーチンが呼び出される。エミュレートされるプロセッサ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、たとえば「C」サブルーチンもしくはドライバ、または好ましい実施形態の説明を理解した後の当業者の技術範囲内にある、特定のハードウェアに対するドライバを提供する何らかの他の方法などにおいて実現される。ボーソレーユ（Beausoleil）らによる「ハードウェア・エミュレーションのためのマルチプロセッサ（Multiprocessor for Hardware Emulation）」と題する米国特許証（U.S. Letters Patent）第5,551,013号、スカルジ（Scalzi）らによる「目標プロセッサにおけるインコンパチブル命令をエミュレートするための保存された目標ルーチンの前処理（Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor）」と題する米国特許証第6,009,261号、デビディアン（Davidian）らによる「ゲスト命令をエミュレートするエミュレーション・ルーチンに直接アクセスするためのゲスト命令の復号化（Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions）」と題する米国特許証第5,574,873号、ゴリシェク（Gorishchek）らによる「システム内で非ネイティブ・コードを実行させるコプロセッサ支持に用い

10

20

30

40

50

られる対称型多重処理バスおよびチップセット (Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System)」と題する米国特許証第 6,308,255 号、レシン (Lethin) らによる「アーキテクチャ・エミュレーションのための動的最適化オブジェクト・コード・トランスレータおよび動的最適化オブジェクト・コード変換法 (Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method)」と題する米国特許証第 6,463,582 号、エリック・トラウト (Eric Traut) による「ホスト命令の動的再コンパイルを通じてホスト・コンピュータ上でゲスト命令をエミュレートする方法 (Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions)」と題する米国特許証第 5,790,825 号、およびその他多数を含むがそれに限定されないさまざまなソフトウェアおよびハードウェア・エミュレーション特許は、当業者が利用可能な目標機械に対する異なる機械のために設計された命令フォーマットのエミュレーションを達成するためのさまざまな公知のやり方を示している。

10

【0169】

20

図 23 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、エミュレート後のホスト・コンピュータ・システム 5092 の例が提供される。エミュレート後のホスト・コンピュータ・システム 5092 において、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、ホスト・コンピュータ 5000' のプロセッサ 5091 とは異なるネイティブ命令セット・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。エミュレート後のホスト・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 にアクセス可能なメモリ 5094 を有する。実施形態例において、メモリ 5094 はホスト・コンピュータ・メモリ 5096 部分とエミュレーション・ルーチン 5097 部分とに分割される。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うエミュレート後のホスト・コンピュータ 5092 のプログラムに対して利用可能である。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設計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 より得られ、エミュレーション・プロセッサ 5093 はさらに、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定めるために、アクセスされたホスト命令 (単数または複数) を復号化してもよい順序およびアクセス / 復号ルーチンにおいて得られる 1 つまたはそれ以上の命令 (単数または複数) を用いることによって、ホスト・コンピュータ・メモリ 5096 内のプログラムからの実行のためのホスト命令にアクセスしてもよい。ホスト・コンピュータ・システム 5000' アーキテクチャに対して定められるその他の機構は、たとえば汎用レジスタ、制御レジスタ、動的アドレス変換ならびに I/O サブシステム支持およびプロセッサ・キャッシュなどの機構を含む設計された機構ルーチンによってエミュレートされてもよい。加えてエミュレーション・ルーチンは、エミュレーション・プロセッサ 5093 において利用可能な機能 (一般的レジスタ、および仮想アドレスの動的変換など) を利用して、エミュレーション・ルーチンの性能を改善してもよい。加えて、ホスト・コンピュータ 5000' の機能のエミュレーションにおいてプロセッサ 5093 を助けるために、特殊なハードウェアおよびオフロード・エンジンが提供されてもよい。

30

40

【0170】

本明細書において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みのもので

50

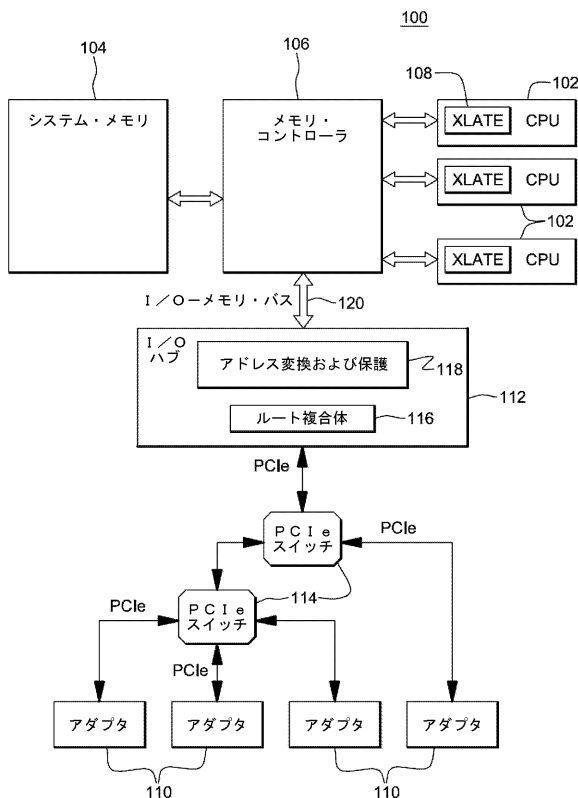
あって、本発明を限定することは意図されない。本明細書において用いられる単数形「a」、「an」および「the」は、状況が明らかに別様を示していない限り、複数形をも含むことが意図される。さらに、「含む (comprises)」もしくは「含む (comprising)」という用語またはその両方が本明細書において用いられるとき、それは述べられる特徴、完全体、ステップ、動作、構成要素もしくは成分またはその組み合わせの存在を特定するが、1つまたはそれ以上の他の特徴、完全体、ステップ、動作、構成要素、成分もしくはその群またはその組み合わせの存在または追加を排除するものではないことが理解されるだろう。

