

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5649200号  
(P5649200)

(45) 発行日 平成27年1月7日(2015.1.7)

(24) 登録日 平成26年11月21日(2014.11.21)

(51) Int. Cl.	F 1		
<b>G 0 6 F</b> 13/24	<b>(2006.01)</b>	G 0 6 F	13/24 3 1 0 D
<b>G 0 6 F</b> 9/48	<b>(2006.01)</b>	G 0 6 F	9/46 3 1 1 Z
<b>G 0 6 F</b> 9/46	<b>(2006.01)</b>	G 0 6 F	9/46 3 5 0

請求項の数 12 (全 49 頁)

(21) 出願番号	特願2013-515718 (P2013-515718)	(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2013-536485 (P2013-536485A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(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/EP2010/067020	(74) 代理人	100108501
(87) 国際公開番号	W02011/160705		弁理士 上野 剛史
(87) 国際公開日	平成23年12月29日 (2011.12.29)	(74) 代理人	100112690
審査請求日	平成25年8月5日 (2013.8.5)		弁理士 太佐 種一
(31) 優先権主張番号	12/821,178	(74) 代理人	100091568
(32) 優先日	平成22年6月23日 (2010.6.23)		弁理士 市位 嘉宏
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 アダプタ割り込みのソース・タイプの識別

(57) 【特許請求の範囲】

【請求項 1】

コンピュータ環境における割り込み処理を容易にする方法であって、

1つ又は複数のアダプタによる1つ又は複数の割り込み要求にตอบสนองして、割り込みコードを設定するステップであって、前記割り込みコードは、複数の異なるアダプタ・タイプの各々のアダプタ・タイプに関するアダプタ割り込みソース・マスク(AISM)、及び前記割り込みを要求しているアダプタの割り込みサブクラスを含み、前記アダプタ割り込みソース・マスクは前記複数の異なるアダプタ・タイプの各々のアダプタ・タイプについてのソース・インジケータを有し、前記設定するステップは、割り込みを要求するアダプタのアダプタ・タイプを判断するステップと、前記判断されたアダプタ・タイプに対応するアダプタ割り込みソース・マスクに前記ソース・インジケータを設定するステップとを含む、前記設定するステップと、

前記割り込みコードを前記オペレーティング・システムに提示して、前記1つ又は複数の割り込み要求を処理するステップと

を含む、前記方法。

【請求項 2】

前記複数の異なるアダプタ・タイプは、周辺装置相互接続(PCI)アダプタ・タイプ及び待ち行列型ダイレクトI/O(QDIO)アダプタ・タイプを含む、請求項1に記載の方法。

【請求項 3】

設定されたソース・インジケータは、前記設定されたソース・インジケータにより示される前記アダプタ・タイプの1つ又は複数のアダプタが割り込みを要求したことを前記オペレーティング・システムに示す、請求項1に記載の方法。

【請求項4】

前記設定されたソース・インジケータに基づいて、前記割り込みを処理するために用いられる1つ又は複数の割り込みイベント・インジケータを判断するステップ

をさらに含む、請求項3に記載の方法。

【請求項5】

前記設定されたソース・インジケータにより示される前記アダプタ・タイプが1つのアダプタ・タイプであることに応答して、そのアダプタ・タイプに対応する1つ又は複数の割り込みイベント・インジケータが調べられ、前記アダプタ・タイプに関するものではない1つ又は複数の割り込みイベント・インジケータは調べられない、請求項4に記載の方法。

10

【請求項6】

前記設定されたソース・インジケータは周辺装置相互接続型アダプタに対応し、前記1つ又は複数の割り込みイベント・インジケータは、1つ又は複数のアダプタ割り込みビット・ベクトルの1つ又は複数のアダプタ割り込みビット及び1つ又は複数のアダプタ割り込みサマリ・ビットのうち少なくとも一方を含む、請求項4に記載の方法。

【請求項7】

前記オペレーティング・システムはゲスト・オペレーティング・システムであり、前記設定するステップは、

ゲスト割り込み状態領域から前記アダプタ・タイプを取得するステップと、

前記取得したアダプタ・タイプに対応する前記A I S M内のソース・インジケータを設定するステップと

を含み、

前記ゲスト割り込み状態領域は、前記ゲスト・オペレーティング・システムに対して割り込みが保留状態にされるホスト制御ブロックを含む、

請求項1に記載の方法。

20

【請求項8】

前記取得するステップは、

前記ゲスト割り込み状態領域内のアダプタ割り込みソース・マスクから前記アダプタ・タイプを取得するステップ

を含み、

前記アダプタ割り込みソース・マスクは、前記割り込みが提示されるゲスト割り込みサブクラスに対応する、

請求項7に記載の方法。

30

【請求項9】

前記割り込みコードがアダプタ割り込みインジケータをさらに含む、前記アダプタ割り込みインジケータは、アダプタ割り込みが、前記アダプタに関連付けられた割り込みサブクラスに関して要求されていることを示す、請求項1に記載の方法。

40

【請求項10】

前記1つ又は複数のアダプタの1つのアダプタにより、割り込み要求を受信するステップ

をさらに含む、

前記アダプタは周辺装置相互接続アダプタを含み、かつ、前記アダプタを識別するこれと関連した機能ハンドルを有する、

請求項1に記載の方法。

【請求項11】

システムであって、請求項1～10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、前記システム。

50

## 【請求項 1 2】

コンピュータ・プログラムであって、コンピュータ・システムに請求項 1 ~ 1 0 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、前記コンピュータ・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ける割り込み処理に関し、具体的には、複数のアダプタ・タイプをサポートする環境におけるアダプタ割り込みの処理を容易にすることに関する。

## 【背景技術】

## 【0002】

コンピューティング環境は、種々のタイプのアダプタを含む、1つ又は複数のタイプの入力/出力デバイスを含むことができる。アダプタのタイプの例として、周辺装置相互接続(Peripheral Component Interconnect、P C I)又は周辺装置相互接続エクスプレス(Peripheral Component Interconnect express、P C I e)アダプタ、待機型直接 I/O (queued direct I/O、Q D I O) アダプタ、暗号化アダプタ等が挙げられる。アダプタは、特定のイベント又は状態をもたらすために割り込みを要求することができる。この割り込みの処理は、アダプタのタイプによって決まる。

## 【0003】

複数のアダプタ・タイプをサポートする環境において、アダプタ割り込みは、単一のイベントが生じたこと、又は異なるタイプの複数のアダプタに対して複数のイベントが生じたことを意味し得る。従って、所定の割り込みについて、全てのアダプタ・タイプの全てのアダプタ・イベント・インジケータを調べて、どのイベントが生じたかを判断しなければならない。

## 【0004】

Armstrong 他による「Virtualization Of a Global Interrupt Queue」という名称の2009年6月9日発行の特許文献1において、論理的に区分化されたシステムにおける仮想割り込みを処理するための方法、システム、及び製造物品が提供される。論理パーティションにおいて実行されている複数の仮想プロセッサと関連付けることができるインテリジェント仮想グローバル割り込みキュー(仮想GIQ)を用いることができる。仮想割り込みを受信すると、仮想GIQは、関連した仮想プロセッサの動作状態を調べることができる。仮想割り込みができるだけ早く処理されることを確実にするために、仮想GIQは、仮想割り込みの処理に最も適した動作状態にあると判断される関連した仮想プロセッサの1つに対して、仮想割り込みを提示することができる。

## 【0005】

Belmar 他による「Managing Input/Output Interruptions in Non-Dedicated Interruption Hardware Environments」という名称の2008年9月25日発行の特許文献2は、割り込みを与えるために、ゲストごとの専用割り込みハードウェアを使用しない単一の論理環境において管理される入力/出力割り込みについて記載する。環境内のディスパッチ可能ゲスト・プログラムは、ハイパーバイザの介入なしに直接I/O割り込みを受け取る。このことは、メモリ内に格納され、各ゲスト・プロセッサと関連した1つ又は複数の割り込み制御によって容易にされる。現在ディスパッチ可能ではないゲスト・プログラムについては、割り込みをゲストに対してポストし、ハイパーバイザへの通知を集めることができる。次に、ハイパーバイザは、単一の呼び出しにおいて複数のゲストに関する複数の通知を処理する。

## 【0006】

Easton 他による「Virtualization of Infiniband Host Channel Adapter Interruptions」という名

10

20

30

40

50

称の、2007年11月22日発行の特許文献3は、2つのレベルのサーバ仮想化を提供するための方法、システム、プログラム製品、及びコンピュータ・データ構造体を記載する。第1のハイパーバイザは、複数の論理パーティションが1組のリソースを共有できるようにし、第1のレベルの仮想化をもたらす。第2のハイパーバイザは、複数の独立した仮想マシンが、単一の論理パーティションに割り当てられたリソースを共有できるようにし、第2のレベルの仮想化をもたらす。単一の論理パーティションについて共有リソースからイベント通知を受信するために、その単一の論理パーティション内の全ての仮想マシンに関する全てのイベントが、単一のパーティション所有のイベント・キューにグループ化される。マシンにより、グループ化されたイベントを多重分離して、パーティション所有イベント・キューから、仮想マシン毎のベースで割り当てられた個々の仮想化イベント・キューにするために、パーティション所有イベント・キューからのグループ化されたイベントに対して、割り込み要求が信号で送られる。

10

【先行技術文献】

【特許文献】

【0007】

【特許文献1】米国特許第7,546,406号明細書

【特許文献2】米国特許公開第2008/0235425 A1号明細書

【特許文献3】米国特許公開第2007/0271559 A1号明細書

【特許文献4】米国特許第7,380,041号明細書

【特許文献5】米国特許第5551013号明細書

20

【特許文献6】米国特許第6009261号明細書

【特許文献7】米国特許第5574873号明細書

【特許文献8】米国特許第6308255号明細書

【特許文献9】米国特許第6463582号明細書

【特許文献10】米国特許第5790825号明細書

【非特許文献】

【0008】

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

【非特許文献2】「z/VM: Running Guest Operating Systems」、IBM出版物、No. SC24-5997-02、2001年10月

30

【非特許文献3】「z/VM: General Information Manual」、IBM出版物、No. GC24-5991-05、2003年5月

【発明の概要】

【発明が解決しようとする課題】

【0009】

コンピュータ環境における割り込み処理を容易にする方法を提供する。

【課題を解決するための手段】

【0010】

本発明の態様によると、調べられる特定のタイプのアダプタ・イベント・インジケータのみをイネーブルにする能力が提供される。

40

【0011】

コンピューティング環境における割り込み処理を容易にするための、請求項1に記載の方法及び対応するコンピュータ・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つ又は複数の態様が、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。本発明の前記及び他の目的、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。

【図面の簡単な説明】

50

## 【 0 0 1 3 】

【図 1】本発明の 1 つ又は複数の態様を組み込み、用いるためのコンピューティング環境の一実施形態を示す。

【図 2】本発明の一態様による、ホストが 1 つ又は複数のゲストを実行している中央演算処理コンプレックスの一実施形態を示す。

【図 3】本発明の一態様による、図 1 のシステム・メモリ及び I / O ハブの更なる詳細の一実施形態を示す。

【図 4】本発明の一態様による、図 1 のシステム・メモリ及び I / O ハブの更なる詳細の別の実施形態を示す。

【図 5】本発明の一態様に従って用いられるゲスト・アダプタ割り込みテーブル ( G A I T ) のエントリの一実施形態を示す。

10

【図 6】本発明の一態様に従って用いられるゲスト割り込み状態領域 ( G I S A ) の一実施形態を示す。

【図 7】本発明の一態様に従って用いられるアダプタ割り込み転送テーブル ( A I F T ) のエントリの一実施形態を示す。

【図 8】本発明の一態様に従って用いられる I / O 割り込みコードの一例を示す。

【図 9】本発明の一態様による、オペレーティング・システムに提示されるアダプタ割り込みの処理と関連した論理の一実施形態を示す。

【図 10】本発明の一態様による、ゲスト・オペレーティング・システムに提示されるアダプタ割り込みの処理と関連した論理の一実施形態を示す。

20

【図 11】本発明の一態様に従って用いられる割り込み制御設定 ( Set Interruption Controls ) 命令の一実施形態を示す。

【図 12】本発明の一態様による、図 11 の割り込み制御設定命令により用いられるフィールドのコンテンツの例を示す。

【図 13】本発明の一態様による、図 11 の割り込み制御設定命令により用いられるフィールドのコンテンツの例を示す。

【図 14】本発明の一態様による、図 11 の割り込み制御設定命令により用いられるフィールドのコンテンツの例を示す。

【図 15】本発明の一態様に従って用いられるアダプタ割り込みパラメータ・ブロック ( A I P B ) の一例を示す。

30

【図 16】本発明の一態様に従って用いられる P C I 機能制御修正 ( Modify PCI Function Controls ) 命令の一実施形態を示す。

【図 17】本発明の一態様による、図 16 の P C I 機能制御修正命令により用いられるフィールドの一実施形態を示す。

【図 18】本発明の一態様による、図 16 の P C I 機能制御修正命令により用いられる別のフィールドの一実施形態を示す。

【図 19】本発明の一態様に従って用いられる機能情報ブロック ( F I B ) のコンテンツの一実施形態を示す。

【図 20】本発明の一態様による、 P C I 機能制御修正命令の論理の概要の一実施形態を示す。

40

【図 21】本発明の一態様による、 P C I 機能制御修正命令により指定することができるアダプタ割り込み登録操作 ( register adapter interruptions operation ) と関連付けられた論理の一実施形態を示す。

【図 22】本発明の一態様による、 P C I 機能制御修正命令により指定することができるアダプタ割り込み登録解除操作 ( unregister adapter interruptions operation ) と関連付けられた論理の一実施形態を示す。

【図 23】本発明の 1 つ又は複数の態様を組み込む、コンピュータ・プログラム製品の一実施形態を示す。

【図 24】本発明の 1 つ又は複数の態様を組み込み、用いるためのホスト・コンピュータ・システムの一実施形態を示す。

50

【図25】本発明の1つ又は複数の態様を組み込み、用いるためのコンピュータ・システムの更に別の例を示す。

【図26】本発明の1つ又は複数の態様を組み込み、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27】本発明の1つ又は複数の態様を組み込み、用いるためのコンピュータ・システムの種々の要素の一実施形態を示す。

【図28】本発明の1つ又は複数の態様を組み込み、用いるための、図27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29】本発明の1つ又は複数の態様を組み込み、用いるための、図27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30】本発明の1つ又は複数の態様を組み込み、用いるための、図27のコンピュータ・システムのロード/ストア・ユニットの一実施形態を示す。

【図31】本発明の1つ又は複数の態様を組み込み、用いるためのエミュレートされたホスト・コンピュータ・システムの一実施形態を示す。

【発明を実施するための形態】

【0014】

本発明の態様によると、その割り込みの処理を容易にするために、割り込みを要求している1つ又は複数のアダプタ・タイプの識別を可能にする能力が提供される。例えば、アダプタ割り込みがオペレーティング・システムに提示されると、割り込みを要求しているアダプタ・タイプに関する情報が利用可能となる。アダプタ・タイプの知識を用いることにより、処理を各アダプタ・タイプに合わせて調整することができ、かつ、識別されないアダプタ・タイプを無視することができる。例えば、割り込み処理が、特定のインジケータを調べることを含む場合、識別されたタイプに対応するインジケータのみを調べる必要がある。このことにより、処理サイクルが低減され、性能が高まる。

【0015】

割り込みを処理するためのオペレーティング・システムは、ホスト、又はゲストではないオペレーティング・システムによって実行されるゲスト・オペレーティング・システム（例えば、ページング可能ストレージ・モード・ゲスト）とすることができる。本明細書で用いられるオペレーティング・システム（又は、他のソフトウェア）は、デバイス・ドライバを含む。

【0016】

一例において、z/Architecture（登録商標）においては、ページング可能ゲストが、レベル2の解釈において、解釈実行開始（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。

【0017】

さらに、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、暗号化アダプタ、PCIアダプタ、他のタイプの入力/出力アダプタ等）を含む。一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つ又は複数の態様は、アダプタが1つのアダプタ機能を含もうと、複数のアダプタ機能を含もうと適用可能である。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、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。

【0018】

本発明の1つ又は複数の態様を組み込み、用いるためのコンピューティング環境の一実

10

20

30

40

50

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、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・マシーンス・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）、及び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・マシーンス・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・マシーンス・コーポレーション又は他の会社の登録商標、商標、又は製品名であり得る。

10

**【0019】**

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つ又は複数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又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することができず、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可能な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又は絶対

20

**【0020】**

（必要に応じて変換された）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・メモリへのアクセスに対する仲裁を行うため、及び、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、CPU102から受信した要求、及び、1つ又は複数のアダプタ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。

**【0021】**

一例において、アダプタ110は、1つ又は複数のPCI機能を含む、周辺装置相互接続（PCI）又はPCI Express（PCIe）アダプタである。PCI機能は要求を発行し、要求は、1つ又は複数のスイッチ（例えばPCIeスイッチ）114を介して、入力/出力ハブ112（例えばPCIハブ）に送られる。一例において、入力/出力ハブは、1つ又は複数の状態マシンを含む、ハードウェアを含む。

30

**【0022】**

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、例として、ダイレクト・メモリ・アクセス（DMA）を実施し、又はメッセージ信号割り込み（MSI）を要求するのに用いられる入力/出力アドレスを含む。このアドレスは、DMA又はMSI要求のために用いられる情報にアクセスするアドレス変換及び保護ユニット118に与えられる。

40

**【0023】**

DMA操作の場合、アドレス変換及び保護ユニット118は、アドレスを、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に変換することができる。次いで、変換されたアドレスを含む、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を有する要求をシステム・メモリに転送する。

**【0024】**

MSI要求の場合は、MSI要求のI/Oアダプタ・イベント通知への変換を容易にす

50

るために、アドレス変換及び保護ユニット 118 内の情報が取得される。

【0025】

更に別の実施形態において、1つ又は複数の中央演算処理装置 102 に加えて又はその代わりに、図 2 に示されるもののような中央演算処理コンプレックスが、メモリ・コントローラ 106 に結合される。この特定の例においては、中央演算処理コンプレックス 150 は、仮想マシン・サポートを提供する。中央演算処理コンプレックス 150 は、例えば、1つ又は複数の仮想マシン 152 と、1つ又は複数の中央プロセッサ 154 と、少なくとも 1つのハイパーバイザ 156 とを含み、その各々が以下で説明される。

【0026】

中央演算処理コンプレックスの仮想マシン・サポートは、各々が `z / Linux` などのゲスト・オペレーティング・システム 158 をホストすることができる多数の仮想マシンを動作させる能力を提供する。各々の仮想マシン 152 は、別個のシステムとして機能することができる。つまり、各々の仮想マシンを、独立してリセットし、ゲスト・オペレーティング・システムをホストし、異なるプログラムで動作することができる。仮想マシン内で実行されるオペレーティング・システム又はアプリケーション・プログラムは、十分かつ完全なシステムにアクセスできるように見えるが、実際は、その一部しか使用可能でない。

【0027】

この特定の例では、仮想マシンのモデルは  $V = V$  モデルであり、仮想マシンのメモリは、実メモリではなく仮想メモリによってバックアップされる。各々の仮想マシンは、仮想線形メモリ空間を有する。物理リソースは、 $VM$  ハイパーバイザのようなハイパーバイザ 156 に所有され、共用物理リソースは、それぞれの処理要求に応じるために、必要に応じて、ゲスト・オペレーティング・システムにディスパッチされる。この  $V = V$  仮想マシン・モデルは、典型的には、多数のゲストにより、ハイパーバイザがハードウェア・リソースを単に区分化し、これを構成されたゲストに割り当てることが妨げられるので、ゲスト・オペレーティング・システムと物理的な共用マシン・リソースとの間の対話が  $VM$  ハイパーバイザによって制御されると仮定する。 $V = V$  モデルの 1つ又は複数の態様は、その全体が引用により本明細書に組み入れられる非特許文献 2 にさらに記載されている。

【0028】

中央プロセッサ 154 は、仮想マシンに割り当てることができる物理プロセッサ・リソースである。例えば、仮想マシン 152 は、1つ又は複数の論理プロセッサを含み、その各々は、仮想マシンに動的に割り当てることができる物理プロセッサ・リソース 154 の全て又は割り当て分 (share) を表す。仮想マシン 152 は、ハイパーバイザ 156 によって管理される。例として、ハイパーバイザは、プロセッサ 154 上で実行されるマイクロコードで実装するか、又はマシン上で実行されるホスト・オペレーティング・システムの一部とすることができる。一例において、ハイパーバイザ 156 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・マシーンス・コーポレーションによって提供される `z / VM` (登録商標) などの  $VM$  ハイパーバイザである。`z / VM` (登録商標) の一実施形態は、その全体が引用により本明細書に組み入れられる非特許文献 3 に記載されている。

【0029】

図 1 及び図 2 を参照すると、1つ又は複数のアダプタは、メッセージ信号割り込み (message signaled interruptions、`MSI`) を発行することができる。これらの割り込みは、1つ又は複数のオペレーティング・システムへの `I / O` アダプタ・イベント通知に変換され、そこで、インジケータが設定され、1つ又は複数の割り込みが要求される。図 3 - 図 4 を参照して説明されるように、この処理を容易にするために、`I / O` ハブ及びメモリ内の種々のデータ構造体が用いられる。

【0030】

特に、図 3 は、ゲストではないオペレーティング・システムへのアダプタ・イベント通知を提示するために用いられる構造体の一実施形態を示し、図 4 は、ゲストへのアダプタ

10

20

30

40

50

・イベント通知を提示するために用いられる構造体の一実施形態を示す。これらの図においては、メモリ・コントローラは図示されないが、これを用いてもよい。I/Oハブは、直接的に又はメモリ・コントローラを介して、システム・メモリ104及び/又はプロセッサ254に結合することができる。

#### 【0031】

図3を参照すると、一例において、システム・メモリ104は、割り込み処理を容易にするのに使用可能な1つ又は複数のデータ構造体を含む。この例において、システム・メモリ104は、特定のアダプタと関連した、アダプタ割り込みビット・ベクトル(AIBV)200及び随意的なアダプタ割り込みサマリ・ビット(AISB)202を含む。各アダプタについて、1つのAIBV及び対応するAISBが存在し得る。

10

#### 【0032】

一例において、アダプタ割り込みビット・ベクトル200は、アダプタ(例えば、PCI機能)と関連した、主ストレージ内の1つ又は複数のインジケータ(例えば、ビット)の一次元アレイである。アダプタ割り込みビット・ベクトル内のビットは、MSIベクトル数を表す。AIBVにおいて1に設定されたビットは、関連したアダプタについての条件、又はイベント・タイプを示す。PCI機能の例では、関連したAIBV内の各ビットは、MSIベクトルに対応する。従って、PCI機能が1つのMSIベクトルのみをサポートしている場合、そのAIBVは単一ビットを含み、PCI機能が複数のMSIベクトルをサポートしている場合、そのAIBVは、MSIベクトル毎に1ビットを含む。図3に示される例では、PCI機能は複数のMSIベクトルをサポートし(例えば、3つ)、

20

#### 【0033】

1つの特定の例では、命令(例えば、PCI機能制御修正命令)を用いて、PCI機能についてのAIBVを指定する。具体的には、命令は、オペレーティング・システムにより発行され、PCI機能の識別情報(identity)、AIBVを含む領域の主ストレージ位置、その位置からAIBVの第1のビットまでのオフセット、及びAIBVを構成するビット数を指定する。AIBVは、任意のバイト境界及び任意のビット境界上に割り当てることができる。これにより、複数のアダプタのAIBVを連続したビット及びバイト範囲に詰め込む、オペレーティング・システムの融通性が可能になる。

30

#### 【0034】

一例において、PCI機能の識別情報は、機能ハンドルである。機能ハンドルは、例えば、PCI機能ハンドルがイネーブルにされているかどうかを示すイネーブル・インジケータ、機能を識別するPCI機能番号(これは、静的識別子であり、特定のエントリを探索するための機能テーブル内への索引付けとして用いることができる)、及びこの機能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を含む。例えば、機能ハンドルがイネーブルにされる度に、インスタンス番号がインクリメントされて、新しいインスタンス番号を提供する。機能ハンドルは、1つ又は複数のエントリを含む機能テーブル内の機能

テーブル・エントリを見つけ出すために用いられる。例えば、機能ハンドルの1つ又は複数のビットを機能テーブルへの索引として使用して、特定の機能テーブル・エントリを見つけ出す。機能テーブル・エントリは、その関連したPCI機能に関する情報を含む。例えば、機能テーブル・エントリは、関連したアダプタ機能の状態に関する種々のインジケータを含むことができ、かつ、このアダプタ機能についてのデバイス・テーブル・エントリを見つけ出すのに用いられる1つ又は複数のデバイス・テーブル・エントリ索引を含むことができる。デバイス・テーブル・エントリは、それぞれのアダプタ機能(例えば、アドレス変換、割り込み処理)のための特定のサービスを提供するために用いられる情報を含む。(一実施形態においては、オペレーティング・システムに対して、ハンドルは、単にアダプタの内部識別子(opaque identifier)である。)

40

50

## 【 0 0 3 5 】

A I B Vに加えて、この例では、アダプタと関連した単一のインジケータ（例えば、ビット）を含む、アダプタについてのA I S B 2 0 2が存在する。1であるA I S Bは、A I S Bと関連したA I B Vにおいて、1つ又は複数のビットが1に設定されたことを示す。A I S Bは随意的なものであり、各アダプタについて1つ、選択されたアダプタの各々について1つ、又はアダプタのグループについて1つあってもよい。

## 【 0 0 3 6 】

P C I機能についての1つの特定の実装においては、命令（例えば、P C I機能制御修正命令）を用いて、P C I機能についてのA I S Bを指定する。具体的には、命令は、オペレーティング・システムにより発行され、かつ、P C I機能（例えば、ハンドル）の識別情報、A I S Bを含む領域の主ストレージ位置、その位置からA I S Bまでのオフセット、及び、サマリ・ビットが存在することを示すアダプタ割り込みサマリ通知イネーブルメント制御を指定する。A I S Bは、任意のバイト境界及び任意のビット境界上に割り当てることができる。これにより、複数のアダプタのA I S Bを連続したビット及びバイト範囲に詰め込む、オペレーティング・システムの融通性が可能になる。

## 【 0 0 3 7 】

オペレーティング・システムは、単一のA I S Bを複数のP C I機能に割り当てることもできる。これにより、複数のA I B Vが単一のサマリ・ビットと関連付けられる。従って、1であるそうしたA I S Bは、オペレーティング・システムが複数のA I B Vを走査しなければならないことを示す。

## 【 0 0 3 8 】

一例において、A I B V及びA I S Bは、I / Oハブ1 1 2内に配置されたデバイス・テーブル2 0 8のデバイス・テーブル・エントリ2 0 6内に位置するアドレスにより指し示される。一例において、デバイス・テーブル2 0 8は、I / Oハブのアドレス変換及び保護ユニット内に配置される。

## 【 0 0 3 9 】

デバイス・テーブル2 0 8は、その各々が特定のアダプタ機能2 1 0に割り当てられている1つ又は複数のエントリ2 0 6を含む。デバイス・テーブル・エントリ2 0 6は多数のフィールドを含み、これらのフィールドは、例えば上述の命令を用いてポピュレートすることができる。1つ又は複数のフィールドの値は、ポリシー及び/又は構成に基づくことができる。フィールドの例は、次のものを含むことができる。：すなわち、

割り込みサブクラス（I S C）2 1 4：割り込みについての割り込みサブクラスを示す。I S Cは、オペレーティング・システムが割り込みを処理する優先順位と関連付けることができる、マスク可能なアダプタ割り込みのクラスを識別する。

A I B Vアドレス（@）2 1 6：例えば、このデバイス・テーブル・エントリに割り当てられた特定のアダプタ機能についてのA I B Vを含むストレージ位置の開始部の絶対アドレスを提供する。

A I B Vオフセット2 1 8：A I B Vの開始部に対する主ストレージ位置へのオフセットである。

A I S Bアドレス（@）2 2 0：オペレーティング・システムがA I B Vを指定している場合に、例えば、このP C I機能についてのA I S Bを含むストレージ位置の開始部の絶対アドレスを提供する。

A I S Bオフセット2 2 2：A I S Bに対する主ストレージ位置へのオフセットである。

アダプタ割り込みサマリ通知イネーブルメント制御（イネーブル）2 2 4：この制御は、A I S Bが存在するかどうかを示す。

割り込みの数（number of interruptions、N O I）2 2 6：このP C I機能について許容されるM S Iベクトルの最大数を示し、ゼロは、許容されているものがないことを示す。

他の実施形態においては、D T Eは、より多くの、より少ない、又は異なる情報を含む

10

20

30

40

50

ことができる。

【0040】

一実施形態において、アダプタによる特定の割り込み要求に使用されるデバイス・テーブル・エントリが、例えば、アダプタ（例えばPCI機能210）により発行される要求内に配置されたリクエスト識別子（RID）（及び/又はアドレスの一部）を用いて見つけ出される。要求内には、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、及び機能番号を指定する16ビットの値）が含まれるだけでなく、割り込みに使用されるアドレスも含まれる。RID及びアドレスを含む要求は、例えばスイッチを介して、例えばコンテンツ・アドレス指定可能メモリ（CAM230）に与えられ、コンテンツ・アドレス可能メモリを用いて索引値が与えられる。例えば、CAM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CAMエントリはRIDの値を含む。例えば、受信したRIDがCAM内のエントリ内に収容された値と合致した場合、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・エントリを見つげ出す。つまり、CAMの出力を、デバイス・テーブル208への索引付けのために用いる。合致が存在しない場合、受信したパケットは破棄される。（他の実施形態においては、CAM又は他のルックアップは必要とされず、RIDが索引として用いられる。）見つかったDTEは、例えば、割り込み要求を処理するのに用いられる。