【0171】

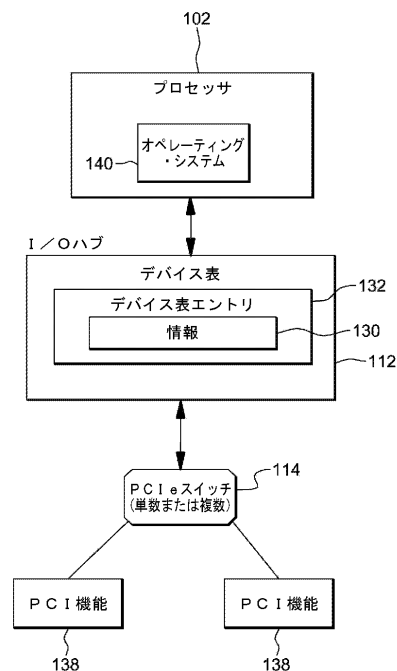
以下の請求項におけるすべての手段またはステップ・プラス機能 (means or step plus function) 要素に対応する構造、材料、動作、および同等物は、もしあれば、特定の請求される他の請求要素と組み合わせてその機能を行うためのあらゆる構造、材料または動作を含むことが意図される。本発明の説明は例示および説明の目的のために提供されたものであるが、網羅的になったり、開示される形に本発明を制限したりすることは意図されない。本発明の範囲から逸脱することなく、通常の当業者には多くの修正および変更が明らかになるだろう。実施形態は、本発明の原理および実際の適用を最も良く説明し、他の通常の当業者が予期される特定の使用に好適であるようなさまざまな修正を伴うさまざまな実施形態に対して本発明を理解できるようにするために選択されて記載されたものである。