10

【0041】

1つの特定の例において、割り込み要求が、特定のゾーン又は論理パーティション内で実行しているゲスト（例えば、ページング可能ストレージ・モード・ゲスト、すなわちV=Vゲスト）に対するものである場合、デバイス・テーブル・エントリはまた、図4に示されるような、ゾーン・フィールド228を含む。このフィールドは、ゲストが属するゾーンを示す。別の実施形態においては、このフィールドが使用されないか、又はこのフィールドをゲストが提供されない状況においてさえも使用することができる（例えば、オペレーティング・システムが実行されているゾーン又は論理パーティションを指定するために）。

20

【0042】

ゲストに対する割り込み処理を容易にするために、他のデータ構造体が用いられ、その一部はホスト・メモリ270内に格納され、他のものはゲスト・メモリ271内にある。これらの構造体の例が以下で説明される。

30

【0043】

一例において、ホスト・メモリ270は、例えば、転送（forwarding）AISBアレイ272と、ゲスト・アダプタ割り込みテーブル（GAIT）274とを含む。転送AISBアレイ272は、MSI要求がゲストをターゲットにしているか、又はそのホストをターゲットにしているかを判断するために、ゲスト・アダプタ割り込みテーブルと共に用いられるAISBのアレイである。転送AISBアレイは、ホストがゲストに割り当て、かつ、ホストがゲストに代わってアダプタ・イベント通知割り込みを要求している、PCI機能のホストAISBを含む。このようなアレイは、ゲストのホスト（例えば、z/VM（登録商標））により、ホスト・ストレージにおいて割り当てられる。

【0044】

ゲスト・アダプタ割り込みテーブル274を転送AISBアレイと共に用いて、MSI要求がホストをターゲットにしているのか、又はそのゲストの1つをターゲットにしているのか、又はゲストをターゲットにしている場合にはどのゲストをターゲットにしているのかを判断する。転送AISBアレイ内のインジケータ（例えば、ビット）とGAITエントリとの間に1対1の対応が存在する。このことは、転送AISBアレイ内のビットが1に設定され、対応するGAIT情報が転送情報を含む場合に、AISBインジケータ（例えば、ビット）及び対応するGAITエントリと関連したゲストのためのアダプタに関して、アダプタ・イベント通知が保留状態にされることを意味する。

40

【0045】

GAITエントリが使用され、かつ、所定の値（例えば、全てゼロ）を含む場合、MS

50

I 要求のターゲットはホストである。G A I T エントリが使用され、かつ、所定の値を含まない場合は、M S I 要求のターゲットはゲストである。さらに、M S I 要求のターゲットがゲストである場合、G A I T エントリは、図 5 に示されるような以下の情報を含む。  
：すなわち、P C I 機能に関するホスト・アドレス及びゲスト A I S B のゲスト・オフセット 2 9 0、ゲスト割り込み状態領域 ( G I S A ) のホスト・アドレス 2 9 1、及び、ゲストのために生成されるためのアダプタ割り込みに関するゲスト割り込みサブクラス ( G I S C ) 2 9 2 である。

【 0 0 4 6 】

ゲスト割り込み状態領域 ( G I S A ) 2 7 6 に関する更なる詳細が、図 6 を参照して提供される。一例において、G I S A 2 7 6 は、ゲスト・アダプタ割り込みが保留状態にされる制御ブロックである。本発明の一態様によると、G I S A 2 7 6 は、例えば、ゲスト割り込みサブクラスごとに 1 ビットを有するマスクであり、サブクラスのための割り込みモードが単一の割り込みモード ( I S C に対して単一のアダプタ割り込み要求が提示される ) であるかどうかを示すために用いられる単一割り込みモード・マスク ( S I M M ) 2 7 7 ; ゲスト割り込みサブクラスごとに 1 ビットを有するマスクであり、I S C のための割り込みモードが非割り込みモード ( 更なる割り込みを提示することはできない ) であるかどうかを示すために用いられる非割り込みモード・マスク ( N I M M ) 2 7 9 ; その各々が、割り込みが I S C に関して保留状態であるかどうかを指定するために用いられる、複数の割り込みサブクラス ( I S C ) のためのインジケータを含む、ゲストと関連したマスクである割り込み保留マスク ( I P M ) ; ゲストに対応する別のマスクであり、各々のビットは、ホストが警告されるかどうかを示す、割り込み警告マスク ( I A M ) 2 8 3 ; これらの I S C のためのアダプタ割り込みのカウントを提供するアダプタ割り込みカウント 2 8 7 ; 及び各々がゲスト I S C に対応するアレイ内の各 A I S M に対応し、かつ、環境内に存在する ( 又は、存在し得る ) 各アダプタ・タイプについてのインジケータ ( 例えば、ビット ) を有する、アダプタ割り込みソース・マスク ( A I S M ) アレイ 2 8 9 を含む。

【 0 0 4 7 】

例として、G I S A の基点又はアドレスは、G A I T 2 7 4 ( 図 4 ) 内だけではなく、状態記述 2 8 0 内にも指定される。状態記述は、例えば、解釈ハードウェア / ファームウェアに対してゲストのための仮想 C P U を定めるホストが保持する制御ブロックである。ゲストごとに一意の G I S A が用いられ、一実施形態においては、ゲストごとに 1 つの G I S A しかない。従って、ゲストが複数の仮想 C P U を有するように定められた場合、ホストにより複数の状態記述が保持され、その各々は、同じ G I S A の基点又はアドレスを含む。

【 0 0 4 8 】

本明細書で用いられるファームウェアは、例えば、プロセッサのマイクロコード、ミクロコード、及び / 又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・レベルの命令及び / 又はデータ構造体を含む。一実施形態において、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、又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。

【 0 0 4 9 】

上記に加えて、ゲスト・メモリ ( ピンニングされ、すなわち固定され、ホスト・メモリ内でページング可能でなくされた ) 内には、ゲスト A I S B アレイ 2 8 2 及びゲスト A I B V アレイ 2 8 4 が存在する。ゲスト A I S B アレイ 2 8 2 は、複数のインジケータ 2 0 2 ' ( 例えば、A I S B ) を含み、その各々を I / O アダプタと関連付けることができる。I / O アダプタのための A I S B は、1 であるときには、I / O アダプタと関連したアダプタ割り込みビット・ベクトル ( A I B V ) において、1 つ又は複数のビットが 1 に設定されたことを示す。

10

20

30

40

50

## 【 0 0 5 0 】

A I B V アレイ 2 8 4 は、1 つ又は複数の A I B V 2 0 0 ' を含み、各々の A I B V 2 0 0 ' は、A I B V 2 0 0 を参照して上述されたように、I / O アダプタと関連した1 つ又は複数のインジケータ（例えば、ビット）の一次元のアレイである。A I B V 内の各ビットは、1 であるときには、関連した I / O アダプタに関するイベントの状態又はタイプを示す。

## 【 0 0 5 1 】

ホスト及びゲスト・メモリ内のデータ構造体に加えて、アダプタ割り込み転送テーブル（A I F T）2 8 5 と呼ばれるデータ構造体が、ホストによっても、ゲストによってもアクセスできない安全なメモリ 2 8 6 内に保持される。アダプタ割り込み転送テーブルは、M S I 要求が、ホスト及びゲストが実行されている論理パーティションをターゲットにしているかどうかを判断するために、システム・ファームウェアにより用いられる。A I F T は、P C I 機能が割り当てられる論理パーティションを識別するゾーン番号により索引付けされる。A I F T エントリが用いられ、エントリが所定の値（例えば、全てゼロ）を含む場合、アダプタ・イベント通知のターゲットは、指定された論理パーティション内で実行されているオペレーティング・システムである。A I F T エントリが用いられ、エントリが所定の値を含まない場合、ファームウェアは、転送 A I S B アレイ及び G A I T を用いて、アダプタ・イベント通知のターゲットが、論理パーティション内で実行されているホストであるか又はゲストであるかを判断する。一例において、図 7 に示されるように、A I F T 2 8 5 の A I F T エントリは、例を挙げれば、例えばパーティションの（ホストの）ストレージ内の転送 A I S B アレイ・アドレス 2 9 4 と、転送 A I S B アレイ及び G A I T エントリ内の G A I T のビット単位での長さ 2 9 5 と、パーティションのストレージ内の G A I T のアドレス 2 9 6 と、そのパーティションに関するゲストに転送される M S I 要求と関連したホスト割り込みサブクラス（I S C）2 9 7 とを含む。

## 【 0 0 5 2 】

図 3 及び / 又は図 4 に戻ると、割り込みを要求するために、アダプタ機能 2 1 0 は I / O ハブにパケットを送る。このパケットは、M S I アドレス 2 3 2 と、関連したデータ 2 3 4 とを有する。I / O ハブは、受信したアドレスの少なくとも一部を、M S I 比較レジスタ 2 5 0 の値と比較する。合致が存在する場合、D M A 操作ではなく、割り込み（例えば M S I）が要求されている。要求の理由（すなわち、発生したイベントのタイプ）は、関連データ 2 3 4 内に示される。例えば、データの下位ビットの 1 つ又は複数を用いて、理由（イベント）を示す特定の割り込みベクトル（すなわち、M S I ベクトル）を指定する。

## 【 0 0 5 3 】

アダプタから受信された割り込み要求が、I / O アダプタ・イベント通知に変換される。つまり、1 つ又は複数のインジケータ（例えば、1 つ又は複数の A I B V、及び随意的に A I S B）が設定され、1 つが既に保留中ではない場合、オペレーティング・システム（ホスト又はゲスト）への割り込みが要求される。一実施形態においては、1 つ又は複数のアダプタからの複数の割り込み要求（例えば、M S I）が融合されて、オペレーティング・システムに対する単一の割り込みになるが、それぞれ A I B V 及び A I S B の表示を有している。例えば、I / O ハブが既に M S I 要求を受信しており、プロセッサへの割り込み要求を提供しており、割り込みが依然として保留中である（例えば、どういうわけか、割り込みがオペレーティング・システムに提示されていない（例えば、割り込みがディスエーブルにされている））場合、ハブが 1 つ又は複数の他の M S I を受信した場合でも、ハブは付加的な割り込みを要求しない。1 つの割り込みが複数の M S I 要求に置き換わり、それらを表す。しかしながら、1 つ又は複数の A I B V、及び随意的に 1 つ又は複数の A I S B が、依然として設定される。

## 【 0 0 5 4 】

M S I（又は、他のアダプタ割り込み要求）を I / O アダプタ・イベント通知に変換することに関する更なる詳細が、以下に説明される。最初に、M S I を、ゲストではないオ

10

20

30

40

50

ペレーティング・システムに提示される I/Oアダプタ・イベント通知に変換することに関する詳細が説明される。その後、MSIを、ゲスト・オペレーティング・システムに提示される I/Oアダプタ・イベント通知に変換することに関する詳細が説明される。

**【0055】**

一例において、MSI要求を I/Oアダプタ・イベント通知に変換するために、特定の初期化が実施される。例えば、初期化の一部として、構成(configuration)内のPCI機能に関する判断がなされる。例として、リスト照会(QueryList)命令のような命令を用いて、要求中の構成に割り当てられた(例えば、特定のオペレーティング・システムに割り当てられた)PCI機能のリストを取得する。この情報は、この情報を保持する構成データ構造体から得られる。

10

**【0056】**

次に、リスト内のPCI機能の各々について、PCI機能に使用されるMSIアドレス、及び、PCI機能によりサポートされるMSIベクトルの数についての判断がなされる。MSIアドレスは、I/Oハブ、及びそれがインストールされるシステムの特性に基づいて判断される。サポートされるMSIベクトルの数は、ポリシーに基づいており、構成可能である。

**【0057】**

付加的に、各PCI機能について、AIBVが割り当てられ、さらに、もしあれば、AISBが割り当てられる。一例においては、オペレーティング・システムは、典型的にはアダプタのクラスに基づいた、1つ又は複数のアダプタの効率的な処理を可能にするように、AIBVの割り当てを決定する。AIBV及びAISBが割り当てられ、ゼロにクリアされ、アダプタ割り込み登録操作が指定される(例えば、PCI機能制御修正命令を用いて)。この操作は、AIBV、AISB、ISC、割り込みの数(MSIベクトル)、及びアダプタ割り込みサマリ通知イネーブルメント制御を登録する。これらのパラメータは、初期化が実行される、PCI機能に対応するデバイス・テーブル・エントリ内に格納される。次に、PCI機能の構成空間が書き込まれる。具体的には、MSIアドレス及びMSIベクトル・カウントが、以前の登録と一致するPCI機能の構成アドレス空間内に書き込まれる。(一例において、PCI機能は、構成空間、I/O空間、及び1つ又は複数のメモリ空間を含む、複数のアドレス空間を含む。)

20

**【0058】**

その後、動作中、PCI機能はMSIを生成し、これがアダプタ・イベント通知に変換される。例えば、動作中、PCI機能がMSIを生成したいと望むとき、PCI機能は、典型的には、条件を説明する何らかの情報を、オペレーティング・システムが利用できるようにする。これにより、PCI機能のMSI要求をオペレーティング・システムへのI/Oアダプタ・イベントに変換するために、1つ又は複数のステップがもたらされる。

30

**【0059】**

例えば、最初に、割り込みを要求するイベントの記述を記録する。つまり、PCI機能は、例えばシステム・メモリ内に格納されている1つ又は複数のアダプタ特有のイベント・記述・記録構造内に、イベントの記述を記録する。これは、イベントのタイプを記録すること、並びに、付加的な情報を記録することを含むことができる。付加的に、MSIアドレス及びMSIベクトル数、並びにリクエストIDを指定する要求が、PCI機能により開始される。この要求はI/Oハブにより受信され、要求の受信に回答して、要求内のリクエストIDを用いて、PCI機能についてのデバイス・テーブル・エントリを見つけ出す。I/Oハブは、要求内のアドレスの少なくとも一部を、MSI比較レジスタ内の値と比較する。それらが等しい場合、MSIアドレスが指定され、従って、MSIが要求される。

40

**【0060】**

その後、要求内で指定されたMSIベクトルの数が、この機能について許容される割り込み数(NOI)より少ないか又はそれと等しいかどうかについての判断がなされる。MSIベクトル数がNOIより大きい場合は、エラーが表示される。他の場合には、I/O

50

ハブは、ビット機能設定 (set bit function) を発行して、ストレージ内に適切な A I B V ビットを設定する。M S I ベクトル数を、デバイス・テーブル・エントリ内に指定された A I B V オフセットに加算し、このビットの数を、デバイス・テーブル・エントリ内で指定された A I B V アドレスから変位させることにより、適切なビットが求められる。さらに、A I S B が示される場合、I / O ハブはビット機能設定を用いて、デバイス・テーブル・エントリ内の A I S B アドレス及び A I S B オフセットを使用し、A I S B を設定する。

#### 【 0 0 6 1 】

次に、一実施形態において、割り込み要求が既に保留中であるかどうかについての判断がなされる (例えば、C P U 又は I / O ハブにより)。この判断を行うために、保留インジケータが使用される。例えば、割り込みを処理することができるコンピューティング環境のプロセッサがアクセス可能である、プロセッサ 2 5 4 のメモリ内に格納された保留インジケータ 2 5 2 (図 3、図 4) をチェックする。これが設定されていない場合は、設定する。既に設定されている場合、処理は完了し、別の割り込み要求は要求されない。従って、その後の割り込み要求は、既に保留中である 1 つの要求に包含される。

10

#### 【 0 0 6 2 】

1 つの特定の例においては、割り込みサブクラス毎に 1 つの保留インジケータがあってもよく、従って、要求している機能に割り当てられた割り込みサブクラスの保留インジケータが、チェックされるインジケータとなる。

#### 【 0 0 6 3 】

非同期的に、1 つ又は複数のプロセッサが保留インジケータをチェックする。特に、I S C (及び、別の実施形態においてはゾーン) に関してイネーブルにされた各々のプロセッサは、例えばそのプロセッサについて (すなわち、プロセッサのオペレーティング・システムについて) 割り込みがイネーブルにされたときに、インジケータ上でポーリングする。プロセッサの 1 つが、インジケータが設定されていると判断すると、同じ I S C (及び、別の実施形態においてはゾーン) に関してイネーブルにされている他のプロセッサとの仲裁を行って、割り込みを処理する。次に、割り込みを処理するためのプロセッサが、割り込みをオペレーティング・システムに提示する。

20

#### 【 0 0 6 4 】

オペレーティング・システムへの割り込みの提示に応答して、オペレーティング・システムは、この割り込み要求に対して、潜在的には他の割り込み要求に対して設定されたインジケータを処理する。I / O 割り込みコードは、オペレーティング・システム及びファームウェアの両方にアクセス可能な周知の記憶位置内に格納される。

30

#### 【 0 0 6 5 】

図 8 を参照すると、一例において、I / O 割り込みコード 3 0 0 が、例えば、次のものを含む。すなわち、：

アダプタ割り込みソース・マスク (A I S M) 3 0 2 : このフィールドは、複数のインジケータ (例えば、ビット) を有するマスクを含み、各々のインジケータは、設定されたとき (例えば、1 に)、1 つ又は複数のアダプタ・イベント通知を作成したアダプタ・タイプを識別する。

40

例えば、例を挙げると待ち行列型ダイレクト I / O (queued direct I/O、Q D I O) アダプタ (その例は、その全体が引用により本明細書に組み入れられる、Belmar 他による「Managing Input/Output Interruptions In Non-Dedicated Interruption Hardware Environments」という名称の 2 0 0 8 年 5 月 2 7 日発行の特許文献 4 に記載されている)、P C I アダプタ、アクセス・ポイント (A P) アダプタ、暗号化アダプタ、及び他のものを含む複数のアダプタ・タイプが、環境内に存在することができ、これらのアダプタの各々について、A I S M 内に 1 つのインジケータがある。そのアダプタ・タイプが割り込みを要求するとき、その対応するビットが設定される。

アダプタ割り込みインジケータ (A) 3 0 4 : このフィールドは、アダプタ割り込みが

50

、アダプタに関連付けられた割り込みサブクラスに関して要求されていることを示す。

割り込みサブクラス 306 : このフィールドは、割り込みを要求しているアダプタの割り込みサブクラスを示す。

#### 【0066】

一実施形態において、アダプタ割り込み要求にตอบสนองして、ファームウェアにより割り込みコードが設定される。例えば、アダプタ・タイプを一意に識別する、割り込みを発行する特有の I/O インフラストラクチャに基づいて、AISM が設定される。1つの特定の例においては、オペレーティング・システムが割り込みの処理に関してイネーブルにされたときに、割り込みコードを設定する（例えば、割り込みを最初に受信したときにファームウェアによりブロックされる制御内に格納された対応する情報から、1つ又は複数のフィールドが設定される）。特定の命令を介して I/O 割り込みコードを取り出すといった、他の例も可能である。

10

#### 【0067】

特定的には、一例において、AISM 内に適切なビットを設定するために、ファームウェアは、イベントを要求しているアダプタのタイプを判断する。例えば、ファームウェアは、指定されたインジケータの設定に基づいて、PCI アダプタ割り込み要求を認識する。同様に、QDIO アダプタ又は他のアダプタについては、特定のアダプタが設定されるか、又は例としてアダプタのタイプを示す情報が格納される。

#### 【0068】

割り込みコード、特定的には AISM を用いて、オペレーティング・システムは、割り込みを要求したアダプタのタイプを判断することができ、従って、その取り組みに合わせて調整することができる。例えば、割り込み要求を発行し、イベント・インジケータを設定させる複数のアダプタ・タイプが存在し得る。しかしながら、これらのイベント・インジケータ及び関連したイベント処理は、アダプタ・タイプによって異なる。例えば、PCI アダプタについては、インジケータは、例えば、ビットのインクリメントで割り当てられ、QDIO アダプタの場合、インジケータは、バイトのインクリメントで割り当てられ、AP キューの場合、インジケータは、構造のインクリメントで示される。多くの他の変形も可能である。さらに、こうしたインジケータの構造及び意味は各アダプタ・タイプ毎に異なるので、これらのイベント・インジケータを処理するプログラム（例えば、デバイス・ドライバ）は、典型的には、デバイス特有のものである。従って、アダプタのタイプを知ることにより、オペレーティング・システムは、その処理に合わせて調整することができる。このことは、図9を参照してさらに詳細に説明される。

20

30

#### 【0069】

図9を参照すると、最初に、オペレーティング・システムにアダプタ割り込みが提示される（ステップ402）。アダプタ割り込みの提示にตอบสนองして、本発明の態様に従って、オペレーティング・システムは、割り込みを要求したアダプタのタイプを判断する（ステップ404）。一例において、この判断は、オペレーティング・システム、特にアダプタ割り込みソース・マスクにアクセス可能な I/O 割り込みコードを調べることによってなされる。オペレーティング・システムは、マスクを調べ、どのビットが設定されているかを判断する。オペレーティング・システムは、どのアダプタ・タイプがマスク内の設定ビットに対応するかをさらに判断する。例えば、ビット1が設定された場合、オペレーティング・システムは、ビット1がPCI アダプタに対応することを（例えば、データ構造体 - テーブル、制御ブロック等から）判断する。

40

#### 【0070】

I/O 割り込みコードのチェックにより判断されたアダプタ・タイプに基づいて、オペレーティング・システムは、調べられるイベント・インジケータを識別する（ステップ406）。特に、オペレーティング・システムは、どのインジケータがこのタイプのアダプタと関連付けられているかを（例えば、格納されたデータ構造体 - 例え、データ構造体、制御ブロック等に基づいて）判断する。例えば、PCI アダプタの場合、オペレーティ

50

ング・システムは、データ構造体から、A I S B及びA I B Vインジケータがチェックされることを知る。オペレーティング・システムはまた、それらのインジケータの位置も知る。次に、オペレーティング・システムは、それらのタイプのインジケータだけを調べる（ステップ408）。オペレーティング・システムは、他のタイプのアダプタに関するインジケータは無視する。設定された（例えば、1に）アダプタ・タイプと関連したいずれのインジケータも処理される。

#### 【0071】

1つの特定の例において、オペレーティング・システムは、いずれかの設定されたA I S B及びA I B Vを処理する。例えば、オペレーティング・システムは、いずれかのA I S Bが設定されているかどうかをチェックする。設定されている場合、オペレーティング・システムは、A I S Bを用いて、1つ又は複数のA I B Vの位置を判断する。例えば、オペレーティング・システムは、A I S B及びA I B Vの位置を記憶している。さらに、オペレーティング・システムは、各々のA I S B及びA I B Vが表すアダプタも記憶している。従って、オペレーティング・システムは、A I S B及びA I B Vの位置と、A I S B、A I B V及びアダプタIDの間の関連付けを含むある形式の制御ブロック又は他のデータ構造体を保持することができる。オペレーティング・システムはこの制御ブロックを用いて、その関連したA I S Bに基づいたA I B Vの位置特定を容易にする。更に別の実施形態においては、A I S Bは使用されない。この状況では、制御ブロックを用いて、いずれかの特定のA I B Vを探索する。1つ又は複数のA I B Vを見つけ出すことに応答して、オペレーティング・システムは、A I B Vを走査し、設定されたあらゆるA I B Vを処理する。オペレーティング・システムは、提示されたイベントと一致する方法で割り込みを処理する（例えば、状態等を提供する）。例えば、ストレージ・アダプタの場合、イベントは、操作が完了したことを表示し得る。この結果、オペレーティング・システムはアダプタにより格納された状態をチェックして、操作が成功裡に完了したのかどうか、及び、操作の詳細を調べる。ストレージの読み出しの場合、これは、システム・メモリにおいて今やアダプタからのデータ読み出しが利用可能であり、処理できるという表示である。

#### 【0072】

上記の例において、アダプタ割り込みソース識別情報を用いて、オペレーティング・システム（例えば、非ゲスト・オペレーティング・システム）に、アダプタ割り込みが提示されたアダプタ・タイプを知らせる。このことにより、オペレーティング・システムは、どのアダプタ・イベント・インジケータ・サブセットを、具体的には、割り込みが提示されたタイプのアダプタと関連したインジケータのみを識別することが可能になる。

#### 【0073】

本発明の更に別の態様によると、アダプタ割り込みソース識別情報は、例えば論理パーティション内で実行しているホストのゲスト（例えば、ページング可能ストレージ・モード・ゲスト）により用いることもできる。このことは、如何なるホストの介入なしに行われる、ページング可能ゲストについては、上述のゲスト割り込み状態領域（G I S A）は、アダプタ割り込みが、特定のゲストに対してファームウェアにより保留状態にされるホスト制御ブロックである。一例において、このことは、別途を、ゲスト割り込みサブクラスに対応する割り込み保留マスク（I P M）内に設定することによって達成される。この処理に関する更なる詳細を以下に説明する。

#### 【0074】

一例において、M S Iを、ゲストに提示されるアダプタ・イベント通知に変換するために、特定の初期化が行われる。この例においては、ホスト初期化及びゲスト初期化が行われる。例えば、ホスト初期化の際（又は、第1のP C I機能がゲストに割り当てられるとき）、ホストは転送A I S Bアレイ及びG A I Tを割り当てる。次いで、ホストは、例えばアダプタ割り込み転送テーブル（A I F T）内に、転送A I S Bアレイ及びG A I Tの位置及び長さを登録する。一例においては、割り込み制御設定命令を用いて、転送A I S Bの位置及び長さを登録する。

10

20

30

40

50

## 【 0 0 7 5 】

さらに、ホストは、ゲストに割り当てられる P C I アダプタに割り当てられるホスト割り込みサブクラスを指定する。ここでも、一例において、割り込み制御設定命令などの命令を用いて、この情報を指定する。この情報はまた、ホストが実行しているパーティションに関する A I F T エントリ内にも保持される。これでホスト初期化が終了する。

## 【 0 0 7 6 】

ゲスト初期化の際、ゲストは、M S I 要求を介して、アダプタ・イベント通知に関するその P C I 機能を構成するために、多数のタスクを実行する。一例において、これらの機能と呼び出す 1 つ又は複数の命令により、ホストへのインターセプトがもたらされ、従って、以下で説明されるように、ホストは、各々のインターセプトに対するアクションを取る。

10

## 【 0 0 7 7 】

最初に、ゲストは、ゲストがアクセスできる P C I 機能及び構成を判断する。一例において、ゲストは、P C I 機能のリストを取得するために命令（例えば、リスト照会命令）を発行し、この命令はホストによりインターセプトされる。ホスト初期化の際、ホストは、どの P C I 機能がホストに割り当てられているかを既に判断しているので、P C I 機能に関するゲスト要求のインターセプトに回答して、ホストは、ゲストへの回答を構成し、これをゲストに戻し、ゲストに割り当てられた P C I 機能のみを含む。

## 【 0 0 7 8 】

その後、ゲスト構成の各 P C I 機能について、特定の処理が行われる。例えば、P C I 機能に使用される M S I アドレス、及び、P C I 機能によりサポートされる M S I ベクトルの数についての判断がなされる。一例において、M S I アドレスは、アダプタ機能のグループに共通の特定を提供するグループ照会（Query Group）命令を用いて判断され、P C I 機能によりサポートされる M S I ベクトルの数は、アダプタの能力に基づいている。ホストは、その初期化の際、この情報を判断しており、従って、ゲスト・コマンドのインターセプトに回答して、ホストは M S I アドレス及び最大数の M S I ベクトルを含む回答を構成し、これをゲストに戻す。

20

## 【 0 0 7 9 】

付加的に、A I B V が割り当てられ、もしあれば、同様に A I S B も割り当てられる。A I B V 及び A I S B は割り当てられ、ゼロにクリアされ、アダプタ割り込み登録操作が指定される。要求されたアダプタ割り込み登録操作に回答して、ホストが操作をインターセプトし、登録を実施する。このことは、例えば、ホスト・ストレージ内にゲスト A I B V をピンニングする（つまり、ホスト・メモリ内にゲスト・ページを固定し、これをページング不可能にする）ことを含む。さらに、ゲストが A I S B を指定した場合、ホストはまた、ホスト・ストレージ内にゲスト A I S B もピンニングする。ホストは、A I S B を、転送 A I S B アレイ及び暗黙的に対応する G A I T エントリから P C I 機能に割り当てる。代替的に、ゲストにより指定された A I S B 及び I S C は、（別の P C I 機能について）ゲストが以前に登録した同じ A I S B 及び I S C である場合、ホストは、その以前の要求に対して割り当てられた同じ転送 A I S B 及び G A I T エントリを使用することができる。これにより、オーバーヘッドが低減される。ホストは、ゲスト割り込みサブクラスを G A I T アレイにコピーする。ゲストが A I S B を指定していた場合、ホストは、ゲスト A I S B のホスト・アドレス及びそのオフセットを G A I T エントリにコピーする。ホストは、ゲストの G I S A 指定を、その状態記述から G A I T エントリにコピーする。さらに、その状態記述からのゲストの G I S A の表示を G A I T エントリにコピーする。

30

40

## 【 0 0 8 0 】

ゲストの代わりに、ホストは、P C I 機能制御修正命令などの命令を実行して、アダプタ割り込み登録操作を指定し、以下の情報を示す。：すなわち、ゲスト A I B V のホスト・アドレス及びゲスト・オフセット、ホスト A I S B 及びアダプタに割り当てられた転送 A I S B アレイのホスト・アドレス及びオフセット、アダプタに関するホスト割り込みサブクラス、及びゲストが指定した M S I の数である。P C I 機能制御修正命令の実行に

50

答して、初期化が実施されている P C I 機能に対応するデバイス・テーブル・エントリが選択され、デバイス・テーブル・エントリ内に種々のパラメータが格納される。例えば、ゲスト A I B V、ホストにより選択された転送 A I S B、ホスト I S C、及び割り込みの数が、機能の構成から得られた値に設定される。さらに、例えば、ゲスト A I S B のホスト・アドレス及びオフセット、ゲスト I S C、及びゲストに関する G I S A のアドレスを含む、種々の情報が G A I T 内に登録される。これで登録プロセスが完了する。

**【 0 0 8 1 】**

その後、P C I 機能の構成空間が書き込まれる。具体的には、M S I アドレス及び M S I ベクトル・カウントが、以前の登録と一致するように、アダプタ機能の構成アドレス空間に書き込まれる。これでゲスト初期化が完了する。