10

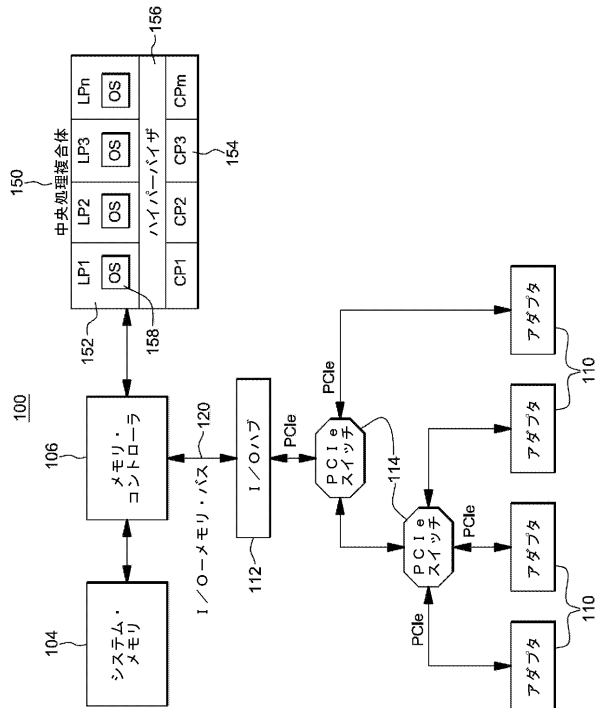
【図1】



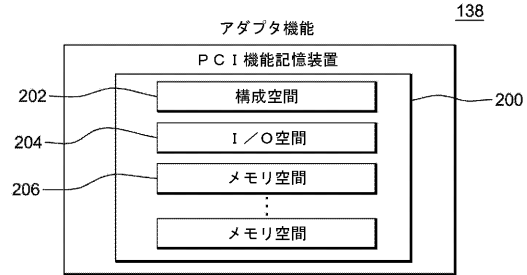
【図2】



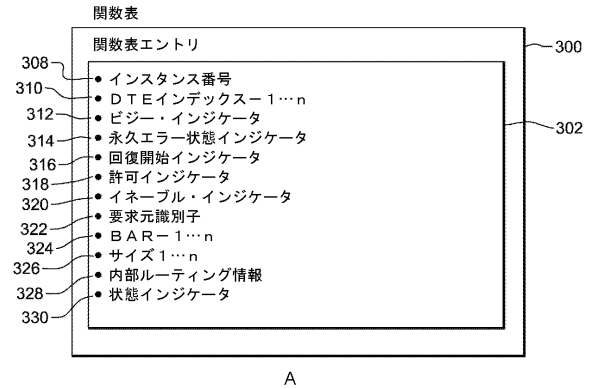
【 図 3 】



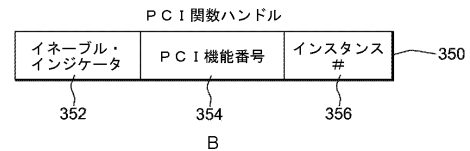
【 図 4 】



【 図 5 】

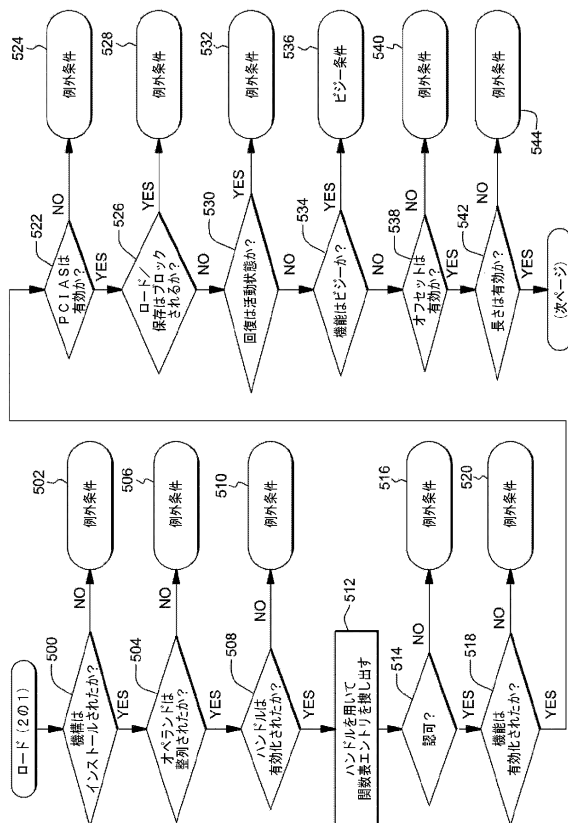


A

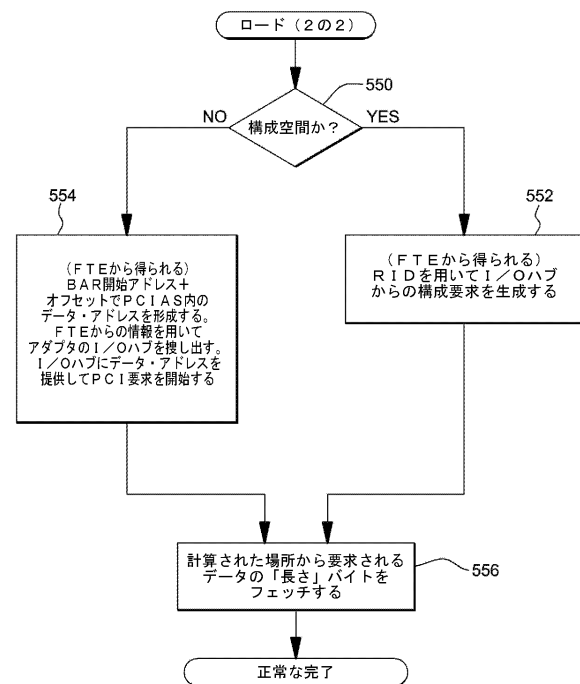