10

**【 0 0 8 2 】**

初期化の実施に続いて、受信した M S I が I / O アダプタ・イベント通知に変換される。最初に、割り込みを要求するイベントの記述を記録する。付加的に、M S I アドレス及び M S I ベクトル数、並びにリクエスト I D を指定する要求が、P C I 機能により開始される。この要求は I / O ハブにより受信され、要求の受信に回答して、要求内のリクエスト I D を使用して、P C I 機能に関するデバイス・テーブル・エントリを見つけ出す。I / O ハブは、要求内のアドレスの少なくとも一部を、M S I 比較レジスタ内の値と比較する。それらが等しい場合、M S I が要求されている。

**【 0 0 8 3 】**

その後、要求内で指定された M S I ベクトル数が、この機能について許容される割り込みの数 ( N O I ) より小さいか又はそれと等しいかどうかについての判断がなされる。M S I ベクトル数が N O I より大きい場合は、エラーが表示される。他の場合には、I / O ハブは、ビット機能設定を発行して、ストレージ内に適切な A I B V ビットを設定する。M S I ベクトル数を、デバイス・テーブル・エントリ内で指定された A I B V オフセットに加算し、このビット数を、デバイス・テーブル・エントリ内で指定された A I B V アドレスから変位させることにより、適切なビットが求められる。ホストがその割り込み情報の登録を設定する方法に基づいて、設定されるビットは、ホスト・ストレージ内でピンニングされたゲスト A I B V である。

20

**【 0 0 8 4 】**

さらに、A I S B が示されている場合、I / O ハブは、ビット設定機能を使用し、ゾーン番号 (再配置ゾーンのような)、デバイス・テーブル・エントリ内の A I S B アドレス及び A I S B オフセットを用いて A I S B を設定する。この場合も、ホストがその割り込み情報の登録を設定する方法に基づいて、設定されるビットは、ホスト・ストレージ内の転送 A I S B アレイ内のホスト A I S B である。システムが単一ビットの設定をサポートしていない場合、複数ビットを設定して (例えば、バイト)、アダプタ・イベント又はサマリ表示を示すことができることに留意されたい。

30

**【 0 0 8 5 】**

次に、一実施形態において、割り込み要求が既に保留中であるかどうかについての判断が (例えば、C P U 又は I / O ハブによって) なされる。この判断を行うために、保留インジケータが使用される。割り込み要求が既に設定されている場合、処理は完了し、別の割り込み要求は要求されない。従って、その後の割り込み要求は、既に保留中の 1 つの要求に包含される。

40

**【 0 0 8 6 】**

非同期的に、1 つ又は複数のプロセッサが保留インジケータをチェックする。特に、I S C (及びゾーン) に関してイネーブルにされた各プロセッサは、例えば、割り込みがそのプロセッサに関してイネーブルにされたとき、インジケータ上でポーリングする。プロセッサの 1 つは、インジケータが設定されていると判断した場合、プロセッサは、他のプロセッサとの仲裁を行って、割り込みを処理する。

**【 0 0 8 7 】**

割り込みを処理するために、ファームウェアは、アダプタ割り込み要求内で指定された

50

ゾーン番号を用いて、論理パーティション（ゾーン）に関する A I F T エントリを見つけ出す。A I F T エントリを見つけ出すことに応答して、ファームウェアは、A I F T エントリが所定の値（例えば、全てゼロ）を含むかどうかをチェックする。A I F T エントリが所定の値を含む場合、論理パーティションにおいてゲストを実行しているホストは存在せず、ゾーン番号により識別される論理パーティションに関して（又は、論理パーティションが構成されない場合は、オペレーティング・システムに関して）、アダプタ割り込みが保留状態にされる。次いで、この割り込みは、ゲストではないオペレーティング・システムへの割り込みの提示を参照して上述されたように処理される。

**【 0 0 8 8 】**

A I F T エントリが、1 又は複数のゲストを実行しているホストが存在することを意味する所定の値を含まない場合、処理は、アダプタ割り込み要求の一部として指定された I S C が、A I F T エントリ内の I S C と等しいかどうかのチェックを続行する。アダプタ割り込み要求の一部として指定された I S C が、A I F T エントリ内の I S C と等しくない場合、アダプタ割り込み要求はゲストをターゲットにしておらず、ゾーン番号により識別されるその論理パーティション（すなわち、ホスト）に関して保留状態にされる。次いで、処理は、ゲストではないオペレーティング・システムへの割り込みの提示を参照して上述されたように進む。

10

**【 0 0 8 9 】**

他の場合には、アダプタ割り込み要求の一部として指定された I S C が、A I F T エントリ内の I S C と等しく、アダプタ割り込み要求がゲストをターゲットにしていることを意味する。ファームウェアは、A I F T エントリ内の転送 A I S B アレイのアドレス及び長さを用いて、ホストにより指定された転送 A I S B アレイを走査し、1 に設定されたインジケータ（例えば、ビット）を探す。1 に設定された各インジケータについて、ファームウェアは、対応する G A I T エントリ内の情報を用いて、そのインジケータを処理する。

20

**【 0 0 9 0 】**

最初に、G A I T エントリが、アダプタ割り込みがゲストをターゲットにしていないことを意味する所定の値（例えば、全てゼロ）を含むかどうかについての判断がなされる。G A I T エントリが所定値を含む場合、アダプタ割り込みはホストに関して保留状態にされ、I / O 割り込みコード内のホスト・アダプタ・イベント通知転送インジケータが 1 に設定される。上述のように、割り込みがホストに提示される。

30

**【 0 0 9 1 】**

しかしながら、G A I T エントリが、アダプタ割り込みが対応するゲストをターゲットにしていることを意味する所定の値を含まない場合、多数のステップを行って、ゲストへのアダプタ・イベント通知の転送を完了する。例えば、G A I T エントリ内のゲスト A I S B アドレスが所定の値（例えば、全てゼロ）を含まない場合、ゲスト A I S B アドレス及びゲスト A I S B オフセットを用いて、ゲスト A I S B を 1 に設定する。さらに、G A I T エントリ内のゲスト割り込みサブクラス及び G I S A 指定を用いて、ゲストに関する G I S A 内の割り込みを保留状態にする。例えば、G I S A の割り込み保留マスク（I P M）内の G I S C に対応するビットが 1 に設定される。さらに、本発明の態様によると、ファームウェアは、アダプタ・タイプ・ビットを、割り込みが保留状態にされたゲスト割り込みサブクラスに対応する G I S A の A I S M アレイ内の A I S M エントリに設定する。さらに、G I S C に関してホスト警告が要求される（例えば、G I S A の割り込み警告マスク（I A M）内の G I S C に対応するビットが 1 に設定される）場合、ホスト警告アダプタ割り込みが保留状態にされる。

40

**【 0 0 9 2 】**

I P M ビットの設定は C P U の保留インジケータと同等であり、ゲスト C P U が I S C に関してイネーブルにされる度に、ホストの介入なしに、割り込みがゲスト C P U に提示される。本明細書で説明されるように、I P M ビットが 1 に設定された時点で、G I S C に対応するアダプタ割り込みソース・マスク内に、P C I 機能に関してアダプタ割り込み

50

が保留状態であると指定するビットが設定される。

【0093】

I P MをG I S A内に設定することに対応して、I S Cに関してイネーブルにされたプロセッサは、このインジケータが設定されており、割り込みをオペレーティング・システムに提示すると判断する。本発明の1つ又は複数の態様に関連するこの処理は、図10を参照してさらに説明される。

【0094】

ゲストに割り込みが提示されることに対応して(ステップ500)、割り込みが提示されるゲスト割り込みサブクラスに対応する、G I S AのA I S Mフィールドが、ゲストに関するI/O割り込みコード内に配置される(ステップ502)。原子的に、G I S AにおいてA I S Mがクリアされ、保留状態のアダプタ割り込みもクリアされる(ステップ504)。その後、ゲストは割り込みを処理することができる。

10

【0095】

本発明の態様によると、ゲストは、I/O割り込みコード内に格納された情報に基づいて、アダプタのタイプを判断し(ステップ506)、上述のように、アダプタのタイプに基づいて、ゲストは、調べられるインジケータを識別する(ステップ508)。ゲストは、それらのインジケータを調べ、設定されたそれらのインジケータを処理する(ステップ510)。他のタイプのアダプタに関するインジケータは調べられず、処理されない。

【0096】

上述したように、初期化の際、割り込み制御設定命令が用いられる。この命令の一実施形態が、図11 - 図15を参照して説明される。図11に示されるように、一例において、割り込み制御設定命令600は、これが割り込み制御設定命令であると指定するオペコード602と、命令に関する操作制御610(図12)を指定する記憶位置(例えば、レジスタ)を含む第1のフィールド(フィールド1)604と、フィールド1により示される操作制御に関する割り込みサブクラス620(図13)を含む記憶位置(例えば、レジスタ)を指定する第2のフィールド(フィールド2)606と、図24Dに示されるように、以下で説明されるアダプタ割り込みパラメータ・ブロック(A I P B)630の論理アドレスを含む第3のフィールド(フィールド3)608とを含む。

20

【0097】

一例において、操作制御610は、以下のようにエンコードすることができる。：すなわち、

30

0 - 全割り込みモード設定(Set All Interruptions) : アダプタ割り込み抑制ファシリティが、指定されたI S Cに関して要求された全てのアダプタ割り込みの提示を可能にするように設定される。

1 - 単一割り込みモード設定(Set Single Interruption) : アダプタ割り込み抑制ファシリティが、指定されたI S Cに関する単一のアダプタ割り込み要求の提示を可能にするように設定される。指定されたI S Cに関するその後のアダプタ割り込み要求は抑制される。

2 - アダプタ・イベント通知解釈制御設定(Set Adapter Event Notification Interruption Control) : フィールド3により指定されるアダプタ割り込みパラメータ・ブロック内に含まれるアダプタ・イベント通知解釈制御が設定される。

40

【0098】

A I P B 630の一例が、図6Gを参照して説明される。示されるように、A I P B 630は、例えば、次のものを含む。：すなわち、

転送A I S Bアレイ・アドレス632 : このフィールドは、I/Oアダプタにより信号で送られたアダプタ割り込み要求がページング可能ストレージ・モード・ゲストをターゲットにしているかどうかを判断するために、ゲスト・アダプタ割り込みテーブル(G A I T)及び指定されたアダプタ・イベント通知転送割り込みサブクラス(A F I)と併せて使用される転送A I S Bアレイを示す。

転送A I S Bアレイ・アドレスがゼロのとき、割り込み要求のターゲットはホストであ

50

る。転送 A I S B アレイ・アドレスがゼロではないとき、割り込み要求のターゲットは、A F I 及び G A I T からさらに判断される。

ゲスト・アダプタ割り込みテーブル ( G A I T ) アドレス 6 3 4 : このフィールドは、I / O アダプタにより信号で送られたアダプタ割り込み要求がページング可能ストレージ・モード・ゲストをターゲットにしているかどうかを判断するのに使用される G A I T のアドレスを提供し、そのようなゲストがターゲットにされている場合、G A I T はまた、ゲスト A I B V の設定及びゲストへのアダプタ割り込み要求の配信のためにも用いられる。

アダプタ・イベント通知転送割り込みサブクラス ( A F I ) 6 3 6 : このフィールドは、I S C の値を示す。この I S C 上の保留中の提示可能な割り込みが、アダプタ・イベント通知転送プロセスを開始し、これにより、転送 A I S B アレイ及び G A I T のコンテンツを用いて、対応する I S C に関する適用可能な I / O アダプタからの割り込み要求のターゲット ( ホスト又はゲスト ) をさらに判断する。割り込み要求が、A F I フィールドにより指定される I S C に関する適用可能なアダプタからなされた場合、割り込みのターゲットは、ページング可能ストレージ・モード・ゲストとすることができ、転送 A I S B アレイ及び G A I T を用いて、転送 A I S B アレイ内に示されたいずれかのアダプタ・イベント通知の実際のターゲット ( ホスト又はゲスト ) を判断する。割り込み要求が、A F I フィールドにより指定される I S C 以外の I S C に関する適用可能なアダプタからなされた場合、転送 A I S B アレイ・アドレス及び G A I T アドレスは適用されず、対応する I S C に関する割り込み要求のターゲットはホストである。

転送 A I S B アレイの長さ ( F A A L ) 6 3 8 : このフィールドは、ビット単位での転送 A I S B アレイ又は G A I T エントリの単位での G A I T の長さを示す。

#### 【 0 0 9 9 】

割り込み制御設定命令の実行にตอบสนองして、フィールド 1 内で指定された操作制御に基づき、1 つ又は複数の割り込み制御が設定される。操作制御の値が全割り込みモード設定、又は単一割り込みモード設定を示している場合、フィールド 2 は、割り込み制御が設定される割り込みサブクラスを指定する値を含む。

#### 【 0 1 0 0 】

操作制御の値がアダプタ・イベント通知解釈制御設定を示す場合、第 2 のオペランド・アドレス ( フィールド 3 ) は、設定される制御を含むアダプタ割り込みパラメータ・ブロック ( A I P B ) の論理アドレスである。アダプタ割り込みパラメータ・ブロックは、ページング可能ストレージ・モード・ゲストに関するアダプタ・イベント通知ファシリティと関連した I / O アダプタから発生するアダプタ割り込み ( つまり、その転送 ) の解釈を容易にするために、ホストによって用いられる。

#### 【 0 1 0 1 】

一例において、操作制御の設定値は、ファームウェア及びオペレーティング・システムにアクセス可能な記憶位置 ( 例えば、制御ブロック ) 内に格納される。

#### 【 0 1 0 2 】

アダプタ割り込みを登録するのに用いられる P C I 機能制御修正命令に関する更なる詳細がここに説明される。図 1 6 を参照すると、P C I 機能制御修正命令 7 0 0 が、例えば、その P C I 機能制御修正命令を示すオペコード 7 0 2 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る記憶位置を指定する第 1 のフィールド 7 0 4 と、P C I 機能情報ブロック ( F I B ) 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 7 0 6 とを含む。フィールド 1 及び 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。

#### 【 0 1 0 3 】

一実施形態において、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 1 7 に示されるように、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 7 1 0 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 7 1 2

10

20

30

40

50

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 7 1 4 と、命令が所定のコードを完了したときに、命令に関する状態を提供する状態 7 1 6 とを含む。

【 0 1 0 4 】

一実施形態において、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。各アダプタ機能について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（FTE）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及び/又は他の情報を含む。一例として、機能テーブル・エントリは、以下のものを含む。：すなわち、

10

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（DTE）索引 1 . . . n：1つ又は複数のデバイス・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（DTE）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つ又は複数のデバイス・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）を処理するために用いられる情報、及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つの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つ又は複数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。

20

ビジー・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（permanent error）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。

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。

30

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ているかどうかを示す（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（RID）：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、及び機能番号を含む。

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、及び/又は1つ又は複数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（又は他の構成）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、及び、RID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、RID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RID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。