B

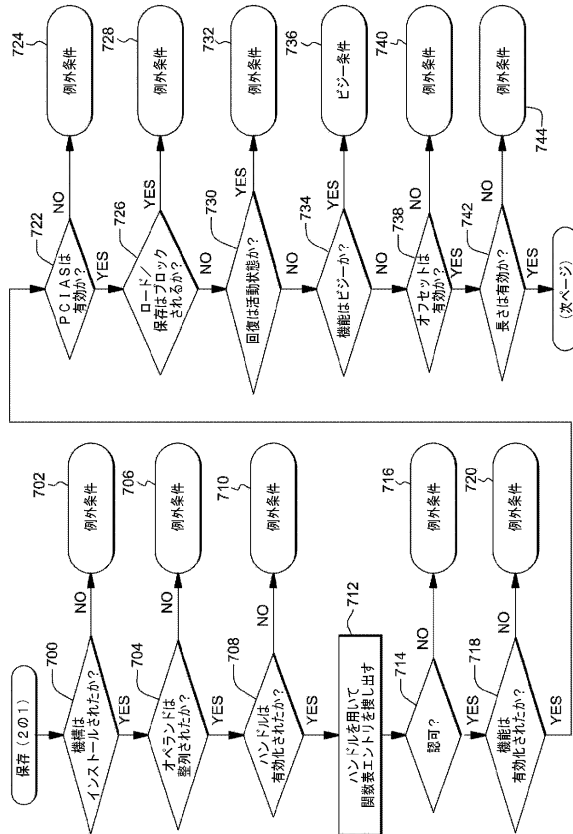
【圖 7】



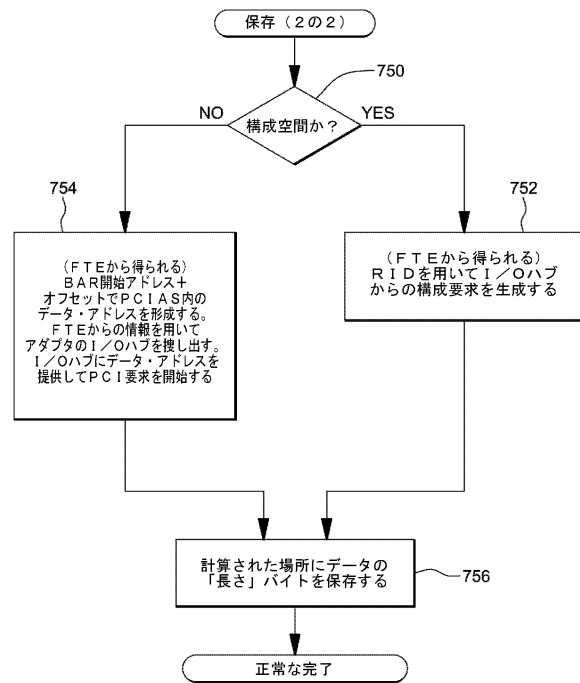
【 図 8 】



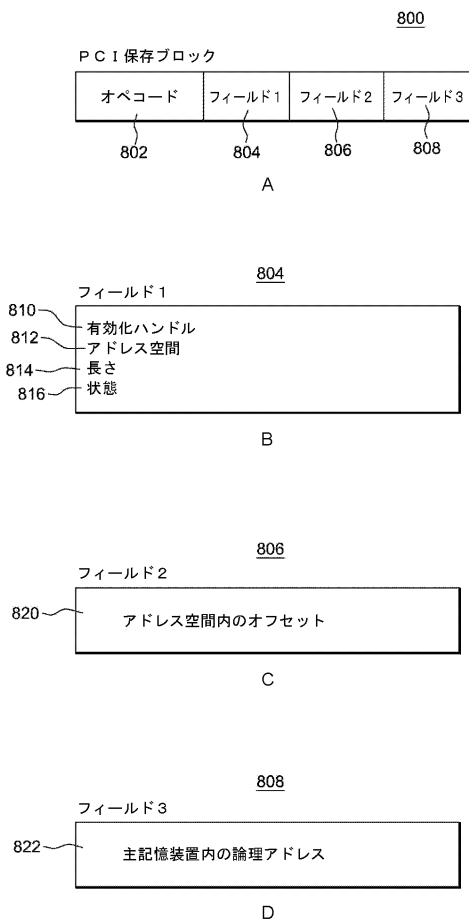
【図 10】



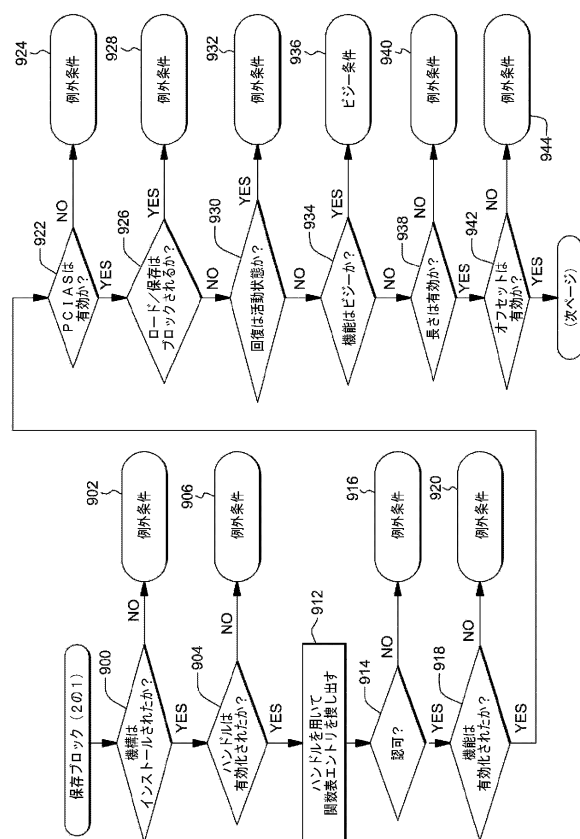
【図 11】



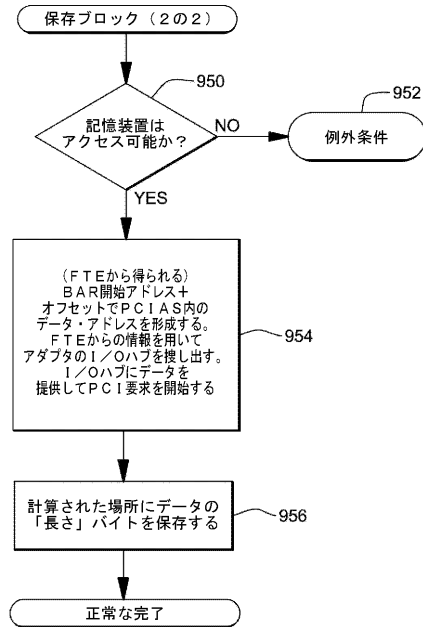
【図 12】



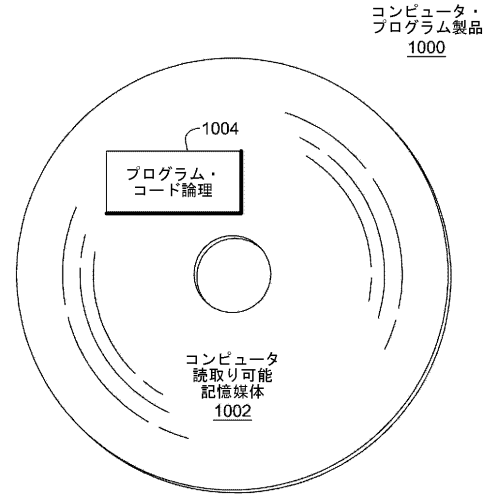