40

基底アドレス・レジスタ（BAR）（1乃至n）：このフィールドは、BAR<sub>0</sub> - BAR<sub>n</sub>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BARは、アダプタ機能内のメモリ空

50

間又は 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、すなわち、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、又は 32 ビットの I / O 空間であるかを示す。

一例において、BAR は、アダプタ機能のメモリ空間及び / 又は 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が得られる。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、及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<sub>0</sub> - SIZE<sub>n</sub> 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。

#### 【0105】

BAR 及びサイズに関する更なる詳細を以下に説明する。

1 . アダプタ機能に関して BAR が実装されない場合、BAR フィールド及び対応するサイズ・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 アドレス空間又は 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<sub>n</sub> 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<sub>n+1</sub> 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<sub>n</sub> 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<sub>n+1</sub> 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、及びハブ・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ているかどうか、又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。

#### 【0106】

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、及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、及び / 又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、又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、又はアダプタ機能に関してイネーブルにされる操作に依存し得る。

#### 【0107】

図 18 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 720 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・エントリ及び / 又は機能テーブル・エントリ (又は他の記憶位置) を更新するために用いられる。情報は、アダプタの初期化及び / 又は構成中に、及び / 又は特定のイベントにตอบสนองして、FIB 内に格納される。

#### 【0108】

機能情報ブロック (FIB) に関する更なる詳細が、図 19 を参照して説明される。一実施形態において、機能情報ブロック 750 は、以下のフィールドを含む。 : すなわち、形式 751 : 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 752 : このフィールドは、ページング可能モード・ゲストによる

10

20

30

40

50

、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。

エラー表示 754 : このフィールドは、ダイレクト・メモリ・アクセス及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き(例えば1)、アダプタ機能に関するダイレクト・メモリ・アクセス又はアダプタ割り込みを実施する間に、1つ又は複数のエラーが検出されている。

ロード/ストアのブロック化 1056 : 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示す。

PCI機能の有効性 758 : このフィールドは、アダプタ機能のイネーブルメント制御を含む。ビットが設定されるとき(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ていると考えられる。

10

アドレス空間の登録 760 : このフィールドは、アダプタ機能のダイレクト・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き(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ている。

ページ・サイズ 761 : このフィールドは、DMAによりアクセスされるページ又は他のメモリ単位のサイズを示す。

PCI基底アドレス(PBA) 762 : 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。

20

PCIアドレス制限(PAL) 764 : 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) 766 : 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のものを指定し、又は、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) 768 : 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 770 : このフィールドは、アダプタ機能に対して受け入れら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・アドレス及びアダプタ割り込みビット・ベクトル・オフセット・フィールドにより指定されるアダプタ割り込みビット・ベクトルのサイズをビット単位で定める。

30

アダプタ割り込みビット・ベクトル・アドレス(AIBV) 772 : 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 774 : 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・ベクトルのビット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) 776 : 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。

40

アダプタ割り込みサマリ・ビット・オフセット 778 : 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(FMB)アドレス 780 : 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 782 : 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 784 : 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。

命令権限付与トークン 786 : このフィールドは、ページング可能ストレージ・モード

50

・ゲストに、ホストの介入なしに P C I 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。及び、

アドレス変換形式 7 8 7 : このフィールドは、変換に用いられる最上位レベルの変換テーブル (例えば、セグメント・テーブル、領域第 3 等) のアドレス変換に関する選択された形式を示す。

#### 【 0 1 0 9 】

P C I 機能制御修正命令内で指定される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、及び / 又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、及び / 又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 / ストアのブロック化のリセット、機能測定パラメータの設定、及びインターセプト制御の設定が含まれる。

10

#### 【 0 1 1 0 】

P C I 機能制御修正命令と関連した論理の一実施形態を、図 2 0 を参照して説明する。一例において、命令が、オペレーティング・システム (又は他の構成) 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、命令及びアダプタ機能は、P C I 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・アーキテクチャ及び対応する命令を用いることができる。

20

#### 【 0 1 1 1 】

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 : すなわち、P C I 機能ハンドル、D M A アドレス空間識別子、操作制御、及び、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 つ又は複数のレジスタにおいて)。

#### 【 0 1 1 2 】

図 2 0 を参照すると、最初に、P C I 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 8 0 0 )。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 8 0 2 )。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 (又は他のゲスト) 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 8 0 4 )。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 (8 0 6 )。

30

#### 【 0 1 1 3 】

他の場合には、オペランドの 1 つ又は複数が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 8 0 8 )。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 8 1 0 )。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 8 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 8 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 8 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 8 1 8 )。

40

#### 【 0 1 1 4 】

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 (8 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

50

トリを見つけ出す。

【0115】

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822）。見つかっていない場合、例外条件が与えられる（ステップ824）。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826）、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828）。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、又は指定された場合、他の権限付与をチェックすることができる。

【0116】

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830）。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（832）。

10

【0117】

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834）。機能テーブル・エントリ内の回復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836）。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838）。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840）。ビジー条件において、命令を、ドロップしないで、再試行することができる。

20

【0118】

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（842）。例えば、FIB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844）。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846）。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（848）。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。

30

【0119】

一例において、操作制御は、アダプタの割り込みを制御するのに用いられるアダプタ割り込み登録操作である。この操作制御にตอบสนองして、機能情報ブロックの適切なコンテンツに基づいて、アダプタ割り込みに関連するアダプタ機能パラメータが、デバイス・テーブル・エントリ内に設定される。

【0120】

この操作と関連した論理の一実施形態が、図21を参照して説明される。一例として、機能情報ブロックから取得される、この操作のためのオペランドが、例えば、割り込みサブクラス（ISC）、許容された割り込み数（NOI）、アダプタ割り込みビット・ベクトル・オフセット（AIBVO）、サマリ通知（S）、アダプタ割り込みサマリ・ビット・ベクトル・オフセット（AISBVO）、アダプタ割り込みビット・ベクトル（AIBV）アドレス、及びアダプタ割り込みサマリ・ビット・ベクトル（AISB）アドレスを含む。

40

【0121】

図21を参照すると、最初に、FIB内で指定された割り込み数（NOI）がモデル依存（model-dependent）最大数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900）。大きい場合は、例外条件が与えられる（ステップ902）。しかしながら、割り込み数がモデル依存最大数より大きくない場合、割り込み数にアダプタ割り込みピッ

50

ト・ベクトル・オフセットを加算したもの (NOI + AIBVO) がモデル依存最大数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 904)。大きい場合は、例外条件が与えられる (ステップ 906)。NOI + AIBVO がモデル依存最大数より大きくない場合は、AIBV アドレス + NOI が 4 k 境界にわた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 908)。AIBV アドレス + NOI が 4 k 境界にわたる場合は、例外条件が与えられる (ステップ 910)。他の場合には、必要とされるあらゆるリソースについて、十分なリ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 912)。十分なリソースが存在しない場合は、例外条件が与えられる (ステップ 914)。

#### 【0122】

他の場合には、この機能に関して、アダプタ割り込みが既に登録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 916)。一実施形態において、このことは、(例えば、DTE / FTE 内の) パラメータの 1 つ又は複数をチェックすることにより判断される。特定的には、NOI のような割り込みと関連したパラメータをチェックする。フィールドがポピュレートされている場合、アダプタは割り込み登録されている。アダプタが既に登録されている場合は、例外条件が与えられる (ステップ 918)。他の場合には、割り込みパラメータが、FIB から取得され、機能テーブル・エントリ (FTE) (又は、他の指定された記憶位置) 及び対応するデバイス・テーブル・エントリ (DTE) 内に置かれる。また、MSI イネーブルメント・インジケータが、DTE 内に設定される (ステップ 920)。つまり、機能情報ブロックから取り出された情報に基づいて、アダプタ割り込みに関連する PCI 機能パラメータが、DTE 内に、随意的に、FTE 内に設定される。これらのパラメータは、例えば、ISC、NOI、AIBVO、S、AISBVO、AIBV アドレス、及び AISB アドレスを含む。

#### 【0123】

上記に加えて、指定することができる別の操作制御は、アダプタ割り込み登録解除操作であり、図 22 を参照してその一例を説明する。この操作により、アダプタ割り込みに関連するアダプタ機能パラメータがリセットされる。

#### 【0124】

図 22 を参照すると、最初に、機能ハンドルにより指定されたアダプタが、割り込みのために登録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 1000)。登録されていない場合は、例外条件が与えられる (ステップ 1002)。他の場合には、機能テーブル・エントリ (又は他の記憶位置) 及び対応するデバイス・テーブル・エントリ内の割り込みパラメータがゼロに設定される (ステップ 1004)。一例において、これらのパラメータは、ISC、NOI、AIBVO、S、AISBVO、AIBV アドレス、及び AISB アドレスを含む。

#### 【0125】

上記で詳細に説明されたのは、割り込みを提示するアダプタのタイプに基づいて割り込み処理を調整する能力である。オペレーティング・システムは、割り込みコードからアダプタのタイプを判断し、その情報を用いて、そのアダプタ・タイプに関するイベントのみを調べ、処理する。オペレーティング・システムが、全ての割り込みの全てのアダプタ・タイプに関して全てのアダプタ・イベント・インジケータを処理する必要はないので、これらの能力の 1 つ又は複数により、アダプタ・イベントの処理のためのオーバーヘッドが低減される。

#### 【0126】

本明細書で説明される実施形態において、アダプタは、PCI アダプタである。本明細書で用いられる PCI という用語は、これらに限定されるものではないが、PCI 又は PCIE を含む、Peripheral Component Interconnect Special Interest Group (PCI-SIG) により定められるような PCI ベースの仕様 ([www.pcisig.com/home](http://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、Peripheral Component I

10

20

30

40

50

nterconnect Express (PCIe) は、I/Oアダプタとホスト・システム間の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。PCIe通信は、PCIeバス上での伝送のためのPCIe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。PCIeトポロジーは、対にされて(例えば、1つの上がり方向リンク、1つの下り方向リンク)PCIeバスを形成する、ポイント・ツー・ポイントの単方向リンクに基づいている。PCIe標準は、PCI-SIGにより保持され、公開されている。

10

## 【0127】

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、又はコンピュータ・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)、又は、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」又は「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つ又は複数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。

## 【0128】

1つ又は複数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、赤外線又は半導体のシステム、装置若しくはデバイス、又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つ又は複数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(RAM)、読み出し専用メモリ(ROM)、消去可能なプログラム可能読み出し専用メモリ(EPROM又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(CD-ROM)、光記憶装置、磁気記憶装置、又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、又はそれらと接続して用いるためにプログラムを収容又は格納することができるいづれかの有形媒体とすることができる。

20

30

## 【0129】

ここで図23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1100が、例えば、本発明の1つ又は複数の態様を提供し、容易にするように、コンピュータ可読プログラム・コード手段又は論理1104をその上に格納するための1つ又は複数のコンピュータ可読ストレージ媒体1102を含む。

## 【0130】

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、RF等、又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。

40

## 【0131】

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、Java、SmallTalk、C++等のようなオブジェクト指向型プログラミング言語、及び、「C」プログラミング言語、アセンブラ、又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つ又は複数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

50

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、又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（LAN）若しくは広域ネットワーク（WAN）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、又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）。

#### 【0132】

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）及びコンピュータ・プログラム製品のフローチャート図及び／又はブロック図を参照して、本明細書で説明される。フローチャート図及び／又はブロック図の各ブロック、並びにフローチャート図及び／又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、又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、コンピュータ又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、フローチャート及び／又はブロック図の1つ又は複数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

#### 【0133】

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、又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、フローチャート及び／又はブロック図の1つ又は複数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。

20

#### 【0134】

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、又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、又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、コンピュータ又は他のプログラム可能装置上で実行される命令が、フローチャート及び／又はブロック図の1つ又は複数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することもできる。

30

#### 【0135】

図面内のフローチャート及びブロック図は、本発明の種々の実施形態によるシステム、方法及びコンピュータ・プログラム製品の可能な実装のアーキテクチャ、機能及び動作を示す。この点に関して、フローチャート又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つ又は複数の実行可能命令を含むモジュール、セグメント、又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、ブロック図及び／又はフローチャート図の各ブロック、並びにブロック図及び／又はフローチャート図内のブロックの組み合わせは、指定された機能又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、又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。

40

#### 【0136】

上記に加えて、本発明の1つ又は複数の態様は、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又は複数の顧客に対して本発明の1つ又は複数の態様を実施するコンピュータ・コード及び／又はコンピュータ・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

50

は、例として、予約申し込み及び/又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。付加的に又は代替的に、サービス・プロバイダは、1又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。

【0137】

本発明の一態様において、本発明の1つ又は複数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つ又は複数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。

【0138】

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つ又は複数の態様を実施することができる。

10

【0139】

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つ又は複数の態様を含む。コードは、コンピュータ・システムと協働して、本発明の1つ又は複数の態様を実施することができる。

【0140】

20

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つ又は複数の態様を組み込み、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・マシーンズ・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、又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つ又は複数の態様を含み、使用し、及び/又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、アダプタ及び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・メモリ及び/又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、アダプタはPCIベースのものであるが、本発明の1つ又は複数の態様は、他のアダプタ又は他のI/Oコンポーネントと共に使用可能である。アダプタ及びPCIアダプタは、単なる例である。さらに、テーブルが説明されるが、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、こうしたデータ構造体の全てを含むものである。さらに、他のタイプの割り込みインジケータを用いることができ、本明細書で用いられるDTE、FTE及び他の構造体は、より多くの、より少ない、又は異なる情報を含むことができる。多くの他の変形が可能である。

30

【0141】

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つ又は複数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又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・コードを格納及び/又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

【0142】

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、及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、又は介在するI/Oコントローラを通して、シス

50

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・ネットワーク又は公衆ネットワークを通して、他のデータ処理システム又はリモート・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・モデム及びイーサネット・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。

#### 【0143】

図24を参照すると、本発明の1つ又は複数の態様を実装するためのホスト・コンピュータ・システム5000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5000は、コンピュータ・メモリ(即ち、中央ストレージ)5002と通信状態にある1つ又は複数のCPU5001に加えて、ストレージ媒体デバイス5011及び他のコンピュータ又はSAN等と通信するためのネットワーク5010へのI/Oインターフェースを含む。CPU5001は、アーキテクチャ化された(architected)命令セット及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。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換するための動的アドレス変換(Dynamic Address Translation、DAT)5003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)5007を含むので、コンピュータ・メモリ5002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5002とプロセッサ5001との間に、キャッシュ5009が用いられる。キャッシュ5009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、命令のフェッチ及びデータ・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5009を介して、命令フェッチ・ユニット5004により、命令がメモリ5002からフェッチされる。命令は、命令デコード・ユニット5006でデコードされ、命令実行ユニット5008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、及び分岐命令実行ユニットなどの幾つかの実行ユニット5008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又はメモリからオペランドにアクセスする(ロード又はストアする)場合、典型的には、ロード/ストア・ユニット5005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、ハードウェア回路又は内部のマイクロコード(ファームウェア)において、又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。

#### 【0144】

既述のように、コンピュータ・システムは、ローカル(又は主)ストレージ内の情報、並びに、アドレッシング、保護、参照、及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、及び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。データ及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)データ及びプログラムの両方は、主ストレージにロードされる。