【図 13】



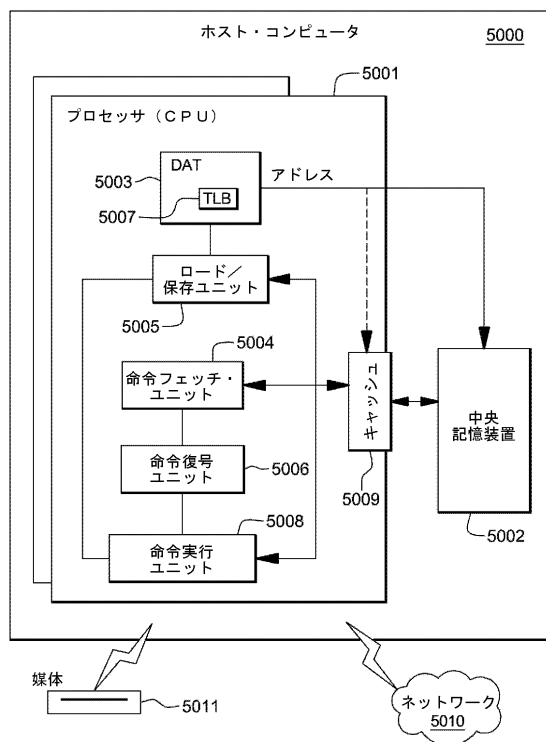
【図 14】



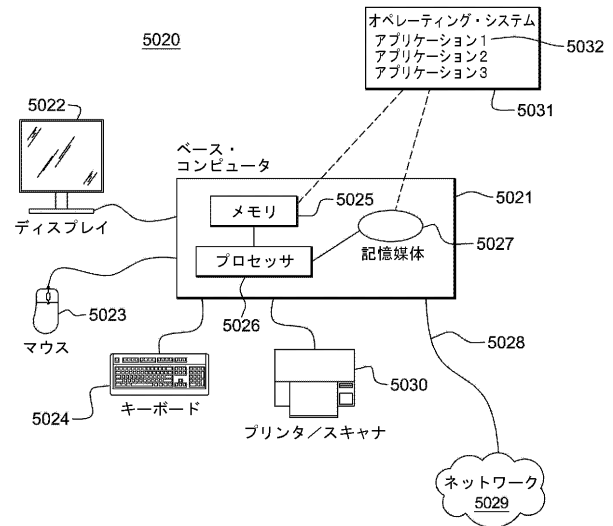
【図 15】



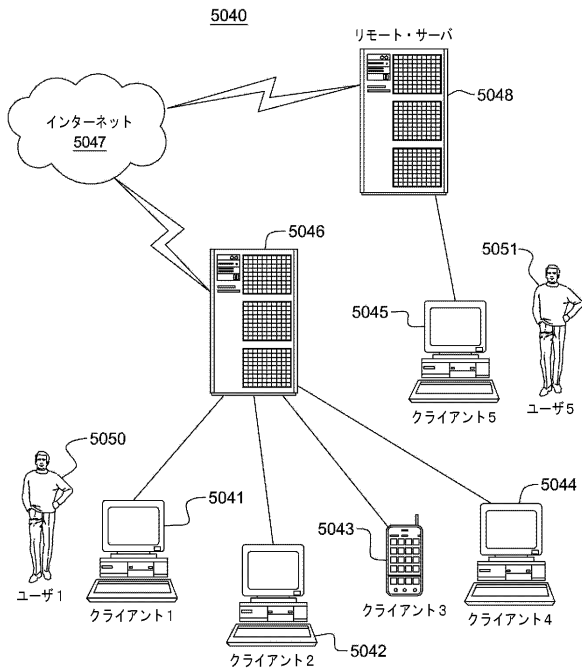
【図 16】



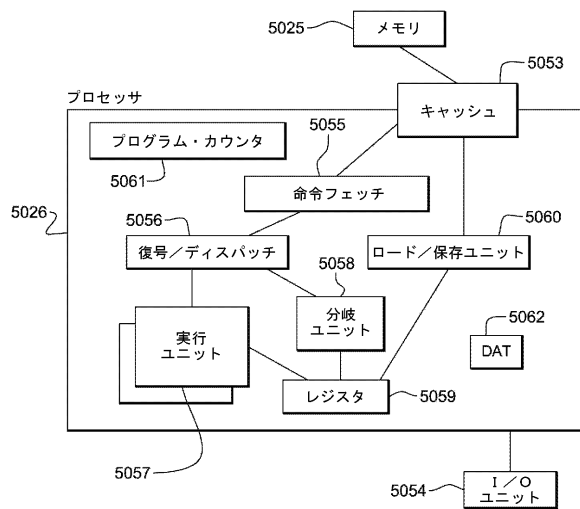
【図 17】



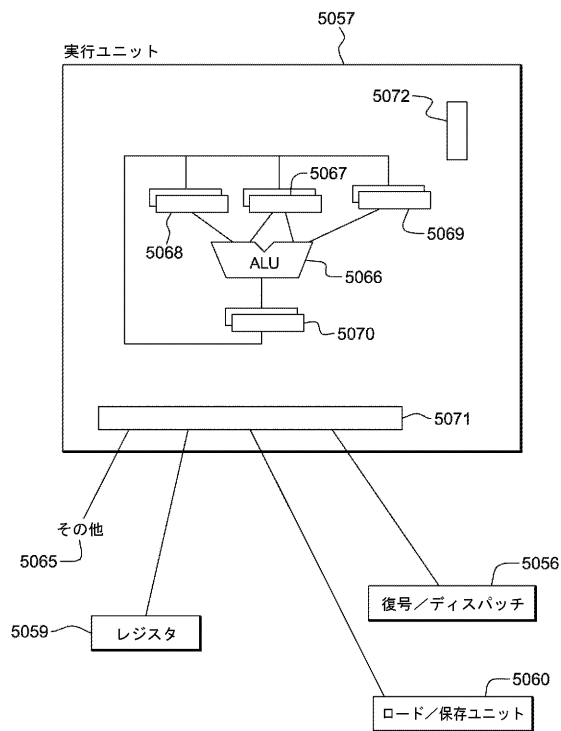
【図 18】