#### 【0145】

主ストレージは、キャッシュと呼ばれることもある、1つ又は複数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又はI/Oプロセッサと物理的に関連付けられる。物理的構成及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。

#### 【0146】

10

20

30

40

50

命令及びデータ・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・ブロック又はキャッシュ・ライン（又は短縮してライン）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、データ若しくは命令キャッシュへのストレージのプリフェッチ、又は、キャッシュからのデータの解放に影響を与える、PREFETCH DATA及びPREFETCH DATA RELATIVE LONG命令を提供することができる。

**【0147】**

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、又は64ビットである。

**【0148】**

情報は、ストレージとCPU又はチャネル・サブシステムとの間で、一度に1バイトずつ、又は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又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、又は、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。複数の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。エラー検出のため及び好ましくは訂正のために、各バイト又はバイト・グループと共に、1つ又は複数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、固定長は、1バイト、2バイト、4バイト、8バイト、又は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（又は、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。

**【0149】**

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、及び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、及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、又は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。

#### 【0150】

命令及びデータ・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。

#### 【0151】

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24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、又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、又は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、又はコンピュータ・メモリ5002からユーザに分散させても、又は、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。

#### 【0152】

ソフトウェア・プログラム・コードは、種々のコンピュータ・コンポーネント及び1つ又は複数のアプリケーション・プログラムの機能及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、及び/又は、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及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。

#### 【0153】

図25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又はサーバ・ハードウェア・システムを示す。図25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、又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つ又は複数のプロセッサ5026と、周知の技術に従ってプロセッサ5026とシステム

10

20

30

40

50

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、及びフラッシュメモリのいずれかを含む）又はテープドライブを含むことができる、メモリ5025及び長期ストレージ5027に接続する。システム5021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、及び/又はタッチ・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つ又は複数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スクリーン又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。

10

#### 【0154】

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、他のコンピュータ又はコンピュータ・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、イーサネット又はモデムである。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）、又はシステム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、又は、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及びソフトウェアは、当技術分野において周知である。

20

#### 【0155】

図26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、無線ネットワーク及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つ又は複数の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。

30

#### 【0156】

さらに図26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）、又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・コンピュータ又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つ又は複数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、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。

40

#### 【0157】

図25及び図26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ドライブ又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、又は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

50

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、又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。

**【0158】**

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・コンポーネント及び1つ又は複数のアプリケーション・プログラム5032の機能及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、及び/又は、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及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。

**【0159】**

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又はレベル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。

**【0160】**

図27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つ又は複数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、又は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（メモリ及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」。

**【0161】**

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ビット又は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 P S W 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（P S W 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。R I S C（Reduced 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、C I S C（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。I B M z / A r c h i t e c t u r e（登録商標）の命令は、2 バイト、4 バイト、又は 6 バイトの長さを有する C I S C 命令である。例えば、コンテキスト・スイッチ操作又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 5 0 6 1 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 5 0 6 1 にロードすることにより、プログラムが判断を下すこと又はプログラム内でループすることを可能にする。

10

## 【 0 1 6 2 】

典型的には、プロセッサ 5 0 2 6 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 5 0 5 5 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、又はコンテキスト・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 1 6 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

## 【 0 1 6 3 】

次いで、フェッチされた命令が、プロセッサ 5 0 2 6 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 5 0 5 6 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 5 0 5 7、5 0 5 8、5 0 6 0 に転送する。実行ユニット 5 0 5 7 は、典型的には、命令フェッチ・ユニット 5 0 5 5 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 5 0 2 5、アーキテクチャ化レジスタ 5 0 5 9、又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 5 0 5 7 に与えられる。実行の結果は、格納された場合には、メモリ 5 0 2 5、レジスタ 5 0 5 9、又は他のマシン・ハードウェア（制御レジスタ、P S W レジスタなどのような）内に格納される。

30

## 【 0 1 6 4 】

プロセッサ 5 0 2 6 は、典型的には、命令の機能を実行するための 1 つ又は複数の実行ユニット 5 0 5 7、5 0 5 8、5 0 6 0 を有する。図 2 8 を参照すると、実行ユニット 5 0 5 7 は、インターフェース論理 5 0 7 1 を介して、アーキテクチャ化された汎用レジスタ 5 0 5 9、デコード/ディスパッチ・ユニット 5 0 5 6、ロード・ストア・ユニット 5 0 6 0、及び他のプロセッサ・ユニット 5 0 6 5 と通信することができる。実行ユニット 5 0 5 7 は、幾つかのレジスタ回路 5 0 6 7、5 0 6 8、5 0 6 9 を用いて、算術論理演算ユニット（ALU）5 0 6 6 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）、及び除算（divide）などの算術演算、並びに、論理積（and）、論理和（or）、及び排他的論理和（XOR）、ローテート（rotate）及びシフト（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

40

50

回路は、例えば条件コード及び回復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。

**【0165】**

例えば ADD 命令は、算術及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。

10

**【0166】**

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又は 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、及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 ALU 5066 もあり、浮動小数点のために設計されたもの ALU 5066 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)、又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、符号及び大きさ、1 の補数、又は 2 の補数とすることができる。2 の補数における負の値又は正の値は、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。

20

**【0167】**

図 29 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及び投機された結果に基づいて、完了されるか又は破棄される。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・フィールド又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 ALU 5074 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、又は他の回路 5073 と通信することができる。

30

40

**【0168】**

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又はエラー、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

50

ドすることが好ましい。状態情報は、例えば、ハードウェア・レジスタ又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、又はファームウェア・コード（マイクロコード、ピココード、又はライセンス内部コード（LIC））単独で又はその組み合わせで実施することができる。

#### 【0169】

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、汎用レジスタ又は専用レジスタ（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つ又は複数のレジスタ・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 / Architecture（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、又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、及び即値フィールド（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。

#### 【0170】

図30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又は別のメモリ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又は別のメモリ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、又は他の要素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、及び制御論理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。

#### 【0171】

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」及び「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つ又は複数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・テーブル及びページ・テーブルを単独で又は組み合わせで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 / Architecture

(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、及び随意的なページ・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い。DAT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TLBのエントリを用いることが可能になる。TLBの内容は、LRU(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。

**【0172】**

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、TLB、及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

**【0173】**

I/Oユニット5054(図27)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、及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。IBM(登録商標)によるSystem 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・アダプタ及びオープン・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。

20

**【0174】**

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つ又は複数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、及びアーキテクチャ化されたレジスタを含む)又はそのサブセットをエミュレートする(例えば、プロセッサ及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、ソフトウェア又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つ又は複数のエミュレーション機能nにより、本発明の1つ又は複数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又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、個々の命令又は操作を実施する。

30

**【0175】**

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、命令及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、随意的に、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、又はデコード・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又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つ又は複数のサブルーチンとして実装される。

40

**【0176】**

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、

50

z / Architecture (登録商標) の IBM (登録商標) サーバにおいて、又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、既存の及び将来の IBM (登録商標) メインフレーム・サーバにおいて、及び、IBM (登録商標) の他のマシン (例えば、Power Systems サーバ及び 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/](http://www.hercules-390.org/)を参照されたい) 又は FSI (Fundamental Software, Inc) ([www.funsoft.com/](http://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

## 【0177】

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又はネイティブ・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つ又は複数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つ又は複数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・システム及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・テーブル及びページ・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、及び 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・システム又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

## 【0178】

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」サブルーチン又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及びハードウェア・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5、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6、Davidian 他による「Decoding Guest Instruction to Di

40

50

rectly Access Emulation Routines that Emulate the Guest Instructions」という名称の特許文献7、Gorishkek他による「Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System」という名称の特許文献8、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9、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10、及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

【0179】

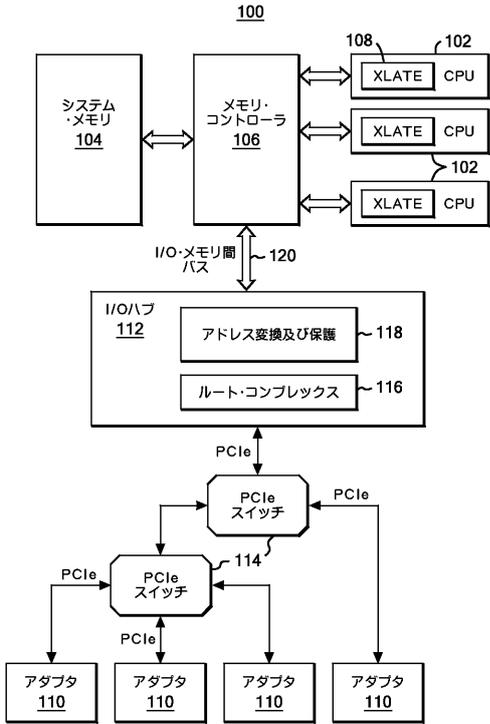
図31において、ホスト・アーキテクチャのホスト・コンピュータ・システム5000'をエミュレートする、エミュレートされたホスト・コンピュータ・システム5092の一例が提供される。エミュレートされたホスト・コンピュータ・システム5092では、ホスト・プロセッサ(CPU)5091は、エミュレートされたホスト・プロセッサ(又は仮想ホスト・プロセッサ)であり、かつ、ホスト・コンピュータ5000'のプロセッサ5091のものとは異なるネイティブな命令セット・アーキテクチャを有するエミュレーション・プロセッサ5093を含む。エミュレートされたホスト・コンピュータ・システム5092は、エミュレーション・プロセッサ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、シーケンス及びアクセス/デコード・ルーチンにおいて取得される1つ又は複数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、このシーケンス及びアクセス/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、及び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、及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及びオフ・ロード・エンジンを設けることもできる。