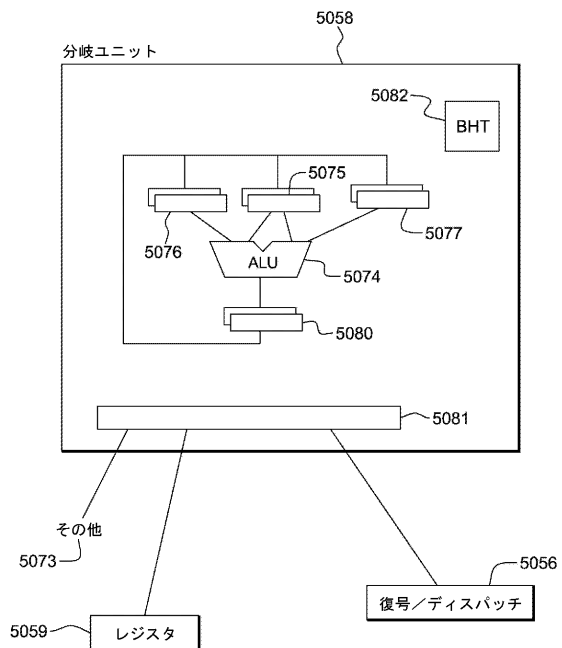
【図 19】



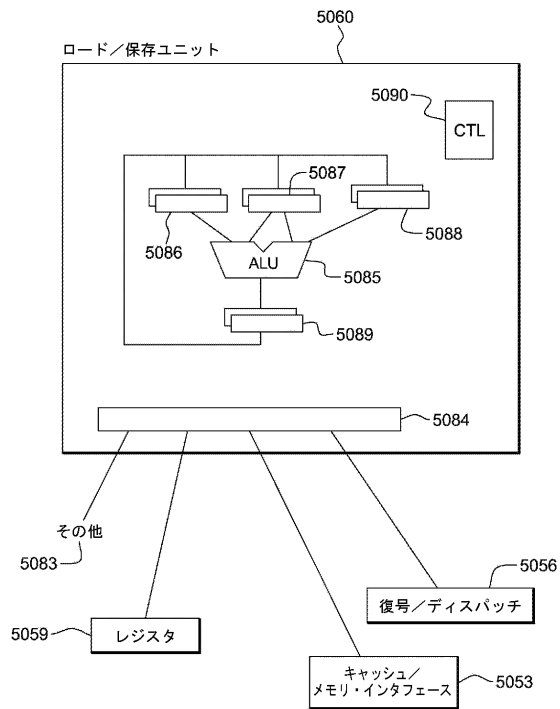
【図 20】



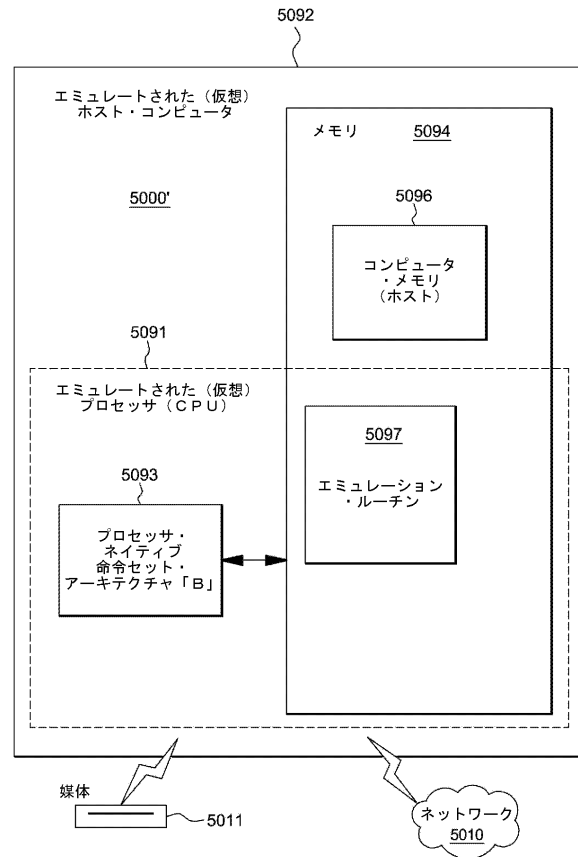
【図 21】



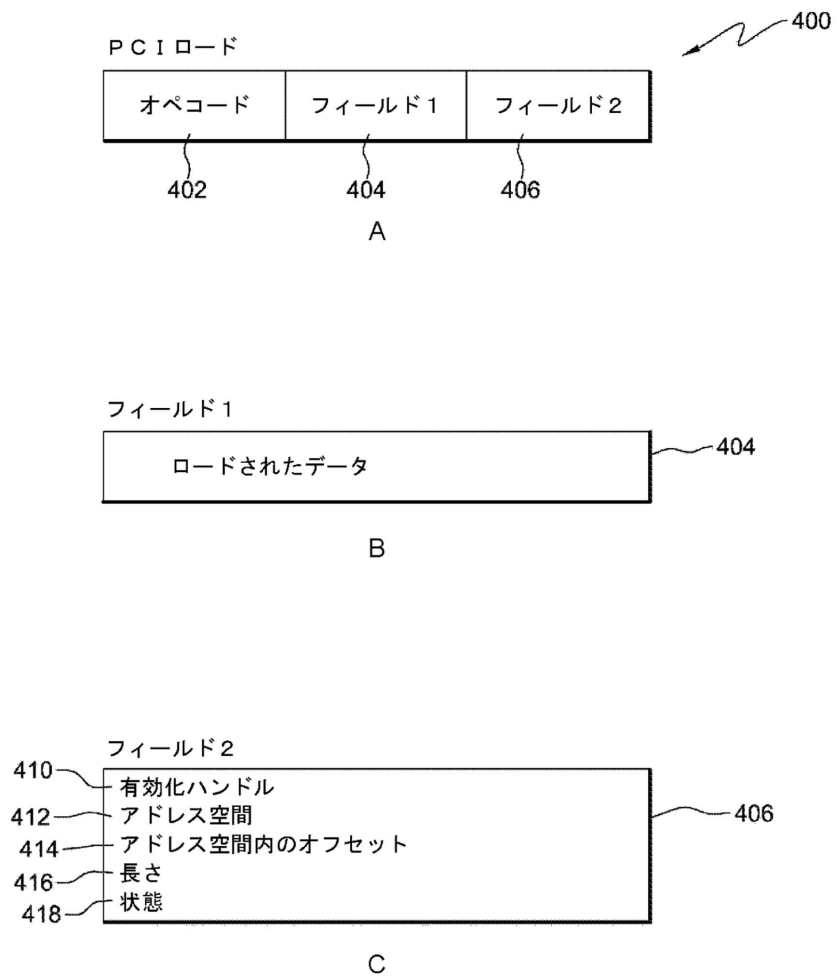
【図 22】



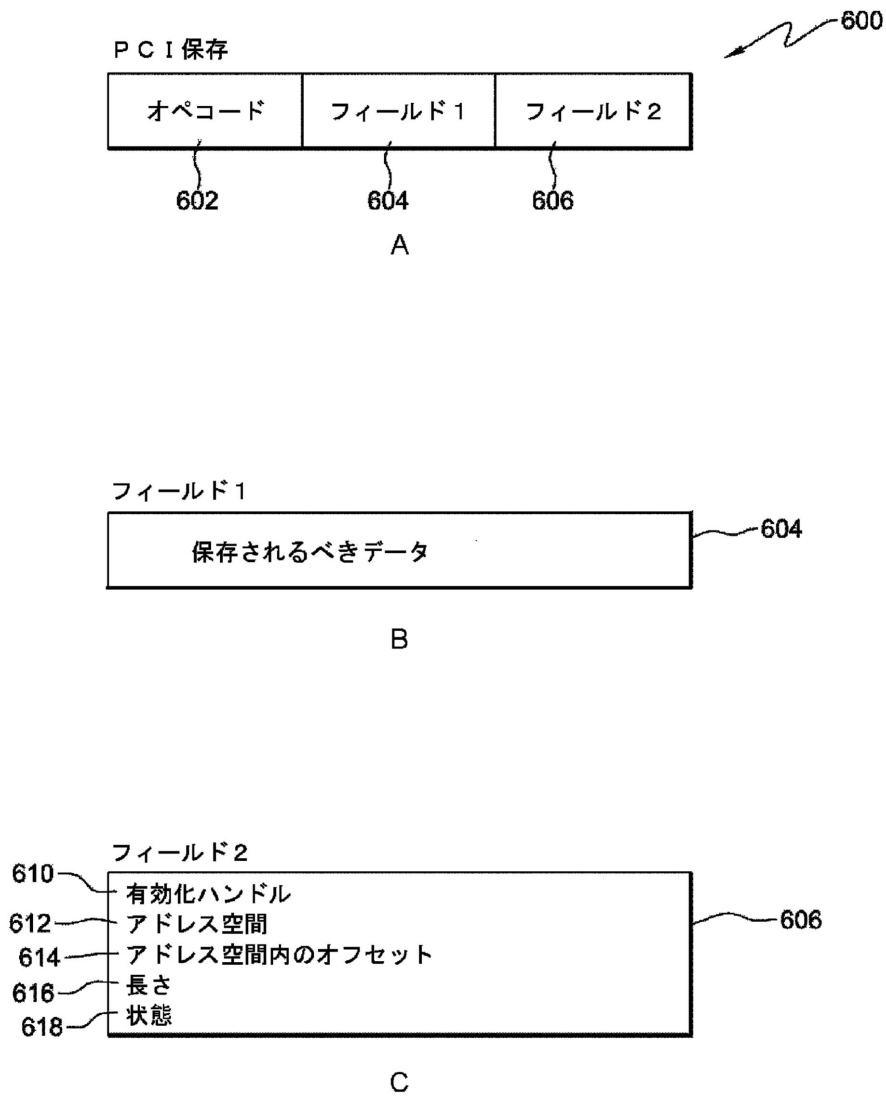
【図 23】



【図 6】



【図 9】



フロントページの続き

- (72)発明者 グレイナー、ダン
アメリカ合衆国 9 5 1 4 1 - 1 0 0 3 カリフォルニア州 サンノゼ サンタ・テレザ・ラブ ベ
イリー・アベニュー 5 5 5 エム/ピー エスブイエル/0 9 0 /エフ 3 7 4
- (72)発明者 クラドック、デイビッド
アメリカ合衆国 1 2 6 0 1 - 5 4 0 0 ニューヨーク州 ボキブシー サウス・ロード 2 4 5 5
エム/ピー ピー 3 1 8
- (72)発明者 グレッグ、トーマス
アメリカ合衆国 1 2 6 0 1 - 5 4 0 0 ニューヨーク州 ボキブシー サウス・ロード 2 4 5 5
エム/ピー ピー 3 1 4
- (72)発明者 ファレル、マーク
アメリカ合衆国 1 2 6 0 1 - 5 4 0 0 ニューヨーク州 ボキブシー サウス・ロード 2 4 5 5
エム/ピー ピー 3 1 0

審査官 三坂 敏夫

- (56)参考文献 特開平 0 1 - 1 5 6 8 5 6 (J P , A)
特開平 0 6 - 0 1 9 8 2 7 (J P , A)
特開 2 0 0 9 - 2 0 5 3 6 6 (J P , A)
米国特許第 0 6 7 0 4 8 3 1 (U S , B 1)

- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 F 9 / 3 0