20

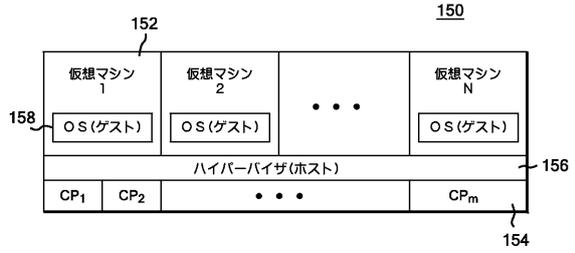
30

40

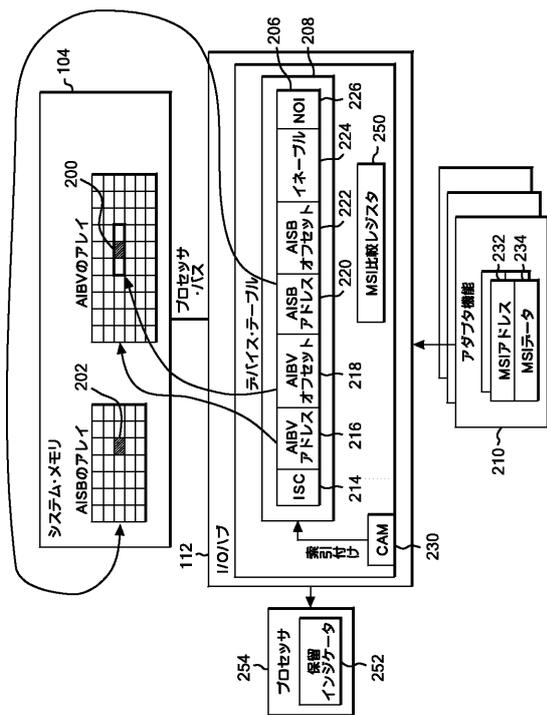
【図1】



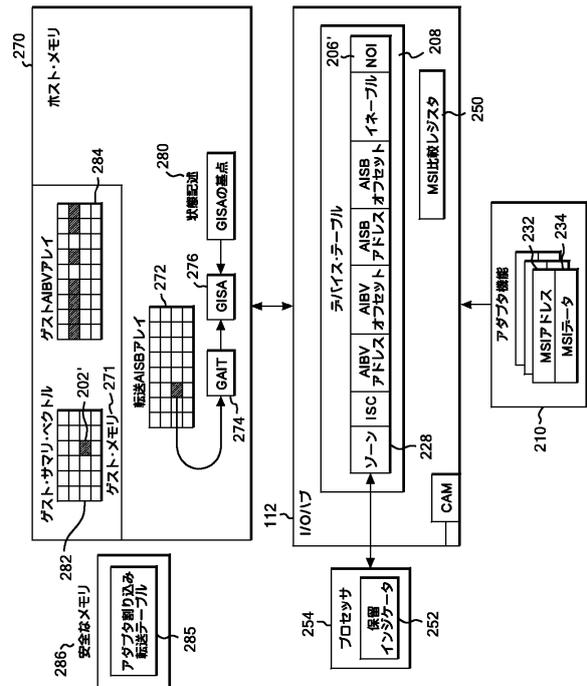
【図2】



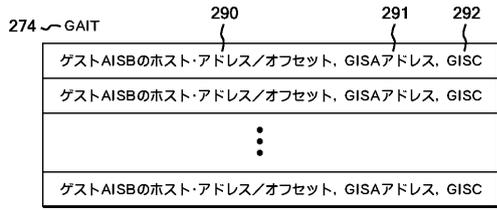
【図3】



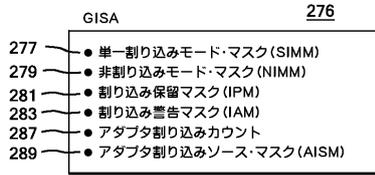
【図4】



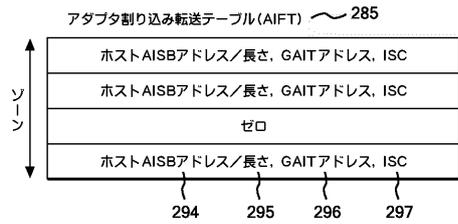
【図 5】



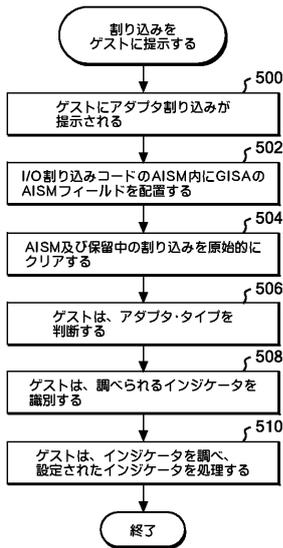
【図 6】



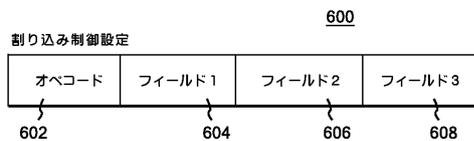
【図 7】



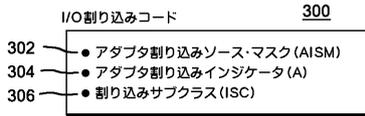
【図 10】



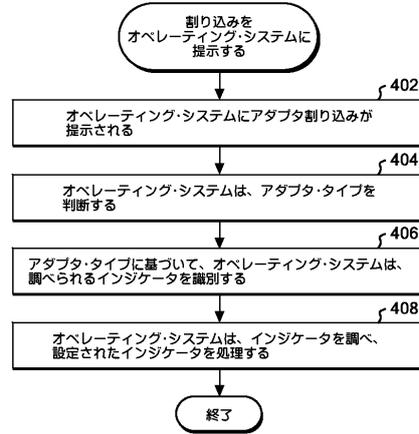
【図 11】



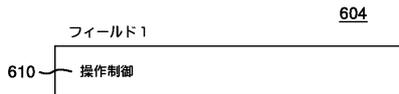
【図 8】



【図 9】



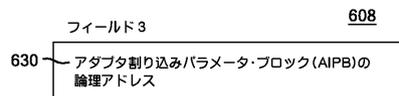
【図 12】



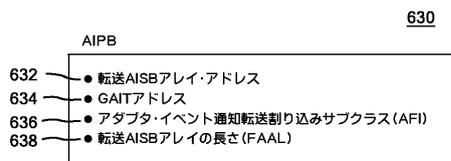
【図 13】



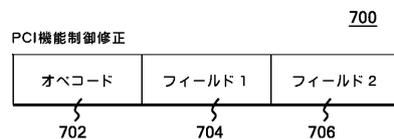
【図 14】



【図 15】



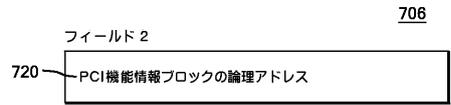
【図 16】



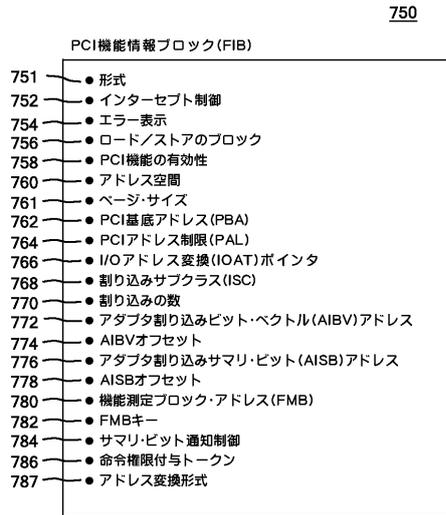
【図17】



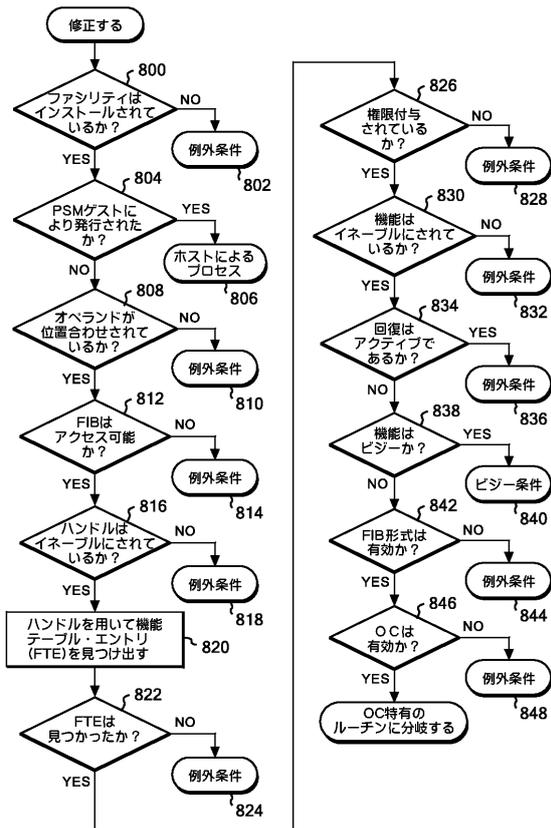
【図18】



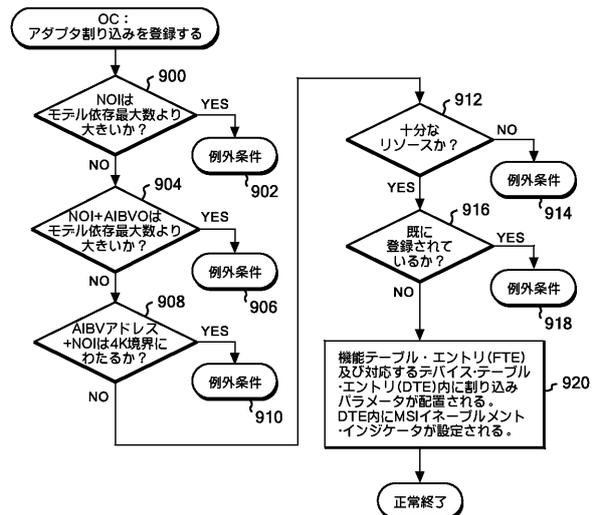
【図19】



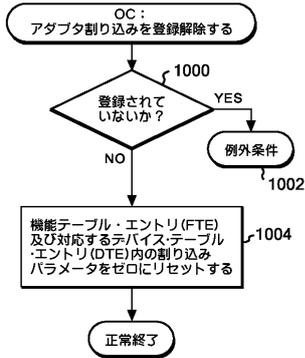
【図20】



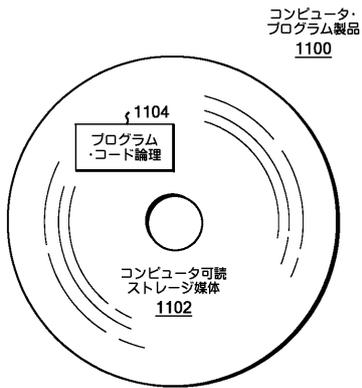
【図21】



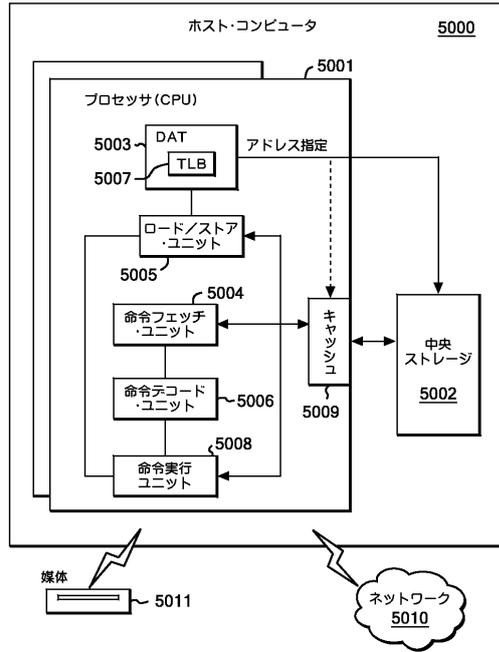
【図22】



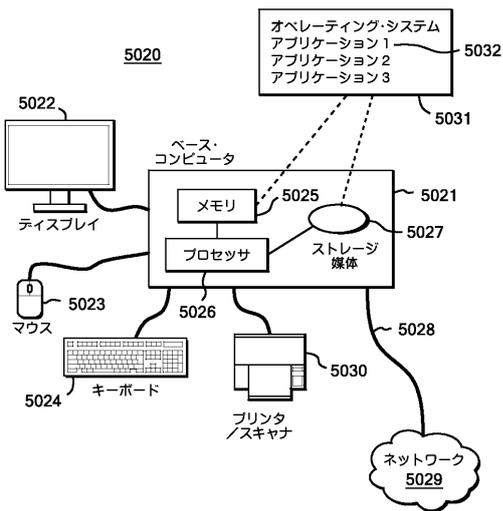
【図23】



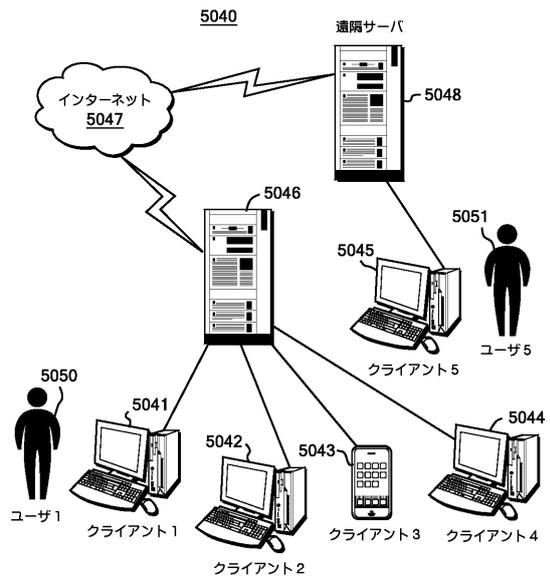
【図24】



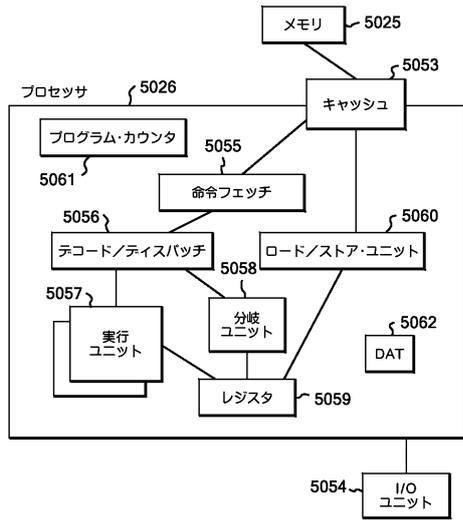
【図25】



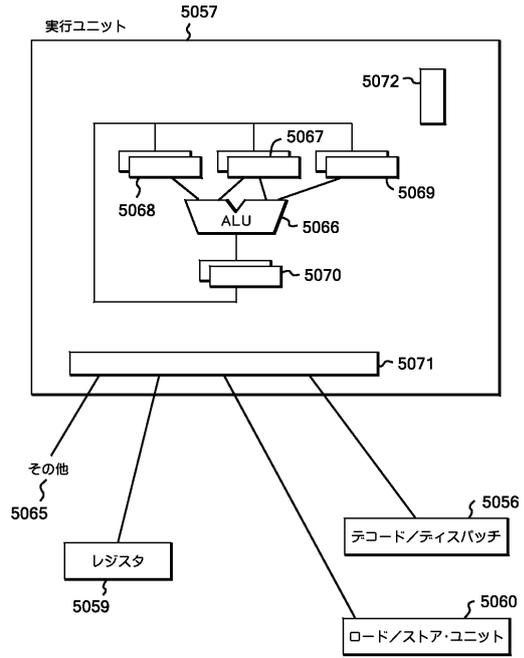
【図26】



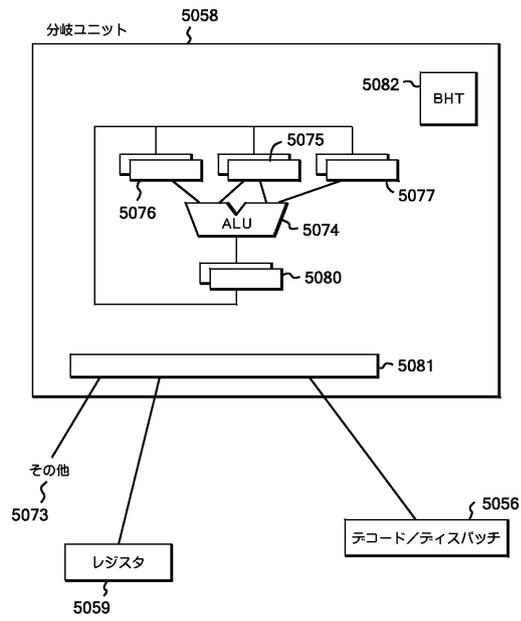
【図 27】



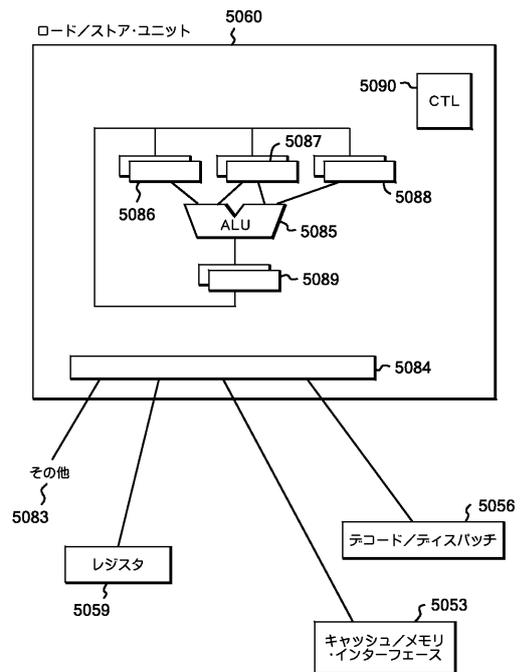
【図 28】



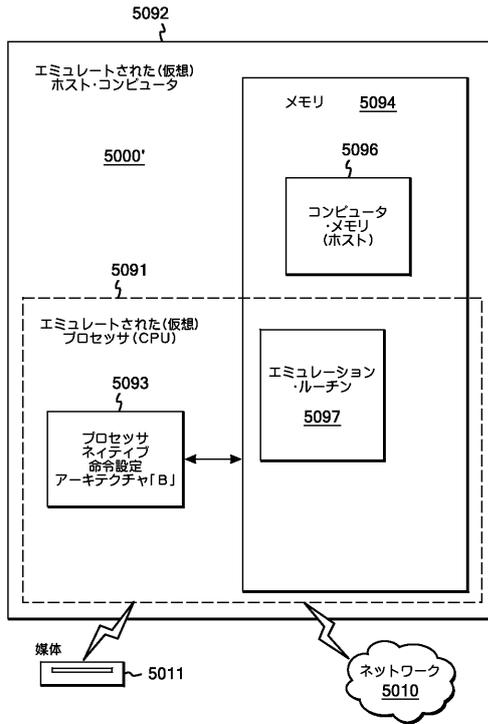
【図 29】



【図 30】



【図 3 1】



## フロントページの続き

- (72)発明者 ジットマン、グスタフ、サード  
 アメリカ合衆国 63042-2513 ミズーリ州 ヘーゼルウッド ジェームズ・エス・マク  
 ドネル・ブールバード 325 エム/ディー 301-2イー
- (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イー エム/ディー ピー318
- (72)発明者 イーストン、ジャネット  
 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 インテ  
 レクチュアル・プロパティ・ロー エムエス ピー386
- (72)発明者 ファレル、マーク  
 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 エムエ  
 ス ピー310
- (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 ピー314
- (72)発明者 シュミット、ドナルド、ウィリアム  
 アメリカ合衆国 12601-5400 ニューヨーク州 ポキプシー サウス・ロード 245  
 5 エム/ディー 7キュー4イー/ピー334
- (72)発明者 オシセック、ダミアン、レオ  
 アメリカ合衆国 13760 ニューヨーク州 エンディコット ノース・ストリート 1701  
 ビルディング 250-2 デパートメント ジー28ジー

審査官 稲葉 崇

- (56)参考文献 特開平09-034726(JP,A)  
 特開昭62-090729(JP,A)  
 特開2009-251802(JP,A)  
 特開平01-245336(JP,A)  
 特開平02-012531(JP,A)

## (58)調査した分野(Int.Cl., DB名)

G06F 13/20-13/378  
 G06F 13/10-13/14  
 G06F 9/46